



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ  
& ΠΛΗΡΟΦΟΡΙΚΗΣ

## ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

---

« ΕΠΕΚΤΑΣΗ H-ANIM FRAMEWORK ΓΙΑ  
ΥΠΟΣΤΗΡΙΞΗ CUSTOM TEXTURES »

---

ΚΑΡΤΣΑΚΑΛΗΣ ΚΩΝΣΤΑΝΤΙΝΟΣ

A.M.: 3393

*ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ:*

*Χ. Μπούρας, Αναπληρωτής Καθηγητής*

ΠΑΤΡΑ 2009



---

# ΠΡΟΛΟΓΟΣ

---

Με την συνεχώς αυξανόμενη δημοφιλία και διάδοση των δικτυακών εικονικών περιβαλλόντων, έχει συντελεστεί μια εξίσου σημαντική μεταβολή στις απαιτήσεις των χρηστών που χρησιμοποιούν τα δικτυακά εικονικά περιβάλλοντα και αναζητούν σε αυτά ποιοτική λειτουργία και διευρυμένες δυνατότητες. Καθώς το 3D Internet συνεχίζει να εξαπλώνεται, θα παρουσιάζεται μια ολοένα αυξανόμενη ανάγκη για την ποιοτική αναπαράσταση ανθρώπινων αναπαραστάσεων σε αυτά. Η επίτευξη αυτού του στόχου, απαιτεί την κατασκευή μεγάλων βιβλιοθηκών από πρότυπα ανθρωποειδή (avatars) διαθέσιμα προς χρήση, καθώς και κατάλληλων εργαλείων που θα επιτρέπουν την κατασκευή νέων ανθρωποειδών από μηδενική βάση και την μοντελοποίηση της κίνησής τους με ποικίλους τρόπους. Τα παραπάνω συνετέλεσαν στην καθιέρωση του H-Anim, ενός προτύπου περιγραφής ανθρωποειδών για την γλώσσα περιγραφής 3d γραφικών X3D, καθορισμένων κατά τέτοιο τρόπο ώστε να παρέχουν συμβατότητα, ευελιξία και απλότητα.

Στα πλαίσια προηγούμενης κατά θεματική και ερευνητική σειρά διπλωματικής εργασίας, παρουσιάστηκε ένας ευέλικτος μηχανισμός παροχής βασικών λειτουργιών για το πρότυπο H-Anim. Συγκεκριμένα, παρουσιάστηκαν λειτουργίες επικύρωσης συμβατότητας με το H-Anim, ανίχνευσης επιπέδου Άρθρωσης (Level of Articulation) και εισαγωγής νέων Animations από κάποιο άλλο αρχείο συμβατό με το H-Anim. Στόχος της παρούσας διπλωματικής είναι η επέκταση της ιδέας της προηγούμενης διπλωματικής, ώστε να παρέχεται ένας πλήρης editor για το H-Anim, η χρήση του οποίου να είναι εύκολη, φιλική προς αυτόν και αποδοτική. Συγκεκριμένα, δίνεται έμφαση πλέον στην προσθήκη λειτουργιών που επιτρέπουν στον χρήστη να επεμβαίνει δυναμικά στην ιεραρχία του μοντέλου που επεξεργάζεται, καθώς και σε θέματα γεωμετρίας και εμφάνισης. Στόχος κατά την δημιουργία του editor είναι οι παραπάνω λειτουργίες να παρέχονται με όσο το δυνατόν πιο εύχρηστο και φιλικό προς τον χρήστη τρόπο, κατά συνέπεια καθιστώντας το H-Anim πλήρως επεξεργάσιμο χωρίς ο χρήστης να χρειαστεί να επεξεργαστεί απευθείας το script του avatar της επιλογής του.

Ο κύριος όγκος της εργασίας που ακολουθεί εστιάζει στην παρουσίαση των λειτουργιών που παρέχει το πρότυπο H-Anim, καθώς και στο πως ο editor που αναπτύχθηκε μπορεί να χειριστεί την καθεμία από τις λειτουργίες που παρέχονται από το H-Anim με εύχρηστο και φιλικό προς τον χρήστη τρόπο. Ιδιαίτερη έμφαση δόθηκε στο να τηρηθούν και από τον editor οι βασικοί σχεδιαστικοί στόχοι που προσδιορίστηκαν κατά τον ορισμό του προτύπου H-Anim: συμβατότητα, ευελιξία, απλότητα. Κάνουμε τις ελάχιστες δυνατές υποθέσεις και παραδοχές σχετικά με τον τύπο εφαρμογών που θα χρησιμοποιηθούν τα avatars που θα επεξεργαστεί/δημιουργήσει ο editor, καθώς και σχετικά με τα X3D primitives που χρησιμοποιούνται ως ενδιάμεσα για την σύνθεση ή τροποποίηση ενός avatar. Επιπρόσθετα, δόθηκε ιδιαίτερη σημασία στο να διατηρηθεί η παροχή λειτουργιών όσο το δυνατόν πιο απλή, απομακρύνοντας τον χρήστη από την απευθείας επεξεργασία του script του avatar όσο το δυνατόν περισσότερο.

Η δομή της εργασίας έχει ως εξής: τα πρώτα κεφάλαια αναφέρονται γενικά στην γλώσσα VRML και τον απόγονό της X3D, παρέχοντας στον αναγνώστη ένα στοιχειώδες υπόβαθρο ώστε να μπορεί στην συνέχεια να κατανοήσει πως χειρίζεται το H-Anim την δομή ενός VRML/X3D αρχείου για να καθορίσει μια ιεραρχία από αρθρώσεις. Εν συνεχεία, παρουσιάζουμε από θεωρητική ματιά τα avatars και ξεχωριστά το πρότυπο H-Anim, δομώντας το υλικό κατά τέτοιο τρόπο ώστε παράλληλα να γίνεται εμφανές πως το πρότυπο ανταποκρίνεται στην θεωρία των avatars για δικτυακά περιβάλλοντα. Ο στόχος ως εδώ είναι, εκτός από την παροχή επαρκούς υποβάθρου στον αναγνώστη για να κατανοήσει περαιτέρω τον χειρισμό του editor, το να παρέχουμε και έναν πλήρη οδηγό αναφοράς για κάποιον που θα ήθελε να εμπλουτίσει περαιτέρω τον ήδη υπάρχοντα κώδικα με πρόσθετα χαρακτηριστικά ή να δημιουργήσει κάποιον εκ νέου. Το τελευταίο μέρος αναφέρεται στην κατασκευή του editor, στην δομή και στα ιδιαίτερα χαρακτηριστικά χειρισμού του H-Anim, στον μηχανισμό πίσω από τις λειτουργίες, τις δυνατότητες επέκτασης και τις εγγενείς δυσκολίες που συναντήθηκαν κατά την συγγραφή του κώδικα, καθώς και το πώς αυτές προσπεράστηκαν.

Στο σημείο αυτό θα ήθελα να ευχαριστήσω όσους συνέβαλαν με γνώσεις, ιδέες και στήριξη στην εκπόνηση της παρούσας διπλωματικής εργασίας και κατέστησαν την με πολλή δουλειά υλοποίηση της αρχικής ιδέας δυνατή.

Θα ήθελα πρώτο από όλους να ευχαριστήσω τον επιβλέποντα της διπλωματικής εργασίας, Καθηγητή του ΤΜΗΥΠ κ. Χρήστο Μπούρα, ο οποίος παρείχε ουσιαστική καθοδήγηση, στήριξη, γνώσεις καθώς και ηθική υποστήριξη. Η βοήθειά του αποδείχθηκε ήδη από τα πρώτα βήματα πολύτιμη για όλες τις δύσκολες φάσεις εκπόνησης της διπλωματικής μου εργασίας. Του εύχομαι πάντα επιτυχίες σε ερευνητικό και προσωπικό επίπεδο.

Επίσης στα πλαίσια της συνεργασίας μου με το Εργαστήριο Κατανεμημένων Συστημάτων & Τηλεματικής του ΤΜΗΥΠ και την Ερευνητική Μονάδα 6 του ΕΑΙΤΥ θα ήθελα να ευχαριστήσω ιδιαίτερω τον επιβλέποντα μεταπτυχιακό φοιτητή Βασίλειο Τριγλιανό, ο οποίος προθύμως με στήριξε και με καθοδήγησε καθ' όλη την διάρκεια εκπόνησης της διπλωματικής εργασίας, παρέχοντας βοήθεια, γνώσεις, συμβουλές, κατευθυντήριες γραμμές και πρωτότυπες ιδέες. Η συμβολή του στην εκπόνηση της παρούσας διπλωματικής ήταν καθοριστική.

Οι τελευταίες ευχαριστίες ανήκουν αναμφισβήτητα στην οικογένειά μου, που με στήριξε πρωτίστως ψυχολογικά σε κάθε δυσκολία που παρουσιάστηκε κατά την διάρκεια εκπόνησης της διπλωματικής μου εργασίας, οικονομικά καθ' όλη την διάρκεια των προπτυχιακών μου σπουδών στο Τμήμα Μηχ. Η/Υ & Πληροφορικής αλλά και μου παρείχαν όλα τα απαραίτητα πνευματικά εφόδια, την επιμονή, την υπομονή, την εργατικότητα και την αυτοπειθαρχία χωρίς τα οποία δεν θα μπορούσα να ανταπεξέλθω στις απαιτήσεις της εργασίας. Οι παραπάνω ευχαριστίες αποτελούν μια ελάχιστη απόδοση ευγνωμοσύνης για όλα όσα τους χρωστώ.

Ανεξάρτητα από το ποσοστό στο οποίο συνέβαλαν οι παραπάνω στην διπλωματική εργασία, αναγνωρίζω ότι για τυχόν λάθη και προβλήματα δομής ή ουσίας του υλικού που παρουσιάζεται στα πλαίσια της εργασίας και πιθανόν να έχει ως ανθρώπινο πόνημα, μοναδικός υπεύθυνος και εγγυητής παραμένω ο συντάκτης της εργασίας.

Πάτρα, Αύγουστος 2009

Καρτσακάλης Κωνσταντίνος

# ΠΕΡΙΕΧΟΜΕΝΑ

## Περιεχόμενα

### ΚΕΦΑΛΑΙΟ 1: ΕΙΣΑΓΩΓΗ.....13

ΕΙΣΑΓΩΓΗ.....	14
ΥΠΟΨΗΦΙΕΣ ΔΗΜΟΣΙΕΥΣΕΙΣ ΣΕ ΣΥΝΕΔΡΙΑ .....	15

### ΚΕΦΑΛΑΙΟ 2: X3D.....16

2.1 ΕΙΣΑΓΩΓΗ.....	17
2.2 ΓΙΑΤΙ VRML; ΓΙΑΤΙ X3D; .....	17
2.3 ΣΧΕΣΗ X3D ΚΑΙ ΔΙΚΤΥΑΚΩΝ ΓΛΩΣΣΩΝ .....	18
2.4 X3D ΚΑΙ ΕΙΚΟΝΙΚΗ ΠΡΑΓΜΑΤΙΚΟΤΗΤΑ .....	19
2.4.1 Εικονική Πραγματικότητα.....	19
2.4.2 Εφαρμογές της Εικονικής Πραγματικότητας .....	20
2.4.3 Πως δουλεύει η Εικονική Πραγματικότητα .....	20
2.5 ΙΣΤΟΡΙΑ ΚΑΙ ΕΞΕΛΙΞΗ ΤΗΣ VRML, Η X3D.....	21
2.6 Βασικά Στοιχεία Χρήσης X3D .....	23
2.6.1 Ο κύκλος ζωής ενός αρχείου σε X3D.....	23
2.6.2 Μηχανές Rendering .....	24
2.6.2.1 Δομές Δεδομένων .....	24
2.6.2.2 Συστήματα Συντεταγμένων .....	24
2.6.2.3 Κορυφές.....	25
2.6.2.4 Ακμές.....	25
2.6.2.5 Όψεις.....	26
2.6.2.6 Αντικείμενα και Transforms .....	27
2.6.2.7 Επίπεδα Συστημάτων Συντεταγμένων.....	28
2.6.2.8 Ιεραρχίες .....	28
2.6.2.9 Διασωλήνωση των Γραφικών.....	29
2.6.3 X3D Φυλλομετρητές .....	31
2.6.3.1 Πως δουλεύει ένας φυλλομετρητής για X3D .....	31
2.6.3.2 Χειρισμός μιας σκηνής X3D .....	32
2.7 ΤΟ ΠΡΟΤΥΠΟ X3D ΚΑΙ ΕΦΑΡΜΟΓΕΣ.....	32
2.7.1 Βασική Σύνταξη X3D.....	32
2.7.1.1 Κόμβοι και πεδία .....	34
2.7.1.2 Ονομασία Κόμβων.....	34
2.7.2 ΑΝΤΙΚΕΙΜΕΝΑ, ΙΕΡΑΡΧΙΕΣ ΚΑΙ ΚΟΜΒΟΙ ΔΙΑΧΩΡΙΣΜΟΥ .....	34
2.7.3 ΚΟΜΒΟΙ TRANSFORMS .....	36
2.7.4 ΒΑΣΙΚΑ ΣΧΗΜΑΤΑ .....	37
2.7.4.1 Σφαίρα .....	38
2.7.4.2 Κώνος .....	38
2.7.4.3 Κύλινδρος.....	38
2.7.4.4 Κύβος.....	38
2.7.4.5 AsciiText ή Text.....	39
2.7.4.6 IndexedFaceSet.....	39
2.7.4.7 IndexedLineSet .....	40
2.7.4.8 PointSet.....	41
2.7.4.9 InfoSet .....	42
2.7.5 ΟΙ ΚΟΜΒΟΙ APPEARANCE ΚΑΙ MATERIAL.....	43
2.7.5.1 Κόμβος Appearance.....	43

2.7.5.2	Κόμβος Material .....	43
2.7.5.3	Κόμβος ColorInterpolator.....	43
2.7.5.4	Κόμβος ScalarInterpolator.....	44
2.7.5.5	Ένα σύνθετο παράδειγμα σε VRML/X3D .....	44
2.7.6	TEXTURES ΚΑΙ TEXTURE MAPPING .....	52
2.7.6.1	Κόμβοι Texture.....	52
2.7.6.2	Ενσωματώνοντας Εικόνες Texture σε ένα X3D αρχείο .....	55
2.7.6.3	Συντεταγμένες Texture .....	55
2.7.6.4	Ορισμός Transform για Συντεταγμένες Texture.....	57
2.7.6.5	Αντιστοίχιση Texture για ElevationGrids.....	57
2.7.7	ΓΕΓΟΝΟΤΑ ΚΑΙ ΔΡΟΜΟΛΟΓΗΣΗ.....	58
2.7.7.1	Γεγονότα.....	58
2.7.7.2	Δρομολόγηση με ROUTEs.....	59
2.8	VRML, X3D ΚΑΙ JAVA.....	60
2.8.1	Vrml, X3D, Java και EAI .....	60
2.8.2	Ενσωμάτωση στην Java.....	62
2.8.2.1	Σύγκριση VRML/X3D και JAVA .....	62
2.8.2.2	Συνδυάζοντας Java και VRML/X3D.....	62
2.8.3	Το πακέτο Xj3D.....	63
2.9	ΣΥΜΠΕΡΑΣΜΑΤΑ .....	64

### **ΚΕΦΑΛΑΙΟ 3: ΑΒΑΤΑΡΣ 65**

3.1	ΓΕΝΙΚΑ ΠΕΡΙ ΑΒΑΤΑΡΣ – ΕΦΑΡΜΟΓΕΣ.....	67
3.1.1	Χρήση των Avatars στην καθημερινότητα.....	67
3.1.1.1	Avatars σε Computer Games .....	68
3.1.1.2	Avatars σε περιβάλλοντα που δεν χρησιμοποιούνται για gaming .....	68
3.1.2	Ψηφιακά Μοντέλα .....	69
3.1.3	Μελλοντική Εξέλιξη και παράγοντες .....	70
3.2	ΤΡΙΣΔΙΑΣΤΑΤΑ ΑΒΑΤΑΡΣ ΚΑΙ ΕΙΚΟΝΙΚΟΣ ΚΟΣΜΟΣ .....	70
3.2.1	Προοπτικές Πρώτου και Τρίτου Προσώπου.....	70
3.2.2	Λειτουργίες των Avatars.....	71
3.2.3	Κατηγορίες Avatars – Agents.....	72
3.3	ΑΛΛΗΛΕΠΙΔΡΑΣΗ ΑΒΑΤΑΡΣ ΣΕ ΕΙΚΟΝΙΚΟΥΣ ΚΟΣΜΟΥΣ .....	72
3.3.1	Επικοινωνία χαρακτηριστικών του προσώπου (facial communication).....	73
3.3.2	Gesturing – Χειρονομίες.....	75
3.3.3	Αλληλεπίδραση του Avatar με αντικείμενα του χώρου .....	76
3.3.4	Κίνηση του Avatar στον Χώρο.....	76
3.4	ΡΕΑΛΙΣΜΟΣ .....	76
3.4.1	Συμπεριφορικός Ρεαλισμός .....	77
3.4.2	Φωτογραφικός Ρεαλισμός .....	77
3.5	ΠΡΟΧΩΡΗΜΕΝΑ ΘΕΜΑΤΑ .....	78
3.5.1	Αλγόριθμοι TSI - Ηθικά Ζητήματα .....	78
3.5.2	Η μεταμόρφωση της συμπεριφοράς .....	78
3.5.3	Έρευνα πάνω στα Avatars .....	79
3.6	ΣΥΜΠΕΡΑΣΜΑΤΑ .....	80

### **ΚΕΦΑΛΑΙΟ 4: ΤΟ ΠΡΟΤΥΠΟ Η-ANIM 81**

4.1	ΕΙΣΑΓΩΓΗ.....	83
4.2	ΤΙ ΕΙΝΑΙ ΤΟ Η-ANIM .....	83
4.2.1	Υπόβαθρο του Προτύπου .....	83
4.2.2	Σχεδιαστικοί Στόχοι.....	83
4.3	ΠΕΡΙΓΡΑΦΙΚΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ .....	84

4.4	H-ANIM FIGURES.....	84
4.4.1	Γενικά.....	84
4.4.2	Συναρμολόγηση.....	84
4.4.3	Ιεραρχία του Σκελετού.....	85
4.4.4	Χειρισμός.....	86
4.5	ΤΟ ΑΝΤΙΚΕΙΜΕΝΟ HUMANOID.....	86
4.5.1	Σκελετική Γεωμετρία.....	87
4.5.2	Skinned body Γεωμετρία.....	87
4.6	ΤΟ ΑΝΤΙΚΕΙΜΕΝΟ JOINT.....	87
4.7	ΤΟ ΑΝΤΙΚΕΙΜΕΝΟ SEGMENT.....	89
4.8	ΤΟ ΑΝΤΙΚΕΙΜΕΝΟ SITE.....	89
4.9	ΤΟ ΑΝΤΙΚΕΙΜΕΝΟ DISPLACER.....	90
4.10	ΜΟΝΤΕΛΟΠΟΙΗΣΗ.....	91
4.10.1	Γενικά.....	91
4.10.2	Μοντελοποίηση Ανθρωποειδών - Συμβάσεις.....	91
4.11	Η ΔΟΜΗ ΕΝΟΣ ΑΝΘΡΩΠΟΕΙΔΟΥΣ.....	93
4.11.1	Ιεραρχία Joints-Segments.....	93
4.11.2	Προσθέτωντας Joints.....	97
4.12	LEVEL OF ARTICULATION – ΕΠΙΠΕΔΑ ΑΡΘΡΩΣΗΣ.....	97
4.12.1	Επίπεδο Αρθρωσης 0.....	98
4.12.2	Επίπεδο Αρθρωσης 1.....	98
4.12.3	Επίπεδο Αρθρωσης 2.....	100
4.12.4	Επίπεδο Αρθρωσης 3.....	104
4.13	ΕΠΙΠΛΕΟΝ ΣΤΟΙΧΕΙΑ.....	109
4.13.1	Viewpoints.....	110
4.13.2	Πλοήγηση.....	110
4.13.3	Πολλαπλά ανθρωποειδή ανά Αρχείο.....	110
4.14	ΣΥΜΠΕΡΑΣΜΑΤΑ.....	111

## **ΚΕΦΑΛΑΙΟ 5: ΔΙΑΘΕΣΙΜΟΙ EDITORS 113**

5.1	ΔΙΑΘΕΣΙΜΟΙ EDITORS.....	113
5.2	ΣΧΕΔΙΑΣΤΙΚΟΙ ΣΤΟΧΟΙ ΓΙΑ ΤΟΝ H-ANIME.....	113
5.3	ΠΡΟΗΓΟΥΜΕΝΗ ΕΡΕΥΝΗΤΙΚΗ ΔΡΑΣΤΗΡΙΟΤΗΤΑ.....	114

## **ΚΕΦΑΛΑΙΟ 6: Ο EDITOR H-ANIME 117**

6.1	ΓΕΝΙΚΑ.....	117
6.2	ΠΑΡΕΧΟΜΕΝΕΣ ΛΕΙΤΟΥΡΓΙΕΣ.....	117
6.2.1	Φόρτωση, προ-φόρτωση και προβολή ενός Avatar.....	117
6.2.2	Edit Materials – Επεξεργασία Εμφάνισης.....	119
6.2.2.1	Texturing.....	120
6.2.3	Επεμβάσεις στην Γεωμετρία.....	121
6.2.3.1	Εισάγοντας από την βιβλιοθήκη.....	121
6.2.3.2	Ταιριάζοντας το αντικείμενο στο avatar.....	122
6.2.3.3	Επέκταση H-Anim Ιεραρχίας.....	125
6.2.4	VRML Parser.....	126
6.3	Ο ΜΗΧΑΝΙΣΜΟΣ ΤΟΥ H-ANIME.....	132
6.4	ΑΛΓΟΡΙΘΜΙΚΑ ΘΕΜΑΤΑ.....	133
6.4.1	Η Χρήση του Πακέτου Xj3D.....	133
6.4.2	Φόρτωση – Αποθήκευση σε Αρχείο.....	134
6.4.3	Βιβλιοθήκες.....	134
6.4.4	Προβλήματα, δυσκολίες και ελλείψεις του Xj3D.....	135
6.5	ΣΥΜΠΕΡΑΣΜΑΤΑ.....	136

**ΚΕΦΑΛΑΙΟ 7: ΣΥΜΠΕΡΑΣΜΑΤΑ 137**

7.1	ΣΥΜΠΕΡΑΣΜΑΤΑ .....	139
-----	--------------------	-----

**ΚΕΦΑΛΑΙΟ 8: ΜΕΛΛΟΝΤΙΚΗ ΕΡΓΑΣΙΑ 143**

8.1	ΔΗΜΙΟΥΡΓΙΑ Χ3D ΠΕΡΙΕΧΟΜΕΝΟΥ ΑΠΟ ΜΗΔΕΝΙΚΗ ΒΑΣΗ .....	143
8.2	ΕΠΕΚΤΑΣΗ ΤΗΣ ΒΙΒΛΙΟΘΗΚΗΣ ΑΝΤΙΚΕΙΜΕΝΩΝ .....	143
8.3	ΕΠΕΜΒΑΣΕΙΣ ΣΤΗ ΓΕΩΜΕΤΡΙΑ ΣΕ ΕΠΙΠΕΔΟ MESH .....	143
8.4	ΕΠΕΚΤΑΣΗ ΤΩΝ ΔΥΝΑΤΟΤΗΤΩΝ ΤΟΥ ΕΡΓΑΛΕΙΟΥ VRMLPARSER 144	

**ΠΑΡΑΡΤΗΜΑ Ι: ΒΙΒΛΙΟΓΡΑΦΙΑ 147**



## ΑΚΡΩΝΥΜΑ - ΟΡΟΙ

### Αγγλικοί όροι

3D	Three (3) Dimensional
AJAX	Asynchronous Java And XML
API	Application Programming Interface
ASCII	American Standard Code for Information Exchange
Audio	Ηχος
Avatar	Αναπαράσταση του Χρήστη
Browser	Φυλλομετρητής
Collaborative Workspace	Συνεργατικό Περιβάλλον Εργασίας
Composability	Συνθεσιμότητα
CPU	Computer Processing Unit
CVE	Collaborative Virtual Environment
EAI	External Authoring Interface
E-Commerce	Ηλεκτρονικό Εμπόριο
EVE	Educational Virtual Environment
Extensibility	Επεκτασιμότητα
H-Anim	Humanoid Animation Working Group, η βασει ISO προτυποποίηση για avatars
HTML	HyperText Markup Language
Humanoid	Ανθρωποειδές
Immersion	Εμβύθιση
Joint	Άρθρωμα
LoA	Level of Articulation – Επίπεδο Άρθρωσης
Mailing list	Λίστα Ηλεκτρονικής Αλληλογραφίας
Mesh	Γεωμετρικό Σχήμα
MPEG-4	Moving Pictures Expert Group – Προτυπο για Πολυμέσα
NVE	Networked Virtual Environment
SAI	Scene Authoring Interface
Scalability	Κλιμακωσιμότητα - κλίμακα
Segment	Μέλος/Τμήμα
Texture	Υφή

---

VE	Virtual Environment
VR	Virtual Reality
VRML	Virtual Reality Modeling Language, γλώσσα για την περιγραφή 3D υλικού
Web Document	Διαδικτυακό Αρχείο
WWW	World Wide Web, ο Παγκόσμιος Ιστός
X3D	Η γλώσσα-διάδοχος της VRML για την κωδικοποίηση 3D περιεχομένου
XML	eXtensible Markup Language

---

## ΠΙΝΑΚΑΣ ΕΙΚΟΝΩΝ

ΕΙΚΟΝΑ 1: ΠΑΡΑΔΕΙΓΜΑ ΧΡΗΣΗΣ ΤΟΥ Χ3D ΣΤΗΝ ΑΝΑΠΑΡΑΣΤΑΣΗ ΕΝΟΣ ΕΙΚΟΝΙΚΟΥ ΜΟΥΣΕΙΟΥ .....	19
ΕΙΚΟΝΑ 2: ΤΡΙΣΔΙΑΣΤΑΤΗ ΑΝΑΠΑΡΑΣΤΑΣΗ ΕΝΟΣ ΓΗΠΕΔΟΥ ΜΕ ΧΡΗΣΗ ΤΟΥ Χ3D.....	23
ΕΙΚΟΝΑ 3: ΠΑΡΑΔΕΙΓΜΑ ΣΥΣΤΗΜΑΤΟΣ ΣΥΝΤΕΤΑΓΜΕΝΩΝ ΓΙΑ 3 ΔΙΑΣΤΑΣΕΙΣ ΣΤΟΝ ΧΩΡΟ .....	25
ΕΙΚΟΝΑ 4: ΠΟΛΥΠΛΟΚΟ ΠΕΡΙΓΡΑΜΜΑ “WIREFRAME” ΓΙΑ ΤΟ ΣΧΗΜΑ ΚΑΡΕΚΛΑΣ, ΜΕ ΧΑΡΑΚΤΗΡΙΣΤΙΚΕΣ ΚΟΡΥΦΕΣ ΚΑΙ ΑΚΜΕΣ. ....	26
ΕΙΚΟΝΑ 5: ΦΩΤΙΣΜΟΣ ΣΕ ΜΙΑ 3Δ ΣΚΗΝΗ.....	29
ΕΙΚΟΝΑ 6: Ο ΦΥΛΛΟΜΕΤΡΗΤΗΣ CORTONA3D ΕΦΑΡΜΟΖΟΜΕΝΟΣ ΩΣ PLUG-IN ΣΤΟΝ ΦΥΛΛΟΜΕΤΡΗΤΗ ΙΣΤΟΥ ΜΟΖΙΛΛΑ FIREFOX .....	31
ΕΙΚΟΝΑ 7: ΠΡΟΒΑΛΛΟΝΤΑΣ ΕΝΑ ΑΠΛΟ ΣΧΗΜΑ ΚΥΛΙΝΔΡΟΥ ΣΕ ΦΥΛΛΟΜΕΤΡΗΤΗ.....	33
ΕΙΚΟΝΑ 8: ΤΟ ΟΠΤΙΚΟ ΑΠΟΤΕΛΕΣΜΑ ΕΝΟΣ INDEXEDLINESET .....	41
ΕΙΚΟΝΑ 9: ΤΟ ΟΠΤΙΚΟ ΑΠΟΤΕΛΕΣΜΑ ΕΝΟΣ POINTSET .....	42
ΕΙΚΟΝΑ 10: ΈΝΑ ΣΥΝΘΕΤΟ ΑΝΤΙΚΕΙΜΕΝΟ ΣΕ VRML ΚΑΙ Χ3D.....	45
ΕΙΚΟΝΑ 11: ΈΝΑΣ ΚΥΒΟΣ ΠΟΥ ΧΡΗΣΙΜΟΠΟΙΕΙ PIXELTEXTURE .....	54
ΕΙΚΟΝΑ 12: ΑΠΟΚΟΠΗ ΤΜΗΜΑΤΟΣ ΑΠΟ TEXTURE.....	56
ΕΙΚΟΝΑ 13: ΠΑΡΑΔΕΙΓΜΑ TEXTURING ΕΝΟΣ ΚΡΑΤΗΡΑ ΜΕ ΧΡΗΣΗ ELEVATIONGRID.....	58
ΕΙΚΟΝΑ 14: ΑΛΛΗΛΕΠΙΔΡΑΣΗ ΜΕΤΑΞΥ JAVA APPLET, ΕΑΙ ΚΑΙ VRML ΣΚΗΝΗΣ.....	61
ΕΙΚΟΝΑ 15: ΠΑΡΑΔΕΙΓΜΑ 3Δ ΑΝΑΠΑΡΑΣΤΑΣΗΣ ΤΗΣ ΚΙΝΗΣΗΣ ΜΕΤΟΧΩΝ ΣΤΟ ΧΡΗΜΑΤΙΣΤΗΡΙΟ .....	62
ΕΙΚΟΝΑ 16: ΤΡΙΣΔΙΑΣΤΑΤΑ ΑΒΑΤΑΡS .....	67
ΕΙΚΟΝΑ 17: ΠΡΟΟΠΤΙΚΗ ΠΡΩΤΟΥ ΚΑΙ ΤΡΙΤΟΥ ΠΡΟΣΩΠΟΥ .....	71
ΕΙΚΟΝΑ 18: ΈΝΑΣ AGENT ΞΕΝΑΓΕΙ ΤΟΝ ΧΡΗΣΤΗ ΣΤΟΥΣ ΧΩΡΟΥΣ ΤΟΥ ΕΙΚΟΝΙΚΟΥ ΠΕΡΙΒΑΛΛΟΝΤΟΣ .....	72
ΕΙΚΟΝΑ 19: VIDEO TEXTURING ΤΟΥ ΠΡΟΣΩΠΟΥ ΤΟΥ ΧΡΗΣΤΗ .....	73
ΕΙΚΟΝΑ 20: MODEL-BASED CODING ΕΚΦΡΑΣΕΩΝ ΤΟΥ ΠΡΟΣΩΠΟΥ .....	74
ΕΙΚΟΝΑ 21: ΠΡΟΚΑΘΟΡΙΣΜΕΝΕΣ ΕΚΦΡΑΣΕΙΣ ΠΡΟΣΩΠΟΥ – ΈΚΠΛΗΞΗ, ΝΥΣΤΑ, ΒΑΡΕΜΑΡΑ .....	75
ΕΙΚΟΝΑ 22: ΧΕΙΡΟΝΟΜΙΕΣ ΚΑΙ ΣΤΑΣΕΙΣ ΣΩΜΑΤΟΣ ΣΕ ΑΒΑΤΑΡS ΠΟΥ ΧΟΡΕΥΟΥΝ.....	76
ΕΙΚΟΝΑ 23: ΔΗΜΙΟΥΡΓΙΑ ΕΝΟΣ ΤΡΙΣΔΙΑΣΤΑΤΟΥ MESH ΣΕ ΣΧΗΜΑ ΚΕΦΑΛΙΟΥ ΚΑΙ ΕΠΙΚΑΛΥΨΗ ΤΟΥ ΜΕ ΕΝΑ ΦΩΤΟΓΡΑΦΙΚΟ TEXTURE ΠΟΥ ΑΠΟΔΙΔΕΙ ΤΗΝ ΕΜΦΑΝΙΣΗ ΤΟΥ ΑΝΘΡΩΠΟΥ .....	77
ΕΙΚΟΝΑ 24: ΕΞ΄ΟΡΙΣΜΟΥ ΣΤΑΣΗ ΕΝΟΣ ΑΝΘΡΩΠΟΕΙΔΟΥΣ ΓΙΑ ΤΟ H-ANIM.....	92
ΕΙΚΟΝΑ 25: ΠΡΟΣΑΝΑΤΟΛΙΣΜΟΣ ΧΕΡΙΟΥ .....	93
ΕΙΚΟΝΑ 26: ΟΡΙΣΜΟΣ ΙΕΡΑΡΧΙΑΣ JOINTS ΕΚ ΝΕΟΥ .....	115
ΕΙΚΟΝΑ 27: Η ΚΕΝΤΡΙΚΗ ΘΘΟΝΗ ΤΟΥ H-ANIME .....	118
ΕΙΚΟΝΑ 28: ΤΑ ΚΕΝΤΡΙΚΑ ΜΕΝΟΥ ΤΟΥ H-ANIME .....	118
ΕΙΚΟΝΑ 29: ΠΡΟΦΟΡΤΩΣΗ ΑΒΑΤΑΡ.....	119
ΕΙΚΟΝΑ 30: ΧΡΩΜΑΤΙΚΗ ΕΠΕΞΕΡΓΑΣΙΑ ΕΝΟΣ ΑΒΑΤΑΡ.....	120
ΕΙΚΟΝΑ 31: Η ΒΙΒΛΙΟΘΗΚΗ ΤΟΥ H-ANIME .....	121
ΕΙΚΟΝΑ 32: ΟΡΙΣΜΟΣ ΟΝΟΜΑΤΟΣ ΝΕΟΥ ΑΝΤΙΚΕΙΜΕΝΟΥ .....	122
ΕΙΚΟΝΑ 33: ΕΙΣΑΓΩΓΗ ΑΝΤΙΚΕΙΜΕΝΟΥ ΣΤΗΝ ΘΘΟΝΗ ΤΟΥ ΑΒΑΤΑΡ.....	123
ΕΙΚΟΝΑ 34: ΕΠΕΞΕΡΓΑΣΙΑ ΘΕΣΗΣ ΑΝΤΙΚΕΙΜΕΝΟΥ .....	123
ΕΙΚΟΝΑ 35: ΣΥΝΤΑΙΡΙΑΣΜΑ ΑΒΑΤΑΡ - ΑΝΤΙΚΕΙΜΕΝΟΥ.....	125
ΕΙΚΟΝΑ 36: 1 <sup>η</sup> ΘΘΟΝΗ/ΜΗΝΥΜΑ ΤΟΥ VRMLPARSER .....	127
ΕΙΚΟΝΑ 37: 2 <sup>η</sup> ΘΘΟΝΗ/ΜΗΝΥΜΑ ΤΟΥ VRMLPARSER .....	127
ΕΙΚΟΝΑ 38: 3 <sup>η</sup> ΘΘΟΝΗ/ΜΗΝΥΜΑ ΤΟΥ VRMLPARSER .....	127
ΕΙΚΟΝΑ 39: 4 <sup>η</sup> ΘΘΟΝΗ/ΜΗΝΥΜΑ ΤΟΥ VRMLPARSER .....	128
ΕΙΚΟΝΑ 40: Ο ΜΗΧΑΝΙΣΜΟΣ ΤΟΥ H-ANIME.....	132

---

## ΛΙΣΤΑ ΣΧΗΜΑΤΩΝ

---

ΠΙΝΑΚΑΣ 1: ΑΝΑΠΑΡΑΣΤΑΣΗ ΚΟΡΥΦΩΝ ΣΕ ΣΥΣΤΗΜΑ ΧΥΖ.....	26
ΠΙΝΑΚΑΣ 2: ΟΡΙΣΜΟΣ ΑΚΜΩΝ ΜΕ ΑΠΑΡΙΘΜΗΣΗ ΚΟΡΥΦΩΝ .....	27
ΠΙΝΑΚΑΣ 3: ΒΑΣΙΚΗ ΙΕΡΑΡΧΙΑ JOINTS.....	94
ΠΙΝΑΚΑΣ 4: ΙΕΡΑΡΧΙΑ JOINTS ΓΙΑ ΤΟ ΧΕΡΙ .....	94
ΠΙΝΑΚΑΣ 5: ΙΕΡΑΡΧΙΑ JOINTS ΓΙΑ ΤΟ ΠΡΟΣΩΠΟ .....	95
ΠΙΝΑΚΑΣ 6: ΕΠΙΠΕΔΟ ΑΡΘΡΩΣΗΣ 0 .....	98
ΠΙΝΑΚΑΣ 7: ΕΠΙΠΕΔΟ ΑΡΘΡΩΣΗΣ 1 .....	98
ΠΙΝΑΚΑΣ 8: ΕΠΙΠΕΔΟ ΑΡΘΡΩΣΗΣ 2 .....	100
ΠΙΝΑΚΑΣ 9: ΕΠΙΠΕΔΟ ΑΡΘΡΩΣΗΣ 3 .....	104

# ΚΕΦΑΛΑΙΟ 1: ΕΙΣΑΓΩΓΗ

---

## ΕΙΣΑΓΩΓΗ

---

Με δεδομένη την ολοένα και αυξανόμενη εξάπλωση και δημοφιλία των Δικτυακών Εικονικών Περιβαλλόντων πάνω από τον Παγκόσμιο Ιστό, οι χρήστες έχουν ανεβάσει τον πήχυ ως προς τις απαιτήσεις που έχουν για ρεαλισμό και εμπύθιση στα περιβάλλοντα που αρέσκονται να χρησιμοποιούν. Σήμερα παρά ποτέ, γίνεται ολοένα και πιο καθημερινό το φαινόμενο ο μέσος χρήστης ενός προσωπικού υπολογιστή να συνδέεται στον Ιστό από την άνεση του σπιτιού του και να επικοινωνεί με πρόσωπα της αρεσκείας του, κάνοντας χρήση εφαρμογών που μπορούν να παρέχουν με ελκυστικό τρόπο πλήρεις και λειτουργικές δυνατότητες επικοινωνίας με τους υπόλοιπους χρήστες.

Αντικείμενο της παρούσας εργασίας είναι η ανάπτυξη ενός editor για την επεξεργασία των avatars που χρησιμοποιεί ο καθημερινός χρήστης όταν συνδέεται σε κάποιο Δικτυακό Εικονικό Περιβάλλον. Συγκεκριμένα, στόχος της εργασίας ήταν η ανάπτυξη ενός editor φιλικού προς τον απλό χρήστη, ο οποίος να είναι εύκολος στην μάθηση και στην χρήση, και ο οποίος να παρέχει δυνατότητες ικανοποιητικών αισθητικών και λειτουργικών επεμβάσεων πάνω στα τρισδιάστατα avatars για την γλώσσα VRML/X3D, χωρίς να εμπλέξει καθόλου τον χρήστη σε θέματα συγγραφής script για την κωδικοποίηση των avatars. Θέσαμε ως ελάχιστο των απαιτήσεων το να μπορεί ο χρήστης, ο οποίος δεν είναι ενημερωμένος για τεχνικά θέματα που αφορούν την γλώσσα VRML/X3D, να μπορεί να επεξεργαστεί πλήρως το avatar της αρεσκείας του συνδυάζοντας απλά βασικά γεωμετρικά αντικείμενα.

Ο editor “H-AnimE” που αναπτύξαμε σχεδιάστηκε με κύριο γνώμονα την πλατφόρμα EVE II του εργαστηρίου Κατανεμημένων Συστημάτων & Τηλεματικής του Τμήματος Μηχανικών Η/Υ & Πληροφορικής. Συγκεκριμένα, η βασική ιδέα ήταν η επέκταση ενός παλαιότερου πλαισίου εργασίας που προσέφερε ευέλικτες λειτουργίες υποστήριξης των avatars πριν αυτά εισαχθούν σε κάποιο Δικτυακό Εικονικό Περιβάλλον, με περισσότερες δυνατότητες για ευρείας-κλίμακας αλλαγές πάνω στην εμφάνιση του επιλεγμένου avatar. Η εφαρμογή που αναπτύχθηκε υποστηρίζει πλήρως συμβατό με ISO VRML/X3D πρότυπο για την υποστήριξη των avatars, H-Anim.

Η εργασία δομείται σε κεφάλαια ως εξής:

- Στο Κεφάλαιο 2, αναφερόμαστε στα βασικά δομικά στοιχεία της γλώσσας VRML/X3D, αναλύοντας τους λόγους ανάπτυξής της ως VRML αρχικά και την μετέπειτα εξέλιξή της σε X3D. Δίνουμε στον αναγνώστη όλα τα απαραίτητα εργαλεία για να κατανοήσει την δομή ενός αρχείου VRML/X3D, ενώ σε ορισμένες περιπτώσεις εμβαθύνουμε σε ορισμένα σημεία για να δώσουμε στον χρήστη το απαραίτητο υπόβαθρο ώστε να κατανοήσει σε μετέπειτα κεφάλαιο το πρότυπο H-Anim για την προτυποποίηση των avatars. Παρουσιάζονται επίσης και ορισμένα ενδιαφέροντα στοιχεία σχετικά με το πως οι μηχανές rendering και οι φυλλομετρητές προβάλλουν το 3d υλικό στην οθόνη του χρήστη.
- Στο Κεφάλαιο 3, επιδιώκουμε μια εποπτική, μη-τεχνική προσέγγιση πάνω στα avatars που χρησιμοποιούνται στα Τρισδιάστατα Εικονικά Περιβάλλοντα. Αναλύουμε θέματα εφαρμογών των avatars, θέματα εμφάνισης, αισθητικής και ρεαλισμού, καθώς και κάποια βασικά στοιχεία από την ψυχολογία των avatars και ηθικά ζητήματα που προκύπτουν από την κακόβουλη χρήση τους.

- Στο Κεφάλαιο 4, παρουσιάζουμε λεπτομερώς το πρότυπο H-Anim. Το Κεφάλαιο μπαίνει σε πολλές τεχνικές λεπτομέρειες σχετικά με το πρότυπο και την προσπάθεια που καταβλήθηκε ώστε να πληρούνται από αυτό οι βασικοί σχεδιαστικοί του στόχοι. Αναλύουμε θέματα άρθρωσης σκελετού, γεωμετρίας και προσομοίωσης κίνησης, καθώς και θέματα κωδικοποίησης για VRML και X3D.
- Στο Κεφάλαιο 5, αναλύουμε υπάρχουσες εργασίες που έχουν ήδη γίνει και διατίθενται στον παγκόσμιο Ιστό, οι οποίες να περιστρέφονται γύρω από το πρότυπο H-Anim. Εξετάζουμε πως η καθεμιά ανταποκρίνεται στις ανάγκες του μέσου χρήστη και τις δυσκολίες που θα μπορούσε να συναντήσει κατά τον σχεδιασμό ή την επεξεργασία ενός avatar ανάλογα με την εμπειρία του. Τέλος, διατυπώνουμε την βασική πρόκληση για τον σχεδιασμό του editor της εργασίας, του H-AnimE.
- Στο Κεφάλαιο 6, παρουσιάζουμε τον editor H-AnimE: τις λειτουργίες που παρέχει, τις δυνατότητες για επεξεργασία και σύνθεση ενός avatar, καθώς και το πως χειρίζεται ο κώδικας του H-AnimE διάφορες δυσκολίες που προκύπτουν κατά την ανάγνωση, την προβολή και την αποθήκευση του avatar χρησιμοποιώντας το πακέτο Xj3D.
- Στο Κεφάλαιο 7, διατυπώνουμε κάποια χρήσιμα συμπεράσματα που εξάγονται με βάση την εργασία πάνω στο H-Anim.
- Στο Κεφάλαιο 8, αναλύουμε κάποιες πιθανές ευκαιρίες για μελλοντική εργασία, ή κάποιες βελτιώσεις χαρακτηριστικών που θα γινόταν εφικτές σε (πιθανή) εξέλιξη των δυνατοτήτων του πακέτου Xj3D στο μέλλον.
- Το Παράρτημα I, περιλαμβάνει αναφορές σε ερευνητικές εργασίες, που συνετέλεσαν στην υλοποίηση και συγγραφή της παρούσας διπλωματικής.

---

## ΥΠΟΨΗΦΙΕΣ ΔΗΜΟΣΙΕΥΣΕΙΣ ΣΕ ΣΥΝΕΔΡΙΑ

---

ΚΕΦΑΛΑΙΟ 2: X3D



---

# X3D

---

## 2.1 ΕΙΣΑΓΩΓΗ

Η VRML - Virtual Reality Modeling Language, γνωστή και ως Virtual Reality Markup Language είναι μια προτυποποίηση για τύπους αρχείων που αναπαριστούν 3D αλληλεπιδραστικά διανυσματικά γραφικά, σχεδιασμένα ειδικότερα για να υποστηρίξουν διαδικτυακές εφαρμογές. (1)

Το X3D είναι το βάσει ISO (2), βασιζόμενο στην XML, πρότυπο τύπων αρχείου για την αναπαράσταση τρισδιάστατων γραφικών, το οποίο έχει διαδεχθεί την VRML ως το κατεξοχήν πρότυπο για αναπαράσταση 3D γραφικών για διαδικτυακές εφαρμογές. Παρουσιάζει σημαντικές βελτιώσεις σε σχέση με την VRML, πχ. Προσομοίωση Κίνησης Ανθρωποειδών, GeoVRML, NURBS, κλπ, καθώς και την ικανότητα να κωδικοποιεί την ίδια 3d σκηνή χρησιμοποιώντας σύνταξη XML.

Εφεξής, στις περισσότερες περιπτώσεις θα αναφερόμαστε και στα δυο πρότυπα μαζί ως VRML/X3D, καθώς η μετάβαση από το VRML πρότυπο στο X3D έγινε με ταυτόχρονη διατήρηση όλων των λειτουργιών της VRML, με έναν προς τα άνω συμβατό τρόπο.

Κρίθηκε σκόπιμο, επειδή έχει παρέλθει πλέον περισσότερο από μια δεκαετία από την δημιουργία της αρχικής έκδοσης της VRML (3), να εστιάσουμε στις δυνατότητες της τρέχουσας εκδοχής της, το X3D.

## 2.2 ΓΙΑΤΙ VRML; ΓΙΑΤΙ X3D;

Η HTML διευκολύνει ένα μεγάλο αριθμό διαδικασιών που συμβαίνουν στο δίκτυο. Εντούτοις, δεν υποστηρίζεται η αλληλεπίδραση του χρήστη με τρισδιάστατα γραφικά περιβάλλοντα, ένα κενό το οποίο ήρθε να συμπληρώσει η standard VRML. Οι εφαρμογές της VRML είναι πολλές, κατά κύριο λόγο: επιχειρήσεις, e-commerce, ψυχαγωγία, εκπαίδευση, επικοινωνία, αρχιτεκτονική, κλπ. Ιδιαίτερη σημασία έχει η συμβολή της VRML στην υποστήριξη τρισδιάστατων collaborative workspaces (συνεργατικά περιβάλλοντα εργασίας) και virtual on-line communities (πολυχρηστικών on-line κοινωνιών). (4)

Συγκρίνοντας HTML και VRML, μπορούμε να παρατηρήσουμε τα εξής:

- Ενώ η HTML υποστηρίζει την χρήση raster εικόνων και μόνο, η VRML παρέχει δυνατότητες χρήσης δισδιάστατου και τρισδιάστατου περιεχομένου (στο οποίο μπορεί να συμπεριληφθούν και raster εικόνες), αλλά και αλληλεπίδραση σε δισδιάστατο ή τρισδιάστατο πλαίσιο με το περιεχόμενο αυτό. Οι δυνατότητες αναπαράστασης οπτικού περιεχομένου της VRML υπερκαλύπτουν αυτές της HTML.
- Ο τύπος αρχείου της VRML και του X3D έχει σχεδιαστεί έτσι ώστε να μπορεί να περιγράφει την μορφή του τρισδιάστατου περιεχομένου (μία σειρά από τρισδιάστατα γεωμετρικά μοντέλα), αλλά και την συμπεριφορά των μοντέλων αυτών που προκύπτει από την αλληλεπίδραση του χρήστη με το εικονικό περιβάλλον.

Με την χρήση ενός VRML/X3D κόσμου, προσφέρονται στον χρήστη οι εξής δυνατότητες:

- Πλοήγησης του ιδίου (μεταβάλλοντας την θέση της οπτικής του) μέσα στο τρισδιάστατο περιβάλλον.
- Μεταβολής της θέσης αντικειμένων σε ένα τρισδιάστατο εικονικό περιβάλλον, δηλαδή της αλληλεπίδρασης με αυτά, κρατώντας σταθερή την οπτική του προς το περιβάλλον.

Ο χρήστης μπορεί να παρατηρεί και να αντιλαμβάνεται αντικείμενα και φαινόμενα με έναν περισσότερο φυσιολογικό και «ενστικτώδη» τρόπο, εκμεταλλευόμενος τις αντιληπτικές και νοητικές του ικανότητες του για πλοήγηση, χειρισμό αντικειμένων και κατανόηση φαινομένων που συμβαίνουν σε ένα τρισδιάστατο χωρικό πλαίσιο.

Ως διάδοχος της VRML, το X3D βελτιώνει την VRML με πολλές νέες λειτουργικότητες, αναβαθμισμένα APIs, επιπρόσθετες κωδικοποιήσεις δεδομένων, πιο «σφιχτά» πλαίσια συμβατότητας, και μια τμηματοποιημένη αρχιτεκτονική η οποία χρησιμοποιεί profiles. Η X3D έχει υλοποιηθεί με έναν προς τα άνω συμβατό τρόπο σε σχέση με τις εκδόσεις της VRML.

Τα επιπρόσθετα χαρακτηριστικά του X3D περιλαμβάνουν:

- Λογισμικό ανοιχτού κώδικα, κατά συνέπεια δεν υπάρχουν θέματα αδειών.
- Έχει ενσωματωθεί πλήρως μέσα στο προτυπο MPEG-4 για πολυμέσα.
- Το X3D υποστηρίζει την κωδικοποίηση της γλώσσας XML. Είναι ευκολότερο κατά συνέπεια με χρήση του X3D να ενσωματωθούν 3D δεδομένα σε υπηρεσίες Web και κατανεμημένες εφαρμογές.
- Είναι συμβατό με την επόμενη γενιά αρχείων γραφικών, πχ Scalable Vector Graphics.
- Τα 3D αντικείμενα μπορούν να υποστούν επεξεργασία με χρήση γλωσσών όπως η C, η C++ και η Java (όπως στην παρούσα διπλωματική).

