

**ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ
ΜΕΤΑΠΤΥΧΙΑΚΟ ΠΡΟΓΡΑΜΜΑ ΣΠΟΥΔΩΝ**



ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

«Σχεδιασμός & Υλοποίηση VRML Browser σε Java»

ΜΥΛΩΝΑΣ ΦΟΙΒΟΣ-ΑΠΟΣΤΟΛΟΣ
Α.Μ.: M467

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ:

Αθήνα, Οκτώβριος 2003



Πρόλογος

Σκοπός του παρόντος κειμένου είναι η πλήρης περιγραφή και παρουσίαση της διπλωματικής μου εργασίας, στα πλαίσια του Προγράμματος Μεταπτυχιακών Σπουδών του Τμήματος Πληροφορικής και Τηλεπικοινωνιών του Εθνικού και Καποδιστριακού Πανεπιστημίου Αθηνών. Αντικείμενο της εργασίας αποτέλεσε η σχεδίαση, υλοποίηση και κατασκευή μιας εφαρμογής λογισμικού, μέσω της οποίας θα είναι δυνατή η πλοήγηση σε **Virtual Reality Modeling Language** [\[L1\]](#), [\[L2\]](#) κόσμους, χρησιμοποιώντας την ευρέως διαδεδομένη υπολογιστική πλατφόρμα της γλώσσας προγραμματισμού **Java** [\[L4\]](#), συνδυάζοντας τις παραπάνω δύο τεχνολογίες με τρόπο πρωτοποριακό και πρωτότυπο..

Η διπλωματική αυτή εργασία εκπονήθηκε με την επίβλεψη και την καθοδήγηση του καθηγητή κ. Δημοσθένη Αναγνωστόπουλου.

Μυλωνάς Φοίβος – Απόστολος

Οκτώβριος 2003

Περίληψη

Η παρούσα διπλωματική εργασία έχει ως κύριο στόχο την ανάπτυξη μιας εφαρμογής λογισμικού, μέσω της οποίας θα καθίσταται δυνατή για τους χρήστες της η πλοήγηση σε Virtual Reality Modeling Language (VRML) [\[L1\]](#), [\[L2\]](#) κόσμους. Η εφαρμογή θα πρέπει απαραίτητα να είναι υλοποιημένη στην πλατφόρμα της Java [\[L4\]](#), για λόγους εύκολης μεταφερσιμότητας και ανεξαρτησίας από λειτουργικά συστήματα και πλατφόρμες υπολογιστών. Περιλαμβάνει, δε, την ανάπτυξη κατάλληλου και εύχρηστου Graphical User Interface (GUI), με αυξημένη λειτουργικότητα και δυνατότητες πλοήγησης.

Αν θελήσουμε να συνοψίσουμε τα βασικά στοιχεία αυτής της διπλωματικής εργασίας, αυτά επικεντρώνονται σε δύο βασικούς άξονες. Ο πρώτος περιστρέφεται γύρω από τις λειτουργικές παραμέτρους, τις δυνατότητες και την αισθητική παρουσίαση της εφαρμογής, και οριοθετεί την ευχρηστία, την ευκολία χρήσης και την «ελαφρότητα» της, όσον αφορά την επιβάρυνση του εκάστοτε συστήματος στο οποίο «τρέχει». Ο δεύτερος αφορά την αξιοπιστία και την διασύνδεση με τα υπάρχοντα περιβάλλοντα με τα οποία καλείται να συνεργαστεί, και περιστρέφεται γύρω από τη χρήση υπαρχόντων VRML κόσμων και την οργάνωση των ίδιων των χρηστών.

Συμπερασματικά, λοιπόν, πρόκειται για μια πλήρη εφαρμογή σε παραθυρικό περιβάλλον, η οποία αναμένεται να προσφέρει άλλη μία εναλλακτική και ανεξάρτητη πλατφόρμας λύση για την πλοήγηση σε VRML κόσμους

Λέξεις Κλειδιά: Java, VRML, GUI, πλοήγηση.

Πίνακας περιεχομένων

1	ΕΙΣΑΓΩΓΗ	7
1.1	Οργάνωση εργασίας.....	10
2	ΒΑΣΙΚΕΣ ΈΝΝΟΙΕΣ	13
2.1	Java:	15
2.1.1	Περιγραφή	15
2.1.2	Βασικά χαρακτηριστικά.....	16
2.1.3	Εργαλεία	18
2.1.4	Περιγραφή βασικών εννοιών	19
2.1.5	Java - Javascript.....	21
2.2	VRML:	21
2.2.1	Περιγραφή	21
2.2.2	VRML 2.0 Specification.....	22
2.3	Borland JBuilder:	25
3	ΠΕΡΙΓΡΑΦΗ ΠΕΡΙΒΑΛΛΟΝΤΟΣ	26
3.1	Ερευνητική Προσέγγιση Περιβάλλοντος.....	28
3.2	Σύγκριση με Εμπορικές Εφαρμογές.....	30
3.3	Παρεμφερείς Ερευνητικές Εργασίες.....	33
3.4	Πλεονεκτήματα έναντι άλλων προσπαθειών	36
3.5	Ομάδες Χρηστών ενός τέτοιου Περιβάλλοντος.....	37
3.6	Τελικό Ζητούμενο.....	38
4	ΥΛΟΠΟΙΗΣΗ ΠΕΡΙΒΑΛΛΟΝΤΟΣ.....	39
4.1	Επιλογή Πλατφόρμας και Εργαλείων	39
4.2	Εργαλεία Ανάπτυξης Περιβάλλοντος	40
4.3	Υλοποίηση Java:	41
4.3.1	Παράδειγμα κώδικα υλοποίησης	42
5	ΕΓΧΕΙΡΙΔΙΟ ΧΡΗΣΗΣ - MANUAL.....	52
5.1	Εισαγωγή.....	52

5.2	Εγκατάσταση.....	53
5.3	Βασικά μενού - Menus:.....	54
5.4	Υπομενού - Submenus:	55
5.4.1	File	55
5.4.2	Navigate.....	62
5.4.3	Display.....	64
5.4.4	Help	68
5.5	Κουμπιά πλοήγησης - Buttons:.....	69
5.5.1	Reset button	69
5.5.2	Zoom buttons	69
5.5.3	Explore mode.....	70
5.5.4	Examine mode	70
5.6	Παράδειγμα εκτέλεσης:	72
6	ΕΠΙΛΟΓΟΣ.....	73
6.1	Μελλοντικές Επεκτάσεις Περιβάλλοντος.....	73
6.2	Εμπειρία χρήσης	74
6.3	Σύνοψη και Συμπεράσματα.....	75
7	ΒΙΒΛΙΟΓΡΑΦΙΑ	76
7.1	Βιβλιογραφικά Αναγνώσματα:	76
7.2	Links:	77
7.3	Request For Comment (RFC):	79
7.4	Papers:.....	80

1

Εισαγωγή

Η παρούσα διπλωματική εργασία έχει ως κύριο στόχο την ανάπτυξη ενός περιβάλλοντος λογισμικού, μέσω του οποίου θα καθίσταται δυνατή για τους χρήστες του η πλοήγηση σε **Virtual Reality Modeling Language (VRML)** [\[L1\]](#), [\[L2\]](#) κόσμους, με τρόπο απλό, κατανοητό και εύχρηστο με άμεση εφαρμογή του στο Internet [\[L6\]](#) γενικά και στο World Wide Web (www) [\[L8\]](#) συγκεκριμένα. Το αντικείμενο της εργασίας τοποθετείται στο χώρο των Γραφικών Υπολογιστών και του Διαδικτύου και επικεντρώνεται στην δημιουργία ενός αξιόπιστου, εύχρηστου, λειτουργικού και φιλικού προς το χρήστη browser, μέσω του οποίου θα είναι δυνατή η πλήρης πλοήγηση και ο χειρισμός των VRML κόσμων. Το εν λόγω περιβάλλον θα πρέπει απαραίτητα να είναι υλοποιημένο στην πλατφόρμα της Java [\[L4\]](#), για λόγους εύκολης μεταφερισιμότητας και ανεξαρτησίας από λειτουργικά συστήματα και πλατφόρμες υπολογιστών. Περιλαμβάνει, δε, την ανάπτυξη κατάλληλου και εύχρηστου **Graphical User Interface (GUI)**, με αυξημένη λειτουργικότητα και δυνατότητες πλοήγησης.

Έμπνευση για την επιλογή του θέματος της εργασίας αποτέλεσε η καινοτομία του όλου εγχειρήματος, η έλλειψη ανάλογης επιστημονικής προσέγγισης στο χώρο, αλλά και η ύπαρξη ευρέως διαδεδομένων παρεμφερών συστημάτων στο Διαδίκτυο, όπως ο γνωστός CosmoPlayer [\[L26\]](#). Η περιέργεια για τον τρόπο λειτουργίας στο εσωτερικό τους, η πρόκληση για την υλοποίηση ενός τέτοιου συστήματος από το μηδέν και η θέληση τόσο για πρακτική ενασχόληση όσο και για θεωρητική προσέγγισή τους, αποτέλεσαν τα κύρια κίνητρά μας. Το τελικό προϊόν πιστεύω ότι δικαιώνει την επιλογή/επιμονή μας.

Αν θελήσουμε να συνοψίσουμε τα βασικά στοιχεία αυτής της εργασίας, αυτά επικεντρώνονται σε δύο βασικούς άξονες. Ο πρώτος περιστρέφεται γύρω από τις λειτουργικές παραμέτρους, τις δυνατότητες και την αισθητική παρουσίαση της εφαρμογής, και οριοθετεί την ευχρηστία, την ευκολία χρήσης και την «ελαφρότητα» της, όσον αφορά την επιβάρυνση του εκάστοτε συστήματος στο οποίο «τρέχει». Ο δεύτερος αφορά την αξιοπιστία και την διασύνδεση με τα υπάρχοντα περιβάλλοντα με τα οποία καλείται να συνεργαστεί, και περιστρέφεται γύρω από τη χρήση υπάρχοντων VRML κόσμων και την οργάνωση των ίδιων των χρηστών.

Στα πλαίσια της υλοποίησης μιας ολοκληρωμένης εφαρμογής, παρέχονται όλες οι τυπικές λειτουργίες και δυνατότητες των τυπικών –proprietary- VRML browsers του εμπορίου, στο πλαίσιο όμως της ανοικτής και εύκολα παραμετροποιήσιμης, ανάλογα με τις εκάστοτε ανάγκες των χρηστών, υπολογιστικής πλατφόρμας της Java. Τον κύριο γνώμονα για όλα τα παραπάνω αποτέλεσε η ευκολία στη χρήση της εφαρμογής και η πληρότητά της στις παρεχόμενες δυνατότητες του browser.

Συμπερασματικά, λοιπόν, πρόκειται για μια πλήρη εφαρμογή σε παραθυρικό περιβάλλον, η οποία αναμένεται να προσφέρει άλλη μία εναλλακτική και ανεξάρτητη πλατφόρμας λύση για την πλοήγηση σε VRML κόσμους, παρέχοντας στον τελικό χρήστη μια πληθώρα από δυνατότητες, τόσο όσον αφορά την πλοήγησή του, όσο και όσον αφορά τη διαχείριση του ίδιου του

περιβάλλοντος. Ιδιαίτερη προσοχή δόθηκε στην ευχρηστία του συστήματος και στη φιλικότητα προς τον χρήστη, ενώ σημαντικό ρόλο στη συγγραφή του κώδικα, ο οποίος αποτέλεσε το τελικό σύστημα, έπαιξε η κατάτμησή του σε components και η επαναχρησιμοποίησή τους σε ποικίλα επαναλαμβανόμενα σημεία. Με τον τρόπο αυτό αυξήθηκε στο έπακρο η στιβαρότητα του όλου περιβάλλοντος και η ενδολειτουργικότητά του. Τέλος, κατέστη, επίσης, δυνατή και η απροβλημάτιστη επεκτασιμότητα της εφαρμογής, μέσω της ορθής και προσεκτικής σχεδίασής της και της υλοποίησής της με τα συγκεκριμένα εργαλεία και τη γλώσσα προγραμματισμού Java.

Ο τελικός χρήστης θα μείνει ικανοποιημένος από το επίπεδο της προσφερόμενης ποιότητας του λογισμικού και χρησιμοποιώντας τα εύχρηστα μενού του περιβάλλοντος θα έλθει πιο κοντά στον κόσμο και την κοινότητα της VRML, με απώτερο σκοπό την ενασχόληση του με αυτήν, μετά από την πρώτη του γνωριμία μέσω του περιβάλλοντός μας. Ωστόσο, και ο προχωρημένος χρήστης της VRML θα βρει ενδιαφέρον στον εν λόγω browser, κυρίως λόγω των δυνατοτήτων που αυτός του προσφέρει σε σχέση με τον εμπορικό ανταγωνισμό και της εύκολης μεταφερσιμότητάς του. Τέλος, και εμείς οι ίδιοι αποκομίσαμε τα βέλτιστα από αυτή την εργασία, μιας και εξαντλήσαμε το χώρο των δύο γλωσσών προγραμματισμού και είδαμε πολλές και διαφορετικές απόψεις επί των σχετικών θεμάτων.

1.1 Οργάνωση εργασίας

Η δόμηση της διπλωματικής αυτής εργασίας αναλύεται σε κεφάλαια και ενότητες κεφαλαίων. Συγκεκριμένα έχουμε:

Στο πρώτο κεφάλαιο (**Κεφάλαιο 1**) της εργασίας ασχολούμαστε με την εισαγωγή στο αντικείμενο, τις ανάγκες και την χρησιμότητα της διπλωματικής εργασίας. Στην πρώτη ενότητα παρουσιάζεται το αντικείμενο της, οι ανάγκες του προβλήματος προς επίλυση και ο απώτερος στόχος του όλου εγχειρήματος, μαζί και κάποια στοιχειώδη χαρακτηριστικά της, ενώ η δεύτερη ενότητα ασχολείται με την δομή της παρούσας εργασίας.

Ακολουθεί (**Κεφάλαιο 2**) το θεωρητικό υπόβαθρο της εργασίας, παραθέτοντας μια αναλυτική αναφορά στις χρησιμοποιούμενες τεχνολογίες και τεχνικές κατά τη διάρκεια ανάπτυξής της. Ξεκινάμε στη πρώτη ενότητα του κεφαλαίου με μια αναφορά στο μέσο (Internet [\[L6\]](#)) και την επίδρασή του στη σημερινή καθημερινή ζωή και στη συνέχεια σταδιακά εισάγουμε τον αναγνώστη στις βασικές έννοιες και τεχνολογίες που χρησιμοποιήθηκαν. Στη συνέχεια, στην δεύτερη ενότητα του κεφαλαίου, πραγματοποιείται μια εκτενής παρουσίαση της γλώσσας προγραμματισμού Java [\[L4\]](#) και των στηριζόμενων σε αυτή χρησιμοποιούμενων από την εφαρμογή μας τεχνολογιών και εργαλείων. Ακολουθεί στην επόμενη ενότητα η περιγραφή και μια σύντομη παρουσίαση της VRML [\[L1\]](#), [\[L2\]](#), [\[L19\]](#) ως γλώσσας μοντελοποίησης, ούτως ώστε να εισαχθεί ο αναγνώστης στις δυνατότητές της. Να σημειωθεί στο σημείο αυτό, ότι η κατανόηση των βασικών θεωρητικών δομών πάνω στις οποίες στηρίχθηκε η εκπόνηση αυτής της διπλωματικής εργασίας κρίθηκε ιδιαίτερα σημαντική και γι' αυτό δόθηκε και ιδιαίτερη βαρύτητα σε αυτή. Το κεφάλαιο κλείνει με μία ενότητα αφιερωμένη στη θεμελιώδη λογική του χρησιμοποιούμενου εργαλείου Borland JBuilder [\[L24\]](#).

Στο τρίτο κεφάλαιο (**Κεφάλαιο 3**) πραγματοποιείται η τοποθέτηση της διπλωματικής εργασίας στο γενικότερο πλαίσιο της σχετικής περιοχής των VRML browsers και του οποίου η σημερινή μορφή αναπτύσσεται διεξοδικά.

Επιχειρείται, δε, η σύγκρισή της με υπάρχουσες σχετικές εφαρμογές και περιβάλλοντα, με παράθεση σχετικών αναλυτικών benchmarks. Ειδικότερα, στην πρώτη ενότητα του κεφαλαίου πραγματοποιείται μια περιγραφή του περιβάλλοντος, στην δεύτερη ενότητα επιχειρείται μια συνολική ερευνητική προσέγγιση του χώρου, στην τρίτη ακολουθεί η κυρίως σύγκρισή του με τα υπάρχοντα εμπορικά πακέτα, ενώ στην τέταρτη ενότητα πραγματοποιείται αναφορά σε παρεμφερείς ερευνητικές εργασίες του χώρου και στην πέμπτη επιχειρείται η παρουσίαση των πλεονεκτημάτων που παρουσιάζει έναντι των τελευταίων. Τέλος, στην έκτη ενότητα του κεφαλαίου, επιχειρείται μια κατηγοριοποίηση στους εν δυνάμει χρήστες ενός τέτοιου συστήματος προκειμένου να διερευνηθεί και με αυτόν τρόπο τυχούσα ή μη χρησιμότητά του και τελικά στην έβδομη και τελευταία ενότητα του κεφαλαίου έχουμε ξεκάθαρα το τελικό ζητούμενο, το οποίο δεν είναι άλλο από την ίδια την κατασκευή ενός τέτοιου συστήματος.

Το επόμενο κεφάλαιο (**Κεφάλαιο 4**) ασχολείται με την καθαυτό υλοποίηση του περιβάλλοντος, ξεκινώντας από την περιγραφή της πλατφόρμας και των εργαλείων που χρησιμοποιήθηκαν κατά την ανάπτυξή του. Στην επόμενη ενότητα γίνεται αναφορά στα επιμέρους εργαλεία που συνέβαλαν στην ανάπτυξή του, ενώ αμέσως μετά παρατίθενται πληροφορίες σχετικές με την αρχιτεκτονική σχεδίασης και υλοποίησης των components του κώδικα του υλοποιηθέντος συστήματος. Τέλος, περιλαμβάνεται και μία ενότητα σχετικά με την υλοποίηση στη γλώσσα προγραμματισμού Java και παρατίθεται και ένα ενδεικτικό παράδειγμα από τον σχετικό κώδικα.

Στο κεφάλαιο (**Κεφάλαιο 5**) του τόμου αυτού αναπτύσσεται το εγχειρίδιο χρήσης της αναπτυχθείσας εφαρμογής. Αρχικά παρατίθεται μια λεπτομερής διαδικασία εγκατάστασης σε κάθε πιθανή υπολογιστική πλατφόρμα, έτσι ώστε ο τελικός χρήστης να είναι κατάλληλα προετοιμασμένος κατά την εγκατάσταση του περιβάλλοντός μας. Κατόπιν, παρουσιάζονται σε επιμέρους ενότητες αναλυτικά τα μενού και υπομενού που είναι σε θέση να χρησιμοποιήσει ο χρήστης της,

καθώς και τα κουμπιά της μπάρας που βρίσκεται στο κάτω μέρος της εφαρμογής και διευκολύνει τον χρήστη στην πλοήγηση σε έναν VRML κόσμο.

Ο επίλογος της διπλωματικής εργασίας αναπτύσσεται στο έκτο κεφάλαιο (**Κεφάλαιο 6**) σε έκταση τριών ενοτήτων. Η πρώτη αναφέρεται στα συμπεράσματα που είμαστε σε θέση να εξάγουμε από το όλο εγχείρημα, ενώ στη δεύτερη γίνεται λόγος για την εμπειρία που αποκομίσαμε από την χρήση του περιβάλλοντος VRMLbro. Κάποιες μελλοντικές επεκτάσεις, οι οποίες μπορούν να πραγματοποιηθούν με σκοπό τον εμπλουτισμό των προσφερομένων υπηρεσιών και λειτουργιών προς τους χρήστες της εφαρμογής, αναφέρονται στην ενότητα τρίτα.

Τέλος, στο έβδομο και τελευταίο κεφάλαιο της εργασίας (**Κεφάλαιο 7**) αναφέρεται η πλήρης βιβλιογραφία που χρησιμοποιήθηκε για τη συγγραφή του παρόντος κειμένου, αλλά και για την ανάπτυξη της ίδιας της εφαρμογής, κατηγοριοποιημένη σε ενότητες ανάλογα το μέσο που χρησιμοποιήθηκε. Συγκεκριμένα, η πρώτη ενότητα παρουσιάζει τα βιβλία που αναγνώστηκαν και χρησιμοποιήθηκαν κατά την υλοποίηση της εργασίας ή στα οποία πραγματοποιούνται αναφορές μέσα στο κείμενο, η δεύτερη ενότητα τα http links από πηγές στο web, η τρίτη τα σχετικά Requests for Comments, και η τέταρτη τα διάφορα άρθρα/κείμενα σε μορφή Portable Data Format (pdf) .

2

Βασικές Έννοιες

Το Διαδίκτυο (Internet) [\[L6\]](#) έχει μπει πλέον για τα καλά στην καθημερινή ζωή μιας μεγάλης πλειοψηφίας ανθρώπων της σημερινής κοινωνίας, και ιδιαίτερα εκείνων που σχετίζονται άμεσα με την ακαδημαϊκή κοινότητα και έχει γίνει αναπόσπαστο κομμάτι της. Πρόκειται για μια νέα πραγματικότητα, για έναν νέο τρόπο εργασίας, εκπαίδευσης και άντλησης πληροφοριών, πέρα βέβαια από τις όποιες άλλες δυνατότητες μπορεί αυτό να παρέχει στον άνθρωπο, όσον αφορά τη διασκέδαση και την ψυχαγωγία του.

Η περιήγηση στο Internet ανοίγει τους κρουνούς της πληροφορίας και της πληροφόρησης, με αποτέλεσμα τα πάντα να βρίσκονται σε απόσταση ενός click, αλλά και καταργεί τις γεωγραφικές αποστάσεις μεταξύ των ανθρώπων κάνοντας την επικοινωνία πιο εύκολη από ποτέ: αναφορές σε σημαντικά γεγονότα, τα τελευταία νέα, τα καιρικά φαινόμενα, οι εξελίξεις στην τεχνολογία ακόμα και τα αποτελέσματα αθλητικών αγώνων, αλλά και οι φίλοι και συγγενείς σε μια άλλη ήπειρο είναι πλέον στο δωμάτιό μας. Επιτυγχάνεται με άλλα λόγια γρήγορη και άμεση επικοινωνία ανθρώπων παγκοσμίως και σε όλα τα επίπεδα (ψυχαγωγικό,

ενημερωτικό, επαγγελματικό, εμπορικό, κ.τ.λ.)...Πώς όμως τα καταφέρνει όλα αυτά το Internet σήμερα;

Το Internet είναι το μεγαλύτερο δίκτυο υπολογιστών. Αριθμεί εκατοντάδες χιλιάδες υπολογιστές συνδεδεμένους μεταξύ τους και καινούριοι να συνδέονται σε αυτό με ολοένα αυξανόμενο ρυθμό. Ξεκίνησε κυρίως για εκπαιδευτικούς σκοπούς, αλλά μετεξελίχθηκε σε ένα παγκόσμιο δίκτυο ποικίλων δραστηριοτήτων. Αποτελείται στην ουσία από πολλούς υπολογιστές συνδεδεμένους μεταξύ τους σε υποδίκτυα τοπικού χαρακτήρα, τα λεγόμενα **Local Area Networks (LAN)**, τα οποία συνδέονται περαιτέρω υλοποιώντας μεγαλύτερα **Wide Area Networks (WAN)**, τα οποία και συνεχίζουν την αλυσίδα κατά κάποιο τρόπο δημιουργώντας τελικά έναν πλήρως αποκεντρωμένο γράφο, με εκατομμύρια κόμβους-υπολογιστές. Χαρακτηριστικό του αποτελεί η εκ φύσεως παντελής έλλειψη κεντρικής διοίκησης, γεγονός που του προσδίδει τεράστια δύναμη ως μέσο.

