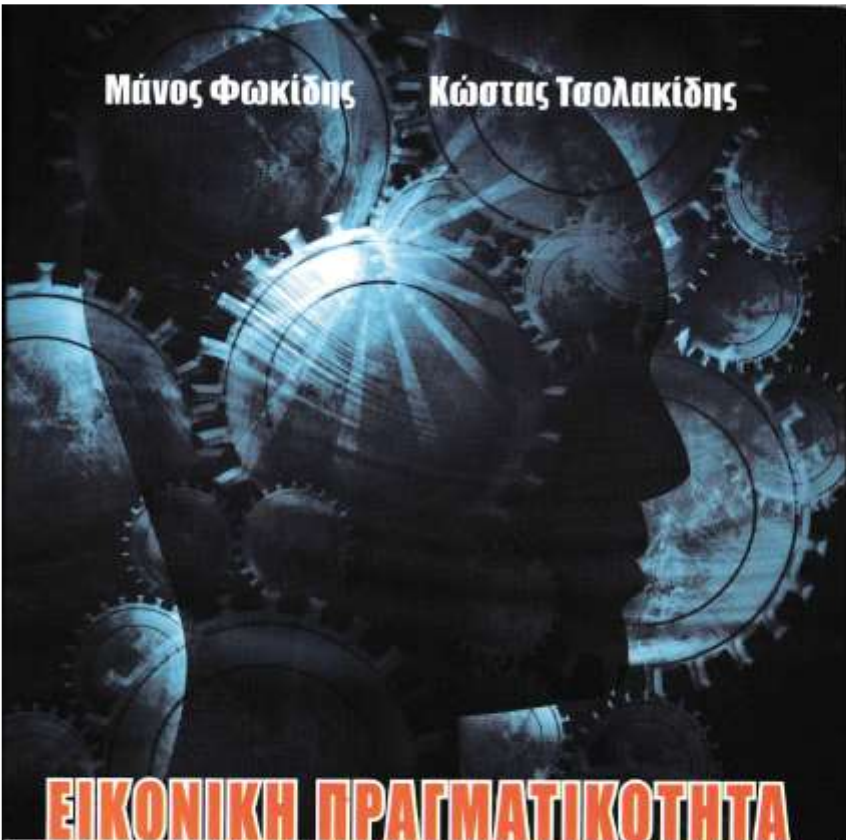




Μάνος Φωκίδης

Κώστας Τσολακίδης

ΕΙΚΟΝΙΚΗ ΠΡΑΓΜΑΤΙΚΟΤΗΤΑ ΣΤΗΝ ΕΚΠΑΙΔΕΥΣΗ

Θεωρία και Πράξη

 *ιδιάδραση*

Εκδόσεις ΔΙΑΔΡΑΣΗ
Σουλίου 73, 13461 Ζεφύρι
Τηλ. 210-2474950, fax. 210-2474902
e-mail: diadrassipublications@yahoo.gr

ISBN: 978-960-9541-35-0

Μάνος Φωκίδης, Κώστας Τσολακίδης

Εικονική Πραγματικότητα στην Εκπαίδευση:

Θεωρία και Πράξη

Στην Κορίνα, στη Δανάη και στη Λίλη

Περιεχόμενα

Εισαγωγή	9
Κεφάλαιο 1 ^ο : Σύγχρονες θεωρίες μάθησης και ο ρόλος της Πληροφορικής	17
Κεφάλαιο 2 ^ο : Γενιές εκπαιδευτικής τεχνολογίας και η διαδικασία μάθησης	35
Κεφάλαιο 3 ^ο : Εικονική Πραγματικότητα	49
Κεφάλαιο 4 ^ο : Τεχνικά θέματα υλικού και λογισμικού	75
Κεφάλαιο 5 ^ο : Εναλλακτική πρόταση στο πρόβλημα κατασκευής εικονικών κόσμων	105
Κεφάλαιο 6 ^ο : Οι μηχανές παιχνιδιών	123
Κεφάλαιο 7 ^ο : Οι νέες “Νέες Τεχνολογίες”	135
Κεφάλαιο 8 ^ο : Τα πρώτα βήματα στο Reality Factory	143
Κεφάλαιο 9 ^ο : Κατασκευή εικόνων/υφών και πακέτων υφών	167
Κεφάλαιο 10 ^ο : Μετατροπή τρισδιάστατων μοντέλων για χρήση στο Reality Factory	185
Κεφάλαιο 11 ^ο : Το περιβάλλον εργασίας του RFEEditPro	215
Κεφάλαιο 12 ^ο : Προσθήκη και χειρισμός υφών. Ιδιότητες γεωμετρικών στερεών	247
Κεφάλαιο 13 ^ο : Ουρανός και ορίζοντας	263
Κεφάλαιο 14 ^ο : Ήχος και φως	279
Κεφάλαιο 15 ^ο : Οντότητες χειρισμού τρισδιάστατων μοντέλων	303
Κεφάλαιο 16 ^ο : Δίνοντας κίνηση σε γεωμετρικά στερεά και οντότητες	325
Κεφάλαιο 17 ^ο : Χειρισμός στερεών και οντοτήτων με σκανδάλες	357
Κεφάλαιο 18 ^ο : Εφέ	371
Κεφάλαιο 19 ^ο : Ειδικά θέματα	399
Κεφάλαιο 20 ^ο : Διανομή της εφαρμογής στους τελικούς χρήστες	431
Βιβλιογραφία	443

Εισαγωγή

Τεχνολογία και εκπαίδευση

Διανύουμε μία περίοδο σημαντικών αλλαγών τόσο σε τεχνολογικό όσο και σε κοινωνικό επίπεδο. Η εκπαίδευση, θεμελιώδης ανθρώπινη δραστηριότητα, δύσκολα θα μπορούσε να μείνει ανεπηρέαστη από τις αλλαγές αυτές. Οι μορφές διδασκαλίας που στηρίζονται σε παραδοσιακές μεθόδους και τεχνικές, στη χρήση σχολικών βιβλίων και συμβατικών εποπτικών μέσων, τίθενται υπό σοβαρή αμφισβήτηση.

Η σύγχρονη παιδαγωγική αντίληψη εστιάζει στους τρόπους με τους οποίους οι μαθητές θα μπορούσαν να γίνουν ικανοί να εντοπίζουν, να ερμηνεύουν και να συνδυάζουν δημιουργικά τις πληροφορίες που προσλαμβάνουν από ετερόκλητες πηγές ώστε να απομονώνουν, να ορίζουν και να λύνουν προβλήματα. Με άλλα λόγια, η εκπαίδευση δίνει πλέον έμφαση στην απόκτηση δεξιοτήτων που είναι απαραίτητες στην οικοδόμηση και την αποτελεσματική εφαρμογή των γνώσεων που αποκτήθηκαν.

Όμως οι αλλαγές δεν περιορίζονται στη μεταβολή στόχων και μεθόδων. Ο όρος “εκπαίδευση” αποκτά μία διευρυμένη έννοια. Δεν αντιμετωπίζεται σαν κάτι τοπικά περιορισμένο ή σαν κάτι εντοπισμένο σε μία συγκεκριμένη φάση της ζωής του ανθρώπου. Επιμηκύνεται σε χρονική διάρκεια, επαναλαμβάνεται και εμπλουτίζεται σε χρονικές περιόδους πέρα από την καθιερωμένη σχολική ηλικία, γίνεται “διά βίου εκπαίδευση”. Απελευθερώνεται από τα στενά όρια μίας αίθουσας διδασκαλίας: πηγή πληροφόρησης και μάθησης είναι εν δυνάμει κάθε τόπος και κάθε μέσο. Νέα μέσα και τεχνολογίες, κυρίως από το χώρο της Πληροφορικής, έρχονται να ικανοποιήσουν τις παραπάνω ανάγκες και σταδιακά ενσωματώνονται στην καθημερινή εκπαιδευτική πρακτική, όπως η από απόσταση εκπαίδευση, το Διαδίκτυο και η χρήση πολυμέσων.

Αυτή η σταδιακή ενσωμάτωση των τεχνολογικών εξελίξεων στην εκπαίδευση είναι αργή και μπορεί να προκαλέσει προβλήματα. Η αδιάκοπη και ταχύτατη τεχνολογική εξέλιξη παρέχει σε κάθε τομέα των ανθρώπινων δραστηριοτήτων νέα μέσα ή κάνει πιο προσιτά ήδη υπάρχοντα, ανεξάρτητα από το εάν αυτά αξιοποιούνται άμεσα. Υπάρχουν τομείς που εκμεταλλεύονται πολύ σύντομα τους τεχνολογικούς καρπούς, όπως για παράδειγμα η ιατρική. Αλλά και η καθημερινή μας ζωή κάθε άλλο παρά ανεπηρέαστη μένει, για παράδειγμα σε λιγότερο από μία δεκαετία το κινητό τηλέφωνο έγινε αναπόσπαστο κομμάτι της καθημερινότητάς μας.

Η εκπαίδευση αντίθετα διακρίνεται από ένα μεγάλο βαθμό αδράνειας, είναι ιδιαίτερα δυσκίνητη στην προσαρμογή στα νέα δεδομένα. Δε θα ήταν υπερβολή να πούμε ότι απ’ όλες τις ανθρώπινες δραστηριότητες ενστερνίζεται τις εξελίξεις με ιδιαίτερα χαμηλούς ρυθμούς. Τι συνέπειες έχει κάτι τέτοιο; Αργά ή γρήγορα η καθημερινότητα, αυτό που ισχύει έξω από τις αίθουσες διδασκαλίας, βρίσκεται σε ασυμφωνία με την εκπαι-

δευτική πρακτική, τους τρόπους και τα μέσα που χρησιμοποιούνται μέσα στις αίθουσες διδασκαλίας. Αρκεί να αναφέρουμε ότι χρειάστηκαν χρόνια για την αλλαγή των από καιρό ξεπερασμένων σχολικών βιβλίων.

Η εισαγωγή της Πληροφορικής στην εκπαίδευση είναι ο τομέας εκείνος με τη μεγαλύτερη υστέρηση. Μεγάλο χρονικό διάστημα πέρασε για τον ανεπαρκή εξοπλισμό των σχολείων με ηλεκτρονικούς υπολογιστές. Και ενώ δε συζητάμε ακόμα για την πλήρη και συστηματική αξιοποίηση των δυνατοτήτων της Πληροφορικής στο σχολικό περιβάλλον, οι εξελίξεις τρέχουν, με αποτέλεσμα η προσαρμογή του σχολείου στα νέα δεδομένα να γίνεται ολοένα και πιο δύσκολη.

Αυτή ακριβώς η κατάσταση είναι που μας παρακινεί να αναζητούμε διαρκώς τρόπους με τους οποίους σύγχρονα τεχνολογικά μέσα μπορούν, έπειτα από αξιολόγηση, να αξιοποιηθούν άμεσα στη μαθησιακή διαδικασία.

Ένα από τα μέσα που βρίσκεται στην αιχμή των τεχνολογικών εξελίξεων είναι η *Εικονική Πραγματικότητα* (*Virtual Reality*, Ε.Π.) και γενικότερα τα *τριδιάστατα γραφικά* (*3D Graphics*). Παρά το ότι εφαρμογές αυτού του είδους χρησιμοποιούνται ήδη σε πολλούς τομείς (από τα κινηματογραφικά εφέ μέχρι τα σύγχρονα παιχνίδια στους ηλεκτρονικούς υπολογιστές) και παρά το ότι θεωρείται πως μπορούν να προσφέρουν σημαντική υποστήριξη στις νέες αντιλήψεις που διαμορφώνονται για τη μάθηση και την εκπαίδευση, ελάχιστα –έως καθόλου– έχουν περάσει στην εκπαιδευτική πρακτική. Οι λόγοι που συμβαίνει αυτό είναι σύνθετοι και τα εμπόδια πολλά, αλλά πιστεύουμε όχι αξεπέραστα.

Όπως συμβαίνει με κάθε νέο μέσο (τεχνολογικό ή μη) πριν γίνει ευρύτερα αποδεκτό και πριν εφαρμοστεί στο σχολικό περιβάλλον, υπάρχει μία σειρά από ζητήματα που πρέπει να ληφθούν υπόψη, να εξεταστούν και να αντιμετωπιστούν. Επειδή η εκπαιδευτική χρήση της Ε.Π. σε παιδιά μικρής ηλικίας είναι σχετικά πρόσφατη και ως εκ τούτου όχι πλήρως τεκμηριωμένη, υπάρχουν ορισμένα ουσιαστικά ερωτήματα που πρέπει να απαντηθούν, όπως για παράδειγμα: πού υπερτερεί η Ε.Π. σε σχέση με άλλα μέσα, ποιο είναι το θεωρητικό υπόβαθρο που στηρίζει εφαρμογές Ε.Π. στην εκπαίδευση, σε ποιους επιμέρους τομείς της εκπαίδευσης μπορεί να εφαρμοστεί, τι προβλήματα (τεχνικά, οικονομικά, κτλ.) πρέπει να ξεπεραστούν, ποιες εναλλακτικές λύσεις υπάρχουν, κ.ά.

Πρέπει να σημειώσουμε ότι κατά την άποψή μας δεν είναι αρκετό μία τεχνολογική εκπαιδευτική καινοτομία να είναι απλά και μόνο αποτελεσματική. Χρειάζεται να ικανο-

ποιούνται τρεις ακόμα προϋποθέσεις ώστε να ισχυροποιηθούν τα επιχειρήματα υπέρ της χρήσης της:

- Πρέπει να μπορεί να χρησιμοποιηθεί από μεγάλο αριθμό μαθητών, αν είναι δυνατόν απ' όλους. Δεν μπορεί να διαχωρίζουμε τους μαθητές σε προνομιούχους (σε αυτούς που έχουν πρόσβαση στην καινοτομία) και σε μη προνομιούχους. Σ' ένα δημοκρατικό σχολείο επιβάλλεται όλοι οι μαθητές να έχουν πρόσβαση σ' όλα τα μέσα.
- Πρέπει να μπορεί να εφαρμοστεί άμεσα. Η τεχνολογία πρέπει να είναι αρκετά ώριμη ώστε να μην υπάρχουν ανυπέρβλητα εμπόδια στη χρήση της. Παράλληλα, δεν πρέπει να απαιτεί εξειδικευμένες υποδομές που χρειάζονται χρόνο για την υλοποίησή τους. Ούτε πρέπει να απαιτείται μακρόχρονη εκπαίδευση των εκπαιδευτικών στη χρήση τόσο του υλικού όσο και του λογισμικού.
- Το οικονομικό κόστος πρέπει να κινείται σε λογικά πλαίσια, αν και το αρχικό κόστος της μαζικής εφαρμογής μιας τεχνολογικής καινοτομίας είναι συνήθως μεγάλο. Πρόβλεψη θα πρέπει να γίνει επίσης και για τη συντήρηση, την ανανέωση και για τις αναβαθμίσεις που ίσως θα χρειαστεί να γίνουν μελλοντικά. Πρέπει να αναζητηθούν τρόποι συμπίεσης του κόστους που συνεπάγεται καθένας από τους παραπάνω τομείς.

Στα κεφάλαια που ακολουθούν επιχειρείται να απαντηθούν τα παραπάνω ερωτήματα. Κυρίως όμως επιχειρείται να δοθούν λύσεις, έχοντας πάντα κατά νου τις προϋποθέσεις που θέσαμε.

Στόχοι και δομή του βιβλίου

Το βιβλίο απευθύνεται σε εκείνους που ενδιαφέρονται για την εισαγωγή στην εκπαίδευση σύγχρονων καινοτομιών της Πληροφορικής και ειδικά εφαρμογών Ε.Π. Επιδιώκει αφενός να αναλύσει το θεωρητικό πλαίσιο που στηρίζει την εισαγωγή της Ε.Π. στην εκπαιδευτική πράξη, αφετέρου να βοηθήσει αυτούς που αναζητούν ένα σημείο εκκίνησης, παρέχοντάς τους χρήσιμα στοιχεία για να κάνουν τα απαραίτητα σχεδιαστικά, οργανωτικά και τεχνικά βήματα που απαιτεί η υλοποίηση μίας εφαρμογής Ε.Π. για εκπαιδευτική χρήση.

Στα πλαίσια των παραπάνω στόχων, εκτός από την εισαγωγή, κρίθηκε σκόπιμο το βιβλίο να είναι χωρισμένο σε δύο μέρη:

Στο πρώτο μέρος, που αποτελείται από τα κεφάλαια 1 έως και 7, αναπτύσσεται ένας γενικότερος προβληματισμός για τη σύγχρονη εκπαίδευση και τους στόχους της, τεκμηριώνεται στα πλαίσια των θεωριών μάθησης η χρήση της Ε.Π. για εκπαιδευτικούς σκοπούς, εξετάζονται οι τρόποι μέσω των οποίων αυτή συνεισφέρει στην υλοποίηση

διδακτικών στόχων, αναπτύσσονται οι δυνατότητες που προσφέρει ως εκπαιδευτικό μέσο σε σύγκριση με άλλα, επισημαίνονται τα τεχνικά θέματα που συνοδεύουν τις εφαρμογές της και προτείνονται λύσεις.

Στο δεύτερο μέρος, που αποτελείται από τα κεφάλαια 8 έως και 20, παρουσιάζεται ένα από τα πιο απλά προγράμματα κατασκευής τρισδιάστατων περιβαλλόντων και εφαρμογών Ε.Π., το Reality Factory. Με αυτό τον τρόπο ο αναγνώστης μπορεί να γνωρίσει και να σχηματίσει ίδια αντίληψη για τις γνώσεις που απαιτούνται στη χρήση αυτού του είδους των προγραμμάτων, τις δυσκολίες που μπορεί να συναντήσει, τους τρόπους με τους οποίους μπορεί να τις αντιμετωπίσει, αλλά και τις δυνατότητες που προσφέρουν οι εφαρμογές Ε.Π.

Συνοπτικά, το περιεχόμενο του κάθε κεφαλαίου έχει ως εξής:

1^ο Μέρος

Κεφάλαιο 1^ο: Παρουσιάζονται οι σύγχρονες θεωρίες μάθησης, η θεωρία του Piaget, του Vygotsky και άλλες, συζητούνται διάφορες διδακτικές μέθοδοι και ποιος μπορεί να είναι ο ρόλος της Πληροφορικής σε αυτές.

Κεφάλαιο 2^ο: Επισημαίνονται οι παράγοντες της εκπαιδευτικής διαδικασίας, ορίζονται τρεις βασικές αρχές μάθησης και αναλύονται οι γενιές της εκπαιδευτικής χρήσης των υπολογιστών σε σχέση με τις αρχές μάθησης.

Κεφάλαιο 3^ο : Αναλύονται τα βασικά στοιχεία και χαρακτηριστικά της Ε.Π. Αναφέρονται τα είδη της και οι κυριότερες συσκευές υλοποίησής της. Επίσης αναφέρονται θεωρίες μάθησης που σχετίζονται με την Ε.Π.

Κεφάλαιο 4^ο: Αναλύονται τεχνικά θέματα υλικού και λογισμικού της Ε.Π. Αναφέρονται οι δυνατότητες των καρτών γραφικών και ερευνώνται οι ιδιότητες, οι κατηγορίες και οι προδιαγραφές των λογισμικών κατασκευής εικονικών κόσμων.

Κεφάλαιο 5^ο: Μελετώνται τα παιχνίδια βολών πρώτου προσώπου. Αναλύονται τα χαρακτηριστικά, οι εφαρμογές των παιχνιδιών αυτών, οι ομοιότητες και οι διαφορές τους με τα προγράμματα κατασκευής εφαρμογών Ε.Π.

Κεφάλαιο 6^ο: Μελετώνται τα χαρακτηριστικά των μηχανών παιχνιδιών και συγκεκριμένα η διαχείριση ήχου, η δικτυακή υποστήριξη, η τεχνητή νοημοσύνη, το περιβάλλον διεπαφής, η διαχείριση όπλων και η διαχείριση συμβάντων.

Κεφάλαιο 7^ο: Τεκμηριώνονται οι λόγοι για τους οποίους οι μέχρι τώρα εφαρμογές της Πληροφορικής στην εκπαίδευση αποτελούν απλώς συνέχεια των συμβατικών μορφών διδασκαλίας και οι λόγοι για τους οποίους η Ε.Π. θα φέρει την πραγματική επανάσταση όσον αφορά την εκπαιδευτική τεχνολογία.

2^ο Μέρος

Κεφάλαιο 8^ο: Γίνεται μία αρχική παρουσίαση του προγράμματος Reality Factory. Αναλύεται η εγκατάστασή του, η δομή των αρχείων του, καθώς επίσης οι ρυθμίσεις, τα πιθανά προβλήματα που μπορεί να προκύψουν και η αντιμετώπισή τους.

Κεφάλαιο 9^ο: Αναλύεται η μεθοδολογία κατασκευής και προσαρμογής εικόνων ώστε αυτές να αποτελέσουν υφές τρισδιάστατων μοντέλων ή στοιχείων των τρισδιάστατων κόσμων. Παρουσιάζεται ο τρόπος κατασκευής πακέτων υφών.

Κεφάλαιο 10^ο: Εξετάζονται τα τρισδιάστατα μοντέλα-Actors. Αναλύεται το πρόβλημα της κίνησης των μοντέλων. Παρουσιάζεται το πρόγραμμα κατασκευής τρισδιάστατων μοντέλων Milkshape.

Κεφάλαιο 11^ο: Παρουσιάζεται το πρόγραμμα RFEEditPro, ο επεξεργαστής κόσμων του Reality Factory. Αναλύονται οι βασικές επιλογές των γραμμών εργαλείων και εντολών. Δίνονται τα βήματα για τη δημιουργία ενός βασικού επιπέδου.

Κεφάλαιο 12^ο: Παρουσιάζονται οι ιδιότητες των στερεών σωμάτων στο πρόγραμμα. Επίσης παρουσιάζονται ιδιαίτερα περιβάλλοντα, όπως νερό, πάγος κτλ. Αναλύεται ο χειρισμός των υφών στα στερεά.

Κεφάλαιο 13^ο: Μελετάται η μεθοδολογία κατασκευής ουρανού και ορίζοντα. Αναλύονται σε βάθος οι οντότητες Skybox και Skydome.

Κεφάλαιο 14^ο: Μελετώνται οι οντότητες που σχετίζονται με το φως και τον ήχο στο πρόγραμμα Reality Factory.

Κεφάλαιο 15^ο: Παρουσιάζονται οι οντότητες χειρισμού των τρισδιάστατων μοντέλων.

Κεφάλαιο 16^ο: Παρουσιάζεται η χρήση και η σύνταξη εντολών για τον έλεγχο της κίνησης στερεών και τρισδιάστατων μοντέλων (κίνηση, περιστροφή, πήδημα κλπ).

Κεφάλαιο 17^ο: Αναλύεται ο χειρισμός στερεών και οντοτήτων με σκανδάλες και σύνθετους διακόπτες.

Κεφάλαιο 18^ο: Αναλύονται τα οπτικά εφέ που βοηθούν στον εντυπωσιασμό του χρήστη, όπως για παράδειγμα ηλεκτρικές εκκενώσεις, εκρήξεις, ομίχλη, εναλλασσόμενες εικόνες, βροχή και άλλα.

Κεφάλαιο 19^ο: Εξετάζονται ειδικά θέματα του προγράμματος, όπως τηλεμεταφορά, μηνύματα στην οθόνη, δημιουργία ανάγλυφου, αλλαγή επιπέδου, διάλογοι και άλλα.

Κεφάλαιο 20^ο: Εξετάζονται θέματα όπως η διαχείριση των αρχείων της τελικής εφαρμογής, η δημιουργία αρχείων εγκατάστασης και απεγκατάστασης.

Φιλοδοξία μας είναι ο αναγνώστης να εξοικειωθεί με τις εφαρμογές Ε.Π., να διαπιστώσει την ελκυστικότητα και την αποτελεσματικότητά τους και κυρίως να αισθανθεί έτοιμος να αναπτύξει δικές του εφαρμογές, αξιοποιώντας το ηλεκτρονικό περιβάλλον “καθημερινής χρήσης”, το οποίο ήδη αποτελεί μέρος του σύγχρονου σχολείου.

Κεφάλαιο 1^ο

Σύγχρονες θεωρίες μάθησης και ο ρόλος της Πληροφορικής

Θα ξεκινήσουμε την ανάλυσή μας απαντώντας σε ορισμένα θεμελιώδη ερωτήματα. Ας υποθέσουμε ότι θέλουμε να διδάξουμε ένα αντικείμενο όπου το σημαντικό στοιχείο δεν είναι μόνο η απόκτηση γνώσεων, αλλά θέλουμε οι μαθητές μας να αποκτήσουν και κάποιες δεξιότητες και συμπεριφορές. Ή ακόμα, ας υποθέσουμε ότι θέλουμε να εξασφαλίσουμε με τον καλύτερο δυνατό τρόπο την κατανόηση από τους μαθητές ενός γνωστικού αντικείμενου. Ανεξάρτητα λοιπόν από τα μέσα και τις διδακτικές μεθόδους που μπορούμε να χρησιμοποιήσουμε, πώς θα γίνει αποτελεσματικότερη η διδασκαλία ενός τέτοιου αντικείμενου; Ποια διδακτικά σχήματα προκρίνονται σ' αυτή την περίπτωση; Σε ποιο βαθμό εμπλέκονται εφαρμογές της Πληροφορικής, τι είδους και με πόση επιτυχία;

Είναι γενικώς παραδεκτό ότι σε σύγκριση με συμβατικές μεθόδους, οι μαθητές αναπτύσσουν σε μεγαλύτερο βαθμό τις προσδοκώμενες και βελτιώνουν τις ήδη υπάρχουσες γνώσεις, δεξιότητες και συμπεριφορές, όταν αναλάβουν ενεργό ρόλο και όταν τους παρέχουμε *εμπράγματα εμπειρίες* (*hands-on experiences*). Αν μάλιστα το αντικείμενο το επιτρέπει, μπορούμε να πούμε ότι ιδιαίτερα θετικά αποτελέσματα έχουμε όταν η εκπαιδευτική διαδικασία περιλαμβάνει πρακτική εξάσκηση σε περιβάλλοντα που προσομοιώνουν πραγματικές καταστάσεις ή –ακόμα καλύτερα– όταν υπάρχει εξάσκηση σε πραγματικές καταστάσεις. Είναι λοιπόν απαραίτητο να αναλυθεί η θεωρητική βάση της παραπάνω πρότασης, ώστε να κατανοηθεί καλύτερα το “γιατί” είναι αποτελεσματική η πρακτική εξάσκηση και η ενεργός δράση του μαθητή.

1.1 Η θεωρία του Piaget

Για τον Piaget, η γνώση είναι αποτέλεσμα μίας εξελικτικής διαδικασίας κατασκευής εσωτερικών δομών, που κατευθύνεται συνεχώς προς την “καλύτερη” αναπαράσταση του κόσμου που περιβάλλει το άτομο. Βασική δομική μονάδα αυτής της διαδικασίας είναι αυτό που ο Piaget ονομάζει *σχήμα* (*scheme*), μία γνωστική μονάδα που αναπαριστά τα βασικά χαρακτηριστικά μίας έννοιας, μίας κατάστασης ή ενός αντικείμενου, καθώς επίσης των ενεργειών που το αφορούν, μία μορφή πλάνου δράσης [1].

Η μάθηση συμβαίνει με τη λειτουργία τριών αλληλοσχετιζόμενων διαδικασιών. Της *αφομοίωσης* (*assimilation*), της *συμμόρφωσης* (*accommodation*) και της *εξισορρόπησης* (*equilibration*). Με την *αφομοίωση* γίνεται το ταίριασμα και η ενσωμάτωση γεγονότων που συμβαίνουν στο περιβάλλον στα προϋπάρχοντα *σχήματα*. Με τη *συμμόρφωση* τα προϋπάρχοντα *σχήματα* αλλάζουν λίγο ώστε να ταιριάζουν καλύτερα με νέες καταστάσεις ή εμπειρίες και έτσι ενσωματώνεται στην υπάρχουσα γνώση και η νέα εμπειρία.

Η *αφομοίωση* και η *συμμόρφωση* συμπληρώνουν η μία την άλλη, με την έννοια ότι παρέχουν τα μέσα μεταβολής των γνωστικών δομών ώστε αυτές να λαμβάνουν υπόψη τους τις νέες πληροφορίες. Συνιστούν ένα απλό σύστημα ανατροφοδότησης: αδυναμία να εφαρμοστεί ένα υπάρχον *σχήμα* με επιτυχία σε μία κατάσταση οδηγεί στην αλλαγή του τρόπου δράσης απέναντί της μέχρι να αντιμετωπιστεί με επιτυχία, οπότε το *σχήμα* αλλάζει για να περιλάβει τα νέα δεδομένα, και το δίδυμο *αφομοίωσης-συμμόρφωσης* έρχεται σε ισορροπία, που είναι η λειτουργία της *εξισορρόπησης*. Η *εξισορρόπηση* μπορεί να θεωρηθεί ως μία διαρκής πίεση για τη διατήρηση της αντιστοίχισης του κόσμου έτσι όπως τον “γνωρίζει” ένα άτομο και του κόσμου έτσι όπως τον βιώνει με τις εμπειρίες του.

Υπάρχουν όμως και οι περιπτώσεις που ένα *σχήμα* δεν ταιριάζει καθόλου με μία κατάσταση, που η *συμμόρφωση* έχει αποτύχει και πρέπει να γίνουν μεγάλες αλλαγές σ' αυτό. Τότε η *εξισορρόπηση* ενεργοποιεί τη *διαφοροποίηση* (*differentiation*) και το *συντονισμό* (*coordination*). Με τη *διαφοροποίηση* ένα υπάρχον *σχήμα* μετατρέπεται σε νέα *σχήματα*, ώστε να υπάρχει διαχωρισμός των ενεργειών για καταστάσεις που μέχρι εκείνη τη στιγμή θεωρούνταν παρόμοιες. Ο *συντονισμός*, από την άλλη, είναι η δημιουργία δομών σχημάτων ώστε να τονίζεται η σχέση και οι ομοιότητες μεταξύ καταστάσεων ή αντικειμένων που μέχρι εκείνη τη στιγμή δεν είχαν γίνει αντιληπτές.

Ένα παράδειγμα *συμμόρφωσης* είναι το εξής: Το παιδί πρέπει να περάσει ένα δρόμο με τον οποίο είναι απόλυτα εξοικειωμένο, αλλά επειδή έχει χιονίσει, τα δεδομένα έχουν αλλάξει λίγο. Όλες οι παράμετροι παραμένουν οι ίδιες και ταιριάζουν μ' ένα υπάρχον *σχήμα*, αλλά το χιόνι αναγκάζει το παιδί να αλλάξει τον τρόπο που περπατάει και το χρόνο που χρειάζεται να περάσει το δρόμο. Επιτυχημένη διάσχιση του δρόμου οδηγεί σε *συμμόρφωση* του *σχήματος*, ώστε να περιλαμβάνει τις αλλαγές στις ενέργειες εξαιτίας του καιρού, και σταδιακά το *σχήμα* μπορεί να αλλάξει, να γενικευτεί, ώστε να περιλαμβάνει όλες τις καιρικές συνθήκες.

Ένα παράδειγμα *διαφοροποίησης* και *συντονισμού* θα μπορούσε να έχει σχέση με τη δραματική αλλαγή στο πλάτος του δρόμου στο οποίο έχει συνηθίσει το παιδί, εξαιτίας μίας μετακόμισης από το χωριό στην πόλη. Σημαντικές αλλαγές στο χρόνο διάσχισης διαφόρων ειδών δρόμων μπορούν να οδηγήσουν στη *διαφοροποίηση* του αρχικού *σχήματος* σε αρκετά νέα *σχήματα*, που το καθένα περιλαμβάνει πληροφορίες για το πόση ώρα χρειάζεται η διάσχιση του κάθε συγκεκριμένου δρόμου, τι “κενό” στην κυκλοφορία είναι απαραίτητο, κτλ. Η εμπειρία του παιδιού στο πέρασμα διαφόρων δρόμων μπορεί να το οδηγήσει τελικά να δημιουργήσει με *συντονισμό* μία ενοποιημένη δομή *σχημάτων* η οποία να τονίζει τη σχέση μεταξύ του πλάτους των δρόμων και του χρόνου διάσχισής τους.

Αυτά τα παραδείγματα δεν πρέπει να θεωρηθεί ότι αποδίδουν την ακριβή διαδικασία απόκτησης δεξιοτήτων με βάση τη θεωρία του Piaget. Ωστόσο, φέρνουν στην επιφάνεια το ρόλο της δράσης μέσα στη μαθησιακή διαδικασία. Η θεωρία του Piaget δίνει την πρωτοκαθεδρία στη δράση σ' όλα τα επίπεδα της ανάπτυξης, η γνώση πάντα κατευθύνεται στη δράση και η δράση πάντα έχει ως αποτέλεσμα τη δόμηση γνώσης [2]. Με άλλα λόγια, η μάθηση είναι το αποτέλεσμα της ενεργού συμμετοχής του ατόμου σε κάποια μορφή έργου. Αυτό είναι από μόνο του αρκετό για να εξηγήσει το γιατί η πρακτική εξάσκηση είναι μία αποτελεσματική μέθοδος διδασκαλίας.

Τέλος, η θεωρία του Piaget δίνει ένα σαφές πλαίσιο κατανόησης του τρόπου ανάπτυξης των δεξιοτήτων (με τη *συμμόρφωση*, δηλαδή αλλάζοντας υπάρχοντα *σχήματα*, ή με τη *διαφοροποίηση*, δηλαδή με τη δημιουργία νέων *σχημάτων*) και της στρατηγικής που αξιοποιούνται οι δεξιότητες (για παράδειγμα μέσω του *συντονισμού*).

1.2 Η κοινωνικο-πολιτιστική θεωρία του Vygotsky

Piaget και Vygotsky φαίνεται ότι αντιμετωπίζουν τη διαδικασία της μάθησης με διαφορετικό τρόπο. Ο πρώτος θεωρεί ότι η μάθηση είναι ένα φαινόμενο εσωτερικής δόμησης γνώσεων, δεξιοτήτων και συμπεριφορών, ενώ ο δεύτερος τη θεωρεί αποτέλεσμα εσωτερίκευσης γνώσεων, δεξιοτήτων και συμπεριφορών που όμως έχουν προέλθει και δομηθεί εξωτερικά, με την κοινωνική αλληλεπίδραση. Ο Vygotsky μάλιστα τονίζει ότι κάθε λειτουργία στην κοινωνικο-πολιτιστική ανάπτυξη του παιδιού εμφανίζεται δύο φορές. Πρώτα σε κοινωνικό επίπεδο μεταξύ ατόμων και μετά σε ατομικό επίπεδο μέσα στο παιδί [3].

Τονίζει επίσης ότι όλες οι ανώτερες πνευματικές λειτουργίες πηγάζουν από την αλληλεπίδραση μεταξύ ατόμων [3]. Δύο σημεία-κλειδιά προέρχονται απ' αυτή τη βασική θέση: α) κάθε οργανωμένη σκέψη είναι το πνευματικό ισοδύναμο της δράσης και έχει την προέλευσή της από τη δράση και β) κάθε οργανωμένη δράση δομείται με τη βοήθεια των ατόμων που ήδη κατέχουν τα βασικά χαρακτηριστικά της.

Μία κεντρική ιδέα της θεωρίας του Vygotsky είναι η *ζώνη εγγύτερης ανάπτυξης* (*zone of proximal development*). Αυτή η ζώνη, που είναι ψυχολογικός και όχι φυσικός χώρος, ορίζεται ως η διαφορά μεταξύ του τι μπορεί να καταφέρει το παιδί μόνο του χωρίς βοήθεια και τι μπορεί να καταφέρει όταν εργάζεται με άλλους που έχουν περισσότερες γνώσεις, που είναι πιο ικανοί από το ίδιο [4]. Συνεπώς, η *ζώνη εγγύτερης ανάπτυξης* είναι ο "χώρος" μέσα στον οποίο λαμβάνει χώρα η μάθηση. Υπάρχει ένα άνω όριο στο τι μπορεί να καταφέρει το παιδί όταν εργάζεται με άλλους πιο έμπειρους. Δε θα υπάρξει μάθηση αν η δραστηριότητα είναι κάτω από το επίπεδο όσων ήδη μπορεί να κατα-

φέρει το παιδί μόνο του, αλλά και αν η δραστηριότητα είναι πολύ πάνω από τα όρια της *ζώνης εγγύτερης ανάπτυξης*.

Κλειδί για την ερμηνεία της εξελικτικής διαδικασίας της μάθησης, σύμφωνα με τον Vygotsky, είναι η ιδέα της *κυριότητας της γνώσης (mastery)*. Αν και το παιδί μπορεί να είναι ικανό για σύνθετες δραστηριότητες ή συμπεριφορές όταν εργάζεται με άλλους, η μάθηση θα συμβεί όταν το ίδιο θα είναι σε θέση από μόνο του να αναλάβει την ευθύνη γι' αυτές τις δραστηριότητες ή συμπεριφορές.

Για να γίνει κατανοητός ο τρόπος με τον οποίο συμβαίνει η εσωτερίκευση και η μάθηση, πρέπει να γίνει κατανοητός ο ρόλος της γλώσσας στη θεωρία του Vygotsky. Η κοινή δραστηριότητα, η αλληλεπίδραση ενός ικανού και ενός λιγότερο ικανού ατόμου, περιέχει και μία σειρά διαλόγων. Η γλώσσα χρησιμοποιείται για να έχει ο λιγότερο ικανός ανατροφοδότηση για τα λάθη του, για να στραφεί η προσοχή του στα σημαντικά σημεία, κτλ (διορθωτικός διάλογος). Όταν το λιγότερο ικανό άτομο είναι μόνο του, μπορεί να ανακαλέσει από τη μνήμη του τους διαλόγους που διαμείφθηκαν, να ακολουθήσει τις οδηγίες και να εκτελέσει τη σειρά των δράσεων που απαιτούνταν στη δραστηριότητα.

Η γλώσσα λειτουργεί ως ένας μηχανισμός μετάδοσης συμπεριφορών από ένα άτομο σ' ένα άλλο. Αυτό όμως δε συμβαίνει επειδή η γλώσσα είναι μία μηχανιστική μετάδοση οδηγιών, αλλά επειδή ενθυλακώνει μία συνειδητή και λογική σειρά σκέψεων. Αυτό που μεταφέρεται από το ικανό στο λιγότερο ικανό άτομο είναι το μέρος ενός ευρύτερου γνωστικού συστήματος.

Παρατηρείται λοιπόν ότι η δράση παίζει σημαντικό ρόλο και στη θεωρία του Vygotsky. Τα άτομα αλληλεπιδρούν μεταξύ τους, εργάζονται πάνω σ' ένα κοινό αντικείμενο και το περισσότερο ικανό άτομο μεταφέρει μέσω της γλώσσας γνώσεις, δεξιότητες και συμπεριφορές στο λιγότερο ικανό άτομο. Όταν η κοινή δράση βρίσκεται μέσα στα όρια της *ζώνης εγγύτερης ανάπτυξης* του λιγότερο ικανού ατόμου, τότε αυτό σταδιακά θα καταστεί ικανό και υπεύθυνο ώστε να αναλαμβάνει να εκτελεί το ίδιο έργο από μόνο του.

1.3 Η γνωστική προσέγγιση της επεξεργασίας των πληροφοριών

Σ' αυτή την προσέγγιση ανήκει μία πληθώρα ερευνών και θεωρητικών τεκμηριώσεων που δεν αποτελούν ενιαία θεωρία, αλλά έχουν ως κοινό σημείο ότι παραλληλίζουν τις πνευματικές λειτουργίες του ατόμου με τον τρόπο λειτουργίας ενός ηλεκτρονικού υπολογιστή. Το ενδιαφέρον τους επικεντρώνεται στο ποιες πνευματικές λειτουργίες χρη-

σιμοποιεί το άτομο για να χειρίζεται τις πληροφορίες, πώς αυτές οργανώνονται και πώς αλλάζουν κατά την ανάπτυξη.

Η εργασία των Schank και Abelson για τα *σενάρια* (*scripts*) [5] έχει ιδιαίτερο ενδιαφέρον, επειδή μπορεί να εξηγήσει τον τρόπο απόκτησης των διαφόρων δεξιοτήτων. Τα *σενάρια* είναι κωδικοποιημένες αναπαραστάσεις της γνώσης για επαναλαμβανόμενα γεγονότα. Με άλλα λόγια, τα *σενάρια* ορίζουν την πιθανή διαδοχή γεγονότων σε κάποιες συγκεκριμένες περιστάσεις. Κάθε περίπτωση που καλείται να αντιμετωπίσει ένα *σενάριο* αναπόφευκτα θα περιέχει παραλλαγές και αποκλίσεις από το βασικό του περίγραμμα, είναι δηλαδή κάτι πιο γενικό απ' ό,τι η ανάμνηση ενός και μόνο συγκεκριμένου γεγονότος. Ωστόσο, περιέχει επαρκείς λεπτομέρειες ώστε να αντιμετωπίζονται αυτές οι παραλλαγές. Συνεπώς, από τη μία επιτρέπει λογικά συμπεράσματα για την πορεία των γεγονότων ("μετά απ' αυτό, θα γίνει αυτό και μετά...") και από την άλλη παρέχει οδηγίες στο άτομο ποιες ενέργειες να κάνει μέσα σ' ένα συγκεκριμένο πλαίσιο ("θα κάνω αυτό, μετά αυτό και μετά...").

Ένα παράδειγμα *σεναρίου* μπορεί να είναι ο τρόπος που διασχίζονται οι διαβάσεις με φωτεινό σηματοδότη. Το άτομο φτάνει στη διάβαση, περιμένει να ανάψει το πράσινο για τους πεζούς, ελέγχει αν όλα τα αυτοκίνητα σταμάτησαν και τελικά διασχίζει το δρόμο. Το *σενάριο* του παραδείγματος έχει εφαρμογή σ' όλες τις αντίστοιχες διαβάσεις. Θα μπορούσε ακόμα να περιλαμβάνει την παραλλαγή όπου το άτομο χρησιμοποιεί το κουμπί στο φωτεινό σηματοδότη για το άναμμα του πράσινου για τους πεζούς.

Τα στοιχεία των ερευνών δείχνουν ότι τα *σενάρια* είναι φυσικοί και ισχυροί μηχανισμοί οργάνωσης των εμπειριών του ατόμου. Για παράδειγμα, ακόμα και παιδιά 3 ετών μπορούν να περιγράψουν με όρους *σεναρίου*, δηλαδή με αρκετή ακρίβεια, οργανωμένα και λογικά, τη διαδοχή γεγονότων για περιστατικά που συμβαίνουν συχνά [6] και για περιστατικά που συμβαίνουν σπάνια ή και μόνο μία φορά [7]. Τα *σενάρια*, ακόμα και σε παιδιά 5 ετών, μπορούν να περιλαμβάνουν υποθετικές καταστάσεις ("αν συμβεί αυτό, τότε θα κάνω αυτό..."), με αποτέλεσμα η πολυπλοκότητά τους να αυξάνει σημαντικά [8].

Κάποιος μπορεί να διακρίνει τη μεγάλη ομοιότητα μεταξύ των *σεναρίων* και των *σχημάτων* της θεωρίας του Piaget. Πρώτα απ' όλα και τα δύο είναι η διαδοχική σειρά δράσεων του ατόμου σ' ένα συγκεκριμένο πλαίσιο. Η εφαρμογή που έχει ένα *σενάριο* σε παρόμοιες καταστάσεις θυμίζει την *αφομοίωση*. Παρόμοια, η διαφοροποίηση υπαρχόντων, η δυνατότητα να περικλείονται υποθετικές καταστάσεις και η δημιουργία νέων *σεναρίων* αντιστοιχεί με τη *συμμόρφωση* και τη *διαφοροποίηση*. Η σύγκλιση των δια-

φορετικών θεωρητικών πλαισίων στις ίδιες δομές και τις ίδιες διαδικασίες παρέχει ισχυρές ενδείξεις για την ορθότητα και των δύο.

1.4 Σχολιασμός

Υπάρχουν δύο σημεία στα οποία αξίζει να γίνει αναφορά σε σχέση με τις θεωρίες μάθησης που αναπτύχθηκαν πιο πάνω. Το πρώτο αφορά σ' αυτό που θα μπορούσε να ονομαστεί *ελάχιστες συνθήκες για μάθηση*. Οι τρεις θεωρίες που παρουσιάστηκαν συγκλίνουν στην άποψη ότι η γνώση είναι μία φυσική εξέλιξη, μία ακολουθία γεγονότων που οδηγούν από τη δράση στην αντίληψη. Συνεπώς, οι διδακτικές μέθοδοι είναι περισσότερο αποτελεσματικές όταν λαμβάνουν υπόψη τους το σύνολο αυτής τη φυσικής ακολουθίας και όχι όταν καταπιάνονται με μερικά στοιχεία της. Οι μέθοδοι που στηρίζονται στην πρακτική εξάσκηση είναι αποτελεσματικές ακριβώς επειδή ξεκινούν από το σωστό σημείο της όλης διαδικασίας, από τη δράση.

Το δεύτερο σημείο σχετίζεται με τον συμπληρωματικό χαρακτήρα των θεωριών του Piaget και του Vygotsky. Η θεωρία του Piaget δεν εξηγεί με λεπτομέρεια τον τρόπο απόκτησης της μεγάλης ποικιλίας δράσεων που έχουν διαθέσιμες ακόμα και μικρά παιδιά, ενώ η θεωρία του Vygotsky έχει ένα καλά ορισμένο μηχανισμό γι' αυτό. Αντίθετα, η θεωρία του Vygotsky είναι πιο αδύναμη στην εξήγηση του πώς η δράση γενικεύεται σε ιδέες και έννοιες, που είναι το δυνατό σημείο της θεωρίας του Piaget. Δε θα ήταν άστοχο να ειπωθεί ότι οι δύο αυτές θεωρίες μπορούν να συνδυαστούν σε μία που να εξηγεί πολύ καλά τη διαδικασία μάθησης.

Έτσι, κάθε σοβαρή προσπάθεια να αναπτυχθούν στα παιδιά οι κατάλληλες γνώσεις, δεξιότητες και συμπεριφορές θα πρέπει να ξεκινά από συγκεκριμένες δράσεις στο φυσικό (ή σε αντίστοιχο του φυσικού) περιβάλλον όπου εφαρμόζονται αυτές οι γνώσεις, δεξιότητες και συμπεριφορές. Αυτό θα δώσει τις βασικές πληροφορίες για τη σχέση μεταξύ της δράσης και του περιβάλλοντος, τις οποίες τα παιδιά θα προσαρμόσουν και θα χρησιμοποιήσουν μόνα τους. Μόλις αρχίσουν να εδραιώνονται κάποια πρότυπα δράσης, αυτά μπορούν να χρησιμοποιηθούν με τη σειρά τους ως βασικές εμπειρίες πάνω στις οποίες θα δομηθεί η κατανόηση των συμπεριφορών που χρησιμοποιούνται σε διαφορετικές καταστάσεις-πλαίσια, οδηγώντας σε γενικεύσεις και στη δημιουργία νέων αντιλήψεων και εννοιών.

1.5 Διδακτικές μέθοδοι με βάση τις θεωρίες μάθησης

Θεωρητικά λοιπόν, η πρακτική εξάσκηση και οι εμπράγματα εμπειρίες είναι πιο αποτελεσματικές από τη διδασκαλία στην τάξη. Χρειάζεται όμως ένα πιο συγκεκριμένο διδακτικό πλαίσιο που να εφαρμόζει τα παραπάνω και να στηρίζεται στις διαπιστώσεις και τα συμπεράσματα από τις κυριότερες θεωρίες μάθησης.

Οι θεωρίες του Piaget και του Vygotsky προτείνουν μάλλον διαφορετικούς μηχανισμούς ενίσχυσης της ψυχοπνευματικής ανάπτυξης. Για τον Piaget σημαντικό στοιχείο είναι η σύγκρουση μεταξύ ιδεών και εμπειριών, ή καλύτερα μεταξύ των ιδεών που έχει ένα άτομο και των ιδεών που έχουν οι άλλοι και που αποκαλύπτεται μέσω του διαλόγου [9]. Υπάρχουν όμως ορισμένες προϋποθέσεις ώστε ο διάλογος να είναι αποτελεσματικός. Πρώτον, οι συμμετέχοντες πρέπει να έχουν διαφορετικές απόψεις. Δεύτερον, ο διάλογος μεταξύ ατόμων της ίδιας ηλικίας (για παράδειγμα μεταξύ παιδιών) είναι πιο αποτελεσματικός από το διάλογο μεταξύ παιδιού και ενήλικα. Άτομα της ίδιας ηλικίας χρησιμοποιούν την ίδια “γλώσσα” για να περιγράψουν καταστάσεις και αντικείμενα με μεγαλύτερη άνεση, αμφισβητούν το ένα το άλλο και συζητούν τις αντικρουόμενες απόψεις. Αλληλεπίδραση μεταξύ ενήλικα και παιδιού συχνά οδηγεί στη επιβολή των απόψεων του ενήλικα και το παιδί συναινεί είτε καταλαβαίνει είτε όχι [10].

Σε αντίθεση, για τον Vygotsky ο κεντρικός μηχανισμός μάθησης είναι η καθοδήγηση των δράσεων ενός λιγότερο έμπειρου ατόμου από ένα πιο έμπειρο. Ο διάλογος δηλαδή γίνεται μεταξύ ατόμων διαφορετικού δυναμικού. Βασική προϋπόθεση είναι βέβαια η δραστηριότητα να βρίσκεται εντός της *ζώνης εγγύτερης ανάπτυξης* του λιγότερο ικανού και να βασίζεται στις ήδη υπάρχουσες δυνατότητες του παιδιού.

Παρά αυτή τη σημαντική διαφορά, θα ήταν λάθος να ειπωθεί ότι η μία θεωρία αναιρεί την άλλη, λαμβάνοντας μάλιστα υπόψη το συμπληρωματικό χαρακτήρα τους. Και οι δύο θεωρίες τονίζουν τη σημασία της κοινωνικής αλληλεπίδρασης, της αλληλεπιδραστικής μάθησης. Εξαιτίας των διαφορετικών σημείων στα οποία εστιάζουν οι δύο θεωρίες, μπορεί με αρκετή ασφάλεια να θεωρηθεί ότι η προσέγγιση του Vygotsky είναι πιο κατάλληλη για την εκμάθηση δράσεων-δεξιοτήτων, ενώ η προσέγγιση του Piaget πιο κατάλληλη για την εκμάθηση εννοιών.

Για παράδειγμα, η απόκτηση πρακτικών δεξιοτήτων είναι εύλογο ότι συμβαίνει πιο φυσικά όταν υπάρχει ένα μοντέλο μαθητείας (έμπειρος και λιγότερο έμπειρος). Από την άλλη, η κατανόηση των αιτίων πίσω από ένα συμβάν, πίσω από μία δράση, συμβαίνει πιο φυσικά όταν στηρίζεται σε διάλογο πάνω σε συγκεκριμένα αντικείμενα και οι συμμετέχοντες έχουν λίγο διαφορετικές απόψεις μεταξύ τους.

Το σημαντικό είναι ότι και οι δύο θεωρίες ταιριάζουν καλά σε συγκεκριμένες διδακτικές μεθοδολογίες οι οποίες εντάσσονται στο γενικό πλαίσιο της “αλληλεπιδραστικής μάθησης”. Οι μεθοδολογίες αυτές είναι η *διδασκαλία μεταξύ συνομηλίκων* (*peer tutoring*), η *διδασκαλία με την καθοδήγηση ενήλικου* (*adult led tutoring*) και η *συνεργασία συνομηλίκων* (*peer collaboration*). Στη συνέχεια εξετάζονται οι παραπάνω τεχνικές.

1.5.1 Διδασκαλία μεταξύ συνομηλίκων

Αυτή η μορφή διδασκαλίας έχει ως βασική ιδέα την αλληλεπίδραση μεταξύ ατόμων της ίδιας ηλικίας που το ένα είναι λιγότερο ικανό από το άλλο. Αυτή η ασυμμετρία της σχέσης είναι για τον Vygotsky βασικός όρος στη θεωρία του, αν και από μόνος του δεν εγγυάται ότι αναπόφευκτα θα προκύψει μάθηση. Ωστόσο, η διδασκαλία μεταξύ ατόμων της ίδιας ηλικίας μπορεί να εξηγηθεί και με όρους της θεωρίας του Piaget. Το πιο έμπειρο μέλος μπορεί να προκαλεί σύγκρουση μεταξύ των *σχημάτων* και των γεγονότων στο περιβάλλον του λιγότερο ικανού, που τελικά οδηγεί στη γνωστική πρόοδο.

Στοιχεία όμως από έρευνα ζευγαρώματος ικανού και λιγότερο ικανού παιδιού και εργασίας επάνω σ' ένα κοινό αντικείμενο δε συνηγορούν σε κάτι τέτοιο. Από τη μελέτη της αλληλεπίδρασης και κυρίως των συζητήσεων μεταξύ των παιδιών φάνηκε ότι κυριαρχούσαν τα στοιχεία της παρατήρησης και της καθοδήγησης της δράσης και όχι η γνωστική διαμάχη που απαιτεί η θεωρία του Piaget [11]. Αντίθετα, η παρατήρηση αυτή είναι σύμφωνη με την ιδέα του Vygotsky για το “διορθωτικό διάλογο” κατά την αλληλεπίδραση μ' ένα περισσότερο ικανό άτομο.

Σε επισκόπηση της βιβλιογραφίας για τη διδασκαλία στην οποία μεσολαβεί δάσκαλος και για τη διδασκαλία μεταξύ συνομηλίκων, οι Greenwood, Carta και Kamps εντόπισαν περιορισμούς όσον αφορά τις δυνατότητες της δεύτερης να διδάξει ανώτερες αντιληπτικές δεξιότητες [12]. Η διδασκαλία συνομηλίκων βελτίωσε την ικανότητα κατανόησης γραπτού κειμένου, αλλά υπήρχε μεγάλη εξάρτηση από δεξιότητες όπως η ικανότητα ανάγνωσης, η ορθογραφία και ο ρυθμός ανάγνωσης. Αυτή η διαπίστωση είναι ευθυγραμμισμένη με την άποψη ότι η διδασκαλία συνομηλίκων ταιριάζει περισσότερο στη διδασκαλία εργασιών που έχουν κανόνες, που επικεντρώνονται στην ανταλλαγή πληροφοριών και όταν η δεξιότητα που απαιτείται έχει εν μέρει αποκτηθεί [13].

1.5.2 Διδασκαλία με την καθοδήγηση ενηλίκου

Η διδασκαλία με αλληλεπίδραση παιδιού-ενηλίκου βρίσκεται σε συμφωνία, όπως και στην προηγούμενη περίπτωση, με τη θέση του Vygotsky ότι ο κεντρικός μηχανισμός μάθησης είναι η καθοδήγηση των δράσεων ενός λιγότερο έμπειρου ατόμου από ένα πιο έμπειρο. Το δυνατό σημείο αυτού του τρόπου διδασκαλίας είναι ότι ο ενήλικας έχει πιο πολλές διδακτικές ικανότητες απ' ό,τι ένα παιδί. Για παράδειγμα, οι ενήλικες εντοπίζουν πιο εύκολα ότι ο εκπαιδευόμενος έχει καταλάβει κάτι λάθος, ανταποκρίνονται πιο καλά όταν αυτός ζητάει βοήθεια [14] και είναι πιο αποτελεσματικοί από τα παιδιά όταν διδάσκουν δεξιότητες τις οποίες γνωρίζουν πολύ καλά [15].

Στο πλαίσιο διδασκαλίας με πρακτική εξάσκηση, αυτό σημαίνει ότι ειδικά εκπαιδευμένοι ενήλικες που κατανοούν τις δυνατότητες του παιδιού και που μπορούν να προ-

σαρμόζουν τη συμπεριφορά τους με βάση τη *ζώνη εγγύτερης ανάπτυξης* του παιδιού, μπορεί να είναι ιδιαίτερα αποτελεσματικοί. Σύμφωνα με τον Wood, οι ενήλικες έχουν στη διάθεσή τους πέντε επίπεδα καθοδήγησης: α) γενικές προφορικές οδηγίες, β) συγκεκριμένες προφορικές οδηγίες, γ) κατάδειξη του σημαντικού υλικού, δ) προετοιμασία του υλικού για δράση και ε) επίδειξη της δράσης [16].

Οι αποτελεσματικοί εκπαιδευτές ξεκινούν τη διδασκαλία χρησιμοποιώντας τις πιο γενικές μορφές καθοδήγησης, για παράδειγμα γενικές προφορικές οδηγίες. Όταν ο διδασκόμενος συναντήσει δυσκολίες, εφαρμόζουν πιο ειδικές μορφές καθοδήγησης και εφαρμόζουν ξανά πιο γενικές όταν ο μαθητευόμενος ανταποκρίνεται με επιτυχία στο έργο. Είναι εύκολα κατανοητή η συμβατότητα του όλου σχήματος με τη *ζώνη εγγύτερης ανάπτυξης* του παιδιού.

1.5.3 Συνεργασία συνομηλίκων

Η συνεργασία συνομηλίκων αναφέρεται σ' ένα διδακτικό σχήμα όπου η αλληλεπίδραση μεταξύ των παιδιών δεν είναι τόσο δομημένη όπως στη διδασκαλία μεταξύ συνομηλίκων. Τα παιδιά εργάζονται μαζί (σε ζεύγη ή σε ομάδες μέχρι και έξι ατόμων) για να βρουν από κοινού λύση σ' ένα συγκεκριμένο πρόβλημα. Επίσης, αντίθετα με τη διδασκαλία μεταξύ συνομηλίκων, δεν είναι απαραίτητη η διαφορά στις δυνατότητες των ατόμων της ομάδας. Αν όμως οι διαφορές στις ικανότητες των μελών μίας ομάδας δεν είναι απαραίτητες, τότε η απόκτηση της γνώσης είναι δύσκολο να εξηγηθεί με όρους της θεωρίας του Vygotsky.

Πράγματι, σε μελέτη η οποία εξέτασε διαφορετικές ομάδες παιδιών που εργάζονταν σε μία κοινή δραστηριότητα, φάνηκε ότι δεν υπήρξε καμία διαφορά στην αξιολόγηση πριν και μετά από την εργασία, αλλά φάνηκε διαφορά σε αξιολόγηση που έγινε αρκετές εβδομάδες μετά, με τη διαφορά να μην μπορεί να αποδοθεί σε παρόμοιες εργασίες που έκαναν τα παιδιά στο μεσοδιάστημα [17]. Το αποτέλεσμα όμως μπορεί να εξηγηθεί με όρους της θεωρίας του Piaget, με την έννοια ότι υπήρξε σύγκρουση μεταξύ των *σχημάτων* και των εμπειριών από το περιβάλλον, που απαίτησε χρόνο για να εσωτερικευτεί ατομικά και για κάθε παιδί.

Ο ρόλος του διαλόγου, που είναι σημαντικό και αναπόσπαστο κομμάτι της συνεργασίας συνομηλίκων –όπως επίσης της διδασκαλίας που καθοδηγείται από ενήλικα και της διδασκαλίας μεταξύ συνομηλίκων–, μπορεί να εξηγηθεί με τη θεωρία του Piaget. Ο *συντονισμός* και η *διαφοροποίηση* εξαρτώνται από τη συνειδητή σκέψη και χρειάζεται να υπάρχει πρόσβαση του παιδιού στις διαφορετικές απόψεις που μπορεί να υπάρχουν για ένα θέμα. Με τη συζήτηση διαφόρων ιδεών στην ομάδα, στην ουσία γίνεται ανταλλαγή των απόψεων που έχει κάθε παιδί για το θέμα.

Οι διαφορετικές απόψεις οδηγούν, σε ατομικό επίπεδο πια, σε σύγκρουση μεταξύ των *σχημάτων* και των απόψεων των άλλων παιδιών. Όταν η συζήτηση είναι τακτική, είναι αρκετά ξεκάθαρο ποιοι *συντονισμοί* ή *διαφοροποιήσεις* μπορεί να γίνουν. Όταν η συζήτηση είναι άτακτη, όπως συμβαίνει τις περισσότερες φορές με τα παιδιά του δημοτικού σχολείου, οι απόψεις μπορεί να σχετίζονται ή να μη σχετίζονται με το θέμα και το παιδί θα πρέπει πρώτα να επιλέξει ποιες απόψεις είναι σχετικές.

Τέλος, φαίνεται ότι υπάρχουν συγκεκριμένα γνωστικά πεδία και δραστηριότητες που η διδασκαλία τους είναι αποτελεσματική όταν γίνεται με τη συνεργασία μεταξύ συνομηλίκων. Υπήρξαν καλύτερα αποτελέσματα όταν οι μαθητές χρειάστηκε να ασχοληθούν με την απόκτηση νέων ιδεών και εννοιών. Εργασίες που στηρίζονταν στην εκτέλεση συγκεκριμένων οδηγιών και την πιστή αναπαραγωγή δεξιοτήτων είχαν λιγότερο καλά αποτελέσματα [18]. Κατ' αυτή την έννοια, μπορεί να υποτεθεί και πάλι ότι η συνεργασία μεταξύ συνομηλίκων ταιριάζει περισσότερο με τις ιδέες του Piaget παρά του Vygotsky.

1.5.4 Συμπεράσματα για τις διδακτικές μεθόδους

Από τις τρεις μεθόδους που αναπτύχθηκαν, τις περισσότερες δυσκολίες στην εφαρμογή έχει η διδασκαλία μεταξύ συνομηλίκων. Ειδικά στις μικρότερες ηλικίες, είναι δύσκολο τα παιδιά να προσαρμοστούν στους ρόλους που η μέθοδος προτείνει (ένα παιδί διδάσκει και το άλλο μαθαίνει). Επιπλέον, ο διάλογος μεταξύ των παιδιών πρέπει να είναι δομημένος, με το ένα παιδί να εξηγεί και να αιτιολογεί και παράλληλα το σκεπτικό του να γίνεται κατανοητό και αποδεκτό από το λιγότερο ικανό.

Η διδασκαλία με την καθοδήγηση ενήλικα λύνει τα παραπάνω προβλήματα, με την έννοια ότι ο ενήλικας έχει τις απαραίτητες γνώσεις και δεξιότητες και έχει επίσης στη διάθεσή του αρκετές διδακτικές προσεγγίσεις για να καθοδηγήσει αποτελεσματικά το παιδί. Από την άλλη, υπάρχει το πρόβλημα ότι ενήλικας και παιδί δε μιλάνε την ίδια “γλώσσα”, με κίνδυνο ο ενήλικας να επιβάλει τελικά τις απόψεις του. Χρειάζεται λοιπόν προσεκτική προσέγγιση του παιδιού από τον ενήλικα, ώστε αυτό που πάει να διδάξει και ο τρόπος που το διδάσκει να βρίσκονται μέσα στα πλαίσια της *ζώνης εγγύτερης ανάπτυξης* του παιδιού.

Η συνεργασία συνομηλίκων λύνει το πρόβλημα της κοινής “γλώσσας”, αλλά φαίνεται ότι ως μέθοδος είναι καταλληλότερη για τη διδασκαλία ιδεών και εννοιών και όχι δεξιοτήτων. Επιπλέον, ο διάλογος μεταξύ των παιδιών πρέπει να είναι επικεντρωμένος στην εργασία με την οποία ασχολούνται. Τέλος, βασική προϋπόθεση είναι τα παιδιά να έχουν διαφορετικές απόψεις αλλά όχι διαφορά δυναμικού, γεγονός που γεννά το θέμα του τρόπου σχηματισμού των ομάδων και της επιλογής των παιδιών που θα

συμμετέχουν σε κάθε ομάδα. Το κάθε παιδί θα πρέπει να είναι σε θέση να συνεισφέρει με τον τρόπο του και ταυτόχρονα όλη η ομάδα να συμμετέχει ενεργά στο έργο που της έχει ανατεθεί.

Παρά τα οργανωτικά και διδακτικά προβλήματα των μεθόδων αυτών, είναι απαραίτητο να τονιστεί ότι πλεονεκτούν σε σχέση με τις παραδοσιακές, δασκαλοκεντρικές μεθόδους διδασκαλίας. Η εργασία σε ζεύγη ή ομάδες επιτρέπει καλύτερη προσαρμογή του διδακτικού υλικού στις ανάγκες του κάθε παιδιού, ενώ με τις παραδοσιακές μεθόδους ο τρόπος παρουσίασής της ύλης γίνεται λαμβάνοντας υπόψη το μέσο όρο της τάξης. Ο δάσκαλος μπορεί να διαθέτει το χρόνο του περισσότερο αποτελεσματικά επισκεπτόμενος την κάθε ομάδα και παρέχοντας καθοδήγηση σε ατομική ή ομαδική βάση.

Από τη σκοπιά των μαθητών, οι αλληλεπιδραστικές μέθοδοι διδασκαλίας ενθαρρύνουν τα παιδιά να συμμετέχουν στη διδασκαλία και να εργάζονται μαζί. Παίρνουν έτσι την ευθύνη για την πορεία της μάθησής τους, παίρνουν αποφάσεις, βοηθούν τους άλλους, μοιράζονται ιδέες και πληροφορίες και προάγουν την εξέλιξη όλης της ομάδας.

Ένας συνδυασμός διδασκαλίας με την καθοδήγηση ενήλικα και συνεργασίας συνομηλίκων –μία “καθοδηγούμενη συνεργασία”– δείχνει να είναι η πιο πολλά υποσχόμενη διδακτική μέθοδος. Αυτή η μέθοδος μπορεί να διαχωριστεί σε δύο σκέλη, σ' ένα πρακτικής εξάσκησης σε πραγματικό ή προσομοιωμένο περιβάλλον και σ' ένα διδασκαλίας μέσα στην τάξη.

Η αλληλεπίδραση ενήλικα-παιδιού μπορεί να χρησιμοποιηθεί για την ανάπτυξη πρακτικών και δεξιοτήτων που κλιμακώνονται σε δυσκολία. Εκτός των άλλων, κατά την πρακτική εξάσκηση η παρουσία ενήλικα είναι απαραίτητη για λόγους ασφάλειας. Επίσης, δεν υπάρχει κάποιος λόγος να μην εφαρμόζεται και κατά την πρακτική εξάσκηση ομαδική εργασία και διάλογος μεταξύ των παιδιών πάνω στο αντικείμενο με το οποίο ασχολούνται, με το δάσκαλο να μπαίνει πρόσκαιρα στο περιθώριο με σκοπό την ενίσχυση της ομαδικής εργασίας.

Με την ολοκλήρωση της πρακτικής εξάσκησης, η ομαδική εργασία μπορεί να συνεχίζεται στην τάξη. Σκοπός της θα είναι να διευρυνθεί η αντίληψη του παιδιού πέρα από τις συγκεκριμένες καταστάσεις που αυτό αντιμετώπισε στην πρακτική εξάσκηση, με άλλα λόγια να γενικευτεί και σε άλλες αντίστοιχες περιπτώσεις.

Σ' αυτό το σημείο ο ρόλος της πρακτικής εξάσκησης είναι καθοριστικός. Πρώτον, δάσκαλος και μαθητές μπορούν να μιλάνε μία κοινή “γλώσσα” με βάση τις κοινές εμπειρίες που είχαν. Δεύτερον, το είδος της πρακτικής εξάσκησης καθορίζει το είδος των

δραστηριοτήτων της ομαδικής εργασίας. Το περιεχόμενο και των δύο πρέπει να είναι άμεσα σχετιζόμενο για να αυξηθούν οι πιθανότητες γενίκευσης της εμπειρίας και εφαρμογής της σε νέα γνωστικά πλαίσια. Με τον καιρό, με διαδοχικές μεταβάσεις “από το ειδικό στο γενικό”, το διδακτικό μοντέλο μπορεί να γενικευτεί σε τέτοιο βαθμό, ώστε να περιλαμβάνει εξαιρετικά περίπλοκες καταστάσεις.

1.6 Διδασκαλία με τη χρήση προσομοιώσεων σε ηλεκτρονικούς υπολογιστές

Ας εξετάσουμε μία ιδιαίτερα δύσκολη περίπτωση όπου από τη μία είναι αυταπόδεικτη η χρησιμότητα της πρακτικής εξάσκησης και του ενεργού ρόλου των μαθητών και από την άλλη υπάρχουν ορισμένα αξεπέραστα εμπόδια. Αυτή είναι η περίπτωση της διδασκαλίας της *Κυκλοφοριακής Αγωγής* (Κ.Α.). Στην Κ.Α. απαιτείται συστηματική εξάσκηση των παιδιών σε δεξιότητες και συμπεριφορές που η απλή διδασκαλία στην τάξη από μόνη της αδυνατεί να προσφέρει.

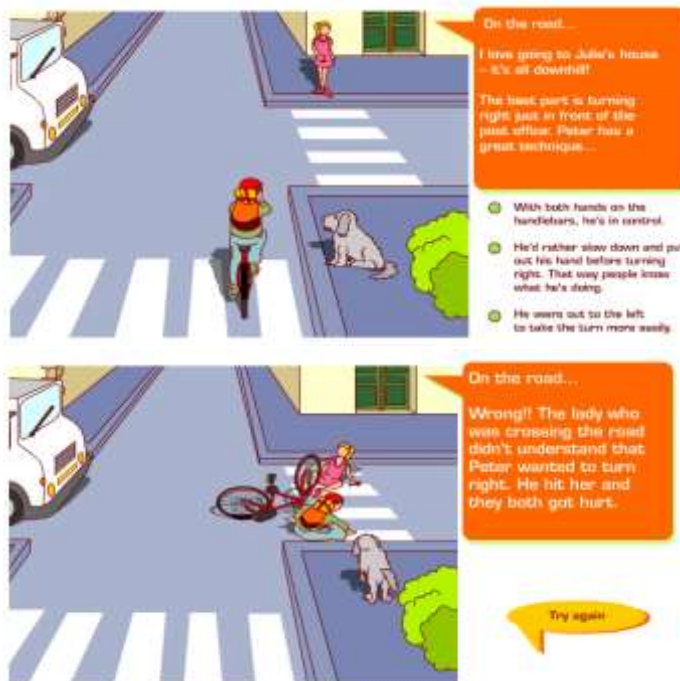
Υπάρχουν όμως σοβαροί περιορισμοί. Η πρακτική εξάσκηση προϋποθέτει άνεση χρόνου, μετακινήσεις μαθητών και κατάλληλους χώρους. Όμως ούτε ο χρόνος είναι διαθέσιμος ούτε συχνές μετακινήσεις μπορούν να γίνουν ούτε κατάλληλοι χώροι είναι εύκολο να βρεθούν. Επιπρόσθετα, η πρακτική εξάσκηση περιορίζεται από παράγοντες όπως ο καιρός, απαιτεί σημαντική προσπάθεια εκ μέρους των ανθρώπων που θα την αναλάβουν και ενδεχομένως κρύβει κινδύνους για τη σωματική ακεραιότητα των παιδιών. Συνεπώς, όσο και αν το επιθυμητό είναι να αφιερωθούν πολλές ώρες πρακτικής εξάσκησης και γενικά διδασκαλίας της Κ.Α., κάτι τέτοιο είναι στην πράξη ανεφάρμοστο.

Μήπως η λύση στα παραπάνω προβλήματα μπορεί να προέλθει από το χώρο της Πληροφορικής; Πράγματι, ένα πολύ σημαντικό μέρος του διδακτικού υλικού για την Κ.Α. αφορά εφαρμογές πολυμέσων και υπερκειμένου. Αυτό που διαπιστώνεται όμως είναι ότι σχεδόν το σύνολο των εφαρμογών αφορούν την απλή παροχή γνώσεων. Το υλικό που περιλαμβάνουν, όσο πρωτότυπο και επιστημονικά σχεδιασμένο κι αν είναι, δε δίνει τη δυνατότητα στα παιδιά να εξασκηθούν σε δεξιότητες και συμπεριφορές, αφήνοντας μετέωρη τη λύση του προβλήματος της πρακτικής εξάσκησης.

Ελάχιστες είναι οι εφαρμογές που αξιοποιούν τις δυνατότητες της Πληροφορικής περιλαμβάνοντας κάποια μορφή αναπαράστασης του οδικού περιβάλλοντος, έστω και σαν εκπαιδευτικό παιχνίδι (ενδεικτικά [19, 20, 21]). Ακόμα και σ' αυτή την περίπτωση, στο παιδί δίνονται ελάχιστες δυνατότητες παρέμβασης ή ενεργού δράσης. Σε γενικές γραμμές παρουσιάζεται μία κατάσταση με τη χρήση κινούμενων γραφικών και ζητείται από το παιδί να επιλέξει τι θα πρέπει να κάνει ή αν η κατάσταση που του παρουσιάζεται είναι σωστή ή λάθος (*εικόνα 1.1*). Ακόμα και αν γίνει δεκτό ότι αυτό αποτελεί μία

μορφή εξάσκησης στον εντοπισμό κινδύνων στο δρόμο, σε καμία περίπτωση δεν αποτελεί εξάσκηση στις υπόλοιπες δεξιότητες της ασφαλούς κυκλοφορίας.

Υπάρχουν ωστόσο δύο σημαντικές ερευνητικές εφαρμογές Κ.Α. με χρήση ηλεκτρονικών υπολογιστών που επιχείρησαν να δώσουν τη δυνατότητα στα παιδιά να εξασκηθούν και σε άλλες δεξιότητες. Η πρώτη ερευνητική εφαρμογή διερεύνησε τη δυνατότητα αυτή με τη χρήση βίντεο και προσομοίωσης στον ηλεκτρονικό υπολογιστή (*simulation*) χρησιμοποιώντας μία εφαρμογή πολυμέσων [22]. Για την προσομοίωση κατασκευάστηκαν 24 σκηνές κυκλοφορίας στο δρόμο. Μετά από την παρουσίαση της καθεμίας ζητούνταν από τα παιδιά να περιγράψουν τι είδαν και τι άκουσαν σε σχέση με την κυκλοφοριακή κατάσταση που παρουσιαζόταν.



Εικόνα 1.1 Παράδειγμα εφαρμογής Κ.Α. με σχηματική αναπαράσταση οδικού περιβάλλοντος

Τα αποτελέσματα της προσομοίωσης σχεδόν ταυτίζονταν με τα αποτελέσματα της ίδιας διαδικασίας αλλά σε πραγματικό περιβάλλον κυκλοφορίας, πράγμα που δεν ίσχυσε στην περίπτωση του βίντεο. Με άλλα λόγια, τα παιδιά συμπεριφέρθηκαν στην προσομοίωση στο ηλεκτρονικό υπολογιστή με τον ίδιο τρόπο που συμπεριφέρθηκαν στο πραγματικό περιβάλλον. Αυτό το γεγονός οδήγησε τους ερευνητές στην εξέταση τρόπων με τους οποίους η προσομοίωση θα μπορούσε να έχει καλύτερα αποτελέ-

σματα. Αυτό που προέκυψε ήταν ότι καλύτερα αποτελέσματα επιτυγχάνονται με την καθοδήγηση από ενήλικα. Η έρευνα κατέληξε τέλος στο συμπέρασμα ότι υπάρχουν ισχυρές ενδείξεις πως οι προσομοιώσεις μπορούν να παίξουν σημαντικό ρόλο στην εξάσκηση σημαντικών δεξιοτήτων κυκλοφορίας των παιδιών.

Αυτός ακριβώς ήταν ο σκοπός της δεύτερης ερευνητικής εφαρμογής. Διερεύνησε τη δυνατότητα εξάσκησης σε προσομοίωση των δεξιοτήτων εύρεσης ασφαλούς σημείου για διάσχιση ενός δρόμου, των στρατηγικών οπτικής αναζήτησης, του εντοπισμού του “κενού” στην κυκλοφορία και της βελτίωσης της αντίληψης των παιδιών για τις προθέσεις των άλλων χρηστών του δρόμου, των οδηγών [23].

Η εξάσκηση σε καθεμία από τις παραπάνω δεξιότητες αποτελούσε και ένα ξεχωριστό *άρθρωμα* (*module*) της εφαρμογής, με τέσσερις ενότητες για κάθε *άρθρωμα*. Όπως και η προηγούμενη εφαρμογή, έτσι και αυτή χρησιμοποίησε τα πολυμέσα για την υλοποίησή της. Σε κάθε ενότητα υπήρχε μία σειρά από δραστηριότητες που αφορούσαν τη λήψη αποφάσεων για το αν, πώς και πότε ο χαρακτήρας που αντιπροσώπευε το παιδί στην εφαρμογή έπρεπε να περάσει ένα δρόμο.

Από τη στιγμή που το παιδί έπαιρνε μία απόφαση, κινούσε το χαρακτήρα και, αν η απόφαση ήταν σωστή, ο χαρακτήρας διέσχιζε το δρόμο, αν η απόφαση ήταν λάθος, ο υπολογιστής έδειχνε τις συνέπειες αυτής της απόφασης (*εικόνα 1.2*). Με τον τρόπο αυτόν υπήρχε μία μορφή περιορισμένης αλληλεπίδρασης μεταξύ υπολογιστή και παιδιού. Η συμπεριφορά των παιδιών στην προσομοίωση ελέγχθηκε με αντίστοιχες δραστηριότητες σε πραγματικό οδικό περιβάλλον.



Εικόνα 1.2 Εφαρμογή προσομοίωσης οδικού περιβάλλοντος

Τα συμπεράσματα αυτής της ερευνητικής εφαρμογής είναι ιδιαίτερα σημαντικά. Η εξάσκηση στην προσομοίωση προκάλεσε αξιόλογες αλλαγές σε επίπεδο συμπεριφορών και αντιλήψεων και στις τέσσερις δεξιότητες που εξετάστηκαν, σ' όλες τις ηλικιακές ομάδες. Ένα δεύτερο σημείο είναι ότι καμία από τις υπό εξέταση δεξιότητες δε στάθηκε δύσκολη για τα μικρά παιδιά, αλλά ούτε και εύκολη για τα μεγάλα. Αυτό δείχνει ότι υπάρχουν περιθώρια βελτίωσης και για τα μεγαλύτερα παιδιά, ακόμα και αν αυτά είναι ικανότερα από τα μικρά. Συνεπώς, η προσομοίωση με ηλεκτρονικό υπολογιστή είναι κατάλληλη για κάθε ηλικία.

Η ερευνητική ομάδα δε θεωρεί ότι η προσομοίωση με ηλεκτρονικούς υπολογιστές μπορεί να υποκαταστήσει πλήρως την εξάσκηση σε πραγματικές συνθήκες. Εντούτοις, κάνει δύο πολύ σημαντικές διαπιστώσεις συγκρίνοντας την προσομοίωση με την εξάσκηση σε πραγματικές συνθήκες:

1. Η προσομοίωση είναι πολύ πιο εύκολη στην εφαρμογή απ' ό,τι η εξάσκηση σε πραγματικές συνθήκες.
2. Η προσομοίωση επιτρέπει την εξάσκηση σε δεξιότητες που είναι αδύνατο να εξασκηθούν σε πραγματικό οδικό περιβάλλον.

Όπως και στην προηγούμενη έρευνα, συστήνεται ως ενδεδειγμένη διδακτική μεθοδολογία η καθοδήγηση από ενήλικα, τουλάχιστον για το κομμάτι της εξάσκησης στην προσομοίωση. Μάλιστα, το λογισμικό που χρησιμοποιήθηκε ήταν έτσι σχεδιασμένο, ώστε να επιτρέπει αυτές τις αλληλεπιδράσεις.

Συνοψίζοντας, προγράμματα προσομοίωσης με τη χρήση ηλεκτρονικών υπολογιστών οδήγησαν στη βελτίωση δεξιοτήτων και συμπεριφορών, κάτω από τις εξής συνθήκες:

- Όταν απαιτήσαν από τα παιδιά να κάνουν κρίσεις αντίστοιχες με εκείνες που γίνονται κάτω από πραγματικές συνθήκες.
- Όταν οι προσομοιώσεις χρησιμοποιήθηκαν επιλεκτικά, λαμβάνοντας υπόψη τα πλεονεκτήματα και τα μειονεκτήματά τους ως εκπαιδευτικού εργαλείου.
- Όταν εξετάστηκαν οι ανάγκες εκείνων που τις χρησιμοποίησαν, δηλαδή των παιδιών. Η κατάρτιση και η παρεχόμενη υποστήριξη πρέπει να είναι ανάλογη με τις δυνατότητές τους.

1.7 Συμπεράσματα για το 1^ο κεφάλαιο

Στην αρχή του κεφαλαίου θέσαμε ως σκοπό της διδασκαλίας την απόκτηση από τους μαθητές όχι μόνο γνώσεων, αλλά και δεξιοτήτων και συμπεριφορών, όπως επίσης και τη –με τον καλύτερο δυνατό τρόπο– κατανόηση ενός γνωστικού αντικειμένου. Αποδείξαμε ότι οι κυριότερες θεωρίες μάθησης, παρά τις όποιες διαφορές τους, υποστηρί-

ζουν ότι ο ενεργός ρόλος του μαθητή είναι το στοιχείο εκείνο στο οποίο πρέπει να εστιάσουμε την προσοχή μας.

Τι είναι όμως ο ενεργός ρόλος, η ενεργός συμμετοχή του μαθητή στην εκπαιδευτική διαδικασία; Τι σημαίνει πρακτική εξάσκηση και εμπράγματα εμπειρίες; Μπορεί να φαντάζουν σαν αφελείς και εύκολο να απαντηθούν ερωτήσεις, αποτελούν όμως θέματα που συχνά παρανοούνται. Όλοι θα συμφωνήσουν ότι ενεργός συμμετοχή και πρακτική εξάσκηση δεν είναι η συμβατική διδασκαλία ενός αντικειμένου όπου στο τέλος της περιλαμβάνει μία πληθώρα ερωτήσεων δίκην αξιολόγησης.

Ας υποθέσουμε ότι οι ερωτήσεις απαιτούν κριτική και συνδυαστική σκέψη και ότι η εργασία πάνω σε αυτές γίνεται σε ομάδες, με συνεργασία των παιδιών και με την καθοδήγηση του δασκάλου. Ή ακόμα, ας υποθέσουμε ότι δεν υπάρχουν καθόλου ερωτήσεις αξιολόγησης, αλλά αναθέτουμε στους μαθητές μας εργασίες που προϋποθέτουν αναζήτηση σε πηγές, αποδελτίωση, συγγραφή και παρουσίαση της εργασίας με κείμενα, θέατρο, παντομίμα, κατασκευές, κτλ. Θα έχουμε τότε πρακτική εξάσκηση και ενεργό συμμετοχή; Ίσως κάποιοι απαντήσουν καταφατικά, κάποιοι άλλοι ίσως να δεχθούν ότι μόνο σ' αυτό το σκέλος της διδασκαλίας πράγματι οι μαθητές ανέλαβαν ενεργό δράση.

Για μας η απάντηση είναι αρνητική για το σύνολο της διδασκαλίας. Σε κάθε στάδιο, σε κάθε βήμα της διδασκαλίας πρέπει να υπάρχει ενεργός συμμετοχή των μαθητών, και κυρίως στο πρώτο. Αυτό γιατί επισημάναμε ότι κάθε σοβαρή προσπάθεια να αναπτυχθούν στα παιδιά γνώσεις, δεξιότητες συμπεριφορές θα πρέπει να ξεκινά από συγκεκριμένες δράσεις στο φυσικό περιβάλλον όπου αυτές εφαρμόζονται, γιατί έτσι αναπτύσσεται η σχέση μεταξύ της γνώσης/δεξιότητας/συμπεριφοράς και του πεδίου εφαρμογής της.

Βηματικά, με συνεχείς μετασχηματισμούς προτύπων δράσης σε βασικές εμπειρίες, επιτυγχάνουμε τη δημιουργία νέων αντιλήψεων και εννοιών. Καταλήξαμε μάλιστα στο ότι το ενδεδειγμένο διδακτικό σχήμα είναι μία μορφή “καθοδηγούμενης συνεργασίας”, όπου οι μαθητές εργάζονται ομαδικά και ο δάσκαλος αναλαμβάνει ένα συντονιστικό, καθοδηγητικό ρόλο.

Θα συμφωνήσουμε με όσους κατακρίνουν το παραπάνω διδακτικό μοντέλο προτάσσοντας το επιχείρημα ότι είναι ανεδαφική η εφαρμογή του. Πράγματι, λαμβάνοντας υπόψη την πληθώρα των γνωστικών αντικειμένων, τα μέσα και τα υλικά που χρειάζονται, το χρόνο που απαιτείται για την προετοιμασία, την οργάνωση και την υλοποίηση τέτοιας μορφής διδασκαλίας, διαπιστώνεται ότι είναι αδύνατο να γίνει. Ακόμα, υπάρ-

χουν περιπτώσεις όπου εκ των πραγμάτων είναι αδύνατη η απόκτηση εμπράγματων εμπειριών (για παράδειγμα επειδή οι μαθητές μπορεί να διατρέξουν κάποιον κίνδυνο). Όλα αυτά όμως δεν απορρίπτουν την ουσία της παραδοχής ότι η δράση, η ενεργός συμμετοχή, είναι ο αποτελεσματικότερος τρόπος επίτευξης των διδακτικών στόχων. Μας ωθούν όμως να αναζητήσουμε εναλλακτικές λύσεις.

Επιλέξαμε να παρουσιάσουμε μία ιδιαίτερα δύσκολη περίπτωση (αυτή της Κυκλοφοριακής Αγωγής), γιατί συγκεντρώνει όλα τα παραπάνω προβλήματα. Είδαμε ότι η συνεισφορά της Πληροφορικής έγκειται κυρίως στην παροχή γνωστικού υλικού (εφαρμογές πολυμέσων και Διαδικτύου), που δε διαφέρει σημαντικά από το συμβατικό (βιβλία, βίντεο, κτλ.). Ξεχωρίσαμε όμως δύο πειραματικές εφαρμογές που αντί της συμβατικής προσέγγισης απεικόνισαν ένα οδικό περιβάλλον και ζητούσαν από τα παιδιά να κάνουν κρίσεις, να λάβουν αποφάσεις και να ενεργήσουν, να λειτουργήσουν δηλαδή με τρόπο αντίστοιχο προς τις πραγματικές συνθήκες. Αυτές οι εφαρμογές, που ανήκουν στην κατηγορία των προσομοιώσεων, είχαν αποτελέσματα αντίστοιχα με την εξάσκηση σε κανονικούς δρόμους, χωρίς όμως τους κινδύνους, την προετοιμασία, τον κόπο και το χρόνο που συνεπάγεται κάτι τέτοιο.

Οι προσομοιώσεις δεν είναι κάτι νέο στις εφαρμογές της Πληροφορικής. Εδώ και χρόνια χρησιμοποιούνται σε τομείς όπου απαιτείται υψηλού επιπέδου κατάρτιση (για παράδειγμα στην εκπαίδευση πιλότων, γιατρών, κτλ.). Όμως αυτές οι προσομοιώσεις έχουν ένα ιδιαίτερο χαρακτηριστικό. Τα γραφικά που χρησιμοποιούν δεν είναι δύο διαστάσεων (*2D Graphics*), όπως οι δύο ερευνητικές εφαρμογές που παρουσιάστηκαν, αλλά τριών διαστάσεων (*3D Graphics*). Η χρήση της τρίτης διάστασης μας εισάγει αυτόματα σε μία άλλη τεχνολογία υλικού και λογισμικού, γνωστή με το όνομα Εικονική Πραγματικότητα. Αν λοιπόν έχουμε θετικά γνωστικά αποτελέσματα με τη χρήση γραφικών δύο διαστάσεων, τι θα συμβεί αν για τον ίδιο σκοπό χρησιμοποιήσουμε εφαρμογές Ε.Π.;

Κεφάλαιο 2°

Γενιές εκπαιδευτικής τεχνολογίας και η διαδικασία μάθησης
Μία αιρετική(;) προσέγγιση

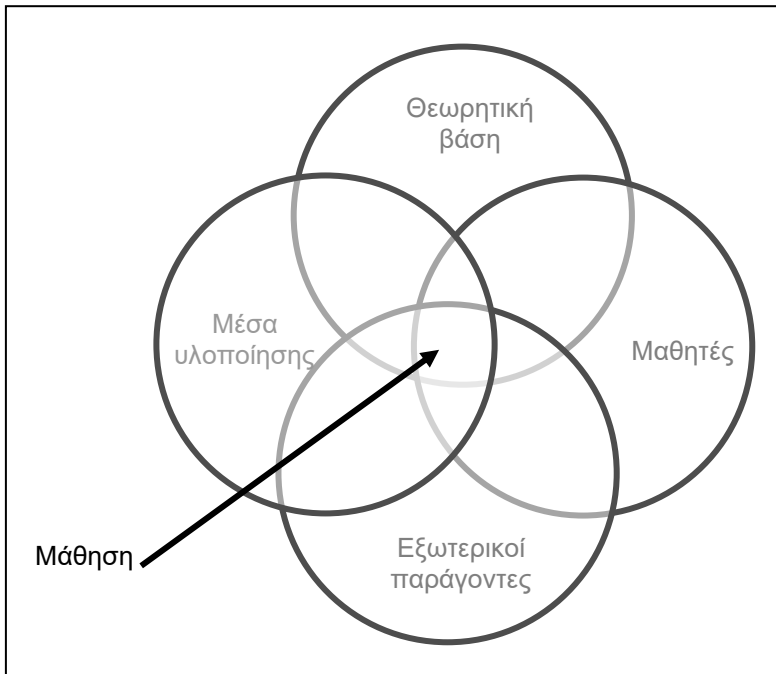
Η ανάλυση των κυριότερων θεωριών μάθησης οδηγεί σ' έναν ευρύτερο προβληματισμό που αφορά τη μαθησιακή διαδικασία ως σύνολο. Ανεξάρτητα του τρόπου με τον οποίο αντιμετωπίζεται η εκπαιδευτική διαδικασία, ανεξάρτητα των διδακτικών μεθόδων και των μέσων που χρησιμοποιούνται, μπορούν να αναγνωριστούν τέσσερις κύριες ομάδες παραγόντων που επηρεάζουν το τελικό αποτέλεσμα, δηλαδή τη μάθηση. Αυτές είναι η θεωρητική βάση, τα μέσα υλοποίησης, οι μαθητές και τέλος εξωτερικοί παράγοντες που επιδρούν σε μέρος ή στο σύνολο των υπολοίπων παραγόντων (σχήμα 2.1).

Θεωρητική βάση. Το θεωρητικό υπόβαθρο βάζει τους στόχους της εκπαίδευσης και καθορίζει τον τρόπο, τη διαδικασία που θα ακολουθηθεί ώστε να επιτευχθούν αυτοί οι στόχοι. Οι στόχοι και η μεθοδολογία είναι τα κοινά στοιχεία που συναντώνται σ' όλο το φάσμα των θεωριών μάθησης, από το συμπεριφορισμό ως το δομητισμό, παρά τις σημαντικές διαφορές τους.

Μέσα υλοποίησης. Αποτελούν το σύνολο των διδακτικών εργαλείων που θα χρησιμοποιηθούν για να υλοποιηθεί η διδασκαλία με βάση τον τρόπο που καθορίζει η θεωρητική βάση. Τα μέσα παρεμβάλλονται μεταξύ θεωρίας και μαθητών και μπορούν να διαχωριστούν σε έμψυχο και άψυχο υλικό. Οι εκπαιδευτικοί αποτελούν το έμψυχο υλικό. Η προσωπικότητά τους, η επάρκεια της εκπαίδευσής τους, ο βαθμός που ασπάζονται τη θεωρητική βάση της εκπαιδευτικής διαδικασίας, οι σχέσεις τους με τους μαθητές, η ικανότητα να χρησιμοποιούν άλλα διδακτικά εργαλεία είναι μερικοί από τους παράγοντες που παίζουν ρόλο στο τελικό αποτέλεσμα. Στο άψυχο υλικό κατατάσσονται τα βιβλία, τα διάφορα έντυπα, τα εποπτικά μέσα, οι ηλεκτρονικοί υπολογιστές και γενικά κάθε άλλο μέσο που παρέχει γνώσεις και συνεισφέρει στην πρακτική εξάσκηση των παιδιών.

Μαθητές. Είναι η ομάδα-στόχος του κάθε εκπαιδευτικού συστήματος. Η ιδιοσυγκρασία και η προσωπικότητά τους, η οικονομική και κοινωνική τους προέλευση, η στάση τους απέναντι στην εκπαίδευση και στα μέσα που χρησιμοποιούνται, παίζουν καθοριστικό ρόλο.

Εξωτερικοί παράγοντες. Είναι το σύνολο των παραγόντων που, ενώ δε σχετίζονται άμεσα με τους παραπάνω, επιδρούν στην αποτελεσματικότητα της μαθησιακής διαδικασίας. Τέτοιοι εξωγενείς παράγοντες μπορεί να αναφερθεί ότι είναι ο κοινωνικός περίγυρος (άλλοι μαθητές, φίλοι, γονείς, συγγενείς, κτλ.), το περιβάλλον (σχολικό, γεωγραφικό, κλιματολογικό), ο χρόνος κατά τον οποίο λαμβάνει χώρα η εκπαιδευτική διαδικασία, πολιτικές, πολιτισμικές οικονομικές και κοινωνικές μεταβλητές.



Σχήμα 2.1 Παράγοντες της εκπαιδευτικής διαδικασίας

Από την παραπάνω κατηγοριοποίηση μπορούν να εξαχθούν μερικές χρήσιμες παρατηρήσεις. Η μάθηση είναι τελικά μία διαδικασία εξαιρετικά ευαίσθητη, που εξαρτάται από αναρίθμητους παράγοντες. Εξαιτίας μάλιστα της αλληλεξάρτησης των παραγόντων που συμμετέχουν σ' αυτή, δεν είναι ασφαλές να ειπωθεί ότι ο ένας παράγοντας είναι πιο σημαντικός από τον άλλο. Η παραμικρή μεταβολή ή παράλειψη σε μία μεταβλητή ενός και μόνο παράγοντα μπορεί να οδηγήσει σε αντίθετα από τα αναμενόμενα αποτελέσματα. Για παράδειγμα, μία θεωρία μάθησης που προτείνει μία συγκεκριμένη διδακτική μεθοδολογία η οποία όμως δεν ταιριάζει στην ιδιοσυγκρασία ορισμένων μαθητών μπορεί να έχει ως αποτέλεσμα να μην είναι αποτελεσματική σε αυτούς.

Συνήθως, για ερευνητικούς λόγους, απομονώνονται και εξετάζονται σημαντικές αλλά λίγες μεταβλητές από έναν ή περισσότερους από τους παραπάνω παράγοντες μάθησης. Κάτι τέτοιο, παρότι είναι λογικό, εντούτοις αφήνει πάντα αμφιβολίες για την ορθότητα των εξαγόμενων συμπερασμάτων. Απόδειξη είναι η πληθώρα των θεωριών μάθησης [24]. Στο σύνολό τους ισχυρίζονται ότι περιγράφουν επαρκώς το φαινόμενο και προτείνουν αποτελεσματικά μοντέλα για τη διαδικασία της μάθησης, παρά το γεγονός των μικρών ή μεγάλων τους διαφορών.

Μία άλλη ένδειξη για την αμφίβολη καθολική ισχύ των θεωριών είναι η καθημερινή πρακτική των δασκάλων. Οι δάσκαλοι ακολουθούν μία ευρεία γκάμα διδακτικών μεθόδων από διαφορετικές θεωρητικές βάσεις, τις οποίες αναπροσαρμόζουν δυναμικά ανάλογα με τις συνθήκες.

Υπό αυτές τις συνθήκες μπορεί κάποιος να υιοθετήσει την ακραία θέση ότι ή όλες οι θεωρίες μάθησης είναι σωστές ή ότι όλες είναι λάθος. Το πιο πιθανό είναι ότι δεν υπάρχει μία και μόνο θεωρία που να ορίζει και να περιγράφει εξ ολοκλήρου το φαινόμενο της μάθησης. Δεν υπάρχει μία λεωφόρος, αλλά πολλά τεμνόμενα ή παράλληλα μονοπάτια που, αν ακολουθηθούν, επιτυγχάνεται η μάθηση. Πράγματι, έχουν καταβληθεί προσπάθειες να συγκεραστούν οι διάφορες θεωρίες ή έχουν προταθεί καταστάσεις στις οποίες μία θεωρία μπορεί να είναι καταλληλότερη από μία άλλη.

Για παράδειγμα, προτείνεται η απόφαση για την εκάστοτε προσέγγιση να είναι αποτέλεσμα των περιστάσεων που ισχύουν σε μία δεδομένη χρονική στιγμή και ότι σε μερικές περιπτώσεις χρειάζεται να εφαρμοστούν λύσεις καθοδηγητικής διδασκαλίας και σε άλλες περιπτώσεις ο μαθητής να μπορεί να έχει περισσότερο έλεγχο στο τι και πώς διδάσκεται [25].

Ο Jonassen στο “*A Manifesto for a Constructive Approach to Technology in Higher Education*” [26], προσδιορίζει τρία στάδια απόκτησης γνώσης και τα αντιστοιχεί με τις θεωρίες μάθησης που πιστεύει ότι ταιριάζουν καλύτερα. Στην *εισαγωγική μάθηση (Introductory Learning)*, όπου οι μαθητές έχουν ακόμα λίγες παραστάσεις, οι κλασικές διδακτικές μέθοδοι είναι καλύτερες, γιατί έχουν προκαθορισμένους στόχους και φυσική συνέχεια, και έτσι τα παιδιά αναπτύσσουν μία βάση για περαιτέρω εξερεύνηση της γνώσης. Στο στάδιο της *προχωρημένης απόκτησης γνώσης (Advanced Knowledge Acquisition)*, μπορούν να εισαχθούν στοιχεία δομητισμού. Στο τελευταίο στάδιο της *εμπειρίας (Expertise)*, όπου ο μαθητής μπορεί να παίρνει ευφυείς αποφάσεις μέσα στο μαθησιακό περιβάλλον, μία δομητιστική διδακτική προσέγγιση ταιριάζει πολύ καλά. Ο Jonassen επισημαίνει όμως ότι είναι σημαντικό να λαμβάνεται υπόψη το περιεχόμενο πριν ληφθεί οποιαδήποτε απόφαση για τη διδακτική μεθοδολογία.

Έπειτα από τη σύγκριση συμπεριφορισμού, γνωστικών θεωριών μάθησης και δομητισμού, οι Ertmer και Newby [27] οδηγούνται σε μία παρόμοια προσέγγιση με αυτή του Jonassen. Θεωρούν ότι μία συμπεριφοριστική διδακτική προσέγγιση ταιριάζει σε εργασίες που χρειάζονται λίγη σκέψη και λογική επεξεργασία, όπως απομνημόνευση, βασικές ταξινομήσεις, διάκριση απλών διαφορών, κτλ. Από την άλλη, μία γνωστική προσέγγιση ταιριάζει καλύτερα σε εκείνες τις διδακτικές καταστάσεις όπου απαιτείται υψηλότερος βαθμός νοητικής επεξεργασίας, όπως ταξινομήσεις, αλγοριθμική επίλυση

προβλημάτων, λογική, κτλ. Τέλος, οι παραπάνω ερευνητές πιστεύουν ότι ο δομητισμός είναι κατάλληλος για εργασίες που απαιτείται πολύ υψηλός βαθμός επεξεργασίας, όπως προσωπικές επιλογές, έλεγχος των γνωστικών στρατηγικών, επίλυση προβλημάτων χωρίς προηγούμενη κρίση (*heuristic problem solving*).

Είναι όμως πραγματικά σημαντικό πρόβλημα ότι δεν υπάρχει μία και μόνο θεωρία μάθησης που να εφαρμόζεται σε κάθε περίπτωση; Μάλλον όχι, γιατί αυτό επιβάλλεται από την ποικιλία και την πολυπλοκότητα της ανθρώπινης φύσης. Στο σημείο αυτό αξίζει να εξεταστεί ένα ακόμα γεγονός. Η αλλαγή στους στόχους της εκπαίδευσης (θεωρητική βάση), που είναι ένα συχνό φαινόμενο, επιβάλλει τη μερική ή ολική αλλαγή στα μέσα υλοποίησης, για παράδειγμα τα βιβλία. Η αλλαγή αυτή είναι μία διαδικασία επίπονη και χρονοβόρα. Μεσολαβεί δε ένα “νεκρό χρονικό διάστημα” κατά το οποίο η δομή και το περιεχόμενο των διδακτικών μέσων δε συμβαδίζει με την παιδαγωγική θεωρία που επιχειρείται να υλοποιηθεί. Στα παραπάνω πρέπει να συνυπολογιστεί το ότι οι εκπαιδευτικοί είναι προϊόντα μίας τρίτης παιδαγωγικής αντίληψης (όταν δηλαδή εκείνοι ήταν στη θέση του διδασκόμενου) και ότι καλούνται να εκτελέσουν το καθήκον τους, τις περισσότερες φορές, κάτω από αντίξοες συνθήκες (εξωτερικοί παράγοντες).

Μάλιστα όλα αυτά συμβαίνουν στα πλαίσια μίας συνεχώς μεταβαλλόμενης κοινωνίας που δυναμικά επαναπροσδιορίζει τις ανάγκες της, αλλάζοντας και πάλι τους εκπαιδευτικούς στόχους, με αποτέλεσμα η όλη διαδικασία να επαναλαμβάνεται στο διηνεκές. Παρ’ όλα αυτά, κανείς δεν μπορεί να αμφισβητήσει το γεγονός ότι οι μαθητές τελικά αποκτούν σε κάποιο βαθμό τις γνώσεις και ικανότητες που οι ίδιοι ή οι άλλοι θεωρούν απαραίτητες.

Σε μία απόπειρα να ερμηνευτεί αυτό το φαινόμενο κάτω από το πρίσμα όλων των παραπάνω παρατηρήσεων, μπορούν να διατυπωθούν τρεις αρχές μάθησης:

Η μάθηση είναι ατομικό φαινόμενο. Ο κάθε άνθρωπος μαθαίνει με το δικό του τρόπο. Ο καθένας αντιλαμβάνεται και ερμηνεύει τον κόσμο που τον περιβάλλει διαφορετικά. Συνεπώς, τα ερεθίσματα από τον περιβάλλοντα κόσμο μετουσιώνονται σε γνώσεις και δεξιότητες που διαφέρουν από άτομο σε άτομο, δεν υπάρχει μόνο μία και σωστή αναπαράσταση της γνώσης. Αυτή η άποψη βρίσκεται διάχυτη στο έργο του Piaget (ενδεικτικά [1, 2, 28, 29]). Όμως αυτό το γεγονός οδηγεί στη υιοθέτηση της άποψης ότι οι υπάρχουσες θεωρίες μάθησης, αλλά και οποιεσδήποτε ακολουθήσουν, πρέπει να αντιμετωπίζονται ως απόπειρες ομαδοποίησης της μαθησιακής διαδικασίας με βάση κάποια από τα κοινά χαρακτηριστικά αυτών που μαθαίνουν και πάντα με την αίρεση ότι δεν ισχύουν τα ίδια για όλους τους ανθρώπους.

Η μάθηση είναι κοινωνικό φαινόμενο. Αυτή η αρχή πηγάζει από το έργο του Vygotsky [3, 4]. Το άτομο μαθαίνει γιατί είναι ουσιώδες για την ύπαρξή του να μάθει. Όμως το άτομο δέχεται και εξωτερικές επιδράσεις. Οι επιδράσεις ασκούνται στο πλαίσιο της κοινωνίας γιατί ο άνθρωπος είναι κοινωνικό ον. Αυτό συνεπάγεται ότι το άτομο με κανέναν τρόπο δεν μπορεί να τις αποφύγει. Άρα, ό,τι τελικά μαθαίνει, είναι και αποτέλεσμα του συνόλου των εξωτερικών επιδράσεων που δέχεται, είτε θετικών είτε αρνητικών. Παρότι υπάρχουν αντιλήψεις που υποστηρίζουν ότι ο άνθρωπος είναι ένα “κλειστό πληροφοριακό σύστημα” και ότι δε λαμβάνει άμεσα γνώσεις από το εξωτερικό περιβάλλον [30], παράλληλα δέχονται ότι διαταραχές σ' αυτό το σύστημα δημιουργούν εσωτερικές δομικές προσαρμογές, που και αυτό είναι μία μορφή επίδρασης.

Η αρχή της απροσδιοριστίας. Οι δύο προηγούμενες αρχές, καθώς επίσης και οι παράγοντες μάθησης που αναφέρθηκαν πιο πάνω, οδηγούν στον εξής προβληματισμό. Το δεδομένο είναι ότι η κάθε θεωρία μάθησης επιδιώκει να περιγράψει επιτυχώς και με πληρότητα τη μαθησιακή διαδικασία. Εφόσον όμως η μάθηση είναι και ατομικό και κοινωνικό φαινόμενο και αφού κατά τη διαδικασία της μάθησης επιδρούν στο άτομο αναρίθμητοι παράγοντες τόσο ενδογενείς όσο και εξωγενείς, τότε η κάθε θεωρία μάθησης πρέπει να λαμβάνει υπόψη της ταυτόχρονα όλους αυτούς τους παράγοντες.

Επειδή μάλιστα η μαθησιακή διαδικασία δεν είναι εντοπισμένη χρονικά σε μία μόνο στιγμή της ζωής του ατόμου, αλλά επεκτείνεται στο σύνολό της, η κάθε θεωρία θα πρέπει να συνυπολογίζει το πώς οι παράγοντες μάθησης μεταβάλλονται χρονικά. Αν μάλιστα επιδιώκεται η μαθησιακή διαδικασία να έχει τα καλύτερα δυνατά αποτελέσματα σ' όλους, θα πρέπει να συνυπολογίζονται όλοι οι παράγοντες που επιδρούν σε κάθε άτομο ξεχωριστά και η μαθησιακή διαδικασία θα πρέπει να περιγράφεται επακριβώς, όσο διαφορετικά και αν είναι τα άτομα μεταξύ τους. Υπάρχει όμως η δυνατότητα να προσδιοριστούν με τη μέγιστη ακρίβεια όλες οι παράμετροι που επιδρούν έστω και σ' ένα μόνο άτομο; Η απάντηση είναι αρνητική.

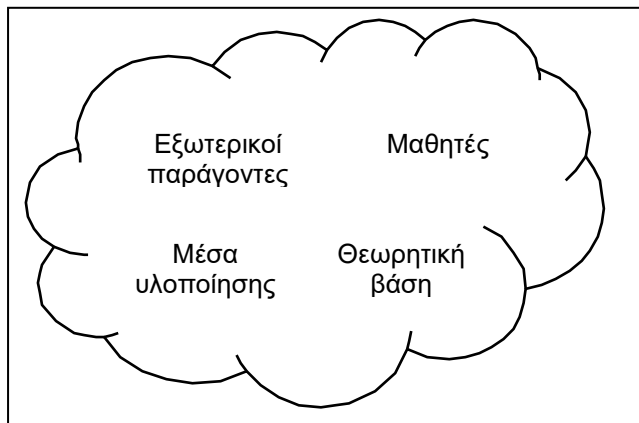
Συνεπώς κρίνεται σκόπιμο να διατυπωθεί μία τρίτη αρχή, η αρχή της απροσδιοριστίας στη μάθηση. Είναι ουσιαστικά η παραδοχή ότι δεν μπορεί να προσδιοριστεί με ακρίβεια η μαθησιακή διαδικασία, ούτε να τοποθετηθεί σε χρονικά πλαίσια ούτε να προβλεφθούν τα αποτελέσματά της, για το λόγο ότι οι μεταβλητές, εκτός του ότι είναι πολυάριθμες, έχουν την τάση να αλλάζουν συνεχώς, ακόμα και για ένα συγκεκριμένο άτομο. Η μάθηση δεν είναι ένα απλό φυσικό φαινόμενο που μπορεί να ερμηνευτεί με συγκεκριμένους νόμους. Τα μόνα δεδομένα είναι ότι υπάρχει και ότι οι διαστάσεις της είναι βιολογικές, ψυχολογικές και κοινωνικές.

Η παραπάνω τοποθέτηση δεν πρέπει να ερμηνευτεί ως ένδειξη απαξίωσης όλων των θεωριών μάθησης. Αντίθετα, εξηγεί το λόγο της αποτυχίας των θεωριών εκείνων που αντιμετωπίζουν τη μάθηση ως ένα κλειστό σύστημα και αναδεικνύει την ανάγκη για την υιοθέτηση μίας ανοικτής προσέγγισης στο φαινόμενο αυτό. Βασικές παραδοχές της ανοικτής προσέγγισης στη μάθηση εξαιτίας της αρχής της απροσδιοριστίας είναι:

- Η μάθηση συντελείται δυναμικά, ακόμα και με τυχαίο τρόπο, σε κάθε χρονική στιγμή και σε κάθε τόπο.
- Οι στόχοι που τίθενται μπορούν και επιβάλλεται να μεταβάλλονται μερικώς ή ολικώς και να προσαρμόζονται ανάλογα με την εξέλιξη της μαθησιακής διαδικασίας.
- Τα αποτελέσματα της μαθησιακής διαδικασίας δεν είναι προκαθορισμένα.

Με αυτή τη λογική, κάθε θεωρητικό σύστημα οφείλει να εκμεταλλεύεται όλες τις καταστάσεις και να τις μετουσιώνει σε ευκαιρίες για μάθηση. Πρέπει να είναι τόσο ευέλικτο, ώστε να προσαρμόζεται στις ανάγκες ενός ατόμου ή μίας ομάδας ατόμων. Τέλος, δεν πρέπει να πάσχει από αγκυλώσεις στους στόχους που θέτει και στα αποτελέσματα που προσδοκά. Έτσι δημιουργείται η ανάγκη επανασχεδιασμού του σχήματος που περιγράφει τη μάθηση και τους παράγοντές της.

Οι τεμνόμενοι κύκλοι του σχήματος 2.1 κατά κάποιον τρόπο υπαινίσσονται ότι η τομή των παραγόντων και τα όριά τους είναι σαφώς προσδιορισμένα. Αντίθετα, με βάση την αρχή της απροσδιοριστίας, δεν υπάρχουν σαφή όρια ούτε στους παράγοντες ούτε στην τομή τους ούτε στην ίδια την μάθηση (σχήμα 2.2).



Σχήμα 2.2 Οι παράγοντες μάθησης με βάση την αρχή της απροσδιοριστίας

2.1 Γενιές εκπαιδευτικής χρήσης των ηλεκτρονικών υπολογιστών σε σχέση με τις αρχές μάθησης

Κάτω από το πρίσμα των τριών αρχών μάθησης μπορούν να εξεταστούν οι εκπαιδευτικές χρήσεις των ηλεκτρονικών υπολογιστών. Από τα πρώτα στάδια της εξέλιξής τους θεωρήθηκαν ένα ιδανικό διδακτικό εργαλείο. Τα χαρακτηριστικά που επιτρέπουν αυτή την πεποίθηση συμβαδίζουν με τις τεχνολογικές εξελίξεις και κυρίως με την αυξανόμενη υπολογιστική τους ισχύ, την ταχύτητα, τα γραφικά περιβάλλοντα και τη δυνατότητα ενσωμάτωσης εικόνας, ήχου και βίντεο.

2.1.1 Οι τρεις πρώτες γενιές εκπαιδευτικής χρήσης των υπολογιστών

Οι πρώτες εκπαιδευτικές εφαρμογές των ηλεκτρονικών υπολογιστών υλοποιούσαν συμπεριφοριστικές αντιλήψεις και έδιναν μεγάλη βαρύτητα στο περιεχόμενο. Ανέλυαν τις γνώσεις και τις δεξιότητες που πρέπει να αποκτηθούν από τους μαθητές σε επιμέρους συστατικά στοιχεία, όπου η γνώση αυτών, αθροιστικά, οδηγούσε στην προσδοκώμενη συμπεριφορά.

Χαρακτηριστικό παράδειγμα αυτής της άποψης ήταν η *υποβοηθούμενη από υπολογιστές διδασκαλία* (*Computer Assisted Instruction, CAI*), που είχε περισσότερο τη μορφή δοκιμής και εξάσκησης, με τον μαθητευόμενο να έχει ελάχιστο έλεγχο στο τι μαθαίνει. Όλα ήταν προκαθορισμένα από το σχεδιαστή, αν και σε κάποιες περιπτώσεις δινόταν η δυνατότητα να παραλειφθούν από τον μαθητευόμενο ορισμένα θέματα [31]. Παρά τη σοβαρή κριτική που δέχτηκε ο συμπεριφορισμός, αρκετές εφαρμογές των υπολογιστών ακολουθούν ακόμα και σήμερα αυτή την παραδοσιακή αντιμετώπιση εκπαιδευτικού σχεδιασμού και διδασκαλίας.

Στη δεύτερη γενιά υπήρξε μία μεταστροφή από την έμφαση στο περιεχόμενο, στην έμφαση στον τρόπο που παρουσιάζεται το περιεχόμενο στους μαθητές. Αυτό προέκυψε από τη συνειδητοποίηση ότι η διαδικασία με την οποία οι μαθητές επεξεργάζονται τις πληροφορίες μπορεί να είναι πιο σημαντική από την ίδια την πληροφορία. Στο πεδίο αυτό ο συμπεριφορισμός αποδείχθηκε ανεπαρκής, ενώ οι γνωστικές θεωρίες προσέφεραν κατάλληλο έδαφος στο οποίο μπόρεσαν να στηριχτούν οι σχεδιαστές εκπαιδευτικών εφαρμογών.

Επειδή όμως ο συμπεριφορισμός και οι γνωστικές θεωρίες αντιμετωπίζουν τη μάθηση με *αντικειμενιστικό* (*Objectivistic*) τρόπο [32], η μετάβαση από τη μία γενιά στην άλλη δεν ήταν δύσκολη. Ο στόχος παρέμεινε η μεταφορά γνώσης στον μαθητευόμενο με τον καλύτερο δυνατό τρόπο [33]. Και στις δύο περιπτώσεις ο σχεδιαστής αναλύει την ύλη σε μικρά κομμάτια. Στην μεν συμπεριφοριστική θεώρηση αναζητείται η καλύτερη

μέθοδος ώστε να επιτευχθεί η επιδιωκόμενη συμπεριφορά, στη δε γνωστική θεώρηση αναζητείται τρόπος και σύστημα μετάβασης από το απλό στο σύνθετο.

Χαρακτηριστικό παράδειγμα αυτής της γενιάς εκπαιδευτικών εφαρμογών σε υπολογιστές είναι προσπάθεια προγραμματισμού τους με τέτοιο τρόπο ώστε να “σκέφτονται” και να απαντούν “έξυπνα” στην είσοδο δεδομένων από το μαθητή, χρησιμοποιώντας μία βάση δεδομένων.

Η τρίτη γενιά εκπαιδευτικής χρήσης των ηλεκτρονικών υπολογιστών στηρίζεται στις αρχές του δομητισμού. Η μεταστροφή στο σχεδιασμό εφαρμογών είναι σημαντική, εξαιτίας του ότι ο δομητισμός δεν αντιμετωπίζει τη μάθηση *αντικειμενιστικά* [32]. Παρά το γεγονός ότι έχουν καταβληθεί προσπάθειες ώστε οι εφαρμογές των ηλεκτρονικών υπολογιστών να συνδυάσουν στοιχεία και από τις γνωστικές θεωρίες επεξεργασίας πληροφοριών και από το δομητισμό (ενδεικτικά [34, 35, 36]), οι διαφορές είναι σημαντικές.

Η *αντικειμενιστική* προσέγγιση της μάθησης επιδιώκει η διδασκαλία να έχει προκαθορισμένα αποτελέσματα και να παρεμβάλλεται στη μαθησιακή διαδικασία με τέτοιο τρόπο, ώστε να αποτυπώσει την πραγματικότητα στο μυαλό του μαθητευομένου. Αντίθετα, η δομητιστική προσέγγιση, επειδή ακριβώς υποστηρίζει ότι δεν υπάρχουν προκαθορισμένα και προβλέψιμα αποτελέσματα, θεωρεί ότι η διδασκαλία θα πρέπει να ενθαρρύνει και όχι να καθοδηγεί τη μάθηση. Όπως επισημαίνει ο Jonassen [37]: “... το πρόβλημα που θέτει ο δομητισμός στους σχεδιαστές εκπαιδευτικών προγραμμάτων είναι ότι, εφόσον κάθε άτομο είναι υπεύθυνο για τη δόμηση της γνώσης, πώς μπορούμε εμείς να προσδιορίσουμε και να εξασφαλίσουμε ένα σύνολο αποτελεσμάτων της μάθησης κοινό για όλους, όπως έχουμε μάθει ότι πρέπει να κάνουμε;”.

Στο ίδιο άρθρο παραθέτει τα χαρακτηριστικά που θα πρέπει να έχει ο γενικός διδακτικός σχεδιασμός σύμφωνα με τη θεώρηση του δομητισμού, όπως:

- Να παρέχει πολλαπλές αναπαραστάσεις της πραγματικότητας και να αποφεύγει τις απλουστεύσεις, παρουσιάζοντας τη φυσική πολυπλοκότητα του κόσμου.
- Να παρέχει στους μαθητές αυθεντικές εργασίες.
- Να παρέχει διδακτικά περιβάλλοντα που να στηρίζονται στη μελέτη πραγματικών περιπτώσεων, παρά να είναι μία προκαθορισμένη διδακτική αλληλουχία.
- Η γνώση να δομείται σε σχέση με το πλαίσιο και το περιεχόμενο που εφαρμόζεται.
- Να ενθαρρύνει τις πρακτικές που στηρίζονται στο συλλογισμό.
- Να υποστηρίζει τη συνεργατική δόμηση της γνώσης και όχι τον ανταγωνισμό μεταξύ των μαθητών.

Τα πολυμέσα και τα υπερμέσα είναι οι εφαρμογές των ηλεκτρονικών υπολογιστών που ικανοποιούν σε μεγάλο βαθμό τις παραπάνω προϋποθέσεις και επιτρέπουν μία μη γραμμική πορεία στη διδασκαλία, αφήνοντας την επιλογή της πορείας στο μαθητευόμενο. Έτσι χρησιμοποιούνται κατεξοχήν για την υλοποίηση δομητιστικών αντιλήψεων. Ωστόσο, από το φόβο μήπως τελικά ο μαθητευόμενος αποπροσανατολιστεί, μήπως “χαθεί” στον κυκλώνα υπερ-συνδέσμων και επιλογών, αρκετοί προτείνουν ένα συνδυασμό παλιών και νέων διδακτικών μεθόδων και σταδιακή αύξηση του ελέγχου που έχει ο μαθητευόμενος στη διδακτική διαδικασία (ενδεικτικά [26, 38]).

Απ' αυτή τη συνοπτική παρουσίαση των τριών πρώτων γενιών, είναι εύκολο να διαπιστωθούν οι ατέλειές τους με βάση τις αρχές που διατυπώθηκαν. Η πρώτη γενιά θεώρησε τη μάθηση ως μονόδρομο, ως κάτι που εύκολα μπορεί να περιγραφεί, να καταμνηθεί και να αποκτηθεί. Αποκόπτει τη μάθηση από την ατομική και κοινωνική της διάσταση, θεωρώντας την ως κάτι ανεξάρτητο. Ουσιαστικά δεν ανταποκρίνεται σε καμία από τις αρχές που τέθηκαν.

Η δεύτερη και τρίτη γενιά αντιμετώπισαν τη μάθηση πιο σφαιρικά. Ωστόσο, η δεύτερη γενιά έδωσε πολύ μεγάλο βάρος στην αρχή ότι η μάθηση είναι ατομικό φαινόμενο. Επίσης δεν ικανοποιεί την αρχή της απροσδιοριστίας, γιατί οι αλληλεπιδράσεις μαθητή-υπολογιστή είναι σε μεγάλο βαθμό εκ των προτέρων σχεδιασμένες, πράγμα που σημαίνει ότι τελικά οδηγούν τη μάθηση σε συγκεκριμένα κανάλια. Η τρίτη γενιά πλησιάζει πολύ στο να ικανοποιεί και τις τρεις αρχές. Όμως, οι αντιρρήσεις στον εντελώς “ανοικτό” σχεδιασμό αποτρέπουν –παρότι επιδιώκεται– την πλήρη ικανοποίηση της αρχής της απροσδιοριστίας. Απ' αυτή τη σκοπιά, η διαδικασία της μάθησης είναι σχετικά περιορισμένη.

2.1.2 Η τέταρτη γενιά εκπαιδευτικής χρήσης των υπολογιστών-Εικονική Πραγματικότητα

Η τέταρτη γενιά εκπαιδευτικής χρήσης των υπολογιστών, στην οποία ανήκουν οι εφαρμογές Ε.Π., υλοποιεί και αυτή όπως και η προηγούμενη τις βασικές αρχές του δομητισμού. Σύμφωνα όμως με τον Winn [39], η γενιά αυτή προχωρά ένα βήμα παραπέρα. Ο Winn, υιοθετώντας έντονη κριτική απέναντι στις γνωστικές θεωρίες, σημειώνει ότι στις εφαρμογές Ε.Π. δεν είναι απαραίτητη η χρήση συμβόλων. Αυτό το σημείο αξίζει να αναλυθεί παραπάνω.

Οι γνωστικές θεωρίες αντιμετωπίζουν τον ανθρώπινο εγκέφαλο σαν έναν ηλεκτρονικό υπολογιστή που η βασική του λειτουργία είναι ο πνευματικός χειρισμός συμβόλων [40, 41]. Η γνώση δεν είναι παρά ο αποτελεσματικός χειρισμός αυτών των συμβόλων. Όμως έτσι δεν εξηγούνται πλήρως όλες οι πνευματικές διεργασίες, ούτε και ο τρόπος

με τον οποίο γίνεται χειρισμός και αντιμετώπιση καταστάσεων όπου δε γίνεται χρήση συμβόλων. Σε ποιες όμως περιπτώσεις γίνεται και σε ποιες δε γίνεται χρήση συμβόλων;

Οι άνθρωποι αντιλαμβάνονται τον κόσμο με δύο τρόπους, από εμπειρίες “πρώτου προσώπου” και από εμπειρίες “τρίτου προσώπου” [42]. Οι πρώτες προέρχονται από την καθημερινή επαφή με τον κόσμο που περιβάλλει το άτομο, είναι άμεσες, προσωπικές, υποκειμενικές, κατά κάποιον τρόπο “αθόρυβες”, γιατί δε γίνεται αντιληπτό ότι αποκτήθηκε κάποια μορφή γνώσης. Κυρίως όμως λόγω της αμεσότητάς τους με το περιβάλλον, δεν απαιτούν την ύπαρξη συμβόλων. Οι δεύτερες προέρχονται από την περιγραφή κάποιου άλλου, είναι έμμεσες, ρητές και πάντα το άτομο γνωρίζει ότι έμαθε κάτι γιατί του το δίδαξαν. Για να γίνει όμως εφικτή η επικοινωνία μεταξύ αυτού που εξηγεί και αυτού που μαθαίνει, είναι απαραίτητη η ύπαρξη συμβόλων, όπως ο προφορικός λόγος, τα γράμματα και οι αριθμοί.

Για παράδειγμα, εμπειρίες “πρώτου προσώπου” είναι τα συναισθήματα που νιώθει κάποιος όταν παρακολουθεί μία κινηματογραφική ταινία. Εμπειρίες “τρίτου προσώπου” είναι όταν του εξηγούν την υπόθεση, το ρόλο των πρωταγωνιστών σε μία σκηνή, κτλ. Η εμπειρία που βιώνει το άτομο δεν είναι ίδια με την πρώτη περίπτωση, “φιλτράρεται” από κάποιον τρίτο.

Οι εμπειρίες “πρώτου προσώπου” δεν απαιτούν ιδιαίτερη και συντονισμένη σκέψη από το άτομο. Στην πραγματικότητα οι περισσότερες ενέργειες στην καθημερινή ζωή γίνονται έτσι. Κάποιος δε σχεδιάζει εκ των προτέρων πώς θα πλυθεί, πώς θα ντυθεί, πώς θα φάει, πώς θα κοιμηθεί. Απλά το κάνει. Η δράση πηγάζει άμεσα από την αντίληψη που έχει το άτομο για τον κόσμο, χωρίς την παρεμβολή έντονης και συνειδητής σκέψης. Η σκέψη παρεμβάλλεται στη δράση όταν υπάρχει κάποιο δίλημμα ή όταν κάτι πάει λάθος ή όταν χρειάζεται προσοχή για να αντιμετωπιστεί μία κατάσταση.

Οι ηλεκτρονικοί υπολογιστές και στις τρεις πρώτες γενιές προσφέρουν εμπειρίες “τρίτου προσώπου” και αυτό συμβαίνει για τους εξής λόγους:

- Ο υπολογιστής παρεμβάλλεται μεταξύ του ατόμου και των πληροφοριών που περιέχει.
- Οι πληροφορίες αυτές προέρχονται από κάποιον τρίτο, κάποιος τις έχει εισαγάγει στον υπολογιστή.
- Οι πληροφορίες δεν είναι άμεσα διαθέσιμες, αλλά μέσω του *περιβάλλοντος διεπαφής (interface)*, της οθόνης, του ποντικιού, του πληκτρολογίου, που όλα χρησιμοποιούν συστήματα συμβόλων.

- Τα σύμβολα αυτά απαιτούν κάποια σκέψη από το χρήστη, ώστε να μπορεί να τα χρησιμοποιεί αποτελεσματικά.

Συνεπώς, όσο σημαντικός και αν είναι ο ρόλος του τρόπου που παρουσιάζεται η πληροφορία (δεύτερη γενιά), όσο σημαντική και αν είναι η αλληλεπίδραση με αυτή (τρίτη γενιά), οι ηλεκτρονικοί υπολογιστές δεν προσφέρουν εμπειρίες “πρώτου προσώπου”, που όπως έγινε σαφές καταλαμβάνουν σημαντικό μέρος της καθημερινής ζωής. Ο δομητισμός, κατά τον Winn, παρέχει αρκετές ιδέες για το πώς θα μπορούσαν να αποκτηθούν εμπειρίες “πρώτου προσώπου” με τη χρήση ηλεκτρονικών υπολογιστών.

Δύο είναι τα κυριότερα σημεία στα οποία επικεντρώνεται το ενδιαφέρον [39]:

1. Πρέπει να απουσιάζει το *περιβάλλον διεπαφής* υπολογιστή-μαθητή, κατά κάποιον τρόπο ο ηλεκτρονικός υπολογιστής πρέπει να είναι “αόρατος”. Αυτό σημαίνει ότι ο χρήστης-μαθητής θα πρέπει να χρησιμοποιεί για την επικοινωνία του με τον υπολογιστή όχι τις παραδοσιακές συσκευές (ποντίκι, πληκτρολόγιο, οθόνη), αλλά άλλες που να μη χρησιμοποιούν σύμβολα.
2. Στις εφαρμογές αυτές η αλληλεπίδραση μεταξύ ατόμου και υπολογιστή ή μεταξύ ατόμων δεν πρέπει να είναι προσχεδιασμένη, αλλά πρέπει εν δυνάμει να είναι δυνατή κάθε είδους αλληλεπίδραση.

Τέτοια περιβάλλοντα που ανήκουν στο χώρο της Ε.Π. μπορούν να ικανοποιήσουν και τις τρεις αρχές μάθησης. Αυτό γιατί: α) επιτρέπουν κοινωνικές αλληλεπιδράσεις, β) δε χρησιμοποιούν σύμβολα, με αποτέλεσμα οι εμπειρίες να είναι πιο άμεσες σε ατομικό επίπεδο, και γ) εξαιτίας του ανοιχτού σχεδιασμού και των μη προκαθορισμένων αλληλεπιδράσεων η πορεία της μάθησης και τα αποτελέσματά της δεν είναι προκαθορισμένα (*πίνακας 2.1*).

Ως εκ τούτου, η μάθηση σ' ένα περιβάλλον Ε.Π. είναι μία δυναμική διαδικασία καθοριζόμενη από το χρήστη που θέτει τους στόχους και τους μεταβάλλει κατά βούληση.

	Γενιά εκπαιδευτικής χρήσης υπολογιστών			
	1 ^η	2 ^η	3 ^η	4 ^η
Θεωρητικό υπόβαθρο	συμπεριφορισμός	γνωστικές θεωρίες	δομητισμός	δομητισμός
Ρόλος του μαθητή	παθητικός	ενεργός		
Έμφαση	περιεχόμενο	αλληλεπίδραση	αλληλεπίδραση	αλληλεπίδραση, ανοιχτός σχεδιασμός
Εμπειρίες πρώτου προσώπου	όχι	όχι	όχι	ναι
Εμπειρίες τρίτου προσώπου	ναι	ναι	ναι	ναι
Χρήση συμβόλων	ναι	ναι	ναι	όχι
Μάθηση ατομικό φαινόμενο	όχι	ναι	ναι	ναι
Μάθηση κοινωνικό φαινόμενο	όχι	όχι	ναι	ναι
Αρχή απροσδιοριστίας	όχι	όχι	όχι	ναι
Ενδεικτικές εφαρμογές Η/Υ	κείμενα, εκπαιδευτικά παιχνίδια	πολυμέσα, υπερμέσα, εκπαιδευτικά παιχνίδια		εικονική πραγματικότητα

Πίνακας 2.1 Σύγκριση των γενιών εκπαιδευτικής χρήσης των υπολογιστών

Κεφάλαιο 3^ο

Εικονική Πραγματικότητα

Ο όρος “Εικονική Πραγματικότητα” επιδέχεται διάφορες ερμηνείες και ορισμούς και έχει επικρατήσει μεταξύ άλλων όπως “Συνθετικό Περιβάλλον”, “Κυβερνοδιάστημα”, “Τεχνητή Πραγματικότητα”, “Τεχνολογία Προσομοίωσης” [43]. Αν θεωρηθεί από καθαρά τεχνολογική σκοπιά, η Ε.Π. είναι ένα σύνολο υλικού (ηλεκτρονικοί υπολογιστές και ειδικές συσκευές) και λογισμικού (προγράμματα γραφικών και κίνησης και ειδικά προγράμματα κατασκευής εικονικών κόσμων) με το οποίο οι άνθρωποι είναι σε θέση να οπτικοποιούν και να αλληλεπιδρούν με εξαιρετικά περίπλοκα δεδομένα στις τρεις διαστάσεις.

Η δυνατότητα των υπολογιστών δεν περιορίζεται στο να δίνουν στο χρήστη μόνο οπτικά ερεθίσματα, αλλά επεκτείνεται και στα ακουστικά και στα απτικά ερεθίσματα. Ο χρήστης μπορεί να αλληλεπιδρά μ' έναν εικονικό, μη-πραγματικό κόσμο και να χειρίζεται άμεσα τα αντικείμενά του [44]. Έτσι, στην ουσία, η Ε.Π. μπορεί να οριστεί ως ένα υψηλής τεχνολογίας μέσο διασύνδεσης ανθρώπου-υπολογιστή που περιλαμβάνει προσομοίωση πραγματικού χρόνου και αλληλεπιδράσεις μέσα από πολλαπλά αισθητηριακά κανάλια. Ένα πρώτο θέμα που προκύπτει είναι ότι ο σχεδιασμός και η υλοποίηση ενός συστήματος Ε.Π. πρέπει να τοποθετήσει τον άνθρωπο στο κέντρο του συστήματος. Η μηχανή πρέπει να είναι συμβατή με τον τρόπο που ο άνθρωπος αντιλαμβάνεται και δρα, και όχι το αντίστροφο [45].

Με βάση την παραπάνω παρατήρηση, ο καθαρά τεχνολογικός ορισμός της Ε.Π. δεν αναφέρεται στις νοητικές διεργασίες και στις επιπτώσεις στον άνθρωπο. Συνυπολογίζοντας λοιπόν και τον ανθρώπινο παράγοντα, μπορεί να διευρυνθεί ο όρος “Εικονική Πραγματικότητα”, υιοθετώντας τον ορισμό που δίνουν οι Macpherson και Kerrill [46] ότι Ε.Π. είναι “... μία κατάσταση που δημιουργείται στο μυαλό και που μπορεί, με μεταβαλλόμενο ποσοστό επιτυχίας, να απασχολεί την προσοχή ενός ανθρώπου με τρόπο παρόμοιο με αυτόν στο πραγματικό περιβάλλον. Οι συσκευές που χρησιμοποιούνται συμβάλλουν στη δημιουργία αυτής της κατάστασης”.

Με άλλα λόγια, η Ε.Π. δεν είναι απλώς και μόνο μία νέα τεχνολογία υπολογιστών. Είναι μία νοητική κυρίως κατάσταση στην οποία το υποκείμενο-χρήστης βυθίζεται –μερικώς ή ολικώς– σ' ένα τεχνητό περιβάλλον, το οποίο μπορεί να έχει μεγάλες ομοιότητες ή διαφορές από την πραγματικότητα. Ένα ενδιαφέρον σημείο αυτού του ορισμού είναι ότι ως Ε.Π. μπορεί να θεωρηθεί και ένα βιβλίο, ένα παιχνίδι ή οποιοδήποτε άλλο μέσο επιτρέπει στο χρήστη να βυθιστεί σ' ένα μη πραγματικό περιβάλλον.

Από ψυχολογικής άποψης, η Ε.Π. είναι ένα εργαλείο που επιτρέπει στους χρήστες να βιώσουν την *τηλε-παρουσία*. Η *τηλε-παρουσία* αναφέρεται στο βαθμό που κάποιος αισθάνεται “παρών” σ' ένα περιβάλλον φτιαγμένο από υπολογιστή όπως αισθάνεται

“παρών” και στο φυσικό περιβάλλον. Με αυτό το σκεπτικό, η Ε.Π. ορίζεται ως ένα αντιληπτικό εργαλείο που αποτελείται από τεχνητούς κόσμους φτιαγμένους σε υπολογιστή, που ανταποκρίνονται στις κινήσεις και γενικά στην είσοδο δεδομένων από το χρήστη [47].

Ο Jaron Lanier —ο πρώτος που χρησιμοποίησε τον όρο “Εικονική Πραγματικότητα”— διακρίνει τις φιλοσοφικές διαστάσεις της εμπειρίας σ' ένα περιβάλλον Ε.Π. [48]. Βλέπει αυτή την τεχνολογία “... σαν μία διαδρομή προς ένα είδος κοινωνικής ένωσης, ότι δίνει τη δυνατότητα να απελευθερωθούν τα δεσμά από την πηγή της εμπειρίας”. Ακόμα πιστεύει ότι η Ε.Π. μπορεί να επεκτείνει την αντίληψη που έχουν οι άνθρωποι για τον κόσμο, πραγματικό και συμβολικό, ότι προσφέρει εμπειρίες από εναλλακτικούς κόσμους. Καταργώντας τα σύνορα του φυσικού κόσμου και μπαίνοντας μέσα σ' ένα συμβολικό κόσμο, επιστρέφοντας στον φυσικό, η ικανότητα να κατανοηθεί και ο φυσικός και ο συμβολικός κόσμος διευρύνεται.

Τέλος, από γνωστικής άποψης, η Ε.Π. θεωρείται ως είναι ένα μείγμα υλικού, λογισμικού και ιδεών, που δημιουργούν το φαινόμενο της *γνωστικής παρουσίας* [49]. Δημιουργείται δηλαδή η πεποίθηση της εγκυρότητας του εικονικού κόσμου, αν και διαφορετικού από τον πραγματικό. Ως κατάσταση είναι παρόμοια με ό,τι συμβαίνει όταν κάποιος βλέπει μία ταινία, ξέρει ότι δεν είναι πραγματικά όσα του παρουσιάζονται και όμως συμμετέχει σαν να είναι πραγματικά.

Βέβαια ο όρος Ε.Π. έχει δεχτεί επικρίσεις, που στηρίζονται κατά πρώτον στο γεγονός ότι τα όρια της Ε.Π. δεν είναι ευδιάκριτα και πολλές φορές υπάρχει σύγχυση ανάμεσα στην Ε.Π. και σε άλλες εφαρμογές που χρησιμοποιούν τρισδιάστατα γραφικά [50]. Η απλή αναπαράσταση αντικειμένων στον υπολογιστή στις τρεις διαστάσεις δεν ανάγει αυτόματα την εφαρμογή σε Ε.Π. Για παράδειγμα, για πολλά χρόνια οι αρχιτέκτονες και οι μηχανικοί χρησιμοποιούν προγράμματα *τρειςδιάστατης απεικόνισης κατασκευών* (*Computer Aided Design, CAD*), που καμία σχέση δεν έχουν με την Ε.Π. Ούτε επίσης τα *κινούμενα τρισδιάστατα γραφικά* (*computer animation*) έχουν καμία σχέση με την Ε.Π., γιατί ο χρήστης δεν έχει κανέναν έλεγχο του τι συμβαίνει.

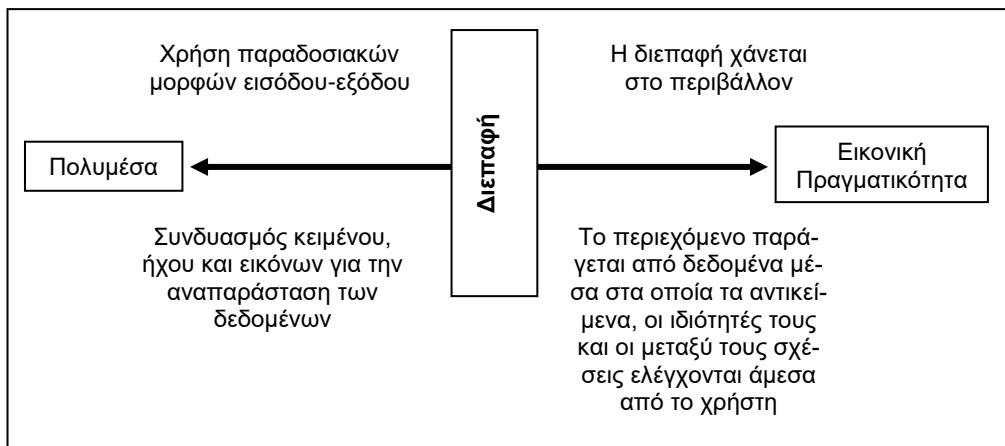
Μία δεύτερη ένσταση αφορά το γεγονός ότι η τεχνολογία δεν μπορεί να προσφέρει ακόμα οπτικά, ακουστικά και απτικά ερεθίσματα τέτοιας ποιότητας που ο χρήστης να ταυτίζει τον εικονικό με τον πραγματικό κόσμο [50]. Αυτό βέβαια μπορεί να μην έχει και τόση σημασία, αν ληφθεί υπόψη ότι ο χρήστης ξέρει ούτως ή άλλως ότι αυτά που βλέπει δεν είναι πραγματικά, αλλά λειτουργεί το φαινόμενο της *γνωστικής παρουσίας*.

3.1 Βασικά χαρακτηριστικά της Εικονικής Πραγματικότητας

Ποια όμως είναι εκείνα τα χαρακτηριστικά που διαφοροποιούν την Ε.Π. από τις άλλες εφαρμογές των ηλεκτρονικών υπολογιστών, πέρα από τη χρήση τρισδιάστατων γραφικών; Σε προηγούμενη ενότητα έγινε αναφορά σε δύο απ' αυτά, το περιβάλλον διεπαφής και το βαθμό αλληλεπίδρασης.

Η διεπαφή είναι ουσιαστικά το μέσο επικοινωνίας με τον ηλεκτρονικό υπολογιστή, ο τρόπος με τον οποίο εισάγονται δεδομένα σ' αυτόν (πληκτρολόγιο, ποντίκι, κτλ.) και ο τρόπος που αυτός εξάγει δεδομένα σε μορφή κατανοητή προς τον άνθρωπο (οθόνη, εκτυπωτής, κτλ.). Όλες αυτές οι συσκευές χρησιμοποιούν συστήματα συμβόλων. Όμως, με όσο περισσότερο φυσικό τρόπο γίνεται αυτή η αμφίδρομη επικοινωνία, όσο δηλαδή η επικοινωνία μοιάζει με τον τρόπο που επικοινωνούν οι άνθρωποι μεταξύ τους ή με το περιβάλλον, τόσο πιο πολύ χάνεται η διεπαφή μέσα στην ίδια την εφαρμογή.

Για παράδειγμα, αν για την κίνηση του χρήστη αντί για το πληκτρολόγιο και το ποντίκι χρησιμοποιηθούν συσκευές που μετατρέπουν το φυσικό περπάτημα σε κίνηση μέσα στην εφαρμογή, τότε η αντίστοιχη διεπαφή έχει χαθεί. Είναι προφανές ότι και πάλι χρησιμοποιούνται συσκευές, αλλά ο τρόπος λειτουργίας τους μιμείται το φυσικό τρόπο. Η απουσία περιβάλλοντος διεπαφής φαίνεται ότι είναι μοναδικό χαρακτηριστικό της Ε.Π. Οι Hedberg και Alexander [51], θέλοντας να διαχωρίσουν την Ε.Π. από τις άλλες μορφές πολυμέσων, τοποθετούν τη διεπαφή στο μέσο του φάσματος και στα δύο άκρα τις εφαρμογές αυτές (σχήμα 3.1). Από το σχήμα μπορεί να εξαχθεί το συμπέρασμα ότι δεν είναι απαραίτητο να απουσιάζει εντελώς το περιβάλλον διεπαφής για να θεωρηθεί ότι μία εφαρμογή ανήκει στο χώρο της Ε.Π.



Σχήμα 3.1 Η σχέση πολυμέσων και Ε.Π. αναφορικά με το περιβάλλον διεπαφής

Η αλληλεπίδραση, ή καλύτερα ο υψηλός βαθμός αλληλεπίδρασης, είναι ένα άλλο χαρακτηριστικό στοιχείο των εφαρμογών Ε.Π. Υψηλός βαθμός αλληλεπίδρασης συνεπάγεται ότι το εικονικό περιβάλλον επιτρέπει την ελεύθερη πλοήγηση μέσα σ' αυτό, την αλλαγή της θέσης παρατήρησης και το χειρισμό των αντικειμένων που περιλαμβάνει [52].

Ο χειρισμός των αντικειμένων μπορεί να γίνεται με το ποντίκι (εμφανές περιβάλλον διεπαφής) ή με άλλες συσκευές που μιμούνται τον τρόπο με τον οποίο γίνεται ο χειρισμός των αντικειμένων στο φυσικό κόσμο (απουσία περιβάλλοντος διεπαφής). Ο δεύτερος τρόπος αλληλεπίδρασης επιτρέπει μάλιστα στους αρχάριους χρήστες να προσαρμόζονται εύκολα και στους έμπειρους να μπορούν να αναπαραστήσουν και να χειριστούν ολόένα και πιο σύνθετες πληροφορίες.

Ένα άλλο χαρακτηριστικό των εφαρμογών Ε.Π. είναι ο τρόπος, η προοπτική με την οποία βλέπει ο χρήστης το εικονικό περιβάλλον. Μία προοπτική “τρίτου προσώπου” σημαίνει ότι ο χρήστης βλέπει μία φιγούρα που τον αναπαριστά μέσα στην εφαρμογή. Τη φιγούρα αυτή τη χειρίζεται με σκοπό να πλοηγηθεί ή να χειριστεί άλλα αντικείμενα. Η προοπτική “τρίτου προσώπου” είναι ο συνηθέστερος τρόπος απεικόνισης του χρήστη στις εφαρμογές πολυμέσων.

Στην προοπτική “πρώτου προσώπου”, που συνηθίζεται στις εφαρμογές Ε.Π., ο χρήστης δε βλέπει αυτή τη φιγούρα, καθώς έχει πιο άμεση επαφή με το περιβάλλον. Όμως ο χρήστης μπορεί και βλέπει τις φιγούρες των άλλων χρηστών στην περίπτωση που ο εικονικός κόσμος επιτρέπει τη συμμετοχή περισσότερων του ενός. Μπορεί επίσης να υπάρχουν και μία ή παραπάνω φιγούρες ελεγχόμενες από τον υπολογιστή που παίζουν το ρόλο του καθοδηγητή ή του βοηθού μέσα στον εικονικό κόσμο. Ο όρος που χρησιμοποιείται για τις φιγούρες αυτές είναι *avatars* και προέρχεται από τον ινδουισμό [53].

Αυτή η αμεσότητα της επαφής του χρήστη με το εικονικό περιβάλλον εξαιτίας της προοπτικής “πρώτου προσώπου” είναι ένας από τους παράγοντες που οδηγούν στο πιο σημαντικό αλλά και αντιφατικό στοιχείο των εφαρμογών Ε.Π., την εμβύθιση. Με τον όρο “εμβύθιση” εννοείται το φαινόμενο κατά το οποίο οι αισθήσεις του χρήστη παραινώνται σε τέτοιο βαθμό, ώστε να πιστεύει ότι βρίσκεται πραγματικά μέσα στον εικονικό κόσμο. Συνεπώς, θεωρώντας την εμβύθιση από καθαρά τεχνική σκοπιά, δύο είναι οι παράμετροι που μπορούν να την επηρεάσουν. Πρώτη παράμετρος είναι ο ρεαλισμός του εικονικού κόσμου, δηλαδή πόσο πιστά αποδίδει έναν πραγματικό ή φανταστικό κόσμο, και σχετίζεται με την ποιότητα των τρισδιάστατων γραφικών της ε-

φαρμογής. Η δεύτερη παράμετρος είναι το πώς μεταφέρεται η αναπαράσταση αυτή στις αισθήσεις του χρήστη και σχετίζεται με τις συσκευές που χρησιμοποιούνται.

Πρέπει να επισημανθεί ότι δεν είναι ακόμα δυνατή η εμπλοκή και των πέντε αισθήσεων (εξαιρούνται η γεύση και η όσφρηση, αν και γίνονται δοκιμές συσκευών που απευθύνονται στην όσφρηση) σ' ένα εικονικό περιβάλλον. Για το λόγο αυτό δεν υπάρχει απόλυτη εμπύθιση. Ωστόσο, επειδή η όραση και η ακοή είναι οι πιο σημαντικές αισθήσεις, παραπλανώντας τις με κατάλληλες συσκευές επιτυγχάνεται ένας υψηλός βαθμός εμπύθισης.

Μελέτες αποδεικνύουν ότι όσο μεγαλύτερος είναι ο βαθμός εμπύθισης, τόσο καλύτερα γνωστικά αποτελέσματα επιτυγχάνονται [54, 55]. Υπάρχουν όμως και μελέτες που δείχνουν το αντίθετο, ότι η εμπύθιση δηλαδή δεν παίζει τόσο σημαντικό ρόλο [56, 57]. Υπάρχει μία ακόμα πτυχή στο θέμα. Όταν δόθηκαν οι ορισμοί της Ε.Π., αναφέρθηκε ότι η Ε.Π. είναι μία νοητική κατάσταση. Αυτό υπαινίσσεται ότι μπορεί να μην είναι απαραίτητη η χρήση εξειδικευμένων συσκευών για να επιτευχθεί η εμπύθιση. Πράγματι, ψυχολογικές και ψυχοσωματικές διαφορές μεταξύ των ατόμων παίζουν σημαντικό ρόλο. Η “παρουσία” στον εικονικό κόσμο εξαρτάται από ψυχολογικούς παράγοντες όπως η φαντασία, η συγκέντρωση, η προσοχή και ο αυτοέλεγχος [58]. Εξαρτάται επίσης από την εξοικείωση του χρήστη με το πραγματικό και το συνθετικό περιβάλλον [59]. Τέλος, στους ψυχοσωματικούς παράγοντες εντάσσονται η ηλικία και οι σωματικές ή νοητικές αδυναμίες [60].

Τα πειραματικά δεδομένα λοιπόν οδηγούν σε αντιφατικά συμπεράσματα αναφορικά με τη σημασία του βαθμού εμπύθισης στην επίτευξη συγκεκριμένων εκπαιδευτικών στόχων. Αυτό αφήνει τα περιθώρια για την υπόθεση ότι η αυξημένη αλληλεπίδραση με το εικονικό περιβάλλον έχει μεγαλύτερη σημασία ή ότι ο συνδυασμός των δύο παραγόντων προσδίδουν την ιδιαίτερη αξία της Ε.Π. Σε κάθε περίπτωση, αυτό που είναι δεδομένο είναι ότι μία εφαρμογή δε χρειάζεται να τροποποιηθεί σημαντικά για να αλλάξει το είδος εμπύθισης στο οποίο ανήκει. Αυτό που μεταβάλλεται είναι οι συσκευές που χρησιμοποιούνται. Συνεπώς υπάρχει η ανάγκη να εξεταστούν οι συσκευές αυτές και τα διάφορα είδη Ε.Π.

3.2 Υλικό και είδη Εικονικής Πραγματικότητας

Αν και εκτενής τεχνική ανάλυση θα γίνει σε επόμενο κεφάλαιο, ωστόσο επιγραμματικά μπορούν να αναφερθούν οι κύριες κατηγορίες υλικού και λογισμικού που συγκροτούν ένα σύστημα Ε.Π., που είναι:

Υλικό (Hardware):

- Συσκευές εισόδου-εξόδου πληρο-φοριών.
- Ηλεκτρονικοί υπολογιστές.

Λογισμικό (Software):

- Προγράμματα τρισδιάστατης μοντελοποίησης.
- Προγράμματα επεξεργασίας γραφικών και ήχου.
- Προγράμματα κατασκευής περιβαλόντων Ε.Π.

Στο σημείο αυτό θα αναφερθούν μόνο οι συσκευές εισόδου-εξόδου, γιατί αυτές είναι που καθορίζουν το βαθμό εμπύθισης, που με τη σειρά του καθορίζει τα διάφορα είδη Ε.Π.

3.2.1 Συσκευές εισόδου-εξόδου

Για το χειρισμό των αντικειμένων του εικονικού κόσμου χρησιμοποιούνται ειδικά γάντια (*datagloves*) στα οποία υπάρχουν ενσωματωμένοι αισθητήρες για τον εντοπισμό της σχετικής θέσης του κάθε δακτύλου ξεχωριστά, αλλά και της παλάμης ως σύνολο (*εικόνα 3.1*). Ο χρήστης μπορεί να αγγίξει, να μετακινήσει, να περιστρέψει και γενικά να μεταχειριστεί τα αντικείμενα όπως θα έκανε αν αυτά ήταν πραγματικά. Κατά την πλοήγησή του μέσα στον εικονικό κόσμο βλέπει την αναπαράσταση των χεριών του, τα οποία μετακινούνται σύμφωνα με τις κινήσεις που κάνει.

Για τη δημιουργία της αίσθησης της αφής υπάρχουν στο γάντι διατάξεις που ασκούν πίεση ανάλογη με τη δύναμη που εφαρμόζει ο χρήστης όταν μεταχειρίζεται ένα αντικείμενο. Παράλληλα, είναι δυνατή η αυξομείωση της θερμοκρασίας ανάλογα με τη θερμοκρασία του αντικειμένου που αγγίζει. Σε πιο εξελιγμένες εφαρμογές, προσαρμόζονται στο χρήστη διατάξεις με τη μορφή εξωσκελετού που μπορούν να καλύπτουν τα χέρια, τα πόδια ή και το σώμα ολόκληρο, δίνοντας πληροφορίες από και προς τα αντίστοιχα μέρη του σώματος.

Η πλοήγηση στο χώρο είναι μία ιδιαίτερη πρόκληση και υλοποιείται αρκετά δύσκολα. Συνήθως σε απλές εφαρμογές χρησιμοποιείται ένα χειριστήριο ή *ραβδί* (*wand*), που όταν ο χρήστης το στρέφει, περιστρέφεται και μετακινείται ο εικονικός κόσμος προς τη συγκεκριμένη κατεύθυνση. Ο χρήστης όμως παραμένει ακίνητος. Σε πιο εξελιγμένες εφαρμογές χρησιμοποιείται ένας κυλιόμενος διάδρομος, ειδικές κινούμενες καρέκλες, ή ακόμα και μία ολόκληρη καμπίνα που μπορεί να κινείται με έξι βαθμούς ελευθερίας (πάνω-κάτω, δεξιά-αριστερά, μπροστά-πίσω).

Το κεφάλι είναι αυτό που συνήθως κινείται προς την κατεύθυνση που θέλει να κοιτάξει κάποιος. Έτσι χρησιμοποιούνται συσκευές που προσαρμόζονται στο κεφάλι του χρή-

στη και εντοπίζουν τη θέση του (*trackers*). Στη συνέχεια δίνουν τις ανάλογες εντολές στην εφαρμογή ώστε να απεικονίσει το περιβάλλον προς την κατεύθυνση που κοιτά ο χρήστης.

Ο τομέας της οπτικής απεικόνισης προσφέρει τη μεγαλύτερη ποικιλία συσκευών. Σε πολύ απλές εφαρμογές χρησιμοποιείται η οθόνη του υπολογιστή, που όμως δεν καλύπτει ολόκληρο το οπτικό πεδίο του χρήστη. Επόμενο βήμα είναι η χρήση γυαλιών, που αποτελούνται από δύο μικρές οθόνες υγρών κρυστάλλων οι οποίες μπορούν να καλύπτουν πλήρως το οπτικό πεδίο (*Head Mounted Displays, H.M.D.*). Έτσι ο χρήστης δέχεται οπτικά ερεθίσματα μόνο από το εικονικό περιβάλλον και είναι πλήρως αποκομμένος από τον εξωτερικό κόσμο (εικόνα 3.1). Μία άλλη προσέγγιση στον τομέα της απεικόνισης είναι η τοποθέτηση του χρήστη μέσα σ' ένα χώρο όπου ολόγυρά του προβάλλεται, με βιντεοπροβολείς, ο εικονικός κόσμος. Τα συστήματα αυτά ονομάζονται *C.A.V.E.* (*Cave Automatic Virtual Environment*).

Μία ιδιαιτερότητα των συστημάτων *C.A.V.E.* αλλά και των *H.M.D.* είναι η δυνατότητα στερεοσκοπικής προβολής της εικόνας (η εικόνα αποτελείται από δύο ίδιες εικόνες με ελαφρά απόκλιση μεταξύ τους). Ο χρήστης, φορώντας ειδικά γυαλιά, ανασυνθέτει τις δύο εικόνες σε μία, με αποτέλεσμα να δημιουργείται η αίσθηση του βάθους, και έτσι ο εικονικός κόσμος γίνεται πραγματικά τρισδιάστατος. Να σημειωθεί ότι η τεχνική της στερεοσκοπικής προβολής είναι δυνατό να εφαρμοστεί και σε μία απλή οθόνη υπολογιστή.

Ο τομέας του ήχου καλύπτεται από ακουστικά, που συνήθως είναι ενσωματωμένα στα γυαλιά και προσφέρουν στερεοφωνικό ήχο. Ο χρήστης μπορεί να έχει στη διάθεσή του ένα μικρόφωνο για την επικοινωνία του με άλλους χρήστες, ενώ με το κατάλληλο υλικό ή λογισμικό αναγνώρισης φωνής μπορεί να δίνει φωνητικές εντολές στο σύστημα αντί να χρησιμοποιεί το πληκτρολόγιο.

Με την παράθεση όλων αυτών των συσκευών, εύκολα μπορεί κάποιος να διαπιστώσει ότι προκύπτουν σοβαρά προβλήματα στο χρήστη που θα θελήσει να τις χρησιμοποιήσει όλες ταυτόχρονα για να πετύχει το μέγιστο βαθμό εμπύθισης. Ο όγκος, το βάρος τους, τα καλώδια και η συνδεσμολογία τους δημιουργούν προβλήματα στην κίνηση, κούραση και δυσφορία, ενώ έχουν αναφερθεί και περιστατικά προβλημάτων υγείας έπειτα από πολύωρη χρήση των γυαλιών (ενδεικτικά [61, 62]). Για το λόγο αυτόν, καταβάλλονται σημαντικές προσπάθειες στους τομείς της εργονομίας των συσκευών, του βάρους και του όγκου τους, ενώ πολλές απ' αυτές είναι ασύρματες, ώστε να μειωθεί ο αριθμός των καλωδίων.



Εικόνα 3.1 Συσσκευές Ε.Π.

3.2.2 Είδη Εικονικής Πραγματικότητας

Με βάση τις παραπάνω συσκευές, χρησιμοποιώντας όλες, μερικές ή καμία απ' αυτές, προκύπτουν τα διάφορα είδη Ε.Π., που είναι:

Δικτυακή-βασισμένη σε κείμενο Ε.Π. (*M.U.D.'s, Multi-User Dungeons* και *M.O.O.'s, MUD Object Oriented*). Είναι εικονικοί κόσμοι πραγματικού χρόνου που στηρίζονται στη χρήση κειμένου παρά γραφικών. Επιτρέπουν τη συμμετοχή μεγάλου αριθμού χρηστών και αρχικά αναπτύχθηκαν για παιχνίδια σε δίκτυα ευρείας περιοχής. Δημιουργούνται με τον τρόπο αυτό “κοινωνίες” παικτών. Ο κάθε παίκτης αναλαμβάνει και ένα “ρόλο” και αναπτύσσονται κώδικες κοινωνικής συμπεριφοράς. Ενθαρρύνεται το γράψιμο και η ανάγνωση. Από την άλλη όμως πλευρά, η έλλειψη συγκεκριμένης δομής και γνωστικών στόχων μπορεί να αφήσει ορισμένους χρήστες να περιπλανώνται στον εικονικό κόσμο χωρίς συγκεκριμένο σκοπό. Η απλή χρήση κειμένου δεν είναι αρκετή ώστε να ικανοποιεί τις παραμέτρους της εκπαιδευτικής εμπειρίας. Η απουσία οπτικοακουστικής παρουσίασης του κόσμου σε συνδυασμό με τα παραπάνω μάλλον καθιστούν αυτό το είδος Ε.Π. ένα ημιτελές εκπαιδευτικό περιβάλλον.

Επιτραπέζια Ε.Π. (*Desktop Virtual Reality*). Θεωρείται από πολλούς ως η φυσική συνέχεια των εφαρμογών πολυμέσων. Το εικονικό περιβάλλον παρουσιάζεται σ' έναν κοινό ηλεκτρονικό υπολογιστή. Είναι επίσης δυνατή η τρισδιάστατη παρουσίαση του περιβάλλοντος με τη χρήση ειδικών στερεοσκοπικών γυαλιών. Το κύριο πλεονέκτημα αυτής της μορφής Ε.Π. είναι το σχετικά μικρό και προσιτό κόστος. Η χρήση μίας απλής οθόνης υπολογιστή επιτρέπει υψηλή ανάλυση και παρέχει καλή ποιότητα γραφικών, σε αντίθεση με τη χαμηλότερη απόδοση των *H.M.D.* Σε συνδυασμό με το Διαδίκτυο ή ένα τοπικό δίκτυο, προσθέτει τη δυνατότητα πρόσβασης σε δικτυακούς κόσμους όπως στην παραπάνω κατηγορία, που στηρίζονται όμως στη χρήση γραφικών. Το βασικό μειονέκτημα είναι η αρκετά μειωμένη αίσθηση της εμβύθισης από την

πλευρά του χρήστη. Αυτό είναι σημαντικό, αφού η ποιότητα της εμπύθισης ίσως να καθορίζει και τα γνωστικά αποτελέσματα.

Ημι-εμβυθισμένη ή προβαλλόμενη εικονική πραγματικότητα (*Semi immersive/Projected Virtual Reality*): Αυτή η κατηγορία χαρακτηρίζεται από την προβολή της εικόνας σε οθόνη όπως στον κινηματογράφο ή σε χώρο όπου η εικόνα προβάλλεται ολόγυρά του (C.A.V.E.) Αυτό δίνει στους συμμετέχοντες μία αρκετά καλή αίσθηση εμπύθισης. Επίσης υπάρχει ομαδική συμμετοχή, πράγμα σημαντικό για την εκπαιδευτική διαδικασία. Μειονέκτημα αυτής της κατηγορίας είναι το γεγονός ότι λόγω της ομαδικής συμμετοχής η προσοχή των συμμετεχόντων μπορεί να αποσπαστεί εύκολα. Το κόστος τέτοιων συστημάτων είναι σημαντικό.