## 2.3 ΣΧΕΣΗ X3D ΚΑΙ ΔΙΚΤΥΑΚΩΝ ΓΛΩΣΣΩΝ

Απο πολλές απόψεις, το X3D μπορεί να παραλληλιστεί με την HTML περισσότερο από οποιαδήποτε άλλη γλώσσα. (5) Όπως και η HTML, η X3D χρησιμοποιεί την τυποποίηση του ASCII για τα αρχεία της. Και όπως και η HTML, η X3D είναι μια γλώσσα περιγραφής την οποία επεξεργάζεται ένας φυλλομετρητής (browser). Αλλά, ενώ η HTML παράγει μια στατική διεπαφή η οποία θα πρέπει να χρησιμοποιήσει άλλες εφαρμογές για να προσδώσουν στις σελίδες δυναμικά στοιχεία, η X3D είναι δυναμική από την ίδια την φύση της. Η X3D δεν περιγράφει την τυποποίηση μια σελίδας, αλλά την χωροθέτηση ενός τρισδιάστατου περιβάλλοντος και των αντικειμένων του.

Ένας τρόπος για να στοιχειωθεθεί η διαφορά μεταξύ HTML και X3D είναι το να σκεφτούμε την διαφορά μεταξύ ενός σχεδίου κτιρίου στο χαρτί και ενός μοντέλου του ίδιου κτιρίου. Δεν μπορούμε να χειριστούμε τα περιεχόμενα μιας καθορισμένης δισδιάστατης οντότητας, όπως μιας σελίδας. Μπορούμε να χειριστούμε ωστόσο ένα τρισδιάστατο μοντέλο ώστε να έχουμε την άποψή του από διαφορετικές προοπτικές. Η X3D προσθέτει ένα επιπρόσθετο επίπεδο διαδραστικότητας επιτρέποντας στον χρήστη να παρέχει συνδέσμους προς πόρους του διαδικτύου. Φανταστείτε ένα υποθετικό σενάριο στο οποίο ο χρήστης περιηγείται στους χώρους ενός 3-διάστατου μουσείου, βλέπει ένα αντικείμενο που τον ενδιαφέρει, και μπορεί να έχει πρόσβαση σε πληροφορίες πάνω σε αυτό ή να του εμφανίζονται άλλα σχετικά με αυτό αντικείμενα απλά δείχνοντας προς το αντικείμενο με τον δείκτη του ποντικιού.



**Εικόνα 1:** Παράδειγμα χρήσης του X3D στην αναπαράσταση ενός Εικονικού Μουσείου

Η X3D παρέχει τέτοιου είδους δυνατότητες, επιτρέποντας την εμφάνιση ενός 3διάστατου αντικειμένου και την υπερσύνδεσή του προς ένα δια-δικτυακό αρχείο (web document) ή κάποιο άλλο X3D αντικείμενο ή περιβάλλον. Εκεί που ο χρήστης θα γυρνούσε τις σελίδες ενός βιβλίου για να βρει πληροφορίες, με την χρήση της X3D μπορεί να περπατήσει μέσα σε μια τρισ-διάστατη αναπαράσταση της πληροφορίας.

## 2.4 X3D ΚΑΙ ΕΙΚΟΝΙΚΗ ΠΡΑΓΜΑΤΙΚΟΤΗΤΑ

Η γλώσσα X3D είναι στενά διασυνδεδεμένη με την Εικονική Πραγματικότητα. Πριν προχωρήσουμε περαιτέρω στην ανάλυση της σχέσης των δυο, κρίνεται σκόπιμο να αναφερθούν ορισμένα βασικά στοιχεία για την Εικονική Πραγματικότητα. (6)

### 2.4.1 *Εικονική Πραγματικότητα*

Σαν πρώτο ορισμό κρίνεται σκόπιμο να αναφερθεί ο παρεχόμενος από την εγκυκλοπαίδεια Britannica: «Η Εικονική Πραγματικότητα αποτελεί την χρήση της μοντελοποίησης και της προσομοίωσης μέσω υπολογιστικών συστημάτων προκειμένου να δώσει στον χρήστη την δυνατότητα να αλληλεπιδράσει με ένα τεχνητό τρισδιάστατο οπτικό περιβάλλον. Οι εφαρμογές της Εικονικής Πραγματικότητας εμβυθίζουν τον χρήστη σε ένα υπολογιστικό περιβάλλον το οποίο προσομοιώνει την πραγματικότητα μέσω της χρήσης ειδικών συσκευών, οι οποίες στέλνουν και λαμβάνουν πληροφορία.» (7)

Κάποιες άλλες αξιολογες διατυπώσεις που αποδίδουν σωστά τον όρο είναι οι ακόλουθες: «Προσομοιώσεις σε ηλεκτρονικούς υπολογιστές οι οποίες χρησιμοποιούν τρισδιάστατα γραφικά και περιφερειακές συσκευές όπως το γάντι δεδομένων (data glove) οι οποίες επιτρέπουν στους χρήστες να αλληλεπιδρούν με την προσομοίωση.» και «Η Εικονική Πραγματικότητα είναι μια προσομοίωση ενός πραγματικού ή φανταστικού περιβάλλοντος το οποίο μπορεί να αναπαρασταθεί εικονικά σε τρεις

διαστάσεις, πλάτους, ύψους και βάθους που μπορεί επίσης να παρέχει μια αλληλεπιδραστική interactive εμπειρία με κίνηση πραγματικού χρόνου, ενσωμάτωση ήχου και αλληλεπίδρασης. Η απλούστερη μορφή Εικονικής Πραγματικότητας είναι μια τρισδιάστατη εικόνα η οποία μπορεί να εξεταστεί αλληλεπιδραστικά σε έναν προσωπικό υπολογιστή, συνήθως με τη χρησιμοποίηση του πληκτρολογίου ή του ποντικιού, έτσι ώστε η εικόνα να μετακινείται προς την κατεύθυνση που τις υποδεικνύουν τα πλήκτρα ή το ποντίκι αντίστοιχα. Όσο οι εικόνες αυτές μεγαλώνουν και οι έλεγχοι για την αλληλεπίδραση αυξάνονται, τόσο αυξάνεται και η αντίληψη της «πραγματικότητας».

Ένα κοινό σημείο μεταξύ των παραπάνω ορισμών είναι ότι δίνεται έμφαση σε δυο κύρια σημεία: το πρώτο είναι το στοιχείο της προσομοίωσης ενός πραγματικού ή φανταστικού περιβάλλοντος, το οποίο, μέσω της αναπαράστασης, προσπαθεί να εξομοιώσει την αναπαριστώμενη οντότητα με ρεαλιστικό τρόπο. Το δεύτερο είναι ότι στην διαδικασία εμπλέκονται τα υπολογιστικά συστήματα.

### **2.4.2 Εφαρμογές της Εικονικής Πραγματικότητας**

Έχοντας διατυπώσει μερικούς στοιχειώδεις ορισμούς σχετικά με το τι αφορά η Εικονική Πραγματικότητα, στην παρούσα υποενότητα παρουσιάζουμε μια γενική αναφορά σχετικά με χρήσεις και εφαρμογές της Εικονικής Πραγματικότητας, και κατ' επέκταση για το που θα μπορούσε να συνεισφέρει η χρήση της VRML για την αναπαράσταση της εικονικής πληροφορίας.

Παρ' όλους τους περιορισμούς και το σχετικά πρόσφατο της ανάπτυξης της, η τεχνολογία της Εικονικής Πραγματικότητας χρησιμοποιείται σε ένα πλήθος διαφορετικών περιοχών. Ενδεικτικά αναφέρουμε: στην τρισδιάστατη αναπαράσταση κτηρίων για την πολιτική μηχανική, προτού αυτά υλοποιηθούν, ώστε να παρέχεται στον πελάτη μια πιο κατατοπιστική εικόνα για το τι ακριβώς θα πρέπει να περιμένει από τον μηχανικό να υλοποιήσει, στην μάθηση από απόσταση, στην οποία η Εικονική Πραγματικότητα χρησιμοποιείται για την παροχή εικονικών τάξεων οι οποίες υπάρχουν μόνο στα πλαίσια του εικονικού κόσμου, καθώς και στην παρουσίαση του σχετικού εκπαιδευτικού υλικού, στην συνεργασία από απόσταση πχ για ιατρικούς σκοπούς, στην προετοιμασία κάποιας περίπλοκης εγχείρισης για την οποία γιατροί και ασθενής δεν βρίσκονται στον ίδιο γεωγραφικό χώρο και η εικονική πραγματικότητα χρησιμοποιείται για να παρέχει αντίληψη της εικόνας (βλέπε: την πρόσφατη εμφάνιση του Ιατρικού VRML). Επίσης άξιο αναφοράς είναι το ενδιαφέρον χρήσης της εικονική πραγματικότητα για αισθητικούς/καλλιτεχνικούς λόγους.

### **2.4.3 Πως δουλεύει η Εικονική Πραγματικότητα**

Στην παρούσα φάση ανάπτυξης της X3D, οι φυλλομετρητές είναι κατά βάση απλά εργαλεία για την εμφάνιση τρισδιάστατων αντικειμένων και την αλληλεπίδραση με αυτά στην οθόνη του υπολογιστή. Η προσθήκη επιπρόσθετων λειτουργιών προγραμματιστικότητας, συμπεριφοράς και δικτύωσης αναμένεται ωστόσο να προσδώσει στους φυλλομετρητές για X3D αισθητά μεγαλύτερη αποδοτικότητα, μετατρέποντάς τους σε πλήρη συστήματα αναπαράστασης Εικονικής Πραγματικότητας. Κρίνουμε σκόπιμο να αναφερθούμε συνοπτικά σε ορισμένα στοιχεία λειτουργίας ενός συστήματος Εικονικής Πραγματικότητας.

Η χρήση ενός συστήματος Εικονικής Πραγματικότητας είναι αρκετή για να παρέχει στον χρήστη ευκολία και απλότητα, κρύβοντας τα περίπλοκα στοιχεία που απαιτούνται για την εκτέλεση των φαινομενικά απλών λειτουργιών. Κάθε σύστημα Εικονικής Πραγματικότητας εκτελεί τις εξής βασικές λειτουργίες:

- Διατηρεί μια αναφορά κατάστασης για κάθε αντικείμενο στον εικονικό κόσμο.
- Αποθηκεύει και ενημερώνει πληροφορία σχετικά με την τοποθεσία και την εμφάνιση του κάθε αντικειμένου.
- Εξομοιώνει την συμπεριφορά των αντικειμένων.
- Παρουσιάζει (rendering) τον κόσμο σε τρεις διαστάσεις.
- Παράγει ήχους για τα εικονικά αντικείμενα (προαιρετικό).
- Επιτρέπει στον χρήστη να περιηγείται/πλοηγείται μέσα στο εικονικό περιβάλλον.
- Παρέχει τον χρήστη με τα μέσα αλληλεπίδρασης με το περιβάλλον.

Για να εκτελεστούν τα παραπάνω με αποδοτικό τρόπο, σε αρκετά συστήματα Εικονικής Πραγματικότητας χρησιμοποιείται επιπλέον υλικό. Συγκεκριμένα, η πλοήγηση και η αλληλεπίδραση με το περιβάλλον επιβάλλει για λόγους ρεαλισμού την χρήση ειδικών συσκευών (εισόδου) όπως γάντια και ειδικά γυαλιά (πχ head-mounted displays). Ωστόσο, στην πραγματικότητα, δεν απαιτείται καμία από τις παραπάνω συσκευές ή κάποιο άλλο υλικό ειδικού-σκοπού, και ένα σύστημα Εικονικής Πραγματικότητας μπορεί να είναι σε θέση να παρέχει μια εμπειρία εμβύθισης (immersion) χωρίς επιπλέον υλικό πέρα από αυτό που παρέχει ο υπολογιστής.

## 2.5 ΙΣΤΟΡΙΑ ΚΑΙ ΕΞΕΛΙΞΗ ΤΗΣ VRML, Η X3D

Το 1994, οι Mark Pesce και Tony Parisi ανέπτυξαν με βάση το πακέτο λογισμικού RealityLab μια πρότυπη επέκταση του WWW με τίτλο «Labyrinth», η οποία επέτρεπε στο WWW να συνδέεται σε αρχεία 3-διάστατων γραφικών, προσθέτοντας έτσι στον ήδη υπάρχοντα τρόπο αλληλεπίδρασης του χρήστη με τον παγκόσμιο ιστό μια Τρίτη διάσταση, μεταξύ χρήστη και κάθε αντικειμένου στην σκηνή. Το Labyrinth επέτρεπε πέραν από την καθιερωμένη κίνηση δεξιά-αριστερά και πάνω-κάτω μέσα σε μια γραφική εικόνα, την κίνηση οπουδήποτε γύρω από την εικόνα, από όλες τις πλευρές. (8)

Ο Labyrinth δεν ήταν ο πρώτος φυλλομετρητής για VRML, δεδομένου ότι η VRML δεν είχε αναπτυχθεί ακόμα, ωστόσο παρείχε την ερευνητική κατεύθυνση η οποία κατέληξε στην VRML. Ο Labyrinth είχε την δυνατότητα να παρουσιάσει τρισδιάστατα αντικείμενα πάνω από τον Ιστό, χρησιμοποιώντας τα ίδια πρωτόκολλα που χρησιμοποιούσαν οι καθιερωμένες σελίδες του Παγκοσμίου Ιστού. Ο Labyrinth παρουσιάστηκε στο 1<sup>ο</sup> Διεθνές Παγκόσμιο Συνέδριο πάνω στον WWW τον Μάιο του 1994 στην Γενεύη της Ελβετίας, το οποίο ακολούθησε η δημιουργία της πρώτης ηλεκτρονικής mailing list συζητήσεων πάνω στην “Virtual Reality Markup Language”, την VRML, η οποία και ονομάστηκε κατ’ αντιστοιχίαν με την HTML – “HyperText Markup Language”. Την λέξη Markup αργότερα αντικατέστησε η λέξη “Modeling” – μοντελοποίηση, καθώς θεωρήθηκε πιο κατάλληλη για την περιγραφή της VRML. Σύντομα δημιουργήθηκε η πρώτη προτυποποίηση για την VRML, η οποία σχεδιάστηκε από τους Gavin Bell, Mark Pesce και Tony Parisi.

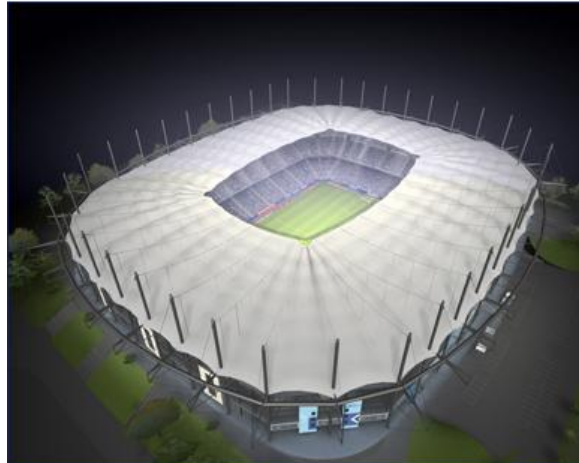
Το γενικό προσχέδιο για την προτυποποίηση της VRML παρουσιάστηκε στο 2<sup>ο</sup> Διεθνές Συνέδριο του WWW τον Οκτώβριο του 1994 στο Chicago. Μερικούς μήνες αργότερα, έγινε διαθέσιμος για το ευρύ κοινό ένας λεκτικός επεξεργαστής για την γλώσσα VRML, ο οποίος αποκαλούνταν QvLib, ο οποίος και θεωρήθηκε σημείο εκκίνησης για την μετέπειτα ιστορία συγγραφής εργαλείων VRML. Έκτοτε έχουν γίνει αρκετές αλλαγές στο αρχικό πρότυπο, με άξια αναφοράς την έκδοση 1.1 του προτύπου η οποία επιζητούσε να επιλύσει πολλά προβλήματα υλοποίησης σε διάφορα συστήματα, και να ξεκαθαρίσει αρκετά ασαφή ζητήματα πάνω στο πρότυπο. Η VRML1.1 πρωτοεισήγαγε audio για την υποστήριξη των 3d γραφικών, καθώς και μια στοιχειώδη υποστήριξη για την προσομοίωση κίνησης.

Οι πρώτες κατευθύνσεις για την ανάπτυξη της VRML2.0 ήταν οι εξής: συνθεσιμότητα (composability), κλίμακα (scalability) και επεκτασιμότητα (extensibility). (9) Η συνθεσιμότητα θα επέτρεπε στον χρήστη να δημιουργήσει ένα σπίτι, να το αυξομειώσει σε μέγεθος και να το τοποθετήσει πάνω σε ένα τραπέζι. Το τραπέζι αυτό μαζί με το μοντέλο εν συνεχεία θα τοποθετούνταν στην κορυφή ενός άλλου μοντέλου και το όλο σύστημα πάνω σε κάποιο άλλο κλπ το οποίο θα σήμαινε ότι σε αυτή την σύνθεση το κάθε αντικείμενο θα είναι ανεξάρτητο από τα υπόλοιπα.

Η ιδιότητα της κλίμακας επιτρέπει την δημιουργία κόσμων με μεταβαλλόμενο μέγεθος. Με την VRML θα πρέπει να είναι εύκολο να προσεγγίσουμε έναν γαλαξία, να εστιάσουμε σε ένα αστρικό σύστημα, κατ' όπιν σε έναν πλανήτη, σε μια πόλη, ένα πάρκο, έναν άνθρωπο που κάθεται σε έναν πάγκο, και σε ακόμη μεγαλύτερα επίπεδα λεπτομέρειας πάνω στον άνθρωπο. Το παραπάνω είναι δύσκολο καθώς περιορίζεται σημαντικά από το διαθέσιμο υλικό του υπολογιστή, ωστόσο με το πρότυπο δόθηκε έμφαση στο να μην μπαίνουν περιορισμοί και όρια στα μεγέθη των κόσμων και στην επιτρεπόμενη λεπτομέρεια. Η επεκτασιμότητα επιτρέπει στον συγγραφέα να επεκτείνει τις δυνατότητες της γλώσσας για να εξυπηρετήσουν κάποιο ειδικό σκοπό. Αυτό επιτρέπει, για παράδειγμα, κόσμους πολλαπλών χρηστών ή νέα γεωμετρικά αντικείμενα να προστεθούν στην VRML.

Προς το τέλος του 1995, αρκετοί ερευνητές με συνεισφορά στην κοινότητα της VRML, καθοδηγούμενοι από τον Mark Pesce, σχημάτισαν την VRML Architecture Group η οποία είχε ως στόχο να συγκεντρώσει τις υπάρχουσες προτάσεις και να προτυποποιήσει την VRML2.0 βάσει των απαιτήσεων που είχαν ως τότε παγιωθεί. Η τελική προτυποποίηση τελείωσε τον Αύγουστο του 1996. Άξια αναφοράς είναι η ημερομηνία του Μαρτίου του 1998, στην οποία πρωτοπαρουσιάστηκε το πρότυπο H-Anim (έκδοση 1.0) για την προσομοίωση κίνησης πάνω σε τρισδιάστατα ανθρωποειδή (humanoids), και το οποίο πρωτοαναπτύχθηκε για την VRML και αργότερα για την τεχνολογία X3D. Με το πρότυπο H-Anim θα ασχοληθούμε σε ξεχωριστό κεφάλαιο καθώς πάνω σε αυτή έχει στηριχθεί η ανάπτυξη του editor H-AnimE.

Στις 9 Αυγούστου 2001, το Web3D Consortium, η πλέον δημοφιλής διαδικτυακή κοινότητα σχετική με την VRML, ανακοινώνει το Ανοικτό Πρότυπο X3D για δικτυακές τρισδιάστατες τεχνολογίες. Η ανακοίνωση από το Web3D Consortium περιγράφει την απαρχή του X3D Προτύπου σαν τον «νέας γενιάς διάδοχο της VRML, ο οποίος θα φέρει πλούσια 3D γραφικά στον Ιστό για μια πληθώρα από εφαρμογές και συσκευές». Το 2004 εγκρίθηκαν οι προδιαγραφές της X3D από τον οργανισμό ISO, ενώ οι κωδικοποιήσεις XML και VRML της γλώσσας αυτής εγκρίθηκαν το 2005. Σήμερα, το X3D έχει πλέον καθιερωθεί σαν ο νόμιμος διάδοχος της VRML, και το σύνολο των εφαρμογών υποστηρίζουν πρωτίστως το πρότυπο X3D και τις προηγούμενες εκδόσεις της VRML για λόγους συμβατότητας.



**Εικόνα 2:** Τρισδιάστατη Αναπαράσταση ενός Γηπέδου με χρήση του X3D

Το X3D σήμερα συνοδεύουν αρκετές βιβλιοθήκες ανοικτού λογισμικού και εμπορικά plug-ins, ενώ πολλοί εμπορικοί editors που υποστηρίζουν επεξεργασία 3D γραφικών ήδη έχουν προσθέσει επιλογές για την εξαγωγή περιεχομένου σε X3D. Επίσης σημαντική είναι και η υποστήριξη από freeware editors. (10)

Τα πιο κοινά στοιχεία διάκρισης μεταξύ X3D και VRML 2.0 είναι τα εξής:

- Πολλαπλές κωδικοποιήσεις δεδομένων (XML, «κλασσική» VRML, δυαδική)
- Νέα γραφικά χαρακτηριστικά (NURBs, Animation, πολλαπλό-texturing, τριγωνικά primitives, 2D σχήματα εσωτερικά σε 3D).
- Νέα χαρακτηριστικά δικτύωσης (LoadSensor, βελτιωμένο Inline)
- Βελτιωμένα APIs, περισσότερα bindings μοντελών (πχ DOM).
- Τμηματικότητα (το πρότυπο είναι διασπασμένο σε profiles και components ώστε να μπορεί να υποστηριχθεί σε πολλαπλά επίπεδα).
- Το MPEG-4 έχει streaming αλληλεπιδραστικά 3D με χρήση VRML.

Ένας άλλος σημαντικός στόχος ήταν η δημιουργία προτυποποίησης που θα επέτρεπε την μέγιστη δυνατή δια-λειτουργικότητα, ότι οι κόσμοι δηλαδή θα εμφανίζονταν γραφικά και θα συμπεριφέρονταν με ταυτόσημο τρόπο σε διαφορετικές υλοποιήσεις. Η VRML 2.0 φτάνει κοντά στο ζητούμενο χωρίς να το καταφέρνει, ενώ για το X3D πιστεύεται ότι ανταποκρίνεται πλήρως, ανάλογα βέβαια με το είδος εφαρμογής που χρησιμοποιείται.

## 2.6 ΒΑΣΙΚΑ ΣΤΟΙΧΕΙΑ ΧΡΗΣΗΣ X3D

### 2.6.1 Ο κύκλος ζωής ενός αρχείου σε X3D

Κάθε X3D αρχείο ξεκινάει σαν μια ιδέα στο μυαλό του δημιουργού, σαν ένα γενικό πλάνο του τι θέλει ο δημιουργός να σχεδιάσει με τα εργαλεία επεξεργασίας X3D που του παρέχονται. Αφού η δημιουργία λάβει υπόσταση σαν X3D αρχείο μέσω του εργαλείου που κρίνεται κατάλληλο, το αρχείο συνήθως διατίθεται σε κάποιον εξυπηρετητή στον Ιστό, με ακριβώς τον ίδιο τρόπο που ο εξυπηρετητής υποστηρίζει ένα κείμενο σε HTML. Η καθορισμένη επέκταση «.wrl» σηματοδοτεί ένα αρχείο «κόσμου» σε X3D.

Για να είναι σωστά προσβάσιμο, η ιστοσελίδα θα πρέπει να έχει τον απαραίτητο εξοπλισμό ώστε να αναγνωρίζει X3D αρχεία. Συνήθως αυτό γίνεται με την χρήση κάποιου X3D browser (αναφέρουμε ενδεικτικά τους: Cortona, FreeWrl, Cosmoplayer κλπ). Το πρώτο πράγμα που κάνει ο X3D browser κατά την εκκίνησή του είναι να διαβάσει το αρχείο. Το «διάβασμα» του αρχείου τυπικά σημαίνει την είσοδο των δεδομένων του αρχείου στον browser, την σωστή αναπαράστασή τους στην μνήμη και ακολούθως την προβολή του εικονικού κόσμου που περιγράφουν στον χρήστη, χρησιμοποιώντας μια μηχανή rendering. (11)

Η μηχανή rendering αναλαμβάνει να σχεδιάσει τα αντικείμενα που περιέχονται στον εικονικό κόσμο στην οθόνη του χρήστη. Καθώς ο χρήστης πλοηγείται στον χώρο, ο φυλλομετρητής συνεχίζει να χρησιμοποιεί την μηχανή για να ανανεώνει την εικόνα που προβάλλεται στον χρήστη με κάθε πιθανή αλλαγή. Συχνά, ο X3D browser ίσως χρειαστεί να φορτώσει άλλους πόρους (πχ, texture maps ή άλλα VRML/X3D αρχεία) από τον Ιστό, ωστόσο αυτό γίνεται γρήγορα και με διαφανή τρόπο.

## 2.6.2 Μηχανές Rendering

Ο όρος «rendering» αφορά στην διαδικασία του σχεδιασμού μιας εικόνας από την αναπαράστασή της στην μνήμη για προβολή στην οθόνη του υπολογιστή (12). Οι μηχανές rendering χρησιμοποιούνται σαν βάση για τους browsers. Η κάθε μηχανή έχει τα προτερήματά της καθώς και τα ελατώματά της, για παράδειγμα είναι εύκολη η δημιουργία μιας μηχανής rendering χρησιμοποιώντας OpenGL, ωστόσο το αποτέλεσμα δεν είναι ικανοποιητικό ως προς την ταχύτητα του browser.

Οι μηχανές rendering έχουν τις δικές τους δομές δεδομένων, και γι' αυτό το λόγο ένα από τα πιο πολύπλοκα και σημαντικά προβλήματα που καλείται να επιλύσει ένας φυλλομετρητής είναι το να μετατρέψει την X3D σε αυτές τις δομές δεδομένων. Παρ' όλο που δυο διαφορετικές μηχανές rendering είναι απίθανο να χρησιμοποιούν τις ίδιες ακριβώς δομές δεδομένων, ωστόσο η λειτουργία περιγράφεται από κάποια κοινά στοιχεία τα οποία θα ήταν σκόπιμο να εξετάσουμε καθώς μαρτυρούν πολλά και για την δομή της X3D.

### 2.6.2.1 Δομές Δεδομένων

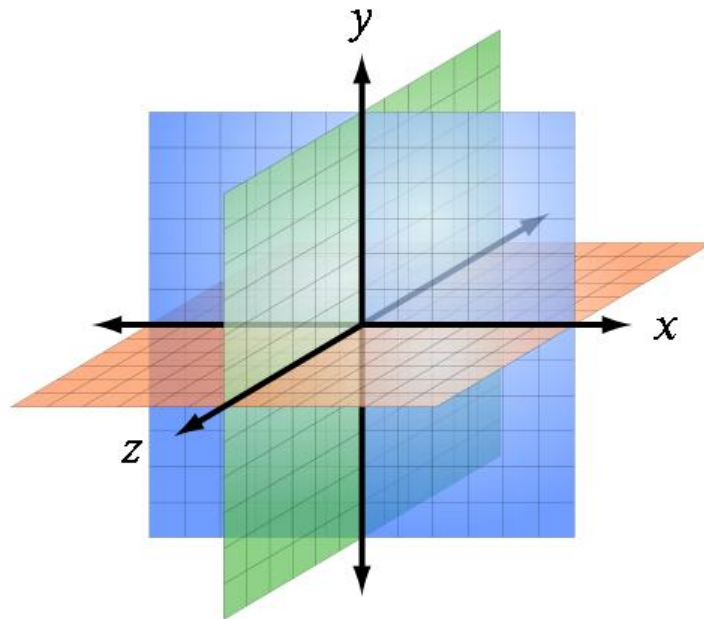
Μια δομή δεδομένων είναι μια συλλογή από πληροφορίες, οργανωμένη κατά τέτοιο τρόπο ώστε να μπορεί να αποθηκευτεί αποδοτικά στην μνήμη του υπολογιστή. Οι περισσότερες μηχανές rendering θεωρούν ότι ο εικονικός κόσμος αποτελείται από έναν αριθμό από αντικείμενα, το καθένα από τα οποία αποτελείται από έναν αριθμό κορυφών (vertices) και ενός συνόλου από πολύγωνα (ή αλλιώς όψεις). Η X3D αντίστοιχα ακολουθεί την παραπάνω παραδοχή, δομώντας τα αντικείμενά της με βάση τον ορισμό κορυφών και πολυγώνων. Η χαρτογράφηση των κορυφών γίνεται ορίζοντας κάποιο σύστημα συντεταγμένων.

### 2.6.2.2 Συστήματα Συντεταγμένων

Ορίζουμε τρεις άξονες για την περιγραφή του εικονικού χώρου: οι X και Y άξονες που ορίζουν τις συντεταγμένες στο επίπεδο είναι κάθετοι μεταξύ τους αλλά και με τον Z άξονα, ο οποίος ορίζει το ύψος στο οποίο βρίσκεται το κέντρο ενός αντικειμένου. (13) Ορίζοντας τους παραπάνω τρεις άξονες, μπορούμε να τοποθετήσουμε μια κορυφή σαν ένα σημείο στον χώρο, αναθέτοντας την τιμή  $(X,Y,Z) = (0,0,0)$  στο κέντρο του εικονικού χώρου. Κατ' αυτόν τον τρόπο



δημιουργείται το σύστημα συντεταγμένων με βάση το οποίο μπορούμε να αναθέσουμε τριάδες τιμών  $X,Y,Z$  για μια γωνία τον χώρο.



**Εικόνα 3:** Παράδειγμα Συστήματος Συντεταγμένων για 3 διαστάσεις στον χώρο

Αναθέτοντας θετικές ή αρνητικές τιμές για τα σημεία του εικονικού χώρου, ανάλογα με την τιμή του προσήμου που ανατίθεται, ορίζεται ένα δεξιόστροφο ή ένα αριστερόστροφο σύστημα συντεταγμένων. Η VRML/X3D χρησιμοποιεί ένα δεξιόστροφο σύστημα συντεταγμένων.

### 2.6.2.3 Κορυφές

Μια κορυφή είναι απλώς ένα μαθηματικό σημείο. Ορίζεται όπως προαναφέρθηκε με βάση το σύστημα συντεταγμένων, αναθέτοντας  $X,Y,Z$  τιμές. Άσχετα από το που ορίζεται το κέντρο του εικονικού κόσμου  $(0,0,0)$ , μπορεί να οριστεί μια ανάθεση τιμών για κάθε κορυφή στον εικονικό κόσμο, ώστε οι μεταξύ τους αποστάσεις να παραμένουν αμετάβλητες. Ένα εικονικό αντικείμενο αποτελείται από έναν αριθμό από τέτοιες κορυφές, οι οποίες συνδέονται μεταξύ τους από κορυφές ή όψεις.

### 2.6.2.4 Ακμές

Δυο κορυφές μπορούν να ενωθούν μεταξύ τους για να δημιουργήσουν μια γραμμή. Εάν έχουμε έναν αριθμό από κορυφές ενωμένες μεταξύ τους με γραμμές, μπορούμε να ορίσουμε ένα “wireframe”, ένα περίγραμμα δηλαδή το οποίο να περιέχει όλες τις κορυφές και τις γραμμές. Για παράδειγμα, το wireframe (14) ενός κύβου περιέχει 8 κορυφές, οι οποίες ενώνονται από 12 γραμμές.



**Εικόνα 4:** Πολύπλοκο περίγραμμα “wireframe” για το σχήμα καρέκλας, με χαρακτηριστικές κορυφές και ακμές.

### 2.6.2.5 Όψεις

Μια όψη (ή πολύγωνο) είναι μια επίπεδη επιφάνεια που ενώνει έναν αριθμό από σημεία. Ένας κύβος αποτελείται από έξι όψεις, κάθε μια από τις οποίες περιβάλλεται από τέσσερις γραμμές, και ενώνει τέσσερις κορυφές. Ας σχεδιάσουμε τις κορυφές ενός κύβου, ορίζοντας τις στο επίπεδο συντεταγμένων:

Θεωρούμε μια αρίθμηση των κορυφών από το 0 έως το 7, και το σύστημα συντεταγμένων  $X, Y, Z$  που αναφέραμε σε προηγούμενη υποενότητα. Μια ανάθεση τιμών για τις κορυφές πάνω στο σύστημα συντεταγμένων θα ήταν η εξής:

**Πίνακας 1:** Αναπαράσταση Κορυφών σε Σύστημα XYZ

#Κορυφής	X	Y	Z
0	-1	-1	1
1	-1	1	1
2	1	1	1
3	1	-1	1
4	1	-1	-1
5	1	1	-1
6	-1	1	-1
7	-1	-1	-1

Τα παραπάνω αποτελούν μια απλή δομή δεδομένων για την αναπαράσταση των κορυφών του κύβου. Το επόμενο βήμα στην αναπαράσταση του κύβου θα ήταν ο

ορισμός μιας δομής δεδομένων που θα καθόριζε ποιες ακμές χρησιμοποιούνται. Η κάθε ακμή θα μπορούσε να περιγραφεί με πολύ απλό τρόπο με απαρίθμηση των κόμβων που χρησιμοποιεί.

**Πίνακας 2: Ορισμός Ακμών με απαρίθμηση κορυφών**

#Ακμής	X	Y	Z
<b>A</b>	0	1	2
<b>B</b>	1	6	5
<b>C</b>	3	2	5
<b>D</b>	4	5	6
<b>E</b>	7	6	1
<b>F</b>	0	3	4

### 2.6.2.6 Αντικείμενα και Transforms

Για να μορφοποιήσουμε ένα αντικείμενο, πέραν του να αλλάξουμε τις κορυφές ή τις όψεις, μπορούμε να επενεργήσουμε πάνω στην μορφή χρησιμοποιώντας απλές λειτουργίες χειρισμού που αποκαλούνται «transformations» ή απλά «transforms» - «μεταμορφώσεις», γιατί μεταμορφώνουν το αντικείμενο με κάποιο τρόπο. Υπάρχουν τρεις βασικοί τύποι μεταμορφώσεων: scaling (κλιμάκωση), rotation (περιστροφή) και translation (μετάφραση). (15)

**Κλιμάκωση:** Η Κλιμάκωση αναφέρεται στην μεγέθυνση ή την σμίκρυνση ενός αντικειμένου, και μπορεί να αναπαρασταθεί από τρεις αριθμούς: τον παράγοντα μεγέθυνσης σε καθέναν από τους άξονες X,Y,Z. Για παράδειγμα, αν θέλουμε να διπλασιάσουμε το μέγεθος ενός κύβου προς όλες τις κατευθύνσεις, θα επιδρούσαμε πάνω του με ένα transform με τιμές 2.0 2.0 2.0 σε κάθε άξονα. Μπορούμε να επεκτείνουμε ένα γεωμετρικό σχήμα μόνο προς έναν ή προς δυο από τους τρεις άξονες παράγοντας το αναμενόμενο αποτέλεσμα.

**Περιστροφή:** η περιστροφή μπορεί να περιγραφεί με μια πληθώρα τρόπων. Για παράδειγμα, μπορούμε να αναθέσουμε γωνίες οι οποίες είναι η βάση για την περιστροφή του αντικειμένου γύρω από κάθε άξονα. Η σειρά με τις οποίες γίνεται η περιστροφή σε κάθε άξονα είναι σημαντική, καθώς το τελικό αποτέλεσμα θα είναι διαφορετικό ανάλογα με την ακολουθία. Μια πιο απλή λειτουργία περιστροφής είναι το να προσδιορίσουμε περιστροφή γύρω από κάποιον άξονα, του οποίου την γωνία ορίζουμε αρχικά (δηλαδή ορίζουμε την περιστροφή σε ένα στάδιο και όχι σε τρια): το αντικείμενο περιστρέφεται γύρω από τον άξονα που ορίζουμε με κάποιο διάνυσμα πχ  $[0\ 1\ 0]$  κατά πχ  $45^\circ$ . (16)

**Μετάφραση:** η μετάφραση αφορά στην μετακίνηση ενός αντικειμένου. Προσδιορίζεται από ένα τρισδιάστατο διάνυσμα το οποίο δίνει την κατεύθυνση πάνω στην οποία το αντικείμενο θα μετακινηθεί. Στην πράξη, έχει το ίδιο οπτικό αποτέλεσμα με το να αλλάζαμε κατευθείαν τις συντεταγμένες X,Y,Z των κορυφών του αντικειμένου. Η σειρά με την οποία κάνουμε κλιμάκωση, περιστροφή και κατ' όπιν μετάφραση είναι σημαντική: το να κλιμακώσουμε και εν συνεχεία να

περιστρέψουμε δεν είναι το ίδιο με το να περιστρέψουμε πρώτα και μετά να κλιμακώσουμε.

### 2.6.2.7 Επίπεδα Συστημάτων Συντεταγμένων

Οι μεταμορφώσεις θα μπορούσαν να παίζουν τον ρόλο μεταφοράς σε κάποιο άλλο σύστημα συντεταγμένων. Στην σύνθεση μιας σκηνής από πολλά αντικείμενα τα οποία προφανώς και θα έχουν κάποια σχέση μεταξύ τους (πχ το στυλό ακουμπάει πάνω στο γραφείο), μπορούμε εύκολα να ορίσουμε άλλο σύστημα συντεταγμένων για το κάθε αντικείμενο όταν αυτό μας διευκολύνει, με κέντρο το σημείο που μας ενδιαφέρει στην εκάστοτε περίπτωση. Στο παράδειγμα που αναφέραμε, μια κορυφή στο σχήμα που αναπαριστά το στυλό έχει κάποια μετατόπιση συντεταγμένων από πχ την πρώτη κορυφή του γραφείου, αλλά έχει εξίσου και κάποια μετατόπιση από κάποια κορυφή στο σχήμα που αναπαριστά τον τοίχο του δωματίου που περιέχει τον χώρο. Αν υποθέσουμε ότι μετακινούμε το γραφείο (προφανώς μαζί με το στυλό) χρησιμοποιώντας κάποιο transform, η μετατόπιση των κορυφών από τον τοίχο αλλάζει, αλλά όχι και η μετατόπιση των κορυφών του στυλού σε σχέση με τις κορυφές του γραφείου. Κατ' αυτόν τον τρόπο μπορούμε να ορίσουμε συστήματα συντεταγμένων με βάση το στυλό, το γραφείο, τον τοίχο κλπ. Η βασική ιδέα είναι ότι η επίδραση ενός transform δεν έχει μόνο δυνατότητες κλιμάκωσης, περιστροφής και μεταφοράς αντικειμένων, αλλά καθορίζουν και μια σχέση μεταξύ δυο συστημάτων συντεταγμένων.

### 2.6.2.8 Ιεραρχίες

Τα συστήματα Εικονικής Πραγματικότητας οργανώνουν τα αντικείμενα σε ιεραρχίες. Μια ιεραρχία μοιάζει αρκετά με ένα δέντρο, ή με έναν οργανωτικό χάρτη, ή ένα διάγραμμα για το σύστημα αρχείων του UNIX. Αν αναλογιστούμε το ανθρώπινο σώμα σε αναπαράσταση X3D, παρατηρούμε ότι αποτελείται από έναν αριθμό από διαφορετικά αντικείμενα τα οποία αναπαριστούν τα μέλη του σώματος (17), το κάθε ένα από τα οποία συνδέεται με κάποιο/-α από τα υπόλοιπα για να σχηματιστεί μια ιεραρχία.

Το κάθε αντικείμενο έχει έναν «γονιό», και επίσης μπορεί να έχει και «παιδιά» πχ η λεκάνη έχει το στήθος και τα δυο άνω πόδια σαν παιδιά, το στήθος με την σειρά του έχει το κεφάλι και τα δυο άνω χέρια σαν παιδιά κλπ. Σε μια ιεραρχία, η κίνηση ενός από τα αντικείμενα που είναι «ψηλότερα» στην ιεραρχία επηρεάζει τα αντικείμενα κατώτερου επιπέδου. Εάν λυγίσουμε για παράδειγμα την μέση, το στήθος κινείται με την σειρά του και η κίνηση αυτή συμπαρασύρει το κεφάλι και τα χέρια. Η κίνηση αντικειμένων χαμηλότερου-επιπέδου δεν έχει καμία επίδραση σε υψηλότερου επιπέδου αντικείμενα: αν γυρίσουμε το κεφάλι μας, το στήθος και η λεκάνη παραμένουν στην θέση τους.

Για να δημιουργήσουμε μια προσομοίωση ανθρώπινου σώματος σε κάποιο σύστημα Εικονικής Πραγματικότητας, θα μπορούμε να δημιουργήσουμε ένα ξεχωριστό αντικείμενο για κάθε μέλος, με άλλα λόγια, ένα ξεχωριστό X3D αρχείο για το κεφάλι, για το δεξί χέρι κλπ. Εν συνεχεία, θα συναρμολογούσαμε το σώμα χρησιμοποιώντας μια σειρά από transformations για να κλιμακώσουμε τα μέλη του σώματος, να τα περιστρέψουμε ώστε να ταιριάζουν, και να τα μεταφέρουμε στην σωστή σχετική θέση μεταξύ τους. Τα παραπάνω transforms μπορούν να συνδυαστούν ώστε το κάθε αντικείμενο να έχει ένα αθροιστικό transform που το συσχετίζει με τον γονιό του.

Συγκεκριμένη λύση για το παραπάνω πρόβλημα κλήθηκε να δώσει το πρότυπο H-Anim, ορίζοντας προ-καθορισμένη (αλλά δυναμικά ορισμένη) ιεραρχία για τα μέλη του ανθρώπινου σώματος. Η επεξεργασία ενός avatar που ακολουθεί το πρότυπο H-Anim ώστε να διατηρούνται οι συμβάσεις που αναπτύχθηκαν στις προηγούμενες παραγράφους είναι το αντικείμενο μελέτης και επεξεργασίας του editor που αναπτύχθηκε για την εκπόνηση της διπλωματικής εργασίας.

### 2.6.2.9 Διασωλήνωση των Γραφικών

Κάθε φορά που η μηχανή του rendering αναλαμβάνει να επανασχεδιάσει την σκηνή χρησιμοποιεί ένας μηχανισμό «διασωλήνωσης των γραφικών». (18) Στην πραγματικότητα, η σχεδίαση X3D κόσμων που να τρέχουν σε αποδοτική ταχύτητα με δεδομένο υλικό του υπολογιστή δεν είναι απλή διαδικασία, ούτε αποδεσμεύεται από τον μηχανισμό με τον οποίο η μηχανή rendering σχεδιάζει τα γραφικά στην οθόνη του υπολογιστή.

Το σύστημα αρχικά θα πρέπει να διατρέξει όλο το «δέντρο» με βάση το οποίο έχει δομηθεί ο κόσμος και να υπολογίσει τις θέσεις, την κλιμάκωση και τον προσανατολισμό κάθε αντικειμένου. Όταν μετακινήσουμε το στήθος ενός avatar σχετικά με την λεκάνη, τα νέα δεδομένα θα πρέπει να σημάνουν αλλαγές για όλα τα εξαρτώμενα μέλη της ιεραρχίας. Αυτό υπολογίζεται σαν μια σειρά από συναθροιζόμενα transforms, καταλήγοντας σε κάθε κορυφή κάθε αντικειμένου μεταφραζόμενες σε συντεταγμένες κόσμου.

Το επόμενο βήμα είναι ο υπολογισμός φωτισμού για κάθε κορυφή. Αυτή ωστόσο είναι μια πολύπλοκη διαδικασία η οποία χρειάζεται λεπτομέρεια για να περιγράψει. Η κάθε κορυφή θα έχει μια κάποια φωτεινότητα σχετιζόμενη με κάποια φωτεινή πηγή στην οθόνη, και το τελικό αποτέλεσμα φωτεινότητας για την κορυφή θα προκύψει συνυπολογίζοντας το πόσο επηρεάζει η κάθε πηγή.



**Εικόνα 5:** Φωτισμός σε μια 3D σκηνή

Αφού οι συντεταγμένες των κορυφών έχουν μεταφραστεί σε συντεταγμένες χώρου κόσμου και οι τιμές φωτεινότητάς τους έχουν υπολογιστεί, υπόκεινται σε ακόμη μια διαδικασία μετάφρασης συντεταγμένων: αυτή την φορά κέντρο αναφοράς είναι η εικόνα που θα πρέπει να δει ο χρήστης (ή η κάμερα). Το «ανθρώπινο μάτι» μπορεί να δει μόνο τα αντικείμενα που περιέχονται μέσα σε μια συγκεκριμένη γωνία, τόσο οριζόντια όσο και κάθετα. Τα αντικείμενα έξω από αυτό το οπτικό πεδίο δεν είναι

ορατά, και δεν χρειάζεται καν η μηχανή να συνεχίζει να τα επεξεργάζεται. Το να διευκρινιστεί ποια αντικείμενα μπορούμε να αγνοήσουμε είναι γνωστό σαν «**επιλογή αντικειμένων**» (αγγλικός όρος: object culling).

Στην συνέχεια η μηχανή αναλαμβάνει την διαδικασία του «**ψαλιδίσματος**»: κάθε όψη ξεχωριστά ελέγχεται για το αν είναι ολόκληρη μέσα στο οπτικό πεδίο, ολόκληρη έξω ή στο ενδιάμεσο. Οι εξ' ολοκλήρου εκτός αγνοούνται πλήρως, οι όψεις που βρίσκονται εξ' ολοκλήρου μέσα δεν ψαλιδίζονται καθόλου, ενώ αυτές που βρίσκονται στο ενδιάμεσο ψαλιδίζονται ώστε μόνο τα τμήματα που βρίσκονται μόνο μέσα στο οπτικό πεδίο να υποστούν επεξεργασία.

Το επόμενο στάδιο είναι η **αφαίρεση πίσω όψης**: υποθέτοντας το παράδειγμα ενός κύβου, θα πρέπει να είναι ξεκάθαρο ότι από κάθε οπτική γωνία μπορούμε να δούμε μόνο (κατά μέγιστο) τρεις από τις έξι όψεις του κύβου. Αυτό είναι αληθές για όλα τα αντικείμενα που θα μπορούσαμε να σχεδιάσουμε στον εικονικό κόσμο: σχεδόν οι μισές όψεις μέσα στην σκηνή θα είναι απομακρυσμένες από την οπτική μας γωνία σε κάθε περίπτωση. Η **αφαίρεση πίσω όψης** χρησιμοποιείται για να αφαιρέσει αυτές τις όψεις, ώστε να μην σπαταλάται χρόνος και επεξεργαστική ισχύς για τον σχεδιασμό τους.

Για να προσδώσουμε βάθος στον εικονικό μας κόσμο, χρησιμοποιούμε την **τεχνική την προοπτικής** (δανεισμός από τον χώρο των καλών τεχνών). Από τεχνικής απόψεως, η προοπτική υπολογίζεται διαιρώντας τις τιμές των X,Y με την τιμή Z για κάθε κορυφή. Οι προκύπτουσες συντεταγμένες X και Y για κάθε κορυφή εν συνεχεία κλιμακώνονται και ολισθαίνουν ώστε να εμφανίζονται στην σωστή μεριά στην προβολή της οθόνης.

Ένα σημαντικό βήμα στην διασωλήνωση είναι η **εξαφάνιση των κρυμμένων επιφανειών**: εξασφαλίζει ότι τα αντικείμενα θα εμφανίζονται στην σωστή σειρά βάθους. Είναι απαραίτητο να γίνει κάποιου είδους ταξινόμηση, ώστε οι επιφάνειες που αποκρύπτονται πίσω από κοντινότερα αντικείμενα να μην σχεδιάζονται πάνω από αυτά. Αυτό γίνεται με δυο τρόπους: είτε σχεδιάζονται οι όψεις στην σωστή σειρά, πράγμα δύσκολο, ή σχεδιάζονται σε αυθαίρετη σειρά και χρησιμοποιείται ένας Z-buffer.

Το τελικό στάδιο στην διασωλήνωση των γραφικών είναι το στάδιο του **rasterization**: σε αυτό το στάδιο σχεδιάζονται οι όψεις των αντικειμένων στην οθόνη. Είναι μακράν το πιο αργό στάδιο σε ένα σύστημα που βασίζεται σε λογισμικό, γι' αυτό και συχνά χρησιμοποιείται κάποιος 3D accelerator. Σε αυτό το στάδιο γίνεται και η **σκίαση**.

Υπάρχουν τρεις τρόποι για να εμφανίσουμε όψεις στην οθόνη: ο πρώτος είναι το wireframe, στο οποίο εμφανίζουμε μόνο τα περιγράμματα των όψεων. Ο δεύτερος τρόπος είναι η επίπεδη σκίαση, στην οποία οι όψεις σχεδιάζονται, αλλά κάθε όψη έχει ένα μονό χρώμα του οποίου η φωτεινότητα είναι ανάλογη στην ένταση του φωτός που πέφτει πάνω του. Ο τρίτος τύπος σκίασης είναι ο Gouraud, ονομασμένος κατά τον δημιουργό, ο οποίος περιλαμβάνει την παρεμβολή (interpolation) της έντασης φωτός κατά μήκος κάθε επιφάνειας, ώστε να εξομαλυνθούν οι αιχμές. Σε ορισμένες περιπτώσεις αποκαλείται και «smooth shading», «ομαλή σκίαση».

Το Dithering είναι μια επιπλέον τεχνική που χρησιμοποιείται για την αλλαγή της πυκνότητας των pixels ώστε να παράγει ένα πιο ομαλό αποτέλεσμα. Επιπλέον χρησιμοποιούνται τεχνικές Texture Mapping στις οποίες μια υφή – για παράδειγμα μια εικόνα που αναπαριστά έναν τοίχο αποτελούμενο από τούβλα – ανατίθεται κατά

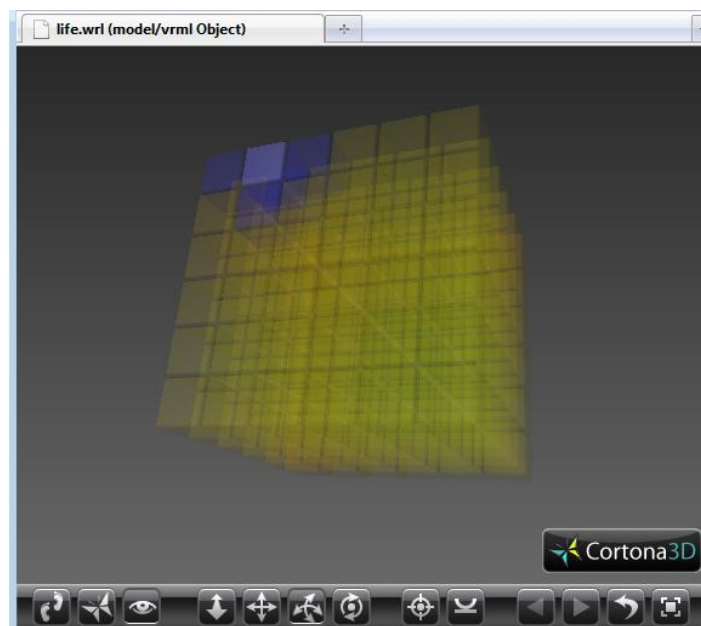
μήκος κάποιας επιφάνειας ώστε να προσδώσει μεγαλύτερο ρεαλισμό στην εικόνα. Το οπτικό αποτέλεσμα της αποτελεσματικής χρήσης texture mapping είναι συχνά πολύ κοντά στο ιδεατό.