Οι προοπτικές ανάπτυξης του ανοίγονται τεράστιες. Ήδη νέες μορφές των βασικών του πρωτοκόλλων κάνουν την εμφάνισή τους, με ριζοσπαστικές αλλαγές στην σχεδίαση και υλοποίησή τους (βλ. IPv6, [\[L3\]](#)), γεγονός που του δίνει νέες διαστάσεις. Το Internet2 [\[L5\]](#) βρίσκεται ήδη προ των πυλών με ταχύτητες πολλαπλάσιες του υπάρχοντος δικτύου, ενώ μια πληθώρα τεχνολογιών και εφαρμογών στηρίζεται, χρησιμοποιεί και αναπτύσσει το ίδιο το Internet, όπως ακριβώς συμβαίνει με τους VRML browsers.

2.1 Java:

2.1.1 Περιγραφή

Η Java [\[L4\]](#) αποτελεί μία ιδιαίτερα διαδεδομένη αντικειμενοστρεφή γλώσσα προγραμματισμού ειδικά σχεδιασμένη για χρήση στο καταναμεμημένο περιβάλλον του Internet [\[L6\]](#), [\[L7\]](#). Πέρα από αυτό θα μπορούσε να ισχυριστεί κανείς ότι διαμορφώνει μια ολόκληρη πλατφόρμα ανάπτυξης λογισμικού, βασισμένη στη δύναμη των δικτύων γενικότερα και με moto την αρχή: «Write Once, Run Anywhere™», η οποία συνοψίζει την ιδέα ότι το ίδιο software θα πρέπει να τρέχει σε διαφορετικά είδη υπολογιστών, λειτουργικών συστημάτων, palmtops, κινητών τηλεφώνων, τηλεοράσεων, gadgets κ.λ.π μηχανημάτων και συσκευών.

Δημιουργήθηκε από την εταιρία Sun Microsystems [\[L41\]](#) το 1995 και σχεδόν αμέσως κυριάρχησε στον κόσμο των δικτυωμένων υπολογιστών. Έχει το “look and feel” της γλώσσας C++, με τη διαφορά ότι είναι πιο εύκολη στην εκμάθηση, χωρίς αυτό να σημαίνει ότι μαθαίνεται και σε ελάχιστο χρόνο. Η Java μπορεί να χρησιμοποιηθεί για την ανάπτυξη εφαρμογών, τόσο σε επίπεδο ενός και μόνο υπολογιστή, όσο και σε επίπεδο πολλαπλών servers και clients σε ένα δίκτυο. Μπορεί επίσης να χρησιμοποιηθεί για την υλοποίηση μικρών modules εφαρμογών ή applets σε μία web σελίδα. Το τελευταίο χαρακτηριστικό επιτρέπει στον χρήστη να αλληλεπιδράσει με την web σελίδα σε επίπεδο client. Όλοι οι μεγάλοι browsers διαθέτουν πλέον τη δική τους JVM για τέτοιου είδους requests, ενώ όλοι οι μεγάλοι κατασκευαστές λειτουργικών συστημάτων διαθέτουν Java compiler στο λειτουργικό που παρέχουν.

2.1.2 Βασικά χαρακτηριστικά

Αναμφισβήτητα η Java αποτελεί μια μικρή επανάσταση στο χώρο των γλωσσών προγραμματισμού. Ήρθε να καλύψει τις ανάγκες ανάπτυξης λογισμικού την εποχή της ραγδαίας εξάπλωσης του Internet [\[L6\]](#) και του www [\[L7\]](#), [\[L8\]](#). Από τα βασικά χαρακτηριστικά της είναι:

Αντικειμενοστρεφής: Πρόκειται για την πλέον object-oriented γλώσσα προγραμματισμού, με ομοιότητες εντολών με τη C++, αλλά και πολλά διαφορετικά νέα στοιχεία. Ένα αντικείμενο¹ είναι σε θέση να εκμεταλλευτεί το γεγονός ότι ανήκει σε μία κλάση και να κληρονομήσει χαρακτηριστικά και ιδιότητες, με τη μορφή κώδικα πάντα, κοινά για ολόκληρη την κλάση. Τη διέπουν όλες οι βασικές αρχές του αντικειμενοστρεφούς προγραμματισμού, όπως:

- ο **ιεραρχία τύπων/κλάσεων, κληρονομικότητα:** Η κληρονομικότητα βασίζεται, όπως αναφέρθηκε, στην ιδέα του να μπορούμε να δώσουμε εξειδικευμένα χαρακτηριστικά σε κάποια αντικείμενα, χωρίς τα τελευταία να χάνουν τις γενικότερες ιδιότητες που θα πρέπει να τα διακρίνουν. Αν θεωρήσουμε, λοιπόν, ένα δένδρο κλάσεων, κάθε κλάση κληρονομεί από την πατρική της όλες τις instance variables και τις μεθόδους της. Σε αυτές, η θυγατρική *υποκλάση* είναι δυνατόν να προσθέσει τις δικές της, εξειδικεύοντας έτσι την πατρική *υπερκλάση*.
- ο **πολλαπλή κληρονομικότητα:** Η περίπτωση όπου μία υποκλάση κληρονομεί λειτουργίες και χαρακτηριστικά από περισσότερες της μίας υπερκλάσεις.
- ο **πολυμορφισμός:** Πολυμορφισμός είναι το γεγονός ότι διάφορα αντικείμενα ή/και κλάσεις ολόκληρες αντιδρούν διαφορετικά στο ίδιο μήνυμα.

¹ Ένα αντικείμενο είναι μια οντότητα στη μνήμη η οποία περιέχει δεδομένα και μεθόδους μέσω των οποίων μπορούμε να αλλάξουμε τα δεδομένα ή να επικοινωνήσουμε με το αντικείμενο. Ο κώδικας που ορίζει ένα αντικείμενο λέγεται κλάση του αντικειμένου αυτού. Η κλάση χρησιμοποιείται για την κατασκευή πανομοιότυπων αντικειμένων-αντιγράφων, ενώ τα τελευταία λέγονται στιγμιότυπα (instances) της κλάσης.

- ο **υπέρβαση:** Είναι η περίπτωση όπου κάποιο μήνυμα υλοποιείται με διαφορετική μέθοδο στην κορυφή της ιεραρχίας από ότι σε κάποιο χαμηλότερο επίπεδο.
- ο **υπερφόρτωση:** Δύο μηνύματα σε διαφορετικές κλάσεις συμφωνούν στο interface τους.

Δημιουργία ανεξάρτητων εφαρμογών και applets

Interpreted γλώσσα: Τα προγράμματα που παράγονται είναι portable στο δίκτυο. Κι αυτό γιατί ο μεταγλωττιστής της Java δεν παράγει απ' ευθείας εκτελέσιμο κώδικα από τον πηγαίο, αλλά μια μορφή ενδιάμεσου ψευδοκώδικα (bytecode), ο οποίος έχει πολύ μικρό μέγεθος (της τάξεως κάποιων Kbytes) και για να τρέξει απαιτεί την ύπαρξη μιας JVM, ή αλλιώς ενός interpreter, που θα αναλάβει την μετατροπή του ενδιάμεσου ψευδοκώδικα σε εκτελέσιμο. Αυτό ακριβώς είναι και το ιδιαίτερο χαρακτηριστικό που επιτρέπει στα προγράμματα αυτά να τρέχουν σε οποιοδήποτε μηχανήμα διαθέτει μία JVM, ανεξάρτητα του αν αυτό είναι υπολογιστής ή τοστιέρα! Καταργείται λοιπόν έτσι η έννοια του platform-specific κώδικα.

Distributed: Είναι δυνατόν διαφορετικά τμήματα ενός προγράμματος να συλλέγουν από διαφορετικά sites στο δίκτυο, δίνοντας έτσι μια κατανεμημένη διάσταση στον προγραμματισμό.

Robust: Ο κώδικας της Java είναι «κλειστός» με τα αντικείμενα να μην μπορούν να θεωρήσουν references σε εξωτερικά σε αυτά ή άλλα αντικείμενα δεδομένα. Αυτό εξασφαλίζει ότι μια εντολή δεν μπορεί να περιέχει την διεύθυνση δεδομένων μιας άλλης εφαρμογής ή/και του ίδιου του λειτουργικού συστήματος, πράγμα που θα εξανάγκαζε την εφαρμογή ή το λειτουργικό να τερματιστεί απότομα (crash). Η JVM πραγματοποιεί έναν αριθμό ελέγχων σε κάθε αντικείμενο προκειμένου να εξασφαλιστεί η ακεραιότητα των προγραμμάτων.

- ο **Ασφάλεια:** Είναι σχεδιασμένη έτσι ώστε να ελαχιστοποιείται η πιθανότητα προσβολής του συστήματος του χρήστη από κάποιο κακόβουλο applet.
- ο **multithreaded:** Υποστηρίζει εγγενώς τη χρήση πολλαπλών threads, ακόμα και σε συστήματα με έναν επεξεργαστή. Προκειμένου να το πετύχει αυτό, η Java υλοποιεί το δικό της χρονοδρομολογητή (scheduler), ο οποίος αναλαμβάνει το multithreading με πλήρη διαφάνεια προς το χρήστη.
- ο **multimedia εφαρμογές:** Τέλος, η Java είναι η νούμερο ένα γλώσσα στην υλοποίηση multimedia εφαρμογών, τόσο σε desktop επίπεδο, όσο και σε επίπεδο δικτύου, λόγω κυρίως των ευκολιών που παρέχει σε αυτό τον τομέα στους προγραμματιστές, με τις ιδιαίτερα πλούσιες βιβλιοθήκες και την ευελιξία της.

2.1.3 Εργαλεία

Τα κυριότερα εργαλεία-προγράμματα του Java Development Kit (JDK), έκδοση 1.4 είναι:

- ο **javac:** ο Java compiler. [\[L9\]](#)
- ο **java:** ο Java interpreter. [\[L10\]](#)
- ο **javaw:** like java, αλλά για το λειτουργικό σύστημα Windows μόνο. [\[L11\]](#)
- ο **jdb:** ο Java debugger. [\[L12\]](#)
- ο **javap:** ο Java disassembler. [\[L13\]](#)
- ο **javadoc:** πρόγραμμα για αυτόματη κατασκευή documentation για εφαρμογές. [\[L14\]](#)

2.1.4 Περιγραφή βασικών εννοιών

- κλάση

Ένα τυπικό παράδειγμα ορισμού κλάσης από την εφαρμογή μας είναι:

```
public class SceneFrame extends Frame implements ActionListener  
{ }
```

Γενικά, η δήλωση των πεδίων/μεθόδων στην κλάση μπορεί να ακολουθεί όποια σειρά θέλει ο προγραμματιστής, ενώ μπορεί να γίνει και χρήση τους πριν ακόμα δηλωθούν.

- *access specifiers*

Υπάρχουν τέσσερις διαφορετικές δηλώσεις access specifiers:

- ο **public**, που σημαίνει ότι το συγκεκριμένο πεδίο ή μέθοδος μπορεί να χρησιμοποιηθεί από οποιονδήποτε, ακόμα και από ξένο αντικείμενο.
- ο **protected**, που σημαίνει ότι το πεδίο/μέθοδος που το έχει είναι ορατό μόνο στις μεθόδους της κλάσης αυτής και των υποκλάσεων της.
- ο **private**, που σημαίνει ότι το πεδίο/μέθοδος είναι ορατό μόνο στις μεθόδους αυτής της κλάσης.
- ο **τίποτα**, που σημαίνει ότι το πεδίο/μέθοδος είναι public, αλλά μόνο για το συγκεκριμένο package.

- *modifier*

Είναι δεσμευμένες λέξεις της Java που προσδίδουν συγκεκριμένες επιπλέον ιδιότητες ή/και χαρακτηρίζουν τα πεδία/μεθόδους που τα έχουν. Οι τρεις πιο σημαντικοί είναι:

- ο **final**: ορίζει ότι το πεδίο είναι σταθερά και όχι μεταβλητή, για τις μεθόδους ορίζει ότι απαγορεύεται το override, ενώ για τις κλάσεις ορίζει ότι δεν επιτρέπεται subclassing.
- ο **static**: το πεδίο/μέθοδος που το έχει είναι μοναδικό για όλο το στιγμιότυπο της κλάσης.

- ο **synchronized**: για μεθόδους σημαίνει ότι το αντικείμενο αυτό είναι threadsafe.

- μεταβλητές

Οι μεταβλητές (πεδία) των κλάσεων/αντικειμένων που έχουν δηλωθεί ως μεταβλητές ενός απλού τύπου δεδομένων (**int**, **float**, κ.λ.π.) ακολουθούν το τυπικό format μεταβλητών που είναι γνωστό από τις κλασικές διαδικαστικές γλώσσες προγραμματισμού. Οι μεταβλητές που έχουν δηλωθεί ως μεταβλητές που δείχνουν σε αντικείμενα, όπως π.χ. από την εφαρμογή μας:

```
private Scene scene_;
```

είναι στην πραγματικότητα δείκτες στο αντικείμενο που τους ανατίθεται.

- εντολές

Είναι πανομοιότυπες με τις κλασικές εντολές μιας διαδικασιακής γλώσσας προγραμματισμού, όπως η C, με τη διαφορά ότι στη Java μπορούμε να δηλώσουμε μεταβλητές σε οποιοδήποτε σημείο ενός block κώδικα, αρκεί η δήλωση της μεταβλητής να προηγείται της χρήσης της.

- exceptions

Τέλος, σε περίπτωση που συμβεί κάποιο λάθος σε ένα πρόγραμμα Java, πετάγεται μία εξαίρεση, με αποτέλεσμα τον τερματισμό του thread που συνέβηκε το λάθος και την εκτύπωση κάποιου μηνύματος (error message). Παράδειγμα τέτοιας συμπεριφοράς από το σύστημά μας:

```
try
{
    // System.out.println ("scene URL: " + url);
    scene_.readScene (new URL (getDocumentBase (), url));
}
catch (MalformedURLException e)
{
    System.err.println ("VRMLbro: invalid or missing sceneURL
parameter <" + url + ">");
}
```

Ειδικοί χειριστές εξαιρέσεων (exception handlers) στο πρόγραμμά μας μπορούν να πιάνουν μια εξαίρεση και να φροντίζουν έτσι ώστε το πρόγραμμα να ανανήψει από το λάθος. Προκειμένου να πραγματοποιηθεί αυτό, περικλείουμε τον επίμαχο προς έγερση εξαίρεσης κώδικα σε try-catch blocks, όπως παραπάνω.

2.1.5 Java - Javascript

Μία κοινή, τέλος, παρανόηση που γίνεται στο χώρο των γλωσσών προγραμματισμού είναι η σύγχυση της Java της Sun Microsystems με τη Javascript [\[L16\]](#) της Netscape [\[L17\]](#). Εκτός όμως από το πρώτο συνθετικό, οι δύο γλώσσες παρουσιάζουν σημαντικότερες διαφορές. Η Javascript επεξεργάζεται σε υψηλότερο επίπεδο, είναι τρομακτικά ευκολότερη στην εκμάθηση και φυσικά ουδεμία σχέση έχει με αντικειμενοστραφή προγραμματισμό, αποτελώντας μία ιδιαίτερα εύχρηστη scripting γλώσσα.

Ακόμα, *μόνο* η Java μπορεί να χρησιμοποιηθεί για την ανάπτυξη standalone εφαρμογών, καθώς και mini-εφαρμογών, των applets. Η Javascript είναι ιδιαίτερα διαδεδομένη για τη διάδραση με τους χρήστες ενός web site και στην συντριπτική της πλειοψηφία βρίσκεται ενσωματωμένη μέσα σε σελίδες html [\[L18\]](#).

2.2 VRML:

2.2.1 Περιγραφή

Η VRML (Virtual Reality Modeling Language) [\[L1\]](#), [\[L2\]](#) είναι μία γλώσσα προγραμματισμού για την περιγραφή τρισδιάστατων (3-D) ακολουθιών εικόνων, με δυνατότητες αλληλεπίδρασης αυτών με τον χρήστη. Με τη χρήση της VRML, ο προγραμματιστής είναι σε θέση να υλοποιήσει μια ακολουθία εικόνων, με την οποία ο χρήστης μπορεί να αλληλεπιδράσει με το να τις παρατηρεί (view), να τις μετακινεί (move), να τις περιστρέφει (rotate) και γενικά να «εισάγεται» σε μία τρισδιάστατη σκηνή (3-D scene). Για παράδειγμα, μπορεί κανείς να παρατηρεί ένα δωμάτιο ενός κτιρίου και χρησιμοποιώντας κατάλληλα χειριστήρια να

μετακινεί το δωμάτιο σαν να επρόκειτο για πραγματική περιπλάνησή του μέσα σε αυτό στον πραγματικό χώρο.

Για την ορθή πλοήγηση μέσα σε έναν τέτοιο 3-D κόσμο και για την πρακτική ανάγνωση ενός VRML αρχείου από τον υπολογιστή [\[L19\]](#), είναι απαραίτητη η χρήση ενός VRML browser, όπως η εφαρμογή μας. Γίνεται άμεσα λοιπόν κατανοητή η χρησιμότητά της. Γενικότερα, πρέπει στο σημείο αυτό να σημειωθεί, ότι η ανάπτυξη της VRML ήταν μια δύσκολη και χρονοβόρα διαδικασία. Η πρώτη version του VRML 1.0 specification βγήκε τον Ιανουάριο του 1996, ενώ η τελική version 2.0 καθιερώθηκε αρκετά πρόσφατα. Μεταξύ της πρώτης και της τελευταίας έκδοσης μεσολάβησαν τουλάχιστον 47 πρόχειρες εκδόσεις (drafts).

2.2.2 VRML 2.0 Specification

Η Virtual Reality Modeling Language (VRML) αποτελεί ένα format περιγραφής τρισδιάστατων αλληλεπιδραστικών κόσμων και αντικειμένων. Μπορεί να χρησιμοποιηθεί σε συνδυασμό με το ευρέως διαδεδομένο World Wide Web (www). Μπορεί, επίσης, να χρησιμοποιηθεί για τη δημιουργία τρισδιάστατων αναπαραστάσεων πολύπλοκων σκηνών, όπως εικονογραφήσεις, ορισμοί προϊόντων και παρουσιάσεις εικονικής πραγματικότητας. Τα κύρια σχεδιαστικά της κριτήρια, με βάση τα οποία σχεδιάστηκε και υλοποιήθηκε ως παγκόσμιο standard συνοψίζονται στα εξής:

- *Δυνατότητες συγγραφής (Authorability)*

Να είναι σε θέση κανείς να αναπτύξει γεννήτορες εφαρμογών (application generators) και editors, καθώς και να εισάγει δεδομένα από άλλα εργοστασιακά ή μη formats.

- *Πληρότητα (Completeness)*

Παροχή όλων των πληροφοριών εκείνων, οι οποίες είναι απαραίτητες για μια υλοποίηση και ενός πλήρους συνόλου από features, έτσι ώστε να γνωρίσει ευρεία αποδοχή από την κοινότητα.

- *Συνθετικότητα (Composability)*

Ικανότητα για συνδυαστική χρησιμοποίηση στοιχείων VRML, με αποτέλεσμα την μέγιστη επαναχρησιμοποίηση.

- ο *Επεκτασιμότητα (Extensibility)*

Ικανότητα για προσθήκη νέων στοιχείων σε αυτή.

- ο *Υλοποιησιμότητα (Implementability)*

Ικανότητα υλοποίησης σε μια πλειάδα συστημάτων.

- ο *Δυναμική για πολλαπλούς χρήστες (Multi-user potential)*

Η VRML δεν θα πρέπει να αποκλείει την υλοποίηση της σε πολυχρηστικά περιβάλλοντα.

- ο *Ορθωγωνιότητα (Orthogonality)*

Τα στοιχεία της VRML θα πρέπει να είναι ανεξάρτητα το ένα από το άλλο, εκτός και αν οι όποιες εξαρτήσεις είναι καλά ορισμένες και ξεκάθαρα δομημένες.

- ο *Απόδοση (Performance)*

Τα στοιχεία πρέπει να είναι σχεδιασμένα με έμφαση στην αλληλεπιδραστική συμπεριφορά και απόδοση σε μια ποικιλία υπολογιστικών πλατφόρμων.

- ο *Scalability*

Τα στοιχεία της VRML πρέπει να είναι σχεδιασμένα για εξαιρετικά μεγάλες δομές.

- ο *Πρακτική προτυποποίησης (Standard practice)*

Μόνο τα στοιχεία εκείνα τα οποία αντικατοπτρίζουν την υπάρχουσα πρακτική θα πρέπει να προτυποποιηθούν.

- ο *Ποιότητα Δόμησης (Well-structured)*

Ένα στοιχείο VRML θα πρέπει να έχει ένα καλώς ορισμένο interface και ένα ξεκάθαρο και απλό στόχο.

Στα κύρια χαρακτηριστικά της VRML ανήκουν η δυνατότητά της για αναπαράσταση στατικών και animated αντικειμένων, καθώς και το γεγονός ότι είναι σε θέση να συμπεριλάβει hyperlinks προς άλλα μέσα, όπως ήχος, ταινίες και εικόνες. Διαφόρων ειδών proprietary και μη browsers για VRML είναι διαθέσιμοι ως προϊόντα, γεγονός που αποτέλεσε και έναυσμα για τη συγκεκριμένη εργασία, ενώ υπάρχει και μια πληθώρα εργαλείων συγγραφής VRML αρχείων. Η VRML

υποστηρίζει ένα μοντέλο επεκτασιμότητας, το οποίο επιτρέπει τον ορισμό νέων αντικειμένων και μία διαδικασία registration, έτσι ώστε να επιτρέψει στις κοινότητες των προγραμματιστών να αναπτύξουν επεκτάσεις του βασικού standard. Υπάρχει ένα mapping μεταξύ στοιχείων VRML και ευρέως διαδεδομένων τρισδιάστατων χαρακτηριστικών και features από Application Programmer Interfaces (APIs).

