

ΕΙΚΟΝΙΚΟΙ ΚΟΣΜΟΙ

*Σύγχρονες Προσεγγίσεις, Εφαρμογές και
Ανάπτυξη σε Περιβάλλον OpenSimulator*

Σπυρίδων Βοσινάκης



Ελληνικά Ακαδημαϊκά Ηλεκτρονικά
Συγγράμματα και Βοηθήματα
www.kallipos.gr

HEALINK
Σύνδεσμος Ελληνικών Ακαδημαϊκών Βιβλιοθηκών



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
ανάπτυξη της κοινωνίας της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΣΠΑ
2007-2013
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

ΣΠΥΡΙΔΩΝ ΒΟΣΙΝΑΚΗΣ
Επίκουρος Καθηγητής Πανεπιστημίου Αιγαίου

Εικονικοί Κόσμοι

Σύγχρονες Προσεγγίσεις, Εφαρμογές και Ανάπτυξη σε
Περιβάλλον OpenSimulator



Ελληνικά Ακαδημαϊκά Ηλεκτρονικά
Συγγράμματα και Βοηθήματα
www.kallipos.gr

Εικονικοί Κόσμοι

Συγγραφή

Σπυρίδων Βοσινάκης

Κριτικός αναγνώστης

Κωνσταντίνος Χωριανόπουλος

Συντελεστές έκδοσης

Γλωσσική Επιμέλεια: Ελένη Ψαθά

Γραφιστική Επιμέλεια: Σπυρίδων Βοσινάκης

Τεχνική Επιμέλεια: Σπυρίδων Βοσινάκης

Ιστότοπος συγγράμματος

<http://www.virtualworlds.gr>

ISBN: 978-960-603-226-4

Copyright © ΣΕΑΒ, 2015



Το παρόν έργο αδειοδοτείται υπό τους όρους της άδειας Creative Commons Αναφορά Δημιουργού - Μη Εμπορική Χρήση - Όχι Παράγωγα Έργα 3.0. Για να δείτε ένα αντίγραφο της άδειας αυτής επισκεφτείτε τον ιστότοπο <https://creativecommons.org/licenses/by-nc-nd/3.0/gr/>

ΣΥΝΔΕΣΜΟΣ ΕΛΛΗΝΙΚΩΝ ΑΚΑΔΗΜΑΪΚΩΝ ΒΙΒΛΙΟΘΗΚΩΝ

Εθνικό Μετσόβιο Πολυτεχνείο

Ηρώων Πολυτεχνείου 9, 15780 Ζωγράφου

www.kallipos.gr

Στη Μάγδα, στον Μανώλη και στην Ευανθία.

Πίνακας περιεχομένων

Εισαγωγή	9
Κεφάλαιο 1: Εικονικοί Κόσμοι.....	11
1 Τεχνητά περιβάλλοντα, αληθινές εμπειρίες.....	11
2 Ορισμός και βασικά χαρακτηριστικά	12
2.1 Τρισδιάστατο πολυχρηστικό περιβάλλον	13
2.2 Πλοήγηση και αλληλεπίδραση με το περιβάλλον	13
2.3 Επικοινωνία, συνεργασία και κοινωνική δικτύωση	14
3 Πλαίσια χρήσης και δυνατότητες αξιοποίησης.....	15
3.1 Επισκόπηση και εξερεύνηση τρισδιάστατου περιεχομένου	16
3.2 Προσομοίωση	17
3.3 Παιχνίδια ρόλων	17
3.4 Συνεργατική σχεδίαση	18
3.5 Ομαδικές συζητήσεις	18
3.6 Περιοχές εφαρμογής.....	19
4 Το όραμα της Εικονικής Πραγματικότητας.....	24
4.1 Ορισμός και βασικές έννοιες	24
4.2 Ιστορική αναδρομή	25
4.3 Περιβάλλοντα επιφάνειας εργασίας.....	27
5 Από τους κόσμους κειμένου στα τρισδιάστατα γραφικά	27
6 Σημερινή κατάσταση.....	29
7 Συμπεράσματα	32
Αναφορές	32
Σύνδεσμοι	34
Περαιτέρω Μελέτη.....	34
Ερωτήσεις Κατανόησης.....	35
Ασκήσεις.....	35
Κεφάλαιο 2: Απεικόνιση	37
1 Τα γραφικά στους εικονικούς κόσμους.....	37
2 Εισαγωγή στην τεχνολογία των 3D γραφικών με υπολογιστή	39
2.1 Τρισδιάστατα μοντέλα	39
2.2 Γεωμετρικοί μετασχηματισμοί.....	43
2.3 Χρώματα και υφές.....	48
2.4 Φωτισμός και σκίαση	50

2.5 Κάμερες.....	57
2.6 Άλλες τεχνικές.....	57
2.7 Διαδικασία απεικόνισης.....	58
2.8 Στερεοσκοπία.....	61
3 Δημιουργία 3Δ σκηνών.....	61
3.1 Γράφος σκηνής.....	62
3.2 Τεχνικές μοντελοποίησης.....	64
3.3 Ψηφιοποίηση.....	66
3.4 Βελτιστοποίηση για εκτέλεση σε πραγματικό χρόνο.....	67
4 Υποστηρικτικό λογισμικό.....	68
5 Συμπεράσματα.....	70
Αναφορές.....	70
Σύνδεσμοι.....	71
Περαιτέρω Μελέτη.....	72
Ερωτήσεις Κατανόησης.....	72
Ασκήσεις.....	72
Κεφάλαιο 3: Κίνηση.....	73
1 Κινούμενα αντικείμενα στους εικονικούς κόσμους.....	73
2 Μοντελοποίηση για την υποστήριξη κίνησης.....	74
2.1 Συμπαγή αντικείμενα.....	74
2.2 Παραμορφώσιμα αντικείμενα.....	76
2.3 Συστήματα σωματιδίων.....	78
2.4 Συνθετικοί χαρακτήρες.....	79
3 Κατασκευή προκαθορισμένων κινήσεων.....	80
3.1 Κίνηση με πλάνα-κλειδιά.....	81
3.2 Σύλληψη κίνησης.....	83
4 Φυσική μοντελοποίηση.....	84
4.1 Κίνηση βασισμένη στη φυσική.....	85
4.2 Αναγνώριση σύγκρουσης.....	87
4.3 Άλλες φυσικές ιδιότητες.....	89
5 Δυναμική συμπεριφορά.....	89
5.1 Μηχανές καταστάσεων.....	90
5.2 Ευφυείς πράκτορες.....	91
5.3 Εύρεση μονοπατιού.....	92
5.4 Δυναμική αποφυγή εμποδίων.....	94
5.5 Συντονισμένη κίνηση ομάδων.....	94
6 Συμπεράσματα.....	95
Αναφορές.....	96
Περαιτέρω Μελέτη.....	97

Ερωτήσεις Κατανόησης.....	98
Ασκήσεις.....	98
Κεφάλαιο 4: Διεπαφή.....	99
1 Αλληλεπίδραση χρηστών στους εικονικούς κόσμους	99
2 Πλοήγηση	100
2.1 Τεχνικές μετακίνησης	101
2.2 Εύρεση πορείας.....	104
3 Αλληλεπίδραση με αντικείμενα	105
3.1 Επιλογή αντικειμένου.....	106
3.2 Χειρισμός και απελευθέρωση.....	106
3.3 Άλλες αλληλεπιδράσεις.....	108
3.4 Περιορισμοί	109
4 Αλληλεπίδραση με άλλους χρήστες.....	110
4.1 Διαμοίραση εμπειριών	110
4.2 Επικοινωνία	110
4.3 Συνεργασία.....	112
4.4 Ομάδες χρηστών.....	112
4.5 Συνέπεια κόσμου.....	112
5 Δισδιάστατη διεπαφή – HUD	113
6 Σύγχρονες φυσικές διεπαφές.....	115
6.1 Φορητές συσκευές.....	115
6.2 Φυσική κίνηση σώματος.....	115
6.3 Εμβυθισμένη προβολή του κόσμου	116
6.4 Άλλες φυσικές διεπαφές.....	117
7 Συμπεράσματα	117
Αναφορές.....	118
Σύνδεσμοι.....	119
Περαιτέρω Μελέτη.....	119
Ερωτήσεις Κατανόησης.....	119
Ασκήσεις.....	120
Κεφάλαιο 5: Σχεδίαση, Ανάπτυξη και Αξιολόγηση.....	121
1 Εισαγωγή.....	121
2 Καταλληλότητα του μέσου	122
3 Μεθοδολογίες σχεδίασης και ανάπτυξης	123
3.1 Συνάντηση στη μέση (S. Bryson)	124
3.2 Μεθοδολογία σχεδίασης εικονικών περιβαλλόντων (C. Fencott).....	125
3.3 VRID (V. Tanriverdi & R. Jacob)	126
4 Αναπαράσταση του κόσμου και αλληλεπιδράσεις χρήστη	128
4.1 Επιλογή κατάλληλων αναπαραστάσεων.....	128

4.2 Σχεδιαστικές κατευθύνσεις με βάση τις δραστηριότητες	131
5 Αξιολόγηση ευχρηστίας	135
5.1 Χαρακτηριστικά ευχρηστίας σε εικονικά περιβάλλοντα (Gabbard)	137
5.2 Μεθοδολογία ακολουθιακής αξιολόγησης (Gabbard, Hix & Swan).....	138
5.3 Αξιολόγηση με πλατφόρμα δοκιμών (Bowman & Hodges)	139
5.4 Γνωστική περιδιάβαση (Sutcliffe & Kaur)	141
6 Συμπεράσματα	143
Αναφορές.....	143
Περαιτέρω Μελέτη.....	145
Ερωτήσεις Κατανόησης.....	145
Ασκήσεις.....	145
Κεφάλαιο 6: Εφαρμογές	147
1 Εισαγωγή.....	147
2 Εκπαίδευση	148
2.1 Εκπαιδευτικές προσεγγίσεις.....	148
2.2 Προσφερόμενες δυνατότητες.....	150
2.3 Επισκόπηση εφαρμογών	151
2.4 Καλές πρακτικές.....	153
3 Πολιτισμός	154
3.1 Χώροι πολιτισμού και εμπειρία επισκεπτών.....	155
3.2 Προσφερόμενες δυνατότητες.....	155
3.3 Επισκόπηση εφαρμογών	156
3.4 Καλές πρακτικές.....	159
4 Σχεδίαση.....	159
4.1 Η σχεδιαστική διαδικασία.....	160
4.2 Προσφερόμενες δυνατότητες.....	161
4.3 Επισκόπηση εφαρμογών	162
4.4 Καλές πρακτικές.....	164
5 Επιχειρείν	165
6 Συμπεράσματα	167
Αναφορές.....	168
Περαιτέρω Μελέτη.....	169
Ερωτήσεις Κατανόησης.....	170
Ασκήσεις.....	170
Κεφάλαιο 7: Το Περιβάλλον OpenSimulator.....	171
1 Εισαγωγή.....	171
2 Τεχνικά χαρακτηριστικά και δυνατότητες.....	172
2.1 Επίπεδο διακομιστή	172
2.2 Επίπεδο πελάτη	174

3 Εγκατάσταση του OpenSimulator	175
3.1 Προετοιμασία εγκατάστασης	176
3.2 Ενδεικτική διαδικασία εγκατάστασης	177
4 Βασικές εργασίες διαχείρισης.....	180
4.1 Έναρξη και τερματισμός διακομιστή	180
4.2 Διαμόρφωση εδάφους και προσθήκη περιοχών.....	180
4.3 Προσθήκη χρηστών.....	183
4.4 Αποθήκευση δεδομένων και αντίγραφα ασφαλείας	183
4.5 Ενεργοποίηση και τροποποίηση λειτουργιών	184
5 Οδηγίες χρήσης.....	186
5.1 Σύνδεση στον κόσμο.....	186
5.2 Πλοήγηση.....	187
5.3 Εξερεύνηση	188
5.4 Επικοινωνία	188
5.5 Αλληλεπίδραση με αντικείμενα.....	189
5.6 Αποθετήριο	189
5.7 Διαμόρφωση ενσάρκωσης	191
6 Συμπεράσματα	192
Σύνδεσμοι	193
Ασκήσεις.....	193
Κεφάλαιο 8: Κατασκευή Περιεχομένου.....	195
1 Εισαγωγή.....	195
2 Κατασκευή και επεξεργασία αντικειμένων	196
2.1 Δημιουργία και βασική επεξεργασία	197
2.2 Υλικό	200
2.3 Σύνθετα αντικείμενα	201
3 Σύνθετη μοντελοποίηση	203
3.1 Τροποποίηση μορφής βασικών στερεών	203
3.2 Παραμορφώσιμα αντικείμενα	208
3.3 Εισαγωγή εξωτερικών μοντέλων	209
4 Διαμόρφωση χώρου.....	211
4.1 Ανάγλυφο εδάφους	211
4.2 Βλάστηση	212
4.3 Κύκλος ημέρας και φωτισμός	212
5 Ενσωματώσεις και HUD	213
6 Κατασκευή συνθετικών κινήσεων χαρακτήρων	216
7 Συμπεράσματα	218
Σύνδεσμοι	219
Ασκήσεις.....	219

Κεφάλαιο 9: Εισαγωγή στη γλώσσα LSL/OSSL	221
1 Εισαγωγή.....	221
2 Σχετικά με τη γλώσσα LSL/OSSL	222
2.1 Μηχανές καταστάσεων	222
2.2 Προγραμματισμός βασισμένος σε συμβάντα.....	223
3 Εισαγωγή στον προγραμματισμό σε LSL	224
3.1 Τύποι δεδομένων, μεταβλητές και σταθερές.....	227
3.2 Συναρτήσεις, καταστάσεις και συμβάντα	229
3.3 Εντολές συνθήκης.....	229
3.4 Εντολές επανάληψης	230
4 Δράσεις των αντικειμένων	231
4.1 Κείμενο.....	231
4.2 Χρώματα και υφές.....	232
4.3 Γεωμετρικοί μετασχηματισμοί.....	235
5 Παρακολούθηση αλλαγών του περιβάλλοντος.....	239
5.1 Κλικ του ποντικιού.....	239
5.2 Παρακολούθηση μηνυμάτων κειμένου	239
5.3 Προσέγγιση και σύγκρουση.....	240
5.4 Χρονικά συμβάντα.....	243
6 Συμπεράσματα	243
Σύνδεσμοι	244
Ασκήσεις.....	244
Κεφάλαιο 10: Ανάπτυξη Σύνθετων Σεναρίων.....	247
1 Εισαγωγή.....	247
2 Επικοινωνία μεταξύ αντικειμένων	247
3 Συνθετική κίνηση	250
3.1 Ομαλή περιστροφή.....	250
3.2 Κινήσεις ενσάρκωσης	251
3.3 Κίνηση με πλάνα-κλειδιά.....	252
4 Φυσική.....	255
4.1 Ομαλή μετακίνηση	255
4.2 Δυνάμεις	255
5. Δυναμική εισαγωγή αντικειμένων.....	256
6 Άλλες αλληλεπιδράσεις	258
6.1 Διάλογοι με τον χρήστη.....	258
6.2 Κάθισμα σε αντικείμενα	260
6.3 Ενσωμάτωση αντικειμένων στο χρήστη.....	261
7 Διαχείριση λιστών και αρχείων κειμένου	262
7.1 Λίστες	262

7.2 Αρχεία κειμένου.....	265
8 Συμπεράσματα	267
Σύνδεσμοι	268
Ασκήσεις.....	268
Κεφάλαιο 11: Συνθετικοί Χαρακτήρες.....	271
1 Εισαγωγή.....	271
2 Κατασκευή και βασικές λειτουργίες.....	271
3 Χωρική συμπεριφορά.....	276
4 Κίνηση πλήθους.....	280
5 Σύνθετη συμπεριφορά χαρακτήρων.....	283
6 Συμπεράσματα	284
Ασκήσεις.....	285
Κεφάλαιο 12: Ολοκληρωμένα Παραδείγματα	287
1 Εισαγωγή.....	287
2 Ένα παιχνίδι τύπου Angry Birds.....	288
2.1 Σχεδίαση περιβάλλοντος και διεπαφής	289
2.2 Τεχνική σχεδίαση	290
2.3 Ανάπτυξη.....	293
2.4 Επεκτάσεις	300
3 Αίθουσα διαλέξεων και συνεργασίας	301
3.1 Σχεδίαση περιβάλλοντος και διεπαφής	301
3.2 Τεχνική σχεδίαση	302
3.3 Ανάπτυξη.....	304
3.4 Επεκτάσεις	311
4 Εικονική Εξιστόρηση.....	311
4.1 Σχεδίαση περιβάλλοντος και διεπαφής	312
4.2 Τεχνική σχεδίαση	313
4.3 Ανάπτυξη.....	315
4.4 Επεκτάσεις	319
5 Συμπεράσματα	320
Ασκήσεις.....	320
Ευρετήριο	323

Πίνακας συντομεύσεων-ακρωνύμια

API	Application Programming Interface
CAD	Computer Aided Design
CAE	Computer Aided Engineering
FFD	Free Form Deformation
FOV	Field of View
FPS	Frames per Second
FSM	Finite State Machine
GPU	Graphics Processing Unit
GUI	Graphical User Interface
HMD	Head Mounted Display
HSV	Hue, Saturation, Value
HUD	Heads Up Display
LSL	Linden Scripting Language
MMOG	Massive Multi User Online Game
MUD	Multi User Dungeon
NPC	Non Player Character
OSSL	OpenSimulator Scripting Language
RGB	Red, Green, Blue
SLERP	Spherical Linear Interpolation
VR	Virtual Reality
WYSIWYG	What you see is what you get

Εισαγωγή

Αντικείμενο αυτού του βιβλίου είναι οι εικονικοί κόσμοι, ένας χώρος ο οποίος έχει κινήσει το ενδιαφέρον μεγάλης μερίδας χρηστών και επιστημόνων τα τελευταία χρόνια. Οι εικονικοί κόσμοι είναι τρισδιάστατα συνθετικά περιβάλλοντα σε υπολογιστή στα οποία πολλαπλοί χρήστες που εμφανίζονται ως «ενσαρκώσεις» επικοινωνούν μεταξύ τους ή και με άλλες συνθετικές οντότητες, εξερευνούν και αλληλεπιδρούν με το περιβάλλον και κατασκευάζουν νέο περιεχόμενο. Την τελευταία δεκαετία αναπτύχθηκαν και εξαπλώθηκαν σχετικά περιβάλλοντα τα οποία έγιναν εξαιρετικά δημοφιλή στους χρήστες, όπως ο κόσμος παιχνιδιού World of Warcraft ή ο κόσμος γενικού σκοπού Second Life. Δεν είναι υπερβολή να πούμε ότι πολλοί χρήστες αντιμετωπίζουν τα περιβάλλοντα αυτά ως παράλληλη ή εναλλακτική «πραγματικότητα», καλλιεργώντας και εξωτερικεύοντας μια δεύτερη προσωπικότητα μέσω της ενσάρκωσής τους. Οι εικονικοί κόσμοι συντηρούνται και εξελίσσονται από μεγάλες και ενεργές κοινότητες χρηστών και εντός αυτών αναδύονται ιδιαίτερα κοινωνικά φαινόμενα άξια μελέτης. Δεν ήταν λοιπόν τυχαίο ότι τα περιβάλλοντα αυτά τράβηξαν την προσοχή της επιστημονικής κοινότητας όταν εξαπλώθηκαν σε μεγάλο εύρος χρηστών και συνεχίζουν να μελετώνται σε βάθος μέχρι σήμερα. Οι εικονικοί κόσμοι αποτελούν την πιο διαδεδομένη υλοποίηση των θεωριών και τεχνικών της Εικονικής Πραγματικότητας, η οποία υπάρχει και εξελίσσεται ως επιστημονική περιοχή από τη δεκαετία του '90, και πέρα από την ψυχαγωγία βρίσκουν σήμερα ποικίλες περιοχές εφαρμογής, όπως η εκπαίδευση, η προσομοίωση, ο πολιτισμός, η συνεργατική εργασία κ.ά.

Οι εικονικοί κόσμοι είναι μια περιοχή η οποία έχει έντονο το στοιχείο της διεπιστημονικότητας. Τα συστήματα αυτά αξιοποιούν τις σύγχρονες εξελίξεις στους χώρους των γραφικών με υπολογιστή, της συνθετικής κίνησης, της προσομοίωσης και της τεχνητής νοημοσύνης για την παρουσίαση του κόσμου και των αντικειμένων του, και τη μοντελοποίηση της συμπεριφοράς τους. Βασίζονται σε προηγμένες διαδικτυακές τεχνολογίες για την υποστήριξη μεγάλου αριθμού ταυτόχρονων χρηστών και τη διαχείριση πολλαπλών διασυνδεδεμένων περιοχών. Ως περιβάλλοντα διεπαφής υιοθετούν το τρισδιάστατο μοντέλο προβολής και αλληλεπίδρασης με τον κόσμο και παρουσιάζουν ενδιαφέρον τόσο ως προς τη μελέτη της ευχρηστίας των διάφορων τεχνικών και σχεδιαστικών λύσεων όσο και ως προς την επαύξησή τους με κατάλληλες συσκευές υλικού για φυσική αλληλεπίδραση και στερεοσκοπική προβολή. Η αντίληψη του συνθετικού περιβάλλοντος από τους χρήστες ως «υπαρκτό» αλλά και τα κοινωνικά φαινόμενα που αναδύονται εντός του κόσμου αποτελούν αντικείμενο μελέτης των κοινωνικών επιστημών. Τέλος, διάφοροι χώροι εφαρμογής οι οποίοι μπορούν να επωφεληθούν από τα ιδιαίτερα χαρακτηριστικά των εικονικών κόσμων, με σημαντικότερους την εκπαίδευση, τον πολιτισμό και τη σχεδίαση, εστιάζουν στη μελέτη της αποδοτικότητας του μέσου καθώς και στην αναζήτηση νέων βελτιωμένων παραδειγμάτων χρήσης.

Σκοπός αυτού του βιβλίου είναι να καλύψει ένα μεγάλο εύρος των παραπάνω περιοχών, ώστε να αποκτήσει ο αναγνώστης μια σφαιρική κατανόηση των εικονικών κόσμων, των τεχνολογιών και των εφαρμογών τους αλλά και τις δεξιότητες να σχεδιάζει και να αναπτύσσει τις δικές του λύσεις σε ένα σύγχρονο περιβάλλον. Βασικός στόχος του είναι να μάθει ο αναγνώστης τα κύρια χαρακτηριστικά και τις προσφερόμενες δυνατότητες των εικονικών κόσμων μέσα από την ιστορική εξέλιξή τους και τις σύγχρονες τάσεις, να κατανοήσει ορισμένες βασικές τεχνολογίες πίσω από τη λειτουργία τους και να γνωρίσει τις θεμελιώδεις αρχές και κατευθύνσεις στη σχεδίαση, ανάπτυξη και αξιολόγησή τους σε διάφορους χώρους εφαρμογής. Οι παραπάνω γνώσεις προσφέρονται μέσα από τη θεωρητική παρουσίασή τους αλλά και από την πρακτική εφαρμογή τους σε μια σύγχρονη τεχνολογία ανοικτού λογισμικού, το OpenSimulator.

Το βιβλίο προορίζεται κυρίως για χρήση σε προπτυχιακά προγράμματα σπουδών τμημάτων σχετικών με τον ευρύτερο χώρο της Πληροφορικής ως κύριο ή βοηθητικό σύγγραμμα. Θα μπορούσε να αξιοποιηθεί σε μαθήματα όπως Εικονική Πραγματικότητα, Γραφικά με Υπολογιστή, Προηγμένες Διεπαφές Χρήστη και Επικοινωνία Ανθρώπου-Υπολογιστή. Επιπλέον, θα μπορούσε να αποτελέσει χρήσιμο βοήθημα σε μεταπτυχιακά προγράμματα σπουδών άλλων τμημάτων στα οποία παρουσιάζει ενδιαφέρον η αξιοποίηση των εικονικών κόσμων σε κάποιο χώρο εφαρμογής, π.χ. στην εκπαίδευση, στις τέχνες ή στον πολιτισμό. Τέλος, ερευνητές ή επαγγελματίες του ευρύτερου χώρου της πληροφορικής οι οποίοι ενδιαφέρονται για τους εικονικούς κόσμους και τις τεχνολογίες τους θα μπορούσαν να αξιοποιήσουν το βιβλίο αυτό ως μια εισαγωγή στο αντικείμενο και βάση για περαιτέρω μελέτη.

Το βιβλίο χωρίζεται σε δύο μέρη. Το πρώτο μέρος (κεφάλαια 1 έως 6) είναι το θεωρητικό και έχει ως αντικείμενο τις σύγχρονες εξελίξεις στους εικονικούς κόσμους, τις τεχνολογίες και τις εφαρμογές τους. Στο πρώτο κεφάλαιο προσφέρεται μια ευρεία επισκόπηση των εικονικών κόσμων. Παρουσιάζονται οι

προσφερόμενες δυνατότητές τους, η ιστορική τους εξέλιξη, η σημερινή κατάσταση και οι μελλοντικές προοπτικές. Στο επόμενο κεφάλαιο γίνεται μια εκτεταμένη παρουσίαση του χώρου των γραφικών υπολογιστή, τα οποία αποτελούν τη βάση για την αναπαράσταση του περιβάλλοντος. Μελετάται τόσο η διαδικασία απεικόνισης όσο και η μοντελοποίηση του χώρου και των αντικειμένων. Στο τρίτο κεφάλαιο μελετούμε πώς από το σχεδιασμένο περιβάλλον παράγεται η κίνηση των αντικειμένων και των ψηφιακών χαρακτήρων του κόσμου. Αναλύονται θέματα συνθετικής κίνησης, φυσικής μοντελοποίησης και δυναμικής συμπεριφοράς χαρακτήρων. Το τέταρτο κεφάλαιο έχει ως αντικείμενο την αλληλεπίδραση του χρήστη με τους εικονικούς κόσμους και μελετώνται οι διάφορες κατηγορίες διάδρασης στον τρισδιάστατο χώρο με τις αντίστοιχες τεχνικές, καθώς και θέματα σχεδίασης της διεπαφής. Στο επόμενο κεφάλαιο μελετώνται ζητήματα σχεδίασης, ανάπτυξης και αξιολόγησης εικονικών κόσμων, όπου παρουσιάζονται και αναλύονται ορισμένες βασικές μεθοδολογίες και κατευθύνσεις της περιοχής. Το θεωρητικό μέρος του βιβλίου κλείνει με το Κεφάλαιο 6, στο οποίο παρουσιάζονται αναλυτικά τέσσερις χώροι εφαρμογής των εικονικών κόσμων και μελετώνται οι ειδικές απαιτήσεις τους, οι διάφορες σχεδιαστικές λύσεις που έχουν προκύψει καθώς και καλές πρακτικές. Τα κεφάλαια του θεωρητικού μέρους συμπληρώνονται με ερωτήσεις κατανόησης και προτεινόμενες δραστηριότητες για την καλύτερη εμπέδωση των νέων γνώσεων.

Το δεύτερο μέρος του βιβλίου (κεφάλαια 7 έως 12) είναι το πρακτικό και εστιάζει στην εκμάθηση της πλατφόρμας εικονικών κόσμων OpenSimulator και μέσω αυτής στην καλύτερη κατανόηση ορισμένων από τα θέματα που αναφέρονται στο πρώτο μέρος. Στο Κεφάλαιο 7 γίνεται μια εισαγωγή στο OpenSimulator και στις αρχές λειτουργίας του, παρουσιάζονται οι διαδικασίες εγκατάστασης και βασικής διαχείρισης και επεξηγείται η χρήση των συχνών λειτουργιών του περιβάλλοντος διεπαφής. Αντικείμενο του επόμενου κεφαλαίου είναι η χρήση του περιβάλλοντος για την ανάπτυξη του περιεχομένου του εικονικού κόσμου. Παρουσιάζονται με παραδείγματα η μοντελοποίηση αντικειμένων, η διαμόρφωση του περιβάλλοντος και η εισαγωγή συνθετικών κινήσεων χαρακτήρων. Στο Κεφάλαιο 9 εισάγεται ο αναγνώστης στη γλώσσα LSL, μέσω της οποίας μπορεί να προγραμματίσει τη διαδραστική συμπεριφορά των αντικειμένων του κόσμου. Παρουσιάζονται και επεξηγούνται οι δομές και οι εντολές της γλώσσας και αναλύονται βασικές συναρτήσεις ανάγνωσης της κατάστασης του περιβάλλοντος και επίδρασης σε αυτό. Το Κεφάλαιο 10 συνεχίζει επεκτείνοντας την παρουσίαση της γλώσσας σε πιο σύνθετες λειτουργίες και παραδείγματα, όπως η επικοινωνία μεταξύ αντικειμένων, η φυσική κίνηση και η διαχείριση αρχείων και δομών δεδομένων. Στο επόμενο κεφάλαιο παρουσιάζονται οι συνθετικοί χαρακτήρες και οι τρόποι κατασκευής και ελέγχου της συμπεριφοράς τους μέσω της γλώσσας. Δίνονται παραδείγματα σύνθετων χωρικών συμπεριφορών και κίνησης πλήθους. Το τελευταίο κεφάλαιο συνοψίζει τις γνώσεις που έχουν αποκτηθεί μέσω της αναλυτικής παρουσίασης τριών ολοκληρωμένων παραδειγμάτων στο OpenSimulator. Για καθένα από αυτά παρουσιάζεται η διαμόρφωση του περιβάλλοντος και των αντικειμένων, η σχεδίαση των αλληλεπιδράσεων και η σταδιακή ανάπτυξη και αποσφαλμάτωση της εφαρμογής. Τα κεφάλαια του πρακτικού μέρους περιλαμβάνουν επεξηγηματικά βίντεο για την επίδειξη της εκτέλεσης μέρους των παραδειγμάτων μέσα στον εικονικό κόσμο και συμπληρώνονται με ασκήσεις μεταβλητής δυσκολίας στη χρήση της γλώσσας και των εργαλείων του κόσμου.

Στην ιστοσελίδα του βιβλίου www.virtualworlds.gr μπορείτε να βρείτε πλούσιο συνοδευτικό υλικό, όπως τον κώδικα των παραδειγμάτων του βιβλίου, λύσεις των ασκήσεων, συνδέσμους σε ενδιαφέρουσες σελίδες και άρθρα και πολλά άλλα.

Κεφάλαιο 1: Εικονικοί Κόσμοι

Σύνοψη

Στόχος του κεφαλαίου είναι να παρουσιάσει μια εκτεταμένη επισκόπηση του χώρου των Εικονικών Κόσμων, που αποτελούν το αντικείμενο αυτού του βιβλίου. Αρχικά αναφέρονται οι σύγχρονοι ορισμοί των εικονικών κόσμων και των σχετικών περιβαλλόντων και περιγράφονται αναλυτικά οι προσφερόμενες λειτουργίες τους. Επιπλέον, παρουσιάζονται οι δυνατοί χώροι εφαρμογής τους και οι ιδιαιτερότητές τους σε σχέση με άλλες σχετικές προσεγγίσεις. Στη συνέχεια γίνεται μια παρουσίαση και ιστορική αναδρομή της Εικονικής Πραγματικότητας και των στόχων της. Εξηγούνται βασικές έννοιες, όπως η παρουσία και η εμπύθιση, και περιγράφονται οι βασικές κατηγορίες υλικού. Αντίστοιχα, παρουσιάζεται η εξέλιξη των Εικονικών Κόσμων από πολυχρηστικά περιβάλλοντα βασισμένα στο κείμενο μέχρι τα σημερινά τρισδιάστατα περιβάλλοντα, καταλήγοντας στις σύγχρονες τεχνολογικές λύσεις και διαθέσιμες πλατφόρμες. Τέλος, αναλύεται η σημερινή κατάσταση ως προς την αποδοχή και χρήση των εικονικών κόσμων, τόσο γενικού τύπου όσο και κόσμων παιχνιδιού, και περιγράφονται οι μελλοντικές προοπτικές.

Προαπαιτούμενη γνώση

Απαιτούνται βασικές γνώσεις πληροφορικής για την κατανόηση των εννοιών και τεχνολογιών που αναφέρονται στο παρόν κεφάλαιο.

1 Τεχνητά περιβάλλοντα, αληθινές εμπειρίες

Πολλοί συγγραφείς και σεναριογράφοι φαντάστηκαν ότι στο μέλλον η τεχνολογία θα μπορεί να δημιουργήσει τεχνητά περιβάλλοντα στα οποία οι άνθρωποι θα συναντιούνται και θα αισθάνονται το ίδιο «φυσικά» όπως στον έξω κόσμο. Για παράδειγμα, ο Neil Stephenson στο διήγημά του Snow Crash (Stephenson, 2003), που εκδόθηκε το 1992, μίλησε για το «μετα-σύμπαν» (metaverse), ένα περιβάλλον εικονικής πραγματικότητας στο οποίο «οι δρόμοι δεν υπάρχουν πραγματικά –είναι τμήματα πρωτοκόλλων γραφικών υπολογιστή γραμμένων κάπου σε χαρτί. Κανένα από τα αντικείμενα δεν είναι φυσικά κατασκευασμένο. Αποτελούν τμήματα λογισμικού που προσφέρονται στο κοινό μέσω ενός παγκόσμιου δικτύου οπτικών ινών». Σε κάποια σενάρια αυτή η ιδέα παρουσιάστηκε ως μια ενδιαφέρουσα εκδοχή του μέλλοντος που εκμηδενίζει τις αποστάσεις, ενισχύει την επικοινωνία και φέρνει την ανθρωπότητα πιο κοντά, ενώ σε άλλα ως ένα εφιαλτικό περιβάλλον στο οποίο οι άνθρωποι ζουν μια «ψεύτικη» ζωή χωρίς να γνωρίζουν ποτέ τον «αληθινό» κόσμο (π.χ. το εικονικό περιβάλλον στην ταινία Matrix, Wachowski & Wachowski, 1999). Σε όλες τις περιπτώσεις πάντως η κεντρική ιδέα είναι ότι οι άνθρωποι θα μπορούσαν να συνυπάρχουν σε ένα εναλλακτικό σύμπαν κατασκευασμένο αποκλειστικά από υπολογιστές.

Αν κοιτάξουμε για λίγο τη σημερινή κατάσταση της τεχνολογίας σε σχέση με την παραπάνω ιδέα, θα διαπιστώσουμε ότι έχουμε ακόμα πολύ δρόμο να διανύσουμε μέχρι το σημείο που θα μπορούμε να δημιουργούμε τεχνητά περιβάλλοντα της ίδιας «ποιότητας» με τον φυσικό κόσμο. Η επιστημονική περιοχή της Εικονικής Πραγματικότητας, η οποία ασχολείται με την ανάπτυξη του κατάλληλου υλικού και λογισμικού για τη δημιουργία αληθοφανών εμπειριών χρήστη σε τεχνητά περιβάλλοντα, έχει κάνει σημαντικά βήματα προς αυτή την κατεύθυνση. Όμως οι δυσκολίες εξακολουθούν να είναι πολλές. Το απαιτούμενο υλικό είναι εξαιρετικά ακριβό, έχει συνδεθεί με προβλήματα ευχρηστίας και η εξέλιξή του είναι ιδιαίτερα αργή. Τα αποτελέσματα είναι αρκετά αληθοφανή σε ό,τι αφορά την όραση και την ακοή, αλλά στο επίπεδο της αφής η κατάσταση είναι σήμερα ακόμα πρώιμη και απέχει πολύ από ένα επίπεδο πιστότητας που θα μπορούσε να χαρακτηρίσει την εμπειρία πιστευτή. Τα τεχνολογικά εξελιγμένα συστήματα εικονικής πραγματικότητας μπορεί να έχουν παρουσιάσει ελπιδοφόρα αποτελέσματα σε εργαστηριακά περιβάλλοντα και να αξιοποιούνται από οργανισμούς που διαθέτουν τα απαραίτητα χρήματα για την ανάπτυξη και χρήση τους (π.χ. βιομηχανία και στρατός), αλλά μέχρι σήμερα δεν έχουν καταφέρει να βρουν τον δρόμο τους στους οικιακούς υπολογιστές. Επομένως, ο μέσος χρήστης είναι σήμερα πρακτικά αποκλεισμένος από αυτά τα περιβάλλοντα.

Από την άλλη μεριά, αυτό που μπορούμε να παρατηρήσουμε τα τελευταία χρόνια είναι η ανάδυση όλο και περισσότερων «τεχνητών κόσμων» στους οποίους ένας πολύ μεγάλος αριθμός χρηστών αφιερώνει σημαντικό μέρος του ελεύθερου χρόνου του χωρίς τη χρήση κάποιου εξειδικευμένου υλικού (Castro-Alfonso, 2001). Στα περιβάλλοντα αυτά εμφανίζονται ρεαλιστικά ή φανταστικά τοπία και χτισμένοι κόσμοι τους οποίους

μπορεί κάποιος να εξερευνήσει και να διαδράσει με το περιεχόμενό τους. Οι χρήστες αποκτούν κάποια μορφή μέσα σε αυτούς τους κόσμους, μπορούν να επικοινωνούν, να συνεργάζονται, να δημιουργούν ομάδες, να πολεμούν και πολλά άλλα. Ένα τέτοιο παράδειγμα είναι οι πολυχρηστικοί κόσμοι παιχνιδιών, όπως το δημοφιλές World of Warcraft, όπου υπάρχει μεν ένα συγκεκριμένο περιβάλλον, μια βασική υπόθεση και έτοιμοι ρόλοι και χαρακτήρες, αλλά ο κόσμος διαμορφώνεται και εξελίσσεται μέσα από τις ενέργειες των ίδιων των παικτών. Ένα δεύτερο, ακόμα πιο ενδιαφέρον παράδειγμα είναι οι γενικού τύπου «κοινωνικοί» κόσμοι, όπως το Second Life, στους οποίους δεν υπάρχει προκαθορισμένο πλαίσιο χρήσης: οι ίδιοι οι χρήστες δημιουργούν τους ρόλους και το περιβάλλον που επιθυμούν και δίνουν νόημα στις δράσεις τους, π.χ. διασκέδαση, εκπαίδευση, διαφήμιση κ.λπ. Το κοινό στοιχείο και στις δύο περιπτώσεις είναι ότι παρουσιάζεται κάποιος κόσμος ο οποίος εξελίσσεται μέσα από τις ενέργειες των ίδιων των χρηστών και αναδύονται κοινωνικά φαινόμενα αντίστοιχα με αυτά του πραγματικού κόσμου (Messinger κ.ά., 2009).

Τα παραδείγματα που αναφέραμε παραπάνω ανήκουν στην κατηγορία περιβαλλόντων στους οποίους αποδίδεται σήμερα ο όρος «εικονικοί κόσμοι», που αποτελεί και το αντικείμενο αυτού του βιβλίου. Ο όρος μπορεί αρχικά να ξενίζει, εφόσον σε τεχνολογικό επίπεδο τα συστήματα αυτά απέχουν πολύ από την πλήρη αντικατάσταση των αισθήσεων από εικονικές. Όμως, παρά το γεγονός ότι η διάδραση με τους κόσμους αυτούς γίνεται συνήθως μέσω ενός τυπικού οικιακού υπολογιστικού συστήματος, πόσο εσφαλμένη είναι η διατύπωση ότι οι χρήστες αυτού του τύπου των περιβαλλόντων «είναι παρόντες σε έναν εικονικό κόσμο» κατά τη διάρκεια αλληλεπίδρασής τους με αυτά; Μήπως όταν διαβάζουμε ένα καλό βιβλίο ή όταν βλέπουμε μια ενδιαφέρουσα ταινία δεν μας δημιουργούνται πραγματικά συναισθήματα παρά το γεγονός ότι η ιστορία είναι εν γνώσει μας κατασκευασμένη; Ακόμα περισσότερο, όταν παίζουμε ένα εθιστικό παιχνίδι δεν είναι το μυαλό μας κυρίως συγκεντρωμένο στη δράση εντός του (φανταστικού) κόσμου του παιχνιδιού και λιγότερο σε αυτά που συμβαίνουν στον πραγματικό κόσμο γύρω μας; Όπως θα δούμε στη συνέχεια του κεφαλαίου, ένας καλά σχεδιασμένος εικονικός κόσμος μπορεί να είναι αντίστοιχα επιτυχημένος, δηλαδή να δημιουργεί στον επισκέπτη την αίσθηση ότι είναι κι αυτός μέρος του κόσμου και όχι απλά εξωτερικός παρατηρητής και ότι μαζί του στον ίδιο χώρο βρίσκονται και άλλοι επισκέπτες. Επιπλέον, σε αντίθεση με τα μη διαδραστικά μέσα, όπως το βιβλίο ή ο κινηματογράφος, σε έναν εικονικό κόσμο ο επισκέπτης έχει τη δυνατότητα να επενεργεί στα αντικείμενα του χώρου, να κοινωνικοποιείται και να συμβάλλει στην εξέλιξη του περιβάλλοντος.

2 Ορισμός και βασικά χαρακτηριστικά

Ο όρος Εικονικός Κόσμος (Virtual World), όπως χρησιμοποιείται σήμερα, αναφέρεται σε τρισδιάστατα περιβάλλοντα μεγάλου αριθμού χρηστών κατασκευασμένα σε υπολογιστή, στα οποία οι χρήστες παρουσιάζονται ως «ενσαρκώσεις» (avatars) και μπορούν να εξερευνούν ελεύθερα το περιβάλλον, να χειρίζονται τα αντικείμενά του, να δημιουργούν νέο περιεχόμενο, να συναντούν και να αλληλεπιδρούν με άλλους χρήστες ή και συνθετικές οντότητες και γενικότερα να δρουν και να επικοινωνούν με τρόπους ανάλογους με αυτούς του πραγματικού κόσμου. Θα προσπαθήσουμε στη συνέχεια να αναλύσουμε ένα ένα τα στοιχεία που αναφέρονται στον παραπάνω ορισμό για να γνωρίσουμε καλύτερα τα ιδιαίτερα χαρακτηριστικά των εικονικών κόσμων. Τα χαρακτηριστικά αυτά συνοψίζονται στον πίνακα 1.1.

Βασικά Χαρακτηριστικά	Επεξήγηση
τρισδιάστατο περιεχόμενο	ποιοτική οπτική απόδοση περιβάλλοντος, αντικειμένων και χρηστών σε τρεις διαστάσεις
περιβάλλον	κανόνες, περιορισμοί, φυσική μοντελοποίηση, τεχνητή νοημοσύνη
διάρκεια	το περιβάλλον υπάρχει και εξελίσσεται ανεξάρτητα από την παρουσία χρηστών
πολυχρηστικότητα	πολλαπλοί χρήστες συνυπάρχουν, γίνονται αντιληπτοί ως «ενσαρκώσεις»
πλοήγηση	ελεύθερη μετακίνηση στο περιβάλλον, φυσικό βάδισμα, εξερεύνηση
χειρισμός αντικειμένων	αντικείμενα που λειτουργούν ως εργαλεία, πλήκτρα, συσκευές, όπλα, ρούχα κ.λπ.
δημιουργία περιεχομένου	διαμόρφωση χώρου, κατασκευή αντικειμένων, ορισμός συμπεριφοράς τους
επικοινωνία	χωροθετημένη επικοινωνία, κείμενο, φωνή, κατάδειξη
συνεργασία	συνύπαρξη χρηστών, αντικειμένων και εργαλείων σε κοινό τόπο, επίγνωση κατάστασης του περιβάλλοντος, από κοινού ενέργειες χρηστών
εικονική ταυτότητα	διαμόρφωση και προσωποποίηση των ενσαρκώσεων
ανάδυση εικονικών κοινοτήτων	ομάδες χρηστών με κοινά ενδιαφέροντα που μοιράζονται εμπειρίες στον εικονικό κόσμο

Πίνακας 1.1 Βασικά χαρακτηριστικά των εικονικών κόσμων και σύντομη επεξήγησή τους.

2.1 Τρισδιάστατο πολυχρηστικό περιβάλλον

Ο ορισμός αναφέρει κόσμους σε *τρεις διαστάσεις*. Παρόλο που έχουν υπάρξει στο παρελθόν περιβάλλοντα βασισμένα σε δισδιάστατα γραφικά ή μόνο σε κείμενο και έχουν χαρακτηριστεί εικονικοί κόσμοι, σήμερα ο όρος αποδίδεται μόνο σε τρισδιάστατα περιβάλλοντα. Η απεικόνιση των περιβαλλόντων αυτών γίνεται με την αξιοποίηση της τεχνολογίας των γραφικών υπολογιστή αποσκοπώντας σε όσο το δυνατόν πιο ποιοτικά αποτελέσματα. Οι κόσμοι περιλαμβάνουν συνήθως κάποιο ανάγλυφο εδάφους, νερό (θάλασσες, λίμνες, κ.λπ.), ουρανό, έναν αριθμό από κατασκευασμένα τρισδιάστατα μοντέλα που απεικονίζουν πραγματικά ή φανταστικά κτίρια και αντικείμενα, και οντότητες που «κατοικούν» σε αυτούς.

Στον ορισμό αναφέρεται ο όρος *περιβάλλον*. Πράγματι, η τρισδιάστατη γεωμετρία από μόνη της δεν είναι αρκετή για να δημιουργήσει την ψευδαίσθηση του περιβάλλοντος. Θα ήταν, για παράδειγμα, αφύσικο το να μπορεί ένας χρήστης να βαδίζει μέσα από τοίχους ή οι κάτοικοι μιας πόλης να είναι ακίνητοι σαν αγάλματα. Όπως τα αντικείμενα του φυσικού κόσμου υπακούουν σε κάποιους κανόνες (π.χ. στους νόμους της φυσικής) και οι ζωντανοί οργανισμοί γύρω μας κινούνται και αντιδρούν σε εξωτερικά ερεθίσματα, το ίδιο περιμένουμε να συμβαίνει και σε ένα εικονικό περιβάλλον για να γίνει πιστευτό. Συνεπώς, θα πρέπει να υπάρχουν κανόνες που ρυθμίζουν τον τρόπο με τον οποίο τα αντικείμενα κινούνται και αλληλεπιδρούν αλλά και τη συμπεριφορά των οντοτήτων που παρουσιάζονται ως «έμψυχες». Τεχνικές από την περιοχή της *συνθετικής κίνησης με υπολογιστή (computer animation)*, της *μοντελοποίησης φυσικών νόμων (physically based modeling)*, της *προσομοίωσης (simulation)* και της *τεχνητής νοημοσύνης (artificial intelligence)* χρησιμοποιούνται για να προσδώσουν αληθοφανή συμπεριφορά στο περιβάλλον. Θα πρέπει να σημειωθεί ότι υπάρχουν και πολλές περιπτώσεις εικονικών κόσμων οι οποίοι υπερβαίνουν ορισμένους φυσικούς περιορισμούς για μεγαλύτερη ευκολία στη χρήση και στην εκτέλεση ενεργειών, π.χ. δυνατότητα χρηστών να πετάξουν ή να τηλεμεταφερθούν, αντικείμενα που μπορούν να σταθούν στον αέρα χωρίς βαρύτητα κ.ά.

Ένα επιπλέον χαρακτηριστικό που έχουν οι εικονικοί κόσμοι σε σχέση με το περιβάλλον είναι η *διάρκεια (persistence)*. Αυτό σημαίνει ότι το περιβάλλον συνεχίζει να υπάρχει και να εξελίσσεται ανεξάρτητα από την παρουσία των χρηστών σε αυτό. Αν, για παράδειγμα, εξετάσουμε τα παραδοσιακά παιχνίδια υπολογιστή, θα παρατηρήσουμε ότι ο κόσμος βρίσκεται σε μια αρχική κατάσταση τη στιγμή που ξεκινάει το παιχνίδι, εξελίσσεται όσο ο χρήστης παίζει και σταματάει όταν το παιχνίδι τερματιστεί. Αν ο χρήστης έχει αποθηκεύσει κάποια προηγούμενη κατάσταση του κόσμου, μπορεί να ξεκινήσει ξανά το παιχνίδι από αυτήν, αλλιώς θα βρεθεί και πάλι στην αρχική κατάσταση. Κάτι τέτοιο δεν συμβαίνει βεβαίως στους εικονικούς κόσμους. Όταν ένας χρήστης εγκαταλείπει τον κόσμο, αυτός συνεχίζει να υπάρχει και να εξελίσσεται λόγω των ιδίων των κανόνων αλλά και των ενεργειών των υπόλοιπων χρηστών. Έτσι, είναι αρκετά πιθανό την επόμενη φορά που ο ίδιος χρήστης συνδεθεί να αντικρίσει μια διαφορετική κατάσταση του κόσμου από αυτήν που θυμόταν.

Τέλος, στον ορισμό αναφερόμαστε σε *πολυχρηστικά περιβάλλοντα* στα οποία οι χρήστες εμφανίζονται στο περιβάλλον ως *ενσάρκωσεις*. Ένα βασικό χαρακτηριστικό των εικονικών κόσμων, σε αντίθεση, για παράδειγμα, με τα δημοφιλή κοινωνικά μέσα, είναι η δυνατότητα των χρηστών να αντιλαμβάνονται την παρουσία των υπόλοιπων χρηστών και να μπορούν να έχουν σύγχρονη επικοινωνία και συνεργασία μαζί τους (Taylor, 2002). Η παρουσία αυτή γίνεται αντιληπτή μέσω της «ενσάρκωσης» (avatar), δηλαδή μιας τρισδιάστατης οντότητας την οποία ελέγχει άμεσα ο χρήστης και η οποία τον αντιπροσωπεύει στο περιβάλλον. Η άποψη του περιβάλλοντος για κάθε χρήστη, ανάλογα με το είδος του εικονικού κόσμου, είτε διαμορφώνεται μέσα από τα μάτια της ενσάρκωσής του (οπτική πρώτου προσώπου) είτε είναι εξωτερική και περιλαμβάνει και την ίδια την ενσάρκωση (οπτική τρίτου προσώπου). Η μορφή της ενσάρκωσης, ανάλογα με το είδος του εικονικού κόσμου, μπορεί να είναι ανθρώπινη ή κάποια άλλη υπαρκτή ή φανταστική οντότητα. Σε κάθε περίπτωση, αυτό που επικοινωνείται μέσω της ενσάρκωσης, πέρα από την παρουσία ενός χρήστη στο περιβάλλον, είναι η θέση στην οποία βρίσκεται, η οπτική του γωνία (άρα και το τι βλέπει εκείνη τη στιγμή) και οι ενέργειες που εκτελεί. Η συν-παρουσία πολλαπλών χρηστών σε έναν εικονικό κόσμο τούς επιτρέπει, κατά συνέπεια, να επικοινωνούν και να συμμετέχουν από κοινού σε δραστηριότητες εντός του κόσμου.

2.2 Πλοήγηση και αλληλεπίδραση με το περιβάλλον

Η βασικότερη ίσως ενέργεια που μπορεί να κάνει ένας χρήστης σε έναν εικονικό κόσμο είναι η *πλοήγηση*. Επειδή ακριβώς, όπως και στον φυσικό κόσμο, από μια μεμονωμένη οπτική γωνία δεν μπορεί κάποιος να αποκτήσει συνολική εικόνα του περιβάλλοντος και των αντικειμένων του, η πλοήγηση είναι μια απαραίτητη

ενέργεια για να μπορούν οι χρήστες να μεταφέρονται σε διάφορες περιοχές του κόσμου και να αλληλεπιδρούν με το περιεχόμενό του. Ενώ η πλοήγηση σε παραδοσιακές δισδιάστατες διεπαφές (π.χ. ιστοσελίδες) γίνεται σχετικά εύκολα με τη χρήση των μπαρών κύλισης (scrollbars) εντός της ίδιας της σελίδας και με υπερσυνδέσμους για μετάβαση σε διαφορετικές σελίδες, στα τρισδιάστατα περιβάλλοντα το θέμα της πλοήγησης είναι περίπλοκο και συνήθως συνδυάζονται παραπάνω από μία τεχνικές για βέλτιστα αποτελέσματα. Η πιο συχνή τεχνική πλοήγησης στους εικονικούς κόσμους είναι το βάδισμα της ενσάρκωσης (από άποψη πρώτου ή τρίτου προσώπου), που δίνει περισσότερο ρεαλιστικό και φυσικό αποτέλεσμα. Όμως σε αρκετές περιπτώσεις το βάδισμα δεν παρέχει την απαιτούμενη ακρίβεια και ευελιξία και έτσι χρησιμοποιούνται επιπρόσθετες τεχνικές και βοηθήματα, όπως η εξερεύνηση, η τηλεμεταφορά, το πέταγμα, ο χάρτης κάτοψης κ.ά.

Μια δεύτερη, εξίσου σημαντική για την ενίσχυση της διαδραστικότητας, κατηγορία ενεργειών χρηστών σε έναν εικονικό κόσμο είναι ο *χειρισμός των αντικειμένων*. Ορισμένα αντικείμενα στο τρισδιάστατο περιβάλλον μπορεί να έχουν διαδραστική συμπεριφορά και να δίνουν τη δυνατότητα στους χρήστες να τα χειριστούν με διάφορους τρόπους, όπως: να κάνουν απλό κλικ πάνω τους, να τα μετακινήσουν ή να τα περιστρέψουν σε μία, δύο ή τρεις διαστάσεις, να τα κρατήσουν, να τα φορέσουν κ.ά. Ανάλογα με το είδος του αντικειμένου και το τι αυτό αναπαριστά είναι αντίστοιχη και η συμπεριφορά του σε ενέργειες όπως οι παραπάνω. Μπορεί να συμπεριφέρεται ως πλήκτρο, μοχλός, ροοστάτης, χειροπιαστό αντικείμενο, ρούχο, όπλο, ηλεκτρονική συσκευή, όχημα ή οτιδήποτε άλλο έχει φανταστεί και υλοποιήσει ο σχεδιαστής του κόσμου. Σε κάποιες περιπτώσεις κόσμων, οι χρήστες μπορούν να πάρουν τα ίδια ή αντίγραφα των αντικειμένων του κόσμου και να τα αποθηκεύσουν στο προσωπικό τους αποθετήριο, να τα δώσουν σε άλλους χρήστες ή να μοιραστούν τη χρήση τους με μια ομάδα χρηστών.

Το ακόμα πιο ενδιαφέρον στοιχείο είναι ότι σε αρκετούς εικονικούς κόσμους, όπως για παράδειγμα στο δημοφιλές *Second Life*, οι χρήστες μπορούν και οι ίδιοι να *δημιουργήσουν περιεχόμενο*. Έχουν τη δυνατότητα να διαμορφώσουν το έδαφος, να χτίσουν νέα κτίρια, να κατασκευάσουν νέα αντικείμενα, να εισάγουν έτοιμες γεωμετρίες που έχουν εντοπίσει στο διαδίκτυο ή έχουν σχεδιάσει σε πρόγραμμα τρισδιάστατης μοντελοποίησης, και να καθορίσουν τη διαδραστική συμπεριφορά των αντικειμένων τους μέσω κάποιας γλώσσας προγραμματισμού. Κατά συνέπεια, στα περιβάλλοντα αυτά η ίδια η κοινότητα χρηστών δημιουργεί και εξελίσσει το περιεχόμενο σε πολύ μεγαλύτερο βαθμό σε σχέση με την ομάδα ανάπτυξης και διαχείρισης, ακολουθώντας την κυρίαρχη λογική του Web 2.0 που συναντάμε σήμερα και στα δημοφιλή κοινωνικά μέσα.

2.3 Επικοινωνία, συνεργασία και κοινωνική δικτύωση

Πέρα από τις δυνατότητες διάδρασης των χρηστών με το τρισδιάστατο περιβάλλον, ένα ακόμη ιδιαίτερο χαρακτηριστικό των εικονικών κόσμων είναι οι διάφορες μορφές *επικοινωνίας* μεταξύ των χρηστών που υποστηρίζονται. Όσοι είναι ταυτόχρονα συνδεδεμένοι σε έναν κόσμο μπορούν να συνομιλούν μέσω κειμένου (chat) ή μέσω φωνής. Σε αντίθεση όμως με άλλα κοινωνικά μέσα, η συνομιλία στους εικονικούς κόσμους είναι συνήθως *χωροθετημένη (spatialized)*. Αυτό σημαίνει ότι τα μηνύματα που στέλνει ένας χρήστης μπορούν να τα διαβάσουν μόνο όσοι βρίσκονται σε κάποια σχετικά κοντινή απόσταση από αυτόν. Αντίστοιχα προσαρμόζεται και η ένταση των φωνών των χρηστών, ανάλογα με την απόστασή τους από τον ακροατή. Αυτή η εξάρτηση της επικοινωνίας από τη χωρική τοποθέτηση των ομιλητών και ακροατών έχει αφενός ρεαλιστικότερα αποτελέσματα, αφού προκαλεί την αίσθηση της ύπαρξης ενός «τόπου», και αφετέρου επιτρέπει την ανάδυση πολλαπλών ομάδων συζητήσεων. Παράλληλα με τη λεκτική επικοινωνία, η άμεση συσχέτιση της ενσάρκωσης του χρήστη που ομιλεί με το περιεχόμενο της ομιλίας επιτρέπει την ενίσχυση της επικοινωνίας και με μη λεκτικές μορφές. Τέτοια παραδείγματα θα μπορούσαν να είναι: η στάση του σώματος της ενσάρκωσης, το σημείο στο οποίο κοιτάζει, η περιοχή ή το αντικείμενο στο οποίο δείχνει, η εκτέλεση κάποιας συνθετικής κίνησης (π.χ. χειροκρότημα, αποδοκιμασία), η αλλαγή στην έκφραση προσώπου κ.ά. Επιπλέον, αρκετοί εικονικοί κόσμοι υποστηρίζουν και τυπικά στοιχεία κοινωνικής δικτύωσης, όπως η δυνατότητα ορισμού «φίλων» και ενημέρωσης για την κατάστασή τους και η ασύγχρονη επικοινωνία μέσω μηνυμάτων.

Οι πολλαπλές μορφές επικοινωνίας, η δυνατότητα διαμοιρασμένης χρήσης αντικειμένων του κόσμου και η συν-παρουσία σε κοινό περιβάλλον αναφοράς μπορούν να αποτελέσουν μια ισχυρή βάση για τη *συνεργασία* μεταξύ των επισκεπτών ενός εικονικού κόσμου (Churchill & Snowdon, 1998). Το πλαίσιο συνεργασίας ποικίλει ανάλογα με το είδος και τις δυνατότητες του κόσμου και μπορεί να περιλαμβάνει ένα πολύ μεγάλο εύρος δραστηριοτήτων, π.χ. συναντήσεις εργασίας απομακρυσμένων στελεχών μιας επιχείρησης,

ομαδική κατασκευή και επισκόπηση περιεχομένου, συνεργατική επίλυση προβλημάτων στα πλαίσια κάποιου εκπαιδευτικού σεναρίου, ακόμα και τη συνεργασία μιας ομάδας παικτών για τη συντονισμένη επίθεση σε έναν δύσκολο εχθρό. Σε κάθε περίπτωση, οι εικονικοί κόσμοι διαθέτουν έναν αριθμό από χαρακτηριστικά που ευνοούν τη συνεργασία. Η επικοινωνία αποτελεί απαραίτητη προϋπόθεση για οποιουδήποτε είδους συνεργασία, και οι εικονικοί κόσμοι επιτρέπουν την επικοινωνία μεταξύ των ενσαρκώσεων με πολλαπλούς τρόπους, τόσο σύγχρονα, δηλαδή άμεσα κατά την ταυτόχρονη παρουσία των χρηστών στο περιβάλλον, όσο και ασύγχρονα μέσω μηνυμάτων, κατασκευασμένων αντικειμένων κ.ά. Επιπλέον, η δυνατότητα από κοινού δημιουργίας, επεξεργασίας και χρήσης αντικειμένων και οι λειτουργίες διαχείρισης των δικαιωμάτων πρόσβασης σε αυτά μπορούν να αποτελέσουν μια καλή βάση για τον καθορισμό και ανάθεση ρόλων στα μέλη μιας συνεργαζόμενης ομάδας και την ταυτόχρονη εργασία τους πάνω σε αντικείμενα ή δεδομένα του κόσμου. Το πιο σημαντικό όμως χαρακτηριστικό είναι ότι ο εικονικός κόσμος προσφέρεται ως ένας *κοινός τόπος* για συνεργασία, στον οποίο μπορεί ο κάθε χρήστης να εξερευνά, να αλληλεπιδρά και να αναφέρεται στο περιεχόμενο και στους υπόλοιπους χρήστες ενόσω δρα μέσα σε αυτόν. Το αποτέλεσμα είναι η αμεσότερη και περισσότερο διαισθητική *επίγνωση* της κατάστασης του περιβάλλοντος, της προόδου των εργασιών και του είδους των ενεργειών που εκτελεί το κάθε μέλος της ομάδας ανά πάσα στιγμή.

Τέλος, οι εικονικοί κόσμοι περιλαμβάνουν και χαρακτηριστικά κοινωνικής δικτύωσης (Davis, κ.ά., 2009). Οι χρήστες που επισκέπτονται συχνά έναν κόσμο δημιουργούν μια *εικονική ταυτότητα* μέσα σε αυτόν μέσω της ενσάρκωσής τους και της συνολικότερης συμπεριφοράς τους. Στις περισσότερες περιπτώσεις, οι εικονικοί κόσμοι παρέχουν πολλαπλές δυνατότητες προσαρμογής της εμφάνισης και των δράσεων των χρηστών σύμφωνα με τις δικές τους προτιμήσεις, οι οποίες συμβάλλουν εντέλει στην ανάδειξη της μοναδικότητας του κάθε χρήστη μέσα από την εικονική ταυτότητα που αυτός επιλέγει να δημιουργήσει (Becerra & Stutts, 2008). Πέρα όμως από χώρο ατομικής έκφρασης, οι εικονικοί κόσμοι προσφέρουν έναν αχανή χώρο επικοινωνίας και κοινωνικής συνεύρεσης με άλλους χρήστες και επομένως μπορούν να αποτελέσουν αφορμή για να έρθουν σε επαφή άνθρωποι με κοινά ενδιαφέροντα και επιδιώξεις και να συμμετέχουν σε διαμοιρασμένες εμπειρίες εντός του κόσμου. Αυτή η εγγενής δυνατότητα των εικονικών κόσμων εμπλουτίζεται σε πολλές περιπτώσεις και με τα κατάλληλα υποστηρικτικά εργαλεία, τα οποία συναντάμε συχνά σε διάφορες πλατφόρμες κοινωνικής δικτύωσης: ορισμός «φίλων», δημιουργία και διαχείριση ομάδων χρηστών, ασύγχρονα μηνύματα και χώροι συνομιλίας (forums) κ.ά. Όλα τα παραπάνω ευνοούν την ανάδυση *εικονικών κοινοτήτων*, δηλαδή μεγάλων ομάδων χρηστών με ένα ή περισσότερα κοινά ενδιαφέροντα, τα μέλη των οποίων επικοινωνούν, συνευρίσκονται και συνεργάζονται εντός του εικονικού κόσμου.

3 Πλαίσια χρήσης και δυνατότητες αξιοποίησης

Μέχρι τώρα είδαμε ορισμένα βασικά χαρακτηριστικά των εικονικών κόσμων, τουλάχιστον σύμφωνα με τη μορφή και τις προσφερόμενες λειτουργίες που έχουν σήμερα οι δημοφιλέστεροι από αυτούς. Το ερώτημα που παραμένει εκκρεμές είναι σε ποιους τομείς και με ποιους τρόπους μπορεί να φανεί χρήσιμη η συγκεκριμένη τεχνολογία. Συνεπακόλουθα θα μπορούσε κάποιος να αναρωτηθεί κατά πόσο οι εικονικοί κόσμοι είναι τελικά κάτι παραπάνω από ένα ακόμη μέσο για να περνούν οι χρήστες «ευχάριστα» τον χρόνο τους ή, ακόμα χειρότερα, κατά πόσο το μέσο αυτό συντελεί στην περαιτέρω απομόνωση των χρηστών του από τις δραστηριότητες του φυσικού κόσμου. Η θέση του συγκεκριμένου βιβλίου είναι ότι οι εικονικοί κόσμοι είναι ένα μέσο που, λόγω των ιδιαίτερων χαρακτηριστικών του, μπορεί να υποστηρίξει με επιτυχία ορισμένες ατομικές ή ομαδικές δραστηριότητες που είναι πολύ δύσκολο να επιτευχθούν με τη χρήση άλλων ανταγωνιστικών μέσων. Παραδείγματα τέτοιων δραστηριοτήτων είναι η παρουσίαση τρισδιάστατου περιεχομένου, η προσομοίωση, τα παιχνίδια ρόλων, η συνεργατική σχεδίαση, οι ομαδικές συζητήσεις. Οι εικονικοί κόσμοι ως τεχνολογία σε συνδυασμό με ένα κατάλληλα σχεδιασμένο περιβάλλον μπορούν να υποστηρίξουν όλες τις παραπάνω δραστηριότητες όχι μόνο μεμονωμένα αλλά και ως σύνολο, παρέχοντας κατ' αυτό τον τρόπο την προστιθέμενη αξία ενός ενιαίου, «ενσωματωμένου» (integrated) διαδραστικού περιβάλλοντος, ενός χώρου συνύπαρξης, δημιουργίας και εξερεύνησης. Η ιδιαιτερότητα αυτή μπορεί να οδηγήσει, όπως θα δούμε στη συνέχεια, σε πολύ ενδιαφέροντα αποτελέσματα σε περιοχές όπως η ψυχαγωγία, η εκπαίδευση, η κατάρτιση, ο πολιτισμός, η αρχιτεκτονική, η σχεδίαση, οι τέχνες, η ψυχολογία κ.ά.

Στη συνέχεια θα παρουσιάσουμε αναλυτικά τις βασικές κατηγορίες δραστηριοτήτων που μπορούν να λάβουν χώρα με επιτυχία στους εικονικούς κόσμους και τις σημαντικότερες περιοχές εφαρμογής.

3.1 Επισκόπηση και εξερεύνηση τρισδιάστατου περιεχομένου

Μια βασική δραστηριότητα η οποία μπορεί να επωφεληθεί από τη χρήση της τεχνολογίας των εικονικών κόσμων είναι η επισκόπηση και εξερεύνηση τρισδιάστατου περιεχομένου. Οι εικονικοί κόσμοι βασίζονται στην τεχνολογία τρισδιάστατων γραφικών σε πραγματικό χρόνο (real-time 3D graphics) και παρέχουν αρκετά φυσικούς/διαισθητικούς τρόπους πλοήγησης στο περιεχόμενο. Για παράδειγμα, οι χρήστες μπορούν να ελέγχουν άμεσα το βάδισμα της ενσάρκωσής τους και να περιηγηθούν σε μεγάλες κατασκευασμένες περιοχές με τρόπο που θυμίζει το πραγματικό βάδισμα. Αντίστοιχα μπορούν να «πιάνουν» και να περιστρέφουν αντικείμενα, να εστιάζουν σε αυτά ή να απομακρύνονται από αυτά και να έχουν μια εμπειρία μελέτης/εξερεύνησης ενός αντικειμένου παραπλήσια με τη φυσική. Όμως και η οπτική απεικόνιση του περιεχομένου μπορεί να είναι εξίσου ποιοτική. Οι εικονικοί κόσμοι εκμεταλλεύονται τη ραγδαία αυξανόμενη ισχύ των καρτών γραφικών και των επεξεργαστών των σύγχρονων υπολογιστών και δημιουργούν οπτικές αναπαραστάσεις αρκετά λεπτομερείς και με πολλά στοιχεία ρεαλισμού όπως φωτοσκίαση, αντανακλάσεις, σκιές, υφές κ.ά. Κατά συνέπεια, η εικόνα που προσφέρεται τόσο στην επισκόπηση αντικειμένων όσο και στην εξερεύνηση χώρων είναι αρκετά πλούσια, αντλεί στοιχεία και προσπαθεί να μιμηθεί την πραγματική εμπειρία. Στην εικόνα 1.1 απεικονίζεται μια αναπαράσταση κτιρίου του Παν/μίου Αιγαίου στον εικονικό κόσμο OpenSimulator.



Εικόνα 1.1 Αναπαράσταση κτιρίου του Παν/μίου Αιγαίου στο OpenSimulator. Εργασία φοιτητών.

Είναι αυτή η κατασκευασμένη εμπειρία ισάξια της πραγματικής; Σίγουρα όχι. Όμως δεν έχουν όλοι τη δυνατότητα πρόσβασης σε αντίστοιχα αντικείμενα και χώρους του πραγματικού κόσμου, για λόγους απόστασης, κόστους, επικινδυνότητας κ.ά., επομένως οι εικονικοί κόσμοι είναι μια ενδιαφέρουσα εναλλακτική. Και γιατί να μην έρθουμε σε επαφή με το αντικείμενο μελέτης μέσω εικόνων, βίντεο ή ψηφιακών ταινιών; θα μπορούσε να αναρωτηθεί εύλογα κάποιος. Προφανώς τα παραπάνω μέσα μπορούν να έχουν μεγαλύτερη πιστότητα στην παρουσίαση του αντικειμένου, αλλά απουσιάζει το στοιχείο της διαδραστικότητας. Απουσιάζει ο έλεγχος, η δυνατότητα του χρήστη να μελετήσει ο ίδιος το αντικείμενο με τον τρόπο και από τις οπτικές γωνίες που ο ίδιος προτιμά. Απουσιάζει το στοιχείο της εξερεύνησης, το ότι ο ίδιος ο χρήστης ανακαλύπτει σταδιακά πληροφορίες ως αποτέλεσμα δικών του ενεργειών και επιλογών και χτίζει βηματικά την κατανόησή του γύρω από το αντικείμενο. Και αυτό ακριβώς, η δυνατότητα ελέγχου και οικοδόμησης της γνώσης, είναι ένα βασικό συγκριτικό πλεονέκτημα των εικονικών κόσμων σε σχέση με τα μη διαδραστικά μέσα παρουσίασης περιεχομένου.

Πέρα όμως από την προβολή ρεαλιστικού τρισδιάστατου περιεχομένου, οι εικονικοί κόσμοι δίνουν επιπλέον τη δυνατότητα προβολής και εξερεύνησης ακόμη και *αφηρημένων ή υποθετικών δομών*. Μπορούν λοιπόν οι χρήστες να παρατηρήσουν και να εξερευνήσουν με διαισθητικό τρόπο ακόμα και δεδομένα που δεν προέρχονται από τον πραγματικό κόσμο. Στοιχεία από επιστημονικές μετρήσεις παρουσιασμένα ως τρισδιάστατες επιφάνειες, αφηρημένες έννοιες, δομές ή διαδικασίες οπτικοποιημένες με τη χρήση κατάλληλων

οπτικών «μεταφορών»” (*metaphors*), περίπλοκα συστήματα ή διεργασίες παρουσιασμένα σε απλούστερες μορφές, ακόμα και εναλλακτικές/υποθετικές εκδοχές του κόσμου μπορούν να συνυπάρχουν σε έναν εικονικό κόσμο. Ο μόνος περιορισμός είναι η φαντασία. Σε κάθε περίπτωση οι εικονικοί κόσμοι δίνουν στους σχεδιαστές την απαραίτητη δημιουργική ελευθερία να κατασκευάσουν τις αναπαραστάσεις που ταιριάζουν καλύτερα στα δεδομένα που θέλουν να επικοινωνήσουν και στους στόχους της εκάστοτε εφαρμογής.

3.2 Προσομοίωση

Μια άλλη κατηγορία δραστηριοτήτων την οποία οι εικονικοί κόσμοι μπορούν να αντιμετωπίσουν με σχετική επιτυχία είναι η ενεργός συμμετοχή των χρηστών σε περιβάλλοντα προσομοίωσης. Αφενός διαθέτουν ενσωματωμένους μηχανισμούς κίνησης, αναγνώρισης σύγκρουσης και φυσικής προσομοίωσης, αφεντέρου δίνεται η δυνατότητα στους δημιουργούς να προγραμματίσουν τη συμπεριφορά των αντικειμένων σε σχέση με την κατάσταση του περιβάλλοντος ή και τις ενέργειες των χρηστών. Κατά συνέπεια, πέρα από την εξερεύνηση χώρων και αντικειμένων, όπως αναφέρεται παραπάνω, οι χρήστες συμμετέχουν σε ένα «ζωντανό» περιβάλλον με το οποίο μπορούν να αλληλεπιδράσουν και να παρατηρήσουν άμεσα το αποτέλεσμα. Ένα περιβάλλον που εξελίσσεται με τον χρόνο και μπορεί να προσομοιώνει πραγματικά, υποθετικά ή φανταστικά φαινόμενα και καταστάσεις. Ο επισκέπτης συμμετέχει λοιπόν σταδιακά σε διαδικασίες παρατήρησης, εξερεύνησης, ελεύθερου πειραματισμού, κατανόησης, εξάσκησης και τελικά μάθησης του αντικειμένου προσομοίωσης και των σχετικών με αυτό εννοιών (Chaturvedi κ.ά., 2011). Υπό την προϋπόθεση, βεβαίως, ότι το περιβάλλον είναι προσεκτικά σχεδιασμένο, ώστε να μπορεί να υποστηρίξει τα παραπάνω με έναν εύχρηστο και διαισθητικό τρόπο.

Και στην περίπτωση της προσομοίωσης, η εναλλακτική της κατασκευής του περιβάλλοντος πειραματισμού στον πραγματικό κόσμο περιορίζεται όχι μόνο από θέματα πρόσβασης αλλά και από τη δυνατότητα να ρυθμίσει κάποιος όλες τις αρχικές παραμέτρους του φαινομένου ή και τον χρόνο έναρξης του πειράματος. Μπορούμε να σκεφτούμε ως παραδείγματα φαινομένων με τέτοιους περιορισμούς τις καταστάσεις εκτάκτου ανάγκης (σεισμός, φωτιά κ.ά.), το χρηματιστήριο, τον κύκλο του νερού κ.ά. Από την άλλη μεριά υπάρχουν εξειδικευμένα εργαλεία προσομοίωσης που μπορούν να παρέχουν πολύ μεγαλύτερη ακρίβεια και λεπτομέρεια στα αποτελέσματα της προσομοίωσης. Αυτό είναι αλήθεια και σε καμία περίπτωση δεν θα μπορούσε να ισχυριστεί κάποιος ότι οι εικονικοί κόσμοι μπορούν να φτάσουν σε αντίστοιχα επίπεδα πιστότητας και ποιότητας αποτελεσμάτων. Όμως, οι εικονικοί κόσμοι είναι περισσότερο διαδραστικοί και επιτρέπουν την εξερεύνηση και παρατήρηση του φαινομένου σε πραγματικό χρόνο, καθώς αυτό εξελίσσεται, καθώς και τη διάδραση με τα στοιχεία που το καθορίζουν. Είναι συμμετοχικοί και επιτρέπουν την ενσωμάτωση του ίδιου του χρήστη μέσα στο περιβάλλον που εξελίσσεται το φαινόμενο και όχι έξω από αυτό ως απλός παρατηρητής, καθώς και άλλων χρηστών που μπορούν να συνυπάρχουν στο ίδιο περιβάλλον και να επηρεάζουν την εξέλιξη της προσομοίωσης. Είναι ευκολότεροι στην εκμάθηση και πολύ περισσότερο διαισθητικοί στη χρήση σε σχέση με εξειδικευμένες εφαρμογές. Τέλος, δίνουν τη δυνατότητα στους ίδιους τους χρήστες να συν-δημιουργήσουν τα περιεχόμενα του περιβάλλοντος, να κατασκευάσουν και να πειραματιστούν με τις δικές τους προσομοιώσεις.

3.3 Παιχνίδια ρόλων

Μια τρίτη κατηγορία δραστηριοτήτων η οποία φαίνεται να υποστηρίζεται με επιτυχία από τους εικονικούς κόσμους είναι τα παιχνίδια ρόλων. Οι χρήστες μπορούν να καθορίσουν την εμφάνισή τους, να φορέσουν ρούχα της επιλογής τους, να επιλέξουν το εμφανιζόμενο όνομά τους και γενικότερα να διαμορφώσουν τη διαδικτυακή ταυτότητά τους σύμφωνα με τις προτιμήσεις τους. Επιπλέον, μπορούν να συνυπάρχουν σε κόσμους οι οποίοι είναι σχεδιασμένοι στα πρότυπα κάποιου υποθετικού, φανταστικού, παρελθοντικού ή ακόμα και πραγματικού περιβάλλοντος. Τα δύο παραπάνω στοιχεία καθιστούν δυνατή τη συμμετοχή πολλαπλών χρηστών σε προσχεδιασμένες ή αυθόρμητες ιστορίες, στις οποίες ο καθένας αναλαμβάνει τον δικό του ρόλο και, ανάλογα με το συμφωνημένο πλαίσιο εξέλιξης, μπορεί να έχει δημιουργική ελευθερία σε μικρότερο ή μεγαλύτερο βαθμό. Τα παιχνίδια ρόλων, βεβαίως, προϋπάρχουν των εικονικών κόσμων ως ομαδικές δραστηριότητες που παίζονται στον φυσικό κόσμο σύμφωνα με ένα σύνολο κανόνων που καθορίζουν την εξέλιξη του παιχνιδιού. Άλλωστε, τα πρώτα ψηφιακά παιχνίδια ρόλων ήταν σαφώς επηρεασμένα από αυτά, όπως θα δούμε στη συνέχεια του κεφαλαίου. Όμως, στην περίπτωση υιοθέτησης ρόλων στους εικονικούς κόσμους μπορεί κάποιος να εντοπίσει αρκετές διαφορές, κυρίως πλεονεκτήματα. Πρώτον, στους εικονικούς κόσμους μπορεί να υπάρξει

πολύ πιο ρεαλιστική απεικόνιση της εμφάνισης των χαρακτήρων, του πλαισίου δράσης και της αποτύπωσης των ενεργειών τους γενικότερα. Άλλωστε, η εικονική πραγματικότητα θεωρείται ένα ακόμα μέσο αφήγησης μαζί με τα βιβλία, τις ταινίες κ.λπ. (Aylett & Louchart, 2003). Τα παραπάνω ενισχύουν την αίσθηση συμμετοχής στο φανταστικό περιβάλλον, περιορίζοντας έτσι το μειονέκτημα της μη φυσικής συν-παρουσίας των παικτών σε σχέση με τα παραδοσιακά παιχνίδια ρόλων. Δεύτερον, οι κανόνες του παιχνιδιού μπορούν να είναι ενσωματωμένοι στο περιβάλλον, με αποτέλεσμα να μην χρειάζεται να μελετήσουν και να θυμούνται όλοι οι παίκτες έναν μεγάλο αριθμό κανόνων για να μπορούν να παίξουν το παιχνίδι. Τρίτον, δεν είναι υποχρεωτικό να υπάρχουν κανόνες, μπορούν οι ίδιοι οι παίκτες να καθορίζουν και να συν-διαμορφώνουν τους κανόνες όσο το παιχνίδι εξελίσσεται. Και τέλος, η συμμετοχή δεν είναι υποχρεωτικό να είναι σύγχρονη, δηλαδή να παίζουν όλοι οι παίκτες στο ίδιο χρονικό διάστημα. Μπορούν να υποστηριχθούν και μοντέλα ασύγχρονης συμμετοχής, όπως άλλωστε συμβαίνει σήμερα στους δημοφιλείς εικονικούς κόσμους παιχνιδιών, όπου ο κάθε παίκτης επιλέγει ο ίδιος τον χρόνο που θα δεσμεύσει για το παιχνίδι.

3.4 Συνεργατική σχεδίαση

Οι εικονικοί κόσμοι μπορούν επιπλέον να υποστηρίξουν με αρκετή επιτυχία δραστηριότητες συνεργατικής σχεδίασης περιεχομένου. Πέρα από τις προσφερόμενες δυνατότητες εξερεύνησης και περιήγησης του κατασκευασμένου περιεχομένου, σε αρκετούς εικονικούς κόσμους παρέχονται και εργαλεία κατασκευής και διαμόρφωσης νέου περιεχομένου εντός του περιβάλλοντος. Ενώσω λοιπόν οι χρήστες περιηγούνται στον κόσμο, έχουν τη δυνατότητα να σχολιάσουν τα αντικείμενα που βλέπουν, να τα επεξεργαστούν, να τους προσθέσουν οπτικά σημάδια ή επεξηγήσεις, να σχεδιάσουν δικά τους αντικείμενα, να τα μοιραστούν με άλλους χρήστες, να τα διαμορφώσουν από κοινού, να τροποποιήσουν την εμφάνισή τους, να καθορίσουν τα δικαιώματα πρόσβασης σε αυτά κ.ά. Κατά συνέπεια, ομάδες χρηστών μπορούν να συνεργάζονται στην κατασκευή, βελτίωση και επέκταση κάποιου τεχνουργήματος, που μπορεί να είναι αντικείμενο, κτίριο ή ακόμα και ολόκληρη περιοχή του κόσμου. Τα σχεδιαστικά εργαλεία εντός των εικονικών κόσμων δεν έχουν βέβαια την απαραίτητη ακρίβεια ούτε το σύνολο των λειτουργιών που παρέχουν τα εξειδικευμένα πακέτα τρισδιάστατης μοντελοποίησης. Επιπλέον, λόγω των περιορισμών σε πόρους και της ανάγκης προβολής σε πραγματικό χρόνο (real-time rendering), δεν είναι δυνατόν να κατασκευάσει κάποιος εντός ενός εικονικού κόσμου ένα αντικείμενο πολύ μεγάλου βαθμού λεπτομέρειας. Όμως, η συνεργατική σχεδίαση μέσω εικονικών κόσμων μπορεί να έχει άλλα πλεονεκτήματα. Καταρχήν, τα προσφερόμενα εργαλεία είναι αρκετά πιο εύκολα στην εκμάθηση και χρήση από μη ειδικούς, επομένως η ομάδα σχεδίασης μπορεί να πλαισιωθεί και από δυνητικούς χρήστες/πελάτες. Επιπλέον, η δυνατότητα εύκολης προβολής και περιήγησης στο υπό σχεδίαση μοντέλο επιτρέπει τη γρήγορη ανατροφοδότηση μέσω σχολίων ή και εναλλακτικών τοποθετήσεων/διαμορφώσεων, τόσο σύγχρονα (π.χ. στο πλαίσιο ομαδικών συναντήσεων) όσο και ασύγχρονα. Τέλος, στον εικονικό κόσμο μπορεί να τοποθετηθεί το σχεδιαζόμενο μοντέλο σε ένα πιο ρεαλιστικό περιβάλλον και να προβληθεί η χρήση του μέσω διαδικασιών προσομοίωσης. Για παράδειγμα, ο πελάτης δεν περιορίζεται στην απλή επισκόπηση ενός σχεδιαζόμενου μοντέλου αυτοκινήτου, αλλά μπορεί να αποκτήσει και μια πρώτη αίσθηση του πώς θα ήταν αν καθόταν μέσα σε αυτό και το οδηγούσε στην πόλη.

3.5 Ομαδικές συζητήσεις

Τέλος, οι ομαδικές συζητήσεις και παρουσιάσεις είναι μια ακόμα ομάδα δραστηριοτήτων που φαίνεται να μπορεί να επωφεληθεί από τις δυνατότητες των εικονικών κόσμων. Σε έναν κατάλληλα διαμορφωμένο εικονικό χώρο μπορούν να συνυπάρχουν πολλαπλοί χρήστες, να επικοινωνούν μέσω κειμένου ή φωνής, να βλέπουν από κοινού τα περιεχόμενα κειμένων ή παρουσιάσεων, να κρατούν και να αναρτούν τις σημειώσεις τους, να επεξεργάζονται κείμενα από κοινού, να σηκώνουν το χέρι τους για να ζητήσουν τον λόγο, να χειροκροτούν, να συμμετέχουν σε φανερές ή μυστικές ψηφοφορίες και πολλά άλλα. Πρακτικά, σχεδόν οτιδήποτε θα μπορούσαν να κάνουν σε έναν αντίστοιχο φυσικό χώρο (π.χ. αίθουσα συνεργασίας, αμφιθέατρο, συνεδριακή αίθουσα), χωρίς όμως την υποχρέωση να είναι φυσικά παρόντες στον χώρο αυτό. Και στην περίπτωση αυτή θα μπορούσε να ισχυριστεί κάποιος ότι υπάρχει εξειδικευμένο λογισμικό που υποστηρίζει τέτοιου τύπου δραστηριότητες. Η αλήθεια είναι ότι ο περισσότερος κόσμος σήμερα χρησιμοποιεί λογισμικό τηλεδιασκέψεων μέσω φωνής ή/και βίντεο για να συνεργαστεί, και πράγματι, η λύση αυτή είναι σίγουρα προτιμότερη και αποδοτικότερη αν μιλάμε

για ένα πλαίσιο συνεργασίας στο οποίο η έμφαση είναι στην επικοινωνία και περιορίζεται σε μικρό αριθμό συμμετεχόντων. Με τη χρήση των εικονικών κόσμων όμως μπορούμε να έχουμε άλλου είδους πλεονεκτήματα:

- μπορεί να υποστηριχθεί ένας μεγάλος αριθμός ταυτόχρονα συνδεδεμένων χρηστών,
- οι χρήστες συνυπάρχουν σε έναν ενιαίο χώρο και μπορούν να αντιλαμβάνονται ο ένας τον άλλο μέσω των ενσαρκώσεών τους,
- οι χρήστες αναφέρονται σε αντικείμενα ή δεδομένα εντός κόσμου που παρατηρούν από κοινού,
- μπορεί να οργανωθεί η χωροθέτησή τους σύμφωνα με τις συμβάσεις που ισχύουν και σε αντίστοιχους φυσικούς χώρους (π.χ. στρογγυλό τραπέζι συνεργασίας, έδρα),
- μπορούν να ανατεθούν ρόλοι στους χρήστες, οι οποίοι να αντικατοπτρίζονται ακόμα και στην εμφάνισή τους,
- μπορεί να οργανωθεί το μοντέλο επικοινωνίας και συνεργασίας μέσω προσφερόμενων ενεργειών και δυνατοτήτων (π.χ. ψηφοφορίες, ανάταση χειρός, αποδοχή/απόρριψη, σχολιασμός σε μορφή κειμένου, καταγραφή πρακτικών).

Τα παραπάνω πλεονεκτήματα βασίζονται στο γεγονός ότι οι εικονικοί κόσμοι παρέχουν τη δυνατότητα ελεύθερης διαμόρφωσης του περιεχομένου και των υποστηριζόμενων δραστηριοτήτων και προγραμματισμού της συμπεριφοράς των διαδραστικών αντικειμένων εντός του περιβάλλοντος. Αυτό είναι κάτι που δεν συνηθίζεται στα εμπορικά προγράμματα απομακρυσμένης συνεργασίας. Στην εικόνα 1.2 απεικονίζεται σεμινάριο εξ αποστάσεως που πραγματοποιήθηκε στον εικονικό κόσμο του OpenSimulator.



Εικόνα 1.2 Σεμινάριο εξ αποστάσεως στο OpenSimulator.

3.6 Περιοχές εφαρμογής

Η καλή υποστήριξη των παραπάνω δραστηριοτήτων σε κατάλληλα κατασκευασμένα περιβάλλοντα μπορεί να οδηγήσει στη δημιουργία καινοτόμων συστημάτων και στην παροχή επιτυχημένων λύσεων σε ένα μεγάλο εύρος περιοχών εφαρμογής. Για κάποιες από τις δυνητικές περιοχές εφαρμογής έχουν ήδη κατασκευαστεί με επιτυχία συστήματα βασισμένα σε εικονικούς κόσμους, ενώ σε ορισμένες άλλες υπάρχουν σημαντικές ενδείξεις επιτυχημένης σχεδίασης σύμφωνα με σχετικές ερευνητικές μελέτες και αποτελέσματα εργαστηριακών πειραμάτων. Στη συνέχεια θα αναφερθούμε ενδεικτικά σε ορισμένες από αυτές και σε τρόπους με τους οποίους θα μπορούσαν να αξιοποιηθούν τις δυνατότητες των εικονικών κόσμων. Τα στοιχεία αυτά παρουσιάζονται συνοπτικά στον πίνακα 1.2.

Περιοχές Εφαρμογής	Δυνατότητες
ψυχαγωγία	ποιοτική αναπαράσταση, συνεργασία και ανταγωνισμός με άλλους χρήστες, επέκταση κόσμου
εκπαίδευση	διδασκαλία εξ αποστάσεως, παρατήρηση και περιήγηση σε εκπαιδευτικό περιεχόμενο, πειραματισμός, συνεργατική μάθηση
εξάσκηση	ρεαλιστικά περιβάλλοντα προσομοίωσης με συμμετοχή πολλαπλών χρηστών
πολιτισμός	εικονικά μουσεία, προσωποποιημένη ξενάγηση, ανασύσταση χώρων, αναβίωση εποχών
αρχιτεκτονική	διαμόρφωση προκαταρκτικών ιδεών, παρουσίαση σε πελάτες, συν-σχεδίαση, περιήγηση και αξιολόγηση
σχεδίαση προϊόντων	πρωτοτυποποίηση, παρουσίαση σε ρεαλιστικό περιβάλλον χρήσης, ανατροφοδότηση από χρήστες
τέχνες	τρισεπίσταντες, κινούμενες, διαδραστικές αναπαραστάσεις
θεραπεία	αντιμετώπιση φοβιών, αποκατάσταση κινητικών προβλημάτων, ανακούφιση από πόνο

Πίνακας 1.2 Περιοχές εφαρμογής των εικονικών κόσμων και δυνατότητες αξιοποίησής τους.

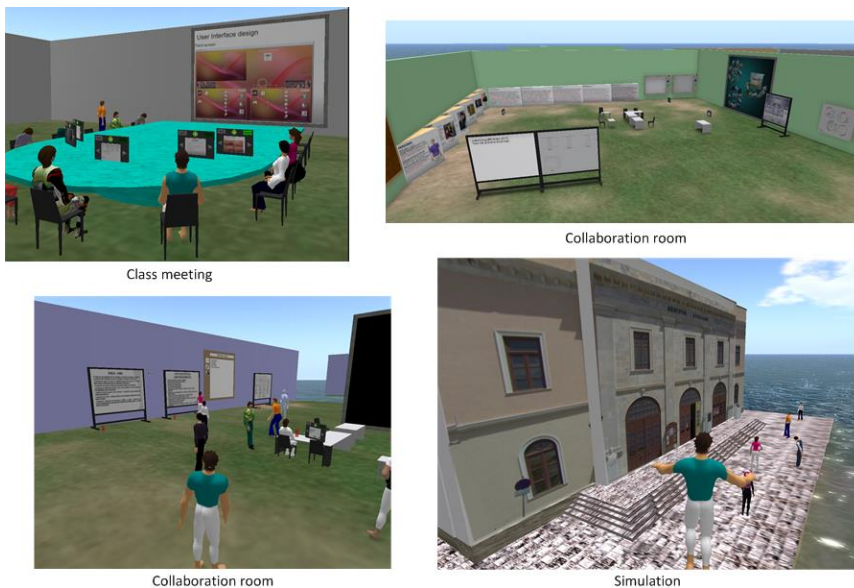
3.6.1 Ψυχαγωγία

Ένας τομέας ο οποίος διαισθητικά φαντάζει ως ο καταλληλότερος για την αξιοποίηση των ιδιαίτερων στοιχείων που προσφέρουν οι εικονικοί κόσμοι είναι η ψυχαγωγία. Πράγματι, οι εικονικοί κόσμοι έχουν παραδοσιακά μια στενή σχέση με τους χώρους των ψηφιακών παιχνιδιών και της κοινωνικής δικτύωσης, και κατά τη διάρκεια της εξέλιξής τους έχουν δανειστεί αρκετά στοιχεία από τους χώρους αυτούς. Δεν είναι λοιπόν τυχαίο το ότι οι περισσότερες εφαρμογές των εικονικών κόσμων, είτε ως αυτόνομα περιβάλλοντα, είτε ως σχεδιασμένοι χώροι εντός κάποιου κόσμου γενικού χαρακτήρα, όπως το Second Life, έχουν καθαρά ψυχαγωγικούς στόχους. Τα τρισεπίσταντα γραφικά, οι ήχοι και τα ειδικά εφέ μπορούν να «ζωντανέψουν» πραγματικά, παρελθοντικά ή υποθετικά περιβάλλοντα εξάπτοντας τη φαντασία, η ελεύθερη πλοήγηση και διάδραση των χρηστών με τον κόσμο ευνοούν την εξερεύνηση και την ανακάλυψη, η συν-παρουσία πολλαπλών χρηστών δημιουργεί συνθήκες συνεργασίας ή/και ανταγωνισμού, ενώ και η ίδια η συμβολή των χρηστών στη διαμόρφωση και εξέλιξη του περιεχομένου απελευθερώνει την προσωπική έκφραση και τη δημιουργικότητα (Eisenbeiss κ.ά., 2012). Τα παραπάνω αποτελέσματα μπορούν να οδηγήσουν σε περισσότερο πλούσιες και διασκεδαστικές εμπειρίες και έχουν αξιοποιηθεί με πολλούς τρόπους σε σχετικές ψυχαγωγικές εφαρμογές.

3.6.2 Εκπαίδευση και εξάσκηση

Ένας δεύτερος τομέας που έχει αποκτήσει μεγάλο ερευνητικό ενδιαφέρον την τελευταία δεκαετία είναι η αξιοποίηση των εικονικών κόσμων στον χώρο της εκπαίδευσης (Duncan κ.ά., 2012). Έχει ήδη αναφερθεί η δυνατότητα των συνδεδεμένων χρηστών να συνυπάρχουν σε κοινό χώρο και να επικοινωνούν μεταξύ τους μέσω φωνής ή/και κειμένου. Η δυνατότητα αυτή μπορεί να αξιοποιηθεί σε κατάλληλα σχεδιασμένους χώρους για την υποστήριξη ψηφιακών τάξεων, δηλαδή εικονικών χώρων σύγχρονης διδασκαλίας εξ αποστάσεως στους οποίους οι διδάσκοντες και οι διδασκόμενοι συμμετέχουν μέσω των ενσαρκώσεών τους. Σε μαθησιακά αντικείμενα στα οποία η μορφή, η αρχιτεκτονική ή ο χώρος αποτελούν κεντρικό στοιχείο (π.χ. μοριακές δομές, γεωγραφία, ιστορικές μάχες, ανθρώπινο σώμα κ.ά.), οι μαθητές θα μπορούσαν μέσω εικονικών κόσμων να παρατηρήσουν ή/και να περιηγηθούν στο περιεχόμενο μάθησης. Αλλά και σε χρονικά μεταβαλλόμενα ή πιο περίπλοκα αντικείμενα (π.χ. η λειτουργία του πεπτικού συστήματος, βολές, κώδικας οδικής κυκλοφορίας), οι εικονικοί κόσμοι θα μπορούσαν να δημιουργήσουν κατάλληλες συνθήκες ελεύθερου ή ελεγχόμενου πειραματισμού, ώστε οι μαθητές να μπορέσουν να δοκιμάσουν εναλλακτικές διατάξεις, να παρατηρήσουν τα αποτελέσματα και να αναστοχαστούν πάνω σε αυτά. Ένα ακόμα ενδιαφέρον χαρακτηριστικό των περιβαλλόντων αυτών είναι και η σύγχρονη παρουσία των χρηστών σε ένα κοινό περιβάλλον, η οποία παρέχει τη δυνατότητα επικοινωνίας, συνεργασίας, από κοινού αντίληψης των προβλημάτων, συντονισμό των ενεργειών που απαιτούνται για την επίλυσή τους κ.ά. Όλες δηλαδή τις προϋποθέσεις για την προετοιμασία σεναρίων συνεργατικής μάθησης και μάλιστα εμπλουτισμένων με ψυχαγωγικά στοιχεία. Μέχρι σήμερα οι εικονικοί κόσμοι φαίνεται να αποτελούν ένα ελπιδοφόρο μέσο για τη σχεδίαση και υλοποίηση εκπαιδευτικών

σεναρίων βασισμένων σε σύγχρονες προσεγγίσεις όπως ο εποικοδομητισμός (constructivism), η προβληματοκεντρική μάθηση (problem-based learning) και η μάθηση μέσω δημιουργίας (constructionism / learning by making). Στην εικόνα 1.3 φαίνεται ένα παράδειγμα χρήσης του OpenSimulator ως περιβάλλοντος συνεργατικής μάθησης στη διδασκαλία της σχεδίασης διαδραστικών συστημάτων (Koutsabasis & Vosinakis, 2012).



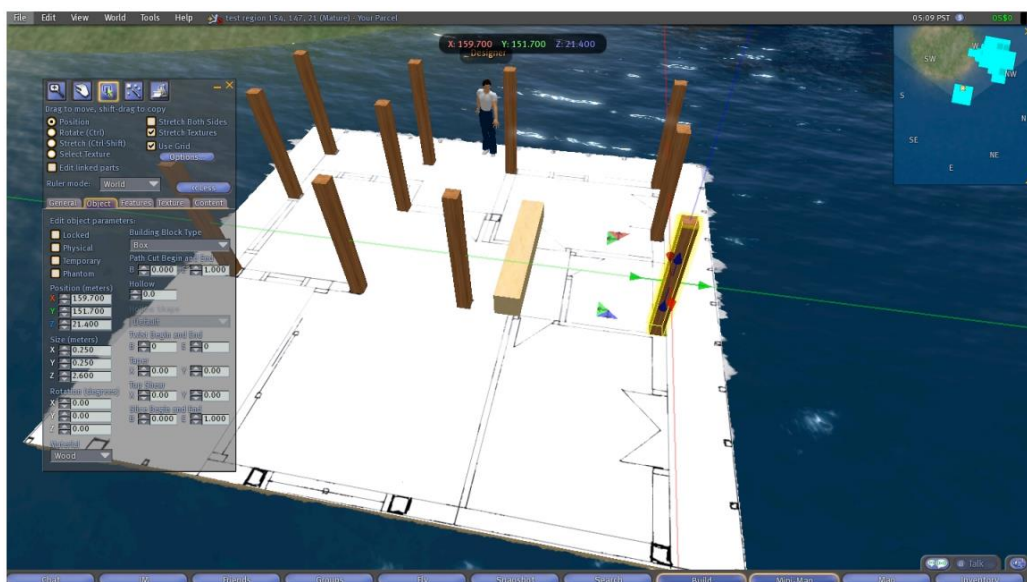
Εικόνα 1.3 Χρήση του OpenSimulator στη διδασκαλία της σχεδίασης διαδραστικών συστημάτων στο Παν/μιο Αιγαίου.

Η σημαντική αξιοποίηση των εικονικών κόσμων στον χώρο της εκπαίδευσης οδήγησε στην αντίστοιχη χρήση των περιβαλλόντων αυτών και σε έναν άλλο, παραπλήσιο χώρο, αυτόν της επαγγελματικής κατάρτισης και εξάσκησης (training). Επαγγελματίες που επιθυμούν να διευρύνουν τις δεξιότητές τους ή/και να εξασκηθούν στην πρακτική εφαρμογή των γνώσεων που έχουν αποκτήσει μπορούν να εκμεταλλευτούν τις δυνατότητες των εικονικών κόσμων να αναπαριστούν ρεαλιστικά περιβάλλοντα προσομοίωσης που επιτρέπουν την ταυτόχρονη συμμετοχή πολλαπλών χρηστών. Οι συμμετέχοντες μπορούν να αναλάβουν ρόλους στα πλαίσια προσομοίωσης σύνθετων διαδικασιών (π.χ. διαδικασία εισαγωγής ασθενούς σε μονάδα εντατικής θεραπείας), να επεκτείνουν τις δεξιότητές τους μέσα από τη χρήση εξειδικευμένων συσκευών (π.χ. προσομοίωση πτήσης, έλεγχος και συντήρηση σύνθετου εξοπλισμού) ή και να κατανοήσουν σε μεγαλύτερο βάθος τη λειτουργία περίπλοκων συστημάτων (π.χ. χρηματιστήριο) μέσα από την ενεργό συμμετοχή τους και την παρατήρηση των αποτελεσμάτων. Χωρίς σε καμία περίπτωση να μπορούν να αντικαταστήσουν πλήρως τις πραγματικές συνθήκες, οι εικονικοί κόσμοι μπορούν να παρουσιάσουν ασφαλείς και προσβάσιμους κόσμους υψηλής πιστότητας στους οποίους συμμετέχουν και αξιολογούνται πολλαπλοί χρήστες και των οποίων η δομή και ο βαθμός δυσκολίας είναι απολύτως ελεγχόμενα από τους εκπαιδευτές. Τα παραπάνω χαρακτηριστικά τους καθιστούν εξαιρετικά χρήσιμα εργαλεία για την υποστήριξη σύγχρονων σεναρίων εκπαίδευσης και κατάρτισης στελεχών.

3.6.3 Πολιτισμός

Ένας ακόμα χώρος στον οποίο οι εικονικοί κόσμοι έχουν σημαντική συνεισφορά αλλά και πολλές δυνατότητες περαιτέρω αξιοποίησης είναι αυτός του πολιτισμού. Τεχνουργήματα, κτίρια, ακόμα και πόλεις ολόκληρες μπορούν να αναπαρασταθούν «εικονικά» και να δοθεί η ευκαιρία στους επισκέπτες να τα παρατηρήσουν και να τα εξερευνήσουν (Sequeira & Morgado, 2013). Επειδή στους εικονικούς κόσμους δεν υπάρχουν περιορισμοί, η αναπαράσταση του πολιτισμικού περιεχομένου μπορεί να αφορά την τρέχουσα μορφή του ή ακόμα και μια επιχειρούμενη ανασύσταση της μορφής που υποθέτουμε ότι είχε στο παρελθόν. Πέρα από την απλή παρουσίαση, εικονικά αποθετήρια αρχαιολογικών ευρημάτων και πολιτισμικών προϊόντων μπορούν να κατασκευαστούν σε εικονικούς κόσμους με τη μορφή ψηφιακών μουσείων που παρέχουν στους επισκέπτες συγκεντρωτικές και θεματικές προβολές του περιεχομένου. Ήδη έχουν κατασκευαστεί πολλά εικονικά μουσεία

με ψηφαγωγικό και παιδαγωγικό χαρακτήρα, τα οποία σε κάποιες περιπτώσεις δίνουν έμφαση στην πιστή αναπαραγωγή της παραδοσιακής εμπειρίας επίσκεψης σε μουσείο (ενίοτε αναπαριστώντας με μεγάλη ακρίβεια τους χώρους και τις συλλογές υπαρκτών μουσείων) και σε κάποιες άλλες επεκτείνουν την εμπειρία αυτή με επιπλέον στοιχεία, όπως η προσωποποιημένη ξενάγηση, η διάδραση με τα εκθέματα, η διασύνδεση με εξωτερικές πληροφορίες και η κοινωνική δικτύωση. Πέρα όμως από τη χρήση τους ως ψηφιακά αποθετήρια τρισδιάστατων αναπαραστάσεων, οι εικονικοί κόσμοι μπορούν να προσφέρουν και πιο πλούσιες, βιωματικές εμπειρίες στον χώρο του πολιτισμού. Χώροι και εποχές του παρελθόντος μπορούν να αναπαρασταθούν οπτικά και να «ζωντανέψουν» με τη συμμετοχή εικονικών χαρακτήρων. Οι επισκέπτες θα μπορούσαν να περιηγηθούν σε τέτοια περιβάλλοντα, ενδεχομένως και με τη δυνατότητα ενεργού εμπλοκής μέσω ανάληψης συγκεκριμένων ρόλων και να αποκτήσουν μια πολύ πιο πλούσια και βιωματική εμπειρία σχετικά με την κουλτούρα και τις συνθήκες ζωής ενός αρχαίου πολιτισμού καθώς και με το πλαίσιο χρήσης σημαντικών αρχαίων μνημείων και τεχνουργημάτων.



Εικόνα 1.4 Σχεδίαση κτιρίου στο περιβάλλον του OpenSimulator.

3.6.4 Αρχιτεκτονική και σχεδίαση προϊόντων

Οι δυνατότητες εύκολης κατασκευής και αναδιαμόρφωσης τρισδιάστατου περιεχομένου, ρεαλιστικής περιήγησης μέσα σε αυτό και άμεσης ανατροφοδότησης κάνουν τους εικονικούς κόσμους ελκυστικούς για χρήση και στον χώρο της αρχιτεκτονικής κ.ά., (2012). Παρά το ότι δεν προσφέρουν τη λεπτομέρεια και ακρίβεια των επαγγελματικών εργαλείων αρχιτεκτονικής σχεδίασης, οι εικονικοί κόσμοι επιτρέπουν την εύκολη αποτύπωση ιδεών, διαμοίραση και προκαταρκτική αξιολόγηση. Πράγματι, με τα εργαλεία κατασκευής που παρέχονται εντός εικονικών κόσμων γενικής χρήσης, όπως το Second Life, μπορεί ένας επαγγελματίας να διαμορφώσει γρήγορα μια πρόχειρη ιδέα, να την παρουσιάσει άμεσα σε συνεργάτες ή/και δυνητικούς πελάτες και να δεχτεί γρήγορη ανατροφοδότηση (π.χ. με τη μορφή οπτικών επισημειώσεων πάνω στο τρισδιάστατο περιβάλλον). Αυτά τα χαρακτηριστικά επιτρέπουν τη συν-σχεδίαση, ταχεία πρωτοτυποποίηση και προκαταρκτική αξιολόγηση χώρων, διαδικασίες ιδιαίτερα επιθυμητές, κυρίως στην περίπτωση καινοτόμων σχεδίων όπου ο έγκαιρος εντοπισμός λαθών και προβλημάτων είναι βασικό ζητούμενο. Ένα υποψήφιο αρχιτεκτονικό σχέδιο μπορεί να εμπλουτιστεί με περισσότερες λεπτομέρειες και να παρουσιαστεί αρκετά ρεαλιστικά σε κάποιον εικονικό κόσμο, τοποθετημένο σε κατάλληλα σχεδιασμένο περιβάλλον, επιτρέποντας σε έναν ή περισσότερους επισκέπτες να βιώσουν την εμπειρία περιήγησης και χρήσης του χώρου υπό σχεδίαση. Αυτή η δυνατότητα αναμένεται να διευκολύνει, να επιταχύνει και να εμπλουτίσει την ανατροφοδότηση των χρηστών σχετικά με τη συνολική εικόνα και λειτουργικότητα του χώρου πριν την τελική κατασκευή ή αναδιαμόρφωσή του, σε σχέση με μια απλή παρουσίαση εικόνων υψηλής ποιότητας. Στην εικόνα 1.4 φαίνεται ένα παράδειγμα σχεδίασης κτιρίου στον εικονικό κόσμο OpenSimulator.

Η σχεδίαση προϊόντων και υπηρεσιών είναι ένας ακόμη χώρος που μπορεί να επωφεληθεί από τους εικονικούς κόσμους αξιοποιώντας τους ως εργαλείο συνεργατικής σχεδίασης, ταχείας πρωτοτυποποίησης και ανατροφοδότησης. Προκαταρκτικά σχέδια προϊόντων με ή χωρίς διαδραστική συμπεριφορά μπορούν να παρουσιαστούν στους δυνητικούς αγοραστές, οι οποίοι με τη σειρά τους θα έχουν σε κάποιες περιπτώσεις τη δυνατότητα να τα χρησιμοποιήσουν μέσω της ενσάρκωσής τους. Ρούχα, έπιπλα, οχήματα, ακόμα και αντικείμενα καθημερινής χρήσης μπορούν να εισαχθούν σε κατάλληλα διαμορφωμένα σενάρια χρήσης και να γίνουν διαθέσιμα για κριτική και ανατροφοδότηση σε πολύ μεγάλο αριθμό ενδιαφερόμενων επισκεπτών. Και στην περίπτωση αυτήν, προφανώς, το μεγάλο πλεονέκτημα της χρήσης εικονικών κόσμων είναι η βιωματική εμπειρία μέσω ενσάρκωσης με τα υπό σχεδίαση προϊόντα σε ένα ρεαλιστικό περιβάλλον πλαίσιο (context) στο οποίο τα αντικείμενα παρουσιάζονται στις πραγματικές τους διαστάσεις. Αυτού του είδους η βιωματική παρουσίαση σε μεγάλο αριθμό ενδιαφερομένων δεν είναι υποχρεωτικό να περιοριστεί σε μεμονωμένα προϊόντα· θα μπορούσε να αναφέρεται σε μια υπό σχεδίαση υπηρεσία που να εμπλέκει χώρους, αντικείμενα και ρόλους για την ικανοποίηση ενός συγκεκριμένου στόχου, π.χ. σχεδίαση ενός νέου συστήματος εξυπηρέτησης πελατών σε υποκατάστημα τράπεζας. Και στην περίπτωση αυτή, η γρήγορη πρωτοτυποποίηση και ανατροφοδότηση από πολλαπλούς χρήστες αναμένεται να έχει σημαντικά οφέλη. Αλλά ακόμα και σε περιπτώσεις διαμόρφωσης ιδεών και προκαταρκτικής σχεδίασης, οι εικονικοί κόσμοι μπορούν να διευκολύνουν τη σχεδιαστική ομάδα στη γρήγορη παρουσίαση, δοκιμή και αξιολόγηση νέων ιδεών μέσω των εργαλείων κατασκευής, εισαγωγής και επεξεργασίας τρισδιάστατων αντικειμένων που διαθέτουν.

3.6.5 Τέχνες

Η δημιουργική ελευθερία που παρέχουν οι εικονικοί κόσμοι μπορεί να αξιοποιηθεί ποικιλοτρόπως και στον χώρο των τεχνών. Ιδεατοί, αφηρημένοι, σουρεαλιστικοί κόσμοι μπορούν να κατασκευαστούν, να αποκτήσουν κίνηση και συμπεριφορά και εντέλει να «ζωντανέψουν» σε ένα ψηφιακό πολυχρηστικό περιβάλλον (Mura, 2012). Χωρίς κανέναν χωρικό, υλικό ή φυσικό περιορισμό, οι δημιουργοί μπορούν να διαμορφώσουν τοπία, κτίρια, αντικείμενα και ζωντανές μορφές της φαντασίας τους. Μπορούν να προσθέσουν στις δημιουργίες τους κίνηση, αφήγηση ή και διαδραστική συμπεριφορά, μπορούν να τις παρουσιάσουν διαδικτυακά αφήνοντας τους επισκέπτες να τις παρατηρήσουν, να περιηγηθούν μέσα σε αυτές, να διαδράσουν, ακόμα και να συνδιαμορφώσουν το αποτέλεσμα. Όλα τα παραπάνω επεκτείνουν πολύ περισσότερο τις δυνατότητες έκφρασης σε σχέση με τις παραδοσιακές τέχνες όπως η ζωγραφική, η γλυπτική ή και πιο σύγχρονες όπως ο κινηματογράφος, και δεν είναι τυχαίο το ότι αρκετοί καλλιτέχνες έχουν αρχίσει να βλέπουν τους εικονικούς κόσμους ως ένα πολλά υποσχόμενο πεδίο δημιουργίας. Πέρα όμως από την αξιοποίησή τους ως μέσο καλλιτεχνικής έκφρασης, οι εικονικοί κόσμοι μπορούν να λειτουργήσουν υποστηρικτικά και προς άλλες μορφές τέχνης και δημιουργίας. Χαρακτηριστικά παραδείγματα είναι η αποτύπωση προκαταρκτικών ιδεών και η γρήγορη πρωτοτυποποίηση σκηνικών για τον κινηματογράφο ή και παιχνίδια υπολογιστών καθώς και η παραγωγή ψηφιακών ταινιών μέσω σύλληψης σε πραγματικό χρόνο της δραστηριότητας στους εικονικούς κόσμους.

3.6.6 Θεραπεία

Τέλος, οι εικονικοί κόσμοι μπορούν να αξιοποιηθούν και σε λιγότερο προφανείς στόχους, όπως η θεραπεία. Σε έναν χώρο στον οποίο τα συστήματα εικονικής πραγματικότητας έχουν ήδη εμφανίσει αξιοσημείωτα αποτελέσματα, οι εικονικοί κόσμοι θα μπορούσαν να διευρύνουν τη συνεισφορά αυτή σε ένα πολύ μεγαλύτερο κοινό (Gorini κ.ά., 2008). Λύσεις όπως η ανακούφιση από τον πόνο, η αποκατάσταση κινητικών προβλημάτων, η θεραπεία ψυχολογικών ασθενειών (π.χ. αγοραφοβία, υψοφοβία, αραχοφοβία) μπορούν να προσφερθούν μερικώς από τους εικονικούς κόσμους μέσα από τη δυνατότητά τους να δημιουργούν στους χρήστες την αίσθηση της «παρουσίας» σε ένα εναλλακτικό περιβάλλον. Με την αξιοποίηση χαμηλού κόστους υλικού για την ενίσχυση της ποιότητας απεικόνισης και αλληλεπίδρασης με το περιβάλλον (όπως, για παράδειγμα, το κράνος εικονικής πραγματικότητας Oculus Rift στο οποίο θα αναφερθούμε αργότερα), θα μπορούσαν κατάλληλα σχεδιασμένοι εικονικοί κόσμοι να μεταφέρουν τους χρήστες τους σε περιβάλλοντα που τους κάνουν να «ξεχάσουν» προσωρινά την πραγματικότητα, που τους παρέχουν κίνητρα να εκτελέσουν κάποιες σωματικές ασκήσεις ή ακόμα και που τους φέρνουν σταδιακά αντιμέτωπους με τις χρόνιες φοβίες τους.

4 Το όραμα της Εικονικής Πραγματικότητας

Οι εικονικοί κόσμοι στη σημερινή τους μορφή είναι αποτέλεσμα της «σύγκλισης» δύο αρχικά ανεξάρτητων μεταξύ τους ρευμάτων. Από τη μια μεριά η επιστημονική περιοχή της «Εικονικής Πραγματικότητας», που υπάρχει και εξελίσσεται εδώ και πολλά χρόνια ως κλάδος της επιστήμης της πληροφορικής, και από την άλλη οι «κοινωνικοί κόσμοι» και τα παιχνίδια ρόλων, που βασίστηκαν στην ανάπτυξη και διάδοση του διαδικτύου και υποστηρίχθηκαν από μια ευρεία βάση χρηστών. Υπό μία έννοια λοιπόν οι εικονικοί κόσμοι συνδυάζουν τα αποτελέσματα πολύχρονων επιστημονικών μελετών και σχετικών τεχνολογικών εξελίξεων με την κουλτούρα της κοινωνικής δικτύωσης και των εικονικών κοινοτήτων που αναπτύχθηκε και εξελίχθηκε από τους χρήστες του διαδικτύου. Οι αφετηρίες όμως ήταν αρκετά διαφορετικές. Στην περίπτωση της εικονικής πραγματικότητας, η εκκίνηση ήταν η ρεαλιστική παρουσίαση και αλληλεπίδραση με ένα συνθετικό περιβάλλον, ενώ οι κοινωνικοί κόσμοι προέκυψαν από την ανάγκη για τη συν-παρουσία πολλαπλών χρηστών σε έναν κοινό «τόπο», με στόχο την επικοινωνία και τη διασκέδαση. Αρχικά θα επικεντρωθούμε στην εικονική πραγματικότητα, στο όραμά της και στις βασικές έννοιες που τη συνοδεύουν.

4.1 Ορισμός και βασικές έννοιες

Η Εικονική Πραγματικότητα είναι μια επιστημονική περιοχή η οποία μελετάει την κατασκευή συνθετικών διαδραστικών περιβαλλόντων και την εμπύθιση των χρηστών σε αυτά μέσω της αξιοποίησης εξειδικευμένου υλικού και λογισμικού. Εύκολα μπορεί να παρατηρήσει κάποιος ότι ο ίδιος ο τίτλος της περιοχής είναι αντιφατικός. Αφού αυτό που ονομάζουμε πραγματικότητα, δηλαδή ο φυσικός κόσμος που αντιλαμβανόμαστε με τις αισθήσεις μας, είναι μία, πώς μπορεί να είναι εικονική; Άλλωστε, το επίθετο «εικονικός» σημαίνει κάτι ψεύτικο ή ιδεατό· σίγουρα όχι πραγματικό. Η αλήθεια είναι ότι, ενώ υπάρχει αυτή η εγγενής αντίφαση στην ονομασία, οι λέξεις καταφέρνουν να συμπυκνώσουν τον πραγματικό στόχο: να δημιουργήσουμε τεχνητά περιβάλλοντα τα οποία παρουσιάζονται τόσο πειστικά στον χρήστη που τα αντιλαμβάνεται σχεδόν σαν «την πραγματικότητα». Βεβαίως, όπως αναφέραμε και στην αρχή της εισαγωγής μας, σήμερα δεν είμαστε ακόμα τόσο κοντά σε μια τέτοια εξιδανικευμένη κατάσταση, αλλά είναι αλήθεια ότι σε πολλά εργαστηριακά συστήματα που αξιοποιούν εξειδικευμένο υλικό, η εμπειρία χρήσης χαρακτηρίζεται αρκετά πλούσια και ρεαλιστική. Σε κάθε περίπτωση, οι ερευνητές της εικονικής πραγματικότητας έδωσαν έμφαση εξαρχής στην αληθοφάνεια και στη φυσικότητα της εμπειρίας· στο «ξεγέλασμα» των αισθήσεων μέσω της αντικατάστασης των δεδομένων του φυσικού κόσμου με αντίστοιχης μορφής και ποιότητας δεδομένα προερχόμενα από ένα σχεδιασμένο συνθετικό περιβάλλον ελεγχόμενο από τον υπολογιστή.

Σύμφωνα με τους Burdea και Coiffet (1994), τα συστήματα εικονικής πραγματικότητας ενοποιούν τρία βασικά στοιχεία: την *εμπύθιση*, την *αλληλεπίδραση* και τη *φαντασία*. Τα στοιχεία αυτά, λόγω του κοινού αρχικού γράμματος στην αγγλική ονομασία, είναι γνωστά και ως τα τρία I της εικονικής πραγματικότητας (Immersion, Interaction, Imagination) και αντιμετωπίζονται ως τρεις ανεξάρτητοι άξονες βάσει των οποίων χαρακτηρίζεται η ποιότητα ενός συστήματος. Ο βαθμός εμπύθισης σχετίζεται με τον αριθμό των αισθητηρίων καναλιών του χρήστη, που τροφοδοτούνται με δεδομένα από τον εικονικό κόσμο και με την ποιότητα των δεδομένων αυτών. Τα συστήματα που σήμερα ονομάζονται «εμβυθισμένα» (immersive) καλύπτουν πλήρως το οπτικό πεδίο του χρήστη μέσω ειδικής συσκευής (π.χ. κράνος), την ακοή του μέσω ακουστικών και μέρος της αφής του (συνήθως το ένα χέρι μέσω ειδικού γαντιού). Ο άξονας της αλληλεπίδρασης έχει να κάνει με τις δυνατότητες του χρήστη να επενεργήσει στο περιβάλλον και με τη φυσικότητα της αλληλεπίδρασης. Ο βαθμός ομοιότητας των ενεργειών που πραγματοποιεί ο χρήστης στο εικονικό περιβάλλον με τις αντίστοιχες ενέργειες του φυσικού περιβάλλοντος καθορίζει και την ποιότητα της αλληλεπίδρασης. Για παράδειγμα, καλής ποιότητας αλληλεπίδραση θα ήταν αν ένας χρήστης μπορούσε να κοιτάξει γύρω του στο εικονικό περιβάλλον απλά κινώντας το κεφάλι του, όπως θα έκανε στην πραγματικότητα (κάτι που είναι εφικτό με τις σημερινές τεχνολογίες). Τέλος, ο άξονας της φαντασίας έχει να κάνει με τον βαθμό στον οποίο ο χρήστης μπορεί να αντιληφθεί τον εικονικό χώρο και τα εικονικά αντικείμενα ως «υπαρκτά». Αυτό είναι βεβαίως υποκειμενικό, αλλά σε κάθε περίπτωση επηρεάζεται από την ποιότητα της απεικόνισης, την ομαλότητα της κίνησης και γενικότερα από το κατά πόσο όλα όσα παρουσιάζονται στον χρήστη μπορούν να γίνουν «πιστευτά», δηλαδή από την αληθοφάνεια (believability) του κόσμου.

Μία ακόμη έννοια που σχετίζεται με την υποκειμενική αίσθηση του χρήστη σε σχέση με το εικονικό περιβάλλον στο οποίο είναι εμβυθισμένος και η οποία εμφανίζεται συχνά στη σχετική βιβλιογραφία είναι η *παρουσία* (presence) (Schuemie & Liebert, 2001). Αρχικά ο όρος «παρουσία» χρησιμοποιήθηκε για να

περιγράφει την αίσθηση του χρήστη ότι «βρίσκεται εκεί» (sense of being there), δηλαδή ότι νιώθει μέρος του εικονικού περιβάλλοντος και όχι απλώς εξωτερικός παρατηρητής αυτού (Steuer, 1992). Παρόλο που η παραπάνω περιγραφή έχει ομοιότητες με την έννοια της εμπύθισης, υπάρχει μια σαφής διάκριση μεταξύ των δύο: Η εμπύθιση σχετίζεται με τη σύνθεση του συστήματος, δηλαδή τον βαθμό στον οποίο το σύστημα καταφέρνει να τροφοδοτήσει τις αισθήσεις του χρήστη με δεδομένα του συνθετικού περιβάλλοντος, απομονώνοντάς τον από τον φυσικό κόσμο. Αντίθετα, η παρουσία είναι περισσότερο υποκειμενική, δηλαδή σχετίζεται με το πώς αντιλαμβάνεται ο χρήστης τα δεδομένα αυτά, κατά πόσο όντως τον «πειθούν» ότι βρίσκεται μέσα σε αυτό το συνθετικό περιβάλλον, κατά πόσο το δέχεται ως κάτι το «φυσιολογικό». Στην πορεία, η έννοια της παρουσίας ερμηνεύτηκε ποικιλοτρόπως, ενώ προέκυψαν και νέες σχετικές έννοιες, όπως η συν-παρουσία (co-presence), δηλαδή η αίσθηση ότι οι υπόλοιποι χρήστες συνυπάρχουν στον ίδιο χώρο (Bulu, 2012). Σήμερα η παρουσία αναλύεται πλέον σε δύο ανεξάρτητες έννοιες σύμφωνα με τον Mel Slater (2009): στην ψευδαίσθηση χώρου (place illusion) και στην ευλογοφάνεια (plausibility). Το πρώτο έχει να κάνει με το κατά πόσο δημιουργείται στον χρήστη η ψευδαίσθηση ότι βρίσκεται σε έναν άλλο χώρο, έννοια που σχεδόν ταυτίζεται τελικά με την εμπύθιση, και το δεύτερο με το κατά πόσο το περιβάλλον που του παρουσιάζεται του φαίνεται εύλογο, δηλαδή το δέχεται ως «πραγματικό». Δεν είναι τυχαίο ότι χρησιμοποιείται ο όρος «ευλογοφάνεια» και όχι «ρεαλισμός», καθώς φαίνεται ότι μπορούμε να φανταστούμε ως «αληθινά» ακόμα και περιβάλλοντα που σε καμία περίπτωση δεν θυμίζουν τον πραγματικό κόσμο, όπως είναι για παράδειγμα οι κόσμοι φαντασίας της λογοτεχνίας και των σημερινών ψηφιακών παιχνιδιών.

4.2 Ιστορική αναδρομή

Όσο παράξενο κι αν ακούγεται, η εικονική πραγματικότητα έχει πολύ μεγαλύτερη ιστορία σε σχέση με άλλες τεχνολογίες που χρησιμοποιούμε καθημερινά, όπως για παράδειγμα το Internet και ο Παγκόσμιος Ιστός. Η ιδέα και οι πρώτες υλοποιήσεις είναι πολύ πιο παλιές από τις παραθυρικές εφαρμογές, τα γραφικά περιβάλλοντα διεπαφής και τα τρισδιάστατα παιχνίδια. Το 1956 ο Morton Heilig δημιούργησε το Sensorama, ένα ψυχαγωγικό σύστημα που πρόσφερε στους χρήστες την εμπειρία ενός προκατασκευασμένου συνθετικού περιβάλλοντος (π.χ. διαδρομή με μοτοσυκλέτα στο Μανχάταν) μέσω πολλαπλών αισθήσεων. Περιλάμβανε εικόνα, ήχο, οσμή, δόνηση και αέρα. Το περιβάλλον δεν παραγόταν από υπολογιστή και δεν υπήρχε κανενός είδους διάδραση με αυτό, όμως για πρώτη φορά έγινε μια προσπάθεια αξιοποίησης πολλαπλών αισθήσεων για την παροχή μιας πιο πλούσιας εμπειρίας χρήστη και την εμπύθισή του σε μια εναλλακτική πραγματικότητα.

Ο βασικός οραματιστής της εικονικής πραγματικότητας, ο Ivan Sutherland, ήδη από το 1965 δημοσίευσε ένα άρθρο με τίτλο *The ultimate display* («Η απόλυτη απεικόνιση») στο οποίο προέβλεπε ότι μια μέρα οι υπολογιστές θα παρέχουν «ένα παράθυρο σε εικονικούς κόσμους». Στο άρθρο αυτό περιέγραφε την ιδέα ενός μηχανισμού απεικόνισης όπου οι χρήστες θα ήταν σε θέση να διαδράσουν με αντικείμενα ενός κόσμου που δεν θα ακολουθούσε υποχρεωτικά τους νόμους της φυσικής. Θα λειτουργούσε ως «καθρέπτης σε μια μαθηματική χώρα των θαυμάτων» (Sutherland, 1965). Η περιγραφή της ιδέας περιλάμβανε την παραγωγή τόσο οπτικών όσο και κιναισθητικών ερεθισμάτων από τον υπολογιστή, τα οποία θα αντικαθιστούσαν τα αντίστοιχα δεδομένα του πραγματικού κόσμου. Παρόλο που ο επίσημος όρος δεν είχε ακόμα επινοηθεί, αυτή ήταν η πρώτη φορά που ένας ερευνητής περιέγραφε ένα σύστημα εικονικής πραγματικότητας διαδραστικό και ελεγχόμενο από υπολογιστή.

Τρία χρόνια αργότερα ο Sutherland κάνει το πρώτο βήμα προς την υλοποίηση αυτής της ιδέας όταν μαζί με τον συνάδελφό του David Evans ιδρύει την Evans and Sutherland Computer Corp. και κατασκευάζουν το πρώτο «κράνος εικονικής πραγματικότητας» ή «απεικόνιση προσαρμοσμένη στο κεφάλι» (head mounted display – HMD). Η συσκευή αυτή, της οποίας οι μετεξελίξεις αποτελούν μέχρι και σήμερα βασικό τμήμα των συστημάτων εικονικής πραγματικότητας, έχει δύο ιδιαίτερα χαρακτηριστικά σε σχέση με μια απλή απεικόνιση σε οθόνη: τη στερεοσκοπία και την ανίχνευση κίνησης. Η στερεοσκοπία είναι μια τεχνική η οποία δημιουργεί στον εγκέφαλο του χρήστη τεχνητή αίσθηση του βάθους μέσω της παροχής διαφορετικών, κατάλληλα διαμορφωμένων, απόψεων ενός περιβάλλοντος στα δύο μάτια. Εκμεταλλεύεται το εγγενές χαρακτηριστικό της στερεοοπτικής όρασης, δηλαδή το ότι ο ανθρώπινος εγκέφαλος συνδυάζει τις ελαφρώς διαφορετικές εικόνες από τα δύο μάτια για να αντιληφθεί το βάθος. Η ανίχνευση κίνησης από την άλλη μεριά δίνει πληροφορίες στο σύστημα σχετικά με τη θέση και τον προσανατολισμό του χρήστη. Τον μηχανισμό αυτόν εκμεταλλεύονται τα συστήματα εικονικής πραγματικότητας ώστε να αναπροσαρμόσουν την άποψη του περιβάλλοντος σε σχέση με τη νέα οπτική του χρήστη, δίνοντας έτσι την αίσθηση ότι τα «εικονικά» αντικείμενα υπάρχουν γύρω του και παραμένουν στη θέση τους ανεξάρτητα από τις δικές του κινήσεις. Μια τέτοια αίσθηση φαίνεται λοιπόν πως

δημιουργούσε και το σύστημα του Sutherland. Παρόλο που τα γραφικά ήταν ιδιαίτερα πρώιμα σε σχέση με αυτό που έχουμε σήμερα στο μυαλό μας ως 3D (για την ακρίβεια ήταν γεωμετρικά στερεά ζωγραφισμένα με γραμμές), ο χρήστης αισθανόταν πως τα τρισδιάστατα αντικείμενα «υπήρχαν» γύρω του.

Παρά τις υψηλές προσδοκίες που ήγειρε η υλοποίηση των Evans και Sutherland, τη δεκαετία του '70 οι εξελίξεις στον χώρο ήταν αργές. Ο βασικός λόγος φαίνεται να είναι το ότι η τεχνολογία ήταν ανέτοιμη να ανταποκριθεί στις απαιτήσεις αυτών των συστημάτων. Εμφανίστηκαν τα πρώτα εμπορικά ψηφιακά παιχνίδια με τη μορφή κονσόλας (π.χ. το Pong της Atari που κυκλοφόρησε το 1972), και συνεπακόλουθα αναπτύχθηκε και εξελίχθηκε σημαντικά ο χώρος των διαδραστικών γραφικών σε πραγματικό χρόνο. Η Evans and Sutherland ανέπτυξε τον πρώτο προσομοιωτή πτήσης βασισμένο σε υπολογιστή (Nonoview, 1973), άρα ουσιαστικά την πρώτη εμπορική εφαρμογή εικονικής πραγματικότητας. Λίγα χρόνια αργότερα κατασκευάστηκε στο Παν/μιο του Illinois το πρώτο «γάντι εικονικής πραγματικότητας» ή «γάντι δεδομένων» (data glove), το Sayre Glove (1977). Είχε τη δυνατότητα να μετράει προσεγγιστικά το λύγισμα του κάθε δακτύλου του χρήστη που το φορούσε στέλνοντας τα δεδομένα σε υπολογιστή, και κατά συνέπεια να ανακατασκευάσει τη συνολική διάταξη του χεριού. Αυτή ήταν μάλλον και η πιο σημαντική συνεισφορά στον χώρο τη συγκεκριμένη δεκαετία, μιας και τα γάντια δεδομένων αποτελούν ακόμα και σήμερα ένα αξιόπιστο μέσο για τη φυσική αλληλεπίδραση των χρηστών με τα αντικείμενα του συνθετικού περιβάλλοντος.

Τη δεκαετία του '80 οι εξελίξεις επιταχύνονται κυρίως χάρη στην ευρεία διάδοση και εξέλιξη των υπολογιστικών συστημάτων. Για πρώτη φορά κατασκευάστηκαν νέα, πιο εξελιγμένα HMDs πέρα από την αρχική υλοποίηση των Evans και Sutherland του 1968: το 1983 από τον Mark Callahan στο MIT και το 1987 από τη NASA, που είχε στο μεταξύ ιδρύσει το δικό της εργαστήριο έρευνας σε τεχνολογίες εικονικής πραγματικότητας. Τα γάντια δεδομένων εξελίχθηκαν κι αυτά (DataGlove, 1984) παρέχοντας επιπλέον πληροφορίες θέσης και προσανατολισμού του χεριού πέρα από τη διάταξη των δακτύλων. Κατά συνέπεια ο υπολογιστής ήταν σε θέση να γνωρίζει την πραγματική θέση του χεριού στον χώρο σε σχέση με το συνθετικό τρισδιάστατο περιβάλλον. Οι χρήστες μπορούσαν πλέον να κατευθύνουν το χέρι τους προς τα αντικείμενα του περιβάλλοντος και να επενεργήσουν σε αυτά. Η NASA ανέπτυξε το 1988 τη συσκευή Convolvotron για τη χωροθέτηση του ήχου με αποτέλεσμα τα ηχητικά συμβάντα των συνθετικών περιβαλλόντων να ακούγονται σαν να προέρχονται από την αντίστοιχη θέση που υποτίθεται ότι τα παράγει. Επιπλέον, η γνωστή από τον χώρο της τρισδιάστατης μοντελοποίησης εταιρεία Autodesk ανέπτυξε το 1989 ένα πρόγραμμα δημιουργίας τρισδιάστατων κόσμων σε PC (CyberSpace). Τέλος, την ίδια χρονιά αποδόθηκε η φράση «Εικονική Πραγματικότητα» σε όλη αυτή την κατηγορία συστημάτων. Η χρήση του όρου έγινε από την εταιρεία VPL, η οποία κατασκεύασε ένα ολοκληρωμένο σύστημα με κράνη και γάντια δεδομένων για δύο χρήστες, το RB-2.

Τα επόμενα χρόνια οι εξελίξεις του χώρου διαδόθηκαν με μεγάλο ενθουσιασμό από τα μέσα της εποχής και δημιούργησαν ακόμα μεγαλύτερες προσδοκίες. Για πρώτη φορά εμφανίστηκαν δημόσιες εγκαταστάσεις εικονικής πραγματικότητας ψυχαγωγικού χαρακτήρα, όπως το Virtuality της W-Industries (1990). Το κοινό είχε πλέον τη δυνατότητα να έρθει σε επαφή με τις νέες, πολλά υποσχόμενες τεχνολογίες και να αποκτήσει την εμπειρία της στερεοσκοπίας και της ανίχνευσης κίνησης που παρείχαν τα HMDs της εποχής. Μια άλλη, ακόμα σημαντικότερη εξέλιξη ήταν η εμφάνιση των προβολικών συστημάτων ως εναλλακτική του κράνους. Έτσι, το 1992 παρουσιάστηκε το CAVE, ένα σύστημα προβολής κατασκευασμένο από το Παν/μιο του Illinois το οποίο τοποθετεί τον χρήστη σε ένα μικρό δωμάτιο και τα δεδομένα του εικονικού περιβάλλοντος προβάλλονται στους τοίχους του. Ο χρήστης δεν χρειάζεται πλέον να φορέσει ειδικό κράνος· ο εικονικός κόσμος υπάρχει γύρω του. Αρκεί να χρησιμοποιήσει αισθητήρες σύλληψης κίνησης και να φοράει ειδικά γυαλιά για στερεοσκοπική όραση για να έχει πλήρη εμπειρία εμπύθισης. Η ιδέα των προβολικών περιβαλλόντων επεκτάθηκε και σε απλούστερα συστήματα, στα οποία η προβολή γινόταν σε μία μεγάλη επιφάνεια αντί δωματίου, όπως τα Responsive Workbench (1994) και ImmersaDesk (1995). Την ίδια περίπου περίοδο εγκαθιδρύθηκε ο χώρος και σε ακαδημαϊκό επίπεδο, καθώς εμφανίστηκαν τα πρώτα διεθνή επιστημονικά συνέδρια με αντικείμενο την Εικονική Πραγματικότητα.

Σήμερα είναι διαθέσιμο ένα μεγάλο εύρος κατηγοριών υλικού, λογισμικού αλλά και ολοκληρωμένων λύσεων για την ανάπτυξη συστημάτων εικονικής πραγματικότητας. Πέρα από ισχυρά υπολογιστικά συστήματα, τα οποία είναι σε κάθε περίπτωση απαραίτητα, μπορεί κάποιος να βρει μηχανισμούς ανίχνευσης κίνησης διάφορων τεχνολογιών (οπτικούς, μηχανικούς, ηλεκτρομαγνητικούς, ηχητικούς), συσκευές εισόδου, όπως γάντια δεδομένων και τρισδιάστατα ποντίκια, μηχανισμούς απτικής ανάδρασης που παρέχουν ερεθίσματα αφής ή/και ασκούν δυνάμεις στο χέρι του χρήστη δίνοντάς του την αίσθηση ότι «αγγίζει» τα αντικείμενα, προβολικά συστήματα σε μία ή περισσότερες επιφάνειες, στερεοσκοπικά γυαλιά, κράνη εικονικής πραγματικότητας και πολλά άλλα (Brooks, 1999). Αντίστοιχα, σε επίπεδο λογισμικού προσφέρονται εργαλεία τρισδιάστατης μοντελοποίησης, «μηχανές» τρισδιάστατων γραφικών και φυσικής, ανοιχτές βιβλιοθήκες

λογισμικού με εξειδικευμένες λειτουργίες (π.χ. φυσική, γραφικά, χειρισμός συσκευών κ.λπ.), ακόμα και ολοκληρωμένα συστήματα που διασυνδέουν το υλικό με τρισδιάστατους κόσμους.

4.3 Περιβάλλοντα επιφάνειας εργασίας

Παρά τις ενθαρρυντικές εξελίξεις των σχετικών τεχνολογιών, τα συστήματα Εικονικής Πραγματικότητας όπως τα οραματίστηκαν οι πρωτοπόροι του χώρου δεν έφτασαν ποτέ στο απαιτούμενο επίπεδο ωριμότητας ώστε να γίνουν μέρος της καθημερινότητάς μας. Υπάρχουν πολλοί λόγοι γι' αυτό, με βασικότερους το κόστος και την ευχρηστία. Πράγματι, αν δει κάποιος τις τιμές πώλησης των παραπάνω κατηγοριών υλικού, γρήγορα θα καταλάβει ότι το κόστος αγοράς και εγκατάστασης ενός ολοκληρωμένου συστήματος μπορεί να φτάσει αρκετές δεκάδες χιλιάδες ευρώ. Συνυπολογίζοντας και το απαραίτητο κόστος συντήρησης, με δεδομένο ότι ο εξοπλισμός αυτός είναι αρκετά ευαίσθητος, η συνολική επένδυση είναι μάλλον μεγάλη για όσους θέλουν να επωφεληθούν από τη συγκεκριμένη τεχνολογία, και σίγουρα πολύ μακριά από τα χρήματα που θα ήταν διατεθειμένος να δαπανήσει ο τελικός χρήστης για ένα τέτοιο σύστημα. Από την άλλη μεριά, οι τεχνολογίες αυτές έχουν κατηγορηθεί για διάφορων ειδών προβλήματα ευχρηστίας, κάτι που είναι μάλλον αναμενόμενο με δεδομένο τον πολύ μικρό αριθμό συνολικών χρηστών σε σχέση με άλλα τεχνολογικά επιτεύγματα με ευρεία διάδοση στο κοινό (π.χ. περιφερειακά, κινητά τηλέφωνα κ.ά.). Το πιο βασικό πρόβλημα φαίνεται να είναι η ζαλάδα που προκαλείται από την παρατεταμένη έκθεση των χρηστών σε περιβάλλοντα εικονικής πραγματικότητας, ένα φαινόμενο που αναφέρεται στη βιβλιογραφία ως «ζαλάδα κίνησης» (cybersickness, simulator sickness). Η στερεοσκοπία, η καθυστέρηση (lag) μεταξύ της κίνησης του κεφαλιού και της αντίστοιχης οπτικής απόκρισης του συστήματος, το επιπρόσθετο βάρος των φορετών συσκευών (π.χ. HMD), προβλήματα στον ρυθμό ανανέωσης των γραφικών και στην απεικόνιση είναι ορισμένοι από τους παράγοντες που συνεισφέρουν αρνητικά στην εμπειρία του χρήστη και σε ορισμένες περιπτώσεις προκαλούν αίσθηση ναυτίας.

Τα παραπάνω προβλήματα έδωσαν ώθηση την τελευταία δεκαετία σε έναν άλλο, αρχικά παρεξηγημένο χώρο, αυτόν της εικονικής πραγματικότητας επιφάνειας εργασίας (Desktop Virtual Reality). Ο όρος αυτός αναφέρεται σε εικονικά περιβάλλοντα που τρέχουν σε οικιακούς υπολογιστές και επομένως η διεπαφή είναι λίγο πολύ η παραδοσιακή: μη στερεοσκοπική απεικόνιση σε οθόνη υπολογιστή, διάδραση μέσω πληκτρολογίου και ποντικιού. Προφανώς, αυτού του τύπου τα περιβάλλοντα στερούνται σε μεγάλο βαθμό το στοιχείο της εμπύθισης που αναφέραμε πιο πάνω, ενώ και η αλληλεπίδραση είναι πολύ λιγότερο φυσική σε σχέση με συστήματα που περιλαμβάνουν ανίχνευση κίνησης και χρήση γαντιού δεδομένων. Δεν είναι τυχαίο ότι τη δεκαετία του '90 οι ερευνητές αναφέρονταν υποτιμητικά σε αυτά ως «εικονική πραγματικότητα ενυδρείου» (fishtank VR) με την έννοια ότι ο χρήστης αισθάνεται περισσότερο εξωτερικός παρατηρητής (όπως ο θεατής ενός ενυδρείου) του περιβάλλοντος παρά εμπυθισμένος σε αυτό. Παρόλα αυτά, ο χώρος της εικονικής πραγματικότητας επιφάνειας εργασίας γνώρισε μεγάλη άνθιση χάρη στις εξελίξεις στον χώρο των παιχνιδιών υπολογιστών που συμπαρέσυραν και την τεχνολογία τρισδιάστατων γραφικών σε αντίστοιχα γρήγορους ρυθμούς ανάπτυξης (Zyda, 2005). Πράγματι, οι σημερινοί οικιακοί υπολογιστές πλαισιώνονται από αρκετά ισχυρούς επεξεργαστές γραφικών που έχουν τη δυνατότητα να αποδώσουν ποιοτικά γραφικά και συνθετική κίνηση οδηγώντας σε υψηλά επίπεδα ρεαλισμού. Χάρη σε αυτές τις δυνατότητες μπορούν πλέον οι χρήστες να έρθουν σε επαφή με τρισδιάστατα περιβάλλοντα προσομοίωσης, χωρίς όμως τα προβλήματα κόστους και ευχρηστίας που αναφέραμε πιο πάνω. Τέλος, παρά τον χαμηλό βαθμό εμπύθισης, η αίσθηση της παρουσίας, με την έννοια της ενεργού εμπλοκής των χρηστών με το περιβάλλον, διατηρείται υψηλή σε καλά σχεδιασμένους κόσμους παιχνιδιών και άλλα τρισδιάστατα περιβάλλοντα, επομένως σε κάποιο βαθμό ο στόχος έχει προσεγγιστεί.

5 Από τους κόσμους κειμένου στα τρισδιάστατα γραφικά

Όπως αναφέρθηκε και παραπάνω, πέρα από την τεχνολογική εξέλιξη της Εικονικής Πραγματικότητας υπήρξε κι ένα δεύτερο ρεύμα που συνεισέφερε στη σημερινή μορφή των εικονικών κόσμων: τα παιχνίδια ρόλων και οι κοινωνικοί κόσμοι. Οι εφαρμογές αυτές ξεκίνησαν ως περιβάλλοντα κειμένου, διαδόθηκαν χάρη στο διαδίκτυο και εξελίχθηκαν σε γραφικά περιβάλλοντα (Bartle, 2004).

Ήδη από τα μέσα της δεκαετίας του '70, όταν οι υπολογιστές ήταν πλέον διαθέσιμοι σε μεγάλο αριθμό φοιτητών και ερευνητών στα πανεπιστήμια, αλλά οι δυνατότητες γραφικής απεικόνισης ήταν από εξαιρετικά περιορισμένες έως ανύπαρκτες, έκαναν την εμφάνισή τους κόσμοι κειμένου με τη μορφή παιχνιδιών. Το πρώτο

παιχνίδι περιπέτειας (adventure game), μια κατηγορία παιχνιδιών που είναι δημοφιλής μέχρι και σήμερα, το Colossal Cave Adventure, δημιουργήθηκε το 1976. Ο χρήστης έλεγχε τις κινήσεις και τις ενέργειες ενός χαρακτήρα παίρνοντας πληροφορίες από τη δική του οπτική και προσπαθούσε να εκπληρώσει τους στόχους του στο περιβάλλον του παιχνιδιού. Μόνο που τόσο οι εντολές που έδινε στον χαρακτήρα όσο και η περιγραφή του περιβάλλοντος ήταν σε μορφή κειμένου. Για παράδειγμα, ο χρήστης μπορούσε να κινήσει τον χαρακτήρα του γράφοντας μία από τις τέσσερις διευθύνσεις, π.χ. «north» (βόρεια). Στη συνέχεια το παιχνίδι του περιέγραφε λεκτικά τη νέα τοποθεσία, τα αντικείμενα και τους χαρακτήρες που βρίσκονταν σε αυτήν και τις ενέργειες που μπορούσε να κάνει ο χαρακτήρας. Ενώ λοιπόν απουσίαζαν παντελώς τα γραφικά, για πρώτη φορά εμφανίζονταν ένα παιχνίδι που παρουσίαζε έναν συνθετικό κόσμο με αντικείμενα και χαρακτήρες (π.χ. τέρατα που έπρεπε να σκοτώσει) και ο χρήστης αλληλεπιδρούσε με αυτά ελέγχοντας έναν μοναδικό χαρακτήρα-ενσάρκωση.

Οι εξελίξεις ήταν ακόμα πιο ενδιαφέρουσες όταν αυτά τα παιχνίδια έγιναν πολυχρηστικά. Έτσι, το 1978 δημιουργήθηκε από φοιτητές του Παν/μίου του Essex το πρώτο παιχνίδι περιπέτειας για πολλούς χρήστες, το οποίο ονομάστηκε MUD (Multi-User Dungeon), δηλαδή «πολυχρηστικό μπουντρούμι». Ο όρος αυτός χρησιμοποιήθηκε στη συνέχεια για να περιγράψει όλη αυτή την κατηγορία περιβαλλόντων, η οποία έγινε εξαιρετικά δημοφιλής μέχρι και τα μέσα της δεκαετίας του '90. Η συμμετοχή των χρηστών σε αυτά τα περιβάλλοντα γινόταν αρχικά μέσω του τοπικού δικτύου των πανεπιστημίων, αργότερα με τηλεφωνική σύνδεση σε ιδιωτικούς παρόχους δικτυακών υπηρεσιών (Bulletin Board Systems) και μετά από κάποια χρόνια μέσω Internet. Για πρώτη φορά λοιπόν εμφανίστηκαν περιβάλλοντα στα οποία συνυπήρχαν πολλαπλοί χρήστες, που μπορούσαν να εξερευνήσουν τον χώρο, να βρεθούν και να επικοινωνήσουν μεταξύ τους, να συνεργαστούν ή να ανταγωνιστούν στα πλαίσια κάποιων στόχων κ.ά. Στοιχεία δηλαδή που εντοπίζει κάποιος ακόμη και σήμερα στους δημοφιλείς εικονικούς κόσμους. Επιπλέον, όσο εξελίσσονταν αυτά τα περιβάλλοντα εμφανίζονταν και ορισμένες ενδιαφέρουσες προσθήκες. Μια μετεξέλιξη του αρχικού MUD «έσπαγε» το παιχνίδι σε δύο κομμάτια: σε μια αμετάβλητη μηχανή παιχνιδιού και στην περιγραφή του κόσμου γραμμένη σε μια γλώσσα που δημιουργήθηκε γι' αυτόν τον σκοπό, την MUDDL. Αυτή η διάκριση επέτρεπε σε άλλους χρήστες να αξιοποιήσουν τη μηχανή του παιχνιδιού και να δημιουργήσουν τους δικούς τους κόσμους, όπως και συνέβη. Ένα MUD που κυκλοφόρησε αρκετά χρόνια αργότερα, το TinyMUD (1989), επέτρεπε στους χρήστες να επεκτείνουν τον κόσμο δημιουργώντας τους δικούς τους νέους χώρους, και αντίστοιχα να ορίσουν τα δικά τους νέα αντικείμενα (πάντα σε μορφή κειμένου). Το αποτέλεσμα ήταν ότι οι χρήστες του κόσμου αυτού ενδιαφέρονταν περισσότερο για την επικοινωνία και τη συν-δημιουργία παρά για την εκπλήρωση κάποιας προκαθορισμένης αποστολής (Curtis & Nichols, 1994). Ήταν στην πραγματικότητα ο πρόγονος των «κοινωνικών» κόσμων (Turkle, 1994). Τέλος, το LPMUD που κυκλοφόρησε την ίδια χρονιά παρείχε για πρώτη φορά και μια γλώσσα προγραμματισμού εντός του κόσμου που επέτρεπε στους χρήστες, εκτός από το να δημιουργήσουν νέα αντικείμενα, να ορίσουν και τη λειτουργικότητά τους.

Η ραγδαία ανάπτυξη της τεχνολογίας γραφικών στους οικιακούς υπολογιστές από τα μέσα της δεκαετίας του '90 και μετά μετεξέλιξε τους κόσμους κειμένου σε διαδραστικά γραφικά περιβάλλοντα. Από τη μία μεριά δημιουργήθηκαν κόσμοι παιχνιδιών πολλαπλών χρηστών ακολουθώντας τη λογική των παλιών MUDs και από την άλλη αναδύθηκαν «κοινωνικοί» κόσμοι με στόχο την επικοινωνία απομακρυσμένων χρηστών και τη δημιουργία εικονικών κοινοτήτων. Τα περιβάλλοντα αυτά είχαν πολλά από τα χαρακτηριστικά που βρίσκει κανείς και σήμερα στους εικονικούς κόσμους: γραφικό περιβάλλον απεικόνισης (σε κάποιες περιπτώσεις διςδιάστατο, αλλά στην πλειοψηφία τριςδιάστατο), αναπαράσταση χρηστών μέσω ενσαρκώσεων, ελεύθερη πλοήγηση στο περιβάλλον, άμεση επικοινωνία με άλλους χρήστες, αλληλεπίδραση με αντικείμενα, μοντέλα συνεργασίας ή ανταγωνισμού. Η κατηγορία των παιχνιδιών, που ονομάστηκαν MMORPGs (Massive Multiplayer Online Role Playing Games – δικτυακά παιχνίδια ρόλων πολλαπλών χρηστών) λόγω του ότι τα περισσότερα από αυτά ήταν βασισμένα σε παιχνίδια ρόλων —ή απλά MMOGs (Massive Multiuser Online Games)— γνώρισε μεγάλη άνθιση και γρήγορα απέκτησε πολύ μεγάλη βάση χρηστών. Ενδεικτικά αναφέρεται ότι τα δύο δημοφιλέστερα παιχνίδια αυτής της κατηγορίας, το Ultima Online και το EverQuest, το 2000 είχαν 230.000 και 300.000 συνδρομητές αντίστοιχα. Στην κατηγορία των κοινωνικών κόσμων εμφανίστηκαν διάφορα γραφικά περιβάλλοντα στα οποία οι χρήστες επέλεγαν την εικόνα της ενσάρκωσής τους, κινούνταν σε χώρους ή δωμάτια και συνομιλούσαν με άλλους χρήστες, με πιο δημοφιλές το the Palace (1995). Επί της ουσίας σε επίπεδο λειτουργιών δεν διέφεραν ιδιαίτερα από τους χώρους συνομιλίας κειμένου (chat rooms), όπως το IRC, αλλά πρόσθεσαν τη γραφική απεικόνιση του περιβάλλοντος και των χρηστών. Το σύστημα που ξεχώρισε από αυτή την κατηγορία ήταν το Active Worlds, ένα τριςδιάστατο περιβάλλον πολλαπλών χρηστών που επέτρεπε την εξερεύνηση, επικοινωνία αλλά και προσθήκη περιεχομένου για χρήστες με συνδρομή. Η πρώτη έκδοση εμφανίστηκε το 1995, αλλά ο κόσμος συνεχίζει και εξελίσσεται μέχρι σήμερα αποκτώντας πολλαπλά νέα χαρακτηριστικά και χώρους εφαρμογής. Επί της ουσίας πρόκειται για τον προάγγελο, και στη συνέχεια

μεγάλο ανταγωνιστή, του δημοφιλούς εικονικού κόσμου Second Life, στον οποίο θα αναφερθούμε εκτενέστερα στη συνέχεια.

Αυτό που προκύπτει λοιπόν από τις παράλληλες εξελίξεις στην εικονική πραγματικότητα, στους κοινωνικούς κόσμους και στα παιχνίδια ρόλων είναι μια σύγκλιση σε περιβάλλοντα που μοιράζονται πολλά κοινά χαρακτηριστικά. Από την οπτική της εικονικής πραγματικότητας μιλάμε για πολυχρηστικά ή διαμοιρασμένα εικονικά περιβάλλοντα επιφάνειας εργασίας (multi-user ή shared desktop virtual environments), από την οπτική των παιχνιδιών για δικτυακούς κόσμους πολλαπλών χρηστών και από την οπτική των κοινωνικών κόσμων για τρισδιάστατους εικονικούς κόσμους. Τα κοινά χαρακτηριστικά τους είναι βεβαίως αυτά που αναφέραμε και στην εισαγωγή μας, δηλαδή το τρισδιάστατο περιβάλλον, η πολυχρηστικότητα, η διάρκεια και οι πολλαπλές δυνατότητες αλληλεπίδρασης και επικοινωνίας. Όλα τα παραπάνω περιβάλλοντα λοιπόν, παρά τις επιμέρους διαφοροποιήσεις τους, αναφέρονται σήμερα στη βιβλιογραφία με τον κοινό όρο «Εικονικοί Κόσμοι» και αποτελούν το αντικείμενο μελέτης του παρόντος βιβλίου.

6 Σημερινή κατάσταση

Σήμερα ο πιο δημοφιλής κόσμος «γενικού σκοπού» είναι χωρίς αμφιβολία το Second Life. Δημιουργήθηκε από την εταιρεία Linden Labs το 2003, γρήγορα έγινε γνωστό και καλύφθηκε εκτενώς από τα έντυπα και ψηφιακά μέσα της εποχής, και μέσα σε λίγα χρόνια κατάφερε να αποκτήσει μια πολύ μεγάλη βάση χρηστών, γύρω στα 20 εκατομμύρια. Το Second Life είναι ένας ακόμη κοινωνικός κόσμος στα πρότυπα του Active Worlds, ο οποίος όμως συγκεντρώνει αρκετά ενδιαφέροντα νέα στοιχεία που τον κάνουν ελκυστικό για χρήση σε πολλούς διαφορετικούς χώρους. Δίνει στους χρήστες τη δυνατότητα να νοικιάσουν περιοχές και να χτίσουν ελεύθερα σε αυτές παρέχοντας ειδικά εργαλεία για την εύκολη κατασκευή περιεχομένου εντός κόσμου. Υποστηρίζει επικοινωνία μέσω φωνής, η οποία είναι μάλιστα «χωροθετημένη» διευκολύνοντας έτσι την απομακρυσμένη συνομιλία πολλαπλών χρηστών. Επιπλέον, δίνει πολλές δυνατότητες δημιουργίας και διαχείρισης ομάδων χρηστών με ρόλους και δικαιώματα, υποστηρίζοντας με αυτόν τον τρόπο συνεργατικά μοντέλα εργασίας. Τέλος, το πιο σημαντικό ίσως χαρακτηριστικό είναι ότι διαθέτει μια γλώσσα σεναρίων (scripting language), την LSL, την οποία μπορούν να χρησιμοποιήσουν οι χρήστες για να ορίσουν τη συμπεριφορά των αντικειμένων. Το αποτέλεσμα της μεγάλης ελευθερίας και των πολλαπλών εργαλείων που παρέχει το περιβάλλον είναι να μπορούν οι χρήστες πρακτικά να ορίσουν οι ίδιοι το πλαίσιο χρήσης της περιοχής τους ανάλογα με τα ενδιαφέροντά τους, όπως και συνέβη. Εκπαιδευτικά περιβάλλοντα, χώροι παρουσιάσεων, παιχνίδια ρόλων, προβολή χώρων πολιτισμού, ακόμα και έκθεση δημιουργιών είναι μερικοί από τους τρόπους αξιοποίησης του περιβάλλοντος (Kaplan & Haenlein, 2009). Το σημαντικό είναι ότι οι ίδιοι οι χρήστες επεκτείνουν τον κόσμο, δημιουργούν περιεχόμενο και ανακαλύπτουν τις δυνατότητες και τα όριά του. Στην εικόνα 1.5 απεικονίζεται μια σκηνή από τον εικονικό κόσμο του Second Life.



Εικόνα 1.5 Σκηνή από τον εικονικό κόσμο του Second Life.

Παρά τα πολλά καινοτόμα χαρακτηριστικά του, το Second Life έχει δυστυχώς και ορισμένους περιορισμούς. Το βασικό πρόβλημα είναι η υποχρέωση των χρηστών να καταβάλουν μηνιαία συνδρομή στη Linden Labs για να μπορούν να νοικιάσουν χώρο και, εντέλει, να κατασκευάσουν περιεχόμενο. Κατά συνέπεια, ενώ ο αριθμός των χρηστών είναι πολύ μεγάλος, αυτοί που έχουν τη δυνατότητα να αναπτύξουν νέους χώρους, αντικείμενα και εφαρμογές είναι μόνο όσοι πληρώνουν συνδρομή. Επιπλέον, το γεγονός ότι οι διακομιστές (servers) του κόσμου ανήκουν στη Linden Labs και οι χρήστες δεν έχουν καμία δυνατότητα παραμετροποίησης ή διαχείρισής τους περιορίζει τους σχεδιαστές εφαρμογών σε αυτά που μπορούν να κάνουν εντός κόσμου. Τροποποιήσεις, επεκτάσεις ή διασυνδέσεις με άλλες πηγές δεδομένων μπορούν να γίνουν μόνο από την ίδια την εταιρεία.

Σε μια προσπάθεια υπέρβασης των παραπάνω περιορισμών, μια ομάδα προγραμματιστών ξεκίνησε το 2007 την ανάπτυξη μιας εναλλακτικής πλατφόρμας, του OpenSimulator. Το OpenSimulator είναι μια πλατφόρμα ανοιχτού λογισμικού η οποία αναπαράγει σε μεγάλο βαθμό τη λειτουργικότητα του Second Life. Επί της ουσίας, το OpenSimulator αποτελείται από ένα σύνολο τεχνολογιών σε επίπεδο διακομιστή, ενώ η σύνδεση σε κόσμους που τρέχουν με αυτή την τεχνολογία γίνεται μέσω των ίδιων προγραμμάτων-πελάτη με τα οποία συνδέονται οι χρήστες και στο Second Life. Συνεπώς, σε επίπεδο διεπαφής οι δύο κόσμοι είναι σχεδόν ίδιοι. Εκεί που υπάρχει μεγάλη διαφορά είναι ότι ο κάθε διαχειριστής μπορεί να εγκαταστήσει και να τρέξει τον δικό του κόσμο OpenSimulator με τη δική του βάση χρηστών και οι χρήστες να αναπτύξουν ελεύθερα περιεχόμενο μέσα σε αυτόν. Επιπλέον, πολλαπλοί OpenSimulator servers μπορούν να διασυνδεθούν σε μια ενιαία αρχιτεκτονική (Grid) με τέτοιο τρόπο ώστε ο τελικός χρήστης να τους αντιλαμβάνεται όλους ως μέρος ενός ενιαίου κόσμου. Τέλος, διαφορετικοί κόσμοι μπορούν να διασυνδέονται μεταξύ τους (HyperGrid) και οι χρήστες να «τηλεμεταφέρονται» από τον έναν κόσμο στον άλλο. Είναι προφανές ότι οι παραπάνω δυνατότητες θυμίζουν την ελευθερία και ανεξαρτησία που υπάρχει και στον παγκόσμιο ιστό, όπου ο κάθε χρήστης μπορεί να εγκαθιστά τον δικό του διακομιστή, να ανεβάζει τις σελίδες του και να παρέχει συνδέσεις σε άλλες σελίδες οπουδήποτε στο διαδίκτυο. Το OpenSimulator είναι αντίστοιχα απείρως κλιμακούμενο (infinitely scalable), με την έννοια ότι νέοι κόσμοι μπορούν να προστίθενται συνεχώς και να διασυνδέονται με υπάρχοντες, χωρίς να επιβαρύνεται συνολικά η απόδοση του συστήματος.

Παράλληλα με τους κόσμους γενικού σκοπού, την τελευταία δεκαετία εμφανίστηκαν και αρκετά νέα παιχνίδια ρόλων τα οποία έγιναν εξαιρετικά δημοφιλή. Ο πιο επιτυχημένος κόσμος παιχνιδιού είναι αναμφίβολα το World of Warcraft, ένα περιβάλλον το οποίο πρωτοεμφανίστηκε το 2004 και μέχρι σήμερα έχουν εγγραφεί παραπάνω από εκατό εκατομμύρια χρήστες. Στο παιχνίδι παρουσιάζεται ένας ανοιχτός, εξελισσόμενος κόσμος φαντασίας όπου ο κάθε παίκτης επιλέγει και διαμορφώνει τον δικό του χαρακτήρα (εικόνα 1.6). Οι χαρακτήρες βελτιώνονται και αποκτούν νέες ικανότητες μέσα από τη διάδραση με το περιβάλλον, μάχονται, συνεργάζονται, οργανώνονται σε ομάδες, συμμετέχουν σε συμμαχίες και φέρνουν σε πέρας αποστολές στον κόσμο του παιχνιδιού. Ο σχηματισμός εικονικών κοινοτήτων και η ανάδυση ιδιαίτερων κοινωνικών φαινομένων εντός του παιχνιδιού έχει προκαλέσει το ενδιαφέρον ερευνητών από τις κοινωνικές επιστήμες, οι οποίοι βλέπουν τον κόσμο του World of Warcraft και άλλων σχετικών παιχνιδιών ως εν δυνάμει πλατφόρμες πειραματισμού και μελέτης (Bainbridge, 2007).



Εικόνα 1.6 Ο κόσμος του World of Warcraft.

Ένας άλλος, αρκετά διαφορετικής λογικής αλλά εξίσου δημοφιλής κόσμος παιχνιδιού είναι το Minecraft. Το Minecraft κυκλοφόρησε το 2011 και γρήγορα έγινε τόσο δημοφιλές, ώστε σήμερα είναι το παιχνίδι με τις περισσότερες πωλήσεις (19 εκατομμύρια) σε περιβάλλον προσωπικού υπολογιστή. Στον κόσμο του Minecraft δεν υπάρχουν συγκεκριμένοι στόχοι που τίθενται από το παιχνίδι. Οι χρήστες έχουν τη δημιουργική ελευθερία να διαμορφώσουν και να εξελίσσουν το περιβάλλον όπως αυτοί επιθυμούν. Ο κόσμος είναι κατασκευασμένος από μικρά τουβλάκια, τα οποία μπορούν να τοποθετηθούν σε συγκεκριμένες θέσεις, όπως σε παιχνίδια τύπου Lego. Τα τουβλάκια αναπαριστούν διάφορα υλικά, όπως χώμα, νερό, πέτρα κ.λπ., και έχουν αντίστοιχη συμπεριφορά κατά την αλληλεπίδραση μαζί τους. Εκτός από τον χαρακτήρα που ελέγχει ο παίκτης, στο παιχνίδι εμφανίζονται και άλλοι χαρακτήρες ελεγχόμενοι από τον υπολογιστή, όπως ζώα και χωρικοί, οι οποίοι αλληλεπιδρούν με αυτόν και με το κατασκευασμένο περιβάλλον. Ένα ακόμα ενδιαφέρον στοιχείο του Minecraft είναι ότι ο κόσμος είναι απείρων διαστάσεων! Κατασκευάζεται προοδευτικά όσο ο χρήστης κινείται προς κάποια νέα διεύθυνση. Το παιχνίδι μπορεί να παιχτεί αυτόνομα από έναν παίκτη ή και σε πολυχρηστικό κόσμο. Παρά το γεγονός ότι η βασισμένη σε κυβάκια εμφάνιση του κόσμου απέχει πολύ από το να χαρακτηρίζεται ρεαλιστική ή έστω εντυπωσιακή, το παιχνίδι έγινε εξαιρετικά δημοφιλές λόγω της ελευθερίας που δίνει στους παίκτες να κατασκευάζουν τους δικούς τους κόσμους και να δώσουν σε αυτούς το νόημα που επιθυμούν. Στην εικόνα 1.7 φαίνεται μια άποψη του κόσμου του Minecraft.



Εικόνα 1.7 Ο κόσμος του Minecraft.

Τέλος, μία ακόμα ενδιαφέρουσα τάση σήμερα είναι η εμφάνιση εξειδικευμένου υλικού για φυσικές αλληλεπιδράσεις, το οποίο ενδέχεται να επηρεάσει τις εξελίξεις στους εικονικούς κόσμους τα επόμενα χρόνια. Πράγματι, τα τελευταία χρόνια έχουν κάνει την εμφάνισή τους στην αγορά συσκευές σχεδιασμένες κυρίως για παιχνίδια, οι οποίες αποσκοπούν στον εμπλουτισμό της εμπειρίας του χρήστη με πιο φυσικές κινήσεις και αναπαραστάσεις. Οι συσκευές αυτές προσπαθούν να πετύχουν τους στόχους εμπύθισης και αλληλεπίδρασης των ακριβών και εξειδικευμένων συσκευών εικονικής πραγματικότητας με πολύ χαμηλότερο κόστος και με μεγαλύτερη συμβατότητα με τις σύγχρονες τεχνολογίες παιχνιδιών. Το Oculus Rift είναι ένα χαρακτηριστικό παράδειγμα τέτοιου προϊόντος. Πρόκειται για ένα κράνος εικονικής πραγματικότητας που αποδίδει στερεοσκοπικά τον κόσμο ενός παιχνιδιού και επιτρέπει την ελεύθερη περιστροφή του κεφαλιού. Ένα άλλο παράδειγμα είναι το Kinect της Microsoft. Κατασκευασμένο αρχικά για την κονσόλα παιχνιδιών X-box, το Kinect αναγνωρίζει την κίνηση του σώματος του χρήστη σε πραγματικό χρόνο και την αξιοποιεί στο περιβάλλον του παιχνιδιού. Τέλος, η συσκευή Leap Motion μπορεί με τη χρήση υπέρυθρων ακτίνων να αναγνωρίζει τη θέση και το σχήμα των χεριών των χρηστών, δίνοντας τα αποτελέσματα που θα έδινε και ένα ακριβό γάντι δεδομένων αλλά με μικρότερη λεπτομέρεια και σε έναν σημαντικά πιο περιορισμένο χώρο δράσης. Ήδη καταγράφονται προσπάθειες αξιοποίησης του Oculus Rift και του Kinect σε περιβάλλοντα όπως το Second Life και το OpenSimulator και αναμένεται μεγαλύτερη υποστήριξη στο μέλλον. Αν το υλικό φυσικής αλληλεπίδρασης διαδοθεί ευρύτερα και γίνει περισσότερο εύχρηστο και προσβάσιμο, ενδέχεται οι εικονικοί κόσμοι να πλησιάσουν ακόμα περισσότερο στο αρχικό όραμα της εικονικής πραγματικότητας.

7 Συμπεράσματα

Στο παρόν κεφάλαιο κάναμε μια εισαγωγική παρουσίαση και επισκόπηση των εικονικών κόσμων και των εφαρμογών τους, ξεκινώντας από το «όραμα» της εικονικής πραγματικότητας και καταλήγοντας σε σύγχρονους δημοφιλείς πολυχρηστικούς κόσμους. Αυτό που προκύπτει ως συμπέρασμα από τη μελέτη της ιστορικής εξέλιξης και των δυνατοτήτων των εικονικών κόσμων είναι ότι πρόκειται για ένα μέσο αρκετά διαφορετικό από τα καθιερωμένα, το οποίο συγκεντρώνει ορισμένες ιδιαίτερες προσφερόμενες λειτουργίες, με σημαντικότερη την τοποθέτηση των επισκεπτών σε ένα ζωντανό και εξελισσόμενο φανταστικό περιβάλλον. Αυτές ακριβώς οι ιδιαιτερότητες του μέσου δημιουργούν πολλές προοπτικές ως προς την αξιοποίησή του σε ένα εύρος περιοχών εφαρμογής που μπορεί να περιλαμβάνουν δραστηριότητες επικοινωνίας, συνεργασίας, εκπαίδευσης ή και ψυχαγωγίας. Ακόμα πιο ενδιαφέρον είναι το γεγονός ότι οι εφαρμογές αυτές θα μπορούσαν να μην είναι απλά μεμονωμένοι, ανεξάρτητοι κόσμοι, αλλά να αποτελούν επιμέρους περιοχές μιας ενιαίας πλατφόρμας εικονικών κόσμων, όπως συμβαίνει σε κάποιο βαθμό σήμερα με το Second Life. Αυτό θα σήμαινε ότι οι χρήστες θα είχαν μία μοναδική ενσάρκωση μέσω της οποίας θα μπορούσαν να συμμετάσχουν σε διάφορων ειδών δραστηριότητες, να δημιουργήσουν κοινότητες με κοινά ενδιαφέροντα και να συνδιαμορφώσουν νέους χώρους ενδιαφέροντος.

Παρά τις διαφαινόμενες δυνατότητες του μέσου, η σημερινή κατάσταση των εικονικών κόσμων δείχνει ότι σε αρκετές περιοχές εφαρμογής δεν έχουν αξιοποιηθεί στον βαθμό που θα μπορούσαν. Ενώ στον χώρο των παιχνιδιών οι εικονικοί κόσμοι ήταν και παραμένουν δημοφιλείς, σε άλλες εφαρμογές υπολείπονται σημαντικά στη χρήση σε σχέση με άλλα μέσα. Αυτό ενδέχεται να οφείλεται στην έλλειψη ειδικά σχεδιασμένων χώρων, εργαλείων και βοηθημάτων τα οποία θα μπορούσαν να υποστηρίξουν τις εξειδικευμένες ανάγκες των χρηστών. Παράλληλα, η κατασκευή περιεχομένου είναι σημαντικά πιο δύσκολη σε σχέση με παραδοσιακές εφαρμογές και ιστοσελίδες και απαιτούνται ομάδες ανάπτυξης με εξειδικευμένες γνώσεις και εμπειρία για την κατάλληλη διαμόρφωση του περιβάλλοντος.

Από την άλλη μεριά, το γεγονός ότι σήμερα είναι διαθέσιμη μια πλατφόρμα κόσμων ανοιχτού λογισμικού, το OpenSimulator, ενδέχεται να βελτιώσει την κατάσταση στο μέλλον. Όσο αυξάνεται η αξιοποίησή του από πανεπιστήμια και εταιρείες τόσο θα μεγαλώνει και θα διαδίδεται η σχετική τεχνογνωσία, και ο πειραματισμός πάνω σε καινοτόμες περιοχές εφαρμογής ενδέχεται να οδηγήσει σε νέα επιτυχημένα παραδείγματα χρήσης και να διευρυνθεί η αξιοποίησή του. Επιπλέον, η εμφάνιση υλικού φυσικής αλληλεπίδρασης χαμηλού κόστους, το οποίο αναμένεται να προσφέρει πιο ζωντανές και πλούσιες εμπειρίες στους επισκέπτες, ενδέχεται να παίξει εξίσου σημαντικό ρόλο στην περαιτέρω εξάπλωση και χρήση των εικονικών κόσμων.

Αναφορές

- Aylett, R. & Louchart, S. (2003). Towards a narrative theory of virtual reality. *Virtual Reality*, 7(1), 2–9.
- Bainbridge, W. S. (2007). The scientific research potential of virtual worlds. *Science (New York, N.Y.)*, 317(5837), 472–6.
- Bartle, R. (2004). *Designing virtual worlds*. New Riders.
- Becerra, E. & Stutts, M. (2008). Ugly duckling by day, super model by night: The influence of body image on the use of virtual worlds. *Journal For Virtual Worlds Research*, 1(2), 1–19.
- Brooks, F. P. (1999). What 's Real About Virtual Reality ? *Computer Graphics and Applications, IEEE*, 19(6), 16–27.

- Bulu, S. T. (2012). Place presence, social presence, co-presence, and satisfaction in virtual worlds. *Computers & Education*, 58(1), 154–161.
- Burdea, G. & Coiffet, P. (1994). *Virtual Reality Technology*. IEEE Press.
- Castronova, E. (2001). *Virtual worlds: A first-hand account of market and society on the cyberian frontier*. CESifo Working Paper.
- Chaturvedi, A. R., Dolk, D. L. & Drnevich, P. L. (2011). Design principles for virtual worlds. *MIS Quarterly*, 35(3), 673–684.
- Churchill, E. & Snowden, D. (1998). Collaborative Virtual Environments: an introductory review of issues and systems. *Virtual Reality*, 3(1), 3–15.
- Curtis, P. & Nichols, D. (1994). MUDs grow up: social virtual reality in the real world. *Proceedings of COMPCON '94*, 1–6.
- Davis, A., Murphy, J., Owens, D., Khazanchi, D. & Zigurs, I. (2009). Avatars, people, and virtual worlds: Foundations for research in metaverses. *Journal of the Association for Information Systems*, 10(2), 90–117.
- Duncan, I., Miller, A. & Jiang, S. (2012). A taxonomy of virtual worlds usage in education. *British Journal of Educational Technology*, 43(6), 949–964.
- Eisenbeiss, M., Blechschmidt, B., Backhaus, K. & Freund, P. A. (2012). “The (Real) World Is Not Enough:” Motivational Drivers and User Behavior in Virtual Worlds. *Journal of Interactive Marketing*, 26(1), 4–20.
- Gorini, A., Gaggioli, A., Vigna, C. & Riva, G. (2008). A second life for eHealth: Prospects for the use of 3-D virtual worlds in clinical psychology. *Journal of Medical Internet Research*, 10(3).
- Kaplan, A. M. & Haenlein, M. (2009). The fairyland of Second Life: Virtual social worlds and how to use them. *Business Horizons*, 52(6), 563–572.
- Koutsabasis, P. & Vosinakis, S. (2012). Rethinking HCI Education for Design: Problem-Based Learning and Virtual Worlds at an HCI Design Studio. *International Journal of Human-Computer Interaction*, 28(8), 485–499.
- Koutsabasis, P., Vosinakis, S., Malisova, K. & Paparounas, N. (2012). On the value of Virtual Worlds for collaborative design. *Design Studies*, 33(4), 357–390.
- Messinger, P. R., Stroulia, E., Lyons, K., Bone, M., Niu, R. H., Smirnov, K. & Perelgut, S. (2009). Virtual worlds — past, present, and future: New directions in social computing. *Decision Support Systems*, 47(3), 204–228.
- Mura, G. (2012). The Advanced Open Metaplastic Platform for Cyber Art. In *Transactions on Computational Science XVI* (pp. 179–190). Springer.
- Schuemie, M. & Liebert, M. A. (2001). Research on presence in virtual reality: A survey. *CyberPsychology & Behavior*, 4(2), 183–201.
- Sequeira, L. M. & Morgado, L. (2013). Virtual Archaeology in Second Life and OpenSimulator. *Virtual World Research*, 6(1).

- Slater, M. (2009). Place illusion and plausibility can lead to realistic behaviour in immersive virtual environments. *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences*, 364(1535), 3549–3557.
- Stephenson, N. (2003). Snow crash. Spectra.
- Steuer, J. (1992). Defining Virtual Reality: Dimensions Determining Telepresence. *Journal of Communication*, 42(4), 73–93.
- Sutherland, I. E. (1965). The ultimate display. In *Proceedings of the IFIP '65*, 2, 506-508.
- Taylor, T. L. (2002). Chapter 3 Living Digitally : Embodiment in Virtual Worlds. *The Social Life of Avatars Presence and Interaction in Shared Virtual Environments*, (3), 40–62.
- Turkle, S. (1994). Constructions and reconstructions of self in virtual reality: Playing in the MUDs. *Mind, Culture, and Activity*, 1(3), 158–167.
- Wachowski, A. & Wachowski, L. (1999). The matrix. USA: Warner Bros.
- Zyda, M. (2005). From visual simulation to virtual reality to games. *Computer*, 38(9), 25–32.

Σύνδεσμοι

Second Life: www.secondlife.org

OpenSimulator: www.opensimulator.org

World of Warcraft: www.warcraft.com

Minecraft: minecraft.net

Περαιτέρω Μελέτη

Μπορείτε να δείτε μια παρουσίαση της ιστορίας των εικονικών κόσμων εδώ: www.vwtimeline.com

Περισσότερα στοιχεία για τους εικονικούς κόσμους παιχνιδιού θα βρείτε στο βιβλίο του Richard Bartle:

Bartle, R. (2004). *Designing virtual worlds*, New Riders

Ένα καλό εισαγωγικό βιβλίο για την εικονική πραγματικότητα είναι το:

Vince, J. (2004). *Introduction to Virtual Reality*, Springer.

Πολύ πιο πλούσια στοιχεία σχετικά με την εικονική πραγματικότητα και τις δυνατότητές της θα βρείτε στο βιβλίο των Sherman και Craig:

Sherman, W. & Craig, A. (2003). *Understanding Virtual Reality*. Morgan Kaufmann.

Ερωτήσεις Κατανόησης

1. Περιγράψτε τα βασικά χαρακτηριστικά των εικονικών κόσμων.
2. Πώς θα μπορούσαν οι εικονικοί κόσμοι να υποστηρίξουν δραστηριότητες συνεργατικής σχεδίασης; Ποια μπορεί να είναι τα πλεονεκτήματα χρήσης τους σε σχέση με άλλα μέσα;
3. Ποιες είναι οι βασικές περιοχές εφαρμογής των εικονικών κόσμων; Σε ποιες από αυτές φαίνεται κατά τη γνώμη σας να αποτελούν οι εικονικοί κόσμοι προτιμότερη επιλογή σε σχέση με εναλλακτικές υλοποιήσεις;
4. Για ποιους λόγους τα συστήματα εικονικής πραγματικότητας δεν αξιοποιούνται σήμερα σε ικανοποιητικό βαθμό;
5. Ποιες είναι οι βασικές διαφορές μιας εφαρμογής εικονικής πραγματικότητας από τους εικονικούς κόσμους στη σημερινή τους μορφή;
6. Εξηγήστε με δικά σας λόγια τη συμβολή του κράνους εικονικής πραγματικότητας και των γαντιών δεδομένων στην εμπειρία του χρήστη.
7. Ποια από τα στοιχεία των σημερινών εικονικών κόσμων μπορούσε κάποιος να βρει στους κόσμους κειμένου της δεκαετίας του '80;
8. Ποιοι είναι κατά τη γνώμη σας οι λόγοι για τους οποίους μπορεί να προτιμήσει κάποιος τη δημιουργία ενός εικονικού κόσμου στο OpenSimulator έναντι του Second Life;
9. Σε ποιες από τις περιοχές εφαρμογής των εικονικών κόσμων θα μπορούσε το σύγχρονο εξειδικευμένο υλικό φυσικών αλληλεπιδράσεων να συνεισφέρει σημαντικά;

Ασκήσεις

1. Δημιουργήστε λογαριασμό στο Second Life και επισκεφτείτε δημοφιλείς περιοχές με αντικείμενο την εκπαίδευση και τον πολιτισμό. Για καθεμία από αυτές δοκιμάστε να εξερευνήσετε τον χώρο και να χρησιμοποιήσετε τις υπηρεσίες που προσφέρονται. Περιγράψτε την εμπειρία σας, τα θετικά στοιχεία που εντοπίσατε και τις ελλείψεις.
2. Συνδεθείτε σε κάποιον δημοφιλή κόσμο παιχνιδιού (μπορείτε να βρείτε και κόσμους χωρίς κόστος συνδρομής) και συγκρίνετε σε σχέση με το Second Life:
 - α) την ποιότητα απεικόνισης,
 - β) το μοντέλο πλοήγησης και αλληλεπίδρασης με το περιβάλλον και
 - γ) τις προσφερόμενες δυνατότητες επικοινωνίας και συνεργασίας.

Κεφάλαιο 2: Απεικόνιση

Σύνοψη

Αντικείμενο του κεφαλαίου είναι η διαδικασία οπτικής απεικόνισης του εικονικού κόσμου, τόσο από την τεχνολογική όσο και από τη σχεδιαστική σκοπιά. Γίνεται μια εισαγωγική επισκόπηση στους αλγορίθμους και στις τεχνικές από τον χώρο των γραφικών που αξιοποιούνται για τη μοντελοποίηση και απεικόνιση του περιβάλλοντος. Παρουσιάζονται τα κατάλληλα μοντέλα αναπαράστασης γεωμετρίας για τις διάφορες κατηγορίες περιεχομένου, οι γεωμετρικοί μετασχηματισμοί, οι υφές και τα υλικά, οι διάφορες τεχνικές φωτισμού και φωτοσκίασης, η τοποθέτηση της κάμερας και η προοπτική προβολή καθώς και άλλα εξειδικευμένα θέματα μοντελοποίησης και οπτικοποίησης. Στη συνέχεια, παρουσιάζεται ο γράφος σκηνής και η χρήση του σε τρισδιάστατους κόσμους, περιγράφονται βασικές τεχνικές και εργαλεία τρισδιάστατης μοντελοποίησης του περιεχομένου και αναφέρονται θέματα βελτιστοποίησης της γεωμετρίας. Τέλος, γίνεται μια επισκόπηση σχετικού υποστηρικτικού λογισμικού για την κατασκευή τρισδιάστατων σκηνών και την απεικόνισή τους σε πραγματικό χρόνο.

Προαπαιτούμενη γνώση

Απαιτούνται γνώσεις πληροφορικής, καθώς και βασικές γνώσεις μαθηματικών.

1 Τα γραφικά στους εικονικούς κόσμους

Η απεικόνιση είναι κεντρικό στοιχείο της διεπαφής με τους εικονικούς κόσμους. Η πλειοψηφία των σημερινών εφαρμογών λογισμικού στηρίζεται βεβαίως στη γραφική απεικόνιση, όμως στις περισσότερες περιπτώσεις η απεικόνιση των εφαρμογών αυτών περιορίζεται στην τοποθέτηση στοιχείων ελέγχου (components) σε κάποιο παραθυρικό περιβάλλον διεπαφής και στην προβολή πληροφοριών και στοιχείων που σχετίζονται με τα δεδομένα της εφαρμογής. Αντίθετα, στους εικονικούς κόσμους, στα παιχνίδια και γενικότερα στα διαδραστικά τρισδιάστατα περιβάλλοντα, κυρίαρχο στοιχείο είναι η παρουσίαση ενός χώρου πλαισιωμένου με αντικείμενα και χαρακτήρες, ενός περιβάλλοντος μέσα στο οποίο ο χρήστης θα κινηθεί, θα επικοινωνήσει και θα δράσει. Το περιβάλλον αυτό φέρει στοιχεία της πραγματικότητας, είτε μιμούμενο πραγματικούς χώρους και αντικείμενα, είτε παρουσιάζοντας εναλλακτικούς κόσμους, και σε κάθε περίπτωση θα πρέπει να γίνει αντιληπτό ως ένα εν δυνάμει περιβάλλον που περιλαμβάνει διακριτές οντότητες.

Στην περίπτωση λοιπόν των τρισδιάστατων περιβαλλόντων, ο ρόλος της απεικόνισης είναι πολύ πιο κρίσιμος σε σχέση με τις τυπικές παραθυρικές εφαρμογές. Δεν παρουσιάζονται κάποια λίγο ή πολύ γνώριμα στοιχεία διεπαφής (όπως πλήκτρα, φόρμες εισόδου κειμένου, πολυμεσικά στοιχεία κ.ά.), αλλά ένα *περιβάλλον* το οποίο καλείται ο ίδιος ο χρήστης να αντιληφθεί ως τρισδιάστατο χώρο, να διακρίνει και να αναγνωρίσει τα επιμέρους στοιχεία του. Λαμβάνοντας υπόψη και το γεγονός ότι η απεικόνιση του τρισδιάστατου περιβάλλοντος γίνεται σε δύο διαστάσεις, δηλαδή στην οθόνη του υπολογιστή, ο χρήστης θα πρέπει επιπλέον να «αποκωδικοποιήσει» αυτήν τη δισδιάστατη εικόνα ως τρισδιάστατη σκηνή. Χρησιμοποιώντας τις ενδείξεις βάθους που του παρέχονται και αναγνωρίζοντας γνώριμες μορφές χώρων και αντικειμένων, θα πρέπει να μπορεί να αντιληφθεί τις σχετικές αποστάσεις και τα σχετικά μεγέθη των αντικειμένων. Τέλος, η διάδραση του χρήστη με έναν εικονικό κόσμο είναι συνήθως *διαρκής*, διότι αφενός οι ενέργειές του μπορεί να έχουν διάρκεια, όπως π.χ. η πλοήγηση, αφετέρου το περιβάλλον μεταβάλλεται συνεχώς. Αυτό σημαίνει ότι η ίδια η απεικόνιση θα πρέπει να ανανεώνεται διαρκώς, και ο χρήστης θα πρέπει να έχει συνεχώς εστιασμένη την προσοχή του στην απεικόνιση για να μπορεί να αντιληφθεί τις αλλαγές στο περιβάλλον.

Οι εικονικοί κόσμοι είναι ιδιαίτερα απαιτητικοί σε ό,τι αφορά τη γραφική απεικόνιση του περιβάλλοντος, με αποτέλεσμα να εκμεταλλεύονται συνήθως στο έπακρο την υπολογιστική ισχύ και τη μνήμη των συστημάτων στα οποία εκτελούνται. Αυτό οφείλεται σε τρεις βασικές ανάγκες: α) μεγάλος όγκος περιεχομένου, β) ποιοτική απεικόνιση και γ) εκτέλεση σε πραγματικό χρόνο. Η ανάγκη για όγκο δεδομένων προκύπτει από το γεγονός ότι οι εικονικοί κόσμοι καλύπτουν συνήθως ευρείες χωρικές εκτάσεις, ενώ παράλληλα μπορεί να περιλαμβάνουν έναν πολύ μεγάλο αριθμό αντικειμένων με λειτουργικό ή καθαρά αισθητικό ρόλο. Εκτός όμως από την ίδια την παρουσία των αντικειμένων, αναδύεται και η ανάγκη για την ποιοτική απεικόνισή, ώστε να αυξηθεί ο ρεαλισμός του περιβάλλοντος. Οι δύο αυτές απαιτήσεις δεν έχουν άνω όριο. Θέλουμε να μπορούμε να συμπεριλάβουμε στο περιβάλλον μας όσο το δυνατόν περισσότερα αντικείμενα,

με τον μεγαλύτερο δυνατό βαθμό λεπτομέρειας και με τη βέλτιστη απεικόνιση. Ιδανικά, θέλουμε η απεικόνιση του περιβάλλοντος να μοιάζει με εικόνα του πραγματικού κόσμου. Η απαίτηση αυτή δεν προκύπτει μόνο από αισθητικούς λόγους. Αρκετοί ερευνητές καταλήγουν στο συμπέρασμα ότι η ποιότητα απεικόνισης επηρεάζει την αίσθηση της «παρουσίας» του χρήστη σε ένα εικονικό περιβάλλον, άρα και συνολικά την εμπειρία της αλληλεπίδρασής του με αυτό (Bracken & Skalski, 2009· Pausch κ.ά., 1996). Από την άλλη μεριά, οι εικονικοί κόσμοι είναι συστήματα «πραγματικού χρόνου», δηλαδή η απεικόνιση ανανεώνεται δυναμικά με ρυθμό αρκετά μεγάλο ώστε η κίνηση να γίνεται αντιληπτή από τον χρήστη ως συνεχής. Αυτό πρακτικά σημαίνει ότι υπάρχει ένα πολύ περιορισμένο χρονικό όριο, της τάξης λίγων εκατοστών του δευτερολέπτου, εντός του οποίου θα πρέπει να έχει ολοκληρωθεί η διαδικασία της απεικόνισης, χωρίς να διαταραχθεί η συνεχής ροή των εικόνων.

Αυτή ακριβώς η ισχυρή εξάρτηση της απεικόνισης του περιβάλλοντος από την υπολογιστική ισχύ του συστήματος έκανε τις ραγδαίες εξελίξεις στην τεχνολογία των τρισδιάστατων γραφικών να συνεισφέρουν καθοριστικά στην ανάπτυξη και διάδοση των εικονικών κόσμων. Τις τελευταίες δύο δεκαετίες η ποιότητα και οι χρόνοι απεικόνισης και κίνησης τρισδιάστατων σκηνών εκτινάχθηκαν. Αυτό συνέβη χάρη στη διαρκή εξέλιξη του σχετικού λογισμικού και κυρίως του υλικού, με βασική κατευθυντήρια δύναμη τη βιομηχανία των ηλεκτρονικών παιχνιδιών. Στα μέσα της δεκαετίας του '90 διαδόθηκαν και άρχισαν να ενσωματώνονται στους οικιακούς υπολογιστές ειδικοί επεξεργαστές για γραφικά, οι λεγόμενες Μονάδες Επεξεργασίας Γραφικών (Graphics Processing Unit – GPU), οι οποίες εκτελούσαν τις διεργασίες απεικόνισης σε επίπεδο υλικού βελτιώνοντας κατά πολύ τους χρόνους. Αυτές οι εξελίξεις σε συνδυασμό με την εμφάνιση ειδικών βιβλιοθηκών λογισμικού για την υποστήριξη τρισδιάστατων γραφικών έκαναν εφικτή την ανάπτυξη τρισδιάστατων περιβαλλόντων ικανών να αντεπεξέλθουν στις απαιτήσεις για ποιοτικά γραφικά σε πραγματικό χρόνο. Όσο η σχετική τεχνολογία εξελισσόταν, τόσο δινόταν η δυνατότητα στους εικονικούς κόσμους να συμπεριλαμβάνουν μεγαλύτερο αριθμό αντικειμένων, να απεικονίζουν τη γεωμετρία των αντικειμένων με μεγαλύτερη λεπτομέρεια, να έχουν περισσότερο ρεαλιστικές απεικονίσεις και να εντάσσουν πληθώρα ειδικών εφέ στο περιβάλλον.

Τα γραφικά υπολογιστών χρησιμοποιούνται στους εικονικούς κόσμους για να αποδώσουν διάφορα είδη περιεχομένου. Συνήθως περιλαμβάνεται κάποιο σκηνικό που περιβάλλει τις οντότητες του κόσμου, το οποίο σε ρεαλιστικά περιβάλλοντα αποτελείται από το ανάγλυφο του εδάφους και μια απεικόνιση του ουράνιου θόλου ανάλογα με την ώρα της ημέρας και τις καιρικές συνθήκες. Υπάρχουν ακόμα τα αντικείμενα του κόσμου τα οποία μπορεί να έχουν μοντελοποιηθεί με βάση αντίστοιχα αντικείμενα του πραγματικού κόσμου ή να είναι υποθετικά. Επιπλέον, στον κόσμο συνυπάρχουν και οι ενσωματώσεις των χρηστών, οι οποίες είναι συνήθως ανθρωπόμορφες, ενώ σε κάποιες περιπτώσεις εμφανίζονται και χαρακτηριστές ελεγχόμενοι από τον υπολογιστή. Εκτός από τα παραπάνω, σε αρκετές κατηγορίες περιβαλλόντων μπορεί να εμφανίζονται και αφηρημένες γεωμετρικές, όπως διαφανή αντικείμενα, περιγράμματα κ.ά. Οι γεωμετρικές αυτές μπορεί να χρησιμοποιούνται για λόγους επαύξησης του περιβάλλοντος με επιπλέον πληροφορίες ή και να αποτελούν οι ίδιες στοιχεία διεπαφής. Τέλος, σε όλους σχεδόν τους εικονικούς κόσμους εμφανίζονται και δισδιάστατες απεικονίσεις, όπως κείμενα, εικόνες ή και δισδιάστατα στοιχεία ελέγχου (π.χ. πλήκτρα).

Τα είδη απεικονίσεων που αναφέρθηκαν εμφανίζουν μεγάλη διαφοροποίηση. Κάποια στοιχεία είναι δισδιάστατα και κάποια άλλα τρισδιάστατα, κάποια είναι στατικά, δηλαδή η κατάστασή τους παραμένει αμετάβλητη στον χρόνο, κάποια κινούμενα και κάποια προσωρινά. Ορισμένα αντικείμενα είναι άκαμπτα (π.χ. μια καρέκλα), ενώ άλλα υφίστανται διαρκείς παραμορφώσεις στη γεωμετρία τους (π.χ. τα μαλλιά ενός χαρακτήρα), κάποια απεικονίζονται ως συμπαγή ρεαλιστικά αντικείμενα και άλλα ως αφηρημένα σχήματα. Για την αντιμετώπιση των παραπάνω καταστάσεων χρησιμοποιούνται και συνδυάζονται πολλαπλές τεχνικές από τον χώρο των γραφικών με υπολογιστή.

Στη συνέχεια του κεφαλαίου θα μελετήσουμε τις σύγχρονες προσεγγίσεις στη γραφική αναπαράσταση και στην ανάπτυξη τρισδιάστατων σκηνών. Αρχικά θα παρουσιάσουμε την τεχνολογία των 3D γραφικών με υπολογιστή, όπου θα εξεταστούν θέματα μοντελοποίησης, μετασχηματισμών και οπτικής απόδοσης τρισδιάστατων χώρων. Στη συνέχεια θα μελετήσουμε τη διαδικασία ανάπτυξης μιας σκηνής με στόχο την ενσωμάτωσή της σε ένα περιβάλλον εικονικού κόσμου. Τέλος, θα γίνει μια σύντομη επισκόπηση του υποστηρικτικού λογισμικού για τη μοντελοποίηση και ανάπτυξη διαδραστικών τρισδιάστατων περιβαλλόντων, όπου θα αναφέρουμε τις δυνατότητες και τους περιορισμούς της κάθε λύσης.

2 Εισαγωγή στην τεχνολογία των 3Δ γραφικών με υπολογιστή

Ο κλάδος των *Γραφικών με Υπολογιστή (Computer Graphics)* είναι μια περιοχή της επιστήμης υπολογιστών που είναι ενεργή από τη δεκαετία του '60 και ασχολείται με την αξιοποίηση των υπολογιστών για την παραγωγή στατικών και κινούμενων εικόνων (Watt, 1994). Οι αλγόριθμοι, οι τεχνικές και τα εργαλεία που προέκυψαν από τον κλάδο αυτόν βρήκαν εφαρμογή σε πάρα πολλούς χώρους, όπως στην επιστήμη (οπτικοποίηση επιστημονικών δεδομένων), στη διασκέδαση (ψηφιακός κινηματογράφος, παιχνίδια), στον πολιτισμό κ.ά. Λόγω των διαφορετικών απαιτήσεων των πεδίων εφαρμογής, οι προσεγγίσεις στην απεικόνιση σκηνών μοιράστηκαν σε αυτές που δίνουν έμφαση στην ποιότητα του οπτικού αποτελέσματος και στον φωτορεαλισμό (π.χ. για την παραγωγή ψηφιακών εφέ στον κινηματογράφο), και σε αυτές που εστιάζουν στην ταχύτητα και μπορούν να αξιοποιηθούν σε περιβάλλοντα πραγματικού χρόνου, όπως είναι τα παιχνίδια και τα εικονικά περιβάλλοντα. Στη συνέχεια θα παρουσιάσουμε ορισμένες βασικές τεχνικές που χρησιμοποιούνται σε περιβάλλοντα πραγματικού χρόνου για τη μοντελοποίηση, τη φωτοσκίαση και την τελική απεικόνιση των τρισδιάστατων σκηνών.

2.1 Τρισδιάστατα μοντέλα

Η γραφική απεικόνιση μίας τρισδιάστατης σκηνής προαπαιτεί την περιγραφή της γεωμετρίας της μέσω κάποιου μαθηματικού μοντέλου. Ένα μοντέλο είναι μια προσεγγιστική αναπαράσταση ενός υπαρκτού ή υποθετικού αντικειμένου, το οποίο περιγράφει όσο το δυνατόν περισσότερες ιδιότητες αυτού. Επειδή ο κλάδος των γραφικών με υπολογιστή έχει αναπτυχθεί με στόχο την ικανοποίηση αναγκών οπτικοποίησης και συνθετικής κίνησης που προκύπτουν από πολλούς διαφορετικούς χώρους, έχει εμφανιστεί και ένας αριθμός από διαφορετικά είδη μοντέλων αναπαράστασης γεωμετρίας, ανάλογα με τις ανάγκες του κάθε χώρου. Σε κάποιες περιπτώσεις (π.χ. στον κινηματογράφο), λαμβάνεται περισσότερο υπόψη η ποιότητα απεικόνισης, σε άλλες (π.χ. μηχανολογία, αρχιτεκτονική κ.ά.), η ακρίβεια στην περιγραφή της γεωμετρίας και η εύκολη επεξεργασία. Στα παιχνίδια, στους εικονικούς κόσμους και γενικότερα σε εφαρμογές γραφικών πραγματικού χρόνου, η ταχύτητα απεικόνισης είναι ένας κρίσιμος παράγοντας επιτυχίας.

Οι δύο βασικές κατηγορίες μοντέλων αναπαράστασης είναι οι *αναπαραστάσεις επιφάνειας* και οι *αναπαραστάσεις όγκου*. Στην πρώτη περίπτωση περιγράφεται μόνο η γεωμετρία της επιφάνειας του αντικειμένου και δεν υπάρχει καμία πληροφορία σχετικά με το εσωτερικό του. Για παράδειγμα, θα μπορούσε να χρησιμοποιηθεί μια τέτοια μορφής αναπαράσταση για να απεικονίσει ένα μήλο. Το αποτέλεσμα οπτικά θα μπορούσε να είναι πολύ ρεαλιστικό, και πράγματι να μοιάζει πάρα πολύ με ένα πραγματικό μήλο. Όμως, αν προσπαθούσε να δει κάποιος μέσα σε αυτό, θα έβλεπε ότι είναι «άδειο», δηλαδή ότι υπάρχει απλά η εξωτερική επιφάνεια και μέσα είναι κενό. Αυτή η προσέγγιση λοιπόν έχει νόημα όταν μας ενδιαφέρει να απεικονίσουμε «συμπαγή» αντικείμενα, τα οποία κατά τη διάρκεια της εφαρμογής θα παρατηρούμε μόνο εξωτερικά. Η ποσότητα της πληροφορίας που απαιτείται για να περιγράψει την εξωτερική επιφάνεια ενός αντικειμένου είναι προφανώς πολύ μικρότερη σε σχέση με την περιγραφή όλου του όγκου του. Υπάρχουν όμως και περιπτώσεις στις οποίες μπορεί πράγματι να θέλουμε να δούμε μέσα στα αντικείμενα, επομένως οι αναπαραστάσεις επιφάνειας δεν επαρκούν. Ένα τέτοιο παράδειγμα είναι η οπτικοποίηση επιστημονικών δεδομένων όπου σε έναν τρισδιάστατο χώρο θέλουμε να δημιουργήσουμε χρωματική απεικόνιση της τιμής μιας μεταβλητής (π.χ. θερμοκρασία, επίπεδα μόλυνσης κ.λπ.) Αντίστοιχα, αν θέλουμε να παρουσιάσουμε το ανθρώπινο σώμα σε διάφορα επίπεδα βάθους (π.χ. δέρμα, μύες, όργανα, σκελετός) οι αναπαραστάσεις επιφάνειας δεν αρκούν.

Σε ό,τι αφορά τα μοντέλα αναπαράστασης επιφάνειας υπάρχουν δύο βασικές προσεγγίσεις για την περιγραφή της γεωμετρίας των αντικειμένων: η *μαθηματική περιγραφή* και η *χρήση πολυγώνων*. Στην πρώτη περίπτωση, η επιφάνεια περιγράφεται βάσει κάποιων εξισώσεων, των οποίων τις παραμέτρους ρυθμίζει ο σχεδιαστής. Παραδείγματα τέτοιων μοντέλων είναι τα βασικά γεωμετρικά στερεά (π.χ. σφαίρα, πυραμίδα, παραλληλεπίπεδο κ.λπ.) που μπορούν να κατασκευαστούν από μαθηματικές εξισώσεις και ο συνδυασμός τους να οδηγήσει σε σύνθετα αντικείμενα, οι επιφάνειες NURBS που αναπαριστούν καμπύλες επιφάνειες μέσω ορισμού κατάλληλων σημείων ελέγχου κ.ά. Στη δεύτερη περίπτωση, της αναπαράστασης μέσω πολυγώνων, η επιφάνεια περιγράφεται από ένα σύνολο τρισδιάστατων σημείων και ένα σύνολο πολυγώνων με κορυφές τα σημεία αυτά. Τα αντίστοιχα μοντέλα περιγραφής ονομάζονται *πολυγωνικά μοντέλα*. Το πλεονέκτημα της μαθηματικής περιγραφής είναι ότι οι αναπαραστάσεις των αντικειμένων έχουν μεγάλη ακρίβεια στη γεωμετρία τους, επιτρέπουν ακριβείς υπολογισμούς μεγεθών που αφορούν το αντικείμενο, ενώ λόγω της δυνατότητας παραμετροποίησής τους μπορούν να χρησιμοποιηθούν ως βάση για τη δημιουργία πολλαπλών παραλλαγών της

ίδιας κατηγορίας αντικειμένων (π.χ. ανθρώπινα μοντέλα με διάφορα ύψη και βάρη). Από την άλλη, τα αντικείμενα που μπορούν να δημιουργηθούν με τα μοντέλα αυτά είναι περιορισμένα· δεν μπορούν να υποστηρίξουν τυχαίες γεωμετρίες. Στην περίπτωση αυτήν είναι προτιμότερα τα πολυγωνικά μοντέλα, που δίνουν σαφώς μεγαλύτερη ευελιξία και δεν έχουν κανέναν περιορισμό ως προς τη μορφή της επιφάνειας. Το μειονέκτημα είναι ότι τα μοντέλα αυτά δεν μπορούν να περιγράψουν πλήρως καμπύλες επιφάνειες αλλά μόνο προσεγγίσεις αυτών, αν ο αριθμός των πολυγώνων είναι αρκετά μεγάλος.

Τα μοντέλα αναπαράστασης όγκου βασίζονται στη χρήση *στοιχειωδών σωματιδίων*, συνήθως παραλληλεπίπεδων, τα οποία τοποθετούνται σε ένα τρισδιάστατο πλέγμα. Τα σωματίδια αυτά, που ονομάζονται *voxels*, χρησιμοποιούνται με έναν τρόπο αντίστοιχο με αυτό των pixels στις δισδιάστατες εικόνες. Όπως λοιπόν μία εικόνα τύπου bitmap είναι στην ουσία ένα πλέγμα από pixels διαφόρων χρωμάτων, έτσι και μια σκηνή που έχει μοντελοποιηθεί με μοντέλο αναπαράστασης όγκου αποτελείται από πολλά χρωματισμένα voxels τοποθετημένα σε προκαθορισμένες θέσεις του πλέγματος που περιγράφουν το αντικείμενο. Όπως μπορείτε να φανταστείτε, η διακριτοποίηση του χώρου σε πολλά μικρά αντικείμενα επιδρά αρνητικά στην ποιότητα απεικόνισης και για να αναπαρασταθούν ικανοποιητικά καμπύλες επιφάνειες θα χρειαστεί να είναι υπερβολικά μικρό το μέγεθος των επιμέρους voxels, άρα και υπερβολικά μεγάλος ο αριθμός τους.

Ο πίνακας 2.1 παρουσιάζει συνοπτικά τα μοντέλα αναπαράστασης που αναφέρθηκαν και τα πλεονεκτήματα και μειονεκτήματά τους.

Μοντέλο Αναπαράστασης	Πλεονεκτήματα	Μειονεκτήματα
αναπαράσταση επιφάνειας με μαθηματική περιγραφή	μεγάλη ακρίβεια, δυνατότητες παραμετροποίησης	δεν μπορούν να αναπαρασταθούν αυθαίρετες γεωμετρίες
αναπαράσταση επιφάνειας με πολύγωνα	απλό μοντέλο, η επιφάνεια μπορεί να πάρει οποιαδήποτε μορφή	οι καμπύλες αποδίδονται προσεγγιστικά
αναπαράσταση όγκου	τα αντικείμενα έχουν περιεχόμενο	όγκος δεδομένων, όχι καλής ποιότητας απεικόνιση

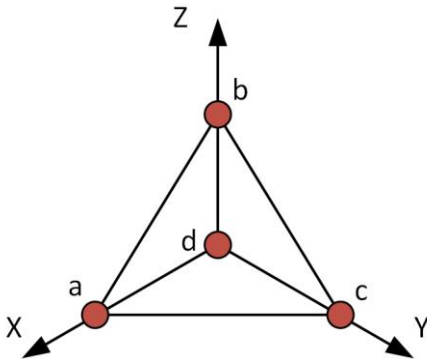
Πίνακας 2.1 Βασικά μοντέλα αναπαράστασης γεωμετρίας, πλεονεκτήματα και μειονεκτήματα.