### 2.6.3 X3D Φυλλομετρητές

Με τον ίδιο τρόπο η HTML απαιτεί την χρήση ενός φυλλομετρητή για την προβολή της πληροφορίας στον χρήστη, τα περιβάλλοντα X3D απαιτούν έναν φυλλομετρητή για X3D (19) για να μετατρέψουν τα συστήματα συντεταγμένων και τα περιεχόμενα αντικείμενά τους σε ένα τρισδιάστατο περιβάλλον στο οποίο ο χρήστης να μπορεί να περιηγηθεί.

#### 2.6.3.1 Πως δουλεύει ένας φυλλομετρητής για X3D

Η κύρια λειτουργία ενός φυλλομετρητή για X3D είναι να διαβάσει το αρχείο χαρακτήρα-προς-χαρακτήρα και να επιχειρήσει να τον μεταφράσει σε μια γραφική εικόνα. Κάθε φορά που ο φυλλομετρητής συναντάει κάποιον μη αναμενόμενο χαρακτήρα ή κάποια έκφραση που δεν προβλέπεται από την σύνταξη της X3D, δημιουργεί ένα μήνυμα λάθους και σταματάει την λειτουργία του. Ανάλογα τις δυνατότητες του φυλλομετρητή, αναμένεται να είναι σε θέση να κατανοήσει και αναπαραστάσεις βασισμένες σε παλαιότερες εκδόσεις της γλώσσας.



**Εικόνα 6:** Ο Φυλλομετρητής Cortona3D εφαρμοζόμενος ως plug-in στον φυλλομετρητή Ιστού Mozilla Firefox

Στην περίπτωση που το X3D υλικό είναι δημοσιευμένο στον Ιστό, το πιθανότερο είναι πως κάποιος καθιερωμένος φυλλομετρητής Ιστού (πχ Internet Explorer, Mozilla Firefox) θα είναι σε θέση να προβάλλει την αναπαράσταση χρησιμοποιώντας κάποιον φυλλομετρητή για X3D σε μορφή plug-in, είτε τρέχοντας απευθείας τον φυλλομετρητή είτε ενσωματώνοντάς τον, με τον ίδιο τρόπο που για να προβάλλει το περιεχόμενο ενός .pdf αρχείου θα ενσωμάτωνε τον Acrobat Reader.



Σε πρώτη φάση, αφού ο φυλλομετρητής X3D διαβάσει τους κόμβους που καθορίζονται στο αρχείο και τα πεδία, θα επιλέξει μια αρχική θέση για την οπτική γωνία (κάμερα). Επίσης θα ελέγξει για το αν το αρχείο έχει συνδέσμους που χρησιμοποιούν τους WWWAnchor και WWWInline κόμβους. Εάν βρει τον WWWInline κόμβο, θα ανατρέξει για επιπλέον κόμβους σε εξωτερικά αρχεία από τον ανάλογο Web Server. Αφού έχουν γίνει όλοι οι απαραίτητοι υπολογισμοί, ο X3D browser προβάλλει με χρήση της μηχανής rendering την πληροφορία που έχει εντοπίσει στην οθόνη του υπολογιστή. Στην συχνή περίπτωση που το X3D περιβάλλον είναι καταναμημένο, δηλαδή η πληροφορία που το συνθέτει είναι διαμοιρασμένη στον Ιστό, η παραπάνω διαδικασία γίνεται σε στάδια. Στο πρώτο στάδιο, ο φυλλομετρητής φορτώνει την βασική περιγραφή της σκηνής. Κατ' όπιν φορτώνει οποιοσδήποτε σκηνές μπορεί να περιέχονται μέσα στην σκηνή (ή εμφωλευμένες σκηνές), πράγμα το οποίο μπορεί να προκαλέσει καθυστέρηση ανάλογα στην κίνηση στο διαδίκτυο, την ποιότητα σύνδεσης, την επεξεργαστική ισχύ του υπολογιστή και την ταχύτητα σύνδεσης.

Καθώς ο χρήστης πλοηγείται μέσα στο περιβάλλον, ο φυλλομετρητής θα ελέγχει διαρκώς να δει αν χρειάζεται να φορτώσει επιπλέον νέο υλικό καθώς η οπτική γωνία αλλάζει. Επίσης θα ελέγχει για τυχόν ενεργοποίηση κόμβων WWWAnchor, ώστε σε αυτή την περίπτωση να ανατρέξει στο προκύπτον URL από το WWWAnchor και να φορτώσει το X3D αρχείο που θα του επιστρέψει ο web server.

### 2.6.3.2 Χειρισμός μιας σκηνής X3D

Όλοι οι X3D browsers χρησιμοποιούν το κεντρικό σημείο του μοντέλου σαν το κέντρο αναφοράς για την προβολή του αντικείμενου. Αυτό σημαίνει ότι εάν θέλουμε να εξετάσουμε ένα αντικείμενο ή κάποια άκρη ενός μοντέλου, δεν περιστρέφουμε στην πραγματικότητα το ίδιο το αντικείμενο, αλλά συνολικά το περιβάλλον. Ένας άλλος τρόπος οπτικοποίησης τέτοιου είδους χειρισμού είναι το να φανταστούμε ολόκληρο το X3D μοντέλο να περιέχεται μέσα σε μια σφαίρα. Όταν μετακινούμε ένα αντικείμενο, ολόκληρο το μοντέλο μετακινείται χρησιμοποιώντας το κέντρο της σφαίρας σαν σημείο αναφοράς. Το πως χειριζόμαστε τα αντικείμενα χρησιμοποιώντας ένα φυλλομετρητή εξαρτάται από την θέση τους στο περιβάλλον.

## 2.7 ΤΟ ΠΡΟΤΥΠΟ X3D ΚΑΙ ΕΦΑΡΜΟΓΕΣ

### 2.7.1 Βασική Σύνταξη X3D

Ένα VRML αρχείο είναι ένα απλό αρχείο μορφής κειμένου το οποίο χρησιμοποιεί ASCII, ενώ ένα X3D αρχείο επίσης χρησιμοποιεί ASCII κείμενο με την διαφορά ότι μπορεί να χρησιμοποιεί την σύνταξη της XML (20). Κατά συνέπεια, μπορούμε να δημιουργήσουμε VRML/X3D περιεχόμενο χρησιμοποιώντας έναν απλό κειμενογράφο όπως το notepad των Windows. Ωστόσο, επειδή είναι δύσκολη η δημιουργία αισθητικά ευχάριστου αποτελέσματος σε X3D με απλή σύνταξη της κωδικοποίησης, χρησιμοποιούνται editors που αναλαμβάνουν να απομακρύνουν τον χρήστη από την απευθείας συγγραφή κώδικα και να τον μεταφέρουν στο επίπεδο οπτικής επεξεργασίας.

Παραθέτουμε τον κώδικα ενός αρχείου VRML για την έκδοση 2.0, όπως θα εμφανιζόταν αν ανοίγαμε το αρχείο με κάποιον απλό κειμενογράφο:



```

#VRML V2.0 utf8
# A Cylinder
Shape {
  appearance Appearance {
    material Material { }
  }
  geometry Cylinder {
    height 2.0
    radius 1.5
  }
}

```



**Εικόνα 7:** Προβάλλοντας ένα απλό σχήμα κυλίνδρου σε φυλλομετρητή

Στην εικόνα φαίνεται ο κύλινδρος που κωδικοποιεί ο παραπάνω κώδικας, όπως θα προβαλλόταν με χρήση του φυλλομετρητή για VRML/X3D Cortona3D.

Σχολιάζοντας τον κώδικα, αρχικά αναφέρουμε πως ότι ακολουθεί τον χαρακτήρα # στην ίδια γραμμή είναι σχόλιο, το οποίο συνεχίζει μέχρι το τέλος της ίδιας γραμμής. Με εξαίρεση την χρήση του χαρακτήρα #, το πρότυπο είναι πλήρους ελευθέρως-μορφής: οπουδήποτε μέσα στο αρχείο μπορούν να υπάρξουν πολλαπλά κενά, χαρακτήρες tabs ή και κενές γραμμές. Η πρώτη γραμμή η οποία παρουσιάζεται ως σχόλιο παραθέτει πληροφορίες στον αναγνώστη σχετικά με την έκδοση της VRML στην οποία αντιστοιχεί ο κώδικας. Όταν ένας φυλλομετρητής διαβάσει το αρχείο ωστόσο, παρ' όλο που η πρώτη γραμμή εμφανίζεται ως «σχόλιο», θα την χρησιμοποιήσει ως πληροφορία σχετικά με την έκδοση της VRML στην οποία αντιστοιχεί ο κώδικας. Το ίδιο ωστόσο δεν θα γίνει και για την δεύτερη γραμμή, η οποία είναι απλώς ένα σχόλιο χωρίς πρακτική αξία.

### 2.7.1.1 Κόμβοι και πεδία

Οποιοσδήποτε έχει εμπειρία από προγραμματισμό μπορεί να αναγνωρίσει τα άγκιστρα '{' και '}' τα οποία χρησιμοποιούνται για να προσδιορίσουν μπλοκ πληροφορίας ή συσχετιζόμενες πληροφορίες. Τα αρχεία VRML/X3D αποτελούνται από «κόμβους», οι οποίοι ακολουθούν την εξής σύνταξη:

```
Τύπος_Κόμβου { <περιεχόμενο κόμβου> }
```

Οι τύποι κόμβου που επιτρέπονται από την σύνταξη της VRML/X3D είναι καθορισμένοι από το πρότυπο (και σε μικροτερο βαθμο απο τον αριθμό έκδοσης). Στο παράδειγμά μας χρησιμοποιούμε τους κόμβους: Shape, appearance, material, geometry, ενώ το περιεχόμενο μέσα στα { } καθορίζει τα **πεδία** του εκάστοτε κόμβου. Κάθε πεδίο έχει όνομα και τιμή. Σε αυτή την περίπτωση, το πεδίο height της γεωμετρίας του κυλίνδρου έχει την τιμή 2.0, το οποίο καθορίζει για τον κύλινδρο ότι θα έχει ύψος 2.0. Οι τιμές των πεδίων μπορεί να είναι απλοί αριθμοί, ομάδες αριθμών (πχ 1 0 0), strings, λέξεις κλειδιά, εικόνες, Boolean τιμές, κλπ. Αρκετά πεδία έχουν πολλαπλές τιμές, οπότε και διαχωρίζονται με κόμμα και περιβάλλονται από τους χαρακτήρες '[' και ']'. Για παράδειγμα θα μπορούσαμε να γράψουμε τον κόμβο Material { diffuseColor [ 1 0 1, 0.5 0.75 0, 0 1 0, 0.3 0.3 0.2 ] }.

Κάθε πεδίο έχει μια προκαθορισμένη τιμή. Στην περίπτωση στην οποία δεν καθορίσουμε εμείς κάποια τιμή για κάποιο πεδίο ενός κόμβου, παραλείποντας το πεδίο, αυτό συμπληρώνεται αυτόματα με την προκαθορισμένη τιμή. Για παράδειγμα, η προκαθορισμένη τιμή ακτίνας για μια σφαίρα είναι το 1 μέτρο.

### 2.7.1.2 Ονομασία Κόμβων

Σε οποιονδήποτε κόμβο μπορεί να ανατεθεί ένα όνομα, ώστε να γίνει αναφορά σε αυτόν αργότερα. Αυτό γίνεται με χρήση του προθέματος DEF, πχ:

```
DEF Pants_Color Material { }
```

Στο παραπάνω παράδειγμα ορίσαμε έναν κόμβο τύπου Material ο οποίος έχει την ονομασία Pants\_Color. Αν υποθέσουμε ότι δημιουργούσαμε την αναπαράσταση ενός ανθρωποειδούς σε VRML/X3D, και δημιουργούσαμε 5-6 διαφορετικούς κόμβους (πχ κυλίνδρους) για την αναπαράσταση των ποδιών του ανθρωποειδούς, θα μπορούσαμε να αναθέσουμε και στους 5 κόμβους το ίδιο Material χρησιμοποιώντας:

```
material USE Pants_Color
```

κάθε φορά που θα καλούμασταν να ορίσουμε το πεδίο material των μελών. Το παραπάνω θα είχε σαν αποτέλεσμα μια αλλαγή στα πεδία του Pants\_Color να αλλάξει την εμφάνιση όλων των κόμβων που τον χρησιμοποιούν ως material.

## 2.7.2 Αντικείμενα, Ιεραρχίες και Κόμβοι Διαχωρισμού

Όπως έχει ήδη προαναφερθεί, ένας εικονικός κόσμος μπορεί να θεωρηθεί ως μια ιεραρχία αντικειμένων. Οι πληροφορίες που σχετίζονται με ένα αντικείμενο μπορούν να αποθηκευτούν με την χρήση ενός Κόμβου Διαχωρισμού (Separator) (21). Ένας

κόμβος διαχωρισμού είναι σαν ένα κέλυφος που περιέχει ένα αντικείμενο. Δεν περιέχουν όλοι οι κόμβοι διαχωρισμού απαραίτητα γεωμετρικές πληροφορίες (κορυφές και όψεις), καθώς αρκετοί χρησιμοποιούνται απλώς για να περιέχουν άλλα αντικείμενα σαν μονάδα. Ο κόμβος διαχωρισμού είναι ένας τύπος κόμβους που χρησιμοποιήθηκε στην VRML1.0 και εγκαταλείφθηκε έπειτα από την VRML2.0 και μετά καθώς οι λειτουργίες του ενσωματώθηκαν και επεκτάθηκαν σε άλλους τύπους κόμβων. Αναφέρουμε τον κόμβο διαχωρισμού για ιστορικούς λόγους, αλλά και για να δώσουμε στον αναγνώστη την βασική ιδέα πίσω από την δημιουργία ιεραρχιών.

Ένας κόμβος διαχωρισμού έχει κόμβους «παιδιά» στο εσωτερικό του, οι οποίοι περιγράφουν τα αντικείμενα που περιέχονται σε αυτόν. Αυτά τα παιδιά με την σειρά τους περιέχουν άλλα παιδιά, κλπ, σχηματίζοντας έτσι μια Ιεραρχία. Για παράδειγμα, αν θέλαμε να αναπαραστήσουμε το ανθρώπινο σώμα χρησιμοποιώντας μια ιεραρχική δομή, θα είχαμε το εξής:

```
DEF Body Separator {
  DEF Pelvis Separator {
    DEF Chest Separator {
      DEF LeftUpperArm Separator {
        DEF LowerArm Separator {}
      }
      DEF RightUpperArm Separator {
        DEF LowerArm Separator {}
      }
      DEF Head Separator {
        DEF Skull Separator {}
        DEF LeftEye Separator {}
        DEF RightEye Separator {}
        DEF Nose Separator {}
      }
    }
  }
  DEF LeftThigh Separator {
    DEF Calf Separator {}
  }
  DEF RightThigh Separator {
    DEF Calf Separator {}
  }
}
```

Να σημειωθεί ότι το παραπάνω δεν χρησιμοποιεί τις προδιαγραφές του προτύπου H-Anim, το οποίο και χρησιμοποιεί ο editor της Διπλωματικής Εργασίας. Παρέχει ωστόσο μια βασική ιδέα για το πως μπορούμε να δομήσουμε μια ιεραρχία χρησιμοποιώντας κόμβους διαχωρισμού. Την πληροφορία που θα περιέχονταν μέσα σε κάθε κόμβο την έχουμε παραλείψει στο συγκεκριμένο παράδειγμα.

Ένας κόμβος διαχωρισμού μπορεί επίσης να περιέχει πληροφορία σχετικά με το σχήμα του αντικειμένου που περιγράφει, καθώς και πληροφορίες σχετικά με τις ιδιότητες του αντικειμένου (όπως ένας κόμβος Material). Επιπρόσθετα, ένας κόμβος διαχωρισμού μπορεί να δώσει την τοποθεσία, τον προσανατολισμό και το μέγεθος των αντικειμένων σε σχέση με τους κόμβους γονέα. Αυτό μπορεί να γίνει

χρησιμοποιώντας `transforms`. Στην πράξη, ένας κόμβος Διαχωρισμού θα έμοιαζε ως εξής:

```
Separator {
    transforms
    surface properties
    shapes
    children
}
```

Εάν οι `surface properties` (ιδιότητες επιφανείας) δεν προσδιοριστούν, «υιοθετούνται» από το αντικείμενο γονέα. Για παράδειγμα, αν δεν έχουμε ορίσει κάποιον κόμβο `Material` στον τρέχονα διαχωριστή κόμβο, χρησιμοποιείται ο κόμβος `Material` του γονέα. Αν ούτε ο γονέας ούτε κάποιος άλλος πρόγονος διαθέτει κόμβο `Material`, τότε χρησιμοποιούνται οι `default` τιμές.

Μπορούμε να χρησιμοποιήσουμε πολλαπλούς κόμβους τύπου `Shape` (σχημάτων) μέσα σε μόλις έναν Διαχωριστή, εάν είναι απαραίτητο. Για παράδειγμα, μπορούμε να έχουμε το εξής:

```
Separator {
    Material {
        diffuseColor 0 1 0
    }
    Transform {
        rotation 1 0 0 0 5
    }
    IndexedFaceSet {
        coordIndex [ ... ]
    }
    IndexedLineSet {
        coordIndex [ ... ]
    }
}
```

Στο παραπάνω παράδειγμα έχουμε δυο κόμβους σχημάτων, τους `IndexedLineSet` και `IndexedFaceSet` οι οποίοι μοιράζονται ένα `material` και ένα `transformation`. Εάν δεν προσδιορίζεται κόμβος σχήματος, τότε το αντικείμενο δεν έχει εμφάνιση. Στην πράξη υπάρχει, αλλά είναι ένας αόρατος στον χρήστη κόμβος που απλά περιέχει πληροφορίες, είτε για τα παιδιά του, είτε για μελλοντική χρήση αν με κάποια λειτουργία του προστεθεί γεωμετρία.

Η παραπάνω στρατηγική προσάρτησης πολλαπλών κόμβων γεωμετρίας σε έναν κόμβο διαχωριστή χρησιμοποιήθηκε κατά την ανάπτυξη του `editor` σε πολλαπλά σημεία (πχ στην περίπτωση που προσαρτούμε ένα καπέλο στο κεφάλι ενός `avatar`, ώστε να διατηρηθούν οι δυο γεωμετρίες αλλά να λειτουργούν σαν μια).

### 2.7.3 *Κόμβοι Transforms*

Οι λειτουργίες κλιμάκωσης, περιστροφής και μετάφρασης που περιγράψαμε σε προηγούμενη ενότητα ορίζονται με παρόμοιο τρόπο για την `X3D`. Η μετάφραση και

η κλιμάκωση αναπαρίστανται από διανύσματα, και η περιστροφή σε μορφή άξονα-γωνίας με τέσσερις αριθμούς (τρεις για τον άξονα περιστροφής, και έναν για τη γωνία). Η γωνία περιστροφής δεν δίνεται σε μοίρες, αλλά σε rads/radians (η μετατροπή από μοίρες σε rads γίνεται με πολλαπλασιασμό με το  $\pi/180.0$  όπου  $\pi$  είναι η γνωστή σταθερά του Αρχιμήδη και ισούται με 3.14159...). Οι μεταφράσεις μετριοούνται σε μέτρα.

Υπάρχουν πολλοί τρόποι για να προσδιορίσουμε transforms σε X3D. Ο πιο απλός είναι με την χρήση ενός Transform κόμβου, ο οποίος συνδυάζει κλιμάκωση, περιστροφή και μετάφραση, πχ:

```
#VRML V2.0 utf8

Transform {
  scale 1.0 1.0 1.0
  scaleOrientation 0.0 0.0 1.0 0.0
  center 1.0 1.0 1.0
  rotation 0.0 0.0 1.0 0.0
  translation 1.0 1.0 1.0
  children [
    Shape { ... }
    geometry Box {}
  ]
}
```

Το πεδίο “center” περιγράφει ένα σημείο στον τρισδιάστατο χώρο με βάση το οποίο το αντικείμενο κλιμακώνεται και περιστρέφεται. Είναι σημαντικό να θυμόμαστε ότι και αυτό είναι ένα σημείο στο σύστημα συντεταγμένων του αντικειμένου. Για παράδειγμα, αν έχουμε μια σφαίρα με ακτίνα 2 μονάδων, και αναθέσουμε ως κέντρο το -2, 0, 0 τότε θα περιστραφεί γύρω από από ένα σημείο στην επιφάνεια της σφαίρας.

Το πεδίο “scaleOrientation” περιγράφει τον τρόπο με τον οποίο θα περιστραφεί το σύστημα συντεταγμένων πριν γίνει κλιμάκωση. Αυτό επιτρέπει στο αντικείμενο να κλιμακωθεί με ασυνήθιστους τρόπους. Τα πεδία μπορούν να απαριθμηθούν με οποιαδήποτε σειρά, αλλά η σειρά με την οποία γίνονται οι επεμβάσεις πάνω στην γεωμετρία είναι πάντα η ίδια: κλιμάκωση, ύστερα περιστροφή, και στην συνέχεια μετάφραση. Με άλλα λόγια, το αντικείμενο πρωτίστως μετακινείται ώστε η αρχή του να είναι το σημείο που περιγράφεται ως κέντρο του, εν συνεχεία περιστρέφεται ανάλογα με το πεδίο scaleOrientation, κλιμακώνεται, και περιστρέφεται πίσω. Κατ’ όπιν περιστρέφεται με βάση την τιμή περιστροφής του, μετακινείται πίσω ώστε η αρχή του να είναι το αρχικό σημείο κέντρου του, και μετακινείται πάλι πίσω σύμφωνα με την μετάφραση.

## 2.7.4 Βασικά Σχήματα

Η X3D επιτρέπει μια πληθώρα από διαφορετικά σχήματα: σφαίρες, κώνους, κυλίνδρους, και κύβους. Στην συνέχεια εξετάζουμε τα σχήματα αυτά όπως αναπαρίστανται στην X3D:

### 2.7.4.1 Σφαίρα

Ο κόμβος της Σφαίρας είναι πολύ απλός, καθώς περιέχει μόλις ένα πεδίο, αποκαλούμενο «ακτίνα», το οποίο προσδίδει την επιθυμητή ακτίνα στην σφαίρα μετρημένη σε μέτρα, πχ:

```
Sphere {
    radius 1
}
```

Η κατ' εξοχήν τιμή της ακτίνας είναι το ένα μέτρο.

### 2.7.4.2 Κώνος

Ένας κώνος αποτελείται από 2 τμήματα: τις πλευρές και την κάτω όψη. Ο προσδιορισμός τους ωστόσο είναι προαιρετικός. Εξ' ορισμού ένας κώνος διαθέτει και τα δυο, ενώ το πεδίο parts μπορεί να προσδιορίσει είτε το ένα ή το άλλο αναθέτοντάς του τιμές SIDES ή BOTTOM αντί για ALL που είναι η εξ' ορισμού. Ένας κώνος έχει δυο ακόμα πεδία: height και bottomRadius, τα οποία προσδιορίζουν αντίστοιχα το ύψος των πλευρών του κώνου και την ακτίνα της βάσης. Οι εξ' ορισμού τιμές είναι 1 και 2 αντίστοιχα.

```
Cone {
    bottomRadius radius
    height height
    parts parts
}
```

### 2.7.4.3 Κύλινδρος

Ένας κύλινδρος είναι παρόμοιος με έναν κώνο, αλλά έχει τρία σκέλη: τις πλευρές, την κάτω όψη και την πάνω όψη. Το πεδίο parts μπορεί να έχει οποιονδήποτε συνδυασμό εκ των τριών (παράδειγμα σύνταξης: (SIDES|BOTTOM)). Η σύνταξη για έναν κύλινδρο είναι η εξής:

```
Cylinder {
    parts parts
    bottomRadius radius
    height height
}
```

Οι κόμβοι τύπου Sphere, Cone και Cylinder δεν είναι παρά συντομεύσεις ωστόσο για γεωμετρικά σχήματα. Όταν ο φυλλομετρητής διαβάζει το X3D αρχείο, μετατρέπει τις παραμέτρους που έχουν οριστεί για αυτούς σε κορυφές και όψεις με μια διαδικασία που ονομάζεται «tessellation».

### 2.7.4.4 Κύβος

Ο κύβος είναι ένα από τα πιο απλά σχήματα. Έχει ύψος, μήκος και βάθος, πχ:

```
Cube {
    width width
    height height
}
```

```

    depth depth
}

```

Και τα τρία πεδία έχουν μια εξ' ορισμού τιμή 2.0. Καθώς η αρχή του κύβου είναι το κέντρο του, οι πλευρές του κύβου είναι όλες ένα μέτρο από την αρχή του.

### 2.7.4.5 AsciiText ή Text

Επιπλέον, υπάρχει ένας κόμβος που ονομάζεται AsciiText ο οποίος επιτρέπει στον χρήστη να δημιουργήσει επίπεδο κείμενο (πχ για την δημιουργία επιγραφών). Η σύνταξη είναι εξαιρετικά απλή:

```

AsciiText {
    string "...
    spacing spacing
    justification justification
    width width
}

```

Το περιεχόμενο του String είναι το επίπεδο κείμενο που θα προβληθεί, και περιέχεται μέσα στο "...". Μπορούμε να έχουμε πολλαπλά strings, διαχωρισμένα από κόμματα και μέσα σε [...]. Το πεδίο justification μπορεί να είναι LEFT, RIGHT ή CENTER. Στις τελευταίες εκδόσεις της VRML καθώς και στο X3D, ο κόμβος AsciiText εμφανίζεται και ως απλά Text.

Παράδειγμα:

```

Text {
    string ["To be or not to be",
           "That is the question"]
    fontStyle FontStyle {
        family "SANS"
        style "BOLD"
        size 0.5
        justify "MIDDLE"
    }
}

```

Στο παραπάνω παράδειγμα έχουμε ορίσει επιπλέον fontStyle για το κείμενό μας, το οποίο καθορίζει επιπλέον θέματα γραμματοσειράς.

### 2.7.4.6 IndexedFaceSet

Οι κόμβοι που εξετάσαμε έως τώρα (Sphere, Cone, Cylinder, Cube και Text) είναι χρήσιμοι μόνο για τον σχεδιασμό εξαιρετικά απλών περιβαλλόντων από γεωμετρικής απόψεως. Ωστόσο πολλά X3D αρχεία χρησιμοποιούν έναν άλλο τύπο σχήματος, το IndexedFaceSet. Ένα IndexedFaceSet είναι ένας τρόπος περιγραφής ενός αντικειμένου χρησιμοποιώντας ένα σύνολο κορυφών οι οποίες ενώνονται μεταξύ τους με την χρήση όψεων.

```

geometry IndexedFaceSet {
    coord Coordinate {

```

```

    point [ -1 0 1, 1 0 1, -1 0 -1, 1 0 -1, 0 1 0 ]
  }
  coordIndex [ 0, 1, 4, -1, 1, 3, 4, -1,
              3, 2, 4, -1, 2, 0, 4, -1,
              1, 0, 2, 3, -1 ]
}

```

Στο παραπάνω παράδειγμα ορίζουμε τις 5 κορυφές σαν τιμές του πεδίου `point` (πχ η πρώτη κορυφή είναι η `-1 0 1`), και τις όψεις που συνδέουν τις 5 κορυφές στο πεδίο `CoordIndex` με αναφορές στο πεδίο `points` των `Coordinates`. Ο παραπάνω κώδικας περιγράφει μια πυραμίδα.

### 2.7.4.7 IndexedLineSet

Μερικές φορές είναι προτιμώτερο αν δεν θέλουμε να χρησιμοποιήσουμε μοντέλα που να βασίζονται σε πολύγωνα, αλλά απλές γραμμές, να χρησιμοποιήσουμε το `IndexedLineSet`. Για παράδειγμα, οι ακτίνες ενός δικύκλου μπορεί να αναπαρασταθούν πιο πειστικά με την χρήση απλών γραμμών, αντί για κόμβων κυλίνδρου, πράγμα το οποίο εκτός από πιο εύκολο, είναι και πιο αποδοτικό ως προς το `rendering`. Η σύνταξη είναι ίδια όπως και για το `IndexedFaceSet`, ωστόσο το οπτικό αποτέλεσμα είναι διαφορετικό. Παραθέτουμε τον παρακάτω κώδικα ως παράδειγμα:

```

DEF Burst Shape {
  geometry IndexedLineSet {
    coord Coordinate {
      point [
        0.00 0.00 0.00, 1.00 0.00 0.00,
        0.92 0.38 0.00, 0.71 0.71 0.00,
        0.38 0.92 0.00, 0.00 1.00 0.00,
        -0.38 0.92 0.00, -0.71 0.71 0.00,
        -0.92 0.38 0.00, -1.00 0.00 0.00,
        -0.92 -0.38 0.00, -0.71 -0.71 0.00,
        -0.38 -0.92 0.00, 0.00 -1.00 0.00,
        0.38 -0.92 0.00, 0.71 -0.71 0.00,
        0.92 -0.38 0.00,
      ]
    }
    coordIndex [
      0, 1, -1, 0, 2, -1,
      0, 3, -1, 0, 4, -1,
      0, 5, -1, 0, 6, -1,
      0, 7, -1, 0, 8, -1,
      0, 9, -1, 0, 10, -1,
      0, 11, -1, 0, 12, -1,
      0, 13, -1, 0, 14, -1,
      0, 15, -1, 0, 16, -1
    ]
  }
}

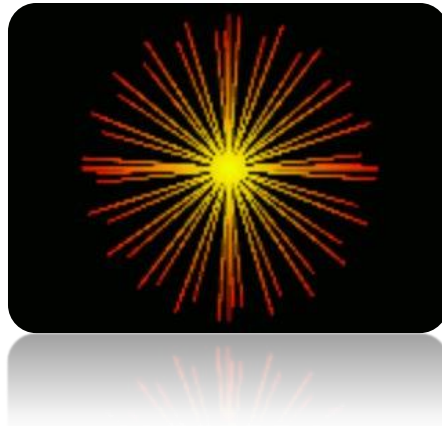
```



Ο παραπάνω κώδικας, με την χρήση επιπλέον των εξής γραμμών ως παραμέτρων μέσα στο {} του IndexedLineSet:

```
colorPerVertex TRUE
  color Color {
    color [
      1.0 1.0 0.0, # burst center color
      1.0 0.3 0.3 # burst ends color
      1.0 0.3 0.3 # burst ends color
      1.0 0.3 0.3 # burst ends color
      1.0 0.3 0.3 # burst ends color
      1.0 0.3 0.3 # burst ends color
      1.0 0.3 0.3 # burst ends color
      1.0 0.3 0.3 # burst ends color
      1.0 0.3 0.3 # burst ends color
      1.0 0.3 0.3 # burst ends color
      1.0 0.3 0.3 # burst ends color
      1.0 0.3 0.3 # burst ends color
      1.0 0.3 0.3 # burst ends color
      1.0 0.3 0.3 # burst ends color
      1.0 0.3 0.3 # burst ends color
      1.0 0.3 0.3 # burst ends color
      1.0 0.3 0.3 # burst ends color
      1.0 0.3 0.3 # burst ends color
    ]
  }
}
```

Δημιουργεί το εξής οπτικό αποτέλεσμα:



**Εικόνα 8:** Το οπτικό αποτέλεσμα ενός IndexedLineSet

#### 2.7.4.8 PointSet

Υπάρχουν φορές που δεν χρειαζόμαστε πολύγωνα, ούτε καν γραμμές. Αυτό που χρειαζόμαστε είναι ξεχωριστά σημεία, πχ όταν δημιουργούμε αστέρια τη νύχτα, θα πρέπει να χρησιμοποιήσουμε points αντί για πολλές μικρές σφαίρες. Το παραπάνω γίνεται με την χρήση ενός PointSet:

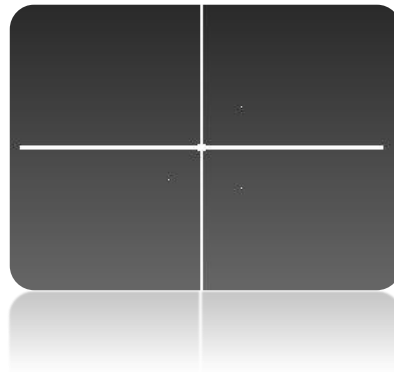
```
Shape {
```

```

appearance Appearance {
    material Material {
        emissiveColor 1 1 1
    }
}
geometry PointSet {
    coord Coordinate {point [
        1.0 1.0 1.0
        -1.0 -1.0 -1.0
        1.0 -1.0 1.0
    ]
}
    color Color { color [
        1.0 1.0 1.0
        1.0 1.0 1.0
        1.0 1.0 1.0
    ]
}
}

```

Ο παραπάνω κώδικας παράγει το εξής οπτικό αποτέλεσμα:



**Εικόνα 9:** Το οπτικό αποτέλεσμα ενός PointSet

### 2.7.4.9 InfoSet

Ο κόμβος Info παρέχει έναν τρόπο αποθήκευσης πληροφορίας κειμένου σε έναν κόμβο. Αυτό μπορεί να χρησιμοποιηθεί κάθε φορά που το χρειάζεται η εφαρμογή. Ο φυλλομετρητής συνήθως το αγνοεί, καθώς η πληροφορία που περιέχει συνήθως ενδιαφέρει μόνο τον χρήστη και δεν έχει κάποιο άμεσο αποτέλεσμα στο rendering. Η σύνταξη είναι πολύ απλή:

```

Info {
    string "some text"
}

```

Αρκετοί φυλλομετρητές κάνουν χρήση ονομασμένων Info κόμβων για να δώσουν έξτρα πληροφορία ως προς το πως θα πρέπει να διαβάσουν την σκηνή. Ωστόσο η προαναφερθείσα τακτική δεν είναι μέρος του προτύπου και δεν συνιστάται.

## 2.7.5 Οι Κόμβοι *Appearance* και *Material*

Οι κόμβοι **Appearance** και **Material** επιτρέπουν τον προσδιορισμό της ακριβούς εμφάνισης των σχημάτων. Η εμφάνιση ενός σχήματος περιγράφεται από δυο σύνολα ιδιοτήτων, το πρώτο εκ των οποίων αφορά τον χρωματισμό του υλικού του σχήματος, και το δεύτερο εκ των οποίων αφορά το texture που ανατίθεται στην επιφάνεια του σχήματος. Επιπρόσθετα στους κόμβους **PositionInterpolator** και **OrientationInterpolator**, η X3D επίσης παρέχει δυο επιπρόσθετους κόμβους Interpolator που μπορούν να χρησιμοποιηθούν για να παράγουν χρώμα και ιδιότητες όπως αδιαφάνεια σε έναν κόμβο **Material**. Αυτοί είναι οι **ColorInterpolator** και **ScalarInterpolator**.

### 2.7.5.1 Κόμβος *Appearance*

Ο κόμβος **Appearance** περιγράφει ιδιότητες εμφάνισης και μπορεί να χρησιμοποιηθεί σαν η τιμή του πεδίου **appearance** σε κάποιον κόμβο σχήματος (**Shape**).

```
Appearance {
    material                NULL
    texture                 NULL
    textureTransform        NULL
}
```

Οι τιμές του πεδίου material προσδιορίζουν έναν κόμβο που περιγράφει τις ιδιότητες του υλικού που ορίζεται σαν η εμφάνιση ενός σχήματος. Ο κόμβος **Material** είναι η συνήθης τιμή που ανατίθεται σε αυτό το πεδίο. Η ανάθεση της εξ' ορισμού τιμής NULL δημιουργεί ένα αστραφτερό άσπρο υλικό.

### 2.7.5.2 Κόμβος *Material*

Παραθέτουμε ενδεικτικά την σύνταξη ενός τέτοιου κόμβου:

```
Material {
    diffuseColor            0.8  0.8  0.8
    emissiveColor           0.0  0.0  0.0
    transparency            0.0
    ambientIntensity        0.2
    specularColor           0.0  0.0  0.0
    shininess               0.2
}
```

Τα πεδία του κόμβου λαμβάνουν ενδεικτικές τιμές από 0.0 έως 1.0, καθορίζοντας πόσο έντονη μπορεί να είναι η κάθε ιδιότητα (πχ transparency) ή τον χρωματισμό της ιδιότητας σε RGB (πχ specularColor).

### 2.7.5.3 Κόμβος *ColorInterpolator*

Ο κόμβος ColorInterpolator χρησιμοποιεί μια λίστα από τμηματικούς χρόνους – κλειδιά και χρωματισμούς – κλειδιά στα πεδία key και keyValue. Όταν η έξοδος fractional time ενός TimeSensor κόμβου δρομολογείται (βλέπε ενότητα περί δρομολόγησης ROUTEs) σε έναν κόμβο ColorInterpolator, ο κόμβος

`ColorInterpolator` χρησιμοποιεί γραμμική παρεμβολή για να υπολογίσει τα ενδιάμεσα χρώματα. Κάθε νεοϋπολογισμένο χρώμα εξάγεται χρησιμοποιώντας το πεδίο **value\_changed** eventOut του κόμβου. Το eventOut μπορεί να δρομολογηθεί στα πεδία χρώματος ενός `Material` κόμβου για να παράγουν εφέ κίνησης στο χρώμα ενός σχήματος ή σε κάποιο άλλο χρώμα του σχήματος (πχ glow).

```
ColorInterpolator {
    key                    [ ]
    keyValue              [ ]
    set_fraction
    value_changed
}
```

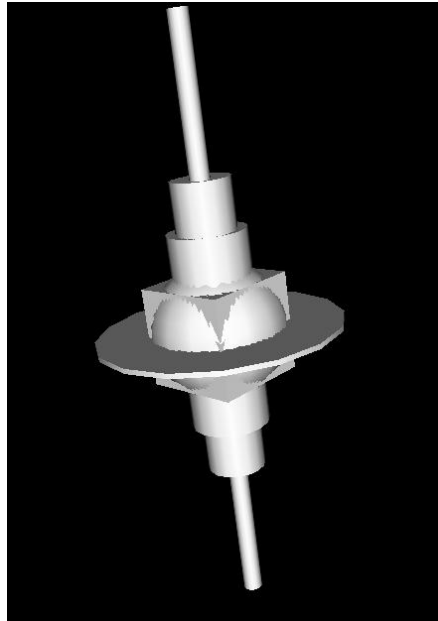
#### 2.7.5.4 Κόμβος `ScalarInterpolator`

Παρόμοια με έναν κόμβο `ColorInterpolator`, ο `ScalarInterpolator` κόμβος χρησιμοποιεί μια λίστα από τμηματικούς χρόνους κλειδιά και τιμές-κλειδιά τύπου float σαν τιμές των πεδίων **key** και **keyValue**. Όταν ένας κόμβος `TimeSensor` δρομολογηθεί με ROUTE σε έναν `ScalarInterpolator` κόμβο, ο κόμβος `ScalarInterpolator` χρησιμοποιεί γραμμική παρεμβολή για να υπολογίσει ενδιάμεσες τιμές κινητής υποδιαστολής. Αυτές οι τιμές εξάγονται με χρήση των **value\_changed** eventOut όπως και στην περίπτωση του `ColorInterpolator` κόμβου. Το eventOut μπορεί εν συνεχεία να δρομολογηθεί μέσω ROUTEs στο πεδίο transparency ενός `Material` κόμβου ώστε να παράγει κίνηση για την αδιαφάνεια (transparency) ενός σχήματος (το οπτικό αποτέλεσμα θα είναι το σχήμα να φαίνεται σαν διάφανο γυαλί). Ο κόμβος `ScalarInterpolator` επίσης μπορεί να χρησιμοποιηθεί για να αποσταλούν τιμές αριθμητικής κινητής υποδιαστολής σε οποιονδήποτε κόμβο που χρησιμοποιεί αριθμητική κινητής υποδιαστολής μονής ακρίβειας σαν η τιμή ενός exposed field.

```
ScalarInterpolator {
    key                    [ ]
    keyValue              [ ]
    set_fraction
    value_changed
}
```

#### 2.7.5.5 Ένα σύνθετο παράδειγμα σε VRML/X3D

Σε αυτή την ενότητα παραθέτουμε ένα παράδειγμα σε VRML/X3D κατασκευασμένο με το Flux Studio 2.1. Παραθέτουμε αρχικά το οπτικό αποτέλεσμα, και στη συνέχεια διαφορετικές εκδόσεις του κώδικα σε 1) .x3d με μορφή XML 2) .x3dv δηλαδή X3D με μορφή VRML αρχείου 3) VRML.



**Εικόνα 10:** Ένα σύνθετο αντικείμενο σε VRML και X3D

### 1) .x3d με μορφή XML

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE X3D PUBLIC "ISO//Web3D//DTD X3D 3.0//EN"
"http://www.web3d.org/specifications/x3d-3.0.dtd">
<X3D profile='Immersive' >
<head>
<meta name='Vizthumbnail' content='Thumb_1_x3d336391249143451.jpg' />
<meta name='title' content='Figure03.10SpaceStation.x3d' />
<meta name='creator' content='Figure 3.10, The VRML 2.0 Sourcebook, Copyright
[1997] By Andrea L. Ames, David R. Nadeau, and John L. Moreland' />
<meta name='translator' content='Don Brutzman' />
<meta name='created' content='6 August 2000' />
<meta name='modified' content='15 September 2002' />
<meta name='description' content='Space Station. Or spark plug? You decide... Also
see Figure10.05MultiColoredSpaceStation.x3d, Figure10.06PulsatingSpaceStation.x3d,
Figure10.10SemiTransparentSpaceStation.x3d and
Figure10.11AnimatingTransparencySpaceStation.x3d' />
<meta name='identifier'
content='http://www.web3d.org/x3d/content/examples/Vrml2.0Sourcebook/Chapter03-
Shapes/Figure03.10SpaceStation.x3d' />
<meta name='generator' content='X3D-Edit 3.2, https://savage.nps.edu/X3D-Edit' />
<meta name='license' content='../..//license.html' />
<meta name='ExportTime' content='19:17:31' />
<meta name='ExportDate' content='8/1/2009' />
<meta name='FluxStudioVersion' content='659' />
</head>
<Scene>
<WorldInfo
title='Untitled'
info='"This Web3D Content was created with Flux Studio, a Web3D authoring tool"'
```

```

"www.mediamachines.com"'/>
<Group DEF='Import_Figure03_10SpaceStation_x3dv'>
<NavigationInfo
containerField='children'
avatarSize='
.25 1.6 .75'
visibilityLimit='0'
speed='1'
headlight='true'
type=""EXAMINE"
"ANY"'/>
<Viewpoint
containerField='children'
description='View from 80m away'
jump='true'
fieldOfView='0.785'
position='0 0 80'
orientation='0 0 1 0'/'>
<Viewpoint
containerField='children'
description='Default viewpoint at (0 0 10) is way too close!! That is a bad place
to be, provide a replacement viewpoint instead.'
jump='true'
fieldOfView='0.785'
position='0 0 10'
orientation='0 0 1 0'/'>
<Group
containerField='children'>
<Shape
containerField='children'>
<Appearance
containerField='appearance'>
<Material DEF='White'
containerField='material'
ambientIntensity='0.200'
shininess='0.200'
diffuseColor='1 1 1'/'>
</Appearance>
<Box
containerField='geometry'
size='10 10 10'/'>
</Shape>
<Shape
containerField='children'>
<Appearance
containerField='appearance'>
<Material

```

```
containerField='material'  
USE='White' />  
</Appearance>  
<Sphere  
containerField='geometry'  
radius='7.000' />  
</Shape>  
<Shape  
containerField='children'>  
<Appearance  
containerField='appearance'>  
<Material  
containerField='material'  
USE='White' />  
</Appearance>  
<Cylinder  
containerField='geometry'  
height='0.500'  
radius='12.500' />  
</Shape>  
<Shape  
containerField='children'>  
<Appearance  
containerField='appearance'>  
<Material  
containerField='material'  
USE='White' />  
</Appearance>  
<Cylinder  
containerField='geometry'  
height='20.000'  
radius='4.000' />  
</Shape>  
<Shape  
containerField='children'>  
<Appearance  
containerField='appearance'>  
<Material  
containerField='material'  
USE='White' />  
</Appearance>  
<Cylinder  
containerField='geometry'  
height='30.000'  
radius='3.000' />  
</Shape>
```

```

<Shape
  containerField='children'>
  <Appearance
    containerField='appearance'>
    <Material
      containerField='material'
      USE='White' />
    </Appearance>
    <Cylinder
      containerField='geometry'
      height='60.000'
      radius='1.000' />
    </Shape>
  </Group>
</Group>
</Scene>
</X3D>

```

## 2) .x3dv

```

#X3D V3.0 utf8
PROFILE Immersive
META "title" "Figure03.10SpaceStation.x3d"
META "creator" "Figure 3.10, The VRML 2.0 Sourcebook, Copyright [1997] By Andrea L. Ames, David R. Nadeau, and John L. Moreland"
META "translator" "Don Brutzman"
META "created" "6 August 2000"
META "modified" "15 September 2002"
META "description" "Space Station. Or spark plug? You decide... Also see Figure10.05MultiColoredSpaceStation.x3d, Figure10.06PulsatingSpaceStation.x3d, Figure10.10SemiTransparentSpaceStation.x3d and Figure10.11AnimatingTransparencySpaceStation.x3d"
META "identifier" "http://www.web3d.org/x3d/content/examples/Vrml2.0Sourcebook/Chapter03-Shapes/Figure03.10SpaceStation.x3d"
META "generator" "X3D-Edit 3.2, https://savage.nps.edu/X3D-Edit"
META "license" "../license.html"
META "ExportTime" "19:17:39"
META "ExportDate" "8/1/2009"
META "FluxStudioVersion" "659"
WorldInfo {
  title "Untitled"
  info ["This Web3D Content was created with Flux Studio, a Web3D authoring tool"
  "www.mediamachines.com"]
}
## Vizthumbnail Thumb_2_x3dv417611249143459.jpg
DEF Import_Figure03_10SpaceStation_x3dv Group {
  children [
  NavigationInfo {
  avatarSize [
  .25 1.6 .75
  ]
  ]
  visibilityLimit 0
  speed 1
  headlight TRUE

```



```
type ["EXAMINE"  
"ANY"]  
}  
Viewpoint {  
description "View from 80m away"  
jump TRUE  
fieldOfView 0.785  
position 0 0 80  
orientation 0 0 1 0  
}  
Viewpoint {  
description "Default viewpoint at (0 0 10) is way too close!! That is a bad place to be, provide a replacement  
viewpoint instead."  
jump TRUE  
fieldOfView 0.785  
position 0 0 10  
orientation 0 0 1 0  
}  
Group {  
children [  
Shape {  
appearance Appearance {  
material DEF White Material {  
ambientIntensity 0.200  
shininess 0.200  
diffuseColor 1 1 1  
}  
}  
geometry Box {  
size 10 10 10  
}  
}  
Shape {  
appearance Appearance {  
material USE White  
}  
geometry Sphere {  
radius 7.000  
}  
}  
Shape {  
appearance Appearance {  
material USE White  
}  
geometry Cylinder {  
height 0.500  
radius 12.500  
}  
}  
Shape {  
appearance Appearance {  
material USE White  
}  
geometry Cylinder {  
height 20.000
```

```
radius 4.000
}
}
Shape {
appearance Appearance {
material USE White
}
geometry Cylinder {
height 30.000
radius 3.000
}
}
Shape {
appearance Appearance {
material USE White
}
geometry Cylinder {
height 60.000
radius 1.000
}
}
]
}
]
}
```

### 3) .wrl με μορφή κώδικα VRML2.0

```
#VRML V2.0 utf8
WorldInfo {
title "Untitled"
info ["This Web3D Content was created with Flux Studio, a Web3D authoring tool"
"www.mediamachines.com"]
}
## Vizthumbnail Thumb_3_wrl466281249143464.jpg
DEF Import_Figure03_10SpaceStation_x3dv Group {
children [
NavigationInfo {
avatarSize [
.25 1.6 .75
]
visibilityLimit 0
speed 1
headlight TRUE
type ["EXAMINE"
"ANY"]
}
Viewpoint {
description "View from 80m away"
jump TRUE
fieldOfView 0.785
position 0 0 80
orientation 0 0 1 0
}
Viewpoint {
```

```
description "Default viewpoint at (0 0 10) is way too close!! That is a bad place to be, provide a replacement
viewpoint instead."
jump TRUE
fieldOfView 0.785
position 0 0 10
orientation 0 0 1 0
}
Group {
children [
Shape {
appearance Appearance {
material DEF White Material {
ambientIntensity 0.200
shininess 0.200
diffuseColor 1 1 1
}
}
geometry Box {
size 10 10 10
}
}
Shape {
appearance Appearance {
material USE White
}
geometry Sphere {
radius 7.000
}
}
Shape {
appearance Appearance {
material USE White
}
geometry Cylinder {
height 0.500
radius 12.500
}
}
Shape {
appearance Appearance {
material USE White
}
geometry Cylinder {
height 20.000
radius 4.000
}
}
Shape {
appearance Appearance {
material USE White
}
geometry Cylinder {
height 30.000
radius 3.000
}
}
```

```

}
Shape {
  appearance Appearance {
    material USE White
  }
  geometry Cylinder {
    height 60.000
    radius 1.000
  }
}
}
}
}
}
}
}
}
}

```

## 2.7.6 *Textures και Texture Mapping*

Οι λειτουργίες Texture Mapping στην X3D εκτελούνται με διαδικασίες πολλαπλών – βημάτων. Η διαδικασία πάντα ξεκινά με το διάβασμα από αρχείο ή με την δημιουργία μιας εικόνας texture χρησιμοποιώντας κάποιον κόμβο ImageTexture, PixelTexture, ή MovieTexture. Τα βήματα στην διαδικασία Texture Mapping μπορούν να συνοψιστούν ως εξής:

- Επιλογή μιας texture εικόνας χρησιμοποιώντας ImageTexture, PixelTexture ή MovieTexture κόμβο.
- Χρησιμοποίηση 2D συντεταγμένων (coordinates) για να περιγράψουν την περίμετρο του texture που πρόκειται να αντιστοιχηθεί.
- Το δέσιμο του texture σε κάποια επιφάνεια πάνω στο VRML σχήμα.
- Η κάθε γωνία ενός κομματιού texture εν συνεχεία προσδέεται σε μια 3D συντεταγμένη πάνω στην επιφάνεια. Αν είναι απαραίτητο, η εικόνα του texture εκτείνεται για να καλύψει πλήρως την επιφάνεια.

### 2.7.6.1 Κόμβοι Texture

Η X3D παρέχει τρεις κόμβους για τον προσδιορισμό του texture που πρόκειται να ταιριαχθεί σε κάποιο σχήμα. Αυτοί είναι οι: ImageTexture, PixelTexture και MovieTexture.

Ο κόμβος **ImageTexture** καθορίζει κάποιες ιδιότητες του texture mapping οι οποίες χρησιμοποιούνται σαν τιμή ενός πεδίου texture σε έναν κόμβο Appearance (εμφάνισης). πχ:

```

ImageTexture {
  url [ ]
  repeatS TRUE
  repeatT TRUE
}

```

Τα repeatS και repeatT ορίζουν κάθετη ή οριζόντια επαναληπτική χρήση του texture αντίστοιχα.

Ο κόμβος **PixelTexture** καθορίζει την τοποθεσία της εικόνας που θα χρησιμοποιηθεί για texturing του σχήματος, καθώς και το αν η εικόνα θα χρησιμοποιηθεί επαναληπτικά (όπως και στον προηγούμενο τύπο κόμβου) κάθετα ή οριζόντια κατά μήκος των άξων του σχήματος.