Η VRML 2.0 ορίζει επίσης ένα format αρχείων για έναν τρισδιάστατο κόσμο με inline εικόνες και hyperlinks [L20], [L21]. Ένα αρχείο VRML αποτελείται από ένα δένδρο κόμβων, όπου κάθε κόμβος είναι είτε ένας κόμβος Σχήμα (Shape node), όπως ένα σετ από πολύγωνα, μία σφαίρα, ένας κύβος, κ.α., είτε ένας κόμβος Ιδιότητας (Property node), όπως ένα σύνολο από Normals, από Materials, ένα Light ή ένας Transformation, είτε, τέλος, ένας κόμβος Ομάδα (Group node), που περιέχει κόμβους παιδιά. Σε ένα VRML file, ένας κόμβος αποτελείται από ένα προαιρετικό όνομα, έναν τύπο κόμβου και μια λίστα από πεδία (fields) ή ιδιότητες (properties). Προφανώς, όταν δοθεί ένα όνομα σε κάποιο κόμβο, αυτός μπορεί να χρησιμοποιηθεί ως έχει στην συνέχεια.

Τέλος, αξίζει να σημειωθεί ότι οι παραπάνω προδιαγραφές δεν καθορίζουν αυστηρά και επακριβώς τους μηχανισμούς εκείνους, οι οποίοι παρέχονται από έναν VRML browser για την πλοήγηση μέσα στον χώρο· χώρος ο οποίος με τη σειρά του περιγράφεται όπως αναλύθηκε ήδη από ένα VRML αρχείο. Η μοναδική αλληλεπίδραση, η οποία περιγράφεται ρητά και σαφώς από τις προδιαγραφές αυτές, είναι η συμπεριφορά του κόμβου WWWAnchor, ο οποίος όταν πατηθεί με το ποντίκι προκαλεί το φόρτωμα ενός άλλου VRML αρχείου. Συνήθως, δε, στον τελικό χρήστη προσφέρονται περισσότεροι του ενός τρόποι εξέτασης και κίνησης μέσα στην VRML σκηνή, παρόλο που η αλληλεπίδραση υλοποιείται διαφορετικά σε καθένα browser.

2.3 *Borland JBuilder:*

Πρόκειται για ένα περιβάλλον ανάπτυξης Java εφαρμογών λογισμικού, πλήρως συμβατού με την J2EE. Περιέχει έναν πολύ καλό Java debugger για το debugging των παραγόμενων εφαρμογών, προσφέρει μια πληθώρα editors και προγραμματιστικών visual εργαλείων για την ανάπτυξη εφαρμογών και components, όπως τα applets, τα JavaBeans, οι JavaServer Pages (**JSP**), τα **servlets**, τα αντικείμενα **CORBA** και τα Enterprise JavaBeans (**EJB**). Ιδιαίτερως δε, προσφέρει υποστήριξη για την τεχνολογία των Business Components for Java (**BC4J**), για την δημιουργία scalable, υψηλής απόδοσης εφαρμογών Internet. Παρέχει στο σύνολό του ένα στέρεο και αποδοτικό user interface για τους προγραμματιστές, όμως αυτό που κάνει το συγκεκριμένο εργαλείο ιδιαίτερα ξεχωριστό είναι το προγραμματιστικό framework που παρέχει.

Στα πλεονεκτήματά του τοποθετούμε την υποστήριξη διασύνδεσης βάσεων δεδομένων, όπως της Oracle, καθώς είναι πλήρως συμβατό με το SQL-92 JDBC standard,. Επίσης, σημαντική είναι και η ευελιξία που παρέχεται στον προγραμματιστή να κάνει deploy τις ίδιες εφαρμογές σε Java Server Pages, EJBs και CORBA components χρησιμοποιώντας τον ίδιο κώδικα γι' αυτές. Με τον τρόπο αυτό ο βασικός κώδικας μιας εφαρμογής καθίσταται περισσότερο επαναχρησιμοποιήσιμος από ποτέ και για διάφορα είδη πλατφόρμων, από web browsers μέχρι και ασύρματες συσκευές χειρός.

Τέλος, στις προδιαγραφές του προϊόντος αναγράφονται ως δυνατές πλατφόρμες εγκατάστασής του οι: Windows NT®, Windows 2000® και Windows XP® με δυνατότητες deployment σε οποιοδήποτε σύστημα υποστηρίζει Java. Οι minima απαιτήσεις του σε hardware είναι: Pentium 200 MHz, 96 MB RAM και 280 MB χώρος στο σκληρό δίσκο.

3

Περιγραφή Περιβάλλοντος

Το θέμα της παρούσας διπλωματικής εργασίας, ο σχεδιασμός, δηλαδή, η κατασκευή και η υλοποίηση ενός VRML browser άπτεται του ενδιαφέροντος της σχετικής κοινότητας των Γραφικών Υπολογιστών, καθώς και ορισμένων εμπορικών προϊόντων του χώρου. Στη συνέχεια θα γίνει μια προσπάθεια για την περιγραφή της ίδιας της εφαρμογής, του ερευνητικού περιβάλλοντος στο οποίο ανήκει και κυρίως της χρηστικότητάς της και της ένταξής της στην περιοχή των τρισδιάστατων απεικονίσεων, μέσω της εκπόνησης κατάλληλων συγκρίσεων.

Η όλη διπλωματική εργασία αναφέρεται στην ανάπτυξη ενός **Virtual Reality Modelling Language (VRML)** browser. Η εφαρμογή διαβάζει VRML αρχεία, σύμφωνα με το VRML 2.0 standard και επιτρέπει την εξερεύνηση των αρχείων αυτών με τη χρήση ενός αριθμού από navigation controls. Η εκτέλεσή της είναι δυνατή σε οποιαδήποτε πλατφόρμα υπολογιστών και σε οποιοδήποτε λειτουργικό σύστημα, μεταξύ των Win3.11(!), Windows 95, Windows 98, Windows Me, Windows 2000, Windows XP, Linux, Solaris, SGI IRIX κ.ο.κ..

Οι γενικές προδιαγραφές της γλώσσας VRML περιγράφουν ένα format αρχείων για τρισδιάστατους κόσμους με χρήση hyperlinks προς άλλους κόσμους. Δεδομένη θεωρείται πλέον και η χρήση τέτοιων αρχείων-κόσμων (και) πάνω από το δίκτυο. Οι παραπάνω προδιαγραφές επιτρέπουν την περιγραφή κόσμων, οι οποίοι περιλαμβάνουν σχήματα που αποτελούνται από πολλά πολύγωνα με χρώματα και υφές, φωτισμούς συγκεκριμένων χρωμάτων και εντάσεων, καθώς και texture maps πάνω σε επιφάνειες.

Στην παρούσα εφαρμογή έχει υλοποιηθεί ένα μεγάλο υποσύνολο από το VRML 2.0 standard [L2], [L19], γεγονός που επιτρέπει την ανάγνωση, την απεικόνιση και την εξερεύνηση τέτοιων VRML αρχείων. Κατά την εξερεύνηση τέτοιων κόσμων, αυτοί αποδίδονται (rendered) στις 3 διαστάσεις με ρεαλιστικούς φωτισμούς, σκιάσεις, texture mappings και όπου αυτό καταστεί απαραίτητο, χρήση animation. Στην περίπτωση, δε, που προσπελαστεί κάποιο ακατάλληλο αρχείο, το οποίο περιέχει τυχόν ανεκπλήρωτα χαρακτηριστικά, κατάλληλα μηνύματα λάθους εμφανίζονται στον χρήστη κατά την διάρκεια του αρχικού φορτώματος της εφαρμογής.

Ο browser που υλοποιήθηκε παρέχει στον τελικό χρήστη έναν αριθμό από user controls για την πλοήγηση στο χώρο. Αυτά περιλαμβάνουν χρήση του πληκτρολογίου και του ποντικιού για real-time κίνηση μέσα στον VRML κόσμο, καθώς και διάφορα κατάλληλα dialog boxes, τα οποία σκοπό έχουν να λειτουργήσουν βοηθητικά για την αλλαγή των διαθέσιμων παραμέτρων του browser, ακόμα και για τον πιο ανίδεο χρήστη.

3.1 Ερευνητική Προσέγγιση Περιβάλλοντος

Παράλληλα με τα ήδη αναφερθέντα στο πρότερο κεφάλαιο χαρακτηριστικά υλοποίησης της εφαρμογής, πραγματοποιήθηκε και μιας μεγάλης κλίμακας ερευνητική εργασία γύρω από το αντικείμενό της. Η ερευνητική δουλειά που επιτελέστηκε, συνάμα με την συγγραφή του κώδικα ο οποίος αποτέλεσε την καθεαυτό εφαρμογή του VRML browser, ήταν καθόλα πρωτοποριακή στο είδος της. Πραγματοποιήσαμε μεγάλης κλίμακας έρευνα και επισταμένη αναζήτηση στα μεγαλύτερα παγκόσμια επιστημονικά sites, όπως το Ieeexplore της IEEE [\[L38\]](#), το site της Association for Computing Machinery (ACM) [\[L37\]](#) και το ευρέως γνωστό και χρησιμοποιούμενο από την επιστημονική κοινότητα ScienceDirect [\[L40\]](#).

Είναι γεγονός, όμως, ότι ο χώρος των τρισδιάστατων γραφικών και η κοινότητα των προγραμματιστών του, δεν έχουν έως σήμερα ασχοληθεί ιδιαίτερα με την σύζευξη των τεχνολογιών των γλωσσών προγραμματισμού Java και VRML. Όπως μπορεί κανείς να παρατηρήσει και διεξοδικότερα στο επόμενο κεφάλαιο 3.3, τα κύρια συμπεράσματα που απορρέουν από τη βιβλιογραφική έρευνα που πραγματοποιήθηκε, εδραιώνουν την άποψη περί επιστημονικής ένδειας του τομέα αυτού. Κατά τη διάρκεια των αναζητήσεών μας παρατηρήσαμε μεν πληθώρα ομάδων οι οποίες δραστηριοποιούνταν στη γενικότερη περιοχή της τρισδιάστατης απεικόνισης, χωρίς ωστόσο καμία να εμβαθύνει στη συσχέτιση Java και VRML που στην παρούσα εργασία εισάγουμε και υλοποιούμε. Οι προσπάθειες που έχουν πραγματοποιηθεί έως τώρα επικεντρώνονται κυρίως στο parsing αρχείων VRML από προγράμματα Java, εν είδη μεταγλωττιστή, χρησιμοποιώντας μετέπειτα την εξαγόμενη πληροφορία εσωτερικά μέσα στις ίδιες τις Java εφαρμογές. Μέχρι τώρα δεν έχει γίνει γνωστή κάποια standalone εφαρμογή VRML browser γραμμένη αποκλειστικά σε Java. Για το λόγο αυτό, μια τέτοια προσέγγιση θεωρείται πρωτοποριακή, μιας και συγκεντρώνει όλα τα πλεονεκτήματα που συνεπάγεται η χρήση της Java ως γλώσσας προγραμματισμού.

Ειδικότερα στον τομέα του parsing VRML αρχείων από προγράμματα γραμμένα στη γλώσσα προγραμματισμού Java έχουν πραγματοποιηθεί διάφορες απόπειρες κατά καιρούς από διάφορες ομάδες εργασίας σε ποικίλα ευρωπαϊκά ερευνητικά ιδρύματα, με κυριότερες και άξιες αναφοράς τις εξής:

- Την δουλειά του Filip Sixta [\[L34\]](#) στο Department of Computer Science and Engineering του Czech Technical University της Πράγας στην Τσεχία. Πρόκειται για έναν πλήρη VRML parser γραμμένο σε Java ο οποίος αποτέλεσε την βάση για έναν VRML browser για τυφλούς ανθρώπους.
- Την υλοποίηση του pw-VRML parser στο Graz University of Technology της Αυστρίας. Πρόκειται και εδώ για έναν πλήρως συμμορφωμένο προς το VRML 2.0 standard VRML parser.
- Το έργο της ομάδας του Ινστιτούτου Fraunhofer στο Darmstadt της Γερμανίας [\[L35\]](#) σχετικά με δυνατότητες διαδραστικής τρισδιάστατης οπτικοποίησης δεδομένων προσομοίωσης με χρήση animation και διαδραστικών περιπάτων (walk-throughs) σε VRML κόσμους.
- Το έργο των Ingo Soetebier και Ralf Dörner [\[4\]](#), επίσης στο Ινστιτούτο Fraunhofer σχετικά με ανάκτηση VRML περιεχομένου από Object-Oriented βάσεις δεδομένων.

Αξίζει, δε, να σημειωθεί ότι αναφορές στην VRML πραγματοποιούνται αρκετά συχνά στα μεγάλα περιοδικά και συνέδρια του χώρου των γραφικών, όπως τα ευρέως γνωστά **ACM SIGGRAPH**, **ACM Crossroads**, **Transactions of ACM on Graphics**, **IEEE Computer**, απ' όπου είναι ευνόητο ότι αντλήθηκαν ποικίλες πληροφορίες για την συγγραφή αυτής της διπλωματικής εργασίας στο σύνολό της.

Βασίζόμενοι, λοιπόν, στις παραπάνω ερευνητικές προσεγγίσεις, επεκτείνοντας την έρευνά μας στο κομμάτι της σύζευξης των εννοιών VRML και Java, και συνεισφέροντας και τη δική μας φαντασία και δημιουργικότητα, κατορθώσαμε να παράγουμε ένα συνολικό έργο με τη μορφή μιας εφαρμογής

ενός VRML browser, πλήρως λειτουργικού και ανεξάρτητου από πλατφόρμες υλοποίησης.

3.2 Σύγκριση με Εμπορικές Εφαρμογές

Στο πεδίο της τρισδιάστατης απεικόνισης υπάρχουν διάφορες κατηγορίες εμπορικών ή μη εφαρμογών, καθεμία με τα πλεονεκτήματα και τα μειονεκτήματά της. Ειδικότερα στην περιοχή της VRML υπάρχει μια πληθώρα από VRML browsers, μερικοί εκ των οποίων θεωρούνται ποιοτικά καλοί και συμβατοί με το standard, όπως οι:

- **Cosmo Player**: ο VRML 2.0 browser της SGI [\[L26\]](#), [\[L27\]](#).
- **Casus Presenter**: ο VRML 97 browser του Fraunhofer Institute for Computer Graphics [\[L28\]](#), [\[L29\]](#).
- **Microsoft VRML 2.0 Viewer**: ο browser της Microsoft [\[L30\]](#).
- **Whirlwind 3D**: VRML browser για Macintosh [\[L31\]](#).
- **Community Place**: ο πρώτος VRML 2.0 browser στο δίκτυο [\[L32\]](#), [\[L33\]](#).

Οι περισσότεροι από τους παραπάνω VRML browsers έχουν χρησιμοποιηθεί από εκατομμύρια χρήστες σε όλον τον κόσμο (χαρακτηριστικά αναφέρεται ότι μόνο τα downloads του Cosmo Player ξεπερνάνε τα 20.000.000!) και συνεπώς είναι αναγνωρισμένης αξίας και ποιότητας. Αποτελούν, δηλαδή, ένα ικανό και θεμιτό μέτρο σύγκρισης για την εφαρμογή VRMLbro που αναπτύχθηκε, στα πλαίσια, βέβαια, των γνωστών δυνατοτήτων ενός μη εμπορικού προϊόντος.

Για το σκοπό αυτό, πραγματοποιήσαμε μια σειρά από πειράματα και δοκιμές, όσον αφορά παραμέτρους χρόνου έναρξης, χρόνου απόκρισης και χρονικής διάρκειας rendering VRML κόσμων. Παρατηρήσαμε, όμως και καταγράψαμε και στοιχεία, αφενός μεν μη μετρήσιμα, όπως η ευχρηστία της εφαρμογής και η user-friendliness της, αφετέρου δε, ιδιαίτερα σημαντικά και χαρακτηριστικά για την ένταξη της εφαρμογής στο χώρο της τρισδιάστατης

απεικόνισης. Αναμφισβήτητα, από το σύνολο των παραπάνω μετρήσεων είναι σε θέση κανείς να εξάγει ασφαλή συμπεράσματα για την ίδια την ποιότητα της εφαρμογής ως υλοποίηση.

Κατασκευάσθηκαν και χρησιμοποιήθηκαν για το σκοπό αυτό συγκεκριμένα αρχεία VRML κόσμων, οι οποίοι έγιναν αντικείμενο μελέτης και όλες οι εφαρμογές εκτελέσθηκαν στον ίδιο υπολογιστή, από πλευράς hardware, έτσι ώστε να έχουμε συγκρίσιμα μεταξύ τους αποτελέσματα. Το μέγεθος των αρχείων ποίκιλε από αρχεία της τάξεως λίγων bytes, που περιελάμβαναν μεμονωμένα τρισδιάστατα γεωμετρικά αντικείμενα, μέχρι αρχεία μερικών Kbytes που απεικόνιζαν τρισδιάστατα αντικείμενα, όπως π.χ. μία λάμπα, και έως αρχεία πολλών Kbytes ή και Mbytes, τα οποία απεικόνιζαν πολύπλοκους και πολλαπλούς τρισδιάστατους κόσμους, όπως π.χ. τα δωμάτια ενός ορόφου του παλαιού κτιρίου του Χαροκόπειου Πανεπιστημίου ή απεικονίσεις εκατοντάδων τρισδιάστατων αντικειμένων μέσα στον ίδιο VRML κόσμο.

Στα μετρήσιμα μεγέθη συμπεριλάβαμε τα ακόλουθα:

- ο **Χρόνος εγκατάστασης** του browser στον υπολογιστή.
- ο **Χρόνος έναρξης** της εφαρμογής του VRML browser, από τη στιγμή που θα «τρέξει» το εκτελέσιμο αρχείο ο τελικός χρήστης.
- ο **Χρόνος απόκρισης** για το άνοιγμα ενός VRML κόσμου.
- ο Χρόνος κατά το **rendering** ενός VRML κόσμου.

Όλες οι μετρήσεις πραγματοποιήθηκαν στο ακόλουθο υπολογιστικό σύστημα:

	Φάση ανάπτυξης λογισμικού συστήματος	Τελική φάση ελέγχου ορθής λειτουργίας & benchmarking
Processor	Intel Pentium 4 2,0 GHZ	Intel Pentium III 1,0 GHZ
RAM	768MB RAM	320MB RAM
Disk space	120GB	30GB
VGA card	Geforce 4 64MB	ATI Mobility 128
Monitor	Sony CPD-200GS 17"	TFT 15"
OS	Windows XP®	Windows 2000®

Πίνακας I

Ειδικότερα, τα αποτελέσματα των μετρήσεών μας αποδίδονται αναλυτικότερα και παραστατικότερα στους παρακάτω πίνακες, όπου παρουσιάζονται αντίστοιχα οι παραπάνω χρόνοι και για τα τρία είδη αρχείων (μικρού μεγέθους: sympan.wrl (46KB), μεσαίου μεγέθους: sympanBIG.wrl (1847KB) και μεγάλου μεγέθους: sympanHUGE.wrl (18.331KB)):

VRML browser	t install	t έναρξης	t απόκρισης	rendering	OS
VRMLbro	~ 2 secs	2441 msec	83 msec	27 msec	all
Cosmo Player	~ 15 secs	753 msec + IE current	147 msec	23 msec	Windows 98/2000/XP/ SGI IRIX
Community Place	~ 10 secs	359 msec + Netscape current	160 msec	25 msec	Windows 98/2000/XP

Πίνακας II

VRML browser	t install	t έναρξης	t απόκρισης	rendering	OS
VRMLbro	~ 2 secs	2441 msec	298 msec	109 msec	all
Cosmo Player	~ 15 secs	753 msec + IE current	329 msec	84 msec	Windows 98/2000/XP/ SGI IRIX
Community Place	~ 10 secs	359 msec + Netscape current	365 msec	111 msec	Windows 98/2000/XP

Πίνακας III

VRML browser	t install	t έναρξης	t απόκρισης	rendering	OS
VRMLbro	~ 2 secs	2441 msec	2122 msec + out of memory!	-	all
Cosmo Player	~ 15 secs	753 msec + IE current	3512 msec	-	Windows 98/2000/XP/ SGI IRIX
Community Place	~ 10 secs	750 msec + Netscape current	plain text!	-	Windows 98/2000/XP

Πίνακας IV

Στο σημείο αυτό θα πρέπει να σημειωθεί ότι οι χρόνοι εγκατάστασης των τριών browsers έχουν αποτυπωθεί κατά προσέγγιση, καθώς μεσολαβούν και άλλοι παράγοντες, πέρα από την ποιότητα υλοποίησης του λογισμικού, πως ένας σημαντικός αριθμός από clicks μεταξύ των διαφόρων steps εγκατάστασης, τα οποία προφανώς αυξάνουν τον χρόνο. Συνολικά, ο VRMLbro συμπεριφέρεται άψογα όσον αφορά την δυνατότητά του να εκτελείται σε οποιοδήποτε λειτουργικό σύστημα. Παρουσιάζει, βέβαια τις κλασικές αδυναμίες τις Java κατά

την πρώτη εκτέλεση από τον χρήστη, όπου ο χρόνος έναρξης είναι συγκριτικά με τον ανταγωνισμό αρκετά μεγάλος, ενώ στους κρίσιμους χρόνους απόκρισης κατά την προσπέλαση ενός VRML κόσμου και κατά το rendering των κινήσεων μέσα σε κάποιο κόσμο, εμφανίζει ιδιαίτερα θετικές και ενθαρρυντικές για την όλη προσπάθεια τιμές, που κυμαίνονται στα επίπεδα των διαδεδομένων εμπορικών πακέτων της περιοχής των VRML browsers, στην περίπτωση που έχουμε λογικά μεγέθη VRML κόσμων. Όταν το μέγεθος του VRML κόσμου αγγίζει πάρα πολύ μεγάλα μεγέθη, τόσο οι εμπορικοί όσο και ο VRMLbro browser αντιμετωπίζουν δυσκολίες, αναφορικά με την μνήμη που απαιτούν για το «άνοιγμα» των κόσμων. Συνεπώς, τα παραπάνω αποτελέσματα, κατατάσσουν το σύστημά μας στα πιο γρήγορα και με άμεση απόκριση συστήματα για ένα πλήθος περιπτώσεων. Πιστεύουμε, δε, ότι το κατατάσσουν και στα πιο φιλικά προς τον χρήστη, λόγω της απλότητας που το διέπει.

3.3 Παρεμφερείς Ερευνητικές Εργασίες

Είναι αδιαμφισβήτητο γεγονός ότι στο χώρο των γραφικών γενικότερα και της κοινότητας της VRML ειδικότερα, δεν έχει πραγματοποιηθεί ανάλογης κλίμακας εργασία, τόσο όσον αφορά την ερευνητική δραστηριότητα στο συγκεκριμένο θέμα της σύζευξης των δύο τεχνολογιών, όσο και όσον αφορά την πρακτική εφαρμογή και υλοποίηση του όλου περιβάλλοντος. Προς επίρρωση του ισχυρισμού αυτού έρχονται όλες οι σχετικές ερευνητικές εργασίες του χώρου, οι οποίες ασχολούνται με διάφορα παρεμφερή θέματα και αντικείμενα, αλλά κανένα με αυτό καθαυτό το αντικείμενο της παρούσης διπλωματικής εργασίας.