Οι εικονικοί κόσμοι, τα παιχνίδια και γενικότερα οι εφαρμογές γραφικών σε πραγματικό χρόνο βασίζονται στην πλειοψηφία τους στο πολυγωνικό μοντέλο. Ο λόγος είναι ότι το μοντέλο αυτό έχει επικρατήσει σήμερα λόγω της απλότητάς του αλλά και της δυνατότητας παραγωγής πολύ καλής ποιότητας εικόνας αν ο αριθμός των πολυγώνων είναι αρκετά μεγάλος. Το πολυγωνικό μοντέλο υποστηρίζεται σήμερα από τους επεξεργαστές γραφικών των σταθερών και φορητών υπολογιστικών συστημάτων, οι οποίοι μπορούν να δώσουν πολύ γρήγορα και ποιοτικά αποτελέσματα. Έτσι, ακόμα και σε περιπτώσεις εφαρμογών στις οποίες η περιγραφή της γεωμετρίας γίνεται με άλλα μοντέλα, όπως για παράδειγμα στον εικονικό κόσμο MineCraft που χρησιμοποιούνται voxels και στο Second Life που οι γεωμετρίες περιγράφονται ως συνδυασμός βασικών στερεών, τα μοντέλα μετατρέπονται σε πολυγωνικά πριν την τελική απεικόνιση.

Ας δούμε ένα παράδειγμα γεωμετρίας και τον τρόπο περιγραφής της μέσω του πολυγωνικού μοντέλου. Στην εικόνα 2.1 παρουσιάζεται ένα τρισδιάστατο αντικείμενο που κατασκευάστηκε από τέσσερα τρίγωνα: μια βάση, δύο κάθετες πλευρές και μια πλευρά που κοιτάζει προς εμάς. Για την ακρίβεια, το αντικείμενο αυτό είναι ένα κανονικό τετράεδρο, δηλαδή μια τριγωνική πυραμίδα αποτελούμενη από τέσσερα ίδια ισόπλευρα τρίγωνα. Έστω ότι ονομάζουμε a , b , c και d τις κορυφές του. Τότε μπορούμε να προσδιορίσουμε καθένα από τα τρίγωνα του τετραέδρου από τις κορυφές του. Κατά συνέπεια η βάση είναι η acd , οι δύο κάθετες πλευρές οι cbd και adb , και η πλευρά που κοιτάζει προς εμάς η abc . Σύμφωνα με το πολυγωνικό μοντέλο, το αντικείμενο του παραδείγματός μας θα μπορούσε να περιγραφεί με τη χρήση δύο πινάκων (πίνακας 2.2). Στον πρώτο, τον πίνακα κορυφών, δηλώνονται οι συντεταγμένες των κορυφών του αντικειμένου, ενώ στον δεύτερο, τον πίνακα τριγώνων, περιγράφονται τα επιμέρους τρίγωνα του αντικειμένου δηλώνοντας τα ονόματα των κορυφών που τα σχηματίζουν.

Η μοντελοποίηση ενός τρισδιάστατου αντικειμένου, όπως το τετράεδρο του παραδείγματός μας, προϋποθέτει τη χρήση ενός *καρτεσιανού συστήματος συντεταγμένων*. Παρά το γεγονός ότι η τελική προβολή γίνεται σε δύο διαστάσεις, δηλαδή στην οθόνη του υπολογιστή μας, το αντικείμενο περιγράφεται στις τρεις διαστάσεις, και για τον λόγο αυτόν απαιτούνται τρεις μετρήσεις για καθμία από τις κορυφές του. Το καρτεσιανό σύστημα συντεταγμένων αποτελείται από ένα σύνολο τρισδιάστατων αξόνων X , Y και Z , όπου ο κάθε άξονας είναι κάθετος προς τους άλλους δύο. Με βάση το σύστημα αυτό μπορεί να προσδιοριστεί μοναδικά

οποιοδήποτε σημείο P με συντεταγμένες x, y, z στον τρισδιάστατο χώρο σε σχέση με ένα σημείο αναφοράς O το οποίο αποτελεί την αρχή των αξόνων. Κατά σύμβαση τα καρτεσιανά συστήματα συντεταγμένων είναι «δεξιόστροφα», δηλαδή η φορά των τριών αξόνων καθορίζεται από τα τρία πρώτα δάκτυλα του δεξιού χεριού: αντίχειρας για τον X , δείκτης για τον Y και μέσος για τον Z . Εδώ αξίζει να σημειωθεί ότι σε αρκετά τρισδιάστατα περιβάλλοντα ο άξονας Z χρησιμοποιείται για το ύψος ενώ σε άλλα ο Y . Για παράδειγμα, ο εικονικός κόσμος του Second Life ανήκει στην πρώτη περίπτωση, ενώ η μηχανή παιχνιδιών Unity στη δεύτερη. Στο παρόν βιβλίο θα χρησιμοποιούμε το σύστημα συντεταγμένων που απεικονίζεται στην εικόνα 2.1 με το Z για ύψος.



Εικόνα 2.1 Απεικόνιση τριγωνικής πυραμίδας.

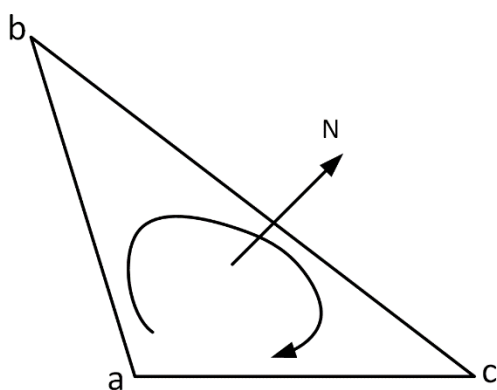
Πίνακας Κορυφών				Πίνακας Τριγώνων			
Όνομα	x	y	z	Κορυφές			
a	1	0	0	abc	a	b	c
b	0	0	1	cbd	c	b	d
c	0	1	0	adb	a	d	b
d	0	0	0	acd	a	c	d

Πίνακας 2.2 Πίνακας κορυφών (αριστερά) και τριγώνων (δεξιά) για την αναπαράσταση της πυραμίδας της εικόνας 2.1.

Η περιγραφή των πολυγώνων του αντικειμένου θα μπορούσε ασφαλώς να γίνει και με έναν πίνακα, δηλώνοντας απλά τις συντεταγμένες των κορυφών του κάθε τριγώνου, γεγονός που προκαλεί το εύλογο ερώτημα γιατί χρειάζονται δύο ξεχωριστοί πίνακες. Η απάντηση βρίσκεται στο ότι στην περίπτωση των δύο πινάκων που περιγράψαμε παραπάνω οι συντεταγμένες της κάθε κορυφής δηλώνονται μόνο μία φορά και όχι τόσες όσα τα πολύγωνα στα οποία ανήκουν. Αυτό το χαρακτηριστικό μειώνει τον απαιτούμενο χώρο αποθήκευσης του μοντέλου, κέρδος που δεν είναι καθόλου αμελητέο, αν σκεφτούμε ότι οι σημερινές τρισδιάστατες σκηνές περιλαμβάνουν εκατοντάδες χιλιάδες πολύγωνα και ότι αναπτύσσονται εφαρμογές γραφικών πραγματικού χρόνου σε υπολογιστικά συστήματα με μειωμένες δυνατότητες αποθήκευσης, όπως ταμπλέτες και κινητά τηλέφωνα. Το πιο σημαντικό όμως είναι ότι αποφεύγονται προβλήματα ακεραιότητας στην περίπτωση που αλλάξει η θέση μιας κορυφής. Αν, για παράδειγμα, αλλάξουμε τις συντεταγμένες της κορυφής a στο παράδειγμά μας και παραμείνουμε στην περιγραφή με πίνακα κορυφών και πίνακα τριγώνων, τότε το αποτέλεσμα θα παραμείνει τετράεδρο (αν και όχι κανονικό), καθώς όλα τα τρίγωνα θα σχεδιαστούν με βάση τη νέα θέση της κορυφής. Στη γενική περίπτωση το αντικείμενο θα παραμείνει συμπαγές. Αν όμως είχαμε έναν μοναδικό πίνακα με συντεταγμένες κορυφών και αλλάξαμε τιμή σε μία από αυτές, θα έπρεπε να κάνουμε το ίδιο και σε όλα τα υπόλοιπα τρίγωνα που περιλαμβάνουν την ίδια κορυφή, αλλιώς το σχήμα θα έπαυε να είναι συμπαγές και θα δημιουργούσε θραύσμα.

Μπορείτε να παρατηρήσετε στο παράδειγμά μας ότι περιγράψαμε τα τρίγωνα διατρέχοντας τις κορυφές με τη φορά των δεικτών του ρολογιού όπως θα τις έβλεπε κάποιος από την εξωτερική πλευρά του αντικειμένου. Υπάρχει συγκεκριμένος λόγος γι' αυτό. Ενώ στον πραγματικό κόσμο όλες οι επιφάνειες, όσο λεπτές κι αν είναι, έχουν, προφανώς, δύο όψεις, στα γραφικά με υπολογιστή μπορούμε να παρακάμψουμε αυτό τον περιορισμό και να είναι ορατή μόνο η μία από τις δύο όψεις. Το πολυγωνικό μοντέλο είναι ένα μοντέλο αναπαράστασης επιφάνειας και, όπως αναφέραμε παραπάνω, χρησιμοποιείται για να περιγράψει συμπαγή αντικείμενα που θα

τα παρατηρούμε μόνο εξωτερικά. Επομένως, από την επιφάνεια ενός αντικειμένου παρατηρούμε πάντα μία όψη, την «εξωτερική» όψη. Εφόσον όμως την εσωτερική όψη δεν θα την παρατηρήσουμε ποτέ (υποθέτοντας ότι η εφαρμογή μας δεν θα μας επιτρέψει να δούμε μέσα στα αντικείμενα), δεν υπάρχει και λόγος να συνυπολογίζεται κατά την απεικόνιση της σκηνής. Μία τέτοια διάκριση λοιπόν μεταξύ «εξωτερικής» (που ζωγραφίζεται) και «εσωτερικής» (που δεν ζωγραφίζεται) όψης των επιφανειών μειώνει τον χρόνο απεικόνισης περίπου στο μισό, αφού οι μισές επιφάνειες δεν θα ζωγραφίζονται καν, και άρα αυξάνει σημαντικά την απόδοση του συστήματος απεικόνισης. Για τον λόγο αυτόν συνηθίζεται στις γεωμετρικές που περιγράφονται με το πολυγωνικό μοντέλο να τηρείται κάποια σύμβαση για τη διάκριση μεταξύ εσωτερικής και εξωτερικής όψης, όπως το να θεωρείται εξωτερική η πλευρά στην οποία η σειρά των κορυφών ακολουθεί τη φορά των δεικτών του ρολογιού. Από τις κορυφές του πολυγώνου υπολογίζεται και το κανονικό διάνυσμα της επιφάνειας, δηλαδή το μοναδιαίο διάνυσμα που είναι κάθετο στο επίπεδο του πολυγώνου και έχει φορά προς την εξωτερική όψη. Το διάνυσμα αυτό χρησιμοποιείται, όπως θα δούμε αργότερα, για τη φωτοσκίαση των αντικειμένων. Η εικόνα 2.2 απεικονίζει την εξωτερική όψη και το κανονικό διάνυσμα ενός τριγώνου.



Εικόνα 2.2 Η εξωτερική όψη του τριγώνου abc είναι αυτή στην οποία οι κορυφές a , b και c διατρέχονται με τη φορά των δεικτών του ρολογιού. Το κανονικό διάνυσμα N είναι κάθετο στο επίπεδο του τριγώνου και δείχνει προς την εξωτερική όψη.

Αν και στο πολυγωνικό μοντέλο τα τρισδιάστατα αντικείμενα αναπαρίστανται ως συλλογές από πολύγωνα, στην πράξη οι περισσότερες υλοποιήσεις του μοντέλου αυτού περιορίζονται στα *τρίγωνα*. Επειδή ένα επίπεδο ορίζεται από τρία σημεία κατ' ελάχιστον, τα τρίγωνα έχουν το χαρακτηριστικό ότι οι κορυφές τους ανήκουν πάντα στο ίδιο επίπεδο. Αυτό δεν ισχύει κατ' ανάγκη και στα τετράπλευρα ή στα πολύγωνα γενικότερα, όπου είναι ευθύνη του σχεδιαστή να διασφαλίσει ότι οι κορυφές τους είναι συνεπίπεδες, άρα και το πολύγωνο επίπεδο. Αν αυτό δεν συμβαίνει, τα αποτελέσματα της οπτικής απεικόνισης θα είναι λανθασμένα. Για να αποφευχθούν τέτοια προβλήματα, αλλά και για ταχύτερες διαδικασίες οπτικής απεικόνισης, προτιμώνται τελικά τα τρίγωνα. Εξάλλου, υπάρχουν αποτελεσματικοί *αλγόριθμοι τριγωνοποίησης* που μπορούν να μετατρέψουν ένα οποιοδήποτε πολύγωνο σε σύνολο από τρίγωνα. Κατά συνέπεια, ακόμα κι αν ο σχεδιαστής κατασκευάσει ένα αντικείμενο χρησιμοποιώντας πολύγωνα, με κάποια προεπεξεργασία μπορούν αυτά να μετατραπούν σε τρίγωνα.

Η μεγάλη πλειοψηφία των τρισδιάστατων αντικειμένων που συναντάμε στους εικονικούς κόσμους έχει κατασκευαστεί με βάση το μοντέλο που παρουσιάσαμε παραπάνω. Το παράδειγμά μας βεβαίως είναι απλοποιημένο για ευκολότερη κατανόηση. Αυτό που συνήθως συμβαίνει σε επίπεδο δομής δεδομένων είναι ότι ορίζεται μία δεικτοδοτημένη συλλογή (π.χ. πίνακας ή λίστα) από κορυφές και μία ακόμη συλλογή από τρίγωνα για τα οποία δηλώνονται οι δείκτες (θέση στην περίπτωση πίνακα ή δείκτης σε θέση μνήμης στην περίπτωση λίστας) στις αντίστοιχες κορυφές που τα ορίζουν. Τα αντικείμενα αυτά, που ονομάζονται και *πολυγωνικά πλέγματα* (*polygonal meshes*) ή απλά *πλέγματα* (*meshes*), συχνά αποτελούνται από πολύ μεγάλο αριθμό τριγώνων (ακόμα και αρκετές χιλιάδες) και το οπτικό αποτέλεσμα με τις σημερινές δυνατότητες των επεξεργαστών γραφικών μπορεί να είναι ιδιαίτερα ποιοτικό. Παρά το γεγονός ότι δεν περιλαμβάνουν καμπύλες επιφάνειες, καθώς οι επιφάνειες περιγράφονται από πολλά επίπεδα τρίγωνα, ο μεγάλος αριθμός τριγώνων και η χρήση κατάλληλων μοντέλων φωτοσκίασης μπορεί να οδηγήσει σε απεικονίσεις οι οποίες δίνουν οπτικά την αίσθηση της καμπυλότητας.

Όπως μπορούμε εύκολα να αντιληφθούμε, ο αριθμός των τριγώνων που περιγράφουν μια σκηνή είναι ευθέως ανάλογος της ποιότητας απεικόνισης. Από την άλλη μεριά όμως είναι και αντιστρόφως ανάλογος της ταχύτητας απεικόνισης, καθώς απαιτείται περισσότερος χρόνος για την οπτική απόδοση (rendering) του κάθε μεμονωμένου πλάνου. Κατά συνέπεια, εφόσον οι εικονικοί κόσμοι αποτελούν κι αυτοί εφαρμογές γραφικών πραγματικού χρόνου, υπάρχει ένας πολύ ισχυρός περιορισμός για τη διατήρηση του ρυθμού ανανέωσης της σκηνής σε αποδεκτά επίπεδα. Αν, για παράδειγμα, το υπολογιστικό σύστημα δεν μπορεί να αποδώσει τουλάχιστον τριάντα πλάνα το δευτερόλεπτο (frames per second / fps), τότε παύει να γίνεται αντιληπτή ως ομαλή και συνεχής η κίνηση, ενώ δυσχεραίνει και η αλληλεπίδραση του χρήστη με το περιβάλλον. Για τους λόγους αυτούς θα πρέπει οι σχεδιαστές να καταλήξουν σε έναν καλό συμβιβασμό ως προς τον αριθμό των τριγώνων, ώστε να πετύχουν καλής ποιότητας απεικόνιση χωρίς να διακινδυνεύουν την ομαλή λειτουργία του συστήματος σε πραγματικό χρόνο. Σήμερα υπάρχουν κατάλληλοι αλγόριθμοι απλοποίησης μεγάλων πολυγωνικών πλεγμάτων, έτσι ώστε να μειωθεί σημαντικά ο αριθμός των τριγώνων χωρίς να αλλάζει σημαντικά η οπτική απεικόνισή τους. Επιπλέον, υπάρχουν και άλλες τεχνικές που χρησιμοποιούνται σε περιβάλλοντα πραγματικού χρόνου για να μειωθεί ακόμα περισσότερο ο αριθμός των πολυγώνων που τελικά λαμβάνονται υπόψη κατά τη διαδικασία της οπτικής απόδοσης, όπως η χρήση πολλαπλών επιπέδων λεπτομέρειας των μοντέλων (level of detail) ανάλογα με την απόστασή τους από το σημείο παρατήρησης, η χρήση δομών δεδομένων διαμέρισης χώρων (π.χ. δέντρα BSP) μέσω των οποίων αγνοούνται τα αντικείμενα που ανήκουν σε μη ορατές περιοχές κ.ά.

Πολλά από τα τρισδιάστατα αντικείμενα μιας σκηνής θέλουμε να έχουν δυναμική συμπεριφορά, δηλαδή να μπορούν να μεταβάλλονται οι γεωμετρικές ιδιότητές τους στον χρόνο. Αυτή η μεταβολή μπορεί να είναι συνέπεια είτε των νόμων του περιβάλλοντος που καθορίζουν την εξέλιξη του είτε της αλληλεπίδρασης του χρήστη με αυτό. Για παράδειγμα, σε ένα περιβάλλον μπορεί να θέλουμε ένα αυτοκίνητο να έχει τη δυνατότητα να κινείται πάνω στον δρόμο. Άρα η θέση του αυτοκινήτου σε σχέση με την υπόλοιπη σκηνή θα πρέπει να έχει τη δυνατότητα να μεταβάλλεται. Επιπλέον, μπορεί να θέλουμε κάθε φορά που στρίβει το αυτοκίνητο να περιστρέφονται και οι ρόδες του προς την ίδια διεύθυνση. Επομένως και στο ίδιο το αντικείμενο του αυτοκινήτου θα πρέπει να μπορεί να αλλάζει η περιστροφή της κάθε ρόδας. Τέτοιες μεταβολές στο περιβάλλον είναι πολύ δύσκολο να γίνουν αν όλα τα αντικείμενα της σκηνής περιγράφονται από το ίδιο πλέγμα και μοιράζονται κοινές κορυφές ή και ακμές. Στην περίπτωση αυτήν είναι προτιμότερο κάθε τμήμα της σκηνής του οποίου η θέση, η περιστροφή ή ακόμα και η μορφή δύναται να αλλάξει ανεξάρτητα από τα υπόλοιπα (όπως το αυτοκίνητο σε σχέση με τη σκηνή ή ακόμα και οι ρόδες σε σχέση με το αυτοκίνητο) να μοντελοποιηθεί ως ξεχωριστό αντικείμενο. Στην περίπτωση αυτήν, οι αλλαγές στη θέση, περιστροφή ή ακόμα και στο μέγεθος ενός δυναμικού αντικειμένου προκύπτουν μέσω γεωμετρικών μετασχηματισμών, όπως θα δούμε στη συνέχεια.

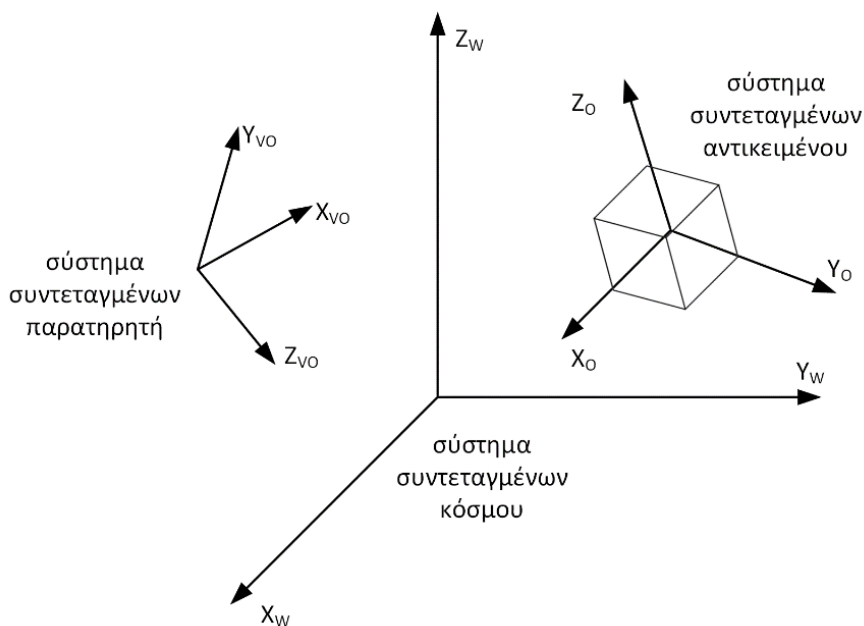
2.2 Γεωμετρικοί μετασχηματισμοί

Για να κατασκευάσουμε ένα εικονικό περιβάλλον θα χρειαστούμε ένα τρισδιάστατο καρτεσιανό πλαίσιο αναφοράς εντός του οποίου θα είναι τοποθετημένα τα αντικείμενα. Το κάθε αντικείμενο είναι μοντελοποιημένο σε σχέση με το δικό του πλαίσιο αναφοράς. Επίσης, η άποψη του χρήστη, δηλαδή η θέση από την οποία θα γίνεται η προοπτική απεικόνιση της σκηνής σε δύο διαστάσεις, έχει κι αυτή το δικό της πλαίσιο αναφοράς. Για λόγους συνέπειας, τα τρία αυτά πλαίσια αναφοράς ορίζονται ως δεξιόστροφα συστήματα συντεταγμένων (εικ. 2.3). Τα αντικείμενα του περιβάλλοντος μοντελοποιούνται αρχικά εντός του δικού τους συστήματος συντεταγμένων (τοπικό σύστημα συντεταγμένων) και στη συνέχεια με τη χρήση μιας σειράς από μετασχηματισμούς καθορίζεται το μέγεθός τους, η θέση τους και η περιστροφή τους στη σκηνή. Το αποτέλεσμα είναι οι κορυφές τους να αποκτήσουν κατάλληλες τιμές συντεταγμένων ως προς το πλαίσιο αναφοράς του κόσμου (σφαιρικό σύστημα συντεταγμένων).

Οι γεωμετρικοί μετασχηματισμοί είναι ένα πολύ σημαντικό θέμα στα γραφικά με υπολογιστή και κατ'επέκταση στα εικονικά περιβάλλοντα, καθώς αποτελούν τα «εργαλεία της αλλαγής» (Θεοχάρης κ.ά., 2012). Χάρη σε αυτούς μπορούμε να καθορίσουμε τις δυνατότητες και τα όρια κίνησης των διάφορων αντικειμένων του περιβάλλοντος, καθώς και να προκαλέσουμε την κίνηση αυτήν κατά τη διάρκεια της λειτουργίας του, είτε ως συνέπεια των «κανόνων» του περιβάλλοντος, είτε λόγω των αλληλεπιδράσεων του χρήστη με αυτό.

Οι πιο συχνά χρησιμοποιούμενοι μετασχηματισμοί είναι η *αλλαγή κλίμακας* (scale), η *περιστροφή* (rotation) και η *μετατόπιση* (translation). Και οι τρεις μετασχηματισμοί μπορούν να εκφραστούν ως πίνακες και στη συνέχεια να συγχωνευτούν σε έναν μοναδικό πίνακα μετασχηματισμού. Κάθε φορά που απεικονίζεται μία άποψη της σκηνής, δηλαδή σε κάθε πλάνο στα συστήματα πραγματικού χρόνου, υπολογίζεται αυτός ο

πίνακας και εφαρμόζεται στις συντεταγμένες του αντικειμένου ώστε να καθοριστεί η πραγματική του θέση στη σκηνή. Παρόλο που η σειρά με την οποία εφαρμόζονται συνήθως οι μετασχηματισμοί είναι πρώτα αλλαγή κλίμακας, μετά περιστροφή και στο τέλος μετατόπιση, θα τους παρουσιάσουμε με τη λογική σειρά μετατόπιση, αλλαγή κλίμακας και περιστροφή.



Εικόνα 2.3 Σύστημα συντεταγμένων κόσμου, παρατηρητή και αντικειμένου.

2.2.1 Μετατόπιση

Ο μετασχηματισμός της μετατόπισης επιτρέπει σε ένα αντικείμενο να τοποθετηθεί σε ένα καθορισμένο σημείο της σκηνής. Ορίζονται τρεις τιμές συντεταγμένων μετατόπισης και αποδίδονται σε κάθε κορυφή του αντικειμένου. Οι τιμές αυτές φυλάσσονται σε έναν πίνακα μετασχηματισμού του αντικειμένου και μόνο όταν απαιτείται η οπτική απόδοση του αντικειμένου ή άλλοι έλεγχοι, όπως η αναγνώριση σύγκρουσης (collision detection), εφαρμόζεται ο πίνακας αυτός στις κορυφές για να αποκαλυφθεί η πραγματική τους θέση. Η ιδέα αυτή είναι θεμελιώδης σε όλους τους τύπους μετασχηματισμών. Οι αλλαγές στη θέση, περιστροφή ή κλίμακα του αντικειμένου συνεπάγονται αλλαγές μόνο στα στοιχεία του σχετικού πίνακα και όχι στις αποθηκευμένες τιμές συντεταγμένων του αντικειμένου.

Επειδή ο μετασχηματισμός της μετατόπισης επί της ουσίας προκαλεί πρόσθεση ενός διανύσματος (των τιμών της μετατόπισης) σε ένα άλλο (τις συντεταγμένες της κορυφής), η πράξη αυτή δεν μπορεί να υποστηριχθεί από πίνακες διαστάσεων 3 x 3. Για τον λόγο αυτόν, στους γεωμετρικούς μετασχηματισμούς χρησιμοποιούμε τις λεγόμενες ομογενείς (homogenous) συντεταγμένες και τους αντίστοιχους πίνακες, προσθέτοντας μία παραπάνω διάσταση. Σύμφωνα με αυτή την προσέγγιση, οι συντεταγμένες των κορυφών περιγράφονται από διανύσματα στήλες 4 x 1 και οι μετασχηματισμοί από πίνακες 4 x 4.

Για να μετατοπίσουμε ένα διάνυσμα με συντεταγμένες (x, y, z) σε μια νέα θέση (x', y', z') θα προσθέσουμε σε καθεμία από τις τιμές των συντεταγμένων του τις τιμές t_x, t_y και t_z αντίστοιχα. Οι τελευταίες αποτελούν τις τιμές μετατόπισης στους τρεις άξονες.

$$x' = x + t_x$$

$$y' = y + t_y$$

$$z' = z + t_z$$

Η παραπάνω μετατόπιση μπορεί να υπολογιστεί και με τη χρήση ομογενούς πολλαπλασιασμού πινάκων ως εξής:

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Αν εφαρμόσουμε έναν μετασχηματισμό μετατόπισης κατά (t_x, t_y, t_z) στις κορυφές ενός αντικειμένου, τότε το σημείο αναφοράς (η αρχή των αξόνων) του τοπικού συστήματος συντεταγμένων του αντικειμένου θα βρεθεί στη θέση (t_x, t_y, t_z) της σκηνής. Για παράδειγμα, αν έχουμε μοντελοποιήσει έναν κύβο με τέτοιο τρόπο ώστε η αρχή των αξόνων να βρίσκεται στο κέντρο του, τότε η παραπάνω μετατόπιση θα φέρει το κέντρο του κύβου στην καθορισμένη θέση.

Ο μετασχηματισμός σε μορφή πίνακα, που είδαμε παραπάνω, μπορεί να συνδυαστεί με άλλους πίνακες που ελέγχουν το μέγεθος και τον προσανατολισμό του αντικειμένου.

2.2.2 Αλλαγή κλίμακας

Ο μετασχηματισμός της αλλαγής κλίμακας επηρεάζει το μέγεθος ενός αντικειμένου μέσω της αλλαγής κλίμακας των συντεταγμένων όλων των κορυφών του σε σχέση με την αρχή των αξόνων του τοπικού συστήματος συντεταγμένων. Ο πολλαπλασιασμός πινάκων που περιγράφει αυτή την ενέργεια είναι:

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

όπου S_x , S_y και S_z είναι οι μεμονωμένοι βαθμοί αλλαγής κλίμακας για τις συντεταγμένες x , y και z αντίστοιχα. Το ότι υπάρχουν μεμονωμένοι βαθμοί σημαίνει ότι ένα αντικείμενο μπορεί να αλλάξει κλίμακα σε διαφορετικό ποσοστό σε κάθε διάσταση με αποτέλεσμα να δείχνει συμπιεσμένο ή τετρωμένο. Για παράδειγμα, ένας κύβος μοναδιαίων διαστάσεων θα μπορούσε να μετασχηματιστεί σε ένα λεπτό επίπεδο σχήμα μέσω ενός μετασχηματισμού με $S_x = 5$, $S_y = 4$ και $S_z = 0.1$.

Αρκετά συχνή πάντως είναι και η περίπτωση *ομοιόμορφης (uniform) αλλαγής κλίμακας*, δηλαδή $s_x = s_y = s_z$, όταν, για παράδειγμα, θέλουμε να έχουμε στη σκηνή πολλαπλά αντίγραφα ενός αντικειμένου με διαφορετικά μεγέθη (π.χ. δέντρα, βότσαλα, σύννεφα καπνού).

2.2.3 Περιστροφή

Η περιστροφή είναι ο πιο σύνθετος μετασχηματισμός και μπορεί να καθοριστεί με διάφορους τρόπους, καθένας από τους οποίους έχει τα δικά του πλεονεκτήματα.

Περιστροφή γύρω από τους άξονες XYZ

Η περιστροφή ενός αντικειμένου μπορεί να καθοριστεί μέσω τριών γωνιών περιστροφής, μία για κάθε άξονα. Οι τρεις πίνακες που μπορούν να αποδώσουν αυτούς τους μετασχηματισμούς είναι:

$$\begin{bmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi & 0 \\ 0 & \sin \phi & \cos \phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} \cos \omega & 0 & \sin \omega & 0 \\ -\sin \omega & 1 & \cos \omega & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

όπου θ , ϕ και ω οι γωνίες περιστροφής στους άξονες X, Y και Z αντίστοιχα.

Οι παραπάνω μπορούν να συνδυαστούν με οποιαδήποτε αυθαίρετη σειρά. Παρόλα αυτά, καθώς στις περιστροφές δεν ισχύει η αντιμεταθετική ιδιότητα, κάθε ακολουθία στη γενική περίπτωση οδηγεί σε διαφορετικό αποτέλεσμα. Για τον λόγο αυτόν, σε συστήματα στα οποία ορίζονται γωνίες περιστροφής, η εφαρμογή των περιστροφών θα πρέπει να ακολουθεί μια προκαθορισμένη σειρά.

Η περιστροφή μπορεί να γίνει είτε ως προς τους άξονες ενός σταθερού συστήματος συντεταγμένων (εξωγενής περιστροφή — *extrinsic*) είτε ως προς τους άξονες του τοπικού συστήματος συντεταγμένων του αντικειμένου (ενδογενής — *intrinsic*). Στη δεύτερη περίπτωση οι γωνίες ονομάζονται και *γωνίες Euler*. Αποδεικνύεται ότι, αν οι πίνακες περιστροφής μιας εξωγενούς περιστροφής πολλαπλασιαστούν με την αντίστροφη σειρά (π.χ. από XYZ η σειρά να γίνει ZYX), τότε προκύπτει η αντίστοιχη ενδογενής περιστροφή για τις ίδιες γωνίες. Συνήθως στα συστήματα γραφικών χρησιμοποιούνται γωνίες Euler οι οποίες εφαρμόζονται με τη σειρά XYZ.

Η περιστροφή με τη χρήση γωνιών Euler, παρά την ευκολία καθορισμού της περιστροφής και την απλότητά της, παρουσιάζει ορισμένα προβλήματα. Πρώτον, υπάρχουν περιπτώσεις στις οποίες μπορεί να θέλουμε να περιστρέψουμε ένα αντικείμενο γύρω από έναν αυθαίρετο άξονα και όχι γύρω από τους άξονες του συστήματος συντεταγμένων του. Στην περίπτωση αυτήν απαιτείται μια σύνθετη ακολουθία μετασχηματισμών που περιλαμβάνει μετατοπίσεις και περιστροφές με γωνίες Euler για να επιτευχθεί το αποτέλεσμα. Επιπλέον, υπάρχουν περιπτώσεις στις οποίες με τη χρήση των γωνιών Euler προκύπτει το πρόβλημα του «κλειδώματος περιστροφών» (*gimbal lock*). Συγκεκριμένα, υπάρχει περίπτωση μετά από την περιστροφή γύρω από κάποιον άξονα οι άλλοι δύο άξονες να οδηγηθούν στην ίδια κατεύθυνση, με αποτέλεσμα να χαθεί ένας βαθμός ελευθερίας. Μια τέτοια κατάσταση μπορεί να δυσκολέψει τόσο την αυθαίρετη αλλαγή περιστροφής αντικειμένων όσο και αυτόματες διαδικασίες συνθετικής κίνησης (*animation*) μέσω παρεμβολής. Τα παραπάνω προβλήματα μπορούν να επιλυθούν με τη χρήση των *quaternions*, που θα δούμε στη συνέχεια.

Quaternions

Τα *quaternions* είναι ένα μαθηματικό μοντέλο που χρησιμοποιείται για την αναπαράσταση των τρισδιάστατων περιστροφών. Έχουν ορισμένες ενδιαφέρουσες ιδιότητες και είναι εξαιρετικά χρήσιμη μορφή αναπαράστασης όταν θέλουμε να υπολογίζουμε ενδιάμεσες θέσεις κατά την περιστροφή ενός αντικειμένου για την παραγωγή συνθετικής κίνησης (Shoemake, 1985).

Ένα *quaternion* q αποτελείται από τέσσερις πραγματικούς αριθμούς:

$$q = (s, x, y, z)$$

από τους οποίους ο s καλείται βαθμωτό μέρος του q και το διάνυσμα (x, y, z) καλείται το διανυσματικό μέρος του q . Επομένως, το q μπορεί να γραφτεί και ως:

$$q = (s, v), \text{ όπου } v = (x, y, z)$$

Τα *quaternions* είναι μια επέκταση των μιγαδικών αριθμών στις τέσσερις διαστάσεις. Χρησιμοποιούν τις «φανταστικές μονάδες» i , j και k , για τις οποίες ισχύει:

$$i^2 = j^2 = k^2$$

$$ij = k \quad jk = i \quad ki = j$$

$$ji = -k \quad ik = -i \quad kj = -j$$

Το quaternion q ορίζεται ως:

$$q = s + xi + yj + zk$$

Ο πολλαπλασιασμός μεταξύ δύο quaternions μπορεί να υπολογιστεί χρησιμοποιώντας τον παραπάνω τύπο και τις ιδιότητες των τριών φανταστικών αριθμών. Έτσι, αν έχουμε $q_1 = (s_1, x_1, y_1, z_1)$ και $q_2 = (s_2, x_2, y_2, z_2)$, το γινόμενό τους είναι:

$$q_1 q_2 = (s_1 s_2 - x_1 x_2 - y_1 y_2 - z_1 z_2) + (s_1 x_2 + s_2 x_1 + y_1 z_2 - y_2 z_1)j + (s_1 y_2 + s_2 y_1 + z_1 x_2 - z_2 x_1)j + (s_1 z_2 + s_2 z_1 + x_1 y_2 - x_2 y_1)k$$

Ένας πραγματικός αριθμός αντιστοιχεί στο quaternion $(u, 0)$ και ένα διάνυσμα στο $(0, v)$.

Αν θέλουμε να εκφράσουμε μια περιστροφή κατά γωνία θ γύρω από αυθαίρετο άξονα που διέρχεται από την αρχή του συστήματος συντεταγμένων και του οποίου η διεύθυνση ορίζεται από ένα μοναδιαίο διάνυσμα n , θα χρησιμοποιήσουμε το quaternion:

$$q = \left(\cos \frac{\theta}{2}, \sin \frac{\theta}{2} n \right)$$

Το παραπάνω quaternion είναι κι αυτό μοναδιαίο, δηλαδή το μέτρο του είναι ίσο με τη μονάδα. Το μέτρο ενός quaternion υπολογίζεται από τον τύπο:

$$\sqrt{s^2 + x^2 + y^2 + z^2}$$

Έστω ότι θέλουμε να εφαρμόσουμε την παραπάνω περιστροφή σε μια κορυφή (x, y, z) . Αρχικά θα εκφράσουμε το σημείο ως quaternion:

$$p = (0, x, y, z)$$

Το περιστραμμένο σημείο p' θα προκύψει από τον παρακάτω τύπο:

$$p_1 = qp\bar{q}$$

όπου \bar{q} το συζυγές quaternion, που ορίζεται ως εξής:

$$\bar{q} = (s, -v)$$

Ένα ακόμη ενδιαφέρον χαρακτηριστικό της χρήσης των quaternions για την αναπαράσταση περιστροφών είναι ότι μπορούμε να πολλαπλασιάσουμε δύο ή περισσότερες περιστροφές εκφρασμένες σε quaternions και το γινόμενό τους θα είναι η συνδυασμένη περιστροφή. Αυτό σημαίνει ότι με τη χρήση quaternions μπορούμε εύκολα να συνδυάζουμε περιστροφές για την κίνηση των αντικειμένων μας στο περιβάλλον, κάτι που είναι απαραίτητο σε σύνθετες κινήσεις (φανταστείτε π.χ. τις κινήσεις των πλανητών). Επιπλέον, άμεση συνέπεια των παραπάνω είναι ότι μπορούμε εύκολα να μετατρέψουμε μια περιστροφή εκφρασμένη σε γωνίες Euler σε quaternion. Αρκεί να δημιουργήσουμε από ένα quaternion για κάθε ζεύγος άξονα και αντίστοιχης γωνίας περιστροφής και να υπολογίσουμε το γινόμενό τους με τη σειρά που ακολουθείται στις γωνίες Euler.

Τέλος, μια περιστροφή εκφρασμένη σε quaternion μπορεί να μετατραπεί εύκολα σε πίνακα περιστροφής και, κατά συνέπεια, να συνδυαστεί με τους υπόλοιπους μετασχηματισμούς του αντικειμένου

(μετατόπιση και αλλαγή κλίμακας) σε έναν μοναδικό πίνακα μετασχηματισμού. Ο πίνακας περιστροφής που αντιστοιχεί σε περιστροφή που αντιπροσωπεύεται από το quaternion $q = (s, x, y, z)$ είναι:

$$\begin{bmatrix} 1 - 2y^2 - 2z^2 & 2xy - 2sz & 2xz + 2sy & 0 \\ 2xy + 2sz & 1 - 2x^2 - 2z^2 & 2yz - 2sx & 0 \\ 2xz - 2sy & 2yz + 2sx & 1 - 2x^2 - 2y^2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Παρά το γεγονός ότι τα quaternions θεωρούνται σήμερα η καταλληλότερη μορφή αναπαράστασης περιστροφών, η απευθείας ανάθεση τιμών περιστροφής σε μορφή quaternion δεν είναι εύκολη. Για τον λόγο αυτόν οι περισσότερες εφαρμογές τρισδιάστατης μοντελοποίησης ή αναπαράστασης επιτρέπουν στους χρήστες να ορίσουν την περιστροφή των αντικειμένων σε γωνίες Euler ή ως ζεύγος άξονα και γωνίας περιστροφής, και στη συνέχεια το ίδιο το σύστημα μετατρέπει τις περιστροφές αυτές σε quaternions. Για παράδειγμα, στο γραφικό περιβάλλον του Second Life και του Unity οι περιστροφές ορίζονται σε γωνίες Euler και μάλιστα εκφρασμένες σε μοίρες, όχι σε ακτίνια. Όμως και στα δύο συστήματα η εσωτερική αναπαράσταση γίνεται με τη χρήση quaternions.

2.3 Χρώματα και υφές

Για την οπτική απεικόνιση ενός αντικειμένου, πέρα από τον καθορισμό της γεωμετρίας του και την τοποθέτησή του στο περιβάλλον, απαιτείται και ο κατάλληλος χρωματισμός του. Απαραίτητο λοιπόν βήμα σε όλα ανεξαιρέτως τα συστήματα τρισδιάστατης απεικόνισης είναι η χρωματική απόδοση της επιφάνειας των αντικειμένων, η οποία προκύπτει ως συνδυασμός του χρώματος ή της υφής που έχει αποδοθεί στο αντικείμενο, του φωτισμού της σκηνής και της οπτικής γωνίας από την οποία παρατηρούμε τη σκηνή. Αρχικά θα εξετάσουμε τα μοντέλα περιγραφής χρώματος και τη διαδικασία απόδοσης χρώματος και υφής στα τρισδιάστατα αντικείμενα.

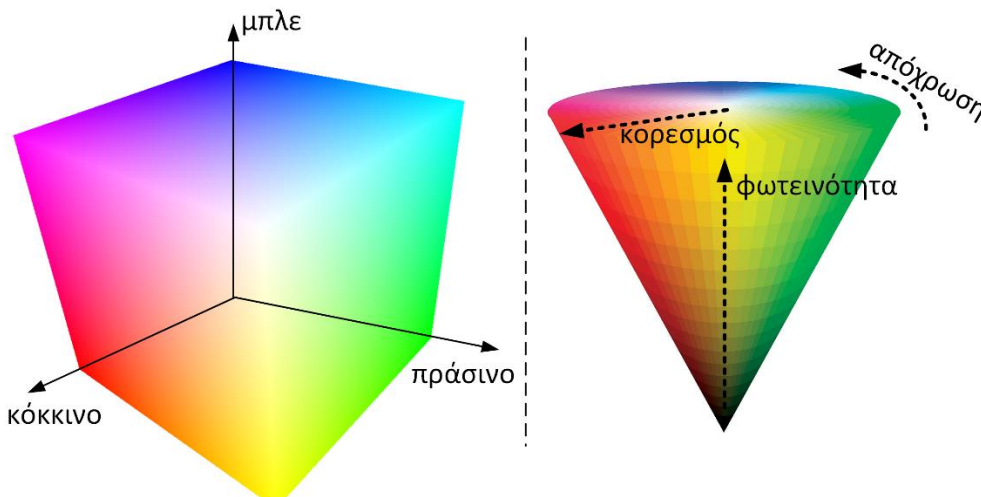
Χρώμα	r	g	b
Λευκό	1	1	1
Μαύρο	0	0	0
Κόκκινο	1	0	0
Μπλε	0	0	1
Πράσινο	0	1	0
Κίτρινο	1	1	0
Πορτοκαλί	1	0.6	0
Μοβ	0.5	0	0.5
Γαλάζιο	0	1	1
Λαδί	0.5	0.5	0

Πίνακας 2.3 Τιμές RGB ορισμένων συνηθισμένων χρωμάτων.

Η έρευνα έχει δείξει ότι το ανθρώπινο μάτι αντιλαμβάνεται το χρωματικό φάσμα σε τρεις επικαλυπτόμενες ζώνες συχνοτήτων με μέγιστη ευαισθησία στα χρώματα κόκκινο, πράσινο και μπλε. Τα χρώματα αυτά έχουν υιοθετηθεί ως τα τρία βασικά προσθετικά χρώματα για τον συνδυασμό πηγών φωτός σε ένα τελικό χρώμα. Για τον λόγο αυτόν, η τεχνολογία των υπολογιστών χρησιμοποιεί συνδυασμούς των τριών αυτών βασικών χρωμάτων RGB (Red, Green, Blue) για να περιγράψει ένα οποιοδήποτε χρώμα. Άλλωστε και η τεχνολογία απεικόνισης τόσο σήμερα με τις οθόνες τύπου TFT/LCD όσο και παλιότερα με τις οθόνες καθοδικού σωλήνα CRT βασίζεται στον ίδιο συνδυασμό: κάθε εικονοστοιχείο (pixel) αποδίδεται ως συνδυασμός τριών χρωματικών στοιχείων (κόκκινο, πράσινο και μπλε) καθένα από τα οποία φωτίζει σε διαφορετική ένταση. Το χρώμα περιγράφεται ως μία τριάδα πραγματικών αριθμών από το μηδέν μέχρι το ένα (σε κάποιες παραλλαγές χρησιμοποιούνται ακέραιοι από το 0 έως το 255), όπου ο κάθε αριθμός περιγράφει το βαθμό συμμετοχής του αντίστοιχου χρώματος στο τελικό αποτέλεσμα. Η τιμή 0 σημαίνει καθόλου συμμετοχή, ενώ το 1 είναι πλήρης συμμετοχή. Κατά συνέπεια, το (0, 0, 0) είναι το μαύρο χρώμα και το (1, 1, 1) το λευκό. Στον πίνακα 2.3

μπορούμε να δούμε ορισμένα γνωστά χρώματα και τις αντίστοιχες τιμές τους στο μοντέλο RGB. Η εικόνα 2.4 (αριστερά) δείχνει μια χωρική ερμηνεία (λέγεται και «χρωματικός χώρος» — color space) του μοντέλου RGB.

Παρά το γεγονός ότι οποιοδήποτε χρώμα μπορεί να αναπαρασταθεί ως μια τριάδα RGB, η χρήση αυτού του μοντέλου αναπαράστασης δεν είναι ιδιαίτερα διαισθητική και είναι δύσκολο να εντοπίσει κάποιος ένα συγκεκριμένο χρώμα απλά αυξάνοντας ή μειώνοντας την ένταση των τριών βασικών χρωμάτων. Για να επιλυθεί το πρόβλημα αυτό, χρησιμοποιείται εναλλακτικά ένας διαφορετικός χρωματικός χώρος για την απόδοση χρώματος, ο HSV (Hue, Saturation, Value). Η πρώτη τιμή περιγράφει την απόχρωση, όπου ο χρήστης επιλέγει ένα από πολλά χρώματα μεταβάλλοντας την τιμή από το 0 μέχρι το 1. Η δεύτερη τον κορεσμό, δηλαδή το πόσο καθαρό είναι το χρώμα σε σχέση με το λευκό. Η μέγιστη τιμή σημαίνει ότι το χρώμα είναι καθαρό, ή πλήρως κορεσμένο, ενώ όσο μειώνεται η τιμή του το χρώμα «λευκαίνει». Η τρίτη τιμή περιγράφει τη φωτεινότητα του χρώματος, όπου όσο η τιμή προσεγγίζει το 0 το χρώμα «μαυρίζει», ενώ στο 1 το χρώμα φωτίζεται πλήρως. Η εικόνα 2.4 (δεξιά) δείχνει το χρωματικό χώρο HSV.



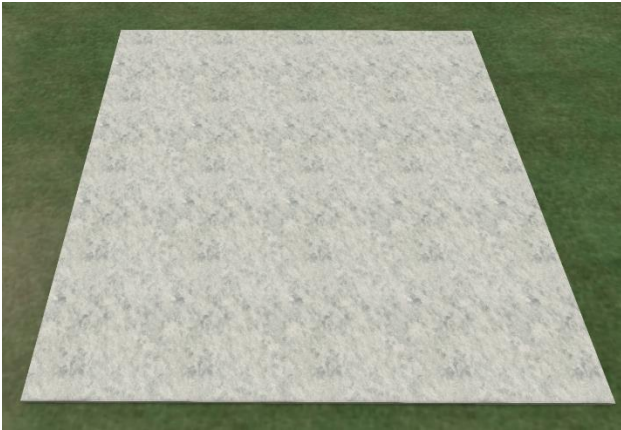
Εικόνα 2.4 Οι «χρωματικοί χώροι RGB και HSV.

Υπάρχουν πολλές περιπτώσεις αντικειμένων σε εικονικά περιβάλλοντα που δεν είναι πλήρως αδιαφανή και απαιτείται πέρα από το χρώμα τους μια περιγραφή του ποσοστού διαφάνειας της επιφάνειάς τους. Μπορούμε να σκεφτούμε διάφορα ημιδιαφανή αντικείμενα που ενδέχεται να εμφανίζονται σε εικονικούς κόσμους, όπως παράθυρα, νερό, καπνός κ.ά. Θα μπορούσε να χρησιμοποιηθεί ημιδιαφάνεια ακόμα και σε μη ρεαλιστικές οπτικοποιήσεις, όπως οπτικά βοηθήματα πλοήγησης, πληροφορίες σχετικά με χώρους, αντικείμενα ή χρήστες του περιβάλλοντος κ.ά. Για όλους τους παραπάνω λόγους, σε όλα τα σύγχρονα συστήματα τρισδιάστατης οπτικοποίησης δίνεται η δυνατότητα προσθήκης πληροφορίας διαφάνειας μαζί με το χρώμα της επιφάνειας των αντικειμένων. Η πληροφορία αυτή, που συνήθως ονομάζεται κανάλι alpha ή βαθμός αδιαφάνειας (opacity), αποδίδεται ως μια τιμή μεταξύ του 0, που σημαίνει πλήρης διαφάνεια, και του 1, που σημαίνει αδιαφάνεια.

Οι πληροφορίες χρώματος και διαφάνειας αποδίδονται είτε συνολικά στο αντικείμενο, είτε (συχνότερα) σε μεμονωμένα πολύγωνα αυτού.

Πέρα από την απόδοση χρώματος, πολύ συχνή είναι και η χρήση υφών για τη ρεαλιστικότερη απεικόνιση του υλικού της επιφάνειας του αντικειμένου. Οι υφές (textures) είναι εικόνες που αποτυπώνονται πάνω στην επιφάνεια των αντικειμένων και χρησιμοποιούνται συνήθως για να προσδώσουν μεγαλύτερο ρεαλισμό στην απεικόνιση (Haeblerli & Segal, 1993). Για παράδειγμα, αν θέλουμε να απεικονίσουμε κάποιο είδος εδάφους, π.χ. άμμο, γρασίδι κ.λπ., είναι πρακτικά αδύνατο με τις σημερινές επιδόσεις της τεχνολογίας να δημιουργήσουμε γεωμετρικά μοντέλα για κάθε μεμονωμένο στοιχείο (π.χ. για κάθε κόκκο άμμου). Η λύση που συνήθως επιλέγεται στην περίπτωση αυτή είναι η απόδοση κάποιας υφής πάνω στη λεία επιφάνεια του εδάφους, έτσι ώστε οπτικά το αποτέλεσμα να θυμίζει πάρα πολύ τον αντίστοιχο πραγματικό χώρο. Συνήθως οι υφές σε αυτή την περίπτωση προέρχονται από φωτογραφίες και είναι επεξεργασμένες με τέτοιο τρόπο ώστε να μπορούν να «κολλάνε» δίπλα δίπλα πολλαπλά αντίγραφα τους χωρίς να φαίνονται ασυνέχειες (εικ. 2.5) Επιπλέον, υπάρχουν υλικά που ακόμα και σε επίπεδες επιφάνειες παρουσιάζουν χρωματικές διαφοροποιήσεις, όπως το ξύλο ή το σκουριασμένο μέταλλο, καθώς και χρωματισμένα τμήματα αντικειμένων, όπως το σπρέι σε

τοίχο, η εικόνα μιας μπλούζας κ.λπ. Οι παραπάνω περιπτώσεις αποδίδονται επίσης αρκετά ικανοποιητικά με τη χρήση υφών.



Εικόνα 2.5 Υφή που επαναλαμβάνεται πάνω σε επιφάνεια αντικειμένου. Σύλληψη από το περιβάλλον OpenSimulator.

Για την απόδοση υφής σε ένα αντικείμενο απαιτείται και ο καθορισμός του τρόπου με τον οποίο θα αντιστοιχιστεί η εικόνα στις επιμέρους επιφάνειες του αντικειμένου. Η διαδικασία αυτή, που ονομάζεται *αντιστοίχιση υφής (texture mapping)*, περιλαμβάνει τον καθορισμό των πολυγώνων της επιφάνειας στα οποία θα αποδοθεί η υφή και τη συσχέτιση των κορυφών του αντικειμένου με σημεία της εικόνας υφής, βάσει των οποίων θα γίνει η απεικόνιση. Στην περίπτωση περίπλοκων αντικειμένων, π.χ. ανθρώπινο πρόσωπο ή ρούχα, η διαδικασία αυτή είναι αρκετά σύνθετη και γίνεται μέσω κατάλληλων προγραμμάτων τρισδιάστατης μοντελοποίησης. Σε ό,τι αφορά την απόδοση υφής σε επίπεδες επιφάνειες, εκεί συνήθως δίνεται η δυνατότητα ρύθμισης παραμέτρων παραπλήσιων με τους τρισδιάστατους μετασχηματισμούς αλλά σε δύο διαστάσεις, δηλαδή της κλίμακας της εικόνας, της περιστροφής και της μετατόπισής της πάνω στο επίπεδο της επιφάνειας.

Οι υφές συνήθως συνδυάζονται με το χρώμα και τη διαφάνεια που έχει αποδοθεί στο αντικείμενο, ενώ υπάρχουν και περιπτώσεις περιβαλλόντων που επιτρέπουν τον συνδυασμό παραπάνω από μίας υφών. Το τελευταίο χαρακτηριστικό είναι ιδιαίτερα χρήσιμο για την ενίσχυση του ρεαλισμού και για τη δημιουργία διαφοροποιήσεων σε πολλαπλές περιοχές του σκηνικού. Για παράδειγμα, θα μπορούσε σε μια σκηνή να συνδυαστεί στο έδαφος η υφή του δρόμου με την υφή του χιονιού για να παράξει παραλλαγές από μερικώς χιονισμένο δρόμο.

Τέλος, μπορούν να οριστούν υφές των οποίων οι εικόνες εμπεριέχουν διαφάνεια, και η διαφάνεια αυτή να αποδοθεί στην τρισδιάστατη επιφάνεια, ή ακόμα και υφές από κινούμενες εικόνες για να αποδώσουν διάφορα οπτικά εφέ, όπως φωτιά, κίνηση νερού, καπνός κ.λπ.

2.4 Φωτισμός και σκίαση

Η οπτική απόδοση μιας σκηνής προϋποθέτει τον καθορισμό των πηγών φωτισμού του περιβάλλοντος και την τελική φωτοσκίαση της επιφάνειας των αντικειμένων βάσει της διεύθυνσης και έντασης του συνολικού φωτός που πέφτει πάνω σε αυτές. Η κατάλληλη τοποθέτηση των φώτων στη σκηνή, η ποιότητα των αλγορίθμων φωτοσκίασης που υποστηρίζει το σύστημα και η χρήση πρόσθετων τεχνικών, όπως σκιές, τραχιές επιφάνειες και αντανάκλασεις, επιδρούν καταλυτικά στον ρεαλισμό του περιβάλλοντος. Με δεδομένο ότι η απεικόνιση του τρισδιάστατου χώρου ενός εικονικού κόσμου επιφάνειας εργασίας γίνεται σε δύο διαστάσεις, δηλαδή στην οθόνη του χρήστη, ο φωτισμός και η φωτοσκίαση μπορούν να υποκαταστήσουν σε μεγάλο βαθμό τη χαμένη διάσταση του βάθους. Καλά τοποθετημένα φώτα βοηθούν την πλοήγηση των χρηστών και την κατανόηση του χώρου, καθώς είναι δύσκολο να διακρίνει κάποιος χώρους και διαδρομές σε σκοτεινά σημεία του περιβάλλοντος. Η τρισδιάστατη μορφή των αντικειμένων διακρίνεται καλύτερα μέσω της κατάλληλης φωτοσκίασης, ενώ και η χρήση σκιών μπορεί να βοηθήσει στην καλύτερη αντίληψη του βάθους και των σχετικών θέσεων μεταξύ των αντικειμένων.

Τα είδη των φώτων που μπορεί να τοποθετήσει κάποιος σε τρισδιάστατες σκηνές είναι συνήθως τέσσερα. Ο *φωτισμός περιβάλλοντος (ambient)* είναι ένα μοναδικό σταθερό φως χωρίς διεύθυνση που περιγράφεται μόνο από το χρώμα και την ένταση. Συνήθως συμπληρώνει τα άλλα είδη φωτός για να αναδείξει

τα χρώματα και τις υφές των αντικειμένων σε σκοτεινές περιοχές της σκηνής. Το *σημειακό φως (point light)* ξεκινάει από ένα συγκεκριμένο σημείο του χώρου και εκτείνεται προς όλες τις διευθύνσεις καλύπτοντας μια νοητή σφαίρα. Η έντασή του μπορεί να είναι σταθερή ή να μειώνεται γραμμικά ή εκθετικά μέχρι να πέσει στο μηδέν. Ο *προβολέας (spot light)* συμπεριφέρεται σαν το σημειακό φως, δηλαδή ξεκινάει από ένα συγκεκριμένο σημείο στον χώρο, αλλά έχει μια καθορισμένη διεύθυνση καλύπτοντας έτσι έναν νοητό κώνο. Τα σημειακά φώτα και οι προβολείς χρησιμοποιούνται συνήθως στη σκηνή για να μοντελοποιήσουν τον τεχνητό φωτισμό του χώρου ή φως από φλόγα. Τέλος, το *φως κατεύθυνσης (directional light)* δεν έχει αρχή, ορίζεται μόνο η διεύθυνσή του και παράγει παράλληλες ακτίνες προς τη διεύθυνση αυτή. Συνήθως χρησιμοποιείται για να μοντελοποιήσει το φως του ήλιου, καθώς η απόσταση του ήλιου είναι τέτοια που οι ακτίνες του είναι σχεδόν παράλληλες.

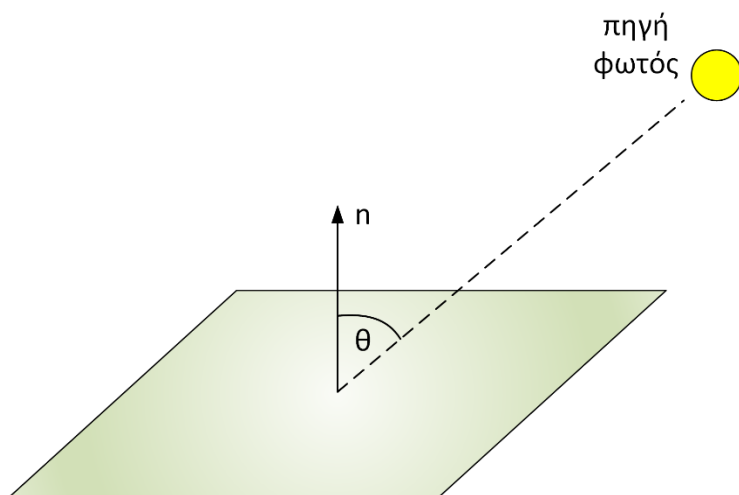
Τώρα που είδαμε διάφορους τρόπους φωτισμού της σκηνής, θα πρέπει να εξετάσουμε αντίστοιχα και τον τρόπο με τον οποίο το φως αντανακλάται από τις επιφάνειες των αντικειμένων και προκύπτει ο τελικός χρωματισμός τους. Έχει γίνει μεγάλη έρευνα σχετικά με την ανάκλαση του φωτός σε διάφορα είδη υλικών και την κατασκευή υπολογιστικών μοντέλων που περιγράφουν αυτά τα φαινόμενα (Goral κ.ά., 1984). Τα πιο πλήρη όμως από αυτά τα μοντέλα απαιτούν υπερβολικό υπολογιστικό χρόνο για να εκτελεστούν, με αποτέλεσμα να είναι ακατάλληλα για χρήση σε εφαρμογές πραγματικού χρόνου.

Μια απλοποιημένη προσέγγιση στην ανάκλαση του φωτός που χρησιμοποιείται συχνότερα σήμερα σε παιχνίδια και εικονικούς κόσμους είναι ο συνδυασμός δύο ειδών ανάκλασης: της διάχυσης (diffuse) και της κατοπτρικής (specular). Το φως που ανακλάται σε μία επιφάνεια διάχυσης εκπέμπεται ομοιόμορφα προς όλες τις διευθύνσεις και ως εκ τούτου είναι ανεξάρτητο από τη θέση του παρατηρητή. Όμως στα γυαλιστερά αντικείμενα δημιουργούνται επιπλέον λάμψεις στην επιφάνειά τους από αντανακλάσεις των πηγών φωτισμού που σχετίζονται με το σημείο παρατήρησης. Η κατοπτρική ανάκλαση μοντελοποιεί αυτό το φαινόμενο.

Για να υπολογίσουμε την ανάκλαση διάχυσης από μία πηγή φωτός σε ένα σημείο της επιφάνειας ενός αντικειμένου, αρκεί να γνωρίζουμε τον παράγοντα διάχυσης της επιφάνειας K_d , την ένταση του φωτός I_i και το συνημίτονο της γωνίας θ μεταξύ της διεύθυνσης του φωτός και του κανονικού διανύσματος της επιφάνειας (εικ. 2.6). Η αντανάκλαση διάχυσης I_d περιγράφεται από τον τύπο:

$$I_d = I_i K_d \cos \theta$$

Το συνημίτονο της γωνίας θ μπορεί να βρεθεί με εσωτερικό γινόμενο μεταξύ του κανονικού διανύσματος και του διανύσματος διεύθυνσης του φωτός.



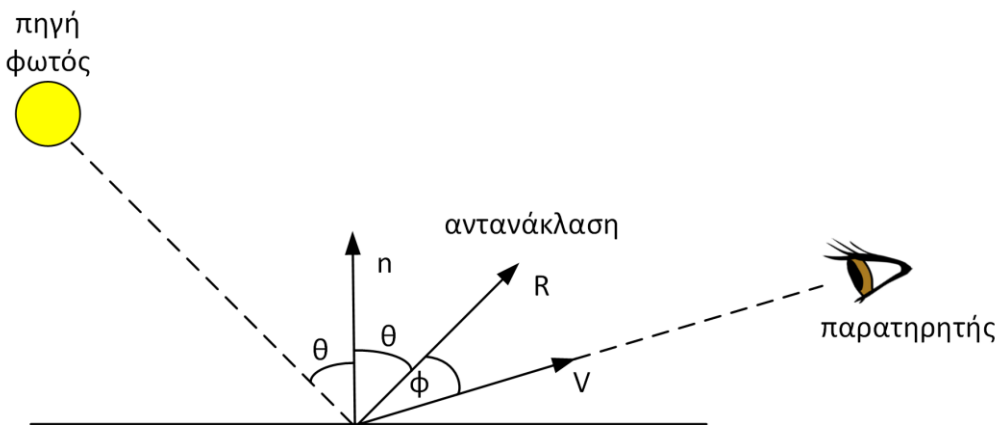
Εικόνα 2.6 Αντανάκλαση διάχυσης στην επιφάνεια ενός αντικειμένου.

Οι λείες και γυαλιστερές επιφάνειες μπορούν να αποδοθούν οπτικά με τον συνδυασμό ανάκλασης διάχυσης με λάμψη λόγω κατοπτρικής ανάκλασης. Αυτή η επιπρόσθετη φωτεινή περιοχή της γυαλάδας αντιπροσωπεύει την αντανάκλαση της πηγής φωτός στο πεδίο όρασης του παρατηρητή και αποτελεί μια ισχυρή οπτική ένδειξη για να ξεχωρίζουν τα γυαλιστερά από τα θαμπά ή τραχιά αντικείμενα. Το μέγεθος της λάμψης πάνω στην επιφάνεια μπορεί να διαφοροποιείται ώστε να αποδίδεται οπτικά ένα μεγάλο εύρος υλικών επιφανειών.

Για τον υπολογισμό της κατοπτρικής ανάκλασης χρειάζεται να συμπεριληφθεί και η θέση του παρατηρητή. Όπως βλέπουμε και στην εικόνα 2.7, όταν μια ακτίνα φωτός προσπίπτει σε ένα σημείο της επιφάνειας, η γωνία ανάκλασης θ ισούται με τη γωνία πρόσπτωσης. Όμως, αν η διεύθυνση παρατήρησης απέχει γωνία ϕ σε σχέση με τη διεύθυνση ανάκλασης, τότε η ένταση της κατοπτρικής ανάκλασης μειώνεται αντίστοιχα. Στον τύπο υπολογισμού της έντασης ανάκλασης θα προσθέσουμε και μια παράμετρο γυαλάδας (gloss parameter) η οποία επηρεάζει το τελικό σχήμα της λάμψης. Ο τύπος είναι:

$$I_s = I_i K_s \cos^g \phi$$

όπου I_i είναι η ένταση της πηγής φωτός, K_s η ένταση του παράγοντα κατοπτρικής ανάκλασης, ϕ η γωνία μεταξύ του διανύσματος παρατήρησης και του διανύσματος ανάκλασης και g η παράμετρος γυαλάδας.



Εικόνα 2.7 Κατοπτρική αντανάκλαση.

Με βάση τα παραπάνω, ο τελικός τύπος υπολογισμού της συνολικής ανάκλασης φωτός συνυπολογίζοντας και τον φωτισμό περιβάλλοντος (ambient) θα προκύπτει ως εξής:

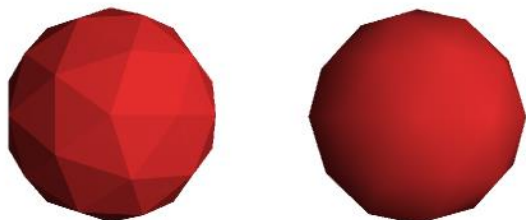
$$I = I_a K_a + [I_i K_d \cos \theta + I_i K_s \cos^g \phi]$$

όπου I_a η ένταση του φωτισμού περιβάλλοντος και K_a ο παράγοντας φωτισμού περιβάλλοντος του αντικειμένου. Ο όρος στις αγκύλες θα πρέπει να επαναληφθεί για κάθε πηγή φωτός στη σκηνή.

Ο παραπάνω τύπος υπολογίζεται τρεις φορές, μία για κάθε ανεξάρτητο κανάλι χρώματος (κόκκινο, πράσινο και μπλε) και το αποτέλεσμα είναι το φωτοσκιασμένο χρώμα της επιφάνειας. Αν κατά τους υπολογισμούς προκύψει κάποια τιμή έντασης εκτός των ορίων $[0,1]$ τότε αντικαθίσταται με την τιμή του κοντινότερου ορίου. Όπως προκύπτει από τα παραπάνω, θα πρέπει για κάθε επιφάνεια να γνωρίζουμε παράγοντα φωτισμού περιβάλλοντος, παράγοντα διάχυσης και παράγοντα κατοπτρικής ανάκλασης για τα τρία χρωματικά κανάλια. Επί της ουσίας λοιπόν θα πρέπει να ορίσουμε τρία είδη χρώματος, το *χρώμα φωτισμού περιβάλλοντος*, το *χρώμα διάχυσης* και το *χρώμα κατοπτρικής ανάκλασης*. Συνήθως στα δύο πρώτα αποδίδουμε το πραγματικό χρώμα της επιφάνειας και στο τρίτο το λευκό.

Όπως μπορούμε να παρατηρήσουμε από τα παραπάνω, το χρώμα της φωτοσκίασης εξαρτάται από το κανονικό διάνυσμα της κάθε επιφάνειας των αντικειμένων. Με δεδομένο όμως ότι τα αντικείμενά μας αποτελούνται από επίπεδα πολύγωνα, κάθε πολύγωνο θα έχει ένα μοναδικό χρώμα μετά την εφαρμογή των αλγορίθμων ανάκλασης. Κάτι τέτοιο θα κάνει τα επιμέρους πολύγωνα ενός αντικειμένου να διακρίνονται με το μάτι. Αυτό είναι επιθυμητό για τη μοντελοποίηση αντικειμένων του πραγματικού κόσμου που έχουν διακριτές επιφάνειες, όπως π.χ. ένας κύβος, όμως στην περίπτωση καμπύλων επιφανειών, τα αποτελέσματα δεν είναι ικανοποιητικά. Για παράδειγμα, οι σφαίρες φαίνονται σαν πολύεδρα. Η λύση στο πρόβλημα αυτό είναι η λεγόμενη *ομαλή φωτοσκίαση (smooth shading)*, σύμφωνα με την οποία, σε αντίθεση με την *επίπεδη φωτοσκίαση (flat shading)* που περιγράψαμε προηγουμένως, αποδίδεται ένα κανονικό διάνυσμα σε κάθε κορυφή του πλέγματος. Συνήθως το διάνυσμα αυτό υπολογίζεται αυτόματα ως ο μέσος όρος των κανονικών διανυσμάτων των πολυγώνων στις οποίες ανήκει η κορυφή. Εφόσον λοιπόν σε κάθε πολύγωνο θα υπάρχει διαφορετικό

κανονικό διάνυσμα για κάθε κορυφή, το πολύγωνο ως προς τη φωτοσκίαση θα συμπεριφέρεται ως καμπύλη επιφάνεια. Δηλαδή σε κάθε σημείο της επιφάνειάς του το κανονικό διάνυσμα θα είναι ελαφρώς διαφοροποιημένο και μια αντίστοιχη διαφοροποίηση θα προκύπτει στην τελική φωτοσκίαση. Αυτό το χαρακτηριστικό αποδίδει οπτικά μια «συνέχεια» με τα γειτονικά πολύγωνα και κατά συνέπεια στο οπτικό αποτέλεσμα τα επιμέρους πολύγωνα δεν είναι διακριτά και οι καμπύλες δείχνουν ομαλές. Στην εικόνα 2.8 απεικονίζεται το οπτικό αποτέλεσμα της επίπεδης και της ομαλής φωτοσκίασης.

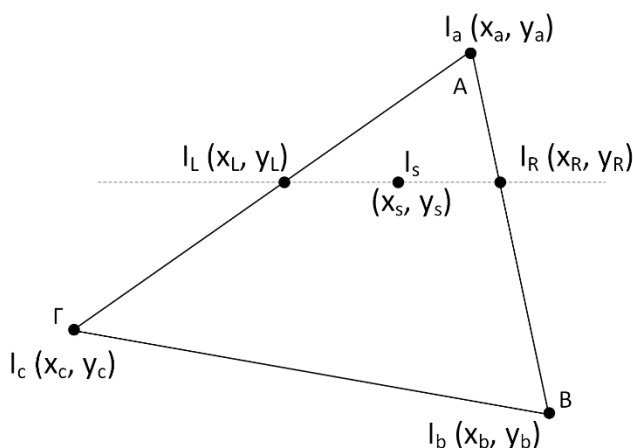


Εικόνα 2.8 Μια σφαίρα με επίπεδη (αριστερά) και ομαλή (δεξιά) φωτοσκίαση.

Ο πιο απλός αλγόριθμος φωτοσκίασης είναι ο *Gouraud*. Σύμφωνα με τον αλγόριθμο αυτόν, υπολογίζεται η συνολική ένταση του φωτός στις τρεις κορυφές του τριγώνου και όλα τα ενδιάμεσα σημεία αυτού υπολογίζονται μέσω παρεμβολής. Αν λοιπόν έχουμε ένα τρίγωνο ΑΒΓ και έχουμε υπολογίσει την ένταση του φωτός I_a , I_b και I_c στις τρεις αντίστοιχες κορυφές για κάθε χρωματικό κανάλι, η ένταση σε ένα τυχαίο σημείο I_s εντός του τριγώνου υπολογίζεται από τον τύπο:

$$I_s = \frac{I_L(x_R - x_s) + I_R(x_s - x_L)}{x_R - x_L}$$

όπου το I_L και το I_R είναι η τιμή της έντασης στην αριστερή και δεξιά πλευρά του τριγώνου αντίστοιχα (βλ. εικ. 2.9). Οι δύο αυτές τιμές μπορούν να υπολογιστούν με γραμμική παρεμβολή μεταξύ των δύο κορυφών που ορίζουν την πλευρά.



Εικόνα 2.9 Υπολογισμός έντασης σε ένα εσωτερικό σημείο του τριγώνου με τον αλγόριθμο *Gouraud*.

Για να υπολογίσουμε την κατοπτρική ανάκλαση σε ένα σημείο χρειάζεται να γνωρίζουμε το κανονικό διάνυσμα αυτού, σύμφωνα με τον τύπο που παρουσιάσαμε παραπάνω. Όμως, με τον αλγόριθμο *Gouraud* δεν έχουμε πρόσβαση στα κανονικά διανύσματα εντός του τριγώνου, με αποτέλεσμα η οπτική απόδοση της κατοπτρικής ανάκλασης να είναι ελλιπής. Τη λύση μπορεί να δώσει ένας άλλος αλγόριθμος φωτοσκίασης, ο *Phong*, ο οποίος, αντί να υπολογίζει τις εντάσεις φωτός των εσωτερικών σημείων με παρεμβολή, χρησιμοποιεί την παρεμβολή

για να υπολογίζει τα κανονικά διανύσματα (Phong, 1975). Έστω λοιπόν ότι έχουμε ένα τρίγωνο ABΓ με κανονικά διανύσματα n_a , n_b και n_c . Μπορούμε να υπολογίσουμε το κανονικό διάνυσμα σε οποιοδήποτε σημείο του τριγώνου με παρεμβολή αντίστοιχη αυτής του αλγορίθμου Gouraud. Τα διανύσματα n_L και n_R θα υπολογιστούν από τους αντίστοιχους τύπους:

$$n_L = \frac{n_a(y_s - y_c) + n_c(y_a - y_s)}{y_a - y_c}$$

$$n_R = \frac{n_a(y_s - y_b) + n_b(y_a - y_s)}{y_a - y_b}$$

και τελικά η τιμή του n_s για ένα τυχαίο σημείο s εντός του τριγώνου θα είναι:

$$n_s = \frac{n_L(x_R - x_s) + n_R(y_s - y_L)}{x_R - x_L}$$

Εφόσον γνωρίζουμε την τιμή του n_s , μπορούμε να υπολογίσουμε την ένταση των τριών χρωματικών καναλιών στο σημείο αυτό χρησιμοποιώντας τον παραπάνω τύπο συνολικής ανάκλασης. Το αποτέλεσμα ποιοτικά θα είναι αρκετά καλύτερο σε σχέση με τη φωτοσκίαση Gouraud, αλλά έχει μεγαλύτερο υπολογιστικό κόστος.

Υπάρχουν τεχνικές φωτοσκίασης που δίνουν ρεαλιστικότερα αποτελέσματα από τις Gouraud και Phong, αλλά το υπολογιστικό κόστος εκτέλεσής τους είναι απαγορευτικό για εφαρμογές πραγματικού χρόνου. Μια τέτοια τεχνική είναι η *radiosity* (Goral κ.ά., 1984), η οποία υπολογίζει τις χρωματικές αντανακλάσεις μεταξύ αντικειμένων, συνήθως σε εσωτερικούς χώρους, και προκαλεί απαλές σκιές που δίνουν πολύ πιο πειστικά και ελκυστικά οπτικά αποτελέσματα. Παρά το γεγονός ότι η τεχνική αυτή δεν μπορεί να υπολογιστεί σε πραγματικό χρόνο, τα αποτελέσματά της μπορούν να ενσωματωθούν σε ένα εικονικό περιβάλλον αν έχουν προϋπολογιστεί κατά την κατασκευή της σκηνής και παραμένουν στατικά στη διάρκεια εκτέλεσης του περιβάλλοντος. Μπορούμε δηλαδή να εφαρμόσουμε την τεχνική αυτή σε μια σκηνή, να υπολογίσουμε τις εντάσεις φωτισμού σε όλες τις επιφάνειες των αντικειμένων και, αποθηκευόντάς τις σε μια εικόνα *απεικόνισης φωτισμού* (*light mapping*), να συνδυάσουμε το αποτέλεσμα με την υπάρχουσα υφή των αντικειμένων. Το αποτέλεσμα είναι ότι η σκηνή δείχνει πολύ πιο ρεαλιστικά φωτοσκιασμένη (βλ. εικ. 2.10). Υπάρχει όμως το μειονέκτημα ότι, αν τυχόν συμβούν αλλαγές στον φωτισμό της σκηνής (π.χ. ανάψει ή σβήσει ένα φως) ή αν μετακινηθούν σημαντικά τα αντικείμενα εντός της, οι αλλαγές αυτές δεν μπορούν να τροποποιήσουν αντίστοιχα τη φωτοαπόδοση της σκηνής.



Εικόνα 2.10 Σκηνή στο Unity στην οποία έχει γίνει απεικόνιση φωτισμού.

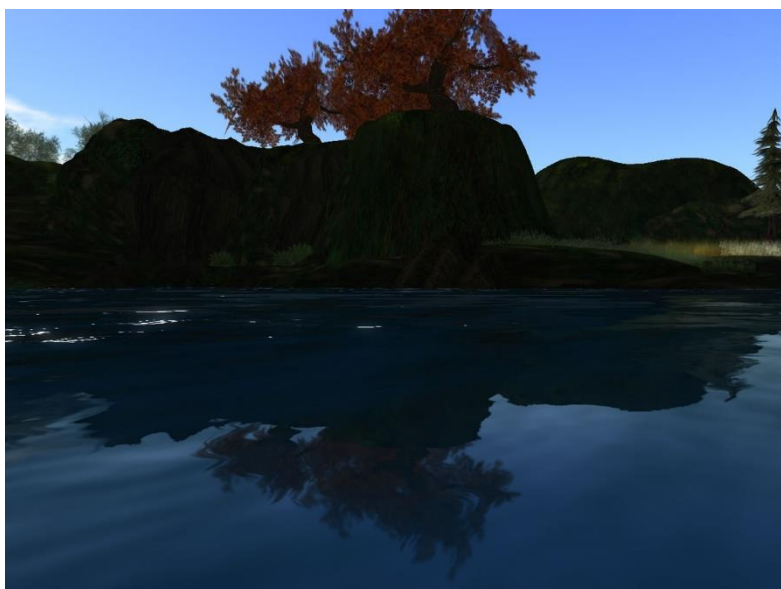


Εικόνα 2.11 Απεικόνιση ανάγλυφου σε σφαίρα. Σύλληψη από το πρόγραμμα Unity.

Μια επιπλέον τεχνική η οποία μπορεί να βελτιώσει τον ρεαλισμό των αντικειμένων είναι η *απεικόνιση ανάγλυφου (bump mapping)*. Σύμφωνα με την τεχνική αυτήν αποδίδεται μια εικόνα πάνω στην επιφάνεια του αντικειμένου με αντίστοιχο τρόπο που αποδίδονται και οι υφές, μόνο που η εικόνα αυτή περιγράφει διαφοροποιήσεις στα κανονικά διανύσματα των αντίστοιχων σημείων της επιφάνειας του αντικειμένου (Blinn, 1978). Με τη χρήση λοιπόν αυτής της εικόνας ανάγλυφου τροποποιείται η φωτοσκίαση στα αντίστοιχα σημεία δίνοντας οπτικά την ψευδαίσθηση εξογκωμάτων ή βαθουλωμάτων πάνω στη γεωμετρία. Έτσι μπορεί κάποιος να αποδώσει τέτοιες ανάγλυφες επιφάνειες σε αντικείμενα (π.χ. την τραχιά επιφάνεια ενός βράχου, τα βαθουλώματα στην επιφάνεια ενός πορτοκαλιού) χωρίς να χρειάζεται να τα περιγράψει σε επίπεδο γεωμετρίας, κάτι που θα επιβάρυνε εξαιρετικά τις υπολογιστικές διεργασίες οπτικοποίησης της σκηνής. Στην εικόνα 2.11 φαίνεται μια απλή σφαίρα στην οποία έχει αποδοθεί απεικόνιση ανάγλυφου χωρίς να επηρεάζονται τα τρίγωνα που περιγράφουν τη γεωμετρία της.

Υπάρχουν υλικά τα οποία έχουν την ικανότητα να καθρεπτίζουν μερικώς τον περιβάλλοντα χώρο παραμορφώνοντάς τον ανάλογα με τη γεωμετρία τους, όπως για παράδειγμα οι μεταλλικές επιφάνειες. Αυτό το οπτικό φαινόμενο μπορεί να προσομοιωθεί και στα εικονικά περιβάλλοντα μέσω της *απεικόνισης περιβάλλοντος (environmental mapping)*. Ο υπολογισμός της πραγματικής αντανάκλασης των αντικειμένων της σκηνής πάνω στην επιφάνεια είναι μια διαδικασία με μεγάλο υπολογιστικό κόστος. Αυτό που συμβαίνει στην απεικόνιση περιβάλλοντος είναι ότι ορίζεται ένας μεγάλος κύβος ή σφαίρα που περικλείει όλη τη σκηνή και στον οποίο αποδίδεται μια προκατασκευασμένη υφή που περιλαμβάνει στοιχεία του εξωτερικού περιβάλλοντος (π.χ. ήλιος και σύννεφα). Ανάλογα με την οπτική γωνία από την οποία βλέπουμε το αντικείμενο, υπολογίζεται σε κάθε σημείο του ένα αντίστοιχο σημείο στην υφή περιβάλλοντος το οποίο θα πρέπει να καθρεπτίζεται, και το χρώμα του σημείου αυτού συνυπολογίζεται με το αντίστοιχο χρώμα του αντικειμένου για την τελική οπτική απόδοση (π.χ. εικόνα 2.12).

Σήμερα, πέρα από τις παραπάνω τεχνικές, δίνεται η δυνατότητα στους σχεδιαστές να εισάγουν δικούς τους αλγορίθμους φωτοσκίασης για την παραγωγή οπτικών αποτελεσμάτων της αρεσκείας τους. Τα προγράμματα που υλοποιούν τέτοιους αλγορίθμους ονομάζονται *shaders* (Lewis, 1994) και υποστηρίζονται τα τελευταία χρόνια από τις κάρτες γραφικών, με αποτέλεσμα η χρήση τους να είναι εξαιρετικά διαδεδομένη. Οι *shaders* μπορούν να τροποποιούν την οπτική απόδοση των επιφανειών ενός αντικειμένου σε επίπεδο τελικού εικονοστοιχείου (pixel) που θα ζωγραφιστεί, σε επίπεδο γεωμετρίας αλλάζοντας τις κορυφές ή/και τα κανονικά διανύσματα, ακόμα και σε επίπεδο υποδιαίρεσης της γεωμετρίας με την παραγωγή νέων τριγώνων. Το αποτέλεσμα είναι διαφόρων ειδών οπτικά εφέ, όπως νερό, καπνός, φωτιά, χιόνι, ομίχλη, σύννεφα κ.ά. ή και εξειδικευμένες απεικονίσεις ανάλογα με τις απαιτήσεις της εφαρμογής.



Εικόνα 2.12 Καθρεπτισμός στο *Second Life*.

Όπως φαίνεται από τα παραπάνω, στις σύγχρονες μηχανές γραφικών απαιτούνται πολλά στοιχεία για να αποδοθεί σωστά ένα αντικείμενο κατά τη διάρκεια οπτικοποίησης της σκηνής. Είδαμε αρχικά ότι δεν αρκεί να προσδιορίσει κάποιος ένα μοναδικό χρώμα στις επιφάνειες του αντικειμένου, αλλά τρία χρώματα: ανάκλασης περιβάλλοντος, ανάκλασης διάχυσης και κατοπτρικής ανάκλασης. Πέρα από αυτά, μπορούν να αποδοθούν μία ή περισσότερες υφές στην επιφάνεια του αντικειμένου, να οριστεί ο βαθμός διαφάνειας, να εισαχθεί απεικόνιση αναγλύφου μέσω μιας κατάλληλα επεξεργασμένης εικόνας αναγλύφου, να ρυθμιστεί το ποσοστό καθρεπτισμού του περιβάλλοντος χώρου εφόσον έχει οριστεί κάποια εξωτερική επιφάνεια που απεικονίζει τον χώρο στα όρια της σκηνής (π.χ. ουρανός, ήλιος και σύννεφα), ακόμα και να εισαχθούν εξειδικευμένοι shaders για την παραγωγή ειδικών εφέ. Οι περισσότερες εφαρμογές δίνουν τη δυνατότητα ορισμού και ρύθμισης των περισσότερων από τις παραπάνω παραμέτρους ή και άλλων πιο εξειδικευμένων, με στόχο τη βελτίωση του οπτικού αποτελέσματος ή της ταχύτητας απεικόνισης της σκηνής. Επειδή θα ήταν εξαιρετικά επίπονο να ξαναορίζονται αυτές οι τιμές σε κάθε αντικείμενο ή τμήμα αντικειμένου που θέλουμε να εμφανίζει παραπλήσια οπτικά χαρακτηριστικά, συνήθως δίνεται η δυνατότητα ορισμού υλικών (materials) στα προγράμματα τρισδιάστατης απεικόνισης. Μπορεί λοιπόν κάποιος να ορίσει έναν αριθμό από υλικά που θα περιλαμβάνει η σκηνή του (π.χ. τζάμι, δρόμος, γρασίδι, σασί αυτοκινήτου, ελαστικά, ζάντες) και να αποδώσει τα υλικά αυτά στα αντικείμενα ή σε τμήματα αυτών.

Τέλος, ένα ακόμη στοιχείο που είναι κρίσιμο όχι μόνο για τον ρεαλισμό της απεικόνισης αλλά και για τη συνολικότερη διάδραση του χρήστη με το περιβάλλον είναι οι σκιές. Οι σκιές βοηθούν τους χρήστες να αντιληφθούν καλύτερα το βάθος· να έχουν μια καλή εκτίμηση για την απόσταση ενός αντικειμένου από το σημείο παρατήρησης βλέποντας τη σχετική τοποθέτηση της σκιάς του. Σκεφτείτε, για παράδειγμα, μια σκηνή ποδοσφαίρου στην οποία η μπάλα βρίσκεται στον αέρα. Αν η απεικόνιση είναι δισδιάστατη και επομένως απουσιάζει η ένδειξη βάθους λόγω στερεοσκοπίας, είναι δύσκολο να εκτιμήσει κάποιος βλέποντας μόνο την μπάλα αν κινείται προς το τέρμα ή εκτός των ορίων του. Αν όμως η μπάλα άφηγε μια κάθετη σκιά στο έδαφος, θα ήταν πολύ πιο εύκολο να εκτιμηθεί η πορεία της, παρόλο που η σκιά δεν θα ήταν ρεαλιστική· στην πραγματικότητα ο φωτισμός γίνεται από τις γωνίες του γηπέδου, όχι από ψηλά πάνω από το κέντρο του. Σε κάθε περίπτωση, είναι δυστυχώς δύσκολο υπολογιστικά πρόβλημα να ζωγραφιστούν όλες οι σκιές σε ένα τρισδιάστατο περιβάλλον (Hasenfratz κ.ά., 2003). Η χρήση προϋπολογισμένης απεικόνισης φωτισμού (light mapping) λύνει το πρόβλημα για τις σκιές των στατικών αντικειμένων, αλλά η ανάγκη για σκίαση των κινούμενων στοιχείων παραμένει. Στις περισσότερες περιπτώσεις δημιουργούνται «ψευδοσκιές», δηλαδή για συγκεκριμένα αντικείμενα του περιβάλλοντος (π.χ. χαρακτήρες σε παιχνίδια) υπολογίζονται οι σκιές τους πάνω στις επιφάνειες άλλων αντικειμένων (κυρίως στο δάπεδο) μέσω κατάλληλων μετασχηματισμών προβολής. Οι browsers των εικονικών κόσμων *Second Life* και *OpenSimulator* δίνουν σήμερα τη δυνατότητα απεικόνισης σκιών σε πραγματικό χρόνο, υπό την προϋπόθεση ότι το υπολογιστικό σύστημα στο οποίο εκτελούνται έχει ισχυρή κάρτα γραφικών (βλ. εικ. 2.13). Επιπλέον, η μηχανή παιχνιδιών *Unity* παράγει σκιές σε πραγματικό χρόνο μόνο για τα φώτα διεύθυνσης (directional light).



Εικόνα 2.13 Σκίές σε πραγματικό χρόνο στο OpenSimulator.

2.5 Κάμερες

Ο ορισμός μίας ή και περισσότερων καμερών είναι απαραίτητο στοιχείο ενός συστήματος τρισδιάστατων γραφικών. Η οπτικοποίηση της τρισδιάστατης σκηνής προϋποθέτει την κατάλληλη τοποθέτηση μίας κάμερας, η οποία καθορίζει την οπτική γωνία από την οποία ζωγραφίζεται η σκηνή. Μέσω της κάμερας γίνεται η προβολή του τρισδιάστατου χώρου σε εικόνα δύο διαστάσεων, που καταλήγει στην οθόνη του υπολογιστή. Επιπλέον, η πλοήγηση, που αποτελεί τη βασικότερη μορφή διάδρασης των χρηστών στα εικονικά περιβάλλοντα, προκαλεί επί της ουσίας έναν συνεχόμενο μετασχηματισμό της κάμερας. Αυτό συμβαίνει τόσο στην απεικόνιση πρώτου προσώπου, όπου η κινούμενη κάμερα υποκαθιστά τα «μάτια» του χρήστη, όσο και στην τρίτου προσώπου, όπου η κάμερα ακολουθεί την κινούμενη ενσάρκωση.

Οι κάμερες τοποθετούνται και αυτές στο σύστημα συντεταγμένων της σκηνής και αποκτούν μια θέση και έναν προσανατολισμό βάσει των οποίων γίνεται η προβολή της σκηνής. Κατά συνέπεια συνδέονται και οι κάμερες με κάποιον γεωμετρικό μετασχηματισμό ο οποίος ρυθμίζει τη μετατόπιση και περιστροφή τους (η αλλαγή κλίμακας δεν έχει νόημα στην περίπτωση μιας κάμερας). Σε κάποιες περιπτώσεις (π.χ. απεικόνιση τρίτου προσώπου) ο μετασχηματισμός αυτός μπορεί να συνδέεται και με τον μετασχηματισμό άλλου αντικειμένου ή ομάδας αντικειμένων της σκηνής.

Πέρα από τη θέση και τον προσανατολισμό, στις κάμερες ορίζονται και ορισμένα εξειδικευμένα στοιχεία, με σημαντικότερα το πεδίο όρασης (field of view – FOV), και δύο επίπεδα αποκοπής (clipping planes), το κοντινό (near) και το μακρινό (far). Το πεδίο όρασης καθορίζει το «άνοιγμα» της κάμερας σε μοίρες. Τα επίπεδα αποκοπής ορίζουν το ελάχιστο και μέγιστο όριο απόστασης από τη θέση της κάμερας, εντός των οποίων θα ζωγραφίζεται η σκηνή. Συνήθως και η ίδια η προβολή της σκηνής γίνεται πάνω στο κοντινό επίπεδο αποκοπής.

Παρά το γεγονός ότι για την απεικόνιση μιας σκηνής απαιτείται μία κάμερα, σε πολλά τρισδιάστατα περιβάλλοντα ορίζονται παραπάνω κάμερες στη σκηνή. Κάτι τέτοιο έχει συνήθως νόημα όταν θέλουμε να δίνουμε τη δυνατότητα στον χρήστη να αλλάξει άμεσα οπτική, π.χ. αλλαγή οπτικής από πρώτου προσώπου σε τρίτου, προβολή κάτοψης, ειδικές αλληλεπιδράσεις με αντικείμενα για τις οποίες απαιτείται συγκεκριμένη οπτική γωνία κ.ά. Σε ειδικές περιπτώσεις μπορεί να απαιτείται η ταυτόχρονη απεικόνιση της εξόδου παραπάνω από μίας καμερών, όπως για παράδειγμα στη σκηνοθετική τακτική picture-in-picture, βάσει της οποίας σε ένα μικρό πλαίσιο εμφανίζεται μια διαφορετική απεικόνιση της σκηνής.

2.6 Άλλες τεχνικές

Μέχρι στιγμής έχουμε παρουσιάσει ορισμένα από τα βασικά στοιχεία των τρισδιάστατων περιβαλλόντων και αντίστοιχες τεχνικές για τη ρεαλιστικότερη απόδοσή τους. Πέρα από αυτά όμως, έχει αναπτυχθεί και ένας αριθμός από άλλες, εξειδικευμένες τεχνικές οι οποίες χρησιμοποιούνται για την καλύτερη οπτικοποίηση

φαινομένων που δεν είναι εύκολο να αναπαρασταθούν με την πολυγωνική γεωμετρία. Στη συνέχεια θα εξετάσουμε τρεις από αυτές: τις υφές τύπου billboard, τα μαλακά αντικείμενα (soft objects) και τα σωματίδια (particles).

Τα billboards είναι επίπεδες επιφάνειες με υφή οι οποίες στρέφονται πάντα προς τη μεριά του παρατηρητή. Αυτό πρακτικά σημαίνει ότι ανεξαρτήτως γωνίας παρατήρησης η εικόνα της υφής δεν φαίνεται ποτέ παραμορφωμένη. Αλλάζει μόνο το μέγεθός της ανάλογα με την απόσταση. Μια τέτοια πρακτική είναι εξαιρετικά χρήσιμη όταν θέλει κάποιος να αναπαραστήσει αντικείμενα για τα οποία δεν παίζει ιδιαίτερο ρόλο η γωνία παρατήρησης αλλά είναι εξαιρετικά δύσκολο να αναπαρασταθούν με τη χρήση πολυγώνων, π.χ. δέντρα και φυτά. Προφανώς σε μια σκηνή στην οποία περιλαμβάνονται δεκάδες δέντρα είναι πολύ πιο ελαφριά υπολογιστικά η απεικόνισή της αν κάθε δέντρο απεικονίζεται με μία μόνο επιφάνεια (δηλαδή δύο τρίγωνα) σε σχέση με μια περίπλοκη γεωμετρική απεικόνιση εκατοντάδων πολυγώνων. Ακόμα πιο επιτακτική είναι η χρήση Billboards σε περιπτώσεις που θέλουμε να απεικονίσουμε πλήθος, π.χ. τους θεατές ενός ποδοσφαιρικού αγώνα (Aubel κ.ά., 2000). Οι υφές των Billboards μπορούν να προέρχονται από φωτογραφίες πραγματικών αντικειμένων, να περιλαμβάνουν διαφάνεια και να είναι κινούμενες. Τέλος, μπορεί κάποιος να αλλάξει την υφή ενός Billboard ανάλογα με τη γωνία παρατήρησης, χρησιμοποιώντας τα έτσι ακόμα και για την απεικόνιση προσανατολισμένων αντικειμένων. Οι χαρακτήρες στα πρώτα τρισδιάστατα παιχνίδια πρώτου προσώπου, όπως το Doom, απεικονίζονταν με βάση αυτή την τεχνική.

Τα γεωμετρικά μοντέλα που έχουμε περιγράψει μέχρι τώρα έχουν μια σαφώς καθορισμένη γεωμετρία επιφάνειας και οι γεωμετρικοί μετασχηματισμοί μπορούν μόνο να προκαλέσουν αλλαγή κλίμακας σε αυτά. Στην πραγματικότητα όμως δεν είναι όλα τα αντικείμενα αυτής της μορφής. Υπάρχουν μαλακά αντικείμενα, των οποίων η γεωμετρία μπορεί να παραμορφωθεί σε περιοχές στις οποίες τους ασκείται δύναμη. Σκεφτείτε, για παράδειγμα, ένα σφουγγάρι το οποίο σφίγγουμε και στη συνέχεια απελευθερώνουμε. Το δέρμα, τα ρούχα, τα μαλλιά ανήκουν όλα σε αυτές τις περιπτώσεις, και μάλιστα ο τρόπος με τον οποίο παραμορφώνονται (λόγω αέρα, βαρύτητας ή άλλων εξωτερικών δυνάμεων) είναι εξαιρετικά περίπλοκος. Για να καλυφθούν περιπτώσεις σαν και αυτές, χρησιμοποιούνται γεωμετρικά μοντέλα που δεν περιγράφονται απλώς από ένα σύνολο πολυγώνων. Περιγράφονται από μαθηματικές εξισώσεις οι οποίες καθορίζουν τη μορφή του αντικειμένου και από τις εξισώσεις αυτές δημιουργούνται τα τελικά πολύγωνα που απεικονίζουν την εξωτερική επιφάνεια. Μέσω αυτών των εξισώσεων μπορεί, για παράδειγμα, να προσδιοριστεί η ελαστική συμπεριφορά ενός αντικειμένου.

Υπάρχουν και περιπτώσεις που θέλουμε να απεικονίσουμε αντικείμενα όχι απλά ελαστικά ή εύπλαστα αλλά με διαρκώς μεταβαλλόμενη μορφή, όπως υγρά και αέρια. Στην περίπτωση αυτή χρησιμοποιούνται τα λεγόμενα «σωματίδια» (particles), δηλαδή σημειακά αντικείμενα τα οποία συνήθως παράγονται δυναμικά, τοποθετούνται στη σκηνή, κινούνται βάσει κάποιων προκαθορισμένων κανόνων (π.χ. κινηματική) και απεικονίζονται μέσω ειδικών shaders. Η απεικόνισή τους γίνεται πολλές φορές με τρόπο τέτοιο ώστε ομάδες γειτονικών σωματιδίων να αποκτούν ενιαία μορφή. Η τεχνική αυτή χρησιμοποιείται για να μοντελοποιήσει την απεικόνιση και κίνηση σταγόνων νερού, φωτιάς, εκρήξεων, σύννεφων και άλλων φυσικών φαινομένων.

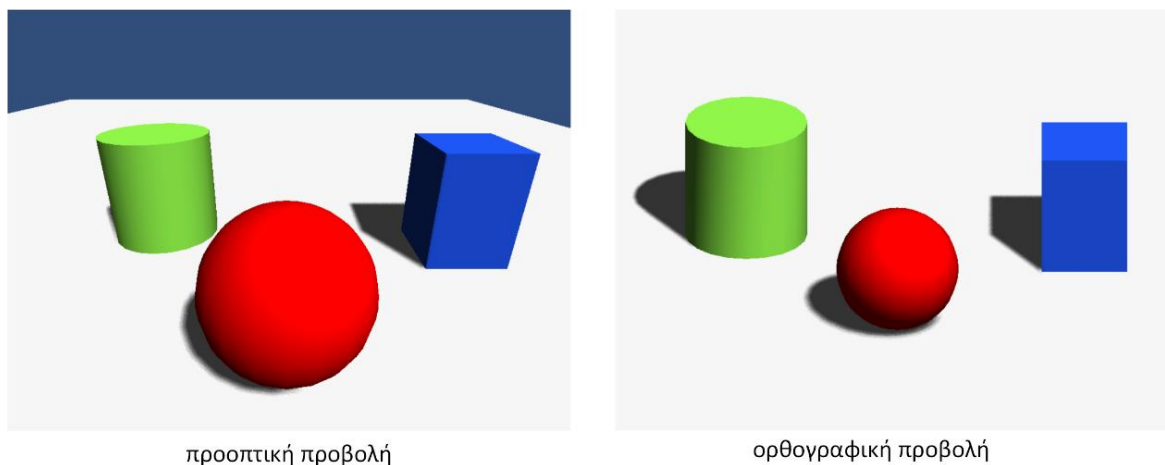
Τα μαλακά αντικείμενα και τα σωματίδια θα παρουσιαστούν αναλυτικότερα στο επόμενο κεφάλαιο, όπου θα ασχοληθούμε με τη συνθετική κίνηση στους εικονικούς κόσμους.

2.7 Διαδικασία απεικόνισης

Η διαδικασία απεικόνισης μιας τρισδιάστατης σκηνής στην οθόνη εκτελείται σε τρία βασικά στάδια: Αρχικά μετασχηματίζεται το σύστημα συντεταγμένων κόσμου σε σύστημα συντεταγμένων προβολής. Με βάση τη θέση, τον προσανατολισμό της κάμερας και το είδος προβολής (προοπτική ή ορθογραφική) υπολογίζεται ένας γεωμετρικός μετασχηματισμός που τοποθετεί τα αντικείμενα στο σύστημα συντεταγμένων της κάμερας. Το επόμενο στάδιο είναι η περικοπή των τμημάτων της σκηνής που είναι εκτός πεδίου όρασης. Η διαδικασία αυτή μπορεί να γίνει αρκετά σύνθετη, καθώς υπάρχουν διάφορα είδη περικοπής που μπορούν να εφαρμοστούν, αλλά μειώνει δραστικά τον χρόνο απεικόνισης βελτιώνοντας τη συνολική απόδοση του περιβάλλοντος. Το τελικό στάδιο είναι αυτό της προβολής, κατά την οποία τα τρίγωνα των αντικειμένων της σκηνής προβάλλονται στην οθόνη με τρόπο τέτοιο ώστε οι κοντινότερες επιφάνειες να επικαλύπτουν τις πιο μακρινές, και για κάθε εικονοστοιχείο που ζωγραφίζεται υπολογίζεται το τελικό του χρώμα βάσει των αλγορίθμων φωτοσκίασης που περιγράψαμε πιο πάνω.

Στα γραφικά υπολογιστών συνηθίζονται δύο είδη προβολής του τρισδιάστατου χώρου στο επίπεδο: η προοπτική και η ορθογραφική. Η προοπτική προβολή μοιάζει με τον τρόπο που απεικονίζεται στα μάτια μας ο κόσμος. Το μέγεθος των προβαλλόμενων αντικειμένων εξαρτάται από την απόστασή τους από το σημείο

παρατήρησης, και οι παράλληλες ευθείες όταν προβάλλονται προοπτικά παύουν να είναι παράλληλες αλλά συγκλίνουν σε κάποιο σημείο φυγής. Αντίθετα, στην ορθογραφική προβολή τα αντικείμενα διατηρούν το σχετικό τους μέγεθος και οι γραμμές παραμένουν παράλληλες. Η ορθογραφική προβολή χρησιμοποιείται κυρίως στο μηχανολογικό και στο αρχιτεκτονικό σχέδιο για την πληρέστερη απόδοση της μορφής των αντικειμένων και του χώρου και των σχετικών αποστάσεων. Από την άλλη μεριά, η προοπτική προβολή, καθώς είναι πιο κοντά στον τρόπο με τον οποίο αντιλαμβανόμαστε οι άνθρωποι το περιβάλλον, προσδίδει περισσότερο ρεαλισμό στο τελικό αποτέλεσμα. Για τον λόγο αυτόν, η συντριπτική πλειοψηφία των εικονικών κόσμων βασίζεται στην προοπτική προβολή. Ιδιαίτερα στην περίπτωση της απεικόνισης πρώτου προσώπου είναι η μοναδική επιλογή, ώστε να μπορεί ο χρήστης να αντιλαμβάνεται τη σχετική του θέση στο περιβάλλον. Στην εικόνα 2.4 φαίνεται η ίδια σκηνή σε προοπτική (αριστερά) και ορθογραφική (δεξιά) προβολή.



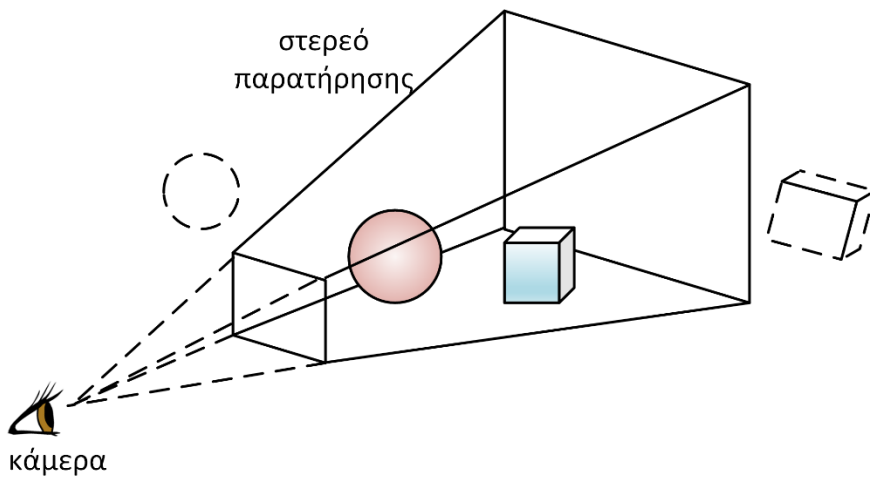
Εικόνα 2.14 Προοπτική (αριστερά) και ορθογραφική (δεξιά) προβολή της ίδιας σκηνής στο Unity.

Για την αλλαγή του συστήματος συντεταγμένων, αρκεί να γνωρίζουμε τη θέση και τον προσανατολισμό της κάμερας και να υπολογίσουμε τον πίνακα μετασχηματισμού. Επί της ουσίας πρόκειται για τον αντίστροφο μετασχηματισμό αυτού που έχει η κάμερα στο σύστημα συντεταγμένων κόσμου. Δηλαδή εφαρμόζουμε μία μετατόπιση κατά $-T$ και μία περιστροφή κατά $-R$ αν T η μετατόπιση και R η περιστροφή της κάμερας στον κόσμο. Το αποτέλεσμα αυτού του μετασχηματισμού θα είναι να επανατοποθετηθούν όλα τα αντικείμενα της σκηνής με τέτοιο τρόπο ώστε η αρχή των αξόνων να βρίσκεται στο σημείο της κάμερας και ο άξονας $-Z$ να συμπίπτει με τη διεύθυνση παρατήρησης της κάμερας.

Το τμήμα της σκηνής το οποίο τελικά θα προβληθεί στην οθόνη βρίσκεται εντός μιας κόλυρης πυραμίδας, του λεγόμενου *στερεού παρατήρησης* (*viewing frustum*). Η πυραμίδα αυτή ορίζεται από το κοντινό και μακρινό επίπεδο αποκοπής, τα οποία αναφέραμε και παραπάνω ως παραμέτρους που σχετίζονται με την κάμερα. Αυτό σημαίνει ότι όσα αντικείμενα ή τμήματα αντικειμένων δεν περιλαμβάνονται μέσα σε αυτό το στερεό δεν θα συμμετέχουν στην τελική εικόνα και επομένως δεν υπάρχει λόγος να συν-υπολογιστούν κατά τη διαδικασία προβολής. Αυτό που συμβαίνει λοιπόν σε αυτό το στάδιο είναι η διαδικασία της περικοπής, δηλαδή της «εξαίρεσης» όλων των τριγώνων για τα οποία είμαστε σίγουροι ότι δεν επηρεάζουν την προβολή. Υπάρχουν δύο βασικά είδη περικοπής:

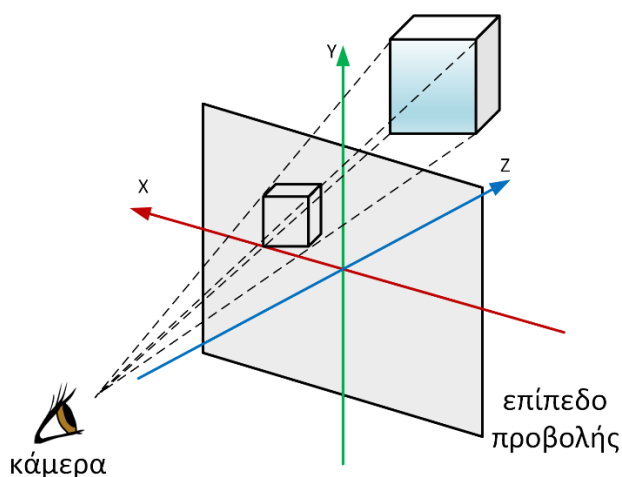
1. περικόπτονται οι επιφάνειες των οποίων η «εξωτερική» όψη δεν είναι προσανατολισμένη προς την κάμερα, και
2. περικόπτονται όλες οι επιφάνειες που βρίσκονται εκτός του στερεού παρατήρησης.

Η πρώτη περίπτωση είναι αρκετά εύκολη στον εντοπισμό. Αρκεί να βρεθεί η γωνία μεταξύ του διανύσματος παρατήρησης της κάμερας και του κανονικού διανύσματος της επιφάνειας. Αν η γωνία είναι μεγαλύτερη των 180° τότε η κάμερα βλέπει την πίσω όψη της επιφάνειας και επομένως η επιφάνεια δεν είναι ορατή και αποκόπτεται.



Εικόνα 2.15 Το στερεό παρατήρησης μιας κάμερας.

Για την αποκοπή των τριγώνων εκτός του στερεού παρατήρησης υπάρχουν διάφορες προσεγγίσεις, ανάλογα και με το μέγεθος της σκηνής. Η πιο απλοϊκή προσέγγιση είναι να επισκεφθούμε μία μία τις κορυφές του κάθε αντικειμένου και να εξετάσουμε αν βρίσκονται εντός ή εκτός του στερεού. Το υπολογιστικό κόστος θα μπορούσε να μειωθεί αν ελέγχαμε συνολικά τα όρια των αντικειμένων βάσει κάποιου ελάχιστου περικλειόμενου στερεού, π.χ. σφαίρα (bounding sphere). Τα στερεά αυτά μπορούν να υπολογιστούν πριν την εκκίνηση του προγράμματος απλά βρίσκοντας τις μέγιστες και ελάχιστες τιμές των κορυφών του αντικειμένου, και μπορούν να επιταχύνουν πολύ την παραπάνω διαδικασία. Αν, για παράδειγμα, η ελάχιστη περικλειόμενη σφαίρα ενός αντικειμένου βρίσκεται συνολικά εκτός του στερεού παρατήρησης, δεν υπάρχει λόγος να εξεταστούν μία μία οι κορυφές του. Σε μεγάλες σκηνές η παραπάνω λογική μπορεί να επεκταθεί και με δομές διαμέρισης του χώρου, μέσω των οποίων ολόκληρες ομάδες αντικειμένων μπορούν να αποκοπούν λόγω της σχετικής τους θέσης σε σχέση με το οπτικό πεδίο της κάμερας. Ένας δημοφιλής αλγόριθμος διαμέρισης του χώρου είναι τα δέντρα *BSP*, με τη χρήση των οποίων περικόπτονται όχι μόνο τα αντικείμενα εκτός στερεού παρατήρησης αλλά και όσα δεν είναι ορατά λόγω παρεμπόδισης από κοντινότερα αντικείμενα. (Fuchs κ.ά., 1980) Λόγω του τελευταίου χαρακτηριστικού ο αλγόριθμος είναι περισσότερο χρήσιμος σε σκηνές εσωτερικού χώρου, όπου οι τοίχοι των δωματίων παρεμποδίζουν μεγάλο μέρος της υπόλοιπης σκηνής.



Εικόνα 2.16 Προοπτική προβολή σε δύο διαστάσεις.

Το τελευταίο στάδιο της διαδικασίας είναι η προβολή των πολυγώνων σε δύο διαστάσεις και η απόδοση χρώματος σε αυτές με τρόπο που να αποκρύπτονται οι παρεμποδιζόμενες επιφάνειες. Η διαδικασία της προοπτικής προβολής είναι εξαιρετικά απλή, όπως φαίνεται και στην εικόνα 2.16. Επί της ουσίας, για κάθε κορυφή χρησιμοποιούνται μόνο οι x και y συντεταγμένες, των οποίων όμως η κλίμακα αλλάζει ανάλογα με την

απόστασή τους από το σημείο παρατήρησης. Αν διαιρέσουμε τις τιμές των x και y με την απόσταση (z) τότε τα σημεία με τη μικρότερη απόσταση θα μεγαλώνουν δείχνοντας και τα αντικείμενα αντίστοιχα μεγάλα, ενώ όσο μεγαλώνει η απόσταση τους τόσο τα αντικείμενα θα συγκλίνουν προς τον ορίζοντα. Για τον τύπο της προβολής θα συνυπολογίσουμε και την απόσταση d μεταξύ του σημείου παρατήρησης και του κοντινού επιπέδου αποκοπής. Ο τελικός τύπος θα είναι:

$$x_p = \frac{xd}{z+d}$$

$$y_p = \frac{yd}{z+d}$$

Η προβολή των κορυφών των πολυγώνων σε δύο διαστάσεις και η μετατροπή τους σε συντεταγμένες οθόνης δεν αρκεί για την παραγωγή της τελικής εικόνας. Θα πρέπει να διασφαλίσουμε επιπλέον ότι οι κοντινότερες επιφάνειες επικαλύπτουν τις πιο μακρινές. Η πιο απλή λύση είναι να ταξινομήσουμε τα πολύγωνα από το μακρινότερο προς το κοντινότερο και να τα ζωγραφίσουμε με αυτή τη σειρά. Ο αλγόριθμος αυτός ονομάζεται *αλγόριθμος του ζωγράφου* (*painter's algorithm*) επειδή αυτός είναι συνήθως και ο τρόπος που οι ζωγράφοι αποδίδουν τις σκηνές. Επειδή όμως ο αλγόριθμος του ζωγράφου έχει το υπολογιστικό κόστος της ταξινόμησης, σήμερα χρησιμοποιείται συχνότερα ο αλγόριθμος *z-Buffer*, (Catmull, 1974) ο οποίος λειτουργεί και με αυθαίρετη σειρά πολυγώνων. Σύμφωνα με τον αλγόριθμο αυτόν όταν προβάλλουμε και ζωγραφίζουμε κάθε τρίγωνο στον πίνακα από εικονοστοιχεία που θα αποτελέσει την τελική εικόνα μας καταγράφουμε για κάθε εικονοστοιχείο και την αντίστοιχη τιμή βάθους (Z) της επιφάνειας η οποία προβλήθηκε στο σημείο αυτό. Κατά συνέπεια, όταν πρόκειται να ζωγραφιστεί ένα εικονοστοιχείο ελέγχεται πρώτα η υπάρχουσα τιμή του z σε σχέση με τη νέα τιμή που πρόκειται να προστεθεί. Αν η υπάρχουσα είναι μικρότερη, δηλαδή το χρώμα που υπάρχει σε αυτό το σημείο προέρχεται από σημείο που είναι κοντινότερο στο σημείο παρατήρησης, τότε το χρώμα δεν αλλάζει. Διαφορετικά αντικαθίσταται από το νέο χρώμα και τη νέα τιμή του z .

2.8 Στερεοσκοπία

Όπως αναφέρθηκε και στην εισαγωγή, τα παραδοσιακά συστήματα εικονικής πραγματικότητας βασίζονται στη στερεοσκοπία, δηλαδή στην παραγωγή ανεξάρτητων εικόνων σε κάθε μάτι ώστε να δημιουργείται η αίσθηση του βάθους. Στερεοσκοπικές απεικονίσεις όμως μπορούν να γίνουν και σε στους εικονικούς κόσμους μέσω ειδικού υλικού, όπως τα στερεοσκοπικά γυαλιά ή τα κράνη εικονικής πραγματικότητας χαμηλού κόστους που έχουν κάνει την εμφάνισή τους τα τελευταία χρόνια (π.χ. Oculus Rift).

Για την παραγωγή στερεοσκοπικής απεικόνισης αρκεί να εκτελεστεί η διαδικασία προοπτικής προβολής που περιγράφηκε στην προηγούμενη υποενότητα δύο φορές, μία για κάθε μάτι. Υπολογίζονται δύο ανεξάρτητα σημεία προβολής που ισαπέχουν από το σημείο τοποθέτησης της κάμερας και τοποθετούνται αριστερά και δεξιά σε σχέση με τη διεύθυνση παρατήρησης. Η απόσταση των δύο σημείων θα πρέπει να ταιριάζει με την απόσταση μεταξύ των ματιών του χρήστη. Η μέση απόσταση ματιών είναι γύρω στα 63.5 mm. Στη συνέχεια, ανάλογα με το είδος του υλικού το σύστημα διασφαλίζει ότι η εικόνα του αριστερού σημείου προβολής θα σταλεί στο αριστερό μάτι και αντίστοιχα στο δεξί. Αυτή η ελαφριά διαφοροποίηση μεταξύ των δύο εικόνων μιμείται τον τρόπο με τον οποίο τα ανθρώπινα μάτια στέλνουν την εικόνα του περιβάλλοντος στον εγκέφαλο και επιτυγχάνεται η αίσθηση της στερεοσκοπίας, δηλαδή αντιλαμβανόμαστε καλύτερα το βάθος των αντικειμένων από τις διαφορές στις δύο εικόνες.

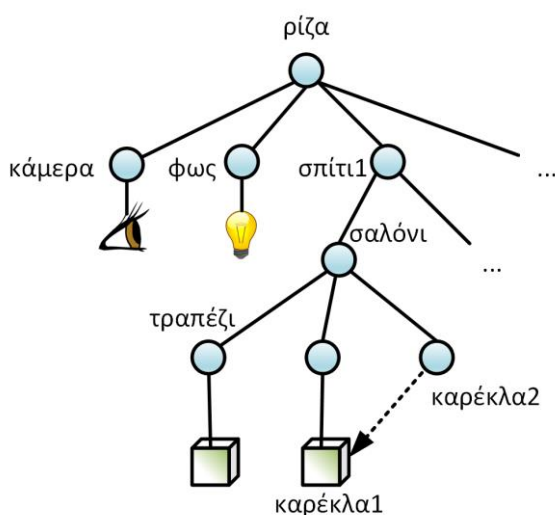
3 Δημιουργία 3D σκηνών

Η δημιουργία μιας τρισδιάστατης σκηνής που θα αξιοποιηθεί από κάποιον εικονικό κόσμο είναι μια διαδικασία που περιλαμβάνει τόσο τον καθορισμό και τη σχεδίαση των στοιχείων που την αποτελούν όσο και την κατάλληλη διαμόρφωσή τους για την υποστήριξη των απαιτούμενων κινήσεων και αλληλεπιδράσεων. Στα σύγχρονα συστήματα γραφικών πραγματικού χρόνου, οι σκηνές δεν είναι απλά συλλογές από πολυγωνικά αντικείμενα. Συνήθως έχουν μια δένδρική δομή που ονομάζεται *γράφος σκηνής* (Strauss & Carey, 1992), με βάση την οποία τα αντικείμενα ομαδοποιούνται και μπορούν να συμπεριφέρονται ως σύνολο ή και

κληρονομούν τους μετασχηματισμούς από άλλα αντικείμενα με αποτέλεσμα η θέση τους να επηρεάζεται από αυτά. Οι δομές αυτές βοηθούν τόσο στην καλύτερη οργάνωση των στοιχείων της σκηνής όσο και στον καθορισμό των δυνατοτήτων κίνησης των επιμέρους αντικειμένων. Ως προς την ίδια τη μορφή των αντικειμένων, η διαμόρφωση της γεωμετρίας της επιφάνειας μπορεί να γίνει με τη χρήση εξειδικευμένων εργαλείων τρισδιάστατης μοντελοποίησης και με την αξιοποίηση διάφορων σχετικών τεχνικών που ποικίλλουν ανάλογα με το είδος του αντικειμένου που θέλουμε να μοντελοποιήσουμε. Αν η γεωμετρία του αντικειμένου ή χώρου είναι εξαιρετικά περίπλοκη, αν θέλουμε να δώσουμε μεγαλύτερη έμφαση στη λεπτομέρεια ή αν, γενικότερα, θέλουμε να επιταχύνουμε τη διαδικασία παραγωγής των τρισδιάστατων μοντέλων, υπάρχει και η δυνατότητα ψηφιοποίησης, δηλαδή σύλληψης της γεωμετρίας αντικειμένων του φυσικού κόσμου και μετατροπής τους σε πολυγωνικά μοντέλα. Τέλος, επειδή ο παράγοντας του χρόνου απεικόνισης είναι κρίσιμος σε συστήματα πραγματικού χρόνου, όπως έχουμε ήδη αναφέρει, θα πρέπει να γίνει ένας συμβιβασμός μεταξύ της ποιότητας απεικόνισης και του όγκου των πολυγώνων που χρησιμοποιούνται για να περιγράψουν τη σκηνή. Για τον σκοπό αυτόν υπάρχουν εργαλεία και τεχνικές βελτιστοποίησης των μοντέλων για εφαρμογές πραγματικού χρόνου.

3.1 Γράφος σκηνής

Ο γράφος σκηνής είναι μια ιεραρχική οργάνωση των αντικειμένων (οπτικών ή μη) που περιλαμβάνονται στον εικονικό κόσμο. Αποτελεί μια δενδρική δομή αποτελούμενη από κόμβους που συνδέονται μεταξύ τους με ακμές. Ο κορυφαίος κόμβος είναι η ρίζα, η οποία περιλαμβάνει όλο τον υπόλοιπο γράφο και αναπαριστά τη σκηνή στο σύνολό της. Συνδέεται προς τα κάτω με πολλαπλούς εσωτερικούς κόμβους, οι οποίοι με τη σειρά τους έχουν ως απογόνους άλλους εσωτερικούς ή εξωτερικούς κόμβους. Οι εξωτερικοί κόμβοι, που ονομάζονται και φύλλα, δεν έχουν άλλους απογόνους. Αναπαριστούν τρισδιάστατα αντικείμενα μαζί με τα χαρακτηριστικά τους (γεωμετρία, υλικό, συμπεριφορά) ή και άλλα χωροθετημένα αντικείμενα που συμμετέχουν στη σκηνή, όπως κάμερες, φώτα, σημεία εκπομπής ήχων κ.λπ. Οι εσωτερικοί κόμβοι συνήθως αναπαριστούν γεωμετρικούς μετασχηματισμούς, ο οποίοι τοποθετούν τους απογονικούς κόμβους στον χώρο σε σχέση με τον κόμβο κορυφής ή σε σχέση με άλλα αντικείμενα. Οποιαδήποτε αλλαγή στον μετασχηματισμό ενός εσωτερικού κόμβου επηρεάζει όλους τους απογονικούς.



Εικόνα 2.17 Τμήμα ενός γράφου σκηνής.

Ο γράφος σκηνής είναι εξαιρετικά χρήσιμος για την ομαδοποίηση των στοιχείων της σκηνής με βάση κάποια «λογική» ή «χωρική» σύνδεση. Αν το σύνολο ορισμένων επιμέρους χωροθετημένων αντικειμένων περιγράφει ένα πιο σύνθετο αντικείμενο, τότε μπορούμε να ομαδοποιήσουμε τα αντικείμενα αυτά τοποθετώντας τα κάτω από έναν ενιαίο μετασχηματισμό στον γράφο σκηνής. Για παράδειγμα, θα μπορούσαμε να ομαδοποιήσουμε το κτίριο και τα έπιπλα ενός σπιτιού σε ένα «λογικό» αντικείμενο σπίτι. Μια τέτοια ενέργεια θα είχε δύο πλεονεκτήματα: Αφενός, κατά τη φάση σχεδίασης και επεξεργασίας της σκηνής θα ήταν πολύ πιο εύκολο να

εντοπίσουμε συγκεκριμένα αντικείμενα μέσω πλοήγησης στις ιεραρχικές ομάδες της σκηνης (π.χ. γειτονιά > σπίτι > δωμάτιο), σε σχέση με την αναζήτηση σε όλο το πλήθος των αντικειμένων που μπορεί να είναι και μερικές εκατοντάδες. Αφετέρου, αν θέλουμε να μετακινήσουμε, περιστρέψουμε ή ακόμα και να δημιουργήσουμε παραπάνω αντίγραφα της ομάδας αντικειμένων (π.χ. του σπιτιού), μπορούμε μετά την ομαδοποίησή του να το διαχειριστούμε ως ένα ενιαίο αντικείμενο, γεγονός που διευκολύνει σημαντικά τον χειρισμό του. Εκτός από τη «λογική» ομαδοποίηση, σε περιπτώσεις μεγάλων σκηνών, ο γράφος σκηνης χρησιμοποιείται και για τη «χωρική» ομαδοποίηση των αντικειμένων, με αποτέλεσμα ολόκληρες ομάδες αντικειμένων να εξαιρούνται γρήγορα από τη διαδικασία απεικόνισης, αν ο χώρος που καλύπτουν είναι εκτός του οπτικού πεδίου. Στόχος μιας τέτοιας ομαδοποίησης είναι, προφανώς, η βελτίωση του χρόνου απεικόνισης και της συνολικής απόδοσης του κόσμου. Η εικόνα 2.17 απεικονίζει ένα παράδειγμα γράφου σκηνης.

Ένας ακόμα λόγος υπέρ της χρήσης ιεραρχικής δομής μετασχηματισμών για την περιγραφή μιας σκηνης είναι και η διευκόλυνση σύνθετων κινήσεων που συνδέουν δύο ή περισσότερα αντικείμενα μεταξύ τους. Ας σκεφτούμε, για παράδειγμα, ένα αυτοκίνητο το οποίο κινείται πάνω στον δρόμο. Οι ρόδες του αυτοκινήτου, ως μέρη αυτού, ακολουθούν την κίνηση του αυτοκινήτου, άρα επηρεάζονται από οποιοσδήποτε αλλαγές στη θέση και στον προσανατολισμό του. Όμως οι μπροστινές ρόδες μπορούν και στρίβουν ανεξάρτητα από το υπόλοιπο αυτοκίνητο. Συνεπώς, η τελική τοποθέτηση μιας ρόδας εξαρτάται τόσο από τον μετασχηματισμό του αυτοκινήτου όσο και από τον μετασχηματισμό της ίδιας της ρόδας. Χρειάζεται ο συνδυασμένος μετασχηματισμός για τον προσδιορισμό της τελικής μετατόπισης και του προσανατολισμού του. Αυτός ο υπολογισμός μπορεί να γίνει εύκολα χάρη στον γράφο σκηνης, καθώς κάθε αντικείμενο στο δέντρο «κληρονομεί» τους μετασχηματισμούς όλων των αντικειμένων που βρίσκονται πάνω από αυτό στην ιεραρχία και εφαρμόζεται πάνω του το γινόμενο τους. Επομένως, αν στο παράδειγμά μας η γεωμετρία του αυτοκινήτου έχει μοιραστεί στο κυρίως σώμα και στις ρόδες, και οι ρόδες έχουν τον δικό τους μετασχηματισμό, μπορούμε εύκολα να προκαλέσουμε τη συνδυασμένη κίνηση αυτοκινήτου και ρόδας που περιγράψαμε. Για να λειτουργήσει αυτή η τεχνική, θα πρέπει να γνωρίζει ο σχεδιαστής εκ των προτέρων ποια τμήματα ενός σύνθετου αντικειμένου εκτελούν ανεξάρτητες κινήσεις και να τα μοντελοποιήσει ως ξεχωριστές γεωμετρίες, ώστε να αποκτήσουν τον δικό τους μετασχηματισμό στον γράφο σκηνης.

Με τον γράφο σκηνης μπορούν να υποστηριχθούν και πολλαπλά στιγμιότυπα μιας γεωμετρίας αντικειμένου. Πράγματι, παραπάνω από ένας κόμβοι μετασχηματισμού στον γράφο της σκηνης μπορούν να αναφέρονται στην ίδια γεωμετρία, με αποτέλεσμα να εμφανίζονται πολλαπλά αντίγραφα του αντικειμένου σε διάφορα σημεία της σκηνης, ενδεχομένως και σε διαφορετικά μεγέθη. Υπάρχουν πολλές περιπτώσεις στις οποίες μπορεί να είναι χρήσιμη η τοποθέτηση πολλαπλών αντιγράφων σε μια σκηνή, όπως, για παράδειγμα, τα δέντρα ενός δρόμου, οι καρέκλες ενός συνεδριακού χώρου ή τα σπίτια μιας γειτονιάς (βλ. και καρέκλες στην εικόνα 2.17). Το προφανές πλεονέκτημα αυτής της προσέγγισης είναι ότι δεν επιβαρύνεται η μνήμη του υπολογιστικού συστήματος με την αποθήκευση των πολλαπλών αντικειμένων, καθώς η αποθηκευμένη γεωμετρία είναι μόνο μία· τα αντικείμενα απεικονίζονται εφαρμόζοντας διαφορετικούς μετασχηματισμούς στην ίδια γεωμετρία.

Κατά τη διάρκεια εκτέλεσης του εικονικού κόσμου είναι πολύ πιθανό ο γράφος σκηνης να μεταβάλλεται. Πέρα από τις πιο προφανείς μεταβολές, όπως είναι η προσθήκη ή η αφαίρεση ενός αντικειμένου από τη σκηνή, υπάρχουν και περιπτώσεις στις οποίες μπορεί να μεταβληθεί η θέση ενός αντικειμένου στην ιεραρχία. Ένα πολύ χαρακτηριστικό τέτοιο παράδειγμα είναι η περίπτωση στην οποία ένας χαρακτήρας πιάνει ένα αντικείμενο. Από τη στιγμή εκείνη και μέχρι να το απελευθερώσει, η θέση του αντικειμένου επηρεάζεται από την κίνηση του χεριού του χαρακτήρα. Σε επίπεδο γράφου σκηνης, αυτή η αλλαγή μπορεί να αποδοθεί αν το αντικείμενο γίνει απογονικός κόμβος στον μετασχηματισμό του χεριού του χαρακτήρα ώστε να «κληρονομεί» τον μετασχηματισμό του. Όταν ο χαρακτήρας απελευθερώσει το αντικείμενο, τότε αυτό θα αποκοπεί από την ιεραρχία του χεριού του και θα συνδεθεί ξανά με τη «ρίζα» του γραφήματος, έχοντας αποκτήσει κατάλληλο μετασχηματισμό ώστε να παραμείνει στη θέση στην οποία αφέθηκε.

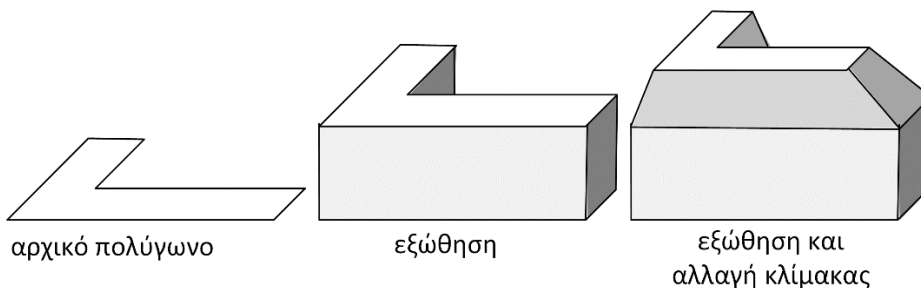
Ο γράφος σκηνης διασχίζεται σε δύο κατευθύνσεις πριν από κάθε ανανέωση εικόνας, τροποποιώντας ανάλογα και τις τιμές των χαρακτηριστικών των κόμβων του. Η διαδικασία είναι αναδρομική, από τη ρίζα προς τα δέντρα και πίσω προς τη ρίζα. Η διάσχιση από πάνω προς τα κάτω αφορά κυρίως τον υπολογισμό του τελικού συνδυασμένου μετασχηματισμού του κάθε κόμβου, ενώ η διάσχιση από κάτω προς τα πάνω γίνεται για τον υπολογισμό του συνολικού όγκου (bounding volume) που καταλαμβάνει κάθε κόμβος. Οι όγκοι αυτοί, που περιγράφονται ως περικλειόμενες σφαίρες (bounding sphere) ή παραλληλεπίπεδα (bounding box) και περικλείουν όλους τους αντίστοιχους όγκους των απογονικών κόμβων, χρησιμοποιούνται για την επιτάχυνση των διαδικασιών απεικόνισης και για τον έλεγχο σύγκρουσης μεταξύ αντικειμένων.

3.2 Τεχνικές μοντελοποίησης

Για την κατασκευή μιας τρισδιάστατης σκηνής απαιτείται ο προσδιορισμός της γεωμετρίας των αντικειμένων που συμμετέχουν σε αυτήν, η κατασκευή της ιεραρχικής δομής του γράφου σκηνής και η τελική τοποθέτηση των αντικειμένων στον χώρο. Ο καθορισμός του τύπου των αντικειμένων, η ομαδοποίηση και ιεράρχησή τους στον γράφο σκηνής και η τοποθέτησή τους στο περιβάλλον είναι διεργασίες σχετικά εύκολες, ακόμα και για αρχάριους σχεδιαστές. Το βήμα που παρουσιάζει σαφώς τη μεγαλύτερη δυσκολία είναι η διαδικασία της γεωμετρικής μοντελοποίησης, δηλαδή η διαμόρφωση της γεωμετρίας των αντικειμένων της σκηνής. Υπάρχει βεβαίως πληθώρα διαθέσιμων τρισδιάστατων μοντέλων στο διαδίκτυο τα οποία μπορεί να κατεβάσει κάποιος δωρεάν ή έναντι χαμηλού κόστους, αλλά είναι τόσα πολλά και διαφορετικά τα είδη των αντικειμένων που ενδέχεται να ενταχθούν σε έναν εικονικό κόσμο, που είναι πρακτικά αδύνατο να μην προκύψει η ανάγκη και για τη μοντελοποίηση κάποιων.

Όπως υπονοεί και το όνομα, κατά τη μοντελοποίηση προσπαθούμε να προσδιορίσουμε τη γεωμετρία της επιφάνειας ενός τρισδιάστατου αντικειμένου το οποίο αποτελεί μοντέλο ενός αντίστοιχου φυσικού ή ιδεατού αντικειμένου. Πρέπει, συνεπώς, τη μορφή που έχουμε στο μυαλό μας να μπορέσουμε να την αποτυπώσουμε ως ένα σύνολο πολυγώνων χρησιμοποιώντας τα εργαλεία του υπολογιστή. Όταν κατασκευάζουμε αντικείμενα στον φυσικό κόσμο έχουμε έναν αριθμό από φυσικούς περιορισμούς. Προκύπτουν θέματα επιλογής υλικών, στήριξης, βαρύτητας, αντοχής σε δυνάμεις κ.ά. Αντίστοιχα, έχουν αναπτυχθεί πολλές υποστηρικτικές τεχνικές και εργαλεία κατασκευής ανάλογα με το είδος του αντικειμένου που θέλουμε να δημιουργήσουμε· μπορούμε να κόψουμε επιμέρους τμήματα, να τα κολλήσουμε μεταξύ τους, να τα βιδώσουμε, να τα συνδέσουμε με μεντεσέδες κ.λπ. Στις τρισδιάστατες σκηνές αυτοί οι περιορισμοί δεν υπάρχουν. Υπάρχει απόλυτη ελευθερία ως προς τη μορφή των αντικειμένων και τις μεταξύ τους σχέσεις, κάτι που διευκολύνει τη δημιουργική διαδικασία. Επιπλέον, στις εφαρμογές μοντελοποίησης προσφέρεται και εκεί ένας σημαντικός αριθμός εργαλείων και τεχνικών για την κατασκευή του μοντέλου, πολύ διαφορετικών όμως από τα αντίστοιχα εργαλεία του πραγματικού κόσμου. Η μεγάλη ελευθερία επιλογών και η πληθώρα εργαλείων δυσκολεύουν τον αρχάριο χρήστη να φανταστεί μια πορεία βημάτων για την κατασκευή του τελικού μοντέλου και η καμπύλη εκμάθησης των σχετικών εφαρμογών είναι συνήθως μεγάλη.

Τα τρισδιάστατα μοντέλα περιγράφονται ως συλλογές από πολύγωνα, όπως έχουμε αναφέρει. Όμως στη φύση οι επίπεδες επιφάνειες σπανίζουν· όλες σχεδόν οι μορφές που συναντάμε περιλαμβάνουν καμπύλες. Υπάρχουν μεν κάποια ανθρώπινα τεχνουργήματα τα οποία έχουν μόνο επίπεδες επιφάνειες, όπως, για παράδειγμα οι τοίχοι ή ορισμένα έπιπλα, όμως και στα σχεδιασμένα αντικείμενα οι καμπύλες εμφανίζονται πολύ συχνά για λόγους αισθητικής ή και λειτουργικότητας. Και ενώ η μοντελοποίηση αντικειμένων με επίπεδες επιφάνειες, για παράδειγμα ενός κουτιού, ως σύνολο από πολύγωνα είναι μια σχετικά απλή διαδικασία, η περιγραφή καμπύλων μορφών δεν είναι εξίσου εύκολη. Επομένως, μια βασική πρόκληση της τρισδιάστατης μοντελοποίησης είναι το να μπορέσει ο σχεδιαστής να περιγράψει και να διαχειριστεί καμπύλες επιφάνειες, οι οποίες θα καταλήξουν σε ένα σύνολο πολυγώνων ομαλά φωτοσκιασμένων.



Εικόνα 2.18 Παράδειγμα διαδοχικών εξωθήσεων πολυγώνου.

Μια απλή σχετικά προσέγγιση στην τρισδιάστατη μοντελοποίηση είναι το να ξεκινήσει κάποιος από τη σχεδίαση ενός πολυγώνου σε δύο διαστάσεις και με βάση αυτό να παραχθεί ένα τρισδιάστατο μοντέλο σύμφωνα με κάποιους κανόνες. Η εξώθηση (extrusion) είναι μια συχνή τεχνική που εντάσσεται σε αυτή την κατηγορία. Στην περίπτωση αυτήν το αρχικό πολύγωνο «εξωθείται» στον κάθετο άξονα ως προς το επίπεδο του σε απόσταση που καθορίζει ο σχεδιαστής. Το αποτέλεσμα είναι ότι δημιουργείται ένα αντίγραφο του πολυγώνου στην τελική θέση εξώθησης, οι κορυφές του νέου πολυγώνου συνδέονται με τις αντίστοιχες του

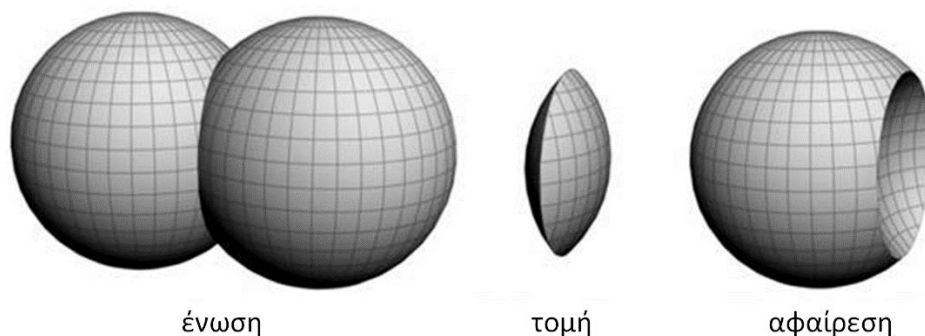
αρχικού και δημιουργούνται όλες οι πλευρικές επιφάνειες σχηματίζοντας τελικά ένα συμπαγές αντικείμενο. Ο σχεδιαστής έχει επιπλέον τη δυνατότητα να μετασχηματίσει το νέο πολύγωνο αλλάζοντάς του τη θέση, την περιστροφή ή και την κλίμακα, ώστε να τροποποιήσει τη μορφή του στερεού. Επιπλέον, η ενέργεια της εξώθησης μπορεί να εφαρμοστεί ξανά στο νέο πολύγωνο, επεκτείνοντας περαιτέρω τη μορφή του στερεού (βλ. εικ. 2.18). Στα περισσότερα εργαλεία τρισδιάστατης μοντελοποίησης μπορεί η διαδικασία αυτή να αυτοματοποιηθεί ακόμα περισσότερο, ορίζοντας ένα μονοπάτι βάσει του οποίου θα γίνουν διαδοχικές εξωθήσεις του αρχικού πολυγώνου.

Όταν ο σχεδιαστής έχει ήδη ένα τρισδιάστατο μοντέλο στη διάθεσή του, για παράδειγμα ένα βασικό στερεό ή ένα μοντέλο που προέκυψε από εξώθηση, μπορεί στη συνέχεια να επεξεργαστεί και τα ίδια τα πολύγωνα του αντικειμένου για να διορθώσει το αποτέλεσμα. Για παράδειγμα, μπορεί να επιλέξει μία ή περισσότερες κορυφές του αντικειμένου και να τις μετατοπίσει ελεύθερα στον χώρο, προκαλώντας αντίστοιχες μεταβολές στα πολύγωνα που περιλαμβάνουν τις κορυφές αυτές. Βεβαίως, αυτού του είδους οι ενέργειες δυσκολεύονται από την ανάγκη μετάφρασης των δύο διαστάσεων εισόδου (ποντίκι) και εξόδου (οθόνη) σε τρεις διαστάσεις. Για τον λόγο αυτόν, οι έμπειροι σχεδιαστές αναγκάζονται να αλλάζουν συχνά απόψεις του τρισδιάστατου μοντέλου, ώστε οι μετατοπίσεις τους να γίνονται από κατάλληλη οπτική γωνία (π.χ. στην κάτοψη του μοντέλου).

Μία διαφορετική προσέγγιση για την κατασκευή τρισδιάστατων μοντέλων που βασίζεται στα προγράμματα σχεδίασης προϊόντων με υπολογιστή (Computer Aided Design –CAD) είναι η στερεά γεωμετρία (Constructive Solid Geometry). Στην περίπτωση αυτήν ο χρήστης κατασκευάζει το αντικείμενο που επιθυμεί τοποθετώντας βασικά στερεά στη σκηνή, όπως κύβο, σφαίρα, κύλινδρο κ.λπ., και συνδυάζοντάς τα για τη δημιουργία πιο σύνθετων μοντέλων. Ο συνδυασμός των στερεών αντικειμένων μπορεί να γίνει με τη χρήση των λεγόμενων Boolean ενεργειών, σύμφωνα με τις οποίες:

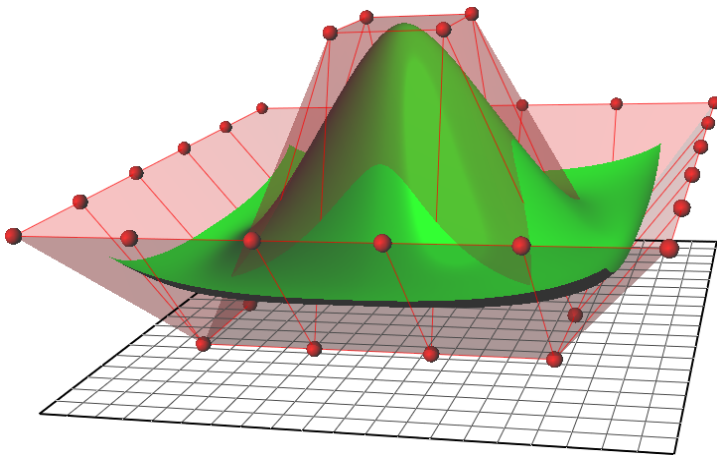
- οι όγκοι δύο στερεών που τέμνονται μπορούν να συνδυαστούν σε ένα ενιαίο μοντέλο (ένωση),
- ο όγκος ενός στερεού μπορεί να αφαιρεθεί από ένα άλλο με το οποίο τέμνεται (αφαίρεση) και
- από δύο στερεά που τέμνονται παραμένει μόνο ο κοινός τους όγκος (τομή).

Η αποδοτική χρήση της προσέγγισης αυτής μπορεί να οδηγήσει αρκετά εύκολα σε περίπλοκα καθημερινά αντικείμενα. Στην εικόνα 2.19 φαίνεται ένα παράδειγμα Boolean ενεργειών μεταξύ δύο σφαιρών.



Εικόνα 2.19 Boolean ενέργειες μεταξύ δύο σφαιρών.

Οι τεχνικές που εξετάσαμε μέχρι τώρα δεν είναι ιδιαίτερα χρήσιμες για την κατασκευή αυθαίρετων καμπύλων επιφανειών και ειδικότερα για την κατασκευή οργανικών μορφών, όπως, για παράδειγμα, οι ανθρώπινες φιγούρες ενός εικονικού κόσμου. Ένα εργαλείο που μπορεί να βοηθήσει προς αυτήν την κατεύθυνση είναι η χρήση καμπύλων επιφανειών με σημεία ελέγχου, όπως είναι οι καμπύλες NURBS. Σύμφωνα με αυτήν την προσέγγιση, ο χρήστης ξεκινάει από μια επίπεδη επιφάνεια, η οποία υποδιαιρείται σε μικρότερα τμήματα (patches). Η επιφάνεια συνδέεται με έναν αριθμό σημείων ελέγχου τα οποία τροποποιούν τις θέσεις των κορυφών του κάθε τμήματος δημιουργώντας καμπυλότητα στην επιφάνεια (βλ. εικ. 2.20). Ο σχεδιαστής μπορεί στη συνέχεια να διαμορφώσει το σχήμα της καμπύλης επιλέγοντας και μετακινώντας τα σημεία ελέγχου.



Εικόνα 2.20 Παράδειγμα επιφάνειας NURBS.

Μία ακόμα ενδιαφέρουσα τεχνική για την κατασκευή σύνθετων και οργανικών μορφών είναι οι επιφάνειες υποδιαίρεσης. Με βάση αυτή την τεχνική, ο σχεδιαστής ξεκινάει από ένα απλό στερεό (με μικρό αριθμό πολυγώνων), το επεξεργάζεται και προοδευτικά επιλέγει επιφάνειες αυτού και τις υποδιαιρεί σε μικρότερες. Μπορεί να κατασκευαστεί μια σύνθετη μορφή με αυτή την τεχνική δίνοντας στην αρχή ένα βασικό σχήμα στο στερεό με άμεσο έλεγχο των κορυφών του και στη συνέχεια εστιάζοντας σε επιφάνειες για τις οποίες απαιτείται μεγαλύτερη λεπτομέρεια, υποδιαιρώντας τις και προσαρμόζοντας ανάλογα τη μορφή τους. Αυτό βοηθάει τον σχεδιαστή να δουλεύει προοδευτικά σε όλο και μεγαλύτερο βαθμό λεπτομέρειας, διορθώνοντας και βελτιώνοντας διαρκώς το μοντέλο του.

Η «ψηφιακή γλυπτική» (digital sculpting) είναι μια εντελώς διαφορετικής λογικής και ιδιαίτερα ενδιαφέρουσα προσέγγιση στη δημιουργία οργανικών μορφών (Alcaide-Marzal κ.ά., 2013). Ο σχεδιαστής ξεκινάει από ένα βασικό στερεό και τροποποιεί τη μορφή του χρησιμοποιώντας εργαλεία που μιμούνται αντίστοιχες ενέργειες του πραγματικού κόσμου πάνω σε εύπλαστα αντικείμενα. Μπορεί να σπρώξει, να πιάσει, να τραβήξει, να εξομαλύνει το στερεό βλέποντας άμεσα το οπτικό αποτέλεσμα. Αν και οι ενέργειες αυτές μπορούν να εφαρμοστούν και μέσω του ποντικιού, η προσέγγιση μπορεί να γίνει αποδοτικότερη όταν συνδυάζεται με κατάλληλο υλικό αλληλεπίδρασης σε τρεις διαστάσεις, όπως οι αρθρωτοί βραχίονες αφής (π.χ. Phantom).

Τέλος, ορισμένα είδη τρισδιάστατων μοντέλων μπορούν να δημιουργηθούν και αλγοριθμικά. Αυτή η προσέγγιση, που ονομάζεται και διαδικαστική μοντελοποίηση (procedural modelling), έχει νόημα κυρίως σε περιπτώσεις αντικειμένων με επαναλαμβανόμενες μορφές, όπως για παράδειγμα το ανάγλυφο του εδάφους ή τα βουνά, ή και στη δημιουργία πολλαπλών στιγμιότυπων μιας κατηγορίας αντικειμένων χωρίς αυτά να είναι πανομοιότυπα, όπως, για παράδειγμα, τα δέντρα. Στα συστήματα που υποστηρίζεται τέτοιου τύπου μοντελοποίηση, συνήθως ο σχεδιαστής προσδιορίζει απλά κάποιες αρχικές παραμέτρους και τα τελικά μοντέλα παράγονται διαδικαστικά μέσω κατάλληλου αλγορίθμου.

3.3 Ψηφιοποίηση

Εναλλακτικά της μοντελοποίησης, οι σχεδιαστές μπορούν να δημιουργήσουν νέα τρισδιάστατα μοντέλα και να τα εισάγουν σε μια σκηνή μέσω ψηφιοποίησης. Ψηφιοποίηση είναι η διαδικασία παραγωγής ενός ψηφιακού μοντέλου μέσω της σύλληψης δεδομένων από ένα φυσικό αντικείμενο. Η προσέγγιση αυτή χρησιμοποιείται ούτως ή άλλως σε μεγάλο βαθμό στις υφές, καθώς είναι πολύ συχνή η χρήση επεξεργασμένων φωτογραφιών για την απόδοση συγκεκριμένων υλικών ή όψεων κάποιου αντικειμένου. Στην περίπτωση της τρισδιάστατης γεωμετρίας η ψηφιοποίηση γίνεται με ειδικές συσκευές και αλγορίθμους που καταγράφουν την επιφάνεια και την υφή ενός φυσικού αντικειμένου και παράγουν ένα όμοιο πολυγωνικό μοντέλο.

Η ψηφιοποίηση ενός αντικειμένου γίνεται μέσω ειδικών συσκευών, των τρισδιάστατων σαρωτών (3D scanners). Υπάρχουν διάφορες κατηγορίες σαρωτών ανάλογα με το μέγεθος των αντικειμένων που μπορούν να σαρώσουν, την ακρίβεια και τον τρόπο σάρωσης (Böhler & Marbs, 2002). Σε ό,τι αφορά την τεχνολογία, η σάρωση μπορεί να γίνει είτε με φυσική επαφή με το αντικείμενο είτε εξ αποστάσεως. Στη δεύτερη περίπτωση

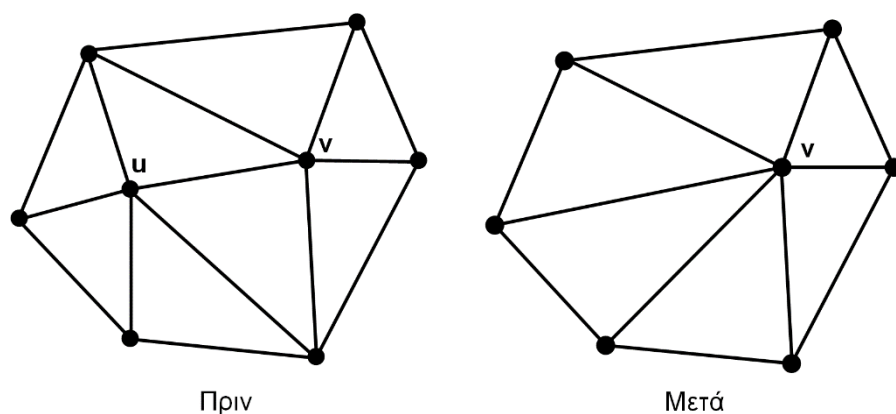
χρησιμοποιείται συνήθως τεχνολογία Laser για τη μέτρηση της απόστασης από κάθε σημείο του αντικειμένου. Μια χαμηλότερου κόστους εναλλακτική είναι η και χρήση πολλαπλών φωτογραφήσεων ενός αντικειμένου για την ανακατασκευή της γεωμετρίας του, όμως το αποτέλεσμα μπορεί να στερείται ακρίβειας. Η σάρωση μπορεί να είναι χειροκίνητη και να έχει ο χρήστης τον έλεγχο της κίνησης του σαρωτή ή εναλλακτικά να γίνεται από σταθερή συσκευή που περιστρέφει μια βάση στην οποία τοποθετείται το αντικείμενο. Υπάρχουν ειδικοί σαρωτές που μπορούν να ψηφιοποιήσουν μεγάλα αντικείμενα ή και κτίρια, καθώς και σαρωτές ανθρώπινου σώματος.

Η ψηφιοποίηση μπορεί να επιταχύνει σημαντικά τη διαδικασία παραγωγής των μοντέλων σε σχέση με την εξαρχής μοντελοποίηση, και τα τελικά μοντέλα να έχουν πολύ μεγαλύτερη ακρίβεια. Για τον λόγο αυτόν χρησιμοποιείται συχνά σε εφαρμογές στις οποίες η ακρίβεια και η λεπτομέρεια στην απεικόνιση της γεωμετρίας παίζει σημαντικό ρόλο, όπως στον πολιτισμό, στην ιατρική, σε περιβάλλοντα εξάσκησης κ.ά. Το μειονέκτημα σε σχέση με τη μοντελοποίηση είναι ότι τα ψηφιακά μοντέλα μπορεί να περιγράφονται από πολύ μεγάλο αριθμό τριγώνων και να απαιτούνται περαιτέρω διαδικασίες απλοποίησης για την τελική χρήση τους σε περιβάλλοντα πραγματικού χρόνου.

3.4 Βελτιστοποίηση για εκτέλεση σε πραγματικό χρόνο

Πολλές φορές τα τρισδιάστατα αντικείμενα που έχουν δημιουργηθεί σε εφαρμογές μοντελοποίησης ή έχουν προκύψει από ψηφιοποίηση έχουν πολύ μεγάλο αριθμό τριγώνων. Κάτι τέτοιο μπορεί να επιδράσει αρνητικά στην απόδοση του συστήματος, ιδιαίτερα αν ο εικονικός κόσμος περιλαμβάνει πληθώρα αντικειμένων. Επιπλέον, η πολύ μεγάλη λεπτομέρεια στη γεωμετρία έχει νόημα μόνο όταν ο χρήστης παρατηρεί από κοντινή απόσταση κάποιο αντικείμενο. Σε πιο μακρινές αποστάσεις είναι πρακτικά αδύνατον να ξεχωρίσουν οι λεπτομέρειες της μορφής των αντικειμένων.

Ο αριθμός των τριγώνων ενός τρισδιάστατου αντικειμένου είναι δυνατόν να μειωθεί χωρίς να επηρεαστεί σημαντικά η τελική ποιότητα απεικόνισης με κατάλληλη επεξεργασία. Η διαδικασία αυτή της *μείωσης πολυγώνων (Polygon Reduction)* υποστηρίζεται από τα περισσότερα σύγχρονα προγράμματα μοντελοποίησης καθώς και από εφαρμογές κατασκευασμένες ειδικά γι' αυτόν τον σκοπό. Η βασική λογική πίσω από αυτούς τους αλγόριθμους είναι η κατάργηση γειτονικών τριγώνων μέσω αφαίρεσης της κοινής τους ακμής και ενοποίησης των δύο κορυφών αυτής (Melax, 1998). Κατά τη διαδικασία της απλοποίησης δίνεται προτεραιότητα στα ζεύγη με το χαμηλότερο «κόστος» ενοποίησης, δηλαδή με τις μικρότερες επιπτώσεις στην τελική ποιότητα. Δύο τέτοια κριτήρια είναι το μήκος της κοινής ακμής και η γωνία που σχηματίζουν τα κανονικά διανύσματα των γειτονικών πολυγώνων. Όσο μικρότερη είναι η κοινή ακμή και η γωνία μεταξύ των δύο διανυσμάτων, τόσο μικρότερη επίδραση αναμένεται να έχει η ενοποίηση στην τελική μορφή του αντικειμένου. Στις περισσότερες εφαρμογές απλοποίησης δίνεται η δυνατότητα στον χρήστη να ρυθμίσει το ποσοστό μείωσης των τριγώνων, και σε αρκετές περιπτώσεις μπορεί να επιτευχθεί σημαντική μείωση χωρίς να αλλοιώνεται ιδιαίτερα το αποτέλεσμα.



Εικόνα 2.21 Μείωση πολυγώνων μέσω ενοποίησης των κορυφών u και v και κατάργησης των τριγώνων που περιλαμβάνουν την ακμή uv .

Η απόδοση μιας σκηνής μπορεί να βελτιωθεί ακόμα περισσότερο αν εισαχθούν *επίπεδα λεπτομέρειας (level of detail)* (Χία κ.ά., 1997). Η τεχνική αυτή βασίζεται στο γεγονός ότι όσο μεγαλύτερη είναι η απόσταση παρατήρησης ενός αντικειμένου, τόσο λιγότερες είναι οι λεπτομέρειές του που γίνονται αντιληπτές και αντίστοιχα μικρότερες είναι και οι απαιτήσεις σε ποιότητα απεικόνισης. Ένας σχεδιαστής μπορεί να αξιοποιήσει τα επίπεδα λεπτομέρειας αν εισάγει παραπάνω από ένα τρισδιάστατα μοντέλα για ένα αντικείμενο, τα οποία παρουσιάζουν μεν την ίδια μορφή, αλλά διαφέρουν στον αριθμό τριγώνων, δηλαδή στη λεπτομέρεια της γεωμετρίας. Στη συνέχεια μπορεί να καθορίσει κατάλληλες αποστάσεις παρατήρησης για καθένα από τα μοντέλα, και το σύστημα αναλαμβάνει την επιλογή του κατάλληλου μοντέλου ανάλογα με τη θέση του παρατηρητή. Αν, για παράδειγμα, σε ένα εικονικό μουσείο περιλαμβάνεται ένα άγαλμα και επιθυμούμε κατά την παρατήρησή του να διακρίνονται σημαντικές λεπτομέρειες, θα μπορούσε το υψηλής ποιότητας μοντέλο να εμφανίζεται σε αποστάσεις μικρότερες των πέντε μέτρων, ένα μεσαίας ποιότητας σε μικρότερες των δέκα και ένα χαμηλής σε μεγαλύτερες αποστάσεις. Οι εφαρμογές απλοποίησης γεωμετρίας μπορούν να δημιουργήσουν από ένα αρχικό μοντέλο πολλαπλά επίπεδα ποιότητας. Με τη χρήση των επιπέδων ποιότητας είναι δυνατόν να υπάρχουν πολλά υψηλής ποιότητας μοντέλα στον εικονικό κόσμο χωρίς να επιβαρύνεται σημαντικά η απόδοσή του, καθώς ανάλογα με την απόσταση παρατήρησης μόνο ένας πολύ μικρός αριθμός από αυτά θα απεικονίζεται σε πλήρη λεπτομέρεια.

4 Υποστηρικτικό λογισμικό

Η ευρεία διάδοση των τεχνολογιών τρισδιάστατων γραφικών και των εφαρμογών τους τις τελευταίες δεκαετίες έχει οδηγήσει στην ανάπτυξη ενός μεγάλου εύρους σχετικών υποστηρικτικών προγραμμάτων και εργαλείων λογισμικού. Είναι διαθέσιμη πληθώρα βιβλιοθηκών και περιβαλλόντων για την ανάπτυξη τρισδιάστατων εφαρμογών πραγματικού χρόνου, διάφορες ολοκληρωμένες εφαρμογές τρισδιάστατης μοντελοποίησης που διευκολύνουν την κατασκευή των τρισδιάστατων σκηνών και άλλα εργαλεία για την επεξεργασία και βελτιστοποίηση των μοντέλων.

Η ανάπτυξη ενός διαδραστικού τρισδιάστατου περιβάλλοντος μπορεί να γίνει με τη χρήση των *βιβλιοθηκών γραφικών χαμηλού επιπέδου* Direct3D και OpenGL, που εκμεταλλεύονται στο έπακρο τη μονάδα επεξεργασίας γραφικών του υπολογιστικού συστήματος. Οι Direct3D και OpenGL είναι δύο βιβλιοθήκες προγραμματισμού εφαρμογών (application programming interface — API) για την απεικόνιση τρισδιάστατων γραφικών, οι οποίες είναι υλοποιημένες σχεδόν εξολοκλήρου σε επίπεδο υλικού από τις κάρτες γραφικών. Αυτό πρακτικά σημαίνει ότι οι υπολογισμοί για την απεικόνιση των σκηνών γίνονται με πολύ μεγάλη ταχύτητα χωρίς να χρησιμοποιείται ο βασικός επεξεργαστής, οδηγώντας σε πολύ υψηλής απόδοσης και ποιότητας αποτελέσματα. Η ανάπτυξη μιας εφαρμογής με τη χρήση μίας από τις παραπάνω βιβλιοθήκες θα πρέπει να γίνει σε κάποια ισχυρή γλώσσα προγραμματισμού (συνήθως C++) και μέσω αυτής να κληθεί η σχετική διεπαφή. Ένα βασικό μειονέκτημα της προσέγγισης αυτής είναι ότι οι Direct3D και OpenGL υποστηρίζουν μόνο χαμηλού επιπέδου ενέργειες για την παραγωγή τρισδιάστατων γραφικών, όπως μετασχηματισμούς τριγώνων και φωτοσκίαση. Δεν υπάρχει εγγενής υποστήριξη για την αναπαράσταση και διαχείριση γράφου σκηνής, για συνθετική κίνηση, αλληλεπιδράσεις του χρήστη κ.λπ. Αυτό πρακτικά σημαίνει ότι είναι στην ευθύνη του προγραμματιστή να αναπτύξει όλη την απαιτούμενη λειτουργικότητα για ένα ολοκληρωμένο τρισδιάστατο περιβάλλον, κάτι που συνεπάγεται πολύ μεγαλύτερο χρόνο ανάπτυξης και ανάγκη για έμπειρους προγραμματιστές σε ένα εύρος πεδίων. Από την άλλη μεριά, η χρήση των βιβλιοθηκών αυτών δίνει στους προγραμματιστές πλήρη έλεγχο, οπότε θεωρητικά μπορούν να βελτιστοποιήσουν την ποιότητα και απόδοση της εφαρμογής τους. Πρόσφατα έχει κυκλοφορήσει και μια έκδοση της OpenGL για περιβάλλον Web, η WebGL. Η τελευταία υποστηρίζεται από τους σύγχρονους φυλλομετρητές Web και μπορεί να απεικονίσει τρισδιάστατο περιεχόμενο σε ιστοσελίδες μέσω της HTML 5.0 και της γλώσσας Javascript χωρίς την ανάγκη χρήσης εξωτερικής εφαρμογής τύπου plug-in.

Οι δυσκολίες στην ανάπτυξη σύνθετων εφαρμογών με τη χρήση των Direct3D ή OpenGL έχουν οδηγήσει στην εμφάνιση *βιβλιοθηκών προγραμματισμού γραφικών υψηλότερου επιπέδου*, οι οποίες ενσωματώνουν έναν αριθμό από επιπλέον λειτουργίες που συναντώνται συχνά σε εικονικά περιβάλλοντα. Η ανάπτυξη μιας εφαρμογής με αυτές τις βιβλιοθήκες απαιτεί και πάλι τη χρήση κάποιας γλώσσας προγραμματισμού, όμως παρέχονται συνήθως στον προγραμματιστή πολύ περισσότερα υλοποιημένα εργαλεία, όπως η διαχείριση του γραφήματος σκηνής, η εισαγωγή τρισδιάστατων μοντέλων από αρχεία, ο χειρισμός συμβάντων όπως ενέργειες του χρήστη κ.ά. Συνεπώς, αν και ο έλεγχος στο τελικό αποτέλεσμα είναι μικρότερος, οι χρόνοι ανάπτυξης εφαρμογών είναι πολύ βελτιωμένοι σε σχέση με τη χρήση των βιβλιοθηκών χαμηλού

επιπέδου που παρουσιάσαμε προηγουμένως. Χαρακτηριστικά παραδείγματα τέτοιων βιβλιοθηκών είναι η Java3D για τη γλώσσα Java και η OpenSceneGraph για τη C++.

Οι *μηχανές γραφικών υψηλού επιπέδου* και οι *μηχανές παιχνιδιών* διευκολύνουν ακόμα περισσότερο την κατασκευή ενός διαδραστικού τρισδιάστατου περιβάλλοντος. Πρόκειται για ολοκληρωμένα περιβάλλοντα ανάπτυξης, τα οποία τις περισσότερες φορές συμπεριλαμβάνουν και ενσωματωμένη εφαρμογή οπτικής επεξεργασίας της σκηνής. Έχουν ήδη υλοποιημένο έναν βασικό μηχανισμό λειτουργίας που φροντίζει για την οπτική απόδοση της σκηνής, την κίνηση των αντικειμένων, τη διαχείριση της εισόδου του χρήστη κ.ά. Οι μηχανές παιχνιδιών περιλαμβάνουν συνήθως και επιπλέον δυνατότητες, όπως μηχανισμούς φυσικής και τεχνητής νοημοσύνης. Οι προγραμματιστές προσδιορίζουν τη συμπεριφορά των αντικειμένων σε ένα υψηλότερο επίπεδο μέσω ειδικής γλώσσας σεναρίου (scripting language) ή κάποιας γλώσσας προγραμματισμού που παρέχεται από αυτές τις μηχανές. Υπάρχει ένας πολύ μεγάλος αριθμός διαθέσιμων μηχανών γραφικών, εμπορικών ή ελεύθερων, για διάφορες γλώσσες προγραμματισμού και πλατφόρμες εκτέλεσης. Οι πιο δημοφιλείς σήμερα είναι οι Unity και Unreal Engine, που διατίθενται δωρεάν για μη εμπορική χρήση, προσφέρουν πολύ ποιοτικά οπτικά αποτελέσματα και είναι σχετικά εύκολες στην εκμάθηση και στη χρήση. Το μεγάλο πλεονέκτημα της χρήσης μηχανών παιχνιδιών για την ανάπτυξη εικονικών κόσμων είναι ότι μειώνεται σημαντικά ο χρόνος ανάπτυξης και δεν απαιτούνται εξειδικευμένες γνώσεις προγραμματισμού γραφικών και συνθετικής κίνησης από την ομάδα ανάπτυξης. Από την άλλη μεριά, οι δυνατότητες των περιβαλλόντων που θα κατασκευαστούν περιορίζονται από την ίδια τη μηχανή και τις προσφερόμενες λειτουργίες της.

Η μοντελοποίηση των αντικειμένων και η κατασκευή των τρισδιάστατων σκηνών υποστηρίζεται σήμερα από πληθώρα σχετικών εφαρμογών. Υπάρχουν εργαλεία επεξεργασίας εικόνας για τη δημιουργία και επεξεργασία των υφών και περιβάλλοντα τρισδιάστατης μοντελοποίησης και κίνησης για την κατασκευή τρισδιάστατων αντικειμένων και σκηνών. Τα σύγχρονα εργαλεία μοντελοποίησης υποστηρίζουν τις περισσότερες σύγχρονες τεχνικές κατασκευής γεωμετρίας (βλ. ενότητα 3.2) και διευκολύνουν σημαντικά τους σχεδιαστές στο έργο τους. Σήμερα κυκλοφορούν ισχυρά εμπορικά εργαλεία, όπως το 3D Studio Max και το SoftImage XSI της εταιρείας Autodesk, αλλά και προγράμματα ελεύθερα στη χρήση, όπως το Blender και το Sketchup. Επιπλέον, υπάρχουν και ειδικά προγράμματα μοντελοποίησης για την κατασκευή και κίνηση ανθρώπινων χαρακτήρων, όπως το Poser και το MakeHuman, τα οποία έχουν έτοιμες βιβλιοθήκες ανθρώπινων μοντέλων και ρούχων και επιτρέπουν την εύκολη διαμόρφωση χαρακτηριστικών του σώματος και προσώπου ενός χαρακτήρα. Το MakeHuman είναι προϊόν ανοιχτού λογισμικού και διατίθεται ελεύθερα. Τα προγράμματα μοντελοποίησης δεν εστιάζουν μόνο σε εφαρμογές πραγματικού χρόνου αλλά και σε φωτορεαλιστικές απεικονίσεις ή βίντεο. Πολλές φορές τα παραγόμενα μοντέλα μπορεί να έχουν μεγάλο αριθμό τριγώνων. Θα πρέπει οι σχεδιαστές να διασφαλίζουν εξαρχής ότι τα μοντέλα τους έχουν απλοποιημένη γεωμετρία ή και να χρησιμοποιούν τις διαθέσιμες λειτουργίες περιορισμού πολυγώνων που συνήθως παρέχονται από τα προγράμματα τρισδιάστατης μοντελοποίησης, ώστε οι τελικές γεωμετρίες να μην επιβαρύνουν σημαντικά την απόδοση του κόσμου.

Όνομα / Πρόγραμμα προέλευσης	Κατάληξη
3D Studio	.3ds, .max
Blender	.blend
Collada	.dae
AutoCAD	.dxf
Autodesk Exchange	.fbx
Lightwave	.lwo
VRML	.wrl
X3D	.x3d

Πίνακας 2.4 Δημοφιλείς μορφότυποι τρισδιάστατων γραφικών.

Τέλος, ένα σημαντικό πρόβλημα στον χώρο των τρισδιάστατων γραφικών είναι η ύπαρξη πολλαπλών διαφορετικών μορφότυπων αποθήκευσης (file formats) των δεδομένων. Το πρόβλημα προκύπτει από το γεγονός ότι πολλά από τα προγράμματα τρισδιάστατης μοντελοποίησης χρησιμοποιούν τους δικούς τους μορφότυπους για την αποθήκευση της σκηνής, καθένας από τους οποίους περιλαμβάνει διαφορετικές πληροφορίες σχετικά με τη σκηνή, τα μοντέλα, τα υλικά, τον φωτισμό και την κίνηση. Το αποτέλεσμα είναι ότι η μετατροπή μεταξύ διαφορετικών τύπων αρχείων μπορεί να προκαλέσει προβλήματα στην κλίμακα, στις υφές

ή ακόμα και στη γεωμετρία των μοντέλων και μπορεί να χρειαστεί αρκετή επιπλέον επεξεργασία για τη σωστή εισαγωγή ενός εξωτερικού μοντέλου σε μια υπάρχουσα σκηνή. Για την αντιμετώπιση τέτοιων προβλημάτων υπάρχουν προγράμματα όπως το MeshLab, το οποίο διατίθεται ελεύθερα, τα οποία μπορούν να διαβάσουν και να αποθηκεύσουν πολλαπλούς διαφορετικούς μορφότυπους, ενώ επιτρέπουν και μερική επεξεργασία των δεδομένων για την αντιμετώπιση προβλημάτων μετατροπής. Στον πίνακα 2.4 παρουσιάζονται ορισμένοι δημοφιλείς μορφότυποι τρισδιάστατων γραφικών.

5 Συμπεράσματα

Στο παρόν κεφάλαιο έγινε μια συνοπτική παρουσίαση εισαγωγικών εννοιών του χώρου των τρισδιάστατων γραφικών με στόχο τη βασική κατανόηση τόσο της τεχνολογίας πίσω από την απεικόνιση όσο και των διαδικασιών μοντελοποίησης και διαμόρφωσης της σκηνής ενός εικονικού κόσμου. Από την επισκόπηση του χώρου προκύπτει ότι υπάρχουν πολλές και διαφορετικές τεχνικές για την περιγραφή των γεωμετριών και για τη φωτοσκίασή τους, οι οποίες αν αξιοποιηθούν σωστά μπορούν να συμβάλουν σε πολύ ποιοτικά και ρεαλιστικά αποτελέσματα. Εντούτοις, οι εικονικοί κόσμοι ως συστήματα πραγματικού χρόνου έχουν ισχυρούς περιορισμούς σε απόδοση και η πραγματική πρόκληση για τους σχεδιαστές είναι η διατήρηση μιας καλής ισορροπίας μεταξύ ποιότητας και απόδοσης. Αν και τα μοντέλα στους εικονικούς κόσμους περιγράφονται ως συλλογές τριγώνων, ο μεγάλος αριθμός τριγώνων σε συνδυασμό με κατάλληλους αλγορίθμους φωτοσκίασης μπορεί να αποδώσει πολύ ικανοποιητικά καμπύλες επιφάνειες και τελικά να περιγράψει με μεγάλη λεπτομέρεια τα αντικείμενα και τους χαρακτήρες ενός περιβάλλοντος. Επιπλέον, η χρήση υφών και κατάλληλων shaders μπορεί να προσθέσει λεπτομέρειες και ειδικά εφέ χωρίς σημαντικό υπολογιστικό κόστος. Η απόδοση μπορεί να βελτιωθεί με τη χρήση εργαλείων βελτιστοποίησης γεωμετρίας, επιπέδων λεπτομέρειας ή ακόμα και billboards, δίνοντας τη δυνατότητα στο περιβάλλον να υποστηρίξει ακόμα μεγαλύτερο αριθμό αντικειμένων ή ακόμα πιο λεπτομερείς γεωμετρίες.

Φαίνεται λοιπόν ότι απαιτείται πολύ προσεκτική κατασκευή και διαμόρφωση τόσο της σκηνής όσο και της γεωμετρίας και των υλικών των επιμέρους μοντέλων του κόσμου, ώστε να μπορέσει να επιτευχθεί η επιθυμητή ισορροπία. Αυτό σημαίνει ότι η ομάδα ανάπτυξης θα πρέπει να επενδύσει σημαντικό χρόνο στη μελέτη των δυνατοτήτων της πλατφόρμας ανάπτυξης, στην επιλογή κατάλληλων μορφών αναπαράστασης, στη μοντελοποίηση και βελτιστοποίηση των γεωμετριών, στην αξιοποίηση των υλικών και των ειδικών εφέ για τα καλύτερα δυνατά οπτικά αποτελέσματα, στη διασφάλιση της καλής απόδοσης του αποτελέσματος και πολλά άλλα. Είναι σε κάθε περίπτωση σημαντικό να αποκτήσει ένας σχεδιαστής εικονικών κόσμων καλή κατανόηση των δυνατοτήτων της τεχνολογίας και των πλεονεκτημάτων και περιορισμών της κάθε επιλογής του, ώστε να είναι σε θέση να πάρει τις κατάλληλες αποφάσεις για την καλύτερη εκμετάλλευση του μέσου, ανάλογα με τις απαιτήσεις της εφαρμογής.

Αναφορές

- Alcaide-Marzal, J., Diego-Más, J. A., Asensio-Cuesta, S. & Piqueras-Fiszman, B. (2013). An exploratory study on the use of digital sculpting in conceptual product design. *Design Studies*, 34(2), 264–284.
- Aubel, A., Boulic, R. & Thalmann, D. (2000). Real-time display of virtual humans: Levels of details and impostors. *IEEE Transactions on Circuits and Systems for Video Technology*, 10(2), 207–217.
- Blinn, J. F. (1978). Simulation of wrinkled surfaces. *ACM SIGGRAPH Computer Graphics*, 12(3), 286–292.
- Böhler, W. & Marbs, A. (2002). 3D scanning instruments. *Proceedings of the CIPA WG 6 International Workshop on Scanning for Cultural Heritage Recording*, 9–18.
- Bracken, C. C. & Skalski, P. (2009). Telepresence and video games: The impact of image quality. *PsychNology Journal*, 7(1), 101–112.

- Catmull, E. E. (1974). *A Subdivision Algorithm for Computer Display of Curved Surfaces*. (No. UTEC-CSC-74-133). UTAH UNIV SALT LAKE CITY SCHOOL OF COMPUTING.
- Fuchs, H., Kedem, Z. M. & Naylor, B. F. (1980). On visible surface generation by a priori tree structures. *ACM SIGGRAPH Computer Graphics*, 14(3), 124–133.
- Goral, C. M., Torrance, K. E., Greenberg, D. P. & Battaile, B. (1984). Modeling the interaction of light between diffuse surfaces. *ACM SIGGRAPH Computer Graphics*, 18(3), 213–222.
- Haeberli, P. & Segal, M. (1993). Texture mapping as a fundamental drawing primitive. *Fourth Eurographics Workshop on Rendering*, 259–266.
- Hasenfratz, J. M., Lapierre, M., Holzschuch, N. & Sillion, F. (2003). A Survey of Real-time Soft Shadows Algorithms. *Computer Graphics Forum*, 22(4).
- Lewis, R. R. (1994). Making Shaders More Physically Plausible. *Computer Graphics Forum*, 13(2), 109–120.
- Melax, S. (1998). A Simple, Fast and Effective Polygon Reduction Algorithm. *Game Developer Magazine*, (November), 44–49.
- Pausch, R., Snoddy, J., Taylor, R., Watson, S. & Haseltine, E. (1996). Disney’s Aladdin: first steps toward storytelling in virtual reality. *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, 193–203.
- Phong, B. T. (1975). Illumination for Computer-Generated Images. *Communications of the ACM*, 18(6), 311–317.
- Shoemake, K. (1985). Animating rotation with quaternion curves. *ACM SIGGRAPH Computer Graphics*, 19(3), 245–254.
- Strauss, P. S. & Carey, R. (1992). An object-oriented 3D graphics toolkit. *ACM SIGGRAPH Computer Graphics*, 26(2).
- Watt, A. (1994). *3D Computer Graphics*. Addison-Wesley.
- Xia, J. C., El-Sana, J. & Varshney, a. (1997). Adaptive real-time level-of-detail based rendering for polygonal models. *IEEE Transactions on Visualization and Computer Graphics*, 3(2), 171–181.
- Θεοχάρης, Θ., Παπαϊωάννου, Γ., Πλατής, Ν. & Πατρικαλάκης, Ν. (2012). *Γραφικά και Οπτικοποίηση Αρχές και Αλγόριθμοι*. Συμμετρία.

Σύνδεσμοι

Unity: www.unity3d.com

Blender: www.blender.org

MakeHuman: <http://www.makehuman.org/>

MeshLab: <http://meshlab.sourceforge.net/>

Περαιτέρω Μελέτη

Το παρόν κεφάλαιο κάνει μόνο μια εισαγωγική επισκόπηση του χώρου των γραφικών με υπολογιστή. Μπορείτε να διαβάσετε πολλά περισσότερα σχετικά με τα τρισδιάστατα γραφικά και τις διαδικασίες οπτικοποίησης στα παρακάτω βιβλία:

Θεοχάρης, Θ., Παπαϊωάννου, Γ., Πλατής, Ν. & Πατρικαλάκης, Ν. Μ. (2012). *Γραφικά και Οπτικοποίηση, Αρχές και Αλγόριθμοι*. Εκδόσεις Συμμετρία.

Slater, M., Steed, A. & Chrysanthou, Y. (2001). *Computer Graphics and Virtual Environments: From Realism to Real-Time*. Addison Wesley.

Ερωτήσεις Κατανόησης

1. Εξηγήστε το πρόβλημα της σύγκρουσης μεταξύ ποιότητας και απόδοσης σε εφαρμογές γραφικών πραγματικού χρόνου, όπως οι εικονικοί κόσμοι.
2. Σε ποιες κατηγορίες εφαρμογών μπορεί να προκύψει ανάγκη χρήσης μοντέλων αναπαράστασης όγκου;
3. Για ποιο λόγο χρησιμοποιούνται δύο πίνακες για την αναπαράσταση της γεωμετρίας ενός αντικειμένου;
4. Πώς αποδίδεται η υφή (texture) στην επιφάνεια ενός αντικειμένου;
5. Ποια είναι τα βασικά είδη φωτισμού στα τρισδιάστατα περιβάλλοντα;
6. Να αναφέρετε ορισμένα βασικά στοιχεία που περιλαμβάνονται στην περιγραφή του υλικού (material) ενός αντικειμένου.
7. Περιγράψτε συνοπτικά τα βασικά βήματα της διαδικασίας απεικόνισης μιας σκηνής.
8. Τι είναι ο γράφος σκηνής και για ποιο λόγο χρησιμοποιείται σήμερα από όλες σχεδόν τις τρισδιάστατες εφαρμογές;
9. Με ποιους τρόπους μπορούμε να αποδώσουμε καμπύλες επιφάνειες σε προγράμματα τρισδιάστατης μοντελοποίησης;
10. Σε ποιες περιπτώσεις αντικειμένων είναι προτιμότερη η διαδικασία ψηφιοποίησης σε σχέση με τη μοντελοποίηση;
11. Με ποιους τρόπους μπορούμε να βελτιώσουμε την απόδοση της σκηνής μας;
12. Να αναφέρετε τα πλεονεκτήματα και τα μειονεκτήματα της χρήσης μηχανών παιχνιδιών για την ανάπτυξη τρισδιάστατων κόσμων.

Ασκήσεις

1. Εγκαταστήστε το πρόγραμμα Blender και προσπαθήστε να μοντελοποιήσετε βασικά χρηστικά αντικείμενα (π.χ. ποτήρι, κουτάλι, πιάτο) χρησιμοποιώντας τεχνικές μοντελοποίησης που αναφέραμε.
2. Εγκαταστήστε το Unity3D και δοκιμάστε να κατασκευάσετε αληθοφανή υλικά ρυθμίζοντας τις διάφορες παραμέτρους χρώματος, διαφάνειας, αντανάκλασης κ.λπ. Αποδώστε τα υλικά σας σε απλά αντικείμενα μιας κατάλληλα φωτισμένης σκηνής και συγκρίνετε την απεικόνισή τους με τα αντίστοιχα αληθινά υλικά.
3. Το 123D Catch (<http://www.123dapp.com/catch>) είναι ένα πρόγραμμα της Autodesk για κινητά και Web το οποίο μπορεί να ψηφιοποιήσει αντικείμενα από πολλαπλές φωτογραφίες τους. Χρησιμοποιήστε το και δοκιμάστε να ψηφιοποιήσετε ένα αντικείμενο με περίπλοκη γεωμετρία, π.χ. ένα αγαλματίδιο. Σχολιάστε την ποιότητα του αποτελέσματος ως προς την οπτική ποιότητα και τον αριθμό τριγώνων του παραγόμενου μοντέλου.

Κεφάλαιο 3: Κίνηση

Σύνοψη

Στο παρόν κεφάλαιο παρουσιάζεται η διαδικασία παραγωγής συνθετικής κίνησης (animation) στους εικονικούς κόσμους, εστιάζοντας τόσο στις απαιτούμενες τεχνολογίες όσο και σε σχεδιαστικά ζητήματα. Αρχικά παρουσιάζονται οι διάφορες επιλογές ως προς τη μοντελοποίηση των αντικειμένων, ανάλογα με το είδος της κίνησης που θέλουμε να εκτελέσουν. Στη συνέχεια, γίνεται μια επισκόπηση των αλγορίθμων και τεχνικών για την αναπαραγωγή προκαθορισμένων κινήσεων καθώς και για τη δυναμική παραγωγή κίνησης μέσω προσομοίωσης. Παρουσιάζεται η διαδικασία καθορισμού και παραγωγής κίνησης με θέσεις κλειδιά (keyframing), η χρήση δεδομένων σύλληψης κίνησης (motion capture), η μοντελοποίηση φυσικών νόμων και θέματα ανίχνευσης συγκρούσεων και αντίδρασης σε αυτές. Τέλος, αναλύονται τεχνικές σύνθετης μετακίνησης και συμπεριφοράς συνθετικών χαρακτήρων ξεκινώντας από την παρουσίαση αρχιτεκτονικών απόφασης, όπως μηχανές καταστάσεων και ευφυείς πράκτορες, και συνεχίζοντας σε τεχνικές όπως η εύρεση μονοπατιού, η δυναμική αποφυγή αντικειμένων και η συντονισμένη κίνηση πλήθους.

Προαπαιτούμενη γνώση

Απαιτούνται γνώσεις πληροφορικής, βασικές γνώσεις γραφικών με υπολογιστή, καθώς και βασικές γνώσεις μαθηματικών.

1 Κινούμενα αντικείμενα στους εικονικούς κόσμους

Όσο καθοριστική είναι η ρεαλιστική απεικόνιση για την επιτυχία ενός εικονικού κόσμου, άλλο τόσο, και ίσως ακόμα περισσότερο, είναι η κίνηση. Στους εικονικούς κόσμους οι κινήσεις είναι πολλαπλές και διαρκείς, κάποιες φορές προκαθορισμένες και κάποιες άλλες δυναμικές, προκαλούμενες από ενέργειες των χρηστών και συμβάντα του περιβάλλοντος. Οι ενσαρκώσεις των χρηστών μετακινούνται ελεύθερα στον χώρο κατά την πλοήγηση των χρηστών, εκτελούν χειρονομίες αλληλεπιδρώντας με αντικείμενα του περιβάλλοντος (π.χ. πιάνουν ένα αντικείμενο, χειρίζονται μια συσκευή) και τροποποιούν την έκφραση του προσώπου τους κατά τη διάρκεια της ομιλίας. Και τα ίδια τα αντικείμενα του περιβάλλοντος όμως μπορεί να μετακινούνται ή γενικότερα να αλλάζουν κατάσταση λόγω των κανόνων του περιβάλλοντος ή και των ενεργειών που ασκούν οι χρήστες σε αυτά. Για παράδειγμα, η πάροδος του χρόνου επηρεάζει τους δείκτες του ρολογιού, τη θέση του ήλιου, την ένταση του φωτισμού της σκηνής (εφόσον χρησιμοποιείται κάποιο είδος φωτός που προσομοιώνει την ήλιο) κ.ά. Η βαρύτητα εξαναγκάζει ένα αντικείμενο που βρίσκεται στον αέρα να εκτελέσει επιταχυνόμενη κίνηση προς το έδαφος, οι εξισώσεις της κρούσης καθορίζουν το αποτέλεσμα της μετέπειτα σύγκρουσής του με το έδαφος και τη νέα ταχύτητα που θα αποκτήσει και, τέλος, όσο κινείται στο έδαφος ασκείται πάνω του επιπλέον η δύναμη της τριβής. Ένας χρήστης πατάει (με κάποιας μορφής διεπαφή) γκάζι σε ένα μοντέλο αυτοκινήτου και αυτό αρχίζει και κινείται. Τέλος, μπορεί να υπάρχουν στον εικονικό κόσμο και «αυτόνομες» οντότητες, δηλαδή χαρακτηριστές ή άλλες «ζωντανές» οντότητες ελεγχόμενες από τον υπολογιστή, οι οποίες επενεργούν στο περιβάλλον βάσει κάποιου μοντέλου συμπεριφοράς, το οποίο τελικά καθορίζει την κίνησή τους. Για παράδειγμα, ένας εικονικός ξεναγός σε ένα μουσείο αντιλαμβάνεται ότι ήρθε ένας νέος επισκέπτης και κινείται προς το μέρος του για να τον χαιρετήσει και να του προσφέρει υπηρεσίες ξενάγησης.

Η κίνηση γίνεται αντιληπτή ως μια συνεχής μεταβολή ιδιοτήτων των αντικειμένων στον χρόνο. Βεβαίως, σε αντίθεση με τον φυσικό κόσμο στον οποίο ο χρόνος είναι συνεχής, στη συνθετική κίνηση (animation) ο χρόνος είναι εκ των πραγμάτων διακριτός. Αυτό που πραγματικά συμβαίνει στα συστήματα αναπαραγωγής κινούμενης εικόνας, όπως στον κινηματογράφο, στην τηλεόραση και στα διάφορα ειδών υπολογιστικά συστήματα, είναι ότι παρουσιάζονται διαδοχικά πλάνα (καρέ) στον θεατή, μεταξύ των οποίων υπάρχουν μικρές διαφοροποιήσεις στα περιεχόμενα της σκηνής (Thalmann & Thalmann, 1990). Αν η ταχύτητα προβολής των πλάνων, δηλαδή ο ρυθμός ανανέωσης της εικόνας (frames per second – FPS) είναι αρκετά μεγάλος, η μεταβολή γίνεται αντιληπτή από τον ανθρώπινο εγκέφαλο ως συνεχής και έτσι δημιουργείται η ψευδαίσθηση της κίνησης. Σύμφωνα με τους ερευνητές απαιτείται ρυθμός ανανέωσης τουλάχιστον 12 πλάνων το δευτερόλεπτο για να γίνει αντιληπτή ως συνεχής η κίνηση και από εκεί και πέρα όσο γρηγορότερα κινούνται τα αντικείμενα του συνθετικού περιβάλλοντος τόσο μεγαλύτερος θα πρέπει να είναι ο ρυθμός ανανέωσης για

να μπορεί ο θεατής να τα παρακολουθήσει. Στα σύγχρονα παιχνίδια υπολογιστών οι απαιτήσεις των παικτών υπερβαίνουν τα 60 πλάνα το δευτερόλεπτο. Εκτός από τον κατάλληλο ρυθμό ανανέωσης, για να είναι επιτυχημένη η κίνηση σε ένα συνθετικό περιβάλλον θα πρέπει να γίνεται με τέτοιο τρόπο ώστε να μπορεί να την αντιλαμβάνεται ο θεατής ως «φυσική». Οι άνθρωποι έχουμε την έμφυτη ικανότητα να επισημαίνουμε πολύ εύκολα «αφύσικες» ή «παράλογες» κινήσεις. Ως εκ τούτου, περιμένουμε από τα «εικονικά» αντικείμενα να συμπεριφέρονται όπως τα αντίστοιχα φυσικά: τα τζάμια να σπάνε, τα ελαστικά αντικείμενα να παραμορφώνονται, τα υγρά να ρέουν, οι άνθρωποι να βαδίζουν διαφορετικά σε τραχύ και ανηφορικό δρόμο σε σχέση με την ευθεία κ.ά. Ως εκ τούτου, είναι στην ευθύνη του σχεδιαστή της κίνησης αλλά και του ίδιου του μηχανισμού αναπαραγωγής, όπως θα δούμε στη συνέχεια, να διασφαλίσει τη φυσικότητα της κίνησης, που θα συνεισφέρει τελικά στη διατήρηση του ενδιαφέροντος και της «παρουσίας» των χρηστών στον εικονικό κόσμο.

Για να μπορέσουμε να κινήσουμε τα αντικείμενα ενός τρισδιάστατου χώρου θα πρέπει λοιπόν να καθορίσουμε τις μεταβολές των ιδιοτήτων τους στον χρόνο. Η ανάγκη αυτή μεταφράζεται σε δύο επιμέρους ενέργειες: Πρώτον, θα πρέπει να έχουμε μοντελοποιήσει τα αντικείμενα με τρόπο τέτοιο ώστε να μπορούν να υποστηριχθούν οι κινήσεις που θέλουμε να εκτελέσουν. Κατά τη μοντελοποίηση καθορίζεται το είδος της κίνησης που θα εκτελούν τα αντικείμενα. Για παράδειγμα, αν ένα αντικείμενο θα συμπεριφέρεται ως συμπαγές ή αν θα μπορεί να παραμορφώνεται η επιφάνειά του, πώς η κίνηση ενός τμήματος ενός σύνθετου αντικειμένου θα επηρεάζει άλλα τμήματα κ.ά. Δεύτερον, θα πρέπει να οριστεί η ίδια η κίνηση που θα εκτελούν τα αντικείμενα. Η κίνηση αυτή μπορεί να είναι προκαθορισμένη ή δυναμική. Αν η κίνηση είναι προκαθορισμένη, οι λεπτομέρειές της μπορεί να ορίζονται εν μέρει από τον σχεδιαστή και ένας μηχανισμός αναπαραγωγής να συμπληρώνει τα ενδιάμεσα πλάνα ή να έχουν προέλθει από σύλληψη κίνησης αντικειμένων (συνήθως ανθρώπων) του πραγματικού κόσμου. Στους εικονικούς κόσμους όμως δεν είναι όλες οι κινήσεις προκαθορισμένες, διότι επηρεάζονται από συμβάντα του περιβάλλοντος και ενέργειες των χρηστών. Κατά συνέπεια μπορεί να παράγεται και αλγοριθμικά η κίνηση, είτε υπακούοντας σε μοντελοποιημένους κανόνες φυσικής είτε αναπαράγοντας κάποιο πιο σύνθετο μοντέλο συμπεριφοράς για την περιγραφή ζωντανών οντοτήτων με δυνατότητες αντίληψης του περιβάλλοντος και απόφασης.

2 Μοντελοποίηση για την υποστήριξη κίνησης

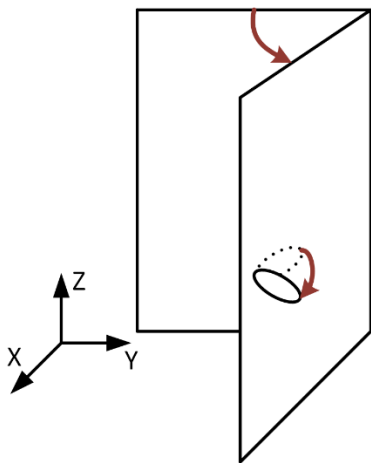
Τα βασικά μοντέλα που χρησιμοποιούνται για την υποστήριξη συνθετικών κινήσεων περιγράφουν τρεις κατηγορίες αντικειμένων: αυτά που είναι συμπαγή, αυτά που μπορούν να παραμορφωθούν και αυτά των οποίων η μορφή είναι τόσο ασαφής ώστε να περιγράφεται από μεμονωμένα σωματίδια. Η κίνηση των συνθετικών χαρακτήρων, δηλαδή των ανθρώπων και άλλων ζωντανών οργανισμών, αναφέρεται ως ξεχωριστό μοντέλο, επειδή συνδυάζει κίνηση συμπαγών αντικειμένων (του σκελετού) με παραμορφώσιμα αντικείμενα (το δέρμα και τα ρούχα).

2.1 Συμπαγή αντικείμενα

Τα *συμπαγή σώματα (rigid bodies)*, δηλαδή τα αντικείμενα των οποίων η γεωμετρία δεν μεταβάλλεται, είναι πιθανότατα τα πιο συχνά εμφανιζόμενα στους εικονικούς κόσμους. Άλλωστε πάρα πολλά από τα αντικείμενα με τα οποία αλληλεπιδρούμε καθημερινά στον φυσικό κόσμο είναι επίσης συμπαγή. Στα συμπαγή αντικείμενα η μετακίνηση επηρεάζει μόνο τη θέση και τον προσανατολισμό τους, εφόσον το σχήμα τους παραμένει αμετάβλητο στον χρόνο. Οποιοδήποτε πολυγωνικό αντικείμενο είναι τοποθετημένο στη σκηνή μπορεί να κινηθεί ως συμπαγές αντικείμενο, αν μεταβάλουμε τον μετασχηματισμό του στη διάρκεια του χρόνου μέσω διαδοχικών μετατοπίσεων ή/και αλλαγών προσανατολισμού. Για παράδειγμα, μεταβάλλοντας τη θέση ενός συρταριού προς την κατάλληλη διεύθυνση μπορούμε να προκαλέσουμε κίνηση που να το δείχνει να ανοίγει. Αντίστοιχα, μεταβάλλοντας τη γωνία περιστροφής μιας πόρτας ως προς τον κάθετο άξονα μπορούμε να τη δείξουμε να ανοίγει.

Σε αρκετές περιπτώσεις τα αντικείμενα που έχουμε μοντελοποιήσει, αν και συμπαγή, μπορεί να αποτελούνται από επιμέρους τμήματα που εκτελούν ανεξάρτητες κινήσεις. Αν επιστρέψουμε στο παράδειγμα της πόρτας, μπορούμε να φανταστούμε ένα χερούλι πάνω σε αυτήν, το οποίο όταν χρησιμοποιείται περιστρέφεται σε έναν διαφορετικό άξονα από αυτόν της πόρτας. Στην περίπτωση αυτήν, το σύνθετο αντικείμενο πόρτας-χερουλιού θα μοντελοποιηθεί με μια ιεραρχία μετασχηματισμών, όπως είδαμε στο προηγούμενο κεφάλαιο. Το αντικείμενο της πόρτας θα έχει έναν μετασχηματισμό T , μέσω του οποίου θα μπορεί να περιστρέφεται ως προς τον άξονα Z μέσα σε κάποια όρια γωνιών. Σε αυτό θα προστεθεί ως απόγονος το

χερούλι με τον δικό του μετασχηματισμό T' μέσω του οποίου θα μπορεί να περιστρέφεται ως προς τον άξονα X στο δικό του τοπικό σύστημα συντεταγμένων. Αν λοιπόν θέλουμε να προκαλέσουμε συνθετική κίνηση τέτοια ώστε να κινείται ταυτόχρονα το χερούλι προς τα κάτω και να ανοίγει η πόρτα προς τα έξω, αρκεί να τροποποιούμε τις αντίστοιχες γωνίες περιστροφής της πόρτας και του χερουλιού στους κατάλληλους άξονες, γεγονός που θα επηρεάζει τα περιεχόμενα των πινάκων μετασχηματισμού T και T' και θα προκαλεί το ανάλογο οπτικό αποτέλεσμα. Η εικόνα 3.1 απεικονίζει τη συνδυασμένη κίνηση.



Εικόνα 3.1 Συνδυασμένη κίνηση πόρτας και χερουλιού

Το παράδειγμα που μόλις αναφέραμε περιλαμβάνει δύο σημεία περιστροφής για την κίνηση του αντικειμένου, όμως υπάρχουν και πιο σύνθετες περιπτώσεις αντικειμένων με πολλαπλά σημεία περιστροφής, τα οποία στη γενική περίπτωση ονομάζουμε *αρθρωτά αντικείμενα* (*articulated objects*). Ένα αρθρωτό αντικείμενο περιλαμβάνει έναν αριθμό από *αρθρώσεις* (*joints*), συνήθως περιστροφικές, και *τμήματα* (*segments*) τα οποία είναι συνδεδεμένα στις αρθρώσεις. Κάθε τμήμα μπορεί με τη σειρά του να συμπεριλαμβάνει και άλλες αρθρώσεις, δημιουργώντας τελικά μία ιεραρχία κατά την οποία η περιστροφή μιας άρθρωσης επηρεάζει τα τμήματα και τις αρθρώσεις που αποτελούν παιδιά, εγγόνια ή γενικότερα απογονικά στοιχεία αυτής. Αυτή η ιεραρχική δομή των αρθρωτών αντικειμένων αναπαρίσταται μέσω μιας αντίστοιχης ιεραρχίας μετασχηματισμών στον γράφο σκηνης. Το πιο χαρακτηριστικό παράδειγμα αρθρωτού αντικειμένου είναι οι ζωντανοί οργανισμοί με σκελετό, και για τον λόγο αυτόν η ιεραρχική δομή των αντικειμένων αυτών ονομάζεται και «*σκελετός*» (*skeleton*), τα τμήματα ονομάζονται και *οστά* (*bones*) και η συνθετική κίνηση ενός αρθρωτού αντικειμένου ονομάζεται *σκελετική κίνηση* (*skeletal animation*).

Αν θέλουμε να προκαλέσουμε κίνηση σε ένα αρθρωτό μοντέλο, αρκεί να καθορίσουμε τις τιμές των περιστροφών των διάφορων αρθρώσεών του και, στην περίπτωση που αυτό κινείται και ως σύνολο, τη συνολική περιστροφή και μετατόπισή του. Η προσέγγιση αυτή, που είναι η απλούστερη και συνηθέστερη, ονομάζεται *ευθεία κινηματική*. Όπως είναι αναμενόμενο, η τελική θέση των τμημάτων του αντικειμένου καθορίζεται από το σύνολο των περιστροφών των αρθρώσεων που βρίσκονται πιο πάνω από αυτό στην ιεραρχία. Για παράδειγμα, η περιστροφή του ώμου, του αγκώνα και του καρπού επηρεάζει τη θέση του χεριού μας. Τι γίνεται όμως στην περίπτωση που θέλουμε στη διάρκεια της κίνησης το χέρι ενός χαρακτήρα να βρεθεί σε ένα συγκεκριμένο σημείο, π.χ. να πιάσει το χερούλι μιας πόρτας; Στην περίπτωση αυτήν, η ευθεία κινηματική δε βοηθάει. Για να λύσουμε το παραπάνω πρόβλημα θα πρέπει να επιλύσουμε ένα σύστημα εξισώσεων, ώστε να καταλήξουμε σε κατάλληλες γωνίες των αρθρώσεων που θα φέρουν ένα καθορισμένο τμήμα του αντικειμένου σε μία συγκεκριμένη θέση. Το πρόβλημα αυτό συνήθως δεν έχει μοναδική λύση. Υπάρχουν περιπτώσεις στις οποίες δεν έχει καμία λύση, για παράδειγμα ο χαρακτήρας βρίσκεται αρκετά μακριά από την πόρτα και θα πρέπει να πλησιάσει περισσότερο για να πιάσει το χερούλι, και άλλες περιπτώσεις στις οποίες έχει άπειρες λύσεις, δηλαδή υπάρχουν πολλοί συνδυασμοί γωνιών που φέρνουν το επιθυμητό αποτέλεσμα. Στη δεύτερη περίπτωση συνήθως η επίλυση του συστήματος συνδυάζεται με κάποια επιπλέον κριτήρια, όπως, για παράδειγμα, να γίνουν οι ελάχιστες δυνατές αλλαγές σε σχέση με την τρέχουσα κατάσταση ή η τελική στάση του σώματος να είναι «άνετη» εργονομικά, αν πρόκειται για ανθρώπινα μοντέλα. Η διαδικασία επίλυσης του προβλήματος που περιγράψαμε ονομάζεται *αντίστροφη κινηματική* (Tolani κ.ά., 2000) και σήμερα πολλά

εργαλεία τρισδιάστατης μοντελοποίησης και συνθετικής κίνησης υποστηρίζουν τέτοιες διαδικασίες για τον ευκολότερο καθορισμό της κίνησης των μοντέλων.

2.2 Παραμορφώσιμα αντικείμενα

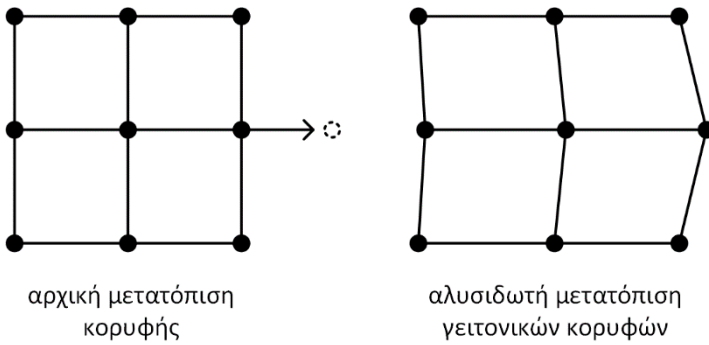
Υπάρχουν πολλά αντικείμενα του πραγματικού κόσμου που δεν διατηρούν αναλλοίωτη τη γεωμετρία τους στη διάρκεια του χρόνου. Μαλακά, εύκαμπτα αντικείμενα, υφάσματα και ρευστά είναι τέτοια παραδείγματα, τα οποία βεβαίως είναι πάρα πολλά για να μη συμπεριλαμβάνονται σε ρεαλιστικούς εικονικούς κόσμους. Μπορούμε, για παράδειγμα, να φανταστούμε έναν εικονικό κόσμο στον οποίο θα περιλαμβάνονται το δέρμα, τα ρούχα και τα μαλλιά των ενσαρκώσεων, οι κυματισμοί στην επιφάνεια της θάλασσας και τα πανιά ενός σκάφους. Όλα τα παραπάνω αντικείμενα δεν είναι άκαμπτα και επομένως η κίνησή τους δεν μπορεί να περιγραφεί μόνο με μεταβολές στον μετασχηματισμό τους· θα πρέπει να συμπεριλαμβάνονται και μεταβολές στην ίδια τη γεωμετρία τους. Οι κινήσεις εύκαμπτων σωμάτων, αν γίνονται με αρκετά φυσικό και «πειστικό» τρόπο, κάνουν πολύ πιο εκφραστικό και «ζωντανό» το τελικό αποτέλεσμα. Υπάρχουν προσεγγίσεις οι οποίες βασίζονται στη φυσική και μοντελοποιούν την αντίδραση των σωμάτων σε εξωτερικές δυνάμεις (Brown κ.ά., 2001). Σε πολλές περιπτώσεις, παρ' όλα αυτά, οι σχεδιαστές προτιμούν να έχουν άμεσο έλεγχο πάνω στην κίνηση των αντικειμένων αντί για το έτοιμο αποτέλεσμα κάποιας προσομοίωσης, και παράλληλα δεν θέλουν να επωμιστεί το σύστημα το επιπλέον υπολογιστικό κόστος της φυσικής μοντελοποίησης. Ανεξαρτήτως του τρόπου με τον οποίο καθορίζονται οι λεπτομέρειες της παραμόρφωσης, το αποτέλεσμα είναι ότι τροποποιούνται οι επιμέρους θέσεις των κορυφών του μοντέλου στον χρόνο. Βεβαίως, το να καθοριστούν μέσω κάποιας χειροκίνητης διαδικασίας όλες οι νέες θέσεις σε κάθε χρονική στιγμή θα ήταν εξαιρετικά επίπονο για τους σχεδιαστές και αμφίβολο το κατά πόσο θα οδηγούσε σε αρκετά φυσικό αποτέλεσμα. Για τον λόγο αυτόν έχουν αναπτυχθεί ορισμένες τεχνικές για την απλοποίηση του καθορισμού της παραμόρφωσης.