Πλήρως εμβυθισμένη εικονική πραγματικότητα (*Fully-immersive Virtual Reality*): Ο χρήστης απομονώνεται από το φυσικό περιβάλλον και “υπάρχει” μόνο μέσα στο εικονικό. Όλες του οι κινήσεις λαμβάνουν χώρα μέσα σ’ αυτό και απ’ αυτό αντλεί οπτικοακουστικά ερεθίσματα. Είναι απαραίτητο η προβολή εικόνας να γίνεται σ’ όλο το οπτικό πεδίο του χρήστη ώστε να έχει μπροστά του μόνο το εικονικό περιβάλλον. Συνήθως χρησιμοποιούνται *H.M.D.* Με άλλες συσκευές γίνεται ανίχνευση της θέσης, των ενεργειών και των κινήσεων του χρήστη (*συσκευές εντοπισμού θέσης, trackers*). Ο χρήστης μπορεί να αλληλεπιδρά με αντικείμενα του κόσμου όπως και στην πραγματικότητα μέσω ειδικών γαντιών (*datagloves*). Έτσι ο βαθμός αλληλεπίδρασης με το εικονικό περιβάλλον είναι σημαντικός.

Σε ορισμένες περιπτώσεις προκύπτουν θέματα υγείας από τη χρήση αυτών των συσκευών, κυρίως από τη χρήση των *H.M.D.* Το κόστος τέτοιων συστημάτων είναι εξαιρετικά υψηλό και ευθέως ανάλογο της ποιότητας που προσφέρουν. Αυτό γιατί η τεχνολογία που χρησιμοποιείται έχει ορισμένους περιορισμούς ταχύτητας προβολής και ποιότητας της προβαλλόμενης εικόνας. Για παράδειγμα, συσκευές που προσφέρουν ασήμαντη καθυστέρηση στην ενημέρωση του προβαλλόμενου περιβάλλοντος με βάση τις κινήσεις και τις ενέργειες του χρήστη και άριστη ποιότητα εικόνας, κοστίζουν αρκετές χιλιάδες ευρώ.

Μεικτή ή επαυξημένη πραγματικότητα (*Augmented Reality*): Είναι η ανάμειξη της Ε.Π. με την πραγματικότητα. Ο χρήστης βρίσκεται σ’ έναν εικονικό κόσμο, στον οποίο όμως προβάλλονται πραγματικά αντικείμενα τα οποία μπορεί να χειριστεί ακόμα και αν βρίσκονται σε μεγάλη απόσταση ή σε εξαιρετικά επικίνδυνα φυσικά περιβάλλοντα, για παράδειγμα συσκευές ελέγχου στην καρδιά ενός πυρηνικού αντιδραστήρα ή στο διάστημα. Λόγω των ιδιαίτερων μεθόδων υλοποίησης αυτού του είδους Ε.Π., το κυριότερο μειονέκτημά της, πέρα από το κόστος, είναι η δυσκολία ακριβούς εντοπισμού της

θέσης τόσο του αντικειμένου όσο και του χρήστη, ειδικά σε εφαρμογές που η ακρίβεια χειρισμού είναι απολύτως απαραίτητη.

3.3 Προβληματισμός για την επιλογή συγκεκριμένου είδους Εικονικής Πραγματικότητας για εκπαιδευτική χρήση

Το βασικό συμπέρασμα που προκύπτει από την παραπάνω παρουσίαση είναι ότι ο βαθμός της εμπύθισης είναι συνδεδεμένος κυρίως με το μέσο απεικόνισης που επιλέγεται. Η επιτραπέζια Ε.Π. χρησιμοποιεί απλώς την οθόνη του υπολογιστή. Αν επιλεγεί η χρήση των γυαλιών, αυτόματα αυξάνεται ο βαθμός εμπύθισης. Οι υπόλοιπες συσκευές μπορούν να χρησιμοποιηθούν ανεξάρτητα από το μέσο απεικόνισης.

Στο σημείο αυτό τίθεται ένα ουσιαστικό πρόβλημα. Ποιο είδος Ε.Π. είναι το καταλληλότερο για εκπαιδευτική χρήση; Η απόφαση για τη χρήση κάποιας απλής μορφής Ε.Π., όπως η επιτραπέζια, λειτουργεί σε βάρος της εμπύθισης που μπορεί να είναι μία σημαντική παράμετρος. Αν αποφασιστεί η χρήση άλλου είδους Ε.Π. προκύπτουν άλλης μορφής προβλήματα (τεχνικά, οικονομικά, υγείας).

Στο μυαλό των περισσότερων η Ε.Π. είναι συνδεδεμένη με τη χρήση των συσκευών που αναφέρθηκαν και γενικά θεωρούν την επιτραπέζια Ε.Π. “φτωχό συγγενή” της Ε.Π. Ανατρέχοντας όμως στους ορισμούς της Ε.Π., διαπιστώνεται η σημασία της αντιμετώπισης της Ε.Π. ως νοητικής κατάστασης. Πράγματι, επειδή η εμπύθιση είναι κυρίως μία εμπειρία, ίσως θα ήταν προτιμότερο να υιοθετηθεί μία αντιληπτική και όχι τεχνολογική προσέγγιση της Ε.Π. Με αυτή την έννοια, το ενδιαφέρον εστιάζει στις γνωστικές, συναισθηματικές και κοινωνικές εμπειρίες του χρήστη, με αποτέλεσμα η τεχνολογία να μην είναι τίποτα παραπάνω από ένα ακόμα εργαλείο δημιουργίας, αναπαραγωγής και ενίσχυσης αυτών των εκπαιδευτικών εμπειριών [63].

Τέλος, πρέπει να γίνει μία ακόμα παρατήρηση. Πέρα από το σημαντικό κόστος των εξειδικευμένων συσκευών, αυτές απέχουν αρκετά από το να χαρακτηριστούν αψεγάδιαστες. Η ποιότητα εικόνας που προσφέρουν υστερεί σε σχέση με την ποιότητα που προσφέρει η οθόνη ενός απλού ηλεκτρονικού υπολογιστή, ενώ ο όγκος και η συνδεσιμολογία τους τελικά τις κάνει δύσχρηστες. Ενώ δηλαδή η επιδίωξη είναι να γίνει η χρήση της συγκεκριμένης τεχνολογίας εύκολη και φιλική, το αποτέλεσμα είναι το αντίθετο. Με αυτό το σκεπτικό, θα πρέπει κανείς να αναλογιστεί τις συγκεκριμένες ανάγκες του χρήστη από μία εφαρμογή και τις εργασίες που θα κάνει με αυτή, απ’ όπου μπορεί να προκύψει ότι η μεγάλη εμπύθιση μπορεί να είναι ή να μην είναι χρήσιμη και απαραίτητη [64].

Είναι αλήθεια ότι είναι πολύ δύσκολο και οικονομικά ασύμφορο να εξοπλιστεί μία ολόκληρη τάξη με εξειδικευμένες συσκευές Ε.Π. Επίσης, τα προβλήματα που παρουσιάζονται κατά τη χρήση τους, όπως αναφέρθηκε, εγείρουν επιφυλάξεις τουλάχιστον μέχρις ότου υπάρξουν σημαντικές βελτιώσεις. Απ' αυτή τη σκοπιά, η επιτραπέζια Ε.Π. είναι τεχνολογικά πιο ώριμη και συνεπώς πιο άμεσα εφαρμόσιμη από την πλήρως εμβυθισμένη.

3.4 Θεωρίες μάθησης που σχετίζονται με την Εικονική Πραγματικότητα

Μένει να απαντηθεί το ερώτημα κατά πόσο η Ε.Π. είναι συμβατή με τις θεωρίες μάθησης, με ποιες και σε ποιο βαθμό. Όπως έχει ήδη διαφανεί από την ανάλυση για τις αρχές μάθησης και τις γενιές εκπαιδευτικής χρήσης των υπολογιστών, η Ε.Π. είναι πολύ κοντά στις αρχές του δομητισμού. Πράγματι, η ανάλυση που ακολουθεί ερμηνεύει γιατί η Ε.Π. είναι ιδιαίτερα συμβατή με τη μαθησιακή διαδικασία έτσι όπως την προσεγγίζει ο δομητισμός.

Επειδή έχει γίνει ήδη εκτενής αναφορά στις θεωρίες μάθησης στο 1^ο κεφάλαιο, στο σημείο αυτό το ενδιαφέρον στρέφεται κυρίως στην τεχνολογική πλευρά, πώς δηλαδή ο δομητισμός σχετίζεται γενικά με την τεχνολογία και με την Ε.Π. ειδικότερα.

3.4.1 Δομητισμός και εκπαιδευτική πρακτική

Όσον αφορά τη μαθησιακή διαδικασία, ο δομητισμός γενικά υποστηρίζει ότι η νέα γνώση που αποκτά το άτομο στηρίζεται στις προηγούμενες γνώσεις και εμπειρίες του. Όταν αντιμετωπίζει μία νέα κατάσταση, αυτή είτε πρέπει να συμφιλιωθεί με τις προηγούμενες γνώσεις και εμπειρίες είτε να τις αλλάξει. Σε κάθε περίπτωση πάντως, η γνώση και η εικόνα για τον κόσμο δομείται ενεργητικά από το άτομο μέσω της εμπειρίας και του συλλογισμού πάνω σ' αυτή την εμπειρία [65].

Έχει ήδη αναφερθεί ο σημαντικός ρόλος της γλώσσας και της *ζώνης εγγύτερης ανάπτυξης* στο έργο του Vygotsky. Αυτές οι ιδέες έχουν ιδιαίτερη σημασία για τη διδασκαλία γενικά. Οι μαθητές χρειάζονται εκπαιδευτικά περιβάλλοντα πλούσια σε κοινωνικές αλληλεπιδράσεις, ώστε να εξερευνούν διάφορα θέματα με τους δασκάλους και τους συμμαθητές τους. Ευκαιρίες για συνεργασία με πιο έμπειρους καθοδηγητές είναι ιδιαίτερα κρίσιμες για την ανάπτυξη ανώτερων γνωστικών λειτουργιών.

Αντίστοιχα, κατά την αναφορά στον Piaget, επισημάνθηκε ο ρόλος των *σχημάτων* και των σταδίων ψυχοπνευματικής ανάπτυξης. Η βασική θέση της θεωρίας του είναι ότι η γνωστική ανάπτυξη του παιδιού είναι μία σταδιακή λειτουργία αλληλεπίδρασης με το φυσικό και κοινωνικό περίγυρο. Το παιδί καθώς αναπτύσσεται αντιμετωπίζει νέες και

άγνωστες καταστάσεις που δεν ταιριάζουν στη μέχρι εκείνη τη στιγμή θεώρησή του για τον κόσμο. Η γνώση χτίζεται βήμα-βήμα μέσω της ενεργού συμμετοχής.

Ο Bruner, αν και δεν μπορεί να χαρακτηριστεί ότι ανήκει στο κυρίως ρεύμα των δομητιστών, διατυπώνει απόψεις που συγκλίνουν σε ορισμένα σημεία με αυτούς. Ενδιαφέρεται για το πώς μπορεί να γίνει η εκπαίδευση πιο προσαρμοσμένη στις ανάγκες του παιδιού και θεωρεί ότι η μάθηση είναι μία ενεργός διαδικασία κατά την οποία οι μαθητές κάνουν ερωτήσεις και δίνουν τις δικές τους δοκιμαστικές απαντήσεις. Η ενεργός συμμετοχή του παιδιού επιτυγχάνεται μέσω της *ανακαλυπτικής μάθησης*, που επιτρέπει στα παιδιά να εξερευνήσουν εναλλακτικές λύσεις και να κατανοήσουν τις σχέσεις μεταξύ διαφόρων ιδεών [66]. Ο ρόλος του δασκάλου είναι να καθοδηγεί τα παιδιά με το διάλογο, έτσι ώστε να τα βοηθά να χτίζουν τη γνώση πάνω σε προϋπάρχουσες δομές.

Αρκετές άλλες έρευνες συνεισέφεραν σημαντικά στην ανάπτυξη της θεωρίας του δομητισμού, ειδικά στον τομέα των εκπαιδευτικών εφαρμογών της τεχνολογίας, όπως για παράδειγμα οι μικρόκοσμοι [66]. Το σημείο συμφωνίας των δομητιστών είναι ότι επιδιώκουν να προσφέρουν στα παιδιά ευκαιρίες για γνωστική ανάπτυξη μέσω της εξερεύνησης, της *μη δομημένης μάθησης* (*unstructured learning*), της ενεργού συμμετοχής και της επίλυσης προβλημάτων.

Γενικά, ως εκπαιδευτική πρακτική ο δομητισμός περιλαμβάνει την ενθάρρυνση του μαθητή να αναλάβει δράση, να ασχοληθεί με την επίλυση πραγματικών προβλημάτων, να πειραματιστεί και στη συνέχεια να συζητήσει πώς η αντίληψη που είχε για τα πράγματα άλλαξε. Ο δάσκαλος, από την άλλη πλευρά, φροντίζει έτσι ώστε κάθε δραστηριότητα να δομείται πάνω σε προηγούμενες γνώσεις του μαθητή και γενικά ο ρόλος του είναι να υποβοηθά την όλη διαδικασία, δεν είναι ο ίδιος φορέας και μεταδότης της γνώσης.

Όπως υπογραμμίζουν οι Brooks και Brooks [67], σε μία τάξη οργανωμένη με τα δομητιστικά πρότυπα:

1. Ενθαρρύνεται η αυτονομία και η πρωτοβουλία των μαθητών.
2. Οι ερωτήσεις προς τους μαθητές είναι ανοικτού τύπου και δίνεται αρκετός χρόνος για να απαντηθούν. Ενθαρρύνεται η κριτική σκέψη.
3. Οι μαθητές συνομιλούν μεταξύ τους ή με το δάσκαλο.
4. Οι δραστηριότητες στην τάξη προκρίνουν τη διατύπωση υποθέσεων και τη συζήτηση. Ως διδακτικό υλικό χρησιμοποιούνται πρωτογενείς πηγές πληροφοριών, ανεπεξέργαστα δεδομένα και αλληλεπιδραστικό υλικό.

Στα παραπάνω, οι Roblyer και Edwards [66] συμπληρώνουν:

5. Οι δραστηριότητες δίνουν έμφαση στην εξερεύνηση, στην επίλυση πρακτικών προβλημάτων σε τεχνολογικά πλούσια περιβάλλοντα.
6. Χρησιμοποιείται η συνεργασία σε ομάδες.
7. Η αξιολόγηση στηρίζεται στην παρουσίαση εργασιών και projects.

Ωστόσο, έχει ασκηθεί κριτική στο δομητισμό για ορισμένα θέματα, όπως:

Προηγούμενη γνώση. Σ' ένα δομητιστικό διδακτικό περιβάλλον ο δάσκαλος θα πρέπει να προσδιορίσει αρχικά τι γνωρίζει ο μαθητής και ποια είναι τα ενδιαφέροντά του και από εκείνο το σημείο να αρχίσει η σταδιακή δόμηση της γνώσης. Συχνά όμως οι δομητιστικές στρατηγικές στηρίζουν τη μάθηση στην επίλυση προβλημάτων που αφορούν σε περίπλοκες καταστάσεις, χωρίς να λαμβάνουν υπόψη τους τις δεξιότητες που απαιτούνται για την αντιμετώπισή τους. Αναμφίβολα κάποιοι μαθητές μπορούν να αντιμετωπίσουν αυτές τις καταστάσεις. Τι γίνεται όμως γι' αυτούς που δεν μπορούν; Πόση και τι είδους προϋπάρχουσα γνώση απαιτείται ώστε να αντιμετωπίζουν όλοι οι μαθητές αυτά τα θέματα [68]; Σ' αυτή την περίπτωση, ο δάσκαλος ή η τεχνολογία πρέπει να είναι σε θέση να προσφέρουν βοήθεια στους μαθητές που την έχουν ανάγκη, ακόμα και αν αυτό πρέπει να γίνει σε μεγάλη κλίμακα.

Γνωστικά πεδία. Αρκετοί είναι αυτοί που υποστηρίζουν ότι ο δομητισμός αντιμετωπίζει καλύτερα ορισμένα γνωστικά θέματα απ' ό,τι άλλα [26, 27]. Γενικά οι δομητιστικές στρατηγικές επιδιώκουν να διδάξουν επίλυση προβλημάτων σε όχι καλά δομημένα (*ill-structured*) γνωστικά πεδία. Επίσης, αντί να καλύπτεται σε βάθος ένα θέμα, προτιμάται να καλύπτεται ένα ευρύ πεδίο θεμάτων ταυτόχρονα. Όμως υπάρχουν περιπτώσεις που είναι πιο επιθυμητή η απόκτηση γνώσης σ' ένα και μόνο γνωστικό πεδίο.

Αξιολόγηση. Αρκετή κριτική έχει ασκηθεί στο δομητισμό για την απόρριψη της αξιολόγησης μέσω αντικειμενικών μεθόδων και κριτηρίων. Αν και αξιολόγηση γίνεται μέσω άλλων μεθόδων (ομαδικές εργασίες, projects, κτλ.), αυτές δεν προσφέρουν στοιχεία για την ατομική επίδοση. Δε δείχνουν δηλαδή κατά πόσο ένας μαθητής έμαθε κάτι ή απέκτησε κάποιες δεξιότητες.

3.4.2 Δομητισμός και τεχνολογικές εφαρμογές στην εκπαίδευση

Η τεχνολογία σ' ένα δομητιστικό εκπαιδευτικό περιβάλλον έχει ένα σημαντικό ρόλο στις καθημερινές δραστηριότητες, αλλά δε γίνεται η ίδια αντικείμενο της διδασκαλίας. Ένα τεχνολογικό περιβάλλον λειτουργεί σαν ένα εργαστήριο μέσα στο οποίο οι μαθητές μπορούν να παρατηρήσουν, να εξασκηθούν, να θέσουν ερωτήσεις και να αξιολογήσουν τη γνώση [69].

Τα παιδιά τού σήμερα ζουν σ' ένα περιβάλλον πολύ διαφορετικό απ' αυτό που ζούσαν τα παιδιά τού χθες. Οι συνθήκες ήταν τέτοιες, που ένα ξύλινο σπαθί, ένα κομμάτι χαρτόνι, μία σκούπα και πολλή φαντασία αρκούσαν για να μετατραπούν τα παιδιά σε ιππότες, να πολεμήσουν σε μαγικούς κόσμους. Η επικοινωνία γινόταν είτε πρόσωπο με πρόσωπο είτε με αλληλογραφία. Τα παιδιά τού σήμερα από τη γέννησή τους βρίσκονται μέσα σ' έναν κόσμο τεχνολογίας. Παίζουν ηλεκτρονικά παιχνίδια, χειρίζονται με μεγάλη άνεση την τηλεόραση, το βίντεο και κάθε είδος ηλεκτρονικής συσκευής. Για την επικοινωνία τους χρησιμοποιούν κινητά τηλέφωνα.

Υπάρχουν δύο σημαντικά πλεονεκτήματα στην εφαρμογή δομητιστικών μεθόδων μέσω της τεχνολογίας. Πρώτον, εκπαιδευτικά περιβάλλοντα που στηρίζονται στην τεχνολογία μπορούν και παρέχουν γνωστικές καταστάσεις που απαιτούν επίλυση προβλημάτων και λήψη αποφάσεων [70]. Δεύτερον, επειδή για το σκοπό αυτό χρησιμοποιούνται προγράμματα προσομοίωσης, CD-ROMs, πολυμέσα και υπερμέσα, είναι απόλυτα συμβατά με την ηλεκτρονική καθημερινότητα των παιδιών, μέσα και έξω από την τάξη.

Η τεχνολογία παρέχει εργαλεία που προηγουμένως δεν ήταν διαθέσιμα. Τα παιδιά έχουν τη δυνατότητα να αντλούν πληροφορίες με διάφορες μορφές (κείμενο, εικόνες, ήχος) και να εξετάζουν διαφορετικές απόψεις καθώς δομούν τη γνώση σε κάποιο θέμα. Η τεχνολογία δίνει τη δυνατότητα στα παιδιά να αφιερώνουν περισσότερη ώρα στην παραπάνω εργασία, αλλά και να συνεργάζονται με τους συμμαθητές και τους δασκάλους τους [69]. Στα παλαιότερα χρόνια, ο χώρος απ' όπου μπορούσαν να αντληθούν πληροφορίες ήταν περιορισμένος. Η κοινωνική αλληλεπίδραση και η συνεργασία, που ενθαρρύνονται από το δομητιστικό μοντέλο, ενισχύονται από τεχνολογίες όπως το Διαδίκτυο, που επιτρέπει κυριολεκτικά την επικοινωνία με κάθε περιοχή του κόσμου [70].

Τα πολυμέσα ενθαρρύνουν τη δημιουργικότητα των μαθητών. Εφαρμογές πολυμέσων που παρέχουν καταστάσεις επίλυσης προβλημάτων βοηθούν τα παιδιά να αναγνωρίσουν και να βελτιώσουν τις στρατηγικές επίλυσης προβλημάτων και να μεταφέρουν τη γνώση σε υψηλές πνευματικές δεξιότητες (*higher-order skills*) [66].

Όμως πρέπει να σημειωθεί ότι η επιτυχία ή η αποτυχία των τεχνολογικά πλούσιων δομητιστικών περιβαλλόντων εξαρτάται σε τελική ανάλυση από τους δασκάλους. Αυτοί έχουν να αντιμετωπίσουν σε καθημερινή βάση το έργο της διδασκαλίας των μαθημάτων. Η εφαρμογή της τεχνολογίας και του δομητισμού στην τάξη απαιτεί καλό σχεδιασμό και άνεση χρόνου. Δυστυχώς, ούτε τα μέσα είναι πολλές φορές διαθέσιμα ούτε ο χρόνος. Όταν μάλιστα υπάρχει πίεση χρόνου, τελικά θα επιστρέψουν στους παλιούς

τρόπους διδασκαλίας, σε αυτούς που και οι ίδιοι διδάχθηκαν. Για να υπάρξει αλλαγή στον τρόπο που διδάσκονται οι μαθητές, πρέπει να υπάρξει αλλαγή στην εκπαίδευση των δασκάλων. Πρέπει και αυτοί να βιώσουν, ως μαθητές, τα νέα μαθησιακά περιβάλλοντα [71].

3.4.3 Δομητισμός και Εικονική Πραγματικότητα

Η Ε.Π. συγκεντρώνει τα παραπάνω πλεονεκτήματα των πολυμέσων και υπερμέσων, αλλά επειδή χρησιμοποιεί μία διαφορετική τεχνολογία, προσθέτει επιπλέον χαρακτηριστικά. Τα εικονικά περιβάλλοντα παρέχουν τη δυνατότητα στο χρήστη να [72]:

- Μεταχειρίζεται και να αλληλεπιδρά με αντικείμενα όπως και στο φυσικό κόσμο, αλλά και να μπορεί να αλλάζει τα σχετικά τους μεγέθη, να εφαρμόζει ή να ανατρέπει τους φυσικούς νόμους.
- Ελέγχει το χρόνο. Μπορεί να μελετήσει μέσα σε λίγα λεπτά την εξέλιξη ενός φαινομένου το οποίο στην πραγματικότητα απαιτεί εκατοντάδες, χιλιάδες ή ακόμα και εκατομμύρια χρόνια για να ολοκληρωθεί, αλλά και το αντίστροφο, να μελετήσει δηλαδή ένα φαινόμενο που στην πραγματικότητα συμβαίνει σε απειροελάχιστο χρόνο.
- Δέχεται πληροφορίες που κάτω από άλλες συνθήκες δε θα ήταν διαθέσιμες στις ανθρώπινες αισθήσεις.
- Απεικονίζει και να μεταχειρίζεται αντικείμενα και γεγονότα που δεν έχουν φυσική μορφή, όπως οι μαθηματικές εξισώσεις.
- Αλληλεπιδρά με άλλους χρήστες παρόντες στον εικονικό κόσμο.

Η εμπύθιση του χρήστη στο εικονικό περιβάλλον, η αλληλεπίδραση, η απουσία περιβάλλοντος διεπαφής και κυρίως η μη-συμβολική επικοινωνία και οι εμπειρίες “πρώτου προσώπου” είναι τα κλειδιά για τη συμβατότητα της Ε.Π. με τη θεωρία του δομητισμού.

Συνεπώς, με την Ε.Π. είναι δυνατό να διδαχθούν κανόνες και αφηρημένες έννοιες χωρίς τη χρήση της γλώσσας και άλλων συμβόλων. Η εμπειρία (πραγματική ή εικονική) με την οποία συνδέεται μία ιδέα ή έννοια είναι σημαντική τόσο για την κατανόηση της έννοιας όσο και για τη μετέπειτα χρήση της [32]. Με άλλα λόγια, η εμπειρία είναι το “όχημα” για τη δόμηση και τη χρήση της γνώσης. Όπως σημειώθηκε, επειδή η Ε.Π. χρησιμοποιεί μη-συμβολική αλληλεπίδραση, μπορεί να δώσει τη δυνατότητα στα παιδιά να κατανοήσουν διαισθητικά, χωρίς τη χρήση των συμβολικών αναπαραστάσεων του αντίστοιχου γνωστικού πεδίου, ακόμα και αφηρημένες έννοιες, δίνοντας έτσι εμπειρίες σε πρωτογενές επίπεδο.

Η Ε.Π. δίνει εμπειρίες μέσω της “πραγματικής” χρήσης αντικειμένων. Έτσι απαιτεί αλληλεπίδραση και κατά συνέπεια ενθαρρύνεται η ενεργός συμμετοχή των παιδιών. Δίνεται δηλαδή στους μαθητές η δυνατότητα ελέγχου στη μαθησιακή διαδικασία [73]. Όπως έχει τονιστεί κατ’ επανάληψη, η ενεργός συμμετοχή παίζει κρίσιμο ρόλο.

Αντικατοπτρίζοντας τον πραγματικό κόσμο δίνεται η ευκαιρία στα παιδιά να μάθουν από τα ίδια τους τα λάθη χωρίς να υπάρχουν συνέπειες και χωρίς να κινδυνεύουν. Μάλιστα αυτό μπορεί να γίνει με τη μορφή παιχνιδιού, ο ρόλος του οποίου τονίζεται και από τον Vygotsky και από τον Piaget και από τον Bruner [74].

Η Ε.Π. παρέχει τη δυνατότητα προσαρμογής του διδακτικού υλικού στις ατομικές ανάγκες και το γνωστικό ύψος του κάθε μαθητή. Μπορεί έτσι το κάθε παιδί να βιώνει μία εμπειρία με το δικό του βηματισμό, αργό ή γρήγορο, χωρίς να περιορίζεται από τα χρονικά πλαίσια του συμβατικού ωρολογίου προγράμματος [75]. Λύνεται με τον τρόπο αυτό το πρόβλημα του πόση προηγούμενη γνώση χρειάζεται ο μαθητής ώστε να αντιμετωπιστεί στις απαιτήσεις ενός δομητιστικού διδακτικού περιβάλλοντος.

Όσον αφορά τη δυνατότητα ομαδικής εργασίας, υπάρχει η γενική αίσθηση ότι όταν τα άτομα εργάζονται ομαδικά πετυχαίνουν καλύτερα γνωστικά αποτελέσματα, επειδή υπάρχουν περισσότεροι γνωστικοί πόροι [76]. Ωστόσο, στις περισσότερες εφαρμογές το πρόβλημα είναι η ανάπτυξη δεξιοτήτων που ευνοούν τη συνεργασία, όπως η αίσθηση της ατομικής ευθύνης μέσα στην ομάδα, το ομαδικό πνεύμα, η ανάληψη πρωτοβουλιών και ευθυνών. Επιπλέον, δεν είναι εξασφαλισμένη η ίση προσπάθεια και επίδοση στα μέλη της ομάδας. Συνεπώς χρειάζεται αρκετή προσοχή στο σχεδιασμό και την υλοποίηση πρακτικών συνεργατικής μάθησης, ειδικά σε περιβάλλοντα Ε.Π.

Είναι όμως γεγονός ότι ένα εκπαιδευτικό περιβάλλον Ε.Π. υποστηρίζει την ομαδική εργασία με πληρέστερο τρόπο απ’ ό,τι τα υπερμέσα και τα πολυμέσα. Μέσω δικτυακής σύνδεσης μπορούν να συνυπάρχουν πολλοί χρήστες στον ίδιο εικονικό κόσμο, να έχουν τα ίδια οπτικοακουστικά ερεθίσματα, να μοιράζονται τον έλεγχο της ροής των πραγμάτων και παράλληλα να επικοινωνούν και να συζητούν μεταξύ τους. Τα εικονικά διαμοιραζόμενα περιβάλλοντα, η εικονική τάξη και η χρήση τρισδιάστατων μορφών που αντικαθιστούν τη φυσική παρουσία σ’ ένα δικτυακό περιβάλλον (*avatars*) είναι μερικοί τρόποι με τους οποίους η Ε.Π. υλοποιεί τις αντιλήψεις της συνεργατικής μάθησης.

Τα κίνητρα για μάθηση παίζουν σημαντικό ρόλο στην εκπαιδευτική διαδικασία και πρέπει να λαμβάνονται υπόψη στο σχεδιασμό ενός εκπαιδευτικού περιβάλλοντος. Στα κίνητρα για μάθηση το ενδιαφέρον εστιάζεται πλέον όχι στο αν το παιδί μπορεί να μά-

θει, αλλά στο τι κάνει το παιδί να θέλει να μάθει [77]. Το περιβάλλον πρέπει να σχεδιάζεται έτσι ώστε να παρέχει στοιχεία που κεντρίζουν το ενδιαφέρον, και μάλιστα με μεγάλη χρονική διάρκεια, πράγμα που δεν είναι πάντοτε εξασφαλισμένο.

Τα κίνητρα για μάθηση εξαρτώνται σε μεγάλο βαθμό από το ενδιαφέρον και οι περισσότεροι άνθρωποι βρίσκουν την Ε.Π. μία εξαιρετικά ενδιαφέρουσα εμπειρία (ενδεικτικά [56, 78]). Δεν είναι όμως μόνο το ενδιαφέρον που παρέχει κίνητρα για μάθηση στην Ε.Π. Είναι ο ευρύτερος συνδυασμός αλληλεπίδρασης, ρεαλισμού, το κέντρισμα της φαντασίας, η πρόκληση, το παιχνίδι [79]. Όπως μάλιστα σημειώνει ο Psołka [79], δεν είναι μόνο το καινούργιο και το άγνωστο που συγκινεί τα παιδιά, η Ε.Π. είναι μία τεχνολογία που ανοίγει πολλά νέα μονοπάτια και ενδυναμώνει τη μάθηση.

Τέλος, όσον αφορά το θέμα της αξιολόγησης, η Ε.Π. έχει τις δυνατότητες ενός πολύ ισχυρού εργαλείου αξιολόγησης και ελέγχου της επίδοσης του κάθε μαθητή, γιατί επιτρέπει την καταγραφή κάθε συνεδρίας σ' ένα εικονικό περιβάλλον [80].

3.5 Συνοπτική παρουσίαση των μέχρι σήμερα ερευνών στον τομέα των παιδαγωγικών εφαρμογών της Εικονικής Πραγματικότητας

Ένας αρκετά σημαντικός αριθμός ερευνών προσπάθησαν και προσπαθούν να δώσουν απαντήσεις σε ερωτήματα που προσδιορίζουν το πλαίσιο μέσα στο οποίο οι διάφορες εφαρμογές Ε.Π. έχουν ιδιαίτερη παιδαγωγική αξία. Πρέπει όμως να επισημανθεί ένα σημαντικό στοιχείο. Η Ε.Π. είναι μία τεχνολογία που ακόμα ωριμάζει. Η ωρίμανση αυτή προκαλεί συνεχείς αλλαγές στα διαθέσιμα μέσα, στην ποιότητα, στις ιδιότητές τους και στον τρόπο με τον οποίο αυτά είναι δυνατό να χρησιμοποιηθούν στην εκπαιδευτική διαδικασία. Αυτές οι συχνές αλλαγές καθιστούν ανεπίκαιρο μεγάλο μέρος από τις έρευνες και δημιουργούν την ανάγκη αναθεώρησης των δεδομένων και των συμπερασμάτων τους.

3.5.1 Γενικά σχόλια για τις εφαρμογές και τις έρευνες της Εικονικής Πραγματικότητας στην εκπαίδευση

Η Youngblut [81], σε εκτενέστατη επισκόπηση των ερευνών και των εφαρμογών Ε.Π. για εκπαιδευτική χρήση, επισημαίνει ότι:

1. Το σύνολο σχεδόν των μέχρι στιγμής ερευνών ασχολείται με τη χρήση της Ε.Π. ως εκπαιδευτικού εργαλείου σε μία περιορισμένη ομάδα μαθητών κάθε φορά και για περιορισμένο χρονικό διάστημα. Δεν υπάρχουν πληροφορίες για τις επιπτώσεις που έχει μία μακρόχρονη εκπαιδευτική διαδικασία που στηρίζεται στην Ε.Π.
2. Οι περισσότερες εφαρμογές αφορούν προκατασκευασμένους εικονικούς κόσμους. Η συντριπτική τους πλειοψηφία προορίζεται για ένα χρήστη, ενώ ελάχιστες υποστηρίζουν πολλαπλούς χρήστες, παρέχοντας όμως περιορισμένο βαθμό αλληλε-

πίδρασης. Τα τρία τέταρτα των εφαρμογών αφορούν περιβάλλοντα πλήρους εμπύθισης. Σχεδόν οι μισές εφαρμογές αναπτύχθηκαν για πρωτογενή έρευνα και οι υπόλοιπες για πρακτική χρήση.

3. Οι εφαρμογές της δεύτερης κατηγορίας είναι ισόποσα μοιρασμένες για χρήση από μαθητές του δημοτικού σχολείου, μαθητές γυμνασίου και λυκείου, και φοιτητές. Αρκετές από τις εφαρμογές σχεδιάστηκαν για να αντιμετωπίσουν ανάγκες παιδιών με κινητικά ή νοητικά προβλήματα. Η θεματολογία είναι πλούσια και αντλείται από τις φυσικές και ανθρωπιστικές επιστήμες και τις τέχνες. Λίγες εφαρμογές είναι σχεδιασμένες με βάση συγκεκριμένες απαιτήσεις της εκπαιδευτικής πολιτικής χωρών ή πολιτειακών κυβερνήσεων (Η.Π.Α.).
4. Από τις εφαρμογές που μελετούν την ανάπτυξη εικονικών περιβαλλόντων από τους μαθητές, οι περισσότερες είναι αποτέλεσμα εργασιών μέσα στην τάξη ως το πρακτικό μέρος της διδασκαλίας και λίγες μόνο μπορούν να θεωρηθούν καθαρά ερευνητικές προσπάθειες. Για τις περισσότερες δεν υπάρχει η προοπτική να επαναλαμβάνονται σε τακτική βάση. Οι καθαρά ερευνητικές προσπάθειες εστίασαν στη δημιουργία, από ομάδες μαθητών με συνεργατική δράση, κόσμων σε περιβάλλον πλήρους εμπύθισης. Υπήρξε ενεργός υποστήριξη από τους ερευνητές στην προσπάθεια αυτή, που αφορούσε κυρίως μεγαλύτερους σε ηλικία μαθητές. Η θεματολογία, παρά το μικρότερο αριθμό των εφαρμογών, υπήρξε πλούσια.
5. Η επιτραπέζια Ε.Π. για πρακτικούς λόγους είναι πιο κατάλληλη και πιο ώριμη για μαζική χρήση απ' ό,τι η εμπυθισμένη Ε.Π. Είναι οικονομικά προσιτή, γιατί εφαρμογές της μπορούν να εκτελεστούν σε απλούς ηλεκτρονικούς υπολογιστές με κανένα ή με μικρό επιπλέον κόστος. Στην πλήρως εμπυθισμένη Ε.Π., εκτός από το θέμα του κόστους, υπάρχουν προβλήματα στον εξοπλισμό που απαιτείται, λόγω τεχνικών αδυναμιών και λόγω της χαμηλής ως μέτριας ποιότητας των γραφικών.

3.5.2 Συνοπτική αποτύπωση των βασικών προβλημάτων που απασχόλησαν τις ερευνητικές προσπάθειες και των συμπερασμάτων τους

Αντλώντας και πάλι στοιχεία από την επισκόπηση της Youngblut, μπορούν να επισημανθούν τα παρακάτω θέματα που απασχόλησαν τις έρευνες για την Ε.Π.:

Η μάθηση σε εικονικό περιβάλλον παρέχει κάτι πολύτιμο που δεν μπορεί να αποκτηθεί με άλλον τρόπο: Οι μελέτες κατέληξαν στο συμπέρασμα ότι κάποιας μορφής μάθηση λαμβάνει χώρα όταν οι μαθητές χρησιμοποιούν την Ε.Π. για την επίτευξη συγκεκριμένων διδακτικών στόχων. Οι ελάχιστες μελέτες που ασχολήθηκαν με την αποτελεσματικότητα της Ε.Π. ως εργαλείου μάθησης σε σύγκριση με τις παραδοσιακές μεθόδους, κατέληξαν στο συμπέρασμα ότι οι μαθητές είχαν εξίσου καλή ή καλύτερη επίδοση. Η διδακτική μέθοδος που ακολουθείται στη μεγάλη πλειοψηφία όλων των εφαρμογών στηρίζεται στο δομητισμό. Δεν είναι εύκολο να προσδιοριστεί αν η αποτε-

λεσματικότητα των εικονικών περιβαλλόντων οφείλεται στην Ε.Π. ή στη διδακτική μέθοδο ή στο συνδυασμό τους.

Προκατασκευασμένα εικονικά περιβάλλοντα και συμβατικές διδακτικές πρακτικές: Οι ερευνητές δεν προσδιόρισαν τα ιδιαίτερα εκείνα χαρακτηριστικά που κάνουν ένα περιβάλλον Ε.Π. αποτελεσματικό σε μικρό ή μεγάλο βαθμό, σε σχέση με την παραδοσιακή διδασκαλία. Επιπλέον, οι έρευνες έγιναν σε διαφορετικά εκπαιδευτικά αντικείμενα, σε διαφορετικές ομάδες μαθητών και με διαφορετικές μεθόδους η καθεμία. Κατά συνέπεια, υπάρχει έντονη ανάγκη για έρευνα στο πεδίο αυτό, ώστε να προσδιοριστούν αυτά τα χαρακτηριστικά.

Εικονικά περιβάλλοντα κατασκευασμένα από τους μαθητές και παραδοσιακή διδασκαλία: Τα αποτελέσματα των ερευνών δίνουν μεικτή εικόνα και δεν είναι συγκρίσιμα μεταξύ τους, λόγω των διαφορετικών προσεγγίσεων που επιχειρούν.

Σύγκριση της αποτελεσματικότητας της Ε.Π. πλήρους εμπύθισης και της επιτραπέζιας Ε.Π.: Οι ερευνητικές ομάδες χρησιμοποίησαν τις ίδιες ή παραπλήσιες εφαρμογές σε πλήρη και σε μη-πλήρη εμπύθιση. Υπάρχουν αμφίβολα αποτελέσματα. Το μόνο βέβαιο είναι ότι οι μαθητές έδειξαν περισσότερο ενδιαφέρον για μάθηση στο περιβάλλον πλήρους εμπύθισης.

Ε.Π., συνεργατική μάθηση και η αποτελεσματικότητα αυτής: Ορισμένοι ερευνητές υποστηρίζουν ότι συνεργατική μάθηση επιτυγχάνεται όταν δύο ή περισσότεροι μαθητές εργάζονται μαζί σε προκατασκευασμένα εικονικά περιβάλλοντα που προορίζονται για ένα χρήστη. Δεν υπάρχουν όμως αναφορές για το πόσο αποτελεσματικό είναι στην πράξη κάτι τέτοιο. Από την άλλη πλευρά, η κατασκευή εικονικών κόσμων από τους μαθητές είναι κατεξοχήν συνεργατική δραστηριότητα, με ανταλλαγή απόψεων και διάλογο μεταξύ τους. Πάλι δεν υπάρχουν επαρκή δεδομένα για την επίδραση που είχε η συνεργασία των μαθητών στην εκπαιδευτική διαδικασία.

Τέλος, στα δικτυακά διαμοιραζόμενα περιβάλλοντα, ειδικά σχεδιασμένα να ενισχύσουν τη συνεργασία μεταξύ των μαθητών, η εικόνα ήταν μεικτή. Υπήρξε όμως η τάση η διαδικασία να εξελίσσεται από ισότιμα συνεργατική, στη δημιουργία ομάδας και αρχηγού ομάδας. Λίγα είναι γνωστά τόσο για τη συνεργατική μάθηση όσο και για τις τεχνολογικές εφαρμογές που θα την υποστηρίζουν. Κατά συνέπεια, οι διάφορες μορφές της συνεργατικής μάθησης πρέπει να ερευνηθούν, και φυσικά συνεργατικές εφαρμογές της Ε.Π. μπορούν να βοηθήσουν στην προσπάθεια αυτή.

Εκπαιδευτικοί σκοποί όπου πρέπει ή δεν πρέπει να χρησιμοποιείται η Ε.Π.: Είναι εύκολο να ειπωθεί ότι η Ε.Π. είναι κατάλληλη για εκείνες τις καταστάσεις όπου ο μαθητής μπορεί να καθοδηγηθεί στη δόμηση της γνώσης και σε γνωστικά αντικείμενα που είναι από τη φύση τους πλούσια σε οπτικοακουστικό υλικό. Τέτοιες δηλώσεις όμως δεν προσφέρουν απάντηση στο ποια συγκεκριμένα χαρακτηριστικά της Ε.Π. είναι κατάλληλα σε συγκεκριμένα γνωστικά αντικείμενα.

Χαρακτηριστικά των μαθητών που επιτρέπουν τον προσδιορισμό του πότε είναι κατάλληλη εκπαίδευση που στηρίζεται στην Ε.Π.: Τα αποτελέσματα δύο ερευνών δείχνουν σχέση φύλου και επίδοσης σε περιβάλλον Ε.Π. Δεν υπάρχουν επαρκή στοιχεία που να τεκμηριώνουν την άποψη ότι η Ε.Π. είναι καταλληλότερη σε "οπτικούς" χαρακτήρες. Επίσης φαίνεται ότι μεγαλύτερη επίδραση υπάρχει σε μαθητές που έχουν μικρή επαφή με την τεχνολογία. Σε κάθε περίπτωση είναι απαραίτητη η συλλογή περισσότερων ερευνητικών στοιχείων.

Η ευχρηστία του εικονικού περιβάλλοντος: Η πλοήγηση σ' ένα εικονικό περιβάλλον είναι θεμελιώδες ζήτημα και ταυτόχρονα η μεγαλύτερη δυσκολία που συναντούν οι μαθητές. Για ένα περιβάλλον μη-εμβύθισης λίγα μπορούν να γίνουν, ενώ αντίθετα υπάρχουν μεγάλα περιθώρια βελτίωσης για τις περιφερειακές συσκευές που χρησιμοποιούνται σε περιβάλλον πλήρους εμβύθισης. Η ευχρηστία των περιφερειακών συσκευών, οι δυνατότητές τους και η ποικιλία τους είναι θέματα με τα οποία ασχολούνται τεχνολογικά κέντρα και εταιρείες κατασκευής ηλεκτρονικών συσκευών. Η έρευνα στη χρήση τέτοιων συσκευών για εκπαιδευτικούς λόγους έχει να δώσει λίγα σχετικά στοιχεία, κυρίως για τις συσκευές προβολής της εικόνας. Γενικά μπορεί να ειπωθεί ότι η χρήση κίνησης και φωνής σ' έναν τεχνητό κόσμο προτιμάται από τη χρήση κειμένου. Ένα θέμα που επίσης χρειάζεται μελέτη είναι τα φαινόμενα αποπροσανατολισμού και ναυτίας από την παρατεταμένη χρήση συσκευών πλήρους εμβύθισης.

Ε.Π. και ο ρόλος του δασκάλου: Τα ερευνητικά δεδομένα παρουσιάζουν αλλαγή στο ρόλο του δασκάλου. Μετατρέπεται από το άτομο με όλες τις απαντήσεις, σε υποστηρικτή της προσπάθειας των μαθητών στην εξερεύνηση των εικονικών κόσμων και στην οικοδόμηση των γνώσεων που αποκτούν απ' αυτή την εξερεύνηση. Όμως, για να διερευνηθεί ο τρόπος με τον οποίο οι δάσκαλοι πρέπει να προετοιμαστούν για το νέο αυτό ρόλο και για να διευκρινιστεί ποια είναι τα εφόδια που τους είναι απαραίτητα, χρειάζεται περισσότερη έρευνα. Επίσης, οι δάσκαλοι θεωρούν ότι μπορεί να υπάρξει πρόβλημα στη διαχείριση του μαθήματος και στον τρόπο που μπορούν να αξιολογήσουν την πορεία των μαθητών σ' ένα περιβάλλον Ε.Π.

Βαθμός αποδοχής από δασκάλους και μαθητές της Ε.Π.: Δεδομένα από χιλιάδες μαθητές που συμμετείχαν σε ερευνητικά προγράμματα κάνοντας χρήση όλων των μορφών της Ε.Π. δείχνουν υψηλό βαθμό αποδοχής. Οι μαθητές βρήκαν ιδιαίτερα διασκεδαστική την εργασία σε περιβάλλοντα Ε.Π. και τα δεδομένα συνηγορούν στο ότι υπάρχει μεγάλη αλλαγή στα κίνητρα για μάθηση. Επίσης, τα παιδιά δείχνουν ανοχή στις ατέλειες και τα προβλήματα των συσκευών απεικόνισης (*H.M.D.*). Δεν υπάρχουν όμως δεδομένα για τη διάρκεια της αποδοχής και αν εξακολουθεί να υπάρχει στην περίπτωση που η Ε.Π. γίνει καθημερινή πρακτική. Από την πλευρά τους οι δάσκαλοι δηλώνουν στη μεγάλη τους πλειοψηφία ότι θα χρησιμοποιούσαν και πάλι τη συγκεκριμένη τεχνολογία, αν ήταν οικονομικά προσιτή, διαθέσιμη και πιο εύκολη στη χρήση από τους ίδιους και από τους μαθητές.

Το κόστος της προμήθειας του απαραίτητου εξοπλισμού: Το κόστος ενός συστήματος επιτραπέζιας Ε.Π. είναι το ίδιο με το κόστος ενός ηλεκτρονικού υπολογιστή. Το κόστος αυξάνεται δραματικά για συστήματα πλήρους εμβύθισης. Όσο όμως η τεχνολογία ωριμάζει, τόσο το κόστος αργά και σταθερά πέφτει, όπως ακριβώς με κάθε ηλεκτρονική συσκευή.

Το λογισμικό ανάπτυξης εφαρμογών Ε.Π.: Υπάρχει πληθώρα προγραμμάτων χαμηλού κόστους αλλά χαμηλών δυνατοτήτων. Τα κορυφαία προγράμματα ανάπτυξης εφαρμογών Ε.Π. είναι ακριβά και απαιτούν χρόνο για την εκμάθησή τους. Γενικά υπάρχει έλλειψη εφαρμογών στη μεσαία κατηγορία.

Σχέση κόστους-οφέλους: Οι ερευνητικές ομάδες σπάνια δίνουν το κόστος των εφαρμογών που ανέπτυξαν, με αποτέλεσμα τα όποια παιδαγωγικά οφέλη να μην είναι δυνατό να συγκριθούν με το κόστος ανάπτυξης των εφαρμογών.

3.6 Προβληματισμός

Αυτό που τελικά πρέπει να απαντηθεί ξεκάθαρα είναι αν πέρα από τους οικονομικούς και πρακτικούς λόγους υπάρχουν και άλλοι που να συνηγορούν στη χρήση της Ε.Π. ως εκπαιδευτικού εργαλείου. Στο θέμα αυτό δίνει απάντηση η Pantelidis, θέτοντας μία σειρά από συγκεκριμένους όρους όπου μπορεί να χρησιμοποιείται ή να μη χρησιμοποιείται η Ε.Π. για εκπαιδευτική χρήση [82].

Μπορεί να χρησιμοποιηθεί η Ε.Π. όταν:

1. Η διδασκαλία και η εξάσκηση στο πραγματικό περιβάλλον είναι επικίνδυνη ή καταστροφική, όταν για παράδειγμα το παιδί ή ο εκπαιδευτής μπορεί να τραυματιστούν.

2. Όταν η διδασκαλία είναι αδύνατη με άλλον τρόπο ή το περιβάλλον δεν υπάρχει στην πραγματικότητα.
3. Όταν υφίστανται θέματα αυξημένου κόστους στις άλλες μορφές διδασκαλίας /εξάσκησης.
4. Η αλληλεπίδραση με το εικονικό περιβάλλον παρέχει κίνητρα μάθησης ίδια ή περισσότερα από τον πραγματικό και κάνει τη διδασκαλία διασκεδαστική.
5. Η ομαδική εργασία είναι απαραίτητη.
6. Η εξάσκηση χρειάζεται να προσομοιάζει όσο το δυνατόν περισσότερο με τις πραγματικές καταστάσεις και συνθήκες.

Δεν μπορεί να χρησιμοποιηθεί η Ε.Π. όταν:

1. Υπάρχουν εναλλακτικές λύσεις διδασκαλίας/εξάσκησης σε πραγματικό περιβάλλον.
2. Το κόστος υλοποίησης μίας εφαρμογής Ε.Π. είναι δυσανάλογα μεγάλο.
3. Η αλληλεπίδραση με πραγματικούς ανθρώπους είναι απαραίτητη.
4. Η χρήση εικονικού περιβάλλοντος μπορεί να οδηγήσει σε σύγχυση με τον πραγματικό κόσμο.

Στα παραπάνω σημεία πρέπει να προστεθούν ορισμένες ακόμα παρατηρήσεις κυρίως από την τεχνολογική σκοπιά της Ε.Π.

Ένα αδύνατο σημείο των εφαρμογών που έχουν ήδη αναπτυχθεί είναι ότι υστερούν στον τομέα του ρεαλισμού, λόγω των τεχνολογικών περιορισμών τη στιγμή που αναπτύχθηκαν. Είναι όμως δύσκολο να ταυτιστεί κάποιος με το τρισδιάστατο μοντέλο ενός αντικειμένου όταν δεν είναι ακριβώς ίδιο με το πραγματικό. Είναι δύσκολο να εμβυθιστεί κάποιος σε μία εικονική πόλη όταν τα σπίτια της είναι αφαιρετικές αναπαραστάσεις των πραγματικών. Κατά κάποιον τρόπο, ενώ επιδιώκεται η αποφυγή της χρήσης συμβόλων, αναγκάζεται ο χρήστης να δημιουργήσει ένα νέο σύνολο συμβόλων που να ανταποκρίνεται στα αντικείμενα που βλέπει. Αναγκάζεται να σκέφτεται ότι αυτό που βλέπει είναι κάτι που μοιάζει με σπίτι, παρά να θεωρεί ως δεδομένο ότι αυτό που βλέπει είναι σπίτι. Η ρεαλιστική απόδοση ενός κόσμου συνεισφέρει σημαντικά στην εμβύθιση σ' αυτόν, ίσως σε ίδιο βαθμό με τη χρήση ειδικών συσκευών. Χρειάζεται λοιπόν να αξιοποιηθούν πλήρως οι δυνατότητες που παρέχουν οι τεχνολογικές εξελίξεις.

Όμως στο χώρο της τεχνολογίας οι εξελίξεις χρειάζονται πολύ λίγο χρόνο για να εκδηλωθούν. Χαρακτηριστικός για την κατάσταση είναι ο “νόμος του Moore” [83], που αναφέρει ότι για δεδομένο ποσό χρημάτων η υπολογιστική ισχύς διπλασιάζεται κάθε δεκαοκτώ μήνες, ως αποτέλεσμα του διπλασιασμού του αριθμού των τρανζιστορ σ' ένα ολοκληρωμένο κύκλωμα.

Με αυτά τα δεδομένα, ο τρόπος υλοποίησης μιας εφαρμογής θα πρέπει να είναι τεχνολογικά επίκαιρος, ώστε: α) να ενσωματώνει όλες τις εξελίξεις που θα επιτρέπουν αυξημένο ρεαλισμό και β) να επιτευχθεί ένας ελάχιστος χρόνος ωφέλιμης ζωής της εφαρμογής μέχρι που να ξεπεραστεί από τις τεχνολογικές εξελίξεις.

Τέλος, η ανάπτυξη μίας εφαρμογής Ε.Π. απαιτεί πολλαπλάσιο χρόνο από την ανάπτυξη μία εφαρμογής πολυμέσων. Πέρα απ' αυτό, τα εργαλεία που χρησιμοποιούνται κάθε άλλο παρά είναι προσιτά στο μέσο χρήστη. Το κόστος τους είναι σημαντικό και η εκμάθησή τους απαιτεί πολύ χρόνο. Η διάδοση ενός μέσου εξαρτάται από πολλούς παράγοντες, μεταξύ των οποίων είναι και η ευκολία χρήσης του. Συνεπώς χρειάζεται να εξευρεθούν ευέλικτα και εύκολα στη χρήση τους εργαλεία ανάπτυξης εφαρμογών Ε.Π.

3.7 Συμπεράσματα για το 3^ο κεφάλαιο

Στο προηγούμενο κεφάλαιο εντοπίστηκαν αρχικά τέσσερις ομάδες παραγόντων που συμμετέχουν στη μαθησιακή διαδικασία (θεωρητικό υπόβαθρο, μαθητές, μέσα υλοποίησης και εξωτερικοί παράγοντες) και από την εξέτασή τους διατυπώθηκαν τρεις αρχές μάθησης, ότι η μάθηση είναι ατομικό φαινόμενο, ότι είναι και κοινωνικό φαινόμενο, και η αρχή της απροσδιοριστίας.

Οι αρχές μάθησης με τη σειρά τους αποτέλεσαν το κριτήριο εξέτασης των γενιών εκπαιδευτικής χρήσης των ηλεκτρονικών υπολογιστών. Από την εξέταση προέκυψαν οι ατέλειες των τριών πρώτων γενιών και το συμπέρασμα ότι μόνο η τέταρτη γενιά, που αντιπροσωπεύει τις εφαρμογές Ε.Π., μπορεί να ικανοποιήσει όλες τις αρχές. Το συγκριτικό πλεονέκτημα της Ε.Π. έναντι των προηγούμενων εκπαιδευτικών εφαρμογών εντοπίστηκε αρχικά σε δύο σημεία:

1. Στη μη-συμβολική αλληλεπίδραση ανθρώπου-μηχανής, δηλαδή στο γεγονός της αμεσότητας της εμπειρίας του χρήστη σ' ένα εικονικό περιβάλλον, χωρίς να μεσολαβούν απαραίτητα η γλώσσα, οι αριθμοί και οποιαδήποτε άλλο σύστημα συμβόλων.
2. Στις εμπειρίες “πρώτου προσώπου”, που είναι αποτέλεσμα της μη-συμβολικής αλληλεπίδρασης. Οι εμπειρίες “πρώτου προσώπου” είναι ιδιαίτερα σημαντικές γιατί και στην πραγματικότητα οι περισσότερες ενέργειες στην καθημερινή ζωή γίνονται έτσι.

Τα παραπάνω επιτυγχάνονται εξαιτίας της απουσίας του περιβάλλοντος διεπαφής και του υψηλού βαθμού των αλληλεπιδράσεων στα εικονικά περιβάλλοντα. Αυτά, μαζί με το ρεαλισμό της εφαρμογής, οδηγούν στο τρίτο συγκριτικό πλεονέκτημα της Ε.Π., που είναι η εμβύθιση. Ως εμβύθιση εννοείται η αίσθηση της “παρουσίας” στον εικονικό κό-

σμο εξαιτίας της παραπλάνησης των αισθήσεων του χρήστη. Όπως διαπιστώθηκε, ο μεγάλος ή μικρός βαθμός εμπύθισης εξαρτάται από τις συσκευές που χρησιμοποιούνται, οι οποίες ουσιαστικά καθορίζουν και τα διάφορα είδη Ε.Π. (πλήρως εμβυθισμένη, ημι-εμβυθισμένη, επιτραπέζια, κτλ.).

Ωστόσο η εμπύθιση είναι μία έννοια που έχει προκαλέσει αρκετό διάλογο σε σχέση με τις δυνατότητες της τεχνολογίας να προσφέρει ικανοποιητικό βαθμό εμπύθισης, αλλά κυρίως λόγω της αντιφατικότητας των πειραματικών δεδομένων για το ρόλο της στη μαθησιακή διαδικασία. Φαίνεται ότι και ψυχολογικοί και ψυχοσωματικοί παράγοντες παίζουν σημαντικό ρόλο, όπως επίσης και ο αλληλεπιδραστικός χαρακτήρας των εφαρμογών Ε.Π. Έτσι υιοθετήσαμε την άποψη ότι η Ε.Π. είναι κυρίως μία νοητική κατάσταση όπου ο ρόλος της τεχνολογίας είναι να παρέχει πλούσιες αισθητηριακές, γνωστικές, συναισθηματικές, αντιληπτικές και κοινωνικές εμπειρίες.

Πράγματι, με την Ε.Π. είναι δυνατό να επιτευχθούν πλούσιες εμπειρίες, γιατί ο χρήστης μπορεί να: α) μεταχειρίζεται και να αλληλεπιδρά με αντικείμενα όπως και στο φυσικό κόσμο, αλλά μπορεί επίσης να αλλάζει τις ιδιότητές τους και να μεταχειρίζεται αντικείμενα που δεν έχουν φυσική μορφή, β) ελέγχει το χρόνο, γ) δέχεται πληροφορίες που κάτω από άλλες συνθήκες δε θα ήταν διαθέσιμες στις ανθρώπινες αισθήσεις και δ) αλληλεπιδρά με άλλους χρήστες, παρόντες στον εικονικό κόσμο.

Αναφορικά με το είδος Ε.Π. που θα ήταν το καταλληλότερο για εφαρμογή στο σχολικό περιβάλλον, ύστερα από την ανάλυση των πλεονεκτημάτων και των μειονεκτημάτων των διαφόρων συσκευών και των ειδών Ε.Π. προέκυψε ότι η επιτραπέζια Ε.Π. είναι το είδος που μπορεί να εφαρμοστεί άμεσα, μαζικά και με μικρό ή μηδενικό για το χρήστη κόστος.

Τέλος, αναλύθηκαν οι λόγοι για τους οποίους ο δομητισμός προκάλεσε μία σημαντική αλλαγή στους τρόπους με τους ερμηνεύεται η διαδικασία της μάθησης. Παράλληλα, διαπιστώθηκε η συμβατότητα της Ε.Π. με τη φιλοσοφία του δομητισμού. Όμως σ' αυτό το σημείο πρέπει να τονιστεί πως υιοθετούμε τη θέση ότι η Ε.Π. είναι κάτι παραπάνω από ένα απλό εργαλείο υλοποίησης των αντιλήψεων που επιβάλλει μία και μόνο θεωρία μάθησης. Υπάρχουν συγκεκριμένοι λόγοι που επιτρέπουν αυτή την άποψη.

Όπως επισημάνει ο Μικρόπουλος [84], η Ε.Π. επιδρά άμεσα στις αισθήσεις και δίνει στους μαθητές ενεργό ρόλο. Επενεργεί έτσι στην αρχική φάση της μαθησιακής διαδικασίας, λειτουργώντας περισσότερο ως μεταδότης αισθήσεων και ενισχυτής εμπειριών, παρά ως γνωστικό εργαλείο. Αυτό το γεγονός από μόνο του την καθιστά ανεξάρτητη των παιδαγωγικών θεωριών.

Επιπρόσθετα, ο τρόπος με τον οποίο θα υλοποιηθεί μία εφαρμογή μπορεί, με μικρές αλλαγές, να ικανοποιεί τα χαρακτηριστικά μίας ή περισσότερων θεωριών μάθησης. Για παράδειγμα, αν το περιβάλλον είναι εντελώς εξερευνησιμο και αλληλεπιδραστικό, ικανοποιούνται οι αρχές του δομητισμού. Αν όμως “κλειδώσουν” ορισμένες από τις περιοχές του και περιοριστεί η αλληλεπίδραση σε κάποια αντικείμενα, ορίζοντας με αυτό τον τρόπο συγκεκριμένες διαδρομές όπου ο μαθητής αντιμετωπίζει ένα κομμάτι της συνολικής γνώσης, τότε το ίδιο περιβάλλον ικανοποιεί τις αρχές των γνωστικών θεωριών μάθησης.

Αν προστεθούν στοιχεία επιβράβευσης του μαθητή, όπως ο αριθμός των σωστών του απαντήσεων με τη μορφή σκορ, ίσως κάποιος να διακρίνει επιδράσεις από το συμπεριφορισμό. Τμήματα της εφαρμογής και όχι αναγκαστικά το σύνολό της είναι εφικτό να κινούνται προς τη μία ή την άλλη κατεύθυνση.

Η αξία της δυνατότητας να συνυπάρχουν στην ίδια εφαρμογή διαφορετικές διδακτικές φιλοσοφίες είναι ιδιαίτερα μεγάλη. Παρέχεται η δυνατότητα ενός ενιαίου εργαλείου πειραματικής σύγκρισης των διαφόρων θεωριών μάθησης. Ο κάθε ερευνητής υλοποιεί μία εφαρμογή με βάση τις παιδαγωγικές αντιλήψεις της επιλογής του και την εφαρμόζει σε κάποια ομάδα-στόχο. Εντελώς διαφορετικές εφαρμογές υλοποιούνται από άλλους ερευνητές και εφαρμόζονται σε διαφορετικές ομάδες-στόχους. Η σύγκριση των αποτελεσμάτων είναι από πολύ δύσκολη έως αδύνατη. Οι εφαρμογές της Ε.Π. επιτρέπουν την υλοποίηση του ίδιου θέματος με πολλές διαφορετικές παραλλαγές και συνδυασμούς. Συνεπώς, δίνεται η δυνατότητα να υπάρχουν συγκρίσιμα μεταξύ τους αποτελέσματα.

Τέλος, έγιναν δύο επισημάνσεις: α) ο τρόπος υλοποίησης οποιασδήποτε εφαρμογής θα πρέπει να είναι τεχνολογικά επίκαιρος ώστε αυτή να εκμεταλλεύεται τις προόδους στον τομέα της Πληροφορικής και β) χρειάζεται να εξευρεθούν ευέλικτα και εύκολα στη χρήση τους εργαλεία ανάπτυξης εφαρμογών Ε.Π.

Κεφάλαιο 4^ο

Τεχνικά θέματα υλικού και λογισμικού

Η αξία της Ε.Π. ως παιδαγωγικού-διδασκτικού εργαλείου είναι καλά τεκμηριωμένη στη βιβλιογραφία, όπως φάνηκε από την ανάλυση στα προηγούμενα κεφάλαια. Από τότε που έκανε για πρώτη φορά την εμφάνισή της χρησιμοποιείται κατά κόρον στην εξάσκηση και εκπαίδευση ειδικών επαγγελματικών κατηγοριών, όπως πιλότων, στρατιωτικών, γιατρών, αλλά και για την οπτικοποίηση δεδομένων. Σημαντικά βήματα έγιναν και προς την κατεύθυνση της χρήσης της για την εκπαίδευση των παιδιών. Παρά όμως την αναγνωρισμένη αξία της Ε.Π., το μέσο δε γνωρίζει την αναμενόμενη διάδοση και στην ουσία παραμένει ανεκμετάλλευτο.

Πράγματι, τα οικονομικά στοιχεία συνηγορούν σ' αυτή την άποψη. Οι εφαρμογές προσομοίωσης και γενικότερα οι εφαρμογές, το υλικό και το λογισμικό Ε.Π. καταλαμβάνουν ένα σχετικά μικρό κομμάτι της βιομηχανίας της Πληροφορικής. Κατά το έτος 2002 πουλήθηκαν 308.000 συστήματα, ο συνολικός κύκλος εργασιών ήταν της τάξης των 32,4 δισεκατομμυρίων ευρώ με 13.000 επιχειρήσεις άμεσα ή έμμεσα εμπλεκόμενες και 650.000 άτομα να απασχολούνται –κυρίως ως χρήστες– στο συγκεκριμένο τομέα [85].

Σ' αυτή την κατάσταση συντελούν πολλοί παράγοντες με κύριο το αυξημένο κόστος απόκτησης ενός πλήρους συστήματος Ε.Π. Αν και η πλειοψηφία των συστημάτων είναι εισαγωγικού επιπέδου με κόστος περίπου 20.000 ευρώ (υλικό και λογισμικό παραγωγής εφαρμογών), εντούτοις το μέσο κόστος είναι πολλαπλάσιο. Αυτό οφείλεται στο γεγονός ότι υπάρχει σημαντική επιβάρυνση στο κόστος όσο πιο εξειδικευμένη είναι η εφαρμογή και όσο πιο εξειδικευμένα περιφερειακά χρησιμοποιούνται.

Το σημαντικότερο μέρος του κόστους ενός συστήματος Ε.Π. κατανέμεται στο *υλικό απεικόνισης (display systems)* 16%, στο *υλικό παραγωγής εικόνων (image generators)* 18% και στο *λογισμικό* 18% [85]. Από τις τρεις αυτές κατηγορίες δαπανών η πιο ανελαστική είναι το λογισμικό παραγωγής και εκτέλεσης εφαρμογών Ε.Π. Το είδος, οι δυνατότητες και το κόστος για την απόκτηση του υλικού ποικίλλει ανάλογα με την εφαρμογή και μπορεί να συμπιεστεί σημαντικά. Αντίθετα, το κόστος του λογισμικού παραμένει σταθερό, γιατί ανεξάρτητα από το είδος της εφαρμογής, υπάρχουν πολύ λίγες επιλογές στο λογισμικό που θα χρησιμοποιηθεί, είτε για την ανάπτυξη είτε για την εκτέλεση εφαρμογών.

Υπάρχουν ωστόσο ενδείξεις ότι η κατάσταση αλλάζει. Η κρίση της περιόδου 1999-2001, όπου παρατηρήθηκε αρνητική ανάπτυξη, ξεπεράστηκε και ο κλάδος αναπτύσσεται πλέον με ετήσιο ρυθμό που ξεπερνά το 29%, πραγματοποιώντας έναν κύκλο εργασιών της τάξης των 56,7 δισεκατομμυρίων ευρώ το 2006 [85]. Αυτό οφείλεται στη

μεταστροφή του ενδιαφέροντος προς τα τρισδιάστατα γραφικά, που παραδοσιακά ανήκει στον κλάδο της Ε.Π., με κινητήρια δύναμη τη βιομηχανία της ψυχαγωγίας.

Αλλαγή της κατάστασης και διεύρυνση της αγοραστικής βάσης στην οποία απευθύνονται εφαρμογές της Ε.Π. σημαίνει και μείωση του κόστους απόκτησης ενός συστήματος Ε.Π. Ο ακριβής ρυθμός μείωσης και ο χρονικός ορίζοντας στον οποίο ένα πλήρες σύστημα Ε.Π. θα είναι προσιτό στο μέσο χρήστη είναι δύσκολο να προβλεφθεί. Εξάλλου, οι προσδοκίες για ραγδαία μείωση του κόστους διαψεύστηκαν ήδη μία φορά την περίοδο 1999-2001, πράγμα που κάνει μία πρόβλεψη ακόμα πιο επισφαλή.

Όλα τα παραπάνω οδηγούν στη διατύπωση του εξής προβληματισμού: Είναι δυνατόν, με τα σημερινά δεδομένα, να αναπτυχθεί μία εκπαιδευτική εφαρμογή Ε.Π. υψηλών ποιοτικών προδιαγραφών στην οποία το κόστος υλικού και λογισμικού για τον τελικό χρήστη να μην υπερβαίνει το κόστος ενός καλού ηλεκτρονικού υπολογιστή;

Μία πρώτη απάντηση στο παραπάνω ερώτημα δίνεται από το είδος Ε.Π. που ενδείκνυται για άμεση εφαρμογή στο σχολικό περιβάλλον. Η επιτραπέζια Ε.Π., όπως αναλύθηκε, χρησιμοποιεί σε γενικές γραμμές τον τυπικό εξοπλισμό που συναντάται σ' όλα τα υπολογιστικά συστήματα και δε χρησιμοποιεί ιδιαίτερα πολύπλοκες και εξειδικευμένες περιφερειακές συσκευές. Αυτό λύνει εν μέρει το θέμα του κόστους για τον τελικό χρήστη.

Δε λύνει όμως το θέμα της ποιότητας της εφαρμογής. Με τον όρο “ποιότητα εφαρμογής” εννοείται η όσο το δυνατόν πιο πιστή απόδοση της πραγματικότητας στον εικονικό κόσμο, ο ρεαλισμός. Η ποιότητα αφορά και τη μεριά του τελικού χρήστη αλλά και τη μεριά του σχεδιαστή μίας εφαρμογής Ε.Π. και εξαρτάται από την αντιμετώπιση μίας σειράς τεχνικών θεμάτων που αφορούν το υλικό και το λογισμικό, όπως αναλύονται στη συνέχεια.

4.1 Απαιτήσεις και προβλήματα από τις κάρτες γραφικών

Πριν γίνει η ανάλυση των απαιτήσεων από το υλικό και το λογισμικό εφαρμογών Ε.Π., είναι απαραίτητο να γίνει αναφορά σε ορισμένα θέματα της λειτουργίας ενός ηλεκτρονικού υπολογιστή και των υποσυστημάτων του, ώστε να γίνει κατανοητό γιατί τα προβλήματα και οι περιορισμοί του υλικού που πρέπει να ξεπεραστούν αφορούν κυρίως τις κάρτες γραφικών. Οι εφαρμογές Ε.Π. απαιτούν γενικά πολύ καλούς ηλεκτρονικούς υπολογιστές. Εξετάζοντας ποια υποσυστήματα ενός ηλεκτρονικού υπολογιστή εμπλέκονται κατά την εκτέλεση μίας εφαρμογής Ε.Π., τρία είναι τα στοιχεία που ενδιαφέρουν: α) η ταχύτητα επεξεργασίας, β) η μνήμη και ο αποθηκευτικός χώρος και γ) τα γραφικά, η ποιότητα και η ταχύτητα της απεικόνισής τους.

Σήμερα ακόμα και ένα μέσων δυνατοτήτων υπολογιστικό σύστημα –με ικανότητα να εκτελέσει εφαρμογές επιτραπέζιας Ε.Π.– διαθέτει ικανοποιητική επεξεργαστική ισχύ, παραπάνω από αρκετό αποθηκευτικό χώρο και έχει κόστος προσιτό για το μέσο χρήστη. Εκεί που υπάρχει μία σχετική υστέρηση είναι στον τομέα των τρισδιάστατων γραφικών. Η ρεαλιστική απεικόνιση ενός περιβάλλοντος απαιτεί μεγάλη επεξεργαστική ισχύ από τις κάρτες γραφικών.

Στα παλαιότερα υπολογιστικά συστήματα, μέχρι το 1993-1994, ο επεξεργαστής ήταν εκείνο το ηλεκτρονικό εξάρτημα που διεκπεραίωνε το σύνολο του υπολογιστικού φορτίου. Σημαντικό μέρος από το φορτίο αυτό καταλάμβαναν οι απαιτούμενοι υπολογισμοί και διεργασίες για τη δημιουργία της εικόνας που θα παρουσιαστεί στην οθόνη του υπολογιστή. Η κάρτα γραφικών, ή προσαρμογέας οθόνης, αποτελούσε εκείνο το εξάρτημα του υπολογιστή που απλώς αναλάμβανε το τελευταίο σκέλος της διαδικασίας, δηλαδή την προβολή της εικόνας στην οθόνη.

Σε απλές εφαρμογές αυτό δεν αποτελούσε ιδιαίτερο πρόβλημα. Σε εφαρμογές όμως πλούσιες σε γραφικά, και μάλιστα κινούμενα, ο επεξεργαστής επιβαρυνόταν με εξαιρετικά σημαντικό φορτίο, στο οποίο δεν μπορούσε να ανταποκριθεί ικανοποιητικά. Η κατάσταση επιδεινωνόταν ακόμα περισσότερο όταν τα γραφικά ήταν τρισδιάστατα. Είναι εύκολα αντιληπτό ότι μέρος από τους απαραίτητους υπολογισμούς έπρεπε να περάσει στην κάρτα γραφικών.

Αρχικά αναπτύχθηκαν ανεξάρτητες κάρτες που αναλάμβαναν την επιτάχυνση των διεργασιών κίνησης και προβολής τρισδιάστατων γραφικών (*3D acceleration cards*) και πολύ σύντομα τα κυκλώματα αυτά ενσωματώθηκαν πλήρως στις κάρτες γραφικών [86]. Σταδιακά, και ενώ οι απαιτήσεις των εφαρμογών διαρκώς αυξάνονταν, οι κάρτες γραφικών αποκτούσαν ολοένα και περισσότερες δυνατότητες.

Αν και δεν είναι απόλυτα ακριβές, ωστόσο είναι πολύ κοντά στην αλήθεια η άποψη ότι σήμερα μία τυπική κάρτα γραφικών είναι σε θέση να αναλάβει πλήρως οποιαδήποτε υπολογισμό αφορά τα γραφικά, λειτουργώντας περίπου ως ανεξάρτητος υπολογιστής μέσα στον υπολογιστή. Χρησιμοποιεί μεγάλο ποσό δικής της μνήμης, ξεχωριστής από την κεντρική μνήμη του συστήματος, χρησιμοποιεί δικό της επεξεργαστή (*Graphics Processing Unit, G.P.U.*) σε αντιστοιχία με τον κεντρικό επεξεργαστή (*Central Processing Unit, C.P.U.*), ενώ επικοινωνεί με ταχύτατους διαύλους (*Accelerated Graphics Port-A.G.P. X8 ή PCI Express*) με το υπόλοιπο σύστημα.

Ο κεντρικός επεξεργαστής, απαλλαγμένος σε μεγάλο βαθμό από οτιδήποτε έχει σχέση με τα γραφικά, εκτελεί όλες τις υπόλοιπες εντολές της εφαρμογής με ικανοποιητική

πλέον ταχύτητα. Όμως η πραγματικότητα είναι διαφορετική. Η αιτία βρίσκεται σε επτά σημεία που, τουλάχιστον με τα σημερινά δεδομένα, δεν μπορούν να ικανοποιούνται ταυτόχρονα στο μέγιστο βαθμό το καθένα, αν και κάτι τέτοιο είναι απολύτως απαραίτητο σε μία εφαρμογή Ε.Π. [87]. Τα σημεία αυτά είναι:

- Το χρωματικό βάθος
- Ο ρυθμός ανανέωσης
- Ο αριθμός πολυγώνων
- Οι υφές
- Η ανάλυση
- Ο αριθμός των καρέ/δευτερόλεπτο
- Οι φωτοσκιάσεις

Χρωματικό βάθος

Με τον όρο αυτό εννοείται ο αριθμός των χρωμάτων που μπορεί να παρουσιάζει η κάρτα γραφικών και η οθόνη. Καθορίζεται από τον αριθμό των bit που χρησιμοποιούνται για τη χρωματική περιγραφή ενός *εικονοστοιχείου* (*pixel*). Έτσι, για βάθος χρώματος 8 bit παρουσιάζονται $2^8 = 256$ χρώματα, που είναι πολύ λίγα, με 24 bit παρουσιάζονται περίπου 16,8 εκατομμύρια χρώματα (*πραγματικό χρώμα*, *true color*), ενώ είναι δυνατό να εμφανίζονται στην οθόνη χρώματα που παράγονται από 32 bit [88].

Θα αναρωτηθεί κανείς αν χρειάζονται τόσα πολλά χρώματα, αφού το ανθρώπινο μάτι δεν είναι σε θέση να τα συλλάβει. Αν όμως θέλουμε να αποδίδονται φυσικά οι σκιάσεις, το σβήσιμο και η ανάμειξη των χρωμάτων μεταξύ τους, είναι απαραίτητο βάθος χρώματος 24 bit.

Ανάλυση

Με τον όρο αυτό εκφράζεται η πυκνότητα των *εικονοστοιχείων* της εικόνας. Όσο περισσότερα *εικονοστοιχεία* σχηματίζουν μία εικόνα, τόσο καλύτερη είναι η ποιότητά της και παύουν να είναι ορατές ορισμένες ατέλειες, όπως μειωμένη ευκρίνεια (θολούρα), αιχμές, κτλ., πλησιάζοντας την ποιότητα μίας φωτογραφίας. Ειδικότερα, όταν γίνεται αναφορά στην ανάλυση της οθόνης, χρησιμοποιείται το γινόμενο των κάθετων και οριζόντιων *εικονοστοιχείων* που παρουσιάζονται σ' αυτή, για παράδειγμα 640x480, που θεωρείται πολύ χαμηλή ανάλυση [88]. Εξαιρετικές θεωρούνται αναλύσεις πάνω από 1600x1200, που όμως λίγα μέσα απεικόνισης μπορούν να πραγματοποιήσουν. Συνήθως μία εφαρμογή τρισδιάστατων γραφικών αναπτύσσεται σε ανάλυση πάνω από 2048x1536 και στη συνέχεια παρουσιάζεται στον τελικό χρήστη σε πολύ μικρότερη ανάλυση, της τάξης του 1024x768.

Ρυθμός ανανέωσης

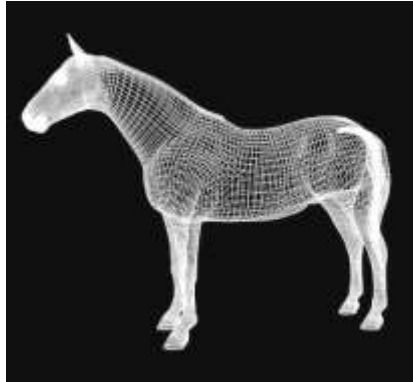
Είναι η συχνότητα με την οποία η κάρτα γραφικών και η οθόνη σχηματίζουν και ανανεώνουν την εικόνα κάθε δευτερόλεπτο. Υψηλές τιμές σ' αυτή την παράμετρο εξαφανίζουν το τρεμόπαιγμα που αντιλαμβάνεται το μάτι κατά την ανανέωση της εικόνας. Για παράδειγμα, με μία συχνότητα ανανέωσης 60Hz το τρεμόπαιγμα είναι εμφανές και πολύ κουραστικό. Εργονομικός θεωρείται ένας ρυθμός ανανέωσης 85Hz, το τρεμόπαιγμα δε γίνεται αντιληπτό πάνω από τα 100Hz, ενώ τιμές πάνω από 120Hz είναι επιθυμητές [89].

Καρέ ανά δευτερόλεπτο

Το ανθρώπινο μάτι αντιλαμβάνεται ως ενιαία κίνηση όταν του παρουσιάζονται περισσότερες από 25 συνεχόμενες εικόνες το δευτερόλεπτο (*καρέ/δευτερόλεπτο, frames per second, fps*) που διαφέρουν ελάχιστα μεταξύ τους. Το φαινόμενο ονομάζεται *μετέκασμα* και πάνω σ' αυτό στηρίζονται η τηλεόραση, ο κινηματογράφος, τα κινούμενα σχέδια και τα κινούμενα γραφικά. Κάτω απ' αυτή την τιμή παρουσιάζονται διακοπές και ανομοιογένειες στην κίνηση. Ορισμένοι άνθρωποι είναι σε θέση να αντιληφθούν τέτοια προβλήματα ακόμα και σε μεγαλύτερες τιμές και για το λόγο αυτό το αμερικανικό πρότυπο τηλεοπτικού σήματος NTSC χρησιμοποιεί 29,75fps. Για τα κινούμενα γραφικά η τιμή αυτή συνιστάται να ξεπερνά τα 60fps σε κάθε δεδομένη στιγμή [90].

Αριθμός πολυγώνων

Τα τρισδιάστατα γραφικά δημιουργούνται από σημεία. Τα σημεία ενώνονται μεταξύ τους με γραμμές, δημιουργώντας ακμές και έδρες. Ένας κύβος, για παράδειγμα, αποτελείται από 8 σημεία, έτσι τοποθετημένα ώστε να δημιουργούνται 12 ακμές και 6 έδρες. Η πιο συνηθισμένη έδρα σ' ένα τρισδιάστατο γραφικό έχει το σχήμα τριγώνου, αν και είναι δυνατό να είναι οποιοδήποτε άλλο πολύγωνο. Όσο μεγαλύτερος είναι ο αριθμός των πολυγώνων που αποτελούν ένα τρισδιάστατο γραφικό, τόσο αυτό πλησιάζει σε ομοιότητα το πραγματικό αντικείμενο. Για να αναπαρασταθούν με λεπτομέρεια περίπλοκα αντικείμενα, όπως ένα άλογο, απαιτείται ένας πραγματικά μεγάλος αριθμός πολυγώνων (*εικόνα 4.1*).



Εικόνα 4.1 Τρισδιάστατο μοντέλο (44.000 πολύγωνα)

Ένας εικονικός κόσμος αποτελείται από εκατοντάδες ή και χιλιάδες αντικείμενα, οπότε ο συνολικός αριθμός πολυγώνων γίνεται τεράστιος. Επιπρόσθετα, πολλά από τα αντικείμενα κινούνται, αλλάζοντας σχετικές θέσεις στο χώρο, με αποτέλεσμα να δημιουργείται η ανάγκη επανασχεδιασμού τους τόσες φορές όσες και τα καρέ το δευτερόλεπτο. Στα τεχνικά χαρακτηριστικά των καρτών γραφικών αναγράφεται ότι αυτές είναι σε θέση να κινούν αρκετές δεκάδες εκατομμύρια πολύγωνα το δευτερόλεπτο. Αυτό όμως δεν είναι ακριβές. Ισχύει μόνο όταν η αποκλειστική εργασία της κάρτας γραφικών είναι να παρουσιάζει πολύγωνα και τίποτε άλλο. Στην πραγματικότητα, η κάρτα γραφικών εκτελεί πολλές ακόμα λειτουργίες, με αποτέλεσμα η τιμή αυτή να είναι κατά πολύ μικρότερη [87].

Φωτοσκιάσεις

Ο τρόπος με τον οποίο φωτίζονται τα αντικείμενα και δημιουργούνται οι σκιές είναι από τις πιο απαιτητικές εργασίες αναφορικά με την υπολογιστική ισχύ που απαιτείται [91]. Οι φωτοσκιάσεις συντελούν σε μεγάλο βαθμό στη δημιουργία της αίσθησης του βάθους και συμβάλλουν καθοριστικά στη συνολική ποιότητα του εικονικού κόσμου (εικόνα 4.2). Ο υπολογισμός των φωτοσκιάσεων γίνεται κατά πολύ πιο περίπλοκος όταν η φωτεινή πηγή κινείται, όπως συμβαίνει με τα φώτα ενός αυτοκινήτου, φωτίζοντας και σκιάζοντας διαδοχικά τα αντικείμενα. Περίπλοκοι είναι επίσης οι υπολογισμοί όταν υπάρχουν πολλαπλές πηγές φωτισμού που δημιουργούν πολλαπλές σκιάσεις.



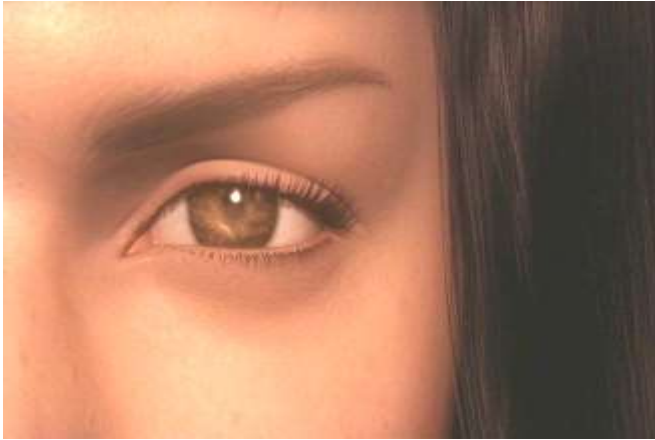
Εικόνα 4.2 Τρισδιάστατα μοντέλα μικρού και μεγάλου αριθμού πολυγώνων, απλών και περίπλοκων φωτοσκιάσεων αντίστοιχα

Υφές

Ένα τρισδιάστατο μοντέλο, αφού κατασκευαστεί από πολύγωνα, πρέπει να “ντυθεί” με τα κατάλληλα χρώματα, αλλά κυρίως με την υφή εκείνη που θα του προσδώσει τις συγκεκριμένες ιδιότητες του αντικειμένου που αναπαριστά. Για παράδειγμα, ένα μεταλλικό αντικείμενο έχει διαφορετική υφή από ένα ρούχο του ίδιου χρώματος. Το μεταλλικό αντικείμενο είναι γυαλιστερό και αντανακλά το φως, στο ρούχο θα πρέπει να φαίνεται η ύφανσή του κ.ο.κ. Οι υφές είναι συνήθως εικόνες υψηλής ανάλυσης έτσι διαμορφωμένες ώστε να καλύπτουν σωστά το μοντέλο [91]. Το ανθρώπινο δέρμα διαθέτει από τις πιο περίπλοκες υφές, γιατί έχει πόρους, πολλές αποχρώσεις, τυχαίες ατέλειες και αντανακλά το φως μ' έναν ιδιαίτερο τρόπο (εικόνα 4.3).

Στον τομέα των υφών, αν και όχι αποκλειστικά, υπάγονται οι διάφοροι βαθμοί διαφάνειας που μπορεί να έχει ένα αντικείμενο, καθώς επίσης και οι αντανακλάσεις. Το γυαλί είναι διαφανές, άρα πρέπει να είναι ορατά αντικείμενα πίσω απ' αυτό και ταυτόχρονα θα πρέπει να γίνεται αντιληπτή η ύπαρξή του. Το νερό παρουσιάζει διαφάνεια που ελαττώνεται ανάλογα με το βάθος, ενώ στην ομίχλη συνυπάρχουν διαφάνεια και θολότητα.

Οι αντανακλάσεις είναι μία ιδιαίτερη περίπτωση. Η υφή του αντικειμένου πρέπει να είναι τέτοια, ώστε να επιτρέπει την απεικόνιση άλλων αντικειμένων πάνω του. Ο καθρεπτισμός είναι η πιο περίπλοκη αντανάκλαση, γιατί πρέπει να σχεδιάζεται όλος ο εικονικός κόσμος επάνω στο αντικείμενο, ανεστραμμένος και κινούμενος.



Εικόνα 4.3 Λεπτομέρεια από μοντέλο εξαιρετικά μεγάλου αριθμού πολυγώνων, με την υφή του ανθρώπινου δέρματος. Αξιοσημείωτες είναι οι ατέλειες στο δέρμα και η αντανάκλαση στο μάτι.

Τέλος, υπάρχει η κατηγορία των κινούμενων υφών. Τέτοιες υφές χρησιμοποιούνται για την επιφάνεια του νερού, για τη λάβα του ηφαιστείου, για τη φωτιά, κτλ. Σε κάθε περίπτωση η υφή, επειδή είναι δεμένη με το μοντέλο, θα πρέπει να κινείται μαζί με αυτό, να μεταβάλλει διαρκώς τις ιδιότητες του φωτισμού, της σκίασης και της αντανάκλασης που έχει. Ο αριθμός των υπολογισμών για το σύνολο των υφών στα αντικείμενα είναι τεράστιος [87].

4.2 Τεχνικές για την ικανοποιητική εκτέλεση της εφαρμογής από το υλικό του τελικού χρήστη

Αυτό που προκύπτει, αν αξιοποιηθούν και τα επτά παραπάνω σημεία στο έπακρο, είναι ένας πολύ υψηλός βαθμός ρεαλισμού για τον εικονικό κόσμο. Ο ρεαλισμός είναι το βασικό ζητούμενο για τους σχεδιαστές εφαρμογών Ε.Π. Οδηγεί τον τελικό χρήστη στην ταύτισή του με ό,τι του παρουσιάζεται από το μέσο απεικόνισης και τον κάνει να πιστεύει ότι ο εικονικός κόσμος μέσα στον οποίο βρίσκεται είναι πραγματικός. Αυτή η “εμβύθιση” του χρήστη, που είναι μία ιδιαίτερη νοητική κατάσταση, επιτυγχάνεται με τη συμμετοχή και άλλων παραγόντων (αλληλεπιδράσεις, ήχος, συνεργασία με άλλους χρήστες, πλοήγηση στον κόσμο, κ.ά.), όμως σ' αυτό το κεφάλαιο αναλύεται αποκλειστικά ο τομέας της εικόνας, γιατί παρουσιάζει τα περισσότερα προβλήματα.

Τα προβλήματα ξεκινούν από τη διαπίστωση ότι, αν ανατεθεί στο υλικό και κυρίως στην κάρτα γραφικών μεγάλο υπολογιστικό φορτίο, αυτή αδυνατεί να αντεπεξέλθει. Έτσι είναι αναγκαία η περικοπή στοιχείων από τον εικονικό κόσμο σε βάρος του ρεαλισμού. Για να περιπλακεί ακόμα περισσότερο η κατάσταση, θα πρέπει να σημειωθεί ότι ο ρεαλισμός είναι υποκειμενική παράμετρος. Αυτό που συναρπάζει και συγκινεί

έναν άνθρωπο, δε συγκινεί κάποιον άλλο. Αν εξαλειφθούν οι περίπλοκες φωτοσκιάσεις, σε κάποιους μπορεί να περάσει απαρατήρητο, ενώ κάποιοι άλλοι μπορεί να το θεωρήσουν σοβαρή παράλειψη. Επίσης, ο σχεδιαστής δε γνωρίζει εκ των προτέρων την ακριβή διαμόρφωση του υλικού που χρησιμοποιεί ο τελικός χρήστης. Με ποια κριτήρια λοιπόν είναι υποχρεωμένος να λειτουργήσει ο σχεδιαστής μίας εφαρμογής Ε.Π.;

Η πρώτη απόφαση που λαμβάνει είναι ο ορισμός των ελάχιστων απαιτήσεων από το υλικό του τελικού χρήστη. Ορίζεται ένα σύνολο ελάχιστων προδιαγραφών τις οποίες ελέγχει ο χρήστης πριν χρησιμοποιήσει τη συγκεκριμένη εφαρμογή. Όμως μία εφαρμογή που εκτελείται σε υλικό ελάχιστων απαιτήσεων δεν εκτελείται ικανοποιητικά. Πρακτικά αυτό σημαίνει ότι ο χρήστης φροντίζει να έχει υλικό που ξεπερνά αυτές τις προδιαγραφές. Ισχύει ο κανόνας: “όσο περισσότερα και γρηγορότερα, τόσο καλύτερα”.

Αν οριστούν ελάχιστες προδιαγραφές που ξεπερνούν κατά πολύ τον εξοπλισμό του μέσου χρήστη, υπάρχει ο κίνδυνος η εφαρμογή να απευθύνεται σ' ένα πολύ περιορισμένο κοινό. Αν οριστούν πολύ χαμηλές προδιαγραφές, τότε η εφαρμογή είναι σίγουρο ότι δε θα έχει υψηλό βαθμό ρεαλισμού. Τέλος, επειδή μία εφαρμογή αναπτύσσεται σε υπολογιστές με συγκεκριμένη διαμόρφωση, υπάρχει πάντα η περίπτωση μία μικρή αλλαγή στο υλικό, για παράδειγμα μία κάρτα γραφικών άλλης εταιρείας αλλά παρόμοιων δυνατοτήτων να οδηγήσει σε αδυναμία εκτέλεσης της εφαρμογής. Συνεπώς οι προδιαγραφές πρέπει να είναι αποτέλεσμα εξαντλητικών δοκιμών.

Ο σχεδιαστής έχει πάντοτε κατά νου τη διατήρηση μίας σταθερά υψηλής τιμής καρέ/δευτερόλεπτο. Από τα επτά σημεία που παρουσιάστηκαν, αυτό κρίνεται ότι είναι το ουσιαστικότερο για την ποιότητα της εφαρμογής. Η κίνηση θέτει τους σοβαρότερους περιορισμούς, γιατί πολλαπλασιάζει τους υπολογισμούς που απαιτούνται για κάθε δευτερόλεπτο κίνησης σε κάθε κινούμενο αντικείμενο.

Πρέπει να διευκρινιστεί ότι κίνηση είναι και η μετακίνηση του οπτικού πεδίου του χρήστη από τη μία θέση στην άλλη. Όταν ο χρήστης στρίβει το εικονικό κεφάλι του προς μία κατεύθυνση, στατικά και κινούμενα αντικείμενα πρέπει να κινηθούν ανάλογα, να φανεί ότι μετακινούνται και ότι εμφανίζονται ή εξαφανίζονται από το οπτικό του πεδίο. Αυτή είναι η αιτία για την οποία ο σχεδιαστής θέτει ένα στόχο καρέ/δευτερόλεπτο πολύ μεγάλο π.χ. 60 καρέ/δευτερόλεπτο, ώστε να έχει περιθώρια η τιμή αυτή να πέσει στα 25 καρέ/δευτερόλεπτο, όταν λαμβάνουν χώρα περίπλοκες κινήσεις στον εικονικό κόσμο [90].

Η πιο εύκολη περίπτωση είναι μία εφαρμογή με λίγα και στατικά μοντέλα. Ο σχεδιαστής μπορεί να αξιοποιήσει πιο ελεύθερα όλα τα υπόλοιπα σημεία. Τέτοιες εφαρμογές συναντώνται, για παράδειγμα, στη ιατρική, με μοντέλα ανθρώπινων οργάνων [92]. Κίνηση υπάρχει σε μικρό βαθμό ή δεν υπάρχει καθόλου, αλλά χρειάζεται εξαιρετικά λεπτομερής απεικόνιση ώστε ο ασκούμενος γιατρός να εξοικειωθεί με τα διάφορα όργανα. Η συντριπτική όμως πλειοψηφία των εφαρμογών Ε.Π. αφορά μεγάλο αριθμό από κινούμενα γραφικά. Ο σχεδιαστής επιδίδεται τότε σε μία επίπονη προσπάθεια να αυξήσει τον αριθμό των καρέ/δευτερόλεπτο, που η γενική κατεύθυνσή της είναι η μείωση του υπολογιστικού φορτίου της κάρτας γραφικών.

Πρώτη του επιλογή είναι να μειώσει τον αριθμό των πολυγώνων στα μοντέλα που αναπαριστούν τα διάφορα αντικείμενα του εικονικού κόσμου. Χρησιμοποιούνται διάφορες τεχνικές που φέρνουν κάποια ισορροπία μεταξύ ποιότητας και κίνησης. Το χρωματικό βάθος, τα χρώματα που παρουσιάζονται σε κάθε δεδομένη στιγμή είναι το επόμενο σημείο το οποίο ο σχεδιαστής θυσιάζει. Έτσι, αντί για 32bit χρησιμοποιούνται 16bit χρωματικού βάθους, αν και υπάρχει η δυνατότητα ορισμένα αντικείμενα να έχουν υψηλό χρωματικό βάθος και ορισμένα να έχουν χαμηλότερο.

Επόμενη κίνηση είναι να μειώσει την ανάλυση, που επιδρά στο σύνολο του εικονικού κόσμου. Ο μειωμένος αριθμός των εικονοστοιχείων συμπαρασύρει προς τα κάτω τον αριθμό των υπολογισμών που τα αφορούν, με αντάλλαγμα όμως την απώλεια της λεπτομέρειας και τη σημαντική υποβάθμιση της ποιότητας.

Οι φωτοσκιάσεις αποτελούν και αυτές αντικείμενο προσαρμογής. Με πολλαπλές πηγές φωτισμού δημιουργούνται πολλαπλές σκιάσεις. Ο σχεδιαστής μπορεί να καθορίσει τον αριθμό των σκιών σε κάθε αντικείμενο, απορρίπτοντας τις σκιές που δημιουργούν δευτερεύουσες πηγές φωτισμού. Ο ρυθμός ανανέωσης μπορεί να μειωθεί στα 85Hz, κερδίζοντας μερικά ακόμα καρέ/δευτερόλεπτο.

Τέλος, οι υφές προσαρμόζονται ανάλογα με την απόσταση του αντικειμένου. Όταν το αντικείμενο είναι κοντά στο οπτικό πεδίο του χρήστη, χρησιμοποιούνται υφές υψηλής ανάλυσης και το αντικείμενο αναπαρίσταται από μοντέλο μεγάλου αριθμού πολυγώνων. Όταν το αντικείμενο βρίσκεται σε μεγάλη απόσταση, τότε χρησιμοποιούνται υφές μικρής ανάλυσης και μοντέλο μικρού αριθμού πολυγώνων.

Αναφέρθηκε ότι ο σχεδιαστής δε γνωρίζει εκ των προτέρων ποια ακριβώς διαμόρφωση έχει ο ηλεκτρονικός υπολογιστής του τελικού χρήστη. Έτσι είναι υποχρεωμένος να του μεταβιβάσει μέρος των αποφάσεων που αφορούν την ποιότητα του εικονικού κόσμου. Αυτό γίνεται με τη μορφή ρυθμίσεων στο πρόγραμμα, που επιδρούν στη συ-

μπεριφορά του υλικού που έχει ο χρήστης. Συνήθως μπορούν να μεταβληθούν η ανάλυση, το χρωματικό βάθος, ο ρυθμός ανανέωσης, ο αριθμός των σκιών και η ανάλυση των υφών, αναγκάζοντας το υλικό να συμπεριφερθεί ανάλογα. Αν τα αποτελέσματα δεν είναι ικανοποιητικά, ο χρήστης έχει τη δυνατότητα να επανέλθει στις ρυθμίσεις και να επιχειρήσει διαφορετικούς συνδυασμούς.

Ολοκληρώνοντας την παρουσίαση των απαιτήσεων από το υλικό, θα πρέπει να τονιστεί ότι αυτό εξελίσσεται με πολύ γρήγορους ρυθμούς. Η μία γενιά καρτών γραφικών διαδέχεται την άλλη περίπου κάθε δώδεκα με δεκαοκτώ μήνες. Οι δυνατότητες που ενσωματώνονται, η αύξηση της ταχύτητας εκτέλεσης των υπολογισμών, η αλλαγή της μεθόδου χειρισμού των γραφικών και μία σειρά από άλλα τεχνικά στοιχεία καθιστούν πολλές από τις εφαρμογές ξεπερασμένες σε πολύ σύντομο χρονικό διάστημα. Το θετικό είναι ότι ο σχεδιαστής, με κάθε νέα γενιά, χρειάζεται να κάνει λιγότερους συμβιβασμούς αναφορικά με την ποιότητα της εφαρμογής του. Το αρνητικό είναι ότι χρειάζεται να ξαναγράψει μεγάλο μέρος της, ώστε αυτή να ανταποκρίνεται στα νέα δεδομένα και να εκμεταλλεύεται τις νέες τεχνικές που υιοθετούνται.

Ο άμεσα ωφελημένος από τις τεχνολογικές εξελίξεις είναι ο τελικός χρήστης. Αντικαθιστώντας την παλιά κάρτα γραφικών με μία καινούργια, του δίνεται η δυνατότητα να εκτελέσει με καλύτερες ποιοτικές ρυθμίσεις τις παλαιότερες εφαρμογές, όπως επίσης μπορεί να εκτελέσει καινούργιες εφαρμογές που αξιοποιούν τις νέες δυνατότητες του υλικού. Βέβαια, όπως συμβαίνει με κάθε τεχνολογική εξέλιξη, έτσι και οι κάρτες γραφικών έχουν αρχικά ένα σημαντικό κόστος, το οποίο όμως λόγω του έντονου ανταγωνισμού μειώνεται σημαντικά μέσα στο πρώτο εξάμηνο από την εμφάνισή τους.

4.3 Απαιτήσεις και προβλήματα από το λογισμικό κατασκευής εικονικών κόσμων

Ο σχεδιαστής μίας εφαρμογής έχει ακόμα ένα πρόβλημα που αφορά και πάλι την αξιοποίηση του υλικού. Υπάρχουν πολλά και διαφορετικά είδη καρτών γραφικών, με πολλές και διαφορετικές δυνατότητες. Κάθε κάρτα γραφικών υλοποιεί τα επτά σημεία στα οποία έγινε αναφορά στην προηγούμενη ενότητα με διαφορετικό τρόπο. Είναι ο σχεδιαστής αναγκασμένος να κατασκευάσει διαφορετικές παραλλαγές του κόσμου του, μία για κάθε είδος κάρτας;

Κάτι τέτοιο είναι πρακτικά αδύνατο. Γίνεται αντιληπτή η αναγκαιότητα ύπαρξης εργαλείων ανάπτυξης ανεξάρτητων από το υλικό, τα οποία όμως να μπορούν να αξιοποιούν ταυτόχρονα κάθε είδος υλικού, παλαιότερου, σημερινού ή και μελλοντικού. Τέτοια εργαλεία υπάρχουν, λέγονται *graphical APIs (Application Programming Interfaces)* και τα πιο σημαντικά είναι η OpenGL και το DirectX. Αν και η φιλοσοφία τους είναι

διαφορετική και είναι μεταξύ τους ασύμβατα, αυτό που προσφέρουν είναι η μεσολάβηση μεταξύ του υλικού και του προγράμματος ανάπτυξης της εφαρμογής ή της τελικής εφαρμογής. Για να γίνει κάτι τέτοιο, έχει συμφωνηθεί από τους κατασκευαστές το υλικό να μπορεί να εκτελεί συγκεκριμένα και κοινά σύνολα εντολών, διασφαλίζοντας ότι μία εφαρμογή θα έχει περίπου τα ίδια αποτελέσματα σε όποιο υλικό και αν χρησιμοποιηθεί. Αυτό που τελικά διαφέρει είναι η μέθοδος υλοποίησης των εντολών εσωτερικά στο υλικό.

Ο σχεδιαστής αναπτύσσει την εφαρμογή του, εν μέρει αγνοώντας τις δυνατότητες του υλικού του χρήστη, καθώς δε γράφει πρόγραμμα για μία συγκεκριμένη κάρτα γραφικών, αλλά χρησιμοποιεί τις εντολές του API της επιλογής του. Οι εντολές αυτές εκτελούνται από το υλικό ανάλογα με τη μέθοδο που έχει επιλέξει ο κατασκευαστής της κάρτας γραφικών.

OpenGL

Η OpenGL (*Open Graphics Language*) αναπτύχθηκε αρχικά από τη Silicon Graphics και, όπως φαίνεται από το όνομά της, είναι “ανοικτή”, δηλαδή ο κώδικάς της είναι ελεύθερος, δίνοντας τη δυνατότητα σε όποιον θέλει να προχωρήσει σε προσθήκες ή τροποποιήσεις. Αποτελείται από 120 περίπου διαφορετικές εντολές, οι οποίες περιγράφουν τα αντικείμενα και τις διαδικασίες για την παραγωγή τρισδιάστατων αλληλεπιδραστικών εφαρμογών [93]. Ένα πολύ σοβαρό πλεονέκτημά της είναι ότι μπορεί να εκτελείται ανεξάρτητα από το λειτουργικό σύστημα (*cross platform*). Όμως, για να γίνει εφικτό αυτό, δεν περιλαμβάνει κανένα σύστημα ελέγχου παραθυρικών εφαρμογών, ούτε δίνεται η δυνατότητα να περιγραφούν πολύπλοκα μοντέλα. Αντίθετα, το κάθε μοντέλο φτιάχνεται από απλά γεωμετρικά στερεά, γραμμές και πολύγωνα.

Η διαδικασία δημιουργίας μίας τρισδιάστατης σκηνής με την OpenGL είναι η ακόλουθη: Αφού στηθεί η σκηνή με πολύγωνα, η OpenGL αναλαμβάνει να δημιουργήσει την αίσθηση του βάθους, θολώνοντας τα πιο μακρινά αντικείμενα, όπως ο φακός μίας φωτογραφικής μηχανής. Τα μοντέλα αποκτούν αρχικά στερεή υπόσταση με τη χρήση ενός μόνο χρώματος και ακολούθως φωτίζονται ανάλογα με τις πηγές φωτισμού που έχουν οριστεί. Στη συνέχεια τοποθετούνται στα αντικείμενα οι υφές και οι σκιάς. Η διαχείριση των σκιών είναι κάπως προβληματική, γιατί δεν υπάρχει συγκεκριμένη εντολή σκίασης, οπότε χρησιμοποιούνται διάφορες τεχνικές που παρακάμπτουν το πρόβλημα. Τέλος, τα κινούμενα αντικείμενα αποκτούν μία θολότητα που υποδηλώνει την κίνηση (*motion blur*).

Η δυνατότητα παρέμβασης στις εντολές της OpenGL, λόγω του ελεύθερου κώδικα, μπορεί να αποτελέσει πρόβλημα. Από τη μία επιτρέπει την άμεση εκμετάλλευση των

νέων δυνατοτήτων που παρέχει το υλικό, γιατί αρκεί να γραφούν εντολές που υλοποιούν κάτι τέτοιο. Από την άλλη, εξαιτίας των διαφορετικών προσεγγίσεων και των διαφορετικών δυνατοτήτων των καρτών γραφικών, μεγάλες εταιρείες έχουν προχωρήσει στη δημιουργία παραλλαγών της OpenGL (*OpenGL extensions*), με σκοπό την προώθηση των δικών τους προϊόντων. Αυτό έχει ως αποτέλεσμα οι διάφορες επεκτάσεις της OpenGL να είναι ασύμβατες μεταξύ τους.

DirectX

Το DirectX, αντίθετα με την OpenGL, δεν είναι κτήμα της παγκόσμιας κοινότητας, αλλά ανήκει στη Microsoft. Χρησιμοποιείται για τη δημιουργία παραθυρικών εφαρμογών πολυμέσων και παιχνιδιών, μονάχα σε περιβάλλον Windows. Εφαρμογές γραμμένες σε DirectX δεν εκτελούνται σε άλλο λειτουργικό σύστημα. Το DirectX προσφέρει υποστήριξη τρισδιάστατων γραφικών, ήχου, δικτύου και κίνησης. Επίσης περιλαμβάνει βιβλιοθήκες για την υποστήριξη περιφερειακών συσκευών [94]. Αιτία για τη δημιουργία του ήταν τα ίδια τα Windows. Τα Windows είναι λειτουργικό σύστημα που δεν επιτρέπει στις εφαρμογές την απευθείας πρόσβαση στο υλικό. Αυτή η παρεμβολή πολλών στρωμάτων λογισμικού μέχρι το υλικό έχει ως αποτέλεσμα την επιβράδυνση της εκτέλεσης των εφαρμογών. Ένα δεύτερο γεγονός είναι ότι τα Windows φτιάχτηκαν αρχικά με στόχο τις εφαρμογές γραφείου, οπότε τους έλειπαν ουσιαστικά στοιχεία για τον έλεγχο γραφικών και ήχου.

Το DirectX εμφανίστηκε για να εξαλείψει αυτά τα δύο προβλήματα, αλλά κυρίως για να δώσει κίνητρα στους κατασκευαστές λογισμικού να γράψουν εφαρμογές για Windows και να τα διαδώσουν ακόμα περισσότερο. Αποτελείται και αυτό από εντολές ομαδοποιημένες στα τα εξής τμήματα:

- Direct3D, για τη δημιουργία και φωτοσκίαση περίπλοκων τρισδιάστατων γραφικών. Στο Direct3D περιλαμβάνονται επίσης διάφορα εφέ φωτισμού, τα είδη του υλικού ενός αντικειμένου και οι σκιάσεις.
- DirectDraw, που επιτρέπει τη δημιουργία κινούμενων εικόνων.
- DirectInput, για τον έλεγχο των συσκευών εισόδου (πληκτρολόγιο, ποντίκι, κτλ.).
- DirectMusic, για τη σύνθεση και εκτέλεση μουσικών κομματιών.
- DirectSound, που επιτρέπει την πρόσβαση στην κάρτα ήχου, τη δημιουργία και εκτέλεση διαφόρων ήχων.
- DirectSetup, που επιτρέπει τη δημιουργία αρχείων εγκατάστασης μίας εφαρμογής. Στα αρχεία αυτά συμπεριλαμβάνονται και οι απαραίτητες βιβλιοθήκες του DirectX για την ομαλή εκτέλεση της εφαρμογής.
- DirectPlay, που δίνει τη δυνατότητα σε μία εφαρμογή να εκτελείται δικτυακά.

Με το DirectX ο κατασκευαστής μίας εφαρμογής μπορεί να είναι σίγουρος πως όλα τα στοιχεία της εφαρμογής του θα εκτελεστούν ακόμα και όταν το υλικό δεν έχει την ανάλογη δυνατότητα. Αυτό γιατί το DirectX αναλαμβάνει να προσομοιώσει μέσω λογισμικού ό,τι θα έπρεπε να κάνει το υλικό [94]. Το αρνητικό σημείο είναι ότι η ταχύτητα της προσομοίωσης είναι κατά πολύ μικρότερη από την ταχύτητα που θα παρείχε το υλικό. Ένα άλλο αρνητικό σημείο είναι ότι η ενσωμάτωση νέων εντολών που να καλύπτουν ή να ξεπερνούν τις εξελίξεις στο υλικό είναι υπόθεση της Microsoft. Αν και υπάρχει μία διαρκής συνεννόηση με τις κατασκευάστριες εταιρείες έτσι ώστε το DirectX να συμβαδίζει με τα σχέδιά τους, εμπλέκονται σε μεγάλο βαθμό διάφορες πολιτικές προώθησης προϊόντων και όχι οι πραγματικές τεχνολογικές εξελίξεις.

4.4 Προσδιορισμός των δομικών στοιχείων του λογισμικού κατασκευής εικονικών κόσμων

Τα δύο παραπάνω *API* λύνουν μέρος του προβλήματος κατασκευής ενός εικονικού κόσμου. Είναι χαμηλά στη βαθμίδα του λογισμικού, είναι πολύ κοντά στο υλικό. Ως εκ τούτου είναι αρκετά δύσκολος ο χειρισμός τους, γιατί απαιτείται πολύ καλή γνώση τους, άριστη γνώση προγραμματισμού, και κυρίως γράψιμο πολλών γραμμών κώδικα. Χρειάζονται προγράμματα που να μπορούν να υλοποιούν το ένα ή και τα δύο *API* και ταυτόχρονα να παρέχουν ένα φιλικό περιβάλλον εργασίας. Τα δομικά στοιχεία που θα πρέπει να περιλαμβάνονται σε κάθε πρόγραμμα κατασκευής εικονικών κόσμων είναι:

- Ο *renderer*
- Το τμήμα διαχείρισης ήχου
- Το τμήμα ελέγχου πλοήγησης
- Το εκτελέσιμο πρόγραμμα-player
- Το τμήμα των κινούμενων γραφικών
- Το τμήμα ελέγχου δικτυακής εκτέλεσης
- Η δυνατότητα προγραμματισμού-scripting

Renderer

Η ψυχή ενός προγράμματος κατασκευής εικονικών κόσμων είναι η μετατροπή ενός τρισδιάστατου μοντέλου που αποτελείται αρχικά μόνο από πολύγωνα, σ' ένα μοντέλο που έχει "ντυθεί" με κάποιο υλικό, έχει χρωματιστεί, έχει φωτιστεί και έχει σκιαστεί. Όλες αυτές τις εργασίες τις αναλαμβάνει ένα συγκεκριμένο τμήμα του προγράμματος που λέγεται *renderer* και στην ουσία είναι υπεύθυνο για την τελική εικόνα του κόσμου που κατασκευάζεται, γι' αυτό που εμφανίζεται στην οθόνη του υπολογιστή [95]. Ο τρόπος με τον οποίο το πρόγραμμα διαχειρίζεται τα πολύγωνα ενός μοντέλου (ορισμός, αποθήκευση, ανάκληση, σχεδιασμός) είναι η πρώτη σημαντική του εργασία.

Το επόμενο βήμα είναι ο καθορισμός των πολυγώνων που είναι ορατά σε κάθε δεδομένη στιγμή. Για παράδειγμα, δεν πρέπει να είναι ορατά τα πολύγωνα που βρίσκονται πίσω από έναν τοίχο ή πάνω από μία ορισμένη απόσταση, αλλά θα πρέπει να είναι

ορατά τα πολύγωνα πίσω από ένα παράθυρο. Στη συνέχεια ακολουθούν ο φωτισμός, οι σκιάσεις και οι υφές των αντικειμένων.

Ο *renderer* είναι το πιο επιρρεπές σε μεταβολές τμήμα ενός προγράμματος κατασκευής εικονικών κόσμων, γιατί εξαρτάται άμεσα από τις εξελίξεις στο υλικό. Λόγω της εξαιρετικής του σημασίας και επειδή καταλαμβάνει πάνω από το 50% των υπολογισμών που απαιτούνται σ' έναν εικονικό κόσμο, θα πρέπει να υπάρχει η δυνατότητα να αντικαθίσταται χωρίς να τροποποιείται σε σημαντικό βαθμό το σύνολο του προγράμματος κατασκευής.

Κινούμενα γραφικά-Animation

Η προσθήκη κίνησης στα γραφικά είναι αυτό που ζωντανεύει τον εικονικό κόσμο. Περιλαμβάνει όλο το φάσμα κινήσεων, από τη μετακίνηση ενός κουτιού σε ευθεία γραμμή μέχρι τις πολυσύνθετες κινήσεις που κάνει ένα ανθρώπινο πρόσωπο όταν μιλά. Οι μέθοδοι με τις οποίες αντιμετωπίζονται οι κινήσεις διαφέρουν ανάλογα με το είδος της κίνησης, αλλά θα πρέπει να μπορούν να συνδυάζονται πάνω στο ίδιο αντικείμενο [96].

Για παράδειγμα, η μετακίνηση ενός αυτοκινήτου πάνω σε μία νοητή τεθλασμένη γραμμή μπορεί να υλοποιηθεί με απλό *path animation* (ο ορισμός διαδοχικών σημείων της τροχιάς που ακολουθεί ένα κινούμενο αντικείμενο). Το στρίψιμο των μπροστινών τροχών του είναι μία άλλη μορφή κίνησης, στην οποία ορίζονται στο μοντέλο “κόκαλα και αρθρώσεις” (*skeletal animation*), που κινούνται με συγκεκριμένο τρόπο, συμπαρασύροντας τα πολύγωνα του μοντέλου που βρίσκονται πάνω από τα κόκαλα.

Μία τρίτη μορφή κίνησης είναι η μετακίνηση ενός πλέγματος πολυγώνων από τη μία θέση στην άλλη ανάλογα με τον αριθμό των καρέ που έχει η κίνηση (*mesh animation*). Η μέθοδος που θα ακολουθηθεί εξαρτάται από πολλούς παράγοντες και είναι αυτονόητο ότι ένα πρόγραμμα κατασκευής εικονικών κόσμων θα πρέπει να παρέχει τη δυνατότητα χρήσης όλων των μεθόδων.

Ήχος

Ο ήχος και η μουσική προσφέρουν αρκετά στην ποιότητα μίας εφαρμογής. Η μουσική συμβάλλει στη δημιουργία κατάλληλης ατμόσφαιρας, ενώ οι διάφοροι ήχοι συμβάλλουν στην αληθοφάνεια, στη ρεαλιστική απόδοση των αντικειμένων. Σύμφωνα με τα σημερινά δεδομένα, τα περισσότερα υπολογιστικά συστήματα περιλαμβάνουν τέσσερα ως έξι ηχεία, που σε συνεργασία με τις κάρτες ήχου δίνουν τη δυνατότητα παραγωγής ήχων που προέρχονται από συγκεκριμένες κατευθύνσεις και υλοποιούν τα πρότυπα Dolby Digital και DTS. Το θέμα του ήχου δεν παρουσιάζει τον ίδιο βαθμό προβλημάτων και περιορισμών όπως τα γραφικά, γιατί απαιτείται λιγότερη υπολογι-

στική ισχύς και το σύνολο των κατασκευαστριών εταιρειών ακολουθούν τα ίδια πρότυπα. Το ζητούμενο λοιπόν από ένα πρόγραμμα ανάπτυξης εικονικών κόσμων είναι να μπορεί να διαχειρίζεται ήχο με πολλά κανάλια, διάφορα ηχητικά εφέ και μουσική [87].

Δικτυακή εκτέλεση

Το συγκεκριμένο θέμα μπορεί να αφορά δύο διαφορετικές φιλοσοφίες. Η μία είναι ότι υπάρχει η δυνατότητα να είναι ο χρήστης σ' ένα σημείο και η εφαρμογή να βρίσκεται εγκατεστημένη σ' ένα άλλο. Κάτι τέτοιο μπορεί να συμβαίνει στην περίπτωση που από τον ηλεκτρονικό υπολογιστή του χρήστη δεν υπάρχει η απαίτηση να εκτελεί όλες τις εργασίες και υπολογισμούς, αλλά απλώς να παρουσιάζει το τελικό αποτέλεσμα.

Κυρίως όμως εννοείται η δυνατότητα να συνυπάρχουν στην ίδια εφαρμογή περισσότεροι του ενός χρήστες. Σ' αυτή την περίπτωση αυξάνεται κατά πολύ η διαδραστικότητα της εφαρμογής, επιτρέποντας αλληλεπιδράσεις και συνεργασίες μεταξύ των χρηστών. Αυτό έχει άμεση επίπτωση στο ρεαλισμό της εφαρμογής, γιατί υλοποιεί κάτι που ισχύει και στον πραγματικό κόσμο. Συνεπώς το πρόγραμμα κατασκευής πρέπει να υιοθετεί και να διαχειρίζεται τα σημαντικότερα δικτυακά πρωτόκολλα επικοινωνίας, με κύριο το *TCP/IP (Transmission Control Protocol/Internet Protocol)*.

Πλοήγηση

Η πλοήγηση αφορά τον τρόπο μετακίνησης του χρήστη μέσα στον εικονικό κόσμο, αλλά και τις περιφερειακές συσκευές που χρησιμοποιούνται για να γίνει κάτι τέτοιο [97]. Για την πλοήγηση χρησιμοποιείται μία φανταστική κάμερα. Αν η κάμερα είναι πάνω στο μοντέλο που αναπαριστά το χρήστη και μάλιστα στο ύψος των ματιών του, τότε αυτός βλέπει τον εικονικό κόσμο με προοπτική “πρώτου προσώπου”, τον βλέπει όπως βλέπουμε και τον πραγματικό κόσμο. Αν είναι πίσω του, ο χρήστης βλέπει τον εικονικό κόσμο με προοπτική “τρίτου προσώπου”, βλέπει δηλαδή και την αναπαράσταση του εαυτού του.

Η κάμερα μπορεί να είναι συνδεδεμένη και με άλλα αντικείμενα, οπότε ο χρήστης βλέπει τον κόσμο από τη σκοπιά των αντικειμένων αυτών. Ένας απλός τρόπος πλοήγησης είναι να χρησιμοποιείται το πληκτρολόγιο ή το ποντίκι για να μετακινείται η φανταστική κάμερα. Όμως στην Ε.Π. προτιμάται η χρήση άλλων συσκευών, που προσομοιώνουν το φυσικό τρόπο μετακίνησης. Συνεπώς το πρόγραμμα κατασκευής θα πρέπει να μπορεί να διαχειρίζεται είσοδο και έξοδο δεδομένων από και προς τις συσκευές αυτές.

Προγραμματισμός-Scripting

Ένα πρόγραμμα κατασκευής εικονικών κόσμων έχει συγκεκριμένες δυνατότητες. Τι συμβαίνει στην περίπτωση που ο σχεδιαστής θέλει να υλοποιήσει κάτι που δεν έχει προβλεφθεί από το πρόγραμμα κατασκευής του εικονικού κόσμου; Στην απλούστερη περίπτωση, ο σχεδιαστής μεταβάλλει μία σειρά παραμέτρων και αριθμητικών τιμών σε έτοιμες καρτέλες προεπιλογών. Για παράδειγμα, μπορεί να αλλάξει την τιμή της βαρύτητας που επικρατεί στον εικονικό κόσμο ή να αλλάξει τον τρόπο με τον οποίο αναπηδούν τα αντικείμενα όταν συγκρούονται μεταξύ τους.

Για να αντιμετωπίσει πιο σύνθετες καταστάσεις, θα πρέπει να του δίνεται η δυνατότητα να γράψει εντολές προς το πρόγραμμα (*scripts*) που να υλοποιούν αυτό που έχει κατά νου [98]. Οι εντολές αυτές έχουν τη μορφή και τη σύνταξη μίας γλώσσας προγραμματισμού και ουσιαστικά επεμβαίνουν στην καρδιά της εφαρμογής. Ένα παράδειγμα χρήσης *script* είναι όταν προσδίδονται στοιχεία “νοημοσύνης” σε κάποια αντικείμενα, για να φαίνεται ότι συμπεριφέρονται έξυπνα. Ένας εικονικός άνθρωπος μπορεί να απαντά σε συγκεκριμένες ερωτήσεις με βάση κάποιες ερωτήσεις και απαντήσεις που έχουν γραφτεί εκ των προτέρων σ' ένα *script*.

Εκτέλεσιμο πρόγραμμα-Player

Τα παραπάνω δομικά στοιχεία αφορούν αποκλειστικά τη μεριά του σχεδιαστή-κατασκευαστή ενός εικονικού κόσμου. Δεν υπάρχει η απαίτηση από τον τελικό χρήστη να έχει ένα πρόγραμμα κατασκευής μόνο και μόνο για να εκτελέσει την εφαρμογή, για να τη χρησιμοποιήσει. Από την άλλη πλευρά, είναι χρήσιμο στο σχεδιαστή να μπορεί να βλέπει, όσο εξελίσσει την εφαρμογή του, ποιο είναι το αποτέλεσμα. Χρειάζεται λοιπόν ένα ξεχωριστό κομμάτι, ένας *player* της εφαρμογής. Ο ρόλος του είναι ακριβώς ό,τι υποδηλώνει το όνομά του, απλώς “παίζει” την εφαρμογή, δεν την τροποποιεί. Οι *players* έχουν διάφορες μορφές, συνήθως αποτελούν τμήμα του προγράμματος κατασκευής και περιλαμβάνουν την εφαρμογή μαζί με όλα τα απαραίτητα αρχεία και βιβλιοθήκες για την εκτέλεσή της.

4.5 Κατηγορίες λογισμικού κατασκευής εικονικών κόσμων

Αφού αναλύθηκαν οι απαιτήσεις από το υλικό και το λογισμικό, πρέπει να εξεταστούν οι διάφορες κατηγορίες λογισμικού κατασκευής εικονικών κόσμων που υπάρχουν, τα πλεονεκτήματα και τα μειονεκτήματα της κάθε κατηγορίας. Διακρίνονται τέσσερις τέτοιες κατηγορίες: α) η προγραμματιστική προσέγγιση, β) η δημιουργία πανοράματος 360°, γ) η δημιουργία κόσμων προκαθορισμένης διαδρομής και δ) η χρήση ολοκληρωμένων πακέτων δημιουργίας εφαρμογών Ε.Π. [99]. Επιπλέον, ο σχεδιαστής πρέπει να χρησιμοποιήσει μία σειρά βοηθητικών προγραμμάτων για την κατασκευή υφών, μοντέλων, ήχων, κτλ.

Προγραμματιστική προσέγγιση

Η κατασκευή ενός εικονικού κόσμου με τη χρήση μίας γλώσσας προγραμματισμού είναι ο πιο δύσκολος τρόπος να προσεγγίσει κάποιος το θέμα. Ο σχεδιαστής έχει τον απόλυτο έλεγχο για ό,τι συμβαίνει, προγραμματίζοντας τις συμπεριφορές των δομικών στοιχείων του κόσμου του, αλλά απαιτείται η άριστη γνώση της γλώσσας που θα επιλέξει. Συνήθως συνεργάζονται ομάδες προγραμματιστών και το κάθε μέλος αναλαμβάνει να ολοκληρώσει ένα συγκεκριμένο κομμάτι. Απαιτούνται συνεχείς έλεγχοι για να διαπιστωθεί η ομαλή λειτουργία του κόσμου, να γίνει η διόρθωση των λαθών και να βελτιωθεί ο κώδικας.

Βασικό πλεονέκτημα είναι η δυνατότητα εκτέλεσης του προγράμματος ανεξάρτητα από το λειτουργικό σύστημα που χρησιμοποιεί ο τελικός χρήστης. Αυτό αποδεδειγμένο το σχεδιαστή από την υποχρέωση να κατασκευάσει διαφορετικές εκδόσεις του ίδιου κόσμου, μία για καθένα από τα κύρια λειτουργικά συστήματα (Windows, Linux, Mac OS, κτλ.). Δύο είναι οι γλώσσες που χρησιμοποιούνται συνήθως:

Η C++, μία πολύ ισχυρή προγραμματιστική γλώσσα με πολλές δυνατότητες, χρήσιμη για την κατασκευή οποιουδήποτε είδους προγράμματος και όχι συγκεκριμένα για την Ε.Π.

Η δεύτερη είναι η VRML (*Virtual Reality Modeling Language*), μία γλώσσα που αναπτύχθηκε από την κοινή προσπάθεια πολλών προγραμματιστών, ειδικά για την Ε.Π. [100]. Παρά τις καλές τους προθέσεις, η γλώσσα αυτή δεν έχει πετύχει μέχρι στιγμής να πραγματοποιήσει όσα υποσχόταν [101]. Κύριος στόχος της είναι η δυνατότητα εκτέλεσης στο Διαδίκτυο όσων εφαρμογών αναπτύσσονται με αυτή. Εξαιτίας όμως της περιορισμένης ταχύτητας σύνδεσης της πλειοψηφίας των χρηστών, τα μοντέλα έχουν σημαντικότερους περιορισμούς στον αριθμό των πολυγώνων, στις υφές και γενικά στις λεπτομέρειές τους. Αυτό είναι ήδη ένα πολύ σημαντικό μειονέκτημα για το ρεαλισμό του κόσμου.

Η ταχύτητα σύνδεσης είναι επίσης η αιτία για τον πολύ χαμηλό αριθμό των καρέ/δευτερόλεπτο και για τις διακοπές κατά τη διάρκεια της πλοήγησης. Τέλος, για την εκτέλεση της εφαρμογής και την πλοήγηση χρησιμοποιείται ένας ειδικός *φύλλομετρητής* (VRML browser), που αφήνει και αυτός μία πολύ χαμηλή αίσθηση ρεαλισμού.

Πανοράματα 360°

Η γενική ιδέα σε αυτά τα προγράμματα είναι ότι δίνουν τη δυνατότητα να γίνει συρραφή από φωτογραφίες κατά τέτοιο τρόπο, ώστε να καλύπτουν έναν πλήρη κύκλο (ει-

κόνα 4.4). Το γεγονός ότι χρησιμοποιούνται φωτογραφίες επιτρέπει υψηλό βαθμό ρεαλισμού. Εδώ όμως σταματούν τα όποια θετικά στοιχεία αυτής της μεθόδου.

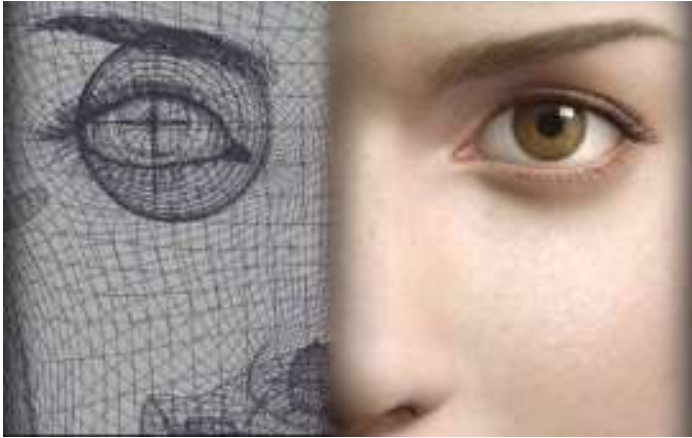


Εικόνα 4.4 Πανόραμα 360°

Ο εικονικός κόσμος, πρώτα απ' όλα, δεν είναι τρισδιάστατος. Ο χρήστης το μόνο που μπορεί να κάνει είναι να περιστρέφεται. Δεν υπάρχει αλληλεπίδραση, δεν υπάρχει κίνηση, δεν υπάρχει εξερεύνηση του κόσμου [99]. Όταν μάλιστα χρειάζεται η αναπαράσταση ενός φανταστικού κόσμου, πρέπει να χρησιμοποιηθούν άλλα προγράμματα για να τον σχεδιάσουν σε πολύ υψηλή ανάλυση, διαδικασία χρονοβόρα που δε δικαιολογεί το τελικό αποτέλεσμα, τη χρήση δηλαδή του παραγόμενου υλικού απλώς για ένα πανόραμα.

Προγράμματα προεπιλεγμένης διαδρομής ή ταινίες

Οι εικόνες 4.3 και 4.5 προέρχονται από μία ταινία (*Final Fantasy: The spirits within*), στην οποία το μόνο πραγματικό στοιχείο ήταν οι φωνές των ηθοποιών. Όλα τα υπόλοιπα, τοπία, κτίρια, ηθοποιοί, φυτά, ζώα, κτλ, ήταν μοντέλα εξαιρετικά υψηλής ανάλυσης. Για να κατασκευαστούν τα μοντέλα χρειάστηκαν 200 ηλεκτρονικοί υπολογιστές. Για να περαστούν οι υφές, οι φωτοσκιάσεις και οι κινήσεις, χρειάστηκε να δουλεύουν ταυτόχρονα πάνω από 1000 ειδικής κατασκευής ηλεκτρονικοί υπολογιστές, μ' ένα σύνολο 1500 επεξεργαστών [102]. Όλα αυτά και δεκαοκτώ μήνες σκληρής δουλειάς απαιτήθηκαν για τη δημιουργία μίας κινηματογραφικής ταινίας 140 λεπτών. Το αποτέλεσμα, όσον αφορά την ρεαλιστική απόδοση των χαρακτήρων και του κόσμου γενικότερα, ήταν κοντά στην τελειότητα.



Εικόνα 4.5 Από τα πολύγωνα στο τελικό αποτέλεσμα

Το παράδειγμα της παραπάνω ταινίας αποτελεί μία ακραία περίπτωση. Υπάρχουν λιγότερο απαιτητικά προγράμματα που επιτρέπουν στο σχεδιαστή να κατασκευάσει κόσμους με προκαθορισμένη διαδρομή και πλοκή. Το τελικό προϊόν, ο εικονικός κόσμος, δεν εκτελείται σε πραγματικό χρόνο, αλλά είναι κατά κάποιον τρόπο “μαγνητοσκοπημένος”, είναι μία ταινία με πολλές σκηνές συγκροτημένες σε ενιαίο σύνολο. Έτσι, οι περιορισμοί των καρτών γραφικών που αναλύθηκαν σε προηγούμενη ενότητα δεν ισχύουν. Ο σχεδιαστής μπορεί να κατασκευάσει πολλά κομμάτια μικρής διάρκειας που, ενώ έχουν μεγάλη ανάλυση, φωτοσκιάσεις, κτλ., λόγω του μεγέθους τους δεν εξαντλούν τις δυνατότητες του υλικού.

Ο τελικός χρήστης αρκεί να έχει έναν ηλεκτρονικό υπολογιστή που μπορεί να εκτελεί αρχεία βίντεο, πράγμα που ισχύει για το σύνολο των σημερινών υπολογιστικών συστημάτων. Όμως ο χρήστης απλώς παρακολουθεί, δεν επεμβαίνει, δεν αλληλεπιδρά με τα αντικείμενα, δεν εξερευνά τον κόσμο. Με άλλα λόγια, ο ρόλος του είναι παθητικός και όχι ενεργητικός. Η ουσία όμως της Ε.Π. βρίσκεται στην ενεργό συμμετοχή του χρήστη.

Ολοκληρωμένα πακέτα δημιουργίας εικονικών κόσμων

Από τις κατηγορίες λογισμικού που αναλύθηκαν, την πιο ικανοποιητική λύση δίνει ο προγραμματισμός σε C++. Όμως είναι η λιγότερο εύχρηστη λύση. Έτσι, υπάρχουν αρκετά προγράμματα που παρέχουν στο σχεδιαστή ένα παραθυρικό, εργονομικό και εύχρηστο περιβάλλον εργασίας, ενώ στο παρασκήνιο ό,τι κατασκευάζεται μεταφράζεται σε C++ ή σε κάποιες παραλλαγές της (ενδεικτικά [103, 104, 105]). Τα πλεονεκτήματα μίας τέτοιας προσέγγισης είναι προφανή. Χρησιμοποιείται μία ισχυρή γλώσσα και ταυτόχρονα δεν είναι απαραίτητη η καλή γνώση της.

Ο χρόνος και το κόστος ανάπτυξης του εικονικού κόσμου μειώνονται ακόμα περισσότερο με τη χρήση των έτοιμων βιβλιοθηκών αντικειμένων. Οι ιδιότητες και η συμπεριφορά των αντικειμένων καθορίζονται και αυτές από έτοιμες βιβλιοθήκες. Όταν ο σχεδιαστής τοποθετεί ένα αντικείμενο στον εικονικό κόσμο και του δίνει ορισμένες ιδιότητες, το πρόγραμμα αναλαμβάνει να γράψει τον απαραίτητο κώδικα. Αν ο σχεδιαστής θελήσει να υλοποιήσει κάτι που δεν έχει προβλεφθεί, τότε είναι δυνατό να γράψει εκείνος τον απαραίτητο κώδικα (*script*), επεμβαίνοντας στο συνολικό κώδικα της εφαρμογής.

Υπάρχουν όμως σοβαρά μειονεκτήματα από τη χρήση έτοιμων προγραμμάτων, με κυριότερο το κόστος απόκτησής τους, το οποίο –σε κάποιες περιπτώσεις– ανέρχεται σε αρκετές χιλιάδες ευρώ. Επίσης, κανένα δε συγκεντρώνει το σύνολο των χαρακτηριστικών που θα επιθυμούσε ένας σχεδιαστής. Άλλα δίνουν βαρύτητα στην κίνηση, άλλα στην υψηλή ποιότητα των γραφικών, άλλα στη δικτυακή εκτέλεση, ενώ παράγουν τύπους αρχείων που είναι ασύμβατοι μεταξύ τους.

Έτσι, η επεξεργασία ενός κόσμου που κατασκευάστηκε μ' ένα πρόγραμμα δεν είναι δυνατή από ένα δεύτερο. Μία λύση είναι να εξαχθεί ο κόσμος σε *VRML* από το ένα πρόγραμμα και να εισαχθεί με αυτή τη μορφή στο άλλο. Έτσι όμως χάνονται όλα τα στοιχεία του κόσμου που δεν προβλέπονται από τη *VRML* και όλα τα στοιχεία που ενσωμάτωσε το πρώτο πρόγραμμα στην εφαρμογή και που είναι ασύμβατα με το δεύτερο. Το τελικό αποτέλεσμα είναι η ποιότητα του κόσμου να είναι σημαντικά υποβαθμισμένη.

Το λογισμικό γενικά παρακολουθεί τις εξελίξεις στο υλικό με κάποια χρονική καθυστέρηση. Το να ενσωματωθούν καινούργιες δυνατότητες στο λογισμικό, ύστερα από σοβαρές εξελίξεις στο υλικό, προϋποθέτει αρκετή εργασία. Οι εταιρείες κατασκευής προγραμμάτων προτιμούν να περιμένουν τη συσσώρευση αρκετών αλλαγών πριν περάσουν σε αναβαθμισμένες εκδόσεις των προγραμμάτων τους. Έτσι, ο σχεδιαστής που θέλει οπωσδήποτε να χρησιμοποιήσει στην εφαρμογή του μία καινούργια δυνατότητα που του παρέχει το υλικό, αναγκαστικά πρέπει να περιμένει.

Αυτό οδηγεί επίσης και σ' ένα ακόμα σημαντικό θέμα, της υποστήριξης των προγραμμάτων κατασκευής εικονικών κόσμων. Η κρίση της περιόδου 1999-2001 οδήγησε πολλές από τις εταιρείες κατασκευής τους σε κλείσιμο ή σε αλλαγή αντικειμένου εργασίας. Το αποτέλεσμα ήταν η σταδιακή απαξίωση των προγραμμάτων λόγω έλλειψης υποστήριξης και λόγω έλλειψης ανανεωμένων εκδόσεων.

Τέλος, πρέπει να επισημανθεί ότι οι παραγόμενοι κόσμοι από ένα έτοιμο πακέτο δεσμεύονται για την εκτέλεσή τους από το λειτουργικό σύστημα που χρησιμοποιείται και από το πρόγραμμα κατασκευής. Με άλλα λόγια, ο τελικός χρήστης, για να εκτελέσει την εφαρμογή, χρειάζεται να έχει ένα συγκεκριμένο λειτουργικό σύστημα και ένα ειδικό μέρος του αρχικού προγράμματος, που αναλαμβάνει την εκτέλεση της εφαρμογής, έναν *player*. Οι *players* παρέχονται συνήθως δωρεάν, ενώ υπόκεινται και αυτοί στους κινδύνους της έλλειψης υποστήριξης και της απαξίωσης.

Βοηθητικά προγράμματα

Τα βοηθητικά προγράμματα δεν ανήκουν σε μία συγκεκριμένη κατηγορία, αλλά αντίθετα είναι μία ετερόκλητη συλλογή προγραμμάτων, με το καθένα να εξυπηρετεί ένα συγκεκριμένο σκοπό. Όσο πλήρεις και αν είναι οι βιβλιοθήκες έτοιμων αντικειμένων, ο σχεδιαστής αρκετά συχνά θα χρειαστεί να τροποποιήσει ή να κατασκευάσει εκ του μηδενός ένα τρισδιάστατο μοντέλο, έναν ήχο, μία εικόνα ή ένα γραφικό. Ενσωματωμένα υποπρογράμματα στο κυρίως πρόγραμμα κατασκευής εικονικών κόσμων τού επιτρέπουν, μέχρι ενός σημείου, να κάνει αυτή τη δουλειά.

Τις περισσότερες φορές θα χρειαστεί να καταφύγει σε εξωτερικά προγράμματα που του δίνουν περισσότερες δυνατότητες. Το κόστος αυτών των προγραμμάτων είναι ιδιαίτερα σημαντικό, ειδικά για τα προγράμματα κατασκευής τρισδιάστατων μοντέλων (3D Studio Max, Lightwave, Maya), που αθροιζόμενο με το κόστος ενός προγράμματος κατασκευής εικονικών κόσμων ανεβάζουν σημαντικά το συνολικό κόστος. Τα προγράμματα κατασκευής εικονικών κόσμων περιλαμβάνουν μηχανισμούς και φίλτρα εισαγωγής διαφόρων τύπων αρχείων από εξωτερικά προγράμματα και αναλαμβάνουν τη διαμόρφωσή τους με τέτοιο τρόπο, ώστε να ενσωματώνονται χωρίς προβλήματα στην κύρια εφαρμογή.

4.6 Άλλα θέματα υλικού και λογισμικού

Η εξερεύνηση και η αλληλεπίδραση χρειάζεται να αναλυθούν από τεχνικής άποψης. Αντίθετα με τον πραγματικό κόσμο, ένας εικονικός κόσμος που στηρίζεται σε τρισδιάστατα γραφικά δεν είναι πάντοτε πλήρως εξερευνησίμος και πλήρως αλληλεπιδραστικός. Για παράδειγμα, σε μία πόλη στον πραγματικό κόσμο μπορεί κάποιος να επισκεφθεί όλες τις οικοδομές, τους δρόμους, τα μαγαζιά και γενικά τις περιοχές της, και αυτό συνιστά την έννοια της *εξερεύνησης*.

Η έννοια της *αλληλεπίδρασης* είναι ότι μπορεί να αγγίξει, να μετακινήσει και να περριεργαστεί όλα τα αντικείμενα που περιέχονται σ' αυτή την πόλη. Σ' έναν τρισδιάστατο κόσμο κάποιες οικοδομές, δρόμοι ή και περιοχές ακόμα, δεν είναι προσιτές στο χρή-

στη. Επίσης δεν μπορεί να αλληλεπιδράσει με όλα τα αντικείμενα, παρά μόνο με αυτά στα οποία ο σχεδιαστής του εικονικού κόσμου έχει δώσει αυτή τη δυνατότητα.

Αυτοί οι περιορισμοί, που δεν είναι αξεπέραστοι, υπάρχουν για λόγους που σχετίζονται με τα γραφικά και το λογισμικό. Ο εξαιρετικά μεγάλος αριθμός πολυγώνων θα ξεπερνούσε τις δυνατότητες της κάρτας γραφικών. Έτσι, πολλές φορές ο αρχικός κόσμος κομματιάζεται σε μικρότερους. Στο παραπάνω παράδειγμα, περιοχές της πόλης μπορούν να αποτελέσουν από ένα ξεχωριστό υποσύνολο του αρχικού κόσμου. Όταν ο χρήστης επισκέπτεται μία συγκεκριμένη περιοχή, οι υπόλοιπες δεν είναι απαραίτητο να είναι σχεδιασμένες. Μάλιστα, αυτή είναι μία από τις πιο συνηθισμένες τεχνικές απαλλαγής της κάρτας γραφικών από σημαντικό μέρος του υπολογιστικού της φορτίου [91].

Μία δεύτερη παράμετρος που επιδρά στην εξερεύνηση είναι ότι για την κατασκευή ακόμα και ενός απλού εικονικού κόσμου απαιτείται εξαιρετικά επίπονη και χρονοβόρα εργασία. Αν κάθε σημείο της εικονική πόλης του παραδείγματος ήταν προσβάσιμο από το χρήστη, θα έπρεπε να κατασκευαστεί με λεπτομέρεια κάθε εσωτερικός και εξωτερικός χώρος και να περιλαμβάνεται ένας τεράστιος αριθμός από μοντέλα αντικειμένων που βρίσκονται στους χώρους αυτούς. Κάτι τέτοιο θα απαιτούσε εργασία ετών. Έτσι, οι σχεδιαστές εικονικών κόσμων ανάλογα με την εφαρμογή καθορίζουν και το βαθμό της εξερεύνησης των χώρων από το χρήστη, βάζοντας “σύνορα” στον εικονικό κόσμο.

Τα ίδια ισχύουν και για την *αλληλεπίδραση*. Τα αντικείμενα με τα οποία μπορεί να αλληλεπιδράσει ο χρήστης πρέπει να έχουν κάποια *scripts* που να κανονίζουν τη “συμπεριφορά” τους όταν τα χρησιμοποιεί ο χρήστης και να προβλέπουν κάθε πιθανή και απίθανη ενέργειά του. Θα πρέπει επίσης να καθορίζουν τη σχέση των αντικειμένων μεταξύ τους και πώς αυτά αλληλεπιδρούν. Κάτι τέτοιο σε μεγάλη κλίμακα, όπως και στην περίπτωση της εξερεύνησης, θα απαιτούσε εξαιρετικά δύσκολη εργασία. Επίσης, επειδή τα *scripts* χρησιμοποιούν για την εκτέλεσή τους τον επεξεργαστή του υπολογιστή, έχουν επίπτωση στην ταχύτητα εκτέλεσης της εφαρμογής. Και πάλι ο σχεδιαστής περιορίζει την αλληλεπίδραση σε αντικείμενα-κλειδιά του εικονικού κόσμου.

4.7 Καθορισμός των τεχνικών προδιαγραφών για την επιλογή ενός προγράμματος κατασκευής εφαρμογών Εικονικής Πραγματικότητας

Από την παρουσίαση των διαφόρων κατηγοριών λογισμικού, αυτό που προκύπτει είναι ότι ο σχεδιαστής πρέπει να επιλέξει με ιδιαίτερη προσοχή το είδος του προγράμματος που θα χρησιμοποιήσει για την ανάπτυξη της εφαρμογής του. Αν πρόκειται για αλληλεπιδραστικές εφαρμογές Ε.Π., η χρήση ολοκληρωμένων πακέτων είναι μονό-

δρομος, γιατί υπερτερούν στον τομέα της ευχρηστίας σε σύγκριση με τις γλώσσες προγραμματισμού. Για την επιλογή όμως ενός τέτοιου πακέτου πρέπει να σταθμίσουν πολλές παράμετροι που σχετίζονται με τις δυνατότητες που παρέχονται από το καθένα, και αυτό είναι ιδιαίτερα δύσκολη υπόθεση.

Αυτό που δεν είναι καθόλου εξασφαλισμένο είναι η βιωσιμότητα, η αντοχή στο χρόνο, της εφαρμογής που θα αναπτυχθεί. Η εφαρμογή αντανakλά τις δυνατότητες του υλικού και του λογισμικού τη στιγμή που κατασκευάστηκε. Μέχρι πριν από λίγα χρόνια το υλικό επέτρεπε τη χρήση μοντέλων με μικρό αριθμό πολυγώνων και ο ρεαλισμός των εφαρμογών ήταν αρκετά μικρός. Σήμερα μπορούν να χρησιμοποιηθούν μοντέλα με μεγαλύτερο αριθμό πολυγώνων, ο ρεαλισμός είναι σαφώς μεγαλύτερος, αλλά για να ενσωματωθούν οι νέες δυνατότητες ο εικονικός κόσμος πρέπει να κατασκευαστεί από την αρχή. Γίνεται κατανοητός ο κόπος και το χρήμα που πρέπει να δαπανηθούν για κάτι τέτοιο.

Ιδιαίτερη αντοχή στο χρόνο έχουν τα βοηθητικά προγράμματα. Αν και λειτουργούν βοηθητικά για τις ανάγκες των εφαρμογών της Ε.Π., είναι ιδιαίτερα στιβαρές εφαρμογές που χρησιμοποιούνται σε πολλούς τομείς. Για παράδειγμα, τα προγράμματα κατασκευής τρισδιάστατων μοντέλων χρησιμοποιούνται από το σύνολο σχεδόν των θετικών επιστημών και των επαγγελματιών τους (αρχιτέκτονες, χημικούς, φυσικούς, μαθηματικούς, μηχανικούς, κτλ.). Υπάρχουν για πολλά χρόνια στην αγορά, εξελίσσονται και αναβαθμίζονται με ικανοποιητικούς ρυθμούς και γενικά καλύπτουν τις ανάγκες του αγοραστικού κοινού. Αντίστοιχης αντοχής προγράμματα για την Ε.Π. είναι ελάχιστα.

Με αυτά τα δεδομένα, αποκτά ιδιαίτερη σημασία η επιλογή του καταλληλότερου προγράμματος κατασκευής εικονικών κόσμων. Για να γίνει αυτό, πρέπει πρώτα να οριστεί ένα σύνολο συγκεκριμένων προδιαγραφών που θα πρέπει να ικανοποιεί ένα πρόγραμμα κατασκευής εικονικών κόσμων. Με βάση τις προδιαγραφές μπορεί να επιλεγεί τελικά ένα από τα διαθέσιμα προγράμματα.

Ευχρηστία

Ένα πρώτο στοιχείο που πρέπει να προσεχθεί κατά την επιλογή ενός προγράμματος κατασκευής εικονικών κόσμων είναι η ευχρηστία του. Ο όρος “ευχρηστία” είναι βέβαια δύσκολο να οριστεί, είναι υποκειμενικός και εξαρτάται σε μεγάλο βαθμό από την εμπειρία αυτού που θα χρησιμοποιήσει την εφαρμογή. Σε γενικές γραμμές πάντως αναφέρεται στο περιβάλλον εργασίας που παρουσιάζεται στο σχεδιαστή. Για παράδειγμα, όταν χρησιμοποιείται μία γλώσσα προγραμματισμού, το περιβάλλον εργασίας είναι ένας χώρος στον οποίο ο σχεδιαστής γράφει τον κώδικα της εφαρμογής.

Ως περιβάλλον είναι πολύ αφηρημένο, δε δίνει τη δυνατότητα στο σχεδιαστή να έχει άμεση αντίληψη του τι κατασκευάζει. Πρέπει πρώτα να διορθώσει τυχόν λάθη, να μετατρέψει τον κώδικα σε εκτελέσιμο αρχείο, να δει το αποτέλεσμα και στη συνέχεια να προχωρήσει σε βελτιώσεις ή στην κατασκευή ενός άλλου κομματιού της εφαρμογής, επαναλαμβάνοντας την όλη διαδικασία. Ακόμα και αν υποθεθεί ότι ο σχεδιαστής είναι άριστος γνώστης της γλώσσας προγραμματισμού που χρησιμοποιεί, ένα τέτοιο περιβάλλον δεν του παρέχει τη δυνατότητα άμεσου και δυναμικού ελέγχου της εφαρμογής του.

Αυτό ακριβώς είναι το πλεονέκτημα των πακέτων κατασκευής εικονικών κόσμων. Το περιβάλλον εργασίας σε αυτά είναι συνήθως μία επιφάνεια πάνω στην οποία ο σχεδιαστής, τοποθετώντας διάφορα αντικείμενα, “χτίζει” σταδιακά τον εικονικό κόσμο. Ορισμένες φορές αυτή η επιφάνεια παρουσιάζεται από διαφορετικές οπτικές γωνίες (κάτοψη, πλάγια όψη κατά πλάτος, πλάγια όψη κατά μήκος και πανοραμική). Με τον τρόπο αυτό έχει άμεση εποπτεία όλων των παραμέτρων της εφαρμογής και βλέπει το αποτέλεσμα χωρίς περίπλοκες και χρονοβόρες διαδικασίες.

Χρησιμοποιώντας εργαλειοθήκες προσδίδει ιδιότητες και κίνηση στα αντικείμενα, και πάλι ελέγχοντας άμεσα το αποτέλεσμα. Η ευχρηστία σ' αυτή την περίπτωση αφορά τη διάταξη και την προσβασιμότητα των εργαλειοθηκών και το πόσο εύληπτα παρουσιάζονται οι λειτουργίες τους στο σχεδιαστή. Ακόμα και ένας έμπειρος σχεδιαστής χρειάζεται ένα χρονικό διάστημα προσαρμογής και εξοικείωσης όταν από το ένα πρόγραμμα κατασκευής μεταπηδά σ' ένα άλλο. Σε κάθε περίπτωση πάντως, ο κανόνας που προκύπτει είναι ότι το πρόγραμμα κατασκευής θα πρέπει να παρέχει εποπτεία και δυνατότητα άμεσης παρέμβασης στην υπό ανάπτυξη εφαρμογή.

Βιβλιοθήκες αντικειμένων-δυνατότητες επέμβασης

Αν η ευχρηστία του προγράμματος κατασκευής είναι το πρώτο βήμα για τη μείωση του χρόνου κατασκευής ενός εικονικού κόσμου, οι βιβλιοθήκες έτοιμων αντικειμένων μειώνουν το χρόνο αυτό ακόμη περισσότερο. Ως αντικείμενα στην περίπτωση αυτή δεν εννοούνται μόνο τα τρισδιάστατα μοντέλα, αλλά οι ήχοι, οι συμπεριφορές των αντικειμένων, τα χρώματα και οι υφές που τα “ντύνουν”. Για παράδειγμα, ας υποθεθεί ότι ο σχεδιαστής θέλει να κατασκευάσει μία μπάλα που να μπορεί να αναπηδά στον εικονικό κόσμο. Υπάρχουν προγράμματα κατασκευής που του δίνουν τη δυνατότητα να ορίσει από τη βιβλιοθήκη τους ένα γεωμετρικό στερεό, μία σφαίρα στην περίπτωση του παραδείγματος.

Στη συνέχεια, από άλλη βιβλιοθήκη του ίδιου προγράμματος, ο σχεδιαστής μπορεί να επιλέξει τα χρώματα που θα “ντύσουν” κάθε ξεχωριστή επιφάνεια της σφαίρας ή μπο-

ρεί να υπάρχει μία έτοιμη εικόνα που να “ντύνει” όλη την επιφάνεια. Από άλλη βιβλιοθήκη επιλέγει και προσδίδει στο αντικείμενο την ιδιαίτερη συμπεριφορά της μπάλας (αναπήδηση το ύψος της οποίας μειώνεται με το χρόνο) την οποία και μπορεί να τροποποιήσει (μικρότερη ή μεγαλύτερη αναπήδηση). Τέλος, από τη βιβλιοθήκη των ήχων, ο σχεδιαστής εισάγει στο αντικείμενο το χαρακτηριστικό ήχο μίας μπάλας.

Αυτή είναι η πιο σύνθετη χρήση των έτοιμων βιβλιοθηκών. Σε αρκετές όμως περιπτώσεις τα προγράμματα κατασκευής περιέχουν εκτός των παραπάνω βιβλιοθηκών και βιβλιοθήκες έτοιμων αντικειμένων. Τότε ο σχεδιαστής δεν έχει παρά να επιλέξει μία έτοιμη μπάλα και να την εισαγάγει στον εικονικό κόσμο, τροποποιώντας πιθανώς ορισμένες λεπτομέρειες. Είναι εύκολα αντιληπτή η σημαντική διαφορά στο χρόνο που απαιτείται στη μία και στην άλλη περίπτωση.

Υπάρχει ένα ακόμα σημαντικό θέμα. Είναι αδύνατον ένα πρόγραμμα κατασκευής να περιλαμβάνει σε βιβλιοθήκη έτοιμων αντικειμένων όλα τα αντικείμενα που μπορεί να χρειαστεί ένας σχεδιαστής. Μάλιστα, το είδος και η ποσότητα των έτοιμων αντικειμένων διαφέρει από πρόγραμμα σε πρόγραμμα. Επιπλέον, ένα πρόγραμμα κατασκευής δε χρησιμοποιείται για τη δημιουργία ενός και μόνο εικονικού κόσμου, με αποτέλεσμα οι βιβλιοθήκες αυτές να πρέπει να περιέχουν μία σημαντική ποικιλία αντικειμένων. Το συμπέρασμα που προκύπτει είναι ότι ο σχεδιαστής πρέπει να επιλέξει ένα πρόγραμμα κατασκευής που οι βιβλιοθήκες του να ικανοποιούν σε μεγάλο βαθμό τις ανάγκες των εικονικών κόσμων που θα κατασκευάσει, και μάλιστα σε βάθος χρόνου.

Όσον αφορά τις συμπεριφορές και τις αλληλεπιδράσεις των αντικειμένων, και πάλι οι έτοιμες βιβλιοθήκες δεν μπορούν να καλύψουν πλήρως κάθε πιθανή περίπτωση. Η αντιμετώπιση αυτού του προβλήματος είναι διαφορετική, απαιτείται προγραμματιστική παρέμβαση, κατασκευή δηλαδή *scripts*. Ο σχεδιαστής πρέπει να ελέγξει αν το πρόγραμμα κατασκευής τού δίνει αυτή τη δυνατότητα, κατά πόσο η γλώσσα προγραμματισμού που χρησιμοποιείται είναι εύκολη στην εκμάθησή της και ποιος είναι ο βαθμός επέμβασης που μπορεί να έχει στην εφαρμογή του μέσω του προγραμματισμού.

Συνεργασία με εξωτερικά προγράμματα

Για το λόγο ότι σε καμία περίπτωση οι παραπάνω βιβλιοθήκες δεν μπορούν να καλύψουν πλήρως τις ανάγκες του σχεδιαστή, είναι ιδιαίτερα σημαντική η δυνατότητα του προγράμματος κατασκευής να συνεργάζεται με βοηθητικά προγράμματα. Η έννοια της συνεργασίας είναι αμφίδρομη. Θα πρέπει δηλαδή το πρόγραμμα κατασκευής να διαχειρίζεται αρχεία εξωτερικών προγραμμάτων, αλλά και να δημιουργεί αρχεία που να είναι δυνατή η επεξεργασία τους από τα εξωτερικά προγράμματα.

Αναφέρθηκε ήδη ότι ένα πρόγραμμα κατασκευής καλό θα ήταν να έχει δυνατότητες πρωτογενούς παραγωγής στοιχείων ενός εικονικού κόσμου τα οποία δεν περιλαμβάνονται στις βιβλιοθήκες του. Να περιέχει δηλαδή υποπρογράμματα που να επιτρέπουν την παραγωγή και επεξεργασία ήχων, την παραγωγή και επεξεργασία υφών, την παραγωγή και επεξεργασία χρωμάτων και την παραγωγή και επεξεργασία τρισδιάστατων μοντέλων. Εξαιτίας και πάλι της γενικής χρήσης των προγραμμάτων κατασκευής, τα υποπρογράμματα αυτά –όταν υπάρχουν– επιτρέπουν υποτυπώδη παραγωγή και επεξεργασία. Δεν είναι άστοχο να ειπωθεί ότι τελικά τα προγράμματα κατασκευής εικονικών κόσμων έχουν ως αντικείμενο τη συναρμολόγηση του κόσμου από διάφορα δομικά στοιχεία που προέρχονται από διάφορες πηγές.

Έτσι, ο σχεδιαστής καταφεύγει σε εξωτερικά και εξειδικευμένα προγράμματα για να καλύψει συγκεκριμένες ανάγκες που προκύπτουν κατά τη διάρκεια κατασκευής του εικονικού κόσμου. Αυτό βέβαια αυξάνει το χρόνο κατασκευής, δημιουργεί την ανάγκη ο σχεδιαστής να γνωρίζει τη χρήση μίας μεγάλης ποικιλίας άλλων προγραμμάτων ή παρουσιάζεται η ανάγκη συνεργασίας με άλλα άτομα γνώστες αυτών των προγραμμάτων. Το πιο σημαντικό όμως πρόβλημα που μπορεί να προκύψει είναι το πρόγραμμα κατασκευής να μην μπορεί να διαχειριστεί τα αρχεία από τα προγράμματα αυτά. Κάτι τέτοιο δε συμβαίνει τόσο στον τομέα του ήχου και των εικόνων, όσο στον τομέα των τρισδιάστατων αντικειμένων.

Τα σημαντικότερα προγράμματα επεξεργασίας ήχου και εικόνας διαθέτουν φίλτρα μετατροπής σχεδόν κάθε τύπου αρχείων. Το πρόβλημα με τα προγράμματα επεξεργασίας τρισδιάστατων γραφικών είναι ότι το καθένα απ' αυτά παράγει αρχεία ασύμβατα με τα αρχεία των άλλων προγραμμάτων. Τα φίλτρα μετατροπής προέρχονται συνήθως από ανεξάρτητους προγραμματιστές ή εταιρείες και αρκετές φορές η μετατροπή ενός τρισδιάστατου μοντέλου από τον έναν τύπο στον άλλο δεν είναι απαλλαγμένη από προβλήματα. Συνεπώς, κατά την επιλογή του προγράμματος κατασκευής πρέπει να προσεχθεί αν αυτό είναι σε θέση να αναγνωρίσει αρχεία από τα πιο δημοφιλή προγράμματα κατασκευής και επεξεργασίας κυρίως τρισδιάστατων μοντέλων.

Υποστήριξη

Το κόστος ενός προγράμματος κατασκευής εικονικών κόσμων είναι ιδιαίτερα σημαντικό. Από την άλλη, όσο έμπειρος και αν είναι ο σχεδιαστής, συχνά θα βρεθεί στη δύσκολη κατάσταση να μην ξέρει πώς να υλοποιήσει ένα στοιχείο του εικονικού του κόσμου ή πώς να βελτιώσει κάποια στοιχεία του. Αν μάλιστα το πρόγραμμα κατασκευής χρησιμοποιεί μία προγραμματιστική γλώσσα για τα *scripts* που δε γνωρίζει καλά ο σχεδιαστής, τότε τα προβλήματα πολλαπλασιάζονται. Σ' αυτή την περίπτωση ο

σχεδιαστής μπορεί να βρεθεί σε σοβαρό αδιέξοδο. Έκανε μία σημαντική επένδυση που δεν μπορεί να αξιοποιήσει πλήρως.

Οι εταιρείες που κατασκευάζουν πακέτα δημιουργίας εικονικών κόσμων αναγνωρίζουν αυτό το πρόβλημα και σε μικρό ή μεγάλο βαθμό παρέχουν υποστήριξη σε διάφορες μορφές. Η πιο απλή είναι η ενσωμάτωση στο κυρίως πρόγραμμα ενός συστήματος βοήθειας. Όταν το σύστημα βοήθειας είναι αρκετά αναλυτικό, λύνονται αρκετά από τα προβλήματα του σχεδιαστή. Στις περιπτώσεις που ο σχεδιαστής δε βρίσκει απάντηση στο πρόβλημά του, οι εταιρείες παρέχουν υποστήριξη με διάφορες άλλες μορφές, όπως μέσω τηλεφώνου, ηλεκτρονικού ταχυδρομείου, σελίδων στο Διαδίκτυο, κτλ. Μία τελευταία μορφή υποστήριξης μπορεί να προέλθει από διαδικτυακές ομάδες συζητήσεων των άλλων χρηστών του ίδιου προγράμματος.