```
PixelTexture {
    image 0 0 0
    repeatS TRUE
    repeatT TRUE
}
```

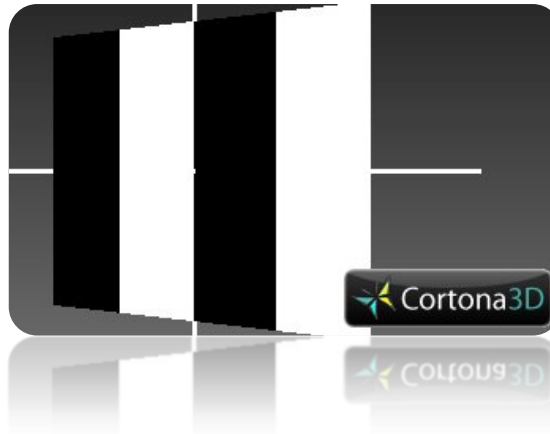
Από τα τρία πεδία που εμφανίζονται σε αυτό τον τύπο Κόμβου, τα δυο τελευταία είναι όπως και στον προηγούμενο. Το πεδίο image καθορίζει την εικόνα που χρησιμοποιεί pixels: οι πρώτες τρεις τιμές του πεδίου image καθορίζουν το πλάτος της εικόνας (σε pixels), το ύψος της εικόνας (σε pixels) και τον αριθμό των bytes για κάθε pixel. Οι πιθανές τιμές για τον αριθμό των bytes είναι:

- Grayscale
- Grayscale με χρήση alpha channel για transparency
- RGB
- RGB με χρήση alpha channel για transparency

Όλα τα πεδία είναι προαιρετικά, ενώ αν κάποιο δεν συμπληρωθεί χρησιμοποιούνται οι προκαθορισμένες τιμές. Ο προσδιορισμός ενός pixel εικόνας pixel-προς-pixel μπορεί να είναι τεχνικά δύσκολος, ωστόσο σε ορισμένες περιπτώσεις με λίγη προσπάθεια μπορούν να επιτευχθούν ενδιαφέροντα αποτελέσματα, πχ:

```
PixelTexture {
    image 2 1 1 0 255
}
```

Το παραπάνω παράγει μια εικόνα μεγέθους 2 pixels, με υψος 1 pixel, εκ των οποίων το 1ο pixel είναι μαύρο και το δεύτερο είναι λευκό. Αν αυτή η εικόνα ανατεθεί σαν texture σε μια επιφάνεια, καθώς θα εκταθεί για να «χωρέσει» στην επιφάνεια, μπορεί να προκύψει οπτικό αποτέλεσμα σαν το παρακάτω:



**Εικόνα 11:** Ένας κύβος που χρησιμοποιεί PixelTexture

Ο κόμβος **MovieTexture** καθορίζει με παρόμοιο τρόπο texture mapping για να χρησιμοποιηθεί σαν τιμή του πεδίου texture σε έναν κόμβο Appearance. Δοθέντος ενός αρχείου ταινίας το οποίο περιέχει το audio που θέλουμε να χρησιμοποιήσουμε, ο κόμβος MovieTexture καθορίζει τον ήχο ο οποίος μπορεί να αναπαραχθεί καθώς ο κόμβος ανατίθεται ως τιμή του πεδίου source στον κόμβο Sound.

```
MovieTexture {
    url                [ ]
    loop               FALSE
    speed              1.0
    startTime          0.0
    stopTime           0.0
    repeats            TRUE
    repeatT            TRUE
    isActive
    duration_changed
}
```

Ο κόμβος **TextureCoordinate** καθορίζει μια λίστα από συντεταγμένες για το texture οι οποίες χρησιμοποιούνται σαν οι τιμές του πεδίου texCoord σε κάποιον κόμβο IndexedFaceSet ή ElevationGrid.

```
TextureCoordinate {
    point              [ ]
}
```

Ο κόμβος **TextureTransform** δομεί ένα νέο σύστημα συντεταγμένων για το texture, σε σχέση με το αρχικό σύστημα συντεταγμένων του texture. Ένας κόμβος TextureTransform τυπικά κατέχει την τιμή ενός πεδίου textureTransform για κάποιον κόμβο Appearance.

```
TextureTransform {
    translation        0.0 0.0
    rotation           0.0
```

```

    scale          1.0 1.0
    center         0.0 0.0
}

```

Όταν ένα texture αντιστοιχίζεται σε ένα primitive σχήμα, η ίδια εικόνα τοποθετείται πάνω σε όλες τις όψεις του σχήματος. Για να αντιστοιχίσουμε μια διαφορετική εικόνα σε κάθε μέρος του σχήματος, μπορούμε να φτιάξουμε διαφορετικά σχήματα, το καθένα με την δικιά του εικόνα για texture, για παράδειγμα, μπορούμε να έχουμε έναν κώνο με μια εικόνα texture για την πλευρά του κώνου και άλλη εικόνα για την βάση.

### 2.7.6.2 Ενσωματώνοντας Εικόνες Texture σε ένα X3D αρχείο

Οι εικόνες texture που χρησιμοποιούνται συνήθως αποθηκεύονται σε ξεχωριστά αρχείων εικόνων (\*.bmp, \*.jpg, \*.gif) και συμπεριλαμβάνονται προς χαρτογράφηση σε έναν X3D κόσμο με χρήση του κόμβου **ImageTexture** (22). Μερικές φορές είναι συμφερότερο να ενσωματώσουμε τα δεδομένα των pixels της εικόνας στον ίδιο τον κόμβο **ImageTexture**, και κατά συνέπεια στο ίδιο το VRML αρχείο.

Το παραπάνω γίνεται χρησιμοποιώντας έναν κόμβο **PixelTexture**, όπως παρουσιάστηκε στην προηγούμενη ενότητα, στην θέση ενός κόμβου **ImageTexture**. Η συμπερίληψη μιας εικόνας texture σε ένα X3D αρχείο ωστόσο είναι συμφέρουσα μόνο στην περίπτωση που το X3D αρχείο είναι πολύ μικρό, και μόνο σε ορισμένες περιπτώσεις. Για παράδειγμα, μια εικόνα texture που περιέχει μόνο ελάχιστα pixels κατά πλάτος και ύψος μπορεί να χρησιμοποιηθεί για την δημιουργία ζωνών χρωμάτων σε μια όψη ενός σχήματος. Τα δεδομένα εικόνας σε έναν τέτοιο κόμβο **PixelTexture** θα μπορούσαν να δημιουργήσουν μια RGB texture εικόνα πλάτους 1 pixel, ύψους 2 pixels, συνολικά 3 bytes, η οποία, εάν αντιστοιχούνταν σε κάποια επιφάνεια σχήματος θα παρήγαγε ένα οπτικό αποτέλεσμα διαχωρισμού ζωνών κατά μήκος όλων των επιφανειών του σχήματος.

### 2.7.6.3 Συντεταγμένες Texture

Όλα τα γεωμετρικά σχήματα σε X3D, συμπεριλαμβανομένων των κόμβων **Box**, **Cone**, **Cylinder**, **Sphere**, **Text**, **IndexedFaceSet** και **Extrusion**, αυτομάτως δημιουργούν ένα δικό τους σύστημα συντεταγμένων για την αντιστοίχιση ενός texture πάνω στις επιφάνειές τους. Αυτό επιτρέπει αυτόματο texturing σε πρωτόγονα γεωμετρικά σχήματα, χωρίς να χρειαστεί να προσδιορίσουμε εμείς κάποιο σύστημα συντεταγμένων. Ωστόσο, για πιο πολύπλοκους γεωμετρικούς κόμβους, όπως ο **IndexedFaceSet** και ο **ElevationGrid** που χρησιμοποιούνται για τον προσδιορισμό μη-καθορισμένης γεωμετρίας, η X3D παρέχει την δυνατότητα επικάλυψης των συντεταγμένων που δημιουργούνται αυτόματα με τον προσδιορισμό ενός νέου σετ συντεταγμένων texture, με την χρήση ενός **TextureCoordinate** κόμβου. Ένας τέτοιος κόμβος χρησιμοποιείται ως το **texCoord** πεδίο αντίστοιχα ενός κόμβου τύπου **IndexedFaceSet** ή **ElevationGrid**.

Το σχήμα της περιμέτρου μιας 2D εικόνας texture προς αντιστοίχιση σε κάποια όψη περιγράφεται με την χρήση μιας λίστας από 2D συντεταγμένες texture. Κάθε 2D συντεταγμένη καθορίζει μια τοποθεσία στο σύστημα συντεταγμένων όπου οι αποστάσεις μετριοούνται σε S (οριζόντια) και T (κάθετη) κατεύθυνση στην επιφάνεια της εικόνας texture. Μια εικόνα texture πάντα εκτείνεται από 0.0 έως 1.0 στην S

κατεύθυνση, και από 0.0 έως 1.0 στην T κατεύθυνση στο σύστημα συντεταγμένων του texture. Αυτό το εύρος από S και T τιμές είναι ανεξάρτητο από το πραγματικό ύψος και πλάτος της εικόνας σε pixels, δεδομένου ότι οι συντεταγμένες (0.5,0.5) είναι πάντα το κέντρο της εικόνας.

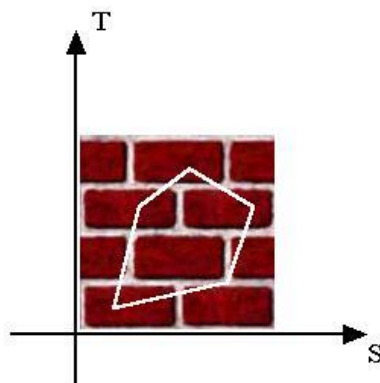
Η λίστα συντεταγμένων που δίνεται στον παρακάτω πίνακα περιγράφει την περίμετρο μιας εικόνας texture προς αντιστοίχιση σε κάποια επιφάνεια, το οποίο θα έχει σαν αποτέλεσμα την αποκοπή ενός τμήματος του texture που περιέχει ολόκληρη την εικόνα.

Texture Coordinate Index	Coordinate ( S, T )	
0	0.0	0.0
1	1.0	0.0
2	1.0	1.0
3	0.0	1.0

Η παραπάνω λίστα συντεταγμένων καθορίζει μια τετράγωνη 2D εικόνα texture βάσει 4 συντεταγμένων. Είναι πιθανό να διαλέξουμε μια περίμετρο οποιουδήποτε μεγέθους και σχήματος από το περιεχόμενο μιας εικόνας texture, χρησιμοποιώντας οποιονδήποτε αριθμό από συντεταγμένες texture. Για παράδειγμα, το παρακάτω σύστημα συντεταγμένων:

Texture Coordinate Index	Coordinate ( S, T )	
0	0.1	0.1
1	0.6	0.2
2	0.9	0.7
3	0.5	0.9
4	0.2	0.7

Έχει ως αποτέλεσμα την αποκοπή του παρακάτω τμήματος από την εικόνα:



**Εικόνα 12:** Αποκοπή Τμήματος από Texture

Οι συντεταγμένες texture μπορούν να χρησιμοποιηθούν επίσης για να επιλέξουμε περιοχές έξω από την εικόνα του texture, δηλαδή εκτός των ορίων της εικόνας. Όταν οι περιοχές εκτός εικόνας επιλέγονται οι συντεταγμένες του texture *αναδιπλώνονται* ώστε οι συντεταγμένες πάνω από την κορυφή της εικόνας να αναδιπλωθούν στο κάτω μέρος της εικόνας, ενώ οι συντεταγμένες κάτω από την κάτω αιχμή της εικόνας



αναδιπλώνονται στην κορυφή. Το ίδιο οπτικό αποτέλεσμα ισχύει για τις δεξιές και αριστερές άκρες. Κατά συνέπεια, ενώ μια συντεταγμένη (1.0, 0.0) περιγράφει μια τοποθεσία πάνω στην δεξιά άκρη της εικόνας texture, μια συντεταγμένη (1.2, 0.0) περιγράφει ένα σημείο στο αριστερό άκρο της διπλανής (ίδιας με επανάληψη) εικόνας στα δεξιά. Η εξ' ορισμού επιλογή για όλους τους κόμβους texturing είναι η αναδίπλωση στο σύστημα συντεταγμένων να είναι ενεργή. Η αναδίπλωση μπορεί να γίνει ανενεργή για τις S και T κατευθύνσεις θέτοντας τα πεδία **repeatS** και **repeatT** στους κόμβους **ImageTexture**, **PixelTexture** και **MovieTexture** στην τιμή FALSE.

Η *στερέωση* Συντεταγμένων Texture χρησιμοποιείται για να εξασφαλίσει ότι μια εικόνα texture εμφανίζεται μόνο μια φορά σε ένα σχήμα, δηλαδή για να εξασφαλίσει ότι η εικόνα δεν θα *αναδιπλωθεί*. Είναι πιθανό η εικόνα να στερεωθεί σε οποιαδήποτε από τις 2 κατευθύνσεις S και T, ή ταυτόχρονα και στις 2. Όταν η στερέωση γίνεται ως προς την S κατεύθυνση, οι S συντεταγμένες με τιμές μικρότερες του 0.0 ή μεγαλύτερες του 1.0 στερεώνονται στις αριστερές και δεξιές πλευρές της εικόνας αντίστοιχα. Η στερέωση συντεταγμένων texture ελέγχεται με χρήση των πεδίων **repeatS** και **repeatT** τα οποία είναι διαθέσιμα στους κόμβους **ImageTexture**, **PixelTexture** και **MovieTexture**. Όταν τα παραπάνω πεδία τίθενται στην τιμή FALSE, συμβαίνει στερέωση των συντεταγμένων.

#### 2.7.6.4 Ορισμός Transform για Συντεταγμένες Texture

Ο κόμβος **TextureTransform** επιτρέπει την δημιουργία ενός νέου συστήματος συντεταγμένων texture. Αλλαγές στις τιμές των πεδίων **translation**, **rotation** και **scale** ενός κόμβου **TextureTransform** επιτρέπουν επανατοποθέτηση, περιστροφή και κλιμάκωση της περιμέτρου μέσα στην εικόνα **texture** η οποία έχει οριστεί προς αντιστοίχιση (ή του τμήματός της).

Οι *μεταφράσεις* συντεταγμένων texture ορίζονται σαν αποστάσεις οι οποίες μετριοούνται κατά μήκος των S και T κατευθύνσεων μιας εικόνας texture και ορίζονται χρησιμοποιώντας το πεδίο translation ενός **TextureTransform** κόμβου.

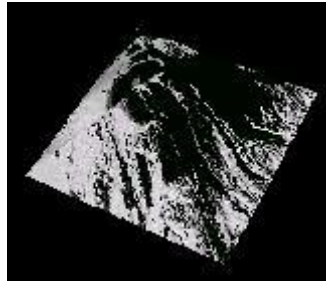
Η *περιστροφή* συντεταγμένων texture έχει το ίδιο αποτέλεσμα με την αλλαγή προσανατολισμού της περιμέτρου μέσα στην εικόνα texture πριν αυτή αποκοπεί βάσει συντεταγμένων. Οι συντεταγμένες μπορούν να περιστραφούν ως προς οποιοδήποτε σημείο μέσα στην εικόνα του texture. Το πεδίο **center** μέσα στον κόμβο **TextureTransform** χρησιμοποιείται για αυτό τον σκοπό. Η αλλαγή προσανατολισμού μετρείται σε rad χρησιμοποιώντας το πεδίο rotation.

Η *κλιμάκωση* των συντεταγμένων texture αυξάνει ή μειώνει το μέγεθος της περιμέτρου μέσα στην εικόνα του texture πριν αυτό αποκοπεί και αντιστοιχηθεί. Όπως και με την περιστροφή συντεταγμένων, ένα σημείο κέντρου μπορεί να επιλεγεί για κλιμάκωση.

#### 2.7.6.5 Αντιστοίχιση Texture για ElevationGrids

Σε περιπτώσεις στις οποίες οι συντεταγμένες texture που έχουν υπολογιστεί αυτομάτως για έναν κόμβο ElevationGrid δεν κριθούν ικανοποιητικές, επιπρόσθετος έλεγχος μπορεί να αποκτηθεί πάνω στην αντιστοίχιση παρέχοντας ρητά συντεταγμένες texture, η κάθε μια για κάθε σημείο πλέγματος. Παραθέτουμε ένα

παράδειγμα στο οποίο η φωτογραφική εικόνα ενός ηφαιστείου αντιστοιχίζεται σε έναν κόμβο ElevationGrid ώστε εμφανισιακά να θυμίζει τοπίο βουνού:



**Εικόνα 13:** Παράδειγμα Texturing ενός κρατήρα με χρήση ElevationGrid

```
Shape {
  appearance Appearance {
    material Material { }
    texture ImageTexture {
      url "images/volcnew.jpg"
    }
  }
  geometry ElevationGrid {
    xDimension 9
    zDimension 9
    xSpacing 1.0
    zSpacing 1.0
    solid FALSE
    height [
      0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
      0.0, 1.0, 0.0, 0.0, 1.0, 0.0, 1.0, 1.0, 0.0,
      0.0, 0.0, 2.0, 0.0, 2.0, 2.0, 2.0, 0.0, 0.0,
      0.0, 0.0, 0.0, 3.0, 3.0, 3.0, 1.0, 0.0, 0.0,
      0.0, 1.0, 2.0, 3.0, 3.5, 3.0, 2.0, 1.0, 0.0,
      0.0, 0.0, 1.0, 3.0, 3.0, 3.0, 0.0, 0.0, 0.0,
      0.0, 0.0, 2.0, 2.0, 2.0, 0.0, 2.0, 0.0, 0.0,
      0.0, 1.0, 1.0, 1.0, 1.0, 0.0, 0.0, 1.0, 0.0,
      0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
    ]
  }
}
```

## 2.7.7 Γεγονότα και δρομολόγηση

### 2.7.7.1 Γεγονότα

Η κυρίως διαφορά μεταξύ των εκδόσεων της VRML 1.0 και 2.0 ήταν το να μπορεί κάθε κόμβος να στείλει και να λάβει γεγονότα (events). Στην VRML2.0 μπορούμε να ορίσουμε γεγονότα που να αντιστοιχίζουν μεταβολές στα αντικείμενα (πχ κίνηση) ως αντίδραση στις επιλογές του χρήστη, με τις παραγόμενες τιμές των γεγονότων εξαρτώνται από τον τύπο των δεδομένων τους.

Στην VRML 2.0, όταν ορισμένα πεδία ενός κόμβου αλλάζουν τιμή αφού ο κόσμος έχει φορτωθεί, ένα συμβάν παράγεται. Για τους παραπάνω σκοπούς χρησιμοποιούνται τα ειδικά πεδία τύπου exposed field, δηλαδή πεδία που μπορούν να

τεθούν όταν ο χρήστης βλέπει τον κόμβο. Ένα exposed field μπορεί να αποσυντεθεί στις ακόλουθες τρεις γραμμές:

```
eventIn set_fieldName
field fieldName
eventOut fieldName_changed
```

όπου το fieldName μπορεί να είναι οποιοδήποτε exposed field.

Η δήλωση ενός exposed field υπονοεί ότι ο κόμβος είναι σε θέση να δεχθεί γεγονότα eventIn (εισερχόμενα γεγονότα), και να παράγει γεγονότα eventOut (εξερχόμενα γεγονότα). Το eventIn σε ένα exposed field χρησιμοποιείται για να θέσει την τιμή του πεδίου. Όταν η τιμή του πεδίου αλλάξει, ο κόμβος στον οποίο το πεδίο καθορίζεται θα παράγει το αντίστοιχο eventOut. Εκτός από exposed fields, οι κόμβοι μπορούν να έχουν άλλα πεδία eventIn ή eventOut, ωστόσο τα πεδία αυτά δεν είναι παρόντα στον ορισμό του κόμβου σε ένα X3D αρχείο.

Όταν συμβαίνει ένα γεγονός, ο κόμβος ο οποίος παρήγαγε το γεγονός παράγει μια τιμή ή ένα σύνολο από τιμές ενός δοθέντος τύπου δεδομένων ανάλογα με τον τύπο του γεγονότος. Όταν κάποιος στέλνει ένα γεγονός σε έναν κόμβο, στην πραγματικότητα στέλνει στον κόμβο τιμές ή ένα σύνολο τιμών. Ο κόμβος προσδιορίζει τι θα πρέπει να κάνει το γεγονός με τις τιμές ή με το σύνολο τιμών που του παρέχονται.

Υπάρχουν δυο τρόποι για να στείλουμε ένα γεγονός σε έναν κόμβο, όπως και το να λάβουμε ένα γεγονός που έχει παραχθεί από κάποιον κόμβο. Ο πρώτος είναι τα ROUTEs (δρομολογήσεις) και ο δεύτερος τα scripts.

Ως προς το πότε παράγονται γεγονότα, εξαιρουμένης της περίπτωσης στην οποία έχουν ως αποτέλεσμα την αλλαγή της τιμής ενός πεδίου, υπάρχουν οι εξής πιθανότητες:

- Timers (Χρονιστές), οι οποίοι παράγουν γεγονότα σε καθορισμένα χρονικά διαστήματα.
- Touch Sensors (αισθητήρες αφής) οι οποίοι παράγουν γεγονότα όταν ο κέρσορας του ποντικιού είναι πάνω από αντικείμενα που ανήκουν στην ίδια ομάδα με τον αισθητήρα.
- Visibility Sensors (Αισθητήρες Ορατότητας), οι οποίοι παράγουν γεγονότα όταν σχήματα μέσα στην ίδια ομάδα με τον αισθητήρα.
- Dragging Sensors (Αισθητήρες συρσίματος) οι οποίοι παράγουν γεγονότα όταν ο χρήστης κάνει κλικ με τον δείκτη του ποντικιού και σύρει στην οθόνη αντικείμενα μέσα στην ίδια ομάδα με τον αισθητήρα.
- Proximity Sensors (Αισθητήρες Αποστάσεως) οι οποίοι παράγουν γεγονότα όταν ο χρήστης είναι μέσα σε ένα προκαθορισμένο αντικείμενο.

### 2.7.7.2 Δρομολόγηση με ROUTEs

Ένας κόμβος, όπως προαναφέρθηκε, μπορεί να στείλει και να λάβει γεγονότα. Τα ROUTEs είναι ένας απλός τρόπος περιγραφής ενός μονοπατιού μεταξύ ενός

γεγονότος που έχει δημιουργηθεί από έναν κόμβο και ενός κόμβου που λαμβάνει το γεγονός (23). Η σύνταξη περιγράφεται ως εξής:

```
ROUTE Node.eventOut_changed TO Node.eventIn_changed
```

Εξαιτίας του ότι όλα τα exposed fields καθορίζουν εμμέσως δυο πεδία, ένα eventIn και ένα eventOut, δεν είναι απαραίτητο να καθορίσουμε ρητά τα προθέματα 'set\_' ή '\_changed'. Η ακόλουθη σύνταξη είναι έγκυρη σε VRML, αρκεί τα eventIn και eventOut τα οποία εμφανίζονται στην πρόταση ROUTE να ανήκουν σε exposed fields.