2.2.1 Μορφοποίηση

Η πιο απλή περίπτωση παραμόρφωσης είναι μέσω «μορφοποίησης» (morphing). Σύμφωνα με αυτή την προσέγγιση καθορίζουμε διαφορετικές «καταστάσεις» του αντικειμένου ως ξεχωριστά αντικείμενα που διατηρούν την ίδια τοπολογία ακμών και κορυφών, τα οποία ονομάζονται και στόχοι μορφοποίησης (morph targets). Για παράδειγμα, στο πανί του σκάφους που αναφέραμε προηγουμένως θα μπορούσαμε να ορίσουμε δύο ή τρεις «μορφές» στις οποίες φαίνεται περισσότερο ή λιγότερο φουσκωμένο. Οι παραμορφώσεις που μπορεί να κάνει τελικά το αντικείμενο συμπίπτουν με τους αντίστοιχους στόχους μορφοποίησης. Αυτό που συμβαίνει σε επίπεδο συνθετικής κίνησης είναι ότι η κάθε κορυφή ξεχωριστά μετατοπίζεται μέσω διαδικασιών παρεμβολής προς την τελική της θέση που περιγράφεται στο αντικείμενο-στόχο. Η προσέγγιση της μορφοποίησης είναι αρκετά απλή, αλλά έχει το μειονέκτημα ότι οι δυνατές παραμορφώσεις που μπορεί να υποστεί ένα αντικείμενο περιορίζονται σε ένα διακριτό σύνολο μορφών. Η τεχνική χρησιμοποιείται συχνά σε περιπτώσεις όπως οι εκφράσεις προσώπου και η ομιλία, όπου αλλάζει η μορφή του προσώπου σε κάποια προκαθορισμένη έκφραση ή, αντίστοιχα, στην απόδοση κάποιου προκαθορισμένου φθόγγου.

2.2.2 Μετατόπιση κορυφών

Ένας ακόμη απλός τρόπος παραμόρφωσης ενός αντικειμένου είναι μέσω της άμεσης μετατόπισης μίας ή περισσότερων κορυφών. Όμως, όπως αναφέραμε και προηγουμένως, η χειροκίνητη μετατόπιση πολλών κορυφών είναι εξαιρετικά επίπονη διαδικασία. Από την άλλη μεριά η μετατόπιση μίας μόνο κορυφής οδηγεί στις περισσότερες περιπτώσεις σε εξαιρετικά «αφύσικα» αποτελέσματα. Μία καλή συμβιβαστική λύση είναι το να δίνεται η δυνατότητα στον σχεδιαστή να μετατοπίσει μία κορυφή (ή μια ομάδα κορυφών) και η μετατόπιση αυτή να προκαλέσει μικρότερες μετατοπίσεις στις γειτονικές τους κορυφές, οι οποίες με τη σειρά τους θα προκαλέσουν ακόμα μικρότερες στις δικές τους γειτονικές, κ.λπ. Το τελικό αποτέλεσμα θα είναι περισσότερο «φυσικό», διότι η μετατόπιση μίας και μόνο κορυφής θα μπορεί να αλλάξει το σχήμα του αντικειμένου με τέτοιο τρόπο ώστε να επηρεάζεται τελικά μεγάλο μέρος της επιφάνειας, όπως θα συνέβαινε, για παράδειγμα, αν ασκούσαν μια δύναμη στο σημείο αυτό. Ένα παράδειγμα τέτοιας μετατόπισης φαίνεται στην εικόνα 3.2.

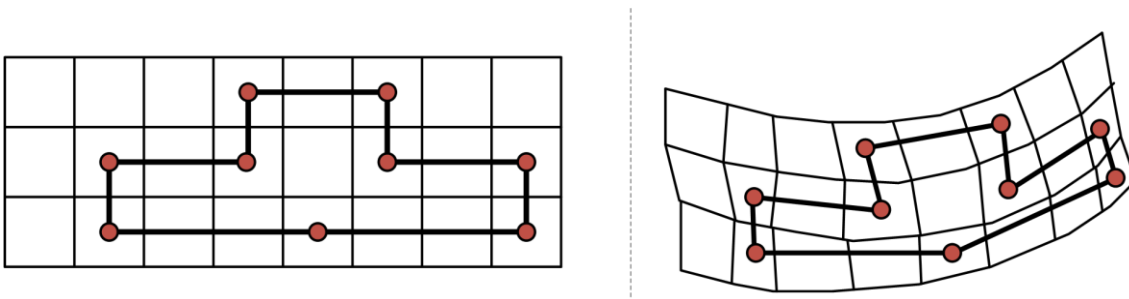


Εικόνα 3.2 Μετατόπιση κορυφής. Πηγή: Parent, 2012.

Ένας απλός τρόπος δημιουργίας μιας τέτοιας «αλυσιδωτής» μετατόπισης είναι ο παρακάτω:

1. Ορίζεται μία *συνάρτηση απόστασης* η οποία περιγράφει πόσο «μακριά» βρίσκονται δύο κορυφές μεταξύ τους. Η συνάρτηση αυτή μπορεί να μετράει τον ελάχιστο αριθμό ακμών που μεσολαβούν μεταξύ των δύο κορυφών ή ακόμα και να υπολογίζει την πραγματική τιμή της απόστασης των δύο κορυφών πάνω στην επιφάνεια του αντικειμένου.
2. Μια *συνάρτηση απόσβεσης* (συνήθως εκθετική) υπολογίζει την κλίμακα μετατόπισης με βάση την απόσταση, έτσι ώστε, όσο μακρύτερα βρίσκεται μια κορυφή από την κορυφή που μετατοπίζεται, τόσο μικρότερη να είναι η μετατόπιση που θα εφαρμοστεί σε αυτήν.
3. Όταν μετατοπίζεται μια κορυφή (ή μια ομάδα κορυφών), υπολογίζεται το διάνυσμα μετατόπισης σε σχέση με την αρχική της θέση.
4. Για κάθε άλλη κορυφή του αντικειμένου υπολογίζεται η απόστασή της από την κορυφή που μετατοπίστηκε και η αντίστοιχη κλίμακα μετατόπισης μέσω της συνάρτησης απόσβεσης. Το τελικό διάνυσμα μετατόπισης, δηλαδή η αρχική μετατόπιση πολλαπλασιασμένη με την κλίμακα, προστίθεται στις συντεταγμένες της.

Μια άλλη, ιδιαίτερα δημοφιλής τεχνική τροποποίησης της μορφής ενός αντικειμένου είναι η παραμόρφωση ελεύθερης μορφής (free form deformation — FFD). Σύμφωνα με αυτή την προσέγγιση, ο σχεδιαστής δεν παραμορφώνει το ίδιο το αντικείμενο μέσω μετατόπισης των κορυφών του, αλλά το τοπικό σύστημα συντεταγμένων βάσει του οποίου καθορίζεται η τελική θέση των κορυφών (Sederberg & Parry, 1986). Για να γίνει καλύτερα κατανοητή αυτή η προσέγγιση, θα παρουσιάσουμε αρχικά το ανάλογο της στις δύο διαστάσεις.



Εικόνα 3.3 Παραμόρφωση ελεύθερης μορφής.

Έστω ότι ορίζουμε ένα πολύγωνο και το τοποθετούμε μέσα σε ένα δισδιάστατο πλέγμα προσανατολισμένο στους άξονες. Τότε η θέση της κάθε κορυφής του πολυγώνου μπορεί να οριστεί σε σχέση με την απόστασή της από τα τέσσερα όρια του κελιού στο οποίο βρίσκεται. Έστω στη συνέχεια ότι παραμορφώνουμε το ίδιο το πλέγμα μετατοπίζοντας τις κορυφές που το ορίζουν. Αν υπολογίσουμε τις νέες θέσεις των κορυφών σε σχέση με τις αντίστοιχες νέες θέσεις και προσανατολισμό των ορίων του κελιού στο οποίο ανήκουν (το οποίο πλέον δεν θα είναι προσανατολισμένο στους άξονες), θα προκύψει το νέο παραμορφωμένο αντικείμενο, του οποίου το σχήμα θα ακολουθεί την αντίστοιχη παραμόρφωση του συστήματος συντεταγμένων (βλ παράδειγμα εικόνας 3.3).

Έστω ότι μια κορυφή P_{uv} απείχε αρχικά οριζόντια απόσταση u από το αριστερό άκρο του κελιού και κάθετη απόσταση v από το κάτω άκρο. Έστω επίσης P_{00} , P_{01} , P_{10} , P_{11} οι νέες, παραμορφωμένες, θέσεις του κάτω αριστερά, πάνω αριστερά, κάτω δεξιά και πάνω δεξιά άκρου του κελιού. Τότε, η τελική θέση της κορυφής θα υπολογιστεί μέσω διγραμμικής (bilinear) παρεμβολής:

$$P_{uv} = (1-u)(1-v)P_{00} + (1-u)vP_{01} + u(1-v)P_{10} + uvP_{11}$$

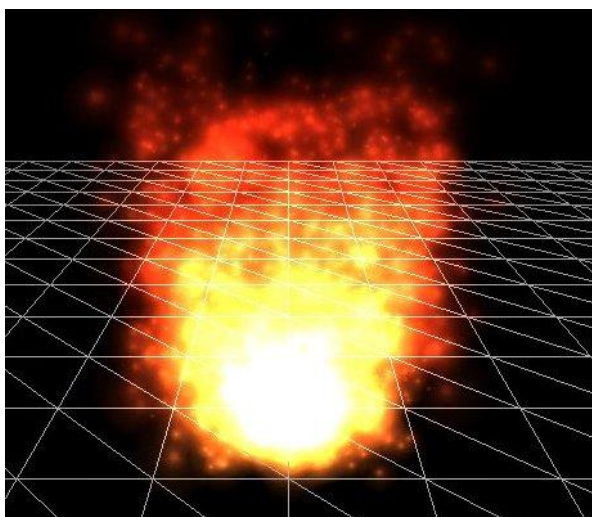
Κάτι αντίστοιχο με το παραπάνω παράδειγμα συμβαίνει και στην περίπτωση της παραμόρφωσης τρισδιάστατων αντικειμένων. Το αντικείμενο τοποθετείται σε ένα τρισδιάστατο πλέγμα, το οποίο μπορεί να παραμορφωθεί, και η τελική θέση των ακμών καθορίζεται παραμετρικά σε σχέση με τα όρια του πλέγματος. Η μόνη διαφορά σε σχέση με τις δύο διαστάσεις είναι ότι τα κελιά καθορίζονται πλέον από οκτώ σημεία και για κάθε κορυφή του αντικειμένου μετράμε τρεις τιμές απόστασης από τα όρια του κελιού u , v και w , μία για κάθε διάσταση. Στην περίπτωση λεπτομερών και περίπλοκων αντικειμένων που περιλαμβάνουν εκατοντάδες κορυφές είναι τελικά πολύ πιο εύκολο και διαισθητικό για τους σχεδιαστές το να παραμορφώνουν ένα εξωτερικό πλέγμα σε σχέση με τις επιμέρους κορυφές.

2.3 Συστήματα σωματιδίων

Τα *συστήματα σωματιδίων* (*particle systems*) αποτελούν μία ισχυρή τεχνική για τη μοντελοποίηση φυσικών φαινομένων όπως νερό, βροχή, φωτιά, καπνός, εκρήξεις, σύννεφα κ.ά. (Reeves, 1983). Όπως περιγράψαμε και στο Κεφάλαιο 2, η τεχνική περιλαμβάνει τη μοντελοποίηση ασαφών αντικειμένων μέσω ενός πολύ μεγάλου αριθμού σημειακών στοιχείων, των σωματιδίων (*particles*). Τα συστήματα σωματιδίων είναι εξαιρετικά χρήσιμα στην παραγωγή συνθετικής κίνησης αντικειμένων τα οποία, όχι απλώς παραμορφώνονται, αλλά μεταβάλλεται διαρκώς η γεωμετρία τους. Ας φανταστούμε, για παράδειγμα, ένα σύννεφο καπνού. Η μοντελοποίησή του ως παραμορφώσιμου πολυγωνικού στερεού δεν θα έδινε ιδιαίτερα ρεαλιστικά αποτελέσματα ούτε σε επίπεδο εμφάνισης ούτε σε επίπεδο κίνησης. Οπτικά ο καπνός δεν θυμίζει κάποιο συμπαγές αντικείμενο που φωτοσκοπάζεται ομαλά από τις πηγές φωτός. Περιλαμβάνει περιοχές περισσότερο πυκνές και άλλες πιο αραιές, και οι περιοχές πυκνότητας μεταβάλλονται σταδιακά κατά την κίνησή του, κάτι που κάνει τα πράγματα ακόμα πιο περίπλοκα. Με τη χρήση συστημάτων σωματιδίων μπορεί να αποδοθεί πολύ καλύτερα αυτό το φαινόμενο. Δεν υπάρχει ένα συμπαγές αντικείμενο αλλά πολλαπλά σωματίδια τα οποία, λόγω μεγάλου αριθμού και με τη χρήση των κατάλληλων *shaders*, δίνουν οπτικά την εικόνα αέριας μάζας. Επιπλέον, κάθε σημειακό αντικείμενο κινείται ανεξάρτητα από τα άλλα, υπακούοντας συνήθως σε κάποιους κανόνες οι οποίοι μοντελοποιούν το φυσικό φαινόμενο (στην περίπτωση των αερίων θα εισαχθεί και «θόρυβος»/τυχαιότητα στην κίνηση). Αυτό κάνει τελικά το συνολικό σχήμα της μάζας σωματιδίων εξαιρετικά εύπλαστο, ώστε να αποδίδει αρκετά καλά το αντίστοιχο φυσικό φαινόμενο. Η εικόνα 3.4. απεικονίζει μια φωτιά κατασκευασμένη από σύστημα σωματιδίων.

Στα συστήματα σωματιδίων υπάρχει συνήθως μια διαδικασία που «γεννάει» σωματίδια και μια παράλληλη διαδικασία προσομοίωσης που καθορίζει την κίνησή τους και τον χρόνο ζωής τους.

Για κάθε φαινόμενο που θέλουμε να μοντελοποιήσουμε μέσω σωματιδίων ορίζουμε αρχικά τη «γεννήτρια» *σωματιδίων* (*particle emitter*), δηλαδή ένα σημείο στον χώρο από το οποίο δημιουργούνται (συνήθως «εκτοξεύονται») τα σωματίδια. Η γεννήτρια αυτή περιλαμβάνει παραμέτρους όπως τον ρυθμό δημιουργίας νέων σωματιδίων, το αρχικό διάνυσμα ταχύτητας, τον χρόνο ζωής των σωματιδίων, στοιχεία της εμφάνισής τους (χρώμα, διαφάνεια, κ.λπ.) κ.ά. Συνήθως οι τιμές των παραπάνω παραμέτρων δεν είναι σταθερές, αλλά ορίζονται με τυχαίο τρόπο μέσα σε κάποιο καθορισμένο εύρος, με αποτέλεσμα κάθε σωματίδιο να έχει ελαφρώς διαφοροποιημένη συμπεριφορά και το τελικό αποτέλεσμα να δείχνει περισσότερο «φυσικό». Η γεννήτρια σωματιδίων είναι και αυτή ένα αντικείμενο του γράφου σκηνής, και ως τέτοιο είναι δυνατόν η θέση και ο προσανατολισμός του να επηρεάζεται από άλλα, κινούμενα αντικείμενα. Για παράδειγμα, η εξάτμιση ενός αυτοκινήτου θα μπορούσε να συνδέεται με μια γεννήτρια σωματιδίων που εκπέμπει σωματίδια καπνού.



Εικόνα 3.4 Φωτιά μοντελοποιημένη με σύστημα σωματιδίων.

Κατά τη διάρκεια της προσομοίωσης καθορίζεται ο αριθμός των νέων σωματιδίων που θα εισαχθούν στη σκηνή την κάθε χρονική στιγμή, η θέση τους και η αρχική τους ταχύτητα. Όλα τα σωματίδια της σκηνής ελέγχονται κατά πόσο έχουν ξεπεράσει τον χρόνο ζωής τους και, στην περίπτωση αυτή, διαγράφονται. Στη συνέχεια, ακολουθώντας κάποια φυσική προσομοίωση υπολογίζεται η κίνησή τους και καθορίζεται η νέα τους θέση τη συγκεκριμένη χρονική στιγμή. Το μοντέλο προσομοίωσης που θα χρησιμοποιηθεί σχετίζεται με το φαινόμενο που θέλουν να μοντελοποιήσουν τα σωματίδια. Μπορεί να συνοψολογίζεται η βαρύτητα (π.χ. το νερό που εκτοξεύεται από ένα σιντριβάνι), η επίδραση του αέρα (π.χ. σε κίνηση αερίων μαζών), οι συγκρούσεις μεταξύ των σωματιδίων, οι συγκρούσεις μεταξύ σωματιδίων και άλλων αντικειμένων της σκηνής κ.ά. Όταν υπολογιστεί η τελική τους θέση, τα σωματίδια απεικονίζονται συνήθως ως επιφάνειες Billboard (βλ. Κεφ. 2) και για την αναπαράστασή τους χρησιμοποιούνται διάφορα οπτικά εφέ, ανάλογα με φαινόμενο που απεικονίζουν.

2.4 Συνθετικοί χαρακτήρες

Η κίνηση συνθετικών χαρακτήρων είναι ένα απαραίτητο στοιχείο των εικονικών κόσμων (Vosinakis & Panayiotopoulos, 2001). Ακόμα κι αν όλη η υπόλοιπη σκηνή είναι στατική, οι ενσαρκώσεις των χρηστών θα πρέπει να είναι σε θέση να κινούνται. Καταρχήν να μπορούν να πλοηγούνται βαδίζοντας με φυσικό τρόπο στο περιβάλλον, αλλά και να αποτυπώνεται κάπως η αλληλεπίδρασή τους με τα αντικείμενα του περιβάλλοντος, η εστίαση του βλέμματός τους, η ομιλία τους ή και άλλες κινήσεις μη λεκτικής επικοινωνίας. Η ρεαλιστική απεικόνιση της κίνησης των ενσαρκώσεων συμβάλλει στην καλύτερη επίγνωση της θέσης και της δραστηριότητάς τους, στην επικοινωνία μεταξύ των χρηστών και γενικότερα στην αίσθηση της συν-παρουσίας. Εκτός όμως από τις ενσαρκώσεις, ενδέχεται να προκύψει η ανάγκη και για άλλες ανθρώπινες ή ανθρωπόμορφες οντότητες των οποίων η κίνηση θα ελέγχεται από τον υπολογιστή. Για παράδειγμα, για λόγους αύξησης του ρεαλισμού θα μπορούσε ένας δημόσιος χώρος, για παράδειγμα ένα εμπορικό κέντρο, να αποκτήσει ανθρώπινους χαρακτήρες που κινούνται με κάποιον προκαθορισμένο τρόπο σε αυτόν. Σε άλλες περιπτώσεις μπορεί να χρειάζονται οντότητες οι οποίες αναλαμβάνουν πρωτοβουλίες και αλληλεπιδρούν με τους χρήστες, όπως, για παράδειγμα, εικονικοί ξεναγοί, εικονικοί διδάσκοντες, χαρακτήρες σε κόσμους παιχνιδιού κ.ά.

Το ιδιαίτερο χαρακτηριστικό με την κίνηση των ανθρώπων και γενικότερα των ζωντανών οργανισμών είναι ότι αφενός αποτελούν αρθρωτά αντικείμενα, συνήθως με μεγάλο αριθμό αρθρώσεων, και αφετέρου υφίστανται παραμορφώσεις κατά την κίνησή τους λόγω του δέρματός τους. Ενώ λοιπόν ένας ρομποτικός βραχίονας, που είναι μια άλλη περίπτωση αρθρωτού αντικειμένου, αποτελείται από διακριτά άκαμπτα αντικείμενα, οι άνθρωποι έχουμε συνεχές δέρμα. Αν απομονώσουμε ένα τμήμα του σώματος σε ένα τρισδιάστατο μοντέλο ανθρώπου και το περιστρέψουμε, αν, για παράδειγμα, μεταβάλουμε τον προσανατολισμό του ώμου, που θα εξαναγκάσει όλο το χέρι να περιστραφεί, τότε μοιραία θα δημιουργηθούν θραύσεις/ασυνέχειες στην επιφάνεια του μοντέλου και το αποτέλεσμα δεν θα δείχνει φυσικό. Για να

προκύπτουν περισσότερο φυσικά αποτελέσματα, οι τεχνικές μοντελοποίησης των συνθετικών χαρακτήρων συνδυάζουν ταυτόχρονα πρακτικές από αρθρωτά και παραμορφώσιμα αντικείμενα.

Η βασική μέθοδος κίνησης ανθρώπινων και άλλων χαρακτήρων είναι η *σκελετική κίνηση* (*skeletal animation*), σύμφωνα με την οποία ορίζεται ένας σκελετός που καθορίζει τις δυνατότητες κίνησης του χαρακτήρα και ένα δέρμα που συνδέεται με τον σκελετό αυτόν και παραμορφώνεται κατά την κίνησή του παραμένοντας συνεχές. Ο σκελετός είναι μια ιεραρχία από αρθρώσεις και τμήματα, τα οποία ονομάζονται *οστά* (*bones*), και κατασκευάζεται όπως όλα τα αρθρωτά αντικείμενα, με τη διαφορά ότι τα οστά δεν απεικονίζονται απευθείας στο περιβάλλον· απλά η θέση και ο προσανατολισμός τους επηρεάζουν την τελική μορφή του δέρματος. Το δέρμα, από την άλλη μεριά, είναι ένα μοναδικό πολυγωνικό αντικείμενο το οποίο παραμορφώνεται ακολουθώντας την κίνηση του σκελετού. Η κάθε κορυφή του δέρματος συσχετίζεται με ένα ή περισσότερα οστά με χρήση βαρών που προσδιορίζουν τον βαθμό στον οποίο η κίνηση κάθε οστού επηρεάζει τη θέση της. Αν η κορυφή v ακολουθεί αποκλειστικά την κίνηση ενός μοναδικού οστού (έστω i), τότε η τιμή του βάρους w_i της συγκεκριμένης κορυφής θα είναι 1 και όλα τα υπόλοιπα βάρη μηδέν. Αν όμως όλες οι κορυφές του δέρματος συσχετίζονται μόνο με ένα οστό, το αποτέλεσμα της κίνησης θα θυμίζει την κίνηση άκαμπτων σωμάτων και θα εμφανίζει αφύσικες παραμορφώσεις στην περιοχή των αρθρώσεων. Για να επιτευχθεί ένα πιο ρεαλιστικό αποτέλεσμα με σταδιακή παραμόρφωση των πολυγώνων κοντά στις αρθρώσεις κατά την κίνηση των αντίστοιχων οστών, κάθε κορυφή θα πρέπει να εξαρτάται από πολλαπλά κοντινά οστά, επομένως να περιλαμβάνει παραπάνω από ένα μη μηδενικά βάρη, των οποίων το άθροισμα θα πρέπει να ισούται με 1. Η τελική θέση της κάθε κορυφής θα προκύπτει από το σταθμισμένο μέσο των αντίστοιχων μετασχηματισμών που προκύπτουν από τα οστά που την επηρεάζουν.

Τέλος, εκτός από το ανθρώπινο δέρμα, ιδιαίτερες περιπτώσεις κίνησης που σχετίζονται με συνθετικούς χαρακτήρες είναι και τα μαλλιά, τα ρούχα και το πρόσωπο. Και στις τρεις αυτές περιπτώσεις έχουν αναπτυχθεί περίπλοκα φυσικά μοντέλα που δίνουν ιδιαίτερα ρεαλιστικά αποτελέσματα. Αυτό μπορεί να το παρατηρήσει κάποιος και στις σύγχρονες κινηματογραφικές παραγωγές που βασίζονται σε τρισδιάστατη συνθετική κίνηση και εμπεριέχουν χαρακτήρες. Δυστυχώς, στην περίπτωση των εικονικών κόσμων, που υπάρχει ο ισχυρός περιορισμός της εκτέλεσης σε πραγματικό χρόνο, το υπολογιστικό κόστος ρεαλιστικής κίνησης των παραπάνω εξακολουθεί να είναι απαγορευτικό. Συνήθως χρησιμοποιούνται παραμορφώσιμα αντικείμενα συνδυασμένα με ορισμένα στοιχεία φυσικής προσομοίωσης (π.χ. κίνηση μαλλιών και ρούχων κατά το βάδισμα) και δίνουν ένα αρκετά καλό αποτέλεσμα.

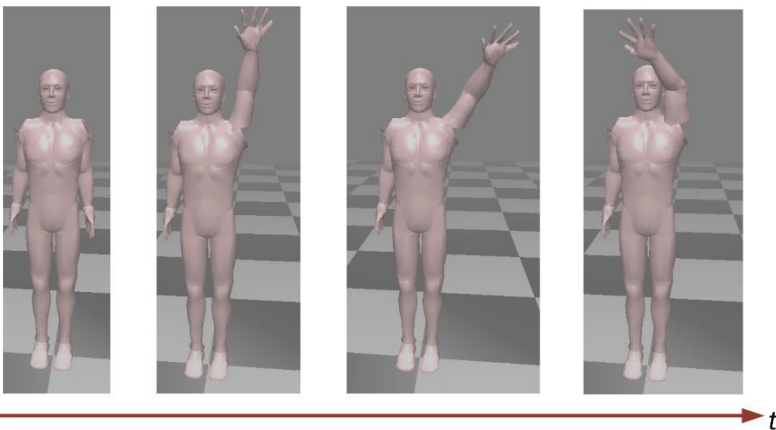
3 Κατασκευή προκαθορισμένων κινήσεων

Ένας μεγάλος αριθμός κινήσεων που εκτελούνται στους εικονικούς κόσμους είναι προκαθορισμένοι. Αυτό πρακτικά σημαίνει ότι ο σχεδιαστής έχει προκαθορίσει πλήρως την κίνηση που θα αποδοθεί σε ένα ή περισσότερα αντικείμενα και το μόνο που αλλάζει είναι ο χρόνος έναρξης και λήξης της κίνησης και τα αντικείμενα στα οποία αυτή αποδίδεται. Ένα πολύ χαρακτηριστικό παράδειγμα τέτοιας κίνησης είναι το βάδισμα των χαρακτήρων. Οι χαρακτήρες μπορεί να μετακινούνται δυναμικά σε διάφορες περιοχές του περιβάλλοντος, είτε λόγω επιλογών του χρήστη, αν πρόκειται για ενσάρκωση, είτε λόγω κάποιου μοντέλου συμπεριφοράς, αν πρόκειται για χαρακτήρα ελεγχόμενο από τον υπολογιστή. Όμως, κατά τη διάρκεια της μετακίνησής τους δεν αλλάζει απλά η συνολική θέση και ο προσανατολισμός τους, καθώς κάτι τέτοιο θα ήταν αφύσικο· αντίθετα, εκτελούν ταυτόχρονα κάποια κίνηση βαδίσματος η οποία είναι συμβατή με τον ρυθμό μετακίνησης, με αποτέλεσμα η συνολική κίνηση να δείχνει φυσική. Αυτός δεν είναι βεβαίως ο τρόπος με τον οποίο βαδίζουμε στην πραγματικότητα, όμως το να χρησιμοποιηθεί ένα περίπλοκο φυσικό μοντέλο για το βάδισμα των χαρακτήρων θα είχε πολύ υψηλό υπολογιστικό κόστος για εφαρμογή πραγματικού χρόνου. Κατά συνέπεια, το βάδισμα των χαρακτήρων συνδυάζει δύο κινήσεις: μια δυναμική μετακίνηση και μια προκαθορισμένη κίνηση στατικού βαδίσματος. Για λόγους ρεαλισμού θα μπορούσε κάποιος να χρησιμοποιεί παραπάνω από μία προκαθορισμένες κινήσεις, για παράδειγμα αργό και γρήγορο βάδισμα, βάδισμα σε ανηφορικό δρόμο κ.ά., και να επιλέγει την κατάλληλη ανάλογα με τις συνθήκες. Ένας χαρακτήρας μπορεί πέρα από το βάδισμα να έχει και άλλες αποθηκευμένες έτοιμες κινήσεις, όπως το κάθισμα, ο χαιρετισμός, τα παλαμάκια κ.λπ., ανάλογα με τις ανάγκες του περιβάλλοντος. Επιπλέον, προκαθορισμένες κινήσεις μπορούν να αποδοθούν και σε πολλές άλλες οντότητες ενός εικονικού κόσμου, εάν η συμπεριφορά τους δεν μεταβάλλεται από εξωτερικές συνθήκες, για παράδειγμα η κίνηση των δεικτών του ρολογιού, ένα τρένο που εκτελεί προκαθορισμένη πορεία, μια περιστρεφόμενη πόρτα κ.ά.

Στην κατασκευή προκαθορισμένων κινήσεων οι βασικές τάσεις είναι δύο: Είτε ο σχεδιαστής να καθορίσει ο ίδιος τις λεπτομέρειες της κίνησης, ορίζοντας κάποιες θέσεις-κλειδιά και αφήνοντας τον υπολογιστή να παρεμβάλει τα ενδιάμεσα πλάνα, είτε να γίνει σύλληψη της πραγματικής κίνησης ενός ανθρώπου και τα δεδομένα να αξιοποιηθούν για την κίνηση ψηφιακών χαρακτήρων.

3.1 Κίνηση με πλάνα-κλειδιά

Η κίνηση με πλάνα-κλειδιά (*keyframing*) έχει τις ρίζες της στην παραδοσιακή διαδικασία παραγωγής κινουμένων σχεδίων την εποχή που οι σκηνές ζωγραφίζονταν με το χέρι. Τότε, οι έμπειροι καλλιτέχνες δημιουργούσαν τα στιγμιότυπα που περιείχαν κρίσιμες μεταβολές στην κίνηση των χαρακτήρων ή στις εκφράσεις τους, τα λεγόμενα πλάνα-κλειδιά (*keyframes*), και άφηναν τους λιγότερο έμπειρους να ζωγραφίσουν όλα τα ενδιάμεσα πλάνα ώστε να παραχθεί συνεχής κίνηση. Στα συστήματα συνθετικής κίνησης μέσω υπολογιστή η λογική της κίνησης με πλάνα-κλειδιά είναι παραπλήσια. Ο σχεδιαστής καθορίζει σε συγκεκριμένες χρονικές στιγμές τις θέσεις και τους προσανατολισμούς των αντικειμένων που θα κινηθούν, και ο υπολογιστής φροντίζει να παρεμβάλει τα ενδιάμεσα πλάνα με ομαλό τρόπο βασισμένος σε τεχνικές παρεμβολής (Lasseter, 1987). Για παράδειγμα, θα μπορούσε να ορίσει ο σχεδιαστής ότι τη χρονική στιγμή $t = 0$ το αριστερό χέρι ενός χαρακτήρα είναι κατεβασμένο, την $t = 1.5$ σηκώνεται ψηλά, την $t = 2.0$ κινείται ελαφρά προς τα αριστερά και την $t = 2.5$ αντίστοιχα προς τα δεξιά. Αυτό θα οδηγούσε σε μία κίνηση χαιρετισμού διάρκειας δύο και μισού δευτερολέπτων. Στην εικόνα 3.5 απεικονίζονται οι τέσσερις παραπάνω θέσεις-κλειδιά στο πρόγραμμα συνθετικής κίνησης QAnimator. Η κίνηση αυτή θα μπορούσε να εφαρμοστεί σε πολλά διαφορετικά μοντέλα χαρακτήρων, ιδιαίτερα αν οι αναλογίες μεταξύ των οστών του σκελετού είναι παραπλήσιες, και θα μπορούσε να εκτελείται παράλληλα και με άλλες κινήσεις του χαρακτήρα, όπως, για παράδειγμα, το βάδισμα. Η προσέγγιση της κίνησης με πλάνα-κλειδιά είναι επίπονη γιατί ο καθορισμός της κίνησης είναι αποκλειστικά στην ευθύνη του σχεδιαστή και αυτός τελικά πρέπει να διασφαλίσει τη φυσικότητα του αποτελέσματος. Όμως παραμένει η δημοφιλέστερη τεχνική, ακριβώς επειδή δίνει στον σχεδιαστή όλο τον έλεγχο, άρα του επιτρέπει να βελτιώσει το αποτέλεσμα, να εισάγει επιπλέον στοιχεία προσωπικότητας στην κίνηση και στις εκφράσεις κ.ά. Στη συνέχεια θα εξετάσουμε ορισμένους βασικούς μηχανισμούς παρεμβολής που χρησιμοποιούνται σε συστήματα κίνησης με πλάνα-κλειδιά.



Εικόνα 3.5 Κίνηση με πλάνα-κλειδιά στο πρόγραμμα QAnimator.

Η πιο απλή περίπτωση παρεμβολής θέσης είναι η γραμμική παρεμβολή. Σύμφωνα με αυτή την προσέγγιση, οι ενδιάμεσες θέσεις ενός αντικειμένου μεταξύ δύο πλάνων-κλειδιών υπολογίζονται με τρόπο τέτοιο ώστε το αντικείμενο να κινείται με σταθερή ταχύτητα και σε ευθεία γραμμή μεταξύ των δύο θέσεων. Έστω λοιπόν ότι για τη θέση ενός αντικειμένου έχουμε ορίσει ότι τη χρονική στιγμή t_1 θα είναι v_1 και τη χρονική στιγμή t_2 θα είναι v_2 και θέλουμε να υπολογίσουμε τη θέση του v τη χρονική στιγμή $t \in [t_1, t_2]$. Τότε, η αναλογία μεταξύ της χρονικής διάρκειας που έχει ήδη διανυθεί και της συνολικής χρονικής διάρκειας μέχρι το επόμενο πλάνα-κλειδί θα είναι:

$$\lambda = \frac{(t-t_1)}{(t_2-t_1)}$$

Αν λοιπόν μετατοπίσουμε το διάνυσμα κατά την ίδια αναλογία πάνω στην ευθεία που ορίζουν τα δύο πλάνα-κλειδιά, μπορούμε να υπολογίσουμε την τελική του θέση ως εξής:

$$v = (1-\lambda)v_0 + \lambda v_1$$

Η γραμμική παρεμβολή, αν και χρησιμοποιείται συχνότερα από όλες τις άλλες, δεν δίνει καλά οπτικά αποτελέσματα σε πιο περίπλοκες κινήσεις. Αν, για παράδειγμα, αλλάζει σημαντικά η διεύθυνση της κίνησης από ένα ζευγάρι πλάνων-κλειδιών στο επόμενο, τότε με γραμμική παρεμβολή η αλλαγή αυτή θα φανεί αρκετά απότομη και πιθανότατα αφύσικη.

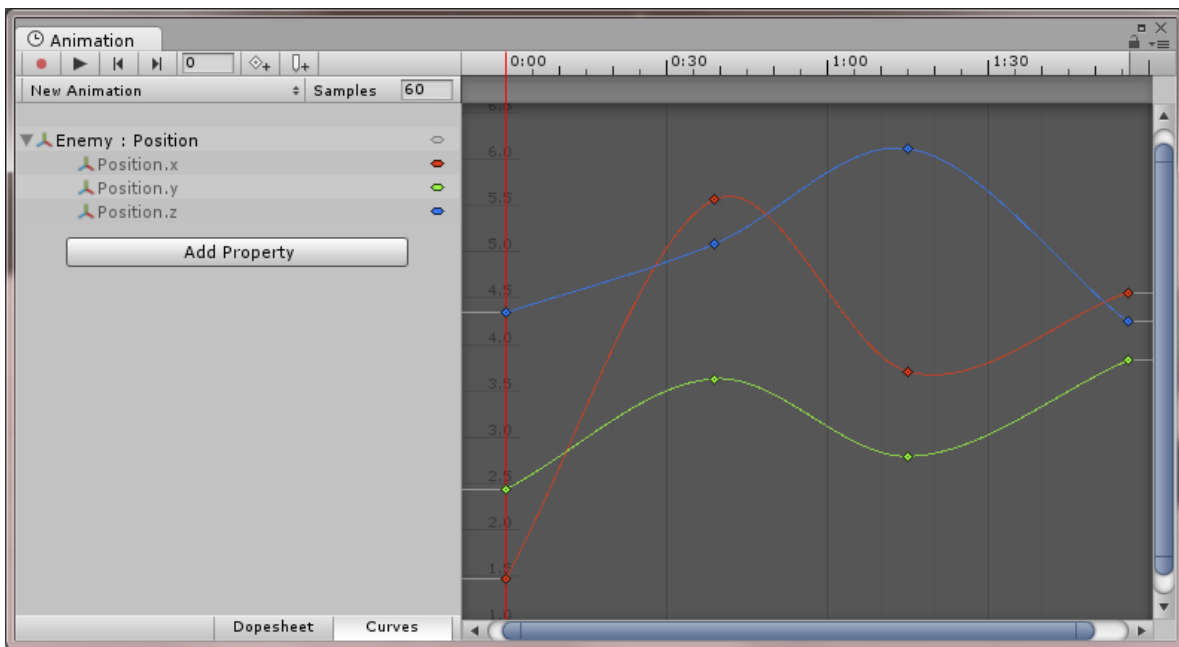
Η χρήση μη-γραμμικών παρεμβολών μπορεί να αντιμετωπίσει το παραπάνω πρόβλημα παράγοντας ομαλές καμπύλες κίνησης. Για παράδειγμα, μπορεί να χρησιμοποιηθεί τετραγωνική παρεμβολή σύμφωνα με τον παρακάτω τύπο:

$$v = (1-\lambda)^2 v_0 + 2\lambda(1-\lambda)v' + \lambda^2 v_1$$

όπου το v' αποτελεί ένα επιπλέον σημείο ελέγχου που τροποποιεί την καμπύλη. Στην περίπτωση αυτήν είναι στην ευθύνη του σχεδιαστή να καθορίσει τα σημεία ελέγχου. Αντίστοιχα, στην κυβική παρεμβολή ο σχεδιαστής θα πρέπει να καθορίσει ένα ζευγάρι σημείων ελέγχου για κάθε ζεύγος πλάνων-κλειδιών. Μια πιο απλή περίπτωση είναι η παρεμβολή Hermite, σύμφωνα με την οποία απαιτούνται οι πρώτες παράγωγοι, δηλαδή οι αντίστοιχες διευθύνσεις κίνησης, στα δύο σημεία παρεμβολής. Τότε η παρεμβολή Hermite μπορεί να επιτευχθεί με τον παρακάτω πίνακα:

$$v(t) = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ R_1 \\ R_2 \end{bmatrix}$$

όπου R_1 και R_2 είναι οι αντίστοιχες πρώτες παράγωγοι των σημείων v_1 και v_2 .



Εικόνα 3.6 Επεξεργασία καμπύλων κίνησης στο πρόγραμμα Unity.

Μία επέκταση της παρεμβολής Hermite είναι οι καμπύλες Catmull-Rom (Twigg, 2003), στις οποίες υπολογίζονται αυτόματα οι παράγωγοι σε κάθε σημείο, με αποτέλεσμα να μην απαιτείται από τον σχεδιαστή να τις καθορίσει χειροκίνητα. Με τη χρήση αυτής της προσέγγισης μπορεί από ένα σύνολο θέσεων στα αντίστοιχα πλάνα-κλειδιά να προκύψει αυτόματα μια ομαλή καμπύλη κίνησης. Βεβαίως, το μειονέκτημα είναι ότι ο σχεδιαστής δεν θα έχει καμία δυνατότητα τροποποίησης του αποτελέσματος. Για τον λόγο αυτόν, συνήθως στα προγράμματα τρισδιάστατης μοντελοποίησης δίνεται η δυνατότητα στον χρήστη να ελέγξει και να τροποποιήσει την παραγόμενη καμπύλη στην περίπτωση μη γραμμικών κινήσεων. Στην εικόνα 3.6 απεικονίζεται το παράθυρο επεξεργασίας καμπύλων κίνησης στο πρόγραμμα Unity.

Στην περίπτωση των περιστροφών θα μπορούσε κάποιος εύλογα να υποθέσει ότι αρκεί να γίνει γραμμική παρεμβολή στις τρεις γωνίες Euler για να παραχθεί ομαλή περιστροφή. Στην πράξη όμως τα πράγματα δεν είναι έτσι. Αν μεταβάλλεται η γωνία ενός μόνο άξονα, π.χ. ένας χαρακτήρας στρέφει το κεφάλι του προς τα αριστερά, τότε η γραμμική παρεμβολή στη γωνία μπορεί να δουλέψει. Αν όμως η μεταβολή στον προσανατολισμό αφορά παραπάνω από έναν άξονες, τότε οι γωνίες Euler είναι πιθανό να δώσουν λανθασμένα και «αφύσικα» αποτελέσματα. Το πρόβλημα αυτό λύνεται με τη χρήση των quaternions, τα οποία μπορούν να οδηγήσουν σε ομαλές μεταβάσεις μεταξύ περιστροφών σε όλες τις περιπτώσεις.

Η απλή γραμμική παρεμβολή μεταξύ δύο quaternions χρησιμοποιώντας τον παραπάνω τύπο δεν θα οδηγήσει σε περιστροφή με σταθερή ταχύτητα, αλλά θα προκύπτει μια κίνηση που επιταχύνεται στο μέσο της διαδρομής. Για να έχει η παρεμβολή σταθερή ταχύτητα περιστροφής χρησιμοποιείται μια τεχνική που ονομάζεται «σφαιρική γραμμική παρεμβολή» (spherical linear interpolation – SLERP), σύμφωνα με την οποία, αν ω η γωνία μεταξύ του αρχικού και του τελικού quaternion q_1 και q_2 , τότε η περιστροφή στη θέση t υπολογίζεται από τον τύπο:

$$q = q_1 \frac{\sin((1-t)\omega)}{\sin \omega} + q_2 \frac{\sin(t\omega)}{\sin \omega}$$

3.2 Σύλληψη κίνησης

Όπως αναφέρθηκε και προηγουμένως, η κίνηση με πλάνα-κλειδιά μπορεί να είναι αρκετά επίπονη για τον σχεδιαστή και δεν διασφαλίζει τη φυσικότητα του αποτελέσματος. Αντίθετα, με τη *σύλληψη κίνησης (motion capture)* μπορούν να επιτευχθούν πολύ πιο εύκολα τα επιθυμητά δεδομένα κίνησης, και μάλιστα με περισσότερο αληθοφανή αποτελέσματα, καθώς θα προέρχονται από αντίστοιχες κινήσεις στον φυσικό κόσμο. Η ιδέα πίσω από τη σύλληψη κίνησης, που ονομάζεται και *συνθετική κίνηση ερμηνείας (performance animation)*, είναι η καταγραφή διαδοχικών μετρήσεων της θέσης και του προσανατολισμού αντικειμένων του φυσικού κόσμου και η αποθήκευση της πληροφορίας αυτής σε μορφή χρησιμοποιήσιμη από υπολογιστή. Στη συνέχεια, τα καταγεγραμμένα δεδομένα αξιοποιούνται σε αντίστοιχα αντικείμενα του εικονικού κόσμου (Gleicher, 1999). Η συνηθέστερη χρήση της σύλληψης κίνησης είναι για την καταγραφή της ανθρώπινης κίνησης, που μπορεί να περιλαμβάνει ανεξάρτητες κινήσεις (π.χ. βάδισμα, χαιρετισμό, άλμα) ή και αλληλεπίδραση με άλλα αντικείμενα του περιβάλλοντος (π.χ. χειρισμό ενός μηχανήματος). Στην περίπτωση της σύλληψης ανθρώπινης κίνησης, ένας ηθοποιός εκτελεί τις επιθυμητές δράσεις και κατά τη διάρκεια της κίνησής του καταγράφονται οι μεταβολές σε αρθρώσεις του σκελετού του που είναι σημαντικές για την απόδοση των ενεργειών του και η συνολική θέση και ο προσανατολισμός του. Εφόσον περιλαμβάνεται και αλληλεπίδραση με άλλα αντικείμενα, θα πρέπει να είναι γνωστή και η σχετική τους θέση ως προς τον ηθοποιό καθώς και οι μεταβολές στα κινούμενα μέρη των αντικειμένων, αν υπάρχουν τέτοια.

Οι δύο βασικές προσεγγίσεις που χρησιμοποιούνται σήμερα είναι η *μαγνητική* και η *οπτική*. Και στις δύο περιπτώσεις χρησιμοποιούνται ειδικοί αισθητήρες που τοποθετούνται πάνω στο σώμα του ηθοποιού, συνηθέστερα στη θέση των αρθρώσεων, και βοηθούν στην καταγραφή της θέσης και του προσανατολισμού του αντίστοιχου τμήματος του σώματος.

Οι μαγνητικοί αισθητήρες, παρόλο που θεωρητικά παρέχουν πολύ μεγαλύτερη ακρίβεια, είναι ευαίσθητοι σε παρεμβολές του μαγνητικού πεδίου. Αν, για παράδειγμα, υπάρχουν μεταλλικά αντικείμενα στον χώρο ή στην ένδυση του ηθοποιού, αυτά εισάγουν θόρυβο στο τελικό αποτέλεσμα επηρεάζοντας την ακρίβεια. Συνήθως για τη μετάδοση της πληροφορίας χρησιμοποιούνται καλώδια, τα οποία επίσης περιορίζουν σημαντικά την ελευθερία των κινήσεων του ηθοποιού. Το μεγάλο πλεονέκτημα των συστημάτων αυτών είναι ότι τα αποτελέσματα είναι διαθέσιμα σε πραγματικό χρόνο.

Από την άλλη μεριά, τα οπτικά συστήματα απαιτούν από τον ηθοποιό να φοράει μικρά αντανακλαστικά σημάδια (reflective markers), η θέση των οποίων καταγράφεται από πολλαπλές σταθερές κάμερες. Ο συνδυασμός των πληροφοριών καταγραφής από τρεις τουλάχιστον κάμερες επιτρέπει τον ακριβή προσδιορισμό της θέσης του σημαδιού στον τρισδιάστατο χώρο. Η αντιστοίχιση των σημαδιών αυτών σε έναν προκαθορισμένο ανθρώπινο σκελετό επιτρέπει τον υπολογισμό των αντίστοιχων γωνιών των αρθρώσεων. Το πρόβλημα με τα οπτικά συστήματα είναι ότι απαιτείται αρκετή επεξεργασία των οπτικών δεδομένων για να παραχθούν τα τελικά δεδομένα κίνησης που θα αξιοποιηθούν από τους ψηφιακούς χαρακτήρες, ενώ τυχόν επικαλύψεις σημαδιών κατά τη διάρκεια της κίνησης μπορεί να προκαλέσουν προβλήματα στην καταγραφή τους, με αποτέλεσμα η θέση τους να υπολογιστεί προσεγγιστικά. Όμως επιτρέπουν μεγαλύτερη ελευθερία κινήσεων σε σχέση με τα μαγνητικά συστήματα. Τέλος, σήμερα αποτελούν αντικείμενο έρευνας και οπτικά συστήματα καταγραφής ανθρώπινης κίνησης χωρίς τη χρήση σημαδιών (markerless). Για παράδειγμα, το Kinect της Microsoft επιτρέπει την καταγραφή της στάσης του σώματος σε σχεδόν πραγματικό χρόνο (Berger κ.ά., 2011). Τα συστήματα αυτά όμως σήμερα δεν έχουν την απαιτούμενη ακρίβεια για την καταγραφή καλής ποιότητας δεδομένων κίνησης.

Η σύλληψη κίνησης έχει αποδειχτεί μια πολύ αποδοτική τεχνική για τη συνθετική κίνηση ανθρώπων και ανθρωποειδών χαρακτήρων. Τα δεδομένα σύλληψης κίνησης κρατούν πολλές από τις μικρές λεπτομέρειες που αντικατοπτρίζουν το στιλ του ηθοποιού, με αποτέλεσμα η ιδιαίτερη προσωπικότητα του ανθρώπου να είναι αναγνωρίσιμη στην αντίστοιχη κίνηση των ψηφιακών χαρακτήρων ενός εικονικού κόσμου. Επειδή η βασική κίνηση ορίζεται σε πραγματικό χρόνο από το υποκείμενο του οποίου η θέση συλλαμβάνεται, η σύλληψη κίνησης παρέχει μια ισχυρή λύση για εφαρμογές, όπου συνθετική κίνηση με τη χαρακτηριστική ποιότητα της ανθρώπινης κίνησης πρέπει να δημιουργηθεί σε σύντομο χρονικό διάστημα.

Ένα βασικό μειονέκτημα της σύλληψης κίνησης είναι ότι τα δεδομένα που παράγονται αφορούν ένα συγκεκριμένο είδος κίνησης και δεν είναι περαιτέρω επεξεργάσιμα. Αυτό πρακτικά σημαίνει ότι, αν, για παράδειγμα, θέλει κάποιος να κατασκευάσει έναν ψηφιακό χαρακτήρα με ρεαλιστική απόδοση βαδίσματος, δεν αρκεί η καταγραφή μίας μόνο κίνησης, αλλά πολλαπλών, ανάλογα με το πλαίσιο αναφοράς. Θα πρέπει να δημιουργηθεί αργό και γρήγορο βάδισμα, βάδισμα σε ευθεία, σε αηφόρα και σε κατηφόρα, χαρούμενο και λυπημένο βάδισμα κ.λπ. Στην περίπτωση καταγραφής ενεργειών όπου ο ηθοποιός αλληλεπιδρά με αντικείμενα του περιβάλλοντος τα πράγματα είναι ακόμα πιο περίπλοκα. Αν τα δεδομένα της κίνησης εφαρμοστούν σε ψηφιακό χαρακτήρα με διαφορετικές αναλογίες σκελετού (π.χ. μακρύτερο χέρι), τότε το αποτέλεσμα δεν θα είναι το επιθυμητό, για παράδειγμα αντί να πιάνει το χερούλι της πόρτας να πιάνει χαμηλότερα, ψηλότερα ή να περνάει και μέσα από την πόρτα το χέρι του. Και βεβαίως, στην περίπτωση που η κίνηση είναι δυναμική, για παράδειγμα το πιάσιμο ενός κινούμενου αντικειμένου, η σύλληψη κίνησης δεν μπορεί να βοηθήσει. Τα παραπάνω ζητήματα αντιμετωπίζονται σε κάποιο βαθμό με τη συνθετική κίνηση συμπεριφοράς (behavioural animation), την οποία θα παρουσιάσουμε στην ενότητα 5.

4 Φυσική μοντελοποίηση

Η σύλληψη κίνησης μπορεί να διασφαλίζει τη φυσικότητα στις κινήσεις των ανθρωπόμορφων χαρακτήρων, αλλά για τη συνολική συμπεριφορά των φυσικών αντικειμένων ενός δυναμικού περιβάλλοντος οι προκαθορισμένες κινήσεις δεν επαρκούν. Θα πρέπει η κίνηση να δείχνει ρεαλιστική σε όλες τις δυνατές συνθήκες που μπορούν να προκύψουν από τη δυναμική του περιβάλλοντος. Αν, για παράδειγμα, ένας χρήστης εκτοξεύσει ένα αντικείμενο προς μία κατεύθυνση, περιμένουμε η κίνησή του να ακολουθήσει μια πορεία παραπλήσια με αυτήν που έχουμε συνηθίσει στον φυσικό κόσμο. Θα πρέπει να υπάρχει η επίδραση της βαρύτητας στην κίνηση, συγκρούσεις με άλλα αντικείμενα ή/και με το έδαφος και αντίστοιχη αντίδραση των αντικειμένων στις συγκρούσεις. Επειδή, προφανώς, οι δυνατές καταστάσεις στις οποίες μπορεί να βρεθεί ένας εικονικός κόσμος είναι πρακτικά άπειρες, είναι αδύνατο να έχουν προκαθοριστεί όλες οι αντίστοιχες κινήσεις των χαρακτήρων και των αντικειμένων. Είναι επομένως αναγκαίο σε δυναμικά εικονικά περιβάλλοντα ένα μέρος της κίνησης να παράγεται αλγοριθμικά τη στιγμή που απαιτείται και συνυπολογίζοντας την τρέχουσα κατάσταση του περιβάλλοντος.

Η κίνηση αυτής της μορφής ονομάζεται *διαδικαστική συνθετική κίνηση (procedural animation)*, διότι το αποτέλεσμα δεν περιγράφεται από προκαθορισμένες θέσεις αντικειμένων, αλλά παράγεται διαδικαστικά μέσω κάποιου αλγορίθμου (Lin κ.ά., 2009). Για παράδειγμα, αν θέλαμε ένα αντικείμενο να κινηθεί σε ευθεία γραμμή, θα μπορούσαμε σε κάθε χρονική στιγμή να προσθέτουμε ένα σταθερό διάνυσμα στη θέση του. Αν θέλαμε να εκτελέσει μια καμπύλη κίνηση, θα μπορούσαμε αντίστοιχα να προσδιορίσουμε και να επιλύσουμε

την εξίσωση που περιγράφει την κίνηση και να αποδίδουμε τα παραγόμενα δεδομένα κίνησης στο αντικείμενο. Παρόλο που τέτοιου τύπου κινήσεις θα μπορούσαμε να περιγράψουμε και μέσω πλάνων-κλειδιών, το ενδιαφέρον στοιχείο σε αυτή την περίπτωση είναι ότι δεν είναι υποχρεωτικό να είναι εκ των προτέρων γνωστή η διεύθυνση της ευθείας κίνησης ή αντίστοιχα οι παράμετροι που ρυθμίζουν την καμπύλη. Θα μπορούσαν, για παράδειγμα, να εξαρτώνται από τη διεύθυνση που κοιτάζει ο χρήστης.

Η πιο ενδιαφέρουσα εφαρμογή της διαδικαστικής συνθετικής κίνησης είναι η φυσική μοντελοποίηση, δηλαδή η μοντελοποίηση νόμων της φυσικής, αρκετές φορές απλοποιημένων για λόγους καλύτερης απόδοσης, και η χρήση τους για τον καθορισμό της κίνησης των φυσικών αντικειμένων. Η σημασία της φυσικής μοντελοποίησης στις σύγχρονες τρισδιάστατες εφαρμογές αντικατοπτρίζεται στο γεγονός ότι έχουν αναπτυχθεί αρκετές εξειδικευμένες βιβλιοθήκες λογισμικού για φυσική μοντελοποίηση, οι λεγόμενες μηχανές φυσικής, ενώ και οι σύγχρονες κάρτες γραφικών ενσωματώνουν ειδική μονάδα επεξεργασίας για την εκτέλεση αλγορίθμων φυσικής.

4.1 Κίνηση βασισμένη στη φυσική

Η φυσική μοντελοποίηση της κίνησης των αντικειμένων σε έναν εικονικό κόσμο ξεκινάει συνήθως από τους βασικούς κανόνες της κινηματικής (Baraff, 2001). Πλέον κάθε αντικείμενο δεν είναι απλά μια γεωμετρική επιφάνεια με συγκεκριμένη θέση στον χώρο, αλλά αποκτάει υλικό και μάζα. Το πρώτο καθορίζει την τριβή και την ελαστικότητα του, ενώ το δεύτερο την αντίδρασή του σε εξωτερικές δυνάμεις. Τα αντικείμενα μπορούν και κινούνται σύμφωνα με κανόνες της φυσικής που έχουμε μοντελοποιήσει στο σύστημα. Σε κάθε χρονική στιγμή έχουν μία ταχύτητα, η οποία περιγράφεται από ένα τρισδιάστατο διάνυσμα που αποδίδει τη διεύθυνση και το μέτρο της κίνησής τους. Δέχονται δυνάμεις όπως η βαρύτητα και η τριβή, συγκρούονται μεταξύ τους, παραμορφώνονται ή/και αλλάζουν διεύθυνση κίνησης ως συνέπεια των συγκρούσεών τους κ.ά. Τέλος, υπάρχουν και αντικείμενα στον κόσμο τα οποία μπορεί να παράγουν και τα ίδια δυνάμεις, όπως οι ζωντανοί οργανισμοί, τα οχήματα, τα μηχανήματα κ.ά. Τα αντικείμενα αυτά τα ονομάζουμε *ενεργητικά*, σε αντίθεση με τα παθητικά αντικείμενα που μόνο δέχονται δυνάμεις.

Ένας απλοποιημένος γενικός αλγόριθμος κίνησης βασισμένης στη φυσική είναι ο παρακάτω:

1. Όλα τα αντικείμενα έχουν μια αρχική θέση και ταχύτητα.
2. Σε κάθε χρονική στιγμή t υπολογίζονται οι δυνάμεις που ασκούνται σε κάθε αντικείμενο και προκύπτει η συνισταμένη δύναμη F .
3. Από τη συνισταμένη δύναμη και τη μάζα m του κάθε αντικειμένου υπολογίζεται η επιτάχυνση a , σύμφωνα με τον τύπο $F = m a$.
4. Από την επιτάχυνση a και με βάση το χρονικό διάστημα Δt που έχει παρέλθει από τον προηγούμενο υπολογισμό καθορίζεται η ταχύτητα v του κάθε αντικειμένου, σύμφωνα με τον τύπο $v = v_0 + a\Delta t$, όπου v_0 η ταχύτητα που είχε το αντικείμενο την προηγούμενη χρονική στιγμή.
5. Από τον μέσο όρο των δύο ταχυτήτων, προηγούμενης και νέας, υπολογίζεται η θέση του $p = p_0 + (v + v_0)/2 \Delta t$, όπου p_0 η προηγούμενη θέση του.
6. Ζωγραφίζεται η σκηνή.
7. Επιστροφή στο βήμα 2 για τη νέα χρονική στιγμή t' .

Όπως φαίνεται στα παραπάνω βήματα, ο χρόνος υποδιαιρείται σε μικρά διακριτά σημεία και υπολογίζονται οι τιμές των δυνάμεων, επιταχύνσεων και ταχυτήτων σε καθένα από αυτά. Αυτό είναι απαραίτητο εφόσον το σύστημα είναι δυναμικό και μπορεί να εξελιχθεί με μη προβλέψιμο τρόπο, για παράδειγμα ο χρήστης να ασκήσει μια νέα δύναμη μια συγκεκριμένη χρονική στιγμή. Η προσέγγιση αυτή βεβαίως δεν είναι συνεπής με τον φυσικό κόσμο στον οποίο ο χρόνος αντιμετωπίζεται ως συνεχής· η διακριτοποίηση του χρόνου εισάγει μικρά λάθη στους υπολογισμούς. Σήμερα υπάρχουν διαδικασίες φυσικής προσομοίωσης οι οποίες μπορούν να δώσουν αποτελέσματα με πολύ μεγάλη ακρίβεια και λεπτομέρεια, αλλά ο χρόνος εκτέλεσής τους είναι απαγορευτικός για συστήματα πραγματικού χρόνου. Αναγκαστικά περιοριζόμαστε σε αρκετά πιο απλοποιημένες προσεγγίσεις στην εφαρμογή των νόμων της φυσικής, με στόχο όχι τόσο την ακρίβεια στα αποτελέσματα όσο την αληθοφάνεια.

Ας δούμε ένα απλό παράδειγμα ενός αντικειμένου που εκτελεί ελεύθερη πτώση. Έστω ότι αρχικά βρίσκεται σε απόσταση ενός μέτρου πάνω από το έδαφος. Έστω επίσης ότι ο ρυθμός ανανέωσης της οθόνης είναι σταθερός στα 60 πλάνα το δευτερόλεπτο, άρα το χρονικό διάστημα μεταξύ δύο ανανεώσεων είναι $\Delta t =$

0.017. Στο αντικείμενο ασκείται μόνο η επιτάχυνση της βαρύτητας $g = 9.81$. Επομένως σε κάθε ανανέωση υπολογίζεται η νέα του ταχύτητα και η νέα του θέση. Οι τιμές ταχύτητας και θέσης εμφανίζονται στον παρακάτω πίνακα. Οι τιμές αφορούν μόνο τον άξονα των Y , καθώς στους X και Z οι τιμές θα είναι μηδενικές. Για λόγους απλοποίησης θεωρούμε ότι το ύψος του εδάφους είναι στο μηδέν και ότι η τιμή του y περιγράφει την απόσταση του κατώτερου σημείου του από το έδαφος. Οι τιμές του ύψους και της ταχύτητας στις διαδοχικές χρονικές στιγμές εμφανίζονται στον πίνακα 4.1.

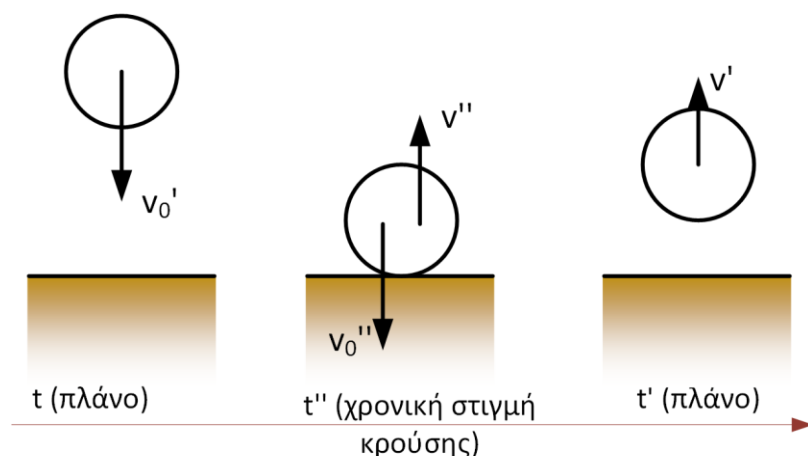
t	y	v
0	1,000	0,000
1	0,999	0,167
2	0,994	0,334
3	0,987	0,500
4	0,977	0,667
...		
26	0,042	4,336
27	-0,033	4,503

Πίνακας 4.1 Απόσταση από το έδαφος (y) και ταχύτητα (v) ενός αντικειμένου που πέφτει με την επιτάχυνση της βαρύτητας. Το διάστημα μεταξύ των χρονικών στιγμών Δt είναι 0.017.

Όπως φαίνεται και στις παραπάνω τιμές, υπάρχει μια χρονική στιγμή ($t = 26$) κατά την οποία το αντικείμενο είναι πάνω από το έδαφος και μια άλλη ($t' = 27$) στην οποία είναι κάτω από αυτό. Στο σημείο αυτό ο μηχανισμός προσομοίωσης θα πρέπει να είναι σε θέση να αντιληφθεί ότι υπάρχει σύγκρουση με το έδαφος, να υπολογίσει τη νέα ταχύτητα του αντικειμένου, ανάλογα με την ελαστικότητα της κρούσης, και να το μετακινήσει στη σωστή θέση. Η σωστή προσέγγιση, η οποία όμως έχει μεγαλύτερο υπολογιστικό κόστος, είναι:

1. να βρεθεί η ακριβής χρονική στιγμή t'' μεταξύ των t και t' κατά την οποία έγινε η σύγκρουση με επίλυση των παραπάνω εξισώσεων κίνησης για $y = 0$,
2. να υπολογιστεί η ταχύτητα v_0'' τη χρονική στιγμή t'' και η νέα ταχύτητα v'' μετά την κρούση και
3. με βάση τα παραπάνω να υπολογιστεί η νέα ταχύτητα v' και η νέα θέση του αντικειμένου y' τη χρονική στιγμή $t' = 27$ που τελικά θα απεικονιστεί.

Σύμφωνα με την προσέγγιση αυτήν σε κανένα πλάνο δεν θα εμφανιστεί το αντικείμενο να εφάπτεται με το έδαφος, όμως η κίνηση θα είναι περισσότερο συνεπής ως προς τους νόμους της κινηματικής και ως τέτοια θα δείχνει και πιο αληθοφανής. Η παραπάνω διαδικασία απεικονίζεται στην εικόνα 3.7.



Εικόνα 3.7 Κρούση σφαίρας με το έδαφος μεταξύ δύο πλάνων.

Το παράδειγμα που μόλις παρουσιάσαμε είναι μια απλοποιημένη περίπτωση φυσικής κίνησης. Συνήθως στους εικονικούς κόσμους προκύπτουν συγκρούσεις μεταξύ πολλαπλών αντικειμένων, καθένα από τα οποία έχει τη

δική του ιδιαίτερη γεωμετρία και θα πρέπει να υπολογίζεται τόσο η περίπτωση σύγκρουσης μεταξύ δύο ή περισσότερων αντικειμένων όσο και η αντίδρασή τους μετά τη σύγκρουση, δηλαδή οι νέες ταχύτητες και γωνιακές ταχύτητες που θα αποκτήσουν.

4.2 Αναγνώριση σύγκρουσης

Η αναγνώριση σύγκρουσης μεταξύ αντικειμένων είναι μια διαδικασία η οποία εκτελείται διαρκώς στους εικονικούς κόσμους. Οι ενσαρκώσεις πρέπει να κινούνται πάνω στο έδαφος και η κίνησή τους να εμποδίζεται από άλλα φυσικά αντικείμενα του περιβάλλοντος, όπως τοίχοι, έπιπλα κ.ά. Σε πολλές περιπτώσεις οι ενσαρκώσεις πρέπει να μπορούν να επενεργούν πάνω στα αντικείμενα, να πιέζουν, να τραβούν, να κλοτσούν κ.ά. Αν ένα όπλο βάλει προς μια διεύθυνση, θα πρέπει το περιβάλλον να μπορεί να εντοπίσει τη χρονική στιγμή που η σφαίρα θα προσκρούσει σε ένα αντικείμενο για να αντιδράσει ανάλογα. Και, φυσικά, στην κίνηση των φυσικών αντικειμένων θα πρέπει να συνυπολογίζονται οι μεταξύ τους συγκρούσεις. Όλα τα παραπάνω παραδείγματα προϋποθέτουν την ικανότητα του περιβάλλοντος να αναγνωρίζει συγκρούσεις, δηλαδή να μπορεί να αποφασίσει τότε οι γεωμετρικές δύο αντικειμένων τέμνονται μεταξύ τους.

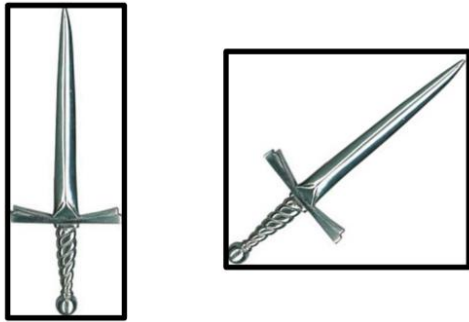
Οι απαιτήσεις ως προς την αναγνώριση σύγκρουσης διαφέρουν από εφαρμογή σε εφαρμογή και για τον λόγο αυτόν έχουν αναπτυχθεί διάφορα σχετικά μοντέλα και προσεγγίσεις (Moore & Wilhelms, 1988). Σε κάποιες περιπτώσεις η λεπτομέρεια δεν παίζει μεγάλο ρόλο. Για παράδειγμα, στην κίνηση των ενσαρκώσεων δεν απαιτείται συνήθως πολύ μεγάλη ακρίβεια στον εντοπισμό της σύγκρουσής τους με αντικείμενα του περιβάλλοντος. Ακόμη και αν σταματήσει η κίνηση της ενσάρκωσης μερικά εκατοστά μακριά από τον τοίχο χωρίς πραγματικά να εφάπτεται το σώμα της σε αυτόν, η διαδικασία της πλοήγησης δεν θα επηρεαστεί σημαντικά. Από την άλλη μεριά, σε μια εφαρμογή εξάσκησης στη χειρουργική είναι σημαντικό να γνωρίζει το σύστημα με μεγάλη ακρίβεια τη θέση στην οποία ακουμπάει το χειρουργικό εργαλείο το σώμα του ασθενούς, ώστε να προκαλέσει την αναμενόμενη αντίδραση, σε επίπεδο προσομοίωσης αλλά και οπτικοποίησης. Επειδή η διαδικασία αναγνώρισης σύγκρουσης είναι επίπονη υπολογιστικά, χρησιμοποιούνται συνήθως απλοποιημένες γεωμετρικές αντί για τις πραγματικές επιφάνειες των αντικειμένων, ειδικά σε περιπτώσεις στις οποίες η ακρίβεια δεν παίζει τόσο μεγάλο ρόλο.

Ένας πολύ γρήγορος τρόπος αναγνώρισης σύγκρουσης μεταξύ δύο στερεών είναι με τη χρήση των περιβαλλόντων ορθογωνίων παραλληλεπίπεδων (*bounding boxes*). Το bounding box είναι το ελάχιστο ορθογώνιο παραλληλεπίπεδο που περιβάλλει πλήρως ένα τρισδιάστατο αντικείμενο. Αν διατρέξουμε όλες τις κορυφές ενός αντικειμένου μπορούμε να βρούμε τις ελάχιστες και τις μέγιστες τιμές στους τρεις άξονες: x_{min} , y_{min} , z_{min} και x_{max} , y_{max} και z_{max} αντίστοιχα. Με τις τιμές αυτές μπορούμε να ορίσουμε τις οκτώ κορυφές ενός ορθογωνίου παραλληλεπίπεδου το οποίο αποτελεί και το bounding box του αντικειμένου. Έστω ότι το bounding box ενός αντικειμένου προσδιορίζεται από τις τιμές x_{min} , y_{min} , z_{min} και x_{max} , y_{max} , z_{max} και ενός δεύτερου αντικειμένου από τις αντίστοιχες τιμές x'_{min} , y'_{min} , z'_{min} , x'_{max} , y'_{max} και z'_{max} . Τότε τα δύο αυτά bounding boxes τέμνονται αν:

$$\begin{aligned}x_{min} < x'_{max} \text{ , } x'_{min} < x_{max} \text{ ,} \\y_{min} < y'_{max} \text{ , } y'_{min} < y_{max} \text{ ,} \\z_{min} < z'_{max} \text{ , } z'_{min} < z_{max}\end{aligned}$$

Έτσι, με έξι απλούς ελέγχους τιμών μπορούμε να αποφασίσουμε αν τα δύο bounding boxes τέμνονται ή όχι. Η διαδικασία είναι πολύ απλή, διότι και τα δύο bounding boxes που συγκρίναμε είχαν τον ίδιο προσανατολισμό, και μάλιστα ήταν προσανατολισμένα στους άξονες του συστήματος συντεταγμένων κόσμου. Για τον λόγο αυτόν, τα παραπάνω bounding boxes ονομάζονται περιβάλλοντα ορθογώνια παραλληλεπίπεδα ευθυγραμμισμένα στους άξονες (*axis aligned bounding boxes*). Ένα πρόβλημα με την προσέγγιση είναι ότι, για να βρούμε τις μέγιστες και ελάχιστες τιμές των συντεταγμένων στους τρεις άξονες, χρειάστηκε να διατρέξουμε όλες τις κορυφές. Κάτι τέτοιο μπορεί να έχει μεγάλο υπολογιστικό κόστος στην περίπτωση κινούμενων αντικειμένων και ιδιαίτερα αν έχουν μεγάλο αριθμό τριγώνων, διότι το bounding box θα επαναπροσδιορίζεται σε κάθε πλάνο. Επιπλέον, η ίδια η κίνηση των αντικειμένων μπορεί να δημιουργήσει bounding boxes τα οποία δεν περικλείουν πολύ καλά το αντικείμενο και αφήνουν αρκετό ελεύθερο χώρο, κάτι που θα εισήγαγε αρκετά λάθη αν χρησιμοποιούνταν για αναγνώριση σύγκρουσης (βλ. εικ. 3.8). Για τους λόγους αυτούς, τα axis aligned

bounding boxes είναι προτιμότερα στην περίπτωση στατικών αντικειμένων και μεγάλων περιοχών του περιβάλλοντος και λιγότερο για κινούμενα αντικείμενα.



Εικόνα 3.8 Καλή (αριστερά) και κακή (δεξιά) εφαρμογή bounding boxes ευθυγραμμισμένων στους άξονες.

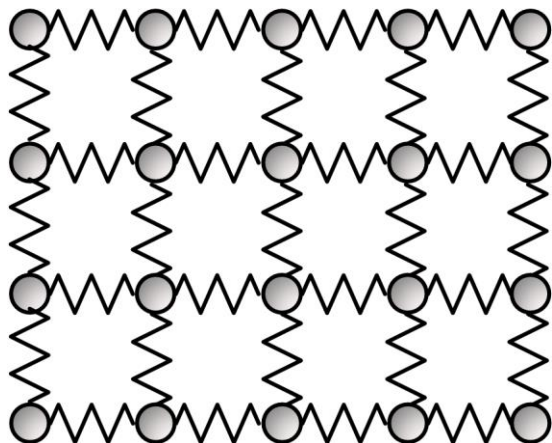
Τα προβλήματα που αναφέραμε παραπάνω μπορούν να αντιμετωπιστούν με τα *προσανατολισμένα παραλληλεπίπεδα* (*oriented bounding boxes*), για τα οποία όμως η ανίχνευση συγκρούσεων είναι δυσκολότερη. Τα *oriented bounding boxes* υπολογίζονται μία μοναδική φορά κατά τη δημιουργία του αντικειμένου ως προς το τοπικό σύστημα συντεταγμένων του. Αυτό σημαίνει ότι, αν είναι κατάλληλα προσανατολισμένο το αντικείμενο, μπορούν να περικλείουν αρκετά καλά τη γεωμετρία του. Όσο κινείται το αντικείμενο, το bounding box ακολουθεί την κίνησή του, δηλαδή εφαρμόζεται και σε αυτό ο ίδιος μετασχηματισμός. Έτσι, αφενός δεν απαιτείται επανυπολογισμός του και αφετέρου συνεχίζει να καλύπτει το ίδιο καλά τη γεωμετρία του αντικειμένου ανεξαρτήτως του προσανατολισμού του. Στην περίπτωση των σύνθετων, αρθρωτών αντικειμένων που περιλαμβάνουν ιεραρχία μετασχηματισμών, αποδίδεται ένα bounding box σε κάθε επιμέρους τμήμα, με αποτέλεσμα η συνολική κάλυψη του αντικειμένου να είναι ακόμα πιο λεπτομερής. Το μειονέκτημα με τα *oriented bounding boxes* είναι ότι είναι πιο επίπονη η αναγνώριση συγκρούσεων, διότι πλέον τα δύο bounding boxes που συγκρίνονται δεν έχουν τον ίδιο προσανατολισμό. Παρόλα αυτά υπάρχουν ικανοποιητικοί αλγόριθμοι που επιλύουν το πρόβλημα, όπως ο αλγόριθμος βασισμένος στο *Θεώρημα του Άξονα Διαχωρισμού* (*Separating Axis Theorem*) (Gottschalk κ.ά., 1996), και για τον λόγο αυτόν τα *oriented bounding boxes* χρησιμοποιούνται αρκετά συχνά σε συστήματα πραγματικού χρόνου και ιδιαίτερα στους εικονικούς κόσμους.

Σε περιπτώσεις που απαιτείται μεγαλύτερη ταχύτητα στους υπολογισμούς ή διαφορετικής μορφής κάλυψη της γεωμετρίας, μπορούν να χρησιμοποιηθούν και άλλες βασικές γεωμετρικές εναλλακτικές του ορθογωνίου παραλληλεπίπεδου ή/και συνδυασμός αυτών. Παραδείγματα τέτοιων γεωμετριών είναι ο κύλινδρος και η σφαίρα. Ιδιαίτερα στην περίπτωση της *περιβάλλουσας σφαίρας* (*bounding sphere*), η αναγνώριση γίνεται σε ένα μόνο βήμα. Αρκεί να ελεγχθεί αν η απόσταση μεταξύ των κέντρων των δύο σφαιρών είναι μικρότερη του αθροίσματος των δύο ακτίνων· στην περίπτωση αυτήν έχουμε σύγκρουση. Παρόλο που η σφαίρα δεν έχει μεγάλη ακρίβεια στην κάλυψη, θα μπορούσαν να χρησιμοποιηθούν ιεραρχίες σφαιρών ή γενικότερα περιβαλλόντων όγκων για να καλύψουν καλύτερα το αντικείμενο. Για παράδειγμα, έστω ότι ένας μεγάλος όγκος περιβάλλει όλο το αντικείμενο. Αν ένας άλλος αντίστοιχος όγκος ενός άλλου αντικειμένου δεν συγκρούεται με αυτόν, τότε σταματάει ο έλεγχος. Αν συγκρούονται, τότε οι έλεγχοι κατεβαίνουν σε όγκους-παιδιά της ιεραρχίας, οι οποίοι καλύπτουν υπο-περιοχές των αντικειμένων. Αντίστοιχα, αν συγκρούεται κάποιο ζευγάρι όγκων, ο έλεγχος κατεβαίνει σε ακόμα μικρότερους όγκους κ.λπ. Η προσέγγιση αυτή μπορεί να δώσει έναν πολύ καλό συμβιβασμό μεταξύ ακρίβειας και λεπτομέρειας.

Τέλος, μπορεί να υπάρχουν και εξειδικευμένες περιπτώσεις για τις οποίες χρειάζεται να ελέγχεται με μεγάλη ακρίβεια η σύγκρουση ή ακόμα και να εντοπίζονται τα ακριβή σημεία επαφής μεταξύ των αντικειμένων. Τέτοιες διαδικασίες έχουν μεγάλο υπολογιστικό κόστος, αλλά έχουν ως αποτέλεσμα περισσότερο φυσική συμπεριφορά των αντικειμένων. Πέρα από τη σωστότερη κάλυψη του αντικειμένου, αν εντοπιστεί με ακρίβεια το σημείο της σύγκρουσης, τότε και η αντίδραση του αντικειμένου στην κρούση θα είναι πιο σωστή και θα δείχνει περισσότερο φυσική. Ακόμα και σε αυτές τις περιπτώσεις, πάντως, χρησιμοποιούνται και περιβάλλοντες όγκοι, οι οποίοι αποκλείουν ομάδες αντικειμένων, αντικείμενα ή και ομάδες τριγώνων ενός αντικειμένου από την πιθανότητα σύγκρουσης, με αποτέλεσμα οι έλεγχοι πάνω στα ίδια τα τρίγωνα να είναι αρκετά περιορισμένοι, για να μην επιβαρύνεται ιδιαίτερα το σύστημα.

4.3 Άλλες φυσικές ιδιότητες

Εκτός από την απλή κίνηση και κρούση φυσικών άκαμπτων αντικειμένων, οι σύγχρονες μηχανές φυσικής μοντελοποίησης παρέχουν τη δυνατότητα εισαγωγής και άλλων φυσικών δομών για την κατασκευή περισσότερο περίπλοκων συστημάτων. Για παράδειγμα, μπορεί κάποιος να ορίσει ελατήρια, αρμούς, αρθρώσεις χαρακτήρων κ.ά., ώστε να επιτύχει την επιθυμητή φυσική συμπεριφορά ενός ή περισσότερων αντικειμένων. Μια δημοφιλής δομή είναι και η *κούκλα (ragdoll)*, όπου το σώμα ενός ψηφιακού χαρακτήρα αντιδρά σε εξωτερικές δυνάμεις σαν να ήταν άψυχη κούκλα. Η τεχνική αυτή χρησιμοποιείται συχνά στα παιχνίδια για να αποδώσει με ρεαλιστικό τρόπο το χτύπημα ενός χαρακτήρα από κάποια σφαίρα ή την πτώση του από ψηλά.



Εικόνα 3.9 Δομή μαζών–ελατηρίων για τη μοντελοποίηση υφασμάτων κ.ά.

Μια άλλη δομή που χρησιμοποιείται συχνά στην περίπτωση ελαστικών ή γενικότερα παραμορφώσιμων αντικειμένων, όπως για παράδειγμα τα υφάσματα, είναι η δομή μαζών ελατηρίων (Provot, 1995). Σύμφωνα με αυτήν, η γεωμετρία του αντικειμένου περιγράφεται από ένα πυκνό πλέγμα, σε κάθε κορυφή του οποίου αποδίδεται μια μικρή σημειακή μάζα. Στις ακμές αποδίδονται ελατήρια, ενώ ενδέχεται να προστεθούν και παραπάνω συνδέσεις (π.χ. διαγώνιες) για μεγαλύτερη συνοχή (εικ. 3.9). Το αποτέλεσμα είναι ότι, αν ασκηθούν δυνάμεις σε μία ή περισσότερες μάζες, τα ελατήρια που συνδέονται με αυτές θα τεντωθούν, συμπαρασύροντας σε κάποιο βαθμό και τις γειτονικές μάζες, οι οποίες με τη σειρά τους θα παραμορφώσουν κάποια άλλα ελατήρια κ.ο.κ. Με αυτήν την τεχνική, η γεωμετρία θα παραμορφωθεί ομαλά σε σχέση με τη δύναμη που της ασκήθηκε και το αποτέλεσμα θα δείχνει αρκετά φυσικό, όπως για παράδειγμα στην περίπτωση ρούχων που ακολουθούν την κίνηση του σώματος του χαρακτήρα που τα φοράει.

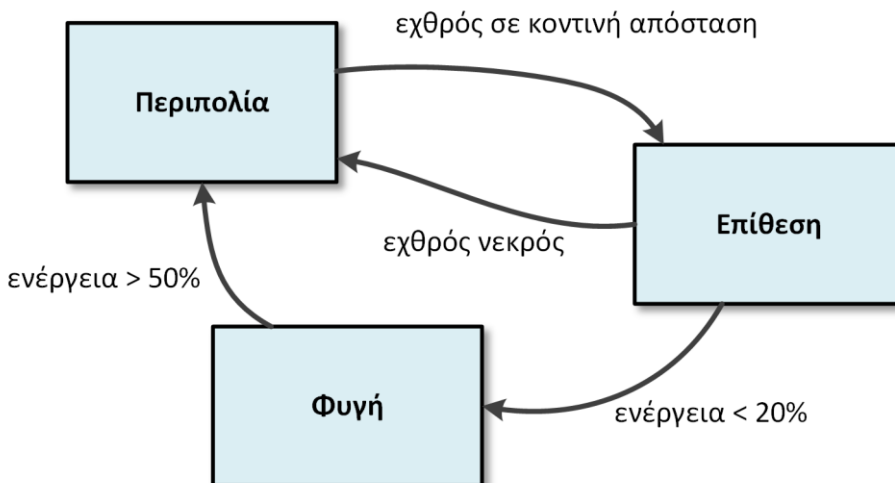
5 Δυναμική συμπεριφορά

Στους εικονικούς κόσμους πολλές φορές συμπεριλαμβάνονται και οντότητες με αυτόνομη, δυναμική συμπεριφορά, όπως συνθετικοί χαρακτήρες ή συσκευές με σύνθετη λειτουργία. Οι οντότητες αυτές δεν εκτελούν προκαθορισμένη κίνηση, αφού οι ενέργειές τους εξαρτώνται από την κατάσταση του περιβάλλοντος. Επιπλέον, δεν συμπεριφέρονται ως παθητικά φυσικά αντικείμενα, καθώς έχουν τη δυνατότητα να επενεργήσουν και οι ίδιες στο περιβάλλον. Για παράδειγμα, ένας εικονικός άνθρωπος μπορεί να αποφασίσει να κινηθεί προς κάποια κατεύθυνση, να χειριστεί ένα αντικείμενο, να επικοινωνήσει με κάποιον άλλο χαρακτήρα, εικονικό ή ενσάρκωση κ.ά. Αντίστοιχα, ένα «έξυπνο» όπλο μπορεί να αποφασίσει να στραφεί προς τη μεριά των χαρακτήρων που αναγνωρίζει ως εχθρικούς και αν πλησιάσουν αρκετά κοντά να σημαδέψει και να βάλει εναντίον τους. Σε αυτές τις περιπτώσεις, η κίνηση παράγεται βάσει κάποιου απλού ή πιο σύνθετου μοντέλου συμπεριφοράς. Τα μοντέλα αυτά εξετάζουν την τρέχουσα κατάσταση του περιβάλλοντος και αποφασίζουν για τις ενέργειες της οντότητας σύμφωνα με κάποιους κανόνες που έχει περιγράψει ο σχεδιαστής. Η πιο απλή περίπτωση ορισμού δυναμικής συμπεριφοράς αντικειμένων είναι η χρήση μηχανών πεπερασμένων καταστάσεων (Finite State Machines – FSMs), μια προσέγγιση που χρησιμοποιείται ευρέως στον χώρο των παιχνιδιών.

5.1 Μηχανές καταστάσεων

Οι *μηχανές πεπερασμένων καταστάσεων* ή απλά *μηχανές καταστάσεων* είναι ένα μαθηματικό μοντέλο υπολογισμού το οποίο χρησιμοποιείται με επιτυχία στη σχεδίαση κυκλωμάτων και στην πληροφορική για τον προγραμματισμό της λειτουργίας διάφορων συστημάτων. Το μοντέλο είναι εξαιρετικά απλό: μια αφηρημένη μηχανή μπορεί να βρεθεί στη διάρκεια ζωής της σε έναν αριθμό από καταστάσεις λειτουργίας. Σε μια συγκεκριμένη χρονική στιγμή όμως μπορεί να βρίσκεται σε μία μόνο κατάσταση. Από την κατάσταση που βρίσκεται, σύμφωνα με κάποιες προκαθορισμένες συνθήκες, μπορεί να μεταβεί σε ορισμένες άλλες καταστάσεις. Η διαδικασία αλλαγής κατάστασης ονομάζεται *μετάβαση* και οι συνθήκες που μπορούν να προκαλέσουν την αλλαγή αυτή συνθήκες *μετάβασης*. Τέλος, μία από τις καταστάσεις ορίζεται ως η αρχική κατάσταση του συστήματος, στην οποία βρίσκεται η μηχανή όταν ξεκινάει η λειτουργία της.

Μπορούμε να αποδώσουμε διαγραμματικά μια μηχανή καταστάσεων ως έναν κατευθυνόμενο γράφο, στον οποίο οι κόμβοι είναι οι καταστάσεις και οι ακμές οι μεταβάσεις μεταξύ τους. Στις καταστάσεις προσθέτουμε το όνομά τους και στις μεταβάσεις τις συνθήκες, ενώ συνήθως χρησιμοποιούμε κάποια ένδειξη για να καταδείξουμε την αρχική κατάσταση, όπως ο διπλός κύκλος. Στην εικόνα 3.10 φαίνεται ένα παράδειγμα μηχανής καταστάσεων για τη συμπεριφορά ενός χαρακτήρα σε ένα παιχνίδι.



Εικόνα 3.10 Παράδειγμα μηχανής καταστάσεων για τη συμπεριφορά χαρακτήρα σε παιχνίδι.

Όπως φαίνεται στο παράδειγμά μας, ο χαρακτήρας μπορεί να βρεθεί σε τρεις καταστάσεις: α) να περιπολεί μια συγκεκριμένη περιοχή (κατάσταση «Περιπολία»), β) να επιτίθεται στον εχθρό (κατάσταση «Επίθεση»), γ) να τραπεί σε φυγή (κατάσταση «Φυγή»). Στην εικόνα περιγράφονται και οι αντίστοιχες μεταβάσεις, δηλαδή μεταβαίνει από περιπολία σε επίθεση, αν εντοπίσει εχθρό σε κοντινή απόσταση, από επίθεση σε περιπολία, αν ο εχθρός πεθάνει ή απομακρυνθεί, από επίθεση σε φυγή, αν η ενέργειά του πέσει πολύ χαμηλά, και από φυγή σε περιπολία, αν η ενέργειά του επιστρέψει σε ικανοποιητικά επίπεδα.

Το σημαντικό πλεονέκτημα των μηχανών καταστάσεων είναι ότι απλοποιούν την περιγραφή μιας περίπλοκης συμπεριφοράς μέσω της τμηματοποίησής της σε διακριτές καταστάσεις και μεταβάσεις μεταξύ τους. Αν προσπαθούσαμε, για παράδειγμα, να περιγράψουμε αλγοριθμικά τη συνολική συμπεριφορά του χαρακτήρα του παραδείγματός μας, ο αλγόριθμος θα ήταν μάλλον αρκετά σύνθετος και δύσκολος στην αποσφαλμάτωση. Αντίθετα, με τη χρήση των μηχανών καταστάσεων είναι πιο εύκολος ο προγραμματισμός των ενεργειών, επειδή ακριβώς δεν απαιτείται να οριστεί συνολικά η συμπεριφορά του χαρακτήρα αλλά αυτή που θα έχει σε κάθε διακριτή κατάσταση. Για παράδειγμα, στην κατάσταση περιπολίας θα ακολουθεί ένα προκαθορισμένο μονοπάτι ελέγχοντας μήπως πλησιάζει ο εχθρός, στην κατάσταση επίθεσης θα κινείται προς το μέρος του εχθρού και θα επιτίθεται, και στην κατάσταση φυγής θα προσπαθεί να απομακρυνθεί όσο το δυνατόν πιο μακριά από τον εχθρό. Διαιρέσαμε λοιπόν, χάρη στις μηχανές καταστάσεων, την περιγραφή της συνολικής συμπεριφοράς του χαρακτήρα σε δύο μέρη: ένα υψηλού επιπέδου, στο οποίο περιγράφουμε τις διάφορες καταστάσεις που μπορεί να έχει και πότε μεταβαίνει από τη μία στην άλλη, και ένα χαμηλότερο, στο οποίο εξειδικεύουμε τη συμπεριφορά του σε κάθε συγκεκριμένη κατάσταση.

Επιπλέον, λόγω της απλότητάς τους, οι μηχανές καταστάσεων είναι πολύ εύκολες στον προγραμματισμό. Η γενική λογική είναι η παρακάτω:

1. Ονοματίζουμε τις καταστάσεις μας και αποδίδουμε από έναν μοναδικό ακέραιο αριθμό σε καθεμία από αυτές (ή χρησιμοποιούμε enumerations, αν μας το επιτρέπει η γλώσσα προγραμματισμού).
2. Δηλώνουμε μια μεταβλητή στην οποία θα καταγράφεται η τρέχουσα κατάσταση.
3. Ορίζουμε την αρχική κατάσταση στην έναρξη του προγράμματος.
4. Σε κάθε ανανέωση ελέγχουμε αν η τιμή της τρέχουσας κατάστασης ταυτίζεται με καθεμία από τις καταστάσεις που έχουμε ορίσει (με δομή if / else if ή με switch/case) και σε κάθε έλεγχο προσθέτουμε:
 - ο τη συμπεριφορά του αντικειμένου στην κατάσταση αυτήν,
 - τους ελέγχους των συνθηκών μετάβασης. Στην περίπτωση μετάβασης απλά αλλάζουμε την τιμή της τρέχουσας κατάστασης.

Ένα σημαντικό μειονέκτημα της χρήσης μηχανών καταστάσεων είναι ότι οδηγούν σε προβλέψιμες συμπεριφορές. Επειδή ακριβώς τόσο οι καταστάσεις όσο και οι συνθήκες μετάβασης είναι προκαθορισμένες, μπορεί ένας εξωτερικός παρατηρητής να ανακαλύψει επαναλαμβανόμενα πρότυπα (patterns) στη συμπεριφορά χαρακτήρων φτιαγμένων με μηχανές καταστάσεων και, τελικά, να «αποκωδικοποιήσει» τη συμπεριφορά τους. Κάτι τέτοιο κάνει τους αντιπάλους στα παιχνίδια αρκετά προβλέψιμους, άρα και πιο εύκολους, αλλά και γενικότερα τη συμπεριφορά των χαρακτήρων να δείχνει λιγότερο ρεαλιστική. Το πρόβλημα μπορεί να περιοριστεί αν οριστεί μεγάλος αριθμός καταστάσεων, με αποτέλεσμα η συμπεριφορά να είναι περισσότερο περίπλοκη και δυσκολότερα προβλέψιμη, αλλά σε κάθε περίπτωση οι μηχανές καταστάσεων δυσκολεύονται να περιγράψουν πιο σύνθετες και «έξυπνες» συμπεριφορές. Προς την κατεύθυνση αυτή μεγαλύτερη επιτυχία έχουν προσεγγίσεις από τον χώρο της τεχνητής νοημοσύνης, όπως οι «ευφυείς πράκτορες» και ο σχεδιασμός ενεργειών (planning), που θα δούμε στη συνέχεια.

5.2 Ευφυείς πράκτορες

Ο χώρος της τεχνητής νοημοσύνης μελετά, μεταξύ άλλων, τη συμπεριφορά συστημάτων ελέγχου σε διαφόρων ειδών περιβάλλοντα, φυσικά ή τεχνητά, στα οποία τα εισερχόμενα ερεθίσματα αντιστοιχίζονται σε εξερχόμενες αντιδράσεις. Οι παραγόμενες συμπεριφορές των συστημάτων αυτών στηρίζονται σε αρχιτεκτονικές ευφυών πρακτόρων (Wooldridge & Jennings, 1995) προσαρμοσμένες στο περιβάλλον λειτουργίας με τους κατάλληλους αισθητήρες και επιδραστές, δηλαδή με μηχανισμούς ανάγνωσης της κατάστασης του περιβάλλοντος και επενέργειας στο περιβάλλον αντίστοιχα.

Ως πράκτορας στον χώρο της τεχνητής νοημοσύνης ορίζεται ένα οποιοδήποτε σύστημα έχει την ικανότητα να αντιλαμβάνεται το περιβάλλον του και να επιδρά σε αυτό. Προφανώς ο ορισμός αυτός είναι πολύ γενικός και μπορεί να περιλαμβάνει από πολύ απλές συσκευές που απλά αντιδρούν σε ενέργειες του χρήστη (π.χ. ο λαμπτήρας) μέχρι εξελιγμένα ρομπότ. Η έμφαση βεβαίως στον χώρο αυτόν δίνεται στους λεγόμενους ευφυείς πράκτορες. Είναι αρκετά δύσκολο να ορίσουμε τι ακριβώς είναι αυτό που κάνει έναν πράκτορα «ευφυή». Ωστόσο, μπορούμε γενικά να πούμε ότι ένας πράκτορας είναι ευφυής όταν έχει την ικανότητα να επιτελεί τους στόχους και τα καθήκοντα που έχει επιφορτιστεί. Έτσι, σε ένα ελάχιστο επίπεδο νοημοσύνης μπορεί να δίνονται στον πράκτορα εντολές με τη μορφή κανόνων και αυτός να ενεργεί με τη βοήθεια κάποιου μηχανισμού εξαγωγής συμπερασμάτων. Σε ένα ανώτερο επίπεδο ο πράκτορας θα είναι ικανός να μαθαίνει και να προσαρμόζεται αυτόματα στο περιβάλλον έτσι ώστε να πετυχαίνει τους σκοπούς του.

Σήμερα υπάρχουν πολλές αρχιτεκτονικές ευφυών πρακτόρων, οι οποίες σχετίζονται σε μεγάλο βαθμό και με τη φύση του περιβάλλοντος στο οποίο οι πράκτορες επενεργούν. Ορισμένες βασικές προσεγγίσεις είναι:

- *Απλοί αντανakλαστικοί πράκτορες*: η πιο απλή προσέγγιση. Οι πράκτορες έχουν μια βάση κανόνων οι οποίοι συσχετίζουν εισερχόμενα ερεθίσματα με ενέργειες. Ανάλογα με την τρέχουσα κατάσταση, όπως την έχουν αντιληφθεί από τους αισθητήρες, επιλέγουν την ενέργεια που θα εκτελέσουν.
- *Αντανakλαστικοί πράκτορες βασισμένοι σε μοντέλο*: μια επέκταση της προηγούμενης αρχιτεκτονικής, όπου ο πράκτορας διατηρεί ένα μοντέλο για το πώς είναι ο κόσμος και πώς

εξελίσσεται, με αποτέλεσμα να μπορεί να βγάλει συμπεράσματα και για περιοχές του κόσμου που δεν είναι άμεσα ορατές.

- *Πράκτορες βασισμένοι στον στόχο*: η συμπεριφορά του πράκτορα δεν βασίζεται πλέον σε κανόνες αλλά σε προγραμματισμό ενεργειών (planning). Ο πράκτορας έχει κάποιους συγκεκριμένους στόχους, τους οποίους έχει θέσει ο σχεδιαστής. Επιπλέον, γνωρίζει τι κάνουν οι ενέργειές του στο περιβάλλον, δηλαδή ποιες είναι οι προϋποθέσεις για να εκτελεστούν και ποια τα αποτελέσματά τους. Με βάση τα παραπάνω, καταλήγει σε ένα πλάνο, δηλαδή μια ακολουθία ενεργειών η οποία αναμένεται να τον οδηγήσει στον στόχο.
- *Πράκτορες βασισμένοι στη χρησιμότητα*: ο πράκτορας δεν έχει συγκεκριμένους στόχους, αλλά έναν μηχανισμό που αξιολογεί την κατάσταση του περιβάλλοντος ως προς το πόσο «χρήσιμη» είναι για τον πράκτορα, τη συνάρτηση χρησιμότητας (utility function). Αποφασίζει για μια σειρά ενεργειών που αναμένεται να βελτιώσει την κατάσταση του περιβάλλοντος.
- *Πράκτορες βασισμένοι στη μάθηση*: ο πράκτορας έχει επιπλέον την ικανότητα να μαθαίνει, βελτιώνοντας έτσι την κατανόησή του σε σχέση με τον τρόπο που λειτουργεί το περιβάλλον, τα αποτελέσματα των ενεργειών του ή και τη συμπεριφορά άλλων οντοτήτων. Οι αρχιτεκτονικές αυτές βασίζονται σε τεχνικές από τον χώρο της μηχανικής μάθησης (machine learning).

Στους εικονικούς κόσμους μπορούν να εφαρμοστούν οι παραπάνω αρχιτεκτονικές για τη συμπεριφορά ευφώνων οντοτήτων, όπως είναι οι εικονικοί άνθρωποι. Για τον σκοπό αυτόν έχει αναπτυχθεί τις τελευταίες δεκαετίες ένας σχετικός χώρος έρευνας, οι Ευφυείς Εικονικοί Πράκτορες (Intelligent Virtual Agents). Τα βασικά θέματα που μελετώνται στην περιοχή αυτήν είναι η αληθοφάνεια στην αντίληψη και στη δράση, δηλαδή η χρήση αισθητήρων και επιδραστών που προσομοιώνουν τους αντίστοιχους ανθρώπινους μηχανισμούς, η ευφυής αλληλεπίδραση με το περιβάλλον, η επικοινωνία με χρήστες μέσω φυσικής γλώσσας και η μοντελοποίηση και αποτύπωση προσωπικότητας και συναισθημάτων στη συμπεριφορά (Luck & Aylett, 2000). Ανάλογα με τους στόχους ενός εικονικού κόσμου, ο βαθμός αληθοφάνειας και ευφύιας των εικονικών χαρακτήρων μπορεί να ποικίλλει. Στις περισσότερες περιπτώσεις οι αντανακλαστικοί πράκτορες με μοντέλο σε συνδυασμό με απλοποιημένους μηχανισμούς όρασης και ακοής επαρκούν.

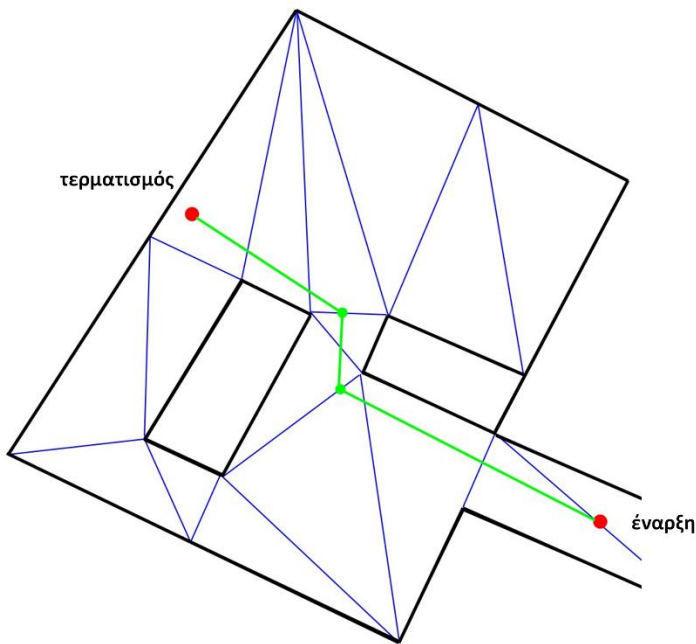
Στη συνέχεια θα εξετάσουμε ορισμένες σύνθετες ενέργειες που εκτελούνται συχνά από συνθετικούς χαρακτήρες σε εικονικούς κόσμους: την εύρεση μονοπατιού, τη δυναμική αποφυγή εμποδίων και την κίνηση πλήθους.

5.3 Εύρεση μονοπατιού

Το πρόβλημα της *εύρεσης μονοπατιού* (pathfinding) προκύπτει σε όλα σχεδόν τα συνθετικά περιβάλλοντα στα οποία υπάρχουν δυναμικά κινούμενες οντότητες και περιγράφεται από την ανάγκη εύρεσης του συντομότερου μονοπατιού μεταξύ δύο σημείων του χώρου (Cui & Shi, 2011). Με δεδομένο ότι σε ένα μεγάλο περιβάλλον περιλαμβάνονται συνήθως αντικείμενα διάφορων μεγεθών τα οποία παρεμποδίζουν τη μετακίνηση, προκύπτει το πρόβλημα της εύρεσης μονοπατιού στην περίπτωση που μια οντότητα θέλει να μετακινηθεί σε μια άλλη απομακρυσμένη περιοχή. Για παράδειγμα, ένας εικονικός πράκτορας που βρίσκεται σε ένα δωμάτιο ενός σπιτιού θέλει να μετακινηθεί σε ένα άλλο και παρεμβάλλονται έπιπλα, τοίχοι κ.ά. Στην περίπτωση αυτή χρησιμοποιούνται κατάλληλοι αλγόριθμοι που υπολογίζουν το συντομότερο ή το ταχύτερο μονοπάτι σε γράφο, με δημοφιλέστερο τον A* (άλφα άστρο – A star). Για να εφαρμοστεί όμως κάποιος αλγόριθμος εύρεσης μονοπατιού απαιτείται να έχει γίνει πρώτα κατάλληλη προ-επεξεργασία του χώρου και να έχει δημιουργηθεί ο αντίστοιχος γράφος που περιγράφει τις διασυνδέσεις μεταξύ περιοχών του περιβάλλοντος.

Η δημοφιλέστερη δομή περιγραφής ενός χώρου σε μορφή γραφήματος μετακίνησης είναι το *πλέγμα πλοήγησης* (navigation mesh). Σύμφωνα με αυτή, δημιουργείται ένα σύνολο κυρτών πολυγώνων που περιγράφουν τις περιοχές που είναι προσβάσιμες από οντότητες που βαδίζουν στο περιβάλλον. Κάθε πολύγωνο είναι ένας κόμβος του γραφήματος, ο οποίος συνδέεται μέσω ακμών με τα γειτονικά πολύγωνα. Επειδή τα πολύγωνα είναι κυρτά, υποθέτουμε ότι οποιοδήποτε σημείο εντός ενός πολυγώνου είναι προσβάσιμο από οποιοδήποτε άλλο εντός αυτού μέσω ευθείας μετακίνησης. Επομένως, αν ένας πράκτορας θέλει να μετακινηθεί από το σημείο A στο σημείο B, αρκεί να εντοπίσουμε το πολύγωνο P_A στο οποίο ανήκει το A και το P_B για το B αντίστοιχα και να τα θέσουμε ως αρχικό και τελικό κόμβο του γραφήματος. Στη συνέχεια ο αλγόριθμος εύρεσης μονοπατιού θα καταλήξει σε μια ακολουθία ενδιάμεσων κόμβων–πολυγώνων την οποία θα διατρέξει

ο χαρακτήρας για να φτάσει με ασφάλεια στον στόχο του. Το πλέγμα πλοήγησης κατασκευάζεται μία φορά, κατά τη διαμόρφωση του χώρου. Στην εικόνα 3.11 απεικονίζεται ένα πλέγμα πλοήγησης και η εύρεση διαδρομής σε αυτό.



Εικόνα 3.11 Πλέγμα πλοήγησης και εύρεση διαδρομής.

Η επίλυση του προβλήματος εύρεσης μονοπατιού γίνεται μέσω του αλγορίθμου A^* . Ο αλγόριθμος διατρέχει το γράφημα ξεκινώντας από τον αρχικό κόμβο και μετακινούμενος σε κόμβους-παιδιά, δηλαδή όσους συνδέονται άμεσα με ακμές σε αυτόν, και από αυτούς σε νέους κόμβους-παιδιά, κ.λπ., αναζητεί τον τελικό κόμβο. Η στρατηγική αναζήτησης είναι τέτοια που διασφαλίζεται ότι ο αλγόριθμος θα βρει το συντομότερο μονοπάτι, δηλαδή αυτό με τον μικρότερο αριθμό ενδιάμεσων κόμβων, ενώ παράλληλα είναι και αρκετά πιο γρήγορος στην αναζήτησή του σε σχέση με άλλους αλγορίθμους. Ο A^* για να επιλέξει ποιον κόμβο-παιδί θα επισκεφτεί σε κάθε βήμα, χρησιμοποιεί:

α) μια συνάρτηση h που κάνει εκτίμηση της απόστασης ενός κόμβου από τον τελικό κόμβο, η οποία ονομάζεται ευρετική συνάρτηση. Σε προβλήματα χωρικής αναζήτησης, όπως η εύρεση μονοπατιού, χρησιμοποιείται συνήθως η Ευκλείδεια απόσταση μεταξύ των δύο σημείων.

β) την τιμή g που περιγράφει την πραγματική απόσταση του κόμβου από τον αρχικό.

Τελικά επιλέγει να επισκεφτεί τον κόμβο με τη μικρότερη τιμή στο άθροισμα $h + g$, δηλαδή αυτόν που αναμένεται να οδηγήσει στον στόχο με το μικρότερο συνολικό αριθμό βημάτων.

Τα βήματα του αλγορίθμου είναι τα παρακάτω:

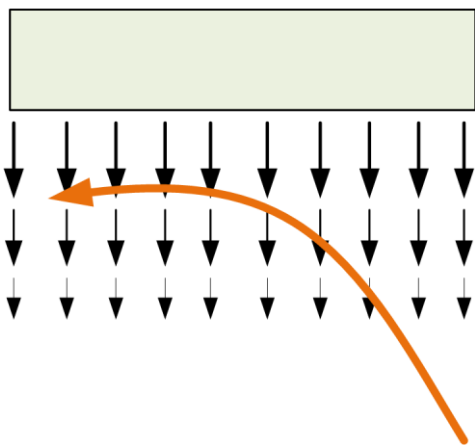
1. Βάλτε την αρχική κατάσταση στο μέτωπο αναζήτησης.
2. Αν το μέτωπο αναζήτησης είναι κενό σταμάτησε.
3. Επίλεξε τον κόμβο του μετώπου αναζήτησης με το μικρότερο άθροισμα $h + g$.
4. Αν ο κόμβος είναι μέλος του κλειστού συνόλου (των κόμβων που έχεις ήδη επισκεφτεί) πήγαινε στο βήμα 2.
5. Αν ο κόμβος είναι ο στόχος σταμάτησε.
6. Βρες τους κόμβους-παιδιά που συνδέονται άμεσα με τον κόμβο.
7. Υπολόγισε την τιμή του $h + g$ για κάθε κόμβο-παιδί.
8. Πρόσθεσε τους κόμβους-παιδιά στο μέτωπο αναζήτησης.
9. Πρόσθεσε τον κόμβο-γονέα στο κλειστό σύνολο.
10. Πήγαινε στο βήμα 2.

Οι αλγόριθμοι εύρεσης μονοπατιού μπορούν να επιλύσουν ικανοποιητικά το πρόβλημα της ασφαλούς και γρήγορης μετακίνησης χαρακτήρων σε στατικούς χώρους. Στην περίπτωση όμως που παρεμβάλλονται κινούμενα αντικείμενα στην πορεία του χαρακτήρα, οι παραπάνω μέθοδοι δεν μπορούν να δώσουν λύση.

5.4 Δυναμική αποφυγή εμποδίων

Η αποφυγή εμποδίων κατά τη μετακίνηση ενός χαρακτήρα είναι μια διαδικασία ανεξάρτητη από την εύρεση μονοπατιού και εκτελείται καθ' όλη την πορεία της κίνησης. Η συνήθης προσέγγιση είναι να ελέγχεται μια περιοχή γύρω από τον χαρακτήρα για εμπόδια και, στην περίπτωση που εντοπιστεί κάποιο στατικό ή κινούμενο αντικείμενο που παρεμβάλλεται στην πορεία του, να τροποποιείται η διεύθυνση ή και η ταχύτητα της κίνησης για την αποφυγή του. Η διαδικασία αυτή είναι ιδιαίτερα χρήσιμη στην περίπτωση κίνησης μεγάλου αριθμού χαρακτήρων, όπως η κίνηση πλήθους, όπου στόχος είναι να αποφεύγονται οι μεταξύ τους συγκρούσεις. Αλλά ακόμα και σε περιβάλλοντα στα οποία γίνεται δυναμική προσθήκη ή τροποποίηση αντικειμένων, τα οποία δεν έχουν συμπεριληφθεί στο πλέγμα πλοήγησης, είναι απαραίτητο να υπάρχει μια διαδικασία για την αποφυγή τους.

Η συνήθης προσέγγιση στην αποφυγή εμποδίων είναι με τη χρήση ενός «πεδίου δυνάμεων» (*force field*). Σύμφωνα με αυτή την πρακτική, κάθε εμπόδιο ασκεί μια βοηθή δύναμη στον κινούμενο χαρακτήρα η οποία επηρεάζει τη διεύθυνση μετακίνησής του (Borenstein & Koren, 1991). Για την εφαρμογή της προσέγγισης αυτής, αρχικά ορίζεται μια απόσταση στην οποία κοιτάζει μπροστά ο κινούμενος χαρακτήρας, το λεγόμενο πεδίο όρασης (*line of sight*). Κατά τη διάρκεια της μετακίνησής του εντοπίζονται τα αντικείμενα που βρίσκονται εντός του πεδίου όρασης και με τα οποία υπάρχει πιθανότητα σύγκρουσης και, αν υπάρχουν, επιλέγεται το κοντινότερο από αυτά ως μεγαλύτερη απειλή. Σε αυτό υπολογίζεται μια δύναμη αποφυγής (*avoidance force*), η οποία μεγαλώνει όσο μικραίνει η απόσταση μεταξύ των ορίων του χαρακτήρα και των ορίων του αντικειμένου και έχει διεύθυνση προς την εξωτερική επιφάνεια του αντικειμένου. Η δύναμη αυτή επηρεάζει την ταχύτητα μετακίνησης και το αποτέλεσμα είναι μια κίνηση γύρω από τα όρια του εμποδίου η οποία καταλήγει με τον χαρακτήρα ξανά προσανατολισμένο στον τελικό του στόχο. Στην εικόνα 3.12 απεικονίζεται ένα παράδειγμα επίδρασης του πεδίου δυνάμεων στην κίνηση ενός αντικειμένου.



Εικόνα 3.12 Αποφυγή εμποδίων με «πεδίο δυνάμεων».

5.5 Συντονισμένη κίνηση ομάδων

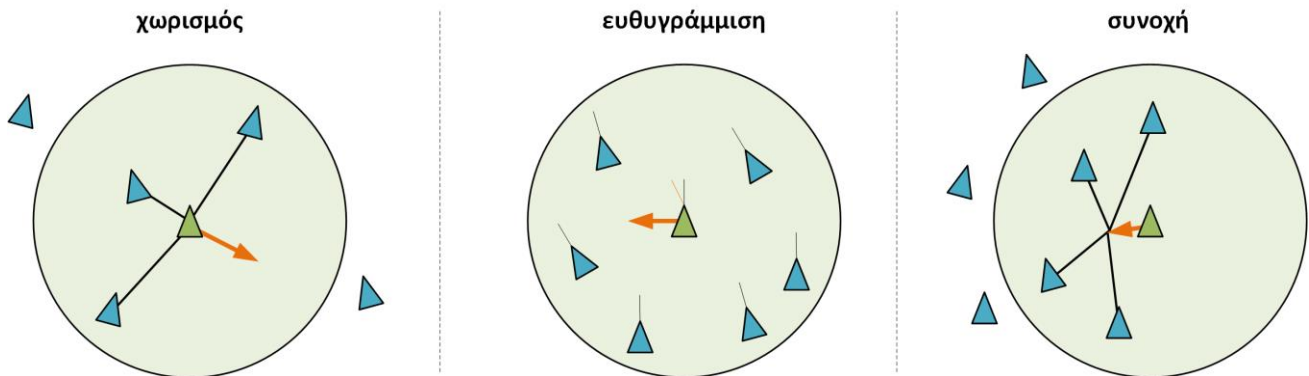
Σε ορισμένες περιπτώσεις εικονικών κόσμων προκύπτει η ανάγκη ταυτόχρονης κίνησης μεγάλου αριθμού χαρακτήρων ελεγχόμενων από τον υπολογιστή, όπως για παράδειγμα σε περιβάλλοντα προσομοίωσης και παιχνίδια. Η κίνηση πλήθους είναι μια αρκετά σύνθετη διαδικασία για την οποία η απλή αποφυγή εμποδίων δεν αρκεί. Πράγματι, αν μελετήσουμε τον τρόπο που κινείται το ανθρώπινο πλήθος σε διάφορους δημόσιους χώρους, για παράδειγμα σε πεζοδρόμια ή σε εμπορικά κέντρα, θα παρατηρήσουμε ότι, παρά τις επιμέρους διαφοροποιήσεις, οι άνθρωποι έχουν την τάση να ακολουθούν μια συγκεκριμένη ροή και να συντονίζονται μεταξύ τους ώστε να διευκολύνεται η κίνηση. Υπάρχουν μάλιστα περιπτώσεις οι οποίες είναι αρκετά σύνθετες και για τις οποίες απαιτείται προσεκτικός συντονισμός, όπως για παράδειγμα όταν άνθρωποι ερχόμενοι από

διαφορετικές διευθύνσεις πρέπει να εισέλθουν και να περάσουν μέσα από έναν διάδρομο συγκεκριμένης διεύθυνσης. Έχουν αναπτυχθεί πολλά εξειδικευμένα μοντέλα προσομοίωσης για την απόδοση της κίνησης του πλήθους, τα οποία διαφοροποιούνται ως προς το υπολογιστικό κόστος αλλά και την ευελιξία τους στην αντιμετώπιση νέων καταστάσεων (Duijves κ.ά., 2013).

Μία βασική τεχνική που χρησιμοποιείται σε συστήματα συνθετικής κίνησης για την απόδοση της κίνησης ομάδας είναι η *κίνηση σμήνους (flocking)*. Η τεχνική αυτή, η οποία προτάθηκε από το Reynolds (1987), μπορεί να κάνει πολλαπλούς χαρακτήρες να κινηθούν με φυσικό τρόπο ακολουθώντας κάποιον «αρχηγό» (*leader*). Το αποτέλεσμα είναι να αποδίδεται με αρκετή αληθοφάνεια η κίνηση ομάδων χαρακτήρων (π.χ. οικογένεια, παρέα) σε έναν εικονικό χώρο, ενώ ο σχεδιαστής χρειάζεται να περιγράψει μόνο την κίνηση του αρχηγού. Οι βασικοί κανόνες για την κίνηση σμήνους είναι τρεις:

- *Χωρισμός*: κάθε χαρακτήρας θα πρέπει να απομακρυνθεί από τους κοντινούς χαρακτήρες για να αποφευχθεί η σύγκρουση.
- *Ευθυγράμμιση*: η διεύθυνση κίνησης του χαρακτήρα θα πρέπει να ευθυγραμμιστεί με τον μέσο όρο των διευθύνσεων των γειτονικών χαρακτήρων.
- *Συνοχή*: ο χαρακτήρας θα πρέπει να κινηθεί προς τη μέση θέση των γειτονικών χαρακτήρων ώστε να παραμείνει στην ομάδα.

Κάθε χαρακτήρας έχει μια περιορισμένη σφαίρα όρασης, εντός της οποίας παρατηρεί τις θέσεις και τους προσανατολισμούς των υπόλοιπων χαρακτήρων. Με βάση αυτές, εφαρμόζει τους παραπάνω κανόνες και συνδυάζει τα αποτελέσματα για να καταλήξει στην τελική κίνηση και προσανατολισμό του. Στην εικόνα 3.13 απεικονίζονται οι τρεις παραπάνω κανόνες για την κίνηση σμήνους.



Εικόνα 3.13 Κίνηση σμήνους.

Τέλος, πέρα από την κίνηση σμήνους, συχνά χρησιμοποιούμενες συμπεριφορές πλήθους είναι επίσης (Musse & Thalmann, 2001):

- η χρήση σημείων έλξης ή απώθησης, τα οποία επηρεάζουν την πορεία των χαρακτήρων,
- η ένταξη σε ομάδα ή ο διαχωρισμός ομάδων σε μικρότερες,
- η αλλαγή στόχων, για παράδειγμα εξαιτίας κάποιου νέου σημείου ενδιαφέροντος,
- η αραίωση του πλήθους όταν εισέρχεται σε μεγαλύτερους χώρους κ.ά.

6 Συμπεράσματα

Από την παρουσίαση των διάφορων τεχνικών παραγωγής συνθετικής κίνησης στους εικονικούς κόσμους προκύπτει το συμπέρασμα ότι υπάρχει ένα πολύ μεγάλο εύρος περιπτώσεων και αναγκών, το οποίο δεν είναι δυνατόν να καλυφθεί με μία και μόνο τεχνική. Πράγματι, σε ένα κατασκευασμένο περιβάλλον μπορεί να παράγονται από πολύ απλές κινήσεις, όπως μια πόρτα που ανοίγει και κλείνει, μέχρι εξαιρετικά σύνθετες, όπως εικονικοί πράκτορες που κινούνται και αλληλεπιδρούν με το περιβάλλον. Μπορεί να υπάρχουν κινήσεις τις οποίες γνωρίζουμε πλήρως εκ των προτέρων, όπως, για παράδειγμα η κίνηση των δεικτών ενός ρολογιού,

κινήσεις οι οποίες μπορούν να παραμετροποιηθούν ώστε να ταιριάζουν καλύτερα με τα μοντέλα στα οποία εφαρμόζονται, για παράδειγμα το βάδισμα των χαρακτήρων, ή και κινήσεις που παράγονται δυναμικά και για τις οποίες δεν μπορούμε να γνωρίζουμε καθόλου την εξέλιξή τους, όπως για παράδειγμα η εκτόξευση ενός αντικειμένου σε έναν κλειστό χώρο με έπιπλα και άλλα αντικείμενα. Σε κάποιες από αυτές μας ενδιαφέρει η φυσικότητα του αποτελέσματος, ενώ άλλες μπορεί να είναι σημαντικά απλοποιημένες. Σε κάθε περίπτωση, ένας εικονικός κόσμος είναι πιθανόν να έχει ανάγκη για τις περισσότερες από τις παραπάνω κατηγορίες που περιγράψαμε. Είναι βέβαιο ότι θα υπάρχουν αντικείμενα που κινούνται δυναμικά, άρα θα χρειαστούν οπωσδήποτε διαδικαστικές τεχνικές, ενώ και η φυσική μοντελοποίηση θα πρέπει να υπάρχει, έστω και απλοποιημένη, για το φυσικό βάδισμα των ενσαρκώσεων. Σε πιο σύνθετες εφαρμογές θα προκύψει ανάγκη και για αυτόνομους χαρακτήρες οι οποίοι θα πρέπει να κινούνται και να λαμβάνουν αποφάσεις εντός του περιβάλλοντος.

Πέρα από τις ανάγκες της εφαρμογής, και οι δυνατές λύσεις στην παραγωγή των δεδομένων κίνησης διαφοροποιούνται αρκετά τόσο ως προς την ποιότητα του αποτελέσματος όσο και ως προς την ευκολία χρήσης τους. Η κίνηση με θέσεις-κλειδιά, αν και είναι εξαιρετικά δημοφιλής και υποστηρίζεται σήμερα από διάφορα εργαλεία σχεδίασης, απαιτεί εξειδικευμένους σχεδιαστές, έχει σημαντικό χρόνο ανάπτυξης και το αποτέλεσμα δεν είναι παραμετροποιήσιμο. Η σύλληψη κίνησης που μπορεί να δώσει φυσικά αποτελέσματα σε σύντομο χρονικό διάστημα απαιτεί ακριβό και εξειδικευμένο υλικό. Τέλος, η διαδικαστική κίνηση περιορίζει τις δυνατότητες ελέγχου στο τελικό αποτέλεσμα από τη μεριά του σχεδιαστή και μπορεί να περιλαμβάνει τη χρήση περίπλοκων και δύσκολων στην ανάπτυξη αλγορίθμων. Θα πρέπει λοιπόν ένας σχεδιαστής να είναι σε θέση να αναγνωρίσει τις ανάγκες κίνησης των αντικειμένων και των χαρακτήρων του περιβάλλοντος και σε συνδυασμό με τις προσφερόμενες δυνατότητες των διάφορων τεχνικών να επιλέξει την κατάλληλη για το είδος της εφαρμογής.

Αναφορές

- Baraff, D. (2001). Physically based modeling: Rigid body simulation. *SIGGRAPH Course Notes, ACM SIGGRAPH*, 2(1), 2–1.
- Berger, K., Ruhl, K., Schroeder, Y., Bruemmer, C., Scholz, A. & Magnor, M. (2011). Markerless Motion Capture using multiple Color-Depth Sensors. *Sensors Peterborough NH, 2011*, 317–324.
- Borenstein, J. & Koren, Y. (1991). The Vector Field Histogram-Fast Obstacle A voidance for Mobile Robots. *IEEE Transactions on Robotics and Automation*, 7(3), 278–288.
- Brown, J., Sorkin, S., Bruyns, C., Latombe, J.-C., Montgomery, K. & Stephanides, M. (2001). Real-time simulation of deformable objects: tools and application. *Proceedings Computer Animation 2001. Fourteenth Conference on Computer Animation (Cat. No.01TH8596)*.
- Cui, X. & Shi, H. (2011). A*-based pathfinding in modern computer games. *International Journal of Computer Science and Network Security*, 11(1), 125-130.
- Duives, D. C., Daamen, W. & Hoogendoorn, S. P. (2013). State-of-the-art crowd motion simulation models. *Transportation Research Part C: Emerging Technologies*, 37, 193–209.
- Gleicher, M. (1999). Animation from observation: Motion capture and motion editing. *ACM SIGGRAPH Computer Graphics*, 33(4), 1–5.
- Gottschalk, S., Lin, M. C. & Manocha, D. (1996). OBB Tree: A Hierarchical Structure for Rapid Interference Detection. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques* (pp. 171–180).

- Lasseter, J. (1987). Principles of traditional animation applied to 3D computer animation. *ACM SIGGRAPH Computer Graphics*, 21(4), 35–44.
- Lin, Y. H., Liu, C. Y., Lee, H. W., Huang, S. L. & Li, T. Y. (2009). Evaluating emotive character animations created with procedural animation. *Lecture Notes in Computer Science*, 5773 LNAI, 308–315.
- Luck, M. & Aylett, R. (2000). Applying artificial intelligence to virtual reality: Intelligent virtual environments. *Applied Artificial Intelligence*, 14(1), 3–32.
- Moore, M. & Wilhelms, J. (1988). Collision Detection and Response for Computer Animation. *ACM SIGGRAPH Computer Graphics*, 22(4), 289–298.
- Musse, S. R. & Thalmann, D. (2001). Hierarchical model for real time simulation of virtual human crowds. *IEEE Transactions on Visualization and Computer Graphics*, 7(2), 152–164.
- Parent, R. (2012). *Computer Animation Algorithms & Techniques* (3rd Edition). Elsevier.
- Provot, X. (1995). Deformation Constraints in a Mass Spring Model to Describe Rigid Cloth Behavior. *Graphics Interface*, 147–154.
- Reeves, W. T. (1983). Particle systems - a technique for modeling a class of fuzzy objects. *ACM SIGGRAPH Computer Graphics*, 17(3), 359–375.
- Reynolds, C. W. (1987). Flocks, herds and schools: A distributed behavioral model. *ACM SIGGRAPH Computer Graphics*, 21(4), 25–34.
- Sederberg, T. W. & Parry, S. R. (1986). Free-form deformation of solid geometric models. *ACM SIGGRAPH Computer Graphics*, 20(4), 151–160.
- Thalmann, N. & Thalmann, D. (1990). *Computer Animation*. Springer.
- Tolani, D., Goswami, A & Badler, N. I. (2000). Real-time inverse kinematics techniques for anthropomorphic limbs. *Graphical Models*, 62(5), 353–388.
- Twigg, C. (2003). Catmull-Rom splines. *Computer*, 41(6), 4–6.
- Vosinakis, S. & Panayiotopoulos, T. (2001). SimHuman: A platform for real-time virtual agents with planning capabilities. *Proc. of Intelligent Virtual Agents*, 210–223.
- Wooldridge, M. & Jennings, N. R. (1995). Intelligent agents: theory and practice. *The Knowledge Engineering Review*, 10(2), 115–152.

Περαιτέρω Μελέτη

Μπορείτε να μελετήσετε σε βάθος τις διάφορες τεχνικές και αλγορίθμους συνθετικής κίνησης στο πολύ καλό βιβλίο του R. Parent:

Parent, R. (2012). *Computer Animation Algorithms & Techniques* (3rd Edition). Elsevier.

Περισσότερες πληροφορίες σχετικά με τη φυσική μοντελοποίηση μπορείτε να βρείτε στο Κεφάλαιο 8 του βιβλίου:

Vince, J. (1995). *Virtual Reality Systems*. Addison-Wesley.

Σχετικά με τους εικονικούς πράκτορες και την κίνηση πλήθους, μπορείτε να μελετήσετε το βιβλίο:
Pelechano, N., Allbeck, J. & Badler, N. (2008). *Virtual Crowds: Methods, Simulation, and Control* (Synthesis Lectures on Computer Graphics and Animation). Morgan and Claypool Publishers.

Ερωτήσεις Κατανόησης

1. Ποια είναι τα είδη κίνησης που μπορεί να συναντήσει κάποιος στους εικονικούς κόσμους;
2. Ποια είναι τα βασικά μοντέλα που χρησιμοποιούνται για την υποστήριξη συνθετικών κινήσεων;
3. Περιγράψτε με δικά σας λόγια το πρόβλημα της αντίστροφης κινηματικής.
4. Εξηγήστε τη διαδικασία κίνησης παραμορφώσιμων αντικειμένων μέσω μορφοποίησης.
5. Σε ποιες περιπτώσεις αντικειμένων είναι προτιμότερη η χρήση συστημάτων σωματιδίων;
6. Περιγράψτε πώς μοντελοποιούνται οι συνθετικοί χαρακτήρες για να υποστηριχθεί η κίνησή τους.
7. Ποια είναι τα μειονεκτήματα της κίνησης με πλάνα-κλειδιά;
8. Τι είναι η διαδικαστική συνθετική κίνηση και πώς μπορεί να χρησιμοποιηθεί στους εικονικούς κόσμους;
9. Περιγράψτε έναν γενικό απλοποιημένο αλγόριθμο κίνησης παθητικών αντικειμένων με βάση τη φυσική.
10. Ποια είναι η διαφορά μεταξύ των ευθυγραμμισμένων στους άξονες και των προσανατολισμένων bounding boxes; Ποια λύση είναι προτιμότερη για κινούμενα αντικείμενα;
11. Περιγράψτε με δικά σας λόγια τη δομή και λειτουργία των μηχανών καταστάσεων.
12. Να αναφέρετε τρεις βασικές αρχιτεκτονικές ευφών πρακτόρων και αντίστοιχα παραδείγματα εφαρμογής τους σε εικονικούς κόσμους.
13. Εξηγήστε τη διαφορά μεταξύ της εύρεσης μονοπατιού και της αποφυγής εμποδίων.
14. Περιγράψτε με δικά σας λόγια πώς θα μπορούσε να παραχθεί συντονισμένη κίνηση μιας ομάδας χαρακτήρων.

Ασκήσεις

1. Μελετήστε κάποιο παιχνίδι της αρεσκείας σας και προσπαθήστε να καταγράψετε τη συμπεριφορά βασικών χαρακτήρων του με μηχανές καταστάσεων.
2. Εγκαταστήστε το QAnimator, ένα ελεύθερο πρόγραμμα για συνθετικές κινήσεις χαρακτήρων με θέσεις-κλειδιά, και δοκιμάστε να κατασκευάσετε φυσικές κινήσεις βαδίσματος και χαιρετισμού.
3. Χρησιμοποιήστε το Unity για να κατασκευάσετε μια σκηνή με φυσικά αντικείμενα που πέφτουν με τη βαρύτητα και συγκρούονται μεταξύ τους. Στη συνέχεια, εξερευνήστε τις δυνατότητες του μηχανισμού φυσικής της εφαρμογής και καταγράψτε τα προβλήματα που εντοπίσατε.
4. Δοκιμάστε να κατασκευάσετε με κώδικα στο Unity τη φυσική κίνηση ενός αντικειμένου που πέφτει από συγκεκριμένο ύψος στο έδαφος χωρίς να χρησιμοποιήσετε τον ενσωματωμένο μηχανισμό φυσικής. Αρχικά δοκιμάστε να το κάνετε να σταματάει στο έδαφος και στη συνέχεια να αναπηδάει με κάποια απώλεια ενέργειας μέχρι, τελικά, να σταματήσει.

Κεφάλαιο 4: Διεπαφή

Σύνοψη

Αντικείμενο του κεφαλαίου είναι η διεπαφή του εικονικού κόσμου με τον χρήστη. Αρχικά παρουσιάζονται οι βασικές κατηγορίες αλληλεπίδρασης χρηστών με το τρισδιάστατο περιβάλλον: πλοήγηση, χειρισμός αντικειμένων και επικοινωνία. Για καθεμία από αυτές παρουσιάζονται οι επιμέρους στόχοι και οι συχνά χρησιμοποιούμενες τεχνικές αλληλεπίδρασης για την εκτέλεσή τους. Μελετώνται ζητήματα ερμηνείας των δεδομένων εισόδου, αντιστοίχισης σε ενέργειες στο περιβάλλον και χρήσης οπτικών βοηθημάτων, και αναλύονται τόσο θέματα υλοποίησης όσο και σχεδιαστικά ζητήματα. Στη συνέχεια, αναφέρονται θέματα σχεδίασης της διδιάστατης διεπαφής (head up display – HUD) του χρήστη με τον κόσμο, με έμφαση στην κατάλληλη επιλογή και διάταξη οπτικών βοηθημάτων και στοιχείων διεπαφής. Τέλος, παρουσιάζονται σύγχρονες κατηγορίες υλικού για φυσικές αλληλεπιδράσεις και οι δυνατότητες αξιοποίησής τους στους εικονικούς κόσμους.

Προαπαιτούμενη γνώση

Απαιτούνται γνώσεις πληροφορικής και κατανόηση των βασικών εννοιών των εικονικών κόσμων που παρουσιάζονται στο 1^ο Κεφάλαιο.

1 Αλληλεπίδραση χρηστών στους εικονικούς κόσμους

Η αλληλεπίδραση των χρηστών με το περιβάλλον είναι ένα αναπόσπαστο συστατικό των εικονικών κόσμων. Όπως αναφέραμε και στο πρώτο κεφάλαιο, βασικό χαρακτηριστικό των εικονικών περιβαλλόντων που τα διαφοροποιεί από άλλα μέσα είναι η δυνατότητα των χρηστών να πλοηγούνται ελεύθερα μέσα σε αυτά, να επικοινωνούν και να αλληλεπιδρούν με τα αντικείμενα του περιβάλλοντος. Όμως, σε αντίθεση με τα διδιάστατα παραθυρικά περιβάλλοντα με τα οποία έχουμε συνηθίσει να αλληλεπιδρούμε καθημερινά μέσω των διάφορων υπολογιστικών συσκευών που χρησιμοποιούμε, οι αλληλεπιδράσεις με τα τρισδιάστατα περιβάλλοντα είναι σημαντικά πιο σύνθετες και περίπλοκες. Η προσθήκη μιας παραπάνω διάστασης στο μοντέλο του περιβάλλοντος εγείρει εξειδικευμένες ανάγκες μετακίνησης και αλλαγής οπτικής γωνίας. Επιπλέον, οι ενέργειες που εκτελούν οι χρήστες μέσω παραδοσιακών διεπαφών κατασκευασμένων για κίνηση στις δύο διαστάσεις, όπως το ποντίκι ή η οθόνη αφής, θα πρέπει να μεταφραστούν σε τρεις με τρόπο διαισθητικό ή έστω εύκολα κατανοητό στους χρήστες, χωρίς παράλληλα να περιορίζονται οι δυνατότητές τους. Οι παραπάνω ιδιαιτερότητες έχουν αποτελέσει σχεδιαστικές προκλήσεις για τους σχεδιαστές εικονικών περιβαλλόντων, με αποτέλεσμα να έχει προκύψει ένας μεγάλος αριθμός από τεχνικές και βοηθήματα για τα διάφορα είδη αλληλεπίδρασης. Η επιτυχία της κάθε σχεδιαστικής λύσης επηρεάζεται από δύο επιπλέον παράγοντες: τις συσκευές εισόδου που χρησιμοποιούν οι χρήστες και το είδος της δραστηριότητας που καλούνται να εκτελέσουν στο περιβάλλον.

Αν προσπαθήσουμε να κατηγοριοποιήσουμε τα είδη των αλληλεπιδράσεων χρήστη στους εικονικούς κόσμους, θα εντοπίσουμε τρεις βασικές κατηγορίες: την πλοήγηση, την αλληλεπίδραση με τα αντικείμενα και την επικοινωνία με τους υπόλοιπους χρήστες (Mine, 1995). Η πλοήγηση είναι μια πολύ βασική ενέργεια για τους χρήστες ενός εικονικού κόσμου, καθώς τους επιτρέπει να εξερευνούν το περιβάλλον, να μετακινούνται σε απομακρυσμένες περιοχές, να παρατηρούν τα περιεχόμενά του από διάφορες οπτικές γωνίες και να προσεγγίζουν αντικείμενα ή άλλους χρήστες με σκοπό τη διάδραση και την επικοινωνία. Η αλληλεπίδραση με τα αντικείμενα του περιβάλλοντος σχετίζεται με τις ενέργειες που μπορεί να κάνει ένας χρήστης στο περιβάλλον και περιλαμβάνει την επιλογή του αντικειμένου και την εκτέλεση ενεργειών πάνω σε αυτό, όπως αλλαγή θέσης και προσανατολισμού, άσκηση δύναμης κ.ά. Τέλος, στην επικοινωνία με τους χρήστες εντάσσονται όλες εκείνες οι αλληλεπιδράσεις που σχετίζονται με τη λεκτική ή μη, σύγχρονη ή ασύγχρονη επικοινωνία μεταξύ των χρηστών του περιβάλλοντος, όπως η ομιλία, το γραπτό κείμενο, η κατασκευή επισημειώσεων, η διαχείριση ομάδων, η ανάθεση ρόλων κ.ά.

Μια επιτυχημένη προσέγγιση για τη βελτίωση της ευχρηστίας των αλληλεπιδράσεων είναι η χρήση μεταφορών διεπαφής (interface metaphors). Οι μεταφορές διεπαφής είναι μια έννοια που προέρχεται από την περιοχή της Αλληλεπίδρασης Ανθρώπου–Υπολογιστή (Human Computer Interaction) και περιγράφει τη σχεδίαση στοιχείων διεπαφής με τρόπο τέτοιο ώστε η χρήση τους να παραπέμπει σε ήδη γνωστά αντικείμενα και έννοιες του πραγματικού κόσμου (Mandel, 1997). Αυτή η αναγνώριση γνώριμων στοιχείων στη διεπαφή

μπορεί να βοηθήσει τους χρήστες να καταλάβουν τι πρέπει να κάνουν με αυτές ή/και να απομνημονεύσουν ευκολότερα τον τρόπο χρήσης τους, με αποτέλεσμα περισσότερο διαισθητικές και επιτυχημένες αλληλεπιδράσεις. Ένα χαρακτηριστικό παράδειγμα επιτυχημένων μεταφορών είναι οι φάκελοι και ο κάδος ανακύκλωσης στα περιβάλλοντα επιφάνειας εργασίας. Τα αντικείμενα αυτά παραπέμπουν στα αντίστοιχα φυσικά αντικείμενα που συναντάει κάποιος σε χώρους εργασίας: οι φάκελοι και οι υποφάκελοι χρησιμοποιούνται για την οργάνωση και ομαδοποίηση των αρχείων, ο κάδος ανακύκλωσης για την καταστροφή τους. Αυτή οι ομοιότητα κάνει ενέργειες όπως η καταχώρηση και η αναζήτηση ψηφιακών αρχείων μέσα σε φακέλους και η διαγραφή τους μέσω της τοποθέτησής τους στο εικονίδιο του κάδου ανακύκλωσης περισσότερο διαισθητικές. Βεβαίως για να είναι επιτυχημένη μια μεταφορά θα πρέπει να έχει γίνει μια καλή αντιστοίχιση μεταξύ των ενεργειών της νέας αλληλεπίδρασης και των γνώριμων ενεργειών στις οποίες αναφέρεται η μεταφορά· διαφορετικά το αποτέλεσμα ενδέχεται να προκαλέσει μεγαλύτερη σύγχυση και εκνευρισμό στον χρήστη.

Οι μεταφορές χρησιμοποιούνται συχνά στις τεχνικές αλληλεπίδρασης σε τρισδιάστατα περιβάλλοντα και εικονικούς κόσμους. Η πιο προφανής μεταφορά είναι ο άμεσος φυσικός έλεγχος, δηλαδή ο χειρισμός των αντικειμένων με τρόπο αντίστοιχο με αυτόν που συμβαίνει στον φυσικό κόσμο. Κάτι τέτοιο θα σήμαινε ιδανικά ότι, για να μετακινήσει ένας χρήστης ένα αντικείμενο από μία θέση σε άλλη, θα έπρεπε να το αγγίξει με το χέρι του, να το μεταφέρει στο επιθυμητό σημείο και στη συνέχεια να το απελευθερώσει. Στην πραγματικότητα τα πράγματα δεν είναι ακριβώς έτσι για δύο λόγους: πρώτον, υπάρχουν περιορισμοί από τις συσκευές εισόδου που κάνουν τις αλληλεπιδράσεις λιγότερο φυσικές και, δεύτερον, διότι πολλές φορές είναι προτιμότερο να θυσιαστεί ο ρεαλισμός για λόγους ευχρηστίας. Έτσι, ενώ στα περιβάλλοντα εμπύθισης είναι πράγματι δυνατό να αλληλεπιδράσουν οι χρήστες με κινήσεις του χεριού μέσω κάποιου γαντιού δεδομένων, στα περιβάλλοντα επιφάνειας εργασίας ο περιορισμός στο ποντίκι μειώνει τη φυσικότητα της κίνησης. Επιπλέον, αν κανείς επιλέγει ένα αντικείμενο και το μετακινεί με το ποντίκι, δεν υπάρχει λόγος να το έχει προσεγγίσει σε απόσταση που να είναι προσβάσιμο από το χέρι του avatar. Θα μπορούσε να το μετακινήσει και από μακριά, κάτι που θα διευκόλυνε την αλληλεπίδραση. Ακριβώς επειδή η μίμηση της πραγματικότητας δεν είναι πάντοτε αναγκαία, πολλές φορές χρησιμοποιούνται και μεταφορές που προσδίδουν στους χρήστες «υπερδυνάμεις», όπως το πέταγμα και η τηλεμεταφορά. Μπορεί να μη θυμίζουν ενέργειες του φυσικού κόσμου, αλλά είναι κινήσεις που μπορεί εύκολα να φανταστεί κάποιος και άρα να τις χρησιμοποιήσει με επιτυχία σε ένα νέο περιβάλλον. Τέλος, πολλές φορές χρησιμοποιούνται μεταφορές από ήδη γνωστές διεπαφές, οι οποίες μεταφέρονται στις τρεις διαστάσεις, όπως για παράδειγμα η τοποθέτηση παραθυρικών στοιχείων ελέγχου (π.χ. πεδίο εισαγωγής κειμένου, λίστα επιλογής) πάνω σε επιφάνειες τρισδιάστατων αντικειμένων.

Ένα σημαντικό σχεδιαστικό ζήτημα στη σχεδίαση αλληλεπιδράσεων σε εικονικούς κόσμους είναι η κατάλληλη αξιοποίηση των συσκευών εισόδου για την εκτέλεση των απαιτούμενων διεργασιών. Η κίνηση ενός άκαμπτου αντικειμένου σε τρεις διαστάσεις έχει έξι βαθμούς ελευθερίας (*6 degrees of freedom*), δηλαδή πρέπει να προσδιορίσει κάποιος την τιμή έξι ανεξάρτητων μεταβλητών για να καθορίσει την τοποθέτηση του αντικειμένου στον χώρο: τρεις για τη θέση και τρεις για την περιστροφή του. Αν λοιπόν κάποιος θέλει να πλοηγείται ελεύθερα στον χώρο (δηλαδή να αλλάζει τη θέση και την περιστροφή της κάμερας) ή να μετακινήσει ένα αντικείμενο οπουδήποτε στον χώρο, θα πρέπει να μπορεί να προσδιορίσει μέσω κάποιας διεπαφής τις έξι αυτές τιμές. Το πρόβλημα είναι ότι οι παραδοσιακές διεπαφές εισόδου έχουν πολύ λιγότερους βαθμούς ελευθερίας. Για παράδειγμα, το ποντίκι που κινείται στο οριζόντιο επίπεδο έχει δύο μόνο βαθμούς ελευθερίας. Μια τρίτη, ανεξάρτητη μεταβλητή είναι το πλήκτρο ή ο συνδυασμός πλήκτρων που μπορεί να είναι πατημένα κατά τη διάρκεια της κίνησης. Στην περίπτωση του πληκτρολογίου, τα πράγματα είναι ακόμα πιο δύσκολα: και εκεί μπορεί κάποιος μέσω των πλήκτρων διεύθυνσης (βέλη) να προσδιορίσει κίνηση σε δύο άξονες, μόνο που οι τιμές, σε αντίθεση με το ποντίκι, είναι διακριτές. Για παράδειγμα, στον κάθετο άξονα μπορεί να οριστεί διεύθυνση: πάνω, τίποτα ή κάτω. Δεν υπάρχει κάτι σαν «πολύ πάνω» ή «λίγο πάνω». Περιορισμοί σαν τους παραπάνω στις συσκευές εισόδου αναγκάζουν τους σχεδιαστές να καταφεύγουν σε λύσεις που περιλαμβάνουν συνδυασμούς πλήκτρων, ενεργειών ή ακόμα και συσκευών για την απόδοση όλων των βαθμών ελευθερίας.

2 Πλοήγηση

Η πλοήγηση είναι μια δράση η οποία είναι απαραίτητη για την επιτυχημένη χρήση ενός εικονικού κόσμου. Κατά τη διάρκεια της παρουσίας των χρηστών στο περιβάλλον προκύπτει συνήθως η ανάγκη να εξερευνήσουν τον χώρο, να μετακινηθούν σε περιοχές ενδιαφέροντος, να προσεγγίσουν αντικείμενα ή χρήστες κ.λπ. Ανάλογα με το είδος της δραστηριότητας στην οποία εμπλέκονται οι χρήστες ενδέχεται να διαφέρουν και οι απαιτήσεις

πλοήγησης. Σε κάποιες περιπτώσεις μπορεί να θέλουν απλά να φτάσουν σε ένα συγκεκριμένο γνωστό σημείο, ενώ σε άλλες μπορεί να ενδιαφέρονται να περιηγηθούν σε έναν νέο χώρο. Επιπλέον, μπορεί να προκύψουν και περιπτώσεις που χρειάζεται να παρατηρήσουν ένα αντικείμενο ή έναν χώρο ενδιαφέροντος από διάφορες οπτικές γωνίες, όπως για παράδειγμα να δουν ένα τρισδιάστατο μοντέλο από πολλές απόψεις για να κατανοήσουν τη γεωμετρία του ή να παρατηρήσουν μια περιοχή από ψηλά. Για τον λόγο αυτόν συνήθως οι εικονικοί κόσμοι προσφέρουν παραπάνω από μία τεχνικές πλοήγησης και αφήνουν τους χρήστες να επιλέξουν την κατάλληλη ανάλογα με τον στόχο.

Η πλοήγηση ως διεργασία μπορεί να έχει σημαντική διάρκεια και απαιτεί συνεχή είσοδο από τους χρήστες, κάτι που ενδέχεται να προκαλέσει δυσκολίες σε νέους, μη εξοικειωμένους χρήστες. Για τη μετακίνηση σε κάποιο απομακρυσμένο σημείο χρειάζεται να προσδιορίζει ο χρήστης την κίνηση της κάμερας ή της ενσάρκωσης και πολλές φορές μπορεί να προκύπτει η ανάγκη μεταβολής της διεύθυνσης για την αποφυγή εμποδίων (π.χ. έπιπλα) ή λεπτοί χειρισμοί για προσεκτική μετακίνηση μέσα από πόρτες, διαδρόμους κ.λπ. Κατά συνέπεια, χρήστες που δεν έχουν προηγούμενη εμπειρία χρήσης τρισδιάστατων εφαρμογών ή παιχνιδιών μπορεί να δυσκολευτούν στον χειρισμό της κίνησης, ιδιαίτερα αν η δομή του χώρου είναι περίπλοκη, ή ακόμα και να χάσουν τον προσανατολισμό τους μην ξέροντας τι πρέπει να κάνουν στη συνέχεια (Burigat & Chittaro, 2007). Αυτά τα προβλήματα έχουν οδηγήσει τους σχεδιαστές στην ανάπτυξη διάφορων οπτικών βοηθημάτων για την καλύτερη υποστήριξη της διαδικασίας.

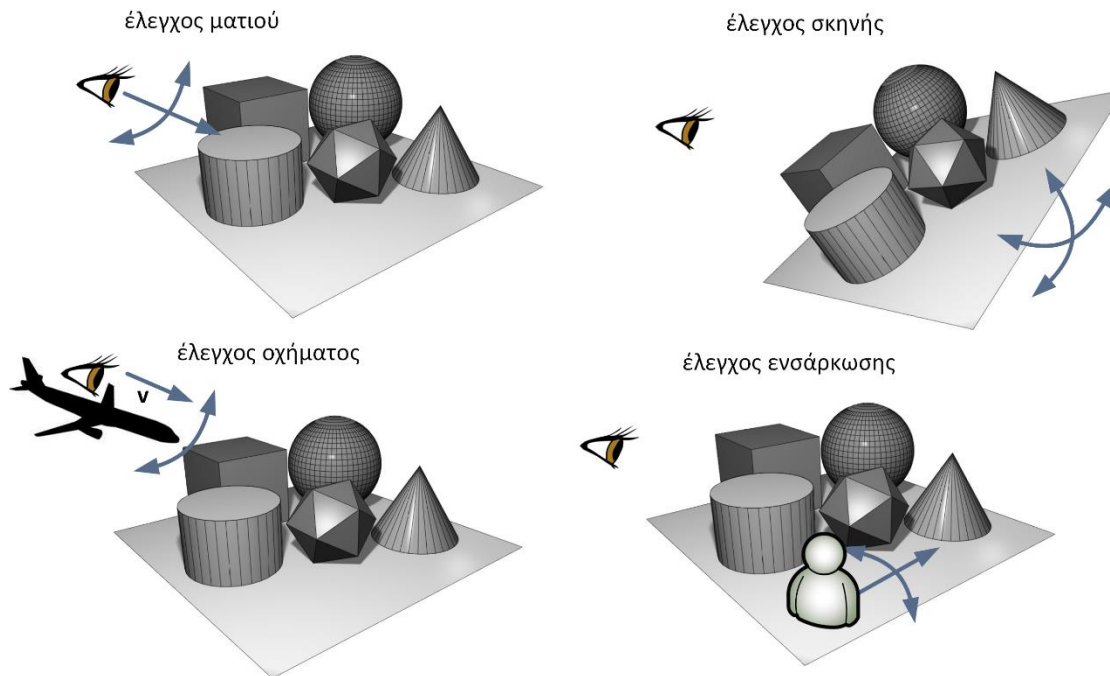
Η πλοήγηση αναλύεται σε δύο διακριτές ενέργειες, τη μετακίνηση (travel) και την εύρεση πορείας (wayfinding) (Bowman κ.ά., 2001). Μετακίνηση είναι ο τρόπος με τον οποίο οι χρήστες ελέγχουν την κίνηση της κάμερας στο περιβάλλον. Είναι η φυσική διεργασία που περιλαμβάνει τη συνεχή χρήση μιας συσκευής εισόδου για τον άμεσο προσδιορισμό της θέσης και του προσανατολισμού της οπτικής του χρήστη. Από την άλλη μεριά, η εύρεση πορείας έχει να κάνει με τη νοητική διεργασία αντίληψης της θέσης στην οποία βρίσκεται ο χρήστης και της πορείας που πρέπει να ακολουθήσει για να βρεθεί σε έναν συγκεκριμένο στόχο. Η ικανότητα του χρήστη να εκτελέσει ικανοποιητικά και τις δύο παραπάνω διεργασίες είναι απαραίτητη για την επιτυχημένη πλοήγησή του σε τρισδιάστατα περιβάλλοντα.

2.1 Τεχνικές μετακίνησης

Οι βασικές μεταφορές μετακίνησης σε τρεις διαστάσεις είναι οι παρακάτω:

- *Έλεγχος «ματιού» (Eyeball in hand)*: ο χρήστης ελέγχει άμεσα τη θέση και τον προσανατολισμό της κάμερας. Οι κινήσεις που κάνει μέσω της συσκευής εισόδου μεταφράζονται σε αντίστοιχες κινήσεις της κάμερας.
- *Έλεγχος σκηνής (Scene in hand)*: σύμφωνα με αυτήν τη μεταφορά, η κάμερα βρίσκεται σε σταθερή θέση και προσανατολισμό και ο χρήστης μετακινεί ολόκληρη τη σκηνή. Φυσικά αυτό που πραγματικά συμβαίνει σε επίπεδο γράφου σκηνικού είναι ότι μετακινείται η κάμερα και όχι η σκηνή, αλλά οι τεχνικές αλληλεπίδρασης που βασίζονται σε αυτή τη μεταφορά είναι σχεδιασμένες με τέτοιο τρόπο ώστε να δίνουν στον χρήστη την αίσθηση ότι μετακινεί τη σκηνή. Για παράδειγμα, μπορεί ο χρήστης να επιλέξει με το ποντίκι του μια περιοχή στο έδαφος και να τη σύρει προς κάποια διεύθυνση και να μετακινηθεί ολόκληρη η σκηνή προς τη διεύθυνση αυτή.
- *Έλεγχος οχήματος (Vehicle control)*: ο χρήστης δεν ελέγχει άμεσα τη θέση και τον προσανατολισμό της κάμερας, αλλά έμμεσα, χειριζόμενος κάποιο νοητό όχημα. Για παράδειγμα, η κίνηση που ορίζει στις συσκευές εισόδου μεταφράζεται σε ενέργειες για το πιλοτάρισμα ενός αεροσκάφους ή την οδήγηση ενός αυτοκινήτου. Πρακτικά η διαφορά σε σχέση με τον έλεγχο ματιού είναι ότι, αντί για τον προσδιορισμό της μετατόπισης της κάμερας, ο χρήστης προσδιορίζει μεταβολές στην ταχύτητα κίνησής της.
- *Έλεγχος ενσάρκωσης (Avatar control)*: ο χρήστης ελέγχει το βάδισμα ενός ανθρώπινου χαρακτήρα (ενσάρκωσης) μέσω της συσκευής εισόδου σε προοπτική πρώτου (από τα μάτια της ενσάρκωσης) ή τρίτου προσώπου. Η κίνηση είναι πιο φυσική σε σχέση με τον άμεσο έλεγχο της κάμερας, ακολουθεί τη μορφολογία του εδάφους λόγω του βαδίσματος της ενσάρκωσης σε αυτό και υπόκειται σε συγκρούσεις με άλλα στερεά αντικείμενα του περιβάλλοντος.

Η κυρίαρχη μεταφορά στους εικονικούς κόσμους είναι ο έλεγχος ενσάρκωσης, κάτι που είναι αναμενόμενο καθώς η αλληλεπίδραση των χρηστών με το περιβάλλον γίνεται μέσω των ενσαρκώσεων. Παρόλα αυτά, υπάρχουν και περιπτώσεις που προκύπτει ανάγκη για διαφορετικού τύπου μετακινήσεις, όπως, για παράδειγμα, όταν ένας χρήστης θέλει να εξερευνήσει ένα αντικείμενο ή να σχεδιάσει έναν χώρο παρατηρώντας τον από πολλές απόψεις. Στη συνέχεια θα εξετάσουμε ορισμένες βασικές τεχνικές μετακίνησης βασισμένες στις παραπάνω μεταφορές και αντίστοιχα σχεδιαστικά ζητήματα όταν η είσοδος περιορίζεται στις τυπικές συσκευές εισόδου των προσωπικών υπολογιστών, δηλαδή στο ποντίκι και στο πληκτρολόγιο.



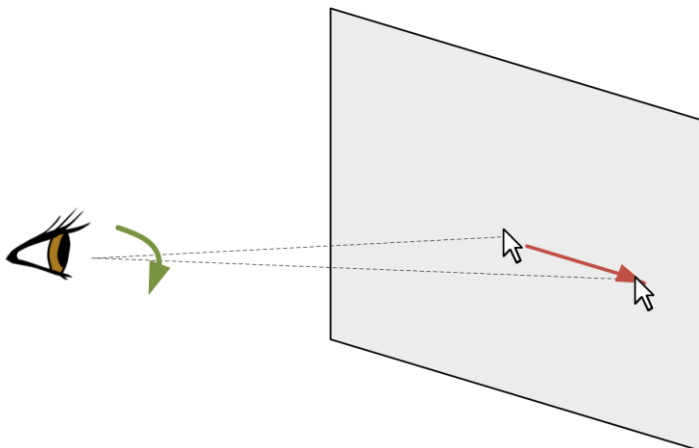
Εικόνα 4.1 Βασικές μεταφορές μετακίνησης σε τρεις διαστάσεις.

Για τον άμεσο έλεγχο της κίνησης της ενσάρκωσης ή της κάμερας μέσω πληκτρολογίου ή ποντικού προκύπτει και πάλι το πρόβλημα του μικρότερου αριθμού βαθμών ελευθερίας των συσκευών εισόδου. Αν θέλουμε να ελέγξουμε πλήρως την κίνηση της κάμερας, χρειαζόμαστε έξι βαθμούς ελευθερίας. Για τον έλεγχο του βαδίσματος μιας ενσάρκωσης επαρκούν αρχικά τρεις: η μετακίνηση στους δύο οριζόντιους άξονες και η περιστροφή στον κάθετο (Z). Όμως πολλές φορές μπορεί να θέλουμε να κοιτάξουμε προς τα κάτω ή προς τα πάνω, ιδιαίτερα στην προοπτική πρώτου προσώπου, άρα χρειάζεται ένας ακόμη βαθμός, η περιστροφή στον άξονα Y. Τέλος, αν ο χαρακτήρας μας μπορεί να κάνει άλματα, να σκύβει ή ακόμα και να πετάει, χρειάζεται να ελέγξουμε και έναν τρίτο άξονα μετακίνησης, τον κάθετο. Συνολικά λοιπόν μπορεί να χρειαστεί να ελέγξουμε πέντε βαθμούς ελευθερίας.

Μια απλή τεχνική για τον έλεγχο κίνησης μιας ενσάρκωσης είναι η κίνηση προς τη διεύθυνση που κοιτάζει η κάμερα και η περιστροφή στον κάθετο άξονα με τη χρήση του πληκτρολογίου. Συνήθως χρησιμοποιούνται τα βέλη ή άλλη αντίστοιχη τετράδα πλήκτρων (στα παιχνίδια πρώτου προσώπου χρησιμοποιούνται και τα πλήκτρα W, A, S, D), έτσι ώστε τα δύο πλήκτρα να ορίζουν κίνηση προς τα εμπρός ή προς τα πίσω και τα άλλα δύο περιστροφή προς τα αριστερά ή τα δεξιά. Η κίνηση αυτή πολλές φορές συμπληρώνεται με κάποιο ειδικό πλήκτρο (π.χ. το Shift), που, αν συνδυαστεί με τα πλήκτρα περιστροφής, προκαλεί πλάγια κίνηση, δηλαδή μετατόπιση προς τα αριστερά ή τα δεξιά. Για την κίνηση στον κάθετο άξονα, για παράδειγμα άλμα, πέταγμα ή σκύψιμο, μπορούν να χρησιμοποιηθούν δύο επιπλέον πλήκτρα.

Η παραπάνω τεχνική μπορεί να συνδυαστεί με τη χρήση του ποντικιού για μεγαλύτερη ακρίβεια στις περιστροφές και για αλλαγή του προσανατολισμού της κάμερας προς τα πάνω ή προς τα κάτω. Η συνδυασμένη τεχνική, που ονομάζεται και «κοίταγμα μέσω ποντικιού» (mouselook), μπορεί να περιγραφεί ως εξής: ο χρήστης ελέγχει την εστίαση της κάμερας μέσω της κίνησης του ποντικιού προκαλώντας αντίστοιχες περιστροφές στον πλάγιο (Y) και στον κάθετο (Z) άξονα. Στην εικόνα 4.2 φαίνεται ένα παράδειγμα περιστροφής στον κάθετο άξονα μέσω mouselook. Το πληκτρολόγιο χρησιμοποιείται μόνο για μετακίνηση

προς ή από τη διεύθυνση της κάμερα και για πλάγια κίνηση. Η συγκεκριμένη τεχνική προκαλεί μια σαφώς πιο πλούσια εμπειρία πλοήγησης λόγω του άμεσου ελέγχου της εστίασης, έχει όμως ένα βασικό πρόβλημα στην περίπτωση των εικονικών κόσμων: δεσμεύει το ποντίκι στην πλοήγηση, στερώντας έτσι από τον χρήστη τη δυνατότητα να κινήσει ελεύθερα τον δείκτη του ποντικιού για να αλληλεπιδράσει με αντικείμενα του περιβάλλοντος, όπως θα δούμε στη συνέχεια.



Εικόνα 4.2 Παράδειγμα περιστροφής στον κάθετο άξονα μέσω mouselook.

Ένα κρίσιμο σχεδιαστικό ζήτημα σχετικά με τη μετακίνηση της ενσάρκωσης ή της κάμερας έχει να κάνει με την ταχύτητα μετακίνησης. Κάθε πλήκτρο του πληκτρολογίου έχει δύο μόνο καταστάσεις, πατημένο ή μη, επομένως στην εντολή «κίνηση προς τα εμπρός» δεν προσδιορίζεται η ταχύτητα μετακίνησης. Αυτό πρακτικά σημαίνει ότι το σύστημα έχει προαποφασίσει κάποια ταχύτητα μετακίνησης και την αποδίδει στην οντότητα υπό έλεγχο. Όμως η επιθυμητή ταχύτητα μετακίνησης μπορεί να σχετίζεται με τη δραστηριότητα του χρήστη. Για παράδειγμα, η μετακίνηση σε μεγάλες αποστάσεις ευνοείται από μεγάλες ταχύτητες, ενώ οι προσεκτικοί χειρισμοί αποφυγής εμποδίων και γενικότερα κίνησης σε περίπλοκους χώρους από μικρότερες. Για να αντιμετωπίσουν αυτό το πρόβλημα οι περισσότεροι εικονικοί κόσμοι προσφέρουν δύο ταχύτητες μετακίνησης, μια πιο αργή και μια πιο γρήγορη, που στην περίπτωση κίνησης της ενσάρκωσης μεταφράζεται σε τρέξιμο.

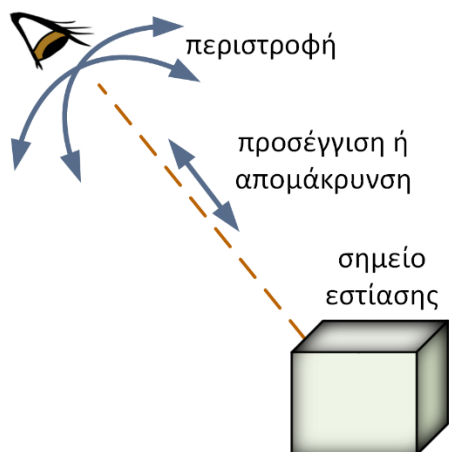
Πολλές φορές στους εικονικούς κόσμους προκύπτει η ανάγκη παρατήρησης ενός αντικειμένου ή χώρου από πολλαπλές απόψεις. Στις περιπτώσεις αυτές είναι προτιμότερη μια τεχνική «εξερεύνησης», η οποία βασίζεται στη μεταφορά ελέγχου σκηνής. Σύμφωνα με την τεχνική αυτήν ο χρήστης προσδιορίζει ένα σημείο εστίασης, που μπορεί να είναι ένα αντικείμενο της σκηνής ή ένα σημείο πάνω στο έδαφος, και με τη χρήση του ποντικιού του ελέγχει την περιστροφή της κάμερας γύρω από αυτό (εικόνα 4.3). Ο έλεγχος γίνεται με τρόπο που να δίνεται η αίσθηση στον χρήστη ότι περιστρέφει το ίδιο το αντικείμενο, με αποτέλεσμα να μπορεί να ελέγξει αρκετά διαισθητικά την κίνησή του σε σχέση με την όψη του αντικειμένου που θέλει να παρατηρήσει. Συνήθως η τεχνική αυτή συνδυάζεται και με τη δυνατότητα προσέγγισης ή απομάκρυνσης από το σημείο εστίασης (zoom in/out), ώστε να μπορεί ο χρήστης να δει περισσότερες λεπτομέρειες ή να έχει πιο σφαιρική εικόνα του χώρου αντίστοιχα. Η τεχνική αυτή είναι πολύ χρήσιμη σε περιπτώσεις εφαρμογών που προκύπτει η ανάγκη να κατανοήσει ο χρήστης καλύτερα την τρισδιάστατη μορφή των αντικειμένων, ή και σε σχεδιαστικές εφαρμογές που απαιτείται η εναλλαγή μεταξύ διάφορων απόψεων του χώρου για την καλύτερη τοποθέτηση των αντικειμένων και επισκόπηση του αποτελέσματος.

Πέρα από τις παραπάνω τεχνικές, πολλές φορές στους εικονικούς κόσμους προσφέρονται και άλλες επιλογές πλοήγησης, όπως η τηλεμεταφορά μέσω κατάδειξης ή χάρτη, ο αυτόματος έλεγχος της κάμερας και η επιβίβαση σε κινούμενο όχημα.

Η τηλεμεταφορά είναι χρήσιμη κυρίως σε αρχάριους χρήστες ή και σε περιπτώσεις που απαιτείται γρήγορη μετακίνηση σε απομακρυσμένες περιοχές. Συνήθως υλοποιείται με δύο τεχνικές: η μία είναι η κατάδειξη συγκεκριμένης περιοχής πάνω στην οθόνη (point and click), για παράδειγμα ένα απομακρυσμένο κτίριο, και η άλλη η κατάδειξη σημείου πάνω σε χάρτη κάτοψης του κόσμου (map-based). Το αποτέλεσμα της ενέργειας είναι ότι η ενσάρκωση, άρα και η οπτική του χρήστη, μεταφέρεται άμεσα στη νέα τοποθεσία.

Ο αυτόματος έλεγχος της κάμερας είναι μια χρήσιμη τεχνική που μπορεί να βελτιώσει την εμπειρία του χρήστη με το περιβάλλον σε ορισμένες ειδικές περιπτώσεις. Για παράδειγμα, μπορεί η κάμερα να αποκτήσει συγκεκριμένη οπτική γωνία για την καλύτερη αλληλεπίδραση του χρήστη με κάποιο αντικείμενο του κόσμου,

π.χ. για τον χειρισμό μιας περίπλοκης συσκευής. Στην προοπτική τρίτου προσώπου μπορεί η κάμερα να τοποθετείται καλύτερα για την αποφυγή ανεπιθύμητων επικαλύψεων, π.χ. αν ένας τοίχος παρεμβάλλεται μεταξύ της κάμερας και της ενσάρκωσης. Σε άλλες περιπτώσεις μπορεί να προσφέρεται στον χρήστη μια *αυτόματη ξενάγηση (fly-through)* ενός νέου χώρου, όπου το σύστημα αποκτά τον έλεγχο της κάμερας και τη μετακινεί ομαλά σε περιοχές ενδιαφέροντος.



Εικόνα 4.3 Η τεχνική «εξερεύνησης».

Η τεχνική της *επιβίβασης σε όχημα* είναι μια εναλλακτική προσέγγιση αυτόματης ξενάγησης, η οποία αφήνει ελεύθερο τον χρήστη να επιλέξει ο ίδιος τις περιοχές εστίασης ενώ η κάμερα κινείται. Η ενσάρκωση του χρήστη επιβιβάζεται σε κάποιο κινούμενο αντικείμενο, που συνήθως έχει τη μορφή κάποιου οχήματος, για παράδειγμα τρένο, αερόστατο κ.λπ., και ακολουθεί την κίνησή του.

2.2 Εύρεση πορείας

Η εύρεση πορείας είναι η γνωστική διεργασία που συντελείται κατά τη διάρκεια της πλοήγησης των χρηστών σε έναν εικονικό κόσμο για την προσέγγιση κάποιου στόχου. Περιλαμβάνει την επίγνωση της θέσης στην οποία βρίσκεται ο χρήστης, την κατασκευή ενός μονοπατιού προς τον στόχο και τη δημιουργία ενός νοητικού μοντέλου του περιβάλλοντος (Chen & Stanney, 1999). Απαραίτητη προϋπόθεση για την επιτυχημένη εύρεση πορείας είναι η *επίγνωση της κατάστασης (situation awareness)*, δηλαδή η ικανότητα των χρηστών να αντιληφθούν σωστά τα αντικείμενα και τη γεωμετρία του χώρου που τους περιβάλλει και να κατασκευάσουν την κατάλληλη νοητική απεικόνιση. Η αποτυχία στην εύρεση πορείας μπορεί να οδηγήσει τον χρήστη σε μια κατάσταση στην οποία νιώθει χαμένος και δεν έχει επίγνωση της θέσης στην οποία βρίσκεται και του τρόπου με τον οποίο μπορεί να προσεγγίσει τον στόχο του. Βεβαίως όλοι οι χρήστες είναι έμπειροι στην εύρεση πορείας στον φυσικό κόσμο και μπορούν να εκμεταλλευτούν αυτές τις ικανότητες στην πλοήγησή τους σε εικονικά περιβάλλοντα. Όμως τόσο ο μη φυσικός τρόπος μετακίνησης όσο και ο σημαντικά μικρότερος βαθμός λεπτομέρειας στην απεικόνιση του περιβάλλοντος, που μπορεί να κάνει αντικείμενα, κτίρια ή και περιοχές να μοιάζουν πανομοιότυπα, επιβαρύνουν τη νοητική διεργασία της εύρεσης μονοπατιού και κάποιες φορές δυσκολεύουν τους αρχάριους χρήστες.

Έχει βρεθεί ότι οι άνθρωποι ακολουθούμε διάφορες στρατηγικές για τη δημιουργία του νοητικού μοντέλου ενός νέου περιβάλλοντος. Μια προσέγγιση είναι η *διαίρεση σε υποπεριοχές*: μοιράζουμε τον χώρο σε επιμέρους τμήματα και προσπαθούμε να θυμηθούμε πώς αυτά σχετίζονται χωρικά μεταξύ τους. Μια άλλη προσέγγιση είναι η *χρήση οροσήμων (landmarks)*, δηλαδή κτιρίων, κατασκευών ή ακόμα και φυσικών διαφοροποιήσεων για την απομνημόνευση συγκεκριμένων σημείων του χώρου, και τελικά η *χαρτογράφηση* του περιβάλλοντος ως ένα δίκτυο από ορόσημα. Κάποιοι άνθρωποι κάνουν *προοδευτική χαρτογράφηση*, δηλαδή απομνημονεύουν μια μικρή αρχική περιοχή του περιβάλλοντος και προοδευτικά επεκτείνουν τα όρια του μοντέλου τους μεγάλωνοντας την περιοχή. Τέλος, μια όχι και τόσο συχνή προσέγγιση είναι και η *εξιστόρηση*: οι διάφορες περιοχές και τα ορόσημα ενός περιβάλλοντος συνδέονται μέσω ιστοριών που περιγράφουν προηγούμενες μετακινήσεις σε αυτά.

Στα τρισδιάστατα εικονικά περιβάλλοντα έχουν χρησιμοποιηθεί διάφορα εργαλεία και βοηθήματα για την υποστήριξη της εύρεσης πορείας. Το γεγονός ότι φαίνεται να υπάρχουν παραπάνω από μία τεχνικές για την κατασκευή του νοητικού μοντέλου του περιβάλλοντος έχει οδηγήσει στη σχεδίαση κατάλληλων βοηθημάτων για την υποστήριξη των τεχνικών αυτών. Γενικά θεωρείται προτιμότερη η ένταξη πολλαπλών βοηθημάτων στο περιβάλλον, ώστε να αξιοποιηθούν από όσο το δυνατόν μεγαλύτερο εύρος χρηστών. Οι πιο συχνοί τύποι εργαλείων και βοηθημάτων για την εύρεση πορείας είναι:

- *Χάρτες κάτοψης (minimaps)*: Απεικονίζεται στην οθόνη ένας μικρός χάρτης της περιοχής, πάνω στον οποίο απεικονίζεται η θέση του χρήστη, περίπου όπως λειτουργούν οι σημερινές εφαρμογές GPS. Οι χάρτες μπορεί να έχουν σταθερό προσανατολισμό (π.χ. ο Βορράς να βρίσκεται προς τα πάνω – north up) ή να περιστρέφονται προς τη διεύθυνση της κίνησης του χρήστη. Στους χάρτες κάτοψης μπορούν να επισημανθούν οι χώροι ενδιαφέροντος ώστε να τους εντοπίζει εύκολα ο χρήστης. Επιπλέον, μπορεί να υποστηρίζεται η τηλεμεταφορά του χρήστη σε απομακρυσμένες περιοχές με click πάνω στο αντίστοιχο σημείο.
- *Πυξίδες*: Δισδιάστατα ανάλογα της πραγματικής πυξίδας, που εμφανίζονται στην οθόνη του χρήστη και δείχνουν προς τον Βορρά για να τον βοηθήσουν στον προσανατολισμό του.
- *Ορόσημα*: Ιδιαίτερα κτίρια, μνημεία, βουνά τα οποία τοποθετούνται σε συγκεκριμένα σημεία του κόσμου για να βοηθήσουν τους χρήστες στην εύρεση πορείας. Εφόσον ο χρήστης εντοπίζει και θυμάται τα ορόσημα, μπορεί αρχικά να αντιληφθεί αν έχει ξαναπεράσει από κάποιο σημείο και στη συνέχεια να δημιουργήσει ένα νοητικό μοντέλο του χώρου βασισμένο σε αυτά. Σε κάποιες περιπτώσεις είναι δυνατόν να προστεθούν και ηχητικά ορόσημα, δηλαδή συγκεκριμένοι ήχοι περιβάλλοντος που μπορούν να βοηθήσουν να συνδέσουμε νοητικά μια περιοχή με αυτούς. Ακόμα και ο ήλιος, αν βρίσκεται σε χαμηλή θέση, μπορεί να λειτουργήσει ως ορόσημο και να βοηθήσει τους χρήστες να προσανατολιστούν με τρόπο ανάλογο με αυτόν μιας πυξίδας.
- *Ίχνη και πινακίδες*: Μπορούν να προστεθούν ενδείξεις στο περιβάλλον που δείχνουν στον χρήστη τη διεύθυνση ή και ολόκληρη τη διαδρομή προς έναν ή περισσότερους προορισμούς. Ο χρήστης εντοπίζει και ακολουθεί το σηματοδοτημένο μονοπάτι παρατηρώντας είτε ίχνη πάνω στο έδαφος είτε πινακίδες διεύθυνσης για να καταλήξει στον επιθυμητό προορισμό.
- *Ίχνη χρήστη*: Ο χρήστης αφήνει πίσω του ίχνη κατά τη μετακίνησή του. Ο εντοπισμός τους τον βοηθάει να αντιληφθεί ότι έχει ξαναπεράσει από αυτό το σημείο.
- *Κατάτμηση σε υποπεριοχές*: Ο χώρος διαιρείται σε υποπεριοχές με ευκολομνημόνευτα ονόματα, τις οποίες ο χρήστης μπορεί να θυμηθεί και να εντοπίσει στον χάρτη ή σε σχετικές πινακίδες.
- *Καθοδήγηση*: Εικονικά οχήματα ή εικονικοί ξεναγοί αναλαμβάνουν να μεταφέρουν τον χρήστη προς τον επιθυμητό προορισμό.

3 Αλληλεπίδραση με αντικείμενα

Υπάρχουν πολλών ειδών αλληλεπιδράσεις των χρηστών με τα αντικείμενα ενός εικονικού περιβάλλοντος. Σε ρεαλιστικούς κόσμους περιλαμβάνονται αντικείμενα που μιμούνται τη συμπεριφορά αντίστοιχων φυσικών αντικειμένων, και ως τέτοια θα πρέπει να υποστηρίζουν κατάλληλες αλληλεπιδράσεις. Μπορεί να περιλαμβάνονται απλά αντικείμενα τα οποία ο χρήστης έχει τη δυνατότητα να πιάσει και να μετακινήσει σε άλλη θέση, αντικείμενα με κινούμενα μέρη (π.χ. πόρτες, συρτάρια κ.λπ.) τα οποία ο χρήστης κινεί βάσει κάποιων περιορισμών, αντικείμενα–εργαλεία τα οποία θα πρέπει να χειριστεί για να εκτελέσει κάποιες ενέργειες στο περιβάλλον ή ακόμα και αντικείμενα με πιο περίπλοκη διεπαφή, όπως συσκευές, οχήματα κ.λπ. Σε εφαρμογές σχεδίασης, οι χρήστες θα πρέπει επιπλέον να μπορούν να εισάγουν, να διαγράφουν και να κλωνοποιούν αντικείμενα, να ορίζουν ή/και να τροποποιούν τη μορφή τους καθώς και να καθορίζουν πολλές άλλες παραμέτρους όπως το υλικό τους, τη φυσική τους συμπεριφορά κ.λπ. Τέλος, οι εικονικοί κόσμοι μπορούν να περιλαμβάνουν ακόμα και αφηρημένα αντικείμενα που χρησιμοποιούνται ως στοιχεία διεπαφής με στόχο την εκτέλεση κάποιων ενεργειών, την ανατροφοδότηση ή και την τροποποίηση παραμέτρων του συστήματος. Για παράδειγμα, δίπλα σε ένα μουσειακό έκθεμα μπορεί να υπάρχουν πλήκτρα για περισσότερες πληροφορίες, για βαθμολόγηση ή για σχολιασμό.

Η σχεδίαση των κατάλληλων αλληλεπιδράσεων με αντικείμενα σαν τα παραπάνω περιορίζεται και πάλι από τις δυνατότητες των συσκευών εισόδου. Ενέργειες όπως η ελεύθερη μετακίνηση ενός αντικειμένου έχουν έξι βαθμούς ελευθερίας, κάτι που προκαλεί δυσκολίες στην απεικόνιση της εισόδου ανάλογες με αυτές που αναφέραμε στην περίπτωση της πλοήγησης. Επιπρόσθετα, το πληκτρολόγιο είναι συνήθως δεσμευμένο από την πλοήγηση, επομένως οι ενέργειες αλληλεπίδρασης με το περιεχόμενο περιορίζονται στη χρήση του ποντικιού, αν δεν υπάρχει κάποια άλλη εξειδικευμένη συσκευή εισόδου. Οι ιδιαιτερότητες αυτές έχουν οδηγήσει στην ανάπτυξη ειδικών τεχνικών ανάλογα με το είδος της δραστηριότητας (π.χ. μετακίνηση ή περιστροφή) και τους απαιτούμενους βαθμούς ελευθερίας, οι οποίες μεταφράζουν την είσοδο των σχετικών συσκευών εισόδου σε ενέργειες στο τρισδιάστατο περιβάλλον.

Η διαδικασία της αλληλεπίδρασης ενός χρήστη με ένα αντικείμενο σε ένα τρισδιάστατο περιβάλλον μοιράζεται σε τρεις φάσεις: επιλογή, χειρισμός και απελευθέρωση (Mine κ.ά., 1997). Αρχικά ο χρήστης θα πρέπει να καταδείξει στο σύστημα το αντικείμενο με το οποίο θέλει να αλληλεπιδράσει, στη συνέχεια να το χειριστεί πραγματοποιώντας ενέργειες που επιδρούν στις γεωμετρικές ή άλλες ιδιότητές του και τέλος να το «απελευθερώσει», δηλαδή να σταματήσει να αλληλεπιδρά μαζί του και να αποκτήσει το αντικείμενο την τελική του θέση και μορφή στον κόσμο.

3.1 Επιλογή αντικειμένου

Η επιλογή ενός αντικειμένου της σκηνής είναι μια διαδικασία η οποία σε περιβάλλοντα επιφάνειας εργασίας γίνεται με άμεσο και διαισθητικό τρόπο με τη χρήση του ποντικιού. Ακολουθώντας την τεχνική της κατάδειξης και επιλογής (point and click) που χρησιμοποιείται στα δισδιάστατα περιβάλλοντα, οι χρήστες αρκεί να φέρουν τον δείκτη του ποντικιού πάνω από το αντικείμενο με το οποίο επιθυμούν να αλληλεπιδράσουν και να το επιλέξουν με click του ποντικιού πάνω του. Η τεχνική αυτή δεν είναι βεβαίως συμβατή με την πλοήγηση με τη χρήση πληκτρολογίου και κοιτάγματος μέσω ποντικιού (mouselook), διότι η κίνηση του ποντικιού επηρεάζει την περιστροφή της κάμερας, κάνοντας την επιλογή του αντικειμένου σημαντικά δυσκολότερη.

Ένα πιθανό πρόβλημα που μπορεί να προκύψει από τη χρήση αυτής της τεχνικής σχετίζεται με την απόσταση από την οποία επιλέγει να αλληλεπιδράσει ένας χρήστης με το αντικείμενο. Αν ο χειρισμός που θα γίνει στο αντικείμενο περιλαμβάνει τον άμεσο έλεγχο της θέσης του μέσω της κίνησης του ποντικιού, τότε, όσο μεγαλύτερη είναι η απόσταση του αντικειμένου από τη θέση της κάμερας, τόσο μεγαλύτερη είναι και η κλίμακα μετακίνησης. Έτσι, η μετακίνηση του ποντικιού κατά μερικά pixels προς μία διεύθυνση μπορεί να μεταφραστεί ως μετακίνηση κατά μερικά δέκατα του μέτρου σε κοντινά αντικείμενα και κατά μερικά μέτρα σε μακρινά. Συνεπώς, η ακρίβεια τοποθέτησης στη δεύτερη περίπτωση θα είναι σαφώς μικρότερη. Επιπλέον, αν η ενσάρκωση ενός χρήστη έχει πολύ μεγάλη απόσταση από το αντικείμενο το οποίο χειρίζεται, θα είναι δύσκολο για τους υπόλοιπους παρατηρητές να αντιληφθούν ποιος χειρίζεται το αντικείμενο, κάτι που θα μειώσει την επίγνωσή τους σχετικά με το περιβάλλον. Πιθανοί τρόποι αντιμετώπισης των παραπάνω θεμάτων είναι να τοποθετείται αυτόματα η κάμερα κοντά στο αντικείμενο που πρόκειται να χειριστεί ο χρήστης ή και να μετακινείται η ενσάρκωση στο σημείο αυτό. Μια άλλη προσέγγιση που περιορίζει το ζήτημα της επίγνωσης είναι η εμφάνιση νοητών ακτίνων που συνδέουν την ενσάρκωση με το αντικείμενο που χειρίζεται.

Σε περιπτώσεις κόσμων που απαιτείται συχνή χρήση αντικειμένων που βρίσκονται σε απομακρυσμένες περιοχές ή δεν είναι εύκολα προσεγγίσιμα, μια συμπληρωματική τεχνική μπορεί να είναι η επιλογή μέσω κάποιου μενού ή λίστας.

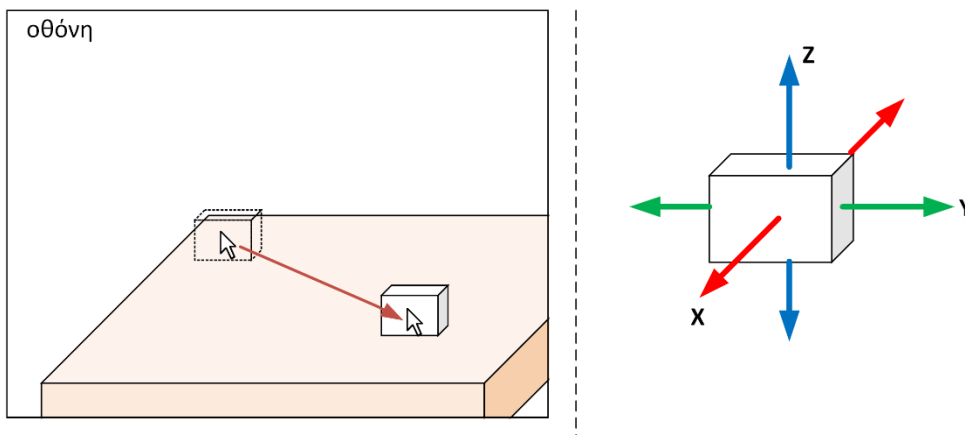
3.2 Χειρισμός και απελευθέρωση

Ο χειρισμός των αντικειμένων ενός τρισδιάστατου περιβάλλοντος μπορεί να γίνει με τους παρακάτω τρόπους (Sherman & Craig, 2003):

- *φυσική κίνηση*: ο χρήστης εκτελεί με το σώμα του τη φυσική κίνηση χειρισμού του αντικειμένου,
- *άμεσος έλεγχος μέσω συσκευής*: ο χρήστης χειρίζεται μια φυσική συσκευή και οι κινήσεις του μεταφράζονται σε αντίστοιχες δράσεις πάνω στο αντικείμενο,
- *εικονικός έλεγχος*: ο χρήστης χρησιμοποιεί εικονικά στοιχεία ελέγχου που εμφανίζονται στο περιβάλλον αλληλεπίδρασης και

- *έλεγχος μέσω πράκτορα*: ο χρήστης δίνει εντολές, συνήθως λεκτικές, μέσω φωνής ή κειμένου σε άλλες οντότητες ή στο σύστημα, και οι ενέργειες πραγματοποιούνται αυτόματα.

Από τις παραπάνω μορφές χειρισμού, ο άμεσος έλεγχος μέσω συσκευής και ο εικονικός έλεγχος είναι οι πιο συχνά χρησιμοποιούμενες στους εικονικούς κόσμους. Οι φυσικές κινήσεις απαιτούν εξειδικευμένο υλικό εισόδου, όπως κάποιες από τις συσκευές που θα εξετάσουμε στη συνέχεια. Ο έλεγχος μέσω πράκτορα, αν και θα διευκόλυνε σημαντικά την εκτέλεση σύνθετων χειρισμών, είναι δύσκολος στην υλοποίηση. Θα πρέπει να βασιστεί σε μηχανισμό επεξεργασίας φυσικής γλώσσας, ενδεχομένως και φωνητικής αναγνώρισης, και θα πρέπει να μπορεί να μεταφράζει με εύλογο τρόπο πιθανά αμφίσημες εντολές του χρήστη, ώστε να εκτελεί τις ενέργειες με επιτυχία. Ο άμεσος έλεγχος στα περιβάλλοντα επιφάνειας εργασίας είναι, ουσιαστικά, η μετάφραση της κίνησης του ποντικιού σε αντίστοιχες μετατοπίσεις και περιστροφές του αντικειμένου, ενώ με τον εικονικό έλεγχο θα πρέπει ο χρήστης να επιλέξει πρώτα κάποιο εικονικό στοιχείο ελέγχου και στη συνέχεια, μέσω αυτού, να προσδιορίσει τον τύπο της κίνησης.

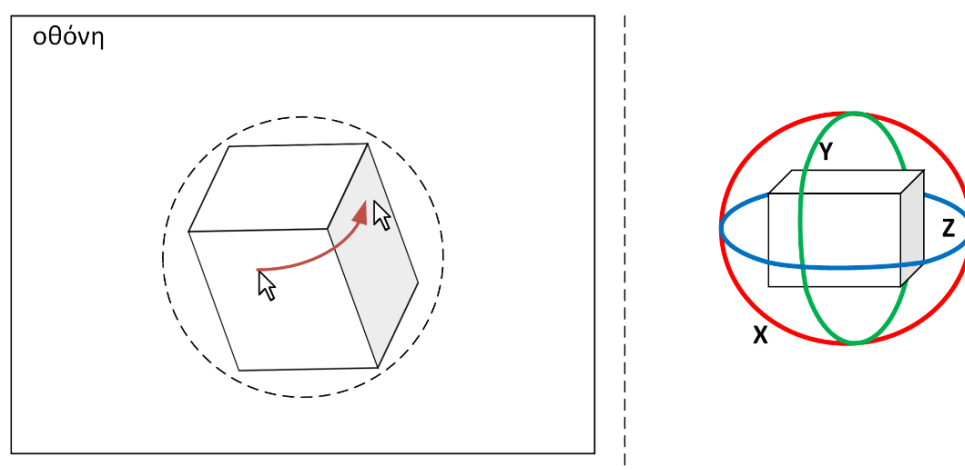


Εικόνα 4.4 Μετατόπιση αντικειμένου άμεσα μέσω ποντικιού (αριστερά) και με εικονικά χειριστήρια (δεξιά).

Στην περίπτωση της μετατόπισης, ο έλεγχος μέσω ποντικιού στερείται μίας διάστασης. Αυτό σημαίνει ότι θα πρέπει να μεταφραστεί η κίνησή του σε αντίστοιχη κίνηση του τρισδιάστατου αντικειμένου σε δύο διαστάσεις. Κατά συνέπεια, ενώ είναι εύλογο η κίνηση του ποντικιού προς τα αριστερά ή προς τα δεξιά να μεταφραστεί σε αντίστοιχη κίνηση στο τρισδιάστατο περιβάλλον, δεν είναι σαφές για τον χρήστη αν η μετακίνηση του ποντικιού προς τα πάνω ή προς τα κάτω μεταφράζεται σε κίνηση σε ύψος ή σε βάθος. Θα πρέπει λοιπόν να είναι ξεκάθαρο το επίπεδο της κίνησης, είτε λόγω περιορισμών του ίδιου του αντικειμένου (π.χ. τοποθέτηση πάνω σε κάποια άλλη επιφάνεια), είτε να υποστηρίζεται από την ίδια τη διεπαφή του περιβάλλοντος η επιλογή μεταξύ επιπέδων μετατόπισης, για παράδειγμα με τη χρήση ενός επιπλέον πλήκτρου. Ακόμα όμως κι αν είναι ξεκάθαρο το επίπεδο μετατόπισης, ένα επιπλέον ερώτημα είναι σε ποιο σύστημα συντεταγμένων θα μεταφερθεί η κίνηση: στο σύστημα συντεταγμένων κόσμου, στο τοπικό σύστημα συντεταγμένων του αντικειμένου ή στο σύστημα συντεταγμένων της κάμερας; Η πιο φυσική/διαισθητική τεχνική είναι να γίνει αντιστοίχιση στο σύστημα συντεταγμένων της κάμερας με τρόπο που η τελική θέση του αντικειμένου θα ακολουθεί την αντίστοιχη θέση του δείκτη του ποντικιού. Στην εικόνα 4.4 (αριστερά) φαίνεται ένα παράδειγμα άμεσης μετατόπισης πάνω σε επιφάνεια στο σύστημα συντεταγμένων της κάμερας.

Οι δυσκολίες προσδιορισμού της μετατόπισης σε τρεις διαστάσεις άμεσα μέσω του ποντικιού έχουν οδηγήσει πολλές εφαρμογές στη χρήση εικονικών στοιχείων ελέγχου. Σύμφωνα με αυτήν την προσέγγιση, εμφανίζονται βέλη που δείχνουν προς τη διεύθυνση των τριών αξόνων, στο τοπικό σύστημα συντεταγμένων του αντικειμένου ή στο σφαιρικό. Τα βέλη είναι κατάλληλα χρωματισμένα ώστε να μπορεί ο χρήστης εύκολα να αναγνωρίσει τους άξονες: οι άξονες X, Y και Z χρωματίζονται με τα αντίστοιχα χρώματα R, G και B (δηλαδή κόκκινος, πράσινος και μπλε — red, green και blue αντίστοιχα) του σχετικού μοντέλου (βλ. εικ. 4.4 δεξιά). Ο χρήστης μετακινεί το αντικείμενο με επιλογή (κλικ) ενός βέλους και η κίνηση του ποντικιού μεταφράζεται σε κίνηση του αντικειμένου στον αντίστοιχο άξονα. Αυτή η τεχνική, αν και καθυστερεί σημαντικά την τοποθέτηση σε σχέση με τον άμεσο χειρισμό, προσφέρει σημαντικά μεγαλύτερη ακρίβεια στη μετακίνηση και βοηθάει στον περιορισμό των λαθών. Αν υπάρχει ένα μειονέκτημα στην τεχνική αυτήν, είναι ότι η εμφάνιση και χρήση των εικονικών στοιχείων ελέγχου περιορίζει τον ρεαλισμό της απεικόνισης και τη φυσικότητα της αλληλεπίδρασης.

Στην περίπτωση της περιστροφής ενός αντικειμένου υπάρχει και πάλι απώλεια ενός βαθμού ελευθερίας με τη χρήση του ποντικιού, και προσφέρονται τεχνικές τόσο άμεσου χειρισμού όσο και εικονικών χειριστηρίων. Μια δημοφιλής τεχνική άμεσης περιστροφής ενός αντικειμένου με τη χρήση του ποντικιού είναι η τεχνική της «εικονικής σφαίρας» (virtual sphere). Σύμφωνα με αυτήν, δημιουργείται μια νοητή σφαίρα γύρω από το αντικείμενο και απεικονίζονται τα όριά της ως κύκλος. Αν ο χρήστης κάνει κλικ με το ποντίκι εντός της σφαίρας και σύρει προς μια διεύθυνση, τότε το αντικείμενο περιστρέφεται προς τους αντίστοιχους άξονες ακολουθώντας την κίνηση του ποντικιού (εικ. 4.5 αριστερά). Η οριζόντια κίνηση του ποντικιού το περιστρέφει γύρω από τον άξονα των Z και η κάθετη γύρω από τον Y. Για τον τρίτο άξονα περιστροφής, αρκεί ο χρήστης να κάνει κλικ εκτός της νοητής σφαίρας και να σύρει το ποντίκι κατά μήκος της περιφέρειας του κύκλου, προς τα αριστερά ή προς τα δεξιά. Η μεταφορά είναι αρκετά φυσική και δίνει στον χρήστη την αίσθηση ότι έχει άμεσο έλεγχο του αντικειμένου. Εναλλακτικά, σε πολλά συστήματα χρησιμοποιούνται εικονικά στοιχεία ελέγχου. Εμφανίζονται τρεις δίσκοι για την περιστροφή στους ισάριθμους άξονες, χρωματισμένοι με τρόπο ανάλογο με αυτόν της μετατόπισης (εικ. 4.5 δεξιά). Ο χρήστης μπορεί να επιλέξει έναν από αυτούς και να σύρει το ποντίκι κατά μήκος της περιφέρειάς του προκαλώντας περιστροφή του αντικειμένου στον αντίστοιχο άξονα.



Εικόνα 4.5 Περιστροφή αντικειμένου με την τεχνική της εικονικής σφαίρας (αριστερά) και με εικονικά χειριστήρια (δεξιά).

Ανεξαρτήτως της τεχνικής που θα χρησιμοποιηθεί, ένα κρίσιμο στοιχείο για την επιτυχία της αλληλεπίδρασης είναι η παροχή κατάλληλης ανάδρασης στον χρήστη. Είναι σημαντικό να δίνονται στον χρήστη οι απαραίτητες οπτικές ενδείξεις μέσω γραφικών απεικονίσεων, κειμένου ή/και αλλαγής του δείκτη του ποντικιού, ώστε να γνωρίζει ποιο είναι το αντικείμενο το οποίο χειρίζεται και πώς μεταφράζονται οι κινήσεις του ποντικιού σε αντίστοιχες μετακινήσεις και περιστροφές στον τρισδιάστατο χώρο. Η χρήση μεταφορών κατά την οπτική ανάδραση μπορεί να βοηθήσει στην ευκολότερη κατανόηση της αλληλεπίδρασης.

Τέλος, κατά την απελευθέρωση του αντικειμένου, δηλαδή όταν ο χρήστης ολοκληρώσει τη μετακίνηση ή την περιστροφή του, το σύστημα μπορεί να χρειαστεί να κάνει κάποιες τελικές «διορθώσεις», ανάλογα πάντα και με τις απαιτήσεις της εφαρμογής. Για παράδειγμα, αν ένα αντικείμενο πρέπει να τοποθετηθεί υποχρεωτικά πάνω σε κάποια επιφάνεια (π.χ. τα έπιπλα να βρίσκονται στο δάπεδο), το σύστημα μπορεί να διορθώσει την τελική του θέση ώστε να διασφαλίσει ότι είναι σωστά τοποθετημένο. Στη γενική περίπτωση, το σύστημα μπορεί κατά την απελευθέρωση να ελέγξει κατά πόσο η τελική θέση παραβιάζει κάποιον από τους φυσικούς ή άλλους περιορισμούς της εφαρμογής και να διορθώσει αυτόματα την τελική θέση του αντικειμένου, ή να μην αφήσει τον χρήστη να απελευθερώσει το αντικείμενο στη συγκεκριμένη θέση παρέχοντας κατάλληλη οπτική ανάδραση.

3.3 Άλλες αλληλεπιδράσεις

Πέρα από τη μετακίνηση και την περιστροφή σε τρεις διαστάσεις, υπάρχει και ένας αριθμός άλλων αλληλεπιδράσεων με αντικείμενα που υποστηρίζονται στους περισσότερους εικονικούς κόσμους.

Μια συχνή αλληλεπίδραση είναι η «ενεργοποίηση» ενός αντικειμένου από τον χρήστη. Η πιο απλή περίπτωση είναι το απλό κλικ του ποντικιού, μεταφορά αντίστοιχη με το πάτημα ενός πλήκτρου στον φυσικό

κόσμο ή στις δισδιάστατες διεπαφές. Η ενεργοποίηση μπορεί να πυροδοτήσει στη συνέχεια κάποιου είδους δράση από τη μεριά του αντικειμένου, ανάλογα με το είδος της εφαρμογής, για παράδειγμα, το πάτημα ενός διακόπτη μπορεί να προκαλεί το άναμμα της λάμπας. Σε κάποιες περιπτώσεις, τα αντικείμενα ενός εικονικού κόσμου ενεργοποιούνται και από άλλες ενέργειες του χρήστη, όπως το άγγιγμα (δηλαδή η επαφή της ενσάρκωσης με το αντικείμενο), η προσέγγιση από κάποια κοντινή απόσταση, το κοίταγμα (δηλαδή το να βρεθεί το αντικείμενο στο οπτικό πεδίο της κάμερας ή της ενσάρκωσης), οι λεκτικές εντολές μέσω φωνής ή κειμένου, η χρήση κάποιου δισδιάστατου παράθυρου διαλόγου κ.ά.

Η μετατόπιση και η περιστροφή ενός αντικειμένου μπορεί συχνά να γίνονται και σε λιγότερες διαστάσεις, λόγω περιορισμών της εφαρμογής. Υπάρχουν περιπτώσεις στις οποίες η μετακίνηση γίνεται μόνο πάνω σε ένα επίπεδο, π.χ. στην τοποθέτηση ενός επίπλου σε ένα δωμάτιο ή στη μετακίνηση ενός πιονιού στη σκακιέρα, ή ακόμα και σε έναν άξονα, π.χ. άνοιγμα και κλείσιμο ενός συρταριού ή μιας συρόμενης πόρτας. Αντίστοιχα, μπορεί να υπάρχουν και περιστροφές που γίνονται σε δύο άξονες ή και σε έναν. Για παράδειγμα, για την περιστροφή ενός προβολέα απαιτούνται συνήθως δύο άξονες, ενώ για το άνοιγμα μιας πόρτας ή για την περιστροφή ενός επίπλου τοποθετημένου στο δάπεδο αρκεί ένας. Σε αυτές τις περιπτώσεις, οι λιγότεροι βαθμοί ελευθερίας ευνοούν τον απευθείας χειρισμό μέσω ποντικιού.

Σε κάποιες λιγότερο συχνές περιπτώσεις μπορεί οι χρήστες να αλληλεπιδράσουν ασκώντας δύναμη πάνω στα αντικείμενα. Για παράδειγμα, μπορεί να ασκηθεί δύναμη για να προσδιοριστεί μια αρχική ταχύτητα και διεύθυνση ενός αντικειμένου, το οποίο στη συνέχεια θα κινηθεί ελεύθερα στο περιβάλλον υπακούοντας στους νόμους της φυσικής προσομοίωσης. Χρησιμοποιώντας μια μεταφορά εμπνευσμένη από την εκτόξευση αντικειμένων στον φυσικό κόσμο, ο χρήστης μπορεί να προσδιορίσει τη διεύθυνση και την ταχύτητα της κίνησης μέσω του ποντικιού ως εξής: κρατώντας πατημένο το πλήκτρο κινεί το ποντίκι προς κάποια διεύθυνση και στη συνέχεια απελευθερώνει το πλήκτρο. Η τελική ταχύτητα και διεύθυνση της κίνησής του μεταδίδονται στο αντικείμενο. Προφανώς τέτοιου είδους αλληλεπιδράσεις έχουν περισσότερο νόημα σε παιχνιδώδη περιβάλλοντα ή γενικότερα σε εφαρμογές με έντονο το στοιχείο της φυσικής προσομοίωσης. Ένα δεύτερο παράδειγμα αλληλεπίδρασης μέσω άσκησης δύναμης είναι η παραμόρφωση της επιφάνειας του αντικειμένου για την τροποποίηση της μορφής του. Μπορεί δηλαδή να γίνει κάποιου είδους «εικονική γλυπτική» ασκώντας δυνάμεις μεταβλητού μεγέθους σε διάφορα σημεία της επιφάνειας ενός αντικειμένου και οπτικά να προκαλείται αντίστοιχη παραμόρφωση. Συνήθως όμως αυτές οι τεχνικές είναι περισσότερο επιτυχημένες όταν γίνονται με τη χρήση κάποιου εξειδικευμένου υλικού φυσικής αλληλεπίδρασης με το περιβάλλον και όχι με την «παραδοσιακή» είσοδο των διεπαφών επιφάνειας εργασίας (βλ. Κεφ. 2).

Τέλος, υπάρχουν και περιπτώσεις στις οποίες ο χρήστης χρειάζεται να τροποποιήσει άλλες ιδιότητες των αντικειμένων, όπως το χρώμα και την υφή, να δημιουργήσει αντίγραφα ή να εισάγει νέα αντικείμενα στη σκηνή. Συνήθως οι αλληλεπιδράσεις αυτές υποστηρίζονται με τη χρήση διαλόγων, μενού και άλλων δισδιάστατων στοιχείων ελέγχου, έχοντας διεπαφή αντίστοιχη με αυτήν των δισδιάστατων παραθυρικών εφαρμογών.

3.4 Περιορισμοί

Ένα σημαντικό σχεδιαστικό ζήτημα σε σχέση με την αλληλεπίδραση και ιδιαίτερα την τοποθέτηση αντικειμένων είναι η χρήση περιορισμών. Η εισαγωγή περιορισμών αναμένεται να βοηθήσει στην καλύτερη διάταξη των αντικειμένων και στην αποφυγή λαθών. Για παράδειγμα, μπορεί να χρησιμοποιηθεί πλέγμα (grid), το οποίο εξαναγκάζει το αντικείμενο να μετακινηθεί σε συγκεκριμένες θέσεις, όπως συμβαίνει στα περισσότερα σχεδιαστικά προγράμματα. Επιπλέον, μπορεί να υπάρχουν περιορισμοί ως προς τις επιφάνειες πάνω στις οποίες μπορεί να τοποθετηθεί ένα αντικείμενο, ή ακόμα και προσαρμογή στους φυσικούς νόμους, όπως αποφυγή συγκρούσεων και βαρύτητα. Από την άλλη μεριά, ο μεγάλος αριθμός περιορισμών μπορεί να δυσκολένει δραστηριότητες στις οποίες απαιτούνται συχνές αλληλεπιδράσεις με τα αντικείμενα του περιβάλλοντος. Για παράδειγμα, μια σχεδιαστική εφαρμογή μπορεί να περιλαμβάνει πολλαπλές μετακινήσεις και τοποθετήσεις αντικειμένων μέχρι να καταλήξουν οι χρήστες σε μια τελική πρόταση. Η άρση των περιορισμών κατά τη διάρκεια αυτών των ενδιάμεσων ενεργειών θα διευκολύνει και εντέλει θα επιταχύνει τη διαδικασία, αφού, σε αντίθεση με τον φυσικό κόσμο, θα μπορούν, για παράδειγμα, να αφήσουν κάποια αντικείμενα στον αέρα, να τα περάσουν μέσα από τοίχους κ.λπ. Είναι λοιπόν στην ευθύνη του σχεδιαστή να καταλήξει σε μια καλή ισορροπία μεταξύ περιορισμών και ελευθερίας δράσης, ανάλογα με τη στόχευση της εφαρμογής και τις δραστηριότητες που αναμένεται να εκτελούν οι χρήστες σε αυτή.

4 Αλληλεπίδραση με άλλους χρήστες

Η ταυτόχρονη παρουσία πολλαπλών χρηστών ενισχύει σημαντικά την εμπειρία στους εικονικούς κόσμους. Ένα βασικό στοιχείο που συμβάλλει προς αυτή την κατεύθυνση είναι η δυνατότητα που παρέχεται σε ομάδες χρηστών να έχουν διαμοιρασμένες εμπειρίες (shared experience) χρήσης του συνθετικού περιβάλλοντος. Μπορούν να παρατηρούν, να εξερευνούν και να αλληλεπιδρούν από κοινού με το περιβάλλον, κάτι που τους επιτρέπει να ανταλλάσσουν ιδέες και απόψεις και να «χτίζουν» σταδιακά και από κοινού την κατανόηση και τη γνώση τους για το περιβάλλον και τα τεκταινόμενα εντός αυτού. Επιπλέον, η δυνατότητα σύγχρονης και ασύγχρονης απομακρυσμένης επικοινωνίας μεταξύ των χρηστών με τη χρήση των κατάλληλων εργαλείων τούς επιτρέπει να επικοινωνούν εντός ενός κοινού κόσμου στον οποίο μετέχουν ως ενσώματες οντότητες, ενισχύοντας την αίσθηση της συν-παρουσίας και θέτοντας τις βάσεις για την ανάπτυξη εικονικών κοινοτήτων. Τα παραπάνω χαρακτηριστικά αποτελούν ισχυρή πρώτη ύλη για τη διαμόρφωση κατάλληλων πλαισίων συνεργασίας ή και ανταγωνισμού εντός του εικονικού κόσμου. Οι αλληλεπιδράσεις μεταξύ χρηστών στους εικονικούς κόσμους σχετίζονται με τη διαμοίραση εμπειριών, τη σύγχρονη και ασύγχρονη επικοινωνία, την από κοινού χρήση αντικειμένων και τη δημιουργία και διαχείριση ομάδων.