Ειδικότερα, παρατηρούμε μια πληθώρα εργασιών σχετικών με Java και VRML, στο επίπεδο όμως της ξεχωριστής διαχείρισης, αφενός του client interface από την Java, αφετέρου των τρισδιάστατων μοντέλων από την VRML. Η απεικόνιση των τελικών μοντέλων δεν πραγματοποιείται από κάποιον integrated Java browser, ούτε καν από κάποιο Java enabled plugin, αλλά επαφίεται σε ξεχωριστά προγράμματα απεικόνισης VRML, τα οποία απαιτείται να είναι εγκατεστημένα στον υπολογιστή που εργάζεται ο χρήστης και εκτελούνται και οι εν λόγω υλοποιήσεις των εργασιών αυτών. Χαρακτηριστική

εργασία στον τομέα αυτόν αποτελεί η δημοσίευση με τίτλο: «*A Java/CGI approach to developing a geographic virtual reality toolkit on the Internet*» των Bo Huang και Hui Lina από το Chinese University of Hong Kong [P5]. Η εργασία έχει ως κύριο στόχο την σχεδίαση ενός toolkit για την διαδραστική προετοιμασία εικονικών περιβαλλόντων από υπάρχουσες βάσεις Γεωγραφικών Συστημάτων Συντεταγμένων (GIS) και στηρίζεται ακριβώς στην παραπάνω ξεχωριστή προσέγγιση των δύο τεχνολογιών. Οι τρισδιάστατες σκηνές μετατρέπονται σε VRML μοντέλα και παραδίδονται σε κάποιον κλασικό www browser προς απεικόνιση και πλοήγηση με το κατάλληλο plugin.

Ανάλογο ύφος και προσέγγιση διέπει και τις εργασίες: «Support for collaborative design reasoning in shared virtual spaces» [P6], «An Internet virtual reality collaborative environment for effective product design» [P7], «Using VRML and JAVA to visualize 3D algorithms in computer graphics education» [P8] και «Web-based three-dimensional geo-referenced visualization» [P9]. Αναλυτικότερα, στην πρώτη από αυτές, γίνεται λόγος για ένα ασύγχρονο συνεργατικό σύστημα, το IDT, το οποίο επιτρέπει στους συνεργαζόμενους φορείς να αναπτύσσουν επιχειρήματα γύρω από τρισδιάστατα μοντέλα πάνω από το Internet χρησιμοποιώντας διαφόρων ειδών διαγραμματικά σημεία, δυναμικές προσομοιώσεις, γεωμετρικές υποκατάστατα σχεδίων και σχολιασμό κειμένων. Το IDT βασίζεται για την απεικόνιση των μοντέλων στην VRML, ενώ ένα interface βασισμένο στη Java καθοδηγεί την αλληλεπιδραστική συμπεριφορά του, συμπεριλαμβανομένης της κατασκευής και αναπαραγωγής των γραφικών σχολιασμών και της διαχείρισης των νηματικών συζητήσεων και των file input/output.

Η δεύτερη εργασία πραγματεύεται επίσης ένα βασισμένο στο Internet συνεργατικό περιβάλλον εικονικής πραγματικότητας, το οποίο αναπτύχθηκε με χρήση της Java και της VRML. Ειδικότερα, αξιοποιήθηκαν οι δυνατότητες της VRML για τη δημιουργία ενός πολυχρηστικού αλληλεπιδραστικού περιβάλλοντος, χρησιμοποιώντας την βοήθεια που παρέχει στον τομέα αυτό η γλώσσα προγραμματισμού Java, σχετικά με το External Authoring Interface

(EAI). Η προσέγγιση αυτή ξεφεύγει από τις μέχρι τώρα κλασικές θεωρήσεις συνεργασίας της Java με τη VRML και η εντύπωση που αποκομίζει κανείς είναι ότι δεν εμπίπτει σαφώς στην ερευνητική περιοχή της παρούσης εργασίας.

Η τρίτη από τις παραπάνω εργασίες ασχολείται με την δημιουργία applets με αλληλεπιδραστικά τρισδιάστατα μοντέλα προς χρήση τους στην εκπαιδευτική διαδικασία εκμάθησης διαφόρων αλγορίθμων γραφικών. Και σε αυτή την εργασία πραγματοποιείται αναφορά στον χώρο των VRML browsers και σημειώνεται ότι οι περισσότεροι από αυτούς δεν συναντώνται ως κάποιο ξεχωριστό περιβάλλον ή εφαρμογή, αλλά αποτελούν κάποιο μικρής έκτασης plugin σε υπάρχοντες δημοφιλείς web browsers. Συμπέρασμα το οποίο αποτελεί και την βάση της παρούσης εργασίας, καθώς και το κίνητρο για την υλοποίηση του Java-based VRMLbro. Τέλος, στον τομέα της υλοποίησης η συγκεκριμένη εργασία επιλέγει την τετριμμένη και σαφώς περιοριστική λύση της χρησιμοποίησης του EAI, για τη διαχείριση της VRML. Το EAI καθορίζει τον τρόπο με τον οποίο ο VRML browser και τα περιεχόμενα του θα διαχειριστούν από κάποιο πρόγραμμα, το οποίο «τρέχει» στο ευρύτερο περιβάλλον το οποίο περιέχει τον browser (γι' αυτό και ο όρος external στον τίτλο του), όπως π.χ. είναι ένα Java applet το οποίο «τρέχει» μέσα στην ίδια σελίδα με έναν VRML browser, μέσα σε έναν web browser.

Η τελευταία από τις παραπάνω εργασίες πραγματεύεται διάφορες προσεγγίσεις για την υλοποίηση web-based τρισδιάστατης γεωσχεσιακής οπτικοποίησης. Εστιάζει, δε, στη σχέση μεταξύ πολυδιάστατων συνόλων δεδομένων και εφαρμογών, καθώς και μεταξύ «ελαφρών» και «ενισχυμένων» υλοποιήσεων, ενώ παρουσιάζονται τα πλεονεκτήματα και τα μειονεκτήματα κάθε μεθόδου, μεταξύ των οποίων συγκαταλέγεται και ο συνδυασμός Java και VRML, στο επίπεδο απλής συνεργασίας για ανάπτυξη real-time εφαρμογών και όχι σύζευξης τεχνολογιών.

Παρατηρούμε, λοιπόν, ότι σε όλες τις παραπάνω προσεγγίσεις η VRML χρησιμοποιείται μεν σε συνδυασμό με την Java προς κοινό σκοπό, αλλά η λειτουργικότητα της μίας τεχνολογίας δεν εμπλέκεται με την λειτουργικότητα και τα χαρακτηριστικά της άλλης. Σχεδόν πάντα. Τόσο στις παραπάνω εργασίες όσο

και γενικότερα (βλ. [P10], [P11], [P12]) η VRML απλά κατασκευάζει τα όποια υπολογιστικά τρισδιάστατα μοντέλα, η Java αναλαμβάνει την διάδραση με τα ενδιαμέσα interfaces και κάποιο εξωτερικό plugin φροντίζει για την ορθή απεικόνιση των αποτελεσμάτων στην οθόνη του τελικού χρήστη. Η χρήση της Java όπως είναι φυσικό είναι εκτεταμένη και υπερκαλύπτει σε πολλές περιπτώσεις την VRML, καθώς κατασκευάζει π.χ. τους γραφικούς σχολιασμούς ή διαχειρίζεται τις νηματικές συζητήσεις και το file input/output.

3.4 Πλεονεκτήματα έναντι άλλων προσπαθειών

Από τα παραπάνω γίνεται εμφανές ότι η εφαρμογή που αναπτύχθηκε έχει σημαντικά πλεονεκτήματα έναντι άλλων ερευνητικών ή μη προσπαθειών, οι οποίες έχουν λάβει χώρα τα τελευταία χρόνια στο χώρο αυτό. Όσον αφορά τις ερευνητικές απόπειρες, αυτές συνοψίζονται, όπως είδαμε, στο κεφάλαιο 3.1 και δεν θα μας απασχολήσουν περαιτέρω. Στο χώρο των εμπορικών εφαρμογών, ο πήχης του «ανταγωνισμού» έχει τεθεί αρκετά ψηλά, από τις δραστηριοποιούμενες στο χώρο εταιρίες, οι οποίες έχουν παράγει διάφορα προϊόντα για την κάλυψη των αναγκών του.

Βασικότερο όλων, καθίσταται το πλεονέκτημα της εκτέλεσής της ανεξάρτητα από την υπολογιστική πλατφόρμα στην οποία βρίσκεται ο χρήστης, σε συνδυασμό με τον μηδαμινό χρόνο που απαιτείται για την εγκατάστασή της. Με άλλα λόγια η ολική μεταφερσιμότητα της εφαρμογής και η ευκολία κατά την εγκατάσταση, δύο παράγοντες που παίζουν ιδιαίτερα σημαντικό ρόλο για έναν browser, αν και δεν είναι σε θέση να μετρηθούν με ποσοτικά κριτήρια.

Ακολουθούν, βεβαίως οι πολύ καλές τιμές απόκρισης και rendering κατά την επεξεργασία των VRML κόσμων, οι οποίες είναι ανάλογες με τα εμπορικά πακέτα του χώρου, όπως παρατηρήσαμε από τα αποτελέσματα της προηγούμενης παραγράφου. Επίσης, όλη η εφαρμογή είναι ιδιαίτερα «ελαφριά» από απόψεως κατανάλωσης υπολογιστικών πόρων και απλή προς τον τελικό χρήστη, σε αντίθεση με τους εμπορικούς browsers που πολλές φορές καταπονούν σε μεγάλο

βαθμό την υπολογιστική ισχύ ενός σύγχρονου υπολογιστή, ενώ σε μία περίπτωση, όπως είδαμε, απαιτείται και η ύπαρξη ήδη ενεργού Internet Explorer για την ορθή λειτουργία του VRML browser.

Τέλος, βασικότατο πλεονέκτημα της εφαρμογής VRMLbro αποτελεί η δυνατότητα render VRML κόσμων μέσω δικτύου, μιας και δεν συναντήσαμε τέτοια δυνατότητα σε κανέναν από τους εμπορικούς browsers που εξετάσαμε. Λόγω της ιδιαιτερότητας της τεχνολογίας Java, οποιαδήποτε δικτυακή τοποθεσία νοείται από την εφαρμογή ως μία απλή τοποθεσία όπου θα βρει το αρχείο VRML προς επεξεργασία. Αυτό έχει ως αποτέλεσμα να παρουσιάζει η εφαρμογή την ίδια επιτυχή συμπεριφορά, τόσο χρονικά, όσο και υπολογιστικά, όπως κατά το άνοιγμα τοπικών αρχείων VRML.

3.5 Ομάδες Χρηστών ενός τέτοιου Περιβάλλοντος

Ένα περιβάλλον, όπως ο VRMLbro browser, είναι σε θέση να παρέχει χρήσιμες υπηρεσίες σε τρία διαφορετικά είδη χρηστών, ανάλογα πάντα με τις ανάγκες τους. Ειδικότερα:

- **Φοιτητές:** Οι φοιτητές είναι ίσως οι κύριοι ενδιαφερόμενοι, αφού αυτοί καλούνται πολλές φορές να πλοηγηθούν σε τρισδιάστατα περιβάλλοντα και κόσμους, στα πλαίσια εργασιών ή/και εκπαιδευτικών σεμιναρίων τους. Προφανώς ευχρηστία, απλότητα και ταχύτητα του συστήματος αποτελούν εκ των ων ουκ άνευ για αυτού, καθώς και ευκολία πρόσβασης.
- **Καθηγητές-Διδακτικό προσωπικό:** Οι διδάσκοντες θα ήθελαν από την πλευρά τους να έχουν στη διάθεσή τους ένα απλό και εύχρηστο εργαλείο για την πλοήγηση σε τέτοιου είδους κόσμους, ώστε να είναι πιο εύκολη και αποδοτική η ανάθεση εργασιών και άλλων σχετικών εκπαιδευτικών λειτουργιών. Προφανώς και σε αυτή την περίπτωση, όλα τα παραπάνω απαιτούν απρόσκοπτη και γρήγορη λειτουργία του browser σε οποιοδήποτε υπολογιστικό σύστημα.

- ο **Απλοί χρήστες:** Στα πλαίσια των λοιπών εμπορικών εφαρμογών, ένα τέτοιο περιβάλλον θα έβρισκε μεγάλη απήχηση μεταξύ των απλών χρηστών του Internet και των τρισδιάστατων γραφικών, καθότι είναι καθόλα εύχρηστο και «ελαφρύ».
- ο **Developers:** Οι developers του χώρου των γραφικών θα βρουν στον VRMLbro browser ένα εργαλείο για την απρόσκοπτη απεικόνιση της δουλειάς τους, εύκολα και γρήγορα και κυρίως σε οποιαδήποτε υπολογιστική πλατφόρμα αυτοί χρησιμοποιούν, χωρίς την παραμικρή απαίτηση αλλαγών εκ μέρους τους στο σύστημά τους. Με τον τρόπο αυτό, απαλλάσσονται από έναν μεγάλο φόρτο εργασίας κατά τη διάρκεια της δουλειάς τους και μειώνουν σημαντικά τις καθυστερήσεις στην εργασία τους στον τομέα αυτό.

3.6 Τελικό Ζητούμενο

Το τελικό ζητούμενο στο συγκεκριμένο πρόβλημα είναι η επιστημονική έρευνα και υλοποίηση, σχετικά με την σχεδίαση, κατασκευή και ανάπτυξη, ενός πλήρους περιβάλλοντος πλοήγησης σε τρισδιάστατους VRML κόσμους, το οποίο με όσο πιο σαφή και εύκολο τρόπο γίνεται να βοηθά τους εκάστοτε δυνητικούς χρήστες του, είτε αυτοί είναι φοιτητές, είτε καθηγητές, είτε απλοί χρήστες, είτε developers του χώρου, να αντλήσουν την κατάλληλη πληροφορία και να αλληλεπιδράσουν με αυτούς. Απαιτείται, δε, συνεργασία του περιβάλλοντος αυτού με τα υπάρχοντα υπολογιστικά συστήματα που χρησιμοποιούνται από αυτούς και πλήρη συμβατότητα με ανάλογα άλλα περιβάλλοντα του χώρου.

4

Υλοποίηση Περιβάλλοντος

Η ανάπτυξη του περιβάλλοντος λογισμικού στηρίχθηκε σε μια ποικιλία εργαλείων και υλοποιήσεων, τα οποία και παρουσιάζονται αναλυτικότερα παρακάτω:

4.1 Επιλογή Πλατφόρμας και Εργαλείων

Η σωστή επιλογή πλατφόρμας και εργαλείων συγγραφής και ανάπτυξης λογισμικού είναι ιδιαίτερα σημαντική για την ανάπτυξη μιας τέτοιας εφαρμογής ως προς την αποδοτικότητα, την επεκτασιμότητα και την μεταφερσιμότητά της. Συνάμα, απλοποιείται κατά πολύ η διαδικασία του προγραμματισμού και ο προγραμματιστής επικεντρώνεται και εστιάζει στη σχεδίαση και τη βελτιστοποίηση του παραγόμενου κώδικα.

Θεωρώντας ότι έχουμε καλύψει τα πεδία των ουσιωδών επιλογών της κύριας γλώσσας προγραμματισμού στο κεφάλαιο 2.1.1 (**Java**), θα επικεντρωθούμε στο κεφάλαιο αυτό στα εργαλεία ανάπτυξης λογισμικού που

χρησιμοποιήθηκαν κατά την ανάπτυξη της εφαρμογής του Java VRML browser μας. Ακολουθώντας, λοιπόν, τα υψηλά standard όλων των σύγχρονων Java-based εφαρμογών επιλέχθηκε ο Borland Jbuilder 8.0 [\[L24\]](#) για την συγγραφή του κώδικα, καθώς και το Ultraedit 10.10a [\[L25\]](#) ως editor.

4.2 Εργαλεία Ανάπτυξης Περιβάλλοντος

Το προγραμματιστικό framework που χρησιμοποιήθηκε για την ανάπτυξη του περιβάλλοντος VRMLbro ήταν εκείνο που παρείχε το εργαλείο Borland JBuilder, υλοποιημένο σε Java και XML. Επιτρέπει, δε, την ανάπτυξη scalable, multi-tier εφαρμογών Java από επαναχρησιμοποιούμενα components κώδικα. Η συγκεκριμένη τεχνική μας παρέχει ως προγραμματιστές ένα σύνολο «έξυπνων» blocks κατασκευής λογισμικού, τα οποία συνεργάζονται μεταξύ τους για να διευθύνουν κοινά facilities, όπως:

- ο Να συγγράφουν και να τεστάρουν παραγωγικά την business logic σε components, τα οποία και αλληλεπιδρούν απ' ευθείας π.χ. με μια δικτυακή τοποθεσία ή μια βάση δεδομένων.
- ο Να επαναχρησιμοποιούν εύελικτα την business logic μέσα από πολλαπλές όψεις δεδομένων.
- ο Να προσπελαίνουν ικανοποιητικά και να ενημερώνουν/ανανεώνουν τα views από servlets, JSPs και thin Java Swing clients, όπως ο VRMLbro.
- ο Να παραμετροποιούν εύκολα την functionality της εφαρμογής σε στρώματα, χωρίς να απαιτείται μεταβολή της ήδη ανεπτυγμένης εφαρμογής.

Με όλα τα παραπάνω, καθίσταται σαφές ότι ο προγραμματιστής ωθείται στην υλοποίηση των λύσεων, με αποτέλεσμα να ελαττώνεται σημαντικά το ρίσκο, ο χρόνος και το κόστος παραγωγής μιας εφαρμογής του μεγέθους του VRMLbro.

4.3 Υλοποίηση Java:

Τέλος, το δυσκολότερο και παραγωγικότερο κομμάτι της διπλωματικής εργασίας αποτέλεσε η συγγραφή περίπου 11.000 γραμμών κώδικα Java, μη συμπεριλαμβανομένου ενός μεγάλου αριθμού υπάρχουσων βιβλιοθηκών ή imported αρχείων. Η Java χρησιμοποιήθηκε ως επί το πλείστον για την ορθή, λιτή και αισθητικά προσεγμένη απεικόνιση των χαρακτηριστικών της εφαρμογής μας. Η χρήση της Java ως γλώσσας προγραμματισμού προσδίδει, όπως είναι φυσικό, ύψιστη ευελιξία στο όλο σύστημα, μεταφερσιμότητα και βέλτιστη απόδοση, με εύκολα παραμετροποιησίμους παράγοντες και δυνατότητες εύκολης αλλαγής υποκείμενων καίριων στοιχείων, όπως οι δυνατότητες πλοήγησης που προσφέρονται στον εκάστοτε χρήστη, χωρίς να απαιτείται ιδιαίτερα δύσκολη προγραμματιστική παρέμβαση κατά την περίπτωση αλλαγών ή /και προσθηκών στην όλη εφαρμογή μελλοντικά.

Επίσης, είναι προφανές ότι ανάμεσα στους πρωταρχικούς στόχους κατά την υλοποίηση του όλου εγχειρήματος ήταν και η κατά το δυνατόν επαναχρησιμοποίηση κώδικα κατά τη φάση ανάπτυξης της εφαρμογής, προκειμένου να επιταχυνθεί η παραγωγική διαδικασία και η όλη εξέλιξή της. Με τη χρήση της Java το παραπάνω επετεύχθη σε πολύ μεγάλο βαθμό. Παρατηρούμε, δε, στο αρχείο υλοποίησης του κώδικα ότι είναι ιδιαίτερα εμφανής η επαναχρησιμοποίηση κώδικα στο όλο εγχείρημα. Για περισσότερες πληροφορίες ο αναγνώστης καλείται να ανατρέξει στον κώδικα της εφαρμογής που επισυνάπτεται σε cd-rom.

4.3.1 Παράδειγμα κώδικα υλοποίησης

```
/*
 * VRMLbro v1.0
 * VRML 2.0 browser, implemented in
 * Java
 *
 *
 * Phivos Mylonas, 2003
 */

package ntua.vrml.vrmlbro;

import iicm.vrml.pw.*;
import iicm.vrml.vrmlbro.pwdat.*;
import iicm.ge3d.GE3D;
import iicm.utils3d.Camera;
import iicm.utils3d.Ray;
import iicm.utils3d.Font3D;
import iicm.widgets.*;
import java.awt.*;
import java.applet.*;
import java.io.*;
import java.net.*;
import java.util.*;

/**
 * <p>Title: VRMLbro</p>
 * <p>Description: </p>
 * <p>Copyright: Copyright (c)
 * 2003</p>
 * <p>Company: </p>
 * @author Phivos Mylonas
 * @version 1.0
 */

public class Scene
    implements ApplyColour
{
    // dashboard - navigation mode
    protected int db_movemode_ =
DB_LOOK;
    public final static int DB_LOOK=0;
    public final static int DB_WALK=1;
    public final static int DB_PAN=2;
    public final static int DB_ZOOM=3;
    public final static int
DB_ROTATE=4;
    public final static int
DB_TRANSLATE=5;
    public final static int
DB_NUMNAVMODES = 6;
    public final static int
DB_BEHAVIOUR=10;
    public final static int
DB_INTERACTION=11;
    public final static int UPDATE_MENU
= 0x1;
    public final static int UPDATE_ICON
= 0x2;
    public final static int UPDATE_ALL
= 0x3;

    // colors
    public final static int
COLBACKGROUND = 0;

    public final static int
COLHEADLIGHT = 1;
    public final static int COLAMBIENT
= 2;
    public final static int NUMCOLORS =
3;
    private int[] color_ = { //
0xRRGGBB
0x000000, // background
0xffffffff, // headlight
0x333333 // ambient (OpenGL
default: 0.2)
};

    // on/auto/off options
    public final static int
TRISTATE_OFF = 0;
    public final static int TRISTATE_ON
= 1;
    public final static int
TRISTATE_AUTO = 2;

    // present fonts
    public final static int HELVETICA =
0;
    public final static int TIMES = 1;

    // the navigationbar
    private NavigationBar navbar_;

    protected GroupNode root;
    private Builder builder = new
Builder ();
    private Drawer drawer = new Drawer
();
    private Picker picker = new Picker
();
    private boolean firstdraw_;
    private Applet applet_;
    protected Frame frame_;
    private MenuItem[] menuitems_;
    private SceneCanvas canvas_ = null;
    private Vector sensors_ = null;
    private Hashtable nodenames_ =
null;

    // timestamp of current frame
    (seconds since 1970, double
precision)
    private double timestamp_ = 0.0;
    /** get frame timestamp */
    public double getTimestamp () {
return timestamp_; }

    public void toggleShowframerate ()
    {
        showframerate_ = !showframerate_;
        clearStatusMessage ();
    }
    private boolean showframerate_ =
false;
    public float framespersecond_ =
0.0f;

    // drawing mode, interaction
```

```

    private int drawmode_ =
GE3D.ge3d_texturing;
    private int intdrawmode_;
    private boolean interact_ = false;

    public void setInteraction (boolean
flag) { interact_ = flag; }
    public boolean interactionRelevant
()
    { return (intdrawmode_ >= 0 &&
intdrawmode_ != drawmode_); }
    public void drawingMode (int mode,
boolean interactive)
    { if (interactive)
        intdrawmode_ = mode;
      else
        drawmode_ = mode;
    }
    public int curDrawingMode ()
    { return ((interact_ &&
intdrawmode_ >= 0) ? intdrawmode_ :
drawmode_); }
    public int normalDrawingMode ()
    { return drawmode_; }
    public int interDrawingMode ()
    { return intdrawmode_; }

    // viewing camera (includes
Viewpoint management)
    private ViewCamera viewcam_ = new
ViewCamera ();
    /** reset camera */
    protected void resetView () {
viewcam_.reset (); }
    /** level view */
    protected void levelView () {
viewcam_.levelize (); }
    /** untilt view */
    protected void untiltView () {
viewcam_.untilt (); }
    /** set camera via GE3D */
    protected void setCamera () {
viewcam_.setCamera (winaspect_); }
    /** get camera */
    public Camera getCamera () {
return viewcam_; }
    /** get near clipping plane */
    float getNearClip () { return
viewcam_.getNearClip (); }
    /** get far clipping plane */
    float getFarClip () { return
viewcam_.getFarClip (); }
    /** set near clipping plane */
    public void setNearClip(float near)
{viewcam_.setNearClip(near); }
    /** set far clipping plane */
    public void setFarClip(float far)
{viewcam_.setFarClip(far); }

    /** zoom out of the scene */
    protected void zoomOut ()
    {
viewcam_.zoomOut(Math.max(2.0f,viewca
m_.getDistance()/2));
    }
    /** zoom into the scene */
    protected void zoomIn ()
    {