Όμως τα προγράμματα κατασκευής εικονικών κόσμων απευθύνονται σ' ένα ιδιαίτερα εξειδικευμένο κοινό και δε γνωρίζουν τη διάδοση που έχουν άλλα είδη προγραμμάτων. Αυτό σημαίνει ότι η υποστήριξη του σχεδιαστή συνδέεται και εξαρτάται σε μεγάλο βαθμό από την κατασκευάστρια εταιρεία. Συνεπώς, κατά την επιλογή του προγράμματος ο σχεδιαστής θα πρέπει να ελέγξει το κατά πόσο και με τι όρους τού παρέχεται υποστήριξη. Επίσης πρέπει να ελεγχθεί και η διάδοση του συγκεκριμένου προγράμματος, παράγοντα που αποτελεί σημαντική ένδειξη για την ποιότητά του.

Διαχείριση των πόρων του υλικού

Έγινε εκτενής αναφορά στα προβλήματα και τους περιορισμούς του υλικού όσον αφορά τις εφαρμογές Ε.Π. και πώς αυτά επιδρούν στην ποιότητα του εικονικού κόσμου. Το πρόγραμμα κατασκευής θα πρέπει να επιτρέπει ιδιαίτερα ακριβείς ρυθμίσεις όλων των παραμέτρων που επηρεάζουν την ποιότητα, τόσο κατά το στάδιο της κατασκευής όσο και κατά τη διάρκεια εκτέλεσης της εφαρμογής από τον τελικό χρήστη.

Όταν μάλιστα ο εικονικός κόσμος είναι απαραίτητο να αναπαριστά εξωτερικούς χώρους, η κατάσταση περιπλέκεται. Το θέμα δεν αφορά μόνο τις βιβλιοθήκες των αντικειμένων, αλλά είναι πολύ πιο σύνθετο και αφορά τη διαχείριση του υλικού. Οι ανοιχτοί χώροι επιτρέπουν στο χρήστη να βλέπει προς όλες τις κατευθύνσεις και σε μεγάλη απόσταση. Αυτό έχει ως αποτέλεσμα, κατά την εκτέλεση της εφαρμογής, να σχεδιάζονται πολλά αντικείμενα που επιβαρύνουν σημαντικά το υπολογιστικό φορτίο. Το πρόγραμμα κατασκευής θα πρέπει να υλοποιεί μεθόδους αντιμετώπισης αυτού του προβλήματος που να μην επιδρούν στην ποιότητα και το ρεαλισμό της εφαρμογής.

Το θέμα της διαχείρισης των πόρων του υλικού έχει άλλη μία διάσταση. Κατά τη χρονική στιγμή που εμφανίζεται ένα πρόγραμμα κατασκευής εικονικών κόσμων, υπάρχει

μία δεδομένη κατάσταση αναφορικά με τις δυνατότητες του υλικού εκείνης της περιόδου. Αναφέρθηκε όμως ότι το υλικό εξελίσσεται πολύ γρήγορα, ενώ παράλληλα υπάρχουν συχνές εξελίξεις που αφορούν τα API. Αν ένα πρόγραμμα κατασκευής δεν παρακολουθεί τις εξελίξεις, με συχνές αναβαθμίσεις του αρχικού προγράμματος, τότε σύντομα χάνει την επαφή με αυτές, με αποτέλεσμα να μην αποδίδει ικανοποιητικά και να μην εκμεταλλεύεται τις νέες δυνατότητες του υλικού. Κατά την επιλογή λοιπόν του προγράμματος κατασκευής πρέπει να ελεγχθεί πόσο παλιά είναι η συγκεκριμένη του έκδοση και με τι συχνότητα εμφανίζονται νέες εκδόσεις.

Δικτυακή υποστήριξη

Ένα από τα σημαντικότερα χαρακτηριστικά του προγράμματος κατασκευής πρέπει να είναι η δυνατότητα ο παραγόμενος εικονικός κόσμος να επιτρέπει την ταυτόχρονη παρουσία παραπάνω του ενός χρηστών. Για το σκοπό αυτό θα πρέπει να υποστηρίζονται τα σημαντικότερα πρωτόκολλα τοπικού δικτύου, χωρίς να αποκλείεται η δυνατότητα διαδικτυακής εκτέλεσης της εφαρμογής. Πρέπει να σημειωθεί πάντως ότι τα τρισδιάστατα γραφικά, εκτός από τις ιδιαίτερες απαιτήσεις που έχουν από το υλικό, δημιουργούν ιδιαίτερα σημαντική κίνηση στο δίκτυο λόγω του μεγέθους των αρχείων τους. Συνεπώς, το πρόγραμμα κατασκευής θα πρέπει να χρησιμοποιεί τεχνικές που ελαττώνουν αυτή την κίνηση.

Χρήση περιφερειακών συσκευών

Αν και το συγκεκριμένο θέμα αφορά κυρίως τις άλλες μορφές Ε.Π. εκτός της επιτραπέζιας, εντούτοις η δυνατότητα ενός προγράμματος κατασκευής να υποστηρίζει εξειδικευμένες περιφερειακές συσκευές είναι ένα επιθυμητό στοιχείο. Πρώτα απ' όλα δε χρειάζεται η κατασκευή μίας καινούργιας εφαρμογής, αλλά μπορεί να τροποποιηθεί η υπάρχουσα και να εκτελεστεί και σε άλλες μορφές Ε.Π., παρέχοντας τη δυνατότητα ελέγχου των επιδράσεων που έχει η κάθε μορφή. Επίσης, όταν κάποιες περιφερειακές συσκευές γίνονται πιο προσιτές στο ευρύ κοινό, μπορούν να ενσωματωθούν χωρίς αλλαγές στην αρχική εφαρμογή και να διευρύνουν τις δυνατότητές της.

Κεφάλαιο 5^ο

*Εναλλακτική προσέγγιση
στο πρόβλημα κατασκευής εικονικών κόσμων*

Στο σημείο αυτό πρέπει να θέσουμε μία σειρά από ερωτήματα: Είναι απαραίτητο να χρησιμοποιηθεί στην κατασκευή εικονικών κόσμων αποκλειστικά και μόνο λογισμικό κατασκευασμένο γι' αυτόν το σκοπό, και μάλιστα υψηλού κόστους; Υπάρχει εναλλακτική –και πιο οικονομική– λύση; Ποιες άλλες αλληλεπιδραστικές εφαρμογές εκτός της Ε.Π. χρησιμοποιούν τρισδιάστατα γραφικά; Η απάντηση είναι μάλλον αναπάντεχη. Τα παιχνίδια για τους ηλεκτρονικούς υπολογιστές είναι οι ευρύτερα διαδεδομένες εφαρμογές τρισδιάστατων αλληλεπιδραστικών γραφικών. Χρειάζεται λοιπόν να εξεταστεί αν τα παιχνίδια μπορούν πράγματι να προσφέρουν μία ικανοποιητική λύση στο πρόβλημα κατασκευής εφαρμογών Ε.Π., πράγμα που γίνεται στο κεφάλαιο αυτό.

5.1 Τα τρισδιάστατα παιχνίδια βολών πρώτου προσώπου

Υπάρχουν πολλά είδη παιχνιδιών για ηλεκτρονικούς υπολογιστές, τα οποία κατατάσσονται σε διάφορες κατηγορίες (στρατηγικής, ρόλων, βολών, κ.ά.). Τα περισσότερα χρησιμοποιούν πλέον τρισδιάστατα γραφικά, με αποτέλεσμα στο σύνολό τους να είναι ιδιαίτερα απαιτητικές εφαρμογές για το υλικό. Είναι χαρακτηριστικό ότι οι ηλεκτρονικοί υπολογιστές που χρησιμοποιούνται για παιχνίδια είναι από τους καλύτερα εξοπλισμένους. Μία άλλη ειδική κατηγορία υλικού που προορίζεται αποκλειστικά για παιχνίδια είναι οι γνωστές *κονσόλες παιχνιδιών* (PlayStation 3, Xbox 360, Nintendo Wii, κτλ.).

Ο κύκλος εργασιών της βιομηχανίας ηλεκτρονικών παιχνιδιών πλησιάζει –και ίσως ξεπερνά– εκείνο των εφαρμογών Ε.Π., με μία ουσιαστική διαφορά. Τα παιχνίδια είναι κατά πολύ περισσότερο διαδεδομένες εφαρμογές, έχουν ένα ευρύτατο κοινό που αγγίζει το 40% του πληθυσμού όλων των ηλικιών στα ανεπτυγμένα κράτη [106].

Θα αναρωτηθεί κανείς ποιοι είναι οι παράγοντες που στρέφουν το ενδιαφέρον στα παιχνίδια, πώς αυτά σχετίζονται με τις εφαρμογές της Ε.Π., από τη στιγμή που ο κάθε τομέας απευθύνεται σε διαφορετικό κοινό. Η απάντηση δε βρίσκεται στο σύνολο των παιχνιδιών, αλλά σε μία συγκεκριμένη κατηγορία που ονομάζεται παιχνίδια “βολών πρώτου προσώπου” (*first person shoot them up's*).

Η γενική τους ιδέα είναι ότι ο χρήστης βρίσκεται σ' ένα τρισδιάστατο περιβάλλον μερικώς ή ολικώς εξερευνησιμο και αλληλεπιδραστικό. Το περιβάλλον το βλέπει με προσωπική πρώτου προσώπου, δηλαδή η κάμερα είναι επάνω του και από το σώμα του βλέπει μόνο τα χέρια του. Με βάση ένα σενάριο καλείται να αντιμετωπίσει μία αναρίθμητη στρατιά “αντιπάλων”, τους οποίους πρέπει να “εξοντώσει” με διάφορα όπλα, να λύσει γρίφους και να αποφύγει παγίδες, με απώτερο σκοπό να μείνει “ζωντανός” και να ολοκληρώσει τα διάφορα επίπεδα του παιχνιδιού.

5.1.1 Το ιστορικό των παιχνιδιών βολών πρώτου προσώπου

Πολύς δρόμος έχει διανυθεί από τη στιγμή που εμφανίστηκε το πρώτο παιχνίδι βολών πρώτου προσώπου μέχρι σήμερα. Η ραγδαία εξέλιξή τους άρχισε το 1992, όταν οι επεξεργαστές 80386 άρχισαν να έχουν ικανοποιητική διάδοση. Ο τίτλος που κυριάρχησε εκείνη την χρονιά ήταν το Wolfenstein 3D [107]. Η επανάσταση όμως έγινε την επόμενη χρονιά με την εμφάνιση του Doom [108]. Αυτό αξιοποίησε άριστα το υλικό, είχε εντυπωσιακά για την εποχή του γραφικά και κυρίως ήταν εξαιρετικά εθιστικό. Γνώρισε τεράστια διάδοση και, παρότι είναι πια ξεπερασμένο, υπάρχουν ακόμα φανατικοί φίλοι του.

Από το Doom ξεπήδησε μία σειρά παιχνιδιών “κλώνων” του, παιχνιδιών δηλαδή με διαφορετικό σενάριο και γραφικά που όμως στηρίζονταν στην ίδια τεχνολογία με την οποία κατασκευάστηκε ο πρωτότυπος τίτλος. Επόμενος σταθμός στην εξέλιξη της συγκεκριμένης κατηγορίας παιχνιδιών ήταν το 1996 με το Quake και τις συνέχειές του (Quake2 1998, Quake3 1999) [109]. Ανταγωνιστικό του Quake ήταν το Unreal (1998), που και αυτό είχε μία σειρά από συνέχειες [110]. Το αξιοσημείωτο είναι ότι τα παραπάνω παιχνίδια εξακολουθούν να βελτιώνονται, με πιο χαρακτηριστικές τις περιπτώσεις του Wolfenstein, του οποίου η τελευταία συνέχεια εμφανίστηκε το 2002 (Return to the castle of Wolfenstein), του Doom III (2003) και του Quake IV (2005).

Δεν είναι υπερβολή να ειπωθεί ότι τα παιχνίδια αυτής της κατηγορίας εκμεταλλεύονται στο έπακρο τις δυνατότητες του υλικού, σε τέτοιο βαθμό μάλιστα, που αποτελούν μέτρο σύγκρισης των επιδόσεών του. Πράγματι, τα μετρονομικά προγράμματα επιδόσεων των καρτών γραφικών χρησιμοποιούν συγκεκριμένα παιχνίδια για την εκτέλεση των μετρήσεών τους [111, 112]. Ένα ακόμα σημαντικό σημείο της εξέλιξης των παιχνιδιών αφορά τη δυνατότητα να παίζουν ταυτόχρονα πολλοί χρήστες εκμεταλλευόμενοι τις δυνατότητες του Διαδικτύου. Έτσι, κάποιος δεν αντιμετωπίζει πια μόνο ψεύτικους αντιπάλους, αλλά και πραγματικούς.

5.1.2 Κοινά τεχνικά χαρακτηριστικά παιχνιδιών και εφαρμογών Ε.Π.

Η προσεκτική μελέτη δείχνει ότι οι ομοιότητες των παιχνιδιών βολών πρώτου προσώπου και των εφαρμογών Ε.Π. είναι πάρα πολλές και σημαντικές. Σε τέτοιο βαθμό μάλιστα, που τα όρια μεταξύ των δύο να είναι πλέον δυσδιάκριτα. Ο όποιες διαφορές είναι δευτερεύουσας σημασίας και, το σημαντικότερο, οφείλονται στη διαφορετική φιλοσοφία με την οποία η κάθε κατηγορία προσεγγίζει το αντικείμενό της. Συνεπώς, αφού οι διαφορές δεν είναι τεχνικές, μπορούν να τροποποιηθούν και να εξαλειφθούν. Συνοπτικά οι ομοιότητες είναι:

- Η έννοια του κόσμου
- Τα API, οι ρυθμίσεις του υλικού
- Η δικτυακή υποστήριξη
- Η φιλικότητα χρήσης-εργονομία
- Η αλληλεπίδραση, οι σκανδάλες και τα γεγονότα
- Οι μη ελεγχόμενοι από το χρήστη χαρακτήρες
- Οι βαθμοί ελευθερίας, η κάμερα και οι περιφερειακές συσκευές
- Η τεχνητή νοημοσύνη
- Ο ρεαλισμός και η εμπύθιση
- Η διαχείριση του ήχου

Κόσμος

Υπάρχει η κοινή έννοια του “κόσμου”. Ο κόσμος είναι το περιβάλλον μέσα στο οποίο είναι τοποθετημένα όλα τα αντικείμενα και όπου εκτυλίσσονται όλα τα γεγονότα. Μπορεί να είναι μικρός όσο ένα δωμάτιο ή μεγάλος, καταλαμβάνοντας έκταση μερικών τετραγωνικών χιλιομέτρων. Ένας κόσμος μπορεί να είναι μία ενιαία επιφάνεια, αλλά μπορεί να είναι κατανεμημένος και σε διάφορα *επίπεδα* (levels). Το πέρασμα από ένα *επίπεδο* στο άλλο μπορεί να είναι ελεύθερο ή αποτέλεσμα των ενεργειών και αποφάσεων του χρήστη.

Ο κόσμος αρχικά δεν είναι τίποτα παραπάνω από μία επίπεδη επιφάνεια πάνω στην οποία ο σχεδιαστής σταδιακά τοποθετεί τα αντικείμενα. Χωρίς να είναι υποχρεωτικό, η πιο συνηθισμένη πρακτική είναι το σπάσιμο σε *υπο-επίπεδα*, ώστε να γίνεται καλύτερη διαχείριση του υλικού, όμοια όπως και στα προγράμματα κατασκευής εικονικών κόσμων. Επίσης, διευκολύνεται η εμποπτεία των κόσμων, ιδιαίτερα σε πολύ μεγάλους, που περιλαμβάνουν υπέργειες και υπόγειες επιφάνειες. Η διαδικασία κατασκευής και προσθήκης κίνησης στα αντικείμενα ενός κόσμου είναι σε μεγάλο βαθμό κοινή και στις δύο περιπτώσεις και περιλαμβάνει τη χρήση εξωτερικών προγραμμάτων κατασκευής τρισδιάστατων μοντέλων.

Αλληλεπίδραση, σκανδάλες, γεγονότα

Ο χρήστης και στις δύο περιπτώσεις έχει τη δυνατότητα να αλληλεπιδρά με τα αντικείμενα. Στις εφαρμογές Ε.Π. αυτό γίνεται συνήθως προγραμματιστικά, με τη χρήση *scripts*. Στα παιχνίδια βολών πρώτου προσώπου χρησιμοποιούνται οι έννοιες της *σκανδάλης* (trigger) και του *γεγονότος* (event). Για παράδειγμα, όταν ο χρήστης ακουμπήσει το πόμολο μίας πόρτας ενεργοποιεί μία σκανδάλη, που έχει ως αποτέλεσμα γεγονός το άνοιγμα της πόρτας.

Σε όποιο αντικείμενο δεν υπάρχει συνδεδεμένη μία *σκανδάλη* και ένα γεγονός, ο χρήστης δεν μπορεί να μεταβάλει τις ιδιότητές του, με αποτέλεσμα μικρότερο ή μηδενικό βαθμό αλληλεπίδρασης. Για παράδειγμα, ένα βάζο μπορεί να το σηκώσει, αλλά δεν μπορεί να το σπάσει αν το ρίξει κάτω. Το σπάσιμο του βάζου προϋποθέτει την ύπαρ-

ξη *σκανδάλης* (πτώση με συγκεκριμένη ταχύτητα) και *γεγονότος* (αντικατάσταση του μοντέλου με άλλο που αναπαριστά το σπασμένο βάζο). Για να γίνει ένας κόσμος απόλυτα αλληλεπιδραστικός, απαιτείται μεγάλος αριθμός από *σκανδάλες* και *γεγονότα*, που ενώ δεν υπάρχει περιορισμός στον αριθμό τους, εντούτοις απαιτεί επίπονη εργασία.

Οι μη ελεγχόμενοι από το χρήστη χαρακτήρες

Υπάρχει μία κατηγορία αντικειμένων τα οποία δεν είναι κατ' ανάγκη αλληλεπιδραστικά, που όμως συμβάλλουν στη συνολική ποιότητα του κόσμου της Ε.Π. και του παιχνιδιού. Σε αυτά τα αντικείμενα εκτυλίσσονται αυτόματα γεγονότα ανεξάρτητα από τη συμπεριφορά του χρήστη και φέρουν την ονομασία *μη ελεγχόμενοι από το χρήστη χαρακτήρες* (*Non Playable Characters, NPCs*) [87]. Αυτοκίνητα και άνθρωποι που κινούνται και περπατούν σ' ένα δρόμο δίνουν ζωντάνια στον κόσμο, χωρίς να παίζουν κάποιο συγκεκριμένο ρόλο. Η συμπεριφορά τους είναι αποτέλεσμα μίας αρκετά δύσκολης διαδικασίας προγραμματισμού, η οποία λαμβάνει υπόψη της πολλές πιθανές καταστάσεις. Για παράδειγμα, εφόσον οι κινήσεις τους είναι ως επί το πλείστον τυχαίες, πρέπει να προβλεφθεί το είδος της συμπεριφοράς τους αν συγκρουστούν μεταξύ τους ή μ' ένα άλλο εμπόδιο.

Τεχνητή Νοημοσύνη

Αντικείμενα με τα οποία ο χρήστης αλληλεπιδρά με σύνθετο τρόπο, που μεταβάλλουν τη συμπεριφορά τους ανάλογα με τις αποφάσεις του, είναι μία ιδιαίτερη περίπτωση αντικειμένων και στις δύο περιπτώσεις εφαρμογών. Η “νοημοσύνη” σε αντικείμενα ενός εικονικού κόσμου ή ενός παιχνιδιού είναι αποτέλεσμα συνδυασμένης χρήσης *γεγονότων* και προγραμματισμού με τη μορφή *scripts*. Το πόσο έξυπνο θα είναι ένα αντικείμενο εξαρτάται από το πόσο έξυπνο θέλει να το κάνει ο σχεδιαστής και πραγματικά δεν υπάρχουν όρια, πέρα από το χρόνο που απαιτείται για να γραφούν *scripts* που να προβλέπουν πολλές πιθανότητες.

Βαθμοί ελευθερίας, κάμερες

Ο χρήστης κατά την πλοήγησή του στον κόσμο έχει έξι βαθμούς ελευθερίας. Μετακινείται μπροστά, πίσω, δεξιά και αριστερά, αλλά μπορεί να κοιτάξει πάνω και κάτω. Μπορεί επίσης να περπατήσει, να τρέξει, να κολυμπήσει ή και να πετάξει. Αναφέρθηκε ήδη η έννοια της κάμερας, που είναι κοινή και στην Ε.Π. και στα παιχνίδια. Και τα δύο είδη εφαρμογών χρησιμοποιούν κατά κόρον την προοπτική πρώτου προσώπου, αλλά δίνουν τη δυνατότητα τρίτου προσώπου ή οπτικής από όποιο σημείο είναι τοποθετημένη μία κάμερα. Μπορεί επίσης να τοποθετηθεί ένας αριθμός από κάμερες οι οποίες ενεργοποιούνται με *σκανδάλες*.

Διαχείριση ήχου

Ο τρόπος με τον οποίο γίνεται η διαχείριση των ήχων είναι επίσης κοινός. Υπάρχει δηλαδή ένα τμήμα που αναλαμβάνει την εκτέλεση μουσικών κομματιών και ένα τμήμα για την εκτέλεση μεμονωμένων ήχων στα αντικείμενα. Η απλή στερεοφωνική υποστήριξη είναι πιο συνηθισμένη στις εφαρμογές Ε.Π. Στα παιχνίδια και επειδή ο ρόλος της μουσικής είναι αναβαθμισμένος, υποστηρίζεται πολυκάναλος ήχος και υπάρχει η δυνατότητα να αλλάζει το τέμπο ή το μουσικό κομμάτι ανάλογα με τα δρώμενα.

Δικτυακή υποστήριξη

Ένα από τα σημαντικότερα κοινά στοιχεία των εφαρμογών Ε.Π. και των παιχνιδιών είναι η δικτυακή υποστήριξη. Με τον όρο “δικτυακή υποστήριξη” εννοούνται δύο πράγματα: α) η εκτέλεση της εφαρμογής σε τοπικό δίκτυο και β) η εκτέλεση της εφαρμογής στο Διαδίκτυο. Το τελευταίο διάστημα και όπως ήδη αναφέρθηκε, ένα πολύ σημαντικό στοιχείο είναι η συνύπαρξη και αλληλεπίδραση πολλών χρηστών μέσα στην ίδια εφαρμογή, πράγμα επίσης που αφορά τη δικτυακή υποστήριξη.

Η ταχύτητα σύνδεσης δύο ή περισσότερων υπολογιστών σε τοπικό δίκτυο είναι υψηλή (10-100 Mbps) και ως εκ τούτου δεν είναι αιτία σημαντικών προβλημάτων. Η εφαρμογή (Ε.Π. ή παιχνίδι) μπορεί ακόμα να βρίσκεται και σ' έναν τρίτο υπολογιστή και οι χρήστες να την εκτελούν μέσω αυτού χωρίς να υπάρχει χρονική καθυστέρηση.

Στο Διαδίκτυο όμως τα πράγματα είναι πολύ διαφορετικά. Η ταχύτητα σύνδεσης ενός μέσου χρήστη είναι εξαιρετικά μικρή και ποικίλλει από τα 56 Kbps στην απλή τηλεφωνική γραμμή μέχρι τα 10 Mbps σε A.D.S.L. συνδέσεις. Συνεπώς, ο όγκος της πληροφορίας που μπορεί να διακινηθεί ανά μονάδα χρόνου είναι μικρότερος. Το πρόβλημα προσπάθησαν να το λύσουν οι εφαρμογές Ε.Π. που υλοποιούνται με VRML ή με συναφείς προγραμματιστικές γλώσσες, έχοντας πολύ μικρό όγκο (μερικά Mb), αλλά και πολύ χαμηλή ποιότητα γραφικών, όπως ήδη επισημάνθηκε. Πάντως, όταν ο χρήστης βρίσκεται σε μία ιστοσελίδα που περιέχει μία τέτοια εφαρμογή, χρειάζεται να περιμένει σχετικά λίγο χρόνο μέχρις ότου όλη η εφαρμογή έρθει στον υπολογιστή του και στη συνέχεια αυτή εκτελείται τοπικά.

Αντίθετα, ορισμένες εφαρμογές Ε.Π. και τα παιχνίδια έχουν όγκο εκατοντάδες ή χιλιάδες Mb. Είναι αδύνατο να περιμένει ο χρήστης να “κατέβει” όλη η εφαρμογή τοπικά και μετά να εκτελεστεί. Αναγκαστικά η εφαρμογή ή το παιχνίδι είναι εγκατεστημένο στον υπολογιστή του χρησιμοποιώντας κάποιο CD ή DVD που έχει προμηθευτεί. Πώς όμως υλοποιείται η δικτυακή υποστήριξη, η συνύπαρξη και αλληλεπίδραση πολλών χρηστών σ' αυτή την περίπτωση;

Εφόσον η εφαρμογή είναι ήδη εγκατεστημένη τοπικά, δεν είναι απαραίτητο να μεταδίδονται πληροφορίες που αφορούν το σχεδιασμό του κόσμου. Η μόνη πληροφορία που είναι απαραίτητο να μεταδίδεται είναι η θέση του κάθε χρήστη μέσα στον κόσμο. Αυτό γίνεται με τη μορφή συντεταγμένων που ανανεώνονται αρκετές φορές το δευτερόλεπτο. Ο κόσμος σχεδιάζεται τοπικά σε κάθε υπολογιστή, οι χρήστες σχεδιάζονται και τοποθετούνται και αυτοί τοπικά με βάση τις συντεταγμένες τους. Ο όγκος της πληροφορίας που μεταδίδεται είναι πραγματικά μικρός. Ελάχιστος είναι και ο όγκος των δεδομένων που αφορά την συνομιλία των χρηστών μεταξύ τους. Αν και δε λείπουν τα προβλήματα λόγω ταχύτητας σύνδεσης, η παραπάνω μέθοδος επιτρέπει ικανοποιητική ποιότητα επικοινωνίας δύο ή περισσότερων χρηστών σ' έναν εικονικό κόσμο [87].

Αξίζει να σημειωθεί ένα στοιχείο που προκύπτει από την παραπάνω μεθοδολογία υλοποίησης της δικτυακής υποστήριξης στα παιχνίδια. Ένα από τα σημαντικά προβλήματα που αντιμετωπίζουν οι ερευνητές σε πολλούς επιστημονικούς κλάδους, όπως για παράδειγμα στις γνωστικές επιστήμες, στην ψυχολογία και στην παιδαγωγική, είναι η ακρίβεια της καταγραφής των ποιοτικών δεδομένων του πειράματος ή της πιλοτικής εφαρμογής τους. Συνήθως καταφεύγουν στη βιντεοσκόπηση ή ηχογράφηση του πειράματος και στη συνέχεια προχωρούν στην ανάλυση των στοιχείων που προκύπτουν απ' αυτήν.

Στα παιχνίδια που υποστηρίζουν τη συμμετοχή πολλών παικτών υπάρχει ένας μηχανισμός που καταγράφει τις ενέργειες του κάθε παίκτη. Αυτός ο μηχανισμός στηρίζεται στην καταγραφή των διαδοχικών θέσεων του μοντέλου που αναπαριστά το χρήστη και η πληροφορία, αντί να μεταδοθεί μέσω δικτύου, αποθηκεύεται τοπικά και μπορεί να ξαναπαιχτεί. Έτσι ο παίκτης ή ο ερευνητής μπορεί να ξαναδεί το παιχνίδι με τη μορφή βίντεο, χωρίς να έχει χρησιμοποιήσει επιπλέον εξοπλισμό. Αυτή η δυνατότητα αξίζει να διερευνηθεί περισσότερο και να εξεταστούν τρόποι αξιοποίησής της.

API, ρυθμίσεις και χρήση του υλικού

Τόσο οι εφαρμογές Ε.Π. όσο και τα παιχνίδια είναι σε θέση να χρησιμοποιήσουν τα δύο API για την υλοποίησή τους. Τις περισσότερες φορές χρησιμοποιούν και τα δύο, δίνοντας στο χρήστη τη δυνατότητα να επιλέξει με βάση ποιο απ' αυτά θα εκτελείται η εφαρμογή. Στο χρήστη αφήνεται επίσης και η επιλογή των ρυθμίσεων που θα του επιτρέψουν την καλύτερη αξιοποίηση του υλικού του. Τα παιχνίδια συνήθως επιτρέπουν περισσότερες ρυθμίσεις, μιας και απευθύνονται σε ευρύτερο κοινό, με μεγαλύτερη ποικιλία υλικού.

Οι εφαρμογές Ε.Π. επιτρέπουν λιγότερες ρυθμίσεις, γιατί και το κοινό τους είναι κατά πολύ πιο περιορισμένο και οι προδιαγραφές των υπολογιστών που χρησιμοποιούνται

είναι πολύ πιο συγκεκριμένες. Ένα ενδιαφέρον σημείο είναι ότι, επειδή τα παιχνίδια πρέπει να εκτελούνται ικανοποιητικά σε μεγάλη ποικιλία διαμορφώσεων ηλεκτρονικών υπολογιστών, υλοποιούν τεχνικές πολύ καλύτερης αξιοποίησης του υλικού του χρήστη, στοιχείο όχι απόλυτα απαραίτητο σ' ένα πρόγραμμα κατασκευής εικονικών κόσμων.

Φιλικότητα χρήσης-εργονομία

Το κοινό αυτό χαρακτηριστικό αφορά τον τελικό χρήστη της εφαρμογής. Μέρος της επιτυχίας μίας εφαρμογής στηρίζεται στο κατά πόσο αυτή είναι εύχρηστη, πόσο φιλικό είναι το περιβάλλον διεπαφής και πόσο χρόνο χρειάζεται ο χρήστης ώστε να εξοικειωθεί με αυτή. Για το χρόνο εκμάθησης της χρήσης ενός προγράμματος χρησιμοποιείται ο όρος *καμπύλη εκμάθησης (learning curve)*. Για παράδειγμα, αν ένα πρόγραμμα χρησιμοποιεί πολλά πλήκτρα ή συνδυασμούς πλήκτρων για την εκτέλεση ορισμένων διαδικασιών, τότε η *καμπύλη εκμάθησης* είναι μεγαλύτερη από ένα πρόγραμμα που εκτελεί τις ίδιες λειτουργίες με λιγότερα πλήκτρα. Το περιβάλλον διεπαφής είναι επίσης σημαντικό, γιατί θα πρέπει να περιλαμβάνει όλες τις απαραίτητες για το χρήστη πληροφορίες χωρίς όμως να τον κουράζει με επουσιώδη και δευτερεύοντα στοιχεία.

Η χρήση ειδικών περιφερειακών συσκευών στις εφαρμογές Ε.Π., ποντικιού και χειριστηρίου στα παιχνίδια, απλοποιεί σε μεγάλο βαθμό τη χρήση τους και επιτυγχάνεται έτσι μικρή *καμπύλη εκμάθησης* [87]. Το περιβάλλον διεπαφής και στις δύο περιπτώσεις είναι αποτέλεσμα προγραμματισμού, οπότε μπορεί να μεταβληθεί αν διαπιστωθεί ότι δεν ικανοποιεί το χρήστη. Τέλος τα παιχνίδια, αν και σε μικρό βαθμό, επιτρέπουν προσαρμογή του περιβάλλοντος από τον ίδιο το χρήστη, με την προσθαφαίρεση στοιχείων του από μενού επιλογών.

Ρεαλισμός-εμβύθιση

Ο ρεαλισμός που προκύπτει από τις αλληλεπιδράσεις, τον ήχο και τα υψηλής ποιότητας γραφικά οδηγούν στην εμβύθιση, που είναι η βασικότερη επιδίωξη και των δύο ειδών εφαρμογών. Βέβαια, το κάθε είδος την επιδιώκει για διαφορετικούς λόγους το καθένα: για εκπαίδευση οι εφαρμογές Ε.Π. και για ψυχαγωγία τα παιχνίδια. Ανεξάρτητα απ' αυτό, το συμπέρασμα είναι ότι για να πετύχουν το σκοπό τους και τα δύο είδη χρησιμοποιούν παρόμοιες μεθόδους, τεχνικές και μέσα, με λίγο ως πολύ τον ίδιο τρόπο.

5.1.3 Διαφορές Ε.Π. και παιχνιδιών βολών πρώτου προσώπου

Οι διαφορές εφαρμογών Ε.Π. και παιχνιδιών είναι επουσιώδεις και πηγάζουν από το διαφορετικό σκοπό που εξυπηρετούν και υπάρχουν τρόποι να εξαλειφθούν. Διαφορές υπάρχουν στα εξής σημεία:

- Στην πλοκή
- Στο σκορ
- Στην υποστήριξη της εφαρμογής
- Στο μέσο και στο υλικό εκτέλεσης της εφαρμογής
- Στη βία
- Στο εκτελέσιμο πρόγραμμα-player

Πλοκή

Τα παιχνίδια είναι αναμενόμενο να έχουν μία συγκεκριμένη πλοκή. Υπάρχει ένα σενάριο το οποίο εκτυλίσσεται οδηγώντας το χρήστη σε μία συγκεκριμένη κατεύθυνση. Πρέπει να ολοκληρώσει μία “αποστολή” για να του ανατεθεί η επόμενη, που συνήθως είναι πιο δύσκολη, με περισσότερους αντιπάλους, παγίδες και γρίφους. Όταν ολοκληρωθούν με επιτυχία όλες οι αποστολές, το παιχνίδι έχει τελειώσει. Διαπιστώνεται ότι υπάρχει μία γραμμική πορεία εξέλιξης των γεγονότων σ' ένα παιχνίδι. Παρότι υπάρχουν παιχνίδια με τυχαία γεγονότα, με παράλληλα σενάρια, εναλλακτικές ή μη υποχρεωτικές αποστολές, όλα καταλήγουν σ' ένα κοινό τέλος με ελαφρές παραλλαγές. Στην ουσία η γραμμικότητα παραμένει.

Πού οφείλεται όμως αυτό; Ο σχεδιαστής του παιχνιδιού τοποθετεί στα *επίπεδα* του κόσμου ένα είδος *σκανδάλης*, που δε διαφέρει απ' αυτές που ενεργοποιούν τοπικά γεγονότα και που ο σκοπός της είναι να ενεργοποιήσει γενικές εξελίξεις. Δεν είναι απαραίτητο να είναι μόνο μία. Πολύ συχνά τοποθετείται ένας αριθμός απ' αυτές και ανάλογα με τη σειρά ενεργοποίησής τους ή τον αριθμό των ενεργοποιημένων, η πλοκή εκτυλίσσεται με διαφορετικό τρόπο.

Αν δεν τοποθετηθούν καθόλου ή αν ο χρήστης δεν τις ενεργοποιήσει, το αποτέλεσμα είναι μία άσκοπη περιπλάνηση στον κόσμο. Κάτι τέτοιο είναι αρνητικό στοιχείο για ένα παιχνίδι που οφείλει να κρατά το ενδιαφέρον του παίκτη συνεχώς σε υψηλά επίπεδα. Βέβαια, μπορεί να είναι συνειδητή επιλογή του σχεδιαστή να τις κρύψει καλά ή να χρησιμοποιήσει περίπλοκους συνδυασμούς τους, έτσι ώστε να αυξήσει το βαθμό δυσκολίας του παιχνιδιού. Υπάρχει μία λεπτή ισορροπία, που αν ανατραπεί, το παιχνίδι γίνεται πολύ δύσκολο και χάνει το ενδιαφέρον του.

Η γραμμικότητα, το σενάριο και η πλοκή δεν είναι απαραίτητα συστατικά στοιχεία μίας εφαρμογής Ε.Π. Σε πολλές περιπτώσεις, όπως για παράδειγμα στις εκπαιδευτικές εφαρμογές, η γραμμική εξέλιξη είναι ένα από τα στοιχεία που κάποιες παιδαγωγικές

αντιλήψεις δε συνιστούν. Το σενάριο, από την άλλη πλευρά, μπορεί να είναι γενικό και αφηρημένο ή συγκεκριμένο. Σε κάθε περίπτωση όμως, ο τρόπος εξέλιξης της πλοκής, αν υπάρχει, είναι στοιχείο που εξαρτάται από τις σκανδάλες και τον τρόπο που αυτές χρησιμοποιούνται.

Βία

Το κύριο συστατικό στοιχείο των παιχνιδιών βολών είναι η βία. Βία που είναι μάλιστα αλόγιστη και σε υπερβολικές ποσότητες [113]. Τα περισσότερα παιχνίδια έχουν χαρακτηριστεί κατάλληλα για νεαρά άτομα άνω των 13 ή 15 ετών. Μέρος της επιτυχίας των παιχνιδιών οφείλεται στην ποικιλία “όπλων” που χρησιμοποιεί ο παίκτης για να εξολοθρεύσει τους αντιπάλους του και στο πόσο εντυπωσιακοί είναι αυτοί οι αντίπαλοι.

Όμως τα παιχνίδια είναι ένα “ψυχαγωγικό” εμπορικό προϊόν. Οι εταιρείες κατασκευής τους έχουν διαπιστώσει ότι η βία αυξάνει τις πωλήσεις και ως εκ τούτου φροντίζουν να την παρέχουν απλόχερα [113]. Για να παρακάμψουν ορισμένους νομοθετικούς περιορισμούς, δίνουν τη δυνατότητα στο χρήστη να αλλάζει το χρώμα ή να μην εμφανίζεται καθόλου αίμα, επιλογές που σπάνια χρησιμοποιούνται. Από τεχνική άποψη όμως, η βία ως στοιχείο μίας εφαρμογής εύκολα εξαλείφεται.

Όλα ξεκινούν από τα μοντέλα που χρησιμοποιούνται σ' έναν κόσμο, στα οποία δίνονται κάποιες ιδιότητες. Μία ομάδα ιδιοτήτων αφορά τη δυνατότητα να προξενούν ή να δέχονται βλάβες από άλλα μοντέλα. Όταν η βλάβη φτάσει σε μία προκαθορισμένη τιμή, το μοντέλο “σκοτώνεται”. Αν δε δοθούν τιμές σ' αυτές τις ιδιότητες και αν παραλειφθούν τα μοντέλα όπλων και εχθρών, η βία παύει να υπάρχει.

Βαθμολογία-σκορ

Ο χρήστης ενός παιχνιδιού ανάλογα με τη συμπεριφορά και τις επιδόσεις του ανταμείβεται με διάφορους τρόπους (ξεκλειδωμά κρυφών επιπέδων, νέα όπλα, περισσότερη δύναμη, κτλ.). Η πιο απλή και αποτελεσματική επιβράβευση είναι η συγκέντρωση βαθμών ανάλογα με τις επιτυχίες του, το σκορ. Είναι ένα μέσο σύγκρισης με τις επιδόσεις άλλων παικτών και συνάμα ένας πολύ αποτελεσματικός τρόπος να αναγκαστεί ο παίκτης να ξαναπαίξει το παιχνίδι, που αποτελεί και δείκτη επιτυχίας του παιχνιδιού. Δεν αρκεί η ολοκλήρωση των επιπέδων ενός παιχνιδιού, αυτό πρέπει να γίνει με τη μεγαλύτερη δυνατή βαθμολογία.

Αναμφισβήτητα υπάρχουν εφαρμογές της Ε.Π. όπου η βαθμολογία είναι απαραίτητη, όπως υπάρχουν άλλες όπου αυτό το στοιχείο δε χρειάζεται. Η βαθμολογία υλοποιείται μέσω *μετρητών* μέσα στο πρόγραμμα που αυξάνονται με βάση στοιχεία που στέλνουν

σκανδάλες. Είναι στο χέρι του σχεδιαστή να τις τοποθετήσει, να τις αφαιρέσει ή να κάνει τα στοιχεία της βαθμολογίας μη προσβάσιμα στο χρήστη.

Το τελευταίο χαρακτηριστικό είναι ιδιαίτερα χρήσιμο σε ερευνητικές εφαρμογές όπου είναι επιθυμητή η καταγραφή στοιχείων για τη συμπεριφορά του χρήστη χωρίς αυτός να είναι ενήμερος. Στα παιχνίδια, το στοιχείο της βαθμολογίας είναι ενσωματωμένο και αναπόσπαστο κομμάτι της εφαρμογής. Συνεπώς η υλοποίηση και διαχείρισή του είναι πιο εύκολη απ' ό,τι στις εφαρμογές Ε.Π., όπου συνήθως χρειάζεται προγραμματιστική προσέγγιση (*scripts*) για να επιτευχθεί το ίδιο αποτέλεσμα. Απ' αυτή τη σκοπιά, τα παιχνίδια μάλλον υπερτερούν των εφαρμογών Ε.Π.

Μέσο και υλικό εκτέλεσης της εφαρμογής

Αναφέρθηκε ότι τα υπολογιστικά συστήματα που χρησιμοποιούνται στις εφαρμογές Ε.Π. έχουν ένα μέσο κόστος που είναι πολλαπλάσιο του κόστους ενός πολύ καλού οικιακού υπολογιστή. Η διαφορά αυτή οφείλεται στη χρήση ειδικών και ιδιαίτερα ακριβών περιφερειακών συσκευών, που εντείνουν το ρεαλισμό της εφαρμογής και που επιτρέπουν την εμβύθιση του χρήστη μέσα στο εικονικό περιβάλλον. Αντίθετα, τα παιχνίδια είναι εφαρμογές που προορίζονται να εκτελεστούν σε οικιακούς υπολογιστές χωρίς την ύπαρξη τέτοιων συσκευών. Όμως, όσοι παίζουν παιχνίδια χρησιμοποιούν πολλές φορές χειριστήρια αντί για το πληκτρολόγιο ή το ποντίκι. Αυτό σημαίνει ότι υπάρχει η δυνατότητα, με κατάλληλες εντολές, να ελέγχονται και άλλες περιφερειακές συσκευές, ιδιαίτερα κρίσιμο χαρακτηριστικό της Ε.Π.

Αν προσεγγιστεί το θέμα από την πλευρά του προγραμματισμού, διαπιστώνεται ότι στα παιχνίδια δεν υπάρχει ο κώδικας εκείνος που θα επέτρεπε είσοδο και έξοδο δεδομένων από και προς εξειδικευμένες περιφερειακές συσκευές. Το πληκτρολόγιο, το ποντίκι και το χειριστήριο αρκούν για να παίξει κάποιος ένα παιχνίδι. Αυτό όμως δεν αποκλείει τη δυνατότητα να γραφεί κώδικας για τον έλεγχο των ίδιων ακριβώς συσκευών που χρησιμοποιούν οι εφαρμογές Ε.Π. Δεν υλοποιείται, γιατί κανένας χρήστης δε θα ενδιαφερόταν να τις αγοράσει μόνο και μόνο για να παίξει ένα παιχνίδι.

Το τελευταίο διάστημα αρκετές κάρτες γραφικών ενσωματώνουν χαρακτηριστικά που συναντώνται σε εξελιγμένα συστήματα Ε.Π. Για παράδειγμα, δίνουν τη δυνατότητα στεροσκοπικής προβολής της εικόνας (μία τεχνική που δίνει την αίσθηση του βάθους από τη σύνθεση δύο ίδιων εικόνων ελαφρά μετατοπισμένων η μία από την άλλη). Οι δύο εικόνες ανασυντίθενται σε μία με τη χρήση ειδικών γυαλιών, τα οποία κάποιος εταιρείες κατασκευής καρτών γραφικών δίνουν μαζί με την αγορά μίας κάρτας. Η στεροσκοπική προβολή όμως είναι κατεξοχήν στοιχείο εφαρμογών ημι-εμβυθισμένης ή εμβυθισμένης Ε.Π.

Μία άλλη περίπτωση είναι τα *Head Mounted Displays (H.M.Ds)*, συσκευές που μοιάζουν με γυαλιά και προβάλλουν την εικόνα απευθείας στα μάτια του χρήστη. Στο παρελθόν χρησιμοποιούνταν αποκλειστικά από την Ε.Π. Πρόσφατα, μετά τη σημαντική μείωση των τιμών τους, άρχισαν να διαδίδονται ως απλές οικιακές συσκευές. Διαπιστώνεται μία σύγκλιση –αν και αρκετά πρώιμη– του υλικού που χρησιμοποιούν αυτά τα δύο είδη εφαρμογών, που σε συνδυασμό με τις δυνατότητες που δίνει ο προγραμματισμός οδηγούν σε ταύτιση.

Εκτελέσιμο πρόγραμμα-Player

Ο χρήστης και στις δύο περιπτώσεις αγνοεί τα προγράμματα με τα οποία αναπτύχθηκαν οι εφαρμογές και χρησιμοποιεί μόνο το τελικό προϊόν, που έχει τη μορφή εκτελέσιμου προγράμματος, αρχείων βιβλιοθηκών δυναμικής διασύνδεσης (*Dynamic Link Libraries, DLLs*), κτλ. Στις εφαρμογές Ε.Π. συνηθίζεται, χωρίς να συμβαίνει πάντα, ο εικονικός κόσμος να είναι εντελώς αποσυνδεδεμένος από το πρόγραμμα κατασκευής, με αποτέλεσμα ο τελικός χρήστης να χρειάζεται μόνο ένα τρίτο πρόγραμμα που εκτελεί την εφαρμογή (*player*).

Στα παιχνίδια συνηθίζεται, χωρίς και πάλι αυτό να αποκλείει το αντίθετο, ο κόσμος να είναι αναπόσπαστο κομμάτι του προγράμματος ανάπτυξης, μέρος του οποίου είναι και ο *player*. Κατά κάποιον τρόπο, στην περίπτωση αυτή, ο χρήστης αγοράζει μαζί με το παιχνίδι και με πολύ χαμηλό κόστος το πρόγραμμα με το οποίο κατασκευάστηκε. Αυτό το στοιχείο είναι ιδιαίτερα σημαντικό, θα αναπτυχθεί σε έκταση σε επόμενο κεφάλαιο και είναι από τους σημαντικότερους λόγους για τους οποίους υποστηρίζεται η θέση ότι τα τρισδιάστατα παιχνίδια μπορούν να ανταγωνιστούν τα ειδικά προγράμματα κατασκευής εικονικών κόσμων.

Υποστήριξη της εφαρμογής

Μία από τις πιο κρίσιμες παραμέτρους σε μία εφαρμογή είναι η υποστήριξη που παρέχεται σ' αυτή από την κατασκευάστρια εταιρεία ή από τρίτους. Τα προγράμματα κατασκευής εφαρμογών Ε.Π. υποστηρίζονται σε δύο επίπεδα από τις κατασκευάστριες εταιρείες. Σε πρώτο επίπεδο υπάρχουν γενικές κατηγορίες τεχνικής υποστήριξης (για την επίλυση προβλημάτων στην εκτέλεση του προγράμματος, για την επίλυση προβλημάτων στην ανάπτυξη εφαρμογών, κτλ.).

Σε δεύτερο επίπεδο υπάρχει η δυνατότητα να παρασχεθεί βοήθεια σε εξατομικευμένη μορφή, όταν το πρόβλημα που παρουσιάζεται δεν μπορεί να επιλυθεί από τις γενικές κατηγορίες. Υπάρχει και μία τρίτη δυνατότητα, αυτή της παροχής βοήθειας από άλλους χρήστες, που οργανώνονται σε ομάδες συζητήσεων και λειτουργούν ανεξάρτητα από την κατασκευάστρια εταιρεία.

Αντίθετα, στα παιχνίδια η κατασκευάστρια εταιρεία παρέχει μεν υποστήριξη, αλλά όχι σε τέτοια έκταση. Όμως οι χρήστες είναι οργανωμένοι σε πολυπληθείς ομάδες συζητήσεων που καλύπτουν όλα τα θέματα. Το σημείο αυτό είναι μία ουσιαστική διαφορά με τα προγράμματα Ε.Π. Αυτά, λόγω της μικρής τους διάδοσης, έχουν σχετικά ολιγομελείς ομάδες συζητήσεων και γενικά είναι αρκετά δύσκολο να παρασχεθεί βοήθεια σε μεγάλη έκταση και βάθος. Τα παιχνίδια, λόγω της μεγάλης τους διάδοσης, συγκεντρώνουν χιλιάδες χρήστες σε κάθε ομάδα. Χαρακτηριστικά, το *Serious Sam*, που είναι παιχνίδι με σχετικά μικρή διάδοση, έχει περίπου 10.000 χρήστες στις ομάδες συζητήσεων [114], που καλύπτουν παραπάνω από ικανοποιητικά τις ερωτήσεις που μπορεί να έχει κάποιος που θέλει να αναπτύξει μία εφαρμογή Ε.Π. χρησιμοποιώντας ένα παιχνίδι.

5.2 Πεδία εφαρμογών των παιχνιδιών βολών πρώτου προσώπου

Από τη σύγκριση εφαρμογών Ε.Π. και παιχνιδιών προκύπτει ότι τα παιχνίδια μπορούν –θεωρητικά τουλάχιστον– να χρησιμοποιηθούν σ' όλα εκείνα τα πεδία όπου ήδη χρησιμοποιούνται εφαρμογές Ε.Π. Ισχύει όμως κάτι τέτοιο; Υπάρχουν παραδείγματα εφαρμογών Ε.Π. όπου χρησιμοποιήθηκαν τρισδιάστατα παιχνίδια, εκτός από ψυχαγωγικούς σκοπούς; Η απάντηση είναι καταφατική. Τα παιχνίδια χρησιμοποιούνται ήδη σ' ένα ευρύ φάσμα δραστηριοτήτων και επιστημών, με πιο χαρακτηριστικά τα παρακάτω παραδείγματα.

5.2.1 Εφαρμογές στον τομέα της άμυνας

Το Ινστιτούτο Ερευνών του Αμερικανικού Στρατού (*Army Research Institute, Simulator Systems Research Unit*) αναζήτησε μεθόδους ώστε να εκπαιδεύσει τους στρατιώτες στη νέα πολεμική τεχνολογία, στα νέα όπλα και γενικά στις νέες τακτικές στο πεδίο της μάχης [115]. Θεώρησε ότι οι εφαρμογές Ε.Π. είναι η ενδεδειγμένη λύση, αλλά λόγω του μεγάλου κόστους του εξοπλισμού και του γεγονότος ότι ο εξοπλισμός αυτός δεν είναι αρκετά εύχρηστος ώστε να συμβάλει στη δημιουργία του αισθήματος της εμπύθισης, διερεύνησε και μία διαφορετική προσέγγιση.

Οι υπεύθυνοι του προγράμματος D.I.V.E. (*Dismounted Infantry Virtual Environment*) έστρεψαν το ενδιαφέρον τους κυρίως στις πνευματικές λειτουργίες και διεργασίες που συντελούνται κατά τη διάρκεια μίας προσομοίωσης και αναζήτησαν παιχνίδια τα οποία θα μπορούσαν να τροποποιήσουν [116]. Κατέληξαν τελικά στο *Half-Life*, που είναι και αυτό ένα από τα παιχνίδια βολών πρώτου προσώπου. Αντικαταστάθηκαν τα όπλα του παιχνιδιού με το τυπικό όπλο που χρησιμοποιεί ο αμερικανικός στρατός, εξαλείφθηκαν όλα τα στοιχεία επιστημονικής φαντασίας (το παιχνίδι εκτυλίσσεται στο μέλλον) και αντικαταστάθηκαν όλοι οι αντίπαλοι-τέρατα από πλήρως ελεγχόμενους από τον παίκτη ανθρώπινους χαρακτήρες. Αυτό που διατηρήθηκε ήταν το *Deathmatch*, που είναι η

δυνατότητα να παίζουν δικτυακά και ταυτόχρονα πολλοί παίκτες, ώστε το παιχνίδι να μπορεί να παιχτεί από μία διμοιρία στρατιωτών.

Η πιλοτική εφαρμογή κράτησε ένα χρόνο και δοκιμάστηκαν σενάρια πολεμικών εμπλοκών σε αστικό περιβάλλον για τα οποία ο στρατός είχε ήδη δεδομένα από πραγματικές εκπαιδεύσεις. Τα αποτελέσματα ήταν ιδιαίτερα θετικά. Το σημαντικότερο στοιχείο ήταν ότι οι διαφορετικές δεξιότητες παιξίματος ενός παιχνιδιού από τον κάθε παίκτη-στρατιώτη δεν επηρέασαν την απόδοσή του κατά την προσομοίωση. Λειτουργήσε το στοιχείο της εμπύθισης, η προσομοίωση αντιμετωπίστηκε ως εκπαίδευση σε πραγματικό περιβάλλον. Αξιοποιώντας τη δυνατότητα καταγραφής των ενεργειών του παίκτη, έγινε πολύ γρηγορότερα η αξιολόγηση των επιδόσεων και η επισήμανση των αδυναμιών και προτερημάτων του κάθε συμμετέχοντα [115].

5.2.2 Εφαρμογές στην αρχιτεκτονική

Το αυξημένο κόστος απόκτησης εξοπλισμού και λογισμικού Ε.Π. ήταν επίσης η αιτία που στο Martin Centre for Architectural and Urban Studies του University of Cambridge ασχολήθηκαν με τα παιχνίδια βολών πρώτου προσώπου [117]. Στην περίπτωση αυτή η αλληλεπίδραση του χρήστη με το περιβάλλον δεν έπαιζε τόσο σημαντικό ρόλο όσο στη στρατιωτική εφαρμογή. Αυτό που ενδιέφερε ήταν: α) η πιστή αναπαράσταση του χώρου και συγκεκριμένα του εσωτερικού ενός κτιρίου και β) η δυνατότητα να είναι προσβάσιμη η εφαρμογή από το Διαδίκτυο. Χρησιμοποιήθηκε το Quake, γιατί, παρόλο που είναι ξεπερασμένο, ο κώδικας του προγράμματος δίνεται δωρεάν από την κατασκευάστρια εταιρεία.

Όπως και στην προηγούμενη περίπτωση, έτσι και σ' αυτή, οι αντίπαλοι-τέρατα αποτέλεσαν πρόβλημα. Και πάλι τα μοντέλα τους αντικαταστάθηκαν από μοντέλα ανθρώπων. Επίσης, επεμβαίνοντας στον κώδικα άλλαξαν τη συμπεριφορά των μοντέλων ώστε να μην είναι επιθετικά, αλλά να συμπεριφέρονται "ανθρώπινα". Στα θετικά σημεία της εφαρμογής αναφέρεται η πολύ καλή ποιότητα φωτισμού και σκιάσεων, που δίνει ικανοποιητικά επίπεδα ρεαλισμού.

Αντίθετα, χρειάστηκαν καινούργιες και πιο λεπτομερείς υφές για τα αντικείμενα, γιατί το παιχνίδι υστερούσε στον τομέα αυτό. Ο προγραμματισμός φάνηκε χρήσιμος για την προσθήκη συμπεριφορών σε αντικείμενα (ανελκυστήρες, αυτόματα μηχανήματα καφέ, πόρτες, κτλ.), αλλά και για την τοποθέτηση της κάμερας με τέτοιο τρόπο, ώστε ο χρήστης να βλέπει τον εαυτό του (οπτική τρίτου προσώπου). Τέλος, αξιοποιήθηκε και σ' αυτή την περίπτωση το Deathmatch για την εκτέλεση της εφαρμογής στο Διαδίκτυο.

5.2.3 Εφαρμογές στην εκπαίδευση

Ένα παράδειγμα από το χώρο της εκπαίδευσης αφορά το πρόγραμμα Virtual Florida Everglades, την αναπαράσταση των υδροβιότοπων της Φλόριντα (Virtual World Heritage Project), που χρηματοδοτήθηκε από την UNESCO. Ο σκοπός του προγράμματος ήταν η ανάπτυξη ενός πλούσιου σε λεπτομέρειες εικονικού περιβάλλοντος για την ευαισθητοποίηση του κοινού σε θέματα που αφορούν οικοσυστήματα σε κίνδυνο [118].

Το ζητούμενο σ' αυτή την περίπτωση ήταν το κόστος της εφαρμογής να είναι χαμηλό και για το λόγο αυτό προτιμήθηκε, όπως και στις προηγούμενες περιπτώσεις, ένα παιχνίδι βολών πρώτου προσώπου, το Unreal. Χρησιμοποιώντας το παιχνίδι, μοντελοποιήθηκε και κατασκευάστηκε μία πιστή αναπαράσταση ενός μέρους του υδροβιότοπου, χρησιμοποιήθηκε πολυκάναλος ήχος και προστέθηκαν κινούμενα ζώα. Ο χρήστης μπορούσε να πλοηγηθεί στον εικονικό κόσμο "πετώντας", έχοντας δηλαδή άποψη πρώτου προσώπου, αλλά από ψηλά.

Ακόμα ένα παράδειγμα χρήσης παιχνιδιού στην εκπαίδευση, συγκεκριμένα και πάλι του Unreal, έρχεται από τη NASA. Αφορά την αναπαράσταση του Διεθνούς Διαστημικού Σταθμού (Virtual International Space Station, VISS) [119]. Ο σταθμός αποδίδεται με μεγάλη λεπτομέρεια χρησιμοποιώντας τα πραγματικά τεχνικά του στοιχεία. Ο χρήστης μπορεί να περπατήσει στο εσωτερικό του, να πετάξει έξω στο διάστημα και να συνεργαστεί με άλλους χρήστες. Επιλεγμένα σημεία του σταθμού είναι σημειωμένα μ' ένα ερωτηματικό και, ακουμπώντας τα, ο χρήστης μπορεί να πάρει παραπάνω πληροφορίες. Η συγκεκριμένη εφαρμογή περιλαμβάνεται στα εκπαιδευτικά προγράμματα της NASA για μαθητές γυμνασίου.

Το τελευταίο παράδειγμα από το χώρο της εκπαίδευσης όπου χρησιμοποιήθηκε ένα παιχνίδι, και συγκεκριμένα το Unreal Tournament, αφορά τη διδασκαλία εννοιών της Φυσικής σε μαθητές του γυμνασίου [120]. Δεν πρόκειται για την κατασκευή ενός και μόνο εικονικού κόσμου, αλλά αρκετών, όπου ο καθένας επιτρέπει σε μαθητές και καθηγητές να εκτελούν πειράματα που έχουν σκοπό να διδάξουν έννοιες όπως η βαρύτητα, η διατήρηση της ενέργειας, η αδράνεια, η ταλάντωση, οι δυνάμεις, κτλ. Όπως και στα παραπάνω παραδείγματα, είναι δυνατή η συνύπαρξη αρκετών χρηστών στην εφαρμογή, έτσι ώστε να ενισχυθεί η συνεργασία και η αλληλεπίδραση μεταξύ των μαθητών, αλλά και μεταξύ του καθηγητή και των μαθητών.

5.3 Πιθανές επιδράσεις των παιχνιδιών βολών πρώτου προσώπου στα παιδιά

Σχεδόν από την πρώτη ημέρα της εμφάνισής τους, τα ηλεκτρονικά παιχνίδια και ειδικότερα τα παιχνίδια βολών πρώτου προσώπου δέχονται σοβαρές επικρίσεις για τις

πιθανές επιπτώσεις που έχουν στα παιδιά. Η κύρια αιτία των επικρίσεων αυτών είναι η βία, που αποτελεί το κύριο συστατικό τους. Ένας μεγάλος αριθμός μελετών παρουσιάζει στοιχεία που συνδέουν το παίξιμο βίαιων ηλεκτρονικών παιχνιδιών με την αύξηση της βίαιης συμπεριφοράς στα παιδιά, είτε ως βραχυπρόθεσμη είτε ως μακροπρόθεσμη επίπτωση (ενδεικτικά [121, 122, 123]).

Υπάρχει βέβαια και ισχυρός αντίλογος, που στηρίζεται μάλιστα στη μετα-ανάλυση όλων των σχετικών ερευνών, που υποστηρίζει ότι τα παραπάνω συμπεράσματα δεν είναι και τόσο βάσιμα, κυρίως λόγω των μικρών δειγμάτων που χρησιμοποίησαν οι έρευνες και των μεθοδολογικών αδυναμιών τους [124, 125]. Σε κάθε περίπτωση, η βία από τα παιχνίδια βολών πρώτου προσώπου είναι στοιχείο που μπορεί να αφαιρεθεί.

Δυστυχώς, στη σχετική βιβλιογραφία υπάρχει η τάση το βάρος να δίνεται στις επιπτώσεις του βίαιου περιεχομένου στα παιδιά και το πιο σημαντικό είναι ότι δεν υπάρχει διαχωρισμός των διαφόρων ειδών ηλεκτρονικών παιχνιδιών, όλα τα είδη καλύπτονται από τον ενιαίο χαρακτηρισμό *βιντεοπαιχνίδια* (*video games*). Έτσι είναι εξαιρετικά δύσκολο να διακρίνει κάποιος αν τα αποτελέσματα μίας έρευνας αφορούν ή όχι τα παιχνίδια βολών πρώτου προσώπου. Διατηρώντας την παραπάνω επιφύλαξη, παρουσιάζονται ορισμένες μελέτες με ενδιαφέροντα συμπεράσματα.

Ένα πρώτο συμπέρασμα είναι ότι τα *βιντεοπαιχνίδια* μπορούν να βελτιώσουν τις δεξιότητες που σχετίζονται με την αντίληψη του χώρου [126] και την ικανότητα να μοιράζεται η προσοχή σε περισσότερα του ενός θέματα [127, 128]. Όσον αφορά την αντίληψη του χώρου, φαίνεται ότι μπορούν επίσης να μειώσουν τη διαφορά που υπάρχει στις επιδόσεις μεταξύ αγοριών και κοριτσιών και τα καλύτερα αποτελέσματα επιτυγχάνονται όταν τα παιδιά δεν έχουν τόσο ανεπτυγμένη αυτή τη δεξιότητα [126].

Σε μελέτη για τον τρόπο που τα βίαια *βιντεοπαιχνίδια* επηρεάζουν τη συμπεριφορά των παιδιών, γίνεται η υπόθεση ότι ο μηχανισμός των *σεναρίων* μπορεί να δώσει μία εξήγηση [128]. Ο παίκτης για να κερδίσει στα βίαια *βιντεοπαιχνίδια* πρέπει να υιοθετήσει μία βίαιη συμπεριφορά απέναντι στους εικονικούς αντιπάλους του. Αυτή η βίαιη συμπεριφορά οδηγεί στην επιβράβευσή του, που έχει τη μορφή υψηλής βαθμολογίας ή το πέραςμα σε επόμενο επίπεδο ή τελικά τη νίκη στο παιχνίδι.

Το ίδιο επαναλαμβάνεται με κάθε νέο βίαιο *βιντεοπαιχνίδι* που παίζει ο παίκτης. Αυτό το σχήμα από μόνο του είναι μία πολύ αποτελεσματική “διδακτική” μέθοδος. Συνεχής επανάληψη αυτής της συμπεριφοράς, σε συνδυασμό με την αληθοφάνεια ορισμένων παιχνιδιών, μπορεί τελικά να οδηγήσει στη σύγχυση των *σεναρίων* που αναπτύχθηκαν για να περιγράψουν συμπεριφορές στο παιχνίδι, με *σενάρια* που περιγράφουν

συμπεριφορές στον πραγματικό κόσμο. Αν όντως ισχύει κάτι τέτοιο, τότε δεν υπάρχει λόγος να μην χρησιμοποιηθούν τα *βιντεοπαιχνίδια* –απαλλαγμένα από βίαια στοιχεία– για πρακτική εξάσκηση των παιδιών σε διάφορες δεξιότητες.

Ένα τελευταίο θέμα που παρουσιάζει ενδιαφέρον είναι ο τρόπος που ερμηνεύεται η απορρόφηση των παιδιών όταν παίζουν βιντεοπαιχνίδια. Αυτή η *ψυχολογική απορρόφηση*, που θεωρείται από ορισμένους ένας τύπος διαφοροποιημένης κατάστασης της συνείδησης, εμφανίζεται όταν κάποιος είναι εμβυθισμένος στην εμπειρία που βιώνει [129] και είναι μία μορφή *μη παθολογικής αποσύνδεσης* (*non-pathological dissociation*). Η *μη παθολογική αποσύνδεση* είναι αρκετά συχνή στα παιδιά (εκδηλώνεται με ονειροπόληση και φαντασιώσεις) και κορυφώνεται περίπου στην ηλικία των 9-10 ετών [130].

Σ' αυτή την κατάσταση αναστέλλεται η *λογική εννοποίηση* (*logical integration*) σκέψεων, συναισθημάτων και εμπειριών, σαν να παύει να υπάρχει η ίδια η πραγματικότητα [131]. Όταν η εμπειρία που βιώνει ο παίκτης από το παιχνίδι είναι θετική, μοιάζει με την *κατάσταση ροής* (*flow state*) που έχει περιγραφεί από τους Csikszentmihalyi και Csikszentmihalyi [132], που συνδέεται με την *ενισχυμένη μάθηση* (*enhanced learning*) [133]. Φαίνεται λοιπόν ότι η εμβύθιση, που είναι σημαντικό στοιχείο της Ε.Π., έχει κοινά στοιχεία –αν όχι ταύτιση– με την απορρόφηση των παιδιών όταν παίζουν *βιντεοπαιχνίδια*.

Αυτό λοιπόν που μπορεί να ειπωθεί με σχετική βεβαιότητα είναι ότι οι ψυχολογικοί και γνωστικοί μηχανισμοί που διέπουν τα παιχνίδια βολών πρώτου προσώπου έχουν σημαντικές ομοιότητες με τους αντίστοιχους μηχανισμούς της Ε.Π., πράγμα που συνηγορεί υπέρ της χρήσης τους σε αντικατάσταση των εξειδικευμένων προγραμμάτων κατασκευής εφαρμογών Ε.Π.

Κεφάλαιο 6^ο

Οι μηχανές παιχνιδιών

Στα προηγούμενα κεφάλαια αναλύθηκαν τα παιχνίδια, οι ομοιότητες και οι διαφορές τους με τις εφαρμογές Ε.Π. και φάνηκε ότι μπορούν να προσαρμοστούν και να τροποποιηθούν ώστε να ικανοποιούν απαιτήσεις διαφορετικές από τις αρχικές, ταυτιζόμενα με εφαρμογές της Ε.Π. Φάνηκε επίσης ότι οι επικρίσεις για τις επιδράσεις τους στα παιδιά περιστρέφονται γύρω από το στοιχείο της βίας, που προφανώς δεν αφορά εκπαιδευτικές εφαρμογές, και ότι υπάρχουν ομοιότητες με την Ε.Π. στο ψυχολογικό και στο γνωστικό επίπεδο.

Όμως δεν πρέπει να διαφεύγει το γεγονός ότι τα παιχνίδια είναι το τελικό προϊόν, η τελική εφαρμογή. Πώς λοιπόν κατασκευάζονται τα παιχνίδια; Η απάντηση είναι ότι και αυτά αναπτύσσονται με ειδικά και με βοηθητικά προγράμματα. Στη συνέχεια εξετάζονται τα προγράμματα κατασκευής τρισδιάστατων παιχνιδιών, οι δυνατότητες και τα συστατικά τους στοιχεία.

Ξεκινώντας από τα βοηθητικά προγράμματα, αρκεί να ειπωθεί ότι είναι ακριβώς τα ίδια με αυτά που χρησιμοποιούνται για την ανάπτυξη εφαρμογών Ε.Π., δηλαδή προγράμματα κατασκευής και επεξεργασίας τρισδιάστατων γραφικών, επεξεργασίας ήχου, μουσικής, γραφικών και εικόνων. Τα προγράμματα κατασκευής παιχνιδιών διαθέτουν και αυτά, όπως και τα προγράμματα κατασκευής εικονικών κόσμων, φίλτρα για την εισαγωγή και σωστή διαχείριση αρχείων που προέρχονται από τα βοηθητικά προγράμματα.

Υπάρχουν αρκετά προγράμματα που ο σκοπός τους είναι να παρέχουν ένα ολοκληρωμένο περιβάλλον κατασκευής παιχνιδιών (ενδεικτικά [134, 135, 136]). Το κόστος τους είναι αρκετά χαμηλό, αλλά το κύριο αρνητικό τους σημείο είναι ότι τα παιχνίδια που παράγονται με αυτά υστερούν αρκετά σε ποιότητα και επιδόσεις από τα παιχνίδια που κυκλοφορούν στο εμπόριο. Αναφέρθηκε ότι σε κάποιες περιπτώσεις το πρόγραμμα κατασκευής και το παιχνίδι αποτελούν ενιαίο και αναπόσπαστο σύνολο. Σ' αυτή την περίπτωση μάλιστα, οι κατασκευάστριες εταιρείες δίνουν υπό προϋποθέσεις (όχι για εμπορική εκμετάλλευση) το δικαίωμα στον αγοραστή να χρησιμοποιήσει το *πρόγραμμα κατασκευής (editor)*, για να φτιάξει δικά του επίπεδα στο αντίστοιχο παιχνίδι.

Καθαρά εμπορικοί λόγοι επιβάλλουν αυτή την τακτική. Τα παιχνίδια βολών πρώτου προσώπου, αλλά και άλλα είδη παιχνιδιών, με την ολοκλήρωσή τους είναι ουσιαστικά άχρηστα στον αγοραστή και χάνουν το ενδιαφέρον τους. Φτιάχνοντας όμως ο παίκτης δικά του επίπεδα και παίζοντας επίπεδα που άλλοι παίκτες έφτιαξαν, το ενδιαφέρον για το παιχνίδι διατηρείται, αυξάνει ο χρόνος παρουσίας του παιχνιδιού στην αγορά, με συνέπεια να αυξάνονται και οι πωλήσεις.

Το *πρόγραμμα κατασκευής* δίνει πρόσβαση στην καρδιά του κάθε παιχνιδιού, που ονομάζεται *μηχανή παιχνιδιού* (*Game Engine*). Η *μηχανή παιχνιδιού* και το παιχνίδι είναι δύο διαφορετικές έννοιες και δεν πρέπει να συγχέονται. Η *μηχανή παιχνιδιού* αντιστοιχεί με τη μηχανή ενός αυτοκινήτου, ενώ το σασί του είναι το παιχνίδι. Η αναλογία αυτή δίνει αρκετά παραστατικά και τη σημασία της *μηχανής παιχνιδιού*, που είναι η κινητήρια δύναμη πίσω από κάθε παιχνίδι. Ο χρήστης, ή καλύτερα πια ο σχεδιαστής, χρησιμοποιώντας το *πρόγραμμα κατασκευής* έχει άμεση πρόσβαση στη *μηχανή του παιχνιδιού*, κατασκευάζοντας –μέσα στα πλαίσια των δυνατοτήτων της *μηχανής παιχνιδιού*– ό,τι ο ίδιος επιθυμεί, χτίζοντας ένα νέο παιχνίδι ή μία νέα εφαρμογή.

Σε πολλά σημεία οι *μηχανές παιχνιδιών* προσφέρουν πολύ περισσότερες δυνατότητες απ' ό,τι τα προγράμματα κατασκευής εικονικών κόσμων, γιατί καλύπτουν τομείς που δεν είναι απαραίτητο να υπάρχουν σε μία τυπική εφαρμογή Ε.Π., όπως για παράδειγμα τη διαχείριση της συμπεριφοράς των διαφόρων όπλων. Αυτό δε σημαίνει ότι και μ' ένα πρόγραμμα Ε.Π. δεν μπορεί να υλοποιηθεί κάτι τέτοιο, αλλά η δυσκολία είναι μεγαλύτερη και τις περισσότερες φορές χρειάζεται προγραμματιστική παρέμβαση.

Ένα άλλο σημείο που δεν πρέπει να διαφεύγει της προσοχής είναι η πολύ καλύτερη διαχείριση του υλικού από τις *μηχανές παιχνιδιών*, γιατί απευθύνονται σε απλούς χρήστες. Οι *μηχανές παιχνιδιών* το καταφέρνουν αυτό υλοποιώντας τεχνικές αποφυγής της σημαντικής επιβάρυνσης του υλικού, που δε συναντώνται συχνά στα αντίστοιχα προγράμματα κατασκευής εικονικών κόσμων.

Τα σημαντικότερα μέρη μίας *μηχανής παιχνιδιού* είναι [87]:

- Διαχείριση ήχου
- Δικτυακή υποστήριξη
- Τεχνητή νοημοσύνη
- Διαχείριση μοντέλων, υφών και κίνησης
- Το περιβάλλον διεπαφής
- Διαχείριση όπλων
- Διαχείριση συμβάντων-scripts

6.1 Διαχείριση ήχου

Ο ήχος στα παιχνίδια είναι μία αρκετά σημαντική παράμετρος και γι' αυτό οι *μηχανές παιχνιδιών* διαχειρίζονται αρκετές καταστάσεις που αφορούν ήχο και μουσική, εκμεταλλευόμενες όλες τις δυνατότητες των σύγχρονων καρτών ήχου. Οι πιο σημαντικές είναι η παραγωγή ήχων που να φαίνεται ότι φράζονται από κάποιο εμπόδιο, η παραγωγή ήχων που να φαίνεται ότι προέρχονται από διάφορα σημεία του χώρου και η *δυναμική μουσική*.

Η φραγή του ήχου μπορεί να περιλαμβάνει δύο διαφορετικές καταστάσεις. Στην πρώτη, το εμπόδιο περιβάλλει την πηγή του ήχου, όπως ήχοι που προέρχονται μέσα από ένα σπίτι ενώ ο παίκτης είναι έξω, δηλαδή οι ήχοι ακούγονται “πνιχτοί”. Στη δεύτερη, μεσολαβεί ένα εμπόδιο μεταξύ της πηγής και του παίκτη, οι ήχοι ακούγονται όχι απευθείας, αλλά από αντηχήσεις. Οι ήχοι επίσης εξαρτώνται και από το υλικό στο οποίο διαδίδονται. Οι καλές *μηχανές παιχνιδιών* δίνουν τη δυνατότητα να αλλοιώνονται οι ήχοι κάτω από το νερό ή σ' ένα χώρο καλυμμένο με ηχομονωτικό υλικό. Επίσης, διαφοροποιούν τους ήχους όταν ο χώρος είναι ανοιχτός, μεγάλος ή μικρός. Αυτό το χαρακτηριστικό είναι ιδιαίτερα χρήσιμο, γιατί μπορεί να υλοποιηθούν σενάρια όπου το παιδί θα πρέπει να ακούσει, να διακρίνει και να αξιολογήσει έναν ήχο.

Ο παίκτης κινείται μέσα σ' έναν τρισδιάστατο χώρο, μεταβάλλοντας συνεχώς τη σχετική του θέση από την πηγή ή τις πηγές των ήχων. Οι *μηχανές παιχνιδιών* μπορούν και μεταβάλλουν την ένταση του ήχου μίας πηγής όσο ο παίκτης πλησιάζει ή απομακρύνεται απ' αυτή. Επίσης, μπορούν και μεταβάλλουν την κατεύθυνση προέλευσης του ήχου ανάλογα με το πού είναι στραμμένος ο παίκτης.

Η *δυναμική μουσική* είναι η δυνατότητα να αλλάζει το τέμπο της μουσικής ή να αλλάζει το κομμάτι που παίζει ανάλογα με το τι συμβαίνει στο παιχνίδι. Για παράδειγμα, μία αργή μουσική παίζει όταν δεν υπάρχει δράση στο παιχνίδι, γίνεται γρήγορη στη δράση και γυρνάει ξανά σε αργή όταν τελειώσει η δράση. Τέτοιου είδους εναλλαγές συμβάλουν στη δημιουργία της κατάλληλης ατμόσφαιρας.

Συνήθως τα προγράμματα κατασκευής εικονικών κόσμων χρησιμοποιούν ήχους σε μορφή wave (.wav) που, επειδή δεν είναι συμπιεσμένοι, καταλαμβάνουν χώρο και είναι αρκετά απαιτητικοί σε επεξεργαστική ισχύ. Αντίθετα, οι περισσότερες *μηχανές παιχνιδιών* χρησιμοποιούν ήχους σε μορφή mp3, που είναι συμπιεσμένοι σε μία αναλογία 11:1 και χρειάζονται ελάχιστο χρόνο και επεξεργαστική ισχύ για να αποσυμπιεστούν και να παιχτούν από την κάρτα ήχου.

6.2 Δικτυακή υποστήριξη

Τα περισσότερα, αν όχι όλα τα παιχνίδια βολών πρώτου προσώπου, δίνουν τη δυνατότητα να παίζουν και να συνομιλούν γραπτώς πολλοί παίκτες ταυτόχρονα, μέσω δικτύου ή μέσω του Διαδικτύου (*multiplayer gaming*). Είναι ένα στοιχείο που κρατά το ενδιαφέρον για το παιχνίδι σε υψηλά επίπεδα και αυξάνει τις πωλήσεις του. Για να επιτευχθεί η σύνδεση πολλών παικτών, χρησιμοποιούνται από τη *μηχανή παιχνιδιού* τρία πολύ κοινά πρωτόκολλα επικοινωνίας, το TCP (*Transmission Control Protocol*), το UDP (*User Datagram Protocol*) και το IP (*Internet Protocol*).

Το IP είναι το πρωτόκολλο που αναλαμβάνει τη διακίνηση πακέτων δεδομένων του *TCP* ή του *UDP* στο Διαδίκτυο. Όμως τα πρωτόκολλα *TCP* και *UDP* έχουν μία πολύ σημαντική διαφορά μεταξύ τους. Το *TCP* εγγυάται την παράδοση των πακέτων με τη σειρά που στάλθηκαν, ενώ το *UDP* δεν την εγγυάται [137]. Για να εγγυηθεί την παράδοση των πακέτων, το *TCP* απαιτεί από τον παραλήπτη να στέλνει στον αποστολέα μία απόδειξη παραλαβής (*Acknowledgement*, *ACK*) για κάθε πακέτο που λαμβάνει.

Αν το παιχνίδι παίζεται σε τοπικό δίκτυο, λόγω των υψηλών ταχυτήτων δε δημιουργούνται προβλήματα και η όλη διαδικασία περνάει εντελώς απαρατήρητη. Στο Διαδίκτυο όμως, όπου οι ταχύτητες είναι πολύ μικρότερες, η επιπλέον κίνηση που δημιουργεί το *TCP* μειώνει ακόμα παραπάνω την ταχύτητα. Στην περίπτωση μάλιστα απώλειας πακέτων, που είναι συχνό φαινόμενο, το παιχνίδι “παγώνει” σ’ όλους τους παίκτες, μέχρις ότου αποκατασταθεί η λήψη τους. Αυτός είναι και ο κύριος λόγος για τον οποίο οι *μηχανές παιχνιδιών* χρησιμοποιούν το *UDP*.

Επισημάναμε όμως ότι το *UDP* δεν εγγυάται τη λήψη των πακέτων. Έτσι, χρειάζεται ένας μηχανισμός που να μπορεί να αντιμετωπίζει “έξυπνα” την απώλειά τους και αυτός ο μηχανισμός είναι η *πρόβλεψη* (*Client Prediction*) [87]. Η *πρόβλεψη* ενεργοποιείται όταν ένας υπολογιστής δε λαμβάνει πακέτα από τους υπόλοιπους για κάποιο διάστημα. Τότε “υποθέτει” τι μπορεί να κάνουν οι άλλοι παίκτες και το παρουσιάζει στον παίκτη που έχει πρόβλημα στη σύνδεση.

Όταν αποκατασταθεί η σύνδεση, παίρνει δεδομένα από τους άλλους παίκτες και γίνεται διόρθωση σε ό,τι παρουσιαζόταν στην οθόνη μέχρι εκείνη τη στιγμή. Η όλη διαδικασία λειτουργεί με αρκετή επιτυχία, γιατί συνήθως οι διακοπές στη μετάδοση πακέτων διαρκούν κλάσματα του δευτερολέπτου. Όταν υπάρχει διακοπή για μεγαλύτερο διάστημα και η *πρόβλεψη* δεν μπορεί να λειτουργήσει, τότε το παιχνίδι “παγώνει” στη μεριά του παίκτη που έχει το πρόβλημα.

Αναφέρθηκε ήδη ότι τα παιχνίδια διακινούν στο δίκτυο ελάχιστο ποσό δεδομένων, στέλλοντας μόνο τις συντεταγμένες της θέσης του κάθε παίκτη. Εννοείται βέβαια ότι στέλλονται και άλλες πληροφορίες που αφορούν τις ενέργειές του και την αλληλεπίδραση με τα αντικείμενα και τους άλλους παίκτες. Για να μειωθεί ακόμα περισσότερο ο όγκος των δεδομένων, οι *μηχανές παιχνιδιών* δε στέλνουν πληροφορίες για ό,τι συμβαίνει έξω από το οπτικό πεδίο του κάθε παίκτη και δε στέλνουν πληροφορίες αν τα δεδομένα μεταξύ δύο καρέ δεν έχουν αλλάξει.

6.3 Τεχνητή νοημοσύνη

Η τεχνητή νοημοσύνη στις *μηχανές παιχνιδιών* αφορά τη συμπεριφορά των αντικειμένων και των χαρακτήρων που δεν ελέγχονται από το χρήστη, αλλά που μπορούν να αλληλεπιδράσουν με αυτόν. Στα προγράμματα κατασκευής εικονικών κόσμων αυτό γίνεται αναγκαστικά με προγραμματισμό και τη χρήση *scripts*. Οι *μηχανές παιχνιδιών* μπορούν να αντιμετωπίσουν ιδιαίτερα σύνθετες καταστάσεις και με ευκολότερο τρόπο, αυξάνοντας σε μεγάλο βαθμό την αληθοφάνεια του παιχνιδιού.

Ένα απλουστευμένο σενάριο χρήσης της τεχνητής νοημοσύνης είναι ο τρόπος που συμπεριφέρονται οι “αντίπαλοι” του παίκτη. Πρέπει πρώτα να καθοριστεί τι κάνουν στο σημείο όπου βρίσκονται, για παράδειγμα αν περιπολούν σε μία μικρή περιοχή, αν απλώς στέκονται, αν συζητούν, κτλ. Αφού καθοριστεί αυτό, πρέπει να αποκτήσουν “συνείδηση” του χώρου που τους περιβάλλει, για παράδειγμα μέχρι πού βλέπουν και ακούν. Έτσι μπορούν να αντιδράσουν όταν ο παίκτης μπει μέσα στο οπτικό ή ακουστικό τους πεδίο. Αν συμβεί αυτό, θα πρέπει να καθοριστεί η αντίδρασή τους, για παράδειγμα αν θα επιτεθούν στον παίκτη, αν θα τον αγνοήσουν ή αν θα περιμένουν μέχρι να τους επιτεθεί εκείνος. Όλες αυτές οι “συμπεριφορές” στις *μηχανές παιχνιδιών* υλοποιούνται με απλές επιλογές από τα αντίστοιχα μενού και δε χρειάζονται προγραμματιστική παρέμβαση από το σχεδιαστή.

6.4 Διαχείριση συμβάντων-scripts

Σε θέματα που δεν καλύπτονται από την τεχνητή νοημοσύνη, δίνεται η δυνατότητα στο σχεδιαστή να χρησιμοποιήσει *scripts* για να υλοποιήσει προγραμματιστικά ένα γεγονός ή μία αλληλουχία γεγονότων. Συνήθως στις *μηχανές παιχνιδιών* συνυπάρχουν δύο συστήματα *scripts*. Ένα είναι ακριβώς ίδιο με τα προγράμματα κατασκευής εικονικών κόσμων. Ο σχεδιαστής γράφει το κώδικα, ο οποίος στη συνέχεια ελέγχεται για την ορθότητά του και εκτελείται.

Το δεύτερο μοιάζει πολύ με το σύστημα οπτικού προγραμματισμού. Τοποθετούνται δηλαδή στον κόσμο αντικείμενα, αόρατα στον παίκτη, που αν κάνει κάποια ενέργεια ή τα πλησιάσει, ενεργοποιούν με τη σειρά τους κάποια διαδοχή γεγονότων. Είναι οι *σκανδάλες* και τα *γεγονότα*, για τα οποία έγινε αναφορά στο προηγούμενο κεφάλαιο. Το σύστημα αυτό απαλλάσσει το σχεδιαστή από το γράψιμο κώδικα και ταυτόχρονα μπορεί να ελέγξει άμεσα το αποτέλεσμα. Ένα παράδειγμα χρήσης αυτού του συστήματος θα μπορούσε να είναι η παροχή οδηγιών και συμβουλών αν το παιδί χρησιμοποιήσει ένα αντικείμενο ή αν μπει μέσα σε μία προκαθορισμένη περιοχή.

6.5 Το περιβάλλον διεπαφής

Το περιβάλλον διεπαφής για τα παιχνίδια είναι οι διάφορες πληροφορίες που παρουσιάζονται στον παίκτη όση ώρα παίζει. Για παράδειγμα, γνωρίζει πόση “ζωή” του απομένει, τι πυρομαχικά έχει διαθέσιμα, κτλ. Έχουν τη μορφή διαφόρων εικονιδίων στην κεντρική οθόνη του παιχνιδιού και ο όρος που χρησιμοποιείται γι’ αυτές είναι *heads-up display (HUD)*. Μία οθόνη γεμάτη από πληροφορίες δυσκολεύει ιδιαίτερα τον παίκτη και, από την άλλη, η απουσία πληροφοριών δεν του δίνει στοιχεία απαραίτητα για τη συνέχεια του παιχνιδιού. Στις *μηχανές παιχνιδιών* το *HUD* είναι ενσωματωμένο στοιχείο, δε χρειάζεται προσπάθεια για να υλοποιηθεί και σε αρκετές περιπτώσεις είναι παραμετροποιήσιμο. Για παράδειγμα, μπορεί να αλλάξει ο βαθμός διαφάνειάς του, να προστεθούν ή να αφαιρεθούν στοιχεία.

6.6 Διαχείριση όπλων

Η βία και τα όπλα στα παιχνίδια είναι μεν αξιοκατάκριτα από κάθε άποψη, αλλά αντιμετωπίζοντάς τα από την τεχνική-προγραμματιστική σκοπιά είναι μία ιδιαίτερη πρόκληση και αποτελούν ξεχωριστό κομμάτι μίας *μηχανής παιχνιδιού*. Για να τεθεί το θέμα σε μία ευρύτερη βάση, τα όπλα είναι αντικείμενα που επηρεάζουν από απόσταση άλλα αντικείμενα με συγκεκριμένους τρόπους, προκαλώντας συγκεκριμένες αντιδράσεις. Γενικά υπάρχουν τρία είδη όπλων: οι “σφαίρες”, που ταξιδεύουν αυτόματα από ένα σημείο σ’ ένα άλλο, οι “πύραυλοι”, που χρειάζονται κάποιο χρόνο για να φτάσουν στο στόχο, και τα όπλα, που προκαλούν “έκρηξη”, που επηρεάζουν δηλαδή και την περιοχή γύρω από το στόχο.

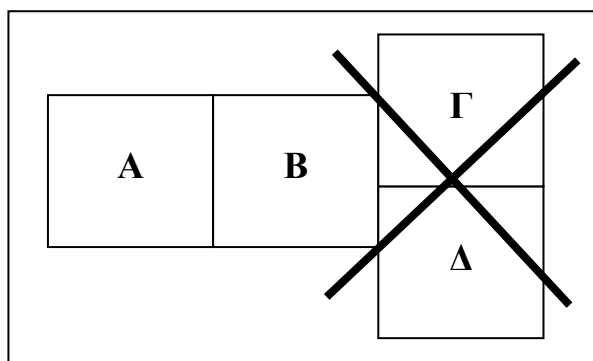
Η ανίχνευση από τη *μηχανή παιχνιδιού* της τροχιάς του κάθε όπλου και της περιοχής στην οποία επιδρά, απαιτεί αρκετούς υπολογισμούς. Για παράδειγμα, πρέπει να ανιχνευτεί αν από το σημείο Α μέχρι το σημείο Β μεσολαβούν άλλα πολύγωνα, ποια επίδραση θα έχουν στην τροχιά και πώς θα επηρεαστούν τα ίδια. Το στοιχείο όμως που ενδιαφέρει είναι ότι, εφόσον τα όπλα αποτελούν ανεξάρτητο στοιχείο της μηχανής παιχνιδιού, μπορούν κάλλιστα να μη χρησιμοποιηθούν.

6.7 Διαχείριση μοντέλων, υφών και κίνησης

Η καρδιά μίας *μηχανής παιχνιδιού* είναι ο τρόπος που διαχειρίζεται τα τρισδιάστατα μοντέλα, τις υφές, τις φωτοσκιάσεις και την κίνηση. Το θέμα είναι ιδιαίτερα περίπλοκο και δεν είναι σκόπιμο να αναλυθεί σε βάθος. Ενδεικτικά παρουσιάζονται μερικές τεχνικές, ώστε να εξηγηθούν οι τρόποι με τους οποίους μία *μηχανή παιχνιδιού* μπορεί να αξιοποιήσει πολύ καλά όλες τις δυνατότητες του υλικού.

Αναφέρθηκε ότι τα μοντέλα αποτελούνται από πολύγωνα, που με τη σειρά τους αποτελούνται από σημεία. Συχνά ο εικονικός κόσμος φτάνει να περιέχει έναν αριθμό πο-

λυγώνων που καμία κάρτα γραφικών δεν μπορεί να χειριστεί και χρειάζεται να μειωθεί ο αριθμός τους. Η πρώτη τεχνική που χρησιμοποιείται είναι η αφαίρεση εκείνων των πολυγώνων που δεν μπορεί να είναι ορατά από το χρήστη, με μία μέθοδο που ονομάζεται *culling* [87]. Στην πιο απλή μορφή της χωρίζεται ο κόσμος σε *τομείς* (*sectors*) και καθορίζεται ποιοι *τομείς* είναι ορατοί από κάθε δεδομένο *τομέα*. Για παράδειγμα, αν ο χρήστης βρίσκεται στο *τομέα* Α και έχει καθοριστεί ότι από τη θέση που βρίσκεται ο μόνος άλλος ορατός *τομέας* εκτός από τον Α είναι ο Β, δεν υπάρχει λόγος να σχεδιάζονται οι *τομείς* Γ και Δ (σχήμα 6.1).



Σχήμα 6.1 Παράδειγμα χρήσης τομέων

Μία δεύτερη τεχνική είναι τα *επίπεδα λεπτομέρειας* (*Level of Detail, L.O.D.*) στα μοντέλα. Η βασική ιδέα είναι ότι δεν έχει νόημα ένα μοντέλο σε μεγάλη απόσταση από το χρήστη να χρησιμοποιεί τον ίδιο αριθμό πολυγώνων σε σχέση με όταν είναι κοντά του, γιατί ούτως ή άλλως δε φαίνεται καθαρά. Στις περισσότερες *μηχανές παιχνιδιών* ο σχεδιαστής θα πρέπει να προκατασκευάσει το ίδιο μοντέλο με διαφορετικό αριθμό πολυγώνων και να καθορίσει στη συνέχεια σε τι αποστάσεις θα παρουσιάζεται το καθένα [87].

Μία τρίτη τεχνική είναι η περιγραφή του μοντέλου με κάποιο *μαθηματικό τύπο* (*Higher-Order Surfaces*) και όχι με συντεταγμένες σημείων και πολυγώνων. Με τον τρόπο αυτό είναι δυνατή η επιλογή του αριθμού των πολυγώνων που θα συνθέτουν το μοντέλο σε κάθε δεδομένη χρονική στιγμή. Τέλος, υπάρχει η τεχνική *ελέγχου του βάθους* (*Depth Testing*). Με αυτή ελέγχεται ποια αντικείμενα είναι μπροστά από άλλα, καταλαμβάνοντας τα ίδια εικονοστοιχεία [87]. Έτσι, η *μηχανή παιχνιδιού* σχεδιάζει πρώτα τα μπροστινά αντικείμενα και δε σχεδιάζει καθόλου όσα είναι από πίσω τους, λαμβάνοντας υπόψη την ύπαρξη διαφανών αντικειμένων.

Οι υφές, επειδή χρησιμοποιούν μεγάλα ποσά μνήμης, αποτελούν αντικείμενο ιδιαίτερου χειρισμού από τις μηχανές παιχνιδιών. Ένας τρόπος είναι η συμπίεσή τους, διευκολύνοντας έτσι τη διακίνησή τους από και προς τη μνήμη της κάρτας γραφικών, με αποτέλεσμα να επιταχύνεται η διαδικασία. Επίσης είναι δυνατό να ελέγχεται η προσωρινή μνήμη της κάρτας, ώστε να μην φορτώνεται σ' αυτή κάποια υφή παρά μόνο μία φορά. Οι υφές έχουν συνήθως χρωματικό βάθος 32bit. Υποβαθμίζοντας όμως επιλεκτικά το χρωματικό βάθος στα 16bit, επιτυγχάνεται ακόμα μεγαλύτερη μείωση του μεγέθους τους και έτσι καταλαμβάνουν λιγότερο χώρο στη μνήμη.

Μία τελευταία τεχνική που εφαρμόζεται στις υφές μοιάζει με τα *επίπεδα λεπτομέρειας* που χρησιμοποιούνται στα μοντέλα και ονομάζεται *Mip Mapping* [87]. Ο σχεδιαστής δημιουργεί διαφορετικές εκδοχές της ίδιας υφής που “ντύνουν” ένα μοντέλο χαμηλής ή υψηλής ανάλυσης. Οι υφές με μεγάλη ανάλυση χρησιμοποιούνται στα κοντινά μοντέλα και της χαμηλής ανάλυσης στα μακρινά.

Τα τελευταία χρόνια και λόγω της εξέλιξης των καρτών γραφικών, είναι δυνατό να συνυπάρχουν σ' ένα αντικείμενο πολλαπλά επίπεδα υφών. Αυτό αυξάνει κατά πολύ την αληθοφάνεια των μοντέλων, αλλά απαιτεί και αρκετή επεξεργαστική ισχύ. Ένα χαρακτηριστικό παράδειγμα χρήσης πολλαπλών υφών είναι το *Bump Mapping*, όπου σε δεύτερο επίπεδο τοποθετείται μία υφή που δίνει την αίσθηση του ανάγλυφου στο αντικείμενο, αλλάζοντας μάλιστα ανάλογα με την οπτική γωνία που το παρατηρεί ο χρήστης [87]. Οι σύγχρονες *μηχανές παιχνιδιών* επιτρέπουν στο σχεδιαστή την επιλογή χρήσης πολλαπλών επιπέδων υφών, ανάλογα με τις επιδόσεις του υλικού όπου θα εκτελεστεί η εφαρμογή του.

Οι *μηχανές παιχνιδιών* χρησιμοποιούν διάφορες τεχνικές για να δώσουν κίνηση στα μοντέλα. Αρκετά συνηθισμένη είναι η χρήση *κοκάλων και αρθρώσεων* (*skeletal animation*). Εκτός του ότι η μέθοδος αυτή δίνει πιο ομαλή και φυσική κίνηση στα μοντέλα, είναι δυνατό να οριστούν και διαφορετικές κινήσεις που ένα *κόκαλο* ή ομάδα *κοκάλων* μπορεί να κάνει. Αυτό είναι ιδιαίτερα χρήσιμο, γιατί μπορεί να μεταβάλλεται κατά βούληση η κίνηση του μοντέλου. Μπορεί ενώ περπατά να αρχίσει τρέχει ή να σκύψει ή να πηδήξει, χωρίς να χρειάζεται να χρησιμοποιείται κάθε φορά διαφορετικό μοντέλο.

6.8 Σχολιασμός

Συνοψίζοντας, τα κυριότερα τεχνικά σημεία των *μηχανών παιχνιδιών* που τις κάνουν κατάλληλες για τη χρήση τους σε εφαρμογές Ε.Π. είναι:

1. Οι σχετικά χαμηλές προδιαγραφές υλικού που απαιτούνται. Αυτό γιατί απευθύνονται στο ευρύ κοινό και γιατί οι σημερινοί ηλεκτρονικοί υπολογιστές διαθέτουν αρκετή υπολογιστική ισχύ.

2. Επειδή σχεδιάστηκαν να ικανοποιούν τις ανάγκες παιχνιδιών με υψηλά ποσά δράσης και αλληλεπιδράσεων, είναι κατάλληλες για τη δημιουργία εξερευνησιμων εικονικών περιβαλλόντων που εκτελούνται ομαλά, λόγω της εκτεταμένης εκμετάλλευσης των δυνατοτήτων των καρτών γραφικών.
3. Η δυνατότητα ταυτόχρονης παρουσίας πλέον του ενός παικτών συμβάλλει στην αύξηση της αλληλεπιδραστικότητας. Η αλληλεπιδραστικότητα αυξάνει ακόμα περισσότερο εξαιτίας της δυνατότητας να συνομιλούν μεταξύ τους.
4. Εξαιτίας της διάδοσης των παιχνιδιών βολών πρώτου προσώπου, πολλοί χρήστες (κυρίως μικρής ηλικίας) είναι εξοικειωμένοι με τα βασικά τους στοιχεία. Συνεπώς, η εξερεύνηση ενός εικονικού περιβάλλοντος φτιαγμένου με μία μηχανή παιχνιδιού μπορεί να θεωρηθεί δραστηριότητα φιλική προς το χρήστη και συνδέεται με την ευχαρίστηση του παιξίματος ενός παιχνιδιού.
5. Η δυνατότητα εμπλουτισμού του εικονικού περιβάλλοντος με διάφορα οπτικά και ηχητικά εφέ αυξάνουν το βαθμό αληθοφάνειας της εφαρμογής.
6. Εμπορικοί λόγοι επιβάλλουν τη διαρκή βελτίωση των δυνατοτήτων τους. Αυτό λειτουργεί και υπέρ του τελικού χρήστη, αλλά και υπέρ του σχεδιαστή, γιατί του δίνονται συνεχώς νέες δυνατότητες.

Υπάρχουν όμως αρνητικά σημεία, τα οποία πρέπει να προσεχθούν ιδιαίτερα. Υπάρχουν *μηχανές παιχνιδιών* που σχεδιάστηκαν και πωλούνται ως ανεξάρτητο εμπορικό προϊόν, που όμως δεν είναι τόσο εξελιγμένες όσο οι *μηχανές παιχνιδιών* των γνωστών –και ευρύτετα διαδεδομένων– παιχνιδιών. Οι αξιόλογες *μηχανές παιχνιδιών* είναι εργαλεία που η κάθε εταιρεία αναπτύσσει για να καλύψει τις δικές της ανάγκες, σύμφωνα με τα δικά της κριτήρια, σ' ένα έντονα ανταγωνιστικό περιβάλλον. Ως εκ τούτου:

1. Είναι μεταξύ τους εντελώς ασύμβατες. Δεν ακολουθούνται κάποια κοινώς αποδεκτά πρότυπα στη μορφή των αρχείων που παράγουν.
2. Ως εργαλεία εσωτερικής και όχι εμπορικής χρήσης, έχουν αρκετές ατέλειες και γενικά είναι προγράμματα αρκετά ασταθή. Αυτό όμως δεν ισχύει για το τελικό προϊόν, το παιχνίδι ή την εφαρμογή που παράγεται με αυτές.
3. Το περιβάλλον εργασίας που παρέχουν δεν είναι απαραίτητα φιλικό και εργονομικό.

6.9 Επιλέγοντας μία μηχανή παιχνιδιού

Έχοντας υπόψη όσα αναφέρθηκαν στη σύγκριση των παιχνιδιών βολών πρώτου προσώπου με τα προγράμματα ανάπτυξης εικονικών κόσμων και την τεχνική ανάλυση των *μηχανών παιχνιδιών*, κάποιος μπορεί να αναζητήσει εκείνη τη *μηχανή παιχνιδιού* που προσφέρει αρκετές δυνατότητες και λύσεις στις ανάγκες κατασκευής εφαρμογών Ε.Π.

Η αναζήτηση αυτή διευκολύνεται σημαντικά από την πληθώρα παιχνιδιών που υπάρχουν στο εμπόριο, που επιτρέπει την άμεση αξιολόγηση των δυνατοτήτων και των περιορισμών της κάθε μηχανής. Αρχικά μπορεί να γίνει ομαδοποίησή τους με βάση τη μηχανή που έχουν κατασκευαστεί. Στη συνέχεια μπορούν να ελεγχθούν ποιοτικά και τεχνικά στοιχεία επιλεγμένων παιχνιδιών, αντιπροσωπευτικών των περισσότερων μηχανών. Τέτοια στοιχεία είναι:

- Η ελάχιστη και η ενδεχόμενη διαμόρφωση του υλικού του χρήστη για την ικανοποιητική εκτέλεση του παιχνιδιού.
- Η δυνατότητα που δίνουν στο χρήστη να κάνει ρυθμίσεις για την καλύτερη αξιοποίηση του υλικού του.
- Η ταχύτητα εκτέλεσης του παιχνιδιού, κάνοντας διαφορετικές ρυθμίσεις που επιδίδωκαν καλύτερη ταχύτητα ή καλύτερη ποιότητα, ή τέλος ισορροπία ποιότητας και ταχύτητας.
- Η γενική εικόνα της ποιότητας των γραφικών, των υφών και της κίνησης των διαφόρων τρισδιάστατων μοντέλων.
- Η δικτυακή υποστήριξη.
- Η ύπαρξη, η πληρότητα και η ευχρηστία των εργαλείων (*editors*) που επιτρέπουν την πρόσβαση στη *μηχανή παιχνιδιού*.
- Η δημοτικότητα του παιχνιδιού, οι ομάδες συζητήσεων γι' αυτό, η υποστήριξη που παρέχεται.
- Εξωτερικά-βοηθητικά προγράμματα που συνεργάζονται με τη μηχανή, τα φίλτρα εισαγωγής-εξαγωγής αρχείων.

Αυτό που διαπιστώνεται είναι ότι, αντίθετα με την ένδεια που επικρατεί στα προγράμματα κατασκευής εικονικών κόσμων που περιορίζει τις δυνατότητες επιλογής, υπάρχει μία πληθώρα σχεδόν ισοδύναμων επιλογών όσον αφορά τις *μηχανές παιχνιδιών*. Μία δεύτερη διαπίστωση που προκύπτει μελετώντας και αξιολογώντας τα παιχνίδια είναι ότι πράγματι οι *μηχανές παιχνιδιών* αξιοποιούν σχεδόν άριστα το υλικό και ότι η ποιότητα των τρισδιάστατων μοντέλων είναι κατά πολύ ανώτερη από τα περισσότερα προγράμματα κατασκευής εικονικών κόσμων.

Διαπιστώνεται επίσης ότι τα εργαλεία που παρέχονται δεν είναι –με την πρώτη ματιά– ιδιαίτερα εύχρηστα, αλλά δίνουν πρόσβαση σ' όλες τις δυνατότητες της μηχανής και είναι μάλλον θέμα συχνής χρήσης τους ώστε κάποιος να εξοικειωθεί με αυτά. Επίσης, όλα τα εργαλεία συνεργάζονται, ή καλύτερα απαιτούν εξωτερικά προγράμματα. Αυτό γιατί έχουν πολύ περιορισμένες δυνατότητες κατασκευής τρισδιάστατων μοντέλων, με αποτέλεσμα να απαιτείται η κατασκευή τους σ' ένα εξωτερικό πρόγραμμα και στη συνέχεια η εισαγωγή τους στον υπό κατασκευή κόσμο.

Τέλος, ένα βασικό συμπέρασμα είναι ότι οι διαφορές μεταξύ των μηχανών παιχνιδιών είναι ελάχιστες και ότι όλες αποσκοπούν σε δύο πράγματα: ομαλή κίνηση και εντυπωσιακά γραφικά. Έτσι, σε τελική ανάλυση, είναι θέμα προσωπικών προτιμήσεων η επιλογή της μίας ή της άλλης μηχανής παιχνιδιού.

Κεφάλαιο 7^ο

Οι νέες “Νέες Τεχνολογίες”

Για αρκετό καιρό η χρήση των ηλεκτρονικών υπολογιστών και άλλων τεχνολογικών μέσων στην εκπαίδευση περιγράφονταν με τον όρο “Νέες Τεχνολογίες”. Στις μέρες μας η χρήση των υπολογιστών τείνει να γίνει κοινός τόπος. Ο υπερβάλλον ενθουσιασμός για τις δυνατότητές τους, αλλά και οι έντονες επιφυλάξεις για την παιδαγωγική τους αξία –και τα δύο συμπτώματα των αρχικών σταδίων εφαρμογής στην εκπαίδευση– έδωσαν θέση σ’ ένα γόνιμο διάλογο σχετικά με το ρόλο τους. Κατ’ αυτή την έννοια ο όρος “Νέες Τεχνολογίες” είναι ξεπερασμένος.

Ο ρυθμός της τεχνολογικής εξέλιξης από τη μία και οι διαρκώς μεταβαλλόμενες παιδαγωγικές αντιλήψεις μας από την άλλη, μας κάνουν να αναρωτιόμαστε ποια τεχνολογία στο κοντινό μέλλον θα προκαλέσει το ίδιο ενδιαφέρον και συζήτηση όπως η εισαγωγή των πρώτων εφαρμογών της Πληροφορικής πριν από μερικά χρόνια. Με άλλα λόγια, ποιες μπορεί να είναι οι νέες “Νέες Τεχνολογίες”;

Το να προσπαθεί κάποιος να προβλέψει το μέλλον είναι μία πρακτική που συχνά –αν όχι πάντα– οδηγεί σε εσφαλμένα συμπεράσματα. Συνεπώς, οποιαδήποτε ανάλυση μπορεί να θεωρηθεί ισορροπία σε τεντωμένο σχοινί. Αν είναι αρκετά συντηρητική, ίσως αποτύχει να εντοπίσει τις επερχόμενες αλλαγές. Αν είναι αρκετά ρηξικέλευθη, μπορεί να χαρακτηριστεί “επιστημονική φαντασία”. Η ερώτηση όμως παραμένει η ίδια: Ποια τεχνολογία μπορεί να παίξει σημαντικό ρόλο στη διαμόρφωση του σχολείου του μέλλοντος; Ένας τρόπος να απαντηθεί αυτή η ερώτηση είναι να αναλογιστούμε το δυναμικό –ή καλύτερα το μη αξιοποιημένο δυναμικό– των υπαρχουσών τεχνολογιών και εφαρμογών και μετά να αναρωτηθούμε αν αυτό το δυναμικό είναι αρκετό για να ξεκινήσει και να συντηρήσει μία νέα εκπαιδευτική επανάσταση.

Ποιες είναι λοιπόν οι υπάρχουσες τεχνολογίες; Η απάντηση είναι αρκετά εύκολη: το Διαδίκτυο και οι εφαρμογές πολυμέσων. Και οι δύο είναι επικρατούσες μορφές χρήσης υπολογιστών σε κάθε εκπαιδευτική βαθμίδα και είναι τόσο αλληλένδετες, που μπορεί να θεωρηθούν ως μία και μόνη μορφή. Πράγματι, κάθε εφαρμογή πολυμέσων, χωρίς ιδιαίτερες δυσκολίες, μπορεί να επεκτείνει τις δυνατότητές της αξιοποιώντας πηγές από το Διαδίκτυο και κάθε εφαρμογή του Διαδικτύου ενσωματώνει σε ποικίλο βαθμό στοιχεία πολυμέσων.

Ποιο είναι όμως το δυναμικό αυτών των εφαρμογών που μέχρι τώρα δεν έχει αξιοποιηθεί απόλυτα; Πρώτα απ’ όλα, η εξάπλωσή τους. Ένας τρόπος για να επιτευχθεί αυτό είναι να μειωθεί το κόστος το οποίο αφορά είτε στην απόκτηση ενός υπολογιστή είτε μίας εφαρμογής είτε της σύνδεσης στο Διαδίκτυο. Δεύτερον, υπάρχουν ήδη διαθέσιμες (με λογικό κόστος) τεχνολογίες που βελτιώνουν κατά πολύ την ταχύτητα σύνδεσης του μέσου χρήστη στο Διαδίκτυο. Τόσο η μείωση του κόστους όσο και η αύξηση της ταχύ-

τητας αναμένεται να έχουν τεράστια επίπτωση στον όγκο των δεδομένων που μπορεί να διακινήθει σε σύντομο χρόνο, σ' ένα μεγάλο αριθμό χρηστών. Με άλλα λόγια, το δημοκρατικό ιδεώδες της "πρόσβασης στη γνώση απ' όλους" μπορεί να γίνει πραγματικότητα, χάρη στις ήδη υπάρχουσες τεχνολογίες.

Ας έρθουμε τώρα στο θέμα της ποιότητας στην εκπαίδευση και στο θέμα της εφαρμογής των νέων παιδαγωγικών θεωριών. Η μελέτη των τρόπων με τους οποίους οι μαθητές αλληλεπιδρούν με το κάθε γνωστικό αντικείμενο επέτρεψε τη διαμόρφωση της άποψης ότι η γνώση δομείται και δε μεταδίδεται απλώς. Ο δομητισμός, αν και δεν είναι μία ενιαία θεωρία, υποστηρίζει αυτή την άποψη, παραδέχεται ότι το κάθε άτομο κατασκευάζει τις δικές του αναπαραστάσεις της πραγματικότητας, αναγνωρίζει ότι μαθαίνουμε μέσω της ενεργού συμμετοχής και δράσης και τέλος επισημαίνει ότι η μάθηση είναι μία διαλεκτική και αλληλεπιδραστική δραστηριότητα με τον κοινωνικό περίγυρο. Ο δομητισμός παρέχει αρκετές ιδέες για το πώς μπορούμε να επιτύχουμε τα παρὰπάνω με τη χρήση υπολογιστών, κυρίως κατασκευάζοντας εφαρμογές που δε θέτουν όρια στο είδος και τη μορφή των αλληλεπιδράσεων που μπορεί να έχει ο χρήστης με αυτές.

Πολλοί υποστηρίζουν ότι όλα αυτά ήδη υλοποιούνται με τις υπάρχουσες διαδικτυακές και πολυμεσικές εφαρμογές. Πράγματι, οι όροι "αλληλεπίδραση", "ελεύθερη και μη γραμμική πλοήγηση", "συνεργασία" και "έλεγχος του μαθητή πάνω στο γνωστικό αντικείμενο" είναι συχνοί σε επιστημονικά συγγράμματα που μελετούν και υποστηρίζουν τη χρήση διαδικτυακών και πολυμεσικών εφαρμογών στην εκπαίδευση.

Ας αναρωτηθούμε όμως αν οι εφαρμογές πολυμέσων και το Διαδίκτυο έφεραν πράγματι επανάσταση στον τρόπο που διδάσκουμε τους μαθητές ή αν είναι απλώς ένα ακόμα εργαλείο που υποβοηθά τη διδασκαλία. Ας είμαστε σκληροί στην κριτική μας. Η επιχειρηματολογία για την έννοια της συνεργασίας υπάρχει από πολύ καιρό πριν την εμφάνιση των υπολογιστών. Το ίδιο ισχύει και για τις έννοιες της διερευνητικής μάθησης, της αλληλεπίδρασης, κτλ. Στην πραγματικότητα λοιπόν αυτό που επιτύχαμε με τους υπολογιστές είναι να έχουμε έναν πιο εύκολο τρόπο για την υλοποίηση των παρὰπάνω. Δεν υπάρχει λοιπόν πραγματική επανάσταση σ' αυτό το θέμα.

Μήπως οι εφαρμογές πολυμέσων και το Διαδίκτυο διαφέρουν πάρα πολύ από τα μέσα που είχε στη διάθεσή του ένας δάσκαλος τριάντα, σαράντα, πενήντα ή ακόμα εκατό χρόνια πριν; Όχι στην πραγματικότητα. Ο μαυροπίνακας, τα βιβλία, η τηλεόραση και το βίντεο βρίσκονται συμπιεσμένα και όμορφα πακεταρισμένα σε μία συσκευή που λέγεται υπολογιστής. Έτσι, αντί να παρέχουμε στους δασκάλους και τους μαθητές βιβλία, μπορούμε να τους εφοδιάσουμε με φορητούς υπολογιστές που να περιέχουν ό,τι

είναι απαραίτητο για τη σχολική χρονιά. Για να είμαστε ειλικρινείς, μία τέτοια εξέλιξη θα ήταν ιδιαίτερα σημαντική. Αλλά και πάλι, δεν υπάρχει τίποτα ριζοσπαστικό. Αντί οι μαθητές να συνεργάζονται χρησιμοποιώντας τη σχολική βιβλιοθήκη, θα μπορούν να συνεργάζονται χρησιμοποιώντας φορητούς υπολογιστές που τους δίνουν πρόσβαση σε ηλεκτρονικές βιβλιοθήκες.

Δεν είναι όμως η “εξ αποστάσεως εκπαίδευση” μία σημαντική καινοτομία που έφεραν οι υπολογιστές και τα δίκτυα; Μάλλον όχι. Και πάλι αυτό που συμβαίνει είναι ότι οι υπολογιστές παρέχουν με διαφορετικό τρόπο λύσεις που υπήρχαν καιρό πριν. Οι μαθητές στη Φιλανδία, την Αυστραλία, τον Καναδά και σε άλλες περιοχές του πλανήτη χρησιμοποιούσαν την από απόσταση εκπαίδευση πολύ πριν από την έλευση των υπολογιστών.

Όλα τα επιχειρήματα που παρουσιάστηκαν μας οδηγούν στο συμπέρασμα ότι η τεχνολογία παρέχει μία πιο ευέλικτη, πιθανώς πιο οικονομική και συζητήσιμη καλύτερη μορφή “παραδοσιακής” εκπαίδευσης, ότι η πολυδιαφημισμένη εκπαιδευτική επανάσταση χάρη στην τεχνολογία δεν είναι πραγματική επανάσταση. Για να βρούμε ποια θα είναι η πραγματική επανάσταση, πρέπει να απαντήσουμε σε πιο θεμελιώδεις ερωτήσεις. Πρέπει να αναρωτηθούμε τι πρόβλημα υπάρχει και με τα βιβλία και με τους υπολογιστές, πρέπει να αναρωτηθούμε τι πρόβλημα υπάρχει με τους τεχνολογικά εμπλουτισμένους και τους παραδοσιακούς τρόπους διδασκαλίας.

Κατά τη γνώμη μας, η απάντηση βρίσκεται στο γεγονός ότι κάποιος (ο δάσκαλος) ή κάτι (ο μαυροπίνακας, το βιβλίο, ο υπολογιστής) παρεμβάλλονται μεταξύ της ανεπεξέργαστης πληροφορίας και γνώσης και του ατόμου που επιθυμεί ή πρέπει να γνωρίσει αυτή τη γνώση. Αυτός που μαθαίνει δεν έχει άμεση πρόσβαση στη γνώση, αλλά έμμεση. Τα δεδομένα έχουν ήδη δεχτεί επεξεργασία από το δάσκαλο, το συγγραφέα του βιβλίου, τον προγραμματιστή που κατασκεύασε την εφαρμογή ή τις ιστοσελίδες. Έτσι όμως οι εκπαιδευτές αποκόπτουν τους μαθητές ίσως από τον πιο σημαντικό τρόπο που εμείς οι άνθρωποι μαθαίνουμε, δηλαδή από τις εμπειρίες πρώτου προσώπου.

Όπως αναλύθηκε, οι εμπειρίες πρώτου προσώπου είναι άμεσες, δεν απαιτούν τη χρήση συμβόλων (γράμματα, αριθμούς, γλώσσα) και είναι καθαρά υποκειμενικές. Από την άλλη, οι εμπειρίες τρίτου προσώπου είναι έμμεσες, είναι απαραίτητη η χρήση συμβόλων και προβάλλουν την άποψη και τα πιστεύω αυτού που επικοινωνεί τη γνώση. Όπως ακριβώς στο απλουστευμένο παράδειγμα που χρησιμοποιήσαμε σε προηγούμενο κεφάλαιο, άλλο πράγμα είναι να βλέπεις μία κινηματογραφική ταινία και άλλο να σου την αφηγούνται, έστω και πολύ παραστατικά.

Αυτό το πρόβλημα έγινε αντιληπτό εδώ και αρκετό καιρό. Πολλοί επιστήμονες που ασχολούνται με τις θεωρίες μάθησης πρότειναν πρακτικές σύμφωνα με τις οποίες οι μαθητές μπορούν να έχουν εμπράγματα εμπειρίες (δηλαδή εμπειρίες πρώτου προσώπου) σε όσο το δυνατόν περισσότερα γνωστικά αντικείμενα. Φυσικά όμως υπάρχουν ορισμένοι περιορισμοί. Δεν μπορούμε να εκθέσουμε τους μαθητές σε κίνδυνο μόνο και μόνο για να αποκτήσουν εμπειρίες πρώτου προσώπου για τις εκρήξεις ηφαιστειών. Δεν μπορούμε να ταξιδέψουμε πίσω στο χρόνο για να γνωρίσουν οι μαθητές τη ζωή των δεινοσαύρων. Ούτε μπορούν να χρησιμοποιήσουν το τηλεσκόπιο Hubble για να δουν τα άστρα, ούτε ηλεκτρονικά μικροσκόπια για να δουν τα μικρόβια. Όχι μόνο οι μαθητές, αλλά όλοι μας είμαστε “παγιδευμένοι” στις εμπειρίες τρίτου προσώπου που παρέχει η βασισμένη στην τεχνολογία και η παραδοσιακή εκπαίδευση.

Πιστεύουμε ότι η Ε.Π. προσφέρει μία διέξοδο απ' αυτή την κατάσταση, γιατί μπορεί να άρει τους περισσότερους από τους περιορισμούς που αναφέρθηκαν. Σύμφωνα με όσα παρουσιάσαμε, η Ε.Π. τοποθετεί το χρήστη σ' ένα συνθετικό περιβάλλον ίδιο ή απόλυτα διαφορετικό από το πραγματικό. Στο περιβάλλον αυτό ο χρήστης μπορεί να επιλέξει τη διαδρομή που επιθυμεί, να εξερευνήσει τα πάντα, όποτε και με όποιον τρόπο εκείνος θέλει, και τέλος μπορεί να αλληλεπιδράσει με τα αντικείμενα που υπάρχουν στον κόσμο αυτό.

Το αποτέλεσμα είναι ότι ο χρήστης αποκτά εμπειρίες πρώτου προσώπου σχετικά με το γνωστικό αντικείμενο της εφαρμογής και δημιουργεί καθαρά προσωπικές αναπαραστάσεις του συνθετικού κόσμου που βρίσκεται. Η μάθηση σ' ένα τέτοιο περιβάλλον είναι σίγουρα μία δυναμική διαδικασία, πλήρως καθορισμένη από το χρήστη, που μόνος του θέτει τους στόχους και τους αλλάζει κατά βούληση. Τα αποτελέσματα δε είναι διαφορετικά από άτομο σε άτομο. Επιπρόσθετα, η Ε.Π. μπορεί να παρέχει και εμπειρίες τρίτου προσώπου εάν, για παράδειγμα, ο προγραμματιστής θέσει περιορισμούς στο τι μπορεί να κάνει ο χρήστης.

Ένα από τα σημαντικά χαρακτηριστικά της Ε.Π. είναι η δυνατότητα να συνυπάρχουν στην εφαρμογή περισσότεροι του ενός χρήστες. Έτσι, αυτοί δεν αλληλεπιδρούν μόνο με τα αντικείμενα του συνθετικού περιβάλλοντος αλλά και μεταξύ τους, έχοντας τη δυνατότητα να συνομιλήσουν, να ανταλλάξουν απόψεις και να καθοδηγήσουν ο ένας τον άλλο. Ο βαθμός αλληλεπίδρασης είναι συνεπώς υψηλός. Αυτό το χαρακτηριστικό μαζί με την εμπύθιση του χρήστη στο εικονικό περιβάλλον και τις εμπειρίες πρώτου προσώπου συντελούν στη μεγάλη συμβατότητα της Ε.Π. με τις αρχές του δομητισμού.

Έχοντας στη διάθεσή μας ένα τόσο ισχυρό εκπαιδευτικό εργαλείο, κάποιος μπορεί να πιστέψει ότι ήδη υπάρχουν αναρίθμητες εκπαιδευτικές εφαρμογές Ε.Π. και ότι η

πραγματική εκπαιδευτική επανάσταση έχει ήδη αρχίσει. Δυστυχώς όμως κάτι τέτοιο δεν ισχύει. Τα αίτια δε βρίσκονται στην αποτυχία των εφαρμογών να έχουν θετικά αποτελέσματα, μάλιστα συμβαίνει το ακριβώς αντίθετο. Το ουσιαστικό πρόβλημα προς το παρόν είναι το κόστος των εφαρμογών και του εξοπλισμού. Ο κανόνας ότι κάθε επόμενη γενιά υπολογιστών είναι φτηνότερη αλλά πιο ισχυρή από την προηγούμενη δε φαίνεται να έχει ισχύ στην Ε.Π.

Η Ε.Π. στηρίζεται αποκλειστικά στα τρισδιάστατα γραφικά και απαιτεί τη χρήση ισχυρών και καλά εξοπλισμένων υπολογιστών. Ένα τυπικό σύστημα Ε.Π. (υλικό και λογισμικό) κοστίζει αρκετές χιλιάδες ευρώ και η απόκτησή του είναι πέρα από τις δυνατότητες του μέσου χρήστη. Αυτό περιορίσε σημαντικά το αρχικά υψηλό ενδιαφέρον για την Ε.Π., ενώ το ερευνητικό ενδιαφέρον μετατοπίστηκε σε ακόμα πιο εξειδικευμένες εφαρμογές Ε.Π. (για παράδειγμα στην πλήρως εμβυθισμένη Ε.Π.) με ακόμα μεγαλύτερο κόστος.

Όμως τα τελευταία χρόνια η κατάσταση φαίνεται να αλλάζει. Η κινητήρια δύναμη πίσω απ' αυτές τις εξελίξεις είναι η παντοδύναμη βιομηχανία της ψυχαγωγίας, που αναζητά διαρκώς νέους τρόπους για να προσελκύσει το ενδιαφέρον των καταναλωτών. Ισχυρές κάρτες γραφικών με λογικό κόστος εξοπλίζουν υπολογιστές που απευθύνονται στο μέσο χρήστη και τέτοιοι υπολογιστές είναι σε θέση να εκτελέσουν εφαρμογές Ε.Π. Οι προδιαγραφές μάλιστα δε διαφέρουν σημαντικά από τις προδιαγραφές των υπολογιστών που εξοπλίζουν τα ήδη υπάρχοντα σχολικά εργαστήρια.

Αν κάτι είναι απαραίτητο να αναβαθμιστεί, είναι οι κάρτες γραφικών, που το κόστος τους είναι κλάσμα της απόκτησης ενός πλήρους συστήματος. Στην ανάλυσή μας μάλιστα θεωρήσαμε ότι η επιτραπέζια Ε.Π., που απαιτεί απλούς υπολογιστές, είναι όχι μόνο πιο προσιτή οικονομικά, αλλά ταυτόχρονα δεν εμφανίζει σύνθετα τεχνικά προβλήματα και δε δημιουργεί προβλήματα υγείας (συνεπώς είναι πιο ώριμη η τεχνολογία που χρησιμοποιεί). Παράλληλα, δε γίνονται σημαντικοί συμβιβασμοί στο θέμα της εμβύθισης, που είναι κρίσιμη παράμετρος στις εφαρμογές Ε.Π.

Αλλαγές όμως έχουν γίνει και στο λογισμικό που απαιτείται για την κατασκευή εικονικών κόσμων, με αποτέλεσμα να χρησιμοποιούνται πιο δημοφιλή εργαλεία, αρκεί να σημειώσουμε ότι η C++ είναι η πλέον διαδεδομένη προγραμματιστική γλώσσα. Όλα αυτά συντελούν στην αναγέννηση του ενδιαφέροντος για την Ε.Π.

Μάλιστα επιχειρήσαμε να κάνουμε μία πολύ πιο τολμηρή προσέγγιση στο πρόβλημα κατασκευής εικονικών κόσμων, ορμώνενοι από τη διαπίστωση ότι οι εφαρμογές Ε.Π. σχεδόν ταυτίζονται σε δυνατότητες με τα τρισδιάστατα παιχνίδια βολών πρώτου προ-

σώπου. Μία προσεκτική επιλογή μηχανής παιχνιδιών, είτε απ' αυτές που δίνονται μαζί με ορισμένα παιχνίδια είτε απ' αυτές που διατίθενται χωρίς χρέωση, μειώνει δραστικά το κόστος και είναι πολύ πιο εύκολο για τον καθένα να αναπτύξει εφαρμογές τρισδιάστατων γραφικών και Ε.Π.

Έχοντας ικανοποιήσει τις τρεις προϋποθέσεις που αναφέρθηκαν στην εισαγωγή (άμεση και μαζική εφαρμογή, οικονομικά συμφέρουσα λύση) και με αποδεδειγμένη την παιδαγωγική αξία της Ε.Π., το τελικό συμπέρασμά μας είναι ότι η χρήση εφαρμογών πολυμέσων και Διαδικτύου δεν είναι τίποτα περισσότερο παρά το αναγκαίο μεταβατικό στάδιο από τη συμβατική εκπαίδευση, στην εκπαίδευση που στηρίζεται σε τεχνολογία με δυνατότητες που μόνο η Ε.Π. μπορεί να παρέχει.

Κεφάλαιο 8^ο

Τα πρώτα βήματα στο Reality Factory

Η Πληροφορική στην εκπαίδευση αποτελεί ένα επιστημονικό πεδίο που κατεξοχήν εισάγει την καινοτομία στην εκπαιδευτική πράξη. Η καινοτομία προκύπτει συνήθως ως αποτέλεσμα γνώσεων και ιδεών, αλλά “καταξιώνεται” και γίνεται κοινά αποδεκτή όταν μπορεί να μετατραπεί σε εκπαιδευτική εφαρμογή με συγκεκριμένα χαρακτηριστικά, ορατά αποτελέσματα και αποδοτικότητα.

Οι καινοτόμες εφαρμογές Πληροφορικής στην εκπαίδευση στηρίζονται συνήθως στην ανάπτυξη συστημάτων και εργαλείων που πρέπει στην πράξη να ανταποκρίνονται σε συγκεκριμένες απαιτήσεις. Τα εργαλεία πρέπει να είναι άμεσα προσπελάσιμα, φιλικά στο χρήστη, ελκυστικά, σχετικά χαμηλού κόστους και θετικού οφέλους. Αν δε συγκεντρώνουν τα παραπάνω χαρακτηριστικά, που τα καθιστούν “κανονικά” αγαθά καθημερινής χρήσης και αντίθετα αποτελούν αγαθά “πολυτελείας”, τότε η εκπαίδευση, που είναι κατά κανόνα μαζική και όχι ελιτίστικη, ελάχιστα μπορεί να ωφεληθεί από την Πληροφορική.

Στα παραπάνω πλαίσια, μία από τις μεγάλες προκλήσεις στο πεδίο της Πληροφορικής στην εκπαίδευση είναι η αναζήτηση και η ενσωμάτωση στην εκπαιδευτική διαδικασία ιδεών που μπορεί να ενέχουν το στοιχείο του ρηξικέλευθου και της εκζήτησης, μετατρέπονται όμως σε εφαρμογές “καθημερινής και ευρείας χρήσης”, εύκολες για τον εκπαιδευτικό που κατά κανόνα δεν είναι ειδικός στην Πληροφορική, ελκυστικές για το μαθητή που καλείται να τις εφαρμόσει, προσιτές από άποψη κόστους και απαιτούμενης υποδομής για τους φορείς του εκπαιδευτικού σχεδιασμού και πειστικές για το ευρύτερο εκπαιδευτικό περιβάλλον.

Έτσι λοιπόν και το εύρος εφαρμογής της Ε.Π. στην εκπαίδευση καθορίζεται από το κατά πόσο αυτή πληροί τα χαρακτηριστικά των εφαρμογών καθημερινής χρήσης. Είναι αναγκαίο, για παράδειγμα, να μη χρειάζεται εξεζητημένο εξοπλισμό εκτός από τα συνήθη μηχανήματα που η σημερινή τάξη διαθέτει, όπως είναι απαραίτητο να στηρίζεται σε λογισμικό ελκυστικό, φτηνό και απλό στη χρήση. Αν η Ε.Π. διαθέτει τέτοια χαρακτηριστικά, τότε οι προοπτικές της στην εκπαίδευση διαγράφονται ευοίωνες. Μάλιστα, όσο η τεχνολογία προχωρά και προσφέρει υψηλότερες ταχύτητες επεξεργασίας και μεγαλύτερες χωρητικότητες, τόσο αυτό θα έχει ως αποτέλεσμα οι εφαρμογές της Ε.Π. να κερδίζουν έδαφος.

Στα προηγούμενα κεφάλαια επισημάναμε την αναγκαιότητα σημαντικής μείωσης του κόστους του υλικού και του λογισμικού που απαιτείται για την κατασκευή και χρήση εφαρμογών Ε.Π. Για το μεν υλικό καταλήξαμε στο συμπέρασμα ότι, αφού η επιτραπέζια Ε.Π. αποτελεί την ενδεδειγμένη λύση τόσο από οικονομικής όσο και από παιδαγωγικής άποψης, το κόστος το οποίο πρέπει να αντιμετωπιστεί είναι η πιθανή αναβάθμι-

ση της κάρτας γραφικών του υπολογιστή, μιας και αυτή είναι που πρέπει να αντεπεξέλθει στο σημαντικό υπολογιστικό φορτίο.

Για δε το λογισμικό θεωρήσαμε ότι η χρήση μίας προσεκτικά επιλεγμένης μηχανής τρισδιάστατων παιχνιδιών υπερκαλύπτει τις προδιαγραφές που πρέπει να ικανοποιεί ένα πρόγραμμα ανάπτυξης εικονικών κόσμων. Το ενδιαφέρον στοιχείο είναι ότι αρκετές μηχανές παιχνιδιών και τα προγράμματα με τα οποία κατασκευάστηκαν τα επίπεδα του παιχνιδιού δίνονται μαζί με την αγορά του παιχνιδιού και παράλληλα η κατασκευάστρια εταιρεία επιτρέπει –και πολλές φορές ενθαρρύνει– τους αγοραστές να κατασκευάσουν και να μοιραστούν με άλλους χρήστες νέα επίπεδα. Οι όροι που τίθενται είναι να έχει αγοραστεί το παιχνίδι, να μην αλλοιωθεί ο κώδικας της μηχανής και το ίδιο το παιχνίδι και να μη χρησιμοποιηθούν τα νέα επίπεδα για εμπορική χρήση. Λαμβάνοντας υπόψη ότι η δαπάνη για την αγορά αυτού του είδους των παιχνιδιών δεν υπερβαίνει τα 50 ευρώ, εύκολα κάποιος καταλαβαίνει ότι το κόστος είναι πολύ μικρό.

Όμως δεν είναι η μόνη λύση που έχει στη διάθεσή του κάποιος που ενδιαφέρεται να αναπτύξει μία εφαρμογή τρισδιάστατων γραφικών. Μέχρι τη στιγμή που γραφόταν αυτό το βιβλίο υπήρχαν καταχωρισμένες 246 μηχανές παιχνιδιών σε διάφορες κατηγορίες [138]. Το ενδιαφέρον μας εστιάστηκε σε εκείνες που χαρακτηρίζονται *ανοιχτού κώδικα* (*Open Source*), γιατί έχουν κατ' αρχάς μηδενικό κόστος (δίνονται δωρεάν) και κατά δεύτερον υπάρχουν ελάχιστοι και μερικές φορές κανένας όρος στις επεμβάσεις που μπορούν να γίνουν και στη διανομή του παραγόμενου προϊόντος.

Αναζητήσαμε μία μηχανή παιχνιδιών ανοιχτού κώδικα που να παρέχει εργαλεία για την εύκολη και γρήγορη ανάπτυξη εφαρμογών, γιατί οι περισσότερες απαιτούν πολύ καλές γνώσεις προγραμματισμού που είναι έξω από τις δυνατότητες του μέσου χρήστη. Ένα βασικό σημείο που προσέξαμε ήταν η μηχανή να μην είναι απλώς μία πειραματική προσπάθεια, αλλά να είναι ένα ολοκληρωμένο προϊόν, να παρέχει αρκετές δυνατότητες και να συνεχίζει να εξελίσσεται. Θέλαμε τέλος να έχει καλή τεκμηρίωση και υποστήριξη, ώστε ο χρήστης να έχει τη δυνατότητα να αναζητήσει βοήθεια σε περίπτωση που αντιμετωπίσει κάποιο πρόβλημα. Από τις μηχανές και τα προγράμματα που πληρούσαν τις προδιαγραφές μας ξεχώρισε το Reality Factory (RF) [139].

Το RF αναπτύχθηκε αρχικά από τους Ed Averil και Terry Morgan το 2000. Συνεχίζει να αναπτύσσεται από τους Terry Morgan, Ralph Deane και Mike Wuetherick (η τελευταία έκδοση του προγράμματος ήταν το Φεβρουάριο του 2007). Στηρίζεται στην *Genesis 3D* [140], μηχανή παιχνιδιών ανοιχτού κώδικα, την οποία εξελίσσει περαιτέρω προσθέτοντας νέες δυνατότητες. Πιστεύουμε ότι η χρήση του RF είναι αρκετά εύκολη,

ώστε ακόμα και ένας αρχάριος στον τομέα αυτό να μπορεί σε σύντομο χρονικό διάστημα να κατασκευάσει την πρώτη του εφαρμογή.

Στα κεφάλαια που ακολουθούν, ο αναγνώστης θα έχει την ευκαιρία να γνωρίσει τα σημαντικότερα σημεία του RF, από τα πρώτα βήματα εγκατάστασής του μέχρι τον τρόπο διανομής της εφαρμογής που έχει κατασκευάσει. Πρέπει να σημειώσουμε ότι δεν περιλάβαμε εκτενή στοιχεία από τη γλώσσα προγραμματισμού που περιέχει –τη SimKin– αφενός γιατί προϋποθέτει καλές γνώσεις προγραμματισμού στη γλώσσα C και αφετέρου γιατί χρησιμοποιείται σε εξειδικευμένα θέματα που σπάνια ο μέσος χρήστης θα συναντήσει.

8.1 Βοηθητικά προγράμματα και προδιαγραφές υλικού

Παρόλο που το RF περιλαμβάνει αρκετά εργαλεία που θα χρειαστούμε στην κατασκευή εικονικών κόσμων, εντούτοις είναι απαραίτητη η χρήση ορισμένων εξωτερικών βοηθητικών προγραμμάτων για την επεξεργασία εικόνων, ήχων και τρισδιάστατων μοντέλων. Στη συνέχεια προτείνουμε κάποια προγράμματα σε καθεμιά από τις παραπάνω κατηγορίες, με την επισημάνση ότι ο αναγνώστης μπορεί να αναζητήσει και να χρησιμοποιήσει άλλα της δικής του επιλογής.

Για την επεξεργασία και την κατασκευή εικόνων μπορεί να χρησιμοποιηθεί το πρόγραμμα “Ζωγραφική” που περιλαμβάνεται στα Windows. Επειδή βέβαια το πρόγραμμα αυτό έχει ελάχιστες δυνατότητες, μπορούν να χρησιμοποιηθούν:

- Το Deep Paint (<http://www.download.com/Deep-Paint/3000-2191-10398243.html>) ή
- Το Gimp (<http://www.gimp.org/>)

που είναι freeware (δηλαδή δίνονται δωρεάν).

Βεβαίως υπάρχουν κορυφαία προγράμματα στην κατηγορία της επεξεργασίας εικόνων, που όμως κοστίζουν αρκετά όπως:

- Το Adobe Photoshop CS3 (<http://www.adobe.com/products/photoshop/main.html>) και
- Το Corel Draw (<http://www.corel.com/>)

Για την ηχογράφηση και την επεξεργασία ήχων μπορεί να χρησιμοποιηθεί η “Ηχογράφηση” που περιλαμβάνεται στα Windows. Όπως και η “Ζωγραφική”, το πρόγραμμα είναι ιδιαίτερα απλοϊκό και ως εκ τούτου προτείνουμε:

- Το Audacity (<http://audacity.sourceforge.net/download/>) ή
- Το WavePad (<http://www.nch.com.au/wavepad/>)

που και αυτά είναι freeware.

Μπορούν βέβαια να χρησιμοποιηθούν επαγγελματικά προγράμματα επεξεργασίας ήχων όπως το Adobe Audition (<http://www.adobe.com/products/audition/main.html>), με σημαντικό βέβαια κόστος.

Η κατασκευή και επεξεργασία μοντέλων είναι η μόνη κατηγορία βοηθητικών προγραμμάτων για την οποία ο χρήστης θα έχει κάποιο κόστος. Λόγω ορισμένων ιδιοτήτων του RF που θα αναλυθούν σε επόμενο κεφάλαιο, συνιστούμε το Milkshape3D (<http://www.milkshape3d.com/>). Το Milkshape3D ανήκει στην κατηγορία των shareware προγραμμάτων (επιτρέπεται η δοκιμαστική χρήση για ένα μικρό χρονικό διάστημα και στη συνέχεια απαιτείται η αγορά του μ' ένα σχετικά μικρό αντίτιμο) και κοστίζει περίπου 25 ευρώ.

Ένα από τα σημαντικότερα προβλήματα που θα αντιμετωπίσει ο χρήστης του RF –και κάθε άλλου προγράμματος κατασκευής εικονικών κόσμων– είναι τα τρισδιάστατα μοντέλα. Θα χρειαστεί ή να τα κατασκευάσει μόνος του (για παράδειγμα με το Milkshape3D) ή να βρει έτοιμα. Για τη δεύτερη περίπτωση, υπάρχουν αναρίθμητοι δικτυακοί τόποι που πωλούν ή δίνουν δωρεάν τρισδιάστατα μοντέλα. Ενδεικτικά, δωρεάν μοντέλα σε διάφορες κατηγορίες μπορούν να βρεθούν:

- Στο 3D Café και στην επιλογή Free Stuff (<http://www.3dcafe.com/>)
- Σύνδεσμοι για δικτυακούς τόπους που δίνουν δωρεάν μοντέλα βρίσκονται στο 3D Valley Models στην επιλογή Links (<http://www.3dvalley.com/>). Στον ίδιο δικτυακό τόπο υπάρχουν και μερικά δωρεάν μοντέλα (<http://www.3dvalley.com/models.shtml>).
- Στο Turbo Squid (<http://www.turbosquid.com/3d>), όπου χρειάζεται δωρεάν εγγραφή.

Τέλος, πρέπει να σημειώσουμε ότι το RF δεν έχει ιδιαίτερες απαιτήσεις από το υλικό του χρήστη και μπορεί να “τρέξει” ακόμα και σε παλαιότερους υπολογιστές. Οι απαιτούμενες προδιαγραφές είναι:

Λειτουργικά συστήματα που υποστηρίζονται

Windows 98 και Windows 98 Second Edition, Windows Millennium

Windows 2000 Professional (Workstation)

Windows XP (Home και Professional)

Windows Vista (όλες οι εκδόσεις, παρουσιάζονται όμως προβλήματα στην ταχύτητα εκτέλεσης της εφαρμογής λόγω του διαφορετικού χειρισμού του υλικού από τα Vista)

Λειτουργικά συστήματα που δεν υποστηρίζονται

Windows 95 και Windows NT

Mac OS

Linux

BEOS

Ελάχιστες προδιαγραφές υλικού

Επεξεργαστής 500 MHz Intel Pentium III ή AMD Athlon

128 Mb μνήμης RAM

250 Mb ελεύθερου χώρου στο σκληρό δίσκο

Κάρτα γραφικών με 32 Mb μνήμη RAM, με οδηγούς συμβατούς με το DirectX 8.1

Κάρτα ήχου με οδηγούς συμβατούς με το DirectX 8.1

Συνιστώμενες προδιαγραφές υλικού

Επεξεργαστής 1(+) GHz Intel Pentium IV ή AMD Athlon

256(+) Mb μνήμης RAM

500(+) Mb ελεύθερου χώρου στο σκληρό δίσκο

Κάρτα γραφικών με 64 Mb μνήμη RAM, με οδηγούς συμβατούς με το DirectX 8.1

Κάρτα ήχου με οδηγούς συμβατούς με το DirectX 8.1

8.2 Εγκατάσταση του Reality Factory

Το RF είναι διαθέσιμο στη διεύθυνση <http://realityfactory.info.ms/>. Τη στιγμή που γραφόταν το βιβλίο η τελευταία έκδοση ήταν η 0.75C, η οποία όμως απαιτεί να έχουμε εγκατεστημένη την έκδοση 0.75A. Χρειάζεται λοιπόν να “κατεβάσουμε” και να αποθηκεύσουμε στο σκληρό μας δίσκο τα αρχεία RealityFactory075A.exe και RealityFactory075C.zip. Εκτελούμε το αρχείο RealityFactory075A.exe (κάνοντάς το διπλό κλικ) και εκκινεί ένα τυπικό πρόγραμμα εγκατάστασης. Η διαδικασία είναι απλή και δε χρειάζεται να γίνουν ρυθμίσεις, εκτός ίσως αν επιθυμούμε να αλλάξουμε το φάκελο που θα εγκατασταθεί το πρόγραμμα. Πατώντας διαδοχικά “Next” ολοκληρώνεται η εγκατάσταση.

Για να εγκαταστήσουμε την έκδοση 0.75C χρειάζεται να έχουμε κάποιο πρόγραμμα που να μπορεί να “ανοίγει” συμπιεσμένα αρχεία με κατάληξη .zip, όπως για παράδειγμα το Winzip ή το Winrar. Στην περίπτωση που το λειτουργικό μας σύστημα είναι τα Windows XP, δε χρειαζόμαστε κάποιο πρόγραμμα, γιατί η ανάγνωση αυτών των αρχείων υποστηρίζεται από το ίδιο το λειτουργικό. Κάνουμε διπλό κλικ στο αρχείο RealityFactory075C.zip και από τα αρχεία που περιέχονται αποσυμπιέζουμε το αρχείο RealityFactory.exe στο φάκελο C:\RealityFactory, αντικαθιστώντας το ήδη υπάρχον. Αποσυμπιέζουμε επίσης το αρχείο msvcrt.dll (βρίσκεται στο φάκελο Tools) στο φάκελο C:\RealityFactory\Tools, αντικαθιστώντας και πάλι το ήδη υπάρχον.

8.3 Η δομή των αρχείων του Reality Factory

Κατά την εγκατάστασή του το RF δημιουργεί συντομεύσεις στην “Έναρξη”, “Όλα τα προγράμματα”, από τις οποίες μπορούμε να εκκινήσουμε τα περισσότερα προγράμματα και εργαλεία που περιέχει. Όμως καλό είναι να γνωρίζουμε τι περιέχεται στο φάκελο και τους υπο-φακέλους του προγράμματος.

Πρέπει επίσης να σημειώσουμε ότι το RF, όπως και τα περισσότερα προγράμματα ανοιχτού κώδικα, επιτρέπουν μεγάλο αριθμό επεμβάσεων στα αρχεία που συγκροτούν το πρόγραμμα. Όμως αρκετές πρέπει να γίνουν “χειροκίνητα”, δεν παρέχονται δηλαδή ειδικά εργαλεία για το σκοπό αυτό. Συνεπώς, ο χρήστης πρέπει να γνωρίζει τη λειτουργία των αρχείων του προγράμματος καθώς επίσης τις αλλαγές που μπορεί να κάνει σε αυτά.

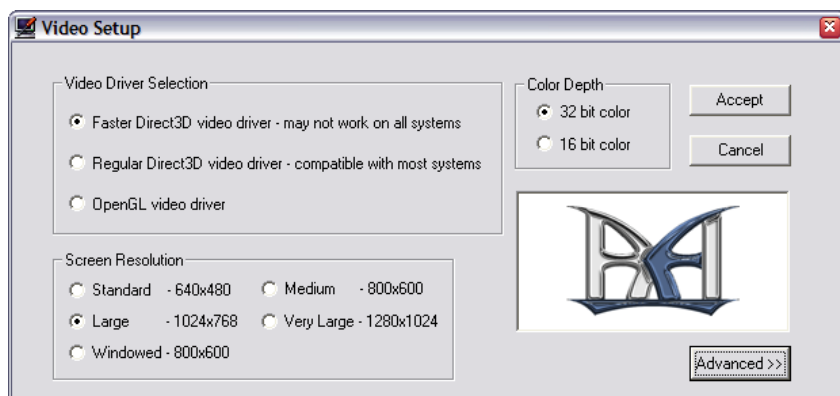
8.3.1 Αρχεία στον κυρίως φάκελο

Κατ’ αρχάς, στο φάκελο C:\RealityFactory, υπάρχει το εκτελέσιμο πρόγραμμα RealityFactory.exe που είναι στην ουσία η μηχανή παιχνιδιού. Όταν κατασκευάσουμε μία εφαρμογή, θα κληθεί μέσω αυτού του αρχείου/προγράμματος και θα εκτελεστεί. Αν αμέσως μετά την εγκατάσταση ή χωρίς να έχουμε ορίσει κάποιες παραμέτρους που αναλύονται σε επόμενο κεφάλαιο, κάνουμε διπλό κλικ στο RealityFactory.exe, θα εκτελεστεί ένα *επίπεδο επίδειξης (Demo Level)*.

Στον ίδιο φάκελο περιέχονται τρία αρχεία με την κατάληξη .log. Πρόκειται για αρχεία καταγραφής και είναι χρήσιμα σε περίπτωση προβλημάτων, ειδικά το αρχείο RealityFactory.log. Κάθε φορά που εκτελείται το πρόγραμμα RealityFactory.exe, προστίθενται νέες πληροφορίες σε αυτά τα αρχεία που αφορούν κάθε βήμα εκτέλεσης της εφαρμογής, όπως για παράδειγμα το φόρτωμα και η αρχικοποίηση των υποσυστημάτων ήχου, εικόνας και δικτύου ή αντίστοιχα ο τερματισμός χρήσης τους. Αναλύοντας τις πληροφορίες αυτές (μπορούμε να τις δούμε κάνοντας διπλό κλικ σ’ ένα από τα αρχεία καταγραφής τα οποία και ανοίγουν με το “Σημειωματάριο”) μπορούμε να διαπιστώσουμε εάν και πού το πρόγραμμα συνάντησε προβλήματα.

Επειδή το υποσύστημα στο οποίο ο χρήστης του RF μπορεί να αντιμετωπίσει τα περισσότερα προβλήματα είναι αυτό της διαχείρισης της εικόνας, στο φάκελο C:\RealityFactory περιλαμβάνεται και ένα ειδικό πρόγραμμα ρύθμισης των παραμέτρων. Το εκκινούμε κάνοντας διπλό κλικ στο αρχείο VideoSetup.exe (εικόνα 8.1).

Εικό-
Το
γραμ-
ρύθ-
πα-

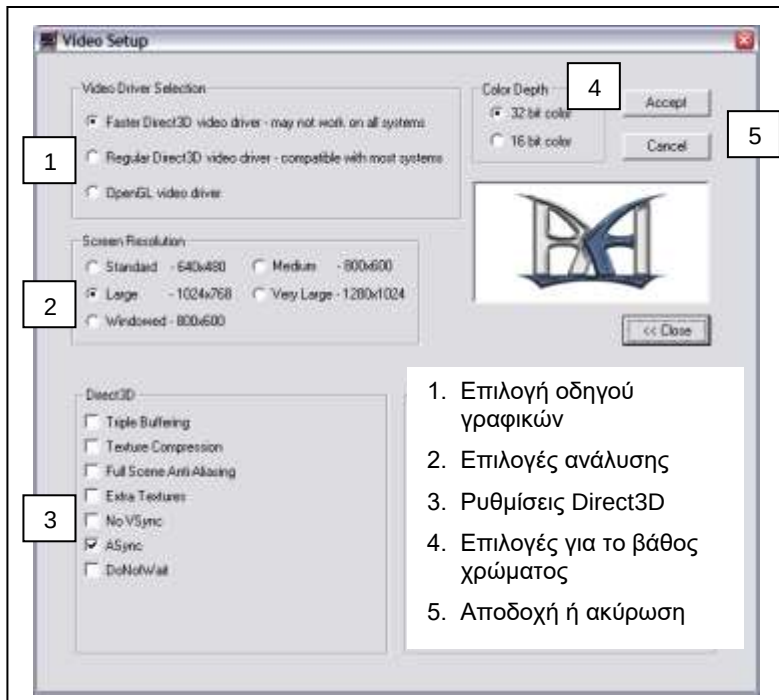


να 8.1
πρό-
μα
μησης
ραμέ-

τρων VideoSetup

Για να έχουμε πρόσβαση σ' όλες τις ρυθμίσεις, κάνουμε κλικ στο πλήκτρο Advanced (εικόνα 8.2). Αν έχουμε μία σχετικά καλή κάρτα γραφικών, μπορούμε να επιλέξουμε το γρήγορο οδηγό Direct3D, να ανεβάσουμε την ανάλυση στα 1280x1024, να επιλέξουμε 32bit βάθος χρώματος και να ενεργοποιήσουμε τις επιλογές Triple Buffering, Full Scene Anti-Aliasing και Extra Textures. Οι επιλογές No VSync, Async και DoNotWait αφορούν το μη-συγχρονισμό της κάθετης και οριζόντιας σάρωσης της οθόνης και δε συνιστάται η αλλαγή της προκαθορισμένης (Async). Υπάρχει το ενδεχόμενο κάποια από τις ρυθμίσεις που κάνουμε να οδηγήσει σε αδυναμία εκτέλεσης του Reality Factory. Αν συμβεί κάτι τέτοιο, ελέγχουμε μία προς μία τις επιλογές μας αλλάζοντας μόνο μία κάθε φορά και εκκινώντας το Reality Factory.

Τέλος, δύο πολύ σημαντικά αρχεία είναι τα RealityFactory.ini και D3d24.ini. Είναι αρχεία παραμέτρων στα οποία μπορούμε να επεμβούμε και να αλλάξουμε τις ρυθμίσεις του RF (ανοίγουν και αυτά με το "Σημειωματάριο"). Στη συνέχεια αυτού του κεφαλαίου θα παρουσιάσουμε αναλυτικά τις ρυθμίσεις που μπορούν να γίνουν.



Εικόνα 8.2 Ρυθμίσεις παραμέτρων με το πρόγραμμα VideoSetup

8.3.2 Αρχεία στους υπο-φακέλους

Συνολικά περιέχονται επτά υπο-φάκελοι στον κυρίως φάκελο του RF, που στον καθένα βρίσκονται αρχεία απαραίτητα για τη λειτουργία του προγράμματος. Αναλυτικά αυτοί είναι:

1. Στο φάκελο C:\RealityFactory\Docs περιέχεται όλη η τεκμηρίωση του προγράμματος. Το αρχείο Index.htm είναι η αρχική σελίδα. Κάνοντας διπλό κλικ σ' αυτή, εκκινεί ο Internet Explorer (ή όποιο άλλο πρόγραμμα έχουμε εγκαταστήσει ως φυλλομετρητή ιστοσελίδων) και μπορούμε να επιλέξουμε ένα από τα θέματα που περιλαμβάνονται.
2. Ο φάκελος C:\RealityFactory\Icons περιέχει τις εικόνες που χρησιμοποιούνται για να αναπαραστήσουν τα διάφορα αντικείμενα που τοποθετούμε στον εικονικό μας κόσμο κατά την κατασκευή του.
3. Ο φάκελος C:\RealityFactory\Install περιέχει αρχεία ρυθμίσεων. Στα κεφάλαια που ακολουθούν θα δούμε ρυθμίσεις που μπορούν να γίνουν στα αρχεία που περιλαμβάνει.
4. Ο φάκελος C:\RealityFactory\Media περιέχει με τη σειρά του ορισμένους υπο-φακέλους. Στο RF απαιτείται να τοποθετούνται σε αυτούς συγκεκριμένοι τύποι αρχείων. Συγκεκριμένα:

- 4.1. Στον υπο-φάκελο C:\RealityFactory\Media\Actors τοποθετούνται τα επεξεργασμένα τρισδιάστατα μοντέλα.
- 4.2. Στον υπο-φάκελο C:\RealityFactory\Media\Audio τοποθετούνται τα αρχεία των ήχων.
- 4.3. Στον υπο-φάκελο C:\RealityFactory\Media\Bitmaps τοποθετούνται τα αρχεία των εικόνων.
- 4.4. Στον υπο-φάκελο C:\RealityFactory\Media\Levels αποθηκεύονται τα αρχεία των εφαρμογών μας, καθώς επίσης και τα πακέτα υφών.
- 4.5. Στον υπο-φάκελο C:\RealityFactory\Media\Midi τοποθετούνται τα αρχεία μουσικής (με κατάληξη .mid).
- 4.6. Στον υπο-φάκελο C:\RealityFactory\Media\Video αποθηκεύονται τα αρχεία βίντεο.
5. Ο φάκελος C:\RealityFactory\Scripts περιέχει αρχεία με κατάληξη .s. Πρόκειται για εντολές γραμμένες με την προγραμματιστική γλώσσα του RF που ορίζουν τη συμπεριφορά ορισμένων αντικειμένων.
6. Ο φάκελος C:\RealityFactory\Source περιέχει ένα και μόνο αρχείο, στο οποίο ορίζονται όλα τα αντικείμενα που χρησιμοποιούνται στο RF.
7. Τέλος, ο φάκελος C:\RealityFactory\Tools περιέχει όλα τα εργαλεία που χρησιμοποιούνται για την κατασκευή των εφαρμογών μας και εκτελούνται κάνοντάς τα διπλό κλικ. Τα πιο σημαντικά απ' αυτά είναι:
 - 7.1. Το αρχείο ActView.exe, για την προεπισκόπηση των τρισδιάστατων μοντέλων του RF.
 - 7.2. Το αρχείο Astudio.exe, για την κατασκευή τρισδιάστατων μοντέλων σε μορφή που απαιτεί το RF.
 - 7.3. Το αρχείο IniEditor.exe, που χρησιμοποιείται στη ρύθμιση των παραμέτρων εκτέλεσης του RF.
 - 7.4. Το αρχείο RFEEditPro.exe, που είναι το πρόγραμμα με το οποίο κατασκευάζονται οι εικονικοί μας κόσμοι.
 - 7.5. Το αρχείο rfPack.exe, με το οποίο κατασκευάζουμε πακέτα υφών.
 - 7.6. Το αρχείο RFTrayBar.exe, το οποίο τοποθετεί ένα εικονίδιο δίπλα στο ρολόι (κάτω δεξιά στην οθόνη μας) και μας δίνει πρόσβαση –σε μορφή αναδιπλούμενης εργαλειοθήκης– σ' όλα τα προγράμματα του RF, χωρίς να χρειάζεται να αναζητούμε τα αντίστοιχα αρχεία στην “Εξερεύνηση των Windows” ή από την “Εναρξη”, “Όλα τα προγράμματα”.
 - 7.7. Το αρχείο RFvfs.exe. Στο RF υπάρχει η δυνατότητα να διανεύουμε την εφαρμογή μας σ' ένα και μόνο αρχείο (με κατάληξη .vfs). Με το RFvfs.exe κατασκευάζουμε τέτοια αρχεία.
 - 7.8. Το αρχείο Texture Image converter.exe, που χρησιμοποιείται για την αλλαγή των διαστάσεων και του τύπου αρχείων των εικόνων μας.

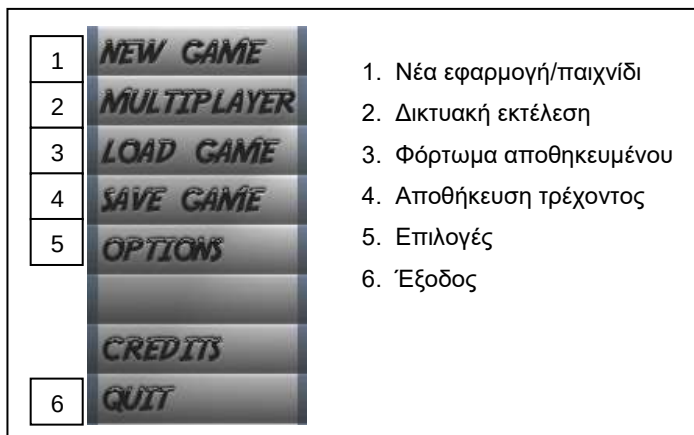
8.4 Αρχικές παραμετροποιήσεις

Κατά την εγκατάστασή του το RF χρησιμοποιεί προεπιλεγμένες ρυθμίσεις και παραμέτρους εκτέλεσης τις οποίες μπορούμε να αλλάξουμε. Για το σκοπό αυτό θα χρειαστεί να επέμβουμε τόσο στις επιλογές του RF όσο και σε ορισμένα αρχεία.

Πριν γίνει οποιαδήποτε επέμβαση σε κάποιο αρχείο του RF, πρέπει πάντοτε να δημιουργούμε ένα αντίγραφο ασφαλείας (με αντιγραφή και επικόλληση στο ίδιο ή σε άλλο σημείο του σκληρού μας δίσκου). Έτσι θα μπορούμε να επαναφέρουμε άμεσα τις προηγούμενες ρυθμίσεις, αν οι αλλαγές που κάναμε οδήγησαν σε δυσλειτουργίες.

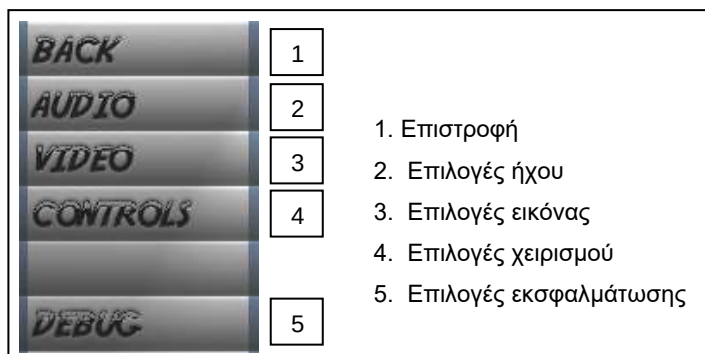
8.4.1 Ρυθμίσεις στο Reality Factory

Μία πρώτη ομάδα ρυθμίσεων που μπορούν να γίνουν, αφορά τα πλήκτρα χειρισμού και ρυθμίσεις για το ποντίκι, τον ήχο και την ποιότητα της εικόνας. Για να κάνουμε τα παραπάνω, εκκινούμε το RF (είτε από την “Εναρξη”, “Όλα τα προγράμματα” είτε από την εργαλειοθήκη-Tray bar είτε κάνοντας διπλό κλικ στο αρχείο RealityFactory.exe). Μετά από τις αρχικές οθόνες, εμφανίζεται η οθόνη του κυρίως μενού του RF (εικόνα 8.3).



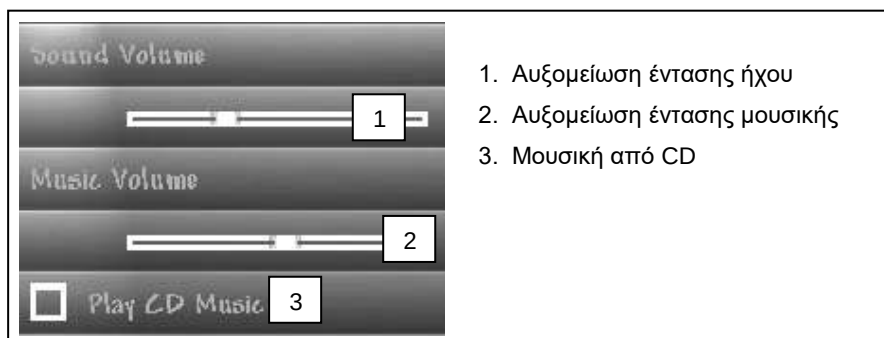
Εικόνα 8.3 Το κυρίως μενού του RF

Κάνουμε κλικ στο Options/Επιλογές και οδηγούμαστε στην οθόνη για τις επιλογές ρυθμίσεων ήχου, εικόνας, χειρισμού/πλήκτρων και εκσφαλμάτωσης (εικόνα 8.4).



Εικόνα 8.4 Το μενού επιλογών

Οι ρυθμίσεις που μπορούν να γίνουν στους ήχους είναι η αυξομείωση της έντασης του ήχου των διαφόρων αντικειμένων και της μουσικής (μετακινώντας τις αντίστοιχες μπάρες κύλισης) και η δυνατότητα να παίζει μουσική που υπάρχει σε CD τσεκάροντας το τετραγωνάκι (εικόνα 8.5).

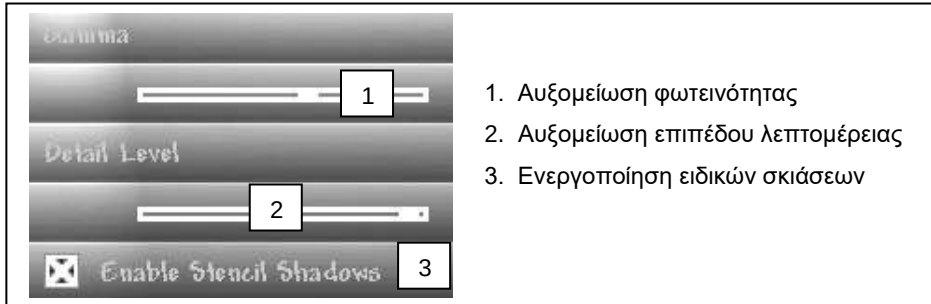


Εικόνα 8.5 Ρυθμίσεις ήχου

Οι καταχωρίσεις των ρυθμίσεών μας γίνονται πάντα πατώντας το πλήκτρο

Back

Αντίστοιχα, οι ρυθμίσεις για την ποιότητα της εικόνας είναι η αυξομείωση της φωτεινότητας, η αυξομείωση του επιπέδου λεπτομέρειας και η ενεργοποίηση μιας ειδικής κατηγορίας λεπτομερών σκιάσεων (stencil shadows) (εικόνα 8.6).