4.1 Διαμοίραση εμπειριών

Ένα διαμοιρασμένο περιβάλλον δεν είναι απαραίτητα πολυχρηστικό. Η διαμοίραση εμπειριών έχει να κάνει με την από κοινού αντίληψη του περιβάλλοντος και της εξέλιξής του, χωρίς να προϋποθέτει την παρουσία και αλληλεπίδραση όλων των χρηστών σε αυτό. Σε παραδοσιακά συστήματα εικονικής πραγματικότητας, όπου ο αριθμός των εμπυθισμένων χρηστών είναι πολύ περιορισμένος λόγω κόστους, η εμπειρία διαμοιράζεται μέσω προβολής σε οθόνες του οπτικού πεδίου ενός χρήστη, ώστε να μπορούν οι εξωτερικοί παρατηρητές να αντιληφθούν τις ενέργειές του μέσα στο περιβάλλον. Σε μεγάλα συστήματα προβολής, για παράδειγμα σε τοίχο ή σε θόλο, η εμπειρία διαμοιράζεται σε όλους τους θεατές, αλλά τον έλεγχο της πλοήγησης έχει ένας από όλους, ή σε κάποιες πιο σπάνιες περιπτώσεις όλοι μαζί από κοινού. Στους εικονικούς κόσμους, που η πολυχρηστικότητα είναι εγγενές χαρακτηριστικό, η συν-παρουσία των χρηστών εγγυάται τη διαμοίραση της εμπειρίας. Αν δύο ή περισσότεροι χρήστες βρεθούν στην ίδια περιοχή του περιβάλλοντος την ίδια χρονική στιγμή, μπορούν να έχουν διαμοιρασμένη εμπειρία αλληλεπίδρασης με αυτό, ο καθένας από τη δική του οπτική. Σε ειδικές περιπτώσεις εφαρμογών μπορεί να απαιτείται διαμοίραση της άποψης ενός χρήστη, αν, για παράδειγμα, θέλουν οι υπόλοιποι χρήστες να παρατηρήσουν τις ενέργειες που εκτελεί μέσα από τη δική του οπτική γωνία. Μια τέτοια δυνατότητα μπορεί να προσφέρεται μέσω ειδικής επιλογής στους εικονικούς κόσμους. Σε άλλες περιπτώσεις, μπορεί να είναι επιθυμητή η διαμοίραση εμπειρίας σε μεταγενέστερο χρόνο. Αυτό μπορεί να επιτευχθεί με καταγραφή των ενεργειών του χρήστη και αναπαραγωγή, είτε σε μορφή βίντεο είτε με την προσθήκη μιας «σκιάδους» ενσάρκωσης (ghost avatar) που εκτελεί τις καταγεγραμμένες ενέργειες στο περιβάλλον.

4.2 Επικοινωνία

Τα προσφερόμενα μέσα για την επικοινωνία μεταξύ των χρηστών σε ένα εικονικό περιβάλλον είναι πολλαπλά. Αναμφισβήτητο ο πιο φυσικός τρόπος επικοινωνίας είναι η φωνή. Η επικοινωνία μέσω φωνής υποστηρίζεται στους περισσότερους σύγχρονους εικονικούς κόσμους με τη χρήση κάποιου μικροφώνου και συνήθως αποδίδεται «χωροθετημένη» στο περιβάλλον, με αποτέλεσμα να μπορεί να ακουστεί μόνο από κοντινούς χρήστες. Η χρήση της φωνής στους εικονικούς κόσμους απαιτεί σημαντικό επιπλέον εύρος δικτύου (bandwidth) για την καλής ποιότητας μετάδοσή της, κάτι που πρακτικά αποκλείει χρήστες με πιο αργές συνδέσεις. Η μετάδοση της φωνής ενεργοποιείται συνήθως με κάποιο πλήκτρο, αφενός γιατί η συνεχής καταγραφή επιβαρύνει ακόμα περισσότερο τις δικτυακές επικοινωνίες και αφετέρου διότι μπορεί ο χρήστης να μην επιθυμεί να μεταδίδονται όλοι οι ήχοι και οι συνομιλίες του περιβάλλοντος χώρου του. Όταν κάποιος χρήστης ομιλεί, συνήθως παρουσιάζεται και κάποια σχετική οπτικοποίηση στην ενσάρκωσή του, ώστε να είναι ξεκάθαρο στους υπόλοιπους χρήστες ποιος ήταν αυτός που μίλησε.

Εναλλακτικά ή συμπληρωματικά της φωνής, στους εικονικούς κόσμους προσφέρεται και η δυνατότητα επικοινωνίας μέσω κειμένου, κάτι που πριν από μερικά χρόνια ήταν ο κανόνας. Η συνομιλία μέσω γραπτού κειμένου, αν και λιγότερο φυσική, σε κάποιες περιπτώσεις πλεονεκτεί της φωνής. Όταν συνομιλούν πολλοί

χρήστες ταυτόχρονα, μπορεί κάποιος να παρακολουθήσει καλύτερα τη συζήτηση όπως αποτυπώνεται στο γραπτό κείμενο. Αντίθετα, στην φωνητική επικοινωνία μπορεί να χαθεί κάποια κρίσιμη πληροφορία. Το γραπτό κείμενο επιτρέπει τη διαμοίραση στοιχείων που δεν μπορούν ειπωθούν εύκολα προφορικά, όπως για παράδειγμα υπερσυνδέσμων, ενώ παρέχεται και η δυνατότητα καταγραφής και περαιτέρω επεξεργασίας του ιστορικού της συζήτησης, κάτι που ενδέχεται να είναι χρήσιμο σε συναντήσεις εργασίας, εικονικούς χώρους διαλέξεων κ.λπ.

Σε πιο σπάνιες περιπτώσεις, οι χρήστες μπορούν να επικοινωνήσουν και μέσω βιντεοπροβολής. Ένας φυσικός χώρος στον οποίο υπάρχουν ένας ή περισσότεροι ομιλητές μπορεί να καταγράφεται και η βιντεοσκοπήση να μεταδίδεται ζωντανά σε έναν εικονικό κόσμο, για παράδειγμα πάνω σε μια επιφάνεια που παίζει τον ρόλο «οθόνης προβολής» (βλ. εικ. 4.6) Η επικοινωνία αυτής της μορφής μπορεί να είναι και αμφίδρομη, αν οι συμμετέχοντες στον φυσικό χώρο μπορούν να βλέπουν το εικονικό περιβάλλον και να ακούνε τους ήχους τους στην περιοχή που προβάλλεται η βιντεοσκοπήση. Αυτής της μορφής η επικοινωνία δημιουργεί ένα «μεικτό» περιβάλλον, με την έννοια ότι εντός του εικονικού κόσμου εμφανίζεται και ένα παράθυρο στον φυσικό κόσμο, και το μεγάλο πλεονέκτημα χρήσης της είναι ότι, εκτός από τη φωνή των ομιλητών, εντός του φυσικού χώρου μεταδίδονται και περισσότερες ενδείξεις μη-λεκτικής επικοινωνίας, όπως οι εκφράσεις και οι χειρονομίες τους. Όμως, απαιτείται και στην περίπτωση αυτήν αρκετά ισχυρή σύνδεση δικτύου, ενώ μπορεί να προκύψουν και χρονικές καθυστερήσεις (lags) στη μετάδοση, οι οποίες θα δυσκολέψουν την αμφίδρομη επικοινωνία.



Εικόνα 4.6 Επικοινωνία μέσω ζωντανής βιντεοπροβολής στο *Second Life* –*Virtual University of Edinburgh*.

Μια άλλη μορφή οπτική επικοινωνία είναι και η εισαγωγή αντικειμένων στο περιβάλλον που φέρουν κάποιο μήνυμα για τους υπόλοιπους χρήστες. Σήματα κατεύθυνσης προς κάποιο προορισμό, μονοπάτια, πληροφορίες, σχόλια ή άλλου τύπου μηνύματα που οπτικοποιούνται ως αντικείμενα ή ως απεικονίσεις πάνω σε επιφάνειες αντικειμένων αποτελούν παραδείγματα τέτοιων στοιχείων επικοινωνίας.

Τέλος, η ίδια η παρουσία και δραστηριοποίηση των χρηστών εντός του περιβάλλοντος αποτελεί κανάλι επικοινωνίας. Η όψη των ενσαρκώσεων, η θέση που καταλαμβάνουν στο περιβάλλον στα πλαίσια μιας διεργασίας, η περιοχή εστίασής τους, η κατάδειξη αντικειμένων, οι κινήσεις και οι χειρονομίες που εκτελούν, είναι όλα μορφές μη-λεκτικής επικοινωνίας που μπορούν να μεταδώσουν χρήσιμες συμπληρωματικές ενδείξεις σχετικά με τις προθέσεις των χρηστών, τα ενδιαφέροντά τους, τη στάση τους απέναντι στους άλλους κ.ά. Στους εικονικούς κόσμους συνηθίζεται να απεικονίζονται στην ενσάρκωση και επιπλέον ενδείξεις σχετικά με την κατάσταση του χρήστη που βοηθούν την επικοινωνία, όπως για παράδειγμα αν έχει κακή σύνδεση δικτύου, αν είναι ανενεργός για πολλή ώρα, αν πληκτρολογεί κάποιο κείμενο κ.ά.

4.3 Συνεργασία

Σε περιβάλλοντα συνεργασίας είναι χρήσιμη η διάθεση εργαλείων για τον καλύτερο συντονισμό και οργάνωση των από κοινού δράσεων των χρηστών. Κατά τη σύγχρονη συνεργασία πολλαπλών χρηστών ένα ενδιαφέρον ζήτημα είναι ο έλεγχος του βήματος ομιλίας (*floor control*) και κατ' επέκταση της πρόσβασης σε κοινά εργαλεία (*access control*) (Dommel κ.ά., 1997). Παραδείγματα στα οποία προκύπτει η ανάγκη για τέτοιου τύπου ελέγχους είναι ο καθορισμός του κεντρικού ομιλητή σε ομάδες εργασίας, ο προσδιορισμός του χρήστη που θα αναλάβει την επεξεργασία ενός αντικειμένου στα πλαίσια μιας δραστηριότητας συνεργατικής σχεδίασης κ.ά. Υπάρχουν διάφορες προσεγγίσεις για τον έλεγχο του βήματος ομιλίας και της πρόσβασης. Η πιο απλή είναι να αποκτά τον έλεγχο όποιος τον ζητήσει πρώτος. Πιο σύνθετες περιπτώσεις είναι η πρόσβαση μέσω αιτήματος σε διαχειριστή, η χρήση μιας προκαθορισμένης ιεραρχίας βάσει της οποίας αποκτά τον έλεγχο όποιος βρίσκεται σε υψηλότερη βαθμίδα μεταξύ αυτών που τον ζήτησαν ή ακόμα και η χρήση ενός συστήματος κανόνων πρόσβασης που σχετίζονται με τη φύση της συνεργατικής εργασίας και τους ρόλους των συμμετεχόντων.

Το βασικό εργαλείο για την υποστήριξη της ασύγχρονης συνεργασίας σε διαμοιρασμένα περιβάλλοντα είναι οι *επισημειώσεις* (*annotations*) (Polys & Bowman, 2004). Οι επισημειώσεις είναι μηνύματα που μπορούν να αφήσουν οι χρήστες σε χώρους ή αντικείμενα του περιβάλλοντος. Τα μηνύματα αυτά απευθύνονται στα μέλη μιας συνεργαζόμενης ομάδας ή και στους ίδιους τους δημιουργούς για μεταγενέστερη χρήση και μπορούν να περιέχουν σχόλια, παρατηρήσεις ή επεξηγήσεις σχετικά με τα αντικείμενα ή τους χώρους αναφοράς. Οι επισημειώσεις συνήθως περιλαμβάνουν, πέρα από το ίδιο το μήνυμα, και στοιχεία όπως ο δημιουργός, η ημερομηνία κατασκευής, το είδος του περιεχομένου, ή ακόμα και λέξεις-κλειδιά που περιγράφουν το περιεχόμενο. Η μορφή τους μπορεί να είναι απλό κείμενο, ηχητικό μήνυμα, κάποια γραφική αναπαράσταση ή και εικόνα. Είναι σημαντικό η εμφάνιση των επισημειώσεων στο περιβάλλον να γίνεται με τέτοιο τρόπο ώστε να είναι εύκολα αναγνωρίσιμες, ενώ η δυνατότητα εύκολης αναζήτησης επισημειώσεων (π.χ. με βάση κριτήρια όπως ο δημιουργός, η ημερομηνία ή κάποιες λέξεις-κλειδιά) μπορεί να διευκολύνει ακόμα περισσότερο τον εντοπισμό και την επιτυχημένη αξιοποίησή τους.

4.4 Ομάδες χρηστών

Πολλοί εικονικοί κόσμοι προσφέρουν εργαλεία για τη δημιουργία και διαχείριση *ομάδων χρηστών* (*user groups*). Οι λόγοι για κάτι τέτοιο είναι αρκετοί. Σε συνεργατικά περιβάλλοντα προκύπτει η ανάγκη για τον καθορισμό ομάδων εργασίας, τη διαχείριση των μελών τους, την ανάθεση ρόλων και δικαιωμάτων κ.ά. Στην περίπτωση αυτή η χρήση των ομάδων χρηστών μπορεί να βοηθήσει στον καλύτερο συντονισμό, στον έλεγχο πρόσβασης και γενικότερα στην ασύγχρονη επικοινωνία και συνεργασία μιας ομάδας. Σε κοινωνικούς κόσμους, οι ομάδες χρηστών μπορούν επιπλέον να υποστηρίξουν «*εικονικές κοινότητες*», δηλαδή έναν μεγάλο αριθμό χρηστών με κοινά ενδιαφέροντα οι οποίοι επικοινωνούν και συμμετέχουν σε ομαδικές συζητήσεις και παρουσιάσεις εντός του κόσμου. Και σε αυτήν την περίπτωση, οι ομάδες χρηστών ενισχύουν την επικοινωνία και τον συντονισμό των δράσεων μιας κοινότητας. Τέλος, οι εικονικοί κόσμοι, στους οποίους οι ομάδες χρηστών φαίνεται να χρησιμοποιούνται ακόμα πιο συχνά, είναι τα πολυχρηστικά παιχνίδια. Εκεί γίνεται συνήθως διάκριση μεταξύ μικρών ομάδων που δημιουργούνται αυθόρμητα και σχηματίζονται για να πετύχουν έναν στόχο (*fellowships*), ομάδων με μεγαλύτερη διάρκεια στον χρόνο αποτελούμενων από χρήστες που συνηθίζουν να παίζουν μαζί (*parties*) ή και πολύ μεγάλων ομάδων με περισσότερο περιστασιακή σχέση (*Clans, Guilds* κ.λπ.). Οι ομάδες χρηστών, ανάλογα με τις ανάγκες της εφαρμογής, μπορεί να είναι τυπικές (δηλαδή να βασίζονται σε κάποια τυπική ομαδοποίηση που σχετίζεται με την εφαρμογή) ή άτυπες, προσωρινές ή μόνιμες, και να περιλαμβάνουν κάποιου είδους ιεραρχία ή να είναι όλα τα μέλη ισότιμα.

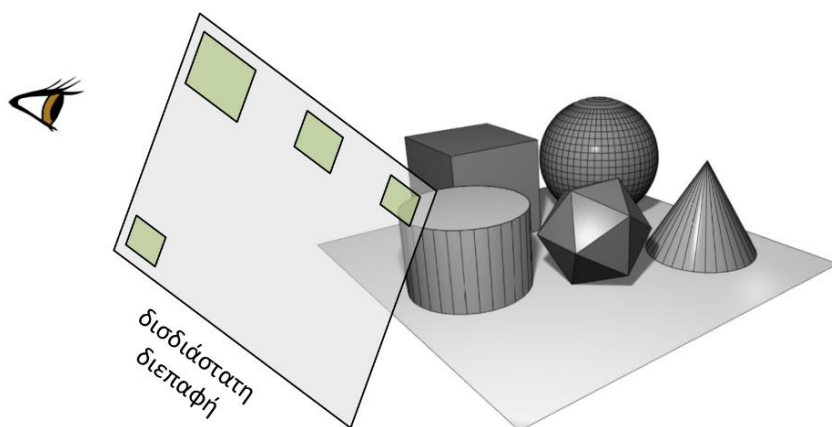
4.5 Συνέπεια κόσμου

Ένα τελευταίο σημαντικό σχεδιαστικό ζήτημα που σχετίζεται με την ταυτόχρονη παρουσία χρηστών σε ένα περιβάλλον έχει να κάνει με τη *συνέπεια κόσμου* (*world congruity*). Συνέπεια κόσμου είναι, πολύ απλά, το κατά πόσο όλοι οι χρήστες βλέπουν την ίδια εικόνα του κόσμου. Αν, για παράδειγμα, τη στιγμή που ένας χρήστης επεξεργάζεται ένα αντικείμενο, οι υπόλοιποι χρήστες βλέπουν όλη τη διαδικασία της επεξεργασίας ή μόνο το τελικό αποτέλεσμα. Είναι μάλλον αυτονόητο ότι όσο μεγαλύτερη συνέπεια κόσμου υπάρχει, τόσο ενισχύεται η επίγνωση των ενεργειών των άλλων χρηστών και γενικότερα το αίσθημα της συν-παρουσίας. Από την άλλη

μεριά, μπορεί να υπάρχουν λόγοι για τους οποίους είναι σκόπιμο να παραβιάζεται η συνέπεια σε κάποιες ειδικές περιπτώσεις. Σε περιβάλλοντα που υποστηρίζουν προσαρμογή (adaptivity) και προσωποποίηση (personalization) της διεπαφής μπορεί οι απόψεις ορισμένων χρηστών να είναι επαυξημένες με πληροφορίες, οπτικά βοηθήματα κ.ά. Σε σχεδιαστικές εφαρμογές ενδέχεται να επιθυμούν οι χρήστες να έχουν έναν προσωπικό σχεδιαστικό χώρο για να προετοιμάσουν το σχέδιό τους, και στη συνέχεια να το μοιραστούν με τους υπόλοιπους χρήστες μόνο όταν αυτό είναι ολοκληρωμένο. Τέλος, λόγοι ταχύτητας και απόδοσης του περιβάλλοντος μπορεί επίσης να καθιστούν την πλήρη συνέπεια απαγορευτική.

5 Δισδιάστατη διεπαφή – HUD

Σε όλα σχεδόν τα τρισδιάστατα περιβάλλοντα επιφάνειας εργασίας συμπεριλαμβάνονται και δισδιάστατα στοιχεία ελέγχου που ζωγραφίζονται πάνω από την απεικόνιση του περιβάλλοντος. Αυτού του είδους η δισδιάστατη διεπαφή με το περιβάλλον ονομάζεται και γραφικό περιβάλλον διεπαφής (Graphical User Interface – GUI) ή Head Up Display (HUD). Ο όρος HUD προέρχεται από τα μαχητικά αεροσκάφη, τα οποία έχουν την ικανότητα να προβάλλουν πληροφορίες σχετικά με την πτήση και τους εχθρικούς στόχους πάνω στο τζάμι, με αποτέλεσμα ο πιλότος να βλέπει τις πληροφορίες αυτές τοποθετημένες πάνω στην εικόνα που λαμβάνει από τον πραγματικό κόσμο. Αρχικά αυτή η απεικόνιση ήταν τοποθετημένη σε σταθερό σημείο (από εκεί προήλθε και ο όρος Head Up, δηλαδή η απεικόνιση είναι ορατή όταν το κεφάλι του πιλότου είναι όρθιο και κοιτάζει ευθεία), αργότερα ενσωματώθηκε στα κράνη των πιλότων ακολουθώντας τις κινήσεις του κεφαλιού τους, όπως τα κράνη εικονικής πραγματικότητας HMD. Μια παραπλήσια λογική υπάρχει και στους εικονικούς κόσμους, όπου η απεικόνιση του HUD ζωγραφίζεται σε ένα νοητό τζάμι μεταξύ του χρήστη και του εικονικού κόσμου (εικ. 4.7). Οι απεικονίσεις του HUD μπορεί να περιέχουν στοιχεία που αναφέρονται σε αντικείμενα του περιβάλλοντος, οπότε η εμφάνιση και τοποθέτησή τους σχετίζεται με αυτήν του αντίστοιχου αντικειμένου, ή και στοιχεία γενικότερου ενδιαφέροντος που είναι μόνιμα τοποθετημένα σε κάποιο σημείο της οθόνης ανεξαρτήτως της άποψης του περιβάλλοντος που μπορεί να έχει εκείνη τη στιγμή ο χρήστης. Τα στοιχεία της δισδιάστατης διεπαφής χρησιμοποιούνται για την απεικόνιση πληροφοριών σχετικά με το περιβάλλον και για την εισαγωγή δεδομένων από τον χρήστη (Chen κ.ά., 2004).



Εικόνα 4.7 Δισδιάστατη διεπαφή (HUD).

Κάποιος θα μπορούσε να αναρωτηθεί ποιος είναι ο λόγος ύπαρξης του HUD εφόσον οποιοδήποτε δισδιάστατο στοιχείο, διαδραστικό ή μη, μπορεί θεωρητικά να απεικονιστεί εντός του τρισδιάστατου χώρου. Η αλήθεια είναι ότι υπάρχουν διάφοροι λόγοι υπέρ της επιλογής της δισδιάστατης διεπαφής, οι οποίοι σχετίζονται κυρίως με την ποιότητα απεικόνισης και την ευχρηστία. Η προβολή μιας εικόνας ή ενός γραφικού στοιχείου στη δισδιάστατη διεπαφή διατηρεί πλήρως την ποιότητά της. Αν όμως προβληθεί ως υφή σε μια επιφάνεια του κόσμου, η απεικόνιση θα υποστεί μια σειρά από αλλοιώσεις: μπορεί να μειωθεί το μέγεθός της λόγω περιορισμών στα μεγέθη των υφών, θα τροποποιηθούν τα χρώματά της λόγω του μοντέλου φωτισμού και θα προβληθεί σε διαφορετική κλίμακα ή/και με παραμόρφωση αν ο χρήστης τη βλέπει από απόσταση ή υπό γωνία. Στην περίπτωση του κειμένου, οι συνέπειες της τοποθέτησής του πάνω σε επιφάνειες του κόσμου μπορεί να είναι ακόμη πιο κρίσιμες για την εμπειρία του χρήστη, καθώς ενδέχεται να το κάνουν δυσανάγνωστο. Ακόμα πιο επιτακτική είναι η χρήση δισδιάστατης διεπαφής στην περίπτωση προσθήκης σύνθετων στοιχείων ελέγχου

και εισαγωγής δεδομένων. Μέσω του HUD ο χρήστης μπορεί να χειριστεί αυτά τα στοιχεία με τρόπο ανάλογο με αυτόν που έχει συνηθίσει στις παραθυρικές εφαρμογές. Αντίθετα, αν κατασκευαστούν ως στοιχεία εντός του κόσμου, θα μειονεκτούν τόσο ως προς την απεικόνιση όσο και ως προς την αλληλεπίδραση λόγω των προβλημάτων μετάφρασης της κίνησης του ποντικιού στις τρεις διαστάσεις που έχουμε ήδη αναφέρει. Τέλος, κάποια από τα στοιχεία που εμφανίζονται στο HUD μπορεί να θέλουμε να είναι μόνιμα τοποθετημένα στην οθόνη του χρήστη και να ακολουθούν την κίνησή του, άρα δεν μπορούν να είναι μέρος του κόσμου. Αν υπάρχει κάποιο μειονέκτημα στη χρήση του HUD για την απεικόνιση στοιχείων διεπαφής είναι ότι η αλληλεπίδραση μεταφέρεται εκτός του τρισδιάστατου κόσμου, κάτι που μπορεί να έχει αρνητική επίπτωση στην αίσθηση της παρουσίας του χρήστη στον κόσμο.

Τα συνηθισμένα στοιχεία εισόδου δεδομένων που εμφανίζονται σε HUD είναι παραπλήσια με τα λειτουργικά τμήματα (components) των παραδοσιακών παραθυρικών διεπαφών. Ανάλογα με το είδος και τις ανάγκες της εφαρμογής μπορεί να αναδύονται μενού επιλογών, φόρμες διαλόγου, πεδία εισαγωγής κειμένου κ.λπ. Στις περισσότερες περιπτώσεις τα στοιχεία αυτά είναι προσωρινά και σχετίζονται με τη δραστηριότητα του χρήστη, δηλαδή συνήθως η εμφάνισή τους προκαλείται από κάποια αλληλεπίδραση με αντικείμενα του περιβάλλοντος. Για παράδειγμα, ένας χρήστης επιλέγει να αλλάξει το χρώμα της επιφάνειας ενός αντικειμένου και εμφανίζεται ένας διάλογος στο HUD, ο οποίος του επιτρέπει να προσδιορίσει το επιθυμητό χρώμα αριθμητικά ή οπτικά μέσω των αντίστοιχων στοιχείων ελέγχου. Υπάρχουν όμως και στοιχεία τα οποία μπορεί να βρίσκονται μόνιμα τοποθετημένα στο HUD, γιατί συνδέονται με ενέργειες που μπορούν να γίνουν σε οποιαδήποτε περιοχή του κόσμου. Χαρακτηριστικά τέτοια παραδείγματα είναι: το πεδίο εισαγωγής κειμένου για την επικοινωνία των χρηστών (chat), πλήκτρα τηλεμεταφοράς προς σημαντικούς προορισμούς του περιβάλλοντος, πλήκτρα αυτόματης εκτέλεσης ορισμένων συχνών ενεργειών (π.χ. κίνηση χαιρετισμού) κ.ά.

Το HUD μπορεί να αξιοποιηθεί και για την παροχή στοιχείων και πληροφοριών προς τον χρήστη. Γενικές πληροφορίες σχετικά με την κατάσταση του συστήματος ή στοιχεία που σχετίζονται με τον εικονικό κόσμο μπορούν να εμφανίζονται μόνιμα μέσω του HUD. Επιπλέον, στις περισσότερες κατηγορίες εικονικών κόσμων προκύπτει η ανάγκη να παρουσιάζονται και πληροφορίες που σχετίζονται με τον τρισδιάστατο χώρο, όπως για παράδειγμα στοιχεία σχετικά με τα αντικείμενα με τα οποία αλληλεπιδρά ο χρήστης ή και με τους υπόλοιπους χρήστες τους οποίους συναντάει. Οι πληροφορίες αυτές, αν απεικονίζονται σε δισδιάστατη μορφή (π.χ. κείμενο ή εικονίδια), εμφανίζονται συνήθως μέσω του HUD και τοποθετούνται με κατάλληλο τρόπο, ώστε ο χρήστης να μπορεί να τις συσχετίσει οπτικά με τις αντίστοιχες οντότητες του περιβάλλοντος. Για παράδειγμα, στο Second Life εμφανίζονται τα ονόματα των χρηστών στις ενσαρκώσεις τους, ενώ, αν το επιθυμεί ο χρήστης, μπορεί να βλέπει και τα μηνύματα που γράφουν τοποθετημένα ως «φούσκες ομιλίας» (speech bubbles) πάνω τους. Τέλος, σε ειδικού τύπου εφαρμογές μπορεί να απεικονίζονται και πληροφορίες που σχετίζονται με τη διεργασία που εκτελεί ο χρήστης, όπως για παράδειγμα ποιοι είναι οι στόχοι, ποια είναι η πρόοδος των εργασιών κ.ά.

Η αναπαράσταση πληροφοριών όπως οι παραπάνω δεν είναι υποχρεωτικό να γίνεται αποκλειστικά με τη μορφή κειμένου. Μπορεί να χρησιμοποιούνται και άλλοι τύποι αναπαραστάσεων, όπως χρωματισμοί, γραφικά, εικονίδια κ.ά. Τέτοια στοιχεία εμφανίζονται αρκετά συχνά στις διεπαφές των ψηφιακών παιχνιδιών, όπως μπάρες προόδου, μετρητές, ραντάρ κ.ά. Όταν ένα στοιχείο της διεπαφής είναι αρκετά μεγάλο σε μέγεθος, συνήθως χρησιμοποιείται και ημιδιαφάνεια στην απεικόνισή του, ώστε να μην εμποδίζεται σημαντικά η ορατότητα του τρισδιάστατου περιβάλλοντος.

Σε πολλές περιπτώσεις είναι χρήσιμο τα στοιχεία του HUD να μπορούν να προσαρμοστούν στις απαιτήσεις και ανάγκες των χρηστών. Συνήθως οι αρχάριοι χρήστες έχουν ανάγκη από περισσότερες πληροφορίες και βοηθήματα, ενώ οι πιο έμπειροι προτιμούν να έχουν μεγαλύτερη ορατότητα στο τρισδιάστατο περιβάλλον. Το HUD μπορεί να δίνει τη δυνατότητα ρύθμισης των στοιχείων διεπαφής, ώστε να μπορούν να απενεργοποιούνται ή να τροποποιείται το μέγεθος και η θέση εμφάνισης, βοηθώντας την εξατομίκευση της διεπαφής. Επιπλέον, κατάλληλα σχεδιασμένα στοιχεία διεπαφής θα μπορούσαν να παρέχουν εξατομικευμένες υποδείξεις στους χρήστες σχετικά με περιεχόμενα και δραστηριότητες του περιβάλλοντος. Για παράδειγμα, να ενεργοποιούνται βοηθήματα πλοήγησης που να κατευθύνουν τους χρήστες σε χώρους ή αντικείμενα που σχετίζονται με τους στόχους της εφαρμογής, να υποδεικνύονται χρήστες με παραπλήσια ενδιαφέροντα ή στόχους, ή ακόμα και να παρέχονται οδηγίες σχετικά με τη χρήση κάποιας νέας μορφής διεπαφής ή την εκτέλεση κάποιας σύνθετης ενέργειας.

Η τοποθέτηση των μόνιμων στοιχείων του HUD συνήθως γίνεται κοντά στα άκρα της οθόνης, ώστε να μην αποσπάται η προσοχή του χρήστη από τον εικονικό κόσμο και να είναι ευκολότερα προσβάσιμα με το ποντίκι, αν πρόκειται για στοιχεία εισόδου.

6 Σύγχρονες φυσικές διεπαφές

Τα τελευταία χρόνια έχουν κάνει την εμφάνισή τους συσκευές διεπαφής οι οποίες υποστηρίζουν περισσότερο «φυσικές» αλληλεπιδράσεις με το περιβάλλον και σε σχετικά χαμηλό κόστος συγκρινόμενες με το εξειδικευμένο υλικό εικονικής πραγματικότητας. Η ανάπτυξη των νέων αυτών μορφών αλληλεπίδρασης οφείλεται αφενός στην εξάπλωση των smartphones και tablets, τα οποία έχουν διαφορετικό μοντέλο αλληλεπίδρασης από το καθιερωμένο, και αφετέρου στη βιομηχανία παιχνιδιών η οποία έχει στραφεί σε πιο «φυσικές» και «ζωντανές» εμπειρίες παιχνιδιού. Οι εικονικοί κόσμοι, οι οποίοι τόσο ως προς την τεχνολογία όσο και ως προς τη διεπαφή έχουν πολλές ομοιότητες με τα παιχνίδια, μπορούν να επωφεληθούν από αυτές τις εξελίξεις και να εισαγάγουν νέες, βελτιωμένες τεχνικές αλληλεπίδρασης με το περιβάλλον.

6.1 Φορητές συσκευές

Η διάδοση των φορητών συσκευών δημιουργεί πρόσθετες δυσκολίες αλλά και προκλήσεις στους εικονικούς κόσμους. Από τη μία μεριά οι περιορισμένες επιδόσεις των συσκευών αυτών στα τρισδιάστατα γραφικά σε σχέση με τους οικιακούς υπολογιστές δυσκολεύουν την ανάπτυξη πλούσιου περιεχομένου και την ποιοτική απεικόνιση. Από την άλλη, οι κινητές συσκευές προσφέρουν νέες δυνατότητες αλληλεπίδρασης μέσω της οθόνης αφής και των αισθητήρων που διαθέτουν, οι οποίες μεταβάλλουν αντίστοιχα και την αλληλεπίδραση του χρήστη με το περιβάλλον.

Η βασική μορφή εισόδου στις φορητές συσκευές είναι η αφή. Μέσω της αφής μπορεί ο χρήστης να επιλέξει ή και να σύρει ένα στοιχείο διεπαφής, όπως θα έκανε αντίστοιχα και με το ποντίκι, αλλά και να πραγματοποιήσει διαφόρων ειδών «χειρονομίες» (gestures) με την κίνηση ενός ή δύο δαχτύλων. Όπως και στο ποντίκι, ο χρήστης ελέγχει δύο βαθμούς ελευθερίας. Όμως στην περίπτωση της αφής, ο έλεγχος είναι περισσότερο άμεσος, με την έννοια ότι τοποθετεί το δάχτυλό του πάνω στο ίδιο το αντικείμενο ενδιαφέροντος αντί να το καταδεικνύει μέσω του ποντικιού. Επιπλέον, η υποστήριξη χειρονομιών επιτρέπει στον χρήστη να επιλέξει εύκολα μεταξύ διαφορετικών μορφών αλληλεπίδρασης (π.χ. μετατόπιση και περιστροφή). Παρότι το μοντέλο είναι αρκετά διαισθητικό στις δύο διαστάσεις, εξακολουθεί να υπάρχει το πρόβλημα της απώλειας ενός βαθμού ελευθερίας. Για τον λόγο αυτόν, σε πολλές τρισδιάστατες εφαρμογές ο έλεγχος γίνεται μέσω εικονικών χειριστηρίων. Για παράδειγμα, για την πλοήγηση ακουμπάει ο χρήστης το δάχτυλό του στο κέντρο ενός στοιχείου διεπαφής και σέρνοντάς το προς μία διεύθυνση κινείται αντίστοιχα η κάμερα. Στην περίπτωση αυτήν, η ταχύτητα μετακίνησης ή περιστροφής μπορεί να ρυθμιστεί από τον χρήστη μέσω της απόστασης του δαχτύλου του από το κέντρο του στοιχείου. Επιπλέον, η υποστήριξη πολλαπλών επαφών από τις φορητές συσκευές δίνει στον χρήστη τη δυνατότητα να χρησιμοποιήσει και τα δύο χέρια, συνεπώς μπορεί να επιλέγει στοιχεία του περιβάλλοντος (π.χ. να κάνει κλικ σε μία πόρτα) ενώ παράλληλα πλοηγείται.

Μια άλλη τεχνική αλληλεπίδρασης που χρησιμοποιείται σε φορητές συσκευές είναι η κλίση (tilt) της συσκευής. Οι σύγχρονες φορητές συσκευές είναι εξοπλισμένες με αισθητήρες επιτάχυνσης και μπορούν να μετρήσουν αλλαγές στην κλίση τους. Εκμεταλλευόμενα αυτήν τη δυνατότητα, διάφορα παιχνίδια και εικονικά περιβάλλοντα έχουν χρησιμοποιήσει τις πληροφορίες κλίσης ως βασικό ή συμπληρωματικό μηχανισμό εισόδου. Για παράδειγμα, αν ο χρήστης στρέψει τη συσκευή προς τα εμπρός θα μπορούσε να κινείται προς την αντίστοιχη διεύθυνση η κάμερα και, ανάλογα με τη γωνία, να τροποποιείται η ταχύτητα βαδίσματος. Αντίστοιχα, αν η συσκευή στραφεί προς τα αριστερά ή προς τα δεξιά να αλλάζει η διεύθυνση κίνησης. Η τεχνική αυτή είναι αρκετά πιο φυσική, κάτι που επιδρά θετικά στην υποκειμενική ικανοποίηση των χρηστών, αλλά υπάρχουν ενδείξεις (Hürst & Nunez, 2013) ότι δεν είναι το ίδιο αποδοτική με την αλληλεπίδραση μέσω αφής.

6.2 Φυσική κίνηση σώματος

Τα τελευταία χρόνια αναδύθηκε στον χώρο της κοινότητας παικτών το ρεύμα του «φυσικού παιχνιδιού» (physical gaming), δηλαδή παιχνίδια στα οποία ο χρήστης συμμετέχει με κινήσεις του σώματός του μιμούμενος κάποια φυσική δραστηριότητα. Η πρώτη οικιακή συσκευή παιχνιδιών που υποστήριξε το μοντέλο αυτό με επιτυχία και έγινε ιδιαίτερα δημοφιλής ήταν το Wii της Nintendo, το οποίο κυκλοφόρησε το 2006. Η ιδιαιτερότητα της συσκευής ήταν ότι ο χρήστης κρατώντας ένα ειδικό χειριστήριο, το Wiimote, μπορούσε να κάνει φυσικές κινήσεις με τα χέρια του και αυτές να μεταφράζονται σε ενέργειες του παιχνιδιού. Για

παράδειγμα, σε ένα παιχνίδι τένις ο χρήστης κινούσε με το χέρι του το Wiimote σαν να κρατούσε μια φανταστική ρακέτα, και ο χαρακτήρας του στον κόσμο του παιχνιδιού μετέφερε την αντίστοιχη κίνηση στη ρακέτα του. Η τάση αυτή συνεχίστηκε με την εμφάνιση και άλλων συσκευών, τόσο από τη Nintendo (π.χ. σανίδα ισορροπίας Balance Board), όσο και από άλλες εταιρίες (π.χ. Playstation Move της Sony). Η πιο ενδιαφέρουσα όμως εξέλιξη στον χώρο αυτόν ήρθε από τη Microsoft με την εμφάνιση του Kinect. Το Kinect είναι μια συσκευή η οποία αναγνωρίζει τη στάση του σώματος του χρήστη μέσω συστήματος καμερών και μπορεί να ερμηνεύσει την κίνησή του χωρίς ο χρήστης να κρατάει κάποια συσκευή.

Η δυνατότητα αναγνώρισης της κίνησης μέρους ή όλου του σώματος του χρήστη δεν έμεινε φυσικά ανεκμετάλλευτη από τους εικονικούς κόσμους και τις τρισδιάστατες εφαρμογές γενικότερα. Μέσω συσκευών σαν τις παραπάνω θα μπορούσε να ενισχυθεί η φυσικότητα και διαισθητικότητα των αλληλεπιδράσεων χρήστη με το περιβάλλον. Το Wiimote χρησιμοποιήθηκε στο παρελθόν σε αρκετές εφαρμογές και δημόσιες εγκαταστάσεις εικονικών περιβαλλόντων ως διεπαφή για την επιλογή στοιχείων του περιβάλλοντος ή και για την πλοήγηση (Vosinakis & Xenakis, 2011). Όμως οι δυνατότητές του είναι σχετικά περιορισμένες και υστερεί σε ακρίβεια σε σχέση με παραδοσιακές συσκευές εισόδου (Ardito κ.ά., 2009) Αντίθετα, το Kinect, που εμφανίστηκε και αξιοποιήθηκε κάποια χρόνια αργότερα, οδήγησε σε πιο πλούσια και ενδιαφέροντα αποτελέσματα. Χρησιμοποιείται στα εικονικά περιβάλλοντα ως μέσο πλοήγησης και χειρισμού αντικειμένων προσφέροντας νέες, πιο φυσικές, τεχνικές αλληλεπίδρασης (Dam κ.ά., 2013). Για παράδειγμα, μία συχνή τεχνική επιλογής και χειρισμού είναι μέσω της κίνησης του χεριού. Ο χρήστης μετακινεί το χέρι του σε ένα νοητό κάθετο επίπεδο προκαλώντας την αντίστοιχη κίνηση ενός δείκτη στο περιβάλλον. Φέρνοντας το χέρι του μπροστά μπορεί να επιλέξει και να αλληλεπιδράσει με το αντικείμενο που βρίσκεται στη θέση του δείκτη, ενώ τραβώντας το χέρι πάλι πίσω το απελευθερώνει. Παράλληλα, για την πλοήγηση ο χρήστης μπορεί να χρησιμοποιήσει την κίνηση των ποδιών του ή και του σώματός του. Μια σχετική τεχνική είναι αυτή του «εικονικού κύκλου». Ορίζεται μια αρχική θέση στην οποία ο χρήστης θεωρείται ακίνητος, και όταν μετακινείται έξω από τη θέση αυτή, κινείται σε αντίστοιχη διεύθυνση στο περιβάλλον με ταχύτητα ανάλογη με την απόστασή του από την αρχική θέση. Επιπρόσθετα, περιστρέφοντας τους ώμους του του προς τα αριστερά ή προς τα δεξιά επιτυγχάνεται αντίστοιχη περιστροφή και στο περιβάλλον.

Σήμερα είναι διαθέσιμο ένα πρόγραμμα σύνδεσης (browser) σε εικονικούς κόσμους βασισμένους στο Second Life και στο OpenSimulator, το CtrlAltStudio, το οποίο υποστηρίζει το Kinect για πλοήγηση χρησιμοποιώντας την τεχνική του εικονικού κύκλου που περιγράψαμε παραπάνω. Επιπλέον, έχουν αναπτυχθεί τεχνολογίες που μπορούν να μεταφέρουν τη στάση του σώματος του χρήστη, όπως αυτή ερμηνεύεται από το Kinect, σε αντίστοιχες στάσεις της ενσάρκωσής του σε κάποιον εικονικό κόσμο (Cassola κ.ά., 2014). Αυτή η δυνατότητα, αν ενσωματωθεί σε μελλοντικά προγράμματα σύνδεσης, αναμένεται να ενισχύσει σημαντικά τη μη-λεκτική επικοινωνία μεταξύ των χρηστών του κόσμου.

6.3 Εμβυθισμένη προβολή του κόσμου

Μία ακόμα εξέλιξη, η οποία αναμένεται να αναζωπυρώσει το ενδιαφέρον για τους εικονικούς κόσμους, είναι η εμφάνιση του Oculus Rift, ενός κράνους εικονικής πραγματικότητας (HMD) χαμηλού κόστους για παιχνίδια. Η συσκευή αυτή, η οποία σήμερα διατίθεται για χρήση από προγραμματιστές και εταιρείες ανάπτυξης αλλά δεν έχει ακόμα κυκλοφορήσει στην αγορά στην τελική της μορφή, παρέχει στερεοσκοπική απεικόνιση και αναγνωρίζει τον προσανατολισμό και την κίνηση του κεφαλιού. Θα έχουν λοιπόν τη δυνατότητα οι χρήστες του να παρατηρούν το περιβάλλον σε τρισδιάστατη μορφή (μέσω στερεοσκοπίας) και να αλλάζουν τη διεύθυνση παρατήρησης με την κίνηση του κεφαλιού τους, που είναι ο πιο φυσικός και διαισθητικός τρόπος. Η χρήση της αναμένεται να προσφέρει στο ευρύ κοινό εμπειρία «εμβύθισης» ανάλογη με αυτήν των υψηλού κόστους συσκευών εικονικής πραγματικότητας, καθώς το περιβάλλον θα μοιάζει να «υπάρχει» γύρω τους, και θα εμπλουτίσει σημαντικά την εμπειρία χρήσης του κόσμου. Τόσο το επίσημο πρόγραμμα σύνδεσης στο Second Life όσο και το CtrlAltStudio που αναφέραμε πιο πάνω υποστηρίζουν σήμερα το Oculus Rift. Αυτό σημαίνει πως, όταν η συσκευή κυκλοφορήσει επίσημα στην αγορά, θα μπορούν οι επισκέπτες του Second Life και των κόσμων που τρέχουν σε πλατφόρμα OpenSimulator να την αξιοποιήσουν άμεσα.

6.4 Άλλες φυσικές διεπαφές

Εκτός από τις κατηγορίες συσκευών που παρουσιάσαμε ως τώρα, διατίθενται ή πρόκειται να διατεθούν στην αγορά και άλλες συσκευές φυσικών αλληλεπιδράσεων, οι οποίες ενδέχεται στο μέλλον να επηρεάσουν τη διεπαφή των εικονικών κόσμων. Το Leap Motion είναι μια χαμηλού κόστους συσκευή η οποία αναγνωρίζει το σχήμα και την κίνηση των χεριών σε ένα περιορισμένο εύρος αλλά με ικανοποιητική ακρίβεια (Coelho & Verweek, 2014). Η συσκευή αυτή θα μπορούσε να αξιοποιηθεί για τον φυσικό χειρισμό των αντικειμένων ενός εικονικού κόσμου με τρόπο ανάλογο με αυτόν των γαντιών δεδομένων. Το Virtuix Omni είναι μια πλατφόρμα βαδίσματος, η οποία επιτρέπει επιτόπιο βάδισμα προς οποιαδήποτε διεύθυνση με τη χρήση ειδικών παπουτσιών χαμηλής τριβής. Το Omni μπορεί να συνδυαστεί και με κράνη εικονικής πραγματικότητας, όπως το Oculus Rift, ενισχύοντας την εμπειρία εμβύθισης των χρηστών μέσω φυσικού βαδίσματος στον χώρο. Τέλος, οι διεπαφές εγκεφάλου-υπολογιστή, όπως το Emotiv EPOC, μπορούν να χρησιμοποιηθούν για την αναγνώριση των συναισθημάτων του χρήστη ή και για τον έλεγχο απλών ενεργειών με τη σκέψη. Οι συσκευές αυτές έχουν χρησιμοποιηθεί ήδη σε περιβάλλοντα εικονικής πραγματικότητας με ενθαρρυντικά αποτελέσματα (Jackson & Mappus, 2010· Renard κ.ά., 2010).

7 Συμπεράσματα

Στο παρόν κεφάλαιο έγινε μια επισκόπηση βασικών ζητημάτων που σχετίζονται με τη διεπαφή του χρήστη με τον εικονικό κόσμο. Παρουσιάστηκαν οι γενικές κατηγορίες αλληλεπίδρασης με το τρισδιάστατο περιβάλλον και οι διαθέσιμες τεχνικές για καθεμία από αυτές. Στη συνέχεια μελετήσαμε τις δυνατότητες της δισδιάστατης διεπαφής (HUD) στο να συνεισφέρει ως ένα επιπλέον κανάλι πληροφόρησης και στο να υποστηρίξει τις ενέργειες των χρηστών. Τέλος, αναφέραμε σύγχρονες κατηγορίες υλικού οι οποίες χρησιμοποιούνται ή πρόκειται να χρησιμοποιηθούν στους εικονικούς κόσμους και ενδέχεται να προσφέρουν περισσότερο φυσικές αλληλεπιδράσεις.

Αυτό που προκύπτει από την παραπάνω επισκόπηση είναι ότι απαιτείται πολύ καλή μελέτη του προβλήματος και προσεκτικός σχεδιασμός της διεπαφής του χρήστη με το περιβάλλον για την αποδοτική εκτέλεση των δραστηριοτήτων του. Ένα κρίσιμο ζήτημα το οποίο θα πρέπει να εξετάσουν οι σχεδιαστές είναι η κατάλληλη «μετάφραση» της δισδιάστατης εισόδου σε αντίστοιχες ενέργειες στον τρισδιάστατο χώρο και η διαχείριση των απαιτούμενων βαθμών ελευθερίας. Οι περιορισμοί μπορούν να αποτελέσουν ένα χρήσιμο εργαλείο για μεγαλύτερη ακρίβεια στον χειρισμό (π.χ. ένα αντικείμενο να μπορεί να τοποθετηθεί μόνο πάνω σε κάποια επιφάνεια) αλλά και για τη μείωση των βαθμών ελευθερίας. Ένα άλλο συναφές ζήτημα είναι η παροχή καλής άποψης του περιβάλλοντος κατά τη διάρκεια της αλληλεπίδρασης, ώστε να μπορεί ο χρήστης να αντιληφθεί καλύτερα την κατάσταση του περιβάλλοντος και τα αποτελέσματα των ενεργειών του. Αντίστοιχα, και στην περίπτωση της πλοήγησης θα πρέπει να προσφέρονται πολλαπλές και κατάλληλα σχεδιασμένες τεχνικές, ώστε να υποστηρίζονται οι ανάγκες των χρηστών ανάλογα με τους στόχους της εφαρμογής και τον απαιτούμενο ρεαλισμό. Επιπλέον, είναι απαραίτητη η παροχή ενός αριθμού από βοηθήματα πλοήγησης για τη διευκόλυνση της εύρεσης μονοπατιού και την υποστήριξη των αρχάριων χρηστών. Σε περιβάλλοντα στα οποία η έμφαση δίνεται στην επικοινωνία και στη συνεργασία μεταξύ των χρηστών, θα πρέπει επίσης να παρέχονται τα κατάλληλα εργαλεία για τη διαμόρφωση, τη διαχείριση και τον συντονισμό ομάδων εργασίας. Σε όλες τις παραπάνω περιπτώσεις, η δισδιάστατη διεπαφή μπορεί να αποτελέσει ένα σημαντικό βοηθητικό εργαλείο παρέχοντας κατάλληλες πληροφορίες και οπτικά βοηθήματα.

Τέλος, η εμφάνιση νέου υλικού αλληλεπίδρασης μεταβάλλει τις δυνατότητες εισόδου από την πλευρά των χρηστών αλλά και την προβολή του περιβάλλοντος στην περίπτωση της στερεοσκοπικής απεικόνισης, κάτι που αναμένεται να επηρεάσει σημαντικά τη σχεδίαση των διεπαφών των εικονικών κόσμων στο άμεσο μέλλον. Ο χρήστης θα μπορεί μέσω αυτών των συσκευών να δίνει δεδομένα κίνησης σε τρεις διαστάσεις και οι τρισδιάστατες εφαρμογές θα πρέπει να είναι σε θέση να τα χειριστούν κατάλληλα, ώστε να υποστηρίζονται περισσότερο φυσικές και διαισθητικές ενέργειες.

Αναφορές

- Ardito, C., Buono, P., Costabile, M. F., Lanzilotti, R. & Simeone, A. L. (2009). Comparing low cost input devices for interacting with 3D Virtual Environments. *Proceedings - 2009 2nd Conference on Human System Interactions, HSI '09*, 292–297.
- Bowman, D. A., Kruijff, E., LaViola, J. J. & Poupyrev, I. (2001). An Introduction to 3-D User Interface Design. *Presence: Teleoperators and Virtual Environments*, 10(1), 96–108.
- Burigat, S. & Chittaro, L. (2007). Navigation in 3D virtual environments: Effects of user experience and location-pointing navigation aids. *International Journal of Human Computer Studies*, 65(11), 945–958.
- Cassola, F., Morgado, L., de Carvalho, F., Paredes, H. & Fonseca, B. (2014). Online-Gym: A 3D Virtual Gymnasium Using Kinect Interaction. *Procedia Technology*, 13, 130–138.
- Chen, J. C. J., Pyla, P. S. & Bowman, D. A. (2004). Testbed evaluation of navigation and text display techniques in an information-rich virtual environment. *IEEE Virtual Reality 2004*.
- Chen, J. L. & Stanney, K. M. (1999). A Theoretical Model of Wayfinding in Virtual Environments: Proposed Strategies for Navigational Aiding. *Presence: Teleoperators and Virtual Environments*, 8(6), 671–685.
- Coelho, J. & Verveek, F. (2014). Pointing Task Evaluation of Leap Motion Controller in 3D Virtual Environment. In *Proceedings of the Chi Sparks 2014 Conference* (pp. 78–85).
- Dam, P. F., Braz, P. & Raposo, A. B. (2013). A Study of Navigation and Selection Techniques in Virtual Environments Using Microsoft Kinect ®. *15th International Conference on Human-Computer Interaction – HCI International 2013*, 139–148.
- Dommel, H.-P. & Garcia-Luna-Aceves, J. J. (1997). Floor control for multimedia conferencing and collaboration. *Multimedia Systems*, 5(1), 23–38.
- Hürst, W. & Nunez, H. C. (2013). Touch me, tilt me - Comparing interaction modalities for navigation in 2D and 3D worlds on mobiles. *Lecture Notes in Computer Science*, 8253 LNCS, 93–108.
- Jackson, M. M. & Mappus, R. (2010). Applications of Brain-Computer Interfaces. In *Brain-Computer Interfaces*. Springer.
- Mandel, T. (1997). *The Elements of User Interface Design*. John Wiley and Sons.
- Mine, M. R. (1995). *Virtual Environment Interaction Techniques*. UNC Chapel Hill computer science technical report TR95-018.
- Mine, M. R., Brooks, F. P. & Sequin, C. H. (1997). Moving objects in space. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques - SIGGRAPH '97* (pp. 19–26).
- Polys, N. F. & Bowman, D. a. (2004). Design and display of enhancing information in desktop information-rich virtual environments: Challenges and techniques. *Virtual Reality*, 8(1), 41–54.
- Renard, Y., Lotte, F., Gibert, G., Congedo, M., Maby, E., Delannoy, V., Lécuyer, A. (2010). OpenViBE: An Open-Source Software Platform to Design, Test, and Use Brain–Computer Interfaces in Real and Virtual Environments. *Presence: Teleoperators and Virtual Environments*, 19(1), 35–53.

Sherman, W. & Craig, A. (2003). *Understanding Virtual Reality*. Morgan Kaufmann.

Vosinakis, S. & Xenakis, I. (2011). A Virtual World Installation in an Art Exhibition: Providing a Shared Interaction Space for Local and Remote Visitors. *Proceedings of Re-Thinking Technology in Museums*.

Σύνδεσμοι

Microsoft Kinect: www.microsoft.com/en-us/kinectforwindows/

CtrlAltStudio Viewer: ctrlaltstudio.com/viewer

Oculus Rift: www.oculus.com

Leap Motion: www.leapmotion.com

Virtuix Omni: www.virtuix.com

Emotiv Epoc: emotiv.com/epoc.php

Περαιτέρω Μελέτη

Μπορείτε να διαβάσετε περισσότερα σχετικά με τις τρισδιάστατες διεπαφές στο βιβλίο:

Bowman, D., Kruijff, E., Jr, J. L. & Poupyrev, I. (2004). *3D user interfaces: theory and practice*.

Επιπλέον, πλούσιο υλικό όπως παρουσιάσεις και σημειώσεις μαθημάτων σχετικά με τρισδιάστατες διεπαφές θα βρείτε και στην ιστοσελίδα:

<http://people.cs.vt.edu/~bowman/3dui.org/Resources.html>

Ερωτήσεις Κατανόησης

1. Τι είναι οι μεταφορές διεπαφής και πώς μπορούν να χρησιμοποιηθούν στη σχεδίαση τεχνικών αλληλεπίδρασης;
2. Ποιες είναι οι βασικές κατηγορίες αλληλεπίδρασης στα εικονικά περιβάλλοντα;
3. Εξηγήστε τη διαφορά μεταξύ μετακίνησης και εύρεσης πορείας.
4. Να αναφέρετε τις βασικές τεχνικές μετακίνησης και παραδείγματα δραστηριοτήτων στις οποίες θα ήταν χρήσιμη η καθεμία από αυτές.
5. Ποια είναι κατά τη γνώμη σας τα πιο σημαντικά εργαλεία για την εύρεση πορείας σε εικονικούς κόσμους;
6. Πώς μπορεί να υποστηριχθεί η ελεύθερη μετατόπιση ενός αντικειμένου σε τρεις διαστάσεις με τη χρήση του ποντικιού;
7. Εξηγήστε τη σημασία των περιορισμών στην υποστήριξη των αλληλεπιδράσεων χρήστη.
8. Με ποιους τρόπους θα μπορούσαν δύο ή περισσότεροι χρήστες να επικοινωνήσουν μεταξύ τους σε έναν εικονικό κόσμο; Να αναφέρετε τα πλεονεκτήματα και τα μειονεκτήματα της κάθε λύσης.
9. Τι είναι οι επισημειώσεις και πώς μπορούν να διευκολύνουν τη συνεργασία;
10. Ποιος είναι ο ρόλος της διςδιάστατης διεπαφής στους εικονικούς κόσμους;
11. Ποιες είναι οι βασικές διαφορές ανάμεσα στις φορητές συσκευές σε σχέση με τους οικιακούς υπολογιστές ως πλατφόρμες εκτέλεσης εικονικών κόσμων;

12. Πώς αναμένεται να μεταβάλουν οι συσκευές ανίχνευσης κίνησης σώματος τις τεχνικές πλοήγησης και χειρισμού αντικειμένων;

Ασκήσεις

1. Συνδεθείτε σε δύο δημοφιλείς κόσμους, έναν κοινωνικό και έναν κόσμο παιχνιδιού, και καταγράψτε:
 - α) τις προσφερόμενες τεχνικές και βοηθήματα πλοήγησης,
 - β) τις δυνατότητες χειρισμού αντικειμένων και
 - γ) τα στοιχεία του HUD, τη μορφή και το ρόλο τους.Εντοπίστε τα κοινά στοιχεία και προσπαθήστε να εξηγήσετε τους λόγους διαφοροποίησης.
2. Συνδεθείτε σε περιβάλλον που επιτρέπει την ελεύθερη κατασκευή περιεχομένου (π.χ. OpenSimulator) και δοκιμάστε να κατασκευάσετε ένα σύνθετο αντικείμενο (π.χ. μία καρέκλα) από την αρχή. Καταγράψτε πιθανά προβλήματα και δυσκολίες χρήσης των εικονικών χειριστηρίων και των επιλογών πλοήγησης για τη συγκεκριμένη διεργασία, και προτείνετε εναλλακτικές τεχνικές ή και βοηθήματα που κατά τη γνώμη σας θα διευκόλυναν τις ίδιες ενέργειες.
3. Επισκεφθείτε κάποιον σχεδιασμένο χώρο ενδιαφέροντος στο Second Life και, αφού τον εξερευνήσετε, προτείνετε κατάλληλα βοηθήματα πλοήγησης που θα διευκολύνουν τους χρήστες στον εντοπισμό περιεχομένου μέσα σε αυτόν.

Κεφάλαιο 5: Σχεδίαση, Ανάπτυξη και Αξιολόγηση

Σύνοψη

Το παρόν κεφάλαιο αφορά τη διαδικασία σχεδίασης, ανάπτυξης και αξιολόγησης ενός εικονικού κόσμου. Αρχικά αναφέρονται ζητήματα καταλληλότητας των εικονικών κόσμων ως μέσου για την υποστήριξη των στόχων μιας εφαρμογής. Παρουσιάζονται και αναλύονται ορισμένες κυρίαρχες μεθοδολογίες σχεδίασης και ανάπτυξης εικονικών περιβαλλόντων και αναφέρονται τρόποι εφαρμογής των βασικών αρχών τους στους εικονικούς κόσμους. Στη συνέχεια παρουσιάζονται ζητήματα σχεδίασης του περιβάλλοντος και των αλληλεπιδράσεων, όπου περιγράφονται οι δυνατές μορφές αναπαράστασης του περιεχομένου και μελετάται η χρήση τους σε διάφορων ειδών περιβάλλοντα. Παρουσιάζονται ορισμένες βασικές κατηγορίες δραστηριοτήτων χρήστη που εμφανίζονται στους εικονικούς κόσμους και προτείνονται κατευθύνσεις ως προς το περιεχόμενο και τις υποστηριζόμενες αλληλεπιδράσεις. Τέλος, αναλύεται η έννοια της ευχρηστίας και οι μέθοδοι αξιολόγησης της και παρουσιάζονται ορισμένες βασικές προσεγγίσεις για την αξιολόγηση εικονικών περιβαλλόντων.

Προαπαιτούμενη γνώση

Απαιτούνται γνώσεις πληροφορικής, κατανόηση βασικών εννοιών αλληλεπίδρασης ανθρώπου-υπολογιστή, καθώς και γνώση των εισαγωγικών εννοιών των εικονικών κόσμων που παρουσιάζονται στο 1^ο Κεφάλαιο.

1 Εισαγωγή

Ένας κρίσιμος παράγοντας επιτυχίας των διαδραστικών εφαρμογών που απευθύνονται σε τελικούς χρήστες είναι η καλή σχεδίαση. Η επιλογή και ορθή χρήση κατάλληλων μεθόδων σχεδίασης των λειτουργιών, της διεπαφής και του περιεχομένου των εφαρμογών μπορεί να οδηγήσει στην υλοποίηση προϊόντων τα οποία είναι πιο κοντά στις πραγματικές ανάγκες και προτιμήσεις των χρηστών, λαμβάνουν υπόψη προηγούμενες γνώσεις και ιδιαιτερότητές τους, είναι ευχάριστα στη χρήση και γενικότερα τους διευκολύνουν στην επιτυχή εκτέλεση των απαιτούμενων δραστηριοτήτων.

Βεβαίως, η σχεδίαση ως διαδικασία λήψης αποφάσεων σχετικά με τις λεπτομέρειες υλοποίησης ενός προϊόντος δεν είναι από μόνη της αρκετή για να διασφαλίσει την καλή λειτουργία του. Απαιτούνται και διαδικασίες αξιολόγησης από ειδικούς αλλά και η συμμετοχή των χρηστών καθ' όλη την πορεία ανάπτυξης, ώστε να εντοπίζονται και να διορθώνονται από νωρίς πιθανά προβλήματα. Τα προβλήματα αυτά ενδέχεται να οδηγήσουν σε επανασχεδίαση ορισμένων πτυχών του προϊόντος, συμπληρωματική υλοποίηση, νέα αξιολόγηση κ.λπ. Επομένως, σύμφωνα με την παραπάνω πορεία, η σχεδίαση, η ανάπτυξη και η αξιολόγηση είναι διαδικασίες που εκτελούνται επαναληπτικά και σε διάφορα επίπεδα λεπτομέρειας κατά την κατασκευή μιας εφαρμογής. Πράγματι, οι σύγχρονες προσεγγίσεις στην ανάπτυξη εφαρμογών δίνουν ιδιαίτερη έμφαση στη συμμετοχή των χρηστών από τα πρώτα στάδια της σχεδίασης, στην κατασκευή πολλαπλών ενδιάμεσων πρωτοτύπων και στην αξιολόγησή τους (Abrás κ.ά., 2004).

Οι εικονικοί κόσμοι δεν θα μπορούσαν να αποτελούν εξαίρεση στα παραπάνω. Με δεδομένο ότι πρόκειται για περιβάλλοντα τα οποία όχι απλά απευθύνονται σε κάποιον τελικό χρήστη αλλά υποστηρίζουν την ταυτόχρονη συνύπαρξη, επικοινωνία και συνδημιουργία πολλαπλών χρηστών, η σχεδίαση με έμφαση στις ανάγκες των χρηστών είναι απαραίτητη.

Στους εικονικούς κόσμους μπορούμε να διακρίνουμε δύο επίπεδα σχεδίασης και ανάπτυξης: α) τη δημιουργία ενός νέου εικονικού κόσμου γενικού ή ειδικού σκοπού και β) τη δημιουργία ενός χώρου ειδικού σκοπού εντός ενός υπάρχοντος εικονικού κόσμου. Στην πρώτη περίπτωση, βεβαίως, ο χρόνος ανάπτυξης θα είναι πολύ μεγαλύτερος και η σύνθεση της ομάδας θα περιλαμβάνει αναγκαστικά ένα εύρος ειδικοτήτων. Πέρα από το ίδιο το περιεχόμενο και τις λειτουργίες του κόσμου, θα πρέπει η ομάδα ανάπτυξης να δημιουργήσει την κατάλληλη υποδομή για τη διαχείριση και αποθήκευση των δεδομένων, την αναπαράσταση και κίνηση της σκηνής, τη δικτυακή επικοινωνία μεταξύ των χρηστών και του κόσμου κ.ά. Αντίθετα, στη δεύτερη περίπτωση, ένα μεγάλο μέρος της παραπάνω υποδομής παρέχεται έτοιμο από το περιβάλλον εκτέλεσης, και η ομάδα ανάπτυξης δίνει έμφαση μόνο στην κατασκευή του περιεχομένου, της συμπεριφοράς των αντικειμένων και της διεπαφής με τον χρήστη. Το μειονέκτημα στην περίπτωση αυτήν είναι ότι οι σχεδιαστικές επιλογές περιορίζονται από τις παρεχόμενες λειτουργίες του κόσμου και τις δυνατότητες προγραμματισμού της συμπεριφοράς των αντικειμένων του. Για παράδειγμα, στο Second Life και στο OpenSimulator ο

ενσωματωμένος μηχανισμός φυσικής δεν είναι πολύ υψηλής ποιότητας και αρκετές φορές τα αντικείμενα έχουν «παράλογη» φυσική συμπεριφορά. Αν λοιπόν θελήσουμε να κατασκευάσουμε έναν εικονικό κόσμο για ελεύθερα πειράματα φυσικής, οι κόσμοι αυτοί δεν είναι οι καταλληλότεροι και ενδέχεται να επιλέξουμε τελικά να αναπτύξουμε το περιβάλλον από την αρχή, για παράδειγμα με τη χρήση κάποιας μηχανής παιχνιδιών.

Σε αντίθεση πάντως με άλλες διαδραστικές εφαρμογές, οι εικονικοί κόσμοι παρουσιάζουν σημαντικές ιδιαιτερότητες που καθιστούν τις διεργασίες σχεδίασης και ανάπτυξης ιδιαίτερα δύσκολες και απαιτητικές, όπως προκύπτει και από σχετικές μελέτες (Fencott, 1999· Wingrave & Laviola, 2010). Η ομάδα ανάπτυξης θα πρέπει να κατέχει πολλαπλές και εξειδικευμένες γνώσεις, με αποτέλεσμα να είναι δύσκολη η συγκέντρωση ατόμων με κατάλληλη εμπειρία και ο συντονισμός τους στα πλαίσια ενός κοινού έργου. Επιπλέον, η ανάπτυξη ικανοποιητικού περιεχομένου είναι μια ιδιαίτερα επίπονη και χρονοβόρα διεργασία που περιλαμβάνει κατασκευή υφών, μοντελοποίηση, συνθετική κίνηση, βελτιστοποίηση μοντέλων κ.ά. Και στην περίπτωση της ανάπτυξης του λογισμικού όμως, τα πράγματα είναι αντίστοιχα σύνθετα: υπάρχουν πολλαπλές διεργασίες που θα πρέπει να εκτελούνται παράλληλα και απαιτείται καλή διαχείριση και συντονισμός για ορθή και αποδοτική λειτουργία. Από την άλλη μεριά, η εκ των προτέρων πρόβλεψη της συμπεριφοράς και της εμπειρίας των χρηστών είναι δύσκολη, με αποτέλεσμα να υπάρχει ανάγκη για συχνότερη κατασκευή πρωτοτύπων σε σχέση με άλλες εφαρμογές. Πράγματι, οι χρήστες τοποθετούνται σε ένα περιβάλλον στο οποίο η διεπαφή είναι πολύ διαφορετική από αυτές που έχουν συνηθίσει (π.χ. παραθυρικά περιβάλλοντα ή περιβάλλοντα αφής στις κινητές συσκευές) και αποκτούν προοδευτικά την ικανότητα αντίληψης και καλής χρήσης του συστήματος. Παράλληλα, οι σχεδιαστικές οδηγίες (guidelines) και τα πρότυπα (design patterns) που θα μπορούσαν να κατευθύνουν τη σχεδίαση εικονικών κόσμων είναι σήμερα αρκετά περιορισμένα. Επομένως, μόνο μέσω πρωτοτύπων μπορεί να αξιολογηθεί και να επαληθευτεί η καλή σχεδίαση των διάφορων πτυχών και λειτουργιών του κόσμου. Η περίπτωση των εικονικών κόσμων γενικού σκοπού είναι ακόμη πιο σύνθετη, διότι δεν υπάρχουν τυπικοί χρήστες με συγκεκριμένους στόχους· εν δυνάμει όλοι μπορούν να είναι χρήστες. Εκεί λοιπόν η σχεδίαση θα πρέπει να περιλαμβάνει πολλαπλές εναλλακτικές στην εκτέλεση των διάφορων λειτουργιών και στοιχεία προσαρμογής και εξατομίκευσης της διεπαφής, με στόχο την υποστήριξη όσο το δυνατόν μεγαλύτερου εύρους χρηστών.

Στη συνέχεια του κεφαλαίου θα μελετήσουμε σε μεγαλύτερο βάθος ζητήματα σχεδίασης, ανάπτυξης και αξιολόγησης εικονικών κόσμων και θα παρουσιάσουμε ορισμένες σημαντικές σχετικές προσεγγίσεις. Αρχικά θα εξετάσουμε κατά πόσο οι εικονικοί κόσμοι αποτελούν κατάλληλο μέσο για την υποστήριξη μιας εφαρμογής ανάλογα με τις ανάγκες και τα χαρακτηριστικά της. Στη συνέχεια θα παρουσιάσουμε συνοπτικά τρεις βασικές μεθοδολογίες σχεδίασης και ανάπτυξης εικονικών περιβαλλόντων και θα εξετάσουμε τον τρόπο με τον οποίο μπορούν αυτές να αξιοποιηθούν στην περίπτωση των εικονικών κόσμων. Θα παρουσιάσουμε ζητήματα κατάλληλης αναπαράστασης του περιβάλλοντος και υποστήριξης των αλληλεπιδράσεων των χρηστών ανάλογα με τους στόχους και τις απαιτήσεις διάφορων δραστηριοτήτων. Τέλος, θα ασχοληθούμε με το κρίσιμο ζήτημα της ευχρηστίας στους εικονικούς κόσμους και θα παρουσιάσουμε ορισμένες σημαντικές μεθόδους για την αξιολόγησή της.

2 Καταλληλότητα του μέσου

Ένα πρώτο, κρίσιμο ερώτημα το οποίο καλείται να απαντήσει μια σχεδιαστική ομάδα πριν την απόφαση ανάπτυξης ενός εικονικού κόσμου είναι το εξής: είναι οι εικονικοί κόσμοι κατάλληλο μέσο για την υποστήριξη της εφαρμογής; Το ερώτημα, αν και απλό στη διατύπωση, δεν έχει πάντα εξίσου προφανή απάντηση. Η ανάπτυξη μιας εφαρμογής βασισμένης στους εικονικούς κόσμους, ακόμα και ως σχεδιασμένη περιοχή εντός ενός υπάρχοντος κόσμου, είναι μια επίπονη και χρονοβόρα διαδικασία. Ταυτόχρονα έχει και ορισμένες επιπτώσεις σε ό,τι αφορά την ευκολία χρήσης και την προσβασιμότητα: αποκλείει χρήστες με υπολογιστικά συστήματα χαμηλών επιδόσεων, συνήθως απαιτεί την εγκατάσταση κάποιου εξειδικευμένου λογισμικού για τη σύνδεση στον κόσμο, ενώ τοποθετεί τους χρήστες και σε ένα αρκετά διαφορετικό, περισσότερο περίπλοκο περιβάλλον διεπαφής. Η λύση λοιπόν των εικονικών κόσμων προσθέτει μια σημαντική επιβάρυνση τόσο στην ομάδα ανάπτυξης όσο και στους τελικούς χρήστες. Ως εκ τούτου, θα πρέπει, όταν επιλέγεται το μέσο αυτό, να είναι βέβαιο ότι είναι όχι απλά ικανό να υποστηρίξει την εφαρμογή, αλλά ότι την υποστηρίζει καλύτερα σε σχέση με τις διαθέσιμες εναλλακτικές.

Όπως αναφέρθηκε και στην εισαγωγή, οι εικονικοί κόσμοι μπορούν να υποστηρίξουν με επιτυχία διάφορες διεργασίες, όπως τρισδιάστατη οπτικοποίηση δεδομένων, προσομοίωση διαδικασιών, συνεργατική σχεδίαση περιεχομένου, ελεύθερη εξερεύνηση και αλληλεπίδραση με τον κόσμο, κοινωνική δικτύωση χρηστών

κ.ά. Το πιο ενδιαφέρον στοιχείο όμως είναι ότι όλα τα παραπάνω υποστηρίζονται σε ένα ενιαίο περιβάλλον. Είναι σημαντικό λοιπόν για μια ομάδα ανάπτυξης να αποφασίσει κατά πόσο οι εικονικοί κόσμοι είναι όντως η πιο κατάλληλη λύση για την εφαρμογή, αποφασίζοντας τον βαθμό στον οποίο η εφαρμογή σχετίζεται με μία ή κατά προτίμηση περισσότερες από τις παραπάνω διεργασίες.

- Είναι το πρόβλημα το οποίο θέλουμε να λύσει η εφαρμογή κατά βάση τρισδιάστατο;
- Εξελίσσεται το περιβάλλον στον χρόνο;
- Απαιτείται η ενεργός εμπλοκή των χρηστών στην εφαρμογή;
- Είναι υποχρεωτική η ταυτόχρονη παρουσία πολλαπλών χρηστών;
- Υπάρχει ανάγκη παρουσίασης κάποιας τρισδιάστατης δομής ή χώρου;
- Απαιτείται αναπαράσταση κάποιου ρεαλιστικού περιβάλλοντος;

Ερωτήσεις σαν τις παραπάνω μπορούν να βοηθήσουν τους σχεδιαστές να κατανοήσουν καλύτερα τη σχέση της εφαρμογής που θέλουν να αναπτύξουν με τις προσφερόμενες δυνατότητες του μέσου.

Ας εξετάσουμε την εφαρμογή των παραπάνω με ένα παράδειγμα: Έστω ότι θέλουμε να κατασκευάσουμε ένα εικονικό μουσείο, δηλαδή έναν μουσειακό χώρο υλοποιημένο σε ένα τρισδιάστατο περιβάλλον με στόχο οι χρήστες να μπορούν να εξερευνήσουν τον χώρο και να μελετήσουν τα εκθέματα. Αν το ενδιαφέρον των χρηστών επικεντρώνεται μόνο στα ίδια τα εκθέματα και όχι στην αρχιτεκτονική του χώρου και στον τρόπο τοποθέτησής τους, τότε ενδέχεται η παρουσίασή τους σε μια ιστοσελίδα να είναι προτιμότερη λύση σε σχέση με έναν εικονικό κόσμο, ιδιαίτερα αν η μορφή των εκθεμάτων είναι κατά βάση δισδιάστατη, για παράδειγμα πίνακες ζωγραφικής. Εναλλακτικά, θα μπορούσαν να χρησιμοποιηθούν τρισδιάστατα πανοράματα κατασκευασμένα από φωτογραφίες των χώρων του αντίστοιχου φυσικού μουσείου. Οι εικονικοί κόσμοι θα είχαν περισσότερο νόημα μόνο εάν υπήρχε ανάγκη και για επιπλέον δραστηριότητες, πέρα από την απεικόνιση των περιεχομένων, οι οποίες δεν μπορούν να υποστηριχθούν από τυπικές ιστοσελίδες μουσείων και στατικά πανοράματα. Τέτοια παραδείγματα είναι: συμμετοχικές δραστηριότητες, όπως παρουσιάσεις, συζητήσεις και παιχνίδια ρόλων εντός του χώρου, αλληλεπίδραση με τα εκθέματα, όπως το να μπορεί ο χρήστης να χειριστεί ένα αντικείμενο, να φορέσει μια ενδυμασία ή να παίξει κάποιο παιχνίδι που σχετίζεται με το έκθεμα, και εικονικοί ξεναγοί που παρουσιάζουν τον χώρο στους επισκέπτες και απαντούν στις ερωτήσεις τους.

Ένα δεύτερο ερώτημα το οποίο θα πρέπει να απαντήσει η ομάδα ανάπτυξης είναι το κατά πόσο η λύση των εικονικών κόσμων είναι *εφικτή* για την εφαρμογή. Είναι δηλαδή πιθανό να μπορεί μεν η εφαρμογή να επωφεληθεί από τις θεωρητικές δυνατότητες των εικονικών κόσμων, αλλά στην πράξη η υλοποίηση ενός τέτοιου συστήματος να μην επιτυγχάνει τον απαιτούμενο βαθμό ποιότητας. Πράγματι, οι εικονικοί κόσμοι, όπως έχουμε αναφέρει και σε προηγούμενα κεφάλαια, έχουν ισχυρή εξάρτηση από τον χρόνο. Θα πρέπει πάση θυσία να αποδίδεται ένας ικανοποιητικός ρυθμός ανανέωσης οθόνης. Αυτό σημαίνει ότι διεργασίες με πολύ μεγάλο υπολογιστικό κόστος πρακτικά αποκλείονται. Κατά συνέπεια, αν η εφαρμογή περιλαμβάνει περίπλοκες προσομοιώσεις που δεν είναι δυνατόν να εκτελεστούν σε πραγματικό χρόνο ή αν για την απαιτούμενη ποιότητα απεικόνισης και κίνησης χρειάζεται υπερβολικά μεγάλος αριθμός πολυγώνων ή και σύνθετοι αλγόριθμοι, τότε ενδέχεται οι εικονικοί κόσμοι να μην μπορούν να αποτελέσουν ικανοποιητική λύση στο πρόβλημα με βάση τις σημερινές δυνατότητες των υπολογιστικών συστημάτων.

3 Μεθοδολογίες σχεδίασης και ανάπτυξης

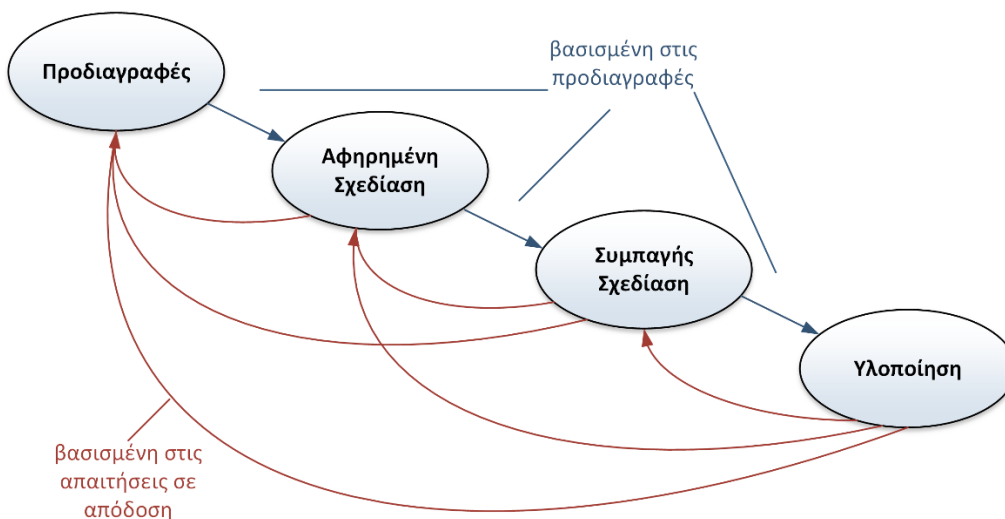
Οι σημαντικές διαφοροποιήσεις στη σχεδίαση και ανάπτυξη εικονικών κόσμων σε σχέση με άλλες διαδραστικές εφαρμογές έχουν οδηγήσει στη διαμόρφωση εξειδικευμένων μεθοδολογιών που λαμβάνουν υπόψη τις ιδιαιτερότητες του μέσου. Οι περισσότερες από αυτές τις μεθοδολογίες αναφέρονται κυρίως σε εφαρμογές εικονικής πραγματικότητας, με αποτέλεσμα να περιλαμβάνουν τη χρήση εξειδικευμένου υλικού εμφύθισης, όπως κράνη εικονικής πραγματικότητας και γάντια δεδομένων, και να δίνουν μικρότερη έμφαση σε θέματα επικοινωνίας και συνεργασίας πολλαπλών χρηστών. Παρά τις διαφοροποιήσεις αυτές, τα κοινά στοιχεία των εφαρμογών εικονικής πραγματικότητας με τους εικονικούς κόσμους είναι πολλά περισσότερα, και ως εκ τούτου η αξία των σχετικών μεθοδολογιών κρίνεται σημαντική. Στη συνέχεια θα παρουσιάσουμε τις περισσότερο δημοφιλείς από αυτές τις μεθοδολογίες, εστιάζοντας κυρίως στις πτυχές εκείνες που βρίσκουν εφαρμογή στους εικονικούς κόσμους.

3.1 Συνάντηση στη μέση (S. Bryson)

Ο Steve Bryson (1994) παρουσίασε μια μεθοδολογία σχεδίασης και ανάπτυξης εφαρμογών εικονικής πραγματικότητας, η οποία βασίζεται στον συμβιβασμό μεταξύ της ικανοποίησης των στόχων της εφαρμογής και της επίτευξης των περιορισμών απόδοσης. Οι βασικές διαπιστώσεις που τον οδήγησαν σε αυτή την πρόταση είναι δύο: Πρώτον, με δεδομένο ότι η εικονική πραγματικότητα υπόσχεται ότι μπορείς να κάνεις «οτιδήποτε φανταστείς», οι απαιτήσεις και οι επιθυμίες των χρηστών δεν έχουν περιορισμό. Ίδανικά οι χρήστες θα ήθελαν όσο το δυνατόν πιο ποιοτικά γραφικά, ρεαλιστική κίνηση, περίπλοκα φυσικά μοντέλα, αληθοφανή συμπεριφορά κ.λπ. Όλα αυτά όμως έχουν ένα υπολογιστικό κόστος, το οποίο μπορεί να καταλήξει να είναι τόσο μεγάλο που η εφαρμογή δεν θα μπορεί να λειτουργήσει με ικανοποιητική απόδοση. Δεύτερον, αν η ανάπτυξη της εφαρμογής γίνει ξεκινώντας από την κάλυψη των δεικτών απόδοσης, μπορεί να καταλήξει σε ένα αποτέλεσμα που δεν είναι συνεπές με τις ανάγκες των χρηστών και στερείται ευχρηστίας.

Στην ανάπτυξη εφαρμογών υπάρχουν δύο βασικά ρεύματα. Το ρεύμα «από την κορυφή προς τα κάτω» (top-down), που ξεκινάει από τις απαιτήσεις των χρηστών και καταλήγει στην ανάπτυξη του συστήματος, και το ρεύμα «από κάτω προς τα πάνω» (bottom-up) που ξεκινάει από την υλοποίηση και προσθέτει σταδιακά λειτουργίες για την κάλυψη των απαιτήσεων.

Οι περισσότερες σύγχρονες εφαρμογές αναπτύσσονται ακολουθώντας την top-down προσέγγιση, επειδή ακριβώς εστιάζουν στις ανάγκες των χρηστών και του γενικότερου προβληματικού χώρου στον οποίο εντάσσεται η εφαρμογή. Αρχικά γίνεται μια αφαιρετική/εννοιολογική σχεδίαση που προσδιορίζει τις βασικές οντότητες και λειτουργίες του συστήματος, στη συνέχεια μια περισσότερο αναλυτική που περιγράφει διεξοδικά τις λεπτομέρειες κατασκευής των επιμέρους τμημάτων του συστήματος και τελικά γίνεται η υλοποίηση. Σε όλα αυτά τα στάδια υπάρχει βεβαίως ισχυρή ανάδραση μέσω πολλαπλών αξιολογήσεων διαφόρων ειδών, αλλά σε κάθε περίπτωση πρώτα προσδιορίζονται οι προδιαγραφές και οι βασικές λειτουργίες του συστήματος και μετά αυτό υλοποιείται. Μία τέτοια πορεία ελλοχεύει πολλούς κινδύνους στην περίπτωση των εφαρμογών πραγματικού χρόνου, όπως είναι τα εικονικά περιβάλλοντα και οι εικονικοί κόσμοι, καθώς είναι πιθανό όταν φτάσει η ομάδα ανάπτυξης στο στάδιο της υλοποίησης να συνειδητοποιήσει ότι το απαιτούμενο περιεχόμενο και οι αντίστοιχες λειτουργίες δεν μπορούν να αποδοθούν σε πραγματικό χρόνο. Αναγκαστικά θα πρέπει από πολύ νωρίς να μπορεί να υπάρχει καλή εκτίμηση της απόδοσης της τελικής εφαρμογής, ώστε να διασφαλιστεί η εφικτότητα της υλοποίησης.

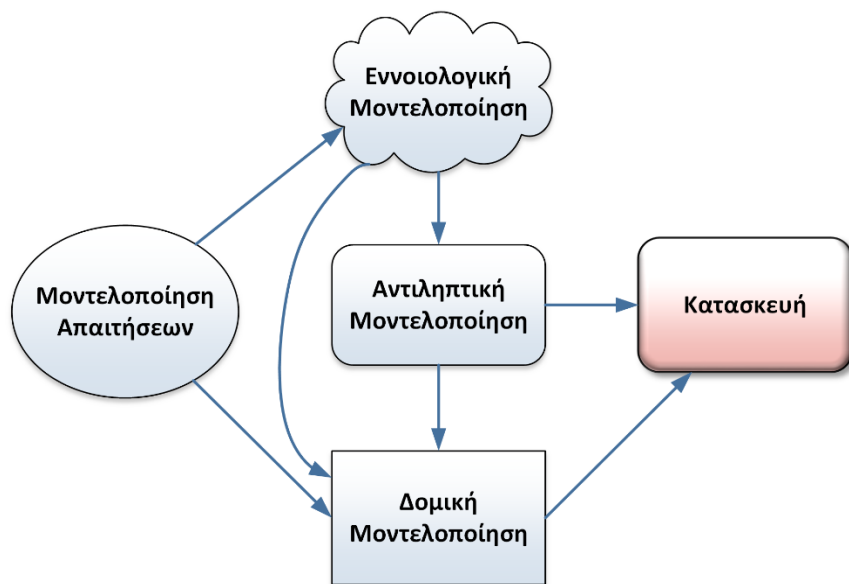


Εικόνα 5.1 Η μεθοδολογία «συνάντησης στη μέση» του S. Bryson.

Η προτεινόμενη πορεία ανάπτυξης λοιπόν, σύμφωνα με τη μεθοδολογία του Bryson, είναι ο συνδυασμός των δύο ρευμάτων ανάπτυξης, έτσι ώστε να «συναντιούνται» στη μέση (εικ. 5.1). Στην top-down πορεία, η έμφαση θα δίνεται στην ικανοποίηση των διεργασιών που θα πρέπει να υποστηρίξει η εφαρμογή και στην επιλογή των κατάλληλων μεταφορών. Οι σχεδιαστές θα πρέπει να μελετήσουν τον χώρο εφαρμογής και να προσδιορίσουν τις βασικές μεταφορές ως προς τρία επίπεδα: την παρουσίαση του περιβάλλοντος, την παρουσίαση

πληροφοριών και τη διεπαφή του χρήστη με το περιβάλλον. Για καθένα από αυτά τα επίπεδα μπορεί να χρειαστεί να προσδιοριστούν περισσότερες από μία επιμέρους μεταφορές. Η επιλογή καλών μεταφορών σχετίζεται με το κατά πόσο η μεταφορά είναι εγγενής σε σχέση με τις διεργασίες που καλείται να πραγματοποιήσει ο χρήστης στην εφαρμογή και κατά πόσο μπορεί να βελτιστοποιήσει τον χρόνο εκτέλεσής τους. Από την άλλη μεριά, στην bottom-up πορεία, η έμφαση θα πρέπει να δοθεί στην απόδοση. Οι σχεδιαστές θα πρέπει να αποφασίσουν από νωρίς για θέματα όπως η επιλογή της πλατφόρμας εκτέλεσης, η αρχιτεκτονική του συστήματος, η επιλογή κατάλληλων αλγορίθμων, η επιλογή δομών δεδομένων και αναπαραστάσεων κ.λπ. Οι δύο αυτές πορείες προτείνεται να εκτελούνται σχεδόν παράλληλα δίνοντας ανάδραση η μία στην άλλη μέσω πολλαπλών πρωτοτύπων και αξιολογήσεων από χρήστες και οδηγώντας σε επανασχεδίαση όπου απαιτείται. Σε κάποιες περιπτώσεις μπορεί οι περιορισμοί στην απόδοση να οδηγήσουν σε τροποποιήσεις στις μεταφορές που θα χρησιμοποιηθούν ή ακόμα και στις λειτουργίες που τελικά θα υποστηρίξει το σύστημα.

Αν και η μεθοδολογία του Bryson αναφέρεται κυρίως σε εμβυθισμένα περιβάλλοντα και προτάθηκε σε μια εποχή που οι δυνατότητες των υπολογιστικών συστημάτων ήταν πολύ πιο περιορισμένες σε σχέση με σήμερα, οι βασικές διαπιστώσεις εξακολουθούν να ισχύουν. Σήμερα εξακολουθεί και υπάρχει η τάση για ποιοτικά αποτελέσματα, και οι χρήστες γίνονται όλο και πιο απαιτητικοί σε σχέση με τα γραφικά και την κίνηση στους εικονικούς κόσμους. Επιπλέον, παρά το γεγονός ότι τα σύγχρονα συστήματα δίνουν πολύ καλά οπτικά αποτελέσματα, υπάρχει μια σαφής μετατόπιση του ενδιαφέροντος των χρηστών από τους σταθερούς υπολογιστές προς τους κινητούς (π.χ. smartphones και tablets), οι οποίοι δεν έχουν αντίστοιχη επεξεργαστική ισχύ. Εικονικοί κόσμοι που αναπτύσσονται γι' αυτές τις πλατφόρμες θα έχουν μεγαλύτερους περιορισμούς λόγω μειωμένης απόδοσης. Δείχνει λοιπόν μάλλον απαραίτητο ακόμα και σήμερα το να λαμβάνονται υπόψη εξαρχής θέματα απόδοσης κατά τη σχεδίαση και ανάπτυξη εικονικών κόσμων και να γίνονται από τα πρώτα στάδια δοκιμές μέσω πρωτοτύπων για τη διασφάλιση της εφικτότητας ανάπτυξης του συστήματος



Εικόνα 5.2 Μεθοδολογία σχεδίασης εικονικών περιβαλλόντων του C. Fencott.

3.2 Μεθοδολογία σχεδίασης εικονικών περιβαλλόντων (C. Fencott)

Λίγα χρόνια αργότερα, ο Fencott (1999) παρουσίασε μια μεθοδολογία για τη σχεδίαση εικονικών περιβαλλόντων, η οποία διακρίνει διάφορα επίπεδα μοντελοποίησης και προσδιορίζει τις μεταξύ τους σχέσεις. Σύμφωνα με τον Fencott, η σχεδίαση επιτυχημένων εικονικών περιβαλλόντων περιλαμβάνει δύο πολύ διαφορετικές μορφές σχεδίασης, οι οποίες συνεισφέρουν εξίσου στο αποτέλεσμα: την αισθητική, η οποία ασχολείται με την παραγωγή των επιθυμητών αντιληπτικών αποκρίσεων του χρήστη, και τη μηχανική, η οποία σχετίζεται με την κατασκευή των σχεδίων και μοντέλων βάσει των οποίων θα αναπτυχθεί το τελικό προϊόν. Η μεθοδολογία βασίζεται σε πέντε διακριτά αλλά αλληλεξαρτώμενα στάδια (εικ. 5.2).

Η *μοντελοποίηση απαιτήσεων* είναι μια διαδικασία απαραίτητη στα αρχικά στάδια ανάπτυξης μιας εφαρμογής και εμφανίζεται ως διακριτό βήμα σε όλες σχεδόν τις σύγχρονες μεθοδολογίες. Έχει ως αντικείμενο

τη μελέτη των απαιτήσεων της εφαρμογής βάσει των αναγκών των χρηστών και του προβλήματος που καλείται να λύσει, και τον προσδιορισμό των βασικών στόχων και προδιαγραφών της εφαρμογής.

Η *εννοιολογική μοντελοποίηση* είναι μια διεργασία που εκτελείται σε πολλά σχεδιαστικά έργα, ιδιαίτερα όταν δίνεται έμφαση στην αισθητική. Περιλαμβάνει τη συλλογή σχετικού υλικού, φωτογραφιών, σκίτσων, ήχων και βίντεο, ενώ μπορεί κατά τη διάρκειά της να προκύψουν και πίνακες διαθέσεων (moodboard) ή και πίνακες εξιστόρησης (storyboard). Σε αυτό το στάδιο, η ομάδα ανάπτυξης γνωρίζει και αποσαφηνίζει καλύτερα τον κόσμο που θα κατασκευαστεί. Η διεργασία αυτή έχει νόημα τόσο σε ρεαλιστικούς όσο και σε μη ρεαλιστικούς κόσμους, όπου οι καλλιτέχνες θα πρέπει να μπορούν να έχουν μια καλή εικόνα της αισθητικής του περιβάλλοντος.

Κατά την *αντιληπτική μοντελοποίηση* δημιουργείται ένα μοντέλο της φύσης των αντιληπτικών δυνατοτήτων του περιβάλλοντος και των μεταξύ τους σχέσεων. Επί της ουσίας μοντελοποιείται η επιθυμητή εμπειρία χρήστη που θέλουμε να δημιουργεί το περιβάλλον. Αν και είναι δύσκολο να σχεδιαστεί κάτι τόσο υποκειμενικό όπως η εμπειρία αλληλεπίδρασης ενός χρήστη με ένα διαδραστικό σύστημα, κατά το στάδιο αυτό μπορεί να προσδιοριστεί ο τρόπος παρουσίασης και ανάδειξης των στοιχείων του περιβάλλοντος, τα υποστηριζόμενα συμβάντα και οι αλληλεπιδράσεις του χρήστη με τον κόσμο, πιθανές ροές εξιστόρησης κ.ά.

Η *δομική μοντελοποίηση* έχει να κάνει με τον προσδιορισμό των στοιχείων υλοποίησης του κόσμου, όπως η επιλογή της πλατφόρμας ανάπτυξης, ο προσδιορισμός του μεγέθους και της διάταξης των στοιχείων του περιβάλλοντος, η κατασκευή των σκηνών κ.ά. Κατά το στάδιο αυτό η σχεδίαση εξειδικεύεται προοδευτικά καταλήγοντας σε λεπτομερή προσδιορισμό της συμπεριφοράς των αντικειμένων αλλά και της τελικής διαμόρφωσης του γραφήματος σκηνικού.

Το τελευταίο βήμα είναι η ανάπτυξη, η οποία οδηγεί στην τελική υλοποιημένη εφαρμογή σύμφωνα με τις αναλυτικές προδιαγραφές που έχουν προκύψει από την αντιληπτική και τη δομική μοντελοποίηση.

Η μεθοδολογία του Fencott εισάγει έντονα την αισθητική διάσταση στη σχεδίαση εικονικών κόσμων μέσα από τον προσδιορισμό δύο επιμέρους σταδίων: της εννοιολογικής και της αντιληπτικής μοντελοποίησης. Πράγματι, το γεγονός ότι οι χρήστες των εικονικών κόσμων τοποθετούνται μέσα σε ένα ρεαλιστικό ή φανταστικό περιβάλλον και καλούνται να το αντιληφθούν ως υπαρκτό και να αναγνωρίσουν τη συνύπαρξη και άλλων χρηστών μέσα σε αυτό (βλ. αίσθηση παρουσίας και συν-παρουσίας) καθιστά το αισθητικό στοιχείο ιδιαίτερα σημαντικό για τη διατήρηση της «ψευδαίσθησης». Σύμφωνα με τη συγκεκριμένη μεθοδολογία λοιπόν θα πρέπει η ομάδα σχεδίασης να προσδιορίσει τόσο τη γενικότερη αισθητική του κόσμου όσο και την επιθυμητή εμπειρία που θα ζουν οι επισκέπτες κατά τη διάρκεια της αλληλεπίδρασής τους με το περιβάλλον. Δυστυχώς, ο συγγραφέας αποφεύγει να προσδιορίσει με μεγαλύτερη λεπτομέρεια μεθόδους μοντελοποίησης της εμπειρίας των χρηστών σε εικονικά περιβάλλοντα, ενώ αναγνωρίζει και τη δυσκολία μετασχηματισμού των επιθυμητών εμπειριών των χρηστών σε δομικά μοντέλα του περιβάλλοντος. Σε κάθε περίπτωση, είναι σημαντικό οι σχεδιαστές εικονικών κόσμων να λαμβάνουν υπόψη και την οπτική της εμπειρίας του χρήστη.

3.3 VRID (V. Tanriverdi & R. Jacob)

Η μεθοδολογία VRID (Virtual Reality Interface Design) των Tanriverdi και Jacob (2001) εστιάζει στη σχεδίαση και ανάπτυξη της διεπαφής ενός περιβάλλοντος εικονικής πραγματικότητας. Η έμφαση στη συγκεκριμένη προσέγγιση δίνεται στη μοντελοποίηση της λειτουργίας των αντικειμένων διεπαφής μέσω μίας αρχιτεκτονικής που διακρίνει επιμέρους λειτουργικά τμήματα και προσδιορίζει τις αλληλεπιδράσεις μεταξύ τους.

Ένα σύστημα εικονικής πραγματικότητας, σύμφωνα με τους συγγραφείς, μπορεί να διαχωριστεί εννοιολογικά σε τρία διακριτά τμήματα: την εφαρμογή, η οποία περιλαμβάνει τα χαρακτηριστικά και τους κανόνες που καθορίζουν τη λογική του συστήματος, τη διεπαφή, που είναι το τμήμα με το οποίο αλληλεπιδρά ο χρήστης, και τον έλεγχο διαλόγου, που διαχειρίζεται την επικοινωνία μεταξύ των δύο τμημάτων. Η διεπαφή, σύμφωνα με τη συγκεκριμένη εννοιολογική μοντελοποίηση, αποτελείται από αντικείμενα διεπαφής και από δεδομένα εισόδου που προέρχονται από τους χρήστες ή από άλλες εξωτερικές πηγές. Η μεθοδολογία αφορά αποκλειστικά τη σχεδίαση της διεπαφής του συστήματος.

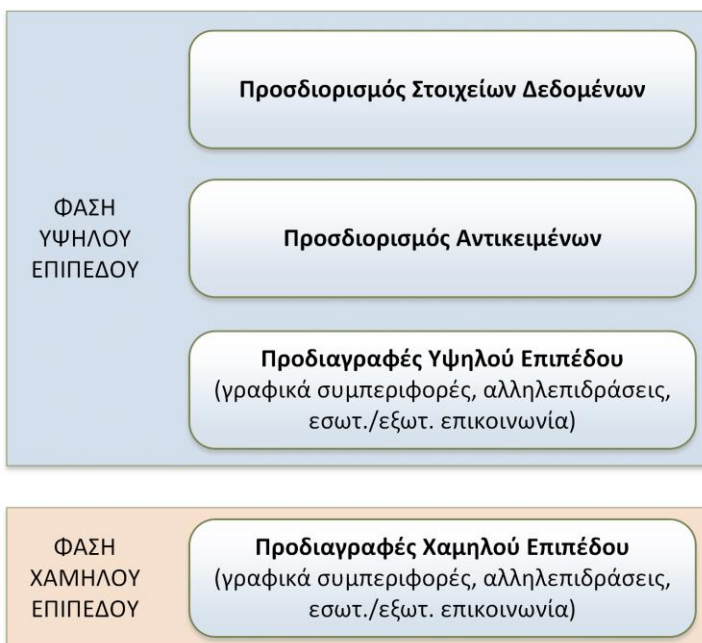
Σύμφωνα με το μοντέλο VRID, η αρχιτεκτονική ενός αντικειμένου διεπαφής περιλαμβάνει τα παρακάτω λειτουργικά τμήματα:

- Το *τμήμα γραφικών*, που προσδιορίζει τις διάφορες μορφές γραφικής αναπαράστασης του αντικειμένου. Συμπεριλαμβάνει και ενεργοποιεί όλες τις εν δυνάμει γραφικές απεικονίσεις και συνθετικές κινήσεις του αντικειμένου κατά τη διάρκεια εκτέλεσης της εφαρμογής.

- Το *τμήμα συμπεριφοράς*, που προσδιορίζει τα διάφορα είδη συμπεριφοράς του αντικειμένου. Οι εν δυνάμει συμπεριφορές των αντικειμένων κατηγοριοποιούνται σε δύο ομάδες: τις φυσικές συμπεριφορές, δηλαδή αυτές που είναι παρατηρήσιμες στον φυσικό κόσμο, και τις «μαγικές» συμπεριφορές, δηλαδή αλλαγές στην κατάσταση ενός αντικειμένου που δεν είναι ρεαλιστικές. Σε πολλές περιπτώσεις τα αντικείμενα μπορεί να εκτελούν σύνθετες συμπεριφορές που αποτελούνται από μια ακολουθία απλούστερων φυσικών ή και μαγικών συμπεριφορών.
- Το *τμήμα αλληλεπίδρασης*, που ερμηνεύει τα δεδομένα εισόδου ανάλογα με την πηγή τους και την κατάσταση του αντικειμένου και αποφασίζει για την επίδραση που θα έχουν στη συμπεριφορά του αντικειμένου
- Το *τμήμα επικοινωνίας*, που χρησιμοποιείται για τη λήψη και αποστολή δεδομένων από και προς άλλα αντικείμενα ή εξωτερικές πηγές.
- Το *τμήμα διαμεσολάβησης*, που αναλαμβάνει τον έλεγχο και συντονισμό όλων των υπόλοιπων τμημάτων. Συγκεντρώνει τα μηνύματα των υπόλοιπων τμημάτων, τα διαχειρίζεται και τα προωθεί στους κατάλληλους παραλήπτες.

Σύμφωνα με τη μεθοδολογία VRID, η σχεδίαση της διεπαφής είναι μια επαναληπτική διαδικασία στην οποία οι απαιτήσεις της διεπαφής μεταφράζονται σε σχεδιαστικές προδιαγραφές προς υλοποίηση. Η διαδικασία μοιράζεται σε δύο στάδια, ένα υψηλού και ένα χαμηλού επιπέδου. Στο στάδιο υψηλού επιπέδου ο στόχος είναι ο προσδιορισμός μιας σχεδιαστικής λύσης σε υψηλό βαθμό αφαίρεσης χρησιμοποιώντας την αρχιτεκτονική πολλαπλών τμημάτων. Το αποτέλεσμα είναι η αρχική αναπαράσταση των δεδομένων και αντικειμένων της διεπαφής. Γραφικά μοντέλα, συμπεριφορές, αλληλεπιδράσεις και επικοινωνίες εντός και εκτός του αντικειμένου προσδιορίζονται σε ένα αφαιρετικό επίπεδο. Στο επόμενο στάδιο στόχος είναι να προκύψουν αναλυτικές προδιαγραφές των διάφορων τμημάτων των αντικειμένων βάσει των αφηρημένων περιγραφών τους. Η διαδικασία δεν είναι κατ' ανάγκη γραμμική. Μπορεί να προκύψει ανάγκη επαναλήψεων και διορθώσεων και στα δύο επίπεδα αφαίρεσης μέχρι η ομάδα σχεδίασης να καταλήξει σε ένα ικανοποιητικό και υλοποιήσιμο αποτέλεσμα.

Τα βήματα της μεθοδολογίας περιγράφονται στην εικόνα 5.3.



Εικόνα 5.3 Η μεθοδολογία VRID.

Το ενδιαφέρον στοιχείο στη μεθοδολογία VRID έγκειται στη μοντελοποίηση της συμπεριφοράς των αντικειμένων διεπαφής ως ένα σύνολο λειτουργικών τμημάτων που αλληλεπιδρούν μεταξύ τους. Η διάκριση μεταξύ απεικόνισης, αλληλεπίδρασης, συμπεριφοράς και επικοινωνίας καθώς και ο προσδιορισμός σύνθετων συμπεριφορών ως ακολουθίες απλών μπορούν να αποτελέσουν χρήσιμα βοηθήματα τόσο για τη σχεδίαση των

αντικειμένων ενός εικονικού κόσμου όσο και για την απλοποίηση της ανάπτυξης του τελικού κώδικα της εφαρμογής. Τέλος, θα πρέπει να σημειωθεί ότι τα αντικείμενα διεπαφής του μοντέλου VRID αναφέρονται σε οποιοδήποτε αντικείμενο επιδεικνύει δυναμική συμπεριφορά εντός ενός εικονικού κόσμου και όχι μόνο σε τυπικά στοιχεία ελέγχου (π.χ. εικονικά χειριστήρια ή στοιχεία του HUD).

4 Αναπαράσταση του κόσμου και αλληλεπιδράσεις χρήστη

Δύο βασικές διαδικασίες κατά τη διάρκεια ανάπτυξης ενός εικονικού κόσμου είναι η αναπαράσταση του περιβάλλοντος και η σχεδίαση των αλληλεπιδράσεων χρήστη. Το αποτέλεσμα των δύο αυτών διαδικασιών προσδιορίζει το συνολικό περιβάλλον διεπαφής του χρήστη με την εφαρμογή, δηλαδή τη μορφή με την οποία απεικονίζονται το περιβάλλον και τα δρώμενα εντός αυτού, και τις δυνατές ενέργειες που μπορεί να εκτελέσει ο χρήστης μέσα στον κόσμο. Στα κεφάλαια 2, 3 και 4 έχουμε αναφερθεί αναλυτικά στην τεχνολογία πίσω από την απεικόνιση του περιβάλλοντος, την κίνηση των αντικειμένων και τις αλληλεπιδράσεις του χρήστη με τον κόσμο, και έχουμε παρουσιάσει διάφορες συχνά χρησιμοποιούμενες τεχνικές σε επιμέρους πτυχές των παραπάνω. Στην παρούσα ενότητα θα εξετάσουμε σχεδιαστικά ζητήματα που αφορούν την επιλογή κατάλληλων μορφών αναπαράστασης και δυνατοτήτων αλληλεπίδρασης και σχετίζονται με το απευθυνόμενο κοινό και το είδος της εφαρμογής.

4.1 Επιλογή κατάλληλων αναπαραστάσεων

Αναπαράσταση είναι η απεικόνιση σκέψεων, ιδεών και δεδομένων σε κάποιο μέσο με στόχο την επικοινωνία τους σε τρίτους. Μια ιδέα δεν μπορεί να μεταφερθεί απευθείας από το μυαλό του δημιουργού στο μυαλό του αποδέκτη, αλλά χρειάζεται κάποιο μέσο για να τη μεταφέρει. Επομένως, είναι αναγκαίο η ιδέα να μπορεί να απεικονιστεί ικανοποιητικά στο μέσο και ο παραλήπτης να μπορέσει να την αντιληφθεί με τρόπο τέτοιο ώστε να επιτευχθεί μια αμοιβαία κατανόηση. Υπάρχουν, βεβαίως, πολλές μορφές αναπαράστασης και δεν υπάρχει υποχρεωτικά μία μορφή που είναι η καλύτερη, απλά κάποιες είναι περισσότερο κατάλληλες από κάποιες άλλες, ανάλογα με την περίπτωση. Είναι δυνατόν για τα ίδια δεδομένα να μεταβάλλεται η καταλληλότητα των αναπαραστάσεων σε σχέση με τους στόχους της εφαρμογής. Για παράδειγμα, η δραματική εξιστόρηση ενός φαινομένου μπορεί να είναι προτιμότερη μορφή αναπαράστασης σε εφαρμογές με στόχο τη συναισθηματική εμπλοκή του χρήστη, για παράδειγμα σε παιχνίδια. Αντίθετα, σε εφαρμογές οι οποίες στοχεύουν στη μελέτη και ανάλυση των δεδομένων καταλληλότερο μέσο μπορεί να είναι η επιστημονική οπτικοποίηση του ίδιου φαινομένου μέσω χρωματισμένων επιφανειών, αριθμητικών δεδομένων κ.λπ.

Στην περίπτωση των εικονικών κόσμων η επιλογή κατάλληλων αναπαραστάσεων είναι ιδιαίτερα κρίσιμη σχεδιαστική φάση που επηρεάζει τόσο την εμπειρία του χρήστη όσο και την απόδοση της εφαρμογής. Σε αντίθεση με άλλες εφαρμογές που χρειάζεται να επιλέξουμε μία ή περισσότερες μορφές αναπαράστασης ενός συνόλου δεδομένων, εδώ οι αναπαραστάσεις αφορούν έναν ολόκληρο κόσμο: ένα περιβάλλον το οποίο ο χρήστης πρέπει να αντιληφθεί ως «υπαρκτό» και να φανταστεί τον εαυτό του ως μέρος του. Τα δεδομένα που θα πρέπει να επικοινωνούνται σχετίζονται με τον τρισδιάστατο χώρο, τα αντικείμενά του, τους χαρακτήρες που ζουν και αλληλεπιδρούν μέσα σε αυτόν, τη ροή των συμβάντων κ.λπ. Επιπλέον, το μέσο υποστηρίζει πληθώρα διαφορετικών μορφών αναπαράστασης, όπως κείμενο, ήχο, βίντεο κ.λπ., οι οποίες θα πρέπει να επιλεγθούν και να συνδυαστούν κατάλληλα για την επίτευξη των στόχων της εφαρμογής. Τέλος, η κάθε μορφή αναπαράστασης έχει τις δικές της υπολογιστικές απαιτήσεις, με αποτέλεσμα η επιλογή να επηρεάζει τη συνολική απόδοση του συστήματος.

Ας εξετάσουμε ένα σχετικό παράδειγμα: Έστω ότι θέλουμε σε έναν εικονικό κόσμο να αναπαραστήσουμε τη συναισθηματική κατάσταση των χαρακτήρων, για παράδειγμα να είναι εμφανές πότε είναι χαρούμενοι, λυπημένοι, εκνευρισμένοι κ.λπ. Μια προφανής επιλογή αναπαράστασης είναι η χρήση εκφράσεων προσώπου που μιμούνται την πραγματικότητα, έτσι ώστε το συναίσθημα να είναι οπτικά αναγνωρίσιμο. Η επιλογή αυτή είναι εφικτή με τις σημερινές δυνατότητες των υπολογιστών και σίγουρα θα οδηγούσε σε πιο φυσικά, ρεαλιστικά αποτελέσματα. Όμως, η επιβάρυνση στην απόδοση μπορεί να είναι σημαντική, ιδιαίτερα σε σκηνές οι οποίες περιλαμβάνουν μεγάλο αριθμό χαρακτήρων. Επιπλέον, η αναγνώριση της συναισθηματικής κατάστασης θα είναι αρκετά πιο δύσκολη αν ο χαρακτήρας βρίσκεται σε μεγάλη απόσταση και δεν είναι πολύ ξεκάθαρα τα χαρακτηριστικά του προσώπου του. Κάτι τέτοιο μπορεί να είναι πρόβλημα, για παράδειγμα, σε μια εφαρμογή προσομοίωσης στην οποία θέλουμε να παρατηρήσουμε τη συμπεριφορά πλήθους σε μία κατάσταση έκτακτης ανάγκης και να εντοπίσουμε εύκολα τους χαρακτήρες που

βρίσκονται σε πανικό. Είναι λοιπόν δυνατόν σε κάποιες εφαρμογές να είναι προτιμότερη η χρήση εναλλακτικών αναπαραστάσεων, παρά το γεγονός ότι περιορίζεται η ρεαλιστικότητα του αποτελέσματος. Τέτοιες αναπαραστάσεις στο παράδειγμά μας θα μπορούσαν να είναι η χρήση χρώματος, κειμένου ή συμβόλου για την κατάδειξη της συναισθηματικής κατάστασης.

4.1.1 Βαθμός ρεαλισμού

Όπως προκύπτει από το παράδειγμά μας, υπάρχουν περιπτώσεις στις οποίες η ρεαλιστική αναπαράσταση δεν είναι πάντα η καταλληλότερη. Άλλωστε συχνά και οι ίδιοι οι κόσμοι δεν μιμούνται την πραγματικότητα αλλά παρουσιάζουν φανταστικά περιβάλλοντα, όπως για παράδειγμα στα παιχνίδια. Αυτό που έχει σημασία είναι η εσωτερική συνέπεια του κόσμου, δηλαδή οι χώροι και τα συμβάντα που έχουμε αντιληφθεί, αλλά και όσα υπονοούνται από την κατάσταση και εξέλιξη του περιβάλλοντος, να είναι μεταξύ τους συμβατά (Sherman & Craig, 2003). Οι άνθρωποι έχουμε την ικανότητα να συμπληρώνουμε μεμονωμένα συμβάντα δημιουργώντας μια ιστορία από αυτά, ένα χαρακτηριστικό που ονομάζεται «κλειστότητα» (*closure*). Αν ο κόσμος είναι συνεπής και επιτρέπει την αντίληψή του ως ένα υπαρκτό περιβάλλον, τότε ενισχύεται σημαντικά η αίσθηση της παρουσίας των χρηστών σε αυτόν. Είναι επίσης γεγονός ότι οι άνθρωποι έχουμε την ικανότητα να γενικεύουμε πολύ εύκολα τα αντικείμενα που αντιλαμβανόμαστε. Αν, για παράδειγμα, δούμε ένα μοντέλο αυτοκινήτου που δεν έχουμε ξαναδεί ποτέ, μπορούμε εύκολα να αντιληφθούμε ότι πρόκειται για αυτοκίνητο. Επιπλέον μπορούμε να αναγνωρίζουμε αφηρημένες μορφές και σύμβολα, εάν αυτά είναι κατάλληλα σχεδιασμένα ώστε να επικοινωνούν χαρακτηριστικές ιδιότητες του αντικειμένου. Αυτό δίνει στους σχεδιαστές τη δυνατότητα να χρησιμοποιούν περισσότερο απλοποιημένες μορφές χωρίς να παρερμηνεύεται το μήνυμα που θέλουν να μεταφέρουν. Οι μορφές αναπαράστασης που μπορούμε να επιλέξουμε για τα διάφορα στοιχεία του περιβάλλοντος ενδέχεται να ποικίλλουν ανάλογα με τον βαθμό ρεαλισμού. Μια κατηγοριοποίηση είναι η παρακάτω:

- *ρεαλιστική*: όσο το δυνατόν πιο κοντά στην πραγματικότητα,
- *δεικτοδοτημένη*: απεικόνιση τιμών ενός φαινομένου σε μια μορφή πιο εύκολα κατανοήσιμη,
- *εικονοδιακή*: χρήση απλοποιημένων μορφών,
- *συμβολική*: απεικόνιση σε μορφές που δεν θυμίζουν το αρχικό αντικείμενο, αλλά είναι εύκολα αναγνωρίσιμες.

Στους εικονικούς κόσμους, η επιλογή λιγότερο ρεαλιστικών αναπαραστάσεων μπορεί να είναι προτιμότερη όχι μόνο για λόγους απόδοσης αλλά και λόγω των απαιτήσεων της εφαρμογής. Η απλοποιημένη παρουσίαση ενός αντικειμένου ή συστήματος το αποσυνδέει από λεπτομέρειες και διευκολύνει την καλύτερη κατανόηση της μορφής ή και της δομής του, κάτι που μπορεί να είναι χρήσιμο σε εκπαιδευτικές εφαρμογές. Η χρήση ειδικών τεχνικών οπτικοποίησης μπορεί να βελτιώσει ακόμα περισσότερο την παρουσίαση αντικειμένων ή συλλογής δεδομένων. Για παράδειγμα, τα παχιά περιγράμματα προσφέρουν οπτικό διαχωρισμό επιμέρους τμημάτων, οι ημιδιαφάνειες μπορούν να απεικονίσουν πολλαπλές επιφάνειες του ίδιου αντικειμένου (π.χ. ανθρώπινο σώμα) κ.ά. Μία άλλη περίπτωση είναι η χρήση απλοποιημένων αναπαραστάσεων σε αντικείμενα χαμηλότερου ενδιαφέροντος. Για παράδειγμα, σε εικονικά μουσεία συνήθως επιθυμούμε το ενδιαφέρον των επισκεπτών να επικεντρώνεται στα ίδια τα εκθέματα, οπότε μπορεί να επιλέξουμε λιγότερο ρεαλιστικές αναπαραστάσεις για τις προθήκες και το κτίριο.

Επιπλέον, σε κάποιες περιπτώσεις μπορεί να είναι επιθυμητή και η χρήση εναλλακτικών μορφών οπτικής αναπαράστασης πέρα από τα τρισδιάστατα αντικείμενα, είτε συμπληρωματικά είτε σε αντικατάσταση αυτών. Το κείμενο μπορεί να επαυξήσει χώρους, αντικείμενα ή χρήστες με επιπλέον πληροφορίες. Για παράδειγμα, η εμφάνιση των ονομάτων των ενσαρκώσεων διευκολύνει την αναγνώριση και την επικοινωνία σε περιβάλλοντα με πολλούς χρήστες. Οι φωτογραφίες, αν και στερούνται μιας διάστασης, μπορούν να προσφέρουν πολύ πιο «ζωντανές» απεικονίσεις του φυσικού κόσμου χωρίς υπολογιστική επιβάρυνση. Η χρήση ζωντανού βίντεο μπορεί να επαυξήσει μια παρουσίαση με στοιχεία μη λεκτικής επικοινωνίας. Τα εικονίδια μπορούν να μεταφέρουν πληροφορίες σχετικά με τις δυνατότητες αλληλεπίδρασης του χρήστη με ένα αντικείμενο του περιβάλλοντος, για παράδειγμα, όταν ο δείκτης του ποντικιού τοποθετείται πάνω από μια κλειστή πόρτα εμφανίζεται ένα εικονίδιο που δείχνει την πόρτα να ανοίγει, υπονοώντας ότι αυτό θα είναι το αποτέλεσμα του κλικ πάνω της.

4.1.2 Ο ρόλος του ήχου

Πέρα από την εικόνα, πολύ σημαντικός είναι και ο ρόλος του ήχου στους εικονικούς κόσμους. Ο ήχος μπορεί να αποτελέσει ένα επιπλέον κανάλι αναπαράστασης και μετάδοσης πληροφοριών και μπορεί να αξιοποιηθεί με πολλαπλούς τρόπους στο περιβάλλον. Για παράδειγμα, οι VanTol & Huijberts (2008) προτείνουν ένα πλαίσιο για την κατηγοριοποίηση των ήχων στα παιχνίδια, το οποίο βασίζεται σε δύο άξονες. Ο ένας άξονας είναι κατά πόσο ο ήχος σχετίζεται με το σκηνικό ή με τη δραστηριότητα του παίκτη και ο άλλος άξονας κατά πόσο ο ήχος είναι μέρος του κόσμου ή εκτός αυτού. Με βάση αυτή την κατηγοριοποίηση διακρίνουν τέσσερα είδη ήχου:

Η πρώτη κατηγορία είναι οι *ήχοι του περιβάλλοντος (zone)*, οι οποίοι είναι χαρακτηριστικοί διάχυτοι ήχοι των διάφορων περιοχών του κόσμου, για παράδειγμα εργοστάσιο, αγορά κ.λπ. Οι άνθρωποι έχουμε μια πολυαισθητηριακή εμπειρία στον φυσικό κόσμο, έχουμε συνηθίσει να συνδυάζουμε χώρους με αντίστοιχους ήχους και οσμές. Συνεπώς, η επαύξηση αντίστοιχων χώρων σε ένα εικονικό περιβάλλον με τυπικούς διάχυτους ήχους επιδρά θετικά στην παρουσία των χρηστών.

Η δεύτερη κατηγορία είναι τα *ειδικά εφέ* που σχετίζονται με συγκεκριμένες δραστηριότητες εντός του χώρου, για παράδειγμα το άνοιγμα μιας πόρτας, η εκπυροκρότηση ενός πιστολιού κ.λπ. Η ήχοι αυτοί βοηθούν τον χρήστη να αναγνωρίσει πιο εύκολα τις ενέργειες που λαμβάνουν χώρα στο περιβάλλον και να τους δώσει την απαιτούμενη προσοχή. Επιπλέον, τον βοηθούν να αντιληφθεί και ενέργειες που εκτελούνται εκτός του οπτικού του πεδίου, για παράδειγμα κάποιος που βαδίζει πίσω από τον χρήστη.

Στην τρίτη κατηγορία βρίσκονται οι *ήχοι συναισθήματος*, δηλαδή ήχοι και μελωδίες που δίνουν τον συναισθηματικό τόνο ή τη γενικότερη διάθεση που σχετίζεται με ένα περιβάλλον, μια δραστηριότητα ή τη ροή μιας ιστορίας. Η χρήση αυτών των ήχων έχει περισσότερο νόημα σε περιβάλλοντα στα οποία στόχος είναι, μεταξύ άλλων, και η συναισθηματική εμπλοκή του χρήστη, όπως τα παιχνίδια και οι εξιστορήσεις.

Η τελευταία κατηγορία είναι οι *ήχοι διεπαφής*, δηλαδή ηχητικά μηνύματα που σχετίζονται με τη διεπαφή του χρήστη με το περιβάλλον, όπως για παράδειγμα το πάτημα ενός πλήκτρου, η ενημέρωση για την ολοκλήρωση μιας διεργασίας κ.λπ. Οι ήχοι αυτοί μπορούν να αποτελέσουν ένα σημαντικό συμπληρωματικό κανάλι της εικόνας, ώστε ο χρήστης να μπορεί να επαληθεύσει την εκτέλεση κάποιας ενέργειας ή να ενημερωθεί για κάποια αλλαγή στην κατάσταση του περιβάλλοντος.

Εκτός από τις παραπάνω κατηγορίες, ο ήχος έχει χρησιμοποιηθεί σε περιβάλλοντα εικονικής πραγματικότητας και για την *ηχοποίηση (sonification)* αφηρημένων δεδομένων. Για παράδειγμα, οι μεταβολές στη θερμοκρασία ενός χώρου μπορεί να επηρεάζουν την ένταση ή τον τόνο ενός συνεχούς ήχου ανάλογα με την τρέχουσα θέση του χρήστη, παρέχοντας με αυτόν τον τρόπο ένα επιπλέον κανάλι πληροφορίας πέρα από το οπτικό.

4.1.3 Αντικατάσταση αισθήσεων

Μία ακόμα προσέγγιση η οποία χρησιμοποιείται συχνά σε αναπαραστάσεις περιβαλλόντων είναι η *αντικατάσταση αισθήσεων (sensory substitution)*. Η ιδέα είναι να αντικαταστήσουμε πληροφορίες από αισθήσεις που δεν μπορούμε να πάρουμε λόγω περιορισμών του συστήματος με αναπαραστάσεις άλλης μορφής, οι οποίες είναι εύκολα αναγνωρίσιμες. Όταν συνδεόμαστε σε έναν εικονικό κόσμο μέσω κάποιου τυπικού οικιακού υπολογιστή, το περιβάλλον δεν μπορεί να μας μεταφέρει την αίσθηση της επαφής ή να δημιουργήσει οσμές. Επιπλέον, δεν έχουμε περιφερειακή όραση και η οπτική απόδοση του κόσμου είναι σίγουρα χαμηλότερης ποιότητας και μικρότερη σε μέγεθος σε σχέση με τον πραγματικό κόσμο. Όλες αυτές οι πληροφορίες που «χάνονται» σε ένα τέτοιο σύστημα μπορούν να αναπαρασταθούν με κατάλληλες εναλλακτικές μορφές. Για παράδειγμα, όταν συγκρουόμαστε με αντικείμενα του περιβάλλοντος, μπορεί να ακούγεται ένας χαρακτηριστικός ήχος σύγκρουσης. Συγκεκριμένες μυρωδιές μπορούν να αντικατασταθούν με την παρουσίαση χρωματισμένων αέριων μαζών ή με τη χρήση αναγνωριστικών συμβόλων. Αν υπάρχει κάποιο αντικείμενο ή χαρακτήρας ειδικού ενδιαφέροντος δίπλα μας, μπορεί να καταδεικνύεται με οπτικά σημάδια (π.χ. βέλος και συνοδευτικό κείμενο). Τέλος, ένα χαρακτηριστικό παράδειγμα αντικατάστασης αισθήσεων είναι η περίπτωση όπου δεχόμαστε ένα χτύπημα σε παιχνίδια πρώτου προσώπου και συνήθως επικοινωνείται η αίσθηση του πόνου κοκκινίζοντας την οθόνη.

4.2 Σχεδιαστικές κατευθύνσεις με βάση τις δραστηριότητες

Όπως έχει ήδη αναφερθεί, οι σχεδιαστικές αποφάσεις ως προς τις αναπαραστάσεις του κόσμου, των αντικειμένων και των ενσάρκώσεων αλλά και ως προς τις υποστηριζόμενες αλληλεπιδράσεις σχετίζονται σε μεγάλο βαθμό με το απευθυνόμενο κοινό και τις απαιτήσεις της εφαρμογής. Για παράδειγμα, ο βαθμός ρεαλισμού και η πολυπλοκότητα των αλληλεπιδράσεων μπορεί να διαφέρουν σημαντικά σε μια εφαρμογή εξάσκησης σε σχέση με ένα παιχνίδι. Αντίστοιχα, τα υποστηριζόμενα μοντέλα πλοήγησης και τα σχετικά βοηθήματα μπορεί επίσης να είναι αρκετά διαφορετικά μεταξύ ενός εικονικού μουσείου και μιας εφαρμογής συνεργατικής σχεδίασης χώρων. Τόσο η απεικόνιση όσο και η αλληλεπίδραση ενός κόσμου θα πρέπει να είναι σχεδιασμένες έχοντας ως βασικό γνώμονα την ικανοποίηση των στόχων των χρηστών και, όπως υπονοείται από τα παραδείγματά μας, οι στόχοι μπορεί να διαφέρουν σημαντικά από εφαρμογή σε εφαρμογή. Στο περιβάλλον εξάσκησης ο χρήστης θέλει να βρεθεί σε έναν χώρο όσο το δυνατόν πιο κοντά στον αντίστοιχο πραγματικό και να βελτιώσει τις δεξιότητές του μέσα από την αλληλεπίδρασή του με αυτόν. Αντίθετα, σε ένα παιχνίδι ο χρήστης θέλει να περάσει ευχάριστα τον χρόνο του σε ένα φανταστικό περιβάλλον αντιμετωπίζοντας προκλήσεις κλιμακούμενης δυσκολίας. Σε μια εφαρμογή σχεδίασης χώρων οι συνεργαζόμενοι χρήστες θέλουν να μπορούν να αλλάζουν εύκολα και γρήγορα οπτική για την καλύτερη τοποθέτηση των αντικειμένων και επισκόπηση του αποτελέσματος, ενώ σε ένα εικονικό μουσείο ο χρήστης πιθανότατα περιμένει να έχει μια εμπειρία περιήγησης παραπλήσια με τον φυσικό κόσμο.

Οι δυνατοί χώροι εφαρμογής των εικονικών κόσμων είναι αρκετοί (βλ. Κεφ. 1). Μπορούμε ωστόσο να διακρίνουμε ορισμένες βασικές κατηγορίες δραστηριοτήτων χρήστη που φαίνεται να εμφανίζονται συχνά σε διάφορες παραλλαγές στις εφαρμογές αυτές (Barrilleaux, 2001· Bowman κ.ά., 2003· Benford κ.ά., 2001). Οι κατηγορίες είναι:

- *Σχεδίαση*: ο χρήστης αξιοποιεί το περιβάλλον για τη σχεδίαση ενός προϊόντος, συστήματος ή υπηρεσίας.
- *Συμμετοχή σε προσομοιωμένο περιβάλλον*: ο χρήστης ενσωματώνεται και συμμετέχει σε ένα περιβάλλον που προσομοιώνει τη λειτουργία ενός πραγματικού, υποθετικού ή φανταστικού συστήματος.
- *Εξερεύνηση δεδομένων και πληροφοριών*: εξωτερικά δεδομένα παρουσιάζονται στον τρισδιάστατο χώρο και απεικονίζονται κατάλληλα ώστε να πληροφορήσουν ή να οδηγήσουν στην εξαγωγή συμπερασμάτων.
- *Συνεργασία*: πολλαπλοί χρήστες συνδιαλέγονται και συνεργάζονται στα πλαίσια κάποιας κοινής δραστηριότητας.

Οι παραπάνω κατηγορίες εμφανίζουν σημαντικές διαφορές ως προς τους στόχους τους και την αναμενόμενη υποστήριξη των χρηστών. Θα εξετάσουμε στη συνέχεια καθεμία από αυτές αναλυτικότερα, ώστε να προκύψουν χρήσιμες κατευθύνσεις για τους τρόπους αξιοποίησης του τρισδιάστατου περιβάλλοντος και για τη σχεδίαση κατάλληλων διαδράσεων σε αυτές. Οι κατηγορίες θα εξεταστούν ως προς: α) το τι θέλουμε να πετύχει ο χρήστης μέσα από τη χρήση τους, β) τη φύση των δεδομένων της κάθε δραστηριότητας, γ) κατάλληλους τρόπους αναπαράστασης των δεδομένων στο περιβάλλον και δ) κατάλληλες υποστηριζόμενες αλληλεπιδράσεις.

Θα πρέπει να σημειώσουμε ότι σε μια εφαρμογή εικονικών κόσμων μπορεί να υποστηρίζονται παραπάνω από μία από τις προαναφερθείσες δραστηριότητες. Για παράδειγμα, μια εφαρμογή αρχιτεκτονικής σχεδίασης βασίζεται κυρίως σε σχεδιαστικές διεργασίες για τον καθορισμό και τη διαμόρφωση του χώρου. Όμως, όταν γίνεται επισκόπηση των σχεδιασμένων χώρων από τους σχεδιαστές ή και από πελάτες, τότε στόχος είναι η προσομοίωση του περιβάλλοντος ώστε να αποκτήσουν οι συμμετέχοντες μία όσο το δυνατόν πιο «ζωντανή» και φυσική εμπειρία περιήγησης.

4.2.1 Σχεδίαση

Στις δραστηριότητες σχεδίασης, ο χρήστης χειρίζεται τον τρισδιάστατο χώρο για να σχεδιάσει αντικείμενα, συστήματα ή περιβάλλοντα, βάσει συγκεκριμένων προδιαγραφών, απαιτήσεων και περιορισμών. Πιθανοί χώροι εφαρμογής είναι η αρχιτεκτονική, η μηχανολογία, η σχεδίαση προϊόντων, η σχεδίαση υπηρεσιών κ.λπ.

Οι χαρακτηριστικές ενέργειες που εκτελεί ένας χρήστης είναι η προσθήκη, η αφαίρεση, η ομαδοποίηση και η τροποποίηση αντικειμένων με στόχο τη διαμόρφωση νέων αντικειμένων, συστημάτων ή περιβαλλόντων.

Τα δεδομένα σε τέτοιου είδους δραστηριότητες είναι συχνά «συμπαγή» και αναπαριστούν σε μικρό ή μεγάλο βαθμό ρεαλισμού φυσικά αντικείμενα και συστήματα. Για παράδειγμα, αντικείμενα σχεδίασης μπορεί να είναι έπιπλα, αυτοκίνητα, δίκτυα επικοινωνιών, εργοστάσια κ.λπ. Υπάρχουν δύο βασικές παραλλαγές στις σχεδιαστικές δραστηριότητες, ανάλογα με το αντικείμενο της σχεδίασης: η *γεωμετρική σχεδίαση*, όπου οι σχετικές θέσεις και οι προσανατολισμοί των αντικειμένων παίζουν ρόλο στο τελικό αποτέλεσμα, και η *τοπολογική σχεδίαση*, όπου μόνο οι συνδέσεις μεταξύ των αντικειμένων έχουν σημασία, ανεξαρτήτως της θέσης στην οποία παρουσιάζονται. Παράδειγμα της πρώτης κατηγορίας είναι η σχεδίαση εσωτερικών χώρων, ενώ της δεύτερης η σχεδίαση ενός δικτύου υπολογιστών.

Σε ό,τι αφορά τις υποστηριζόμενες αλληλεπιδράσεις με τον χρήστη, κατά τη σχεδίαση η έμφαση θα πρέπει να δίνεται κυρίως στον χειρισμό των αντικειμένων, καθώς οι ενέργειες του χρήστη σχετίζονται με τη δημιουργία, την τοποθέτηση και την τροποποίηση αντικειμένων. Επίσης, κρίσιμο ρόλο στη διαδικασία παίζει και η πλοήγηση, ώστε να είναι σε θέση ο σχεδιαστής εύκολα και γρήγορα να ρυθμίζει την άποψή του σε σχέση με το αντικείμενο της σχεδίασης. Οι τεχνικές πλοήγησης στις εφαρμογές αυτές δεν είναι απαραίτητο να είναι εντελώς ρεαλιστικές, για παράδειγμα φυσικό βάδισμα. Αντίθετα, πολλές φορές για λόγους ευκολίας, όπως στην περίπτωση της δυνατότητας γρήγορης αλλαγής οπτικής, είναι σκόπιμο ο ρεαλισμός να παρακάμπτεται. Θα πρέπει να υπάρχουν επίσης εξειδικευμένες τεχνικές πρόσβασης σε δεδομένα, καθώς στις περισσότερες κατηγορίες σχεδίασης, πέρα από την τρισδιάστατη γεωμετρία, υπάρχουν και συγκεκριμένα δεδομένα που συνοδεύουν το αντικείμενο σχεδίασης, όπως εξειδικευμένα χαρακτηριστικά, περιορισμοί κ.ά.

Για την αναπαράσταση του περιβάλλοντος δεν απαιτούνται συνήθως εξειδικευμένες τεχνικές οπτικοποίησης, καθώς η ποιοτική οπτική απόδοση του χώρου δεν είναι το βασικό ζητούμενο. Αντίθετα, θα πρέπει να παρέχονται τα κατάλληλα οπτικά βοηθήματα, ώστε να υποστηρίξουν τον σχεδιαστή στη σύνθεση προκαταρκτικών ιδεών, στην επισκόπηση και στην τροποποίηση των αποτελεσμάτων. Οι σκιές μπορούν να βοηθήσουν στην καλύτερη κατανόηση της σχετικής θέσης των αντικειμένων στον χώρο και κατά συνέπεια στη σωστότερη τοποθέτησή τους. Απεικονίσεις με ημιδιαφάνειες και πλέγματα (wireframes) μπορούν να διευκολύνουν την οπτική του σχεδιαστή παρέχοντάς του τη δυνατότητα να δει και μέσα ή πίσω από αντικείμενα. Τέλος, άλλα οπτικά βοηθήματα, όπως μετρήσεις αποστάσεων μεταξύ αντικειμένων, ειδικές αναπαραστάσεις που καταδεικνύουν την παραβίαση κάποιων περιορισμών κ.ά., μπορούν να ενισχύσουν ακόμα περισσότερο τη διαδικασία.

4.2.2 Συμμετοχή σε προσομοιωμένο περιβάλλον

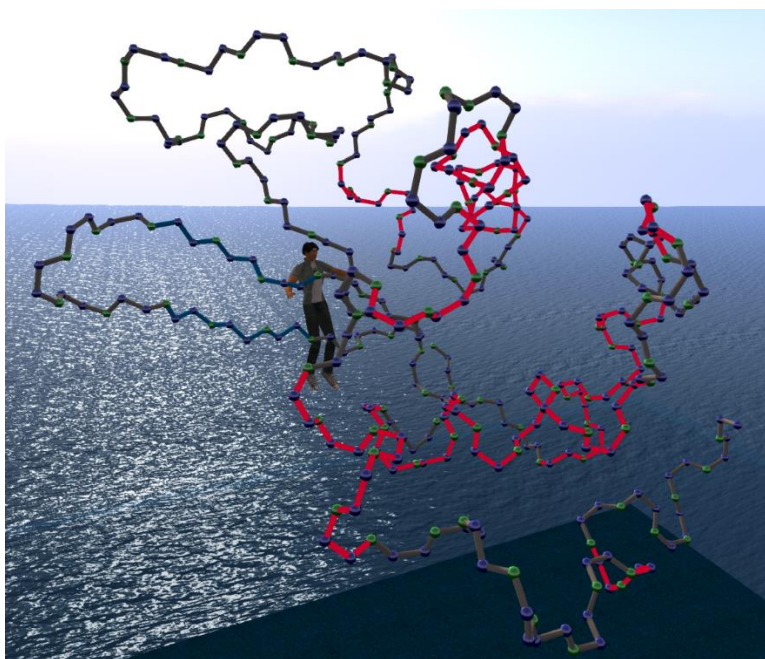
Κατά τη συμμετοχή τους σε ένα προσομοιωμένο περιβάλλον οι χρήστες αποκτούν εμπειρία μιας συγκεκριμένης κατάστασης ή περιβάλλοντος, φανταστικού, υποθετικού ή πραγματικού, μέσω διαδικασιών που εκτελούνται σε πραγματικό χρόνο και αναπαριστούν τη λειτουργία του. Το ιδιαίτερο χαρακτηριστικό σε αυτή την περίπτωση είναι ότι ο χρήστης αλληλεπιδρά και πλοηγείται στον χώρο με τρόπο που συχνά μιμείται τον πραγματικό κόσμο. Δηλαδή, σε αντίθεση με τη σχεδίαση, στην προσομοίωση υπάρχει έντονο το στοιχείο του ρεαλισμού ή έστω της αληθοφάνειας, διότι συνήθως ο στόχος είναι η εμπειρία του χρήστη από τη διάδρασή του με το περιβάλλον να είναι όσο το δυνατόν πιο «φυσική». Ορισμένες παραλλαγές που μπορούν να εντοπιστούν σε εφαρμογές προσομοίωσης ανάλογα με τους στόχους τους είναι: α) *εξάσκηση*, όπου οι εκπαιδευόμενοι καλούνται να εξασκηθούν στον χειρισμό ενός συστήματος ή σε κάποια ειδική κατάσταση, β) *πρωτοτυποποίηση*, όπου παρουσιάζεται η λειτουργία ενός πρωτοτύπου (προϊόντος ή συστήματος) σε ένα περιβάλλον προσομοίωσης, γ) *παρουσίαση*, όπου απεικονίζεται ένα υπαρκτό περιβάλλον ή σύστημα μέσω προσομοίωσης και δ) *ψυχαγωγία*, όπου η προσομοίωση εξυπηρετεί τους σκοπούς κάποιου παιχνιδιού ή άλλης μορφής ψυχαγωγίας.

Τα δεδομένα στα περιβάλλοντα προσομοίωσης είναι «συμπαγή» και αναπαριστούν ρεαλιστικά αντικείμενα ή χώρους, με βασικό ζητούμενο την ποιοτική οπτική και ηχητική αναπαράσταση. Σε αρκετές περιπτώσεις τα αντικείμενα θα πρέπει επιπλέον να εμφανίζουν και αληθοφανή φυσική συμπεριφορά. Κατά συνέπεια, η εκμετάλλευση των σύγχρονων αλγορίθμων και τεχνικών από τον χώρο των τρισδιάστατων γραφικών και της συνθετικής κίνησης είναι επιτακτική, ενώ και η χρήση υλικού στερεοσκοπίας είναι επιθυμητή για την περαιτέρω ενίσχυση της εμπύθισης των χρηστών. Σε περιβάλλοντα ψυχαγωγίας η καλή χρήση του ήχου, οι φυσικές κινήσεις και εκφράσεις των χαρακτήρων και η γενικότερη αισθητική του περιβάλλοντος μπορούν να συνεισφέρουν θετικά στην εξιστόρηση και στη συναισθηματική εμπλοκή των χρηστών.

Σε ό,τι αφορά τις αλληλεπιδράσεις χρηστών, στα περιβάλλοντα προσομοίωσης η έμφαση δίνεται στη φυσική πλοήγηση και στις αλληλεπιδράσεις με τα αντικείμενα του περιβάλλοντος. Μάλιστα, όταν στόχος του συστήματος είναι η εκπαίδευση ή η εξάσκηση των χρηστών σε κάποιο αντικείμενο, γίνεται προσπάθεια ο τρόπος διάδρασης και η πολυπλοκότητα χρήσης του περιβάλλοντος να είναι ανάλογη με την πραγματικότητα. Παράδειγμα τέτοιας περίπτωσης διεργασίας θα μπορούσε να είναι η εξάσκηση φοιτητών ιατρικής σε συγκεκριμένες χειρουργικές επεμβάσεις μέσω ενός περιβάλλοντος προσομοίωσης σε συνδυασμό με τη χρήση εξειδικευμένου υλικού.

4.2.3 Εξερεύνηση δεδομένων και πληροφοριών

Υπάρχουν περιπτώσεις περιβαλλόντων στα οποία ο χρήστης καλείται να εξερευνήσει και να μελετήσει έναν μεγάλο όγκο, εξωτερικών συνήθως, δεδομένων τα οποία απεικονίζονται εντός του τρισδιάστατου χώρου. Τέτοια παραδείγματα είναι τα περιβάλλοντα επιστημονικής οπτικοποίησης (*scientific visualization*) στα οποία τα δεδομένα αποκτούν συνήθως κάποια τρισδιάστατη μορφή για να διευκολύνουν τη μελέτη και ανάλυσή τους από ειδικούς (Bryson, 1996) και πολλών ειδών εκπαιδευτικά περιβάλλοντα στα οποία το τρισδιάστατο περιεχόμενο συσχετίζεται με περαιτέρω πληροφορίες για την παρουσίαση και επεξήγηση νέων εννοιών. Παρά το γεγονός ότι το πρώτο παράδειγμα έχει ως βασικό στόχο την ανάλυση των δεδομένων και την εξαγωγή χρήσιμων συμπερασμάτων ενώ το δεύτερο την κατανόηση ήδη γνωστών εννοιών ή φαινομένων, και στις δύο περιπτώσεις το περιβάλλον θα πρέπει να υποστηρίζει ικανοποιητικά τις διεργασίες εξερεύνησης, αναγνώρισης και μελέτης των δεδομένων. Στην εικόνα 5.4 παρουσιάζεται ένα παράδειγμα οπτικοποίησης επιστημονικών δεδομένων στον κόσμο του *Second Life* (Bourke, 2008).



Εικόνα 5.4 Οπτικοποίηση επιστημονικών δεδομένων στο *Second Life*, περιοχή *Genome Island*.

Κατά την εκτέλεση διεργασιών εξερεύνησης δεδομένων, οι χρήστες θα πρέπει να προσαρμόζουν συχνά την παρουσίαση των δεδομένων, καθώς για να εξαχθούν συμπεράσματα ενδέχεται να πρέπει να παρατηρηθούν τα δεδομένα από διαφορετικές θέσεις και οπτικές γωνίες αλλά και με διαφορετικούς τρόπους απεικόνισης. Σε εφαρμογές ανάλυσης, στόχος είναι να προκύψουν ποσοτικά και ποιοτικά συμπεράσματα σχετικά με τη διαδικασία, το σύστημα ή την κατάσταση υπό μελέτη. Τα δεδομένα που παρουσιάζονται σε τέτοιες εφαρμογές είναι συνήθως αφηρημένα από τη φύση τους (π.χ. τιμές θερμοκρασίας και πίεσης), ενώ πολλές φορές μπορεί να αναπαραστούν και δεδομένα υψηλότερων διαστάσεων. Αντίθετα, σε εφαρμογές με στόχο την παρουσίαση ή επεξήγηση εννοιών τα δεδομένα δεν είναι κατ' ανάγκη αφηρημένα. Συνήθως περιλαμβάνονται και πληροφορίες που συσχετίζονται με αντικείμενα, χώρους ή χρήστες του περιβάλλοντος και αποδίδονται σε μορφή κειμένου, βίντεο, αφηγήσεων κ.ά. Στους στόχους των εφαρμογών αυτών μπορεί να περιλαμβάνεται η αναζήτηση

δεδομένων και η κατανόηση της σχέσης της πληροφορίας με τα περιεχόμενα και δρώμενα του περιβάλλοντος. Οι χρήστες μπορεί να θέλουν να εντοπίσουν σχετικές πληροφορίες που συνδέονται με ένα συγκεκριμένο αντικείμενο ή ενέργεια του περιβάλλοντος ή, αντίστροφα, να εντοπίσουν τμήματα του περιβάλλοντος που συσχετίζονται με συγκεκριμένες πληροφορίες.

Κατά τη σχεδίαση εφαρμογών που υποστηρίζουν διαδικασίες εξερεύνησης δεδομένων και πληροφοριών, η έμφαση θα πρέπει να δίνεται σε τεχνικές οπτικοποίησης, πλοήγησης και χειρισμού δεδομένων, καθώς οι διαδικασίες αυτές είναι κρίσιμες για την επιτυχία της δραστηριότητας. Η ευκολία στην πλοήγηση θα βοηθήσει στη γρήγορη αλλαγή οπτικής. Δεν είναι βεβαίως απαραίτητο σε εφαρμογές ανάλυσης η πλοήγηση να είναι άκρως ρεαλιστική, καθώς η έμφαση δίνεται στην ευχρηστία. Επιπρόσθετα, ο μηχανισμός οπτικοποίησης θα πρέπει να επιτρέπει την προβολή δεδομένων πολλών διαστάσεων ποιοτικών ή και ποσοτικών ανάλογα με την περίπτωση —για παράδειγμα χρησιμοποιώντας στρώματα (layers) με διαφάνειες, προσθέτοντας ταμπέλες με επεξηγήσεις και τιμές κ.λπ.— και την εύκολη τροποποίηση του τύπου προβολής —π.χ. αφαίρεση ή προσθήκη στρωμάτων, περισσότερες ή λιγότερες επεξηγήσεις κ.λπ. Σε εφαρμογές παρουσίασης πληροφοριών θα πρέπει επιπλέον τα δεδομένα να είναι συσχετισμένα με αντικείμενα του περιβάλλοντος, να τοποθετούνται στην οθόνη με τρόπο που να είναι εύκολος ο εντοπισμός τους και να περιορίζονται οι επικαλύψεις, να περιλαμβάνονται δυνατότητες τροποποίησης της απεικόνισης, όπως για παράδειγμα διαμόρφωση του βαθμού λεπτομέρειας, απόκρυψη κ.λπ., και να υποστηρίζεται η αναζήτηση σε αυτά.

4.2.4 Συνεργασία

Στόχος στις συνεργατικές δραστηριότητες είναι η επίτευξη κάποιου κοινού στόχου μέσω της συντονισμένης εργασίας των μελών μιας ομάδας. Η συνεργασία στους εικονικούς κόσμους, ανάλογα και με το αντικείμενο εφαρμογής, μπορεί να είναι σύγχρονη ή και ασύγχρονη. Στην πρώτη περίπτωση οι χρήστες συμμετέχουν σε μία ή περισσότερες «συνεδρίες» για την εκτέλεση των ομαδικών εργασιών, όπως για παράδειγμα ομαδικές συζητήσεις εργασίας, συζήτηση και λήψη αποφάσεων σχετικά με τα επόμενα βήματα, ομαδική κριτική κ.ά. Σε άλλες περιπτώσεις, και ιδιαίτερα όταν το επιμέρους έργο που έχει ανατεθεί στον κάθεναν είναι ήδη γνωστό, οι εργασίες μπορούν να πραγματοποιούνται σε διαφορετικό χρόνο πάνω στο κοινό περιβάλλον. Οι συνεργατικές δραστηριότητες αναλύονται σε ενέργειες που σχετίζονται με: την εκτέλεση εργασιών πάνω σε αντικείμενα του περιβάλλοντος, την καλή επικοινωνία μεταξύ των μελών, τη λήψη αποφάσεων σχετικά με την πορεία του έργου και στρατηγικές αποφάσεις σχετικά με την ανάθεση ρόλων και δικαιωμάτων στα μέλη της ομάδας (Vosinakis, κ.ά. 2008).

Το περιβάλλον θα πρέπει να είναι σχεδιασμένο με τρόπο που να μπορεί να υποστηρίξει με επιτυχία τη συνεργασία, τόσο σύγχρονα όσο και ασύγχρονα. Κατά τη σύγχρονη συνεργασία θα πρέπει να λαμβάνονται υπόψη ζητήματα περιφερειακής επίγνωσης, δηλαδή να μπορεί ένας χρήστης να αντιληφθεί τις εργασίες στις οποίες εμπλέκονται άλλα μέλη της ομάδας. Επιπλέον, τόσο στη σύγχρονη όσο και στην ασύγχρονη συνεργασία, θα πρέπει να μπορούν οι συμμετέχοντες να αναγνωρίσουν την πρόοδο των εργασιών, το έργο που έχει ανατεθεί σε κάθε μέλος της ομάδας και τους αντίστοιχους ρόλους, τα δικαιώματα πρόσβασης και χρήσης που έχουν στα αντικείμενα και εργαλεία του περιβάλλοντος κ.λπ.

Οι παραπάνω απαιτήσεις μπορούν να καλυφθούν εν μέρει σε ένα εικονικό περιβάλλον με τη χρήση εξειδικευμένων αναπαραστάσεων. Η επίγνωση και ο καλύτερος συντονισμός μεταξύ των χρηστών μπορεί να υποστηριχθεί μέσω χαρτών κάτοψης που απεικονίζουν τα μέλη της ομάδας και καταδεικνύουν τις δραστηριότητές τους (π.χ. με κείμενο ή χρωματικά). Κατάλληλα μηνύματα θα μπορούσαν να στέλνονται στα μέλη της ομάδας όταν επιτυγχάνεται κάποια σημαντική πρόοδος στις εργασίες, για παράδειγμα όταν ένας χρήστης ολοκληρώνει κάποιο επιμέρους έργο. Κατά την εργασία στο περιβάλλον είναι χρήσιμο να επικοινωνείται με σαφήνεια το ποιος χρήστης χειρίζεται ποιο ή ποια αντικείμενα, για παράδειγμα με κατάλληλη απεικόνιση της ενσάρκωσης και τη χρήση οπτικών ενδείξεων που τη συνδέουν με τα αντικείμενα. Επιπλέον, ο ρόλος του κάθε χρήστη θα μπορούσε να απεικονίζεται στην ενσάρκωσή του με κάποια ξεκάθαρη ένδειξη. Σε αρκετές περιπτώσεις είναι χρήσιμο επίσης να δίνεται η δυνατότητα διαμοίρασης οπτικής γωνίας, ώστε να μπορούν ορισμένα μέλη της ομάδας να παρακολουθούν τις εργασίες από την οπτική ενός χρήστη και να συζητούν πάνω σε αυτές. Τέλος, η ασύγχρονη συνεργασία μπορεί να υποστηριχθεί επιπλέον με τη χρήση επισημειώσεων κειμένου (π.χ. σχόλια, ελλείψεις, επόμενα βήματα κ.λπ.) πάνω στο περιβάλλον, ενώ μπορούν να συμπεριληφθούν και πιο αφηρημένες αναπαραστάσεις που καταδεικνύουν συγκεκριμένες περιοχές, απαιτήσεις χρηστών κ.ά.

Σε ό,τι αφορά τις αλληλεπιδράσεις χρηστών προκύπτει ανάγκη για κατάλληλη υποστήριξη της πλοήγησης, του χειρισμού αντικειμένων και της επικοινωνίας. Οι τεχνικές πλοήγησης θα πρέπει να μπορούν να παρέχουν κατάλληλη οπτική για την πραγματοποίηση των εργασιών αλλά και να διευκολύνουν τον εντοπισμό των συνεργατών. Ο χειρισμός συνήθως γίνεται μέσω εξειδικευμένων εργαλείων που σχετίζονται με το αντικείμενο συνεργασίας και θα πρέπει να διευκολύνουν την εκτέλεση εργασιών. Επιπλέον, θα πρέπει να παρέχονται πολλαπλά μέσα σύγχρονης και ασύγχρονης επικοινωνίας, ιδανικά και με δυνατότητες αναζήτησης στο περιεχόμενο. Τέλος, σε αρκετές περιπτώσεις απαιτείται και η υποστήριξη ενεργειών διαμόρφωσης ομάδων, ανάθεσης ρόλων και διαχείρισης δικαιωμάτων πρόσβασης.

5 Αξιολόγηση ευχρηστίας

Η *ευχρηστία (usability)* είναι μια κεντρική έννοια στην ανάπτυξη διαδραστικών συστημάτων τις τελευταίες δεκαετίες. Όπως προκύπτει και από την ίδια την ετυμολογία της λέξης, η ευχρηστία αναφέρεται στην ευκολία χρήσης ενός προϊόντος ή μιας εφαρμογής. Αν προσπαθήσουμε να δώσουμε έναν ευρύ ορισμό της ευχρηστίας, θα λέγαμε ότι αποτελεί τη σύνθεση των εξής επιμέρους εννοιών: της χρησιμότητας, της απόδοσης, της ευκολίας εκμάθησης και της ικανοποίησης από τη χρήση. Επομένως, για να χαρακτηρίσουμε εύχρηστη μια διαδραστική εφαρμογή θα πρέπει να διασφαλίσουμε ότι είναι χρήσιμη για την επίτευξη κάποιου στόχου, ότι μέσω αυτής ο χρήστης μπορεί να πετύχει τον στόχο του γρήγορα και με μικρή πιθανότητα λαθών, ότι μπορεί να μάθει εύκολα τη χρήση της και ότι είναι ευχάριστη στη χρήση. Αναγνωρίζοντας ότι η ευχρηστία είναι κρίσιμος παράγοντας επιτυχίας, οι σύγχρονες μεθοδολογίες ανάπτυξης εφαρμογών στηρίζονται πάρα πολύ στην εμπλοκή των χρηστών και στην αξιολόγηση της ευχρηστίας από τα πρώτα στάδια σχεδίασης και καθ' όλο τον κύκλο ζωής της εφαρμογής (Mack & Nielsen, 1993). Ως εκ τούτου, έχει αναπτυχθεί ένας μεγάλος αριθμός από σχεδιαστικές κατευθύνσεις (guidelines), παραδείγματα (design paradigms), πρότυπα (design patterns), μεθόδους και τεχνικές σχεδίασης και αξιολόγησης βασισμένες στην ευχρηστία. Ορισμένα από αυτά έχουν γενικότερη στόχευση, ενώ άλλα αναφέρονται σε πιο συγκεκριμένες περιοχές εφαρμογής. Σε κάθε περίπτωση, η μεγάλη πλειοψηφία των παραδειγμάτων και προτύπων εστιάζει στα κυρίαρχα μοντέλα διάδρασης, όπως οι παραθυρικές εφαρμογές, ο παγκόσμιος ιστός και τα τελευταία χρόνια οι εφαρμογές για κινητές συσκευές.

Στην περίπτωση των διαδραστικών τρισδιάστατων περιβαλλόντων, όπως είναι και οι εικονικοί κόσμοι, η διασφάλιση της ευχρηστίας είναι ιδιαίτερα δύσκολη, με αποτέλεσμα να είναι ακόμα πιο επιτακτική η ανάγκη πολλαπλών αξιολογήσεων κατά τη διάρκεια της ανάπτυξης (Stanney κ.ά., 2003).

- Στις περισσότερες παραθυρικές εφαρμογές η διάδραση είναι «διαλογικής» μορφής, δηλαδή ο χρήστης εκτελεί κάποια ενέργεια συνήθως στιγμιαία, όπως το πάτημα ενός πλήκτρου, το σύστημα ανταποκρίνεται με κάποιον τρόπο και στη συνέχεια περιμένει από τον χρήστη να εκτελέσει την επόμενη ενέργεια. Αντίθετα, στους εικονικούς κόσμους η αλλαγή είναι συνεχής και ο χρήστης πρέπει να έχει την προσοχή του εστιασμένη στο περιβάλλον για να την παρατηρήσει.
- Στα παραθυρικά περιβάλλοντα είναι εύκολη η παρατήρηση της τρέχουσας κατάστασης της εφαρμογής μέσα από κατάλληλες ενδείξεις που παρέχονται από τη διεπαφή, ενώ σε τρισδιάστατους χώρους η οπτική του χρήστη είναι περιορισμένη και απαιτείται πλοήγηση για να αποκτήσει επίγνωση της κατάστασης στις διάφορες περιοχές του περιβάλλοντος.
- Στα δισδιάστατα περιβάλλοντα υπάρχουν αναγνωρίσιμα στοιχεία διεπαφής βασισμένα σε ήδη γνωστά πρότυπα, για παράδειγμα πλήκτρα, πεδία εισαγωγής κειμένου, λίστες κ.λπ, ενώ στους εικονικούς κόσμους οποιοδήποτε τρισδιάστατο αντικείμενο ή δισδιάστατο στοιχείο μπορεί να είναι εν δυνάμει μέρος της διεπαφής.
- Οι ενέργειες πλοήγησης στους εικονικούς κόσμους είναι μεγάλης διάρκειας και συνήθως γνωστικά επίπονες για τους νέους χρήστες, καθώς απαιτούν από αυτούς να αναπτύξουν ικανότητες εύρεσης μονοπατιού σε ένα συνθετικό περιβάλλον.
- Τέλος, η μετάφραση από τις δύο διαστάσεις στις τρεις, τόσο κατά την εκτέλεση ενεργειών του χρήστη μέσω ποντικιού ή πληκτρολογίου όσο και κατά την απεικόνιση του χώρου στην οθόνη, επίσης επιβαρύνουν γνωστικά και δυσκολεύουν τους αρχάριους χρήστες.

Οι περιπτώσεις που αναφέραμε παραπάνω είναι ορισμένες από τις αιτίες για τις οποίες οι εφαρμογές εικονικής πραγματικότητας μπορούν να προκαλέσουν σημαντικά προβλήματα ευχρηστίας μετριάζοντας τον αρχικό

ενθουσιασμό των χρηστών από το νέο μέσο και κάποιες φορές καταλήγοντας σε αποτυχία. Είναι συνεπώς απαραίτητο να στοχεύσει από νωρίς η ομάδα ανάπτυξης ενός εικονικού κόσμου στη διασφάλιση της ευχρηστίας της εφαρμογής, κάτι που μπορεί να πετύχει με την εφαρμογή κατάλληλων μεθόδων αξιολόγησης.

Οι ιδιαιτερότητες στο μοντέλο απεικόνισης και αλληλεπίδρασης των εικονικών περιβαλλόντων προκαλούν ορισμένες δυσκολίες στην απευθείας εφαρμογή παραδοσιακών μεθόδων αξιολόγησης διαδραστικών συστημάτων. Σε μεθόδους στις οποίες ο αξιολογητής χρειάζεται να παρακολουθεί τις αντιδράσεις του χρήστη για τον εντοπισμό προβλημάτων, είναι δύσκολο να παρατηρεί ταυτόχρονα και τις εκφράσεις προσώπου του και τις ενέργειες που εκτελεί στο περιβάλλον. Οι ενέργειές του αναμένεται να έχουν σημαντικά μεγαλύτερη συχνότητα και διάρκεια σε σχέση, για παράδειγμα, με τη χρήση μιας ιστοσελίδας, επομένως χρειάζονται διαρκή παρατήρηση. Επιπλέον, με δεδομένο ότι οι εικονικοί κόσμοι είναι πολυχρηστικά περιβάλλοντα και σε πολλές περιπτώσεις θα χρειαστεί να αξιολογηθεί η ευχρηστία σε σχέση με την εκτέλεση μιας ομαδικής δραστηριότητας, το πρόβλημα γίνεται ακόμα πιο σύνθετο. Θα πρέπει να μπορούν να παρακολουθούνται ταυτόχρονα πολλαπλοί χρήστες, ιδανικά ο καθένας από τη δική του οπτική του κόσμου, κάτι που απαιτεί παραπάνω αξιολογητές ή και διαδικασίες καταγραφής. Στους εικονικούς κόσμους παράγεται ένας μεγάλος όγκος δεδομένων, όπως κινήσεις και ενέργειες χρηστών, ομιλία, επικοινωνία μέσω κειμένου κ.ά., και θα πρέπει να αναπτυχθούν κατάλληλοι μηχανισμοί καταγραφής και συγχρονισμένης αναπαραγωγής για μελέτη και εξαγωγή συμπερασμάτων. Η συνομιλία των αξιολογητών με τους χρήστες κατά τη διάρκεια εκτέλεσης μιας δραστηριότητας μπορεί να διαταράξει την αίσθηση της παρουσίας των χρηστών και να επηρεάσει τα αποτελέσματα. Τέλος, το πολύ μεγάλο εύρος σχεδιαστικών επιλογών σε συνδυασμό με τις περιορισμένες σχεδιαστικές κατευθύνσεις και πρότυπα δυσκολεύει τη διεξαγωγή μελετών ως προς την ευχρηστία της διεπαφής από έμπειρους αξιολογητές.

Τα τελευταία χρόνια έχουν εμφανιστεί διάφορες μέθοδοι και τεχνικές αξιολόγησης εστιασμένες σε εικονικά περιβάλλοντα, ώστε να λαμβάνουν υπόψη τις ιδιαιτερότητες του μέσου (Bowman κ.ά., 2002). Οι βασικές κατηγορίες των μεθόδων αυτών είναι:

- *Γνωστική περιδιάβαση (Cognitive Walkthrough)*: η αξιολόγηση της διεπαφής γίνεται μέσω της διάσχισης των διάφορων βημάτων που θα ακολουθήσει ο χρήστης για την εκτέλεση συχνών διεργασιών και την αξιολόγηση της ικανότητας της διεπαφής να υποστηρίξει κάθε βήμα.
- *Διαμορφωτική αξιολόγηση*: αξιολογείται η ευχρηστία μιας εφαρμογής μέσω της τοποθέτησης αντιπροσωπευτικών χρηστών σε σενάρια σχετιζόμενα με συγκεκριμένες διεργασίες. Στόχος είναι να εξεταστεί κατά πόσο η εφαρμογή μπορεί να υποστηρίξει την εξερεύνηση, εκμάθηση και απόδοση των χρηστών, ενώ τα αποτελέσματα που παράγονται μπορεί να είναι ποσοτικά ή και ποιοτικά.
- *Ευρετική αξιολόγηση από ειδικούς (Heuristic Evaluation)*: ένας αριθμός από έμπειρους σε θέματα ευχρηστίας αξιολογούν ξεχωριστά ένα πρωτότυπο της εφαρμογής εφαρμόζοντας έναν αριθμό από σχετικές ευρετικές μεθόδους ή σχεδιαστικές κατευθύνσεις, με στόχο την ανακάλυψη προβλημάτων ευχρηστίας. Δεν εμπλέκονται αντιπροσωπευτικοί χρήστες.
- *Συνεντεύξεις και ερωτηματολόγια*: συλλογή δημογραφικών δεδομένων και υποκειμενικών απόψεων των χρηστών μετά τη χρήση της εφαρμογής μέσω ερωτηματολογίων και δομημένων ή ανοιχτών συνεντεύξεων.
- *Συγκριτική αξιολόγηση*: Σύγκριση δύο ή περισσότερων διαμορφώσεων στη σχεδίαση μιας διεπαφής, σε συγκεκριμένα στοιχεία διεπαφής ή και σε τεχνικές αλληλεπίδρασης, με στόχο την επιλογή της καταλληλότερης. Η μέθοδος εμπλέκει αντιπροσωπευτικούς χρήστες και μπορεί να έχει ποιοτικά ή και ποσοτικά αποτελέσματα.

Μια ταξινόμηση των μεθόδων αυτών (Bowman κ.ά., 2002) τις ομαδοποιεί βάσει τριών διαφορετικών χαρακτηριστικών: της εμπλοκής τελικών χρηστών, του πλαισίου αξιολόγησης και του τύπου των αποτελεσμάτων. Σε ό,τι αφορά την εμπλοκή χρηστών, οι μέθοδοι διακρίνονται σε αυτές που απαιτούν τη συμμετοχή των χρηστών και την εκτέλεση από μέρος τους κάποιων σεναρίων και σε αυτές που πραγματοποιούνται από έμπειρους αξιολογητές χωρίς χρήστες. Ως προς το πλαίσιο αξιολόγησης, οι μέθοδοι διακρίνονται σε αυτές που αξιολογούν μια συγκεκριμένη εφαρμογή και σε αυτές που αξιολογούν μεμονωμένες τεχνικές διάδρασης εκτός κάποιου πλαισίου εφαρμογής, με στόχο την παραγωγή γενικότερων συμπερασμάτων. Τέλος, τα αποτελέσματα των μεθόδων μπορεί να είναι ποσοτικά ή ποιοτικά. Τα ποσοτικά δεδομένα που προκύπτουν από μία μέθοδο (π.χ. χρόνος εκτέλεσης μιας διεργασίας και αριθμός λαθών) μπορούν να

καταδείξουν ότι υπάρχουν προβλήματα ευχρηστίας, ενώ τα ποιοτικά (π.χ. κρίσιμα συμβάντα, σχόλια και αντιδράσεις χρηστών) μπορούν να δώσουν χρήσιμες κατευθύνσεις ως προς το είδος και τα αίτια των προβλημάτων. Το συμπέρασμα που προκύπτει από την ταξινόμηση των μεθόδων αξιολόγησης ευχρηστίας εικονικών περιβαλλόντων σύμφωνα με τα παραπάνω κριτήρια είναι ότι οι υπάρχουσες μέθοδοι που δεν απαιτούν τη συμμετοχή χρηστών μπορούν να οδηγήσουν μόνο σε ποιοτικά συμπεράσματα (π.χ. γνωστική περιδιάβαση ή ευρετική αξιολόγηση). Αυτό πρακτικά σημαίνει ότι η συχνή παραγωγή ενδιάμεσων λειτουργικών πρωτοτύπων και η εμπλοκή αντιπροσωπευτικών χρηστών στην εκτέλεση διεργασιών είναι απαραίτητες κατά την ανάπτυξη εφαρμογών βασισμένων σε εικονικούς κόσμους.

5.1 Χαρακτηριστικά ευχρηστίας σε εικονικά περιβάλλοντα (Gabbard)

Ο Gabbard (1997) έκανε μια προσπάθεια συλλογής, προσαρμογής και ταξινόμησης χαρακτηριστικών ευχρηστίας για εικονικά περιβάλλοντα με στόχο αφενός την καταγραφή και ομαδοποίησή τους και αφετέρου την παραγωγή βασικών σχεδιαστικών κατευθύνσεων από αυτά. Η συλλογή των χαρακτηριστικών ευχρηστίας μπορεί να αποτελέσει βάση για την εξέλιξη και βελτίωση των μεθοδολογιών αξιολόγησης των εικονικών περιβαλλόντων αλλά και να βοηθήσει ευρετικές αξιολογήσεις εφαρμογών από ειδικούς ευχρηστίας.

Η συλλογή περιέχει 195 σχεδιαστικές κατευθύνσεις ταξινομημένες σε τέσσερις βασικές κατηγορίες και σε επιμέρους υποκατηγορίες. Οι βασικές κατηγορίες είναι:

- *οι χρήστες και οι δραστηριότητές τους*: τα χαρακτηριστικά ευχρηστίας αυτής της κατηγορίας σχετίζονται με τους χρήστες και τις ικανότητές τους (δημογραφικά χαρακτηριστικά, προηγούμενη εμπειρία και γνώσεις κ.λπ.), το είδος της δραστηριότητας που καλούνται να εκτελέσουν στο περιβάλλον και την υποστήριξη των βασικών τεχνικών διάδρασης, δηλαδή της πλοήγησης, της επιλογής και του χειρισμού αντικειμένων,
- *το εικονικό μοντέλο*: θέματα αναπαράστασης των χρηστών, των εικονικών πρακτόρων, του περιβάλλοντος και πληροφοριών σχετικά με την εφαρμογή,
- *η εμφάνιση της διεπαφής του περιβάλλοντος*: χαρακτηριστικά ευχρηστίας που σχετίζονται με τις δυνατότητες και τους περιορισμούς του υλικού που θα χρησιμοποιηθεί για την οπτική, ακουστική και απτική έξοδο,
- *οι μηχανισμοί εισόδου του περιβάλλοντος*: χαρακτηριστικά ευχρηστίας σχετικά με τους μηχανισμούς εισόδου και την ικανότητά τους να υποστηρίξουν τις ενέργειες των χρηστών στο περιβάλλον.

Παρά το γεγονός ότι στην παραπάνω ταξινόμηση συμπεριλαμβάνονται και θέματα σχετικά με υλικό εμπύθισης, ένα σημαντικό μέρος των οδηγιών μπορεί να βρει εφαρμογή στους εικονικούς κόσμους, ιδιαίτερα αυτές των δύο πρώτων κατηγοριών. Ακολουθούν ορισμένες ενδεικτικές σχεδιαστικές κατευθύνσεις:

- Να υποστηρίζονται χρήστες με διαφορετικά επίπεδα κατανόησης του χώρου εφαρμογής.
- Να υποστηρίζεται η ταυτόχρονη εκτέλεση ενεργειών.
- Σε συνεργατικές εφαρμογές να υποστηρίζεται η κοινωνική διάδραση μεταξύ χρηστών (ομαδικές συζητήσεις, παιχνίδια ρόλων, άτυπες αλληλεπιδράσεις κ.ά.).
- Να υποστηρίζεται ο συντονισμός ομαδικών δραστηριοτήτων.
- Η πλοήγηση να υποστηρίζεται με χωρικές πινακίδες, ορόσημα και ορίζοντα.
- Οι μεταφορές πλοήγησης θα πρέπει να ταιριάζουν με την περιοχή εφαρμογής.
- Η επιλογή αντικειμένων βάσει χωρικών ιδιοτήτων να γίνεται με άμεσο τρόπο (π.χ. με κατάδειξη από το ποντίκι).
- Να υποστηρίζεται η επιλογή πολλαπλών αντικειμένων.
- Να παρέχεται κατάλληλη οπτική χρήση κατά τον χειρισμό αντικειμένων.
- Να μπορεί ο χρήστης να αντιληφθεί μέσω της διεπαφής ποιες ενέργειες είναι δυνατές πάνω στα αντικείμενα.
- Να παρέχεται οπτική πρώτου προσώπου αν απαιτείται ο χρήστης να έχει ισχυρή αίσθηση της παρουσίας.

- Να παρέχεται οπτική τρίτου προσώπου αν η θέση και κίνηση του χρήστη σε σχέση με τα υπόλοιπα αντικείμενα του χώρου είναι κρίσιμη.
- Να επιτρέπεται στους χρήστες να αλλάζουν οπτική ή σημείο εστίασης.
- Η συμπεριφορά των πρακτόρων να προσαρμόζεται δυναμικά ανάλογα με το πλαίσιο, τη δραστηριότητα των χρηστών κ.ά.
- Να χρησιμοποιείται το σκηνικό κατάλληλα ώστε να αυξήσει την παρουσία των χρηστών.
- Οι πληροφορίες σχετικά με την εφαρμογή να παρουσιάζονται με ξεκάθαρο τρόπο και να συνδέονται με τους χώρους και τα αντικείμενα του περιβάλλοντος.

5.2 Μεθοδολογία ακολουθιακής αξιολόγησης (Gabbard, Hix & Swan)

Οι Gabbard, Hix & Swan (1999), αναγνωρίζοντας τις δυσκολίες εφαρμογής παραδοσιακών μεθόδων αξιολόγησης στα εικονικά περιβάλλοντα, πρότειναν μια ολοκληρωμένη μεθοδολογία ανθρωποκεντρικής σχεδίασης και αξιολόγησης εφαρμογών βασισμένων σε εικονικά περιβάλλοντα, η οποία στηρίζεται στην ακολουθιακή εκτέλεση τεσσάρων σταδίων. Τα στάδια αυτά περιλαμβάνουν πολλαπλές και διαφορετικού τύπου μεθόδους αξιολόγησης, οι οποίες είναι σύμφωνα με τους συγγραφείς περισσότερο κατάλληλες να αντεπεξέλθουν στις ιδιαιτερότητες των τρισδιάστατων περιβαλλόντων. Τα επιμέρους στάδια είναι:

- ανάλυση δραστηριοτήτων των χρηστών,
- αξιολόγηση από ειδικούς βάσει σχεδιαστικών κατευθύνσεων,
- διαμορφωτική ανθρωποκεντρική αξιολόγηση και
- συμπερασματικές συγκριτικές αξιολογήσεις.

Στο στάδιο της ανάλυσης δραστηριοτήτων των χρηστών προσδιορίζεται μια πλήρης περιγραφή των δραστηριοτήτων, διεργασιών και μεθόδων που απαιτούνται στην περιοχή εφαρμογής. Επιπλέον καθορίζονται και όλοι οι υπόλοιποι πόροι που μπορεί να χρειαστούν από τους χρήστες κατά τη συνεργατική εκτέλεση των διεργασιών τους. Η ανάλυση αυτή θα πρέπει να λαμβάνει υπόψη τις ανάγκες και τους στόχους των χρηστών καθώς και τις οργανωσιακές και κοινωνικές διεργασίες που σχετίζονται με τον χώρο εφαρμογής, και να καταλήγει στον προσδιορισμό των ομάδων χρηστών με τα χαρακτηριστικά τους και των στόχων και προδιαγραφών του υπό σχεδίαση συστήματος. Από τη λεπτομερή καταγραφή και μελέτη των παραπάνω στοιχείων θα πρέπει η σχεδιαστική ομάδα να καταλήξει στον προσδιορισμό και στην περιγραφή όλων των βασικών δραστηριοτήτων που θα πρέπει να υποστηρίζονται από το σύστημα, στην προτεραιοποίηση στις μεταξύ τους σχέσεις και αλληλεξαρτήσεις.

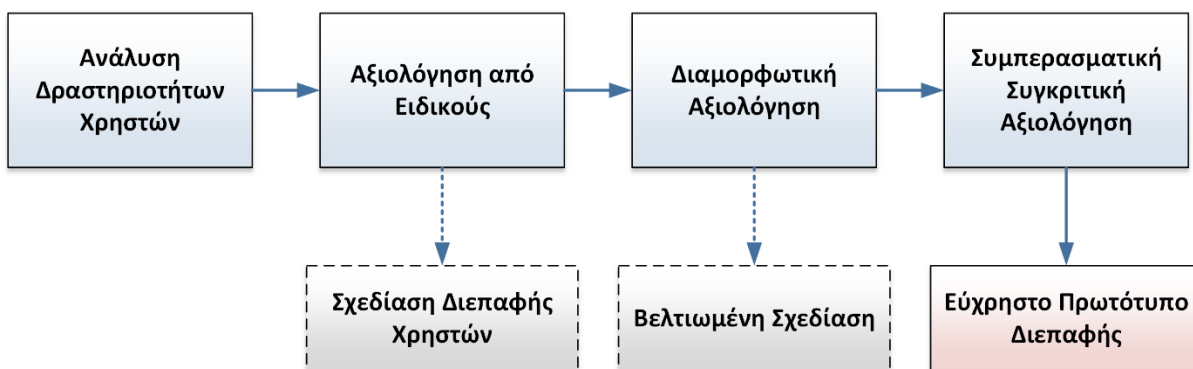
Στο επόμενο στάδιο σχεδιάζονται και καταγράφονται ορισμένες αρχικές λύσεις σχετικά με τις αλληλεπιδράσεις των χρηστών και τη διεπαφή του περιβάλλοντος, και αξιολογούνται από ειδικούς ως προς την ευχρηστία τους. Ένας ή περισσότεροι ειδικοί ευχρηστίας μελετούν τις προτεινόμενες διεπαφές ή και υλοποιημένα πρωτότυπα και εξετάζουν τον βαθμό που αυτές είναι συνεπείς με τις σχεδιαστικές κατευθύνσεις ευχρηστίας. Εάν εντοπίζονται σημαντικές παραβιάσεις, γίνονται προτάσεις επανασχεδίασης και τροποποιούνται ανάλογα οι προτεινόμενες διεπαφές. Μια βασική πρόκληση σε αυτό το στάδιο είναι ότι οι σχεδιαστικές κατευθύνσεις ευχρηστίας ειδικά για εικονικά περιβάλλοντα είναι σχετικά περιορισμένες, επομένως οι ειδικοί θα πρέπει να έχουν καλή εμπειρία σε σχετικές περιοχές ώστε να μπορούν να διακρίνουν ορισμένα εξειδικευμένα ζητήματα ευχρηστίας που σχετίζονται με τις διεπαφές αυτές. Η συλλογή σχεδιαστικών κατευθύνσεων του Gabbard (βλ. ενότητα 5.1) μπορεί να αποτελέσει μια καλή αρχική βάση. Κατά την ολοκλήρωση αυτού του σταδίου παράγονται μία ή περισσότερες σχεδιαστικές λύσεις και αντιπροσωπευτικά σενάρια χρήστη για την εκτέλεση των διεργασιών που έχουν προσδιοριστεί.

Στο στάδιο της διαμορφωτικής αξιολόγησης καλούνται αντιπροσωπευτικοί χρήστες να εκτελέσουν τα σενάρια που έχει προκαθορίσει η ομάδα ανάπτυξης, με στόχο τον εντοπισμό προβλημάτων ευχρηστίας και τη βελτίωση της διεπαφής. Οι αξιολογήσεις αυτές συνήθως εκτελούνται επαναληπτικά μέχρι να οδηγηθεί το σύστημα σε ένα ικανοποιητικό επίπεδο ευχρηστίας. Αρχικά οι σχεδιαστές προσδιορίζουν τα ενδεικτικά σενάρια χρήσης. Στη συνέχεια οι αντιπροσωπευτικοί χρήστες καλούνται να τα εκτελέσουν χρησιμοποιώντας υλοποιημένο πρωτότυπο της εφαρμογής. Κατά την εκτέλεση των σεναρίων οι χρήστες εφαρμόζουν το πρωτόκολλο «σκέψου δυνατά» (think aloud), σύμφωνα με το οποίο θα πρέπει να λένε δυνατά τις σκέψεις τους σχετικά με όσα βλέπουν, καταλαβαίνουν και πρόκειται να κάνουν (Barnard & Sandberg, 1994). Αυτό βοηθάει

τους αξιολογητές να αντιληφθούν καλύτερα τι είχαν στο μυαλό τους οι χρήστες όταν έκαναν συγκεκριμένες επιλογές, ώστε να προσπαθήσουν να εντοπίσουν πιθανά αίτια προβλημάτων ευχρηστίας. Οι αξιολογητές συλλέγουν επιπλέον πλούσια ποσοτικά και ποιοτικά δεδομένα για κάθε σενάριο που εκτελείται με μεθόδους όπως καταγραφή δεδομένων, ερωτηματολόγια και συνεντεύξεις. Μετά την ολοκλήρωση των αξιολογήσεων μελετώνται τα αποτελέσματα και, εφόσον εντοπιστούν σημαντικά ζητήματα ευχρηστίας, λαμβάνονται αποφάσεις επανασχεδίασης τμημάτων της διεπαφής και κατασκευάζεται νέα έκδοση του πρωτοτύπου. Αυτό με τη σειρά του αξιολογείται ξανά από χρήστες κ.λπ.

Στο τελευταίο στάδιο των συμπερασματικών συγκριτικών αξιολογήσεων στόχος είναι η βελτίωση της συνολικής απόδοσης του συστήματος. Αφού έχει διασφαλιστεί από το προηγούμενο στάδιο ότι η ευχρηστία της εφαρμογής κινείται σε αποδεκτά επίπεδα και η βασική μορφή της διεπαφής είναι ολοκληρωμένη, στη φάση αυτή γίνονται συγκρίσεις μεταξύ διαφορετικών αποδεκτών λύσεων για τις ίδιες διεργασίες. Στην αξιολόγηση αυτή συμμετέχουν και πάλι αντιπροσωπευτικοί χρήστες, οι οποίοι εκτελούν προκαθορισμένες διεργασίες, αλλά συλλέγονται κυρίως ποσοτικά δεδομένα. Στόχος είναι να προσδιοριστούν οι λύσεις εκείνες που βελτιώνουν τη χρήση εφαρμογής σύμφωνα με τα κριτήρια που έχει θέσει η σχεδιαστική ομάδα, για παράδειγμα η ταχύτερη δυνατή ολοκλήρωση του έργου με τα λιγότερα δυνατά λάθη. Οι λύσεις που συγκρίνονται μπορεί να αφορούν λειτουργικά στοιχεία της διεπαφής, τεχνικές αλληλεπίδρασης ή ακόμα και συσκευές εισόδου.

Τα βήματα της μεθοδολογίας παρουσιάζονται στην εικόνα 5.5.



Εικόνα 5.5 Μεθοδολογία ακολουθιακής αξιολόγησης των Gabbard κ.ά.

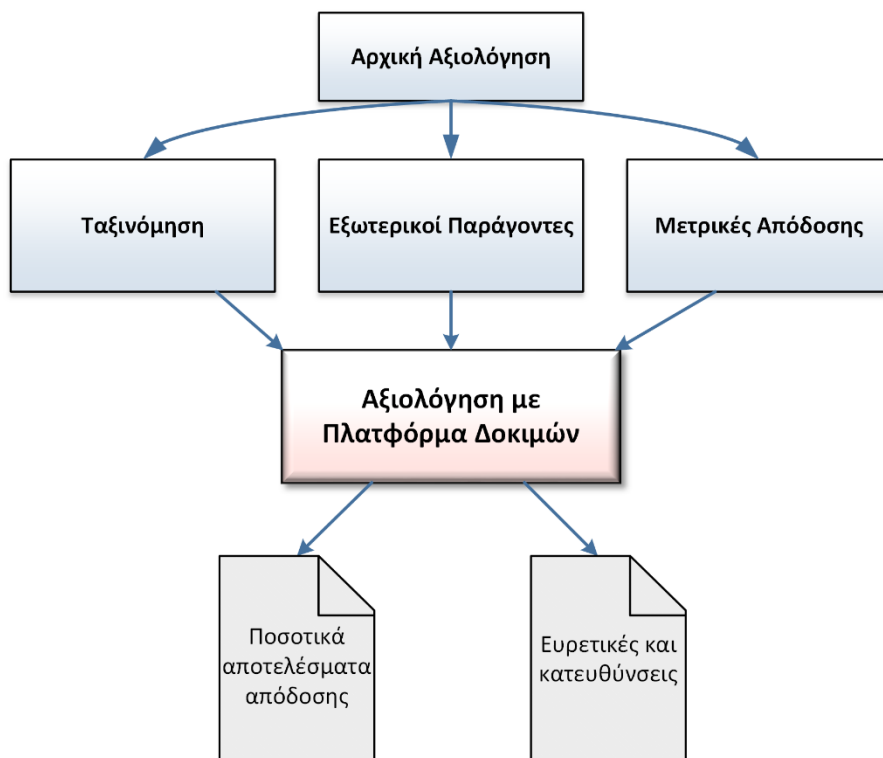
Η μεθοδολογία των Gabbard, Hix και Swan, αν και παρουσιάζεται ως μεθοδολογία σχεδίασης και αξιολόγησης, επί της ουσίας επικεντρώνεται στη διαδοχική χρήση τριών μεθόδων αξιολόγησης. Η ευχρηστία αξιολογείται από τα αρχικά στάδια βάσει οδηγιών για τον καλύτερο προσδιορισμό της μορφής της διεπαφής, στη συνέχεια και αφού αναπτυχθούν υψηλής πιστότητας πρωτότυπα η διεπαφή δοκιμάζεται με χρήστες, ώστε να προκύψουν κατάλληλες σχεδιαστικές τροποποιήσεις, και τέλος βελτιώνονται επιμέρους στοιχεία της εφαρμογής μέσω στοχευμένων συγκριτικών αξιολογήσεων.

5.3 Αξιολόγηση με πλατφόρμα δοκιμών (Bowman & Hodges)

Οι Bowman & Hodges (1999) παρουσίασαν μια μεθοδολογία αξιολόγησης τεχνικών αλληλεπίδρασης για εικονικά περιβάλλοντα. Η μεθοδολογία τους, που ονομάζεται αξιολόγηση με πλατφόρμα δοκιμών (testbed evaluation) έχει ως στόχο τη συστηματική αξιολόγηση τεχνικών αλληλεπίδρασης εκτός πλαισίου συγκεκριμένων εφαρμογών, ώστε να εξαχθούν γενικευμένα και επαναχρησιμοποιήσιμα συμπεράσματα σχετικά με την ευχρηστία τους. Σε αντίθεση δηλαδή με την ακολουθιακή αξιολόγηση που παρουσιάσαμε στην προηγούμενη ενότητα, η συγκεκριμένη μεθοδολογία δεν εστιάζει στην καλή εκτέλεση περίπλοκων εργασιών χρηστών στα πλαίσια μιας εφαρμογής αλλά στη χρήση βασικών τεχνικών διάδρασης για την εκτέλεση απλών διεργασιών, οι οποίες δοκιμάζονται από χρήστες σε ειδικά κατασκευασμένα περιβάλλοντα, τις «πλατφόρμες δοκιμών». Στην εικόνα 5.6 παρουσιάζονται τα βήματα της μεθοδολογίας.

Στο πρώτο στάδιο της μεθοδολογίας πραγματοποιείται μια *αρχική αξιολόγηση*. Στόχος είναι να προκύψει μια βασική κατανόηση των γενικών διεργασιών για τις οποίες ενδιαφέρεται η σχεδιαστική ομάδα και των διαθέσιμων διαδραστικών τεχνικών για την εκτέλεσή τους. Η αξιολόγηση θα βασιστεί στην εμπειρία της

ομάδας σε διαδραστικές τεχνικές καθώς και στην παρατήρηση ομάδων χρηστών κατά την εκτέλεση των απαιτούμενων διεργασιών.



Εικόνα 5.6 Αξιολόγηση με πλατφόρμα δοκιμών των Bowman & Hodges

Το επόμενο στάδιο είναι η δημιουργία μιας *ταξινόμησης* των διαδραστικών τεχνικών για την υποστήριξη της διεργασίας που θα αξιολογηθεί. Η ταξινόμηση διαχωρίζει τη διεργασία σε επιμέρους ενέργειες, για τις οποίες χρειάζεται να ληφθούν σχεδιαστικές αποφάσεις. Υπό αυτήν την έννοια, η ταξινόμηση αποτελεί μια προσεκτική ανάλυση διεργασιών των χρηστών. Η διεργασία αναλύεται σε επαρκώς απλοποιημένες υπο-εργασίες και καθεμία από αυτές συσχετίζεται με μια λίστα με τις πιθανές τεχνικές που μπορούν να την υποστηρίξουν.

Στη συνέχεια καθορίζονται οι *εξωτερικοί παράγοντες* που ενδέχεται να επηρεάζουν τη χρήση των τεχνικών για την εκτέλεση της διεργασίας. Οι παράγοντες αυτοί μπορεί να σχετίζονται:

- με την ίδια τη διεργασία, π.χ. απόσταση που πρέπει να καλυφθεί ή μέγεθος των αντικειμένων που θα χειριστεί ο χρήστης,
- με το ευρύτερο περιβάλλον χρήσης, π.χ. εμπόδια, ορατότητα κ.ά.,
- με τις απαιτήσεις και τις ικανότητες των χρηστών αλλά και
- με τους περιορισμούς και τις δυνατότητες του συστήματος, π.χ. μέγεθος απεικόνισης.

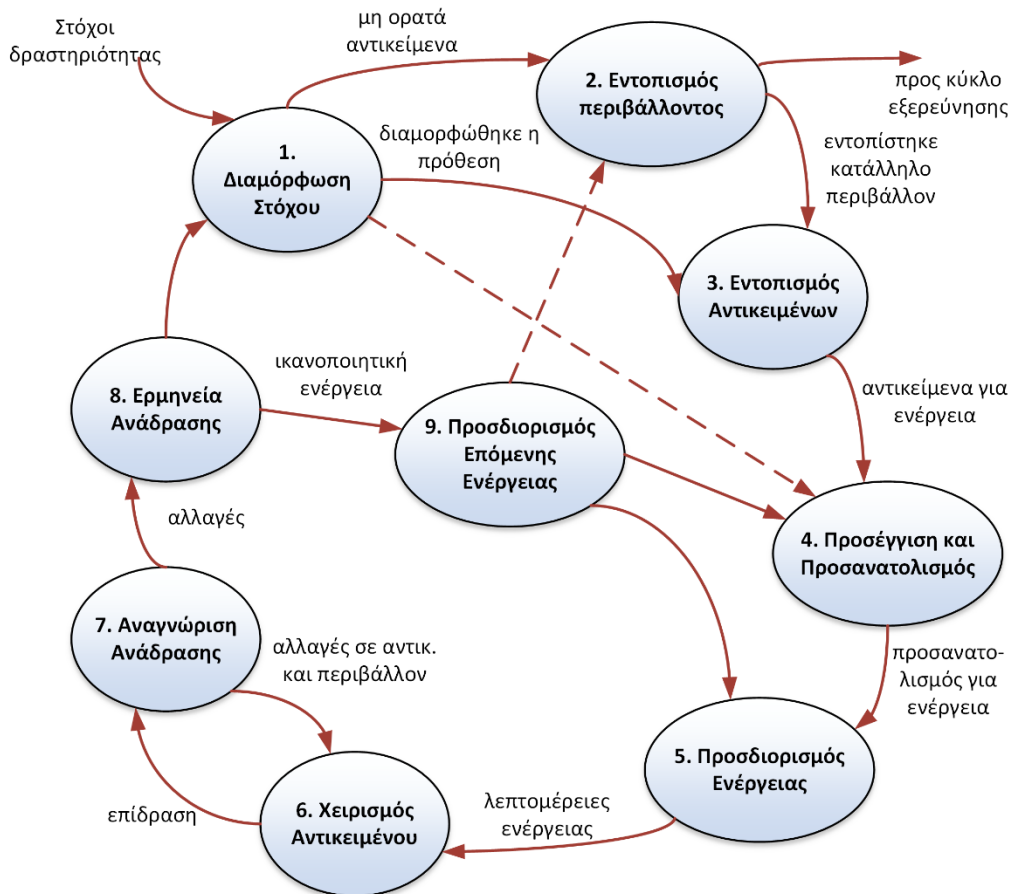
Στο επόμενο στάδιο προσδιορίζονται οι *μετρικές απόδοσης*, δηλαδή τα κριτήρια βάσει των οποίων θα προκύπτουν συμπεράσματα σχετικά με την απόδοση της τεχνικής. Τα πιο συνηθισμένα κριτήρια είναι η ταχύτητα και η ακρίβεια, αλλά μπορούν να συμπεριλαμβάνονται και άλλα, περισσότερο υποκειμενικά κριτήρια, όπως η ευκολία χρήσης, η ευκολία εκμάθησης και η άνεση των χρηστών.

Ακολουθεί το στάδιο των πειραμάτων. Η σχεδιαστική ομάδα αναπτύσσει ένα ειδικό περιβάλλον για τη διεξαγωγή των μετρήσεων, την πλατφόρμα δοκιμών. Το περιβάλλον αυτό θα πρέπει να μπορεί να υποστηρίξει την εκτέλεση των διεργασιών με τη χρήση όλων των τεχνικών οι οποίες θα αξιολογηθούν, την τροποποίηση των εξωτερικών παραγόντων και την καταγραφή των απαραίτητων δεδομένων για τις μετρικές απόδοσης. Χρησιμοποιείται ένας μεγάλος αριθμός χρηστών, οι οποίοι καλούνται να εκτελέσουν τις διεργασίες και συλλέγονται τα αντίστοιχα δεδομένα απόδοσης. Στα πειράματα με τους χρήστες δοκιμάζονται διαφορετικές τεχνικές αλληλεπίδρασης καθώς και τροποποιήσεις στους εξωτερικούς παράγοντες, ώστε να προκύψουν πλούσια αποτελέσματα που θα επιτρέψουν την εξαγωγή αξιοποιήσιμων συμπερασμάτων.

Τα αποτελέσματα των πειραμάτων είναι μοντέλα που χαρακτηρίζουν την ευχρηστία τεχνικών διάδρασης για συγκεκριμένες διεργασίες. Η ευχρηστία κάθε τεχνικής περιγράφεται ως πολλαπλές μετρικές αξιολόγησης συσχετιζόμενες με διάφορα επίπεδα εξωτερικών παραγόντων. Τα αποτελέσματα των αξιολογήσεων αυτού του τύπου μπορούν να δημιουργήσουν μια πλούσια βάση δεδομένων για την κατηγοριοποίηση τεχνικών διάδρασης, η οποία θα εμπλουτίζεται κάθε φορά που θα δοκιμάζονται νέες τεχνικές ή θα μελετώνται νέες διεργασίες. Επιπλέον, από τη μελέτη των αποτελεσμάτων μπορούν να προκύψουν και νέες ευρετικές και σχεδιαστικές κατευθύνσεις ευχρηστίας, οι οποίες συσχετίζουν τεχνικές διάδρασης με διεργασίες και εξωτερικούς παράγοντες.

Η σχεδίαση μιας νέας εφαρμογής, σύμφωνα με την παραπάνω μεθοδολογία, μπορεί να βασιστεί στα αποτελέσματα και στα συμπεράσματα που προέκυψαν από τα πειράματα στην πλατφόρμα δοκιμών. Όταν η σχεδιαστική ομάδα χρειαστεί να επιλέξει κατάλληλες σχεδιαστικές τεχνικές για την υποστήριξη των διεργασιών μιας εφαρμογής, μπορεί να αναζητήσει στα αποτελέσματα προηγούμενων αξιολογήσεων τις τεχνικές εκείνες οι οποίες είναι καταλληλότερες για το είδος των διεργασιών και τα επίπεδα εξωτερικών παραγόντων της εφαρμογής.

Η μεθοδολογία των Bowman & Hodges έχει το πλεονέκτημα ότι τα αποτελέσματα των αξιολογήσεων είναι αποσυνδεδεμένα από συγκεκριμένη εφαρμογή και κατά συνέπεια περισσότερο γενικεύσιμα και επαναχρησιμοποιήσιμα. Από την άλλη μεριά, απαιτείται η εμπλοκή πολύ μεγάλου αριθμού χρηστών και η διεξαγωγή πολλαπλών πειραμάτων για την παραγωγή χρήσιμων αποτελεσμάτων, κάτι που ενδέχεται να έχει σημαντικό χρονικό και οικονομικό κόστος για μια ομάδα ανάπτυξης.



Εικόνα 5.7 Μοντέλο δραστηριοτήτων χρηστών, σύμφωνα με τη μέθοδο γνωστικής περιδιάβασης των Sutcliffe & Kaur.

5.4 Γνωστική περιδιάβαση (Sutcliffe & Kaur)

Οι Sutcliffe & Kaur (2000) πρότείνουν μία μέθοδο αξιολόγησης διεπαφών εικονικής πραγματικότητας η οποία δεν απαιτεί τη συμμετοχή αντιπροσωπευτικών χρηστών. Η μέθοδος βασίζεται στην τεχνική της γνωστικής περιδιάβασης (*cognitive walkthrough*) και για την εκτέλεσή της έχουν προσδιοριστεί τρία βασικά μοντέλα

αλληλεπίδρασης χρηστών σε εικονικά περιβάλλοντα: Το πρώτο έχει να κάνει με τις δραστηριότητες που εκτελούν οι χρήστες για την εκτέλεση των διεργασιών τους στο περιβάλλον (εικ. 5.7), το δεύτερο με την απόκριση των χρηστών σε πρωτοβουλίες του συστήματος και το τρίτο με την πλοήγηση και εξερεύνηση των χρηστών στο περιβάλλον.

Σύμφωνα με το μοντέλο εκτέλεσης διεργασιών, ο χρήστης αρχικά διαμορφώνει τους στόχους του και εντοπίζει τα αντικείμενα που μπορούν να τον βοηθήσουν στις διεργασίες του. Στη συνέχεια προσεγγίζει ένα αντικείμενο, προσδιορίζει την ενέργεια που θέλει να κάνει σε αυτό και το χειρίζεται. Κατά τη διάρκεια του χειρισμού του παρακολουθεί και ερμηνεύει την ανάδραση του αντικειμένου ή και του περιβάλλοντος μέχρι το σημείο που θα αναγνωρίσει ότι η ενέργειά του ολοκληρώθηκε με επιτυχία. Στην περίπτωση αυτήν επιλέγει την επόμενη ενέργεια στο ίδιο αντικείμενο ή εντοπίζει το επόμενο αντικείμενο και επαναλαμβάνει τα ίδια βήματα μέχρι την ολοκλήρωση όλων των ενεργειών που απαιτούνται για την επίτευξη του στόχου του.

Στο μοντέλο απόκρισης σε πρωτοβουλίες του συστήματος, ο χρήστης αρχικά αναγνωρίζει ότι το σύστημα έχει τον έλεγχο. Στη συνέχεια παρατηρεί και ερμηνεύει τις αλλαγές στο περιβάλλον. Μετά από κάθε ερμηνεία αποφασίζει κατά πόσο θα πρέπει να αποκριθεί και να αναλάβει ξανά τον έλεγχο ή να συνεχίσει την παθητική παρατήρηση. Στην πρώτη περίπτωση εκτελεί κάποια ενέργεια ανάκτησης του ελέγχου και μεταφέρεται στο αρχικό μοντέλο εκτέλεσης διεργασιών.

Τέλος, στο μοντέλο εξερεύνησης και πλοήγησης, ο χρήστης καθορίζει τη διεύθυνση της κίνησής του για τον εντοπισμό κάποιου στόχου. Αν ο στόχος είναι γνωστός εκ των προτέρων, προσπαθεί αρχικά να εντοπίσει ένα μονοπάτι προς αυτόν, αλλιώς εξερευνά το περιβάλλον. Κατά την κίνησή του ερμηνεύει τις αλλαγές στη θέση του και στην οπτική του και μπορεί είτε να παρατηρήσει ενδιαφέροντα αντικείμενα που μπορούν να αποτελέσουν μελλοντικούς στόχους είτε να εντοπίσει το αντικείμενο-στόχο. Αν δεν συμβεί τίποτα από τα δύο, μπορεί να αποφασίσει στη συνέχεια να αλλάξει τη διεύθυνσή του, ώστε να εξερευνήσει άλλες περιοχές του περιβάλλοντος.

Η μέθοδος αξιολόγησης περιλαμβάνει τις παρακάτω φάσεις:

1. Αρχικά προετοιμάζονται σενάρια που περιγράφουν τυπικές διεργασίες χρηστών και συμπεριλαμβάνουν στόχους και ενέργειες που θα εκτελεστούν, γενική πλοήγηση και εξερεύνηση στο περιβάλλον και αντιδράσεις των χρηστών σε ενέργειες του συστήματος.
2. Στο επόμενο βήμα οι αξιολογητές διασχίζουν καθένα από τα παραπάνω σενάρια ακολουθώντας το κατάλληλο από τα τρία μοντέλα αλληλεπίδρασης. Σε κάθε ενέργεια του μοντέλου θέτουν ερωτήσεις σχετικές με την υποστήριξή της από τη διεπαφή που αξιολογείται. Στον πίνακα 5.1 παρουσιάζονται ορισμένες ενδεικτικές ερωτήσεις.
3. Η σχεδιασμένη διεπαφή αξιολογείται βάσει του κατά πόσο απαντάει ικανοποιητικά σε κάθε ερώτηση. Εάν εντοπιστούν προβλήματα, αυτά καταγράφονται ως σχεδιαστικές ελλείψεις του περιβάλλοντος.
4. Συνεχίζεται η περιδιάβαση μέχρι την ολοκλήρωση όλων των σεναρίων και συγκεντρώνονται και ομαδοποιούνται όλα τα σχεδιαστικά προβλήματα που εντοπίστηκαν. Από αυτά προκύπτει μια λίστα από ελλιπή στοιχεία και χαρακτηριστικά του περιβάλλοντος.
5. Τα προβλήματα προτεραιοποιούνται βάσει υποκειμενικής εκτίμησης της κρισιμότητάς τους.

Στάδιο Ενέργειας	Ερώτηση περιδιάβασης
Διαμόρφωση στόχου	Μπορεί ο χρήστης να θυμηθεί ή να διατυπώσει τον στόχο;
Εντοπισμός ενεργού περιβάλλοντος	Είναι τα κατάλληλα αντικείμενα ή τμήματα του περιβάλλοντος ορατά;
Εντοπισμός αντικειμένων	Μπορούν να εντοπιστούν τα απαραίτητα αντικείμενα;
Προσέγγιση και προσανατολισμός	Μπορεί ο χρήστης να προσεγγίσει και να προσανατολιστεί κατάλληλα ώστε να εκτελέσει την απαιτούμενη ενέργεια;
Συγκεκριμένη ενέργεια	Μπορεί ο χρήστης να αποφασίσει ποια ενέργεια θα εκτελέσει και πώς;
Χειρισμός αντικειμένων	Μπορεί ο χρήστης να χειριστεί εύκολα το αντικείμενο;
Αναγνώριση ανάδρασης	Είναι εμφανής η συνέπεια της ενέργειας;
Αξιολόγηση ανάδρασης	Μπορεί ο χρήστης να ερμηνεύσει την αλλαγή;
Προσδιορισμός επόμενης ενέργειας	Μπορεί ο χρήστης να αποφασίσει τι θα κάνει μετά;

Πίνακας 5.1 Ενδεικτικές ερωτήσεις της μεθόδου γνωστικής περιδιάβασης των Sutcliffe & Kaur.

Το αποτέλεσμα της παραπάνω μεθόδου είναι ο εντοπισμός κρίσιμων προβλημάτων και η διαμόρφωση μιας σχεδιασμένης διεπαφής με στόχο τη βελτίωση της ευχρηστίας της. Το θετικό στοιχείο με τη μέθοδο γνωστικής περιδιάβασης των Sutcliffe & Kaur είναι ότι η αξιολόγηση δεν απαιτεί τη συμμετοχή αντιπροσωπευτικών χρηστών και μπορεί να πραγματοποιηθεί από τα πρώτα βήματα της σχεδίασης χωρίς να είναι απαραίτητη η κατασκευή ενός υψηλής πιστότητας πρωτοτύπου. Ένα μειονέκτημα σε σχέση με τους εικονικούς κόσμους είναι ότι στα μοντέλα αλληλεπίδρασης της μεθόδου δεν περιλαμβάνονται δραστηριότητες επικοινωνίας και συνεργασίας με άλλους χρήστες, οι οποίες είναι ιδιαίτερα συχνές σε τέτοια περιβάλλοντα.

6 Συμπεράσματα

Όπως προκύπτει από το παρόν κεφάλαιο, η ανάπτυξη ενός εικονικού κόσμου, είτε εκ νέου είτε ως σχεδιασμένου περιβάλλοντος εντός μιας υπάρχουσας πλατφόρμας, θα πρέπει να βασίζεται σε πολύ προσεκτικά και καλά σχεδιασμένα βήματα, ώστε να είναι σε θέση να οδηγήσει σε ένα ικανοποιητικό αποτέλεσμα. Είδαμε ότι οι εικονικοί κόσμοι ως μέσο δεν είναι κατάλληλοι για όλων των ειδών τις εφαρμογές. Είναι συνεπώς απαραίτητο να μελετήσει η ομάδα ανάπτυξης τις ανάγκες των χρηστών και τις ιδιαιτερότητες του χώρου εφαρμογής και να αποφασίσει τον βαθμό στον οποίο οι εικονικοί κόσμοι πλεονεκτούν σε σχέση με εναλλακτικές λύσεις. Η καλή κατανόηση τόσο των αναγκών της εφαρμογής όσο και του τρόπου αξιοποίησης του μέσου για την επίτευξη των στόχων μπορεί να κατευθύνει την ομάδα σε ορισμένες βασικές σχεδιαστικές επιλογές. Εφόσον γίνει σαφές το είδος των δραστηριοτήτων που θα υποστηρίζονται, για παράδειγμα σχεδίαση, επικοινωνία, ανταγωνισμός κ.λπ., και το κατά πόσο ο βασικός στόχος θα είναι η διευκόλυνση των διεργασιών ή η προσομοίωση ενός ρεαλιστικού περιβάλλοντος, μπορούν να προκύψουν κατάλληλες μεταφορές αλληλεπίδρασης και αναπαραστάσεις του περιβάλλοντος. Από εκεί και πέρα, το πώς αυτές οι αρχικές επιλογές θα μετασχηματιστούν σε ένα αναλυτικά σχεδιασμένο περιβάλλον σχετίζεται σε μεγάλο βαθμό με τις δυνατότητες της ομάδας ανάπτυξης και τις λειτουργίες της πλατφόρμας που θα υποστηρίξει το περιβάλλον. Όπως είδαμε και στις σχετικές μεθοδολογίες, η τελική απόδοση του συστήματος παίζει κρίσιμο ρόλο στην επιτυχία του αποτελέσματος και θα πρέπει να γίνονται σχετικές μελέτες και μετρήσεις από τα πρώτα στάδια της σχεδίασης. Επιπλέον, εκτός από τον προσδιορισμό των στοιχείων του περιβάλλοντος, θα πρέπει να λαμβάνεται υπόψη και η εμπειρία του χρήστη σε αυτό. Θα πρέπει, δηλαδή, κατά τη σχεδίαση να μελετώνται σχετικά σενάρια χρήσης και αλληλεπίδρασης με το περιβάλλον, ώστε να διασφαλιστεί ότι καλύπτονται οι στόχοι της εφαρμογής στο αισθητικό και λειτουργικό επίπεδο. Τέλος, ίσως το πιο κρίσιμο σημείο για την επιτυχία μιας εφαρμογής να είναι η διασφάλιση της ευχρηστίας της, κάτι που είναι δυνατό να συμβεί μόνο μέσω συστηματικών διαδικασιών αξιολόγησης και μελέτης σχετικών σχεδιαστικών οδηγιών. Μέχρι σήμερα οι σχετικές μέθοδοι και τεχνικές προέρχονται κυρίως από τον χώρο της εικονικής πραγματικότητας, όμως μπορούν εύκολα να προσαρμοστούν στους εικονικούς κόσμους και να εμπλουτιστούν περαιτέρω μέσα από τη συστηματική μελέτη και χρήση του μέσου και την καταγραφή προβλημάτων ευχρηστίας.