```

```

        viewcam_.zoomOut(-
Math.max(2.0f,viewcam_.getDistance()/
2));
    }

    /** default Viewpoint (first one
encountered in file) */
    private Viewpoint defview_ = null;
    /** Viewpoint encountered in build
*/
    void hasViewpoint (Viewpoint vp)
    {
        if (defview_ == null) // only of
interest in initial build
            defview_ = vp;
    }

    /** set frame (for dialogs) */
    protected void setFrame (Frame
frame) { frame_ = frame; }

    /** set menuitems (created by
frame) */
    protected void setMenus (MenuItem[]
items) { menuitems_ = items; }

    /** get menuitems (to update
states) */
    MenuItem[] getMenus () { return
menuitems_; }

    private ImageButton[] navbuttons_;
    private ImageButton behavbutton_,
interactbutton_;

    public void redraw ()
    { // repaint active canvas
      if (canvas_ != null)
        canvas_.repaint ();
    }

    // statusbar, progress indicator
    private StatusBar statusbar_;
    private Progress progress_;

    void setProgressbar (StatusBar
statusbar, Progress progress)
    {
        statusbar_ = statusbar;
        progress_ = progress;
    }

    public void statusMessage (String
label, int fnum)
    {
        if (statusbar_ != null)
            statusbar_.setLabel (0, fnum,
label);
        if (navbar_ != null)
            navbar_.setStatusText(label);
    }

    public void clearStatusMessage ()
    {
        statusMessage ("", 0);
    }

    public void workingMessage (String
label)
    {
        if (statusbar_ != null)

```

```

        statusBar_.setLabel (1, 0,
label);
        navbar_.setStatusText(label);
    }

    public void workingState (boolean
flag)
    {
        if (statusBar_ != null)
        {
            if (flag)
                statusBar_.last ();
            else
                statusBar_.first ();
        }
    }

    /** set progress (0.0 to 1.0) */
    public void setProgress (float
level)
    {
        if (progress_ != null)
            progress_.setValue (level);
    }

    // window aspect
    private float winaspect_ = 1.33f;
    /** set window aspect */
    public void setWinAspect (float
val) { winaspect_ = val; }
    /** get window aspect */
    public float getWinAspect () {
return winaspect_; }

    // scene center
    private float[] center_ = { 0.0f,
0.0f, 0.0f };
    /** set center of scene */
    public void setCenter (float[] c)
{ center_ = c; }
    /** get center of scene */
    public float[] getCenter () {
return center_; }

    // picking purpose (set flags in
Hitpoint)
    // public final static int PICK_POI
= 0;

    // lighting
    private boolean haslight_ = false;
    /** scene has light source; no need
for headlight */
    public void setHasLight () {
haslight_ = true; }
    /** is there a light source? */
    public boolean hasLightSource () {
return haslight_; }
    private boolean headlight_ = false;
    /** toggle headlight */
    public void toggleHeadlight () {
headlight_ = !headlight_; }
    /** get headlight flag */
    public boolean getHeadlight () {
return headlight_; };

    /** get a color (RGB values) */
    public int getColor (int i) {
return color_[i]; }
    /** change a color (RGB values) */
    public void setColor (int i, int
rgb) { color_[i] = rgb; }

    // dialogs
    private DialogFileOpen dlgfileopen_
= null;
    private DialogOpenLocation
dlgopenlocation_ = null;
    String lastfilename_ = null;
    URL lasturl_ = null;
    URL baseurl_ = null;
    private DialogAbout dlgabout_ =
null;
    private DLGColourChoose
dlgcolchooser_ = null;
    private DialogSettings dlgsettings_
= null;

    // backfaceculling
    private int backfaceculling_ =
TRISTATE_AUTO;
    public int backfaceCulling ()
{ return backfaceculling_; }
    public void setBackfaceCulling (int
val)
    { backfaceculling_ = val; }

    // line antialiasing
    private int supportsantialiasing_ =
0; // not known before GE3D is
loaded
    private int lineantialiasing_ = 0;
    public int getLineAntialiasing ()
{ return lineantialiasing_; }
    public void toggleLineAntialiasing
()
    { // no polygon anti-aliasing yet
(requires ordered polygons for good
results)
        if (supportsantialiasing_ != 0)
        {
            lineantialiasing_ =
GE3D.AA_LINES - lineantialiasing_;
            GE3D.antialiasing
(lineantialiasing_);
        }
    }

    // texture mipmapping
    private int texturemipmapping_ = 0;
    public int getTextureMipmapping ()
{ return texturemipmapping_; }
    public void setTextureMipmapping
(int quality)
    { texturemipmapping_ = quality; }

    // lighting
    private int lighting_ =
TRISTATE_AUTO;
    public int getLighting ()
{ return lighting_; }
    public void setLighting (int val)
{ lighting_ = val; }

    // texture lighting
    private boolean texlighting_ =
true;
    public boolean getTexLighting ()
{ return texlighting_; }
    public void setTexLighting (boolean
val)
    { texlighting_ = val; }

    // materials on/off

```

```

    private boolean materials_ = true;
    public boolean materials ()
    { return materials_; }
    public void setMaterials (boolean
val)
    { materials_ = val; }

    // texture transparency
    private boolean texturetransparen_
= true;
    public
        boolean
getTextureTransparency ()
    { return texturetransparen_; }
    public void setTextureTransparency
(boolean val)
    { texturetransparen_ = val; }

    // transparency method
    private int transpmethod_ =
GE3D.TRANSP_STIPPLE;
    public int getTranspMethod ()
    { return transpmethod_; }
    public void setTranspMethod (int
val)
    {
        if (val != transpmethod_)
        {
            transpmethod_ = val;
            GE3D.hint
(GE3D.HINT_TRANSPARENCY, val);
            redraw ();
        }
    }

    // settings
    private int quadslices_ = 16;
    public void setQuadslices (int val)
    {
        if (val != quadslices_)
        {
            quadslices_ = val;
            redraw ();
        }
    }
    public int getQuadslices ()
    { return quadslices_; }

    // behavior (sensors, events and
scripts)
    private boolean behavior_ = true;
    public void toggleBehavior ()
    {
        behavior_ = !behavior_;
        ((CheckboxMenuItem)
menuitems_[MenuDef.M_Behaviour]).setS
tate (behavior_);

    navbar_.setCurrentButtonsVisible(DB_B
EHAVIOUR,behavior_);

    }
    /** check if behavior is enabled */
    public boolean getBehavior () {
return behavior_; }
    /** check if behavior is running
(continuous repaints driven by
TimeSensor) */
    public boolean behavior () {
return (behavior_ && (sensors_ !=
null)); }

```

```

    // interaction: mouse handles
sensors/anchors instead of navigation
    private boolean interaction_ =
false;
    public void toggleInteraction ()
    {
        interaction_ = !interaction_;
        ((CheckboxMenuItem)
menuitems_[MenuDef.M_Interaction]).se
tState (interaction_);

    navbar_.setCurrentButtonsVisible(DB_I
NTERACTION,interaction_);

    }
    /** check if interaction is enabled
(otherwise navigation) */
    public boolean getInteraction () {
return interaction_; }

    /** add a SensorNode to a list of
sensors which are to check before a
redraw */
    public void addSensor (Node sens)
    {
        if (sensors_ == null)
            sensors_ = new Vector ();
        sensors_.addElement (sens);
    }

    /**
     * get current time in
<em>seconds</em> (double precision)
    since Jan 1 1970 00:00:00 GMT
     */
    public static double currentTime ()
    {
        return
            (((double)
System.currentTimeMillis ()) /
1000.0);
    }

    // 3D fonts of the landscape
    public static Font3D font_;
    public Font3D[] fonts_;
    public int current_font_ =
Scene.TIMES;
    public boolean font_enabled_ =
true;
    Font3D[] get3DFonts () { return
fonts_; }
    Font3D get3DFont () { return font_;
}
    // set an activ font
    public void set3DFont ()
    {
        fonts_ = VRwave.fonts;
        font_ = fonts_[current_font_];
    }

    private synchronized void clear ()
    {
        root = null; // destroy old
scene
        firstdraw_ = true; // need build
on next draw
        lastfilename_ = null;
        lasturl_ = null;
        baseurl_ = null;
        nodenames_ = null;
    }

```

```

        haslight_ = false;

        sensors_ = null; // clear list
        of sensors which are to check

        viewcam_.clear ();
        viewcam_.reset ();
        if (canvas_ != null)
            canvas_.reset ();

        clearStatusMessage (); // clear
        old info (e.g. anchor descriptions)

        drawer.clearThreads ();
        builder.clearThreads ();
        if (frame_!=null)
            frame_.setTitle("VRwave");

        viewpoints_.removeAllElements();
        viewpoint_number_=0;
        activ_viewpoint_=null;
        if (menuitems_!=null)
        {
            /*                      menuitems_
            [MenuDef.M_PrevViewpoint].setEnabled
            (false);
            menuitems_
            [MenuDef.M_NextViewpoint].setEnabled
            (false);

            canvas_.setViewpoints(viewpoints_);*/

            navbar_.toggleNextViewState(false);
            navbar_.togglePrevViewState(false);
        }

        } // clear

        // use with care
        public void clearScene ()
        {
            clear ();
        }

        /**
         * get a new, empty scene
         */
        protected void newScene () { clear
        (); }

        /**
         * readScene - read VRML stream
         from file
         */
        public synchronized void readScene
        (String filename, String baseurl)
        {
            clear (); // clear old data,
            reset flags
            lastfilename_ = filename;

            if (baseurl == null)
                baseurl = "file:" + (new File
                (filename)).getAbsolutePath ();

            try
            { baseurl_ = new URL (baseurl);

```

```

            } catch (MalformedURLException e)
            { }
            // else baseurl_ set to null in
            clear

            try
            {
                parseInput (new VRMLparser
                (Decompression.filter (filename)),
                filename);
                if (frame_!=null)
                    frame_.setTitle("VRwave -- " +
                    baseurl);
            }
            catch (IOException e)
            {
                System.out.println ("error on
                reading " + filename);
                // e.printStackTrace ();
                System.out.println
                (e.getMessage ()); // just prints
                file name
            }

            } // readScene (filename, baseurl)

            /**
             * readScene - read VRML stream
             from URL
             */
            public synchronized void readScene
            (URL url)
            {
                clear ();
                lasturl_ = url;
                baseurl_ = url;

                String location = url.toString
                ();

                // TODO: handle compressed
                streams

                try
                {
                    parseInput (new VRMLparser
                    (url.openStream ()), location);
                    if (frame_!=null)
                        frame_.setTitle("VRwave -- " +
                        location);
                }
                catch (IOException e)
                {
                    System.out.println ("error on
                    reading " + location);
                    // e.printStackTrace ();
                    System.out.println
                    (e.getMessage ()); // just prints
                    file name
                }
            } // readScene (URL)

            public synchronized void readScene
            (InputStream input, String baseurl,
            String location)
            {

```

```

clear (); // clear old data,
reset flags
// lastfilename_ is null, reload
only possible via URL

try
{
    if (baseurl != null)
        baseurl_ = new URL (baseurl);
} catch (MalformedURLException e)
{ }

if (location == null)
    location = (baseurl != null) ?
baseurl : "input stream";

parseInput (new VRMLparser
(input), location);
if (frame_!=null)
frame_.setTitle("VRwave -- " +
location);

} // readScene (stream)

// do the parsing
private void parseInput (VRMLparser
parser, String location)
{
    root = parser.readStream ();

    if (root != null)
    {
        nodenames_ =
parser.getNodeNames ();

        if (VRwave.verbose)
            System.out.println ("parsing
was ok.");
    }
    else
    {
        System.out.println ("error on
parsing " + location);
        if (parser.getVersion () ==
0.0f)
            System.out.println ("bad
header");
    }

} // parseInput

/**
 * draw the scene
 */
public synchronized void draw ()
{
    if (firstdraw_)
    {
        GE3D.initGE3D ();
        supportsantialiasing_ =
GE3D.antialiasingSupport ();
        if (menuitems_ != null &&
supportsantialiasing_ == 0)
            menuitems_
[MenuDef.M_AAALines].setEnabled
(false);

        defview_ = null;

```

```

        if (VRwave.verbose)
            System.out.println ("****
build ****");
        builder.buildScene (this, root,
baseurl_);

        if (defview_ != null)
        {
            viewcam_.initialViewpoint
(defview_); // becomes active if
first one
            if (menuitems_ != null)
                ((RadioMenuItem) menuitems_
[MenuDef.M_VP1]).setState (true);
        }

        firstdraw_ = false;
    }

    timestamp_ = currentTime ();
    if (behavior_) // handle events
and scripts only when behavior is ON
        if (sensors_ != null)
            for (int i = 0; i <
sensors_.size (); i++)
                ((Sensor)
sensors_.elementAt (i)).evaluate
(timestamp_);

        drawer.drawScene (this,
root);
    } // draw

    void finishedDraw ()
    {
        if (!showframerate_) // might
calculate fps for EAI
            return;

        float drawtime = (float)
(currentTime () - timestamp_); //
seconds
        if (drawtime < 0.01f)
            drawtime = 0.01f;
        int drawsec = (int) drawtime;
        int dsechd = (int) (drawtime *
100.0f) - 100 * drawsec;
        framespersecond_ = (1.0f /
drawtime);
        statusMessage ("last frame: " +
drawsec + (dsechd < 10 ? ".0" :
".") + dsechd +
"s (" + (int) framespersecond_
+ (framespersecond_ < 100 ? " " : "")
+
(framespersecond_ < 10 ? " " :
"") + " fps )", 0); // ###
translator
    }

    public synchronized void
buildInline (Inline node1, GroupNode
node2)
    {
        builder.buildScene (this, node2,
((InlineData) node1.userdata).url_);
        ((InlineData)
node1.userdata).inline_ = (GroupNode)
node2;
        redraw ();
    }

```

```

    }

    public synchronized void buildNode
    (GroupNode node)
    {
        while (!canvas_.hasContext ())
        {
            try {
                Thread.sleep (500);
            } catch (Throwable e) { }
        }
        builder.buildScene (this, node,
        null);
    }

    public synchronized Node pick
    (float fx, float fy, VHitpoint hit)
    {
        if (firstdraw_)
            return null;

        Ray ray = getRay (fx, fy);
        return picker.pickScene (this,
        root, ray, hit, false, false);
    }

    public synchronized Ray getRay
    (float fx, float fy)
    {
        return (viewcam_.viewingRay (fx,
        fy, winaspect_));
    }

    public synchronized Node pick
    (float fx, float fy, VHitpoint hit,
    boolean sensors, boolean keeptrf)
    {
        if (firstdraw_)
            return null;

        Ray ray = viewcam_.viewingRay
        (fx, fy, winaspect_);

        return picker.pickScene (this,
        root, ray, hit, sensors, keeptrf);
    }

    public void activateAnchor (String
    urlstr, String[] params, int
    numparams)
    {
        if (applet_ != null)
        {
            URL url;
            try
            {
                url = new URL (urlstr);
            }
            catch (MalformedURLException e)
            {
                System.err.println ("bad URL:
                \"\" + urlstr + "\"");
                return;
            }

            System.err.println ("activate
            Anchor \"\" + urlstr + "\"");

            String target = null;
            while (numparams-- > 0)

```

```

                if (params
                [numparams].startsWith ("target="))
                {
                    target = params
                    [numparams].substring (7);
                    break;
                }

                AppletContext context =
                applet_.getAppletContext ();
                if (target != null)
                {
                    System.err.println ("anchor
                    target: \" + target);
                    context.showDocument (url,
                    target);
                }
                else
                    context.showDocument (url);

                return;
            }

            urlstr = substChar (urlstr, '\\',
            "%27"); // get a shell safe URL
            encoding
            urlstr = substChar (urlstr, ',',
            "%2c"); // netscape interprets , as
            arg separator in openURL

            String[] command;
            String[] w_command = {
            VRwave.helpurl_ + "viewhtml.bat",
            urlstr };
            String[] u_command = { "/bin/sh",
            "-c", "netscape -remote 'openURL(\" +
            urlstr + \")'" };
            if (System.getProperties
            ().getProperty ("os.name").startsWith
            ("Windows"))
            {
                command = w_command;
                if (MessageDialog.run (frame_,
                "VRwave Security Question",
                "Open '\" + urlstr + '\"?",
                MessageDialog.YESNO,
                MessageDialog.NO
                ) !=
                MessageDialog.YES)
                    return;
            }
            else
            {
                command = u_command;
                if (frame_ != null)
                {
                    frame_.setCursor
                    (Cursor.getPredefinedCursor(Cursor.WA
                    IT_CURSOR));
                }

                if (VRwave.verbose)
                    System.out.println ("netscape
                    remote call:\n" +
                    iicm.vrml.pw.Decompression.cmdToStrin
                    g (command));
                try
                {
                    Process netscape =
                    Runtime.getRuntime ().exec (command);
                    int exitval = netscape.waitFor
                    ();
                    if (exitval != 0)

```



```

        {
            System.err.println ("VRwave:
could not execute " +
iicm.vrml.pw.Decompression.cmdToStrin
g (command) +
"\nnetscape must be running
to handle remote requests");
        }
    }
    catch (Exception e)
    {
        System.err.println ("Error on
netscape remote call: " +
iicm.vrml.pw.Decompression.cmdToStrin
g (command));
        System.err.println ("got
exception: " + e);
    }

    if (frame_ != null)
    {
        frame_.setCursor
(Cursor.getPredefinedCursor(Cursor.DE
FAULT_CURSOR));
    }

    }

    public void showHelpfile (String
topic)
    {
        activateAnchor (VRwave.helpurl_ +
topic, null, 0);
    }

    public static String substChar
(String str, char c, String s)
    {
        int pos = str.indexOf (c);
        if (pos < 0)
            return str;

        StringBuffer buf = new
StringBuffer ();
        int from = 0;
        do
        {
            buf.append (str.substring
(from, pos));
            buf.append (s);
            from = pos + 1;
            pos = str.indexOf (c, from);
        } while (pos >= 0);

        buf.append (str.substring
(from));

        return buf.toString ();
    }

    public void openFile ()
    {
        if (dlgfileopen_ == null)
            dlgfileopen_ = new
DialogFileOpen (frame_);

        DialogFileOpen dlg =
dlgfileopen_;

```

```

        dlg.show ();

        String file = dlg.getFile ();
        if (file == null)
            return;

        String dir =
dlgfileopen_.getDirectory ();
        String filename = dir + file;

        System.out.println ("reading
scene " + filename);
        readScene (filename, null);
    }

    public void openLocation ()
    {
        if (dlgopenlocation_ == null)
            dlgopenlocation_ = new
DialogOpenLocation (frame_);

        DialogOpenLocation dlg =
dlgopenlocation_;
        dlg.show ();

        String url = dlg.getURLString ();
        if (url == null)
            return;
        System.out.println ("opening
location: " + url);

        try
        {
            readScene (new URL (url));
        }
        catch (MalformedURLException e)
        {
            System.err.println ("invalid
URL <" + url + ">");
        }
    }

    public boolean reloadFile ()
    {
        if (lastfilename_ != null)
            readScene (lastfilename_,
(baseurl_ != null) ?
baseurl_.toString () : null);
        else if (lasturl_ != null)
            readScene (lasturl_);
        else
        {
            System.out.println ("no scene
loaded or input not available.");
            return false;
        }

        return true;
    }

    public static void toggleFrame
(Frame f)
    {
        if (f.isVisible ())
            f.setVisible (false);
        else
            f.show ();
    }

    public void toggleColorChooser ()
    {
        if (dlgcolchooser_ == null)

```

```

        dlgcolchooser_ = new
DLGColourChoose
(Translator.L_ColChooser_Targets,
this);
        dlgcolchooser_.setColours
(color_, 0);
        toggleFrame (dlgcolchooser_);
    }

    public void applyColour
(DLGColourChoose dlg)
    {
        dlgcolchooser_.getColours
(color_);
        redraw ();
    }

    public void toggleAbout ()
    {
        if (dlgabout_ == null)
            dlgabout_ = new DialogAbout ();
        toggleFrame (dlgabout_);
    }

    public void toggleSettings ()
    {
        if (dlgsettings_ == null)
            dlgsettings_ = new
DialogSettings (this);
        toggleFrame (dlgsettings_);
        viewcam_.reset();
        viewcam_.setCamera(winaspect_);
    }

    public String getWorldURL ()
    {
        if (baseurl_ != null)
            return (baseurl_.toString ());
        else
            return ("");
    }

    public Hashtable getNodeNames ()
    {
        return nodenames_;
    }

    public synchronized void
replaceScene (GroupNode node)
    {
        clear ();
        root = node;
        redraw ();
    }

    void setCanvas (SceneCanvas canvas)
    {
        canvas_ = canvas;
    }

    public void setNavigationBar
(NavigationBar navbar)
    {
        navbar_ = navbar;
    }

    navbar_.setCurrentButtonsVisible(DB_I
NTERACTION,interaction_);

    navbar_.setCurrentButtonsVisible(DB_B
EHAVIOUR,behavior_);
    }

    public void setNavigationMode (int
mode, int whatupdates)
    {
        db_movemode_ = mode;
        if (whatupdates == UPDATE_MENU)
        {
            ((RadioMenuItem) menuitems_
[MenuDef.M_Look + mode]).setState
(true);
        }
        if (whatupdates == UPDATE_ALL)
        {
            ((RadioMenuItem) menuitems_
[MenuDef.M_Look + mode]).setState
(true);
        }

        navbar_.setCurrentButtonsVisible(mode
,true);
    } // setNavigationMode

    // jump to the next Viewpoint
    public void nextViewpoint()
    {
        if (!viewpoints_.isEmpty())
        {
            if
(viewpoint_number_+1<viewpoints_.size
())
            {
                viewpoint_number_++;
                if
(viewpoint_number_+1==viewpoints_.siz
e())
                navbar_.toggleNextViewState(false);
                navbar_.togglePrevViewState(true);
                setViewpoint(viewpoint_number_);
            }
        }

        setViewpointText(viewpoint_number_);
    }

    // jump to the previous Viewpoint
    public void prevViewpoint()
    {
        if (!viewpoints_.isEmpty())
        {
            if (viewpoint_number_>0)
            {
                if (viewpoint_number_== 1)
                navbar_.togglePrevViewState(false);
                viewpoint_number_--;
            }
        }

        navbar_.toggleNextViewState(true);

        setViewpointText(viewpoint_number_);

        setViewpoint(viewpoint_number_);
    }

    public Vector viewpoints_ = new
Vector(); // saves all Viewpoints
dynamically
    int viewpoint_number_=0;
    Viewpoint activ_viewpoint_;