Εικόνα 8.6 Ρυθμίσεις εικόνας

Στις επιλογές χειρισμού και πλήκτρων μπορούμε να ορίσουμε τα πλήκτρα κίνησης του χαρακτήρα που ελέγχουμε στην εφαρμογή, για παράδειγμα τα πλήκτρα με τα οποία κάνουμε βήματα δεξιά και αριστερά, πηδάμε, τρέχουμε, κινούμαστε μπροστά ή πίσω, κτλ. Επίσης μπορούμε να ορίσουμε και τα πλήκτρα με τα οποία εκτελούμε ορισμένες ενέργειες, για παράδειγμα αλλαγή της προοπτικής της κάμερας, αύξηση και μείωση του ζουμ, γρήγορη αποθήκευση και φόρτωση, κτλ.

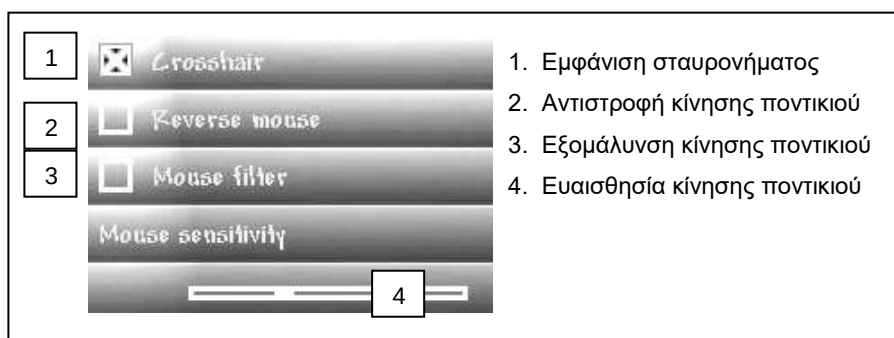
Για να αλλάξουμε κάποιο πλήκτρο, κάνουμε κλικ για να επιλέξουμε σε ποια ενέργεια θα γίνει η αλλαγή, πατάμε Enter και στη συνέχεια πατάμε το νέο πλήκτρο που επιθυμούμε να εκτελεί τη συγκεκριμένη ενέργεια. Για να δούμε όλα τα πλήκτρα, κάνουμε συνεχόμενα κλικ στο πάνω ή στο κάτω βελάκι της μπάρας κύλισης (εικόνα 8.7). Όταν αλλάζουμε κάποιο πλήκτρο, θα πρέπει να προσέξουμε μην τυχόν το νέο πλήκτρο που ορίσαμε είχε ήδη ανατεθεί σε μία άλλη ενέργεια. Στην περίπτωση αυτή, η ενέργεια θα εμφανιστεί ότι δεν έχει ορισμένο κάποιο πλήκτρο (Empty).

Το RF επιτρέπει 6 βαθμούς ελευθερίας στις κινήσεις, δηλαδή κίνηση μπροστά, πίσω, δεξιά και αριστερά και κοίταγμα πάνω και κάτω. Απ' αυτές, η κίνηση μπροστά και πίσω και τα βήματα δεξιά και αριστερά ελέγχονται με πλήκτρα ενώ το κοίταγμα δεξιά, αριστερά, πάνω και κάτω ελέγχονται από το ποντίκι.



Εικόνα 8.7 Ρυθμίσεις πλήκτρων

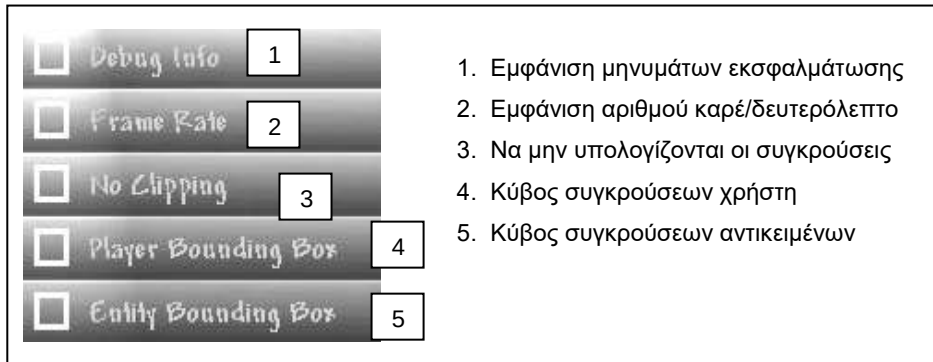
Στο ίδιο μενού υπάρχει η επιλογή για περισσότερες ρυθμίσεις (*Advanced*). Κάνοντας κλικ σ' αυτή, οδηγούμαστε σ' ένα μενού στο οποίο μπορούμε να ορίσουμε την εμφάνιση του σταυρονήματος, να αντιστρέψουμε την κίνηση του ποντικιού (κινώντας το ποντίκι κάτω βλέπουμε επάνω και αντίστροφα), να εξομαλύνουμε την κίνηση του ποντικιού και να αυξήσουμε την ευαισθησία του (με μικρότερη κίνηση θα κινείται περισσότερο) (εικόνα 8.8).



Εικόνα 8.8 Περισσότερες ρυθμίσεις

Το μενού εκσφαλμάτωσης περιλαμβάνει επιλογές χρήσιμες για τον έλεγχο και τη διόρθωση σφαλμάτων στην εφαρμογή μας, καθώς επίσης και για τον έλεγχο των επιδόσεων (εικόνα 8.9). Έτσι, μπορούμε: α) να εμφανίσουμε μηνύματα εκσφαλμάτωσης

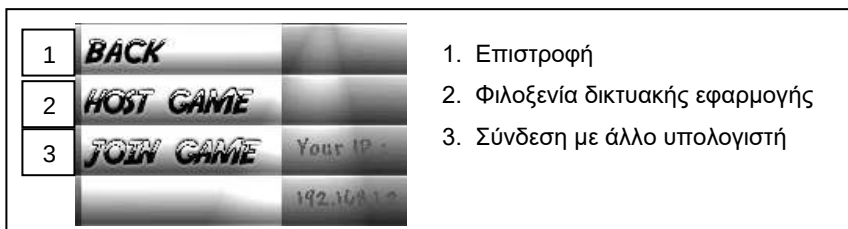
κατά την εκτέλεση της εφαρμογής, β) να εμφανίσουμε τον αριθμό των καρτέ ανά δευτερόλεπτο για να ελέγξουμε αν η εφαρμογή “τρέχει” με ικανοποιητική ταχύτητα, γ) να υποχρεώσουμε την εφαρμογή να μην ελέγχει τις συγκρούσεις του χαρακτήρα με άλλα αντικείμενα (να περνά μέσα απ' αυτά), δ) να δούμε τον κύβο υπολογισμού συγκρούσεων που περιβάλλει το χαρακτήρα και ε) να δούμε τον κύβο υπολογισμού συγκρούσεων που περιβάλλει τα άλλα αντικείμενα.



Εικόνα 8.9 Το μενού εκσφαλμάτωσης

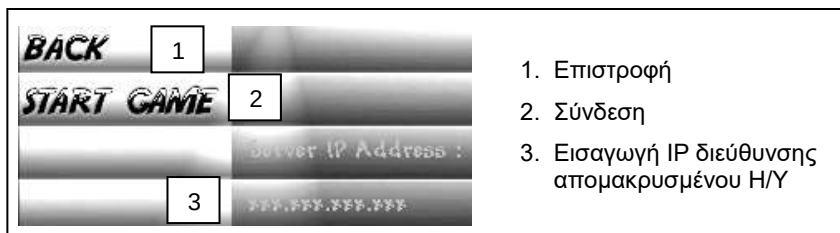
Αν επιστρέψουμε στην αρχική οθόνη και κάνουμε κλικ στην επιλογή της δικτυακής εκτέλεσης (*Multiplayer*), θα οδηγηθούμε σ' ένα υπο-μενού που περιλαμβάνει τις επιλογές σύνδεσης με άλλο υπολογιστή και της φιλοξενίας της εφαρμογής (εικόνα 8.10). Στην πρώτη περίπτωση θα πρέπει να δώσουμε την IP διεύθυνση του υπολογιστή που επιθυμούμε να συνδεθούμε (την οποία και θα πρέπει να γνωρίζουμε εκ των προτέρων). Αυτό γίνεται κάνοντας κλικ στα αστεράκια, γράφοντας τη διεύθυνση (για παράδειγμα 192.168.1.2), πατώντας Enter και κάνοντας κλικ στο “Start Game” (εικόνα 8.11). Αν επιλέξουμε “Host Game”, άλλοι υπολογιστές θα μπορούν να συνδεθούν στο δικό μας.

Εικόνα 8.10
με-
Mul-



να
Το
νού

tiplayer



Εικόνα 8.11 Το μενού σύνδεσης με άλλο υπολογιστή

Τέλος, η επιλογή New Game του αρχικού μενού, μάς επιτρέπει να επιλέξουμε το χαρακτήρα που θα χρησιμοποιήσουμε (εικόνα 8.12). Μπορούμε επίσης να του δώσουμε και διαφορετικό όνομα απ' αυτό που προτείνεται, κάνοντας κλικ επάνω στο προεπιλεγμένο όνομα και γράφοντας ένα άλλο (εικόνα 8.13).



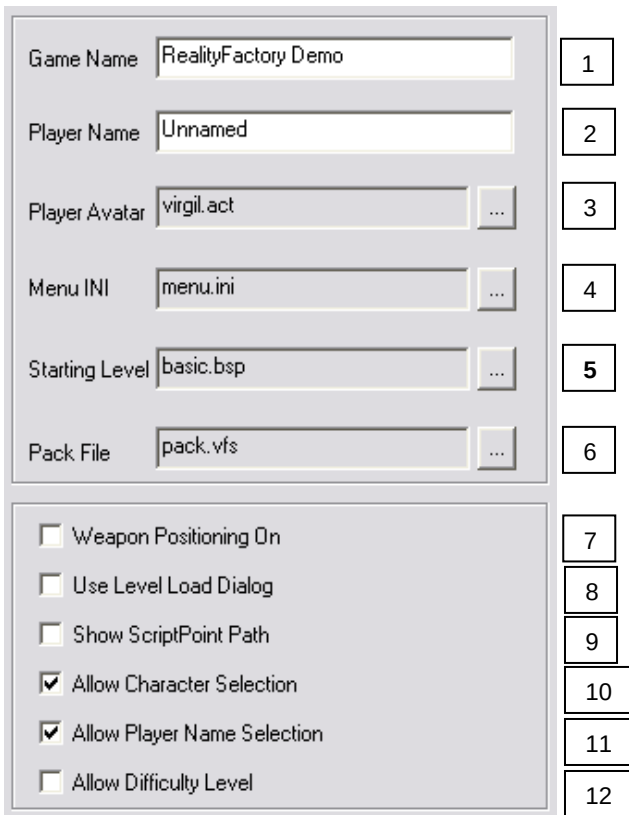
Εικόνα 8.12 Επιλογή χαρακτήρα



Εικόνα 8.13 Αλλαγή ονόματος

8.4.2 Ρυθμίσεις με το πρόγραμμα IniEditor

Στο φάκελο των εργαλείων (C:\RealityFactory\Tools) υπάρχει το αρχείο IniEditor.exe, το οποίο μας επιτρέπει περαιτέρω ρυθμίσεις. Κάνοντάς το διπλό κλικ εκκινεί (εικόνες 8.14α και 8.14β). Πρέπει να σημειώσουμε ότι κάποιες από τις επιλογές δε λειτουργούν, γιατί πλέον υλοποιούνται με άλλο τρόπο. Κάποιες άλλες επιλογές χρησιμοποιούνται μόνο κατά τη διαδικασία εύρεσης και διόρθωσης σφαλμάτων της εφαρμογής μας (debugging process).



Εικόνα 8.14α Το πρόγραμμα IniEditor

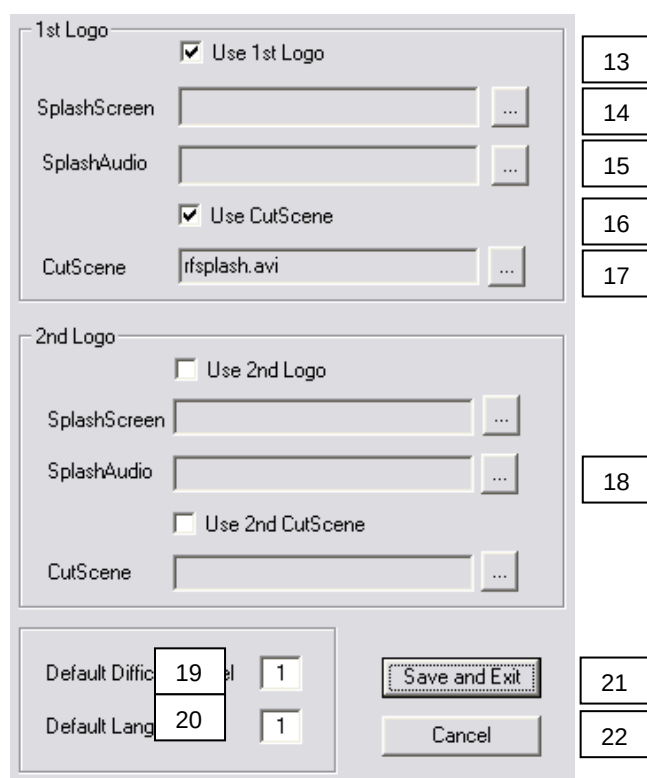
1. Το όνομα της εφαρμογής/παιχνιδιού.
2. Το όνομα του χρήστη /παίκτη.
3. Το αρχείο του χαρακτήρα που χρησιμοποιείται στην εφαρμογή. Κάνοντας κλικ στις τρεις τελείτσες μπορούμε να επιλέξουμε άλλο αρχείο /χαρακτήρα.
4. Ποιο αρχείο θα χρησιμοποιηθεί για τα μενού των επιλογών που παρουσιάστηκαν στο κεφάλαιο 8.4.1. Αν δεν έχουμε κατασκευάσει τέτοιο αρχείο, δεν πρέπει να γίνει καμία αλλαγή.
5. Ποιο επίπεδο/εφαρμογή θα φορτώνει με την έναρξη του RF.
6. Ποιο αρχείο vfs (συμπίεσμένα αρχεία εφαρμογής) θα χρησιμοποιηθεί. Αν δεν έχουμε κατασκευάσει τέτοιο αρχείο, δε χρειάζεται να δηλωθεί τίποτα.

7. Η επιλογή χρησιμοποιείται κατά τη διαδικασία εκσφαλμάτωσης για να γίνουν μικρο-ρυθμίσεις στη θέση του όπλου επάνω στο χαρακτήρα. Δε θα χρειαστούμε μία τέτοια ρύθμιση.

8. Εάν τσεκάρουμε το τετραγωνάκι, πριν από την εκτέλεση του RF θα εμφανιστεί ένα παράθυρο διαλόγου για να επιλέξουμε ποια εφαρμογή θα φορτωθεί.

9. Ενεργοποιημένη η επιλογή μάς επιτρέπει κατά τη διαδικασία εκσφαλμάτωσης να δούμε ποια διαδρομή ορίσαμε να ακολουθούν ορισμένα αντικείμενα στην εφαρμογή μας.

10. Εάν τσεκαριστεί το τετραγωνάκι, θα μπορούμε να επιλέξουμε και άλλους χαρακτήρες εκτός από τον προεπιλεγμένο. Χρησιμοποιείται παράλληλα με ρυθμίσεις που έχουμε κάνει στο αρχείο character.ini που βρίσκεται στο φάκελο C:\RealityFactory\Install που αναλύεται σε επόμενο κεφάλαιο. Το αρχείο αυτό περιέχει και άλλους χαρακτήρες.



Εικόνα 8.14β Το πρόγραμμα IniEditor

Media\Bitmaps.

15. Ομοίως, ποιος ήχος θα χρησιμοποιηθεί. Πρέπει να βρίσκεται στο φάκελο C:\RealityFactory\Media\Audio.

16. Εάν τσεκαριστεί το τετραγωνάκι, αντί για εικόνα και ήχο μπορούμε να χρησιμοποιήσουμε αρχείο βίντεο.

17. Ποιο αρχείο βίντεο θα χρησιμοποιηθεί. Θα πρέπει να βρίσκεται στο φάκελο C:\RealityFactory\Media\Video.

18. Ομοίως, οι επιλογές αφορούν τυχόν δεύτερο λογότυπο.

19. Προεπιλεγμένο επίπεδο δυσκολίας. Δε θα χρησιμοποιήσουμε αυτή την επιλογή.

20. Προεπιλεγμένη γλώσσα. Δε θα χρησιμοποιήσουμε αυτή την επιλογή.

21. Αποθήκευση των αλλαγών και έξοδος.

22. Ακύρωση των αλλαγών και έξοδος.

11. Ενεργοποιημένη η επιλογή μάς επιτρέπει να αλλάξουμε το όνομα του χαρακτήρα.

12. Ενεργοποιημένη η επιλογή μάς επιτρέπει να αλλάξουμε το επίπεδο δυσκολίας. Δε θα τη χρησιμοποιήσουμε. Αν την ενεργοποιήσουμε χωρίς όμως να έχουν γίνει ορισμένες ρυθμίσεις κατά την κατασκευή της εφαρμογής μας, τότε το RF θα τερματίσει απότομα.

13. Να χρησιμοποιηθεί κάποιο λογότυπο κατά την έναρξη του RF.

14. Ποια εικόνα να χρησιμοποιηθεί για το λογότυπο. Κάνοντας κλικ στις τρεις τελείτσες μπορούμε να επιλέξουμε κάποια. Η εικόνα πρέπει να βρίσκεται στο φάκελο C:\RealityFactory\

8.4.3 Ρυθμίσεις με επέμβαση σε αρχεία

Πολύ σημαντικές ρυθμίσεις γίνονται με απευθείας επέμβαση σε δύο σημαντικά αρχεία. Αυτά είναι το RealityFactory.ini και το d3d24.ini, που βρίσκονται στο φάκελο C:\RealityFactory. Και τα δύο μπορούμε να τα επεξεργαστούμε με το “Σημειωματάριο”. Οι κατασκευαστές του RF φρόντισαν να δίνουν μία σύντομη επεξήγηση για το τι λειτουργία επιτελεί η κάθε εντολή. Κάθε γραμμή που αρχίζει με “;” είναι είτε σχόλιο είτε ανενεργή.

Οι αλλαγές στα αρχεία πρέπει να γίνονται βήμα-βήμα. Κάνουμε μία αλλαγή σ' ένα σημείο, δοκιμάζουμε τι επιπτώσεις έχει εκτελώντας το RF και στη συνέχεια κάνουμε την επόμενη τροποποίηση. Κατ' αυτό τον τρόπο θα εξασφαλίζουμε ότι θα μπορούμε να επαναφέρουμε τις ρυθμίσεις εύκολα στην προηγούμενη κατάσταση. Και βέβαια πάντοτε δημιουργούμε αντίγραφο ασφαλείας του αρχείου που επεξεργαζόμαστε.

Ανοίγοντας το αρχείο RealityFactory.ini, βλέπουμε ότι οι πρώτες γραμμές ορίζουν στοιχεία που συναντήσαμε στο πρόγραμμα IniEditor, για παράδειγμα:

```
;
;
; GameName defines the name of the game (ορισμός του ονόματος της εφαρμογής)
;
GameName=RealityFactory Demo
;
; PlayerName defines the name of the player (ορισμός του ονόματος του χρήστη)
;
PlayerName=Unnamed
;
; PlayerAvatar defines the player actor (ορισμός του χαρακτήρα)
;
PlayerAvatar=VIRGIL.ACT
```

Στη συνέχεια ακολουθεί μία ομάδα γραμμών όπου ορίζονται οι φάκελοι στους οποίους πρέπει να είναι αποθηκευμένοι συγκεκριμένοι τύποι αρχείων, όπως ήδη αναφέραμε στην ενότητα 8.3.2:

```
LevelDirectory=media\levels
BitmapDirectory=media\bitmaps
ActorDirectory=media\actors
AudioDirectory=media\audio
```

AudioStreamDirectory=media\audio

VideoDirectory=media\video

MIDIIDirectory=media\midi

Αν αλλάξουμε κάποια από τις παραπάνω δηλώσεις, αντίστοιχα θα πρέπει να αλλάξουμε και τους φακέλους στους οποίους αποθηκεύουμε αυτά τα αρχεία. Δε συνιστούμε κάποια αλλαγή.

Μία ομάδα δηλώσεων αφορά το αν η εφαρμογή θα εκτελείται σε παράθυρο ή σε πλήρη οθόνη και την ανάλυση που θα χρησιμοποιεί:

FullScreen=true

Width=800

Height=600

Αν αλλάξουμε τη δήλωση FullScreen σε false, τότε η εφαρμογή θα εκτελείται σε παράθυρο. Η δήλωση Width δέχεται τις τιμές 640, 800, 1024, 1088, 1152, 1280, 1360, 1600, 1920, 2048. Η δήλωση Height δέχεται τις τιμές 480, 600, 612, 720, 768, 800, 864, 900, 960, 1024, 1080, 1200, 1440, 1536. Με τον τρόπο αυτό μπορούμε να δηλώσουμε τους συνηθισμένους συνδυασμούς ανάλυσης, για παράδειγμα 800x600, 1024x768, 1280x1024, κτλ. Πρέπει να σημειώσουμε ότι όσο μεγαλύτερη ανάλυση χρησιμοποιήσουμε, τόσο περισσότερο επιβαρύνεται η εφαρμογή και συνεπώς πρέπει να διαθέτουμε μία καλή κάρτα γραφικών. Μία δήλωση για Width=1024 και Height=768 μπορεί να χρησιμοποιηθεί χωρίς προβλήματα σ' όλες τις πρόσφατες κάρτες γραφικών. Αποθηκεύοντας το αρχείο και εκτελώντας το RF, μπορούμε να δούμε την αλλαγή στην ανάλυση.

Αντίστοιχα, το αρχείο d3d24.ini περιέχει τις ρυθμίσεις που έχουμε κάνει με το πρόγραμμα VideoSetup, όπως φαίνεται παρακάτω:

[D3D24]

Bpp = 32

ZBufferD = 16

BBufferCount = 1

CompressTextures = 0

FSAntiAliasing = 1

ExtraTextures = 1

NoVsync = 0

Async = 1

DoNotWait = 0

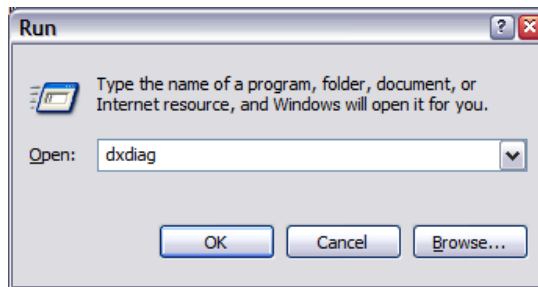
8.5 Αντιμετώπιση προβλημάτων

Το RF, παρότι είναι πρόγραμμα *ανοιχτού κώδικα*, σε αντίθεση με άλλα προγράμματα αυτού του είδους, δεν παρουσιάζει μεγάλο αριθμό προβλημάτων. Παρ' όλα αυτά, ο χρήστης μπορεί να βρεθεί στη δυσάρεστη κατάσταση το πρόγραμμα να μην εκτελείται καθόλου μετά την εγκατάστασή του. Πρέπει να διευκρινίσουμε ότι αναφερόμαστε στο RealityFactory.exe (τη μηχανή δηλαδή) και όχι το πρόγραμμα RFEditionPro, με το οποίο κατασκευάζουμε εφαρμογές, που δεν παρουσιάζει προβλήματα εκτέλεσης.

Ένα πρώτο σημείο που πρέπει να ελέγξουμε είναι αν η εγκατάσταση έχει γίνει σωστά, αν έχουμε ακολουθήσει τα βήματα που περιγράφονται στο κεφάλαιο 8.2 και κυρίως ό,τι αφορά την αντικατάσταση των αρχείων κατά την αναβάθμιση στην έκδοση 0.75C. Αν δεν είμαστε σίγουροι, μπορούμε να απο-εγκαταστήσουμε το πρόγραμμα και να δοκιμάσουμε νέα εγκατάστασή του.

Πρέπει επίσης να είμαστε βέβαιοι ότι στον υπολογιστή μας είναι εγκατεστημένοι οι τελευταίοι οδηγοί της κάρτας γραφικών που διαθέτουμε. Μπορούμε να επισκεφθούμε το δικτυακό τόπο του κατασκευαστή, να “κατεβάσουμε” και να εγκαταστήσουμε την πιο πρόσφατη έκδοση. Το ίδιο ισχύει και για το DirectX, που η έκδοσή του πρέπει να είναι τουλάχιστον η 8.1 (τη στιγμή που γραφόταν το βιβλίο η τελευταία έκδοση ήταν η 10.0). Αν δε γνωρίζουμε ποια έκδοση έχουμε εγκατεστημένη, κάνουμε τα εξής βήματα:

1. Από την “Εναρξη” επιλέγουμε “Εκτέλεση”.
2. Στο παράθυρο διαλόγου που εμφανίζεται πληκτρολογούμε dxdiag και πατάμε OK (εικόνα 8.15).



Εικόνα 8.15 Εκτέλεση διαγνωστικού εργαλείου dxdiag

3. Στην καρτέλα που εμφανίζεται εντοπίζουμε την έκδοση του DirectX που έχουμε εγκατεστημένη (εικόνα 8.16).



Εικόνα 8.16 Η έκδοση του DirectX

- Μπορούμε να “κατεβάσουμε” την τελευταία έκδοση από το δικτυακό τόπο της Microsoft (<http://www.microsoft.com/windows/directx/default.mspx>).

Ένα τρίτο σημείο που πρέπει να ελεγχθεί είναι αν ο υπολογιστής στον οποίο εγκαταστήσαμε το RF ικανοποιεί τις προτεινόμενες ή τις ελάχιστες προδιαγραφές που αναφέραμε στην αρχή αυτού του κεφαλαίου. Επίσης, υπάρχει πρόβλημα με ορισμένες παλαιότερες κάρτες γραφικών που είναι ενσωματωμένες στη μητρική πλακέτα (*onboard*) και κυρίως με τις SiS620, Intel 82810 και Intel 82815. Παρόμοιο πρόβλημα υπάρχει και με ορισμένες κάρτες γραφικών που είναι ενσωματωμένες σε φορητούς υπολογιστές. Για να αντιμετωπίσουμε το πρόβλημα, πρέπει να επέμβουμε στο αρχείο *RealityFactory.ini*:

- Εντοπίζουμε τη γραμμή με τη δήλωση *Driver=d3d*.
- Την αλλάζουμε σε *Driver=d3d16* (θα υπάρξει σχετική υποβάθμιση της ποιότητας των γραφικών) και αποθηκεύουμε το αρχείο.
- Αν και αυτή η αλλαγή δεν επιτρέψει το RF να εκτελεστεί, αλλάζουμε τη δήλωση σε *Driver=opengl* (έχει σημαντική επίπτωση στην ποιότητα των γραφικών).
- Επίσης, ίσως να χρειαστεί να επέμβουμε και στο αρχείο *d3d24.ini*, αλλάζοντας τη δήλωση *ZBufferD = 24* σε *ZBufferD = 16*.

Ένα πρόβλημα που εμφανίζεται σε υπολογιστές με καινούργιες κάρτες γραφικών της NVidia είναι ότι τα μενού “τρεμοπαίζουν”. Για την αντιμετώπισή του, ανοίγουμε το αρχείο *d3d24.ini* και αλλάζουμε τις δηλώσεις ως εξής:

ExtraTextures = 1

NoVsync = 0

Async = 1

DoNotWait = 0

Τέλος, για κάθε άλλο πρόβλημα ή απορία, μπορούμε να καταφύγουμε στο φόρουμ συζητήσεων του RF (<http://www.realityfactory.info/forum/>). Η κοινότητα των χρηστών του RF είναι ιδιαίτερα ενεργή και πρόθυμη να απαντήσει στις όποιες ερωτήσεις.

Κεφάλαιο 9^ο

Κατασκευή εικόνων/υφών και πακέτων υφών

Οι υφές (*textures*) είναι εικόνες που “ντύνουν” τα τρισδιάστατα μοντέλα και γενικά όλα τα αντικείμενα του εικονικού κόσμου, προσδίδοντας σε αυτά τα ιδιαίτερα χαρακτηριστικά τους. Για παράδειγμα, αν τοποθετήσουμε στον εικονικό μας κόσμο ένα ορθογώνιο παραλληλεπίπεδο για να αναπαριστά έναν τοίχο, θα πρέπει να το “ντύσουμε” με μία εικόνα τούβλων. Ένα άλλο ορθογώνιο παραλληλεπίπεδο, για να αναπαριστά δάπεδο, θα πρέπει να το ντύσουμε” με εικόνες από σανίδια ή πλακάκια.

Ο τρόπος με τον οποίο μία υφή/εικόνα “ντύνει” ένα αντικείμενο καθορίζεται και ελέγχεται είτε από το πρόγραμμα κατασκευής τρισδιάστατων μοντέλων είτε από το πρόγραμμα κατασκευής εικονικών κόσμων. Παρ' όλα αυτά, απαιτείται επεξεργασία στις εικόνες μας πριν αυτές χρησιμοποιηθούν από τα παραπάνω προγράμματα. Το είδος και η έκταση της επεξεργασίας εξαρτάται από τις ιδιαιτερότητες του προγράμματος και από το ρόλο που θα παίξει η υφή στην εφαρμογή μας.

Στο RF οι διάφορες εικόνες δεν χρησιμοποιούνται απευθείας. Αντίθετα, πρέπει πρώτα να κατασκευάσουμε ένα πακέτο, μία βιβλιοθήκη υφών, η οποία θα καλείται από το RF. Απ' αυτή τη βιβλιοθήκη επιλέγουμε μία ή παραπάνω εικόνες για να τις τοποθετήσουμε πάνω σ' ένα αντικείμενο. Επιπρόσθετα, οι εικόνες που συγκροτούν το πακέτο υφών πρέπει υποχρεωτικά να ικανοποιούν δύο συνθήκες:

1. Να είναι εικόνες Bitmap (δηλαδή με κατάληξη .bmp) και
2. Να έχουν διαστάσεις σε εικονοστοιχεία (*pixels*) που είναι δυνάμεις του 2 ($2^6=64$, $2^7=128$, $2^8=256$, $2^9=512$, $2^{10}=1024$). Μία εικόνα μπορεί να έχει, για παράδειγμα, διαστάσεις 256x256, 128x512 αλλά και ό,τι άλλο συνδυασμό προκύπτει από τους παραπάνω αριθμούς.

Να σημειώσουμε ότι δε συνιστάται η χρήση πολλών υφών σε μία εφαρμογή που η μία ή και οι δύο διαστάσεις τους να είναι ίσες ή να υπερβαίνουν τα 1.024 εικονοστοιχεία λόγω της σημαντικής επιβάρυνσης που προκύπτει για την εφαρμογή.

Το βασικό ερώτημα που μπορεί να θέσει κάποιος είναι το πού μπορούμε να βρούμε εικόνες για να κατασκευάσουμε υφές ή αν υπάρχουν έτοιμες υφές. Κατ' αρχάς, υπάρχουν πολλοί δικτυακοί τόποι που δίνουν δωρεάν έτοιμες υφές. Για παράδειγμα, αυτοί που αναφέραμε στο προηγούμενο κεφάλαιο, που προσφέρουν και τρισδιάστατα μοντέλα. Για να αναφέρουμε και μερικούς άλλους, υφές μπορούν να βρεθούν και στους παρακάτω:

- <http://www.mega-tex.nl/textures.php>
- <http://www.mayang.com/textures/>
- <http://www.vb3d.com/Textures.html>

- <http://www.imageafter.com/>
- <http://www.digitalcraftsman.com/textureBin/textureBin.htm>

Ένας δεύτερος τρόπος είναι να βρούμε στο Διαδίκτυο εικόνες ή να τραβήξουμε φωτογραφίες, να τις ψηφιοποιήσουμε μ' ένα *scanner* αν δεν είναι ήδη ψηφιακές, να απομονώσουμε μ' ένα πρόγραμμα επεξεργασίας εικόνων το σημείο που μας ενδιαφέρει και να το αποθηκεύσουμε ως ξεχωριστή εικόνα.

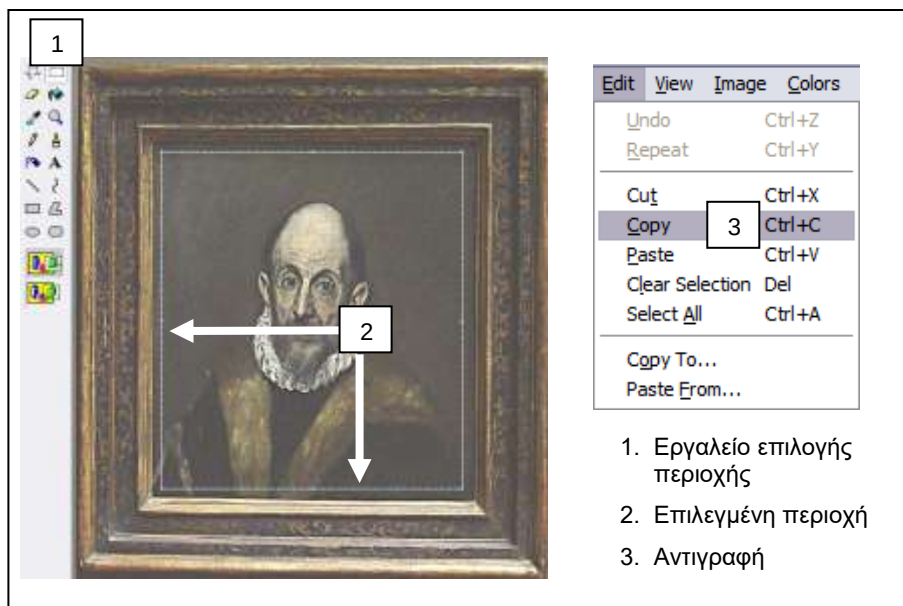
9.1 Υφές που θα καλύψουν μικρές ή μεγάλες επιφάνειες

Υφές θα χρειαστούμε είτε για να καλύψουμε μία μικρή επιφάνεια, για παράδειγμα μία πόρτα, έναν πίνακα ζωγραφικής, ένα αντικείμενο είτε για να καλύψουμε μεγάλες επιφάνειες, για παράδειγμα τοίχους, δάπεδα, οροφές. Τα βήματα που ακολουθούμε και στις δύο περιπτώσεις είναι τα ίδια, με την εξαίρεση ότι για τις μεγάλες επιφάνειες χρειάζεται να προσέξουμε ένα επιπλέον σημείο. Θα αναλύσουμε και τις δύο αυτές περιπτώσεις, παρουσιάζοντας τον τρόπο εργασίας τόσο με τη “Ζωγραφική” των Windows όσο και με το Deep Paint.

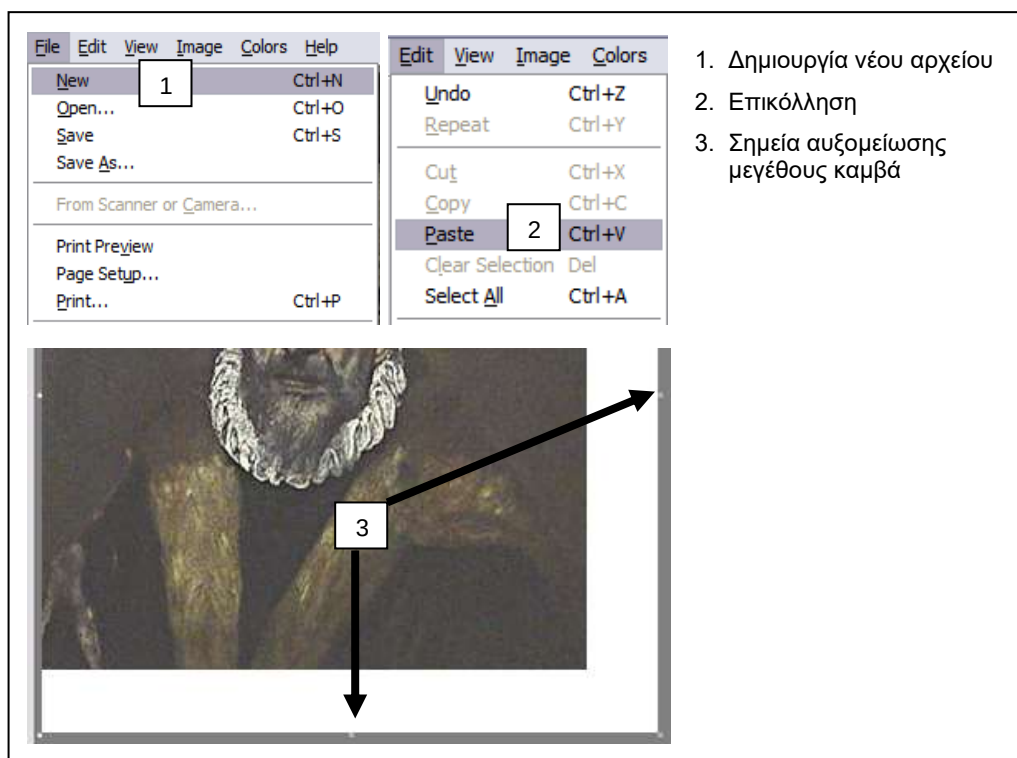
9.1.1 Επεξεργασία υφών για μικρές επιφάνειες

Ας υποθέσουμε ότι σε κάποιο σημείο του εικονικού μας κόσμου θέλουμε να τοποθετήσουμε έναν πίνακα ζωγραφικής. Κάναμε μία αναζήτηση στο Διαδίκτυο και βρήκαμε την εικόνα που μας ενδιαφέρει. Εκκινούμε τη “Ζωγραφική” των Windows και ανοίγουμε την εικόνα. Χρησιμοποιώντας το εργαλείο επιλογής , επιλέγουμε την περιοχή που θέλουμε και την αντιγράφουμε (εικόνα 9.1).

Δημιουργούμε ένα νέο αρχείο και επικολλούμε την επιλεγμένη περιοχή. Στη “Ζωγραφική” των Windows υπάρχει η περίπτωση ο καμβάς (η άσπρη επιφάνεια) να είναι μεγαλύτερος ή μικρότερος από το μέγεθος της εικόνας που επικολλούμε. Αυξομειώνουμε το μέγεθος του καμβά τραβώντας και σέρνοντας τα σημεία στις άκρες του, φροντίζοντας να μη μείνει καθόλου άσπρο (εικόνα 9.2). Υπάρχει το ενδεχόμενο, αν ο καμβάς είναι μικρότερος από την επικολλώμενη εικόνα, να χρειαστεί πρώτα να μεγαλώσουμε τον καμβά και ύστερα να κάνουμε επικόλληση. Αποθηκεύουμε την εικόνα μας σε μορφή 24bit bitmap (.bmp).



Εικόνα 9.1 Επιλογή περιοχής σε εικόνα και αντιγραφή



Εικόνα 9.2 Επικόλληση και αυξομείωση μεγέθους καμβά

Το πρόβλημα με την παραπάνω διαδικασία είναι ότι δεν είμαστε σίγουροι για τις διαστάσεις της εικόνας που μόλις κατασκευάσαμε. Για το λόγο αυτό, στο RF υπάρχει ειδικό εργαλείο μετατροπής των διαφόρων τύπων αρχείων εικόνας σε .bmp και αλλαγής των διαστάσεών τους. Αυτό είναι το Texture Image Converter και το εκκινούμε από το C:\RealityFactory\Tools (εικόνα 9.3). Το συγκεκριμένο εργαλείο, που μπορεί να “διαβάσει” εικόνες .bmp και .jpg, μας επιτρέπει και τη μαζική επιλογή και μετατροπή εικόνων. Δυστυχώς δε μας επιτρέπει να ορίσουμε διαστάσεις που δεν είναι ίσες μεταξύ τους (για παράδειγμα 512x256). Ανοίγουμε την προς μετατροπή εικόνα, ορίζουμε τις επιθυμητές διαστάσεις από τον επιλογέα και κάνουμε κλικ στο πλήκτρο μετατροπής



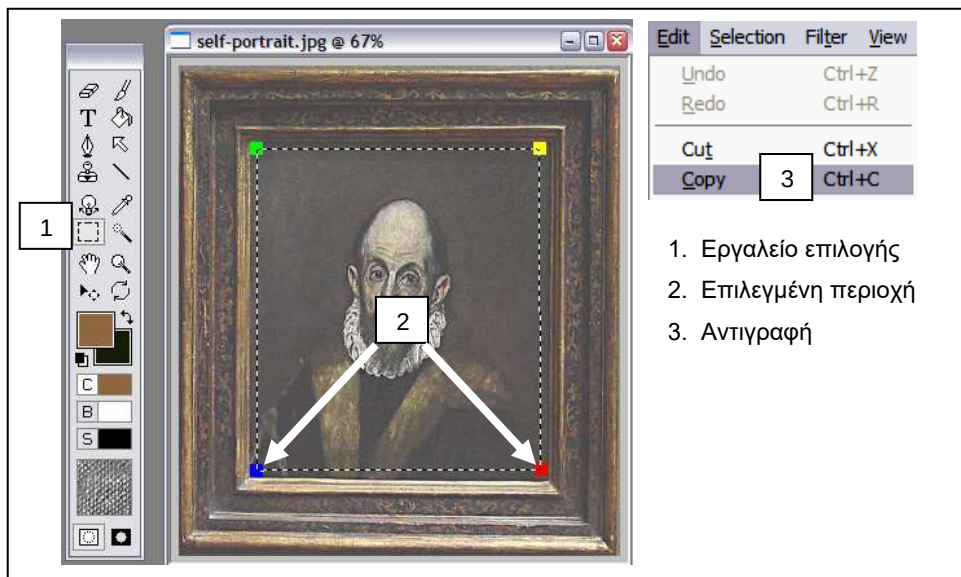
- 1 & 2. Αναζήτηση και επιλογή της προς μετατροπή εικόνας
3. Προεπισκόπηση εικόνας
4. Επιλογέας διαστάσεων
5. Πλήκτρο μετατροπής
6. Φάκελος αποθήκευσης της εικόνας που έχει μετατραπεί (δεν αλλάζει)

Εικόνα 9.3 Το πρόγραμμα Texture Image Converter

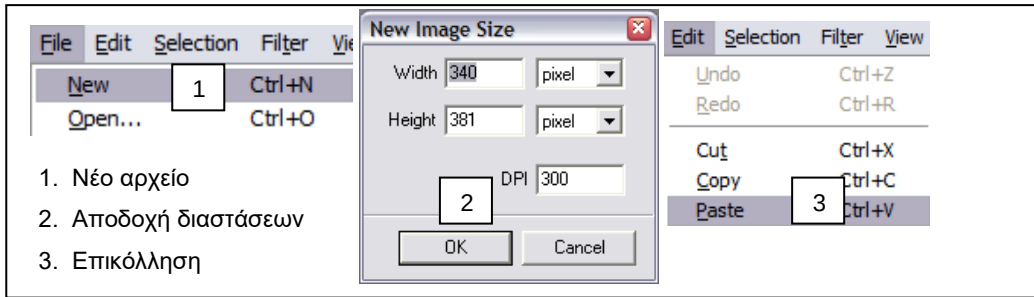
Αν επιλέξουμε να εργαστούμε με το Deep Paint, τα βήματα που ακολουθούμε είναι παρόμοια. Ανοίγουμε την εικόνα που θέλουμε να επεξεργαστούμε είτε από το εικονίδιο  είτε από την γραμμή εντολών (εικόνα 9.4), επιλέγουμε την περιοχή που επιθυμούμε  (εικόνα 9.5), την αντιγράφουμε, δημιουργούμε νέο αρχείο και την επικολλούμε (εικόνα 9.6).



Εικόνα 9.4 Ανοιγμα εικόνας



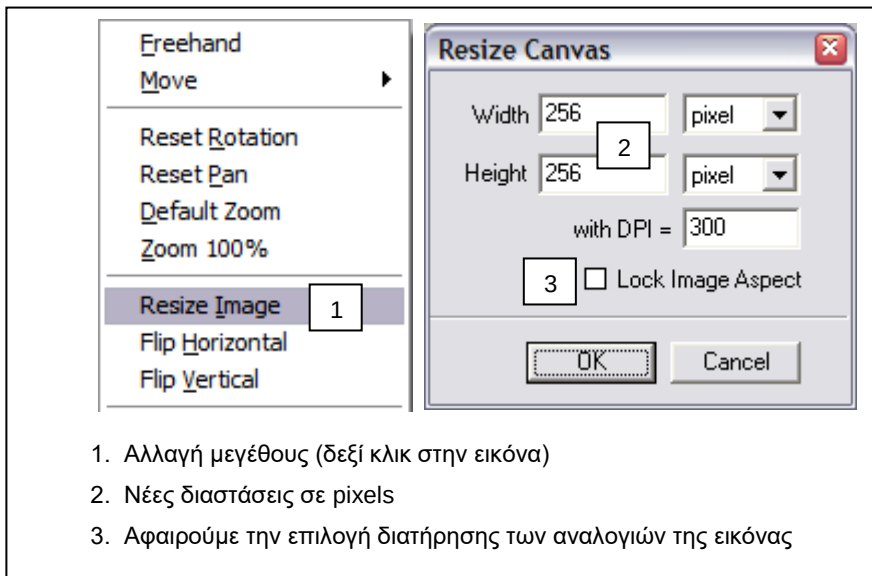
Εικόνα 9.5 Επιλογή και αντιγραφή περιοχής



Εικόνα 9.6 Επικόλληση περιοχής

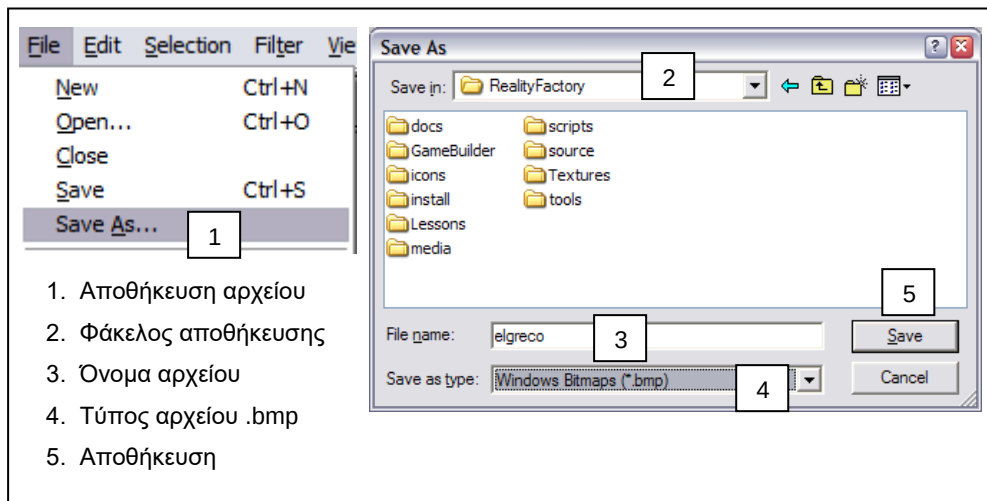
Η διαφορά είναι ότι το Deep Paint (και άλλα πιο πλήρη από τη “Ζωγραφική” προγράμματα) μας επιτρέπει να ορίσουμε τις ακριβείς διαστάσεις της εικόνας μας και έτσι δεν είναι απαραίτητη η χρήση του Texture Image Converter. Η αλλαγή στις διαστάσεις της εικόνας γίνεται ως εξής:

1. Κάνουμε κλικ ξανά στο εργαλείο επιλογής περιοχής .
2. Κάνουμε δεξί κλικ επάνω στην εικόνα.
3. Από το μενού των επιλογών που εμφανίζεται κάνουμε κλικ στο “Resize image”.
4. Στο παράθυρο του διαλόγου, δίνουμε τις νέες διαστάσεις σε pixels (εικόνα 9.7).



Εικόνα 9.7 Αλλαγή διαστάσεων εικόνας

Όπως και με τη “Ζωγραφική” των Windows, έτσι και με το Deep Paint, αποθηκεύουμε τη νέα εικόνα είτε από το εικονίδιο  είτε από τη γραμμή εντολών, σε μορφή .bmp (εικόνα 9.8).

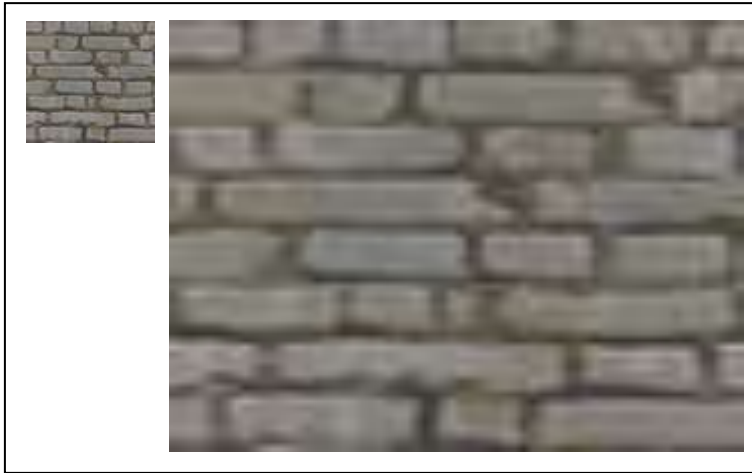


Εικόνα 9.8 Αποθήκευση νέας εικόνας

9.1.2 Επεξεργασία υφών για μεγάλες επιφάνειες

Η διαδικασία επεξεργασίας των εικόνων που θα “ντύσουν” μεγάλες επιφάνειες είναι η ίδια με αυτή που περιγράφηκε στην προηγούμενη ενότητα, όμως χρειάζεται προσοχή στον τρόπο που θα τις επεξεργαστούμε. Στο RF και σ’ όλα τα προγράμματα κατασκευής εικονικών κόσμων και επεξεργασίας τρισδιάστατων γραφικών, όταν μία υφή είναι μικρότερη από τις διαστάσεις της επιφάνειας που θα “ντύσει”, υπάρχουν δύο επιλογές: α) η εικόνα να “διασταλεί” έτσι ώστε να καλύψει όλη την επιφάνεια και β) η εικόνα να “επαναληφθεί” τόσες φορές όσες χρειάζεται για να καλυφθεί όλη η επιφάνεια.

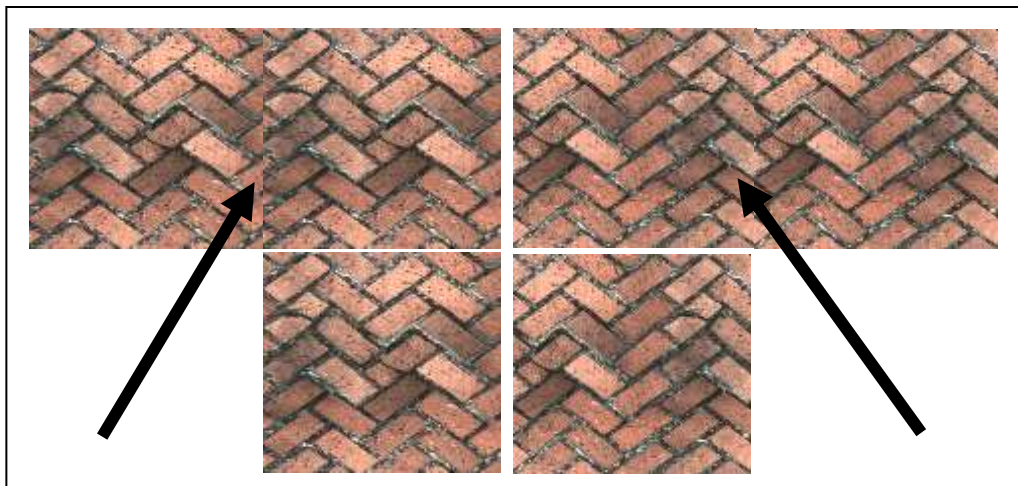
Στην πρώτη επιλογή δε χρειάζεται κάποια μορφή επεξεργασίας στις εικόνες (εκτός βέβαια από τη μετατροπή τους σε .bmp με τις κατάλληλες διαστάσεις), αλλά μας οδηγεί σ’ ένα σημαντικό πρόβλημα. Μία εικόνα μικρών διαστάσεων, όταν τη μεγαλώσουμε κατά τη μία ή και τις δύο διαστάσεις της, χάνει πολύ σε ποιότητα, γίνεται θολή, παραμορφώνεται, εμφανίζονται “δοντάκια” (εικόνα 9.9). Προφανώς, όσο μεγαλύτερη είναι η επιφάνεια, τόσο χειρότερη γίνεται η ποιότητα της εικόνας. Συνεπώς, αυτή η επιλογή δεν μπορεί να χρησιμοποιηθεί.



Εικόνα 9.9 Αλλοίωση εικόνας μετά από αύξηση των διαστάσεών της

Η άλλη επιλογή που έχουμε, δηλαδή της “επανάληψης” της εικόνας, μπορεί να προκαλέσει ένα άλλου είδους πρόβλημα. Παρά την επανάληψή της (συνήθως κατά δεκάδες φορές), η τελική εικόνα δε θα πρέπει να δίνει την εντύπωση ότι αποτελείται από πολλές άλλες, θα πρέπει να φαίνεται σαν μία και μόνο. Για το λόγο αυτό, η εικόνα θα πρέπει να παρουσιάζει κάποιας μορφής συμμετρία του πάνω μέρους της με το κάτω, καθώς επίσης του δεξιού με το αριστερό. Κατ’ αυτό τον τρόπο, κατά την επανάληψή της, δε θα εμφανιστούν “ραφές”, ασυμμετρίες και ασυνέχειες (εικόνα 9.10).

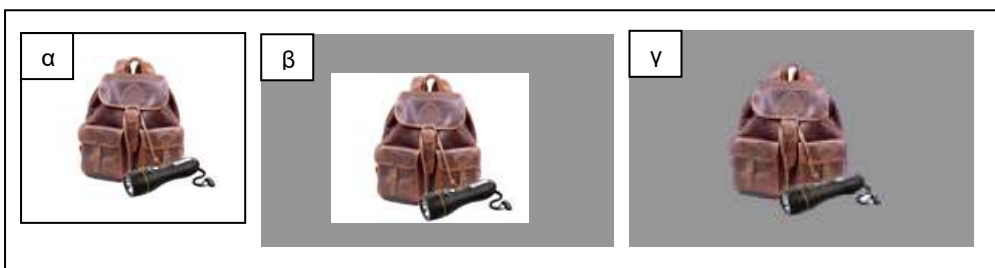
Όταν το μοτίβο της εικόνας είναι απλό, εύκολα μπορούμε να επιτύχουμε το επιθυμητό αποτέλεσμα. Τα πράγματα γίνονται ιδιαίτερα δύσκολα όταν το μοτίβο είναι περίπλοκο και δύσκολα μπορούμε να εντοπίσουμε ένα κομμάτι που να παρουσιάζει συμμετρία. Στους δικτυακούς τόπους που αναφέραμε αλλά και σε πολλούς άλλους που ο ενδιαφερόμενος μπορεί να εντοπίσει, οι εικόνες που προορίζονται για υφές μεγάλων επιφανειών είναι ήδη κατάλληλα επεξεργασμένες.



Εικόνα 9.10 Εμφάνιση “ραφών” και ασυμμετρίας κατά την επανάληψη της εικόνας (αριστερά πάνω λανθασμένη επανάληψη εικόνων, δεξιά πάνω σωστή, αριστερά κάτω λανθασμένα επεξεργασμένη αρχική εικόνα, δεξιά κάτω σωστά επεξεργασμένη)

9.2 Διαφανές τμήμα εικόνων

Όταν κατασκευάζουμε την εικόνα μίας υψής, ανεξάρτητα με το αντικείμενο που παριστάνει, ο καμβάς πάνω στον οποίο είναι “ζωγραφισμένη” η εικόνα είναι τετράγωνος ή ορθογώνιο παραλληλόγραμμο. Όμως ο καμβάς αποτελεί και αυτός τμήμα της εικόνας, πράγμα που σημαίνει ότι όταν μεταφέρουμε την εικόνα του αντικειμένου πάνω σε μία άλλη εικόνα, τοποθετείται και φαίνεται ο καμβάς της πρώτης. Υπάρχει όμως η δυνατότητα να επεξεργαστούμε την αρχική μας εικόνα μ' ένα τέτοιο τρόπο ώστε το Reality Factory να γνωρίζει ποιο κομμάτι της εικόνας είναι το αντικείμενο και ποιο ο καμβάς και να μπορεί να τον αγνοεί, θεωρώντας τον κατά κάποιον τρόπο διαφανή (εικόνα 9.11).



Εικόνα 9.11 Τοποθέτηση εικόνας πάνω σε εικόνα
α) Αρχική εικόνα, β) Τοποθέτηση εικόνας πάνω σε άλλη εικόνα με εμφανή τον καμβά της πρώτης, γ) Επεξεργασμένη αρχική εικόνα ώστε ο καμβάς της να είναι διαφανής

Τέτοιου είδους εικόνες θα χρειαστούμε σε αρκετές περιπτώσεις, όπως για παράδειγμα στις σκίες που δημιουργούν τα διάφορα αντικείμενα. Ο καμβάς της σκιάς θα πρέπει να είναι διαφανής, γιατί το φόντο που προβάλλεται η σκιά συνεχώς αλλάζει. Είναι λοιπόν αρκετά σημαντικό να δούμε τη μεθοδολογία κατασκευής διαφανών τμημάτων εικόνων.

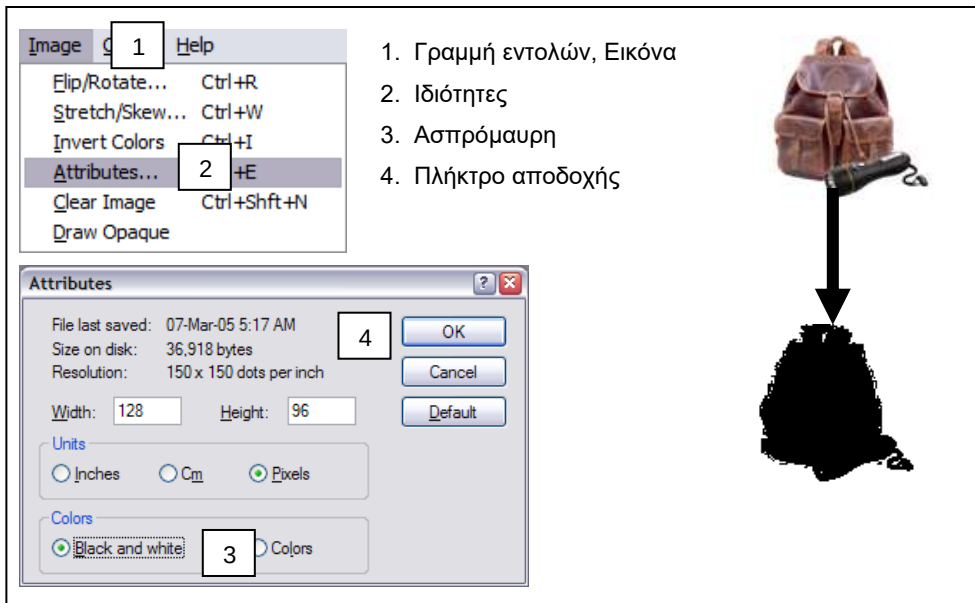
Η βασική ιδέα είναι ότι, πέρα από την αρχική εικόνα, κατασκευάζουμε και μία δεύτερη με τις ίδιες διαστάσεις, όπου ορίζουμε ποια εικονοστοιχεία της πρώτης είναι διαφανή και ποια αδιαφανή. Η δεύτερη εικόνα ονομάζεται Μάσκα Άλφα (*Alpha Mask*). Κάθε εικονοστοιχείο της *Alpha Mask* επηρεάζει το αντίστοιχό του στην κύρια εικόνα. Στο *Reality Factory* διαφανές θεωρείται ένα εικονοστοιχείο που το αντίστοιχό του στην *Alpha Mask* έχει την τιμή 0 για τα τρία βασικά του χρώματα (Κόκκινο=0, Πράσινο=0, Μπλε=0 ή RGB=0,0,0). Αντίθετα, αδιαφανές θεωρείται το εικονοστοιχείο που έχει την τιμή 255 για τα τρία βασικά του χρώματα (RGB=255,255,255). Η τιμή RGB=0,0,0 αντιστοιχεί στο μαύρο χρώμα και η τιμή RGB=255,255,255 στο άσπρο.

Είτε χρησιμοποιήσουμε τη “Ζωγραφική” είτε το *Deep Paint*, τα βήματα που ακολουθούμε για την κατασκευή *Alpha Mask* είναι τα ίδια: α) μετατρέπουμε την εικόνα σε ασπρόμαυρη, β) αν είναι απαραίτητο αντικαθιστούμε κάθε τόνο του γκρι με μαύρο χρώμα, γ) αν είναι απαραίτητο αντιστρέφουμε τα χρώματα, δ) διορθώνουμε τυχόν ατέλειες και ε) αποθηκεύουμε τη νέα εικόνα με άλλο όνομα.

Στη συνέχεια παρουσιάζουμε τη διαδικασία και με τα δύο προγράμματα.

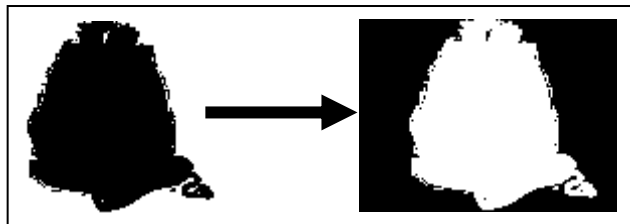
9.2.1 Κατασκευάζοντας *Alpha Mask* με τη “Ζωγραφική”

Εκκινούμε τη “Ζωγραφική” των *Windows* και ανοίγουμε την εικόνα που μας ενδιαφέρει. Από την επιλογή “Εικόνα” της γραμμής εντολών κάνουμε κλικ στο “Ιδιότητες”. Στην καρτέλα που ανοίγει, επιλέγουμε “Ασπρόμαυρη” και πατάμε το πλήκτρο OK (εικόνα 9.12).



Εικόνα 9.12 Μετατροπή εικόνας σε ασπρόμαυρη στη “Ζωγραφική”

Παρατηρούμε ότι τα χρώματα της εικόνας είναι τα αντίθετα απ' αυτά που θέλουμε. Ο καμβάς είναι άσπρος (δηλαδή αδιαφανής) και η εικόνα μαύρη (δηλαδή διαφανής). Για να αντιστρέψουμε τα χρώματα, κάνουμε ξανά κλικ στην επιλογή “Εικόνα” της γραμμής εργαλείων, αλλά αυτή τη φορά επιλέγουμε “Αρνητικά Χρώματα” (εικόνα 9.13).



Εικόνα 9.13 Αντιστροφή χρωμάτων

Μερικές φορές γύρω από το περίγραμμα της εικόνας μένουν μερικά εικονοστοιχεία “σκουπιδάκια”. Αν και δεν επηρεάζουν σε σημαντικό βαθμό το τελικό αποτέλεσμα, εντούτοις μπορούμε με το εργαλείο  και επιλέγοντας μαύρο ή άσπρο χρώμα να τα απαλείψουμε.

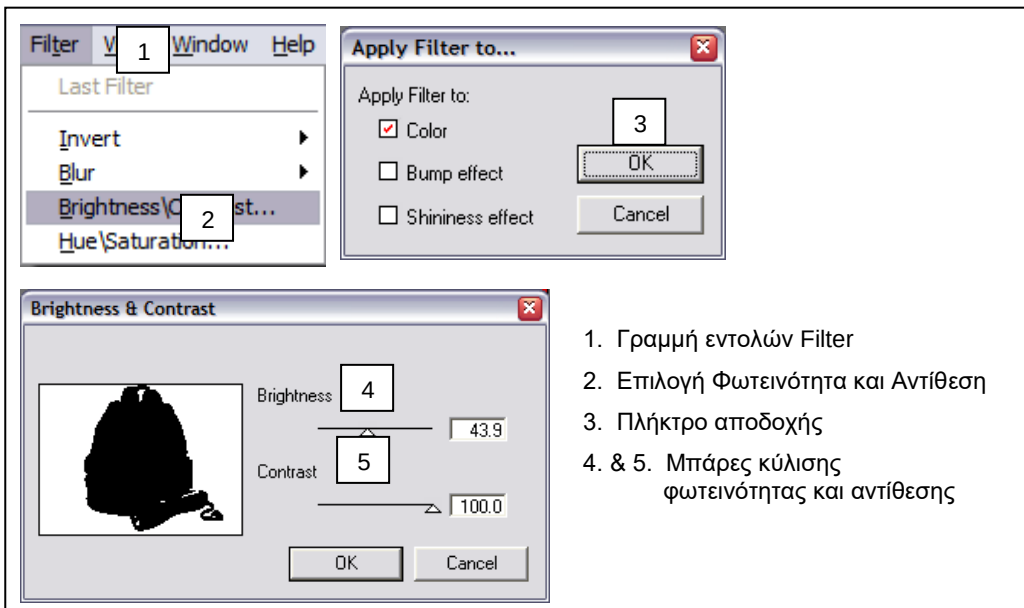
Τέλος, αποθηκεύουμε την εικόνα με διαφορετικό όνομα, ποτέ όμως με ελληνικούς χαρακτήρες και κενά. Μάλιστα είναι αρκετά συνηθισμένο (όχι όμως υποχρεωτικό) το όνομα του αρχείου να ξεκινά με το γράμμα “a” ακολουθούμενο από κάτω παύλα “_”,

όπου το “a” σημαίνει alpha mask. Αν, για παράδειγμα, η αρχική μας εικόνα ήταν η item.bmp, η αντίστοιχή της alpha mask μπορεί να ονομαστεί a_item.bmp.

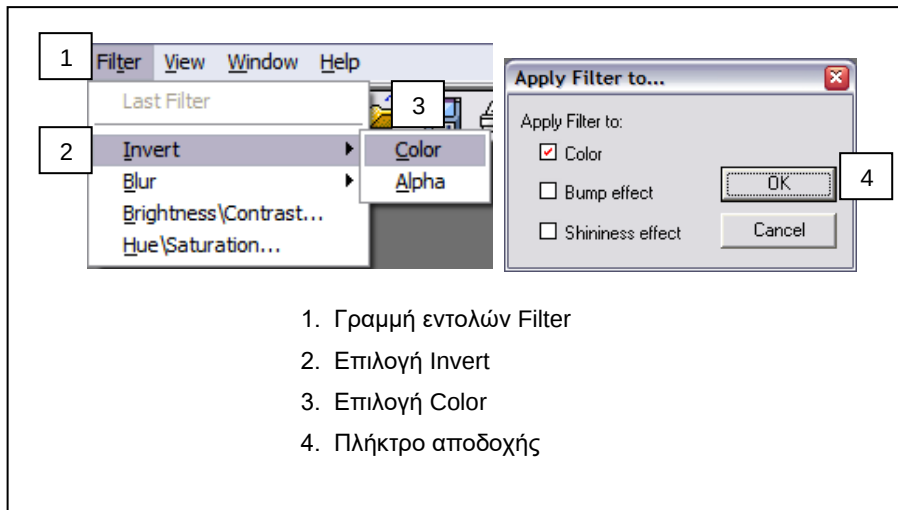
9.2.2 Κατασκευάζοντας Alpha Mask με το Deep Paint

Παρόμοια είναι τα βήματα που ακολουθούμε με το Deep Paint. Ανοίγουμε την εικόνα που θέλουμε να επεξεργαστούμε είτε από το εικονίδιο  είτε από τη γραμμή εντολών. Από τη γραμμή εντολών “Filter” και την επιλογή “Brightness/Contrast” (Φωτεινότητα/Αντίθεση) αυξάνουμε την αντίθεση στο 100% και μειώνουμε βηματικά τη φωτεινότητα, μέχρις ότου να σχηματιστεί σωστά το περίγραμμα της εικόνας και δεν υπάρχουν “σκουπιδάκια” μέσα σ' αυτή. Πρέπει να προσέξουμε να μη μειώσουμε σε μεγάλο βαθμό τη φωτεινότητα, γιατί η εικόνα θα γίνει όλη μαύρη (εικόνα 9.14).

Στη συνέχεια, και πάλι από τη γραμμή εντολών “Filter” αλλά αυτή τη φορά από την επιλογή “Invert”, “Color”, αντιστρέφουμε τα χρώματα, γιατί όπως και με τη “Ζωγραφική” ήταν λάθος (εικόνα 9.15). Τέλος, αποθηκεύουμε την εικόνα μας.



Εικόνα 9.14 Μετατροπή εικόνας σε ασπρόμαυρη στο Deep Paint



Εικόνα 9.15 Αντιστροφή χρωμάτων στο Deep Paint

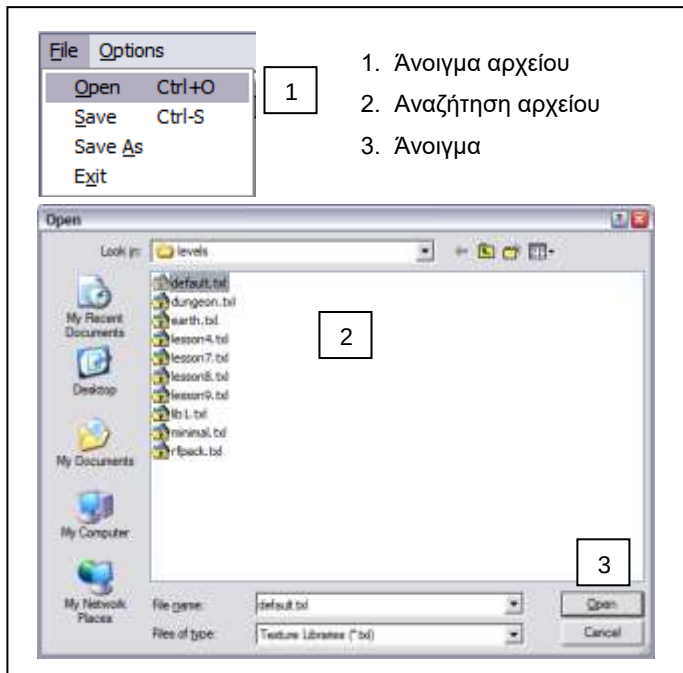
9.3 Κατασκευή πακέτων υφών

Έχοντας επεξεργαστεί τις εικόνες μας και έχοντας συγκεντρώσει τις υφές που μπορεί να χρειαστούμε, το επόμενο βήμα είναι να κατασκευάσουμε ένα πακέτο υφών που θα χρησιμοποιήσει το RF. Αυτό γίνεται με το πρόγραμμα Texture Packer (rfPack.exe), που βρίσκεται όπως και τα υπόλοιπα εργαλεία του RF στο φάκελο C:\RealityFactory\Tools.

Όλα τα πακέτα υφών αποθηκεύονται υποχρεωτικά στο φάκελο
C:\RealityFactory\Media\Levels.

Αποθήκευση σε άλλο φάκελο είναι εφικτή μόνο αν αλλάξουμε την αντίστοιχη δήλωση στο αρχείο RealityFactory.ini. Κάτι τέτοιο όμως δε συνιστάται.

Ως πρόγραμμα είναι πολύ απλό στη χρήση του. Μπορούμε να ανοίξουμε ένα υπάρχον πακέτο υφών (εικόνα 9.16) και να δούμε τις εικόνες που περιέχει (εικόνα 9.17). Επιλέγοντας μία υφή και πατώντας το πλήκτρο Delete μπορούμε να τη διαγράψουμε από το πακέτο. Για να αποθηκεύσουμε το πακέτο, επιλέγουμε "File", "Save" αν επεξεργαστήκαμε ένα πακέτο που ήδη υπάρχει ή "Save As" αν αποθηκεύουμε για πρώτη φορά το πακέτο.



Εικόνα 9.16 Άνοιγμα υπάρχοντος πακέτου υφών

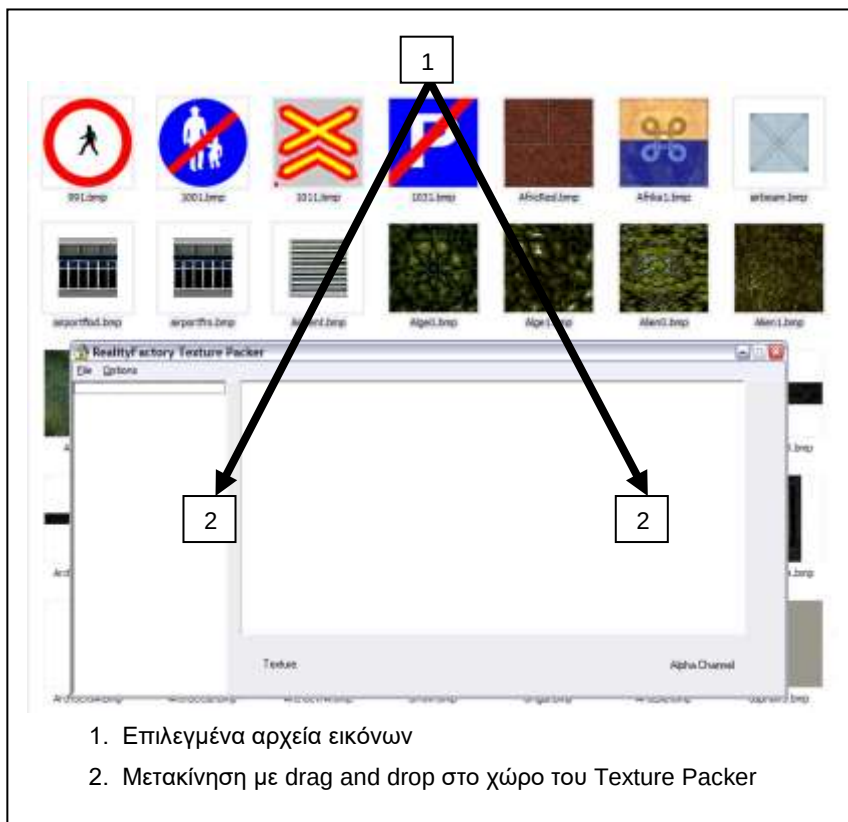


Εικόνα 9.17 Προεπισκόπηση των υφών που περιέχονται στο πακέτο

Μπορούμε να προσθέσουμε εικόνες σ' ένα πακέτο που υπάρχει ήδη ή να δημιουργήσουμε ένα νέο. Και στις δύο περιπτώσεις η διαδικασία που ακολουθούμε είναι η ίδια:

1. Ανοίγουμε παράλληλα με το Texture Packer και την “Εξερεύνηση των Windows”.
2. Αναζητούμε και ανοίγουμε το φάκελο με τις εικόνες που προορίζονται να γίνουν υφές στο πακέτο μας.
3. Τις επιλέγουμε και τις σέρνουμε στο χώρο του Texture Packer (είτε στο δεξιό είτε στον αριστερό χώρο) (εικόνα 9.18).

4. Στην περίπτωση κατά την οποία καθώς επιλέγουμε τις εικόνες από την “Εξερεύνηση των Windows” το παράθυρο του Texture Packer ελαχιστοποιηθεί, τις σέρνουμε στο ελαχιστοποιημένο παράθυρο (στο κάτω μέρος της οθόνης μας), το οποίο και μεγιστοποιείται (εικόνα 9.19).
5. Αποθηκεύουμε τις αλλαγές ή δίνουμε νέο όνομα στο αρχείο του πακέτου υφών.



Εικόνα 9.18 Μετακίνηση αρχείων στο Texture Packer



Εικόνα 9.19 Μετακίνηση αρχείων στο ελαχιστοποιημένο παράθυρο του Texture Packer

Η διαδικασία δεν είναι δεσμευτική. Αυτό σημαίνει ότι, εάν κατά τη διάρκεια κατασκευής της εφαρμογής μας χρειαστεί να προσθέσουμε νέες υφές στο πακέτο που χρησιμοποιούμε, μπορούμε να το κάνουμε. Σε περίπτωση που διαγράψουμε μία υφή από το πακέτο, που όμως την έχουμε χρησιμοποιήσει στην εφαρμογή μας, τότε το αντικείμενο ή τα αντικείμενα που τη χρησιμοποιούσαν θα εμφανιστούν χωρίς υφή (θα έχουν όμως ένα έντονο χρώμα δηλωτικό της απουσίας υφής) και θα πρέπει να τα “ντύσουμε” με κάποια άλλη.

Τα πακέτα υφών δεν πρέπει να περιέχουν έναν εξαιρετικά μεγάλο αριθμό εικόνων.

Στο RF το πακέτο υφών φορτώνεται ολόκληρο στην προσωρινή μνήμη (*RAM*).

Ακόμα και αν έχουμε αρκετή μνήμη, δεν είναι σίγουρο ότι κάποιος άλλος χρήστης της εφαρμογής μας θα έχει τις ίδιες δυνατότητες.

Κεφάλαιο 10^ο

Μετατροπή τρισδιάστατων μοντέλων για χρήση στο Reality Factory

Η κατασκευή τρισδιάστατων μοντέλων είναι μία σύνθετη αλλά και δημιουργική εργασία, που όμως η περιγραφή της είναι έξω από τους στόχους αυτού του βιβλίου. Αυτό που θα μας απασχολήσει είναι η επεξεργασία που πρέπει να υποστούν υπάρχοντα μοντέλα έτσι ώστε να μπορούν να χρησιμοποιηθούν στο RF. Η επεξεργασία τους αποτελεί ίσως την πιο σημαντική εργασία κατά τη δημιουργία ενός εικονικού κόσμου. Για να γίνει κατανοητή η σπουδαιότητά της, μπορούμε να φανταστούμε το RF απλώς σαν ένα χώρο συρραφής ετερογενών στοιχείων (ήχοι, εικόνες, μοντέλα, κτλ.), όπου τα μοντέλα των αντικειμένων παίζουν πρωταγωνιστικό ρόλο.

Ωστόσο, όσο σημαντική είναι η σωστή επεξεργασία των μοντέλων, άλλο τόσο περίπλοκη μπορεί να γίνει. Στο 4^ο και στο 6^ο κεφάλαιο περιγράψαμε τα προβλήματα που δημιουργεί ο μεγάλος αριθμός των πολυγώνων και τις τεχνικές που χρησιμοποιούνται για τον περιορισμό τους. Πρέπει να προσθέσουμε ότι αυτό δεν είναι το μόνο πρόβλημα που πρέπει να αντιμετωπιστεί. Υπάρχουν αρκετά προγράμματα επεξεργασίας τρισδιάστατων γραφικών που το καθένα παράγει αρχεία που είναι ασύμβατα με τα άλλα και, επιπρόσθετα, το RF απαιτεί ένα δικό του τύπο αρχείων. Συνεπώς, χρειαζόμαστε ένα “πολυεργαλείο” το οποίο να μπορεί να μειώνει τον αριθμό των πολυγώνων και γενικά να επεξεργάζεται τα μοντέλα, να αναγνωρίζει διαφορετικούς τύπους αρχείων και να μπορεί να δημιουργεί τύπους αρχείων συμβατούς με το RF. Για το λόγο αυτό στο 8^ο κεφάλαιο προτείνουμε το Milkshape3D, το οποίο μπορεί και κάνει όλα τα παραπάνω.

Η διαδικασία όμως δεν τελειώνει σ' αυτό το σημείο. Στο RF ένα τρισδιάστατο μοντέλο συγκροτείται από τρία διαφορετικά στοιχεία:

1. Το σώμα του μοντέλου, δηλαδή τα πολύγωνα που το απαρτίζουν.
2. Την υφή του μοντέλου, δηλαδή την εικόνα που το “ντύνει”.
3. Το/τα animations του μοντέλου, δηλαδή τις κινήσεις που του έχουμε δώσει (αν βέβαια είναι στατικό μοντέλο δεν είναι απαραίτητο να υπάρχουν κινήσεις).

Τα στοιχεία αυτά, εκτός από την επεξεργασία των υφών που γίνεται με τις μεθόδους που περιγράψαμε στο προηγούμενο κεφάλαιο, μπορούμε να τα επεξεργαστούμε στο Milkshape3D και να εξαγάγουμε τα αντίστοιχα αρχεία. Στη συνέχεια, με άλλο εργαλείο που περιλαμβάνεται στο RF, το *Actor Studio*, τα συνδυάζουμε και δημιουργούμε ένα τρισδιάστατο μοντέλο στη μορφή που απαιτεί το RF. Επιγραμματικά λοιπόν, η όλη διαδικασία είναι η εξής:

1. Αναζήτηση ή κατασκευή τρισδιάστατου αντικειμένου.
2. Εισαγωγή του στο Milkshape3D.
3. Επεξεργασία του μοντέλου, μείωση πολυγώνων, αυξομείωση μεγέθους κτλ, στο Milkshape3D.

4. Επεξεργασία των συνοδευτικών υφών και κυρίως μετατροπή τους σε .bmp με τις κατάλληλες διαστάσεις χρησιμοποιώντας το *Texture Image Converter* ή άλλο πρόγραμμα επεξεργασίας εικόνων.
5. Εφαρμογή των υφών με το Milkshape3D.
6. Κατασκευή των animations του μοντέλου με το Milkshape3D.
7. Εξαγωγή του σώματος του μοντέλου μαζί με τις υφές του από το Milkshape3D.
8. Εξαγωγή των animations από το Milkshape3D.
9. Συγκρότηση τρισδιάστατου μοντέλου και των animations που περιλαμβάνει με το *Actor Studio*.

10.1 Εισάγοντας ένα μοντέλο στο Milkshape3D, αρχική επεξεργασία

Αν αναζητήσουμε τρισδιάστατα μοντέλα στο Διαδίκτυο, αυτά που θα βρούμε εμπίπτουν σε τρεις κατηγορίες: α) μοντέλα χωρίς συνοδευτικές εικόνες για υφές, β) μοντέλα με μόνο μία εικόνα για υφή και γ) μοντέλα με περισσότερες από μία υφές.

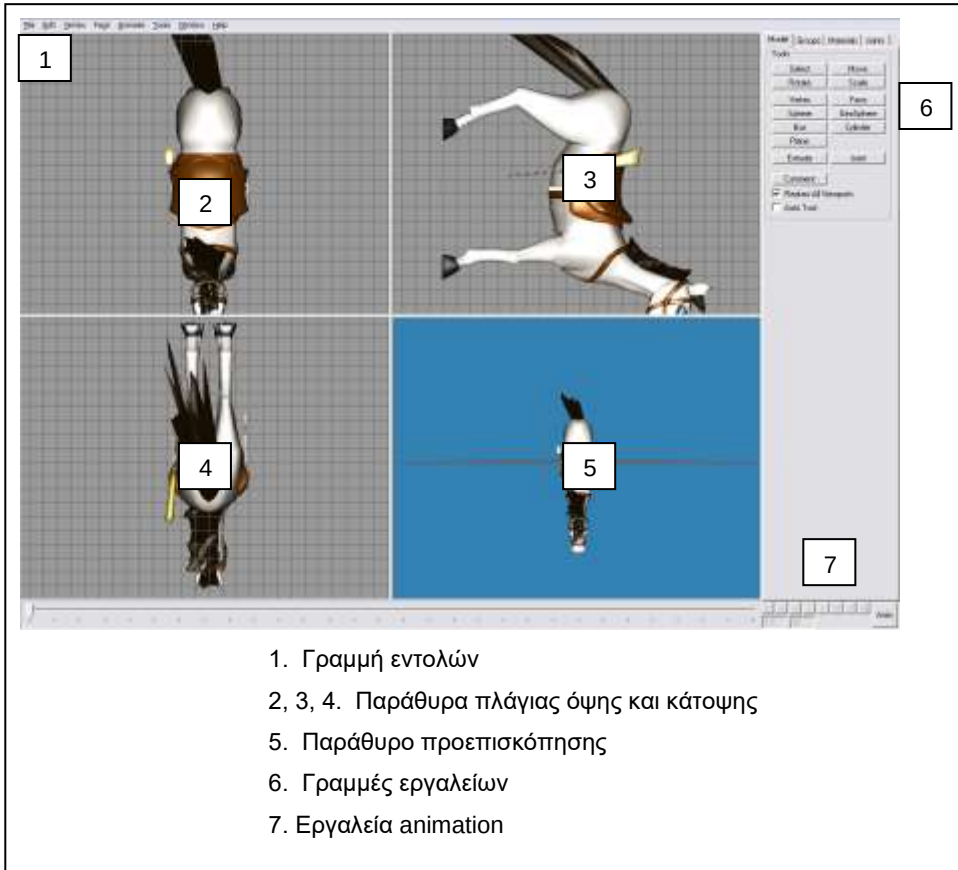
Στην πρώτη περίπτωση, το αντικείμενο αποτελείται συνήθως από περισσότερα του ενός κομμάτια και είτε είναι “γυμνό” είτε έχουν χρησιμοποιηθεί ειδικά χρώματα/υλικά (*materials*) για να αναπαρασταθούν τα υλικά που απαρτίζουν τα διάφορα τμήματά του. Αυτοί οι χρωματισμοί παρέχονται από το πρόγραμμα με το οποίο κατασκευάστηκε το αντικείμενο και έτσι δεν είναι απαραίτητη η χρήση εικόνων για τις υφές του.

Στη δεύτερη περίπτωση, το αντικείμενο απαρτίζεται συνήθως από ένα και μόνο κομμάτι και χρησιμοποιείται μία και μόνο εικόνα για υφή, που το καλύπτει με μία τεχνική που ονομάζεται *UV Mapping*. Πριν επεξεργαστούμε το μοντέλο με το Milkshape3D, θα χρειαστεί να επεξεργαστούμε πρώτα την εικόνα αυτή (μετατροπή σε .bmp και αλλαγή διαστάσεων) και να την αποθηκεύσουμε στο ίδιο σημείο που είναι αποθηκευμένο το μοντέλο. Στην τρίτη περίπτωση ισχύουν ό,τι και στην προηγούμενη, με τη διαφορά ότι θα πρέπει να επεξεργαστούμε όλες τις συνοδευτικές εικόνες.

Κατά την επεξεργασία ενός μοντέλου, οι επεξεργασμένες εικόνες των υφών του πρέπει να βρίσκονται στον ίδιο φάκελο με το μοντέλο.

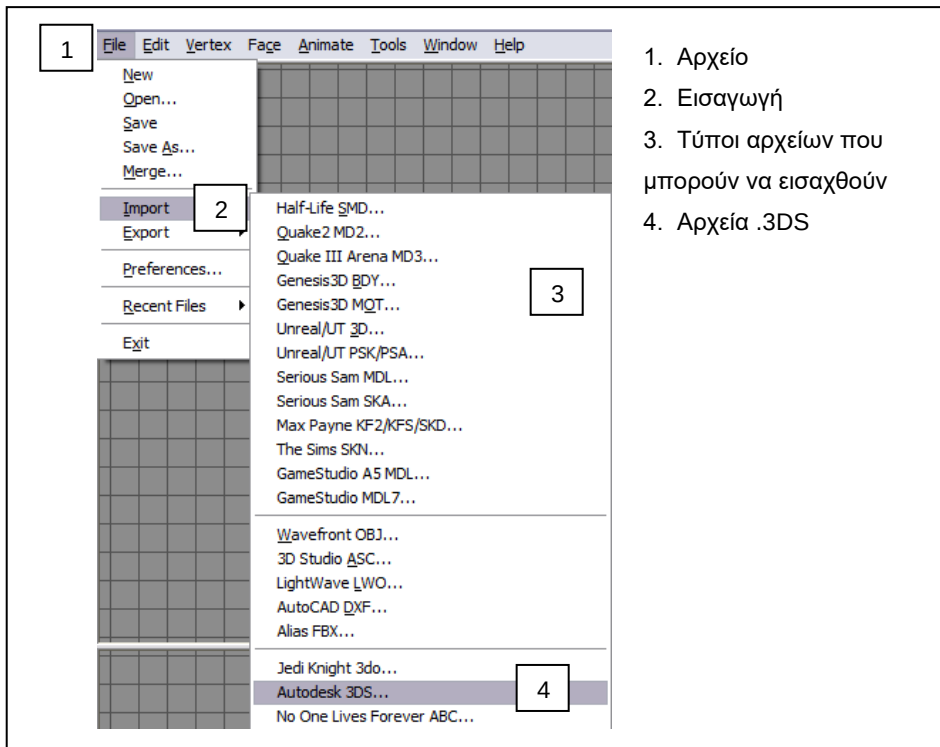
Σε αντίθετη περίπτωση, κατά την εξαγωγή του μοντέλου στη μορφή που απαιτείται από το RF, δε θα “περαστούν” οι υφές.

Εκκινούμε το Milkshape3D από το εικονίδιο που δημιουργεί κατά την εγκατάστασή του στην επιφάνεια εργασίας ή από την “Έναρξη”, “Όλα τα προγράμματα” (εικόνα 10.1).



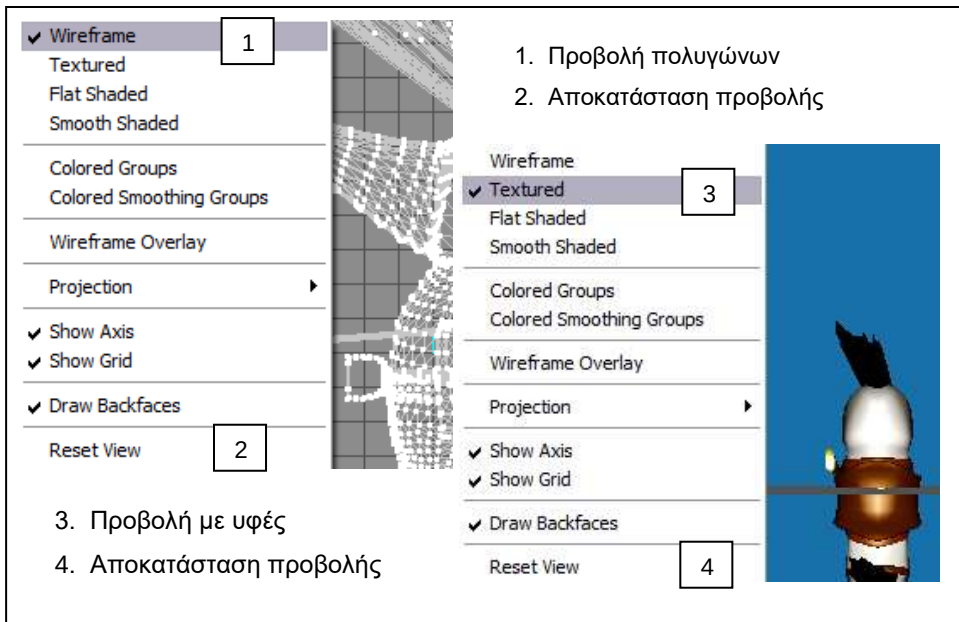
Εικόνα 10.1 Το περιβάλλον εργασίας του Milkshape3D

Για να εισαγάγουμε ένα μοντέλο στο Milkshape3D πρέπει πρώτα να γνωρίζουμε τον τύπο του αρχείου του, δηλαδή από ποιο πρόγραμμα κατασκευάστηκε. Η πιο συνηθισμένες μορφές είναι αρχεία τύπου .3DS, που υποδηλώνουν ότι έχουν κατασκευαστεί με το 3D Studio Max, και αρχεία τύπου .LWO, που υποδηλώνουν ότι έχουν κατασκευαστεί με το Lightwave. Στο παράδειγμά μας θα χρησιμοποιήσουμε ένα μοντέλο .3DS. Το ανοίγουμε χρησιμοποιώντας την ανάλογη επιλογή από τη γραμμή εντολών “File”, “Import” (εικόνα 10.2).



Εικόνα 10.2 Εισαγωγή μοντέλου στο Milkshape3D

Όταν εισάγουμε ένα μοντέλο στο Milkshape3D, το πρόγραμμα προσαρμόζει τα τέσσερα παράθυρα ώστε να φαίνεται όλο το μοντέλο. Έτσι όμως δεν έχουμε πλήρη εντύπωση για το μέγεθός του. Για το λόγο αυτό, κάνουμε δεξί κλικ σε καθεμία από τις τρεις πλάγιες όψεις και επιλέγουμε διαδοχικά “Reset View” και “Wireframe”. Η ενέργεια αυτή μας επιτρέπει να βλέπουμε το πραγματικό του μέγεθος καθώς επίσης και τα πολύγωνα που το αποτελούν. Στο παράθυρο της προεπισκόπησης κάνουμε και πάλι δεξί κλικ, επιλέγουμε “Reset View”, αλλά αυτή τη φορά επιλέγουμε το “Textured”, ώστε να βλέπουμε και τις υφές που καλύπτουν το μοντέλο μας (εικόνα 10.3).



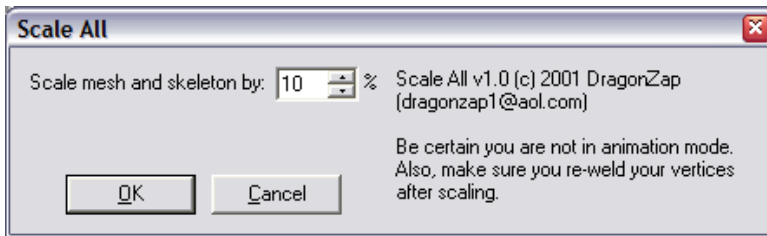
Εικόνα 10.3 Προσαρμογή προβολών

Κάνοντας αποκατάσταση της προβολής (Reset View), δύο πράγματα μπορεί να συμβούν: α) το μοντέλο να είναι τόσο μεγάλο, που δε θα χωράει ολόκληρο στο παράθυρο και θα βλέπουμε μόνο ένα ελάχιστο τμήμα του, και β) το μοντέλο να είναι τόσο μικρό, που δε θα φαίνεται καθόλου ή θα φαίνεται πάρα πολύ μικρό. Σε κάθε περίπτωση, θα πρέπει να γίνει μεταβολή του μεγέθους του, έτσι ώστε το τελικό αποτέλεσμα να είναι ικανοποιητικό για το RF. Πρέπει να σημειώσουμε ότι μπορεί να προκύψουν σημαντικές διαφοροποιήσεις μεγέθους από μοντέλο σε μοντέλο, λόγω των διαφορετικών ρυθμίσεων που μπορεί να έχει κάνει ο αρχικός κατασκευαστής και λόγω των διαφορετικών μονάδων μέτρησης που χρησιμοποιούν τα προγράμματα κατασκευής τρισδιάστατων μοντέλων.

Να σημειώσουμε επίσης ότι, αν το ποντίκι μας διαθέτει μεσαίο ροδάκι, μπορούμε κινώντας το να κάνουμε αύξηση ή μείωση του ζουμ (zoom in/out) στις προβολές. Πατώντας το αριστερό πλήκτρο και κινώντας το ποντίκι, μπορούμε να περιστρέψουμε το μοντέλο στην προβολή προεπισκόπησης. Επίσης, με πατημένο το πλήκτρο Ctrl (Control) και κινώντας το ποντίκι, μετακινούμε το μοντέλο σ' όλες τις προβολές.

Η αυξομείωση του μεγέθους γίνεται από τη γραμμή εντολών "Tools", "Scale All". Στο παράθυρο διαλόγου που ανοίγει, αν δώσουμε τιμές κάτω από το 100%, μειώνουμε το μέγεθος και, αν δώσουμε τιμές πάνω από το 100%, το αυξάνουμε (εικόνα 10.4). Συ-

νήθως χρειάζεται σημαντική μείωση του μεγέθους (της τάξης του 10% του αρχικού) ή σημαντική αύξηση (της τάξης του 2000% του αρχικού). Έπειτα από κάθε αυξομείωση μεγέθους πρέπει να κάνουμε αποκατάσταση προβολής (Reset View) στα παράθυρα προβολών για να δούμε το αποτέλεσμα.

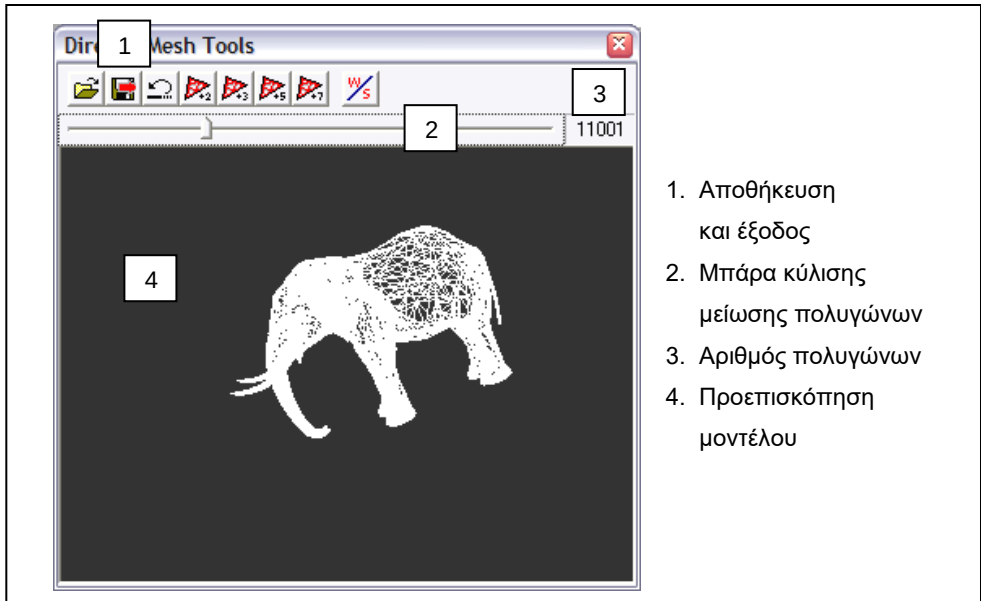


Εικόνα 10.4 Εργαλείο αυξομείωσης μεγέθους

Το τελικό μέγεθος του μοντέλου μετά την αύξηση ή τη μείωση του μεγέθους του και αποκατάστασης της προβολής, πρέπει να καταλαμβάνει από τα τρία τέταρτα ως ολόκληρο το πλέγμα που εμφανίζεται στο παράθυρο προεπισκόπησης. Σε κάποιες περιπτώσεις μπορεί να χρειαστούν διαδοχικές αυξήσεις ή μειώσεις μεγέθους.

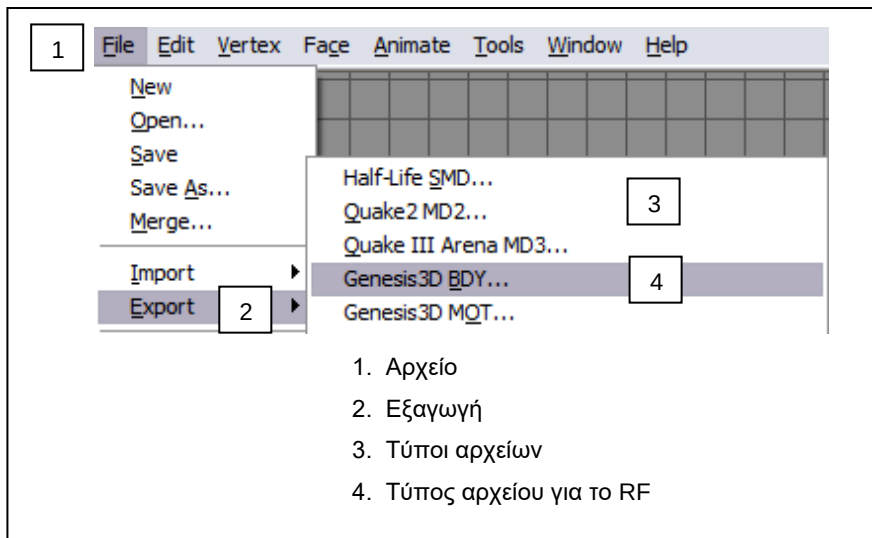
Το επόμενο βήμα μας είναι να μειώσουμε τον αριθμό των πολυγώνων που απαρτίζουν το μοντέλο. Αυτό γίνεται από τη γραμμή εντολών “Tools”, “DirectX Mesh Tools” (εικόνα 10.5). Σέρνουμε την μπάρα κύλισης  και παρατηρούμε το μοντέλο στην προεπισκόπησης του. Προσέχουμε να μη μειώσουμε τα πολύγωνα σε τέτοιο βαθμό που θα έχει ως αποτέλεσμα το μοντέλο να χάσει βασικές του λεπτομέρειες. Αποθηκεύουμε  και επιστρέφουμε στο κυρίως πρόγραμμα.

Χρειάζεται μεγάλη προσοχή στον τελικό αριθμό των πολυγώνων ενός μοντέλου, γιατί, όπως έχουμε υπογραμμίσει, μεγάλος αριθμός συνεπάγεται σημαντική επιβάρυνση στην εφαρμογή. Το ιδανικό είναι να έχει λιγότερο από 1.000, χωρίς βέβαια να αποκλείεται να έχει 2.000-3.000. Τα μοντέλα που βρίσκουμε στο Διαδίκτυο μπορεί να έχουν κατά πολύ μεγαλύτερο αριθμό. Από την άλλη, στην προσπάθειά μας να μειώσουμε σημαντικά τον αριθμό των πολυγώνων μπορεί να καταλήξουμε να έχουν χαθεί βασικά κομμάτια από το μοντέλο μας. Το DirectX Mesh Tools είναι ένα πολύ ισχυρό εργαλείο μείωσης του αριθμού των πολυγώνων, με παράλληλη διατήρηση μεγάλου ποσοστού της ποιότητας, αλλά βεβαίως υπάρχουν και όρια στις δυνατότητές του.



Εικόνα 10.5 Εργαλείο μείωσης αριθμού πολυγώνων

Το συγκεκριμένο μοντέλο δε συνοδεύεται από εικόνες υφών, οπότε μπορούμε να προχωρήσουμε στην εξαγωγή του στη μορφή που απαιτείται από το RF. Αυτό γίνεται από τη γραμμή εντολών "File", "Export", "Genesis3D BDY" (εικόνα 10.6). Το εξαγόμενο μοντέλο το αποθηκεύουμε στον ίδιο φάκελο στον οποίο βρίσκεται το αρχικό.

Εί-
να
Ε-κό-
10.6
ξα-

γγωγή μοντέλου

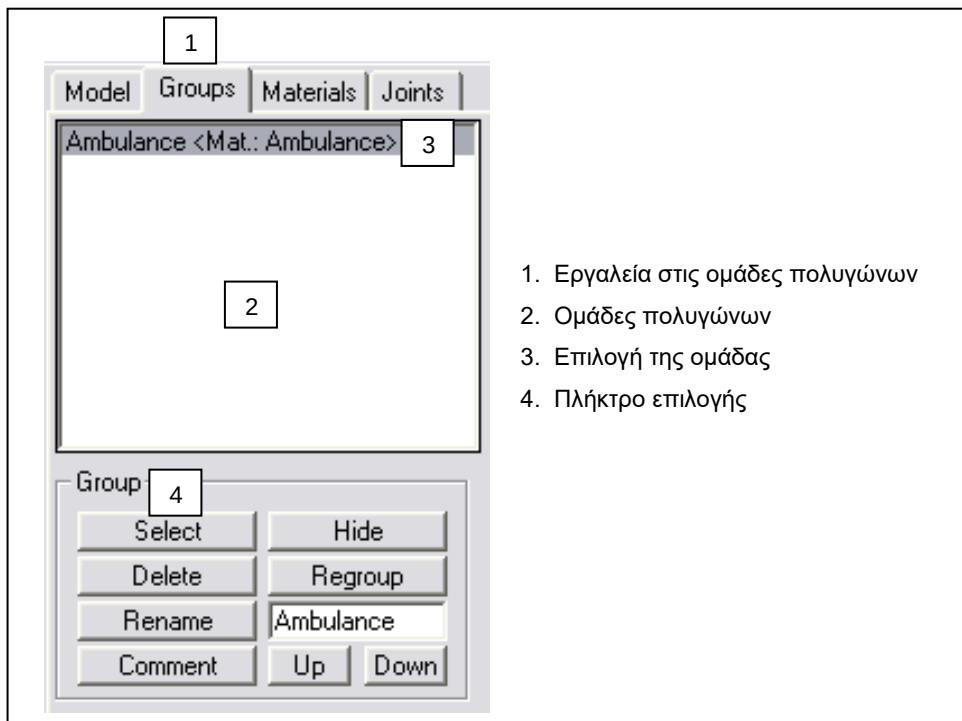
10.2 Επεξεργασία μοντέλου που περιλαμβάνει εικόνες υφών

Στο παράδειγμα της αρχικής επεξεργασίας ενός μοντέλου χρησιμοποιήσαμε ένα το οποίο δεν είχε καθόλου εικόνες για υφές. Στη συνέχεια θα παρουσιάσουμε τα επιπλέον βήματα της όλης διαδικασίας όταν το μοντέλο συνοδεύεται από μία ή περισσότερες εικόνες που θα χρησιμοποιηθούν για τις υφές του. Εννοείται ότι τις έχουμε ήδη μετατρέψει σε Bitmaps (.bmp) με διαστάσεις δυνάμεις του 2 και όλες βρίσκονται στον ίδιο φάκελο με το μοντέλο.

10.2.1 Μοντέλα με μία εικόνα για υφή

Κατά κανόνα, όταν το μοντέλο συνοδεύεται από μία και μόνο εικόνα, τότε έχει μικρό αριθμό πολυγώνων και δε χρειάζεται να τα μειώσουμε. Σε κάθε περίπτωση μπορούμε να ελέγξουμε τον αριθμό τους με το εργαλείο DirectX Tools. Αν είναι περίπου 1.000, δεν προχωρούμε σε περαιτέρω μείωση. Επίσης, τέτοιου είδους μοντέλα αποτελούνται από ένα και μόνο κομμάτι, δεν έχουν δηλαδή επιμέρους τμήματα.

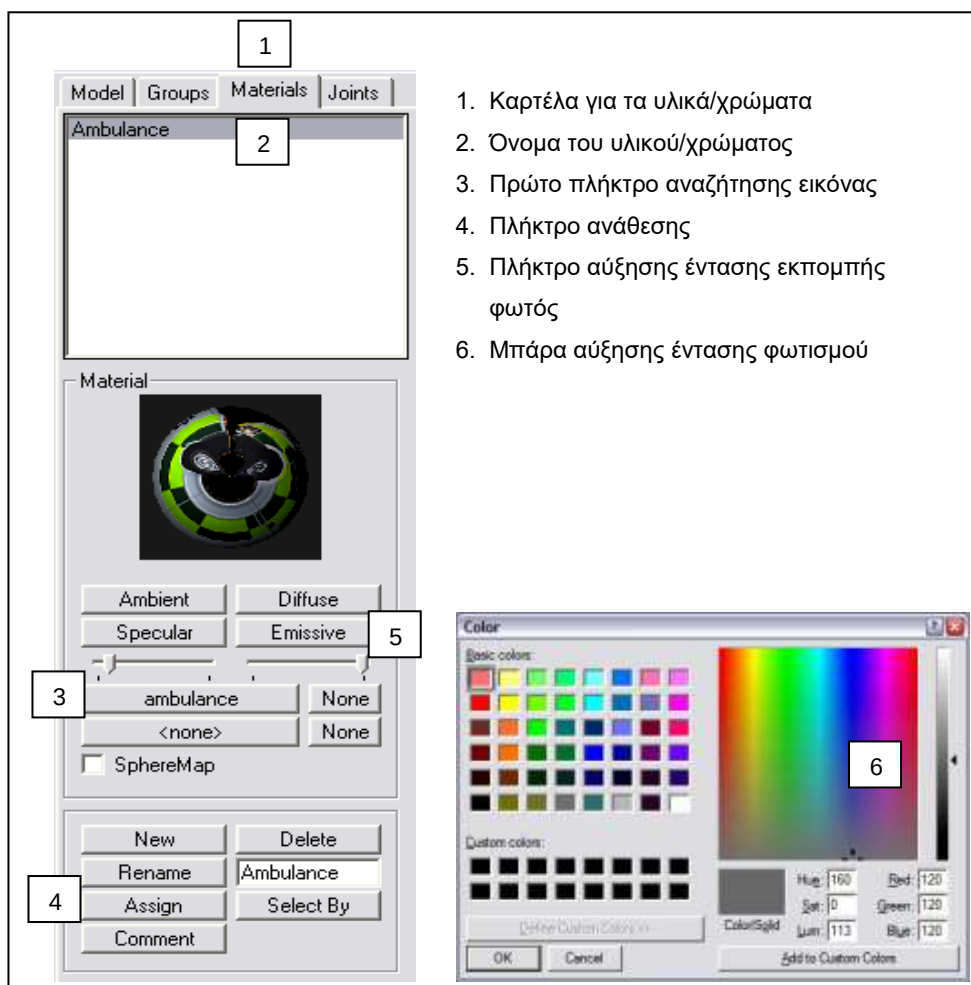
Μπορούμε να δούμε από πόσα τμήματα αποτελείται το μοντέλο από τη γραμμή εργαλείων στην καρτέλα “Groups” . Κάνουμε κλικ στο ένα και μοναδικό γκρουπ πολυγώνων και πατάμε το πλήκτρο επιλογής  (εικόνα 10.7).



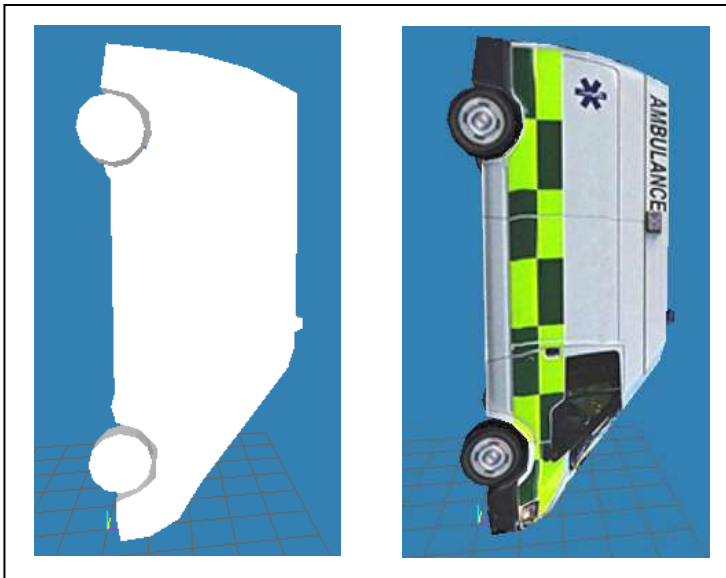
Εικόνα 10.7 Επιλογή ομάδας πολυγώνων

Έχοντας επιλεγμένη τη μοναδική ομάδα πολυγώνων (ουσιαστικά όλο το μοντέλο), κάνουμε κλικ στην καρτέλα “Materials” **Model Groups Materials Joints** της γραμμής εργαλείων.

Πιέζουμε το πρώτο από τα δύο πλήκτρα αναζήτησης υφών **ambulance** **<none>**, βρίσκουμε την εικόνα που θα “ντύσει” το μοντέλο και κάνουμε κλικ στο πλήκτρο ανάθεσης **Assign**. (εικόνα 10.8). Αν το μοντέλο μας είναι αρκετά “σκοτεινό” και δε φαίνεται η υφή, αυξάνουμε την ένταση του φωτός που εκπέμπει κάνοντας κλικ στο πλήκτρο εκπομπής φωτός **Emissive**. Αφού στο μοντέλο έχει περαστεί η υφή (εικόνα 10.9), το εξάγουμε σε μορφή .BDY με τη γνωστή διαδικασία, στον ίδιο φάκελο με το αρχικό μοντέλο.



Εικόνα 10.8 Ανάθεση υφής στο μοντέλο



Εικόνα 10.9 Το μοντέλο πριν και μετά την ανάθεση της υφής

Δεν ξεχνάμε ότι, για να δούμε τις υφές, πρέπει να είναι ενεργοποιημένη η επιλογή “Textured” στο παράθυρο προεπισκόπησης. Είναι πολύ πιθανό ανοίγοντας το αρχείο ενός μοντέλου να δούμε στην προεπισκόπηση ότι έχει ήδη περασμένη μία υφή. Όμως πρόκειται για την αρχική εικόνα που συνόδευε το μοντέλο. Πρέπει οπωσδήποτε να την αντικαταστήσουμε με αυτή που εμείς έχουμε μετατρέψει σε .bmp (δηλαδή χρησιμοποιούμε την ίδια εικόνα με άλλο τύπο αρχείου).

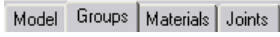
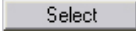
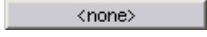
Πάντα επιλέγουμε πρώτα την ομάδα των πολυγώνων πριν της αναθέσουμε μία υφή. Το πλήκτρο “Assign” το πιέζουμε ακόμα και αν στην προβολή προεπισκόπησης έχει φανεί ότι περάστηκε η υφή.

Και βέβαια, δεν ξεχνάμε ότι η εικόνα πρέπει να είναι στον ίδιο φάκελο με το μοντέλο.

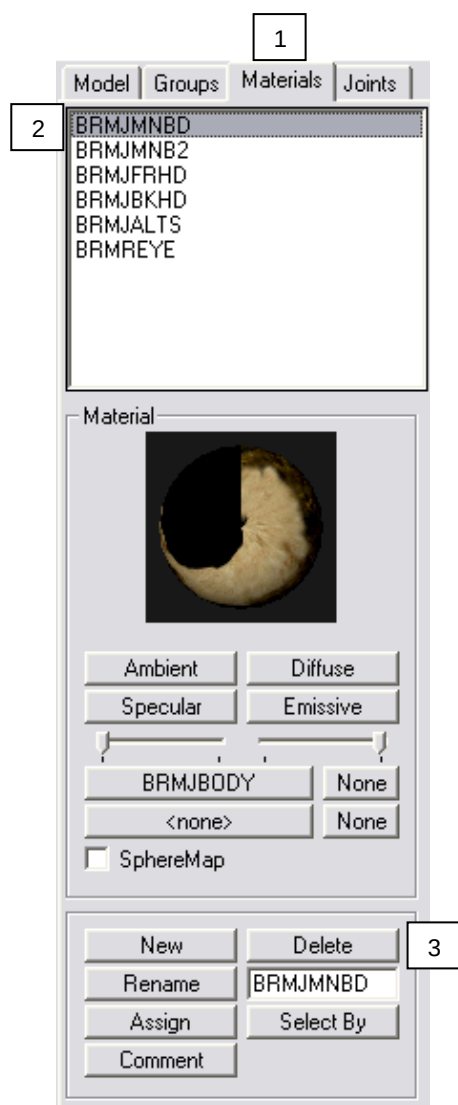
10.2.2 Μοντέλα με δύο ή περισσότερες εικόνες για υφές

Η διαδικασία γίνεται πιο σύνθετη όταν το μοντέλο μας αποτελείται από περισσότερα του ενός κομμάτια και υπάρχουν εικόνες υφών για το καθένα απ' αυτά. Σ' αυτή την περίπτωση θα υπάρχουν τόσα Groups και Materials όσα και τα κομμάτια του.

Η διαδικασία που ακολουθούμε είναι η παρακάτω:

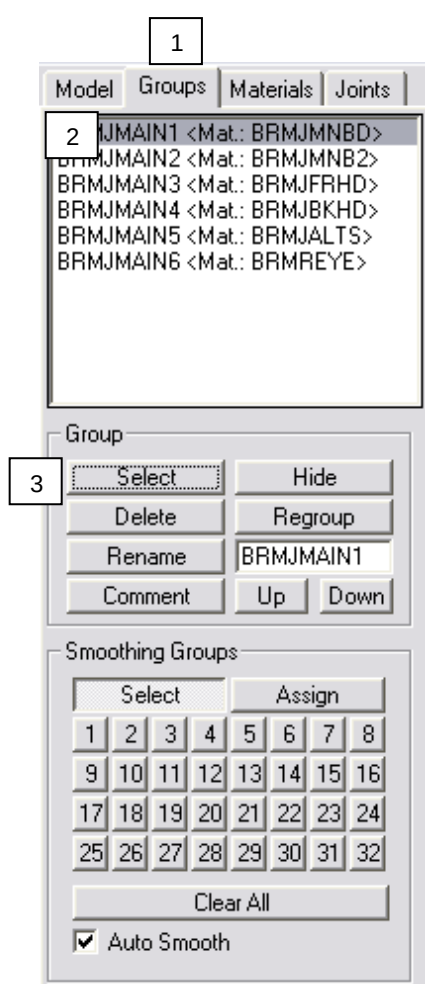
1. Το πρώτο βήμα είναι να διαγράψουμε όλα τα Materials. Από τη γραμμή εργαλείων, στην καρτέλα “Materials” , επιλέγουμε ένα, πατάμε διαγραφή , επιλέγουμε το επόμενο, το διαγράφουμε και αυτό, κ.ο.κ. (εικόνα 10.10).
2. Στη συνέχεια, επιλέγουμε την πρώτη ομάδα πολυγώνων από τη γραμμή εργαλείων, καρτέλα “Groups” , κάνοντας κλικ στην πρώτη και πατώντας στη συνέχεια το πλήκτρο επιλογής  (εικόνα 10.11).
3. Δημιουργούμε ένα Material γι' αυτή την ομάδα από τη γραμμή εργαλείων, καρτέλα “Materials” , νέο υλικό  (εικόνα 10.12).
4. Αναθέτουμε μία εικόνα για υφή. Πιέζουμε το  πρώτο από τα δύο πλήκτρα αναζήτησης υφών  βρίσκουμε την εικόνα που θα “ντύσει” το μοντέλο και κάνουμε κλικ στο πλήκτρο ανάθεσης  (εικόνα 10.13).
5. Επιστρέφουμε στην καρτέλα Groups, απο-επιλέγουμε το Group (πατώντας και πάλι “Select”), επιλέγουμε το επόμενο και επαναλαμβάνουμε τη διαδικασία (εικόνα 10.14).

Αν το μοντέλο μας έχει πολλά κομμάτια και πολλές εικόνες υφών, είναι εύκολο να μπερδευτούμε. Καλό είναι λοιπόν, πριν προχωρήσουμε στη διαγραφή των υλικών, να σημειώσουμε ποιο υλικό αντιστοιχεί σε ποιο κομμάτι και ποια εικόνα χρησιμοποιείται. Παρ' όλα αυτά, αρκετές φορές, επιλέγοντας ένα κομμάτι, είναι εμφανές ποια εικόνα αντιστοιχεί σ' αυτό.

Βήμα 1^ο Διαγραφή Materials

1. Καρτέλα για υλικά/χρώματα
2. Επιλογή υλικού/χρώματος
3. Διαγραφή

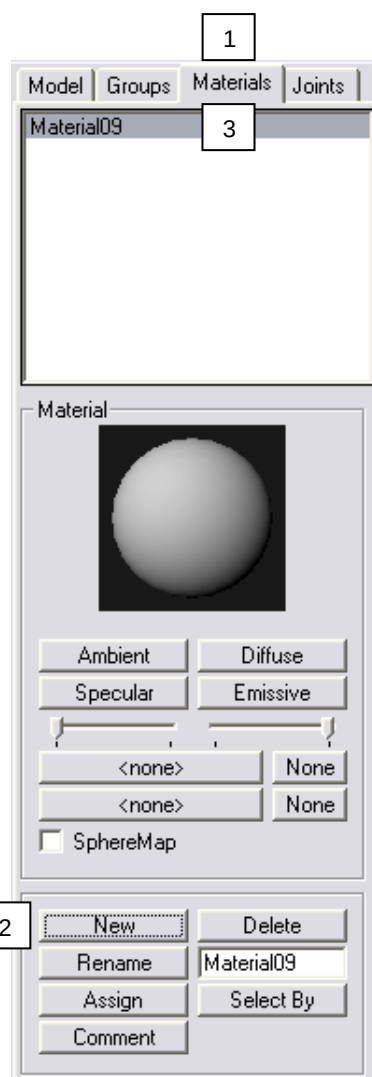
Εικόνα 10.10

Βήμα 2^ο Επιλογή ενός Group

1. Καρτέλα για Groups
2. Επιλογή πρώτης ομάδας
3. Πλήκτρο επιλογής

Εικόνα 10.11

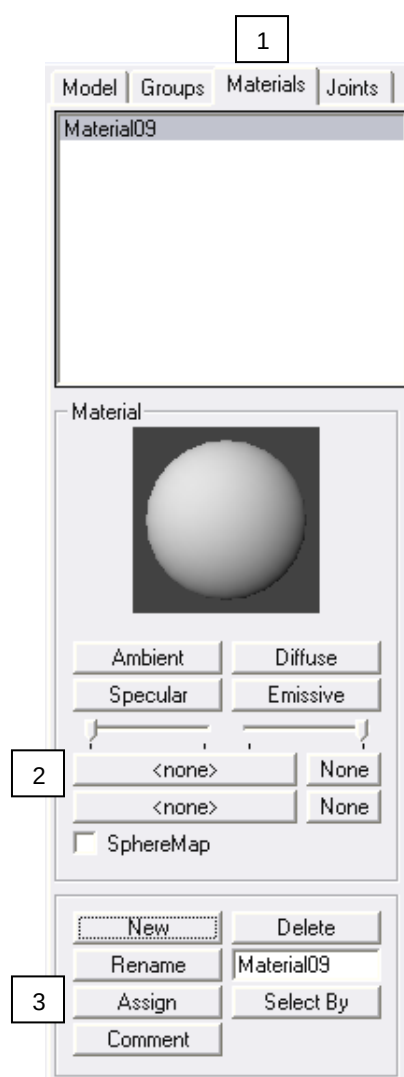
Βήμα 3^ο Δημιουργία νέου Material



1. Καρτέλα για υλικά/χρώματα
2. Δημιουργία νέου υλικού/χρώματος
3. Όνομα νέου υλικού/χρώματος

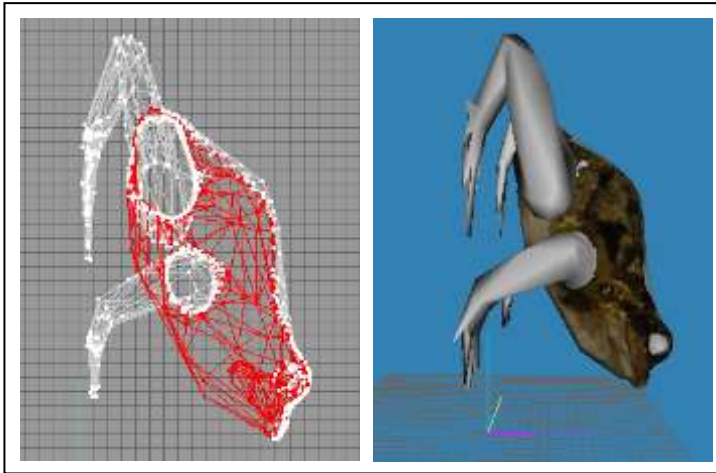
Εικόνα 10.12

Βήμα 4^ο Ανάθεση νέας εικόνας



- 1 Καρτέλα για υλικά/χρώματα
2. Επιλογή εικόνας
3. Πλήκτρο ανάθεσης

Εικόνα 10.13

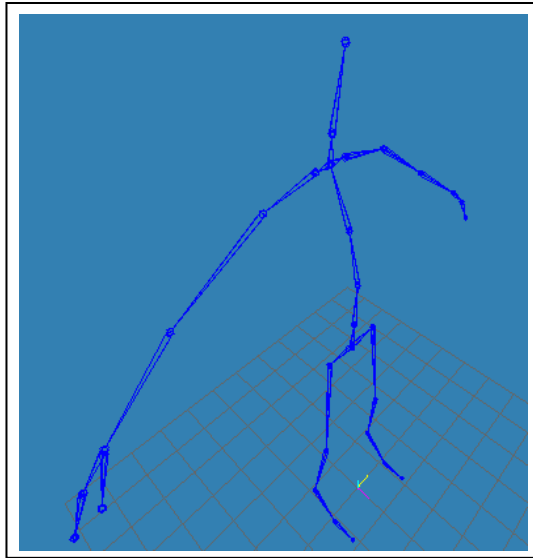


Εικόνα 10.14 Στιγμιότυπο από την επιλογή ομάδας πολυγώνων και ανάθεσης υφής

10.3 Δίνοντας κίνηση στο μοντέλο, μία απλή περίπτωση

Η προσθήκη κίνησης (*animation*) σ' ένα μοντέλο είναι ίσως η πιο χρονοβόρα εργασία κατά τη φάση της επεξεργασίας του. Στο Milkshape3D μπορούμε να προσθέσουμε στο μοντέλο μία μορφή *animation* που ονομάζεται "*bone animation*". Σε γενικές γραμμές είναι ένας τρόπος κίνησης ανάλογος με αυτόν που κινούνται τα έμβια όντα, με κόκαλα (*bones*) και αρθρώσεις (*joints*). Προσθέτουμε δηλαδή μέσα στο μοντέλο μας σημεία που παίζουν το ρόλο αρθρώσεων και τα ενώνουμε με κόκαλα. Τα κόκαλα συνδέονται με ομάδες πολυγώνων (συνήθως με τα πολύγωνα που περιβάλλουν το κάθε κόκαλο). Έτσι, καθώς κινούμε τα κόκαλα σε διαδοχικές θέσεις και τα περιστρέφουμε ανάλογα με τις αρθρώσεις που τα συνδέουν, συμπαρασύρονται και τα πολύγωνα.

Η όλη διαδικασία απαιτεί ακρίβεια, υπομονή και μπορεί να γίνει ιδιαίτερα περίπλοκη, ειδικά αν επιθυμούμε πολύ ακριβείς και λεπτομερείς κινήσεις (αρκεί να σκεφτούμε πόσες αρθρώσεις έχει η ανθρώπινη παλάμη). Τις περισσότερες φορές βέβαια, αρκούμαστε σ' ένα απλουστευμένο μοντέλο αρθρώσεων και κοκάλων, με το οποίο όμως μπορούμε να αναπαραστήσουμε τους περισσότερους συνδυασμούς κινήσεων (εικόνα 10.15).

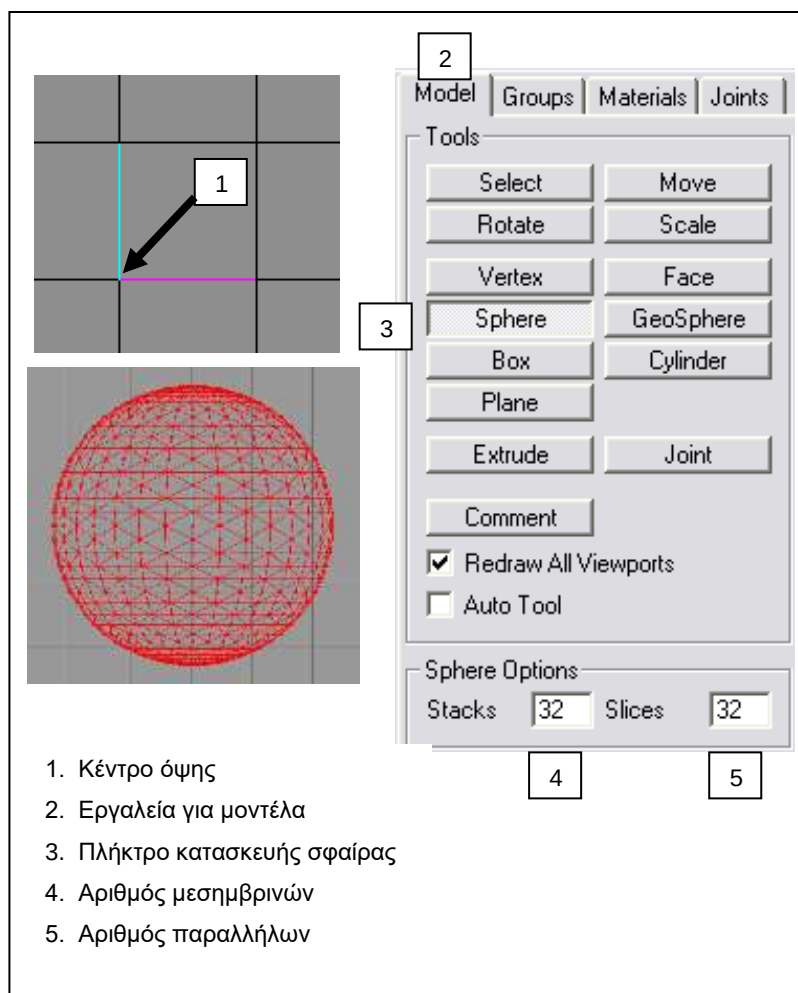


Εικόνα 10.15 Παράδειγμα αρθρώσεων και κοκάλων που προορίζονται για μοντέλο ανθρώπου

10.3.1 Ορισμός κοκάλων και αρθρώσεων

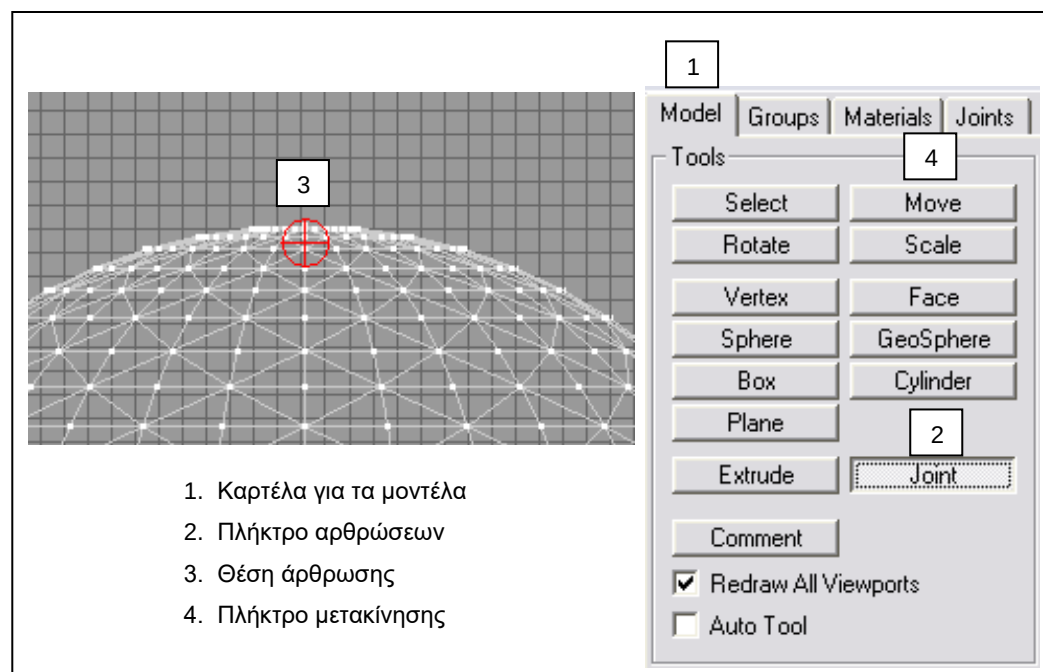
Θα παρουσιάσουμε μία πολύ απλή περίπτωση *animation* μοντέλου με τη χρήση μόνο ενός κοκάλου και δύο αρθρώσεων, για να δείξουμε τα επιπλέον βήματα που απαιτούνται. Ας υποθέσουμε λοιπόν ότι θέλουμε να κατασκευάσουμε το μοντέλο της Γης, και μάλιστα θέλουμε να περιστρέφεται.

Κατ' αρχάς κατασκευάζουμε τη σφαίρα από τη γραμμή εργαλείων, καρτέλα "Model" Model Groups Materials Joints και πατώντας το πλήκτρο κατασκευής σφαίρας. Ορίζουμε τον αριθμό των μεσημβρινών Stacks 32 και παραλλήλων Slices 32 που θέλουμε να έχει (εικόνα 10.16). Μεγαλύτερος αριθμός ισοδυναμεί σε καλύτερης ποιότητας σφαίρα, αλλά και με μεγαλύτερο αριθμό πολυγώνων. Κάνουμε κλικ στο κέντρο μίας από τις τρεις προβολές πλάγιων όψεων και σέρνουμε το ποντίκι μέχρι να οριστεί μία σφαίρα στο μέγεθος που επιθυμούμε.



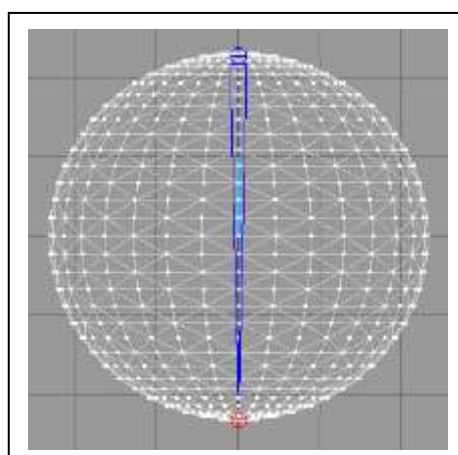
Εικόνα 10.16 Κατασκευή σφαίρας

Το επόμενο βήμα μας είναι να προσθέσουμε μία άρθρωση. Αυτό γίνεται από τη γραμμή εργαλείων, στην καρτέλα “Model” Model Groups Materials Joints και πατώντας το πλήκτρο για τη δημιουργία αρθρώσεων Joint (εικόνα 10.17). Κάνουμε κλικ σε μία από τις πλάγιες όψεις για να την τοποθετήσουμε. Φροντίζουμε η θέση της να είναι σ' έναν από τους δύο πόλους. Αν δε μας ικανοποιεί η θέση της, κάνουμε κλικ στο εργαλείο μετακίνησης Move και με το ποντίκι μας τη μετακινούμε σε άλλη θέση.



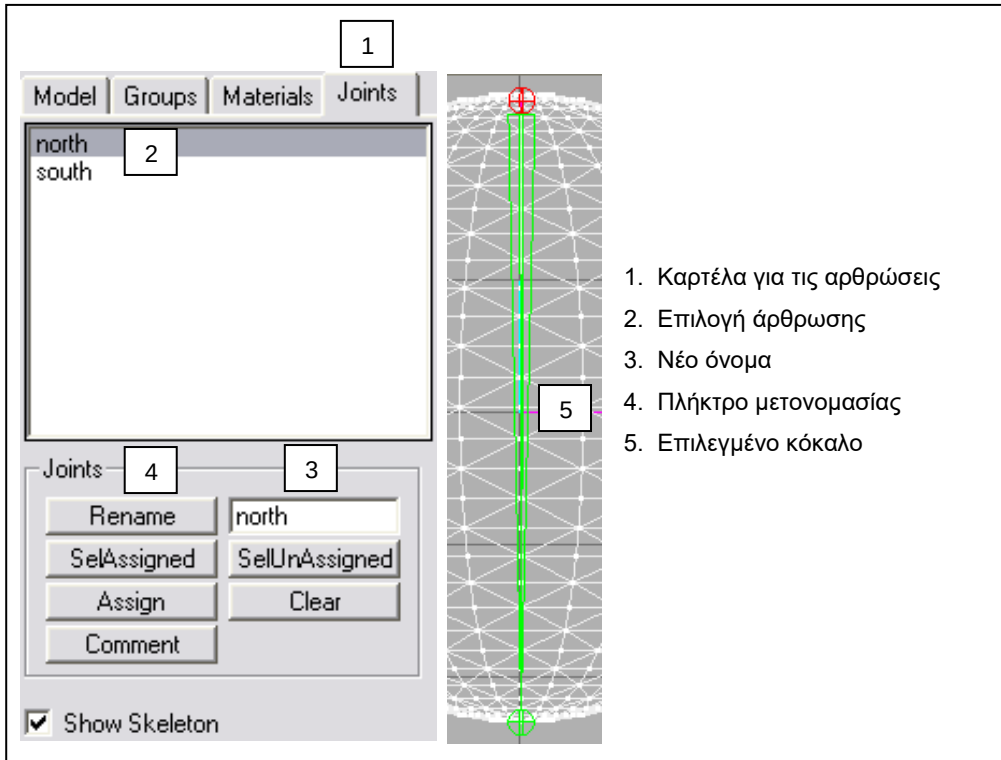
Εικόνα 10.17 Τοποθέτηση άρθρωσης

Μόλις ολοκληρώσουμε την τοποθέτηση και μετακίνηση της άρθρωσης, επιλέγουμε και πάλι το εργαλείο για τη δημιουργία αρθρώσεων . Για να ορίσουμε τη δεύτερη άρθρωση και το μεταξύ τους κόκαλο, κάνουμε κλικ στο δεύτερο πόλο ή κάνουμε κλικ στην πρώτη άρθρωση και σέρνουμε το ποντίκι μέχρι τον άλλο πόλο (εικόνα 10.18).



Εικόνα 10.18 Τοποθέτηση δεύτερης άρθρωσης και κοκάλου

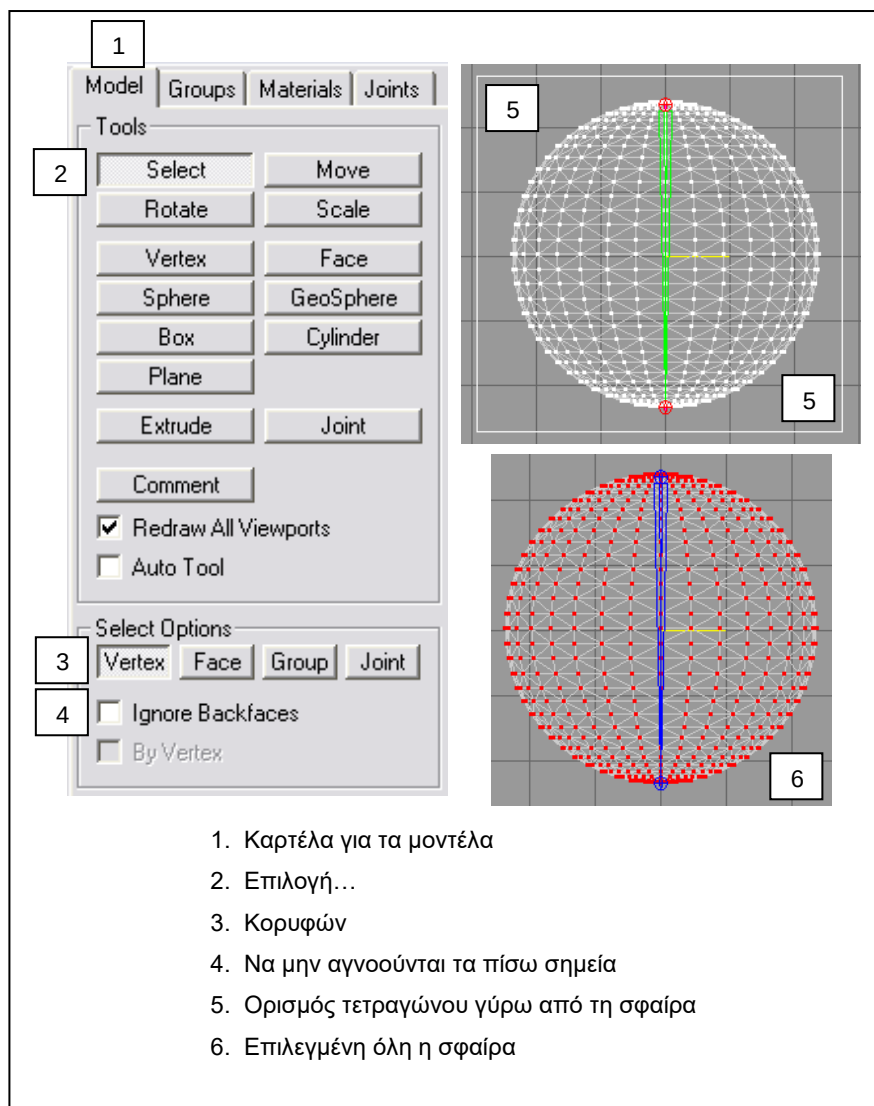
Και πάλι, αν η θέση του κοκάλου δεν είναι η επιθυμητή, με το εργαλείο μετακίνησης **Move** μπορούμε να του αλλάξουμε θέση. Μπορούμε επίσης να αλλάξουμε τα ονόματα των αρθρώσεων από τη γραμμή εργαλείων στην καρτέλα για τις αρθρώσεις “Joints” **Model** **Groups** **Materials** **Joints** (εικόνα 10.19). Επιλέγουμε την άρθρωση, δίνουμε το νέο όνομα στο αντίστοιχο πλαίσιο και κάνουμε κλικ στο πλήκτρο αλλαγής ονόματος **Rename**. Παρατηρούμε ότι, όταν επιλέξουμε την πρώτη άρθρωση, επιλέγεται και το κόκαλο που αντιστοιχεί σ' αυτή (το χρώμα γίνεται πράσινο).



Εικόνα 10.19 Μετονομασία αρθρώσεων

Έχοντας επιλεγμένο το κόκαλο, πηγαίνουμε και πάλι στην καρτέλα “Model” **Model** **Groups** **Materials** **Joints**, κάνουμε κλικ στο πλήκτρο επιλογής **Select** και αμέσως μετά πατάμε το πλήκτρο επιλογής κορυφών **Vertex**. Αφαιρούμε επίσης την επιλογή “Ignore Backfaces” (δηλαδή να μην αγνοούνται τα σημεία πίσω απ' αυτά που θα επιλέξουμε). Με αυτό τον τρόπο μπορούμε να επιλέξουμε ολόκληρη τη σφαίρα, ορίζοντας ένα τετράγωνο γύρω απ' αυτή με κλικ και σύρσιμο του ποντικιού σε οποιαδήποτε από τις πλάγιες όψεις (εικόνα 10.20). Αν για κάποιο λόγο το κόκαλο δεν ήταν επιλεγμένο, μπορούμε να το επιλέξουμε πάλι κάνοντας κλικ στο **Select**, κλικ στο

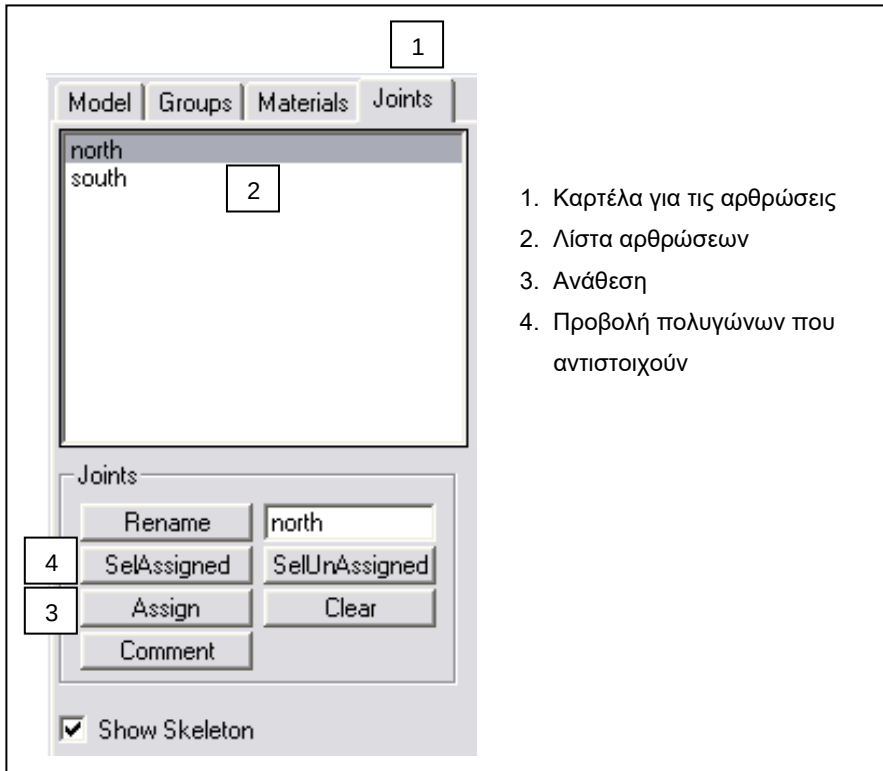
Joint και κλικ και σύρσιμο του ποντικιού γύρω απ' αυτό σε οποιαδήποτε από τις πλάγιες όψεις.



Εικόνα 10.20 Επιλογή πολυγώνων

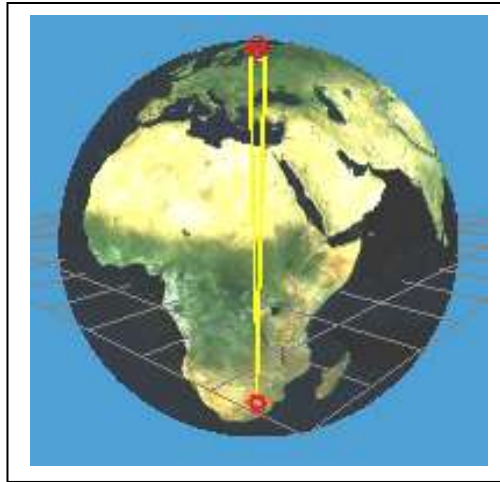
Με την επιλογή των πολύγωνων που επιθυμούμε (όλα στο παράδειγμά μας), το κόκαλο ξαναπαίρνει το αρχικό του μπλε χρώμα, αλλά εξακολουθεί να είναι επιλεγμένο. Πηγαίνουμε ξανά στην καρτέλα “Joints” **Model** **Groups** **Materials** **Joints** και πιέζουμε το πλήκτρο ανάθεσης **Assign** (εικόνα 10.21). Με αυτό τον τρόπο αναθέτουμε στο κόκα-

λο και την άρθρωση τα ανάλογα πολύγωνα. Από τώρα και στο εξής, όποτε κινήσουμε με οποιαδήποτε τρόπο το κόκαλο, θα κινηθεί μαζί του και όλη η σφαίρα. Για να βεβαιωθούμε ότι όντως αναθέσαμε τα σωστά πολύγωνα σε μία άρθρωση/κόκαλο, κάνουμε κλικ έξω από τη σφαίρα σε οποιαδήποτε από τις τρεις πλάγιες όψεις για να απο-επιλεγούν τα πολύγωνα, επιλέγουμε μία άρθρωση από τη λίστα (την πρώτη στο παράδειγμά μας) και κάνουμε κλικ στο πλήκτρο προβολής των πολυγώνων που αντιστοιχούν στην επιλεγμένη άρθρωση **SelAssigned**.



Εικόνα 10.21 Ανάθεση πολυγώνων σε άρθρωση

Για να έχουμε καλύτερη εποπτεία του μοντέλου μας, σ' αυτό το σημείο καλό είναι να ορίσουμε και να αναθέσουμε τις υφές του. Εφόσον το μοντέλο του παραδείγματος το κατασκευάσαμε μόνοι μας, δεν έχουν οριστεί το/τα διάφορα υλικά/χρώματα (Materials). Από τη στιγμή που αποτελείται από ένα και μόνο κομμάτι, πηγαίνουμε στην καρτέλα για τις ομάδες και το επιλέγουμε. Στη συνέχεια πηγαίνουμε στην καρτέλα για τα υλικά/χρώματα, ορίζουμε ένα νέο Material και αναθέτουμε μία εικόνα για υφή σ' αυτό (εικόνα 10.22).



Εικόνα 10.22 Το μοντέλο με την υφή, τις αρθρώσεις και το κόκαλο

Όταν σ' ένα μοντέλο δεν έχουν οριστεί χρώματα/υλικά (Materials), είτε γιατί το κατασκευάσαμε μόνοι μας είτε επειδή έτσι το βρήκαμε στο Διαδίκτυο, τότε πρέπει να ορίσουμε εμείς ένα ή περισσότερα πριν αναθέσουμε στο μοντέλο μας υφές. Στην ουσία κάθε ομάδα πολυγώνων, κάθε κομμάτι του μοντέλου, έχει ένα υλικό. Το υλικό είναι αυτό που μπορεί να έχει ένα χρώμα ή κάποια εικόνα.

10.3.2 Ορισμός της κίνησης

Έχοντας ορίσει τα κόκαλα και τις αρθρώσεις στο μοντέλο μας, το επόμενο βήμα είναι να ορίσουμε την κίνηση. Το πρώτο πράγμα που πρέπει να σκεφτούμε είναι η διάρκεια του animation και κατ' επέκταση ο αριθμός των καρτέ που θα χρειαστούμε.

Ας υποθέσουμε ότι η περιστροφή της Γης θέλουμε να ολοκληρώνεται σε 30 δευτερόλεπτα. Αυτό σημαίνει ότι χρειαζόμαστε 750 καρτέ (25 καρτέ/δευτερόλεπτο * 30 δευτερόλεπτα). Εφόσον ο κύκλος έχει 360°, κάθε δευτερόλεπτο θα πρέπει να περιστρέφεται κατά 12° (360° ÷ 30 δευτερόλεπτα). Αν θέλαμε η περιστροφή να είναι πιο γρήγορη ή πιο αργή, θα έπρεπε να αυξομειώσουμε ανάλογα τον αριθμό των καρτέ και τη γωνία περιστροφής.

Βεβαιωνόμαστε ότι είναι επιλεγμένο το κόκαλο που έχουμε ορίσει και οι αρθρώσεις του. Αν για κάποιο λόγο δεν είναι, μπορούμε να το επιλέξουμε πάλι κάνοντας κλικ στην καρτέλα “Model” Model Groups Materials Joints, κλικ στο Select, κλικ στο Joint, κλικ και σύρσιμο του ποντικιού γύρω απ' αυτό σε οποιαδήποτε από τις πλάγιες όψεις. Ενεργοποιούμε τη δυνατότητα προσθήκης κίνησης κάνοντας κλικ στο

πλήκτρο Anim Anim που βρίσκεται στο κάτω δεξιό μέρος της οθόνης του Milkshape3D. Το πρώτο μας βήμα είναι να ορίσουμε τον αριθμό των καρέ που επιθυμούμε, πληκτρολογώντας τον αριθμό τους στο αντίστοιχο πλαίσιο 750 (εικόνα 10.23).



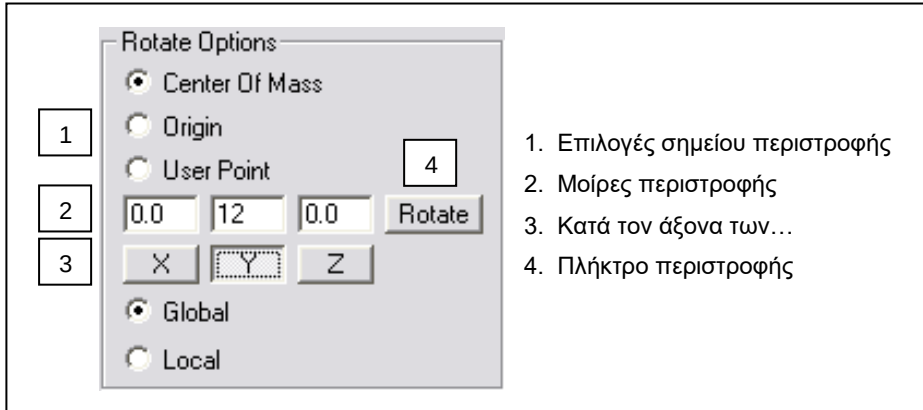
Εικόνα 10.23 Τα χειριστήρια του animation

Αυτό που θα κάνουμε στη συνέχεια είναι να αποθηκεύσουμε τις διαδοχικές θέσεις της Γης καθώς περιστρέφεται. Καθεμία τέτοια αποθηκευμένη θέση ονομάζεται Keyframe και θα χρειαστούμε 30 τέτοια (ένα για κάθε δευτερόλεπτο κίνησης). Προφανώς το πρώτο μας Keyframe είναι και το πρώτο καρέ.

Βεβαιωνόμαστε ότι έχουμε πιάσει το πλήκτρο Anim Anim και ότι το τρέχον μας καρέ είναι το πρώτο 1.0. Από τη γραμμή εντολών “Animate” επιλέγουμε “Set Keyframe”. Εναλλακτικά μπορούμε να πατήσουμε τα πλήκτρα Control και K (Ctrl+K).

Μετακινούμε την μπάρα κύλισης των καρέ 1.0 έτσι ώστε να μεταβούμε στο 25^ο ή πληκτρολογούμε απευθείας το νούμερο στο πλαίσιο του τρέχοντος καρέ 25 750. Από τη γραμμή εργαλείων, στην καρτέλα “Model” Model Groups Materials Joints κάνουμε κλικ στο πλήκτρο επιλογών περιστροφής Rotate. Στις ρυθμίσεις που εμφανίζονται δι-

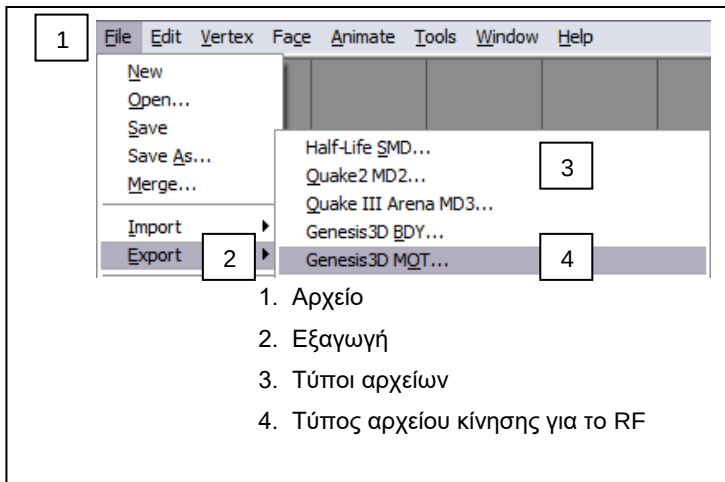
νομε την τιμή 12 στον άξονα των y (δηλαδή περιστροφή 12° κατά τον κάθετο άξονα) και κάνουμε κλικ στο πλήκτρο περιστροφής  που είναι ακριβώς δίπλα (εικόνα 10.24).



Εικόνα 10.24 Περιστροφή μοντέλου

Επαναλαμβάνουμε τη διαδικασία ορίζοντας και τα υπόλοιπα Keyframes. Μπορούμε ανά πάσα στιγμή να ελέγχουμε την κίνηση που κατασκευάζουμε, κάνοντας κλικ στα πλήκτρα εκτέλεσης (play) , προηγούμενου Keyframe , επόμενου Keyframe .

Όταν ολοκληρώσουμε το animation, είμαστε έτοιμοι να εξαγάγουμε το μοντέλο μας με τον τρόπο που αναφέραμε στην ενότητα 10.1. Το επιπλέον στοιχείο είναι ότι πρέπει να εξαγάγουμε σε ξεχωριστό αρχείο και την κίνησή του. Αυτό γίνεται και πάλι από τη γραμμή εντολών “File”, “Export”, αλλά αυτή τη φορά επιλέγουμε το Genesis 3D MOT (αρχείο κινήσεων συμβατό με το RF) (εικόνα 10.25). Κάθε animation ενός μοντέλου εξάγεται σε ξεχωριστό αρχείο, στον ίδιο φάκελο με το μοντέλο.



Εικόνα 10.25 Εξαγωγή animation

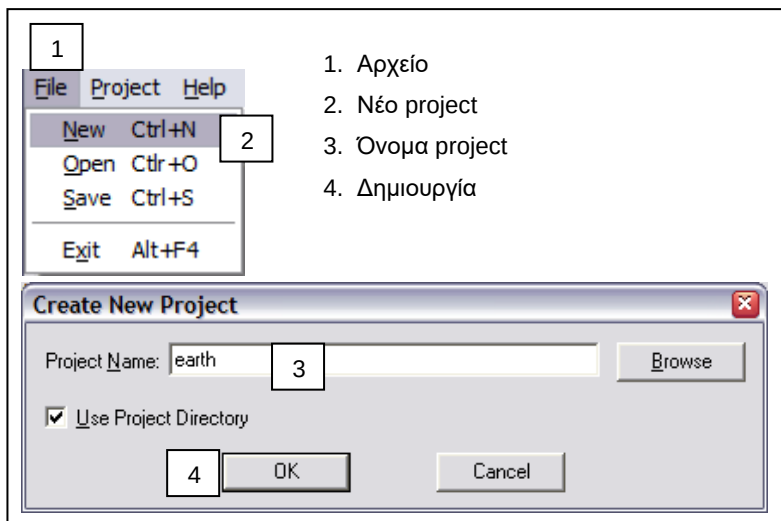
10.4 Συνθέτοντας έναν Actor

Τα τρισδιάστατα μοντέλα στο RF ονομάζονται *Actors* και είναι η σύνθεση των αρχείων .bdy (αρχείο μοντέλου με τις υφές) και .mot (αρχείο animation) τα οποία κατασκευάσαμε με το Milkshape3D. Η διαδικασία σύνθεσης είναι απλή και γίνεται με τη χρήση του ανάλογου προγράμματος/εργαλείου, του *Genesis 3D Actor Studio*, που βρίσκεται όπως και τα υπόλοιπα εργαλεία του RF στο φάκελο C:\RealityFactory\Tools (Astudio.exe). Μπορούμε να το εκκινήσουμε είτε από την “Εξερεύνηση των Windows” είτε από την “Εναρξη”, “Όλα τα προγράμματα” είτε από την εργαλειοθήκη-Traybar.

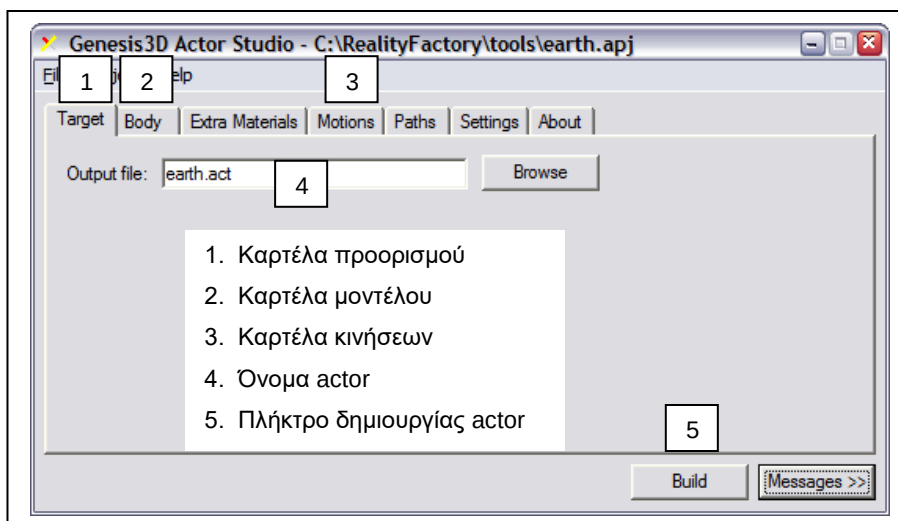
Για την κατασκευή των actors είναι απαραίτητο να μεταφερθούν τα αρχεία .bdy και .mot από το φάκελο όπου τα αποθηκεύσαμε κατά την επεξεργασία με το Milkshape3D, στο φάκελο C:\RealityFactory\Tools. Δε χρειάζεται να μεταφερθούν τα αρχεία εικόνων/υφών, γιατί οι υφές έχουν ήδη ενσωματωθεί στα αρχεία .bdy.

Το πρώτο βήμα είναι η δημιουργία ενός νέου project (εικόνα 10.26). Πρέπει να σημειώσουμε ότι το όνομα του project δεν παίζει ρόλο στο όνομα του αρχείου του actor που θα δημιουργήσουμε. Μέσω ενός και μόνο project μπορούμε να δημιουργήσουμε πολλούς actors.

Αμέσως μετά τη δημιουργία του project, στην καρτέλα προορισμού “Target” το πρόγραμμά μας προτείνει το όνομα του actor, το οποίο μπορούμε να αλλάξουμε (εικόνα 10.27).