Αναφορές

- Abras, C., Maloney-Krichmar, D. & Preece, J. (2004). User-centered design. *Bainbridge, W. Encyclopedia of Human-Computer Interaction. Thousand Oaks: Sage Publications, 37(4), 445–56.*
- Barnard, Y. F. & Sandberg, J. C. (1994). *THE THINK ALOUD METHOD A practical guide to modelling cognitive processes.*
- Barrilleaux, J. (2001). *3D User Interfaces with Java 3D. Interfaces.* Manning.
- Benford, S., Greenhalgh, C., Rodden, T. & Pycocock, J. (2001). Collaborative virtual environments. *Communications of the ACM, 44(7), 79–85.*

- Bourke, P. (2008). Evaluating Second Life as a tool for collaborative scientific visualisation. In Proc. of *Computer Games and Allied Technology Conference*.
- Bowman, D. A & Hodges, L. F. (1999). Formalizing the Design, Evaluation, and Application of Interaction Techniques for Immersive Virtual Environments. *Journal of Visual Languages and Computing*, 37–53.
- Bowman, D. A., Gabbard, J. L. & Hix, D. (2002). A Survey of Usability Evaluation in Virtual Environments: Classification and Comparison of Methods. *Presence: Teleoperators and Virtual Environments*, 11(4), 404–424.
- Bowman, D. A., North, C., Chen, J., Polys, N. F., Pyla, P. S. & Yilmaz, U. (2003). Information-rich virtual environments. *Proceedings of the ACM Symposium on Virtual Reality Software and Technology - VRST '03*, 81.
- Bryson, S. (1994). Approaches to the Successful Design and Implementation of VR Applications. *Virtual Reality Applications*, 3–15.
- Bryson, S. (1996). Virtual reality in scientific visualization. *Communications of the ACM*, 39(5), 62–71.
- Fencott, C. (1999). Towards a design methodology for virtual environments. *Workshop on User Centered Design and Implementation of Virtual Environments. University of York*, 1–8.
- Gabbard, J. L. (1997). *A Taxonomy of Usability Characteristics in Virtual Environments*. Virginia Polytechnic Institute and State University.
- Gabbard, J. L., Hix, D. & Swan, J.E., I. (1999). User-centered design and evaluation of virtual environments. In *IEEE Computer Graphics and Applications* (Vol. 19, pp. 2–10).
- Mack, R. & Nielsen, J. (1993). Usability inspection methods. *ACM SIGCHI Bulletin*, 25(1), 28–33.
- Sherman, W. & Craig, A. (2003). *Understanding Virtual Reality*. Morgan Kaufmann.
- Stanney, K. M., Mollaghasemi, M., Reeves, L., Breaux, R. & Graeber, D. a. (2003). *Usability engineering of virtual environments (VEs): Identifying multiple criteria that drive effective VE system design. International Journal of Human Computer Studies* (Vol. 58).
- Sutcliffe, A. G. & Kaur, K. D. (2000). Evaluating the usability of virtual reality user interfaces. *Behaviour & Information Technology*, 19(6), 415–426.
- Tanriverdi, V. & Jacob, R. J. K. (2001). VRID: a design model and methodology for developing virtual reality interfaces. *VRST '01: Proceedings of the ACM Symposium on Virtual Reality Software and Technology*, 175–182.
- Van Tol, R. & Huiberts, R. (2008). IEZA: A Framework For Game Audio. Ανακτήθηκε από: http://www.gamasutra.com/view/feature/3509/ieza_a_framework_for_game_audio.php
- Vosinakis, S., Koutsabasis, P., Stavrakis, M., Viorres, N. & Darzentas, J. (2008). Virtual environments for collaborative design: requirements and guidelines from a social action perspective. *CoDesign*, 4(3), 133–150.
- Wingrave, C. A & Laviola, J. J. (2010). Reflecting on the Design and Implementation Issues of Virtual Environments. *Presence*, 19(2), 179–195.

Περαιτέρω Μελέτη

Μπορείτε να διαβάσετε περισσότερα σχετικά με την ευχρηστία και τις διαδικασίες ανθρωποκεντρικής σχεδίασης διαδραστικών συστημάτων στα βιβλία:

Dix, A, Finlay, J, Abowd, G. & Beale, R. (2004) *Επικοινωνία Ανθρώπου-Υπολογιστή*. Τρίτη έκδοση, Μ. Γκιούρδας

Κουτσαμπάσης, Π. (2011) *Αλληλεπίδραση Ανθρώπου-Υπολογιστή*. Κλειδάριθμος.

Επίσης, αναλυτικές κατευθύνσεις και παραδείγματα σχεδίασης περιεχομένου σε περιβάλλοντα όπως το Second Life και OpenSimulator μπορείτε να βρείτε στο βιβλίο:

Curworth, A.L. (2014) *Virtual World Design*. CRC press.

Ερωτήσεις Κατανόησης

1. Ποιες είναι οι ιδιαιτερότητες στη σχεδίαση εφαρμογών βασισμένων σε εικονικούς κόσμους σε σχέση με την παραδοσιακή σχεδίαση εφαρμογών;
2. Πώς μπορούμε να αποφασίσουμε αν οι εικονικοί κόσμοι είναι κατάλληλο μέσο για την εφαρμογή που σχεδιάζουμε;
3. Περιγράψτε με δικά σας λόγια τη μεθοδολογία «συνάντησης στη μέση» του S. Bryson.
4. Εξηγήστε τη διαφορά της αντιληπτικής από τη δομική μοντελοποίηση στη μεθοδολογία του Fencott.
5. Ποια είναι τα επιμέρους λειτουργικά τμήματα ενός αντικειμένου διεπαφής σύμφωνα με τη μοντελοποίηση της μεθοδολογίας VRID?
6. Αναφέρετε περιπτώσεις εφαρμογών για τις οποίες απαιτούνται και μη ρεαλιστικές μορφές οπτικής αναπαράστασης.
7. Πώς μπορεί να αξιοποιηθεί ο ήχος στα τρισδιάστατα περιβάλλοντα;
8. Εξηγήστε με παραδείγματα την έννοια της «αντικατάστασης αισθήσεων».
9. Ποιες είναι οι βασικές διαφορές των εφαρμογών σχεδίασης και προσομοίωσης σε ό,τι αφορά τους στόχους των χρηστών και πώς αυτές επηρεάζουν την επιλογή τεχνικών πλοήγησης και χειρισμού;
10. Με ποιες τεχνικές αλληλεπίδρασης και μορφές αναπαράστασης μπορούμε να υποστηρίξουμε συνεργατικές δραστηριότητες;
11. Αναφέρετε ορισμένες βασικές δυσκολίες εφαρμογής παραδοσιακών μεθόδων αξιολόγησης στους εικονικούς κόσμους.
12. Ποια είναι η διαφορά της διαμορφωτικής από τη συμπερασματική συγκριτική αξιολόγηση στη μεθοδολογία ακολουθιακής αξιολόγησης;
13. Περιγράψτε με δικά σας λόγια την αξιολόγηση με πλατφόρμα δοκιμών.
14. Πώς μπορεί να αξιολογηθεί ένας σχεδιασμένος εικονικός κόσμος με τη μέθοδο της γνωστικής περιδιάβασης των Sutcliffe & Kaur;

Ασκήσεις

1. Εντοπίστε κάποιον σχεδιασμένο χώρο στο Second Life και προετοιμάστε μια απλή διαμορφωτική αξιολόγησή του: σχεδιάστε ένα ενδεικτικό σενάριο χρήσης, καλέστε χρήστες χωρίς προηγούμενη εμπειρία στο περιβάλλον να το εκτελέσουν και καταγράψτε τα προβλήματα ευχρηστίας που προέκυψαν. Προσπαθήστε να εντοπίσετε κατά πόσο αυτά σχετίζονται με τη διεπαφή του Second Life ή με τη σχεδιασμένη περιοχή και προτείνετε πιθανές βελτιώσεις.

2. Επαναλάβετε την παραπάνω διαδικασία χωρίς χρήστες, χρησιμοποιώντας τη μέθοδο γνωστικής περιδιάβασης των Sutcliffe & Kaur. Συγκρίνετε τα αποτελέσματά σας.
3. Επιλέξτε μια ιστορική περίοδο και σχεδιάστε έναν εικονικό κόσμο για την αναβίωσή της με τη συμμετοχή χρηστών και συνθετικών χαρακτήρων. Ακολουθήστε τα τρία πρώτα στάδια της μεθοδολογίας του Fencott και καταγράψτε τα αποτελέσματά σας.
4. Σχεδιάστε έναν εκπαιδευτικό εικονικό κόσμο για εκμάθηση βασικών εννοιών φυσικής. Επιλέξτε κατάλληλες τεχνικές αλληλεπίδρασης και μοντέλα αναπαράστασης και τεκμηριώστε τις επιλογές σας. Περιγράψτε ένα ενδεικτικό σενάριο χρήσης και εντοπίστε πιθανά ζητήματα ευχρηστίας με τη μέθοδο της γνωστικής περιδιάβασης.

Κεφάλαιο 6: Εφαρμογές

Σύνοψη

Αντικείμενο του κεφαλαίου είναι η παρουσίαση των διάφορων χώρων εφαρμογής των εικονικών κόσμων και η κατανόηση των ειδικών απαιτήσεων των περιοχών αυτών και των αντίστοιχων σχεδιαστικών ζητημάτων που εγείρονται. Κατηγοριοποιούνται και παρουσιάζονται ενδεικτικές εφαρμογές στις παρακάτω περιοχές: Εκπαίδευση, Πολιτισμός, Σχεδίαση και Επιχειρείν. Για καθεμία από τις περιοχές αυτές αναλύονται οι ιδιαίτερες απαιτήσεις και τα χαρακτηριστικά στα οποία δίνεται έμφαση κατά τη σχεδίαση και κυρίως το πώς τα στοιχεία αυτά επηρεάζουν τελικά την επιλογή της τεχνολογίας και τις σχεδιαστικές αποφάσεις. Παρουσιάζονται συνοπτικά οι σχεδιαστικές λύσεις που έχουν δοθεί στις περιοχές αυτές και προτείνονται καλές πρακτικές για την αξιοποίηση των προσφερόμενων δυνατοτήτων των εικονικών κόσμων.

Προαπαιτούμενη γνώση

Απαιτούνται βασικές γνώσεις πληροφορικής και κατανόηση των βασικών εννοιών των εικονικών κόσμων που παρουσιάζονται στο 1^ο Κεφάλαιο.

1 Εισαγωγή

Όπως αναφέραμε και στο πρώτο κεφάλαιο του βιβλίου, υπάρχουν πολλοί και διαφορετικοί χώροι εφαρμογής οι οποίοι θα μπορούσαν να επωφεληθούν από τις δυνατότητες των εικονικών κόσμων. Σε κάποιους από αυτούς τους χώρους οι εικονικοί κόσμοι έχουν ήδη προσφέρει σημαντικά αποτελέσματα, ενώ σε άλλους οι εφαρμογές δεν έχουν φτάσει ακόμα στο απαιτούμενο επίπεδο ωριμότητας παρότι διαφαίνεται μια δυναμική. Αναμφίβολα η πιο ευρεία και επιτυχημένη περιοχή εφαρμογής είναι η ψυχαγωγία και τα παιχνίδια. Άλλωστε οι περισσότεροι από τους δημοφιλείς εικονικούς κόσμους με πολύ μεγάλη επισκεψιμότητα, όπως το World of Warcraft, έχουν αναπτυχθεί με αποκλειστικό στόχο την ψυχαγωγία των παικτών. Πέρα από τα παιχνίδια, μπορούν να εντοπιστούν και αρκετές άλλες περιοχές εφαρμογής με σημαντικές επιτυχίες. Ο χώρος της εκπαίδευσης με τη χρήση τεχνολογιών πληροφορικής έχει αναγνωρίσει από πολύ νωρίς την αξία των εικονικών κόσμων ως εκπαιδευτικά περιβάλλοντα και η διεθνής επιστημονική κοινότητα έχει επενδύσει σημαντικό ερευνητικό κεφάλαιο στη μελέτη των δυνατοτήτων του μέσου και στη διαμόρφωση κατευθύνσεων για την καλύτερη εκμετάλλευσή του. Και στον χώρο του πολιτισμού όμως, όπου υπάρχει η ανάγκη για πλούσιες διαδραστικές αναπαραστάσεις, φαίνεται να προκύπτουν σημαντικά οφέλη από την εκμετάλλευση των τρισδιάστατων πολυχρηστικών περιβαλλόντων και ήδη είναι διαθέσιμες αρκετές σχετικές εφαρμογές. Ο χώρος της σχεδίασης είναι μια ακόμα περιοχή η οποία μπορεί να αξιοποιήσει εποικοδομητικά τις δυνατότητες των εικονικών κόσμων για εύκολη και συνεργατική διαμόρφωση χώρων και αντικειμένων, και φαίνεται να υπάρχει ενδιαφέρον τόσο από τους σχεδιαστές όσο και από την επιστημονική κοινότητα για την ενίσχυση του μέσου προς αυτήν την κατεύθυνση. Τέλος, η μεγάλη επισκεψιμότητα και συμμετοχικότητα που παρατηρείται στους δημοφιλείς εικονικούς κόσμους δεν θα μπορούσε να αφήσει αδιάφορο τον χώρο του ηλεκτρονικού επιχειρείν. Αρκετοί εικονικοί κόσμοι, με χαρακτηριστικότερη περίπτωση το Second Life, έχουν αξιοποιηθεί για διαφήμιση και προώθηση προϊόντων και υπηρεσιών τόσο εντός όσο και εκτός του κόσμου.

Αντικείμενο αυτού του κεφαλαίου είναι η μελέτη ορισμένων βασικών περιοχών εφαρμογής των εικονικών κόσμων με στόχο την ανάδειξη καλών πρακτικών και σχεδιαστικών λύσεων σε καθεμία από αυτές. Σκόπιμα στις περιοχές που θα αναφερθούμε δεν περιλαμβάνονται τα πολυχρηστικά παιχνίδια πολλαπλών χρηστών, καθώς η σχεδίασή τους ακολουθεί εν πολλοίς τις βασικές αρχές και κατευθύνσεις του χώρου των ηλεκτρονικών παιχνιδιών και καλύπτεται σε μεγάλο εύρος από πληθώρα σχετικών βιβλίων (π.χ. Bartle, 2004, Rollings & Adams, 2003). Είναι άλλωστε τόσες πολλές οι σχεδιαστικές πτυχές ενός πολυχρηστικού παιχνιδιού, όπως η εξιστόρηση, οι χαρακτήρες, το gameplay, η αισθητική του περιβάλλοντος, η κοινωνική διάδραση κ.λπ., που θα ήταν αδύνατο να καλυφθούν ικανοποιητικά στις σελίδες αυτού του κεφαλαίου. Οι χώροι εφαρμογής που εξετάζουμε είναι: η εκπαίδευση, ο πολιτισμός, η σχεδίαση και το επιχειρείν. Για καθέναν από τους παραπάνω χώρους θα μελετήσουμε τις ειδικές απαιτήσεις της περιοχής και τις προσφερόμενες δυνατότητες των εικονικών κόσμων σε σχέση με τις απαιτήσεις αυτές. Θα επιχειρήσουμε μια επισκόπηση και ταξινόμηση των εφαρμογών των εικονικών κόσμων σε κάθε περιοχή και θα εστιάσουμε σε ενδεικτικές περιπτώσεις επιτυχημένων εφαρμογών. Τέλος, θα καταγράψουμε ορισμένες καλές σχεδιαστικές πρακτικές και στοιχεία

επιτυχίας με στόχο την καλύτερη κατανόηση των δυνατοτήτων και περιορισμών του μέσου και την υποστήριξη μελλοντικών προσπαθειών.

2 Εκπαίδευση

Η εκπαίδευση είναι πιθανότατα ο χώρος όπου έχουν βρει οι εικονικοί κόσμοι τη μεγαλύτερη εφαρμογή μετά την ψυχαγωγία. Ήδη από τα πρώτα υλοποιημένα συστήματα εικονικής πραγματικότητας της δεκαετίας του '90 αναγνωρίστηκε η δυναμική του μέσου ως εκπαιδευτικού περιβάλλοντος και άρχισαν να αναδύονται σχετικές εφαρμογές. Αρχικά η έμφαση δόθηκε στην αξιοποίηση της τρισδιάστατης οπτικοποίησης και ελεύθερης περιήγησης που παρείχαν αυτά τα περιβάλλοντα, και οι εφαρμογές εστίαζαν στην προβολή επιστημονικών μοντέλων και σχετικών πληροφοριών. Μετά από κάποια χρόνια, όταν τα εικονικά περιβάλλοντα έγιναν περισσότερο προσβάσιμα και κυρίως πολυχρηστικά, άρχισαν να διαφαίνονται περαιτέρω δυνατότητες αξιοποίησής τους στην εκπαίδευση. Η διαμόρφωση προσωπικής ταυτότητας μέσω της ενσάρκωσης, οι πολλαπλές μορφές επικοινωνίας, η υποστήριξη εικονικών κοινοτήτων και η διασύνδεση των περιβαλλόντων αυτών με εξωτερικές πηγές είναι ορισμένα από τα στοιχεία που συνέβαλλαν σε μια διευρυμένη προσπάθεια εφαρμογής τους σε πληθώρα μαθησιακών αντικειμένων και με τη χρήση αρκετών διαφορετικών εκπαιδευτικών πρακτικών.

2.1 Εκπαιδευτικές προσεγγίσεις

Ο χώρος της εκπαίδευσης περιλαμβάνει έναν αριθμό από εκπαιδευτικές θεωρίες με αρκετά διαφορετικές αφετηρίες, οι οποίες μετασχηματίζονται σε αντίστοιχες εκπαιδευτικές πρακτικές. Η λεπτομερής παρουσίαση και ανάλυση των θεωριών αυτών ξεφεύγει από το αντικείμενο του παρόντος βιβλίου. Θα παρουσιάσουμε συνοπτικά ορισμένες βασικές από αυτές δίνοντας έμφαση στους πιθανούς τρόπους εφαρμογής τους στους εικονικούς κόσμους.

2.1.1 Εγκαθιδρυμένη μάθηση

Σύμφωνα με τη θεωρία της *εγκαθιδρυμένης μάθησης (Situated Learning)* (Brown κ.ά., 1989) η διδασκαλία θα πρέπει να γίνεται σε ένα περιβάλλον που δείχνει και συμπεριφέρεται όπως το αντίστοιχο περιβάλλον του αντικειμένου μάθησης στον πραγματικό κόσμο. Οι θεωρητικοί της εγκαθιδρυμένης μάθησης ισχυρίζονται ότι η μάθηση λαμβάνει χώρα εντός κάποιου συγκεκριμένου πλαισίου και ότι το πλαίσιο στο οποίο μαθαίνουμε μπορεί τελικά να διευκολύνει ή να εμποδίσει τη μάθηση. Αν λοιπόν υπάρχει μια καλή αντιστοίχιση μεταξύ του αντικειμένου της μάθησης και του πλαισίου εντός του οποίου αυτό διδάσκεται, τότε η αντιστοίχιση αυτή αναμένεται να ενισχύσει τη μάθηση. Ο διδασκόμενος που μαθαίνει σε ένα τέτοιο περιβάλλον κερδίζει όχι μόνο από το ίδιο το εκπαιδευτικό υλικό αλλά και από την καλύτερη κατανόηση της κουλτούρας του περιβάλλοντος, του λεξιλογίου που χρησιμοποιείται και των αναμενόμενων συμπεριφορών στο περιβάλλον αυτό. Από την άλλη μεριά, αν το πλαίσιο δεν είναι κατάλληλο, θα είναι δυσκολότερη η μεταφορά της γνώσης που αποκτήθηκε στον χώρο πραγματικής εφαρμογής της.

Η προσέγγιση της εγκαθιδρυμένης μάθησης ταιριάζει καλύτερα σε αντικείμενα που περιλαμβάνουν την απόκτηση δεξιοτήτων που θα εφαρμοστούν στον πραγματικό κόσμο. Αν, για παράδειγμα, κάποιος θέλει να μάθει να διδάσκει σε μια τάξη με παιδιά, ένας καλός τρόπος είναι να βρεθεί σε μία τέτοια τάξη, να παρακολουθήσει άλλους διδάσκοντες την ώρα του μαθήματος και να αναλάβει τελικά και ο ίδιος αυτόν τον ρόλο. Στην περίπτωση του παραδείγματός μας το πλαίσιο όχι απλά δείχνει και συμπεριφέρεται όπως το πραγματικό, αλλά είναι το πραγματικό. Αντίστοιχα, για να μάθει κάποιος να οδηγεί αυτοκίνητο δεν αρκεί να διαβάσει βιβλία ή να δει ταινίες σχετικά με το αυτοκίνητο. Ούτε καν να παίξει παιχνίδια οδήγησης τα οποία έχουν μια απλοποιημένη διεπαφή τόσο ως προς την απεικόνιση όσο και ως προς τον χειρισμό. Θα χρειαστεί να εξασκηθεί σε έναν καλό προσομοιωτή και αργότερα σε ένα αληθινό αυτοκίνητο με τη βοήθεια κάποιου εκπαιδευτή για να αποκτήσει τις απαιτούμενες δεξιότητες. Υπάρχουν όμως και αρκετές περιπτώσεις στις οποίες η ένταξη του εκπαιδευόμενου στο πραγματικό πλαίσιο μπορεί να μην είναι πάντα εφικτή για λόγους κόστους, διαθεσιμότητας ή επικινδυνότητας. Στις περιπτώσεις αυτές, εκπαιδευτικές τεχνολογίες όπως οι εικονικοί κόσμοι μπορούν να αποβούν εξαιρετικά χρήσιμες μέσα από την αξιοποίηση ρεαλιστικών αναπαραστάσεων και αλληλεπιδράσεων για την απόδοση του κατάλληλου περιβάλλοντος πλαισίου.

2.1.2 Εποικοδομισμός

Η θεωρία του *εποικοδομισμού* (*Constructivism*) βασίζεται στην ιδέα ότι η μάθηση είναι μια διαδικασία προοδευτικής κατανόησης κατά την οποία η γνώση «χτίζεται» από μέσα προς τα έξω. Σε αντίθεση με την κάποτε καθιερωμένη θεώρηση ότι υπάρχει μια «αντικειμενική» αλήθεια ανεξάρτητη από τους μαθητές και ότι οι μαθητές θα πρέπει να «απορροφήσουν» την αλήθεια αυτήν όσο το δυνατόν καλύτερα, ο εποικοδομισμός θεωρεί ότι η γνώση και η αλήθεια δεν είναι ούτε καθολικές ούτε αιώνιες. Αντίθετα, κάθε άνθρωπος κατασκευάζει τη δική του γνώση η οποία είναι μια ατομική άποψη της αλήθειας. Σύμφωνα με τη θεώρηση αυτή (Jonassen, 1999):

«τα άτομα αντιλαμβάνονται τον κόσμο και ο,τιδήποτε έρχονται σε επαφή μέσω της κατασκευής των δικών τους αναπαραστάσεων ή μοντέλων των εμπειριών τους... Η διδασκαλία δεν είναι απλά μια διαδικασία μετάδοσης γνώσης, διότι ο εκπαιδευόμενος δεν μπορεί να γνωρίζει όσα ο εκπαιδευτής γνωρίζει και αυτά που γνωρίζει ο εκπαιδευτής δεν μπορεί να τα μεταφέρει με κάποιο τρόπο στον εκπαιδευόμενο».

Αυτή η κατασκευή της γνώσης επιτυγχάνεται μέσω της διαμόρφωσης και ελέγχου υποθέσεων βασισμένων σε προηγούμενες εμπειρίες και της αλληλεπίδρασης με εξωτερικά «αντικείμενα». Με δεδομένο ότι κάθε άτομο έχει ένα λιγότερο ή περισσότερο διαφορετικό σύνολο προηγούμενων εμπειριών και ερμηνευτικές θεωρήσεις, καθένας θα έχει μια μοναδική κατασκευή γνώσης. Αυτό βεβαίως δεν σημαίνει ότι όλες οι κατασκευασμένες γνώσεις είναι ίσης αξίας. Η «βιωσιμότητα» της γνώσης του κάθε ατόμου καθορίζεται από το πόσο καλά λειτουργούν οι ιδέες του σε σχέση με τις ευρύτερες πεποιθήσεις μιας κοινότητας. Στόχος ενός εκπαιδευόμενου είναι η σταδιακή κατασκευή πολύπλοκων και βιώσιμων ατομικών μοντέλων κατανόησης του κόσμου.

Υπάρχουν αρκετές διαφορετικές διδακτικές προσεγγίσεις βασισμένες στον εποικοδομισμό. Η βασική ιδέα σε όλες τις περιπτώσεις είναι ότι οι μαθητές ανακαλύπτουν σταδιακά τη γνώση μέσα από τη δική τους ενεργή ενασχόληση, εξερεύνηση και συνεργασία. Η έμφαση λοιπόν δίνεται κυρίως σε δραστηριότητες όπως οι εργασίες και η μάθηση μέσω ανακάλυψης, όπου οι μαθητές καλούνται να εξερευνήσουν και να αποκτήσουν μόνοι τους τη γνώση μέσα από την προσπάθειά τους να επιτύχουν κάποιον στόχο ή να δώσουν απάντηση σε κάποιο ερώτημα. Οι εικονικοί κόσμοι έχουν τη δυνατότητα να υποστηρίξουν μαθησιακά περιβάλλοντα που απαιτούν την ενεργό εμπλοκή των μαθητών, καθώς προσφέρουν από τη φύση τους μεγάλη ελευθερία στους χρήστες να εξερευνούν τον χώρο, να αλληλεπιδρούν με τα αντικείμενα, να συναντούν και να επικοινωνούν με άλλους χρήστες και να παρατηρούν άμεσα τα αποτελέσματα των ενεργειών τους. Για τον λόγο αυτόν η επιστημονική κοινότητα εξετάζει περιπτώσεις εφαρμογής της θεωρίας του εποικοδομισμού από τα πρώτα εκπαιδευτικά περιβάλλοντα εικονικής πραγματικότητας (Dede, 1995) μέχρι τους εικονικούς κόσμους (Dickey, 2005) και η πλειοψηφία των εφαρμογών των εικονικών κόσμων στην εκπαίδευση φαίνεται να βασίζεται στον εποικοδομισμό ως βασικό θεωρητικό μοντέλο (Mikropoulos & Natsis, 2011).

2.1.3 Κοινωνικός Εποικοδομισμός

Μια θεωρία αρκετά δημοφιλής στους σχεδιαστές εικονικών κόσμων, η οποία είναι συγγενής τόσο με τον εποικοδομισμό όσο και με την εγκαθιδρυμένη μάθηση, είναι ο *κοινωνικός εποικοδομισμός*. Ο κοινωνικός εποικοδομισμός διατηρεί την κεντρική ιδέα της μαθητο-κεντρικής ανάπτυξης της γνώσης αλλά δίνει έμφαση στο στοιχείο της συνεργασίας. Η βασική ιδέα είναι ότι κανείς δεν μπορεί να μάθει κάτι μόνος του. Αντίθετα, μαθαίνουμε δοκιμάζοντας λύσεις, παίρνοντας ανάδραση από το περιβάλλον και από άλλους, τροποποιούμε την προσέγγισή μας με βάση την ανάδραση και δοκιμάζουμε ξανά. Η μάθηση υπό την οπτική του κοινωνικού εποικοδομισμού είναι μια κοινωνική δραστηριότητα που λαμβάνει χώρα εντός μιας κοινότητας ανθρώπων οι οποίοι μαθαίνουν νέα πράγματα αλλά και μεταλαμπαδεύουν την υπάρχουσα γνώση λειτουργώντας ως σύμβουλοι και καθοδηγητές των νέων μελών.

Η μάθηση σε ένα περιβάλλον κοινωνικού εποικοδομισμού περιλαμβάνει την ανάληψη διαφορετικών ρόλων. Για παράδειγμα, αν θέλει κάποιος να μάθει πώς να γίνει καλός δάσκαλος, θα πρέπει να αναλάβει τον ρόλο του δασκάλου και να το δοκιμάσει. Αυτό όμως δεν μπορεί να γίνει από ένα άτομο. Θα πρέπει ο μαθητευόμενος να δοκιμάσει τις διδακτικές του ικανότητες και να χτίσει τη γνώση του σχετικά με τη διδασκαλία με άλλους ανθρώπους οι οποίοι προσπαθούν να κάνουν το ίδιο. Αυτή η προσπάθεια θα γίνεται υπό την καθοδήγηση ατόμων ήδη εμπειρών στη διδασκαλία οι οποίοι θα συνεισφέρουν με τις απαραίτητες συμβουλές και ανατροφοδότηση. Οι εικονικοί κόσμοι θα μπορούσαν να αξιοποιηθούν με επιτυχία για την

υποστήριξη κοινοτήτων μάθησης, καθώς προσφέρουν τόσο τη δυνατότητα ανάληψης ρόλων και εκτέλεσής τους σε ένα ρεαλιστικό πλαίσιο όσο και ένα πλούσιο περιβάλλον σύγχρονης και ασύγχρονης επικοινωνίας που μπορεί να υποστηρίξει την ανάπτυξη κοινοτήτων.

2.1.4 Συμπεριφορισμός

Το μοντέλο του *συμπεριφορισμού* στηρίζεται στην ιδέα του «ερεθίσματος και της απόκρισης», δηλαδή ότι η συμπεριφορά μας τροποποιείται ανάλογα με τα ερεθίσματα που δεχόμαστε από το περιβάλλον. Η κύρια θεώρηση του μοντέλου αυτού είναι ότι μπορεί ο εκπαιδευόμενος να επιτύχει την επιθυμητή συμπεριφορά αν του παρέχονται κατάλληλες αρνητικές ή θετικές αποκρίσεις ανάλογα με τις επιδόσεις του. Επειδή ακριβώς ο συμπεριφορισμός βασίζεται στην αρχή ότι υπάρχει μια δεδομένη συμπεριφορά η οποία αποτελεί μαθησιακό στόχο, έχει κατηγορηθεί από τους εκφραστές των πιο σύγχρονων θεωριών ότι οδηγεί σε παθητική μάθηση και εντέλει σε στείρα απομνημόνευση. Παρόλα αυτά, ακόμα και τα ασθενώς ορισμένα μεγάλης κλίμακας προβλήματα που εμφανίζονται στην πλειοψηφία των μαθησιακών αντικειμένων, αν αναλυθούν σε μικρότερα τμήματα και αυτά σε ακόμα μικρότερα κ.λπ., καταλήγουμε σε μικρά υπο-προβλήματα τα οποία λύνονται με ένα ξεκάθαρο σύνολο ενεργειών. Επομένως, σε κάποιο βαθμό απαιτείται τελικά και η ικανότητα του εκπαιδευόμενου να μπορεί να επιλύει με επιτυχία μικρά, σαφώς ορισμένα προβλήματα για τα οποία η λύση είναι δεδομένη και «αντικειμενική». Επιπλέον, στα πρώτα τους βήματα στο σχολείο τα παιδιά αναγκάζονται να απομνημονεύσουν βασικές γνώσεις τις οποίες δεν είναι δυνατόν να αποκτήσουν με κάποιον άλλο τρόπο, όπως το αλφάβητο, η προπαίδεια, οι κανόνες ορθογραφίας κ.λπ.

Σε περιπτώσεις σαν αυτές που αναφέραμε παραπάνω, προσεγγίσεις βασισμένες στο μοντέλο του συμπεριφορισμού μπορούν να έχουν αξία. Ο μαθητευόμενος καλείται να δοκιμάσει τις ικανότητες ή τις γνώσεις του σε κάποιο αντικείμενο, οι επιδόσεις του αξιολογούνται και λαμβάνει θετική ή αρνητική ανάδραση ανάλογα με την επιτυχία του. Οι υπολογιστές μπορούν να υποστηρίξουν ικανοποιητικά τέτοια μοντέλα μάθησης μέσα από περιβάλλοντα εξάσκησης που δημιουργούν προβλήματα προοδευτικής δυσκολίας, αξιολογούν τις επιδόσεις των χρηστών και παρέχουν την κατάλληλη ανάδραση σε διάφορες μορφές. Τα τελευταία χρόνια αυτού του τύπου τα περιβάλλοντα εξάσκησης έχουν ενισχυθεί και με παιγνιώδη χαρακτηριστικά, ώστε να παρακινήσουν και να διατηρήσουν ζωντανό το ενδιαφέρον των συμμετεχόντων. Οι εκπαιδευτικές δραστηριότητες που βασίζονται στον συμπεριφορισμό μπορούν να μεταφερθούν και στους εικονικούς κόσμους μέσα από τη σχεδίαση διαδικασιών παρακίνησης, παρακολούθησης και αξιολόγησης των συμμετεχόντων. Οι προσεγγίσεις ποικίλλουν από απλά κουίζ και παιχνίδια μέχρι ανθρωπόμορφους εικονικούς εκπαιδευτές που επικοινωνούν με τους χρήστες σε φυσική γλώσσα (Soliman & Guetl, 2010).

2.2 Προσφερόμενες δυνατότητες

Οι προσφερόμενες δυνατότητες των εικονικών κόσμων πηγάζουν τόσο από την ικανότητά τους να παρουσιάσουν ρεαλιστικά ή υποθετικά περιβάλλοντα, όσο και από την ίδια τη συμμετοχή και επικοινωνία των χρηστών σε αυτά. Αποτελούν ένα σύγχρονο μέσο στο οποίο μπορούν να σχεδιαστεί και να διαμορφωθεί κατάλληλος χώρος για την ενσωμάτωση εκπαιδευτικών στοιχείων και την εκτέλεση δραστηριοτήτων που οδηγούν στη μάθηση. Όπως αναφέρθηκε και πιο πάνω, η μεγάλη ελευθερία επιλογών στη σχεδίαση των κόσμων και στις υποστηριζόμενες ενέργειες των χρηστών επιτρέπει την υλοποίηση διαφόρων εκπαιδευτικών πρακτικών που πηγάζουν από αρκετά διαφορετικές μεταξύ τους προσεγγίσεις.

Ακολουθούν ορισμένες από τις βασικές προσφερόμενες δυνατότητες των εικονικών κόσμων για την αξιοποίησή τους στο χώρο της εκπαίδευσης. (Warburton, 2009)

- *Πλούσιες διαδράσεις*: παρέχεται η δυνατότητα κοινωνικών διαδράσεων μεταξύ ατόμων και κοινοτήτων, διαδράσεις χρηστών με αντικείμενα, αλλά και ευφρείς διαδράσεις μεταξύ αντικειμένων
- *Οπτικοποίηση και τοποθέτηση σε εννοιολογικό πλαίσιο*: παραγωγή και αναπαραγωγή μη προσβάσιμου περιεχομένου το οποίο είναι ιστορικά χαμένο, απομακρυσμένο, ακριβό, φανταστικό, μελλοντικό ή αδύνατο να παρατηρηθεί με γυμνό μάτι
- Έκθεση σε *αυθεντικό περιεχόμενο και κουλτούρα* που παράγεται εντός των εικονικών κόσμων
- Ατομική και συλλογική *ανάληψη ρόλων*

- *Εμβύθιση* σε τρισδιάστατο περιβάλλον όπου η αίσθηση της παρουσίας μέσω της ενσάρκωσης και οι εκτεταμένες μορφές επικοινωνίας μπορούν να επιδράσουν στις συναισθηματικές πτυχές της εμπειρίας και να παρέχουν επιπλέον κίνητρα για συμμετοχή.
- *Προσομοίωση* για την αναπαραγωγή περιβαλλόντων που είναι πολύ ακριβά να κατασκευαστούν στον πραγματικό κόσμο και που μπορούν να παρακάμψουν ορισμένους φυσικούς περιορισμούς
- *Παρουσία σε κοινότητα*: δημιουργείται η αίσθηση του ανήκειν και του κοινού στόχου μέσα από τις ομάδες που σχηματίζονται στους εικονικούς κόσμους
- *Παραγωγή περιεχομένου*: ευκαιρίες δημιουργίας και ιδιοκτησίας αντικειμένων και μέρους του μαθησιακού περιβάλλοντος.



Εικόνα 6.1 Το Εικονικό Πανεπιστήμιο του Εδιμβούργου στο OpenSimulator.

2.3 Επισκόπηση εφαρμογών

Σύμφωνα με σχετικά πρόσφατη μελέτη (Hew & Cheung, 2010) ως προς το πλαίσιο χρήσης των εικονικών κόσμων στη δευτεροβάθμια και τριτοβάθμια εκπαίδευση, προέκυψαν τρεις βασικές κατηγορίες χρήσης του μέσου, οι οποίες σε αρκετές εφαρμογές μπορεί να συνδυάζονται:

- χώρος επικοινωνίας,
- προσομοίωση ενός χώρου και
- βιωματικός χώρος.

Η πρώτη κατηγορία αναφέρεται σε εφαρμογές που εκμεταλλεύονται τη δυνατότητα των εικονικών κόσμων να υποστηρίζουν τη σύγχρονη επικοινωνία πολλαπλών χρηστών συμπεριλαμβάνοντας τόσο λεκτικά όσο και μη λεκτικά χαρακτηριστικά. Η αξιοποίηση αυτής της δυνατότητας σε εκπαιδευτικά πλαίσια έγκειται κυρίως στη διοργάνωση και εκπόνηση ομαδικών δραστηριοτήτων που περιλαμβάνουν κάποιου είδους συνεργασία ή ανταλλαγή απόψεων. Για παράδειγμα, σε εκπαιδευτικά αντικείμενα όπως η εκμάθηση ξένων γλωσσών οι μαθητές μπορούν να συμμετέχουν εξ αποστάσεως σε παιγνιώδεις δραστηριότητες οι οποίες τους παρακινούν να κατανοήσουν και να εκφραστούν σε κάποια ξένη γλώσσα. Σε κάποιες περιπτώσεις μπορούν να συμμετέχουν και να επικοινωνούν μαζί τους και μαθητές που έχουν τη γλώσσα αυτή ως μητρική. Οι εφαρμογές αυτής της κατηγορίας μεταφράζουν συνήθως παραδοσιακές δράσεις που λαμβάνουν χώρα στην τάξη σε αντίστοιχες δραστηριότητες εντός του εικονικού κόσμου. Εκμεταλλεύονται με αυτόν τον τρόπο τα παιγνιώδη χαρακτηριστικά του περιβάλλοντος, τις δυνατότητες αυτοέκφρασης μέσω της ενσάρκωσης που προσφέρονται στους μαθητές και κυρίως την απομακρυσμένη πρόσβαση.

Στη δεύτερη κατηγορία περιλαμβάνονται οι εφαρμογές που παρουσιάζουν στους μαθητές ένα ρεαλιστικό περιβάλλον, πιθανότατα εμπλουτισμένο με διαδραστικά στοιχεία. Για παράδειγμα, πολλές εκπαιδευτικές περιοχές σε δημοφιλείς εικονικούς κόσμους όπως το Second Life ή το OpenSimulator περιέχουν

κατασκευές που απεικονίζουν υπαρκτούς πανεπιστημιακούς ή σχολικούς χώρους με μεγάλη λεπτομέρεια. Η προσέγγιση αυτή βοηθάει στην άμεση μεταφορά της οργάνωσης των διαδικασιών και της χρήσης των αιθουσών από τον φυσικό χώρο στον αντίστοιχο εικονικό. Επιπλέον, οι νέοι και οι μελλοντικοί σπουδαστές μπορούν να γνωρίσουν και να εξοικειωθούν ευκολότερα με τον φυσικό χώρο μέσω της εμπειρίας τους στον εικονικό, ενώ και οι εν ενεργεία φοιτητές αισθάνονται ότι βρίσκονται σε έναν χώρο εκπαίδευσης όταν ο εικονικός κόσμος στον οποίο συνυπάρχουν θυμίζει το οικείο σχολικό ή πανεπιστημιακό περιβάλλον. Στην εικόνα 12.1 απεικονίζεται μια εικονική αναπαράσταση του Πανεπιστημίου του Εδιμβούργου στο OpenSimulator.

Η τρίτη κατηγορία είναι η χρήση του εικονικού κόσμου ως βιοματικού περιβάλλοντος στο οποίο ο σπουδαστής μαθαίνει όχι μόνο μέσα από την ανάγνωση ή ανταλλαγή πληροφοριών αλλά και μέσα από την ίδια την αλληλεπίδρασή του με τα αντικείμενα του κόσμου. Με βάση το μοντέλο εμπειρικής μάθησης του Kolb (1984), οι χρήστες μπορούν να επενεργήσουν πάνω στα περιεχόμενα του κόσμου μαθαίνοντας μέσω των ενεργειών τους, να παρατηρήσουν τα αποτελέσματα, να ελέγξουν τις υποθέσεις τους και να αναστοχαστούν πάνω στο υπό μελέτη φαινόμενο. Οι εφαρμογές που ακολουθούν αυτήν την κατηγορία στηρίζονται στην αξιοποίηση των διαδραστικών στοιχείων του μέσου και εμπλέκουν τους χρήστες σε σενάρια που περιλαμβάνουν την κατασκευή, την τροποποίηση ή τον χειρισμό αντικειμένων ή χώρων με στόχο την επαύξηση των γνώσεων και ικανοτήτων των συμμετεχόντων. Για παράδειγμα, αντικείμενα εκπαίδευσης όπως η σχεδίαση και η τρισδιάστατη μοντελοποίηση μπορούν να αξιοποιήσουν τα ενσωματωμένα εργαλεία δημιουργίας και διαμόρφωσης περιεχομένου σε εικονικούς κόσμους όπως το Second Life και να παρακινήσουν τους φοιτητές να κατασκευάσουν περιεχόμενο, να το διαμοιραστούν σε ένα κοινό περιβάλλον και να αξιολογήσουν τις δουλειές των συμφοιτητών τους μέσω εξερεύνησης και περιήγησης. Σε άλλες περιπτώσεις εφαρμογών η εμπειρική μάθηση υποστηρίζεται από ειδικά κατασκευασμένα εργαλεία με τη μορφή λειτουργικών τρισδιάστατων αντικειμένων ή δισδιάστατων στοιχείων διεπαφής, τα οποία φέρνουν τους χρήστες σε βιοματική επαφή με το αντικείμενο μελέτης.



Εικόνα 6.2 Το περιβάλλον της εκπαιδευτικής εφαρμογής River City.

Μια επιτυχημένη εφαρμογή των εικονικών κόσμων στην εκπαίδευση η οποία συνδυάζει το βιοματικό περιβάλλον με την επικοινωνία και τη συνεργασία είναι το River City (Ketelhut, 2007). Η εφαρμογή αυτή σχεδιάστηκε και αναπτύχθηκε από το Πανεπιστήμιο του Harvard βασισμένη στην πλατφόρμα Active Worlds με στόχο την ενεργό εμπλοκή ομάδων μαθητών γυμνασίου στη διεξαγωγή επιστημονικής έρευνας και στη μάθηση μέσω αυτής. Παρουσιάζει μια ιστορική πόλη στα τέλη του 19ου αιώνα στην οποία έχει ξεσπάσει μια μεγάλη επιδημία (εικ. 6.2). Οι μαθητές συμμετέχουν και οι ίδιοι στην ιστορία, εξερευνούν την πόλη μέσω των ενσαρκώσεών τους και καλούνται να ανακαλύψουν τα αίτια διάδοσης της ασθένειας και να εμποδίσουν την περαιτέρω εξάπλωσή της. Στην προσπάθειά τους αυτή θα πρέπει να περιηγηθούν, να συνομιλήσουν με τους κατοίκους της και να διεξάγουν εικονικά πειράματα με στόχο να ελέγξουν τις υποθέσεις τους σχετικά με τα αίτια της επιδημίας. Στο τέλος θα πρέπει να συντάξουν ένα γράμμα προς τον δήμαρχο της πόλης στο οποίο να παρέχουν συμβουλές για την αντιμετώπιση του προβλήματος.

Το River City χρησιμοποιήθηκε από πολύ μεγάλο αριθμό σχολείων και μαθητών την τελευταία δεκαετία και έχουν γίνει πολλαπλές μελέτες σχετικά με τις επιπτώσεις της χρήσης του. Τα αποτελέσματα των μελετών είναι ιδιαίτερα θετικά, και φαίνεται να παρέχει αυξημένα κίνητρα συμμετοχής, ιδιαίτερα σε μαθητές χαμηλών επιδόσεων, αλλά και να οδηγεί στην εξερεύνηση περισσότερων εναλλακτικών λύσεων σε σχέση με

αντίστοιχες εργασίες που διεξήχθησαν χωρίς τη χρήση εικονικών κόσμων. Επιπλέον, φαίνεται να ενισχύει σημαντικά και την αυτο-αποτελεσματικότητα των μαθητών, δηλαδή την υποκειμενική αντίληψη που σχηματίζουν σχετικά με την ικανότητά τους να επιλύουν αυτού του ίδιους τα προβλήματα

Αν και θεωρητικά στα περισσότερα αντικείμενα εκπαίδευσης θα μπορούσαν να αξιοποιηθούν οι εικονικοί κόσμοι για την κατασκευή κατάλληλων εκπαιδευτικών σεναρίων και δραστηριοτήτων, πρακτικά κάποια αντικείμενα μπορούν να υποστηριχθούν πιο εύκολα σε σχέση με άλλα. Αυτό φαίνεται να σχετίζεται σε μεγάλο βαθμό και με τον απαιτούμενο χρόνο και τις τεχνικές γνώσεις που χρειάζονται για την προετοιμασία ενός περιβάλλοντος κατάλληλου για τις επιθυμητές δραστηριότητες. Πράγματι, ανάλογα με τις εκπαιδευτικές ανάγκες της περιοχής, μπορεί να απαιτείται εκτεταμένη κατασκευή νέου περιεχομένου, όπως για παράδειγμα χώροι, κτίρια, αντικείμενα, ρούχα κ.λπ., ή και σχεδίαση και ανάπτυξη εξειδικευμένων διαδραστικών εργαλείων και διεπαφών. Σε δραστηριότητες οι οποίες βασίζονται περισσότερο στην επικοινωνία μεταξύ των χρηστών ή και στην παραγωγή περιεχομένου από τους ίδιους, ο απαιτούμενος χρόνος προετοιμασίας του περιβάλλοντος μπορεί να είναι σημαντικά μικρότερος. Αντίθετα, οπτικοποιήσεις, προσομοιώσεις και παιχνίδια ρόλων στηρίζονται περισσότερο σε προκατασκευασμένο περιεχόμενο και εξειδικευμένες διαδράσεις.

Τα αντικείμενα εκπαίδευσης που φαίνεται να αξιοποιούν συχνότερα τους εικονικούς κόσμους για την υποστήριξη μέρους των εκπαιδευτικών δραστηριοτήτων είναι:

- *Οι τέχνες και τα μέσα*: οι εικονικοί κόσμοι αξιοποιούνται κυρίως για τρισδιάστατη μοντελοποίηση.
- *Η υγεία και το περιβάλλον*: εκπονούνται σεναρία για την ευαισθητοποίηση μαθητών ή φοιτητών σε θέματα όπως η διατροφή, η μόλυνση κ.λπ.
- *Η εκπαίδευση*: δίνεται έμφαση στη μελέτη των δυνατοτήτων του μέσου ως διδακτικό εργαλείο.
- *Το εμπόριο*: μελετώνται οι δυνατότητες αξιοποίησης των τρισδιάστατων κόσμων στο ηλεκτρονικό εμπόριο.
- *Η πληροφορική*: περιλαμβάνεται η τεχνολογία των εικονικών κόσμων και η εκμάθηση προγραμματισμού σε αυτούς.
- *Η γλώσσα*: βοηθά στην εκμάθηση ξένων γλωσσών.
- *Η βιβλιοθηκονομία*: αφορά τη μελέτη και συζητήσεις γύρω από διεπιστημονικά θέματα.

2.4 Καλές πρακτικές

Μια καλή σχεδιαστική πρακτική για την υποστήριξη εκπαιδευτικών δραστηριοτήτων σε εικονικά περιβάλλοντα φαίνεται να είναι η ενσωμάτωση στοιχείων και χαρακτηριστικών που κινητοποιούν τους χρήστες και διατηρούν ζωντανό το ενδιαφέρον τους. Οι εικονικοί κόσμοι μπορούν σχετικά εύκολα να εμπλουτιστούν με παιγνιώδη χαρακτηριστικά στη σχεδίαση του περιβάλλοντος, στη διεπαφή και στις δράσεις των χρηστών, ώστε να κάνουν την εμπειρία περισσότερο διασκεδαστική. Άλλωστε σήμερα είναι αρκετά διαδεδομένη η προσέγγιση της *παιγνιοποίησης (gamification)* στη μάθηση, καθώς και τα «σοβαρά παιχνίδια» (*serious games*), και οι εικονικοί κόσμοι έχουν από τη φύση τους αρκετά κοινά στοιχεία με τα σύγχρονα περιβάλλοντα παιχνιδιού, τα οποία θα μπορούσαν να γίνουν αντικείμενο εκμετάλλευσης προς αυτήν την κατεύθυνση. Τα εκπαιδευτικά περιβάλλοντα μπορούν να ενσωματώσουν ελκυστικές γραφικές απεικονίσεις του περιβάλλοντος και των ενσαρκώσεων, πλούσια διαδραστικά στοιχεία, μεγάλους και ενδιαφέροντες χώρους προς εξερεύνηση κ.ά. Επιπλέον, μπορούν να εισαχθούν στις μαθησιακές δραστηριότητες ενδιαφέροντα προβλήματα και προκλήσεις, συνθήκες συνεργασίας ή και ανταγωνισμού και διάφορες μορφές ανταμοιβής (όπως σκορ, «ξεκλειδωμα» νέων δυνατοτήτων κ.ά.) για την εκπλήρωση των στόχων. Όλα τα παραπάνω στοιχεία μπορούν να συμβάλλουν θετικά στην αύξηση της ελκυστικότητας του μέσου, αρκεί να συνυπάρχουν σε καλή ισορροπία με τα μαθησιακά στοιχεία του περιβάλλοντος.

Ένα δεύτερο στοιχείο που φαίνεται να είναι σημαντικός παράγοντας επιτυχίας σε πολυχρηστικά εκπαιδευτικά περιβάλλοντα είναι η καλή υποστήριξη συνεργατικών δραστηριοτήτων εντός του περιβάλλοντος. Παρά το γεγονός ότι οι εικονικοί κόσμοι εμπεριέχουν εγγενείς δυνατότητες επικοινωνίας και συνεργασίας, οι ανάγκες των μαθησιακών δραστηριοτήτων πολλές φορές απαιτούν καλύτερα οργανωμένα περιβάλλοντα ενισχυμένα με κατάλληλα εργαλεία. Σε δραστηριότητες στις οποίες απομακρυσμένοι συμμετέχοντες θα πρέπει να μελετήσουν και να λύσουν από κοινού κάποιο πρόβλημα, σε κοινότητες μάθησης στις οποίες τα πιο έμπειρα μέλη προσπαθούν να μεταλαμπαδεύσουν τη γνώση στους αρχάριους αλλά και σε εικονικές τάξεις που γίνεται διδασκαλία εξ αποστάσεως, θα πρέπει να υπάρχει η κατάλληλη οργάνωση για τη διεξαγωγή των συνεργατικών διεργασιών. Περιβάλλοντα σαν τα παραπάνω έχουν ανάγκη από εργαλεία για ανταλλαγή αρχείων, κοινοποίηση

εγγράφων και σημειώσεων, από κοινού επεξεργασία κειμένου, προβολή παρουσιάσεων, ασύγχρονες συζητήσεις, διαχείριση ρόλων κ.ά.

Ένα άλλο χαρακτηριστικό το οποίο είναι επίσης σημαντικό να υπάρχει σε εκπαιδευτικούς εικονικούς κόσμους είναι οι πλούσιες αλληλεπιδράσεις και η δυνατότητα πειραματισμού εντός του περιβάλλοντος. Όπως αναφέραμε και στο Κεφάλαιο 5, η αξιοποίηση των εικονικών κόσμων ως μέσου για την ανάπτυξη κάποιας εφαρμογής έχει νόημα κυρίως όταν το αποτέλεσμα έχει ξεκάθαρο πλεονέκτημα σε σχέση με τα εναλλακτικά μέσα. Στην περίπτωση των εκπαιδευτικών περιβαλλόντων δεν έχει ιδιαίτερο ενδιαφέρον το να αξιοποιηθεί το τρισδιάστατο περιβάλλον για να μεταφερθούν εντός αυτού δραστηριότητες που θα μπορούσαν να γίνουν και σε απλές ιστοσελίδες, όπως για παράδειγμα η επίλυση ενός κουίζ. Τα σημεία στα οποία φανερά πλεονεκτούν οι εικονικοί κόσμοι είναι οι πολλαπλές δυνατότητες αλληλεπίδρασης και δημιουργίας στο περιβάλλον. Οι χρήστες εμπυθίζονται στον τρισδιάστατο κόσμο, μπορούν να αναλάβουν ρόλους, να εξερευνήσουν τον χώρο, να παρακολουθήσουν ή να συμμετάσχουν σε προσομοιώσεις, να κατασκευάσουν νέο περιεχόμενο, να λάβουν μέρος σε ομαδικές δραστηριότητες και παιχνίδια κ.ά. Είναι χρήσιμο να αξιοποιούνται δυνατότητες σαν τις παραπάνω για την εμπλοκή των χρηστών σε βιωματικές μαθησιακές δραστηριότητες οι οποίες θα έχουν σημαντική προστιθέμενη αξία σε σχέση με άλλες λύσεις.

Τέλος, μια ακόμη διάσταση στην οποία θα πρέπει να δοθεί ιδιαίτερη έμφαση κατά τη σχεδίαση εκπαιδευτικών περιβαλλόντων είναι η συλλογή δεδομένων και η αξιολόγηση. Οι εικονικοί κόσμοι είναι ένας χώρος στον οποίο παράγονται πολύ πιο πλούσια δεδομένα σε σχέση με άλλα μέσα όπως τα παραθυρικά προγράμματα ή οι ιστοσελίδες. Οι χρήστες κατά τη διάρκεια της σύνδεσής τους πλοηγούνται, παρατηρούν, αλληλεπιδρούν και επικοινωνούν. Είναι χρήσιμο να υπάρχουν μηχανισμοί που να καταγράφουν κάποια από αυτά τα στοιχεία, ώστε να μπορούν στη συνέχεια να αξιοποιηθούν για την ερμηνεία των δραστηριοτήτων των χρηστών. Τα δεδομένα αυτά είναι ιδιαίτερα χρήσιμα στους εκπαιδευτικούς, για να έχουν μια όσο το δυνατόν πιο πλούσια εικόνα σχετικά με την εμπλοκή των χρηστών στο μαθησιακό περιβάλλον.

3 Πολιτισμός

Ο χώρος του πολιτισμού έχει εκμεταλλευτεί σε μεγάλο βαθμό τις ψηφιακές τεχνολογίες τις τελευταίες δεκαετίες με στόχο τον εμπλουτισμό της εμπειρίας των επισκεπτών και την περαιτέρω διάδοση της πολιτιστικής κληρονομιάς. Μουσεία, αρχαιολογικοί χώροι και εκθέσεις παρέχουν πλούσιο οπτικό και πληροφοριακό υλικό μέσω διαδικτύου, ενσωματώνουν αλληλεπιδραστικές εφαρμογές εντός του χώρου, διαθέτουν στους επισκέπτες συσκευές προσωποποιημένης ξενάγησης, συμπληρώνουν την επίσκεψη μέσω ειδικών διαδραστικών εφαρμογών για κινητά τηλέφωνα και πολλά άλλα. Η τεχνολογία έρχεται να βοηθήσει στην καλύτερη προβολή των εκθεμάτων σε απομακρυσμένους επισκέπτες, όπως είναι για παράδειγμα οι επισκέπτες μιας ιστοσελίδας, ώστε να τους παρακινήσει να επισκεφθούν στο μέλλον τον αντίστοιχο φυσικό χώρο, ή αν τον έχουν ήδη επισκεφτεί να πάρουν περισσότερες πληροφορίες για όσα είδαν. Επιπλέον μπορεί να επαυξήσει και την ίδια την εμπειρία της επίσκεψης με πληροφορίες και δραστηριότητες εστιασμένες στα ενδιαφέροντα και στις ανάγκες των διάφορων κατηγοριών επισκεπτών.

Οι εικονικοί κόσμοι δεν θα μπορούσαν βεβαίως να λείπουν από τις τεχνολογίες προβολής και διάδοσης των πολιτιστικών προϊόντων. Οι πρώτες σχετικές προσπάθειες βασίστηκαν σε μονοχρηστικά περιβάλλοντα εικονικής πραγματικότητας και στόχος ήταν η ρεαλιστική τρισδιάστατη αναπαράσταση των εκθεμάτων (Tsichritzis & Gibbs, 1991). Εκείνη την εποχή διατυπώθηκε για πρώτη φορά και ο όρος «εικονικό μουσείο» αναφερόμενος σε έναν τρισδιάστατο χώρο παρουσίασης εκθεμάτων στον οποίο οι χρήστες μπορούν να περιηγηθούν ελεύθερα αποκτώντας μια εμπειρία παραπλήσια με αυτήν της επίσκεψης σε ένα κανονικό μουσείο. Τα επόμενα χρόνια με την ευρεία διάδοση των τεχνολογιών τρισδιάστατων γραφικών και την ανάδυση των εικονικών περιβαλλόντων επιφάνειας εργασίας τα εικονικά μουσεία επεκτάθηκαν με περαιτέρω χαρακτηριστικά. Διαδραστικά εκθέματα, εικονικοί ξεναγοί και προσωποποιημένες περιηγήσεις είναι μερικά από αυτά (Sylaiou κ.ά., 2009). Πράγματι, γρήγορα άρχισε να γίνεται σαφές ότι τα μουσεία στα εικονικά περιβάλλοντα μπορούν να περιλαμβάνουν πολλαπλές αναπαραστάσεις και να υποστηρίζουν νέες διαδράσεις πέρα από την απλή παρατήρηση. Αυτά τα νέα χαρακτηριστικά έκαναν την εμπειρία πιο διασκεδαστική και έδωσαν ώθηση για την ανάπτυξη μεγάλου αριθμού σχετικών εφαρμογών στη μορφή εικονικών περιβαλλόντων ή παιχνιδιών, εμπλουτισμένων με πλούσια διαδραστικά στοιχεία. Η ευρεία διάδοση των εικονικών κόσμων τα τελευταία χρόνια πρόσθεσε μια ακόμη σημαντική διάσταση στα εικονικά μουσεία και στις αναπαραστάσεις χώρων πολιτισμού: τη συμμετοχικότητα. Οι χρήστες δεν είναι πλέον απλοί επισκέπτες, αλλά μπορούν να αναλάβουν και ρόλο «συν-δημιουργού». Το εικονικό πολιτιστικό περιβάλλον μετατρέπεται μέσα από την

τεχνολογία των εικονικών κόσμων σε έναν ζωντανό, εξελισσόμενο χώρο που υποστηρίζεται από μια ενεργή κοινότητα χρηστών.

3.1 Χώροι πολιτισμού και εμπειρία επισκεπτών

Τα μουσεία, οι αρχαιολογικοί χώροι και γενικότερα οι χώροι προβολής πολιτισμικών προϊόντων έχουν ως βασικό στόχο τη συντήρηση και την διάδοση των τεχνουργημάτων που περιλαμβάνουν. Αποσκοπούν λοιπόν αφενός στην προστασία των χώρων και των αντικειμένων από φθορές και αφετέρου στη μέγιστη δυνατή ανάδειξή τους. Θέλουν να προσφέρουν στους επισκέπτες πλούσια εικόνα και πληροφορίες σχετικά με τα εκθέματα, τα κτίρια και τον περιβάλλοντα χώρο, αλλά στις περισσότερες περιπτώσεις δεν επιτρέπεται η κοντινή προσέγγιση και το άγγιγμα των αντικειμένων. Ο επισκέπτης περιορίζεται στην οπτική παρατήρηση του χώρου και των περιεχομένων του και οι επιπλέον πληροφορίες έρχονται μέσα από συμπληρωματικά κείμενα ή και από αφηγήσεις και ιστορικά στοιχεία που παρέχονται από ξεναγούς. Ενώ λοιπόν οι χρήστες βρίσκονται στον φυσικό χώρο και έχουν μπροστά τους τα αντικείμενα του ενδιαφέροντός τους, το απαραίτητο συνοδευτικό πλαίσιο κατασκευάζεται στη φαντασία τους από τις λεκτικές περιγραφές.

Οι επισκέπτες των φυσικών χώρων πολιτισμού δεν έχουν όλοι τα ίδια ενδιαφέροντα και τις ίδιες προσδοκίες. Ορισμένοι μπορεί να ενδιαφέρονται για ένα υποσύνολο των εκθεμάτων, π.χ. ένας φοιτητής που μελετά μια συγκεκριμένη ιστορική περίοδο, άλλοι μπορεί να έχουν ευρύτερα ενδιαφέροντα και άλλοι απλώς να περιηγούνται μέχρι να εντοπίσουν κάτι που θα τραβήξει την προσοχή τους. Μελέτες σχετικά με τους στόχους των επισκεπτών σε εκθέσεις και μουσεία έδειξαν ότι στους βασικούς στόχους περιλαμβάνονται: η μάθηση και η ανακάλυψη, η παθητική ευχαρίστηση, η ψυχαγωγία, η κοινωνική αλληλεπίδραση και η προσωπική καταξίωση. Οι προτεραιότητες φαίνεται να αλλάζουν ανάλογα με το αντικείμενο της συλλογής. Σε σχέση με τη συμπεριφορά των επισκεπτών στον φυσικό χώρο ενός μουσείου αναγνωρίστηκαν τέσσερις διαφορετικοί τύποι επισκεπτών (Veron & Lvasseur, 1989) οι οποίοι περιγράφονται βάσει αντίστοιχων ζωικών συμπεριφορών:

- ο επισκέπτης «μυρμήγκι», ο οποίος αφιερώνει πολύ χρόνο για να μελετήσει όλα τα εκθέματα κινούμενος κυρίως κοντά στους τοίχους και στα εκθέματα,
- ο επισκέπτης «ψάρι», ο οποίος κινείται κατά βάση στους διαδρόμους κάνοντας μόνο μερικές στάσεις και βλέπει τα περισσότερα εκθέματα, αλλά για λίγο,
- ο επισκέπτης «ακρίδα», που βλέπει μόνο τα εκθέματα που τον ενδιαφέρουν κινούμενος στους διαδρόμους και κάνοντας μεγάλη στάση σε αυτά και
- ο επισκέπτης «πεταλούδα», που αλλάζει συχνά τη διεύθυνση της επίσκεψης και βλέπει σχεδόν όλα τα εκθέματα αλλά διαφέρει ο χρόνος που θα αφιερώσει σε καθένα από αυτά.

Φαίνεται λοιπόν να υπάρχει ένα μεγάλο εύρος αναγκών και συμπεριφορών των επισκεπτών ενός χώρου πολιτισμού. Ορισμένοι εστιάζουν στη διδακτική πτυχή της επίσκεψης, μπορεί να ενδιαφέρονται για συγκεκριμένα στοιχεία του χώρου και για τα στοιχεία αυτά να θέλουν να συλλέξουν όλες τις σχετικές πληροφορίες. Άλλοι μπορεί να ενδιαφέρονται περισσότερο για την αισθητική της εμπειρίας και να αντιλαμβάνονται τον χώρο ως ένα μέσο ψυχαγωγίας, ενώ μια τρίτη κατηγορία επισκεπτών ενδιαφέρεται και για την κοινωνική πτυχή της επίσκεψης και μπορεί να έχει ως στόχο να συζητήσει για τα εκθέματα, να εντοπίσει άλλους επισκέπτες με παραπλήσια ενδιαφέροντα κ.λπ.

3.2 Προσφερόμενες δυνατότητες

Οι εικονικοί κόσμοι έχουν τη δυνατότητα να βοηθήσουν προς την κατεύθυνση της λεπτομερέστερης παρουσίασης των εκθεμάτων και της ένταξής τους σε κατάλληλο περιβάλλον-πλαίσιο. Μπορούν να προσφέρουν ποιοτικές οπτικές αναπαραστάσεις και να δώσουν στον θεατή τη δυνατότητα προσεκτικής επισκόπησης των εκθεμάτων και επιλεκτικής εστίασης. Επιπλέον, τα εκθέματα μπορούν να είναι κινούμενα ή και διαδραστικά έτσι ώστε να μπορεί ο χρήστης να παρακολουθήσει την ιστορική εξέλιξή τους, να τα χειριστεί, να τα φορέσει, να τα αποσυναρμολογήσει κ.λπ. Τέλος, μπορεί να παρουσιαστεί το ευρύτερο ιστορικό πλαίσιο με το οποίο έχουν συνδεθεί τα προβαλλόμενα πολιτιστικά προϊόντα μέσα από διαδραστικά περιβάλλοντα προσομοίωσης με εικονικούς χαρακτήρες. Αν και στερείται ο επισκέπτης την άμεση επαφή με το φυσικό

αντικείμενο, μπορεί εντούτοις να αποκομίσει πλούσιες γνώσεις σχετικά με αυτό συμμετέχοντας σε διαδραστικά περιβάλλοντα εξιστόρησης ή και προσομοίωσης που το αναδεικνύουν.

Οι διαφορετικές ανάγκες των επισκεπτών μπορούν να καλυφθούν σε μεγάλο βαθμό στους εικονικούς κόσμους μέσα από πολλαπλές αναπαραστάσεις και διαδικασίες προσαρμογής και προσωποποίησης. Στους επισκέπτες που ενδιαφέρονται να αποκτήσουν πλούσιες πληροφορίες σχετικά με τους χώρους και τα εκθέματα θα παρουσιάζονται επιπλέον σελίδες και πολυμεσικά στοιχεία με πληροφοριακό περιεχόμενο. Επιπλέον, η ξενάγησή τους στον χώρο μπορεί να κατευθύνεται βάσει του ιστορικού περιήγησης και των ενδιαφερόντων τους· το περιβάλλον μπορεί να υποδεικνύει διαδρομές που να οδηγούν σε εκθέματα παραπλήσια με αυτά που έχουν ήδη παρατηρήσει και που πιθανώς να τους ενδιαφέρουν. Οι ψυχαγωγικές ανάγκες των χρηστών μπορούν να καλυφθούν μέσα από εξιστορήσεις, παιχνίδια ρόλων ή και άλλες δραστηριότητες (π.χ. προκλήσεις εξερεύνησης) που θα εμπεριέχονται στο περιβάλλον. Τέλος, με δεδομένο ότι οι εικονικοί κόσμοι υποστηρίζουν πολλαπλές μορφές κοινωνικής αλληλεπίδρασης και συνεργασίας, οι εικονικοί χώροι πολιτισμού έχουν τη δυνατότητα να υποστηρίξουν την επικοινωνία μεταξύ των επισκεπτών, την κοινωνική πλοήγηση (π.χ. σχόλια, επισημειώσεις, βαθμολογίες), και κυρίως την ανάδυση εικονικών κοινοτήτων συνεργασίας.



Εικόνα 6.3 Το μουσείο διαστημικών πτήσεων στο Second Life.

3.3 Επισκόπηση εφαρμογών

Η ευρεία χρήση του Second Life από πολύ μεγάλο αριθμό επισκεπτών οδήγησε, μεταξύ άλλων, σε πολλαπλές προσπάθειες αξιοποίησής του στον χώρο των εικονικών μουσείων. Κατασκευάστηκε και υποστηρίχθηκε ενεργά ένας μεγάλος αριθμός από χώρους που έχουν ως στόχο να φέρουν τους επισκέπτες σε επαφή με εικονικές αναπαραστάσεις τεχνουργημάτων που σχετίζονται με τον πολιτισμό και να τους παρέχουν πλούσιες πληροφορίες σχετικά με αυτά. Οι περισσότερες από αυτές τις προσπάθειες συνεχίζονται μέχρι σήμερα, παρά τη μειωμένη πλέον δημοτικότητα του Second Life, ενώ αναδύονται και νέες σε περιβάλλοντα όπως το OpenSimulator (Sequeira & Morgado, 2013). Σε κάποιες περιπτώσεις η έμφαση δίνεται στα ίδια τα εκθέματα, τα οποία παρουσιάζονται σε απλοποιημένους χώρους ώστε ο χρήστης να επικεντρώσει την προσοχή του σε αυτά, ενώ σε άλλες ο περιβάλλον χώρος αποτελεί επίσης αντικείμενο ενδιαφέροντος και περιλαμβάνεται στην εικονική αναπαράσταση. Κάποια από τα εικονικά μουσεία έχουν μόνιμα, διαρκή εκθέματα, ενώ άλλα ανανεώνουν το περιεχόμενό τους, πολλές φορές και με τη συμμετοχή ή τις υποδείξεις των επισκεπτών (Urban κ.ά., 2007). Το είδος των πληροφοριών που συνοδεύουν τα αντικείμενα ποικίλλει από απλό κείμενο, συνήθως στη μορφή ενός σημειώματος που εμφανίζεται όταν ο χρήστης επιλέγει το αντικείμενο, μέχρι πολυμεσικά στοιχεία όπως εικόνες, βίντεο και σύνδεσμοι σε ιστοσελίδες.

Ένα ενδεικτικό παράδειγμα εικονικού μουσείου στο Second Life με πλούσια διαδραστικά στοιχεία είναι το διεθνές μουσείο διαστημικών πτήσεων (εικ. 6.3). Πρόκειται για έναν μεγάλο χώρο που κατασκευάστηκε από ερασιτέχνες και περιέχει λεπτομερή αντίγραφα ιστορικών διαστημικών οχημάτων και κατασκευών, που παράλληλα προσφέρει πολλαπλές δυνατότητες κοινωνικής αλληλεπίδρασης. Ο επισκέπτης

μπορεί να παρατηρήσει ένα τρισδιάστατο μοντέλο της σεληνακάτου, να δει πολλές φωτογραφίες από διαστημικές αποστολές και να εξερευνήσει έναν μεγάλο κεντρικό χώρο στον οποίο εκτίθενται πύραυλοι και άλλα διαστημικά οχήματα, αποκτώντας πλούσια εικόνα των σχετικών τεχνολογιών και της χρονολογικής εξέλιξής τους. Ο χώρος δεν περιορίζεται όμως μόνο στην προβολή των εκθεμάτων. Υπάρχουν αρκετές επιπρόσθετες κινούμενες ή και διαδραστικές εγκαταστάσεις, όπως μια προσομοίωση του ηλιακού συστήματος, ένα πλανητάριο, η δυνατότητα εικονικού ταξιδιού με πύραυλο στο διάστημα αλλά και αυτόματη ξενάγηση στον χώρο με πλοήγηση και ηχητική παρουσίαση. Στο μουσείο περιλαμβάνεται ακόμα ένας μεγάλος χώρος διαλέξεων και παρουσιάσεων, μικροί συνεδριακοί χώροι καθώς και άλλοι χώροι συνεύρεσης και συζήτησης μεταξύ των επισκεπτών. Οι διαχειριστές διοργανώνουν συχνά συζητήσεις και δρώμενα με στόχο τη διατήρηση του ενδιαφέροντος των επισκεπτών και την ανάδυση εικονικών κοινοτήτων.

Ένα δεύτερο παράδειγμα είναι το Second Louvre, ένα εικονικό μουσείο που περιλαμβάνει αναπαραστάσεις των εκθεμάτων του μουσείου του Λούβρου. Στην περίπτωση αυτή η έμφαση δεν δόθηκε μόνο στην παρουσίαση και ανάδειξη των εκθεμάτων αλλά και στην πιστή αναπαραστάση της αρχιτεκτονικής και του περιβάλλοντος. Κατασκευάστηκε μια ολόκληρη πτέρυγα του μουσείου εντός της οποίας τοποθετήθηκαν τα εκθέματα και αρκετά διακοσμητικά στοιχεία, όπως κιονοστοιχίες, σιντριβάνια κ.λπ. Στόχος ήταν να μεταφερθεί σε κάποιον βαθμό η αίσθηση μεγαλοπρέπειας που αποπνέεται στον φυσικό χώρο του Λούβρου και να τοποθετηθούν τα εκθέματα σε ένα περιβάλλον αντίστοιχης αισθητικής. Βεβαίως, η πιστή αναπαραγωγή ενός φυσικού χώρου σε έναν εικονικό κόσμο σε κλίμακα 1:1 προκαλεί δυσκολίες στην περιήγηση και στην εξερεύνηση. Ο χρήστης έχοντας οπτική τρίτου προσώπου αισθάνεται αρκετά «περιορισμένος» σε έναν κλειστό χώρο με μικρές αποστάσεις από τους τοίχους τόσο ως προς την οπτική όσο και ως προς τα περιθώρια μετακίνησης. Κατά συνέπεια, ο χώρος του εικονικού μουσείου του Λούβρου, παρά την εντυπωσιακή απεικόνισή του, δυσκολεύει τις ομαδικές επισκέψεις. Τα συνοδευτικά μέσα και οι συμπληρωματικές πληροφορίες σχετικά με τα εκθέματα είναι επίσης αρκετά περιορισμένα, καθώς η έμφαση δίνεται περισσότερο στην αισθητική και λιγότερο στην επαύξηση του χώρου με χρήσιμες πληροφορίες.

Το ενδιαφέρον στοιχείο με τα εικονικά μουσεία είναι η αξιοποίηση του μέσου για την υποστήριξη πολλαπλών πλούσιων δραστηριοτήτων πέρα από την απλή επισκόπηση συλλογών. Για παράδειγμα, σε ορισμένους χώρους προσφέρεται η δυνατότητα να αποκτήσει ο χρήστης την εμπειρία ενός φαινομένου μέσω της ενσάρκωσής του, όπως για παράδειγμα το ταξίδι με διαστημόπλοιο, η εκδήλωση ενός τσουνάμι κ.ά. Σε άλλες περιπτώσεις παρουσιάζονται πλούσιες προσομοιώσεις λειτουργίας μηχανισμών ή εκδήλωσης φαινομένων, όπου ο χρήστης μπορεί να παρατηρήσει ζωντανά τη χρονική εξέλιξη του αντικείμενου προσομοίωσης. Οι προσομοιώσεις αυτές μπορούν να ενισχυθούν και με διαδραστικά χαρακτηριστικά, με αποτέλεσμα να είναι σε θέση ο χρήστης να ελέγξει τις παραμέτρους ή και τον ρυθμό εξέλιξης του φαινομένου. Για να διατηρήσουν το ενδιαφέρον των επισκεπτών, πολλές φορές οι διαχειριστές μουσείων και εκθεσιακών χώρων διοργανώνουν διαλέξεις, συζητήσεις και άλλου είδους κοινωνικές δραστηριότητες, όπως στο παράδειγμα του μουσείου διαστημικών πτήσεων που παρουσιάσαμε πιο πάνω. Τέλος, σε κάποιες περιπτώσεις ενισχύεται ο εκπαιδευτικός χαρακτήρας της επίσκεψης μέσω δραστηριοτήτων που έχουν ως στόχο να παρακινήσουν τους επισκέπτες, κυρίως τα παιδιά, να ανακαλύψουν περισσότερες πληροφορίες σχετικά με τα εκθέματα. Παιχνίδια εξερεύνησης (π.χ. κυνήγι θησαυρού), παιχνίδια ρόλων και διαδραστικές εξιστορήσεις είναι κάποιες ενδεικτικές δραστηριότητες που παρακινούν στην ενεργό εμπλοκή των επισκεπτών με το αντικείμενο μάθησης.

Ένα ακόμη στοιχείο το οποίο άρχισε να κάνει την εμφάνισή του σε εικονικά μουσεία και χώρους πολιτισμού και επαυξάνει την εμπειρία των επισκεπτών είναι η προσωποποιημένη παρουσίαση του περιεχομένου. Όπως αναφέρθηκε παραπάνω, οι επισκέπτες των φυσικών χώρων πολιτισμού έχουν διαφορετικά ενδιαφέροντα και ανάγκες. Αυτή η διαφοροποίηση είναι πιθανότατα ακόμα πιο έντονη στα μουσεία εντός εικονικών κόσμων, όπου αρκετοί επισκέπτες μπορεί να βρέθηκαν τυχαία στη σχετική περιοχή ή να τα επισκέπτονται από απλή περιέργεια. Οι σύγχρονες ψηφιακές διεπαφές προσπαθούν σε μικρότερο ή μεγαλύτερο βαθμό να προσαρμοστούν στα προσωπικά μας ενδιαφέροντα, π.χ. ηλεκτρονικά καταστήματα τα οποία μας προτείνουν προϊόντα με βάση τις αναζητήσεις και τις αγορές που έχουμε ήδη πραγματοποιήσει. Η ίδια λογική μπορεί να μεταφερθεί σε κάποιο βαθμό και στην επίσκεψη σε έναν ψηφιακό χώρο πολιτισμού, όπου το σύστημα μπορεί να παρακολουθήσει τις ενέργειες του χρήστη, να βγάλει κάποια συμπεράσματα σχετικά με τις προτιμήσεις και τα ενδιαφέροντά του, δημιουργώντας ένα ψηφιακό προφίλ του χρήστη, και να προσαρμόσει την παρουσίαση σε σχέση με αυτά. Το περιβάλλον ReVEP είναι πολυχρηστικό εικονικό περιβάλλον που υποστηρίζει αυτού του είδους την προσωποποίηση σε ψηφιακές εκθέσεις: κατασκευάζει εξατομικευμένους εκθεσιακούς χώρους με βάση τα ενδιαφέροντα του κάθε επισκέπτη (Bonis κ.ά., 2009). Σε μια άλλη, πιο πρόσφατη προσέγγιση, ένας εικονικός ξεναγός αναλαμβάνει να παρουσιάσει τα εκθέματα στους επισκέπτες

ενός εκθεσιακού χώρου στο Second Life, λαμβάνοντας υπόψη τις προηγούμενες αλληλεπιδράσεις τους και προσαρμόζοντας αντίστοιχα τις πληροφορίες. Το σύστημα συνθέτει περιγραφές σε φυσική γλώσσα (στα ελληνικά και στα αγγλικά) συμπεριλαμβάνοντας μόνο τις πληροφορίες εκείνες που θεωρεί ότι είναι χρήσιμες στον επισκέπτη και αποφεύγοντας επαναλήψεις (Karakatsiotis κ.ά., 2008).

Πέρα από την αναπαράσταση υπαρκτών χώρων και εκθεμάτων, οι εικονικοί κόσμοι χρησιμοποιούνται και για την ανασύσταση κτιρίων και αντικειμένων του παρελθόντος. Το τρισδιάστατο περιβάλλον αξιοποιείται στην περίπτωση αυτή ως ένας χώρος δοκιμής και ελέγχου χώρων και μοντέλων που αναπαριστούν την αρχική μορφή αρχαιολογικών ευρημάτων. Υπάρχει μια μεγάλη κοινότητα «ερασιτεχνών αρχαιολόγων» στο Second Life, η οποία προσπαθεί να ανακατασκευάσει κτίρια και πόλεις της αρχαιότητας. Προτείνονται μοντέλα αντικειμένων, κτιρίων, ενδυμάτων κ.λπ., αποφασίζεται βάσει πλειοψηφίας ποια θα γίνουν αποδεκτά και μια επιτροπή ειδικών αξιολογεί την εγκυρότητα του αποτελέσματος, χωρίς όμως να βασίζονται σε επιστημονικές τεχνικές. Οι εικονικοί κόσμοι έχουν αξιοποιηθεί και από επαγγελματίες αρχαιολόγους στην προσπάθεια αναπαράστασης κτιρίων του παρελθόντος. Ειδικοί από διάφορα πεδία συγκεντρώνονται στον εικονικό χώρο, δοκιμάζουν διάφορες υποθέσεις, προτείνουν κατάλληλες αναπαραστάσεις, σχολιάζουν, συμπληρώνουν ή διορθώνουν, μέχρις ότου να καταλήξουν σε ένα κοινώς αποδεκτό και επιστημονικά έγκυρο αποτέλεσμα. Αν και οι εικονικοί κόσμοι δεν είναι το καταλληλότερο μέσο για την απεικόνιση μοντέλων υψηλής λεπτομέρειας, έχουν εντούτοις τη δυνατότητα να υποστηρίξουν συνεργατικές διεργασίες κατασκευής, επισκόπησης και λήψης αποφάσεων, κάτι που έχουν ανάγκη οι αρχαιολόγοι λόγω της διεπιστημονικής φύσης της δουλειάς τους. Σε κάποιες περιπτώσεις έχουν αξιοποιηθεί οι εικονικοί κόσμοι και για την εκπαίδευση νέων αρχαιολόγων, όπου επισκέφθηκαν έναν χώρο εικονικής ανασύστασης και κλήθηκαν να μελετήσουν το αποτέλεσμα και να τεκμηριώσουν τις αποφάσεις που ελήφθησαν.



Εικόνα 6.4 Η εφαρμογή *City of Uruk* στο *Second Life*.

Τέλος, μια ακόμα ενδιαφέρουσα και ανερχόμενη τάση είναι η χρήση εικονικών πρακτόρων για την αναβίωση πολιτισμών του παρελθόντος. Το πιο γνωστό παράδειγμα είναι η ανακατασκευή της αρχαίας πόλης Uruk στο Second Life (Bogdanovych κ.ά., 2012), η οποία χρησιμοποιήθηκε για τη διδακτική της ιστορίας. Στο περιβάλλον της αρχαίας πόλης κινούνται ελεύθερα «εικονικοί κάτοικοι», δηλαδή πράκτορες που αναπαριστούν χαρακτήρες της πόλης και έχουν αναλάβει χαρακτηριστικούς ρόλους της εποχής, π.χ. εργάτης, ιερέας κ.λπ. Κάθε χαρακτήρας είναι επιφορτισμένος με συγκεκριμένα καθήκοντα, ανάλογα με τον ρόλο του, και για την εκτέλεσή τους αλληλεπιδρά με αντικείμενα του κόσμου και με άλλους χαρακτήρες βάσει κάποιου περίπλοκου μοντέλου συμπεριφοράς. Οι επισκέπτες μπορούν να συνομιλήσουν με τους χαρακτήρες αυτούς μέσω φυσικής γλώσσας και να πάρουν περισσότερες πληροφορίες σχετικά με τους ρόλους και τα καθήκοντά τους. Επομένως, μπορεί να αποκομίσει κάποιος μια πλούσια εμπειρική εικόνα της καθημερινότητας και του τρόπου ζωής των κατοίκων της αρχαίας πόλης τόσο μέσα από την παρατήρηση των ενεργειών τους όσο και μέσω διαλόγου μαζί τους. Σε μια πιο πρόσφατη προσπάθεια (Lim κ.ά., 2013) προσομοιώνεται η καθημερινότητα σε ένα εμπορικό λιμάνι του 19ου αιώνα, όπου συνυπάρχουν χαρακτήρες πολλαπλών εθνικοτήτων. Στόχος είναι η ανάδειξη των διαφορετικών συμπεριφορών και των αλληλεπιδράσεων μεταξύ εθνικοτήτων.

3.4 Καλές πρακτικές

Όπως παρατηρούμε από την επισκόπηση που προηγήθηκε, οι εφαρμογές ανάδειξης και διάδοσης της πολιτιστικής κληρονομιάς στους εικονικούς κόσμους προσπαθούν να πάνε ένα βήμα πιο πέρα σε σχέση με την εμπειρία της φυσικής επίσκεψης στο μουσείο. Είναι άλλωστε σαφές ότι μια τρισδιάστατη αναπαράσταση, ιδιαίτερα σε περιβάλλοντα μικρής εμπύθισης όπως η οθόνη ενός υπολογιστή, απέχει πολύ από την φυσική επαφή με το αντικείμενο. Εντούτοις, οι εικονικοί κόσμοι προσφέρουν άλλου είδους δυνατότητες οι οποίες μπορούν να επαυξήσουν την εμπειρία, όπως η λεπτομερής παρατήρηση, η διάδραση, η πλαισίωση των αντικειμένων σε ρεαλιστικά περιβάλλοντα, η παρουσίαση εναλλακτικών αναπαραστάσεων, η προσομοίωση, κλπ. Φαίνεται συνεπώς σκόπιμη η ενσωμάτωση πρακτικών όπως:

- ο εμπλουτισμός του χώρου και των αντικειμένων με διαδραστικά στοιχεία που οδηγούν σε μεγαλύτερη εμπλοκή των χρηστών και κάνουν την εμπειρία περισσότερο βιωματική,
- η υποστήριξη κοινωνικών δραστηριοτήτων, όπως σχολιασμός, ανατροφοδότηση, συζητήσεις, ανάδυση ομάδων ενδιαφέροντος, κ.ά.,
- η παροχή προσωποποιημένων στοιχείων και πληροφοριών, και
- η χρήση συνθετικών χαρακτήρων για ξενάγηση, προσομοιώσεις και παιχνίδια ρόλων

4 Σχεδίαση

Η σχεδίαση ως πρακτική στηρίζεται σε μεγάλο βαθμό στην καλή οπτική αναπαράσταση και στη γρήγορη διαμόρφωση νέων αντικειμένων. Παράλληλα, τα τρισδιάστατα περιβάλλοντα πραγματικού χρόνου δίνουν έμφαση τόσο στην ποιοτική απεικόνιση αντικειμένων σε διάφορα επίπεδα ρεαλισμού όσο και στην αλληλεπίδραση με το περιεχόμενο του κόσμου. Ήταν λοιπόν αναμενόμενο να εξεταστεί η δυνατότητα αξιοποίησης της εικονικής πραγματικότητας στον χώρο της σχεδίασης από τα πρώιμα στάδια εξέλιξης του μέσου. Η κοινότητα των σχεδιαστών είδε με ενδιαφέρον τις δυνατότητες του νέου μέσου, παρόλο που αρχικά το κόστος ήταν απαγορευτικό και οι λειτουργίες περιορισμένες. Η τεχνολογία των τρισδιάστατων γραφικών έδωσε για πρώτη φορά στους σχεδιαστές τη δυνατότητα να αποτυπώσουν τις ιδέες τους πέρα από το χαρτί ή τις μακέτες με αρκετά ρεαλιστική μορφή των αντικειμένων και των χώρων και με πολυάριθμες δυνατότητες εύκολης και γρήγορης επεξεργασίας των σχεδίων. Δεν είναι τυχαίο το ότι σήμερα έχουν αναπτυχθεί πολλαπλά εξειδικευμένα εργαλεία σχεδίασης και μηχανικής με υπολογιστή (Computer Aided Design – CAD, Computer Aided Engineering – CAE) βασισμένα σε τρισδιάστατα γραφικά και ότι θεωρούνται απαραίτητα σε επαγγελματίες σχετικών ειδικοτήτων.

Οι εικονικοί κόσμοι, ως πιο σύγχρονο και ευρέως προσβάσιμο μέσο, κίνησαν ξανά την τελευταία δεκαετία το ενδιαφέρον των σχεδιαστών λόγω ορισμένων σημαντικών χαρακτηριστικών τους που δεν συμπεριλαμβάνονται στα επαγγελματικά εργαλεία σχεδίασης: της πολυχρηστικότητας, της εμπύθισης και της ευκολίας χρήσης. Η σχεδίαση ως διαδικασία προϋποθέτει τη συνεργασία πολλαπλών σχεδιαστών, πιθανότατα διαφορετικών ειδικοτήτων, πάνω στο αντικείμενο σχεδίασης. Η ταυτόχρονη παρουσία πολλαπλών χρηστών σε ένα διαμοιρασμένο περιβάλλον όπως είναι ένας εικονικός κόσμος μπορεί να υποστηρίξει σε μεγάλο βαθμό συνεργατικές δραστηριότητες αυτής της μορφής. Επιπλέον, οι σύγχρονες τάσεις στη σχεδίαση δίνουν έμφαση στη συμμετοχικότητα, δηλαδή στην εμπλοκή τελικών χρηστών στα διάφορα στάδια της σχεδίασης. Η εμπύθιση μέσω της ενσάρκωσης που προσφέρουν οι εικονικοί κόσμοι μπορεί να δώσει τη δυνατότητα στους σχεδιαστές, αλλά και στους τελικούς χρήστες, να εξερευνήσουν και να μελετήσουν το αντικείμενο της σχεδίασης, ώστε να προσφέρουν ουσιαστικότερη ανατροφοδότηση. Τέλος, η δυνατότητα κατασκευής περιεχομένου εντός των εικονικών κόσμων με εύκολα στην εκμάθηση εργαλεία μπορεί να βοηθήσει τους τελικούς χρήστες να συνδιαμορφώσουν μια λύση ή και να προτείνουν εναλλακτικές, έστω και σε ένα υψηλότερο επίπεδο αφαίρεσης.

Παρά το ενδιαφέρον της σχεδιαστικής κοινότητας, η χρήση των εικονικών κόσμων στη σχεδίαση εξακολουθεί σήμερα να βρίσκεται σε ερευνητικό/πειραματικό στάδιο, κυρίως λόγω της έλλειψης εξειδικευμένων εργαλείων λεπτομερειακής σχεδίασης, τα οποία είναι απαραίτητα για τους επαγγελματίες στα τελικά στάδια.

4.1 Η σχεδιαστική διαδικασία

Η σχεδίαση περιγράφεται ως διαδικασία μετασηματισμού ενός περιβάλλοντος μέσω ένσκηνης και βάσει περιορισμών λήψης αποφάσεων που απαιτεί εξερεύνηση δημιουργικότητα και μάθηση (Hybs & Gero, 1992). Ο ορισμός αυτός είναι βεβαίως πολύ γενικός και συμπεριλαμβάνει όλες τις σύγχρονες εκφάνσεις της σχεδίασης. Πράγματι, σήμερα εντοπίζονται πολλά διαφορετικά αντικείμενα σχεδίασης, όπως γραφιστική, αρχιτεκτονική, σχεδίαση προϊόντων, σχεδίαση λογισμικού, σχεδίαση υπηρεσιών κ.ά. Το κοινό χαρακτηριστικό, όπως αφήνει και ο αρχικός ορισμός να εννοηθεί, είναι ότι η έμφαση δεν δίνεται μόνο στο ίδιο το αντικείμενο της σχεδίασης αλλά και στις συνέπειες που έχει η εισαγωγή του στο περιβάλλον. Επομένως, βασικός στόχος οφείλει να είναι η κάλυψη κάποιων δεδομένων ανθρώπινων αναγκών ή γενικότερα η επίλυση ενός υπαρκτού προβλήματος. Και τελικά το αντικείμενο της σχεδίασης αξιολογείται από τον βαθμό στον οποίο καλύπτει τις αρχικές απαιτήσεις. Για παράδειγμα, στη σχεδίαση προϊόντων δεν αρκεί να πούμε ότι θέλουμε να σχεδιάσουμε μια «καλή» καρέκλα. Το επίθετο καλή είναι άλλωστε αρκετά ασαφές για να προσδιορίσει τους στόχους μας. Θα πρέπει να ξεκινήσουμε από το πρόβλημα που θέλουμε να λύσουμε. Θέλουμε να είναι αναπαυτική; Θέλουμε να ακολουθεί κάποια αισθητικά πρότυπα; Πώς πρόκειται να χρησιμοποιηθεί; Σε τι περιβάλλον θα τοποθετηθεί; Ποιες θα είναι οι αναμενόμενες συνθήκες χρήσης; Για λόγους όπως οι παραπάνω, η σχεδίαση τις τελευταίες δεκαετίες δίνει μεγάλη έμφαση σε έννοιες που σχετίζονται με το πλαίσιο χρήσης του αντικειμένου, όπως η εργονομία, η ευχρηστία (στην οποία αναφερθήκαμε εκτενώς στο Κεφ. 5), η εμπειρία του χρήστη κ.ά.

Υπάρχουν διάφορες θεωρίες και μεθοδολογίες σχετικά με τη σχεδίαση, και οι πιο σύγχρονες από αυτές τονίζουν την ανάγκη της επαναληπτικότητας και της διαρκούς αξιολόγησης και ανατροφοδότησης από τα πρώτα κιόλας στάδια. Αν παρά τις διαφοροποιήσεις προσπαθήσουμε να διακρίνουμε κάποια γενικά στάδια στη σχεδιαστική διαδικασία, θα διακρίνουμε τα εξής τέσσερα:

- Ανάλυση απαιτήσεων: μελετάται το πρόβλημα, καταγράφονται οι απαιτήσεις των τελικών χρηστών, εντοπίζονται περιορισμοί που σχετίζονται με το περιβάλλον πλαίσιο χρήσης, την τεχνολογία, το κόστος, τη σύνθεση της ομάδας ανάπτυξης κ.λπ. και τα παραπάνω μετασηματίζονται σε σχεδιαστικές προδιαγραφές.
- Προκαταρκτική σχεδίαση: οι ιδέες διαμορφώνονται σε αφηρημένες περιγραφές της μορφής και της λειτουργίας πιθανών σχεδιαστικών λύσεων και σε σενάρια χρήσης αυτών.
- Αναλυτική σχεδίαση: τα προκαταρκτικά σχέδια εξειδικεύονται, αποκτούν περισσότερο συμπαγή μορφή, εξελίσσονται στον απαιτούμενο βαθμό λεπτομέρειας ώστε να υπακούουν στους περιορισμούς που έχουν τεθεί.
- Αξιολόγηση: τα παραγόμενα αναλυτικά σχέδια κατασκευάζονται ως πρωτότυπα και δοκιμάζονται από ενδεικτικούς τελικούς χρήστες με στόχο την ανατροφοδότηση σχετικά με την ποιότητα του αποτελέσματος, λαμβάνονται αποφάσεις επανασχεδίασης, αν και όπου χρειάζεται.

Ένα χαρακτηριστικό που θεωρείται δεδομένο σήμερα στη σχεδίαση είναι η συνεργασία. Ένα σύγχρονο σχεδιαστικό έργο είναι συνήθως αρκετά περίπλοκο και είναι πρακτικά αδύνατο να αναλάβει ένας άνθρωπος εξολοκλήρου την περάτωσή του. Άλλωστε, στα πρώτα, δημιουργικά στάδια της σχεδίασης είναι σημαντική η διαδικασία ανταλλαγής ιδεών και απόψεων μεταξύ μιας ομάδας σχεδιαστών, και αυτό φαίνεται να ενισχύει τη δημιουργικότητα. Κατά συνέπεια, οι σχεδιαστικές διαδικασίες πραγματοποιούνται συνήθως από ομάδα σχεδιαστών, ενώ πολλές φορές χρειάζεται να συνεργαστούν επιπλέον και με επαγγελματίες άλλων ειδικοτήτων. Ορίζονται λοιπόν ξεκάθαροι ρόλοι και καθήκοντα κατά τη διάρκεια των διεργασιών και απαιτείται συχνή και καλή επικοινωνία μεταξύ των συνεργαζόμενων μερών. Είναι απαραίτητη η διαμοίραση των διάφορων εγγράφων και αναπαραστάσεων που παράγονται στα ενδιάμεσα στάδια, ενώ υπάρχουν και εργασίες που εκτελούνται ομαδικά στον ίδιο χώρο ή εξ αποστάσεως και θα πρέπει να υποστηρίζονται από εργαλεία σύγχρονης συνεργασίας.

Τα τελευταία χρόνια έχει αρχίσει και εξετάζεται η εμπλοκή των τελικών χρηστών στη διαδικασία της σχεδίασης, μια πρακτική που ονομάζεται και συμμετοχική σχεδίαση. Έναυσμα για την προσέγγιση αυτήν αποτελεί η διαπίστωση ότι η συμμετοχή ενδεικτικών χρηστών σε αξιολογήσεις συνεισφέρει μεν σημαντικά στην ανατροφοδότηση επί των σχεδιαστικών λύσεων, αλλά οι ίδιες οι λύσεις αποτελούν ιδέες και προτάσεις που προέρχονται αποκλειστικά από τη σχεδιαστική ομάδα. Η ιδέα είναι λοιπόν να συμμετέχουν και οι τελικοί χρήστες σε διαδικασίες διαμόρφωσης ιδεών και προκαταρκτικής σχεδίασης, έτσι ώστε το τελικό προϊόν να αντικατοπτρίζει σε μεγαλύτερο βαθμό τις πραγματικές ανάγκες και επιθυμίες τους. Η ευρεία συμμετοχή

χρηστών εγείρει βεβαίως διάφορα διαδικαστικά θέματα, όπως η διαχείριση των ιδεών και των προτάσεών τους, ο βαθμός συνεισφοράς τους στο τελικό προϊόν και η γενικότερη διαδικασία λήψης αποφάσεων. Επιπλέον, δημιουργεί νέες ανάγκες διαμοίρασης εγγράφων που αποτυπώνουν διάφορες όψεις των σχεδιαστικών λύσεων σε χρήστες που δεν κατέχουν εξειδικευμένα εργαλεία λογισμικού. Τέλος, θα ήταν επιθυμητή και η δυνατότητα των ίδιων των χρηστών να μπορούν να παρέμβουν άμεσα πάνω στις λύσεις και να δοκιμάσουν τις δικές τους ιδέες και εναλλακτικές διαμορφώσεις στην πράξη.

4.2 Προσφερόμενες δυνατότητες

Η σημαντικότερη ίσως προσφερόμενη δυνατότητα των εικονικών κόσμων στον χώρο της σχεδίασης είναι η τρισδιάστατη απεικόνιση. Το αντικείμενο της σχεδίασης μπορεί να παρουσιαστεί με τις απαιτούμενες λεπτομέρειες και στον απαιτούμενο βαθμό ρεαλισμού αξιοποιώντας τις σύγχρονες δυνατότητες των τρισδιάστατων γραφικών και να μεταδοθεί εύκολα η ιδέα στα μέλη της σχεδιαστικής ομάδας ή και στους πελάτες. Άλλωστε οι απεικονίσεις αυτές δεν είναι στατικές· αφενός μπορεί κάποιος να πλοηγηθεί σε αυτές και να τις παρατηρήσει από διάφορες απόψεις, αφετέρου θα μπορούσαν να συμπεριλαμβάνουν και διαδραστικά στοιχεία ή και πολλαπλά «στρώματα» πληροφοριών, ώστε να εμπλουτίζεται ακόμα περισσότερο η αναπαράσταση. Για τους παραπάνω λόγους, οι περιοχές σχεδίασης που φαίνεται να επωφελούνται περισσότερο από τις δυνατότητες των εικονικών κόσμων είναι αυτές στις οποίες η φυσική μορφή του τελικού προϊόντος παίζει σημαντικό ρόλο στο αποτέλεσμα, όπως π.χ. η αρχιτεκτονική, η διακόσμηση ή η σχεδίαση προϊόντων.

Εκτός όμως από την απεικόνιση, οι εικονικοί κόσμοι μπορούν να συνεισφέρουν σημαντικά και ως περιβάλλον υποστήριξης των διάφορων σχεδιαστικών διεργασιών.

Στο προκαταρκτικό στάδιο της ανάλυσης απαιτήσεων θα μπορούσαν να αξιοποιηθούν με δύο τρόπους: για τη ρεαλιστική αναπαράσταση του περιβάλλοντος πλαισίου του προβλήματος και για τη συγκέντρωση στοιχείων και πληροφοριών σχετικά με το πρόβλημα. Με δεδομένο ότι οι εικονικοί κόσμοι παρέχουν τη δυνατότητα εύκολης διαμόρφωσης χώρων και αντικειμένων, μπορεί μια σχεδιαστική ομάδα να κατασκευάσει ένα περιβάλλον το οποίο θα περιγράφει την τρέχουσα κατάσταση, για την οποία αναμένεται να προταθεί κάποια σχεδιαστική λύση. Αυτό θα βοηθήσει την ομάδα να έχει συγκεντρωμένα σε οπτική μορφή όλα τα στοιχεία που αναμένεται να βοηθήσουν στη φάση του ιδεασμού. Επιπλέον, μπορεί να προσθέσει στον εικονικό κόσμο και άλλα στοιχεία που περιγράφουν το πρόβλημα και πιθανές σχεδιαστικές κατευθύνσεις, όπως φωτογραφίες, κείμενα, ιστοσελίδες, βίντεο κ.λπ., δημιουργώντας έναν εικονικό πίνακα διαθέσεων (moodboard). Ο πίνακας αυτός, ο οποίος στη φυσική του μορφή είναι ένας πίνακας ανακοινώσεων στον οποίο αναρτώνται φωτογραφίες και σημειώσεις και χρησιμοποιείται συχνά από τις σχεδιαστικές ομάδες, δεν είναι υποχρεωτικό να αναπαρασταθεί σε δύο διαστάσεις. Θα μπορούσε να επεκτείνεται στον χώρο μοιράζοντας τα σχετικά δεδομένα σε κάποια λογική μορφή ή τοποθετώντας τα σε σχέση με αντίστοιχες περιοχές του περιβάλλοντος πλαισίου.

Στο στάδιο της προκαταρκτικής σχεδίασης μπορούν οι σχεδιαστές να επωφεληθούν από τις προσφερόμενες λειτουργίες γρήγορης διαμόρφωσης και αναδιάταξης αντικειμένων και να παρουσιάσουν τις ιδέες τους ή να δοκιμάσουν εναλλακτικές λύσεις. Επιπλέον, ενδεικτικά σενάρια χρήσης ενός προϊόντος, μιας υπηρεσίας ή ενός χώρου μπορούν να εκτελεστούν μέσω των ενσαρκώσεων των χρηστών και των αλληλεπιδράσεών τους με αντικείμενα του περιβάλλοντος. Τα σενάρια αυτά θα μπορούσαν να καταγραφούν μέσω διαδικασιών καταγραφής οθόνης (screen recording) και να παραχθούν βίντεο που μεταδίδουν την ιδέα. Η τεχνική αυτή (machinima) χρησιμοποιείται συχνά στους εικονικούς κόσμους για την απόδοση ιστοριών.

Στο στάδιο της αναλυτικής σχεδίασης οι προσφερόμενες δυνατότητες των εικονικών κόσμων είναι σαφώς πιο περιορισμένες, καθώς τα σημερινά περιβάλλοντα που υποστηρίζουν κατασκευή περιεχομένου εντός κόσμου, όπως το Second Life, στερούνται εξειδικευμένων εργαλείων μοντελοποίησης. Υποστηρίζουν όμως λειτουργίες εισαγωγής εξωτερικών τρισδιάστατων μοντέλων, με αποτέλεσμα να μπορεί να παρουσιαστεί εντός του κόσμου ένα αναλυτικό σχέδιο κατασκευασμένο σε κάποιο εξειδικευμένο πρόγραμμα μοντελοποίησης.

Τέλος, στο στάδιο της αξιολόγησης μπορούν οι σχεδιαστές να επωφεληθούν από τη δυνατότητα εξερεύνησης και διάδρασης με το κατασκευασμένο περιεχόμενο και να χρησιμοποιήσουν το μέσο για να κατασκευάσουν και να αξιολογήσουν «εικονικά πρωτότυπα». Αν και οι διαστάσεις του αντικειμένου σχεδίασης που θα αξιολογηθούν θα είναι σαφώς περιορισμένες, θα υπάρχει παρόλα αυτά δυνατότητα γρήγορης επισκόπησης, χρήσης και ανατροφοδότησης τόσο από ειδικούς όσο και από τελικούς χρήστες πολύ πριν τα τελικά στάδια παραγωγής. Με τον τρόπο αυτόν είναι ευκολότερο να εντοπιστούν πιθανά σχεδιαστικά λάθη και παραβιάσεις περιορισμών, τόσο σε αισθητικό όσο και σε λειτουργικό επίπεδο, και να διαμορφωθούν κατάλληλες προτάσεις επανασχεδίασης. Η συμμετοχή των χρηστών μέσω της ενσάρκωσής τους στην

αξιολόγηση κάποιου «εικονικού πρωτοτύπου» τούς επιτρέπει να αντιληφθούν τα σχεδιασμένα αντικείμενα και τους χώρους σε κατάλληλη κλίμακα, αλλά και να δοκιμάσουν σενάρια ομαδικής χρήσης ενός χώρου, προϊόντος ή υπηρεσίας.

Οι απαιτήσεις συνεργασίας των σχεδιαστών μπορούν να καλυφθούν σε μεγάλο βαθμό μέσω των εργαλείων σύγχρονης και ασύγχρονης επικοινωνίας που προσφέρονται στους εικονικούς κόσμους. Όπως έχουμε αναφέρει και σε προηγούμενα κεφάλαια, οι εικονικοί κόσμοι μπορούν να υποστηρίξουν συνεργατικές δραστηριότητες μεγάλων ομάδων, όπως συναντήσεις εργασίας εντός του κόσμου. Σε αντίθεση με συχνά χρησιμοποιούμενα εργαλεία τηλεδιάσκεψης μέσω ήχου ή/και βίντεο, όπως το Skype, στους εικονικούς κόσμους οι παρευρισκόμενοι μπορούν να συνομιλούν ενώ συνυπάρχουν, κινούνται και αλληλεπιδρούν σε ένα διαμοιρασμένο περιβάλλον. Αυτό σημαίνει ότι οι σχεδιαστές μπορούν να αναφέρονται πάνω στο ίδιο το αντικείμενο σχεδίασης καταδεικνύοντας συγκεκριμένες πτυχές του, να επιδεικνύουν άμεσα εναλλακτικές διαμορφώσεις και προτάσεις, να επισυνάπτουν συνοδευτικά έντυπα και να δέχονται σχόλια και αξιολογήσεις κατά τη διάρκεια των σχεδιαστικών διεργασιών. Η διαδικασία αυτή μπορεί να λειτουργήσει και ασύγχρονα. Καθένας από τους συνεργάτες μπορεί να επισκεφτεί το περιβάλλον, να αξιολογήσει το μέχρι τώρα αποτέλεσμα, να προτείνει αλλαγές, να επισυνάψει σχόλια κ.λπ. Η διαδικασία αυτή μπορεί βεβαίως να οργανωθεί ακόμα πιο αποτελεσματικά αν υποστηρίζονται ρόλοι και δικαιώματα πρόσβασης, κάτι που προσφέρεται σήμερα σε μεγάλο βαθμό στους δημοφιλείς εικονικούς κόσμους.

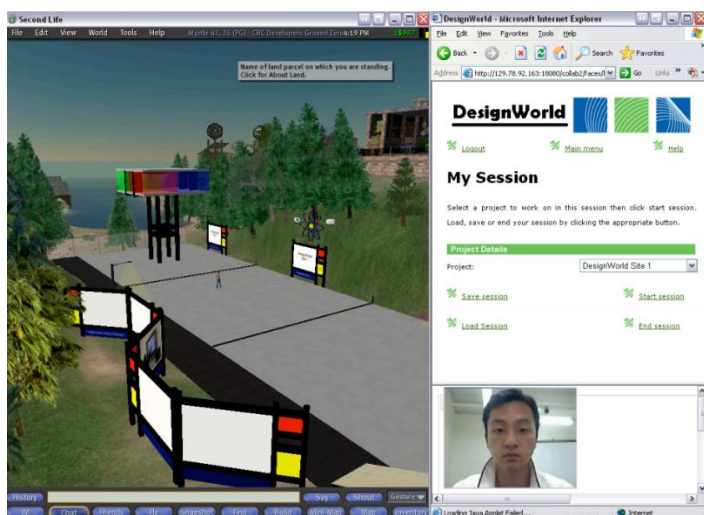
Τέλος, οι εικονικοί κόσμοι μπορούν να υποστηρίξουν και διεργασίες συμμετοχικής σχεδίασης, καθώς προσφέρουν απλές και εύκολες στη χρήση και στην εκμάθηση σχεδιαστικές λειτουργίες και ένα περιβάλλον περισσότερο φιλικό προς τον χρήστη σε σχέση με τα επαγγελματικά εργαλεία. Άλλωστε, το γεγονός ότι ένα απέραντο εικονικό περιβάλλον χτίστηκε στον κόσμο του Second Life από τους χρήστες του, που δεν ήταν κατ' ανάγκη επαγγελματίες σχεδιαστές, είναι μια ισχυρή ένδειξη ότι τα σχεδιαστικά του εργαλεία μπορούν να αξιοποιηθούν από απλούς χρήστες και να οδηγήσουν σε ενδιαφέροντα αποτελέσματα. Στα πλαίσια λοιπόν διεργασιών συμμετοχικής σχεδίασης, οι ίδιοι οι τελικοί χρήστες μπορούν να συνεισφέρουν με τις δικές τους σχεδιαστικές ιδέες, έστω και χαμηλής πιστότητας, ή και να συμπληρώσουν τα σχέδιά τους με επιπλέον εικόνες και έγγραφα που μπορούν να ανεβάσουν στον κόσμο. Ακόμα όμως κι αν δεν θέλουν να εμπλακούν σε διεργασίες διαμόρφωσης νέων αντικειμένων, έχουν τη δυνατότητα πλοηγούμενοι μέσα στο περιβάλλον να εξερευνήσουν τις υπάρχουσες λύσεις, να προσφέρουν ανατροφοδότηση και να αντιπροτείνουν τις δικές τους ιδέες και κατευθύνσεις σε αφηρημένη μορφή, τις οποίες μπορούν στη συνέχεια να επεξεργαστούν οι σχεδιαστές και να συνυπολογίσουν σε μελλοντικές προτάσεις.

4.3 Επισκόπηση εφαρμογών

Η πιο συχνή χρήση των εικονικών κόσμων στον χώρο της σχεδίασης είναι στα πλαίσια της συνεργασίας σχεδιαστών πάνω σε κάποιο κοινό έργο, ιδιαίτερα στις πρώιμες φάσεις της προκαταρκτικής σχεδίασης. Στις εφαρμογές αυτές μια ομάδα σχεδιαστών προσπαθεί εντός του κόσμου να διαμορφώσει και να εξετάσει πιθανές λύσεις αποτυπώνοντας τις ιδέες σε τρισδιάστατη μορφή. Μία από τις πρώτες εφαρμογές αυτής της μορφής ήταν το DesignWorld (Gu κ.ά., 2011), ένα περιβάλλον βασισμένο στο Active Worlds, το οποίο αποτελούνταν από δύο τμήματα: τον τρισδιάστατο κόσμο στον οποίο συνεργάζονταν οι σχεδιαστές και μια δισδιάστατη διεπαφή βασισμένη στο Web για επικοινωνία, ανταλλαγή αρχείων και διαχείριση έργου. Το Active Worlds δεν περιλαμβάνει εργαλεία κατασκευής περιεχομένου εντός κόσμου, αλλά υποστηρίζει την εισαγωγή εξωτερικών αρχείων. Κατά συνέπεια, οι σχεδιαστές δούλευαν σε μεγάλο βαθμό σε σχεδιαστικά εργαλεία εκτός κόσμου, ενώ η συνεργασία τους περιοριζόταν στην αποδοχή ή απόρριψη των επιμέρους τμημάτων του έργου και στην τοποθέτησή τους για τη σύνθεση της τελικής λύσης. Το Design World αργότερα μεταφέρθηκε στην πλατφόρμα του Second Life, οπότε υπήρχε η επιπρόσθετη δυνατότητα συνεργασίας πάνω στην κατασκευή των διάφορων στοιχείων του αντικειμένου σχεδίασης (εικ. 6.5). Μελέτες που πραγματοποιήθηκαν πάνω στη χρήση του Design World έδειξαν ότι οι σχεδιαστές μπορούσαν να συνεργαστούν αποδοτικά στη διαμόρφωση των προκαταρκτικών λύσεων και να καταλήξουν σε ένα ικανοποιητικό αποτέλεσμα.

Σε ορισμένες περιπτώσεις κατασκευάστηκαν και ειδικά εργαλεία εντός εικονικού κόσμου για την υποστήριξη της συνεργατικής σχεδίασης. Έτσι, ενώ το Design World βασιζόταν σε μια δεύτερη διεπαφή σε μορφή ιστοσελίδας για να παρέχει εξειδικευμένες λειτουργίες προς όφελος των σχεδιαστών, σε άλλες εφαρμογές κατασκευάστηκαν εργαλεία βασισμένα στη γλώσσα LSL του Second Life και του OpenSimulator, τα οποία ήταν διαθέσιμα στους σχεδιαστές εντός του κόσμου. Κατά συνέπεια, δεν απαιτούνταν από αυτούς να αλλάζουν διαρκώς στιλ διεπαφής και μοντέλο αλληλεπίδρασης κατά τη διάρκεια της εργασίας τους. Σε μια

τέτοια εφαρμογή στο περιβάλλον OpenSimulator (Koutsabasis κ.ά., 2012) αναπτύχθηκαν εξειδικευμένα εργαλεία, όπως χώροι ανακοινώσεων που επέτρεπαν την από κοινού προσθήκη και επεξεργασία κειμένου, σημειώσεις τύπου Post-It για σχόλια πάνω σε ειδικούς πίνακες, τρισδιάστατες επισημειώσεις που μπορούσαν να τοποθετηθούν σε χώρους και αντικείμενα, προβολείς παρουσιάσεων κ.ά. Τα εργαλεία αυτά αξιοποιήθηκαν σε τρεις ανεξάρτητες μελέτες. Η πρώτη ήταν μια συνεδρία επισκόπησης, όπου ένας αρχιτέκτονας παρουσίασε μια λύση σε ένα ζευγάρι πελατών και δεχόταν ανατροφοδότηση και αντιπροτάσεις. Η δεύτερη ήταν η συνεργατική σχεδίαση ενός χώρου επιστημονικού εργαστηρίου από ομάδα σχεδιαστών και η τρίτη η συνεργατική σχεδίαση ενός πολυμεσικού συστήματος πληροφοριών από φοιτητές στα πλαίσια μαθήματος. Οι μελέτες κατέδειξαν την προστιθέμενη αξία των εικονικών κόσμων στην εννοιολογική σχεδίαση και στην ανατροφοδότηση από τους πελάτες.



Εικόνα 6.5 Το περιβάλλον του Design World στο Second Life.

Μια άλλη ενδιαφέρουσα κατηγορία εφαρμογών σχεδίασης σε εικονικούς κόσμους είναι η σχεδίαση και μελέτη «έξυπνων» κτιρίων. Τα έξυπνα κτίρια είναι ιδιωτικοί ή δημόσιοι χώροι οι οποίοι είναι επαυξημένοι με «διάχυτη ευφυΐα», δηλαδή περιλαμβάνουν έναν αριθμό από αισθητήρες και επιδραστές που εκτελούν ενέργειες ανάλογα με την κατάσταση του περιβάλλοντος για να εξυπηρετήσουν ανάγκες του χρήστη. Για παράδειγμα, θα μπορούσαν να ενεργοποιούνται τα φώτα σε δωμάτια στα οποία ανιχνεύεται ανθρώπινη παρουσία και να μεταβάλλεται η έντασή τους ανάλογα με τον φυσικό φωτισμό του περιβάλλοντος. Το Second Life έχει χρησιμοποιηθεί για τη σχεδίαση και αξιολόγηση έξυπνων κτιρίων, καθώς δίνει τη δυνατότητα προγραμματισμού της διαδραστικής συμπεριφοράς των αντικειμένων του. Σε ένα σχετικό σχεδιαστικό έργο (Merrick κ.ά., 2011) κατασκευάστηκε ένα εικονικό αντίγραφο ενός χώρου συνεργασιών, προστέθηκαν εικονικοί αισθητήρες και επιδραστές οι οποίοι επικοινωνούσαν μεταξύ τους και δοκιμάστηκαν διάφορες εναλλακτικές τοποθετήσεις και κανόνες λειτουργίας. Στόχος ήταν να καταλήξει η ομάδα σχεδιαστών σε μια καλή λύση για την αυτόματη ενεργοποίηση λειτουργιών του χώρου, όπως τα φώτα και οι συσκευές παρουσίασης, με βάση τις ενέργειες των χρηστών σε αυτόν.

Οι εικονικοί κόσμοι αξιοποιούνται μεταξύ άλλων και στην εκπαίδευση μελλοντικών σχεδιαστών. Ένα περιβάλλον το οποίο επιτρέπει τον καθορισμό υποθετικών σεναρίων, δίνει τη δυνατότητα εύκολης και γρήγορης συνεργατικής διαμόρφωσης διάφορων εναλλακτικών λύσεων και τελικά μπορεί να οδηγήσει σε χρήσιμα συμπεράσματα μέσω της επισκόπησης της ποιότητας των λύσεων αυτών είναι σίγουρα ένα χρήσιμο εργαλείο εξάσκησης. Μία ενδεικτική εφαρμογή αυτής της μορφής είναι ένα κατάλληλα διαμορφωμένο περιβάλλον στο OpenSimulator που αναπτύχθηκε στο Πανεπιστήμιο Αιγαίου και αξιοποιήθηκε με επιτυχία σε μαθήματα τύπου Στούντιο. (Vosinakis & Koutsabasis, 2013) Αυτού του είδους τα μαθήματα υπάρχουν σχεδόν σε όλα τα τμήματα σχεδίασης και απαιτούν από ομάδες φοιτητών να φέρουν σε πέρας κάποιο σχεδιαστικό έργο, πραγματικό ή υποθετικό, αντίστοιχης πολυπλοκότητας. Στο περιβάλλον του εικονικού κόσμου αναπτύχθηκαν κατάλληλα εργαλεία και διαμορφώθηκαν ειδικοί χώροι για την υποστήριξη των διάφορων φάσεων της σχεδιαστικής διαδικασίας, με αποτέλεσμα οι φοιτητές να χρησιμοποιούν κατά βάση το μέσο αυτό για να συλλέξουν τα στοιχεία τους, να κατασκευάσουν, να παρουσιάσουν και αξιολογήσουν το τελικό τους σχέδιο (εικ. 6.6). Σε ένα άλλο παράδειγμα χρησιμοποιήθηκε εικονικός κόσμος στην εκπαίδευση πολιτικών μηχανικών

πάνω στη ροή εργασιών και την ασφάλεια στην υλοποίηση κατασκευαστικών έργων (Ku & Mahabaleshwar, 2011) Τα διάφορα στάδια ανάπτυξης ενός οικοδομήματος μοντελοποιήθηκαν στον κόσμο και οι φοιτητές μπορούσαν να εξασκηθούν μέσα από την κατασκευή σκαλωσιών ενός εξελισσόμενου έργου αλλά και την επισκόπησή τους ως προς την παραβίαση κανόνων ασφαλείας.



Εικόνα 6.6 Σχεδίαση διαδραστικών συστημάτων στον εικονικό κόσμο του Παν/μίου Αιγαίου.

Τέλος, και στον χώρο της συμμετοχικής σχεδίασης έχουν πραγματοποιηθεί αξιοσημείωτες προσπάθειες. Σε μία περίπτωση αξιοποιήθηκε ένα εικονικό περιβάλλον εμπύθισης με στόχο τη συμμετοχική σχεδίαση εργαζομένων και σχεδιαστών για την επαναδιαμόρφωση των κτιρίων και της οργάνωσης των εργασιών ενός φαρμακείου (Mobach, 2008). Μέσω του εικονικού περιβάλλοντος μπορούσαν οι εργαζόμενοι να εξετάσουν τις διάφορες προτεινόμενες λύσεις βλέποντας πώς αυτές αναμένεται να επηρεάσουν τη διάταξη του χώρου στον οποίο εργάζονται και τη ροή των εργασιών τους, και στη συνέχεια να παρέχουν τα δικά τους σχόλια προς τους σχεδιαστές και να συμπληρώνουν τις δικές τους ιδέες. Το περιβάλλον ήταν εμπλουτισμένο και με εικονικούς πράκτορες οι οποίοι αναπαριστούσαν εργαζομένους και πελάτες κατά την εκτέλεση τυπικών διεργασιών. Το αποτέλεσμα κρίθηκε επιτυχημένο και η λύση του εικονικού περιβάλλοντος θεωρήθηκε αρκετά οικονομική, καθώς μπορεί να οδηγήσει στον έγκαιρο εντοπισμό σχεδιαστικών λαθών πριν την υλοποίηση των εργασιών. Σε μια άλλη περίπτωση πραγματοποιήθηκε συμμετοχική σχεδίαση από σχεδιαστές και απλούς χρήστες εντός του Second Life σε τρία διαφορετικά έργα, δύο εκ των οποίων αφορούσαν προϊόντα και υπηρεσίες γνωστών πολυεθνικών εταιριών (Kohler κ.ά., 2011). Η διαδικασία οργανώθηκε με τρόπο που να παρέχονται κίνητρα συμμετοχής στους απλούς χρήστες εστιάζοντας στη διασκέδαση και στην πλούσια εμπειρία τους εντός του περιβάλλοντος. Οι χρήστες ανάλογα με την εμπειρία και τις ικανότητές τους συνεισέφεραν με τις ιδέες τους, με την κατασκευή νέων σχεδίων εντός του κόσμου και με την επισκόπηση και αξιολόγηση των προτεινόμενων λύσεων. Τα αποτελέσματα των μελετών έδειξαν ότι, αν έχει οργανωθεί σωστά η διαδικασία και παρέχονται κατάλληλα κίνητρα, οι επισκέπτες είναι διατεθειμένοι να αφιερώσουν χρόνο και να συνεισφέρουν με τις ιδέες τους προς τη διαμόρφωση περισσότερο φιλικών προς τον τελικό χρήστη λύσεων.

4.4 Καλές πρακτικές

Αυτό που προκύπτει από την επισκόπηση των προσφερόμενων δυνατοτήτων και των εφαρμογών είναι ότι οι εικονικοί κόσμοι φαίνεται να παρέχουν αρκετές δυνατότητες αξιοποίησης στον χώρο της σχεδίασης. Θα πρέπει εντούτοις να είναι εξοπλισμένοι με εξειδικευμένα εργαλεία και να περιλαμβάνουν κατάλληλα διαμορφωμένους χώρους για την καλύτερη υποστήριξη των διάφορων δραστηριοτήτων.

Μια πρώτη καλή πρακτική είναι η συμπερίληψη ειδικών εργαλείων ενίσχυσης της επικοινωνίας και συνεργασίας. Τα σχεδιαστικά έργα είναι συνήθως μεγάλης διάρκειας, αρκετά περίπλοκα και απαιτούν τη

συνεργασία πολλαπλών εμπλεκόμενων μερών διαφορετικών ειδικοτήτων και με διαφορετικές ανάγκες πληροφόρησης σχετικά με το έργο. Εργαλεία που επιτρέπουν τον εύκολο σχολιασμό και ανατροφοδότηση, τη διαχείριση ρόλων και δικαιωμάτων πρόσβασης, την οργάνωση ενός σύνθετου έργου μπορούν να έχουν σημαντικό υποστηρικτικό ρόλο.

Ένα χαρακτηριστικό στοιχείο των σχεδιαστικών έργων είναι ότι περιλαμβάνουν δραστηριότητες διαφόρων ειδών, όπως συζητήσεις, συγκέντρωση και οργάνωση στοιχείων, καταγραφή ιδεών, διαμόρφωση λύσεων, επισκόπηση και ανατροφοδότηση κ.ά. Για τον λόγο αυτόν, μια ακόμη καλή πρακτική είναι η διαμόρφωση κατάλληλων χώρων για την υποστήριξη των διαφορετικών φάσεων και η αντίστοιχη οργάνωση των διαδικασιών. Θα μπορούσαν να περιλαμβάνονται χώροι συνομιλιών και παρουσιάσεων, χώροι συνεργατικής κατασκευής λύσεων, χώροι επίδειξης και χρήσης των κατασκευασμένων αντικειμένων κ.ά.

Οι δυνατότητες προσομοίωσης των εικονικών κόσμων τούς επιτρέπουν να συμπεριλαμβάνουν διαδραστικές συμπεριφορές στα αντικείμενά τους. Το χαρακτηριστικό αυτό είναι χρήσιμο να αξιοποιείται σε σχεδιαστικά έργα, ώστε να δοκιμάζεται η λειτουργία ενός σχεδιασμένου προϊόντος ή υπηρεσίας από αξιολογητές και τελικούς χρήστες και να προσφέρεται μια πιο πλούσια εμπειρία που θα οδηγήσει σε αντίστοιχα εμπλουτισμένη ανατροφοδότηση. Εκτός από τη διαδραστική συμπεριφορά των ίδιων των αντικειμένων, μπορούν να χρησιμοποιηθούν εικονικοί πράκτορες για την αναπαράσταση τυπικών ανθρώπινων συμπεριφορών σε σχεδιασμένα κτίρια και δημόσιους χώρους.

Ένα τελευταίο στοιχείο το οποίο είναι απαραίτητο στη διαδικασία της σχεδίασης και φαίνεται να λείπει από τις σημερινές πλατφόρμες εικονικών κόσμων είναι τα εξειδικευμένα εργαλεία μοντελοποίησης. Παρόλο που το Second Life προσφέρει λειτουργίες τρισδιάστατης μοντελοποίησης, οι δυνατότητές του είναι πολύ περιορισμένες σε σχέση με αυτές των σύγχρονων σχεδιαστικών προγραμμάτων. Αυτή η έλλειψη δυσχεραίνει το έργο των επαγγελματιών σχεδιαστών και έχει επισημανθεί ως μία από τις βασικές αιτίες για την περιορισμένη υιοθέτηση των εικονικών κόσμων ως μέσο συνεργατικής σχεδίασης. Είναι απαραίτητο λοιπόν να ενισχυθούν τα περιβάλλοντα αυτά με τα απαραίτητα εξειδικευμένα εργαλεία μοντελοποίησης, ώστε να είναι πιο αποδοτική η χρήση τους από τους σχεδιαστές.

5 Επιχειρείν

Ο τελευταίος χώρος εφαρμογής που θα εξετάσουμε είναι η επιχειρηματική δραστηριότητα. Η ανάπτυξη και διάδοση του παγκόσμιου ιστού έδωσε από τα τέλη της δεκαετίας του '90 σε επιχειρήσεις και ιδιώτες τη δυνατότητα να διεκπεραιώνουν συναλλαγές με πελάτες ή και με άλλες επιχειρήσεις μέσω διαδικτύου. Αυτό οδήγησε σταδιακά στην εμφάνιση και εξάπλωση ηλεκτρονικών καταστημάτων καθώς και στην ανάδυση νέων μορφών διαφήμισης και προώθησης προϊόντων και υπηρεσιών με ψηφιακά μέσα. Τα μοντέλα αυτά εξελίχθηκαν με τον χρόνο και προσαρμόστηκαν στις νέες κατευθύνσεις του διαδικτύου, όπως τα κοινωνικά δίκτυα και οι κινητές συσκευές. Οι εικονικοί κόσμοι και τα δικτυακά παιχνίδια πολλαπλών χρηστών δεν θα μπορούσαν φυσικά να μείνουν ανεπηρέαστα. Άλλωστε αποτελούν και αυτά χώρους κοινωνικής δικτύωσης στους οποίους μάλιστα ορισμένοι χρήστες έχουν την τάση να αφιερώνουν αρκετό χρόνο της ημέρας τους. Πέρα από τη διαφήμιση ή και πώληση προϊόντων του πραγματικού κόσμου, στους εικονικούς κόσμους προκύπτει η ανάγκη και για αντίστοιχα προϊόντα που αφορούν το ίδιο το εικονικό περιβάλλον, την ενσάρκωση του χρήστη ή και τις δυνατότητες δράσης του σε αυτό (Kock, 2008). Έτσι, αναδύθηκαν «εικονικά νομίσματα» (virtual currencies) για την πραγματοποίηση συναλλαγών εντός κόσμου και προέκυψε ένα νέο είδος επιχειρηματικότητας που αναφερόταν σε προϊόντα και υπηρεσίες του εικονικού κόσμου.

Το εικονικό νόμισμα είναι ένα είδος νομίσματος το οποίο δεν έχει φυσική υπόσταση και χρησιμοποιείται αποκλειστικά από τα μέλη κάποιας εικονικής κοινότητας. Περιβάλλοντα παιχνιδιών, όπως το World of Warcraft, αλλά και εικονικοί κόσμοι γενικού τύπου, όπως το Second Life, δημιούργησαν εικονικά νομίσματα με στόχο να διευκολύνουν τις συναλλαγές μεταξύ των χρηστών τους (Papagiannidis κ.ά., 2008). Μπορούν λοιπόν στους κόσμους αυτούς οι χρήστες να πουλήσουν ή να αγοράσουν «αντικείμενα» του παιχνιδιού, να ζητήσουν ή να προσφέρουν υπηρεσίες, να επεκτείνουν τις δυνατότητές τους στο παιχνίδι ανταλλάσσοντας τα εικονικά νομίσματα του περιβάλλοντος. Οι δοσοληψίες αφορούν τόσο τον χρήστη με τον κόσμο του παιχνιδιού, π.χ. στο Second Life μπορεί κάποιος να νοικιάσει γη με εικονικά νομίσματα, όσο και τους χρήστες μεταξύ τους, π.χ. ένας χρήστης κατασκεύασε ένα ελκυστικό αντικείμενο και το πουλάει έναντι κάποιου ποσού. Ορισμένα εικονικά νομίσματα είναι «κλειστά», δηλαδή σχετίζονται μόνο με τον κόσμο του παιχνιδιού, όπως π.χ. στο World of Warcraft που μπορεί κάποιος να κερδίσει χρήματα μέσω των επιτευγμάτων του στο παιχνίδι. Σε άλλες περιπτώσεις τα νομίσματα έχουν μονόδρομη ροή, δηλαδή μπορεί κάποιος με

πραγματικά χρήματα να αγοράσει εικονικά νομίσματα. Τέλος, υπάρχουν και περιπτώσεις όπως το Second Life, όπου το νόμισμά του, το δολάριο Linden (L\$) μπορεί να μετατραπεί σε πραγματικά δολάρια με βάση κάποια ισοτιμία που διαμορφώνει η εταιρεία.

Η χρήση εικονικών νομισμάτων στους κόσμους, και ιδιαίτερα στην περίπτωση του «ανοικτού» νομίσματος του Second Life, έδωσε την δυνατότητα σε ιδιώτες αλλά και επιχειρήσεις να επενδύσουν σε ένα νέο είδος απασχόλησης: την κατασκευή προϊόντων και την παροχή υπηρεσιών εντός κόσμου (Hemp, 2006). Πράγματι, πάρα πολλοί χρήστες των κόσμων αυτών ενδιαφέρονταν να βελτιώσουν την εμφάνιση της ενσάρκωσής τους, τους κατασκευασμένους χώρους που τους ανήκουν, να αποκτήσουν νέα κομψά ή «έξυπνα» αντικείμενα και πολλά άλλα. Αυτό έδωσε ώθηση σε δημιουργούς περιεχομένου και προγραμματιστές να κατασκευάσουν και να προωθήσουν ένα τεράστιο εύρος από αντικείμενα που περιλαμβάνουν ρούχα, σώματα και μαλλιά ενσάρκωσης, έπιπλα, σπίτια, έτοιμες κινήσεις χαρακτήρων, προγράμματα (scripts) για διάφορων ειδών συμπεριφορές κ.λπ. Έτσι αναπτύχθηκε μια πολύ μεγάλη αγορά με εικονικά νομίσματα μέσα στον κόσμο του Second Life, κατασκευάστηκαν εικονικά καταστήματα με μορφή και διάταξη παραπλήσια των αντίστοιχων φυσικών χώρων, προστέθηκαν διαφημίσεις προϊόντων ή καταστημάτων σε δημοφιλείς χώρους κ.ά. (εικ. 6.7). Παράλληλα προέκυψε και η ανάγκη για την παροχή υπηρεσιών εντός κόσμου, όπως η ξενάγηση επισκεπτών, η επικοινωνία με πελάτες ενός εικονικού καταστήματος κ.ά. Με δεδομένο ότι το δολάριο Linden έχει αντίκρισμα σε πραγματικά χρήματα, υπήρξαν χρήστες οι οποίοι κέρδιζαν από τη δραστηριότητά τους στο Second Life χρήματα αντίστοιχα με αυτά μιας πραγματικής δουλειάς. Το φαινόμενο αυτό ήταν ιδιαίτερα έντονο πριν από μία πενταετία, όταν το Second Life ήταν εξαιρετικά δημοφιλές. Ένα ιδιαίτερο χαρακτηριστικό αυτού του είδους των συναλλαγών σε σχέση με τις τυπικές αγοραπωλησίες μέσω ηλεκτρονικών καταστημάτων είναι ότι υπάρχει πιο άμεση επαφή του κατασκευαστή/παρόχου με τον αγοραστή, καθώς είναι και οι δύο μέλη της ίδιας εικονικής κοινότητας. Αυτό συνήθως κάνει τους αγοραστές να νιώθουν μεγαλύτερη ασφάλεια ώστε να προχωρήσουν στη δοσοληψία, ενώ παρέχει πιο άμεση ανατροφοδότηση και στους κατασκευαστές.



Εικόνα 6.7 Άποψη εικονικού πολυκαταστήματος στον κόσμο του *OpenSimulator*.

Η παρουσία μιας ευρείας και ενεργού κοινότητας χρηστών στους εικονικούς κόσμους κίνησε το ενδιαφέρον των μεγάλων επιχειρήσεων για την περαιτέρω προβολή και προώθηση των προϊόντων τους. Έτσι, εμφανίστηκαν κατασκευασμένα αντικείμενα που είχαν τη μορφή ή και το λογότυπο γνωστών εταιρειών, από ρούχα και υποδήματα μέχρι ακριβά αυτοκίνητα. Τα αντικείμενα αυτά φάνηκαν ελκυστικά σε μεγάλο μέρος των χρηστών, οι οποίοι μέσω της αγοράς και χρήσης τους διαφήμιζαν ουσιαστικά την εταιρεία. Σε ορισμένες περιπτώσεις υπήρξαν εταιρείες που έδωσαν «ειδικές ικανότητες» στα αντικείμενά τους ώστε να διαφημίσουν τις δυνατότητές τους, π.χ. κυκλοφόρησαν υποδήματα της εταιρείας Nike που έκαναν την ενσάρκωση να τρέχει πιο γρήγορα! Σε άλλες περιπτώσεις κατασκευάστηκαν ειδικοί χώροι, κτίρια, καταστήματα ή και ολόκληρες περιοχές από τις εταιρείες για την προβολή και προώθηση των προϊόντων τους. Σε πολλούς από αυτούς τους κατασκευασμένους χώρους πέρα από «εικονικά» αντικείμενα υπήρχαν ειδικοί σύνδεσμοι για την αγορά αντίστοιχων ή άλλων πραγματικών προϊόντων. Για παράδειγμα, η γνωστή εταιρεία υπολογιστών Dell είχε κατασκευάσει έναν εικονικό χώρο εργοστασίου κατασκευής υπολογιστών στο Second Life. Μέσα στον χώρο οι χρήστες μπορούσαν να συνθέσουν τον δικό τους υπολογιστή και στη συνέχεια μπορούσαν να αγοράσουν τον

υπολογιστή που κατασκεύασαν μέσω συνδέσμου στην αντίστοιχη σελίδα της εταιρείας. Τέλος, έχουν υπάρξει και εταιρείες που έχουν κατασκευάσει εικονικούς κόσμους εξ ολοκλήρου, αποσκοπώντας στην έμμεση προώθηση των προϊόντων τους. Ένα τέτοιο παράδειγμα είναι η Coca Cola, η οποία δημιούργησε το MyCoke, έναν εικονικό χώρο συνομιλίας για εφήβους.



Εικόνα 6.8 Το νησί της Dell στο Second Life.

Πέρα από την προώθηση και πώληση, μια εταιρεία μπορεί να εκμεταλλευτεί την κοινότητα χρηστών ενός εικονικού κόσμου για τη γρήγορη ανατροφοδότηση και τη συν-σχεδίαση νέων προϊόντων και υπηρεσιών. Οι εικονικοί κόσμοι μπορούν να φέρουν μια εταιρεία σε πιο στενή επαφή με δυνητικούς πελάτες της και να αξιοποιήσει τη γνώμη τους για τη βελτίωση των προϊόντων της. Όπως αναφέρθηκε και στην υποενότητα των εφαρμογών σχεδίασης, υπάρχουν εταιρείες οι οποίες έχουν εντάξει χρήστες σε διαδικασίες συμμετοχικής σχεδίασης νέων προϊόντων και υπηρεσιών.

Τέλος, οι εικονικοί κόσμοι έχουν αξιοποιηθεί και για τη μοντελοποίηση επιχειρηματικών διαδικασιών (Brown κ.ά., 2011). Με βάση την υπάρχουσα «γλώσσα» μοντελοποίησης BPML κατασκευάστηκαν αντικείμενα για τη σχεδίαση και προβολή επιχειρηματικών μοντέλων στο Second Life και προστέθηκαν επιπλέον εργαλεία συνεργασίας. Η ύπαρξη του εικονικού κόσμου και των σχετικών εργαλείων βοήθησε τους αναλυτές να έρθουν σε πιο άμεση επαφή με τους διάφορους επιχειρηματίες και ειδικούς του χώρου που εμπλέκονται στη διαδικασία που μοντελοποιήθηκε και να μπορούν να μελετήσουν μαζί και να αξιολογήσουν το μοντέλο. Με αυτόν τον τρόπο διευκολύνεται η συνεργασία και η αμοιβαία κατανόηση μεταξύ των συνεργαζόμενων μερών που μπορεί να βρίσκονται σε απομακρυσμένες περιοχές, οδηγώντας σε περισσότερο έγκυρα και κοινώς αποδεκτά μοντέλα.

σε απομακρυσμένες περιοχές, οδηγώντας σε περισσότερο έγκυρα και κοινώς αποδεκτά μοντέλα.

6 Συμπεράσματα

Από τη μελέτη των παραπάνω χώρων εφαρμογής μπορούμε να συμπεράνουμε πως οι εικονικοί κόσμοι είναι ένα μέσο με πλούσιες προσφερόμενες δυνατότητες που μπορεί να καλύψει τις ανάγκες πολλών και διαφορετικών περιοχών. Μέχρι σήμερα φαίνεται να έχουν βρει σημαντική εφαρμογή στους χώρους της εκπαίδευσης και του πολιτισμού, ενώ διαφαίνονται δυνατότητες περαιτέρω αξιοποίησης και στον χώρο της συνεργατικής σχεδίασης. Τα εκπαιδευτικά περιβάλλοντα επωφελούνται από την προβολή ποιοτικών αναπαραστάσεων του περιεχομένου, από την υποστήριξη εικονικών κοινοτήτων με εργαλεία σύγχρονης και ασύγχρονης επικοινωνίας καθώς και από τη δυνατότητα διεξαγωγής ελεύθερων δραστηριοτήτων, πειραμάτων και κατασκευής περιεχομένου εντός του κόσμου. Όλα τα παραπάνω μπορούν να οδηγήσουν σε πιο «ζωντανές», βιωματικές και παιγνιώδεις δραστηριότητες και να κεντρίσουν το ενδιαφέρον των εκπαιδευομένων.

Αντίστοιχα, ο χώρος του πολιτισμού μπορεί να εκμεταλλευτεί τα γραφικά, την κίνηση και την αλληλεπίδραση για τη δημιουργία πλούσιων εμπειριών στους επισκέπτες, που τους φέρνουν σε πιο άμεση επαφή με το πολιτισμικό περιεχόμενο. Οι εμπειρίες αυτές μπορούν να επαυξηθούν με στοιχεία προσωποποίησης ή και συμμετοχής σε ψηφιακές ιστορίες. Στον χώρο της συνεργατικής σχεδίασης η συμμετοχή πολλαπλών χρηστών σε ένα ενιαίο περιβάλλον, η δυνατότητα επικοινωνίας, συνεργασίας και συν-σχεδίασης στον κοινό χώρο και τα εργαλεία σχολιασμού και ανατροφοδότησης μπορούν να βοηθήσουν στην καλύτερη συνεργασία και στην επιτάχυνση των διαδικασιών προκαταρκτικής σχεδίασης. Τέλος, οι εικονικοί κόσμοι φαίνεται να αποτελούν και ένα ενδιαφέρον πεδίο δράσης στον χώρο της επιχειρηματικότητας, όπου έχουν αναδυθεί εικονικά νομίσματα και νέες μορφές δραστηριότητας για την κατασκευή αντικειμένων και την παροχή υπηρεσιών εντός κόσμου.

Αναφορές

- Bartle, R. (2004). *Designing virtual worlds*. New Riders.
- Bogdanovych, A., Ijaz, K. & Simoff, S. (2012). The city of uruk: Teaching ancient history in a virtual world. *Lecture Notes in Computer Science, 7502 LNAI*, 28–35.
- Bonis, B., Stamos, J., Vosinakis, S., Andreou, I. & Panayiotopoulos, T. (2009). A platform for virtual museums with personalized content. *Multimedia Tools and Applications*, 42(2), 139–159.
- Brown, J. S., Collins, A. & Duguid, P. (1989). Situated cognition and the culture of learning. *Educational Researcher*, 18(1), 32–42.
- Brown, R. A., Recker, J. C. & West, S. (2011). Using Virtual Worlds for Collaborative Business Process Modeling. *Business Process Management Journal*, 17, 546–564.
- Dede, C. (1995). The Evolution of Constructivist Learning Environments: Immersion in Distributed, Virtual Worlds. *Educational Technology*, 35(5), 46–52.
- Dickey, M. D. (2005). Three-dimensional virtual worlds and distance learning: Two case studies of Active Worlds as a medium for distance education. *British Journal of Educational Technology*, 36(3), 439–451.
- Gu, N., Kim, M. J. & Maher, M. Lou. (2011). Technological advancements in synchronous collaboration: The effect of 3D virtual worlds and tangible user interfaces on architectural design. *Automation in Construction*, 20(3), 270–278.
- Hemp, P. (2006). Avatar-based marketing. *Harvard Business Review*, 84(6).
- Hew, K.F. & Cheung, W.S. (2010). Use of three-dimensional (3-D) immersive virtual worlds in K-12 and higher education settings: A review of the research. *British Journal of Educational Technology*. 41, 1, 33–55.
- Hybs, I. & Gero, J. S. (1992). An evolutionary process model of design. *Design Studies*, 13(3), 273–290.
- Jonassen, D. (1999). Designing constructivist learning environments. In C. M. Reigeluth (Ed.), *Instructional design theories and models* (Vol. 2, pp. 215–239). Lawrence Erlbaum Associates.
- Karakatsiotis, G., Oberlander, J. & Isard, A. (2008). Building an adaptive museum gallery in Second Life. *Museums and the Web 2008. Proceedings*.
- Ketelhut, D. J. (2007). The impact of student self-efficacy on scientific inquiry skills: An exploratory investigation in river city, a multi-user virtual environment. *Journal of Science Education and Technology*, 16(1), 99–111.
- Kock, N. (2008). E-Collaboration and E-Commerce in Virtual Worlds. *Knowledge Networks*, 4(3), 308–319.
- Kolb, D. A. (1984). The Process of Experiential Learning. In *Experiential learning: Experience as the source of learning and development* (pp. 20–38).

- Koutsabasis, P., Vosinakis, S., Malisova, K. & Paparounas, N. (2012). On the value of Virtual Worlds for collaborative design. *Design Studies*, 33(4), 357–390.
- Ku, K. & Mahabaleshwar, P. S. (2011). Building interactive modeling for construction education in virtual worlds. *Electronic Journal of Information Technology in Construction*, 16(September 2010), 189–208.
- Lim, C. K., Cani, M. P., Galvane, Q., Pettre, J. & Talib, A. Z. (2013). Simulation of past life: Controlling agent behaviors from the interactions between ethnic groups. *Proceedings of the DigitalHeritage 2013*, 589–596.
- Merrick, K. E., Technology, I., Defence, A., Academy, F., Gu, N., Lecturer, S., Wales, S. (2011). Case Studies using Multiuser Virtual Worlds as an Innovative Platform for Collaborative Design, *Journal of Information Technology in Construction*, 16, 165-188.
- Mikropoulos, T. a. & Natsis, A. (2011). Educational virtual environments: A ten-year review of empirical research (1999-2009). *Computers and Education*, 56(3), 769–780.
- Mobach, M. P. (2008). Do virtual worlds create better real worlds? *Virtual Reality*, 12(3), 163–179.
- Papagiannidis, S., Bourlakis, M. & Li, F. (2008). Making real money in virtual worlds: MMORPGs and emerging business opportunities, challenges and ethical implications in metaverses. *Technological Forecasting and Social Change*, 75(5), 610–622.
- Rollings, A. & Adams, E. (2003). Andrew Rollings and Ernest Adams on game design, New Riders.
- Sequeira, L. M. & Morgado, L. (2013). Virtual Archaeology in Second Life and OpenSimulator. *Virtual World Research*.
- Soliman, M. & Guetl, C. (2010). Intelligent Pedagogical Agents in Immersive Virtual Learning Environments: A Review. *Mipro*, 827–832.
- Sylaiou, S., Liarokapis, F., Kotsakis, K. & Patias, P. (2009). Virtual museums, a survey and some issues for consideration. *Journal of Cultural Heritage*, 10(4), 520–528.
- Thomas Kohler, Johann Fueller, Kurt Matzler & Daniel Stieger. (2011). Co-Creation In Virtual Worlds: The Design Of The User Experience. *MIS Quarterly*, 35(3), 773–788.
- Tsichritzis, D. & Gibbs, S. (1991). Virtual Museums and Virtual Realities. *International Conference on Hypermedia and Interactivity in Museums*, 927–954.
- Urban, R., Marty, P. & Twidale, M. (2007). A Second Life for Your Museum: 3D Multi-User Virtual Environments and Museums. In *Museums and the Web*.
- Verón, E. & Levasseur, M. (1989). Ethnographie de l'exposition: l'espace, le corps et le sens. Centre Georges Pompidou, Bibliothèque publique d'information.
- Vosinakis, S. & Koutsabasis, P. (2013). Interaction design studio learning in virtual worlds. *Virtual Reality*, 17(1), 1919–1932.
- Warburton, S. (2009). Second Life in higher education: Assessing the potential for and the barriers to deploying virtual worlds in learning and teaching. *British Journal of Educational Technology*, 40(3), 414–426.

Περαιτέρω Μελέτη

Μπορείτε να διαβάσετε περισσότερα σχετικά με τις εκπαιδευτικές εφαρμογές των εικονικών κόσμων στο βιβλίο:

Nelson, B. C. & Erlandson, B. E. (2012). *Design for learning in virtual worlds*. Routledge.

Επίσης, μπορείτε να βρείτε χρήσιμες πληροφορίες σχετικά με την αξιοποίηση τεχνολογιών εικονικής πραγματικότητας και παιχνιδιών στο χώρο του πολιτισμού στο βιβλίο:

Champion, E. (2011). *Playing with the Past*. Springer London.

Ερωτήσεις Κατανόησης

1. Πώς θα μπορούσαν οι εικονικοί κόσμοι να αξιοποιηθούν στην εγκαθιδρυμένη μάθηση;
2. Ποιες είναι οι προσφερόμενες δυνατότητες των εικονικών κόσμων στο χώρο της εκπαίδευσης;
3. Ποιες είναι οι βασικές κατηγορίες χρήσης των εικονικών κόσμων στην εκπαίδευση; Αναφέρατε από ένα παράδειγμα.
4. Ποιοι είναι οι στόχοι των επισκεπτών ενός μουσείου; Σε ποιο βαθμό και με ποια μέσα θα μπορούσαν τα εικονικά μουσεία να τους καλύψουν;
5. Ποιες είναι οι προσφερόμενες δυνατότητες των εικονικών κόσμων στον πολιτισμό;
6. Ποια είναι τα βασικά χαρακτηριστικά των εικονικών μουσείων και τι είδους δραστηριότητες υποστηρίζουν;
7. Αναφέρατε καλές πρακτικές για τη σχεδίαση εφαρμογών πολιτισμού με τη χρήση εικονικών κόσμων.
8. Ποια είναι τα βασικά στάδια της σχεδίασης και με ποιους τρόπους θα μπορούσαν να συνεισφέρουν οι εικονικοί κόσμοι σε αυτά;
9. Αναφέρατε παραδείγματα και καλές πρακτικές συμμετοχικής σχεδίασης μέσω εικονικών κόσμων.
10. Ποιες είναι οι βασικές μορφές ηλεκτρονικής επιχειρηματικότητας στους εικονικούς κόσμους;

Ασκήσεις

1. Στη διεύθυνση <http://secondlife.com/destinations> εμφανίζονται δημοφιλείς περιοχές του Second Life κατηγοριοποιημένες ανάλογα με το βασικό χώρο εφαρμογής τους. Επισκευτείτε δύο χώρους εκπαίδευσης και δύο χώρους πολιτισμού που σας κίνησαν το ενδιαφέρον, εξερευνήστε τους και αλληλεπιδράστε με το περιεχόμενό τους. Δημιουργήστε μια αναφορά στην οποία καταγράφονται κρίσιμα σχεδιαστικά στοιχεία, ενδιαφέρουσες λύσεις, αλλά και προβλήματα που αντιμετωπίσατε.

Κεφάλαιο 7: Το Περιβάλλον OpenSimulator

Σύνοψη

Το δεύτερο μέρος του βιβλίου έχει ως αντικείμενο την εκμάθηση του περιβάλλοντος OpenSimulator για τη σχεδίαση και ανάπτυξη Εικονικών Κόσμων σε αυτό. Στο παρόν κεφάλαιο γίνεται μια εισαγωγική παρουσίαση της πλατφόρμας, της διαδικασίας εγκατάστασης και της βασικής χρήσης. Παρουσιάζεται η ιστορία και η εξέλιξη του OpenSimulator ως περιβάλλοντος ανοιχτού λογισμικού βασισμένου στην τεχνολογία του Second Life. Αναλύονται τα βασικά χαρακτηριστικά και οι δυνατότητες του περιβάλλοντος, παρουσιάζονται τα διαφορετικά μοντέλα λειτουργίας και περιγράφονται οι προσφερόμενες εναλλακτικές διανομές (distributions). Στη συνέχεια δίνονται αναλυτικές οδηγίες εγκατάστασης του περιβάλλοντος και περιγράφονται οι βασικές διαδικασίες διαχείρισης, όπως η εισαγωγή χρηστών και τα δικαιώματα πρόσβασης που θα έχουν, η διαμόρφωση του κόσμου, τα αντίγραφα ασφαλείας (backups), οι ομάδες χρηστών κ.ά. Το κεφάλαιο ολοκληρώνεται με την παρουσίαση βασικών οδηγιών χρήσης του περιβάλλοντος σύνδεσης σε ό,τι αφορά την πλοήγηση, την επικοινωνία, την αλληλεπίδραση με τα αντικείμενα του κόσμου, τη διαχείριση του αποθετηρίου και τη διαμόρφωση της ενσάρκωσης.

Προαπαιτούμενη γνώση

Απαιτούνται γνώσεις πληροφορικής, καθώς και κατανόηση βασικών εννοιών διαδικτύου και βάσεων δεδομένων.

1 Εισαγωγή

Το OpenSimulator είναι μια πλατφόρμα εικονικών κόσμων η οποία αποτελεί έργο ανοιχτού λογισμικού και βασίζεται στην τεχνολογία του Second Life. Όπως αναφέραμε και στο πρώτο κεφάλαιο, το ενδιαφέρον χαρακτηριστικό εικονικών κόσμων όπως τα Second Life και OpenSimulator είναι ότι δεν είναι προσχεδιασμένοι να εξυπηρετούν τις ανάγκες συγκεκριμένου χώρου εφαρμογής (σε αντίθεση π.χ. με κόσμους παιχνιδιών). Τα περιβάλλοντα αυτού του τύπου έχουν σχεδιαστεί ώστε να διευκολύνουν την επικοινωνία μεταξύ απομακρυσμένων χρηστών, το χτίσιμο νέων περιοχών, την ανταλλαγή ιδεών και τη συνεργασία, με αποτέλεσμα ο κάθε χρήστης να μπορεί να δημιουργήσει τον ρόλο που τον εξυπηρετεί στα πλαίσια των δραστηριοτήτων που επιθυμεί να εκτελέσει στον κόσμο (εικ. 7.1). Ένα ιδιαίτερο χαρακτηριστικό του Second Life είναι ότι παρέχει εύχρηστα εργαλεία κατασκευής και διαμόρφωσης περιεχομένου στον κόσμο καθώς και μια γλώσσα σεναρίων (script), την LSL, μέσω της οποίας μπορεί να προγραμματιστεί η συμπεριφορά των τρισδιάστατων αντικειμένων. Τα χαρακτηριστικά αυτά έκαναν το περιβάλλον εξαιρετικά δημοφιλές, με αποτέλεσμα ο κόσμος του Second Life να αποκτήσει γρήγορα μια πολύ μεγάλη βάση χρηστών. Όμως το γεγονός ότι η ανάπτυξη περιεχομένου και η απόκτηση ιδιωτικού χώρου στο περιβάλλον αυτό απαιτούν μηνιαία συνδρομή και το ότι το κόστος μπορεί τελικά να είναι αρκετά υψηλό οδήγησε πολλούς χρήστες στην αναζήτηση εναλλακτικών λύσεων με μικρότερο κόστος και περισσότερες δυνατότητες διαχείρισης. Το OpenSimulator αποτελεί σήμερα μια τέτοια λύση και καταγράφεται πλέον η τάση μετακίνησης αρκετών χρηστών αλλά και ακαδημαϊκών ιδρυμάτων προς αυτήν την πλατφόρμα.

Η ανάπτυξη του OpenSimulator ξεκίνησε το 2007 και στόχος του, όπως αναφέρουν οι δημιουργοί του, είναι όχι απλά να αποτελέσει «κλώνο» του Second Life, αλλά να προσφέρει και επιπλέον δυνατότητες, αποτελώντας μια βάση για την ανάπτυξη καινοτόμων λειτουργιών σε εικονικούς κόσμους. Πράγματι, από τη μία μεριά το OpenSimulator προσπάθησε να ενσωματώσει τις περισσότερες από τις λειτουργίες του Second Life και από την άλλη εισήγαγε και συνεχίζει να επεκτείνει έναν αριθμό από νέες δυνατότητες, όπως ορισμένες χρήσιμες επεκτάσεις της γλώσσας LSL και τους συνθετικούς χαρακτήρες (NPCs). Σήμερα το OpenSimulator εξακολουθεί και βρίσκεται σε έκδοση beta, αλλά οι δυσλειτουργίες και τα προβλήματα είναι αρκετά περιορισμένα, με αποτέλεσμα να μπορεί να χρησιμοποιηθεί με επιτυχία για την υποστήριξη πολυχρηστικών κόσμων και πλούσιων διαδραστικών εφαρμογών μέσα σε αυτούς.

Το παρόν κεφάλαιο εστιάζει στην παρουσίαση και επεξήγηση βασικών στοιχείων εγκατάστασης και χρήσης του OpenSimulator. Αρχικά θα περιγράψουμε τα τεχνικά χαρακτηριστικά και τις δυνατότητες του περιβάλλοντος. Στη συνέχεια θα παρουσιάσουμε βήμα προς βήμα την εγκατάσταση των απαιτούμενων προγραμμάτων και ορισμένων σημαντικών επιπρόσθετων λειτουργικών στοιχείων. Θα αναφερθούμε σε

χρήσιμες ενέργειες διαμόρφωσης και διαχείρισης του κόσμου και, τέλος, θα παρουσιάσουμε τη χρήση του περιβάλλοντος διεπαφής για βασικές διεργασίες όπως η πλοήγηση και η επικοινωνία.



Εικόνα 7.1 Στιγμιότυπο από τον εικονικό κόσμο του Second Life.

2 Τεχνικά χαρακτηριστικά και δυνατότητες

Το OpenSimulator, όπως και οι περισσότεροι πολυχρηστικοί εικονικοί κόσμοι, βασίζεται σε τεχνολογία πελάτη-διακομιστή (client-server), σύμφωνα με την οποία οι λειτουργίες του εικονικού κόσμου εκτελούνται σε κάποιον απομακρυσμένο διακομιστή και οι χρήστες συνδέονται μέσω ειδικών προγραμμάτων που εκτελούνται στο μηχάνημά τους και τους παρέχουν την απεικόνιση του κόσμου και τη διεπαφή με αυτόν. Η ομάδα ανάπτυξης του OpenSimulator συντηρεί και εξελίσσει μόνο την τεχνολογία διακομιστή, και το πρωτόκολλο επικοινωνίας μεταξύ πελάτη και διακομιστή που χρησιμοποιείται είναι αυτό του Second Life. Κατά συνέπεια, μπορεί να συνδεθεί κάποιος σε κόσμο του OpenSimulator με το ίδιο το πρόγραμμα σύνδεσης (viewer) του Second Life. Επιπλέον, διατίθενται ελεύθερα αρκετά εναλλακτικά προγράμματα σύνδεσης, τα οποία έχουν εξελίξει τον ανοιχτό κώδικα των διάφορων εκδόσεων του Second Life viewer. Αυτό σημαίνει ότι το βασικό περιβάλλον διεπαφής για τον τελικό χρήστη είναι το ίδιο ή παραπλήσιο με αυτό του Second Life. Όμως οι χρήστες δεν είναι υποχρεωμένοι να πληρώνουν συνδρομή στη Linden Labs για να διαχειρίζονται περιοχές και να κατασκευάζουν περιεχόμενο στον κόσμο, ενώ και οι διαχειριστές έχουν τον πλήρη έλεγχο της εγκατάστασης, συντήρησης και διαμόρφωσης του εικονικού κόσμου.

2.1 Επίπεδο διακομιστή

Οι λειτουργίες που προσφέρει το OpenSimulator στα προγράμματα σύνδεσης μοιράζονται σε έναν αριθμό από υπηρεσίες (services) που εκτελούνται παράλληλα όσο ο διακομιστής είναι σε λειτουργία. Ορισμένες από τις βασικές υπηρεσίες είναι:

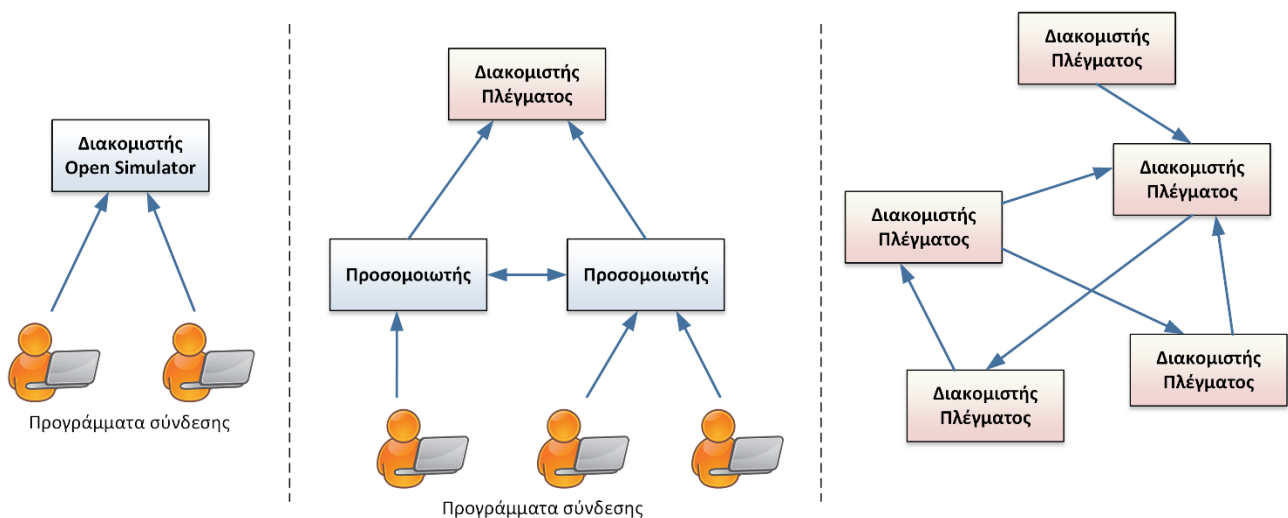
- Login service: σύνδεση των χρηστών στο περιβάλλον,
- Inventory service: διαχείριση αποθετηρίων χρηστών,
- Asset service: δημιουργία και ανάκτηση αντικειμένων του κόσμου,
- Grid service: πληροφορίες περιοχών του κόσμου,
- User account service: πληροφορίες λογαριασμού χρηστών και
- Presence service: ενέργειες συνδεδεμένου χρήστη στο περιβάλλον.

Κατά την εκτέλεση των υπηρεσιών αυτών ο διακομιστής του OpenSimulator αξιοποιεί μια *βάση δεδομένων* για την αποθήκευση και ανάκτηση των δεδομένων του περιβάλλοντος. Στη βάση δεδομένων καταγράφονται τα δεδομένα των χρηστών, τα αντικείμενα (assets) του κόσμου, η διαμόρφωση των περιοχών και πολλά άλλα. Το OpenSimulator προσφέρει στους διαχειριστές την επιλογή να χρησιμοποιήσουν μια ελαφριά βάση δεδομένων που παρέχεται μαζί με την εγκατάσταση του περιβάλλοντος, την SQLite, ή να εγκαταστήσουν και να χρησιμοποιήσουν το περιβάλλον MySQL, που αποτελεί ένα ολοκληρωμένο και ισχυρό σύστημα διαχείρισης βάσεων δεδομένων. Η επιλογή της MySQL είναι σαφώς προτιμότερη. Αφενός οι ενέργειες αποθήκευσης και ανάκτησης εκτελούνται με πολύ μεγαλύτερη ταχύτητα σε σχέση με την SQLite και αφετέρου ορισμένες από τις απαιτούμενες λειτουργίες δεν έχουν υλοποιηθεί πλήρως στην SQLite. Η τελευταία είναι καλό να χρησιμοποιείται μόνο σε εγκαταστάσεις που έχουν προσωρινό χαρακτήρα.

Το OpenSimulator έχει δύο προσφερόμενες αρχιτεκτονικές λειτουργίας:

- την αυτόνομη κατάσταση (standalone mode) και
- την κατάσταση πλέγματος (grid mode).

Στην αυτόνομη κατάσταση λειτουργίας (εικ. 7.2 αριστερά) εγκαθίσταται και εκτελείται ένας διακομιστής στον οποίο τρέχουν όλες οι υπηρεσίες και που διαχειρίζεται συνολικά τον κόσμο. Ο διακομιστής αυτός αξιοποιεί τη βάση δεδομένων για την καταγραφή και ανάκτηση των απαιτούμενων στοιχείων και όλοι οι χρήστες του κόσμου συνδέονται και επικοινωνούν αποκλειστικά με αυτόν μέσω των προγραμμάτων σύνδεσης.



Εικόνα 7.2 Αυτόνομη κατάσταση λειτουργίας (αριστερά), λειτουργία πλέγματος (κέντρο) και υπερσύνδεση πλέγματος (δεξιά).

Στην κατάσταση πλέγματος (εικ. 7.2 κέντρο) οι περιοχές του εικονικού κόσμου μπορούν να μοιραστούν σε παραπάνω από έναν διακομιστές. Καθένας από αυτούς τους διακομιστές τρέχει το δικό του *περιβάλλον προσομοίωσης (simulator)* για μία ή περισσότερες υπο-περιοχές του κόσμου και χρησιμοποιεί τη δική του βάση δεδομένων για να αποθηκεύει τα δεδομένα των περιοχών αυτών. Επιπλέον, υπάρχει ένας ακόμα διακομιστής, ο *διακομιστής πλέγματος (grid server)*, στον οποίο εκτελούνται όλες οι απαραίτητες υπηρεσίες και καταγράφονται στοιχεία που αφορούν συνολικά τον κόσμο, όπως οι λογαριασμοί των χρηστών, μέσω μιας επιπρόσθετης κεντρικής βάσης δεδομένων. Ως συνέπεια της αρχιτεκτονικής αυτής, η λειτουργία του εικονικού κόσμου μοιράζεται σε πολλαπλούς διακομιστές, οι οποίοι μπορούν να τρέχουν σε διαφορετικά υπολογιστικά συστήματα, με αποτέλεσμα να μπορεί να υποστηριχτεί πολύ μεγαλύτερος αριθμός ταυτόχρονων χρηστών αλλά και να μπορεί να επεκτείνεται ο κόσμος με νέες περιοχές, εφόσον προστίθενται επιπλέον διακομιστές στο πλέγμα.

Μια ακόμη ενδιαφέρουσα υπηρεσία που παρέχεται στο OpenSimulator είναι η *υπερσύνδεση πλέγματος (hypergrid)*. Με τη χρήση της υπερσύνδεσης μπορούν οι χρήστες ενός εικονικού κόσμου να μεταφερθούν ως επισκέπτες σε έναν άλλο, ανεξάρτητο κόσμο, εφόσον ο δεύτερος το αποδέχεται (εικ. 7.2 δεξιά). Κατά συνέπεια,

είναι δυνατόν κόσμοι που ανήκουν σε διαφορετικά πλέγματα να διασυνδέονται μεταξύ τους, δημιουργώντας κατ' αυτόν τον τρόπο ένα ευρύτερο δίκτυο κόσμων τους οποίους μπορούν να επισκέπτονται οι χρήστες.

Εκτός από τα παραπάνω, προσφέρεται και ένας αριθμός από πρόσθετες υπηρεσίες, οι οποίες μπορούν να εγκατασταθούν προαιρετικά και δεν υποστηρίζονται από την ομάδα ανάπτυξης του OpenSimulator. Οι πιο σημαντικές από αυτές είναι η *φωνητική επικοινωνία* και η *διεπαφή μέσω Web*. Η δυνατότητα επικοινωνίας μεταξύ των χρηστών μέσω φωνής μπορεί να είναι απαραίτητη σε αρκετές περιπτώσεις εφαρμογών, παραδείγματος χάριν σε εφαρμογές που περιλαμβάνουν τηλεδιάσκεψη ή συνεργασία, και δεν υποστηρίζεται άμεσα από το OpenSimulator. Προσφέρονται διαθέσιμες λύσεις ανοιχτού λογισμικού για την υποστήριξη φωνής, όπως τα Freeswitch και Mumble, ενώ και η εταιρεία Vinox παρέχει δωρεάν υπηρεσίες φωνής για διακομιστές OpenSimulator για μη εμπορική χρήση. Η διεπαφή μέσω Web είναι μία ακόμη επιθυμητή λειτουργία, τόσο για την καλύτερη υποστήριξη της εγγραφής των χρηστών του κόσμου όσο και για τη διευκόλυνση των διαδικασιών διαχείρισης. Προσφέρονται διάφορες λύσεις Web διεπαφής, με δημοφιλέστερο το WIFI (σημ: η ονομασία δεν σχετίζεται με τα γνωστά πρωτόκολλα ασύρματης δικτύωσης), το οποίο είναι εύκολο στην εγκατάσταση και παρέχεται προεγκατεστημένο σε ορισμένες διανομές του OpenSimulator.

Το OpenSimulator προσφέρεται σήμερα σε τρία διαφορετικά πακέτα εγκατάστασης (διανομές):

- το OpenSim, που είναι η βασική διανομή του προγράμματος και περιλαμβάνει μόνο το διακομιστή,
- το diva/d2, που έχει ενσωματώσει ορισμένες σημαντικές επεκτάσεις στο παραπάνω (όπως Web διεπαφή και υπηρεσία αυτόματης ενημέρωσης νεών εκδόσεων) και παρέχει αρκετά πιο απλοποιημένη διαδικασία εγκατάστασης και διαχείρισης και
- το sim-on-a-stick, το οποίο περιλαμβάνει προεγκατεστημένους τον διακομιστή, τη βάση και το πρόγραμμα πελάτη, με αποτέλεσμα να τρέχει απευθείας, χωρίς εγκατάσταση. Ο βασικός περιορισμός του είναι η μη υποστήριξη πολλαπλών χρηστών.

2.2 Επίπεδο πελάτη

Οι δυνατότητες του περιβάλλοντος διεπαφής είναι αντίστοιχες με αυτές του Second Life. Ο κάθε χρήστης διαθέτει ένα προσωπικό *αποθετήριο αντικειμένων (inventory)* στο οποίο μπορεί να αποθηκεύει σημειώσεις, αντίγραφα αντικειμένων του κόσμου, συνθετικές κινήσεις (animations), εικόνες που ανεβάζει και που μπορεί να χρησιμοποιήσει ως υφές (textures) κ.ά. Υπάρχουν εργαλεία διαμόρφωσης της εμφάνισης της ενσάρκωσης μέσω των οποίων μπορούν να τροποποιηθούν σωματικά χαρακτηριστικά αλλά και να προστεθούν ρούχα των οποίων την εμφάνιση ο χρήστης μπορεί να προσαρμόσει. Οι χρήστες έχουν λοιπόν τη δυνατότητα να τροποποιούν τη μορφή του χαρακτήρα τους ανάλογα με τις απαιτήσεις της κάθε εφαρμογής.

Η πλοήγηση των χρηστών υποστηρίζεται με διάφορους τρόπους. Υπάρχει η δυνατότητα άμεσου ελέγχου του βαδίσματος της ενσάρκωσης μέσω πληκτρολογίου ή του ποντικιού, ενώ οι ενσάρκώσεις έχουν και τη δυνατότητα να πετάζουν παρέχοντας κατ' αυτόν τον τρόπο ταχύτερη μετακίνηση και σφαιρικότερη απεικόνιση του χώρου. Μπορεί να τροποποιηθεί η άποψη του χρήστη σε σχέση με την ενσάρκωσή του (άποψη τρίτου προσώπου), όπως επίσης μπορεί ο χρήστης να βλέπει και μέσα από τα μάτια της ενσάρκωσής του (πρώτου προσώπου). Παρέχεται δυνατότητα προβολής χάρτη κάτοψης (minimap) με τοποθετημένες τις θέσεις των υπόλοιπων χρηστών και με δυνατότητα άμεσης τηλεμεταφοράς σε οποιαδήποτε θέση αυτού, ενώ υπάρχουν και εργαλεία για την επισκόπηση αντικειμένων ή περιοχών του κόσμου και την περιστροφή της κάμερας γύρω από το κέντρο τους. Όλα τα παραπάνω εργαλεία πλοήγησης βοηθούν τόσο στην εξερεύνηση του χώρου όσο και στην καλύτερη διαμόρφωση και τοποθέτηση αντικειμένων στον χώρο.

Σε ό,τι αφορά την επικοινωνία με τους άλλους χρήστες, υπάρχει η δυνατότητα συνομιλιών μέσω γραπτού κειμένου ή φωνής. Και στις δύο περιπτώσεις οι συνομιλίες είναι «χωροθετημένες», με την έννοια ότι μόνο όσοι βρίσκονται σε κοντινή απόσταση από τον συνομιλητή μπορούν να διαβάσουν ή να ακούσουν το μήνυμα. Υπάρχει επίσης η δυνατότητα αποστολής άμεσων μηνυμάτων καθώς και αναπαραγωγής έτοιμων συνθετικών κινήσεων (animations) της ενσάρκωσης από βιβλιοθήκη οι οποίες γίνονται άμεσα ορατές στον κόσμο. Οι τελευταίες μπορούν να χρησιμοποιηθούν και για μη λεκτική επικοινωνία, όπως χαιρετισμός, επιβράβευση κ.λπ. Οι προσφερόμενες μορφές επικοινωνίας δίνουν στους συμμετέχοντες τη δυνατότητα απομακρυσμένης σύγχρονης συνεργασίας μέσω παρουσιάσεων, συνομιλιών κ.λπ (εικ. 7.3).



Εικόνα 7.3 Απομακρυσμένη συνεργασία φοιτητών του Παν/μίου Αιγαίου στο περιβάλλον του OpenSimulator.

Οι χρήστες του εικονικού κόσμου μπορούν να αλληλεπιδράσουν με τα «ενεργά» αντικείμενα, αυτά δηλαδή των οποίων η συμπεριφορά είναι προγραμματισμένη. Η αλληλεπίδραση μπορεί να ενεργοποιείται με πολλούς τρόπους, όπως: με απλά κλικ, μέσω γραπτού κειμένου, με ειδικά διαμορφωμένα μενού, μέσω χωρικής προσέγγισης κ.λπ. Αντίστοιχα, τα αντικείμενα μπορούν να αντιδρούν με αλλαγή στην εμφάνιση, πρόκληση κίνησης, ενεργοποίηση συνθετικών κινήσεων των χρηστών, επικοινωνία με άλλα αντικείμενα κ.λπ. Επιπροσθέτως οι χρήστες μπορούν να καθίσουν σε ειδικά διαμορφωμένα αντικείμενα (π.χ. καρέκλες), να τα φορέσουν (π.χ. ρούχα), να τα κρατήσουν (π.χ. εργαλεία) ή και να αποθηκεύσουν επιλεγμένα αντικείμενα στο αποθετήριό τους και να δημιουργούν στη συνέχεια πολλαπλά αντίγραφα τους.

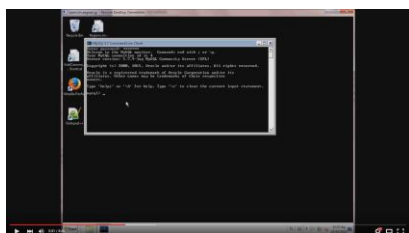
Ένα ακόμα ενδιαφέρον χαρακτηριστικό του εικονικού κόσμου του OpenSimulator είναι και η δυνατότητα «χτισίματος» μέσα στον κόσμο. Παρέχεται δηλαδή η δυνατότητα κατασκευής σύνθετων τρισδιάστατων αντικειμένων (από μικροαντικείμενα και έπιπλα μέχρι μεγάλα κτίρια) αλλά και διαμόρφωσης του τοπίου (π.χ. δημιουργία βουνών, λιμνών, νησιών, δέντρων κ.λπ.) στους χρήστες που έχουν τα κατάλληλα δικαιώματα. Κατά συνέπεια, όπως ακριβώς συμβαίνει με τις εφαρμογές που βασίζονται στη φιλοσοφία του Web 2.0, το περιεχόμενο του περιβάλλοντος συν-διαμορφώνεται και εξελίσσεται από τους χρήστες. Σε αντίθεση όμως με παλαιότερους εικονικούς κόσμους, όπως για παράδειγμα το Active Worlds, στους οποίους τα τρισδιάστατα αντικείμενα κατασκευάζονταν σε επαγγελματικά προγράμματα τρισδιάστατης μοντελοποίησης και ανέβαιναν στη συνέχεια ως αρχείο στον κόσμο, στο Second Life και στο OpenSimulator υπάρχουν ενσωματωμένα εργαλεία κατασκευής περιεχομένου μέσα στον κόσμο. Αυτό επιτρέπει στους χρήστες να δημιουργούν και να επεξεργάζονται συνεργατικά το περιεχόμενο και επιπλέον, επειδή τα εργαλεία αυτά απευθύνονται κυρίως σε αρχάριους χρήστες, δεν απαιτεί εξειδικευμένες γνώσεις τρισδιάστατης μοντελοποίησης. Σε κάθε αντικείμενο του εικονικού κόσμου μπορούν να προστεθούν ένα ή περισσότερα προγράμματα στη γλώσσα σεναρίων LSL ώστε να αποκτήσουν διαδραστική συμπεριφορά.

Τέλος, το OpenSimulator ενσωματώνει και ορισμένα χαρακτηριστικά κοινωνικής δικτύωσης. Οι χρήστες μπορούν να ορίζουν φίλους ή και να δημιουργούν και να διαχειρίζονται ομάδες χρηστών (groups), στις οποίες έχουν τη δυνατότητα να ορίζουν και να διαμορφώνουν τους ρόλους των χρηστών. Οι φίλοι και οι ομάδες μπορούν να έχουν ιδιωτικές συνομιλίες και φόρουμ συζητήσεων, να διαμοιράζονται αντικείμενα κ.λπ.

3 Εγκατάσταση του OpenSimulator

Η εγκατάσταση ενός διακομιστή OpenSimulator είναι μια περίπλοκη διαδικασία, η οποία απαιτεί κατάλληλη προετοιμασία και σχεδιασμό. Στη συνέχεια αναφέρονται ορισμένα βασικά βήματα και αποφάσεις που θα πρέπει

να ληφθούν πριν από την εγκατάσταση του προγράμματος καθώς και μια ενδεικτική διαδικασία εγκατάστασης του διακομιστή και του προγράμματος σύνδεσης. Η διαδικασία αυτή παρουσιάζεται στο βίντεο 7.1.



Σύνδεσμος:

<https://youtu.be/5VBR2U2FXHM>

Video 7.1 Ενδεικτική διαδικασία εγκατάστασης του OpenSimulator.

3.1 Προετοιμασία εγκατάστασης

3.1.1 Υποστήριξη απομακρυσμένων συνδέσεων

Αν θέλουμε ο εικονικός κόσμος να είναι πολυχρηστικός και να μπορεί να δεχτεί απομακρυσμένες συνδέσεις, θα πρέπει η συσκευή στην οποία θα γίνει η εγκατάσταση αφενός να έχει μια γρήγορη σύνδεση στο διαδίκτυο (κατά προτίμηση μόνιμη) και αφετέρου να έχει σταθερή διεύθυνση IP. Αν το δεύτερο δεν είναι εφικτό, για παράδειγμα σύνδεση ADSL μέσω κάποιου παρόχου, τότε θα πρέπει να χρησιμοποιήσουμε κάποια υπηρεσία δυναμικής παροχής διεύθυνσης, όπως η DynDNS, που επιτρέπει σε υπολογιστές με μεταβαλλόμενη IP να μπορούν να εντοπίζονται μέσω σταθερού ονόματος τομέα (domain name) που τους παρέχεται. Επιπλέον, αν ο υπολογιστής μας βρίσκεται πίσω από κάποιο τείχος προστασίας (Firewall), θα πρέπει να ελευθερώσουμε έναν αριθμό από θύρες εισερχόμενων συνδέσεων. Περισσότερες λεπτομέρειες σχετικά με τα παραπάνω παρέχονται στη σελίδα του OpenSimulator.

3.1.2 Διανομή

Όπως αναφέραμε και παραπάνω, υπάρχουν τρεις διαφορετικές διαθέσιμες διανομές του OpenSimulator. Αν θέλουμε να εγκαταστήσουμε προσωρινά το περιβάλλον και να εξερευνήσουμε τις δυνατότητές του, η προτιμότερη λύση είναι το sim-on-a-stick, καθώς η διανομή αυτή είναι προεγκατεστημένη. Ο κόσμος δημιουργείται άμεσα και μπορούμε να συνδεθούμε απευθείας σε αυτόν μέσω εγκατεστημένου προγράμματος πελάτη. Το μειονέκτημα είναι ότι το περιβάλλον μας δεν θα δέχεται απομακρυσμένες συνδέσεις και θα είμαστε ο μοναδικός χρήστης του κόσμου. Από την άλλη μεριά, αν θέλουμε να εγκαταστήσουμε έναν πολυχρηστικό κόσμο με διάρκεια στον χρόνο, είναι προτιμότερο να επιλέξουμε μια εκ των εγκαταστάσεων OpenSim ή Diva/d2. Η δεύτερη συμπεριλαμβάνει κάποιες πρόσθετες δυνατότητες, αλλά υποχρεώνει τον χρήστη να εκτελέσει τον κόσμο σε κατάσταση πλέγματος (grid) και να χρησιμοποιήσει το σύστημα βάσεων δεδομένων MySQL.

3.1.3 Σύστημα βάσεων δεδομένων

Μία ακόμη κρίσιμη επιλογή είναι το σύστημα βάσεων δεδομένων που θα χρησιμοποιηθεί. Για εγκαταστάσεις με προσωρινό χαρακτήρα, η χαμηλών δυνατοτήτων βάση SQLite που προσφέρεται μαζί με τη βασική εγκατάσταση του OpenSimulator επαρκεί. Αν όμως επιθυμούμε ο εικονικός κόσμος να έχει διάρκεια, πλούσιο περιεχόμενο και μεγάλο αριθμό χρηστών, είναι σαφώς προτιμότερη η χρήση της βάσης δεδομένων MySQL. Η τελευταία δεν προσφέρεται μαζί με την εγκατάσταση. Θα πρέπει να κατεβεί και να εγκατασταθεί πριν την πρώτη χρήση του OpenSimulator.

3.1.4 Κατάσταση λειτουργίας

Το επόμενο βήμα είναι η επιλογή κατάστασης λειτουργίας. Η κατάσταση πλέγματος είναι αναγκαία αν θέλουμε ο κόσμος να κατανέμεται σε πολλαπλά απομακρυσμένα υπολογιστικά συστήματα. Κάτι τέτοιο έχει νόημα αν

θέλουμε να εγκαταστήσουμε και να διαχειριζόμαστε κάποιον εικονικό κόσμο με μεγάλο αριθμό περιοχών και χρηστών ή να ενσωματώσουμε στο ίδιο δίκτυο εικονικούς κόσμους πολλαπλών συνεργαζόμενων φορέων ή ατόμων, για παράδειγμα τμημάτων ενός πανεπιστημίου. Από την άλλη μεριά, η αυτόνομη κατάσταση λειτουργίας είναι λιγότερο περίπλοκη στην εγκατάσταση και στη διαχείριση και, ως τέτοια, προτιμότερη αν δεν έχουμε ανάγκη ένα πλέγμα υπολογιστών.

3.2 Ενδεικτική διαδικασία εγκατάστασης

Παρακάτω θα παρουσιάσουμε βήμα προς βήμα την εγκατάσταση ενός διακομιστή OpenSimulator. Για τη συγκεκριμένη εγκατάσταση χρησιμοποιούμε τη βασική διανομή (OpenSim), επιλέγουμε αυτόνομη κατάσταση λειτουργίας (standalone) και χρησιμοποιούμε το σύστημα βάσεων δεδομένων MySQL. Μπορείτε να ακολουθήσετε τη διαδικασία αυτή στο μηχανήμά σας, ανεξαρτήτως του αν έχετε σταθερή διεύθυνση IP ή όχι. Στη δεύτερη περίπτωση, απλά, θα μπορείτε να συνδέεστε μόνο από το ίδιο μηχανήμα, με αποτέλεσμα να μην μπορούν να συνυπάρχουν στον κόσμο απομακρυσμένοι χρήστες.

Βήμα 1. Αρχικά κατεβάζουμε τα προγράμματα εγκατάστασης. Θα μας χρειαστεί η τελευταία έκδοση του OpenSimulator και του MySQL. Τα προγράμματα μπορούν να κατέβουν από τις αντίστοιχες επίσημες σελίδες: <http://opensimulator.org> και <http://dev.mysql.com/downloads/mysql/>. Από τη σελίδα του OpenSimulator επιλέγουμε στα downloads τη μορφή 'Binaries Zip' της τελευταίας έκδοσης και την αποσυμπιέζουμε σε κάποιον φάκελο. Από τη σελίδα του MySQL κατεβάζουμε το MySQL Community Server.

Βήμα 2. Εγκαθιστούμε το σύστημα διαχείρισης βάσεων δεδομένων MySQL. Το εργαλείο που θα χρειαστούμε οπωσδήποτε είναι το MySQL Server, δηλαδή ο βασικός εξυπηρετητής του συστήματος που θα τρέχει ως υπηρεσία (service) στο σύστημά μας. Επιπλέον, είναι χρήσιμο να εγκατασταθεί και το MySQL Workbench, που αποτελεί ένα χρήσιμο γραφικό εργαλείο διαχείρισης των βάσεων. Μετά την επιτυχημένη ολοκλήρωση της εγκατάστασης, θα πρέπει να εκτελείται το MySQL Server ως μόνιμη υπηρεσία στο παρασκήνιο.

Βήμα 3. Στη συνέχεια θα πρέπει να δημιουργήσουμε μια νέα βάση δεδομένων για το OpenSimulator. Τα βήματα είναι τα εξής:

Τρέχουμε το MySQL Client (εκτελείται στη γραμμή εντολών) και δίνουμε το password του διαχειριστή που έχουμε ορίσει κατά την εγκατάσταση. Στη συνέχεια δίνουμε τις παρακάτω εντολές. Μπορείτε να ορίσετε όνομα χρήστη και συνθηματικό της αρεσκείας σας στη θέση των 'opensimuser' και 'opensimpassword':

```
mysql> create database opensim;
mysql> use opensim;
mysql> create user 'opensimuser'@'localhost'
    identified by 'opensimpassword';
mysql> grant all on opensim.* to
    'opensimuser'@'localhost';
mysql> quit
```

Βήμα 4. Ενημερώνουμε τον διακομιστή μας ώστε να χρησιμοποιήσει το MySQL και τη βάση που μόλις δημιουργήσαμε εντός αυτής. Αρχικά εντοπίζουμε το αρχείο 'StandaloneCommon.ini', που βρίσκεται στη διαδρομή 'bin\config-include\' του φακέλου στον οποίο αποσυμπιέσαμε το OpenSim, και το ανοίγουμε για επεξεργασία. Τοποθετούμε την παρακάτω γραμμή σε σχόλια προσθέτοντας ένα ελληνικό ερωτηματικό (;) στην αρχή της:

```
; Include-Storage = "config-include/storage/
    SQLiteStandalone.ini";
```

Απενεργοποιούμε τα σχόλια (αφαιρώντας το ερωτηματικό) στις παρακάτω δύο γραμμές:

```
StorageProvider = "OpenSim.Data.MySQL.dll"
ConnectionString = "Data Source=localhost";
```

```
Database=opensim;UserID=opensimuser;  
Password=opensimpassword;Old Guids=true;"
```

Συμπληρώνουμε τα σωστά ονόματα της βάσης (Database), του χρήστη (User ID) και του συνθηματικού (Password), σύμφωνα με αυτά που δώσαμε στο MySQL για τη δημιουργία της βάσης (βλ. βήμα 3: 'opensim', 'opensimuser' και 'opensimpassword', αντίστοιχα).

Βήμα 5. Εκτελούμε το πρόγραμμα διακομιστή και εισάγουμε τις αρχικές ρυθμίσεις εγκατάστασης. Αρχικά εντοπίζουμε τον φάκελο στον οποίο αποσυμπιέσαμε τα αρχεία και εκτελούμε την εφαρμογή OpenSim.exe που βρίσκεται στον υποφάκελο 'bin\'.
Κατά την εκτέλεσή της η εφαρμογή μας ρωτάει το όνομα της περιοχής που θα δημιουργήσει για εμάς. Μια *περιοχή (region)* είναι ένας χώρος 256 x 256 τ.μ. στον οποίο οι χρήστες μπορούν να κινούνται ελεύθερα και να χτίζουν. Θα πρέπει να δώσουμε ένα μοναδικό όνομα για την περιοχή μας. Π.χ. στην εγκατάστασή μας δίνουμε το όνομα Athens.

```
New region name []: Athens
```

Στη συνέχεια μας ζητείται ένα μοναδικό κλειδί (UUID) που προσδιορίζει την περιοχή μας. Ένα νέο κλειδί έχει δημιουργηθεί για εμάς, οπότε μπορούμε να πατήσουμε Enter για να γίνει αποδεκτό.

```
RegionUUID [a1351254-d5b0-49f9-9a5a-17a52d6b3c06]:
```

Οι επόμενες επιλογές σχετίζονται με την τοποθέτηση της περιοχής μας σε συντεταγμένες κόσμου και τη σύνδεση των χρηστών με αυτήν. Και εδώ είναι προτιμότερο να χρησιμοποιήσουμε τις εξ' ορισμού προτεινόμενες τιμές πατώντας Enter σε καθεμία από τις επιλογές.

```
Region Location [1000,1000]:  
Internal IP address [0.0.0.0]:  
Internal port [9000]:  
Allow alternate ports [False]:
```

Στη συνέχεια μας ζητάει το όνομα του διακομιστή. Αν το μηχάνημά μας έχει σταθερή διεύθυνση IP, αρκεί να τη δηλώσουμε ή να πατήσουμε Enter στην εξ' ορισμού επιλογή (SYSTEMIP). Αν υπάρχει πιθανότητα να αλλάξει IP διεύθυνση στο μέλλον, τότε είναι προτιμότερο να δηλωθεί η ονομασία του μηχανήματος, π.χ. 'opensim.aegean.gr'. Στην περίπτωση που θα τρέχει το πρόγραμμα σύνδεσης στο ίδιο μηχάνημα με τον διακομιστή, μπορούμε επίσης να αφήσουμε την εξ' ορισμού τιμή.

```
External host name [SYSTEMIP]:
```

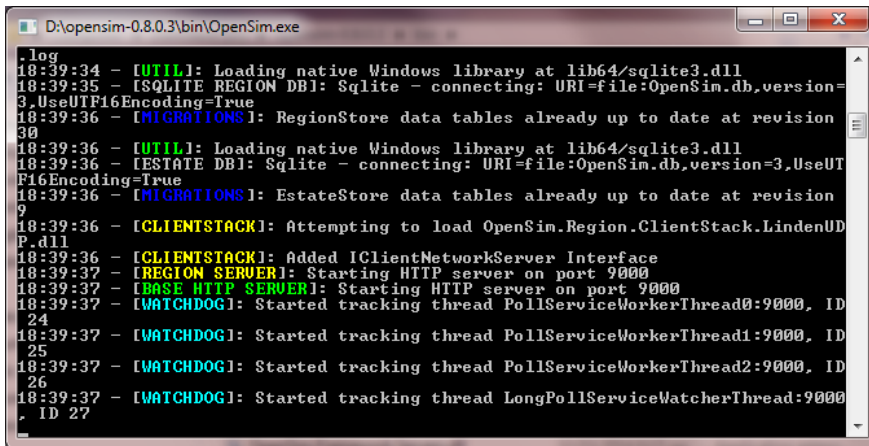
Οι επόμενες επιλογές αφορούν την ιδιοκτησία. Στο OpenSimulator δεν μπορούν να υπάρχουν περιοχές ή τμήματα περιοχών χωρίς ιδιοκτήτη. Ο ιδιοκτήτης είναι υπεύθυνος για τη διαχείριση των δικαιωμάτων πρόσβασης στους χώρους που του ανήκουν. Κατά συνέπεια, εκτός από τις περιοχές (regions) δηλώνονται και οι *ιδιοκτησίες (estates)*. Μία ιδιοκτησία μπορεί να περιλαμβάνει μία ή περισσότερες περιοχές καθώς και τμήματα περιοχών. Κατά την αρχική εγκατάσταση του προγράμματος δημιουργείται μία περιοχή (βλ. παραπάνω) και αντίστοιχα μία νέα ιδιοκτησία η οποία περιλαμβάνει την περιοχή αυτή. Στα επόμενα βήματα δίνουμε το όνομα της ιδιοκτησίας (με Enter κρατάμε το εξ' ορισμού 'My Estate') και τα στοιχεία του ιδιοκτήτη, ο οποίος θα είναι και ο διαχειριστής του κόσμου. Εδώ δηλώνετε όνομα, επώνυμο, συνθηματικό και e-mail. Όπως στις περιοχές, έτσι και στην περίπτωση του διαχειριστή ζητείται επιπλέον ένα μοναδικό κλειδί προσδιορισμού (User ID). Το σύστημα δημιουργεί αυτόματα ένα κλειδί, το οποίο μπορείτε να αποδεχτείτε πατώντας Enter.

```
New estate name [My Estate]:  
Estate owner first name [Test]: Spyros  
Estate owner last name [User]: Vosinakis  
Password: password
```


Email: **spyrosvos@hotmail.com**

User ID [90abadb7-0c75-406c-babf-d3da76d0357a]:

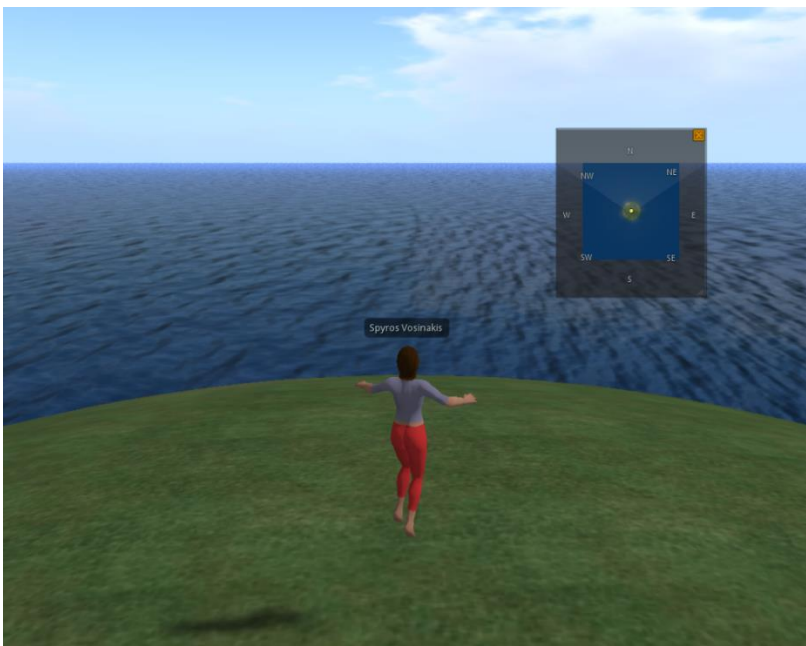
Αν όλα τα παραπάνω βήματα έχουν γίνει σωστά, θα πρέπει να εκτελείται ο διακομιστής του OpenSimulator. Αφήνουμε την κονσόλα ανοιχτή για να μπορεί να δεχτεί συνδέσεις (εικ. 7.4).



```
D:\opensim-0.8.0.3\bin\OpenSim.exe
.log
18:39:34 - [UTIL]: Loading native Windows library at lib64/sqlite3.dll
18:39:35 - [SQLITE REGION DB]: Sqlite - connecting: URI=file:OpenSim.db,version=3,UseUTF16Encoding=True
18:39:36 - [MIGRATIONS]: RegionStore data tables already up to date at revision 30
18:39:36 - [UTIL]: Loading native Windows library at lib64/sqlite3.dll
18:39:36 - [ESTATE DB]: Sqlite - connecting: URI=file:OpenSim.db,version=3,UseUTF16Encoding=True
18:39:36 - [MIGRATIONS]: EstateStore data tables already up to date at revision 9
18:39:36 - [CLIENTSTACK]: Attempting to load OpenSim.Region.ClientStack.LindenUDP.dll
18:39:36 - [CLIENTSTACK]: Added IClientNetworkServer Interface
18:39:37 - [REGION SERVER]: Starting HTTP server on port 9000
18:39:37 - [BASE HTTP SERVER]: Starting HTTP server on port 9000
18:39:37 - [WATCHDOG]: Started tracking thread PollServiceWorkerThread0:9000, ID 24
18:39:37 - [WATCHDOG]: Started tracking thread PollServiceWorkerThread1:9000, ID 25
18:39:37 - [WATCHDOG]: Started tracking thread PollServiceWorkerThread2:9000, ID 26
18:39:37 - [WATCHDOG]: Started tracking thread LongPollServiceWatcherThread:9000, ID 27
```

Εικόνα 7.4 Κονσόλα λειτουργίας του διακομιστή.

Βήμα 6. Στο επόμενο βήμα εγκαθιστούμε ένα πρόγραμμα σύνδεσης και πραγματοποιούμε δοκιμαστική σύνδεση με τον κόσμο μας. Κυκλοφορούν διάφορες εκδόσεις προγραμμάτων πελάτη για σύνδεση στο OpenSimulator, συμπεριλαμβανομένου του επίσημου προγράμματος σύνδεσης στο Second Life, όπως έχουμε ήδη αναφέρει. Σήμερα τα πιο δημοφιλή προγράμματα σύνδεσης είναι τα Singularity και Firestorm. Για μηχανήματα χαμηλότερων δυνατοτήτων μια καλή επιλογή είναι και το Imprudence, το οποίο όμως έχει πάψει να υποστηρίζεται και να εξελίσσεται πλέον. Οι οδηγίες, τα παραδείγματα, οι εικόνες και το πολυμεσικό υλικό αυτού του βιβλίου βασίζονται στο Singularity.



Εικόνα 7.5 Πρώτη σύνδεση στο περιβάλλον OpenSimulator

Κατεβάζουμε και εγκαθιστούμε το Singularity από τη διεύθυνση <http://www.singularityviewer.org> Στη συνέχεια εκτελούμε το πρόγραμμα για να συνδεθούμε στον κόσμο μας.

Αρχικά πατάμε το πλήκτρο Grid Manager, που βρίσκεται στο κάτω μέρος του παράθυρου, για να δημιουργήσουμε μια νέα σύνδεση. Στο παράθυρο που θα ανοίξει πατάμε την επιλογή Create και δίνουμε τα παρακάτω στοιχεία.

- Login URI: η διεύθυνση σύνδεσης στον κόσμο μας. Θα πρέπει να γνωρίζουμε την IP διεύθυνση ή το δικτυακό όνομα του διακομιστή στον οποίο εκτελείται ο κόσμος και να δηλώσουμε σύνδεση στη θύρα 9000 μέσω του πρωτοκόλλου HTTP. Η δήλωση γίνεται ως εξής: `http://<διεύθυνση / όνομα διακομιστή>:9000/` Αν θέλουμε να χρησιμοποιήσουμε τη διεύθυνση του τοπικού μηχανήματος (local host), για παράδειγμα στην περίπτωση που το πρόγραμμα σύνδεσης και ο διακομιστής τρέχουν στο ίδιο μηχάνημα, μπορούμε να δηλώσουμε τη διεύθυνση: `http://127.0.0.1:9000/`
- Platform: επιλέγουμε OpenSim
- Grid Name: δίνουμε ένα όνομα στον κόσμο μας, π.χ. myserver

Πατάμε Ok, και η σύνδεση έχει δημιουργηθεί. Οι παραπάνω ενέργειες χρειάζεται να γίνουν μόνο μία φορά, καθώς η παραπάνω σύνδεση αποθηκεύεται στη λίστα συνδέσεων του προγράμματος.

Για να συνδεθούμε στον κόσμο αρκεί να επιλέξουμε από την αρχική οθόνη στο Grid το όνομα της σύνδεσης που μόλις δηλώσαμε (π.χ. myserver), να γράψουμε στο πεδίο 'Name or Username' το όνομα και το επώνυμο του διαχειριστή που δηλώσαμε στην εγκατάσταση του OpenSimulator (βλ. βήμα 5, 'Spyros Vosinakis'), να προσθέσουμε στο πεδίο Password το συνθηματικό και να πατήσουμε το πλήκτρο Log In.

Αν όλα τα βήματα έχουν γίνει σωστά θα πρέπει να είμαστε σε θέση να βλέπουμε την ενσάρκωσή μας να πετάει πάνω από ένα μικρό νησί (εικ. 7.5).

Βήμα 7. Εγκαθιστούμε την υπηρεσία φωνής. Όπως αναφέραμε και πιο πάνω, προσφέρονται ορισμένες λύσεις ανοιχτού λογισμικού για επικοινωνία μέσω φωνής, αλλά η πιο απλή λύση για μη εμπορική χρήση είναι η υπηρεσία που παρέχεται από την εταιρία Vivot. Για να την ενεργοποιήσουμε, συνδεόμαστε στον παρακάτω σύνδεσμο και αποστέλλουμε μια αίτηση στην εταιρεία: <http://support.vivox.com/opensim/> Στην απάντησή της η εταιρεία μάς δίνει οδηγίες τροποποίησης ενός αρχείου διαμόρφωσης του OpenSimulator και μετά την επανεκκίνηση του διακομιστή η υπηρεσία λειτουργεί κανονικά.

4 Βασικές εργασίες διαχείρισης

Στη συνέχεια θα παρουσιάσουμε έναν αριθμό από εργασίες διαχείρισης του εικονικού κόσμου, οι οποίες είναι συνήθως απαραίτητες για την αρχική διαμόρφωση και συντήρησή του. Οι διεργασίες σχετίζονται με τη λειτουργία του κόσμου, τη διαμόρφωση του εδάφους και των περιοχών, τη διαχείριση των χρηστών και την αποθήκευση και ανάκτηση δεδομένων. Τα παραδείγματα εργασιών διαχείρισης που θα ακολουθήσουμε βασίζονται στην ενδεικτική εγκατάσταση που παρουσιάσαμε στην προηγούμενη υποενότητα.

4.1 Έναρξη και τερματισμός διακομιστή

Ο διακομιστής του OpenSimulator είναι ενεργός όσο εκτελείται το πρόγραμμα κονσόλας OpenSim.exe. Συνεπώς το πρόγραμμα αυτό θα πρέπει να παραμένει ανοιχτό για να λειτουργούν οι υπηρεσίες του εικονικού κόσμου και να δέχεται εισερχόμενες συνδέσεις. Στην περίπτωση που επιθυμούμε να κλείσουμε το πρόγραμμα, θα πρέπει να γράψουμε στην κονσόλα την εντολή shutdown, ώστε να εκτελεστούν όλες οι απαραίτητες διεργασίες τερματισμού του διακομιστή και να μην διακινδυνεύσουμε απώλεια δεδομένων.

Το πρόγραμμα κονσόλας είναι χρήσιμο να ρυθμιστεί να ξεκινάει αυτόματα με την έναρξη του λειτουργικού συστήματος, ώστε να μην διακοπεί για μεγάλο χρονικό διάστημα η λειτουργία του κόσμου σε περίπτωση απρόβλεπτης επανεκκίνησης, για παράδειγμα λόγω διακοπής ρεύματος.

4.2 Διαμόρφωση εδάφους και προσθήκη περιοχών

Μία από τις πρώτες ενέργειες που εκτελεί ένας διαχειριστής πριν από τη διαμόρφωση των περιοχών του κόσμου είναι ο καθορισμός του *αναγλύφου του εδάφους (terrain)*. Όπως αναφέραμε και παραπάνω, μια περιοχή (region)

καταλαμβάνει 256 x 256 τ.μ. Κάθε νέα περιοχή που δημιουργούμε στον κόσμο, άρα και η πρωταρχική περιοχή που δημιουργήσαμε κατά την εγκατάσταση, έχει τη μορφή ενός μικρού νησιού. Αυτό που καθορίζει τη μορφή της περιοχής, δηλαδή τη διαμόρφωση του εδάφους, είναι το ανάγλυφο εδάφους (terrain), το οποίο μπορεί προφανώς να τροποποιηθεί. Υπάρχουν τρεις τρόποι για να γίνει κάτι τέτοιο:

- μέσω απευθείας εντολών στο πρόγραμμα κονσόλας,
- φορτώνοντας ένα έτοιμο αρχείο αναγλύφου,
- τροποποιώντας το ανάγλυφο μέσω ειδικών οπτικών εργαλείων που παρέχονται στο πρόγραμμα πελάτη.

Προς το παρόν θα εξετάσουμε τις δύο πρώτες επιλογές. Στην τρίτη θα αναφερθούμε εκτενέστερα στο Κεφάλαιο 8, όπου θα παρουσιαστούν βασικά εργαλεία και τεχνικές δημιουργίας περιεχομένου στον κόσμο.

Οι δυνατότητες τροποποίησης του αναγλύφου μέσω απευθείας εντολών είναι περιορισμένες. Μπορούμε να διαμορφώσουμε το ανάγλυφο έτσι ώστε να έχει το ίδιο ύψος σε όλη την περιοχή (εντολή terrain fill), να αυξήσουμε συνολικά το ύψος του (terrain elevate), να το μειώσουμε (terrain lower) ή να το πολλαπλασιάσουμε με κάποιον παράγοντα (terrain multiply). Αυτό που θα πρέπει επίσης να γνωρίζουμε είναι ότι το ύψος της θάλασσας είναι εξ ορισμού στα 20 μέτρα. Επομένως, οποιοδήποτε τμήμα του εδάφους έχει ύψος μικρότερο από 20 βρίσκεται κάτω από τη θάλασσα.

Η πιο απλή και ταυτόχρονα χρηστική περιοχή που θα μπορούσαμε να δημιουργήσουμε είναι να έχουμε ένα επίπεδο έδαφος. Για να μετατρέψουμε το μικρό μας νησί σε μια επίπεδη περιοχή, αρκεί να γράψουμε στην κονσόλα την παρακάτω εντολή:

```
terrain fill 21
```

Η εντολή αυτή προσδίδει στο ανάγλυφο του εδάφους ενιαίο ύψος 21 μέτρων, άρα πάνω από το επίπεδο της θάλασσας, σε όλο το εύρος της περιοχής. Μετά την εκτέλεσή της ο κόσμος μας δεν είναι πλέον περιορισμένος σε ένα μικρό νησάκι· μπορούμε να κινηθούμε και να χτίσουμε ελεύθερα σε όλη την περιοχή.