```

```

        // adds a Viewpoint to the Vector
        public void addViewpoint(Viewpoint
vp)
        {
            viewpoints_.addElement(vp);
            if (menuitems_!=null)
            {
                menuitems_
[MenuDef.M_PrevViewpoint].setEnabled
(true);
                menuitems_
[MenuDef.M_NextViewpoint].setEnabled
(true);
                if (viewpoints_.size())>1)
                {

navbar_.toggleNextViewState(true);
                setViewpointText(0);
                }
            }

canvas_.setViewpoints(viewpoints_);
        }

        public void setViewpoint(int i)
        {
            viewpoint_number_=i;
            if (viewpoint_number_ == 0)
            {

navbar_.togglePrevViewState(false);
                if (viewpoints_.size())>1)
navbar_.toggleNextViewState(true);
                }
                if (viewpoint_number_+1
==viewpoints_.size())
                {

navbar_.toggleNextViewState(false);
                if (viewpoints_.size())>1)
navbar_.togglePrevViewState(true);
                }
                activ_viewpoint_=(Viewpoint)
viewpoints_.elementAt(i);
                if ((menuitems_!=null) &&
(i<MenuBuilder.VIEWPOINT_NUM))
                {
                    ((RadioMenuItem) menuitems_
[MenuDef.M_VP1 + i]).setState (true);
                }
                else
                for (int j=0;j<MenuBuilder.VIEWPOINT_NUM;j++)
                    ((RadioMenuItem) menuitems_
[MenuDef.M_VP1 + j]).setStateOff();
                viewcam_.reset();

viewcam_.setActivViewPoint(activ_view
point_);
                viewcam_.setCamera(winaspect_);

setViewpointText(viewpoint_number_);

navbar_.setStatusText(activ_viewpoint
_.description.getValue()+" ");
            }
        }

        private void setViewpointText(int
position)
        {
            Viewpoint view;
            if
(position+1<viewpoints_.size())
            {
                view=(Viewpoint)
viewpoints_.elementAt(position+1);

navbar_.setNextViewText(view.descript
ion.getValue()+" ");
            }

            if (position>0)
            {
                view=(Viewpoint)
viewpoints_.elementAt(position-1);

navbar_.setPrevViewText(view.descript
ion.getValue()+" ");
            }
        }

        boolean sensor_=false;

        public boolean isSensorActiv()
        {
            return sensor_;
        }

        public void setSensor(boolean val)
        {
            sensor_=val;
        }

        public boolean cursor_changed_;

        public void setCursor (int mode)
        {
            if (!VRwave.oldcursor)
            {
                Toolkit toolkit =
Toolkit.getDefaultToolkit();
                Image h_ico =
navbar_.getCursor(mode);
                Point point = new Point(16,
16);
                Cursor cursor =
toolkit.createCustomCursor(h_ico,
point , "SceneCursor");
                canvas_.setCursor(cursor);
            }
            else
            {
                if (isSensorActiv())
canvas_.setCursor(Cursor.getPredefine
dCursor(Cursor.DEFAULT_CURSOR));
                else
canvas_.setCursor(Cursor.getPredefine
dCursor(Cursor.CROSSHAIR_CURSOR));
            }
        } // setCursor

    } // Scene

```

5

Εγχειρίδιο χρήσης - Manual

5.1 Εισαγωγή

Το περιβάλλον VRMLbro είναι ένας αυτόνομος VRML browser που παρέχει στους χρήστες του τη δυνατότητα για πλήρη απεικόνιση, πλοήγηση και αποθήκευση VRML κόσμων, έχοντας ανά πάσα στιγμή τον έλεγχο του πλήρους ελέγχου της όλης διαδικασίας.

Για την ορθή εκτέλεση της εφαρμογής, απαιτείται η ύπαρξη τουλάχιστον Java JDK $\geq 1.2.x$ ή JRE $\geq 1.2.x$ στο υπολογιστικό σύστημα του χρήστη, καθώς και η ύπαρξη των σχετικών εκτελέσιμων αρχείων. Είναι, δε, προσβάσιμη στον καθένα για download μέσω του Internet στην ηλεκτρονική διεύθυνση:

<http://www.softlab.ntua.gr/~fmylonas/vrmlbro/>

5.2 Εγκατάσταση

Για την επιτυχή εγκατάσταση του καινοτόμου περιβάλλοντος πλοήγησης VRMLbro απαιτείται αρχικά το download των απαραίτητων αρχείων από τον παραπάνω διαδικτυακό τόπο. Για να είναι ενεργοποιημένες όλες οι δυνατότητες του VRMLbro θα πρέπει να γίνει download το συμπιεσμένο αρχείο VRMLbro-v1.5, ενώ θεωρείται απαραίτητη η ύπαρξη υποστήριξης OpenGL από τον υπολογιστή του τελικού χρήστη.

Για το λειτουργικό σύστημα UNIX και τα παράγωγά του θα πρέπει να γίνει download η αντίστοιχη κατάλληλη version έτοιμων binaries, ή φυσικά, η λήψη του πηγαίου κώδικα του συστήματος και το manual compiling του σε εκτελέσιμο αρχείο στο εκάστοτε Unix-οειδές περιβάλλον λειτουργικού. Ο τελικός χρήστης είναι σε θέση να μετονομάσει κατά το δοκούν το downloaded binary αρχείο, εφόσον κάτι τέτοιο τον βολεύει καλύτερα στο σύστημά του.

Προκειμένου να είναι σε θέση να εκτελέσει τον VRMLbro και να διαβάσει ένα αρχείο VRML, είτε από τον τοπικό σκληρό δίσκο του μηχανήματος στο οποίο εργάζεται, είτε από κάποια δικτυακή τοποθεσία, δεν απαιτούνται κανενός είδους αλλαγές ή άλλες παραμετροποιήσεις πέραν της εντολής:

java VRMLbro


Εφόσον ο χρήστης επιθυμεί, είναι σε θέση να συνδέσει όλες τις VRML σκηνές που συναντά στις δικτυακές περιηγήσεις του με τον VRMLbro, ούτως ώστε να τις επεξεργάζεται άμεσα. Για παράδειγμα, σε περιβάλλον win32 οι αντιστοιχίες που πρέπει να γίνουν στο λειτουργικό σύστημα είναι οι ακόλουθες:

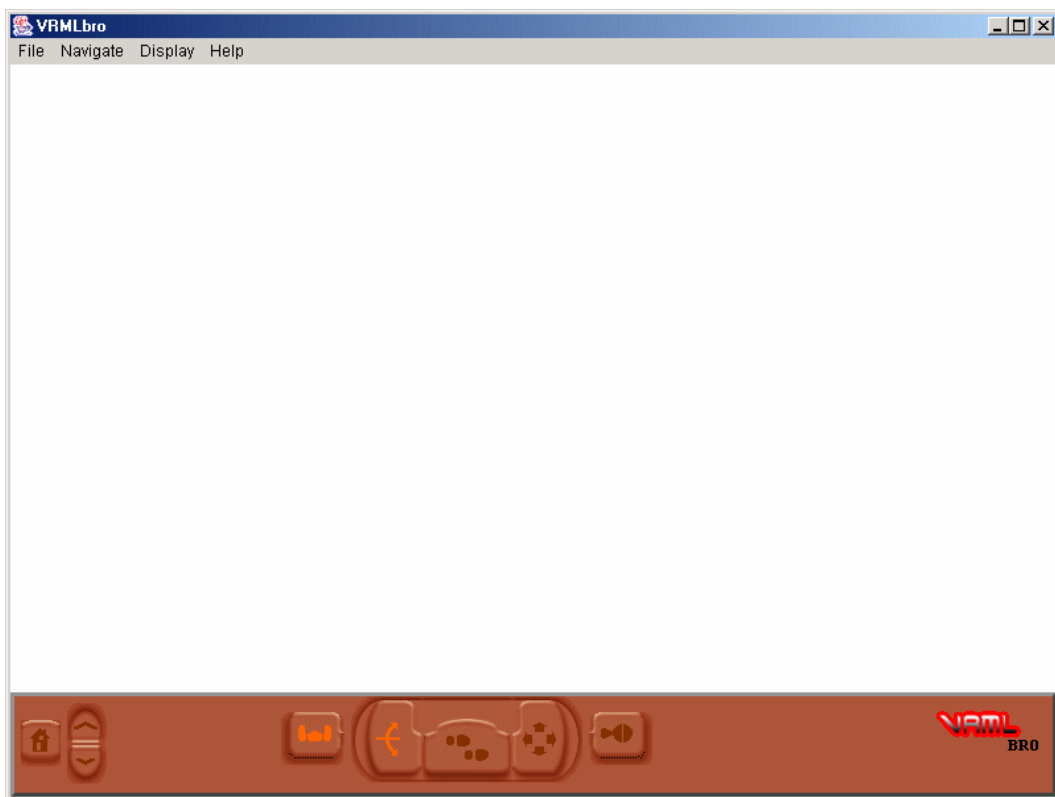
```
x-world/x-vrml; vrmlbro-mesa -URL '%u' -remote %s
model/vrml; vrmlbro-mesa -URL '%u' -remote %s
```

, ενώ για χειρισμό τοπικών αρχείων .wrl από τον σκληρό δίσκο απαιτούνται παρεμφερείς κατάλληλες αλλαγές στο αρχείο mime.types του λειτουργικού συστήματος, είτε κατασκευάζοντας εξ αρχής έναν νέο mime-type, είτε αλλάζοντας την ήδη υπάρχουσα αντιστοίχιση γι' αυτόν.

5.3 Βασικά μενού - Menus:

Κατά την εκτέλεση του προγράμματος από τον χρήστη, παρουσιάζεται σε αυτόν το βασικό μενού του browser, όπου και παρέχονται τέσσερις κατηγορίες



υπομενού:  File Navigate Display Help, ενώ στο κάτω μέρος της εφαρμογής βρίσκεται ορατή και προσβάσιμη ανά πάσα στιγμή μια ομάδα οκτώ κουμπιών πλοήγησης. Η εικόνα που αντικρίζει ο χρήστης, αμέσως μετά την πρώτη εκτέλεση του εκτελέσιμου αρχείου του VRMLbro browser, είναι η ακόλουθη:



Σχήμα 5.1

Η προγραμματιστική επιλογή του να μην εμφανίζεται αρχικά φορτωμένος κάποιος VRML κόσμος έγινε συνειδητά, με κύριο στόχο την απλότητα και την ταχύτητα κατά την πρώτη εκτέλεση της εφαρμογής, χωρίς την ύπαρξη περιττών χρόνων rendering κάποιου default κόσμου. Συνεπώς αυτόματα μετά την εκτέλεση του VRMLbro browser στον χρήστη εμφανίζονται όλες οι διαθέσιμες λειτουργίες σε λευκό φόντο.

Tips:

- Προκειμένου να βελτιώσει ο χρήστης την ικανότητα οπτικής αναπαράστασης των κόσμων που θα φορτώσει μέσω του VRMLbro, είναι σε θέση να παραμετροποιήσει το interface του περιβάλλοντος, μεγιστοποιώντας τις διαστάσεις του παραθύρου. Στο πάνω δεξιό μέρος κάθε ενεργού παραθύρου υπάρχει πάντα μια τριάδα κουμπιών διαχείρισης του παραθύρου: . Χρησιμοποιώντας το μεσαίο από αυτά επέρχεται η μεγιστοποίηση του παραθύρου σε Full Screen mode, έτσι ώστε να πλοηγείται κανείς σε κόσμους χρησιμοποιώντας σχεδόν το σύνολο της οθόνης του.
- Για να επανέλθει κανείς σε μικρότερο παράθυρο από το μέγιστο, καλείται να χρησιμοποιήσει πάλι το μεσαίο εκ των τριών κουμπιών  και στη συνέχεια το ποντίκι για να ορίσει τα καινούρια όρια το παραθύρου του.

5.4 Υπομενού - Submenus:

Θα εξετάσουμε τώρα καθεμία από τις τέσσερις υποκατηγορίες menu που παρέχονται στον χρήστη ξεχωριστά:

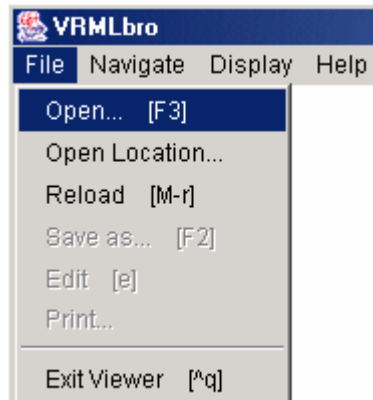
5.4.1 File

Από το menu **File** ο χρήστης είναι σε θέση να προσπελάσει μια πληθώρα λειτουργιών σχετικών με:

- ο το άνοιγμα αρχείων VRML από τον τοπικό υπολογιστή: **Open – [F3]**
- ο το άνοιγμα αρχείων VRML από απομακρυσμένο υπολογιστή: **Open Location**
- ο επαναφόρτωση επιλεγμένου VRML κόσμου: **Reload – [M-r]**
- ο αποθήκευση αρχείου VRML στον τοπικό υπολογιστή, με δυνατότητα επιλογής του ονόματός και της τοποθεσίας αποθήκευσής του: **Save as – [F2]**
- ο επεξεργασία αρχείου VRML: **Edit – [e]**
- ο εκτύπωση screenshot ενός κόσμου VRML στον εκτυπωτή της επιλογής του:

Print

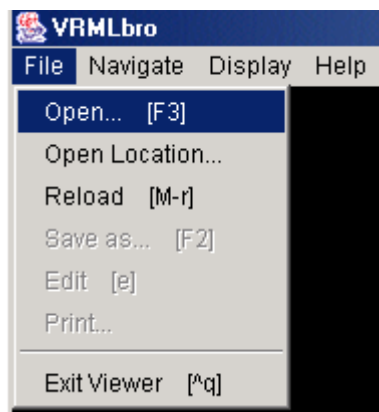
- ο ολική έξοδο από την εφαρμογή: **Exit Viewer - [^q]**



Σχήμα 5.2

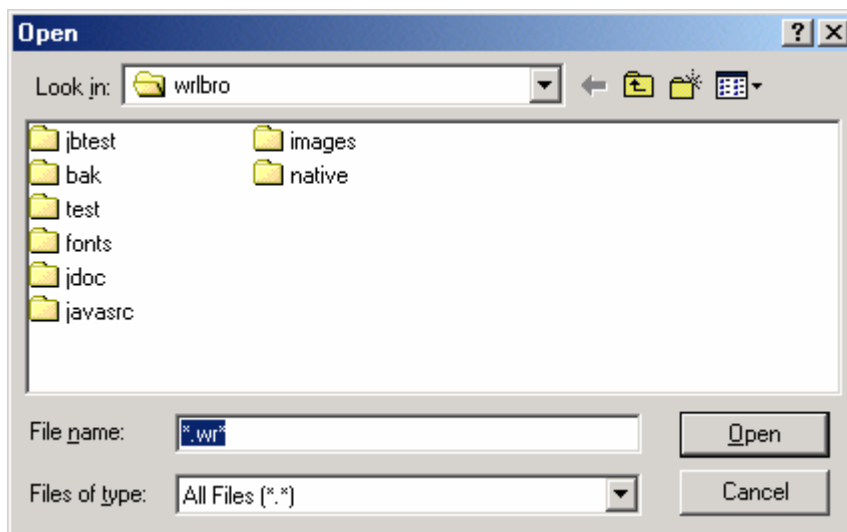
Όπως είναι φανερό και από τα συνοδευτικά σχήματα, ο χρήστης έχει την δυνατότητα να επιτελέσει τις πιο χρηστικές από τις παραπάνω λειτουργίες με το πάτημα ενός συνδυασμού πλήκτρων κατευθείαν από το πληκτρολόγιό του, γεγονός που επιτείνει την ευχρηστία της εφαρμογής, αφού δεν χρειάζεται να πηγαίνει κάθε με το mouse στην επιθυμητή επιλογή. Π.χ. για την επιλογή του **File – Open** αρκεί το πάτημα του κουμπιού F3 από το πληκτρολόγιο.

Το υπομενού **File – Open**



Σχήμα 5.3

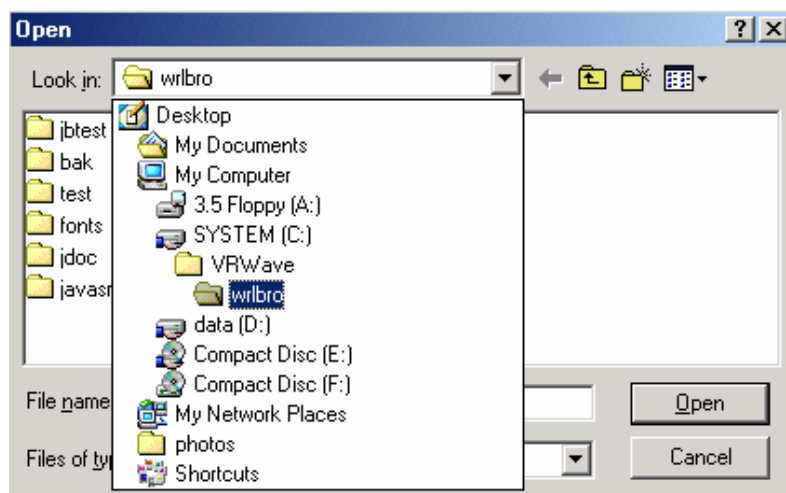
οδηγεί τον χρήστη του περιβάλλοντος VRMLbro σε ένα καινούριο ενεργό παράθυρο:






Σχήμα 5.4

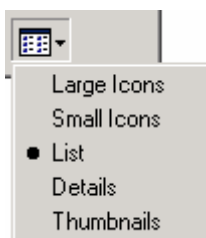
Το παράθυρο αυτό προτρέπει τον χρήστη να «ανοίξει» ένα οποιοδήποτε αρχείο VRML βρίσκεται στον τοπικό σκληρό δίσκο του υπολογιστή στον οποίο εργάζεται. Η κατάληξη του αρχείου έχει ήδη προσυμπληρωθεί για αυτόν, έτσι ώστε να μην υπάρχει περίπτωση να μπερδευτεί και να προσπαθήσει να προσπελάσει κάποιο μη έγκυρο αρχείο VRML. Τα αρχεία VRML συνήθως έχουν κατάληξη .wrl, με το πρώτο συνθετικό τους να μην υπόκεινται σε κάποιο standard και να επαφίεται στην επιλογή του εκάστοτε VRML προγραμματιστή και να είναι χαρακτηριστικό για το περιεχόμενό του. Για παράδειγμα, “office.wrl”. Ο κατάλογος στον οποίο ανοίγει από default το εν λόγω νέο παράθυρο είναι ο κατάλογος στο σύστημα, όπου είχε επιλεγεί να εγκατασταθεί ο VRMLbro.

Παρατηρούμε ότι στο νέο αυτό παράθυρο, παρέχονται όλες οι τυπικές λειτουργίες για τη διαχείριση φακέλων και καταλόγων του λειτουργικού συστήματος. Ειδικότερα, παρέχεται στον χρήστη δυνατότητα να αλλάξει τον κατάλογο όπου αναζητεί αρχεία VRML μέσω του παρακάτω drop-down menu, το οποίο εμφανίζεται μόλις ο χρήστης πατήσει πάνω στο βελάκι στο δεξιό σημείο:




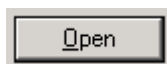
Σχήμα 5.5

Ακόμα, παρέχεται η δυνατότητα να «ανεβεί» ο χρήστης στην ιεραρχία των καταλόγων που βρίσκεται, μέσω του: , να δημιουργήσει καινούριο κατάλογο μέσα στον current κατάλογο , δίνοντάς του ταυτόχρονα και ένα καινούριο όνομα (με την προϋπόθεση βέβαια ότι ο εν λόγω χρήστης διαθέτει τέτοια δικαιώματα στο υποκείμενο λειτουργικό σύστημα), να μεταβάλει την απεικόνιση των αρχείων και των καταλόγων  στο εν λόγω παράθυρο

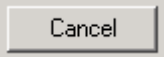


Σχήμα 5.6

και τέλος να επανέλθει σε κατάλογο που είχε επισκεφθεί προγενέστερα στην ιεραρχία των καταλόγων . Μόλις εντοπιστεί το επιθυμητό αρχείο, ο χρήστης μπορεί να το «ανοίξει» με το κουμπί **Open**, δηλαδή ουσιαστικά να το στείλει στον VRMLbro προς parsing, έλεγχο και rendering .

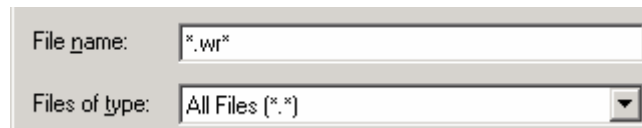


Σχήμα 5.7

Προφανώς η όλη διαδικασία μπορεί να ακυρωθεί ανά πάσα στιγμή πατώντας το κουμπί **Cancel** , οπότε ο χρήστης επιστρέφει στο αρχικό και κύριο παράθυρο του VRMLbro browser.

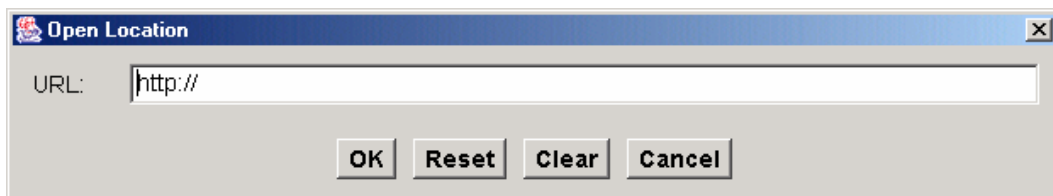
Tips:

- ο χρήστης είναι σε θέση ανά πάσα στιγμή να εισάγει free text στην περιοχή αναζήτησης αρχείων, τόσο όσον αφορά το είδος των αρχείων που θέλει να φορτώσει, όσο και όσον αφορά το όνομα του συγκεκριμένου αρχείου που αναζητά. Πέρα, δηλαδή, από την οπτική αναζήτηση παρέχεται και η καθεαυτό αναζήτηση με βάση την ονοματολογία των αρχείων VRML στον υπολογιστή του χρήστη:



Σχήμα 5.8


Το υπομενού **File – Open Location** έχει ανάλογη λειτουργία με το παραπάνω **File – Open**, αλλά διαφοροποιείται ως προς την υλοποίηση και την τελική απόδοσή του προς τον χρήστη. Συγκεκριμένα, με το πάτημα του υπομενού **File – Open location**, στον χρήστη εμφανίζεται ένα καινούριο ενεργό παράθυρο:



Σχήμα 5.9

Πρόκειται για τη γνωστή από τους απλούς HTML browsers μορφή της address bar για δικτυακές διευθύνσεις. Και εδώ η διαδικασία είναι παρόμοια με: Ο χρήστης μπορεί να πλοηγηθεί σε νέους δικτυακούς VRML κόσμους αφού εισάγει το URL τους – δηλαδή την τοποθεσία τους (address) στο web. Τα URLs συνήθως ξεκινούν με το http://, οπότε αυτό έχει ήδη προσυμπληρωθεί για τους χρήστες, αν και δεν αποκλείεται και εδώ η εισαγωγή free text. Μετά το http://

ακολουθεί κάποιο αναγνωριστικό όνομα της διεύθυνσης, το οποίο και χαρακτηρίζει τον VRML κόσμο. Για παράδειγμα: "http://www.softlab.ntua.gr/~fmylonas/vrmlbro/myvrmlworld.wrl".