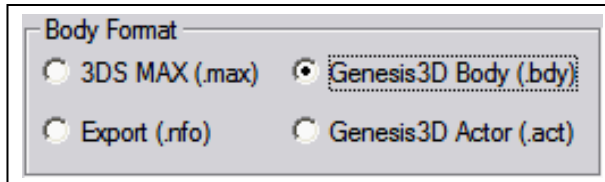


Εικόνα 10.26 Δημιουργία project για την κατασκευή actor



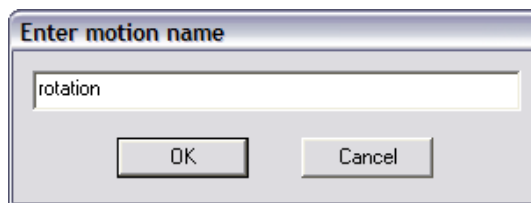
Εικόνα 10.27 Οι επιλογές του Actor Studio

Το επόμενο βήμα είναι να δηλώσουμε ποιο αρχείο .bdy θα χρησιμοποιήσουμε για το σώμα του actor. Αυτό γίνεται από την καρτέλα “Body” **Target** **Body** **Extra Materials**. Με το πλήκτρο της αναζήτησης **Browse** επιλέγουμε το ανάλογο αρχείο. Ο τύπος του αρχείου δηλώνεται αυτόματα με την επιλογή του αρχείου (εικόνα 10.28).



Εικόνα 10.28 Επιλογή τύπου αρχείου

Στη συνέχεια δηλώνουμε το/τα αρχεία animation στην καρτέλα “Motions” **Extra Materials** **Motions** **Paths**. Πρώτα δηλώνουμε το όνομα που θέλουμε να έχει η κίνηση κάνοντας κλικ στο πλήκτρο προσθήκης **Add** και πληκτρολογώντας ένα όνομα (εικόνα 10.29). Στη συνέχεια, από το πλήκτρο αναζήτησης **Browse** επιλέγουμε το αντίστοιχο αρχείο .mot. Και πάλι ο τύπος του αρχείου επιλέγεται αυτόματα. Αν υπάρχει και άλλο animation, επαναλαμβάνουμε τη διαδικασία προσθέτοντας νέο όνομα κίνησης και συνδέοντάς τη με άλλο αρχείο animation.



Εικόνα 10.29 Εισαγωγή ονόματος κίνησης

Το τελευταίο βήμα μας είναι να κάνουμε κλικ στο πλήκτρο κατασκευής **Build** που βρίσκεται κάτω δεξιά. Το πρόγραμμα με μήνυμα θα μας ενημερώσει για την επιτυχή ή μη κατασκευή του αρχείου actor (.act).

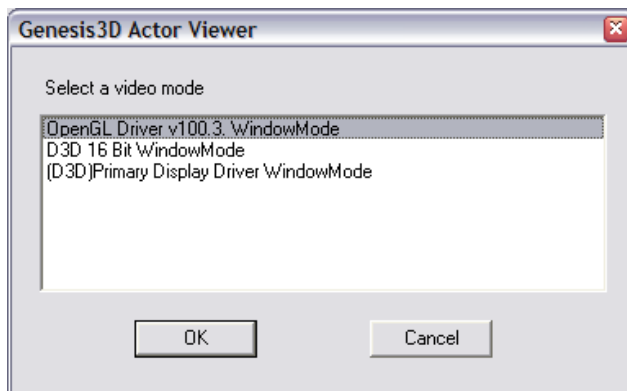
Ένα μοντέλο μπορεί να έχει περισσότερα από ένα animation. Για παράδειγμα, το μοντέλο ενός ανθρώπου μπορεί να περπατά, να τρέχει, να σκύβει, κτλ. Καθεμία τέτοια ενέργεια προϋποθέτει την ύπαρξη αντίστοιχου αρχείου .mot (αρχείο animation). Το μοντέλο (actor) που κατασκευάζουμε αποθηκεύεται αυτόματα στο φάκελο C:\RealityFactory\Tools.

10.5 Προεπισκόπηση Actors

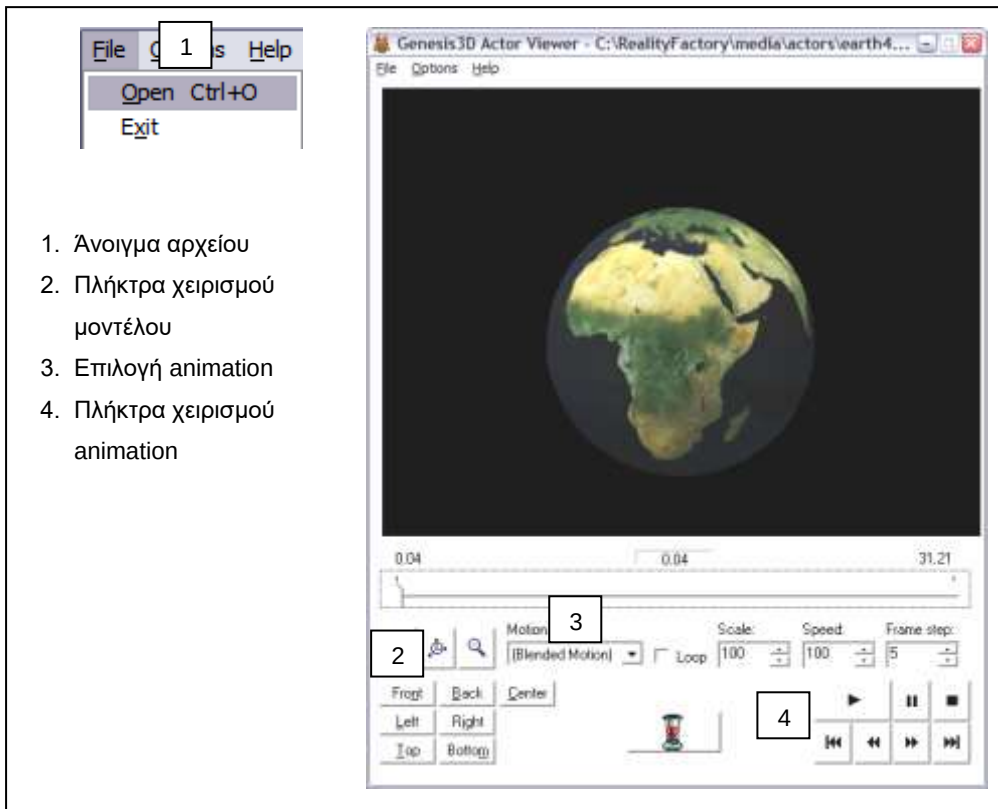
Έχοντας ολοκληρώσει τη διαδικασία κατασκευής ενός actor, μπορούμε να μεταφέρουμε το αρχείο που δημιουργήθηκε από το φάκελο C:\RealityFactory\Tools στο φάκελο που είναι υποχρεωτικό να βρίσκονται όλοι οι actors, δηλαδή στο C:\RealityFactory\Media\Actors. Πριν διαγράψουμε τα αρχεία που δε μας είναι πια απαραίτητα, μπορούμε να δούμε το μοντέλο που φτιάξαμε, χρησιμοποιώντας το πρόγραμμα Act

View που βρίσκεται στο φάκελο C:\RealityFactory\Tools (ActView.exe). Όπως και τα υπόλοιπα προγράμματα, μπορούμε να το εκκινήσουμε είτε από την “Εξερεύνηση των Windows” είτε από την “Εναρξη”, “Όλα τα προγράμματα” είτε από την εργαλειοθήκη-Traybar.

Θα μας ζητηθεί αρχικά να ορίσουμε τον οδηγό βίντεο που θα χρησιμοποιήσει το πρόγραμμα (μπορούμε να επιλέξουμε οποιονδήποτε από τους τρεις που προτείνονται) (εικόνα 10.30). Στη συνέχεια μπορούμε να ανοίξουμε ένα αρχείο actor από τη γραμμή εργαλείων “File”, “Open”, να το μετακινήσουμε στην οθόνη , να το περιστρέψουμε  και να το μεγεθύνουμε . Καθεμία από τις παραπάνω ενέργειες εκτελείται αφού επιλέξουμε το ανάλογο πλήκτρο, κάνουμε κλικ και σύρουμε το ποντίκι στο χώρο της προεπισκόπησης. Μπορούμε επίσης να δούμε το/τα animations που έχει το μοντέλο μας από την αντίστοιχη επιλογή και πατώντας το πλήκτρο  (εικόνα 10.31).



Εικόνα 10.30 Επιλογή οδηγού βίντεο που θα χρησιμοποιήσει το Act View



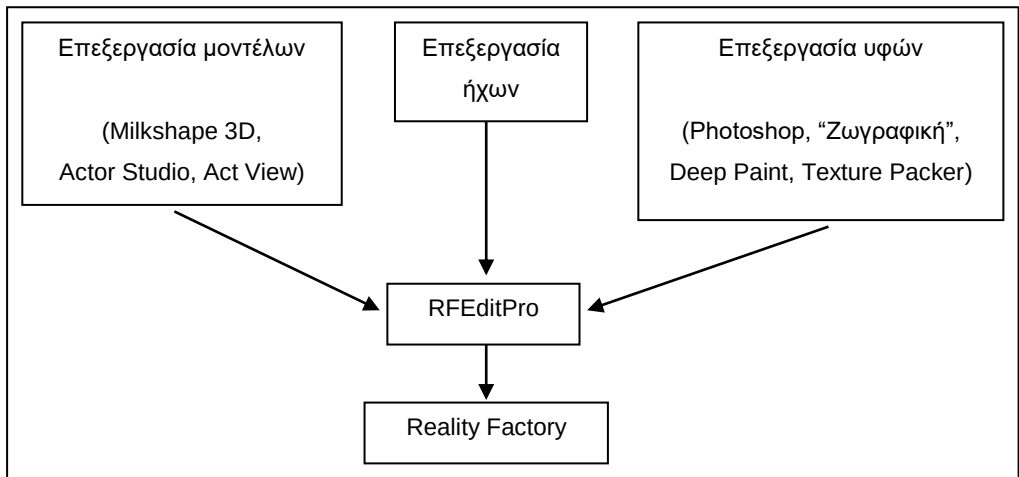
Εικόνα 10.31 Το πρόγραμμα Act View

Εφόσον με το Act View διαπιστώσουμε ότι το μοντέλο που κατασκευάσαμε ικανοποιεί τις απαιτήσεις μας, μπορούμε πλέον να διαγράψουμε τα αρχεία που δε μας είναι πια απαραίτητα. Βέβαια, αν επιθυμούμε να τα διατηρήσουμε για λόγους ασφαλείας, μπορούμε να τα μεταφέρουμε σε κάποιον άλλο φάκελο. Τα αρχεία αυτά είναι: οι εικόνες που χρησιμοποιήσαμε για υφές που βρίσκονται στο φάκελο όπου επεξεργαστήκαμε αρχικά το μοντέλο με το Milkshape 3D και τα αρχεία .bdy και .mot που μεταφέραμε στο φάκελο C:\RealityFactory\Tools. Μπορούμε επίσης να διαγράψουμε άφοβα τα αρχεία που βρίσκονται στο φάκελο C:\RealityFactory\Tools\BldTemp, γιατί είναι προσωρινά αρχεία που δημιουργούνται κατά την κατασκευή του actor με το Actor Studio. Δεν ξεχνάμε βέβαια να μεταφέρουμε τον/τους actors που κατασκευάσαμε στο φάκελο C:\RealityFactory\Media\Actors.

Κεφάλαιο 11°

*Το περιβάλλον εργασίας του RFEditPro
Κατασκευή και εκτέλεση ενός βασικού επιπέδου*

Μέχρι στιγμής στην παρουσίασή μας ασχοληθήκαμε με τη διαδικασία διαμόρφωσης και προσαρμογής του υλικού (εικόνες/υφές και τρισδιάστατα μοντέλα), έτσι ώστε να μπορούμε να το χρησιμοποιήσουμε στο RF. Όμως η κυρίως εργασία γίνεται με το RFEEditPro, που ανήκει στην κατηγορία προγραμμάτων που ονομάζονται επεξεργαστές κόσμων (*World Editors*). Θα πρέπει να θεωρήσουμε το RFEEditPro ως ένα πρόγραμμα που μας παρέχει το χώρο στον οποίο τοποθετούμε και συνθέτουμε τα στοιχεία που θα αποτελέσουν τον εικονικό μας κόσμο και παράγει αρχεία τα οποία εκτελούνται από τη μηχανή παιχνιδιού, που είναι το RF (σχήμα 11.1).



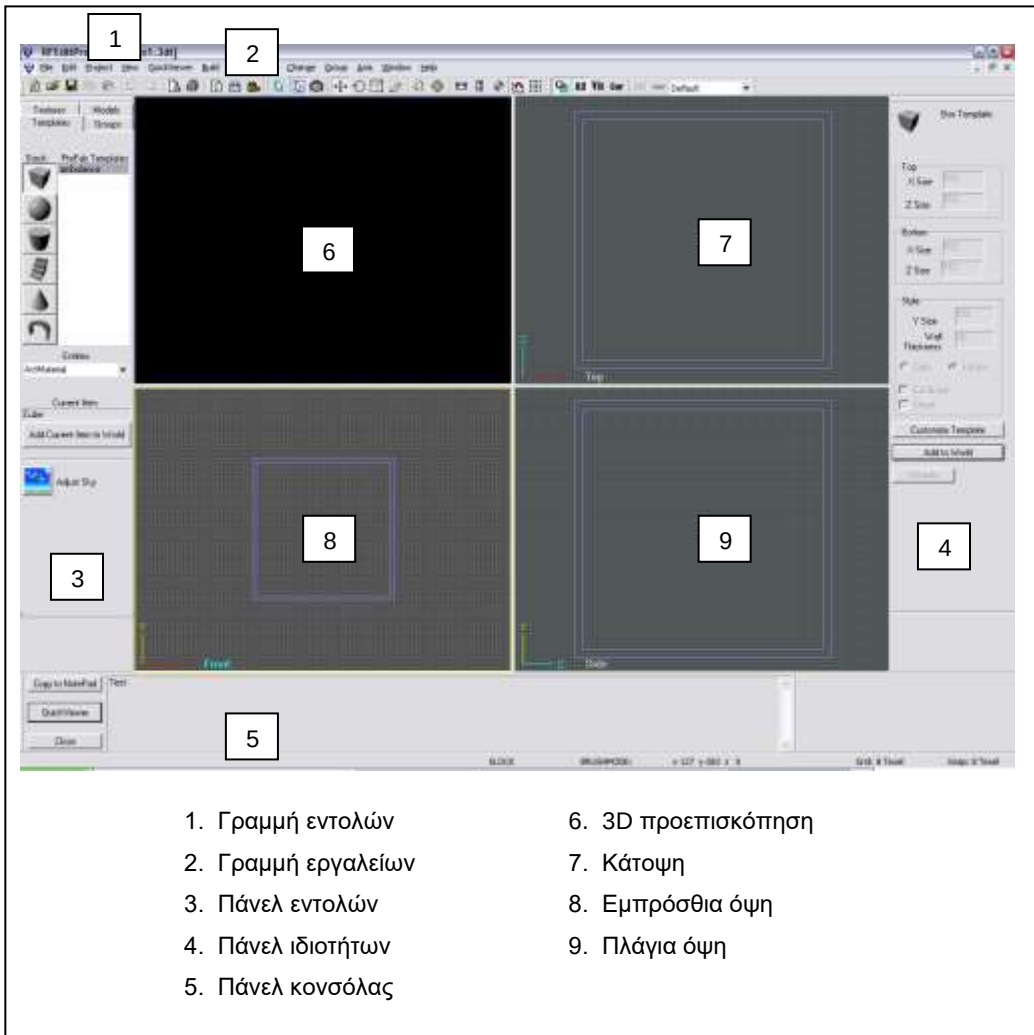
Σχήμα 11.1 Ο ρόλος του RFEEditPro

Συγκεκριμένα, το RFEEditPro παράγει δύο τύπους αρχείων:

1. Αρχεία *.3dt*, που είναι αρχεία κειμένου τα οποία περιέχουν όλες τις απαραίτητες πληροφορίες για τη δημιουργία του εικονικού κόσμου. Το RFEEditPro “διαβάζει” αυτές τις πληροφορίες και παρουσιάζει τον κόσμο στην οθόνη μας για να τον επεξεργαστούμε.
2. Αρχεία *.bsp (Binary Space Partitioning)*, που είναι οι κόσμοι/επίπεδα που “φορτώνονται” και εκτελούνται στο RF. Τα αρχεία αυτά υλοποιούν μία συγκεκριμένη τεχνική δημιουργίας μίας μεγάλης περιοχής με γεωμετρικά στερεά διαιρώντας τη σε πολλές μικρότερες.

Κατά τη διάρκεια της κατασκευής ενός εικονικού κόσμου επεξεργαζόμαστε ένα αρχείο *.3dt*, το οποίο μπορούμε ανά πάσα στιγμή να μετατρέψουμε σε *.bsp* με τη διαδικασία της μεταγλώττισης (*compilation*) και να το εκτελέσουμε στο RF.

Το πρόγραμμα βρίσκεται στο φάκελο C:\RealityFactory\Tools (RFEEditPro.exe). Όπως και τα υπόλοιπα προγράμματα, μπορούμε να το εκκινήσουμε είτε από την “Εξερεύνηση των Windows” είτε από την “Εναρξη”, “Όλα τα προγράμματα” είτε από την εργαλειοθήκη-Traybar (εικόνα 11.1).



Εικόνα 11.1 Το περιβάλλον εργασίας του RFEEditPro

Παρατηρούμε ότι το περιβάλλον εργασίας θυμίζει αρκετά αυτό του Milkshape3D. Το κέντρο της οθόνης καταλαμβάνεται από τα τέσσερα παράθυρα προεπισκόπησης (τρισδιάστατη, κάτοψη, εμπρόσθια και πλάγια όψη). Όπως στις περισσότερες παραθυρικές εφαρμογές, η πρώτη γραμμή είναι η γραμμή εντολών και από κάτω της βρί-

σκεται η γραμμή εργαλείων. Με τρόπο αντίστοιχο με το Milkshape3D, το αριστερό, δεξιό και κάτω μέρος της οθόνης καταλαμβάνονται από τρία πάνελ. Το πάνελ εντολών (αριστερά) σχετίζεται άμεσα με το πάνελ ιδιοτήτων (δεξιά), με την έννοια ότι οι επιλογές που κάνει ο χρήστης στο πρώτο αναλύονται και παραμετροποιούνται περαιτέρω με επιλογές στο δεύτερο.

Παρότι ένας άπειρος χρήστης μπορεί να βρεθεί σε σύγχυση από την πληθώρα των επιλογών, εντούτοις πρέπει να σημειώσουμε ότι το μεγαλύτερο μέρος της εργασίας για την κατασκευή ενός εικονικού κόσμου γίνεται με τη χρήση μερικών μόνο επιλογών απ' αυτά τα δύο πάνελ και ορισμένων –λίγων επίσης– εργαλείων από τις γραμμές εντολών και εργαλείων.

Η ορολογία που χρησιμοποιείται στο RFEditPro

Actors: Τα επεξεργασμένα με το Actor Studio τρισδιάστατα μοντέλα αντικειμένων.

Πλέγμα-Grid: Ο χώρος όπου τοποθετούνται όλα όσα απαρτίζουν τον εικονικό κόσμο.

Εφαρμογή-Project και Επίπεδο-Level: Η βασική εργασία του RFEditPro είναι η κατασκευή τρισδιάστατων παιχνιδιών (εφαρμογών), που αποτελούνται από διαφορετικά επίπεδα. Στην περίπτωση των εφαρμογών Ε.Π. τα επίπεδα μπορούν να αντιπροσωπεύουν διαφορετικά μέρη της εφαρμογής.

Οντότητα-Entity: Στοιχεία σε μία εφαρμογή που μπορούν να παραμετροποιηθούν. Για παράδειγμα, τοποθετούμε στο επίπεδο μία οντότητα χειρισμού τρισδιάστατων αντικειμένων και στη συνέχεια ορίζουμε ποιο αντικείμενο θα ελέγχει και με ποιον τρόπο. Οντότητες χειρίζονται επίσης τα φώτα, τα διάφορα εφέ, τους ήχους, την κίνηση αντικειμένων, κτλ.

Στερεό-Brush: Είναι τα βασικά γεωμετρικά στερεά καθώς επίσης οι ασπίδες και οι σκάλες.

Έδρα-Face: Οι έδρες ενός στερεού.

Μοντέλο-Model: Γεωμετρικό στερεό-brush που μπορεί να κινηθεί μέσω κάποιας οντότητας-entity ή που μπορεί να αποκτήσει συγκεκριμένες ιδιότητες.

Texel: Ιδιαίτερη μονάδα μέτρησης του RFEditPro που αντιστοιχεί σε 1 ίντσα. Για παράδειγμα, 72 texels αντιστοιχούν περίπου σε 180 εκατοστά και τα 4.000 texels είναι περίπου 100 μέτρα.

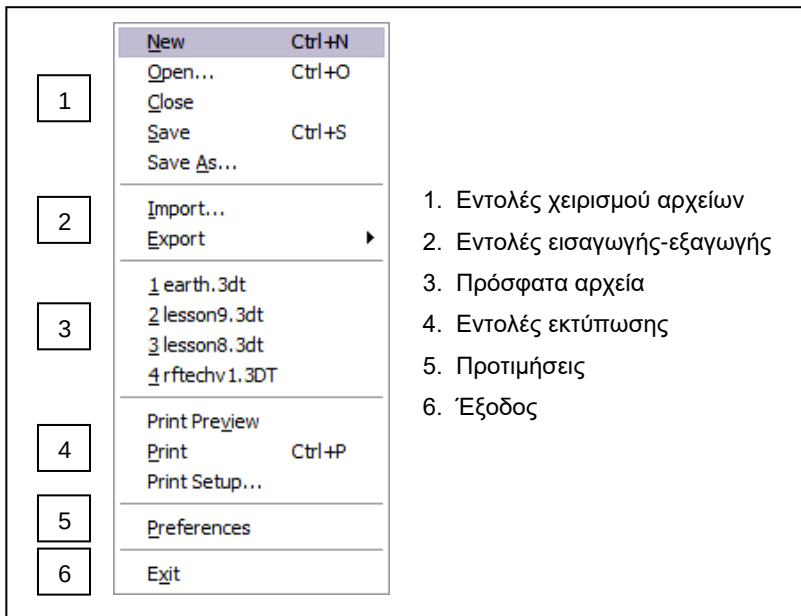
Πιόνι-Pawn: Οντότητα που μπορούμε να χειριστούμε προγραμματιστικά.

11.1 Γραμμές εντολών και εργαλείων

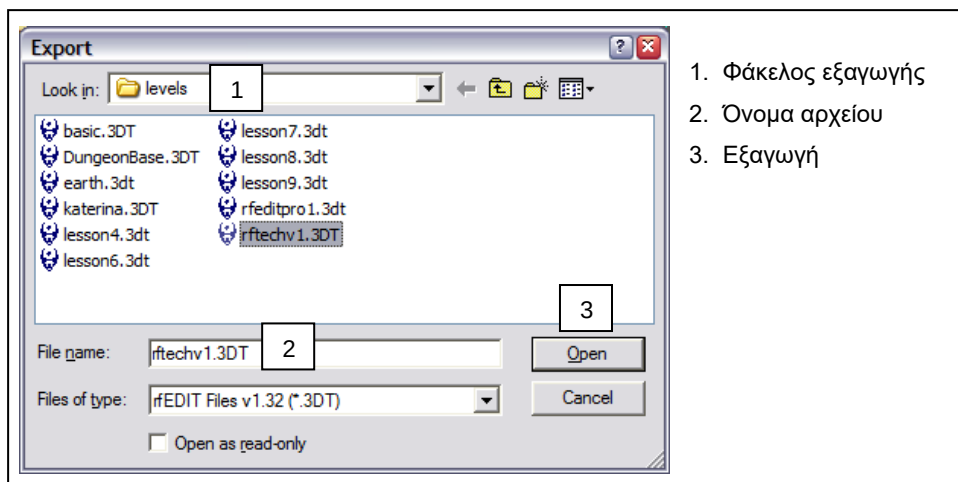
11.1.1 Γραμμή εντολών Αρχείο-File

Η πρώτη επιλογή από τη γραμμή εντολών “Αρχείο-File” αφορά το χειρισμό των αρχείων του RFEditPro (εικόνα 11.2). Συγκεκριμένα, μπορούμε να δημιουργήσουμε καινούργιο (New), να ανοίξουμε ένα ήδη υπάρχον (Open), να αποθηκεύσουμε κάποιο για πρώτη φορά ή με άλλο όνομα (Save as) και να αποθηκεύσουμε τις αλλαγές σε κάποιο αρχείο που επεξεργαστήκαμε (Save). Τα τρία πρώτα εικονίδια της γραμμής εργαλείων αφορούν αντίστοιχα τη δημιουργία , το άνοιγμα  και την αποθήκευση .

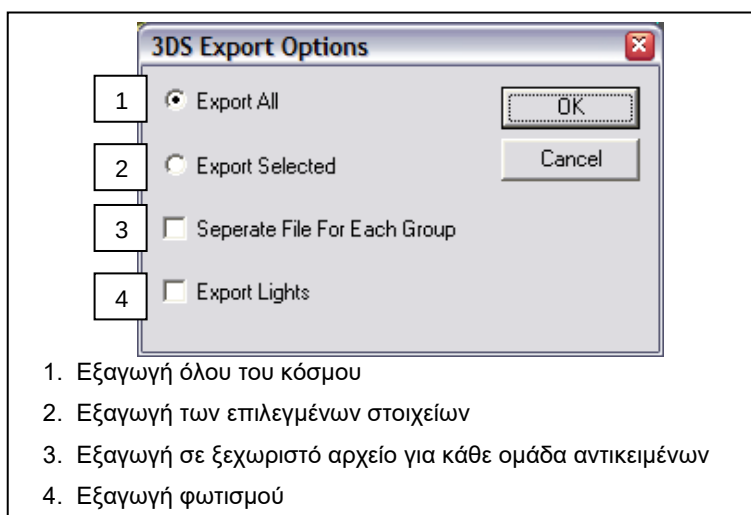
Μπορούμε επίσης να εισαγάγουμε (Import) και να εξαγάγουμε (Export) αρχεία .3dt (εικόνα 11.3). Κάτι τέτοιο μάς επιτρέπει να εισαγάγουμε έτοιμα κομμάτια ενός εικονικού κόσμου σ' έναν άλλο, επιταχύνοντας τη διαδικασία κατασκευής. Όσον αφορά την εξαγωγή, έχουμε τη δυνατότητα αυτή να γίνει με τη μορφή αρχείων .3dt (για επεξεργασία από το RFEditPro), αλλά και σε μορφή αρχείων Autodesk .3ds, έτσι ώστε να επεξεργαστούμε τον εικονικό μας κόσμο με το 3D Studio Max, που είναι από τα κορυφαία προγράμματα επεξεργασίας τρισδιάστατων γραφικών (εικόνα 11.4).



Εικόνα 11.2 Επιλογές γραμμής εντολών-Αρχείο

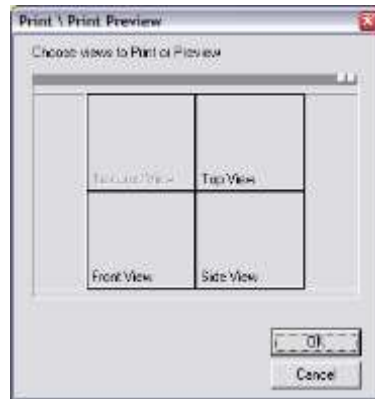


Εικόνα 11.3 Καρτέλα εισαγωγής-εξαγωγής αρχείων σε μορφή .3dt



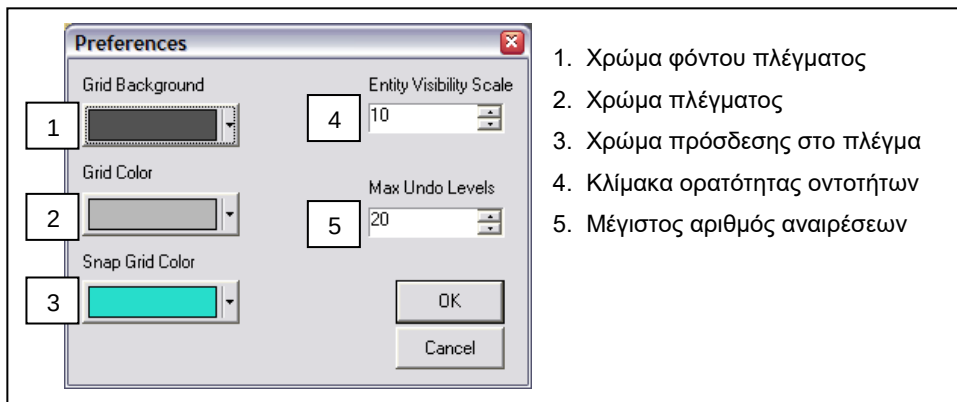
Εικόνα 11.4 Καρτέλα επιλογών εξαγωγής αρχείων σε μορφή .3ds

Μπορούμε να εκτυπώσουμε την κάτοψη, την πλάγια ή την εμπρόσθια όψη του κόσμου μας με την επιλογή Print Preview (εικόνα 11.5) και να καθορίσουμε τις παραμέτρους εκτύπωσης από το Print Setup. Αντίστοιχα, στη γραμμή εργαλείων υπάρχουν τα πλήκτρα προεπισκόπησης εκτύπωσης  και εκτύπωσης .



Εικόνα 11.5 Καρτέλα επιλογής όψης για εκτύπωση

Η επιλογή Προτιμήσεις (Preferences) μας επιτρέπει να καθορίσουμε το χρώμα φόντου του πλέγματος, το χρώμα του πλέγματος και το χρώμα των κουκκίδων που απεικονίζουν τα σημεία πρόσδεσης των αντικειμένων με το πλέγμα (εικόνα 11.6). Αυτές οι επιλογές μάς βοηθούν να εργαζόμαστε με μεγαλύτερη εργονομία όταν τοποθετούμε διάφορα αντικείμενα στο πλέγμα. Όταν τοποθετούμε διάφορες οντότητες στον εικονικό μας κόσμο, αυτές αναπαριστούνται μ' ένα τετραγωνάκι. Η επιλογή της κλίμακας ορατότητας οντοτήτων μάς επιτρέπει να καθορίσουμε το μέγεθος που θα έχει το τετραγωνάκι αυτό, ώστε να εντοπίζεται ευκολότερα μέσα στον κόσμο. Τέλος, μπορούμε να ορίσουμε πόσες ενέργειές μας μπορεί να αιρέσει το RFEEditPro.

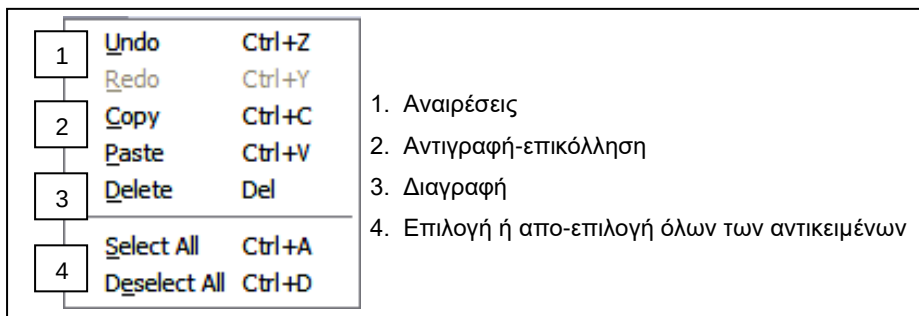


Εικόνα 11.6 Καρτέλα προτιμήσεων

11.1.2 Γραμμή εντολών Επεξεργασία-Edit

Η δεύτερη επιλογή από τη γραμμή εντολών είναι αυτή της “Επεξεργασίας-Edit”. Αφορά τις δυνατότητες αναιρέσης, αντιγραφής και επικόλλησης, διαγραφής αντικειμένων,

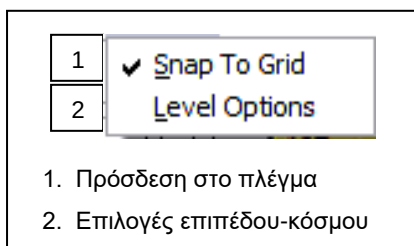
καθώς επίσης και τη δυνατότητα επιλογής ή απο-επιλογής όλων των στοιχείων του ει-
κονικού κόσμου (εικόνα 11.7). Αντίστοιχα, τα εικονίδια της γραμμής εργαλείων που
αφορούν εντολές επεξεργασίας είναι της αντιγραφής , της επικόλλησης  και των
αναιρέσεων  .



Εικόνα 11.7 Επιλογές γραμμής εντολών-Επεξεργασία

11.1.3 Γραμμή εντολών Project

Στο RFEditPro, Project θεωρείται ο εικονικός κόσμος που κατασκευάζουμε. Έτσι, από
την αντίστοιχη γραμμή εντολών (εικόνα 11.8) μπορούμε κατ' αρχάς να ορίσουμε να
προσδένονται όλα τα αντικείμενα στο πλέγμα, κάτι που είναι πάντοτε προεπιλεγμένο.
Αυτό είναι ιδιαίτερα σημαντικό, γιατί μας επιτρέπει να τοποθετούνται όλα τα αντικείμε-
νά μας ακριβώς στη θέση που επιθυμούμε, αναγκάζοντάς τα να πάρουν συγκεκριμένη
θέση πάνω στο πλέγμα (στις μικρές κουκκίδες του). Η επιλογή αυτή μάς διευκολύνει
επίσης στην τοποθέτηση ενός αντικειμένου δίπλα σ' ένα άλλο. Στη σπάνια περίπτωση
που ένα αντικείμενο δεν τοποθετείται στη θέση που θέλουμε, μπορούμε να απο-
επιλέξουμε την πρόσδεση στο πλέγμα. Το αντίστοιχο εικονίδιο στη γραμμή εργαλείων
είναι το .



Εικόνα 11.8 Επιλογές γραμμής εντολών-Project

Από τις Επιλογές Επιπέδου (Level Options) μπορούμε να κάνουμε μία σειρά από πο-
λύ σημαντικές ρυθμίσεις (εικόνα 11.9). Για γρήγορη πρόσβαση στο μενού αυτό, μπο-
ρούμε να κάνουμε κλικ στο εικονίδιο  της γραμμής εργαλείων. Οι πρώτες τέσσερις

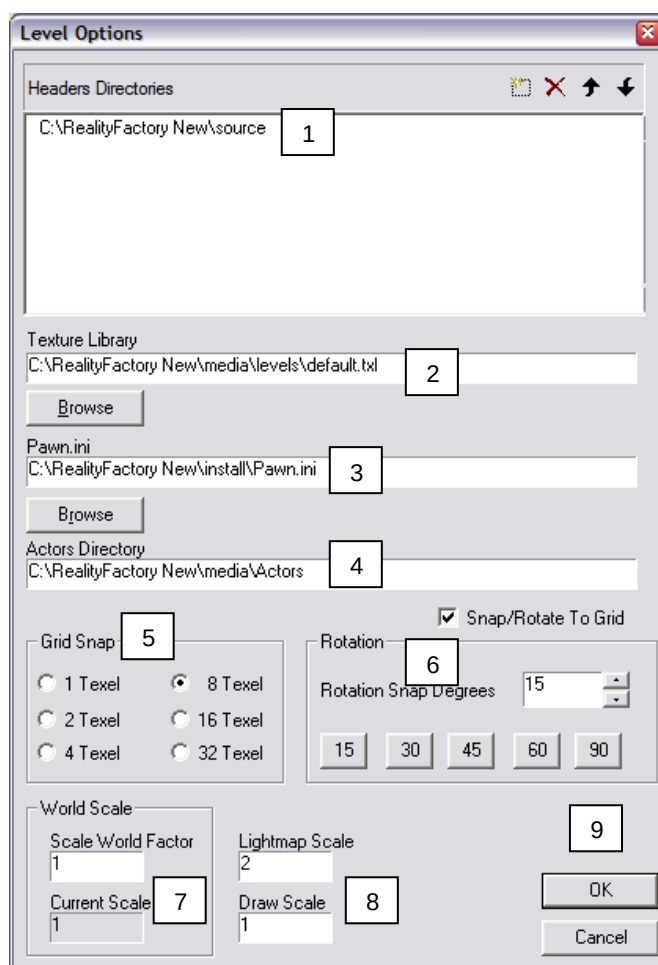
επιλογές αφορούν το ορισμό της θέσης και του ονόματος ορισμένων αρχείων που θα χρησιμοποιεί η εφαρμογή μας. Κατ' αρχάς ορίζουμε σε ποιο φάκελο είναι τοποθετημένο το αρχείο που περιγράφει όλες τις οντότητες του RFEEditPro (προεπιλεγμένος είναι ο C:\RealityFactory\source και δε χρειάζεται να τον τροποποιήσουμε).

Στη συνέχεια, μπορούμε να ορίσουμε ποιο πακέτο υφών θα χρησιμοποιεί η εφαρμογή μας, η παραγωγή του οποίου περιγράφηκε στο 9^ο κεφάλαιο. Εφόσον όλα τα πακέτα υφών είναι αποθηκευμένα στον ίδιο φάκελο (C:\RealityFactory\media\levels\), το μόνο που έχουμε να κάνουμε είναι να αλλάξουμε το όνομα του αρχείου, προσέχοντας να διατηρήσουμε την κατάληξη .txt.

Η επόμενη επιλογή αφορά τον ορισμό του αρχείου που περιγράφει τα πόνια της εφαρμογής μας. Αν δε χρησιμοποιούμε πόνια, αφήνουμε τη ρύθμιση ως έχει. Η τελευταία επιλογή ορισμού αρχείων αφορά το φάκελο στον οποίο βρίσκονται τα τρισδιάστατα μοντέλα μας (προεπιλεγμένος είναι ο C:\RealityFactory\media\Actors).

Η προεπιλογή της ακρίβειας πρόσδεσης στο πλέγμα είναι τα 8 texels. Σε περίπτωση που επιθυμούμε μεγαλύτερη ή μικρότερη ακρίβεια, μπορούμε να το κάνουμε από την επιλογή Grid Snap. Με την επιλογή Snap/Rotate to Grid μπορούμε να ορίσουμε αν τα αντικείμενά μας θα προσδένονται στο πλέγμα και κατά την περιστροφή τους. Για να προσδεθεί όμως στο πλέγμα ένα αντικείμενο κατά την περιστροφή του, αυτή πρέπει να γίνει κατά ένα συγκεκριμένο αριθμό μοιρών, ο οποίος καθορίζεται από την επιλογή Rotation Snap Degrees. Αν οι μοίρες που περιστρέφουμε ένα αντικείμενο είναι λιγότερες από την επιλογή που κάναμε, τότε το αντικείμενο επιστρέφει στην προηγούμενη θέση του.

Μπορούμε να αυξήσουμε ή να ελαττώσουμε την κλίμακα ολόκληρου του κόσμου μας από την επιλογή Scale World Factor. Τιμές μικρότερες του 1 (σε δεκαδικό αριθμό γραμμένο με το αγγλικό σύστημα, π.χ. 0.3) προκαλούν σμίκρυνση, ενώ τιμές μεγαλύτερες του 1 προκαλούν μεγέθυνση. Αντίστοιχα, μπορούμε να αυξομειώσουμε την κλίμακα υπολογισμού των φωτισμών –και των συνακόλουθων σκιάσεων– σ' όλη την εφαρμογή μας από την επιλογή Lightmap Scale. Τιμές μικρότερες του 2 δημιουργούν καλύτερες σκίες, αλλά έχουν επίπτωση στην ταχύτητα εκτέλεσης της εφαρμογής. Τέλος, η επιλογή Draw Scale αφορά την κλίμακα –σε σχέση με το μέγεθος της αρχικής εικόνας– με την οποία τοποθετούνται οι υφές πάνω στα αντικείμενα.

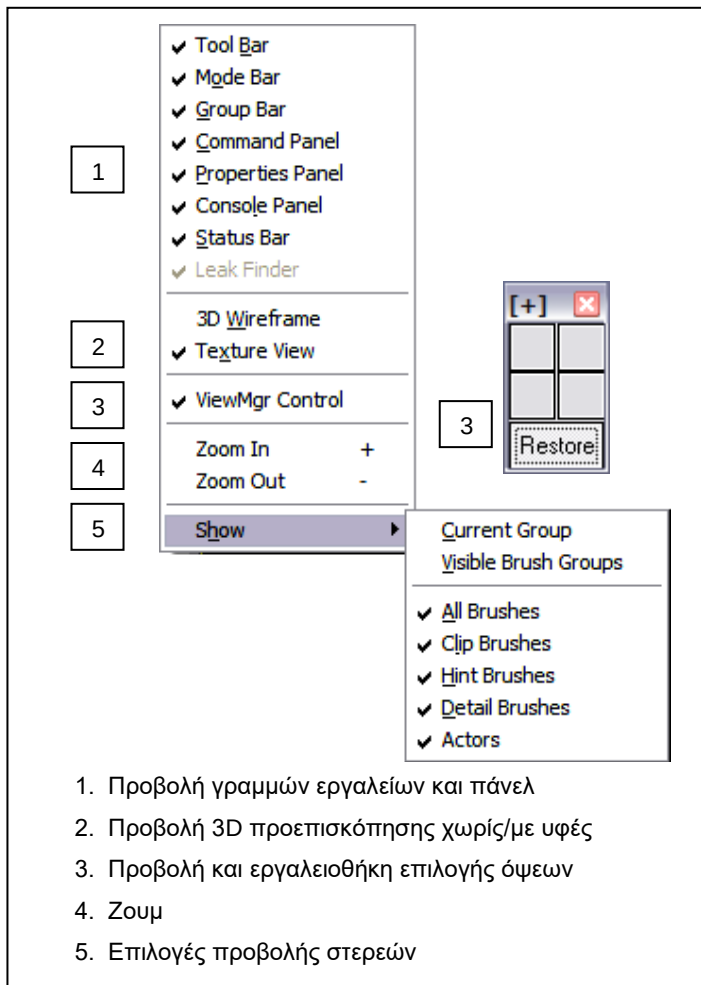


1. Φάκελος όπου βρίσκεται το αρχείο περιγραφής οντοτήτων
2. Φάκελος όπου βρίσκεται το πακέτο υφών και το όνομά του
3. Φάκελος όπου βρίσκεται το αρχείο περιγραφής των πιονιών και το όνομά του
4. Φάκελος όπου βρίσκονται αποθηκευμένα τα τρισδιάστατα μοντέλα
5. Πρόσδεση στο πλέγμα κατά συγκεκριμένο αριθμό Texels
6. Πρόσδεση στο πλέγμα κατά συγκεκριμένο αριθμό μοιρών κατά την περιστροφή
7. Ορισμός κλίμακας του εικονικού κόσμου και η υπάρχουσα κλίμακα
8. Ορισμός κλίμακας φωτισμού και κλίμακας σχεδίασης των υφών
9. Καταχώριση-Ακύρωση ρυθμίσεων

Εικόνα 11.9 Καρτέλα ρυθμίσεων επιπέδου

11.1.4 Γραμμή εντολών Προβολή-View

Από τη γραμμή εντολών “Προβολή-View” μπορούμε να επιλέξουμε ποιες γραμμές εργαλείων και ποια πάνελ θα εμφανίζονται, αν στην τρισδιάστατη προεπισκόπηση τα αντικείμενα θα παρουσιάζονται με τις υφές τους ή με απλό πλέγμα γραμμών και αν θα είναι ορατή η εργαλειοθήκη επιλογής όψης προεπισκόπησης (εικόνα 11.10). Από την εργαλειοθήκη αυτή μπορούμε να επιλέξουμε ποια από τις τέσσερις προεπισκοπήσεις θα καταλαμβάνει όλη την οθόνη μας για καλύτερη επεξεργασία (το πλήκτρο Restore επαναφέρει την προβολή και των τεσσάρων όψεων). Τέλος, μπορούμε να επιλέξουμε ποια είδη στερεών θα είναι ορατά.



Εικόνα 11.10 Γραμμή εντολών προβολής

Μπορούμε να κάνουμε ζουμ και στις τέσσερις όψεις περνώντας το ποντίκι μας σε μία από αυτές και πιέζοντας το πλήκτρο + από το αριθμητικό πληκτρολόγιο για μεγέθυνση ή το πλήκτρο - από το αριθμητικό πληκτρολόγιο για σμίκρυνση. Εναλλακτικά, μπορούμε να χρησιμοποιούμε το μεσαίο ροδάκι του ποντικιού μας.

Η επιλογή Γρήγορη Προβολή-QuickViewer από τη γραμμή εντολών χρησιμοποιείται για να βλέπουμε σε προεπισκόπηση .bsp αρχεία και γενικά δε χρησιμοποιείται. Οι επιλογές στα Εργαλεία-Tools και στην Κατασκευή-Build θα αναλυθούν στη συνέχεια του κεφαλαίου.

11.1.5 Γραμμή εντολών Κατάσταση-Mode

Η γραμμή εντολών “Κατάσταση-Mode” περιέχει αρκετά σημαντικές επιλογές (εικόνα 11.11). Η Κάμερα-Camera είναι ένα χρήσιμο εργαλείο που μας επιτρέπει να μετακινούμαστε ελεύθερα στις τρεις όψεις του σχεδιασμού ενός επιπέδου (πλην της 3D). Κατ’ αυτό τον τρόπο μπορούμε να εστιάζουμε (κάνοντας ζουμ) σε όποιο σημείο θέλουμε. Το αντίστοιχο εικονίδιο στη γραμμή εργαλείων είναι το . Κάνουμε κλικ σ’ αυτό για να το επιλέξουμε και στη συνέχεια κάνουμε κλικ και σέρνουμε το ποντίκι μας σε όποια από τις τρεις όψεις επιθυμούμε. Η κάμερα απο-επιλέγεται πατώντας το πλήκτρο  με πράσινο χρώμα ή το  με μπλε χρώμα στη γραμμή εργαλείων.

Το εργαλείο Επιλογής Στερεών ή Οντοτήτων-Brush Select μας επιτρέπει να επιλέξουμε αντικείμενα ήδη τοποθετημένα στο επίπεδο, ώστε να μπορούμε να τροποποιήσουμε τις ιδιότητές τους, το μέγεθος, τη θέση τους, κτλ. Στη γραμμή εργαλείων αντιπροσωπεύεται από το πλήκτρο  με πράσινο χρώμα. Αν μάλιστα το αντικείμενο που επιλέγουμε είναι στερεό-brush, μπορούμε να το χειριστούμε είτε ως σύνολο είτε την κάθε του έδρα ξεχωριστά. Αυτό γίνεται με την επιλογή της Εναλλαγής Μεταξύ Στερεού/Εδρών-Toggle Brush/Face Mode ή πιο απλά πατώντας το πλήκτρο Page Down.

Για να επιλέξουμε περισσότερα του ενός αντικείμενα ή έδρες, κρατάμε πατημένο το πλήκτρο Ctrl (Control) και κάνουμε διαδοχικά κλικ στα αντικείμενα. Εναλλακτικά, σε μία από τις τρεις όψεις, μπορούμε να σύρουμε το ποντίκι μας ορίζοντας ένα νοητό τετράγωνο που περιβάλλει τα προς επιλογή αντικείμενα.

Κατά την εκκίνηση του RFEEditPro είναι προεπιλεγμένο το εργαλείο Επιλογής Προτύπων πριν αυτά τοποθετηθούν στο επίπεδο-Template Select. Το αντίστοιχο εικονίδιο από τη γραμμή εργαλείων είναι το  με μπλε χρώμα. Στο σημείο αυτό θα πρέπει να

διευκρινίσουμε ότι ένα στερεό μπορούμε να το χειριστούμε στο επίπεδό μας σε δύο διαφορετικές φάσεις:

1. Πριν ακόμα τοποθετηθεί οριστικά στο επίπεδο. Ο χειρισμός του γίνεται με επιλεγμένο το εργαλείο Επιλογής Προτύπων ( με μπλε χρώμα).
2. Αφού τοποθετηθεί οριστικά στο επίπεδο. Ο χειρισμός του γίνεται με γίνεται το εργαλείο Επιλογής Στερεών ή Οντοτήτων ( με πράσινο χρώμα).

Αυτή η δυνατότητα μάς επιτρέπει, για παράδειγμα, να κάνουμε μία αρχική προσαρμογή στις διαστάσεις ενός στερεού, να το τοποθετούμε στο επίπεδό μας και στη συνέχεια να μεταβάλλουμε και πάλι τις διαστάσεις του κατά την εξέλιξη της κατασκευής του επιπέδου μας.

1	<u>C</u> amera	C
2	<u>B</u> rush Select	B
	✓ Brush Attributes	
	Face Attributes	
3	Toggle Brush/Face Mode	PgDn
4	✓ <u>T</u> emplate Select	T
5	✓ <u>A</u> uto Rebuild BSP	

1. Ελεύθερη μετακίνηση σ' όλες τις όψεις πλην της 3D

2. Επιλογή στερεών ή οντοτήτων ήδη τοποθετημένων και επιλογή εδρών

3. Εναλλαγή μεταξύ επιλογής στερεού/εδρών

4. Επιλογή προτύπων πριν τοποθετηθούν οριστικά στο επίπεδο

5. Αυτόματη επανασχεδίαση του bsp αρχείου

Ε-

κόνα

11.11 Γραμμή εντολών κατάστασης

11.1.6 Γραμμή εντολών Μεταβολή-Change και γραμμή εντολών Άξονα-Axis

Από τη γραμμή εντολών Μεταβολή-Change μπορούμε να μετακινήσουμε, να περιστρέψουμε, να αλλάξουμε τις διαστάσεις και να παραμορφώσουμε τα στερεά πριν ή αφού τα τοποθετήσουμε στον κόσμο μας. Τα αντίστοιχα εικονίδια της γραμμής εργαλείων είναι: για τη μετακίνηση το , για την ελεύθερη περιστροφή το , για την αλλαγή των διαστάσεων το , για την παραμόρφωση το , για την περιστροφή κατά 45° το  και για το κεντράρισμα το  (εικόνα 11.12).

Στη γραμμή εντολών Μεταβολή συναντάμε επίσης και τις επιλογές Αλλαγής Κλίμακας Κόσμου και Αλλαγής Κλίμακας Υφών, που υπήρχαν στις Επιλογές Επιπέδου της γραμμής εντολών Project (Κεφάλαιο 11.1.3).

1	<u>M</u> ove	M	1. Μετακίνηση
2	<u>R</u> otate		2. Ελεύθερη περιστροφή
3	<u>R</u> otate <u>45</u>		3. Περιστροφή κατά 45°
4	<u>S</u> cale	L	4. Αλλαγή διαστάσεων/κλίμακας
5	<u>S</u> hear	R	5. Παραμόρφωση
6	<u>S</u> cale World		6. Αλλαγή κλίμακας κόσμου
7	<u>T</u> exture Scale		7. Αλλαγή κλίμακας υφών
8	Center Brush/Entity		8. Κεντράρισμα στερεού/οντότητας

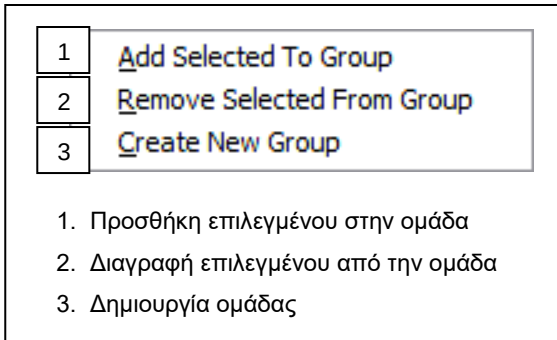
Εικόνα 11.12 Γραμμή εντολών μεταβολής

Στο RFEditPro υπάρχει η δυνατότητα να μεταβάλουμε τις διαστάσεις ή να παραμορφώσουμε ένα στερεό καθ' όλους ή μόνο κατά συγκεκριμένους άξονες. Αυτό γίνεται από τη γραμμή εντολών Άξονας-Axis, όπου μπορούμε να “κλειδώσουμε” έναν ή περισσότερους, ώστε οι όποιες μεταβολές να μην μπορούν να γίνουν σε αυτούς. Έτσι αποφεύγουμε, για παράδειγμα, την τυχαία αλλαγή του μεγέθους ενός στερεού κατά τον y άξονα. Τα αντίστοιχα εικονίδια από τη γραμμή εργαλείων είναι τα , ,  για τους άξονες x, y και z αντίστοιχα. Η προεπιλογή είναι η δυνατότητα μεταβολής όλων των αξόνων.

11.1.7 Γραμμή εντολών Ομάδα-Group

Από τη γραμμή εντολών Ομάδα-Group μπορούμε να δημιουργήσουμε ομάδες στερεών, να εντάξουμε ή να αφαιρέσουμε στερεά από μία ομάδα (εικόνα 11.13). Στη συνέχεια μπορούμε να εμφανίζουμε ή να αποκρύπτουμε ομάδες στερεών κατά την κατασκευή της εφαρμογής μας, κάτι ιδιαίτερα χρήσιμο όταν αυτή περιλαμβάνει μεγάλο αριθμό στερεών και η γεωμετρία του επιπέδου έχει γίνει περίπλοκη.

Τα εικονίδια στη γραμμή εργαλείων που μας επιτρέπουν τις παραπάνω εργασίες είναι το  για την προσθήκη ενός επιλεγμένου στερεού στην τρέχουσα ομάδα, το  για την αφαίρεση ενός επιλεγμένου στερεού από την τρέχουσα ομάδα και το  για να επιλέγουμε ποια ομάδα θα είναι η τρέχουσα ώστε να μπορούμε να προσθαφαιρούμε στερεά απ' αυτήν. Εννοείται βέβαια ότι θα πρέπει να έχουμε δημιουργήσει κάποια ομάδα από την επιλογή Δημιουργία Ομάδας-Create New Group (εικόνα 11.14).



Εικόνα 11.13 Γραμμή εντολών ομάδας

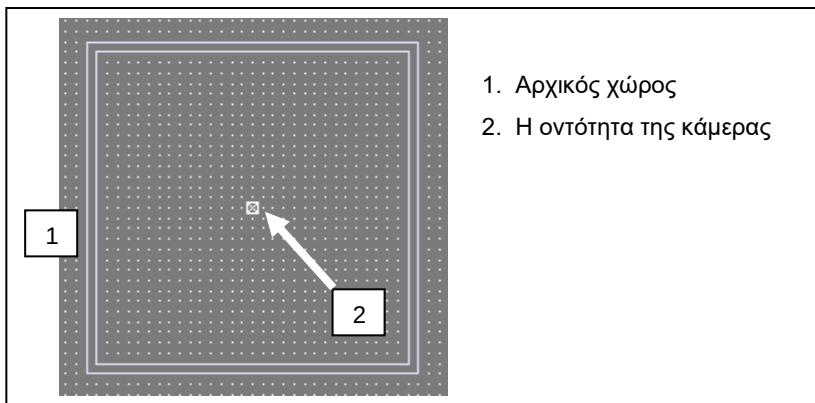


Εικόνα 11.14 Ορισμός ομάδας

Αφού εντάξουμε στερεά σε ομάδες, μπορούμε να επιλέγουμε αν θα είναι ορατές όλες οι ομάδες **All** (προεπιλεγμένο), οι ορατές από τη συγκεκριμένη γωνία θέασης **Vis** ή μόνο η τρέχουσα **Cur**. Μπορούμε επίσης να επιλέξουμε αν θα είναι ορατές και οι οντότητες με το πλήκτρο  (προεπιλεγμένο).

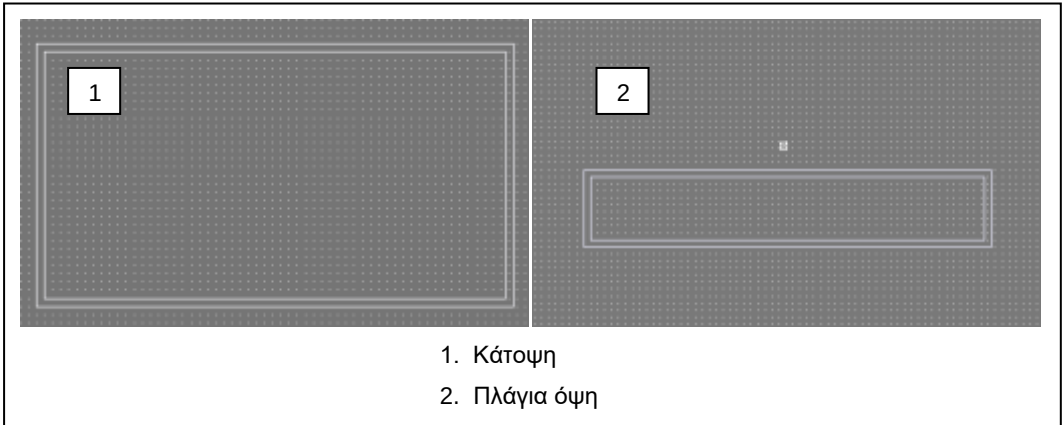
11.2 Κατασκευή και εκτέλεση ενός απλού επιπέδου

Κατά την εκκίνησή του το RFEditPro μας προτείνει να εισαγάγουμε στο επίπεδο έναν κύβο διαστάσεων 512x512x512 texels (εικόνα 11.15). Ο κύβος μάλιστα είναι κούφιος και παίζει το ρόλο ενός δωματίου (καταλαβαίνουμε ότι είναι κούφιος από τη διπλή μπλε γραμμή που ορίζει τις διαστάσεις του). Η μόνη οντότητα που περιλαμβάνεται στο επίπεδο είναι αυτή της κάμερας, που μας επιτρέπει να βλέπουμε τις τέσσερις προεπισκοπήσεις, και δεν πρέπει να συγχέεται με την κάμερα ελεύθερης μετακίνησης .



Εικόνα 11.15 Ο αρχικός χώρος

Με το εργαλείο αυξομείωσης διαστάσεων  κάνουμε το χώρο μεγαλύτερο σε έκταση αλλά και πιο μικρό σε ύψος, δουλεύοντας αρχικά στην κάτοψη και στη συνέχεια σε μία από τις πλάγιες όψεις (εικόνα 11.16).



Εικόνα 11.16 Μεταβολή των διαστάσεων του αρχικού χώρου

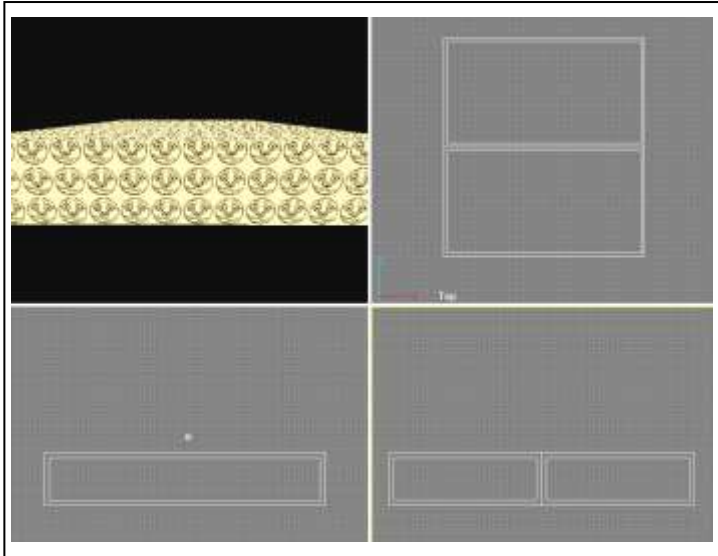
Μόλις οι διαστάσεις του “δωματίου” πάρουν το επιθυμητό μέγεθος, πιέζουμε το πλήκτρο Enter για να τοποθετηθεί το στερεό οριστικά στο χώρο του επιπέδου μας. Παρατηρούμε ότι ακούγεται ένας χαρακτηριστικός ήχος και στο παράθυρο 3D προεπισκόπησης ο κύβος, που αρχικά παριστανόταν μ' ένα απλό πλέγμα γραμμών, τώρα έχει αποκτήσει και μία υφή (πάντα την πρώτη στη σειρά από το τρέχον πακέτο υφών) (εικόνα 11.17).



Εικόνα 11.17 Η 3D προεπισκόπηση του δωματίου που μόλις τοποθετήθηκε

Παρατηρούμε ότι, παρόλο που εισαγάγαμε ένα “δωμάτιο”, αυτό εξακολουθεί να έχει μπλε περίγραμμα. Αυτό συμβαίνει για το λόγο ότι το RFEditPro μας προτείνει να εισαγάγουμε ένα δεύτερο ίδιο “δωμάτιο” στην ίδια ακριβώς θέση με το πρώτο. Επιλέγουμε το εργαλείο της μετακίνησης , τοποθετούμε το δεύτερο δωμάτιο δίπλα στο πρώτο

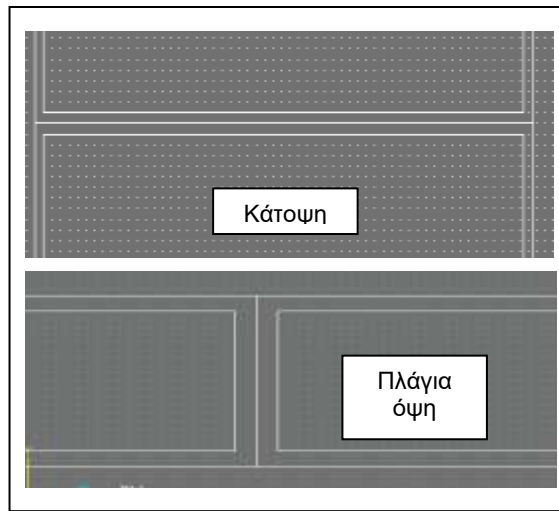
και πιέζουμε το πλήκτρο Enter ώστε κι αυτό να τοποθετηθεί οριστικά στο επίπεδο. Το τελικό αποτέλεσμα είναι να έχουμε δύο ίδια “δωμάτια” που εφάπτονται (εικόνα 11.18).



Εικόνα 11.18 Η τοποθέτηση και του δεύτερου δωματίου

Τα στερεά που δεν έχουν τοποθετηθεί οριστικά στο επίπεδο έχουν μπλε χρώμα και είναι ενεργοποιημένο το εργαλείο Επιλογής Προτύπων ( με μπλε χρώμα). Ένα στερεό τοποθετείται οριστικά στο επίπεδο πατώντας το πλήκτρο Enter. Τα στερεά που έχουν τοποθετηθεί οριστικά στο επίπεδο επιλέγονται ενεργοποιώντας το εργαλείο Επιλογής Στερεών ( με πράσινο χρώμα) και κάνοντάς τα κλικ με το αριστερό πλήκτρο του ποντικιού μας. Όταν επιλεγούν αποκτούν πράσινο χρώμα. Ένα από τα πολύ συχνά λάθη είναι η διαδοχική πίεση του πλήκτρου Enter, που έχει ως αποτέλεσμα την τοποθέτηση όμοιων στερεών το ένα μέσα στο άλλο, πράγμα που καθιστά σχεδόν αδύνατη την σωστή επεξεργασία τους.

Πρέπει να σημειώσουμε ότι είναι ιδιαίτερα σημαντικό το δύο δωμάτια να εφάπτονται και να μη μένει κενός χώρος μεταξύ τους, αλλά ούτε και να αλληλεπικαλύπτονται. Για να πετύχουμε την ακριβή τοποθέτηση, είναι απαραίτητο να κάνουμε ζουμ κινώντας το μεσαίο ροδάκι του ποντικιού μας (εικόνα 11.19). Αν έχουμε ήδη τοποθετήσει το δεύτερο “δωμάτιο” και διαπιστώσουμε ότι δεν έχει τη σωστή θέση, επιλέγουμε το εργαλείο Επιλογής Στερεών ( με πράσινο χρώμα), κάνουμε κλικ σ' ένα από τα δύο δωμάτια, παίρνουμε το εργαλείο μετακίνησης  και τοποθετούμε το “δωμάτιο” στη σωστή θέση.



Εικόνα 11.19 Η σωστή τοποθέτηση των δύο δωματίων

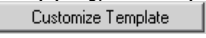
Εάν μεταξύ δύο “δωματίων” υπάρχει κενός χώρος, τότε ο εικονικός μας κόσμος δεν πρόκειται να λειτουργήσει. Αυτό συμβαίνει γιατί στα αρχεία .bsp είναι απαραίτητη η προϋπόθεση να μην υπάρχουν σημεία “διαρροής”, οι κόσμοι πρέπει να είναι στεγανοί χωρίς ασυνεχείς γραμμές. Εάν υπάρχει επικάλυψη των τοίχων των δύο “δωματίων”, ο εικονικός μας κόσμος θα λειτουργήσει, αλλά στο σημείο επικάλυψης θα υπάρχει “ανακάτεμα” των υφών.

Το επόμενο μας βήμα είναι να αναθέσουμε δύο διαφορετικές υφές στα “δωμάτιά” μας. Με το εργαλείο Επιλογής Στερεών (με πράσινο χρώμα) κάνουμε κλικ σ' ένα “δωμάτιο” το οποίο και αποκτά πράσινο περίγραμμα, ένδειξη ότι έχει επιλεγεί. Από το πάνελ εντολών (αριστερά στην οθόνη μας) επιλέγουμε την καρτέλα Υφές-Textures (εικόνα 11.20). Κάνουμε κλικ σε μία υφή που παριστάνει πέτρες ή τούβλα και πιέζουμε το πλήκτρο ανάθεσης .

Εναλλακτικά, μπορούμε να κάνουμε διπλό κλικ στην υφή. Επαναλαμβάνουμε τη διαδικασία επιλέγοντας το δεύτερο “δωμάτιο” και αναθέτοντας μία διαφορετική υφή. Στην περίπτωση που στην 3D επισκόπηση δε φαίνονται οι υφές που αναθέτουμε, πιέζουμε το πλήκτρο της ανανέωσης  από τη γραμμή εργαλείων.

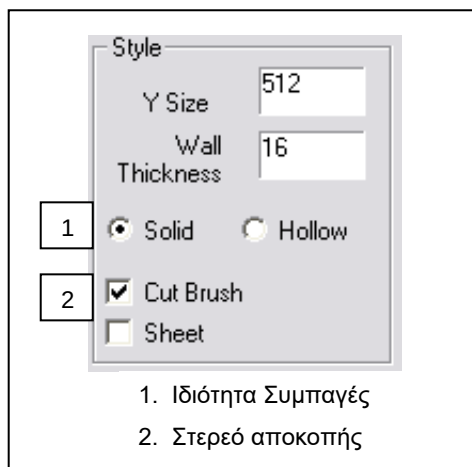


Εικόνα 11.20 Ανάθεση υφής από το πάνελ εντολών

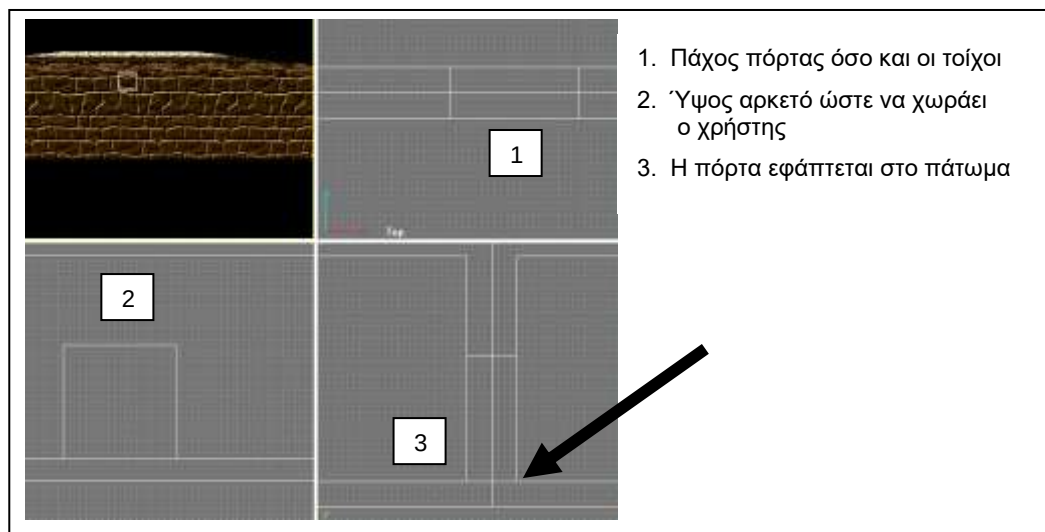
Στη συνέχεια θα δημιουργήσουμε μία “πόρτα” που θα ενώσει τα δύο “δωμάτια”. Κάνουμε κλικ στο εργαλείο  με μπλε χρώμα και παρατηρούμε ότι το RFEditPro μας προτείνει και πάλι τη δημιουργία ενός “δωματίου” με διαστάσεις ίσες με το τελευταίο που τοποθετήσαμε. Δε χρειαζόμαστε όμως κάτι τέτοιο. Για το λόγο αυτό, από το πάνελ ιδιοτήτων (δεξιά στην οθόνη μας) κάνουμε κλικ στο πλήκτρο Προσαρμογής Προτύπου-Customize Template  και επιλέγουμε τις ιδιότητες Συμπαγές-Solid (το αντίθετο είναι η ιδιότητα Κούφιο-Hollow) και Στερεό Αποκοπής-Cut Brush (εικόνα 11.21).

Με αυτό τον τρόπο, το στερεό θα μπορέσει να “αφαιρέσει” τους τοίχους εκεί που θα το τοποθετήσουμε. Μειώνουμε το μέγεθός του με το εργαλείο , έτσι ώστε να έχει πάχος ακριβώς όσο οι δύο τοίχοι, και το μετακινούμε με το εργαλείο , έτσι ώστε να ακουμπά στο πάτωμα, και πατάμε Enter για να τοποθετηθεί οριστικά (εικόνα 11.22).

Όταν τοποθετηθεί η πόρτα, το στερεό που την αναπαριστά αποκτά κόκκινο περίγραμμα, ένδειξη ότι είναι στερεό αποκοπής. Μπορούμε μάλιστα να πλοηγηθούμε στη 3D προεπισκόπηση για να διαπιστώσουμε ότι όντως άνοιξε μία τρύπα που ενώνει τα δύο “δωμάτια”.



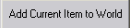
Εικόνα 11.21 Ιδιότητες συμπαγούς στερεού αποκοπής



Εικόνα 11.22 Οι διαστάσεις της πόρτας

Για να πλοηγηθούμε στη 3D προεπισκόπηση, κρατάμε πατημένο το αριστερό πλήκτρο του ποντικιού μας και σέρνουμε μπροστά, πίσω, δεξιά και αριστερά για κίνηση προς τις αντίστοιχες κατευθύνσεις. Με πατημένο το δεξιό πλήκτρο και σέρνοντας μπροστά ή πίσω, μπορούμε να μεταβάλουμε τη γωνία θέασης πάνω ή κάτω.

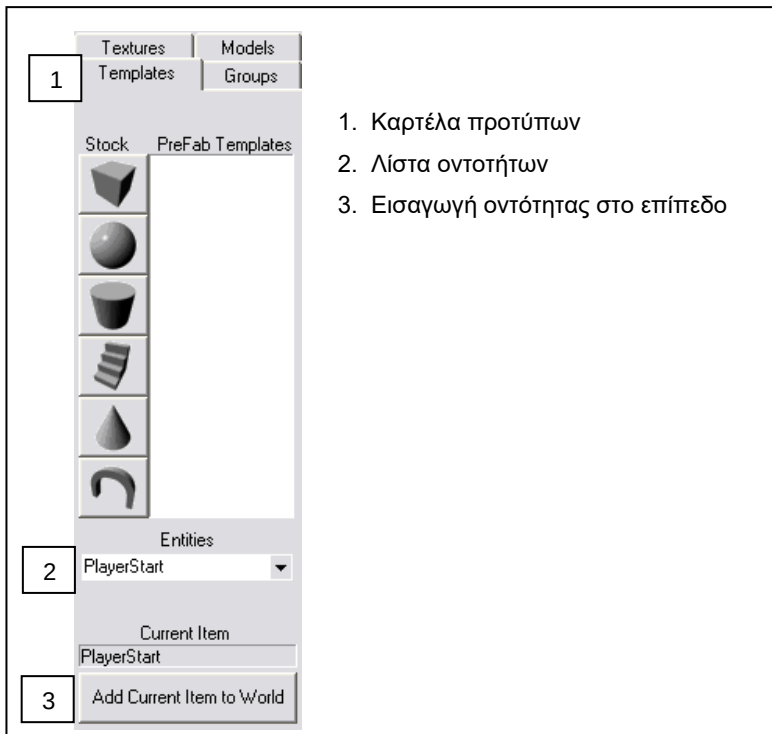
Έχοντας ολοκληρώσει την τοποθέτηση των βασικών γεωμετρικών στερεών που θα αποτελέσουν το επίπεδό μας, πρέπει να τοποθετήσουμε ένα τρισδιάστατο μοντέλο που θα αναπαριστά το χρήστη. Στο Reality Factory χρειάζεται να τοποθετήσουμε όχι μία αλλά δύο οντότητες επιφορτισμένες με τον έλεγχο του τρισδιάστατου μοντέλου, την PlayerSetup και την PlayerStart. Σε γενικές γραμμές, με την πρώτη μπορούμε να ορίσουμε παραμέτρους που αφορούν το μοντέλο και με τη δεύτερη παραμέτρους που αφορούν το σημείο εκκίνησής του.

Για να εισαγάγουμε αυτές τις δύο οντότητες, από το πάνελ εντολών επιλέγουμε την καρτέλα των Προτύπων-Templates (εικόνα 11.23), αναζητούμε την PlayerSetup από τη λίστα οντοτήτων και πατάμε το πλήκτρο εισαγωγής . Επαναλαμβάνουμε τη διαδικασία και για την PlayerStart. Να σημειώσουμε ότι όποτε εισάγουμε ίδιες οντότητες, το RFEEditPro τις δίνει και έναν αύξοντα αριθμό, π.χ. StaticMesh1, StaticMesh2, κτλ. Ειδικά όμως οι οντότητες PlayerSetup και PlayerStart πρέπει να είναι μοναδικές.

Όταν εισάγουμε οντότητες στο επίπεδό μας, πρέπει να προσέξουμε ιδιαίτερα ώστε αυτές να μην τοποθετηθούν εκτός του/των κύβων που ορίζουν το επίπεδό μας. Όπως και στην περίπτωση της ύπαρξης σημείων διαρροής, έτσι και στην περίπτωση των οντοτήτων εκτός του επιπέδου, η εφαρμογή δεν πρόκειται να λειτουργήσει.

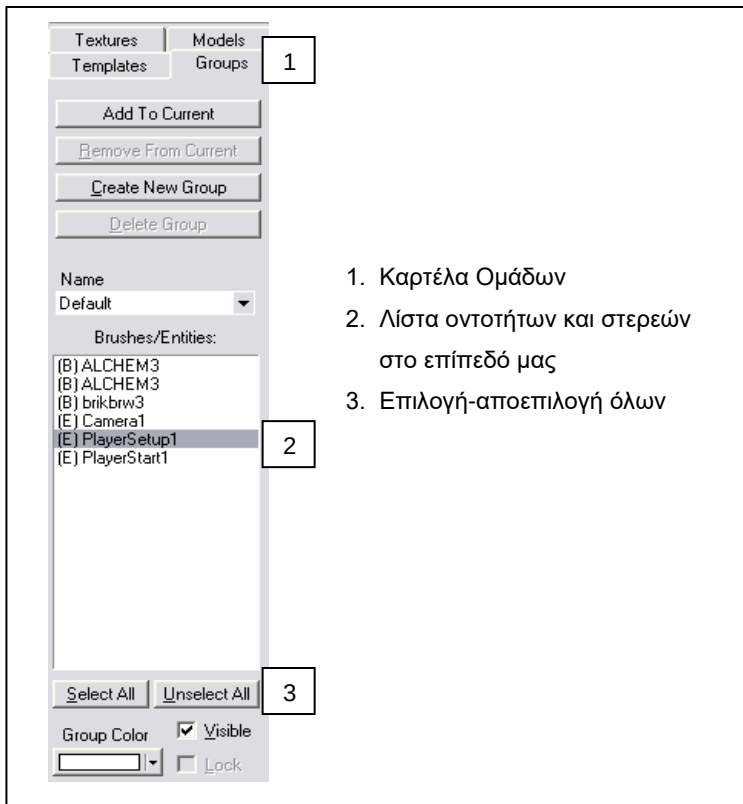
Η τοποθέτηση των οντοτήτων PlayerStart και PlayerSetup είναι υποχρεωτική. Χωρίς αυτές το επίπεδό μας δεν πρόκειται να εκτελεστεί από το Reality Factory.

Πριν προχωρήσουμε σε ρύθμιση των παραμέτρων των οντοτήτων PlayerSetup και την PlayerStart, καλό είναι να εντοπίσουμε την οντότητα PlayerStart, να την επιλέξουμε και να τη μετακινήσουμε κοντά στο πάτωμα με το εργαλείο . Αν η οντότητα αυτή είναι “ψηλά”, τότε ο χρήστης κατά την εκκίνηση του επιπέδου θα “πέφτει” από το αντίστοιχο ύψος.



Εικόνα 11.23 Εισαγωγή οντοτήτων στο επίπεδο

Ένας τρόπος επιλογής στερεών και οντοτήτων, ιδιαίτερα χρήσιμος στην περίπτωση περίπλοκων επιπέδων, είναι η επιλογή να γίνεται από το πάνελ εντολών και την καρτέλα Ομάδες-Groups (εικόνα 11.24). Παρατηρούμε ότι, όποτε επιλέγουμε ένα στοιχείο από τη λίστα, τότε αυτό επιλέγεται και στις τέσσερις προεπισκοπήσεις. Παράλληλα, στο πάνελ ιδιοτήτων φαίνονται οι ιδιότητες που μπορούν να παραμετροποιηθούν για το στοιχείο αυτό κάνοντας διπλό κλικ πάνω σε αυτές. Να σημειώσουμε ότι δε χρειάζονται να παραμετροποιούνται κάθε φορά όλες οι ιδιότητες, παρά μόνο αυτές που κατά περίπτωση μας ενδιαφέρουν.



Εικόνα 11.24 Επιλογή οντοτήτων και στερεών από την καρτέλα Ομάδες

Επιλέγοντας την οντότητα PlayerSetup1, οι ιδιότητες που φαίνονται στο πάνελ ιδιοτήτων είναι οι εξής:

Πεδίο	Περιγραφή
ActorName	Ένα όνομα που δίνουμε εμείς στο μοντέλο που αναπαριστά το χρήστη.
ActorRotation	Πόσες μοίρες χρειάζεται να περιστραφεί το τρισδιάστατο μοντέλο κατά τους άξονες x, y και z ώστε να τοποθετηθεί σωστά στο χώρο. Δε χρειάζεται να επέμβουμε στις τιμές 0 180 0.
ActorScaleFactor	Η κλίμακα του μοντέλου σε σχέση με το αρχικό του μέγεθος. Τιμές κάτω από 1 το κάνουν μικρότερο και τιμές μεγαλύτερες του 1 το κάνουν μεγαλύτερο.

Πεδίο	Περιγραφή
AttributelInfoFile	Από ποιο αρχείο αντλούνται πληροφορίες για επιπλέον χαρακτηριστικά του παίκτη/χρήστη. Προεπιλεγμένο είναι το αρχείο player.ini βρίσκεται στο φάκελο C:\RealityFactory\In-stall. Παρ' όλα αυτά δε χρειάζεται να δηλώσουμε όλη τη διαδρομή παρά μόνο το αρχείο που χρησιμοποιείται.
DeathFogColor	Όταν ο χρήστης “σκοτώνεται”, με ποιο χρώμα θα καλύπτεται η οθόνη. Με διπλό κλικ στην ιδιότητα εμφανίζεται ένα πάνελ χρωμάτων για να επιλέξουμε το χρώμα της αρεσκείας μας.
DeathFontSize	Το μέγεθος των γραμμάτων του μηνύματος που εμφανίζεται μετά το “θάνατο” του χρήστη.
Deathlevel	Ποιο επίπεδο (αρχείο με κατάληξη .bsp) θα “φορτώνεται” μετά το “θάνατο” του χρήστη.
DeathMessage	Ποιο μήνυμα θα εμφανίζεται μετά το “θάνατο” του χρήστη. Δυστυχώς, δεν μπορούν να εμφανιστούν ελληνικοί χαρακτήρες.
HeadBobbing	Υπάρχουν δύο τιμές: TRUE αν θέλουμε το κεφάλι του μοντέλου να κινείται καθώς περπατά (στην προβολή πρώτου προσώπου, και FALSE αν δε θέλουμε.
HeadBobLimit	Αριθμητική τιμή που προσδιορίζει πόσο θα κινείται το κεφάλι.
HeadBobSpeed	Αριθμητική τιμή που προσδιορίζει την ταχύτητα κίνησης του κεφαλιού.
HUDInfoFile	Το HUD (Heds Up Display) είναι πληροφορίες που εμφανίζονται στο χρήστη στο πάνω μέρος της οθόνης. Με τη ρύθμιση αυτή ορίζουμε ποιο αρχείο θα χρησιμοποιείται.
KeepAttrAtDeath	Αν δώσουμε την τιμή TRUE, τότε θα διατηρούνται στοιχεία, όπως για παράδειγμα το σκορ, και μετά το “θάνατο” του χρήστη.
LevelViewPoint	Η τιμή 0 υποδηλώνει προοπτική πρώτου προσώπου, η τιμή 1 προοπτική τρίτου προσώπου και η τιμή 2 ισομετρική προβολή.
LockView	Κατά την εκτέλεση της εφαρμογής μπορούμε να αλλάζουμε την προοπτική με τα πλήκτρα F1, F2 και F3. Αν στο πεδίο αυτό δώσουμε την τιμή TRUE, τότε δε θα είναι δυνατή αυτή η εναλλαγή και θα είναι διαθέσιμη μόνο η προοπτική που θα ορίσουμε στο πιο πάνω πεδίο.

Πεδίο	Περιγραφή
MovieMode	Αν δώσουμε την τιμή TRUE, τότε ο χρήστης δε θα μπορεί να κινήσει το χαρακτήρα που τον αντιπροσωπεύει και θα εκτυλίσσεται μία προκαθορισμένη από εμάς διαδρομή του χαρακτήρα.
Origin	Η θέση όπου τοποθετούμε την οντότητα PlayerSetup. Μεταβάλλεται αυτόματα όταν τη μετακινούμε.
OpeningCutScene	Το όνομα του αρχείου βίντεο με κατάληξη .avi ή εικόνας με κατάληξη .gif που θα παίζει ως εναρκτήρια σκηνή. Τα αρχεία αυτά πρέπει να είναι αποθηκευμένα στο φάκελο C:\RealityFactory\Media\Video.
PlayerScaleFactor	Η κλίμακα του χρήστη σε σχέση με το αρχικό του μέγεθος. Τιμές κάτω από 1 τον κάνουν μικρότερο και τιμές μεγαλύτερες του 1 τον κάνουν μεγαλύτερο. Η κλίμακα αυτή αναφέρεται στο μέγεθος του χρήστη σε σχέση με τα άλλα αντικείμενα του επιπέδου και όχι στο μέγεθος του τρισδιάστατου μοντέλου που ρυθμίζεται από την παράμετρο ActorScaleFactor.
ShadowAlpha	Τιμές από το 0 ως το 255, που καθορίζουν το βαθμό διαφάνειας της σκιάς που δημιουργεί το μοντέλο του χρήστη. Το 0 αντιπροσωπεύει απόλυτα διαφανή σκιά και το 255 την απόλυτα αδιαφανή.
ShadowAlphamap	Το αρχείο με κατάληξη .bmp, που χρησιμοποιείται για να κάνει τη σκιά διαφανή. Πρέπει να βρίσκεται αποθηκευμένο στο φάκελο C:\RealityFactory\Media\Bitmaps.
ShadowBitmap	Το αρχείο με κατάληξη .bmp, που χρησιμοποιείται για τη σκιά. Πρέπει να βρίσκεται αποθηκευμένο στο φάκελο C:\RealityFactory\Media\Bitmaps.
ShadowSize	Το μέγεθος της σκιάς. Η τιμή 0 αντιπροσωπεύει καθόλου σκιά.
UseDeathFog	Αν δώσουμε την τιμή TRUE, τότε θα εμφανίζεται μία ομίχλη που στροβιλίζεται κατά το "θάνατο" του χρήστη.
UseProjectedShadows	Αν δώσουμε την τιμή TRUE, τότε το μοντέλο του χρήστη θα δημιουργεί προβολική σκιά. Οι προβολικές σκιές δε χρειάζονται αρχεία εικόνων για να δημιουργηθούν.
UseStencilShadows	Αν δώσουμε την τιμή TRUE, τότε το μοντέλο του χρήστη θα δημιουργεί λεπτομερείς (Stencil) σκιές. Για να δημιουργηθούν Stencil σκιές, θα πρέπει να έχουμε ορίσει τα αρχεία σκιών στα πεδία ShadowAlphamap και ShadowBitmap.

Παρατηρούμε ότι αρκετές παράμετροι αφορούν γεγονότα και διεργασίες που θα συμβούν με το “θάνατο” του χρήστη. Όπως αναφέραμε στην ανάλυσή μας για τη χρήση παιχνιδιών βολών (5^ο κεφάλαιο), μπορούμε να παραλείψουμε τη χρήση αυτών των παραμέτρων με σκοπό να απαλείψουμε τα στοιχεία βίας.

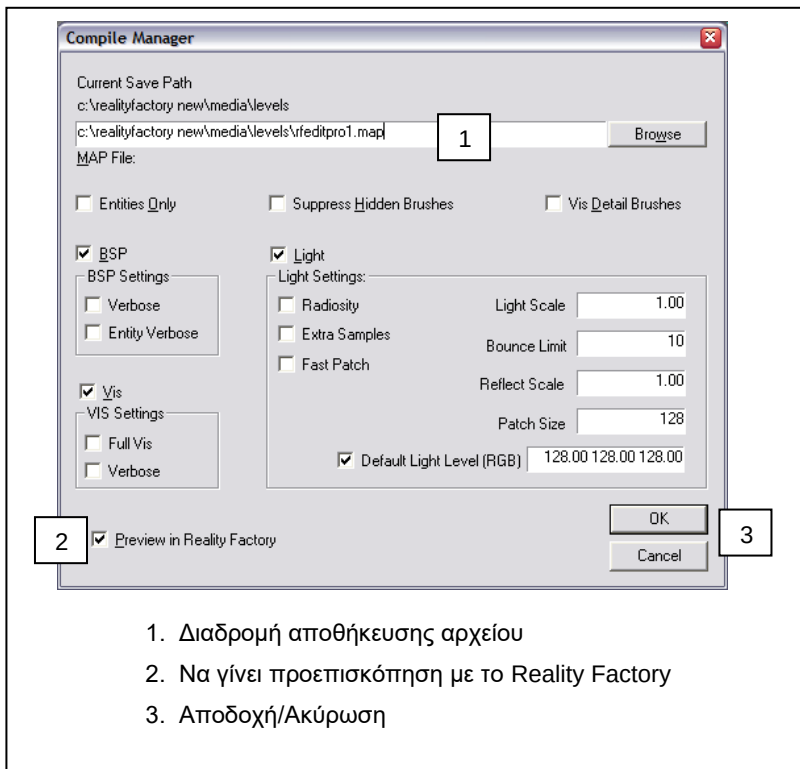
Οι ρυθμίσεις που μπορούν να γίνουν στην οντότητα PlayerStart είναι:

Πεδίο	Περιγραφή
Angle	Πού θα “κοιτάει” το μοντέλο του χρήστη κατά την έναρξη του επιπέδου. Οι τιμές αντιπροσωπεύουν μοίρες κατά τους x, y και z άξονες.
bSoundtrackLoops	Μπορούμε να συνδέσουμε στην οντότητα PlayerStart ένα μουσικό κομμάτι. Δίνοντας στο πεδίο αυτό την τιμή TRUE, το μουσικό κομμάτι θα παίζει αέναα.
Origin	Η θέση που τοποθετούμε την οντότητα PlayerStart. Μεταβάλλεται αυτόματα όταν τη μετακινούμε.
szCDTrack	Ο αριθμός του μουσικού κομματιού από CD που θα παίζει.
szEntityName	Το όνομα που δίνουμε εμείς στην οντότητα PlayerStart. Σε περίπτωση που η εφαρμογή μας έχει παραπάνω από ένα επίπεδα, αυτό το όνομα χρησιμοποιείται για να γνωρίζει η εφαρμογή από ποιο σημείο θα εκκινεί ο χρήστης με τη φόρτωση του τρέχοντος επιπέδου.
szMIDIFile	Ποιο μουσικό κομμάτι (αρχείο με κατάληξη .mid) θα εκτελείται. Το αρχείο πρέπει να είναι αποθηκευμένο στο φάκελο C:\RealityFactory\Media\Midi.
szStreamingAudio	Ποιο μουσικό κομμάτι –αρχείο με κατάληξη .wav, .mp3 ή .ogg– θα εκτελείται. Το αρχείο πρέπει να είναι αποθηκευμένο στο φάκελο C:\RealityFactory\Media\Audio.

Οι ιδιότητες όλων των οντοτήτων μεταβάλλονται κάνοντας διπλό κλικ πάνω σ' αυτές.

Έχοντας ολοκληρώσει την παραμετροποίηση των δύο οντοτήτων, το επόμενο μας βήμα είναι να “μεταγλωττίσουμε” την εφαρμογή, ώστε να μπορεί να εκτελεστεί από το Reality Factory. Αυτό γίνεται από την επιλογή Μεταγλώττιση-Compile του μενού Κατασκευή-Build της γραμμής εντολών ή πιο απλά από το πλήκτρο  της γραμμής εργαλείων.

Στην καρτέλα που ανοίγει (εικόνα 11.25), παρατηρούμε ότι μπορούμε να δώσουμε ένα όνομα αρχείου με την κατάληξη .map, που θα αποθηκευτεί στο φάκελο C:\Reality-Factory\Media\Levels\. Από το αρχείο αυτό θα δημιουργηθεί στη συνέχεια ένα άλλο αρχείο, με κατάληξη .bsp, που θα είναι το αρχείο της εφαρμογής μας και που θα “διαβάσει” το Reality Factory. Για να εκτελεστεί το επίπεδο από το Reality Factory, τσεκάρουμε την επιλογή Preview in Reality Factory. Οι υπόλοιπες επιλογές θα αναλυθούν σε επόμενο κεφάλαιο. Αν η διαδικασία της μεταγλώττισης δε βρει λάθη στο σχεδιασμό του επιπέδου μας, θα πάρουμε ένα μήνυμα ότι η διαδικασία ολοκληρώθηκε επιτυχώς και αν επιθυμούμε να εκτελεστεί το επίπεδο με το Reality Factory.



Εικόνα 11.25 Επιλογές στο μενού μεταγλώττισης

Κατά την εκτέλεση του επιπέδου παρατηρούμε ότι το μοντέλο που αναπαριστά το χρήστη κρατά ένα όπλο, το οποίο μάλιστα μπορούμε να αλλάξουμε με τα πλήκτρα των αριθμών 1-7 ή να το αφαιρέσουμε προσωρινά πατώντας το πλήκτρο H. Κάτι τέτοιο βέβαια δεν το επιθυμούμε αν η εφαρμογή μας δεν είναι παιχνίδι βολών. Μπορούμε όμως πολύ εύκολα να αφαιρέσουμε όλα τα όπλα με την αλλαγή μερικών παραμέ-

τρων στο αρχείο `player.ini`, που βρίσκεται στο φάκελο `C:\RealityFactory\Install`. Το αρχείο ανοίγει με το “Σημειωματάριο” κάνοντάς το διπλό κλικ.

Αναζητούμε προς το τέλος του αρχείου τις δηλώσεις των όπλων:

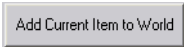
```
; Weapons  
[c8] (αυτό είναι το πρώτο από τα επτά όπλα)  
initial = 1  
low = 0  
high = 1
```

Αλλάζουμε την τιμή `initial` από 1 σε 0 και στα επτά όπλα και αποθηκεύουμε το αρχείο. Την επόμενη φορά που θα εκτελέσουμε την εφαρμογή μας (αλλά και κάθε άλλη) το μοντέλο που αναπαριστά το χρήστη δε θα κρατά κανένα όπλο. Στην αρχή του αρχείου υπάρχουν οι δηλώσεις για την υγεία, την αντοχή και το οξυγόνο του χρήστη:

[health]	[stamina]	[oxygen]
initial = 100	initial = 100	initial = 100
low = 0	low = 0	low = 0
high = 100	high = 100	high = 100

Μπορούμε να τις αλλάξουμε και αυτές κατά βούληση, μεταβάλλοντας την αρχική τιμή (`initial`), τη χαμηλή (`low`) και την υψηλότερη δυνατή (`high`).

Μία ενδιαφέρουσα οντότητα που μπορούμε να τοποθετήσουμε προαιρετικά κατά τα αρχικά στάδια κατασκευής του επιπέδου μας είναι η `EnvironmentSetup`. Η οντότητα μάς επιτρέπει να παρακάμψουμε ορισμένες βασικές προκαθορισμένες ρυθμίσεις του επιπέδου μας. Όπως και οι προηγούμενες οντότητες, για να την τοποθετήσουμε στο επίπεδό μας την αναζητούμε και την επιλέγουμε από τη λίστα οντοτήτων (πάνελ εντολών, καρτέλα των Προτύπων-`Templates` και πατάμε το πλήκτρο εισαγωγής



Για να δούμε τις ιδιότητές της την επιλέγουμε από το πάνελ εντολών και την καρτέλα Ομάδες-`Groups`. Οι ρυθμίσεις που μπορούν να γίνουν φαίνονται στα δεξιά της οθόνης μας στο πάνελ ιδιοτήτων και αναλυτικά είναι:

Πεδίο	Περιγραφή
AudibleRadius	Η ακτίνα μέσα στην οποία θα ακούγονται όλοι οι ήχοι.
DistanceFogColor	Στο Reality Factory μπορούμε να ορίσουμε μία απόσταση πέρα από την οποία το πρόγραμμα δε θα σχεδιάζει τα αντικείμενα, με σκοπό να “ελαφρύνουμε” την εφαρμογή. Για να μην είναι αντιαισθητικό το αποτέλεσμα, μπορούμε να εμφανίζουμε μία “ομίχλη” που καλύπτει τα αντικείμενα. Το πεδίο αυτό μάς επιτρέπει τον καθορισμό του χρώματος της “ομίχλης”. Με διπλό κλικ ανοίγει πάνελ επιλογής χρωμάτων.
EnableDistanceFog	Αν δώσουμε την τιμή TRUE, τότε θα ενεργοποιηθεί το εφέ της ομίχλης που περιγράψαμε πιο πάνω.
FarClipPlaneDistHigh	Η μέγιστη απόσταση σε texels πέρα από την οποία το Reality Factory δε θα σχεδιάζει τίποτα.
FarClipPlaneDistLow	Η ελάχιστη απόσταση σε texels πέρα από την οποία το Reality Factory δε θα σχεδιάζει τίποτα.
FogStartDistHigh	Η μέγιστη απόσταση σε texels πέρα από την οποία θα αρχίζει το εφέ της ομίχλης.
FogStartDistLow	Η ελάχιστη απόσταση σε texels πέρα από την οποία θα αρχίζει το εφέ της ομίχλης. Οι παράμετροι FogStartDist και FarClipPlaneDist παίρνουν τις τιμές min και max ανάλογα με τη θέση της μπάρας ρύθμισης της ποιότητας της εικόνας (κεφάλαιο 8.4.1).
Gravity	Η δύναμη της βαρύτητας που θα επικρατεί στο επίπεδό μας. Η τιμή 0 συμβολίζει τη χρήση της προεπιλεγμένης (-4.6). Μπορούμε να δώσουμε <u>μόνο</u> αρνητικές τιμές. Μία τιμή, για παράδειγμα -1.0, τη μειώνει σημαντικά, ενώ μία τιμή, για παράδειγμα -7.0, την αυξάνει σημαντικά.
JumpSpeed	Η ταχύτητα με την οποία ο χρήστης κάνει άλματα. Τιμές κάτω από το 0 θα έχουν ως αποτέλεσμα ο χρήστης να αναπηδά με συνεχώς αυξανόμενο ύψος.
LODAnimation	Το LOD (Level of Detail) είναι μία τεχνική αντικατάστασης των τρισδιάστατων μοντέλων-actors με όμοια μικρότερου αριθμού πολυγώνων όσο αυξάνει η απόσταση από το χρήστη (κεφάλαιο 6.7). Αν δώσουμε την τιμή TRUE, τότε τα μοντέλα αυτά θα έχουν τα δικά τους animations.

Πεδίο	Περιγραφή
LODdistance1	Η απόσταση σε texels στην οποία θα εμφανίζεται το πρώτο (το πιο κοντινό) από τη σειρά των μοντέλων που θα χρησιμοποιήσουμε για την υλοποίηση της τεχνικής LOD. Τα μοντέλα μας θα πρέπει να ακολουθούν μία συγκεκριμένη ονοματολογία. Αν, για παράδειγμα, έχουμε ένα τρισδιάστατο μοντέλο με το όνομα soldier.act, το πρώτο μοντέλο θα πρέπει να το έχουμε ονομάσει soldier_LOD1.act.
LODdistance2	Η απόσταση σε texels πέρα από την οποία θα εμφανίζεται το δεύτερο από τη σειρά των μοντέλων που θα χρησιμοποιήσουμε για την υλοποίηση της τεχνικής LOD. Για το παράδειγμά μας, θα πρέπει να ονομάζεται soldier_LOD2.act.
LODdistance3	Η απόσταση σε texels πέρα από την οποία θα εμφανίζεται το τρίτο μοντέλο της τεχνικής LOD.
LODdistance4	Η απόσταση σε texels πέρα από την οποία θα εμφανίζεται το τέταρτο μοντέλο της τεχνικής LOD.
LODdistance5	Η απόσταση σε texels πέρα από την οποία θα εμφανίζεται το πέμπτο από τη σειρά των μοντέλων της τεχνικής LOD. Δεν είναι όμως απαραίτητο να χρησιμοποιούμε πέντε μοντέλα.
MinShakeDist	Στο Reality Factory μπορούμε να εισαγάγουμε το εφέ της έκρηξης, η οποία μάλιστα μπορεί και “ταρακουνά” το χρήστη. Στο πεδίο αυτό ορίζουμε την απόσταση σε texels μέχρι την οποία η δόνηση από την έκρηξη είναι η μέγιστη.
Origin	Η θέση που τοποθετούμε την οντότητα. Μεταβάλλεται αυτόματα όταν τη μετακινούμε.
RealJumping	Αν δώσουμε την τιμή TRUE, τότε όταν ο χρήστης πηδά (πατώντας το πλήκτρο που έχουμε ορίσει γι' αυτή την ενέργεια), αυτό γίνεται με πιο ρεαλιστικό τρόπο.
SlideSlope	Η ελάχιστη κλίση που απαιτείται να έχει μία επιφάνεια ώστε ο χρήστης να “γλιστρά” προς τα πίσω. Οι τιμές κυμαίνονται από το 1 για οριζόντια επιφάνεια ως το 0 για κάθετη. Μεγάλες τιμές (κοντά στο 1) αναγκάζουν το χρήστη να γλιστρά ακόμα και σε σχεδόν επίπεδες επιφάνειες.

Πεδίο	Περιγραφή
SlideSpeed	Η μέγιστη ταχύτητα με την οποία ο χρήστης θα “γλιστρά” προς τα πίσω.
Speed	Η ταχύτητα με την οποία κινείται ο χρήστης. Τιμές κοντά στο 0 έχουν ως αποτέλεσμα ο χρήστης να κινείται υπερβολικά αργά.
SshadowsAlpha	Ο βαθμός διαφάνειας (0-255) των Stencil σκιάσεων που δημιουργεί το μοντέλο του χρήστη.
SshadowsColor	Το χρώμα των Stencil σκιάσεων.
SshadowsMaxLightToUse	Ο μέγιστος αριθμός φώτων που μπορούν να δημιουργήσουν Stencil σκιές (0-8).
StepHeight	Το ύψος σε texels που μπορεί να ανέβει ο χρήστης μ' ένα βήμα.
TotalFogDistHigh	Η μέγιστη απόσταση, σε texels, από τη νοητή κάμερα όπου η ομίχλη γίνεται αδιαπέραστη (ο χρήστης δε βλέπει τίποτα πέρα απ' αυτό το σημείο).
TotalFogDistLow	Η ελάχιστη απόσταση σε texels από τη νοητή κάμερα όπου η ομίχλη γίνεται αδιαπέραστη.
UseFarClipPlane	Αν θέσουμε την τιμή TRUE, τότε ενεργοποιούνται οι επιλογές FarClipPlaneDistHigh και FarClipPlaneDistLow.

Συμπερασματικά, η οντότητα EnvironmentSetup μπορεί να φανεί εξαιρετικά χρήσιμη αν το επίπεδό μας είναι μεγάλο σε έκταση και θέλουμε να περιορίσουμε το υπολογιστικό φορτίο της κάρτας γραφικών (με τη χρήση της ομίχλης, της τεχνικής του Far Clip Plane και της τεχνικής του Level of Detail). Είναι επίσης χρήσιμη όταν θέλουμε να αναγκάσουμε το χρήστη να κινείται αργά ή γρήγορα, να μπορεί ή να μην μπορεί να ανέβει επιφάνειες με κλίση. Τέλος, με το χειρισμό της βαρύτητας μπορούμε να υλοποιήσουμε επίπεδα όπου το κυρίαρχο στοιχείο θα είναι η απουσία της ή η πολύ μικρή της τιμή.

Στο Reality Factory δεν είναι απαραίτητο να δίνουμε τιμές σε όλα τα πεδία των ιδιοτήτων των οντοτήτων. Μπορούμε να συμπληρώνουμε μόνο τα πεδία εκείνα που κατά περίπτωση μας ενδιαφέρουν και τα υπόλοιπα να τα αφήνουμε κενά ή αμετάβλητα.

Κεφάλαιο 12°

Προσθήκη και χειρισμός υφών
Ιδιότητες των γεωμετρικών στερεών

Στο 9^ο κεφάλαιο ασχοληθήκαμε με τη μεθοδολογία επεξεργασίας εικόνων που θα “ντύσουν” τρισδιάστατα αντικείμενα, που θα μετατραπούν δηλαδή σε υφές. Υφές όμως μπορούμε να εφαρμόσουμε και στα γεωμετρικά στερεά-brushes που απαρτίζουν το επίπεδό μας. Ήδη, στο προηγούμενο κεφάλαιο, δώσαμε υφές στα δύο δωμάτια που κατασκευάσαμε. Στο κεφάλαιο αυτό θα αναλύσουμε με μεγαλύτερη λεπτομέρεια τον χειρισμό των υφών καθώς επίσης και ορισμένες ενδιαφέρουσες ιδιότητες που μπορούμε να δώσουμε στα γεωμετρικά στερεά.

12.1 Υφές σε όλες ή σε ορισμένες έδρες ενός στερεού

Εκτελώντας με το Reality Factory το επίπεδο που κατασκευάσαμε, μπορεί να παρατηρήσαμε ότι οι πέτρες ή τα τούβλα που αναθέσαμε σαν υφές στα δωμάτια ήταν πολύ μεγάλα σε σχέση με το μέγεθος του μοντέλου που αναπαριστά το χρήστη. Αυτό μπορούμε να το διαπιστώσουμε πατώντας F2 κατά την εκτέλεση του επιπέδου, ώστε να μπούμε σε προβολή τρίτου προσώπου. Επίσης, τα δωμάτια ήταν αρκετά “μονότονα”, αφού τοίχοι, πάτωμα και ταβάνι ντύνονταν με την ίδια υφή. Ας διορθώσουμε αυτά τα προβλήματα.

Κάνουμε κλικ στο εργαλείο  με πράσινο χρώμα ώστε να μπορούμε να επεξεργαστούμε στερεά που έχουν ήδη τοποθετηθεί και επιλέγουμε ένα δωμάτιο κάνοντας κλικ σ' αυτό ή επιλέγοντάς το από τη λίστα εντολών και την καρτέλα των Ομάδων-Groups. Πατώντας το πλήκτρο Page Down, μπαίνουμε στην κατάσταση επεξεργασίας εδρών και το στερεό μας αποκτά μοβ περίγραμμα. Αυτό σημαίνει ότι μπορούμε να επεξεργαστούμε όλες τις έδρες μαζί ή καθεμία ξεχωριστά. Παράλληλα, παρατηρούμε ότι στα δεξιά της οθόνης μας ανοίγει το πάνελ ιδιοτήτων, το οποίο περιέχει αρκετές επιλογές (εικόνα 12.1).

Κατ' αρχάς μπορούμε να ορίσουμε ορισμένες πολύ βασικές ιδιότητες των εδρών, όπως για παράδειγμα:

1. Αν θα είναι πλήρως φωτισμένες (Full Bright) ανεξάρτητα από την ύπαρξη φώτων που έχουμε τοποθετήσει στο επίπεδο.
2. Το είδος των σκιών που δημιουργούνται σε αυτές (Flat, Gouraud).
3. Αν θα είναι καθρέφτες ή ουρανός (Mirror, Sky).
4. Αν θα εκπέμπουν φως και πόσο (Light, Intensity).

Μπορούμε επίσης να καθορίσουμε πόσο θα μετατοπιστεί η εικόνα της υφής πάνω στο στερεό (Texture Offset), με τι κλίμακα θα σχεδιαστεί η υφή (Draw Scale), την κλίμακα καθορισμού του φωτισμού της έδρας (Light Map Scale), πόσο θα περιστραφεί η υφή και αν η έδρα θα είναι διαφανής και πόσο.

1. Γενικές ιδιότητες

2. Επιλογές μετατόπισης υφής

3. Επιλογές κλίμακας υφής

4. Επιλογές κλίμακας φωτισμού

5. Επιλογές περιστροφής υφής

6. Κλίμακα αντανάκλασεων

7. Επιλογές οριζόντιας/κάθετης περιστροφής υφής

8. Διαφανής έδρα ή στερεό και τιμή διαφάνειας

9. Επαναφορά προεπιλογών

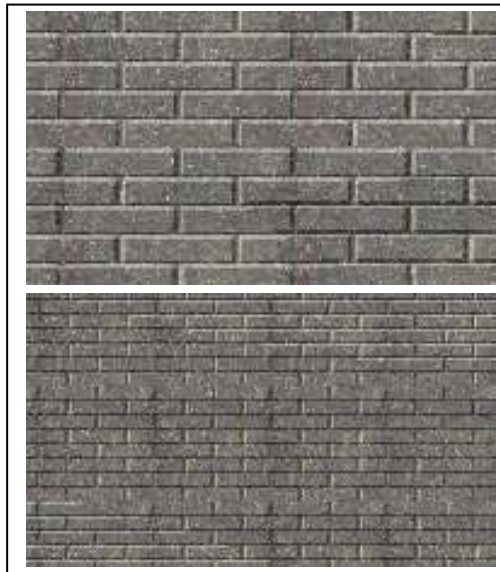
10. Εφαρμογή

11. Κατάσταση επεξεργασίας εδρών

Εικόνα 12.1 Οι επιλογές στο πάνελ ιδιοτήτων των εδρών

Οι επιλογές Texture Offset, Draw Scale και Light Map Scale συνοδεύονται από 2 ζεύγη πάνω και κάτω βέλους . Το πρώτο ζεύγος μάς επιτρέπει να αλλάζουμε τις τιμές κατά μία μονάδα, ενώ το δεύτερο κατά όσες μονάδες επιλέξουμε από το ακριβώς δίπλα αναδιπλούμενο μενού, π.χ. .

Ενώ είμαστε σε κατάσταση επεξεργασίας εδρών, επιλέγουμε τους τέσσερις τοίχους του δωματίου (κρατώντας πατημένο το πλήκτρο Ctrl και κάνοντας κλικ στον καθένα) και αναθέτουμε μία άλλη υφή (από το πάνελ εντολών, καρτέλα Υφές-Textures). Αν δε φαίνονται οι αλλαγές, κάνουμε κλικ στο εργαλείο ανανέωσης . Έχοντας ακόμα επιλεγμένους τους τοίχους, μεταβάλλουμε την κλίμακα της υφής (Draw Scale) τόσο στον άξονα x όσο και στον άξονα y, δίνοντας την τιμή 0.5. Αυτό έχει ως αποτέλεσμα η υφή να “πυκνώσει”, αφού οι διαστάσεις γίνονται οι μισές (εικόνα 12.2). Τιμές μεγαλύτερες του 1 έχουν ως αποτέλεσμα η υφή να “αραιώσει”, αφού οι διαστάσεις της μεγαλώνουν τόσες φορές όσο η τιμή που εισάγουμε.



Εικόνα 12.2 Πριν και μετά την αλλαγή της παραμέτρου Draw Scale

Επιλέγοντας την έδρα που ορίζει το ταβάνι και στη συνέχεια την έδρα που ορίζει το πάτωμα, αναθέτουμε διαφορετικές υφές και προσαρμόζουμε το μέγεθός τους όπως πιο πάνω. Επαναλαμβάνουμε τη διαδικασία και για το άλλο δωμάτιο. Εκτελούμε το επίπεδο στο Reality Factory κάνοντας κλικ στο εικονίδιο  για να δούμε αν το αποτέλεσμα μας ικανοποιεί.

Σ' ένα από τα δύο δωμάτια θα τοποθετήσουμε έναν πίνακα ζωγραφικής. Εννοείται βέβαια ότι έχουμε προσθέσει την ανάλογη εικόνα στο πακέτο υφών που χρησιμοποιούμε (9^ο κεφάλαιο). Εισάγουμε λοιπόν αρχικά ένα συμπαγή κύβο (πάνελ ιδιοτήτων, Προσαρμογή Προτύπου-Customize Template, Συμπαγές-Solid), μικραίνουμε τις διαστάσεις του με το εργαλείο  ώστε να πάρει τις διαστάσεις ενός πίνακα, τον μετακινούμε

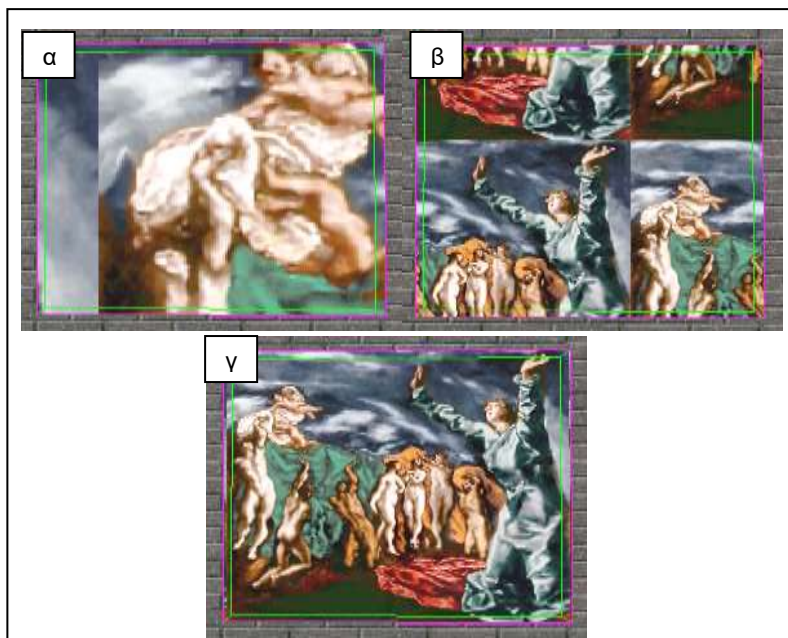
με το εργαλείο  για να τοποθετηθεί πάνω σ' έναν τοίχο και πατάμε Enter για να τοποθετηθεί οριστικά στο επίπεδο. Αν η θέση ή οι διαστάσεις δε μας ικανοποιούν, τις μεταβάλλουμε αφού πρώτα ενεργοποιήσουμε το εργαλείο  με πράσινο χρώμα.

Βεβαιωνόμαστε ότι ο “πίνακας” είναι επιλεγμένος και μπαίνουμε σε κατάσταση επεξεργασίας εδρών (Page Down). Επιλέγουμε την μπροστινή έδρα και της αναθέτουμε την υφή της εικόνας του πίνακά μας. Αν η υφή που τοποθετήσαμε δε φαίνεται, κάνουμε κλικ στο εργαλείο ανανέωσης . Με την πρώτη ματιά φαίνεται ότι η υφή είναι σε μέγεθος αρκετά μεγαλύτερη από τις διαστάσεις του πίνακά μας (εικόνα 12.3α) και για το λόγο αυτό θα την μικρύνουμε (πάνελ ιδιοτήτων, αλλαγή κλίμακας σχεδιασμού υφών-Draw Scale, *εικόνα 12.3β*). Παρ' όλα αυτά, η εικόνα φαίνεται να είναι έκκεντρα τοποθετημένη. Είναι λοιπόν απαραίτητο να χρησιμοποιήσουμε το εργαλείο μετατόπισης της εικόνας πάνω στο στερεό (Texture Offset) μετακινώντας την κατά τους άξονες x και y (*εικόνα 12.3γ*).

Μπορούμε να ορίσουμε ότι η επιφάνεια του πίνακά μας θα είναι πλήρως φωτισμένη (Full Bright) τσεκάροντας την αντίστοιχη επιλογή (*εικόνα 12.4*). Αυτό όμως θα έχει ως αποτέλεσμα η επιφάνεια να φωτίζεται έντονα ανεξάρτητα με την ύπαρξη –ή την απουσία– άλλων πηγών φωτός, οδηγώντας μερικές φορές σε ανεπιθύμητα αποτελέσματα.

Δεν υπάρχει κάποιος κανόνας για το πόσο πρέπει να μεταβάλλουμε τις τιμές στα εργαλεία Draw Scale και Texture Offset. Οι τιμές εξαρτώνται από το αρχικό μέγεθος της εικόνας και από το μέγεθος του στερεού που θα “ντύσει”. Χρειάζεται να κάνει κάποιος αρκετές δοκιμές μέχρι να βρει τις κατάλληλες. Στο Draw Scale μπορούμε να χρησιμοποιήσουμε δεκαδικούς αριθμούς, ενώ στο Texture Offset δεν μπορούμε.

Για να εντάξουμε νέες υφές στο τρέχον πακέτο υφών, πρέπει πρώτα να τερματίσουμε το RFEEditPro. Με την επόμενη εκκίνησή του θα έχουμε διαθέσιμες τις νέες υφές.



Εικόνα 12.3 Ακριβής τοποθέτηση υφής



Εικόνα 12.4 Υφή με και χωρίς πλήρη φωτισμό

12.2 Ιδιαίτερες ιδιότητες στερεών

Τα γεωμετρικά στερεά-Brushes του επιπέδου μας μπορούν να αποκτήσουν ιδιαίτερα χαρακτηριστικά που σχετίζονται με τις υφές που τα “ντύνουν”, αλλά και με τις ιδιότητες που τους δίνουμε ενώ είμαστε σε κατάσταση επεξεργασίας εδρών. Στο σημείο αυτό θα δούμε τις πιο χαρακτηριστικές.

12.2.1 Καθρέφτης

Θα τοποθετήσουμε ένα δεύτερο στερεό, ίσο σε διαστάσεις με τον πίνακα, πολύ απλά αντιγράφοντας και επικολλώντας σε άλλη θέση τον πίνακά μας. Επιλέγουμε το νέο στερεό και μπαίνουμε σε κατάσταση επεξεργασίας εδρών (με Page Down). Επιλέγουμε την μπροστινή έδρα και της αναθέτουμε μία μονόχρωμη υφή (δεν έχει σημασία το χρώμα). Από το πάνελ ιδιοτήτων κάνουμε κλικ στις ιδιότητες Καθρέφτης-Mirror και Διαφανές-Transparent. Δίνουμε μία πολύ χαμηλή Τιμή Διαφάνειας-Transparency Value, για παράδειγμα 1. Δεν ξεχνάμε να κάνουμε κλικ στο πλήκτρο εφαρμογής

Apply

Στο παράθυρο της τρισδιάστατης προεπισκόπησης δε φαίνεται καμία αλλαγή πέρα από την εφαρμογή της υφής. Παρ' όλα αυτά, αν εκτελέσουμε το επίπεδό μας (κάνοντας κλικ στο εικονίδιο ) , θα δούμε ότι το νέο μας στερεό λειτουργεί ως καθρέφτης (εικόνα 12.5). Μάλιστα, στην προοπτική τρίτου προσώπου (πατώντας το πλήκτρο F2) φαίνεται ολόκληρο το μοντέλο που αναπαριστά το χρήστη.



Εικόνα 12.5 Καθρέφτης

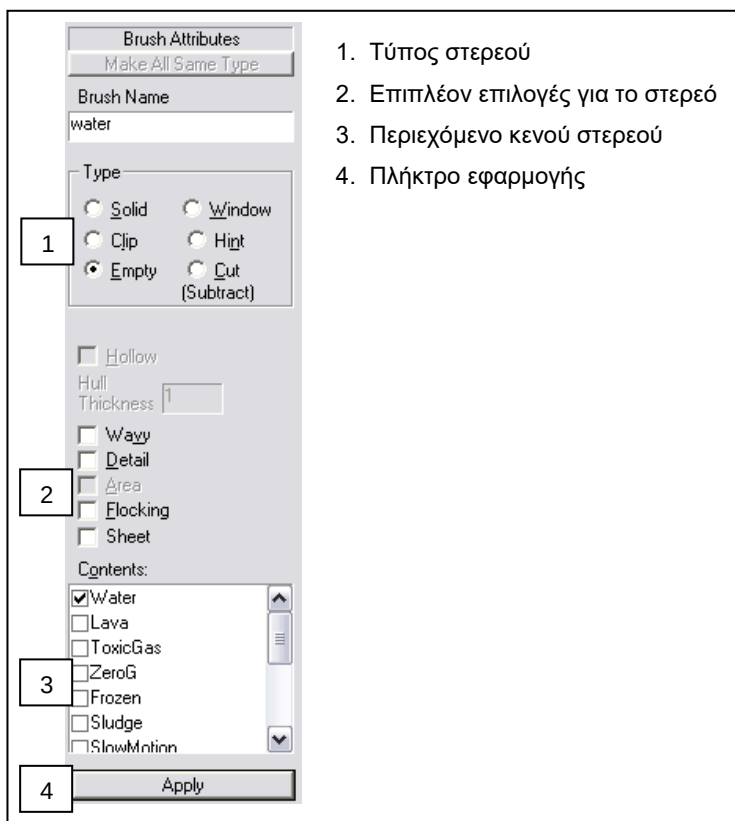
Για να φανεί το μοντέλο που αναπαριστά το χρήστη και στην προοπτική πρώτου προσώπου (πατώντας το πλήκτρο F1), πρέπει να επεμβούμε στο αρχείο PlayerSetup.ini, που βρίσκεται στο φάκελο C:\RealityFactory\Install. Το ανοίγουμε με το "Σημειωματάριο" και προς το τέλος του θα δούμε μία ομάδα παραμέτρων κάτω από την επικεφαλίδα [Misc]. Σε μία νέα σειρά πληκτρολογούμε τα εξής:

mirror1st = true

και αποθηκεύουμε το αρχείο.

12.2.2 Νερό, λάβα, πάγος και άλλα περιβάλλοντα

Ας υποθέσουμε ότι στην εφαρμογή μας θέλουμε να τοποθετήσουμε σημεία στα οποία θα υπάρχει νερό ή πάγος ή λάβα ή λάσπη. Εισάγουμε ένα γεωμετρικό στερεό που, ενώσω είμαστε στην κατάσταση επεξεργασίας στερεών πριν τοποθετηθούν στο επίπεδο (με μπλε χρώμα), ορίζουμε να είναι Συμπαγές-Solid (στο πάνελ ιδιοτήτων κάνουμε κλικ στο πλήκτρο **Customize Template** και στη συνέχεια κλικ στην ιδιότητα Συμπαγές-Solid). Αφού το τοποθετήσουμε, το επιλέγουμε (κάνοντας κλικ στο εργαλείο με πράσινο χρώμα και κλικ επάνω του), του δίνουμε την ιδιότητα Κενό-Empty (εικόνα 12.6) και κάνουμε κλικ στο πλήκτρο εφαρμογής **Apply**.



Εικόνα 12.6 Ιδιότητες στερεού

Τα “κενά” στερεά, ενώ επιτρέπουν στο χρήστη να περνά από μέσα τους, μπορούν ταυτόχρονα να μεταβάλλουν ορισμένες παραμέτρους που τον αφορούν, όπως για παράδειγμα τι ήχος θα ακούγεται όταν περπατά. Έτσι, αν επιλέξουμε το κενό στερεό να περιέχει νερό (κάνοντας κλικ στην ιδιότητα Νερό-Water), όταν εκτελέσουμε το επίπεδο μας θα ακούγεται ο χαρακτηριστικός ήχος των βημάτων σε νερό. Δεν πρέπει βέ-

βαια να ξεχάσουμε να “ντύσουμε” το στερεό με την κατάλληλη υφή νερού και να δώσουμε μία τιμή διαφάνειας περίπου 150-180 (σε κατάσταση επεξεργασίας εδρών με Page Down). Το Reality Factory περιέχει μία τέτοια υφή (water.bmp) στο φάκελο C:\RealityFactory\Media\Bitmaps, την οποία και μπορούμε να ενσωματώσουμε στο πακέτο υφών που χρησιμοποιούμε στην εφαρμογή μας.

Σε ορισμένες μάλιστα περιπτώσεις μπορούμε να κάνουμε επιλογή δύο ή παραπάνω ιδιοτήτων ταυτόχρονα. Για παράδειγμα, μπορούμε να επιλέξουμε το κενό στερεό να έχει τις ιδιότητες Παγωμένο-Frozen και Γρήγορη Κίνηση-Fast Motion. Εκτελώντας το επίπεδο θα παρατηρήσουμε ότι, όταν ο χρήστης μπει στην περιοχή του κενού στερεού, θα κινηθεί πιο γρήγορα. Δε θα ακούγεται όμως διαφορετικός ήχος βημάτων. Αυτό οφείλεται στο γεγονός ότι λείπει η ανάλογη δήλωση σ' ένα αρχείο παραμέτρων. Το αρχείο αυτό είναι το environment.ini, που βρίσκεται στο φάκελο C:\RealityFactory\Install. Το ανοίγουμε με διπλό κλικ (όπως και σ' όλες τις άλλες περιπτώσεις, ανοίγει με το “Σημειωματάριο”). Παρατηρούμε ότι, ενώ υπάρχουν δηλώσεις για τον ήχο των απλών βημάτων (default) και των βημάτων σε νερό, δεν υπάρχουν δηλώσεις αρχείων ήχου για όλα τα άλλα περιβάλλοντα:

[Default]	[Lava]
footsteps\footstep.wav	[ToxicGas]
[Water]	[ZeroG]
footsteps\w1.wav	[Frozen]
footsteps\w2.wav	
footsteps\w3.wav	

Μπορούμε να βρούμε κατάλληλους κατά περίπτωση ήχους (πρέπει να είναι τύπου wave, .wav), να τους αποθηκεύσουμε στο φάκελο C:\RealityFactory\Media\Audio και να κάνουμε τις ανάλογες δηλώσεις στο αρχείο αυτό. Να επισημάνουμε ότι οι ήχοι των βημάτων είναι μέσα σε υπο-φάκελο (footsteps) του φακέλου C:\RealityFactory\Media\Audio και για το λόγο αυτό η δήλωση των ήχων έχει τη μορφή footsteps\footstep.wav.

Αν σ' ένα κενό στερεό δώσουμε την ιδιότητα Σκάλα-Ladder, τότε μπορούμε να σκαρφαλώσουμε σ' αυτό. Είναι μία χρήσιμη ιδιότητα όταν, για παράδειγμα, έχουμε επίπεδα με δύο ή παραπάνω ορόφους και θέλουμε ο χρήστης να μπορεί να ανεβαίνει σ' αυτούς χωρίς να κάνουμε χρήση του γεωμετρικού στερεού των σκαλοπατιών.

12.2.3 Μεγαλύτερος έλεγχος περιβαλλόντων που περιέχουν υγρά

Στην προηγούμενη ενότητα μελετήσαμε τις δυνατότητες που μας δίνουν τα γεωμετρικά στερεά στα οποία έχουμε προσδώσει την ιδιότητα Κενό-Empty και παράλληλα ορίσαμε να περιέχουν κάτι υγρό (water, lava, sludge). Όμως τέτοια στερεά επηρεάζουν καθόλου ή πολύ λίγο το χρήστη. Τι γίνεται σ' εκείνη την περίπτωση που θέλουμε μεγαλύτερη αλληλεπίδραση των στερεών αυτών με το χρήστη; Χρειάζεται να καταφύγουμε στη χρήση της οντότητας που ονομάζεται Liquid. Πριν όμως την εισαγάγουμε στην εφαρμογή μας, πρέπει να προετοιμάσουμε το στερεό εκείνο που θα συνδεθεί με την παραπάνω οντότητα. Θα χρησιμοποιήσουμε ως παράδειγμα την κατασκευή μίας πισίνας γεμάτης με νερό.

Κατ' αρχάς, εισάγουμε σ' ένα από τα δωμάτιά μας ένα στερεό με την ιδιότητα Κούφιο-Hollow που θα παίζει το ρόλο της πισίνας, ακολουθώντας τα βήματα που περιγράφηκαν στις προηγούμενες ενότητες:

1. Από το πάνελ εντολών επιλέγουμε ένα Πρότυπο-Template (έναν κύβο).
2. Από το πάνελ ιδιοτήτων κάνουμε κλικ στο πλήκτρο .
3. Επιλέγουμε την ιδιότητα Κούφιο-Hollow.
4. Ενώσω είμαστε σε κατάσταση επεξεργασίας στερεών πριν τοποθετηθούν στο επίπεδο ( με μπλε χρώμα), διαμορφώνουμε τις διαστάσεις του και το τοποθετούμε στο χώρο.
5. Πατάμε Enter για να εισαχθεί.

Επειδή η πισίνα μας μέχρι στιγμής είναι στην ουσία ένα μικρό δωμάτιο και επειδή θέλουμε το πάνω μέρος της να είναι ανοιχτό, εισάγουμε ένα ακόμα στερεό με την ιδιότητα Στερεό-Solid, με διαστάσεις όμοιες με το “ταβάνι” της πισίνας και το τοποθετούμε ακριβώς στο πάνω μέρος της. Πατάμε Enter για να εισαχθεί, το επιλέγουμε και από το πάνελ ιδιοτήτων του προσδίδουμε την ιδιότητα Στερεού αποκοπής-Cut. Με αυτό τον τρόπο κάνουμε την πισίνα μας ανοιχτή. Σ' ένα από τα τοιχώματα της πισίνας και από την εξωτερική πλευρά, τοποθετούμε ένα στερεό με ύψος όσο και η πισίνα και στενό, που θα παίζει το ρόλο της σκάλας. Για το λόγο αυτό, αφού το τοποθετήσουμε, το επιλέγουμε και του δίνουμε τις ιδιότητες Κενό-Empty και Σκάλα-Ladder. Δεν ξεχνάμε να πιέσουμε το πλήκτρο εφαρμογής . Αντιγράφουμε και επικολλούμε το στερεό αυτό και στη μέσα πλευρά της πισίνας, ώστε να μπορεί ο χρήστης να βγει έξω.

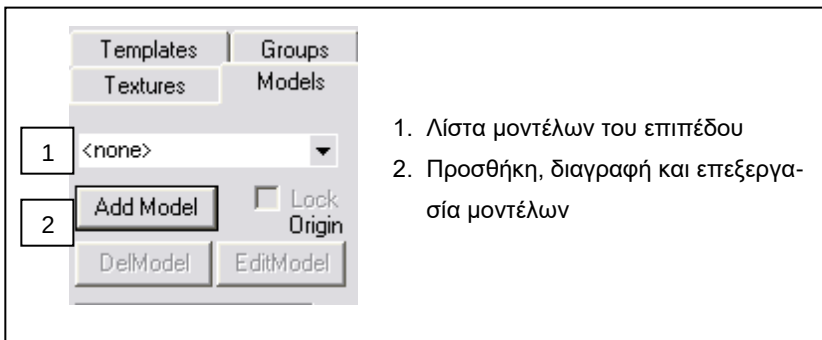
Μέχρι στιγμής η πισίνα μας δεν έχει νερό. Εισάγουμε λοιπόν ένα ακόμα στερεό με διαστάσεις ελάχιστα μικρότερες από το εσωτερικό της πισίνας (για να μην υπάρχει ανάμειξη υφών) και του δίνουμε τις ιδιότητες Κενό-Empty και Νερό-Water. Του δίνουμε την υφή νερού. Δεν ξεχνάμε να μπορούμε σε κατάσταση επεξεργασίας εδρών και με επιλεγμένες όλες τις έδρες να δώσουμε την ιδιότητα Διαφανές-Transparent και τιμή δια-

φάνειας 150-180. Και πάλι, για να εφαρμοστούν οι αλλαγές, πρέπει να πιέσουμε το πλήκτρο .

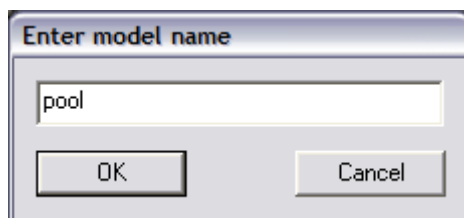
Αν εκτελέσουμε την εφαρμογή μας (κάνοντας κλικ στο εικονίδιο ) , θα διαπιστώσουμε ότι είναι τουλάχιστον παράδοξο ο χρήστης να μπορεί να περπατά ενώ είναι σκεπασμένος με νερό. Θα περιμέναμε να κολυμπά και ακόμα περισσότερο να μην μπορεί να μένει για πολλή ώρα μέσα σ' αυτό. Για το λόγο αυτό, θα χρειαστεί να εισαγάγουμε μία οντότητα που θα μπορέσει να προσδώσει στο νερό της πισίνας τις πραγματικές ιδιότητες του νερού. Σε γενικές γραμμές, στο Reality Factory μπορούμε να ελέγχουμε γεωμετρικά στερεά-brushes μέσω οντοτήτων. Τα βήματα που απαιτούνται είναι τα εξής:

1. Εισάγουμε και διαμορφώνουμε ένα στερεό.
2. Το μετατρέπουμε σε μοντέλο-model.
3. Εισάγουμε και παραμετροποιούμε μία οντότητα που συνδέεται με το μοντέλο και το ελέγχει.

Στην περίπτωση μας, το στερεό που μας ενδιαφέρει το έχουμε ήδη εισαγάγει (το νερό της πισίνας). Το επιλέγουμε και από το πάνελ εντολών επιλέγουμε την καρτέλα Μοντέλα-Models (εικόνα 12.7). Στο 16^ο κεφάλαιο θα παρουσιάσουμε τις δυνατότητες που έχουμε σ' αυτή την καρτέλα. Στο σημείο αυτό, το μόνο που χρειάζεται είναι να κάνουμε κλικ στο πλήκτρο εισαγωγής μοντέλου  και να δώσουμε ένα χαρακτηριστικό όνομα στο μοντέλο, για το παράδειγμά μας pool (εικόνα 12.8). Από τώρα και στο εξής, αυτό το γεωμετρικό στερεό είναι σε θέση να ελέγχεται από ειδικές οντότητες.



Εικόνα 12.7 Καρτέλα μοντέλων



Εικόνα 12.8 Εισαγωγή ονόματος μοντέλου

Κάθε γεωμετρικό στερεό, για να μπορεί να το χειριστεί μία οντότητα, χρειάζεται να το μετατρέψουμε πρώτα σε Μοντέλο-Model.

Όπως αναφέραμε, η οντότητα που μας ενδιαφέρει είναι η Liquid (υγρό). Την εισάγουμε από το πάνελ εντολών, καρτέλα Πρότυπα-Templates (δεν ξεχνάμε να πατήσουμε το πλήκτρο εισαγωγής Add Current Item to World) και την επιλέγουμε από το πάνελ εντολών, καρτέλα Ομάδες-Groups). Στα δεξιά της οθόνης μας, στο πάνελ ιδιοτήτων, ανοίγει η καρτέλα με τις ιδιότητες που μπορούμε να παραμετροποιήσουμε, που αναλυτικά είναι:

Πεδίο	Περιγραφή
Alphamap	Το όνομα του αρχείου εικόνας που μπορεί να δώσει διαφάνεια και μία ελαφριά απόχρωση στο υγρό στοιχείο. Στην περίπτωση του νερού, το Reality Factory περιέχει μία τέτοια εικόνα στο φάκελο C:\RealityFactory\Media\Bitmaps την a_water.bmp. Συνεπώς η δήλωση στο πεδίο αυτό έχει τη μορφή a_water.bmp.
Bitmap	Το όνομα του αρχείου εικόνας που έχει τη υφή του νερού. Όμοια με πιο πάνω, δηλώνουμε το αρχείο water.bmp. Αν θέλουμε το υγρό να είναι κάτι άλλο εκτός από νερό, θα πρέπει να βρούμε ανάλογες υφές και να τις αποθηκεύσουμε στο φάκελο C:\RealityFactory\Media\ Bitmaps.
DamageAltAttr	Στο πεδίο αυτό ορίζουμε ποια παράμετρος υγείας του χρήστη θα είναι η δευτερεύουσα. Για την περίπτωση του νερού δίνουμε την health.
DamageAltDelay	Η τιμή (σε δευτερόλεπτα) που θα καθυστερεί η μείωση της τιμής της δευτερεύουσας παραμέτρου υγείας.
DamageAmt	Πόσο θα μειώνεται η πρωτεύουσα παράμετρος υγείας του χρήστη σε χρόνο που θα ορίσουμε σε επόμενο πεδίο.
DamageAttr	Ποια παράμετρος υγείας θα είναι η πρωτεύουσα. Στην περίπτωση του νερού είναι η oxygen.

Πεδίο	Περιγραφή
DamageDelay	Η τιμή (σε δευτερόλεπτα) που θα καθυστερεί η μείωση της πρωτεύουσας παραμέτρου υγείας.
DamageIn	Αν δώσουμε την τιμή TRUE, τότε ο χρήστης θα βλάπτεται ακόμα και αν ακουμπά το υγρό (δε θα χρειάζεται να είναι ολόκληρος μέσα σ' αυτό).
GravityCoeff	Ποιο θα είναι το ποσοστό της αρχικής βαρύτητας όσο ο χρήστης είναι μέσα στο υγρό (οι τιμές μπορούν να είναι από 0 ως 100).
InLiquidSound	Ποιος ήχος θα ακούγεται όσο ο χρήστης περπατά στο υγρό. Για το νερό μπορούμε να κάνουμε τη δήλωση footsteps\w1.wav, που σημαίνει ότι υπάρχει το αρχείο w1.wav στο φάκελο C:\RealityFactory\Media\Audio\Footsteps.
Model	Είναι ίσως η πιο σημαντική δήλωση, γιατί στο πεδίο αυτό ορίζουμε ποιο μοντέλο ελέγχει η οντότητα liquid. Στο παράδειγμά μας το μοντέλο είναι το pool.
Origin	Η θέση που τοποθετούμε την οντότητα. Μεταβάλλεται αυτόματα όταν τη μετακινούμε.
SpeedCoeff	Το ποσοστό της αρχικής ταχύτητας κίνησης του χρήστη όσο είναι μέσα στο υγρό (οι τιμές μπορούν να είναι από 0 ως 100).
SurfaceSound	Ποιος ήχος θα ακούγεται όταν ο χρήστης μπαίνει μέσα στο υγρό. Μπορούμε να κάνουμε τη δήλωση footsteps\w2.wav.
SwimSound	Ποιος ήχος θα ακούγεται όσο ο χρήστης κολυμπά στο υγρό. Μπορούμε να κάνουμε τη δήλωση footsteps\w1.wav.
szEntityName	Ποιο θα είναι το όνομα της οντότητας liquid, εάν επιθυμούμε και αυτή να ελέγχεται με τη σειρά της από άλλη οντότητα. Θα πρέπει να επισημάνουμε ότι το όνομα στο πεδίο αυτό δεν έχει καμία σχέση με το όνομα που παίρνει η οντότητα όταν την εισάγουμε στο επίπεδο. Παρατηρούμε ότι όταν εισαγάγαμε την οντότητα liquid, αυτή ονομάστηκε αυτόματα liquid1 στο πάνελ ιδιοτήτων και στη λίστα επιλογής. Αν είχαμε και μία δεύτερη, θα είχε το όνομα liquid2, κ.ο.κ. Στο αυτό το πεδίο όμως πρέπει να δώσουμε ένα πιο χαρακτηριστικό όνομα, έτσι ώστε να θυμόμαστε με ποιο μοντέλο είναι συνδεδεμένη ή σε ποιο σημείο βρίσκεται ή τι εργασία εκτελεί. Βέβαια δεν απαγορεύεται να την ονομάσουμε liquid1 ή να μη δώσουμε καθόλου όνομα (αν δεν πρόκειται να συνδεθεί με άλλη οντότητα).
TintColor	Ποιο θα είναι το βασικό χρώμα (πέρα από την υφή) του υγρού. Με διπλό κλικ ανοίγει πάνελ επιλογής χρωμάτων.

Πεδίο	Περιγραφή
Transparency	Ο βαθμός διαφάνειας του παραπάνω χρώματος. Οι τιμές μπορεί να είναι από 0 (πλήρως διαφανές) ως 255 (πλήρως αδιαφανές).

Αν εκτελέσουμε την εφαρμογή μας (κάνοντας κλικ στο εικονίδιο ) και οδηγήσουμε το μοντέλο που αναπαριστά το χρήστη μέσα στην πισίνα, θα διαπιστώσουμε ότι αλλάζει το animation κίνησης από περπάτημα σε κολύμβηση (εικόνα 12.9). Η αλλαγή του animation γίνεται αυτόματα όταν το μοντέλο του χρήστη συναντήσει στερεό συνδεδεμένο με την οντότητα liquid και μόνο όταν το στερεό έχει τέτοιο ύψος ώστε να καλύπτει πλήρως το μοντέλο. Επιπλέον, παρατηρούμε ότι η παράμετρος υγείας “οξυγόνο” μειώνεται κατά μία μονάδα το δευτερόλεπτο όσο ο χρήστης βρίσκεται κάτω από την επιφάνεια του νερού.



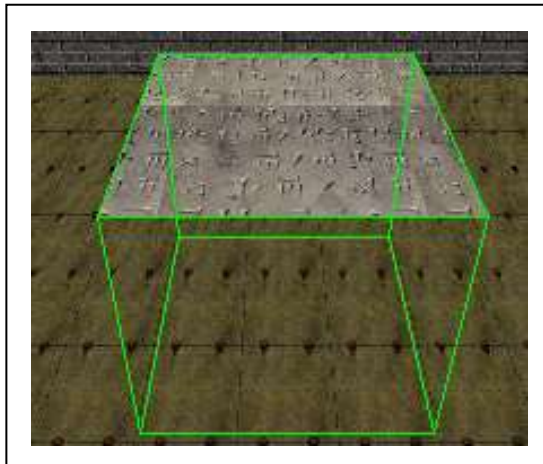
Εικόνα 12.9 Αλλαγή animation κίνησης και οι παράμετροι “υγείας” του χρήστη

Με τον ίδιο τρόπο που κατασκευάσαμε το νερό της πισίνας θα μπορούσαμε να κατασκευάσουμε ένα οποιοδήποτε άλλο είδος υγρού. Θα μπορούσαμε να του δώσουμε άλλη ιδιότητα, για παράδειγμα Λάβα-Lava, να συνδέαμε σ' αυτό άλλους ήχους και να ορίζαμε να βλάπτει σε μεγαλύτερο βαθμό την υγεία και όχι το οξυγόνο του χρήστη. Θα μπορούσαμε να αλλάξουμε και τον τρόπο που αλληλεπιδρά ο χρήστης με το υγρό μεταβάλλοντας την έλξη της βαρύτητας και την ταχύτητα κίνησης. Ένα τελευταίο σημείο που θα έπρεπε να προσέξουμε είναι η υφή που θα χρησιμοποιούσαμε και ο βαθμός διαφάνειάς της.

12.2.4 Άλλες ιδιότητες γεωμετρικών στερεών

Τα γεωμετρικά στερεά μπορούν να αποκτήσουν μερικές ακόμα ενδιαφέρουσες ιδιότητες που δίνονται από το πάνελ ιδιοτήτων, έχοντας βέβαια επιλεγμένο το στερεό που μας ενδιαφέρει:

1. Αν δώσουμε σ' ένα στερεό τις ιδιότητες Κενό-Empty και Κυματιστό-Wavy, τότε ο χρήστης μπορεί να μπει μέσα σ' αυτό (Empty), αλλά όλος ο εσωτερικός χώρος τρεμοπαίζει (Wavy).
2. Αν δώσουμε την ιδιότητα Φύλλο-Sheet, τότε μόνο η επάνω έδρα αποκτά υφή και υπολογίζεται στο Reality Factory. Όλες οι άλλες έδρες μένουν αόρατες στο χρήστη, παρότι δεν μπορεί να τις διαπεράσει (εικόνα 12.10). Η υφή της έδρας (που αποκτά κίτρινο περίγραμμα) είναι ορατή και από τις δύο πλευρές. Τέτοιου είδους στερεά είναι χρήσιμα για να χρησιμοποιηθούν ως παράθυρα, αν επιλέξουμε τη μοναδική έδρα που φαίνεται (πατώντας πρώτα Page Down, για να μπούμε σε κατάσταση επεξεργασίας εδρών) και την κάνουμε διαφανή.
3. Στερεά που δεν εμποδίζουν πλήρως το οπτικό πεδίο του χρήστη, όπως καλώδια και μικροαντικείμενα, καλό είναι να αποκτούν την ιδιότητα Λεπτομέρεια-Detail. Αυτό βοηθά την εφαρμογή να εκτελείται πιο ομαλά και με μεγαλύτερο αριθμό καρέ/δευτερόλεπτο. Να επισημάνουμε ότι αυτού του είδους τα στερεά πρέπει να τοποθετούνται σε απόσταση τουλάχιστον 1 texel από τα απλά στερεά, γιατί σε αντίθετη περίπτωση μπορεί να έχουμε προβλήματα στις σκιές ή να πέσει τελικά ο αριθμός των καρέ/δευτερόλεπτο.
4. Αν δώσουμε την ιδιότητα Clip, τότε το στερεό γίνεται αόρατο αλλά αδιαπέραστο, υπολογίζονται δηλαδή οι συγκρούσεις με το χρήστη. Κάτι τέτοιο είναι χρήσιμο εάν θέλουμε το στερεό αυτό να ενεργοποιεί γεγονότα στην εφαρμογή μας (περισσότερα για το θέμα αυτό σε επόμενα κεφάλαια).



Εικόνα 12.10 Η ιδιότητα Sheet σε στερεό

Κεφάλαιο 13^ο

Ουρανός και ορίζοντας

Μέχρι στιγμής τα παραδείγματα που παρουσιάσαμε αφορούσαν κλειστούς χώρους. Σε αυτούς ο χρήστης κινείται σε δωμάτια, διαδρόμους και γενικά σε χώρους με μικρή έκταση. Τέτοιου είδους εφαρμογές απαιτούν αρκετή επιδεξιότητα κατά την κατασκευή τους και μπορούν να αποκτήσουν σύνθετη γεωμετρία. Από την άλλη όμως πλευρά, οι κλειστοί χώροι επιβαρύνουν σε σχετικά μικρό βαθμό τον ηλεκτρονικό υπολογιστή του τελικού χρήστη εξαιτίας του γεγονότος ότι η μηχανή παιχνιδιού πρέπει να σχεδιάζει κάθε φορά μικρές εκτάσεις και ως εκ τούτου έχει μικρότερο υπολογιστικό φορτίο.

Υπάρχει όμως και μία άλλη κατηγορία εφαρμογών όπου υλοποιούνται ανοιχτοί χώροι. Η έκτασή τους εξαρτάται από τις δυνατότητες της μηχανής παιχνιδιών, αλλά σε κάθε περίπτωση το υπολογιστικό φορτίο αυξάνεται σημαντικά. Όπως είδαμε στο 11^ο κεφάλαιο, στο Reality Factory μπορούμε να περιορίσουμε το υπολογιστικό φορτίο μέσω της οντότητας EnvironmentSetup. Στους ανοιχτούς χώρους προκύπτει επίσης το επιπρόσθετο πρόβλημα ότι ο χρήστης θα πρέπει να βλέπει τον ουρανό αλλά και τον ορίζοντα, και μάλιστα να έχει την ψευδαίσθηση ότι αυτός είναι πολύ μακριά. Μπορούμε να υλοποιήσουμε τα παραπάνω στο Reality Factory με δύο διαφορετικούς τρόπους, με τη χρήση Skybox και με τη χρήση της οντότητας SkyDome.

13.1 Ουρανός και ορίζοντας με τη χρήση Skybox

Ο πρώτος τρόπος αφορά την απόδοση μίας συγκεκριμένης ιδιότητας στις έδρες ενός στερεού-brush και την ανάθεση των ανάλογων υφών. Όμως η ανάθεση των υφών στην περίπτωση του ουρανού δε γίνεται με τον τρόπο που παρουσιάσαμε στο προηγούμενο κεφάλαιο. Αυτό γιατί στον ουρανό δεν πρέπει να φαίνεται η παραμικρή “ραφή” μεταξύ των υφών και, για να δίνεται η ψευδαίσθηση της απόστασης, το μέγεθος των υφών δε θα πρέπει να μεταβάλλεται με την απόσταση.

Επιπλέον, δεν μπορούμε να χρησιμοποιήσουμε οποιεσδήποτε εικόνες για υφές ουρανού και ορίζοντα. Πέρα από την τυπική επεξεργασία που περιγράφηκε στο 9^ο κεφάλαιο, οι εικόνες μας πρέπει να ανήκουν στην κατηγορία εκείνη που ονομάζεται Skybox. Πρόκειται για πανοραμικές εικόνες ουρανού (360°) που έχουν διαμορφωθεί με τέτοιο τρόπο ώστε να σχηματίζουν έναν κύβο και που στη συνέχεια “κόβονται” σε έξι κομμάτια (πάνω, μπροστά, πίσω, δεξιά, αριστερά και κάτω-έδαφος). Μπορούμε να βρούμε αρκετά έτοιμα και δωρεάν Skyboxes στο Διαδίκτυο, για παράδειγμα στις διευθύνσεις:

<http://www.alusion-fr.com/an1ffa.htm>

<http://kell.leveldesign.org/q3skyboxes.html>

<http://www.scentednectar.com/skyboxes/index.htm>

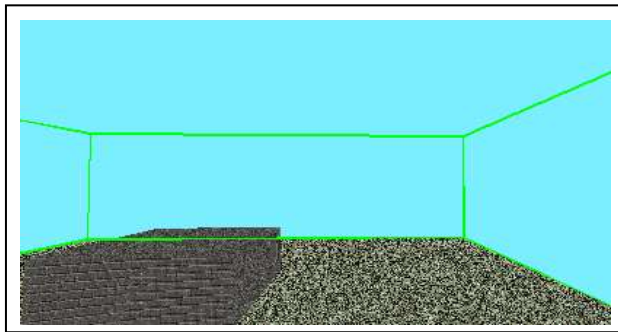
<http://www.redsorceress.com/skybox.html>

Αποθηκεύουμε στο σκληρό μας δίσκο και εντάσσουμε στο πακέτο των υφών που χρησιμοποιούμε τις έξι εικόνες που συγκροτούν το Skybox που μας ενδιαφέρει (αν και η κάτω εικόνα δε μας είναι πάντα απαραίτητη).

Το πρώτο μας βήμα είναι να εισαγάγουμε στην εφαρμογή μας ένα στερεό μεγάλων διαστάσεων με την ιδιότητα Κούφιο-Hollow που να περικλείει τα δύο δωμάτια που ήδη κατασκευάσαμε. Τοποθετούμε δηλαδή ένα πολύ μεγάλων διαστάσεων δωμάτιο. Αφού το εισαγάγουμε, το επιλέγουμε και μπαίνουμε σε κατάσταση επεξεργασίας εδρών πατώντας Page Down. Επιλέγουμε όλες τις εσωτερικές έδρες εκτός του πατώματος. Από το πάνελ ιδιοτήτων δίνουμε σ' αυτές την ιδιότητα Ουρανός-Sky. Δεν ξεχνάμε να κάνουμε κλικ στο πλήκτρο εφαρμογής . Στην κάτω έδρα δίνουμε μία υφή της επιλογής μας.

Αν η επιλογή των εδρών ήταν σωστή, στην τρισδιάστατη προεπισκόπηση οι έδρες που θα αποτελέσουν τον ορίζοντα και τον ουρανό θα γίνουν γαλάζιες (εικόνα 13.1). Αν δε φανεί κάποια αλλαγή, για να βεβαιωθούμε ότι όλα έγιναν σωστά, κάνουμε κλικ στο εργαλείο ανανέωσης .

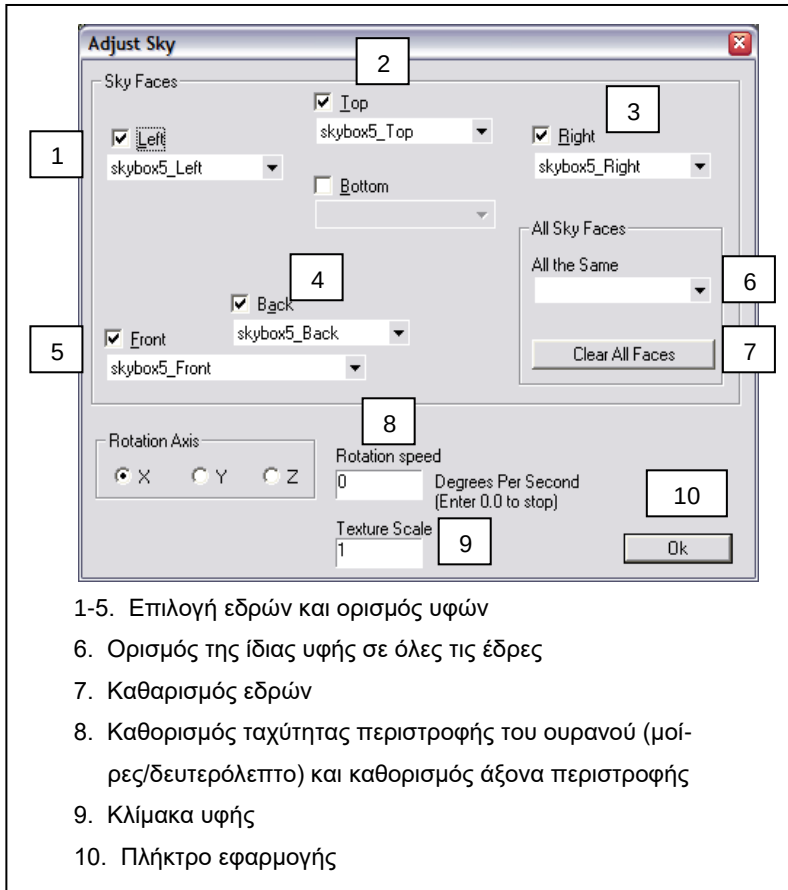
Δεν ξεχνάμε να ανοίξουμε μία πόρτα σε κάποιο από τα δωμάτια, ώστε ο χρήστης να μπορεί να βγει έξω. Εναλλακτικά μπορούμε να μεταφέρουμε την οντότητα PlayerStart εκτός των δωματίων.



Εικόνα 13.1 Έδρες με την ιδιότητα του ουρανού

Είναι πολύ σημαντικό να επιλέξουμε τις εσωτερικές και όχι τις εξωτερικές έδρες του στερεού που θα αποτελέσει τον ουρανό μας. Για το λόγο αυτό πρέπει να μετακινηθούμε στην τρισδιάστατη προεπισκόπηση στο εσωτερικό του στερεού-δωματίου. Αν επιλέξουμε εξωτερική έδρα, ο ουρανός δε θα λειτουργήσει.

Το επόμενο βήμα είναι να αναθέσουμε υφές που παριστάνουν ουρανό και ορίζοντα. Αυτό δε θα γίνει με το γνωστό τρόπο, αλλά από το πάνελ εντολών επιλέγουμε την καρτέλα Πρότυπα-Templates και κάνουμε κλικ στο πλήκτρο Προσαρμογής Ουρανού-Adjust Sky . Ανοίγει μία καρτέλα στην οποία ορίζουμε ποιες έδρες θα έχουν υφή ουρανού (τσεκάροντας τα αντίστοιχα τετραγωνάκια) και επιλέγοντας την ανάλογη εικόνα/υφή από το αναδιπλούμενο μενού (εικόνα 13.2). Πρέπει να προσέξουμε ποια εικόνα αναθέτουμε σε κάθε έδρα, γιατί σε αντίθετη περίπτωση δε θα έχουμε τα επιθυμητά αποτελέσματα (συνήθως οι εικόνες του Skybox έχουν ονόματα που μας διευκολύνουν). Όπως ήδη αναφέραμε, δε χρειάζεται να αναθέσουμε κάποια εικόνα στην κάτω έδρα. Μπορούμε να ορίσουμε ότι ο ουρανός μας εκτελεί περιστροφική κίνηση, δίνοντας την ταχύτητα περιστροφής στο ανάλογο πεδίο. Κάτι τέτοιο θα προϋπόθετε όμως μία ακόμα πιο ειδική κατηγορία Skybox και για το λόγο αυτό δίνουμε την τιμή 0 (καθόλου περιστροφή). Ελέγχουμε το αποτέλεσμα εκτελώντας την εφαρμογή (εικόνα 13.3).



Εικόνα 13.2 Ορισμός ιδιοτήτων ουρανού



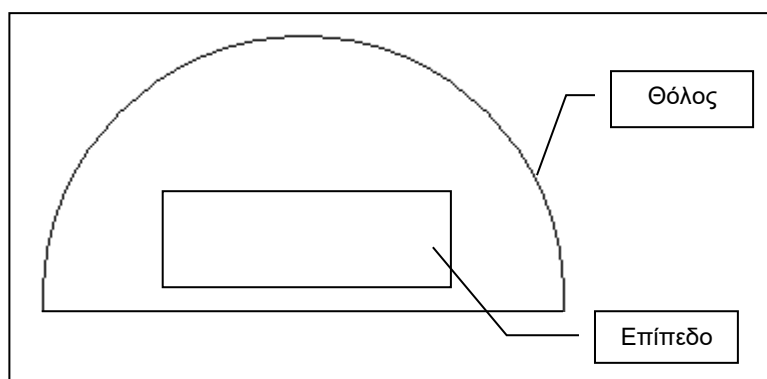
Εικόνα 13.3 Το τελικό αποτέλεσμα χρήσης Skybox

13.2 Ουρανός και ορίζοντας με τη χρήση SkyDome

Χρησιμοποιώντας εικόνες σ' ένα Skybox μπορούμε να πετύχουμε θεαματικά αποτελέσματα όσον αφορά τη ρεαλιστική απεικόνιση, όμως, επειδή χρησιμοποιούμε εικόνες, ο ουρανός είναι στατικός. Υπάρχει ήλιος και σύννεφα, αλλά δεν κινούνται. Αν επιθυμούμε ο ουρανός μας από στατικός να γίνει κινούμενος, και μάλιστα να γίνεται εναλλαγή ημέρας-νύχτας, χρειάζεται να καταφύγουμε σε μία άλλη τεχνική με τη χρήση της οντότητας του Ουράνιου θόλου-SkyDome.

Σε πρώτη φάση επιλέγουμε το εξωτερικό στερεό στις έδρες του οποίου είχαμε ορίσει τις υφές του ουρανού. Με page down μπαίνουμε στην κατάσταση επεξεργασίας εδρών και δίνουμε την ιδιότητα Ουρανός-Sky στην εσωτερική κάτω έδρα που είχαμε παραλείψει με την προηγούμενη τεχνική. Από το πάνελ εντολών και την καρτέλα των Προτύπων-Templates κάνουμε κλικ στο πλήκτρο Προσαρμογής Ουρανού-Adjust Sky  και στην καρτέλα που ανοίγει αφαιρούμε το τσεκάρισμα από όλα τα τετραγωνάκια.

Από το πάνελ εντολών και την καρτέλα Προτύπων-Templates επιλέγουμε και εισάγουμε στο επίπεδό μας την οντότητα SkyDome. Στην ουσία, η οντότητα αυτή δημιουργεί ένα θόλο που περιβάλλει το επίπεδό μας (σχήμα 13.1). Ο θόλος αυτός μπορεί να παραμετροποιηθεί με τη χρήση εντολών που βρίσκονται αποθηκευμένες σε αρχείο δέσμης εντολών (script file).



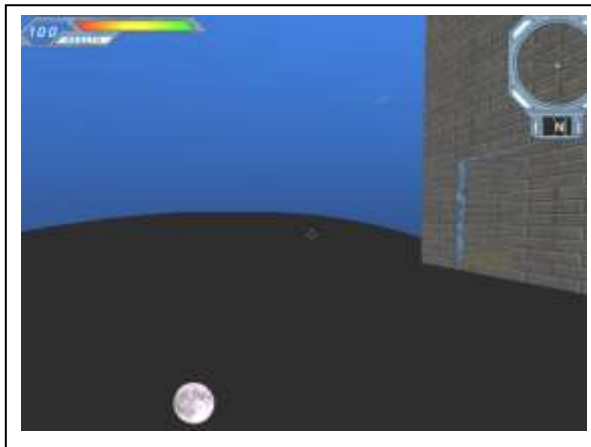
Σχήμα 13.1 Η λειτουργία της οντότητας Skydome

Επιλέγουμε τη SkyDome από την καρτέλα Ομάδες-Groups στο πάνελ εντολών και στο πάνελ ιδιοτήτων, στα δεξιά μας, εμφανίζονται οι ιδιότητές της, που μπορούμε να μεταβάλουμε:

Πεδίο	Περιγραφή
Enable	Αν δώσουμε την τιμή TRUE, τότε η οντότητα θα είναι ενεργή.
InitOrder	Το αρχείο της δέσμης εντολών (script file) που περιέχει τις εντολές που ρυθμίζουν τη συμπεριφορά της οντότητας αποτελείται από δύο ενότητες. Στην πρώτη αρχικοποιούμε την οντότητα, δίνουμε δηλαδή κάποιες αρχικές παραμέτρους. Η δεύτερη ενότητα περιέχει εντολές που εκτελούνται κατά τη διάρκεια εκτέλεσης της εφαρμογής. Στο πεδίο αυτό εισάγουμε το όνομα της <u>πρώτης</u> ενότητας. Πρέπει να προσέξουμε να το γράψουμε όπως ακριβώς το έχουμε γράψει και στο αρχείο δέσμης εντολών (παίζουν ρόλο ακόμα και τα πεζά ή κεφαλαία γράμματα).
Origin	Η θέση της οντότητας στην εφαρμογή μας, στους άξονες x, y και z.
ScriptName	Το όνομα του αρχείου δέσμης εντολών. Το αρχείο έχει κατάληξη .s και πρέπει να βρίσκεται στο φάκελο C:\RealityFactory\Scripts. Και πάλι πρέπει να προσέξουμε να γράψουμε το όνομα του αρχείου σωστά.
StartOrder	Στο πεδίο αυτό εισάγουμε το όνομα της <u>δεύτερης</u> ενότητας. Όπως και στα παραπάνω πεδία, πρέπει να προσέξουμε να γράψουμε το όνομα σωστά.

Αν δεν κάνουμε καμία αλλαγή στις ιδιότητες της οντότητας (πέρα από το να δώσουμε την τιμή TRUE στο πεδίο Enable), υπάρχει περίπτωση η οντότητα να έχει τοποθετηθεί

σε λάθος θέση σε σχέση με το επίπεδό μας. Αυτό θα φανεί μόνο εάν εκτελέσουμε την εφαρμογή (εικόνα 13.4). Η θέση του χρήστη είναι τέτοια, ώστε βρίσκεται κάτω από τη θέση της οντότητας SkyDome και όχι στο ίδιο σημείο ή πάνω απ' αυτή. Τη θέση της οντότητας PlayerStart τη διαπιστώνουμε εύκολα αν δούμε την παράμετρο Θέση-Origin στις ιδιότητες της, για παράδειγμα -1214,-265,12. Η τιμή που μας ενδιαφέρει είναι η μεσαία (-265), που ορίζει τη θέση στον κάθετο άξονα (y). Αντίθετα, η SkyDome έχει τοποθετηθεί στη θέση 0,0,0, που είναι και η προεπιλεγμένη θέση αρχικής τοποθέτησης όλων των οντοτήτων.