Στη συνέχεια θα δούμε πώς μπορούμε να φορτώσουμε ένα έτοιμο αρχείο αναγλύφου. Για να το κάνουμε αυτό, θα δημιουργήσουμε πρώτα μια δεύτερη περιοχή, την οποία θα τοποθετήσουμε γειτονικά της πρώτης, ώστε να είναι εύκολη η μετακίνηση των χρηστών μεταξύ των περιοχών αυτών.

Ένας διακομιστής OpenSimulator μπορεί θεωρητικά να υποστηρίξει απεριόριστο αριθμό περιοχών. Πρακτικά όμως ο αριθμός αυτός περιορίζεται σημαντικά από τις υπολογιστικές δυνατότητες του συστήματος στο οποίο τρέχει ο διακομιστής. Η συνολική επιβάρυνση του συστήματος επηρεάζεται βεβαίως και από τον αριθμό των αντικειμένων που έχει η κάθε περιοχή καθώς και από τα προγράμματα (scripts) που τρέχουν σε αυτά, αλλά καλό είναι σε κάθε περίπτωση ο αριθμός των περιοχών σας να μην υπερβαίνει τις τέσσερις. Αν επιθυμείτε να έχετε κόσμους με μεγάλο αριθμό περιοχών, κάτι που έχει νόημα αν είναι και ανάλογα μεγάλος ο αριθμός των χρηστών του κόσμου σας, είναι προτιμότερο να επιλέξετε λύση πλέγματος (grid), ώστε να καταναμηθούν οι περιοχές σε πολλαπλούς διακομιστές.

Οι περιοχές ενός κόσμου τοποθετούνται σε ένα δισδιάστατο σύστημα συντεταγμένων, όπου η πρώτη διάσταση αναφέρεται στον άξονα Δύσης–Ανατολής και η δεύτερη στον άξονα Νότου–Βορρά. Η αρχική περιοχή τοποθετείται εξ ορισμού στη θέση (1000,1000). Για να δημιουργήσουμε μία δεύτερη περιοχή, θα χρησιμοποιήσουμε την εντολή create region. Η εντολή αυτή παίρνει ως παραμέτρους το όνομα της νέας περιοχής και το όνομα ενός αρχείου το οποίο θα δημιουργηθεί και θα περιλαμβάνει τα βασικά χαρακτηριστικά της περιοχής. Μπορούμε, για παράδειγμα, να δημιουργήσουμε μια νέα περιοχή βόρεια της αρχικής με την παρακάτω εντολή:

```
create region Piraeus piraeus.ini
```

Η νέα μας περιοχή θα ονομάζεται ‘Piraeus’ και τα στοιχεία της θα αποθηκευτούν στο αρχείο ‘piraeus.ini’. Στη συνέχεια η κονσόλα μάς ρωτάει έναν αριθμό από στοιχεία σχετικά με τη νέα περιοχή μας. Ένα κρίσιμο στοιχείο είναι οι *συντεταγμένες τοποθέτησης (region location)*. Δεν μπορούμε να έχουμε δύο περιοχές στην ίδια θέση, επομένως θα πρέπει να προσέξουμε να δώσουμε τιμή διαφορετική από την αρχική μας περιοχή. Το ίδιο ισχύει και με τη *θύρα σύνδεσης (internal port)*, όπου και πάλι θα πρέπει κάθε περιοχή να έχει τη δική της. Επιλέγουμε να δώσουμε τις συντεταγμένες (1000,1001) και τη θύρα 9001 στη νέα περιοχή μας. Σε όλα τα υπόλοιπα χαρακτηριστικά αποδεχόμαστε τις εξ ορισμού τιμές πατώντας Enter.

```
RegionUUID [68d27311-a532-4336-98a7-756a4e16003a]:  
Region Location [1000,1000]: 1000,1001  
Internal IP address [0.0.0.0]:  
Internal port [9000]: 9001  
Allow alternate ports [False]:  
External host name [SYSTEMIP]:
```

Τέλος, το πρόγραμμα μάς ρωτάει αν η νέα περιοχή θα ανήκει σε υπάρχουσα ιδιοκτησία (Estate) ή αν θα δημιουργηθεί καινούρια. Επιλέγουμε το πρώτο, ορίζοντας ως ιδιοκτησία αυτήν που δημιουργήσαμε κατά την εγκατάσταση του κόσμου.

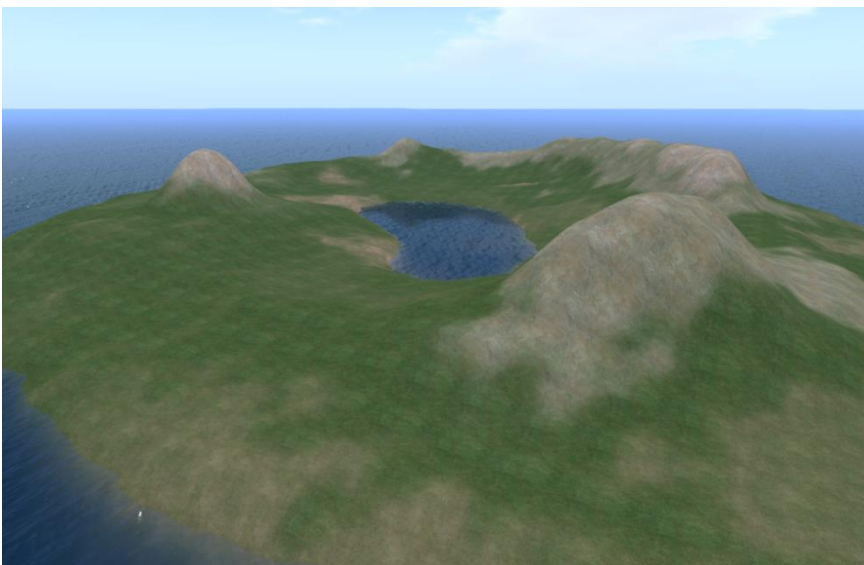
```
Do you wish to join region Piraeus to an existing  
estate (yes/no)? [yes]:  
Name of estate to join. Existing estate names are  
(My Estate) [My Estate]:
```

Αν συνδεθούμε στον κόσμο μας με το πρόγραμμα σύνδεσης, θα παρατηρήσουμε ότι δημιουργήθηκε μια νέα περιοχή, ένα νέο νησάκι όπως το αρχικό, βορειότερα της περιοχής μας.

Στη συνέχεια θα φορτώσουμε ένα έτοιμο αρχείο εδάφους στη νέα μας περιοχή. Υπάρχουν πολλά ελεύθερα αρχεία εδάφους για OpenSimulator που μπορείτε να αναζητήσετε στο διαδίκτυο. Ορισμένα ενδεικτικά αρχεία μπορείτε να βρείτε και στην επίσημη σελίδα του OpenSimulator, στη διεύθυνση: http://opensimulator.org/wiki/Free_Terrains Κατεβάζουμε το αρχείο μας (π.χ. myisland.raw) και το αποθηκεύουμε στον υποφάκελο bin/ του καταλόγου εγκατάστασης του OpenSimulator. Στη συνέχεια γράφουμε στην κονσόλα:

```
change region Piraeus
```

Η εντολή αυτή είναι απαραίτητη για να επισημάνουμε την περιοχή στην οποία αναφερόμαστε. Αν την παραλείψουμε και φορτώσουμε απευθείας το αρχείο εδάφους, θα αλλάξει το έδαφος της αρχικής μας περιοχής (Athens). Μετά την εκτέλεση της εντολής, θα παρατηρήσουμε ότι η προτροπή εντολών (command prompt) της κονσόλας αλλάζει από Region (root) # σε Region (Piraeus) #. Αυτό σημαίνει ότι οι εντολές μας από εδώ και στο εξής θα αναφέρονται μόνο στην περιοχή Piraeus.



Εικόνα 7.6 Ανέβασμα εξωτερικού αρχείου εδάφους στη νέα περιοχή.

Στη συνέχεια φορτώνουμε το αρχείο εδάφους με την παρακάτω εντολή:

```
terrain load myisland.raw
```

Παρατηρούμε ότι το μικρό νησάκι της περιοχής Piraeus αλλάζει σε ένα πιο μεγάλο και ενδιαφέρον νησί (εικ. 7.6).

4.3 Προσθήκη χρηστών

Μέχρι τώρα μοναδικός χρήστης του κόσμου μας είναι αυτός που δηλώσαμε κατά την εγκατάσταση (Spyros Vosinakis). Εφόσον όμως θέλουμε ο κόσμος μας να είναι πραγματικά πολυχρηστικός και να δέχεται εισερχόμενες συνδέσεις, θα χρειαστεί να μπορούμε να δημιουργήσουμε και νέους χρήστες. Υπάρχουν δύο τρόποι να προστεθούν χρήστες στο OpenSimulator: από τον διαχειριστή μέσω της κονσόλας ή από τους ίδιους τους χρήστες μέσω κάποιας Web διεπαφής. Το πλεονέκτημα στη δεύτερη περίπτωση είναι ότι οι χρήστες μπορούν να διαχειριστούν τα συνθηματικά τους καθώς και να επιλέξουν την αρχική μορφή της ενσάρκωσής τους. Συνεπώς, για κόσμους μεγάλης διάρκειας στους οποίους θέλουμε να προστίθενται συχνά νέοι χρήστες είναι προτιμότερη η εγκατάσταση και αξιοποίηση κάποιας Web διεπαφής.

Η προσθήκη ενός νέου χρήστη μέσω κονσόλας γίνεται με την εντολή `create user`. Θα την εισάγουμε για να προσθέσουμε στον κόσμο μας έναν χρήστη-επισκέπτη. Μετά την εκτέλεσή της η κονσόλα μάς ζητάει στοιχεία όπως το ονοματεπώνυμο, το συνθηματικό, το e-mail και το κλειδί (User ID). Το e-mail δεν είναι υποχρεωτικό, ενώ στο κλειδί μπορούμε να αποδεχτούμε την τιμή που έχει δημιουργηθεί πατώντας Enter. Ο διάλογός μας θα είναι ο παρακάτω:

```
Region (Piraeus) # create user  
First name [Default]: Guest  
Last name [User]: User  
Password: Guest  
Email []:  
User ID [fd7f7297-eaf0-4477-9a48-41412ce47b49]:
```

Τώρα μπορούμε να συνδεθούμε με τα στοιχεία του νέου μας χρήστη (Guest User) μέσω του προγράμματος σύνδεσης. Παρατηρούμε ότι η μορφή της ενσάρκωσης είναι ακριβώς η ίδια. Αυτό μπορεί να αλλάξει στη συνέχεια αν τροποποιήσουμε την εμφάνισή μας μέσω της διεπαφής του προγράμματος σύνδεσης. Αν πραγματοποιήσουμε δύο ξεχωριστές συνδέσεις στον κόσμο, μία ως Spyros Vosinakis και μία ως Guest User, θα δούμε ότι ο καθένας από τους δύο χρήστες μπορεί να αντιληφθεί τη θέση και την κίνηση του άλλου, εφόσον είναι στο οπτικό του πεδίο. Οι δύο αυτές συνδέσεις μπορούν να γίνουν και από το ίδιο μηχάνημα, εκτελώντας δύο φορές το Singularity με διαφορετικά στοιχεία σύνδεσης.

4.4 Αποθήκευση δεδομένων και αντίγραφα ασφαλείας

Πολλές φορές είναι χρήσιμη η αποθήκευση και επανάκτηση δεδομένων. Κάτι τέτοιο μπορεί να χρειαστεί για λόγους μεταφοράς περιεχομένου από έναν κόσμο σε έναν άλλο, για δημιουργία αντιγράφων ασφαλείας ή και για την επαναφορά του κόσμου σε προηγούμενη κατάσταση. Υπάρχουν δύο διαφορετικά είδη αρχείων με περιεχόμενα κόσμου που μπορούν να δημιουργηθούν στο OpenSimulator. Τα *αρχεία κόσμου* (*Opensim Archives*) με κατάληξη `.oar` και τα *αρχεία αποθετηρίου* (*Inventory Archives*) με κατάληξη `.iar`.

Στα αρχεία κόσμου αποθηκεύεται όλο το περιεχόμενο μιας περιοχής, δηλαδή το ανάγλυφο του εδάφους και όλα τα αντικείμενα που βρίσκονται σε αυτήν. Αν ο κόσμος μας έχει παραπάνω από μία περιοχές και θέλουμε να αποθηκεύσουμε τα περιεχόμενά του, θα πρέπει να δημιουργήσουμε από ένα αρχείο κόσμου για κάθε περιοχή. Εναλλακτικά, υπάρχει μια παράμετρος (`-all`) στην εντολή αποθήκευσης η οποία προκαλεί τη δημιουργία αρχείου πολλαπλών περιοχών και μπορεί να αποθηκεύσει όλες τις περιοχές του κόσμου σε ένα αρχείο τύπου `.oar`. Αν ανακτηθεί ένα αρχείο κόσμου, τότε διαγράφεται όλο υπάρχον περιεχόμενο της περιοχής και αντικαθίσταται από το αποθηκευμένο. **Προσοχή:** πριν την αποθήκευση ή ανάκτηση αρχείου κόσμου θα πρέπει να βεβαιωθούμε ότι αναφερόμαστε στη σωστή περιοχή (βλ. εντολή `change region`).

Η εντολή αποθήκευσης είναι η `save oar` ακολουθούμενη από το όνομα του αρχείου. Μπορούμε, για παράδειγμα, να δημιουργήσουμε ένα αποθηκευμένο αρχείο κόσμου της περιοχής Piraeus με τις παρακάτω εντολές:

```
change region Piraeus
save oar Piraeus.oar
```

Το αρχείο που θα δημιουργηθεί θα αποθηκευτεί στον υποφάκελο 'bin/' του φακέλου εγκατάστασης του OpenSimulator. Αν θέλετε να αποθηκευτεί σε διαφορετική θέση, μπορείτε να δηλώσετε σχετική ή απόλυτη διαδρομή του αρχείου, π.χ. 'D:\backup_files\Piraeus.oar'

Μπορείτε αντίστοιχα να φορτώσετε ένα αποθηκευμένο αρχείο κόσμου με την εντολή `load oar`. Για παράδειγμα, αν μετά από κάποιες ώρες πειραματισμού με τα εργαλεία κατασκευής του κόσμου θέλετε να επιστρέψει η περιοχή Piraeus στην κατάσταση στην οποία την είχατε αποθηκεύσει, αρκεί να γράψετε στην κονσόλα:

```
load oar Piraeus.oar
```

Στα αρχεία αποθετηρίου αποθηκεύονται αντικείμενα που βρίσκονται στο αποθετήριο του χρήστη διατηρώντας την οργάνωσή τους σε φακέλους και υποφακέλους, αν υπάρχουν. Τα αρχεία αυτά είναι χρήσιμα αν θέλουμε να κρατήσουμε αντίγραφα ασφαλείας από αντικείμενα που έχουμε δημιουργήσει ή να μεταφέρουμε μαζικά αντικείμενα σε άλλους χρήστες. Οι εντολές αποθήκευσης και ανάκτησης είναι `save iar` και `load iar` αντίστοιχα και θα πρέπει να προσδιορίσουμε το ονοματεπώνυμο του χρήστη, τον υποφάκελο του αποθετηρίου που θα αποθηκεύσουμε, το συνθηματικό του χρήστη και το όνομα του αρχείου που θα πρέπει να δημιουργηθεί. Έστω ότι θέλουμε να αποθηκεύσουμε τον φάκελο 'Body Parts' του χρήστη Guest User σε ένα αρχείο. Ο φάκελος αυτός υπάρχει εξ ορισμού σε όλα τα αποθετήρια και περιέχει αντικείμενα που περιγράφουν την ενσάρκωση. Η εντολή που θα δώσουμε είναι η παρακάτω:

```
save iar Guest User "Body Parts" Guest test.iar
```

Θα δημιουργηθεί το αρχείο `test.iar` στον υποφάκελο `bin/` του φακέλου εγκατάστασης. Αντίστοιχα, μπορούμε να μεταφέρουμε τα αποθηκευμένα αντικείμενα ενός αρχείου σε έναν φάκελο αποθετηρίου ενός χρήστη μέσω της εντολής `load iar`. Για παράδειγμα, με την παρακάτω εντολή τα περιεχόμενα του `test.iar` θα μεταφερθούν στο αποθετήριο του χρήστη Guest User στον φάκελο 'saved':

```
load iar Guest User saved Guest test.iar
```

4.5 Ενεργοποίηση και τροποποίηση λειτουργιών

Το OpenSimulator περιλαμβάνει πληθώρα λειτουργιών, οι οποίες μπορούν να ρυθμιστούν και να ενεργοποιηθούν ή να απενεργοποιηθούν ανάλογα με τις ανάγκες του εικονικού κόσμου. Το μεγαλύτερο μέρος των ρυθμίσεων του διακομιστή καθορίζεται από τα περιεχόμενα του αρχείου `Opensim.ini` που βρίσκεται στον υποφάκελο `bin/`. Ορισμένες επιπρόσθετες ρυθμίσεις μπορούν να γίνουν από τον διαχειριστή και εντός του κόσμου μέσω της διεπαφής του προγράμματος σύνδεσης, όπως θα δούμε στη συνέχεια.

Αν εντοπίσουμε και ανοίξουμε το `Opensim.ini` με κάποιον editor απλού κειμένου, θα δούμε ότι υπάρχει ένας πολύ μεγάλος αριθμός ενότητων, καθεμία από τις οποίες περιλαμβάνει χαρακτηριστικά και αντίστοιχες τιμές. Τα αρχεία ρυθμίσεων τύπου `ini` του OpenSimulator ακολουθούν τη συνήθη σύνταξη των αρχείων αυτού του είδους: οι ενότητες δηλώνονται μέσα σε αγκύλες, π.χ. `[Startup]`, και τα χαρακτηριστικά με τις τιμές τους κάτω από την αντίστοιχη ενότητα στη μορφή `όνομα = τιμή`, π.χ. `timeout = 50`. Οι γραμμές που ξεκινούν με ελληνικό ερωτηματικό αντιμετωπίζονται ως σχόλια και δεν λαμβάνονται υπόψη. Μπορείτε να βρείτε στα σχόλια της κάθε ενότητας του `OpenSim.ini` περιγραφές και επεξηγήσεις των λειτουργιών και των διάφορων ρυθμίσεων. Για να ενεργοποιηθούν οι όποιες αλλαγές κάνετε στο αρχείο, απαιτείται τερματισμός και επανεκκίνηση του διακομιστή.

Δύο ρυθμίσεις στις οποίες θα χρειαστεί να προβείτε για την εκτέλεση των παραδειγμάτων και ασκήσεων αυτού του βιβλίου είναι η ενεργοποίηση της γλώσσας OSSL και των ψηφιακών χαρακτήρων (NPCs).

Η OSSL, όπως θα εξηγήσουμε σε επόμενο κεφάλαιο, είναι ένα σύνολο επεκτάσεων της βασικής γλώσσας σεναρίων του OpenSimulator, το οποίο θα μας είναι χρήσιμο σε αρκετές περιπτώσεις προγραμμάτων. Επιπλέον, οι ψηφιακοί χαρακτήρες και οι εφαρμογές τους είναι το βασικό αντικείμενο του ενδέκατου κεφαλαίου αυτού του βιβλίου.

Οι ενέργειες για την ενεργοποίηση της OSSL είναι οι παρακάτω:

1. Μεταβείτε στην ενότητα [XEngine]
2. Απενεργοποιήστε το σχόλιο στην ιδιότητα AllowOSFunctions και αλλάξτε την τιμή σε true:

```
AllowOSFunctions = true
```

3. Αλλάξτε την τιμή της ιδιότητας OSFunctionThreatLevel από VeryLow σε Severe. Με βάση την τιμή αυτής της ιδιότητας καθορίζεται το υποσύνολο εντολών που επιτρέπεται να χρησιμοποιηθούν ανάλογα με τους κινδύνους ασφαλείας. Με την αλλαγή της τιμής σε Severe δεν υπάρχει κανένας περιορισμός, αλλά ο κόσμος είναι πιο ευάλωτος σε κακόβουλες επιθέσεις από τους χρήστες του:

```
OSFunctionThreatLevel = Severe
```

Για την ενεργοποίηση των ψηφιακών χαρακτήρων:

1. Μεταβείτε στην ενότητα [NPC]
2. Απενεργοποιήστε το σχόλιο στην ιδιότητα enabled και αλλάξτε την τιμή σε true:

```
Enabled = true
```

Τέλος, μία λειτουργία η οποία είναι εξαιρετικά χρήσιμη στην περίπτωση πολυχρηστικών κόσμων είναι οι ομάδες χρηστών (Groups). Για να την ενεργοποιήσετε, αρκεί να εντοπίσετε τη σχετική ενότητα ([Groups]), να αφαιρέσετε τα σχόλια και να δηλώσετε τις παρακάτω τιμές στις αντίστοιχες ιδιότητες:

```
Enabled = true
Module = "Groups Module V2"
ServicesConnectorModule = "Groups Local Service Connector"
MessagingModule = "Groups Messaging Module V2"
```

Στη συνέχεια μπορείτε να δημιουργήσετε ή να τροποποιήσετε τις ομάδες χρηστών ενόσω είστε συνδεδεμένοι στον κόσμο μέσω του προγράμματος σύνδεσης.

Περαιτέρω αλλαγές και διαμορφώσεις στον κόσμο μπορούν να γίνουν μέσω του προγράμματος σύνδεσης, εφόσον συνδεθείτε ως διαχειριστής. Οι ρυθμίσεις γίνονται με τη χρήση των σχετικών παραθυρικών διαλόγων από το μενού του προγράμματος. Τα παραδείγματα που θα δώσουμε παρακάτω βασίζονται στις επιλογές και στα μενού του Singularity, αλλά η πλειοψηφία των εναλλακτικών προγραμμάτων σύνδεσης περιλαμβάνει τις ίδιες ή παραπλήσιες επιλογές.

Η πρώτη κατηγορία επιλογών που θα εξετάσουμε σχετίζεται με τη διαμόρφωση της ιδιοκτησίας σας (Estate). Από το κεντρικό μενού του Singularity επιλέξτε World > Region Estate... Στο παράθυρο διαλόγου που θα εμφανιστεί επιλέξτε την καρτέλα "Estate". Εδώ μπορείτε να διαμορφώσετε έναν αριθμό από επιλογές, όπως:

- αν θα μπορεί οποιοσδήποτε χρήστης να την επισκεφτεί (επιλογή "Allow Public Access"),
- αν θα υποστηρίζεται φωνητική επικοινωνία (επιλογή "Allow Voice Chat"), με την προϋπόθεση να έχετε εγκαταστήσει υπηρεσία φωνής στον κόσμο (βλ. 3.2 βήμα 7),
- αν θα μπορούν οι χρήστες να τηλεμεταφέρονται στην περιοχή (επιλογή "Allow Direct Teleport") και
- αν ο ήλιος θα έχει σταθερή θέση ή θα μεταβάλλεται (επιλογή "Fixed Sun"). Στη δεύτερη περίπτωση, ο κόσμος θα ακολουθεί έναν κύκλο ημέρας-νύχτας διάρκειας τεσσάρων ωρών.

Ορισμένες επιπλέον επιλογές σχετικά με τα δικαιώματα των χρηστών στην περιοχή μας μπορούμε να βρούμε στο μενού World > About Land... επιλέγοντας την καρτέλα "Options". Στον σχετικό διάλογο μπορούμε να επιλέξουμε μεταξύ άλλων αν οι επισκέπτες μας θα επιτρέπεται:

- να αλλάζουν τη μορφή του εδάφους (Edit Terrain),
- να πετούν (Fly),
- να δημιουργούν αντικείμενα (Create Objects) και
- να γράφουν και να εκτελούν κώδικα (Run Scripts).

Τέλος, μπορείτε να δημιουργήσετε και να διαχειριστείτε ομάδες χρηστών μέσα από το μενού Edit > Groups του προγράμματος, υπό την προϋπόθεση ότι έχετε ενεργοποιήσει τη σχετική υπηρεσία. Με την επιλογή Create μπορείτε να δημιουργήσετε μία νέα ομάδα, να της δώσετε ονομασία και να επιλέξετε αν οι υπόλοιποι χρήστες θα μπορούν να γραφτούν ελεύθερα σε αυτήν (open enrollment) ή κατόπιν πρόσκλησης. Επιπλέον, μπορείτε στην καρτέλα Members & Roles να ρυθμίσετε τα δικαιώματα πρόσβασης των μελών της ομάδας στις διάφορες λειτουργίες της ομάδας ή και να ορίσετε νέους ρόλους. Οι ομάδες χρηστών είναι ιδιαίτερα χρήσιμες σε συνεργατικά περιβάλλοντα, γιατί επιτρέπουν την από κοινού πρόσβαση και επεξεργασία αντικειμένων και τμημάτων κώδικα του κόσμου.

Είναι προφανές ότι οι αποφάσεις που θα πάρουμε για όλες τις παραπάνω επιλογές εξαρτώνται από το πλαίσιο χρήσης του κόσμου. Αν οι χρήστες μας έχουν περισσότερο τον ρόλο του επισκέπτη ενός προκατασκευασμένου κόσμου τον οποίο καλούνται να εξερευνήσουν και να αλληλεπιδράσουν μαζί του, είναι προτιμότερο να μην τους δίνεται μεγάλη ελευθερία ενεργειών (π.χ. αλλαγή μορφής εδάφους και δημιουργία αντικειμένων). Σε αντίθετη περίπτωση, ο κόσμος μας κινδυνεύει λόγω απειρίας των χρηστών να υποστεί αλλαγές που θα τον καταστήσουν δυσλειτουργικό. Από την άλλη μεριά, αν οι χρήστες είναι συν-δημιουργοί του περιεχομένου και δεν είναι πιθανή κάποια κακόβουλη ενέργεια από μέρους τους, είναι απαραίτητο να έχουν πρόσβαση σε όσο το δυνατόν περισσότερα εργαλεία και ενέργειες του κόσμου.

5 Οδηγίες χρήσης

Η χρήση ενός εικονικού κόσμου με πολλαπλές δυνατότητες όπως το OpenSimulator δεν είναι το ίδιο διαισθητική όπως τα κοινά παραθυρικά προγράμματα. Οι δυσκολίες είναι αναμενόμενες τόσο λόγω του διαφορετικού μοντέλου αναπαράστασης και παρουσίασης του περιβάλλοντος όσο και λόγω των σαφώς πιο περίπλοκων μορφών αλληλεπίδρασης. Κατά συνέπεια, είναι χρήσιμο οι αρχάριοι χρήστες να εξοικειωθούν πρώτα με το περιβάλλον και τις δυνατότητές του πριν από την αξιοποίησή του στα πλαίσια κάποιας εφαρμογής.

Ακολουθούν συνοπτικές οδηγίες χρήσης του Singularity για την πραγματοποίηση βασικών ενεργειών στον κόσμο. Οι λειτουργίες που θα εξετάσουμε υποστηρίζονται από όλα τα προγράμματα σύνδεσης, απλά ενδέχεται να υπάρχουν διαφορές μεταξύ τους σε επίπεδο διεπαφής. Η βασική χρήση του κόσμου παρουσιάζεται στο βίντεο 7.2.



Σύνδεσμος:
<https://youtu.be/xlAdDLsmBgs>

Video 7.2 Βασικές ενέργειες χρήση σε κόσμους του OpenSimulator.

5.1 Σύνδεση στον κόσμο

Για να συνδεθεί ένας χρήστης σε εικονικό κόσμο που τρέχει σε διακομιστή OpenSimulator απαιτείται να έχει εγκατεστημένο το πρόγραμμα σύνδεσης, να γνωρίζει τη διεύθυνση του διακομιστή και να έχει δημιουργηθεί ένας λογαριασμός χρήστη γι' αυτόν.

Η διεύθυνση του διακομιστή είναι η διεύθυνση IP του υπολογιστή στον οποίο τρέχει το OpenSimulator ακολουθούμενη από θύρα σύνδεσης, η οποία εξ ορισμού είναι η 9000. Επομένως, αν έχουμε εγκαταστήσει έναν διακομιστή σε μηχανήμα με διεύθυνση IP την 195.251.170.25 και δεν έχουμε τροποποιήσει τη θύρα σύνδεσης, η διεύθυνση σύνδεσης στον κόσμο μας θα είναι η: <http://195.251.170.25:9000>. Αν συνδεόμαστε από τον ίδιο υπολογιστή στον οποίο τρέχει ο διακομιστής, μπορούμε να δηλώσουμε ως διεύθυνση σύνδεσης το τοπικό μηχανήμα (localhost), δηλαδή: <http://127.0.0.1:9000>

Ο χρήστης θα πρέπει την πρώτη φορά που θα εκτελέσει το πρόγραμμα σύνδεσης να δηλώσει τη διεύθυνση του διακομιστή. Αυτό από το Singularity γίνεται μέσω της επιλογής Grid Manager και Create (βλ. ενότητα 3.2, βήμα 6). Στη συνέχεια, για να συνδεθεί στον κόσμο, θα πρέπει να επιλέξει τον κόσμο (Grid) που δήλωσε, να συμπληρώσει τα στοιχεία του λογαριασμού του στα πεδία Name και Password και να πατήσει το πλήκτρο 'Log In'. Αν όλα τα βήματα έχουν γίνει σωστά, θα πρέπει να βλέπει την ενσάρκωσή

5.2 Πλοήγηση

Η πλοήγηση είναι η βασικότερη μορφή αλληλεπίδρασης στον εικονικό κόσμο, καθώς επιτρέπει τη μετακίνηση της ενσάρκωσης και την αλλαγή οπτικού πεδίου, που είναι απαραίτητη σε τρισδιάστατα περιβάλλοντα. Η βασική μεταφορά πλοήγησης που χρησιμοποιείται στο OpenSimulator, όπως και στην πλειοψηφία των εικονικών κόσμων, είναι ο άμεσος έλεγχος της ενσάρκωσης με οπτική τρίτου προσώπου.

Η πιο άμεση και διαισθητική τεχνική έλεγχου του βαδίσματος της ενσάρκωσης είναι με τη χρήση των πλήκτρων διεύθυνσης (arrow keys). Με τα πλήκτρα πάνω και κάτω η ενσάρκωση κινείται προς τα μπρος ή προς τα πίσω αντίστοιχα, ενώ με τα αριστερά και δεξιά περιστρέφεται προς την αντίστοιχη διεύθυνση. Η ενσάρκωση μπορεί να εκτελέσει και πλάγιο βάδισμα κρατώντας πατημένο το Shift και το αριστερό ή δεξί πλήκτρο διεύθυνσης. Επιπλέον, αν πατηθεί το πάνω πλήκτρο διεύθυνσης γρήγορα δύο φορές και κρατηθεί πατημένο, η ενσάρκωση τρέχει αντί να βαδίζει, αυξάνοντας σημαντικά την ταχύτητα μετακίνησης.

Εκτός από τη μετακίνηση στο έδαφος, η ενσάρκωσή μας έχει και τη δυνατότητα να πετάξει! Το πέταγμα ενεργοποιείται με το πλήκτρο Page Up. Όσο παραμένει το πλήκτρο πατημένο, ο χαρακτήρας κινείται προς τα πάνω. Αν το αφήσουμε, ο χαρακτήρας θα παραμείνει αιωρούμενος και με τα πλήκτρα βαδίσματος θα πετάει πλέον οριζόντια πάνω από το έδαφος. Αντίστοιχα, ο χαρακτήρας μπορεί να κατέβει χαμηλότερα με το πλήκτρο Page Down. Αν αγγίξει το έδαφος, θα επιστρέψει σε κατάσταση βαδίσματος. Αν πατήσουμε στιγμιαία τα πλήκτρα Page Up και Page Down όσο ο χαρακτήρας μας βρίσκεται στο έδαφος, τότε αυτός θα κάνει άλμα ή θα σκύψει αντίστοιχα.

Στο περιβάλλον μάς δίνεται η δυνατότητα να μεταβάλλουμε τη θέση της κάμερας σε σχέση με την ενσάρκωσή μας ή ακόμη και να αλλάξουμε σε οπτική πρώτου προσώπου. Με κύλιση της ρόδας προς τα κάτω η κάμερα απομακρύνεται από την ενσάρκωσή μας, ενώ με κύλιση προς τα πάνω την πλησιάζει. Αν η κάμερα πλησιάσει πολύ την ενσάρκωση, τότε η οπτική αλλάζει σε πρώτου προσώπου και βλέπουμε μέσα από τα μάτια της ενσάρκωσης. Στην κατάσταση αυτήν, η πλοήγηση ακολουθεί την τεχνική του mouse look (βλ. Κεφ. 4), δηλαδή η κίνηση του ποντικιού τροποποιεί ανάλογα την εστίαση και τη διεύθυνση μετακίνησης. Μπορούμε να επιστρέψουμε στην αρχική κατάσταση (τρίτου προσώπου) πατώντας το πλήκτρο ESC.

Ο πίνακας 7.1 παρουσιάζει συνοπτικά τα πλήκτρα πλοήγησης και τις αντίστοιχες ενέργειες της ενσάρκωσης.

Πλήκτρα	Λειτουργία
πάνω/κάτω	βάδισμα εμπρός/πίσω
αριστερά/δεξιά	περιστροφή αριστερά/δεξιά
διπλό πάτημα πλήκτρου πάνω	τρέξιμο
Shift + αριστερά/δεξιά	πλάγια κίνηση αριστερά/δεξιά
Page Up	(στο έδαφος στιγμιαία) άλμα, πέταγμα ψηλότερα
Page Down	(στο έδαφος) σκύψιμο, πέταγμα χαμηλότερα
κύλιση ρόδας πάνω/κάτω	εστίαση στον χαρακτήρα (mouselook)/ απομάκρυνση από τον χαρακτήρα
ESC	επιστροφή σε κατάσταση τρίτου προσώπου

Πίνακας 7.1 Πλήκτρα πλοήγησης και αντίστοιχες λειτουργίες.

5.3 Εξερεύνηση

Η πλοήγηση μέσω ενσάρκωσης, αν και αρκετά φυσική και διαισθητική, δεν είναι η καταλληλότερη τεχνική σε όλες τις περιπτώσεις. Πολλές φορές μπορεί να χρειαστεί να παρατηρήσουμε έναν χώρο, ένα αντικείμενο ή έναν χρήστη από πολλές διαφορετικές οπτικές γωνίες. Μια τέτοια ανάγκη μπορεί να προκύψει είτε κατά τη διάρκεια της εξερεύνησης του περιβάλλοντος για λόγους προσεκτικότερης παρατήρησης και καλύτερης κατανόησης του περιεχομένου είτε κατά τη διάρκεια ενεργειών δημιουργίας νέου περιεχομένου για την ταχύτερη και ακριβέστερη προσαρμογή και τοποθέτηση των στοιχείων στο περιβάλλον.

Αν κρατήσουμε πατημένο το πλήκτρο ALT, θα παρατηρήσουμε ότι ο δείκτης του ποντικιού μας αλλάζει σε μεγεθυντικό φακό. Αυτή η αλλαγή καταδεικνύει ότι είμαστε σε κατάσταση εξερεύνησης. Αν κάνουμε κλικ (κρατώντας πατημένο το ALT) σε ένα αντικείμενο του κόσμου, θα παρατηρήσουμε ότι με την κίνηση του ποντικιού μπορούμε να το εξερευνήσουμε. Μετακινώντας το ποντίκι προς τα πάνω ή προς τα κάτω πλησιάζουμε ή απομακρυνόμαστε από το αντικείμενο αντίστοιχα. Με την οριζόντια μετακίνηση του ποντικιού αλλάζει η οπτική γωνία από την οποία το παρατηρούμε· επί της ουσίας, κινούμαστε σε έναν νοητό οριζόντιο δακτύλιο γύρω από το αντικείμενο. Κατά συνέπεια, κινώντας το ποντίκι προς τα δεξιά παρατηρούμε το αντικείμενο από δεξιότερη οπτική γωνία πάνω στον δακτύλιο και αντίστοιχα προς τα αριστερά. Αν θέλουμε να παρατηρήσουμε το αντικείμενο από πιο ψηλά ή από πιο χαμηλά, θα πρέπει μαζί με το ALT να κρατήσουμε πατημένο και το πλήκτρο CONTROL. Στην περίπτωση αυτήν, η κίνηση προς τα πάνω ή προς τα κάτω του ποντικιού μεταφράζεται σε αντίστοιχη κίνηση σε έναν κάθετο δακτύλιο με κέντρο το αντικείμενο. Έτσι, μετακινώντας το ποντίκι προς τα πάνω ή προς τα κάτω βλέπουμε το αντικείμενο από πιο ψηλή ή από πιο χαμηλή θέση αντίστοιχα.

Ο πίνακας 7.2 παρουσιάζει συνοπτικά τα πλήκτρα εξερεύνησης και τις αντίστοιχες ενέργειες που προκαλούν.

Πλήκτρα	Λειτουργία
ALT + κλικ + κίνηση πάνω/κάτω	εστίαση στο αντικείμενο/απομάκρυνση
ALT+ κλικ + κίνηση αριστερά/δεξιά	περιστροφή αριστερά/δεξιά γύρω από το αντικείμενο
ALT+ CONTROL + κλικ + κίνηση πάνω/κάτω	περιστροφή πάνω/κάτω γύρω από το αντικείμενο

Πίνακας 7.2 Πλήκτρα εξερεύνησης και αντίστοιχες λειτουργίες.

Δοκιμάστε να πειραματιστείτε με τη χρήση των πλήκτρων εξερεύνησης και να εξασκηθείτε στη χρήση τους, διότι θα μας είναι απαραίτητα στη διαδικασία κατασκευής νέου περιεχομένου.

5.4 Επικοινωνία

Ο εικονικός κόσμος μάς δίνει τη δυνατότητα επικοινωνίας με χρήστες που βρίσκονται σε σχετικά κοντινή απόσταση μέσω κειμένου (chat) ή φωνής. Για το δεύτερο, θα πρέπει να έχει ενεργοποιηθεί η υπηρεσία φωνής στον διακομιστή (βλ. ενότητα 3.2, βήμα 7).

Για να στείλετε ένα γραπτό μήνυμα θα πρέπει να εμφανίσετε το πεδίο εισαγωγής κειμένου πατώντας το συννεφάκι στην κάτω αριστερή γωνία της οθόνης του Singularity και στη συνέχεια να γράψετε το κείμενό σας. Μόλις πατήσετε Enter, το κείμενο θα αποσταλεί στον κόσμο και οι γειτονικοί χρήστες θα μπορούν να το διαβάσουν. Αν θέλετε το κείμενό σας να διαβαστεί σε μεγαλύτερη ακτίνα, μπορείτε να αλλάξετε το πλήκτρο Say δίπλα στο πεδίο εισαγωγής σε Shout. Αντίστοιχα, αν το αλλάξετε σε Whisper, θα φτάσει μόνο στους πολύ κοντινούς χρήστες.

Όταν κάποιος χρήστης στέλνει μήνυμα κειμένου, αυτό μπορεί να διαβαστεί για περιορισμένο χρονικό διάστημα στην οθόνη του προγράμματος σύνδεσης. Αν θέλουμε να έχουμε πρόσβαση και σε προηγούμενα μηνύματα ώστε να παρακολουθήσουμε προσεκτικότερα τους διαλόγους μεταξύ των χρηστών, μπορούμε με το πλήκτρο Local Chat να δούμε σε ένα παράθυρο όλα τα προηγούμενα μηνύματα που έχει «ακούσει» η ενσάρκωσή μας.



Εικόνα 7.7 Οπτικοποίηση ηχητικής εκπομπής κατά τη διάρκεια ομιλίας.

Τέλος, μπορούμε να συνομιλήσουμε μέσω φωνής αν κρατήσουμε πατημένο το μεσαίο πλήκτρο (ρόδα) του ποντικιού μας. Όσο το πλήκτρο είναι πατημένο, η φωνή μας μεταδίδεται μέσω του εικονικού κόσμου στους γειτονικούς χρήστες και σχηματίζεται μια σχετική οπτικοποίηση ηχητικής εκπομπής πάνω στην ενσάρκωσή μας (εικ. 7.7). Μπορούμε να αλλάξουμε τις ρυθμίσεις ομιλίας (π.χ. επιλογή μικροφώνου, ένταση κ.λπ.) από το μενού του Singularity επιλέγοντας Edit > Preferences > καρτέλα Voice Chat.

5.5 Αλληλεπίδραση με αντικείμενα

Η συμπεριφορά των αντικειμένων σε έναν εικονικό κόσμο βασισμένο στο OpenSimulator καθορίζεται από το πρόγραμμα ή τα προγράμματα (scripts) που αυτά εκτελούν. Όπως θα δούμε σε επόμενα κεφάλαια, τα αντικείμενα του κόσμου μπορούν να περιέχουν κώδικα ο οποίος καθορίζει την απόκρισή τους σε ερεθίσματα του περιβάλλοντος. Έτσι, είναι πιθανό ένα αντικείμενο να ενεργοποιήσει κάποια συμπεριφορά επειδή κάποιος χρήστης το πλησίασε ή συγκρούστηκε μαζί του. Υπάρχουν όμως και περιπτώσεις στις οποίες η συμπεριφορά των αντικειμένων ενεργοποιείται κατόπιν επιλογής του ίδιου του χρήστη. Τότε ο δείκτης του ποντικιού αλλάζει ανάλογα με την ενέργεια που μπορεί να κάνει ο χρήστης σε αυτό. Οι δυνατές ενέργειες είναι:

- απλό κλικ πάνω στο αντικείμενο για την ενεργοποίηση κάποιας συμπεριφοράς,
- άσκηση δύναμης πάνω σε φυσικό αντικείμενο με κρατημένο κλικ και μετακίνηση του ποντικιού στην επιθυμητή διεύθυνση,
- κάθισμα πάνω στο αντικείμενο,
- άνοιγμα των περιεχομένων του αντικειμένου,
- αναπαραγωγή πολυμέσων (ήχος, ιστοσελίδα ή βίντεο).

Ενώσω πλοηγούμαστε στον εικονικό κόσμο μπορούμε να διαπιστώσουμε αν ένα αντικείμενο έχει διαδραστική συμπεριφορά, για παράδειγμα αν μπορούμε να ανοίξουμε μια πόρτα, φέρνοντας τον δείκτη του ποντικιού πάνω του και παρατηρώντας αν αλλάζει μορφή καταδεικνύοντας κάποια από τις παραπάνω ενέργειες.

5.6 Αποθετήριο

Κάθε χρήστης του εικονικού κόσμου έχει το δικό του αποθετήριο (Inventory), το οποίο περιέχει τα αντικείμενα που του ανήκουν. Μπορείτε να δείτε τα περιεχόμενα του αποθετηρίου σας κάνοντας κλικ στο πλήκτρο “Inventory” στην κάτω δεξιά γωνία της οθόνης. Θα εμφανιστεί ένα παράθυρο που δείχνει σε μορφή δέντρου διάφορους φακέλους, και αντικείμενα μέσα σε αυτούς, (βλ. εικ. 7.8). Τα αντικείμενα είναι οργανωμένα σε

φακέλους ανάλογα με το είδος τους, ενώ κάθε είδος έχει και το δικό του εικονίδιο. Υπάρχει ένας φάκελος-ρίζα που ονομάζεται My Inventory και όλοι οι υπόλοιποι είναι υποφάκελοι αυτού. Τα σημαντικότερα είδη αντικειμένων που μπορούν να αποθηκευτούν στο αποθετήριο είναι:

- γεωμετρικά αντικείμενα (objects),
- εικόνες και υφές (textures),
- σημειώματα κειμένου (notecards),
- τμήματα σώματος της ενσάρκωσης (body parts),
- ρούχα (clothes),
- συνθετικές κινήσεις (animations),
- ενδιαφέρουσες τοποθεσίες (landmarks).

Στο αποθετήριο μπορούμε να διαχειριστούμε τους φακέλους και τα αντικείμενά μας με τρόπο ανάλογο των κλασικών παραθυρικών λειτουργικών συστημάτων. Κάνοντας δεξί κλικ πάνω σε έναν φάκελο, μπορούμε να επιλέξουμε 'New Folder' και να δημιουργήσουμε έναν νέο υποφάκελο. Μπορούμε επίσης μέσα από τα προσφερόμενα μενού να μετονομάζουμε και να διαγράψουμε αντικείμενα και φακέλους, να κάνουμε αντιγραφή, επικόλληση και αποκοπή στοιχείων κ.λπ.



Εικόνα 7.8 Ενδεικτικό αποθετήριο χρήστη.

Στη συνέχεια θα εξετάσουμε τη δημιουργία και διαχείριση τριών βασικών κατηγοριών αντικειμένων, τα οποία θα μας χρησιμεύσουν στα επόμενα κεφάλαια. Αυτά είναι τα σημειώματα, οι εικόνες και τα γεωμετρικά αντικείμενα.

5.6.1 Σημειώματα (notecards)

Τα σημειώματα είναι χρήσιμα για να προσθέτουμε σημειώσεις σε αντικείμενα του κόσμου. Μπορείτε να δημιουργήσετε ένα νέο σημείωμα αν κάνετε δεξί κλικ σε κάποιον φάκελο του αποθετηρίου και πατήσετε 'New Note'. Τότε θα δημιουργηθεί ένα νέο αντικείμενο τύπου Notecard στον φάκελο που επιλέξατε και θα έχει το προσωρινό όνομα 'New Note'. Μπορείτε απευθείας να διορθώσετε το όνομά του ή να το αλλάξετε αργότερα με δεξί πλήκτρο πάνω του και 'Rename'. Παράλληλα θα δείτε να εμφανίζεται ένα παράθυρο επεξεργασίας κειμένου, στο οποίο μπορείτε να γράψετε το σημειώμά σας. Μόλις ολοκληρώσετε το κείμενο, πατήστε το πλήκτρο 'Save' για να αποθηκευτεί.

Μπορείτε να επεξεργαστείτε οποιοδήποτε σημείωμα του αποθετηρίου με διπλό κλικ πάνω του. Θα ανοίξει το παράθυρο επεξεργασίας κειμένου με τα περιεχόμενα του σημειώματος και, σε περίπτωση που κάνετε αλλαγές, μπορείτε να αποθηκεύσετε το επεξεργασμένο σημείωμα πατώντας το πλήκτρο ‘Save’.

5.6.2 Εικόνες (Textures)

Μπορείτε να ανεβάσετε μια εικόνα από τον υπολογιστή σας μέσω της επιλογής από το μενού File > Upload Image (free). Διάφορα είδη εικόνας αναγνωρίζονται από το περιβάλλον, όπως JPEG αλλά και PNG. Στην περίπτωση που θέλετε η εικόνα σας να έχει και διαφάνεια ανεβάστε τη σε μορφή PNG και διασφαλίστε ότι έχετε ορίσει την κατάλληλη περιοχή διαφάνειας σε πρόγραμμα επεξεργασίας εικόνας (π.χ. Photoshop). Αφού επιλέξετε εικόνα, το πρόγραμμα σας ζητάει ένα όνομα και μία περιγραφή, και στη συνέχεια με το πλήκτρο “Upload” την ανεβάζει στον διακομιστή. Μόλις η εικόνα ανέβει, προστίθεται αυτόματα στο αποθετήριό σας στον φάκελο “Textures”. Με διπλό κλικ σε μια εικόνα μπορείτε να τη δείτε και να τροποποιήσετε την περιγραφή της. Επίσης, με την επιλογή “Rename” μπορείτε να τη μετονομάσετε. Αν θέλετε να ανεβάσετε πολλές εικόνες μαζί, μπορείτε να το κάνετε με την επιλογή File > Bulk Upload (free per file). Οι διαστάσεις των εικόνων που ανεβάζετε τροποποιούνται έτσι ώστε ο αριθμός των εικονοστοιχείων της κάθε διάστασης να είναι δύναμη του δύο (π.χ. 512 x 512). Για λόγους καλύτερης απόδοσης καλό είναι να μην ανεβάζετε εικόνες μεγάλων διαστάσεων στον εικονικό κόσμο.

5.6.3 Αντικείμενα (Objects)

Αν κάνετε δεξί κλικ σε ένα αντικείμενο που υπάρχει στον κόσμο, μία από τις επιλογές είναι η ‘Take’. Με την επιλογή αυτή το αντικείμενο εξαφανίζεται από τον κόσμο και μεταφέρεται στο αποθετήριό σας στον φάκελο ‘Objects’. Αν θέλετε το αντικείμενο να παραμείνει στον κόσμο, προσφέρεται η εναλλακτική επιλογή ‘Take Copy’. Για να πάρετε ένα αντικείμενο θα πρέπει είτε να είστε ο ιδιοκτήτης του, δηλαδή να το έχετε ανεβάσει ή κατασκευάσει εσείς, είτε να έχετε δικαιώματα να το πάρετε, δηλαδή ο κατασκευαστής του να επιτρέπει σε άλλους χρήστες να πάρουν αντίγραφο του αντικειμένου.

Αν θέλετε να μεταφέρετε ένα αντικείμενο που υπάρχει στο αποθετήριό σας στον εικονικό κόσμο, μπορείτε να σύρετε (drag) το εικονίδιο του στο περιβάλλον. Τότε θα δημιουργηθεί ένα αντίγραφο του αντικειμένου στο περιβάλλον, αλλά το αντικείμενο θα εξακολουθεί να υπάρχει και στο αποθετήριό σας.

Τέλος, μπορείτε να ανεβάσετε αντικείμενα στον κόσμο μέσω των επιλογών: File > Import Object XML και File > Upload Mesh. Η πρώτη περίπτωση αναφέρεται σε αντικείμενα που έχουν δημιουργηθεί σε κάποιον εικονικό κόσμο βασισμένο στο Second Life ή στο OpenSimulator και έχουν αποθηκευτεί σε μορφή XML. Η δεύτερη σε πολυγωνικά αντικείμενα που έχουν κατασκευαστεί σε εξωτερικά προγράμματα τρισδιάστατης μοντελοποίησης. Θα αναφερθούμε με μεγαλύτερη λεπτομέρεια στην κατασκευή και εισαγωγή αντικειμένων στο επόμενο κεφάλαιο.

5.7 Διαμόρφωση ενσάρκωσης

Αν θέλετε να διαμορφώσετε την όψη και τα ρούχα της ενσάρκωσής σας, μπορείτε να το κάνετε με δεξί κλικ πάνω στην ενσάρκωση και επιλογή ‘Appearance’. Θα εμφανιστεί ένα παράθυρο με τις διάφορες επιλογές τροποποίησης της ενσάρκωσης (εικ. 7.9). Μπορείτε να παρατηρήσετε ότι στο αριστερό μέρος του παραθύρου υπάρχει ένας αριθμός από καρτέλες. Οι τέσσερις πρώτες αναφέρονται σε μέρη του σώματος (Body Parts) της ενσάρκωσης και οι υπόλοιπες σε ρούχα. Τα μέρη του σώματος που μπορείτε να επιλέξετε και να τροποποιήσετε είναι το σώμα (Shape), το δέρμα (Skin), τα μαλλιά (Hair) και τα μάτια (Eyes). Για καθένα από τα παραπάνω παρέχονται διάφορες υπο-επιλογές και ρυθμίσεις που διαμορφώνουν το φύλο, το σώμα, το πρόσωπο και τη γενικότερη εμφάνιση του χαρακτήρα σας. Αντίστοιχα, μπορείτε να δημιουργήσετε και να τροποποιήσετε διαφόρων ειδών ρούχα, όπως μπλούζες, παντελόνια, παπούτσια κ.λπ. Κατά τη διάρκεια της επεξεργασίας της εμφάνισης, η άποψη του εικονικού κόσμου εστιάζει στην μπροστινή όψη της ενσάρκωσής σας, ώστε να μπορείτε να παρατηρήσετε άμεσα το αποτέλεσμα των ενεργειών σας.



Εικόνα 7.9 Διαμόρφωση εμφάνισης της ενσάρκωσης.

Οι αλλαγές που κάνετε στο σώμα και στα ρούχα της ενσάρκωσής σας αποθηκεύονται στο αποθετήριό σας με το πλήκτρο 'Save'. Τόσο τα ρούχα όσο και τα διάφορα μέρη του σώματος συμπεριφέρονται και αυτά ως αντικείμενα του αποθετηρίου σας. Μπορείτε να έχετε παραπάνω από ένα αντικείμενα σε καθένα από τα παραπάνω είδη (π.χ. δύο ζευγάρια παπούτσια ή δύο διαφορετικά σώματα), και να επιλέγετε κάθε φορά ποιο από αυτά θα «φοράει» η ενσάρκωσή σας. Για να το πετύχετε αυτό, αρκεί να κάνετε δεξί κλικ πάνω από το αντικείμενο του αποθετηρίου και να επιλέξετε Wear. Κατά συνέπεια, κάθε χρήστης μπορεί να δημιουργήσει τη δική του βιβλιοθήκη με διαφορετικές όψεις και εμφανίσεις και να τροποποιεί την εμφάνιση της ενσάρκωσής του ανάλογα με το περιβάλλον και το είδος της δραστηριότητας που εκτελεί.

6 Συμπεράσματα

Στο παρόν κεφάλαιο έγινε μια εισαγωγική παρουσίαση του περιβάλλοντος του OpenSimulator, της εγκατάστασης, διαχείρισης και βασικής χρήσης του. Ένα βασικό συμπέρασμα που προκύπτει από τη μελέτη χαρακτηριστικών και λειτουργιών του περιβάλλοντος είναι ότι το OpenSimulator είναι μια ανοιχτή πλατφόρμα εικονικών κόσμων με μεγάλο εύρος δυνατοτήτων, για το οποίο όμως απαιτούνται προσεκτικές και στοχευμένες επιλογές σε ό,τι αφορά τη διανομή, τα στοιχεία εγκατάστασης και τη διαχείριση. Επιβάλλεται να προηγηθεί ενδελεχής μελέτη των διάφορων επιλογών που προσφέρει το περιβάλλον και να ληφθούν αποφάσεις με βάση το πλαίσιο χρήσης του εικονικού κόσμου που θα στηθεί πάνω σε αυτό. Ορισμένες τέτοιες κρίσιμες επιλογές είναι η διαμόρφωση της συσκευής του διακομιστή, η επιλογή διανομής, το μοντέλο λειτουργίας (αυτόνομο ή πλέγμα), το σύστημα βάσεων δεδομένων, το εύρος του κόσμου και η κατανομή του σε υπο-περιοχές, τα δικαιώματα των χρηστών, η δημιουργία ομάδων χρηστών με διαφορετικούς ρόλους και δικαιώματα, η υποστήριξη φωνής, οι διαδικασίες δημιουργίας αντιγράφων ασφαλείας κ.ά. Πριν επενδύσει κάποιος στην εγκατάσταση ενός εικονικού κόσμου που αναμένεται να χρησιμοποιηθεί από μεγάλο αριθμό χρηστών, είναι καλό να δοκιμάσει πρώτα έναν αριθμό από εγκαταστάσεις και τροποποιήσεις, να πειραματιστεί με τις διάφορες επιλογές και να εντοπίσει από νωρίς τυχόν προβλήματα και δυσλειτουργίες που μπορεί να προκύψουν.

Στο επόμενο κεφάλαιο θα εστιάσουμε στη σχεδίαση και διαμόρφωση του χώρου και των αντικειμένων ενός εικονικού κόσμου μέσω των εργαλείων που προσφέρονται στο OpenSimulator. Πριν μελετήσετε όμως τα παραδείγματα του επόμενου κεφαλαίου, είναι καλό να έχετε στη διάθεσή σας μια ολοκληρωμένη εγκατάσταση του OpenSimulator (ή πρόσβαση σε εγκατεστημένο κόσμο) και να έχετε αποκτήσει τις βασικές δεξιότητες χρήσης που περιγράφονται σε αυτό το κεφάλαιο.

Σύνδεσμοι

OpenSimulator: <http://www.opensimulator.org>

MySQL: <http://dev.mysql.com/>

OSGrid: <http://www.osgrid.org>, δημοφιλής διακομιστής (τύπου πλέγματος) OpenSimulator με πολύ μεγάλο αριθμό περιοχών και ενεργή κοινότητα χρηστών

ZADAROO: <http://zadaroo.com/>, ιστότοπος με πλούσιο περιεχόμενο σε μορφή αρχείων αποθετηρίου (IAR) και αρχείων κόσμου (OAR) για το OpenSimulator

Διανομή diva: <http://metaverseink.com/Downloads.html>

Διανομή Sim-on-a-Stick: <http://simonastick.com/>

Υποστήριξη φωνής (Vivox) στο OpenSimulator: <https://support.vivox.com/opensim/>

Ασκήσεις

1. Εγκαταστήστε διακομιστή OpenSimulator στον υπολογιστή σας, ανεβάστε κάποιον έτοιμο κόσμο (σε μορφή OAR), και δοκιμάστε να συνδεθείτε δύο ή περισσότεροι χρήστες σε αυτόν. Προσπαθήστε να επικοινωνήσετε εντός κόσμου και να εξερευνήσετε μαζί το περιβάλλον. Συγκρίνετε τη διεπαφή και τις προσφερόμενες λειτουργίες του OpenSimulator με εναλλακτικά περιβάλλοντα επικοινωνίας, εξερεύνησης και προβολής τρισδιάστατου περιεχομένου.
2. Επισκεφθείτε τη σελίδα του OSGrid, δημιουργήστε λογαριασμό, συνδεθείτε στον κόσμο μέσω του Singularity (το OS Grid περιλαμβάνεται ήδη στη λίστα συνδέσεων) και περιηγηθείτε στις περιοχές του. Καταγράψτε τις περιοχές και τα διαδραστικά στοιχεία που σας έκαναν μεγαλύτερη εντύπωση.

Κεφάλαιο 8: Κατασκευή Περιεχομένου

Σύνοψη

Αντικείμενο του κεφαλαίου είναι η ανάπτυξη και διαμόρφωση του περιεχομένου ενός εικονικού κόσμου με τα εργαλεία που παρέχει το OpenSimulator. Αρχικά παρουσιάζεται η δημιουργία και τοποθέτηση απλών αντικειμένων βασισμένων σε βασικά στερεά (prims), η διαμόρφωση του υλικού τους και η ομαδοποίησή τους σε πιο σύνθετες κατασκευές. Στη συνέχεια αναλύονται σε μεγαλύτερο βάθος οι δυνατότητες παραγωγής περίπλοκων μορφών μέσω προσαρμογής των ιδιοτήτων των βασικών στερεών καθώς και η διαδικασία κατασκευής παραμορφώσιμων αντικειμένων. Περιγράφονται τρόποι εισαγωγής εξωτερικών μοντέλων με τη μορφή ειδικών υφών γλυπτών (sculpted textures) αλλά και πολυγωνικών πλεγμάτων. Στο κεφάλαιο αναφέρονται ακόμα μέθοδοι και εργαλεία διαμόρφωσης του αναλύφου εδάφους και προσαρμογής του φωτισμού του περιβάλλοντος. Στη συνέχεια περιγράφεται η διαδικασία κατασκευής ενσωματώσεων, δηλαδή αντικειμένων που μπορούν να κρατήσουν ή να φορέσουν οι ενσαρκώσεις των χρηστών καθώς και αντικείμενα που τοποθετούνται στην οθόνη (HUD) ως επεκτάσεις της διεπαφής χρήστη. Τέλος, περιγράφεται αναλυτικά η διαδικασία κατασκευής νέων συνθετικών κινήσεων χαρακτήρων που μπορούν να εισαχθούν στο περιβάλλον και να εκτελεστούν κατά τη διάρκεια της αλληλεπίδρασης του χρήστη με τον κόσμο. Τα παραπάνω παρουσιάζονται με αναλυτικά παραδείγματα που μπορεί να επαναλάβει ο αναγνώστης και συμπεριλαμβάνουν πρακτικές οδηγίες για την κατασκευή σύνθετων περιβαλλόντων και αντικειμένων.

Προαπαιτούμενη γνώση

Απαιτούνται γνώσεις πληροφορικής και δεξιότητες βασικής χρήσης του OpenSimulator.

1 Εισαγωγή

Η κατασκευή του περιεχομένου είναι ένα ιδιαίτερα μεγάλο σε διάρκεια και κρίσιμο για την επιτυχία στάδιο στην ανάπτυξη εικονικών κόσμων. Όπως έχουμε αναφέρει και σε προηγούμενα κεφάλαια, τα περιεχόμενα ενός εικονικού κόσμου, δηλαδή τα τρισδιάστατα και δισδιάστατα στοιχεία που συνθέτουν το τελικό οπτικό αποτέλεσμα, εξυπηρετούν πολλαπλούς σκοπούς. Ένα μέρος του περιεχομένου έχει καθαρά αισθητικό χαρακτήρα, δηλαδή αξιοποιείται για να απεικονίσει το κατάλληλο περιβάλλον πλαίσιο, να δημιουργήσει την επιθυμητή ατμόσφαιρα στον χρήστη, να αυξήσει την αληθοφάνεια του χώρου και εντέλει να συνεισφέρει στην ενίσχυση της αίσθησης της «παρουσίας». Ένα άλλο μέρος περιγράφει στοιχεία με «ενεργό» ρόλο στο περιβάλλον: αντικείμενα και χαρακτήρες που κινούνται, αλληλεπιδρούν και εκτελούν ενέργειες στον κόσμο ως συνέπεια των κανόνων του ή και κάποιας διαδικασίας προσομοίωσης. Ένα τρίτο μέρος αφορά τα στοιχεία διεπαφής, δηλαδή τα δισδιάστατα ή τρισδιάστατα αντικείμενα του κόσμου με τα οποία έρχεται σε άμεση επαφή και επικοινωνία ο χρήστης και τα αξιοποιεί για την αλληλεπίδρασή του με το περιβάλλον, όπως πλήκτρα, στοιχεία ελέγχου, εργαλεία κ.λπ. Τέλος, η ίδια η διάταξη των στοιχείων στο περιβάλλον επηρεάζει σε μεγάλο βαθμό την πλοήγηση του χρήστη, την ευκολία κατανόησης και χαρτογράφησης, την προσβασιμότητα σε αντικείμενα και χώρους ειδικού ενδιαφέροντος κ.ά. Από όλα τα παραπάνω γίνεται σαφές ότι η κατάλληλη επιλογή και διάταξη του περιεχομένου επιδρά καταλυτικά τόσο στην αισθητική όσο και στη λειτουργικότητα του κόσμου, και απαιτείται προσεκτική μελέτη και σχεδίαση για την κατασκευή εύχρηστων και αποδοτικών τρισδιάστατων περιβαλλόντων. Επιπλέον, η διαδικασία αυτή περιλαμβάνει την ανάπτυξη ενός μεγάλου αριθμού αντικειμένων, υφών και δισδιάστατων στοιχείων, ανάλογα πάντα και με το επιθυμητό μέγεθος του υπό σχεδίαση περιβάλλοντος, και μπορεί εντέλει να είναι αρκετά επίπονη και μεγάλη σε διάρκεια.

Στο περιβάλλον του OpenSimulator ένα σημαντικό τμήμα της κατασκευής και διάταξης του περιεχομένου γίνεται εντός κόσμου. Αυτό είναι ένα ιδιαίτερο στοιχείο που είχε εξαρχής το Second Life και το οποίο συνέβαλε καθοριστικά στη μεγάλη εξάπλωση και επιτυχία του. Παρά το γεγονός ότι τα εργαλεία που προσφέρονται υπολείπονται σημαντικά σε σχέση με αντίστοιχα επαγγελματικά εργαλεία, για πρώτη φορά εμφανίστηκε ένας εικονικός κόσμος που επέτρεπε την κατασκευή περιεχομένου με τη λογική WYSIWYG (what you see is what you get – αυτό που βλέπεις είναι στην τελική του μορφή) που υπάρχει σήμερα σε πληθώρα άλλων εφαρμογών, όπως για παράδειγμα στους επεξεργαστές κειμένου. Την ίδια στιγμή που κατασκευάζει ένας χρήστης νέο περιεχόμενο, το παρατηρεί, το τοποθετεί και τελικά το «δοκιμάζει» μέσα στον ίδιο τον κόσμο. Και

όλα αυτά χωρίς να έχει την υποχρέωση να διαθέτει κάποιο εξωτερικό πρόγραμμα τρισδιάστατης μοντελοποίησης. Αυτή η προσφερόμενη δυνατότητα έκανε τη δημιουργία περιεχομένου πολύ πιο άμεση και συμμετοχική. Το μειονέκτημα είναι βεβαίως ότι τα εργαλεία στερούνται της λεπτομέρειας, της ευχρηστίας και της λειτουργικότητας των σύγχρονων εργαλείων μοντελοποίησης. Επιπλέον, υπάρχουν στοιχεία του περιεχομένου, όπως για παράδειγμα οι συνθετικές κινήσεις των χαρακτήρων, τα οποία δεν μπορούν να κατασκευαστούν εντός κόσμου, αλλά θα πρέπει να εισαχθούν από αρχεία που έχουν δημιουργηθεί σε εξωτερικές εφαρμογές.

Αντικείμενο του κεφαλαίου είναι η παρουσίαση των βασικών εργαλείων και τεχνικών κατασκευής περιεχομένου στους κόσμους του OpenSimulator. Θα δώσουμε έμφαση στα πιο σημαντικά στοιχεία που σχετίζονται με τη δημιουργία αντικειμένων, τη σχεδίαση του χώρου, τη διεπαφή με τον χρήστη και την παραγωγή συνθετικών κινήσεων. Αρχικά θα εξετάσουμε τα βασικά εργαλεία κατασκευής και τοποθέτησης απλών αντικειμένων και διαμόρφωσης του υλικού τους. Στη συνέχεια θα εστιάσουμε στις δυνατότητες περιγραφής πιο σύνθετων γεωμετριών με τα εργαλεία του κόσμου καθώς και στην εισαγωγή εξωτερικών αντικειμένων. Θα παρουσιάσουμε τα εργαλεία διαμόρφωσης του εδάφους και προσαρμογής του φωτισμού του χώρου, θα εξετάσουμε τρόπους με τους οποίους μπορούμε να ενσωματώνουμε αντικείμενα στο σώμα της ενσάρκωσης αλλά και στη δισδιάστατη διεπαφή (HUD) του χρήστη και, τέλος, θα μελετήσουμε εργαλεία και τεχνικές δημιουργίας και εισαγωγής συνθετικών κινήσεων χαρακτήρων. Όλα τα παραπάνω συμπληρώνονται με παραδείγματα χρήσης σε μορφή εικόνων ή και βίντεο.

2 Κατασκευή και επεξεργασία αντικειμένων

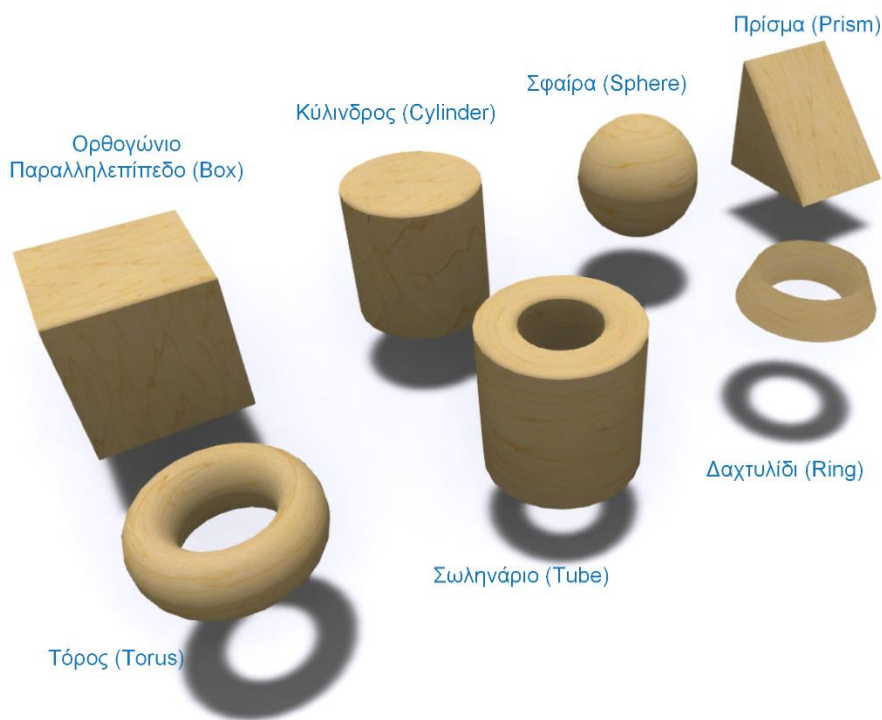
Τα αντικείμενα που μπορούν να κατασκευαστούν μέσα στον κόσμο του OpenSimulator, αν και αποδίδονται οπτικά ως συλλογές πολυγώνων, δεν αποτελούν αυθαίρετα πολυγωνικά πλέγματα (meshes). Ο χρήστης μπορεί να κατασκευάσει μόνο βασικά γεωμετρικά στερεά, όπως κύβος, σφαίρα, κύλινδρος κ.λπ. Τα στερεά αυτά μπορεί κάποιος στη συνέχεια να τα επεξεργαστεί και να τα συνδυάσει δημιουργώντας πιο περίπλοκες μορφές που αποδίδουν αρκετά καλά ένα μεγάλο εύρος αντικειμένων. Αυτή η ιδιαιτερότητα αρχικά ξενίζει τους έμπειρους σχεδιαστές, που είναι συνηθισμένοι στη χρήση περίπλοκων περιβαλλόντων μοντελοποίησης με πληθώρα εργαλείων. Αν δοκιμάσετε κι εσείς να εφαρμόσετε στην πράξη κάποιες από τις τεχνικές μοντελοποίησης που παρουσιάζονται στο Κεφάλαιο 2 αυτού του βιβλίου μέσω κάποιου επαγγελματικού εργαλείου, είναι πιθανό να αισθανθείτε την ίδια απογοήτευση όταν θα αντικρίσετε τις περιορισμένες δυνατότητες κατασκευής στο OpenSimulator. Από την άλλη μεριά όμως, η απλότητα στα υποστηριζόμενα αντικείμενα και εργαλεία κάνει το περιβάλλον περισσότερο προσιτό σε αρχάριους χρήστες, οι οποίοι μπορούν να υλοποιήσουν βασικές κατασκευές με εύκολο και διασκεδαστικό τρόπο. Επιπλέον, η τροποποίηση ορισμένων παραμέτρων των βασικών στερεών μπορεί να οδηγήσει σε πιο ενδιαφέρουσες μορφές, όπως θα δούμε στην επόμενη ενότητα, ώστε τελικά να είναι σε θέση ένας σχεδιαστής που θα μάθει καλά τις δυνατότητες αυτές να δημιουργήσει σχεδόν οτιδήποτε. Άλλωστε, αν παρατηρήσει κάποιος το κατασκευασμένο περιεχόμενο στους κόσμους του Second Life, θα εντυπωσιαστεί από την ποιότητα και τη λεπτομέρεια των κατασκευών, παρά το γεγονός ότι βασίζονται ως επί το πλείστον σε συλλογές βασικών στερεών.

Εκτός από την απλότητα υπάρχει και ένας ακόμη σημαντικός λόγος υπέρ της χρήσης βασικών στερεών: η σημαντική μείωση στον όγκο δεδομένων που διακινούνται μέσω διαδικτύου. Ας σκεφτούμε την περίπτωση που το πρόγραμμα σύνδεσης του χρήστη συνδέεται για πρώτη φορά σε έναν εικονικό κόσμο και η ενσάρκωσή του βρίσκεται σε ένα πλούσιο σε αντικείμενα περιβάλλον. Ο διακομιστής θα πρέπει να μεταφέρει στο πρόγραμμα πελάτη όλες τις απαιτούμενες πληροφορίες σχετικά με τη γεωμετρία του περιβάλλοντος, ώστε το δεύτερο να μπορέσει να αποδώσει οπτικά τη σκηνή. Αν κάθε αντικείμενο ήταν μοντελοποιημένο ως συλλογή πολυγώνων, θα έπρεπε να μεταφερθεί μεταξύ άλλων ο πίνακας κορυφών και πολυγώνων που το περιγράφει (βλ. Κεφ. 2). Τα πολύγωνα αυτά όμως μπορεί να ήταν και μερικές εκατοντάδες ή και χιλιάδες. Αντίθετα, ένα περίπλοκο αντικείμενο μπορεί να κατασκευαστεί από λίγα μόνο βασικά στερεά, για καθένα από τα οποία απαιτούνται πολύ λίγες πληροφορίες, για παράδειγμα το κέντρο και η ακτίνα μιας σφαίρας, οι διαστάσεις ενός κύβου κ.λπ. Συνεπώς η ποσότητα των δεδομένων που διακινείται στην περίπτωση της μοντελοποίησης μέσω βασικών στερεών είναι σημαντικά μικρότερη, κάτι που οδηγεί σε αντίστοιχη αύξηση της ταχύτητας της δικτυακής επικοινωνίας, άρα και της τελικής απόδοσης της εφαρμογής.

Τα βασικά στερεά του OpenSimulator ονομάζονται prims (από το primitive) και είναι τα παρακάτω (εικόνα 8.1):

- ορθογώνιο παραλληλεπίπεδο (box),
- κύλινδρος (cylinder),
- σφαίρα (sphere),
- πρίσμα (prism),
- τόρος (torus),
- σωληνάριο (tube),
- δαχτυλίδι (ring).

Η μορφή των στερεών αυτών επηρεάζεται από έναν αριθμό παραμέτρων, τις οποίες θα εξετάσουμε αναλυτικότερα στην επόμενη ενότητα. Οι αλλαγές στις τιμές των παραμέτρων των βασικών στερεών μπορεί να οδηγήσει σε άλλες μορφές, όπως πυραμίδα, κώνος, ημισφαίριο κ.λπ. Πράγματι, αν δούμε τις επιλογές που μας δίνει το πρόγραμμα σύνδεσης για την κατασκευή ενός νέου αντικείμενου θα εντοπίσουμε αντικείμενα σαν τα παραπάνω πέρα από τα βασικά στερεά. Στη συνέχεια θα δοκιμάσουμε να κατασκευάσουμε το πρώτο μας αντικείμενο.



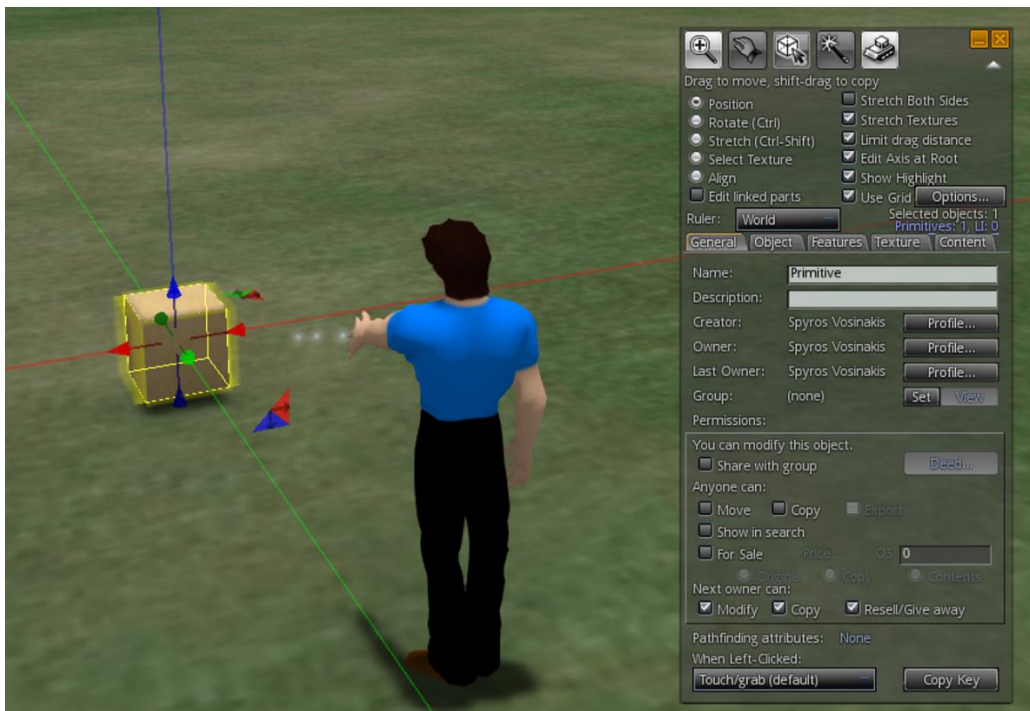
Εικόνα 8.1 Τα βασικά στερεά του OpenSimulator.

2.1 Δημιουργία και βασική επεξεργασία

Μπορείτε να προσθέσετε ένα νέο αντικείμενο στον κόσμο με δύο τρόπους: πατώντας το πλήκτρο “Build” στο κάτω μέρος της οθόνης ή κάνοντας δεξί κλικ στο έδαφος και επιλέγοντας από το μενού “Create”. Και στις δύο περιπτώσεις εμφανίζεται ένα παράθυρο επεξεργασίας το οποίο περιλαμβάνει έναν αριθμό από πλήκτρα για την κατασκευή βασικών αντικειμένων. Τα εικονίδια των πλήκτρων απεικονίζουν τη μορφή του αντίστοιχου αντικείμενου που θα προστεθεί στον κόσμο. Εκτός από τα δύο τελευταία που οδηγούν στην κατασκευή δέντρων και φυτών (θα αναφερθούμε σε αυτά στην ενότητα 4 αυτού του κεφαλαίου), όλα τα υπόλοιπα αναφέρονται σε βασικά στερεά: παραλληλεπίπεδο, πρίσμα, πυραμίδα, τετράεδρο, κύλινδρος, ημικύλινδρος, κώνος, ημικώνος, σφαίρα, ημισφαίριο, τόρος, σωληνάριο και δαχτυλίδι. Επιλέξτε το παραλληλεπίπεδο και κάντε κλικ σε ένα σημείο του εδάφους. Θα παρατηρήσετε ότι προστέθηκε ένα νέο αντικείμενο στον κόσμο: ένας κύβος με υφή ξύλου.

Αν εκτελέσετε την παραπάνω ενέργεια, θα παρατηρήσετε ότι τον κύβο που μόλις προσθέσαμε συμπληρώνουν τρεις χρωματιστοί άξονες. Επιπλέον, το παράθυρο επεξεργασίας μέσω του οποίου τον

δημιουργήσαμε άλλαξε καρτέλα και οι επιλογές του πλέον σχετίζονται με τη θέση, τον προσανατολισμό και την κλίμακα του νέου στερεού. Πράγματι, από τη στιγμή που προστίθεται ένα νέο βασικό στερεό, το περιβάλλον μπαίνει σε κατάσταση επεξεργασίας και, μέσω αυτής, μπορείτε να τροποποιήσετε το μέγεθος και την τοποθέτηση του αντικειμένου (εικ. 8.2).



Εικόνα 8.2 Εισαγωγή ενός νέου στερεού.

Οι τρεις άξονες που εμφανίζονται στο νέο αντικείμενο είναι εικονικά στοιχεία ελέγχου (βλ. Κεφ. 4), μέσω των οποίων μπορούμε να αλλάξουμε τη θέση του στον χώρο. Αν κάνετε κλικ σε ένα βέλος του κόκκινου άξονα και σύρετε το ποντίκι προς μία από τις δύο διευθύνσεις του, το αντικείμενο θα μετατοπιστεί αντίστοιχα στον άξονα των X. Ο πράσινος άξονας αναφέρεται στο Y και ο κόκκινος στο Z. Μπορείτε λοιπόν με αυτόν τον τρόπο να επιλέξετε την ακριβή θέση στην οποία θα τοποθετηθεί το αντικείμενο. Αν κρατήσετε πατημένο το πλήκτρο Control, οι άξονες αλλάζουν σε χρωματιστούς δίσκους. Επιλέγοντας έναν από τους δίσκους και μετακινώντας το ποντίκι κατά μήκος της περιφέρειάς του περιστρέφεται το αντικείμενο γύρω από τον αντίστοιχο άξονα. Τέλος, κρατώντας πατημένο το Shift και το Control μπορείτε να αλλάξετε το μέγεθος του αντικειμένου. Θα δείτε να εμφανίζονται έξι χρωματιστά στοιχεία ελέγχου, ένα για κάθε επιφάνεια του κύβου. Με τα στοιχεία αυτά μπορούμε να αυξομειώσουμε το μέγεθός του στην αντίστοιχη διάσταση. Αν θέλουμε να αλλάξουμε συνολικά το μέγεθος και όχι σε μεμονωμένες διαστάσεις, μπορούμε να χρησιμοποιήσουμε ένα από τα οκτώ στοιχεία ελέγχου λευκού χρώματος στις γωνίες του κύβου. Δοκιμάστε να πειραματιστείτε με την οπτική τοποθέτηση και αλλαγή κλίμακας του αντικειμένου.

Εναλλακτικά της χρήσης των συμπληρωματικών πλήκτρων Control και Shift μπορείτε να αλλάξετε την κατάσταση επεξεργασίας από μετατόπιση σε περιστροφή ή αλλαγή κλίμακας επιλέγοντας στο παράθυρο επεξεργασίας Rotate ή Stretch αντίστοιχα. Στην περίπτωση αυτήν εμφανίζονται τα κατάλληλα στοιχεία ελέγχου χωρίς να χρειάζεται να κρατάτε πατημένο κάποιο πλήκτρο.

Η οπτική επεξεργασία των αντικειμένων είναι εύκολη και διαισθητική διεργασία, όμως στερείται ακρίβειας. Σε αρκετές περιπτώσεις μπορεί να χρειαστεί το αντικείμενό σας να έχει προκαθορισμένες διαστάσεις ή να βρίσκεται σε συγκεκριμένη θέση στον κόσμο. Μπορείτε να καθορίσετε με ακρίβεια τις τιμές της θέσης, της περιστροφής και της κλίμακας του αντικειμένου με τη χρήση των στοιχείων ελέγχου του παράθυρου επεξεργασίας. Υπάρχουν πεδία εισαγωγής κειμένου για τις τρεις τιμές θέσης, τις τρεις τιμές μεγέθους και τις τρεις γωνίες προσανατολισμού.

Η θέση δίνεται σε μέτρα και αναφέρεται σε απόλυτες συντεταγμένες εντός της περιοχής (region – βλ. Κεφ. 7) στην οποία βρίσκεται το αντικείμενο. Μια περιοχή έχει διαστάσεις 256 x 256 μέτρα και το ύψος της θάλασσας είναι στα 20 μέτρα. Επομένως η θέση στο κέντρο της περιοχής ακριβώς πάνω στο ύψος της θάλασσας

είναι η (128, 128, 20). Η τοποθέτηση του αντικειμένου γίνεται σε σχέση με το κέντρο του. Επομένως, αν ο κύβος που δημιουργήσετε έχει τις εξ' ορισμού διαστάσεις (0.5 x 0.5 x 0.5) και το ύψος του εδάφους είναι στα 21 μέτρα, η τιμή του Z στη θέση του θα πρέπει να είναι 21.25 για να ακουμπάει στο έδαφος.

Το μέγεθος δίνεται επίσης σε μέτρα και οι επιτρεπτές τιμές σε κάθε διάσταση είναι μεταξύ 0.01 και 96.

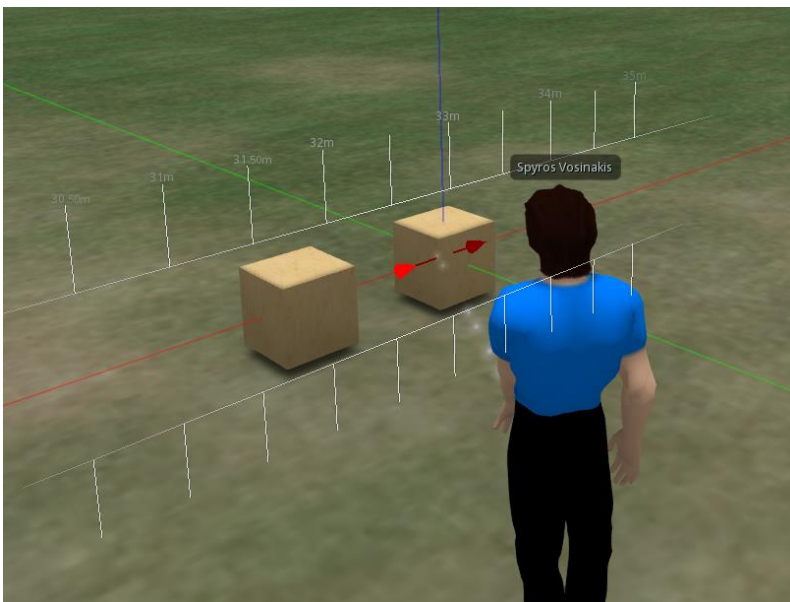
Τέλος, οι τιμές περιστροφής δίνονται σε μοίρες και αναφέρονται στις γωνίες περιστροφής γύρω από τον κάθε άξονα (βλ. Κεφ. 2, γωνίες Euler).

Αν κλείσετε το παράθυρο επεξεργασίας, βγαίνετε από την κατάσταση επεξεργασίας, εξαφανίζονται τα στοιχεία ελέγχου και το αντικείμενο παίρνει την τελική του θέση στον χώρο. Μπορείτε να το επεξεργαστείτε ξανά αν κάνετε δεξί κλικ πάνω του και επιλέξετε 'Edit...'. Αν βρίσκεστε σε κατάσταση επεξεργασίας και θέλετε να επεξεργαστείτε κάποιο άλλο αντικείμενο, αρκεί να κάνετε κλικ στο νέο αντικείμενο.

Κάθε αντικείμενο που δημιουργείτε έχει εξ' ορισμού το όνομα 'Primitive'. Μπορείτε να αλλάξετε το όνομα του αντικειμένου σας ή και να προσθέσετε κάποια περιγραφή, αν στο παράθυρο επεξεργασίας επιλέξετε την καρτέλα General. Εκεί θα βρείτε τις επιλογές Name και Description στις οποίες μπορείτε να θέσετε τιμές της αρεσκείας σας.

Μπορείτε επίσης αποκτήσετε ένα αντίγραφο του αντικειμένου στο αποθετήριό σας αν κάνετε δεξί κλικ πάνω του και επιλέξετε More > Take Copy. Ένα αντίγραφο με το ίδιο όνομα θα προστεθεί στο αποθετήριο στον φάκελο 'Objects'. Αν στη συνέχεια σύρετε με το ποντίκι το αντικείμενο αυτό από το αποθετήριο στον κόσμο, θα προστεθεί ένα ακόμη αντίγραφο στον κόσμο. Μπορείτε να επαναλάβετε τη διαδικασία και να δημιουργήσετε πολλαπλά αντίγραφα. Αν αντί για 'Take Copy' επιλέξετε 'Take', τότε το ίδιο το αντικείμενο θα βρεθεί στο αποθετήριο και θα εξαφανιστεί από τον κόσμο.

Ένας εναλλακτικός τρόπος να δημιουργείτε αντίγραφα είναι αν βρίσκεστε σε κατάσταση επεξεργασίας ενός αντικειμένου και κρατώντας πατημένο το Shift το μετακινήσετε σε κάποια διεύθυνση μέσω των εικονικών χειριστηρίων. Τότε θα εισαχθεί ένα νέο αντίγραφο του αντικειμένου στον κόσμο που θα ακολουθεί την κίνηση του ποντικιού, ενώ το αρχικό θα παραμείνει στη θέση του (εικ. 8.3). Η τεχνική αυτή είναι εξαιρετικά χρήσιμη κατά τη διάρκεια κατασκευής σύνθετων αντικειμένων που περιλαμβάνουν πολλαπλά αντίγραφα βασικών στοιχείων, όπως είναι για παράδειγμα οι τοίχοι ενός κτιρίου, τα πόδια ενός τραπέζιου κ.λπ.



Εικόνα 8.3 Δημιουργία αντιγράφου με Shift και μετατόπιση.



Σύνδεσμος:
<https://youtu.be/s11dk6X1dqM>

Video 8.1 Δημιουργία και βασική επεξεργασία αντικειμένων.

Τέλος, για να διαγράψετε ένα αντικείμενο από τον κόσμο, αρκεί να το έχετε επιλέξει για επεξεργασία και να πατήσετε το πλήκτρο 'Del' του πληκτρολογίου. Εναλλακτικά μπορείτε να κάνετε δεξί κλικ πάνω του και να επιλέξετε More > Delete από το σχετικό μενού. Η δημιουργία και βασική επεξεργασία αντικειμένων παρουσιάζεται στο βίντεο 8.1.

2.2 Υλικό

Μπορείτε να επεξεργαστείτε το υλικό του αντικειμένου αν επιλέξετε από το παράθυρο επεξεργασίας την καρτέλα 'Texture'. Εκεί μπορείτε να δείτε και να αλλάξετε το *χρώμα (color)*, την *υφή (texture)*, τη *διαφάνεια (alpha)* και τη *φωτεινότητα (glow)*.

Αν επιλέξετε το χρώμα, μπορείτε να το αλλάξετε στο χρώμα της αρεσκείας σας μέσω του παράθυρου επιλογής που εμφανίζεται. Θα παρατηρήσετε ότι το τελικό αποτέλεσμα στο αντικείμενο είναι ένας συνδυασμός του χρώματος με την τρέχουσα υφή. Όσο πιο φωτεινό είναι το χρώμα που επιλέξατε τόσο πιο εμφανής είναι η υφή στην επιφάνεια του αντικειμένου.

Αν επιλέξετε την επιλογή Full Bright (ακριβώς κάτω από την επιλογή χρώματος), τότε το χρώμα και η υφή του αντικειμένου δεν φωτοσκιάζονται και διατηρούν τη φωτεινότητά τους σε όλη την επιφάνεια. Η επιλογή αυτή είναι χρήσιμη στην περίπτωση υφών που περιέχουν κάποιο κείμενο που θα πρέπει να διαβάσει ο χρήστης ή εικόνες τις οποίες θέλουμε να εμφανίσουμε στη σωστή φωτεινότητα (π.χ. διαφάνειες που προβάλλονται σε μια εικονική τάξη).

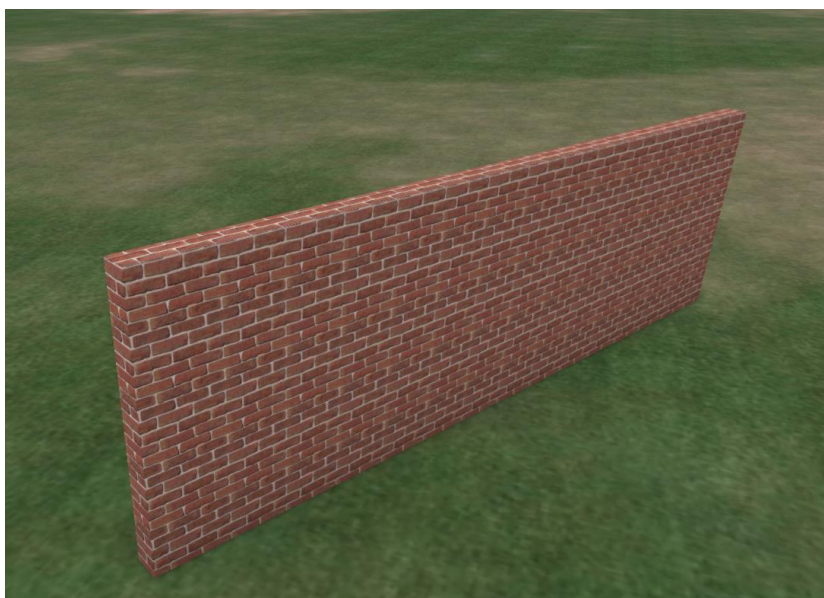
Μπορείτε να μεταβάλλετε τη διαφάνεια του αντικειμένου σας αλλάζοντας την τιμή του Alpha. Οι τιμές που δέχεται είναι από το μηδέν μέχρι το 100, όπου στο μηδέν το αντικείμενο είναι πλήρως αδιαφανές και στο 100 αόρατο.

Αν θέλετε το αντικείμενό σας να λάμπει, για παράδειγμα να αποδοθεί υλικό όπως χρυσός ή ασήμι, μπορείτε να μεταβάλλετε την τιμή του Glow. Η μέγιστη τιμή λάμψης είναι το 1.

Επιλέγοντας το Texture μπορείτε να αλλάξετε την υφή του αντικειμένου. Μπορείτε να διαλέξετε μια από τις δικές σας εικόνες που βρίσκονται στον φάκελο Textures του αποθετηρίου σας ή μία από τις γενικές υφές του περιβάλλοντος που βρίσκονται στον φάκελο 'OpenSim Library / Texture Library'. Υπενθυμίζεται ότι μπορείτε να ανεβάσετε και νέες εικόνες στο αποθετήριό σας μέσω της επιλογής File > Upload Image (free) στο κεντρικό μενού του Singularity. Αν δεν θέλετε το αντικείμενό σας να έχει εικόνα υφής, μπορείτε να πατήσετε το πλήκτρο Blank στο παράθυρο επιλογής υφής. Τέλος, αν αλλάξετε στη λίστα επιλογών του παράθυρου το Texture (diffuse) σε Bumpiness ή Shininess μπορείτε να δημιουργήσετε τραχιές επιφάνειες ή να προσθέσετε γυαλάδα και αντανάκλασεις του περιβάλλοντος αντίστοιχα (βλ. Κεφ. 2).

Οι αλλαγές στο υλικό εφαρμόζονται εξ ορισμού σε όλη την επιφάνεια του αντικειμένου. Αν θέλετε να αλλάξετε μόνο επιμέρους όψεις, μπορείτε να το κάνετε επιλέγοντας 'Select Texture' στο παράθυρο επεξεργασίας. Στη συνέχεια μπορείτε να επιλέξετε μία μόνο επιφάνεια κάνοντας κλικ πάνω της ή περισσότερες με Shift και κλικ. Οι αλλαγές που θα κάνετε θα εφαρμοστούν μόνο στις επιλεγμένες επιφάνειες.

Κάποιες επιπλέον επιλογές της καρτέλας 'Texture' ρυθμίζουν τον τρόπο απεικόνισης της υφής (*texture mapping*) στο αντικείμενο. Οι επιλογές *οριζόντιας και κάθετης κλίμακας (horizontal scale, vertical scale)* ρυθμίζουν τον αριθμό των επαναλήψεων της εικόνας υφής σε κάθε επιφάνεια. Αν η τιμή είναι και στις δύο επιλογές 1.0, τότε η εικόνα θα καλύπτει πλήρως την επιφάνεια. Αν αυξηθεί η τιμή σε κάποια από τις δύο διαστάσεις, τότε το μέγεθος της εικόνας θα μικρύνει και η υφή θα επαναλαμβάνεται καλύπτοντας την επιφάνεια με παραπάνω αντίγραφα. Αντίθετα, αν γίνει μικρότερη του 1.0, τότε θα φαίνεται ένα μέρος της εικόνας μόνο. Εναλλακτικά της οριζόντιας και κάθετης κλίμακας μπορείτε να αλλάξετε την *τιμή επανάληψης ανά μέτρο (repeats per meter)*. Με βάση την τιμή αυτή θα τροποποιηθεί η κλίμακα της εικόνας υφής έτσι ώστε να έχει σταθερό μέγεθος ανεξαρτήτως του μεγέθους της επιφάνειας στην οποία θα εφαρμοστεί. Η ρύθμιση αυτή είναι χρήσιμη κυρίως σε υφές που επαναλαμβάνονται, για παράδειγμα τούβλα, γρασίδι κ.λπ., ώστε να απεικονίζονται σε σωστή κλίμακα σε όλες τις επιφάνειες. Στην εικόνα 8.4 φαίνεται ένας τοίχος στον οποίο έχει εφαρμοστεί υφή τούβλου με τιμή επανάληψης ανά μέτρο 1.5. Τέλος, μπορείτε να μετατοπίσετε ή να περιστρέψετε την εικόνα υφής στην επιφάνεια μέσω των επιλογών 'Horizontal offset', 'Vertical offset' και 'Rotation'.



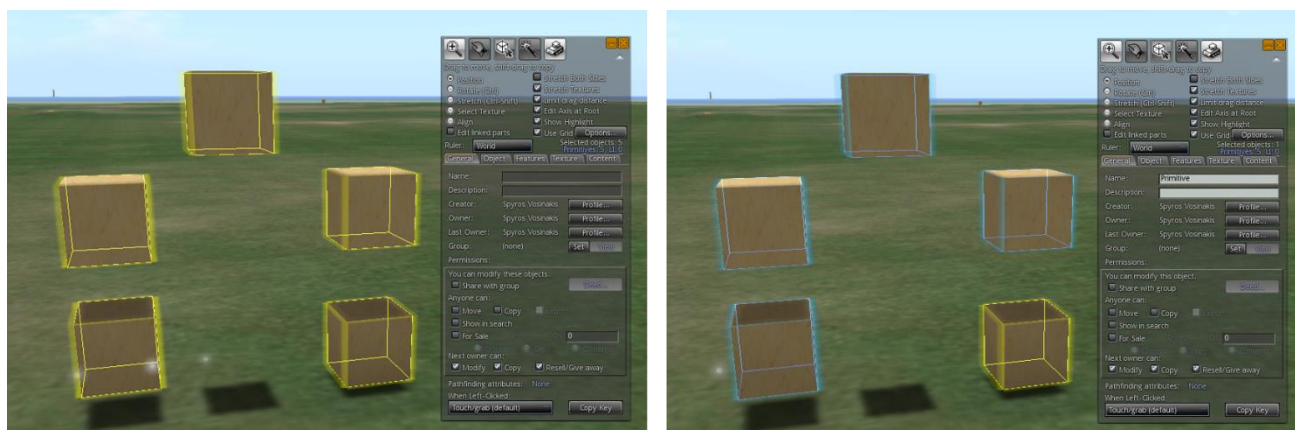
Εικόνα 8.4 Εφαρμογή υφής σε στερεό με ενιαία τιμή επανάληψης ανά μέτρο 1.5.

2.3 Σύνθετα αντικείμενα

Ένα βασικό στερεό από μόνο του δεν αρκεί τις περισσότερες φορές για να αποδώσει κάποιο πραγματικό ή υποθετικό αντικείμενο. Μπορείτε όμως να κατασκευάσετε πιο σύνθετες μορφές αν συνδυάσετε πολλαπλά βασικά στερεά. Αν στη συνέχεια θέλετε η τελική μορφή να συμπεριφέρεται σαν ένα αντικείμενο, μπορείτε να ομαδοποιήσετε τα στερεά που την αποτελούν δημιουργώντας έναν *σύνδεσμο* (*link*).

Για να το πετύχετε αυτό θα πρέπει αρχικά να επιλέξετε όλα τα αντικείμενα που θέλετε να ομαδοποιηθούν (εικ. 8.5). Αυτό μπορεί να γίνει με δύο τρόπους. Ο πρώτος τρόπος είναι να επιλέξετε ένα από αυτά με δεξί κλικ και 'Edit' και στη συνέχεια να προσθέσετε στην επιλογή και τα υπόλοιπα με Shift + κλικ. Ο δεύτερος τρόπος είναι να σχηματίσετε στην οθόνη με πατημένο κλικ ένα ορθογώνιο ενώ είστε ήδη σε κατάσταση επεξεργασίας. Όλα τα αντικείμενα εντός του ορθογωνίου θα επιλεγούν.

Όταν έχετε ολοκληρώσει την πολλαπλή επιλογή, μπορείτε να ομαδοποιήσετε τα αντικείμενα πατώντας Ctrl + L. Τότε το περίγραμμα όλων των αντικειμένων που είχατε επιλέξει εκτός του τελευταίου θα αλλάξει σε μπλε χρώμα, ενώ το τελευταίο αντικείμενο, που ονομάζεται ρίζα (root) της ομάδας, θα παραμείνει κίτρινο (εικ. 8.5).



Εικόνα 8.5 Επιλογή πολλαπλών αντικειμένων (αριστερά) και ομαδοποίηση (δεξιά).

Μετά την ομαδοποίηση η ομάδα αντικειμένων συμπεριφέρεται σαν ένα, δηλαδή πλέον μπορεί να μετατοπιστεί, περιστραφεί και αυξομειωθεί το μέγεθός του συνολικά. Το ενιαίο αυτό αντικείμενο διατηρεί τα στοιχεία (το όνομα, την περιγραφή, τα περιεχόμενα κ.λπ.) που αρχικά είχε το αντικείμενο «ρίζα». Αν θέλουμε να

επεξεργαστούμε ένα μεμονωμένο αντικείμενο μιας ομάδας αντικειμένων, τότε θα πρέπει να επιλέξουμε την επιλογή “Edit Linked Parts” στο παράθυρο επεξεργασίας και στη συνέχεια να επιλέξουμε το αντικείμενο.

Τέλος, αν θέλουμε να καταργήσουμε έναν σύνδεσμο αντικειμένων, αρκεί να τον επιλέξουμε και να πατήσουμε Shift + Ctrl + L.

Στο σημείο αυτό οφείλουμε να επισημάνουμε ότι το OpenSimulator δεν μας δίνει τη δυνατότητα ιεραρχικών ομάδων αντικειμένων. Αυτό σημαίνει πρακτικά ότι, αν επιλέξουμε μια υπάρχουσα ομάδα, προσθέσουμε ένα ακόμη αντικείμενο ή μια άλλη ομάδα και τα ομαδοποιήσουμε ξανά, το αποτέλεσμα θα είναι μια νέα ομάδα που θα περιέχει όλα τα βασικά στερεά των προηγούμενων και καμία επιπλέον πληροφορία. Το αποτέλεσμα δηλαδή θα είναι ακριβώς το ίδιο με το να είχαμε επιλέξει εξ αρχής όλα τα στερεά και να τα είχαμε ομαδοποιήσει. Αυτή η έλλειψη είναι σίγουρα περιοριστική σε σχέση με τη χρήση των γραφημάτων σκηνοικού άλλων περιβαλλόντων, όπως για παράδειγμα το Unity, που επιτρέπουν ιεραρχίες μετασχηματισμών.

Ας δούμε ένα παράδειγμα δημιουργίας ενός σύνθετου αντικειμένου. Θα δοκιμάσουμε να κατασκευάσουμε μια απλή καρέκλα αποτελούμενη από πέντε βασικά στερεά.

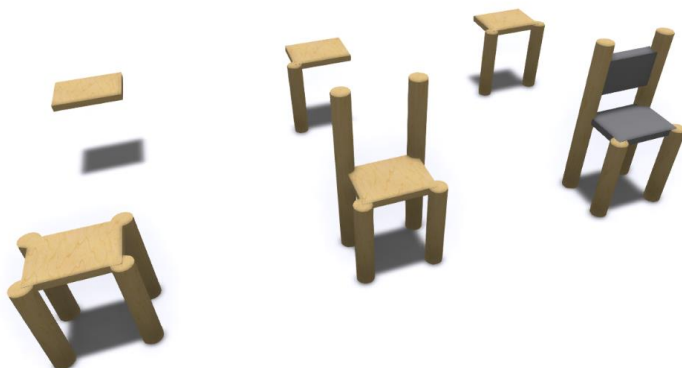
Αρχικά εισάγουμε ένα ορθογώνιο παραλληλεπίπεδο, το οποίο θα είναι η έδρα της καρέκλας, δηλαδή η επιφάνεια στην οποία θα κάθονται οι ενσαρκώσεις. Του δίνουμε με τα οπτικά εργαλεία τις διαστάσεις που μας ικανοποιούν. Αν θέλουμε μπορούμε να στρογγυλέψουμε τους αριθμούς για ευκολότερους υπολογισμούς. Στο παράδειγμά μας οι διαστάσεις της έδρας είναι (0.4, 0.3, 0.04). Την τοποθετούμε στο επιθυμητό ύψος πάνω από την επιφάνεια του εδάφους. Μπορούμε να φέρουμε την ενσάρκωσή μας δίπλα ώστε να βεβαιωθούμε ότι το ύψος είναι κατάλληλο.

Στη συνέχεια δημιουργούμε έναν κύλινδρο για το πόδι της καρέκλας. Και πάλι, τον επεξεργαζόμαστε οπτικά, ώστε να του δώσουμε το κατάλληλο μέγεθος και να τον τοποθετήσουμε στη σωστή θέση. Στο παράδειγμά μας ο κύλινδρος έχει διαστάσεις (0.05, 0.05, 0.5). Θα χρειαστεί να αλλάζουμε συχνά οπτική για να τον μεταφέρουμε σε κατάλληλη θέση σε σχέση με την έδρα της καρέκλας. Όταν βεβαιωθούμε ότι το αποτέλεσμα είναι εντάξει, δημιουργούμε ένα αντίγραφο του ποδιού και το μεταφέρουμε στην άλλη άκρη της έδρας στον άξονα X. Μπορούμε να διορθώσουμε την τοποθέτηση αν διαβάσουμε και επεξεργαστούμε τις τιμές. Αν για παράδειγμα στον άξονα X η θέση της έδρας (το κέντρο της) είναι x και του πρώτου ποδιού $x-\delta$, τότε του δεύτερου θα πρέπει να είναι $x+\delta$, ώστε να ισαπέχουν από το κέντρο.

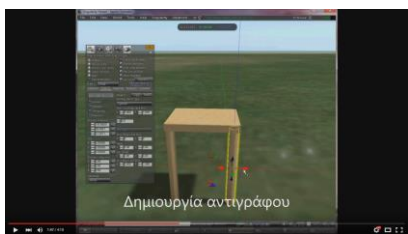
Επιλέγουμε τα δύο «μπροστινά» πόδια και με SHIFT και μετακίνηση στον άξονα Y δημιουργούμε δύο αντίγραφα τα οποία μετακινούμε στο πίσω μέρος της καρέκλας. Διορθώνουμε την τοποθέτησή τους ως προς τον Y σε σχέση με τα μπροστινά πόδια, όπως στο προηγούμενο βήμα.

Επιλέγουμε ένα από τα δύο πίσω πόδια και ρυθμίζουμε οπτικά το ύψος του χρησιμοποιώντας το πάνω στοιχείο ελέγχου. Όταν αποκτήσουν το κατάλληλο ύψος, αντιγράφουμε τις τιμές θέσης και μεγέθους για τον άξονα Z στο άλλο πίσω πόδι. Κατά συνέπεια, τα δύο πίσω πόδια έχουν το ίδιο ύψος, αλλά η βάση τους εξακολουθεί να είναι ίδια με τα δύο μπροστινά.

Τέλος, δημιουργούμε ένα ακόμη ορθογώνιο παραλληλεπίπεδο, το οποίο θα αποτελέσει την «πλάτη» της καρέκλας. Αρχικά το τοποθετούμε οπτικά ανάμεσα στα πίσω πόδια της καρέκλας και στη συνέχεια επεξεργαζόμαστε τις τιμές τοποθέτησής του για μεγαλύτερη ακρίβεια. Η τιμή θέσης του X θα πρέπει να ταυτίζεται με το κέντρο της έδρας και του Y με το κέντρο των δύο πίσω ποδιών. Όταν το σχήμα μας είναι έτοιμο, μπορούμε να επεξεργαστούμε τα επιμέρους υλικά και να ομαδοποιήσουμε τα στερεά σε ένα τελικό αντικείμενο (βλ. εικ. 8.6). Η διαδικασία δημιουργίας του σύνθετου αντικειμένου παρουσιάζεται στο βίντεο 8.2.



Εικόνα 8.6 Παράδειγμα κατασκευής σύνθετου αντικειμένου.



Σύνδεσμος:
<https://youtu.be/46sX22CFyL8>

Video 8.2 Διαδικασία κατασκευής σύνθετου αντικειμένου.

Ορισμένες καλές πρακτικές για την κατασκευή σύνθετων αντικειμένων μέσω βασικών στερεών είναι:

- Η αρχική τοποθέτηση των στοιχείων να είναι προσανατολισμένη στους άξονες για ευκολότερη τοποθέτηση και μετακίνηση των στερεών.
- Επιλέξτε κατάλληλη οπτική γωνία κατά τη διάρκεια των διεργασιών τοποθέτησης χρησιμοποιώντας τα πλήκτρα ALT και CONTROL.
- Όπου χρειάζονται πολλαπλά αντίγραφα στερεών δημιουργήστε τα με SHIFT και μετακίνηση για γρηγορότερη οπτική τοποθέτηση.
- Υπολογίστε αριθμητικά τις απαιτούμενες αποστάσεις και μεγέθη και κάντε μικροδιορθώσεις στις τιμές των αντίστοιχων χαρακτηριστικών για μεγαλύτερη ακρίβεια.
- Ενεργοποιήστε τις σκιές (Edit > Preferences > Graphics > Quality & Performance: Ultra) για καλύτερη κατανόηση του βάθους. Αλλάξτε την ώρα σε μεσημβρινή (World > Environment Settings > Midday) για να είναι κάθετες οι σκιές.

3 Σύνθετη μοντελοποίηση

3.1 Τροποποίηση μορφής βασικών στερεών

Όπως αναφέραμε και πιο πάνω, η μορφή των βασικών στερεών μπορεί να τροποποιηθεί μέσω του παράθυρου επεξεργασίας και να γίνει αρκετά πιο σύνθετη και ενδιαφέρουσα. Το χαρακτηριστικό αυτό μας επιτρέπει τελικά να δημιουργήσουμε πιο περίπλοκα και λεπτομερή μοντέλα συνδυάζοντας κατάλληλα τροποποιημένα βασικά στερεά. Το μειονέκτημα είναι πως οι τροποποιήσεις μορφής στις οποίες θα αναφερθούμε δεν υποστηρίζονται από εύχρηστα οπτικά εργαλεία, αλλά προκαλούνται μέσω αλλαγής τιμών ορισμένων χαρακτηριστικών που συνοδεύουν τα βασικά στερεά και μεταβάλλουν τη μορφή τους. Στη συνέχεια θα εξετάσουμε ένα ένα τα χαρακτηριστικά αυτά και θα δούμε την επίδρασή τους στην τελική μορφή του στερεού. Τα χαρακτηριστικά στα οποία αναφερόμαστε υπάρχουν όλα στην καρτέλα 'Object' του παράθυρου επεξεργασίας.

Το πρώτο χαρακτηριστικό που θα εξετάσουμε είναι το *building block type*, δηλαδή ο *τύπος του βασικού στερεού* ο οποίος θα χρησιμοποιηθεί ως αρχέτυπο για να δημιουργήσουμε την τελική μορφή μας. Οι τύποι που μπορούμε να επιλέξουμε είναι τα υποστηριζόμενα είδη των βασικών στερεών που αναφέραμε στην προηγούμενη ενότητα, δηλαδή: ορθογώνιο παραλληλεπίπεδο (box), κύλινδρος (cylinder), πρίσμα (prism), σφαίρα (sphere), τόρος (torus), σωληνάριο (tube), δαχτυλίδι (ring). Προφανώς, αλλάζοντας τον τύπο βασικού στερεού ενός αντικειμένου που έχετε ήδη κατασκευάσει, για παράδειγμα ενός παραλληλεπίπεδου σε σφαίρα, θα μεταβληθεί αντίστοιχα και η τελική μορφή του αντικειμένου. Αυτό που μπορούμε να αντιληφθούμε ως συνέπεια του παραπάνω είναι ότι τα πλήκτρα δημιουργίας νέων αντικειμένων επί της ουσίας είναι συντομεύσεις σε συγκεκριμένες ρυθμίσεις αντικειμένων. Μπορούμε κάλλιστα να δημιουργούμε μόνο παραλληλεπίπεδα και να τροποποιούμε τη μορφή τους χειροκίνητα μέσω των επιλογών της καρτέλας Object.

Το επόμενο χαρακτηριστικό είναι το *path cut* (αποκοπή μονοπατιού). Για το χαρακτηριστικό αυτό μπορούμε να τροποποιήσουμε δύο τιμές, την *έναρξη* (*begin*) και τον *τερματισμό* (*end*). Και οι δύο τιμές θα πρέπει να βρίσκονται μεταξύ του 0 και 1 με την τιμή του *begin* να είναι μικρότερη από αυτήν του *end*. Το αποτέλεσμα της αποκοπής μονοπατιού είναι να μην δημιουργείται ολόκληρο το βασικό στερεό αλλά ένα μόνο τμήμα του, με αποτέλεσμα να φαίνεται σαν να έχει κοπεί ένα κομμάτι σε σχήμα πίτας από αυτό (βλ. εικ. 8.7).



Εικόνα 8.7 Αποκοπή μονοπατιού σε βασικά στερεά.



Εικόνα 8.8 Δημιουργία οπής σε βασικά στερεά. Στα τελευταία τέσσερα έχει γίνει και αποκοπή μονοπατιού για να φανεί το εσωτερικό τους.

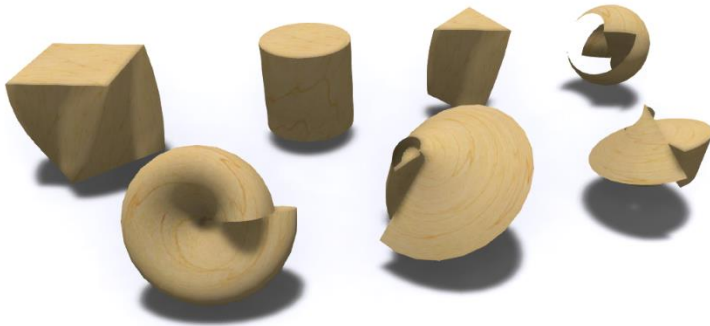
Το χαρακτηριστικό *hollow* δημιουργεί οπτικά μια οπή που διαπερνά ομοιόμορφα το βασικό στερεό σε μια διάσταση (βλ. εικ. 8.8). Η τιμή του μπορεί να αλλάξει μεταξύ 0 (καθόλου τρύπα) και 95. Κάθε βασικό στερεό έχει ένα εξ ορισμού σχήμα τρύπας ανάλογα με τη μορφή του ίδιου του στερεού. Για παράδειγμα, το παραλληλεπίπεδο έχει τετράγωνο, ο κύλινδρος δίσκο, το πρίσμα τρίγωνο κ.λπ. Αν όμως θέλουμε να αλλάξουμε το σχήμα, για παράδειγμα να κάνουμε μια κυκλική τρύπα σε έναν κύβο, μπορούμε να το μεταβάλλουμε μέσω της επιλογής *Hollow Shape* στο σχήμα της αρεσκείας μας (τετράγωνο, κύκλος ή τρίγωνο).

Η *συστροφή* (*twist*) είναι ένα χαρακτηριστικό που μπορεί να μεταβάλλει σημαντικά την τελική όψη του στερεού. Δέχεται δύο τιμές, μία για τη βάση (*begin*) και μία για την κορυφή (*end*), καθεμία από τις οποίες μπορεί να είναι στο εύρος από -180 έως 180. Το αποτέλεσμα δείχνει οπτικά σαν να έχει πιάσει και περιστρέψει κάποιος τη βάση ή και την κορυφή του αντικειμένου στις αντίστοιχες μοίρες. Το αποτέλεσμα της συστροφής δεν γίνεται ορατό στον κύλινδρο λόγω συμμετρίας, εκτός αν συνδυαστεί και με άλλες μεταβολές στο σχήμα του. Στη σφαίρα, στον τόρο, στο σωληνάριο και στο δαχτυλίδι οδηγεί σε ιδιαίτερα ακανόνιστες και περίεργες γεωμετρίες (βλ. εικ. 8.9).

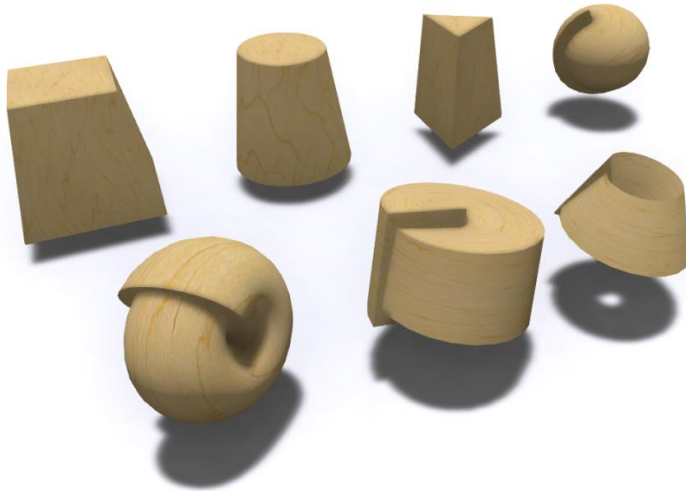
Το χαρακτηριστικό *taper* μεταβάλλει το μέγεθος της κορυφής του αντικειμένου σε σχέση με τη βάση, δίνοντας ένα κωνικό σχήμα. Σε σχέση με το χαρακτηριστικό αυτό μπορεί να μεταβάλλει κάποιος τις τιμές του X και του Y στο εύρος μεταξύ -1 και 1. Όσο μεγαλύτερη είναι η τιμή τόσο μικραίνει το μέγεθος της κορυφής του αντικειμένου στην αντίστοιχη διάσταση. Αν η τιμή είναι αρνητική, μεταβάλλεται η βάση. Και στην περίπτωση του *Taper*, στερεά όπως ο τόρος, το σωληνάριο και το δαχτυλίδι στα οποία τα δύο άκρα (κορυφή και βάση) εφάπτονται, το αποτέλεσμα οδηγεί σε διάσπαση της συνοχής του σχήματος (βλ. εικ. 8.9).

Το χαρακτηριστικό *Top Shear* είναι παραπλήσιο με το *Taper*, μόνο που αντί να αλλάζει το μέγεθος μετατοπίζεται οριζόντια η κορυφή σε σχέση με τη βάση. Αν για παράδειγμα εφαρμοστεί σε ένα

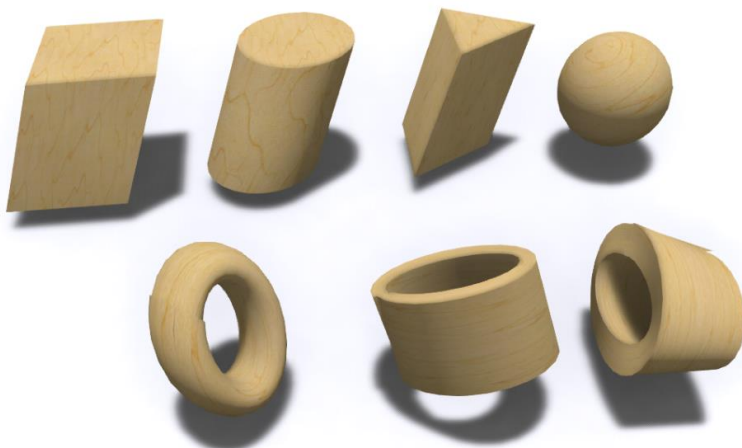
παραλληλεπίπεδο, οι γωνίες του παύουν να είναι ορθές. Δέχεται και αυτό τιμές στα X και Y μεταξύ -0.5 και 0.5 και προκαλεί οριζόντια μετατόπιση στην αντίστοιχη διάσταση (βλ. εικ. 8.10).



Εικόνα 8.9 Συστροφή βασικών στερεών.



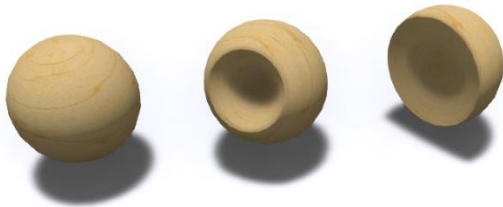
Εικόνα 8.10 Μεταβολή μεγέθους κορυφής (*taper*) βασικών στερεών.



Εικόνα 8.11 Οριζόντια μετατόπιση κορυφής (*top shear*) βασικών στερεών.

Το χαρακτηριστικό slice begin and end υπάρχει μόνο στο παραλληλεπίπεδο, τον κύλινδρο και το πρίσμα. Η εφαρμογή του οδηγεί στην αποκοπή ενός τμήματος της γεωμετρίας του σχήματος. Όσο μεγαλύτερη είναι η τιμή του Begin (από 0 μέχρι 1), τόσο μεγαλύτερο μέρος αποκόπτεται από τη βάση προς τα πάνω. Αντίστοιχα, αλλάζοντας την τιμή του End, η αποκοπή εφαρμόζεται από την κορυφή προς τα κάτω.

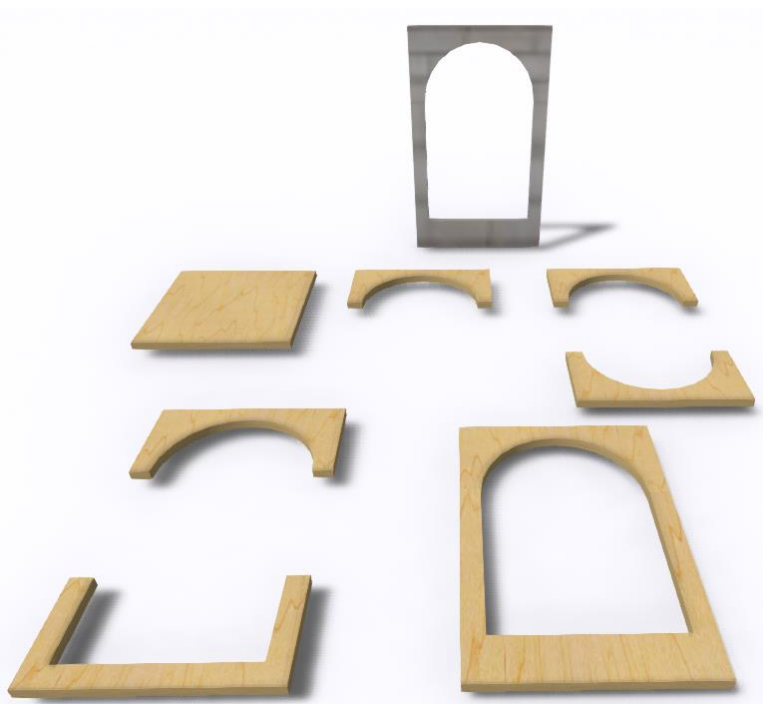
Το dimple είναι ένα χαρακτηριστικό που υπάρχει μόνο στη σφαίρα και οδηγεί στην αφαίρεση από τη γεωμετρία της ενός κώνου από το κέντρο της σφαίρας μέχρι την επιφάνειά της. Αυξάνοντας την τιμή του begin, αυξάνεται το μέγεθος ενός κώνου που αποκόπτεται από το κάτω μέρος της σφαίρας, και του end αντίστοιχα από το πάνω (βλ. εικ. 8.11).



Εικόνα 8.12 Παραμόρφωση σφαίρας με αλλαγή της τιμής του dimple.

Τέλος, η σφαίρα, ο τόρος, το σωληνάριο και το δαχτυλίδι έχουν ορισμένα ακόμη χαρακτηριστικά, όπως το revolutions, που επεκτείνει τη γεωμετρία τους δίνοντάς τους σπειροειδή όψη, και το hole size, που μεταβάλλει το πάχος τους και τη διάμετρο της οπής τους.

Παρά το γεγονός ότι τα παραπάνω χαρακτηριστικά μπορούν να μετασχηματίσουν τα βασικά στερεά σε νέες, πιο περίπλοκες γεωμετρίες, απαιτούνται αρκετές δοκιμές και εξάσκηση μέχρι να μπορέσει κάποιος να τα εκμεταλλευτεί παραγωγικά για την κατασκευή των επιθυμητών μορφών. Αν επιθυμείτε να αποκτήσετε τις παραπάνω δεξιότητες, θα είναι χρήσιμο να πειραματιστείτε αρκετά με διάφορα βασικά στερεά και συνδυασμούς τιμών των παραπάνω χαρακτηριστικών, ώστε να κατανοήσετε καλύτερα την επίδρασή τους στο τελικό αποτέλεσμα. Επιπλέον, μπορείτε να επισκεφτείτε κατασκευασμένους κόσμους και να κάνετε Edit σε αντικείμενα που σας κίνησαν το ενδιαφέρον. Δεν θα έχετε, προφανώς, τη δυνατότητα να τα επεξεργαστείτε, αλλά θα μπορείτε να δείτε τα επιμέρους αντικείμενα από τα οποία είναι κατασκευασμένα και ενδεχομένως να καταλάβετε τον τρόπο με τον οποίο αυτά έχουν δημιουργηθεί.

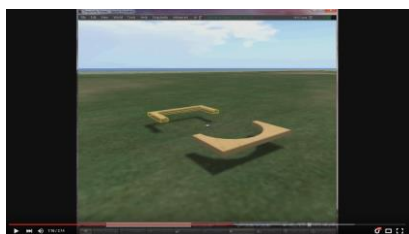


Εικόνα 8.13 Κατασκευή παραθύρου από δύο βασικά στερεά.

Ας δούμε ορισμένα παραδείγματα μοντελοποίησης με τη χρήση των παραπάνω. Έστω ότι θέλουμε να κατασκευάσουμε ένα παράθυρο με ημικυκλική κορυφή. Το παράθυρο αυτό θα μπορούσαμε να το φανταστούμε ως δύο τμήματα: το πάνω τμήμα είναι ένα παραλληλεπίπεδο από το οποίο έχει αποκοπεί ένα ημικύκλιο και το κάτω ένα άλλο παραλληλεπίπεδο από το οποίο έχει αποκοπεί ένα ορθογώνιο. Με βάση την παρατήρηση αυτή, μπορούμε να κατασκευάσουμε το αντικείμενό μας από δύο βασικά στερεά εκμεταλλευόμενοι το χαρακτηριστικό 'Hollow', που θα μας δημιουργήσει τις κατάλληλες εσωτερικές αποκοπές, αλλά και το 'Path Cut', που θα απομονώσει ένα μόνο μέρος του στερεού. Τα βήματα είναι τα εξής:

1. Δημιουργούμε ένα Box δίνοντας στις οριζόντιες διαστάσεις του (X και Y) το επιθυμητό πλάτος του παράθυρου, π.χ. (2.0, 2.0, 0.1).
2. Ορίζουμε την τιμή του 'Hollow' στο 80 και επιλέγουμε ως σχήμα τον κύκλο (Hollow Shape: Circle).
3. Τροποποιούμε τις τιμές 'Begin' και 'End' του 'Path Cut' έτσι ώστε να μείνει μόνο η μισή γεωμετρία με αποκοπή του ημικυκλίου (begin: 0.375, end: 0.875).
4. Δημιουργούμε αντίγραφο του αντικειμένου και το μετακινούμε στον άξονα Y. Στη συνέχεια αλλάζουμε τον προσανατολισμό του αντιγράφου δίνοντάς του περιστροφή 180° στον άξονα Z.
5. Αλλάζουμε το σχήμα του Hollow στο αντίγραφο σε τετράγωνο (Hollow Shape: Square). Αλλάζουμε την κλίμακα και την τοποθέτηση του αντιγράφου στον άξονα Y έτσι ώστε τα δύο αντικείμενα να εφάπτονται και το παράθυρο να έχει το επιθυμητό ύψος.
6. Ενοποιούμε τα δύο αντικείμενα σε ένα και αλλάζουμε τον προσανατολισμό (περιστροφή 90° μοιρών στον άξονα X), ώστε το παράθυρό μας να είναι κάθετο.

Η παραπάνω διαδικασία απεικονίζεται στην εικόνα 8.13 και παρουσιάζεται στο βίντεο 8.3.



Σύνδεσμος:
<https://youtu.be/Jc5eVuZDgkE>

Video 8.3 Κατασκευή παραθύρου με ημικυκλική κορυφή.



Εικόνα 8.14 Λάμπα γραφείου κατασκευασμένη από ένα βασικό στερεό.

Υπάρχουν δημιουργοί οι οποίοι καταφέρνουν με συνδυασμό τροποποιήσεων στα χαρακτηριστικά των αντικειμένων να κατασκευάσουν σύνθετες μορφές με ένα μόνο βασικό στερεό. Σε ένα τέτοιο παράδειγμα θα δούμε πώς να κατασκευάσουμε μια λάμπα γραφείου από ένα σωληνάριο (πηγή: <http://ayumicassini.blogspot.com/>).

1. Δημιουργήστε ένα σωληνάριο (tube).
2. Αλλάξτε το μέγεθος σε (0.5, 0.5, 0.5).
3. Αλλάξτε την τιμή του Hollow σε 85.0.

4. Στο Hole Size θέστε τις τιμές $X=1.00$ και $Y=0.50$.
5. Στο Profile Cut θέστε τις τιμές $B=0.250$ και $E=0.850$.

Το εντυπωσιακό ομολογουμένως αποτέλεσμα φαίνεται στην εικόνα 8.14.

3.2 Παραμορφώσιμα αντικείμενα

Εκτός από σύνθετες μορφές, στα βασικά στερεά μπορούμε να προσδώσουμε και ελαστικότητα, με αποτέλεσμα να παραμορφώνεται η γεωμετρία τους κατά τη διάρκεια της κίνησής τους. Κάτι τέτοιο έχει νόημα όταν θέλουμε να προσομοιώσουμε τη συμπεριφορά μαλακών αντικειμένων και υφασμάτων. Αρκετά από τα ρούχα και τις κομμώσεις μαλλιών που κατασκευάζονται και διανέμονται ελεύθερα ή πωλούνται στους κόσμους του Second Life βασίζονται σε ελαστικά στερεά.

Για να προσδώσετε ελαστικότητα σε ένα αντικείμενο θα πρέπει σε κατάσταση επεξεργασίας να μεταβείτε στην καρτέλα Features και να επιλέξετε Flexible Path. Αν το δοκιμάσετε, για παράδειγμα, σε έναν απλό κύβο και στη συνέχεια τον μετακινήσετε σε κάποιον από τους οριζόντιους άξονες, θα παρατηρήσετε ότι, ενώ η επιφάνεια της βάσης του παραμένει σταθερή, η επιφάνεια της κορυφής του παραμορφώνεται συμπαρασύροντας την υπόλοιπη γεωμετρία του σχήματος. Ο τρόπος με τον οποίο παραμορφώνεται ο κύβος δίνει οπτικά την αίσθηση της ελαστικότητας που θα είχε ένα μαλακό αντικείμενο (π.χ. ζελέ) αν το μετακινούσαμε απότομα.

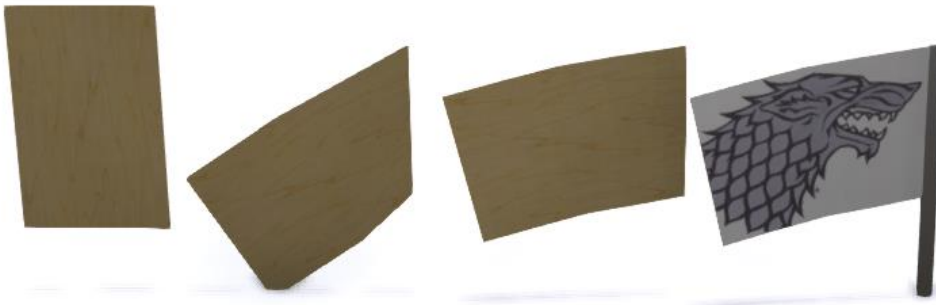
Ας δούμε τις παραμέτρους που μπορούμε να ρυθμίσουμε σε σχέση με την ελαστικότητα:

- Softness: πόσο «μαλακό» θα είναι το αντικείμενο, δηλαδή πόσο εύκολα θα υφίσταται παραμορφώσεις,
- Gravity: σε ποιο βαθμό θα υφίσταται την επίδραση της βαρύτητας,
- Drag: πόσο αργή θα είναι η κίνησή του λόγω παραμόρφωσης,
- Wind: σε ποιο βαθμό θα επηρεάζεται από τον άνεμο της περιοχής,
- Tension: πόσο έντονα θα αντιδρά στην κίνηση,
- Force X, Force Y, Force Z: αν θέλουμε να ασκείται κάποια σταθερή δύναμη στο αντικείμενο σε έναν ή περισσότερους από τους τρεις άξονες του τοπικού συστήματος συντεταγμένων.

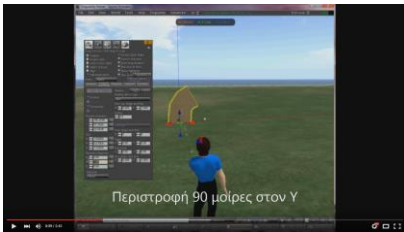
Μπορούμε να εξετάσουμε την πρακτική χρήση των παραπάνω σε ένα απλό παράδειγμα. Θα κατασκευάσουμε μία σημαία.

1. Αρχικά δημιουργούμε ένα αντικείμενο τύπου Box με διαστάσεις (1.0, 0.01, 1.5). Επειδή η ελαστικότητα δημιουργείται εξ ορισμού από τη βάση προς την κορυφή, προσανατολίζουμε αρχικά τη σημαία μας κάθετα.
2. Επιλέγουμε το Flexible Path για να γίνει ελαστική.
3. Για να μελετήσουμε καλύτερα τη συμπεριφορά της σημαίας μας και να προβούμε στις απαραίτητες διορθώσεις, θα πρέπει να την περιστρέψουμε κατάλληλα. Την περιστρέφουμε κατά 90° στον άξονα Y.
4. Μετακινούμε οριζόντια τη σημαία και κάνουμε διορθώσεις στις παραμέτρους ελαστικότητας. Καταλήγουμε στις παρακάτω αλλαγές:
 - Το Wind αλλάζει σε 10.0 ώστε να επηρεάζεται σημαντικά από τη διεύθυνση και την ταχύτητα του ανέμου.
 - Το Drag αλλάζει σε 1.0 και το Tension σε 2.0, ώστε να έχει πιο γρήγορη και άμεση απόκριση στις αλλαγές.
5. Το τελικό στάδιο είναι να προσθέσουμε κατάλληλη υφή στη σημαία μας, να δημιουργήσουμε έναν επιπλέον κύλινδρο για κοντάρι και να ενοποιήσουμε τα δύο αντικείμενα σε μία ομάδα.

Τα βήματα και το τελικό αποτέλεσμα των παραπάνω ενεργειών απεικονίζονται στην εικόνα 8.15 και παρουσιάζονται στο βίντεο 8.4.



Εικόνα 8.15 Κατασκευή μιας σημαίας ως παραμορφώσιμου αντικειμένου.



Σύνδεσμος:
<https://youtu.be/Ak1sO1o1GL0>

Video 8.4 Κατασκευή σημαίας ως παραμορφώσιμου αντικειμένου.

3.3 Εισαγωγή εξωτερικών μοντέλων

Είδαμε ότι μπορούμε να δημιουργήσουμε αρκετά ενδιαφέρουσες μορφές τροποποιώντας και συνδυάζοντας βασικά στερεά. Είναι πολύ δύσκολο όμως να δημιουργήσουμε οποιαδήποτε γεωμετρία μπορούμε να φανταστούμε, για παράδειγμα ένα ανθρώπινο πρόσωπο. Το OpenSimulator μάς προσφέρει δύο λύσεις για το πρόβλημα αυτό: τα «γλυπτά» στερεά (*sculpted prims*) και την εισαγωγή πολυγωνικών πλεγμάτων.

Η ιδέα των γλυπτών στερεών είναι εξαιρετικά απλή αλλά και ιδιαίτερα αποδοτική για την παραγωγή αυθαίρετων γεωμετριών. Ξεκινάμε από ένα βασικό στερεό, π.χ. σφαίρα, και εφαρμόζουμε σε αυτό μια ειδική υφή, η οποία αντί για χρώμα περιέχει γεωμετρικές μετατοπίσεις των σημείων στα οποία εφαρμόζεται. Έτσι, για παράδειγμα, ξεκινώντας από μία σφαίρα μπορούμε με τα κατάλληλα βαθουλώματα και εξογκώματα να δημιουργήσουμε ένα μήλο ή ακόμα και ένα ανθρώπινο πρόσωπο.



Εικόνα 8.16 Παράδειγμα «γλυπτού» στερεού (*sculpted prim*).

Η δημιουργία αυτής της «ειδικής» υφής, που ονομάζεται *υφή γλυπτού (sculpt texture)*, γίνεται σε κάποιο εξωτερικό πρόγραμμα τρισδιάστατης μοντελοποίησης με τη χρήση ειδικών εργαλείων. Υπάρχουν ειδικά templates και plug-ins για διάφορα προγράμματα μοντελοποίησης, συμπεριλαμβανομένης της δημοφιλούς εφαρμογής ανοικτού λογισμικού Blender, για τη δημιουργία γλυπτών στερεών και την εξαγωγή της υφής τους. Ο σχεδιαστής ξεκινάει από ένα βασικό στερεό και, χωρίς να προσθέτει ή να αφαιρεί κορυφές, επιλέγει κορυφές και τις μετατοπίζει μετατρέποντας το σχήμα του στερεού στη μορφή της αρεσκείας του. Στη συνέχεια, το πρόγραμμα κάνει εξαγωγή της κατάλληλης υφής και ο σχεδιαστής την ανεβάζει στο OpenSimulator και την αποδίδει ως υφή γλυπτού στο στερεό.

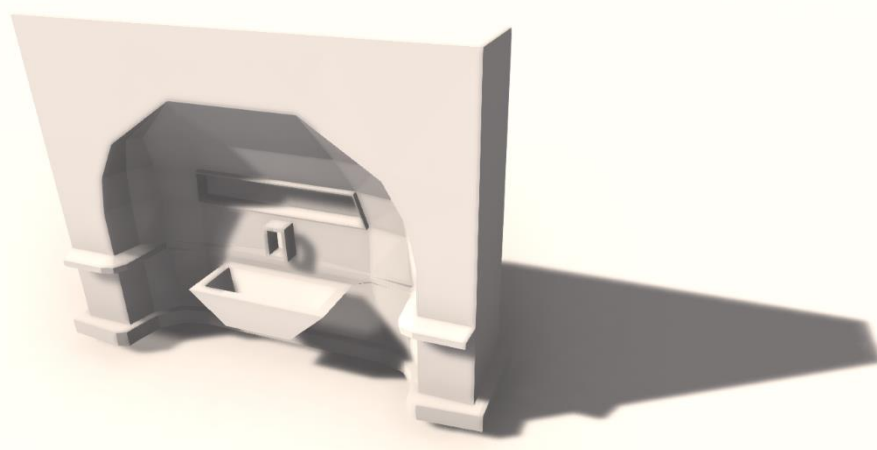
Αν θέλετε να δείτε ένα παράδειγμα γλυπτού στερεού, δημιουργήστε ένα βασικό στερεό και από την καρτέλα 'Object' αλλάξτε το 'Building Block Type' σε 'Sculpted'. Θα δείτε μια σφαίρα που έχει μετατραπεί σε μήλο μέσω μιας υφής γλυπτού που έχει αποδοθεί σε αυτήν. Στην εικόνα 8.16 φαίνεται ένα δαγκωμένο μήλο που έχει κατασκευαστεί ως γλυπτό στερεό και του έχει αποδοθεί κατάλληλη υφή.

Τέλος, το OpenSimulator μάς δίνει τη δυνατότητα εισαγωγής στον κόσμο τρισδιάστατων αντικειμένων σε μορφή πολυγωνικού πλέγματος. Τα μοντέλα που θα εισαχθούν στο περιβάλλον θα πρέπει να είναι υποχρεωτικά αποθηκευμένα σε μορφή COLLADA (κατάληξη .dae). Τα περισσότερα από τα προγράμματα τρισδιάστατης μοντελοποίησης προσφέρουν τη δυνατότητα εξαγωγής μοντέλων στη μορφή αυτήν, ενώ υπάρχουν και εφαρμογές ειδικά σχεδιασμένες για τη σωστή μετατροπή μεταξύ μορφότυπων τρισδιάστατων μοντέλων, όπως το MeshLab.

Για να εισάγουμε ένα μοντέλο στο OpenSimulator θα χρησιμοποιήσουμε την επιλογή File > Upload Mesh. Αφού επιλέξουμε το αρχείο του μοντέλου, το περιβάλλον μάς εμφανίζει έναν διάλογο με ρυθμίσεις σχετικά με τη διαδικασία εισαγωγής. Η πρώτη καρτέλα σχετίζεται με τα *επίπεδα λεπτομέρειας (Levels of Detail)* που θα δημιουργηθούν για το μοντέλο μας, η δεύτερη σχετικά με τη *φυσική συμπεριφορά* του (στην περίπτωση που επιλέξουμε να είναι φυσικό αντικείμενο) και στην τρίτη περιλαμβάνονται επιπλέον επιλογές, όπως η κλίμακα και η εισαγωγή των υφών.

Αν δεν έχει προκύψει κάποιο πρόβλημα στη φόρτωση του μοντέλου, θα πρέπει να μπορούμε να το παρατηρήσουμε στο παράθυρο Preview. Αλλάζοντας τις επιλογές από High σε Medium, Low ή Lowest στο μενού ακριβώς από κάτω, μπορούμε να εξετάσουμε τα επίπεδα λεπτομέρειας που δημιουργήθηκαν αυτόματα. Αν δεν μας ικανοποιούν, μπορούμε να παραλείψουμε κάποια από αυτά ή να ανεβάσουμε άλλο χαμηλότερης λεπτομέρειας πολυγωνικό μοντέλο.

Όταν έχουμε ολοκληρώσει τις ρυθμίσεις, επιλέγουμε 'Calculate weights & fee' και στη συνέχεια 'Upload'. Τότε το μοντέλο ανεβαίνει στον διακομιστή ως αντικείμενο του κόσμου και δημιουργείται στιγμιότυπό του στο αποθετήριο μας, στον φάκελο Objects. Όπως συμβαίνει και με οποιοδήποτε άλλο αντικείμενο, μπορούμε να το εισάγουμε στο περιβάλλον μεταφέροντάς το από το αποθετήριο στο έδαφος του κόσμου. Στην εικόνα 8.17 φαίνεται ένα παράδειγμα πολυγωνικού μοντέλου που έχει ανέβει στον κόσμο.



Εικόνα 8.17 Εισαγωγή πολυγωνικού μοντέλου στον εικονικό κόσμο

Στο σημείο αυτό οφείλουμε να επισημάνουμε ότι η εισαγωγή πολυγωνικών πλεγμάτων στο περιβάλλον θα πρέπει να είναι προσεκτική και περιορισμένη. Τα πολυγωνικά πλέγματα, και ιδιαίτερα αυτά με πολύ μεγάλο αριθμό τριγώνων, επιβαρύνουν σημαντικά τόσο τον διακομιστή όσο και τη διαδικασία απεικόνισης που

εκτελείται σε επίπεδο πελάτη. Αν σε έναν κόσμο υπάρχει μεγάλος αριθμός από πολυγωνικά μοντέλα, η απόδοση του κόσμου θα πέσει λόγω επιβάρυνσης τόσο της διαδικτυακής επικοινωνίας όσο και του επεξεργαστή γραφικών των υπολογιστικών συστημάτων. Συνεπώς, χρήστες οι οποίοι συνδέονται με υπολογιστικά συστήματα χαμηλότερων δυνατοτήτων θα έχουν μεγαλύτερη δυσκολία να πλοηγηθούν και να αλληλεπιδράσουν με το περιβάλλον. Τέλος, τα παλιότερα προγράμματα σύνδεσης, όπως για παράδειγμα το *Impudence*, δεν υποστηρίζουν την απεικόνιση πολυγωνικών πλεγμάτων, με αποτέλεσμα οι χρήστες των προγραμμάτων αυτών να μην μπορούν να τα δουν μέσα στον κόσμο.

4 Διαμόρφωση χώρου

Σε αυτήν την ενότητα θα εξετάσουμε τις διάφορες ρυθμίσεις και εργαλεία που μας προσφέρονται στο *OpenSimulator* για τη διαμόρφωση του χώρου. Θα δούμε πώς μπορούμε να αλλάξουμε τη μορφή του αναγλύφου εδάφους, να προσθέσουμε δέντρα και φυτά, να τροποποιήσουμε τη θέση του ήλιου και τον κύκλο ημέρας-νύχτας και να δημιουργήσουμε νέες πηγές φωτισμού.

4.1 Ανάγλυφο εδάφους

Όπως αναφέραμε και στο προηγούμενο κεφάλαιο, ο χειριστής έχει τη δυνατότητα να ανεβάσει έτοιμο ανάγλυφο εδάφους (*terrain*), το οποίο περιγράφεται σε κάποιο αρχείο εικόνας, καθώς και να κάνει ορισμένες βασικές τροποποιήσεις στη γεωμετρία του μέσω των εντολών του διακομιστή. Σε ορισμένες περιπτώσεις όμως είναι προτιμότερο οι τροποποιήσεις να γίνονται εντός του κόσμου με τη χρήση των οπτικών εργαλείων που προσφέρονται στο *OpenSimulator*, ώστε να προσαρμόζεται το αποτέλεσμα στις ανάγκες του κόσμου.

Η προσαρμογή του αναγλύφου μέσω του προγράμματος σύνδεσης γίνεται από το παράθυρο επεξεργασίας επιλέγοντας την καρτέλα με εικονίδιο έναν εκσκαφέα και τίτλο *Land*. Στην καρτέλα αυτήν μπορεί να βρει κανείς τις παρακάτω επιλογές εργαλείων:

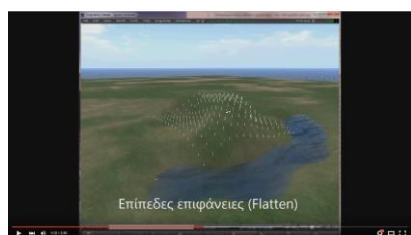
- Δημιουργία επίπεδης επιφάνειας (*flatten*): μειώνονται οι υψομετρικές διαφορές μέχρι η επιφάνεια να γίνει επίπεδη.
- Ανύψωση (*raise*): αυξάνεται το ύψος του εδάφους δημιουργώντας λόφους.
- Μείωση ύψους (*lower*): μειώνεται το ύψος του εδάφους δημιουργώντας κοιλάματα. Αν το έδαφος βρεθεί σε χαμηλότερο ύψος από την επιφάνεια του νερού (εξ ορισμού 20 μέτρα), τότε σχηματίζονται λίμνες.
- Εξομαλύνση (*smooth*): το ανάγλυφο του εδάφους εξομαλύνεται και περιορίζονται οι απότομες γωνίες και αλλαγές κλίσης.
- Εκτράχυνση (*roughen*): το έδαφος γίνεται περισσότερο τραχύ.
- Ανάκληση (*revert*): το έδαφος επιστρέφει στην προηγούμενη κατάσταση που βρισκόταν πριν τις αλλαγές.

Πριν από την εφαρμογή των εργαλείων αυτών είναι καλό να έχετε επιλέξει κατάλληλη οπτική γωνία, ώστε να μπορείτε να παρατηρήσετε καλύτερα τις αλλαγές που συμβαίνουν στο έδαφος. Επιπλέον, στην καρτέλα επεξεργασίας του εδάφους μπορείτε να προσαρμόσετε την *έκταση* (*size*) και την *ισχύ* (*strength*) του κάθε εργαλείου. Για να χρησιμοποιήσετε ένα εργαλείο, αρκεί να μεταφέρετε τον δείκτη του ποντικιού στο σημείο του εδάφους που θέλετε να εφαρμοστεί και να κρατήσετε πατημένο το αριστερό πλήκτρο. Όσο περισσότερη ώρα το κρατάτε πατημένο τόσο πιο ισχυρή θα είναι η εφαρμογή του εργαλείου στο σημείο αυτό. Δοκιμάστε αρχικά να κάνετε μικρές μεταβολές, μέχρι να αποκτήσετε τη δεξιότητα αποτελεσματικής χρήσης των εργαλείων.

Αν θέλετε να αλλάξετε την υφή του εδάφους, μπορείτε να το κάνετε επιλέγοντας από το μενού *World > Region / Estate* και στο παράθυρο που θα ανοίξει την καρτέλα *Terrain*. Όπως θα παρατηρήσετε, έχουν δηλωθεί τέσσερις εξ ορισμού υφές για το έδαφος, ανάλογα με το ύψος. Από το χαμηλότερο στο ψηλότερο αυτές είναι: χώμα, γρασίδι, βράχος και λευκός βράχος. Αν δοκιμάσετε να αλλάξετε μία από αυτές και να πατήσετε *Apply*, η όψη του εδάφους θα αλλάξει ανάλογα. Το περιβάλλον ρυθμίζει και προσαρμόζει την υφή ανάλογα με το ύψος, κάποιες φορές συνδυάζοντας δύο εικόνες. Μπορείτε λοιπόν να ανεβάσετε και να ορίσετε τις δικές σας υφές μεταβάλλοντας έτσι τον τρόπο που ζωγραφίζεται το έδαφος. Στην ίδια καρτέλα μπορείτε να προσαρμόσετε

τα όρια ελάχιστου (low) και μέγιστου (high) ύψους για τα τέσσερα άκρα της περιοχής σας (northwest, northeast, southwest, southeast). Αν, για παράδειγμα, μειώσετε την τιμή του μέγιστου, τότε η υφή του λευκού βράχου θα αρχίσει να εμφανίζεται σε χαμηλότερα ύψη.

Η χρήση των εργαλείων διαμόρφωσης του εδάφους παρουσιάζεται στο βίντεο 8.5.



Σύνδεσμος:
<https://youtu.be/EneKjXzwdt8>

Video 8.5 Επίδειξη εργαλείων διαμόρφωσης εδάφους.

4.2 Βλάστηση

Το OpenSimulator μάς δίνει επιπλέον τη δυνατότητα να προσθέσουμε δέντρα και φυτά στον κόσμο. Αυτό μπορούμε να το κάνουμε μέσω της επιλογής Build αν, αντί για κάποιο βασικό στερεό, επιλέξουμε μία από τις δύο τελευταίες επιλογές: δέντρο (tree) ή γρασίδι (grass). Αφού κάνουμε κλικ στην επιλογή του δέντρου, μπορούμε στη συνέχεια να επιλέξουμε τον τύπο του δέντρου που θέλουμε να προστεθεί. Υπάρχει ένας μεγάλος αριθμός από έτοιμα δέντρα που μπορούμε να προσθέσουμε στο περιβάλλον, ενώ μπορούμε να διαλέξουμε και την επιλογή 'Random' για να προστεθεί τυχαία ένα από αυτά. Κάνοντας κλικ πάνω στο έδαφος, το δέντρο μας προστίθεται στη θέση αυτή. Στη συνέχεια, ενόσω βρισκόμαστε σε κατάσταση επεξεργασίας, μπορούμε να αλλάξουμε τη θέση, τον προσανατολισμό και το μέγεθός του. Το ίδιο ακριβώς συμβαίνει με την επιλογή του γρασιδιού. Υπάρχουν ορισμένοι τύποι γρασιδιού που μπορούμε να επιλέξουμε, και κάνοντας κλικ σε κάποιο σημείο στο έδαφος δημιουργούνται διάσπαρτα τμήματα γρασιδιού γύρω από αυτό. Αν θέλουμε να πολλαπλασιάσουμε τα δέντρα ή το γρασίδι σε μια περιοχή, ένας εύκολος τρόπος είναι να δημιουργούμε αντίγραφα (με shift και κλικ) και να τα μετακινούμε σε γειτονικές εστίες.

4.3 Κύκλος ημέρας και φωτισμός

Όπως έχουμε αναφέρει και σε προηγούμενη ενότητα, το OpenSimulator ακολουθεί έναν κύκλο ημέρας-νύχτας διάρκειας τεσσάρων ωρών. Αυτό σημαίνει ότι με την πάροδο του χρόνου η θέση του ήλιου μετατοπίζεται, μεταβάλλεται η όψη του ουράνιου θόλου και αλλάζει αντίστοιχα ο φωτισμός του περιβάλλοντος. Κάτι τέτοιο μπορεί να είναι πρόβλημα σε περιοχές που δεν είναι επαρκώς φωτισμένες όταν το περιβάλλον αλλάζει σε νύχτα. Αν για κάποιον λόγο θέλετε να τροποποιήσετε τις ρυθμίσεις κίνησης του ήλιου, μπορείτε να μεταβείτε στο μενού World > Region / Estate και να επιλέξετε την καρτέλα Estate. Εκεί μπορείτε να απενεργοποιήσετε την επιλογή Use Global Time και να επιλέξετε Fixed Sun. Στη συνέχεια, μπορείτε να ρυθμίσετε την επιθυμητή θέση του ήλιου στο στοιχείο ελέγχου Phase που υπάρχει ακριβώς από κάτω μετακινώντας ανάλογα τον δείκτη. Ο κόσμος θα παραμείνει μόνιμα σε αυτήν την κατάσταση. Αν θέλετε να πειραματιστείτε περισσότερο με την απεικόνιση του ουρανού και της θάλασσας και με τον κύκλο ημέρας-νύχτας, μπορείτε να επιλέξετε από το μενού World > Environment Settings τις επιλογές Environment Editor και Day Cycle Editor και να τροποποιήσετε τις σχετικές παραμέτρους.

Τέλος, σε περιπτώσεις που υπάρχουν τμήματα της σκηνής που δεν φωτίζονται επαρκώς από τον ήλιο, είτε λόγω της ώρας της ημέρας είτε λόγω σκίασης, είναι απαραίτητη η προσθήκη κατάλληλων φωτών στον κόσμο.

Για να προσθέσουμε ένα φως στον κόσμο, αρκεί να κατασκευάσουμε ένα βασικό στερεό και να δηλώσουμε ότι αυτό θα λειτουργεί ως φως. Αυτό μπορεί να γίνει μέσω της καρτέλας Features, όπου θα πρέπει να ενεργοποιήσουμε την επιλογή Light. Το αντικείμενο θα συμπεριφέρεται σαν σημειακή πηγή φωτός. Μπορούμε να ρυθμίσουμε το χρώμα του (color), την έντασή του (intensity), την ακτίνα σε μέτρα (radius) και τον βαθμό εξασθένησης (falloff). Για να αντιληφθούμε καλύτερα την επίδραση του φωτός στη σκηνή είναι προτιμότερο να αλλάξουμε προσωρινά τη θέση του ήλιου σε μεσάνυχτα (World > Environment Settings > Midnight), ώστε να μην φωτίζεται καθόλου η σκηνή από τον ήλιο (εικ. 8.18). Όταν ολοκληρώσουμε την

τοποθέτηση των φώτων, μπορούμε να επιστρέψουμε στον κανονικό φωτισμό της σκηνής με την επιλογή World > Settings > Revert to Region Default. Αν θέλουμε το στερεό που λειτουργεί ως φως να μην είναι ορατό, μπορούμε να το αποκρύψουμε αλλάζοντας τον βαθμό διαφάνειάς του (Alpha) σε 100% (καρτέλα Texture).



Εικόνα 8.18 Φωτεινό αντικείμενο σε νυχτερινό τοπίο.

Τα φώτα που προστίθενται ως αντικείμενα στον κόσμο, σε αντίθεση με το φως του ήλιου, δεν δημιουργούν σκιές. Επηρεάζουν μόνο τη φωτοσκίαση των αντικειμένων που βρίσκονται στην ακτίνα τους.

5 Ενσωματώσεις και HUD

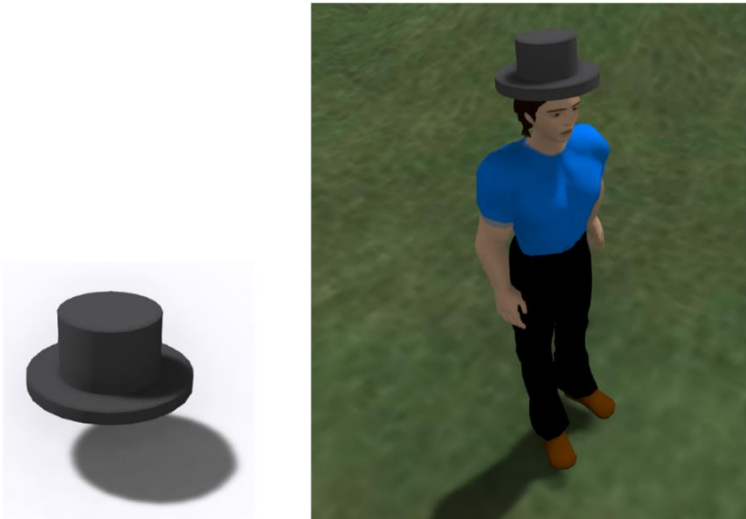
Τα αντικείμενα που κατασκευάζουμε στο περιβάλλον μπορούμε να τα προσθέσουμε στις ενσωματώσεις μας ως «ενσωματώσεις» (*attachments*). Κατά συνέπεια, μπορούμε να κατασκευάσουμε αντικείμενα-εργαλεία τα οποία οι χαρακτήρες μας κρατούν στα χέρια τους: αντικείμενα, ρούχα ή αξεσουάρ που οι χαρακτήρες μας μπορούν να φορέσουν κ.λπ.

Αν κάνετε δεξί κλικ σε ένα αντικείμενο που σας ανήκει και επιλέξετε More > Attach, θα παρατηρήσετε να αναγράφεται ένας αριθμός από πιθανές θέσεις ενσωμάτωσης. Οι αρχικές επιλογές είναι: κεφάλι, αριστερό/δεξί χέρι, αριστερή/δεξιά παλάμη, αριστερό/δεξί πόδι και μέση. Σε καθεμία από αυτές τις επιλογές εμφανίζονται περαιτέρω υπο-επιλογές με μεγαλύτερη ακρίβεια ως προς το σημείο τοποθέτησης.

Αν επιλέξουμε την ενσωμάτωση ενός αντικειμένου σε κάποιο σημείο του σώματος της ενσάρκωσης, αυτό που θα συμβεί πρακτικά είναι ότι το αντικείμενο θα κληρονομήσει τον μετασχηματισμό του αντίστοιχου σημείου του σώματος και θα ακολουθεί την κίνησή του. Βεβαίως, η τοποθέτηση του αντικειμένου ως προς το σώμα δεν θα είναι πάντα η επιθυμητή. Μπορούμε όμως να επεξεργαστούμε τη θέση και τον προσανατολισμό του αντικειμένου ενόσω αυτό είναι ενσωματωμένο στην ενσάρκωση για να βελτιώσουμε την τοποθέτηση.

Ας δούμε ένα απλό παράδειγμα: Θα κατασκευάσουμε ένα παραδοσιακό πλατύγυρο καπέλο με δύο κυλίνδρους και θα προσπαθήσουμε να το φορέσουμε στην ενσάρκωσή μας.

Αρχικά κατασκευάζουμε το καπέλο με έναν λεπτό και φαρδύ και έναν στενότερο και ψηλό κύλινδρο, όπως στην εικόνα 8.19. Τους ομαδοποιούμε, αφαιρούμε την υφή και θέτουμε ένα σκούρο χρώμα συνολικά στο αντικείμενό μας.



Εικόνα 8.19 Κατασκευή αντικειμένου και ενσωμάτωσή του στην ενσάρκωση

Στη συνέχεια δοκιμάζουμε να το ενσωματώσουμε στον χαρακτήρα μας. Κάνουμε δεξί κλικ στο αντικείμενο και More > Attach > Head > Skull, ώστε να ακολουθεί τη συνολική κίνηση του κρανίου. Παρατηρούμε ότι το καπέλο έχει τοποθετηθεί πράγματι πάνω στο κεφάλι του χαρακτήρα μας. Αν δοκιμάσουμε να κινήσουμε τον χαρακτήρα, θα το δούμε να ακολουθεί την κίνησή του.

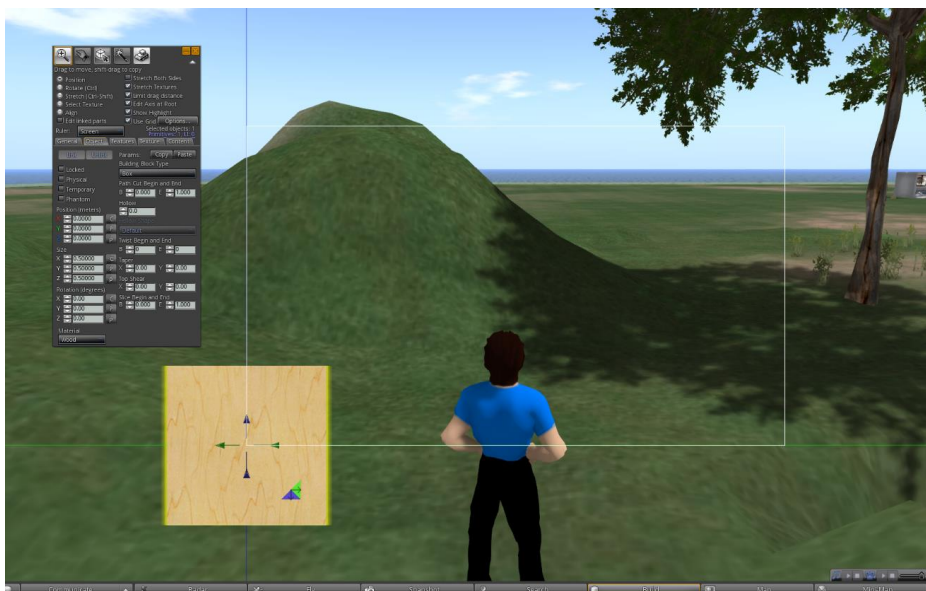
Τέλος, εστιάζουμε στο κεφάλι του χαρακτήρα μας, κάνουμε Edit στο καπέλο και τροποποιούμε τη θέση και το μέγεθος για καλύτερη τοποθέτηση. Μπορούμε να αλλάξουμε ελαφρώς και τον προσανατολισμό του (π.χ. να το γείρουμε λίγο προς κάποια διεύθυνση), ώστε να δείχνει πιο φυσικό.

Μπορούμε να αφαιρέσουμε μια ενσωμάτωση αν με δεξί κλικ πάνω σε αυτήν επιλέξουμε Detach ή Drop. Στην πρώτη περίπτωση το αντικείμενο μεταφέρεται στο αποθετήριο μας, ενώ στη δεύτερη προστίθεται ξανά στον κόσμο αποσυνδεδεμένο από την ενσάρκωσή μας.

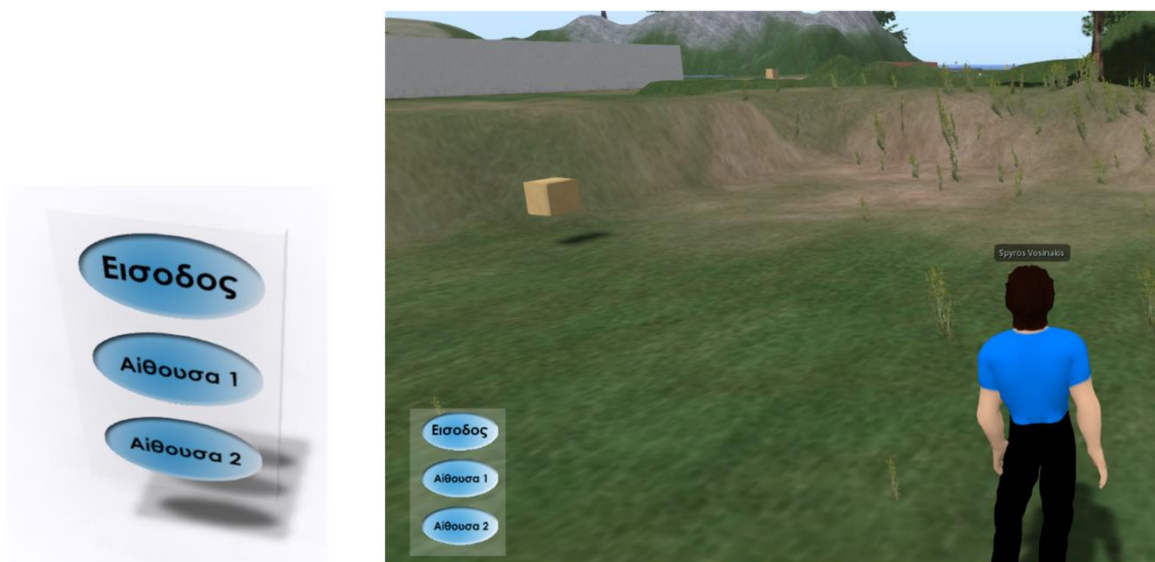
Το ενδιαφέρον στοιχείο με τις ενσωματώσεις είναι ότι «θυμούνται» τις ρυθμίσεις τοποθέτησης που έχουμε κάνει σε αυτές. Έτσι, αν κάνετε στο αποθετήριο σας δεξί κλικ στο καπέλο που έχετε κάνει Detach και επιλέξετε Wear, το καπέλο θα τοποθετηθεί αυτόματα στην τελευταία θέση που του είχατε ορίσει. Αν κάποιος άλλος χρήστης πάρει αντίγραφο του αντικειμένου σας και δοκιμάσει και αυτός να το φορέσει (Wear), το αντικείμενο θα ενσωματωθεί στη δική του ενσάρκωση σε αντίστοιχη θέση, επομένως είναι πιθανό να είναι αρκετά καλή η τοποθέτηση. Αυτό σημαίνει ότι όταν κατασκευάζουμε ένα αντικείμενο που θέλουμε να λειτουργήσει ως ενσωμάτωση (π.χ. εργαλείο, ρούχο κ.λπ.) μπορούμε να καθορίσουμε και τον τρόπο με τον οποίο θα ενσωματώνεται στις ενσαρκώσεις.

Μια άλλη ενδιαφέρουσα τοποθέτηση των αντικειμένων είναι η ενσωμάτωση στην οθόνη του χρήστη ως στοιχείο του HUD. Πράγματι, αν κάνουμε δεξί κλικ σε ένα αντικείμενο, επιλέξουμε More > Attach HUD, θα δούμε ότι μας προσφέρονται ορισμένες επιλογές τοποθέτησης στην οθόνη: πάνω, κάτω, πάνω αριστερά, πάνω δεξιά, κέντρο κ.λπ. Ανάλογα με την επιλογή που θα κάνουμε, το κέντρο του αντικειμένου τοποθετείται στην αντίστοιχη θέση της οθόνης μας. Αυτό σημαίνει ότι, αν για παράδειγμα δημιουργήσουμε ένα παραλληλεπίπεδο και το τοποθετήσουμε στην κάτω αριστερή θέση της οθόνης, θα μπορούμε να δούμε μόνο το πάνω δεξιά τέταρτο του αντικειμένου. Εξ ορισμού η όψη του αντικειμένου που εμφανίζεται στο HUD μας είναι η -X, δηλαδή η όψη στην οποία δείχνει το βέλος του άξονα X.

Και στην περίπτωση των ενσωματώσεων του HUD μπορούμε να επεξεργαστούμε τη θέση και το μέγεθος του αντικειμένου και να τροποποιήσουμε την τοποθέτησή του. Έστω για παράδειγμα ότι κατασκευάζουμε έναν κύβο και τον τοποθετούμε κάτω αριστερά στην οθόνη μας μέσω ενσωμάτωσης HUD. Αν κάνουμε Edit στον κύβο μας, μπορούμε μετακινώντας τη ρόδα του ποντικιού προς τα κάτω να φέρουμε την κάμερα πιο πίσω και να δούμε το περίγραμμα της οθόνης μας. Έτσι, μπορούμε χρησιμοποιώντας τα πλήκτρα μετακίνησης να αλλάξουμε θέση στον κύβο τοποθετώντας τον ολόκληρο μέσα στην οθόνη (εικ. 8.20). Αντίστοιχα, μπορούμε να του αλλάξουμε το μέγεθος ώστε να καλύπτει ικανοποιητικά την περιοχή που θέλουμε. Βγαίνοντας από την κατάσταση επεξεργασίας, η ενσωμάτωση θα παραμείνει στην κατάσταση που επιλέξαμε. Επιπλέον, αν αφαιρέσουμε την ενσωμάτωση και τη «φορέσουμε» (wear) ξανά, αυτή θα «θυμάται» τις ρυθμίσεις τοποθέτησης, όπως ακριβώς συμβαίνει και με τις απλές ενσωματώσεις.



Εικόνα 8.20 Επεξεργασία ενσωμάτωσης στην οθόνη.



Εικόνα 8.21 Κατασκευή ενός στοιχείου διεπαφής και ενσωμάτωσή του στο HUD του χρήστη.

Τα αντικείμενα που ενσωματώνονται στο HUD είναι χρήσιμα αν θέλουμε να εξοπλίσουμε τους χρήστες μας με πλήκτρα που ενεργοποιούν λειτουργίες του κόσμου ή και να τους παρέχουμε μηνύματα και εξατομικευμένες πληροφορίες κατά τη διάρκεια της αλληλεπίδρασης τους.

Ας δούμε ένα παράδειγμα κατασκευής μιας ενσωμάτωσης HUD με τρία πλήκτρα. Έστω ότι θέλουμε να προσφέρουμε στον χρήστη τρία πλήκτρα ελέγχου τα οποία τον τηλεμεταφέρουν σε αντίστοιχες περιοχές του κόσμου. Για παράδειγμα, έστω ότι οι περιοχές μας είναι η Είσοδος, η Αίθουσα 1 και η Αίθουσα 2.

Αρχικά κατασκευάζουμε τρεις υφές για τα αντίστοιχα πλήκτρα χρησιμοποιώντας κάποιο πρόγραμμα επεξεργασίας εικόνας, π.χ. Photoshop. Αν δεν θέλουμε τα πλήκτρα μας να έχουν ορθογώνιο σχήμα, μπορούμε να προσθέσουμε διαφάνεια στην εικόνα μας και να την αποθηκεύσουμε σε μορφότυπο PNG. Ανεβάζουμε τις υφές στον κόσμο του OpenSimulator.

Στη συνέχεια κατασκευάζουμε ένα κουτί στον κόσμο. Μικραίνουμε σημαντικά τις διαστάσεις του για να μην καταλάβει μεγάλο χώρο στην οθόνη. Στο παράδειγμά μας οι διαστάσεις του γίνονται (0.01, 0.1, 0.05). Αφαιρούμε την υφή από το αντικείμενο και το κάνουμε «αόρατο» θέτοντας τιμή διαφάνειας (alpha) 100%. Επιλέγουμε μόνο την επιφάνεια της μπροστινής όψης (-X) και αποδίδουμε τη μία από τις τρεις υφές.

Δημιουργούμε δύο αντίγραφα του αντικείμενου μας τοποθετώντας τα τρία αντικείμενα το ένα πάνω στο άλλο. Αλλάζουμε τις υφές των δύο αντιγράφων στις αντίστοιχες υφές των άλλων δύο πλήκτρων.

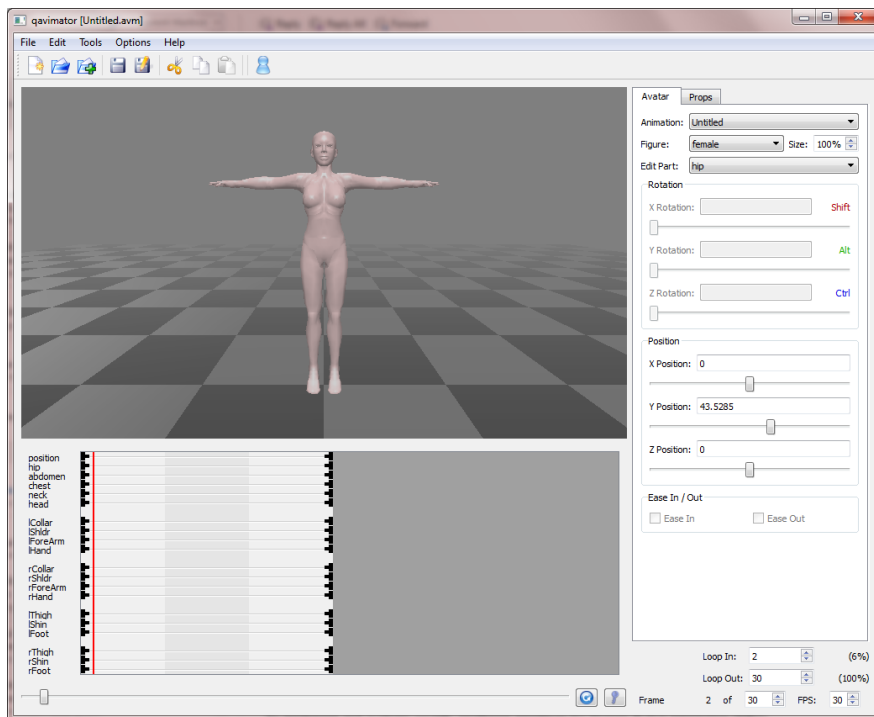
Τέλος, δημιουργούμε ένα ακόμη ορθογώνιο παραλληλεπίπεδο, του οποίου τη θέση και τον προσανατολισμό τροποποιούμε με τρόπο που να πλαισιώνει τα τρία πλήκτρα. Μπορούμε να κάνουμε το αντικείμενο αυτό εντελώς αόρατο ή να του θέσουμε ένα μεγάλο ποσοστό διαφάνειας (π.χ. 80%), ώστε να εμφανίζεται ένα απαλό περιθώριο γύρω από τα τρία πλήκτρα.

Ενώνουμε τα τέσσερα αντικείμενα σε μια ομάδα θέτοντας το τελευταίο αντικείμενο (πλαίσιο) ως ρίζα. Ενσωματώνουμε το αντικείμενο στο HUD, για παράδειγμα κάτω αριστερά, και επεξεργαζόμαστε τη θέση του ώστε να τοποθετηθεί κατάλληλα.

Το αποτέλεσμα φαίνεται στην εικόνα 8.21. Δημιουργήσαμε ένα δικό μας στοιχείο διεπαφής στον κόσμο το οποίο είναι μόνιμα αγκιστρωμένο στην οθόνη. Αν επιτρέπουμε στους επισκέπτες του κόσμου να πάρουν αντίγραφα αυτού του αντικείμενου και να τα «φορέσουν», θα έχουν στη διάθεσή τους τη διεπαφή που κατασκευάσαμε. Στο επόμενο κεφάλαιο θα μάθουμε πώς μπορούμε να προσθέσουμε λειτουργικότητα στα πλήκτρα της διεπαφής μας μέσω της γλώσσας σεναρίων LSL που προσφέρεται στο OpenSimulator.

6 Κατασκευή συνθετικών κινήσεων χαρακτήρων

Σε αρκετές περιπτώσεις η σχεδίαση και ανάπτυξη του περιεχομένου ενός εικονικού κόσμου μπορεί να περιλαμβάνει την εισαγωγή νέων, εξειδικευμένων συνθετικών κινήσεων (animations) των ανθρωπόμορφων χαρακτήρων. Οι λόγοι για κάτι τέτοιο είναι αρκετοί. Μπορεί να θέλουμε η ενσάρκωση του χρήστη να εκτελεί ειδικές κινήσεις κατά την αλληλεπίδρασή της με το περιβάλλον και τα αντικείμενά του. Για παράδειγμα, να προσαρμόσουμε τον τρόπο με τον οποίο κάθεται σε συγκεκριμένες θέσεις, να αποδώσουμε οπτικά τη χρήση διάφορων εργαλείων, να μεταφέρονται μηνύματα μη λεκτικής επικοινωνίας, όπως χειροκρότημα, αποδοκιμασία κ.λπ. Επιπλέον, οι κινήσεις που θα κατασκευάσουμε μπορεί να είναι χρήσιμες και σε ψηφιακούς χαρακτήρες που ενδέχεται να περιλαμβάνει το περιβάλλον μας, όπως αντίπαλοι παίκτες, εικονικοί ξεναγοί κ.λπ.



Εικόνα 8.22 Το πρόγραμμα δημιουργίας συνθετικών κινήσεων χαρακτήρων QAvimator.

Οι συνθετικές κινήσεις χαρακτήρων που υποστηρίζονται στο OpenSimulator περιγράφουν τις μεταβολές στις αρθρώσεις και στη συνολική θέση ενός ανθρώπινου σκελετού στον χρόνο. Ο σκελετός είναι αρκετά απλοποιημένος και δεν υποστηρίζει κινήσεις δακτύλων και εκφράσεις προσώπου. Τα δεδομένα κίνησης περιγράφονται στη μορφή των πλάνων-κλειδιών (keyframing, βλ. Κεφ. 3), δηλαδή σε συγκεκριμένες χρονικές

στιγμές δηλώνονται οι αντίστοιχες τιμές περιστροφής των αρθρώσεων (πλάνα-κλειδιά) και το σύστημα κατά την εκτέλεση της συνθετικής κίνησης υπολογίζει τις ενδιάμεσες τιμές μεταξύ των πλάνων αυτών με τεχνικές παρεμβολής.

Το OpenSimulator δεν παρέχει εργαλεία για τη δημιουργία συνθετικών κινήσεων εντός του κόσμου. Τα δεδομένα συνθετικής κίνησης θα πρέπει να έχουν δημιουργηθεί σε εξωτερικές εφαρμογές και να εισαχθούν στον κόσμο μέσω του προγράμματος σύνδεσης (π.χ. Singularity). Ο μορφότυπος αρχείων δεδομένων κίνησης που υποστηρίζεται από το OpenSimulator είναι ο Biovision Hierarchy (BVH). Τα δημοφιλή προγράμματα τρισδιάστατης μοντελοποίησης και συνθετικής κίνησης, όπως για παράδειγμα το 3D Studio, το Blender και το Poser, έχουν τη δυνατότητα εξαγωγής αρχείων BVH. Για μεγαλύτερη ευκολία όμως μπορούμε να χρησιμοποιήσουμε ένα απλό στη χρήση πρόγραμμα που έχει κατασκευαστεί ειδικά για αυτόν τον σκοπό, το QAvimator. Το QAvimator διατίθεται ελεύθερα και μπορείτε να το βρείτε στη διεύθυνση: <http://www.gavimator.org>

Θα δοκιμάσουμε να κατασκευάσουμε την πρώτη μας συνθετική κίνηση με το QAvimator. Αφού εγκαταστήσουμε και τρέξουμε το πρόγραμμα, παρατηρούμε ότι η διεπαφή του είναι εξαιρετικά απλοϊκή (εικ. 8.22). Στο κέντρο της οθόνης εμφανίζεται ένας τρισδιάστατος χαρακτήρας, στο κάτω μέρος βρίσκεται ο χρονοδιάδρομος στον οποίο δηλώνουμε τα πλάνα-κλειδιά και στα δεξιά οι γενικές ρυθμίσεις του χαρακτήρα.

Έστω ότι θέλουμε να κατασκευάσουμε μια συνθετική κίνηση στην οποία ο χαρακτήρας μας σηκώνει ψηλά τα δύο του χέρια, σαν να πανηγυρίζει για κάποιο συμβάν, και τα ξανακατεβάζει. Αρχικά θα πρέπει να υπολογίσουμε τον χρόνο που θα διαρκέσει η συνθετική κίνησή μας. Κάτω δεξιά το QAvimator γράφει Frame 2 of 30 και δίπλα FPS: 30. Αυτό σημαίνει ότι ο ρυθμός ανανέωσης της κίνησης είναι τριάντα πλάνα το δευτερόλεπτο και ο συνολικός αριθμός πλάνων τριάντα, δηλαδή η κίνηση διαρκεί ένα δευτερόλεπτο. Η δική μας κίνηση θέλουμε να είναι λίγο μεγαλύτερη από αυτό, επομένως αλλάζουμε τον συνολικό αριθμό των πλάνων σε 60.

Η κίνησή μας θα περιγράφεται ως εξής: στην αρχή τα χέρια θα βρίσκονται κάτω, στα μισά της κίνησης θα έχουν σηκωθεί πάνω, και στο τέλος θα επιστρέψουν στην αρχική θέση. Πηγαίνουμε λοιπόν στο καρτέ ν. 2, που είναι το πρώτο ενεργό καρτέ (στο ν.1 αποδίδεται η εξ ορισμού αρχική στάση του σώματος), και φέρνουμε τα δύο χέρια κάτω (εικ. 8.23 αριστερά).

Αρχικά επιλέγουμε το αριστερό πάνω μέρος του χεριού (lShldr), κρατάμε πατημένο το Ctrl και σέρνουμε το ποντίκι προς τα δεξιά. Αυτό εξαναγκάζει τον ώμο να περιστραφεί στον άξονα Z και να φέρει όλο το χέρι προς τα κάτω. Κάνουμε το ίδιο και με το δεξί χέρι. Παρατηρήστε ότι οι αλλαγές στις περιστροφές γίνονται επιλέγοντας το μέρος του σώματος που επιθυμούμε να περιστρέψουμε και πατώντας ένα από τα πλήκτρα Shift, Alt και Control για περιστροφή στους άξονες X, Y και Z αντίστοιχα.

Στη συνέχεια επιλέγουμε τη μέση του χρόνου εκτέλεσης της κίνησης χρησιμοποιώντας τον οριζόντιο δείκτη κάτω από τον χρονοδιάδρομο. Το πλάνο μας (Frame) θα πρέπει να είναι το 30ό. Σηκώνουμε τα χέρια ψηλά αλλάζοντας τις περιστροφές τόσο του πάνω μέρους (lShldr) όσο και του κάτω (lForeArm) (βλ. εικ. 8.23 κεντρικά). Οι τιμές περιστροφής που επιλέγουμε στον ώμο είναι (-60, -10, 45) για τον αριστερό και (-60, -10, -45) για τον δεξιό. Αντίστοιχα προσαρμόζουμε την περιστροφή του αγκώνα αλλάζοντας την περιστροφή του Y σε -60 για τον αριστερό και 60 για τον δεξιό.

Τέλος, επιλέγουμε το τελικό πλάνο και φέρνουμε τα χέρια του χαρακτήρα στην ίδια θέση με αυτήν στο 2ο πλάνο. Μπορείτε να δείτε συνολικά την κίνηση αν μετακινήσετε τον δείκτη του χρονοδιαδρόμου ή αν πατήσετε το πλήκτρο εκτέλεσης δίπλα από αυτόν. Αν θέλουμε να βελτιώσουμε περαιτέρω την κίνηση, μπορούμε να κάνουμε διορθώσεις στα υπάρχοντα πλάνα-κλειδιά ή και να ορίσουμε νέα επιλέγοντας το επιθυμητό πλάνο και κάνοντας αλλαγές στις περιστροφές.

Αποθηκεύουμε την κίνηση σε αρχείο BVH με την εντολή Save. Στη συνέχεια ανοίγουμε το Singularity και ανεβάζουμε την κίνηση στον κόσμο με File > Import Animation (free). Εμφανίζεται ένας διάλογος, ο οποίος μας ζητάει να κάνουμε ορισμένες ρυθμίσεις σχετικά με την κίνηση που ανεβάσαμε. Οι βασικές ρυθμίσεις, πέρα από το όνομα και την περιγραφή, είναι:

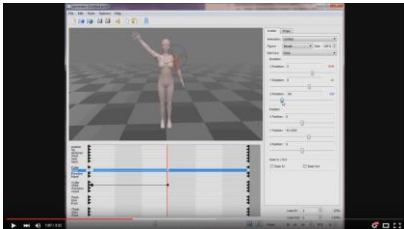
- Βαθμός προτεραιότητας (priority): ο βαθμός αυτός καθορίζει κατά πόσο η συγκεκριμένη κίνηση θα έχει προτεραιότητα πάνω σε άλλες κινήσεις του χαρακτήρα. Αν θέλετε η κίνησή σας να έχει μεγαλύτερη προτεραιότητα από το βάδισμα ή το κάθισμα, είναι καλό να ορίσετε βαθμό 3 ή 4.
- Επανάληψη (loop): καθορίζει αν η κίνηση θα επαναλαμβάνεται ή όχι και μεταξύ ποιων πλάνων (εκφράζεται σε ποσοστό % του συνολικού χρόνου κίνησης).

- Στάση χεριού (hand pose): μπορείτε να επιλέξετε μεταξύ προεπιλεγμένων στάσεων χεριού (π.χ. ανοιχτό, γροθιά κ.λπ.)
- Έκφραση (expression): υπάρχει επιλογή από ένα σύνολο εκφράσεων προσώπου.

Για την κίνηση που ανεβάσαμε επιλέγουμε βαθμό προτεραιότητας 4 και στάση χεριού γροθιά (fist). Ελέγχουμε την κίνηση στο παράθυρο προεπισκόπησης (preview) και πατάμε Upload για να ανέβει.



Εικόνα 8.23 Κατασκευή μιας νέας συνθετικής κίνησης και ανέβασμα στο OpenSimulator



Σύνδεσμος:
<https://youtu.be/CAys49FDnB0>

Video 8.6 Κατασκευή συνθετικής κίνησης στο QAnimator και εισαγωγή στον εικονικό κόσμο.

Η κίνησή μας ανέβηκε στο αποθετήριο μας στον φάκελο Animations. Αν κάνουμε διπλό κλικ πάνω της και πατήσουμε “Play in World” θα εκτελέσουμε την κίνηση στον κόσμο (βλ. εικ. 8.23 δεξιά) και θα μπορούν να τη δουν οι υπόλοιποι χρήστες.

Η διαδικασία κατασκευής της συνθετικής κίνησης και εισαγωγής της στον εικονικό κόσμο παρουσιάζεται στο βίντεο 8.6.

Στα κεφάλαια 10 και 11 θα δούμε πώς μπορούμε να ενεργοποιήσουμε συνθετικές κινήσεις ενσαρκώσεων και ψηφιακών χαρακτήρων μέσω κώδικα.

7 Συμπεράσματα

Στο παρόν κεφάλαιο εστίασαμε στην ανάπτυξη περιεχομένου για το OpenSimulator. Μελετήσαμε τους τρόπους με τους οποίους μπορούν να κατασκευαστούν απλά και περίπλοκα αντικείμενα μέσω των παρεχόμενων οπτικών εργαλείων, παρουσιάσαμε τεχνικές τροποποίησης του αναγλύφου του εδάφους και του φωτισμού του περιβάλλοντος και περιγράψαμε τη διαδικασία παραγωγής και ενσωμάτωσης νέων συνθετικών κινήσεων μέσω του προγράμματος QAnimator.

Από τη μελέτη των παραπάνω γίνεται σαφές ότι το περιβάλλον του OpenSimulator δεν είναι όσο ώριμο και εύχρηστο θα περίμενε κανείς για τη δημιουργία υψηλής ποιότητας περιεχομένου. Παρόλα αυτά, αν επενδύσει κάποιος χρόνο στη μελέτη των δυνατοτήτων του και αποκτήσει τις απαιτούμενες δεξιότητες, μπορεί να καταλήξει σε αρκετά ποιοτικά αποτελέσματα, όπως μαρτυρά το πλούσιο περιεχόμενο στους

κατασκευασμένους κόσμους του Second Life ή σε άλλους κόσμους βασισμένους στο OpenSimulator (π.χ. OS Grid). Είναι επίσης σημαντικό το ότι υπάρχει ήδη αρκετό κατασκευασμένο περιεχόμενο που διατίθεται δωρεάν και μπορεί να προστεθεί σε υπάρχοντες κόσμους. Με την κατάλληλη αναζήτηση μπορεί κάποιος να εντοπίσει ομάδες αντικειμένων, προκατασκευασμένους κόσμους, ανάγλυφα εδάφους, έτοιμους χαρακτήρες και ρούχα, ακόμη και περίπλοκες συνθετικές κινήσεις. Τέλος, χάρη στη δυνατότητα εισαγωγής αρχείων πολυγωνικού πλέγματος (mesh), μπορεί κάποιος να εισαγάγει οποιοδήποτε διαθέσιμο τρισδιάστατο μοντέλο και να το εμφανίσει στον κόσμο, αφού το μετατρέψει πρώτα στον κατάλληλο μορφότυπο.

Αυτό που είναι πιο σημαντικό από την ευκολία κατασκευής και τη διαθεσιμότητα περιεχομένου είναι το μοντέλο συμμετοχικής ανάπτυξης περιεχομένου. Μέσα από τα απλοϊκά έστω εργαλεία του περιβάλλοντος μπορούν ακόμη και αρχάριοι χρήστες να συνεργαστούν, να συνεισφέρουν και να συνδιαμορφώσουν το περιβάλλον. Η δυνατότητα αυτή είναι εξαιρετικά σημαντική σε κόσμους οι οποίοι περιλαμβάνουν σχεδιαστικές διεργασίες, συμμετοχική λήψη αποφάσεων και περιεχόμενο δημιουργημένο από τους χρήστες.

Σύνδεσμοι

QAvimator: <http://www.qavimator.org>

Sculpted prims: http://wiki.secondlife.com/wiki/Sculpted_prim, οδηγίες για τη δημιουργία τους.

Ελεύθερο περιεχόμενο για το OpenSimulator: http://opensimulator.org/wiki/Artist_Home

Ασκήσεις

1. Κατασκευάστε τον εσωτερικό χώρο μιας αίθουσας διδασκαλίας χρησιμοποιώντας διαστάσεις όσο το δυνατόν πιο κοντά στις πραγματικές. Η αίθουσά σας θα πρέπει να περιλαμβάνει τουλάχιστον: καρέκλες, θρανία, ασπροπίνακα, πανί προβολής και χώρο ανακοινώσεων.
2. Προσπαθήστε να μοντελοποιήσετε καθημερινά χρηστικά αντικείμενα με τη χρήση βασικών στερεών. Για παράδειγμα, δοκιμάστε να κατασκευάσετε ένα τραπέζι φαγητού πάνω στο οποίο θα υπάρχουν πιάτα, πιρούνια, μαχαίρια, κουτάλια και ποτήρια. Καταγράψτε τις δυσκολίες που αντιμετωπίσατε καθώς και καλές πρακτικές που ανακαλύψατε.
3. Τοποθετήστε σε μεγάλη οριζόντια επιφάνεια την κάτοψη ενός πραγματικού κτιρίου ως υφή και δοκιμάστε να το κατασκευάσετε στον εικονικό κόσμο προσθέτοντας και πόρτες ή παράθυρα όπου χρειάζεται.
4. Δημιουργήστε ένα μικρό νησί για ένα πειρατικό παιχνίδι περιπέτειας. Το νησί σας θα πρέπει να έχει αρκετή βλάστηση και να περιλαμβάνει ηφαίστειο και λίμνη. Προσπαθήστε να κάνετε το αποτέλεσμα να δείχνει όσο το δυνατόν πιο φυσικό.
5. Κατασκευάστε πειρατικά καπέλα και σπαθιά τα οποία να φοριούνται σωστά για το υποθετικό παιχνίδι της άσκησης 4.
6. Δημιουργήστε μία μικρή ενσωμάτωση HUD για την αξιολόγηση εκθεμάτων σε μουσείο, για παράδειγμα Like, Dislike, Προσθήκη Σχολίου. Προτιμήστε πλήκτρα με εικονίδια αντί για κείμενο.
7. Βελτιώστε τη συνθετική κίνηση «πανηγυρισμού» που δημιουργήσαμε κάνοντάς τη μεγαλύτερη σε διάρκεια και πιο αληθοφανή.

Κεφάλαιο 9: Εισαγωγή στη γλώσσα LSL/OSSL

Σύνοψη

Η γλώσσα LSL είναι μια γλώσσα σεναρίων που αναπτύχθηκε για τον προγραμματισμό της συμπεριφοράς των αντικειμένων στο περιβάλλον του Second Life. Το περιβάλλον OpenSimulator υποστηρίζει το μεγαλύτερο μέρος της LSL και αρκετές επιπλέον εντολές (OSSL) που ενισχύουν τις λειτουργίες του. Στο παρόν κεφάλαιο γίνεται μια εισαγωγή στη σύνταξη και χρήση των παραπάνω γλωσσών. Αρχικά παρουσιάζεται το μοντέλο μηχανής καταστάσεων (state machine) και ο προγραμματισμός βασισμένος σε συμβάντα (event-based programming) που υιοθετούνται από τη γλώσσα. Στη συνέχεια παρουσιάζονται οι τύποι δεδομένων, η σύνταξη, ο ορισμός καταστάσεων και η μετάβαση μεταξύ αυτών και οι βασικές εντολές της γλώσσας (δομές επανάληψης και εκτέλεσης υπό συνθήκη, μετασχηματισμός δεδομένων κ.ά.). Έπειτα παρουσιάζεται ένας αριθμός από συναρτήσεις που μπορούν να επηρεάσουν την εμφάνιση και τις ιδιότητες των αντικειμένων καθώς και κατάλληλα συμβάντα για να αντιλαμβάνονται τα αντικείμενα μεταβολές στο περιβάλλον τους. Τα παραπάνω επεξηγούνται μέσω παραδειγμάτων κώδικα που μπορεί να εκτελέσει ο αναγνώστης και συμπληρώνονται από μεγάλο αριθμό ασκήσεων για πληρέστερη κατανόηση.

Προαπαιτούμενη γνώση

Απαιτούνται γνώσεις πληροφορικής, γνώσεις προγραμματισμού, και βασικές δεξιότητες χρήσης και κατασκευής περιεχομένου στο OpenSimulator.

1 Εισαγωγή

Ένα μεγάλο πλεονέκτημα των ανοιχτών εικονικών κόσμων γενικού σκοπού, όπως το Second Life και το OpenSimulator, έναντι των περισσότερο στοχευμένων περιβαλλόντων, είναι ότι δίνουν τη δυνατότητα στους ίδιους τους χρήστες να διαμορφώσουν τον χώρο και να προσδώσουν το νόημα που αυτοί επιθυμούν. Ο χρήστης είναι ταυτόχρονα και σχεδιαστής και ο κόσμος λειτουργεί σαν χώρος πειραματισμού αλλά και περιβάλλον ανάπτυξης που επιτρέπει την ανάδυση νέων λύσεων και την εκμετάλλευση των δυνατοτήτων του κόσμου σε πολλαπλούς χώρους εφαρμογής. Έτσι, πέρα από τα εργαλεία κατασκευής νέου περιεχομένου εντός κόσμου, το OpenSimulator δίνει επιπλέον στους χρήστες του την ευκαιρία να προγραμματίσουν τη συμπεριφορά των αντικειμένων τους μέσω μιας ενσωματωμένης γλώσσας που έχει κατασκευαστεί ειδικά για αυτόν τον σκοπό. Συνδυάζοντας τα δύο παραπάνω χαρακτηριστικά μπορεί ένας χρήστης ή μια ομάδα ανάπτυξης να σχεδιάσει εξ ολοκλήρου τη μορφή και τη λειτουργικότητα μιας περιοχής, να δημιουργήσει κατάλληλα στοιχεία διεπαφής για τους επισκέπτες και να καταλήξει σε μια λειτουργική και εύχρηστη εφαρμογή. Όπως έχουμε άλλωστε ήδη παρουσιάσει στο Κεφάλαιο 6, οι εικονικοί κόσμοι γενικού σκοπού έχουν καταφέρει να βρουν εφαρμογή με επιτυχία σε διάφορους χώρους, όπως η εκπαίδευση, η σχεδίαση, ο πολιτισμός, η ψυχαγωγία κ.ά.

Στόχος αυτού του κεφαλαίου είναι η εισαγωγή στη γλώσσα σεναρίων (scripting language) LSL/OSSL του OpenSimulator, η επίδειξη της χρήσης της σε διάφορα πλαίσια και η καλλιέργεια βασικών δεξιοτήτων για την υλοποίηση πλούσιων σεναρίων σε εικονικούς κόσμους. Απαραίτητη προϋπόθεση για τη μελέτη του κεφαλαίου είναι να έχει ο αναγνώστης βασική κατανόηση προγραμματιστικών εννοιών και κατά προτίμηση να γνωρίζει ήδη κάποια γλώσσα προγραμματισμού. Για την καλύτερη εμβάθυνση στα παραδείγματα και στην επίλυση των ασκήσεων αυτού του κεφαλαίου θα πρέπει επίσης ο αναγνώστης να έχει αποκτήσει βασικές δεξιότητες χρήσης και κατασκευής περιεχομένου στο Second Life ή στο OpenSimulator. Κατ' ελάχιστο θα πρέπει να γνωρίζει πώς να πλοηγείται, να επικοινωνεί και να κατασκευάζει απλά ή σύνθετα αντικείμενα στο περιβάλλον. Στην επόμενη ενότητα θα παρουσιαστεί συνοπτικά η ιστορία, η λειτουργικότητα και τα κύρια προγραμματιστικά υποδείγματα που ακολουθεί η γλώσσα LSL, δηλαδή οι μηχανές καταστάσεων και ο προγραμματισμός βασισμένος σε συμβάντα. Στη συνέχεια θα γίνει μια εισαγωγή στη σύνταξη και στις βασικές έννοιες της γλώσσας, όπου θα αναφερθούμε μεταξύ άλλων στους τύπους δεδομένων, στις μεταβλητές, στις συναρτήσεις και στις εντολές που υποστηρίζονται. Στην επόμενη ενότητα θα εξετάσουμε τεχνικές με τις οποίες μπορούν τα αντικείμενα του κόσμου να εκτελέσουν ενέργειες στο περιβάλλον μέσω εντολών της γλώσσας. Αντίστοιχα, θα δούμε πώς μπορούν τα αντικείμενα να παρακολουθούν αλλαγές του περιβάλλοντος, ώστε να είναι σε θέση να αποφασίσουν πότε και αν θα εκτελέσουν κάποια ενέργεια.

2 Σχετικά με τη γλώσσα LSL/OSSL

Η γλώσσα Linden Scripting Language (LSL) είναι μια γλώσσα σεναρίων η οποία δημιουργήθηκε από την εταιρεία Linden Labs για τον προγραμματισμό της διαδραστικής συμπεριφοράς των αντικειμένων στον εικονικό κόσμο του Second Life. Ο προγραμματιστής μπορεί μέσω της γλώσσας αυτής να αλλάζει δυναμικά την εμφάνιση και τη συμπεριφορά των αντικειμένων βάσει εξωτερικών συμβάντων, όπως οι ενέργειες των χρηστών ή η πάροδος του χρόνου. Περιλαμβάνει πολλές υλοποιημένες συναρτήσεις (πάνω από τριακόσιες) για την ανάγνωση και αλλαγή ιδιοτήτων των αντικειμένων, μετατροπές δεδομένων, στοιχεία σχετικά με το περιβάλλον, επικοινωνία κ.λπ. και κατά συνέπεια μπορεί να αξιοποιηθεί πρακτικά με άπειρους τρόπους, από απλές πόρτες που ανοίγουν όταν ο χρήστης κάνει κλικ πάνω τους μέχρι περίπλοκες διαδραστικές προσομοιώσεις.

Το OpenSimulator υποστηρίζει το μεγαλύτερο μέρος της λειτουργικότητας της LSL. Επομένως, στις περισσότερες περιπτώσεις τα προγράμματα (scripts) που έχουν γραφτεί για το Second Life αναμένεται να παράγουν τα ίδια ακριβώς αποτελέσματα στο OpenSimulator. Επιπλέον, η ομάδα ανάπτυξης του OpenSimulator έχει επεκτείνει τη γλώσσα LSL με ένα σύνολο συναρτήσεων (που ονομάζονται συναρτήσεις OS), οι οποίες αυξάνουν περαιτέρω τις δυνατότητες της γλώσσας (π.χ. παραγωγή δυναμικών υφών, κίνηση χαρακτήρων κ.λπ.) Το σύνολο των επεκτάσεων αυτών ονομάζεται OpenSimulator Scripting Language (OSSL). Επομένως, η γλώσσα προγραμματισμού του OpenSimulator είναι ο συνδυασμός των LSL και OSSL, ο οποίος συνήθως γράφεται συνοπτικά LSL/OSSL.

Ένα ενδιαφέρον χαρακτηριστικό της LSL είναι ότι τα προγράμματα συσχετίζονται με αντικείμενα του κόσμου. Κάθε πρόγραμμα τρέχει σε ένα μόνο αντικείμενο και συνήθως αποκτά πρόσβαση στις ιδιότητες ή/και επηρεάζει τη συμπεριφορά του αντικειμένου αυτού. Εντούτοις, στο ίδιο αντικείμενο μπορεί να τρέχουν παραπάνω από ένα προγράμματα. Στην περίπτωση αυτή τα προγράμματα τρέχουν παράλληλα. Επιπλέον, αν δημιουργηθεί αντίγραφο από ένα αντικείμενο εντός του οποίου τρέχει κάποιο πρόγραμμα, τότε στο νέο αντικείμενο θα τρέξει ένα νέο στιγμιότυπο του προγράμματος εντελώς ανεξάρτητο από το πρώτο. Ένας εικονικός κόσμος με πλούσια διαδραστική συμπεριφορά μπορεί να περιλαμβάνει δεκάδες χιλιάδες προγράμματα τα οποία τρέχουν παράλληλα. Αυτό μπορεί να ξαφνιάζει όσους έχουν συνηθίσει μοντέλα προγραμματισμού στα οποία η τελική έξοδος καθορίζεται από ένα μόνο πρόγραμμα ή έστω από πολλαπλά αρχεία κώδικα που αντιμετωπίζονται όμως ως ενιαίο πρόγραμμα. Η περίπτωση της LSL είναι αρκετά διαφορετική. Τρέχουν πολλαπλά μικρά τμήματα κώδικα τα οποία ελέγχουν τα διάφορα αντικείμενα του κόσμου και μπορούν να επικοινωνούν μεταξύ τους μέσω μηνυμάτων, όπως θα δούμε στην επόμενη ενότητα. Στη σημείο αυτό θα πρέπει να αναφερθεί ότι, ενώ οι δυνατότητες είναι πρακτικά άπειρες, οι υπολογιστικοί πόροι παραμένουν πεπερασμένοι. Όλα τα προγράμματα εκτελούνται στον διακομιστή και επομένως υπάρχουν περιορισμοί ως προς τη συνολική υπολογιστική ισχύ και μνήμη που καταναλώνουν τα προγράμματα αυτά, ανάλογα πάντα και με τις δυνατότητες του υλικού και το λειτουργικό σύστημα. Κατά συνέπεια, οι χρήστες θα πρέπει να είναι προσεκτικοί στα προγράμματα που γράφουν και να αποφεύγουν αλγορίθμους που απαιτούν πολλή μνήμη ή υπολογιστική ισχύ.

Η γλώσσα LSL υιοθετεί δύο γνωστά προγραμματιστικά υποδείγματα: τις μηχανές καταστάσεων και τα συμβάντα. Ακολουθεί μια σύντομη εισαγωγή στις δύο αυτές έννοιες και στη χρήση τους στον προγραμματισμό.