Εκτός από την παραπάνω κλασική μπάρα διευθύνσεων στο καινούριο παράθυρο υπάρχουν τέσσερα κουμπιά: . Το **Ok** απλά αποτελεί την επιβεβαίωση κι αποστολή της όποιας παραπάνω εισαχθείσας διεύθυνσης, το **Reset** επαναφέρει την μπάρα στην αρχική της μορφή μετά από οποιοδήποτε input του χρήστη, χωρίς να αποστείλει τίποτα από αυτό, το **Clear** απλώς καθαρίζει το free text της εισαγωγής του χρήστη και τέλος το **cancel** ακυρώνει την όλη διαδικασία ανοίγματος δικτυακού αρχείου VRML και επιστρέφει το focus και την προσοχή του χρήστη στο κύριο παράθυρο του VRMLbro browser.

Tips:

- Για γρήγορη επιλογή όλου του εισαχθέντος στην address bar κειμένου από το σημείο που γράφει ο χρήστης μέχρι την αρχή της, καλείται να χρησιμοποιήσει το Shift + Home.
- Για γρήγορη επιλογή όλου του εισαχθέντος στην address bar κειμένου από το σημείο που γράφει ο χρήστης μέχρι το τέλος της, καλείται να χρησιμοποιήσει το Shift + End.

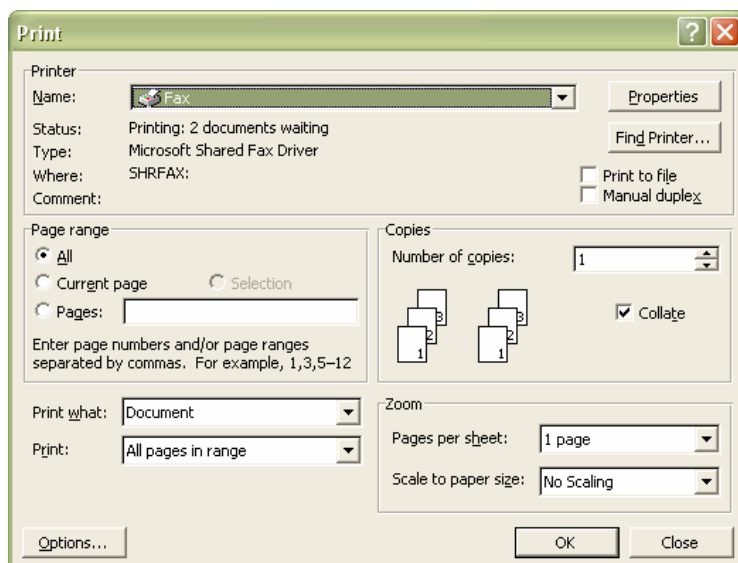
Η λειτουργία του υπομενού **File – Reload** έχει να κάνει με την επαναρτικοποίηση της σκηνής που βλέπει κάθε στιγμή ο χρήστης στην αρχική της κατάσταση. Αποβαίνει ιδιαίτερα χρήσιμη στις περιπτώσεις εκείνες κατά τις οποίες «κολλάει» ή «κρεμάει» ο VRML κόσμος λόγω προγραμματιστικού σφάλματος μέσα στον κώδικα της VRML, οπότε και επαναφέρεται ολόκληρος ο κόσμος στην αρχική του κατάσταση, ενώ δεν παύει να είναι χρήσιμη και ανάλογα με διαχειριστική θέληση του χρήστη στον εκάστοτε κόσμο.

Το υπομενού **File – Save as** δίνει την δυνατότητα στον χρήστη να αποθηκεύσει μια VRML σκηνή στον τοπικό σκληρό δίσκο του υπολογιστή στον

οποίο εργάζεται. Προϋποθέτει φυσικά την ύπαρξη κάποιου VRML κόσμου ήδη φορτωμένου από τον browser και με το πάτημα του υπομενού προσφέρεται στον χρήστη ένα καινούριο ενεργό παράθυρο, σε αναλογία με τη διαδικασία κατά το άνοιγμα ενός VRML αρχείου από τον τοπικό σκληρό δίσκο κάποιου υπολογιστή και με όμοια χαρακτηριστικά.

Το υπομενού **File – Edit** παραμένει σε αυτήν την έκδοση του VRMLbro browser απενεργοποιημένο, μιας και οι δυνατότητες editing κόσμων VRML μέσω αυτού ωθούν το αντικείμενο αυτής της εργασίας πέρα από τα όρια τα οποία θεσπίστηκαν και αποτελούν αντικείμενο περαιτέρς ερευνητικής και υπολογιστικής έρευνας.

Το υπομενού **File – Print** παρέχει την δυνατότητα εκτύπωσης κάποιου τρισδιάστατου VRML κόσμου. Η ικανότητα απεικόνισης ενός τέτοιου εγχειρήματος, βέβαια, εναπόκειται στα χαρακτηριστικά του εκάστοτε printer που είναι συνδεδεμένος με τον υπολογιστή του χρήστη και προφανώς περιορίζεται από αυτά. Το παράθυρο που ανοίγει είναι το κλασικό παράθυρο διαχείρισης εκτυπωτών συστήματος:



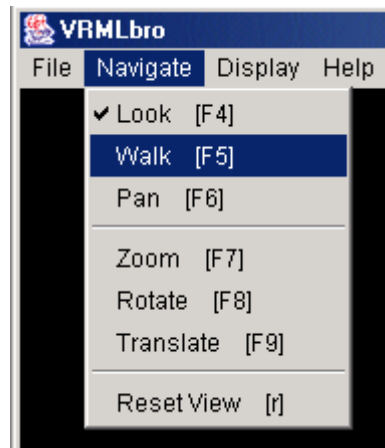
Σχήμα 5.10

Τέλος, κλείνοντας την πρώτη ενότητα για το μενού **File** υπάρχει εδώ η δυνατότητα για ολική έξοδο από το περιβάλλον του VRMLbro, επιλέγοντας το υπομενού **Exit Viewer**, οπότε και πραγματοποιείται ομαλή έξοδος από αυτό.

5.4.2 Navigate

Από το menu *Navigate* παρέχονται στον χρήστη όλες οι λειτουργίες πλοήγησης, χωρισμένες σε τρεις ομάδες, ανάλογα με την ιδιαιτερότητα τους. Αναλυτικότερα, υπάρχει δυνατότητα για:

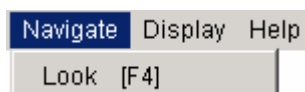
- ο Παρατήρηση του VRML κόσμου: **Look – F4**
- ο Περιπάτημα μέσα στον VRML κόσμο: **Walk – [F5]**
- ο Εκτέλεση πανοραμικής λήψης: **Pan – [F6]**
- ο Εμφάνιση ενός αντικειμένου κοντύτερα/μακρύτερα: **Zoom – [F7]**
- ο Περιστροφή ενός αντικειμένου: **Rotate – [F8]**
- ο Μεταφορά ενός αντικειμένου: **Translate – [F9]**
- ο Επαναρύθμιση της απεικόνισης του VRML κόσμου: **Reset View – [r]**




Σχήμα 5.11

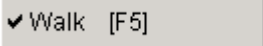

Όπως καταδεικνύεται και από τα συνοδευτικά σχήματα, ο χρήστης έχει πάλι όπως και προηγουμένως την δυνατότητα να επιτελέσει τις πιο χρηστικές από τις παραπάνω λειτουργίες με το πάτημα ενός συνδυασμού πλήκτρων κατευθείαν από το πληκτρολόγιό του, γεγονός που επιτείνει την ευχρηστία της εφαρμογής. Π.χ. για την επιλογή του **Navigate – Look** αρκεί το πάτημα του κουμπιού F4 από το πληκτρολόγιο.

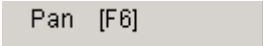

Το υπομενού **Navigate – Look**



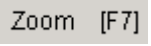

Σχήμα 5.12

προϋποθέτει την ύπαρξη ενός φορτωμένου στον browser VRML κόσμου. Επιδρά στο ποντίκι του χρήστη, μετατρέποντας τον κλασικό pointer cursor από βελάκι στο ειδικό σύμβολο . Μέσω αυτού ο χρήστης είναι σε θέση να παρατηρήσει («look») τον VRML κόσμο, κάνοντας click and drag από ένα σημείο του κόσμου σε κάποιο άλλο. Η φορά παρατήρησης ορίστηκε αντίθετη με την κατεύθυνση του drag, καθώς τέτοια είναι και στους πιο δημοφιλείς ανάλογους εμπορικούς VRML browsers.

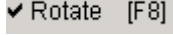

Το υπομενού **Navigate – Walk**  μετατρέπει τον pointer cursor του χρήστη στο σύμβολο  και του επιτρέπει να περιδιαβαίνει («walk») μέσα στον φορτωμένο VRML κόσμο. Αυτό επιτυγχάνεται με click and drag του ποντικιού από την αρχική στην τελική θέση περιπάτου. Και εδώ η φορά του περιπάτου είναι αντίθετη με την κίνηση του ποντικιού, κατ' αναλογία με τα παραπάνω.

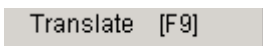

Από το υπομενού **Navigate – Pan**  ο pointer cursor του χρήστη μετατρέπεται στο σύμβολο  και ο χρήστης είναι πλέον σε θέση να πραγματοποιήσει πανοραμικές λήψεις («pan») του VRML κόσμου μέσα στον οποίο πλοηγείται. Η διαδικασία που ακολουθείται και εδώ είναι η ίδια, με ανάποδη της κίνησης του mouse διεύθυνση απόκρισης της λήψης.

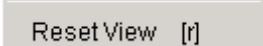
Τα τρία παραπάνω υπομενού του menu **Navigate** αποτελούν και την πρώτη ομάδα του. Η διάκριση σε ομάδες κρίθηκε αναγκαία, καθώς ορισμένες λειτουργίες του menu **Navigate** είναι απόλυτα παραπλήσιες και σχετικές μεταξύ τους. Η επόμενη ομάδα υπομενού αποτελείται πάλι από τρία νέα υπομενού και αφορά λειτουργίες οπτικής απεικόνισης και χειρισμού των αντικειμένων μέσα στον VRML κόσμο. Πρόκειται για τα υπομενού **Zoom**, **Rotate** και **Translate** τα οποία και αναλύουμε ευθύς αμέσως.

Το υπομενού **Navigate – Zoom**  παρέχει στον τελικό χρήστη δυνατότητες εστίασης («zoom») σε κάποιο αντικείμενο του χώρου. Αρχικά τροποποιεί τον pointer cursor του χρήστη στο σύμβολο , ενώ ταυτόχρονα

παρέχονται δυνατότητες zoom in και zoom out σε αντικείμενα του κόσμου. Η διαδικασία υλοποιείται με click and drag προς το επάνω μέρος του ενεργού παραθύρου του browser για το zoom out και προς το κάτω μέρος για το zoom in.

Το υπομενού **Navigate – Rotate**  [F8] προσφέρει δυνατότητες περιστροφής («rotate») των αντικειμένων του VRML κόσμου. Τροποποιεί και αυτό με την επιλογή του τον pointer cursor του χρήστη στο σύμβολο  και κάνοντας click and drag σε κάποιο σημείο του κόσμου, τα αντικείμενα περιστρέφονται γύρω από αυτό.

Τέλος, το υπομενού **Navigate – Translate**  [F9] έρχεται να ολοκληρώσει τις προσφερόμενες λειτουργίες πλοήγησης της δεύτερης ομάδας λειτουργιών, προσφέροντας στον τελικό χρήστη δυνατότητες μεταφοράς («translate») αντικειμένων του VRML κόσμου. Πραγματοποιείται κατάλληλη τροποποίηση του pointer cursor του χρήστη στο σύμβολο , ενώ η μεταφορά των αντικειμένων υλοποιείται με click and drag του επιθυμητού αντικειμένου.

Το menu **Navigate** ολοκληρώνεται με ένα τελευταίο υπομενού, το υπομενού **Reset View**  [r], το οποίο και «καθαρίζει» την εκάστοτε φορτωμένη σκηνή από τις όποιες παρεμβάσεις του χρήστη και την επαναφέρει («reset») στην αρχική της μορφή. Με τον τρόπο αυτό είναι δυνατή η απεμπλοκή του χρήστη από τυχούσες περίπλοκες αλλαγές στον VRML κόσμο και η εκ του μηδενός προσέγγιση του αρχικού setup.

5.4.3 Display

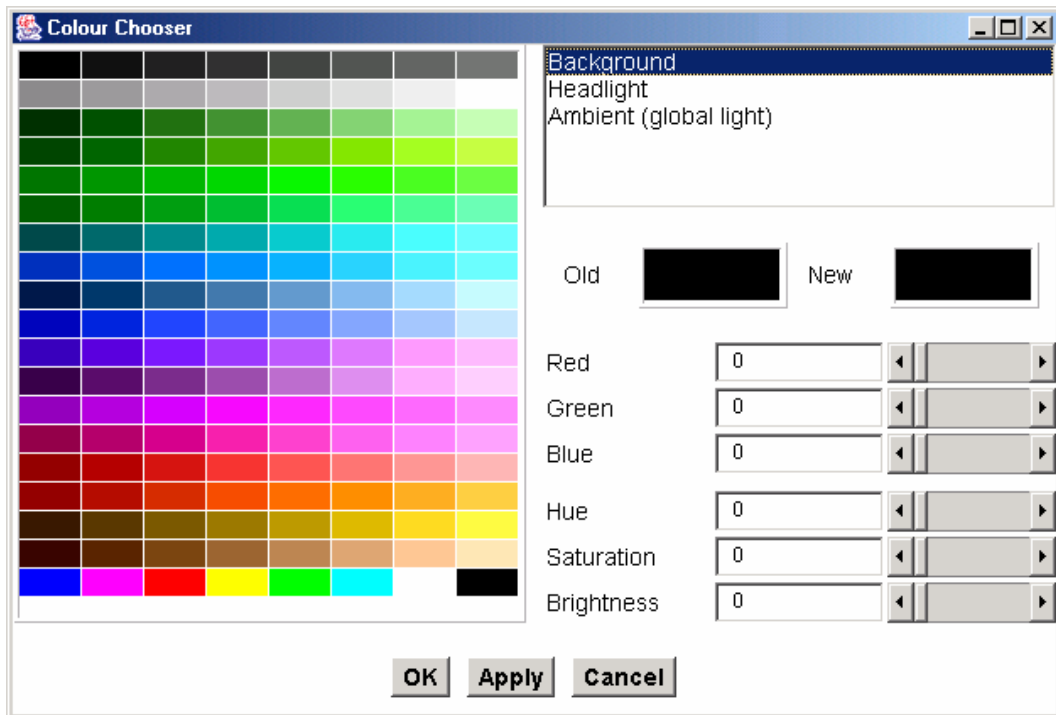
Από το menu **Display** ο χρήστης έχει την δυνατότητα να παραμετροποιήσει κατά το δοκούν την επιφάνεια εργασίας του browser, όπως παρατηρούμε και στο σχήμα 5.13 που ακολουθεί:



Σχήμα 5.13

Διατίθενται δύο επιλογές υπομενού: η επιλογή **Colours** για την επιλογή της παλέτας των χρωμάτων και η επιλογή **Background** για τον ορισμό του χρωματικού φόντου στον browser.

Αναλυτικότερα, με την επιλογή του υπομενού **Colours** παρουσιάζεται στον χρήστη του περιβάλλοντος VRMLbro ένα καινούριο ενεργό παράθυρο:



Σχήμα 5.14

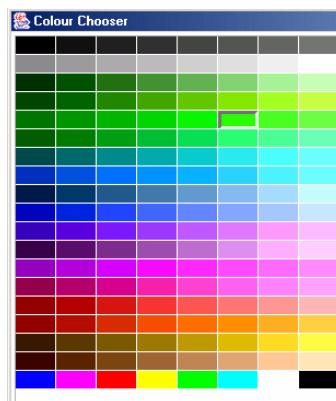
Το παράθυρο αυτό περιέχει μια πλειάδα επιλογών και παραμέτρων, μέσω των οποίων ο χρήστης είναι σε θέση να μεταβάλει την οπτική απεικόνιση του ίδιου του περιβάλλοντος του VRMLbro, καθώς και παραμέτρους σχετικούς με τους VRML κόσμους που προσπελαύνει, όπως ο ambient φωτισμός και το headlight. Συνοπτικά, οι παράμετροι γραφικών που μπορούν να μεταβληθούν από το παράθυρο αυτό είναι οι εξής:

- Το χρώμα του background
- Το χρώμα του headlight φωτισμού
- Το χρώμα του ambient φωτισμού
- Αναλυτικές RGB values [0-256] για κάθε χρωματική συνιστώσα ξεχωριστά

- ο Τιμές απόχρωσης (hue values) [0-256]
- ο Τιμές κορεσμού (saturation values) [0-256]
- ο Τιμές φωτεινότητας (brightness values) [0-256]

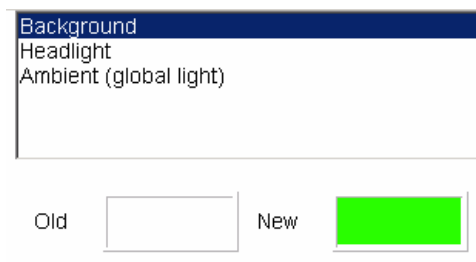
Εκτός από τις παραπάνω παραμέτρους προσφέρεται στο ίδιο παράθυρο στον χρήστη και μια παλέτα 152 διαφορετικών χρωματικών αποχρώσεων, η οποία χρησιμεύει για την οπτική αναπαράσταση των παραπάνω αλλαγών. Επίσης, προς μεγαλύτερη διευκόλυνση του χρήστη παρέχονται 2 ενεργές περιοχές, όπου απεικονίζεται η πρότερη και η επόμενη της αλλαγής χρωματική συνιστώσα, πριν από την επιβολή της ίδιας της αλλαγής αυτής καθεαυτής. Παρέχεται, δηλαδή, στον χρήστη μία μορφή preview των αλλαγών που επίκειται, όσον αφορά τις χρωματικές επιλογές του.

Αναλυτικότερα, ξεκινώντας την περιγραφή μας από το πάνω αριστερά τμήμα του παραθύρου προς το κάτω δεξιά, συναντάμε αρχικά την παλέτα των χρωμάτων.




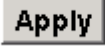
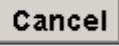
Σχήμα 5.15

Η παλέτα αυτή είναι λειτουργική και για τα τρία είδη χρωματικών παραμέτρων που διατίθενται. Λίγο δεξιότερα από την παλέτα βρίσκονται στο παράθυρο των χρωματικών επιλογών οι τρεις αυτές παράμετροι, καθώς και οι δύο περιοχές προεπισκόπησης (preview panels).

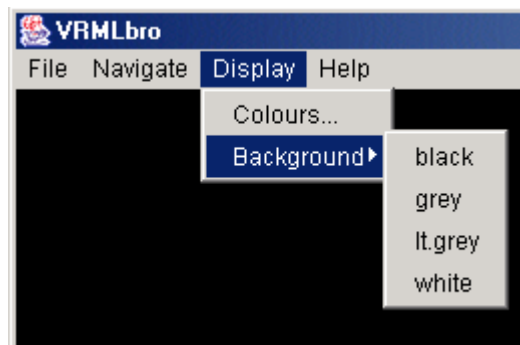


Σχήμα 5.16

Ακολουθούν, στην ακριβώς από κάτω θέση για λόγους λειτουργικότητας και ευχρηστίας, έξι sliders: οι τρεις πρώτες για την παραμετροποίηση τιμών RGB, προσφέροντας μία μπάρα ανά χρωματική συνιστώσα, στα πλαίσια πάντα των επιτρεπτών τιμών $[0 - 255]$. Η τέταρτη slider παραμετροποιεί την τιμή απόχρωσης, η πέμπτη την τιμή κορεσμού και η τελευταία την τιμή φωτεινότητας ολόκληρου του εκάστοτε κόσμου. Ο χρήστης είναι σε θέση να εισάγει τιμές στα παραπάνω και με χρήση free text στα κατάλληλα πεδία δίπλα στις sliders, γεγονός που αυξάνει την χρηστικότητα και την φιλικότητα του περιβάλλοντος.

Το παράθυρο επιλογής χρωμάτων ολοκληρώνεται με τρία κουμπιά στο κάτω τμήμα του: **Ok**, **Apply** και **Cancel**   . Το **Ok** αποτελεί την επιβεβαίωση και αποστολή των όποιων αλλαγών, κλείνοντας το εν λόγω παράθυρο, το **Apply** εφαρμόζει μεν τις όποιες αλλαγές, αλλά δεν κλείνει το παράθυρο επιλογής χρωμάτων, έτσι ώστε ο χρήστης να προχωρήσει και σε άλλες και το **Cancel** ακυρώνει την όλη διαδικασία επιλογής χρωμάτων και επιστρέφει το focus και την προσοχή του χρήστη στο κύριο παράθυρο του VRML browser.

Εκτός από τα παραπάνω, στο menu **Display** προσφέρεται και το υπομενού **Background** για τον καθορισμό του background του browser ανάμεσα σε τέσσερα δυνατά χρώματα, κατάλληλα για το ανθρώπινο μάτι (μαύρο, γκρι, «ανοιχτό» γκρι και άσπρο). Η επιλογή πραγματοποιείται με το ποντίκι και εφαρμόζεται άμεσα στον browser.



Σχήμα 5.17

5.4.4 Help

Τέλος, το τελευταίο από τα menus του browser είναι το menu **Help**, απ' όπου ο χρήστης είναι σε θέση να ενημερωθεί – μέσω του ακόλουθου pop-up window – κάποιες πληροφορίες σχετικά με το περιβάλλον του VRMLbro browser και τον συγγραφέα του.



Σχήμα 5.18



Σχήμα 5.19

5.5 Κουμπιά πλοήγησης - Buttons:

Στο κάτω μέρος του περιβάλλοντος VRMLbro βρίσκεται ανά πάσα στιγμή προσβάσιμη στον χρήστη μια ομάδα από οκτώ κουμπιά πλοήγησης. Στο δεξί μέρος, δε, της μπάρας αυτής έχει τοποθετηθεί το λογότυπό του.