Εικόνα 13.4 Πρόβλημα θέσης της οντότητας Skydome

Καθώς κατασκευάζουμε την εφαρμογή και μετακινούμε σε άλλες θέσεις οντότητες και στερεά, είναι πιθανό όλο ή μέρος του επιπέδου να έχει κατασκευαστεί κάτω από τη θέση 0,0,0. Κάτι τέτοιο έχει συμβεί και στο παράδειγμά μας. Όχι μόνο η οντότητα PlayerStart, αλλά όλα τα γεωμετρικά στερεά βρίσκονται σε τέτοιες θέσεις, ώστε το κάτω μέρος τους να βρίσκεται σε αρνητική τιμή στον άξονα των y.

Ακόμα και αν επιλέξουμε και μετακινήσουμε τη SkyDome σε αρνητική θέση στον άξονα των y, πάλι το αποτέλεσμα θα είναι όπως στην εικόνα 13.4. Αυτό οφείλεται στο γεγονός ότι χωρίς τη χρήση κάποιου αρχείου δέσμης εντολών η οντότητα αρχικοποιείται πάντα στη θέση 0,0,0 ανεξάρτητα από την πραγματική της θέση στο επίπεδο. Ο μόνος τρόπος για να διορθώσουμε το πρόβλημα είναι να μετακινήσουμε όλα τα στοιχεία (οντότητες και στερεά), έτσι ώστε η νέα τους θέση κατά τον άξονα των y να είναι ίση ή μεγαλύτερη από 0.

Για να επιλεγούν και για να μετακινηθούν όλα τα στοιχεία του επιπέδου, κάνουμε τα εξής βήματα:

1. Βεβαιωνόμαστε ότι είμαστε σε κατάσταση επιλογής στερεών (☞ με πράσινο χρώμα) και όχι σε κατάσταση επιλογής προτύπων (☞ με μπλε χρώμα).
2. Σ' ένα από τα παράθυρα προεπισκόπησης κάνουμε αρκετό zoom out (μετακινώντας το ροδάκι του ποντικιού μας), ώστε να μπορούμε να βλέπουμε όλο το επίπεδο.
3. Σε κάποια από τις τρεις προεπισκοπήσεις (πλην της τρισδιάστατης) κάνουμε κλικ και σέρνουμε το ποντίκι, έτσι ώστε να δημιουργηθεί ένα νοητό τετράγωνο που περιβάλλει όλο το επίπεδό μας.
4. Με το εργαλείο μετακίνησης ☞ μετακινούμε όλο το επίπεδο προς τα επάνω.

Επιλέγουμε την οντότητα PlayerStart και στις ιδιότητές της ελέγχουμε εάν η νέα της θέση στον άξονα των y είναι σε θετική τιμή (οπωσδήποτε πάνω από 0). Εάν η μετακίνηση των στοιχείων του επιπέδου μας έχει γίνει σωστά, η οντότητα SkyDome θα λειτουργήσει και αυτή σωστά. Πρέπει όμως να προσέξουμε ένα ακόμα σημείο. Με τη χρήση της τεχνικής Skybox είχαμε εκμεταλλευτεί την κάτω έδρα του εξωτερικού στερεού και τη χρησιμοποιούσαμε ως έδαφος. Κάτι τέτοιο δεν μπορεί να γίνει όταν χρησιμοποιούμε την οντότητα SkyDome, γιατί το μοντέλο του χρήστη “βουλιάζει” μέσα στην κάτω έδρα (εικόνα 13.5) ή, ακόμα, βρίσκεται κάτω απ' αυτή (εικόνα 13.4).



Εικόνα 13.5 Πρόβλημα της κάτω έδρας με τη χρήση της οντότητας Skydome

Για να διορθωθεί το πρόβλημα, πρέπει να προσθέσουμε ένα ακόμα γεωμετρικό στερεό, που θα παίξει το ρόλο του “εδάφους”. Το μήκος και πλάτος του θα είναι όσο ή σχεδόν όσο το επίπεδό μας, αλλά με μικρό ύψος και με την ιδιότητα Συμπαγές-Solid. Του δίνουμε επίσης και την ανάλογη υφή. Έχοντας ολοκληρώσει και αυτό το βήμα, μπορούμε να εκτελέσουμε την εφαρμογή μας. Παρατηρούμε ότι στον ουρανό εμφανίζονται σύννεφα, τα οποία με το πέρασμα του χρόνου μετακινούνται. Αν περιμένουμε αρκετή ώρα, θα διαπιστώσουμε ότι και ο “ήλιος” αλλάζει θέση (εικόνα 13.6). Μάλιστα,

με το πέρασμα μεγάλου χρονικού διαστήματος θα πέσει σκοτάδι σ' όλο το επίπεδο, θα εμφανιστούν αστέρια και φεγγάρι.



Εικόνα 13.6 Ήλιος με την οντότητα Skydome

Όμως, οι μεγάλες δυνατότητες της οντότητας SkyDome βρίσκονται στον ορισμό και την παραμετροποίηση του αρχείου δέσμης εντολών. Στο Reality Factory περιλαμβάνεται ένα δείγμα τέτοιου αρχείου, το Sky2.s, που βρίσκεται στο φάκελο C:\RealityFactory\Scripts\Sample. Το αντιγράφουμε και το επικολλούμε στο φάκελο C:\RealityFactory\Scripts. Για να δούμε τις εντολές που περιέχει, κάνουμε διπλό κλικ επάνω του και επιλέγουμε να ανοίγει με το “Σημειωματάριο”. Οι δηλώσεις που περιλαμβάνει είναι:

```
{
    Start[ ()
    {
SetHeightOrigin(-1000);
SetLandscapeSize(40200);
SetSkyDome(4920, 128);
SetDesiredTriangles(1024);
SetTerrain("flat.bmp", "desert256.bmp");
SetDistanceFromSunFactor(40200);
SetTwilightDistanceFromHorizon(0.6);
SetLocation(920, 10, 16, 23);
SetTwilightColor(255, 101, 114, 255);
SetSkyColor(116, 208, 241, 255);
SetTimeScale(512);
DisableSun(false);
```

```

SetSunColor(255, 226, 146, 255);
SetSunScale(8);
SetAmbientLightColor(128, 128, 128, 128);
DisableCloud(false);
UseSkyFog(true);
    }

    Run[ ()
    {
        SetWindDirection("northeast");
    }
}

```

Ό,τι εγγραφές περιλαμβάνονται από τη δήλωση Start και κάτω αφορούν εντολές αρχικοποίησης της οντότητας SkyDome. Ό,τι εγγραφές βρίσκονται από τη δήλωση Run και κάτω αφορούν εντολές προς την οντότητα κατά την εκτέλεση της εφαρμογής. Συνεπώς, αν θέλουμε να εκτελείται αυτό το αρχείο δέσμης εντολών, συμπληρώνουμε τα πεδία στις ιδιότητες της οντότητας SkyDome στο πάνελ ιδιοτήτων ως εξής:

Πεδίο	Τιμή
Enable	TRUE
InitOrder	Start
ScriptName	Sky2.s
StartOrder	Run

Στο σημείο αυτό καλό είναι να αναλύσουμε τη δομή αλλά και τις εντολές που μπορούμε να χρησιμοποιήσουμε σ' ένα οποιαδήποτε αρχείο δέσμης εντολών-script file παραμετροποίησης της οντότητας SkyDome.

Κατ' αρχάς παρατηρούμε ότι το αρχείο Sky2.s αρχίζει με ένα άγκιστρο ανοίγματος ({) και τελειώνει με ένα κλεισίματος (}). Τα άγκιστρα ορίζουν την αρχή και το τέλος όλων των δηλώσεων. Ο προγραμματιστής που έγραψε το συγκεκριμένο αρχείο αποφάσισε να ονομάσει την ομάδα δηλώσεων αρχικοποίησης Start και την ομάδα δηλώσεων κατά τη διάρκεια εκτέλεσης την ονόμασε Run. Μπορούμε να δώσουμε ό,τι όνομα θέλουμε, αρκεί μετά απ' αυτό να υπάρχει μία αγκύλη αρχής και ένα άνοιγμα-κλείσιμο παρένθεσης, για παράδειγμα Initialize[(). Στο τέλος των δηλώσεων αρχικοποίησης υπάρχει βέβαια μία αγκύλη κλεισίματος (}). Μέσα σε άγκιστρα ({}) περιλαμβάνονται όλες οι εντολές και στο τέλος κάθε εντολής υπάρχει ένα ερωτηματικό (;).

Όσον αφορά τις εντολές που μπορούμε να δώσουμε σε κάθε γραμμή, χωρίζονται σε τρεις κατηγορίες:

1. Σε εντολές που μπορούν να δοθούν μόνο στο τμήμα αρχικοποίησης και δεν έχουν καμία ισχύ αν χρησιμοποιηθούν στο τμήμα εκτέλεσης.
2. Σε εντολές που μπορούν να χρησιμοποιηθούν μόνο στο τμήμα εκτέλεσης και δεν έχουν καμία ισχύ αν χρησιμοποιηθούν στο τμήμα αρχικοποίησης.
3. Σε εντολές που μπορούν να χρησιμοποιηθούν και στα δύο τμήματα και πάντα εκτελούνται, ανεξάρτητα από το τμήμα που είναι τοποθετημένες.

Δεν έχει σημασία η σειρά με την οποία χρησιμοποιούνται οι εντολές, αρκεί να είναι τοποθετημένες στο ανάλογο για το είδος τους τμήμα. Αν δε χρησιμοποιήσουμε κάποια εντολή, τότε χρησιμοποιείται η προεπιλεγμένη γι' αυτή τιμή.

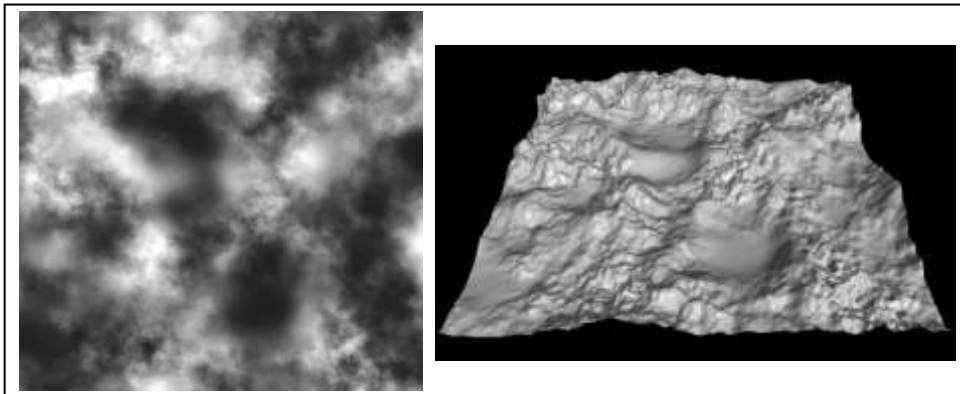
Στη συνέχεια αναλύεται κάθε εντολή, δίνοντας τη σύνταξή της και ένα παράδειγμα.

13.2.1 Εντολές της 1^{ης} κατηγορίας (τμήματος αρχικοποίησης)

Σύνταξη εντολής: SetTerrain(“χάρτης ύψους”, “όνομα υφής”);

Παράδειγμα: SetTerrain(“flat.bmp”, “desert256.bmp”);

Επεξήγηση εντολής: Μπορούμε να δηλώσουμε μία ειδική κατηγορία εικόνας με βάση την οποία η οντότητα SkyDome δημιουργεί έναν υψομετρικό χάρτη (heightmap) που θα χρησιμοποιηθεί στον ορίζοντα (εικόνα 13.7).



Εικόνα 13.7 Εικόνα κατασκευής υψομετρικού χάρτη και το ανάγλυφο που δημιουργεί

Ανάλογα με τον τόνο του γκρι, δημιουργείται και το αντίστοιχο ύψος (μαύρο=επίπεδο=0 texels, άσπρο=μέγιστο ύψος=255 texels). Υπάρχουν πάρα πολλές τέ-

τοιες εικόνες στο Διαδίκτυο, που δημιουργούν ενδιαφέροντα ανάγλυφα. Η εικόνα πρέπει να είναι 8bit στην κλίμακα του γκρι (grayscale), τετράγωνη και με διαστάσεις δύναμη του 2. Η εικόνα του υψομετρικού χάρτη πρέπει είναι αποθηκευμένη στο φάκελο C:\RealityFactory\Media\ Bitmaps\Terrain. Στο δεύτερο μέρος της εντολής ορίζουμε ποια υφή θα ντύνει τον υψομετρικό χάρτη. Η εικόνα πρέπει να βρίσκεται και αυτή στο φάκελο C:\RealityFactory\Media\ Bitmaps\Terrain. Στο Reality Factory περιλαμβάνεται μία απλή μαύρη εικόνα για heightmap (black.bmp) και μία εικόνα για έδαφος (desert256.bmp).

Σύνταξη εντολής: SetVertScale(κλίμακα);

Παράδειγμα: SetVertScale(4.5);

Επεξήγηση εντολής: Η εντολή ορίζει το πόσο θα πολλαπλασιαστεί (κλίμακα) το ύψος κάθε σημείου στον υψομετρικό χάρτη της προηγούμενης εντολής. Αν, για παράδειγμα, είχαμε έναν τόνο του γκρι που δημιουργούσε σ' ένα σημείο ύψος 195 texels, με το παράδειγμα αυτής της εντολής το τελικό ύψος θα είναι $195 \times 4.5 = 877.5$ texels. Πρέπει να προσέξουμε να μη δημιουργούνται ύψη που υπερβαίνουν το ύψος του "κουτιού" που περιβάλλει το επίπεδό μας. Η προεπιλεγμένη τιμή είναι 1.

Σύνταξη εντολής: SetLandscapeSize(πλάτος);

Παράδειγμα: SetLandscapeSize(40200);

Επεξήγηση εντολής: Ορίζει το πλάτος (και το μήκος) της βάσης του θόλου, που πρέπει να είναι αρκετά μεγάλο, ώστε να καλύπτεται όλο το επίπεδο. Η προεπιλεγμένη τιμή είναι 4096.

Σύνταξη εντολής: SetHorizScale(κλίμακα);

Παράδειγμα: : SetHorizScale(4.3);

Επεξήγηση εντολής: Η εντολή ορίζει την κλίμακα της τιμής της προηγούμενης εντολής. Η προεπιλεγμένη τιμή είναι 4.

Σύνταξη εντολής: SetHeightOrigin(ύψος);

Παράδειγμα: SetHeightOrigin(-250);

Επεξήγηση εντολής: Ορίζει τη θέση του χαμηλότερου σημείου της βάσης του θόλου στον άξονα των y. Μπορεί να πάρει θετικές ή αρνητικές τιμές, έτσι ώστε το χαμηλότερο σημείο να βρίσκεται κάτω από τη γεωμετρία του επιπέδου μας (όπως και πρέπει να είναι). Η προεπιλεγμένη τιμή είναι 0.

Σύνταξη εντολής: SetDesiredTriangles(τιμή);

Παράδειγμα: SetDesiredTriangles(1024);

Επεξήγηση εντολής: Ορίζει τον αριθμό των πολυγώνων που θα αποτελέσουν τη βάση του θόλου. Μεγαλύτερες τιμές έχουν ως αποτέλεσμα περισσότερη λεπτομέρεια, αλλά επιβαρύνουν την εφαρμογή. Η προεπιλεγμένη τιμή είναι 500.

Σύνταξη εντολής: SetTwilightDistanceFromHorizon(ποσοστό);

Παράδειγμα: SetTwilightDistanceFromHorizon(0.2);

Επεξήγηση εντολής: Η εντολή ορίζει το σημείο (ποσοστό του ζενίθ) κάτω από το οποίο θα δημιουργείται λυκόφως. Η προεπιλεγμένη τιμή είναι 0.4.

Σύνταξη εντολής: SetSkyDome(ύψος, ανάλυση);

Παράδειγμα: SetSkyDome(1024, 128);

Επεξήγηση εντολής: Ορίζονται το ύψος και η ανάλυση του θόλου. Η προεπιλεγμένη τιμή για το ύψος είναι 400 texels. Η ανάλυση είναι ο αριθμός των πολυγώνων που χρησιμοποιούνται για τη δημιουργία του θόλου. Μεγαλύτερες τιμές έχουν ως αποτέλεσμα περισσότερη λεπτομέρεια, αλλά επιβαρύνουν την εφαρμογή. Η προεπιλεγμένη τιμή είναι 32.

Σύνταξη εντολής: SetSkyTexture("όνομα υφής");

Παράδειγμα: SetSkyTexture("blue.bmp");

Επεξήγηση εντολής: Ορίζεται το όνομα της υφής που θα "ντύσει" το θόλο. Το αρχείο πρέπει να είναι αποθηκευμένο στο φάκελο C:\RealityFactory\Media\Bitmaps\Terrain. Αρκεί και μία μονόχρωμη εικόνα.

Σύνταξη εντολής: DisableSun(true/false);

Παράδειγμα: DisableSun(false);

Επεξήγηση εντολής: Αν δώσουμε την τιμή true, τότε στο θόλο δε θα εμφανίζονται ο ήλιος, το φεγγάρι και τα αστέρια. Προεπιλεγμένη τιμή είναι η false.

Σύνταξη εντολής: SetSunScale(κλίμακα);

Παράδειγμα: SetSunScale(6);

Επεξήγηση εντολής: Η τιμή με την οποία θα πολλαπλασιαστεί το αρχικό μέγεθος του ήλιου. Η προεπιλεγμένη τιμή είναι 4.

Σύνταξη εντολής: SetDistanceFromSunFactor(απόσταση);

Παράδειγμα: SetDistanceFromSunFactor(1200);

Επεξήγηση εντολής: Καθορίζει την απόσταση μεταξύ της βάσης του θόλου και του ήλιου στην οποία η αντανάκλαση του φωτός είναι 1.0. Στην ουσία καθορίζει τη διάχυση του φωτός από τον ήλιο. Μεγάλες τιμές προκαλούν μεγάλη διάχυση. Η προεπιλεγμένη τιμή είναι 900.

Σύνταξη εντολής: SetMoonColor(κόκκινο, πράσινο, μπλε, διαφάνεια);

Παράδειγμα: SetMoonColor(240, 240, 240, 255);

Επεξήγηση εντολής: Ορίζει το χρώμα του φεγγαριού (οι τρεις πρώτες τιμές) και το βαθμό διαφάνειάς του (τέταρτη τιμή 0=απόλυτα διαφανές, 255=απόλυτα αδιαφανές). Η προεπιλεγμένη τιμή είναι (255, 255, 255, 255).

Σύνταξη εντολής: SetMoonPhase(φάση);

Παράδειγμα: SetMoonPhase(3);

Επεξήγηση εντολής: Η εντολή ορίζει τη φάση στην οποία βρίσκεται η σελήνη. Οι τιμές είναι: 0 για νέα σελήνη, 1 για το πρώτο τέταρτο, 2 για πανσέληνο και 3 για τελευταίο τέταρτο. Η προεπιλεγμένη τιμή είναι 2.

Σύνταξη εντολής: DisableCloud(true/false);

Παράδειγμα: DisableCloud(false);

Επεξήγηση εντολής: Αν δοθεί η τιμή true, τότε δε θα εμφανίζονται σύννεφα στον ουρανό. Η προεπιλεγμένη τιμή είναι false.

Σύνταξη εντολής: SetLocation(γεωγραφικό πλάτος, μήνας, μέρα, ώρα);

Παράδειγμα: SetLocation(52, 9, 25, 12);

Επεξήγηση εντολής: Η γεωγραφική θέση του παίκτη, ο μήνας, η μέρα και η ώρα. Η εντολή χρησιμοποιείται για να καθοριστεί η θέση του ήλιου στο θόλο. Ειδικά για το γεωγραφικό πλάτος, χρησιμοποιούμε θετικές τιμές για θέση βόρεια του ισημερινού και αρνητικές για νότια. Οι τιμές για το μήνα είναι 1-12, για τη μέρα 1-31 και για την ώρα 0-23. Η προεπιλεγμένη τιμή είναι (49, 10, 24, 12).

13.2.2 Εντολές της 2^{ης} κατηγορίας (τμήματος εκτέλεσης)

Σύνταξη εντολής: self.think=όνομα

Παράδειγμα: self.think=run

Επεξήγηση εντολής: Μία μεταβλητή όπου ορίζουμε ποια ομάδα εντολών θα εκτελεστεί την επόμενη φορά που θα “τρέξει” το αρχείο δέσμης εντολών.

Σύνταξη εντολής: self.time=τιμή

Παράδειγμα: self.time=800

Επεξήγηση εντολής: Μία μεταβλητή όπου ορίζουμε πόσα δευτερόλεπτα πέρασαν από την εκτέλεση του αρχείου δέσμης εντολών. Στην ουσία χρησιμοποιείται ως μετρητής χρόνου.

Σύνταξη εντολής: self.daytime=τιμή

Παράδειγμα: self.daytime=1000

Επεξήγηση εντολής: Μία μεταβλητή όπου ορίζουμε πόσα δευτερόλεπτα απομένουν μέχρι τα μεσάνυχτα της τρέχουσας ημέρας.

13.2.3 Εντολές της 3^{ης} κατηγορίας (κοινές στα δύο τμήματα)

Σύνταξη εντολής: SetWindDirection(“κατεύθυνση”);

Παράδειγμα: SetWindDirection(“west”);

Επεξήγηση εντολής: Η εντολή καθορίζει την κατεύθυνση του ανέμου. Επηρεάζει την κίνηση των νεφών. Οι τιμές που μπορεί να δεχτεί είναι: “north”, “northeast”, “northwest”, “south”, “southeast”, “southwest”, “east”, “west”.

Σύνταξη εντολής: SetTimeScale(κλίμακα);

Παράδειγμα: SetTimeScale(20);

Επεξήγηση εντολής: Καθορίζει την ταχύτητα με την οποία κυλάει ο χρόνος και κατ’ επέκταση πόσο γρήγορα κινούνται ο ήλιος, η σελήνη και τα αστέρια. Αν, για παράδειγμα, δώσουμε την τιμή 10, κάθε 6 λεπτά θα περνά 1 ώρα (10πλάσια ταχύτητα).

Σύνταξη εντολής: SetSunColor(κόκκινο, πράσινο, μπλε, διαφάνεια);

Παράδειγμα: SetSunColor(255, 226, 146, 255);

Επεξήγηση εντολής: Καθορίζει το χρώμα του ήλιου και το βαθμό διαφάνειάς του. Το πιο πάνω παράδειγμα δίνει ένα κίτρινο χρώμα. Η προεπιλεγμένη τιμή είναι (255, 128, 0, 255).

Σύνταξη εντολής: SetSkyColor(κόκκινο, πράσινο, μπλε, διαφάνεια);

Παράδειγμα: SetSkyColor(116, 208, 241, 255);

Επεξήγηση εντολής: Καθορίζει το χρώμα του ουρανού και το βαθμό διαφάνειάς του. Το πιο πάνω παράδειγμα δίνει ένα ανοιχτό γαλαζοπράσινο χρώμα. Η προεπιλεγμένη τιμή είναι (164, 200, 255, 255).

Σύνταξη εντολής: SetTwilightColor(κόκκινο, πράσινο, μπλε, διαφάνεια);

Παράδειγμα: SetTwilightColor(255, 101, 114, 255);

Επεξήγηση εντολής: Καθορίζει το χρώμα του λυκόφωτος και το βαθμό διαφάνειάς του. Το πιο πάνω παράδειγμα δίνει ένα ανοιχτό κόκκινο χρώμα. Η προεπιλεγμένη τιμή είναι (255, 0, 0, 255).

Σύνταξη εντολής: SetAmbientLightColor(κόκκινο, πράσινο, μπλε, διαφάνεια);

Παράδειγμα: SetAmbientLightColor(128, 128, 128, 128);

Επεξήγηση εντολής: Καθορίζει το χρώμα του διάχυτου φωτός και το βαθμό διαφάνειάς του. Το πιο πάνω παράδειγμα δίνει ένα ανοιχτό γκρι χρώμα. Η προεπιλεγμένη τιμή είναι (255, 255, 255, 255).

Σύνταξη εντολής: UseSkyFog(true/false);

Παράδειγμα: UseSkyFog(true);

Επεξήγηση εντολής: Η εντολή καθορίζει το χρώμα της ομίχλης (ίδιο με το χρώμα του ουρανού) που δεν επιτρέπει το χρήστη να δει σε μεγάλη απόσταση.

Στο παρακάτω παράδειγμα, που μπορεί ο ενδιαφερόμενος να δοκιμάσει, ορίσαμε ένα μικρό αριθμό πολυγώνων για τη βάση του θόλου και έναν ικανοποιητικό αριθμό πολυγώνων για τον ίδιο το θόλο, ο οποίος έχει ύψος 1.024 texels. Αυξήσαμε λίγο το μέγεθος του ήλιου, στον οποίο δώσαμε ένα έντονο κίτρινο χρώμα. Καθορίσαμε να υπάρχει ένας αρκετά μεγάλος βαθμός διάχυσης του φωτός. Το γεωγραφικό πλάτος αντιστοιχεί με το μέσο περίπου της Ελλάδας, ενώ η ημερομηνία είναι 25 Ιουλίου και η ώρα 6 το πρωί. Το φεγγάρι είναι στο τελευταίο τέταρτο και ο αέρας φυσά από τα βορειοανατολικά. Τέλος, δώσαμε μία πολύ μεγάλη ταχύτητα με την οποία κυλά ο χρόνος.

```
{
    Start[ ()
    {
SetDesiredTriangles(254);
SetSkyDome(1024, 128);
SetSunScale(5);
SetSunColor(255, 255, 50, 255);
SetDistanceFromSunFactor(2024);
SetLocation(38, 7, 25, 6);
SetTimeScale(1024);
SetMoonPhase(3);
    } ]

    Run[ ()
    {
        SetWindDirection("northeast");
    } ]
}
```

Κεφάλαιο 14^ο

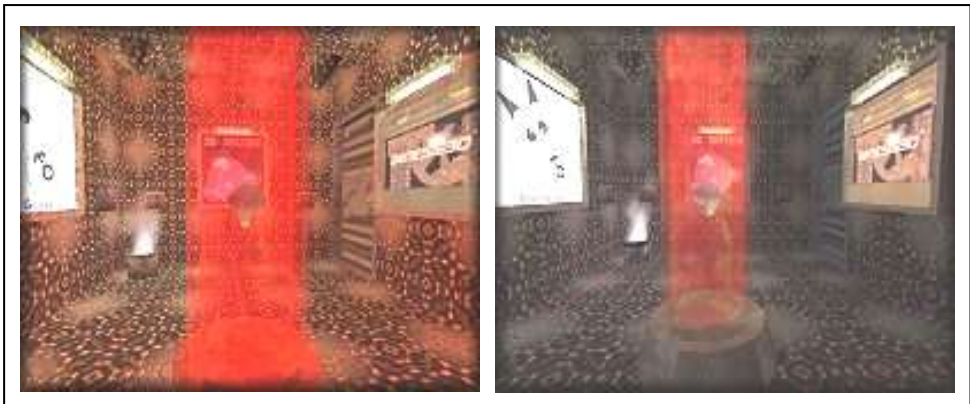
Φως και ήχος

Πολύ μεγάλο ρόλο στο ρεαλισμό μίας εφαρμογής παίζουν ο τρόπος με τον οποίο χρησιμοποιούμε τους ήχους, τα φώτα και τις σκιές που αυτά δημιουργούν. Για το λόγο αυτό άλλωστε, το Reality Factory περιλαμβάνει αρκετές οντότητες που χειρίζονται αυτές τις δύο παραμέτρους, τις οποίες και θα αναλύσουμε στη συνέχεια του κεφαλαίου. Πρέπει όμως να επισημάνουμε ότι η χρήση των οντοτήτων χειρισμού φωτός και ήχου θα πρέπει να γίνει χωρίς υπερβολές, γιατί εύκολα μπορεί να επιβαρυνθεί με σημαντικό υπολογιστικό φορτίο ο ηλεκτρονικός υπολογιστής του τελικού χρήστη.

14.1 Οντότητες φωτός

14.1.1 Η ιδιότητα Φως και η ιδιότητα Ακτινοβολία

Μία μέθοδος να φωτίσουμε τους διάφορους χώρους της εφαρμογής μας, χωρίς τη χρήση κάποιας ειδικής οντότητας, είναι να δώσουμε την ιδιότητα Φως-Light σε έδρες γεωμετρικών στερεών που επιθυμούμε να λειτουργούν ως φώτα (επιλέγοντας το στερεό, πατώντας Page Down, επιλέγοντας μία έδρα και δίνοντας την παραπάνω ιδιότητα από το πάνελ ιδιοτήτων). Το παραγόμενο χρώμα φωτός εξαρτάται από το χρώμα της υφής που “ντύνει” τη συγκεκριμένη έδρα (εικόνα 14.1). Για να λειτουργήσει αυτός ο τρόπος φωτισμού, πρέπει στην καρτέλα παραμέτρων μεταγλώττισης (πατώντας το πλήκτρο ) να τσεκάρουμε το τετραγωνάκι Ακτινοβολία-Radiosity (εικόνα 14.2). Αν δεν τσεκάρουμε το τετραγωνάκι αυτό, παρότι οι έδρες θα έχουν την ιδιότητα Φως, δε θα παράγουν καθόλου φως.



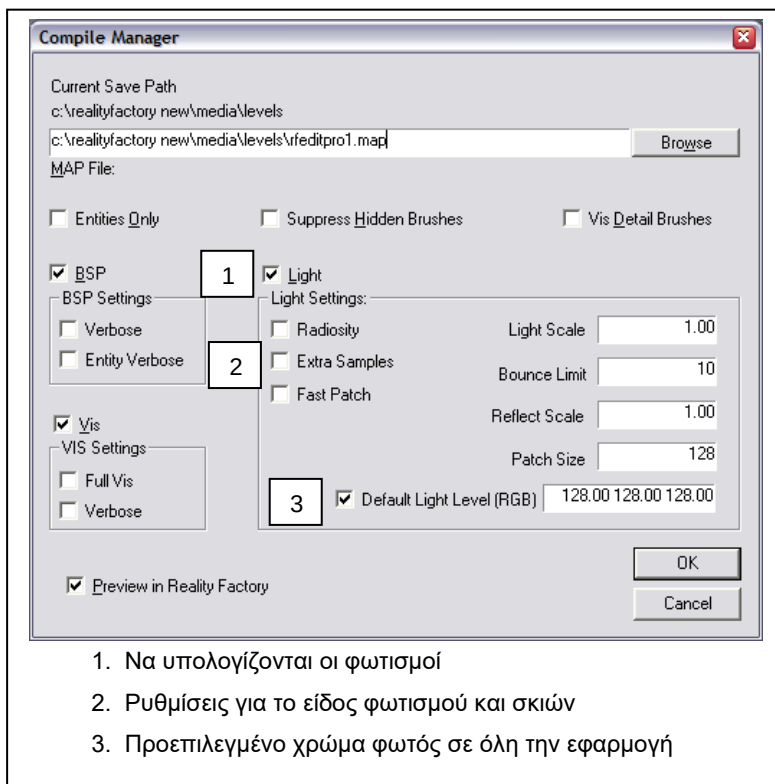
Εικόνα 14.1 Με και χωρίς Ακτινοβολία
(Πηγή: <http://realityfactory.info.ms/>)

Οι επιπλέον ρυθμίσεις που μπορούμε να κάνουμε με ενεργοποιημένη την Ακτινοβολία είναι (εικόνα 14.2):

1. Το φως μπορεί να “αναπηδά” στις άλλες επιφάνειες και ο αριθμός των αναπηδήσεων καθορίζεται από την τιμή που θα δώσουμε στο πλαίσιο Όριο Αναπηδήσεων-Bounce Limit.
2. Το πόσο λεπτομερής θα είναι ο φωτισμός εξαρτάται από την τιμή που θα δώσουμε στο πλαίσιο Μέγεθος Κομματιού-Patch Size. Μεγάλες τιμές παράγουν λιγότερο λεπτομερές και ευκρινές φως. Μικρότερες τιμές παράγουν περισσότερο λεπτομερή φωτισμό. Η τιμή 128 δίνει ικανοποιητικά αποτελέσματα.
3. Ακόμα περισσότερο λεπτομερή φωτισμό θα έχουμε αν τσεκάρουμε το τετραγωνάκι Επιπλέον Δειγματοληψία-Extra Samples.
4. Πόσο μεγάλος θα είναι ο χώρος που θα φωτίζεται καθορίζεται από την τιμή της παραμέτρου Κλίμακα Φωτός-Light Scale. Η τιμή αυτή επιδρά επίσης στο χώρο που φωτίζουν και οι άλλες οντότητες φωτισμού, που θα δούμε στη συνέχεια.
5. Για να γίνει λίγο πιο γρήγορος ο υπολογισμός, καλό είναι να τσεκάρουμε το τετραγωνάκι Fast Patch.
6. Το πόσο αντανάκλαστικές θα είναι οι επιφάνειες των στερεών καθορίζεται από την τιμή που θα δώσουμε στο πλαίσιο Κλίμακα Αντανάκλασης-Reflect Scale. Μεγαλύτερες τιμές κάνουν το χώρο φωτεινότερο, ενώ μικρότερες κάνουν πιο έντονες τις σκιές.

Εννοείται βέβαια ότι πρέπει να είναι τσεκαρισμένο το τετραγωνάκι Φως-Light (προεπιλεγμένο) για να υπολογίζονται από το Reality Factory οι φωτισμοί είτε από την Ακτινοβολία είτε από οντότητες φωτισμού. Μία τελευταία παράμετρος που μπορεί να ρυθμιστεί από την καρτέλα μεταγλώττισης είναι το χρώμα που θα έχει το διάχυτο φως σ' όλη την εφαρμογή μας. Η προεπιλεγμένη ρύθμιση είναι 128.00, 128.00, 128.00 (για τα χρώματα κόκκινο, πράσινο και μπλε), που δίνει έναν απαλό γκρι φωτισμό. Είτε κάνουμε χρήση της Ακτινοβολίας είτε εισάγουμε στην εφαρμογή μας οντότητες φωτισμού, πρέπει να ρυθμίσουμε αυτή την παράμετρο ώστε να είναι αισθητή η χρήση φώτων. Μικρότερες τιμές κάνουν πιο σκοτεινή την εφαρμογή, ενώ μεγαλύτερες πιο φωτεινή.

Η χρήση της Ακτινοβολίας αυξάνει σημαντικά το χρόνο μεταγλώττισης (από μερικά δευτερόλεπτα σε αρκετά λεπτά). Παράλληλα δημιουργεί ένα ιδιαίτερα μεγάλο αρχείο με πληροφορίες για την επίδραση του φωτός σε όλες τις έδρες των στερεών, που μπορεί να υπερβαίνει και το 1Gb. Τα παραπάνω καθιστούν δύσκολο το συνεχή έλεγχο της πορείας κατασκευής της εφαρμογής μας, ενώ αντίστοιχα αποτελέσματα μπορούμε να επιτύχουμε με τη χρήση οντοτήτων φωτισμού.



Εικόνα 14.2 Ρυθμίσεις φωτισμού στην καρτέλα μεταγλώττισης

14.1.2 Η οντότητα Light

Για τις ανάγκες του παραδείγματός μας, θα μειώσουμε αρχικά το διάχυτο φωτισμό σ' όλη την εφαρμογή, δίνοντας την τιμή 50.00 50.00 50.00 ως προεπιλεγμένο χρώμα φωτισμού στην καρτέλα μεταγλώττισης. Αυτό θα έχει ως αποτέλεσμα όλοι οι χώροι μας να γίνουν κατά πολύ πιο σκοτεινοί (εικόνα 14.3).



Εικόνα 14.3 Μειώνοντας το διάχυτο φωτισμό

Από το πάνελ εντολών και την καρτέλα Πρότυπα-Templates εισάγουμε μία οντότητα Light. Ο ρόλος της είναι να παράγει μία σφαίρα φωτός με ακτίνα που εμείς ορίζουμε. Την επιλέγουμε από τη λίστα της καρτέλας Ομάδες-Groups του πάνελ εντολών και στα δεξιά της οθόνης μας, στο πάνελ ιδιοτήτων, φαίνονται οι ιδιότητές της, που είναι οι εξής:

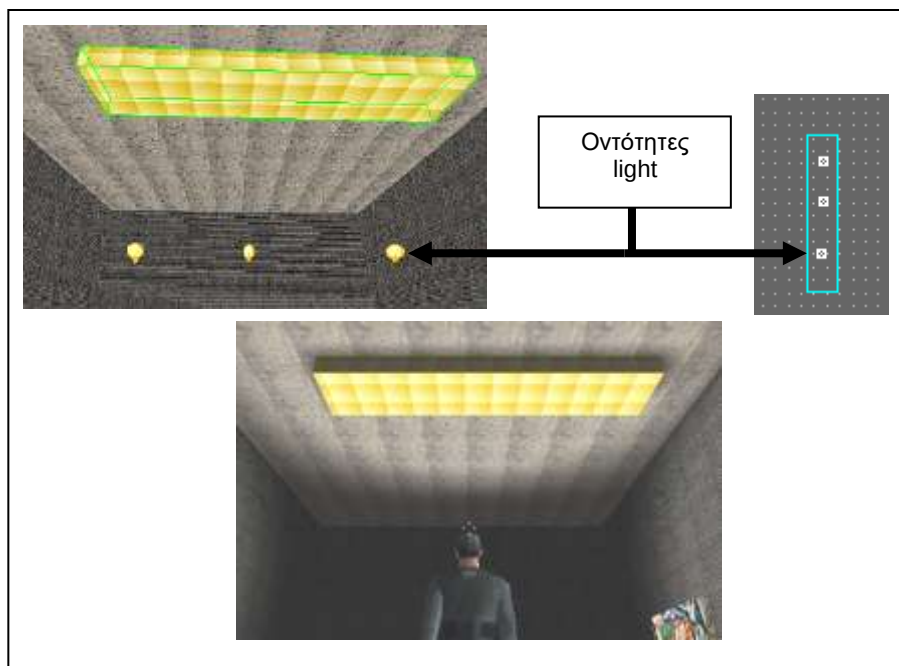
Πεδίο	Περιγραφή
Color	Το χρώμα του παραγόμενου φωτός. Με διπλό κλικ ανοίγει πάνελ επιλογής χρωμάτων.
Light	Η ακτίνα του παραγόμενου φωτός σε texels.
Origin	Η θέση της οντότητας. Η τιμή συμπληρώνεται αυτόματα.
Style	Το είδος του φωτός. Η τιμή 0 παράγει φως, η τιμή 1 παράγει σκοτάδι και η τιμή 11 παράγει ένα φως που τρεμοπαίζει. Τιμές πάνω από 11 προκαλούν πρόβλημα στο RFEEditPro, ενώ οι τιμές 2-10 δε χρησιμοποιούνται.

Η χρήση της οντότητας Light είναι ιδιαίτερα απλή. Μετά την εισαγωγή της την επιλέγουμε, την τοποθετούμε στην επιθυμητή θέση (με το εργαλείο ) και δίνουμε το χρώμα του φωτός που θα παράγει (εκτός αν ορίσουμε να παράγει σκοτάδι). Το μόνο σημείο που θα πρέπει να προσέξουμε είναι ο συνδυασμός της ακτίνας φωτός που παράγει (παράμετρος light στις ιδιότητες) και της θέσης της οντότητας. Υπάρχει δηλαδή η περίπτωση να έχουμε δώσει μία μικρή ακτίνα και ταυτόχρονα να έχουμε τοποθετήσει την οντότητα αρκετά ψηλά. Το αποτέλεσμα θα είναι το φως να μη φτάνει μέχρι το πάτωμα ή, ακόμα, να μη φαίνεται καθόλου φως.

Όπως και κάθε άλλη οντότητα, μπορούμε να την αντιγράψουμε και να την επικολλήσουμε σε άλλη θέση. Κάνοντας κάτι τέτοιο, δε χρειάζεται κάθε φορά να εισάγουμε άλλες και να τις παραμετροποιούμε ξανά (εκτός βέβαια εάν θέλουμε άλλο χρώμα φωτός ή άλλη ένταση). Για μεγαλύτερο ρεαλισμό μπορούμε να εισαγάγουμε στην εφαρμογή μας ένα στερεό, που θα το τοποθετήσουμε στο ταβάνι και θα το ντύσουμε με την κατάλληλη υφή, ώστε να παίζει το ρόλο του φωτιστικού σώματος (εικόνα 14.4).

Ενδεικτική παραμετροποίηση μίας οντότητας Light:

Πεδίο	Τιμή
Light	400
Style	0



Εικόνα 14.4 Τοποθετώντας οντότητες Light

14.1.3 Η οντότητα Spotlight

Την οντότητα light τη χρησιμοποιούμε όταν θέλουμε να φωτίσουμε χώρους. Μπορεί όμως να επιθυμούμε να εστιάσουμε το φωτισμό μόνο σε μία συγκεκριμένη επιφάνεια, και μάλιστα με αρκετή ένταση, όπως για παράδειγμα στον πίνακα που είχαμε τοποθετήσει σ' ένα από τα δωμάτιά μας σε προηγούμενο κεφάλαιο. Σ' αυτή την περίπτωση καταφεύγουμε στη χρήση της οντότητας Spotlight. Η Spotlight δεν παράγει μία σφαίρα φωτός αλλά έναν κώνο, με ένταση και γωνία πρόσπτωσης που εμείς καθορίζουμε.

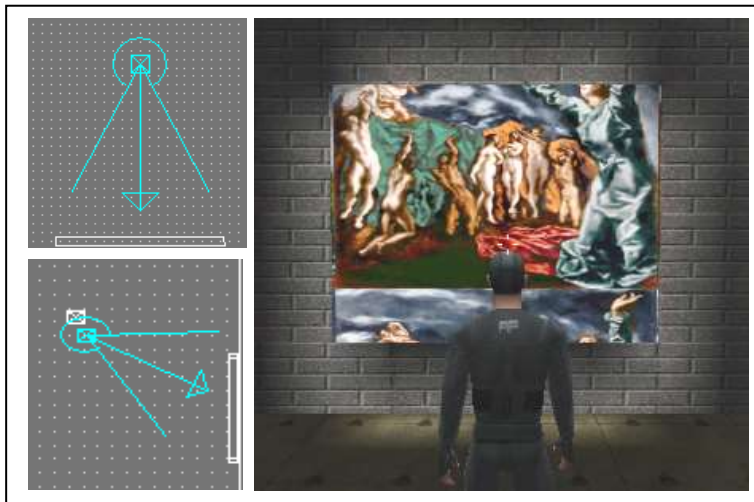
Την εισάγουμε από το πάνελ εντολών και την καρτέλα Πρότυπα-Templates. Την επιλέγουμε από τη λίστα της καρτέλας Ομάδες-Groups του πάνελ εντολών και στο πάνελ ιδιοτήτων φαίνονται οι ιδιότητές της, που είναι οι εξής:

Πεδίο	Περιγραφή
Angles	Οι γωνίες περιστροφής της οντότητας στους άξονες x, y και z. Τροποποιούμε τις τιμές έτσι, ώστε η οντότητα να φωτίζει το σημείο που θέλουμε. Μπορούμε να εισαγάγουμε και αρνητικές τιμές για περιστροφή στην αντίθετη κατεύθυνση.
Arc	Το τόξο φωτός σε μοίρες. Μικρές τιμές εστιάζουν το φως σε μικρότερη έκταση. Μεγαλύτερες προκαλούν μεγαλύτερη διάχυση.

Πεδίο	Περιγραφή
Color	Το χρώμα του παραγόμενου φωτός. Με διπλό κλικ ανοίγει πάνελ επιλογής χρωμάτων.
Light	Η ακτίνα του παραγόμενου φωτός σε texels.
Origin	Η θέση της οντότητας. Η τιμή συμπληρώνεται αυτόματα.
Style	Το είδος του φωτός. Η τιμή 0 παράγει φως, η τιμή 1 παράγει σκοτάδι και η τιμή 11 παράγει ένα φως που τρεμοπαίζει. Τιμές πάνω από 11 προκαλούν πρόβλημα στο RFEEditPro, ενώ οι τιμές 2-10 δε χρησιμοποιούνται.

Το RFEEditPro, για να διευκολύνει την τοποθέτηση της οντότητας, μάς δείχνει τον κώνο και τη γωνία πρόσπτωσης καθώς αλλάζουμε τις τιμές angles και arc (εικόνα 14.5). Επειδή ακριβώς το παραγόμενο φως είναι κώνος, μπορούμε να αυξήσουμε αρκετά την έντασή του. Ενδεικτικά, μπορούμε να παραμετροποιήσουμε μία οντότητα Spotlight ως εξής:

Πεδίο	Τιμή
Angles	0 90 -25
Arc	55
Light	500



Εικόνα 14.5 Τοποθετώντας την οντότητα spotlight

14.1.4 Η οντότητα DynamicLight

Η οντότητα DynamicLight χρησιμοποιείται όταν θέλουμε να τοποθετήσουμε ένα φως με δυναμικά και όχι με στατικά χαρακτηριστικά. Με τη συγκεκριμένη οντότητα μπορούμε να:

1. Μεταβάλλουμε την ακτίνα του παραγόμενου φωτός με βάση το χρόνο. Αυτή είναι και η πιο συνηθισμένη της χρήση.
2. Συνδέσουμε το φως της με κάποιο Κόκαλο-Bone ενός τρισδιάστατου μοντέλου (actor) της εφαρμογής μας, με αποτέλεσμα το φως να κινείται μαζί με τον actor.
3. Συνδέσουμε το φως της με κάποια άλλη οντότητα. Αυτό θα έχει ως αποτέλεσμα η DynamicLight να εξαρτάται από γεγονότα που συμβαίνουν στην άλλη οντότητα.
4. Συνδέσουμε το παραγόμενο φως με κάποιο γεωμετρικό στερεό που έχουμε μετατρέψει σε Μοντέλο-Model (κεφάλαιο 12.2.3). Αυτό θα έχει ως αποτέλεσμα η οντότητα DynamicLight να ακολουθεί το μοντέλο ή να ενεργοποιείται απ' αυτό (περισσότερα για τη δυνατότητα αυτή στο κεφάλαιο που αναλύονται οι σκανδάλες και τα γεγονότα).
5. Συνδέσουμε την οντότητα με κάποια σκανδάλη η οποία θα ενεργοποιεί ή θα απενεργοποιεί το φως (περισσότερα για τη δυνατότητα αυτή στο κεφάλαιο που αναλύονται οι σκανδάλες και τα γεγονότα).

Εννοείται βέβαια ότι είναι δυνατό να μεταβάλλεται η ακτίνα του φωτός και ταυτόχρονα η DynamicLight να είναι συνδεδεμένη με άλλη οντότητα, actor, μοντέλο ή σκανδάλη.

Δεν μπορούμε να έχουμε ταυτόχρονα πάνω από τριάντα ενεργές DynamicLight σε μία εφαρμογή. Αν όμως κάποιες DynamicLight δεν είναι ορατές από το σημείο που βρίσκεται ο χρήστης, τότε δεν υπολογίζονται στον παραπάνω αριθμό.

Για να δούμε τον τρόπο με τον οποίο λειτουργεί η DynamicLight (εικόνα 14.6), θα εισαγάγουμε μία στην εφαρμογή μας (από το πάνελ εντολών, καρτέλα Πρότυπα-Templates). Την επιλέγουμε από τη λίστα της καρτέλας Ομάδες-Groups του πάνελ εντολών και τη μετακινούμε στο δεύτερο δωμάτιο που έχουμε κατασκευάσει (με το εργαλείο ) . Στο πάνελ ιδιοτήτων φαίνονται οι ιδιότητές της, που είναι οι εξής:

Πεδίο	Περιγραφή
BoneName	Το όνομα του κοκάλου ενός actor στο οποίο συνδέεται η DynamicLight.
Color	Το χρώμα του παραγόμενου φωτός. Με διπλό κλικ ανοίγει πάνελ επιλογής χρωμάτων.

Πεδίο	Περιγραφή
EntityName	Το όνομα της οντότητας με την οποία είναι συνδεδεμένη η DynamicLight.
InterpolateValues	Αν δώσουμε την τιμή TRUE, τότε η μεταβολή της ακτίνας του φωτός θα γίνεται με ομαλό-συνεχή-μη διακριτό τρόπο στις τιμές που θα δώσουμε στο πεδίο RadiusFunction. Αν δώσουμε την τιμή FALSE, τότε η μεταβολή θα γίνεται με διακριτό τρόπο από τη μία θέση στην άλλη.
MaxRadius	Η μέγιστη ακτίνα του παραγόμενου φωτός σε texels.
MinRadius	Η ελάχιστη ακτίνα του παραγόμενου φωτός σε texels.
Model	Το όνομα του μοντέλου με το οποίο είναι συνδεδεμένη η οντότητα.
Origin	Η θέση της οντότητας. Η τιμή συμπληρώνεται αυτόματα.
RadiusFunction	Στο πεδίο αυτό χρησιμοποιούμε ένα συνδυασμό γραμμάτων της αγγλικής αλφαβήτου (a-z). Με βάση τη θέση του γράμματος στην αλφάβητο υπολογίζεται η ακτίνα του παραγόμενου φωτός. Το a είναι η ελάχιστη ακτίνα (η τιμή που δώσαμε στο πεδίο MinRadius), το z η μέγιστη (η τιμή που δώσαμε στο πεδίο MaxRadius) και κάθε άλλο γράμμα αντιπροσωπεύει μία ενδιάμεση τιμή ανάλογα με τη θέση του στην αλφάβητο (για παράδειγμα το k δίνει μία ακτίνα περίπου στο μέσο των τιμών της MinRadius και της MaxRadius).
RadiusSpeed	Ο χρόνος σε δευτερόλεπτα που θα διαρκεί ένας πλήρης κύκλος εναλλαγής της ακτίνας του φωτός. Αν, για παράδειγμα, δώσουμε την τιμή 2.0, όλος ο κύκλος θα διαρκεί 2 δευτερόλεπτα. Ο χρόνος κατανέμεται ισόποσα σε κάθε γράμμα που δώσαμε στο πεδίο RadiusFunction. Αν υποθέσουμε ότι στο πεδίο RadiusFunction δώσαμε το συνδυασμό azaz, στο πεδίο InterpolateValues δώσαμε την τιμή FALSE και η τιμή στο πεδίο RadiusSpeed είναι 4.0, τότε το πρώτο δευτερόλεπτο η ακτίνα θα είναι η ελάχιστη, το δεύτερο η μέγιστη, το τρίτο πάλι η ελάχιστη και το τέταρτο ξανά η μέγιστη. Αν στο πεδίο InterpolateValues δώσαμε την τιμή TRUE, τότε θα έχουμε ομαλή μετάβαση από την ελάχιστη στη μέγιστη (a-z) διάρκειας 1 δευτερολέπτου.
szEntityName	Το όνομα της συγκεκριμένης οντότητας DynamicLight. Αυτό το όνομα θα χρησιμοποιηθεί σε πεδία ιδιοτήτων άλλων οντοτήτων που θέλουμε να συνδέονται με αυτή.
TriggerName	Το όνομα της σκανδάλης που ενεργοποιεί τη συγκεκριμένη οντότητα DynamicLight.

Ενδεικτικά, οι παράμετροι σε μία Dynamiclight μπορεί να είναι:

Πεδίο	Τιμή	Πεδίο	Τιμή	Πεδίο	Τιμή
InterpolateValues	TRUE	MinRadius	150	RadiusSpeed	4
MaxRadius	500	RadiusFunction	azaz		



Εικόνα 14.6 Εναλλαγή ελάχιστης και μέγιστης ακτίνας στη Dynamiclight

14.1.5 Η οντότητα DSpotLight

Η οντότητα DSpotLight είναι ένας συνδυασμός DynamicLight και SpotLight. Παράγει δηλαδή κωνικό δυναμικό φως. Οι ιδιότητές της είναι οι εξής:

Πεδίο	Περιγραφή
Angles	Οι γωνίες περιστροφής της οντότητας στους άξονες x, y και z. Όμοια με τη SpotLight.
Arc	Το τόξο φωτός σε μοίρες. Όμοια με τη SpotLight.
BoneName	Το όνομα του κοκάλου ενός actor στο οποίο συνδέεται η οντότητα.
CastShadows	Αν δώσουμε την τιμή TRUE, τότε θα παράγονται σκιές.
Color	Το χρώμα του παραγόμενου φωτός.
EntityName	Το όνομα της οντότητας με την οποία είναι συνδεδεμένη η DSpotLight.
InterpolateValues	Αν δώσουμε την τιμή TRUE, τότε η μεταβολή της ακτίνας του φωτός θα γίνεται με ομαλό-συνεχή τρόπο. Όμοια με τη DynamicLight.
MaxRadius	Η μέγιστη ακτίνα του παραγόμενου φωτός σε texels.
MinRadius	Η ελάχιστη ακτίνα του παραγόμενου φωτός σε texels.
Model	Το όνομα του μοντέλου με το οποίο είναι συνδεδεμένη η οντότητα.
Origin	Η θέση της οντότητας. Η τιμή συμπληρώνεται αυτόματα.

Πεδίο	Περιγραφή
RadiusFunction	Στο πεδίο αυτό χρησιμοποιούμε ένα συνδυασμό γραμμάτων της αγγλικής αλφαβήτου (a-z). Όμοια με τη DynamicLight.
RadiusSpeed	Ο χρόνος σε δευτερόλεπτα που θα διαρκεί ένας πλήρης κύκλος εναλλαγής της ακτίνας του φωτός. Όμοια με τη DynamicLight.
Rotate	Αν δώσουμε την τιμή TRUE, τότε η DSpotLight θα περιστρέφεται μαζί με το μοντέλο ή τον actor που είναι συνδεδεμένη.
Shadow	Αν δώσουμε την τιμή TRUE, τότε η DSpotLight θα παράγει το αρνητικό του χρώματος που ορίσαμε στο πεδίο Color.
Style	Αν δώσουμε την τιμή 0, τότε το φως θα “σβήνει” σταδιακά καθώς απομακρυνόμαστε από το κέντρο. Με την τιμή 1 το φως θα “σβήνει” πιο ομαλά, ενώ με την τιμή 2 θα “σβήνει” απότομα.
szEntityName	Το όνομα της συγκεκριμένης οντότητας DSpotLight. Αυτό το όνομα θα χρησιμοποιηθεί σε πεδία ιδιοτήτων άλλων οντοτήτων που θέλουμε να συνδέονται με αυτή.
TriggerName	Το όνομα της σκανδάλης που ενεργοποιεί τη συγκεκριμένη οντότητα DSpotLight.

Ενδεικτικά, οι παράμετροι σε μία DSpotLight μπορεί να είναι:

Πεδίο	Τιμή	Πεδίο	Τιμή	Πεδίο	Τιμή
Angles	0 -90 -5	MaxRadius	1000	RadiusSpeed	2
Arc	55	MinRadius	300	Style	1
InterpolateValues	TRUE	RadiusFunction	aza		

14.1.6 Οι οντότητες SunLight και Sun

Για να φανούν τα αποτελέσματα της χρήσης των οντοτήτων Light, Spotlight, DynamicLight και DSpotLight, μειώσαμε το διάχυτο φωτισμό σ' όλη την εφαρμογή, δίνοντας την τιμή 50.00 50.00 50.00 ως προεπιλεγμένο χρώμα φωτισμού στην καρτέλα μεταγλώττισης. Αυτό είχε ως αποτέλεσμα ακόμα και οι ανοιχτοί χώροι να γίνουν κατά πολύ πιο σκοτεινοί. Χρειάζεται λοιπόν να φωτίσουμε και τους ανοιχτούς χώρους.

Στο Reality Factory περιλαμβάνονται δύο οντότητες που μπορούν να εξυπηρετήσουν το σκοπό αυτό, η Sunlight και η Sun. Και με τις δύο οντότητες μπορούμε, αν θέλουμε, να αφαιρέσουμε εντελώς το διάχυτο φωτισμό (αφαιρώντας το τσεκάρισμα στο τετραγωνάκι “Προεπιλεγμένο χρώμα φωτός σ' όλη την εφαρμογή”-“Default Light level”). Αυτό βέβαια θα δημιουργήσει την ανάγκη να στηριχτούμε αποκλειστικά στη χρήση οντοτήτων φωτισμού για να φωτίσουμε ανοιχτούς και κλειστούς χώρους.

Να σημειώσουμε επίσης ότι, επειδή οι οντότητες Sunlight και Sun παράγουν φως συνεχώς, χρειάζεται να καταφύγουμε στην τεχνική της χρήσης Skybox για τη δημιουργία ορίζοντα και ουρανού (κεφάλαιο 13.1). Δεν μπορούν δηλαδή οι οντότητες αυτές να συνδεθούν με την οντότητα Skydome και να παύουν να φωτίζουν όταν έχει πέσει το σκοτάδι.

Για τις ανάγκες του παραδείγματος ξεκινάμε μία νέα εφαρμογή, τοποθετούμε ένα Κούφιο-Hollow γεωμετρικό στερεό μεγάλων διαστάσεων, με Page Down δίνουμε την ιδιότητα Ουρανός-Sky σ' όλες τις εσωτερικές έδρες εκτός της κάτω και αναθέτουμε τις υφές του ουρανού με το πλήκτρο Προσαρμογής Ουρανού-Adjust Sky , διαδικασία την οποία παρουσιάσαμε στο κεφάλαιο 13.1. Τέλος, δεν παραλείπουμε να τοποθετήσουμε τις οντότητες Playerstart και Playersetup.

Τοποθετούμε αρχικά την οντότητα SunLight και την επιλέγουμε. Από το πάνελ ιδιοτήτων το σημείο που πρέπει να προσέξουμε είναι η γωνία πρόσπτωσης, γιατί λειτουργεί περίπου όπως η οντότητα Spotlight. Δημιουργεί δηλαδή φως μόνο προς την κατεύθυνση που είναι στραμμένη, αλλά καταλαμβάνει όλη την έκταση του επιπέδου. Για το λόγο αυτό, η ιδιότητα light πρέπει να θεωρηθεί περισσότερο ως ένταση παρά ως ακτίνα φωτός (μικρές τιμές παράγουν χαμηλό φωτισμό, υψηλές έντονο). Οι ιδιότητες της οντότητας SunLight αναλυτικά είναι:

Πεδίο	Περιγραφή
Angles	Οι γωνίες περιστροφής της οντότητας κατά τους άξονες x, y και z. Τροποποιούμε τις τιμές έτσι, ώστε η οντότητα να φωτίζει το σημείο που θέλουμε. Μπορούμε να εισάγουμε και αρνητικές τιμές για περιστροφή στην αντίθετη κατεύθυνση.
Color	Το χρώμα του παραγόμενου φωτός. Με διπλό κλικ ανοίγει πάνελ επιλογής χρωμάτων.
Light	Η ένταση του παραγόμενου φωτός.
Origin	Η θέση της οντότητας. Η τιμή συμπληρώνεται αυτόματα.
Style	Το είδος του φωτός. Η τιμή 0 παράγει φως, η τιμή 1 παράγει σκοτάδι και η τιμή 11 παράγει ένα φως που τρεμοπαίζει. Τιμές πάνω από 11 προκαλούν πρόβλημα στο RFEEditPro, ενώ οι τιμές 2-10 δε χρησιμοποιούνται.

Στο RFEEditPro υπάρχει και η οντότητα Sunlight (με I και όχι με L) που είναι κατάλοιπο από προηγούμενη έκδοση του RFEEditPro και δε χρησιμοποιείται. Οι ιδιότητές της είναι ίδιες με αυτές της SunLight.

Λίγο διαφορετική είναι η χρήση της οντότητας Sun (εικόνα 14.7). Κατ' αρχάς πρέπει να διαγράψουμε την οντότητα SunLight (την επιλέγουμε από την καρτέλα Ομάδες-Groups του πάνελ εντολών και πατάμε το πλήκτρο Delete). Εισάγουμε την οντότητα Sun και την επιλέγουμε για να μπορέσουμε να δούμε τις ιδιότητές της, που είναι οι εξής:

Πεδίο	Περιγραφή
Color	Το χρώμα του παραγόμενου φωτός. Με διπλό κλικ ανοίγει πάνελ επιλογής χρωμάτων.
DaviName	Το προκαθορισμένο όνομα της οντότητας. Δεν το αλλάζουμε, γιατί δημιουργούνται προβλήματα με το φως που παράγεται.
FallOffRadiusInTexels	Η ακτίνα του φωτός σε texels. Χρησιμοποιείται μόνο αν δώσουμε άλλη τιμή εκτός από 0 στο πεδίο FallOffType.
FallOffRadiusTopIntensity	Η ακτίνα όπου η ένταση του φωτός είναι μεγαλύτερη. Η τιμή πρέπει να είναι μικρότερη ή ίση με το πιο πάνω πεδίο. Χρησιμοποιείται μόνο αν δώσουμε άλλη τιμή εκτός από 0 στο πεδίο FallOffType.
FallOffSecretNumber	Οι τιμές που δίνουμε στο πεδίο αυτό επηρεάζουν τον τρόπο υπολογισμού της μείωσης της έντασης του φωτός. Οι τιμές που μπορεί να πάρει είναι 0-100. Η προεπιλεγμένη τιμή 1 δίνει και τα καλύτερα αποτελέσματα.
FallOffType	Όσο απομακρυνόμαστε από το κέντρο μίας πηγής φωτός, τόσο μειώνεται η έντασή του. Στο πεδίο αυτό καθορίζουμε τον τρόπο που θα εμφανίζεται η μείωση. Οι τιμές που μπορούμε να δώσουμε είναι: 0=καθόλου μείωση, 1=απότομη μείωση, 2=σταδιακή μείωση, 3=εναλλακτική σταδιακή μείωση.
Light	Η ένταση του παραγόμενου φωτός.
Origin	Η θέση της οντότητας. Η τιμή συμπληρώνεται αυτόματα.

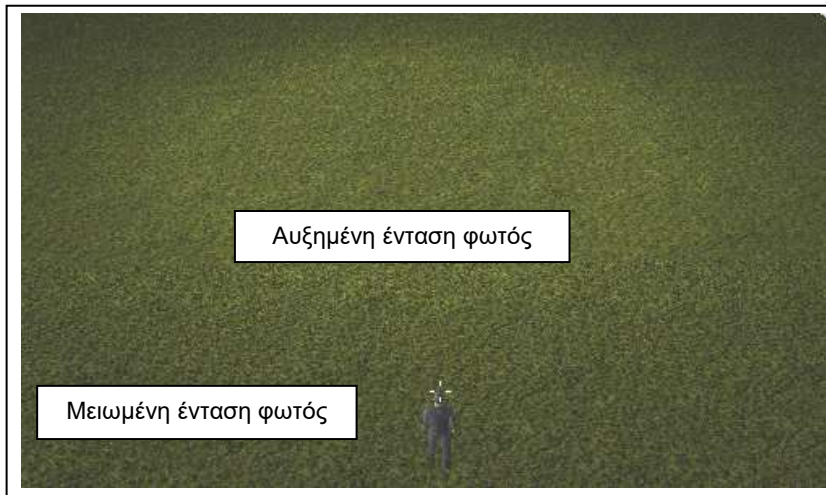
Ενδεικτική παραμετροποίηση της Sun:

Πεδίο	Τιμή	Πεδίο	Τιμή
FallOffRadiusInTexels	500	FallOffType	3

FallOffRadiusTopIntensity 200

Light

500



Εικόνα 14.7 Η χρήση της οντότητας Sun

Μία μέθοδος για να παραγάγουμε καλύτερης ποιότητας σκιές και ταυτόχρονα να μειώσουμε την επιβάρυνση της εφαρμογής από περιττούς υπολογισμούς είναι να μειώσουμε την κλίμακα φωτισμού στα γεωμετρικά στερεά μας (εικόνα 12.1, κεφάλαιο 12.1). Επιλέγουμε τις έδρες του στερεού (έχοντας πατήσει Page Down) και στο πάνελ ιδιοτήτων τις τιμές του Light Map Scale από 2 τις κάνουμε 1.

14.1.7 Οι οντότητες Corona και Flame

Καθώς τοποθετήσαμε την οντότητα Sun ή τη SunLight, είχαμε μεν την παραγωγή φωτός σε μεγάλη έκταση, αλλά απουσίαζε αυτός καθεαυτός ο ήλιος. Για να καλύψουμε το κενό, μπορούμε να τοποθετήσουμε μία οντότητα που ο ρόλος της δεν είναι να παράγει φως, αλλά ένα φαινόμενο που δημιουργεί το φως, την κορώνα, γι' αυτό άλλωστε η οντότητα ονομάζεται Corona (εικόνα 14.8). Οι ιδιότητές της είναι:

Πεδίο	Περιγραφή
BoneName	Το όνομα του κοκάλου ενός actor στο οποίο συνδέεται η Corona.
Color	Το χρώμα της κορώνας φωτός. Με διπλό κλικ ανοίγει πάνελ επιλογής χρωμάτων.

EntityName	Το όνομα της οντότητας με την οποία είναι συνδεδεμένη η οντότητα Corona.
------------	--

Πεδίο	Περιγραφή
FadeTime	Ο χρόνος σε δευτερόλεπτα που απαιτείται ώστε να εξαφανιστεί η οντότητα όταν βγει εκτός του οπτικού πεδίου του χρήστη.
MaxRadius	Η μέγιστη ακτίνα σε texels που μπορεί να έχει η οντότητα.
MaxRadiusDistance	Πάνω απ' αυτή την απόσταση, σε texels, η οντότητα θα έχει τη μέγιστη δυνατή ακτίνα (την τιμή MaxRadius).
MaxVisibleDistance	Η μέγιστη απόσταση σε texels που η οντότητα είναι ορατή.
MinRadius	Η ελάχιστη τιμή σε texels που μπορεί να έχει η ακτίνα της οντότητας.
MinRadiusDistance	Κάτω απ' αυτή την απόσταση, σε texels, η οντότητα θα έχει την ελάχιστη δυνατή ακτίνα (την τιμή MinRadius).
Model	Το όνομα του μοντέλου με το οποίο είναι συνδεδεμένη η οντότητα Corona.
Origin	Η θέση της οντότητας. Η τιμή συμπληρώνεται αυτόματα.
szEntityName	Το όνομα της συγκεκριμένης οντότητας Corona. Αυτό το όνομα θα χρησιμοποιηθεί σε πεδία ιδιοτήτων άλλων οντοτήτων που θέλουμε να συνδέονται με αυτή.
TriggerName	Το όνομα της σκανδάλης που ενεργοποιεί τη συγκεκριμένη οντότητα Corona.

Την κορώνα μπορούμε να τη χρησιμοποιήσουμε και σε εσωτερικούς χώρους, τοποθετώντας την κοντά σε πηγές φωτισμού όπως η Light και η Spotlight (εικόνα 14.9).

Ενδεικτική παραμετροποίηση της Corona στη θέση του ήλιου:

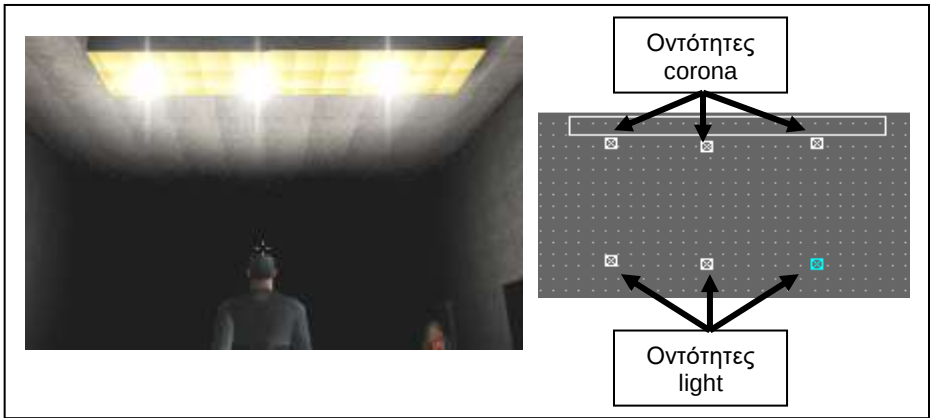
Πεδίο	Τιμή	Πεδίο	Τιμή
Color	255 255 174	MaxVisibleDistance	10000
MaxRadius	25	MinRadius	25
MaxRadiusDistance	1	MinRadiusDistance	1

Ενδεικτική παραμετροποίηση της Corona για εσωτερικό φωτισμό:

Πεδίο	Τιμή	Πεδίο	Τιμή
MaxRadius	4	MinRadius	1
MaxRadiusDistance	400	MinRadiusDistance	250
MaxVisibleDistance	500		



Εικόνα 14.8 Η χρήση της οντότητας Corona στη θέση του ήλιου



Εικόνα 14.9 Η χρήση της οντότητας Corona για εσωτερικό φωτισμό

Μία ιδιαίτερη πηγή φωτός, που είναι ταυτόχρονα και ένα ενδιαφέρον εφέ, είναι η φωτιά. Τη φωτιά τη χειρίζεται η οντότητα Flame. Η οντότητα παράγει έναν αριθμό κινούμενων σωματιδίων που τελικά δημιουργούν την ψευδαίσθηση της φωτιάς (εικόνα 14.10). Αναλυτικά οι ιδιότητές της είναι:

Πεδίο	Περιγραφή
AlphaName	Το όνομα του αρχείου εικόνας για το διαφανές μέρος της φω-

	τιάς. Δεν είναι απαραίτητο να δηλώσουμε κάποιο αρχείο.
Angles	Οι γωνίες περιστροφής της οντότητας κατά τους άξονες x, y και z. Τροποποιούμε τις τιμές έτσι, ώστε η φωτιά να έχει την κατεύθυνση που θέλουμε. Μπορούμε να εισαγάγουμε και αρνητικές τιμές για περιστροφή στην αντίθετη κατεύθυνση.
Πεδίο	Περιγραφή
BmpName	Το όνομα του αρχείου εικόνας για τη φωτιά. Δεν είναι απαραίτητο.
BoneName	Το όνομα του κοκάλου ενός actor στο οποίο συνδέεται η Flame.
ColorMax	Το μέγιστο χρώμα που θα παίρνει το κάθε σωματίδιο της φωτιάς. Με διπλό κλικ ανοίγει πάνελ επιλογής χρωμάτων.
ColorMin	Το ελάχιστο χρώμα που θα παίρνει το κάθε σωματίδιο της φωτιάς. Με διπλό κλικ ανοίγει πάνελ επιλογής χρωμάτων. Με τις ιδιότητες ColorMax και ColorMin μπορούμε να επιτύχουμε να αλλάζει χρώμα η φωτιά.
DestVariance	Τα σωματίδια που παράγουν τη φωτιά “ψεκάζονται” από ένα σημείο και καταλήγουν σ' ένα άλλο. Επειδή η φωτιά τρεμοπαίζει, το σημείο προορισμού μεταβάλλεται διαρκώς. Στο πεδίο αυτό ορίζουμε σε πόση έκταση (σε texels) θα μεταβάλλεται το σημείο προορισμού.
EntityName	Το όνομα της οντότητας με την οποία είναι συνδεδεμένη η Flame.
GcastShadows	Αν δώσουμε την τιμή TRUE, τότε η φωτιά θα παράγει και σκιές.
GcolorMax	Το μέγιστο χρώμα της παραγόμενης σκιάς. Με διπλό κλικ ανοίγει πάνελ επιλογής χρωμάτων.
GcolorMin	Το ελάχιστο χρώμα της παραγόμενης σκιάς. Με διπλό κλικ ανοίγει πάνελ επιλογής χρωμάτων.
Gintensity	Η μέγιστη ένταση του παραγόμενου φωτός.
GradiusMax	Η μέγιστη ακτίνα σε texels του παραγόμενου φωτός.
GradiusMin	Η ελάχιστη ακτίνα σε texels του παραγόμενου φωτός. Αν η διαφορά μεταξύ GradiusMax και GradiusMin είναι μεγάλη, το τρεμπάιγμα του φωτός μπορεί να γίνει ενοχλητικό.
Gravity	Η βαρύτητα κατά τους άξονες x, y και z. Οι τιμές επηρεάζουν τον τρόπο κίνησης των σωματιδίων της φωτιάς. Αν, για παράδειγμα, δώσουμε μία μεγάλη τιμή μόνο ως προς τον έναν άξονα, η φωτιά θα “λυγίζει” προς εκείνη την κατεύθυνση.

MaxScale	Η μέγιστη κλίμακα των υφών που “ντύνουν” τα σωματίδια.
MaxSpeed	Η μέγιστη ταχύτητα κίνησης των σωματιδίων της φωτιάς.
MaxUnitLife	Η μέγιστη διάρκεια ζωής, σε δευτερόλεπτα, του κάθε σωματιδίου.
MinScale	Η ελάχιστη κλίμακα των υφών που “ντύνουν” τα σωματίδια.
Πεδίο	Περιγραφή
MinSpeed	Η ελάχιστη ταχύτητα κίνησης των σωματιδίων.
MinUnitLife	Η ελάχιστη διάρκεια ζωής σε δευτερόλεπτα του κάθε σωματιδίου.
Model	Το όνομα του μοντέλου με το οποίο είναι συνδεδεμένη η οντότητα Flame.
Origin	Η θέση της οντότητας. Η τιμή συμπληρώνεται αυτόματα.
ParticleCreateRate	Κάθε πόσα δευτερόλεπτα να προστίθεται ακόμα ένα σωματίδιο φωτιάς. Μπορούμε να δώσουμε πολύ μικρές τιμές της τάξης των 0.05 δευτερολέπτων.
Scale	Η κλίμακα μεγέθους της φωτιάς. Μπορούμε να μεγαλώσουμε όλη τη φωτιά δίνοντας τιμές πάνω από 1.0.
SorceVariance	Στο πεδίο αυτό ορίζουμε σε πόση έκταση (σε texels) θα μεταβάλλεται το σημείο “ψεκασμού” των σωματιδίων της φωτιάς.
szEntityName	Το όνομα της συγκεκριμένης οντότητας Flame. Αυτό το όνομα θα χρησιμοποιηθεί σε πεδία ιδιοτήτων άλλων οντοτήτων που θέλουμε να συνδέονται με αυτή.
TotalLife	Πόσα δευτερόλεπτα να διαρκεί ο κάθε πίδακας σωματιδίων. Η τιμή 0 ορίζει συνεχή ροή.
TriggerName	Το όνομα της σκανδάλης που ενεργοποιεί τη συγκεκριμένη οντότητα Flame.

Ενδεικτική παραμετροποίηση της Flame:

Πεδίο	Τιμή	Πεδίο	Τιμή	Πεδίο	Τιμή
Angles	0 0 -90	GRadiusMin	180	MinSpeed	10
DestVariance	10	MaxScale	0.5	MinUnitLife	0.5
GCastShadows	TRUE	MaxSpeed	30	ParticleCreateRate	1
GIntensity	10	MaxUnitLife	1	Scale	2
GRadiusMax	200	MinScale	0.5	SorceVariance	1



Εικόνα 14.10 Η χρήση της οντότητας Flame

14.2 Οντότητες ήχου

Οι διάφοροι ήχοι στην εφαρμογή μας παίζουν και αυτοί σημαντικό ρόλο, όπως και η χρήση των φωτών. Μάλιστα, στο Reality Factory με τη χρήση ανάλογης οντότητας υπάρχει η δυνατότητα να χρησιμοποιήσουμε ήχους που μεταβάλλουν τις ιδιότητές τους (ένταση, κατεύθυνση) ανάλογα με τη θέση του χρήστη. Μία πρώτη οντότητα που μπορεί να διαχειριστεί ήχους και που ήδη αναλύσαμε τις ιδιότητές της είναι η PlayerStart. Σε γενικές γραμμές, στην PlayerStart μπορούμε να συνδέσουμε αρχεία ήχων που θα χρησιμοποιούνται ως μουσική υπόκρουση της εφαρμογής μας. Οι ιδιότητές της που αφορούν τους ήχους είναι:

Πεδίο	Περιγραφή
bSoundtrackLoops	Μπορούμε να συνδέσουμε στην οντότητα PlayerStart ένα μουσικό κομμάτι. Δίνοντας στο πεδίο αυτό την τιμή TRUE, το μουσικό κομμάτι θα παίζει αέναα.
szCDTrack	Ο αριθμός του μουσικού κομματιού από CD που θα παίζει.
szMIDIFile	Ποιο μουσικό κομμάτι –αρχείο με κατάληξη .mid– θα εκτελείται. Το αρχείο πρέπει να είναι αποθηκευμένο στο φάκελο C:\RealityFactory\Media\Midi.
szStreamingAudio	Ποιο μουσικό κομμάτι –αρχείο με κατάληξη .wav, .mp3 ή .ogg– θα εκτελείται. Το αρχείο πρέπει να είναι στο φάκελο C:\Reality–

Factory\Media\Audio.

Εννοείται βέβαια ότι πρέπει να επιλέξουμε αν το μουσικό κομμάτι προέρχεται από CD, από αρχείο Midi ή από αρχείο ήχου Wave και να συμπληρώσουμε μόνο το ανάλογο πεδίο.

14.2.1 Η οντότητα AudioSource3D

Στον πραγματικό κόσμο, όσο απομακρυνόμαστε από μία πηγή ήχου, τόσο μειώνεται η έντασή του. Επίσης, μπορούμε και αντιλαμβανόμαστε την κατεύθυνση από την οποία προέρχεται ένας ήχος. Στο Reality Factory υπάρχει η ανάλογη οντότητα που μπορεί να χειριστεί τέτοιες καταστάσεις, η AudioSource3D. Όπως κάθε οντότητα, την εισάγουμε από το πάνελ εντολών και μετά την επιλογή της οι ιδιότητές της φαίνονται στο πάνελ ιδιοτήτων:

Πεδίο	Περιγραφή
bLoopSound	Αν δώσουμε την τιμή TRUE, τότε ο ήχος θα επαναλαμβάνεται αένα.
BoneName	Το όνομα του κοκάλου ενός actor στο οποίο συνδέεται η AudioSource3D.
EntityName	Το όνομα της οντότητας με την οποία είναι συνδεδεμένη η AudioSource3D. Αν η οντότητα κινείται, τότε κινείται μαζί της και η οντότητα AudioSource3D, συνεπώς κινείται και ο ήχος.
FRadius	Η μέγιστη ακτίνα που ακούγεται ο ήχος (σε texels). Μετά από το σημείο αυτό ο ήχος παύει να ακούγεται. Σταδιακά, καθώς απομακρυνόμαστε από το σημείο που είναι τοποθετημένη η οντότητα, ο ήχος μειώνεται σε ένταση.
Model	Το όνομα του μοντέλου με το οποίο είναι συνδεδεμένη η οντότητα AudioSource3D. Όπως και με τις άλλες οντότητες, αν το μοντέλο κινείται, κινείται και ο ήχος.
Origin	Η θέση της οντότητας. Η τιμή συμπληρώνεται αυτόματα.
szEntityName	Το όνομα της συγκεκριμένης οντότητας AudioSource3D. Αυτό το όνομα θα χρησιμοποιηθεί σε πεδία ιδιοτήτων άλλων οντοτήτων που θέλουμε να συνδέονται με αυτή.
szSoundFile	Ποιο αρχείο ήχου με κατάληξη .wav, .mp3 ή .ogg θα εκτελείται. Το αρχείο πρέπει να είναι αποθηκευμένο στο φάκελο C:\RealityFactory\Media\Audio.
TriggerName	Το όνομα της σκανδάλης που ενεργοποιεί τη συγκεκριμένη οντότητα AudioSource3D.

14.2.2 Οι οντότητες SoundtrackToggle και StreamingAudioProxy

Με την οντότητα PlayerStart μπορούμε να καθορίσουμε μόνο ένα μουσικό κομμάτι που θα παίζει συνεχώς ανεξάρτητα με τα γεγονότα που συμβαίνουν στην εφαρμογή μας. Είναι φανερό ότι κάτι τέτοιο δε μας εξυπηρετεί όταν θέλουμε το μουσικό κομμάτι να αλλάζει. Για να επιτύχουμε κάτι τέτοιο, χρησιμοποιούμε την οντότητα SoundtrackToggle. Τοποθετώντας οντότητες SoundtrackToggle σε καίρια σημεία του επιπέδου μας, με το πέρασμα του χρήστη απ' αυτά, αλλάζουμε και το μουσικό κομμάτι που παίζει.

Επειδή ακριβώς η ενεργοποίηση της οντότητας στηρίζεται στο πέρασμα του χρήστη μέσα από το χώρο της, πρέπει να προσέξουμε τόσο το σημείο που την τοποθετούμε όσο και το μέγεθος που θα έχει η ακτίνα δράσης της (από το αντίστοιχο πεδίο στις ιδιότητές της). Αφού την τοποθετήσουμε στην εφαρμογή μας και τη μετακινήσουμε στη θέση της, παραμετροποιούμε τις ιδιότητές της από το πάνελ ιδιοτήτων της:

Πεδίο	Περιγραφή
bCDLoops	Αν δώσουμε την τιμή TRUE, τότε ο ήχος από το CD θα επαναλαμβάνεται αέναα.
bMIDI Loops	Αν δώσουμε την τιμή TRUE, τότε ο ήχος MIDI θα επαναλαμβάνεται αέναα.
bOneShot	Αν δώσουμε την τιμή TRUE, τότε η οντότητα θα ενεργοποιηθεί μόνο μία φορά (η εναλλαγή από τον έναν ήχο στον άλλο θα γίνει μόνο μία φορά).
bStreamLoops	Αν δώσουμε την τιμή TRUE, τότε ο ήχος wave ή mp3 θα επαναλαμβάνεται αέναα.
CDTrackOne	Το όνομα του 1 ^{ου} κομματιού από το CD.
Origin	Η θέση της οντότητας. Η τιμή συμπληρώνεται αυτόματα.
Range	Η ακτίνα του χώρου σε texels με κέντρο τη SoundtrackToggle μέσα στον οποίο πρέπει να βρεθεί ο χρήστης ώστε να ενεργοποιηθεί η οντότητα.
SleepTime	Η χρονική καθυστέρηση σε δευτερόλεπτα. Πριν διανυθεί ο χρόνος αυτός η οντότητα SoundtrackToggle δεν μπορεί να ενεργοποιηθεί ξανά.
szMIDIFileOne	Το όνομα του 1 ^{ου} MIDI (αρχείο με κατάληξη .mid) κομματιού.
szStreamFileOne	Το όνομα του 1 ^{ου} κομματιού wave, ogg ή mp3.

Παρόμοια με τη SoundtrackToggle είναι η οντότητα StreamingAudioProxy. Η οντότητα όμως αυτή πρέπει να χρησιμοποιείται με προσοχή, γιατί δεν μπορεί να σταματήσει το προηγούμενο μουσικό κομμάτι, δημιουργώντας τον κίνδυνο να παίζουν ταυτόχρονα

δύο ή περισσότερα κομμάτια. Από την άλλη όμως πλευρά, με τη StreamingAudioProxy δε φορτώνεται όλο το κομμάτι στην προσωρινή μνήμη, με αποτέλεσμα να γίνεται καλύτερη διαχείριση των πόρων του υλικού του τελικού χρήστη. Όπως και η SoundtrackToggle, έτσι και η StreamingAudioProxy μπορεί να διαχειριστεί αρχεία με κατάληξη .wav, .mp3 .και .ogg. Από το πάνελ ιδιοτήτων της μπορούμε να μεταβάλουμε τις εξής παραμέτρους:

Πεδίο	Περιγραφή
bLoops	Αν δώσουμε την τιμή TRUE, τότε ο ήχος θα επαναλαμβάνεται αέναα.
bOneShot	Αν δώσουμε την τιμή TRUE, τότε η οντότητα θα ενεργοποιηθεί μόνο μία φορά.
Origin	Η θέση της οντότητας. Η τιμή συμπληρώνεται αυτόματα.
Range	Η ακτίνα του χώρου σε texels με κέντρο τη StreamingAudioProxy μέσα στον οποίο πρέπει να βρεθεί ο χρήστης ώστε να ενεργοποιηθεί η οντότητα.
SleepTime	Η χρονική καθυστέρηση σε δευτερόλεπτα. Πριν διανυθεί ο χρόνος αυτός, η οντότητα StreamingAudioProxy δεν μπορεί να ενεργοποιηθεί ξανά.

Τόσο η SoundtrackToggle όσο και η StreamingAudioProxy παρουσιάζουν κάποια προβλήματα στη χρήση τους, που οφείλονται κυρίως στις πολλές και διαφορετικές μεθόδους κωδικοποίησης αρχείων ήχου. Για μικρά σε μέγεθος αρχεία ήχου, σαν αυτά που χρησιμοποιούμε όχι για μουσική αλλά για ηχητικά εφέ, μπορούμε να χρησιμοποιούμε αρχεία Wave (με κατάληξη .wav). Τέτοια αρχεία αναθέτουμε συνήθως στην οντότητα AudioSource3D. Για μεγάλα σε μέγεθος αρχεία ήχου (συνήθως μουσική υπόκρουση) συνιστούμε τη χρήση αρχείων Ogg Vorbis (αρχεία με κατάληξη .ogg).

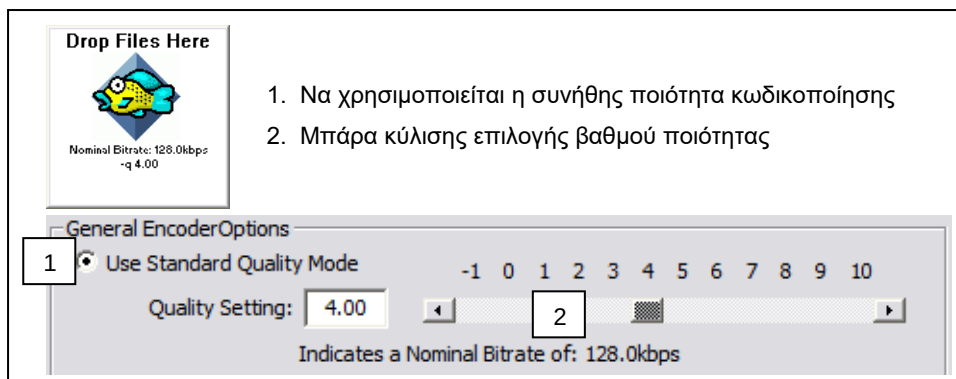
szStreamFile Το όνομα του αρχείου –με κατάληξη .wav, .mp3 ή .ogg– που θα παίζει.

14.2.3 Κατασκευάζοντας αρχεία ήχου Ogg Vorbis για χρήση στο Reality Factory

Τα αρχεία με κατάληξη .ogg είναι αρχεία ήχου συμπιεσμένα με βάση το πρότυπο Ogg Vorbis. Το πρότυπο αυτό πετυχαίνει υψηλό βαθμό συμπίεσης, αντίστοιχο με το πολύ γνωστό πρότυπο MP3, με τη βασική διαφορά ότι είναι ανοιχτού κώδικα και κατά συνέπεια εντελώς δωρεάν. Υπάρχουν αρκετά προγράμματα μετατροπής αρχείων ήχου

στο πρότυπο Ogg Vorbis (τα περισσότερα προγράμματα επεξεργασίας ήχου προσφέρουν αυτή τη δυνατότητα), στο σημείο όμως αυτό θα ασχοληθούμε με το πιο απλό, το OggdropXPd, διαθέσιμο στη διεύθυνση <http://www.rarewares.org/ogg.html>. Το πρόγραμμα είναι συμπίεμένο σε μορφή .zip (ανοίγει, για παράδειγμα, με το winzip ή το winrar). Αφού το “κατεβάσουμε” και το αποσυμπιέσουμε σε κάποιο σημείο του σκληρού μας δίσκου, αρκεί να κάνουμε διπλό κλικ στο μοναδικό αρχείο (oggdropXPd.exe) για να εκτελεστεί το πρόγραμμα, που έχει τη μορφή ενός πολύ μικρού παραθύρου το οποίο μένει πάντα σε πρώτο πλάνο (εικόνα 14.11).

Από την “Εξερεύνηση των Windows” αναζητούμε το/τα αρχεία ήχου που θέλουμε να μετατραπούν σε .ogg, τα σέρνουμε και τα αφήνουμε στο παράθυρο του OggdropXPd και αυτά μετατρέπονται και αποθηκεύονται με το ίδιο όνομα (αλλά με κατάληξη .ogg) στον ίδιο φάκελο. Η μόνη ρύθμιση που πρέπει να κάνουμε, πριν από την πρώτη μετατροπή μας, είναι να ορίσουμε την ποιότητα κωδικοποίησης. Κάνουμε δεξί κλικ στο παράθυρο του OggdropXPd και από το μενού που ανοίγει επιλέγουμε Encoding Options (η πρώτη επιλογή). Στο νέο παράθυρο που ανοίγει κάνουμε κλικ στην επιλογή Use Standard Quality Mode και, αν το επιθυμούμε, σέρνουμε την μπάρα κύλισης αυξομειώνοντας την ποιότητα (εικόνα 14.12). Εννοείται βέβαια ότι τα παραγόμενα αρχεία τα μετακινούμε στο φάκελο C:\RealityFactory\Media\Audio, για να μπορεί να τα “δει” το Reality Factory.



Εικόνα 14.11 Το πρόγραμμα OggdropXPd

Κεφάλαιο 15°

Οντότητες χειρισμού τρισδιάστατων μοντέλων

Στα προηγούμενα κεφάλαια ασχοληθήκαμε κυρίως με τα στερεά-brushes, που ο ρόλος τους είναι να καθορίσουν τη βασική γεωμετρία της εφαρμογής μας (δωμάτια, διάδρομοι, ανοιχτοί χώροι, κτλ.) και με τις ιδιότητες που μπορούμε να προσδώσουμε σε αυτά. Όμως το κυρίαρχο στοιχείο σε μία εφαρμογή είναι τα τρισδιάστατα μοντέλα, οι actors, την κατασκευή των οποίων περιγράψαμε στο 10^ο κεφάλαιο. Ο χειρισμός των τρισδιάστατων μοντέλων γίνεται με τη χρήση ειδικών οντοτήτων που θα περιγράψουμε στη συνέχεια. Να σημειώσουμε ότι το Reality Factory περιλαμβάνει μερικά μόνο βασικά τρισδιάστατα μοντέλα, πράγμα που σημαίνει ότι ο ενδιαφερόμενος θα πρέπει να αναζητήσει ή να κατασκευάσει μόνος εκείνα που θα του χρειαστούν κατά την ανάπτυξη της εφαρμογής του.

15.1 Η οντότητα StaticMesh

Η πρώτη οντότητα που θα παρουσιάσουμε είναι η StaticMesh. Είναι η πιο απλή από τις οντότητες που μπορούμε να μεταχειριστούμε για τον έλεγχο ενός actor, αλλά δίνει και τις λιγότερες δυνατότητες. Ας υποθέσουμε ότι στην εφαρμογή μας θέλουμε να προσθέσουμε μερικά δένδρα. Επιλέγουμε να χρησιμοποιήσουμε τη StaticMesh, γιατί η μόνη αλληλεπίδραση που μπορεί να έχει ο χρήστης μ' ένα δένδρο είναι απλώς η σύγκρουσή του με αυτό. Ακριβώς αυτό είναι και το πιο σημαντικό στοιχείο που πρέπει να ληφθεί υπόψη με τη χρήση της StaticMesh. Κανένα γεγονός δεν μπορεί να ενεργοποιηθεί άμεσα απ' αυτή (για παράδειγμα μετά από τη σύγκρουση του χρήστη). Απλώς μπορούμε να ορίσουμε αν ο χρήστης μπορεί ή δεν μπορεί να περνά μέσα από μία StaticMesh.

Όπως κάθε οντότητα, την εισάγουμε από το πάνελ εντολών και την καρτέλα Πρότυπα-Templates, την επιλέγουμε από τη λίστα οντοτήτων της καρτέλας Ομάδες-Groups του πάνελ των εντολών και παραμετροποιούμε τις ιδιότητές της από το πάνελ ιδιοτήτων στα δεξιά της οθόνης μας. Οι ιδιότητες της StaticMesh είναι αναλυτικά:

Πεδίο	Περιγραφή
ActorRotation	Οι μοίρες (στους άξονες x, y,z) που πρέπει να περιστραφεί το μοντέλο ώστε να τοποθετηθεί σωστά στο χώρο.
Alpha	Ο βαθμός διαφάνειας του μοντέλου (0-255). Η τιμή 0 σημαίνει απόλυτα διαφανές, ενώ η τιμή 255 απόλυτα αδιαφανές.
AmbientColor	Το χρώμα του περιβάλλοντος φωτισμού του μοντέλου. Με διπλό κλικ ανοίγει πάνελ επιλογής χρωμάτων. Ενεργοποιείται μόνο αν δώσουμε την τιμή TRUE σ' ένα από τα πεδία Comp-Light, CompLightSmooth ή CompSunLight.

Πεδίο	Περιγραφή
Backfaced	Αν δώσουμε την τιμή TRUE, τότε το Reality Factory θα σχεδιάζει και τα πίσω πολύγωνα (αυτά που κανονικά δεν μπορεί να δει ο χρήστης). Είναι χρήσιμο να δίνουμε την τιμή TRUE όταν το τρισδιάστατο μοντέλο που χρησιμοποιούμε δε σχεδιάζεται σωστά.
BoneName	Το όνομα του κοκάλου ενός actor στο οποίο μπορούμε να συνδέσουμε τη StaticMesh.
ColCheckLevel	Η τιμή που θα δώσουμε στο πεδίο αυτό, καθορίζει αν και με τι τρόπο το Reality Factory θα ελέγχει τις συγκρούσεις του χρήστη με τη StaticMesh. Η τιμή 0 ορίζει ότι δε θα γίνεται αυτός ο έλεγχος (ο χρήστης θα περνά μέσα από το μοντέλο). Με την τιμή 1 ο έλεγχος θα γίνεται με βάση τα AABB (Axis Aligned Bounding Box, κύβοι σύγκρουσης) του χρήστη και της οντότητας. Με την τιμή 2 ο έλεγχος θα γίνεται με βάση τα AABB του χρήστη και της οντότητας και μετά με βάση τις έδρες της StaticMesh και τα OBB (Oriented Bounding Boxes) του χρήστη. Με την τιμή 3 ο έλεγχος θα γίνεται μόνο με βάση τις έδρες της StaticMesh και τα OBB του χρήστη. Τέλος, με την τιμή 4 ο έλεγχος θα γίνεται με βάση την έδρα της StaticMesh και την ακτίνα που ξεκινά από τη θέση του χρήστη. Συνήθως χρησιμοποιούμε τις τιμές 1 ή 2.
CompLight	Αν δώσουμε την τιμή TRUE, τότε θα υπολογίζεται το χρώμα του περιβάλλοντος φωτισμού (από το πεδίο AmbientColor).
CompLightSmooth	Αν δώσουμε την τιμή TRUE, τότε ο περιβάλλον φωτισμός θα υπολογίζεται σε κάθε πολύγωνο του actor. Αν δώσουμε την τιμή FALSE, ο φωτισμός θα υπολογίζεται σε κάθε έδρα. Η πρώτη επιλογή δίνει καλύτερης ποιότητας φωτισμό, αλλά επιβαρύνει την εφαρμογή.
CompSunLight	Αν δώσουμε την τιμή TRUE, τότε θα υπολογίζεται και η επίδραση του φωτός από την οντότητα SunLight, που πιθανώς να έχουμε χρησιμοποιήσει.
EntityName	Το όνομα της οντότητας με την οποία είναι συνδεδεμένη η StaticMesh.
FillColor	Το χρώμα γεμίσματος του actor.
Model	Το όνομα του μοντέλου με το οποίο είναι συνδεδεμένη η StaticMesh.

Πεδίο	Περιγραφή
Origin	Η θέση της οντότητας. Η τιμή συμπληρώνεται αυτόματα.
Scale	Η κλίμακα του actor. Τιμές κάτω από 1 τον κάνουν μικρότερο και τιμές πάνω από 1 μεγεθύνουν το μοντέλο.
StaticShadowCaster	Αν στην ιδιότητα CompLight έχουμε δώσει την τιμή TRUE, τότε η παρούσα StaticMesh θα μπορεί να δημιουργεί σκιά σε άλλες οντότητες StaticMesh.
szActorFile	Το όνομα του αρχείου του τρισδιάστατου μοντέλου. Οι actors βρίσκονται αποθηκευμένοι στο φάκελο C:\RealityFactory\Media\Actors. Στο συγκεκριμένο παράδειγμα θα χρησιμοποιήσουμε το αρχείο citytree1.act, που βρίσκεται στο φάκελο C:\RealityFactory\Media\Actors\props. Συνεπώς, η δήλωση που θα κάνουμε είναι props\citytree1.act.
szEntityName	Το όνομα της συγκεκριμένης οντότητας StaticMesh. Αυτό το όνομα θα χρησιμοποιηθεί σε πεδία ιδιοτήτων άλλων οντοτήτων που θέλουμε να συνδέονται με αυτή.
UseFillColor	Αν δώσουμε την τιμή TRUE, τότε θα χρησιμοποιείται το χρώμα γεμίσματος που δώσαμε στο πεδίο FillColor.
VisCheckLevel	Η τιμή που θα δώσουμε στο πεδίο αυτό καθορίζει αν και με ποιο τρόπο το Reality Factory θα ελέγχει αν φαίνεται οντότητα από τη θέση του χρήστη (αν δεν είναι ορατή, μπορεί να μην τη σχεδιάζει, πετυχαίνοντας μικρότερο φορτίο υπολογισμών). Η τιμή 0 ορίζει να μην γίνεται αυτός ο έλεγχος, η τιμή 1 ορίζει να γίνεται ο έλεγχος με βάση το αόρατο πλαίσιο που περιβάλλει την οντότητα (bounding box) και η τιμή 2 ορίζει να γίνεται ο έλεγχος με βάση κάθε πολύγωνο της οντότητας (μεγάλος αριθμός υπολογισμών). Συνιστάται η χρήση της τιμής 1.
ZSort	Αν έχουμε δώσει κάποιο βαθμό διαφάνειας στον actor (στο πεδίο Alpha), τότε το Reality Factory θα πρέπει να υπολογίζει το βάθος που βρίσκονται τα πολύγωνα του actor (ποια είναι μπροστά και ποια πίσω). Κάτι τέτοιο το πετυχαίνουμε αν δώσουμε την τιμή TRUE σ' αυτό το πεδίο.

Μόλις τοποθετήσουμε την οντότητα στην εφαρμογή μας, τη μετακινήσουμε στη θέση της και την παραμετροποιήσουμε (εικόνα 15.1), είναι πολύ πιθανό να μη βλέπουμε το τρισδιάστατο μοντέλο. Τότε πρέπει οπωσδήποτε να κάνουμε κλικ στο εργαλείο ανανέωσης . Ανανέωση πρέπει να κάνουμε και όταν αλλάζουμε την κλίμακα του μοντέ-

λου. Σε περίπτωση που παρά την ανανέωση εξακολουθούμε να μη βλέπουμε το μοντέλο, τότε δύο πράγματα μπορεί να συμβαίνουν:

1. Έχουμε δώσει λάθος όνομα ή διαδρομή αρχείου actor. Πρέπει να προσέξουμε να δώσουμε το όνομα ακριβώς όπως και το όνομα του αρχείου (πεζά και κεφαλαία γράμματα παίζουν ρόλο). Επίσης, δεν πρέπει να χρησιμοποιούμε ελληνικούς χαρακτήρες ή κενά στο όνομα, για παράδειγμα `big tree.act`.
2. Το τρισδιάστατο μοντέλο είναι τόσο μικρό, που δεν μπορούμε να το δούμε. Σ' αυτή την περίπτωση πρέπει να αυξήσουμε την τιμή της κλίμακας του.

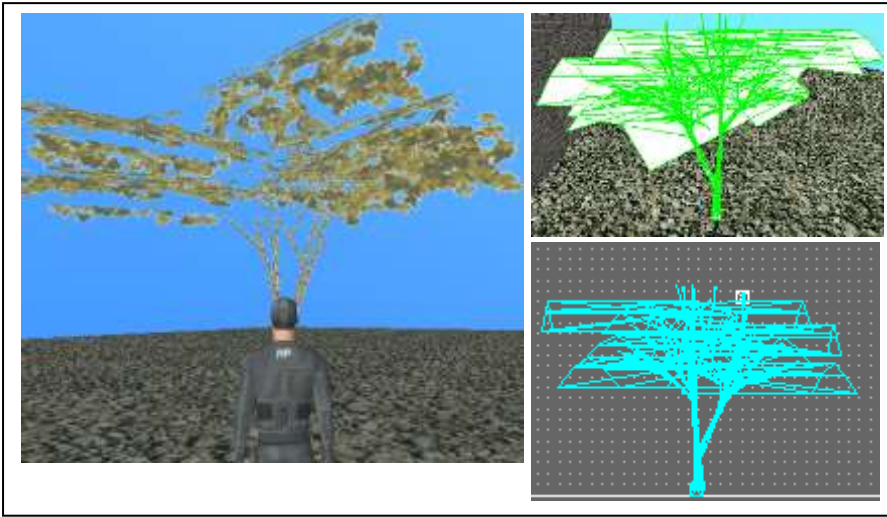
Πρέπει να προσέξουμε μην τυχόν έχουμε μεγαλώσει τόσο το τρισδιάστατο μοντέλο, ώστε μέρος του να βρίσκεται εκτός του εξωτερικού κύβου της εφαρμογής μας. Κάτι τέτοιο συνεπάγεται ότι δε θα γίνει με επιτυχία η μεταγλώττιση της εφαρμογής.

Τέλος, ιδιαίτερη προσοχή πρέπει να δίνουμε στον αριθμό των actors που εισάγουμε στην εφαρμογή μας, αλλά κυρίως στο μέγεθος του αρχείου του κάθε actor. Στο παράδειγμά μας έχουμε εισαγάγει ένα δέντρο που το μέγεθος του αρχείου του είναι περίπου 7Mb, που θεωρείται εξαιρετικά μεγάλο. Αν εισαγάγουμε αρκετά τέτοια δένδρα, τότε είναι σχεδόν βέβαιο ότι θα πέσει σημαντικά ο αριθμός των καρέ/δευτερόλεπτο που αποδίδει η εφαρμογή μας.

Ενδεικτική παραμετροποίηση μίας StaticMesh:

Πεδίο	Τιμή	Πεδίο	Τιμή
Backfaced	true	Scale	1.5
ColCheckLevel	2	szActorFile	props\citytree1.act*
CompLight	true	VisCheckLevel	1
CompLightSmooth	false	Zsort	false
CompSunLight	false		

* Το αρχείο περιλαμβάνεται στην εγκατάσταση του Reality Factory



Εικόνα 15.1 Τοποθετώντας μία StaticMesh

15.2 Η οντότητα StaticEntityProxy

Ο δεύτερος τρόπος να εισαγάγουμε ένα τρισδιάστατο μοντέλο στην εφαρμογή μας είναι μέσω της οντότητας StaticEntityProxy. Η οντότητα αυτή είναι κατά πολύ πιο σύνθετη από τη StaticMesh, γιατί μπορούμε να μεταβάλουμε ένα μεγάλο αριθμό ιδιοτήτων της. Για τις ανάγκες του παραδείγματός μας, θα εισαγάγουμε ένα μοντέλο ανθρώπου (εικόνα 15.2). Κατά τα γνωστά, εισάγουμε την οντότητα, τη μετακινούμε στη θέση της, την επιλέγουμε και προσαρμόζουμε τις ιδιότητές της, που είναι:

Πεδίο	Περιγραφή
ActorRotation	Οι μοίρες (στους άξονες x, y, z) που πρέπει να περιστραφεί το μοντέλο-actor ώστε να τοποθετηθεί σωστά στο χώρο.
AllMaterial	Αν δώσουμε την τιμή TRUE, τότε να εφαρμοστεί η τεχνική Environmental Mapping σ' όλες τις υφές του actor (η τεχνική εξηγείται στη συνέχεια, στο πεδίο EnvironmentalMapping).
AmbientColor	Το χρώμα του περιβάλλοντος φωτισμού του μοντέλου. Με διπλό κλικ ανοίγει πάνελ επιλογής χρωμάτων.
AmbientLightFromFloor	Αν δώσουμε την τιμή TRUE, ο φωτισμός θα προέρχεται από το πάτωμα και όχι από το χρώμα που ορίσαμε στο πεδίο AmbientColor.
AttributeAmt	Πόση είναι η "υγεία" του actor. Αν δώσουμε την τιμή -1, δεν μπορεί να καταστραφεί.
bCanPush	Αν δώσουμε την τιμή TRUE, τότε το μοντέλο μπορεί να μετακινηθεί από το χρήστη.

Πεδίο	Περιγραφή
BoxSize	Οι τιμές σε texels που δίνουμε στους άξονες x, y και z που προσδιορίζουν το μέγεθος του κύβου υπολογισμού συγκρούσεων που περιβάλλει κάθε τρισδιάστατο μοντέλο. Αν δώσουμε τις τιμές 0,0,0, τότε ο κύβος αυτός έχει το προκαθορισμένο μέγεθος. Το πεδίο αυτό το χρησιμοποιούμε είτε όταν δεν υπάρχει κύβος υπολογισμού συγκρούσεων είτε όταν θέλουμε να αλλάξουμε τις διαστάσεις του.
ChangeMaterial	Όταν κατασκευάζουμε το τρισδιάστατο μοντέλο, μπορούμε να ορίσουμε διαφορετικά υλικά-υφές που το “ντύνουν”. Όταν το εξάγουμε σε μορφή .BDY (κεφάλαιο 10.1), μπορούμε να εξάγουμε και αυτά τα υλικά τα οποία υπάρχουν και κατά τη σύνθεση του actor με το Actor Studio (το οποίο πάντα αναθέτει το πρώτο υλικό στη σειρά). Στο συγκεκριμένο πεδίο μπορούμε να ορίσουμε ένα διαφορετικό από το προεπιλεγμένο υλικό που θα “ντύνει” τον actor, δίνοντας το όνομα του υλικού (στην ουσία το όνομα της υφής).
Damage	Η τιμή ορίζει το μέγεθος της “βλάβης” που μπορεί να προκαλέσει το μοντέλο στο χρήστη και ο χρήστης στο μοντέλο.
DamageAlt	Η τιμή ορίζει το μέγεθος της εναλλακτικής “βλάβης” που μπορεί να προκαλέσει το μοντέλο στο χρήστη και το αντίστροφο.
DamageAttribute	Στο πεδίο αυτό εισάγουμε το όνομα της πρωτεύουσας παραμέτρου “υγείας” του actor (health, stamina, oxygen) που θα βλάπτεται από το χρήστη. Αν αφήσουμε το πεδίο κενό, η προεπιλεγμένη παράμετρος είναι η health. Συνήθως αφήνουμε κενό το πεδίο, γιατί οι actors έχουν μόνο την παράμετρο health.
DamageDelay	Πόση χρονική καθυστέρηση σε δευτερόλεπτα θα υπάρξει πριν ο actor αρχίσει να “βλάπτει” το χρήστη και το αντίστροφο. Δίνουμε την τιμή 0 για άμεση επίδραση.
DamageTo	Στο πεδίο εισάγουμε το όνομα της πρωτεύουσας παραμέτρου “υγείας” του χρήστη (health, stamina, oxygen) που θα βλάπτεται από τον actor. Αν αφήσουμε το πεδίο κενό, η προεπιλεγμένη παράμετρος είναι η health.

Πεδίο	Περιγραφή
DamageToAlt	Εισάγουμε το όνομα της δευτερεύουσας παραμέτρου “υγείας” του χρήστη (health, stamina, oxygen) που θα βλάπτεται από τον actor.
DeathDissappear	Αν δώσουμε την τιμή TRUE, τότε ο actor θα εξαφανιστεί μετά το “θάνατό” του.
DoDamage	Αν δώσουμε την τιμή TRUE, τότε ο actor θα μπορεί να βλάπτει το χρήστη όταν έρχονται σε επαφή. Το πόσο το ορίσαμε στο πεδίο Damage, το πού το ορίσαμε στα πεδία DamageTo και DamageToAlt.
EnvironmentalMapping	Αν δώσουμε την τιμή TRUE, τότε στον actor θα χρησιμοποιηθεί η τεχνική της δημιουργίας αντανakλάσεων επάνω στην επιφάνειά του με τη χρήση υφών (Environmental Mapping). Προϋπόθεση είναι βέβαια να έχουμε δημιουργήσει τέτοιες υφές που να έχουν “ντύσει” τον actor.
FaceDestination	Αν δώσουμε την τιμή TRUE, τότε ο actor θα περιστραφεί για να κοιτά προς τον προορισμό του.
FadeOut	Αν δώσουμε την τιμή TRUE και έχουμε δώσει επίσης την τιμή TRUE στο πεδίο DeathDissappear, τότε ο actor θα σβήνει αργά-αργά μέχρι να εξαφανιστεί.
FadeOutTime	Ο χρόνος, σε δευτερόλεπτα, που θα χρειαστούν για να εξαφανιστεί ο actor που έχει “πεθάνει”.
FillColor	Το χρώμα γεμίσματος του actor. Με διπλό κλικ ανοίγει πάνελ επιλογής χρωμάτων.
FRadius	Η ακτίνα μέχρι την οποία ακούγεται ο ήχος που έχουμε ορίσει στο πεδίο szSoundFile.
GetDamaged	Αν δώσουμε την τιμή TRUE, τότε ο actor θα βλάπτεται κατά την επαφή του με το χρήστη. Το πόσο το ορίσαμε στο πεδίο Damage. Για να μπορεί ο χρήστης να βλάψει τον actor, θα πρέπει να έχουμε δώσει την τιμή TRUE <u>και</u> στο πεδίο DoDamage (να βλάπτεται ο χρήστης από τον actor). Μόνο όταν και τα δύο αυτά πεδία έχουν την τιμή TRUE βλάπτεται ο actor (χωρίς μάλιστα να βλάπτει το χρήστη).
HideFromRadar	Αν δώσουμε την τιμή TRUE, τότε ο actor που είναι συνδεδεμένος με την οντότητα StaticEntityProxy δε θα εμφανίζεται στο ραντάρ του HUD.

Πεδίο	Περιγραφή
InitialAlpha	Ο αρχικός βαθμός διαφάνειας του actor. Ισχύει μόνο όταν είναι στατικό μοντέλο (δεν έχουμε ορίσει κάποιο animation στο πεδίο szDefaultAction).
IsAVehicle	Αν δώσουμε την τιμή TRUE, τότε ο actor θα μπορεί να χρησιμοποιηθεί σαν όχημα (χρειάζεται να υπάρχει αρχείο δέσμης εντολών που να ορίζει τη συμπεριφορά των οχημάτων).
NoCollision	Αν δώσουμε την τιμή TRUE, τότε δε θα υπολογίζονται οι συγκρούσεις του μοντέλου με άλλα μοντέλα, αλλά και ούτε με το χρήστη.
Origin	Η θέση της οντότητας. Η τιμή συμπληρώνεται αυτόματα.
PercentMapping	Το ποσοστό του Environmental Mapping που θα χρησιμοποιηθεί για τον actor.
PercentMaterial	Το ποσοστό του υλικού του Environmental Mapping που θα χρησιμοποιηθεί για τον actor.
ScaleFactor	Η κλίμακα του actor. Τιμές κάτω από 1 τον κάνουν μικρότερο και τιμές πάνω από 1 μεγεθύνουν το μοντέλο.
ShadowAlpha	Ο βαθμός διαφάνειας της σκιάς του μοντέλου.
ShadowAlphamap	Το όνομα του αρχείου εικόνας που θα χρησιμοποιηθεί για το διαφανές τμήμα της σκιάς του actor. Όπως όλα τα αρχεία εικόνων, πρέπει να είναι bitmap (με κατάληξη .bmp) και αποθηκευμένο στο φάκελο C:\RealityFactory\Media\Bitmaps ή σε υπο-φάκελο αυτού.
ShadowBitmap	Το όνομα του αρχείου εικόνας που θα χρησιμοποιηθεί για τη σκιά του actor.
ShadowSize	Η ακτίνα σε texels της σκιάς. Η τιμή 0 υποδηλώνει καθόλου σκιά.
SoundLoop	Αν δώσουμε την τιμή TRUE, τότε ο ήχος που δηλώνουμε στο πεδίο szSoundFile θα παίζει αέναα.
StepHeight	Το ύψος σε texels που μπορεί να ανεβεί ο actor μ' ένα βήμα.
SubjectToGravity	Αν δώσουμε την τιμή TRUE, τότε ο actor θα υπόκειται στη δύναμη της βαρύτητας.
szActorFile	Το όνομα του αρχείου του τρισδιάστατου μοντέλου. Ισχύουν τα ίδια με την οντότητα StaticMesh.
szDeathAction	Ποιο θα είναι το animation του actor όταν “πεθάνει”. Ισχύει ό,τι και στο επόμενο πεδίο szDefaultAction.

Πεδίο	Περιγραφή
szDefaultAction	Ποιο θα είναι το προεπιλεγμένο animation του actor. Το όνομα πρέπει να δοθεί ακριβώς όπως το ορίσαμε κατασκευάζοντας τον actor με το Actor Studio. Αν δεν το θυμόμαστε, μπορούμε να εκκινήσουμε το Actor Studio, να φορτώσουμε το συγκεκριμένο actor και να δούμε όλα τα animation από το αντίστοιχο μενού (εικόνα 10.31, κεφάλαιο 10.5).
szEntityName	Το όνομα της συγκεκριμένης οντότητας StaticEntityProxy. Αυτό το όνομα θα χρησιμοποιηθεί σε πεδία ιδιοτήτων άλλων οντοτήτων που θέλουμε να συνδέονται με αυτή.
szImpactAction	Ποιο animation θα εκτελείται όταν ο χρήστης συγκρουστεί με τον actor.
szSoundFile	Ποιο αρχείο ήχου θα εκτελείται όταν ο χρήστης συγκρουστεί με τον actor.
UseKey	Αν δώσουμε την τιμή TRUE, τότε ο actor θα ενεργοποιείται όταν ο χρήστης πατήσει το πλήκτρο χρήσης (το E είναι το προεπιλεγμένο).
UseProjectedShadows	Αν δώσουμε την τιμή TRUE, τότε ο actor θα δημιουργεί προβολικές σκιές. Οι προβολικές σκιές δε χρειάζονται αρχεία εικόνων για να δημιουργηθούν.
UseStencilShadows	Αν δώσουμε την τιμή TRUE, τότε ο actor θα δημιουργεί Stencil σκιές. Οι Stencil σκιές χρειάζονται αρχεία εικόνων για να δημιουργηθούν (από τα πεδία ShadowAlphamap και ShadowBitmap).

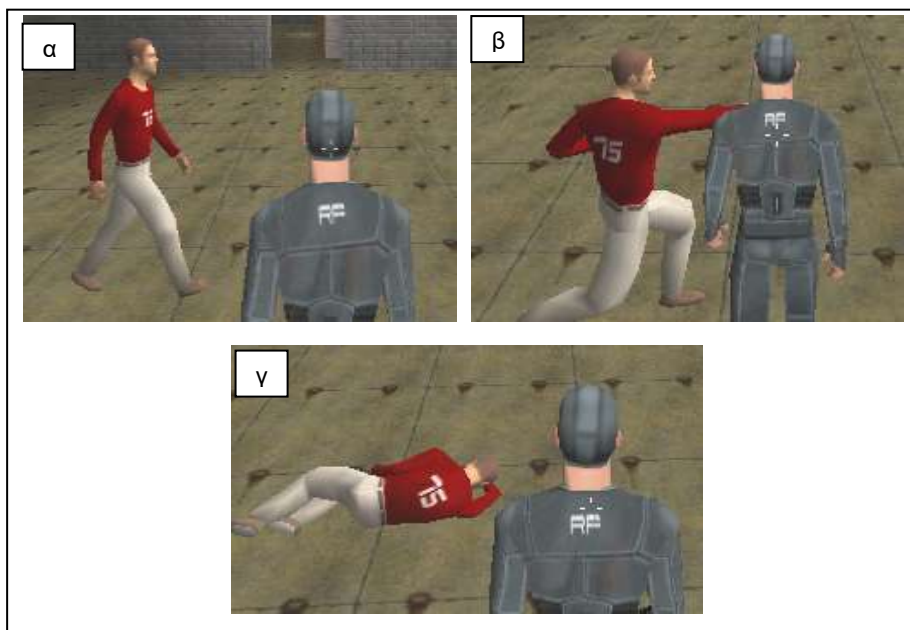
Από την παράθεση των ιδιοτήτων της οντότητας, μπορούμε να συμπεράνουμε ότι η οντότητα StaticEntityProxy σε σύγκριση με τη StaticMesh προσθέτει τα εξής επιπλέον στοιχεία:

- Τη δυνατότητα να βλάπτει ή να βλάπτεται από το χρήστη.
- Τη δυνατότητα να παίζει κάποιος ήχος και να προσδιορίζεται η εμβέλειά του.
- Τη δυνατότητα να μετακινείται ο actor που ελέγχεται από τη StaticEntityProxy όταν συγκρούεται με το χρήστη.
- Το animation και την εναλλαγή animation.
- Τη δυνατότητα να κινείται ο actor που ελέγχεται από τη StaticEntityProxy. Το θέμα αυτό θα το αναλύσουμε στο επόμενο κεφάλαιο, στο σημείο αυτό όμως επισημαίνουμε ότι με τη StaticMesh δεν μπορούμε να κάνουμε κάτι τέτοιο με άμεσο τρόπο.

Ενδεικτική παραμετροποίηση μίας StaticEntityProxy

Πεδίο	Τιμή	Πεδίο	Τιμή
ActorRotation	-90,0,90	FRadius	200
AttributeAmt	100	GetDamaged	TRUE
BCanPush	TRUE	NoCollision	FALSE
BoxSize	0 0 0	ScaleFactor	1.4
Damage	10	SoundLoop	TRUE
DamageAttribute	-	StepHeight	30
DamageDelay	1	SubjectToGravity	TRUE
DamageTo	-	szActorFile	Virgil_red.act*
DeathDissappear	TRUE	szDeathAction	Die
DoDamage	TRUE	szImpactAction	Punch
FadeOut	TRUE	UseProjectedShadows	TRUE
FadeOutTime	5		

* Το αρχείο περιλαμβάνεται στην εγκατάσταση του Reality Factory



Εικόνα 15.2 Τοποθετώντας μία StaticEntityProxy
α) Αρχικό animation, β) Animation μετά από σύγκρουση, γ) Animation “θανάτου”

15.3 Η οντότητα Attribute

Είδαμε ότι η οντότητα StaticEntityProxy έχει τη δυνατότητα να προκαλεί βλάβη στο χρήστη. Είναι λοιπόν λογικό να υπάρχει και μία οντότητα που θα ωφελεί το χρήστη αυξάνοντας κάποια από τις παραμέτρους υγείας του. Αυτή η οντότητα είναι η Attribute (εικόνα 15.3). Η Attribute γενικότερα χρησιμοποιείται για κάθε αντικείμενο που ο χρήστης μπορεί να “μαζέψει” από το περιβάλλον για να το χρησιμοποιήσει άμεσα, έπειτα από κάποιο χρονικό διάστημα ή σε κάποιο άλλο σημείο της εφαρμογής.

Πρέπει να επισημάνουμε πως, παρά το γεγονός ότι το Reality Factory είναι κυρίως πρόγραμμα κατασκευής παιχνιδιών, η εκτενής παραμετροποίηση της κάθε οντότητας, ο ανοιχτός του σχεδιασμός και η δυνατότητα να παρέμβουμε στα αρχεία του μας επιτρέπει να ξεφύγουμε από τα όρια του παιχνιδιού. Έτσι, με επέμβαση στα κατάλληλα αρχεία μπορούμε να γενικεύσουμε τη χρήση της Attribute, όπως θα δούμε στο 19^ο κεφάλαιο, όπου αναλύουμε ειδικά θέματα.

Οι ιδιότητες της Attribute που μπορούν να παραμετροποιηθούν είναι:

Πεδίο	Περιγραφή
ActorRotation	Οι μοίρες (στους άξονες x, y,z) που πρέπει να περιστραφεί το μοντέλο-actor ώστε να τοποθετηθεί σωστά στο χώρο.
AllMaterial	Αν δώσουμε την τιμή TRUE, τότε θα εφαρμοστεί η τεχνική Environmental Mapping σ' όλες τις υφές του actor.
AmbientColor	Το χρώμα του περιβάλλοντος φωτισμού του μοντέλου. Με διπλό κλικ ανοίγει πάνελ επιλογής χρωμάτων.
AmbientLightFrom-Floor	Αν δώσουμε την τιμή TRUE, ο περιβάλλον φωτισμός θα προέρχεται από το πάτωμα και όχι από το χρώμα που ορίσαμε στο πεδίο AmbientColor.
AttributeAltAmount	Η τιμή που δίνουμε στο πεδίο αυτό αυξάνει την εναλλακτική παράμετρο “υγείας” του χρήστη.
AttributeAltName	Το όνομα της εναλλακτικής παραμέτρου υγείας (health, stamina, oxygen).
AttributeAmount	Η τιμή που δίνουμε στο πεδίο αυτό αυξάνει την πρωτεύουσα παράμετρο “υγείας” του χρήστη.
AttributeName	Το όνομα της πρωτεύουσας παραμέτρου υγείας.
BoneName	Το όνομα του κοκάλου ενός actor στο οποίο μπορούμε να συνδέσουμε τη Attribute.
ChangeMaterial	Στο πεδίο αυτό μπορούμε να ορίσουμε ένα διαφορετικό από το προεπιλεγμένο υλικό που θα “ντύνει” τον actor, δίνοντας το όνομα του υλικού (όπως και στη StaticEntityProxy).

Πεδίο	Περιγραφή
Delay	Ο χρόνος σε δευτερόλεπτα που πρέπει να διανυθεί πριν η οντότητα επανεμφανιστεί.
EntityName	Το όνομα της οντότητας με την οποία είναι συνδεδεμένη η Attribute.
EnvironmentMapping	Αν δώσουμε την τιμή TRUE, τότε στον actor θα χρησιμοποιηθεί η τεχνική της δημιουργίας αντανάκλασεων επάνω στην επιφάνεια του actor με τη χρήση υφών (Environmental Mapping). Προϋπόθεση είναι βέβαια να έχουμε δημιουργήσει τέτοιες υφές που να έχουν “ντύσει” τον actor.
FillColor	Το χρώμα γεμίσματος του actor.
Gravity	Αν δώσουμε την τιμή TRUE, τότε ο actor θα υπόκειται στη δύναμη της βαρύτητας.
HideFromRadar	Αν δώσουμε την τιμή TRUE, τότε ο actor που είναι συνδεδεμένος με την οντότητα Attribute δε θα εμφανίζεται στο ραντάρ του HUD.
Model	Το όνομα του μοντέλου με το οποίο είναι συνδεδεμένη η Attribute.
Origin	Η θέση της οντότητας. Η τιμή συμπληρώνεται αυτόματα.
PercentMapping	Το ποσοστό του Environmental Mapping που θα χρησιμοποιηθεί για τον actor.
PercentMaterial	Το ποσοστό του υλικού του Environmental Mapping που θα χρησιμοποιηθεί για τον actor.
PlayerOnly	Αν δώσουμε την τιμή TRUE, τότε μόνο ο χρήστης θα μπορεί να χρησιμοποιήσει την οντότητα.
PowerUp	Αν δώσουμε την τιμή TRUE, τότε η οντότητα αυξάνει τη μέγιστη τιμή της παραμέτρου “υγείας” που ορίσαμε στο πεδίο AttributeName κατά το ποσό που ορίσαμε στο πεδίο AttributeAmount.
ReSpawn	Αν δώσουμε την τιμή TRUE, τότε η οντότητα θα επανεμφανιστεί μετά το χρόνο που ορίσαμε στο πεδίο Delay.
Scale	Η κλίμακα του actor. Τιμές κάτω από 1 τον κάνουν μικρότερο και τιμές πάνω από 1 μεγεθύνουν το μοντέλο.
SetMaximumAmount	Αν δώσουμε την τιμή TRUE, τότε η οντότητα θα αυξήσει στο μέγιστο την παράμετρο που ορίσαμε στο πεδίο AttributeName.
ShadowAlpha	Ο βαθμός διαφάνειας της σκιάς του μοντέλου.

Πεδίο	Περιγραφή
ShadowAlphamap	Το όνομα του αρχείου εικόνας που θα χρησιμοποιηθεί για το διαφανές μέρος της σκιάς του actor.
ShadowBitmap	Το όνομα του αρχείου εικόνας που θα χρησιμοποιηθεί για τη σκιά του actor.
ShadowSize	Η ακτίνα σε texels της σκιάς. Η τιμή 0 υποδηλώνει καθόλου σκιά.
szActorName	Το όνομα του αρχείου του τρισδιάστατου μοντέλου. Ισχύουν τα ίδια με την οντότητα StaticMesh.
szEntityName	Το όνομα της συγκεκριμένης οντότητας Attribute. Αυτό το όνομα θα χρησιμοποιηθεί σε πεδία ιδιοτήτων άλλων οντοτήτων που θέλουμε να συνδέονται με αυτή.
szReSpawnSound	Ποιο αρχείο ήχου θα εκτελείται όταν επανεμφανίζεται η οντότητα.
szSoundFile	Ποιο αρχείο ήχου θα εκτελείται όταν ο χρήστης συγκρουστεί με τον actor της Attribute.
TriggerName	Το όνομα της σκανδάλης που ενεργοποιεί τη συγκεκριμένη οντότητα Attribute.
UseKey	Αν δώσουμε την τιμή TRUE, τότε ο actor θα ενεργοποιείται όταν ο χρήστης πατήσει το πλήκτρο χρήσης (το E είναι το προεπιλεγμένο).
UseProjectedShadows	Αν δώσουμε την τιμή TRUE, τότε ο actor θα δημιουργεί προβολικές σκιές. Οι προβολικές σκιές δε χρειάζονται αρχεία εικόνων για να δημιουργηθούν.
UseStencilShadow	Αν δώσουμε την τιμή TRUE, τότε ο actor θα δημιουργεί Stencil σκιές.

Αν η παράμετρος που επιχειρεί να μεταβάλει η Attribute έχει ήδη τη μέγιστη τιμή που μπορεί να πάρει, τότε ο χρήστης δεν μπορεί να “μαζέψει” το αντικείμενο που είναι συνδεδεμένο με την οντότητα, ούτε μεταβάλλεται η τιμή της.

Μπορούμε να κάνουμε την Attribute να βλάπτει το χρήστη εάν στο πεδίο AttributeAmount δώσουμε αρνητική τιμή.

Όπως έχουμε ήδη αναφέρει, οι παράμετροι του χρήστη που μπορούν να μεταβληθούν, καθώς επίσης και οι αρχικές, μέγιστες και ελάχιστες τιμές τους, ορίζονται στο αρχείο player.ini, που βρίσκεται στο φάκελο C:\RealityFactory\Install. Στο ίδιο αρχείο προσθέτουμε και τις δικές μας παραμέτρους.

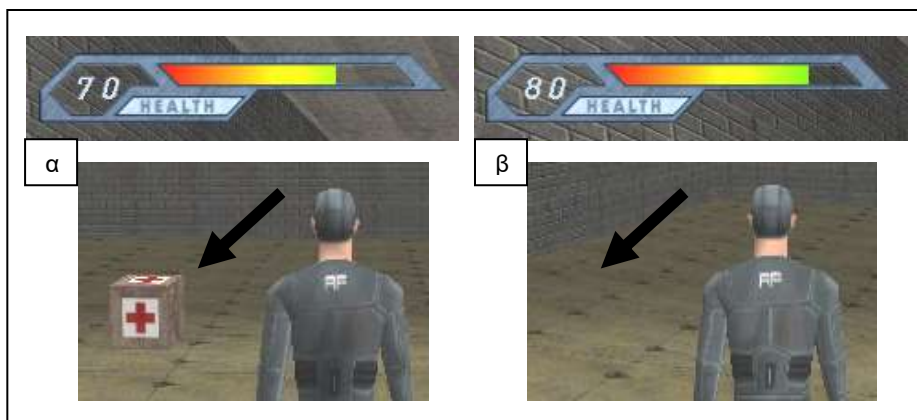
Για να δώσουμε ένα παράδειγμα χρήσης της Attribute, θα χρειαστεί σε πρώτη φάση να ανοίξουμε το αρχείο player.ini και να μειώσουμε την αρχική “υγεία” του χρήστη ως εξής:

```
[health]
initial = 70
low = 0
high = 100
```

Αποθηκεύοντας το αρχείο, το αποτέλεσμα θα είναι ο χρήστης να ξεκινά το επίπεδο με τα 70/100 της μέγιστης υγείας που μπορεί να αποκτήσει. Αυτό θα του επιτρέψει να “μαζέψει” ένα αντικείμενο που θα αυξήσει την υγεία του και θα δούμε το αποτέλεσμα στο HUD. Εναλλακτικά, θα μπορούσαμε να αυξήσουμε τη μέγιστη τιμή της υγείας του χρήστη, για παράδειγμα high=140.

Επανερχόμαστε στο RFEEditPro και κάνουμε τις εξής ρυθμίσεις στην Attribute που έχουμε εισαγάγει και τοποθετήσει σε κάποιο σημείο:

- AttributeAmount=10, για να αυξάνεται η παράμετρος που θα ορίσουμε στο πεδίο AttributeName κατά 10 μονάδες.
- AttributeName=health, που είναι η παράμετρος που θέλουμε να αυξάνεται.
- Delay=5, για να υπάρχει καθυστέρηση 5 δευτερολέπτων πριν επανεμφανιστεί ο actor.
- Gravity=FALSE, για να μην υπόκειται στη δύναμη της βαρύτητας ο actor (σε αντίθετη περίπτωση, αν τον τοποθετούσαμε σε κάποιο ύψος, θα έπεφτε στο πάτωμα).
- PlayerOnly=TRUE, για να μπορεί μόνο ο χρήστης να χρησιμοποιήσει την οντότητα.
- ReSpawn=TRUE, για να επανεμφανίζεται ο actor.
- Scale=1.5, για να μεγαλώσουμε λίγο το τρισδιάστατο μοντέλο.
- szActorName=Attributes\medkit.act, για να χρησιμοποιήσουμε το μοντέλο medkit.act, που βρίσκεται στο φάκελο C:\RealityFactory\Media\Actors\Attributes.
- szSoundFile=menu\menuclick.wav, για να χρησιμοποιήσουμε τον ήχο menuclick.wav που βρίσκεται στο φάκελο C:\RealityFactory\Media\Audio\menu.
- UseKey=FALSE, για να μπορεί ο χρήστης να πάρει την οντότητα χωρίς να χρησιμοποιήσει κάποιο πλήκτρο.
- UseProjectedShadows=TRUE, για να δημιουργεί το αντικείμενο προβολικές και όχι άλλου είδους σκιές.



Εικόνα 15.3 Η λειτουργία της οντότητας Attribute

- α) Πριν ο χρήστης ακουμπήσει το μοντέλο που συνδέεται με την Attribute (Health 70),
β) Αφού ο χρήστης έχει ακουμπήσει το μοντέλο (Health 80)

15.4 Η οντότητα Pawn

Η πιο σύνθετη οντότητα με την οποία χειριζόμαστε τρισδιάστατα μοντέλα είναι η Pawn. Τοποθετώντας μία Pawn στο επίπεδό μας, θα διαπιστώσουμε ότι έχει ένα σχετικά μικρό αριθμό παραμέτρων. Αυτό οφείλεται στο γεγονός ότι ο έλεγχός της και κατ'επέκταση ο έλεγχος του τρισδιάστατου μοντέλου-actor που είναι συνδεδεμένο με αυτή γίνεται προγραμματιστικά μέσω αρχείου δέσμης εντολών (script file). Επιπρόσθετα, πρέπει να ορίσουμε τον τύπο της οντότητας στο αρχείο Pawn.ini, που βρίσκεται στο φάκελο C:\RealityFactory\Install. Το όνομα αυτό θα το χρησιμοποιήσουμε σ' ένα από τα πεδία ιδιοτήτων της όταν την τοποθετήσουμε στο επίπεδό μας. Αναλυτικά, οι ιδιότητες της Pawn που παραμετροποιούνται από το πάνελ ιδιοτήτων είναι:

Πεδίο	Περιγραφή
Angle	Οι μοίρες (στους άξονες x, y, z) που πρέπει να περιστραφεί το μοντέλο-actor όταν εμφανιστεί για πρώτη φορά στο επίπεδο. Δε χρειάζεται να συμπληρώσουμε τις τιμές, γιατί η θέση ελέγχεται από το αρχείο pawn.ini και από αρχείο δέσμης εντολών.
ChangeMaterial	Στο πεδίο αυτό μπορούμε να ορίσουμε ένα διαφορετικό από το προεπιλεγμένο υλικό που θα "ντύνει" τον actor, δίνοντας το όνομα του υλικού.
ConvOrder	Σε κάθε τμήμα ενός διαλόγου μεταξύ της pawn και του χρήστη δίνεται και ένα όνομα. Στο πεδίο αυτό δηλώνουμε το όνομα του τμήματος εκείνου που θα εμφανιστεί όταν ξεκινά ο διάλογος. Περισσότερα για τους διαλόγους στο 19 ^ο κεφάλαιο.

Πεδίο	Περιγραφή
ConvScriptName	Το όνομα του αρχείου δέσμης εντολών που χειρίζεται το διάλογο. Το αρχείο πρέπει να βρίσκεται στο φάκελο C:\RealityFactory\Scripts και έχει πάντα την κατάληξη .s (αρχείο απλού κειμένου που ανοίγει με το “Σημειωματάριο”). Τα κείμενα του διαλόγου βρίσκονται συνήθως (εκτός αν ορίζεται διαφορετικά) στο αρχείο conversation.txt που είναι αποθηκευμένο στο φάκελο C:\RealityFactory\Install.
HideFromRadar	Αν δώσουμε την τιμή TRUE, τότε ο actor που είναι συνδεδεμένος με την οντότητα Pawn δε θα εμφανίζεται στο ραντάρ του HUD.
Origin	Η θέση της οντότητας. Η τιμή συμπληρώνεται αυτόματα.
PawnType	Ο τύπος της οντότητας Pawn, έτσι όπως τον ορίζουμε στο αρχείο Pawn.ini.
ScriptName	Το όνομα του αρχείου δέσμης εντολών που χειρίζεται τη συγκεκριμένη οντότητα Pawn. Το αρχείο πρέπει να βρίσκεται στο φάκελο C:\RealityFactory\Scripts και έχει πάντα την κατάληξη .s (αρχείο απλού κειμένου που ανοίγει με το “Σημειωματάριο”).
SpawnOrder	Οι εντολές που υπάρχουν στο αρχείο δέσμης εντολών, το οποίο ελέγχει την οντότητα Pawn, είναι ομαδοποιημένες. Σε κάθε ομάδα δίνεται και ένα όνομα. Στο πεδίο αυτό δηλώνουμε το όνομα της ομάδας που θα εκτελεστεί πρώτη, μόλις δηλαδή εμφανιστεί ο actor στο επίπεδο.
SpawnPoint	Ο actor που είναι συνδεδεμένος με τη συγκεκριμένη οντότητα Pawn μπορεί να κινηθεί σε μία καθορισμένη πορεία την οποία ορίζουν σημεία που ονομάζονται ScriptPoints (είναι και αυτά οντότητες που τις τοποθετούμε στο επίπεδο). Στο πεδίο αυτό εισάγουμε το όνομα του πρώτου σημείου (στην ουσία είναι η θέση που θα εμφανιστεί για πρώτη φορά ο actor).
SpawnTrigger	Το όνομα της σκανδάλης που ενεργοποιεί την εμφάνιση του actor στο επίπεδο. Αν δεν υπάρχει τέτοια σκανδάλη, ο actor εμφανίζεται με την έναρξη του επιπέδου.
SzEntityName	Το όνομα της συγκεκριμένης οντότητας Pawn. Αυτό το όνομα θα χρησιμοποιηθεί σε πεδία ιδιοτήτων άλλων οντοτήτων που θέλουμε να συνδέονται με αυτή.

Όπως ήδη αναφέραμε, δεν αρκεί να παραμετροποιήσουμε την οντότητα στο RFEditPro, αλλά χρειάζεται να επέμβουμε σε αρχεία. Το πρώτο απ' αυτά είναι το Pawn.ini (στο φάκελο Install), το οποίο ανοίγουμε με το “Σημειωματάριο”. Η οργάνωση του αρχείου μοιάζει αρκετά με αυτή του Player.ini. Κάθε τύπος Pawn ορίζεται από μία ομάδα

παραμέτρων που η αρχή της είναι ένα όνομα μέσα σε αγκύλες, για παράδειγμα [Virgil]. Να σημειώσουμε ότι αυτό το όνομα θα πρέπει να χρησιμοποιήσουμε στο πεδίο PawnType στο πάνελ ιδιοτήτων της Pawn. Όταν μία γραμμή δε θέλουμε να εκτελείται, προσθέτουμε στην αρχή της ένα ερωτηματικό (;). Αναλυτικά, οι παράμετροι που μπορούμε να χρησιμοποιήσουμε για να προσδιορίσουμε μία οντότητα στο Pawn.ini είναι:

Παράμετρος	Περιγραφή
Actorname	Το όνομα του τρισδιάστατου μοντέλου-actor που θα χρησιμοποιήσουμε, για παράδειγμα robot.act.
Actorrotation	Οι μοίρες στους άξονες x, y και z που πρέπει να περιστραφεί το μοντέλο ώστε να είναι σωστά τοποθετημένο, για παράδειγμα -90 180 0.
Actorscale	Η κλίμακα του actor. Τιμές κάτω από 1 τον κάνουν μικρότερο και τιμές πάνω από 1 μεγεθύνουν το μοντέλο.
Actoralpha	Ο βαθμός διαφάνειας του actor. Προεπιλεγμένη είναι η τιμή 255 (απόλυτα αδιαφανές).
Animationspeed	Η τιμή καθορίζει την ταχύτητα κίνησης του actor. Τιμές πάνω από 1 τον κινούν πιο γρήγορα. Προεπιλεγμένη είναι η τιμή 1.
Fillcolor	Το χρώμα γεμίσματος του actor. Προεπιλεγμένη είναι η τιμή 255 255 255 (λευκό).
Ambientcolor	Το χρώμα του περιβάλλοντος φωτισμού του μοντέλου. Προεπιλεγμένη είναι η τιμή 255 255 255 (λευκό).
Subjecttogravity	Αν δώσουμε την τιμή TRUE, τότε ο actor θα υπόκειται στη δύναμη της βαρύτητας.
Boundingboxanimation	Το όνομα του animation που θα εκτελεί ο actor. Το όνομα πρέπει να δοθεί ακριβώς όπως το ορίσαμε κατασκευάζοντας τον actor με το Actor Studio. Αν δε θέλουμε να υπολογίζονται οι συγκρούσεις με τα άλλα αντικείμενα, μπορούμε στο πεδίο αυτό να δώσουμε την τιμή nocollide.
Shadowsize	Αριθμητική τιμή που προσδιορίζει το μέγεθος της σκιάς.
Shadowalpha	Ο βαθμός διαφάνειας της σκιάς (0-255).
Shadowalphamap	Το όνομα της εικόνας για το διαφανές τμήμα της σκιάς.
Shadowbitmap	Το όνομα της εικόνας που χρησιμοποιείται για σκιά.
Projectedshadows	Αν δώσουμε την τιμή TRUE, τότε ο actor θα δημιουργεί προβολικές σκιάς (οι οποίες δε χρειάζονται αρχεία εικόνων).

Παράμετρος	Περιγραφή
Stencilshadows	Αν δώσουμε την τιμή TRUE, τότε ο actor θα δημιουργεί Stencil σκιές (οι οποίες χρειάζονται αρχεία εικόνων που ορίσαμε πιο πάνω).
Icon	Το όνομα μίας εικόνας που αναπαριστά τον actor και θα χρησιμοποιείται εάν και όταν εμφανίζεται παράθυρο διαλόγου με το χρήστη (προαιρετικό πεδίο).
Environmentmapping	Αν δώσουμε την τιμή TRUE, τότε στον actor θα χρησιμοποιηθεί η τεχνική Environmental Mapping.
Allmaterial	Αν δώσουμε την τιμή TRUE, τότε το Environmental Mapping θα εφαρμοστεί σ' όλα τα υλικά που "ντύνουν" τον actor.
Percentmapping	Το ποσοστό του Environmental Mapping που θα χρησιμοποιηθεί για τον actor.
Percentmaterial	Το ποσοστό του υλικού του Environmental Mapping που θα χρησιμοποιηθεί για τον actor.

Για τις ανάγκες του παραδείγματός μας, θα ορίσουμε μία οντότητα Pawn που θα την ονομάσουμε Robot. Μετακινούμαστε λοιπόν στο τέλος του αρχείου Pawn.ini και κάνουμε τις εξής δηλώσεις:

[Robot]

```
actorname = robot.act
actorrotation = 0 90 0
actorscale = 2
subjecttogravity = false
boundingboxanimation = walk
projectedshadows = true
```

Θα χρησιμοποιήσουμε δηλαδή τον actor robot.act, τον οποίο θα περιστρέψουμε κατά 0, 90, 0 μοίρες και θα διπλασιάσουμε το μέγεθός του. Δε θα υπόκειται στη βαρύτητα, το animation που θα εκτελεί θα είναι το walk και θα δημιουργεί προβολικές σκιές. Παρατηρούμε ότι δε χρειάζεται να χρησιμοποιήσουμε όλες τις παραμέτρους, παρά μονάχα αυτές που είναι απαραίτητες. Εννοείται βέβαια ότι η πιο σημαντική είναι το όνομα του actor. Παρατηρούμε επίσης ότι μεταξύ της παραμέτρου και της τιμής της μεσολαβεί ο χαρακτήρας ίσον (=). Να σημειώσουμε ότι οι κενοί χαρακτήρες δεξιά και αριστερά του δεν είναι απαραίτητοι.

Αφού ολοκληρώσουμε τις απαραίτητες δηλώσεις στο αρχείο Pawn.ini, μεταφερόμαστε στις ιδιότητες της Pawn. Στο πάνελ εντολών και συγκεκριμένα στο πεδίο PawnType, δηλώνουμε το όνομα Robot. Παρ' όλα αυτά, αν εκτελέσουμε την εφαρμογή μας, ο actor δε θα εμφανιστεί. Αυτό γιατί πρέπει να κατασκευάσουμε και ένα αρχείο δέσμης εντολών το όνομα του οποίου θα προσθέσουμε στο πεδίο ScriptName του πάνελ ιδιοτήτων. Όπως ήδη τονίσαμε, το αρχείο πρέπει να βρίσκεται στο φάκελο C:\RealityFactory\Scripts και να έχει την κατάληξη .s. Η δομή του αρχείου καθώς επίσης και η σύνταξη των εντολών είναι παρόμοια με αυτή που χρησιμοποιήσαμε στην οντότητα SkyDome (κεφάλαιο 13.2). Χρησιμοποιούμε δηλαδή άγκιστρα και αγκύλες για να ορίσουμε την αρχή και το τέλος του αρχείου ή μίας ομάδας εντολών και η κάθε ομάδα έχει ένα όνομα που συνοδεύεται από [().

Τις εντολές που χρησιμοποιούμε θα τις αναλύσουμε στο επόμενο κεφάλαιο που διαπραγματεύεται την κίνηση των rawns και τη γλώσσα προγραμματισμού του Reality Factory. Για τις ανάγκες του παραδείγματός μας, αρκεί να ανοίξουμε το “Σημειωματάριο”, να προσθέσουμε τις παρακάτω εντολές και να αποθηκεύσουμε το αρχείο ονομάζοντάς το για παράδειγμα test.s:

```
{
Start[ ()
{
RotateMoveToPoint(“walk”, 20, 20, false, “”);
}]
}
```

Αφού συμπληρώσουμε το όνομα test.s στο πεδίο ScriptName στο πάνελ ιδιοτήτων (δηλαδή ποιο script ελέγχει την Pawn) και το όνομα Start στο πεδίο SpawnOrder (δηλαδή ποια ομάδα εντολών εκτελείται πρώτη), μπορούμε να εκτελέσουμε την εφαρμογή μας και να δούμε τον actor στο επίπεδό μας. Τα υπόλοιπα πεδία προς το παρόν δε χρειάζεται να συμπληρωθούν.

Συνοψίζοντας, τα θετικά και αρνητικά σημεία των οντοτήτων που μπορούν να χειριστούν τρισδιάστατα μοντέλα, καθώς επίσης και το πότε χρησιμοποιούνται, είναι τα εξής:

Animation για το μοντέλο που χειρίζεται η οντότητα

StaticMesh:	Όχι
StaticEntityProxy:	Ναι
Attribute:	Όχι
Pawn:	Ναι

Είδος αλληλεπιδράσεων που υποστηρίζονται

StaticMesh:	Μόνο υπολογισμός σύγκρουσης με το χρήστη
StaticEntityProxy:	Υπολογισμός σύγκρουσης με το χρήστη, βλάβη στο χρήστη
Attribute:	Υπολογισμός σύγκρουσης με το χρήστη, αύξηση παραμέτρων του χρήστη, το αντικείμενο “μαζεύεται” για χρήση αργότερα
Pawn:	Όλα τα παραπάνω, διάλογοι

Σκιές

StaticMesh:	Μόνο σε άλλες StaticMesh
StaticEntityProxy:	Stencil και Projected
Attribute:	Stencil και Projected
Pawn:	Stencil και Projected

Συνδεδεμένος ήχος

StaticMesh:	Μόνο έμμεσα μέσω σκανδάλης
StaticEntityProxy:	Ναι
Attribute:	Ναι
Pawn:	Ναι

Η οντότητα μπορεί να μετακινηθεί από το χρήστη

StaticMesh:	Όχι
StaticEntityProxy:	Ναι
Attribute:	Όχι
Pawn:	Ναι (προγραμματιστικά)

Δυνατότητα κίνησης του τρισδιάστατου μοντέλου-actor

StaticMesh:	Έμμεσα (συνδέοντάς τη με μοντέλο που κινείται)
StaticEntityProxy:	Άμεσα (με ειδική οντότητα)
Attribute:	Έμμεσα (συνδέοντάς τη με μοντέλο που κινείται)
Pawn:	Άμεσα (με ειδική οντότητα και προγραμματιστικά)

Ευκολία χρήσης

StaticMesh:	Μεγάλη
StaticEntityProxy:	Αρκετά μεγάλη
Attribute:	Αρκετά μεγάλη
Pawn:	Απαιτείται γνώση της προγραμματιστικής γλώσσας του Reality Factory

Κατάλληλη για

StaticMesh:	Στατικά αντικείμενα με σχεδόν μηδενική αλληλεπίδραση με το χρήστη
StaticEntityProxy:	Στατικά και κινούμενα αντικείμενα με αρκετή αλληλεπίδραση με το χρήστη
Attribute:	Στατικά αντικείμενα με μόνη αλληλεπίδραση την αύξηση παραμέτρων του χρήστη
Pawn:	Στατικά και κινούμενα αντικείμενα με πολύ μεγάλη αλληλεπίδραση με το χρήστη

Κεφάλαιο 16°

Δίνοντας κίνηση σε γεωμετρικά στερεά και οντότητες

Τα γεωμετρικά στερεά και οι actors, όταν τοποθετούνται στην εφαρμογή μας, είναι στατικοί. Ακόμα και αν ορίσουμε να χρησιμοποιείται κάποιο από τα animation ενός actor, κάτι τέτοιο δε συνεπάγεται ότι αυτός θα μετακινηθεί από το σημείο Α στο σημείο Β. Ο κατασκευαστής της εφαρμογής θα πρέπει, με τη χρήση οντοτήτων, να δώσει τη δυνατότητα κίνησης. Στο κεφάλαιο αυτό θα ασχοληθούμε με τις διαφορετικές τεχνικές που χρησιμοποιούνται ανάλογα με το αν θέλουμε να δώσουμε κίνηση σε στερεά ή actors και ανάλογα με το είδος κίνησης θέλουμε να επιτύχουμε κάθε φορά.

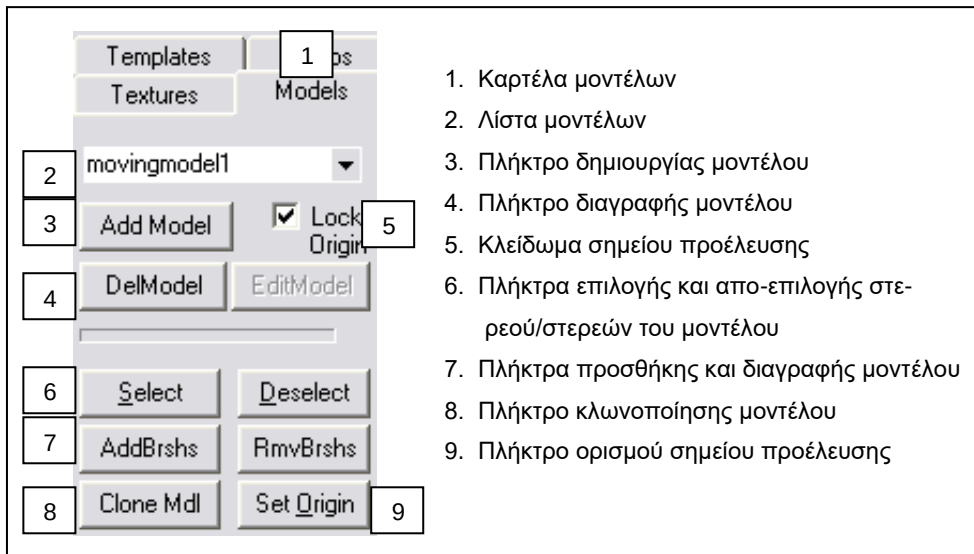
16.1 Δίνοντας κίνηση σε γεωμετρικά στερεά

Τα γεωμετρικά στερεά που έχουμε εισαγάγει στην εφαρμογή μας από την καρτέλα των Προτύπων-Templates του πάνελ εντολών μπορούν να κινούνται, αφού όμως πρώτα μετατραπούν σε μοντέλα. Η διαδικασία περιγράφηκε με συντομία στο κεφάλαιο 12.2.3, όταν κατασκευάσαμε το νερό μίας πισίνας. Στο σημείο όμως αυτό θα δούμε αναλυτικά τις επιλογές που μας δίνει η καρτέλα Μοντέλα-Models του πάνελ εντολών.

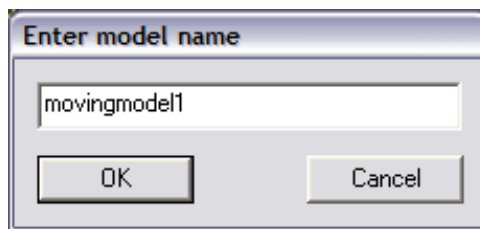
Μπορούμε να χωρίσουμε την καρτέλα σε δύο μέρη. Στο πάνω μέρος χειριζόμαστε τα μοντέλα και στο κάτω χειριζόμαστε τις κινήσεις τους. Η τυπική διαδικασία εισαγωγής ενός μοντέλου είναι η εξής:

- Προσαρμόζουμε και εισάγουμε ένα γεωμετρικό στερεό. Το επιλέγουμε (αποκτά πράσινο περίγραμμα).
- Έχοντας το στερεό επιλεγμένο, ανοίγουμε την καρτέλα Μοντέλα-Models από το πάνελ εντολών.
- Στο πάνω μισό της καρτέλας (εικόνα 16.1) κάνουμε κλικ στο πλήκτρο εισαγωγής μοντέλου  και στο παράθυρο διαλόγου (εικόνα 16.2) δίνουμε ένα χαρακτηριστικό όνομα για το μοντέλο μας. Από τώρα και στο εξής, αυτό το γεωμετρικό στερεό έχει αποκτήσει τις ιδιότητες που έχουν τα μοντέλα και μπορούμε να το χειριστούμε με μία σειρά οντοτήτων.

Δεν πρέπει να συγχέουμε τις έννοιες γεωμετρικό στερεό και μοντέλο. Θα πρέπει να έχουμε κατά νου ότι το μοντέλο βρίσκεται ένα επίπεδο πάνω από το στερεό, κατά κάποιον τρόπο το περιβάλλει. Για το λόγο αυτόν άλλωστε, όταν διαγράψουμε ένα μοντέλο με το πλήκτρο διαγραφής , δε διαγράφεται και το στερεό.



Εικόνα 16.1 Πρώτο μέρος καρτέλας μοντέλων



Εικόνα 16.2 Παράθυρο διαλόγου εισαγωγής ονόματος μοντέλου

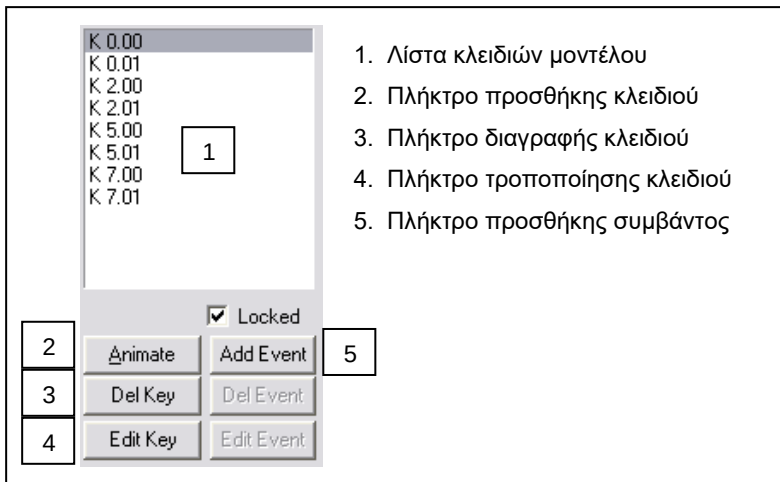
Έχοντας επιλεγμένο το μοντέλο, μπορούμε να προσθέσουμε σ' αυτό και άλλα στερεά, επιλέγοντάς τα και πατώντας το πλήκτρο προσθήκης στερεών **AddBrshs**. Έτσι, μπορούμε να δημιουργήσουμε περίπλοκους συνδυασμούς στερεών που όλα θα ανήκουν στο ίδιο μοντέλο, με αποτέλεσμα να μπορούμε να τα μετακινήσουμε πλέον όλα μαζί και να ελέγχονται ταυτόχρονα από κάποια οντότητα χειρισμού μοντέλων. Το ή τα στερεά που ανήκουν σ' ένα μοντέλο μπορούμε να τα επιλέξουμε πατώντας το πλήκτρο επιλογής στερεών **Select**, αφού πρώτα επιλέξουμε ένα μοντέλο από τη λίστα. Τέλος, μπορούμε να κλωνοποιήσουμε ένα μοντέλο (παράγοντας αντίγραφα των στερεών που περιέχει) επιλέγοντάς το από τη λίστα και πατώντας το πλήκτρο κλωνοποίησης **Clone Mdl**.

Στα γεωμετρικά στερεά που ανήκουν σε κάποιο μοντέλο, καλό είναι να μην κάνουμε αντιγραφή και επικόλληση. Αυτό γιατί δημιουργείται ταυτόχρονα και αντίγραφο του μοντέλου στο οποίο ανήκουν. Προτιμότερο είναι να χρησιμοποιούμε το πλήκτρο κλωνοποίησης. Στη συνέχεια μπορούμε να διαγράψουμε το μοντέλο-κλώνο, παραμένουν όμως τα αντίγραφα των στερεών που περιείχε.

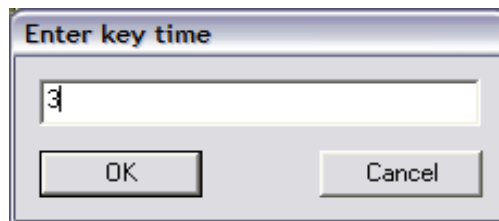
Ένα θέμα που πρέπει να προσεχθεί ιδιαίτερα είναι το σημείο προέλευσης. Το σημείο αυτό καθορίζει τη θέση του μοντέλου στο χώρο και κανονικά βρίσκεται στο κέντρο του στερεού. Σε μερικές περιπτώσεις όμως το RFEEditPro μπορεί να αστοχήσει, τοποθετώντας το σημείο προέλευσης εκτός του στερεού. Για να βεβαιωθούμε για τη θέση του σημείου προέλευσης, κάνουμε κλικ στο πλήκτρο καθορισμού σημείου προέλευσης . Στις τρεις προεπισκοπήσεις (πλην της τρισδιάστατης) παρατηρούμε ότι εμφανίζεται ένα πράσινο τετραγωνάκι μ' ένα Χ που αντιπροσωπεύει το σημείο προέλευσης και το οποίο και μπορούμε να μετακινήσουμε. Όταν ολοκληρώσουμε τη μετακίνηση, κάνουμε κλικ στο πλήκτρο , που έχει πάρει τη θέση του πλήκτρου . Τη θέση του σημείου προέλευσης μπορούμε να την κλειδώσουμε έχοντας τσεκαρισμένο το αντίστοιχο τετραγωνάκι .

Το επόμενο βήμα είναι να δώσουμε κίνηση στο μοντέλο μας. Αυτό γίνεται από το κάτω μέρος της καρτέλας μοντέλων (εικόνα 16.3). Το RFEEditPro χρησιμοποιεί τον όρο Κλειδί-Key, όπως το Milkshape3D χρησιμοποιεί τον όρο Keyframe (κεφάλαιο 10.3.2). Το Κλειδί, με άλλα λόγια, αντιπροσωπεύει χρονικές στιγμές στην κίνηση του μοντέλου. Σε κάθε χρονική στιγμή το μοντέλο, καθώς κινείται, προφανώς βρίσκεται και σε διαφορετική θέση. Έτσι λοιπόν, το Κλειδί είναι η θέση του μοντέλου σε μία δεδομένη χρονική στιγμή. Η διαδικασία που ακολουθούμε για τον ορισμό των κλειδιών είναι η ακόλουθη:

- Επιλέγουμε ένα μοντέλο από τη λίστα και ταυτόχρονα επιλέγονται το/τα στερεά που περιλαμβάνει.
- Κάνουμε κλικ στο πλήκτρο προσθήκης κλειδιού .
- Σε όποια από τις τρεις προεπισκοπήσεις θέλουμε (πλην της τρισδιάστατης) μετακινούμε το μοντέλο στη νέα του θέση.
- Το πλήκτρο  έχει γίνει πλήκτρο διακοπής . Το κάνουμε κλικ.
- Εμφανίζεται παράθυρο διαλόγου (εικόνα 16.4) στο οποίο εισάγουμε τη χρονική στιγμή (σε δευτερόλεπτα) στην οποία θα βρίσκεται το μοντέλο στη νέα του θέση.
- Αν θέλουμε να συνεχίσουμε την κίνηση του μοντέλου, κάνουμε πάλι κλικ στο πλήκτρο προσθήκης κλειδιού  και επαναλαμβάνουμε τη διαδικασία μετακινώντας το μοντέλο σε άλλη θέση και δίνοντας καινούργιο κλειδί.



Εικόνα 16.3 Δεύτερο μέρος καρτέλας μοντέλων



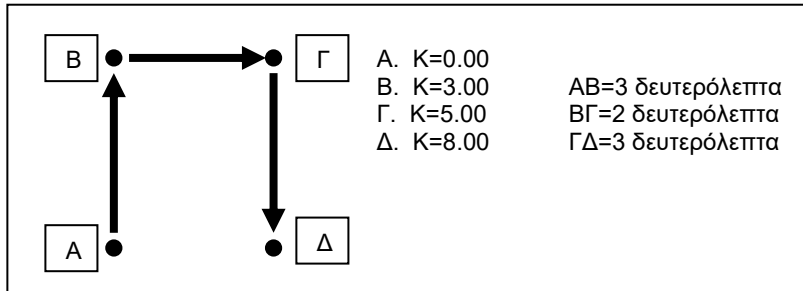
Εικόνα 16.4 Παράθυρο εισαγωγής χρονικής στιγμής

Αν κάποιος κλειδί δεν το χρειαζόμαστε, το επιλέγουμε από τη λίστα κλειδιών και κάνουμε κλικ στο πλήκτρο διαγραφής **Del Key**. Μπορούμε τέλος να αλλάξουμε την τιμή ενός κλειδιού, κάνοντας κλικ στο αντίστοιχο πλήκτρο **Edit Key** και δίνοντας νέα τιμή.