```
ROUTE Node.eventOut TO Node.eventIn
```

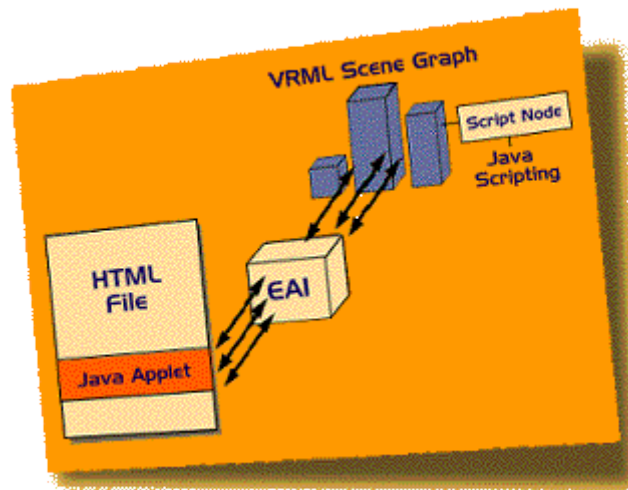
Κάθε φορά που ένα exposed field αλλάζει, παράγεται ένα αντίστοιχο γεγονός, πράγμα που σημαίνει ότι είναι δυνατόν να έχουμε την δημιουργία επικαλυπτόμενων γεγονότων. Μια ένδειξη χρόνου ανατίθεται σε κάθε γεγονός που δημιουργείται για να επιλύσει το παραπάνω πρόβλημα.

Η νέα τιμή για το πεδίο που σχετίζεται με το eventIn είναι η τιμή του πεδίου η οποία προκάλεσε την δημιουργία του eventOut.

## 2.8 VRML, X3D ΚΑΙ JAVA

### 2.8.1 *Vrml, X3D, Java και EAI*

Ήδη από την πρώτη εμφάνιση της VRML έχει προκληθεί αρκετό ενδιαφέρον για την χρησιμοποίηση της VRML στον προγραμματισμό Java εφαρμογών, οι οποίες να χρησιμοποιούν ένα αλληλεπιδραστικό 3D περιβάλλον διεπαφής που να αξιοποιεί τις δυνατότητες της VRML (24). Μια τέτοια διεπαφή μπορεί να χρησιμοποιεί για την οργάνωση πολύπλοκων δεδομένων με την επιπρόσθετη δυνατότητα δυναμικής επέμβασης πάνω στα δεδομένα σε πραγματικό χρόνο. Κρίσιμος στα παραπάνω είναι ο ρόλος του πακέτου External Authoring Interface (EAI) στην γεφύρωση και την σύνδεση των Java Εφαρμογών σε μια 3D VRML σκηνή.



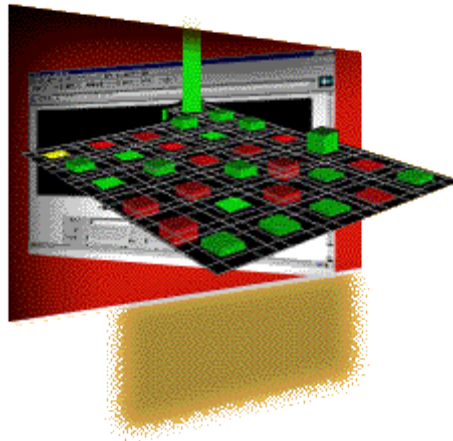
**Εικόνα 14:** Αλληλεπίδραση μεταξύ Java Applet, EAI και VRML σκηνής

Το παραπάνω σχήμα απεικονίζει την σχέση μεταξύ ενός Java Applet, του του πακέτου EAI και μιας VRML σκηνής. Το EAI παρέχει την διεπαφή που επιτρέπει στο Java Applet να αλληλεπιδράει με την VRML σκηνή.

Η ευρεία αποδοχή της VRML σαν τον πρώτο πολυχρησιμοποιούμενο τύπο αρχείου για την δημιουργία πραγματικού-χρόνου, πλούσιου σε συμπεριφορικές δυνατότητες 3D υλικού έχει εμφανείς επιπτώσεις στο πως μπορούμε να φανταστούμε την πληροφορία. Ενώ η Java σε πρώτη φάση άλλαξε δραματικά τον τρόπο με τον οποίο δημιουργούνται και διαμοιράζονται εφαρμογές, η VRML προχωράει ένα επίπεδο παραπέρα, αλλάζοντας την ίδια την φύση των εφαρμογών, παράλληλα εμπλουτίζοντας και εμβαθύνοντας το νόημα των δεδομένων που επεξεργάζεται.

Σαν ένα παράδειγμα μπορούμε να αναφέρουμε την Transanction Information Systems, μια εταιρία με έδρα την Νέα Υόρκη, η οποία χρησιμοποιεί Java, VRML, και το πακέτο EAI για να αναπτύξει λογισμικό οπτικοποίησης για οικονομικούς σκοπούς. Μια από τις χαρακτηριστικές εφαρμογές που έχουν αναπτύξει επεκτείνει τον τρόπο που οικονομικά δεδομένα (μετοχών - stocks) αναπαριστώνται στον υπολογιστή, αυτή τη φορά χρησιμοποιώντας VRML για να αλλάξουν την φύση μιας μηχανής εισιτηρίων της Wall Street. Δεδομένα ανατροφοδότησης από την ίδια την αγορά αποστέλλονται στην εφαρμογή, η οποία τα επεξεργάζεται και αναθέτει σε ένα Java Application από την πλευρά του εξυπηρετητή να αποστείλει τα δεδομένα σε ένα Java applet στην πλευρά του πελάτη. Ένα VRML αρχείο, το οποίο αποτελείται από τρισδιάστατες γραμμές πλέγματος, με τις κυψέλες να αναπαριστούν μετοχές, χτίζεται δυναμικά και ενημερώνεται μέσω του EAI, βασισμένο στις αλλαγές που λαμβάνει από το Java applet.

Κάθε φορά που οι τιμές των μετοχών αλλάζουν, το applet πυροδοτεί ένα animation ενός VRML κύβου, προκαλώντας τον κύβο να περιστραφεί κατά 90° μέσα στον 3D VRML κόσμο, ενώ ταυτόχρονα προβάλλει την ανανεωμένη του τιμή. Άλλες οπτικοποιήσεις μέσα στον VRML κόσμο, όπως το χρώμα και το μέγεθος μιας μετοχής σχετιζόμενο με κάποια VRML γεωμετρία, χρησιμοποιούνται για να σηματοδοτήσουν την άνοδο ή την πτώση κάποιας μετοχής, καθώς και την σχετική αγοραστική ισχύ της. Επιπρόσθετα, ο κάθε κύβος μπορεί να προβάλλει πληροφορίες σχετικά με την μετοχή όταν ο χρήστης κάνει κλικ με τον δείκτη του ποντικιού πάνω του. Με το παραπάνω σύστημα, η συνολική δυναμική των μετοχών της αγοράς μπορεί να παρατηρηθεί σαν εννιαία εικόνα.



**Εικόνα 15:** παράδειγμα 3D αναπαράστασης της κίνησης μετοχών στο Χρηματιστήριο

Μια καθαρή αντίληψη του πως το EAI χτίζει δυναμικά τα VRML αρχεία βασιζόμενο στα δεδομένα που λαμβάνει μέσω του Java Applet και πως, με την σειρά τους, τα δεδομένα της εφαρμογής μπορούν να ανανεωθούν δυναμικά μέσα από την διεπαφή της VRML είναι κρίσιμη για τον προγραμματιστή που θέλει να παράγει τέτοιου είδους προϊόντα και αλληλεπιδράσεις. Αυτό συμπεριλαμβάνει προγραμματιστές που δουλεύουν σε ποικίλους χώρους όπως η διαχείριση βάσεων δεδομένων, η ψυχαγωγία, τα πολυμέσα, οι ιατρικές και επιστημονικές οπτικοποιήσεις, και η εκπαίδευση, όλοι από τους οποίους βρίσκουν τον συνδυασμό Java, VRML και EAI ταιριαστό στις ξεχωριστές στρατηγικές ανάπτυξής τους.

## 2.8.2 Ενσωμάτωση στην Java

### 2.8.2.1 Σύγκριση VRML/X3D και JAVA

Η Java και η VRML/X3D μπορούν να λειτουργήσουν σχεδόν σαν τέλειο συμπλήρωμα στον παγκόσμιο Ιστό. Και οι δυο σχεδιάστηκαν από την αρχή σαν τεχνολογίες διαδικτύου. Αλλά η καθεμιά επέλυε ένα διαφορετικό, και συμπληρωματικό, πρόβλημα στον σχεδιασμό ιστοσελίδων. Η Java επιτρέπει σε έναν προγραμματιστή να επικαλεστεί μεθόδους και να γράψει αλγορίθμους για να ελέγξει την ροή μια παρουσίασης στον Ιστό. Η VRML/X3D επιτρέπει σε σχεδιαστές γραφικών να σχεδιάσουν αντικείμενα που ζουν σε έναν 3D κόσμο, και να γράψουν κώδικα απλής συμπεριφοράς για αυτά τα αντικείμενα. Για πιο πολύπλοκες λειτουργίες συμπεριφοράς, τυπικά συμπεριφορά που εξαρτάται σε εξωτερικά δεδομένα, χρησιμοποιείται η Java.

### 2.8.2.2 Συνδυάζοντας Java και VRML/X3D

Η Java μπορεί να χρησιμοποιηθεί με VRML/X3D με δυο τρόπους. Η VRML/X3D διαθέτει έναν κόμβο Script, με βάση τον οποίο μπορεί να χρησιμοποιηθεί η Java σαν γλώσσα scripting για τον κόμβο αυτόν. Το java script αλληλεπιδρά με τον VRML/X3D κόσμο μέσα από το Java Script Authoring Interface (JSAI), το οποίο επιτρέπει στην Java να στείλει γεγονότα σε άλλους κόμβους στον VRML/X3D κόσμο, να δημιουργήσει νέα τμήματα στην σκηνή και να επερωτήσει για

πληροφορίες σχετικά με την σκηνή. Το Java script λαμβάνει επίσης γεγονότα από άλλους κόμβους, τα οποία και το θέτουν σε εκτέλεση, όπου μπορεί να εκτελέσει αλγορίθμους ή να χρησιμοποιήσει άλλα πακέτα της Java (όπως το πακέτο Δικτύων). Το script μπορεί έπειτα να στείλει τα αποτελέσματα αυτής της εκτέλεσης σε άλλους κόμβους σαν γεγονότα. Ένα καλό παράδειγμα του παραπάνω σε μια σκηνή στην οποία η Java εκτελεί πολύπλοκους φυσικούς υπολογισμούς είναι ο έλεγχος του κυματισμού μιας σημαίας ή το χτύπημα μιας μπάλας. Η VRML/X3D έχει έναν πρωτότυπο μηχανισμό ο οποίος επιτρέπει σε κόμβους της σκηνής και στο Java script που τους ελέγχει να συναρμολογηθούν εννιαία και να συντεθούν σε άλλες σκηνές.

Η Java επίσης μπορεί να ελέγξει έναν VRML/X3D κόσμο εξωτερικά, χρησιμοποιώντας το External Authoring Interface (EAI). Το EAI επιτρέπει σε έναν ή περισσότερους X3D κόσμους να προστεθούν σε ένα Java applet ή σε μια εφαρμογή. Αυτό επιτρέπει στην X3D να είναι μέρος μιας μεγαλύτερης παρουσίασης πολυμέσων στον Ιστό. Το EAI επίσης επιτρέπει σε μεγάλο βαθμό την ίδια λειτουργικότητα με το JSAI, δηλαδή ο χρήστης μπορεί να αποστείλει γεγονότα σε X3D κόμβους, να φτιάξει νέους κόμβους, και να επερωτήσει τον X3D κόσμο σχετικά με την κατάστασή του και τα χαρακτηριστικά του. Αλλά, αντί η εκτέλεση να πυροδοτηθεί από κάποιο γεγονός εισερχόμενο σε έναν script κόμβο, το EAI επιτρέπει σε ένα Java applet που τρέχει εκείνη τη στιγμή να ελέγξει τον X3D κόσμο, όπως θα ήλεγχε οποιοδήποτε άλλο μέσο. Για παράδειγμα, πατώντας ένα κουμπί από το πακέτο AWT της Java θα μπορούσε ο χρήστης να προκαλέσει την αποστολή ενός γεγονότος στον X3D κόσμο το οποίο θα άλλαζε το χρώμα μιας σφαίρας, ή το ύψος μιας μπάρας γραφήματος, ή την τιμή μιας μετοχής.

Η Java έχει γίνει ένας πολύ δημοφιλής και ισχυρός τρόπος για να ελέγξουμε την παρουσίαση μιας Ιστοσελίδας. Η X3D παρέχει στους σχεδιαστές ένα απλό και δυνατό τρόπο για να προσθέσουν 3D περιεχόμενο σε αυτή την Ιστοσελίδα.

### 2.8.3 Το πακέτο Xj3D

Το πακέτο Xj3D είναι μια δημιουργία του Web3D Consortium Source Working Group, η οποία στοχεύει στην παροχή ενός εύχρηστου πακέτου εργαλείων για την επεξεργασία VRML και X3D υλικού, γραμμένου απευθείας σε Java (25). Εξυπηρετεί τον διπλό στόχο του να αποτελεί μια πειραματική βάση κώδικα για την εξερεύνηση νέων περιοχών ερευνητικού ενδιαφέροντος για το X3D, αλλά και σαν βιβλιοθήκη που ο προγραμματιστής εφαρμογών μπορεί να χρησιμοποιήσει για να γράψει την δικιά του εφαρμογή που να υποστηρίζει την τεχνολογία X3D. Πρόκειται για ένα πακέτο με υψηλό βαθμό τμηματοποίησης που επιτρέπει στον προγραμματιστή να χρησιμοποιεί κάθε φορά μόνο τα τμήματα του πακέτου που χρειάζεται για να αναπτύξει λογισμικό.

Μπορεί να χρησιμοποιηθεί είτε για την απευθείας εισαγωγή X3D υλικού σε κάποια εφαρμογή, είτε για την ανάπτυξη ενός πλήρους-λειτουργικότητας browser για X3D. Η παρούσα επίσημη έκδοση του πακέτου είναι η 1.0, ενώ ανεπίσημα έχει ήδη κάνει την εμφάνισή της η 2.0 με πολλά από τα προβλήματα της προηγούμενης διορθωμένα.

Το πακέτο Xj3D (συγκεκριμένα η έκδοση 2.0) χρησιμοποιήθηκε για την ανάπτυξη του editor HAniME της παρούσης διπλωματικής εργασίας.

## 2.9 ΣΥΜΠΕΡΑΣΜΑΤΑ

Στο Κεφάλαιο αυτό παρουσιάστηκε αναλυτικά η γλώσσα VRML και η εξέλιξή της X3D. Επίσης παρουσιάστηκαν βασικές έννοιες σχετικές με την Εικονική Πραγματικότητα, την διασώληνωση των γραφικών και τις μηχανές προβολής του, τους φυλλομετρητές για VRML/X3D και την υποστήριξη της επεξεργασίας VRML περιεχομένου από τεχνολογίες σχετικές με την γλώσσα Java.



## ΚΕΦΑΛΑΙΟ 3: AVATARS



## AVATARS

Σε αυτό το κεφάλαιο παρουσιάζουμε το απαραίτητο θεωρητικό υπόβαθρο για την χρήση των avatars σε συστήματα Εικονικής Πραγματικότητας χωρίς να μπαίνουμε σε τεχνικές λεπτομέρειες για το πώς υλοποιούνται. Στο επόμενο κεφάλαιο αναφέρονται τα τεχνικά χαρακτηριστικά του προτύπου H-Anim για την υποστήριξη των αναπαραστάσεων avatars σε περιβάλλοντα Εικονικής Πραγματικότητας.

### 3.1 Γενικά περί Avatars – Εφαρμογές

Η λέξη avatar κατάγεται από την αντίστοιχη σανσκριτική (26), της οποίας το νόημα ήταν στο περίπου «μια μορφή του εαυτού». Ο λόγος για τον οποίο υιοθετήθηκε η συγκεκριμένη λέξη είναι ότι αποδίδει πολύ εύστοχα αυτό για το οποίο χρησιμοποιείται σε συστήματα Εικονικής Πραγματικότητας: πρόκειται για την αναπαράσταση ενός χρήστη μέσα σε ένα περιβάλλον, είτε στην μορφή ενός τρισδιάστατου μοντέλου, είτε σαν μια δισδιάστατη εικόνα όπως χρησιμοποιείται σε Internet forums ή άλλες κοινότητες. Πρόκειται για ένα «αντικείμενο» που αναπαριστά την παρουσία ενός συνδεδεμένου χρήστη. Θα επικεντρωθούμε στα avatars που αναπαρίστανται σαν τρισδιάστατα μοντέλα.



Εικόνα 16: Τρισδιάστατα Avatars

#### 3.1.1 Χρήση των Avatars στην καθημερινότητα

Με βάση το πόσο χαλαρώνει κάποιος τα πλαίσια ορισμού της ψηφιακής αναπαράστασης, μπορεί να διατυπωθεί το επιχείρημα ότι τα avatars είναι κάτι πολύ συνηθισμένο και καθημερινό στην κοινωνία. Για παράδειγμα, ο ήχος μεταμορφώνεται σε ψηφιακή πληροφορία καθώς μεταφέρεται μέσα από τα καλώδια οπτικών ινών και τα κινητά δίκτυα επικοινωνιών. Κατά συνέπεια, η ηχητική

αναπαράσταση που αντιλαμβανόμαστε πάνω από τις τηλεφωνικές γραμμές είναι στην πραγματικότητα ένα avatar του ομιλητή. Αυτό το παράδειγμα μπορεί να φανεί τετριμμένο αρχικά, αλλά γίνεται λιγότερο τετριμμένο όταν πάνω στην ροή του ήχου εφαρμόζονται προκαθορισμένοι αλγόριθμοι που αλλοιώνουν την αναπαράσταση, για παράδειγμα για να «καθαρίσουν» και να ενισχύσουν το σήμα. Αυτό μπορεί να γίνει αποτελεσματικά καθώς η φωνή μεταφράζεται σε ψηφιακή πληροφορία.

Η συνηθέστερη περίπτωση ωστόσο αναφέρεται στην περίπτωση των οπτικών αναπαραστάσεων των avatars. Εκατομμύρια ανθρώπων χρησιμοποιούν avatars σε on-line παιχνίδια ρόλων καθώς και σε chat-rooms τα οποία χρησιμοποιούνται για βιντεοδιάσκεψη. Σε αυτά τα περιβάλλοντα, οι χρήστες αλληλεπιδρούν ο ένας με τον άλλο χρησιμοποιώντας πληκτρολόγιο ή joystick, πληκτρολογώντας μηνύματα ο ένας προς τον άλλο, αναμένοντας απάντηση και παρατηρώντας τα avatars άλλων χρηστών καθώς περιηγούνται στον ψηφιακό κόσμο. Τυπικά, και αυτά είναι avatars με την ευρεία έννοια του όρου, καθώς σε αυτές τις περιπτώσεις ο φωτογραφικός ρεαλισμός είναι συνήθως χαμηλός. Στην περίπτωση των on-line παιχνιδιών ρόλων, οι χρήστες συνήθως περιηγούνται στον εικονικό κόσμο χρησιμοποιώντας «έτοιμα» avatars που διατίθενται από το περιβάλλον, με περιορισμένες συμπεριφορικές δυνατότητες.

### 3.1.1.1 Avatars σε Computer Games

Χρησιμοποιούμενα σαν αναπαράσταση του χρήστη στον υπολογιστή, ο όρος μας πηγαίνει πίσω στο 1985 όταν πρωτοχρησιμοποιήθηκε στην βιομηχανία των Computer Games. Σιγά σιγά ο όρος καθιερώθηκε και έκτοτε χρησιμοποιείται αποκλειστικά για την αναπαράσταση του χρήστη, η οποία θα μπορούσε είτε να είναι προκαθορισμένη είτε να μπορεί τροποποιηθεί ως προς τα εμφανισιακά της χαρακτηριστικά ανάλογα με τις προδιαγραφές του παιχνιδιού.

Πολλά σύγχρονα video games, εκμεταλλευόμενα τις αυξημένες δυνατότητες της τεχνολογίας για υποστήριξη των 3D γραφικών, έχουν ανεβάσει τον πήχη ως προς τον βαθμό ρεαλισμού που επιδιώκουν για τα avatars του χρήστη. Στις περισσότερες περιπτώσεις, το video game επιτρέπει την χρήση κάποιου βασικού προτύπου και την επεξεργασία του όπως ο παίκτης κρίνει σκόπιμο. Για παράδειγμα, παιχνίδια όπως τα Sims που προσομοιώνουν την ανθρώπινη ζωή, επιτρέπουν στον χρήστη να διαλέξει φύλο, φυσικά και ηλικιακά χαρακτηριστικά, ρουχισμό, ταττού, αξεσουάρ κλπ για το avatar του, ενώ από τις επιλογές του χρήστη μέσα στο ίδιο το παιχνίδι επηρεάζεται ρεαλιστικά και το ίδιο το avatar, πχ. καθώς τα χρόνια παίρνουν μέσα στο παιχνίδι το avatar παρουσιάζει σημάδια ωρίμανσης/γήρανσης κλπ.

Εκτός από την φυσική εμφάνιση του άβαταρ, ενδιαφέρον παρουσιάζει και ο διάλογος που το συνοδεύει σε διάφορες φάσεις του παιχνιδιού, προσδίδοντας μεγαλύτερο βαθμό ρεαλισμού και αποκαλύπτοντας περισσότερα για τον χαρακτήρα που επιχειρεί ο δημιουργός του παιχνιδιού ή ο παίκτης να προσδώσει στο avatar.

### 3.1.1.2 Avatars σε περιβάλλοντα που δεν χρησιμοποιούνται για gaming

Τα avatars σε περιβάλλοντα που δεν χρησιμοποιούνται για ψυχαγωγία χρησιμοποιούνται επίσης σαν δισδιάστατες ή τρισδιάστατες ανθρώπινες ή και φανταστικές αναπαραστάσεις του χρήστη. Τέτοιες αναπαραστάσεις μπορούν να εξερευνήσουν το εικονικό σύμπαν με το οποίο αλληλεπιδρούν, προσθέτουν σε αυτό και συνομιλούν με άλλους χρήστες. Συνήθως, ο στόχος τέτοιων περιβαλλόντων είναι να παρέχουν μεγαλύτερη αίσθηση ρεαλισμού και βελτιώσεις στις καθιερωμένες

δυνατότητες συνομιλίας στο διαδίκτυο, και να επιτρέπουν στον χρήστη να αλληλεπιδρά με το περιβάλλον χωρίς απαραίτητα να υπάρχει κάποιος προκαθορισμένος στόχος.

Τα κριτήρια που πρέπει να πλοιοούν τα avatars σε τέτοια περιβάλλοντα ώστε να είναι χρήσιμα μπορεί να ποικίλλουν ανάλογα με την εφαρμογή, πχ. η ηλικία των πιθανών χρηστών. Η έρευνα προτείνει ότι οι νεότεροι χρήστες των εικονικών κοινοτήτων δίνουν μεγάλη έμφαση στην διασκεδαστική και ψυχαγωγική πλευρά των avatars, καθώς και στην πρακτικότητα των λειτουργιών τους (πχ ψυθίρισμα). Οι νεότεροι χρήστες περαιτέρω ενδιαφέρονται στην εύκολη χρήση των avatars, και στην ικανότητα να διατηρήσουν την ανωνυμία του χρήστη. Μεγαλύτεροι ηλικιακά χρήστες δίνουν έμφαση στην πειστική αναπαράσταση της δικιάς τους εμφάνισης, ταυτότητας και προσωπικότητας. Επιπρόσθετα, η πλειοψηφία των μεγαλύτερων ηλικιακά χρηστών θέλει να είναι σε θέση να χρησιμοποιήσει τις λειτουργικότητες ενός εκφραστικού avatar (πχ, την ικανότητα να δείχνει συναισθήματα), ενώ είναι σε θέση να μάθουν νέες μεθόδους πλοήγησης ώστε να μπορούν να χειριστούν όλο και πιο πολύπλοκα avatars.

Ενδιαφέρον είναι το παράδειγμα του Second Life (27), ενός εικονικού κόσμου που είναι προσβάσιμος μέσω του διαδικτύου. Η πρόσβαση στον εικονικό κόσμο γίνεται με χρήση ενός ελεύθερου προγράμματος-πελάτη (client) ο οποίος αποκαλείται Second Life Viewer, και επιτρέπει στους χρήστες, τους αποκαλούμενους «κατοίκους» να αλληλεπιδρούν μεταξύ τους με την χρήση avatars. Οι κάτοικοι μπορούν να εξερευνούν, να συναντούν άλλους κατοίκους, να συμμετέχουν σε ατομικές ή ομαδικές δραστηριότητες και να εμπορεύονται την εικονική τους περιουσία. Το Second Life, το οποίο απευθύνεται στο ενήλικο κοινό, έχει ως στόχο να παρέχει μια εικονική «δεύτερη ζωή» στους χρήστες, παράλληλα με την ζωή τους στον πραγματικό κόσμο. Μια παραλλαγή του Second Life, το Teen Second Life, απευθύνεται σε άτομα μικρότερης ηλικίας.

Όπως είναι αναμενόμενο, μεταξύ χρήστη και avatar δημιουργείται μια ιδιαίτερη σχέση η οποία θα μπορούσε να χαρακτηριστεί ως «ψυχολογικό δέσιμο», ειδικά όταν ο χρήστης χρησιμοποιεί συχνά το ίδιο avatar και για πολλαπλές εφαρμογές.

### 3.1.2 Ψηφιακά Μοντέλα

Για τον σκοπό της αναπαράστασης έχουν προταθεί ψηφιακά μοντέλα (28), τα οποία χρησιμοποιούν τις τελευταίες τεχνολογίες για παραγόμενη-από-υπολογιστή εικόνα, αλλά συνήθως τέτοια ψηφιακά μοντέλα δεν χαρακτηρίζονται ως avatars, καθώς χρησιμοποιούνται σαν να είχαν την δική τους προσωπικότητα.

Πολλά ψηφιακά μοντέλα που χρησιμοποιούν τεχνολογίες για παραγωγή εικόνας από τον υπολογιστή χρησιμοποιούνται στο ηλεκτρονικό εμπόριο για εφαρμογές που σχετίζονται με την μόδα, με βάση το σχήμα, μέγεθος, και ύψος του σώματος του πελάτη για να παρέχουν μια ιδέα του πως θα φαίνεται το τελικό προϊόν, ή για να δώσουν μια γενική ιδέα των αποτελεσμάτων της φυσικής άσκησης και κάποιου προγράμματος διαίτας. Πρόκειται για σχετικά στατικά αλλά πλήρως αλληλεπιδραστικά avatars των χρηστών τους.

### 3.1.3 *Μελλοντική Εξέλιξη και παράγοντες*

Τα avatars μπορούν να χρησιμοποιηθούν σε μια ποικιλία εφαρμογών όπου εμπλέκεται η επικοινωνία μεταξύ χρηστών, όπως η τρισδιάστατη αναπαράσταση μιας θεατρικής αναπαράστασης, μια εικονική τάξη για εφαρμογές e-learning ή η εικονική και ρεαλιστική αναπαράσταση διάσκεψης. Καθώς οι δυνατότητες των υπολογιστών για ρεαλιστική αναπαράσταση 3D περιεχομένου σε πραγματικό χρόνο (real time rendering) θα αυξάνουν προσεγγίζοντας το ιδανικό, στο μέλλον αναμένεται οι εφαρμογές αυτές να γνωρίσουν ιδιαίτερη άνθηση. Ιδιαίτερα σημαντικό ρόλο αναμένεται να παίζει και η καθιέρωση των ευρυζωνικών τεχνολογιών οι οποίες επιτρέπουν την μεταφορά μεγάλου όγκου δεδομένων (που συνήθως απαιτείται για την ρεαλιστική αναπαράσταση 3D εικονικών κόσμων) στο μικρό χρονικό διάστημα που είναι διατεθειμένος ο χρήστης να περιμένει καθώς και των φυσικών μέσων που χρησιμοποιούν αυτές οι τεχνολογίες (οπτικές ίνες).

### 3.2 *Τρισδιάστατα Avatars και Εικονικός Κόσμος*

Τα τρισδιάστατα avatars επιτρέπουν στον χρήστη να προβάλει μέσα στον εικονικό κόσμο ένα μεγάλο ποσοστό της δικιάς του προσωπικότητας χρησιμοποιώντας την εμφάνιση του avatar που επέλεξε να τον αντιπροσωπεύει, ενώ παραμένει ο ίδιος ανώνυμος. Η πλειοψηφία των 3D avatars που χρησιμοποιούνται είναι ανθρωποειδή στην μορφή και πολλά επιτρέπουν χειρονομίες και εκφράσεις προσώπου.

Η ανωνυμία που διατηρούν οι χρήστες με την χρήση ενός avatar βοηθάει στο άνοιγμα καναλιών επικοινωνίας, ενθαρρύνει τους χρήστες στο να εκφράζονται πιο ελεύθερα, ενώ αφαιρεί κοινωνικά κολλήματα. Κατ' αυτό τον τρόπο, τα avatars βοηθούν στον καλύτερο διαμοιρασμό της πληροφορίας. Με την ελευθερία επιλογής τόσο στην αναπαράσταση όσο και στην ανωνυμία, οι χρήστες αποκτούν μια πιο άνετη εκδοχή του εαυτού τους, η οποία μπορεί να τους βοηθήσει στο να αυξήσουν την αυτοπεποίθηση που επιδεικνύουν όταν επικοινωνούν με άλλους. Σύμφωνα με τους Brown και Bell (2004), η ανωνυμία ενθαρρύνει την επικοινωνία με ξένους, πράγμα που δεν συναντούμε στον φυσικό κόσμο. Τα avatars επίσης μπορούν να χρησιμοποιηθούν από επιχειρήσεις και μεγάλες εταιρίες να διεξάγουν χρήσιμες συναντήσεις.

#### 3.2.1 *Προοπτικές Πρώτου και Τρίτου Προσώπου*

Εκτός του ότι είναι πιο ελκυστικά από αισθητικής απόψεως στον χρήστη, τα 3D avatars εμβυθίζουν αποτελεσματικότερα τον χρήστη στο περιβάλλον καθώς ο χρήστης έχει επιλογή στην οπτική γωνία που βλέπει τον κόσμο: μια *πρώτου-* ή *τρίτου-προσώπου* οπτική αναπαράστασης (29) του εικονικού κόσμου.



**Εικόνα 17:** Προοπτική Πρώτου και Τρίτου Προσώπου

Αυτά τα avatars όχι μόνο αναπαριστούν την παρουσία του χρήστη στον εικονικό κόσμο, αλλά επίσης προβάλλουν και τον προσανατολισμό και την τοποθεσία του χρήστη. Οι χρήστες μπορούν να αλληλεπιδρούν με αντικείμενα ή άλλα avatars μέσα στον εικονικό κόσμο με παρόμοιο τρόπο με αυτό που συμβαίνει στον πραγματικό κόσμο. Μερικοί 3D εικονικοί κόσμοι είναι ενισχυμένοι με 3D ήχο (με απόσβεση στην απόσταση και στερεοφωνική τοποθέτηση) για να παρέχουν ανάδραση ως προς την σχετική τοποθέτηση των άλλων συμμετεχόντων και των στοιχείων μέσα στον εικονικό κόσμο. Κατ' αυτόν τον τρόπο, τα avatars δεν έχουν μόνο την προοπτική να ενισχύουν τις εικονικές κοινότητες, αλλά μπορούν επίσης να ενθαρρύνουν όλους τους χρήστες να παραμένουν στον εικονικό κόσμο για μεγάλα χρονικά διαστήματα.

### 3.2.2 *Λειτουργίες των Avatars*

Οι βασικές απαιτήσεις για ένα avatar σε ένα τρισδιάστατο Εικονικό Περιβάλλον που υποστηρίζει πολλαπλούς χρήστες συνοψίζονται ως εξής (30):

- *Αντίληψη:* Θα πρέπει η οθόνη κάθε χρήστη να ενημερώνεται αυτόματα και σε πραγματικό χρόνο για τυχόν αλλαγές που προκαλούνται από άλλους χρήστες. Σημαντικό είναι η λειτουργία αυτή να παρέχεται χωρίς διακοπές ή ελλείψεις καθ' όλη τη διάρκεια παραμονής των χρηστών στο Εικονικό Περιβάλλον.
- *Εντοπισμός:* Θα πρέπει κάθε χρήστης να είναι σε θέση να εντοπίζει τις πραγματικές θέσεις των άλλων συμμετεχόντων και του προσανατολισμού τους στο χώρο.
- *Αναγνώριση:* Θα πρέπει ο κάθε συμμετέχοντας στον Εικονικό Κόσμο να είναι σε θέση να αναγνωρίζει τους υπόλοιπους συμμετέχοντες. Στην ιδεατή περίπτωση, ο κάθε χρήστης θα πρέπει να έχει τη δική του μοναδική αναπαράσταση.
- *Αντίληψη της Προσοχής των άλλων Χρηστών:* ο κάθε συμμετέχοντας θα πρέπει να είναι σε θέση να καταλάβει που εστιάζει ή που είναι στραμμένη η προσοχή οποιουδήποτε άλλου χρήστη.
- *Παρακολούθηση Ενεργειών:* Κάθε χρήστης πρέπει να μπορεί να παρακολουθεί με σαφήνεια τις κινήσεις και τις ενέργειες των άλλων χρηστών.
- *Εμφάνιση:* Η εμφάνιση του avatar του χρήστη θα πρέπει να αλλάζει ανάλογα με τις ενέργειες του χρήστη στον εικονικό κόσμο, πχ. αν το avatar του χρήστη προσαρτήσει πάνω του κάποιο αξεσουάρ, θα πρέπει αυτό να εμφανίζεται ως μέρος του avatar.

### 3.2.3 Κατηγορίες Avatars – Agents

Μια κατηγοριοποίηση των avatars μπορεί να γίνει με βάση τον έλεγχο που έχει πάνω σε αυτά ο χρήστης (31). Έτσι μπορούμε να διακρίνουμε τις εξής κατηγορίες:

- *Απευθείας ελεγχόμενα Avatars:* Το avatar μετακινείται και αλληλεπιδρά με τον κόσμο με απευθείας εντολές του χρήστη με τη βοήθεια αισθητήρων που τοποθετεί στο σώμα του ο χρήστης .
- *Avatars κατευθυνόμενα από το χρήστη:* Ο χρήστης κατευθύνει το avatar μέσα στον εικονικό κόσμο με τη χρήση συσκευών εισόδου και αυτός δίνει τις εντολές για τις ενέργειες που θα εκτελέσει η ψηφιακή του αναπαράσταση στο εικονικό περιβάλλον. Ο συγκεκριμένος τύπος συναντάται ευρύτατα σε επιτραπέζια (desktop) συστήματα Δικτυακής Εικονικής Πραγματικότητας.
- *Αυτόνομα Avatars:* Το avatar δεν λειτουργεί με βάση τον έλεγχο κάποιου χρήστη, αλλά το ίδιο το Εικονικό Περιβάλλον αναλαμβάνει να το λειτουργήσει με βάση το πώς λειτουργούν οι χρήστες και την ατμόσφαιρα που θέλει να προσδώσει στο περιβάλλον. Στην πραγματικότητα το παραπάνω δύσκολα θεωρείται avatar, καθώς ο όρος υπονοεί την ύπαρξη κάποιου φυσικού προσώπου το οποίο αναπαρίσταται στην οθόνη από κάποιο μοντέλο. Τέτοιου είδους avatars ονομάζονται αυτόνομοι agents.



**Εικόνα 18:** Ένας Agent ξεναγεί τον χρήστη στους χώρους του Εικονικού Περιβάλλοντος

### 3.3 Αλληλεπίδραση Avatars σε Εικονικούς Κόσμους

Ένας χρήστης που χρησιμοποιεί ένα 3D avatar στα πλαίσια ενός πειστικού εικονικού κόσμου θα πρέπει να είναι σε θέση να αλληλεπιδράσει τόσο με αντικείμενα του εικονικού περιβάλλοντος όσο και με τις αναπαραστάσεις των άλλων χρηστών. Αυτό μπορεί να επιτευχθεί με κάποια από τις παρακάτω μεθόδους (32):

- **Επικοινωνία χαρακτηριστικών του προσώπου (facial communication)**
- **Χειρονομία (gestures)**
- **Αλληλεπίδραση του avatar με αντικείμενα του χώρου**
- **Κίνηση του avatar στο χώρο (animation)**



### 3.3.1 *Επικοινωνία χαρακτηριστικών του προσώπου (facial communication)*

Τα avatars έχουν γίνει εξαιρετικά δημοφιλή, ωστόσο η επικοινωνία μέσω των χαρακτηριστικών του προσώπου είναι συνήθως περιορισμένη. Αυτό συμβαίνει εν μέρει χάρη στο γεγονός ότι πολλές ερωτήσεις, ειδικά σχετικά με την δυναμική των εκφράσεων του προσώπου, είναι ακόμα ανοικτές. Επιπρόσθετα, τα λίγα εμπορικά εργαλεία μορφοποίησης κίνησης για το πρόσωπο έχουν περιορισμένες εφαρμογές, και δεν στοχεύουν στις «ελαφριές» εφαρμογές διαδικτύου.

Τα ανθρώπινα πρόσωπα μεταφέρουν πολλαπλές πληροφορίες στην καθημερινή επικοινωνία, ανεξάρτητα από τον προφορικό λόγο. Πολλές πληροφορίες, (πχ η συναισθηματική κατάσταση) μπορούν να διαβαστούν, σχεδόν αποκλειστικά, από το πρόσωπο, και σε πολλές περιπτώσεις το πρόσωπο παρέχει βοηθητικές πληροφορίες για να κατανοήσουμε την επικοινωνία μέσα από άλλα κανάλια (πχ, η παρακολούθηση του στόματος βοηθάει στην κατανόηση του λόγου). Το πρόσωπο επίσης χαρακτηρίζει και προσδίδει ταυτότητα για την αναγνώριση των ανθρώπων.

Τα avatars έχουν γίνει η λύση για την παροχή στον χρήστη μιας ανθρωποειδούς αναπαράστασης του εαυτού του, των άλλων ή/και κάποιων βοηθών του συστήματος. Παρ'όλο που ένα avatar δεν χρειάζεται να φαίνεται, απαραίτητα, ρεαλιστικό, θα πρέπει να είναι σε θέση να αναπαραστήσει τρόπους επικοινωνίας που να είναι εύκολα αναγνωρίσιμοι από τους άλλους. Οι εκφράσεις του προσώπου (ή των συναισθημάτων, των γνωσιακών καταστάσεων, του λόγου) είναι μεγάλης σπουδαιότητας, τόσο για να βελτιώσουν την αποδοτικότητα της αλληλεπίδρασης και να κάνουν τον χρήστη να νιώθει άνετα όταν χρησιμοποιεί ένα σύστημα. Ωστόσο, καθώς οι άνθρωποι είναι πολύ εκπαιδευμένοι και έντονα κριτικοί όταν διαβάζουν πραγματικά πρόσωπα, είναι μεγάλη πρόκληση η δημιουργία avatars με «σωστές» εκφράσεις προσώπου.

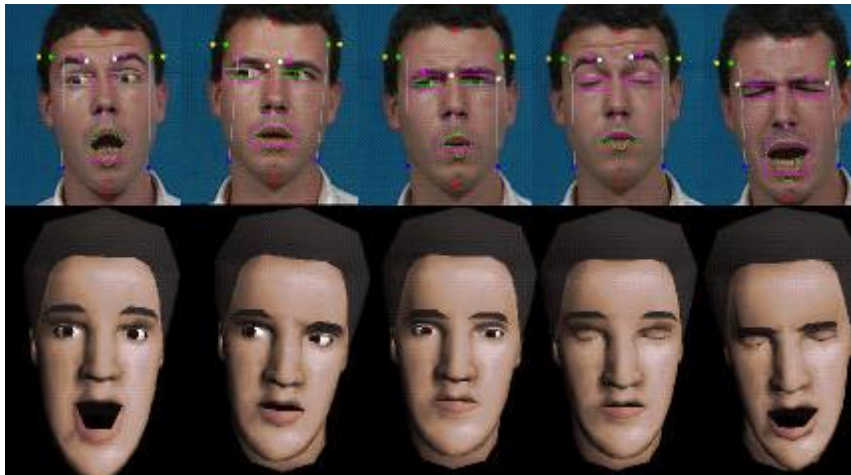
Διάφορες μέθοδοι που χρησιμοποιούνται:

1. Video Texturing of the face: Η τεχνική αυτή βασίζεται στην χρήση μιας ψηφιακής κάμερας (33), η οποίας καταγράφει την κίνηση του προσώπου του χρήστη στον πραγματικό χρόνο. Εν συνεχεία, χρησιμοποιούνται κάποιοι αλγόριθμοι ανάλυσης εικόνας, οι οποίοι αντιστοιχίζουν την κίνηση του προσώπου του χρήστη στον πραγματικό χρόνο στην κίνηση του προσώπου του avatar στον εικονικό κόσμο.



**Εικόνα 19:** Video Texturing του Προσώπου του Χρήστη

2. Model-based coding of facial expressions: Σε αντίθεση με την προηγούμενη μέθοδο όπου μεταφέρονται ολόκληρες εικόνες με τις εκφράσεις του προσώπου, σε αυτή την τεχνική οι εικόνες αναλύονται και από την ανάλυση εξάγεται ένα σύνολο παραμέτρων που περιγράφουν την έκφραση του προσώπου. Όπως και στην προηγούμενη μέθοδο, ο χρήστης θα πρέπει να είναι μπροστά από την ψηφιακή κάμερα η οποία και αναλαμβάνει να ψηφιοποιήσει τις βιντεοεικόνες. Η εξακρίβωση της ανάλυσης των εκφράσεων του προσώπου από τις αλληλουχίες εικόνων του βίντεο χρειάζεται λεπτομερή περιγραφή των στοιχείων των εκφράσεων του προσώπου, πράγμα που συχνά είναι υπολογιστικά ακριβό. Κάποιες ενδεικτικές παράμετροι είναι: η περιστροφή του κεφαλιού κάθετα, ή οριζόντια περιστροφή του κεφαλιού, το πλάγιασμα του κεφαλιού προς μια πλευρά, η κίνηση των οφθαλμών, η οριζόντια θέση της ίριδας του οφθαλμού, η απόσταση μεταξύ των φρυδιών, το ανασήκωμα των φρυδιών κατά το συνοφρύωμα, το διάνοιγμα του στόματος, η περιστροφή του σαγωνιού κλπ.



**Εικόνα 20:** Model-based coding εκφράσεων του προσώπου

3. Lip Movement Synthesis from Speech: Μπορεί και να μην είναι πάντα πρακτικό για τον χρήστη να βρίσκεται μπροστά από την κάμερα. Σε αυτή την περίπτωση, η επικοινωνία μέσω των εκφράσεων του προσώπου δεν χρειάζεται να εγκαταλειφθεί. Είναι δυνατό να εξάγουμε οπτικές παραμέτρους της κίνησης των χειλιών αναλύοντας το σήμα ήχου της ομιλίας. Μια εξαιρετικά απλή εκδοχή ενός συστήματος που αναλαμβάνει την παραπάνω δουλειά απλώς θα άνοιγε και θα έκλεινε το στόμα όταν υπάρχει ομιλία, αφήνοντας έτσι τους άλλους χρήστες να καταλάβουν ότι κάποιος μιλάει. Ένα πιο πολύπλοκο σύστημα θα ήταν σε θέση να συνθέσει μια ρεαλιστική κίνηση χειλιών, η οποία θα ήταν μια πολύ σημαντική βοήθεια για την κατανόηση της ομιλίας.
4. Predefined Expressions or Animation: Σε αυτή την εκδοχή επιλέγεται απλά ένα σύνολο από προκαθορισμένες εκφράσεις προσώπου ή κινήσεις. Η επιλογή μπορεί να γίνει από το πληκτρολόγιο από τον χρήστη με την χρήση smileys, παρόμοια με αυτά που στέλνει κάποιος με e-mail.



**Εικόνα 21:** Προκαθορισμένες Εκφράσεις Προσώπου – Έκπληξη, νύστα, βαρεμάρα

### 3.3.2 *Gesturing* – Χειρονομίες

Οι χειρονομίες παίζουν σημαντικό ρόλο στην ανθρώπινη επικοινωνία. Χρησιμοποιώντας το σώμα, πολλαπλά μηνύματα μπορούν να μεταδοθούν. Οι κινήσεις σώματος μπορούν να διαχωριστούν γενικά σε τρεις ομάδες:

- Στιγμαϊές χειρονομίες: τις περισσότερες φορές, ακόμη και ασυναίσθητα, συνοδεύουμε τα λόγια μας με κινήσεις. Δίνουν έμφαση στον λόγο και σε συγκεκριμένες λέξεις. Επίσης συχνά έχουν οι ίδιες νόημα χωρίς να χρειάζονται παραπάνω λέξεις. Ολόκληρη η στάση του σώματος επίσης μεταφέρει πληροφορία σχετικά με την κατάσταση του προσώπου και πιθανότατα κάποια συναισθήματα. Για παράδειγμα από την στάση του σώματος μπορούμε να πούμε αν ένα πρόσωπο είναι κουρασμένο, έχει ένταση ή είναι χαλαρό.
- Χειρονομίες-Εντολές: αυτές οι χειρονομίες κάνουν τον χρήστη να υποδείξει κάποια πράξη. Για παράδειγμα, η χειρονομία «ελα εδώ» μπορεί να αναπαρασταθεί από ανάταση του χειρός. Αυτές οι κινήσεις μπορεί να αλλάζουν από πρόσωπο σε πρόσωπο και από την μια κουλτούρα στην άλλη, κατά συνέπεια δεν υπάρχει κάποιο καλά καθορισμένο σύνολο από κανόνες για το νόημά τους.
- Γλώσσα που βασίζεται σε κανόνες: αυτές είναι χειρονομίες, για παράδειγμα που χρησιμοποιούνται από κωφούς και ακολουθούν κάποιο καλά βασισμένο σύνολο από κανόνες για την αναπαράσταση λέξεων και ήχων. Τα σημάδια τυπικά λειτουργούν σαν μεταφορές για τον προσδιορισμό άλλων αντικειμένων ή της γλώσσας. Οι χειρονομίες μπορούν να χρησιμοποιηθούν επίσης από το λογισμικό για να προσδιορίσουν ειδικές δουλειές (πχ, να δείξουν την εμπρόσθια κίνηση για το ξεκίνημα του περπατήματος).



**Εικόνα 22:** Χειρονομίες και στάσεις σώματος σε avatars που χορεύουν

Όλοι οι παραπάνω τύποι κινήσεων μπορούν να εξομοιωθούν με χρήση δυο μεθόδων: απευθείας εντοπισμό και προκαθορισμένες στάσεις σώματος ή κινήσεις. Αυτός ο τύπος ελέγχου μπορεί να είναι κατάλληλος για έναν συγκεκριμένο τύπο κινήσεων, ωστόσο, ένας συνδυασμός από αυτούς μπορεί να χρησιμοποιηθεί για διάφορα προβλήματα.

### 3.3.3 *Αλληλεπίδραση του Avatar με αντικείμενα του χώρου*

Η αλληλεπίδραση ενός Avatar με αντικείμενα του χώρου είναι θεμελιώδες χαρακτηριστικό ενός Εικονικού Περιβάλλοντος και απαντάται σε πολλά σενάρια (πχ σε μια εικονική τάξη). Θα πρέπει το avatar να είναι σε θέση να εξομοιώσει σε ρεαλιστικό βαθμό την αλληλεπίδραση, ενώ το παραπάνω θα πρέπει να μεταδίδεται ως εικόνα στην οθόνη των υπολοίπων χρηστών του εικονικού περιβάλλοντος.

### 3.3.4 *Κίνηση του Avatar στον Χώρο*

Παρόμοια με την απαίτηση το avatar να αλληλεπιδρά με τον Εικονικό Χώρο, θα πρέπει το avatar να είναι σε θέση να εκτελέσει μια σειρά από κινήσεις (animations) μέσα στον χώρο. Οι παραπάνω κινήσεις θα πρέπει να είναι αληθοφανείς ως προς την κίνηση των μελών του σώματος του avatar, και ως προς το που μπορεί να κινηθεί το avatar (πχ διέλευση μέσα από τοίχους). Η κίνηση θα πρέπει να προβάλλεται με ομοιόμορφο τρόπο σε όλους τους υπόλοιπους χρήστες όπως και στον ίδιο, ώστε να παρέχεται άμεση ανάδραση.

## 3.4 *Ρεαλισμός*

Τα avatars μπορούν να ομοιάζουν στο ανθρώπινο καλούπι τους σε μια σειρά από διαστάσεις, αλλά οι δυο διαστάσεις που έχουν λάβει την περισσότερη προσοχή στην βιβλιογραφία είναι ο συμπεριφορικός ρεαλισμός (34) (ο οποίος αντανακλάται σε μια σειρά από ανθρώπινες συμπεριφορές που επιδεικνύει το avatar) και ο φωτογραφικός ρεαλισμός (ο οποίος αντανακλάται στο πόσα από τα δοθέντα στατικά οπτικά χαρακτηριστικά του ανθρώπου επεξεργάζεται το avatar).

### 3.4.1 Συμπεριφορικός Ρεαλισμός

Ο συμπεριφορικός ρεαλισμός έγκειται στην δυνατότητα του συστήματος υλοποίησης να εντοπίσει και να εμφανίσει στην οθόνη συμπεριφορά σε πραγματικό χρόνο. Ως τώρα, η τεχνολογία συμπεριφορικού εντοπισμού πραγματικού χρόνου, ενώ βελτιώνεται σταδιακά, δεν ανταποκρίνεται στις προσδοκίες που έχει αποκτήσει ο κόσμος από την αντίστοιχη λογοτεχνία, πχ οι on-line αναπαραστάσεις του χαρακτήρα Neo στην ταινία *The Matrix* (1999), του Hiro από την *Snow Crash* (1992) ή του Case από την νουβέλα *Neuromancer* (1984). Σε αυτές τις δημιουργίες φαντασίας, τις κινήσεις και τις χειρονομίες των avatars και των αναπαριστώμενων ανθρώπων θεωρείται ότι οι αισθήσεις δεν μπορούν να τις ξεχωρίσουν. Ωστόσο, στην πράξη, ο πλήρης συμπεριφορικός εντοπισμός πραγματικού χρόνου είναι εξαιρετικά δύσκολος. Παρ' όλο που ο εντοπισμός χειρονομιών μέσα από διάφορα μηχανικά, οπτικά και άλλα μέσα έχει βελτιωθεί, το χάσμα μεταξύ των πραγματικών κινήσεων και των κινήσεων των avatars παραμένει αγεφύρωτο, μειώνοντας τον συμπεριφορικό ρεαλισμό τουλάχιστον σε περιπτώσεις όπου ο εντοπισμός και η παρουσίαση εικόνας στην οθόνη είναι πραγματικού-χρόνου, όπως πχ σε εφαρμογές κοινωνικής αλληλεπίδρασης, σε συνεργατικές ομάδες εικονικής εργασίας κλπ.

### 3.4.2 Φωτογραφικός Ρεαλισμός

Στον φωτογραφικό ρεαλισμό τα προβλήματα είναι λιγότερα. Οι τρισδιάστατοι σαρωτές και το φωτογραμμικό λογισμικό επιτρέπουν για την φωτογραφικά ρεαλιστική αναδημιουργία στατικών, ψηφιακών ανθρώπινων κεφαλιών και προσώπων που δεν μπορούν εύκολα να ξεχωριστούν από φωτογραφίες και videos που να περιλαμβάνουν τα αναπαριστώμενα πρόσωπα. Ωστόσο, η κυρίως πρόκληση για τους σχεδιαστές avatars είναι η δημιουργία προσώπων και σωμάτων σε αρκετή λεπτομέρεια ώστε να επιτρέπεται η ρεαλιστική παρουσίαση στην οθόνη της συμπεριφοράς, πράγμα που μας οδηγεί πάλι πίσω στον συμπεριφορικό ρεαλισμό.



**Εικόνα 23:** Δημιουργία ενός τρισδιάστατου mesh σε σχήμα κεφαλιού και επικάλυψή του με ένα φωτογραφικό texture που αποδίδει την εμφάνιση του ανθρώπου

Κλείνοντας, τα στατικά avatars μπορεί την τρέχουσα χρονική στιγμή να μοιάζουν κάπως με τους ανθρώπους που αναπαριστούν, αλλά μπορούν να εκτελέσουν μόνο ένα υποσύνολο των δυναμικών ανθρώπινων πράξεων στον πραγματικό χρόνο.



### 3.5 Προχωρημένα Θέματα

#### 3.5.1 Αλγόριθμοι TSI - Ηθικά Ζητήματα

Η επικοινωνία μέσω avatars επιτρέπει την απατηλή αλληλεπίδραση μεταξύ των χρηστών. Το 2003 η ερευνητική ομάδα του Bailenson εισήγαγε την έννοια των «μεταμορφωμένων» κοινωνικών αλληλεπιδράσεων (transformed social interactions – TSIs). Το να χρησιμοποιήσει κάποιος ένα avatar για αλληλεπίδραση με άλλο πρόσωπο διαφέρει από πολλές άλλες μορφές επικοινωνίας όπως η πρόσωπο-με-πρόσωπο, η τηλεφωνική ή η βιντεοδιάσκεψη. Ένα avatar το οποίο διαρκώς ανανεώνεται στην οθόνη σε πραγματικό χρόνο επιτρέπει στον χρήστη του να φιλτράρει συστηματικά την εμφάνιση ή και τις συμπεριφορές του (ή να αναθέσει στους διαχειριστές του συστήματος να το κανουν) μέσα σε εικονικά περιβάλλοντα, ενισχύοντας ή μειώνοντας τα σήματα επικοινωνίας.

Οι αλγόριθμοι TSI μπορούν να επιδράσουν πάνω στις δυνατότητες των αλληλεπιδρώντων για να επηρεάσουν τους τα πρόσωπα που επικοινωνούν. Για παράδειγμα, οι διαχειριστές συστήματος μπορούν να ταιριάζουν τις μη-λεκτικές συμπεριφορές ενός on-line δασκάλου σε περισσότερους από έναν χρήστη ταυτόχρονα σε μια εικονική τάξη πλήρους εμπύθισης με τρόπους που σχετίζονται με τον κάθε μαθητή ξεχωριστά και ταυτόχρονα. Ο μαθητής A μπορεί να αποδίδει καλύτερα σε έναν καθηγητή που χαμογελά ενώ διδάσκει, ενώ ο μαθητής B μπορεί να αποδίδει καλύτερα σε κάποιον με ουδέτερες εκφράσεις. Μέσω ενός avatar το οποίο ανανεώνεται ξεχωριστά για κάθε μαθητή, ο δάσκαλος μπορεί να αναπαρίσταται ταυτόχρονα από διαφορετικά avatars σε διαφορετικούς μαθητές, κατά συνέπεια επικοινωνώντας με τον κάθε μαθητή με τρόπο βέλτιστο για αυτόν. Ο ψυχολόγος Andrew Beall και οι συνεργάτες του χρησιμοποίησαν avatars για να υλοποιήσουν μια παρόμοια στρατηγική σχετικά με την επαφή με τα μάτια, αποδεικνύοντας ότι οι αλγόριθμοι του TSI που χρησιμοποιήθηκαν προκάλεσαν εντονότερα την προσοχή των μαθητών προς τον καθηγητή.

Ωστόσο, υπάρχουν ηθικά ζητήματα σχετικά με τέτοιους αλγορίθμους. Ο καθένας θα μπορούσε να φανταστεί το μέλλον της on-line επικοινωνίας να εξελίσσεται παρόμοια με έναν πύργο της βαβέλ, με τα avatars που αναπαριστούν τον ίδιο άνθρωπο να διαφέρουν τόσο πολύ από περίπτωση σε περίπτωση που να θολώνουν την βάση με την οποία κρίνεται η ειλικρίνεια και η ευθύτητα του χρήστη και της κοινωνικής αλληλεπίδρασης. Η έρευνα έδειξε ότι ο εντοπισμός αλγορίθμων TSI που να εμπλέκουν avatars είναι πολύ δύσκολος. Πρόκειται λοιπόν για μια πρόκληση των ερευνητών το να βρουν έναν βέλτιστο τρόπο να διαχειριστούν αυτό το θέμα καθώς η χρήση avatars γίνεται όλο και πιο διαδεδομένη.

#### 3.5.2 Η μεταμόρφωση της συμπεριφοράς

Υπάρχουν στην τρέχουσα χρονική περίοδο πολλά παραδείγματα ανθρώπων που αλληλεπιδρούν ο ένας με τον άλλο με την χρήση avatars. Ως επί το πλείστον, αυτά τα avatars είναι απλοικά και από συμπεριφορικής και φωτορεαλιστικής απόψεως μακριά από το τέλειο ή το ρεαλιστικό. Εξάιρεση ωστόσο αποτελούν οι περιπτώσεις των ερευνητικών εργαστηρίων, στα οποία οι επιστήμονες αναπτύσσουν και δοκιμάζουν avatars που είναι παρόμοια σε εμφάνιση και συμπεριφορά με τον άνθρωπο-χρήστη τους. Καθώς τα avatars γίνονται όλο και πιο καθημερινής χρήσεως (35), ίσως είναι πιθανό ότι θα δούμε πρωτοφανείς ποιοτικές αλλαγές στην κοινωνική αλληλεπίδραση

χάρη στην απόδοση και μεταμόρφωση της συμπεριφοράς από άνθρωπο σε avatar. Ενώ παρουσιάζονται ηθικοί κίνδυνοι στην μεταμόρφωση των συμπεριφορών καθώς παίρνουν από φυσικές πράξεις στις ψηφιακές τους αναπαραστάσεις, επίσης υπάρχουν θετικές ευκαιρίες τόσο για τους χρήστες on-line συστημάτων και για ερευνητές στην αλληλεπίδραση ανθρώπου-υπολογιστή.

### 3.5.3 Έρευνα πάνω στα Avatars

Οι επιστήμονες των Υπολογιστών, καθώς και επιστήμονες από άλλους τομείς, έχουν αφιερώσει πολύ κόπο και προσπάθεια για την ανάπτυξη συστημάτων ικανών να παράγουν λειτουργικά και αποτελεσματικά avatars. Έχουν επιδιώξει να δημιουργήσουν δυνατότητες γραφικής, λογικής και εντοπισμού (tracking) για να αναπαραστήσουν στην οθόνη πραγματικές κινήσεις από ανθρώπους σε ψηφιακά avatars με εκπληκτική ακρίβεια, και να βελτιώσουν αυτές τις κινήσεις χρησιμοποιώντας αλγορίθμους ελέγχου που συμπληρώνουν τα ελλειπή δεδομένα ελέγχου ή τις πληροφορίες σχετικά με στατικά οπτικά χαρακτηριστικά.

Επιπρόσθετα, οι επιστήμονες που ασχολούνται με τον συμπεριφορισμό έχουν αφιερώσει ένα μεγάλο μέρος της έρευνάς τους στην αλληλεπίδραση των ανθρώπων με την χρήση avatars. Αυτοί οι ερευνητές προσπαθούν να κατανοήσουν την κοινωνική παρουσία, ή αλλιώς την συμπαρουσία (copresence), έναν όρο που αναφέρεται στο βαθμό στον οποίο τα άτομα αντιδρούν κοινωνικά προς άλλους κατά την διάρκεια επικοινωνίας μεταξύ avatars, σε σύγκριση με τον βαθμό με τον οποίο αντιδρούν στους πραγματικούς ανθρώπους.

Ο συμπεριφορικός επιστήμονας Jim Blascovich και οι συνεργάτες του έχουν δημιουργήσει ένα θεωρητικό μοντέλο για την κοινωνική επιρροή μέσα στα εικονικά περιβάλλοντα εμπύθισης το οποίο παρέχει ακριβείς προβλέψεις σχετικά με το πως η αλληλεπίδραση του φωτογραφικού και συμπεριφορικού ρεαλισμού των avatars θα επηρεάσει την ανθρώπινη αίσθηση του πόσο σχετική είναι η συνάντησή τους με την χρήση avatars με την πραγματικότητα. Προτείνουν ότι η συμπερίληψη ορισμένων οπτικών χαρακτηριστικών είναι απαραίτητη αν το avatar χρειάζεται να εκτελέσει κάποια σημαντική, κοινωνικά συσχετίσιμη πράξη συμπεριφοράς (36). Για παράδειγμα, ένα avatar χρειάζεται να έχει αναγνωρίσιμα φρύδια ώστε να εξομοιωθεί πειστικά κάποια συμπεριφορά όπου το avatar συνοφρύνεται.

Αλλα δεδομένα δίνουν έμφαση στην σημασία του συμπεριφορικού ρεαλισμού. Το 2001 ο Jeremy Bailenson και οι συνεργάτες του έδειξαν ότι το να κάνει κάποιος μια ψηφιακή αναπαράσταση πιο φωτογραφικά ρεαλιστική δεν αυξάνει την κοινωνική παρουσία της σε σχέση με κάποιον πράκτορα ο οποίος μοιάζει με καρτούν, αρκεί και οι δυο χαρακτήρες πρακτόρων να επιδεικνύουν ρεαλιστικές συμπεριφορές ως προς την παρατήρηση. Σε ευρήματα που δημοσιεύτηκαν το 2003, η Maia Garau και οι συνεργάτες της απέτυχαν να επιδείξουν κάποιο γενικά αποδεκτό πλεονέκτημα για τα πιο φωτογραφικά ρεαλιστικά avatars. Επιπρόσθετα, αυτοί οι ερευνητές επέδειξαν ότι αυξάνοντας τον φωτογραφικό ρεαλισμό ενός avatar μπορεί και να έχει ως αποτέλεσμα να μειώνεται η κοινωνική παρουσία αν αυτή η αύξηση δεν συνοδεύεται και από κάποια βελτίωση στον συμπεριφορικό ρεαλισμό.

Κλείνοντας, παρ' όλο που η έρευνα πάνω στα avatars στην τρέχουσα φάση είναι εν πολλοίς στα πρώτα της βήματα, οι ερευνητές επεκτείνουν την κατανόησή μας σχετικά με την ανθρώπινη επικοινωνία διαμέσου υπολογιστών. Καθώς τα avatars γίνονται όλο

και πιο κοινά στην καθημερινότητα, η έρευνα που προσανατολίζεται στην περαιτέρω κατανόηση τέτοιων εφαρμογών θα επεκτείνεται.

### ***3.6 Συμπεράσματα***

Η τεχνολογία των avatars χρησιμοποιείται κατά κόρο για να υποστηρίξει τις εικονικές κοινότητες και αποκτά μεγάλη απήχηση καθώς καθιερώνεται. Προσφέρει πειστικές εμπειρίες στους χρήστες Εικονικών Περιβαλλόντων και ενθαρρύνει περισσότερους χρήστες να εισέρθουν στους εικονικούς κόσμους και να παραμείνουν σε αυτούς για μεγαλύτερα χρονικά διαστήματα. Όπως κάθε αναδυόμενη τεχνολογία, τα avatars ευεργετούν την ανθρωπότητα, ωστόσο φέρνουν μαζί τους κάποια αρνητικά στοιχεία, όπως ο εθισμός στην εικονική πραγματικότητα και η κατάχρηση της ανωνυμίας.



ΚΕΦΑΛΑΙΟ 4: ΤΟ ΠΡΟΤΥΠΟ H-  
ANIM



---

# ΤΟ ΠΡΟΤΥΠΟ H-ANIM

---

## 4.1 ΕΙΣΑΓΩΓΗ

Στο κεφάλαιο αυτό παρουσιάζουμε το πρότυπο H-Anim, για την επεξεργασία του οποίου αναπτύχθηκε ο editor HAnimE, τον οποίο και θα παρουσιάσουμε στο επόμενο κεφάλαιο. Εστιάζουμε στην παρουσίαση του H-Anim 2.0, της τελευταίας δηλαδή έκδοσης του H-Anim.

## 4.2 ΤΙ ΕΙΝΑΙ ΤΟ H-ANIM

Το πρότυπο **Humanoid Animation (H-Anim)** είναι ένα εγκεκριμένο από ISO πρότυπο για την μοντελοποίηση και την προσομοίωση κίνησης ανθρωποειδών, το οποίο ορίζει μια προτυποποίηση (ένα σύνολο κόμβων σε VRML) για τον καθορισμό ανθρώπινων τρισδιάστατων μορφών οι οποίες να μπορούν να χρησιμοποιηθούν σε μια ποικιλία από 3D εφαρμογές, παιχνίδια και περιβάλλοντα προσομοίωσης. Το πρότυπο αναπτύχθηκε στα τέλη της δεκαετίας του '90 και επηρεάστηκε σημαντικά από το Jack human modeling system και τα ερευνητικά πειράματα στην Γραφική, την Εργονομία, την Προσομοίωση και την Βιομηχανία Παιχνιδιών.

### 4.2.1 *Υπόβαθρο του Προτύπου*

Με το ολοένα και αυξανόμενο ενδιαφέρον στην 3D Γραφική κατά την διάρκεια της προηγούμενης δεκαετίας, εξίσου μεγάλη ανάπτυξη γνώρισε το λογισμικό μοντελοποίησης χαρακτήρων, για την δημιουργία και την προσομοίωση κίνησης ανθρώπινων 3D φιγούρων. Κατά την διάρκεια της ίδιας περιόδου, ένας αριθμός από συστήματα αναπτύχθηκαν για την χαρτογράφηση των κινήσεων ενός ανθρώπου στον «πραγματικό» κόσμο. Το κυρίαρχο εμπόδιο που συναντήθηκε με την χρήση τέτοιων πακέτων λογισμικού και συστημάτων εντοπίζεται στο πεδίο της ανταλλαγής πληροφορίας. Η έλλειψη ενός προτύπου συστήματος που να προσομοιώνει τον ανθρώπινο σκελετό συχνά εξαναγκάζει τα studio που ασχολούνται με την εικόνα και την κίνηση να αναπτύξουν τις δικές τους πρωτότυπες λύσεις για να βοηθήσουν στην εξομάλυνση μεταβάσεων μεταξύ συστημάτων και λογισμικού που θέλουν να χρησιμοποιήσουν.

Το Διεθνές Πρότυπο του H-Anim παρουσιάζει μια γενική αναπαράσταση για την μοντελοποίηση τρισδιάστατων ανθρώπινων φιγούρων. Επίσης περιγράφει έναν καθιερωμένο τρόπο αναπαράστασης ανθρωποειδών ο οποίος, εφόσον ακολουθείται, θα επιτρέπει σε ανθρώπινες φιγούρες που έχουν δημιουργηθεί με εργαλεία μοντελοποίησης από έναν κατασκευαστή, να υποστούν επεξεργασία και προσομοίωση κίνησης από κάποιον άλλο κατασκευαστή με το δικό του λογισμικό.

### 4.2.2 *Σχεδιαστικοί Στόχοι*

Τρεις θεμελιώδεις στόχοι τέθηκαν κατά τον σχεδιασμό και την επέκταση του προτύπου H-Anim. Αυτοί οι στόχοι είναι (37):

- **Συμβατότητα:** Τα χαρακτηριστικά μιας H-Anim ανθρώπινης φιγούρας να είναι εφαρμόσιμα σε οποιονδήποτε συμβατό φυλλομετρητή.

- **Ευελιξία:** Δεν γίνονται καθόλου υποθέσεις σχετικά με τους τύπους των εφαρμογών που θα χρησιμοποιήσουν μια H-Anim ανθρώπινη φιγούρα.
- **Απλότητα:** Χαρακτηριστικά που θα μπορούσαν να προκαλέσουν σύγχυση έχουν παραλειφθεί. Το πρότυπο ωστόσο είναι ανοιχτό σε επεκτάσεις.

Χρησιμοποιώντας αυτούς τους τρεις απλούς στόχους, ένα ευθύ πρότυπο έχει δημιουργηθεί το οποίο επιτρέπει άμεση πρόσβαση στην ιεραρχία αρθρώσεων (joints) της ανθρώπινης φιγούρας καθώς και στις κορυφές της γεωμετρίας οι οποίες συναποτελούν το κάθε μέλος (segment) του σώματος ξεχωριστά, κατά τέτοιο τρόπο ώστε να επιτρέπεται η δημιουργία animations με τέτοιο τρόπο που να μην εξαρτάται από το μοντέλο.

## 4.3 ΠΕΡΙΓΡΑΦΙΚΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ

Πριν προχωρήσουμε, θα ήταν καλό να αναφερθούν μερικοί ορισμοί οι οποίοι χρησιμοποιούνται στην συνέχεια για την περιγραφή της δομής του H-Anim (38).

- **Avatar** – ένας εικονικός χαρακτήρας ο οποίος χρησιμεύει σαν αναπαράσταση ενός πραγματικού χαρακτήρα.
- **Displacer** – περιγραφή των περιορισμών κίνησης για μια H-Anim φιγούρα.
- **End effector** – σημείο για χρήση από συστήματα αντίστροφης κινηματικής.
- **H-Anim figure** – 3D αναπαράσταση με αρθρώσεις για την περιγραφή ενός κινηματοποιημένου χαρακτήρα.
- **Humanoid** – ο όρος αναφέρεται σε H-Anim figure που αναπαριστά μόνο έναν χαρακτήρα.
- **Joint** – σημείο σε μια H-Anim figure στην οποία μπορεί να συμβαίνει άρθρωση.
- **Level of articulation** – αριθμός Joints σε μια H-Anim figure
- **Representation System** – σύστημα με δυνατότητες προβολής και επεξεργασίας H-Anim figures.
- **Segment** – μέλος μιας H-Anim figure, το οποίο αναπαριστά την γεωμετρία και την εμφάνιση
- **Site** – σημείο με σημαντικό ενδιαφέρον μέσα σε μια H-Anim figure.

## 4.4 H-ANIM FIGURES

### 4.4.1 Γενικά

Το διεθνές πρότυπο του H-Anim περιγράφει την δομή και τον χειρισμό H-Anim figures. Τα H-Anim figures είναι 3D αναπαραστάσεις που υποστηρίζουν άρθρωση και αναπαριστούν χαρακτήρες με κίνηση. Ενώ κατά κύριο λόγο το H-Anim σχεδιάστηκε με τον ανθρώπινο σκελετό κατά νου, η δομή των αρθρώσεων είναι πολύ γενική και δεν περιορίζεται στον ίδιο αριθμό ποδιών, χεριών ή άλλων μελών του σώματος που αντιστοιχούν σε έναν άνθρωπο. Μια αναπαράσταση μιας και μόνης ανθρώπινης μορφής καλείται *humanoid*.

### 4.4.2 Συναρμολόγηση

Τα H-Anim figures περιγράφονται χρησιμοποιώντας τους ακόλουθους τύπους H-Anim αντικειμένων:

- Humanoid
- Joint
- Segment
- Site και
- Displacer

Το **Humanoid** είναι η ρίζα της H-Anim figure και παρέχει το «πλαίσιο εργασίας» πάνω στο οποίο χτίζονται όλα τα υπόλοιπα μέλη του ανθρωποειδούς.

Ένα **Joint (άρθρωση)** είναι προσκολλημένο είτε πάνω στο αντικείμενο Humanoid είτε πάνω σε άλλα αντικείμενα Joints, χρησιμοποιώντας ένα transform (όπως περιγράφηκε στις ενότητες σχετικά με την VRML) το οποίο περιγράφει την τρέχουσα κατάσταση άρθρωσης, καθώς και την γεωμετρία που σχετίζεται με το προσκολλημένο μέλος σώματος.

Ένα **Segment** περιγράφει τα χαρακτηριστικά των φυσικών συνδέσμων μεταξύ των joints της ανθρωποειδούς φιγούρας. Τυπικά θα αντιστοιχούσε σε ένα ανθρώπινο μέλος, ωστόσο λόγω του ότι το επίπεδο άρθρωσης μπορεί να είναι πιο λεπτομερές, ένα segment θα μπορούσε να περιέχει την 3d αναπαράσταση για λεπτομερέστερα τμήματα ενός μέλους (πχ ένα segment για τον αντίχειρα).

Το **Site** περιγράφει σημεία πάνω στο ανθρωποειδές στα οποία γνωστά semantics μπορεί να συσχετιστούν.

Το **Displacer** παρέχει πληροφορίες σχετικά με το εύρος της κίνησης που επιτρέπεται στο αντικείμενο στο οποίο είναι ενσωματωμένο.

### 4.4.3 *Ιεραρχία του Σκελετού*

Η σκελετική περιγραφή μιας H-Anim figure αποτελείται από ένα «δέντρο» αντικειμένων Joints που περιγράφουν τις μεταφράσεις από το Joint HumanoidRoot (αρχή) μέχρι την τελευταία άρθρωση προς κάθε κατεύθυνση πάνω στο ανθρωποειδές. Η μόνη απαίτηση του H-Anim για το προσδιορισμό της σκελετικής ιεραρχίας είναι ότι θα πρέπει να υπάρχει τουλάχιστον το HumanoidRoot σε κάθε H-Anim figure, ενώ δεν απαιτείται καθορισμένος αριθμός από τα υπόλοιπα Joints ώστε ένα ανθρωποειδές να είναι συμβατό με το H-Anim. Αναμένεται ωστόσο, για να έχει μια H-Anim figure νόημα, πως θα υπάρχουν και άλλα Joints πέρα από το HumanoidRoot.

Ο όρος *Level of Articulation (LOA – Επίπεδο Άρθρωσης)* αναφέρεται στον αριθμό των αρθρώσεων (Joints) που χρησιμοποιούνται για ένα ανθρωποειδές. Μια ανθρωποειδής φιγούρα με 14 αρθρώσεις λέγεται πως έχει ένα «χαμηλό επίπεδο άρθρωσης», ενώ μια φιγούρα με 72 αρθρώσεις μπορεί να χαρακτηριστεί ότι έχει «υψηλό επίπεδο άρθρωσης». Μια σκελετική ιεραρχία που περιέχει μόνο το HumanoidRoot Joint θεωρείται πως έχει το χαμηλότερο δυνατό επίπεδο άρθρωσης που επιτρέπεται για μια H-Anim figure. Για το H-Anim 2.0 έχουν οριστεί 4 επίπεδα άρθρωσης, στα οποία θα αναφερθούμε σε επόμενη παράγραφο.

Τα επίπεδα άρθρωσης μπορούν να χρησιμοποιηθούν για την γενικότερη κατηγοριοποίηση και περιγραφή των δυνατοτήτων κίνησης μιας H-Anim

ανθρωποειδούς φιγούρας. Ωστόσο, ένα H-Anim figure δεν περιορίζεται ώστε να είναι συμβατό με οποιοδήποτε από τα 4 επίπεδα άρθρωσης, και μπορεί να έχει οποιοδήποτε LOA επιθυμεί ο δημιουργός του. Επίσης, μπορεί να χρησιμοποιείται εναλλακτική ιεραρχία σκελετού ταιριαστή στον στόχο της. Με την προϋπόθεση ότι διατηρείται η δομή «προγόνου-απογόνου» μεταξύ των Joints, επιπρόσθετα Joints μπορούν να εισαχθούν ανάμεσα σε δυο ήδη υπάρχοντα Joints στην ιεραρχία.

#### 4.4.4 Χειρισμός

Η προσομοίωση κίνησης για H-Anim figures υλοποιείται εφαρμόζοντας μεταφράσεις (transformations) στα joints, τα οποία ωστόσο περιορίζονται από την σχετική πληροφορία των αντίστοιχων displacers. Η ικανότητα μιας H-Anim figure να επαναδιατάξει τα μέλη και συνολικά το σώμα της εξαρτάται από τον αριθμό joints και segments που χρησιμοποιούνται για να περιγράψουν κάθε μέλος του σώματος. Περισσότερα joints και segments έχουν ως αποτέλεσμα μια πιο λεπτομερή και ευέλικτη φιγούρα, ενώ λιγότερα joints σημαίνουν μια λιγότερο ευέλικτη φιγούρα.

### 4.5 ΤΟ ΑΝΤΙΚΕΙΜΕΝΟ HUMANOID

Η γεωμετρία που περιγράφει το σώμα μιας H-Anim ανθρωποειδούς φιγούρας μπορεί να περιγραφεί με δυο τρόπους: Σκελετική, και Skinned. Περιγράψουμε στις επόμενες παραγράφους τους δυο τύπους γεωμετρίας.

Παραθέτουμε την γενική (αφηρημένη) δομή ενός Humanoid:

```
interface Humanoid {
    float[3] bboxCenter 0 0 0
    float[3] bboxSize -1 -1 -1
    float[3] center 0 0 0
    sequence<string> info []
    sequence<Object> joints []
    string name ""
    float[4] rotation 0 0 1 0
    float[3] scale 1 1 1
    float[4] scaleOrientation 0 0 1 0
    sequence<Object> segments []
    sequence<Object> sites []
    sequence<Object> skeleton []
    sequence<Object> skin []
    sequence<float[3]> skinCoord []
    sequence<float[3]> skinNormal []
    float[3] translation 0 0 0
    string version ""
    sequence<Object> viewpoints []
}
```

### 4.5.1 Σκελετική Γεωμετρία

Η μέθοδος αυτή περιγράφει την γεωμετρία που περιέχεται μέσα στον γράφο σκηνής της σκελετικής ιεραρχίας, που καθορίζεται στο πεδίο *skeleton* του αντικειμένου *Humanoid*. Η γεωμετρία που καθορίζεται μέσα στα αντικείμενα *Segments* αυτής της ιεραρχίας περιγράφει το σώμα σαν ξεχωριστά γεωμετρικά κομμάτια. Αυτή η μέθοδος, ενώ είναι υπολογιστικά επαρκής, μπορεί να προκαλέσει ορισμένες οπτικές ανωμαλίες (όπως εγκοπές ή «τσακίσεις») οι οποίες και αφαιρούν από την εμφάνιση του ανθρωποειδούς.

### 4.5.2 *Skinned body* Γεωμετρία

Η μέθοδος αυτή περιγράφει το σώμα σαν ένα συνεχόμενο κομμάτι γεωμετρίας, χρησιμοποιώντας το *skin* πεδίο του αντικειμένου *Humanoid*. Για αυτή τη μέθοδο, διανυσματικά και σημειακά σύνολα δεδομένων καθορίζονται αρχικά στα πεδία *skinCoord* και *skinNormal* του αντικειμένου *Humanoid*. Τα δεδομένα καθορίζονται κατ' αυτό τον τρόπο ώστε να διαχωριστούν από τους εσωτερικούς μηχανισμούς του αντικειμένου *Humanoid* που χρησιμοποιούν αυτή την πληροφορία. Το αντικείμενο *Humanoid* χρησιμοποιείται τα διανυσματικά σύνολα δεδομένων για συντεταγμένες και κανονικά δεδομένα για να περιγράψει την γεωμετρία που αποτελεί την επιφάνεια της ανθρωποειδούς φιγούρας. Αυτή η φιγούρα μπορεί να υλοποιηθεί σαν ένα μονό *IndexedFaceSet*, ή σαν πολλαπλά *IndexedFaceSets* ή με χρήση κάποιας άλλης αναπαράστασης που παρέχει την ίδια λειτουργικότητα.

Με βάση το πώς το *IndexedFaceSet* γίνεται *render* διαμέσου του διασωληνωτή γραφικών και την διαμόρφωση της ανθρωποειδούς φιγούρας, είναι πιθανό πολλαπλά *IndexedFaceSets* να παρέχουν καλύτερη απόδοση απομονώνοντας τις συνεχείς γεωμετρικές αλλαγές σε τοπικές επιφάνειες. Για τον παραπάνω λόγο, το *H-Anim* δεν περιορίζει την υλοποίηση της επιφάνειας σε μια μονή μέθοδο. Το αντικείμενο *Humanoid* επίσης χειρίζονται τα σύνολα δεδομένων που καθορίζονται στα πεδία *skinCoord* και *skinNormal* για να αντανακλούν τις αλλαγές που συμβαίνουν μέσα στον γράφο σκηνής του πεδίου *skeleton*. Στο πλαίσιο παραμόρφωσης της επιφάνειας, το κάθε αντικείμενο *Joint* της σκελετικής Ιεραρχίας εξυπηρετεί τον σκοπό του καθορισμού του πλαισίου συντεταγμένων μέσα στο οποίο οι κορυφές του συνεχούς πολυγώνου παραμορφώνονται.

## 4.6 ΤΟ ΑΝΤΙΚΕΙΜΕΝΟ JOINT

Το αντικείμενο *Joint* χρησιμοποιείται σαν ένα μπλοκ πάνω στο οποίο χτίζουμε για να περιγράψουμε τις αρθρώσεις της ανθρωποειδούς φιγούρας. Η κάθε άρθρωση μιας φιγούρας αναπαρίσταται από ένα αντικείμενο *Joint*. Αυτά τα αντικείμενα οργανώνονται σε μια ιεραρχία που περιγράφει την εγγενή σχέση προγόνου-απογόνου των αντικειμένων *Joints* του σκελετού, και παρέχει ένα σύνολο πληροφοριών συγκεκριμένα για κάθε *Joint* του σκελετού.

Ένα *Joint* περιγράφει ένα σύστημα συντεταγμένων για τον εαυτό του και τα αντικείμενα που περιέχονται στο πεδίο *children* του. Αυτό το σύστημα συντεταγμένων είναι σχετικό στα συστήματα συντεταγμένων του προγόνου του, το

οποίο, στις περισσότερες περιπτώσεις είναι ένα ακόμη αντικείμενο Joint, ενώ μπορεί να είναι και το αντικείμενο Humanoid. Κατά συνέπεια, το αντικείμενο Joint είναι ένα εξειδικευμένο αντικείμενο για ομαδοποιήσεις (grouping) το οποίο μπορεί να είναι απόγονος ενός άλλου Joint, ή στην περίπτωση του HumanoidRoot, το πρώτο αντικείμενο της σκελετικής ιεραρχίας που περιγράφεται στο πεδίο *skeleton* του αντικειμένου Humanoid.

Για τον χειρισμό μεμονωμένων κορυφών που περιγράφονται στο πεδίο *skinCoord* του αντικειμένου Humanoid, ένα Joint διαθέτει δυο πεδία. Τα εισερχόμενα γεγονότα στο πεδίο *rotation* του αντικειμένου Joint επηρεάζουν τις κορυφές που σημαίνονται από το πεδίο *skinCoordIndex* κατά έναν παράγοντα που καθορίζεται από τις αντίστοιχες τιμές του *vertexWeight* πεδίου του αντικειμένου Joint. Το πεδίο *vertexWeight* περιέχει μια λίστα από τιμές αριθμητικής κινητής υποδιαστολής οι οποίες περιγράφουν έναν παράγοντα «βάρους» που χρησιμοποιείται για να επηρεαστούν οι κατάλληλες κορυφές (όπως σημαίνεται από το πεδίο *skinCoordIndex*) του πεδίου *skinCoord* του αντικειμένου Humanoid. Τα πεδία *vertexWeight* και *skinCoordIndex* χρησιμοποιούνται μόνο όταν καθορίζεται ένα H-Anim μοντέλο με χρήση ενός συνεχούς σχήματος.

Το αντικείμενο Joint επίσης χρησιμοποιείται για να αποθηκεύσει άλλες πληροφορίες σχετικές με την άρθρωση. Συγκεκριμένα, καθορίζεται ένα όνομα για το joint, ώστε οι εφαρμογές να μπορούν να προσδιορίσουν την ταυτότητά του. Επίσης μπορεί να περιέχει ιδιότητες για συστήματα αντίστροφης κινηματικής που ελέγχουν το H-Anim figure. Αυτές οι ιδιότητες περιλαμβάνουν τα άνω και κάτω όρια του joint, τον προσανατολισμό των ορίων, καθώς και μια τιμή αντίστασης. Η εφαρμογή είναι υπεύθυνη για την επιβολή αυτών των ορίων, και υπάρχουν και περιπτώσεις όπου μια εφαρμογή μπορεί να επιλέξει να αγνοήσει αυτά τα όρια.