Σχήμα 5.20

Θα εξετάσουμε τώρα τις επιμέρους λειτουργίες των κουμπιών αυτών, ξεκινώντας από τα αριστερά της μπάρας προς τα δεξιά. Η υλοποίηση των κουμπιών με αυτή τη μορφή έχει ως αποκλειστικό σκοπό την απλοποίηση των δυνατοτήτων και την αύξηση της ευχρηστίας του όλου περιβάλλοντος, καθώς και την αναβάθμιση του αισθητικού αποτελέσματος του όλου εγχειρήματος. Να σημειωθεί στο σημείο αυτό ότι σε κάθε συγκεκριμένο κουμπί παραθέτουμε κατάλληλο screenshot, όπου το αναφερόμενο κουμπί διαθέτει διαφορετική απόχρωση, ούτως ώστε να ξεχωρίζει από τα υπόλοιπα. Η διαφορετική αυτή απόχρωση εμφανίζεται και κατά την περίπτωση που ο χρήστης περάσει το ποντίκι του πάνω από το εκάστοτε κουμπί, προκειμένου να γνωρίζει ανά πάσα στιγμή ποιο από όλα είναι το ενεργό.

5.5.1 Reset button

Από το κουμπί αυτό ο χρήστης μπορεί να επαναφέρει τον VRML κόσμο στην αρχική του μορφή, μετά από οποιεσδήποτε ρυθμίσεις κι αν είχε κάνει.



Σχήμα 5.21

5.5.2 Zoom buttons

Πρόκειται για ένα ζευγάρι κουμπιών για την εστίαση σε VRML αντικείμενα. Με το επάνω κουμπί φέρνει κανείς το αντικείμενο κοντύτερα, ενώ με το κάτω απομακρύνεται από αυτό.



Σχήμα 5.22

5.5.3 Explore mode

Το επόμενο κουμπί θέτει την κεντρική μπάρα κουμπιών στο μέσο της εφαρμογής σε «explore mode»,



Σχήμα 5.23

οπότε τα τρία κουμπιά της μετατρέπονται σε: **Zoom – Rotate – Translate**. Με άλλα λόγια υλοποιείται η δεύτερη ομάδα λειτουργιών από το υπομενού *Navigate*, που είδαμε παραπάνω.



Σχήμα 5.24



Σχήμα 5.25



Σχήμα 5.26

5.5.4 Examine mode

Το κουμπί που βρίσκεται δεξιά των τριών κεντρικών, θέτει την μπάρα σε «examine mode»,



Σχήμα 5.27

οπότε αυτά μετατρέπονται αντίστοιχα σε: **Look – Walk – Move**. Και πάλι πρόκειται για υλοποίηση με γραφικά κουμπιά των βασικών λειτουργιών πλοήγησης της πρώτης ομάδας του υπομενού *Navigate*.



Σχήμα 5.28



Σχήμα 5.29

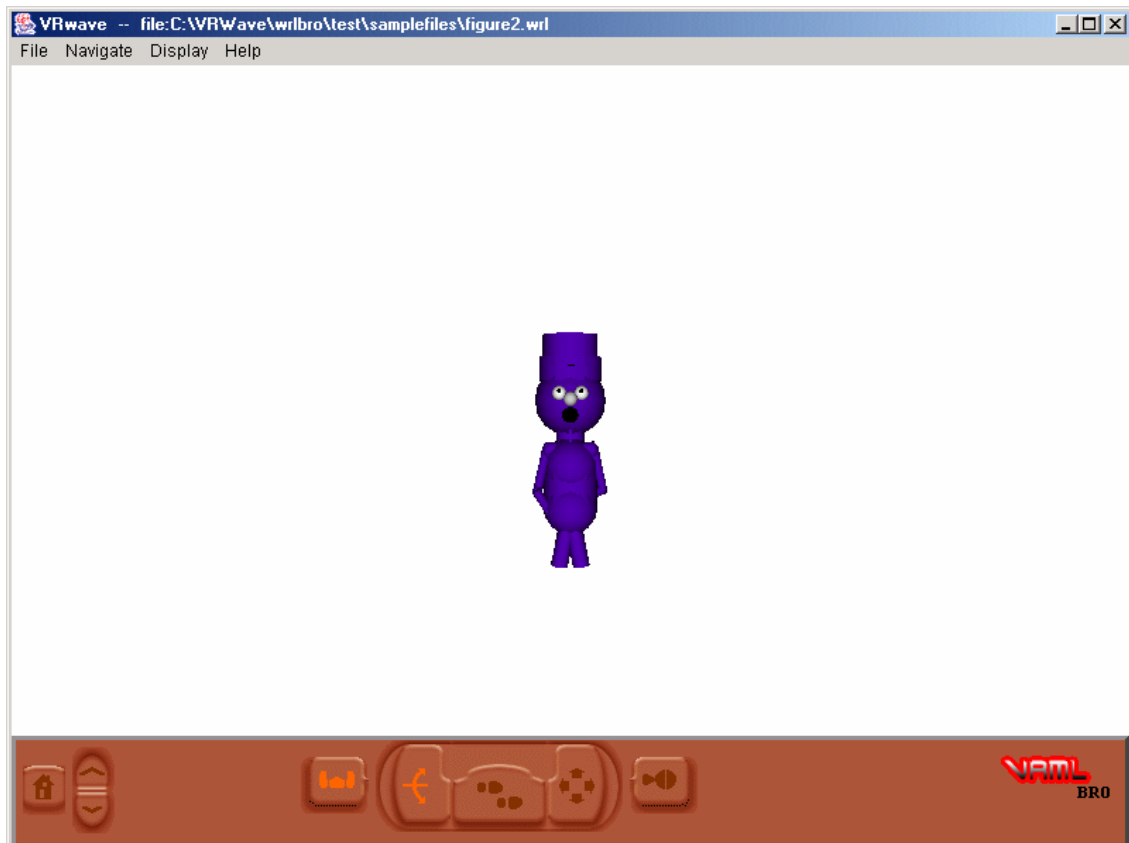


Σχήμα 5.30

Να σημειωθεί στο σημείο αυτό, ότι όπως παρατηρούμε και από τις συνημμένες εικόνες, κάθε φορά που το mouse του χρήστη περνάει ή βρίσκεται πάνω από κάποιο κουμπί, εμφανίζεται στο κάτω δεξιά μέρος της εφαρμογής η περιγραφή της λειτουργίας που επιτελεί το συγκεκριμένο κουμπί, έτσι ώστε ο χρήστης να ξέρει ανά πάσα στιγμή ποια λειτουργία πρόκειται να επιτελέσει με το πάτημα του.

5.6 Παράδειγμα εκτέλεσης:

Παραθέτουμε, τέλος, για λόγους πληρότητας και παρουσίασης του αισθητικού αποτελέσματος της εφαρμογής μας και ένα παράδειγμα εκτέλεσής της. Παρουσιάζεται ένας VRML κόσμος, που αποτελείται από μία VRML σφαίρα, η οποία είναι επικαλυμμένη με μια τυχαία εικόνα bitmap.



Σχήμα 5.31

6

Επίλογος

6.1 Μελλοντικές Επεκτάσεις Περιβάλλοντος

Η εφαρμογή VRMLbro στην παρούσα έκδοσή της αναπτύχθηκε με κύριο γνώμονα την άμεση, απροβλημάτιστη και εύκολη χρησιμοποίησή της από τους πιθανούς χρήστες της για την διευκόλυνση και απλοποίηση της διαδικασίας πλοήγησης σε VRML κόσμους χωρίς την χρήση εμπορικών browsers και/ή εξωτερικών plugins. Ένα πιθανό σημείο μελλοντικής επέκτασης από την πλευρά της υλοποίησης αποτελεί ο τρόπος με τον οποίο πραγματοποιείται το parsing και rendering των VRML κόσμων. Στην παρούσα φάση, η παραπάνω διαδικασία πραγματοποιείται με τη χρησιμοποίηση, συρραφή και κατάλληλη επεξεργασία έτοιμων βιβλιοθηκών (“pw library - VRML 97 parser” [\[L22\]](#)), οι οποίες συμμορφώνονται πλήρως με το νεώτερο του VRML 2.0 standard [\[L2\]](#), [\[L19\]](#), VRML97 standard [\[L42\]](#).

Επίσης, προκειμένου να γίνει το περιβάλλον VRMLbro πιο φιλικό και προσβάσιμο σε μη καταρτισμένους με τους υπολογιστές χρήστες, θα πρέπει να αναπτυχθεί ένα διαχειριστικό περιβάλλον εγκατάστασής του, πιθανώς με τη

χρήση κάποιου έτοιμου Installer για win32 εφαρμογές, όπως ο ευρέως διαδεδομένος InstallShield [\[L43\]](#). Τέλος, μία αρκετά ενδιαφέρουσα πρόταση επέκτασης του όλου εγχειρήματος θα αποτελούσε η προσθήκη χαρακτηριστικών editor στον υπάρχοντα Java-based browser, γεγονός που θα έδινε εξαιρετική ώθηση στην εφαρμογή. Για την υλοποίηση της παραπάνω επέκτασης απαιτείται, βέβαια, ριζική αναθεώρηση και επανασχεδίαση της εφαρμογής, ούτως ώστε ο προσανατολισμός της να συμπεριλάβει χαρακτηριστικά διάδρασης με τον εκάστοτε VRML κόσμο σε επίπεδο διαφορετικό από την «απλή» πλοήγηση σε αυτόν.

6.2 Εμπειρία χρήσης

Το ολοκληρωμένο σύστημα VRMLbro χρησιμοποιήθηκε και ελέγχθηκε εξαντλητικά από εμάς τους ίδιους. Έλεγχοι και διορθώσεις πραγματοποιήθηκαν αναφορικά με τις παρεχόμενες δυνατότητες και λειτουργίες του, τόσο από τη σκοπιά του απλού χρήστη, όσο και από τη σκοπιά του διαχειριστή του συστήματος.

Ως χρήστες αποπειραθήκαμε με πλήρη επιτυχία και ευκολία να πλοηγηθούμε σε VRML κόσμους ποικίλων μεγεθών και αντικειμένων, με ή χωρίς texture mappings. Η πλοήγηση στον εκάστοτε κόσμο μας φάνηκε ιδιαίτερα εύκολη και εύχρηστη. Για την συγκομιδή ακόμα καλύτερων αποτελεσμάτων και ανακάλυψη τυχόν παραλείψεων, ζητήσαμε και από διάφορους άλλους χρήστες, εξοικειωμένους ή μη με ηλεκτρονικούς υπολογιστές, να το δοκιμάσουν, οι οποίοι και έμειναν απόλυτα ευχαριστημένοι. Οι όποιες παρατηρήσεις τους, λήφθηκαν σοβαρά υπόψη και ενσωματώθηκαν στο τελικό περιβάλλον.

Ως διαχειριστές ασχοληθήκαμε κυρίως με την προσθαφαίρεση λειτουργιών και menus από το κυρίως παράθυρο του browser, έως ότου επιτευχθεί η μέγιστη λειτουργικότητα και ποιότητα για το παραγόμενο λογισμικό. Λάβαμε διαφόρων ειδών σχεδιαστικές αποφάσεις προ και κατά τη διάρκεια της υλοποίησης, ενώ παρεμβήκαμε καταλυτικά σε ορισμένες περιπτώσεις ασυμβατοτήτων που παρατηρήθηκαν, κυρίως σχετικά με το parsing των VRML αρχείων.

6.3 Σύνοψη και Συμπεράσματα

Με την ολοκλήρωση της εφαρμογής του VRMLbro ικανοποιείται πλήρως η ανάγκη για ομαλή και απροβλημάτιστη πλοήγηση σε VRML κόσμους μέσω ενός non-proprietary VRML browser και ανεξαρτήτως υπολογιστικής πλατφόρμας. Υλοποιήθηκε σε Java ένα πλήρες περιβάλλον αλληλεπίδρασης με VRML κόσμους, εγχείρημα πρωτοποριακό και καινοτόμο, όσον αφορά την εκ του μηδενός σχεδίαση, κατασκευή και δοκιμή του. Προσεγγίστηκε ένα πλήθος από ερευνητικές εργασίες του χώρου, οι οποίες στη συντριπτική πλειοψηφία τους πραγματεύονταν παρεμφερή προβλήματα και ερευνητικά θέματα, όπως το parsing των VRML αρχείων ή το EAI, χωρίς ωστόσο να αγγίζουν το πρόβλημα της σύζευξης των δύο πρωτοποριακών τεχνολογιών της Java και της VRML στο σύνολό τους. Η πρόταση που παρατίθεται στην διπλωματική αυτή εργασία και η οποία υλοποιήθηκε πλήρως στα πλαίσια της, αποτελεί μια ολοκληρωμένη μελέτη του προβλήματος και οδηγεί την σχετική κοινότητα σε νέους δρόμους, διευρύνοντας τους ορίζοντες της προς την νέα περιοχή των Java-based VRML browsers.

Τέλος, τα συμπεράσματα, τα οποία εξήχθησαν από αυτή την υλοποίηση της διπλωματικής εργασίας, είμαστε σε θέση να τα συνοψίσουμε στα εξής:

- Η πρόσβαση σε VRML κόσμους απλοποιείται με τη χρήση ενός σωστού VRML browser, ενώ μέσω της υλοποίησης του όλου περιβάλλοντος στη γλώσσα προγραμματισμού Java το όλο αποτέλεσμα είναι ιδιαίτερα «ελαφρύ».
- Το αισθητικό αποτέλεσμα της όλης εφαρμογής οφείλει να είναι υπέρ του δέοντος ικανοποιητικό, έτσι ώστε να «τραβάει» τον χρήστη και να του είναι αρεστό και εύχρηστο.
- Η επιλογή της γλώσσας προγραμματισμού Java αποτελεί μια ιδιαίτερα σταθερή, αξιόπιστη και ανεξάρτητη πλατφόρμας λύση, διευκολύνοντας κατά πολύ την μεταφερσιμότητα της εφαρμογής και την ικανοποίηση των απαιτήσεων ευχρηστίας και user-friendliness της εφαρμογής.

7

Βιβλιογραφία

Κατά τη συγγραφή της διπλωματικής μου εργασίας, πολύτιμες πληροφορίες, γνώσεις και ιδέες χρησιμοποιήθηκαν από τις ακόλουθες πηγές:

7.1 Βιβλιογραφικά Αναγνώσματα:

- [Σκο91] Ε. Σκορδαλάκης. Εισαγωγή στην Τεχνολογία Λογισμικού, Αθήνα 1991
- [Eck98] B. Eckel. Thinking in Java - Prentice Hall PTR 1998
- [Flu95] F. Fluckiger Understanding Networked Multimedia, Prentice Hall PTR 1995
- [Bel96] G. Bell, R. Carey, and C. Marrin: The Virtual Reality Modeling Language Specification: Version 2.0, 1996

7.2 *Links:*

- [L1] <http://www.vrml.org/>
[VRML official web site]
- [L2] <http://www.web3d.org/VRML2.0/FINAL/spec/index.html>
[VRML 2.0 specification]
- [L3] <http://www.ipv6.org>
[IPv6 web site]
- [L4] <http://java.sun.com/>
[Java web site]
- [L5] <http://www.internet2.edu>
[Internet 2 web site]
- [L6] <http://www.ietf.org>
[IETF web site]
- [L7] <http://public.web.cern.ch/Public/ACHIEVEMENTS/WEB/history.html>
[CERN web site]
- [L8] <http://www.w3.org>
[W3C web site]
- [L9] <http://java.sun.com/j2se/1.3/docs/tooldocs/win32/javac.html>
[javac]
- [L10] <http://java.sun.com/docs/books/tutorial/essential/system/runtime.html>
[java interpreter]
- [L11] <http://java.sun.com/j2se/1.4.2/docs/tooldocs/windows/java.html>
[javaw]
- [L12] <http://java.sun.com/products/jpda/doc/soljdb.html>
[jdb]
- [L13] <http://java.sun.com/j2se/1.3/docs/tooldocs/solaris/javap.html>
[javap]
- [L14] <http://java.sun.com/j2se/1.4.2/docs/tooldocs/javadoc/>
[javadoc]
- [L15] <http://www.vislabs.usyd.edu.au/vrml/repository.html>
[VRML repository]

- [L16] <http://developer.netscape.com/docs/manuals/communicator/jsref/index.htm> [Javascript reference]
- [L17] <http://developer.netscape.com/docs/javascript/> [Netscape Javascript reference]
- [L18] <http://www.w3.org/TR/html4/> [HTML 4.01 web site]
- [L19] <http://www.web3d.org/technicalinfo/specifications/vrml2.0/FINAL/spec/part1/fieldsRef.html> [VRML 2.0 syntax reference]
- [L20] <http://www.vrmlsite.com/feb97/a.cgi/spot2.html> [Java and VRML 2.0 Part 1: Basic theory]
- [L21] <http://www.vrmlsite.com/mar97/a.cgi/spot3.html> [Java and VRML 2.0 Part 2: Putting theory into practice]
- [L22] <http://www.fbi-medialab.fh-karlsruhe.de/persons/phennig/TBMM/Kap13/PW/README-pw> [pw README file]
- [L23] <http://www.tugraz.at> [Graz University of Technology]
- [L24] <http://www.borland.com/jbuilder/index.html> [JBuilder web site]
- [L25] <http://www.ultraedit.com/products/index.html> [Ultraedit web site]
- [L26] <http://ca.com/cosmo/> [Cosmo Player website]
- [L27] <http://www.sgi.com> [Silicon Graphics, Inc.]
- [L28] <http://www.igd.fhg.de/CP/> [Casus Presenter website]
- [L29] <http://www.igd.fhg.de> [Fraunhofer Institute for Computer Graphics]
- [L30] <http://www.microsoft.com/vrml/> [Microsoft VRML 2.0 Viewer website]

- [L31] <http://www.apple.com/quicktime/>
[Whirlwind 3D website]
- [L32] <http://www.community-place.com/>
[Community Place website]
- [L33] <http://www.sony.com>
[Sony Corporation]
- [L34] <http://www.cg.tuwien.ac.at/studentwork/CESCG98/FSixta/>
[Filip Sixta's VRML parser in Java]
- [L35] <http://decweb.ethz.ch/WWW6/Posters/734/>
[Fraunhofer Institute for Computer Graphics]
- [L36] <http://www.siggraph.org/>
[ACM SIGGRAPH]
- [L37] <http://www.acm.org>
[ACM publications website]
- [L38] <http://www.ieee.org>
[IEEE publications website]
- [L39] <http://ieeexplore.ieee.org/Xplore/DynWel.jsp>
[IEEE Explore]
- [L40] <http://www.sciencedirect.com/>
[ScienceDirect web site]
- [L41] <http://www.sun.com>
[Sun Microsystems web site]
- [L42] http://www.web3d.org/technicalinfo/specifications/ISO_IEC_14772-All/index.html
[VRML 97 specification]
- [L43] <http://www.installshield.com>
[InstallShield web site]

7.3 Request For Comment (RFC):

- [1] RFC-1766: Tags for the Identification of Languages
- [2] RFC-2854: The "text/html" Media type

7.4 Papers:

- [P1] «Java VRML EAI (External Authoring Interface)»
[<http://tecfa.unige.ch/guides/tie/pdf/files/java-vrml.pdf>]
- [P2] «The Virtual Reality Modeling Language and Java»
[<http://www.web3D.org/WorkingGroups/vrtp/docs/vrmljava.pdf>]
- [P3] A. Schäfer: «*Animation and Behaviour Description in VRML*», Diploma Thesis, Darmstadt University of Technology, Dept. of Computer Science, Interactive Graphics Systems Group, Darmstadt, 1996.
- [P4] «A VRML and Java-Based Interface for Retrieving VRML Content in Object-Oriented Databases»
[http://www.gris.informatik.tu-darmstadt.de/~nbraun/publications/1999_Retrieving_VRML_Content_in_Object-Oriented_Databases.pdf]
- [P5] Bo Huang and Hui Lin: «A Java/CGI approach to developing a geographic virtual reality toolkit on the Internet»
[http://www.sciencedirect.com/science?_ob=MIimg&_imagekey=B6V7D-44VDT4G-2-C&_cdi=5840&_orig=search&_coverDate=02%2F28%2F2002&_sk=999719998&view=c&wchp=dGLbVtz-zSkzV&_acct=C000006498&_version=1&_userid=604493&md5=1695d3e94d996d8691102e79203754de&ie=f.pdf]
- [P6] David Latch Craig and Craig Zimring: «Support for collaborative design reasoning in shared virtual spaces»
[http://www.sciencedirect.com/science?_ob=MIimg&_imagekey=B6V20-448Y6N4-D-K&_cdi=5688&_orig=search&_coverDate=02%2F28%2F2002&_sk=999889997&view=c&wchp=dGLbVtz-zSkzV&_acct=C000006498&_version=1&_userid=604493&md5=19c2596a3db389fdff048b75c07ad3c0&ie=f.pdf]
- [P7] H. Y. Kan, Vincent G. Duffy¹ and Chuan-Jun Su: «An Internet virtual reality collaborative environment for effective product design»
[http://www.sciencedirect.com/science?_ob=MIimg&_imagekey=B6V2D-4372W0G-7-W&_cdi=5700&_orig=search&_coverDate=06%2F30%2F2001&_sk=999549997&view=c&wchp=dGLbVtz-zSkzV&_acct=C000006498&_version=1&_userid=604493&md5=13339db38ff6189b76973a12c337272c&ie=f.pdf]

- [P8] Herbert Baerten and Frank Van Reeth1: «Using VRML and JAVA to visualize 3D algorithms in computer graphics education»
[http://www.sciencedirect.com/science?_ob=MImg&_imagekey=B6TYT-3VXYPF8-2-20&_cdi=5627&_orig=search&_coverDate=11%2F12%2F1998&_sk=999699979&view=c&wchp=dGLbVtz-zSkzV&_acct=C000006498&_version=1&_userid=604493&md5=33a26f88bca5fd1aa7d2ac1d75c75def&ie=f.pdf]
- [P9] Hui Lin, Jianhua Gong and Freeman Wang: «Web-based three-dimensional geo-referenced visualization»
[http://www.sciencedirect.com/science?_ob=MImg&_imagekey=B6V7D-3Y3PRH8-8-11&_cdi=5840&_orig=search&_coverDate=12%2F31%2F1999&_sk=999749989&view=c&wchp=dGLbVtz-zSkzV&_acct=C000006498&_version=1&_userid=604493&md5=f58b780f5c16e84276d10ec5293a3462&ie=f.pdf]
- [P10] Wolf-D. Ihlenfeldt: «Virtual Reality in Chemistry»
[http://link.springer.de/link/service/journals/00894/papers/7003009/70030386.pdf#xml=http://athene.em.springer.de/search97cgi/s97_cgi?action=view&VdkVgwKey=%2Fjour%2Fjour%2F00894%2Fpapers%2F7003009%2F70030386.pdf&doctype=xml&collection=springer02&queryZIP=VRML,Java]
- [P11] Luiz Carlos Guimarães, Rafael Garcia Barbastefano, Elizabeth Belfort: «Tools for teaching mathematics: A case for Java and VRML»
[<http://www3.interscience.wiley.com/cgi-bin/fulltext/75501338/PDFSTART>]
- [P12] Gerhard Reitmayr, Shane Carroll, Andrew Reitemeyer, Michael G. Wagner: «DeepMatrix - An open technology based virtual environment system»
[http://link.springer.de/link/service/journals/00371/papers/9015007/90150395.pdf#xml=http://athene.em.springer.de/search97cgi/s97_cgi?action=view&VdkVgwKey=%2Fjour%2Fjour%2F00371%2Fpapers%2F9015007%2F90150395.pdf&doctype=xml&collection=springer02&queryZIP=VRML,Java]
-