Για να δώσουμε ένα παράδειγμα, ας υποθέσουμε ότι ένα μοντέλο θέλουμε να κάνει μία κίνηση που να σχηματίζει ένα Π και θέλουμε η συνολική διάρκεια της κίνησης να είναι 8 δευτερόλεπτα. Πέρα από το σημείο εκκίνησης, θα χρειαστούμε άλλα τρία κλειδιά, δύο στην κορυφή του Π και ένα ακόμα στη βάση του (σχήμα 16.1). Κάνουμε λοιπόν τα εξής βήματα:

- Το σημείο εκκίνησης έχει πάντα κλειδί με την τιμή 0.
- Κάνουμε κλικ στο πλήκτρο **Animate** και μετακινούμε το μοντέλο στη θέση Β. Πατάμε το πλήκτρο **Stop Animating** και δίνουμε την τιμή 3 στο κλειδί, η κίνηση δηλαδή από το Α στο Β θα διαρκέσει 3 δευτερόλεπτα.
- Κάνουμε κλικ στο πλήκτρο **Animate** και μετακινούμε το μοντέλο στη θέση Γ. Πατάμε το πλήκτρο **Stop Animating** και δίνουμε την τιμή 5 στο κλειδί, δηλαδή η κίνηση από το σημείο Β στο σημείο Γ θα έχει διάρκεια 2 δευτερόλεπτα.

- Κάνουμε κλικ στο πλήκτρο  και μετακινούμε το μοντέλο στη θέση Δ. Πατάμε το πλήκτρο  και δίνουμε την τιμή 8 στο κλειδί, δηλαδή η κίνηση από το σημείο Γ μέχρι το σημείο Δ θα είναι διάρκειας 3 δευτερολέπτων.



Σχήμα 16.1 Καθορισμός σημείων-κλειδιών

Μπορούμε να ελέγξουμε τις θέσεις του μοντέλου κάνοντας κλικ σε κάποιο κλειδί της λίστας κλειδιών. Καλό είναι επίσης σε κάθε θέση να ελέγξουμε και τα σημεία προέλευσης, κάνοντας κλικ στο πλήκτρο . Αν η θέση κάποιου σημείου προέλευσης δεν ταυτίζεται με τη θέση του μοντέλου, μπορούμε να το μετακινήσουμε στη σωστή θέση.

Τα μοντέλα μπορούν να κινηθούν σε ευθεία ή σε τεθλασμένη γραμμή, δεξιά ή αριστερά, μπροστά ή πίσω, πάνω ή κάτω ή σε συνδυασμό κατευθύνσεων, για παράδειγμα πάνω και δεξιά. Δεν πρέπει όμως ποτέ να τα περιστρέψουμε (με το εργαλείο ) και μετά να τα κινήσουμε σε ευθεία γραμμή. Κάτι τέτοιο έχει ως αποτέλεσμα την οριστική απορρύθμιση του σημείου προέλευσης και της θέσης.

16.1.1 Η οντότητα MovingPlatform

Αν εκτελέσουμε την εφαρμογή μας, θα διαπιστώσουμε ότι το/τα γεωμετρικά στερεά που έχουμε μετατρέψει σε μοντέλο δεν κινούνται. Αυτό οφείλεται στο ότι δεν έχουμε τοποθετήσει κάποια οντότητα που να ελέγχει την κίνηση του. Για να κινηθεί λοιπόν το μοντέλο, χρειάζεται να εισαγάγουμε την οντότητα MovingPlatform. Οι ιδιότητές της είναι:

Πεδίο	Περιγραφή
AnimationSpeed	Η ταχύτητα του animation. Η ρύθμιση αυτή παρακάμπτει τη διάρκεια που έχει προκύψει όταν ορίζαμε τα σημεία κλειδιά του animation. Η τιμή 1 αντιπροσωπεύει το 100% της αρχικής διάρκειας. Τιμές κάτω από 1 επιβραδύνουν το animation, ενώ τιμές πάνω από το 1 το επιταχύνουν.

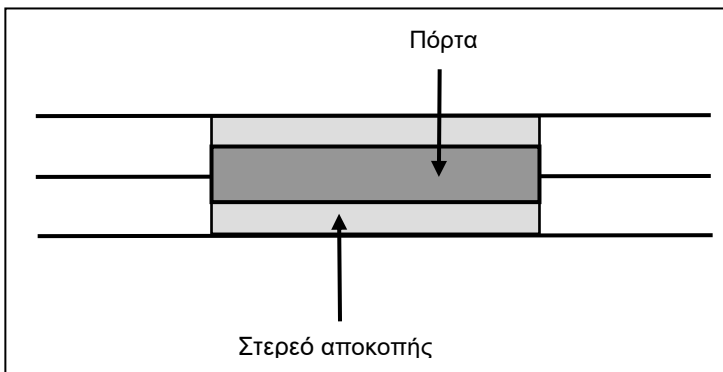
Πεδίο	Περιγραφή
bAllowInside	Αν δώσουμε την τιμή TRUE, τότε δε θα γίνεται ο έλεγχος της σύγκρουσης του χρήστη με το μοντέλο, επιτρέποντάς του να μπει μέσα σ' αυτό.
bAudioLoops	Αν δώσουμε την τιμή TRUE, τότε ο ήχος που συνδέεται με την οντότητα MovingPlatform θα παίζει αέναα.
bAutoStart	Αν δώσουμε την τιμή TRUE, τότε το animation θα εκτελείται με την έναρξη της εφαρμογής.
bLooping	Αν δώσουμε την τιμή TRUE, τότε το animation θα επαναλαμβάνεται αέναα.
bNoCollide	Αν δώσουμε την τιμή TRUE, τότε το animation δε θα ξεκινά με τη σύγκρουση του χρήστη με το μοντέλο, αλλά με άλλον τρόπο, για παράδειγμα με τη χρήση σκανδάλης.
bOneShot	Αν δώσουμε την τιμή TRUE, τότε το animation θα εκτελείται μόνο μία φορά.
bReverse	Αν δώσουμε την τιμή TRUE, τότε το animation θα εκτελείται και αντίστροφα (το μοντέλο θα επιστρέφει στην αρχική του θέση).
bRunFromList	Αν δώσουμε την τιμή TRUE, τότε το animation θα εκτελείται με βάση τους χρόνους που θα δώσουμε στο πεδίο TimeList (ανεξάρτητα από τους χρόνους των σημείων-κλειδιών).
bRunTimed	Αν δώσουμε την τιμή TRUE, τότε το animation θα εκτελείται κάθε φορά που η οντότητα ενεργοποιείται, για χρόνο που προσδιορίζουμε στο πεδίο TimeEachTrig.
bRunToNextEvent	Αν δώσουμε την τιμή TRUE, τότε το animation θα εκτελείται κατά χρόνο που προσδιορίζεται στη λίστα γεγονότων του μοντέλου.
bRunWhileTrig	Αν δώσουμε την τιμή TRUE, τότε το animation θα εκτελείται για όσο χρόνο η σκανδάλη θα είναι σε κατάσταση ON (ενεργή).
bShoot	Αν δώσουμε την τιμή TRUE, τότε το animation θα εκτελείται μόνο αν το μοντέλο πυροβοληθεί και όχι με τη σύγκρουση του χρήστη με αυτό.
Model	Ποιο μοντέλο είναι συνδεδεμένο με αυτή την οντότητα MovingPlatform.
NextToTrigger	Ποιο μοντέλο θα ενεργοποιηθεί στη συνέχεια. Και το άλλο μοντέλο χρειάζεται να έχει τη δική του MovingPlatform.
Origin	Η θέση της οντότητας. Η τιμή συμπληρώνεται αυτόματα.

Πεδίο	Περιγραφή
ParentModel	Το όνομα του μοντέλου “πατέρα”. Αν δώσουμε στο πεδίο αυτό το όνομα κάποιου άλλου μοντέλου, τότε το μοντέλο που ελέγχει η συγκεκριμένη MovingPlatform θα λειτουργεί με βάση τις ρυθμίσεις της MovingPlatform που ελέγχει το μοντέλο “πατέρα”. Είναι ο πιο γρήγορος τρόπος ώστε διαφορετικά μοντέλα που ελέγχονται από διαφορετικές MovingPlatforms να ελέγχονται τελικά από την πατρική MovingPlatform. Στις MovingPlatforms των μοντέλων “παιδιών” δε χρειάζεται να κάνουμε κάποια άλλη ρύθμιση εκτός από τη δήλωση στο πεδίο αυτό.
PlayerOnly	Αν δώσουμε την τιμή TRUE, τότε μόνο ο χρήστης θα μπορεί να ενεργοποιήσει τη συγκεκριμένη MovingPlatform.
Rideable	Αν δώσουμε την τιμή TRUE, τότε ο χρήστης θα μπορεί να ανέβει στο μοντέλο. Η τιμή FALSE επιτρέπει στο χρήστη να περνά μέσα απ' αυτό.
szEntityName	Το όνομα της συγκεκριμένης οντότητας MovingPlatform. Αυτό το όνομα θα χρησιμοποιηθεί σε πεδία ιδιοτήτων άλλων οντοτήτων που θέλουμε να συνδέονται με αυτή.
szSoundFile	Το αρχείο ήχου που θα εκτελείται όταν η MovingPlatform είναι ενεργή.
TimeEachTrig	Ο χρόνος που θα ενεργοποιείται κάθε φορά η MovingPlatform. Το πεδίο ισχύει μόνο όταν έχουμε δώσει την τιμή TRUE στο πεδίο bRunTimed.
TimeList	Λίστα τιμών που προσδιορίζει τα χρονικά διαστήματα που θα είναι ενεργή η MovingPlatform, για παράδειγμα 1.0 3.5 4.0. Το πεδίο ισχύει μόνο όταν έχουμε δώσει την τιμή TRUE στο πεδίο bRunFromList.
TriggerName	Το όνομα της σκανδάλης που θα ενεργοποιεί τη συγκεκριμένη MovingPlatform.
UseKey	Αν δώσουμε την τιμή TRUE, τότε ο χρήστης θα μπορεί να ενεργοποιήσει τη συγκεκριμένη MovingPlatform πατώντας το πλήκτρο χρήσης.

Η καλύτερη αξιοποίηση των δυνατοτήτων της MovingPlatform, αλλά και των περισσότερων οντοτήτων, γίνεται με τις σκανδάλες (αν συμπληρώσουμε δηλαδή το πεδίο TriggerName), η χρήση των οποίων αναλύεται στο επόμενο κεφάλαιο. Στο σημείο αυτό θα αρκεστούμε σε μερικά απλά παραδείγματα χρήσης της MovingPlatform και στις ρυθμίσεις που πρέπει να γίνουν.

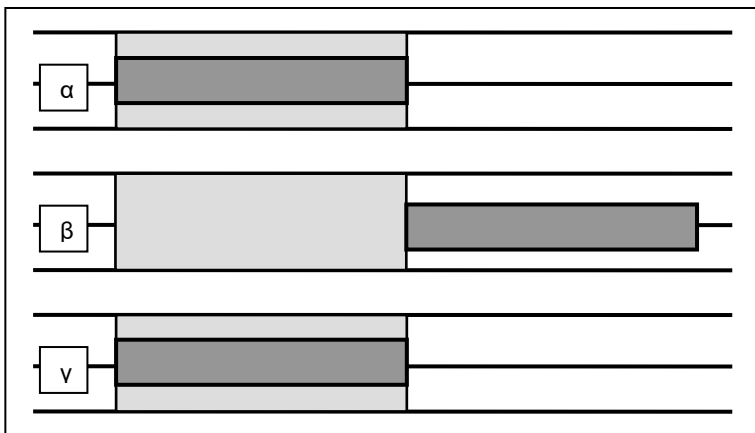
Μία πρώτη χρήση της MovingPlatform είναι να κατασκευάσουμε συρόμενες πόρτες, που ανοίγουν με την επαφή του χρήστη σε αυτές. Για να κατασκευάσουμε μία τέτοια πόρτα μεταξύ δύο δωματίων, ακολουθούμε τα εξής βήματα:

Μεταξύ των δωματίων τοποθετούμε ένα στερεό με πάχος όσο και οι δύο τοίχοι μαζί και με μήκος και πλάτος αρκετό ώστε να μπορεί να περάσει το μοντέλο που αναπαριστά το χρήστη. Στο στερεό δίνουμε τις ιδιότητες Συμπαγές-Solid και Αποκοπή-Cut, ώστε να ανοίξει ένα κενό μεταξύ των δύο δωματίων. Στο κενό τοποθετούμε ακόμα ένα συμπαγές στερεό (αλλά όχι Cut), με μήκος και ύψος όσο το στερεό αποκοπής αλλά με μικρότερο πάχος, που θα είναι η κινούμενη πόρτα (σχήμα 16.2).



Σχήμα 16.2 Κάτοψη κατασκευής πόρτας

Επιλέγουμε την πόρτα και τη μετατρέπουμε σε μοντέλο. Στο μοντέλο θα δώσουμε τρία κλειδιά κίνησης, ένα στην κλειστή θέση, ένα στην ανοιχτή θέση και ένα που θα είναι η επιστροφή στην κλειστή θέση (σχήμα 16.3α-γ).



Σχήμα 16.3 Οι διαδοχικές θέσεις της πόρτας

Τα πεδία που πρέπει να προσέξουμε στη MovingPlatform και οι τιμές τους είναι:

Πεδίο	Τιμή	Πεδίο	Τιμή
bAutoStart	FALSE	bOneShot	FALSE
bLooping	FALSE	PlayerOnly	TRUE
bReverse	FALSE*	Rideable	FALSE

* Αν στο πεδίο bReverse δώσουμε την τιμή TRUE, τότε μπορούμε να παραλείψουμε το τρίτο κλειδί στο animation του μοντέλου της πόρτας.

Για να κάνουμε ένα στερεό να συμπεριφέρεται ως κυλιόμενη σκάλα ή ανελκυστήρας, δίνουμε τις εξής τιμές:

Πεδίο	Τιμή	Πεδίο	Τιμή
bAutoStart	FALSE	bOneShot	FALSE
bLooping	TRUE	PlayerOnly	TRUE
bReverse	TRUE	Rideable	TRUE

Για να λειτουργεί ως διακόπτης μίας χρήσης:

Πεδίο	Τιμή	Πεδίο	Τιμή
bAutoStart	FALSE	bOneShot	TRUE
bLooping	FALSE	PlayerOnly	TRUE
bReverse	TRUE	Rideable	FALSE

Για να λειτουργεί ως κινούμενο στερεό με την έναρξη της εφαρμογής:

Πεδίο	Τιμή	Πεδίο	Τιμή
bAutoStart	TRUE	bOneShot	FALSE
bLooping	TRUE	PlayerOnly	FALSE
bReverse	TRUE	Rideable	FALSE

Ένα σημείο που πρέπει να προσέξουμε είναι η περίπτωση να δώσουμε την τιμή TRUE στο πεδίο bLooping και την τιμή FALSE στο πεδίο bReverse. Αυτό που θα συμβεί είναι το στερεό να εκτελεί την κίνηση, να επιστρέφει ακαριαία στην αρχική του θέση και να ξεκινά πάλι. Επειδή κάτι τέτοιο φαίνεται παράξενο, προτιμούμε να δίνουμε την τιμή TRUE και στη bReverse.

Όταν θέλουμε η MovingPlatform να μεταφέρει το χρήστη από ένα σημείο σ' ένα άλλο, πρέπει να έχουμε υπόψη μας τρία σημαντικά σημεία:

- 1) Να δώσουμε αρκετό χρόνο στο χρήστη ώστε να μπορεί να ανέβει ή να κατέβει από το στερεό που κινεί η MovingPlatform. Αυτό το επιτυγχάνουμε δίνοντας επιπλέον κλειδιά στην κίνηση όπου το στερεό, για το χρόνο που καθορίζουμε, μένει ακίνητο. Για παράδειγμα, θέλουμε ένα στερεό να κινείται από το σημείο A στο σημείο B μέσα σε 5 δευτερόλεπτα, αλλά θέλουμε επίσης να δώσουμε περιθώριο 2 δευτερολέπτων στο χρήστη να ανέβει και 2 για να κατέβει. Αντί να χρησιμοποιήσουμε δύο κλειδιά, θα χρησιμοποιήσουμε τέσσερα ως εξής:
 - i) Στο πρώτο κλειδί ($K=0$) το στερεό είναι στη θέση A.
 - ii) Στο δεύτερο κλειδί ($K=2$) το στερεό παραμένει και πάλι στη θέση A (εισάγουμε δηλαδή ένα κλειδί χωρίς να μετακινήσουμε το στερεό).
 - iii) Στο τρίτο κλειδί ($K=7$) μετακινούμε το στερεό στη θέση B.
 - iv) Στο τέταρτο κλειδί ($K=9$) το στερεό παραμένει και πάλι στη θέση B (εισάγουμε δηλαδή και πάλι ένα κλειδί χωρίς να μετακινήσουμε το στερεό).

Έτσι, καταφέρνουμε να κάνουμε το στερεό παραμένει ακίνητο για δύο δευτερόλεπτα στο σημείο έναρξης και για δύο στο σημείο τερματισμού.

- 2) Εμφανίζονται αστοχίες στο Reality Factory όταν η κατεύθυνση της κίνησης είναι διαφορετική από κάθετη ή οριζόντια, για παράδειγμα όταν ο χρήστης πρέπει να μετακινηθεί προς τα πάνω ή κάτω σ' ένα κεκλιμένο επίπεδο. Αρκετές φορές, ενώ ο χρήστης είναι πάνω στο στερεό, το διαπερνά και πέφτει έξω απ' αυτό. Σε κάποιες περιπτώσεις το πρόβλημα διορθώνεται μετακινώντας το σημείο προέλευσης. Καλό είναι επίσης στην αφετηρία και τον τερματισμό το κινούμενο στερεό να είναι στο ίδιο ακριβώς επίπεδο με το επίπεδο όπου βρίσκεται ο χρήστης και όχι λίγο πάνω ή λίγο κάτω απ' αυτό.
- 3) Εισάγοντας τα κλειδιά του 1^{ου} σημείου όπου το στερεό δεν κινείται, είναι πιθανό να παρουσιαστεί και μία ταλάντωση του στερεού κατά το διάστημα που θα έπρεπε να παραμένει ακίνητο. Αν και το φαινόμενο δεν προκαλεί προβλήματα, αν επιθυμούμε να το εξαλείψουμε, πρέπει να χρησιμοποιήσουμε επιπλέον κλειδιά, που το καθένα να προσθέτει χρόνο 0.01 δευτερολέπτων στο προηγούμενό του. Τα κλειδιά της 1^{ης} περίπτωσης γίνονται τα εξής:

- i) Στο πρώτο κλειδί ($K=0$) το στερεό είναι στη θέση A.
- ii) Στο δεύτερο κλειδί ($K=0.01$) το στερεό παραμένει και πάλι στη θέση A.
- iii) Στο τρίτο κλειδί ($K=2$) το στερεό παραμένει και πάλι στη θέση A.
- iv) Στο τέταρτο κλειδί ($K=2.01$) το στερεό παραμένει και πάλι στη θέση A.
- v) Στο πέμπτο κλειδί ($K=7$) μετακινούμε το στερεό στη θέση B.
- vi) Στο έκτο κλειδί ($K=7.01$) το στερεό παραμένει και πάλι στη θέση B.
- vii) Στο έβδομο κλειδί ($K=9$) το στερεό παραμένει και πάλι στη θέση B.
- viii) Στο όγδοο κλειδί ($K=9.01$) το στερεό παραμένει και πάλι στη θέση B.

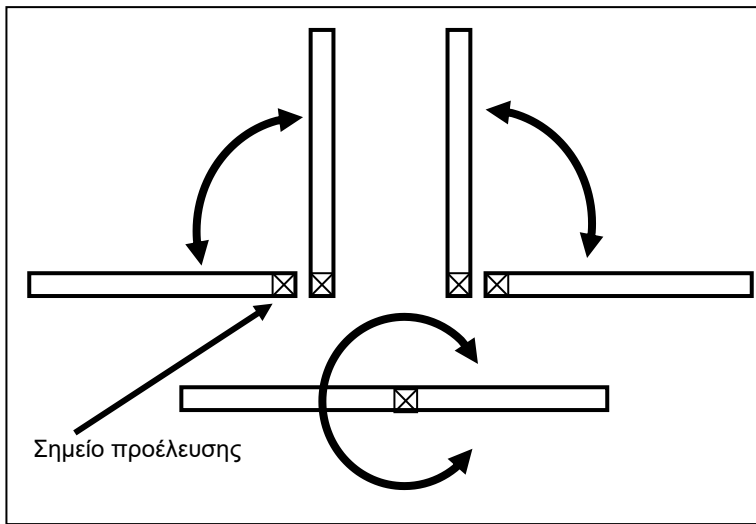
16.1.2 Η οντότητα Door

Παρόμοια με τη MovingPlatform είναι η οντότητα Door. Η χρήση της όμως περιορίζεται σε μοντέλα που εκτελούν περιστροφική κίνηση χωρίς να μετακινούνται στο χώρο. Συνήθως τέτοια μοντέλα χρησιμοποιούνται ως πόρτες. Οι ιδιότητες που μπορούμε να παραμετροποιήσουμε είναι οι ίδιες με αυτές της MovingPlatform, με εξαίρεση τις εξής:

Πεδίο	Περιγραφή
bReverse	Αν δώσουμε την τιμή TRUE, τότε το animation θα εκτελείται και αντιστρόφως (το μοντέλο θα επιστρέφει στην αρχική του θέση) <u>εάν</u> ο χρήστης συγκρουστεί με την πόρτα καθώς αυτή ανοίγει.
bRotating	Αν δώσουμε την τιμή TRUE, τότε η πόρτα είναι περιστρεφόμενη και κάποιοι υπολογισμοί σύγκρουσης με το μοντέλο δε γίνονται. Η τιμή TRUE μπορεί να οδηγήσει σε ανεπιθύμητα αποτελέσματα, για παράδειγμα ο χρήστης να περνά μέσα από την πόρτα ενώ αυτή είναι ακόμα κλειστή.

Εννοείται ότι όταν πρόκειται να δώσουμε τις διαδοχικές θέσεις του μοντέλου της πόρτας, δεν τη μετακινούμε με το εργαλείο , αλλά την περιστρέφουμε χρησιμοποιώντας το εργαλείο . Επειδή οι πόρτες παρουσιάζουν ιδιαιτερότητες στην κατασκευή τους, καλό είναι ο ενδιαφερόμενος να ακολουθήσει τις οδηγίες που δίνουμε στη συνέχεια.

Κατ' αρχάς, τις πόρτες μπορούμε να τις χωρίσουμε σε δύο κατηγορίες: α) σε αυτές που ανοίγουν προς μία κατεύθυνση σχηματίζοντας γωνία 90° σε σχέση με την αρχική τους θέση και β) στις περιστρεφόμενες που εκτελούν δηλαδή έναν πλήρη κύκλο. Το σημείο προέλευσης για την πρώτη περίπτωση πρέπει να τοποθετηθεί στο δεξιό ή το αριστερό άκρο του μοντέλου (ανάλογα με τη φορά που ανοίγει η πόρτα). Αντίθετα, το σημείο προέλευσης για τις περιστρεφόμενες πόρτες πρέπει να τοποθετηθεί ακριβώς στο μέσο του μοντέλου (σχήμα 16.4).



Σχήμα 16.4 Τοποθέτηση σημείου προέλευσης στις πόρτες

Για τις πόρτες που ανοίγουν προς μία κατεύθυνση, χρειαζόμαστε τουλάχιστον τρία κλειδιά (κλειστή, ανοιχτή και επιστροφή στην κλειστή θέση). Για τις περιστρεφόμενες χρειαζόμαστε πέντε αν εκτελείται πλήρης κύκλος (κλειστή, περιστροφή κατά 90° , περιστροφή κατά 180° , περιστροφή κατά 270° και περιστροφή κατά 360°) ή τρία αν εκτελείται περιστροφή 180° (κλειστή, περιστροφή κατά 90° και περιστροφή κατά 180°). Ειδικά για τις πόρτες που ανοίγουν προς μία κατεύθυνση, είναι χρήσιμο να δίνουμε ένα επιπλέον κλειδί, στο οποίο η πόρτα θα παραμένει ακίνητη στην ανοιχτή θέση για μερικά δευτερόλεπτα, ώστε να δοθεί χρόνος στο χρήστη να περάσει μέσα απ' αυτή.

Καλό είναι να δώσουμε την τιμή FALSE στα πεδία bReverse και bRotating για να αποφύγουμε ο χρήστης να περνά μέσα από τις πόρτες και για να μην αντιστρέφουν τη φορά κίνησής τους κατά τη σύγκρουσή τους με το χρήστη (που συμβαίνει συνεχώς εφόσον ο χρήστης καθώς περνά την πόρτα κρατά πατημένο το πλήκτρο κίνησης προς τα εμπρός). Τέλος, μπορεί να προτιμήσουμε η πόρτα να ανοίγει με το πλήκτρο χρήσης και όχι με τη σύγκρουση του χρήστη σ' αυτή, δίνοντας στα πεδία bNoCollide και UseKey την τιμή TRUE.

16.1.3 Δίνοντας κίνηση σε actors που ελέγχονται από StaticMesh

Κατά την παρουσίαση των οντοτήτων που χειρίζονται actors, αναφέραμε ότι η StaticMesh μπορεί να κινηθεί έμμεσα. Αυτό γίνεται αν συνδεθεί μ' ένα μοντέλο το οποίο κινείται από μία MovingPlatform. Η σειρά των ενεργειών είναι η εξής:

1. Εισάγουμε ένα γεωμετρικό στερεό.
2. Αλλάζουμε τις διαστάσεις του ώστε να χωράει μέσα στον actor που έχουμε σκοπό να εισαγάγουμε (εικόνα 16.5).
3. Μετατρέπουμε το στερεό σε μοντέλο.
4. Δίνουμε κλειδιά κίνησης στο μοντέλο.
5. Εισάγουμε μία οντότητα MovingPlatform η οποία θα ελέγχει το μοντέλο.
6. Εισάγουμε μία οντότητα StaticMesh για τον έλεγχο του actor, την οποία και συνδέουμε με το μοντέλο.

Αν υποθέσουμε ότι θέλουμε ο actor να κινείται με την έναρξη της εφαρμογής και να υπολογίζονται οι συγκρούσεις με το χρήστη, οι κυριότερες ρυθμίσεις στις οντότητες είναι:

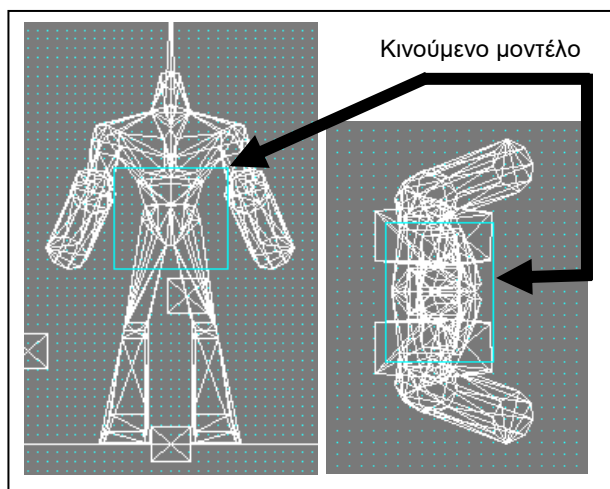
Για τη MovingPlatform

Πεδίο	Τιμή
bAutoStart	TRUE
bNoCollide	TRUE
bLooping	TRUE
bReverse	TRUE
Model	ModelforStaticMesh
Rideable	FALSE
PlayerOnly	FALSE

Για τη StaticMesh

Πεδίο	Τιμή
ColCheckLevel	2
Model	ModelforStaticMesh
szActorFile	Robot.act*

* Το αρχείο περιλαμβάνεται στην εγκατάσταση του Reality Factory.



Εικόνα 16.5 Εισάγοντας μοντέλο που θα κινήσει μία StaticMesh

Η μέθοδος κίνησης που περιγράψαμε, είναι αρκετά προβληματική γιατί:

1. Δεν υπολογίζονται σωστά οι συγκρούσεις με το χρήστη, με αποτέλεσμα ο actor σε αρκετές περιπτώσεις να περνά μέσα από αυτόν.
2. Δεν μπορούμε να περιστρέφουμε τον actor και έχουμε μόνο ευθύγραμμη κίνηση.
3. Δεν εκμεταλλευόμαστε τυχόν animation του actor.

Παρ' όλα αυτά, είναι ο μόνος τρόπος για να κινηθεί μία StaticMesh.

16.2 Δίνοντας κίνηση σε actors που ελέγχονται από StaticEntityProxy

Όταν παρουσιάσαμε την οντότητα StaticEntityProxy (κεφάλαιο 15.2), διαπιστώσαμε ότι μας παρέχει πολύ περισσότερες επιλογές χειρισμού των actors και για το λόγο αυτό είναι προτιμότερη από τη StaticMesh. Αν θέλαμε να κινήσουμε τον actor που χειρίζεται μία StaticEntityProxy, θα μπορούσαμε να χρησιμοποιήσουμε τη διαδικασία που περιγράψαμε πιο πάνω. Το Reality Factory όμως περιλαμβάνει την οντότητα PathFollower με παραπλήσιες ιδιότητες με τη MovingPlatform, που μας απαλλάσσει όμως από την ανάγκη να εισαγάγουμε, να κινήσουμε ένα μοντέλο και στη συνέχεια να το συνδέσουμε με μία οντότητα χειρισμού actors.

Πρέπει να σημειώσουμε ότι η PathFollower είναι ο μόνος τρόπος για να κινήσουμε (πέρα από τη StaticEntityProxy) τις παρακάτω οντότητες: AudioSource3D, Corona, DynamicLight, ElectricBolt, ElectricBoltTerminus, MorphingField, ParticleSystemProxy και TeleportTarget.

Για να ακολουθήσει η PathFollower μία συγκεκριμένη διαδρομή, θα χρειαστεί να τοποθετήσουμε οντότητες PathPoint σε καίρια σημεία. Η διαδικασία που ακολουθούμε είναι η εξής:

- Εισάγουμε μία StaticEntityProxy, την παραμετροποιούμε και ορίζουμε ποιος actor συνδέεται σ' αυτή.
- Εισάγουμε μία PathFollower, την παραμετροποιούμε και ορίζουμε ότι η πιο πάνω StaticEntityProxy συνδέεται σ' αυτή.
- Εισάγουμε οντότητες PathPoint που καθορίζουν τη διαδρομή που θα ακολουθήσει η PathFollower.

Οι ιδιότητες της PathFollower είναι:

Πεδίο	Περιγραφή
CollisionTest	Αν δώσουμε την τιμή TRUE, τότε θα εκτελείται ένας βασικός έλεγχος συγκρούσεων. Παρότι ο έλεγχος αυτός είναι σχετικά ανακριβής, εντούτοις θα εμποδίσει την οντότητα από το να διαπερνά, για παράδειγμα, τοίχους.
EntityName	Το όνομα της οντότητας με την οποία συνδέεται η PathFollower. Είναι το πιο σημαντικό πεδίο. Στο παράδειγμά μας θα δώσουμε το όνομα της StaticEntityProxy που θα χρησιμοποιήσουμε.
MotionLoops	Αν δώσουμε την τιμή TRUE, τότε η κίνηση θα επαναλαμβάνεται αέναα.
Origin	Η θέση της οντότητας. Η τιμή συμπληρώνεται αυτόματα.
PathFirstNode	Το όνομα του πρώτου PathPoint.
RandomRollTime	Ο χρόνος σε χιλιοστά του δευτερολέπτου μεταξύ των αλλαγών στη φορά της κίνησης όταν το πεδίο PointType στις οντότητες PathPoint έχει την τιμή 2.
RangeTrigger	Η ακτίνα δράσης μίας ειδικής σκανδάλης που, όταν ο χρήστης εισέλθει στο χώρο της, ενεργοποιείται η κίνηση.
Reverse	Αν δώσουμε την τιμή TRUE, τότε η κίνηση θα εκτελείται και αντίστροφα.
Speed	Η ταχύτητα της κίνησης σε texels/δευτερόλεπτο.
Trigger	Το όνομα της σκανδάλης που θα ενεργοποιεί τη συγκεκριμένη PathFollower.

Οι ιδιότητες της οντότητας PathPoint είναι:

Πεδίο	Περιγραφή
NextPointName	Το όνομα του επόμενου PathPoint.
Origin	Η θέση της οντότητας. Η τιμή συμπληρώνεται αυτόματα.
PointName	Το όνομα του συγκεκριμένου PathPoint.
PointType	Ο τύπος του PathPoint. Η τιμή 0 χρησιμοποιείται για το πρώτο στη σειρά (σημείο έναρξης της κίνησης), η τιμή 1 για κάθε επόμενο. Αν δώσουμε την τιμή 2, τότε προσδιορίζουμε μία σφαίρα με κέντρο το PathPoint. Μέσα στην ακτίνα της σφαίρας αυτής μπορεί να κινηθεί η PathFollower.
Range	Η ακτίνα σε texels μέσα στην οποία περιορίζεται η κίνηση, σε περίπτωση που δώσαμε την τιμή 2 στο πεδίο PointType.

Όταν χρησιμοποιούμε την οντότητα PathFollower, πρέπει να έχουμε υπόψη μας τα εξής:

1. Αν χρησιμοποιήσουμε σε κάποιο PathPoint την τιμή 1 στο πεδίο PointType, τότε πρέπει οπωσδήποτε το πρώτο PathPoint στη σειρά να είναι τύπου 0.
2. Η τιμή 2 στο πεδίο PointType παράγει τυχαίες διαδρομές του actor μέσα στην ακτίνα που έχουμε ορίσει. Πρέπει όμως να λάβουμε υπόψη ότι οι κινήσεις εμφανίζονται προς όλες τις κατευθύνσεις, συνεπώς ο actor μπορεί τελικά να κινείται και προς τα πάνω. Κάτι τέτοιο μπορεί να μην το θέλουμε.
3. Αν στην PathFollower έχουμε δώσει την τιμή TRUE στο πεδίο MotionLoops, τότε στο τελευταίο στη σειρά PathPoint δεν πρέπει να κάνουμε καμία δήλωση στο πεδίο NextPointName.
4. Με την PathFollower δεν μπορεί να περιστραφεί η StaticEntityProxy και κατ'επέκταση ο actor ώστε να "κοιτά" προς την κατεύθυνση κίνησής του (ο actor "κοιτά" πάντα προς την κατεύθυνση που αρχικά ορίσαμε στην StaticEntityProxy).

Για να δώσουμε ένα παράδειγμα χρήσης της PathFollower και των PathPoints, ας υποθέσουμε ότι θέλουμε ο actor Robot.act να κινείται απλά σε μία ευθεία γραμμή. Θα χρειαστούμε μία StaticEntityProxy, μία PathFollower και δύο οντότητες PathPoint. Οι κυριότερες ρυθμίσεις που κάνουμε είναι οι εξής:

Για τη StaticEntityProxy

Πεδίο	Τιμή
ActorRotation	0 0 0
bCanPush	FALSE
NoCollision	FALSE
SubjectToGravity	FALSE
szActorFile	robot.act
szEntityName	movingentityproxy
UseProjectedShadows	TRUE

Για την PathFollower

Πεδίο	Τιμή
CollisionTest	TRUE
EntityName	movingentityproxy
MotionLoops	TRUE
PathFirstNode	p1
Reverse	TRUE
Speed	50

Για το πρώτο PathPoint

Πεδίο	Τιμή
NextPointName	p2
PointName	p1
PointType	0

Για το δεύτερο PathPoint

Πεδίο	Τιμή
NextPointName	-
PointName	p2
PointType	1

Για να κινήσουμε ένα στερεό σε καμπύλη γραμμή με μία MovingPlatform, χρειάζεται να ορίσουμε αρκετά κλειδιά, έτσι ώστε οι διαδοχικές θέσεις του μοντέλου να σχηματίζουν μία καμπύλη. Όμοια, για να κινήσουμε έναν actor σε καμπύλη γραμμή με PathPoints, χρειάζεται ένας ικανός αριθμός τους έτσι τοποθετημένων ώστε να σχηματίζουν την επιθυμητή καμπύλη.

16.3 Δίνοντας κίνηση σε actors που ελέγχονται από την οντότητα Pawn

Στο κεφάλαιο 15.4, όπου παρουσιάσαμε την οντότητα Pawn, αναφέραμε ότι είναι η πιο ευέλικτη οντότητα, αλλά η χρήση της απαιτεί προγραμματιστικές γνώσεις. Πριν προχωρήσουμε στην παρουσίαση ορισμένων στοιχείων για τις εντολές και τη σύνταξή τους, πρέπει να κάνουμε τη σειρά ενεργειών που περιγράψαμε στο κεφάλαιο 15.4.

Ξεκινάμε μία νέα εφαρμογή, εισάγουμε ένα δωμάτιο και τις οντότητες PlayerStart, PlayerSetup και Pawn. Στο αρχείο C:\RealityFactory\Install\pawn.ini έχουμε ορίσει τα βασικά στοιχεία της οντότητας, την οποία και ονομάζουμε Robot (μπορούμε να χρησιμοποιήσουμε αυτούσιο το παράδειγμα που δώσαμε στο κεφάλαιο 15.4 για να χρησιμοποιηθεί ο actor Robot.act). Επίσης, έχουμε αποθηκεύσει στο φάκελο C:\RealityFactory\Scripts ένα αρχείο με το όνομα test.s, που περιέχει μία ομάδα εντολών με το όνομα Start (με σύνταξη όπως στο αντίστοιχο παράδειγμα του κεφαλαίου 15.4).

Παρατηρούμε ότι στις ιδιότητες της Pawn περιέχεται η παράμετρος SpawnPoint, που είναι το σημείο στο οποίο θα εμφανιστεί η οντότητα με την έναρξη της εφαρμογής. Το πεδίο αυτό είναι το αντίστοιχο του πεδίου PathFirstNode της οντότητας PathFollower που παρουσιάσαμε πιο πάνω. Κατ' αντιστοιχία λοιπόν με την οντότητα PathPoint, υπάρχει η οντότητα ScriptPoint, που ορίζει τα σημεία που θα ακολουθήσει κατά την κίνηση της μία οντότητα Pawn. Οι ιδιότητες του ScriptPoint είναι:

Πεδίο	Περιγραφή
Angle	Η γωνία περιστροφής στους άξονες x, y και z του ScriptPoint. Δε χρειάζεται να δώσουμε κάποια τιμή.
NextOrder	Το όνομα της επόμενης ομάδας εντολών που θα εκτελεστεί έτσι όπως το ορίζουμε στο αντίστοιχο αρχείο δέσμης εντολών.
NextPoint	Το όνομα του επόμενου ScriptPoint.
Origin	Η θέση της οντότητας. Η τιμή συμπληρώνεται αυτόματα.
szEntityName	Το όνομα του συγκεκριμένου ScriptPoint.

Για τις ανάγκες του παραδείγματός μας, εισάγουμε στην εφαρμογή μας τρεις οντότητες ScriptPoint και τις τοποθετούμε κατά τέτοιο τρόπο ώστε να σχηματίζεται ένα τρίγωνο. Οι ρυθμίσεις που κάνουμε είναι οι εξής:

Για την Pawn

Πεδίο	Τιμή
Angle	0 0 0
PawnType	Robot
ScriptName	test.s
SpawnOrder	Start
SpawnPoint	s1

Για το πρώτο ScriptPoint

Πεδίο	Τιμή
NextPoint	s2
szEntityName	s1

Για το δεύτερο ScriptPoint

Πεδίο	Τιμή
NextPoint	s3
szEntityName	s2

Για το τρίτο ScriptPoint

Πεδίο	Τιμή
NextPoint	s1
szEntityName	s3

Παρατηρούμε ότι στο τελευταίο ScriptPoint υπάρχει η δήλωση ότι το επόμενο ScriptPoint θα είναι και πάλι το πρώτο στη σειρά. Αυτό γίνεται γιατί δεν υπάρχει κάποιο πεδίο στην οντότητα Pawn που να επιτρέπει την αέναη κίνηση.

Αν εκτελέσουμε την εφαρμογή μας, θα παρατηρήσουμε ότι ο actor δεν ακολουθεί τα ScriptPoints, αλλά περιστρέφεται συνεχώς γύρω από το πρώτο. Αυτό οφείλεται στο γεγονός ότι δεν υπάρχουν οι κατάλληλες εντολές στο αρχείο test.s. Ας δούμε λοιπόν πώς μπορούμε να το διαμορφώσουμε.

Κατ' αρχάς, το αρχείο ξεκινά με ένα άγκιστρο ({) και τελειώνει με άλλο ένα (}). Η πιο απλή λοιπόν μορφή ενός αρχείου δέσμης εντολών είναι:

```
{
}
```

Στο χώρο μέσα στα άγκιστρα περιέχονται ομάδες εντολών που ελέγχουν την οντότητα Pawn. Μπορούν να υπάρχουν όσες ομάδες κρίνουμε απαραίτητο, αλλά όλες πρέπει να περικλείονται από τα άγκιστρα. Οι ομάδες έχουν συγκεκριμένη διάταξη, που είναι η εξής:

```
ΌνομαΟμάδας[()
{

}}
```

Στο όνομα της ομάδας παίζουν ρόλο τα πεζά και τα κεφαλαία γράμματα, δηλαδή το όνομα Start είναι διαφορετικό από το όνομα start. Να σημειώσουμε ότι ποτέ δε χρησιμοποιούμε ελληνικούς χαρακτήρες, ούτε μέσα στο αρχείο δέσμης εντολών αλλά ούτε και για το ίδιο το αρχείο.

Μέσα στην ομάδα εντολών υπάρχουν αυτές καθαυτές οι εντολές προς την Pawn και εκτελούνται γραμμικά, δηλαδή με βάση τη σειρά που τις έχουμε γράψει. Κάθε εντολή έχει διαφορετικό αριθμό παραμέτρων, που η καθεμία απ' αυτές χωρίζεται μ' ένα κόμμα (,). Κάθε εντολή τελειώνει μ' ένα ερωτηματικό (;). Για παράδειγμα:

Η εντολή NextPoint(); δεν έχει καμία παράμετρο.

Η εντολή Delay("idle", 2, ""); έχει τρεις παραμέτρους.

Μία εντολή πρέπει να τη συντάσσουμε με όλες τις παραμέτρους της. Είναι λάθος να παραλείψουμε κάποια από αυτές, γιατί κάτι τέτοιο μπορεί να οδηγήσει σε αδυναμία εκτέλεσης όλου του αρχείου δέσμης εντολών.

Για να δώσουμε ένα απλό παράδειγμα αρχείου δέσμης εντολών, θα μπορούσαμε στο αρχείο test.s να γράψουμε τα παρακάτω:

```
{
Start[ ()
{
RotateToPoint("idle", 90, "");
MoveToPoint("walk", 20, "");
NewOrder("Next");
}]
Next[ ()
{
Jump("jump", 40, false, "ding.wav");
RestartOrder();
}]
}
```

Εκτελώντας την εφαρμογή, θα δούμε ότι ο actor θα περιστραφεί προς το πρώτο ScriptPoint εκτελώντας το animation idle και θα κινηθεί προς αυτό εκτελώντας το animation walk. Μόλις φτάσει στο ScriptPoint, θα αρχίσει να εκτελείται η ομάδα εντολών Next, όπου ο actor θα περάσει στο animation jump και θα ακούγεται ο ήχος ding.wav. Επειδή η τελευταία γραμμή της ομάδας περιέχει την εντολή επανεκτέλεσης (Restart-Order), ο actor θα συνεχίσει να πηδά.

Στο συγκεκριμένο παράδειγμα χρησιμοποιήσαμε τις εντολές μίας κατηγορίας, που ονομάζεται Εντολές Υψηλού Επιπέδου (*High Level Commands*). Οι εντολές αυτές μπορεί να χρειάζονται αρκετά δευτερόλεπτα για την εκτέλεσή τους. Για παράδειγμα, ο χρόνος εκτέλεσης της εντολής MoveToPoint εξαρτάται από την απόσταση που βρίσκεται το ScriptPoint. Μόνο όταν ολοκληρωθεί η εντολή ο έλεγχος περνά στην επόμενη στη σειρά. Αντίθετα, οι εντολές της κατηγορίας Χαμηλού Επιπέδου (*Low Level Commands*) εκτελούνται η μία μετά την άλλη σ' ένα μόνο καρέ.

Στη συνέχεια του κεφαλαίου θα ασχοληθούμε με μερικές μόνο εντολές υψηλού επιπέδου, και κυρίως με αυτές που αφορούν την κίνηση. Ο ενδιαφερόμενος μπορεί να βρει την πλήρη τεκμηρίωση όλων των εντολών στο φάκελο C:\RealityFactory\Docs, κάνοντας διπλό κλικ στο αρχείο index.htm (ανοίγει με τον Internet Explorer).

16.3.1 Εντολές αρχικοποίησης

Εντολή: StepHeight(δεκαδικός);

Παράδειγμα: StepHeight(40);

Επεξήγηση: Πόσο ψηλά, σε texels, μπορεί να ανέβει η Pawn μ' ένα βήμα.

Εντολή: Gravity(true/false);

Παράδειγμα: Gravity(false);

Επεξήγηση: Αν η Pawn θα επηρεάζεται από τη βαρύτητα. Προεπιλεγμένο είναι να επηρεάζεται.

Εντολή: IsPushable(true/false);

Παράδειγμα: IsPushable(true);

Επεξήγηση: Αν ο χρήστης μπορεί να σπρώξει την Pawn.

Εντολή: IsVehicle(true/false);

Παράδειγμα: IsVehicle(false);

Επεξήγηση: Αν θα μπορούμε να χρησιμοποιήσουμε την Pawn σαν όχημα.

Εντολή: SetNoCollision(); ή SetCollision();

Παράδειγμα: SetNoCollision(); ή SetCollision();

Επεξήγηση: Να μη γίνεται ή να γίνεται υπολογισμός συγκρούσεων της Pawn με άλλα αντικείμενα.

16.3.2 Εντολές κίνησης

Εντολή: Move(“όνομα_animation”, δεκαδικός_ταχύτητα, δεκαδικός_απόσταση, δεκαδικός_γωνία_x, δεκαδικός_γωνία_y, δεκαδικός_γωνία_z, “όνομα_αρχείου_ήχου”);

Παράδειγμα: Move(“walk”, 50, 200, 5, 0, 10, “ding.wav”);/Move(“”, 50, 200, 0, 0, 0, “”);

Επεξήγηση: Κίνηση της Pawn με ορισμένη ταχύτητα και για ορισμένη απόσταση σε texels. Η κατεύθυνση κίνησης είναι αυτή που προκύπτει από το συνδυασμό των μοιρών στους άξονες x, y και z. Εκτελείται επίσης κάποιο αρχείο ήχου. Αν δεν επιθυμούμε να ορίσουμε animation ή αρχείο ήχου, στη θέση των παραμέτρων αυτών χρησιμοποιούμε “”.

Εντολή: MoveForward(“όνομα_animation”, δεκαδικός_ταχύτητα, δεκαδικός_απόσταση, “όνομα_αρχείου_ήχου”);

Παράδειγμα: MoveForward(“walk”, 100, 500, “”);

Επεξήγηση: Κίνηση της Pawn μπροστά με ορισμένη ταχύτητα και για ορισμένη απόσταση σε texels, εκτελώντας κάποιο αρχείο ήχου.

Με όμοια σύνταξη υπάρχουν οι εντολές: MoveBackward για κίνηση προς τα πίσω, MoveRight για κίνηση δεξιά και MoveLeft για κίνηση αριστερά.

Εντολή: MoveToPoint(“όνομα_animation”, δεκαδικός_ταχύτητα, “όνομα_αρχείου_ήχου”);

Παράδειγμα: MoveToPoint(“walk”, 100, “ding.wav”);

Επεξήγηση: Κίνηση της Pawn προς το τρέχον ScriptPoint με ορισμένη ταχύτητα και εκτελώντας κάποιο αρχείο ήχου.

Εντολή: MoveToTarget(“όνομα_animation”, δεκαδικός_ταχύτητα, “όνομα_αρχείου_ήχου”);

Παράδειγμα: MoveToTarget(“jump”, 100, “”);

Επεξήγηση: Κίνηση της Pawn προς τον τρέχοντα στόχο με ορισμένη ταχύτητα και εκτελώντας κάποιο αρχείο ήχου.

Εντολή: TeleportToPoint(“όνομα_ScriptPoint”); ή TeleportToPoint (“όνομα_ScriptPoint”, δεκαδικός_OffsetX, δεκαδικός_OffsetY, δεκαδικός_OffsetZ);

Παράδειγμα: TeleportToPoint(“s1”); ή TeleportToPoint(“s1”, 10, 0, 0);

Επεξήγηση: Τηλεμεταφέρει την Pawn σε ορισμένο ScriptPoint ή τηλεμεταφέρει την Pawn σε ορισμένο ScriptPoint εφαρμόζοντας μετατόπιση στους άξονες x, y και z.

Εντολή: TeleportToPoint(“όνομα_ScriptPoint”, δεκαδικός_Offset_X, δεκαδικός_Offset_Y, δεκαδικός_Offset_Z, true/false);

Παράδειγμα: TeleportToPoint(“s1”, 10, 0, 0, true);

Επεξήγηση: Όμοια με την προηγούμενη εντολή, με την προσθήκη μίας ακόμα παραμέτρου. Αν σ’ αυτή την τελευταία παράμετρο δώσουμε την τιμή true, τότε ο προσανατολισμός της Pawn είναι αυτός του πεδίου Angle στις ιδιότητες του ScriptPoint.

Εντολή: TeleportToPoint(“όνομα_ScriptPoint”, δεκαδικός_OffsetX, δεκαδικός_OffsetY, δεκαδικός_OffsetZ, true/false, “όνομα_οντότητας”);

Παράδειγμα: TeleportToPoint(“s1”, 10, 0, 0, true, “player”);

Επεξήγηση: Όμοια με την προηγούμενη εντολή, με την προσθήκη μίας ακόμα παραμέτρου. Η παράμετρος αυτή μάς επιτρέπει να τηλεμεταφέρουμε και άλλη οντότητα. Αν δώσουμε την τιμή “player”, τηλεμεταφέρεται ο χρήστης.

16.3.3 Εντολές περιστροφής και πηδήματος

Εντολή: Rotate(“όνομα_animation”, δεκαδικός_ταχύτητα_περιστροφής, δεκαδικός_γωνία_x, δεκαδικός_γωνία_y, δεκαδικός_γωνία_z, “όνομα_αρχείου_ήχου”);

Παράδειγμα: Rotate(“idle”, 50, 0, 45, 0, “ding.wav”);

Επεξήγηση: Περιστρέφει την Pawn με συγκεκριμένο animation και ταχύτητα, στους άξονες x, y και z, παίζοντας κάποιο αρχείο ήχου.

Εντολή: Align();

Παράδειγμα: Align();

Επεξήγηση: Ευθυγραμμίζει την Pawn έτσι, ώστε να κοιτά προς την κατεύθυνση του τρέχοντος ScriptPoint.

Εντολή: RotateToAlign(“όνομα_animation”, δεκαδικός_ταχύτητα_περιστροφής, true/false, “όνομα_αρχείου_ήχου”);

Παράδειγμα: RotateToAlign(“walk”, 50, true, “”);

Επεξήγηση: Περιστρέφει την Pawn έτσι, ώστε να κοιτά προς την κατεύθυνση του τρέχοντος ScriptPoint με συγκεκριμένο animation και ταχύτητα, παίζοντας κάποιο αρχείο ήχου. Αν στην προτελευταία παράμετρο δώσουμε την τιμή true, τότε η περιστροφή γίνεται στους άξονες x και y και όχι μόνο στον άξονα y.

Με όμοια σύνταξη υπάρχει η εντολή: RotateToPlayer που περιστρέφει την Pawn ώστε να κοιτά το χρήστη.

Εντολή: RotateAroundPointRight(“όνομα_animation”, δεκαδικός_ταχύτητα_περιστροφής, δεκαδικός_χρόνος, “όνομα_αρχείου_ήχου”);

Παράδειγμα: RotateAroundPointRight(“walk”, 50, 30, “ding.wav”);

Επεξήγηση: Περιστρέφει δεξιόστροφα την Pawn γύρω από το τρέχον ScriptPoint με συγκεκριμένο animation, ταχύτητα και σε συγκεκριμένο χρόνο σε δευτερόλεπτα. Ταυτόχρονα παίζει και κάποιο αρχείο ήχου.

Με όμοια σύνταξη υπάρχει η εντολή: RotateAroundPointLeft για αριστερόστροφη περιστροφή.

Εντολή: RotateMoveToPoint(“όνομα_animation”, δεκαδικός_ταχύτητα_περιστροφής, δεκαδικός_ταχύτητα, true/false, “όνομα_αρχείου_ήχου”);

Παράδειγμα: RotateMoveToPoint(“idle”, 50, 50, false, “”);

Επεξήγηση: Περιστρέφει την Pawn έτσι, ώστε να κοιτά προς την κατεύθυνση του τρέχοντος ScriptPoint και ταυτόχρονα την κινεί προς αυτό, με συγκεκριμένο animation, ταχύτητα περιστροφής, ταχύτητα κίνησης και παίζοντας κάποιο αρχείο ήχου. Αν στην προτελευταία παράμετρο δώσουμε την τιμή true, τότε η περιστροφή γίνεται στους άξονες x και y και όχι μόνο στον άξονα y.

Με όμοια σύνταξη υπάρχει η εντολή: RotateMoveToTarget που περιστρέφει και κινεί την Pawn προς το στόχο.

Εντολή: RotateMove(“όνομα_animation”, δεκαδικός_ταχύτητα_περιστροφής, δεκαδικός_ταχύτητα, δεκαδικός_γωνία_y, δεκαδικός_γωνία_z, “όνομα_αρχείου_ήχου”);

Παράδειγμα: RotateMove(“walk”, 50, 50, 10, 0, 0, “ding.wav”);

Επεξήγηση: Περιστρέφει την Pawn κατά ορισμένες μοίρες στους άξονες x, y και z με συγκεκριμένη ταχύτητα περιστροφής. Ταυτόχρονα, την κινεί προς τα εμπρός με συγκεκριμένη ταχύτητα και παίζει ένα αρχείο ήχου.

Εντολή: RotateToTarget(“όνομα_animation”, δεκαδικός_ταχύτητα_περιστροφής, true/false, “όνομα_αρχείου_ήχου”);

Παράδειγμα: RotateToTarget(“walk”, 50, true, “”);

Επεξήγηση: Περιστρέφει την Pawn με συγκεκριμένη ταχύτητα περιστροφής και παίζει ένα αρχείο ήχου. Αν στο προτελευταίο πεδίο δώσουμε την τιμή true, τότε η περιστροφή γίνεται με φάλτσο.

Εντολή: FacePlayer(true/false, true/false);

Παράδειγμα: FacePlayer(true, false);

Επεξήγηση: Περιστρέφει την Pawn ώστε πάντα να κοιτά το χρήστη. Αν στην πρώτη παράμετρο δώσουμε την τιμή true, τότε η περιστροφή θα γίνεται με βάση άλλες περι-

στροφές που θα δώσουμε σε επόμενες εντολές. Αν η τιμή της δεύτερης παραμέτρου είναι true, τότε η περιστροφή γίνεται στους άξονες x και y και όχι μόνο στον άξονα y.

Εντολή: Jump(“όνομα_animation”, δεκαδικός_ταχύτητα, true/false, “όνομα_αρχείου_ήχου”);

Παράδειγμα: Jump(“jump”, 50, true, “”);

Επεξήγηση: Η Pawn θα πηδήξει με ορισμένη ταχύτητα και παίζει ένα αρχείο ήχου. Αν στο προτελευταίο πεδίο δώσουμε την τιμή true, τότε η επόμενη εντολή εκτελείται αμέσως ενώ σε αντίθετη περίπτωση, η Pawn θα πρέπει πρώτα να επιστρέψει κάτω.

16.3.4 Εντολές απόστασης και σύγκρουσης

Εντολή: PlayerDistOrder(δεκαδικός_απόσταση, “όνομα_ομάδας_εντολών”);

Παράδειγμα: PlayerDistOrder(200, “neworder”);

Επεξήγηση: Καθορίζει την ελάχιστη απόσταση του χρήστη από την Pawn. Όταν ο χρήστης πλησιάσει σε μικρότερη από την καθορισμένη απόσταση, τότε εκτελείται μία άλλη ομάδα εντολών, η neworder στο παράδειγμά μας. Αν δώσουμε την τιμή 0, αυτός ο έλεγχος δε γίνεται. Αν δώσουμε τιμές κάτω από το 0, τότε καθορίζεται η μέγιστη απόσταση της Pawn από το χρήστη. Όταν εκτελεστεί η νέα ομάδα εντολών, ο έλεγχος απόστασης παύει να ισχύει. Αν σε προηγούμενη εντολή είχαμε χρησιμοποιήσει την εντολή SetFOV (εξηγείται σε παρακάτω παράδειγμα), τότε ο χρήστης πρέπει να βρίσκεται στην ορισμένη απόσταση, αλλά και εντός του πεδίου που η Pawn μπορεί να “δεί”.

Εντολή: DelCollisionOrder();

Παράδειγμα: DelCollisionOrder();

Επεξήγηση: Απενεργοποιεί τον έλεγχο συγκρούσεων της Pawn με το χρήστη.

Εντολή: AddCollisionOrder(“όνομα_ομάδας_εντολών”);

Παράδειγμα: AddCollisionOrder(“neworder”);

Επεξήγηση: Αν η Pawn συγκρουστεί με το χρήστη, τότε εκτελείται κάποια ομάδα εντολών, η neworder στο παράδειγμά μας.

Εντολή: AvoidOrder(“όνομα_ομάδας_εντολών”);

Παράδειγμα: AvoidOrder(“neworder”);

Επεξήγηση: Αν η Pawn συγκρουστεί με κάποιο εμπόδιο καθώς εκτελεί την εντολή MoveToPoint, θα εκτελεστεί μία άλλη ομάδα εντολών. Τη χρησιμοποιούμε για να κατασκευάσουμε δικές μας ρουτίνες αποφυγής εμποδίων.

16.3.5 Εντολές για τα ScriptPoints και τις ομάδες εντολών

Εντολή: RestartOrder();

Παράδειγμα: RestartOrder();

Επεξήγηση: Ξεκινά την τρέχουσα ομάδα εντολών από την αρχή.

Εντολή: NextPoint();

Παράδειγμα: NextPoint();

Επεξήγηση: Τρέχον ScriptPoint γίνεται το επόμενο στη σειρά. Χρησιμοποιείται σε συνδυασμό με άλλες εντολές για να κινηθεί η Pawn στο επόμενο ScriptPoint.

Εντολή: NextOrder();

Παράδειγμα: NextOrder();

Επεξήγηση: Εκτελεί τις εντολές που περιέχονται στην επόμενη ομάδα εντολών.

Εντολή: NewPath("όνομα_ ScriptPoint");

Παράδειγμα: NewPath("s2");

Επεξήγηση: Μεταφέρει την Pawn στο ScriptPoint που προσδιορίζουμε και εκτελεί την ομάδα εντολών που περιέχονται στο πεδίο NextOrder.

Εντολή: NewPoint("όνομα_ ScriptPoint");

Παράδειγμα: NewPoint("s2");

Επεξήγηση: Μεταφέρει την Pawn στο ScriptPoint που προσδιορίζουμε, αλλά η τρέχουσα ομάδα εντολών εξακολουθεί να εκτελείται.

Εντολή: NewOrder("όνομα_ ομάδας_εντολών");

Παράδειγμα: NewOrder("after_Start");

Επεξήγηση: Αρχίζει να εκτελεί τις εντολές που περιέχονται στην ομάδα εντολών που προσδιορίζουμε.

16.3.6 Παραδείγματα αρχείων δέσμης εντολών που σχετίζονται με την κίνηση

Στη συνέχεια δίνουμε μερικά απλά παραδείγματα αρχείων δέσμης εντολών και εξηγούμε τη χρήση τους. Για να ελέγξουμε τα αποτελέσματά τους, αρκεί να κατασκευάσουμε ένα απλό επίπεδο που να περιλαμβάνει τις οντότητες PlayerStart και PlayerSetUp, μία οντότητα Pawn και τρία-τέσσερα ScriptPoints με ονόματα s1, s2, s3, κτλ. Για λόγους ευκολίας, το αρχείο δέσμης εντολών το ονομάζουμε test.s και πάντα σ' αυτό χρησιμοποιούμε την ομάδα εντολών Start ως πρώτη ομάδα (την οποία και δηλώνουμε στο πεδίο SpawnOrder της Pawn). Τέλος, στο αρχείο pawn.ini έχουμε ορίσει τα βασικά στοιχεία της οντότητας Pawn (θα χρησιμοποιήσουμε τον actor robot.act όπως

και στο κεφάλαιο 15.4), την οποία και ονομάζουμε Robot (αυτό το όνομα χρησιμοποιούμε και στο πεδίο PawnType της Pawn).

Χρησιμοποιώντας το “Σημειωματάριο”, γράφουμε τις παρακάτω εντολές:

```
{
Start[ ()
{
NextPoint();
MoveToPoint(“walk”, 100, “”);
RestartOrder();
}]
}
```

Η Pawn εμφανίζεται πάντα στο πρώτο ScriptPoint (s1), ανεξάρτητα με τη θέση που θα την τοποθετήσουμε κατά την κατασκευή της εφαρμογής μας. Με την πρώτη εντολή λοιπόν, τρέχον ScriptPoint γίνεται το s2. Με τη δεύτερη εντολή, η Pawn κινείται προς το τρέχον ScriptPoint, που είναι πια το s2, άρα εκτελείται η διαδρομή s1-s2. Αν δεν υπήρχε η τρίτη εντολή, όταν η Pawn θα έφτανε στο s2 θα σταματούσε. Με την εντολή όμως RestartOrder, οι εντολές της ομάδας εκτελούνται και πάλι από την αρχή, τρέχον ScriptPoint γίνεται το s3 και η Pawn εκτελεί τη διαδρομή s2-s3.

Παρατηρούμε όμως ότι δεν υπήρξε περιστροφή της Pawn ώστε να κοιτά το σημείο προορισμού. Χρειάζεται να προσθέσουμε μία ακόμα εντολή που να κάνει ακριβώς αυτό. Συμπληρώνουμε λοιπόν στο παραπάνω αρχείο τη γραμμή:

```
...
NextPoint();
RotateMoveToPoint(“walk”, 100, 100, false, “”);
MoveToPoint(“walk”, 100, “”);
...
```

Για να ολοκληρωθεί η κίνηση, καλό είναι να προσθέσουμε και έναν ήχο βημάτων. Για το σκοπό αυτό, θα χρησιμοποιήσουμε το αρχείο ήχου m1.wav που βρίσκεται στο φάκελο C:\RealityFactory\Media\footsteps και θα προσθέσουμε την παρακάτω γραμμή, μάλιστα ως πρώτη εντολή:

...

```
AddRandomSound(1, 0.6, 0.6, "footsteps\\m1.wav");  
NextPoint();
```

...

Το σημαντικό σ' αυτή την εντολή είναι ότι χρησιμοποιούμε δύο \ και όχι μία \ για να δώσουμε τη διαδρομή του φακέλου που είναι ο ήχος. Προτιμούμε να δηλώσουμε τον ήχο σε ξεχωριστή εντολή και όχι μέσα στη RotateMoveToPoint ή τη MoveToPoint, παρότι θα μπορούσαμε. Αυτό γιατί ο ήχος θα έπαιζε στη θέση του εκάστοτε ενεργού ScriptPoint και όχι μαζί με την Pawn καθώς κινείται.

Για να προσθέσουμε ένα ακόμα εφέ, θα κάνουμε την Pawn να πηδά 100 texels όταν φτάνει σε κάθε ScriptPoint, προσθέτοντας τις παρακάτω γραμμές:

...

```
Gravity(true);  
AddRandomSound(1, 0.6, 0.6, "footsteps\\m1.wav");  
NextPoint();  
RotateMoveToPoint("walk", 100, 100, false, "");  
Jump("jump", 100, false, "");  
MoveToPoint("walk", 100, "");
```

...

Το επόμενο παράδειγμα εισάγει δύο νέα στοιχεία. Το πρώτο είναι η χρήση της εντολής random που μας επιτρέπει να παράγουμε τυχαίους αριθμούς. Το δεύτερο είναι η χρήση της εντολής if. Η εντολή αυτή, μαζί με την else, μας επιτρέπουν να ελέγξουμε αν μία παράμετρος έχει πάρει κάποια τιμή, αν είναι αληθής ή ψευδής και ανάλογα με την περίπτωση να εκτελεστούν κάποιες εντολές. Στο παράδειγμά μας θα υλοποιήσουμε την επιλογή τυχαίας διαδρομής με τη χρήση των παρακάτω εντολών:

...

```
AddRandomSound(1, 0.6, 0.6, "footsteps\\m1.wav");  
NextPoint();  
  
    if(random(1,10)<5)  
    {  
        NextPoint();  
    }  
    if(random(1,10)<3)  
    {  
        NextPoint();  
    }
```

```

}
RotateMoveToPoint("walk", 100, 100, false, "");
...

```

Η εντολή `random` παράγει ένα τυχαίο αριθμό που βρίσκεται μέσα στο εύρος που δίνουμε στην παρένθεση (για το παράδειγμά μας από το 1 ως το 10). Με το πρώτο `if`, αν ο αριθμός είναι μικρότερος του 5, τότε η Pawn παρακάμπτει ένα `ScriptPoint` (αφήνει το `s2` και πάει στο `s3`). Με το δεύτερο `if`, αν ο αριθμός είναι και μικρότερος του 3, τότε η Pawn παρακάμπτει ακόμα ένα `ScriptPoint` (αφήνει και το `s3` και πάει στο `s4`). Με τη `RestartOrder` εκτελούνται από την αρχή οι εντολές, αλλά εξαιτίας των `if` δεν είναι σίγουρο ότι η Pawn βρίσκεται στο `s2` (μπορεί να είναι στο `s3` ή στο `s4`). Παράγεται εκ νέου ένας τυχαίος αριθμός και συνεπώς ο νέος προορισμός δεν είναι απαραίτητα το επόμενο στη σειρά `ScriptPoint`.

Στο επόμενο παράδειγμα θα χρησιμοποιήσουμε δύο ομάδες εντολών, μία για την αρχικοποίηση της Pawn και μία δεύτερη που θα ενεργοποιείται αν συμβεί κάποιο γεγονός. Ο σκοπός μας είναι να κάνουμε την Pawn να πηδά όταν θα “δει” το χρήστη. Για το σκοπό αυτό θα χρησιμοποιήσουμε μία εντολή που καθορίζει το οπτικό πεδίο της Pawn (Field of View, FOW). Με άλλη εντολή θα εντάξουμε το χρήστη στην ομάδα των στόχων της Pawn. Γράφουμε λοιπόν τις παρακάτω εντολές:

```

{
Start[ ()
    {
        Gravity(true);
        SetFOV(180);
        HostilePlayer(true);
        FindTargetOrder(300, "FoundTarget", "");
    }
]

FoundTarget[ ()
    {
        Jump("jump", 100, false, "");
        RestartOrder();
    }
]
}

```

Με τη δεύτερη και την τρίτη εντολή της ομάδας Start, ορίζουμε ότι η Pawн θα έχει οπτικό πεδίο 180° και ότι ο χρήστης ανήκει στους “στόχους” της. Με την τέταρτη εντολή, όταν βρεθεί κάποιος στόχος σε απόσταση 300 texels, ενεργοποιείται η ομάδα εντολών FoundTarget.

Στο τελευταίο παράδειγμα θα συνδυάσουμε όλα τα προηγούμενα για να κάνουμε την Pawн να εκτελεί μία τυχαία διαδρομή μεταξύ των ScriptPoints. Όταν όμως ο χρήστης μπει στο οπτικό της πεδίο και σε απόσταση μικρότερη των 300 texels, θα τον ακολουθεί μέχρις ότου βγει και πάλι εκτός του οπτικού της πεδίου ή η απόσταση γίνει μεγαλύτερη των 300 texels. Γράφουμε λοιπόν τις παρακάτω εντολές:

```
{
Start[ ()
{
    SetCollision();
    Gravity(true);
    SetFOV(180);
    HostilePlayer(true);
    NewOrder("Patrol");
}]
Patrol[ ()
{
    AddRandomSound(1, 0.6, 0.6, "footsteps\\m1.wav");
    FindTargetOrder(300, "FoundTarget", "");
    NextPoint();
    if(random(1,10)<5)
    {
        NextPoint();
    }
    if(random(1,10)<3)
    {
        NextPoint();
    }
    RotateMoveToPoint("walk", 100, 100, false, "");
    MoveToPoint("walk", 100, "");
    RestartOrder();
}]
```

```
FoundTarget[ ()  
    {  
        RotateMoveToTarget("walk", 100, 100, false, "");  
        MoveToTarget("walk", 100, "");  
        NewOrder("Patrol");  
        RestartOrder();  
    }]  
}
```

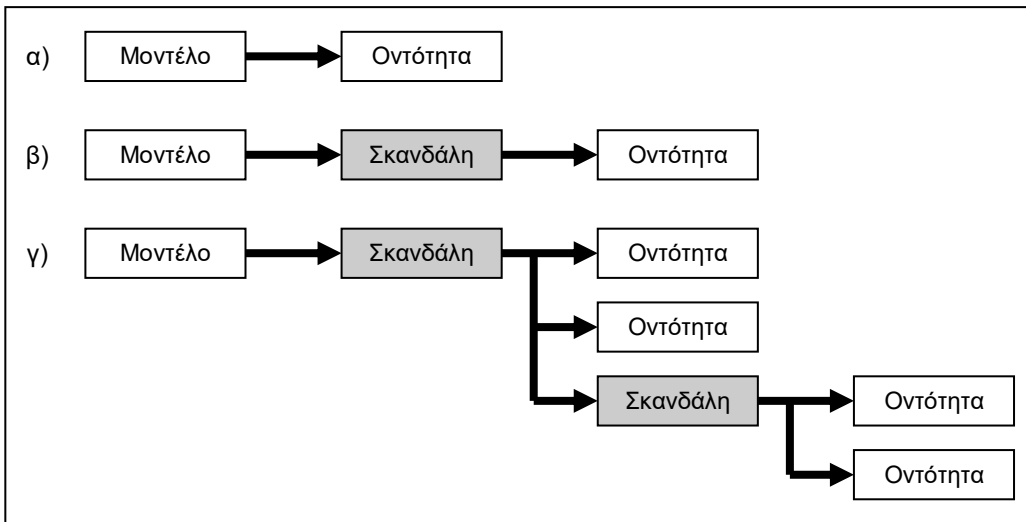

Κεφάλαιο 17°

Χειρισμός στερεών και οντοτήτων με σκανδάλες

Στο προηγούμενο κεφάλαιο είδαμε ότι τα μοντέλα και οι οντότητες χειρισμού των τρισδιάστατων μοντέλων (actors) μπορούν να κινηθούν με την έναρξη της εφαρμογής, όταν ο χρήστης ακουμπήσει πάνω σε αυτά ή όταν πατήσει το πλήκτρο χρήσης. Υπάρχουν όμως πιο περίπλοκα σενάρια που δεν μπορούν να υλοποιηθούν με όσα έχουμε παρουσιάσει. Για παράδειγμα, μπορεί να θέλουμε η κίνηση να αρχίζει όταν ο χρήστης έρθει σε επαφή μ' ένα τρίτο μοντέλο, διαφορετικό απ' αυτό με το οποίο συνδέεται η οντότητα κίνησης. Ή ακόμα, μπορεί να θέλουμε να αρχίζει η κίνηση δύο ή περισσότερων μοντέλων ταυτόχρονα. Για να μην περιοριστούμε μόνο σε σενάρια που περιλαμβάνουν κίνηση, μπορεί να θέλουμε τα φώτα σ' ένα δωμάτιο να μην είναι διαρκώς αναμμένα, αλλά να μπορεί ο χρήστης να τα ανάβει και να τα σβήνει κατά βούληση, ή κάτι αντίστοιχο να συμβαίνει με τους ήχους.

Το πρόβλημα, πάντα με όσα έχουμε παρουσιάσει, είναι ότι μία οντότητα είτε είναι συνεχώς ενεργή είτε η ενεργοποίησή της εξαρτάται άμεσα από το μοντέλο με το οποίο είναι συνδεδεμένη, υπάρχει δηλαδή μία σχέση ένα-προς-ένα (σχήμα 17.1α). Στο κεφάλαιο αυτό θα γνωρίσουμε ίσως τη σημαντικότερη οντότητα του Reality Factory, τη σκανδάλη (*Trigger*). Η βασική της εργασία είναι να κάνει αυτό που υποδηλώνει και το όνομά της, να ενεργοποιεί δηλαδή οντότητες. Παρεμβάλλεται μεταξύ του μοντέλου και της οντότητας, λειτουργεί ως διάμεσο. Ο χρήστης, ερχόμενος σε επαφή με το μοντέλο δεν ενεργοποιεί την οντότητα, αλλά τη σκανδάλη. Η σκανδάλη με τη σειρά της θα ενεργοποιήσει την οντότητα (σχήμα 17.1β). Για να είμαστε πιο ακριβείς, η σκανδάλη δεν ενεργοποιεί η ίδια άλλες οντότητες. Το μόνο που κάνει είναι να αλλάζει την κατάσταση της από ανενεργή σε ενεργή και το αντίστροφο. Η οντότητα που συνδέεται με αυτή “διαβάζει” την κατάστασή της και αναλόγως ενεργοποιείται ή απενεργοποιείται.

Θα αναρωτηθεί κανείς για ποιο λόγο να κάνουμε την ενεργοποίηση των οντοτήτων πιο σύνθετη χρησιμοποιώντας σκανδάλες. Ο λόγος είναι ότι πολλές οντότητες ταυτόχρονα μπορούν να “διαβάζουν” την κατάσταση μίας σκανδάλης. Ακριβώς σ' αυτό το σημείο βρίσκεται και η μεγάλη της σημασία. Δεν είναι δηλαδή δυνατή η ενεργοποίηση μόνο μίας οντότητας αλλά πολλών, και μάλιστα μεταξύ αυτών μπορεί να είναι άλλες σκανδάλες, που με τη σειρά τους είναι υπεύθυνες για την ενεργοποίηση άλλων οντοτήτων. Με τις σκανδάλες δημιουργούμε σχέσεις ενός-προς-πολλά ή πολλών-προς-πολλά, μεταξύ μοντέλων και οντοτήτων ή μεταξύ οντοτήτων. Μπορούμε λοιπόν να κατασκευάσουμε αλυσίδες γεγονότων που είναι δυνατό να ξεκινήσουν ή να διακοπούν από την ενεργοποίηση μίας και μόνο σκανδάλης (σχήμα 17.1γ).



Σχήμα 17.1 Η χρήση της Trigger

α) ενεργοποίηση οντότητας χωρίς σκανδάλη, β) απλή ενεργοποίηση οντότητας με σκανδάλη
 γ) πολλαπλή ενεργοποίηση οντοτήτων με σκανδάλες

Οι ιδιότητες της Trigger είναι:

Πεδίο	Περιγραφή
AnimationSpeed	Η ταχύτητα του animation του συνδεδεμένου με την Trigger μοντέλου. Η ρύθμιση αυτή παρακάμπτει τη διάρκεια που έχει προκύψει όταν ορίζαμε τα σημεία-κλειδιά του animation στο μοντέλο.
bAudioLoops	Αν δώσουμε την τιμή TRUE, τότε ο συνδεδεμένος με την Trigger ήχος θα παίζει αέναα.
bNoCollide	Αν δώσουμε την τιμή TRUE, τότε η Trigger δε θα ενεργοποιείται από την επαφή του χρήστη μ' ένα μοντέλο.
bOneShot	Αν δώσουμε την τιμή TRUE, τότε η Trigger θα μπορεί να ενεργοποιηθεί μόνο μία φορά.
bShoot	Αν δώσουμε την τιμή TRUE, τότε η Trigger θα μπορεί να ενεργοποιηθεί μόνο αν ο χρήστης την πυροβολήσει.
Model	Το όνομα του μοντέλου στο οποίο συνδέεται η οντότητα Trigger.
Origin	Η θέση της οντότητας. Η τιμή συμπληρώνεται αυτόματα.
PlayerOnly	Αν δώσουμε την τιμή TRUE, τότε η Trigger θα μπορεί να ενεργοποιηθεί μόνο από το χρήστη.

Πεδίο	Περιγραφή
szEntityName	Το όνομα της συγκεκριμένης οντότητας Trigger. Αυτό το όνομα θα χρησιμοποιηθεί σε πεδία ιδιοτήτων άλλων οντοτήτων που θέλουμε να συνδέονται με αυτή.
szSoundFile	Το αρχείο ήχου που παίζει όταν πυροβοληθεί η Trigger. <u>Προσοχή</u> , δεν παίζει ήχος όταν ενεργοποιείται η Trigger.
TimeOff	Πόσο χρόνο σε δευτερόλεπτα θα παραμένει ανενεργή η Trigger, μέχρις ότου να μπορεί να ενεργοποιηθεί και πάλι.
TimeOn	Πόσο χρόνο σε δευτερόλεπτα θα παραμένει ενεργή η Trigger όταν ενεργοποιηθεί.
TriggerName	Το όνομα άλλης σκανδάλης που ενεργοποιεί τη συγκεκριμένη οντότητα Trigger.
UseKey	Αν δώσουμε την τιμή TRUE, τότε η Trigger θα ενεργοποιείται όταν ο χρήστης πατήσει το πλήκτρο χρήσης.

17.1 Η οντότητα AreaSwitch

Οι σκανδάλες μπορούν να ενεργοποιηθούν με τρεις τρόπους: α) με την επαφή του χρήστη σ' ένα μοντέλο, β) από άλλες σκανδάλες και γ) από μία οντότητα που ονομάζεται AreaSwitch. Την οντότητα αυτή θα εξετάσουμε στη συνέχεια.

Η AreaSwitch ελέγχει αν ο χρήστης ή άλλη οντότητα που εμείς ορίζουμε (για παράδειγμα μία Pawn) είναι ή δεν είναι μέσα στο χώρο της. Για όσο διάστημα δεν είναι, η κατάσταση της σκανδάλης που ελέγχει η AreaSwitch είναι ανενεργή (ή ενεργή αν έτσι το ορίσουμε). Μόλις όμως ο χρήστης (ή η οντότητα) μπει στο χώρο της AreaSwitch, η κατάστασή της σκανδάλης αντιστρέφεται. Όταν βγει, η κατάσταση της σκανδάλης επιστρέφει στην πρώτη της κατάσταση. Οι ιδιότητες της AreaSwitch είναι:

Πεδίο	Περιγραφή
Diameter	Η διάμετρος που έχει η περιοχή της AreaSwitch.
Entity	Ποια οντότητα θα ελέγχει αν θα μπει στην περιοχή της. Αν αφήσουμε το πεδίο κενό, εννοείται ο χρήστης.
InState	Ποια θα είναι η κατάσταση της σκανδάλης για όσο χρόνο κάποια οντότητα είναι μέσα στο χώρο της AreaSwitch. Αν δώσουμε την τιμή TRUE, τότε η σκανδάλη θα είναι ενεργή (On).
Trigger	Το όνομα της σκανδάλης που θα ενεργοποιείται απ' αυτή την AreaSwitch.

17.2 Παραδείγματα χρήσης της Trigger

Στο σημείο αυτό θα δώσουμε μία σειρά παραδειγμάτων χρήσης της Trigger για να γίνει κατανοητή η χρήση της. Για την υλοποίησή τους θα χρειαστούμε ένα απλό δωμάτιο και τις οντότητες PlayerStart και PlayerSetup.

17.2.1 Ενεργοποίηση DynamicLight μέσω σκανδάλης και AreaSwitch

Στο πρώτο μας παράδειγμα θα υλοποιήσουμε την ενεργοποίηση μίας DynamicLight μέσω μίας σκανδάλης. Η σκανδάλη με τη σειρά της θα ενεργοποιείται από μία AreaSwitch. Εισάγουμε λοιπόν στο δωμάτιό μας μία DynamicLight, μία AreaSwitch και μία Trigger και τις παραμετροποιούμε με τον ακόλουθο τρόπο:

AreaSwitch:		Trigger:		DynamicLight:	
Πεδίο	Τιμή	Πεδίο	Τιμή	Πεδίο	Τιμή
Diameter	200	bNoCollide	TRUE	Color	255 0 0
InState	TRUE	Model	-	InterpolateVlues	TRUE
Trigger	tr	PlayerOnly	FALSE	MaxRadius	500
		szEntityName	tr	MinRadius	200
		TimeOff	-	RadiusFunction	aza
		TimeOn	-	RadiusSpeed	1
		UseKey	FALSE	TriggerName	tr

Οι παραπάνω ρυθμίσεις θα λειτουργήσουν ως εξής: Μόλις ο χρήστης μπει στο χώρο της AreaSwitch, ενεργοποιείται η σκανδάλη. Σ' αυτή, επειδή η ενεργοποίησή της δεν εξαρτάται από την επαφή του χρήστη με κάποιο στερεό, έχουμε εισαγάγει στα πεδία bNoCollide, PlayerOnly και UseKey τις τιμές TRUE, FALSE και FALSE αντίστοιχα. Επίσης, επειδή η ενεργοποίηση της σκανδάλης εξαρτάται από την AreaSwitch, δεν είναι απαραίτητο να εισαγάγουμε τιμές στα πεδία TimeOff και TimeOn, αφού η σκανδάλη θα είναι ενεργή για όσο χρόνο παραμένει ο χρήστης στην περιοχή της AreaSwitch. Τέλος, συνδέουμε τη DynamicLight με τη σκανδάλη, αφού έχουμε συμπληρώσει το πεδίο TriggerName με το όνομά της. Όταν λοιπόν ενεργοποιηθεί η σκανδάλη, θα ενεργοποιηθεί με τη σειρά της και η DynamicLight.

17.2.2 Ενεργοποίηση DynamicLight και AudioSource3D μέσω σκανδάλης και μοντέλου

Θα τροποποιήσουμε το προηγούμενο παράδειγμα, έτσι ώστε η ενεργοποίηση της σκανδάλης να γίνεται με την επαφή του χρήστη σ' ένα μοντέλο ή με το πάτημα του πλήκτρου "χρήσης". Διαγράφουμε την AreaSwitch και τοποθετούμε ένα στερεό κοντά στη DynamicLight. Το στερεό το μετατρέπουμε σε μοντέλο με το όνομα Model. Για να γίνει το παράδειγμα λίγο πιο σύνθετο, εισάγουμε και μετακινούμε κοντά στο μοντέλο

μία οντότητα AudioSource3D η οποία θα παίζει ένα αρχείο ήχου. Η ενεργοποίησή της θα γίνεται ταυτόχρονα με τη DynamicLight από την ίδια σκανδάλη. Παραμετροποιούμε τις οντότητες ως εξής:

Trigger:		AudioSource3D:		DynamicLight:	
Πεδίο	Τιμή	Πεδίο	Τιμή	Πεδίο	Τιμή
bNoCollide	FALSE	bLoopSound	TRUE	Color	255 0 0
Model	Model	fRadius	200	InterpolateVlues	TRUE
PlayerOnly	FALSE	Model	-	MaxRadius	500
szEntityName	tr	szSoundFile	ding.wav*	MinRadius	200
TimeOff	1	TriggerName	tr	RadiusFunction	aza
TimeOn	5			RadiusSpeed	1
UseKey	TRUE			TriggerName	tr

* Το αρχείο περιλαμβάνεται στην εγκατάσταση του Reality Factory

Το σημαντικό στοιχείο στο παράδειγμα είναι ότι εισάγουμε τιμές στα πεδία TimeOff και TimeOn της σκανδάλης. Αν αφήναμε αυτά τα πεδία κενά, τότε η ενεργοποίηση και απενεργοποίηση της σκανδάλης θα ήταν ακαριαίες, με αποτέλεσμα το φως και ο ήχος να μην εκτελούνται. Αν εισάγουμε μεγάλη τιμή στο πεδίο TimeOff, δε θα είναι δυνατή η εκ νέου ενεργοποίηση της σκανδάλης πριν περάσει αυτό το διάστημα.

17.2.3 Ενεργοποίηση μίας MovingPlatform μέσω σκανδάλης

Στο επόμενο παράδειγμά μας η σκανδάλη θα ενεργοποιείται με την επαφή του χρήστη ή με τη “χρήση” ενός μοντέλου που θα αντιπροσωπεύει ένα διακόπτη δίπλα σε μία πόρτα. Η σκανδάλη, με τη σειρά της, θα ενεργοποιεί μία MovingPlatform που θα κινεί την πόρτα ώστε αυτή ν’ ανοίξει. Με άλλα λόγια, η πόρτα δε θα ανοίγει άμεσα με την επαφή του χρήστη με αυτή, αλλά μόνο εφόσον χρησιμοποιήσουμε το διακόπτη.

Κατασκευάζουμε δύο δωμάτια, τη μεταξύ τους ένωση και μία πόρτα σύμφωνα με όσα αναφέραμε στο κεφάλαιο 16.1.1 και δίνουμε το όνομα DoorModel στο μοντέλο της πόρτας. Χρησιμοποιούμε τρία κλειδιά κίνησης (στο χρόνο 0 κλειστή η πόρτα, στο χρόνο 3 ανοιχτή και στο χρόνο 6 και πάλι κλειστή). Κοντά σ’ αυτή, στον τοίχο, εισάγουμε ένα στερεό με μικρές διαστάσεις, που θα είναι ο διακόπτης, το μετατρέπουμε σε μοντέλο με το όνομα SwitchModel και του δίνουμε τρία κλειδιά κίνησης. Το πρώτο στην αρχική θέση στο χρόνο 0, το δεύτερο στο χρόνο 0.5, που θα είναι βυθισμένο στον τοίχο, και το τρίτο στο χρόνο 1, που θα επιστρέφει στην αρχική θέση. Εισάγουμε επίσης μία MovingPlatform και μία σκανδάλη. Οι τιμές που δίνουμε στα πεδία των οντοτήτων είναι οι εξής:

Στη σκανδάλη:

Πεδίο	Τιμή
bNoCollide	FALSE
Model	SwitchModel
PlayerOnly	TRUE
szEntityName	DoorTrigger
TimeOff	-
TimeOn	-
UseKey	TRUE

Στη MovingPlatform:

Πεδίο	Τιμή	Πεδίο	Τιμή
bLooping	TRUE	PlayerOnly	FALSE
bNoCollide	TRUE	szSoundFile	ding.wav*
bReverse	FALSE	TimeEachTrigger	3
bRunTimed	TRUE	TriggerName	DoorTrigger
Model	DoorModel	UseKey	FALSE

* Το αρχείο περιλαμβάνεται στην εγκατάσταση του Reality Factory

Παρατηρούμε ότι στη MovingPlatform δώσαμε την τιμή FALSE στο πεδίο bReverse, γιατί στο μοντέλο της πόρτας το τρίτο κλειδί είναι η επιστροφή στην αρχική θέση. Αν παραλείπαμε αυτό το κλειδί, θα μπορούσαμε στο πεδίο bReverse να δώσουμε την τιμή TRUE. Δώσαμε επίσης την τιμή TRUE στο πεδίο bLooping, γιατί θέλουμε η πόρτα να μπορεί να κινηθεί όσες φορές θέλουμε. Αν δίναμε την τιμή FALSE, η πόρτα θα μπορούσε να ανοίξει και να κλείσει μόνο μία φορά.

Η MovingPlatform μας δίνει τη δυνατότητα να εκτελεί και έναν ήχο για όσο διάστημα θα είναι ενεργή. Για το λόγο αυτό εισαγάγαμε έναν στο πεδίο szSoundFile. Έτσι, δεν είναι απαραίτητο να εισαγάγουμε στην εφαρμογή μας και μία AudioSource3D, που θα έπαιζε τον ήχο που κάνει η πόρτα καθώς ανοιγοκλείνει.

Το πιο σημαντικό στοιχείο στην παραμετροποίηση των οντοτήτων είναι ότι αναθέσαμε το πόσο χρόνο θα κινείται το μοντέλο της πόρτας στη MovingPlatform και όχι στη σκανδάλη. Αυτό το επιτυγχάνουμε αφήνοντας τα πεδία TimeOff και TimeOn της σκανδάλης κενά και δίνοντας την τιμή TRUE στο πεδίο bRunTimed της MovingPlatform. Δώσαμε μάλιστα την τιμή 3 στο πεδίο TimeEachTrigger, που είναι ίση με το χρόνο που κάνει το μοντέλο της πόρτας για να ανοίξει ή να κλείσει. Έτσι, ο χρήστης πρέπει να πατήσει το διακόπτη μία φορά για να ανοίξει η πόρτα και άλλη μία για να κλείσει. Αυτή η τεχνική μάς απαλλάσσει από την ανάγκη να εισάγουμε επιπλέον κλειδιά κίνησης.

σης όπου το μοντέλο παραμένει ακίνητο σε μία θέση, έτσι ώστε ο χρήστης να έχει αρκετό χρόνο να περάσει από την πόρτα, όπως κάναμε στο κεφάλαιο 16.1.1.

Γενικά, όταν μία σκανδάλη πρόκειται να θέσει σε κίνηση μία MovingPlatform, αφήνουμε τα πεδία TimeOff και TimeOn κενά και αναθέτουμε το χρόνο κίνησης στην ίδια τη MovingPlatform. Αυτό το κάνουμε γιατί παρατηρούνται αστοχίες και προβλήματα στην ενεργοποίηση της MovingPlatform από τη σκανδάλη όταν υπάρχουν τιμές στα πεδία TimeOff και TimeOn.

17.3 Η οντότητα LogicGate

Η κατάσταση στην οποία βρίσκεται μία σκανδάλη εξαρτάται από το χρόνο που έχουμε εισαγάγει στα πεδία TimeOff και TimeOn και από το εάν έχει ενεργοποιηθεί από κάποιο μοντέλο ή από μία AreaSwitch. Όταν όμως θέλουμε να υλοποιήσουμε την εκτέλεση πιο περίπλοκων γεγονότων, για παράδειγμα γεγονότων που εξαρτώνται από την καθυστέρηση ενεργοποίησης μίας σκανδάλης ή από την ταυτόχρονη ενεργοποίηση δύο ή και περισσότερων, χρησιμοποιούμε μία άλλη οντότητα, τη LogicGate. Οι ιδιότητές της είναι:

Πεδίο	Περιγραφή
Amount	Η ελάχιστη τιμή μίας ιδιότητας του χρήστη που ενεργοποιεί τη LogicGate. Το πεδίο αυτό χρησιμοποιείται σε συνδυασμό με συγκεκριμένη τιμή που θα δώσουμε στο πεδίο Type.
Delay	Ο χρόνος σε δευτερόλεπτα που θα καθυστερήσει η ενεργοποίηση της LogicGate. Το πεδίο αυτό χρησιμοποιείται σε συνδυασμό με συγκεκριμένη τιμή που θα δώσουμε στο πεδίο Type.
Origin	Η θέση της οντότητας. Η τιμή συμπληρώνεται αυτόματα.
szEntityName	Το όνομα της συγκεκριμένης οντότητας LogicGate. Αυτό το όνομα θα χρησιμοποιηθεί σε πεδία ιδιοτήτων άλλων οντοτήτων που θέλουμε να συνδέονται με αυτή.
Trigger1Name	Το όνομα της πρώτης Trigger η κατάσταση της οποίας θα επηρεάσει τη LogicGate.
Trigger2Name	Το όνομα της δεύτερης Trigger η κατάσταση της οποίας θα επηρεάσει τη LogicGate.
Type	Ο τύπος της LogicGate*.

* Οι πιο συνηθισμένοι τύποι της LogicGate είναι:

Τιμή	Περιγραφή
0	Λογικό AND. Πρέπει να χρησιμοποιήσουμε δύο σκανδάλες. Η LogicGate θα μπει σε κατάσταση ON όταν <u>και οι δύο</u> σκανδάλες θα είναι σε κατάσταση ON. Όταν κάποια από τις δύο ή και οι δύο βρίσκονται σε κατάσταση OFF, τότε και η LogicGate είναι σε κατάσταση OFF.
1	DELAY. Χρησιμοποιούμε μία σκανδάλη (συμπληρώνουμε μόνο το πεδίο Trigger1Name). Όταν ενεργοποιηθεί η σκανδάλη, η LogicGate θα μπει σε κατάσταση ON μετά από τον αριθμό των δευτερολέπτων που έχουμε εισαγάγει στο πεδίο Delay.
2	Λογικό NOT. Χρησιμοποιούμε μία σκανδάλη (συμπληρώνουμε μόνο το πεδίο Trigger1Name). Η LogicGate είναι πάντοτε σε κατάσταση αντίστροφη απ' αυτή της σκανδάλης.
3	Λογικό OR. Πρέπει να χρησιμοποιήσουμε δύο σκανδάλες. Η LogicGate θα μπει σε κατάσταση ON όταν μία από τις δύο ή και οι δύο σκανδάλες είναι σε κατάσταση ON. <u>Μόνο</u> όταν και οι δύο είναι σε κατάσταση OFF, τότε και η LogicGate θα μπει σε κατάσταση OFF.
4	Λογικό XOR. Πρέπει να χρησιμοποιήσουμε δύο σκανδάλες. Η LogicGate θα μπει σε κατάσταση ON όταν μία από τις δύο σκανδάλες είναι σε κατάσταση ON. Όταν <u>και οι δύο</u> είναι σε κατάσταση OFF ή <u>και οι δύο</u> είναι σε κατάσταση ON, τότε η LogicGate είναι σε κατάσταση OFF.
5	FLIPFLOP. Χρησιμοποιούμε μία σκανδάλη (συμπληρώνουμε μόνο το πεδίο Trigger1Name). Όταν η κατάσταση της σκανδάλης περάσει από OFF σε ON, η LogicGate μπαίνει σε κατάσταση ON. Παραμένει σε κατάσταση ON ακόμα και όταν η σκανδάλη επιστρέψει σε κατάσταση OFF. Όταν όμως η σκανδάλη μπει και πάλι σε κατάσταση ON, η LogicGate μπαίνει σε κατάσταση OFF και παραμένει σ' αυτή την κατάσταση μέχρις ότου η σκανδάλη ενεργοποιηθεί και πάλι. Ο τύπος αυτός είναι χρήσιμος για την προσομοίωση λειτουργίας διακοπών.
6	CALLBACK. Συμπληρώνουμε μόνο το πεδίο Trigger1Name. Όταν ο χρήστης έρχεται σε επαφή με μοντέλα που συνδέονται με οντότητες Trigger, Door, MovingPlatform και Teleporter, στέλνεται ένα σήμα προς τις οντότητες αυτές. Το σήμα όμως, παρότι στιγμιαίο, μπορεί να το χρησιμοποιήσει και μία LogicGate για να ενεργοποιήσει με τη σειρά της άλλες οντότητες. Στο πεδίο Trigger1Name εισάγουμε το όνομα του μοντέλου.
7	HOLD. Χρησιμοποιούμε μία σκανδάλη (συμπληρώνουμε μόνο το πεδίο Trigger1Name). Όταν η σκανδάλη αλλάζει την κατάστασή της από OFF σε ON, τότε η LogicGate παραμένει σε κατάσταση ON για όσα δευτερόλεπτα έχουμε εισαγάγει στο πεδίο Delay.

Τιμή	Περιγραφή
8	ATTRIBUTE. Στο πεδίο της πρώτης σκανδάλης, αντί για σκανδάλη εισάγουμε το όνομα μίας ιδιότητας του χρήστη. Όταν η ιδιότητα αυτή ξεπεράσει την τιμή που ορίζουμε στο πεδίο Amount, τότε η LogicGate μπαίνει σε κατάσταση ON.
9	ALWAYS ON. Χρησιμοποιούμε μία σκανδάλη (συμπληρώνουμε μόνο το πεδίο Trigger1Name). Όταν η σκανδάλη αλλάξει την κατάστασή της από OFF σε ON, τότε η LogicGate παραμένει σε κατάσταση ON διαρκώς.
12	CHECK ENTITY CONTENTS. Η LogicGate ελέγχει το περιεχόμενο του στερεού μέσα στο οποίο βρίσκεται ο χρήστης ή μία άλλη οντότητα και μπαίνει σε κατάσταση ON όταν αυτό συμπίπτει με το όνομα που εισάγουμε στο πεδίο Trigger1Name. Τα ονόματα μπορεί να είναι: Water, ToxicGas, Lava, ZeroG, Frozen, Sludge, SlowMotion, FastMotion, Ladder, Impenetrable, Unclimbable (πρόκειται για τις ιδιότητες που μπορούμε να δώσουμε στα στερεά). Στο πεδίο Trigger2Name συμπληρώνουμε το όνομα της οντότητας που θέλουμε να ελέγχει η LogicGate (αν εισαγάγουμε την τιμή Player, εννοείται ο χρήστης). Για παράδειγμα, μπορούμε να κάνουμε τη LogicGate να ενεργοποιείται όταν ο χρήστης μπει μέσα σ' ένα στερεό που έχει την ιδιότητα του νερού.

Για να γίνει κατανοητή η χρήση της LogicGate, θα δώσουμε τρία παραδείγματα. Εννοείται βέβαια ότι σε κάθε μας παράδειγμα πρέπει να εισαγάγουμε τις οντότητες PlayerStart και PlayerSetup για να λειτουργήσουν.

17.3.1 Προσομοίωση διακόπτη On/Off

Στο πρώτο μας παράδειγμα θα κατασκευάσουμε κάτι το οποίο φαίνεται με την πρώτη ματιά απλό, ένα διακόπτη για τα φώτα ενός δωματίου. Αν επιχειρούσαμε να υλοποιήσουμε το διακόπτη μόνο με μία σκανδάλη, τα φώτα θα έμεναν αναμμένα για όσο διάστημα ορίζουμε στο πεδίο TimeOn, θα έσβηναν μόλις η σκανδάλη θα έμπαινε σε κατάσταση OFF, και μάλιστα η σκανδάλη δε θα μπορούσε να ενεργοποιηθεί και πάλι πριν περάσει το διάστημα που ορίζουμε στο πεδίο TimeOff. Είναι προφανές ότι οι διακόπτες δε λειτουργούν με αυτό τον τρόπο. Συνεπώς χρειαζόμαστε μία LogicGate και μάλιστα τύπου 5, η οποία έχει την ιδιότητα να παραμένει σε κατάσταση ON ακόμα και όταν η σκανδάλη απενεργοποιηθεί και μπαίνει σε κατάσταση OFF όταν η σκανδάλη ενεργοποιηθεί και πάλι.

Εισάγουμε λοιπόν σ' ένα απλό δωμάτιο ένα μικρό στερεό το οποίο το τοποθετούμε σ' έναν τοίχο και το μετατρέπουμε σε μοντέλο με το όνομα SwitchModel. Το μοντέλο αυτό θα παίξει το ρόλο του διακόπτη που θα χρησιμοποιεί ο χρήστης. Θα έχει τρία κλειδιά κίνησης, το πρώτο στην αρχική θέση στο χρόνο 0, το δεύτερο, στο χρόνο 0.5, θα είναι βυθισμένο στον τοίχο, και το τρίτο, στο χρόνο 1, θα επιστρέφει στην αρχική θέση.

Θα χρειαστούμε επίσης μία σκανδάλη, που θα της δώσουμε το όνομα `TriggerLights`, μία `LogicGate` με όνομα `LogicSwitch` και μία οντότητα `DynamicLight`. Δε θα χρησιμοποιήσουμε την οντότητα `Light`, γιατί αυτή δεν μπορεί να ελεγχθεί από σκανδάλη ή `LogicSwitch`. Οι ρυθμίσεις στις οντότητες θα είναι:

Για την `Trigger`:

Πεδίο	Τιμή
<code>bNoCollide</code>	<code>FALSE</code>
<code>Model</code>	<code>SwitchModel</code>
<code>PlayerOnly</code>	<code>TRUE</code>
<code>szEntityName</code>	<code>TriggerLights</code>
<code>TimeOff</code>	<code>1</code>
<code>TimeOn</code>	<code>1</code>
<code>UseKey</code>	<code>TRUE</code>

Για τη `LogicGate`:

Πεδίο	Τιμή
<code>szEntityName</code>	<code>LogicSwitch</code>
<code>Trigger1Name</code>	<code>TriggerLights</code>
<code>Type</code>	<code>5</code>

Για τη `DynamicLight`:

Πεδίο	Τιμή
<code>InterpolateValues</code>	<code>FALSE</code>
<code>MaxRadius</code>	<code>350</code>
<code>MinRadius</code>	<code>350</code>
<code>RadiusFunction</code>	<code>a</code>
<code>RadiusSpeed</code>	<code>0</code>
<code>TriggerName</code>	<code>LogicSwitch</code>

Αφού ρυθμίσουμε τη `DynamicLight`, μπορούμε να την αντιγράψουμε και να την επικολλήσουμε άλλες δύο-τρεις φορές για να γίνει πιο έντονος ο φωτισμός του δωματίου. Δεν έχει σημασία πόσο χρόνο θα παραμένει η σκανδάλη σε κατάσταση `ON` ή `OFF`. Αυτό γιατί, όπως αναφέραμε, τα φώτα ελέγχονται από τη `LogicGate`, η οποία θα παραμείνει σε κατάσταση `ON` ακόμα και όταν η σκανδάλη επιστρέψει σε κατάσταση `OFF`. Η `LogicGate` θα επιστρέψει σε κατάσταση `OFF` όταν θα χρησιμοποιηθεί και πάλι ο διακόπτης. Αν θέλουμε ο διακόπτης να μπορεί να χρησιμοποιηθεί μόνο μία φορά και τα φώτα να παραμένουν αναμμένα, κάνουμε τη `LogicGate` τύπου 9.

17.3.2 Ενεργοποίηση οντοτήτων με έλεγχο του είδους του στερεού όπου βρίσκεται ο χρήστης

Στο δεύτερο παράδειγμά μας θα χρησιμοποιήσουμε μία `LogicGate` τύπου 12, που θα ελέγχει το περιεχόμενο του στερεού στο οποίο βρίσκεται ο χρήστης. Για όσο διάστημα θα είναι μέσα στο στερεό που θα ορίσουμε, θα είναι αναμμένα τα φώτα του προηγού-

μενου παραδείγματος. Θα διατηρήσουμε λοιπόν τις DynamicLights, αλλά θα διαγράψουμε την Trigger και το μοντέλο του διακόπτη.

Εισάγουμε ένα στερεό αρκετά μεγάλο σε ύψος, ώστε το μοντέλο του χρήστη να είναι εξ ολοκλήρου μέσα σ' αυτό και του δίνουμε τις ιδιότητες Συμπαγές-Solid, Κενό-Empty και Αργής Κίνησης-SlowMotion. Έχοντάς το επιλεγμένο, πατάμε Page Down για να μπορούμε να επεξεργαστούμε τις έδρες του και, καθώς παραμένουν όλες επιλεγμένες, ορίζουμε να είναι διαφανείς και δίνουμε τιμή διαφάνειας 1 (απόλυτα διαφανές). Δεν ξεχνάμε να πιέσουμε το πλήκτρο εφαρμογής .

Οι παραπάνω ενέργειές μας θα έχουν ως αποτέλεσμα το στερεό να είναι αόρατο από το χρήστη, αλλά όταν εισέρχεται σ' αυτό να αρχίσει να κινείται αργά. Αυτό που απομένει είναι να ρυθμίσουμε τη LogicGate ώστε να μπορεί να ελέγχει αν ο χρήστης είναι μέσα στο στερεό, για να ανάψει τα φώτα. Κάνουμε λοιπόν σ' αυτή τις εξής ρυθμίσεις:

Πεδίο	Τιμή
szEntityName	LogicSwitch
Trigger1Name	SlowMotion
Trigger2Name	Player
Type	12

17.3.3 Σύνθετοι διακόπτες

Στο τελευταίο μας παράδειγμα θα υλοποιήσουμε μία πόρτα η οποία θα ανοίγει μόνο όταν ο χρήστης πιέσει δύο διακόπτες μέσα σε 10 δευτερόλεπτα. Αρχικά θα κατασκευάσουμε δύο δωμάτια, που θα τα συνδέσουμε μεταξύ τους μ' ένα στερεό με την ιδιότητα Αποκοπή-Cut. Θα εισάγουμε ένα ακόμα στερεό, που θα παίζει το ρόλο της πόρτας. Το μετατρέπουμε σε μοντέλο με το όνομα DoorModel και του δίνουμε τρία κλειδιά κίνησης. Το πρώτο στην αρχική θέση τη χρονική στιγμή 0, το δεύτερο στην ανοιχτή θέση της πόρτας (μετακινούμε το στερεό μέσα στους τοίχους) τη χρονική στιγμή 4 και το τρίτο κλειδί στην αρχική θέση της πόρτας τη χρονική στιγμή 8.

Με άλλα λόγια, η όλη κίνηση της πόρτας από την κλειστή στην ανοιχτή θέση και πάλι στην κλειστή θα διαρκεί 8 δευτερόλεπτα. Θα χρειαστούμε ακόμα δύο μοντέλα διακοπών (με ονόματα SwitchModel1 και SwitchModel2) με κλειδιά όπως και στο πρώτο μας παράδειγμα, δύο σκανδάλες (Trigger1 και Trigger2), μία MovingPlatform και φυσικά μία LogicGate (με το όνομα LG1). Παραμετροποιούμε τις οντότητες ως εξής:

Για τις δύο σκανδάλες:

Πεδίο	Τιμή
bNoCollide	FALSE
Model	SwitchModel1/ SwitchModel2
PlayerOnly	TRUE
szEntityName	Trigger1/ Trigger2
TimeOff	1
TimeOn	10
UseKey	TRUE

Για τη LogicGate:

Πεδίο	Τιμή
szEntityName	LG1
Trigger1Name	Trigger1
Trigger2Name	Trigger2
Type	0

Για τη MovingPlatform:

Πεδίο	Τιμή	Πεδίο	Τιμή
bLooping	TRUE	PlayerOnly	FALSE
bNoCollide	TRUE	TimeEachTrigger	1
bReverse	FALSE	TriggerName	LG1
bRunTimed	TRUE	UseKey	FALSE
Model	DoorModel		

Όσον αφορά τις σκανδάλες, θέλουμε να ενεργοποιούνται με την επαφή του χρήστη στα μοντέλα των διακοπών ή να μπορεί ο χρήστης να πιέσει το πλήκτρο “χρήσης” και να παραμένουν σε κατάσταση ON για 10 δευτερόλεπτα. Για την κατάσταση OFF μπορούμε να χρησιμοποιήσουμε όποια τιμή θέλουμε. Δεν έχει σημασία με ποια σειρά θα ενεργοποιηθούν, αρκεί η ενεργοποίηση της δεύτερης να γίνεται πάντα σε χρόνο μικρότερο των 10 δευτερολέπτων.

Η LogicGate είναι τύπου 0, που σημαίνει ότι θα μπει σε κατάσταση ON μόνο όταν και οι δύο σκανδάλες είναι σε κατάσταση ON. Αν ο χρήστης καθυστερήσει και περάσει το διάστημα αυτό, τότε μόνο η τελευταία ενεργοποιημένη θα είναι ON. Αυτό σημαίνει ότι θα πρέπει να πιέσει ξανά τον πρώτο διακόπτη, για να τεθεί και πάλι η πρώτη σκανδάλη σε κατάσταση ON. Χρησιμοποιώντας μικρό χρόνο στα πεδία TimeOff στις σκανδάλες, εξασφαλίζουμε ότι οι διακόπτες με τους οποίους συνδέονται θα μπορούν γρήγορα να επαναχρησιμοποιηθούν.

Η MovingPlatform δεν ενεργοποιείται με κάποια από τις δύο σκανδάλες, αλλά από τη LogicGate, γι’ αυτό άλλωστε χρησιμοποιούμε το όνομά της στο πεδίο TriggerName. Αφού έχουμε έμμεση ενεργοποίηση και όχι με την επαφή του χρήστη στο μοντέλο της πόρτας, οι τιμές στα πεδία bNoCollide, PlayerOnly και UseKey είναι TRUE, FALSE και FALSE αντίστοιχα.

Επειδή δε γνωρίζουμε πόση ώρα θα χρειαστεί ο χρήστης για να ενεργοποιήσει και τις δύο σκανδάλες και επειδή θέλουμε να μπορεί να ενεργοποιήσει πολλές φορές την κίνηση της πόρτας, στο πεδίο `bLooping` χρησιμοποιήσαμε την τιμή `TRUE`. Επειδή τα κλειδιά κίνησης της πόρτας περιείχαν τις θέσεις κλειστή-ανοιχτή-κλειστή, στο πεδίο `bReverse` χρησιμοποιήσαμε την τιμή `FALSE`.

Τέλος, επειδή η `MovingPlatform` θα κινεί την πόρτα για όσο διάστημα παραμένει η `LogicGate` σε κατάσταση `ON` και συνεπώς όχι ανεξάρτητα από μία και μόνο αρχική ενεργοποίησή της, εισάγουμε την τιμή `TRUE` στο πεδίο `bRunTimed` και την τιμή `1` στο πεδίο `TimeEachTrigger`. Αυτό σημαίνει ότι η κίνηση θα εκτελεστεί για ένα δευτερόλεπτο, αν η `LogicGate` εξακολουθεί να είναι ενεργή θα εκτελεστεί για ακόμα ένα, κ.ο.κ. Η συνολική διάρκεια κίνησης της πόρτας θα είναι 10 δευτερόλεπτα μείον το χρόνο που χρειάστηκε ο χρήστης για την ενεργοποίηση της δεύτερης σκανδάλης.

Κεφάλαιο 18^ο

Εφέ

Το Reality Factory περιλαμβάνει μία σειρά από εφέ τα οποία μπορούμε να χρησιμοποιήσουμε στην εφαρμογή μας. Στη συνέχεια του κεφαλαίου παρουσιάζουμε τα σημαντικότερα, δίνοντας για το καθένα απ' αυτά και από ένα παράδειγμα. Το μόνο που θα χρειαστούμε αρχικά είναι ένα απλό δωμάτιο και τις οντότητες `PlayerSetup` και `PlayerStart`. Να σημειώσουμε ότι η υπερβολική χρήση εφέ μπορεί να αποβεί σε βάρος του αριθμού των καρέ/δευτερόλεπτο που αποδίδει η εφαρμογή μας και γι' αυτό συνιστάται η προσεκτική χρήση τους.

18.1 Echaos

Με την οντότητα `Echaos` μπορούμε να κάνουμε την υφή με την οποία είναι συνδεδεμένη να πάλλεται σε βαθμό και με ταχύτητα που εμείς καθορίζουμε. Αυτό το εφέ είναι ιδανικό για να προσδώσει κίνηση σε γεωμετρικά στερεά που αναπαριστούν υγρά (νερό, λάβα, κτλ.). Οι ιδιότητες της `Echaos` είναι:

Πεδίο	Περιγραφή
<code>AttachBmp</code>	Το όνομα της υφής με την οποία συνδέεται η οντότητα. Αν πρόκειται για εικόνα που βρίσκεται μέσα στο πακέτο υφών που χρησιμοποιούμε, δε χρειάζεται να δώσουμε την κατάληξη <code>.bmp</code> . Αν το πεδίο <code>EntityName</code> είναι συμπληρωμένο, τότε η υφή αφορά τον actor της οντότητας εκείνης (και πρέπει να δώσουμε την κατάληξη <code>.bmp</code> στο όνομα του αρχείου εικόνας). Τα ονόματα των υφών του πακέτου φαίνονται στην καρτέλα των υφών στο πάνελ εντολών.
<code>EntityName</code>	Το όνομα της οντότητας στον actor της οποίας υπάρχει η υφή που δηλώσαμε στο πεδίο <code>AttachBmp</code> και στην οποία θα εφαρμοστεί το εφέ <code>Echaos</code> . Για να εφαρμόσουμε το εφέ σε υφή του actor του χρήστη, δίνουμε την τιμή <code>Player</code> .
<code>MaxXSway</code>	Η τιμή που δίνουμε στο πεδίο προσδιορίζει το πόσο θα ταλαντώνονται τα εικονοστοιχεία της υφής κατά τον άξονα των x .
<code>MaxYSway</code>	Η τιμή που δίνουμε στο πεδίο προσδιορίζει το πόσο θα ταλαντώνονται τα εικονοστοιχεία της υφής κατά τον άξονα των y .
<code>Origin</code>	Η θέση της οντότητας. Η τιμή συμπληρώνεται αυτόματα.
<code>Xstep</code>	Η ταχύτητα της ταλάντωσης των εικονοστοιχείων στον άξονα των x .
<code>Ystep</code>	Η ταχύτητα της ταλάντωσης των εικονοστοιχείων στον άξονα των y .

Ας υποθέσουμε ότι κάπου στο επίπεδό μας υπάρχει ένα στερεό που του δώσαμε τις ιδιότητες του νερού και χρησιμοποιούμε σ' αυτό την υφή `water` από το πακέτο υφών (κεφάλαιο 12.2.2). Με τις ρυθμίσεις της `Echaos` του πρώτου από τα παρακάτω παραδείγματα θα προκύψει ένας ελαφρύς κυματισμός, ενώ με τις ρυθμίσεις του δεύτερου θα έχουμε την αίσθηση ότι το νερό είναι έντονα ταραγμένο.

Πεδίο	Παράδειγμα 1	Παράδειγμα 2
AttachBmp	water	water
EntityName	-	-
MaxXSway	20	40
MaxYSway	20	40
Xstep	2	4
Ystep	2	4

Επειδή στην EChaos μπορούμε να συνδέσουμε υφή από το πακέτο που χρησιμοποιούμε, υπάρχει η πιθανότητα, αν η υφή χρησιμοποιείται και αλλού στην εφαρμογή μας, να εφαρμοστεί το εφέ και σε άλλο σημείο που μπορεί να μην το θέλουμε. Για να αποφύγουμε προβλήματα, μπορούμε στο πακέτο υφών να εντάξουμε δύο ίδιες υφές με διαφορετικό όμως όνομα. Τη μία θα τη χρησιμοποιούμε όπου χρειαζόμαστε το εφέ και την άλλη όπου δεν το χρειαζόμαστε.

18.2 ElectricBolt και ElectricBoltTerminus

Για να παραγάγουμε το εφέ της ηλεκτρικής εκκένωσης χρειάζεται να χρησιμοποιήσουμε στην εφαρμογή μας δύο οντότητες, την ElectricBolt και την ElectricBoltTerminus. Η πρώτη παράγει το εφέ, ενώ η δεύτερη είναι το σημείο που καταλήγει η εκκένωση. Οι ιδιότητες της ElectricBolt είναι:

Πεδίο	Περιγραφή
AlphaName	Το όνομα του αρχείου εικόνας για το διαφανές μέρος του εφέ.
BmpName	Το όνομα του αρχείου εικόνας για το εφέ.
BoneName	Το όνομα του κοκάλου ενός actor στο οποίο συνδέεται η οντότητα ElectricBolt.
Color	Το χρώμα του εφέ. Οι τιμές των δύο μη βασικών χρωμάτων πρέπει να είναι ίδιες. Για παράδειγμα, αν θέλουμε η εκκένωση να έχει μπλε χρώμα, οι τιμές για το κόκκινο και το πράσινο χρώμα πρέπει να είναι ίδιες. Με διπλό κλικ ανοίγει πάνελ επιλογής χρωμάτων.
DamageAltAmt	Η τιμή ορίζει το μέγεθος της “βλάβης” που μπορεί να προκαλέσει το εφέ στην εναλλακτική παράμετρο “υγείας” του χρήστη.
DamageAltAttribute	Στο πεδίο εισάγουμε το όνομα της εναλλακτικής παραμέτρου “υγείας” του χρήστη (health, stamina, oxygen) που θα βλάπτεται από το εφέ.

Πεδίο	Περιγραφή
DamageAmt	Η τιμή ορίζει το μέγεθος της “βλάβης” που μπορεί να προκαλέσει το εφέ στην κύρια παράμετρο “υγείας” του χρήστη.
DamageAttribute	Στο πεδίο εισάγουμε το όνομα της κύριας παραμέτρου “υγείας” του χρήστη (health, stamina, oxygen) που θα βλάπτεται από το εφέ.
DamageTime	Η τιμή, σε δευτερόλεπτα, προσδιορίζει το χρόνο μέχρις ότου επαναληφθεί η βλάβη στο χρήστη.
DoDamage	Αν δώσουμε την τιμή TRUE, τότε το εφέ θα προκαλεί βλάβη στο χρήστη.
DominantColor	Το κυρίαρχο χρώμα της ηλεκτρικής εκκένωσης (0=κόκκινο, 1=πράσινο, 2=μπλε).
EntityName	Το όνομα της οντότητας στην οποία συνδέεται η οντότητα ElectricBolt.
fRadius	Η ακτίνα σε texels στην οποία ακούγεται ο ήχος που ορίζουμε στο πεδίο SoundFile.
Intermittent	Αν δώσουμε την τιμή TRUE, τότε το εφέ της εκκένωσης θα επαναλαμβάνεται κατά διαστήματα. Αν δώσουμε την τιμή FALSE, τότε το εφέ θα είναι συνεχές.
MaxFrequency	Αν οι εκκενώσεις είναι επαναλαμβανόμενες, τότε στο πεδίο αυτό προσδιορίζουμε το μέγιστο χρόνο σε χιλιοστά του δευτερολέπτου μέχρις ότου επαναληφθεί μία εκκένωση.
MinFrequency	Αν οι εκκενώσεις είναι επαναλαμβανόμενες, τότε στο πεδίο αυτό προσδιορίζουμε τον ελάχιστο χρόνο σε χιλιοστά του δευτερολέπτου μέχρις ότου επαναληφθεί μία εκκένωση.
Model	Το όνομα του μοντέλου στο οποίο συνδέεται η οντότητα ElectricBolt.
NumPoints	Πόσα σημεία θα ορίζουν την εκκένωση. Η τιμή πρέπει να είναι δύναμη του 2 (32, 64, 128, κτλ.).
Origin	Η θέση της οντότητας. Η τιμή συμπληρώνεται αυτόματα.
SoundFile	Ποιο αρχείο ήχου θα εκτελείται μαζί με το εφέ.
szEntityName	Το όνομα της συγκεκριμένης οντότητας ElectricBolt. Αυτό το όνομα θα χρησιμοποιηθεί σε πεδία ιδιοτήτων άλλων οντοτήτων που θέλουμε να συνδέονται με αυτή.
Terminus	Το όνομα της οντότητας ElectricBoltTerminus στην οποία θα καταλήγει η ηλεκτρική εκκένωση.
TriggerName	Το όνομα της σκανδάλης που ενεργοποιεί τη συγκεκριμένη οντότητα ElectricBolt.

Πεδίο	Περιγραφή
Width	Το πλάτος σε texels της ηλεκτρικής εκκένωσης.
Wildness	Η τιμή (0-1) προσδιορίζει το πόσο έντονη θα είναι η εκκένωση. Αν δώσουμε την τιμή 0, τότε η εκκένωση είναι μία ευθεία γραμμή και μπορούμε να τη χρησιμοποιήσουμε για να αναπαραστήσουμε καλώδια χωρίς επίπτωση στον αριθμό των καρέ/δευτερόλεπτο.

Αντίστοιχα, οι ιδιότητες της ElectricBoltTerminus είναι:

Πεδίο	Περιγραφή
BoneName	Το όνομα του κοκάλου ενός actor στο οποίο συνδέεται η οντότητα ElectricBoltTerminus.
EntityName	Το όνομα της οντότητας στην οποία συνδέεται η οντότητα ElectricBoltTerminus.
Model	Το όνομα του μοντέλου στο οποίο συνδέεται η οντότητα ElectricBoltTerminus.
Origin	Η θέση της οντότητας. Η τιμή συμπληρώνεται αυτόματα.
szEntityName	Το όνομα της συγκεκριμένης οντότητας ElectricBoltTerminus. Αυτό το όνομα θα χρησιμοποιηθεί σε πεδία ιδιοτήτων άλλων οντοτήτων που θέλουμε να συνδέονται με αυτή.

Σ' ένα απλό δωμάτιο εισάγουμε το δίδυμο ElectricBolt και ElectricBoltTerminus. Εισάγουμε επίσης δύο στερεά για να φαίνεται ότι απ' αυτά ξεκινά και καταλήγει η εκκένωση. Επίσης, θα εισαγάγουμε μία σκανδάλη που με την ενεργοποίησή της θα εκκινεί την ElectricBolt. Για το λόγο αυτό θα χρειαστούμε και ένα τρίτο στερεό, που θα το μετατρέψουμε σε μοντέλο (κεφάλαιο 16.1) το οποίο θα συνδέεται με τη σκανδάλη και θα παίζει το ρόλο του διακόπτη της (θα ενεργοποιείται με την επαφή του χρήστη στο στερεό).

Για τις ανάγκες του παραδείγματός μας, το στερεό που θα συνδεθεί με τη σκανδάλη θα το ονομάσουμε trmodel και τη σκανδάλη trelectricbolt. Τοποθετούμε τις ElectricBolt και ElectricBoltTerminus δίπλα στα δύο στερεά (εικόνα 18.1). Οι ρυθμίσεις που κάνουμε στις οντότητες είναι:

Για την ElectricBolt:

Πεδίο	Τιμή
Color	101 101 252
DamageAmt	1
DoDamage	TRUE
DominantColor	2
DamageAttribute	Health
DamageTime	1
fRadius	200
Intermittent	TRUE

Πεδίο	Τιμή
MaxFrequency	1
MinFrequency	4
NumPoints	64
SoundFile	Electricity.wav*
Terminus	ElectricBoltTerminus1
TriggerName	trelectricbolt
Width	5
Wildness	.8

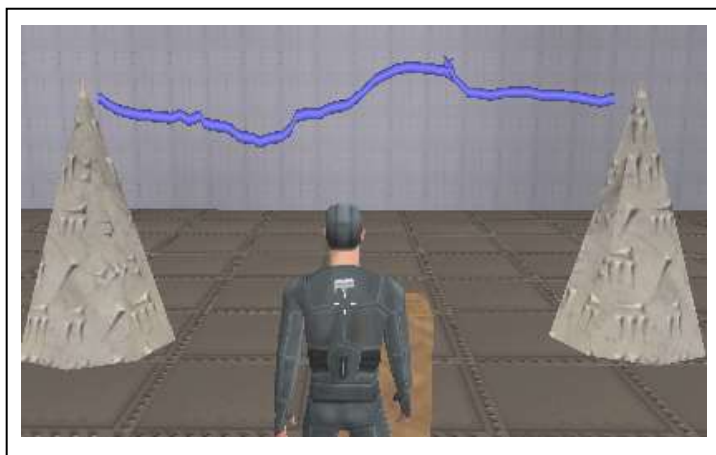
* Το αρχείο ήχου δεν περιλαμβάνεται στην εγκατάσταση του Reality Factory.

Για την ElectricBoltTerminus:

Πεδίο	Τιμή
Δε χρειάζεται καμία αλλαγή	

Για την Trigger:

Πεδίο	Τιμή
bNoCollide	FALSE
Model	trmodel
PlayerOnly	TRUE
szEntityName	trelectricbolt
TimeOff	1
TimeOn	20



Εικόνα 18.1 ElectricBolt και ElectricBoltTerminus

Αν δε συνδέσουμε μία σκανδάλη με την ElectricBolt, τότε το εφέ εκτελείται με την έναρξη της εφαρμογής.

18.3 Explosion

Το εφέ της έκρηξης παράγεται από την οντότητα Explosion. Η ιδιαιτερότητά της έγκειται στο γεγονός ότι μπορούμε να ορίσουμε εμείς το είδος της έκρηξης κάνοντας τις ανάλογες δηλώσεις στο αρχείο Explosion.ini, που βρίσκεται στο φάκελο C:\RealityFactory\Install. Στο αρχείο έχουν ήδη οριστεί αρκετές εκρήξεις. Το όνομα του είδους της έκρηξης (που βρίσκεται μέσα σε άγκιστρα) το δηλώνουμε σε ανάλογο πεδίο των ιδιοτήτων της Explosion. Οι ιδιότητές της είναι:

Πεδίο	Περιγραφή
BoneName	Το όνομα του κοκάλου ενός actor στο οποίο συνδέεται η οντότητα Explosion.
DamageAltAmt	Η τιμή ορίζει το μέγεθος της “βλάβης” που μπορεί να προκαλέσει το εφέ στην εναλλακτική παράμετρο “υγείας” του χρήστη.
DamageAltAttribute	Στο πεδίο εισάγουμε το όνομα της εναλλακτικής παραμέτρου “υγείας” του χρήστη (health, stamina, oxygen) που θα βλάπτεται από το εφέ.
DamageAmt	Η τιμή ορίζει το μέγεθος της “βλάβης” που μπορεί να προκαλέσει το εφέ στην κύρια παράμετρο “υγείας” του χρήστη.
DamageAttribute	Στο πεδίο εισάγουμε το όνομα της κύριας παραμέτρου “υγείας” του χρήστη (health, stamina, oxygen) που θα βλάπτεται από το εφέ.
EntityName	Το όνομα της οντότητας στην οποία συνδέεται η Explosion.
ExplosionName	Το όνομα του είδους της έκρηξης έτσι όπως ορίζεται στο αρχείο Explosion.ini.
Model	Το όνομα του μοντέλου στο οποίο συνδέεται η οντότητα Explosion.
Origin	Η θέση της οντότητας. Η τιμή συμπληρώνεται αυτόματα.
Radius	Η ακτίνα της έκρηξης σε texels. Μέσα στην ακτίνα αυτή προκαλεί τη βλάβη που ορίσαμε στα πιο πάνω πεδία.
ShakeAmt	Πόσο θα “κουνηθεί” η οθόνη του χρήστη από την έκρηξη.
ShakeDecay	Η τιμή καθορίζει το πόσο θα καθυστερήσει να “σβήσει” το “κούνημα” της οθόνης. Μεγαλύτερες τιμές μειώνουν το χρόνο.
TriggerName	Το όνομα της σκανδάλης που ενεργοποιεί τη συγκεκριμένη οντότητα Explosion.

Τοποθετούμε την Explosion στο σημείο που μας ενδιαφέρει. Εισάγουμε ένα στερεό που θα μετατρέψουμε σε μοντέλο με το όνομα trmodel, που θα ενεργοποιεί τη σκαν-

δάλη και αυτή με τη σειρά της θα ενεργοποιεί το εφέ (εικόνα 18.2). Οι ρυθμίσεις που κάνουμε στις οντότητες είναι:

Για την Explosion:

Πεδίο	Τιμή
DamageAmt	50
DamageAttribute	health
ExplosionName	GeneralExplosion
Radius	400
ShakeAmt	100
ShakeDecay	0
TriggerName	tr

Για την Trigger:

Πεδίο	Τιμή
bNoCollide	FALSE
Model	trmodel
PlayerOnly	TRUE
szEntityName	tr
TimeOff	1
TimeOn	1



Εικόνα 18.2 Η οντότητα Explosion

Στην Explosion δε χρειάζεται να δηλώσουμε κάποιον ήχο, γιατί το εφέ έχει ενσωματωμένους τους δικούς του. Για να λειτουργήσει όμως το εφέ του “κουνήματος της οθόνης”, χρειάζεται να έχουμε εισαγάγει και παραμετροποιήσει την οντότητα EnvironmentSetup.

Αν θέλουμε μόνο να “κουνιέται” η οθόνη χωρίς έκρηξη, μπορούμε να χρησιμοποιήσουμε την οντότητα ScreenShake. Και πάλι όμως χρειάζεται η οντότητα EnvironmentSetup.

18.4 FlipBook

Η οντότητα FlipBook χρησιμοποιείται για να προβάλουμε σε μία επιφάνεια μία σειρά από εναλλασσόμενες εικόνες. Με αυτό τον τρόπο μπορούμε να αλλάξουμε την υφή μίας οντότητας, ενός μοντέλου ή τέλος μπορούμε να αντικαταστήσουμε μία υφή που χρησιμοποιούμε σ' ένα γεωμετρικό στερεό. Οι ιδιότητές της είναι:

Πεδίο	Περιγραφή
Alpha	Ο αρχικός βαθμός διαφάνειας των εικόνων που θα χρησιμοποιήσουμε.
AlphaNameBase	Το όνομα βάσης για τις εικόνες διαφάνειας που θα χρησιμοποιήσουμε. Κάθε εικόνα πρέπει να έχει ένα όνομα βάσης, ακολουθούμενο από έναν αριθμό που καθορίζει τη σειρά εμφάνισής της. Για παράδειγμα, flip0.bmp, flip1.bmp, flip2.bmp (το όνομα βάσης είναι flip).
AlphaRate	Η τιμή καθορίζει το πόσο θα μειώνεται η διαφάνεια το δευτερόλεπτο.
AtLevelStart	Αν δώσουμε την τιμή TRUE, τότε το εφέ θα λειτουργήσει μόνο μία φορά με την έναρξη της εφαρμογής.
BitmapCount	Ο αριθμός των εικόνων που θα χρησιμοποιήσουμε.
BmpNameBase	Το όνομα βάσης για τις εικόνες που θα χρησιμοποιήσουμε. Ισχύουν τα ίδια με το πεδίο AlphaNameBase.
BoneName	Το όνομα του κοκάλου ενός actor στο οποίο συνδέεται η οντότητα FlipBook.
Color	Το χρώμα του εφέ.
EntityName	Το όνομα της οντότητας στην οποία συνδέεται η FlipBook.
Model	Το όνομα του μοντέλου στο οποίο συνδέεται η οντότητα FlipBook.
Origin	Η θέση της οντότητας. Η τιμή συμπληρώνεται αυτόματα.
Rotation	Η αρχική περιστροφή των εικόνων σε μοίρες.
RotationRate	Ο βαθμός περιστροφής των εικόνων το δευτερόλεπτο.
Scale	Η αρχική κλίμακα των εικόνων που θα χρησιμοποιήσουμε. Η τιμή 1 αντιπροσωπεύει το 100%.
ScaleRate	Η τιμή αντιπροσωπεύει το πόσο θα μειώνεται η κλίμακα των εικόνων το δευτερόλεπτο.
Speed	Ο αριθμός των εικόνων που θα προβάλλονται το δευτερόλεπτο.

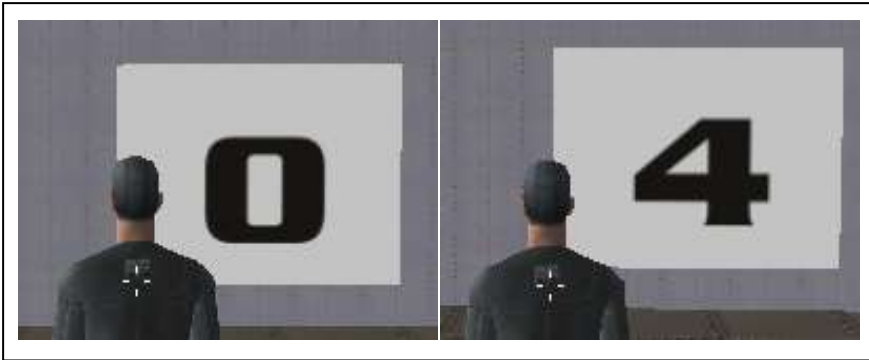
Πεδίο	Περιγραφή
Style	Η τιμή προσδιορίζει το είδος του εφέ. 0=κανένα animation, προβολή μόνο της πρώτης εικόνας 1=στο τέλος του animation, προβολή από την αρχή 2= στο τέλος του animation, αντιστροφή της σειράς των εικόνων 3=επιλογή και προβολή τυχαίας εικόνας 4= στο τέλος του animation, τέλος και του εφέ
szEntityName	Το όνομα της συγκεκριμένης οντότητας FlipBook. Αυτό το όνομα θα χρησιμοποιηθεί σε πεδία ιδιοτήτων άλλων οντοτήτων που θέλουμε να συνδέονται με αυτή.
TextureName	Το όνομα της υφής με την οποία συνδέεται η οντότητα. Αν πρόκειται για εικόνα που βρίσκεται μέσα στο πακέτο υφών που χρησιμοποιούμε, δε χρειάζεται να δώσουμε την κατάληξη .bmp. Τα ονόματα των υφών του πακέτου φαίνονται στην καρτέλα των υφών στο πάνελ εντολών.
TriggerName	Το όνομα της σκανδάλης που ενεργοποιεί τη συγκεκριμένη οντότητα FlipBook.

Στο παράδειγμά μας θα εφαρμόσουμε τη FlipBook στην υφή ενός γεωμετρικού στερεού και το εφέ θα ξεκινά με την έναρξη του επιπέδου, χωρίς τη χρήση δηλαδή κάποιας σκανδάλης (εικόνα 18.3). Για το λόγο αυτό, εισάγουμε στο δωμάτιο ένα στερεό κύβο, διαμορφώνουμε τις διαστάσεις του ώστε να γίνει ένα λεπτό ορθογώνιο παραλληλεπίπεδο και το τοποθετούμε σαν πίνακα σ' έναν από τους τοίχους. Η υφή που θα του δώσουμε είναι η brickgre2 του βασικού πακέτου υφών. Οι ρυθμίσεις που κάνουμε στη FlipBook είναι:

Πεδίο	Τιμή
AtLevelStart	FALSE
BitmapCount	5
BmpNameBase	Flip*

Πεδίο	Τιμή
Speed	1
Style	1
TextureName	brickgre2

* Οι εικόνες flip0-flip4.bmp περιλαμβάνονται στην εγκατάσταση του Reality Factory.



Εικόνα 18.3 Η οντότητα FlipBook

Για να εμφανιστεί σωστά το εφέ, πρέπει πρώτα να τοποθετήσουμε σωστά την υφή του στερεού, προσαρμόζοντας τις παραμέτρους Scale και Offset (επιλέγοντάς το και πατώντας Page Down).

18.5 FloatingParticles

Στο Reality Factory μπορούμε να δημιουργήσουμε το εφέ των αιωρούμενων σωματιδίων με τη χρήση της οντότητας FloatingParticles. Το εφέ είναι κατάλληλο για την αναπαράσταση φυσαλίδων μέσα σε νερό, για χιόνι (η βροχή δημιουργείται από άλλη οντότητα), καπνού, σκόνης, κτλ. Τα σωματίδια μπορούν να κινούνται προς τα πάνω ή προς τα κάτω και με ταχύτητα που ορίζουμε στα ανάλογα πεδία. Αναλυτικά, οι ιδιότητες της οντότητας είναι:

Πεδίο	Περιγραφή
AlphaName	Το όνομα του αρχείου εικόνας για το διαφανές μέρος του εφέ.
BmpName	Το όνομα του αρχείου εικόνας για το εφέ.
BoneName	Το όνομα του κοκάλου ενός actor στο οποίο συνδέεται η οντότητα FloatingParticles.
Color	Το χρώμα του εφέ.
EntityName	Το όνομα της οντότητας στην οποία συνδέεται η FloatingParticles.
Height	Η μέγιστη απόσταση (σε texels) που θα διανύει το κάθε σωματίδιο από το σημείο που εμφανίζεται. Πάντα δίνουμε θετικές τιμές.
MaxSpeed	Η μέγιστη ταχύτητα του κάθε σωματιδίου. Δίνοντας αρνητικές τιμές, τα σωματίδια κινούνται προς τα κάτω.
MinSpeed	Η ελάχιστη ταχύτητα του κάθε σωματιδίου. Δίνοντας αρνητικές τιμές, τα σωματίδια κινούνται προς τα κάτω.

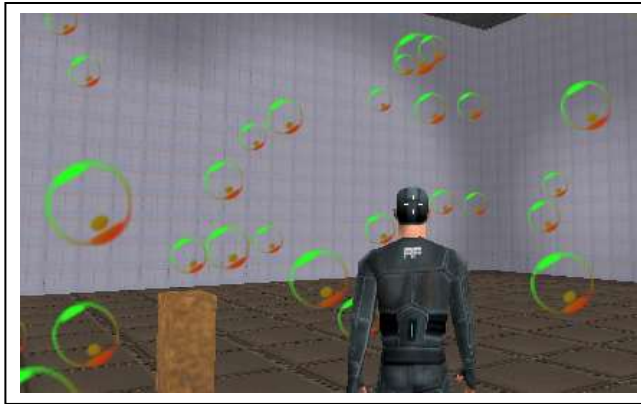
Πεδίο	Περιγραφή
Model	Το όνομα του μοντέλου στο οποίο συνδέεται η οντότητα FloatingParticles.
Origin	Η θέση της οντότητας. Η τιμή συμπληρώνεται αυτόματα.
ParticleCount	Ο αριθμός των σωματιδίων που θα δημιουργεί η FloatingParticles.
Radius	Η ακτίνα (σε texels) της βάσης του κυλίνδρου που θα απλώνεται το εφέ της FloatingParticles.
Scale	Η κλίμακα των σωματιδίων.
szEntityName	Το όνομα της συγκεκριμένης οντότητας FloatingParticles. Αυτό το όνομα θα χρησιμοποιηθεί σε πεδία ιδιοτήτων άλλων οντοτήτων που θέλουμε να συνδέονται με αυτή.
TriggerName	Το όνομα της σκανδάλης που ενεργοποιεί τη συγκεκριμένη οντότητα FloatingParticles.
XSlant	Το ποσοστό παρέκκλισης των σωματιδίων από τον άξονα των x. Η τιμή δίνεται με δεκαδικό αριθμό.
ZSlant	Το ποσοστό παρέκκλισης των σωματιδίων από τον άξονα των z. Η τιμή δίνεται με δεκαδικό αριθμό.

Στο παράδειγμά μας θα δημιουργήσουμε φουσαλίδες που θα κινούνται προς τα πάνω (εικόνα 18.4). Για το λόγο αυτό, θα τοποθετήσουμε τη FloatingParticles χαμηλά στο πάτωμα. Παρότι μπορούμε, δε θα χρησιμοποιήσουμε κάποια σκανδάλη για να ενεργοποιείται το εφέ. Συνεπώς, το εφέ θα εκτελείται με την έναρξη της εφαρμογής. Οι ρυθμίσεις που κάνουμε στην FloatingParticles είναι:

Πεδίο	Τιμή
AlphaName	a_bubble.bmp*
BmpName	g_bubble.bmp*
Height	200
MaxSpeed	150
MinSpeed	60

Πεδίο	Τιμή
Radius	300
Scale	.5
ParticleCount	60
XSlant	.65
ZSlant	.65

* Τα αρχεία των εικόνων περιλαμβάνονται στην εγκατάσταση του Reality Factory.



Εικόνα 18.4 Η οντότητα FloatingParticles

18.6 FogLight

Ένα πάρα πολύ απλό εφέ, το οποίο όμως μπορεί να συμβάλει σημαντικά στη δημιουργία της κατάλληλης ατμόσφαιρας στην εφαρμογή μας, είναι η ομίχλη. Η οντότητα που χειρίζεται τη δημιουργία της είναι η FogLight. Η FogLight δεν μπορεί να συνδεθεί με κάποια σκανδάλη, με αποτέλεσμα το εφέ να ξεκινά με την έναρξη της εφαρμογής. Οι ιδιότητές της είναι:

Πεδίο	Περιγραφή
Brightness	Η ένταση/φωτεινότητα του εφέ.
Color	Το χρώμα του εφέ.
Origin	Η θέση της οντότητας. Η τιμή συμπληρώνεται αυτόματα.
Radius	Η ακτίνα της παραγόμενης ομίχλης.

Στο παράδειγμά μας θα παραμετροποιήσουμε τη FogLight ώστε να έχει μικρή έκταση και θα αλλάξουμε το αρχικό της χρώμα ως εξής (εικόνα 18.5):

Πεδίο	Τιμή
Brightness	1500
Color	230 0 115
Radius	200



Εικόνα 18.5 Η οντότητα FogLight

18.7 MorphingField

Η οντότητα MorphingField παράγει και αυτή ομίχλη όπως η FogLight, με τη διαφορά ότι έχει δυναμικά χαρακτηριστικά. Μπορούμε δηλαδή να την κάνουμε να αλλάζει χρώμα, φωτεινότητα, ακτίνα, ή όλα μαζί. Ενδιαφέροντα αποτελέσματα θα έχουμε αν χρησιμοποιήσουμε ακραίες εναλλαγές χρωμάτων. Η χρήση της όμως πρέπει να γίνεται με προσοχή, γιατί μπορεί να έχει επίπτωση στον αριθμό των καρέ/δευτερόλεπτο. Οι ιδιότητές της είναι:

Πεδίο	Περιγραφή
BoneName	Το όνομα του κοκάλου ενός actor στο οποίο συνδέεται η οντότητα MorphingField.
clrEnd	Το τελικό χρώμα που θα αποκτή η MorphingField.
clrStart	Το αρχικό χρώμα που θα έχει η MorphingField.
EntityName	Το όνομα της οντότητας στην οποία συνδέεται η MorphingField.
fogRadiusEnd	Η τελική ακτίνα σε texels της MorphingField.
fogRadiusStart	Η αρχική ακτίνα σε texels της MorphingField.
fogSpeed	Η ταχύτητα με την οποία θα γίνονται οι διάφορες αλλαγές σε χιλιοστά του δευτερολέπτου.
fogVariance	Πόσο θα μεταβάλλεται η ένταση/φωτεινότητα του εφέ.
Model	Το όνομα του μοντέλου στο οποίο συνδέεται η οντότητα MorphingField.
Origin	Η θέση της οντότητας. Η τιμή συμπληρώνεται αυτόματα.
szEntityName	Το όνομα της συγκεκριμένης οντότητας MorphingField. Αυτό το όνομα θα χρησιμοποιηθεί σε πεδία ιδιοτήτων άλλων οντοτήτων που θέλουμε να συνδέονται με αυτή.

Πεδίο	Περιγραφή
szSoundFile	Ποιο αρχείο ήχου θα εκτελείται μαζί με το εφέ.
TriggerName	Το όνομα της σκανδάλης που ενεργοποιεί τη συγκεκριμένη οντότητα MorphingField.

Στο παράδειγμά μας η ομίχλη θα ξεκινά από μπλε και θα καταλήγει μοβ, η ακτίνα της θα είναι από 150 έως 500 texels, οι αλλαγές θα διαρκούν 5 δευτερόλεπτα και η μέγιστη ένταση της φωτεινότητας θα είναι 1000, όπως φαίνεται από τις παρακάτω ρυθμίσεις:

Πεδίο	Τιμή	Πεδίο	Τιμή
clrEnd	255 0 128	fogRadiusStart	150
clrStart	0 0 160	fogSpeed	5000
fogRadiusEnd	500	fogVariance	1000

Πολύ υψηλές τιμές στο πεδίο fogVariance μπορεί να προκαλέσουν ανεπιθύμητα αποτελέσματα κατά την εναλλαγή των χρωμάτων.

18.8 ParticleSystemProxy

Είδαμε ότι με την οντότητα FloatingParticles μπορούμε να δημιουργήσουμε το εφέ των αιωρούμενων σωματιδίων. Όμως στο Reality Factory μπορούμε να δημιουργήσουμε δώδεκα ακόμα εφέ σωματιδίων, απλώς με τη χρήση μίας και μόνο οντότητας, της ParticleSystemProxy. Να σημειώσουμε ότι, για να εμφανιστούν τα εφέ που χειρίζεται η ParticleSystemProxy, πρέπει οπωσδήποτε να ορίσουμε μία εικόνα για τα σωματίδια και μία εικόνα για το διαφανές τμήμα της πρώτης. Οι ιδιότητες της ParticleSystemProxy είναι:

Πεδίο	Περιγραφή
BirthRate	Ο αριθμός των σωματιδίων που δημιουργούνται ανά δευτερόλεπτο.
BoneName	Το όνομα του κοκάλου ενός actor στο οποίο συνδέεται η οντότητα ParticleSystemProxy.
clrColor	Το χρώμα των σωματιδίων.
EntityName	Το όνομα της οντότητας στην οποία συνδέεται η ParticleSystemProxy.
fRadius	Η μέγιστη ακτίνα σε texels που μπορεί να ακουστεί ο ήχος που συνοδεύει την οντότητα.

Πεδίο	Περιγραφή
MaxParticleCount	Ο μέγιστος αριθμός σωματιδίων που μπορεί να είναι ενεργός σε κάθε δεδομένη χρονική στιγμή.
Model	Το όνομα του μοντέλου στο οποίο συνδέεται η οντότητα ParticleSystemProxy.
nStyle	Το είδος των σωματιδίων που παράγει η ParticleSystemProxy. Οι τιμές είναι από 1-12.*
Origin	Η θέση της οντότητας. Η τιμή συμπληρώνεται αυτόματα.
ParticleLifespan	Ο χρόνος σε χιλιοστά του δευτερολέπτου που διαρκεί η ζωή των σωματιδίων.
Scale	Η κλίμακα των σωματιδίων.
szAlpha	Το όνομα της εικόνας για το διαφανές μέρος της υφής που “ντύνει” τα σωματίδια.
szEntityName	Το όνομα της συγκεκριμένης οντότητας ParticleSystemProxy. Αυτό το όνομα θα χρησιμοποιηθεί σε πεδία ιδιοτήτων άλλων οντοτήτων που θέλουμε να συνδέονται με αυτή.
szSoundFile	Ποιο αρχείο ήχου θα εκτελείται μαζί με το εφέ.
szTexture	Το όνομα της εικόνας που χρησιμοποιείται για την υφή των σωματιδίων.
TriggerName	Το όνομα της σκανδάλης που ενεργοποιεί τη συγκεκριμένη οντότητα ParticleSystemProxy.

* Οι τιμές του πεδίου nStyle και τα αποτελέσματά τους είναι:

Τιμή nStyle	Περιγραφή
1	SHOCKWAVE, τα σωματίδια εκπέμπονται στο οριζόντιο επίπεδο.
2	FOUNTAIN, τα σωματίδια εκπέμπονται προς τα πάνω και υπολογίζεται η ταχύτητα και η επίδραση της βαρύτητας.
3	RAIN, τα σωματίδια εκπέμπονται σε τυχαίες αποστάσεις στο οριζόντιο και στο κάθετο επίπεδο.
4	SPHERE, τα σωματίδια εκπέμπονται με τυχαίες διανυσματικές ταχύτητες.
5	COLUMN, τα σωματίδια εκπέμπονται προς τα πάνω και υπολογίζεται μόνο η ταχύτητα και όχι η επίδραση της βαρύτητας.
6	EXPLOSIVE ARRAY, τα σωματίδια εκπέμπονται προς τα πάνω από το επίπεδο εκπομπής.

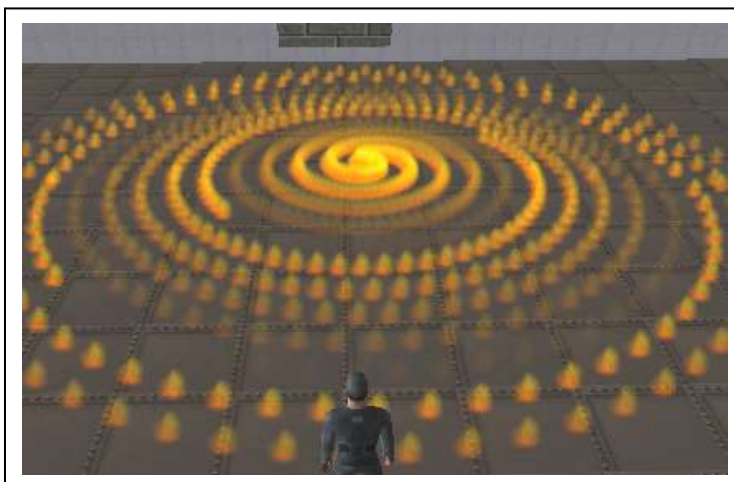
Τιμή nStyle	Περιγραφή
7	SPIRAL ARM, τα σωματίδια εκπέμπονται κατά τέτοιον τρόπο ώστε να δημιουργείται το σχήμα σπειροειδούς γαλαξία.
8	TRAIL, τα σωματίδια εκπέμπονται κατά τέτοιον τρόπο ώστε να δημιουργείται ένα μονοπάτι από τα ίχνη τους.
9	GUARDIAN, τα σωματίδια εκπέμπονται κατά τέτοιον τρόπο ώστε να δημιουργείται το σχήμα σπειροειδούς κυλίνδρου.
10	IMPLODING SPHERE, το αντίθετο από το εφέ που προκαλεί η τιμή 4.
11	IMPLODING SHOCKWAVE, το αντίθετο από το εφέ που προκαλεί η τιμή 1.
12	IMPLODING SPIRAL ARM, το αντίθετο από το εφέ που προκαλεί η τιμή 7.

Για να δώσουμε ένα παράδειγμα χρήσης της ParticleSystemProxy (εικόνα 18.7), θα κάνουμε τις εξής ρυθμίσεις στις ιδιότητές της:

Πεδίο	Τιμή
BirthRate	100
MaxParticleCount	2000
nStyle	7
ParticleLifespan	10000

Πεδίο	Τιμή
Scale	6
szAlpha	a_flame.bmp*
szTexture	flame03.bmp*

* Οι εικόνες περιλαμβάνονται στην εγκατάσταση του Reality Factory.



Εικόνα 18.7 Η οντότητα ParticleSystemProxy (nStyle 7)

Για να παρακαμφθούν οι προεπιλεγμένες ρυθμίσεις της καθεμίας από τις παραπάνω τιμές `nStyle`, τα πεδία `BirthRate`, `MaxParticleCount` και `ParticleLifespan` δεν πρέπει να είναι κενά.

18.9 Rain

Παρότι το εφέ της βροχής μπορεί να δημιουργηθεί και με την οντότητα `Floating-Particles`, εντούτοις στο `Reality Factory` υπάρχει ειδική οντότητα επιφορτισμένη με αυτή την εργασία, η `Rain`. Πρέπει να σημειώσουμε ότι λόγω ενός σφάλματος στο `Reality Factory`, η βροχή διαπερνά τα κτίρια. Συνεπώς, η οντότητα της βροχής είναι καλό να χρησιμοποιείται σε εφαρμογές που υλοποιούν ανοιχτούς χώρους.

Ένας τρόπος για να ξεπεραστεί αυτό το πρόβλημα είναι να τοποθετηθούν αρκετές οντότητες `Rain` με μικρή ακτίνα δράσης η καθεμία αντί για μία με μεγάλη ακτίνα δράσης. Έτσι, μπορούμε να τοποθετήσουμε τις οντότητες σε τέτοιες θέσεις που το/τα κτίρια να μην βρίσκονται μέσα στην εμβέλειά τους. Εξάλλου, κάτι τέτοιο μπορεί να μας βοηθήσει να δημιουργήσουμε εναλλαγές στην ένταση της βροχής, δίνοντας διαφορετικές τιμές στην αντίστοιχη ιδιότητα σε ορισμένες από τις οντότητες `Rain`. Αναλυτικά οι ιδιότητές της είναι:

Πεδίο	Περιγραφή
<code>AlphaName</code>	Το όνομα της εικόνας για το διαφανές μέρος της υφής που “ντύνει” τα σωματίδια της βροχής. Δε χρειάζεται να δηλώσουμε κάποιο αρχείο, εκτός και αν έχουμε κατασκευάσει δική μας βροχή.
<code>BmpName</code>	Το όνομα της υφής που “ντύνει” τα σωματίδια της βροχής. Δε χρειάζεται να δηλώσουμε κάποιο αρχείο, εκτός και αν έχουμε κατασκευάσει δική μας βροχή.
<code>BoneName</code>	Το όνομα του κοκάλου ενός <code>actor</code> στο οποίο συνδέεται η οντότητα <code>Rain</code> .
<code>ColorMax</code>	Η μέγιστη τιμή του χρώματος που μπορεί να πάρει κάθε σταγόνα.
<code>ColorMin</code>	Η ελάχιστη τιμή του χρώματος που μπορεί να πάρει κάθε σταγόνα.
<code>DropLife</code>	Η διάρκεια ζωής σε δευτερόλεπτα της κάθε σταγόνας.
<code>EntityName</code>	Το όνομα της οντότητας στην οποία συνδέεται η <code>Rain</code> .
<code>Gravity</code>	Η επιτάχυνση της βαρύτητας σε κάθε σωματίδιο της βροχής, κάθε δευτερόλεπτο, στους άξονες <code>x</code> , <code>y</code> και <code>z</code> .
<code>Model</code>	Το όνομα του μοντέλου στο οποίο συνδέεται η οντότητα <code>Rain</code> .
<code>Origin</code>	Η θέση της οντότητας. Η τιμή συμπληρώνεται αυτόματα.

Πεδίο	Περιγραφή
Radius	Η ακτίνα της παραγόμενης βροχής.
Severity	Η ένταση της βροχής σε δεκαδικό αριθμό (0=ψιλόβροχο, 1=σφοδρή καταιγίδα).
szEntityName	Το όνομα της συγκεκριμένης οντότητας Rain. Αυτό το όνομα θα χρησιμοποιηθεί σε πεδία ιδιοτήτων άλλων οντοτήτων που θέλουμε να συνδέονται με αυτή.
TriggerName	Το όνομα της σκανδάλης που ενεργοποιεί τη συγκεκριμένη οντότητα Rain.

Στο παράδειγμα χρήσης της Rain που θα δώσουμε στη συνέχεια (εικόνα 18.8), θα υλοποιήσουμε την ενεργοποίηση της βροχής υπό συνθήκες. Όπως κάθε εφέ, η βροχή επιβαρύνει την εφαρμογή. Καλό είναι λοιπόν να ξεκινά το εφέ όποτε το χρειαζόμαστε και να σταματά όταν δεν το χρειαζόμαστε πια, για παράδειγμα όταν ο χρήστης δεν μπορεί πλέον να το δει.

Έτσι, θα περιβάλουμε το δωμάτιο των προηγούμενων παραδειγμάτων μ' ένα μεγαλύτερο, που θα παίξει το ρόλο του ανοιχτού χώρου. Κατά τα γνωστά, μετατρέπουμε τις έδρες στου εξωτερικού στερεού σε Sky και δίνουμε τις ανάλογες υφές. Ανοίγουμε και μία πόρτα στο δωμάτιό μας, έτσι ώστε ο χρήστης να μπορεί να βγει έξω. Στην πόρτα, στη μεριά του δωματίου, τοποθετούμε ένα λεπτό στερεό, ώστε ο χρήστης να μπορεί να ανέβει σ' αυτό, και αρκετά πλατύ, ώστε να είναι σίγουρο ότι ο χρήστης θα το πατήσει. Το στερεό το μετατρέπουμε σε μοντέλο και του δίνουμε το όνομα rainstartmodel.

Η βασική ιδέα είναι όταν ο χρήστης θα θελήσει να βγει έξω από το δωμάτιο να αρχίζει η βροχή και όταν επιστρέφει σ' αυτό η βροχή να σταματά. Με άλλα λόγια, τοποθετώντας το μοντέλο στην είσοδο, όταν ο χρήστης πατήσει αυτό, θα αρχίζει η βροχή και θα σταματά όταν θα ξαναπατήσει σ' αυτό, δηλαδή θα έχουμε κάτι σαν “διακόπτη” βροχής. Πέρα από την οντότητα Rain θα χρειαστούμε μία Σκανδάλη-Trigger, μία Λογική Πύλη-Logic Gate και μία οντότητα AudioSource3D. Τοποθετούμε τη Rain και την AudioSource3D κοντά τη μία στην άλλη και παραμετροποιούμε όλες τις οντότητες ως εξής:

Για την AudioSource3D

Πεδίο	Τιμή
bLoopSound	TRUE
szSoundFile	rain.wav*
TriggerName	logicrain

Για την Trigger

Πεδίο	Τιμή
Model	rainstartmodel
PlayerOnly	TRUE
szEntityName	trrain
TimeOff	1
TimeOn	1

* Το αρχείο ήχου δεν περιλαμβάνεται στην εγκατάσταση του Reality Factory.

Για τη LogicGate

Πεδίο	Τιμή
szEntityName	logicrain
Trigger1Name	trrain
Type	5

Για τη Rain

Πεδίο	Τιμή
DropLife	2
Radius	400
Severity	.8

Αν θέλαμε η βροχή να μη λειτουργεί με το διακόπτη της LogicGate, θα μπορούσαμε να δώσουμε μία πολύ μεγάλη τιμή στα πεδία TimeOn και TimeOff της Trigger και με τον τρόπο αυτό να έχουμε συγκεκριμένη διάρκεια βροχής.

TriggerName logicrain



Εικόνα 18.8 Η οντότητα Rain

18.10 Spout

Μία ακόμα οντότητα που χειρίζεται σωματίδια είναι και η Spout. Είναι κατάλληλη για την παραγωγή ταχύτητα κινούμενων υγρών, όπως πίδακες, καταρράκτες, κτλ. Όπως και οι περισσότερες οντότητες, αν κριθεί σκόπιμο, μπορεί και αυτή να ελεγχθεί από σκανδάλη. Οι ιδιότητές της είναι:

Πεδίο	Περιγραφή
AlphaName	Το όνομα της εικόνας για το διαφανές μέρος της υψής που “ντύνει” τα σωματίδια.
Angles	Οι γωνίες στους άξονες x, y και z απ’ όπου θα πηγάζουν τα σωματίδια.
BmpName	Το όνομα της εικόνας που “ντύνει” τα σωματίδια.
BoneName	Το όνομα του κοκάλου ενός actor στο οποίο συνδέεται η οντότητα Spout.
Bounce	Αν δώσουμε την τιμή TRUE, τότε τα σωματίδια θα αναπηδούν αν συναντήσουν κάποιο εμπόδιο.
ColorMax	Το μέγιστο χρώμα που θα αποκτά το εφέ.
ColorMin	Το ελάχιστο χρώμα που θα αποκτά το εφέ.
DestVariance	Η τιμή προσδιορίζει το βαθμό διαφοροποίησης του σημείου προορισμού του εφέ.
EntityName	Το όνομα της οντότητας στην οποία συνδέεται η Spout.
Gravity	Η επιτάχυνση της βαρύτητας σε κάθε σωματίδιο της οντότητας, κάθε δευτερόλεπτο, στους άξονες x, y και z.
MaxPauseTime	Η μέγιστη τιμή του χρόνου, σε δευτερόλεπτα, που το εφέ θα σταματά.
MinPauseTime	Η ελάχιστη τιμή του χρόνου, σε δευτερόλεπτα, που το εφέ θα σταματά.
MaxScale	Η μέγιστη τιμή της κλίμακας που μπορεί να έχουν τα σωματίδια.
MinScale	Η ελάχιστη τιμή της κλίμακας που μπορεί να έχουν τα σωματίδια.
MaxSpeed	Η μέγιστη ταχύτητα των σωματιδίων.
MinSpeed	Η ελάχιστη ταχύτητα των σωματιδίων.
MaxUnitLife	Η μέγιστη διάρκεια ζωής των σωματιδίων.
MinUnitLife	Η ελάχιστη διάρκεια ζωής των σωματιδίων.
Model	Το όνομα του μοντέλου στο οποίο συνδέεται η οντότητα Spout.
Origin	Η θέση της οντότητας. Η τιμή συμπληρώνεται αυτόματα.
ParticleCreateRate	Η τιμή προσδιορίζει κάθε πόσα δευτερόλεπτα θα παράγονται τα σωματίδια του εφέ.

Πεδίο	Περιγραφή
SourceVariance	Η τιμή προσδιορίζει το βαθμό διαφοροποίησης του σημείου εκκίνησης του εφέ.
szEntityName	Το όνομα της συγκεκριμένης οντότητας Spout. Αυτό το όνομα θα χρησιμοποιηθεί σε πεδία ιδιοτήτων άλλων οντοτήτων που θέλουμε να συνδέονται με αυτή.
TotalLife	Πόσα δευτερόλεπτα θα διαρκέσει το εφέ. Αν δώσουμε την τιμή 0, το εφέ δε θα σταματά.
TriggerName	Το όνομα της σκανδάλης που ενεργοποιεί τη συγκεκριμένη οντότητα Spout.