```
interface Joint {
    float[3] bboxCenter 0 0 0
    float[3] bboxSize -1 -1 -1
    float[3] center 0 0 0
    sequence<Object> children []
    sequence<Object> displacers []
    sequence<float[3]> llimit []
    float[4] limitOrientation 0 0 1 0
    string name ""
    float[4] rotation 0 0 1 0
    float[3] scale 1 1 1
    float[4] scaleOrientation 0 0 1 0
    sequence<integer> skinCoordIndex []
    sequence<float> skinCoordWeight []
    sequence<float[3]> stiffness [0 0 0]
    float[3] translation 0 0 0
    sequence<float[3]> ulimit []
}
```



## 4.7 ΤΟ ΑΝΤΙΚΕΙΜΕΝΟ SEGMENT

Το κάθε μέλος σώματος (πχ, αντιβράχιο, μηρός κλπ) του ανθρωποειδούς αναπαρίσταται από ένα αντικείμενο Segment. Αυτά τα αντικείμενα είναι οργανωμένα στην σκελετική ιεραρχία των Joints του ανθρωποειδούς και παρέχουν αποθήκευση πληροφορίας συγκεκριμένα για κάθε Segment του σώματος. Το αντικείμενο Segment είναι ένα εξειδικευμένο αντικείμενο ομαδοποίησης (grouping) το οποίο παρέχει μια ομαδοποίηση για τα αντικείμενα στο πεδίο *children* του. Ένα αντικείμενο Segment μπορεί να καθοριστεί σαν απόγονος ενός μόνο Joint και ταιριάζεται με το αντίστοιχο αντικείμενο Joint με βάση έναν πίνακα αντιστοίχισης (ο πίνακας παρατίθεται στην ενότητα **Ιεραρχία Joints-Segments** πιο κάτω).

```

Interface Segment {
    float[3] bboxCenter 0 0 0
    float[3] bboxSize -1 -1 -1
    float[3] centerOfMass 0 0 0
    sequence<Object> children []
    sequence<float[3]> coord []
    sequence<Object> displacers []
    float mass 0
    float[9] momentsOfInertia 0 0 0 0 0 0 0 0 0
    string name ""
    sequence<Object> addChildren []
    sequence<Object> removeChildren []
}

```

## 4.8 ΤΟ ΑΝΤΙΚΕΙΜΕΝΟ SITE

Το αντικείμενο Site μπορεί να χρησιμοποιηθεί για τρεις στόχους. Ο πρώτος είναι να καθορίσει μια τοποθεσία πάνω στο ανθρωποειδές που μπορεί να χρησιμοποιηθεί από κάποιο σύστημα αντίστροφης κινηματικής. Ο δεύτερος είναι να καθορίσει ένα σημείο προσκόλλησης για αντικείμενα όπως μενταγιόν ή ρουχισμό. Ο τρίτος είναι να καθορίσει ένα σημείο για μια εικονική κάμερα στο πλαίσιο αναφοράς ενός αντικειμένου Segment (όπως η όψη «διαμέσου των ματιών» του ανθρωποειδούς για χρήση σε κόσμους πολλαπλών χρηστών). Τα αντικείμενα Site που έχουν ως στόχο να χρησιμοποιηθούν σαν σημεία προσκόλλησης και από τα οποία προκύπτει κάποια συγκεκριμένη οπτική (όπως το δεξί ή το αριστερό μάτι), θα πρέπει να περιστρέφονται ώστε να αντικρύζουν στην διεύθυνση που βλέπει η κάμερα.

Τα αντικείμενα Site είναι αντικείμενα ομαδοποίησης τα οποία μπορούν και να οριστούν μόνο με τα πεδία *children* ενός Segment αντικειμένου. Τα πεδία *rotation* και *translation* του αντικειμένου Site καθορίζουν την τοποθεσία και τον προσανατολισμό του τελικού αποτελέσματος μέσα στο πλαίσιο συντεταγμένων του Segment. Το πεδίο *children* του αντικειμένου Site χρησιμοποιείται για να αποθηκεύσει επιπλέον σχήματα (πχ αξεσουάρ) που μπορεί να είναι προσκολλημένα στο αντικείμενο Segment. Το αντικείμενο Site περιγράφει ένα σύστημα συντεταγμένων για αντικείμενα στο πεδίο *children* του, το οποίο είναι σχετικό με το σύστημα συντεταγμένων του αντικειμένου προγόνου του.

Η δομή ενός Site σε VRML:

```

Interface Site {
    float[3] bboxCenter 0 0 0
    float[3] bboxSize -1 -1 -1
    float[3] center 0 0 0
    sequence<Object> children []
    string name ""
    float[4] rotation 0 0 1 0
    float[3] scale 1 1 1
    float[4] scaleOrientation 0 0 1 0
    float[3] translation 0 0 0
    sequence<Object> addChildren []
    sequence<Object> removeChildren []
}

```

## 4.9 ΤΟ ΑΝΤΙΚΕΙΜΕΝΟ DISPLACER

Το σχήμα ενός 3d αντικειμένου μπορεί να χρειαστεί να αλλάξει ανάλογα με τις απαιτήσεις των εφαρμογών. Στο πιο βασικό επίπεδο, αυτό μπορεί να γίνει αλλάζοντας τα δεδομένα στο πεδίο *coord* των αντικειμένων που ορίζουν το σχήμα. Στην περίπτωση avatars με άρθρωση, τα αντικείμενα αυτά περιέχονται σε αντικείμενα Segments, ενώ στην περίπτωση παραμορφώσιμων αντικειμένων καθορίζονται από το πεδίο *skin* του αντικειμένου Humanoid.

Σε μερικές περιπτώσεις, ίσως είναι απαραίτητο να αναγνωρίσουμε συγκεκριμένες ομάδες κορυφών μέσα σε ένα σχήμα. Σε άλλες πάλι, ίσως χρειάζεται να παρέχουμε πληροφορία σχετικά με την κατεύθυνση στην οποία θα πρέπει να κινείται η κάθε κορυφή. Τέτοιου είδους πληροφορία παρέχεται από ένα αντικείμενο Displacer. Για αρθρωμένα avatars, τα αντικείμενα Displacer για ένα συγκεκριμένο αντικείμενο Segment είναι αποθηκευμένα στο πεδίο *displacers* του Segment. Για παραμορφώσιμα mesh avatars, τα αντικείμενα Displacer αποθηκεύονται στα πεδία *displacers* των αντικειμένων Joints στο avatar. Αυτή η πληροφορία, καλούμενη *displacements*, ορίζεται στον τοπικό χώρο του συγκεκριμένου κάθε φορά αντικειμένου Joint, και εν συνεχεία μεταφράζεται στον χώρο του ανθρωποειδούς πριν εφαρμοστεί στο σχήμα.

Ένα αντικείμενο Displacer μπορεί να χρησιμοποιηθεί με τρεις βασικούς τρόπους: Στο πιο βασικό επίπεδο μπορεί να χρησιμοποιηθεί για να εξακριβώσει τις κορυφές που αντιστοιχούν σε ένα συγκεκριμένο τμήμα του σχήματος. Στο επόμενο επίπεδο, μπορεί να χρησιμοποιηθεί για να αναπαραστήσει μια συγκεκριμένη μυική κίνηση η οποία μεταφέρει τις κορυφές προς διάφορες κατευθύνσεις. Ο τρίτος τρόπος με τον οποίο ένα Displacer αντικείμενο μπορεί να χρησιμοποιηθεί είναι για να αναπαραστήσει μια πλήρη διαμόρφωση των κορυφών σε ένα σχήμα.

Το κάθε αντικείμενο Displacer περιγράφει μια τοποθεσία, η οποία καλείται *στόχος μορφοποίησης*, η οποία μπορεί να χρησιμοποιηθεί για να τροποποιήσει τις ιδιότητες displacement της φιγούρας. Το κλιμακωτό μέγεθος του displacement αυτών των αντικειμένων Displacer μπορεί να αλλάξει δυναμικά από κάποια εξωγενή πηγή, όπως έναν interpolator. Κατά συνέπεια, το σχήμα μπορεί να μορφοποιείται ομαλά

χρησιμοποιώντας το βασικό σχήμα και έναν γραμμικό συνδυασμό των *displacements* που καθορίζονται από τα αντικείμενα Displacer.

Παρ' όλο που τα αντικείμενα Displacer χρησιμοποιούνται για να ελέγξουν το σχήμα του προσώπου, μπορούν επίσης να χρησιμοποιηθούν για άλλα μέλη του σώματος.

Η VRML δομή ενός displacer είναι η εξής:

```
Interface Displacer {
    sequence<integer> coordIndex []
    sequence<float> displacements []
    string name ""
    float weight 0
}
```

## 4.10 ΜΟΝΤΕΛΟΠΟΙΗΣΗ

### 4.10.1 Γενικά

Το H-Anim περιορίζει την μοντελοποίηση των ανθρωποειδών φιγουρών ώστε να εξασφαλίζεται το ότι η προσομοίωση κίνησης που έχει σχεδιαστεί για μια H-Anim συμβατή ανθρώπινη φιγούρα να είναι εφαρμόσιμη και σε οποιαδήποτε άλλη H-Anim ανθρώπινη φιγούρα. Αυτοί οι περιορισμοί καθορίζουν την κατάσταση ενός μοντέλου πριν εφαρμοστούν animations, έτσι ώστε να εξασφαλίζεται συνέπεια μεταξύ H-Anim μοντέλων ανθρωποειδών.

### 4.10.2 Μοντελοποίηση Ανθρωποειδών - Συμβάσεις

Το ανθρωποειδές μοντελοποιείται σε όρθια θέση, κοιτάζοντας στην +Z κατεύθυνση με το +Y πάνω και το +X στα αριστερά του ανθρωποειδούς. Η απαρχή των αξόνων (0,0,0) εντοπίζεται στο επίπεδο του εδάφους, ανάμεσα στα πόδια του ανθρωποειδούς.

Τα πόδια βρίσκονται επίπεδα στο πάτωμα, διαχωρισμένα μεταξύ τους περίπου την ίδια απόσταση όσο το πλάτος των ισχίων. Ο πάτος των ποδιών βρίσκεται στο  $Y = 0$ , ενώ τα χέρια είναι ευθεία και παράλληλα στις πλευρές του σώματος με τις παλάμες των χεριών να κοιτούν εσωτερικά προς τους μηρούς. Τα χέρια είναι επίπεδα, με τους άξονες των Joints «1» έως «3» των δαχτύλων να είναι παράλληλοι στον Y άξονα και τον άξονα του αντίχειρα να έχει υποστεί περιστροφή κατά  $\pi/2$  rads προς την κατεύθυνση +Z. Κατά συνέπεια, το σύστημα συντεταγμένων για κάθε Joint στον αντίχειρα είναι ακόμα προσανατολισμένο κατά τέτοιο τρόπο ώστε να ταιριάζει με το συνολικό ανθρωποειδές. Η κίνηση των «0» joints των δαχτύλων τυπικά περιορίζεται αισθητά, και το πόσο εύκαμπτες είναι οι αρθρώσεις διαφέρει από δάχτυλο σε δάχτυλο.

Το πρόσωπο μοντελοποιείται με τα φρύδια σε ανάπαυση, το στόμα κλειστό και τα μάτια ανοιχτά σε πλήρη έκταση.

Το ανθρωποειδές χτίζεται με τις ανθρώπινες διαστάσεις κατά νου, δηλαδή όλες οι διαστάσεις μετριοούνται σε μέτρα. Ένα τυπικό ανθρωποειδές είναι περίπου 1.75 μέτρα

στο ύψος. Στην εικόνα που ακολουθεί παραθέτουμε δυο avatars που να πληρούν τις συμβάσεις που μόλις περιγράψαμε:

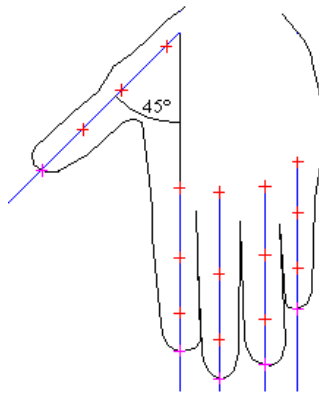


**Εικόνα 24:** Εξ'ορισμού στάση ενός Ανθρωποειδούς για το H-Anim

Σε αυτή τη θέση, όλες οι γωνίες των joints είναι μηδενικές, δηλαδή όλα τα πεδία *rotation* σε όλα τα αντικείμενα Joint θα έχουν την προκαθορισμένη τιμή (0 0 1 0). Επιπρόσθετα, τα πεδία *translation* θα έχουν την προκαθορισμένη τιμή του (0 0 0) και ο παράγοντας *scale* θα έχει επίσης την τιμή (1 1 1). Το μόνο πεδίο που δεν έχει την προκαθορισμένη τιμή για κάθε Joint είναι το *centre*, το οποίο είναι το σημείο γύρω από το οποίο περιστρέφεται το joint (και οι προσκολλημένοι σε αυτό απόγονοι κόμβοι, όπως και το Segment, αν υπάρχουν). Εφαρμόζοντας τις προκαθορισμένες τιμές για την μετάφραση, την περιστροφή και την κλιμάκωση σε όλα τα Joints στο σώμα επιστρέφουμε το σώμα στην ουδέτερη θέση που περιγράψαμε με τις προηγούμενες συμβάσεις. Για να γίνει αυτό, το σύστημα συντεταγμένων για κάθε Joint αντικείμενο προσανατολίζεται για να ταιριάζει με αυτό του συνολικού ανθρωποειδούς.

Το πεδίο *centre* κάθε Joint αντικείμενου θα πρέπει να τοποθετηθεί ώστε τα Joints να περιστρέφονται με τον ίδιο τρόπο που θα περιστρέφονται σε ένα πραγματικό σώμα.

Η παρακάτω εικόνα δείχνει τον προσανατολισμό του χεριού. Οι κόκκινοι «σταυροί» περιγράφουν πιθανά σημεία για τις τιμές του πεδίου *centre* για τα Joints των δαχτύλων, καθώς και των Sites για τις άκρες των δαχτύλων.



**Εικόνα 25:** Προσανατολισμός Χεριού

Προτείνεται, χωρίς να απαιτείται, τα αντικείμενα Segment να είναι κατασκευασμένα με δεδομένες τις εξ'ορισμού τιμές των πεδίων translation, rotation και scaling. Δηλαδή, να μην απαιτείται κάποιου είδους μετασχηματισμός στο σχήμα του Segment ώστε οπτικά να ταιριάζει με τα υπόλοιπα μέλη του σώματος.

Όλες οι συντεταγμένες του σώματος μοιράζονται κοινή αρχή, η οποία είναι αυτή του ίδιου του humanoid. Εάν αυτό αποδειχθεί δύσκολο για ένα εργαλείο επεξεργασίας προς υλοποίηση, είναι αποδεκτό να χρησιμοποιείται οποιοσδήποτε μηχανισμός γεωμετρικών μετασχηματισμών διατίθεται στην γλώσσα κωδικοποίησης για να μετακινήσει την γεωμετρία μέσα σε κάθε αντικείμενο Segment στην σωστή μεριά. Χρησιμοποιώντας τέτοιους μετασχηματισμούς αντί για την απευθείας κατασκευή του επιθυμητού σχήματος μπορεί να έχει επιβάρυνση στην απόδοση του συστήματος.

## 4.11 Η ΔΟΜΗ ΕΝΟΣ ΑΝΘΡΩΠΟΕΙΔΟΥΣ

### 4.11.1 *Ιεραρχία Joints-Segments*

l_hip	l_knee	l_ankle	l_subtalar	l_midtarsal	l_metatarsal
r_hip	r_knee	r_ankle	r_subtalar	r_midtarsal	r_metatarsal
vl5	vl4	vl3	vl2	vl1	
vt12	vt11	vt10	vt9	vt8	vt7

vt6	vt5	vt4	vt3	vt2	vt1	
vc7	vc6	vc5	vc4	vc3	vc2	
l_sternoclavicular	l_acromioclavicular	l_shoulder	l_elbow	l_wrist	vc1	vc1
r_sternoclavicular	r_acromioclavicular	r_shoulder	r_elbow	r_wrist		
HumanoidRoot	sacroiliac (pelvis)	skullbase				

**Πίνακας 3: Βασική Ιεραρχία Joints**

Ένα H-Anim συμβατό σώμα χτίζεται σαν μια σειρά από εμφωλευμένα Joint αντικείμενα, το καθέ ένα από τα οποία μπορεί να έχει ένα σχετιζόμενο Segment. Η ιεραρχία στο παρακάτω σχήμα δείχνει το ζεύγος Joint:Segment το οποίο καθορίζεται από το πρότυπο. Εάν ορίσουμε ένα Joint, και το Joint αυτό χρησιμοποιεί κάποιο όνομα που προβλέπεται από το H-Anim τότε το αντίστοιχο Segment που θα χρησιμοποιηθεί θα πρέπει να είναι ονομασμένο κατά σύμβαση με βάση το πρότυπο, πχ αν χρησιμοποιήσουμε ένα Segment με όνομα l\_upperarm, θα πρέπει να καθοριστεί σαν απόγονος του Joint l\_shoulder, και αντίστοιχα, αν καθορίσουμε ένα Joint r\_knee, και ορίσουμε Segment για αυτό, το αντίστοιχο Segment θα πρέπει να ονομάζεται r\_calf.

Τα v15 και sacroiliac Joints είναι απόγονοι του Humanoidroot. Το HumanoidRoot με τη σειρά του αποθηκεύεται στο πεδίο humanoidBody του αντικείμενου Humanoid, αλλά όλα τα υπόλοιπα αντικείμενα Joints απορρέουν από είτε το v15 είτε το sacroiliac. Εάν αυτά τα αντικείμενα Joints δεν οριστούν, τα Joints χαμηλού επιπέδου μπορεί να είναι απόγονοι του HumanoidRoot.

Η ιεραρχία Joints – Segments για την αναπαράσταση των χεριών είναι πιο λεπτομερής:

**Πίνακας 4: Ιεραρχία Joints για το χέρι**

l_pinky0	l_pinky1	l_pinky2	l_pinky3	l_ring0	l_ring1	l_ring2	l_ring3
l_middle0	l_middle1	l_middle2	l_middle3	l_index0	l_index1	l_index2	l_index3
l_thumb1	l_thumb2	l_thumb3					

r_pinky0	r_pinky1	r_pinky2	r_pinky3	r_ring0	r_ring1	r_ring2	r_ring3
r_middle0	r_middle1	r_middle2	r_middle3	r_index0	r_index1	r_index2	r_index3
r_thumb1	r_thumb2	r_thumb3					

Όσον αφορά το πρόσωπο, πολλές υλοποιήσεις ανθρωποειδών κάνουν χρήση αρθρωμένων δομών προσώπου για να εξομοιώσουν εκφράσεις του προσώπου. Ακολουθεί ένα βασικό σύνολο Joints προσώπου και των αντίστοιχων Segments τους που υποστηρίζουν animation εκφράσεων προσώπου:

**Πίνακας 5: Ιεραρχία Joints για το πρόσωπο**

l_eyeball_joint	r_eyeball_joint
l_eyebrow_joint	r_eyebrow_joint
l_eyelid_joint	r_eyelid_joint
temporomandibular	

Το πρόθεμα “\_joint” χρησιμοποιείται εδώ γιατί τέτοια χαρακτηριστικά ελέγχονται από ομάδες μυών αντί για πραγματικά joints, με το joint temporomandibular να εξαιρείται του παραπάνω. Το πρόθεμα παρέχει έναν διαχωρισμό μεταξύ του ονόματος του Joint και του αντιστοίχου Segment.

Όλα τα αντικείμενα Joints προσώπου είναι απόγονοι του αντικειμένου Joint skullbase. Το κέντρο περιστροφής του ματιού και της κόρης είναι το γεωμετρικό κέντρο της σφαίρας του ματιού. Η περιστροφή της κόρης αντιστοιχεί εξ’ ορισμού σε 0 rad, και μια θετική περιστροφή των  $\pi$  rads θα έκλεινε την κόρη μέχρι να μην φαίνεται καθόλου. Τα φρύδια βρίσκονται σε 0 rad περιστροφή εξ’ ορισμού, ενώ μπορούν να περιστραφούν ως προς την μέση του φρυδιού. Το στόμα είναι κλειστό όταν το Joint temporomandibular βρίσκεται στα 0 rad.

Τα Joints προσώπου από τον παραπάνω πίνακα παρέχουν μόνο μια πρωτόγονη μορφή κίνησης για το πρόσωπο. Μια πιο στιβαρή μορφή κίνησης προσώπου παρέχεται από τις MPEG-4 Παραμέτρους Κίνησης Προσώπου.

Παραθέτουμε την πλήρη ιεραρχία για το H-Anim, στην μορφή <Όνομα\_Joint> : <Όνομα\_Segment>, όπου με εσοχή στην αμέσως επόμενη γραμμή δηλώνεται ο αμέσως επόμενος απόγονος, ενώ με | ενώνονται δυο παράλληλα στο βάθος της ιεραρχίας Joints.

```

HumanoidRoot : sacrum
sacroiliac : pelvis
|   l_hip : l_thigh
|   l_knee : l_calf
|   l_ankle : l_hindfoot
|   l_subtalar : l_midproximal
|   l_midtarsal : l_middistal
|   l_metatarsal : l_forefoot
|   r_hip : r_thigh
|   r_knee : r_calf
|   r_ankle : r_hindfoot
|   r_subtalar : r_midproximal
|   r_midtarsal : r_middistal
|   r_metatarsal : r_forefoot
v15 : 15
v14 : 14
v13 : 13
v12 : 12
v11 : 11
vt12 : t12
vt11 : t11
vt10 : t10
vt9 : t9
vt8 : t8
vt7 : t7
vt6 : t6
vt5 : t5
vt4 : t4
vt3 : t3
vt2 : t2
vt1 : t1
vc7 : c7
| vc6 : c6
|   vc5 : c5
|     vc4 : c4
|     vc3 : c3
|     vc2 : c2
|     vc1 : c1
|       skullbase : skull
|         l_eyelid_joint : l_eyelid
|         r_eyelid_joint : r_eyelid
|         l_eyeball_joint : l_eyeball
|         r_eyeball_joint : r_eyeball
|         l_eyebrow_joint : l_eyebrow
|         r_eyebrow_joint : r_eyebrow
|         temporomandibular : jaw
l_sternoclavicular : l_clavicle
l_acromioclavicular : l_scapula
| l_shoulder : l_upperarm
|   l_elbow : l_forearm
|     l_wrist : l_hand
|       l_thumb1 : l_thumb_metacarpal
|       l_thumb2 : l_thumb_proximal
|       l_thumb3 : l_thumb_distal
|       l_index0 : l_index_metacarpal
|       l_index1 : l_index_proximal
|       l_index2 : l_index_middle
|       l_index3 : l_index_distal
|       l_middle0 : l_middle_metacarpal
|       l_middle1 : l_middle_proximal
|       l_middle2 : l_middle_middle
|       l_middle3 : l_middle_distal
|       l_ring0 : l_ring_metacarpal
|       l_ring1 : l_ring_proximal
|       l_ring2 : l_ring_middle
|       l_ring3 : l_ring_distal
|       l_pinky0 : l_pinky_metacarpal
|       l_pinky1 : l_pinky_proximal
|       l_pinky2 : l_pinky_middle
|       l_pinky3 : l_pinky_distal
r_sternoclavicular : r_clavicle
r_acromioclavicular : r_scapula
| r_shoulder : r_upperarm
|   r_elbow : r_forearm

```



```

r_wrist : r_hand
r_thumb1 : r_thumb_metacarpal
r_thumb2 : r_thumb_proximal
r_thumb3 : r_thumb_distal
r_index0 : r_index_metacarpal
r_index1 : r_index_proximal
r_index2 : r_index_middle
r_index3 : r_index_distal
r_middle0 : r_middle_metacarpal
r_middle1 : r_middle_proximal
r_middle2 : r_middle_middle
r_middle3 : r_middle_distal
r_ring0 : r_ring_metacarpal
r_ring1 : r_ring_proximal
r_ring2 : r_ring_middle
r_ring3 : r_ring_distal
r_pinky0 : r_pinky_metacarpal
r_pinky1 : r_pinky_proximal
r_pinky2 : r_pinky_middle
r_pinky3 : r_pinky_distal

```

### 4.11.2 Προσθέτοντας Joints

Στην παραπάνω ιεραρχία μπορούμε να προσθέσουμε Joints και Segments επιπλέον, αρκεί να τηρούνται οι εξής προϋποθέσεις:

1. Οι κόμβοι Joints που απαριθμούνται στην ιεραρχία, εάν είναι παρρόντες, πρέπει να χρησιμοποιούν τα ενδεικνυόμενα ονόματα.
2. Δεν επιτρέπονται νέοι κόμβοι να παρεμβάλλονται στην προτυποποιημένη ιεραρχία Joints. Οι νέοι κόμβοι επιτρέπεται να είναι απόγονοι είτε ορισμένων από το πρότυπο κόμβων, είτε μη-προτυποποιημένων κόμβων Joints πχ ένας επιπρόσθετος αγκώνας δεν μπορεί να προστεθεί στο χέρι, ωστόσο νέα πρόσθετα (μαλλιά, νύχια), μπορούν να προστεθούν στο ανθρωποειδές χρησιμοποιώντας νέους κόμβους Joints.
3. Οι νέοι κόμβοι θα προστεθούν κατά τέτοιο τρόπο ώστε να μην εμπλέκονται με την κίνηση των υπάρχοντων κόμβων Joints, ακόμα και εάν δεν προβλέπεται κίνηση για αυτούς.

Η προσομοίωση κίνησης για κόμβους Joints από τους βασικούς δεν θα πρέπει να βασίζεται πάνω σε επιπρόσθετους κόμβους Joint (ή απογόνους τους), για λόγους συμβατότητας με άλλα avatars που ανήκουν στο πρότυπο. Τα συστήματα αντίστροφης κινηματικής μπορούν να θεωρήσουν επιπρόσθετους κόμβους Joints όταν κάνουν υπολογισμούς, χωρίς ωστόσο αυτό να είναι απαραίτητο.

Οι επιπρόσθετοι κόμβοι επίσης θα πρέπει να έχουν το πρόθεμα “hanim\_”, πχ “hanim\_x\_newhair” ώστε να διαχωρίζονται από το βασικό σύνολο Joints με το οποίο μπορεί να έχουν παρόμοια ονόματα.

## 4.12 LEVEL OF ARTICULATION – ΕΠΙΠΕΔΑ ΑΡΘΡΩΣΗΣ

Μερικοί τύποι κίνησης για ανθρωποειδή είναι ανεξάρτητοι από τις πραγματικές διαστάσεις του σώματος. Για παράδειγμα, το λύγισμα του κεφαλιού σε μια συγκεκριμένη γωνία θα έχει το ίδιο αποτέλεσμα σε κάθε ανθρωποειδές που έχει έναν κόμβο Joint skullbase. Ωστόσο, αρκετές κινήσεις είναι εξαρτημένες από το μήκος των ξεχωριστών segments ή από τις αναλογίες των μηκών των segments. Για παράδειγμα, το άγγιγμα των ακρών ενός δαχτύλου με την κορυφή της μύτης θα απαιτεί γνώση των διαστάσεων του σώματος.

Οι πίνακες που ακολουθούν παρέχουν τις προτεινόμενες θέσεις για τα κέντρα των joints, για κάθε αντικείμενο Joint, και για την τοποθέτηση των Site αντικειμένων που μπορεί να χρησιμοποιούνται. Οποιοδήποτε ανθρωποειδές δομείται σε τέτοιες διαστάσεις θα είναι σε θέση να μοιράζεται animations με οποιοδήποτε άλλο ανθρωποειδές με τις ίδιες διαστάσεις. Τα ανθρωποειδή με διαφορετικό μέγεθος, και τα οποία χρησιμοποιούν τις ίδιες αναλογίες μηκών segments ίσως να είναι σε θέση να μοιραστούν συγκεκριμένα animations αρκεί η εφαρμογή να αλλάξει τα μεγέθη των τιμών των animations ανάλογα.

Δεδομένου ότι πολλές εφαρμογές μπορεί να μην χρειάζονται το πλήρες σύνολο των joints, οι παρακάτω πίνακες έχουν οργανωθεί ώστε να αντιστοιχούν σε τέσσερα «Επίπεδα Άρθρωσης» - Level of Articulation (LOA). Το να χτίσουμε ένα συγκεκριμένο LOA εξασφαλίζει ότι το ανθρωποειδές θα είναι συμβατό με animations από άλλα ανθρωποειδή που είναι χτισμένα για το ίδιο ή υψηλότερο LOA. Οι προτεινόμενες διαστάσεις σώματος και τα επίπεδα άρθρωσης παρέχονται για λόγους πληροφόρησης, και δεν είναι δεσμευτικά από την προτυποποίηση του H-Anim.

#### 4.12.1 Επίπεδο Άρθρωσης 0

Προβλέπεται μόνο ένα Joint, το HumanoidRoot. Πρόκειται για το ελάχιστο επίπεδο άρθρωσης που είναι συμβατό με το H-Anim.

Πίνακας 6: Επίπεδο Άρθρωσης 0

Joint object	Default centre
HumanoidRoot	0.0000 0.8240 0.0277

#### 4.12.2 Επίπεδο Άρθρωσης 1

Το επίπεδο Άρθρωσης-1 παρέχει μια τυπική χαμηλού-επιπέδου 3D ιεραρχία. Στην πράξη χρησιμοποιείται συχνά, και παρ' όλο τον μικρό αριθμό Joints ( $=<18$ ), είναι αρκετά ευέλικτη.

Πίνακας 7: Επίπεδο Άρθρωσης 1

Joint object	Default center
HumanoidRoot	0.0000 0.8240 0.0277
sacroiliac	0.0000 0.9149 0.0016

l_hip	0.0961 0.9124 -0.0001
l_knee	0.1040 0.4867 0.0308
l_ankle	0.1101 0.0656 -0.0736
l_midtarsal	0.1086 0.0001 0.0368
r_hip	-0.0950 0.9171 0.0029
r_knee	-0.0867 0.4913 0.0318
r_ankle	-0.0801 0.0712 -0.0766
r_midtarsal	-0.0801 0.0000 0.0368
vl5	0.0028 1.0568 -0.0776
skullbase	0.0044 1.6209 0.0236
l_shoulder	0.2029 1.4376 -0.0387
l_elbow	0.2014 1.1357 -0.0682
l_wrist	0.1984 0.8663 -0.0583
r_shoulder	-0.1907 1.4407 -0.0325
r_elbow	-0.1949 1.1388 -0.0620
r_wrist	-0.1959 0.8694 -0.0521

### 4.12.3 Επίπεδο Άρθρωσης 2

Το Επίπεδο Άρθρωσης-2 ορίζει ένα βασικό μοντέλο σκελετού με αρθρωμένα δάχτυλα. Ο αριθμός των Joints που προβλέπεται είναι  $\leq 71$ .

**Πίνακας 8: Επίπεδο Άρθρωσης 2**

<b>Joint object</b>	<b>Default center</b>
HumanoidRoot	0.0000 0.8240 0.0277
Sacroiliac	0.0000 0.9149 0.0016
l_hip	0.0961 0.9124 -0.0001
l_knee	0.1040 0.4867 0.0308
l_ankle	0.1101 0.0656 -0.0736
l_subtalar	0.1086 0.0001 -0.0368
l_midtarsal	0.1086 0.0001 0.0368
l_metatarsal	0.1086 0.0000 0.0762
r_hip	-0.0950 0.9171 0.0029
r_knee	-0.0867 0.4913 0.0318
r_ankle	-0.0801 0.0712 -0.0766
r_subtalar	-0.0801 0.0000 -0.0368
r_midtarsal	-0.0801 0.0000 0.0368

r_metatarsal	-0.0801 0.0039 0.0732
v15	0.0028 1.0568 -0.0776
v13	0.0041 1.1276 -0.0796
v11	0.0048 1.1912 -0.0805
vt10	0.0056 1.2848 -0.0822
vt6	0.0059 1.3866 -0.0800
vt1	0.0065 1.4951 -0.0387
vc4	0.0066 1.5662 -0.0084
vc2	0.0066 1.5928 -0.0103
Skullbase	0.0044 1.6209 0.0236
l_sternoclavicular	0.0820 1.4488 -0.0353
l_acromioclavicular	0.0962 1.4269 -0.0424
l_shoulder	0.2029 1.4376 -0.0387
l_elbow	0.2014 1.1357 -0.0682
l_wrist	0.1984 0.8663 -0.0583
l_thumb1	0.1924 0.8472 -0.0534
l_thumb2	0.1951 0.8226 0.0246

l_thumb3	0.1955 0.8159 0.0464
l_index0	0.1983 0.8024 -0.0280
l_index1	0.1983 0.7815 -0.0280
l_index2	0.2017 0.7363 -0.0248
l_index3	0.2028 0.7139 -0.0236
l_middle0	0.1987 0.8029 -0.0530
l_middle1	0.1987 0.7818 -0.0530
l_middle2	0.2013 0.7273 -0.0503
l_middle3	0.2026 0.7011 -0.0494
l_ring0	0.1956 0.8019 -0.0794
l_ring1	0.1956 0.7815 -0.0794
l_ring2	0.1973 0.7287 -0.0777
l_ring3	0.1983 0.7045 -0.0767
l_pinky0	0.1925 0.8066 -0.1036
l_pinky1	0.1925 0.7866 -0.1036
l_pinky2	0.1938 0.7452 -0.1024
l_pinky3	0.1948 0.7277 -0.1017

r_sternoclavicular	-0.0694 1.4600 -0.0330
r_acromioclavicular	-0.0836 1.4281 -0.0401
r_shoulder	-0.1907 1.4407 -0.0325
r_elbow	-0.1949 1.1388 -0.0620
r_wrist	-0.1959 0.8694 -0.0521
r_thumb1	-0.1899 0.8502 -0.0473
r_thumb2	-0.1874 0.8256 0.0306
r_thumb3	-0.1864 0.8190 0.0506
r_index0	-0.1961 0.8055 -0.0218
r_index1	-0.1961 0.7846 -0.0218
r_index2	-0.1954 0.7393 -0.0185
r_index3	-0.1945 0.7169 -0.0173
r_middle0	-0.1972 0.8060 -0.0468
r_middle1	-0.1972 0.7849 -0.0468
r_middle2	-0.1950 0.7304 -0.0441
r_middle3	-0.1939 0.7042 -0.0432
r_ring0	-0.1951 0.8049 -0.0732

r_ring1	-0.1951 0.7845 -0.0732
r_ring2	-0.1920 0.7318 -0.0716
r_ring3	-0.1908 0.7077 -0.0706
r_pinky0	-0.1926 0.8096 -0.0975
r_pinky1	-0.1926 0.7896 -0.0975
r_pinky2	-0.1902 0.7483 -0.0963
r_pinky3	-0.1908 0.7540 -0.0960

#### 4.12.4 Επίπεδο Άρθρωσης 3

Το Επίπεδο Άρθρωσης-3 είναι πλήρες, με αριθμό Joints =< 89.

**Πίνακας 9: Επίπεδο Άρθρωσης 3**

<b>Joint object</b>	<b>Default center</b>
HumanoidRoot	0.0000 0.8240 0.0277
sacroiliac	0.0000 0.9149 0.0016
l_hip	0.0961 0.9124 -0.0001
l_knee	0.1040 0.4867 0.0308
l_ankle	0.1101 0.0656 -0.0736
l_subtalar	0.1086 0.0001 -0.0368



l_midtarsal	0.1086 0.0001 0.0368
l_metatarsal	0.1086 0.0000 0.0762
r_hip	-0.0950 0.9171 0.0029
r_knee	-0.0867 0.4913 0.0318
r_ankle	-0.0801 0.0712 -0.0766
r_subtalar	-0.0801 0.0000 -0.0368
r_midtarsal	-0.0801 0.0000 0.0368
r_metatarsal	-0.0801 0.0039 0.0732
vl5	0.0028 1.0568 -0.0776
vl4	0.0035 1.0925 -0.0787
vl3	0.0041 1.1276 -0.0796
vl2	0.0045 1.1546 -0.0800
vl1	0.0048 1.1912 -0.0805
vt12	0.0051 1.2278 -0.0808
vt11	0.0053 1.2679 -0.0810
vt10	0.0056 1.2848 -0.0822
vt9	0.0057 1.3126 -0.0838

vt8	0.0057 1.3382 -0.0845
vt7	0.0058 1.3625 -0.0833
vt6	0.0059 1.3866 -0.0800
vt5	0.0060 1.4102 -0.0745
vt4	0.0061 1.4320 -0.0675
vt3	0.0062 1.4583 -0.0570
vt2	0.0063 1.4761 -0.0484
vt1	0.0065 1.4951 -0.0387
vc7	0.0066 1.5132 -0.0301
vc6	0.0066 1.5357 -0.0143
vc5	0.0066 1.5520 -0.0082
vc4	0.0066 1.5662 -0.0084
vc3	0.0066 1.5800 -0.0103
vc2	0.0066 1.5928 -0.0103
vc1	0.0066 1.6144 -0.0034
skullbase	0.0044 1.6209 0.0236
l_eyeball_joint	0.0336 1.6332 0.0502

r_eyeball_joint	-0.0236 1.6331 0.0510
l_sternoclavicular	0.0820 1.4488 -0.0353
l_acromioclavicular	0.0962 1.4269 -0.0424
l_shoulder	0.2029 1.4376 -0.0387
l_elbow	0.2014 1.1357 -0.0682
l_wrist	0.1984 0.8663 -0.0583
l_thumb1	0.1924 0.8472 -0.0534
l_thumb2	0.1951 0.8226 0.0246
l_thumb3	0.1955 0.8159 0.0464
l_index0	0.1983 0.8024 -0.0280
l_index1	0.1983 0.7815 -0.0280
l_index2	0.2017 0.7363 -0.0248
l_index3	0.2028 0.7139 -0.0236
l_middle0	0.1987 0.8029 -0.0530
l_middle1	0.1987 0.7818 -0.0530
l_middle2	0.2013 0.7273 -0.0503
l_middle3	0.2026 0.7011 -0.0494

l_ring0	0.1956 0.8019 -0.0794
l_ring1	0.1956 0.7815 -0.0794
l_ring2	0.1973 0.7287 -0.0777
l_ring3	0.1983 0.7045 -0.0767
l_pinky0	0.1925 0.8066 -0.1036
l_pinky1	0.1925 0.7866 -0.1036
l_pinky2	0.1938 0.7452 -0.1024
l_pinky3	0.1948 0.7277 -0.1017
r_sternoclavicular	-0.0694 1.4600 -0.0330
r_acromioclavicular	-0.0836 1.4281 -0.0401
r_shoulder	-0.1907 1.4407 -0.0325
r_elbow	-0.1949 1.1388 -0.0620
r_wrist	-0.1959 0.8694 -0.0521
r_thumb1	-0.1899 0.8502 -0.0473
r_thumb2	-0.1874 0.8256 0.0306
r_thumb3	-0.1864 0.8190 0.0506
r_index0	-0.1961 0.8055 -0.0218

r_index1	-0.1961 0.7846 -0.0218
r_index2	-0.1954 0.7393 -0.0185
r_index3	-0.1945 0.7169 -0.0173
r_middle0	-0.1972 0.8060 -0.0468
r_middle1	-0.1972 0.7849 -0.0468
r_middle2	-0.1950 0.7304 -0.0441
r_middle3	-0.1939 0.7042 -0.0432
r_ring0	-0.1951 0.8049 -0.0732
r_ring1	-0.1951 0.7845 -0.0732
r_ring2	-0.1920 0.7318 -0.0716
r_ring3	-0.1908 0.7077 -0.0706
r_pinky0	-0.1926 0.8096 -0.0975
r_pinky1	-0.1926 0.7896 -0.0975
r_pinky2	-0.1902 0.7483 -0.0963
r_pinky3	-0.1908 0.7540 -0.0960

## 4.13 ΕΠΙΠΛΕΟΝ ΣΤΟΙΧΕΙΑ

Τα H-Anim figures συχνά εμφωλεύονται σε VRML και X3D κόσμους. Επιπρόσθετα μπορούν να χρησιμοποιηθούν και κάποια επιπλέον στοιχεία στους VRML κόσμους που περιέχουν H-Anim avatars για να εξομαλύνουν την χρήση τους μέσα στα

περιβάλλοντα. Σε αυτά τα περιβάλλοντα, τα H-Anim αντικείμενα αναπαρίστανται στην μορφή VRML ή X3D κόμβων.

### 4.13.1 Viewpoints

Συγκεκριμένες οπτικές του humanoid μπορεί να διευκρινιστούν με δυο τεχνικές. Στην πρώτη τεχνική, ο συγγραφέας του ανθρωποειδούς δημιουργεί κόμβους Site σε κατάλληλες μεριές. Ένας κόμβος Viewpoint μπορεί να δημιουργηθεί και να προσκολληθεί σε οποιονδήποτε από τους κόμβους Site. Ο κόμβος τότε θα είναι σε θέση να καταγράφει στο οπτικό του πεδίο την κίνηση των μελών σώματος του ανθρωποειδούς.

Στην δεύτερη τεχνική, οι κόμβοι Viewpoints υλοποιούνται χρησιμοποιώντας το πεδίο *viewpoints* του κόμβου humanoid για να δημιουργήσουν μια οπτική του ανθρωποειδούς που καταγράφει την κίνησή του μέσα στον χώρο χωρίς να επηρεάζεται από την κίνηση οποιουδήποτε από τους κόμβους Segment του σώματος. Οι κόμβοι Viewpoint που τοποθετούνται στο πεδίο επίσης είναι διαθέσιμοι οποτεδήποτε το ανθρωποειδές φορτώνεται σε έναν φυλλομετρητή, και κατά συνέπεια είναι χρήσιμοι για την προεπισκόπηση του χαρακτήρα.

Υπάρχουν αρκετά viewpoints που μπορούν να περικλυθούν σε ένα τέτοιο αρχείο. Η εμπροσθεν οπτική θα πρέπει να κοιτάει στην  $-Z$  κατεύθυνση, η πλάγια οπτική θα πρέπει να κοιτάει στην  $-X$  κατεύθυνση, ενώ η από πάνω οπτική θα πρέπει να κοιτάει στην  $-Y$  κατεύθυνση. Μια πλάγια οπτική θα κοιτούσε στην κατεύθυνση  $(-1\ 0\ -1)$ , ενώ μια βέλτιστη οπτική θα μπορούσε να παρέχει μια όψη από οποιαδήποτε κατεύθυνση τονίζει καλύτερα τα ξεχωριστά χαρακτηριστικά ενός ανθρωποειδούς. Όλοι αυτοί οι Site κόμβοι θα πρέπει να τοποθετούνται μακριά από το κέντρο του ανθρωποειδούς σε μια ικανοποιητική απόσταση ώστε να φαίνεται ολόκληρο το ανθρωποειδές από την εκάστοτε οπτική.

### 4.13.2 Πλοήγηση

Ένας κόμβος NavigationInfo μπορεί να συμπεριληφθεί στην κορυφή κάθε αρχείου για να τοποθετήσει τον VRML/X3D φυλλομετρητή σε EXAMINE mode, πχ:

```
NavigationInfo { type ["EXAMINE", "ANY"] }
```

Η παραπάνω ρύθμιση επιτρέπει να εξετάσουμε το ανθρωποειδές από όλες τις οπτικές γωνίες.

### 4.13.3 Πολλαπλά ανθρωποειδή ανά Αρχείο

Υπάρχουν περιπτώσεις στις οποίες είναι χρήσιμο να έχουμε πολλαπλά ανθρωποειδή στο ίδιο αρχείο, ίσως μαζί με κάποια άλλα 3D αντικείμενα. Σε τέτοια περίπτωση, κρίνεται απαραίτητο στα ανθρωποειδή να δίνονται ονόματα με χρήση της DEF, και με βάση αυτά τα ονόματα να γίνεται αναφορά σε αυτά σαν παιδιά ενός κόμβου Group με όνομα «HumanoidGroup»:

```
DEF Fred Humanoid { ... }
DEF Jane Humanoid { ... }
DEF Sally Humanoid { ... }
```