2.1 Μηχανές καταστάσεων

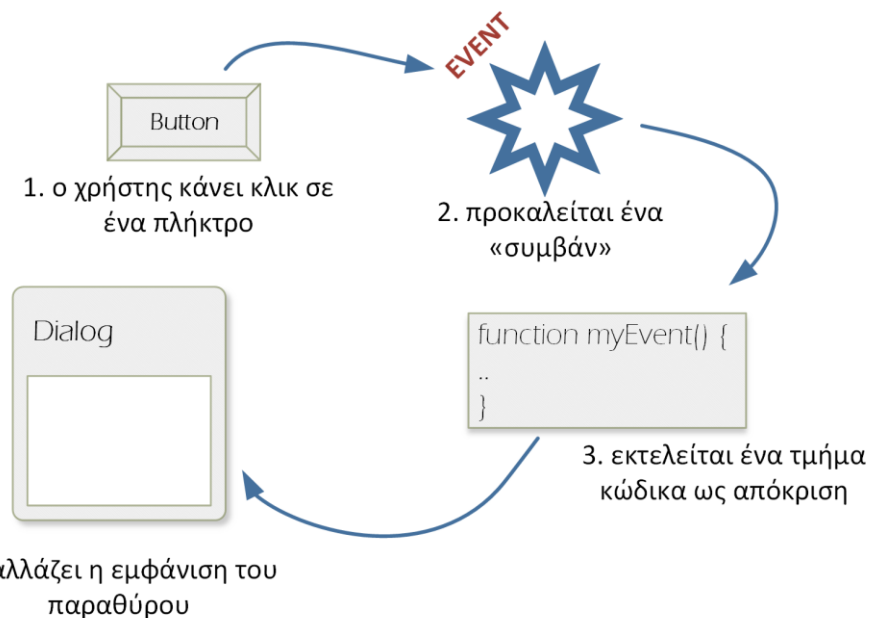
Οι μηχανές καταστάσεων μπορούν να αποτελέσουν μια καλή λύση στον προγραμματισμό σύνθετων συμπεριφορών μέσω της απλοποίησής τους σε επιμέρους καταστάσεις. Έχουμε ήδη αναφερθεί σε αυτές στο τρίτο κεφάλαιο ως έναν απλό μηχανισμό παραγωγής δυναμικής συμπεριφοράς σε χαρακτήρες ή αντικείμενα του περιβάλλοντος. Με τις μηχανές καταστάσεων περιγράφουμε τη συμπεριφορά μιας οντότητας ως ένα σύνολο διακριτών καταστάσεων στις οποίες μπορεί να βρεθεί η οντότητα αυτή και συνθηκών που μπορούν να προκαλέσουν μετάβαση από μία κατάσταση σε άλλη. Η οντότητα αρχικά βρίσκεται σε μία κατάσταση την οποία έχουμε ορίσει ως αρχική και, ανάλογα με την εξέλιξη του περιβάλλοντος, μπορεί να μεταβεί σε κάποια άλλη κατάσταση, εάν πληρούνται οι συνθήκες μετάβασης που έχουμε καθορίσει, από αυτή σε κάποια επόμενη κ.ο.κ. Με την πρακτική αυτή διευκολύνεται ο προγραμματισμός της συμπεριφοράς των αντικειμένων ενός κόσμου, διότι η διαδικασία μοιράζεται σε δύο στάδια: ένα στάδιο υψηλού επιπέδου στο οποίο η συμπεριφορά αναλύεται σε καταστάσεις και μεταβάσεις μεταξύ τους και ένα χαμηλότερου επιπέδου στο οποίο

προσδιορίζεται με λεπτομέρεια και εντέλει προγραμματίζεται η συμπεριφορά σε κάθε επιμέρους κατάσταση. Η απλότητα και ταυτόχρονα η αποτελεσματικότητα αυτής της προσέγγισης την έχει κάνει αρκετά δημοφιλή σε εικονικούς κόσμους και παιχνίδια, όπου ο προσδιορισμός πολλαπλών σύνθετων συμπεριφορών είναι ο κανόνας.

Μια μηχανή καταστάσεων μπορεί να υλοποιηθεί εύκολα σε οποιαδήποτε γλώσσα προγραμματισμού. Το ενδιαφέρον στοιχείο με τη γλώσσα LSL είναι ότι το μοντέλο εκτέλεσης και κατ' επέκταση η δομή των προγραμμάτων βασίζονται εγγενώς σε μηχανές καταστάσεων. Επομένως, είτε το επιθυμεί ο προγραμματιστής είτε όχι, οι καταστάσεις υπάρχουν. Αν κάποιος δεν θέλει να χρησιμοποιήσει τη λογική των μηχανών καταστάσεων στο πρόγραμμά του, αρκεί να γράψει όλο του το πρόγραμμα σε μία μοναδική κατάσταση, την αρχική.

2.2 Προγραμματισμός βασισμένος σε συμβάντα

Στα εισαγωγικά μαθήματα προγραμματισμού οι εκπαιδευόμενοι γράφουν συνήθως προγράμματα με ξεκάθαρη ροή. Ο κώδικας εκτελείται από μία καθορισμένη αρχή και καταλήγει (συνήθως) σε κάποιο τέλος, πιθανότατα καλώντας στην πορεία και ορισμένες συναρτήσεις που έχει δημιουργήσει ο χρήστης. Το μοντέλο αυτό όμως δεν είναι ιδιαίτερα χρήσιμο στις σύγχρονες γλώσσες προγραμματισμού και περιβάλλοντα εκτέλεσης. Ας σκεφτούμε, για παράδειγμα, μια παραθυρική εφαρμογή. Την ώρα που η εφαρμογή εκτελείται, πολλές διεργασίες λαμβάνουν χώρα στο περιθώριο. Ο χρήστης κινεί τον δείκτη του ποντικιού, κάνει κλικ σε πλήκτρα του παράθυρου, πατάει πλήκτρα στο πληκτρολόγιο, μετακινεί εικονίδια, ελαχιστοποιεί ή μεγιστοποιεί παράθυρα, κάνει δεξί κλικ σε αντικείμενα του περιβάλλοντος κ.λπ. Η συγγραφή ενός ενιαίου προγράμματος το οποίο παρακολουθεί όλες αυτές τις πιθανές διεργασίες και ανταποκρίνεται κατάλληλα με βάση τη λογική του προγράμματος θα ήταν εξαιρετικά περίπλοκη και επίπονη για έναν προγραμματιστή και είναι πιθανό να οδηγούσε σε λάθη και παραλείψεις. Για να αποφύγουν τέτοιου είδους δυσκολίες, οι περισσότερες σύγχρονες γλώσσες προγραμματισμού για ανάπτυξη παραθυρικών προγραμμάτων ακολουθούν το παράδειγμα του προγραμματισμού βασισμένου σε συμβάντα.



Εικόνα 9.1 Παράδειγμα διαχείρισης συμβάντων σε παραθυρική γλώσσα προγραμματισμού.

Σύμφωνα με τη λογική του προγραμματισμού βασισμένου σε συμβάντα, το περιβάλλον εκτέλεσης της εφαρμογής μπορεί να εντοπίσει έναν αριθμό από ενέργειες του χρήστη ή αλλαγές στις ιδιότητες του συστήματος, τα οποία ονομάζει *συμβάντα* (*events*). Τα πιο συνηθισμένα συμβάντα στον προγραμματισμό παραθυρικών εφαρμογών αφορούν ενέργειες του πληκτρολογίου και του ποντικιού, ενώ στις κινητές εφαρμογές αναγνωρίζονται αντίστοιχα η επαφή και οι κινήσεις των δακτύλων στην οθόνη αφής. Τα συμβάντα δεν σχετίζονται μόνο με δραστηριότητες του χρήστη· μπορούν επίσης να αφορούν την εκτέλεση του προγράμματος

και δραστηριότητες του συστήματος, για παράδειγμα ξεκίνησε η αναπαραγωγή ήχου, φορτώθηκε το αρχείο κ.λπ. Ο προγραμματιστής αποφασίζει ποια από τα συμβάντα είναι χρήσιμα για τη δική του εφαρμογή και γράφει *διαχειριστές συμβάντων (event handlers)* γι' αυτά. Οι διαχειριστές συμβάντων είναι ειδικές συναρτήσεις οι οποίες καλούνται από το σύστημα αν εντοπιστεί το αντίστοιχο συμβάν. Επομένως, η λογική ενός προγράμματος που ακολουθεί το παραπάνω παράδειγμα είναι «σπασμένη» σε μικρά τμήματα κώδικα που γράφονται μέσα σε έναν αριθμό από διαχειριστές συμβάντων. Η ροή εκτέλεσης του προγράμματος εξαρτάται από τα συμβάντα που θα δημιουργηθούν κατά τη διάρκεια εκτέλεσής του. Στην εικόνα 9.1 παρουσιάζεται ένα παράδειγμα επεξεργασίας ενός συμβάντος σε γλώσσα προγραμματισμού που υποστηρίζει συμβάντα.

Η γλώσσα LSL βασίζεται στα συμβάντα. Τα συμβάντα που μπορεί να αναγνωρίσει σχετίζονται κυρίως με τη δραστηριότητα των χρηστών στον εικονικό κόσμο, για παράδειγμα κάποιος έκανε κλικ πάνω στο αντικείμενο, κάποιος έστειλε ένα μήνυμα κειμένου στον κόσμο, κάποιος πλησίασε ή συγκρούστηκε με ένα αντικείμενο κ.λπ. Κάθε κατάσταση στη γλώσσα LSL έχει τους δικούς της διαχειριστές συμβάντων, επομένως κάθε φορά που γίνεται μετάβαση σε άλλη κατάσταση παρακολουθείται ένα διαφορετικό σύνολο συμβάντων.

3 Εισαγωγή στον προγραμματισμό σε LSL

Στη συνέχεια θα δημιουργήσουμε και θα τρέξουμε το πρώτο μας πρόγραμμα, ώστε να κάνουμε μια εισαγωγή στα βασικά στοιχεία και στη σύνταξη της LSL. Όπως έχουμε ήδη αναφέρει, κάθε πρόγραμμα τρέχει σε ένα μόνο αντικείμενο. Ουσιαστικά το πρόγραμμα αποτελεί μέρος των «περιεχομένων» του αντικειμένου. Για να δημιουργήσουμε ένα νέο πρόγραμμα σε ένα αντικείμενο, θα πρέπει να είμαστε ο ιδιοκτήτης του (owner), δηλαδή να έχουμε κατασκευάσει εμείς οι ίδιοι το αντικείμενο.

Ακολουθήστε τα παρακάτω βήματα:

1. Δημιουργήστε ένα νέο αντικείμενο ή επιλέξτε ένα αντικείμενο που σας ανήκει.
2. Κάντε δεξί κλικ και επιλέξτε Edit...
3. Επιλέξτε την καρτέλα Content.
4. Πατήστε το πλήκτρο New Script.

Θα παρατηρήσετε ότι προστέθηκε ένα νέο αρχείο με το όνομα New Script στα περιεχόμενα του αντικειμένου. Ένας εναλλακτικός τρόπος δημιουργίας νέου προγράμματος είναι να το δημιουργήσετε στο αποθετήριο σας και να το μεταφέρετε στη συνέχεια (με drag & drop) στα περιεχόμενα του αντικειμένου σας.

Αμέσως μετά τη δημιουργία του αρχείου θα παρατηρήσετε στην οθόνη το μήνυμα “Primitive: Script running”, αν το όνομα του αντικειμένου σας είναι Primitive. Αυτό που συνέβη είναι ότι το πρόγραμμα ξεκίνησε την εκτέλεσή του και έστειλε αυτό το μήνυμα εκ μέρους του αντικειμένου στο δημόσιο κανάλι συνομιλιών. Ο λόγος είναι ότι κάθε νέο πρόγραμμα που δημιουργείτε περιλαμβάνει κώδικα που του λέει να κάνει ακριβώς αυτό: να γράψει στο chat τη φράση “Script running”.

Ας δούμε τον κώδικα που περιέχεται στο αρχείο. Αν κάνετε διπλό κλικ στο “New Script” στα περιεχόμενα του αντικειμένου, θα ανοίξει ένα παράθυρο επεξεργασίας και θα εμφανιστεί ο κώδικας του προγράμματος. Σε αυτό το περιβάλλον επεξεργασίας θα μπορείτε στη συνέχεια να γράφετε τα προγράμματά σας σε LSL τροποποιώντας τον αρχικό αυτόν κώδικα. Ο κώδικας που θα δείτε είναι ο παρακάτω:

```
default
{
    state_entry()
    {
        llSay(0, "Script running");
    }
}
```

Μπορείτε να παρατηρήσετε ότι η LSL έχει έναν τρόπο σύνταξης που θυμίζει γλώσσες όπως η C και η Java. Όλες οι εντολές ενός προγράμματος γράφονται μέσα σε κάποιο ζευγάρι άγκιστρων { }. Οι εντολές που βρίσκονται μέσα σε άγκιστρα ονομάζονται τμήμα κώδικα (codeblock ή block).

Το νόημα του παραπάνω προγράμματος είναι το εξής:

1. Υπάρχει ακριβώς μία κατάσταση η οποία ονομάζεται default και είναι η αρχική κατάσταση.

2. Μέσα σε αυτήν την κατάσταση, δηλαδή στο τμήμα κώδικα της κατάστασης default, υπάρχει ένας διαχειριστής συμβάντων που ονομάζεται state_entry. Το συμβάν “state entry”, το οποίο δηλώνεται ως state_entry(), είναι ένα ειδικό συμβάν που λαμβάνει χώρα όταν η μηχανή καταστάσεων εισέρχεται σε μία νέα κατάσταση. Με δεδομένο ότι το συγκεκριμένο συμβάν έχει δηλωθεί στην αρχική κατάσταση, ο κώδικάς του θα εκτελεστεί μόλις ξεκινήσει η εκτέλεση του προγράμματος.
3. Μέσα σε αυτόν τον διαχειριστή συμβάντων υπάρχει μία μοναδική εντολή. Η εντολή καλεί μία ενσωματωμένη συνάρτηση της γλώσσας με όνομα HSay, η οποία δέχεται δύο ορίσματα: έναν ακέραιο αριθμό και μία συμβολοσειρά. Η συνάρτηση αυτή, όταν εκτελείται, εξαναγκάζει το αντικείμενο να στείλει ένα μήνυμα κειμένου στο περιβάλλον. Στην περίπτωση του κώδικα που εξετάζουμε το μήνυμα είναι το “Script running”. Το πρώτο όρισμα είναι ο αριθμός του καναλιού επικοινωνίας και το δεύτερο το μήνυμα. Το δημόσιο κανάλι είναι το κανάλι 0 και τα μηνύματα που στέλνονται στο κανάλι αυτό τα βλέπουν όλοι οι χρήστες εντός κάποιας απόστασης. Τα υπόλοιπα κανάλια επικοινωνίας χρησιμοποιούνται για την επικοινωνία μεταξύ αντικειμένων, όπως θα δούμε στο επόμενο κεφάλαιο.

Αν αλλάξετε το μήνυμα και αποθηκεύσετε το νέο πρόγραμμα, τότε αυτό θα εκτελεστεί ξανά από την αρχή και το αποτέλεσμα θα είναι, προφανώς, η αποστολή του αλλαγμένου μηνύματος στο δημόσιο κανάλι.

Ακολουθούν ορισμένες βασικές συντακτικές αρχές του προγραμματισμού σε LSL:

- Είναι υποχρεωτικό για όλα τα προγράμματα να έχουν μια κατάσταση με όνομα default, καθώς αυτή είναι πάντα η αρχική κατάσταση. Τη δήλωση της κατάστασης ακολουθεί ένα τμήμα κώδικα
- Μέσα στο τμήμα κώδικα μίας κατάστασης επιτρέπεται να δηλωθούν μόνο διαχειριστές συμβάντων.
- Ένας διαχειριστής συμβάντων μπορεί να έχει ορίσματα, τα οποία δηλώνονται εντός των παρενθέσεων δίπλα στο όνομα του διαχειριστή, όπως δηλαδή συντάσσονται και οι δηλώσεις συναρτήσεων στις περισσότερες γλώσσες προγραμματισμού. Στο σώμα του δηλώνεται ένα τμήμα κώδικα το οποίο εκτελείται όταν εντοπιστεί το αντίστοιχο συμβάν.
- Όλες οι εντολές πρέπει να τελειώνουν με ελληνικό ερωτηματικό (;)

Ακολουθεί ένα αρκετά πιο σύνθετο πρόγραμμα με στόχο την παρουσίαση της δομής ενός τυπικού προγράμματος σε LSL. Αν η ερμηνεία του προγράμματος σας φαίνεται αρχικά δύσκολη, αυτό δεν θα πρέπει να σας ανησυχεί. Θα αναλύσουμε στη συνέχεια ένα ένα τα στοιχεία του και θα περιγράψουμε πολύ αναλυτικότερα τους βασικούς κανόνες σύνταξης και σημασιολογίας για καθένα από αυτά.

```
//σχόλιο γραμμής

/* αυτό είναι
ένα σχόλιο
τιμήματος (block comment) */

//δήλωση σφαιρικών μεταβλητών, προσβάσιμων από όλες
//τις καταστάσεις
integer num = 15;

//συναρτήσεις χρήστη, επίσης προσβάσιμες από όλες τις
//καταστάσεις
integer add(integer a, integer b) {
    return a+b;
}

default {
    state_entry() {
        //συνένωση strings με +
```

```

//μετατροπή τύπου δεδομένων (type casting)
llSay(0, "num = "+(string)num);
//κλήση στη συνάρτηση χρήστη
integer num2 = add(10,25);
llSay(0, "num2 = "+(string)num2);

//με το παρακάτω η μηχανή καταστάσεων αλλάζει κατάσταση
state anotherState;
}
}

//μια δεύτερη κατάσταση
state anotherState {
    touch_start(integer num_detected) {
        llSay(0, "Hello World");
    }
}
}

```

Αν δοκιμάσετε να γράψετε το πρόγραμμα αυτό στο παράθυρο επεξεργασίας και υπό την προϋπόθεση ότι δεν σας έχει ξεφύγει κάποιο συντακτικό λάθος, τότε το πρόγραμμα θα εκτελεστεί και το αντικείμενό σας θα στείλει τα παρακάτω μηνύματα:

```

Primitive: num=15
Primitive: num2=35

```

Κάτι άλλο που θα παρατηρήσετε είναι ότι μπορείτε πλέον να κάνετε κλικ πάνω στο αντικείμενό σας. Αν κλείσετε το παράθυρο επεξεργασίας του αντικείμενου και μεταφέρετε τον δείκτη του ποντικιού πάνω στο αντικείμενο, θα παρατηρήσετε ότι ο δείκτης άλλαξε σε χεράκι. Αν κάνετε κλικ στο αντικείμενο, θα στείλει το μήνυμα “Hello world”. Τη συμπεριφορά αυτή θα την επαναλαμβάνει κάθε φορά που θα του κάνετε κλικ.

Ας δούμε αναλυτικά τον παραπάνω κώδικα. Αν έχετε εμπειρία στον προγραμματισμό, θα αναγνωρίσετε την ομοιότητα με τη C και την Java και θα εντοπίσετε επίσης γνώριμες ενέργειες, όπως η κλήση συναρτήσεων και η μετατροπή τύπου δεδομένων (type casting). Στη συνέχεια θα παρουσιάσουμε τα βασικά στοιχεία του παραπάνω κώδικα, τα οποία καλύπτουν σχεδόν όλα τα τυπικά στοιχεία που μπορεί να βρει κάποιος σε ένα πρόγραμμα LSL.

Σχόλια: Υποστηρίζονται σχόλια γραμμής (//) και σχόλια τμήματος (/* */) όπως σε όλες σχεδόν τις σύγχρονες γλώσσες προγραμματισμού.

Σφαιρικές μεταβλητές: δηλώνονται στην αρχή του προγράμματος, πριν τη δήλωση οποιασδήποτε κατάστασης. Η δήλωση είναι της μορφής <τύπος δεδομένων> <όνομα> και μπορεί κατ’ επιλογή να ακολουθείται από ανάθεση τιμής, π.χ. `integer i, string s = "hello world"`. Οι μεταβλητές αυτές είναι προσβάσιμες από όλες τις καταστάσεις και η διάρκεια ζωής τους συμπίπτει με τη διάρκεια εκτέλεσης του προγράμματος.

Συναρτήσεις καθορισμένες από τον χρήστη (user-defined functions): Οι συναρτήσεις του χρήστη δηλώνονται επίσης στην αρχή του προγράμματος, συνήθως μετά τις σφαιρικές μεταβλητές, πριν τη δήλωση οποιασδήποτε κατάστασης. Όπως ακριβώς και οι σφαιρικές μεταβλητές, οι συναρτήσεις είναι προσβάσιμες από όλες τις καταστάσεις. Η δήλωσή τους είναι συντακτικά παρόμοια με τις αντίστοιχες δηλώσεις στις γλώσσες C και Java, με τη διαφορά ότι, αν η συνάρτηση δεν επιστρέφει κάποια τιμή, παραλείπεται το void από τη δήλωση. Για παράδειγμα, δηλώνουμε `typeMessage(string message)` αντί `void typeMessage(string message)`.

Καταστάσεις: δηλώνονται μετά τις σφαιρικές μεταβλητές και συναρτήσεις. Κάθε κατάσταση έχει το δικό της τμήμα κώδικα, το οποίο περιλαμβάνει μόνο διαχειριστές συμβάντων. Επομένως, δεν επιτρέπεται η δήλωση ένθετων καταστάσεων, δηλαδή μια κατάσταση να έχει δηλωθεί μέσα στο τμήμα κώδικα μιας άλλης. Η αρχική κατάσταση γράφεται ως default, ενώ οποιαδήποτε άλλη κατάσταση δηλώνεται ως state <όνομα>. Στον παραπάνω κώδικα δηλώσαμε μία δεύτερη κατάσταση με το όνομα anotherState.

Διαχειριστές συμβάντων: δηλώνονται εντός του τμήματος κώδικα μιας κατάστασης και μπορούν να περιέχουν ορίσματα. Στον παραπάνω κώδικα υπάρχουν δύο διαχειριστές καταστάσεων: `state_entry` στην κατάσταση default και `touch_start` στην κατάσταση anotherState.

Μετάβαση καταστάσεων: η εντολή `state <όνομα>` αναγκάζει το πρόγραμμα να μεταβεί στην αντίστοιχη νέα κατάσταση.

Ο νέος διαχειριστής συμβάντων που μόλις είδαμε στο παραπάνω πρόγραμμα είναι ο `touch_start`. Αυτή η συνάρτηση καλείται όταν ένας χρήστης κάνει κλικ πάνω στο αντικείμενο. Ουσιαστικά, αν ένας διαχειριστής `touch_start` δηλωθεί εντός μιας κατάστασης και η κατάσταση αυτή γίνει η τρέχουσα, τότε ο δείκτης του ποντικιού αυτόματα αλλάζει σε `χεράκι` για να δείξει ότι το αντικείμενο μπορεί να δεχτεί κλικ. Όπως μπορούμε να παρατηρήσουμε, η συνάρτηση δέχεται ένα όρισμα. Η πλήρης δήλωση είναι:

```
touch_start(integer num_detected)
```

Όταν κληθεί ο διαχειριστής, στη μεταβλητή του μοναδικού ορίσματος αποθηκεύεται ο αριθμός των χρηστών που έκαναν κλικ ταυτόχρονα στο αντικείμενο εκείνη τη στιγμή. Επομένως, στις περισσότερες περιπτώσεις η τιμή του `num_detected` θα είναι 1. Όπως θα δούμε αργότερα, ο αριθμός αυτός μπορεί να χρησιμοποιηθεί για τη λήψη πληροφοριών σχετικά με τους χρήστες που έκαναν κλικ στο αντικείμενο, για παράδειγμα τα ονόματά τους ή τη θέση τους στον χώρο.

Θα πρέπει να σημειώσουμε ότι όταν γράφουμε τον κώδικα των διαχειριστών συμβάντων δεν είναι υποχρεωτικό να χρησιμοποιούμε τα ονόματα των ορισμάτων που υπάρχουν στη δήλωσή τους, για παράδειγμα `num_detected`. Μπορούμε να χρησιμοποιήσουμε όποιο όνομα θέλουμε, αρκεί ο τύπος δεδομένων και η σειρά να είναι τα σωστά. Συνεπώς, μπορούμε για λόγους ευκολίας να γράψουμε `touch_start(integer num)` όταν υλοποιούμε αυτόν τον διαχειριστή συμβάντων.

Τέλος, σε ό,τι αφορά τη σύνταξη υπάρχουν δύο προσεγγίσεις για την τοποθέτηση των άγκιστρων: το άγκιστρο έναρξης (`{}`) να τοποθετείται δίπλα στην αντίστοιχη δήλωση ή να τοποθετείται σε νέα γραμμή. Στον κώδικα που δημιουργείται αυτόματα στα νέα προγράμματα LSL ακολουθείται η πρώτη προσέγγιση. Εντούτοις, στα παραδείγματα αυτού του βιβλίου θα ακολουθήσουμε τη δεύτερη.

Στη συνέχεια θα παρουσιάσουμε με μεγαλύτερη λεπτομέρεια τα δομικά στοιχεία και τη σύνταξη της γλώσσας LSL.

3.1 Τύποι δεδομένων, μεταβλητές και σταθερές

Η γλώσσα LSL υποστηρίζει τους παρακάτω τύπους δεδομένων: `integer`, `float`, `vector`, `rotation`, `key`, `string` και `list`. Ακολουθεί η επεξήγησή τους:

- Οι `integer` και `float` είναι αριθμητικοί τύποι δεδομένων για ακέραιους και δεκαδικούς αριθμούς αντίστοιχα, π.χ. 4 και 5.7.
- Σε μορφή `string` αποθηκεύεται μια συμβολοσειρά, δηλαδή μια ακολουθία χαρακτήρων. Η συμβολοσειρά γράφεται μέσα σε διπλά εισαγωγικά, π.χ. "hello world".
- Το κλειδί (`key`) είναι ένας μοναδικός προσδιοριστής για αντικείμενα, ενσαρκώσεις και άλλα στοιχεία του κόσμου. Ο τύπος δεδομένων `key` περιγράφει ένα κλειδί και η τιμή του δηλώνεται μέσα σε διπλά εισαγωγικά, π.χ. "01234567-89ab-cdef-0123-456789abcdef".
- Οι τύποι δεδομένων `vector` και `rotation` χρησιμοποιούνται κυρίως για γεωμετρικούς μετασχηματισμούς. Το `vector` είναι ένα τρισδιάστατο διάνυσμα που περιέχει τρεις αριθμούς τύπου `float`, και το `rotation` ένα quaternion που περιέχει τέσσερις αριθμούς τύπου `float`. Οι τιμές στους δύο αυτούς τύπους δεδομένων γράφονται μέσα στα σύμβολα μικρότερο (<) και μεγαλύτερο (>) και οι επιμέρους αριθμοί χωρίζονται με κόμμα, για παράδειγμα <4.5, 3.0, 1.0> και <0.0, 0.0, 0.0, 1.0> αντίστοιχα.
- Η λίστα είναι μια συλλογή ετερογενών τύπων δεδομένων και περιγράφεται στον τύπο δεδομένων `list`. Μια λίστα δηλώνεται μέσα σε αγκύλες ([]) και τα επιμέρους στοιχεία της χωρίζονται με κόμμα, για παράδειγμα [1, 4.5, "hello world", <3.0, 2.0, 1.0>].

Οι μεταβλητές στην LSL πρέπει να δηλώνονται ρητώς γράφοντας τον τύπο δεδομένων και το όνομά τους. Προαιρετικά μπορεί να ανατεθεί και αρχική τιμή στη δήλωση. Τα ονόματα των μεταβλητών μπορούν να ξεκινούν με γράμμα ή με κάτω παύλα (`underscore`, `_`) που ακολουθείται από γράμματα, αριθμούς ή κάτω παύλες. Για λόγους αναγνωσιμότητας βεβαίως θα ήταν καλύτερα να αποφεύγεται η κάτω παύλα στην αρχή

ενός ονόματος. Τα ονόματα μεταβλητών είναι ευαίσθητα σε κεφαλαία και μικρά, οπότε τα ονόματα number και Number αντιμετωπίζονται ως δύο διαφορετικές μεταβλητές. Παραδείγματα δήλωσης μεταβλητών:

```
integer num;  
float x = 4.57;  
string userName21 = "John Smith";
```

Σημειώστε ότι σε αντίθεση με τη C και την Java δεν μπορείτε να δηλώσετε παραπάνω από μία μεταβλητές του ίδιου τύπου δεδομένων στην ίδια δήλωση διαχωρίζοντάς τες με κόμμα, επομένως δεν μπορείτε να γράψετε κάτι σαν αυτό: integer i, j, k;

Οι μεταβλητές που δηλώνονται μέσα σε κάποιο τμήμα κώδικα, για παράδειγμα εντός ενός διαχειριστή συμβάντων ή εντός ενός βρόχου for, έχουν τοπικό εύρος. Αυτό σημαίνει ότι είναι προσβάσιμες μόνο από εντολές που βρίσκονται εντός του ίδιου τμήματος κώδικα μετά τη δήλωσή τους. Από την άλλη μεριά, οι σφαιρικές μεταβλητές που δηλώνονται στην αρχή του προγράμματος είναι προσβάσιμες από παντού. Αν μια τοπική και μια σφαιρική μεταβλητή έχουν το ίδιο όνομα, η τοπική έχει προτεραιότητα.

Η LSL περιλαμβάνει ακόμα έναν αριθμό από προκαθορισμένες σταθερές. Οι σταθερές χρησιμοποιούνται όπως οι μεταβλητές, με τη διαφορά ότι η τιμή τους είναι προκαθορισμένη και δεν αλλάζει. Από σύμβαση όλες οι σταθερές στην LSL γράφονται με κεφαλαία γράμματα. Για παράδειγμα, μια σταθερά είναι η PUBLIC_CHANNEL, η οποία έχει την τιμή 0. Επομένως, αντί να γράφουμε `ISay(0, "Hello")`, μπορούμε να γράψουμε το περισσότερο φιλικό προς τον αναγνώστη: `ISay(PUBLIC_CHANNEL, "Hello")`.

Μεταξύ των σταθερών της LSL περιλαμβάνονται και οι TRUE (τιμή 1) και FALSE (τιμή 0) οι οποίες μπορούν να χρησιμοποιηθούν σε λογικές εκφράσεις, καθώς δεν υπάρχει τύπος δεδομένων boolean.

Οι τελεστές που χρησιμοποιούνται στην LSL είναι αντίστοιχοι με αυτούς της Java και της C. Οι πιο συχνοί είναι:

- τελεστής ανάθεσης:
 - ανάθεση της τιμής μίας έκφρασης σε μία μεταβλητή: =, π.χ. `a=3+5`;
- αριθμητικοί τελεστές:
 - πρόσθεση, αφαίρεση, πολλαπλασιασμός, διαίρεση: +, -, *, /
 - ακέραιο υπόλοιπο: %
- τελεστές σύγκρισης:
 - μικρότερο, μεγαλύτερο: <, >
 - μικρότερο ή ίσο, μεγαλύτερο ή ίσο: <=, >=
 - ίσο: ==
- λογικοί τελεστές:
 - λογικό ΚΑΙ, λογικό Η: &&, ||
 - άρνηση: !
- τελεστές προσαύξησης:
 - προσαύξηση, μείωση κατά ένα: ++, --
 - πρόσθεση και ανάθεση, αφαίρεση και ανάθεση: +=, -=
- τελεστής συνένωσης:
 - συνένωση δύο συμβολοσειρών: +, π.χ. `s = "hello"+" "+"world"`;

Η μετατροπή μεταξύ τύπων δεδομένων γίνεται με τρόπο αντίστοιχο με τις σύγχρονες γλώσσες προγραμματισμού, δηλαδή γράφοντας τον τύπο μετατροπής σε παρένθεση πριν από την έκφραση. Με τον παραπάνω τρόπο μπορεί να γίνει μετατροπή από συμβολοσειρά σε άλλους τύπους δεδομένων, όπως integer, float, vector και quaternion, αλλά και αντίστροφα. Παραδείγματα:

```
integer num=32; string s = (string) num;  
float f = (float) "4.5 ";  
vector v = (vector) "<1,2,3>";
```

Τέλος, στην LSL μπορούμε να έχουμε και πρόσβαση στους επιμέρους αριθμούς ενός vector ή ενός rotation. Αυτό μπορούμε να το κάνουμε χρησιμοποιώντας τη σύνταξη: <διάνυσμα>.<όνομα στοιχείου>, όπου τα τρία στοιχεία του vector είναι τα x, y και z και του rotation τα x, y, z και s. Παράδειγμα:

```
vector v = <1.0, 0.5, 3.0>;
float x = v.x; // η τιμή του x θα γίνει 1.0
v.z = 2.0 // το διάνυσμα θα αλλάξει σε <1.0, 0.5, 2.0>
```

3.2 Συναρτήσεις, καταστάσεις και συμβάντα

Η LSL έχει μια μεγάλη βιβλιοθήκη έτοιμων συναρτήσεων οι οποίες μπορούν να χρησιμοποιηθούν στα προγράμματα. Έχουμε ήδη δει μία από αυτές: τη συνάρτηση llSay μέσω της οποίας μπορούμε να στέλνουμε μηνύματα σε ένα κανάλι επικοινωνίας. Όλα τα ονόματα των έτοιμων συναρτήσεων της LSL ξεκινάνε από σύμβαση με 'll', δηλαδή τα αρχικά της εταιρείας που την κατασκεύασε (Linden Labs). Αντίστοιχα, όλες οι επεκτάσεις που δημιουργήθηκαν για το OpenSimulator, δηλαδή η γλώσσα OSSL, ξεκινούν με 'os'.

Όπως αναφέρθηκε και προηγουμένως, οι χρήστες μπορούν να δημιουργήσουν και τις δικές τους συναρτήσεις. Οι συναρτήσεις των χρηστών πρέπει να δηλωθούν πριν από τη δήλωση καταστάσεων και μπορούν προαιρετικά να επιστρέφουν και κάποια τιμή. Στην περίπτωση που επιστρέφουν τιμή θα πρέπει να γραφτεί ο τύπος δεδομένων επιστροφής στη δήλωση της συνάρτησης και ο κώδικας να καταλήγει σε εντολή return ακολουθούμενη από έκφραση του ίδιου τύπου δεδομένων. Παραδείγματα συναρτήσεων του χρήστη:

```
sendGreeting(string userName) {
    llSay(0, "Hello, "+userName);
}
integer sum(integer a, integer b) {
    return a+b;
}
```

Κάθε κατάσταση του προγράμματος θα πρέπει να περιλαμβάνει τουλάχιστον έναν διαχειριστή συμβάντων. Για τη δήλωση μιας νέας κατάστασης, π.χ. με το όνομα my_state, ο κώδικας θα έχει την παρακάτω μορφή:

```
state my_state {
    state_entry() {
        //τι θα συμβεί όταν γίνει ενεργή αυτή η κατάσταση
    }
    //άλλα συμβάντα
}
```

Η LSL επιτρέπει τη δήλωση καταστάσεων σε οποιαδήποτε σειρά, αλλά συνήθως είναι μια καλή πρακτική να δηλώνονται όλες οι καταστάσεις κάτω από την αρχική (default). Η εντολή που προκαλεί μετάβαση σε μια άλλη κατάσταση είναι η state <όνομα νέας κατάστασης>, για παράδειγμα state my_state;

Τα συμβάντα δηλώνονται εντός του τμήματος κώδικα κάθε κατάσταση. Η LSL αναγνωρίζει 37 διαφορετικά συμβάντα που σχετίζονται με μια πληθώρα αντικειμένων, όπως μηνύματα, συγκρούσεις, κλικ ποντικιού, κοντινά αντικείμενα, αλλαγές σε ιδιότητες κ.λπ. Υπάρχει ακόμη και ένα χρονικό συμβάν για ενέργειες που θέλουμε να εκτελούνται κατ' επανάληψη ή με κάποια χρονοκαθυστέρηση.

3.3 Εντολές συνθήκης

Η LSL περιλαμβάνει εντολές εκτέλεσης υπό συνθήκη τύπου if – else όπως όλες οι γλώσσες προγραμματισμού. Η δήλωση είναι

```
if ( <συνθήκη> ) <διακλάδωση>;
```

όπου η διακλάδωση είναι είτε μία μοναδική εντολή είτε ένα τμήμα κώδικα. Εάν κατά την εκτέλεση της εντολής η συνθήκη αληθεύει, τότε η διακλάδωση θα εκτελεστεί. Αν θέλουμε να χρησιμοποιήσουμε και την else, η δήλωση θα γίνει:

```
if (<συνθήκη>) <διακλάδωση1> else <διακλάδωση2>;
```

όπου η εντολή ή το τμήμα εντολών της διακλάδωσης 1 θα εκτελεστεί εφόσον η συνθήκη αληθεύει, ενώ στην αντίθετη περίπτωση θα εκτελεστεί η διακλάδωση 2.

Στη συνέχεια θα δούμε ένα παράδειγμα στο οποίο ένα αντικείμενο αντιδρά ανάλογα με το όνομα του χρήστη ο οποίος έκανε κλικ πάνω του. Πριν δούμε τον κώδικα θα πρέπει πρώτα να παρουσιάσουμε μια δεύτερη συνάρτηση της LSL:

```
string llDetectedName (integer item);
```

Η συνάρτηση αυτή έχει νόημα μόνο σε συμβάντα που προκαλούνται από χρήστες, όπως το touch_start. Αν ένας αριθμός χρηστών, έστω N, προκάλεσαν το συμβάν, και η τιμή του ορίσματος item αναφέρεται σε έναν από τους χρήστες αυτούς (μεταξύ του 0 και του N-1), τότε η συνάρτηση επιστρέφει το όνομα του χρήστη ως συμβολοσειρά. Σε ό,τι αφορά την touch_start συγκεκριμένα, στις περισσότερες περιπτώσεις μόνο ένας χρήστης θα έχει κάνει κλικ στο αντικείμενο σε μια δεδομένη χρονική στιγμή (εκτός κι αν το αντικείμενο γίνει εξαιρετικά δημοφιλές), επομένως μπορούμε να μάθουμε το όνομά του καλώντας την llDetectedName για το αντικείμενο με αριθμό 0, δηλαδή llDetectedName(0).

Ο κώδικας του παραδείγματός μας είναι ο παρακάτω:

```
string myName = "Spyros Vos";

default {
    touch_start(integer num) {
        if(llDetectedName(0)==myName) {
            llSay(0, "Hello there!");
        }
        else {
            llSay(0, "I don't know you");
        }
    }
}
```

Όπως είναι προφανές από τον κώδικα, το αντικείμενο θα πει “Hello there!” αν ο χρήστης που έκανε κλικ πάνω του έχει το όνομα “Spyros Vos” και “I don’t know you” αν έχει οποιοδήποτε άλλο όνομα. Δοκιμάστε να αλλάξετε το όνομα σε αυτό της δικής σας ενσάρκωσης για να δείτε αν δουλεύει.

3.4 Εντολές επανάληψης

Η LSL υποστηρίζει τις γνωστές και συχνά χρησιμοποιούμενες εντολές επανάληψης do-while, while και for. Θα δούμε συνοπτικά τη σύνταξη και λειτουργία τους.

```
do <βρόχος> while (<συνθήκη>);
```

Ο βρόχος, που μπορεί να είναι είτε μία εντολή είτε ένα τμήμα εντολών, θα εκτελείται επαναληπτικά όσο η συνθήκη είναι αληθής. Η συνθήκη δεν ελέγχεται την πρώτη φορά που εκτελείται η εντολή, επομένως ο κώδικας του βρόχου θα εκτελεστεί σε κάθε περίπτωση τουλάχιστον μία φορά.

```
while (<συνθήκη>) <βρόχος>;
```

Η εντολή `while` είναι παραπλήσια με την `do-while`, με τη διαφορά ότι η συνθήκη ελέγχεται στην αρχή, επομένως υπάρχει πιθανότητα ο βρόχος να μην εκτελεστεί ποτέ, αν η συνθήκη είναι αρχικά ψευδής.

```
for (<έναρξη>; <συνθήκη>; <προσαύξηση>) <βρόχος>;
```

Στην εντολή `for` η έναρξη είναι μια εντολή που εκτελείται μία φορά στην αρχή, η συνθήκη ελέγχεται πριν την έναρξη κάθε βρόχου (και αν είναι ψευδής ο βρόχος δεν εκτελείται και η εντολή τερματίζεται), και η προσαύξηση είναι μια εντολή που εκτελείται μετά από κάθε βρόχο και πριν τον έλεγχο της συνθήκης. Η συνήθης χρήση της εντολής `for` είναι για την εκτέλεση ενός προκαθορισμένου αριθμού επαναλήψεων και χρησιμοποιείται ως εξής:

- Έναρξη: ορίζεται αρχική τιμή σε μια προϋπάρχουσα μεταβλητή, π.χ. `x=5`;
- Συνθήκη: ελέγχεται αν η μεταβλητή είναι εντός του απαιτούμενου εύρους, π.χ. `x < 10`;
- Προσαύξηση: αλλάζει η τιμή της μεταβλητής, π.χ. `x++`;

Αν χρησιμοποιήσουμε τις εντολές έναρξης, συνθήκης και προσαύξησης των παραπάνω παραδειγμάτων, τότε ο βρόχος θα τρέξει πέντε φορές και οι τιμές του `x` σε κάθε επανάληψη θα είναι: 5, 6, 7, 8 και 9.

Σημειώστε ότι σε αντίθεση με τη C ή την Java, ΔΕΝ μπορείτε να δηλώσετε νέα μεταβλητή στην έναρξη της `for`, για παράδειγμα η παρακάτω δήλωση είναι λάθος:

```
for (integer i=0; i<10; i++) { .. } //λάθος - δεν υποστηρίζεται
```

Ο παρακάτω κώδικας τυπώνει όλους τους μονούς αριθμούς μεταξύ 1 και 10.

```
default {
  state_entry() {
    integer i;
    string nums;
    for (i=1; i<=10; i+=2) {
      nums = nums + (string)i + " ";
    }
    llSay(0, nums);
  }
}
```

Πράγματι, αν εκτελέσετε το πρόγραμμα, η έξοδος θα είναι: 1 3 5 7 9

Παρατηρήστε ότι στον κώδικα του παραδείγματος χρησιμοποιήσαμε ένα `string` για να αποθηκεύσουμε όλες τις τιμές του `i`. Για να το πετύχουμε αυτό χρησιμοποιήσαμε συνένωση συμβολοσειρών και μετατροπή τύπου δεδομένων:

```
nums = nums + (string)i + " ";
```

4 Δράσεις των αντικειμένων

Στη συνέχεια θα δημιουργήσουμε πιο περίπλοκα και ενδιαφέροντα προγράμματα σε LSL. Προοδευτικά θα παρουσιάσουμε τις δυνατότητες της γλώσσας να αλλάζει δυναμικά την κατάσταση του αντικειμένου χρησιμοποιώντας τις έτοιμες συναρτήσεις του περιβάλλοντος. Οι υποενότητες είναι οργανωμένες με βάση την κατηγορία των ενεργειών. Θα εξετάσουμε την εμφάνιση κειμένου, τις αλλαγές στο χρώμα και στην υφή και τους γεωμετρικούς μετασχηματισμούς.

4.1 Κείμενο

Έχουμε ήδη παρουσιάσει τη συνάρτηση `llSay`, η οποία εξαναγκάζει το αντικείμενο να στείλει ένα μήνυμα σε ένα κανάλι επικοινωνίας. Εκτός από την `llSay` προσφέρονται δύο ακόμη παραπλήσιες εντολές, οι `llShout` και

llWhisper, και οι διαφορές τους, όπως προκύπτει από τα ονόματά τους, έχουν να κάνουν μόνο με την απόσταση στην οποία μπορεί να διαβαστεί το μήνυμα. Οι ορισμοί και οι αντίστοιχες αποστάσεις είναι:

```
llWhisper(integer channel, string msg); // απόσταση 10μ
llSay (integer channel, string msg); // απόσταση 20μ

llShout(integer channel, string msg); // απόσταση 100μ
```

Εκτός από την αποστολή μηνύματος στο δημόσιο κανάλι, τα αντικείμενα μπορούν να έχουν και μηνύματα τυπωμένα μόνιμα πάνω τους (βλ. εικ. 9.2). Αυτό μπορεί να συμβεί μέσω της παρακάτω συνάρτησης:

```
llSetText(string text, vector color, float alpha);
```

Τα ορίσματα της συνάρτησης είναι: το κείμενο που θα εμφανιστεί (text), το χρώμα του (color) και η διαφάνειά του (alpha). Θα αναφερθούμε αναλυτικά στο χρώμα και στη διαφάνεια στην επόμενη υποενότητα. Προς το παρόν, για το παράδειγμά μας θα χρησιμοποιήσουμε το λευκό χρώμα (<1.0, 1.0, 1.0>) και πλήρη αδιαφάνεια (1.0). Αν λοιπόν θέλουμε να εμφανίζεται με λευκό χρώμα το κείμενο “hello world” πάνω από το αντικείμενό μας, τότε αρκεί να γράψουμε την εντολή: llSetText(“hello world”, <1.0,1.0,1.0>, 1.0);



Εικόνα 9.2 Μήνυμα πάνω σε αντικείμενο του κόσμου.

Για να αφαιρεθεί το μήνυμα μπορούμε να χρησιμοποιήσουμε τις εξής τιμές: llSetText(“”, <0.0, 0.0, 0.0>, 0.0);

Στο παρακάτω παράδειγμα εμφανίζεται ένα μήνυμα πάνω από το αντικείμενό μας, το οποίο εξαφανίζεται μόλις κάνουμε κλικ στο αντικείμενο.

```
default {
    state_entry() {
        llSetText("click to remove this text",
            <1.0, 1.0, 1.0>, 1.0);
    }
    touch_start(integer num) {
        llSetText(“”, <0.0, 0.0, 0.0>, 0.0);
    }
}
```

4.2 Χρώματα και υφές

Τα χρώματα δηλώνονται στην LSL ως τρισδιάστατα διανύσματα (vectors). Οι τρεις τιμές του διανύσματος θα πρέπει να είναι μεταξύ του 0 και του 1 και αντιστοιχούν στις τιμές του κόκκινου, πράσινου και μπλε καναλιού στο χρωματικό μοντέλο RGB (βλ. 2^ο κεφάλαιο). Ακολουθούν ορισμένα ενδεικτικά χρώματα και οι τιμές τους:

```
vector white = <1.0, 1.0, 1.0>;
vector black = <0.0, 0.0, 0.0>;
vector red = <1.0, 0.0, 0.0>;
vector lightBlue = <0.5, 0.5, 1.0>;
```

Για να αλλάξουμε το χρώμα του αντικειμένου μας μέσω της LSL αρκεί να χρησιμοποιήσουμε την παρακάτω συνάρτηση:

```
llSetColor (vector color, integer face);
```

Το πρώτο όρισμα είναι το νέο χρώμα του αντικειμένου και το δεύτερο είναι η επιφάνεια στην οποία θα εφαρμοστεί. Αν θέλουμε το χρώμα να εφαρμοστεί σε όλες τις επιφάνειες του αντικειμένου, μπορούμε να χρησιμοποιήσουμε τη σταθερά ALL_SIDES στο δεύτερο όρισμα. Αν θέλουμε να αλλάξει σε μία μόνο επιφάνεια, θα πρέπει να δώσουμε τον αριθμό της επιφάνειας του αντικειμένου. Ο αριθμός αυτός εξαρτάται από το βασικό στερεό από το οποίο προήλθε. Στο παραλληλεπίπεδο (box) οι αριθμοί των επιφανειών σε σχέση με τον προσανατολισμό τους απεικονίζονται στον πίνακα 2.1. Επομένως, οι αριθμοί επιφάνειας 0 και 5 αναφέρονται στην πάνω και κάτω επιφάνεια του κύβου αντίστοιχα, και οι αριθμοί 1, 2, 3 και 4 στις πλαϊνές.

Αρ. επιφάνειας	Προσανατολισμός
0	+Z
1	-Y
2	+X
3	+Y
4	-X
5	-Z

Πίνακας 9.1 Συσχέτιση αριθμού επιφάνειας και προσανατολισμού της σε ορθογώνια παραλληλεπίπεδα.

Στο επόμενο παράδειγμα θα κάνουμε το αντικείμενό μας να αλλάξει προοδευτικά το χρώμα του από μαύρο σε κόκκινο. Προτού εξετάσουμε τον κώδικα θα πρέπει να παρουσιάσουμε μία ακόμα έτοιμη συνάρτηση:

```
llSleep(float sec);
```

Η συνάρτηση αυτή εξαναγκάζει την εκτέλεση του προγράμματος να σταματήσει (να «κοιμηθεί», όπως λέει το όνομά της) για τον αριθμό των δευτερολέπτων που δίνονται στο όρισμα. Για παράδειγμα, η llSleep(0.5) κάνει το πρόγραμμα να παγώσει για μισό δευτερόλεπτο. Ο κώδικας του παραδείγματός μας είναι ο παρακάτω:

```
default {
    touch_start(integer num) {
        float red;
        for(red=0; red<1.0; red+=0.1) {
            llSetColor(<red, 0.0, 0.0>, ALL_SIDES);
            llSleep(0.5);
        }
    }
}
```

Στο παραπάνω παράδειγμα αυξάνουμε προοδευτικά την τιμή του κόκκινου στο χρώμα για να προκαλέσουμε το απαιτούμενο αποτέλεσμα. Δείτε τη χρήση της μεταβλητής 'red'. Επιπλέον, εξαναγκάζουμε το πρόγραμμα να περιμένει για λίγο (μέσω της llSleep) μετά από κάθε αλλαγή χρώματος, έτσι ώστε το αποτέλεσμα να είναι ορατό.

Αν δοκιμάσατε να γράψετε και να εκτελέσετε τον κώδικα, θα παρατηρήσατε ότι, αν το αντικείμενο έχει ήδη κάποια υφή, όπως η υφή ξύλου που εμφανίζεται εξ ορισμού στα νέα αντικείμενα, τότε το οπτικό

αποτέλεσμα στην επιφάνειά του είναι ο συνδυασμός υφής και χρώματος. Όσο πιο ανοιχτό είναι το χρώμα, τόσο περισσότερο ορατή είναι η υφή.

Η υφή ενός αντικειμένου μπορεί να αλλάξει με μια διαδικασία παραπλήσια με την αλλαγή χρώματος. Χρησιμοποιείται η παρακάτω συνάρτηση:

```
llSetTexture(string texture, integer face);
```

Το πρώτο όρισμα είναι το όνομα της υφής και το δεύτερο η επιφάνεια στην οποία θα εφαρμοστεί. **Προσοχή:** η υφή με το όνομα που θα δηλώσετε θα πρέπει να υπάρχει στα περιεχόμενα του αντικειμένου. Για να τοποθετήσετε μια υφή στα περιεχόμενα του αντικειμένου ακολουθήστε τα παρακάτω βήματα:

1. Ανεβάστε την υφή στο αποθετήριό σας (από το μενού File > Upload Image).
2. Επεξεργαστείτε (Edit...) το αντικείμενό σας και επιλέξτε την καρτέλα των περιεχομένων (Content).
3. Σύρετε την υφή από το αποθετήριο στα περιεχόμενα του αντικειμένου.

Αν η υφή σας είναι κάποια εικόνα που θα εμφανιστεί σε παραλληλεπίπεδο, είναι συνήθως προτιμότερο να φαίνεται σε μία μόνο επιφάνεια του αντικειμένου. Επιλέξτε μία από τις πλαϊνές επιφάνειες (βλ. πίνακα) και χρησιμοποιήστε το αντίστοιχο νούμερο. Για παράδειγμα, η επιφάνεια στον θετικό άξονα X έχει τον αριθμό 2. Το παρακάτω παράδειγμα κάνει ένα αντικείμενο να μεταβάλλει την μπροστινή του όψη μεταξύ ενός βάτραχου και μιας τίγρης όταν γίνεται κλικ πάνω του (εικ. 9.3). Προφανώς, για να μπορέσετε να τρέξετε το παράδειγμα θα πρέπει να έχετε τοποθετήσει υφές με τα ίδια ονόματα στα περιεχόμενα του αντικειμένου σας.

```
//οι εικόνες "tiger" και "frog" θα πρέπει
//να υπάρχουν στα περιεχόμενα του αντικειμένου
string image;

default {
    state_entry() {
        image = "tiger";
        llSetTexture(image, 1);
    }
    touch_start(integer num) {
        if(image=="tiger")
            image = "frog";
        else image = "tiger";
        llSetTexture(image, 1);
    }
}
```



Εικόνα 9.3 Αλλαγή υφής του αντικειμένου μέσω κώδικα.

Τέλος, η διαφάνεια ενός αντικειμένου μπορεί να αλλάξει με την παρακάτω συνάρτηση:

```
llSetAlpha( float alpha, integer face);
```

Η τιμή του πρώτου ορίσματος (alpha) είναι ο βαθμός αδιαφάνειας και παίρνει τιμές μεταξύ του 0 (αόρατο) και του 1 (πλήρως αδιαφανές). Το δεύτερο όρισμα είναι ο αριθμός της επιφάνειας και χρησιμοποιείται με τον ίδιο ακριβώς τρόπο όπως με το χρώμα και την υφή.

4.3 Γεωμετρικοί μετασχηματισμοί

Τα αντικείμενά μας έχουν τη δυνατότητα να αλλάζουν τη θέση, τον προσανατολισμό και το μέγεθός τους μέσω συναρτήσεων της γλώσσας LSL. Στη συνέχεια θα παρουσιάσουμε με παραδείγματα κάθε τις εντολές για κάθε μία από τις τρεις αυτές κατηγορίες ενεργειών.

4.3.1 Μετατόπιση

Η θέση ενός αντικειμένου στον τρισδιάστατο χώρο μπορεί να αλλάξει με τη χρήση της παρακάτω συνάρτησης:

```
llSetPos( vector pos );
```

Η τιμή του ορίσματος είναι η *απόλυτη θέση* (δηλ. σε συντεταγμένες κόσμου) στην οποία θα μετατοπιστεί το αντικείμενο. Παρόλα αυτά, στις περισσότερες περιπτώσεις δεν είναι πρακτικό να δουλεύουμε με απόλυτες συντεταγμένες, για παράδειγμα μπορεί να χρειαστεί να μετατοπίσουμε το αντικείμενο κατά ένα μέτρο στον θετικό άξονα των X. Για να χρησιμοποιήσουμε σχετικές θέσεις θα πρέπει να γνωρίζουμε την τρέχουσα θέση του αντικειμένου, κάτι το οποίο μπορούμε να επιτύχουμε με την παρακάτω συνάρτηση:

```
vector llGetPos();
```

Η συνάρτηση επιστρέφει την τρέχουσα θέση του αντικειμένου. Στο σημείο αυτό θα πρέπει να επισημάνουμε ότι οι αριθμητικοί τελεστές μπορούν να χρησιμοποιηθούν και για πράξεις μεταξύ διανυσμάτων. Έτσι, με τους τελεστές + και - μπορούμε να προσθέτουμε και να αφαιρούμε διανύσματα αντίστοιχα. Κατά συνέπεια, αν θέλουμε να μετατοπίσουμε το αντικείμενό μας κατά ένα μέτρο στον θετικό άξονα των X, μπορούμε να γράψουμε την παρακάτω εντολή: `llSetPos(llGetPos() + <1.0, 0.0, 0.0>`). Δηλαδή η νέα απόλυτη θέση του αντικειμένου είναι η τρέχουσα θέση συν το διάνυσμα μετατόπισης.

Στο παρακάτω παράδειγμα το αντικείμενο μετακινείται κατά μισό μέτρο όταν γίνεται κλικ πάνω του.

```
default {
    touch_start(integer num) {
        llSetPos(llGetPos()+<0.5,0.0,0.0>);
    }
}
```

Ας δοκιμάσουμε στη συνέχεια να κατασκευάσουμε μια κάπως πιο σύνθετη συμπεριφορά για το αντικείμενό μας. Θα κάνουμε το αντικείμενο να κινείται κατά μισό μέτρο προς τη διεύθυνση του χρήστη που έκανε κλικ σε αυτό! Για να επιτύχουμε κάτι τέτοιο θα χρειαστεί πρώτα να εξετάσουμε ορισμένες ακόμα συναρτήσεις. Η πρώτη μας συνάρτηση μάς δίνει τη θέση του χρήστη:

```
vector llDetectedPos(integer number);
```

Η συνάρτηση αυτή είναι παραπλήσια με την `llDetectedName`, δηλαδή χρησιμοποιείται μόνο σε διαχειριστές συμβάντων που προκαλούνται από χρήστες, όπως το `touch_start`. Επιστρέφει το διάνυσμα της απόλυτης θέσης του χρήστη με αριθμό `number` που έκανε κλικ στο αντικείμενο.

Μια άλλη συνάρτηση η οποία θα μας είναι χρήσιμη είναι η:

```
vector llVecNorm(vector vec);
```

Η συνάρτηση αυτή επιστρέφει ένα νέο διάνυσμα το οποίο αποτελεί την κανονικοποίηση του διανύσματος που δίνεται ως όρισμα. Η κανονικοποίηση είναι ένα μοναδιαίο (μήκους 1) διάνυσμα που έχει την ίδια διεύθυνση με το αρχικό.

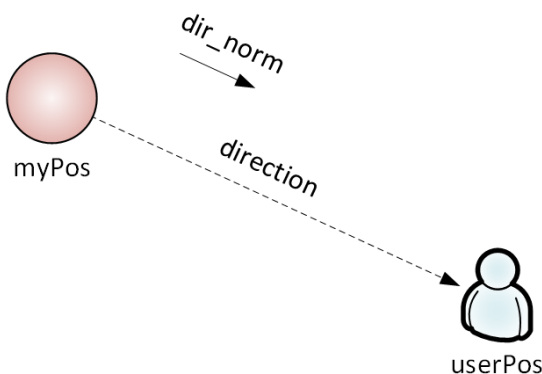
Τέλος, θα πρέπει να γνωρίζουμε ότι μπορούμε να πολλαπλασιάσουμε έναν αριθμό με ένα διάνυσμα χρησιμοποιώντας τον τελεστή του πολλαπλασιασμού. Για παράδειγμα `vector v = 0.5 * <1.0, 0.0, 2.0>;`

Ο κώδικας του παραδείγματός μας είναι ο παρακάτω:

```
default {
    touch_start(integer num) {
        vector userPos = llDetectedPos(0);
        vector myPos = llGetPos();
        vector direction = userPos - myPos;
        vector dir_norm = llVecNorm(direction);

        llSetPos(myPos+dir_norm*0.5);
    }
}
```

Όπως μπορείτε να δείτε στον παραπάνω κώδικα, αρχικά βρίσκουμε το διάνυσμα που συνδέει το αντικείμενο με τον χρήστη που έκανε κλικ πάνω του (μεταβλητή `direction`), στη συνέχεια το κανονικοποιούμε (μεταβλητή `dir_norm`) και τέλος πολλαπλασιάζουμε το κανονικοποιημένο διάνυσμα με την τιμή 0.5 ώστε να γίνει η αντίστοιχη μετατόπιση σε μέτρα (εικ. 9.4). Αν δεν θέλουμε η κίνηση του αντικειμένου να αλλάζει σε ύψος αλλά να παραμένει στο ύψος που είχε, τότε μπορούμε να μηδενίσουμε την τιμή του Z στη μεταβλητή `direction`. Αυτό μπορεί να γίνει αν προσθέσουμε πριν την κανονικοποίηση την εντολή `direction.z = 0;`



Εικόνα 9.4 Υπολογισμός του διανύσματος μετατόπισης και κανονικοποίησή του.

Δύο ακόμα χρήσιμες συναρτήσεις όταν γράφουμε κώδικα που σχετίζεται με τη μετατόπιση αντικειμένων είναι οι παρακάτω:

```
float llVecDist( vector a, vector v );
float llVecMag( vector v );
```

Η πρώτη μάς επιστρέφει την απόσταση μεταξύ δύο διανυσμάτων και η δεύτερη το μήκος ενός διανύσματος. Οι συναρτήσεις αυτές μπορούν να χρησιμοποιηθούν στην περίπτωση κινούμενων αντικειμένων για να εξετάσουμε αν το αντικείμενο έχει φτάσει στον προορισμό του. Αν για παράδειγμα ένα αντικείμενο μετατοπίζεται κατά μισό μέτρο μέχρι να φτάσει σε κάποιο προορισμό, έστω `target`, τότε έχει πλησιάσει αρκετά αν η απόσταση

μεταξύ του αντικειμένου και του target είναι μικρότερη του μισού μέτρου, δηλαδή `llVecDist(llGetPos(), target) < 0.5`.

4.3.2 Περιστροφή

Όταν επεξεργαζόμαστε ένα αντικείμενο στο Second Life ή στο OpenSimulator, παρατηρούμε ότι η περιστροφή του περιγράφεται με τη χρήση τριών γωνιών, των γωνιών Euler (βλ. Κεφ. 2). Παρόλα αυτά, οι περιστροφές στην LSL δεν περιγράφονται ως γωνίες Euler αλλά ως quaternions, μια επέκταση των μιγαδικών αριθμών χρήσιμη για τον συνδυασμό περιστροφών και τη συνθετική κίνηση. Η αναπαράσταση των περιστροφών ως quaternions γίνεται με τον τύπο δεδομένων rotation. Με βάση τα παραπάνω, μπορούμε να αλλάξουμε την περιστροφή ενός αντικειμένου ή να μάθουμε την τρέχουσα περιστροφή του με τις συναρτήσεις:

```
llSetRot( rotation rot );  
rotation llGetRot();
```

Το ενδιαφέρον χαρακτηριστικό των quaternions είναι ότι δύο περιστροφές μπορούν να συνδυαστούν αν τις πολλαπλασιάσουμε μεταξύ τους. Στην LSL μπορούμε να χρησιμοποιήσουμε τον τελεστή * για τον πολλαπλασιασμό των περιστροφών. Επομένως, αν θέλουμε το αντικείμενό μας να περιστραφεί σύμφωνα με μια περιστροφή που έχουμε αποθηκεύσει στη μεταβλητή rot, ο κώδικάς μας θα είναι:

```
llSetRot(llGetRot() * rot);
```

Δηλαδή, η νέα απόλυτη περιστροφή του αντικειμένου θα προκύψει από το συνδυασμό της τρέχουσας περιστροφής του και της σχετικής περιστροφής που θέλουμε να εφαρμόσουμε σε αυτό.

Το πρόβλημά μας είναι ότι δεν είναι εύκολο να ορίσει κάποιος περιστροφές απευθείας σε quaternions. Ευτυχώς για εμάς, η LSL μάς παρέχει δύο συναρτήσεις για τη μετατροπή από γωνίες Euler σε quaternions και αντίστροφα:

```
rotation llEuler2Rot( vector v );  
vector llRot2Euler( rotation quat );
```

Στην πρώτη συνάρτηση, περιγράφουμε τις γωνίες Euler ως τρισδιάστατο διάνυσμα, όπου στη θέση του X μπαίνει η περιστροφή στον άξονα των X κ.ο.κ. και μας επιστρέφει την περιστροφή. Στη δεύτερη, μια υπάρχουσα περιστροφή μετατρέπεται σε γωνίες Euler. Στο σημείο αυτό θα πρέπει να σημειώσουμε ότι, σε αντίθεση με το περιβάλλον επεξεργασίας, στην LSL οι γωνίες γράφονται σε ακτίνια (από 0 έως 2π) και όχι σε μοίρες. Επειδή συνήθως είναι ευκολότερο να σκέφτεται κάποιος σε μοίρες αντί σε ακτίνια, προσφέρεται μια σταθερά στην LSL η οποία μετατρέπει τις μοίρες σε ακτίνια αν πολλαπλασιαστεί με την τιμή. Η σταθερά αυτή είναι η DEG_TO_RAD.

Για να συνοψίσουμε τα παραπάνω, ας δούμε ένα παράδειγμα κώδικα που περιστρέφει ένα αντικείμενο κατά 30° στον άξονα των X κάθε φορά που γίνεται κλικ πάνω του.

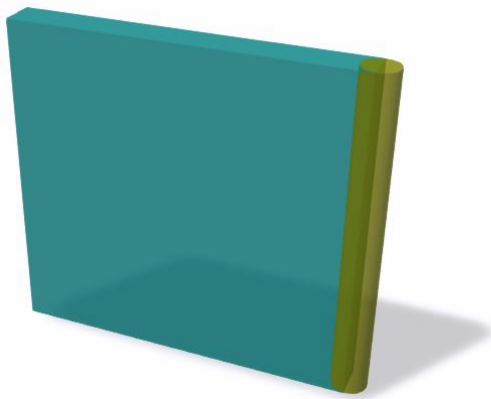
```
default {  
    touch_start(integer num) {  
        rotation rotChange = llEuler2Rot(<30*DEG_TO_RAD,0,0>);  
        llSetRot(llGetRot()*rotChange);  
    }  
}
```

Παρατηρήστε ότι στο παραπάνω παράδειγμα υπολογίζουμε ξανά την τιμή της μεταβλητής rotChange κάθε φορά που γίνεται κλικ στο αντικείμενο. Θα μπορούσαμε να αποφύγουμε τον επαναλαμβανόμενο υπολογισμό αν δηλώναμε τη μεταβλητή ως σφαιρική και κάναμε τον υπολογισμό μία φορά στην έναρξη της κατάστασης (state_entry). Δοκιμάστε το ως άσκηση.

Ως κέντρο περιστροφής ορίζεται το κέντρο του αντικειμένου. Εάν θέλουμε το αντικείμενό μας να περιστραφεί γύρω από διαφορετικό κέντρο, τα πράγματα γίνονται ελαφρώς πιο περίπλοκα, καθώς θα πρέπει,

εκτός από περιστροφή, να γίνει και μετατόπιση του αντικειμένου. Η ολοκληρωμένη προσέγγιση είναι να υπολογιστεί το σωστό διάνυσμα μετατόπισης μέσω διανυσματικών πράξεων και να αποδοθεί ο συνδυασμένος μετασχηματισμός στο αντικείμενο. Δυστυχώς, αν η περιστροφή και η μετατόπιση αποδοθούν με δύο ξεχωριστές εντολές (δηλ. `l1SetPos` και `l1SetRot`) το αποτέλεσμα οπτικά θα φαίνεται σαν δύο διαδοχικές μετακινήσεις του αντικειμένου. Η λύση είναι η χρήση της εντολής `l1SetPrimitiveParams` μέσω της οποίας μπορεί κάποιος να αλλάζει ταυτόχρονα πολλαπλά χαρακτηριστικά του αντικειμένου.

Μια εναλλακτική και αρκετά πιο απλή προσέγγιση για την αλλαγή του κέντρου περιστροφής είναι η χρήση ομάδων αντικειμένων (συνδεδεμένα αντικείμενα). Αν περιστρέψετε μια ομάδα αντικειμένων, τότε το κέντρο περιστροφής θα είναι το κέντρο της ρίζας της ομάδας, δηλαδή του τελευταίου αντικειμένου που επιλέξατε όταν δημιουργήσατε την ομάδα. Επομένως, μπορείτε να τοποθετήσετε ένα φανερό ή κρυφό αντικείμενο στο κέντρο περιστροφής που επιθυμείτε και να το συνδέσετε με το αντικείμενό σας κάνοντάς το ρίζα της ομάδας.



Εικόνα 9.5 Αλλαγή κέντρου περιστροφής μέσω ομαδοποίησης.

Για παράδειγμα, έστω ότι επιθυμείτε να κατασκευάσετε μια περιστρεφόμενη πόρτα. Μπορείτε να δημιουργήσετε μια ομάδα όπως αυτή της εικόνας 9.5, στην οποία η πόρτα συνδέεται με έναν κύλινδρο στον επιθυμητό άξονα περιστροφής. Αν ο κύλινδρος οριστεί ως ρίζα, τότε μπορείτε μέσω της LSL να την κάνετε να περιστρέφεται στον κάθετο άξονα (*Z*) ανοίγοντας και κλείνοντας όταν ο χρήστης κάνει κλικ πάνω της.

4.3.3 Αλλαγή κλίμακας

Η αλλαγή κλίμακας των αντικειμένων μπορεί να γίνει με τη χρήση των παρακάτω συναρτήσεων:

```
l1SetScale( vector size );  
vector l1GetScale( );
```

Η `l1SetScale` θα πρέπει να χρησιμοποιείται με προσοχή, καθώς τα πολύ μεγάλα αντικείμενα μπορούν να δημιουργήσουν προβλήματα στον κόσμο, για παράδειγμα να επικαλύψουν άλλα αντικείμενα ή να «παγιδέψουν» χρήστες. Όπως και στην περίπτωση της μετατόπισης και περιστροφής, η `l1SetScale` καθορίζει απόλυτη και όχι σχετική αλλαγή κλίμακας. Επομένως, αν θέλουμε, για παράδειγμα, να αυξήσουμε το μέγεθος ενός αντικειμένου κατά 10%, θα πρέπει να πολλαπλασιάσουμε την προσαύξηση με την τρέχουσα κλίμακα του αντικειμένου, η οποία δίνεται από την `l1GetScale()`. Η ελάχιστη τιμή κλίμακας που μπορεί να πάρει ένα αντικείμενο είναι $\langle 0.01, 0.01, 0.01 \rangle$, δηλαδή ένα κυβικό εκατοστό.

Στο παρακάτω παράδειγμα κατασκευάζουμε ένα αντικείμενο το οποίο αυξάνει το μέγεθός του κατά 10% κάθε φορά που γίνεται κλικ πάνω του. Για να το κάνουμε αυτό πολλαπλασιάζουμε την τρέχουσα κλίμακα με 1.1 και ορίζουμε το αποτέλεσμα ως νέα κλίμακα.

```
default {  
    touch_start(integer num) {
```

```

        vector myScale = llGetScale();
        llSetScale(myScale*1.1);
    }
}

```

Η αλλαγή κλίμακας δεν είναι υποχρεωτικό να είναι ομοιόμορφη. Μπορεί κάποιος να παραμορφώσει ένα αντικείμενο αν ορίσει διαφορετικές τιμές στους τρεις άξονες.

5 Παρακολούθηση αλλαγών του περιβάλλοντος

Μέχρι τώρα οι διάφορες δράσεις των αντικειμένων μας ενεργοποιούνταν αποκλειστικά με κλικ πάνω τους. Στην ενότητα αυτή θα παρουσιάσουμε διάφορα είδη συμβάντων που μπορούν να εντοπιστούν μέσω της LSL, ούτως ώστε να μπορούμε να δημιουργήσουμε περισσότερο περίπλοκες συμπεριφορές.

5.1 Κλικ του ποντικιού

Γνωρίζουμε ήδη ένα συμβάν μέσω του οποίου μπορούμε να αναγνωρίζουμε αν κάποιος χρήστης έκανε κλικ πάνω στο αντικείμενο:

```
touch_start(integer num_detected)
```

Ουσιαστικά το παραπάνω συμβάν ενεργοποιείται όταν ένας χρήστης ξεκινάει να κάνει κλικ στο αντικείμενο. Για όλη τη διάρκεια που κρατάει πατημένο το πλήκτρο του ποντικιού δημιουργούνται επαναλαμβανόμενα συμβάντα (ονομάζονται touch), ενώ όταν το πλήκτρο απελευθερώνεται δημιουργείται ένα ακόμα συμβάν, το touch_end. Οι δηλώσεις των δύο νέων συμβάντων είναι:

```
touch(integer num_detected)
touch_end(integer num_detected)
```

Στον παρακάτω κώδικα το αντικείμενό μας αυξάνει τον όγκο του όσο ο χρήστης κρατάει πατημένο το αριστερό πλήκτρο του ποντικιού. Προσέξτε να μην το παρακάνετε!

```

default {
    touch(integer num) {
        vector myScale = llGetScale();
        llSetScale(myScale*1.01);
    }
}

```

5.2 Παρακολούθηση μηνυμάτων κειμένου

Τα αντικείμενα μπορούν να ακούνε μηνύματα που στέλνονται στα κανάλια επικοινωνίας. Η εντολή που επιτρέπει σε ένα αντικείμενο να κάνει κάτι τέτοιο είναι η παρακάτω:

```
llListen (integer channel, string name, key id, string msg)
```

Το πρώτο όρισμα είναι το κανάλι στο οποίο θα ακούει για μηνύματα. Αν θέλουμε το αντικείμενό μας να καταγράφει τα μηνύματα που στέλνονται στο δημόσιο κανάλι επικοινωνίας θα χρησιμοποιήσουμε την τιμή μηδέν. Το δεύτερο όρισμα είναι το όνομα του χρήστη ή του αντικειμένου που θα στείλει το μήνυμα. Αυτή η επιλογή είναι χρήσιμη αν θέλουμε να περιορίσουμε το αντικείμενό μας να ακούει μηνύματα μόνο από συγκεκριμένους αποστολείς. Αν θέλουμε να ακούμε τα μηνύματα οποιουδήποτε αποστολέα, θα δηλώσουμε ένα κενό string, δηλαδή "", σε αυτήν την τιμή. Το τρίτο όρισμα (id) είναι το κλειδί του χρήστη ή αντικειμένου που θα στείλει το μήνυμα, το οποίο μπορεί να χρησιμοποιήσει κάποιος εναλλακτικά του ονόματος για να

περιορίσει τα μηνύματα σε αυτά που προέρχονται από συγκεκριμένο αποστολέα. Και πάλι, αν δεν θέλουμε κάποιον περιορισμό, χρησιμοποιούμε το κενό string. Το τελευταίο όρισμα (msg) είναι το μήνυμα που θέλουμε να ακούσουμε. Αυτή η τιμή έχει νόημα μόνο αν περιμένουμε να ακούσουμε κάποιο συγκεκριμένο μήνυμα. Και εδώ, αν θέλουμε να ακούμε οποιοδήποτε μήνυμα μπορούμε να δηλώσουμε το κενό string.

Σύμφωνα με τα παραπάνω, αν θέλουμε, για παράδειγμα, το αντικείμενό μας να ακούει μόνο μηνύματα “hello” στο δημόσιο κανάλι, η εντολή μας θα είναι `llListen(0, "", "", "hello")`, ενώ αν θέλουμε να ακούμε όλα τα μηνύματα του δημόσιου καναλιού θα γράψουμε `llListen(0, "", "", "")`;

Αν θέλουμε να συνδυάσουμε φίλτρα, για παράδειγμα να ακούμε και “hello” και “hi”, μπορούμε να καλέσουμε την `llListen` παραπάνω από μία φορές.

Μετά την κλήση της `llListen` και αν εντοπιστεί κάποιο μήνυμα που ταιριάζει στα φίλτρα που έχουμε θέσει, τότε καλείται ένα συμβάν ‘listen’. Μπορούμε να γράψουμε κώδικα να διαχειριστούμε το συμβάν αυτό ούτως ώστε να προγραμματίσουμε την αντίδραση του αντικειμένου μας στα εισερχόμενα μηνύματα. Η δήλωση του συμβάντος είναι:

```
listen (integer channel, string name, key id, string message)
```

Παρατηρούμε ότι τα ορίσματα του συμβάντος είναι τα ίδια με αυτά της `llListen`. Πράγματι, στην πρώτη μεταβλητή (channel) αποθηκεύεται το κανάλι στο οποίο εστάλη το μήνυμα, στη δεύτερη (name) το όνομα του χρήστη ή αντικειμένου που το έστειλε, στην τρίτη (id) το κλειδί του και στην τελευταία (message) το ίδιο το μήνυμα που ελήφθη.

Το παρακάτω πρόγραμμα κάνει το αντικείμενο να αντιδρά σε μηνύματα hello ή hi που στέλνουν οι χρήστες στο chat ανταποδίδοντας τον χαιρετισμό.

```
default {
    state_entry() {
        llListen(0, "", "", "hello");
        llListen(0, "", "", "hi");
    }
    listen(integer chn, string name, key id, string message) {
        llSay(0, "Hello, "+name);
    }
}
```

5.3 Προσέγγιση και σύγκρουση

Τα αντικείμενα στο Second Life ή στο OpenSimulator μπορούν να εντοπίσουν άλλα αντικείμενα ή χρήστες που βρίσκονται σε κοντινή απόσταση με τη χρήση της LSL. Η συνάρτηση εντοπισμού γειτονικών αντικειμένων είναι η παρακάτω:

```
llSensor( string name, key id, integer type, float range,
          float arc);
```

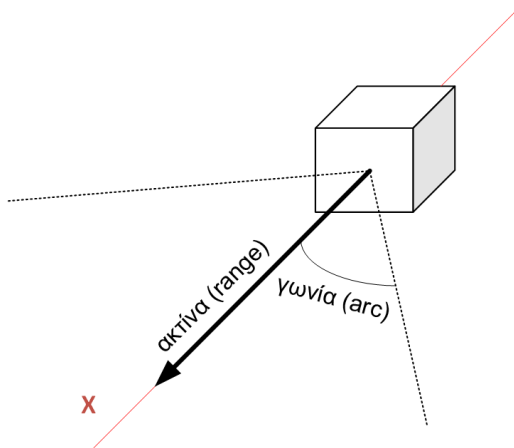
Το πρώτο όρισμα είναι το όνομα του αντικειμένου ή χρήστη για το οποίο θα ψάξει. Αν θέλουμε να μπορεί να ψάξει για αντικείμενα οποιουδήποτε ονόματος θα χρησιμοποιήσουμε κενό string (“”). Το δεύτερο όρισμα είναι το κλειδί (id) του αντικειμένου, όπου και πάλι μπορούμε να χρησιμοποιήσουμε κενό string αν δεν θέλουμε να περιορίσουμε την αναζήτηση σε συγκεκριμένο αντικείμενο.

Το τρίτο όρισμα (type) είναι ο τύπος των αντικειμένων για τα οποία θα ψάχνουμε. Οι τύποι εισάγονται ως σταθερές και οι επιτρεπτές τιμές είναι:

- AGENT – για χρήστες,
- ACTIVE – για φυσικά αντικείμενα, χρήστες και αντικείμενα που περιέχουν ενεργά προγράμματα (scripts),
- PASSIVE – για αντικείμενα χωρίς ενεργό πρόγραμμα,
- SCRIPTED – για αντικείμενα στα οποία τρέχει κάποιο ενεργό πρόγραμμα.

Οι παραπάνω τιμές μπορούν να συνδυαστούν με τον τελεστή δυαδικού Η (bitwise OR), ο οποίος γράφεται με το σύμβολο |. Αν για παράδειγμα θέλουμε να αναζητήσουμε όλα τα αντικείμενα, ανεξαρτήτως του αν τρέχουν κάποιο πρόγραμμα ή όχι, στον τύπο μπορούμε να θέσουμε την τιμή `PASSIVE | SCRIPTED`.

Το τέταρτο όρισμα (range) είναι η ακτίνα αναζήτησης σε μέτρα. Η μέγιστη επιτρεπτή ακτίνα είναι 96m. Το τελευταίο όρισμα είναι η γωνία αναζήτησης, η οποία δίδεται σε ακτίνια. Η γωνία αναζήτησης ορίζεται σε σχέση με το μπροστινό διάνυσμα του αντικειμένου που είναι εξ ορισμού ο θετικός άξονας των X στο τοπικό σύστημα συντεταγμένων. Αν η γωνία είναι π , η κάλυψη είναι σφαιρική, ενώ με $\pi/2$ καλύπτεται όλο το μπροστά ημισφαίριο (βλ. εικ. 9.6).



Εικόνα 9.6 Γωνία και ακτίνα του αισθητήρα απόστασης (sensor).

Η LSL μάς παρέχει δύο σταθερές στις οποίες μπορούμε να αναφερθούμε για να δηλώσουμε γωνίες π και $\pi/2$: `PI` και `PI_BY_TWO` αντίστοιχα. Φυσικά μπορούμε να χρησιμοποιήσουμε και οποιαδήποτε άλλη τιμή μεταξύ του 0 και του π .

Με βάση τα παραπάνω, αν θέλουμε να ψάξουμε για όλους τους χρήστες που βρίσκονται εντός της μέγιστης δυνατής απόστασης κάλυψης, η εντολή μας θα είναι:

```
llSensor("", "", AGENT, 96.0, PI);
```

Μετά την εκτέλεση της εντολής και εφόσον εντοπιστούν χρήστες ή αντικείμενα που ανταποκρίνονται στις ρυθμίσεις αναζήτησης, δημιουργείται ένα συμβάν με το όνομα `sensor`. Μπορούμε να δημιουργήσουμε διαχειριστή του συμβάντος αυτού με την παρακάτω μορφή:

```
sensor ( integer num_detected )
```

Η τιμή του `num_detected` είναι ο αριθμός των στοιχείων που εντοπίστηκαν. Ο μέγιστος αριθμός αποτελεσμάτων αναζήτησης είναι 16 στοιχεία. Αν βρεθούν παραπάνω, θα επιστραφούν μόνο τα 16 πρώτα που εντοπίστηκαν. Όπως ακριβώς συμβαίνει με τα συμβάντα `touch_start`, `touch` και `touch_end`, μπορούμε και εδώ να χρησιμοποιήσουμε τις συναρτήσεις `llDetectedName` και `llDetectedPos` για να πάρουμε παραπάνω πληροφορίες σχετικά με τα στοιχεία που εντοπίστηκαν.

Αν η αναζήτηση δεν επιστρέψει κανένα αποτέλεσμα, τότε θα δημιουργηθεί το συμβάν `no_sensor`. Η δήλωσή του είναι:

```
no_sensor()
```

Ας δούμε ένα παράδειγμα χρήσης του παραπάνω μηχανισμού ανίχνευσης:

```
default {
    touch_start(integer num) {
        llSensor("", "", AGENT, 10.0, PI);
    }
}
```

```

}

sensor(integer num) {
    integer i;
    for(i=0;i<num;i++) {
        llSay(0, "I see "+llDetectedName(i));
    }
}

no_sensor() {
    llSay(0, "I don't see anyone");
}
}

```

Στο παράδειγμα αυτό μόλις κάποιος κάνει κλικ πάνω στο αντικείμενο, αυτό θα απαντήσει με τα ονόματα όλων των χρηστών που βρίσκονται σε απόσταση το πολύ δέκα μέτρων. Αν δεν βρει κανέναν, θα απαντήσει με αντίστοιχο μήνυμα. Δοκιμάστε το!

Παρατηρήστε ότι η αναζήτηση γίνεται μόνο μία φορά, αμέσως μετά την κλήση της συνάρτησης. Αν θέλουμε η αναζήτηση να επαναλαμβάνεται σε καθορισμένα χρονικά διαστήματα, θα χρησιμοποιήσουμε την παρακάτω συνάρτηση:

```

llSensorRepeat(string name, key id, integer type, float
    range, float arc, float rate);

```

Τα ορίσματα είναι τα ίδια με της llSensor, εκτός από το τελευταίο, τον ρυθμό (rate). Ο ρυθμός είναι ο αριθμός δευτερολέπτων που θα μεσολαβούν μεταξύ δύο διαδοχικών αναζητήσεων. Δοκιμάστε να αλλάξετε τον κώδικα του παραπάνω παραδείγματος δίνοντας την παρακάτω εντολή αναζήτησης αντί της llSensor:

```

llSensorRepeat("", "", AGENT, 10.0, PI, 10.0);

```

Αν τρέξετε ξανά το πρόγραμμα, θα παρατηρήσετε ότι, αφότου κάνετε κλικ στο αντικείμενο, αυτό θα επαναλαμβάνει την αναζήτηση και θα παρουσιάζει τα αποτελέσματά του κάθε δέκα δευτερόλεπτα. Για να σταματήσουμε την επαναληπτική αναζήτηση θα πρέπει να χρησιμοποιήσουμε την παρακάτω εντολή:

```

llSensorRemove();

```

Αν η μέγιστη ακτίνα αναζήτησης (96 μέτρα) ή ο μέγιστος αριθμός αποτελεσμάτων (16) σας φαίνονται περιοριστικά για τα προγράμματά σας, μπορείτε να αυξήσετε τις τιμές αυτές παρεμβαίνοντας στο αρχείο Opensim.ini (βλ. Κεφ. 7) και αλλάζοντας τις τιμές των ιδιοτήτων SensorMaxRange και SensorMaxResults.

Τα αντικείμενα μπορούν επίσης να εντοπίσουν αν άλλοι χρήστες ή φυσικά αντικείμενα συγκρούστηκαν με αυτά. Η LSL προσφέρει τρία διαφορετικά συμβάντα για τη σύγκρουση. Όπως και στην περίπτωση του κλικ του ποντικιού, υπάρχει ένα συμβάν που καταδεικνύει την έναρξη της σύγκρουσης (collision_start), ένα επαναλαμβανόμενο όσο διαρκεί η σύγκρουση (collision) και ένα που ενεργοποιείται όταν σταματήσει η σύγκρουση (collision_end). Οι δηλώσεις των τριών συμβάντων είναι:

```

collision_start( integer num_detected)
collision      (integer num_detected)
collision_end  (integer num_detected)

```

Και πάλι ο αριθμός των χρηστών ή αντικειμένων που συγκρούονται ταυτόχρονα με το αντικείμενο αποθηκεύεται στη μεταβλητή num_detected και μπορούμε να χρησιμοποιήσουμε τις συναρτήσεις llDetectedName και llDetectedPos για να πάρουμε περισσότερα στοιχεία γι' αυτά. Θα δούμε ένα παράδειγμα στο οποίο το αντικείμενο προειδοποιεί όποιον χρήστη συγκρουστεί μαζί του και απομακρύνεται από αυτόν κατά δύο μέτρα!


```

default {
    collision_start(integer num) {
        llSay(0,"Ouch! Please be careful, "+llDetectedName(0));
        vector myPos = llGetPos();
        vector userPos = llDetectedPos(0);
        vector direction = myPos-userPos;
        vector dirNorm = llVecNorm(direction);
        dirNorm.z = 0;
        llSetPos(myPos+dirNorm*2);
    }
}

```

5.4 Χρονικά συμβάντα

Αν θέλουμε το αντικείμενό μας να εκτελεί επαναλαμβανόμενες ενέργειες στο περιβάλλον, μπορούμε να χρησιμοποιήσουμε έναν χρονιστή (timer). Η συνάρτηση για να πετύχουμε κάτι τέτοιο είναι:

```
llSetTimerEvent( float sec );
```

Στο όρισμα δίνονται τα δευτερόλεπτα μεταξύ δύο διαδοχικών επαναλήψεων. Μετά την εκτέλεση της εντολής θα παράγεται επαναληπτικά ένα συμβάν 'timer':

```
timer()
```

Μπορούμε να σταματήσουμε τον χρονιστή αν τον καλέσουμε με παράμετρο την τιμή 0:

```
llSetTimerEvent(0);
```

Στο επόμενο παράδειγμα, όταν κάνουμε κλικ στο αντικείμενο αυτό ξεκινάει να μετράει από το 1 μέχρι 10 δευτερόλεπτα και μετά σταματάει.

```

integer i = 0;

default {
    touch_start(integer num) {
        llSetTimerEvent(1.0);
    }

    timer() {
        i++;
        llSay(0, (string) i);
        if(i==10) {
            //μηδενισμός του μετρητή και
            //σταμάτημα χρόνου
            i=0;
            llSetTimerEvent(0);
        }
    }
}

```

6 Συμπεράσματα

Η LSL και οι επεκτάσεις της (OSSL) είναι μία γλώσσα σεναρίων ειδικού σκοπού για τον προγραμματισμό της συμπεριφοράς των αντικειμένων ενός εικονικού περιβάλλοντος. Πράγματι, η γλώσσα αυτή, όπως

παρατηρήσαμε, δεν περιλαμβάνει μεγάλο αριθμό τύπων δεδομένων, έχει μόνο μία μορφή συλλογών, τη λίστα, και δεν υποστηρίζει σύγχρονες προγραμματιστικές δομές, όπως κλάσεις και αντικείμενα. Αντίθετα, έχει ενσωματωμένη τη λειτουργία μιας μηχανής καταστάσεων, μιας δομής που συναντάται συχνά στη σχεδίαση παιχνιδιών, δίνοντας στον προγραμματιστή τη δυνατότητα να αναλύσει μια περίπλοκη συμπεριφορά σε έναν αριθμό από απλούστερες καταστάσεις λειτουργίας. Επιπλέον, είναι εξοπλισμένη με πολύ μεγάλο αριθμό έτοιμων συναρτήσεων που μπορούν να προσαρμόσουν την εμφάνιση και την κίνηση των αντικειμένων με πολλούς τρόπους: εμφανίζουν κείμενα, στέλνουν μηνύματα, αλλάζουν χρώματα και υφές, μετατοπίζονται, περιστρέφονται, μεγαλώνουν ή μικραίνουν κ.λπ. Τέλος, έχει τη δυνατότητα αναγνώρισης συμβάντων του περιβάλλοντος, όπως το κλικ από τον χρήστη, τα μηνύματα που ανταλλάσσονται, τους χρήστες και τα αντικείμενα που βρίσκονται κοντά κ.λπ. Η γλώσσα δεν μπορεί να θεωρηθεί εύκολη στην εκμάθηση, γιατί απαιτεί από τον προγραμματιστή να έχει καταλάβει πρώτα καλά τη λειτουργία του εικονικού κόσμου και να εντοπίσει ορισμένες ιδιαιτερότητές του. Το κέρδος όμως είναι ότι η γλώσσα μπορεί να οδηγήσει σε ενδιαφέρουσες συμπεριφορές από σχετικά μικρά σε μέγεθος προγράμματα.

Με τις γνώσεις που έχουμε αποκτήσει από αυτό το κεφάλαιο μπορούμε ήδη να φανταστούμε αρκετές περιπτώσεις διαδραστικών αντικειμένων που θα μπορούσαμε να υλοποιήσουμε. Γνωρίζουμε πώς να κάνουμε τα αντικείμενά μας να αντιλαμβάνονται την κατάσταση του περιβάλλοντος και γνωρίζουμε επίσης πώς να τροποποιούμε την εμφάνισή τους. Το επόμενο βήμα είναι να αξιοποιήσουμε τη φαντασία μας και να εκμεταλλευτούμε τις παραπάνω γνώσεις σε εύρηστα και αποδοτικά σενάρια. Τα παραδείγματα που εξετάσαμε και οι ασκήσεις που προτείνουμε για μελέτη και εξάσκηση δείχνουν ήδη προς αυτή την κατεύθυνση. Στο επόμενο κεφάλαιο θα εμπλουτίσουμε ακόμη περισσότερο τις γνώσεις και τις δεξιότητές μας στην LSL. Θα μάθουμε επιπρόσθετες, περισσότερο εξειδικευμένες λειτουργίες της γλώσσας, που μπορούν να αποτελέσουν βάση για σύνθετες εφαρμογές με συνδυασμένη συμπεριφορά πολλαπλών αντικειμένων και στοιχείων διεπαφής.

Σύνδεσμοι

LSL Portal: http://wiki.secondlife.com/wiki/LSL_Portal, επίσημος online ιστότοπος της LSL με αναλυτική τεκμηρίωση όλων των εντολών της.

LSL Wiki: <http://lslwiki.net/>, εναλλακτικός ιστότοπος για την LSL

Ασκήσεις

1. Γράψτε πρόγραμμα LSL με τρεις καταστάσεις: την αρχική (default), την κατάσταση A και την κατάσταση B. Όταν το πρόγραμμα ξεκινήσει, το αντικείμενο θα πει “I am in default state”. Αν γίνει κλικ πάνω του θα μεταβεί στην κατάσταση A και θα πει “I am in state A”, ενώ αν γίνει ξανά κλικ πάνω του θα μεταβεί στη B και θα πει αντίστοιχα “I am in state B”. Αν γίνει κλικ πάνω του για τρίτη φορά θα μεταβεί ξανά στην αρχική κατάσταση κ.ο.κ. Συμβουλή: Χρησιμοποιήστε τους διαχειριστές συμβάντων `state_entry` και `touch_start` σε καθεμία από τις τρεις καταστάσεις.
2. Γράψτε ένα πρόγραμμα με την παρακάτω συμπεριφορά: όταν ένας χρήστης κάνει κλικ στο αντικείμενο, το πρόγραμμα θα προσθέτει το όνομά του σε ένα string, εκτός αν είναι ο ιδιοκτήτης. Αν ο ιδιοκτήτης κάνει κλικ στο αντικείμενο, θα τυπώσει αυτό το string δείχνοντας τα ονόματα των υπόλοιπων χρηστών που έκαναν κλικ. Για παράδειγμα, έστω ότι ο ιδιοκτήτης είναι ο “Spyros Vos” και δύο άλλοι χρήστες ο “John Smith” και η “Mary Jane”. Ο John Smith κάνει κλικ μία φορά. Το αντικείμενο δεν αντιδρά. Η Mary Jane κάνει κλικ δύο φορές. Το αντικείμενο εξακολουθεί και δεν αντιδρά. Ο Spyros Vos κάνει κλικ και το αντικείμενο γράφει “John Smith Mary Jane Mary Jane”. Ο John Smith κάνει ξανά κλικ και τίποτα δεν συμβαίνει. Στη συνέχεια κάνει κλικ ο Spyros Vos και το

- αντικείμενο γράφει “John Smith Mary Jane Mary Jane John Smith”. Συμβουλή: το string που θα χρησιμοποιήσετε για την αποθήκευση των ονομάτων θα πρέπει να είναι σφαιρική μεταβλητή.
3. Κατασκευάστε ένα αντικείμενο το οποίο εμφανίζει με μόνιμο μήνυμα πάνω του το όνομα του τελευταίου ατόμου που του έκανε κλικ και τον συνολικό αριθμό των κλικ από όλους τους χρήστες, για παράδειγμα “last person: Spyros Vos, total clicks: 12”.
 4. Κατασκευάστε ένα αντικείμενο το οποίο είναι αρχικά λευκό. Όταν κάνει κάποιος κλικ πάνω του, το χρώμα του θα αλλάζει σε κόκκινο, με το επόμενο κλικ θα αλλάζει σε πράσινο, μετά σε μπλε και μετά πάλι σε κόκκινο. Θα έχει και ένα μόνιμο μήνυμα κειμένου πάνω του με το όνομα του χρώματός του. Δοκιμάστε να κάνετε την άσκηση με και χωρίς τη χρήση επιπλέον καταστάσεων (states).
 5. Ανεβάστε τέσσερις εικόνες μιας παρουσίασης με ονόματα “image1”, “image2” κ.λπ. Κατασκευάστε ένα παραλληλεπίπεδο στις διαστάσεις μιας μεγάλης επιφάνειας προβολής στην οποία θα εμφανίζεται η εικόνα “image1”. Αν γίνει κλικ πάνω της, θα πρέπει να αλλάζει στην επόμενη εικόνα της παρουσίασης, δηλαδή στην “image2”. Αν εμφανίζεται η τελευταία εικόνα και γίνει ξανά κλικ, θα πρέπει να επιστρέφει στην αρχική. Η λύση σας θα πρέπει να είναι όσο το δυνατόν πιο γενική. Δηλαδή θα πρέπει να μπορεί να δουλέψει με οποιονδήποτε αριθμό διαφανειών απλά αλλάζοντας την τιμή μιας σφαιρικής μεταβλητής με το όνομα numOfSlides.
 6. Κατασκευάστε ένα αντικείμενο το οποίο θα κινείται μισό μέτρο προς τον θετικό άξονα των Y κάθε φορά που θα γίνεται κλικ πάνω του. Αν η συνολική απόσταση που διάνυσε γίνει μεγαλύτερη των 3 μέτρων, τότε θα πρέπει να αλλάζει κατάσταση και να κινείται προς την αντίθετη κατεύθυνση μέχρι να φτάσει στην αρχική του θέση. Μετά θα πρέπει να επιστρέφει στην αρχική κατάσταση κινούμενο προς το θετικό Y ξανά κ.λπ. Συμβουλή: χρησιμοποιήστε μια σφαιρική μεταβλητή για να αποθηκεύετε την απόσταση που διάνυσε από την αρχική θέση το αντικείμενο.
 7. Χρησιμοποιήστε έναν βρόχο for και τη συνάρτηση II_Sleep() και κατασκευάστε ένα αντικείμενο το οποίο κινείται προοδευτικά προς τον χρήστη που έκανε κλικ πάνω του μέχρι να φτάσει στην ενσάρκωσή του.
 8. Κατασκευάστε μια περιστρεφόμενη πόρτα και γράψτε ένα πρόγραμμα ώστε η πόρτα να ανοίγει όταν γίνεται κλικ πάνω της και να κλείνει με το δεύτερο κλικ.
 9. Κατασκευάστε μια κόκκινη διάφανη σφαίρα. Όταν γίνεται κλικ πάνω της θα πρέπει να μεγαλώνει το μέγεθός της προοδευτικά μέχρι να τριπλασιαστεί. Όταν γίνεται ξανά κλικ θα πρέπει να επιστρέφει σταδιακά στο αρχικό της μέγεθος.
 10. Γράψτε ένα πρόγραμμα που ακούει συγκεκριμένα ονόματα χρωμάτων στο chat (π.χ. red, green και blue) και αλλάζει αντίστοιχα το χρώμα του αντικειμένου.
 11. Κατασκευάστε ένα αντικείμενο το οποίο θα εμφανίζει ως κείμενο (βλ. εντολή II_SetText) το τελευταίο μήνυμα που άκουσε στο δημόσιο κανάλι.
 12. Κατασκευάστε ένα απλό αντικείμενο-ραντάρ με δύο καταστάσεις (ανοικτό/κλειστό). Όταν θα ξεκινάει θα είναι κλειστό. Αν γίνει κλικ πάνω του θα ανοίξει αλλάζοντας το χρώμα του από γκρι σε πράσινο και θα ανακοινώνει τα ονόματα των γειτονικών χρηστών κάθε δέκα δευτερόλεπτα. Αν γίνει κλικ ξανά πάνω του θα επιστρέφει στην κατάσταση κλειστό, το χρώμα του θα ξαναγίνει γκρι και θα σταματήσει την αναζήτηση.
 13. Δημιουργήστε ένα εικονικό κατοικίδιο! Θα πρέπει να ψάχνει για τον ιδιοκτήτη του (χρησιμοποιήστε το όνομα του avatar σας στην αναζήτηση) κάθε δύο δευτερόλεπτα και να κινείται προς το μέρος του αν η απόσταση από αυτόν είναι μεγαλύτερη των δύο μέτρων.
 14. Κατασκευάστε μία περιστρεφόμενη πόρτα η οποία περιστρέφεται κατά 30° κάθε φορά που κάποιος συγκρούεται με αυτήν.
 15. Κατασκευάστε μια συρόμενη πόρτα (οριζόντια κίνηση χωρίς περιστροφή) η οποία ανοίγει όταν πλησιάσει κάποιος χρήστης (απόσταση μικρότερη των πέντε μέτρων) και κλείνει ξανά μετά από πέντε δευτερόλεπτα.
 16. Κατασκευάστε ένα μικρό αυτοκίνητο. Αν κάποιος γράψει “start” στο chat, αυτό θα πρέπει να κινείται επαναληπτικά προς τα εμπρός. Αν κάποιος κάνει κλικ πάνω του, θα πρέπει να περιστρέφεται κατά 90° αριστερά και να συνεχίσει να κινείται επαναληπτικά στη νέα του διεύθυνση. Αν κάποιος γράψει “stop” στο chat, θα πρέπει να σταματήσει να κινείται. Συμβουλή: χρησιμοποιήστε ένα διάνυσμα

διεύθυνσης για τη διαδοχική κίνηση του αυτοκινήτου, το οποίο θα προστίθεται στην τρέχουσα θέση του. Κάθε φορά που θα κάνετε κλικ, μαζί με το αυτοκίνητο θα περιστρέφετε και το διάνυσμα διεύθυνσης. Μπορείτε να πολλαπλασιάσετε την περιστροφή με το διάνυσμα.

Κεφάλαιο 10: Ανάπτυξη Σύνθετων Σεναρίων

Σύνοψη

Στο παρόν κεφάλαιο παρουσιάζονται ορισμένες σημαντικές επεκτάσεις και επιπλέον δυνατότητες της γλώσσας με στόχο την ανάπτυξη σύνθετων σεναρίων συμπεριφοράς σε πληθώρα εφαρμογών. Παρουσιάζονται θέματα όπως η επικοινωνία μεταξύ αντικειμένων μέσω ιδιωτικών καναλιών, η φυσική συμπεριφορά, η εκτέλεση συνθετικών κινήσεων, η υποστήριξη ειδικών αλληλεπιδράσεων με τον χρήστη κ.ά. Τέλος, παρουσιάζονται συνοπτικά και ορισμένες χρήσιμες ιδιότητες της γλώσσας, όπως η διαχείριση λιστών και η ανάγνωση και εγγραφή σε αρχεία κειμένου. Οι παραπάνω έννοιες επεξηγούνται με την παρουσίαση απλών ή και σύνθετων παραδειγμάτων περιβαλλόντων, για τα οποία παρουσιάζεται τόσο η κατασκευή των επιμέρους αντικειμένων όσο και ο κώδικας που τα συνοδεύει.

Προαπαιτούμενη γνώση

Απαιτείται η κατανόηση του μοντέλου προγραμματισμού και των βασικών εντολών της γλώσσας LSL που παρουσιάζονται στο Κεφάλαιο 9.

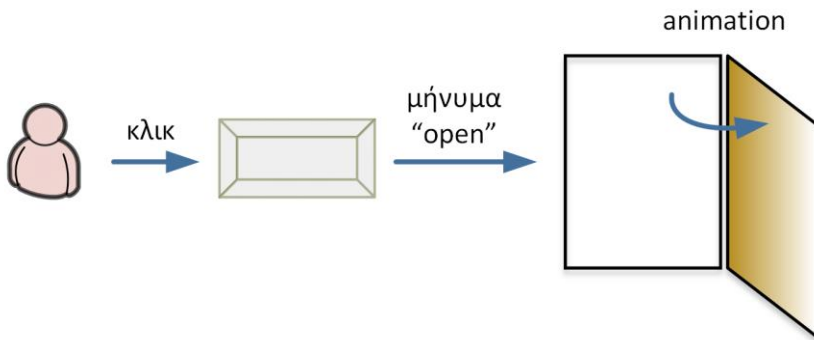
1 Εισαγωγή

Σε αυτό το κεφάλαιο θα παρουσιάσουμε ορισμένες επιπλέον δυνατότητες της γλώσσας LSL/OSSL τις οποίες μπορούμε να αξιοποιήσουμε για την ανάπτυξη πλούσιων διαδραστικών στοιχείων. Μέχρι τώρα έχουμε μάθει τις βασικές αρχές τις γλώσσας και ένα σύνολο εργαλείων, συναρτήσεων και συμβάντων για τον προγραμματισμό της συμπεριφοράς των αντικειμένων των οποίων η δράση ανταποκρίνεται στις αλλαγές του περιβάλλοντος. Στη συνέχεια θα μελετήσουμε έναν αριθμό από χρήσιμες επεκτάσεις των παραπάνω σε αρκετά διαφορετικές μεταξύ τους κατευθύνσεις. Αρχικά θα εξετάσουμε πώς μπορούμε να κάνουμε τα μεμονωμένα αντικείμενα να συνεργάζονται μεταξύ τους στα πλαίσια κάποιας ευρύτερης εφαρμογής. Στα πλαίσια αυτά παρουσιάζονται κατάλληλες λύσεις και παραδείγματα για την επικοινωνία μεταξύ αντικειμένων του κόσμου που μπορεί να οδηγήσει στη συντονισμένη λειτουργία τους. Επιπλέον, θα ανακαλύψουμε νέες δυνατότητες παραγωγής κίνησης στα αντικείμενα πέρα από την απλή αλλαγή θέσης καθώς και τεχνικές που ενεργοποιούν συνθετικές κινήσεις των ενσαρκώσεων. Στη συνέχεια θα εξετάσουμε θέματα φυσικής και θα δούμε πώς μπορούμε να αξιοποιήσουμε τον ενσωματωμένο μηχανισμό φυσικής μοντελοποίησης του OpenSimulator για να ασκήσουμε δυνάμεις και να προκαλέσουμε κίνηση βάσει φυσικών νόμων στα αντικείμενά μας. Επιπλέον, θα παρουσιάσουμε τεχνικές και παραδείγματα για τη δυναμική εισαγωγή νέων αντικειμένων στο περιβάλλον, μια διαδικασία η οποία εμφανίζεται αρκετά συχνά σε εικονικούς κόσμους και περιβάλλοντα παιχνιδιών. Θα εξετάσουμε πώς μπορούμε να υποστηρίξουμε και να διαχειριστούμε ορισμένες επιπλέον αλληλεπιδράσεις με τον χρήστη, όπως η παραγωγή διαλόγων μέσω ειδικών μενού, η σωστή στάση καθίσματος σε αντικείμενα που έχουμε κατασκευάσει και η δυναμική προσθήκη ενσωματώσεων στην ενσάρκωση χρήστη ή και στην οθόνη του. Τέλος, θα μελετήσουμε ορισμένες χρήσιμες συναρτήσεις για τη διαχείριση λιστών και αρχείων κειμένου, τα οποία μπορούν να φανούν χρήσιμα σε πολλές περιπτώσεις εφαρμογών.

Όλες οι νέες γνώσεις που θα αποκτήσουμε σε αυτό το κεφάλαιο προϋποθέτουν από τη μεριά σας την καλή κατανόηση των βασικών αρχών της γλώσσας και των συναρτήσεων και παραδειγμάτων που παρουσιάσαμε στο προηγούμενο κεφάλαιο. Είναι συνεπώς χρήσιμο, πριν μελετήσετε το παρόν κεφάλαιο, να έχετε βεβαιωθεί ότι είστε σε θέση να κατανοήσετε τα παραδείγματα που παρουσιάσαμε, να επιλύσετε τις ασκήσεις του κεφαλαίου και γενικότερα να διαχειριστείτε προβλήματα αντίστοιχου βαθμού δυσκολίας.

2 Επικοινωνία μεταξύ αντικειμένων

Μέχρι τώρα έχουμε δει πώς να κάνουμε ένα αντικείμενο να αντιδρά σε εξωτερικά συμβάντα και να εκτελεί διάφορες ενέργειες. Σε πολλές περιπτώσεις μπορεί να θέλουμε παρόλα αυτά να κάνουμε κλικ σε ένα αντικείμενο και, αντί γι' αυτό, να αντιδράει κάποιο άλλο. Για παράδειγμα, ο χρήστης να πατάει ένα πλήκτρο και να ανοίγει μια πόρτα (εικ. 10.1). Αυτό μπορεί να επιτευχθεί στην LSL μέσω επικοινωνίας αντικειμένων.



Εικόνα 10.1 Επικοινωνία μεταξύ δύο αντικειμένων του κόσμου.

Ο απλούστερος τρόπος να επικοινωνήσουν δύο αντικείμενα μεταξύ τους είναι μέσω κρυφών καναλιών επικοινωνίας: ένα αντικείμενο στέλνει μηνύματα σε κάποιο κανάλι και ένα άλλο ακούει μηνύματα στο ίδιο κανάλι. Έτσι, όταν στο παράδειγμά μας ο χρήστης κάνει κλικ στον αντικείμενο «κουμπί», αυτό θα στείλει ένα μήνυμα, για παράδειγμα “open”, σε ένα κρυφό κανάλι και το αντικείμενο «πόρτα» μόλις λάβει το μήνυμα θα εκτελέσει την απαιτούμενη ενέργεια.

Το μοναδικό μειονέκτημα αυτής της προσέγγισης είναι ότι θα πρέπει να αποφεύγονται κανάλια που χρησιμοποιούνται ήδη από άλλα αντικείμενα γιατί μπορεί να οδηγήσουν σε λάθη. Ευτυχώς ο αριθμός των καναλιών που προσφέρονται στο Second Life και στο OpenSimulator είναι πρακτικά άπειρος, αφού μπορεί να χρησιμοποιηθεί οποιοσδήποτε ακέραιος αριθμός. Κατά συνέπεια, αν χρησιμοποιούνται λόγου χάρη τριψήφιοι ή τετραψήφιοι αριθμοί, είναι πολύ μικρή η πιθανότητα να συμπέσουν δύο κανάλια.

Τέλος, θα πρέπει να θυμόμαστε ότι η εντολή llSay έχει συγκεκριμένη απόσταση μέχρι την οποία μπορεί να γίνει αντιληπτό το μήνυμα. Επομένως, αν τα δύο αντικείμενά μας απέχουν αρκετά, είναι προτιμότερο να χρησιμοποιείται η llShout για τη μεταξύ τους επικοινωνία.

Ας δούμε ένα παράδειγμα: Δημιουργούμε ένα αντικείμενο, έστω αντικείμενο A, και του τοποθετούμε τον παρακάτω κώδικα.

```
default {
    touch_start(integer num) {
        llSay(135, "touched");
    }
}
```

Το αντικείμενό μας λοιπόν στέλνει ένα μήνυμα “touched” στο κρυφό κανάλι 135. Προφανώς, αν κάνουμε κλικ πάνω του δεν θα δούμε κάτι να συμβαίνει. Ας κατασκευάσουμε και ένα δεύτερο αντικείμενο, έστω B, στο οποίο θα τοποθετήσουμε τον παρακάτω κώδικα:

```
default {
    state_entry() {
        llListen(135, "", "", "");
    }

    listen(integer chn, string name, key id, string message) {
        llSetColor(<1.0, 0.0, 0.0>, ALL_SIDES);
    }
}
```

Το αντικείμενό μας ακούει στο ίδιο κανάλι και, όταν λάβει ένα οποιοδήποτε μήνυμα, αλλάζει το χρώμα του σε κόκκινο. Αν έχουμε τα δύο αντικείμενά μας σε σχετικά κοντινή απόσταση και κάνουμε κλικ στο A, το αποτέλεσμα είναι να αλλάξει το χρώμα του B, διότι έλαβε το μήνυμα μέσω του κοινού καναλιού.

Μια καλή πρακτική για να αποφεύγουμε προβλήματα στην επικοινωνία είναι να χρησιμοποιούμε φίλτρα στην εντολή llListen, όπως για παράδειγμα να προσδιορίσουμε το όνομα του αποστολέα ή το

περιεχόμενο του μηνύματος. Σε μια τέτοια περίπτωση, τα υπόλοιπα μηνύματα που θα στέλνονται στο ίδιο κανάλι θα αγνοούνται.

Μια άλλη καλή πρακτική όταν κατασκευάζουμε αντικείμενα που επικοινωνούν μεταξύ τους είναι να τα δοκιμάζουμε πρώτα στέλνοντας εμείς οι ίδιοι τα μηνύματα που πρέπει να ακούσουν στο κρυφό κανάλι επικοινωνίας. Αυτή η διαδικασία θα διευκολύνει την αποσφαλμάτωση στην περίπτωση που στον κώδικα κάποιου από τα δύο αντικείμενα έχει υπεισέλθει κάποιο λογικό σφάλμα. Μπορούμε να στείλουμε μήνυμα σε οποιοδήποτε κανάλι αν στο πεδίο κειμένου (chat) γράψουμε *</αρ. καναλιού> <μήνυμα>*

Για παράδειγμα, αν έχουμε κατασκευάσει το αντικείμενο B και γράψουμε */135 touched* αυτό θα ανταποκριθεί με τον ίδιο ακριβώς τρόπο.

Υπάρχει και ένας δεύτερος τρόπος επικοινωνίας μεταξύ αντικειμένων, ο οποίος όμως λειτουργεί μόνο σε αντικείμενα της ίδιας ομάδας, δηλαδή συνδεδεμένα αντικείμενα. Η εντολή είναι:

```
llMessageLinked( integer link, integer num, string str,
                key id);
```

Το πρώτο όρισμα χρησιμοποιείται για να δηλώσει σε ποια από τα μέλη της ομάδας θα σταλεί το μήνυμα. Μπορεί κάποιος να τοποθετήσει απευθείας τον αριθμό του αντικειμένου στην ομάδα, αλλά είναι πρακτικότερο να χρησιμοποιήσει κάποια από τις παρακάτω σταθερές:

- LINK_ROOT: η ρίζα της ομάδας,
- LINK_SET: όλα τα μέλη της ομάδας,
- LINK_ALL_OTHERS: όλα τα μέλη εκτός από αυτό που έστειλε το μήνυμα,
- LINK_ALL_CHILDREN: όλα τα μέλη εκτός από τη ρίζα.

Τα υπόλοιπα τρία ορίσματα χρησιμοποιούνται για την επικοινωνία. Αν το αντικείμενο θέλει να στείλει έναν αριθμό, θα τον αποθηκεύσει στη num, κάποιο μήνυμα κειμένου στην str και το κλειδί κάποιου αντικειμένου ή χρήστη στην id. Κάθε φορά που στέλνεται ένα τέτοιο μήνυμα δημιουργείται αντίστοιχο συμβάν:

```
link_message( integer sender_num, integer num, string str,
              key id)
```

Η πρώτη μεταβλητή περιέχει τον αριθμό (θέση στην ομάδα) του αντικειμένου που έστειλε το μήνυμα. Αν είναι η ρίζα, ο αριθμός αυτός είναι το 1, ενώ για οποιοδήποτε άλλο αντικείμενο είναι μεγαλύτερος. Οι υπόλοιπες τρεις μεταβλητές περιέχουν τα αντίστοιχα δεδομένα που έστειλε ο αποστολέας μέσω της llMessageLinked.

Αν συνδέσουμε τα αντικείμενα A και B του προηγούμενου παραδείγματός μας σε μια ομάδα και θέλουμε να χρησιμοποιήσουμε αυτόν τον μηχανισμό μηνυμάτων, ο κώδικας του αντικειμένου A θα γίνει:

```
default {
    touch_start(integer num) {
        llMessageLinked(LINK_ALL_OTHERS, 0, "red", "");
    }
}
```

Και αντίστοιχα του αντικειμένου B:

```
default {
    link_message(integer sender, integer num, string str,
                key id) {
        llSetColor(<1.0, 0.0, 0.0>, ALL_SIDES);
    }
}
```

Παρατηρήστε ότι, αν κάνουμε το αντικείμενο A ρίζα της ομάδας, τότε μπορεί όλη η ομάδα να δεχτεί κλικ. Αυτό συμβαίνει διότι ένα πρόγραμμα που τρέχει στη ρίζα μιας ομάδας επηρεάζει όλη την ομάδα. Αν θέλουμε το A

να δουλεύει ως πλήκτρο και να είναι το μοναδικό στοιχείο της ομάδας που μπορεί να δεχτεί κλικ, θα πρέπει να κάνουμε το B ρίζα.

Το πλεονέκτημα της χρήσης της `IMessageLinked` για την επικοινωνία αντικειμένων είναι ότι αποφεύγεται η χρήση κρυφών καναλιών. Είναι όμως εφικτή μόνο αν τα αντικείμενα αποτελούν μέρος της ίδιας ομάδας.

3 Συνθετική κίνηση

Στη συνέχεια θα παρουσιάσουμε τρόπους με τους οποίους μπορούμε να δημιουργήσουμε ομαλή συνθετική κίνηση στα αντικείμενα και στους χαρακτήρες μέσω της LSL. Έχουμε ήδη δει στο προηγούμενο κεφάλαιο πώς να αλλάζουμε τη θέση, την περιστροφή και την κλίμακα ενός αντικειμένου. Οι αλλαγές αυτές όμως όταν εκτελούνται είναι «απότομες» και σε καμία περίπτωση δεν μπορούν να δώσουν την αίσθηση της ομαλής κίνησης, ακόμα κι αν εκτελεστούν επανειλημμένα με μικρό χρονικό βήμα. Θα εξετάσουμε σε αυτήν την ενότητα τρόπους με τους οποίους μπορούμε να προκαλέσουμε ομαλή περιστροφή ενός αντικειμένου γύρω από κάποιο άξονα, να ενεργοποιήσουμε μια προκατασκευασμένη συνθετική κίνηση της ενσάρκωσης του χρήστη και να ορίσουμε σύνθετη κίνηση αντικειμένων με πλάνα-κλειδιά (keyframing).

3.1 Ομαλή περιστροφή

Η LSL μάς δίνει τη δυνατότητα να προκαλέσουμε συνεχή ομαλή περιστροφή ενός αντικειμένου γύρω από έναν σταθερό άξονα και στην ταχύτητα με την οποία επιθυμούμε. Η συνάρτηση είναι:

```
llTargetOmega(vector axis, float spinrate, float gain);
```

Το πρώτο όρισμα είναι ο *άξονας περιστροφής*, ο οποίος θα πρέπει να είναι κανονικοποιημένος, και το δεύτερο (spinrate) η *ταχύτητα περιστροφής* σε ακτίνια το δευτερόλεπτο. Αν, για παράδειγμα, θέλουμε το αντικείμενό μας να κάνει μια πλήρη περιστροφή σε δέκα δευτερόλεπτα, η ταχύτητα θα πρέπει να είναι $2\pi/10$, δηλαδή περίπου 0.628. Το τρίτο όρισμα σχετίζεται με τη δύναμη που ασκείται στο αντικείμενο και αφορά μόνο τα φυσικά αντικείμενα, επομένως μπορούμε να χρησιμοποιήσουμε οποιαδήποτε μη μηδενική τιμή, για παράδειγμα 1.0.

Για να σταματήσουμε την περιστροφή, θα πρέπει να μηδενίσουμε όλες τις τιμές των τριών παραμέτρων:

```
llTargetOmega(<0, 0, 0>, 0, 0);
```

Στο παρακάτω παράδειγμα το αντικείμενό μας περιστρέφεται ομαλά γύρω από τον κάθετο άξονα με ταχύτητα 2 ακτίνια το δευτερόλεπτο όταν ο χρήστης κάνει κλικ πάνω του και για όση ώρα κρατάει το ποντίκι πατημένο.

```
default {
    touch_start(integer num) {
        llTargetOmega(<0.0, 0.0, 1.0>, 2.0, 1.0);
    }

    touch_end(integer num) {
        llTargetOmega(<0, 0, 0>, 0, 0);
    }
}
```

Το κέντρο περιστροφής ταυτίζεται και σε αυτήν την περίπτωση με το κέντρο του αντικειμένου. Αν θέλουμε να περιστρέψουμε το αντικείμενο γύρω από διαφορετικό κέντρο, μπορούμε να κατασκευάσουμε ομάδα αντικειμένων, στην οποία η ρίζα βρίσκεται στο επιθυμητό κέντρο (βλ. Κεφ. 9).

3.2 Κινήσεις ενσάρκωσης

Η ενσάρκωση του χρήστη μπορεί να εκτελέσει συνθετικές κινήσεις στον κόσμο οι οποίες γίνονται ορατές σε όλους τους χρήστες. Οι κινήσεις αυτές μπορεί να είναι είτε προκατασκευασμένες είτε να έχουν δημιουργηθεί από τον ίδιο τον χρήστη και να έχουν αποθηκευτεί στο αποθετήριό του. Η χρήση συνθετικών κινήσεων των ενσαρκώσεων είναι ένα χρήσιμο χαρακτηριστικό των εικονικών κόσμων, καθώς μπορούν να επαυξήσουν τη μη λεκτική επικοινωνία, για παράδειγμα να εκφράσει ο χρήστης τα συναισθήματά του σχετικά με κάτι που συμβαίνει αλλά και να οπτικοποιήσουν ανθρώπινες δραστηριότητες σε περιβάλλοντα εξιστόρησης ή παιχνίδια ρόλων. Η εντολή για να εκτελέσει η ενσάρκωση του χρήστη μια συνθετική κίνηση είναι:

```
llStartAnimation (string anim);
```

Στο όρισμα `anim` εισάγουμε το όνομα της συνθετικής κίνησης. Υπάρχει ένας μεγάλος αριθμός από έτοιμα `animations` στο `Second Life` και στο `OpenSimulator` που μπορούν να χρησιμοποιηθούν, για παράδειγμα “`bow`” (υπόκλιση), “`clap`” (χειροκρότημα), “`drink`” (πίνει κάποιο ποτό) κ.λπ. Στους συνδέσμους στο τέλος του κεφαλαίου μπορείτε να βρείτε μια ιστοσελίδα ([Second Life Wiki: Internal Animations](#)) που περιέχει τα ονόματα και τις περιγραφές όλων των έτοιμων `animations`.

Μπορείτε ακόμα να δημιουργήσετε τις δικές σας κινήσεις, να τις ανεβάσετε και να τις μεταφέρετε στα περιεχόμενα του αντικειμένου σας με τον ίδιο τρόπο που ακολουθήσαμε και στις υφές στο προηγούμενο κεφάλαιο. Σε αυτήν την περίπτωση θα δηλώσετε το όνομα του δικού σας `animation` στο όρισμα `anim`.

Ένα απλό εργαλείο για τη δημιουργία συνθετικών κινήσεων, το οποίο διατίθεται δωρεάν, είναι το `QAvatarator`. Μπορείτε να βρείτε περισσότερα στοιχεία σχετικά με τη δημιουργία κινήσεων και το ανέβασμά τους στο περιβάλλον του `OpenSimulator` στο Κεφάλαιο 8.

Όταν ένα αντικείμενο ενεργοποιεί κάποιο `animation` του χρήστη, ζητάει πρώτα την άδειά του για να το εκτελέσει (εικ. 10.2). Αυτή η απαίτηση υπάρχει στο `Second Life` για λόγους ασφάλειας και προστασίας από κακόβουλα προγράμματα. Η διαδικασία αυτή όμως κάνει τον κώδικα αρκετά περίπλοκο, γιατί πρέπει πρώτα να διασφαλιστεί ότι ο χρήστης έχει δώσει άδεια και μετά να κληθεί η `llStartAnimation`. Επιπλέον, σε ορισμένα σενάρια είναι κάπως άβολο για τον χρήστη να δίνει την άδειά του κάθε φορά που πρόκειται να εκτελέσει κάποια καινούρια κίνηση.



Εικόνα 10.2 *Διάλογος με τον χρήστη για την ενεργοποίηση κάποιου animation.*

Ευτυχώς για εμάς, η ομάδα ανάπτυξης του `OpenSimulator` έχει δημιουργήσει μια συνάρτηση εναλλακτική της `llStartAnimation`, η οποία παρακάμπτει την άδεια του χρήστη και κάνει την ενσάρκωση να εκτελεί άμεσα την κίνηση. Η συνάρτηση είναι η:

```
osAvatarPlayAnimation( key avatar, string animation );
```

Το πρώτο όρισμα είναι το κλειδί της ενσάρκωσης και το δεύτερο το όνομα του animation. Αν θέλουμε να βρούμε το κλειδί του χρήστη ο οποίος έκανε κλικ πάνω στο αντικείμενο, μπορούμε να χρησιμοποιήσουμε μια συνάρτηση παραπλήσια των llDetectedName και llDetectedPos. Η συνάρτηση είναι η παρακάτω:

```
key llDetectedKey(integer number);
```

Ακολουθεί ένα απλό πρόγραμμα το οποίο κάνει το avatar του χρήστη που έκανε κλικ στο αντικείμενο να κάνει μια υπόκλιση!

```
default {
    touch_start(integer num) {
        osAvatarPlayAnimation(llDetectedKey(0), "bow");
    }
}
```

Η κίνηση bow εκτελείται και στη συνέχεια η ενσάρκωση επιστρέφει στην αρχική της στάση. Αν παρόλα αυτά προσπαθήσετε να αλλάξετε το “bow” σε “sit” και ξαναδοκιμάσετε να κάνετε κλικ στο αντικείμενο, θα παρατηρήσετε κάτι περίεργο. Η κίνηση εκτελείται και στη συνέχεια η ενσάρκωση παραμένει καθιστή. Για να επιστρέψει στην αρχική κατάσταση η ενσάρκωση, θα πρέπει πρώτα να τερματιστεί η κίνηση. Αυτό γίνεται μέσω της συνάρτησης:

```
osAvatarStopAnimation(key avatar, string animation);
```

Επομένως, αν θέλουμε η ενσάρκωσή μας να παραμείνει σε καθιστή στάση για πέντε δευτερόλεπτα και στη συνέχεια να επιστρέψει στην αρχική της θέση, θα πρέπει να χρησιμοποιήσουμε κώδικα σαν τον παρακάτω:

```
default {
    touch_start(integer num) {
        osAvatarPlayAnimation(llDetectedKey(0), "sit");
        llSleep(5.0);
        osAvatarStopAnimation(llDetectedKey(0), "sit");
    }
}
```

Το κατά πόσο η κατάληξη ενός animation κρατάει τον χρήστη στην τελική στάση ή τον επαναφέρει στην αρχική εξαρτάται από τον βαθμό προτεραιότητας που έχει και από το αν περιλαμβάνει επανάληψη (loop) (βλ. Κεφ. 8). Δοκιμάστε να πειραματιστείτε με διάφορες κινήσεις.

3.3 Κίνηση με πλάνα-κλειδιά

Η LSL μάς δίνει επιπλέον τη δυνατότητα να ορίσουμε συνθετική κίνηση των αντικειμένων του κόσμου με πλάνα-κλειδιά (keyframing, βλ. Κεφ. 3), ελέγχοντας κατ’ αυτόν τον τρόπο πλήρως το αποτέλεσμα και τον χρονισμό της κίνησής τους. Μπορούμε να ορίσουμε έναν αριθμό από χρονικά σημεία και αντίστοιχα πλάνα-κλειδιά (με συγκεκριμένη θέση και προσανατολισμό του αντικειμένου σε καθένα από αυτά) και να ζητήσουμε από το αντικείμενο να εκτελέσει τη συνολική κίνηση χρησιμοποιώντας τεχνικές παρεμβολής. Η προκαθορισμένη ομαλή κίνηση που παράγεται με αυτήν την τεχνική μπορεί να φανεί ιδιαίτερα χρήσιμη στην περίπτωση που θέλουμε τα αντικείμενά μας να εκτελούν σύνθετες κινήσεις στο περιβάλλον χωρίς να επηρεάζονται από εξωτερικές συνθήκες, για παράδειγμα ένα όχημα που ξεναγεί τους επισκέπτες στο περιβάλλον.

Η συνάρτηση για την εκτέλεση κίνησης με θέσεις-κλειδιά είναι η παρακάτω:

```
llSetKeyframedMotion( list keyframes, list options );
```

Μπορούμε να παρατηρήσουμε ότι στη συνάρτηση αυτή για πρώτη φορά τα ορίσματα δίνονται σε μορφή λίστας. Αυτό είναι απαραίτητο, καθώς ο αριθμός των πλάνων-κλειδιών δεν είναι εκ των προτέρων γνωστός και έτσι θα πρέπει να αποθηκευτεί σε κάποια δομή συλλογής· στην περίπτωσή μας η μόνη προσφερόμενη τέτοια δομή είναι η λίστα. Η μορφή όμως με την οποία περιγράφονται τα πλάνα-κλειδιά (όρισμα keyframes) σχετίζεται και με τις ρυθμίσεις που θα δώσουμε στη συνάρτηση μέσω του δεύτερου ορίσματος (options).

Ας δούμε αναλυτικότερα τις διαθέσιμες ρυθμίσεις. Προσφέρονται τρία είδη ρυθμίσεων, καθένα εκ των οποίων θα πρέπει να συνοδεύεται από μία τιμή. Τα είδη αυτά είναι:

- **KFM_COMMAND**: εντολή χειρισμού της κίνησης. Οι δυνατές τιμές είναι:
 - **KFM_CMD_PLAY**: έναρξη της κίνησης ή συνέχιση μετά από παύση,
 - **KFM_CMD_PAUSE**: παύση της κίνησης,
 - **KFM_CMD_STOP**: τερματισμός της κίνησης.
- **KFM_MODE**: η μορφή εξέλιξης της κίνησης. Οι δυνατές τιμές είναι:
 - **KFM_FORWARD** (εξ ορισμού επιλεγμένη): εκτελείται η κίνηση μέχρι την ολοκλήρωσή της και σταματάει,
 - **KFM_LOOP**: μετά την ολοκλήρωσή της η κίνηση αρχίζει ξανά από την αρχή,
 - **KFM_PING_PONG**: μόλις ολοκληρωθεί η κίνηση, αρχίζει να εκτελείται στην αντίστροφη φορά μέχρι να φτάσει στην αρχή, μετά ξανά στην ορθή φορά κ.ο.κ.,
 - **KFM_REVERSE**: η κίνηση εκτελείται στην αντίστροφη φορά.
- **KFM_DATA**: το είδος των δεδομένων για κάθε πλάνο-κλειδί στη λίστα keyframes (πρώτο όρισμα της συνάρτησης). Οι δυνατές τιμές είναι:
 - **KFM_ROTATION**: η λίστα περιέχει μόνο περιστροφές,
 - **KFM_TRANSLATION**: η λίστα περιέχει μόνο μετατοπίσεις,
 - ο συνδυασμός τους, **KFM_TRANSLATION | KFM_ROTATION** (εξ ορισμού επιλεγμένη): η λίστα περιέχει μετατοπίσεις και περιστροφές.

Ο χρήστης μπορεί να δηλώσει στη λίστα options όποιες από τις τρεις ρυθμίσεις θέλει, με όποια σειρά θέλει, αρκεί να συνοδεύονται από τις αντίστοιχες τιμές. Επομένως, η λίστα options θα περιέχει δύο, τέσσερις ή έξι τιμές. Συνήθως χρησιμοποιούμε την **KFM_DATA** και την **KFM_MODE** όταν θέλουμε να δηλώσουμε νέα πλάνα-κλειδιά και να ξεκινήσει η κίνηση και την **KFM_COMMAND** όταν θέλουμε να παγώσουμε, να σταματήσουμε ή να ξαναξεκινήσουμε την κίνηση. Στην τελευταία περίπτωση δεν χρειάζεται να δηλωθούν ξανά τα πλάνα-κλειδιά, παραμένουν αποθηκευμένα τα τελευταία που έχουν δηλωθεί.

Το κάθε πλάνο-κλειδί που θα δηλώσουμε στη λίστα (όρισμα keyframes) έχει τη μορφή:

<θέση>, <προσανατολισμός>, <χρονική απόσταση>

Αν έχει αλλαχτεί το είδος των δεδομένων (βλ. **KFM_DATA**) σε **KFM_ROTATION** ή **KFM_TRANSLATION** θα πρέπει να παραλειφθεί η θέση ή ο προσανατολισμός αντίστοιχα.

Προσοχή: στην κίνηση με πλάνα-κλειδιά η θέση και ο προσανατολισμός δεν είναι απόλυτα αλλά σχετικά με την τρέχουσα θέση και προσανατολισμό του αντικείμενου! Επομένως, αν θέλουμε π.χ. το αντικείμενό μας να μετακινηθεί 2 μέτρα σε ύψος, θα δηλώσουμε τη θέση <0,0,2.0> στο αντίστοιχο πλάνο-κλειδί. Αν στη συνέχεια θέλουμε να μετακινηθεί άλλο 1 μ. (δηλαδή συνολικά 3 από την αρχική του θέση), θα δηλώσουμε τη θέση <0,0,1.0>.

Το ίδιο ισχύει και με τη χρονική απόσταση. Είναι η απόσταση σε δευτερόλεπτα σε σχέση με το προηγούμενο πλάνο, όχι ο συνολικός χρόνος που έχει περάσει από την έναρξη της κίνησης.

Ας δούμε ένα πρώτο παράδειγμα: Έστω ότι θέλουμε να κατασκευάσουμε ένα αντικείμενο το οποίο εκτελεί μια απλή κίνηση όταν γίνεται κλικ πάνω του. Ο παρακάτω κώδικας μετακινεί το αντικείμενο 1 μ. στον άξονα των Z για 1 δευτερόλεπτο και 1.5 μ. στον άξονα Y για ακόμη 2.

```
default {
    touch_start(integer num) {
        llSetKeyframedMotion([<0,0,1>, 1, <1.5,0,0>,
            2],[KFM_DATA, KFM_TRANSLATION]);
    }
}
```

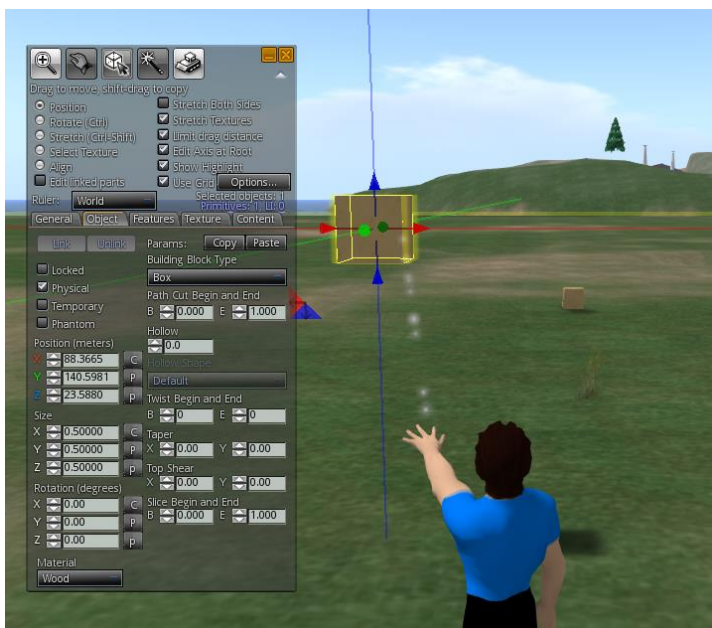
Στο επόμενο παράδειγμα θα κατασκευάσουμε επαναλαμβανόμενη κίνηση. Το αντικείμενό μας θα κινείται στα όρια ενός ορθογωνίου διαστάσεων 3 x 4 μ. στο οριζόντιο επίπεδο. Θα μπορούσε, για παράδειγμα, να είναι ένα όχημα που εκτελεί προδιαγεγραμμένη πορεία. Αν κάνουμε κλικ πάνω του θα παγώνει η κίνηση και με το επόμενο κλικ θα επανέρχεται.

```
integer paused = FALSE;

default {
  state_entry() {
    llSetKeyframedMotion([<4,0,0>,4, <0,3,0>, 3, <-4,0,0>, 4,
      <0,-3,0>, 3], [KFM_MODE, KFM_LOOP, KFM_DATA,
        KFM_TRANSLATION]);
  }

  touch_start(integer num) {
    if(!paused) {
      llSetKeyframedMotion([], [KFM_COMMAND, KFM_CMD_PAUSE]);
      paused = TRUE;
    }
    else {
      llSetKeyframedMotion([], [KFM_COMMAND, KFM_CMD_PLAY]);
      paused = FALSE;
    }
  }
}
```

Μπορείτε να παρατηρήσετε ότι στον παραπάνω κώδικα η χρονική απόσταση είναι ανάλογη της αντίστοιχης μετατόπισης. Αυτό έγινε εσκεμμένα, ώστε το αντικείμενό μας να κινείται με σταθερή ταχύτητα (1 μέτρου το δευτερόλεπτο στο συγκεκριμένο παράδειγμα).



Εικόνα 10.3 Ορισμός αντικείμενου ως φυσικού.

Αν το αντικείμενό μας έχει την όψη κάποιου οχήματος, ο παραπάνω κώδικας δεν αρκεί για να δώσει το απαιτούμενο αποτέλεσμα. Θα πρέπει επιπλέον να περιστρέφουμε το αντικείμενο κατά 90° στον άξονα Z κάθε φορά που αλλάζει διεύθυνση. Η αλλαγή στην περιστροφή θα πρέπει να ξεκινάει λίγο πριν τη χρονική στιγμή

αλλαγής διεύθυνσης και να ολοκληρώνεται λίγο μετά. Ο υπολογισμός των τιμών αυτών «χειροκίνητα» μπορεί να γίνει αρκετά επίπονος στην περίπτωση περίπλοκων κινήσεων και, για τον λόγο αυτόν, είναι προτιμότερο οι τιμές να υπολογίζονται αλγοριθμικά, π.χ. βάσει κάποιας εξίσωσης κίνησης.

4 Φυσική

Το OpenSimulator έχει έναν ενσωματωμένο μηχανισμό φυσικής (physics engine, βλ. Κεφ. 3) ο οποίος επιτρέπει σε επιλεγμένα αντικείμενα του περιβάλλοντος να έχουν μια προσομοιωμένη φυσική συμπεριφορά. Αν θέλουμε ένα αντικείμενό μας να υπακούει στους κανόνες της φυσικής, θα πρέπει να το επεξεργαστούμε και στην καρτέλα Object να επιλέξουμε Physical (εικ. 10.3).

Το αντικείμενο αποκτά μάζα και υπακούει στους νόμους της βαρύτητας, ροπής, τριβής κ.λπ. Μπορείτε να παρατηρήσετε ότι με το που θα κλείσετε το παράθυρο επεξεργασίας το αντικείμενο θα αρχίσει να κινείται προς το έδαφος, αν είχε ήδη κάποια απόσταση από αυτό, λόγω της επίδρασης της βαρύτητας. Κάθε φορά που κάνετε κλικ στο αντικείμενο αυτό και μετακινείτε το ποντίκι προς κάποια διεύθυνση, ασκείται πάνω του μια ανάλογη δύναμη και το αντικείμενο κινείται με φυσικό τρόπο. Αν μετακινήσετε την ενσάρκωσή σας προς το αντικείμενο, θα δείτε ότι μπορείτε να το σπρώξετε ή να το κλοτσήσετε!

4.1 Ομαλή μετακίνηση

Η πρώτη εντολή την οποία θα εξετάσουμε προκαλεί την κίνηση ενός φυσικού αντικειμένου προς κάποιο στόχο. Η κίνηση στην περίπτωση αυτήν είναι συνεχής και ομαλή, σε αντίθεση, για παράδειγμα, με την επαναλαμβανόμενη κλήση της `llSetPos` σε σύντομα χρονικά διαστήματα. Η παρακάτω εντολή κάνει το αντικείμενό μας να κινείται ομαλά και με σταθερή ταχύτητα προς κάποιο σημείο του κόσμου:

```
llMoveToTarget(vector target, float tau);
```

Το πρώτο όρισμα είναι η θέση-στόχος και το δεύτερο ο χρόνος σε δευτερόλεπτα που θα χρειαστεί το αντικείμενο για να φτάσει στον στόχο.

Ας δούμε ένα παράδειγμα: Κατασκευάστε ένα βασικό αντικείμενο (π.χ. σφαίρα ή κύβο), κάντε το φυσικό και προσθέστε τον παρακάτω κώδικα:

```
default {
    touch_start(integer num) {
        llMoveToTarget(llGetPos()+<4.0,0.0,0.0>, 3.0);
    }
}
```

Κάθε φορά που κάνετε κλικ στο αντικείμενο, κινείται τέσσερα μέτρα προς το θετικό άξονα των X.

4.2 Δυνάμεις

Μπορούμε να ασκήσουμε μια δύναμη σε ένα αντικείμενο με τη χρήση της παρακάτω εντολής:

```
llSetForce(vector force, integer local);
```

Το πρώτο όρισμα είναι το διάνυσμα της δύναμης. Στο δεύτερο επιλέγουμε αν το διάνυσμα αναφέρεται στο τοπικό σύστημα συντεταγμένων του αντικειμένου (αν η τιμή είναι TRUE) ή στο σφαιρικό (αν είναι FALSE). Μετά την κλήση της εντολής, η δύναμη ασκείται διαρκώς πάνω στο αντικείμενο. Αυτό δεν είναι πολύ πρακτικό και κάνει τα αντικείμενα να χάνονται εύκολα από τον κόσμο. Μια εναλλακτική, περισσότερο χρήσιμη, εντολή είναι η παρακάτω:

```
llApplyImpulse(vector force, integer local);
```

Η κλήση της είναι αντίστοιχη της `llSetForce`, μόνο που η δύναμη ασκείται στιγμιαία. Ας δοκιμάσουμε το παρακάτω: Κατασκευάστε μια σφαίρα με τις εξ ορισμού διαστάσεις (0.5 x 0.5 x 0.5) ως φυσικό αντικείμενο και προσθέστε το παρακάτω:

```
default {
    touch_start(integer num) {
        llApplyImpulse(<0, 0, 2.0>, FALSE);
    }
}
```

Αν κάνετε κλικ στη σφαίρα, θα δεχτεί μια στιγμιαία δύναμη προς τα πάνω. Θα κινηθεί γρήγορα προς τα πάνω και στη συνέχεια θα πέσει στην αρχική της θέση λόγω βαρύτητας. Μπορείτε φυσικά να πειραματιστείτε αλλάζοντας τη διεύθυνση της δύναμης, την ένταση ή και τις διαστάσεις της σφαίρας. Αν, για παράδειγμα, αλλάξετε την εντολή σε `llApplyImpulse(<3.0, 0.3, 0>, FALSE)` η σφαίρα θα κινηθεί διαγώνια σαν βλήμα.

Αν έχετε τρέξει τα παραδείγματα και πειραματιστεί αρκετά με τα φυσικά αντικείμενα, είναι πιθανό να παρατηρήσατε «παράλογες» κινήσεις και «αφύσικα» αποτελέσματα. Πράγματι, η μηχανή φυσικής του `OpenSimulator` δεν είναι το ίδιο καλής ποιότητας όπως άλλες αντίστοιχες μηχανές, για παράδειγμα σε κόσμους παιχνιδιών. Παρόλα αυτά, μπορείτε να βρείτε ενδιαφέρουσες χρήσεις του μηχανισμού φυσικής, όπως για την κατασκευή ειδικών εφέ, για παιχνίδια βολών κ.ά. Θα δούμε ορισμένα σχετικά παραδείγματα στην επόμενη ενότητα, όπου θα ασχοληθούμε με τη δυναμική εισαγωγή αντικειμένων στο περιβάλλον.

Τέλος, αξίζει να αναφερθεί ότι το `Second Life` έχει έναν αρκετά καλό μηχανισμό για τη δημιουργία οχημάτων, θαλάσσης, αέρος ή εδάφους, τα οποία οι χρήστες μπορούν να οδηγούν ελεύθερα στον κόσμο. Παρά το ότι οι συναρτήσεις για την κατασκευή οχημάτων μέσω `LSL` έχουν μεταφερθεί και στο `OpenSimulator`, τα αποτελέσματα δεν είναι το ίδιο καλά. Στους συνδέσμους αυτού του κεφαλαίου μπορείτε να βρείτε μια σελίδα με περισσότερες πληροφορίες για τα οχήματα (`Second Life Wiki: Linden Vehicle Tutorial`) και να πειραματιστείτε με αυτά.

5. Δυναμική εισαγωγή αντικειμένων

Με τη χρήση της `LSL` μπορούν να προστεθούν δυναμικά νέα αντικείμενα στο περιβάλλον. Τα αντικείμενα αυτά είναι υποχρεωτικά αντίγραφα υαρχόντων αντικειμένων. Η διαδικασία δημιουργίας αντιγράφων στο περιβάλλον ονομάζεται ‘rezzing’ στην ορολογία του `Second Life`. Η ίδια ακριβώς διαδικασία συμβαίνει όταν μεταφέρετε αποθηκευμένα αντικείμενα από το αποθετήριό σας στον κόσμο ή όταν δημιουργείτε αντίγραφα κατά τη διάρκεια της επεξεργασίας (βλ. Κεφ. 8). Ας υποθέσουμε ότι έχουμε ένα αντικείμενο `A` (π.χ. ένα κανόνι) και θέλουμε από αυτό να δημιουργείται ένα νέο αντικείμενο `B` (π.χ. το βλήμα) με τη χρήση της `LSL`. Για να συμβεί αυτό, θα πρέπει αρχικά το αντικείμενο `B` να βρίσκεται στα περιεχόμενα του `A`. Επομένως η διαδικασία είναι η παρακάτω:

1. Κατασκευάστε το αντικείμενο από το οποίο θέλετε να δημιουργούνται αντίγραφα (αντικείμενο `B`).
2. Δημιουργήστε ένα αντίγραφο του στο αποθετήριό σας (δεξί κλικ > More > Take Copy).
3. Ανοίξτε τα περιεχόμενα του άλλου αντικειμένου, αυτού στο οποίο θα τρέχει το πρόγραμμα (αντικείμενο `A`).
4. Σύρετε (drag) το αντίγραφο του αντικειμένου `B` από το αποθετήριό σας (θα το βρείτε στον κατάλογο `Objects`) στα περιεχόμενα του `A`.

Η εντολή που δημιουργεί αντίγραφα είναι η παρακάτω:

```
llRezObject(string inventory, vector pos, vector vel, rotation
rot, integer param);
```

Το πρώτο όρισμα είναι το όνομα του αντικειμένου που θα δημιουργηθεί. Θα πρέπει να βρίσκεται στα περιεχόμενα του αντικειμένου στο οποίο τρέχει το πρόγραμμα. Το δεύτερο είναι η θέση στην οποία θα τοποθετηθεί, σε απόλυτες συντεταγμένες κόσμου. Το τρίτο είναι η αρχική ταχύτητα που θα έχει, το οποίο ισχύει

μόνο σε φυσικά αντικείμενα. Το τέταρτο είναι η περιστροφή του. Αν δεν θέλετε να περιστραφεί το αντικείμενο, μπορείτε να χρησιμοποιήσετε τη σταθερά ZERO_ROTATION. Το τελευταίο όρισμα είναι απλά ένας αριθμός ο οποίος περνιέται ως μήνυμα στο αντικείμενο που θα αντιγραφεί. Μπορεί να αξιοποιηθεί στην περίπτωση που θέλουμε τα αντίγραφα μας να διαφοροποιούνται ως προς τη συμπεριφορά τους.

Το αντικείμενο-αντίγραφο μπορεί να τρέχει κι αυτό κάποιο πρόγραμμα. Στην περίπτωση αυτήν, όταν δημιουργηθεί το νέο αντίγραφο θα ενεργοποιηθεί το συμβάν:

```
on_rez ( integer start_param)
```

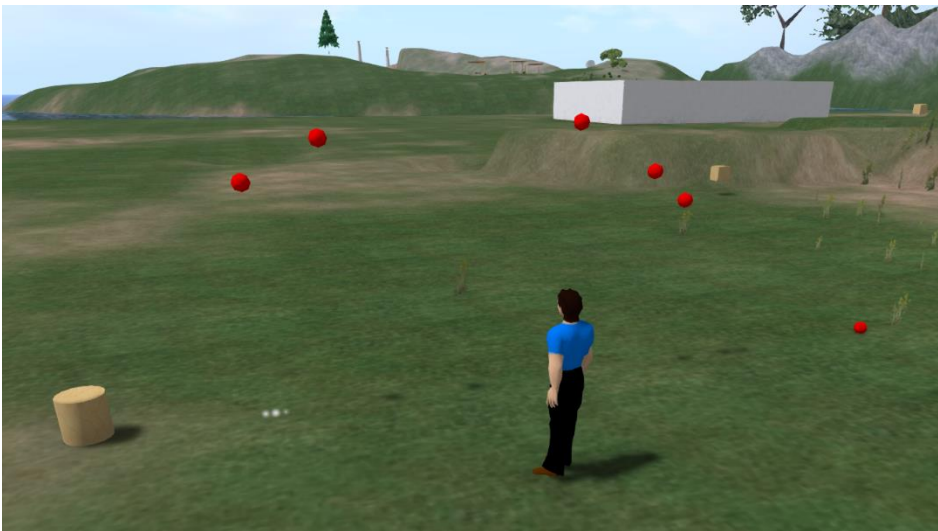
Η τιμή της start_param θα είναι αυτή που μεταφέρθηκε ως μήνυμα μέσω του τελευταίου ορίσματος της llRezObject.

Ας δούμε ένα πρώτο παράδειγμα χρήσης της δυναμικής εισαγωγής αντικειμένων: Κατασκευάστε μια μικρή σφαίρα και ονομάστε την 'ball'. Κάντε την φυσικό αντικείμενο και προσωρινό. Για να κάνετε ένα αντικείμενο προσωρινό, αρκεί να επιλέξετε την επιλογή Temporary στην καρτέλα Object. Η τελευταία επιλογή διασφαλίζει ότι το αντικείμενο θα διαγραφεί αυτόματα από τον κόσμο μετά από ένα λεπτό, προστατεύοντας κατ' αυτόν τον τρόπο τον κόσμο σας από υπερβολικά μεγάλο αριθμό αντικειμένων. Δημιουργήστε ένα αντίγραφο του αντικειμένου σας στο αποθετήριό σας.

Στη συνέχεια κατασκευάστε ένα δεύτερο αντικείμενο με το όνομα 'cannon'. Σύρτε τη σφαίρα στα περιεχόμενά του και γράψτε το παρακάτω πρόγραμμα.

```
default {
  touch_start(integer num) {
    llRezObject("ball", llGetPos()+<0,0,0.3>, <3.0, 0, 4.0>,
      ZERO_ROTATION, 0);
  }
}
```

Αν έχετε ακολουθήσει τα παραπάνω βήματα και κάνετε κλικ πάνω στο αντικείμενο-κανόνι, ένα νέο βλήμα θα εκτοξευτεί (εικ. 10.4). Μπορείτε να κάνετε συνεχόμενα κλικ και θα δείτε να εκτοξεύονται πολλαπλά βλήματα. Αν έχετε κάνει τα αντικείμενα προσωρινά, δεν χρειάζεται να ανησυχείτε για τον αριθμό τους.



Εικόνα 10.4 Εισαγωγή στον κόσμο πολλαπλών νέων αντικειμένων. Παράδειγμα κανονιού και βλήματος.

Στη συνέχεια θα κάνουμε μια μικρή μετατροπή στον κώδικά μας. Στόχος μας είναι η κίνηση της σφαίρας να ακολουθεί τον προσανατολισμό του κανονιού. Για να το πετύχουμε αυτό, θα ορίσουμε αρχικά μια ταχύτητα εκτόξευσης προς τη διεύθυνση του κανονιού (π.χ. αν έχει τη μορφή κυλίνδρου προς το θετικό Z) και στη συνέχεια θα περιστρέψουμε την ταχύτητα με την τρέχουσα περιστροφή του κανονιού. Η περιστροφή μπορεί να

γίνει απλά με πολλαπλασιασμό του quaternion της τρέχουσας περιστροφής του κανονιού με το διάνυσμα της ταχύτητας. Ο κώδικας είναι ο παρακάτω:

```
default {
    touch_start(integer num) {
        llRezObject("ball", llGetPos() + <0,0,0.3>*llGetRot(),
            <0.0,0.0,5.0>*llGetRot(),
            ZERO_ROTATION, 0);
    }
}
```

Θα παρατηρήσετε ότι, αν κάνετε κλικ πάνω στο κανόνι, το βλήμα θα κινηθεί κάθετα. Όμως, αν δοκιμάσετε να περιστρέψετε το κανόνι (π.χ. να κάνετε Edit το αντικείμενο και να αλλάξετε την περιστροφή του), θα δείτε ότι η σφαίρα εκτοξεύεται πάντα προς τη σωστή διεύθυνση.

Ας δούμε ένα ακόμη παράδειγμα: Θα δημιουργήσουμε ένα εφέ «συντριβανιού» εκτοξεύοντας πολλαπλά μικρά σωματίδια προς ελαφρώς διαφορετικές διευθύνσεις. Για να πετύχουμε το αποτέλεσμα της διαφοροποίησης, θα χρησιμοποιήσουμε τυχαιότητα. Η εντολή που παράγει ψευδο-τυχαίους αριθμούς στην LSL είναι η παρακάτω:

```
float llFrand ( float mag);
```

Η συνάρτηση επιστρέφει έναν τυχαίο αριθμό μεταξύ του 0 και του mag. Επομένως αν, για παράδειγμα, χρειαστεί να έχουμε έναν ψευδο-τυχαίο αριθμό μεταξύ του -1 και του 1, η εντολή μας θα είναι: float num = llFrand(2.0) - 1.0;

Στον παρακάτω κώδικα δημιουργούμε δέκα σφαίρες με ψευδοτυχαία τροποποίηση της ταχύτητας στις τιμές x και y. Το αποτέλεσμα είναι οι σφαίρες να μην εκτοξεύονται εντελώς κάθετα και να προκαλείται κίνηση που θυμίζει συντριβάνι.

```
default {
    touch_start(integer num) {
        vector vel;
        integer i;
        for(i=0;i<10;i++) {
            vel = <llFrand(3.0)-1.5, llFrand(3.0)-1.5, 3.0>;
            llRezObject("ball", llGetPos(), vel*llGetRot(),
                ZERO_ROTATION, 0);
        }
    }
}
```

6 Άλλες αλληλεπιδράσεις

Στη συνέχεια θα παρουσιάσουμε τρεις αρκετά διαφορετικές μεταξύ τους μορφές αλληλεπίδρασης με τον χρήστη. Αρχικά θα δούμε πώς μπορούμε να κάνουμε διαλόγους με τον χρήστη μέσω ειδικών μενού, στη συνέχεια πώς να προσαρμόζουμε τα αντικείμενά μας έτσι ώστε ο χρήστης να μπορεί να καθίσει με σωστό τρόπο σε αυτά και τέλος θα δούμε πώς μπορούμε να προσθέσουμε δυναμικά ενσωματώσεις στην ενσάρκωση ή στο HUD του χρήστη.

6.1 Διάλογοι με τον χρήστη

Μέσω της LSL μπορούμε να δημιουργούμε προσαρμοσμένα παράθυρα διαλόγου με τον χρήστη. Τα παράθυρα αυτά εμφανίζουν ένα μήνυμα και περιέχουν ένα σύνολο πλήκτρων ως δυνατές επιλογές του χρήστη. Όταν ο χρήστης κάνει μια από αυτές τις επιλογές, στέλνεται ένα μήνυμα με το όνομα του πλήκτρου σε κάποιο κρυφό κανάλι επικοινωνίας με στόχο να το διαχειριστεί στη συνέχεια το πρόγραμμα κατά το δοκούν.

Η εντολή δημιουργίας ενός παράθυρου διαλόγου είναι η παρακάτω:

```
llDialog( key avatar, string message, list buttons,  
integer channel);
```

Η πρώτη παράμετρος είναι το κλειδί του χρήστη στον οποίο θα εμφανιστεί ο διάλογος. Αυτό μπορεί να βρεθεί μέσω της llDetectedKey στην περίπτωση που ο χρήστης, για παράδειγμα, κάνει κλικ στο αντικείμενο. Η δεύτερη είναι το μήνυμα που θα εμφανιστεί στον χρήστη. Η παράμετρος buttons περιέχει μια λίστα με τα ονόματα των πλήκτρων που θα εμφανιστούν στον χρήστη, και, τέλος, στην παράμετρο channel τοποθετούμε τον αριθμό του καναλιού επικοινωνίας στο οποίο θέλουμε να σταλεί η επιλογή του χρήστη. Για να μπορούμε, βεβαίως, να κάνουμε το αντικείμενό μας να αντιδρά ανάλογα με την επιλογή του χρήστη στο παράθυρο διαλόγου, θα πρέπει πρώτα αυτό να ακούει στο σωστό κανάλι μέσω της llListen.



Εικόνα 10.5 Παράθυρο διαλόγου με τον χρήστη.

Στο παρακάτω παράδειγμα όταν γίνεται κλικ στο αντικείμενο εμφανίζεται ένας διάλογος με τα τρία βασικά χρώματα ως επιλογές. Ανάλογα με την επιλογή του χρήστη, αλλάζει αντίστοιχα και το χρώμα του αντικειμένου.

```
default {  
    state_entry() {  
        llListen(115, "", "", "");  
    }  
  
    touch_start(integer num) {  
        llDialog(llDetectedKey(0), "Select color",  
            ["red", "green", "blue"], 115);  
    }  
  
    listen(integer chn, string name, key id, string message) {  
        if(message=="red") llSetColor(<1.0,0.0,0.0>, ALL_SIDES);  
        else if(message=="green") llSetColor(<0.0,1.0,0.0>,  
            ALL_SIDES);  
        else if(message=="blue") llSetColor(<0.0,0.0,1.0>,  
            ALL_SIDES);  
    }  
}
```

Ο διάλογος που δημιουργείται από το πρόγραμμα αυτό απεικονίζεται στην εικόνα 10.5.

6.2 Κάθισμα σε αντικείμενα

Αν έχετε επισκεφθεί έτοιμους κόσμους στο Second Life ή σε κόσμους βασισμένους στο OpenSimulator, θα έχετε σίγουρα παρατηρήσει ότι υπάρχουν αντικείμενα στα οποία μπορεί η ενσάρκωσή σας να καθίσει σωστά. Αντίθετα, αν δοκιμάσετε να βάλετε την ενσάρκωσή σας να καθίσει σε ένα τυχαίο δικό σας αντικείμενο (δεξί κλικ > Sit here), θα παρατηρήσετε ότι ο τρόπος με τον οποίο κάθεται είναι λανθασμένος και «αφύσικος» (βλ. εικ. 10.6 αριστερά).

Η LSL μάς δίνει τη δυνατότητα να ελέγχουμε τον τρόπο με τον οποίο θα κάθεται η ενσάρκωση πάνω σε κάποιο αντικείμενο. Για να συμβεί αυτό, είναι απαραίτητο το αντικείμενο να περιλαμβάνει ένα πρόγραμμα το οποίο θα χρησιμοποιεί την παρακάτω εντολή:

```
llSitTarget(vector offset, rotation rot);
```

Το πρώτο όρισμα περιγράφει τη σχετική μετατόπιση της ενσάρκωσης σε σχέση με το κέντρο του αντικειμένου και το δεύτερο τη σχετική περιστροφή της σε σχέση με αυτήν του αντικειμένου. Δυστυχώς η εύρεση κατάλληλων τιμών για σωστό κάθισμα σε ένα τυχαίο αντικείμενο είναι μια διαδικασία συχνά επίπονη, η οποία ακολουθεί συνήθως πρακτικές δοκιμής και λάθους. Μια απλουστευμένη διαδικασία είναι να ξεκινήσουμε από ένα βασικό αντικείμενο, να φροντίσουμε ώστε η ενσάρκωση να κάθεται σωστά σε αυτό και στη συνέχεια να κατασκευάσουμε το σύνθετο αντικείμενό μας (π.χ. μια καρέκλα) ορίζοντας το αρχικό ως ρίζα της ομάδας.



Εικόνα 10.6 Αφύσικο και σωστό κάθισμα σε κατασκευασμένο αντικείμενο καρέκλα.

Ας δοκιμάσουμε ένα απλό παράδειγμα:

1. Κατασκευάστε ένα παραλληλεπίπεδο (box) με διαστάσεις 0.5 x 0.5 x 0.05 τοποθετημένο περίπου μισό μέτρο πάνω από το έδαφος.
2. Προσθέστε του το παρακάτω πρόγραμμα:

```
default {
    state_entry() {
        llSitTarget(<0.3, 0, 0.3>, ZERO_ROTATION);
    }
}
```

3. Κάντε την ενσάρκωσή σας να καθίσει στο αντικείμενο και βεβαιωθείτε ότι κάθεται σωστά. Αν όχι, προσαρμόστε την πρώτη τιμή (x) του offset για να την μετακινήσετε προς τα εμπρός ή πίσω και την τρίτη (z) για να αλλάξετε το ύψος.

4. Κατασκευάστε την υπόλοιπη καρέκλα και συνδέστε όλα τα επιμέρους τμήματα σε ομάδα, φροντίζοντας να είναι στη ρίζα της ομάδας (το τελευταίο που θα επιλέξετε) το αρχικό παραλληλεπίπεδο.

Αν ακολουθήσατε τα παραπάνω βήματα, θα μπορείτε να κάθεστε σωστά στην καρέκλα σας (βλ. εικ. 10.6 δεξιά).

6.3 Ενσωμάτωση αντικειμένων στο χρήστη

Σε προηγούμενο κεφάλαιο είδαμε πώς μπορούμε να κατασκευάζουμε αντικείμενα τα οποία ενσωματώνονται στο σώμα της ενσάρκωσης, για παράδειγμα αντικείμενα που μπορεί η ενσάρκωση να κρατήσει ή να φορέσει. Μέσω της LSL είναι δυνατόν να ενσωματωθεί δυναμικά ένα αντικείμενο στην ενσάρκωση, χωρίς να χρειαστεί ο ίδιος ο χρήστης να επιλέξει και να ρυθμίσει τις λεπτομέρειες της ενσωμάτωσης. Η επιλογή αυτή μπορεί να είναι χρήσιμη σε σενάρια στα οποία θέλουμε οι χρήστες μας να κρατούν ή να φορούν συγκεκριμένα αντικείμενα, για παράδειγμα παιχνίδια ρόλων, ή ακόμα και στην περίπτωση που θέλουμε να δημιουργήσουμε ένα ειδικό HUD για τους επισκέπτες μας και να ενσωματώνεται αυτόματα.

Όπως και στην περίπτωση των συνθετικών κινήσεων της ενσάρκωσης, οι εντολές της LSL για ενσωμάτωση αντικειμένων προϋποθέτουν ότι ο χρήστης θα πρέπει να αποδεχτεί τη διαδικασία μέσω κατάλληλου διαλόγου. Αν θέλουμε να παρακάμψουμε το στάδιο της αποδοχής, μπορούμε να χρησιμοποιήσουμε τις αντίστοιχες συναρτήσεις της OSSL, οι οποίες εκτελούνται άμεσα. Με την παρακάτω εντολή το αντικείμενο ενσωματώνεται στην ενσάρκωση του ιδιοκτήτη του:

```
osForceAttachToAvatar(integer attachmentPoint);
```

Στο όρισμα `attachmentPoint` τοποθετούμε το σημείο του σώματος στο οποίο θα γίνει η ενσωμάτωση. Προσφέρονται έτοιμες σταθερές για καθεμία από τις πιθανές θέσεις. Π.χ. `ATTACH_CHEST` για το στήθος, `ATTACH_HEAD` για το κεφάλι, `ATTACH_LHAND` και `ATTACH_RHAND` για το αριστερό και το δεξί χέρι αντίστοιχα, `ATTACH_HUD_BOTTOM_LEFT` για την κάτω αριστερή γωνία του HUD κ.λπ. Στους συνδέσμους του κεφαλαίου μπορείτε να βρείτε σελίδα ([Second Life Wiki: AttachToAvatar](#)) με όλα τα ονόματα των σταθερών για τις αντίστοιχες θέσεις ενσωμάτωσης.

Ας δούμε ένα απλό παράδειγμα: Κατασκευάστε έναν τόρο (torus) διαστάσεων 0.1 x 0.5 x 0.5. Προσθέστε του τον παρακάτω κώδικα:

```
default {
    touch_start(integer num) {
        osForceAttachToAvatar(ATTACH_HEAD);
    }
}
```

Αν στη συνέχεια κάνετε κλικ πάνω στο αντικείμενο θα το δείτε να τοποθετείται στο κεφάλι της ενσάρκωσής σας! Μπορείτε να το αφαιρέσετε με δεξί κλικ > Drop.

Αν θέλετε να μην ενσωματώνεται το ίδιο το αντικείμενο στο οποίο γίνεται κλικ, αλλά κάποιο άλλο, τότε μπορείτε να χρησιμοποιήσετε την παρακάτω εντολή:

```
osForceAttachToAvatarFromInventory(string itemName,
    integer attachmentPoint);
```

Στην πρώτη παράμετρο θα δώσετε το όνομα του αντικειμένου που θα ενσωματωθεί, το οποίο θα πρέπει να βρίσκεται στα περιεχόμενα (Contents) του αντικειμένου που τρέχει το script, και στη δεύτερη τη θέση ενσωμάτωσης.

Αν, τέλος, θέλετε η ενσάρκωση στην οποία θα ενσωματωθεί το αντικείμενο να είναι διαφορετική από τον ιδιοκτήτη του αντικειμένου, τότε θα χρησιμοποιήσετε την παρακάτω συνάρτηση με το μάλλον μακρύ όνομα:

```
osForceAttachToOtherAvatarFromInventory(string rawAvatarId,  
    string itemName, integer attachmentPoint);
```

Στην πρώτη παράμετρο θα περάσετε το κλειδί της ενσάρκωσης στην οποία θα γίνει η ενσωμάτωση.

Στο παρακάτω παράδειγμα, ένα αντικείμενο με όνομα "hud" που θα πρέπει να βρίσκεται στα περιεχόμενα του αντικειμένου σας θα ενσωματωθεί στο HUD οποιουδήποτε επισκέπτη κάνει κλικ πάνω του.

```
default {  
    touch_start(integer num) {  
        osForceAttachToOtherAvatarFromInventory (  
            llDetectedKey(0), "hud", ATTACH_HUD_BOTTOM_LEFT);  
        }  
    }  
}
```

7 Διαχείριση λιστών και αρχείων κειμένου

Σε αυτή την ενότητα θα εξετάσουμε δύο επιπλέον σημαντικές πτυχές της LSL, οι οποίες δεν σχετίζονται άμεσα με τον εικονικό κόσμο, αλλά διευκολύνουν τον προγραμματιστή στην ανάπτυξη περίπλοκων προγραμμάτων και διευρύνουν τις δυνατότητές του. Θα δούμε τρόπους με τους οποίους μπορούμε να εκμεταλλευτούμε τις λίστες στα προγράμματά μας, ώστε να διαχειριζόμαστε συλλογές δεδομένων, αλλά και πώς να φορτώνουμε και να αποθηκεύουμε δεδομένα σε αρχεία κειμένου τύπου notecard.

7.1 Λίστες

Όπως έχουμε ήδη αναφέρει, η LSL μάς παρέχει μία μοναδική δομή συλλογής δεδομένων, τη λίστα. Θα παρουσιάσουμε τις βασικές ενέργειες που μπορούν να γίνουν με τις λίστες καθώς και ορισμένα παραδείγματα εφαρμογής τους σε σενάρια του κόσμου.

Οι λίστες δηλώνονται μέσα σε αγκύλες και μπορούν να περιλαμβάνουν δεδομένα οποιουδήποτε τύπου εκτός από λίστες χωρισμένα με κόμμα. Μια λίστα θα μπορούσε να είναι η: ["hello", 1, 2.7, <1.0, 1.0, 2.0>, ""].

Ο τελεστής της πρόσθεσης (+) χρησιμοποιείται για να συνενώσει λίστες. Αυτός είναι και ο μοναδικός τρόπος με τον οποίο μπορούμε να προσθέσουμε ένα νέο στοιχείο σε μία υπάρχουσα λίστα. Τοποθετούμε το νέο μας στοιχείο σε μια προσωρινή λίστα και την προσθέτουμε στην υπάρχουσα. Για παράδειγμα:

```
list mylist = [1, 2, 3];  
mylist += [4]; // το στοιχείο 4 προστέθηκε στο  
               //τέλος της λίστας
```

Για να αφαιρέσουμε ένα ή περισσότερα στοιχεία από τη λίστα μπορούμε να χρησιμοποιήσουμε την παρακάτω συνάρτηση:

```
list llDeleteSubList(list src, integer start, integer end);
```

Το πρώτο όρισμα περιέχει την αρχική μας λίστα και τα υπόλοιπα δύο τις θέσεις έναρξης και τερματισμού. Η αρίθμηση των θέσεων ξεκινάει από το μηδέν, όπως στις περισσότερες γλώσσες προγραμματισμού. Από την αρχική λίστα διαγράφονται τα στοιχεία μεταξύ της θέσης έναρξης και της θέσης τερματισμού και το αποτέλεσμα επιστρέφεται ως νέα λίστα. Για παράδειγμα:

```
list myList = ["hello", "world", 1, 2, "hi"];  
myList = llDeleteSublist(myList, 1,3);  
//η myList θα γίνει ["hello","hi"]
```

Για να προσπελάσουμε ένα στοιχείο σε μια συγκεκριμένη θέση της λίστας, θα πρέπει να γνωρίζουμε τον τύπο δεδομένων του. Η LSL μάς δίνει τις παρακάτω συναρτήσεις προσπέλασης, ανάλογα με τον τύπο δεδομένων του στοιχείου:

```
integer llList2Integer( list src, integer index );
float llList2Float( list src, integer index );
string llList2String( list src, integer index );
vector llList2Vector( list src, integer index );
rotation llList2Rot( list src, integer index );
key llList2Key( list src, integer index );
```

Σε όλες τις παραπάνω συναρτήσεις το πρώτο όρισμα αναφέρεται στη λίστα στην οποία θα γίνει η αναζήτηση και το δεύτερο στη θέση του στοιχείου της λίστας που θα επιστραφεί.

Αν θέλουμε να διατρέξουμε όλη τη λίστα μπορούμε να χρησιμοποιήσουμε μια εντολή τύπου for. Θα πρέπει να γνωρίζουμε όμως τον αριθμό των στοιχείων της λίστας. Αυτός μπορεί να βρεθεί με τη συνάρτηση:

```
integer llGetListLength( list src );
```

Στο παρακάτω παράδειγμα διατρέχουμε μια λίστα από δεδομένα τύπου string, ελέγχουμε αν κάποιο από αυτά έχει την τιμή "hello" και τυπώνουμε σχετικό μήνυμα.

```
list myList = [ ... ];
integer i;
for(i=0;i<llGetListLength(myList);i++) {
    if(llList2String(myList,i)=="hello")
        llSay(0,"Found 'hello' in place "+(string)i);
}
```

Για να κάνουμε αναζήτηση σε λίστα, η LSL μάς παρέχει την παρακάτω συνάρτηση:

```
integer llListFindList( list src, list test );
```

Η πρώτη λίστα είναι αυτή στην οποία θέλουμε να ψάξουμε και η δεύτερη τα στοιχεία που αναζητούμε. Αν τα στοιχεία της δεύτερης λίστας μπορούν να εντοπιστούν συνεχόμενα και με την ίδια σειρά στην πρώτη, η συνάρτηση επιστρέφει τη θέση του πρώτου στοιχείου από αυτά. Διαφορετικά επιστρέφει -1. Συνεπώς, αν θέλουμε να ψάξουμε για ένα μεμονωμένο στοιχείο, θα πρέπει να δημιουργήσουμε μια νέα λίστα που να περιέχει μόνο αυτό και να την περάσουμε ως όρισμα στην llListFindList. Για παράδειγμα:

```
list myList = ["a", "b", "c", "d", "e"];
integer i = llListFindList(myList, ["c","d"]);
//η τιμή του i θα γίνει 2
```

Για να κάνουμε εισαγωγή μιας νέας λίστας σε μια υπάρχουσα ή αντικατάσταση ενός τμήματος της λίστας με στοιχεία μιας νέας, προσφέρονται οι παρακάτω συναρτήσεις:

```
list llListInsertList( list dest, list src, integer start );
list llListReplaceList( list dest, list src, integer start,
    integer end );
```

Στην πρώτη συνάρτηση, η λίστα του δεύτερου ορίσματος (src) εισάγεται στη λίστα του πρώτου (dest) στη θέση start. Στη δεύτερη συνάρτηση, αφαιρείται το τμήμα της λίστας dest από τις θέσεις start μέχρι end και αντικαθίσταται από τα στοιχεία της src.

Ας δούμε ένα παράδειγμα χρήσης των παραπάνω: Έστω ότι θέλουμε να έχουν συγκεκριμένοι χρήστες πρόσβαση σε ένα διαδραστικό αντικείμενο που κατασκευάζουμε. Στην περίπτωση αυτή θέλουμε να αντιδρά το αντικείμενο διαφορετικά στους χρήστες που έχουν πρόσβαση σε αυτό σε σχέση με τους υπόλοιπους. Για να

κάνουμε έναν τέτοιο έλεγχο μπορούμε απλά να δηλώσουμε στο πρόγραμμά μας τη λίστα των ονομάτων που θέλουμε να έχουν πρόσβαση στο αντικείμενο. Όταν κάποιος χρήστης κάνει κλικ σε αυτό, θα ελέγχουμε αν το όνομά του ανήκει στη λίστα ή όχι και θα τροποποιούμε ανάλογα τη συμπεριφορά του αντικειμένου. Ένα παράδειγμα κώδικα αυτής της μορφής είναι το παρακάτω:

```
list access_list=["Spyros Vos", "John Smith", "Mary Jane"];

default {
    touch_start(integer num) {
        if(llListFindList(access_list,[llDetectedName(0)])>-1) {
            //do something
        }
    }
}
```

Ένα κάπως πιο σύνθετο παράδειγμα θα μπορούσε να είναι το εξής: να έχει το αντικείμενό μας διαφορετική συμπεριφορά στους χρήστες που κάνουν για πρώτη φορά κλικ σε αυτό σε σχέση με όσους κάνουν για δεύτερη ή παραπάνω. Η λύση είναι να αποθηκεύουμε σε λίστα όσους έχουν κάνει κλικ τουλάχιστον μία φορά, ώστε να μπορούμε να γνωρίζουμε σε κάθε νέο κλικ αν ο χρήστης ανήκει στη λίστα αυτήν ή όχι. Ο κώδικας του παραδείγματός μας είναι:

```
list users_clicked;
default {
    touch_start(integer num) {
        string user_name = llDetectedName(0);
        integer found = llListFindList(users_clicked,
            [user_name]);
        if(found<0) {
            users_clicked += [user_name];
            llSay(0, "Hello newcomer");
        }
        else {
            llSay(0, "Hi again, "+user_name);
        }
    }
}
```

Τέλος, μια ακόμη πολύ χρήσιμη δυνατότητα που μας προσφέρεται μέσω της LSL είναι η ανάλυση (parsing) μιας συμβολοσειράς σε επιμέρους τμήματα, τα οποία αποθηκεύονται σε λίστα. Αυτό μπορεί να γίνει μέσω της εντολής:

```
list llParseString2List( string src, list separators,
    list spacers );
```

Η πρώτη παράμετρος (src) είναι η συμβολοσειρά που θα αναλυθεί, η δεύτερη λίστα (separators) περιέχει συμβολοσειρές οι οποίες χωρίζουν τα τμήματα και αγνοούνται, και η τρίτη (spacers), επίσης συμβολοσειρές που χωρίζουν τα τμήματα, οι οποίες όμως διατηρούνται. Το αποτέλεσμα της συνάρτησης είναι μια νέα λίστα η οποία περιέχει τα επιμέρους τμήματα και τις συμβολοσειρές της τρίτης λίστας (spacers) που εντοπίστηκαν. Έστω για παράδειγμα ότι καλούμε την παρακάτω συνάρτηση:

```
list res = llParseString2List("Hello, how are you?", [" "],
    [",", "?"]);
```

Τότε το αποτέλεσμα θα είναι ["Hello", ",", "how", "are", "you", "?"], διότι ζητήσαμε να χρησιμοποιούνται ως διαχωριστές το κενό, το κόμα και το ερωτηματικό, αλλά τα τελευταία δύο να καταγράφονται.

Μπορούμε να εκμεταλλευτούμε την ανάλυση συμβολοσειρών για να κατασκευάζουμε και να χρησιμοποιούμε απλές γλώσσες εντολών τις οποίες καταλαβαίνουν τα αντικείμενά μας. Στο παρακάτω παράδειγμα μπορούμε να στείλουμε στο αντικείμενό μας τις εντολές: pos <x,y,z> και size <x,y,z> σε ένα κρυφό κανάλι και αυτό θα ανταποκριθεί αντίστοιχα. Στην πρώτη περίπτωση θα εκτελέσει σχετική μετατόπιση και στη δεύτερη αλλαγή κλίμακας.

```
default {
    state_entry() {
        llListen(111, "", "", "");
    }

    listen(integer ch, string name, key id, string msg) {
        list commands = llParseString2List(msg,
            [" ", "<", ">", ",", ""], []);
        string cmd = llList2String(commands, 0);
        float x = llList2Float(commands, 1);
        float y = llList2Float(commands, 2);
        float z = llList2Float(commands, 3);

        if(cmd=="pos") {
            llSetPos(llGetPos()+<x, y, z>);
        }
        else if(cmd=="size") {
            llSetScale(<x, y, z>);
        }
    }
}
```

Παρατηρήστε ότι με τους διαχωριστές που ορίσαμε μπορούμε να δώσουμε τις τιμές της θέσης ή της κλίμακας σε μορφή τριών συνεχόμενων αριθμών (π.χ. 1 0.5 0.8) ή σε μορφή vector (π.χ. <1, 0.5, 0.8>). Δοκιμάστε το.

7.2 Αρχεία κειμένου

Η LSL/OSSL μάς δίνει επιπλέον τη δυνατότητα χειρισμού αρχείων κειμένου. Πράγματι, ένα πρόγραμμα μπορεί να ανοίξει αρχεία τύπου notecard που βρίσκονται στα περιεχόμενα του αντικειμένου και να διαβάσει τα περιεχόμενά τους. Επιπλέον, μπορεί να δημιουργήσει νέα τέτοια αρχεία και να αποθηκεύσει δεδομένα σε αυτά. Θα παρουσιάσουμε στη συνέχεια ορισμένες χρήσιμες συναρτήσεις σε OSSL για τον χειρισμό αρχείων και σχετικά παραδείγματα.

Μπορούμε να διαβάσουμε τα περιεχόμενα ενός αρχείου με δύο τρόπους: α) μεταφέροντάς τα όλα σε ένα ενιαίο string και β) διαβάζοντας μία μία τις γραμμές ως ξεχωριστά strings.

Για τον πρώτο τρόπο υπάρχει η παρακάτω συνάρτηση:

```
string osGetNotecard(string name);
```

Στο όρισμα δίνουμε το όνομα του αρχείου, που θα πρέπει να βρίσκεται στα περιεχόμενα του αντικειμένου και μας επιστρέφονται τα δεδομένα του αρχείου σε ένα μοναδικό string. Για να προσθέσουμε ένα notecard στα περιεχόμενα του αντικειμένου μας αρκεί να το δημιουργήσουμε στο αποθετήριο μας, να το σώσουμε και να το μεταφέρουμε (drag) στο αντικείμενο. Για παράδειγμα, θα μπορούσαμε να έχουμε αποθηκευμένο σε αρχείο ένα μήνυμα χαιρετισμού που θα απευθύνει το αντικείμενό μας, έτσι ώστε να μπορούμε να το αλλάζουμε εύκολα χωρίς να χρειαστεί να παρέμβουμε στον κώδικα. Το πρόγραμμα θα είναι το παρακάτω:

```
string msg;

default {
```

```

state_entry() {
    msg = osGetNotecard("message");
}

touch_start(integer num) {
    llSay(0,msg);
}
}

```

Αν από την άλλη μεριά θέλουμε να διαβάζουμε ένα ένα τα περιεχόμενα του notecard, μπορούμε να χρησιμοποιήσουμε τη συνάρτηση:

```
string osGetNotecardLine(string name, integer line);
```

Στο δεύτερο όρισμα δίνουμε τον αριθμό της γραμμής που θέλουμε να διαβάσουμε και μας επιστρέφεται η γραμμή αυτή ως string.

Αν δεν γνωρίζουμε τον αριθμό των γραμμών του notecard μπορούμε να τον βρούμε με την παρακάτω συνάρτηση:

```
integer osGetNumberOfNotecardLines(string name);
```

Στο παρακάτω παράδειγμα γράφουμε σε ένα notecard ονόματα χρηστών και αντίστοιχα μηνύματα που θέλουμε να τους στείλουμε, για παράδειγμα:

```

Spyros Vosinakis
Hello
John Smith
Hi there
Mary Jane
What's up?

```

Μπορούμε στη συνέχεια με ένα πρόγραμμα LSL να διαβάσουμε τα περιεχόμενα του notecard, να τα αποθηκεύσουμε σε λίστα και, ανάλογα με τον χρήστη, να ανταποκριθούμε με αντίστοιχο μήνυμα. Ο κώδικας θα είναι:

```

list users_messages;

default {
    state_entry() {
        integer i;
        for(i=0;i<osGetNumberOfNotecardLines("message");i++) {
            users_messages += [osGetNotecardLine("message",i)];
        }
    }

    touch_start(integer num) {
        string name = llDetectedName(0);
        integer found = llListFindList(users_messages, name);
        if(found<0) {
            llSay(0,"I don't know you!");
        }
        else {
            string msg = llList2String(users_messages,found+1);
            llSay(0,msg);
        }
    }
}

```



```
}  
}
```

Μπορείτε φυσικά σε ένα notecard να γράφετε και αριθμητικά δεδομένα ή διανύσματα και να τα μετατρέπετε από string σε κατάλληλους τύπους δεδομένων μέσω μετατροπής (type casting). Για παράδειγμα, θα μπορούσε ένα notecard να περιέχει ένα σύνολο από θέσεις στις οποίες θα μπορούσε να βρεθεί το αντικείμενο στις διάφορες φάσεις λειτουργίας του ή ακόμα και να περιέχει δεδομένα που θα χρησιμοποιηθούν για συνθετική κίνηση με πλάνα-κλειδιά.

Εκτός από ανάγνωση, η LSL δίνει τη δυνατότητα δημιουργίας και εγγραφής δεδομένων σε notecard. Τα δεδομένα σας μπορούν να είναι αποθηκευμένα είτε σε ένα μοναδικό string είτε σε μία λίστα από strings. Στο παρακάτω παράδειγμα το αντικείμενο αποθηκεύει τα ονόματα των χρηστών που συγκρούστηκαν με αυτό. Αν γίνει κλικ πάνω του, όλα τα αποθηκευμένα ονόματα καταγράφονται σε ένα νέο notecard με όνομα “users”.

```
list users;  
  
default {  
    collision_start(integer num) {  
        string name = llDetectedName(0);  
        integer found = llListFindList(users, [name]);  
        if(found<0) users += [name];  
    }  
  
    touch_start(integer num) {  
        llSay(0, "saving data to notecard 'users'");  
        osMakeNotecard("users", users);  
    }  
}
```

8 Συμπεράσματα

Οι πρόσθετες δυνατότητες και τα παραδείγματα που εξετάσαμε σε αυτό το κεφάλαιο μας οδηγούν στο συμπέρασμα ότι μπορεί κάποιος με την κατάλληλη αξιοποίηση της LSL να αναπτύξει εφαρμογές με πλούσια διαδραστική συμπεριφορά, σε μεγάλο βαθμό ανάλογη των προσχεδιασμένων εικονικών περιβαλλόντων. Το κάθε αντικείμενο του κόσμου μπορεί να έχει τη δική του σχεδιασμένη λειτουργία, να παρατηρεί το περιβάλλον, να εκτελεί προσχεδιασμένες ή δυναμικές συνθετικές κινήσεις, να έχει φυσική συμπεριφορά, να ανταποκρίνεται σε ενέργειες του χρήστη κ.λπ. Επιπλέον, πολλαπλά αντικείμενα μπορούν να επικοινωνούν και να συνεργάζονται μεταξύ τους, νέα αντικείμενα μπορούν να εισάγονται στον κόσμο, να προστίθενται ή και να αφαιρούνται δυναμικά ενσωματώσεις κ.λπ. Οι λειτουργίες αυτές μπορούν να συνδυαστούν με πολλαπλούς και ευρηματικούς τρόπους.

Βεβαίως, οφείλουμε να επισημάνουμε ότι όσα είδαμε ως τώρα είναι μερικές μόνο από τις δυνατότητες της γλώσσας. Αν με αφετηρία τις μέχρι τώρα γνώσεις που αποκτήσατε μελετήσετε τα περιεχόμενα του ιστοτόπου της LSL και των επεκτάσεών της, θα ανακαλύψετε πολλές ακόμη ενδιαφέρουσες λειτουργίες, όπως η επικοινωνία με εξωτερικούς διακομιστές μέσω HTTP, η δημιουργία δυναμικών υφών, ο έλεγχος της κάμερας, η κατασκευή οχημάτων και πολλά άλλα. Ανάλογα με τη μέχρι τώρα εμπειρία σας στον προγραμματισμό και στα τρισδιάστατα περιβάλλοντα, ενδέχεται να χρειαστεί να επενδύσετε αρκετό χρόνο στη μελέτη των δυνατοτήτων της γλώσσας και στη δοκιμή διάφορων λύσεων μέχρι να φτάσετε στο επίπεδο να μπορείτε να αναπτύξετε με ευκολία κάποιο σύνθετο διαδραστικό περιβάλλον. Είναι σκόπιμο σε κάθε περίπτωση να βεβαιωθείτε ότι κατέχετε το υλικό που παρουσιάσαμε μέχρι αυτό το κεφάλαιο, να προσπαθήσετε να λύσετε τις ασκήσεις, να δοκιμάσετε εναλλακτικούς τρόπους αντιμετώπισης και να επεκτείνετε τις λύσεις σας με περαιτέρω λειτουργίες.

Στο επόμενο κεφάλαιο θα εξετάσουμε πώς μπορούμε να επαυξήσουμε τους κατασκευασμένους κόσμους μας με συνθετικούς χαρακτήρες ελεγχόμενους από τον υπολογιστή.

Σύνδεσμοι

Internal Animations: http://wiki.secondlife.com/wiki/Internal_Animations, ενσωματωμένες συνθετικές κινήσεις χαρακτήρων που μπορούν να κληθούν δυναμικά.

LSL Attachment: http://wiki.secondlife.com/wiki/Category:LSL_Attachment, διαθέσιμα σημεία ενσωμάτωσης στο σώμα του χαρακτήρα και στο HUD.

Vehicle Tutorial: http://wiki.secondlife.com/wiki/Linden_Vehicle_Tutorial, κατασκευή οχημάτων με τη χρήση της LSL.

OSSL: http://opensimulator.org/wiki/Category:OSSL_Functions, οι επεκτάσεις της LSL για το OpenSimulator.

Ασκήσεις

1. Κατασκευάστε συσκευή προβολής διαφανειών με πλήκτρα 'προηγούμενο', 'επόμενο' και 'αρχή'.
2. Κατασκευάστε ένα ρομπότ-σκύλο με τηλεχειριστήριο. Θα πρέπει να μπορεί να κινείται στις τέσσερις διευθύνσεις και να γαβγίζει.
3. Κατασκευάστε ένα έκθεμα ενός εικονικού μουσείου / γκαλερί (π.χ. έναν πίνακα ζωγραφικής) και την περιγραφή του (ένα άλλο αντικείμενο στο οποίο η περιγραφή απεικονίζεται ως υφή). Η περιγραφή θα πρέπει αρχικά να είναι κρυφή (π.χ. αόρατη ή πολύ μικρού μεγέθους). Όταν κάποιος κάνει κλικ στον πίνακα, θα πρέπει να εμφανίζεται η περιγραφή στη σωστή θέση και με το σωστό μέγεθος. Αν γίνει κλικ ξανά στο έκθεμα, θα πρέπει η περιγραφή να εξαφανιστεί.
4. Κατασκευάστε σύστημα πλανήτη και δορυφόρου που περιστρέφεται γύρω από αυτόν χρησιμοποιώντας ομάδα αντικειμένων και την `IITargetOmega`. Χρησιμοποιήστε εξωτερικά πλήκτρα για να αλλάξετε την ταχύτητα ή και τον άξονα της περιστροφής. Συμβουλή: αν αποθηκεύσετε τον άξονα περιστροφής σε μεταβλητή διανύσματος, τότε μπορείτε να τον περιστρέψετε πολλαπλασιάζοντας κάποια περιστροφή με αυτόν.
5. Κατασκευάστε ένα HUD με τρία πλήκτρα, τα οποία ενεργοποιούν διαφορετικές συνθετικές κινήσεις της ενσάρκωσής σας, για παράδειγμα `bow`, `clap`, `dance1`.
6. Κάντε κάθε μπάλα του συντριβανιού του κώδικα 11.4 να έχει το δικό της χρώμα. Θα πρέπει η μπάλα να περιλαμβάνει και αυτή script και να αλλάζει το χρώμα όταν ενεργοποιείται το συμβάν `onRez`. Για τον ορισμό τυχαίου χρώματος θα χρησιμοποιήσετε την `IFFrand`.
7. Κατασκευάστε έναν ελεγκτή για το κανόνι σας (περιστροφή πάνω και κάτω και εκτόξευση) και έναν στόχο που αλλάζει προσωρινά χρώμα αν χτυπηθεί από το βλήμα. Χρησιμοποιήστε το συμβάν `collision_start` για να εντοπίσετε τη σύγκρουση.
8. Κατασκευάστε ένα αντικείμενο του οποίου το μέγεθος και η θέση μπορούν να αλλάξουν μέσω παράθυρων διαλόγου. Αρχικά θα ρωτάει τον χρήστη ποια από τις δύο ιδιότητες θέλει να αλλάξει και στη συνέχεια, ανάλογα με την απάντηση, θα προσφέρει επιλογές τροποποίησης (πανω/κάτω και μικρότερο/μεγαλύτερο αντίστοιχα).
9. Κατασκευάστε ένα αντικείμενο το οποίο θα χαιρετάει μία μόνο φορά κάθε νέο χρήστη που θα το πλησιάζει σε ακτίνα 20 μέτρων. Θα χρειαστεί να χρησιμοποιήσετε επαναλαμβανόμενο sensor και λίστα για να αποθηκεύετε τα ονόματα όσων έχει ήδη χαιρετήσει.
10. Κατασκευάστε ένα αντικείμενο το οποίο θα καταγράφει τις δημόσιες συνομιλίες (chat) των χρηστών. Θα μπορεί ο χρήστης να επιλέξει μέσω διαλόγου μία από τις εξής επιλογές: Έναρξη, Τερματισμός και Καταγραφή. Με την πρώτη επιλογή το αντικείμενο θα καταγράφει οτιδήποτε ακούει στο δημόσιο κανάλι, με τη δεύτερη θα σταματάει την καταγραφή και με την τρίτη θα αποθηκεύει τα

καταγεγραμμένα δεδομένα σε notecard. Μπορείτε να χρησιμοποιήσετε είτε ένα μεγάλο string είτε λίστα για να αποθηκεύσετε τα μηνύματα.

11. Κατασκευάστε ένα αντικείμενο για quiz ερωταπαντήσεων. Θα διαβάζει από ένα notecard μία ερώτηση (πρώτη γραμμή), τρεις πιθανές απαντήσεις (δεύτερη με τέταρτη γραμμή) και τη σωστή απάντηση (πέμπτη γραμμή). Με βάση αυτά, θα δημιουργεί διάλογο στον χρήστη και, ανάλογα με την ορθότητα της απάντησης, θα του δίνει ανατροφοδότηση. Για ανατροφοδότηση θα εμφανίζεται προσωρινά ένα αντικείμενο (π.χ. ένα ταμπελάκι «Σωστά!» ή «Λάθος, ξαναπροσπαθήστε») και μετά από λίγα δευτερόλεπτα θα εξαφανίζεται πάλι.
12. Επεκτείνετε την άσκηση 11 ώστε να δέχεται μόνο μία απάντηση ανά χρήστη και να εμφανίζει μετά από κάθε απάντηση τα ονόματα των χρηστών που απάντησαν σωστά.
13. Επεκτείνετε την άσκηση 12 ώστε να περιλαμβάνει πολλαπλές ερωτήσεις και κρατήστε σκορ για κάθε χρήστη. Συμβουλή: μπορείτε να χρησιμοποιήσετε μια λίστα με τα ονόματα των χρηστών ακολουθούμενα από το σκορ τους και κάθε φορά που θα δίνουν μια νέα σωστή απάντηση να αντικαθιστάτε το σκορ τους από το καινούριο (βλ. `l1ListReplaceList`).

Κεφάλαιο 11: Συνθετικοί Χαρακτήρες

Σύνοψη

Αντικείμενο του παρόντος κεφαλαίου είναι η δημιουργία και ο χειρισμός συνθετικών χαρακτήρων (ή NPCs, όπως ονομάζονται στο OpenSimulator) μέσω της γλώσσας LSL/OSSL. Παρουσιάζεται με παραδείγματα η εισαγωγή ενός NPC στον κόσμο και οι βασικές ενέργειες που μπορεί να εκτελέσει στο περιβάλλον: ομιλία, εκτέλεση προκαθορισμένων κινήσεων (animations), βάδισμα ή πέταγμα προς κάποια κατεύθυνση, αλλαγή προσανατολισμού, κάθισμα, αλληλεπίδραση με αντικείμενα κ.ά. Στη συνέχεια παρουσιάζονται χρήσιμες τεχνικές και παραδείγματα κώδικα για την αξιοποίηση των NPCs σε σύνθετα σενάρια, όπως η κίνηση σε προκαθορισμένο μονοπάτι, η παρακολούθηση ενός άλλου κινούμενου χαρακτήρα ή αντικειμένου κ.ά. Τέλος, παρουσιάζονται παραδείγματα και σενάρια συντονισμένης ή ελεύθερης κίνησης πλήθους.

Προαπαιτούμενη γνώση

Απαιτείται καλή γνώση της γλώσσας LSL/OSSL και των δυνατοτήτων της, όπως παρουσιάζεται στα Κεφάλαια 9 και 10.

1 Εισαγωγή

Η ενότητα αυτή έχει ως αντικείμενο τους συνθετικούς χαρακτήρες και τις δυνατότητές τους στο περιβάλλον του OpenSimulator. Οι συνθετικοί χαρακτήρες του OpenSimulator ονομάζονται NPCs (από το Non Player Characters, όρο που χρησιμοποιείται στα παιχνίδια) και είναι ενσάρκώσεις ελεγχόμενες από τον υπολογιστή, οι οποίες μπορούν να εκτελέσουν διάφορες δραστηριότητες στον εικονικό κόσμο, όπως το να βαδίζουν, να αναπαράγουν συνθετικές κινήσεις, να μιλάνε με μηνύματα κειμένου κ.λπ. Οι συνθετικοί χαρακτήρες δημιουργούνται μέσω εντολών της γλώσσας OSSL, δηλαδή των επεκτάσεων της LSL για το OpenSimulator. Για να έχετε τη δυνατότητα να τους χρησιμοποιήσετε και να τρέξετε τα παραδείγματα του βιβλίου θα πρέπει να είναι ενεργοποιημένες στον διακομιστή του κόσμου οι εντολές της OSSL και οι NPCs (βλ. Κεφ. 7).

Αρχικά θα παρουσιάσουμε την κατασκευή και τις βασικές προσφερόμενες λειτουργίες των συνθετικών χαρακτήρων. Στη συνέχεια θα εξετάσουμε σύνθετες χωρικές συμπεριφορές, θα δημιουργήσουμε εικονικά πλήθη με μοντέλο κίνησης οδηγού-ακολουθών (leader-follower) και θα παρουσιάσουμε ορισμένες τεχνικές για την παραγωγή περίπλοκων συμπεριφορών με τη χρήση μηχανών καταστάσεων.

2 Κατασκευή και βασικές λειτουργίες

Οι συνθετικοί χαρακτήρες δεν ανήκουν στην ίδια κατηγορία με τα υπόλοιπα αντικείμενα που κατασκευάζουμε και διαχειριζόμαστε στον κόσμο. Δημιουργούνται ως ενσάρκώσεις, «αντιγράφοντας» τη μορφή μιας υπάρχουσας ενσάρκωσης χρήστη. Ως εκ τούτου, δεν μπορούμε να τους επεξεργαστούμε όπως επεξεργαζόμαστε τα αντικείμενα του κόσμου. Δεν έχουμε τη δυνατότητα να τους μετακινήσουμε, να τροποποιήσουμε την εμφάνισή τους ή να τους προσθέσουμε κάποιο πρόγραμμα (script) προς εκτέλεση. Το μόνο που μπορούμε να κάνουμε είναι να τους δημιουργούμε και να τους βάζουμε να εκτελούν ενέργειες στο περιβάλλον μέσω προγράμματος που εκτελείται σε κάποιο άλλο αντικείμενο του κόσμου. Ας δούμε πώς:

Θα ξεκινήσουμε δημιουργώντας τον πρώτο μας συνθετικό χαρακτήρα. Η εντολή για να το πετύχουμε αυτό είναι η παρακάτω:

```
key osNpcCreate(string firstname, string lastname,  
vector position, string clonefrom);
```

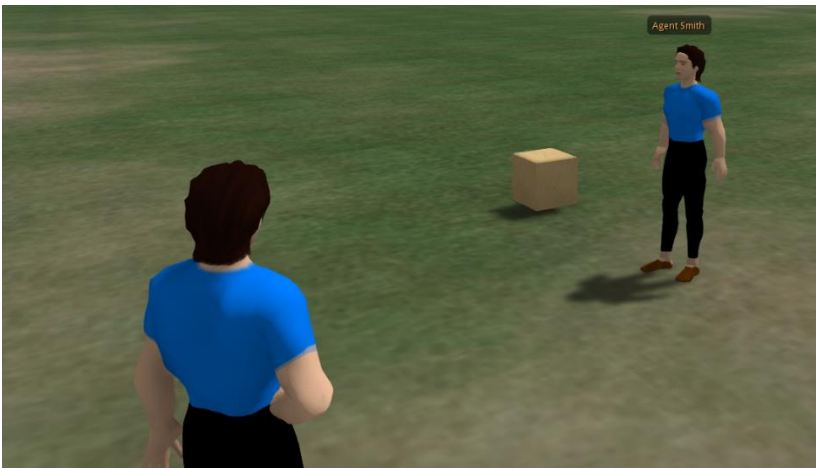
Η εντολή αυτή δημιουργεί έναν χαρακτήρα και μας επιστρέφει το κλειδί του. Τα δύο πρώτα ορίσματα περιέχουν το μικρό όνομα και το επώνυμο του νέου χαρακτήρα μας αντίστοιχα. Το τρίτο είναι η θέση στην οποία θα τοποθετηθεί σε συντεταγμένες κόσμου και το τέταρτο η μορφή την οποία θα έχει. Σε ό,τι αφορά τη μορφή υπάρχουν δύο πιθανές προσεγγίσεις:

1. Να χρησιμοποιήσουμε το κλειδί ενός υπάρχοντα χρήστη. Στην περίπτωση αυτήν ο χαρακτήρας θα αντιγράψει απευθείας την εμφάνιση του χρήστη αυτού.
2. Να σώσουμε την εμφάνιση ενός χρήστη σε αρχείο τύπου notecard και να χρησιμοποιήσουμε το όνομα του αρχείου αυτού για να ορίσουμε την εμφάνιση του χαρακτήρα.

Ας δούμε αρχικά ένα παράδειγμα της πρώτης προσέγγισης: Δημιουργήστε ένα αντικείμενο και προσθέστε τον παρακάτω κώδικα:

```
default {
    touch_start(integer num) {
        osNpcCreate("Agent", "Smith", llGetPos()+<2.0,0.0,0.0>,
            llDetectedKey(0));
    }
}
```

Αν κάνετε κλικ στο αντικείμενο, θα δείτε να δημιουργείται ένα αντίγραφό σας με το όνομα “Agent Smith” και να τοποθετείται δίπλα στο αντικείμενο (εικ. 11.1).



Εικόνα 11.1 Δημιουργία ενός ψηφιακού χαρακτήρα (NPC).

Η εντολή για να διαγράψουμε έναν χαρακτήρα από τον κόσμο είναι:

```
osNpcRemove(key npc);
```

Όπως μπορείτε να παρατηρήσετε, είναι απαραίτητο να γνωρίζουμε το κλειδί του χαρακτήρα για να εκτελέσουμε την ενέργεια αυτήν. Μπορείτε να βρείτε το κλειδί ενός χαρακτήρα αν κάνετε δεξί κλικ πάνω του και επιλέξετε ‘Profile’. Επομένως η εντολή που θα χρησιμοποιήσουμε θα είναι αυτής της μορφής:

```
osNpcRemove("35168bd2-70ae-11e5-9d70-feff819cdc9f");
```

Για να κάνουμε τα πράγματα ευκολότερα, είναι προτιμότερο να αποθηκεύουμε το κλειδί του χαρακτήρα που δημιουργούμε σε μια σφαιρική μεταβλητή και να αναφερόμαστε σε αυτό όταν θέλουμε να τον διαγράψουμε. Άλλωστε, όλες οι συναρτήσεις που κάνουν τους χαρακτήρες να εκτελούν ενέργειες χρειάζεται να γνωρίζουν το κλειδί τους, όπως θα δούμε στη συνέχεια. Συνεπώς, μια βελτιωμένη έκδοση του παραπάνω κώδικα είναι η εξής:

```
key npc;
integer alive = FALSE;

default {
    touch_start(integer num) {
```

```

    if(alive==FALSE) {
        npc = osNpcCreate("Agent", "Smith", llGetPos()+
            <2.0,0.0,0.0>, llDetectedKey(0));
        alive = TRUE;
    }
    else {
        osNpcRemove(npc);
        alive = FALSE;
    }
}
}

```

Στη νέα αυτή έκδοση, αν κάνουμε κλικ στο αντικείμενο, δημιουργείται η ενσάρκωση, και, αν κάνουμε κλικ ξανά, διαγράφεται.

Ο δεύτερος τρόπος για να ορίσουμε την εμφάνιση του χαρακτήρα είναι μέσω αρχείου τύπου notecard. Σε ορισμένες περιπτώσεις η προσέγγιση αυτή είναι χρησιμότερη, γιατί μας επιτρέπει να έχουμε πολλαπλές εμφανίσεις αποθηκευμένες σε ένα αντικείμενο και να επιλέγουμε ποια θέλουμε όταν κατασκευάζουμε έναν χαρακτήρα. Αν δημιουργήσετε ένα νέο αντικείμενο, γράψετε τον παρακάτω κώδικα και τον εκτελέσετε, θα παρατηρήσετε ότι προστέθηκε ένα νέο αρχείο στα περιεχόμενα του αντικειμένου.

```

default {
    state_entry() {
        osOwnerSaveAppearance("app1");
    }
}

```

Το αρχείο είναι τύπου notecard, ονομάζεται 'app1' και έχει αποθηκευμένη την τρέχουσα εμφάνισή σας. Μπορείτε να μεταφέρετε το αρχείο αυτό στο αποθετήριό σας και να το χρησιμοποιείτε για να δημιουργείτε χαρακτήρες με την εμφάνιση αυτήν. Η εντολή για να κατασκευαστεί χαρακτήρας με την εμφάνιση 'app1' θα είναι στην περίπτωση αυτή:

```

npc = osNpcCreate("Agent", "Smith", llGetPos() +
    <2.0, 0.0, 0.0>, "app1");

```

Απαραίτητη προϋπόθεση για τη σωστή εκτέλεσή της είναι προφανώς το να υπάρχει το αρχείο 'app1' στα περιεχόμενα του αντικειμένου που θα τρέξει το πρόγραμμα.

Προσφέρονται δύο εντολές για την αποθήκευση της εμφάνισης ενός χρήστη σε αρχείο. Η πρώτη είναι για τον ιδιοκτήτη του αντικειμένου και η δεύτερη για οποιονδήποτε χρήστη:

```

key osOwnerSaveAppearance(string notecard);
key osAgentSaveAppearance(key agentId, string notecard);

```

Επομένως, αν θέλετε να αποθηκεύσετε σε αρχείο την εμφάνιση του κάθε χρήστη που κάνει κλικ στο αντικείμενο και να δώσετε στο αρχείο το όνομα του χρήστη, ο κώδικας θα είναι ο παρακάτω:

```

default {
    touch_start(integer num) {
        osAgentSaveAppearance(llDetectedKey(0),
            llDetectedName(0));
    }
}

```

Στη συνέχεια θα παρουσιάσουμε ορισμένες από τις βασικές δυνατότητες επενέργειας στο περιβάλλον των συνθετικών χαρακτήρων.

Μπορούμε να μάθουμε την τρέχουσα θέση και προσανατολισμό ενός χαρακτήρα με τις παρακάτω συναρτήσεις:

```
vector osNpcGetPos(key npc);  
rotation osNpcGetRot(key npc);
```

Και στις δύο συναρτήσεις περνάμε ως όρισμα το κλειδί του χαρακτήρα και μας επιστρέφονται η θέση και ο προσανατολισμός αντίστοιχα.

Για να αλλάξουμε τον προσανατολισμό ενός χαρακτήρα μπορούμε να χρησιμοποιήσουμε την παρακάτω συνάρτηση:

```
osNpcSetRot(key npc, rotation rot);
```

Στο όρισμα rot θα δώσουμε την τελική περιστροφή που θέλουμε να έχει ο χαρακτήρας.

Ο χαρακτήρας μας μπορεί να μετακινηθεί προς κάποια θέση-στόχο με την παρακάτω συνάρτηση:

```
osNpcMoveToTarget(key npc, vector target, integer options);
```

Το δεύτερο όρισμα είναι η θέση στην οποία θέλουμε να βρεθεί σε συντεταγμένες κόσμου. Στο τρίτο όρισμα δηλώνουμε παραμέτρους σχετικά με τον τύπο της κίνησης που θέλουμε να εκτελεί. Προς το παρόν οι διαθέσιμες τιμές είναι:

- OS_NPC_FLY: ο χαρακτήρας «πετάει» προς τη θέση-στόχο,
- OS_NPC_NO_FLY: ο χαρακτήρας προσπαθεί να βαδίζει προς τον στόχο,
- OS_NPC_LAND_AT_TARGET: αν ο χαρακτήρας πετάει, τότε υποχρεωτικά θα «προσγειωθεί» στη θέση-στόχο,
- OS_NPC_RUNNING: ο χαρακτήρας θα μετακινηθεί τρέχοντας.

Οι παραπάνω τιμές μπορούν να συνδυαστούν με λογικό ή (!), π.χ. αν θέλουμε ο χαρακτήρας μας να πετάξει αλλά και να προσγειωθεί στον στόχο, η τιμή θα είναι: OS_NPC_FLY | OS_NPC_LAND_AT_TARGET.