Στο παράδειγμα που θα δώσουμε θα αναπαραστήσουμε έναν καταρράκτη (εικόνα 18.9). Φροντίζουμε να τοποθετήσουμε την οντότητα σε τέτοια απόσταση από τον τοίχο, ώστε να μην υπάρχει κάποιο τμήμα του εφέ που να βυθίζεται μέσα σ' αυτόν. Επίσης, θα δώσουμε αρνητική τιμή στη βαρύτητα κατά τους άξονες x και y , ώστε το “νερό” να κυλά προς τα κάτω. Το εφέ δε θα έχει χρονικό περιορισμό. Στη Spout χρησιμοποιούμε τις παρακάτω παραμέτρους:

Πεδίο	Τιμή	Πεδίο	Τιμή
AlphaName	Terrain\alpha_clouds4.bmp*	MaxSpeed	30
Angles	0 -90 0	MaxUnitLife	5
BmpName	Terrain\clouds_256.bmp*	MinPauseTime	0
Bounce	TRUE	MinScale	.5
ColorMax	255 255 255	MinSpeed	10
ColorMin	217 236 255	MinUnitLife	3
DestVariance	8	ParticleCreateRate	0.1
Gravity	-60 -60 0	SourceVariance	1
MaxPauseTime	0	TotalLife	0
MaxScale	1		

* Τα αρχεία εικόνων περιλαμβάνονται στην εγκατάσταση του Reality Factory.



Εικόνα 18.9 Η οντότητα Spout

18.11 VideoTextureReplacer

Στο Reality Factory έχουμε τη δυνατότητα να προβάλουμε αρχεία βίντεο πάνω σε γεωμετρικά στερεά. Το ενδιαφέρον με την οντότητα που χειρίζεται αυτό το εφέ, την VideoTextureReplacer, είναι ότι ενεργοποιείται όταν ο χρήστης πλησιάσει μέσα σε μία καθορισμένη ακτίνα από τη θέση της χωρίς τη χρήση άλλων οντοτήτων. Μπορούμε να ορίσουμε το εφέ να επαναλαμβάνεται συνεχώς ή να παύει μετά την απομάκρυνση του χρήστη από την ακτίνα δράσης της. Εκτός από αρχεία βίντεο, η συγκεκριμένη οντότητα μπορεί να διαχειριστεί και αρχεία .gif που περιέχουν animation (animated gifs).

Η VideoTextureReplacer λειτουργεί μόνο εφόσον το αρχείο .gif ή το βίντεο έχουν διαστάσεις δύναμη του 2 (256x256, 512x512, κτλ.). Επίσης, το αρχείο βίντεο θα πρέπει να είναι της μορφής .avi, είτε πλήρως ασυμπίεστο είτε συμπιεσμένο κατά το πρότυπο Cinepak. Σε κάθε άλλη περίπτωση το Reality Factory ή δε θα εμφανίσει τίποτα ή θα τερματίσει απότομα. Για τη μετατροπή των αρχείων βίντεο στις παραπάνω προδιαγραφές, ο ενδιαφερόμενος μπορεί να χρησιμοποιήσει μία μεγάλη γκάμα εμπορικών ή ελεύθερων στη χρήση προγραμμάτων. Ενδεικτικά και μόνο, ένα τέτοιο πρόγραμμα είναι το Video Converter, διαθέσιμο στη διεύθυνση:
http://www.free-codecs.com/Free_Video_Converter_download.htm

Τα αρχεία βίντεο ή .gif πρέπει να είναι αποθηκευμένα στο φάκελο
C:\RealityFactory\Media\Video.

Οι ιδιότητες της οντότητας είναι:

Πεδίο	Περιγραφή
OnlyPlayInRadius	Αν δώσουμε την τιμή TRUE, τότε το βίντεο παίζει μόνο όταν ο χρήστης βρίσκεται μέσα στην ακτίνα που ορίζεται στο πεδίο Radius. Με την τιμή FALSE, το βίντεο παίζει συνεχώς.
Origin	Η θέση της οντότητας. Η τιμή συμπληρώνεται αυτόματα.
Radius	Η ακτίνα από την VideoTextureReplacer μέσα στην οποία πρέπει να βρεθεί ο χρήστης ώστε να ενεργοποιηθεί η οντότητα. Αν δώσουμε την τιμή 0.0, η οντότητα είναι συνεχώς ενεργή.
szTextureName	Το όνομα της υφής πάνω στην οποία θα προβάλλεται το βίντεο. Όπως και με τη FlipBook, εφόσον η υφή βρίσκεται μέσα στο πακέτο υφών που χρησιμοποιούμε, δε χρειάζεται να δώσουμε την κατάληξη .bmp. Τα ονόματα των υφών του πακέτου φαίνονται στην καρτέλα των υφών στο πάνελ εντολών.
szVideoName	Το όνομα του αρχείου βίντεο ή κινούμενης εικόνας .gif που θα εκτελείται.

Όπως και με τη FlipBook, εισάγουμε στο δωμάτιο που χρησιμοποιούμε στα παραδείγματά μας ένα στερεό κύβο και διαμορφώνουμε τις διαστάσεις του ώστε να γίνει ένα λεπτό ορθογώνιο παραλληλεπίπεδο και το τοποθετούμε ως πίνακα σ' έναν από τους τοίχους. Η υφή που θα του δώσουμε είναι η ALCHEM3 του βασικού πακέτου υφών. Τοποθετούμε τη VideoTextureReplacer κοντά στον πίνακα. Οι ρυθμίσεις που κάνουμε στην οντότητα είναι:

Πεδίο	Τιμή
OnlyPlayInRadius	TRUE
Radius	150
szTextureName	ALCHEM3
szVideoName	rfsplash.avi*

* Το αρχείο βίντεο περιλαμβάνεται στην εγκατάσταση. Οι διαστάσεις του όμως πρέπει να μεταβληθούν έτσι, ώστε να πληρούνται οι προδιαγραφές που επισημάναμε.

18.12 ViewSwitch

Αρκετές φορές είναι χρήσιμο να αλλάζει η προοπτική με την οποία ο χρήστης βλέπει την εφαρμογή (από πρώτου προσώπου σε τρίτου ή το αντίστροφο), έτσι ώστε να επισημανθεί κάποιο στοιχείο. Στο Reality Factory μπορούμε να το κάνουμε αυτό αυτόμα-

τα με τη χρήση της οντότητας ViewSwitch. Για την ενεργοποίησή της χρειάζεται να χρησιμοποιήσουμε και μία σκανδάλη. Αναλυτικά οι ιδιότητές της είναι:

Πεδίο	Περιγραφή
LevelView	Η τιμή του πεδίου αυτού καθορίζει με ποια προοπτική θα βλέπει ο χρήστης την εφαρμογή μόλις ενεργοποιηθεί η ViewSwitch. Η τιμή 0 είναι για προοπτική πρώτου προσώπου, η τιμή 1 για προοπτική τρίτου προσώπου και η τιμή 2 για ισομετρική προβολή.
Origin	Η θέση της οντότητας. Η τιμή συμπληρώνεται αυτόματα.
Restore	Αν δώσουμε την τιμή TRUE, τότε η προοπτική με την οποία ο χρήστης βλέπει την εφαρμογή θα επανέλθει στην αρχική της ρύθμιση, όταν η σκανδάλη απενεργοποιηθεί.
TriggerName	Το όνομα της σκανδάλης που ενεργοποιεί τη συγκεκριμένη οντότητα ViewSwitch.

Για τις ανάγκες του παραδείγματός μας θα κατασκευάσουμε δύο δωμάτια και θα τα ενώσουμε με μία πόρτα. Στο πρώτο δωμάτιο, μπροστά στην πόρτα, θα τοποθετήσουμε, όπως κάναμε και στο παράδειγμα της βροχής, ένα στερεό στο οποίο ο χρήστης οπωσδήποτε θα πατήσει για να περάσει στο δεύτερο δωμάτιο. Το στερεό αυτό θα το μετατρέψουμε σε μοντέλο με το όνομα viewswitchmodel.

Η σκανδάλη θα είναι συνδεδεμένη με το παραπάνω μοντέλο και θα την ονομάσουμε trswitchview. Θα παραμένει ενεργή για 10 δευτερόλεπτα και μετά θα απενεργοποιείται. Η ViewSwitch θα είναι συνδεδεμένη με αυτή τη σκανδάλη και θα αλλάζει την προβολή από πρώτο σε τρίτο πρόσωπο για χρόνο ίσο με το διάστημα που μένει ενεργή η σκανδάλη και μετά θα επαναφέρει την προβολή σε πρώτο πρόσωπο. Οι ρυθμίσεις λοιπόν στις οντότητες είναι:

Για την Trigger

Πεδίο	Τιμή
bOneShot	FALSE
Model	viewswitchmodel
szEntityName	trswitchview
TimeOff	1
TimeOn	10

Για τη ViewSwitch

Πεδίο	Τιμή
LevelView	1
Restore	TRUE
TriggerName	trswitchview

18.13 WallDecal

Οι τοίχοι, τα πατώματα ή τα ταβάνια έχουν συνήθως ψεγάδια, ρωγμές και γενικά ατέλειες. Είναι δυνατόν η εικόνα της υφής να περιλαμβάνει αυτές τις ατέλειες. Όμως, ε-

πειδή η εικόνα επαναλαμβάνεται επάνω στην επιφάνεια του στερεού που παίζει το ρόλο του τοίχου, επαναλαμβάνονται και οι ατέλειες, πράγμα που οδηγεί σε αντιαισθητικά αποτελέσματα. Με την οντότητα WallDecal είναι δυνατό να προβάλουμε κάποια άλλη εικόνα πάνω σε μία υφή, όπως και με τη FlipBook, με τις εξής διαφορές: α) μπορούμε να ορίσουμε με ακρίβεια τη γωνία προβολής και β) η εικόνα δεν αντικαθιστά την υφή του τοίχου, αλλά προβάλλεται πάνω σ' αυτή. Οι ιδιότητες της οντότητας είναι:

Πεδίο	Περιγραφή
Alpha	Ο βαθμός διαφάνειας του εφέ.
AlphaName	Το όνομα βάσης της/των εικόνων για το διαφανές μέρος της εικόνας του εφέ (όπως και στη FlipBook).
Angle	Οι γωνίες στους άξονες x, y και z που πρέπει να περιστραφεί η οντότητα ώστε να προβάλλει το εφέ στον τοίχο.
Animated	Αν δώσουμε την τιμή TRUE, τότε το εφέ έχει animation (όπως και στη FlipBook).
BitmapCount	Ο αριθμός των εικόνων που συγκροτούν το animation.
BmpName	Το όνομα βάσης της/των εικόνων που χρησιμοποιεί το εφέ της WallDecal (όπως και στη FlipBook).
BoneName	Το όνομα του κοκάλου του actor στο οποίο συνδέεται η οντότητα WallDecal.
Color	Το χρώμα του εφέ.
EntityName	Το όνομα της οντότητας στην οποία συνδέεται η WallDecal.
Height	Το ύψος σε texels της/των εικόνων του εφέ.
Model	Το όνομα του μοντέλου στο οποίο συνδέεται η οντότητα WallDecal.
Origin	Η θέση της οντότητας. Η τιμή συμπληρώνεται αυτόματα.
Speed	Ο αριθμός των εικόνων που θα προβάλλονται το δευτερόλεπτο.
Style	Αν δώσουμε την τιμή 0, τότε θα προβάλλεται μόνο η πρώτη εικόνα, με την τιμή 1 θα προβάλλονται οι εικόνες ξανά από την αρχή, με την τιμή 2 θα αντιστρέφεται η σειρά προβολής και με την τιμή 3 θα προβάλλονται με τυχαία σειρά.
szEntityName	Το όνομα της συγκεκριμένης οντότητας WallDecal. Αυτό το όνομα θα χρησιμοποιηθεί σε πεδία ιδιοτήτων άλλων οντοτήτων που θέλουμε να συνδέονται με αυτή.
TriggerName	Το όνομα της σκανδάλης που ενεργοποιεί τη συγκεκριμένη οντότητα WallDecal.
Width	Το πλάτος σε texels της/των εικόνων του εφέ.

Για να λειτουργήσει το animation στη WallDecal, πρέπει οπωσδήποτε η οντότητα να ενεργοποιείται από σκανδάλη. Αν θέλουμε να προβάλλεται απλώς και μόνο μία στατική εικόνα πάνω σε μία υφή, δε χρειάζεται ενεργοποίηση της οντότητας.

Για τις ανάγκες του παραδείγματός μας θα τοποθετήσουμε μία WallDecal με animation πάνω σ' ένα τοίχο (εικόνα 18.10). Η οντότητα θα ενεργοποιείται όταν ο χρήστης πλησιάζει σ' ένα σημείο του χώρου. Για το λόγο αυτό, πέρα από τη σκανδάλη, θα χρησιμοποιήσουμε και μία AreaSwitch. Τοποθετούμε τη WallDecal κοντά σ' έναν τοίχο και κοντά της τοποθετούμε την AreaSwitch. Οι ρυθμίσεις που κάνουμε στις οντότητες είναι:

Για την Trigger

Πεδίο	Τιμή
bOneShot	FALSE
PlayerOnly	FALSE
szEntityName	trwalldecal

Για την AreaSwitch

Πεδίο	Τιμή
Diameter	200
InState	TRUE
Trigger	trwalldecal

Για τη WallDecal

Πεδίο	Τιμή	Πεδίο	Τιμή
AlphaName	Explode\explo1\greys\explo*	Height	200
Angle	0 90 0	Speed	25
Animated	TRUE	Style	1
BitmapCount	40	TriggerName	trwalldecal
BmpName	Explode\explo1\explo*	Width	200
Color	255 100 100		

* Τα αρχεία εικόνων περιλαμβάνονται στην εγκατάσταση του Reality Factory.



Εικόνα 18.10 Η οντότητα WallDecal

Κεφάλαιο 19^ο

Ειδικά θέματα

Στο κεφάλαιο αυτό θα ασχοληθούμε με εξειδικευμένα θέματα κατασκευής εφαρμογών για το Reality Factory, και κυρίως με τις μεθόδους που χρησιμοποιούμε έτσι ώστε αυτές να αποδίδουν καλύτερα. Θα ασχοληθούμε επίσης με την προσθήκη χαρακτηριστικών με ιδιαίτερο ενδιαφέρον που όμως δεν έχουν καλυφθεί από τα προηγούμενα κεφάλαια.

19.1 Τηλεμεταφορά

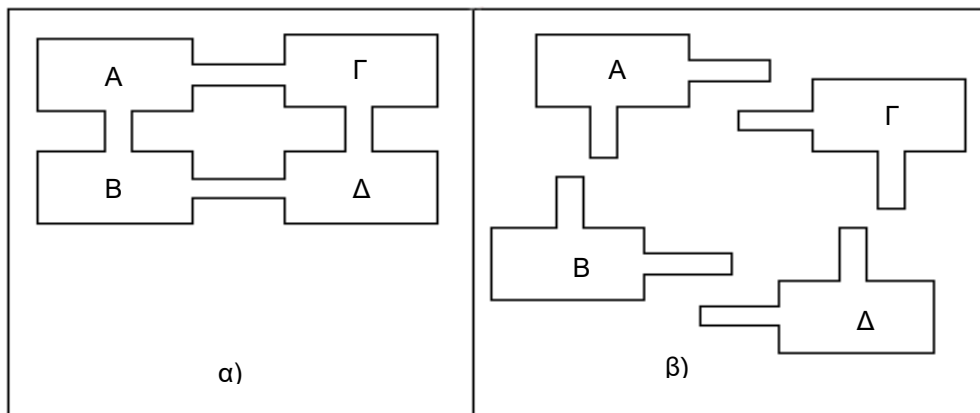
Το πρώτο θέμα που θα μας απασχολήσει είναι τα προβλήματα που προκύπτουν από την περιπλοκότητα της εφαρμογής. Μία απλή εφαρμογή συνήθως δε δημιουργεί προβλήματα διαχείρισης των γεωμετρικών στερεών και των οντοτήτων που περιλαμβάνει. Καθώς όμως προσθέτουμε περισσότερα στοιχεία, η ακριβής τοποθέτησή τους και η επιλογή τους από τις τέσσερις προεπισκοπήσεις μπορεί να γίνει ιδιαίτερα επίπονη εργασία. Επιπρόσθετα, μπορεί να εμφανιστούν προβλήματα κατά την εκτέλεση της εφαρμογής, για παράδειγμα αυξημένος χρόνος “φορτώματος”, μικρός αριθμός καρέ/δευτερόλεπτο, κτλ. Για να αντιμετωπιστούν όλα τα παραπάνω, χρειάζεται να καταφύγουμε σε τεχνικές, έτσι ώστε να “ελαφρύνει” η εφαρμογή. Η κυριότερη απ’ αυτές είναι το “σπάσιμο” του αρχικού χώρου σε μικρότερους και η μεταξύ τους σύνδεση με κάποιον τρόπο.

Για να δώσουμε ένα παράδειγμα, ας υποθέσουμε ότι θέλουμε να αναπαραστήσουμε ένα χώρο που περιλαμβάνει τέσσερα δωμάτια τα οποία συνδέονται με διαδρόμους. Η πιο απλή προσέγγιση θα ήταν να κατασκευάσουμε το πρώτο δωμάτιο, το διάδρομο, το δεύτερο δωμάτιο, κ.ο.κ. (σχήμα 19.1α). Το πρώτο σημείο που πρέπει να προσέξουμε είναι ότι τα διάφορα δομικά στοιχεία πρέπει να τοποθετηθούν με ακρίβεια, έτσι ώστε να μην υπάρχουν κενά μεταξύ τους, γεγονός που οδηγεί σε αδυναμία εκτέλεσης της εφαρμογής.

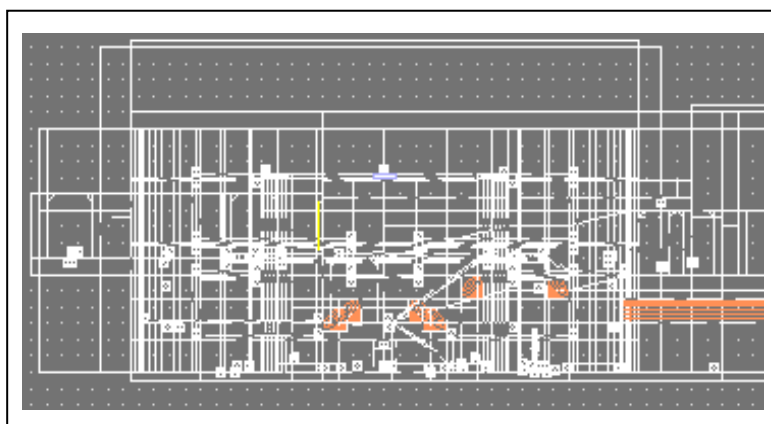
Επίσης, επειδή όλα τα δωμάτια θα βρίσκονται στο ίδιο επίπεδο, είναι εξαιρετικά δύσκολο στις δύο πλάγιες προεπισκοπήσεις να διακρίνουμε ποια στοιχεία ανήκουν σε ποιο δωμάτιο (εικόνα 19.1). Το τελευταίο και πιο σημαντικό σημείο είναι ότι το Reality Factory τοποθετεί όλες τις οντότητες και σχεδιάζει όλους τους χώρους εκείνους που μπορεί να είναι ορατοί από το χρήστη σε κάθε δεδομένη χρονική στιγμή. Αυτό σημαίνει ότι μπορεί να έχουμε ένα μεγάλο αριθμό οντοτήτων που πρέπει να κινηθούν και πολυγώνων που πρέπει να σχεδιαστούν ακόμα και αν είναι ορατοί για μία απειροελάχιστη στιγμή.

Η λύση είναι τα τέσσερα δωμάτια να αποτελούν ανεξάρτητους και κλειστούς χώρους (σχήμα 19.1β). Έτσι, το Reality Factory υποχρεώνεται να σχεδιάσει μόνο το δωμάτιο που βρίσκεται ο χρήστης και τίποτε άλλο. Πώς όμως θα είναι εφικτή η μετάβαση του

χρήστη στο επόμενο δωμάτιο; Αυτό γίνεται με τη χρήση ενός ζεύγους οντοτήτων, της Teleporter και της TeleportTarget.



Σχήμα 19.1 Ενιαία και κερματισμένη κατασκευή εφαρμογής



Εικόνα 19.1 Περίπλοκη δομή επιπέδου

Τοποθετούμε την Teleporter στο δωμάτιο αφετηρίας και την TeleportTarget στο δωμάτιο προορισμού. Οι ιδιότητες της κάθε οντότητας είναι:

Για την TeleportTarget:

Πεδίο	Περιγραφή
Angle	Οι γωνίες στους άξονες x, y και z που είναι τοποθετημένη η Teleport–Target.
Name	Το όνομα της συγκεκριμένης οντότητας TeleportTarget. Το όνομα αυτό θα χρησιμοποιηθεί σε ανάλογο πεδίο της Teleporter.

Πεδίο	Περιγραφή
Origin	Η θέση της οντότητας. Η τιμή συμπληρώνεται αυτόματα.
UseAngle	Αν δώσουμε την τιμή TRUE, τότε ο χρήστης θα περιστραφεί έτσι ώστε οι γωνίες στους άξονες x, y και z να ταυτίζονται με τις γωνίες που είναι τοποθετημένη η TeleportTarget.
UseOffset	Αν δώσουμε την τιμή TRUE, τότε θα υπάρξει μετατόπιση της θέσης της TeleportTarget, σύμφωνα με τη θέση της Teleporter.

Για την Teleporter:

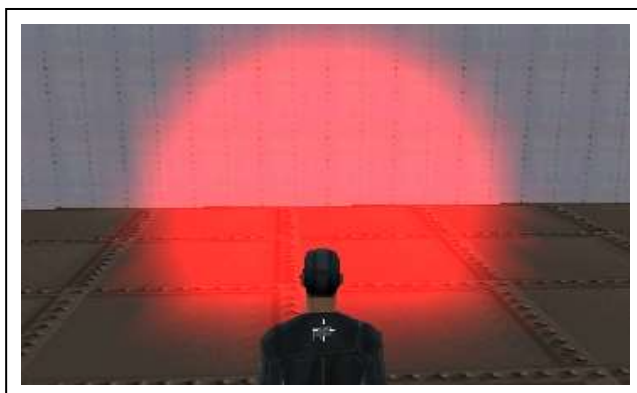
Πεδίο	Περιγραφή
bOneShot	Αν δώσουμε την τιμή TRUE, τότε η τηλεμεταφορά θα λειτουργήσει μόνο μία φορά.
bUseFogEffect	Αν δώσουμε την τιμή TRUE, τότε εμφανίζεται ένα εφέ ομίχλης γύρω από την οντότητα που διευκολύνει το χρήστη να βρει το σημείο τηλεμεταφοράς.
bUseTeleportEffect	Αν δώσουμε την τιμή TRUE, τότε εμφανίζεται ένα δεύτερο εφέ ομίχλης λίγο πριν τηλεμεταφερθεί ο χρήστης.
cColor	Το χρώμα που θα έχει το εφέ ομίχλης γύρω από την οντότητα Teleporter.
cFadeColor	Το χρώμα που θα έχει το εφέ ομίχλης λίγο πριν από την τηλεμεταφορά του χρήστη.
fFadeTime	Ο χρόνος σε χιλιοστά του δευτερολέπτου που θα διαρκεί η τηλεμεταφορά.
fogRadius	Η ακτίνα που θα έχει το εφέ της ομίχλης γύρω από την οντότητα Teleporter.
fogSpeed	Η ταχύτητα σε χιλιοστά του δευτερολέπτου με την οποία θα πάλλεται το εφέ της ομίχλης γύρω από την Teleporter.
fogVariance	Πόσο θα μεταβάλλεται η πυκνότητα της ομίχλης γύρω από την Teleporter.
Model	Το όνομα του μοντέλου με το οποίο συνδέεται η Teleporter.
Origin	Η θέση της οντότητας. Η τιμή συμπληρώνεται αυτόματα.
PlayerOnly	Αν δώσουμε την τιμή TRUE, τότε μόνο ο χρήστης θα μπορεί να ενεργοποιήσει την οντότητα.
szEntityName	Το όνομα της συγκεκριμένης οντότητας Teleporter.
szSoundFile	Το όνομα του αρχείου ήχου που θα εκτελείται με την ενεργοποίηση της οντότητας.

Πεδίο	Περιγραφή
Target	Το όνομα της οντότητας TeleportTarget, που αποτελεί τον προορισμό αυτής της Teleporter.
TriggerName	Το όνομα της σκανδάλης που ενεργοποιεί την Teleporter.

Για να λειτουργήσει η Teleporter πρέπει οπωσδήποτε να συνδέεται με κάποιο μοντέλο. Αν αυτό δε θέλουμε να φαίνεται, μπορούμε στο γεωμετρικό στερεό να δώσουμε την ιδιότητα Clip. Έτσι γίνεται αόρατο, παρ' όλα αυτά όμως υπολογίζεται η σύγκρουση του χρήστη με αυτό και κατ' επέκταση γίνεται η ενεργοποίηση της Teleporter.

Για παράδειγμα, ας υποθέσουμε ότι σ' ένα δωμάτιο έχουμε τοποθετήσει ένα στερεό που το μετατρέψαμε σε μοντέλο και το ονομάσαμε telemodel και μία Teleporter (εικόνα 19.2). Σ' ένα δεύτερο δωμάτιο έχουμε τοποθετήσει μία TeleportTarget με το όνομα teletarget. Οι παράμετροι στην Teleporter μπορεί να είναι:

Πεδίο	Τιμή	Πεδίο	Τιμή
bOneShot	false	fogSpeed	1000
bUseFogEffect	true	fogVariance	500
bUseTeleportEffect	true	Model	telemodel
cColor	255 0 0	PlayerOnly	true
cFadeColor	0 128 255	szSoundFile	ding.wav
fFadeTime	1000	Target	teletarget
fogRadius	100		

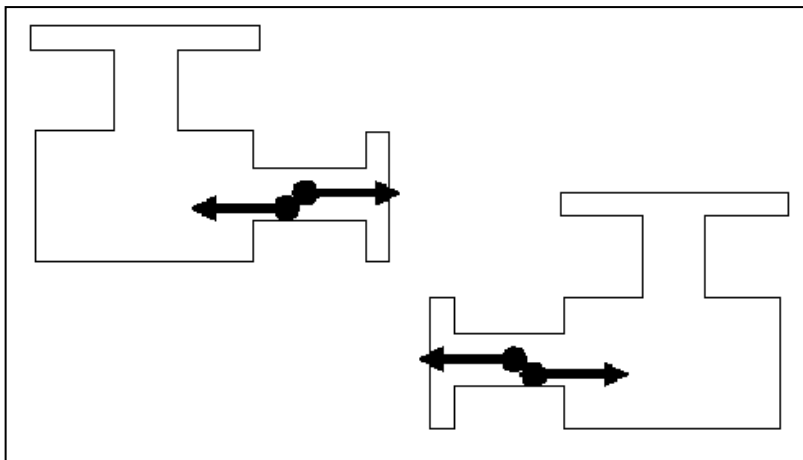


Εικόνα 19.2 Η οντότητα Teleporter

Μπορεί όμως να θέλουμε η τηλεμεταφορά να μη γίνεται αντιληπτή από το χρήστη. Να δημιουργείται δηλαδή η εντύπωση ότι όλοι οι χώροι είναι ενιαίοι. Σ' αυτή την περίπτωση η Teleporter πρέπει να είναι αόρατη, αλλά το μοντέλο πρέπει να είναι ορατό, αναπαριστώντας, για παράδειγμα, μία πόρτα. Οι παράμετροι στην Teleporter τότε είναι:

Πεδίο	Τιμή	Πεδίο	Τιμή
bOneShot	false	fogSpeed	0
bUseFogEffect	false	fogVariance	0
bUseTeleportEffect	false	Model	telemodel
cColor	255 0 0	PlayerOnly	true
cFadeColor	0 128 255	szSoundFile	
fFadeTime	0	Target	teletarget
fogRadius	0		

Εννοείται βέβαια ότι, εάν επιθυμούμε να μπορεί ο χρήστης να επιστρέψει στο προηγούμενο δωμάτιο, πρέπει να τοποθετήσουμε άλλο ένα ζεύγος Teleporter/Teleport-Target, με αντίθετη όμως σειρά. Στο αρχικό μας παράδειγμα χρησιμοποιήσαμε διαδρόμους που ένωναν τα δωμάτια. Είναι αρκετά συνηθισμένο να τοποθετούμε οντότητες τηλεμεταφοράς σε διαδρόμους, που μάλιστα τους διαμορφώνουμε έτσι, ώστε σε κάθε άκρο τους να φαίνεται ότι οδηγούν στο επόμενο δωμάτιο. Για να πετύχουμε αυτή την ψευδαίσθηση, κατασκευάζουμε ένα πολύ μικρό τμήμα του επόμενου δωματίου, φροντίζουμε όμως η οντότητα τηλεμεταφοράς να τοποθετηθεί αρκετά πιο πριν (σχήμα 19.2).



Σχήμα

Ένωση δωματίων με διάδρομο και τηλεμεταφορά

19.2

Η τηλεμεταφορά είναι ο καλύτερος τρόπος να αναμείξουμε εσωτερικούς και εξωτερικούς χώρους σε μία εφαρμογή. Όσο ο χρήστης είναι στον εξωτερικό χώρο, τα κτίρια δεν είναι τίποτε παραπάνω από κύβοι ντυμένοι με τις κατάλληλες υφές. Όταν όμως θελήσει να μπει μέσα σ' ένα, πλησιάζοντας την είσοδο, τηλεμεταφέρεται σε άλλο σημείο της εφαρμογής που αναπαριστά το εσωτερικό του κτιρίου.

19.2 Αλλαγή επιπέδου

Η εφαρμογή μας μπορεί να φτάσει σ' έναν τέτοιο βαθμό περιπλοκότητας, ώστε ούτε η τεχνική της τηλεμεταφοράς να μπορεί να μας διευκολύνει. Υπάρχει ακόμα η πιθανότητα να έχει ολοκληρωθεί το αντικείμενο με το οποίο ασχολείται η εφαρμογή και να θέλουμε να “φορτώνεται” αυτόματα η επόμενη. Σε τέτοιες περιπτώσεις χρησιμοποιούμε την οντότητα *ChangeLevel* που μας επιτρέπει να αλλάζουμε επίπεδο. Η χρήση της είναι απλή και αρκεί η σύνδεσή της μ' ένα μοντέλο ή μία σκανδάλη, ενώ προσοχή πρέπει να δοθεί μόνο στο όνομα του επιπέδου που θα φορτώνεται. Αναλυτικότερα οι ιδιότητές της είναι:

Πεδίο	Περιγραφή
EffectColor	Το χρώμα που θα έχει το εφέ μετάβασης στο επόμενο επίπεδο.
Font	Το μέγεθος των γραμμάτων του μηνύματος που θα εμφανίζεται με την αλλαγή του επιπέδου. Το μήνυμα το ορίζουμε στο πεδίο <i>szMessage</i> .
KeepAttributes	Αν δώσουμε την τιμή <i>TRUE</i> , τότε θα διατηρούνται οι τιμές των ιδιοτήτων του χρήστη (για παράδειγμα της υγείας) και στο επόμενο επίπεδο.
Model	Ποιο μοντέλο θα ενεργοποιεί την <i>ChangeLevel</i> .
Origin	Η θέση της οντότητας. Η τιμή συμπληρώνεται αυτόματα.
SplashHold	Αν δώσουμε την τιμή <i>TRUE</i> , τότε η οθόνη φορτώματος που θα ορίσουμε στο πεδίο <i>szSplashFile</i> θα παραμένει μέχρις ότου ο χρήστης πατήσει το πλήκτρο <i>Spacebar</i> .
StartPointName	Μπορούμε να δώσουμε προαιρετικά το όνομα της οντότητας <i>PlayerStart</i> του επόμενου επιπέδου.
szAudioFile	Το όνομα του αρχείου ήχου που θα παίζει κατά την αλλαγή του επιπέδου.
szCutScene	Το όνομα του αρχείου βίντεο (ή Gif) που θα παίζει κατά την αλλαγή του επιπέδου. Για τα αρχεία βίντεο ισχύουν όσα αναφέραμε για την οντότητα <i>VideoTextureReplacer</i> (κεφάλαιο 18.11).

Πεδίο	Περιγραφή
szEntityName	Το όνομα της συγκεκριμένης οντότητας ChangeLevel.
szMessage	Το όνομα του μηνύματος που θα εμφανίζεται κατά το φόρτωμα του επόμενου επιπέδου αντί για εικόνα φορτώματος (που μπορεί να ορίσουμε στο πεδίο szSplashFile).
szNewLevel	Το όνομα του επιπέδου που θα φορτώνεται. Πρέπει να χρησιμοποιηθεί η κατάληξη .bsp, πράγμα που σημαίνει ότι έχει προηγηθεί μεταγλώττιση του επόμενου επιπέδου.
szSplashFile	Το όνομα του αρχείου εικόνας που θα εμφανίζεται κατά το φόρτωμα του επόμενου επιπέδου.
Trigger	Το όνομα της σκανδάλης που θα ενεργοποιεί την ChangeLevel.
TriggerChange	Το όνομα της σκανδάλης που θα ενεργοποιεί <u>υποχρεωτικά</u> την ChangeLevel.
UseAngle	Αν δώσουμε την τιμή TRUE, τότε ο χρήστης θα περιστραφεί έτσι ώστε οι γωνίες του στους άξονες x, y και z να συμφωνούν με αυτές της επόμενης PlayerStart.
UseEffect	Αν δώσουμε την τιμή TRUE, τότε θα χρησιμοποιηθεί το εφέ της μετάβασης στο επόμενο επίπεδο.
UseKey	Αν δώσουμε την τιμή TRUE, τότε η ChangeLevel θα ενεργοποιείται όταν ο χρήστης πατήσει το πλήκτρο χρήσης.
UseOffset	Αν δώσουμε την τιμή TRUE, τότε υπολογίζεται η απόσταση του χρήστη από την ChangeLevel. Αυτή η απόσταση θα εφαρμοστεί κατά την εμφάνιση του χρήστη στο επόμενο επίπεδο σε σχέση με την PlayerStart. Κάτι τέτοιο μάς επιτρέπει να ξεκινά ο χρήστης το επόμενο επίπεδο στην ίδια σχετική θέση με το προηγούμενο.

Μπορούμε να ορίσουμε περισσότερες από μία οθόνες φορτώματος στο πεδίο szSplashFile, οι οποίες θα φορτώνονται με τυχαία σειρά. Για να συμβεί αυτό, πρέπει στο φάκελο C:\RealityFactory\Install να δημιουργήσουμε ένα αρχείο με το όνομα Splash.ini. Το αρχείο θα πρέπει να περιλαμβάνει το όνομα της λίστας, μέσα σε αγκύλες, ακολουθούμενο από τα ονόματα των αρχείων εικόνων, όπως το παρακάτω παράδειγμα:

```
[splashtest]
Splash1.bmp
Splash2.bmp
Splash3.bmp
```

Αν λοιπόν στο πεδίο `szSplashFile` εισαγάγουμε το όνομα `splashtest`, θα φορτώνεται με τυχαίο τρόπο μία από τις εικόνες που δηλώσαμε. Εννοείται ότι τα αρχεία εικόνων πρέπει να βρίσκονται στο φάκελο `C:\RealityFactory\Media\Bitmaps`.

Αν η σκανδάλη που ορίζουμε στο πεδίο `Trigger` είναι σε κατάσταση `off`, τότε δε γίνεται αλλαγή επιπέδου. Αν είναι σε κατάσταση `on`, τότε μπορεί να ενεργοποιήσει μία `LogicGate`, που αυτή με τη σειρά της θα ενεργοποιεί την αλλαγή επιπέδου. Αντίθετα, αν η κατάσταση της σκανδάλης στο πεδίο `TriggerChange` είναι `on`, τότε γίνεται αλλαγή του επιπέδου, ανεξάρτητα από την κατάσταση της σκανδάλης του πεδίου `Trigger`. Το πεδίο `Trigger` το χρησιμοποιούμε όταν θέλουμε η αλλαγή του επιπέδου να συμβαίνει όταν ικανοποιούνται παραπάνω από μία προϋποθέσεις.

Ο χρόνος του φορτώματος του επόμενου επιπέδου, χωρίς να υπολογίζονται εικόνες, ήχοι, βίντεο και μηνύματα φορτώματος, εξαρτάται από το μέγεθός του. Κατά συνέπεια, είναι πιθανό να μην μπορούμε να επιτύχουμε την ψευδαίσθηση της αυτόματης μετάβασης, όπως με τις οντότητες της τηλεμεταφοράς. Ένα παράδειγμα παραμέτρων που μπορούμε να χρησιμοποιήσουμε στην `ChangeLevel` χωρίς εφέ φορτώματος είναι το ακόλουθο:

Πεδίο	Τιμή	Πεδίο	Τιμή
<code>KeepAttributes</code>	<code>true</code>	<code>szSplashFile</code>	-
<code>Model</code>	<code>levelchangemodel</code>	<code>Trigger</code>	-
<code>SplashHold</code>	<code>false</code>	<code>TriggerChange</code>	-
<code>StartPointName</code>	-	<code>UseAngle</code>	<code>true</code>
<code>szAudioFile</code>	-	<code>UseEffect</code>	<code>false</code>
<code>szCutScene</code>	-	<code>UseKey</code>	<code>false</code>
<code>szMessage</code>	-	<code>UseOffset</code>	<code>true</code>
<code>szNewLevel</code>	<code>newlevel.bsp</code>		

19.3 Αντίστροφη μέτρηση

Ένα στοιχείο χρήσιμο σε αρκετές περιπτώσεις είναι η χρονομέτρηση που μπορεί να ενεργοποιηθεί και να απενεργοποιηθεί κατά βούληση και που μπορεί με τη σειρά της να ενεργοποιήσει άλλα γεγονότα. Η οντότητα που διαχειρίζεται τα παραπάνω είναι η `CountDownTimer`, που, όπως υποδηλώνει το όνομά της, ο ρόλος της είναι να κάνει αντίστροφη μέτρηση.

Για να εμφανιστεί όμως ο υπολειπόμενος χρόνος στην οθόνη του χρήστη, πρέπει να προσθέσουμε δηλώσεις σε δύο αρχεία. Το πρώτο είναι το `player.ini` (που βρίσκεται

στο φάκελο C:\RealityFactory\Install). Σ' αυτό προσθέτουμε μία νέα παράμετρο, για παράδειγμα την countdown. Πληκτρολογούμε λοιπόν τις εξής γραμμές:

```
[countdown]
initial = 0
low = 0
high = 1
```

Δεν έχει σημασία τι τιμές θα δώσουμε ως αρχική, μικρότερη και μεγαλύτερη, γιατί η οντότητα CountdownTimer θα τις τροποποιήσει με βάση την τιμή που θα δώσουμε σ' ένα από τα πεδία των ιδιοτήτων της.

Το δεύτερο αρχείο είναι το hud.ini (που βρίσκεται στον ίδιο φάκελο με το player.ini). Σ' αυτό πρέπει να προσθέσουμε τις παρακάτω γραμμές:

```
[countdown]
type = numeric
framex = center
framey = 200
indicatoroffsetx = 0
indicatoroffsety = 0
font = 9
width = 1
active = modify
modifydirection = both
displaytime = 1
```

Οι δηλώσεις framex και framey καθορίζουν τη θέση στην οθόνη όπου θα εμφανιστεί ο υπολειπόμενος χρόνος. Αν η τιμή στο framex είναι θετικός αριθμός, τότε υπολογίζουμε τόσα εικονοστοιχεία όσα ο αριθμός, από τα αριστερά της οθόνης. Αν είναι αρνητικός αριθμός, τότε η μέτρηση γίνεται από τα δεξιά. Αντίστοιχα, αν στο framey δώσουμε θετικό αριθμό, η μέτρηση των εικονοστοιχείων γίνεται από το επάνω μέρος, ενώ με αρνητικό αριθμό από το κάτω. Αν δώσουμε την τιμή center, τότε το χρονόμετρο θα εμφανίζεται στο κέντρο της οθόνης του χρήστη. Στο πεδίο font καθορίζουμε το χρώμα και το μέγεθος των γραμμάτων (έτσι όπως είναι δηλωμένα στο αρχείο menu.ini). Δεκτές γίνονται οι τιμές 7=γαλάζιο χρώμα γραμμάτων, 8=κόκκινο, 9=μπλε, 10=άσπρο.

Αφού κάνουμε τις παραπάνω δηλώσεις στα αρχεία, μπορούμε να τοποθετήσουμε την CountdownTimer στην εφαρμογή μας. Οι ιδιότητές της είναι:

Πεδίο	Περιγραφή
Attribute	Ποια παράμετρος θα μεταβάλλεται. Προεπιλεγμένη είναι η count-down (που ορίσαμε στο αρχείο player.ini).
Origin	Η θέση της οντότητας. Η τιμή συμπληρώνεται αυτόματα.
SoundFile	Ποιος ήχος θα παίζει κατά την αντίστροφη μέτρηση.
StopTrigger	Το όνομα της σκανδάλης που μπορεί να σταματήσει την αντίστροφη μέτρηση.
szEntityName	Το όνομα της συγκεκριμένης οντότητας CountdownTimer.
TimeAmt	Πόσο χρόνο σε δευτερόλεπτα θα μετρήσει η CountdownTimer. Με τη λήξη αυτού του χρόνου η CountdownTimer μπορεί να λειτουργήσει ως σκανδάλη, ενεργοποιώντας άλλες οντότητες.
TriggerName	Το όνομα της σκανδάλης που ενεργοποιεί την CountdownTimer.

Για να δώσουμε ένα παράδειγμα χρήσης της CountdownTimer, θα εισαγάγουμε μία στην εφαρμογή μας και θα την ονομάσουμε countdowntimer. Θα τοποθετήσουμε δύο στερεά, που θα μετατρέψουμε σε μοντέλα. Το ένα, με όνομα modelstart, θα χρησιμοποιηθεί για να εκκινεί η CountdownTimer. Το άλλο, με όνομα modelstop, θα χρησιμοποιηθεί για να σταματά η CountdownTimer. Θα χρειαστούμε επίσης δύο σκανδάλες, με ονόματα trstart και trstop. Η πρώτη θα συνδέεται με το μοντέλο εκκίνησης (θα χρησιμοποιηθεί για να εκκινεί η CountdownTimer) και η δεύτερη με το μοντέλο παύσης της αντίστροφης μέτρησης. Τέλος, θα τοποθετήσουμε και μία AudioSource3D, η οποία θα εκτελείται με τη λήξη της αντίστροφης μέτρησης (εικόνα 19.3). Οι παράμετροι στις οντότητες θα είναι:

Για τη σκανδάλη εκκίνησης (trstart)

Πεδίο	Τιμή
Model	modelstart
szEntityName	trstart
TimeOff	1
TimeOn	1

Για τη σκανδάλη παύσης (trstop)

Πεδίο	Τιμή
Model	modelstop
szEntityName	trstop
TimeOff	1
TimeOn	1

Για την CountdownTimer

Πεδίο	Τιμή
Attribute	countdown
SoundFile	ding.wav*
StopTrigger	trstop
szEntityName	countdowntimer
TimeAmt	30
TriggerName	trstart

Για την AudioSource3D

Πεδίο	Τιμή
szSoundFile	die1.wav*
Trigger	countdowntimer

* Τα αρχεία ήχου περιλαμβάνονται στην εγκατάσταση του RealityFactory.



Εικόνα 19.3 Η οντότητα CountdownTimer

19.4 Μηνύματα στην οθόνη

Ένας τρόπος για να δίνει η εφαρμογή μας πληροφορίες στο χρήστη είναι μέσω ήχων. Ένας άλλος είναι να εμφανίζει τις πληροφορίες με κείμενο στην οθόνη. Για να συμβεί αυτό, πρέπει να χρησιμοποιήσουμε την οντότητα Message. Όπως και με την CountdownTimer, έτσι και με τη Message, πρέπει να προηγηθούν ορισμένες δηλώσεις σε αρχεία πριν μπορέσουμε να τη χρησιμοποιήσουμε.

Το πρώτο αρχείο είναι το Message.ini, στο φάκελο C:\RealityFactory\Install. Ανοίγοντάς το παρατηρούμε ότι περιέχει δύο παραμέτρους, τη Static και τη Special, που προσδιορίζουν τον τύπο του μηνύματος. Και οι δύο περιέχουν αρκετές κοινές δηλώσεις. Αναλυτικά, ένας τύπος μηνύματος μπορεί να περιέχει τα εξής:

Δήλωση	Περιγραφή
type	static. Είναι ο μόνος τύπος που επιτρέπεται μέχρι στιγμής στο RF.
positionx	Αριθμός. Η θέση του μηνύματος στην οθόνη, στον άξονα των x (από την επάνω αριστερή μεριά της οθόνης). Ισχύουν τα ίδια με το framex της CountdownTimer.
positiony	Αριθμός. Όμοια για τον άξονα των y.
fontsize	Αριθμός (7-10). Η γραμματοσειρά που θα χρησιμοποιηθεί.
displaytime	Αριθμός. Τα δευτερόλεπτα τα οποία θα παραμείνει το μήνυμα στην οθόνη.
fadeintime	Αριθμός. Πόσα δευτερόλεπτα θα κάνει το μήνυμα για να μετατραπεί από διαφανές σε αδιαφανές. Προαιρετικό πεδίο.

Δήλωση	Περιγραφή
fadeouttime	Αριθμός. Το αντίστροφο του προηγούμενου. Προαιρετικό πεδίο.
graphic	Όνομα αρχείου. Το αρχείο εικόνας που θα χρησιμοποιηθεί ως πλαίσιο μηνύματος. Προαιρετικό πεδίο.
graphicalpha	Όνομα αρχείου. Το αρχείο διαφανούς μέρους της εικόνας που θα χρησιμοποιηθεί ως πλαίσιο μηνύματος. Προαιρετικό πεδίο.
graphicpositionx	Αριθμός. Η θέση του πλαισίου στον άξονα των x. Προαιρετικό πεδίο.
graphicpositiony	Αριθμός. Η θέση του πλαισίου στον άξονα των y. Προαιρετικό πεδίο.
graphicstyle	Αριθμός (0-4). Το στυλ του πλαισίου. Το 0 προσδιορίζει στατική εικόνα. Οι τιμές 1-4 προσδιορίζουν διάφορους τύπους μετάβασης από τη μία στην άλλη, εάν χρησιμοποιήσουμε αλληλουχία εικόνων. Σ' αυτή την περίπτωση, στο πεδίο graphic θα πρέπει να δηλώσουμε το όνομα βάσης, όπως και με την οντότητα Flip-Book (κεφάλαιο 18.4). Προαιρετικό πεδίο.
graphicfadeintime	Αριθμός. Πόσα δευτερόλεπτα θα κάνει το πλαίσιο για να μετατραπεί από διαφανές σε αδιαφανές. Προαιρετικό πεδίο.
graphicfadeouttime	Αριθμός. Το αντίστροφο του προηγούμενου. Προαιρετικό πεδίο.
graphicframes	Αριθμός. Ο αριθμός των εικόνων που αποτελούν το πλαίσιο (αν στο πεδίο graphicstyle δηλώσαμε τιμές 1-4). Προαιρετικό πεδίο.
graphicspeed	Αριθμός. Πόσες εικόνες θα εμφανίζονται το δευτερόλεπτο. Προαιρετικό πεδίο.

Ενδεικτικά, μπορούμε να προσαρμόσουμε την παράμετρο Static ως εξής:

[Static]

```

type = static           fadeintime = 2
positionx = center      fadeouttime = 2
positiony = 200         fontsize = 7
displaytime = 10

```

Το δεύτερο αρχείο που πρέπει να επεξεργαστούμε είναι το message.txt, που περιέχει όλα τα μηνύματα που θα χρειαστούμε σε μία εφαρμογή. Κάθε μήνυμα έχει το δικό του όνομα που περικλείεται μέσα σε αγκύλες και στις επόμενες γραμμές γράφουμε το κείμενο του μηνύματος. Αν θέλουμε αλλαγή γραμμής, πρέπει σε μία σειρά να εισαγάγουμε τη δήλωση <CR>. Αν στο μήνυμα θέλουμε να εμφανίζεται το όνομα του χρήστη, εισάγουμε σ' αυτό τη δήλωση <Player>.

Για παράδειγμα, μπορούμε να γράψουμε το εξής μήνυμα:

```
[test]
This is a test
<CR>
Of the message entity
```

Στο Reality Factory δεν μπορούμε να χρησιμοποιήσουμε ελληνικούς χαρακτήρες στα μηνύματα.

Αφού έχουμε ολοκληρώσει τη διαμόρφωση των αρχείων message.ini και message.txt, μπορούμε να εισαγάγουμε την οντότητα Message στην εφαρμογή μας. Οι ιδιότητές της είναι:

Πεδίο	Περιγραφή
DisplayType	Ο τύπος του μηνύματος έτσι όπως τον ορίσαμε στο αρχείο message.ini.
Origin	Η θέση της οντότητας. Η τιμή συμπληρώνεται αυτόματα.
szEntityName	Το όνομα της συγκεκριμένης οντότητας Message.
TextName	Το όνομα του μηνύματος που θα εμφανιστεί έτσι όπως το ορίσαμε στο αρχείο message.txt.
TriggerName	Το όνομα της σκανδάλης που ενεργοποιεί τη Message.

Για να δώσουμε ένα παράδειγμα χρήσης της Message, θα χρειαστούμε μία σκανδάλη που θα ονομάσουμε tr, ένα γεωμετρικό στερεό που το μετατρέπουμε σε μοντέλο για να ενεργοποιεί τη σκανδάλη και το ονομάζουμε messagemodel και μία Message (εικόνα 19.4). Οι παράμετροι στις οντότητες θα είναι:

Για την Trigger

Πεδίο	Τιμή
Model	messagemodel
szEntityName	tr
TimeOff	1
TimeOn	1

Για τη Message

Πεδίο	Τιμή
DisplayType	Static
szEntityName	-
TextName	test
TriggerName	tr



Εικόνα 19.4 Η οντότητα Message

19.5 Δημιουργία ανάγλυφου

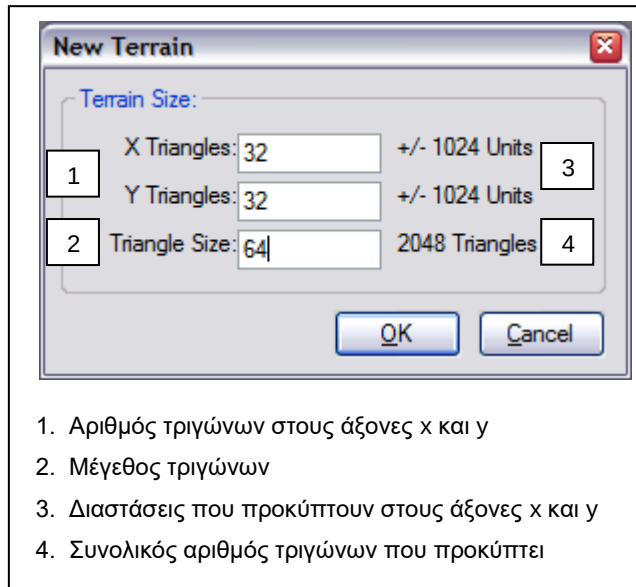
Μέχρι στιγμής, σε όσα παραδείγματα εξωτερικού χώρου έχουμε δώσει, αυτός είναι επίπεδος. Είναι όμως δυνατό να εισαγάγουμε στην εφαρμογή μας λόφους, βουνά, κοιλάδες και ό,τι άλλο αναπαριστά γεωγραφικό ανάγλυφο. Όμως, με το RFEEditPro δεν μπορούμε να παραγάγουμε και να επεξεργαστούμε κάτι τέτοιο και για το λόγο αυτό πρέπει να καταφύγουμε στη χρήση εξωτερικών προγραμμάτων.

Στην εγκατάσταση του Reality Factory περιλαμβάνεται ένα τέτοιο πρόγραμμα, το GenSurf (που βρίσκεται στο φάκελο C:\RealityFactory\Tools), που όμως λόγω των περιορισμένων δυνατοτήτων του δε χρησιμοποιείται. Αντ' αυτού, θα παρουσιάσουμε το Terrain Generator, που διατίθεται δωρεάν στη διεύθυνση <http://nemesis.thewavelength.net/index.php?p=9>. Με την ολοκλήρωση της εγκατάστασής του μπορούμε να το εκκινήσουμε από την “Εναρξη”, “Όλα τα προγράμματα”.

Από τη γραμμή εντολών “File” και την επιλογή “New”, δίνουμε την εντολή για την κατασκευή ενός βασικού επίπεδου χώρου. Αμέσως μας παρουσιάζεται μία καρτέλα, στην οποία ορίζουμε τις διαστάσεις του χώρου αυτού (εικόνα 19.5). Προσοχή πρέπει να δώσουμε στο συνδυασμό του αριθμού των τριγώνων κατά τους άξονες x και y και του μεγέθους του κάθε τριγώνου.

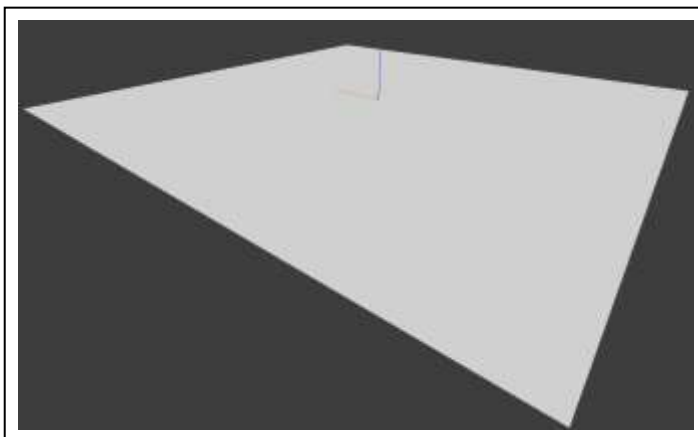
Μικρό μέγεθος μας δίνει μεγαλύτερο επίπεδο λεπτομέρειας, αλλά, για μία δεδομένη έκτασης επιφάνεια, απαιτείται μεγαλύτερος αριθμός τριγώνων, σε σχέση με μικρό αριθμό τριγώνων που όμως έχουν μεγάλο μέγεθος. Για παράδειγμα, 32x32 τρίγωνα μεγέθους 64 texels, παράγουν ένα χώρο 1.024x1.024 texels και απαιτούνται 2048 τρίγωνα. Όμως χώρο 1.024x1.024 texels μπορούμε να παράγουμε και με 16x16 τρίγωνα μεγέθους 128 texels, αλλά απαιτούνται μόλις 512 τρίγωνα. Πρέπει να έχουμε πάντα υπόψη μας ότι ο μεγάλος αριθμός τριγώνων μπορεί να επιβαρύνει σημαντικά την ε-

φαρμογή μας. Να επισημάνουμε ότι στα πεδία καλό είναι να χρησιμοποιούμε τιμές που είναι δυνάμεις του 2.



Εικόνα 19.5 Ορισμός διαστάσεων χώρου

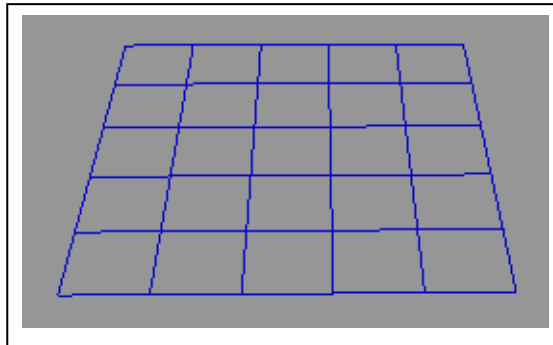
Μόλις δώσουμε τις επιθυμητές διαστάσεις, ερχόμαστε στην κύρια οθόνη του προγράμματος. Με τα βελάκια του πληκτρολογίου μπορούμε να κινηθούμε προς τις τέσσερις κατευθύνσεις, ενώ κρατώντας πατημένο το δεξί πλήκτρο του ποντικιού μπορούμε να περιστραφούμε (εικόνα 19.6).



Εικόνα 19.6 Προεπισκόπηση χώρου

Για την κατασκευή του ανάγλυφου θα χρησιμοποιήσουμε μία σειρά από πλήκτρα που βρίσκονται στη γραμμή εργαλείων. Συγκεκριμένα, με το πλήκτρο  μπορούμε να ανυψώσουμε ή να βυθίσουμε το έδαφος, με το πλήκτρο  προκαλούμε εντονότερες ανυψώσεις ή βυθίσεις, με το  εξομαλύνουμε τις απότομες εσοχές ή εξοχές, με το  παράγουμε τυχαίες εσοχές ή εξοχές και με το  κάνουμε το έδαφος και πάλι επίπεδο.

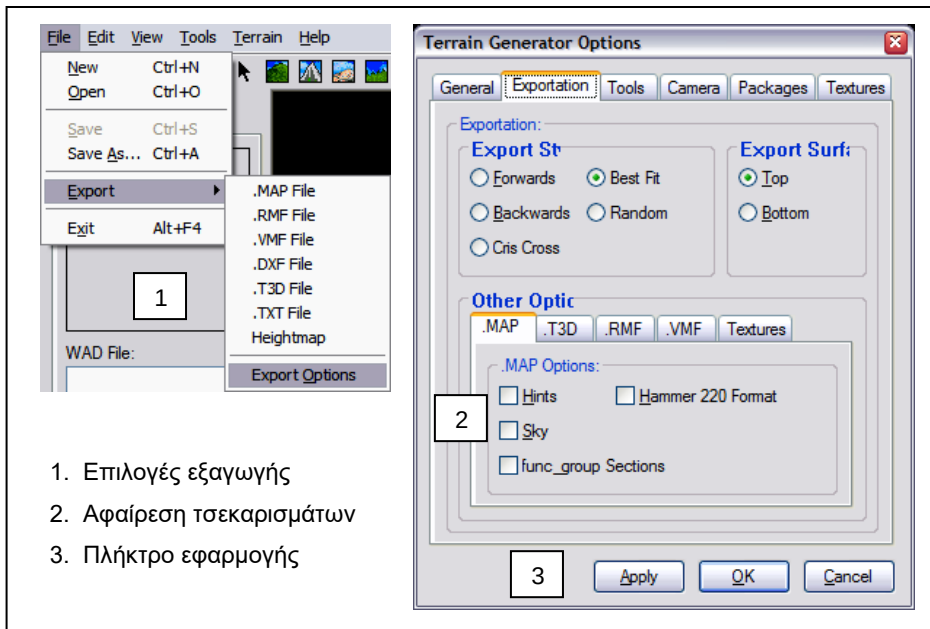
Αφού επιλέξουμε κάποιο από τα εργαλεία, κάνουμε κλικ, κρατάμε πατημένο το αριστερό πλήκτρο του ποντικιού και σέρνουμε πάνω ή κάτω, με εξαίρεση το εργαλείο εξομάλυνσης , στο οποίο αρκεί να κάνουμε κλικ. Μαζί με το δείκτη του ποντικιού εμφανίζεται και ένα πλέγμα, που μας δείχνει σε πόση έκταση θα εφαρμοστεί το εργαλείο που έχουμε επιλέξει (εικόνα 19.7).



Εικόνα 19.7 Πλέγμα που υποδηλώνει την έκταση εφαρμογής του εργαλείου

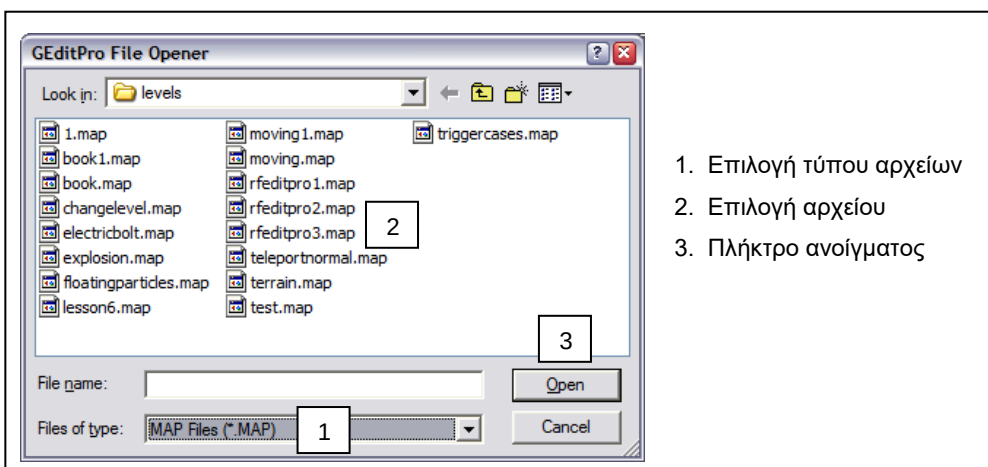
Αν επιθυμούμε την αύξηση ή τη μείωση της έκτασης που επηρεάζει το κάθε εργαλείο, κάνουμε κλικ στο πλήκτρο . Αν επιθυμούμε να υψώσουμε ή να βυθίσουμε μόνο ένα σημείο, χρησιμοποιούμε το εργαλείο . Τέλος, μπορούμε να κλειδώσουμε περιοχές ώστε να είναι αδύνατη η περαιτέρω αλλαγή τους με το πλήκτρο  (ξεκλειδώνουν με το πλήκτρο ).

Αφού ολοκληρώσουμε τη διαμόρφωση του χώρου, πρέπει να τον εξαγάγουμε σε μορφή .map, που αναγνωρίζεται από το Reality Factory. Πριν όμως κάνουμε την εξαγωγή, πρέπει να ρυθμίσουμε τις παραμέτρους της. Αυτό γίνεται από την επιλογή “File”, “Export”, “Export Options”, της γραμμής εργαλείων. Το μόνο που χρειάζεται να κάνουμε είναι να αφαιρέσουμε το τσεκάρισμα στις επιλογές Hints και Sky (εικόνα 19.8).



Εικόνα 19.8 Επιλογές εξαγωγής

Αφού κάνουμε τις παραπάνω ρυθμίσεις, εξάγουμε το ανάγλυφο σε μορφή .map (από την επιλογή “File”, “Export”, “.MAP FILE”), προσέχοντας να αποθηκεύσουμε το αρχείο στο φάκελο C:\RealityFactory\media\Levels. Το ανάγλυφο μπορούμε να το εισάγουμε σε οποιαδήποτε εφαρμογή μας στο RFEEditPro από τη γραμμή εντολών “File” και την επιλογή “Open”, προσέχοντας να επιλέξουμε τύπο αρχείων .map (εικόνα 19.9).



Εικόνα 19.9 Εισαγωγή ανάγλυφου στην εφαρμογή

Επειδή κατά την εισαγωγή του ανάγλυφου στην εφαρμογή μας αυτό αποτελείται από ανεξάρτητα τρίγωνα, η διαχείρισή του μπορεί να γίνει εξαιρετικά δύσκολη. Επίσης, κάθε φορά που τοποθετούμε κάποιο άλλο στοιχείο, το RFEEditPro επανασχεδιάζει όλη την εφαρμογή και ο μεγάλος αριθμός τριγώνων του ανάγλυφου μπορεί να προκαλέσει μεγάλη καθυστέρηση. Για να αποφύγουμε τα παραπάνω προβλήματα, μπορούμε να επιλέξουμε όλα τα τρίγωνα του ανάγλυφου και να τα μετατρέψουμε σε μοντέλο. Μπορούμε επίσης να ορίσουμε να μην επανασχεδιάζεται η εφαρμογή αφαιρώντας το τσεκάρισμα στην επιλογή Auto Rebuild BSP από τη γραμμή εντολών Mode.

Τέλος, μπορούμε να αναθέσουμε όποια υφή θέλουμε στο ανάγλυφό μας και για το λόγο αυτό καλό είναι να έχουμε συμπεριλάβει στο πακέτο υφών που χρησιμοποιούμε κάποια εικόνα εδάφους.

19.6 Διάλογοι

Το τελευταίο θέμα που θα μας απασχολήσει σ' αυτό το κεφάλαιο είναι η δυνατότητα να προσθέσουμε στην εφαρμογή μας διαλόγους με μοντέλα που τα χειρίζεται η οντότητα Pawn. Για το λόγο αυτό θα παρουσιάσουμε βήμα-βήμα ένα αρκετά σύνθετο παράδειγμα, το οποίο περιλαμβάνει στοιχεία από θέματα που παρουσιάσαμε σε προηγούμενα κεφάλαια. Η βασική ιδέα είναι η εξής: Σ' ένα δωμάτιο θα υπάρχει μία πόρτα που θα ανοίγει μόνο εάν ο χρήστης απαντήσει σωστά σε μία ερώτηση που θα του θέσει ένας actor. Για να ενεργοποιηθεί όμως η εμφάνιση της ερώτησης, ο χρήστης θα πρέπει να μαζέψει πρώτα έναν αριθμό αντικειμένων που θα είναι σκορπισμένα στο δωμάτιο.

Ας ξεκινήσουμε κατασκευάζοντας τον αρχικό χώρο και την πόρτα. Θα χρειαστούμε δύο δωμάτια, γιατί εάν ανοίξει η πόρτα το επίπεδο δε θα είναι κλειστό, με αποτέλεσμα να τερματιστεί απότομα η εφαρμογή. Δεν πρέπει να ξεχάσουμε να εισαγάγουμε τις οντότητες PlayerStart και PlayerSetup. Θα τοποθετήσουμε ένα στερεό μεταξύ των δύο δωματίων, που θα του δώσουμε την ιδιότητα Cut, για να ανοίξει ένα κενό μεταξύ των δύο δωματίων.

Στο κενό θα τοποθετήσουμε ένα ακόμα στερεό, που θα παίζει το ρόλο της πόρτας. Το στερεό αυτό θα το μετατρέψουμε σε μοντέλο με το όνομα doormodel και θα του δώσουμε κλειδιά κίνησης (ένα αρχικό για την κλειστή θέση και ένα δεύτερο στα 5 δευτερόλεπτα για την ανοιχτή). Τέλος, θα εισαγάγουμε μία οντότητα MovingPlatform που θα ελέγχει την κίνηση της πόρτας και μία Trigger που θα ενεργοποιεί τη MovingPlatform.

Οι παράμετροι στις οντότητες θα είναι:

Για τη σκανδάλη

Πεδίο	Τιμή
bNoCollide	TRUE
Model	-
szEntityName	triggerdoor
TimeOff	-
TimeOn	-

Για τη MovingPlatform

Πεδίο	Τιμή
bAutoStart	FALSE
bNoCollide	TRUE
bLooping	FALSE
bRunTimed	TRUE
Model	doormodel
Rideable	FALSE
TimeEachTrigger	1
TriggerName	triggerdoor

Με τις παραπάνω ρυθμίσεις, η σκανδάλη και η MovingPlatform δε θα ενεργοποιούνται με την επαφή του χρήστη με κάποιο μοντέλο. Η MovingPlatform δε θα ξεκινά αυτόματα και δε θα μπορεί ο χρήστης να ανέβει ή να κινηθεί μαζί με το μοντέλο που ελέγχει. Τέλος, επειδή η σκανδάλη και η MovingPlatform θα ελέγχονται προγραμματιστικά, δίνουμε την τιμή TRUE στο πεδίο bRunTimed της MovingPlatform, καθώς επίσης και την τιμή 1 στο πεδίο TimeEachTrigger.

Το επόμενο βήμα είναι να εισαγάγουμε τα αντικείμενα που θα είναι σκορπισμένα στο δωμάτιο. Αναφέραμε όμως ότι ο χρήστης θα πρέπει να μαζέψει ένα συγκεκριμένο αριθμό τέτοιων αντικειμένων ώστε να ενεργοποιηθεί η ερώτηση. Η εφαρμογή λοιπόν θα πρέπει με κάποιο τρόπο να “γνωρίζει” πόσα τέτοια αντικείμενα συγκέντρωσε ο χρήστης. Συνεπώς τα αντικείμενα θα πρέπει να μεταβάλλουν κάποια ιδιότητα του χρήστη και η οντότητα που μπορεί να το κάνει αυτό είναι η Attribute. Πριν όμως εισαγάγουμε έναν αριθμό από Attributes, θα πρέπει να ορίσουμε την ιδιότητα που θα μεταβάλλεται στο αρχείο player.ini. Το ανοίγουμε και προσθέτουμε στο τέλος του τις εξής γραμμές:

```
[boxcounter]
initial = 0
low = 0
high = 5
```

Ορίσαμε δηλαδή μία ιδιότητα που την ονομάζουμε boxcounter, με αρχική και ελάχιστη τιμή 0 και μέγιστη 5.

Καλό είναι να φαίνεται στην οθόνη πόσα αντικείμενα έχει συλλέξει ο χρήστης. Θα πρέπει λοιπόν να επεμβούμε και στο αρχείο hud.ini. Το ανοίγουμε και προσθέτουμε στο τέλος του τις εξής γραμμές:

```
[boxcounter]
type = numeric
framex = -170
framey = 10
font = 9
width = 1
active = modify
modifydirection = both
displaytime = 100000
```

```
[boxpicture]
type = horizontal
frame = hud\button4.bmp
framex = -100
framey = 10
active = true
```

Στην παράμετρο boxcounter (το όνομά της πρέπει να είναι ίδιο με το όνομα της ιδιότητας που ορίσαμε στο αρχείο Player.ini) ορίσαμε να εμφανίζεται ένας αριθμός σε συγκεκριμένη θέση στην οθόνη και με συγκεκριμένη γραμματοσειρά, ενώ με την παράμετρο boxpicture ορίσαμε να εμφανίζεται μία εικόνα αριστερά του αριθμού. Η εικόνα είναι προαιρετικό στοιχείο και θα πρέπει να είναι αποθηκευμένη στο φάκελο C:\RealityFactory\Media\Bitmaps ή σε κάποιον υπο-φάκελο αυτού. Να σημειώσουμε ότι οι θέσεις εμφάνισης τόσο του αριθμού όσο και της εικόνας βρίσκονται έπειτα από δοκιμές.

Ορίσαμε ένα εξαιρετικά μεγάλο χρόνο κατά τον οποίο ο αριθμός θα παραμένει στην οθόνη (displaytime = 100000), για να εξασφαλίσουμε ότι ο χρήστης θα βλέπει πόσα αντικείμενα έχει συγκεντρώσει όλο το χρονικό διάστημα που εκτελείται η εφαρμογή (αν δώναμε μικρό χρόνο, για παράδειγμα displaytime = 5, μετά από 5 δευτερόλεπτα ο αριθμός θα εξαφανιζόταν από την οθόνη). Τέλος, με τη δήλωση active = true, στην παράμετρο boxpicture, ορίζουμε να εμφανίζεται η εικόνα (με την τιμή false δεν εμφανίζεται).

Αφού αποθηκεύσουμε τις αλλαγές στο αρχείο hud.ini, μπορούμε να επανέλθουμε στην εφαρμογή μας και να εισαγάγουμε μία οντότητα Attribute, την οποία μετά την παραμε-

τροποποίησή της μπορούμε να την αντιγράψουμε και να την επικολλήσουμε άλλες 3-4 φορές. Οι παράμετροί της θα είναι οι εξής:

Πεδίο	Τιμή	Πεδίο	Τιμή
ActorRotation	0 180 0	Gravity	FALSE
AmbientColor	255 255 255	Scale	1
AmbientLight	TRUE	szActorFile	props\screen.act
AttributeAmount	1	szSoundFile	menu\menuclick.wav
AttributeName	boxcounter	UseProjectedShadows	TRUE
FillColor	255 255 255		

Οι πιο σημαντικές παράμετροι είναι η AttributeAmount=1, δηλαδή η αύξηση του μετρητή κατά μία μονάδα, η AttributeName=boxcounter, δηλαδή η ιδιότητα που θα μεταβάλλεται είναι η boxcounter (που την ορίσαμε στο αρχείο Player.ini) και η szActorFile=props\screen.act, δηλαδή ότι θα χρησιμοποιήσουμε τον actor screen.act.

Το επόμενο βήμα είναι να ορίσουμε μία οντότητα Pawn που θα ελέγχει έναν actor. Ανοίγουμε το αρχείο pawn.ini και προσθέτουμε τις παρακάτω γραμμές:

```
[doorcontrol]
fillcolor = 255 255 255
ambientcolor = 255 255 255
actorname=Virgil_red.act
actorrotation=-90 180 0
actorscale=1.4
subjecttogravity=false
boundingboxanimation=Aim1
projectedshadows=true
```

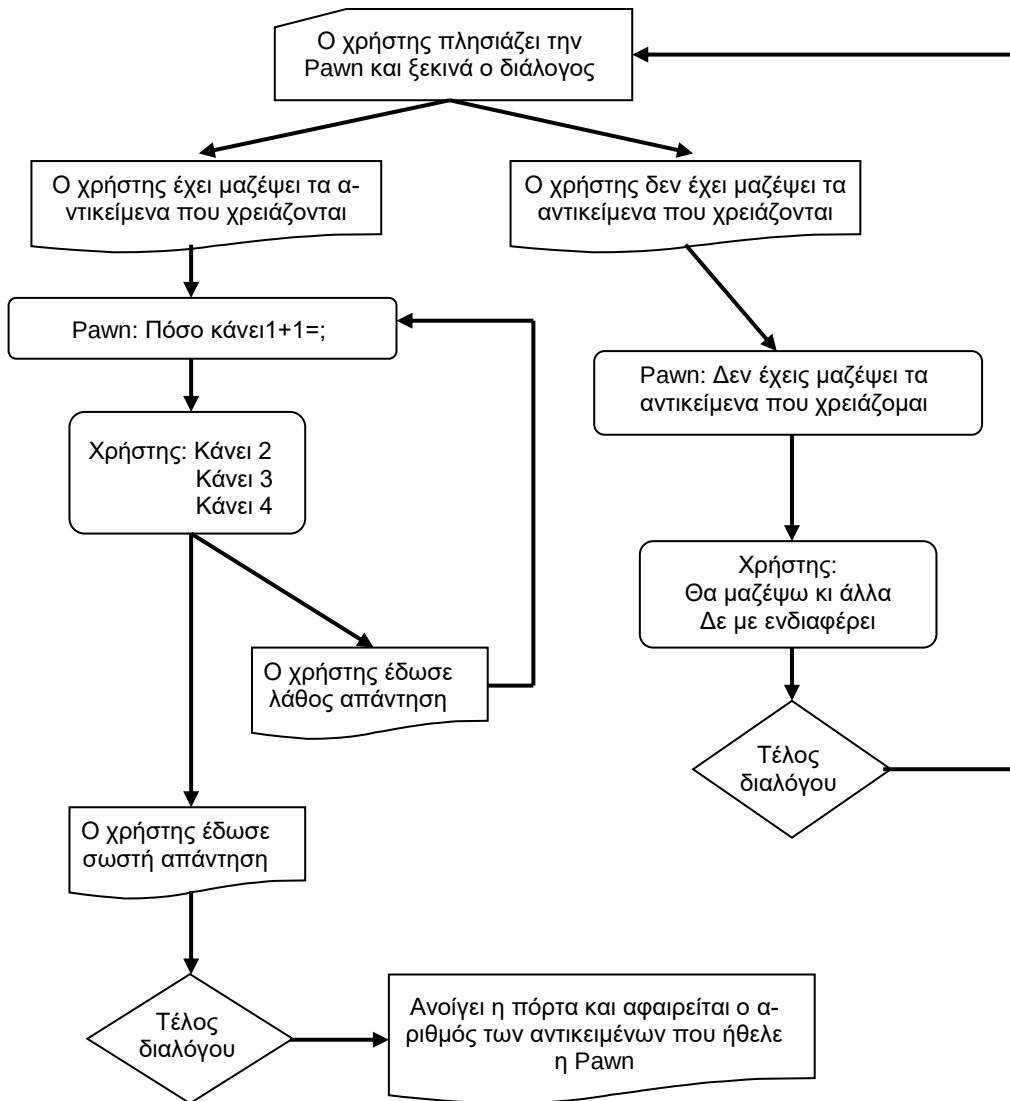
Ορίζουμε δηλαδή ένα νέο τύπο Pawn, που ονομάζεται doorcontrol, χρησιμοποιεί τον actor Virgil_red.act και ως αρχικό animation έχει το Aim1. Εισάγουμε στην εφαρμογή μία οντότητα Pawn, που την παραμετροποιούμε ως εξής:

Πεδίο	Τιμή	Πεδίο	Τιμή
Angle	0 0 0	SpawnOrder	-
ConvOrder	Start	SpawnPoint	-
ScriptName	Question.s	SpawnTrigger	-
PawnType	doorcontrol	szEntityName	-
ScriptName	nothing.s		

Να σημειώσουμε ότι δε χρειάζεται να τοποθετήσουμε Script Points, γιατί η οντότητα δε θα κινείται. Για τον ίδιο λόγο δε χρειάζεται να συμπληρώσουμε μία τιμή στο πεδίο SpawnPoint. Ούτε πρόκειται να ελέγξουμε την οντότητα μέσω κάποιου script. Έτσι, δε συμπληρώνουμε τιμή στο πεδίο SpawnOrder. Παρ' όλα αυτά, χρειάζεται να δώσουμε ένα όνομα script στο πεδίο ScriptName (nothing.s) για να εμφανιστεί ο actor στην εφαρμογή, αλλά δεν είναι απαραίτητο να υπάρχει αυτό το αρχείο (δημιουργείται ένα κενό αρχείο με το ίδιο όνομα κατά τη μεταγλώττιση της εφαρμογής).

Τέλος, πρέπει να εισαγάγουμε το όνομα του αρχείου δέσμης εντολών που θα ελέγχει το διάλογο μεταξύ της Pawn και του χρήστη (στο πεδίο ScriptName) και το όνομα της πρώτης εντολής που θα εκτελεστεί (στο πεδίο ConvnOrder). Για τις ανάγκες του παραδείγματός μας, το αρχείο δέσμης εντολών θα λέγεται Question.s και θα περιέχει ως πρώτη εντολή τη Start.

Επόμενη κίνησή μας είναι να εισαγάγουμε σ' ένα αρχείο το διάλογο που θα διαμειφθεί μεταξύ του χρήστη και της Pawn. Ας δούμε όμως πρώτα τη δομή του διαλόγου που θα χρησιμοποιήσουμε ως παράδειγμα:



Στην αρχή του διαλόγου υπάρχει μία βασική συνθήκη, εάν δηλαδή ο χρήστης έχει ή δεν έχει συλλέξει τον αριθμό των αντικειμένων που απαιτούνται (ας υποθέσουμε ότι θα χρειαστεί τρία κουτιά). Αν δεν έχει αυτό τον αριθμό, η Pawn θα τον προτρέψει να συλλέξει κι άλλα. Σε απάντηση, ο χρήστης θα έχει δύο επιλογές (“Θα μαζέψω κι άλλα” ή “Δε με ενδιαφέρει”). Και στις δύο περιπτώσεις οδηγούμαστε σε έξοδο από το διάλογο, είτε για να μαζέψει ο χρήστης και άλλα κουτιά είτε για να ακολουθήσει μία εντελώς διαφορετική πορεία στην εφαρμογή. Ο διάλογος όμως μπορεί να ξεκινήσει και πάλι, όποτε βέβαια το επιθυμήσει ο χρήστης.

Στην περίπτωση που ο χρήστης έχει ικανό αριθμό κουτιών (τρία ή και περισσότερα), η Pawn εμφανίζει την ερώτηση ("1+1="). Ο χρήστης έχει τρεις επιλογές, εκ των οποίων η μία είναι η σωστή. Αν επιλέξει λάθος απάντηση, η Pawn επανεμφανίζει την ερώτηση, έτσι ώστε ο χρήστης να επιλέξει και πάλι μία από τις πιθανές απαντήσεις. Αυτό συνεχίζεται μέχρι να επιλεγεί η σωστή απάντηση. Όταν συμβεί αυτό, έχουμε έξοδο από το διάλογο και άνοιγμα της κλειστής πόρτας. Επίσης, αφαιρείται ο αριθμός των κουτιών που χρειαζόταν η Pawn. Ο διάλογος έχει ολοκληρωθεί και δε χρειάζεται να ξεκινήσει πάλι, γιατί ο στόχος (το άνοιγμα της πόρτας) έχει επιτευχθεί.

Στο Reality Factory όλοι οι διάλογοι τοποθετούνται στο αρχείο Conversation.txt, που βρίσκεται στο φάκελο C:\RealityFactory\Install (όπως και τα μηνύματα που αποθηκεύονται στο αρχείο Message.txt). Το ανοίγουμε με το "Σημειωματάριο" και προσθέτουμε τις παρακάτω γραμμές:

[Has_boxes]

Oh! You have the boxes I need.

<CR>

Now let's see if you can answer my question.

<CR>

1+1=?

[Has_not_boxes]

I am sorry, but it seems you do not have enough boxes!

<CR>

I can open the door if you bring me at least three boxes.

[Try_find]

I will try to find them.

[Not_try_find]

I don't care.

[Answer1]

Two

[Answer2]

Three

[Answer3]

Four

[Right]

Well done!

<CR>

1+1=2

<CR>

Press Escape and the door will open!

[Wrong]

No!

<CR>

That is not the correct answer

<CR>

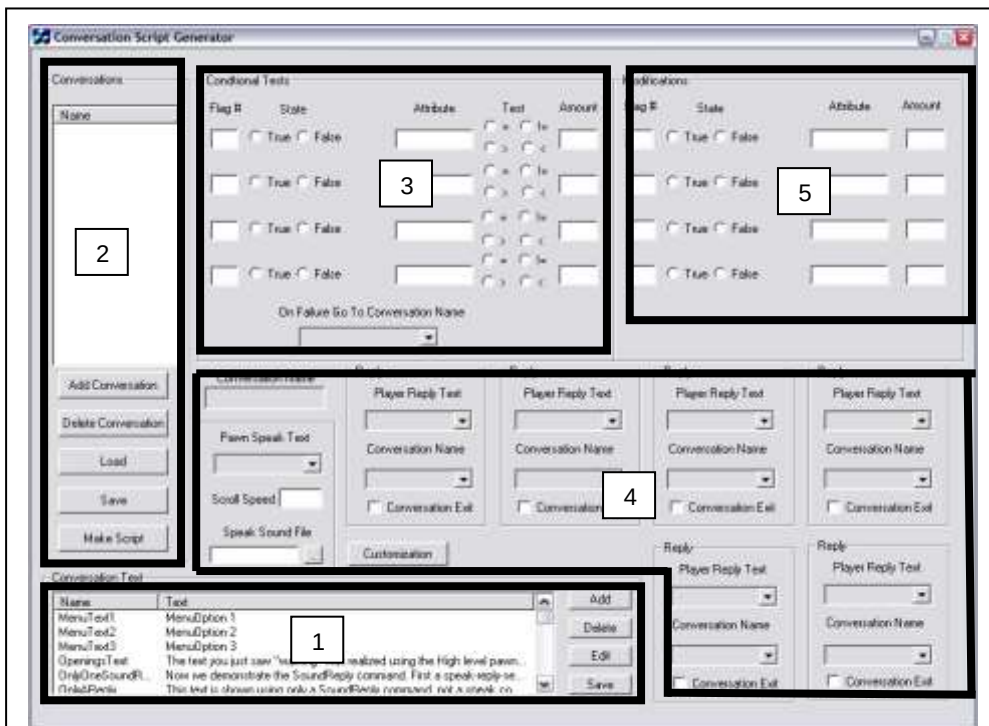
Try again.

Παρατηρούμε ότι, όπως και με τα μηνύματα στην οθόνη, έτσι και κάθε μέρος του διαλόγου (είτε απόκριση της Pawn είτε του χρήστη) έχει και από μία επικεφαλίδα (που περικλείεται μέσα σε αγκύλες). Όπως και με τα μηνύματα, δεν μπορούμε να χρησιμοποιήσουμε ελληνικούς χαρακτήρες.

Το τελευταίο μας βήμα είναι να κατασκευάσουμε το αρχείο Question.s, που θα ελέγχει το διάλογο. Αυτό θα μπορούσαμε να το κάνουμε χρησιμοποιώντας το “Σημειωματάριο”, όπως κάναμε και όταν συντάσσαμε το αρχείο δέσμης εντολών για την κίνηση της Pawn. Όμως στο Reality Factory περιέχεται εργαλείο για τη δημιουργία διαλόγων, το Conversation Script Builder, που εκτελείται κάνοντας διπλό κλικ στο αρχείο Tree-Builder.exe στο φάκελο C:\RealityFactory\Tools.

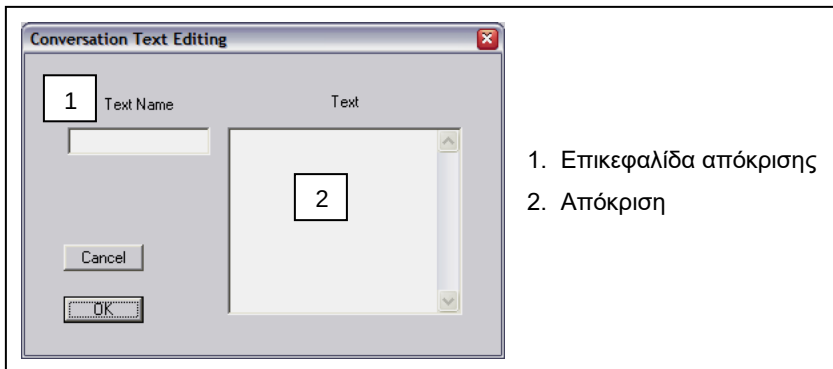
Η οθόνη του προγράμματος αποτελείται από πέντε μέρη (εικόνα 19.10). Στο πρώτο μέρος μπορούμε να εισαγάγουμε τις αποκρίσεις του διαλόγου. Δε θα το χρησιμοποιήσουμε, γιατί ήδη έχουμε ορίσει το διάλογο επεμβαίνοντας στο αρχείο Conversation.txt. Αν όμως θέλουμε, μπορούμε να εισαγάγουμε αποκρίσεις πατώντας το πλήκτρο “Προσθήκης” . Στην καρτέλα που εμφανίζεται, πληκτρολογούμε την επικεφαλίδα απόκρισης (χωρίς αγκύλες) και το κείμενο της απόκρισης (εικόνα 19.11). Επανερχόμενοι στην κυρίως οθόνη του προγράμματος, δεν παραλείπουμε να αποθηκεύσουμε τις αλλαγές, κάνοντας κλικ στο πλήκτρο “Αποθήκευσης” . Μπορούμε βέβαια να επεξεργαστούμε κάποια απόκριση με το πλήκτρο  ή και να τη διαγράψουμε με το πλήκτρο .

Στο δεύτερο μέρος καθορίζουμε τα τμήματα του διαλόγου. Με βάση το διάγραμμα ροής του διαλόγου που παρουσιάσαμε πιο πάνω, μπορούμε να διακρίνουμε τα εξής τμήματα: α) ο χρήστης έχει τον αριθμό των αντικειμένων που χρειάζονται και η Pawn δίνει την ερώτηση, β) ο χρήστης δεν έχει τον αριθμό των αντικειμένων που χρειάζονται και η Pawn τον προτρέπει να βρει και άλλα, γ) ο χρήστης δίνει τη σωστή απάντηση και τερματίζει ο διάλογος και δ) ο χρήστης δίνει λάθος απάντηση και επανεμφανίζονται οι πιθανές απαντήσεις. Ας ονομάσουμε τα τμήματα αυτά Start, Need_more_boxes, Right_answer, Wrong_answer. Για να εισαγάγουμε τα τμήματα του διαλόγου, χρησιμοποιούμε το πλήκτρο **Add Conversation**. Μπορούμε να διαγράψουμε κάποιο τμήμα με το πλήκτρο **Delete Conversation**.



1. Εισαγωγή αποκρίσεων Pawn και χρήστη
2. Καθορισμός τμημάτων του διαλόγου
3. Συνθήκες που πρέπει να ικανοποιούνται στο τμήμα του διαλόγου
4. Καθορισμός αποκρίσεων Pawn και χρήστη στο τμήμα του διαλόγου
5. Μεταβολές που προκαλεί το τμήμα του διαλόγου

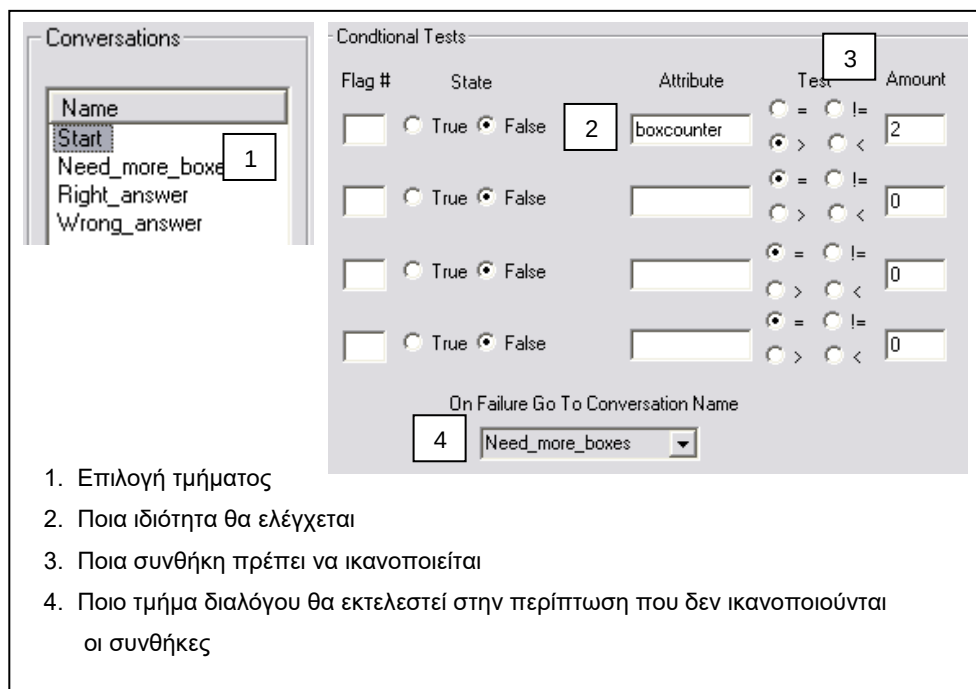
Εικόνα 19.10 Το πρόγραμμα Conversation Script Builder

Εικό-
19.11να
Καρ-

τέλα εισαγωγής αποκρίσεων

Δεν πρέπει να συγχέουμε τα τμήματα του διαλόγου με τις επικεφαλίδες αποκρίσεων. Ένα τμήμα διαλόγου μπορεί να περιέχει αρκετές επικεφαλίδες. Προς αποφυγή λαθών, καλό είναι να δίνουμε διαφορετικά ονόματα στα τμήματα και στις επικεφαλίδες.

Στο τρίτο τμήμα ορίζουμε τις συνθήκες κάτω από τις οποίες θα εκτελεστεί το τμήμα ενός διαλόγου. Για παράδειγμα, στο πρώτο τμήμα, για να δώσει η Ραση την ερώτηση στο χρήστη, θα πρέπει αυτός να έχει τουλάχιστον τρία κουτιά. Επιλέγουμε λοιπόν το τμήμα Start (που είναι το πρώτο) και στο χώρο που ορίζονται οι συνθήκες ζητούμε η ιδιότητα boxcounter (με την οποία ελέγχουμε τον αριθμό των αντικειμένων που συνέλεξε ο χρήστης) να έχει τιμή μεγαλύτερη του 2. Αν η ιδιότητα έχει τιμή ίση με 2 ή μικρότερη, τότε θα εκτελείται το τμήμα του διαλόγου με το όνομα Need_more_boxes, που το επιλέγουμε από το κυλιόμενο μενού (εικόνα 19.12).



Εικόνα 19.12 Καθορισμός εκτέλεσης τμήματος διαλόγου υπό συνθήκες

Στο τέταρτο τμήμα ορίζουμε ποιες επικεφαλίδες θα αποτελούν το τμήμα του διαλόγου που επεξεργαζόμαστε. Στο τμήμα Start, η Pawn “λέει” την επικεφαλίδα Has_boxes και ο χρήστης μπορεί να αποκριθεί με τις επικεφαλίδες Answer1, Answer2 και Answer3. Κάθε απόκριση οδηγεί σε άλλο τμήμα του διαλόγου (η Answer1 στο τμήμα Right_answer, ενώ οι Answer2 και Answer3 στο τμήμα Wrong_answer). Παρατηρούμε ότι ο χρήστης μπορεί να αποκριθεί με έως έξι διαφορετικές απαντήσεις, γιατί τόσα είναι τα πεδία που προβλέπει το πρόγραμμα. Κάθε απόκριση μπορεί να οδηγήσει σε έξοδο από το διάλογο αν τσεκάρουμε το πεδίο ☐ Conversation Exit ή/και σε άλλο τμήμα του διαλόγου, αν επιλέξουμε κάποιο στο πεδίο Conversation Name.

Μπορούμε επίσης να καθορίσουμε την ταχύτητα εμφάνισης του κειμένου που “λέει” η Pawn δίνοντας μία τιμή στο πεδίο Scroll Speed (μεγάλες τιμές συνεπάγονται μικρή ταχύτητα). Τέλος, μπορούμε να ορίσουμε ένα αρχείο ήχου που θα εκτελείται καθώς “μιλάει” η Pawn (τύπου .mp3 ή .wav) από το πεδίο Speak Sound File.

Η διαμόρφωση του κάθε τμήματος διαλόγου φαίνεται στην εικόνα 19.13.

The image displays a configuration window for a dialogue system, organized into four main sections corresponding to different dialogue states: **Start**, **Need_more_boxes**, **Right_answer**, and **Wrong_answer**. Each section contains the following elements:

- Conversation Name:** A text field containing the state name (e.g., 'Start', 'Need_more_boxes').
- Pawn Speak Text:** A dropdown menu for selecting the text to be spoken by the pawn.
- Scroll Speed:** A numeric input field, currently set to 50.
- Reply Section:**
 - Player Reply Text:** A dropdown menu for selecting the player's reply.
 - Conversation Name:** A dropdown menu for selecting the next state to transition to.
 - Conversation Exit:** A checkbox to determine if the dialogue should end.

For the **Start** state, the 'Conversation Name' dropdowns are set to 'Answer1', 'Answer2', and 'Answer3'. For **Need_more_boxes**, they are 'Try_find' and 'Not_try_find'. For **Right_answer**, they are empty. For **Wrong_answer**, they are 'Answer1', 'Answer2', and 'Answer3'. The 'Conversation Exit' checkbox is checked for 'Need_more_boxes' and 'Right_answer', and unchecked for 'Start' and 'Wrong_answer'.

Εικόνα 19.13 Καθορισμός αποκρίσεων των τμημάτων του διαλόγου

Στο τμήμα **Start** ο χρήστης μπορεί να αποκριθεί με μία από τις τρεις πιθανές απαντήσεις στην ερώτηση, που, εάν επιλέξει τη σωστή, οδηγείται στο τμήμα **Right_answer** και, εάν είναι η λανθασμένη, οδηγείται στο τμήμα **Wrong_answer**. Ο διάλογος δεν τερματίζεται, αλλά σε κάθε περίπτωση οδηγείται σε άλλο τμήμα του.

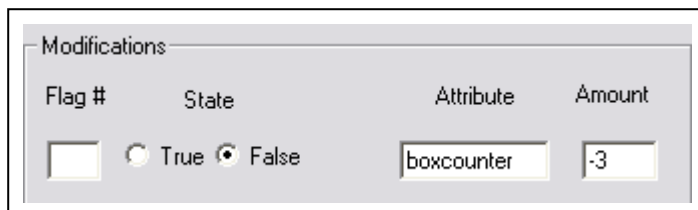
Με βάση το σχεδιασμό του διαλόγου, κάθε απόκριση του χρήστη στο τμήμα **Need_more_boxes** οδηγεί σε έξοδο από το διάλογο, αλλά το επόμενο τμήμα που θα εκτελεστεί, όταν ο διάλογος ξεκινήσει πάλι, θα είναι το **Start** (αν δεν ορίζαμε αυτό το τμήμα, όταν θα ξεκινούσε και πάλι ο διάλογος θα επανεμφανιζόταν το τμήμα **Ne-**

ed_more_boxes, ανεξάρτητα από το εάν ο χρήστης είχε μαζέψει εντωμεταξύ τον απαραίτητο αριθμό αντικειμένων).

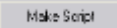
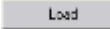
Στο τμήμα Right_answer δεν υπάρχει κάποια απόκριση από το χρήστη και έχουμε έξοδο από το διάλογο. Εφόσον δεν ορίζουμε ποιο τμήμα του διαλόγου θα εκτελεστεί αν ξεκινήσει εκ νέου ο διάλογος, αυτό σημαίνει ότι θα εμφανιστεί και πάλι το τμήμα Right_answer. Αυτό είναι λογικό, γιατί ο χρήστης ήδη έχει απαντήσει σωστά και θα δει πάλι το τμήμα εκείνο του διαλόγου που θα τον πληροφορεί ότι απάντησε σωστά.

Τέλος, στο τμήμα Wrong_answer έχουμε επανεμφάνιση των πιθανών απαντήσεων στην ερώτηση και διαμόρφωση ανάλογη με το τμήμα Start.

Μία τελευταία λεπτομέρεια που μένει να ρυθμίσουμε, είναι η αφαίρεση τριών αντικειμένων όταν ο χρήστης απαντήσει σωστά. Έχοντας επιλεγμένο το τμήμα Right_answer, στο κομμάτι της οθόνης του προγράμματος όπου καθορίζουμε τις μεταβολές που προκαλούνται, ορίζουμε η ιδιότητα boxcounter να μεταβληθεί κατά -3 (να αφαιρεθούν 3 μονάδες) (εικόνα 19.14).



Εικόνα 19.14 Καθορισμός μεταβολής που προκαλεί το τμήμα του διαλόγου

Το μόνο που απομένει είναι να παραγάγουμε το αρχείο δέσμης εντολών πατώντας το πλήκτρο . Αποθηκεύουμε το αρχείο στο φάκελο C:\RealityFactory\Scripts, δίνοντας το όνομα Question.s. Αν θέλουμε, μπορούμε να αποθηκεύσουμε την εργασία μας με το πρόγραμμα Conversation Script Builder πατώντας το πλήκτρο  και να την φορτώσουμε κάποια άλλη στιγμή με το πλήκτρο .

Το πρόγραμμα που χρησιμοποιήσαμε μπορεί να αξιοποιήσει μόνο ένα μικρό μέρος των εντολών που υπάρχουν διαθέσιμες για τη δημιουργία διαλόγων. Έτσι, δεν μπορούμε με αυτό να ορίσουμε ποια σκανδάλη θα ενεργοποιείται όταν ο χρήστης δώσει τη σωστή απάντηση, κάτι όμως που θέλουμε σύμφωνα με το παράδειγμά μας (για να ανοίξει η πόρτα). Πρέπει λοιπόν να ανοίξουμε με το “Σημειωματάριο” το αρχείο Question.s και να προσθέσουμε την παρακάτω γραμμή:


```
Right_answer[ ()  
{  
  ModifyAttribute("boxcounter",-3);  
  SetEventState("triggerdoor", true);  
  Speak("Right","");  
  Conversation(50);  
}]
```

Ο ενδιαφερόμενος μπορεί να αναζητήσει τον πλήρη κατάλογο των εντολών στην τεκμηρίωση του Reality Factory, που βρίσκεται στο φάκελο Docs.

Πριν από την κατασκευή των διαλόγων συνιστούμε την κατασκευή ενός διαγράμματος ροής, έτσι ώστε να φαίνεται: η πορεία του διαλόγου, ποιο τμήμα του θα εκτελείται κατά περίπτωση, σε ποιο τμήμα θα οδηγείται ο διάλογος, πότε θα τερματίζεται και ποιες αποκρίσεις θα περιλαμβάνονται σε κάθε τμήμα. Είναι εύκολο να γίνουν λάθη που θα έχουν απροσδόκητα αποτελέσματα αν δεν υπάρχει σωστή οργάνωση του διαλόγου.

Για να ξεκινήσει ο διάλογος, ο χρήστης πρέπει να πλησιάσει τον actor που ελέγχεται από την Pawn και να πατήσει το πλήκτρο "χρήσης".

Κεφάλαιο 20^ο

Διανομή της εφαρμογής στους τελικούς χρήστες

Το τελευταίο βήμα στην κατασκευή μίας εφαρμογής είναι η συγκρότηση ενός πακέτου διανομής της. Πρέπει δηλαδή να συγκεντρώσουμε όλα τα αρχεία που απαρτίζουν την εφαρμογή και είτε να τα αντιγράψουμε ως έχουν σ' ένα CD ή DVD ανάλογα με το μέγεθός τους είτε να τα συμπίεσουμε σ' ένα και μόνο αρχείο και μετά να το εγγράψουμε σε CD ή DVD. Ο τελικός χρήστης, ανάλογα με τη μέθοδο που ακολουθήσαμε, θα μπορεί να αντιγράψει ή να αποσυμπίεσει τα αρχεία σ' ένα δικό του φάκελο.

Να σημειώσουμε ότι στο Reality Factory, αν και θα παρουσιάσουμε τη διαδικασία στη συνέχεια του κεφαλαίου, δεν είναι απαραίτητη η χρήση εξειδικευμένου λογισμικού δημιουργίας αρχείων εγκατάστασης (*Installers*). Αυτό συμβαίνει επειδή δε χρειάζεται να γίνουν δηλώσεις στο μητρώο των Windows, ούτε να εγγραφούν αρχεία σε άλλους φακέλους πλην αυτού που τοποθετούμε την εφαρμογή μας. Αν βέβαια επιθυμούμε να δημιουργηθούν εικονίδια στην επιφάνεια εργασίας ή στην “Εναρξη”, “Όλα τα προγράμματα”, τότε θα πρέπει να χρησιμοποιήσουμε κάποιο πρόγραμμα εγκατάστασης.

20.1 Προετοιμασία των αρχείων της εφαρμογής

Το πρώτο μας βήμα –και το πιο σημαντικό– είναι να βεβαιωθούμε ότι η εφαρμογή μας είναι απ' όλες τις απόψεις έτοιμη. Πριν κάνουμε οτιδήποτε άλλο, πρέπει να γίνουν εξαντλητικές δοκιμές κάθε σημείου της και από λειτουργική και από αισθητική άποψη. Ένα συνηθισμένο λάθος που γίνεται είναι ότι ο σχεδιαστής, επειδή ακριβώς γνωρίζει πολύ καλά την εφαρμογή του, κατά τη δοκιμή της κάνει τις αναμενόμενες ενέργειες. Ο τελικός χρήστης όμως είναι πιθανό να μην ακολουθήσει την επιθυμητή διαδρομή, να μην ακολουθήσει την προβλεπόμενη σειρά ενεργειών ή, ακόμα, να κάνει πράγματα εντελώς αντίθετα απ' αυτά που περιμένουμε. Πρέπει λοιπόν ο σχεδιαστής να έχει προβλέψει τέτοιες καταστάσεις και να έχει δώσει λύσεις. Για να δώσουμε ένα παράδειγμα, πρέπει να έχουμε προβλέψει τα όρια του εικονικού κόσμου να είναι δυσπρόσιτα στο χρήστη έχοντας τοποθετήσει ανυπέρβλητα εμπόδια. Σε αντίθετη περίπτωση ο χρήστης μπορεί να βρεθεί στον εξωτερικό κύβο που περιβάλλει το επίπεδό μας, χωρίς τη δυνατότητα να επανέλθει.

Και η αισθητική αρτιότητα παίζει σημαντικό ρόλο. Ο χρήστης σχηματίζει πολύ άσχημη άποψη για την εφαρμογή μας, αν για παράδειγμα παρατηρήσει ότι μερικά δένδρα δεν εφάπτονται στο έδαφος, αλλά αιωρούνται. Μία συνηθισμένη πρακτική που ακολουθούμε είναι να δώσουμε την τελική εκδοχή της εφαρμογής μας σε μία ομάδα έμπειρων ή/και άπειρων χρηστών που ο ρόλος τους είναι να κάνουν τέτοιου είδους παρατηρήσεις. Έτσι, όποια και όσα προβλήματα διαπιστωθούν, διορθώνονται πριν από την κανονική διανομή της εφαρμογής.

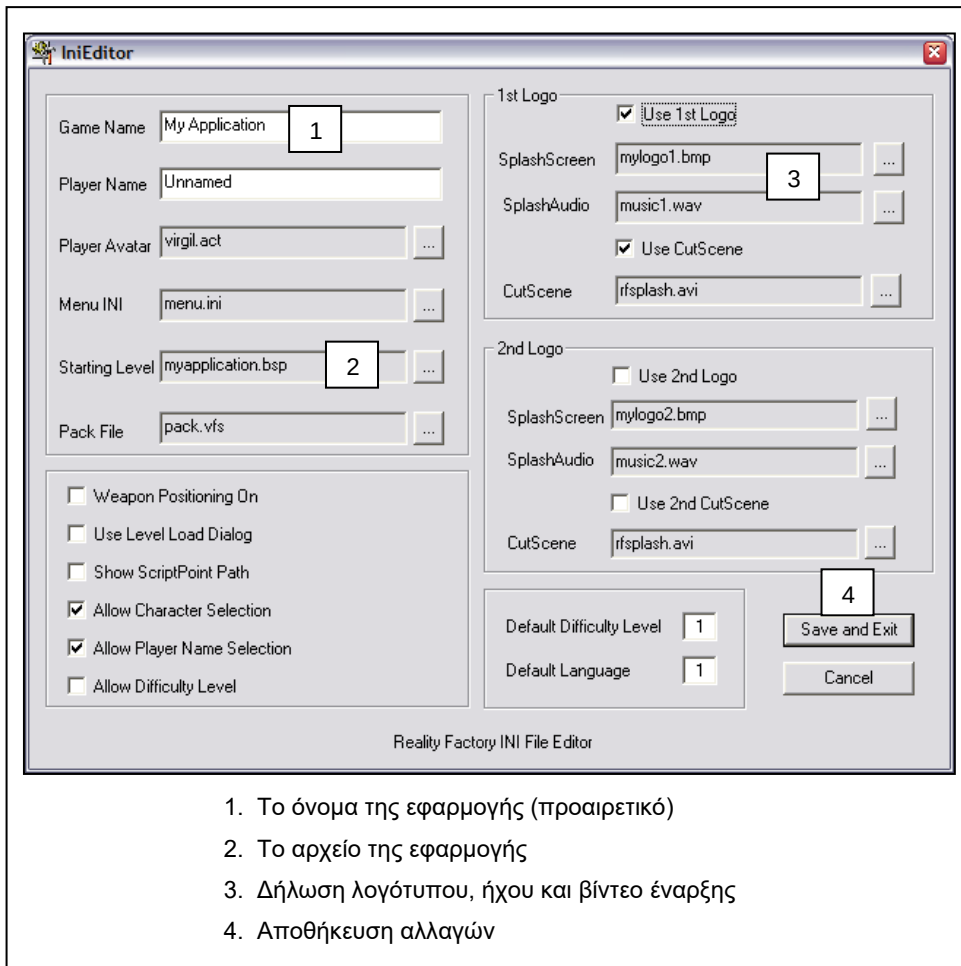
Υποθέτοντας ότι έχει ολοκληρωθεί η παραπάνω διαδικασία, είμαστε έτοιμοι για τη δημιουργία του πακέτου διανομής. Υπάρχουν διαφορετικές μέθοδοι για να το κάνουμε αυτό, εμείς όμως θα περιγράψουμε την πιο απλή αλλά και πιο αποτελεσματική.

Η πρώτη μας κίνηση είναι να ελέγξουμε τα ονόματα των αρχείων που έχουν δημιουργηθεί από το RFEEditPro. Αυτά πρέπει να βρίσκονται στο φάκελο C:\RealityFactory\Media\Levels. Αν υποθέσουμε ότι είχαμε αποθηκεύσει αρχικά την εφαρμογή μας με το όνομα myapplication, τότε στο φάκελο αυτό είναι σίγουρο ότι υπάρχουν τα αρχεία myapplication.bsp, myapplication.3dt και myapplication.map. Αν δεν υπάρχει το τελευταίο αρχείο, αυτό οφείλεται στο ότι κατά τη μεταγλώττιση της εφαρμογής μας δεν είχαμε δώσει το ανάλογο όνομα για τη δημιουργία του αρχείου .map (εικόνα 11.25 κεφάλαιο 11.2). Πολύ απλά μπορούμε να μεταγλωττίσουμε εκ νέου την εφαρμογή, δίνοντας το σωστό όνομα.

Επίσης, στο φάκελο C:\RealityFactory\Media\Levels πιθανώς να υπάρχουν και άλλα αρχεία με το όνομα myapplication. Ένα απ' αυτά μπορεί να είναι το myapplication.txl, που είναι το πακέτο υφών που χρησιμοποιούμε στην εφαρμογή μας (αν είχαμε φροντίσει να το ονομάσουμε έτσι και να το έχουμε δηλώσει στο RFEEditPro, *εικόνα 11.9* κεφάλαιο 11.1.3). Αν το αρχείο με το πακέτο υφών έχει διαφορετικό όνομα, δε χρειάζεται να κάνουμε κάποια ενέργεια, αρκεί να θυμόμαστε ποιο είναι (θα μας χρειαστεί σε επόμενο βήμα). Άλλα αρχεία με το ίδιο όνομα μπορεί να είναι το myapplication.GPF και το myapplication.REC (που δημιουργείται όταν επιλέξουμε φωτισμό ακτινοβολίας και είναι ιδιαίτερα μεγάλο σε μέγεθος).

Το δεύτερο βήμα είναι να δημιουργήσουμε ένα φάκελο στο σκληρό μας δίσκο, για παράδειγμα C:\MyApplication. Στο φάκελο αντιγράφουμε όλα ανεξαιρέτως τα αρχεία, φακέλους και υπο-φακέλους του Reality Factory (από την “Εξερεύνηση των Windows” μεταφερόμαστε στο φάκελο C:\RealityFactory, επιλέγουμε όλα τα περιεχόμενά του, τα αντιγράφουμε, μεταφερόμαστε στο φάκελο C:\MyApplication και επικολλούμε. Στην ουσία δημιουργούμε ένα δεύτερο αντίγραφο του Reality Factory.

Το τρίτο βήμα είναι να εκτελέσουμε το πρόγραμμα IniEditor (αρχείο IniEditor.exe), όχι πλέον από την αρχική εγκατάσταση, αλλά από το φάκελο C:\MyApplication\Tools. Η λειτουργία του προγράμματος αναλύθηκε στο κεφάλαιο 8.4.2, όμως στο σημείο αυτό παρουσιάζονται οι ρυθμίσεις με βάση την εφαρμογή που έχουμε κατασκευάσει (εικόνα 20.1).



Εικόνα 20.1 Ρυθμίσεις στο IniEditor

Η πιο βασική δήλωση είναι το όνομα του αρχείου .bsp, το οποίο είναι η μεταγλωττισμένη μορφή της εφαρμογής μας. Επίσης, αν έχουμε κατασκευάσει κάποιο λογότυπο, μπορούμε να το δηλώσουμε στο αντίστοιχο πεδίο, φροντίζοντας να τσεκάρουμε το τετραγωνάκι Use 1st Logo. Το ίδιο ισχύει και αν επιθυμούμε να παιχτεί και κάποιο εισαγωγικό βίντεο.

Το τέταρτο βήμα είναι να τροποποιήσουμε το αρχείο Character.ini, που βρίσκεται στο φάκελο C:\MyApplication\Install. Το ανοίγουμε με το “Σημειωματάριο” και παρατηρούμε ότι υπάρχουν δηλώσεις για τους τρεις διαθέσιμους χαρακτήρες του προγράμματος. Σε καθέναν απ’ αυτούς υπάρχει η γραμμή startlevel = “κάποιο αρχείο”.bsp. Η γραμμή αυτή καθορίζει από ποιο επίπεδο θα ξεκινά ο κάθε χαρακτήρας. Αν θέλουμε και στους

τρεις να ισχύει η ρύθμιση που κάναμε με τον IniEditor (StartingLevel), πρέπει ή να βάλουμε ένα ερωτηματικό (;) μπροστά από τις τρεις γραμμές για να αγνοούνται οι εντολές ή να σβήσουμε το όνομα του αρχείου αφήνοντας μόνο startlevel = . Αν θέλουμε ο κάθε χαρακτήρας να ξεκινά σε διαφορετικό επίπεδο, παρακάμπτοντας τη ρύθμιση στον IniEditor, πρέπει να δηλώσουμε το όνομα κάποιου .bsp αρχείου. Σ' αυτή την περίπτωση όμως θα πρέπει να έχουμε περιλάβει και το ανάλογο αρχείο στο πακέτο της εφαρμογής μας.

Η τελευταία μας ενέργεια αφορά τη διαγραφή των περιττών φακέλων και αρχείων για να μειωθεί το συνολικό μέγεθος της εφαρμογής. Μπορούμε να διαγράψουμε το φάκελο C:\MyApplication\Docs και το φάκελο C:\MyApplication\Tools. Μπορούμε επίσης να διαγράψουμε τα αρχεία με κατάληξη .log από το φάκελο C:\MyApplication (θα δημιουργηθούν ξανά μόνα τους στον υπολογιστή του τελικού χρήστη).

Στο φάκελο C:\MyApplication\Media\Levels διαγράψουμε όλα τα αρχεία που δεν αφορούν την εφαρμογή, χρειάζεται όμως προσοχή μην τυχόν διαγράψουμε και το πακέτο υφών που χρησιμοποιήσαμε (ειδικά αν του έχουμε δώσει άλλο όνομα). Αν επιθυμούμε ο τελικός χρήστης να μην μπορεί να επεξεργαστεί την εφαρμογή μας, μπορούμε να διαγράψουμε και τα αρχεία myapplication.3dt και myapplication.map.

Από τους φακέλους C:\MyApplication\Media\Actors και C:\MyApplication\Media\Audio μπορούμε να διαγράψουμε όσους actors και ήχους δεν έχουμε χρησιμοποιήσει στην εφαρμογή. Δε διαγράψουμε όμως τους υπο-φακέλους που περιέχουν. Η διαγραφή των αρχείων στους φακέλους Actors και Audio πρέπει να γίνει με ιδιαίτερη προσοχή και αφού είμαστε απόλυτα σίγουροι ότι πράγματι δε χρησιμοποιούνται πια. Για παράδειγμα, είναι πολύ εύκολο να θεωρήσουμε ότι το αρχείο Die1.wav δε μας χρειάζεται, ενώ στην πραγματικότητα συνδέεται αυτόματα με το "θάνατο" του χρήστη. Για όποιο αρχείο δεν είμαστε σίγουροι, καλύτερα να μην το διαγράψουμε.

Τέλος, καλό είναι μετά τη διαγραφή των αρχείων να εκτελέσουμε την εφαρμογή για έναν τελικό έλεγχο. Η εκτέλεσή της πρέπει να γίνει κάνοντας διπλό κλικ στο αρχείο RealityFactory.exe στο φάκελο C:\MyApplication, αφού τα περιεχόμενα αυτού του φακέλου θα αποτελέσουν το πακέτο διανομής στους χρήστες.

Με τις ρυθμίσεις που κάναμε πιο πάνω, στην ουσία αναγκάζουμε το Reality Factory να εκτελεί μόνο την εφαρμογή μας. Το αρχείο που εκκινεί την εφαρμογή είναι το RealityFactory.exe.

20.2 Δημιουργία αρχείου εγκατάστασης

Ο πιο απλός τρόπος για να διανεύουμε την εφαρμογή είναι να εγγράψουμε όλα τα περιεχόμενα του φακέλου C:\MyApplication σ' ένα CD ή σ' ένα DVD ανάλογα με το συνολικό μέγεθος, αν και πολύ δύσκολα μπορούμε να ξεπεράσουμε τη χωρητικότητα ενός CD. Ως τρόπος όμως είναι αρκετά “άκομψος”, γιατί ο τελικός χρήστης πρέπει να αντιγράψει και να επικολλήσει όλα τα αρχεία του CD σε δικό του φάκελο.

Ένας περισσότερο κομψός τρόπος είναι να χρησιμοποιήσουμε ένα από τα δύο πολύ δημοφιλή προγράμματα συμπίεσης, το WinZip ή το WinRar. Τα πλεονεκτήματα αυτής της προσέγγισης είναι ότι:

- Μειώνεται σημαντικά το μέγεθος των αρχείων.
- Έχουμε ένα και μόνο αρχείο που, όταν το αποσυμπιέσουμε, δημιουργεί την αρχική δομή αρχείων, φακέλων και υπο-φακέλων.
- Μπορούμε να δημιουργήσουμε ένα αυτο-αποσυμπιεζόμενο αρχείο, έτσι ώστε να μην είναι απαραίτητο ο τελικός χρήστης να έχει εγκατεστημένο στο σύστημά του το WinZip ή το WinRar.
- Δίνεται η επιλογή στο χρήστη να διαλέξει διαφορετικό φάκελο αποσυμπίεσης απ' αυτόν που προτείνεται.

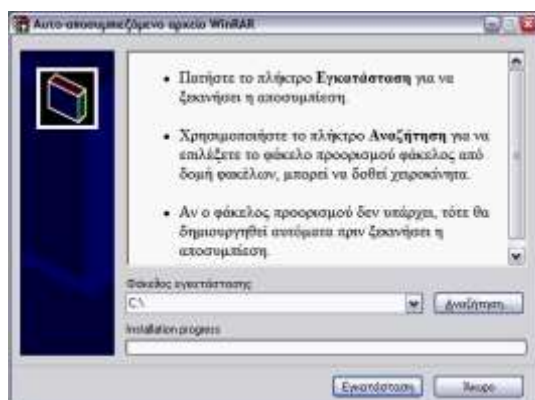
Για τις ανάγκες του παραδείγματός μας, θα παρουσιάσουμε τη διαδικασία με τη χρήση του WinRar, δοκιμαστική έκδοση του οποίου είναι διαθέσιμη στη διεύθυνση <http://www.winrar.com/download.html>. Αφού το εγκαταστήσουμε, το εκκινούμε από την “Εναρξη”, “Όλα τα προγράμματα”. Στη γραμμή εργαλείων κάνουμε κλικ στο εικονίδιο Οδηγός .

Στην καρτέλα που ανοίγει, κάνουμε την επιλογή “Δημιουργία νέου αρχείου συμπίεσης” και πατάμε το πλήκτρο “Next”. Στη νέα καρτέλα αναζητούμε και επιλέγουμε όλο το φάκελο C:\MyApplication και πατάμε το πλήκτρο “OK”. Στην επόμενη καρτέλα δίνουμε το όνομα που θέλουμε να έχει το συμπιεσμένο αρχείο και πατάμε το πλήκτρο “Next”. Στην τελευταία καρτέλα προσέχουμε να κάνουμε την επιλογή “Δημιουργία αυτο-αποσυμπιεζόμενου αρχείου” και πατάμε το πλήκτρο “Finish”. Αν δεν έχουμε κάνει καμία ρύθμιση στο WinRar, το αρχείο θα δημιουργηθεί στην Επιφάνεια Εργασίας και μπορούμε πλέον να το εγγράψουμε σ' ένα CD (εικόνα 20.2).

Ο τελικός χρήστης το μόνο που χρειάζεται είναι να κάνει διπλό κλικ στο αρχείο που θα περιέχεται στο CD και να διαλέξει σε ποιο σκληρό δίσκο θα αποσυμπίεστούν τα αρχεία, για παράδειγμα C:\ (εικόνα 20.3). Μπορεί να διαγράψει το φάκελο που προτείνεται, γιατί κατά την αποσυμπίεση δημιουργείται αυτόματα φάκελος που έχει το ίδιο όνομα με αυτόν όπου έγινε η συμπίεση.



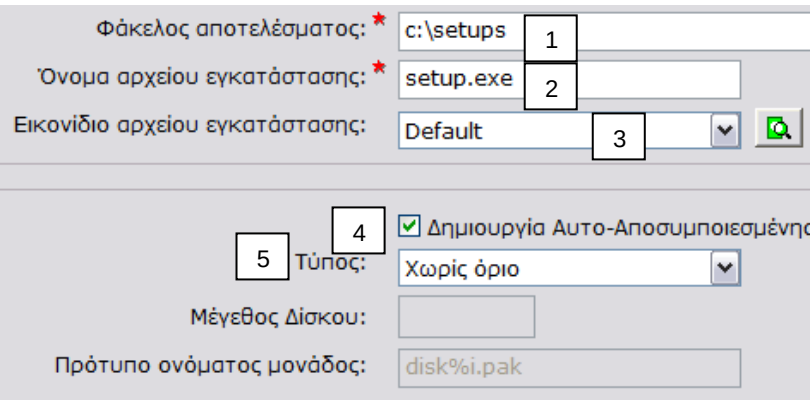
Εικόνα 20.2 Τα βήματα συμπίεσης των αρχείων με το WinRAR



Εικόνα 20.3 Επιλογή σκληρού δίσκου αποσυμπίεσης

Η μέθοδος που περιγράφηκε πιο πάνω, όπως και η πρώτη, έχουν το εξής αρνητικό σημείο. Δε δημιουργούν συντομεύσεις ούτε στην επιφάνεια εργασίας ούτε στην “Έναρξη”, “Όλα τα προγράμματα”. Αυτό αναγκάζει το χρήστη είτε να τις φτιάξει μόνος του είτε να εκκινεί κάθε φορά την εφαρμογή από την “Εξερεύνηση των Windows”. Ο τελευταίος λοιπόν τρόπος που θα περιγράψουμε περιλαμβάνει και τη δημιουργία των αντίστοιχων συντομεύσεων και εικονιδίων. Για το σκοπό αυτό θα χρησιμοποιήσουμε ένα πρόγραμμα δημιουργίας αρχείων εγκατάστασης (*Installer*). Επιλέξαμε το CreateInstall Free, επειδή διατίθεται δωρεάν στη διεύθυνση <http://www.createinstall.com/cifree/cifree.exe> και η χρήση του είναι ιδιαίτερα απλή.

Αφού το εγκαταστήσουμε, το εκκινούμε από την “Έναρξη”, “Όλα τα προγράμματα” και ξεκινάμε ένα νέο έργο από το εικονίδιο . Στα αριστερά της οθόνης μας υπάρχει μία σειρά από εικονίδια, που το καθένα ενεργοποιεί και από μία καρτέλα ρυθμίσεων. Στην πρώτη, που εμφανίζεται με την έναρξη του προγράμματος, δίνουμε τη διαδρομή αποθήκευσης του αρχείου εγκατάστασης, το όνομά του, το εικονίδιο που θα χρησιμοποιεί, το αν θα είναι αυτο-αποσυμπιεζόμενο και αν θα υπάρχει περιορισμός στο μέγεθός του (εικόνα 20.4).



Φάκελος αποτελέσματος: * c:\setups 1

Όνομα αρχείου εγκατάστασης: * setup.exe 2

Εικονίδιο αρχείου εγκατάστασης: Default 3

4 ☒ Δημιουργία Αυτο-Αποσυμπίεσμένης

5 Τύπος: Χωρίς όριο

Μέγεθος Δίσκου:

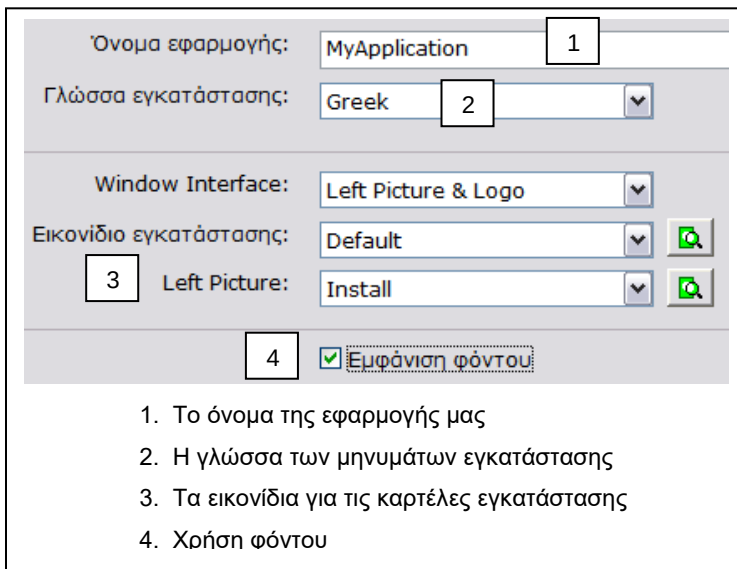
Πρότυπο ονόματος μονάδος: disk%i.pak

1. Φάκελος αποθήκευσης αρχείου εγκατάστασης
2. Όνομα του αρχείου εγκατάστασης
3. Εικονίδιο εγκατάστασης
4. Αυτο-αποσυμπιεζόμενο αρχείο
5. Τύπος αρχείου

Εικό-
20.4
μί-
αρ-
εγκα-
στα-

να
Ρυθ-
σεις
χείου
τά-
σης

Στη δεύτερη καρτέλα, που εμφανίζεται κάνοντας κλικ στο εικονίδιο γενικών ρυθμίσεων , διαμορφώνουμε την εμφάνιση που θα έχουν οι καρτέλες εγκατάστασης, τη γλώσσα που θα χρησιμοποιείται, το φόντο και τα εικονίδια (μπορούμε να επιλέξουμε διαφορετικά κάνοντας κλικ στα εικονίδια ) (εικόνα 20.5).



Όνομα εφαρμογής: MyApplication 1

Γλώσσα εγκατάστασης: Greek 2

Window Interface: Left Picture & Logo

Εικονίδιο εγκατάστασης: Default 3

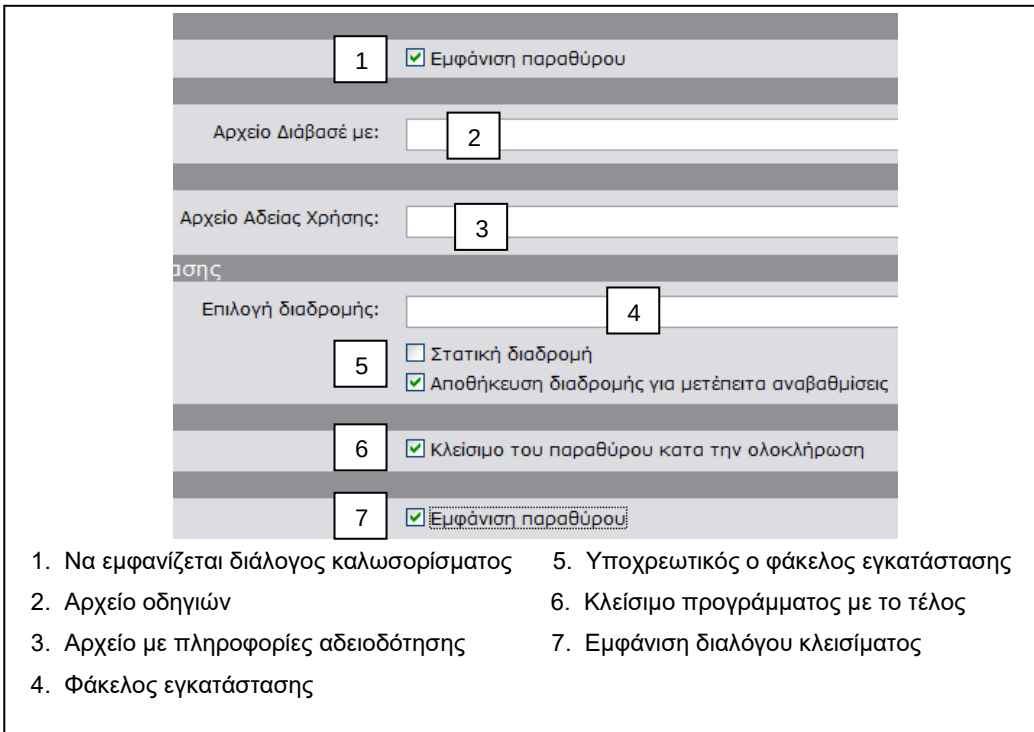
Left Picture: Install 3

Εμφάνιση φόντου: 4 ☒

1. Το όνομα της εφαρμογής μας
2. Η γλώσσα των μηνυμάτων εγκατάστασης
3. Τα εικονίδια για τις καρτέλες εγκατάστασης
4. Χρήση φόντου

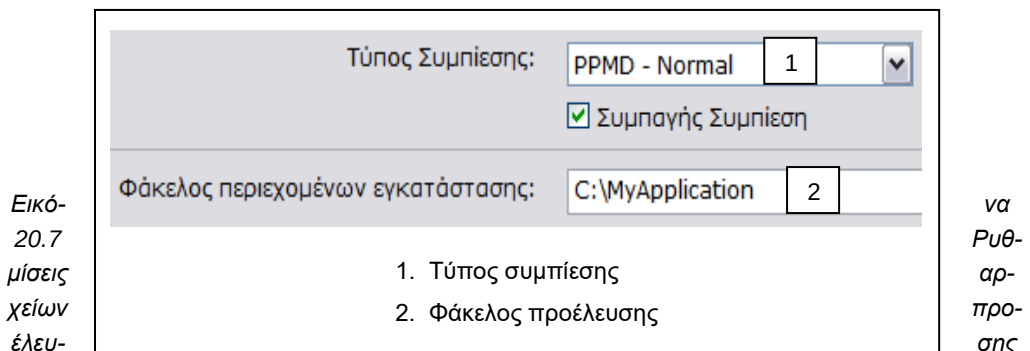
Εικόνα 20.5 Γενικές ρυθμίσεις

Στην καρτέλα παραθύρων, από το εικονίδιο , η βασική ρύθμιση είναι ο φάκελος εγκατάστασης και αν θα είναι αναγκαστική η εγκατάσταση σ' αυτόν ή θα επιτρέπεται στο χρήστη να διαλέξει κάποιον άλλο. Αν αφήσουμε τη διαδρομή του φακέλου εγκατάστασης κενή, τότε θα προτείνεται η εφαρμογή μας να εγκατασταθεί στο φάκελο C:\Program Files\Όνομα Εφαρμογής. Επίσης, μπορούμε να ορίσουμε αν θα υπάρχουν –και ποια– αρχεία με γενικές πληροφορίες και με πληροφορίες για την άδεια χρήσης, αν θα κλείνει το πρόγραμμα εγκατάστασης αυτόματα και αν θα εμφανίζεται διάλογος κλεισίματος της εγκατάστασης (εικόνα 20.6).



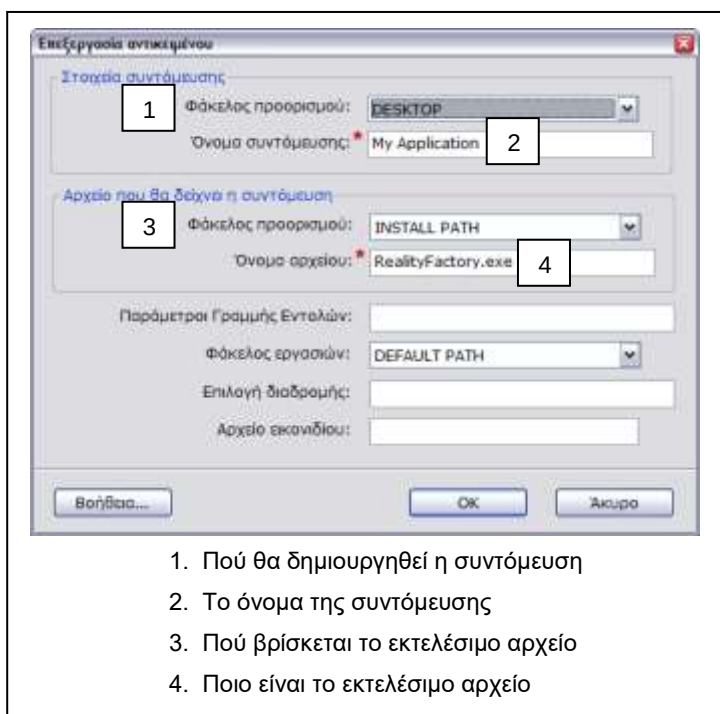
Εικόνα 20.6 Ρυθμίσεις διαλόγων

Στην καρτέλα αρχείων , επιλέγουμε ποια είναι τα αρχεία που θέλουμε να περιληφθούν στην εγκατάσταση. Στην περίπτωση μας είναι ο φάκελος C:\MyApplication, με όλα τα αρχεία και τους υπο-φακέλους του. Επιλέγουμε επίσης τον τύπο/βαθμό συμπίεσης. Τέλος, μπορούμε να συμπεριλάβουμε και άλλα αρχεία εκτός από τον κύριο φάκελο προέλευσης, πράγμα που δεν ισχύει στην περίπτωση μας (εικόνα 20.7).



Στην καρτέλα συντομεύσεων , ορίζουμε τη δημιουργία συντομεύσεων στην επιφάνεια εργασίας και στην “Εναρξη”, “Όλα τα προγράμματα”. Κατ’ αρχάς πρέπει να διαγράψουμε ό,τι δηλώσεις υπάρχουν ως παράδειγμα, επιλέγοντάς τες και πατώντας το πλήκτρο διαγραφής .

Στη συνέχεια, κάνουμε κλικ στο πλήκτρο δημιουργίας νέας εγγραφής  και στη νέα καρτέλα ορίζουμε αν η συντόμευση θα δημιουργηθεί στην επιφάνεια εργασίας ή στην “Εναρξη”, “Όλα τα προγράμματα”, ποιο θα είναι το όνομά της, σε ποιο φάκελο θα αναζητηθεί το εκτελέσιμο αρχείο και ποιο είναι αυτό. Τα εικονίδια δημιουργούνται αυτόματα, γιατί περιέχονται μέσα στο εκτελέσιμο αρχείο (εικόνα 20.8). Επαναλαμβάνουμε τη διαδικασία εάν επιθυμούμε τη δημιουργία συντομεύσεων και στην επιφάνεια εργασίας και στην “Εναρξη”, “Όλα τα προγράμματα” (εικόνα 20.9).

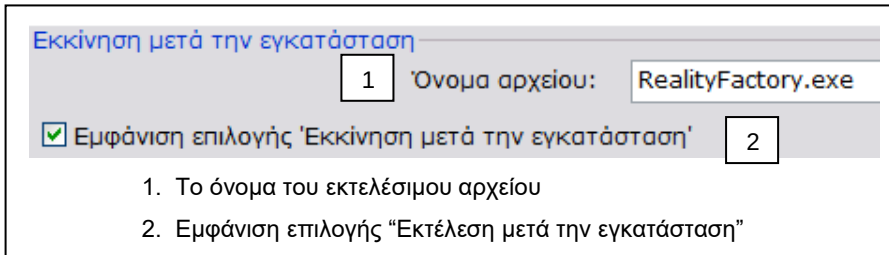


Εικόνα 20.8 Ρυθμίσεις δημιουργίας συντομεύσεων

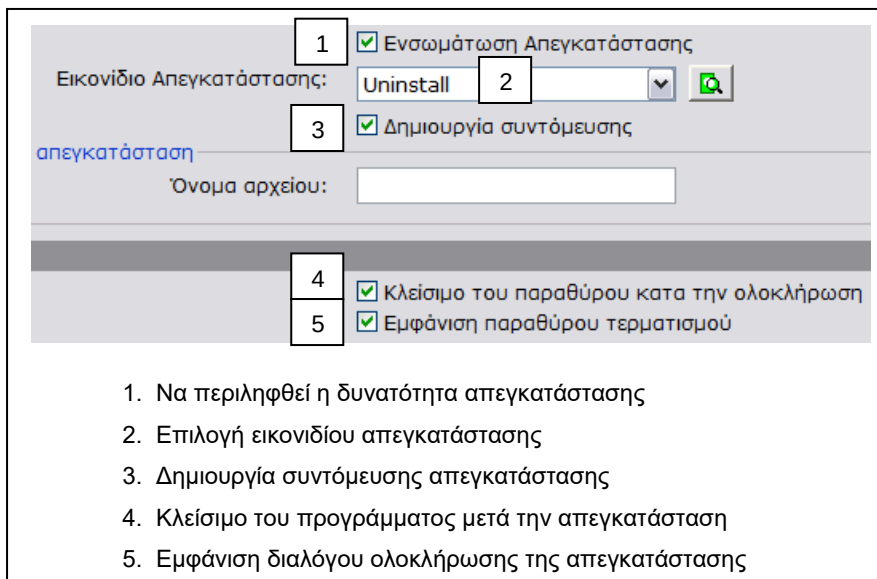
Φάκελος συντόμευσης	Όνομα συντόμευσης	Φάκελος προορισμού	Όνομα αρχείου
DESKTOP	My Application	INSTALL PATH	RealityFactory.exe
PROGRAM GROUP	My Application	INSTALL PATH	RealityFactory.exe

Εικόνα 20.9 Τελική μορφή ρυθμίσεων για τις συντομεύσεις

Τα επόμενα δύο βήματα είναι προαιρετικά και αφορούν τη δυνατότητα εκτέλεσης της εφαρμογής μετά το πέρας της εγκατάστασης και τη δυνατότητα απεγκατάστασης της εφαρμογής. Στην καρτέλα εκκίνησης  ορίζουμε ποιο αρχείο θα εκτελεστεί μετά την εγκατάσταση και αν θα εμφανίζεται η επιλογή “Εκτέλεση μετά την εγκατάσταση” (εικόνα 20.10). Από την καρτέλα απεγκατάστασης  επιλέγουμε αν θα περιληφθεί η δυνατότητα αυτή, αν θα δημιουργηθεί το αντίστοιχο εικονίδιο και αν θα εμφανίζεται διάλογος ολοκλήρωσης της διαδικασίας (εικόνα 20.11).



Εικόνα 20.10 Ρυθμίσεις καρτέλας εκτέλεσης



Εικόνα 20.11 Ρυθμίσεις καρτέλας απεγκατάστασης

Αφού ολοκληρώσουμε όλες τις ρυθμίσεις κάνουμε κλικ στο εικονίδιο έναρξης δημιουργίας του αρχείου εγκατάστασης . Το αρχείο που θα παραχθεί είναι έτοιμο για εγγραφή σ' ένα CD. Εάν θέλουμε μπορούμε βέβαια να αποθηκεύσουμε το έργο μας  για να το τροποποιήσουμε ή να το χρησιμοποιήσουμε κάποια άλλη χρονική στιγμή.

Βιβλιογραφία

- [1] Piaget J. (1985), *The Equilibration of Cognitive Structures*, Chicago, University of Chicago Press
- [2] Smith L. (1993), *Necessary Knowledge: Piagetian Perspectives on Constructivism*, Hove, Lawrence Erlbaum
- [3] Vygotsky L. S. (1978), *Mind in Society: The Development of Higher Psychological Processes*, Cambridge MA, Harvard University Press
- [4] Vygotsky L. S. (1962), *Thought and Language*, Cambridge MA, M. I. T. Press
- [5] Schank R., Abelson A. (1977) *Scripts, Plans, Goals and Understanding*, Hillsdale N. J., Lawrence Erlbaum
- [6] Nelson K., Fivush R., Hudson J., Lucariello J. (1983), "Scripts and the development of memory", in M. T. H. Chi (Ed.), *What is Memory Development the Development of?*, Basel, S. Karger
- [7] Nelson K., Gruendel J. (1986), "Children's scripts", in K. Nelson (Ed.), *Event Knowledge: Structure and Function in Development*, Hillsdale, NJ, Lawrence Erlbaum
- [8] Fivush R. (1984), "Learning about school: The development of kindergartners' school scripts", *Child Development*, 55, 1697-1709
- [9] Piaget J., Inhelder B. (1956), *The Child's Conception of Space*, New York, Humanities Press
- [10] Κολιάδης Ε. Α. (1997), *Θεωρίες Μάθησης και Εκπαιδευτική πράξη*, Τόμος Γ': *Γνωστικές θεωρίες*, Αθήνα
- [11] Azmitia M. (1988), "Peer interaction and problem solving: When are two heads better than one?", *Child Development*, 59, 87-96
- [12] Greenwood C. R., Carta J. J., Kamps D. (1990), "Teacher-mediated versus peer-mediated instruction: a review of educational advantages and disadvantages", in Foot, H. C., Morgan, M. J., Shute, R. H. (Eds.), *Children Helping Children*, Chichester, Wiley
- [13] Damon W., Phelps E. (1989), "Critical distinctions among three approaches to peer education", *International Journal of Educational Research*, 13, 9-19
- [14] Foot H. C., Shute R. J., Morgan M. J., Barron A. (1990), "Theoretical issues in peer tutoring", in H. C. Foot, M. J. Morgan, R. J. Shute (Eds.), *Children Helping Children*, Chichester, John Wiley
- [15] Tudge J. R. H., Rogoff B. (1989), "Peer influences on cognitive conflict: Piagetian and Vygotskian perspectives", in M. Bornstein, J. S. Bruner (Eds.), *Interaction in Human Development*, Hillsdale, NJ, Lawrence Erlbaum.
- [16] Wood D. (1986), "Aspects of teaching and learning", in M. Richards, P. Light (Eds.), *Children of Social Worlds: Development in a Social Context*, Cambridge, Polity
- [17] Howe C. J., Tolmie A., Rodgers C. (1992): "The acquisition of conceptual knowledge in science by primary school children: group interaction and the understanding of motion down an incline", *British Journal of Developmental Psychology*, 10, 113-130
- [18] Damon W., Phelps E. (1988), "Strategic uses of peer learning in children's education", in T. Berndt, G. Ladd (Eds.), *Children's Peer Relations*, New York, Wiley

- [19] Renault, "Safety for all programme", www.safety-for-all.com/safety/index.htm
- [20] Department for Transport, "The Hedgehogs Project", www.hedgehogs.gov.uk/index.htm
- [21] Scottish Road Safety Campaign (2002), "The Streetwiseguys Project", www.streetwiseguys.co.uk/
- [22] Tolmie A., Thomson J., Foot H., McLaren B., Whelan K. (1998), *Problems of Attention and Visual Search in the Context of Child Pedestrian Behaviour*, Department of Transport, Road Safety Report No. 10
- [23] Tolmie A., Thomson J., Foot H (2002), *Development and evaluation of a computer-based pedestrian training resource for children ages 5 to 11 years*, DFT, Road Safety Research Report No.27, www.roads.dft.gov.uk/roadsafety/research27/index.htm
- [24] Kearsley G. (2002), "The Theory into practice database", www.gwu.edu/~tip/index.html
- [25] Schwier R. A. (1995), "Issues in emerging interactive technologies", in Anglin G. J. (Ed.), *Instructional technology: Past, present and future*, Englewood, Libraries Unlimited Inc
- [26] Jonassen D. H., Mayes J. T., McAleese, R. (1993), "A manifesto for a constructivist approach to technology in higher education", in Duffy T., Jonassen D., Lowyck J. (Eds.), *Designing constructivist learning environments*, Heidelberg, FRG, Springer-Verlag
- [27] Ertmer P. A., Newby T. J. (1993), "Behaviorism, cognitivism, constructivism: Comparing critical features from an instructional design perspective", *Performance Improvement Quarterly*, 6 (4), 50-70
- [28] Piaget J. (1955), "Les stades du développement intellectuel de l'enfant et de l'adolescent", in P. Osterrieth (Ed.), *Le Problème des Stades en Psychologie de L'Enfant*, Paris, Presses Universitaires de France
- [29] Piaget J. (1969), *The Child's Conception of Time*, London, Routledge and Kegan Paul
- [30] Maturana H., Varela F. (1987), *The tree of knowledge*, Boston, New Science Library
- [31] Saettler P. (1990), *The evolution of American educational technology*, Englewood, CO, Libraries Unlimited Inc
- [32] Jonassen D. H. (1992), "Evaluating constructivist learning", in Duffy T. M., Jonassen D. H. (Eds.), *Constructivism and the technology of instruction: a conversation*, Hillsdale, NJ, Lawrence Erlbaum Associates Publishers
- [33] Bednar A. K., Cunningham D., Duffy T. M., Perry, J. P. (1995), "Theory into practice: How do we link?", in G. J. Anglin (Ed.), *Instructional technology: Past, present and future*, Englewood, CO, Libraries Unlimited, Inc
- [34] Spiro R. J., Feltovich M. J., Coulson R. J. (1991), "Cognitive flexibility, constructivism, and hypertext: Random access instruction for advanced knowledge acquisition in ill-structured domains", *Educational Technology*, May, 24-33
- [35] Tolhurst, D. (1992), "A checklist for evaluating content-based hypertext computer software", *Educational Technology*, 32 (3), 17-21
- [36] Dede, C. (1992), "The future of multimedia: Bridging to virtual worlds", *Educational Technology*, 32 (5), 54-60

- [37] Jonassen D. H. (1994), "Thinking technology: Toward a constructivist design model", *Educational Technology*, 34 (4), 34-37
- [38] Davidson K. (1998), "Education in the internet-linking theory to reality", www.oise.on.ca/~kdavidson/cons.html
- [39] Winn, W. (1993), *A Conceptual Basis for Educational Applications of Virtual Reality*, Human Interface Technology Laboratory, Report No. TR-93-9
- [40] Siegler R. S. (1983), "Information processing approaches to development", in W. Kessen (Ed.), *Handbook of Child Psychology*, Vol. 1, New York, Wiley
- [41] Johnson-Laird P. N. (1988), *The Computer and the Mind*, Cambridge, Harvard University Press
- [42] Clancey, W. J. (1993), "Situated action: A neuropsychological interpretation: Response to Vera and Simon", *Cognitive Science*, 17, 87-116
- [43] Samir Abdelmawla (1998), *Virtual Environment for Structural Education*, ACADIA'98 Paper #98-033
- [44] Wolf-Dietrich Ihlenfeldt (1997), "Virtual Reality in Chemistry", *Journal of Molecular Modeling*, 3, 386-402
- [45] Mikropoulos T. A. (1996), "Virtual Geography", *VR in the Schools*, 2 (2)
- [46] Macpherson C., Keppell M. (1998), "Virtual reality: What is the state of play in education?", *Australian Journal of Educational Technology*, 14 (1), 60-74
- [47] Chou C., Hsu H.-L., Yao Y.-S. (1997), "Construction of a virtual reality learning environment for teaching structural analysis", *Computer Applications in Engineering Education*, 5 (4), 223-230
- [48] Lanier J., Biocca F. (1992), "An Insider's View of the Future of Virtual Reality", *Journal of Communications*, 42 (4), 150-172
- [49] Bricken, W. (1990), *Learning in virtual reality*, Seattle, WA, Human Interface Technology Laboratory Technical Report HITL-M-90-5
- [50] Wann J., Mon-Williams M. (1996), "What does virtual reality NEED? Human factors issues in the design of three-dimensional computer environments", *International Journal of Human - Computer Studies*, 44, 829-847
- [51] Hedberg J., Alexander S. (1994), "Virtual reality in education: Defining researchable issues", *Educational Media International*, 31 (4), 214-220
- [52] Brey P. (1999), "The ethics of representation and action in virtual reality", *Ethics and Information Technology*, 1, 5-14
- [53] Moore, P. (1995), "Learning and teaching in virtual worlds: Implications of virtual reality for education", *Australian Journal of Educational Technology*, 11 (2), 91-102
- [54] Winn, W. D. (1994), "Educational applications of virtual reality: The case for immersion", Πρακτικά συνεδρίου *ACCE Conference on Multimedia and Education*, Vancouver, BC, ACCE

- [55] Winn, W. (1994), "Designing and using virtual environments: The advantage of immersion", Πρακτικά συνεδρίου *ED-MEDIA 94: Educational Multimedia and Hypermedia*, (pp. 695), Charlottesville, VA, Association for the Advancement of Computers in Education
- [56] Byrne C. (1996), *Water on Tap: The Use of Virtual Reality as an Educational Tool*, Διδακτορική διατριβή, University of Washington, Washington.
- [57] Gay E. (1994), "Is Virtual Reality a good teaching tool?", *Virtual Reality Special Report*, Winter, 51-59
- [58] Psotka J., Davison S. (1993) *Cognitive factors associated with immersion in virtual environments*, Alexandria, VA, Army Research Institute
- [59] Held R. M., Durlach N. I. (1992), "Telepresence", *Presence*, 1 (1), 109-112
- [60] Stanney K. M., Mourant R. R. et al. (1998), "Human factors issues in virtual environments: a review of literature", *Presence: Teleoperators & Virtual Environments*, 7 (4), 327-352
- [61] Lo W. T., So Richard H. Y. (2001), "Cyber sickness in the presence of scene rotational movements along different axes", *Applied Ergonomics*, 32, 1-14
- [62] Rushton S. K., Ribell P. M. (1999), "Developing visual systems and exposure to Virtual Reality and stereo displays: some concerns and speculations about the demands on accommodation and vergence", *Applied Ergonomics*, 30, 69-78
- [63] Helsel S. (1992), "Virtual reality in the classroom", *Educational Technology*, May 1992, 38-42
- [64] Travis D., Watson T., Atyeo M. (1994), "Human psychology in virtual environments", in L. MacDonald, J. Vince (Eds.), *Interacting with Virtual Environments*, London, John Wiley
- [65] Colburn A. (2000), "Constructivism: Science Education's Grand Unifying Theory", *Clearing House*, 74 (1), 9-12
- [66] Roblyer M. D., Edwards J. (2000), "Integrating educational technology into teaching", Upper Saddle River, NJ, Prentice-Hall, Inc
- [67] Brooks J. G., Brooks, M. G (1999), "In search of understanding: The case for constructivist classrooms", www.ascd.org/readingroom/books/brooks99toc.html
- [68] Dick W. (1991) "An instructional designer's view of constructivism", *Educational Technology*, 31 (5), 41-44
- [69] Boethel M., Dimock K. V. (1999), "Constructing knowledge with technology. A review of the literature", www.sedl.org/pubs/tec27/nonflash.html
- [70] Tam M. (2000), "Constructivism, instructional design, and technology: Implications for transforming distance learning", http://ifets.ieee.org/periodical/vol_2_2000/tam.html
- [71] Salomon G. (2000), "It's not just the tool, but the educational rationale that counts" <http://construct.haifa.ac.il/~gsalomon/edMedia2000.html>
- [72] Kalawsky R. S. (1996), "Exploiting Virtual Reality Techniques in Education and Training: Technological Issues", AGOCR report, www.agocg.ac.uk/reports/virtual/vrtech/
- [73] Pantelidis V. S. (1993) "Virtual reality in the classroom", *Educational Technology*, April, 23-27

- [74] Bruner J. S. (1968), *Process of cognitive growth: Infancy*, Clark University Press, USA
- [75] Salzman M. C., Dede C., Loftin R. B., Chen J. (1999), "A Model for Understanding How Virtual Reality Aids Complex Conceptual Learning", *Presence*, 8, 293-316
- [76] Slavin R. E. (1991), *Cooperative Learning: Theory, Research, and Practice*, Engelwood Cliffs, NJ, Prentice-Hall
- [77] Chan T.-W., (1996), "Learning Companion Systems, Social Learning Systems and the Global Social Learning Club", *Journal of Artificial Intelligence in Education*, 7 (2), 125-159
- [78] Mikropoulos T. A., Chalkidis A., Katsikis A., Emvalotis A. (1998), "Students' Attitudes Towards Educational Virtual Environment", *Education and Information Technologies*, 3 (2), 137-148, June
- [79] Psotka J. (1996), *Immersive tutoring systems, virtual reality and education and training*, U.S. Army Research Institute, ATTN, PERI-IIC
- [80] Rose H. (1995), *Assessing Learning in VR: Towards Developing a Paradigm Virtual Reality Roving Vehicle*, R-95-1, Human Interface Technology Laboratory
- [81] Youngblut, C. (1998), *Educational Uses of Virtual Reality Technology*, Technical Report IDA Document D-2128, Institute for Defence Analyses, Alexandria, VA
- [82] Pantelidis S. V. (1996), "Suggestions on When to Use and When Not to Use Virtual Reality in Education", *VR in the Schools*, 2 (1), 18, June, VREL Publications
- [83] Moore G. E. (1965), "Cramming more components onto integrated circuits", *Electronics*, 38 (8), April 19
- [84] Mikropoulos, T. A. (1997), "Virtual Environments in Science Education", International Conference *Virtual Reality in Education & Training*, (pp. 43-48), Loughborough, UK
- [85] Delaney D. (2002), *The market for visual simulation/virtual reality systems*, 5th edition, CyberEdge Information Services Inc.
- [86] Paschedag D. (2003), "Video Card History (1996 to the present)", www.fastsilicon.com/showarticle.php?a=28&p=1
- [87] Simpson, J. (2002), "Game Engine Anatomy", www.extremetech.com/article2/0,3973,19746,00.asp
- [88] The PC Guide (2001), "Video modes, Resolution and Color", www.pcguide.com/ref/video/modes.htm
- [89] Webopedia, "What is refresh rate? - A Word Definition From the Webopedia Computer Dictionary", www.webopedia.com/TERM/R/refresh.html
- [90] Walrath J. (1999), "30 Frames per Second vs. 60 Frames per Second", PenStar Systems, LLC, www.daniele.ch/school/30vs60/30vs60_3.html
- [91] Ebert D. S., Musgrave F. K., Peachey D., Perlin K., Worley S. (2003), *Texturing & Modeling: A Procedural Approach*, 3rd edition, Morgan Kaufmann Publishers, An Imprint of Elsevier Science
- [92] Satava R. M., Jones S. B. (1999), "Medical Applications of Virtual Reality", in M. K. Stanney (Ed.), *VE Handbook*, Lawrence Erlbaum Associates, Inc

- [93] Neider, J., Davis, T., Woo, M. (1994), "OpenGL Programming Guide", www.njnet.edu.cn/info/ebook/graph/OpenGLRedBook/479.html
- [94] Sink, K. (2002), "DirectX8 and VB.NET Development", www.c-sharpcorner.com/FreeLib/Pearson/0672322250/0672322250_ch01.html
- [95] Crockett T. W. (1996), "Beyond the Renderer: Software Architecture for Parallel Graphics and Visualization", Πρακτικά συνεδρίου *First Eurographics Workshop on Parallel Graphics and Visualisation*, Sept. 1996, (pp. 1-15), A. Chalmers, A. Jansen (Eds.), Alpha Books, Bristol, U.K.
- [96] Kerlow I. V. (2003), *The Art of 3D: Computer Animation and Effects*, 3rd edition, New York, Wiley
- [97] Tan D. S., Robertson G. G., Czerwinski M. (2000), "Exploring 3D Navigation: Combining Speed-coupled Flying with Orbiting", Πρακτικά συνεδρίου *CHI 2001 Conference on Human Factors in Computing Systems*, Seattle, WA
- [98] Isdale J. (1999), "Authoring and Simulation Toolkits", VR News, http://vr.isdale.com/vrTech-Reviews/AuthoringTools_July1999.htm
- [99] Brosens, J., Hoekman, R., Willems, D., Dijkstra, J., de Vries I. B. (2001), "Game engines in Architecture", *Capita Selecta Ontwerpsystemen*, www.bwk.tue.nl
- [100] *The Virtual Reality Modeling Language*. International Standard ISO/IEC 14772-1:1997
- [101] Wright G. (2000), "VRML Never Dies, It Just Changes Its Name", 3D Direct, www.3dgate.com/news_and_opinions/000208/0208wright.html
- [102] Stokes, J., Ragan-Kelley, J. (2001), "Final Fantasy: The Technology Within", www.ars-technica.com/wankerdesk/01q3/ff-interview/ff-interview-1.html
- [103] Sense8 Corporation, "WorldToolKit Release 8 Technical Overview", www.sense8.com
- [104] SmartVR, "SmartVerse 2.0 Technology", www.smartvr.com/Technology.html
- [105] Virtools SA, "Virtools Dev", www.virtools.com/solutions/products/virtools_dev.asp
- [106] *The Economist* newspaper (2002), "Console wars", June 20, www.economist.com/business/displayStory.cfm?story_id=1189352
- [107] Id Software, "Wolfenstein 3D", www.idsoftware.com/games/wolfenstein/wolf3d/
- [108] Id Software, "Doom 3D", www.doom3.com/
- [109] Id Software, "Quake 3D", www.idsoftware.com/games/quake/quake3-gold/
- [110] Infogrames Entertainment S.A., "Unreal Tournament", <http://unreal.com/index2.html>
- [111] 3D Mark 03 Pro, <http://www.futuremark.com/products/3dmark03/>
- [112] 3D Mark 2001 SE Pro, <http://www.futuremark.com/products/3dmark2001/>
- [113] Smith, S. L., Lachlan, K. A., Tamborini, R. (2003), "Popular Video Games: Quantifying the Presentation of Violence and its Context", *Journal of Broadcasting and Electronic Media*, 47 (1)
- [114] Seriously ! Forums, <http://forums.seriouszone.com/>
- [115] Crace, J. (2002), "War game", *The Guardian*, July 9
- [116] Knerr, B. W., Lampton, D. R., Singer, M. J., Witmer, B. G., Goldberg, S. L. (1998), *Virtual Environments for Dismounted Soldier Training and Performance: Results, Recommendations, and Issues*, Army Research Institute, Simulator Systems Research Unit, Orlando, FL

- [117] Richens, P. (2000), "Playing games", Martin Centre for Architectural and Urban Studies, University of Cambridge, www.arct.cam.ac.uk/research/pubs/pdfs/rich00a.pdf
- [118] DeLeon V. J., Berry H. R. (1998), "Virtual Florida Everglades", www.digitalo.com/de-leon/vrglades/
- [119] NASA, ISS commercial development, "Virtual International Space Station", <http://commercial.hq.nasa.gov/viss/viss.html>
- [120] Price C., (2006), "A Crisis in Physics Education: Games to the Rescue!", Innovation in Teaching And Learning in Information and Computer Sciences, Vol. 5, Issue 3, <http://www.ics.heacademy.ac.uk/italics/vol5iss3/price.pdf>
- [121] Uhlmann E., Swanson J. (2003), "Exposure to violent video games increases automatic aggressiveness", *Journal of Adolescence*, www.elsevier.com/locate/jado
- [122] Anderson C. A. (2003), "An update on the effects of playing violent video games", *Journal of Adolescence*, www.elsevier.com/locate/jado
- [123] Gentile D. A., Lynch P. J., Linder J. R., Walsh D. A. (2003), "The effects of violent video game habits on adolescent hostility, aggressive behaviors, and school performance", *Journal of Adolescence*, www.elsevier.com/locate/jado
- [124] Bensley L., Eenwyk J. van (2001), "Video Games and Real-Life Aggression: Review of the Literature", *Journal of Adolescent Health*, 29, 244-257
- [125] Harris J. (2001), *The effects of computer games on young children-a review of the research*, Research, Development and Statistics Directorate, RDS Occasional Paper No 72
- [126] Subrahmanyam, K., Greenfield, P. M. (1994), "Effect of video game practice on spatial skills in girls and boys", Special issue: "Effects of interactive entertainment technologies on development", *Journal of Applied Developmental Psychology*, 15, 13-32
- [127] Subrahmanyam K., Kraut R., Greenfield P., Gross E. (2001), "New Forms of Electronic Media", D. Singer, J. Singer (Eds.), *Handbook of Children and the Family* (pp. 73-99), Thousand Oaks, CA, Sage
- [128] Funk J. B., Pasold T., Baumgardner J. (2003), "How Children Experience Playing Video Games", ACM International Conference Proceeding Series archive, *Proceedings of the Second International Conference on Entertainment Computing*, (pp. 1-14), Pittsburgh, Pennsylvania
- [129] Irwin H. J. (1999), "Pathological and non-pathological dissociation: The relevance of childhood trauma", *The Journal of Psychology*, 133, 157-164
- [130] Putnam F. W. (1993), "Dissociative disorders in children: Behavioral problems and profiles", *Child Abuse and Neglect*, 17, 39-45
- [131] Preston J. (1998), "Mediated environments and consciousness", in J. Gackenbach (Ed.), *Psychology and the Internet*, San Diego, Academic Press
- [132] Csikszentmihalyi M., Csikszentmihalyi, I. S. (1988), *Optimal experience: Psychological studies of flow in consciousness*, Cambridge, UK, Cambridge University Press
- [133] Moneta G. B., Csikszentmihalyi M. (1996), "The effect of perceived challenges and skills on the quality of subjective experience", *Journal of Personality*, 64, 275-310

- [134] Conitec Corporation, "3D Game Studio overview", www.conitec.net/a4info.htm
- [135] Slam Software, "AMP II 3D engine overview", www.4drulers.com/amp.html
- [136] Criterion Software, "RenderWare overview", www.renderware.com/products.asp
- [137] Comer D. E. (2002), *Δίκτυα και διαδίκτυα υπολογιστών*, 3η έκδοση, Κλειδάριθμος, Αθήνα
- [138] DevMaster.Net, "Game engines list", <http://www.devmaster.net/engines/list.php>
- [139] Reality Factory, "Reality Factory overview", <http://www.dhost.info/realityfactory/>
- [140] Eclipse Entertainment , "Genesis 3D", <http://www.genesis3d.com/>