Με βάση τα παραπάνω, αν για παράδειγμα θέλουμε ο χαρακτήρας μας να βαδίζει δέκα μέτρα στη διεύθυνση του άξονα X, η εντολή θα είναι:

```
osNpcMoveToTarget(npc, osNpcGetPos()+<10.0, 0.0, 0.0>,  
OS_NPC_NO_FLY);
```

Αν για κάποιον λόγο θέλουμε να σταματήσει η μετακίνηση του χαρακτήρα πριν φτάσει στον στόχο του, η εντολή είναι:

```
osNpcStopMoveToTarget(key npc);
```

Ο χαρακτήρας μας μπορεί ακόμα να καθίσει σε κάποιο αντικείμενο, δηλαδή να εκτελέσει την ενέργεια 'sit' όπως ακριβώς θα έκανε και η ενσάρκωση ενός χρήστη. Η εντολή είναι:

```
osNpcSit(key npc, key target, integer options);
```

Η δεύτερη παράμετρος είναι το κλειδί του αντικειμένου στο οποίο θα καθίσει ο χαρακτήρας και η τρίτη οι παράμετροι σχετικά με τον τρόπο που θα καθίσει ο χαρακτήρας. Προς το παρόν, υποστηρίζεται μόνο η τιμή OS_NPC_SIT_NOW.

Αν θέλουμε να σηκωθεί πάνω ένας χαρακτήρας που είναι ήδη σε καθιστή στάση, η εντολή είναι:

```
osNpcStand(key npc);
```

Οι χαρακτήρες μας έχουν ακόμη τη δυνατότητα να στείλουν μηνύματα στο δημόσιο ή σε κρυφά κανάλια επικοινωνίας. Για να μιλήσει ένας χαρακτήρας, μπορούμε να χρησιμοποιήσουμε τις παρακάτω συναρτήσεις:


```

osNpcSay(key npc, string message);
osNpcSay(key npc, int channel, string message);
osNpcShout(key npc, int channel, string message);
osNpcWhisper(key npc, int channel, string message);

```

Με την κλήση της πρώτης συνάρτησης, ο χαρακτήρας μας λέει το μήνυμα message στο δημόσιο κανάλι επικοινωνίας. Οι υπόλοιπες τρεις χρησιμοποιούνται για να στείλει μήνυμα στο κανάλι που θα καθορίσουμε στο όρισμα channel μιλώντας, φωνάζοντας ή ψιθυρίζοντας αντίστοιχα.

Ο χαρακτήρας μας μπορεί ακόμη να κάνει κλικ σε κάποιο αντικείμενο. Εφόσον στο αντικείμενο αυτό τρέχει κάποιο script το οποίο ανταποκρίνεται στο κλικ, δηλαδή περιλαμβάνει διαχειριστή για ένα ή περισσότερα από τα συμβάντα touch, touch_start και touch_end, το αντικείμενο θα αντιδράσει σαν να είχε κάνει κλικ πάνω του κάποιος χρήστης. Η εντολή είναι η παρακάτω:

```

osNpcTouch(key npcKey, key objectKey, integer linkNum);

```

Το δεύτερο όρισμα είναι το κλειδί του αντικειμένου στο οποίο θα κάνει κλικ και στο τρίτο προσδιορίζουμε αν θα γίνει κλικ στο ίδιο το αντικείμενο ή στη ρίζα της ομάδας στην οποία ανήκει. Αν η τιμή του τρίτου ορίσματος (linkNum) είναι LINK_THIS, γίνεται κλικ στο ίδιο το αντικείμενο, ενώ αν ανήκει σε ομάδα και θέλουμε να γίνει κλικ στο αντικείμενο-ρίζα της ομάδας, η τιμή θα πρέπει να γίνει LINK_ROOT.

Τέλος, μπορούμε να κάνουμε τον χαρακτήρα μας να εκτελέσει μια συνθετική κίνηση ή να διακόψει την εκτέλεσή της. Οι εντολές είναι οι παρακάτω:

```

osNpcPlayAnimation(key npc, string animation);
osNpcStopAnimation(key npc, string animation);

```

Το δεύτερο όρισμα είναι το όνομα ενός αρχείου animation που θα πρέπει να υπάρχει στα περιεχόμενα του αντικειμένου ή κάποιο από τα έτοιμα animations του Second Life (βλ. Κεφ. 10).

Από τα παραπάνω μπορούμε να παρατηρήσουμε ότι οι συνθετικοί χαρακτήρες μπορούν μέσω των κατάλληλων συναρτήσεων να κάνουν περίπου ό,τι και οι χρήστες του κόσμου. Μετακινούνται, συνομιλούν, κάθονται σε αντικείμενα, κάνουν κλικ σε αντικείμενα και εκτελούν συνθετικές κινήσεις. Έχει λοιπόν κάποιος τη δυνατότητα να συνδυάσει μερικές από τις παραπάνω εντολές και να δημιουργήσει μια μικρή «ιστορία» με συνθετικούς χαρακτήρες. Να ένα παράδειγμα:

```

key npc;
//notecard με όνομα app1 θα πρέπει να βρίσκεται στα περιεχόμενα
default {
    touch_start(integer num) {
        npc = osNpcCreate("Agent", "Smith", llGetPos()+
            <2.0,0.0,0.0>, "app1");
        llSleep(3.0);
        osNpcSay(npc,"Hello World!!");
        osNpcMoveToTarget(npc, llGetPos()+<12.0,0.0,0.0>,
            OS_NPC_NO_FLY);
        llSleep(5.0);
        osNpcMoveToTarget(npc, llGetPos()+<2.0,0.0,0.0>,
            OS_NPC_NO_FLY);
        llSleep(5.0);
        osNpcPlayAnimation(npc, "bow");
        llSleep(3.0);
        osNpcSay(npc, "Goodbye for now!");
        llSleep(2.0);
        osNpcRemove(npc);
    }
}

```

Στο παράδειγμά μας χρησιμοποιήσαμε ορισμένες από τις εντολές ενεργειών των χαρακτήρων που παρουσιάσαμε παραπάνω και την `llSleep` για να υπάρχει χρονική καθυστέρηση μεταξύ της εκτέλεσης της καθημίας εντολής. Όταν κάνουμε κλικ στο αντικείμενο, δημιουργείται ο χαρακτήρας, λέει “Hello World”, κινείται δέκα μέτρα προς τα μπρος, γυρίζει πίσω στην αρχική του θέση, κάνει μια υπόκλιση, λέει “Goodbye” και εξαφανίζεται.

Προφανώς μπορείτε να δημιουργήσετε και πιο σύνθετα και ενδιαφέροντα παραδείγματα από το παραπάνω. Όμως, αν η διάρκεια εκτέλεσης των εντολών σας υπερβαίνει το μισό λεπτό, η εκτέλεση θα διακοπεί, λόγω περιορισμών της LSL. Θυμηθείτε ότι η LSL βασίζεται στο μοντέλο προγραμματισμού συμβάντων. Θα πρέπει λοιπόν να μοιράσετε τη συμπεριφορά του χαρακτήρα σας σε επιμέρους τμήματα και να χρησιμοποιείτε χρονικά συμβάντα (`llSetTimerEvent`) αντί για την `llSleep`.

3 Χωρική συμπεριφορά

Στην παρούσα ενότητα θα παρουσιάσουμε ορισμένες πιο σύνθετες συμπεριφορές των χαρακτήρων που σχετίζονται με την κίνησή τους στον χώρο.

Αρχικά θα δούμε πώς μπορούμε να κάνουμε τον χαρακτήρα μας να κινείται σε ένα προκαθορισμένο μονοπάτι. Για να το πετύχουμε αυτό αρκεί να έχουμε καταχωρημένα όλα τα σημεία του μονοπατιού σε μια λίστα, να ελέγχουμε κατά πόσο ο χαρακτήρας έχει φτάσει σε κάποιον στόχο και να ορίζουμε το επόμενο σημείο της λίστας ως νέο στόχο.

Στον κώδικα ο οποίος παρουσιάζεται παρακάτω χρησιμοποιούμε έναν ακέραιο αριθμό ως δείκτη (`index`) για να αποθηκεύσουμε την τρέχουσα θέση-στόχο στη λίστα. Χρησιμοποιώντας έναν `timer` ελέγχουμε επαναλαμβανόμενα τη θέση του χαρακτήρα μας ώστε να διαπιστώσουμε αν έχει φτάσει στη θέση-στόχο. Αν αυτό συμβεί, προχωράμε τον δείκτη κατά ένα και θέτουμε την επόμενη θέση ως στόχο. Αν δεν υπάρχει επόμενη θέση, επιστρέφουμε στη θέση μηδέν και η διαδικασία ξεκινάει από την αρχή.

Ο κώδικάς μας είναι ο παρακάτω:

```
key npc;
integer alive = FALSE;
list path = [<87.0,76.0,21.0>, <97.0,76.0,21.0>,
            <97.0, 66.0, 21.0>, <87.0, 66.0, 21.0>];
vector target;
integer index;

moveToNextTarget () {
    index++;
    if(index==llGetListLength(path))
        index = 0;
    target = llList2Vector(path,index);
    osNpcMoveToTarget(npc, target, OS_NPC_NO_FLY);
}

default {
    touch_start(integer num) {
        if(alive==FALSE) {
            npc = osNpcCreate("Agent", "Smith", llGetPos()+
                <2.0,0.0,0.0>, llDetectedKey(0));
            alive = TRUE;
            index = -1;
            moveToNextTarget();
            llSetTimerEvent(2.0);
        }
        else {
            osNpcRemove(npc);
            alive = FALSE;
        }
    }
}
```

```

        llSetTimerEvent(0);
    }
}

timer() {
    vector npcPos = osNpcGetPos(npc);
    vector diff = target-npcPos;
    if(llVecMag(diff)<2.0) {
        moveToNextTarget();
    }
}
}

```

Αν εκτελέσουμε τον κώδικα και κάνουμε κλικ στο αντικείμενο, θα δημιουργηθεί ένας νέος χαρακτήρας ο οποίος θα ακολουθεί διαρκώς το μονοπάτι που περιγράφεται στη λίστα path. Αν κάνουμε ξανά κλικ στο αντικείμενο, ο χαρακτήρας εξαφανίζεται.

Μια δυσκολία με την παραπάνω λύση είναι ότι ο υπολογισμός των επιμέρους θέσεων του μονοπατιού, δηλαδή οι τιμές της λίστας path, μπορεί να γίνει επίπονος. Μια απλή λύση για να ξεπεράσουμε τη δυσκολία αυτήν είναι να τοποθετήσουμε μικρά αντικείμενα στις θέσεις που θέλουμε να βρεθεί ο χαρακτήρας μας και να αντιγράψουμε από αυτά τις θέσεις των κέντρων τους. Μια σειρά βημάτων που θα μπορούσατε να ακολουθήσετε είναι η παρακάτω:

1. Κατασκευάστε ένα μικρό και εύκολο στον εντοπισμό αντικείμενο με όνομα “marker”. Προσθέστε του script το οποίο εμφανίζει στο δημόσιο κανάλι συνομιλίας τη θέση του όταν γίνεται κλικ πάνω του.
2. Αντιγράψτε το αντικείμενο αυτό στο αποθετήριό σας.
3. Αλλάξτε την οπτική σας σε κάτωψη του εδάφους (με τη χρήση των πλήκτρων Alt και Ctrl – βλ. Κεφ. 7) και προσθέστε σημάδια στις θέσεις του μονοπατιού μεταφέροντας το αντικείμενο marker από το αποθετήριο στα σημεία που επιθυμείτε.
4. Όταν ολοκληρώσετε τον καθορισμό των σημαδιών, κάντε κλικ σε καθένα από αυτά με τη σειρά που θέλετε να τα επισκεφτεί ο χαρακτήρας. Το κάθε αντικείμενο θα στείλει τις δικές του συντεταγμένες και εσείς μπορείτε να τις αντιγράψετε από το ιστορικό συνομιλίας (πλήκτρο ‘Local Chat’).

Η διαδικασία απεικονίζεται στην εικόνα 11.2.



Εικόνα 11.2 Δημιουργία σημαδιών (markers) για τον προσδιορισμό μονοπατιού που θα βαδίζει ο χαρακτήρας.

Μια εναλλακτική προσέγγιση είναι να καταγράφουμε τις κινήσεις ενός πραγματικού χρήστη σε μια λίστα και να τις αποθηκεύουμε στη συνέχεια σε ένα αρχείο τύπου notecard. Ο κώδικας στην περίπτωση αυτή θα είναι ο εξής:

```
list positions;
integer recording = FALSE;

default {
    touch_start(integer num) {
        if(recording==FALSE) {
            llSay(0,"Recording path of "+llDetectedName(0));
            llSensorRepeat("",llDetectedKey(0),AGENT,96.0,PI,2.0);
            recording = TRUE;
        }
        else {
            llSensorRemove();
            recording = FALSE;
            string file = llDetectedName(0)+"-path";
            llSay(0,"Recording stopped. Creating file '"+file+"'");
            osMakeNotecard(llDetectedName(0)+"-path",positions);
            positions = [];
        }
    }

    sensor(integer num) {
        positions += llDetectedPos(0);
    }
}
```

Όταν ένας χρήστης κάνει κλικ πάνω στο αντικείμενο, αυτό καταγράφει τη θέση του στον κόσμο ανά δύο δευτερόλεπτα και το αποθηκεύει στη λίστα positions. Όταν γίνεται κλικ ξανά στο αντικείμενο, η καταγραφή διακόπτεται και οι αποθηκευμένες τιμές καταχωρούνται σε ένα νέο αρχείο τύπου notecard στα περιεχόμενα του αντικειμένου. Αφού εκτελέσετε το πρόγραμμα και ολοκληρώσετε την καταγραφή, μπορείτε να αξιοποιήσετε τις τιμές του notecard για να ορίσετε το μονοπάτι που θα ακολουθήσει ο συνθετικός χαρακτήρας.

Η παραπάνω διαδικασία παρουσιάζεται στο βίντεο 11.1.



Σύνδεσμος:

<https://youtu.be/VJhZdoAj5U0>

Video 11.1 Εισαγωγή «σημαδιών» για τον ορισμό μονοπατιού βαδίσματος του χαρακτήρα.

Στο παράδειγμα που θα παρουσιάσουμε στη συνέχεια θα κάνουμε τον χαρακτήρα μας να ακολουθεί έναν χρήστη, για παράδειγμα τον ιδιοκτήτη του αντικειμένου. Αντί για timer μπορούμε να χρησιμοποιήσουμε έναν επαναλαμβανόμενο sensor ο οποίος θα παρακολουθεί τη θέση του χρήστη. Στο παρακάτω πρόγραμμα ο χαρακτήρας ακολουθεί τον χρήστη 'Test User' προσπαθώντας να κρατήσει μια ελάχιστη απόσταση δύο μέτρων από αυτόν. Ο κώδικας είναι ο παρακάτω:

```
key npc;
integer alive = FALSE;

default {
    touch_start(integer num) {
```

```

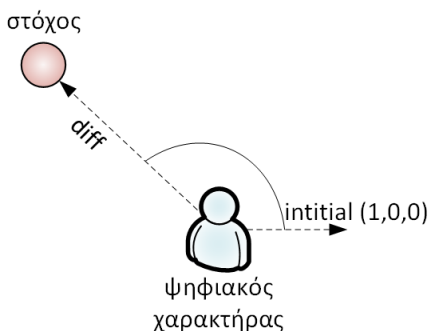
if(alive==FALSE) {
    npc = osNpcCreate("Agent", "Smith", llGetPos()+
        <2.0,0.0,0.0>, llDetectedKey(0));
    alive = TRUE;
    llSensorRepeat("Test User","", AGENT, 96.0, PI, 2.0);
}
else {
    osNpcRemove(npc);
    alive = FALSE;
    llSensorRemove();
}
}

sensor(integer num) {
    vector userPos = llDetectedPos(0);
    vector npcPos = osNpcGetPos(npc);
    vector diff = userPos - npcPos;
    if(llVecMag(diff) > 3.0) {
        vector diffNorm = llVecNorm(diff);
        vector targetPos = <userPos.x-diffNorm.x*2.0,
            userPos.y-diffNorm.y*2.0, userPos.z>;
        osNpcMoveToTarget(npc, targetPos, OS_NPC_NO_FLY);
    }
}
}
}

```

Όπως μπορείτε να παρατηρήσετε στον κώδικα, ως θέση-στόχος δεν ορίζεται η τρέχουσα θέση του χρήστη, αλλά ένα σημείο δύο μέτρα πίσω από αυτόν στον άξονα που ενώνει τον χαρακτήρα με τον χρήστη. Η λογική είναι παραπλήσια με αυτήν που παρουσιάσαμε στο Κεφάλαιο 9. Αν χρησιμοποιούσαμε απλά τη θέση του χρήστη ως στόχο, θα υπήρχε κίνδυνος να συγκρούεται συχνά ο χαρακτήρας με τον χρήστη, κάτι που θα ήταν προτιμότερο να αποφεύγεται. Μπορούμε επίσης να παρατηρήσουμε στον κώδικα ότι ο χαρακτήρας ξεκινάει να βαδίζει μόνο όταν η απόστασή του από τον χρήστη είναι μεγαλύτερη των τριών μέτρων. Αυτό μας επιτρέπει να αποφεύγουμε άσκοπες μικρομετακινήσεις του χαρακτήρα.

Ας δούμε στη συνέχεια ένα παράδειγμα στο οποίο κάνουμε τον χαρακτήρα μας να κοιτάζει πάντα προς το σημείο στο οποίο βρίσκεται ένας συγκεκριμένος χρήστης. Μια τέτοια συμπεριφορά είναι, προφανώς, χρήσιμη σε περιπτώσεις που θέλουμε ο χαρακτήρας μας να απευθύνεται σε κάποιον χρήστη, για παράδειγμα να του λέει κάτι και να έχει σωστό προσανατολισμό.



Εικόνα 11.3 Υπολογισμός κατάλληλης περιστροφής του χαρακτήρα ώστε να έχει διεύθυνση προς κάποιον στόχο.

Πριν παρουσιάσουμε τον σχετικό κώδικα, θα πρέπει να κατανοήσουμε τη λογική με την οποία θα γίνει αυτή η περιστροφή. Αρχικά, θα πρέπει να γνωρίζουμε ότι το μόνο είδος περιστροφής το οποίο έχει νόημα για το σώμα ενός ψηφιακού χαρακτήρα είναι γύρω από τον άξονα Z. Η περιστροφή σε οποιονδήποτε άλλον άξονα θα δημιουργούσε «αφύσικα» αποτελέσματα και ούτως ή άλλως δεν λαμβάνεται υπόψη από την `osNpcSetRot`.

Επομένως, αρκεί να βρούμε την κατάλληλη γωνία περιστροφής στον άξονα αυτόν. Γνωρίζουμε επίσης ότι το εξ ορισμού μπροστινό διάνυσμα σε οποιοδήποτε αντικείμενο ή ενσάρκωση του Second Life και του OpenSimulator είναι το μοναδιαίο διάνυσμα που δείχνει στον άξονα X, δηλαδή το $\langle 1, 0, 0 \rangle$. Για να βρούμε λοιπόν τη συνολική περιστροφή του χαρακτήρα μας ώστε να κοιτάζει προς μία θέση-στόχο, αρκεί να υπολογίσουμε τη γωνία μεταξύ του μπροστινού διανύσματος και του διανύσματος που δείχνει προς τον στόχο (εικ. 11.3). Η LSL μάς δίνει μια συνάρτηση που υπολογίζει αυτόματα την περιστροφή μεταξύ δύο διανυσμάτων. Η συνάρτηση είναι:

```
rotation llRotBetween( vector start, vector end );
```

Χρησιμοποιώντας τη συνάρτηση αυτήν εντοπίζουμε την περιστροφή που πρέπει να κάνει ο χαρακτήρας από την αρχική του διεύθυνση προς την τελική διεύθυνση-στόχο (εικ. 11.3) και τη θέτουμε στον χαρακτήρα με τη χρήση της `osNpcSetRot`.

Ο παρακάτω κώδικας κάνει τον χαρακτήρα να στρέφεται διαρκώς προς τον χρήστη 'Test User'.

```
key npc;
integer alive = FALSE;

default {
    touch_start(integer num) {
        if(alive==FALSE) {
            npc = osNpcCreate("Agent", "Smith", llGetPos()+
                <2.0,0.0,0.0>, llDetectedKey(0));
            alive = TRUE;
            llSensorRepeat("Test User","",AGENT, 96.0, PI, 2.0);
        }
        else {
            osNpcRemove(npc);
            alive = FALSE;
            llSensorRemove();
        }
    }

    sensor(integer num) {
        vector userPos = llDetectedPos(0);
        vector npcPos = osNpcGetPos(npc);
        vector diff = userPos - npcPos;
        diff.z = 0;
        vector initial = <1.0,0.0,0.0>;
        osNpcSetRot(npc, llRotBetween(initial,diff));
    }
}
```

4 Κίνηση πλήθους

Σε όλα τα προηγούμενα παραδείγματα χρησιμοποιούσαμε ένα ξεχωριστό αντικείμενο το οποίο έλεγχε τη συμπεριφορά ενός χαρακτήρα. Ας ονομάσουμε το αντικείμενο αυτό αντικείμενο-ελεγκτή. Έστω ότι θέλουμε να δημιουργήσουμε πολλαπλούς διαδραστικούς χαρακτήρες στο περιβάλλον. Θα μπορούσαμε με ένα αντικείμενο-ελεγκτή να ελέγχουμε τη συμπεριφορά όλων των χαρακτήρων μας; Η απάντηση είναι ότι ναι, θα μπορούσαμε, αλλά ο κώδικας θα γινόταν αρκετά περίπλοκος. Οι λόγοι είναι οι εξής:

1. Το αντικείμενο-ελεγκτής θα πρέπει να χειρίζεται πολλαπλά κλειδιά αντί για ένα, και επομένως είναι απαραίτητη η χρήση λίστας για τη διαχείριση των χαρακτήρων.
2. Αν κάθε χαρακτήρας έχει τη δική του συμπεριφορά, θα χρειάζεται τις δικές του μεταβλητές και τους δικούς του αισθητήρες και θα έχει τις δικές του αντιδράσεις στα συμβάντα. Ο

συνδυασμός όλων των συμπεριφορών σε ένα πρόγραμμα θα το έκανε εξαιρετικά περίπλοκο και δύσκολο στην αποσφαλμάτωση.

Μια πιο ενδιαφέρουσα προσέγγιση, η οποία οδηγεί σε απλούστερα και ευκολότερα κλιμακούμενα αποτελέσματα, είναι το να έχουμε έναν ελεγκτή για κάθε χαρακτήρα. Όμως το να έχουμε πολλαπλά αντικείμενα-ελεγκτές στον κόσμο είναι δυνατόν να οδηγήσει σε προβλήματα, όπως το να μην γνωρίζουμε ποιο αντικείμενο ελέγχει ποιον χαρακτήρα. Για να αποφύγουμε το παραπάνω ενδεχόμενο, μπορούμε να έχουμε το αντικείμενο-ελεγκτή ενσωματωμένο στο σώμα του χαρακτήρα τον οποίο ελέγχει! Αυτή η προσέγγιση έχει ένα ακόμα πλεονέκτημα: η εντολή `llSensor` θα ψάχνει για γειτονικά αντικείμενα και χρήστες με κέντρο αναζήτησης το σώμα του χαρακτήρα και όχι τη θέση του αντικειμένου-ελεγκτή.

Ας δούμε ένα παράδειγμα που ακολουθεί την παραπάνω προσέγγιση: Αρχικά δημιουργήστε ένα αντικείμενο, για παράδειγμα έναν τόρο (`torus`), και προσθέστε τον παρακάτω κώδικα:

```
key npc;  
  
default {  
    state_entry() {  
        npc = llGetOwner();  
        llListen(110, "", "", "go");  
    }  
  
    listen(integer chn, string name, key id, string message) {  
        if(message=="go")  
            osNpcMoveToTarget(npc, llGetPos()+<5.0,0.0,0.0>,  
                OS_NPC_NO_FLY);  
    }  
}
```

Στο πρόγραμμα που μόλις γράψαμε κάνουμε το αντικείμενο να παρακολουθεί τα μηνύματα σε ένα συγκεκριμένο κανάλι επικοινωνίας και μόλις λάβει τη φράση “go” να μετακινεί τον χαρακτήρα 5 μέτρα στον θετικό άξονα X. Το κλειδί του χαρακτήρα το αποκτά μέσω της παρακάτω συνάρτησης, η οποία επιστρέφει το κλειδί του ιδιοκτήτη του αντικειμένου.

```
key llGetOwner();
```

Αν τρέξετε το πρόγραμμα, δεν θα συμβεί τίποτα, ακόμα και αν γράψετε “go” στο κρυφό κανάλι. Κι αυτό διότι δεν υπάρχει κανένας χαρακτήρας για να ελέγξει το πρόγραμμα. Ιδιοκτήτης του αντικειμένου είσατε εσείς, και επομένως στη μεταβλητή `npc` αποθηκεύεται το κλειδί της ενσάρκωσής σας. Όμως, αν ενσωματώσετε το αντικείμενο στην ενσάρκωσή σας και στη συνέχεια δημιουργήσετε έναν ψηφιακό χαρακτήρα βάσει της δικής σας εμφάνισης, ο χαρακτήρας αυτός θα έχει και αυτός ενσωματωμένο ένα αντίγραφο του αντικειμένου! Και στο αντίγραφο αυτό θα τρέχει ξανά ο παραπάνω κώδικας. Μόνο που αυτήν τη φορά θα έχει το κλειδί του ψηφιακού χαρακτήρα ο οποίος το φοράει, και ο οποίος θεωρείται «ιδιοκτήτης» του αντικειμένου, κι έτσι θα μπορεί να ελέγξει την κίνησή του.

Αν λοιπόν ενσωματώσετε τον τόρο στην ενσάρκωσή σας (π.χ. στο κεφάλι) και δημιουργήσετε ένα νέο αντικείμενο που παράγει συνθετικό χαρακτήρα-αντίγραφο, θα πρέπει να δείτε το αντίγραφό σας να «φοράει» τον ίδιο τόρο. Αυτήν τη φορά μπορείτε να γράψετε “go” στο κανάλι 110 και να το δείτε να εκτελεί την εντολή και να μετακινείται κατά πέντε μέτρα. Αν ξαναγράψετε την ίδια εντολή, ο χαρακτήρας θα μετακινηθεί άλλα πέντε μέτρα, γιατί μαζί με αυτόν άλλαξε και η θέση του αντικειμένου-ελεγκτή. Πράγματι, ακολουθώντας την προσέγγιση του ενσωματωμένου ελεγκτή, όλες οι σχετικές μετακινήσεις γίνονται σε σχέση με την τρέχουσα θέση του χαρακτήρα και όχι από τη σταθερή θέση ενός εξωτερικού αντικειμένου-ελεγκτή.

Η παραπάνω λογική μπορεί να επεκταθεί και στην κίνηση πλήθους. Μπορείτε να αποθηκεύσετε τις επιθυμητές συμπεριφορές των χαρακτήρων σε αντικείμενα-ελεγκτές και να δημιουργήσετε αντίστοιχα αρχεία εμφάνισης. Στη συνέχεια, από ένα νέο πρόγραμμα μπορείτε να εισάγετε τον αριθμό των χαρακτήρων που επιθυμείτε και να τους τοποθετήσετε στις κατάλληλες θέσεις. Έτσι, θα έχετε δημιουργήσει ένα εικονικό πλήθος όπου κάθε χαρακτήρας ακολουθεί κάποιο από τα προκαθορισμένα πρότυπα συμπεριφοράς. Στο παρακάτω

παράδειγμα κατασκευάζουμε εννέα χαρακτήρες και τους τοποθετούμε σε διάταξη πίνακα. Αν στην αποθηκευμένη συμπεριφορά υπάρχει ο ελεγκτής που παρουσιάσαμε προηγουμένως, τότε θα κινηθούν όλοι μαζί συντονισμένα μόλις στείλετε την εντολή “go” (εικ. 11.4).

```
integer alive=FALSE;
list keys;

default {
  touch_start(integer num) {
    if(alive==FALSE) {
      integer i;
      integer j;
      for(i=0;i<3;i++)
        for(j=0;j<3;j++) {
          key npc = osNpcCreate("Agent", "Smith",
            llGetPos()+<3.0+i*2.0,j*2.0,0>,
            llGetOwner());
          keys += [npc];
        }
      alive=TRUE;
    }
    else {
      integer i;
      for(i=0;i<llGetListLength(keys);i++) {
        osNpcRemove(llList2Key(keys,i));
      }
      alive=FALSE;
    }
  }
}
```



Εικόνα 11.4 Συντονισμένη κίνηση πλήθους

Αν δεν θέλετε να φαίνεται το αντικείμενο-ελεγκτής, μπορείτε να το κάνετε «αόρατο», δηλαδή να θέσετε βαθμό αδιαφάνειας μηδέν πριν το φορέσετε και πριν δημιουργήσετε χαρακτήρες βάσει της εμφάνισής σας. Μπορείτε, φυσικά, να αποθηκεύσετε την εμφάνισή σας και σε ένα notecard, ώστε να μη χρειάζεται να φοράτε το αντικείμενο-ελεγκτή κάθε φορά που θέλετε να δημιουργήσετε έναν νέο χαρακτήρα.

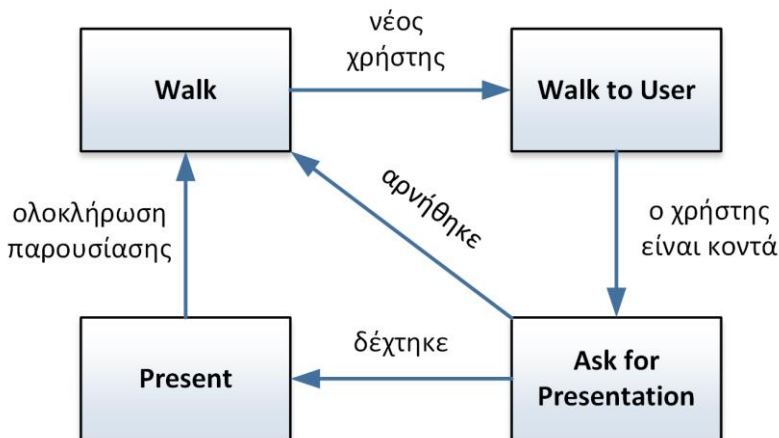
Τέλος, είναι προφανές ότι μπορείτε να δημιουργήσετε και άλλες, πιο ενδιαφέρουσες συμπεριφορές στον ελεγκτή σας και να τις συσχετίσετε με αντίστοιχα μηνύματα.

5 Σύνθετη συμπεριφορά χαρακτήρων

Μέχρι αυτό το σημείο είδαμε πώς μπορούμε να δημιουργήσουμε έναν ή περισσότερους χαρακτήρες με ενσωματωμένους ελεγκτές οι οποίοι τους οδηγούν σε ενδιαφέρουσες συμπεριφορές. Παρόλα αυτά, στα περισσότερα σενάρια αναμένεται να χρειαστούμε αρκετά πιο σύνθετες ενέργειες και αντιδράσεις σε ερεθίσματα του περιβάλλοντος, που θα κάνουν τους χαρακτήρες μας να δείχνουν πιο «έξυπνοι» και αληθοφανείς. Σε τέτοιες περιπτώσεις είναι χρήσιμο να εκμεταλλευτούμε τις μηχανές καταστάσεων που παρέχονται εγγενώς στη γλώσσα LSL και να τις χρησιμοποιήσουμε για να «απλοποιήσουμε» τη συμπεριφορά των χαρακτήρων σε επιμέρους καταστάσεις.

Ας υποθέσουμε ότι θέλουμε να κατασκευάσουμε έναν εικονικό χαρακτήρα ο οποίος συμπεριφέρεται ως ξεναγός σε μια έκθεση ή σε ένα μουσείο. Η συμπεριφορά που περιμένουμε να έχει ο χαρακτήρας είναι η εξής: Αρχικά θα βαδίζει ελεύθερα μέσα στο μουσείο παρατηρώντας τους επισκέπτες. Αν αντιληφθεί ότι ένας νέος επισκέπτης εισήλθε στον χώρο, ο χαρακτήρας θα βαδίσει προς το μέρος του, θα σταθεί και θα τον ρωτήσει αν ενδιαφέρεται να του προσφέρει μια ξενάγηση στον χώρο. Αν ο επισκέπτης συμφωνήσει, ο ξεναγός θα επισκεφθεί ένα ένα τα εκθέματα και θα τα παρουσιάσει. Αν όχι, θα επιστρέψει στην ελεύθερη κίνηση στον χώρο.

Η παραπάνω υποθετική συμπεριφορά θα μπορούσε να απλοποιηθεί μέσω μιας μηχανής καταστάσεων. Μπορούμε να εντοπίσουμε τέσσερις καταστάσεις στις οποίες είναι δυνατό να βρεθεί ο ξεναγός: να βαδίζει ελεύθερα στον χώρο, να κινηθεί προς το μέρος κάποιου χρήστη, να τον ρωτήσει αν επιθυμεί ξενάγηση και να κάνει την ξενάγηση. Για καθεμία από αυτές τις καταστάσεις μπορούμε επιπλέον να σκεφτούμε τις συνθήκες που θα οδηγούσαν στη μετάβαση σε κάποια άλλη. Αν ο ξεναγός βρίσκεται στην κατάσταση ελεύθερου βαδίσματος, θα αλλάξει σε κατάσταση κίνησης προς κάποιον χρήστη αν εντοπίσει έναν νέο επισκέπτη. Στη συνέχεια, ενώ κινείται προς τον χρήστη, θα αλλάξει στην κατάσταση ερώτησης για ξενάγηση αν πλησιάσει αρκετά κοντά. Αν ο επισκέπτης είναι θετικός, θα αλλάξει σε κατάσταση ξενάγησης, ενώ, αν είναι αρνητικός, θα επιστρέψει στην αρχική κατάσταση. Αυτή η ανάλυση σε καταστάσεις και συνθήκες μετάβασης περιγράφεται στο διάγραμμα καταστάσεων της εικόνας 11.5.



Εικόνα 11.5 Μηχανή καταστάσεων που περιγράφει τη συμπεριφορά του εικονικού ξεναγού.

Το επόμενο βήμα είναι να μελετήσουμε καθεμία από τις καταστάσεις που περιγράψαμε και να εξετάσουμε πώς θα μπορούσε αυτή να υλοποιηθεί σε LSL/OSSL διασφαλίζοντας τόσο ότι εκτελείται η απαιτούμενη συμπεριφορά όσο και ότι ελέγχονται σωστά οι συνθήκες μετάβασης σε άλλη κατάσταση.

Ας εξετάσουμε μία μία τις καταστάσεις και την πιθανή υλοποίησή τους.

1. *Κατάσταση ελεύθερου βαδίσματος (walk)*: ο χαρακτήρας θα ακολουθεί ένα προκαθορισμένο μονοπάτι στον εσωτερικό χώρο. Θα μπορούσαμε να κάνουμε την κίνησή του περισσότερο αληθοφανή αν προσθέταμε μερικές τυχαίες παύσεις (μέσω της εντολής `!Sleep`) κάθε φορά που πλησιάζει σε έναν στόχο. Παράλληλα, ο χαρακτήρας θα πρέπει να ελέγχει για νέους επισκέπτες. Για να το πετύχει αυτό, θα πρέπει να διατηρεί μια λίστα με τους χρήστες τους

οποίους έχει ήδη συναντήσει, για παράδειγμα `known_users`. Μπορεί να χρησιμοποιεί έναν επαναλαμβανόμενο αισθητήρα (`sensor`), για παράδειγμα κάθε δέκα δευτερόλεπτα, και να ελέγχει τα ονόματα των avatars στην περιοχή. Το πρώτο που θα βρεθεί να μην ανήκει στη λίστα θα σημειωθεί ως χρήστης-στόχος (π.χ. μεταβλητή `target_user` με το όνομα ή το κλειδί του χρήστη) και η κατάσταση θα αλλάξει σε βάδισμα προς τον χρήστη.

2. *Κατάσταση βαδίσματος προς τον χρήστη (`walk_to_user`):* ο χαρακτήρας θα ελέγχει επανειλημμένα τη θέση του χρήστη-στόχου μέσω κάποιου αισθητήρα και θα βαδίζει προς αυτή την κατεύθυνση, φροντίζοντας να κρατήσει μια σχετική απόσταση. Αν πλησιάσει αρκετά κοντά, θα αλλάξει στην κατάσταση ερώτησης σχετικά με παρουσίαση.
3. *Κατάσταση ερώτησης για παρουσίαση (`ask_presentation`):* ο χαρακτήρας θα στραφεί προς τον επισκέπτη και θα του στείλει ένα μήνυμα της μορφής: «Γεια σας, είμαι ο ξεναγός της έκθεσης. Θα θέλατε να σας παρουσιάσω τα εκθέματα; Απαντήστε ναι ή όχι». Ο χαρακτήρας θα ακούει για μηνύματα στο δημόσιο κανάλι μόνο από τον συγκεκριμένο επισκέπτη. Αν ο χρήστης αργήσει πολύ να απαντήσει, η απάντηση θα θεωρηθεί όχι. Αν η απάντηση είναι όχι, η κατάσταση θα αλλάξει σε ελεύθερο βάδισμα. Αν είναι ναι, θα αλλάξει σε παρουσίαση. Σε κάθε περίπτωση, ο επισκέπτης θα προστεθεί στη λίστα των γνωστών χρηστών (`known_users`).
4. *Κατάσταση ξενάγησης (`present`):* ο χαρακτήρας βαδίζει προς ένα έκθεμα, στέκεται δίπλα του, κοιτάζει προς τον χρήστη και λέει ορισμένες σχετικές πληροφορίες. Συνεχίζει στο επόμενο μέχρι να ολοκληρωθεί η ξενάγηση. Στο τέλος επιστρέφει στην αρχική κατάσταση ελεύθερου βαδίσματος.

Παρά το γεγονός ότι η συνολική συμπεριφορά του ξεναγού είναι αρκετά περίπλοκη, οι επιμέρους καταστάσεις που παρουσιάσαμε έχουν ανάλογη δυσκολία με τα παραδείγματα και τις ασκήσεις που έχουμε ήδη εξετάσει. Συνεπώς, μπορούμε να προχωρήσουμε στην υλοποίηση της παραπάνω συμπεριφοράς σε LSL κατασκευάζοντας αρχικά τις τέσσερις αυτές καταστάσεις και συμπληρώνοντας και ελέγχοντας σταδιακά τον κώδικα της καθεμίας. Η συμπεριφορά θα μπορούσε να υλοποιηθεί είτε με έναν σταθερό εξωτερικό ελεγκτή είτε με έναν ελεγκτή ενσωματωμένο στην ενσάρκωση. Αν και η δεύτερη περίπτωση έχει αρκετά πλεονεκτήματα, είναι σημαντικά πιο δύσκολη στην αποσφαλμάτωση, καθώς θα πρέπει κάθε φορά που κάνετε αλλαγή στον κώδικα του ελεγκτή να δημιουργείτε νέα ενσάρκωση και να δοκιμάζετε τη λειτουργία της από την αρχή.

6 Συμπεράσματα

Στο κεφάλαιο αυτό είδαμε πώς μπορούμε να εκμεταλλευτούμε τη γλώσσα LSL/OSSL για τη δημιουργία συνθετικών χαρακτήρων και τον έλεγχο της κίνησης και της συμπεριφοράς τους. Εξετάσαμε τους τρόπους με τους οποίους μπορούμε να εισάγουμε έναν νέο χαρακτήρα στον κόσμο και να τροποποιήσουμε την εμφάνισή του, παρουσιάσαμε τις δυνατότητες επένεργειας των χαρακτήρων στο περιβάλλον και προσεγγίσαμε περισσότερο σύνθετες συμπεριφορές, όπως η κίνηση πλήθους και η συμπεριφορά με μηχανές καταστάσεων.

Οι συνθετικοί χαρακτήρες μπορούν να αποτελέσουν μια ενδιαφέρουσα προσθήκη σε έναν εικονικό κόσμο σε πολλές και διαφορετικές περιπτώσεις. Μπορούν να λειτουργήσουν ως κοινό που βρίσκεται και περιφέρεται σε έναν χώρο αυξάνοντας την αληθοφάνειά του, να αναλάβουν ρόλο εικονικού ξεναγού ή εικονικού εκπαιδευτή, να δρουν ως ψηφιακοί χαρακτήρες σε περιβάλλοντα παιχνιδιού κ.λπ. Βεβαίως, η αλληλεπίδραση μεταξύ χρηστών και ψηφιακών χαρακτήρων θα μπορούσε να γίνει πιο ενδιαφέρουσα μέσω διεπαφής φυσικής γλώσσας ή ακόμα και φωνής, στοιχεία που δεν υποστηρίζονται άμεσα από τη γλώσσα σεναρίων και τις υποστηρικτικές βιβλιοθήκες του περιβάλλοντος. Είναι όμως δυνατό να επεκταθεί η λειτουργικότητα του κόσμου και προς αυτήν την κατεύθυνση παρεμβαίνοντας στο λογισμικό του διακομιστή ή και με διασύνδεση σε εξωτερικούς διακομιστές. Σε κάθε περίπτωση, τα εργαλεία που προσφέρονται από την OSSL για τον έλεγχο και τη δράση των ψηφιακών χαρακτήρων αποτελούν μια καλή αρχή για την παραγωγή πλούσιων συμπεριφορών.

Ασκήσεις

1. Κατασκευάστε τρία εκθέματα, για παράδειγμα πίνακες, μιας εικονικής γκαλερί και δημιουργήστε μια σύντομη ιστορία, βάσει της οποίας ο χαρακτήρας στέκεται δίπλα από καθένα από αυτά και το παρουσιάζει. Αλλάξτε την περιστροφή του χαρακτήρα σας όπου χρειάζεται.
2. Κατασκευάστε μια καρέκλα στην οποία μπορεί να κάθεται κάποιος (με τη χρήση της `ISitTarget`) και δημιουργήστε πολλαπλά αντίγραφα αυτής με το όνομα “chair”. Δημιουργήστε τόσους πράκτορες όσες και οι καρέκλες του χώρου σας και βάλτε τους να καθίσουν σε αυτές. Συμβουλή: μπορείτε με κατάλληλη κλήση της `ISensor` να εντοπίσετε όλα τα αντικείμενα με το όνομα “chair” και να μάθετε τα κλειδιά τους.
3. Κατασκευάστε ένα αντικείμενο το οποίο καταγράφει τις κινήσεις ενός χρήστη και δημιουργεί έναν συνθετικό χαρακτήρα με την εμφάνιση του χρήστη ο οποίος επαναλαμβάνει τις κινήσεις του. Χρησιμοποιήστε έναν διάλογο (`IDialog`) μέσω του οποίου ο χρήστης μπορεί να επιλέξει: έναρξη καταγραφής, τερματισμό καταγραφής, δημιουργία χαρακτήρα, αναπαραγωγή κίνησης και διαγραφή χαρακτήρα.
4. Κατασκευάστε έναν χαρακτήρα ο οποίος μετακινείται προς τον κοντινότερο χρήστη, του λέει “hello” και σταματάει. Συμβουλή: θα χρειαστεί να χρησιμοποιήσετε ένα πιο γενικό φίλτρο στο sensor, ώστε να ψάξετε για όλους τους χρήστες, και στη συνέχεια να ελέγξετε όλα τα αποτελέσματα εντοπίζοντας αυτόν που είναι κοντινότερος στον χαρακτήρα σας.
5. Κατασκευάστε έναν χαρακτήρα ο οποίος μόλις ακούσει ‘hello’ ή ‘hi’ στο δημόσιο κανάλι στρέφεται προς τον χρήστη που το είπε και του λέει ‘hello to you too’.
6. Δημιουργήστε εικονικό πλήθος με συμπεριφορά οδηγού–ακολουθών (leader–follower). Ένας οδηγός θα βαδίζει σε κάποιο προκαθορισμένο μονοπάτι και οι ακόλουθοι θα κινούνται πίσω του προσέχοντας να μην βρεθούν πολύ κοντά ο ένας στον άλλον. Συμβουλή: ο οδηγός και οι ακόλουθοι θα πρέπει να τρέχουν διαφορετικά προγράμματα. Κατασκευάστε πρώτα τον οδηγό και στη συνέχεια πολλαπλούς κλώνους από ακολούθους.
7. Κατασκευάστε έναν απλό εκθεσιακό χώρο και προσπαθήστε να υλοποιήσετε τη συμπεριφορά ξεναγού που περιγράψαμε στην ενότητα 5.

Κεφάλαιο 12: Ολοκληρωμένα Παραδείγματα

Σύνοψη

Στο παρόν κεφάλαιο παρουσιάζεται αναλυτικά η σχεδίαση και ανάπτυξη τριών ολοκληρωμένων παραδειγμάτων εφαρμογής στο *OpenSimulator*. Τα παραδείγματα είναι ένα παιχνίδι βολών τύπου *Angry Birds* βασισμένο στον μηχανισμό φυσικής του περιβάλλοντος, ένα περιβάλλον σύγχρονης συνεργασίας πλαισιωμένο από εργαλεία παρουσιάσεων, σημειώσεων και ψηφοφορίας, και ένας μηχανισμός εικονικής εξιστόρησης με ψηφιακούς χαρακτήρες. Για καθένα από τα παραπάνω γίνεται αναλυτική περιγραφή των επιμέρους βημάτων για τη σχεδίαση και την ανάπτυξή του. Παρουσιάζονται, μεταξύ άλλων, η σχεδίαση του περιβάλλοντος και της διεπαφής, η τεχνική σχεδίαση, η σταδιακή υλοποίηση, ο έλεγχος και η αποσφαλμάτωση. Επιπλέον παρατίθεται ένας αριθμός από πιθανές βελτιώσεις καθώς και επεκτάσεις σε ευρύτερους χώρους εφαρμογής.

Προαπαιτούμενη γνώση

Απαιτούνται γνώσεις κατασκευής περιεχομένου στο *OpenSimulator* και δεξιότητες προγραμματισμού σε *LSL/OSSL*.

1 Εισαγωγή

Σε αυτό το τελευταίο κεφάλαιο θα προσπαθήσουμε να θυμηθούμε και να αξιοποιήσουμε πολλές από τις γνώσεις και δεξιότητες που αποκτήσαμε στα προηγούμενα κεφάλαια του βιβλίου. Έχουμε ήδη εξετάσει αρκετές από τις θεωρητικές πτυχές των εικονικών κόσμων, της τεχνολογίας και των εφαρμογών τους. Μελετήσαμε θέματα απεικόνισης του περιβάλλοντος, κίνησης των ενεργών αντικειμένων και διεπαφής με τον χρήστη. Παρουσιάσαμε βασικές μεθόδους σχεδίασης, ανάπτυξης και αξιολόγησης εικονικών περιβαλλόντων και εξετάσαμε διάφορες περιοχές εφαρμογής οι οποίες έχουν επωφεληθεί από τους εικονικούς κόσμους και έχουν την προοπτική να εκμεταλλευτούν ακόμα περισσότερο τις προσφερόμενες λειτουργίες τους. Επιπλέον, εστίασαμε σε μια σύγχρονη τεχνολογία ανοιχτών κόσμων γενικού σκοπού, το *OpenSimulator*. Παρουσιάσαμε τις δυνατότητές του και τη διαδικασία εγκατάστασης και διαχείρισης, μελετήσαμε μεθόδους και τεχνικές ανάπτυξης περιεχομένου και εμβαθύνουμε στον προγραμματισμό της συμπεριφοράς των αντικειμένων και των συνθετικών χαρακτήρων μέσω της γλώσσας *LSL*. Το επόμενο βήμα είναι να μπορέσουμε να συνδυάσουμε αποδοτικά και δημιουργικά τις παραπάνω γνώσεις.

Στόχος μας στο παρόν κεφάλαιο είναι η βαθύτερη κατανόηση της διαδικασίας σχεδίασης και ανάπτυξης εφαρμογών για εικονικούς κόσμους μέσα από τη μελέτη τριών μεγαλύτερων και περισσότερο χρηστικών παραδειγμάτων. Έχουμε πλέον αποκτήσει μια καλή γνώση των λειτουργιών και των δυνατοτήτων της *LSL/OSSL* και έχουμε εξασκηθεί στην ανάπτυξη απλών προγραμμάτων μέσω των παραδειγμάτων και ασκήσεων που παρουσιάσαμε στα τρία προηγούμενα κεφάλαια. Όμως όλα αυτά δεν ήταν παρά μικρά μεμονωμένα παραδείγματα για την καλύτερη κατανόηση συγκεκριμένων συναρτήσεων και δομών της γλώσσας. Στη συνέχεια είναι σημαντικό να μπορέσουμε να «ενώσουμε τα κομμάτια». Να συνδυάσουμε τις δεξιότητες που αποκτήσαμε στην κατασκευή του περιβάλλοντος και στον προγραμματισμό μεμονωμένων συμπεριφορών σχεδιάζοντας και αναπτύσσοντας πιο μεγάλα και περίπλοκα παραδείγματα τα οποία εμπεριέχουν κάποιου είδους χρηστική αξία. Θα προσπαθήσουμε λοιπόν να δούμε και στην πράξη πώς θα μπορέσουμε να κάνουμε τα μεμονωμένα αντικείμενα να συνεργάζονται μεταξύ τους ώστε να επιδεικνύουν περισσότερες και πιο σύνθετες λειτουργίες. Επιπλέον, θα μπορούμε σε μια σχεδιαστική διαδικασία, δηλαδή θα προσπαθήσουμε να κατανοήσουμε καλύτερα τις ανάγκες του χώρου εφαρμογής και των χρηστών μας και βάσει αυτών να προσαρμόσουμε την εμφάνιση, τη διεπαφή και τις λειτουργίες των παραδειγμάτων μας.

Τα παραδείγματα που θα εξετάσουμε είναι τρία: ένα παιχνίδι βολών, ένας χώρος συνεργασίας και ένα σύστημα ψηφιακής εξιστόρησης. Στο πρώτο παράδειγμά μας θα προσπαθήσουμε να δημιουργήσουμε ένα παιχνίδι που ακολουθεί τη λογική του δημοφιλούς παιχνιδιού *Angry Birds*. Ο παίκτης θα προσπαθεί να χτυπήσει τους εχθρούς με όσο το δυνατόν λιγότερες προσπάθειες. Ο ίδιος θα ελέγχει ένα κανόνι που εκτελεί βολές, ενώ στον κόσμο μεταξύ του κανονιού και των εχθρών θα παρεμβάλλονται κατασκευές από τούβλο. Στο δεύτερο παράδειγμα θα δημιουργήσουμε έναν χώρο συνεργασίας για απομακρυσμένες διαλέξεις, παρουσιάσεις και συσκέψεις. Θα εξοπλίσουμε τον χώρο μας με εργαλεία προβολής και ελέγχου παρουσιάσεων, χώρο κοινών

σημειώσεων και μηχανισμό ψηφοφοριών. Παράλληλα, θα διαμορφώσουμε και θα διατάξουμε κατάλληλα τα στοιχεία στον χώρο με στόχο την ενίσχυση της λειτουργικότητας και ευχρηστίας του. Τέλος, στο τρίτο παράδειγμά μας θα εκμεταλλευτούμε τους συνθετικούς χαρακτήρες για την αφήγηση ιστοριών. Θα αναπτύξουμε έναν μηχανισμό μέσω του οποίου θα μπορούν οι χρήστες σχετικά εύκολα να σχεδιάσουν και να δοκιμάσουν τις δικές τους ιστορίες. Θα μπορούν να δημιουργήσουν χαρακτήρες και να γράψουν σενάρια τα οποία θα εκτελούνται από δύο ή παραπάνω χαρακτήρες και θα συσχετίζονται με αντικείμενα και περιοχές του περιβάλλοντος στα πλαίσια σύνθετων ιστοριών.

Η επιλογή των παραδειγμάτων έγινε με γνώμονα την κάλυψη όσο το δυνατόν μεγαλύτερου εύρους χαρακτηριστικών και λειτουργιών των εικονικών κόσμων, με βάση πάντα και τις προσφερόμενες δυνατότητες του OpenSimulator. Στο παιχνίδι θα μελετήσουμε θέματα φυσικής, συγκρούσεων και δυναμικής εισαγωγής και κίνησης αντικειμένων. Στον χώρο συνεργασίας θα μάθουμε τρόπους διασύνδεσης του περιεχομένου του κόσμου με εξωτερικές πηγές, το οποίο θα μας χρειαστεί για την προβολή των παρουσιάσεων. Επιπλέον, θα δούμε θέματα οργάνωσης και διάταξης αντικειμένων, διεπαφής με τον χρήστη κ.ά. Τέλος, στο περιβάλλον εξιστόρησης θα εξετάσουμε τεχνικές με τις οποίες μπορούμε να διαβάζουμε και να επεξεργαζόμαστε ακολουθίες εντολών από αρχεία κειμένου και θα δούμε πώς υλοποιούνται και συνδυάζονται σύνθετες λειτουργίες εικονικών χαρακτήρων. Οι δραστηριότητες χρηστών που υποστηρίζουν τα παραπάνω είναι επίσης πολλαπλές. Το ένα περιβάλλον θα εστιάζει στην πρόκληση και στην ψυχαγωγία, το δεύτερο στην αποδοτική επικοινωνία και συνεργασία και το τρίτο στην ανάλυση, σχεδίαση και εκτέλεση ιστοριών. Επεκτάσεις και εφαρμογές των συστημάτων που θα υλοποιήσουμε μπορούμε επίσης να δούμε αρκετές: μάθηση μέσω παιχνιδιού, εικονικοί χώροι διδασκαλίας, συνεργατική σχεδίαση, εικονικοί παρουσιαστές, εικονικοί ξεναγοί, αναπαράσταση ιστορικών γεγονότων κ.ά.

Κατά τη μελέτη του κεφαλαίου αυτού θα είναι χρήσιμο να προσπαθήσετε να κατασκευάσετε τους κόσμους των παραδειγμάτων, να κατανοήσετε τη διαδικασία σχεδίασης και τις λύσεις σε επίπεδο κώδικα που παρουσιάζουμε και, κυρίως, να δοκιμάσετε τις δικές σας παραλλαγές και βελτιώσεις. Σε καθένα από τα παραδείγματα δίνουμε την ιδέα, μια προτεινόμενη λύση σε επίπεδο σχεδίασης του κόσμου και υλοποίησης της συμπεριφοράς, και πιθανές επεκτάσεις. Εσείς από τη μεριά σας θα είναι καλό να συμπληρώσετε τις δικές σας ιδέες, να δοκιμάσετε εναλλακτικές υλοποιήσεις, να εξετάσετε τρόπους βελτίωσης της διεπαφής και της λειτουργικότητας και, γενικότερα, να πειραματιστείτε τόσο με τις σχεδιαστικές επιλογές όσο και με τις τεχνικές λύσεις που επιλέξαμε. Η διαδικασία αυτή αναμένεται να σας βοηθήσει να αποκτήσετε μια πιο ολιστική κατανόηση του χώρου και να αποτελέσει ένα καλό πρώτο βήμα για τη δημιουργική αξιοποίηση των γνώσεων που αποκτήσατε σε αυτό το βιβλίο.

2 Ένα παιχνίδι τύπου Angry Birds

Το παράδειγμα το οποίο θα μελετήσουμε βασίζεται στην ιδέα του δημοφιλούς παιχνιδιού Angry Birds. Αν και το εν λόγω παιχνίδι είναι σχετικά πρόσφατο, παιχνίδια παραπλήσιας λογικής υπάρχουν εδώ και δεκαετίες. Η πρόκληση, συνοπτικά, είναι η εξής: Ο παίκτης ελέγχει έναν μηχανισμό που εκτοξεύει βλήματα και έχει ως στόχο να πετύχει και τελικά να καταστρέψει κάποιον «εχθρό». Μεταξύ του παίκτη και του εχθρού μπορεί να παρεμβάλλονται διάφορα αντικείμενα, τα οποία είτε επηρεάζονται από το βλήμα και μετακινούνται είτε καταστρέφονται (π.χ. τζάμια, τούβλα κ.λπ.), είτε μένουν ανεπηρέαστα και εμποδίζουν το βλήμα να φτάσει στον στόχο του (π.χ. μορφολογία του εδάφους). Συνήθως τα παιχνίδια αυτά έχουν έναν καλό μηχανισμό φυσικής που αποδίδει την κίνηση και την κρούση των αντικειμένων με αρκετά αληθοφανή τρόπο. Πολλές φορές μια βολή μπορεί να προκαλέσει σειρά συγκρούσεων και να έχει απρόβλεπτες συνέπειες. Η βασική πρόκληση για τον παίκτη είναι να σχεδιάσει όσο το δυνατόν καλύτερα τις βολές του ώστε να πετύχει τον στόχο με τις λιγότερες δυνατές προσπάθειες. Θα πρέπει σταδιακά να αποκτήσει την ικανότητα να προβλέπει το πώς η γωνία, η δύναμη βολής αλλά και οι πιθανές συγκρούσεις του βλήματος με αντικείμενα του περιβάλλοντος επηρεάζουν το αποτέλεσμα, και να μάθει τελικά να ελέγχει αποδοτικότερα τον μηχανισμό εκτόξευσης. Ένα βασικό στοιχείο που οδηγεί σε διασκέδαση στα παιχνίδια αυτού του τύπου είναι η απρόβλεπτη εξέλιξη των ενεργειών του παίκτη, το γεγονός ότι τα αποτελέσματα των πολλαπλών κρούσεων μπορεί να είναι διασκεδαστικά, αλλά και το ότι συνήθως δεν υπάρχει ένας μοναδικός τρόπος για ολοκλήρωθεί η πίστα.

Στόχος του παραδείγματός μας είναι να κατασκευάσουμε μια διασκεδαστική μεταφορά του παραπάνω στιλ παιχνιδιού στους εικονικούς κόσμους. Καταρχήν μπορούμε να εκμεταλλευτούμε το γεγονός ότι το περιβάλλον μας είναι τρισδιάστατο και να προσθέσουμε μία ακόμη διάσταση στο παιχνίδι. Πράγματι, τα περισσότερα παιχνίδια αυτής της μορφής είναι δισδιάστατα, κυρίως για λόγους απλότητας. Η χρήση της τρίτης

διάστασης θα μπορούσε όμως να κάνει το παιχνίδι πιο ενδιαφέρον και να οδηγήσει σε πιο περίπλοκες πίστες. Άλλωστε, οι χρήστες των εικονικών κόσμων έχουν αποκτήσει ήδη σε κάποιο βαθμό την ικανότητα να αντιλαμβάνονται το περιβάλλον και τα αντικείμενα και να πλοηγούνται στον τρισδιάστατο χώρο, και θεωρούν φυσιολογικό και αναμενόμενο οι δραστηριότητές τους στον κόσμο να εκμεταλλεύονται επαρκώς και τις τρεις διαστάσεις. Δεύτερον, οι εικονικοί κόσμοι προσφέρονται για παιχνίδια βασισμένα στη φυσική, καθώς διαθέτουν ως επί το πλείστον σχετικό ενσωματωμένο μηχανισμό. Αυτό μας απεμπλέκει από την ανάγκη να υλοποιήσουμε εξαρχής αλγορίθμους φυσικής κίνησης και κρούσης. Στην υλοποίησή μας απλά θα ορίσουμε κατάλληλες αρχικές συνθήκες και θα αφήσουμε τον μηχανισμό να χειριστεί την κίνηση των αντικειμένων. Τέλος, το γεγονός ότι οι εικονικοί κόσμοι είναι πολυχρηστικά περιβάλλοντα θα μπορούσε να οδηγήσει σε ενδιαφέρουσες επεκτάσεις του παιχνιδιού για πολλαπλούς παίκτες και να συμπεριληφθούν συνθήκες συνεργασίας ή και ανταγωνισμού.

2.1 Σχεδίαση περιβάλλοντος και διεπαφής

Αρχικά θα μελετήσουμε τις βασικές οντότητες αυτού του είδους παιχνιδιών με στόχο να καταλήξουμε στα αντίστοιχα αντικείμενα του εικονικού κόσμου, στις ιδιότητες και στην αναμενόμενη συμπεριφορά τους.

Τα είδη οντοτήτων που μπορεί να εντοπίσει κάποιος σε αυτού του είδους τα παιχνίδια είναι:

- ένας ή περισσότεροι *εκτοξευτήρες* τους οποίους ελέγχουν οι χρήστες,
- ένα ή περισσότερα είδη *βλημάτων* με διαφορές στις αλληλεπιδράσεις τους με τις υπόλοιπες οντότητες,
- οι *εχθροί*, οι οποίοι έχουν κάποιου είδους «ενέργεια», που μειώνεται όταν συγκρούονται με βλήματα ή άλλα αντικείμενα,
- *αντικείμενα που επηρεάζονται από συγκρούσεις* και ενίοτε καταστρέφονται και
- *αντικείμενα που μένουν ανεπηρέαστα από συγκρούσεις*.

Με βάση τις παραπάνω κατηγορίες και προσπαθώντας να δώσουμε μία όσο το δυνατό πιο απλή λύση στον σχεδιασμό του παιχνιδιού καταλήγουμε στα εξής είδη αντικειμένων που θα υπάρχουν σε ένα ή περισσότερα στιγμιότυπα στις πίστες μας: κανόνι, βλήμα, γουρουνάκι και τούβλο. Ακολουθεί η ανάλυση της εμφάνισης και βασικής λειτουργίας των αντικειμένων αυτών:

Κανόνι. Ο εκτοξευτήρας μας θα έχει τη μορφή κανονιού. Θα έχει τη δυνατότητα να κινείται στο έδαφος, να αλλάζει διεύθυνση και να εκτοξεύει βλήματα. Ο έλεγχος του κανονιού θα γίνεται από τον χρήστη.

Βλήμα. Το βλήμα θα εκτοξεύεται από το κανόνι όταν ο χρήστης το επιλέξει. Η εκτόξευσή του θα γίνεται από την κάννη και προς τη διεύθυνση στην οποία είναι στραμμένο το κανόνι, ενώ η ταχύτητα εκτόξευσης θα είναι μεταβλητή. Θα συμπεριφέρεται ως φυσικό αντικείμενο, δηλαδή η κίνησή του θα επηρεάζεται από τη βαρύτητα, και θα συγκρούεται με άλλες οντότητες του περιβάλλοντος καθώς και με το έδαφος.

Γουρουνάκι. Το γουρουνάκι θα είναι ο εχθρός μας. Θα υπάρχουν παραπάνω από ένα στην πίστα του παιχνιδιού και στόχος του παίκτη θα είναι να τα σκοτώσει όλα. Τα γουρουνάκια θα έχουν μια αρχική ενέργεια, η οποία θα μειώνεται σταδιακά όσο δέχονται συγκρούσεις από άλλες οντότητες. Θα έχουν και αυτά φυσική συμπεριφορά, δηλαδή οι συγκρούσεις θα επηρεάζουν τη θέση και τον προσανατολισμό τους. Επιπλέον, η κατάσταση της ενέργειάς τους θα αντικατοπτρίζεται στην εμφάνισή τους. Αν η ενέργεια πέσει στο μηδέν, το γουρουνάκι θεωρείται νεκρό και παύει να έχει φυσική συμπεριφορά.

Τούβλο. Στην πίστα θα υπάρχουν πολλαπλά τούβλα, τα οποία θα σχηματίζουν φυσικά εμπόδια μεταξύ του εκτοξευτήρα και των εχθρών. Τα τούβλα θα έχουν φυσική συμπεριφορά, δηλαδή θα δέχονται δυνάμεις από τα βλήματα και θα αντιδρούν ανάλογα.

Τέλος, οποιοδήποτε άλλο μη φυσικό αντικείμενο ή και το ίδιο το έδαφος θα μπορούν να λειτουργούν ως εμπόδια στο παιχνίδι μας, τα οποία μένουν ανεπηρέαστα από συγκρούσεις.

Εκτός από παραπάνω είδη αντικειμένων, θα κατασκευάσουμε ένα ακόμη, το **αντικείμενο παιχνιδιού (game object)**, του οποίου ο ρόλος θα είναι να συντονίζει τα παραπάνω, να κατασκευάζει την πίστα, να καταγράφει το σκορ και να ελέγχει τη ροή του παιχνιδιού. Σε περιβάλλοντα εκτέλεσης στα οποία δεν υπάρχει κάποιος κεντρικός κώδικας παρά μόνο επιμέρους τμήματα κώδικα που τρέχουν στα αντικείμενα, είναι πάντοτε σκόπιμο να υπάρχει κάποιο αντικείμενο που συντονίζει τη λειτουργία των υπολοίπων και εμπεριέχει τη γενικότερη λογική της εφαρμογής. Το OpenSimulator είναι ένα τέτοιο περιβάλλον, και επομένως αναμένεται

να διευκολύνει σημαντικά την καλή οργάνωση και διαχείριση του κώδικα η ύπαρξη ενός αντικειμένου με τα παραπάνω καθήκοντα.

Στη συνέχεια θα πρέπει να λάβουμε ορισμένες βασικές αποφάσεις σχετικά με τη μηχανική του παιχνιδιού και τον τρόπο παιζίματος (gameplay). Οι επιλογές μας είναι οι παρακάτω:

- Τυχασιότητα στην κατασκευή της πίστας. Επιλέγουμε να κατασκευάσουμε την πίστα εισάγοντας στοιχεία τυχασιότητας, με στόχο να μεταβάλλεται ο βαθμός δυσκολίας αλλά και η πρόκληση που παρουσιάζει η κάθε πίστα.
- Περιορισμός των βαθμών ελευθερίας του κανονιού. Θεωρητικά το κανόνι μας θα μπορούσε να έχει τέσσερις βαθμούς ελευθερίας στην κίνησή του. Δύο για τη μετατόπιση στο οριζόντιο επίπεδο (μπροστά–πίσω και αριστερά–δεξιά) και δύο για την αλλαγή της διεύθυνσής του (περιστροφή πάνω–κάτω και αριστερά–δεξιά). Κάτι τέτοιο όμως θα έκανε το παιχνίδι μας αρκετά πολύπλοκο, καθώς θα έπρεπε ο παίκτης να ελέγξει πέντε διαφορετικές μεταβλητές συνυπολογίζοντας την ένταση της βολής. Περιορίζουμε τους βαθμούς ελευθερίας σε δύο: κίνηση του κανονιού αριστερά ή δεξιά και περιστροφή του πάνω ή κάτω.
- Συγκρούσεις αντικειμένων με εχθρό. Επιλέγουμε να είναι διαφορετική η μείωση της ενέργειας του γουρουνιού ανάλογα με το είδος του αντικειμένου που συγκρούστηκε με αυτό. Το βλήμα θα έχει μεγαλύτερη επίδραση και το τούβλο μικρότερη. Θεωρητικά όμως μπορεί να επιλέξει ο παίκτης να χτυπήσει τον εχθρό καταφέροντας να ρίξει έναν αριθμό από τούβλα πάνω του μέσω κατάλληλα σχεδιασμένης βολής.

Συνθήκη τερματισμού είναι το να σκοτωθούν όλοι οι εχθροί στην πίστα και η αξιολόγηση του παίκτη γίνεται βάσει του αριθμού των προσπαθειών και του αριθμού των εχθρών που σκότωσε.

Τέλος, θα προσδιορίσουμε τα βασικά στοιχεία διεπαφής του χρήστη με το περιβάλλον.

- Οι επιλογές του παιχνιδιού θα προσφέρονται μέσω αλληλεπίδρασης με το αντικείμενο παιχνιδιού. Ο χρήστης θα μπορεί μέσω κάποιου ειδικού μενού να:
 - δημιουργήσει μια νέα πίστα,
 - επανεκκινήσει την υπάρχουσα πίστα ή
 - τερματίσει το παιχνίδι.
- Κατά τη διάρκεια του παιχνιδιού θα μπορεί να διαβάσει την επίδοσή του μέσω κειμένου που θα εμφανίζεται στο αντικείμενο του παιχνιδιού.
- Τον κόσμο του παιχνιδιού θα μπορεί να τον παρατηρήσει μέσω των ενεργειών πλοήγησης/εξερεύνησης στον εικονικό κόσμο.
- Η κατάσταση των εχθρών θα απεικονίζεται τόσο με το χρώμα τους όσο και με κάποιο κείμενο στο οποίο θα αναφέρεται αριθμητικά η ενέργειά τους.
- Ο χειρισμός του κανονιού θα γίνεται με εισαγωγή κειμένου. Ο χρήστης θα προσδιορίζει:
 - την οριζόντια μετατόπιση (offset) του κανονιού,
 - τη γωνία βολής (angle) και
 - την ένταση (strength).

Ο τρόπος χειρισμού του κανονιού είναι, προφανώς, αρκετά διαφορετικός σε σχέση π.χ. με το Angry Birds, και σίγουρα λιγότερο διασκεδαστικός. Όμως, στο OpenSimulator είναι δύσκολο να υποστηριχθούν ενέργειες τύπου drag και να γίνει αναλογικός προσδιορισμός της γωνίας και της δύναμης μέσα από τη διεπαφή του κόσμου. Άλλωστε, σε παλιότερες εκδόσεις αυτών των παιχνιδιών η είσοδος δινόταν αριθμητικά από τον χρήστη (π.χ. στο παιχνίδι Gorillas στο MS Dos), επομένως είναι μια επιλογή που έχει λειτουργήσει και στο παρελθόν.

2.2 Τεχνική σχεδίαση

Τα αντικείμενα που θα πρέπει να κατασκευάσουμε στο OpenSimulator είναι αυτά που προσδιορίσαμε στην προηγούμενη φάση: το αντικείμενο παιχνιδιού, το κανόνι, το βλήμα, το γουρουνάκι και το τούβλο. Από αυτά, τα τρία τελευταία θα είναι φυσικά αντικείμενα, καθώς θα κινούνται και θα συγκρούονται μεταξύ τους στη διάρκεια του παιχνιδιού. Επιπλέον, το βλήμα θα είναι και προσωρινό (temporary) αντικείμενο, ώστε να διαγράφεται αυτόματα από τον κόσμο μετά από κάποιο χρονικό διάστημα.

2.2.1 Έλεγχος παιχνιδιού

Το αντικείμενο παιχνιδιού θα πρέπει να μπορεί να ελέγχει και να δημιουργεί στιγμιότυπα από όλα τα υπόλοιπα αντικείμενα, ώστε να υποστηρίζει τις επιλογές του χρήστη, δηλαδή τη δημιουργία νέας πίστας, την επανεκκίνηση της πίστας και τον τερματισμό του παιχνιδιού.

Για να μπορεί το αντικείμενο παιχνιδιού να δημιουργήσει μια νέα πίστα, θα πρέπει να μπορεί να εισάγει πολλαπλά στιγμιότυπα των υπόλοιπων αντικειμένων στον κόσμο και να τα τοποθετήσει κατάλληλα. Επομένως, θα έχει στα περιεχόμενά του το κανόνι, το γουρουνάκι και το τούβλο και θα τα προσθέτει στον κόσμο μέσω της διαδικασίας `rez` που έχουμε περιγράψει στο Κεφάλαιο 10.

Για να ξεκινήσει μια πίστα ξανά, θα πρέπει όλα τα αντικείμενα να επιστρέψουν στην αρχική τους κατάσταση. Η λύση γι' αυτό είναι η χρήση μιας συνάρτησης `reset()` την οποία θα έχει υλοποιημένη στο `script` του καθένα από τα τρία παραπάνω αντικείμενα. Και τα τρία αντικείμενα (καθώς και όλα τα αντίγραφα τους που θα εισαχθούν στον κόσμο) θα «ακούνε» σε ένα κρυφό κανάλι, στο οποίο το αντικείμενο παιχνιδιού θα στέλνει μηνύματα. Έτσι, όταν λάβουν το μήνυμα 'reset', θα εκτελούν τη σχετική συνάρτηση, η οποία θα τους επαναφέρει στην αρχική κατάσταση.

Για να τερματιστεί το παιχνίδι θα πρέπει τα αντικείμενα να μπορούν να «εξαφανιστούν» από τον κόσμο. Η ίδια διαδικασία θα πρέπει να εκτελείται και πριν από τη δημιουργία μιας νέας πίστας. Όταν λάβουν μήνυμα 'die' στο κρυφό κανάλι επικοινωνίας από τον ελεγκτή παιχνιδιού, θα πρέπει να διαγράφονται από τον κόσμο. Αυτό γίνεται μέσω LSL με τη συνάρτηση `IIDie()`.

2.2.2 Απεικόνιση κατάστασης

Όπως αναφέραμε και πιο πάνω, το αντικείμενο παιχνιδιού θα απεικονίζει τις επιδόσεις του παίκτη, δηλαδή τον αριθμό των βολών που έριξε και τον αριθμό των εχθρών που σκότωσε. Τα στοιχεία αυτά θα αναγράφονται σε μορφή κειμένου με τη χρήση της `IISetText`. Θα πρέπει όμως το αντικείμενο να ενημερώνεται σχετικά με την εξέλιξη του παιχνιδιού. Για τον σκοπό αυτόν θα αξιοποιηθεί ξανά το κανάλι επικοινωνίας που αναφέραμε πιο πάνω και θα ειδοποιούν μέσω αυτού τόσο το κανόνι (μήνυμα 'shot') όσο και το γουρουνάκι (μήνυμα 'died') όταν εκτελείται μια νέα βολή ή η ενέργεια έπεσε στο μηδέν αντίστοιχα. Στον πίνακα 12.1 περιγράφονται όλα τα μηνύματα που θα ανταλλάσσονται στο παιχνίδι.

Μήνυμα	Ερμηνεία
reset	επιστροφή αντικειμένου στην αρχική κατάσταση
die	διαγραφή αντικειμένου από τον κόσμο
shot	εκτέλεση βολής από το κανόνι
died	θάνατος εχθρού

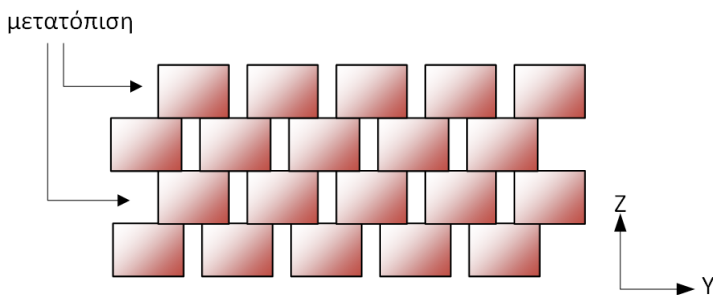
Πίνακας 12.1 Τα μηνύματα που ανταλλάσσονται στο ειδικό κανάλι του παιχνιδιού.

Ο εχθρός θα πρέπει και αυτός να κάνει ορατή την κατάστασή του. Θα περιλαμβάνει μια εσωτερική μεταβλητή ενέργειας, η οποία θα μειώνεται κάθε φορά που δέχεται χτύπημα από βλήμα ή τούβλο. Η τρέχουσα τιμή θα γίνεται ορατή με δύο τρόπους: με κείμενο που θα απεικονίζεται πάνω του (μέσω της `IISetText`) και με αλλαγή στο χρώμα του. Επιλέγουμε να ξεκινάει από το πράσινο χρώμα (όπως είναι τα γουρουνάκια στο *Angry Birds*) και να καταλήγει σταδιακά στο γκρίζο. Αυτό μπορούμε να το πετύχουμε αν μετασχηματίσουμε κατάλληλα την τιμή της ενέργειας και την αναθέσουμε στην τιμή του πράσινου αλλάζοντας το χρώμα του αντικειμένου. Έτσι, το αντικείμενο θα ξεκινάει από την τιμή χρώματος (0.5, 1, 0.5) που είναι ανοιχτό πράσινο και θα καταλήγει στην (0.5, 0.5, 0.5). Τέλος, όταν ο εχθρός χάνει τη ζωή του, θα στέλνει σχετικό μήνυμα στο δημόσιο κανάλι για την ενημέρωση του παίκτη και θα «αναποδογυρίζει» (περιστροφή 180° στον άξονα X) δίνοντας την εικόνα του πεθαμένου.

2.2.3 Σχεδίαση Πίστας

Οι πίστες των παιχνιδιών δημιουργούνται βάσει αντίστοιχων αρχείων περιγραφής, έτσι ώστε να είναι σε θέση οι δημιουργοί να τροποποιήσουν ή και να επαυξήσουν τις πίστες χρησιμοποιώντας ειδικά προγράμματα επεξεργασίας (level editors). Στην περίπτωση μας, για λόγους απλότητας, θα υπάρχει ένα μόνο είδος πίστας το οποίο θα δημιουργείται αλγοριθμικά και θα περιλαμβάνεται κάποιος βαθμός τυχαιότητας στην τοποθέτηση των στοιχείων.

Επιλέγουμε να έχουμε μόνιμα τοποθετημένο έναν τούβλινο τοίχο στο κέντρο της πίστας αποτελούμενο από 20 τούβλα (τέσσερις γραμμές από πέντε τούβλα η καθεμία). Θα αφήνουμε ένα μικρό οριζόντιο κενό μεταξύ των τούβλων, έτσι ώστε να είναι ευκολότερη η κατάρρευση τμημάτων του τοίχου. Επιπλέον, κάθε τούβλο μιας γραμμής θα τοποθετείται ανάμεσα σε δύο τούβλα της πιο κάτω γραμμής, όπως συμβαίνει και με τους πραγματικούς τοίχους. Η τελική μορφή του τοίχου απεικονίζεται στην εικόνα 12.1. Η τοποθέτηση των τούβλων θα γίνει στο επίπεδο YZ με δύο ένθετες εντολές επανάληψης τύπου for. Η εξωτερική θα δημιουργεί τις γραμμές ξεκινώντας από το ύψος του αντικείμενου παιχνιδιού (που θα πρέπει να βρίσκεται στο έδαφος) και ανεβαίνοντας προς τα πάνω. Η εσωτερική θα τοποθετεί τα τούβλα της κάθε γραμμής ξεκινώντας από μια αρχική οριζόντια απόσταση (Y) από το αντικείμενο παιχνιδιού, αφήνοντας ένα μικρό κενό μεταξύ των τούβλων και προσθέτοντας μια μετατόπιση στις ζυγές γραμμές.

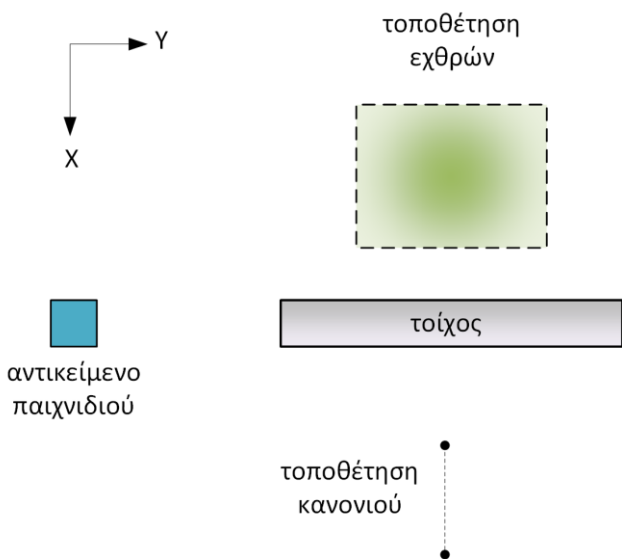


Εικόνα 12.1 Κατασκευή του τούβλινου τοίχου στο παιχνίδι.

Ο αριθμός των εχθρών και η τοποθέτησή τους θα είναι τυχαία. Οι εχθροί θα είναι από τρεις μέχρι πέντε και θα τοποθετούνται σε ένα νοητό ορθογώνιο πίσω από τον τούβλινο τοίχο. Ο υπολογισμός του αριθμού των εχθρών και της οριζόντιας θέσης (X και Y) κάθε εχθρού θα γίνεται με κατάλληλη χρήση της συνάρτησης παραγωγής ψευδοτυχαίων αριθμών της LSL, δηλαδή της IFRand.

Τέλος, και η απόσταση (X) του κανονιού από τον τοίχο θα είναι τυχαία μέσα σε κάποια νοητά όρια, ώστε να αλλάζει κάθε φορά η πρόκληση.

Η κάτοψη της πίστας και τα όρια τοποθέτησης των στοιχείων απεικονίζεται στην εικόνα 12.2.



Εικόνα 12.2 Κάτοψη πίστας παιχνιδιού και περιοχές τοποθέτησης στοιχείων.

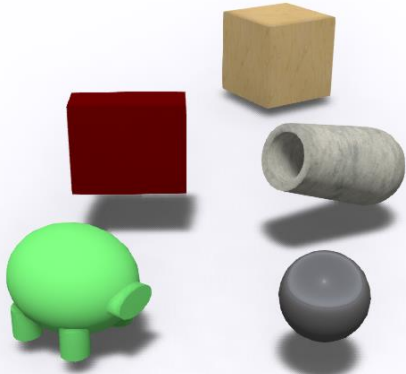
2.2.4 Μηχανική Παιχνιδιού

Τα στοιχεία μηχανικής του παιχνιδιού είναι εξαιρετικά απλά, με δεδομένο ότι η φυσική μοντελοποίηση και οι συγκρούσεις υποστηρίζονται ήδη από το περιβάλλον και δεν μπορούν να τροποποιηθούν. Αφορούν μόνο τις παραμέτρους εκτόξευσης του βλήματος και τις συνέπειες των συγκρούσεων με τους εχθρούς.

Σύμφωνα με τις σχεδιαστικές αποφάσεις που λάβαμε πιο πάνω, ο χρήστης θα ελέγχει την οριζόντια μετατόπιση, τη γωνία και την ένταση της βολής. Για λόγους απλότητας θα ζητάμε από τον χρήστη να μας δώσει τις τιμές αυτές σε ένα εύρος που θα μπορεί εύκολα να θυμηθεί. Η τιμή μετατόπισης θα είναι μεταξύ του -1 (αριστερό άκρο) και 1 (δεξί άκρο), η τιμή της γωνίας θα είναι μεταξύ του 0 και 90 (μοίρες), και η τιμή της έντασης μεταξύ του 0.1 (ελάχιστη) και 1 (μέγιστη). Όταν θα εκτελείται η βολή, οι τιμές θα μετασχηματίζονται κατάλληλα.

- Η μετατόπιση θα γίνεται στον άξονα Y. Η τιμή μετατόπισης θα πολλαπλασιάζεται επί το μισό μήκος του τοίχου, ώστε να μπορεί το κανόνι να καλύψει όλες τις πιθανές θέσεις βολής. Το κανόνι θα μετατοπίζεται μέσω της `IISetPos`.
- Η τιμή αλλαγής γωνίας θα μετασχηματίζεται σε αντίστοιχη περιστροφή στον άξονα Y και θα αποδίδεται στο κανόνι μέσω της `IISetRot`.
- Η τιμή έντασης θα μετασχηματίζεται κατάλληλα, ώστε η ταχύτητα που θα αποκτά το βλήμα να είναι σε εύλογα για την πίστα όρια. Το κανόνι θα εισάγει (`IIRezObject`) ένα νέο αντικείμενο βλήματος στον κόσμο. Θα το τοποθετεί στη θέση που βρίσκεται η κάννη και θα αποκτά αρχική ταχύτητα με την επιθυμητή ένταση και διεύθυνση.

Όταν ένα αντικείμενο προσκρούσει στον εχθρό, αυτός θα πρέπει να είναι σε θέση να το αντιληφθεί. Για τον λόγο αυτόν θα χρησιμοποιήσουμε το συμβάν `collision_start`. Στη συνέχεια, ο εχθρός θα εξετάζει το είδος του αντικειμένου που προσέκρουσε πάνω του διαβάζοντας το όνομά του (`IIDetectedName`). Αν είναι τούβλο 'brick', η εσωτερική του ενέργεια θα μειώνεται κατά ένα ποσοστό, αν είναι βλήμα 'ball', θα μειώνεται κατά ένα μεγαλύτερο. Και στις δύο περιπτώσεις θα ανανεώνεται η εμφάνισή του (χρώμα και κείμενο) και θα ελέγχεται αν η τιμή έχει πέσει κάτω από το μηδέν, που σημαίνει ότι ο εχθρός σκοτώθηκε.



Εικόνα 12.3 Αντικείμενα του παιχνιδιού.

2.3 Ανάπτυξη

Η φάση της ανάπτυξης ξεκινάει με την κατασκευή των αντικειμένων που θα εμφανίζονται στο παιχνίδι.

- Κατασκευάζουμε το αντικείμενο παιχνιδιού (`ShootingGame`) ως ένα απλό ορθογώνιο κουτί.
- Το τούβλο το κατασκευάζουμε επίσης ως ορθογώνιο παραλληλεπίπεδο, του δίνουμε διαστάσεις (0.2, 0.5, 0.4) και χρώμα κεραμίδι (κάποια σκούρα απόχρωση του κόκκινου). Είναι απαραίτητο να έχει το όνομα 'brick', για να μπορεί να αναγνωριστεί ο τύπος του αν συγκρουστεί με τον εχθρό.
- Το βλήμα μας είναι μια σφαίρα διαστάσεων (0.3, 0.3, 0.3) και θα πρέπει να έχει όνομα 'ball'. Επιλέγουμε να δώσουμε ελαφριά λάμψη (`shininess: low`) για να δείχνει βαρύ μεταλλικό.
- Το κανόνι είναι ένα σύνθετο αντικείμενο αποτελούμενο από δύο βασικά στερεά: έναν κύλινδρο και μία σφαίρα. Προσαρμόζουμε τον κύλινδρο ώστε να μοιάζει περισσότερο με κανόνι ανοίγοντας οπή (`hollow`) και μικραίνοντας ελαφρώς τις διαστάσεις της κορυφής (`taper`). Στο

τελικό σύνθετο αντικείμενο ρίζα είναι η σφαίρα και ο συνδυασμός έχει γίνει με τέτοιο τρόπο ώστε το αντικείμενο με μηδενική περιστροφή να κοιτάζει προς τα αρνητικά X.

- Το γουρουνάκι μας είναι επίσης σύνθετο και αποτελείται από έξι βασικά στερεά. Μια ελαφρώς παραμορφωμένη σφαίρα για το σώμα, τέσσερις κυλίνδρους για τα πόδια και έναν ακόμη για μύτη. Του δίνουμε χρώμα ανοικτό πράσινο (0.5, 1, 0.5). Το σύνθετο αντικείμενο θα πρέπει να έχει ρίζα τη σφαίρα και ο αρχικός προσανατολισμός του να είναι προς τα θετικά X.

Τα τελικά αντικείμενα που κατασκευάσαμε απεικονίζονται στην εικόνα 12.3.

Το επόμενο βήμα είναι η δημιουργία κώδικα για τα επιμέρους αντικείμενα, ο έλεγχος και η αποσφαλμάτωση.

Ξεκινάμε με το τούβλο. Οι μοναδικές λειτουργίες που θα έχει, σύμφωνα με την ανάλυση που προηγήθηκε, είναι να μπορεί να εξαφανίζεται όταν δέχεται το μήνυμα 'die' και να επανέρχεται στην αρχική του θέση με το μήνυμα 'reset'. Συνεπώς, θα πρέπει όταν δημιουργείται να καταγράφει την αρχική του τοποθέτηση, ώστε να μπορεί να επιστρέψει σε αυτήν. Η δημιουργία του γίνεται μέσω `rez` από το αντικείμενο παιχνιδιού, οπότε η καταγραφή της θέσης θα πρέπει να εκτελείται με το συμβάν `on_rez`. Τέλος, θα πρέπει να αποφασίσουμε ποιο θα είναι το κρυφό κανάλι επικοινωνίας των αντικειμένων. Επιλέγουμε τυχαία το 167 και του δίνουμε την ονομασία `gameChannel` για μεγαλύτερη αναγνωσιμότητα του κώδικα. Ο κώδικάς μας θα είναι:

```
vector initPos;
integer gameChannel = 167;

reset() {
    llSetStatus(STATUS_PHYSICS, FALSE);
    llSetPos(initPos);
    llSetRot(ZERO_ROTATION);
    llSetStatus(STATUS_PHYSICS, TRUE);
}

default {
    state_entry() {
        llListen(gameChannel, "", "", "");
    }

    on_rez(integer param) {
        initPos = llGetPos();
    }

    listen(integer chn, string name, key id, string message) {
        if(message=="reset") reset();
        else if(message=="die") llDie();
    }
}
```

Παρατηρούμε στον παραπάνω κώδικα ότι η συνάρτηση `reset`, εκτός από το να επαναφέρει το τούβλο στην αρχική του θέση, του δίνει και αρχικό προσανατολισμό. Επιπλέον, απενεργοποιεί τη φυσική κατά τη διάρκεια της τοποθέτησης και την ενεργοποιεί ξανά. Ο λόγος για το τελευταίο είναι ότι, αν το αντικείμενο έχει αποκτήσει κάποια ορμή λόγω κρούσης, η ορμή αυτή θα παραμείνει κατά την αλλαγή θέσης, με αποτέλεσμα να είναι πιθανό να μετακινηθεί και να πέσει. Η ενεργοποίηση/απενεργοποίηση της φυσικής γίνεται με την εντολή:

```
llSetStatus (integer status, integer value);
```

Η παραπάνω συνάρτηση μεταβάλλει την κατάσταση ενός αντικειμένου. Αν θέλουμε ένα αντικείμενο να γίνει φυσικό, στην παράμετρο `status` θέτουμε τη μεταβλητή `STATUS_PHYSICS` και στην παράμετρο `value` την τιμή `TRUE`. Αντίστοιχα, με `FALSE` στο `value` το αντικείμενο παύει να έχει φυσική συμπεριφορά.

Δοκιμάζουμε την καλή λειτουργία του αντικειμένου μας ως εξής:

1. παίρνουμε αντίγραφο του στο αποθετήριο μας,
2. δημιουργούμε παραπάνω αντίγραφα στον κόσμο (με drag από το αποθετήριο),
3. τα μετατοπίζουμε ή και περιστρέφουμε και στέλνουμε στο κανάλι 167 την εντολή reset (/167 reset στο chat)
 - θα πρέπει τα αντικείμενα να βρεθούν στην αρχική του θέση,
4. στέλνουμε στο κανάλι 167 την εντολή die
 - θα πρέπει τα αντικείμενα να εξαφανιστούν.

Αν όλα έχουν δουλέψει σωστά, συνεχίζουμε με τον κώδικα του γουρουνιού. Αρχικά το αντικείμενο θα έχει παραπλήσιο κώδικα με το τούβλο. Θα αποθηκεύει την αρχική του θέση, θα ακούει στο ίδιο κανάλι επικοινωνίας και θα μπορεί να κάνει reset και να εξαφανιστεί. Οι επιπλέον σφαιρικές μεταβλητές που θα έχει ο κώδικας του εχθρού μας θα είναι οι εξής:

- η ενέργεια (health),
- η απώλεια ενέργειας κατά τη σύγκρουση με τούβλο (hitByBrick),
- η απώλεια ενέργειας κατά τη σύγκρουση με βλήμα (hitByBall).

Οι δύο τελευταίες μεταβλητές μάς χρειάζονται για να μπορούμε να τροποποιούμε σχετικά εύκολα και γρήγορα στον κώδικα τις παραπάνω τιμές σε περίπτωση που θέλουμε να κάνουμε βελτιώσεις στη λογική του παιχνιδιού.

Μία επιπλέον συνάρτηση που προσθέτουμε στο αντικείμενό μας είναι η updateHealth. Η συνάρτηση αυτή τροποποιεί την «εμφάνιση» του γουρουνιού ανάλογα με την κατάσταση της υγείας του. Αν η υγεία είναι πάνω από το μηδέν, την εμφανίζει ως κείμενο και προσαρμόζει το χρώμα του σύμφωνα με τη λογική που έχουμε αναφέρει πιο πάνω. Αν πέσει κάτω από το μηδέν, το αντικείμενο παύει να έχει φυσική συμπεριφορά και «αναποδογυρίζει». Επιπλέον, στέλνει ένα μήνυμα στο κανάλι του παιχνιδιού ότι πέθανε.

Τέλος, προσθέτουμε το συμβάν αναγνώρισης σύγκρουσης (collision_start). Εκεί διατρέχουμε όλα τα αντικείμενα που συνέβαλλαν στο συμβάν (num_detected) και ελέγχουμε τα ονόματά τους. Αν εντοπίσουμε 'brick' ή 'ball', αφαιρούμε τους αντίστοιχους πόντους και ενημερώνουμε την updateHealth για να εμφανιστεί οπτικά η νέα κατάσταση του γουρουνιού. Αν το γουρουνάκι είναι ήδη νεκρό (health < 0), δεν κάνουμε καθόλου έλεγχο.

Ο τελικός μας κώδικας είναι:

```
integer health;
integer hitByBrick = 23;
integer hitByBall = 40;
integer gameChannel = 167;
vector initPos;

updateHealth() {
    if(health<0) {
        llSetText("dead",<1.0,1.0,1.0>,1.0);
        llSetLinkColor(LINK_SET,<0.5,0.5,0.5>,ALL_SIDES);
        llSetRot(llEuler2Rot(<180*DEG_TO_RAD,0,0>));
        llSetStatus(STATUS_PHYSICS,FALSE);
        llSay(gameChannel,"died");
    }
    else {
        llSetText("health: "+(string)health,<1.0,1.0,1.0>,1.0);
        llSetLinkColor(LINK_SET,<0.5,0.5+health*0.005,0.5>,
            ALL_SIDES);
    }
}

reset() {
    llSetPos(initPos);
}
```

```

    llSetRot(ZERO_ROTATION);
    health = 100;
    updateHealth();
    llsetStatus(STATUS_PHYSICS, TRUE);
}

default {
    state_entry() {
        llListen(gameChannel, "", "", "");
    }

    on_rez(integer param) {
        initPos = llGetPos();
        reset();
    }

    listen(integer chn, string name, key id, string message) {
        if(message=="reset") reset();
        else if(message=="die") llDie();
    }

    collision_start(integer num) {
        if(health < 0) return;
        integer i;
        string name;
        for(i=0;i<num;i++) {
            name = llDetectedName(i);
            if(name=="brick") {
                health -= hitByBrick;
                updateHealth();
            }
            else if(name=="ball") {
                health -= hitByBall;
                updateHealth();
            }
        }
    }
}

```

Παρατηρούμε ότι για την αλλαγή του χρώματος δεν χρησιμοποιήσαμε την `llSetColor`, αλλά την `llSetLinkColor`. Με τη συνάρτηση αυτήν μπορούμε να αλλάξουμε συνολικά το χρώμα μιας ομάδας αντικειμένων και όχι μόνο του αντικειμένου στο οποίο τρέχει το script.

Για την αποσφαλμάτωση του γουρουνιού που κατασκευάσαμε θα πρέπει και πάλι να πάρουμε ένα αντίγραφο στο αποθετήριο και να εισάγουμε μερικά αντικείμενα στον κόσμο. Στη συνέχεια, εκτός από τη σωστή λειτουργία του `die` και `reset` που είδαμε πιο πάνω, δοκιμάζουμε και την αλληλεπίδρασή τους με το τούβλο και την μπάλα κανονιού. Τα δεύτερα θα πρέπει να είναι φυσικά αντικείμενα. Τα τοποθετούμε πάνω από ένα γουρουνάκι και τα αφήνουμε να πέσουν με τη βαρύτητα σε αυτό. Θα πρέπει να μειωθεί η υγεία τους και να αλλάξει αντίστοιχα το χρώμα τους.

Το επόμενο αντικείμενό μας είναι το κανόνι. Και σε αυτό υλοποιούμε τις διαδικασίες `reset` και `die` όπως στα προηγούμενα. Επιπλέον, το κανόνι μας θα πρέπει να αποθηκεύει τρεις μεταβλητές οι οποίες καθορίζουν τη βολή του:

- την οριζόντια απόσταση (offset),
- τη γωνία (angle) και
- την ένταση (strength).

Κατασκευάζουμε μια διαδικασία fire για την εκτέλεση της βολής σύμφωνα με τις παραπάνω τιμές. Ο κώδικας θα κάνει αρχικά μετατόπιση σύμφωνα με το offset, στη συνέχεια περιστροφή σύμφωνα με τη γωνία και τέλος δημιουργία ενός νέου αντικειμένου τύπου ball με κατάλληλη τοποθέτηση και ταχύτητα. Θα πρέπει το αντικείμενο ball να υπάρχει στα περιεχόμενα του κανονιού ώστε να μπορεί να το εισάγει στον κόσμο. Μετά την εκτόξευση, το κανόνι επιστρέφει στην αρχική του θέση. Ανάμεσα σε καθεμία από τις παραπάνω ενέργειες προσθέτουμε κατάλληλη παύση, ώστε να προλαβαίνει ο χρήστης να παρατηρήσει τη διαδικασία. Επιπλέον, στέλνουμε το μήνυμα 'shot' στο κανάλι παιχνιδιού.

Η εισαγωγή των τιμών από τον χρήστη θα γίνεται αφότου κάνει κλικ πάνω στο κανόνι. Στην περίπτωση αυτήν, το κανόνι θα αλλάζει κατάσταση, θα διαβάζει τις τιμές, θα εκτελεί τη βολή και θα επιστρέφει στην αρχική κατάσταση. Για τον σκοπό αυτόν δημιουργούμε μια νέα κατάσταση, τη userInput. Στην κατάσταση αυτήν, όταν ληφθεί μήνυμα μετατρέπεται σε λίστα χρησιμοποιώντας το κενό ή το κόμμα για διαχωριστή. Τα τρία πρώτα στοιχεία ερμηνεύονται ως τιμές offset, angle και strength.

Ο κώδικας του κανονιού είναι ο παρακάτω:

```
float offset;
float angle;
float strength;
integer gameChannel = 167;
vector initPos;

fire() {
    vector pos = llGetPos();
    llSetPos(pos+<0, offset*2.0, 0>);
    llSleep(1.0);
    llSetRot(llEuler2Rot(<0, angle*DEG_TO_RAD, 0>));
    llSleep(1.0);
    llRezObject("ball", llGetPos()+<-0.5, 0, 0>*llGetRot(),
        <-10.0*strength, 0, 0>*llGetRot(), ZERO_ROTATION, 0);
    llSay(gameChannel, "shot");
    llSleep(2.0);
    llSetPos(pos);
}

reset() {
    llSetPos(initPos);
}

default {
    state_entry() {
        llListen(gameChannel, "", "", "");
    }

    on_rez(integer param) {
        initPos = llGetPos();
    }

    listen(integer chn, string name, key id, string message) {
        if(message=="reset") reset();
        else if(message=="die") llDie();
    }

    touch_start(integer num) {
        llSay(0, "Enter offset, angle, strength");
        state userInput;
    }
}
```

```

    }
}

state userInput {
    state_entry() {
        llListen(0, "", "", "");
    }

    listen(integer chn, string name, key id, string message) {
        list values = llParseString2List(message, [" ", ",", ""], []);
        offset = llList2Float(values, 0);
        angle = llList2Float(values, 1);
        strength = llList2Float(values, 2);

        fire();
        state default;
    }
}
}

```

Στον παραπάνω κώδικα δεν γίνεται έλεγχος αν οι τρεις τιμές που δίνει ο χρήστης είναι στο σωστό εύρος. Προσθέστε το ως άσκηση. Δοκιμάζουμε στη συνέχεια να κάνουμε το κανόνι μας να εκτελέσει ορισμένες βολές ώστε να επαληθεύσουμε την καλή λειτουργία του.

Τέλος, κατασκευάζουμε το αντικείμενο παιχνιδιού. Αρχικά θα πρέπει να μεταφέρουμε στα περιεχόμενά του τα τρία αντικείμενα που κατασκευάσαμε, δηλαδή το κανόνι (cannon), το γουρουνάκι (piggy) και το τούβλο (brick). Στη συνέχεια δημιουργούμε τη διαδικασία κατασκευής μιας νέας πίστας. Τοποθετούμε τον τοίχο από τούβλα, τα γουρουνάκια και το κανόνι σύμφωνα με τη λογική που έχουμε περιγράψει στην τεχνική ανάλυση. Ο κώδικας είναι ο παρακάτω:

```

newStage() {
    reset();
    //build wall
    integer row;
    integer col;
    float add;

    for(row=0;row<4;row++) {
        if(row%2==0) add=0;
        else add = 0.2;
        for(col=0;col<5;col++) {
            vector pos = <startPos.x,
                startPos.y+2.0+col*0.55+add,
                startPos.z+row*0.4>;
            llRezObject("brick",pos,<0,0,0>,ZERO_ROTATION,0);
        }
    }

    //place pigs
    integer numOfPigs = 3+(integer)llFrand(3.0);
    integer i;
    for(i=0;i<numOfPigs;i++) {
        llRezObject("piggy", <startPos.x - 2.0 - llFrand(3.0),
            startPos.y + 2.0 + llFrand(2.5), startPos.z>, <0,0,0>,
            ZERO_ROTATION, 0);
    }
    //place cannon
}

```



```

    llRezObject("cannon", <startPos.x+ 4.0 + llFrand(3.0),
        startPos.y + 3.2, startPos.z>,
        <0,0,0>, ZERO_ROTATION, 0);
}

```

Όπως παρατηρούμε, η τοποθέτηση όλων των στοιχείων γίνεται βάσει μιας μεταβλητής `startPos`, η οποία περιγράφει τη θέση του αντικειμένου παιχνιδιού. Στην τοποθέτηση των τούβλων χρησιμοποιείται μία ακόμη μεταβλητή, η `add`, η οποία ανάλογα με τη γραμμή γίνεται 0 ή 0.2 ώστε να δίνει διαφοροποίηση στη διάταξη των τούβλων σε διαδοχικές γραμμές. Στο σημείο αυτό δοκιμάζουμε να εκτελέσουμε τη συνάρτηση και να βεβαιωθούμε ότι η πίστα μας στήνεται σωστά.

Ο υπόλοιπος κώδικας του αντικειμένου παιχνιδιού εμφανίζει το μενού με τις επιλογές του χρήστη, στέλνει μηνύματα στα υπόλοιπα αντικείμενα όταν χρειάζεται (`reset` και `die`) και ενημερώνει το σκορ όταν λάβει μήνυμα (`shot` και `died`). Ο κώδικας είναι ο παρακάτω:

```

integer shots;
integer kills;
integer gameChannel = 167;
vector startPos;

updateStatus() {
    llSetText("shots: "+(string)shots+", kills: "
        +(string)kills, <1.0,1.0,1.0>,1.0);
}

reset() {
    shots = 0;
    kills = 0;
    updateStatus();
}

newStage() {
    .. // ο κώδικας που δείξαμε πιο πάνω
}

default {
    state_entry() {
        llListen(gameChannel, "", "", "");
        startPos = llGetPos();
    }

    touch_start(integer num) {
        llDialog(llDetectedKey(0), "Please select", ["reset",
            "new stage", "end game"], gameChannel);
    }

    listen(integer chn, string name, key id, string message) {
        if(message=="reset") reset();
        else if(message=="new stage") {
            llShout(gameChannel, "die");
            newStage();
        }
        else if(message=="end game") {
            llShout(gameChannel, "die");
        }
        else if(message=="shot") {

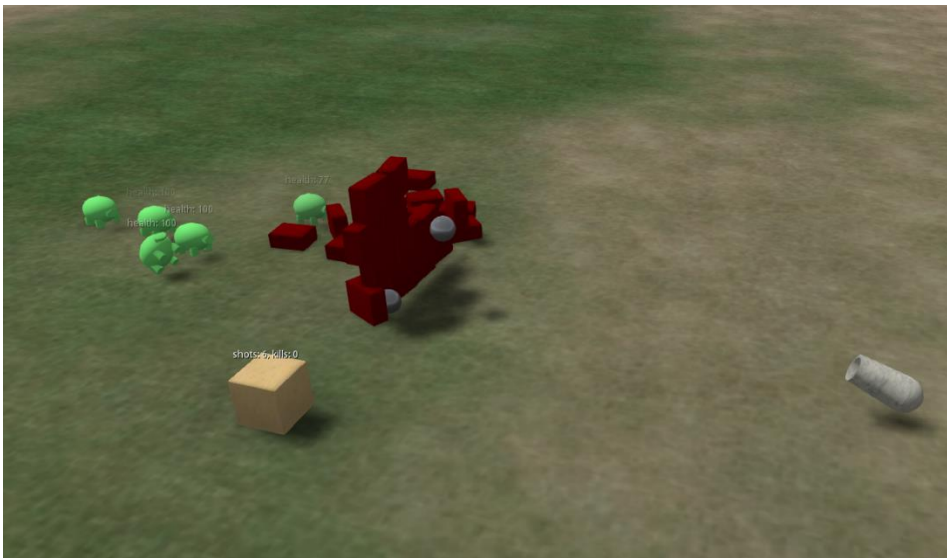
```

```

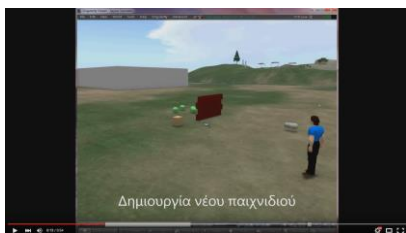
shots++;
updateStatus();
}
else if(message=="died") {
kills++;
updateStatus();
}
}
}

```

Τέλος, δοκιμάζουμε να τρέξουμε συνολικά το παιχνίδι και κάνουμε όποιες διορθώσεις κρίνουμε απαραίτητες για να βελτιωθεί το παίξιμο. Για παράδειγμα, μπορούμε να αλλάξουμε τον τρόπο με τον οποίο στήνεται η πίστα, τη μέγιστη ένταση εκτόξευσης, το ποσοστό υγείας που χάνεται κατά την κρούση κ.ά. Η εκτέλεση του παιχνιδιού παρουσιάζεται στο βίντεο 12.1 και ένα στιγμιότυπο απεικονίζεται στην εικόνα 12.4.



Εικόνα 12.4 Στιγμιότυπο από την εκτέλεση του παιχνιδιού.



Σύνδεσμος:
<https://youtu.be/JQ7w4Jv0H8c>

Video 12.1 Εκτέλεση του παιχνιδιού βολών.

2.4 Επεκτάσεις

Υπάρχουν αρκετές ενδιαφέρουσες βελτιώσεις και προσθήκες που μπορείτε να κάνετε στο παιχνίδι που κατασκευάσαμε. Αρχικά, μπορείτε να ελέγχετε τις τιμές που δίνει ο χρήστης για τη βολή και αν είναι εκτός ορίων να του στέλνετε σχετικό μήνυμα. Στη συνέχεια, μπορείτε να δοκιμάσετε να κατασκευάσετε πιο σύνθετες δομές με τα τούβλα, π.χ. να συνδυάσετε μικρούς οριζόντιους και κάθετους τοίχους. Επιπλέον, μπορείτε να δημιουργήσετε παραπάνω από ένα είδη τούβλων και γουρουνιών. Τα πρώτα θα μπορούσαν, για παράδειγμα, να έχουν διαφορετικά μεγέθη ή και διαφορετική επίδραση στην ενέργεια του εχθρού και τα δεύτερα να έχουν μεγαλύτερα ή μικρότερα αρχικά επίπεδα ενέργειας. Αλλάζοντας και επεκτείνοντας τον αλγόριθμο κατασκευής της πίστας, θα μπορούσατε έτσι να έχετε πιο σύνθετους κόσμους παιχνιδιού που να θυμίζουν περισσότερο το δημοφιλές Angry Birds. Τέλος, θα μπορούσατε να αλλάξετε τη διεπαφή σε πλήκτρα που πατιούνται στον κόσμο

ή στο HUD. Με βάση τις τιμές που δέχεται αυτήν την στιγμή το κανόνι, θα έπρεπε να υπάρχουν επτά πλήκτρα (κίνηση δεξιά—αριστερά, περιστροφή πάνω—κάτω, αύξηση—μείωση έντασης και πυρ).

Εκτός από τα παραπάνω, που είναι σχετικά εύκολα και θα μπορούσατε να δοκιμάσετε ως άσκηση, μπορούμε να δούμε και άλλες, αρκετά πιο σύνθετες επεκτάσεις του παιχνιδιού.

1. Θα μπορούσαν να παίζουν δύο παίκτες ανταγωνιστικά, όπου καθένας θα έχει τις δικιές του μονάδες και στόχος του είναι να χτυπήσει τις μονάδες του αντιπάλου.
2. Αντίστοιχα, θα μπορούσαμε να προβλέψουμε και συνθήκες συνεργασίας στο παιχνίδι, π.χ. ο ένας παίκτης να ορίζει τη μετακίνηση και τη γωνία και ο άλλος την ένταση και τούμπαλιν.
3. Θα μπορούσαν οι πίστες να δημιουργούνται βάσει κάποιου αρχείου (notecard) που περιγράφει τη λογική τους, π.χ. πώς στήνονται τα τούβλα και σε ποια όρια τοποθετούνται το κανόνι και οι εχθροί. Έτσι, το παιχνίδι θα μπορούσε να επεκτείνεται διαρκώς με νέες πίστες.
4. Θα μπορούσε το παιχνίδι να χρησιμοποιηθεί ως βάση σε εφαρμογή εκπαιδευτικής ψυχαγωγίας (edutainment) για παιδιά με στόχο την εκμάθηση βασικών αρχών κινηματικής. Π.χ. να παρακινεί το παιδί να μάθει μέσω στοχευμένων προβλημάτων την επίδραση της γωνίας και της ταχύτητας στην πορεία της βολής.

3 Αίθουσα διαλέξεων και συνεργασίας

Στο δεύτερο παράδειγμά μας θα ασχοληθούμε με την κατασκευή ενός περιβάλλοντος σύγχρονης συνεργασίας από απόσταση. Έχουμε ήδη τονίσει σε διάφορα σημεία αυτού του βιβλίου τη δυνατότητα των εικονικών κόσμων να προκαλέσουν το αίσθημα της συν-παρουσίας, δηλαδή να νιώθει ο επισκέπτης ότι βρίσκεται σε έναν κοινό τόπο μαζί με άλλους χρήστες και να επικοινωνεί μαζί τους με πολλαπλά μέσα. Όμως το χαρακτηριστικό αυτό, αν και απαραίτητο, δεν αρκεί από μόνο του για την υποστήριξη συνεργατικών δραστηριοτήτων. Είναι σημαντικό να υπάρχει κατάλληλα διαμορφωμένος χώρος και εργαλεία για τη διεξαγωγή των επιθυμητών διεργασιών μεταξύ ομάδων συζητήσεων και συνεργασίας. Για παράδειγμα, σε έναν φυσικό χώρο παρουσιάσεων υπάρχει μια συγκεκριμένη διάταξη των καθισμάτων, ώστε να μπορούν να εστιάσουν όλοι στον παρουσιαστή, υπάρχουν συμπληρωματικά εργαλεία, όπως συσκευή παρουσιάσεων (π.χ. προβολέας) και χώρος σημειώσεων (π.χ. πίνακας). Σε έναν χώρο συναντήσεων εργασίας η διάταξη των καθισμάτων μπορεί να είναι αρκετά διαφορετική, διότι γίνονται συζητήσεις μεταξύ όλων των συμμετεχόντων. Και στον χώρο αυτόν μπορεί να υπάρχει η ανάγκη για προβολή και για σημειώσεις. Σε ομάδες με μεγάλο αριθμό συμμετεχόντων μπορεί να υπάρχει επιπλέον συσκευή ψηφοφοριών για την υποστήριξη συλλογικών αποφάσεων. Πολλές από τις παραπάνω λειτουργίες είναι επιθυμητό να μπορούμε να τις προσφέρουμε εντός του εικονικού κόσμου, ώστε να μπορούν οι επισκέπτες να οργανώσουν τις προγραμματισμένες συζητήσεις και συνεργατικές δραστηριότητές τους.

Στόχος του παραδείγματός μας είναι η κατασκευή ενός χώρου συζητήσεων πλαισιωμένου με έναν αριθμό από υποστηρικτικά εργαλεία για την υποστήριξη σύγχρονων συνεργασιών. Θα προσπαθήσουμε η λύση μας να είναι όσο το δυνατόν πιο γενική, δηλαδή ο χώρος μας να μπορεί με την κατάλληλη διαμόρφωση να υποστηρίξει πολλαπλών ειδών δραστηριότητες, όπως η διδασκαλία ενός μαθήματος, διεξαγωγή συνεδρίων και ημερίδων, συναντήσεις ομάδων εργασίας κ.ά. Για την ανάπτυξη ορισμένων από τα συνεργατικά εργαλεία που θα περιλαμβάνει ο χώρος μας θα αξιοποιήσουμε τη δυνατότητα του OpenSimulator να προβάλλει ιστοσελίδες πάνω σε επιφάνειες αντικειμένων καθώς και τις online εφαρμογές γραφείου που προσφέρει η Google.

3.1 Σχεδίαση περιβάλλοντος και διεπαφής

Το πρώτο μας βήμα πριν τη σχεδίαση του περιβάλλοντος είναι ο προσδιορισμός των αναγκών των χρηστών. Παρατηρώντας τη διαμόρφωση, τις δραστηριότητες και τα εργαλεία των φυσικών χώρων συνεργασίας, διαπιστώνουμε τις πιο κάτω βασικές ανάγκες:

1. Η διάταξη των καθισμάτων αλλάζει ανάλογα με το είδος της συνεργασίας η οποία πρόκειται να υποστηριχθεί.
2. Απαιτείται μια συσκευή παρουσιάσεων μέσω της οποίας θα μπορεί ένας χρήστης να ανεβάζει την παρουσίασή του και να ελέγχει τη ροή της.

3. Υπάρχει ανάγκη για κάποιον κοινό χώρο σημειώσεων στον οποίο θα μπορεί ο κάθε χρήστης να προσθέσει το δικό του κείμενο.
4. Επιθυμητή ανάγκη είναι και η υποστήριξη κρυφών ή φανερών ψηφοφοριών μεταξύ των χρηστών για τη λήψη συλλογικών αποφάσεων.

Είναι προφανές ότι μπορεί να εντοπίσει κάποιος αρκετές ακόμη ανάγκες υποστήριξης της συνεργασίας, όμως για λόγους απλότητας θα παραμείνουμε σε αυτές, τις οποίες θεωρούμε πιο βασικές.

Στη συνέχεια αναζητούμε λύσεις για τον μετασχηματισμό των παραπάνω αναγκών σε αντικείμενα και εργαλεία του εικονικού κόσμου.

Η ανάγκη για διαφορετικές διατάξεις καθισμάτων θα μπορούσε να οδηγήσει στην κατασκευή πολλαπλών διαφορετικών χώρων συνεργασίας. Η λύση αυτή όμως θα απαιτούσε αφενός την επέκταση των δωματίων αυτών σε πολύ μεγαλύτερο τμήμα του εικονικού κόσμου και αφετέρου δεν θα μπορούσε να υποστηρίξει τυχόν νέες απαιτήσεις διάταξης που θα προέκυπταν στην πορεία. Με δεδομένο ότι τα αντικείμενά μας μπορούν να αλλάζουν θέση δυναμικά, είναι δυνατόν να έχουμε πολλές διαφορετικές διατάξεις των ίδιων αντικειμένων και στον ίδιο χώρο δωματίου κατόπιν επιλογής των χρηστών. Έτσι, θα μπορεί το ίδιο δωμάτιο συνεργασίας σε μία περίπτωση να μετατρέπεται σε τάξη και σε μια άλλη σε στρογγυλό τραπέζι για ομαδικές συζητήσεις. Επιλέγουμε λοιπόν να προσθέσουμε στον χώρο μας ένα ειδικό εργαλείο επιλογής διάταξης.

Η συσκευή παρουσιάσεων θα πρέπει να είναι όσο το δυνατόν πιο γενική και εύκολη στη χρήση. Ιδανικά θέλουμε να μεταφέρουμε τη λειτουργικότητα των δημοφιλών εργαλείων παρουσίασης, όπως το powerpoint, στο εικονικό περιβάλλον. Επομένως, η συσκευή μας θα πρέπει κατ' ελάχιστο να επιτρέπει στους χρήστες να ανεβάσουν μια έτοιμη παρουσίαση, να την εμφανίζει στον κόσμο και να μπορεί κάποιος χρήστης να ελέγξει τη ροή με πλήκτρα επόμενης και προηγούμενης διαφάνειας.

Αντίστοιχα, ο κοινός χώρος σημειώσεων θα πρέπει να είναι ορατός και προσβάσιμος από όλους. Θα μπορούσε να επιτελεί τη λειτουργία του πίνακα σε μια ψηφιακή τάξη, να χρησιμεύει ως χώρος λήψης πρακτικών μιας συνεδρίασης ή ως χώρος καταγραφής ιδεών και προτάσεων σε μια ομάδα συνεργασίας. Οι χρήστες θα πρέπει να είναι τουλάχιστον σε θέση να γράψουν απευθείας κείμενο πάνω σε αυτόν.

Τέλος, για τις ψηφοφορίες θα μπορούσε να κατασκευαστεί ένα εργαλείο αντίστοιχο με αυτά που συναντάει κάποιος σε χώρους συνεδριάσεων. Δηλαδή να προσφέρεται στον χρήστη ένας αριθμός από επιλογές στη μορφή διαφορετικών πλήκτρων και όταν ξεκινάει η ψηφοφορία ο χρήστης να κάνει την επιλογή του μέσω των πλήκτρων αυτών. Με τη λήξη της ψηφοφορίας η συσκευή θα ανακοινώνει τα αποτελέσματα.

Εκτός από τα παραπάνω, μπορούμε να εντοπίσουμε μία ακόμη ανάγκη στον εικονικό κόσμο για την καλύτερη διευθέτηση των συζητήσεων: την ανάταση χεριός. Πράγματι, σε φυσικούς χώρους συναντήσεων συνηθίζεται να οργανώνεται η συζήτηση και να δίνεται ο λόγος στους ομιλητές μέσω ανάτασης του χεριού. Στο OpenSimulator όμως κάτι τέτοιο δεν είναι εύκολο. Θα είναι χρήσιμο λοιπόν να κατασκευαστεί ένα πλήκτρο το οποίο θα κάνει την ενσάρκωσή μας να σηκώνει το χέρι της, ώστε να μπορούν να αντιληφθούν οι υπόλοιποι χρήστες ότι θέλουμε να ζητήσουμε τον λόγο.

3.2 Τεχνική σχεδίαση

Η προσθήκη στα αντικείμενα της δυνατότητας πολλαπλών διατάξεων είναι σχετικά απλή με τις γνώσεις που έχουμε αποκτήσει στο OpenSimulator. Όταν θα δέχονται κάποιο μήνυμα σε ειδικό κανάλι σχετικά με τη διάταξη που πρέπει να αποκτήσουν, θα αλλάζουν σε κάποια προαποθηκευμένη θέση και προσανατολισμό μέσω των `IISetPos` και `IISetRot`. Όμως η χειροκίνητη εισαγωγή των τιμών θέσης και προσανατολισμού σε καθένα από τα αντικείμενα μπορεί να γίνει επίπονη. Για τον λόγο αυτόν θα κατασκευάσουμε αρχικά ένα πρόγραμμα μέσω του οποίου θα καταχωρούν αυτόματα τις θέσεις τους στις διάφορες διατάξεις και θα τις αποθηκεύουν σε λίστα. Οι προαποθηκευμένες θέσεις θα διαβάζονται από αυτήν τη λίστα όταν απαιτείται.

Η συσκευή παρουσιάσεων και ο χώρος σημειώσεων θα βασιστούν στην αναπαράσταση ιστοσελίδων πάνω σε επιφάνειες αντικειμένων. Πράγματι, στο OpenSimulator έχουμε τη δυνατότητα να επιλέξουμε σε μία ή περισσότερες επιφάνειες να γίνεται «ζωντανή» προβολή κάποιας ιστοσελίδας. Αυτό που συμβαίνει στην πράξη είναι ότι ανοίγει ένας «ελαφρύς» Web browser και αναπαριστά το περιεχόμενο της σελίδας πάνω στην επιφάνεια του αντικειμένου. Για να επιτύχουμε κάτι τέτοιο, αρκεί να ακολουθήσουμε τα παρακάτω βήματα:

- σε κατάσταση επεξεργασίας επιλέγουμε την επιφάνεια που μας ενδιαφέρει,
- στην καρτέλα Texture αλλάζουμε την επιλογή Materials σε Media και

- πατάμε το πλήκτρο 'Choose' και στο νέο παράθυρο δίνουμε το όνομα της σελίδας που θέλουμε να δούμε.

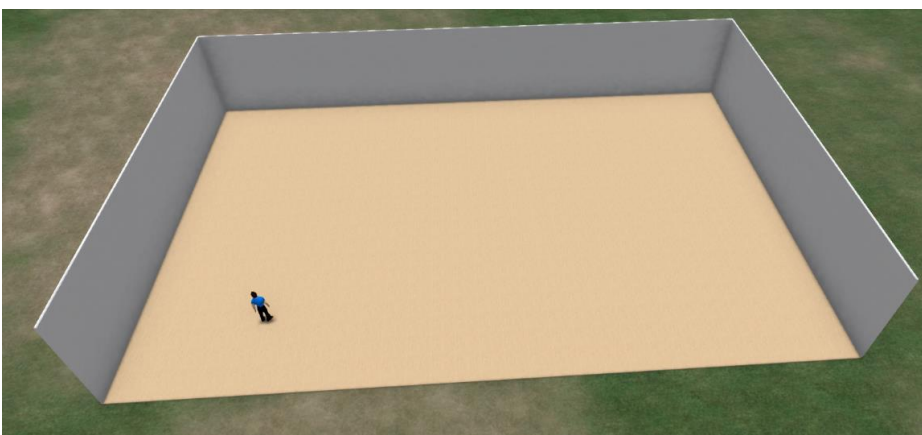
Αν βγούμε από την κατάσταση επεξεργασίας και κάνουμε κλικ πάνω στην επιφάνεια, θα δούμε, όπως φαίνεται και στην εικόνα 12.5, την ιστοσελίδα να εμφανίζεται! Επιπλέον, μπορούμε να πλοηγηθούμε κανονικά πατώντας πάνω σε υπερσυνδέσμους και μετακινώντας τις μπάρες κύλισης (scroll bars). Για να φαίνεται πιο φωτεινό το περιεχόμενο της σελίδας μας, μπορούμε επιπλέον να επιλέξουμε από την καρτέλα Texture την επιλογή full bright.



Εικόνα 12.5 Προβολή ιστοσελίδας σε επιφάνεια του περιβάλλοντος.

Εκμεταλλευόμενοι το παραπάνω χαρακτηριστικό, μπορούμε να προσφέρουμε διαμοιρασμένα κείμενα και παρουσιάσεις μέσω των Google docs, δηλαδή των online εφαρμογών γραφείου της Google. Αν εμφανίσουμε στην επιφάνειά μας ένα αρχείο κειμένου το οποίο έχουν όλοι οι χρήστες δικαίωμα να δουν και να επεξεργαστούν, τότε θα μπορούν όλοι μέσω του mini-browser να διαβάσουν τα περιεχόμενά του και να γράψουν σε αυτό. Το ίδιο ισχύει και με αρχεία παρουσιάσεων. Απλά στη δεύτερη περίπτωση θα δημιουργήσουμε τα κατάλληλα πλήκτρα αλλαγής διαφάνειας, ώστε να μπορούν οι χρήστες να ελέγχουν τη ροή της παρουσίασης.

Οι ψηφοφορίες και η ανάταση χεριού θα υποστηρίζονται μέσω HUD. Για τις πρώτες, θα υπάρχει ειδικό αντικείμενο το οποίο θα καταγράφει τις αποκρίσεις των χρηστών ελέγχοντας ότι δεν έχει διπλοψηφίσει κάποιος χρήστης. Με τη λήξη της ψηφοφορίας θα ανακοινώνει τα αποτελέσματα. Για το δεύτερο, θα κατασκευάσουμε ειδική συνθετική κίνηση στο QAnimator και θα την αποδίδουμε στο avatar του χρήστη όταν πατάει το σχετικό πλήκτρο.



Εικόνα 12.6 Αρχική μορφή του δωματίου συνεργασίας.

3.3 Ανάπτυξη

Το πρώτο μας βήμα είναι η σχεδίαση του χώρου. Επιλέγουμε να κατασκευάσουμε ένα δωμάτιο μεγάλων διαστάσεων για να διευκολύνεται η κίνηση και η οπτική. Επιπλέον, αφήνουμε αρχικά την μπροστινή και πάνω όψη ελεύθερες, ώστε να μπορούμε να κάνουμε εύκολα και γρήγορα τροποποιήσεις στις διαστάσεις, αν χρειαστεί. Το δωμάτιο που κατασκευάσαμε φαίνεται στην εικόνα 12.6.

Στη συνέχεια κατασκευάζουμε τα καθίσματα και τα τραπέζια που θα χρησιμοποιήσουμε στον χώρο μας. Δημιουργούμε ένα αντικείμενο με μορφή καρτέκλας, στο οποίο έχουμε διασφαλίσει μέσω της `llSitTarget` ότι οι ενσαρκώσεις κάθονται σωστά, και ένα αντικείμενο τραπέζιού. Θα προσθέσουμε σε αυτά τα αντικείμενα κώδικα ο οποίος μπορεί να αποθηκεύει και να ανακαλεί θέσεις. Ο κώδικας είναι ο παρακάτω:

```
list positions;
integer posChannel = 187;

default {
    state_entry() {
        llListen(posChannel, "", "", "");
    }

    listen(integer chn, string name, key id, string message) {
        list msgParts = llParseString2List(message, " ", "");
        if(llList2String(msgParts, 0) == "record") {
            string posName = llList2String(msgParts, 1);
            positions += [posName, llGetPos(), llGetRot()];
        }
        else if(llList2String(msgParts, 0) == "setPos") {
            string posName = llList2String(msgParts, 1);
            integer index = llListFindList(positions, [posName]);
            llSetPos(llList2Vector(positions, index+1));
            llSetRot(llList2Rot(positions, index+2));
        }
    }
}
```

Όπως μπορείτε να παρατηρήσετε, χρησιμοποιούμε μια λίστα, την `positions`, για να αποθηκεύουμε τριάδες της εξής μορφής: όνομα, θέση, προσανατολισμός. Κάθε φορά που το αντικείμενο δέχεται μήνυμα που ξεκινάει από `record`, ορίζει ως όνομα διάταξης την επόμενη λέξη (π.χ. στο μήνυμα `record round_table`, η διάταξη είναι `round_table`) και αποθηκεύει στη λίστα μαζί με το όνομα την τρέχουσα θέση και τον προσανατολισμό. Όταν δεχτεί μήνυμα `setPos`, χρησιμοποιεί τη λέξη που ακολουθεί ως όνομα θέσης και προσπαθεί να την εντοπίσει στη λίστα `positions`. Θέτει τις επόμενες δύο τιμές ως νέα θέση και προσανατολισμό.



Εικόνα 12.7 Διάταξη «στρογγυλό τραπέζιού» (αριστερά) και διάταξη τάξης (δεξιά).

Για λόγους ελέγχου και αποσφαλμάτωσης κάνουμε δοκιμές με ένα αντικείμενο (π.χ. την καρέκλα) και βεβαιωνόμαστε ότι λειτουργεί σωστά. Κάνουμε record σε δύο ή περισσότερες διαφορετικές διατάξεις και ζητάμε από το αντικείμενο να επιστρέψει στις θέσεις αυτές.

Όταν βεβαιωθούμε ότι όλα λειτουργούν όπως πρέπει, αντιγράφουμε τον κώδικα και στο άλλο αντικείμενο και παίρνουμε αντίγραφα τους στο αποθετήριο μας, π.χ. chair και table. Στη συνέχεια, δημιουργούμε δέκα καρέκλες και οκτώ τραπέζια στον κόσμο. Τα τοποθετούμε στις επιθυμητές διατάξεις, π.χ. class και round_table, και τα αποθηκεύουμε. Οι διατάξεις που δημιουργήσαμε απεικονίζονται στην εικόνα 12.7.

Το τελικό βήμα είναι να κατασκευάσουμε ένα αντικείμενο το οποίο μας επιτρέπει να αλλάξουμε, μέσω μενού, τη διάταξη του χώρου. Τοποθετούμε το αντικείμενο σε κάποιον τοίχο δίνοντάς του ευδιάκριτο σχήμα και του προσθέτουμε τον παρακάτω κώδικα:

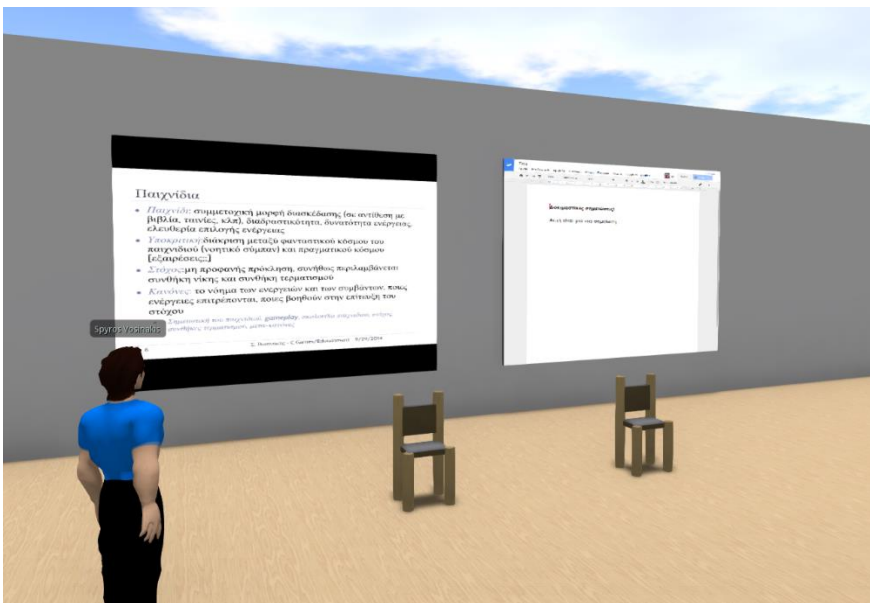
```
list positions = ["class", "round_table"];
integer posChannel = 187;

default {
    state_entry() {
        llListen(111, "", "", "");
    }

    touch_start(integer num) {
        llDialog(llDetectedKey(0), "Select arrangement",
            positions, 111);
    }

    listen(integer chn, string name, key id, string message) {
        llShout(posChannel, "setPos "+message);
    }
}
```

Ο κώδικάς μας απλά στέλνει την επιλογή διάταξης του χρήστη ως μήνυμα στο κανάλι που ακούνε οι καρέκλες και τα τραπέζια, ώστε να αλλάξει αντίστοιχα η διάταξή τους.



Εικόνα 12.8 Προβολή παρουσιάσεων και χώρος σημειώσεων στον χώρο συνεργασίας.

Στη συνέχεια θα δημιουργήσουμε τον χώρο σημειώσεων και τον χώρο παρουσιάσεων. Κατασκευάζουμε δύο μεγάλες και λεπτές επιφάνειες και τις τοποθετούμε σε έναν από τους τοίχους του δωματίου. Δημιουργούμε ένα

νέο έγγραφο κειμένου στο Google Docs και επιλέγουμε “Get shareable link” επιτρέποντας στους επισκέπτες να το επεξεργαστούν. Αντιγράφουμε τον σύνδεσμο στα περιεχόμενα (Media) της μπροστινής όψης του αντικειμένου που θα λειτουργήσει ως χώρος σημειώσεων. Αντίστοιχα, δημιουργούμε μια παρουσίαση στο Google Slides και κατασκευάζουμε ένα δεύτερο αντικείμενο στον κόσμο μας στο οποίο μεταφέρουμε την παρουσίαση. Το αποτέλεσμα φαίνεται στην εικόνα 12.8.

Ο χώρος σημειώσεων δεν χρειάζεται περαιτέρω τροποποιήσεις. Όλοι οι χρήστες μπορούν διαβάσουν το κείμενο, να κάνουν κλικ πάνω του και να το επεξεργαστούν.

Αντίθετα, ο χώρος παρουσιάσεων χρειάζεται να βελτιωθεί. Αν δοκιμάσουμε να αλλάξουμε slides μέσω των πλήκτρων της Google και υπάρχουν άλλοι χρήστες στον χώρο, θα παρατηρήσουμε ότι δεν μπορούν να αντιληφθούν την αλλαγή! Ο λόγος είναι ότι δεν αλλάζει το URL, επομένως δεν ενημερώνονται οι browsers των υπόλοιπων χρηστών ότι άλλαξε η διαφάνεια.

Από την άλλη μεριά, μπορούμε να μεταφερθούμε «χειροκίνητα» σε μια οποιαδήποτε διαφάνεια της παρουσίασης, αν στο τέλος του URL η παράμετρος slide έχει τον αριθμό της διαφάνειας, π.χ. slide=1. Θα εκμεταλλευτούμε το χαρακτηριστικό αυτό, ώστε να αλλάζουμε τη διεύθυνση μέσω κώδικα. Συγκεκριμένα, το αντικείμενό μας θα μπορεί να «ακούει» τρία μηνύματα σε κάποιο κρυφό κανάλι: next για επόμενη διαφάνεια, prev για προηγούμενη, reset για επανέξαρξη, και το url της παρουσίασης. Ο κώδικας είναι ο παρακάτω:

```
integer slide;
integer PROJECTOR_FACE = 3;
string baseUrl;

show() {
  llSetPrimMediaParams(PROJECTOR_FACE,
    [PRIM_MEDIA_AUTO_PLAY, TRUE,
     PRIM_MEDIA_CURRENT_URL, baseUrl+(string)slide,
     PRIM_MEDIA_HOME_URL, baseUrl+(string)slide,
     PRIM_MEDIA_HEIGHT_PIXELS, 512,
     PRIM_MEDIA_WIDTH_PIXELS, 512]);
}

default {
  state_entry() {
    llListen(55, "", "", "");
  }

  listen(integer chn, string name, key id, string message){
    if(message=="prev") {
      if(slide>1) slide--;
      show();
    }
    else if(message=="next") {
      slide++;
      show();
    }
    else if(message=="reset") {
      slide = 1;
      show();
    }
    else {
      integer index = llSubStringIndex(message, "edit");
      if(index<0) {
        llSay(0, "unknown url");
        return;
      }
      baseUrl = llGetSubString(message, 0,
```



```

        index-1)+"present?slide=";
        slide = 1;
        show();
    }
}
}

```

Στον παραπάνω κώδικα κατασκευάσαμε τη συνάρτηση `show`, μέσω της οποίας αλλάζουμε τη διεύθυνση της ιστοσελίδας στην επιφάνεια. Χρησιμοποιούμε τον σωστό αριθμό επιφάνειας (στην περίπτωσή μας 3, διότι το αντικείμενο κοιτάζει στο θετικό Y) και σχηματίζουμε τη διεύθυνση από μια συμβολοσειρά που χρησιμοποιούμε ως «βάση» (`baseURL`) και από τον τρέχοντα αριθμό διαφάνειας (`slide`). Το αντικείμενό μας ακούει σε ένα κρυφό κανάλι (55) και ανάλογα με το μήνυμα αυξάνει τον αριθμό διαφάνειας κατά ένα (`next`), τον μειώνει κατά ένα (`prev`), τον επαναφέρει στην αρχή (`reset`) ή αναγνωρίζει το μήνυμα ως URL και κατασκευάζει τη «βάση» της διεύθυνσης. Για να κατασκευάζει τη βάση, εντοπίζει τη θέση του `'edit'` στη διεύθυνση και το αντικαθιστά με `'present?slide='`.

Δοκιμάζουμε αν ο κώδικάς μας λειτουργεί σωστά δίνοντας URL από παρουσίαση που έχουμε κάνει `share` και στέλνοντας στη συνέχεια μηνύματα `next` και `prev` για να αλλάξουμε διαφάνειες. Επίσης, βεβαιωνόμαστε ότι όλοι οι χρήστες μπορούν να αντιληφθούν την αλλαγή.

Το τελικό στάδιο για τον μηχανισμό παρουσίασης είναι να κατασκευάσουμε τρία πλήκτρα (επόμενο, προηγούμενο και επανέναρξη), να τα τοποθετήσουμε κάτω από τον χώρο προβολής και να τους προσθέσουμε τον κώδικα που στέλνει `next`, `prev` και `reset` στο κανάλι 55 αντίστοιχα.

Το επόμενο βήμα μας είναι η κατασκευή ενός μηχανισμού ψηφοφορίας. Θα κατασκευάσουμε ένα αντικείμενο το οποίο θα έχει δύο καταστάσεις: κατάσταση αναμονής και κατάσταση ψηφοφορίας. Όταν ενεργοποιείται η κατάσταση ψηφοφορίας, θα ακούει μηνύματα σε ένα κρυφό κανάλι μέσω του οποίου θα καταγράφονται οι ψήφοι. Θα δίνουμε τέσσερις επιλογές στους χρήστες με αρίθμηση από το 1 μέχρι το 4. Ο κάθε χρήστης θα στέλνει την ψήφο του μέσω ειδικού πλήκτρου του HUD που θα κατασκευάσουμε αργότερα. Κάθε νέα ψήφος που στέλνεται θα καταγράφεται στον αριθμό των ψήφων της αντίστοιχης επιλογής. Για τον σκοπό αυτόν θα έχουμε μια λίστα (`results`) στην οποία θα καταγράφονται τέσσερις ακέραιοι. Αρχικά οι αριθμοί θα είναι μηδέν. Όταν θα λαμβάνεται ένα μήνυμα, π.χ. ψήφος v.3, θα εντοπίζεται ο τρέχων αριθμός ψήφων στην αντίστοιχη θέση της λίστας, δηλαδή τη θέση v.2 (θυμίζουμε ότι στις λίστες η αρίθμηση ξεκινάει από το μηδέν), θα αυξάνεται κατά ένα και θα ενημερώνεται η λίστα μέσω αντικατάστασης του συγκεκριμένου στοιχείου (εντολή `IIIListReplaceList`). Ο μηχανισμός θα διασφαλίζει επίσης ότι οι χρήστες δεν έχουν διπλοψηφίσει καταχωρώντας σε μια άλλη λίστα (`voters`) τα ονόματα των ψηφισάντων. Τέλος, όταν ολοκληρωθεί η ψηφοφορία, το αντικείμενο θα ανακοινώνει τα αποτελέσματα, δηλαδή τον αριθμό των ψήφων για κάθε επιλογή.

Ο κώδικάς μας είναι ο παρακάτω:

```

integer voteChannel = 119;
list voters;
list results;

default {
    state_entry() {
        llSetColor(<1,1,1>,ALL_SIDES);
        llSetText("Voting OFF",<0.5,1,0.5>,1);
    }
    touch_start(integer num) {
        state voting;
    }
}

state voting {
    state_entry() {
        llSetColor(<0,1,0>,ALL_SIDES);
        llSetText("Voting ON",<0.5,1,0.5>,1);
        llListen(voteChannel,"","","");
    }
}

```

```

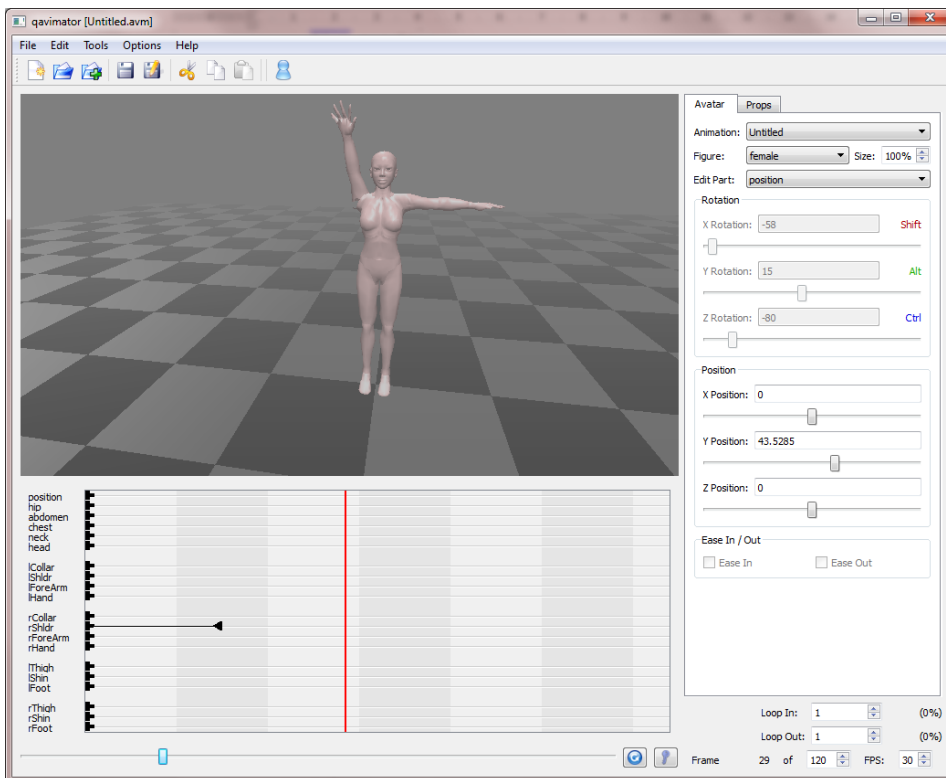
voters = [];
results = [0,0,0,0];
}

listen(integer chn, string name, key id, string message) {
integer index = llListFindList(voters,[name]);
if(index!=-1) {
voters += [name];
integer vote = (integer)message;
integer numOfVotes = llList2Integer(results,vote-1);
numOfVotes++;
results = llListReplaceList(results,[numOfVotes],
vote-1,vote-1);
}
else llSay(0,"You have already voted, "+name);
}

touch_start(integer num) {
llSay(0,"END OF VOTING! Results:");
integer i;
for(i=0;i<4;i++) {
llSay(0,(string)(i+1)+" - "+llList2Integer(results,i)+
" votes");
}
state default;
}
}
}

```

Δοκιμάζουμε το αντικείμενό μας κάνοντας ψηφοφορίες με μηνύματα στο κρυφό κανάλι επικοινωνίας (119) και βεβαιωνόμαστε ότι η λειτουργία του είναι σωστή.

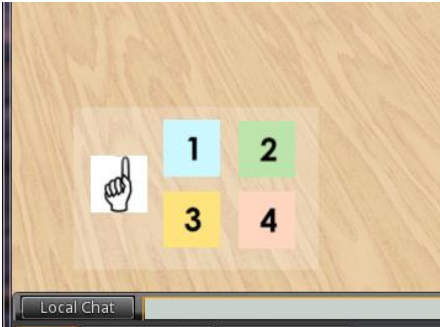


Εικόνα 12.9 Κατασκευή της συνθετικής κίνησης ανάτασης χεριού στο QAnimator.

Πριν κατασκευάσουμε το αντικείμενο που θα ενσωματωθεί στο HUD του χρήστη θα πρέπει να δημιουργήσουμε ένα animation ανάτασης χεριού. Ανοίγουμε το QAnimator και κατασκευάζουμε ένα animation στο οποίο η ενσάρκωση κρατάει σηκωμένο το δεξί χέρι για τρία δευτερόλεπτα και στη συνέχεια το κατεβάζει (εικόνα 12.9).

Εισάγουμε την κίνηση στον κόσμο δίνοντας προτεραιότητα (priority) 5 ώστε να μπορεί να εκτελείται και από καθιστούς χρήστες και αλλάζοντας τη στάση χεριού (hand pose) σε point right ώστε να υψώνεται ο δείκτης κατά την ανάταση. Δοκιμάζουμε την κίνηση στην ενσάρκωσή μας για να βεβαιωθούμε ότι εκτελείται σωστά. Αν προκύψουν προβλήματα, διορθώνουμε στο QAnimator.

Τέλος, κατασκευάζουμε ένα σύνθετο αντικείμενο με πέντε πλήκτρα: τέσσερα πλήκτρα ψηφοφορίας και ένα πλήκτρο ανάτασης. Κατασκευάζουμε και ανεβάζουμε τις απαραίτητες υφές για κάθε πλήκτρο, συνθέτουμε το τελικό αντικείμενο και το τοποθετούμε στο HUD μας δίνοντάς του κατάλληλο μέγεθος και θέση. Το αποτέλεσμα φαίνεται στην εικόνα 12.10.



Εικόνα 12.10 Τα πλήκτρα ανάτασης χεριού και ψηφοφορίας του αντικειμένου HUD που κατασκευάσαμε.

Τέλος, επεξεργαζόμαστε τα παραπάνω πλήκτρα για να προσθέσουμε τον κατάλληλο κώδικα. Στο πλήκτρο χεριού θα πρέπει να προσθέσουμε στα περιεχόμενα το animation που κατασκευάσαμε (raise_arm). Ο κώδικάς του θα είναι:

```
default {
    touch_start(integer num) {
        osAvatarPlayAnimation(llDetectedKey(0), "raise_arm");
    }
}
```

Για καθένα από τα πλήκτρα ψηφοφορίας ο κώδικας θα πρέπει να είναι όπως ο παρακάτω, αλλάζοντας κάθε φορά τον αριθμό που στέλνεται ως μήνυμα:

```
integer voteChannel = 119;

default {
    touch_start(integer num) {
        llShout(voteChannel, "1");
    }
}
```

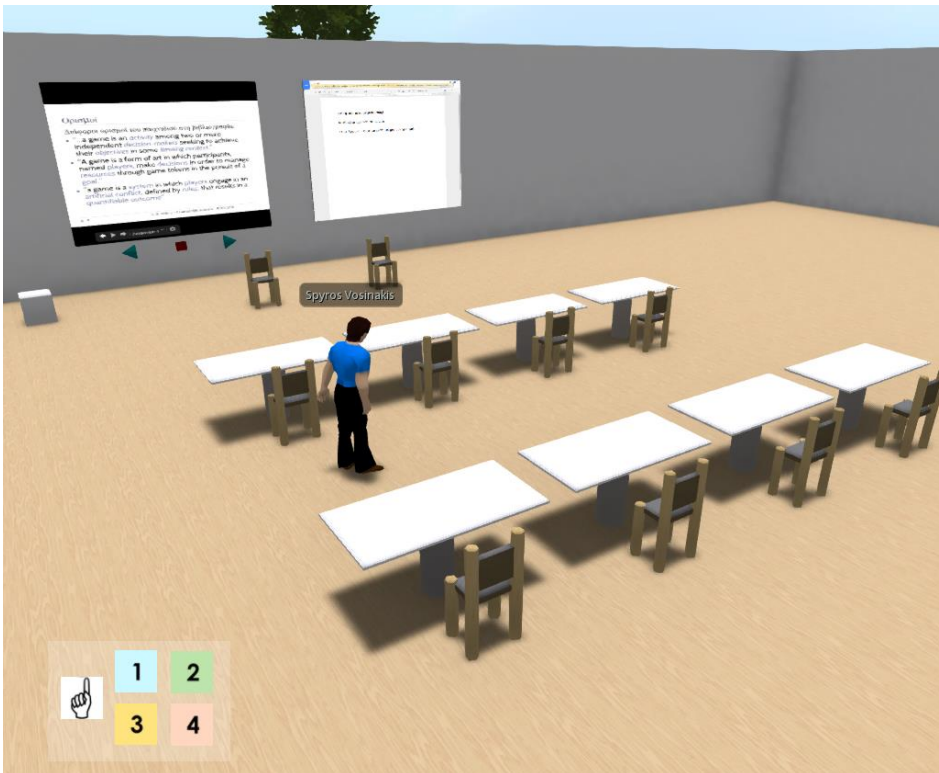
Αν όμως δοκιμάσουμε να τα χρησιμοποιήσουμε για ψηφοφορία, θα παρατηρήσουμε κάτι περίεργο. Το αντικείμενο ψηφοφορίας δεν αποθηκεύει πλέον το όνομά μας αλλά το όνομα του πλήκτρου που πατήσαμε, δηλαδή 'Primitive', αν δεν του έχουμε αλλάξει ονομασία. Αυτό είναι λογικό, αφού το μήνυμα δεν προήλθε από εμάς, αλλά από το πλήκτρο μας. Επομένως, θα πρέπει να διορθώσουμε τον κώδικα του αντικειμένου ψηφοφορίας για να υπολογίζονται οι ψήφοι σωστά. Αυτό που θα κάνουμε είναι να διαβάζουμε το κλειδί του αντικειμένου που έστειλε το μήνυμα και, μέσω αυτού, να εντοπίζουμε τον ιδιοκτήτη του. Μπορούμε να πετύχουμε κάτι τέτοιο με την εντολή llGetOwnerKey, η οποία θα μας δώσει το κλειδί του ιδιοκτήτη, και την llKey2Name, η οποία από το κλειδί θα μας δώσει το όνομά του.

Επομένως, για να διορθώσουμε το παραπάνω πρόβλημα, αλλάζουμε τον κώδικα του αντικειμένου ψηφοφορίας ως εξής:

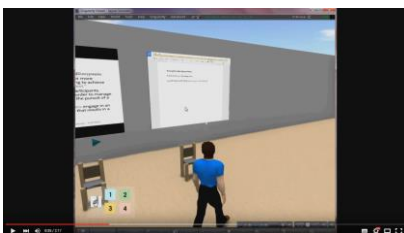
```
listen(integer chn, string name, key id, string message) {
    string owner = llKey2Name(llGetOwnerKey(id));

    integer index = llListFindList(voters, [owner]);
    if(index==-1) {
        voters += [owner];
        integer vote = (integer)message;
        integer numOfVotes = llList2Integer(results, vote-1);
        numOfVotes++;
        results = llListReplaceList(results, [numOfVotes],
            vote-1, vote-1);
    }
    else llSay(0, "You have already voted, "+owner);
}
```

Δηλαδή αρχικά εντοπίζουμε το όνομα του ιδιοκτήτη του αντικειμένου που ψήφισε, το αποθηκεύουμε στη μεταβλητή owner και στη συνέχεια χρησιμοποιούμε αυτήν ως όνομα ψηφοφόρου.



Εικόνα 12.11 Άποψη του κατασκευασμένου χώρου συνεργασίας.



Σύνδεσμος:
<https://youtu.be/OLxXvXoi3EI>

Video 12.2 Παρουσίαση χώρου συνεργασίας και λειτουργιών του.

Στο σημείο αυτό έχουμε ολοκληρώσει τη βασική κατασκευή του χώρου και των εργαλείων μας (βλ. εικ. 12.11). Δοκιμάζουμε συνολικά τη λειτουργία του χώρου, κατά προτίμηση με περισσότερους χρήστες, και συμπληρώνουμε ή διορθώνουμε τα στοιχεία στα οποία έχουμε εντοπίσει προβλήματα. Στο βίντεο 12.2 παρουσιάζεται ο χώρος συνεργασίας και οι λειτουργίες του.

3.4 Επεκτάσεις

Ο χώρος συνεργασίας που μόλις κατασκευάσαμε μπορεί να βελτιωθεί και να επεκταθεί με πολλούς διαφορετικούς τρόπους. Αρχικά, μπορούμε να καταγράψουμε και να προσθέσουμε παραπάνω διατάξεις του δωματίου και να προσαρμόσουμε τις επιλογές του αντίστοιχου πλήκτρου. Επιπλέον, μπορούμε να επαυξήσουμε τη λειτουργικότητα του μηχανισμού παρουσιάσεων προσθέτοντας πλήκτρο αλλαγής παρουσίασης, το οποίο ζητάει από τον χρήστη που το πάτησε να του στείλει στο δημόσιο κανάλι το URL της νέας παρουσίασης. Ένα άλλο σημείο που μπορεί να επεκταθεί σημαντικά είναι ζητήματα ασφάλειας και δικαιωμάτων πρόσβασης. Θα μπορούσαμε να ορίσουμε (π.χ. σε κάποιο potecard) ομάδα διαχειριστών, οι οποίοι θα είναι εξουσιοδοτημένοι να κάνουν ορισμένες ενέργειες στο περιβάλλον, π.χ. να ξεκινήσουν ψηφοφορία, να αλλάξουν διάταξη αίθουσας ή να αλλάξουν παρουσίαση. Και ο μηχανισμός ψηφοφοριών όμως θα μπορούσε να επεκταθεί περαιτέρω. Θα μπορούσε π.χ. να αλλάζει ο αριθμός των επιλογών (από 2 μέχρι 4) όταν ξεκινάμε μια νέα παρουσίαση, να κοινοποιούνται τα ονόματα των ψηφισάντων, να τίθενται χρονικά όρια λήξης της ψηφοφορίας κ.λπ.

Προχωρώντας ένα βήμα πιο πέρα, θα μπορούσαμε, μελετώντας τις ανάγκες συνεργασίας των πιθανών χρηστών ενός τέτοιου χώρου, να σχεδιάσουμε και να υλοποιήσουμε αρκετές ακόμη επεκτάσεις. Για παράδειγμα, ορισμένες περαιτέρω προσθήκες που ενδέχεται να έχουν νόημα είναι:

- να αξιοποιηθεί κάποια Web τεχνολογία για ζωντανό βίντεο streaming, ώστε απομακρυσμένοι παρουσιαστές να μπορούν να εμφανίζονται σε κάποια οθόνη του χώρου συνεργασίας,
- να κατασκευαστεί αντικείμενο στο οποίο ανακοινώνονται τα ονόματα όσων ζήτησαν να πάρουν τον λόγο με σειρά προτεραιότητας,
- να σχεδιαστούν επιπλέον εργαλεία συνεργασίας, όπως: πίνακες ανακοινώσεων, πίνακες για μικρές σημειώσεις post-it, πίνακες διαθέσεων (Moodboards).

Οι παραπάνω προσθήκες αναμένεται να κάνουν τον χώρο μας ακόμη πιο εύχρηστο και λειτουργικό.

4 Εικονική Εξιστόρηση

Αντικείμενο του τρίτου και τελευταίου παραδείγματός μας είναι η εικονική εξιστόρηση. Με τον όρο *εικονική εξιστόρηση* (*virtual storytelling*) αναφερόμαστε στην αναπαράσταση και αφήγηση ιστοριών με ψηφιακά μέσα, συνήθως βάσει κάποιου σχεδιασμένου εικονικού περιβάλλοντος. Οι ιστορίες αυτές μπορούν να είναι προκαθορισμένες ή και διαδραστικές επιτρέποντας τη συμμετοχή και παρέμβαση των χρηστών. Στη δεύτερη περίπτωση η διαδικασία περιγράφεται ως *διαδραστική εξιστόρηση* (*interactive storytelling*). Το περιβάλλον του OpenSimulator πληροί τις βασικές προϋποθέσεις για την προβολή ψηφιακών ιστοριών. Μπορεί να απεικονίσει φυσικά περιβάλλοντα με ανάγλυφο εδάφος, θάλασσα, ουρανό, ήλιο, σύννεφα κ.λπ. Μπορούν να προστεθούν κτίρια και αντικείμενα στο περιβάλλον, και, το κυριότερο, μπορούν να δημιουργηθούν ψηφιακοί χαρακτήρες οι οποίοι κινούνται και αλληλεπιδρούν μέσα στον κόσμο (βλ. Κεφ. 11). Δεν είναι τυχαίο ότι σε εικονικούς κόσμους, όπως το Second Life, αλλά και σε άλλους κόσμους παιχνιδιών έχει αναπτυχθεί και υποστηρίζεται από ενεργές κοινότητες χρηστών η τεχνική που ονομάζεται *machinima*, δηλαδή η παραγωγή ψηφιακών ιστοριών μέσω καταγραφής της δραστηριότητας του εικονικού κόσμου. Στις ιστορίες όμως που δημιουργούνται με το *machinima* οι ενέργειες εκτελούνται συνήθως από τους ίδιους τους χρήστες. Αυτό που έχει μεγαλύτερο ενδιαφέρον είναι η παραγωγή ιστοριών από ψηφιακούς χαρακτήρες βάσει κάποιου προσχεδιασμένου σεναρίου. Δυστυχώς, η δημιουργία κώδικα για την εκτέλεση ενεργειών από τους χαρακτήρες είναι, όπως παρατηρήσαμε, αρκετά επίπονη και μπορεί να την επιλέξει ως λύση μόνο ένας έμπειρος προγραμματιστής της LSL/OSSL. Αυτό που θα ήταν χρήσιμο για τους επίδοξους σκηνοθέτες/δημιουργούς θα ήταν η ύπαρξη μιας πιο απλής γλώσσας για την κατασκευή σεναρίων.

Στόχος του παραδείγματος που θα υλοποιήσουμε είναι να κατασκευάσουμε έναν μηχανισμό εικονικής εξιστόρησης με πολλαπλούς χαρακτήρες. Ο μηχανισμός μας θα μπορεί να διαβάσει ένα «σενάριο» σε μια ειδική γλώσσα που θα ορίσουμε και να το εκτελέσει δημιουργώντας τους κατάλληλους χαρακτήρες και ελέγχοντας

την εκτέλεση της ιστορίας. Τα οφέλη ενός τέτοιου μηχανισμού μπορούν να είναι πολλαπλά, καθώς διάφοροι χώροι εφαρμογής θα μπορούσαν εν δυνάμει να εκμεταλλευτούν τη δυνατότητα περιγραφής και αναπαραγωγής ιστοριών με ψηφιακούς χαρακτήρες. Ενδεικτικά αναφέρονται τα παιχνίδια, η εκπαίδευση, η προσομοίωση και ο πολιτισμός.

4.1 Σχεδίαση περιβάλλοντος και διεπαφής

Το πρώτο μας βήμα για τη σχεδίαση του μηχανισμού εξιστόρησης είναι να μελετήσουμε ποια είναι τα βασικά χαρακτηριστικά μιας ιστορίας και σε ποιο βαθμό μπορούν αυτά να μεταφερθούν στο OpenSimulator. Θα μπορούσαμε να πούμε ότι μια ιστορία έχει κατ' ελάχιστο:

- ένα στατικό σκηνικό που λειτουργεί ως περιβάλλον πλαίσιο της ιστορίας,
- έναν αριθμό από ενεργά αντικείμενα,
- ορισμένες προκαθορισμένες τοποθεσίες στις οποίες λαμβάνουν χώρα οι δράσεις,
- έναν αριθμό από χαρακτήρες οι οποίοι υπάρχουν και δρουν στον χώρο και
- δράσεις τις οποίες εκτελούν οι χαρακτήρες στη διάρκεια του χρόνου, όπως κίνηση, ομιλία, χρήση αντικειμένων κ.ά.

Θεωρητικά όλα τα παραπάνω μπορούν να υποστηριχθούν σε κάποιον βαθμό στο OpenSimulator. Ένας σχεδιαστής μπορεί εύκολα να διαμορφώσει το σκηνικό, να τοποθετήσει τα αντικείμενα στις κατάλληλες θέσεις και να ορίσει προκαθορισμένες τοποθεσίες για τη δράση των χαρακτήρων. Αυτό που είναι όμως σημαντικά πιο δύσκολο να καθοριστεί είναι η ακολουθία των δράσεων, που θα διαμορφώσει την ιστορία. Σε αυτό το στοιχείο θα εστιάσει ο μηχανισμός που θέλουμε να αναπτύξουμε.

Θα δίνουμε μέσω του μηχανισμού εξιστόρησης τη δυνατότητα στον σχεδιαστή να περιγράψει ένα σενάριο το οποίο θα περιλαμβάνει πολλαπλούς χαρακτήρες. Οι χαρακτήρες θα δημιουργούνται και θα τοποθετούνται σε κατάλληλες θέσεις κατά την εξέλιξη της ιστορίας. Καθένας από αυτούς θα έχει τη δική του εμφάνιση, η οποία θα πρέπει να είναι προ-αποθηκευμένη ως αντικείμενο notecard στα περιεχόμενα του μηχανισμού. Στο σενάριο θα περιγράψουμε τις δράσεις των χαρακτήρων και τον χρονισμό τους, δηλαδή τη χρονική διάρκεια που θα μεσολαβεί μεταξύ αυτών. Οι δράσεις τις οποίες θα υποστηρίξουμε θα σχετίζονται με το βάδισμα, την κίνηση, την ομιλία και τη χρήση αντικειμένων. Θεωρητικά βεβαίως σε μια ιστορία μπορούν να συμβούν πολύ περισσότερες αλλαγές στον κόσμο, όπως να εισαχθούν, να διαγραφούν ή και να τροποποιηθούν αντικείμενα του περιβάλλοντος, να παραχθούν νέοι ήχοι, να αλλάξει η ώρα της ημέρας κ.ά. Όμως, για λόγους απλότητας ο μηχανισμός μας θα εστιάσει σε ενέργειες των χαρακτήρων.

Έχοντας λοιπόν κατά νου τις δυνατότητες των ψηφιακών χαρακτήρων του OpenSimulator καταλήγουμε στην υποστήριξη των παρακάτω ενεργειών:

1. δημιουργία και τοποθέτηση χαρακτήρων στον κόσμο
 - η εμφάνιση θα βασίζεται σε προκαθορισμένα αρχεία και η τοποθέτηση σε προκαθορισμένες τοποθεσίες,
2. ομιλία χαρακτήρων,
3. εκτέλεση συνθετικών κινήσεων
 - για την περίπτωση κινήσεων που εκτελούνται σε βρόχο (loop) θα δίνεται η δυνατότητα τερματισμού,
4. βάδισμα προς κάποια προκαθορισμένη τοποθεσία,
5. κοίταγμα προς κάποια τοποθεσία ή χαρακτήρα.

Τέλος, ο μηχανισμός μας θα υποστηρίζει και μία επιπλέον ενέργεια χρονοκαθυστέρησης για τον χρονισμό των παραπάνω.

Παρατηρώντας τις παραπάνω ενέργειες διαφαίνεται η ανάγκη χρήσης προκαθορισμένων ονομάτων τοποθεσιών. Πράγματι, δεν θα είναι ιδιαίτερα εύχρηστο το να δημιουργεί ο σχεδιαστής ένα σενάριο τοποθετώντας συγκεκριμένες συντεταγμένες τοποθεσιών. Η λύση στην οποία καταλήγουμε είναι να υπάρχει ένα επιπλέον αρχείο με προκαθορισμένες θέσεις στον χώρο, η καθεμία από τις οποίες θα έχει μια ονομασία. Έτσι, στο σενάριο οι εντολές θα δίνονται με βάση την ονομασία των θέσεων και όχι με πραγματικές συντεταγμένες.

Συνοπτικά λοιπόν, τα βήματα ενός σχεδιαστή για τη δημιουργία μιας νέας ιστορίας είναι τα παρακάτω:

- διαμόρφωση του σκηνικού και των αντικειμένων,
- δημιουργία και αποθήκευση της εμφάνισης των χαρακτήρων,
- καθορισμός των θέσεων στις οποίες θα βρεθούν οι χαρακτήρες στη διάρκεια της ιστορίας και
- συγγραφή και δοκιμή του σεναρίου.

4.2 Τεχνική σχεδίαση

Η βασική διεπαφή των χρηστών με τον μηχανισμό θα είναι μέσω των αρχείων που θα προσθέτουν ή θα τροποποιούν σε αυτόν. Αυτό σημαίνει ότι θα πρέπει το αντικείμενό μας να έχει έναν απλό λεκτικό αναλυτή (parser) ώστε να μπορεί να επεξεργαστεί τα δεδομένα των αρχείων. Η LSL μάς δίνει τη δυνατότητα να χωρίσουμε μια συμβολοσειρά σε επιμέρους τμήματα μέσω της εντολής `IParseString2List`. Φαινομενικά η χρήση της εντολής αυτής χρησιμοποιώντας το κενό ως διαχωριστικό θα μπορούσε να αποτελέσει λύση. Δυστυχώς η κατάσταση είναι κάπως πιο σύνθετη. Κάποια από τα δεδομένα των εντολών του σεναρίου μας θα περιέχουν κενά, αλλά δεν θα θέλουμε να διαχωριστούν. Ένα παράδειγμα είναι το μήνυμα που θα λέει ένας ψηφιακός χαρακτήρας, το οποίο συνήθως θα είναι μία ή περισσότερες προτάσεις, επομένως θα περιλαμβάνει κενά. Δυστυχώς οι δυνατότητες λεκτικής επεξεργασίας που μας προσφέρονται στην LSL μέσω της παραπάνω εντολής δεν είναι τόσο ισχυρές ώστε, για παράδειγμα, να μας επιτρέπουν να αντιμετωπίσουμε μια φράση εντός εισαγωγικών ως ενιαία. Η εύκολη λύση στην οποία καταλήγουμε είναι να χρησιμοποιήσουμε ένα διαφορετικό σύμβολο ως διαχωριστή. Επιλέγουμε την κάθετη μπάρα (|).

Το αρχείο τοποθεσιών θα έχει το όνομα “locations” και σε κάθε γραμμή του θα δηλώνεται μια νέα περιοχή. Η μορφή κάθε γραμμής θα είναι: <όνομα περιοχής> <θέση>, όπου το όνομα θα είναι μια συμβολοσειρά χωρίς κενά και η θέση ένα διάνυσμα στη μορφή σύνταξης της LSL (<x, y ,z>). Για παράδειγμα, τα περιεχόμενα ενός αρχείου “locations” θα μπορούσαν να είναι τα παρακάτω:

```
entrance1|<72, 80, 21.5>
entrance2|<74, 78, 21.5>
front_of_house|<84, 98, 21.5>
...
```

Το αρχείο σεναρίου θα περιέχει μια ακολουθία εντολών προς εκτέλεση, όπου κάθε εντολή θα δηλώνεται σε διαφορετική γραμμή. Επειδή οι εντολές του σεναρίου μας θα αναφέρονται σε πολλαπλούς χαρακτήρες, θα πρέπει να υπάρχει ένας εύκολος τρόπος να τους ξεχωρίζουμε. Επιλέγουμε να δίνεται ένα «ψευδώνυμο» (alias) σε καθέναν από αυτούς κατά τη δημιουργία τους και η αναφορά σε αυτούς στις υπόλοιπες εντολές θα γίνεται μέσω του ψευδωνύμου. Οι εντολές που θα υποστηρίζουμε είναι οι παρακάτω:

- `npc|<ψευδώνυμο>|<όνομα>|<επώνυμο>|<εμφάνιση>|<τοποθεσία>`: δημιουργείται ένας νέος χαρακτήρας με το όνομα και το επώνυμο που δώσαμε. Η εμφάνισή του θα περιγράφεται σε αντίστοιχο αρχείο εμφάνισης (τύπου notecard) που θα υπάρχει στα περιεχόμενα του αντικειμένου, και η τοποθεσία στην οποία θα βρεθεί θα πρέπει να έχει δηλωθεί στο αρχείο τοποθεσιών. Στον νέο χαρακτήρα θα αποδοθεί το ψευδώνυμο που ορίσαμε.
- `say|<ψευδώνυμο>|<μήνυμα>`: ο χαρακτήρας με αυτό το ψευδώνυμο θα πει το μήνυμα στο δημόσιο κανάλι.
- `walkTo|<ψευδώνυμο>|<τοποθεσία>`: ο χαρακτήρας θα βαδίζει προς την επιλεγμένη τοποθεσία.
- `turnTo|<ψευδώνυμο>|<τοποθεσία>`: ο χαρακτήρας θα στραφεί προς την επιλεγμένη τοποθεσία.
- `anim|<ψευδώνυμο>|<κίνηση>`: ο χαρακτήρας εκτελεί συνθετική κίνηση. Το αρχείο της κίνησης θα πρέπει να υπάρχει στα περιεχόμενα του αντικειμένου ή να είναι μια από τις έτοιμες κινήσεις του Second Life.
- `stopAnim|<ψευδώνυμο>|<κίνηση>`: ο χαρακτήρας σταματάει την εκτέλεση της συνθετικής κίνησης.
- `wait|<χρονικό διάστημα>`: προκαλείται χρονοκαθυστέρηση της εκτέλεσης της ιστορίας για το χρονικό διάστημα που ορίσαμε (σε δευτερόλεπτα).

Στο σημείο αυτό θα πρέπει να σημειώσουμε πως μετά την εκτέλεση κάθε εντολής ο μηχανισμός μετακινείται στην επόμενη χωρίς να περιμένει να τερματιστεί, με εξαίρεση την wait. Αυτό σημαίνει ότι όποτε καλούμε ενέργειες με κάποια διάρκεια, π.χ. βάδισμα ή συνθετική κίνηση, θα πρέπει να προσθέσουμε και ανάλογη χρονοκαθυστέρηση ώστε να μπορέσουν οι ενέργειες αυτές να γίνουν ορατές. Ένα απλό παράδειγμα σεναρίου θα μπορούσε να είναι το παρακάτω:

```
npc|john|John|Smith|male|entrance1
npc|mary|Mary|Jane|female|entrance2
wait|2.0
say|john>Hello world!
walkTo|mary|front_of_house
wait|1.0
anim|john|bow
wait|2.0
turnTo|john|front_of_house
wait|1.0
say|john>Hello Mary
```

Στο παραπάνω παράδειγμα δημιουργούνται δύο NPCs με ψευδώνυμα 'john' και 'mary' και εκτελούν ενέργειες ομιλίας, βαδίσματος, συνθετικής κίνησης και περιστροφής με κατάλληλη χρονοκαθυστέρηση ανάμεσά τους.

Τέλος, θα εξετάσουμε πώς μπορούμε να κατασκευάσουμε έναν απλό μεταγλωττιστή (interpreter) για τη γλώσσα σεναρίων που μόλις ορίσαμε. Στόχος μας είναι να εκτελούνται οι εντολές ακολουθιακά και να μπορούμε να σταματήσουμε ή να επανεκκινήσουμε την ιστορία. Η λύση στην οποία καταλήγουμε περιγράφεται στις παρακάτω επιλογές:

- Όταν φορτώνεται το σενάριο, όλες οι εντολές αποθηκεύονται σε μία λίστα, τη λίστα commands.
- Χρησιμοποιούμε έναν δείκτη με όνομα activeCommand για να καταδείξουμε ποια είναι η θέση της τρέχουσας εντολής προς εκτέλεση στη λίστα.
 - Αν η τιμή της activeCommand γίνει -1 τότε έχει σταματήσει η εκτέλεση του προγράμματος.
- Κατασκευάζουμε μια συνάρτηση, την processCommand, η οποία ανακαλεί την τρέχουσα εντολή προς εκτέλεση (activeCommand) από τη λίστα (commands) και την αναλύει σε επιμέρους τμήματα χρησιμοποιώντας την κάθετη μπάρα ως διαχωριστή.
 - Το πρώτο στοιχείο (τύπου string) είναι το όνομα της εντολής, π.χ. npc. Ανάλογα με το όνομα διαβάζονται τα υπόλοιπα στοιχεία και εκτελείται η εντολή.
 - Μόλις ολοκληρωθεί η εκτέλεση της εντολής, καλούμε μια συνάρτηση, τη next, η οποία προχωράει τη θέση της τρέχουσας εντολής ένα βήμα εμπρός και ξανακαλεί την processCommand.
 - Αν η θέση βγει εκτός λίστας, αυτό σηματοδοτεί το τέλος του προγράμματος.
- Δημιουργούμε δύο ακόμη συναρτήσεις για τον έλεγχο της ροής, την startScenario, που αρχικοποιεί τη θέση της τρέχουσας εντολής και καλεί την εκτέλεσή της, και την endScenario, που την κάνει -1 για να σηματοδοτήσει τη λήξη της εκτέλεσης.
- Στην ειδική περίπτωση εκτέλεσης της εντολής wait καλούμε τον χρονιστή (HSetTimerEvent) για το χρονικό διάστημα που απαιτείται, και αφού παρέλθει ο χρόνος, δηλαδή κληθεί το συμβάν, καλούμε τη next για να προχωρήσουμε στην επόμενη εντολή.

Ο παραπάνω μηχανισμός εκτέλεσης εντολών είναι σε μεγάλο βαθμό ανεξάρτητος από το είδος των εντολών που θα εκτελεστούν. Νέες λειτουργίες μπορούν να προστεθούν εύκολα στη συνάρτηση processCommand χωρίς να επηρεάζεται η συνολική λειτουργία και χωρίς να χρειάζεται οποιαδήποτε άλλη αλλαγή στο πρόγραμμα.

4.3 Ανάπτυξη

Ξεκινάμε με την ανάπτυξη του βασικού μηχανισμού ανάγνωσης των αρχείων και εκτέλεσης των εντολών. Κατασκευάζουμε ένα απλό αντικείμενο και δημιουργούμε τον κώδικα που θα διαχειρίζεται και θα εκτελεί το σενάριο. Στο ίδιο αντικείμενο θα πρέπει υποχρεωτικά να υπάρχουν:

- τα αρχεία τύπου notecard της εμφάνισης των διάφορων χαρακτήρων της ιστορίας,
- ένα αρχείο με όνομα locations στο οποίο θα καταγράφονται οι περιοχές και
- ένα αρχείο με όνομα scenario που θα περιέχει τις εντολές του σεναρίου.

Υπενθυμίζεται ότι μπορούμε να αποθηκεύσουμε την εμφάνιση μιας οποιασδήποτε ενσάρκωσης σε αρχείο μέσω της εντολής osAgentSaveAppearance. Επομένως, θα πρέπει να κατασκευάσουμε ένα απλό αντικείμενο αποθήκευσης εμφάνισης (βλ. Κεφ. 11) και αφού τροποποιήσουμε κατάλληλα την εμφάνισή μας (ή άλλων χρηστών) να την αποθηκεύσουμε δίνοντας περιγραφικά ονόματα. Για τα παραδείγματα που θα δοκιμάσουμε έχουμε κατασκευάσει δύο εμφανίσεις, τη male και τη female, με μια απλή αντρική και γυναικεία εμφάνιση αντίστοιχα.

Ένας εύκολος τρόπος να καταγράψουμε περιοχές είναι να κατασκευάσουμε μικρά αντικείμενα τα οποία, όταν τους κάνει ένας χρήστης κλικ, ανακοινώνουν τη θέση τους. Έτσι, μπορούμε να τα τοποθετήσουμε στις τοποθεσίες που επιθυμούμε, να ελέγξουμε ότι αυτές μας ικανοποιούν και τέλος να αντιγράψουμε τις τιμές θέσης τους στο αρχείο locations. Για να δοκιμάσουμε το παράδειγμά μας δημιουργήσαμε τρεις τοποθεσίες, την entrance1, entrance2 και front_of_house.

Τέλος, επιλέγουμε το αντικείμενο στο οποίο θα τρέχει ο μηχανισμός να δίνει στον χρήστη τις παρακάτω επιλογές μέσα από μενού:

1. να φορτώσει το σενάριο,
2. να ξεκινήσει την ιστορία,
3. να σταματήσει την ιστορία και
4. να διαγράψει τους χαρακτήρες από τον κόσμο.

Ο αρχικός μας κώδικας είναι ο παρακάτω:

```
integer storyChannel = 188;
list npcs;
list locations;
list commands;
integer activeCommand;

vector getLocation(string location) {
    integer index = llListFindList(locations, [location]);
    return llList2Vector(locations, index+1);
}

key getNPC(string alias) {
    integer index = llListFindList(npcs, [alias]);
    return llList2Key(npcs, index+1);
}

next() {
    if(activeCommand<0) return;
    activeCommand++;
    processCommand();
}

processCommand() {
    //θα συμπληρωθεί στη συνέχεια
```

```

}

loadScenario() {
    locations = [];
    integer lines = osGetNumberOfNotecardLines("locations");
    integer i;
    for(i=0;i<lines;i++) {
        string location = osGetNotecardLine("locations",i);
        locations += llParseString2List(location,"|","");
    }
    llSay(0,"Locations loaded, "+(string)lines+" lines.");

    commands = [];
    lines = osGetNumberOfNotecardLines("scenario");
    for(i=0;i<lines;i++) {
        commands += [osGetNotecardLine("scenario",i)];
    }
    llSay(0,"Scenario loaded, "+(string)lines+" lines.");
}

startScenario() {
    activeCommand = 0;
    processCommand();
}

stopScenario() {
    llSay(0,"Scenario stopped!");
    activeCommand = -1;
}

deleteChars() {
    integer i;
    integer npcNum = llGetListLength(npcs);
    for(i=1;i<npcNum;i+=2) {
        osNpcRemove(llList2Key(npcs,i));
    }
    npcs=[];
}

default {
    state_entry() {
        llListen(storyChannel,"","","");
    }

    touch_start(integer num) {
        llDialog(llDetectedKey(0),"Select", ["load",
            "start","stop","delete characters"],storyChannel);
    }

    listen(integer chn, string name, key id, string message) {
        if(message=="load") loadScenario();
        else if(message=="start") startScenario();
        else if(message=="stop") stopScenario();
        else if(message=="delete characters") deleteChars();
    }
}

```

}

Στον παραπάνω κώδικα έχουμε κατασκευάσει τον βασικό μηχανισμό, χωρίς την επεξεργασία των επιμέρους εντολών. Όταν γίνεται κλικ στο αντικείμενο, δημιουργείται ένας διάλογος με τις επιλογές “load”, “start”, “stop” και “delete characters”. Η πρώτη επιλογή φορτώνει τις περιοχές και το σενάριο. Αρχικά ανοίγει το αρχείο locations, διαβάζει τις γραμμές του και για καθεμία από αυτές προσθέτει δύο εγγραφές σε μια λίστα locations: το όνομα και τη θέση. Ο διαχωρισμός της γραμμής σε δύο εγγραφές γίνεται μέσω της εντολής `llParseString2List`. Στη συνέχεια ανοίγει το αρχείο scenario και αποθηκεύει τις εντολές στη λίστα commands. Η επιλογή start καλεί τη συνάρτηση `startScenario` μέσω της οποίας ξεκινάει η εκτέλεση της ιστορίας. Η θέση της ενεργού εντολής (`activeCommand`) γίνεται μηδέν (πρώτη θέση στη λίστα) και καλείται η `processCommand` για την εκτέλεσή της. Με την επιλογή stop σταματάει η ιστορία δίνοντας στη μεταβλητή θέσης την τιμή -1 και με την `delete characters` διαγράφονται όλοι οι χαρακτήρες που δημιουργήθηκαν.

Ο κώδικάς μας χρησιμοποιεί τρεις λίστες για να αποθηκεύει τα δεδομένα της ιστορίας:

- Στη λίστα `npcs` καταχωρούνται όλοι οι χαρακτήρες που δημιουργήθηκαν στο σενάριο (μέσω της εντολής `npc`) στη μορφή `<ψευδώνυμο>, <κλειδί>`. Χάρη σε αυτήν τη λίστα μπορούμε να εντοπίσουμε το κλειδί ενός χαρακτήρα βάσει του ψευδωνύμου του (βλ. συνάρτηση `getNpc`), αλλά και να διαγράψουμε όλους τους χαρακτήρες που δημιουργήσαμε (βλ. συνάρτηση `deleteChars`).
- Η λίστα `locations` έχει όλες τις περιοχές στη μορφή `<όνομα>, <θέση>`. Έχουμε δημιουργήσει επιπλέον τη συνάρτηση `getLocation`, η οποία μας επιστρέφει τη θέση μιας περιοχής βάσει του ονόματός της.
- Τέλος, η λίστα `commands` περιέχει όλες τις εντολές του σεναρίου. Η μεταβλητή `activeCommand` εξυπηρετεί ως δείκτης στην τρέχουσα εντολή προς εκτέλεση και η συνάρτηση `next` προχωράει την εκτέλεση κατά ένα βήμα.

Εκτελώντας τον κώδικα θα πρέπει να βεβαιωθούμε ότι η φόρτωση του σεναρίου και των περιοχών γίνεται σωστά. Βεβαίως, δεν έχουμε υλοποιήσει ακόμη την `processCommand` ώστε να μπορούμε να δούμε την εκτέλεση των ιστοριών μας. Αυτό θα είναι το επόμενο μας βήμα.

Αρχικά στην `processCommand` θα ελέγχουμε αν έχει ολοκληρωθεί η εκτέλεση του σεναρίου. Αν η `activeCommand` είναι εκτός λίστας ή αν είναι ήδη -1, τότε το σενάριο δεν εκτελείται, επομένως τερματίζεται η συνάρτηση. Στη συνέχεια ανακαλούμε την εντολή από τη λίστα και μέσω της `llParseString2List` τη χωρίζουμε σε μικρότερα τμήματα χρησιμοποιώντας την κάθετη μπάρα ως διαχωριστή. Το πρώτο τμήμα θα είναι και το όνομα της εντολής. Διαβάζουμε τις τιμές των υπόλοιπων τμημάτων ανάλογα με το είδος της εντολής και καλούμε τη συνάρτηση `next` για να προχωρήσουμε στην επόμενη. Όμως στην περίπτωση της `wait` θα πρέπει να κάνουμε μια εξαίρεση. Η `next` δεν θα κληθεί απευθείας, αλλά μόλις παρέλθει ο απαιτούμενος χρόνος. Για να ελέγξουμε την πάροδο του χρόνου θα πρέπει να προσθέσουμε ένα συμβάν `timer()` στην `default` κατάσταση του προγράμματος.

Οι προσθήκες στον κώδικά μας θα είναι οι παρακάτω:

```
processCommand() {
    if(activeCommand >= llGetListLength(commands))
        activeCommand = -1;
    if(activeCommand == -1) return;

    string cmd = llList2String(commands, activeCommand);
    list commList = llParseString2List(cmd, "|", "");
    string command = llList2String(commList, 0);
    if(command=="npc") {
        string alias = llList2String(commList, 1);
        string fname = llList2String(commList, 2);
        string lname = llList2String(commList, 3);
        string app = llList2String(commList, 4);
        string location = llList2String(commList, 5);
        vector pos = getLocation(location);
```

```

    key npcKey = osNpcCreate(fname, lname, pos, app);
    npcs += [alias, npcKey];
    next();
}
else if(command=="wait") {
    float time = llList2Float(commList,1);
    llSetTimerEvent(time);
}
else if(command=="say") {
    string alias = llList2String(commList, 1);
    key npc = getNpc(alias);
    string msg = llList2String(commList, 2);
    osNpcSay(npc,msg);
    next();
}
else if(command=="walkTo") {
    string alias = llList2String(commList, 1);
    key npc = getNpc(alias);
    string location = llList2String(commList, 2);
    vector pos = getLocation(location);
    osNpcMoveToTarget(npc,pos,OS_NPC_NO_FLY);
    next();
}
else if(command=="anim") {
    string alias = llList2String(commList, 1);
    key npc = getNpc(alias);
    string animName = llList2String(commList, 2);
    osNpcPlayAnimation(npc, animName);
    next();
}
else if(command=="stopAnim") {
    string alias = llList2String(commList, 1);
    key npc = getNpc(alias);
    string animName = llList2String(commList, 2);
    osNpcStopAnimation(npc, animName);
    next();
}
else if(command=="turnTo") {
    string alias = llList2String(commList, 1);
    key npc = getNpc(alias);
    string location = llList2String(commList, 2);
    vector pos = getLocation(location);
    vector npcPos = osNpcGetPos(npc);
    vector diff = pos-npcPos;
    diff.z = 0;
    osNpcSetRot(npc,llRotBetween(<1,0,0>,diff));
    next();
}
}
...
default {
...
    timer() {
        llSetTimerEvent(0);
        next();
    }
}

```

}
}

Τώρα μπορούμε να ελέγξουμε συνολικά τον κώδικά μας. Κατασκευάζουμε έναν αριθμό από απλά σενάρια για να δοκιμάσουμε ότι οι εντολές εκτελούνται σωστά, φροντίζοντας να έχουμε προσθέσει τα απαραίτητα χρονικά κενά μεταξύ τους. Κάθε φορά που κάνουμε μια αλλαγή στο σενάριο δεν είναι υποχρεωτικό να επανεκκινούμε τον κώδικα του αντικειμένου. Μπορούμε απλά να διαγράψουμε τους χαρακτήρες και να κάνουμε load ξανά το σενάριο. Το βίντεο 12.3 παρουσιάζει μια μικρή ιστορία που κατασκευάσαμε και η εικόνα 12.12 απεικονίζει ένα στιγμιότυπο αυτής.



Σύνδεσμος:

<https://youtu.be/XCjHxX4z1zI>

Video 12.3 Αναπαραγωγή ιστορίας μέσω του μηχανισμού εικονικής εξιστόρησης.

4.4 Επεκτάσεις

Ο μηχανισμός εξιστόρησης που αναπτύξαμε ενσωματώνει ορισμένες μόνο βασικές λειτουργίες για την απόδοση ιστοριών και θα μπορούσε να βελτιωθεί και να επεκταθεί με διάφορους τρόπους. Αρχικά, θα ήταν χρήσιμο να μην χρειάζεται να διαγραφούν οι χαρακτήρες κάθε φορά που ξεκινάει εκ νέου η εκτέλεση της ιστορίας. Θα μπορούσε το πρόγραμμα να ελέγχει αν το ψευδώνυμο ενός χαρακτήρα υπάρχει ήδη και, αν ναι, να μην τον δημιουργεί ξανά. Μια ακόμη πιο ενδιαφέρουσα βελτίωση θα ήταν να έχουν ενεργό ρόλο και τα αντικείμενα του περιβάλλοντος στις ιστορίες. Θα μπορούσε, εκτός από τις τοποθεσίες και το σενάριο, να προστεθεί κι ένα τρίτο αρχείο στον μηχανισμό μας που θα περιλαμβάνει ονόματα αντικειμένων και αντίστοιχα κλειδιά (keys). Έτσι, θα μπορούσε και η γλώσσα σεναρίων να επεκταθεί με εντολές, όπως το κάθισμα σε ένα αντικείμενο και η ενεργοποίηση (touch) ενός αντικειμένου. Επιπλέον, θα ήταν ίσως πιο χρήσιμο αντί για τις πολλαπλές εντολές wait να δηλώνεται απλά η θέση της κάθε εντολής σε έναν νοητό χρονοδιάδρομο, π.χ. στο πρώτο δευτερόλεπτο θα εκτελεστεί η πρώτη εντολή, η επόμενη στο τρίτο, η μεθεπόμενη στο όγδοο κ.λπ. Τέλος, αν θέλουμε να κάνουμε τον μηχανισμό μας ακόμη πιο ισχυρό, μπορούμε να προσθέτουμε ενσωματώσεις σε κάθε ψηφιακό χαρακτήρα, οι οποίες εκτελούν ειδικές συμπεριφορές, και μέσω του σεναρίου μας να ενεργοποιούμε αντίστοιχες λειτουργίες, για παράδειγμα να βάλουμε έναν χαρακτήρα να ακολουθεί κάποιον άλλον, να ακολουθεί ένα προκαθορισμένο μονοπάτι κ.λπ. Έχουμε ήδη εξετάσει στο Κεφάλαιο 11 ορισμένες βασικές τεχνικές για να υλοποιήσουμε σύνθετες χωρικές λειτουργίες.



Εικόνα 12.12 Στιγμιότυπο αφήγησης ιστορίας μέσω του μηχανισμού εξιστόρησης που αναπτύξαμε.

Ο μηχανισμός που υλοποιήσαμε στο παράδειγμά μας θα μπορούσε να επεκταθεί και να αξιοποιηθεί περαιτέρω σε διάφορους χώρους εφαρμογής. Ορισμένα παραδείγματα είναι τα παρακάτω:

- σε περιβάλλοντα εξάσκησης θα μπορούσε να αξιοποιηθεί για την παρουσίαση της εκτέλεσης περίπλοκων διαδικασιών ή και του χειρισμού συσκευών,
- σε εκπαιδευτικά περιβάλλοντα θα μπορούσαν να εμφανίζονται εικονικοί εκπαιδευτές και να παρουσιάζουν νέες έννοιες και αντικείμενα μελέτης,
- σε χώρους πολιτισμού θα μπορούσε να γίνεται αναπαράσταση διαδικασιών, ιστορικών γεγονότων, τρόπου κατασκευής ή χρήσης αντικειμένων κ.λπ.,
- σε περιβάλλοντα ψυχαγωγίας θα μπορούσαν να παρουσιάζονται ιστορίες με ψηφιακούς χαρακτήρες για λόγους αύξησης της αληθοφάνειας του περιβάλλοντος και
- οι ίδιοι οι χρήστες θα μπορούσαν να δημιουργούν τις δικές τους ιστορίες για ψυχαγωγία ή για την απόκτηση ορισμένων δεξιοτήτων, π.χ. προγραμματισμό.

5 Συμπεράσματα

Από την ολοκλήρωση του δωδέκατου και τελευταίου κεφαλαίου γίνεται σαφές ότι το περιβάλλον OpenSimulator και η γλώσσα προγραμματισμού LSL/OSSL που το συνοδεύει είναι σε θέση, παρά τους όποιους περιορισμούς, να οδηγήσουν στην ανάπτυξη σύνθετων εφαρμογών σε διάφορες περιοχές. Στα τρία παραδείγματα που παρουσιάσαμε συνδύασαμε τις γνώσεις που αποκτήσαμε στα προηγούμενα κεφάλαια και αναπτύξαμε με σχετική ευκολία αντίστοιχες εφαρμογές που ικανοποιούν τους βασικούς στόχους τους. Είδαμε πώς μπορούμε να αξιοποιήσουμε τον μηχανισμό φυσικής και τη δυναμική δημιουργία αντικειμένων για την κατασκευή ενός παιχνιδιού βολών, πώς να κατασκευάσουμε βασικά εργαλεία συνεργασίας εκμεταλλευόμενοι την προβολή ιστοσελίδων σε επιφάνειες του κόσμου και πώς να σχεδιάσουμε μια απλή γλώσσα σεναρίων και να υλοποιήσουμε έναν απλό μεταγλωττιστή για την κίνηση και εκτέλεση ενεργειών ψηφιακών χαρακτήρων. Παρατηρήσαμε, τέλος, ότι καθένα από τα τρία αυτά παραδείγματα μπορεί να επεκταθεί πολύ περισσότερο και να αποκτήσει μεγαλύτερη αξία σε χώρους όπως η εκπαίδευση, η ψυχαγωγία, η συνεργατική εργασία και ο πολιτισμός. Αυτό καταδεικνύει και το εύρος των δυνατοτήτων του συγκεκριμένου περιβάλλοντος.

Στο σημείο αυτό ολοκληρώνεται το δεύτερο μέρος αυτού του βιβλίου, που αφορούσε την εκμάθηση του εικονικού κόσμου OpenSimulator. Γνωρίσαμε ένα μεγάλο εύρος των δυνατοτήτων του εργαλείου και αποκτήσαμε δεξιότητες ανάπτυξης σύνθετων παραδειγμάτων σε αυτό. Αυτό όμως δεν σημαίνει σε καμία περίπτωση ότι έχουμε εξαντλήσει το αντικείμενο. Υπάρχουν ακόμη πολλά στοιχεία που μπορείτε να ανακαλύψετε και πολλές περισσότερες εφαρμογές που μπορείτε να δοκιμάσετε. Άλλωστε σε όλα αυτά τα χρόνια ύπαρξης του Second Life και κατ' επέκταση του OpenSimulator έχει κατασκευαστεί ένας τεράστιος αριθμός περιοχών με πλούσια διαδραστικά στοιχεία και με μεγάλο αριθμό χρηστών. Εφόσον έχετε αποκτήσει κι εσείς τις δεξιότητες σχεδίασης και ανάπτυξης περιεχομένου στους κόσμους αυτούς μέσα από τη μελέτη των παραδειγμάτων και την επίλυση των ασκήσεων αυτού του βιβλίου, μπορείτε να εξελίξετε τις ικανότητές σας ακόμη περισσότερο μελετώντας υπάρχουσες λύσεις και δοκιμάζοντας δικές σας ιδέες και επεκτάσεις.

Ασκήσεις

1. Κατασκευάστε ένα παιχνίδι τύπου Space Invaders. Θα ελέγχετε μέσω πλήκτρων του HUD την οριζόντια κίνηση ενός διαστημόπλοιου και θα εκτοξεύετε σφαίρες ενάντια σε κινούμενους εχθρούς.
2. Σχεδιάστε έναν χώρο που θα λειτουργήσει ως εργαστήριο ελεύθερου πειραματισμού για την εκμάθηση βασικών εννοιών της φυσικής, όπως βαρύτητα, κρούση, τριβή κ.ά.
3. Κατασκευάστε ένα σύνολο εργαλείων για την ενίσχυση της εμπειρίας των επισκεπτών σε ένα εικονικό μουσείο. Οι επισκέπτες θα πρέπει να μπορούν: να διαβάζουν πληροφορίες σχετικά με κάθε έκθεμα, να αξιολογούν (π.χ. like/dislike), να αφήνουν σχόλια και να προκαλούν αυτόματα περιστροφή των εκθεμάτων για καλύτερη επισκόπηση.

4. Δημιουργήστε έναν χώρο συνεργασίας αρχιτεκτόνων ή και σχεδιαστών εσωτερικών/εξωτερικών χώρων πλαισιωμένο με τα κατάλληλα εργαλεία συνεργασίας, όπως ανακοινώσεις, τεχνικά σχέδια, επισημειώσεις με σχόλια που τοποθετούνται σε αντικείμενα ή περιοχές, απεικόνιση ρόλων, μηχανισμούς αυτόματης ξενάγησης κ.ά.
5. Συνδυάστε το δεύτερο και τρίτο παράδειγμα για την κατασκευή ενός εικονικού παρουσιαστή ο οποίος κινείται, μιλά και ελέγχει μια παρουσίαση βάσει σεναρίου.

Ευρετήριο

- Angry Birds, 288
- animation. *βλ. συνθετική κίνηση*
- billboard, 58
- Collada, 210
- event handler, 224, 226
- gimbal lock, 46
- Head Mounted Display
 - Oculus Rift, 31, 116
- HUD. *βλ. διαδιάστατη διεπαφή - HUD*
- keyframing. *βλ. συνθετική κίνηση*
- Linden Scripting Language (LSL), 222
- LSL εντολές
 - do-while, 230
 - for, 231
 - if-then-else, 229
 - while, 230
- LSL συμβάντα
 - collision, 242
 - collision_end, 242
 - collision_start, 242
 - link_message, 249
 - listen, 240
 - no_sensor, 241
 - on_rez, 257
 - sensor, 241
 - timer, 243
 - touch, 239
 - touch_end, 239
 - touch_start, 227, 239
- LSL συναρτήσεις
 - llApplyImpulse, 255
 - llDeleteSublist, 262
 - llDetectedKey, 252
 - llDetectedName, 230
 - llDetectedPos, 235
 - llDialog, 259
 - llEuler2Rot, 237
 - llFrnd, 258
 - llGetListLength, 263
 - llGetOwner, 281
 - llGetPos, 235
 - llGetRot, 237
 - llGetScale, 238
 - llList2Float, 263
 - llList2Integer, 263
 - llList2Key, 263
 - llList2Rot, 263
 - llList2String, 263
 - llList2Vector, 263
 - llListen, 239
 - llListFindList, 263
 - llListInsertList, 263
 - llListReplaceList, 263
 - llParseList2String, 264
 - llMessageLinked, 249
 - llMoveToTarget, 255
 - llRezObject, 256
 - llRot2Euler, 237
 - llRotBetween, 280
 - llSay, 232
 - llSensor, 240
 - llSensorRemove, 242
 - llSensorRepeat, 242
 - llSetAlpha, 235
 - llSetColor, 233
 - llSetForce, 255
 - llSetKeyframedMotion, 252
 - llSetPos, 235
 - llSetRot, 237
 - llSetScale, 238
 - llSetText, 232
 - llSetTexture, 234
 - llSetTimerEvent, 243
 - llShout, 232
 - llSitTarget, 260
 - llSleep, 233
 - llStartAnimation, 251
 - llTargetOmega, 250
 - llVecDist, 236
 - llVecMag, 236
 - llVecNorm, 236
 - llWhisper, 232
- LSL τύποι δεδομένων
 - float, 227
 - integer, 227
 - key, 227
 - list, 227
 - rotation, 227
 - string, 227
 - vector, 227
- machinima, 161, 311
- MeshLab, 210
- Minecraft, 31
- MySQL, 177
- OpenSimulator, 30
 - grid, 173
 - hypergrid, 173
 - standalone, 173
 - ανάγλυφο εδάφους (terrain), 180, 211
 - αρχεία αποθετηρίου, 184
 - αρχεία κόσμου, 183

βάση δεδομένων, 173
 διανομές, 176
 ιδιοκτησία (estate), 178
 περιοχή (region), 178
 υπηρεσίες (services), 172
 OpenSimulator Scripting Language (OSSL), 222
 OSSL συναρτήσεις
 osAgentSaveAppearance, 273
 osAvatarPlayAnimation, 251
 osAvatarStopAnimation, 252
 osForceAttachToAvatar, 261
 osGetNotecard, 265
 osGetNotecardLine, 266
 osGetNumberOfNotecardLines, 266
 osNpcCreate, 271
 osNpcGetPos, 274
 osNpcGetRot, 274
 osNpcMoveToTarget, 274
 osNpcPlayAnimation, 275
 osNpcRemove, 272
 osNpcSay, 275
 osNpcSetRot, 274
 osNpcShout, 275
 osNpcSit, 274
 osNpcStand, 274
 osNpcStopAnimation, 275
 osNpcStopMoveToTarget, 274
 osNpcTouch, 275
 osNpcWhisper, 275
 osOwnerSaveAppearance, 273

QAvimator, 217
 quaternions, 46

Second Life, 29
 εικονικά μουσεία, 156
 επιχειρείν, 166
 Singularity viewer, 179

virtual storytelling, 311
 Vivox voice, 180

World of Warcraft, 30

αλληλεπίδραση με αντικείμενα, 13, 105, 189
 απελευθέρωση, 108
 επιλογή, 106
 περιορισμοί, 109
 χειρισμός, 106
 αναγνώριση σύγκρουσης, 87
 bounding box, 87
 αναπαράσταση κόσμου
 αντικατάσταση αισθήσεων, 130
 βαθμός ρεαλισμού, 129
 επιλογή αναπαραστάσεων, 128
 ήχος, 130
 ρεαλιστική, 129
 απεικόνιση, 58
 προοπτική προβολή, 60
 στερεό παρατήρησης, 59
 στερεοσκοπία, 61
 αποθετήριο (inventory), 189
 αρθρωτά αντικείμενα, 75

βασικά στερεά (prims), 196
 building block type, 203
 dimple, 206
 hollow, 204
 path cut, 203
 slice begin and end, 206
 taper, 204
 twist, 204
 σύνδεσμος (link), 201

γεωμετρικοί μετασχηματισμοί, 43
 αλλαγή κλίμακας, 45, 198
 μετατόπιση, 44, 198
 περιστροφή, 45, 198
 γλυπτά στερεά (sculpted prims), 209
 γράφος σκηνής (scene graph), 62
 γωνίες Euler, 46

διαμοίραση εμπειριών, 110
 διαφάνεια, 49, 200
 δισδιάστατη διεπαφή - HUD, 113, 214

εικονικά μουσεία, 157
 εικονική πραγματικότητα, 24
 ιστορική αναδρομή, 25
 ορισμός, 24
 περιβάλλοντα επιφάνειας εργασίας, 27
 εικονική ταυτότητα, 15
 εικονικό νόμισμα, 165
 εικονικοί κόσμοι
 αλληλεπίδραση χρηστών, 99
 εφαρμογές, 147
 ιστορική αναδρομή, 27
 ορισμός, 12
 εκπαίδευση
 εγκαθιδρυμένη μάθηση, 148
 εμπειρική μάθηση, 152
 εποικοδομισμός, 149
 κοινωνικός εποικοδομισμός, 149
 παιγνιοποίηση (gamification), 153
 συμπεριφορισμός, 150
 εμπύθιση (immersion), 24, 116
 ενσάρκωση (avatar)
 διαμόρφωση, 191
 ενσωμάτωση (attachment), 213
 εξερεύνηση περιεχομένου, 16
 επικοινωνία, 110, 188
 ομαδικές συζητήσεις, 18, 301
 χωροθετημένη, 14
 επίπεδα λεπτομέρειας (LOD), 43, 210
 επιστημονική οπτικοποίηση, 133
 ευφείς πράκτορες, 91
 αποφυγή εμποδίου, 94
 αρχιτεκτονικές, 91
 εύρεση μονοπατιού, 92
 κίνηση πλήθους, 94

ευχρηστία, 135
 αξιολόγηση, 135
 σχεδιαστικές κατευθύνσεις, 137

ηχοποίηση (sonification), 130

κάμερα, 57
 πεδίο όρασης (FOV), 57
 κανονικό διάνυσμα, 42, 52, 53

λογισμικό
 Direct3D, 68
 OpenGL, 68
 βιβλιοθήκες προγραμματισμού, 68
 μηχανές γραφικών, 69
 μηχανές παιχνιδιών, 69
 μοντελοποίηση, 69

μέθοδοι αξιολόγησης
 γνωστική περιδιάβαση, 136
 διαμορφωτική αξιολόγηση, 136
 ευρετική αξιολόγηση, 136
 συγκριτική αξιολόγηση, 136

μεθοδολογίες αξιολόγησης
 ακολουθιακή αξιολόγηση, 138
 γνωστική περιδιάβαση, 141
 πλατφόρμα δοκιμών, 139

μεθοδολογίες σχεδίασης, 123
 VRID, 126
 συνάντηση στη μέση, 124
 σχεδίαση εικονικών περιβαλλόντων - Fencott, 125

μηχανή καταστάσεων (state machine), 90, 222
 μοντελοποίηση επιχειρηματικών διαδικασιών, 167

παιχνίδια ρόλων, 17

παραμορφώσιμα αντικείμενα, 58, 76, 208
 free form deformation, 77
 μετατόπιση κορυφών, 76
 μορφοποίηση, 76

παρουσία (presence), 24

περιοχές εφαρμογής, 19
 αρχιτεκτονική, 22
 εκπαίδευση, 20, 148
 επιχειρείν, 165
 θεραπεία, 23
 πολιτισμός, 21, 154
 σχεδίαση, 22, 159
 τέχνες, 23
 ψυχαγωγία, 20

πλάνα ανά δευτερόλεπτο (FPS), 43, 73, 217

πλοήγηση, 13, 100, 187
 εύρεση πορείας, 104
 τεχνικές μετακίνησης, 101

πολυγωνικό πλέγμα, 42, 209

πολυχρηστικό περιβάλλον, 13

προσομοίωση, 17, 132

σημειώματα (notecards), 190

συμβάντα (events), 223

συμπαγή αντικείμενα (rigid bodies), 74

συνέπεια κόσμου, 112

συνεργασία, 14, 112, 134, 301
 ομάδες χρηστών, 112
 συνεργατική σχεδίαση, 18, 162

συνθετική κίνηση (animation), 73
 procedural animation, 84
 πλάνα κλειδιά (keyframing), 81, 216, 252
 σκελετική κίνηση, 75
 σύλληψη κίνησης, 83

συνθετικοί χαρακτήρες, 79

σχεδίαση
 συμμετοχική σχεδίαση, 160, 164

σωματίδια (particles), 58, 78

τρισδιάστατα γραφικά
 voxels, 40
 αναπαραστάσεις επιφάνειας, 39
 αναπαραστάσεις όγκου, 39
 μοντέλα, 39
 μορφότυποι (file formats), 69
 πολυγωνικό μοντέλο, 40

τρισδιάστατα μοντέλα
 βελτιστοποίηση, 67

τρισδιάστατη αναπαράσταση, 133

τρισδιάστατη μοντελοποίηση, 64
 NURBS, 65
 εξώθηση (extrusion), 64
 στερεά γεωμετρία (CSG), 65
 ψηφιακή γλυπτική, 66

τρισδιάστατο περιβάλλον, 13

τρισδιάστατη μοντελοποίηση
 επιφάνειες υποδιαίρεσης, 66

υφή, 48, 191, 200
 αντιστοίχιση (mapping), 50, 200

φυσικές διεπαφές
 kinect, 115
 wiimote, 116
 φορητές συσκευές, 115

φυσική μοντελοποίηση, 85
 δομή μαζών-ελατηρίων, 89
 κούκλα (ragdoll), 89

φωτισμός, 50, 212
 ανάκλαση διάχυσης (diffuse), 51
 κατοπτρική ανάκλαση (specular), 51
 περιβάλλοντος (ambient), 50
 προβολέας (spot light), 51
 σημειακό φως (point light), 51
 σκιές, 56
 φως κατεύθυνσης (directional light), 51

φωτοσκίαση
 Gouraud, 53
 Phong, 53
 radiosity, 54
 shaders, 55
 απεικόνιση αναγλύφου (bump mapping), 55
 απεικ. περιβάλλοντος (environmental mapping), 55
 απεικόνιση φωτισμού (light mapping), 54
 επίπεδη (flat), 52

ομαλή (smooth), 52

χάρτης κάτοψης (minimap), 105
χρώμα, 48, 200

HSV, 49

RGB, 48

ψηφιοποίηση, 66