```
DEF HumanoidGroup Group { children [ USE Fred, USE
Jane, Use Sally ] }
```

Μια εξωτερική εφαρμογή μπορεί να αποκτήσει μια αναφορά στο HumanoidGroup για την σκηνή, και, αναζητώντας στα children του Group να αποκτήσει αναφορές στα ξεχωριστά Humanoids.

## 4.14 ΣΥΜΠΕΡΑΣΜΑΤΑ

Στο παρρόν κεφάλαιο αναπτύχθηκε το απαραίτητο θεωρητικό υπόβαθρο που χρειάζεται κανείς για να κατανοήσει την δομή ενός H-Anim αρχείου. Το H-Anim είναι εύχρηστο, απλό και ευέλικτο σαν πρότυπο, ενώ έχει δοθεί ιδιαίτερη σημασία κατά την προτυποποίηση ώστε να επιλύονται εύκολα τυχόν ζητήματα συμβατότητας που θα μπορούσαν να προκύψουν.

Ενώ το H-Anim, ακολουθώντας την προτυποποίηση της VRML, χρησιμοποιεί τον κώδικα ASCII για την κωδικοποίηση του περιεχομένου, η συγγραφή ενός H-Anim συμβατού avatar με την χρήση κάποιου κειμενογράφου κρίνεται εξαιρετικά δύσκολη και προφανώς το αισθητικό αποτέλεσμα ακόμα και με πολύ κόπο θα είναι δύσκολα αποδεκτό. Κρίνεται απαραίτητη η ανάπτυξη ενός συνόλου εργαλείων ή ενός πολυ-εργαλείου που θα μπορεί να χειριστεί με τρόπο φιλικό και αποδοτικό προς τον χρήστη τις λειτουργικότητες του προτύπου H-Anim.

Στο επόμενο κεφάλαιο αναλύουμε επακριβώς το κενό που υπάρχει στο λογισμικό που διατίθεται στο διαδίκτυο για την δημιουργία X3D περιεχόμενου που να χρησιμοποιεί το H-Anim, καθώς και το πως επιδιώκει ο συνοδευτικός editor της διπλωματικής εργασίας να καλύψει αυτό το κενό.

ΚΕΦΑΛΑΙΟ 5: ΔΙΑΘΕΣΙΜΟΙ  
EDITORS



---

## ΔΙΑΘΕΣΙΜΟΙ EDITORS

---

Στο παρρόν κεφάλαιο επιδιώκουμε να παρουσιάσουμε συνοπτικά τους διαθέσιμους editors στο διαδίκτυο που καλύπτουν την κατασκευή X3D περιεχομένου (συμβατού με το H-Anim), τις ελλείψεις από πλευράς δυνατοτήτων που υπάρχουν στα διαθέσιμα εργαλεία καθώς και το πως ο συνοδευτικός editor της διπλωματική εργασίας επιδιώκει να τις καλύψει.

### 5.1 ΔΙΑΘΕΣΙΜΟΙ EDITORS

Οι διαθέσιμοι editors είναι εξαιρετικά περιορισμένοι, δεδομένου ότι, λόγω του διαδικτυακού προσανατολισμού του X3D, οι επαγγελματίες δημιουργοί 3D περιεχομένου αρέσκονται στην χρήση πιο στιβαρών προτύπων τα οποία υποστηρίζονται από πανίσχυρα εμπορικά προγράμματα επεξεργασίας 3D περιεχομένου, όπως είναι το 3D Studio Max, το Lightwave και η Maya.

Η πιο διαδεδομένη προσέγγιση για την κατασκευή ποιοτικού X3D περιεχομένου είναι ίσως η κατασκευή του σε κάποια από τα πρότυπα που υποστηρίζουν τα παραπάνω προγράμματα (πχ .3ds) και η μετατροπή τους σε X3D με χρήση κάποιας ενδιάμεσης εφαρμογής που να παρέχει την παραπάνω λειτουργία. Μια τέτοια εφαρμογή είναι η AccuTrans 3D την οποία και χρησιμοποιήσαμε για την δημιουργία της συνοδευτικής βιβλιοθήκης μελών/αντικειμένων του editor H-AnimE.

Στην παρούσα φάση η πληθώρα των διαθέσιμων editors που μπορεί να χρησιμοποιήσει κάποιος παρέχονται στην μορφή Java Applet σε ιστοσελίδες διάσπαρτες στο διαδίκτυο. Οι παραπάνω εφαρμογές περιορίζονται στην δημιουργία X3D περιεχομένου χωρίς να λαμβάνουν υπ'όψιν τους το H-Anim πρότυπο, ωστόσο είναι εξαιρετικά δημοφιλείς για εκπαιδευτικούς σκοπούς, για την εκμάθηση του X3D, καθώς συχνά παρέχουν μια διεπαφή με το X3D script και ταυτόχρονη ανάδραση με τον φυλλομετρητή X3D ώστε να επιτρέπουν τον πειραματισμό με το script.

### 5.2 ΣΧΕΔΙΑΣΤΙΚΟΙ ΣΤΟΧΟΙ ΓΙΑ ΤΟΝ H-ANIME

Η βασική ιδέα πίσω από την δημιουργία ενός editor για το H-Anim ήταν η επέκταση του παλαιότερου framework που αναπτύχθηκε στα πλαίσια διπλωματική εργασίας, στο Εργαστήριο Κατανεμημένων Συστημάτων & Τηλεματικής του ΤΜΗΥΠ. Όπως έχει ήδη αναφερθεί, το framework εκείνο είχε σαν στόχο την υποστήριξη των avatars πριν αυτά εισαχθούν σε κάποιο Δικτυακό Εικονικό Περιβάλλον, συγκεκριμένα σχεδιάστηκε με την πλατφόρμα EVE-II του εργαστηρίου κατά νού. Οι λειτουργίες του framework εκείνου ωστόσο δεν παρείχαν κάποια υποστήριξη για επέμβαση σε θέματα γεωμετρίας του avatar.

Με τον editor H-AnimE που αναπτύχθηκε για την παρούσα διπλωματική εργασία, ο στόχος που τέθηκε ήταν η ανάπτυξη ενός εύχρηστου εργαλείου που να επιτρέπει δυναμικές επεμβάσεις στην γεωμετρία και στις αρθρώσεις ενός H-Anim avatar, πριν αυτό εισαχθεί σε κάποιο Δικτυακό Εικονικό Περιβάλλον. Με δεδομένο τον περιορισμένο αριθμό διαθέσιμων avatars για το H-Anim στο διαδίκτυο, καθώς και το ψυχολογικό «δέσιμο» του χρήστη με το avatar του, θεωρήθηκε σκόπιμο το καινούριο framework να μπορεί να χρησιμοποιηθεί επικουρικά για περιορισμένες επεμβάσεις

---

στην γεωμετρία, πχ. την προσθήκη σχολικής τσάντας στον ώμο ενός avatar πριν αυτό εισαχθεί σε κάποιο ΔΕΠ για μαθησιακούς σκοπούς ή εργατικής κάσκας για κάποιο ΔΕΠ που εξομοιώνει την κατασκευή κτιρίων κλπ.

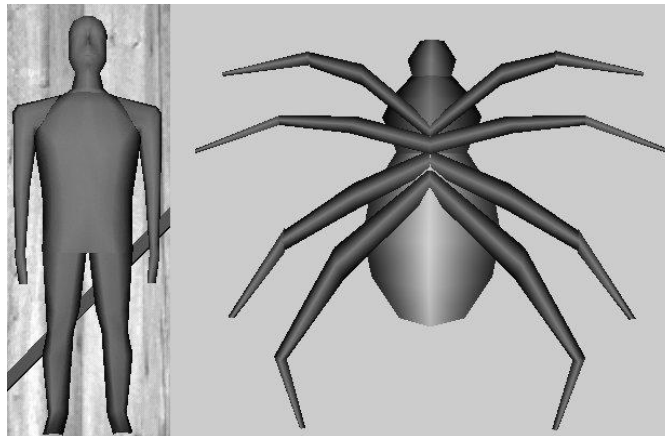
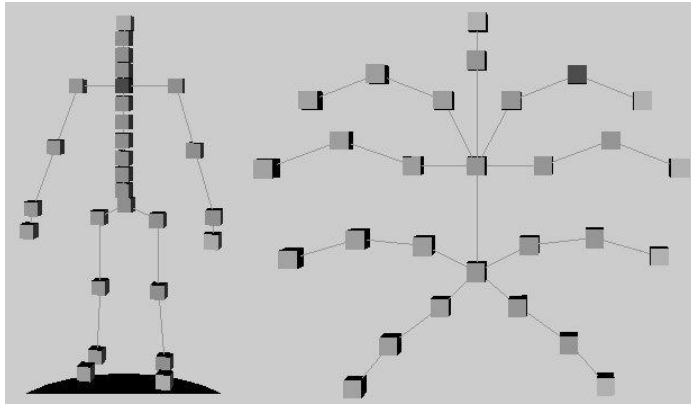
Θέσαμε ως το ελάχιστο των απαιτήσεων το να παρέχονται πλήρεις δυνατότητες για αισθητικά αποδεκτές επεμβάσεις πάνω σε ήδη υπάρχοντα avatar, χωρίς να μπλέξουμε τον χρήστη καθόλου σε θέματα που αφορούν την γεωμετρική αναπαράσταση του avatar. Ο χρήστης εμπλέκεται στο ελάχιστο με αριθμητικούς υπολογισμούς, τους οποίους αναλαμβάνει αυτόματα ο H-AnimE, ενώ τα εργαλεία που του παρέχονται για την τοποθέτηση των γεωμετρικών αντικειμένων ή την αλλαγή των χαρακτηριστικών των κόμβων είναι εξαιρετικά απλά στην χρήση και δεν τον αναμειγνύουν καθόλου με τον κώδικα που θα γραφεί τελικά στο script κατά την αποθήκευση του avatar. Τέλος, το παραπάνω υλοποιήθηκε χωρίς καθόλου πρόσθετους περιορισμούς από πλευράς αναπαράστασης X3D περιεχομένου, συγκεκριμένα, κάποιος χρήστης με όρεξη και μεράκι θα μπορούσε να κατασκευάσει μια πλήρη βιβλιοθήκη με 3d μέλη/αντικείμενα με την οποία να καταφέρνει επεμβάσεις υψηλού επιπέδου στην γεωμετρία κάποιου avatar, ακόμη και την σύνθεση ενός πολυπλοκου avatar ξεκινώντας με πρότυπο κάποιο ήδη υπάρχον avatar με την ελάχιστη δυνατή γεωμετρία.

Ο H-AnimE απευθύνεται στον χρήστη των Δικτυακών Εικονικών Περιβαλλόντων και όχι στον δημιουργό X3D περιεχόμενου, αν και ο τελευταίος θα μπορούσε να βρει τις λειτουργίες του editor εξαιρετικά ενδιαφέρουσες και χρήσιμες.

### 5.3 ΠΡΟΗΓΟΥΜΕΝΗ ΕΡΕΥΝΗΤΙΚΗ ΔΡΑΣΤΗΡΙΟΤΗΤΑ

Προηγούμενες εργασίες οι οποίες αφορούν το H-Anim κατά κύριο λόγο εστιάζουν σε θέματα προσομοίωσης κίνησης, συγκεκριμένα στην δημιουργία animations για ήδη υπάρχοντα avatars (39). Πιο υψηλών προδιαγραφών εργασίες χρησιμοποιούν οπτικά μέσα ή αισθητήρες με τα οποία συλλαμβάνουν την κίνηση από κάποιο φυσικό πρόσωπο και δημιουργούν το απαραίτητο X3D υλικό που αποτελεί προσομοίωση της κίνησης του φυσικού προσώπου στον 3D εικονικό κόσμο. Η προηγούμενη διπλωματική κατά ερευνητική σειρά του εργαστηρίου Κατανεμημένων & Τηλεματικής του ΤΜΗΥΠ αφορούσε στην ενσωμάτωση animations σε κάποιο H-Anim avatar από άλλα avatars που είναι συμβατά με αυτό ως προς το επίπεδο άρθρωσης. Το ενδιαφέρον για θέματα αναπαράστασης κίνησης είναι δικαιολογημένο από την προτυποποίηση του H-Anim. Άλλες εργασίες αφορούν στην αναπαράσταση εκφράσεων του προσώπου.

Οι εργασίες που αφορούν την γεωμετρία των avatars παρουσιάζουν εξαιρετικό ενδιαφέρον δεδομένου ότι επιτρέπουν στον χειριστή παρεμβάσεις που να αλλάζουν την μορφή των μελών ενός avatar, πχ την διόγκωση κάποιου μέλους του σώματος. Μια εξαιρετικά ενδιαφέρουσα εργασία από το Πανεπιστήμιο του Twente της Ολλανδίας εστιάζει στην επεξεργασία της γεωμετρίας των meshes για IndexedFaceSet nodes (40), ενώ επιτρέπει τον συνδυασμό τους πάνω σε ένα μοντέλο της H-Anim Ιεραρχίας με τρόπο παρόμοιο με τον H-AnimE. Το οπτικό αποτέλεσμα που προκύπτει από την παραπάνω επεξεργασία μπορεί να είναι εξαιρετικά θεαματικό, συγκεκριμένα ο φιλόδοξος καλλιτέχνης μπορεί με μια πιο «ευρεία αντίληψη» της ιεραρχίας αρθρώσεων του ανθρώπινου σώματος να δημιουργήσει αρθρωμένες 3D αναπαραστάσεις για άλλους οργανισμούς, πχ αράχνες.



**Εικόνα 26:** Ορισμός Ιεραρχίας Joints εκ νέου

Τέλος, υπάρχοντα προγράμματα εξετάζουν το θέμα της δημιουργίας H-Anim συμβατών avatars από μια ακόμη πιο χαμηλού επιπέδου σκοπιά, δηλαδή ξεκινώντας την δημιουργία της γεωμετρίας από το μηδέν. Ένα τέτοιο πρόγραμμα είναι το Viz3D (41), το οποίο επιτρέπει αρχικά την δημιουργία της γεωμετρίας σαν «ακατέργαστο» τρισδιάστατο περιεχόμενο και τον εν συνεχεία ορισμό του τι ποιο αντικείμενο στην σκηνή θα αποτελέσει το Joint πχ για το χερι, το κεφάλι του avatar κλπ.

Οι παραπάνω εργασίες απευθύνονται κατά κύριο λόγο στους σχεδιαστές X3D περιεχομένου ή σε επαγγελματίες που ενδιαφέρονται για την προσομοίωση κίνησης πάρα στον καθημερινό χρήστη που απλώς θα ήθελε να κάνει ορισμένες μόνο αλλαγές στο avatar που χρησιμοποιεί στο Διαδίκτυο.

## ΚΕΦΑΛΑΙΟ 6: Ο EDITOR H- ANIME

---

# Ο EDITOR H-ANIME

---

## 6.1 ΓΕΝΙΚΑ

Στο κεφάλαιο αυτό αναλύουμε τον editor H-AnimE καθ' αυτόν. Ο editor H-AnimE αναπτύχθηκε σε Java με χρήση της πλατφόρμας Eclipse 3.4.1 ([www.eclipse.org/](http://www.eclipse.org/)) και του πακέτου Xj3D 2.0 (<http://www.xj3d.org/>). Η ανάπτυξη του κώδικα του προγράμματος έγινε κατά κύριο λόγο στο εργαστήριο Κατανεμημένων Συστημάτων & Τηλεματικής του ΤΜΗΥΠ, στο οποίο αναπτύχθηκε και το προηγούμενο σε χρονολογική σειρά framework για την υποστήριξη avatars σε NVE's.

Η βασική ιδέα πίσω από την ανάπτυξη του H-AnimE ήταν η επέκταση των δυνατοτήτων του προηγούμενου framework, για την αποτελεσματικότερη υποστήριξη επεξεργασίας avatars γραμμένων σε VRML και συμβατών με το H-Anim, πριν αυτά εισαχθούν σε κάποιο Δικτυακό Εικονικό Περιβάλλον (συγκεκριμένα την πλατφόρμα EVE-II του εργαστηρίου). Το προηγούμενο framework παρέχει δυνατότητες επισκόπησης ενός avatar, υπολογισμού του επιπέδου άρθρωσης (Level of Articulation) και εισαγωγής των animations από κάποιο άλλο H-Anim συμβατό avatar, εφόσον τα επίπεδα άρθρωσης των δυο avatars το επιτρέπουν. Εν συνεχεία το avatar εισάγεται στο εικονικό περιβάλλον, έτοιμο προς χρήση.

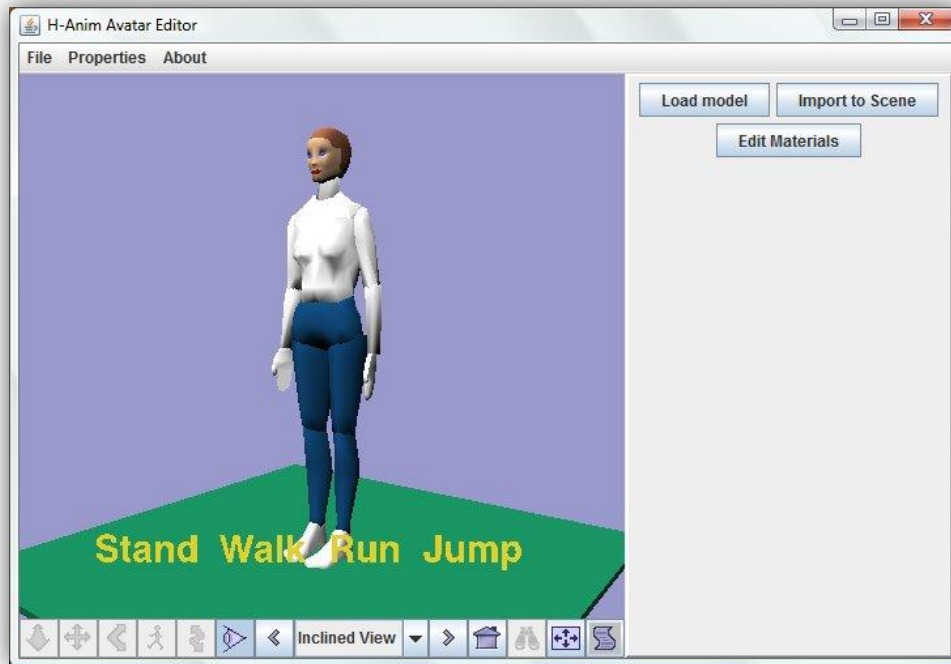
Ως επέκταση του παραπάνω framework, αναπτύξαμε ένα (ανεξάρτητο από το προηγούμενο) framework το οποίο επιτρέπει υψηλότερου επιπέδου επεξεργασία του avatar. Συγκεκριμένα, ο βασικός στόχος είναι το να δίνονται στον χρήστη δυνατότητες επέμβασης πάνω στην γεωμετρία του avatar, στην εμφάνιση του avatar (χρωματισμός, texturing), αλλά και δυνατότητες επέμβασης πάνω στην άρθρωση του avatar με γνώμονα πάντα το να διατηρείται η συμβατότητα με το H-Anim.

## 6.2 ΠΑΡΕΧΟΜΕΝΕΣ ΛΕΙΤΟΥΡΓΙΕΣ

Μια πρώτη γενική αναφορά στις λειτουργίες που παρέχει ο H-AnimE έχει ήδη γίνει στην ενότητα 6.1. Σε αυτή την ενότητα επεκτεινόμαστε απαριθμώντας και αναλύοντας τις λειτουργίες που παρέχει ο H-AnimE.

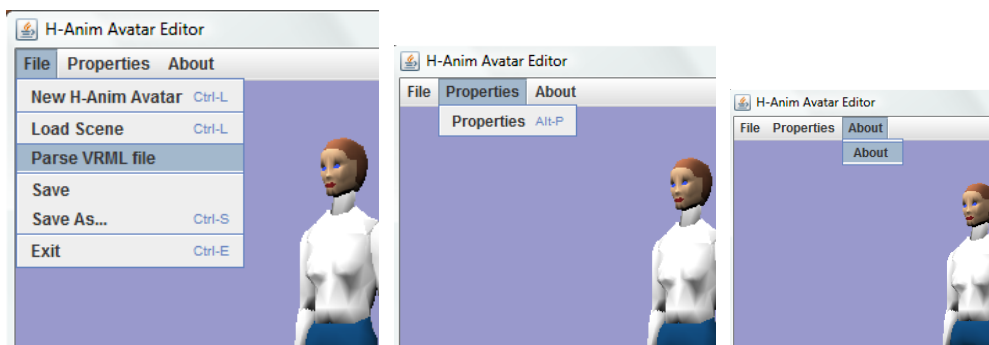
### 6.2.1 *Φόρτωση, προ-φόρτωση και προβολή ενός Avatar*

Σε ένα πρώτο και πιο βασικό επίπεδο, είναι απαραίτητο ο χρήστης να έχει οπτική επαφή με το avatar που επιθυμεί να επεξεργαστεί. Η πρώτη οθόνη του editor παρέχει ένα βασικό layout που προσομοιάζει σε λειτουργία με έναν φυλλομετρητή για VRML, ενώ παρέχει βασικά μενού για την φόρτωση ενός νέου avatar στην οθόνη:



Εικόνα 27: Η κεντρική οθόνη του H-AnimE

Στην παραπάνω εικόνα παρουσιάζουμε την αρχική οθόνη που συναντά ο χρήστης μόλις εκκινεί τον H-AnimE. Συγκεκριμένα, στο αριστερό μέρος της οθόνης βλέπουμε τον φυλλομετρητή (υλοποιημένος με χρήση του πακέτου **Xj3D**), στον οποίο έχουμε φορτώσει ένα H-Anim συμβατό avatar (την nancy), και στο δεξί μέρος της οθόνης βλέπουμε ένα βασικό panel με τρεις δυνατότητες: **Load Model** (φορτώση), **Import to Scene** (εισαγωγής αντικειμένου στο avatar), και **Edit Materials** (επεξεργασίας εμφάνισης του avatar). Επίσης παρέχονται μέσω των μενού οι εξής δυνατότητες:



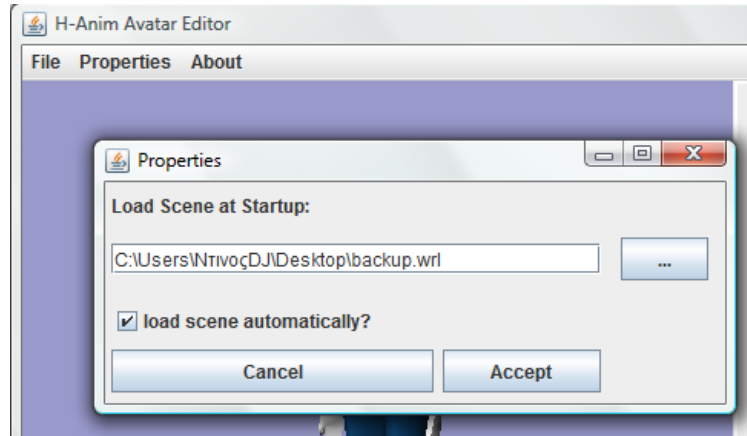
Εικόνα 28: Τα κεντρικά μενού του H-AnimE

Στο πρώτο μενού (μενού **File**) παρέχεται μια βασική λίστα δημιουργίας νέου, **φόρτωσης** και **αποθήκευσης** του επιλεγμένου (φορτωμένου) avatar. Επίσης παρέχεται η λειτουργία **Parse VRML file**, η οποία παρέχει δυνατότητες επικύρωσης ενός VRML αρχείου, στην οποία θα αναφερθούμε στην συνέχεια.

Η φόρτωση ενός avatar (**Load Scene** από το μενού File ή **Load Model** από το δεξί panel) εμφανίζει μια τυπική διεπαφή επιλογής αρχείου από τα διαθέσιμα directories του υπολογιστή. Η επιτυχής επιλογή κάποιου επιτρεπόμενου VRML ή X3D αρχείου

έχει ως αποτέλεσμα την προβολή του στον Xj3D browser στο αριστερό μέρος της οθόνης.

Στο δεύτερο μενού (μενού **Properties**) παρέχεται μια στοιχειώδης διεπαφή για τον ορισμό προφόρτωσης. Με τον όρο προφόρτωση εννοούμε την αυτόματη φόρτωση ενός προ-επιλεγμένου H-Anim avatar κατά την εκκίνηση του H-AnimE, ορίζοντας την τοποθεσία του avatar, καθώς και το αν η εφαρμογή θα το φορτώνει αυτόματα:



**Εικόνα 29:** Προφόρτωση avatar

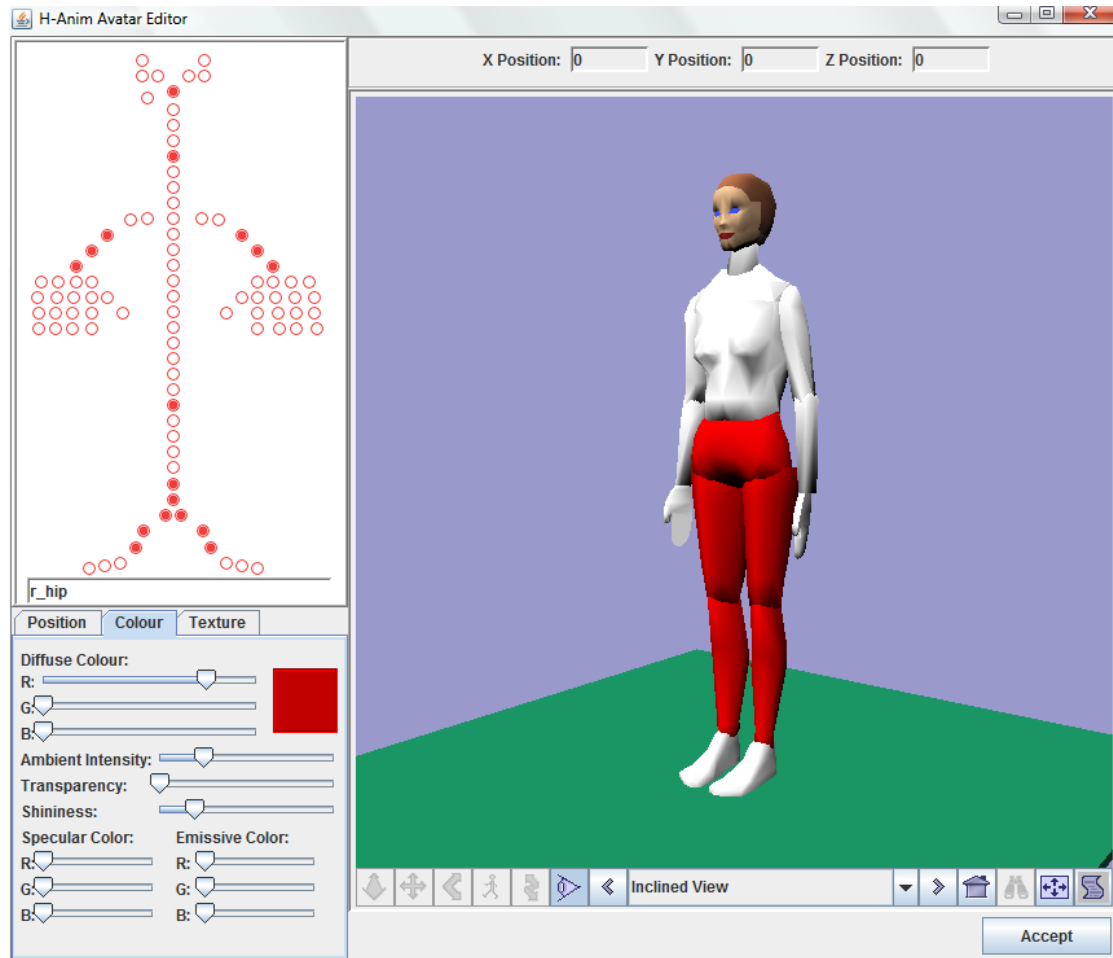
Η λειτουργία προφόρτωσης εκτελεί ακριβώς την ίδια διαδικασία με το να επέλεγε κάποιος χρήστης την λειτουργία φόρτωσης **Load Scene** από το μενού **File** ή από το κουμπί **Load Model** στο δεξί panel της αρχικής οθόνης, με την μοναδική διαφορά ότι εκτελείται αυτόματα και όχι χειροκίνητα. Οι πληροφορίες για το αν θα φορτωθεί κάποιο αρχείο αυτόματα και συγκεκριμένα που βρίσκεται το αρχείο στον σκληρό δίσκο του χρήστη (ή σε κάποιο άλλο μέσο) αποθηκεύονται σε ένα απλό αρχείο με τίτλο startup.txt στο directory του editor.

Το μενού **about** εμφανίζει μια απλή ενημερωτική φόρμα σχετικά με την έκδοση του editor.

### 6.2.2 *Edit Materials – Επεξεργασία Εμφάνισης*

Επιλέγοντας **Edit Materials** από το δεξί panel της οθόνης, ο χρήστης μεταβαίνει αυτόματα σε μια οθόνη επεξεργασίας του avatar. Συγκεκριμένα, η οθόνη χωρίζεται σε τέσσερα μέρη: α) το άνω αριστερό παρουσιάζεται ένα πρότυπο μοντέλο σκελετού με βάση το H-Anim, στο οποίο εμφανίζονται μαρκαρισμένες οι χρησιμοποιούμενες αρθρώσεις (Joints) β) το άνω δεξί, στο οποίο εμφανίζεται ένας VRML φυλλομετρητής που προβάλλει το avatar και ανανεώνει την εικόνα που προβάλλει κάθε φορά που ο χρήστης αλλάζει κάποιο χαρακτηριστικό χρησιμοποιώντας τα υπόλοιπα panel γ) το κάτω αριστερό, στο οποίο παρέχεται ένα τριπλό panel επεξεργασίας των χαρακτηριστικών του avatar με βάση το επιλεγμένο Joint από το μοντέλο του σκελετού και δ) το κάτω δεξί, στο οποίο παρέχονται δυνατότητες προβολής της εικονας του φυλλομετρητή, καθώς και ένα κουμπί αποδοχής αλλαγών.

Η διεπαφή επεξεργασίας της εμφάνισης του avatar φαίνεται στην παρακάτω εικόνα:



**Εικόνα 30:** Χρωματική Επεξεργασία ενός Avatar

Η διαδικασία αλλαγής της εμφάνισης ενός κόμβου του avatar συνοψίζεται στα εξής: επιλογή ενός μαρκαρισμένου Joint (κόμβος με γεμάτο κόκκινο χρώμα) από τον πρότυπο H-Anim σκελετό του άνω αριστερού panel, φόρτωση του ονόματος (r\_hip στην εικόνα) και των στοιχείων του (Diffuse Colour, Ambient Intensity κλπ) στο κάτω αριστερό panel η οποία και γίνεται ακαριαία από το πρόγραμμα, και στην συνέχεια χειροκίνητη από τον χρήστη επεξεργασία των χρωματικών χαρακτηριστικών του κόμβου. Σε όσα χαρακτηριστικά είναι απαραίτητο, παρέχονται κλίμακες R,G,B με τιμές από 0 έως 255 για την σύνθεση χρώματος.

### 6.2.2.1 Texturing

Το texturing υλοποιείται με παρόμοιο τρόπο, χρησιμοποιώντας το τρίτο tab “Texture”. Η διεπαφή που παρουσιάζεται είναι ιδιαίτερα απλή, συγκεκριμένα ένα textbox στο οποίο ο χρήστης μπορεί να παρέχει χειροκίνητα το URL για το που βρίσκεται το texture του στον σκληρό του δίσκο (ή στο διαδίκτυο) και ένα κουμπί που προβάλλει μια οθόνη παρόμοια με την εξερεύνηση των windows, και την οποία μπορεί να χρησιμοποιήσει ο χρήστης για να αποφύγει το χειροκίνητο γράψιμο της τοποθεσίας του texture. Η ανανέωση του texture στην οθόνη γίνεται αυτόματα μόλις επιλεγεί.

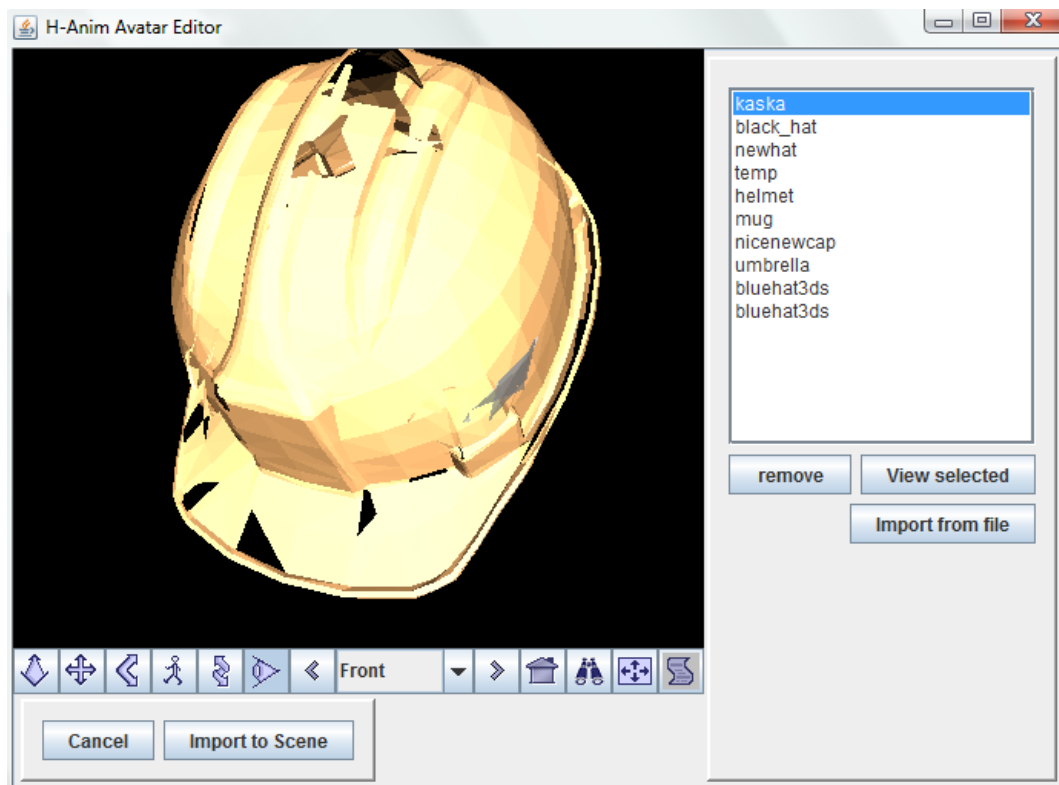


### 6.2.3 Επεμβάσεις στην Γεωμετρία

Ένας από τους πρωταρχικούς κατασκευαστικούς στόχους είναι το να παρέχεται στον χρήστη η δυνατότητα, με την χρήση του editor, να επεμβαίνει δυναμικά στην γεωμετρία του avatar του πριν το εισάγει σε κάποιο εικονικό περιβάλλον. Η χρησιμότητα του παραπάνω είναι προφανής: ανάλογα με τις απαιτήσεις ή την χρήση του περιβάλλοντος το avatar θα πρέπει να μπορεί να προσαρμόζεται, προσφέροντας μεγαλύτερο βαθμό ελευθερίας και ρεαλισμού. Για παράδειγμα, αν η Nancy (το 3D μοντέλο που παρουσιάζεται στις εικόνες) εισαχθεί σε ένα Εικονικό Περιβάλλον το οποίο εξομοιώνει ένα γήπεδο γκολφ, στοιχεία που θα προσέθεταν βαθμούς λειτουργικότητας και ρεαλισμού θα ήταν η προσθήκη ενός μαστουνιού του γκολφ στο δεξί χέρι της, ένα καπελό γκολφ στο κεφάλι, και ίσως ένας σάκος μαστουνιών στην πλάτη της. Το κρίσιμο στοιχείο στα παραπάνω είναι τα πρόσθετα στοιχεία να ακολουθούν την συνολική κίνηση του σώματος του avatar.

#### 6.2.3.1 Εισάγοντας από την βιβλιοθήκη

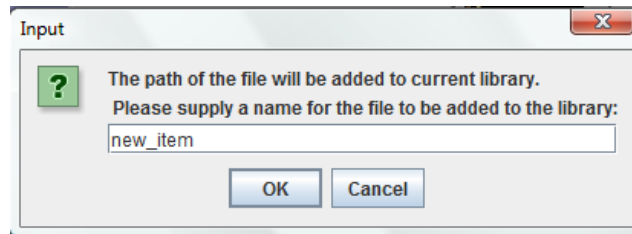
Για να εξυπηρετηθεί καλύτερα ο χρήστης, δημιουργήθηκε μια βιβλιοθήκη από πρότυπα VRML αντικείμενα/μέλη, από τα οποία να μπορεί να επιλέξει αυτό που θέλει να προσαρτήσει στο avatar του. Μέσω του κουμπιού **Import to Scene** της αρχικής φόρμας, ο χρήστης μεταβαίνει στην βιβλιοθήκη:



Εικόνα 31: Η βιβλιοθήκη του H-AnimE

Στην οθόνη αυτή παρέχεται στον χρήστη η διεπαφή μέσω της οποίας μπορεί να επιλέξει αντικείμενα από την βιβλιοθήκη, να προσθέσει σε αυτήν, να διαγράψει ή απλώς να περιηγηθεί στα περιεχόμενά της:

- Επιλέγοντας ένα αντικείμενο της λίστας στα δεξιά, πχ το kaska, ο editor αυτομάτως προβάλλει το αρχείο στον φυλλομετρητή στα αριστερά. Την ίδια λειτουργία εκτελεί και το κουμπί **View Selected**, το οποίο προστέθηκε για την (συχνή) περίπτωση στην οποία ο Xj3D browser αποτυγχάνει με την πρώτη να προβάλλει το αντικείμενο.
- Επιλέγοντας ένα αντικείμενο και πατώντας **remove**, το αντικείμενο διαγράφεται από την λίστα. Την επόμενη φορά που ο χρήστης θα ανοίξει τον H-AnimE, το αντικείμενο δεν θα υπάρχει στην λίστα.
- Πατώντας **Import from file**, παρέχεται στον χρήστη μια διεπαφή πλοήγησης στα directories του σκληρού του δίσκου με την οποία καλείται να επιλέξει ένα VRML/X3D αρχείο που να περιέχει το αντικείμενο προς προσθήκη. Τυπική απαίτηση, ωστόσο όχι δεσμευτική, είναι το αρχείο να μην περιέχει περιττή πληροφορία, δηλαδή να περιέχει μόνο τους κόμβους προς προσθήκη. Αν η παραπάνω διαδικασία είναι επιτυχής, ο χρήστης καλείται να παρέχει ένα αναγνωριστικό όνομα:



Εικόνα 32: Ορισμός ονόματος νέου αντικειμένου

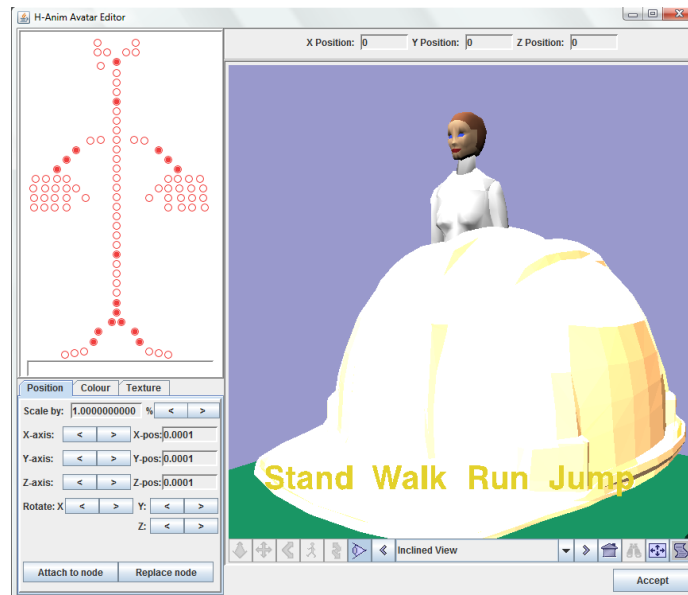
Αφού ολοκληρωθεί η παραπάνω διαδικασία, το αντικείμενο βρίσκεται στην λίστα, έτοιμο προς προβολή.

- Με το κουμπί **Import to Scene** το αντικείμενο που έχει επιλεγεί μεταβαίνει στην αμέσως επόμενη οθόνη μαζί με το avatar, ώστε ο χρήστης να ταιριάζει τις διαστάσεις των δυο αντικειμένων.
- Με το κουμπί **Cancel** ακυρώνεται η συνολική διαδικασία και επιστρέφουμε στην αρχική οθόνη.

Η όλη διαδικασία γίνεται με διαχείριση ενός txt αρχείου το οποίο κρατάει ζεύγη της μορφής (τοποθεσία αρχείου, αναγνωριστικό αρχείου). Η διαχείριση του txt αρχείου εξασφαλίζει ότι την επομένη φορά που θα ανοίξει ο editor, η βιβλιοθήκη θα έχει συγκρατήσει τα δεδομένα της από την τελευταία φορά που χρησιμοποιήθηκε.

### 6.2.3.2 Ταιριάζοντας το αντικείμενο στο avatar

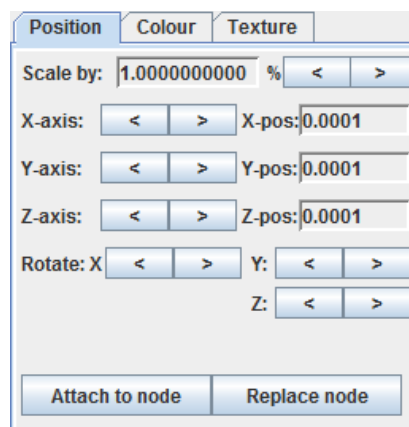
Η αμέσως επόμενη οθόνη είναι ίδια με την οθόνη επεξεργασίας εμφάνισης, με την διαφορά ότι αυτή τη φορά, στο κάτω αριστερό panel είναι ενεργοποιημένο το subpanel **position**. Το subpanel αυτό παρέχει πλήρη εργαλεία για την ακριβή τοποθέτηση του αντικειμένου, την περιστροφή και την κλιμάκωση του αντικειμένου.



**Εικόνα 33:** Εισαγωγή αντικειμένου στην οθονη του avatar

Η παραπάνω εικόνα παρουσιάζει το οπτικό αποτέλεσμα αμέσως μετά την επιλογή του κουμπιού **Import to Scene** αν επιλέξουμε το εργατικό καπέλο. Βασική υπόθεση του editor, όπως και του προτύπου H-Anim, είναι ότι οι διαστάσεις του avatar είναι στην κλίμακα του μέτρου. Ωστόσο δεν είναι εξασφαλισμένο ότι την ίδια σύμβαση ακολουθούν και τα αντικείμενα της βιβλιοθήκης, κατά συνέπεια, το πρόγραμμα επιλέγει αυτομάτως να κλιμακώσει (μεγέθυνση ή σμίκρυνση) τις διαστάσεις του αντικειμένου με βάση το 1 μέτρο.

Όπως προαναφέρθηκε, το subpanel **Position** παρέχει πλήρη εργαλεία για την τοποθέτηση του αντικειμένου στο σωστό σημείο, πριν αυτό ενσωματωθεί στο avatar. Το subpanel είναι αόρατο στην περίπτωση που ο χρήστης επιλέξει απλώς να επεξεργαστεί την εμφάνιση του avatar, ωστόσο, στην περίπτωση που επιλέξει να προσθέσει κάποιο αντικείμενο σε κάποιο avatar, οι δυνατότητες επεξεργασίας εμφάνισης (subpanels **Colour** και **Texture**) συνεχίζουν να παρέχονται.



**Εικόνα 34:** Επεξεργασία Θέσης Αντικειμένου

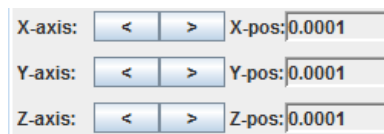
Στο subpanel **Position** παρέχονται τα εξής εργαλεία:

- **Κλιμάκωσης:** παρέχεται στον χρήστη ένας παράγοντας κλιμάκωσης, ο οποίος αρχικά είναι στην τιμή 1.000...0%, και τον οποίο μπορεί να αυξομειώνει πιέζοντας τα δυο κουμπια δεξιά, κλιμακώνοντας ανάλογα το αντικείμενο στην οθόνη.



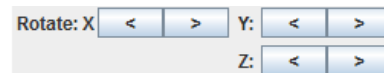
Το οπτικό αποτέλεσμα φαίνεται άμεσα στην οθόνη.

- **Μετάφρασης:** παρέχονται στον χρήστη εργαλεία για την αλλαγή του κέντρου του αντικειμένου στους άξονες X,Y,Z:



Πιέζοντας τα κουμπιά που αντιστοιχούν σε κάθε άξονα το αντικείμενο μεταφέρεται ανάλογα στην οθόνη.

- **Περιστροφής:** παρέχονται στον χρήστη εργαλεία περιστροφής σε τρεις άξονες:



Τα παραπάνω εργαλεία αντιστοιχούν στην καταγραφή τιμών ενός Transform κόμβου, απόγονος του οποίου ορίζεται το αντικείμενο που θα εισάγουμε. Ο κόμβος με την σειρά του ορίζεται απόγονος του επιθυμητού Joint.

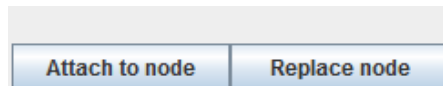
Ωστόσο, πριν ο χρήστης επιλέξει οποιαδήποτε επέμβαση πάνω στο αντικείμενο, θα πρέπει να επιλέξει το Joint στο οποίο θα επιλέξει να προσαρτήσει το αντικείμενο, με χρήση του H-Anim μοντέλου σκελετού στο άνω αριστερό μέρος της οθόνης. Αφού επιλέξει κάποιο Joint, το αντικείμενο αυτομάτως αλλάζει συντεταγμένες κέντρου ώστε να έρθει όσο πιο κοντά γίνεται στο κέντρο του επιθυμητού Joint.

Το οπτικό αποτέλεσμα μιας διαδικασίας ταιριάσματος για το παράδειγμά μας θα ήταν:



**Εικόνα 35:** Συνταίριασμα Avatar - Αντικειμένου

Τέλος ο χρήστης καλείται να επιλέξει αν θέλει να προσθέσει το αντικείμενο στην γεωμετρία του Joint ή να αντικαταστήσει την παλιά του γεωμετρία.

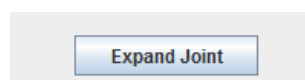


Επιλέγοντας **Attach to node** το αντικείμενο θα προστεθεί στην γεωμετρία του κόμβου όταν ο χρήστης επιχειρήσει να αποθηκεύσει το avatar, ενώ αν επιλέξει **Replace Node** αντικαθιστά πλήρως την γεωμετρία του κόμβου.

### 6.2.3.3 Επέκταση H-Anim Ιεραρχίας

Θα ήταν εξαιρετικά χρήσιμο για τον χρήστη να μπορεί να επεκτείνει την ήδη υπάρχουσα ιεραρχία Joints προσθέτοντας κάποιο νέα άρθρωση στο avatar του. Περιορισμοί που υπάρχουν στην προσθήκη επιπλέον Joints και οι οποίοι επιβάλλονται από το πρότυπο του H-Anim είναι ότι τυχόν νέα Joints θα πρέπει να έχουν κάποιον ήδη υπάρχοντα πατέρα με γεωμετρία στο avatar, καθώς και το ότι θα πρέπει να ακολουθούν την ονοματοδοσία του προτύπου.

Η λειτουργία επέκτασης της H-Anim Ιεραρχίας ενός avatar με νέες αρθρώσεις συνοψίζεται στα εξής: Ο χρήστης εισάγει ένα γεωμετρικό αντικείμενο (πχ. την αναπαράσταση ενός δακτύλου) με την ίδια διαδικασία που παρουσιάστηκε στην υποενότητα «Εισάγοντας από την βιβλιοθήκη», επιλέγει ένα μη-χρησιμοποιούμενο Joint του οποίου το Joint-πατέρας χρησιμοποιείται και, αφού ταιριάζει την γεωμετρία με τα εργαλεία που παρουσιάζονται στην υποενότητα «ταιριάζοντας το αντικείμενο στο avatar» επιλέγει να προσαρτήσει το αντικείμενο ως νέο Joint χρησιμοποιώντας το κουμπί “Expand Joint”.



Όπως και προηγουμένως, οι νέες προσθήκες στην γεωμετρία φαίνονται απευθείας στον φυλλομετρητή του Xj3D, ενώ για να αποθηκευτούν θα πρέπει ο χρήστης να πατήσει το κουμπί “Accept” και να γυρίσει στην προηγούμενη οθόνη.

Το μόνο σημείο στο οποίο η διαδικασία δεν εκτελείται είναι αν ο χρήστης επιλέξει κάποιο Joint του οποίου το Joint-πατέρας δεν χρησιμοποιείται, όπως ακριβώς ορίζει

το H-Anim. Ένα παράδειγμα της παραπάνω περίπτωσης θα ήταν πχ η αλλόκοτη περίπτωση στην οποία ο χρήστης εισάγει την γεωμετρία για κάποιο δάκτυλο χωρίς να έχει εισάγει άρθρωση για τον καρπό.

Από τεχνικής απόψεως, ο λόγος για τον οποίο το H-Anim δεν υποστηρίζει Joints χωρίς χρησιμοποιούμενο Joint-Πατέρα είναι ο εξής: όταν ο χρήστης επιλέξει να αποθηκεύσει το avatar του σε VRML/X3D αρχείο, ο H-AnimE αναλαμβάνει να επέμβει στο πεδίο “children” του επιλεγμένου Joint – πατέρα και να αποθηκεύσει εκεί όλες τις πληροφορίες που χρειάζονται για την αναπαράσταση του νέου Joint.

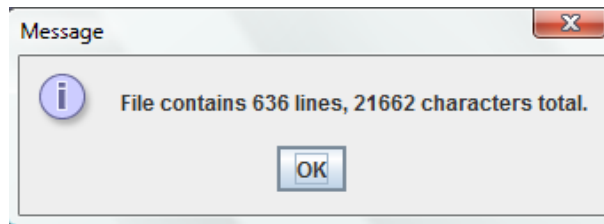
```
DEF hanim_vl5 Joint {
  name          "vl5"
  center        0.0028 1.0568 -0.0776
  children [
    DEF hanim_skullbase Joint {
      name       "skullbase"
      center     0.0044 1.6209 0.0236
      children [
        DEF hanim_skull Segment {
          name    "skull"
          children [
            ...
          ]
        }
      ]
    }
  ]
}
```

Το παραπάνω είναι ένα παράδειγμα στο οποίο πατέρας-κόμβος είναι ο h\_animvl5, και σε αυτόν προστίθεται ο hanim\_skullbase μαζί με το Segment του, hanim\_skull. Ο H-AnimE αναλαμβάνει να συμπληρώσει αυτόματα το κέντρο για κάθε Joint, το οποίο θεωρητικά θα επέτρεπε κάποιος χρήστης να δημιουργήσει ένα avatar από το μηδέν, έχοντας στην διάθεσή του τις επιθυμητές γεωμετρίες. Στην πράξη κάτι τέτοιο βέβαια θα ήταν εφικτό αλλά εξαιρετικά χρονοβόρο, καθώς θα απαιτούσε αλληπάλληλες αποθηκεύσεις-επαναφορτώσεις του avatar κάθε φορά για να ανανεωθούν οι νέες προσθήκες. Να σημειωθεί ότι το παραπάνω δεν είναι πρόβλημα μηχανισμού του H-AnimE, αλλά εγγενής αδυναμία του πακέτου Xj3D, το οποίο δεν είναι σε θέση να χειριστεί πολλαπλές/μαζικές παρεμβάσεις στην γεωμετρία και χρειάζεται επαναφόρτωση της επιθυμητής σκηνής για να τις ανανεώσει.

## 6.2.4 VRML Parser

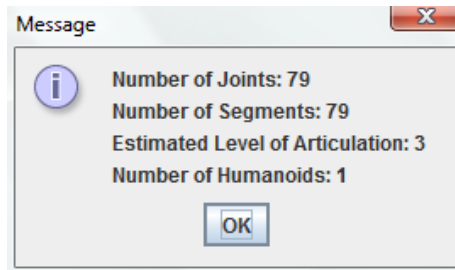
Από την αρχική οθόνη ο χρήστης μπορεί να επιλέξει από το μενού **File** την επιλογή **Parse VRML file**, με την οποία του παρέχονται χρήσιμες πληροφορίες σχετικά με την εγκυρότητα του αρχείου που θα επιλέξει, και την συμβατότητά του με το πρότυπο H-Anim. Παρά το όνομά του, δουλεύει και για X3D αρχεία, αρκεί να μην χρησιμοποιούν XML σύνταξη.

Αρχικά παρέχεται μια διεπαφή με την οποία επιλέγει το επιθυμητό αρχείο. Εν συνεχεία, ο editor αναλαμβάνει να διαβάσει το αρχείο με χρήση ενός FileReader της βιβλιοθήκης java.io.\*, ο οποίος διαβάζει έναν-έναν τους χαρακτήρες του αρχείου. Ο editor παρέχει χρήσιμες πληροφορίες όπως ο αριθμός γραμμών και χαρακτήρων που χρησιμοποιούνται (ο ‘\n’ για αλλαγή γραμμών δεν προσμετράται):



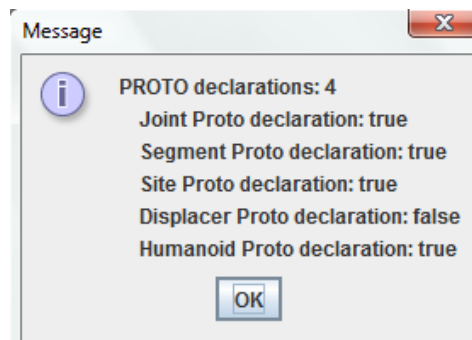
**Εικόνα 36:** 1<sup>η</sup> οθόνη/μήνυμα του VRMLParser

Ο αριθμός των Joints, Segments, το επίπεδο Άρθρωσης που χρησιμοποιείται:



**Εικόνα 37:** 2<sup>η</sup> οθόνη/μήνυμα του VRMLParser

Καθώς και χρήσιμες πληροφορίες σχετικά με τα PROTO declarations:



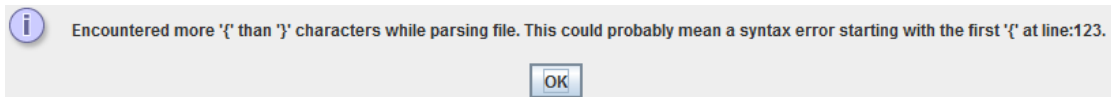
**Εικόνα 38:** 3<sup>η</sup> οθόνη/μήνυμα του VRMLParser

Συνολικά, από τον VRML parser παρέχονται οι εξής λειτουργίες:

- Απαρίθμηση **γραμμών** και **χαρακτήρων**.
- Απαρίθμηση κόμβων **Joints, Segments, Humanoids, Displacers** και **Sites**, οι οποίοι χρησιμοποιούνται από κάποιο H-Anim αρχείο.
- Υπολογισμός του **Επιπέδου Άρθρωσης – Level of Articulation** του avatar. Στην περίπτωση που το αρχείο δεν περιέχει ούτε έναν κόμβο Joint, το επιστρεφόμενο Level of Articulation είναι το -1.
- Αναγνώριση **της έκδοσης VRML** που χρησιμοποιείται (πχ 2.0).
- **Υπολογισμός** συνολικών **ROUTES** (για τα animations), και **παροχή μηνύματος λάθους** στην περίπτωση που κάποιο ROUTE δεν έχει σωστή σύνταξη.
- **Επιβεβαίωση** για το ποια **PROTOs** έχουν δήλωση και ποια χρησιμοποιούνται και δεν έχουν δήλωση. Στην περίπτωση, πχ που

χρησιμοποιούνται Displacers και απουσιάζει το αντίστοιχο PROTO declaration, ο VRML Parser παρέχει αντίστοιχο ειδοποιητικό μήνυμα.

- **Έλεγχος για λογικές ασυνέπειες**, πχ αν σε ένα αρχείο υπάρχουν παραπάνω Segments απ' ό,τι Joints τότε κάποιο Segment δεν θα πρέπει να έχει δηλωθεί ή δεν χρησιμοποιείται, ενώ αν το αρχείο έχει παραπάνω Joints απ' ό,τι Segments τότε κάποιο Joint είναι κενό γεωμετρικού περιεχομένου.
- **Έλεγχος για συντακτικά λάθη**, πχ στην περίπτωση που υπάρχουν παραπάνω { απ' ό,τι } ή [ χωρίς αντίστοιχο ], τότε προκύπτει συντακτικό λάθος, για το οποίο ο χρήστης ενημερώνεται με μήνυμα λάθους, το οποίο του παρέχει και τον αριθμό γραμμής που πιθανότατα αυτό εντοπίζεται.



Εικόνα 39: 4<sup>η</sup> οθόνη/μήνυμα του VRMLParser

Παραθέτουμε ενδεικτικά τον κώδικα του VRML parser:

```

_fileChooser.setCurrentDirectory(new java.io.File("."));
_fileChooser.setDialogTitle("Choose!");
_fileChooser.setFileSelectionMode(JFileChooser.FILES_ONLY);
_fileChooser.setAcceptAllFileFilterUsed(false);

_fileChooser.setFileFilter(null);
if ( _fileChooser.showOpenDialog(this) == JFileChooser.APPROVE_OPTION) {
    File file = _fileChooser.getSelectedFile();
    JOptionPane.showMessageDialog(null,"Attempting to parse: " +
file.getAbsolutePath() + "... ");
    String URLfrom = file.getAbsolutePath();
    FileReader file2; int uCount=0; int LineCount=0; int CharCount=0;
    boolean iread = true; char c; int uj=0; char backstrip; int
JointCount=0; int SegmentCount=0;
    int HumanoidCount=0; int ROUTEcount=0; int PROTOcount=0; int
SiteCount=0; int DisplacerCount=0;
    boolean HumanoidPROTOdeclaration = false; boolean
SitePROTOdeclaration = false;
    boolean DisplacerPROTOdeclaration = false; boolean
JointPROTOdeclaration = false; boolean SegmentPROTOdeclaration = false;
    String CurrWord = ""; String Back1Word = ""; String Back2Word = "";
String Back3Word = ""; String Back4Word = "";
    try {
        file2 = new FileReader(URLfrom);
        BufferedReader reader = new BufferedReader(file2);
        while (iread == true)
        {
            if (! reader.ready()) break;
            c = (char) reader.read();
            //controls
            if (c == '\n'){LineCount++;}else{CharCount++;}
            //construct words
            if (c == '#')
            {
                boolean jread = true;
                //ignore everything until \n, since this is comments.
                while (jread == true)
                {
                    if (! reader.ready()) break;
                    c = (char) reader.read();
                    if (c == '\n'){jread = false;}
                }
            }
            if ((c > 47 && c < 58 || c > 63 && c < 91 || c > 93 && c < 123 || c
> 41 && c < 47)&& c != '#')
            {
                CurrWord = CurrWord + c;
            }
        }
    }
}

```



```

        else
        {
            backstrip = c;
            Back4Word = Back3Word; Back3Word = Back2Word; Back2Word = Back1Word;
            Back1Word = CurrWord; CurrWord = "";
            //checks for data
            //checks for VRML edition
            if (Back1Word.equalsIgnoreCase("V2.0") &&
            Back2Word.equalsIgnoreCase("VRML")) {JOptionPane.showMessageDialog(null,"file VRML
            version is: VRML 2.0");}
            if (Back1Word.equalsIgnoreCase("V1.0") &&
            Back2Word.equalsIgnoreCase("VRML")) {JOptionPane.showMessageDialog(null,"file VRML
            version is: VRML 1.0");}
            if (Back1Word.equalsIgnoreCase("V1.1") &&
            Back2Word.equalsIgnoreCase("VRML")) {JOptionPane.showMessageDialog(null,"file VRML
            version is: VRML 1.1");}
            //checks for Joints
            if (Back3Word.equals("DEF") &&
            Back1Word.equals("Joint")) {JointCount++;}
            if (Back3Word.equals("DEF") &&
            Back1Word.equals("Segment")) {SegmentCount++;}
            if (Back3Word.equals("DEF") &&
            Back1Word.equals("Humanoid")) {HumanoidCount++;}
            if (Back3Word.equals("DEF") && Back1Word.equals("Site")) {SiteCount++;}
            if (Back3Word.equals("DEF") &&
            Back1Word.equals("Displacer")) {DisplacerCount++;}
            //checks for ROUTES
            if (Back4Word.equals("ROUTE") && Back2Word.equals("TO")) {ROUTEcount++;}
            if (Back4Word.equals("ROUTE") &&
            !Back2Word.equals("TO")) {JOptionPane.showMessageDialog(null,"Encountered ROUTE without
            corresponding TO in line " + LineCount);}
            if (Back3Word.equals("PROTO")) {PROTOcount++;}
            if (Back3Word.equals("PROTO") &&
            Back2Word.equals("Joint")) {JointPROTOdeclaration = true;}
            if (Back3Word.equals("PROTO") &&
            Back2Word.equals("Segment")) {SegmentPROTOdeclaration = true;}
            if (Back3Word.equals("PROTO") &&
            Back2Word.equals("Site")) {SitePROTOdeclaration = true;}
            if (Back3Word.equals("PROTO") &&
            Back2Word.equals("Displacer")) {DisplacerPROTOdeclaration = true;}
            if (Back3Word.equals("PROTO") &&
            Back2Word.equals("Humanoid")) {HumanoidPROTOdeclaration = true;}
        }
        }
        reader.close();
    } catch (IOException e) {
        JOptionPane.showMessageDialog(null,"File contains " + LineCount + " lines,
        " + CharCount + " characters total.");
        //calculate Level of Articulation
        int LevelOA = -1;
        if (JointCount == 1){LevelOA = 0;} if (JointCount > 1 && JointCount < 19
        ){LevelOA = 1;}
        if (JointCount > 18 && JointCount < 72 ){LevelOA = 2;}if (JointCount > 71
        ){LevelOA = 3;}
        if (JointCount >
        SegmentCount){JOptionPane.showMessageDialog(null,"Encountered " + (JointCount -
        SegmentCount) + " more Joints than Segments while parsing the file. It is possible
        that one or more Joints has no corresponding Segment.");}
        if (SegmentCount >
        JointCount){JOptionPane.showMessageDialog(null,"Encountered " + (SegmentCount -
        JointCount) + " more Segment declarations than their corresponding Joints. It is
        possible that there are stray Segment declarations.");}
        if (HumanoidCount < 2)
        {JOptionPane.showMessageDialog(null,"Number of Joints: " + JointCount +
        "\nNumber of Segments: " + SegmentCount + "\nEstimated Level of Articulation: " +
        LevelOA + "\nNumber of Humanoids: "+ HumanoidCount);}
        if (LevelOA == -1)
        {JOptionPane.showMessageDialog(null,"File does not seem to be H-Anim
        Compliant.");}
        else
        {
            JOptionPane.showMessageDialog(null,"PROTO declarations: " + PROTOcount +
            "\n    Joint Proto declaration: " + JointPROTOdeclaration
            + "\n    Segment Proto declaration: " + SegmentPROTOdeclaration
            + "\n    Site Proto declaration: " + SitePROTOdeclaration
            + "\n    Displacer Proto declaration: " + DisplacerPROTOdeclaration

```

```

+ "\n    Humanoid Proto declaration: " + HumanoidPROTOdeclaration);
    if (JointPROTOdeclaration ==
false){JOptionPane.showMessageDialog(null,"Encountered Joint(s) without the
declaration of the Joint PROTO. This could be the cause of an error for many
browsers.");}
        if (SegmentPROTOdeclaration == false && SegmentCount >
0){JOptionPane.showMessageDialog(null,"Encountered Segment(s) without the declaration
of the Segment PROTO. This could be the cause of an error for many browsers.");}
            if (HumanoidPROTOdeclaration == false && HumanoidCount >
0){JOptionPane.showMessageDialog(null,"Encountered Humanoid(s) without the declaration
of the Humanoid PROTO. This could be the cause of an error for many browsers.");}
                if (SitePROTOdeclaration == false && SiteCount >
0){JOptionPane.showMessageDialog(null,"Encountered Site(s) without the declaration of
the Site PROTO. This could be the cause of an error for many browsers.");}
                    if (DisplacerPROTOdeclaration == false && DisplacerCount >
0){JOptionPane.showMessageDialog(null,"Encountered Displacer(s) without the
declaration of the Displacer PROTO. This could be the cause of an error for many
browsers.");}
                }
            if (JointCount > 0 && HumanoidCount ==
0){JOptionPane.showMessageDialog(null,"Encountered: " + JointCount + " Joints but no
Humanoid declaration!");}

        //parse the file again! this time check for syntax errors.
        LineCount=0; int depthopened = 0; boolean depthclosed = true; int
errorlinestart=0;
        int s_depthopened=0; boolean s_depthclosed = true; int s_errorlinestart=0;
        try {
            file2 = new FileReader(URLfrom);
            BufferedReader reader = new BufferedReader(file2);
            while (iread == true)
            {
                if (! reader.ready()) break;
                c = (char) reader.read();
                //controls
                if (c == '\n'){LineCount++;}else{CharCount++;}
                //construct words
                if (c == '#')
                {
                    boolean jread = true;
                    //ignore everything until \n, since this is comments.
                    while (jread == true)
                    {
                        if (! reader.ready()) break;
                        c = (char) reader.read();
                        if (c == '\n'){jread = false;}
                    }
                }
                if ((c > 47 && c < 58 || c > 63 && c < 91 || c > 93 && c < 123 || c
> 41 && c < 47)&& c != '#')
                {CurrWord = CurrWord + c;}
                else
                {
                    backstrip = c;
                    Back4Word = Back3Word; Back3Word = Back2Word; Back2Word = Back1Word;
                    Back1Word = CurrWord; CurrWord = "";
                    if (backstrip == '{')
                    {
                        depthopened = depthopened + 1;
                        if (depthopened == 1 && depthclosed == true){depthclosed = false;
errorlinestart = LineCount; Stack_push(Back1Word);}
                    }
                    if (backstrip == '}'){
                        depthopened = depthopened - 1;
                        if (depthopened ==0){depthclosed = true;}
                    }
                    if (depthopened < 0){JOptionPane.showMessageDialog(null,"Encountered
stray '}' character in line: #" + LineCount + ". This could possibly mean a syntax
error.");iread=false;break;}

                    if (backstrip == '[')
                    {
                        s_depthopened = s_depthopened + 1;
                        if (s_depthopened == 1 && s_depthclosed == true){s_depthclosed = false;
s_errorlinestart = LineCount; Stack_push(Back1Word);}
                    }
                }
            }
        }
    }
}

```

```

        if (backstrip == ']') {
            s_depthopened = s_depthopened - 1;
            if (s_depthopened == 0) {s_depthclosed = true;}
        }
        if (s_depthopened < 0) {JOptionPane.showMessageDialog(null, "Encountered
stray ']' character in line: #" + LineCount + ". This could possibly mean a syntax
error."); iread=false; break;}
    }
    reader.close();
} catch (IOException e) {}
if (depthclosed == false)
{
    JOptionPane.showMessageDialog(null, "Encountered more '[' than ']'
characters while parsing file. This could probably mean a syntax error starting with
the first '[' at line:" + errorlinestart + ".");
}
if (s_depthclosed == false)
{
    JOptionPane.showMessageDialog(null, "Encountered more '[' than ']'
characters while parsing file. This could probably mean a syntax error starting with
the first '[' at line:" + s_errorlinestart + ".");
}
}
else {
    JOptionPane.showMessageDialog(null, "Action Cancelled.");
}
}

```

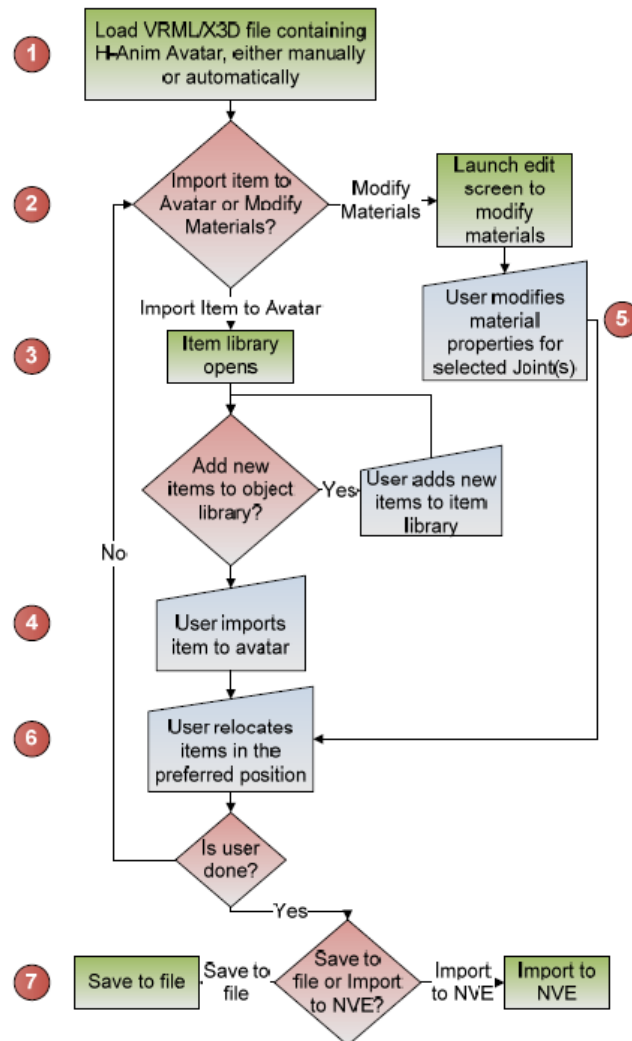
Η λογική του παραπάνω κώδικα συνοψίζεται στην ανάγνωση του επιθυμητού αρχείου χαρακτήρα-προς-χαρακτήρα, και στην χρησιμοποίηση 4 μεταβλητών-«λέξεων» που αντιστοιχούν στην τρέχουσα, την προηγούμενη, την προ-προηγούμενη και την 2-θέσεις πίσω από την τρέχουσα λέξη που διαβάζει ο κώδικας.

Κάθε φορά που ολοκληρώνεται η ανάγνωση μιας λέξης, ο κώδικας «σπρώχνει» με την μορφή στοίβας την καινούρια λέξη ως τρέχουσα, την παλιά τρέχουσα ως προηγούμενη κλπ. Κάθε φορά που αναγνωρίζεται κάποια σημαντική λέξη ως τρέχουσα, πχ η λέξη «Segment», εξετάζεται η προηγούμενή τους και η προ-προηγούμενη για να διερευνηθεί ο λόγος αναφοράς της λέξης. Με τη λογική των συγκρίσεων, ο κώδικας εξάγει κάποια συμπεράσματα σχετικά με την δομή του αρχείου, πχ τον αριθμό των Segments, Joints και Displacers που έχουν χρησιμοποιηθεί.

Τέλος, άξιος αναφορά είναι ο μηχανισμός εξακρίβωσης για το αν υπάρχει πρόβλημα σύνταξης που να αφορά αγκύλες '{','}', '[' ,']'. Κάθε φορά που ο χρήστης συναντά έναν χαρακτήρα με το οποίο ανοίγει κάποιο κομμάτι κώδικα που σηματοδοτείται από τους χαρακτήρες '{' ή '[', ο VRMLparser χρησιμοποιεί μια μετρητή δείκτη η οποία αυξομειώνεται τόσες φορές όσες οι χαρακτήρες που συναντά. Αν ο δείκτης αυτός πάρει κάποια φορά αρνητική τιμή, τότε υπάρχει κάποιος παραπληθισμός χαρακτήρα ή κάποιος χαρακτήρας που θα πρεπε να υπάρχει και έχει παραλειφθεί.

## 6.3 Ο ΜΗΧΑΝΙΣΜΟΣ ΤΟΥ H-ANIME

Σε αυτή την ενότητα ρίχνουμε μια λεπτομερή ματιά στον μηχανισμό που παρέχει το H-AnimE στον χρήστη για την επεξεργασία των avatars. Ο μηχανισμός μπορεί να συνοψιστεί σε 7 βασικά σημεία, όπως φαίνεται και από το παρακάτω διάγραμμα ροής (flow chart):



**Εικόνα 40:** ο μηχανισμός του H-AnimE

Επεξηγούμε αναλυτικά το κάθε βήμα:

- 1) Φόρτωση ενός VRML/X3D αρχείου το οποίο περιέχει κάποιο H-Anim συμβατό Avatar, είτε χειροκίνητα από τον χρήστη, είτε χρησιμοποιώντας την λειτουργία προφόρτωσης του εργαλείου.
- 2) Στο 2<sup>ο</sup> βήμα, ο χρήστης επιλέγει είτε να εισάγει κάποιο γεωμετρικό αντικείμενο στο avatar, είτε απλώς να τροποποιήσει τα χαρακτηριστικά εμφάνισης του avatar. Στην πρώτη επιλογή, ο χρήστης μεταβαίνει στο βήμα 3, ενώ στην δεύτερη επιλογή προβάλλεται απευθείας η οθόνη επεξεργασίας του avatar και ο χρήστης μεταβαίνει απευθείας στο βήμα 5.

- 3) Στο 3<sup>ο</sup> βήμα, ο χρήστης βλέπει στην οθόνη του την βιβλιοθήκη γεωμετρικών αντικειμένων από την οποία μπορεί είτε να επιλέξει κάποιο αντικείμενο για εισαγωγή στο avatar, είτε να τροποποιήσει την βιβλιοθήκη (διαγραφή, εισαγωγή αντικειμένου κλπ). Αν επιλέξει κάποιο αντικείμενο για εισαγωγή στο avatar, μεταβαίνει στο βήμα 4.
- 4) Στο 4<sup>ο</sup> βήμα, ο χρήστης εισάγει το αντικείμενο στο avatar. Και τα δυο μοντέλα προβάλλονται στην οθόνη. Το επόμενο βήμα είναι είτε το βήμα 5 και κατ'όπιν το 6, είτε απευθείας το 6.
- 5) Στο 5<sup>ο</sup> βήμα, ο χρήστης μπορεί να επιλέξει να τροποποιήσει τα χρώματα ή τα textures κάποιου μέλους του σώματος του avatar. Το βήμα αυτό δεν αφορά την εισαγωγή καινούριας γεωμετρίας στο avatar.
- 6) Στο 6<sup>ο</sup> βήμα, ο χρήστης «ταιριάζει» το αντικείμενο πάνω στο avatar. Επιλέγει σε ποιο Joint θα προσαρτηθεί, και κατ'όπιν επιλέγει κάποιο από τα εξής: α) να αντικαταστήσει ολοκληρωτικά την γεωμετρία του Joint με το καινούριο αντικείμενο β) να προσαρτήσει το καινούριο αντικείμενο στην παλιά γεωμετρία, διατηρώντας την γ) να προσαρτήσει το αντικείμενο ως καινούριο Joint, υποθέτοντας ότι έχει επιλέξει κάποιο Joint που δεν χρησιμοποιείται ήδη και ότι ο πατέρας-Joint χρησιμοποιείται.
- 7) Στο 7<sup>ο</sup> βήμα, ο χρήστης μπορεί να επιλέξει είτε να αποθηκεύσει το avatar σε κάποιο αρχείο, είτε να μεταβεί απευθείας σε κάποιο NVE το οποίο θα αναλάμβανε να «διαβάσει» από την σκηνή που έχει φορτώσει ο Xj3D browser της εφαρμογής όλα τα απαραίτητα στοιχεία.

## 6.4 ΑΛΓΟΡΙΘΜΙΚΑ ΘΕΜΑΤΑ

### 6.4.1 Η Χρήση του Πακέτου Xj3D

Η ανάπτυξη του editor έγινε κατά κύριο λόγο με χρήση του ελεύθερου πακέτου Xj3D 2.0. Το Xj3D παρέχει βιβλιοθήκες για την Java με κλάσεις και εργαλεία για την δημιουργία στιγμιοτύπων που να αναπαριστούν VRML/X3D κόμβους, καθώς και δυνατότητες οπτικής προβολής του VRML περιεχομένου στην οθόνη.

Η χρήση του Xj3D μπορεί να συνοψιστεί στα εξής:

- α) Υλοποίηση αντικειμένου Φυλλομετρητή (Xj3D browser) σε κάθε οθόνη όπου χρειάζεται για να παρέχει επεξεργασία VRML περιεχομένου.
- β) Φόρτωση ενός αρχείου VRML/X3D στην οθόνη, και εξαγωγή από αυτή ενός αντικειμένου X3DScene.
- γ) Ένα αντικείμενο X3DScene περιέχει όλους τους κόμβους, πλήρως ιεραρχημένους μαζί με όλα τα χαρακτηριστικά τους. Από την εξαγωγή των στοιχείων/χαρακτηριστικών ενός κόμβου προκύπτουν μεταβλητές που χρησιμοποιούν καθιερωμένους τύπους δεδομένων, πχ η γεωμετρική αναπαράσταση ενός IndexedFaceSet που χρησιμοποιεί αριθμούς κινητής υποδιαστολής για τον προσδιορισμό των κορυφών επιστρέφει ένα διάνυσμα-μητρώο από τιμές Float.
- δ) Το αντικείμενο X3DScene δεν χρειάζεται ενημέρωση για αλλαγές. Αυτές γίνονται αυτόματα μετά από κάθε επεξεργασία κόμβου, και προβάλλονται σχεδόν ακαριαία στον φυλλομετρητή που προβάλλει την X3DScene.

Παρέχονται και κάποιες επιπλέον λειτουργίες για κάθε τύπο δεδομένων που περιλαμβάνει το Xj3D πχ. για η προσμέτρηση απογόνων-κόμβων, η απομάκρυνση απογόνων-κόμβων κλπ.

Παράδειγμα εξαγωγής παραμέτρων από κόμβο:

```
SFNode Appear = (SFNode) ShapeToGet.getField("appearance");
X3DNode AppearSub = (X3DNode) Appear.getValue();
boolean e6nullbool=false;
SFNode textureGot = (SFNode) AppearSub.getField("texture");
X3DNode textureContent = (X3DNode) textureGot.getValue();
```

## 6.4.2 Φόρτωση – Αποθήκευση σε Αρχείο

Το Xj3D παρέχει λειτουργίες φόρτωσης μιας VRML/X3D σκηνής στον φυλλομετρητή αυτόματα. Η φόρτωση, κατά συνέπεια, δεν χρειάστηκε κάποιον επιπλέον κώδικα ο οποίος να διαβάζει script του αρχείου που θέλουμε να φορτώσουμε. Κάτι τέτοιο ωστόσο έχει υλοποιηθεί για το εργαλείο του VRMLparser.

Η αποθήκευση είναι μια μακράν πιο πολύπλοκη διαδικασία, δεδομένου ότι δεν την παρέχει το Xj3D, και υλοποιείται σε φάσεις. Ακριβώς επειδή υπάρχουν προβλήματα ανάγνωσης των παραμέτρων των κόμβων (ενότητα 6.4.3) λόγω εγγενών αδυναμιών του Xj3D, η αποθήκευση προυποθέτει την προ-ύπαρξη του αρχείου του avatar, δηλαδή δεν είναι δυνατή η δημιουργία ενός αρχείου από μηδενική βάση. Στο προ-υπάρχον αρχείο, ο H-AnimE επεμβαίνει με χρήση βιβλιοθηκών BufferedReader και BufferedWriter, διαβάζοντας έναν-έναν χαρακτήρα και εγγράφοντάς τους σε κάποιο προσωρινό αρχείο. Στα επιθυμητά σημεία, ο κώδικας επεμβαίνει και προσθέτει τους επιπλέον ASCII χαρακτήρες που αναπαριστούν τις αλλαγές σε μορφή script. Σε ορισμένες περιπτώσεις χρειάζονται κάποιοι χαρακτήρες (πχ, ο χαρακτήρες '[' που κλείνει το περιεχόμενο του χαρακτήρα '[') να παραλειφθούν. Η παραπάνω εργασία επαναλαμβάνονται όσες φορές επιτάσσουν οι αλλαγές του χρήστη.

## 6.4.3 Βιβλιοθήκες

Για τον H-AnimE χρησιμοποιήθηκαν οι εξής βιβλιοθήκες:

```
import java.awt.*;
import java.util.HashMap;
import javax.swing.*;
import java.io.*;
import java.awt.*;
import java.awt.event.*;
import java.awt.image.BufferedImage;
import java.io.IOException;
import vrml.field.*;
import javax.swing.event.ChangeEvent;
import javax.swing.event.ChangeListener;
import java.awt.event.ItemEvent;
import java.awt.event.ItemListener;
import java.lang.Math.*;
import vrml.node.*;
```

```
import vrml.*;
import org.web3d.vrml.lang.NodeException;
import org.web3d.vrml.scripting.external.sai.SAINode;
import org.web3d.x3d.sai.*;
import javax.swing.ImageIcon;
import javax.swing.event.ChangeListener;
import javax.swing.filechooser.*;
```

Ο κώδικας του H-AnimE ανέρχεται στις 3641 γραμμές.

#### 6.4.4 Προβλήματα, δυσκολίες και ελλείψεις του Xj3D

Τα περισσότερα προβλήματα που παρατηρήθηκαν στην πράξη οφείλονται κυρίως στην αστάθεια του πακέτου Xj3D 2.0 και σε διάφορες ελλείψεις του που παρατηρήθηκαν στην πορεία. Συγκεκριμένα, ενώ οι κατασκευαστές του Xj3D έριξαν το βάρος στην ανάπτυξη ενός σταθερού φυλλομετρητή για VRML που να μπορεί να ενσωματωθεί σε εφαρμογές Java, δεν έδωσαν καθόλου σημασία στο να μπορεί ο φυλλομετρητής να αποθηκεύει τυχόν αλλαγές στην σκηνή πίσω στο αρχείο από το οποίο διάβαζε αρχικά. Ενώ παρέχονται πολλές λειτουργίες με τις οποίες ο κώδικας Java θα διάβαζε και θα επεξεργαζόταν VRML/X3D υλικό, πιθανότατα δημιουργώντας το από μηδενική βάση, τα παραπάνω φαίνονται κάπως περιττά από την στιγμή που δεν υπάρχει τρόπος να αποθηκευτούν. Φαίνεται σχεδόν σαν καθήκον του εκάστοτε προγραμματιστή να αναπτύξει δικό του κώδικα που να αποθηκεύει το script σαν «ακατέργαστο» ASCII αρχείο κειμένου.

Επιπλέον, παρατηρήθηκαν προβλήματα σχετικά με τις επεμβάσεις στην ιεραρχία των κόμβων. Συγκεκριμένα, κατά την επέκταση κόμβων με μηδενικό αριθμό εγγραφών στο πεδίο “children” (δηλαδή χωρίς απογόνους), η αλλαγή θα φαινόταν στην οθόνη του φυλλομετρητή ωστόσο θα ήταν σαν να μην είχε συντελεστεί ποτέ αν κάποιος επιχειρούσε να διαβάσει πίσω από τις νέες τιμές που έχει εγγράψει. Το παραπάνω είναι ιδιαίτερα απαγορευτικό για επεξεργασία μεγάλης κλίμακας, από την στιγμή που δεν παρέχεται κάποια λειτουργία αποθήκευσης, όπως προαναφέρθηκε, η οποία θα αναλάμβανε να κάνει αυτόματα την δύσκολη δουλειά, και είναι δύσκολο για τον προγραμματιστή να βάλει τον κώδικά του να αποθηκεύσει πληροφορίες τις οποίες σχεδόν δεν μπορεί να διαβάσει.

Στην περίπτωση του H-AnimE, τα παραπάνω προβλήματα αντιμετωπίστηκαν χρησιμοποιώντας βοηθητικούς πίνακες στους οποίους αποθηκεύονται οι αλλαγές που κάνει ο χρήστης καθώς επεξεργάζεται το avatar του, καθώς και με την ανάπτυξη ανεξάρτητου από το Xj3D κώδικα για την αποθήκευση του avatar σε μορφή script.

Επιπλέον παρατηρήθηκαν κάποια προβλήματα ολοκληρωτικής ανανέωσης μιας σκηνής σε κάποιο φυλλομετρητή, στον οποίο έχει φορτωθεί κάποια σκηνή ήδη. Για το παραπάνω πρόβλημα, αν και ζητήθηκε τεχνική υποστήριξη από το Web3D Consortium, η μόνη λύση που βρέθηκε είναι η αποθήκευση του αρχείου και η φόρτωση του από τον φυλλομετρητή αντί για την ανανέωση της σκηνής του φυλλομετρητή.

Κάποια επιπλέον προβλήματα που ίσως θα πρέπει να αναφερθούν είναι run-time ζητήματα της Java, προβλήματα σταθερότητας του Xj3D, καθώς και κάποιες

αδυναμίες του Xj3D να φορτώσει VRML/X3D σκηνές οι οποίες είναι καθ'όλα έγκυρες. Τα τελευταία ωστόσο δεν δημιούργησαν προβλήματα στον κώδικα του H-AnimE.

## **6.5 ΣΥΜΠΕΡΑΣΜΑΤΑ**

Στην ενότητα αυτή καλύφθηκε ο editor H-AnimE, οι σχεδιαστικοί του στόχοι και το πώς καλύπτονται από τις λειτουργίες που προσφέρει. Επίσης καλύφθηκαν ζητήματα σχετικά με τον μηχανισμό του H-AnimE, καθώς και θέματα που αφορούν τον κώδικα του προγράμματος και το πακέτο Xj3D 2.0.



## ΚΕΦΑΛΑΙΟ 7: ΣΥΜΠΕΡΑΣΜΑΤΑ



---

## ΚΕΦΑΛΑΙΟ 7 - ΣΥΜΠΕΡΑΣΜΑΤΑ

---

### 7.1 ΣΥΜΠΕΡΑΣΜΑΤΑ

Στόχος της παρούσας διπλωματικής εργασίας ήταν η επέκταση του παλαιότερου framework υποστήριξης της πλατφόρμας Εικονικών Διαδικτυακών Περιβαλλόντων, EVE-II, με ακόμη περισσότερες λειτουργικότητες που θα ενισχύσουν τον ρόλο της τόσο σαν εκπαιδευτικό περιβάλλον όσο και σαν μια πολύ-χρηστική πλατφόρμα γενικού σκοπού.

Η αρχική ιδέα της επέκτασης του παλαιότερου framework οδήγησε στην δημιουργία μιας ξεχωριστής και ανεξάρτητης εφαρμογής Java, η οποία παρέχει εύχρηστους, φιλικούς προς τον χρήστη και ευέλικτους μηχανισμούς για την επεξεργασία των avatars. Εξετάζοντας συνολικά την εφαρμογή H-AnimE και την πλατφόρμα EVE-II, διαπιστώνουμε ότι πλέον παρέχεται στον μέσο χρήστη ένα ελκυστικό και ανεξάρτητο από άλλες εφαρμογές πακέτο για εφαρμογές Εικονικής Πραγματικότητας πάνω στον Παγκόσμιο Ιστό.

Οι λειτουργίες που παρέχονται από τον H-AnimE σχεδιάστηκαν, όπως ήδη έχει προαναφερθεί, με γνώμονα τις εξής απαιτήσεις:

- Ο editor να είναι απλός στην χρήση και φιλικός προς τον χρήστη.
- Οι λειτουργίες του να απευθύνονται στον μέσο χρήστη και όχι στον σχεδιαστή/καλλιτέχνη 3D υλικού.
- Οι λειτουργίες που παρέχονται να μην χρειάζονται άλλες εφαρμογές για να παρέχουν ολοκληρωμένο αποτέλεσμα.
- Ο editor να απομακρύνει τον χρήστη από την συγγραφή VRML κώδικα.
- Οι δυνατότητες του editor για ικανοποιητικές αισθητικές επεμβάσεις στην γεωμετρία να είναι επεκτάσιμες από τον χρήστη.

Ο editor ανταποκρίνεται στα παραπάνω, ως εξής:

- Στον χρήστη παρέχονται πλήρεις διεπαφές, με οπτικό έλεγχο και ανάδραση. Σε μεγάλο βαθμό, ο χειρισμός του editor περιορίζεται στην χρήση ορισμένων μόνο στοιχείων ελέγχου (κουμπιών κλπ) της διεπαφής, των οποίων η λειτουργία είναι αυτονόητη και έχει άμεση οπτική ανάδραση. Ο αριθμός των στοιχείων ελέγχου είναι μικρός.
- Ο editor το μόνο που προαπαιτεί είναι κάποια βασικά γεωμετρικά αντικείμενα (πχ καπέλα) τα οποία ο χρήστης δεν χρειάζεται να γνωρίζει πώς να τα σχεδιάζει, αλλά του διατίθενται ελεύθερα στο διαδίκτυο και σε κοινότητες σχετικές με το X3D. Ο συνολικός σχεδιασμός του editor θα δυσκόλευε μάλλον τον σχεδιαστή 3D υλικού, ο οποίος είναι συνηθισμένος στο να δημιουργεί avatars καθορίζοντας την γεωμετρία τους.
- Οι λειτουργίες που παρέχονται ανταποκρίνονται πλήρως στις απαιτήσεις που θα μπορούσε να έχει ο μέσος χρήστης πριν εισάγει το avatar του σε κάποιο Δικτυακό Εικονικό Περιβάλλον. Παρέχονται μηχανισμοί με τους οποίους ο χρήστης μπορεί να αλλάξει πλήρως την εμφάνιση του avatar, να επεκτείνει την ιεραρχία με περισσότερα Joints κλπ. Ένα βήμα παραπέρα θα ήταν η προσομοίωση κίνησης, την οποία ωστόσο αφορά το προηγούμενο framework

υποστήριξης της EVE-II, και για το οποίο υπάρχουν αρκετοί διαθέσιμοι editors.

- Ο editor επιτρέπει φόρτωση/αποθήκευση/επεξεργασία χωρίς να εμπλέξει τον χρήστη στην συγγραφή κώδικα.
- Η βιβλιοθήκη αντικειμένων είναι δυναμική, κατά συνέπεια ο χρήστης μπορεί να προσθέτει και να αφαιρεί γεωμετρικά αντικείμενα και να εμπλουτίζει την συλλογή του. Μετά την παρέλευση κάποιου διαστήματος εξοικείωσης, ο χρήστης λογικά θα έχει επεκτείνει την συλλογή του καθώς και την κρίση του πάνω σε θέματα αισθητικής ώστε να πετυχαίνει καλύτερες αισθητικά επεμβάσεις πάνω στο avatar του. Το παραπάνω είναι συνηθισμένο για όλα τα προγράμματα που αφορούν την δημιουργία 3D υλικού.

Στο αμέσως επόμενο Κεφάλαιο παρουσιάζουμε κάποιες πιθανές βελτιώσεις που θα μπορούσαν να γίνουν πάνω στην εφαρμογή H-AnimE, καθώς και κάποια πεδία που προσφέρονται για περαιτέρω δουλειά και τα οποία ο editor δεν καλύπτει πλήρως.

Συμπερασματικά, ενώ υπάρχει ακόμα περιθώριο για την παροχή περισσότερων και ακόμη πιο εξειδικευμένων λειτουργιών, ο H-AnimE αναγνωρίζει τις ολοένα και αυξανόμενες απαιτήσεις των χρηστών από τα Δικτυακά Εικονικά Περιβάλλοντα, και ανταποκρίνεται σε αυτές του μέσου χρήστη.

ΚΕΦΑΛΑΙΟ 8: ΜΕΛΛΟΝΤΙΚΗ  
ΕΡΓΑΣΙΑ



## ΜΕΛΛΟΝΤΙΚΗ ΕΡΓΑΣΙΑ

Πιθανές επεκτάσεις πάνω στον editor H-AnimE θα μπορούσαν να αποτελέσουν αντικείμενο μελλοντικής εργασίας. Συγκεκριμένα, ο H-AnimE θα μπορούσε να εμπλουτιστεί περαιτέρω με εργαλεία που να επιτρέπουν την δημιουργία X3D περιεχομένου από μηδενική βάση, καθώς και με εργαλεία που να επιτρέπουν επεμβάσεις υπάρχουσας γεωμετρίας σε επίπεδο mesh.

### 8.1 ΔΗΜΙΟΥΡΓΙΑ X3D ΠΕΡΙΕΧΟΜΕΝΟΥ ΑΠΟ ΜΗΔΕΝΙΚΗ ΒΑΣΗ

Πρόκειται για μια λειτουργία η οποία οριακά υποστηρίζεται από τον H-AnimE στην τρέχουσα εκδόσή του. Θα μπορούσε κάποιος να αξιοποιήσει κάποιο έτοιμο script με μηδενικό επίπεδο άρθρωσης (LoA-0) και με χρήση του εργαλείου Expand Joint που διατίθεται από τον H-AnimE να δημιουργήσει ένα avatar από έτοιμα meshes. Το παραπάνω ωστόσο θα ήταν εξαιρετικά χρονοβόρο, και δύσκολα θα παρήγαγε ένα αισθητικά ελκυστικό αποτέλεσμα.

Θα ήταν ενδιαφέρουσα μια επέκταση των δυνατοτήτων του H-AnimE ώστε να υποστηρίζει αποτελεσματικά την παραπάνω διαδικασία. Ωστόσο το παραπάνω θα απαιτούσε κάποια μορφή «σύμβασης» με κάποιο άλλο πρόγραμμα το οποίο θα χρησιμοποιείται για την παραγωγή του X3D μοντέλου (που δεν θα υποστηρίζει ακόμα το H-Anim πρότυπο), ώστε ο χρήστης να απομακρύνεται από τον ορισμό των Joints ένα-προς-ένα. Το πρόγραμμα θα αναλάμβανε να υπολογίζει αυτόματα τα νέα κέντρα των Joints (όπως δουλεύει τώρα η λειτουργία Expand Joint), ενώ θα πρέπει να υπάρξει και κάποια παραμετροποίηση στην «σύμβαση» ώστε το πρόγραμμα να καταλαβαίνει αυτόματα ποια αντικείμενα θα αποτελέσουν την γεωμετρία του κάθε κόμβου.

### 8.2 ΕΠΕΚΤΑΣΗ ΤΗΣ ΒΙΒΛΙΟΘΗΚΗΣ ΑΝΤΙΚΕΙΜΕΝΩΝ

Μια ενδιαφέρουσα και άμεσα εφικτή βελτίωση στον editor θα ήταν η προσθήκη μιας μεγαλύτερης γκάμας αντικειμένων διαθέσιμα στον χρήστη, πχ μαλλιά, ρούχα, παπούτσια, ακρυλικά νύχια, φακοί επαφής, γυαλιά ηλίου κλπ. Καθώς η βιβλιοθήκη έχει απεριόριστες δυνατότητες για προσθήκη περισσότερων αντικειμένων, το παραπάνω δεν χρειάζεται κάποιος που να γνωρίζει τον κώδικα του H-AnimE για να το καταφέρει. Ωστόσο, αυτό που θα ήταν ενδιαφέρον από πλευράς κώδικα, θα ήταν να επιτρέπονται κάποιες παραμετροποιήσεις πάνω στα αντικείμενα, πχ κύρτωση στο κατω μέρος της περούκας, πτυχώσεις στα ρούχα κλπ.

### 8.3 ΕΠΕΜΒΑΣΕΙΣ ΣΤΗ ΓΕΩΜΕΤΡΙΑ ΣΕ ΕΠΙΠΕΔΟ MESH

Η ενσωμάτωση ενός τέτοιου εργαλείου θα εκτόξευε τις δυνατότητες του H-AnimE για σχεδιασμό avatars, ωστόσο κρίνεται δύσκολη. Θα ήταν ενδιαφέρον να παρέχεται στον χρήστη κάποιο εργαλείο με το οποίο να μπορεί να σχεδιάσει τα πολύγωνα που θα αποτελέσουν την γεωμετρία για τα Joints και Segments του avatar, και εν συνεχεία

να χρησιμοποιήσει τις ήδη υπάρχουσες δυνατότητες του H-AnimE για αλλαγή χρωμάτων και texturing. Επίσης, να μπορεί να αλλάξει την γεωμετρία ήδη υπάρχοντων κόμβων σε avatar το οποίο φορτώνει.

Επιπρόσθετα θα ήταν ενδιαφέρον να παρέχεται στον χρήστη ένα σύνολο από εργαλεία που θα επέμβαιναν στην «αισθητική» του avatar, για παράδειγμα, κάποιο εργαλείο με το οποίο ο χρήστης θα μπορούσε να κάνει το avatar να φαίνεται πιο αδύνατο ή πιο παχουλό κλπ. Το παραπάνω ίσως και να βρίσκεται έξω από τις δυνατότητες εμπορικών editors που έχουν καθιερωθεί και να ήταν μια πραγματική πρόκληση για όποιον το επιχειρούσε.

## **8.4 ΕΠΕΚΤΑΣΗ ΤΩΝ ΔΥΝΑΤΟΤΗΤΩΝ ΤΟΥ ΕΡΓΑΛΕΙΟΥ VRMLPARSER**

Ο VRMLParser στην παρούσα φάση του είναι εξαιρετικά αποδοτικός και ευέλικτος ως προς τις δυνατότητες που παρέχει, ωστόσο περιορίζεται σε συντακτικά ζητήματα και στον εντοπισμό οφθαλμοφανών λαθών τα οποία πιθανότατα ο ίδιος ο χρήστης με κάποιο κόπο θα έβρισκε. Θα ήταν ενδιαφέρον να επεκταθούν οι δυνατότητές του περαιτέρω σε θέματα που να αφορούν την εσωτερική δομή του avatar, πχ στον εντοπισμό λογικών ανακολουθιών, στην παρουσία ανεπιθύμητων/περιττών κόμβων μέσα στο αρχείο ή σε θέματα γεωμετρίας που θα μπορούσαν να απλουστευτούν. Τα παραπάνω προσφέρονται μόνο σε περιορισμένο βαθμό στην παρούσα φάση, ωστόσο θα ήταν εξαιρετικά ενδιαφέρον να αναπτυχθούν και να επεκταθούν, ώστε το συνολικό εργαλείο που θα προκύψει να είναι εξαιρετικά ισχυρό και χρήσιμο.



ΠΑΡΑΡΤΗΜΑ Ι: ΒΙΒΛΙΟΓΡΑΦΙΚΕ  
Σ ΑΝΑΦΟΡΕΣ



---

## ΒΙΒΛΙΟΓΡΑΦΙΑ

---

### 8.5 ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] Βικιπαιδεία, Ελεύθερη Εγκυκλοπαίδεια. <http://en.wikipedia.org/wiki/VRML>
- [2] Βικιπαιδεία, Ελεύθερη Εγκυκλοπαίδεια. <http://en.wikipedia.org/wiki/X3D>
- [3] **Pesce, Mark**. Ιστοσελίδα <http://www.3d-test.com/> Συνεντεύξεις: [http://www.3d-test.com/interviews/mediamachines\\_2.htm](http://www.3d-test.com/interviews/mediamachines_2.htm) 2004.
- [4] **Δ. Χαρίτος, Δ. Μαρτάκος**. Εισαγωγή στην Virtual Reality Modelling Language (VRML). <http://alexandra.di.uoa.gr/mmtech/VirtualReality/eisagvgh.pdf>
- [5] Web3D Consortium. <http://www.web3d.org/about/faq/>
- [6] Βικιπαιδεία, Ελεύθερη Εγκυκλοπαίδεια. [http://en.wikipedia.org/wiki/Virtual\\_Reality](http://en.wikipedia.org/wiki/Virtual_Reality)
- [7] **Μπούρας Χρήστος, Καθηγητής του Τμήματος Μηχ. Η/Υ & Πληροφορικής. Τηλεματική & Νέες Υπηρεσίες, σύγγραμμα μαθήματος.**
- [8] VRML History. <http://www-winfo.uni-siegen.de/vrmlHistory/docs/>
- [9] Virtual Reality Modelling Language (VRML). [http://www.it.uom.gr/project/MultimediaTechnologyNotes/extra/append9\\_3.htm](http://www.it.uom.gr/project/MultimediaTechnologyNotes/extra/append9_3.htm)
- [10] **Βασίλειος, Τριγλιανός**. *Υλοποίηση Πλατφόρμας για Δικτυακά Εικονικά Περιβάλλοντα με Υποστήριξη του Προτύπου X3D - Διπλωματική Εργασία.*
- [11] **Andrea L. Ames, David R. Nadeau and John L. Moreland**. *The VRML 2.0 Sourcebook*.
- [12] Βικιπαιδεία, Ελεύθερη Εγκυκλοπαίδεια. [http://en.wikipedia.org/wiki/Rendering\\_%28computer\\_graphics%29](http://en.wikipedia.org/wiki/Rendering_%28computer_graphics%29) .
- [13] **Mathworks**:  
[http://www.mathworks.com/access/helpdesk\\_r13/help/toolbox/vr/ch\\_int15.html](http://www.mathworks.com/access/helpdesk_r13/help/toolbox/vr/ch_int15.html)
- [14] Βικιπαιδεία, Ελεύθερη Εγκυκλοπαίδεια. [http://en.wikipedia.org/wiki/Wireframe\\_model](http://en.wikipedia.org/wiki/Wireframe_model)
- [15] **Σγουρός, Νικήτας**. Η γλώσσα VRML. [http://dtps.unipi.gr/files/notes/2006-2007/eksamino\\_5/grafika\\_ypologistikwn\\_kai\\_eikonikh\\_pragmatikothta/vrml.pdf](http://dtps.unipi.gr/files/notes/2006-2007/eksamino_5/grafika_ypologistikwn_kai_eikonikh_pragmatikothta/vrml.pdf)
- [16] 3D Translation, Rotation and Scaling. [www.it.nuigalway.ie/~sredfern/CT404/04.pdf](http://www.it.nuigalway.ie/~sredfern/CT404/04.pdf)
- [17] VRML Scene Graphs. [http://www.engineering.uiowa.edu/~ie\\_246/Lecture/VRMLSceneGraphs.ppt](http://www.engineering.uiowa.edu/~ie_246/Lecture/VRMLSceneGraphs.ppt)
- [18] ZDNet. <http://dictionary.zdnet.com/definition/graphics+pipeline.html>
- [19] X3DGraphics. <http://x3dgraphics.com/tools/browsers.php>
- [20] Web3D X3D XML Syntax . <http://hypermultimedia.com/x3d/x3dxmlsyntax/>
- [21] Understanding VRML and Creating VRML Objects. <http://docs.rinet.ru/EtoHTML/ch24.htm>

- [22] Texture Mapping. <http://wally.cs.iupui.edu/~aharris/n355/texture/texture.html>
- [23] VRML routing tutorial. <http://sim.di.uminho.pt/vrmltut/frmroute.htm>
- [24] **Daniel K. Schneider, Sylvere Martin-Michiellot.** VRML Primer and Tutorial. <http://tecfa.unige.ch/guides/vrml/vrmlman/node24.html>
- [25] Xj3D. <http://www.xj3d.org/>
- [26] AllExperts Encyclopedia. [http://en.allexperts.com/e/a/av/avatar\\_%28virtual\\_reality%29.htm](http://en.allexperts.com/e/a/av/avatar_%28virtual_reality%29.htm)
- [27] Second Life. <http://secondlife.com/whatis/avatar/?lang=en-US>
- [28] Free Avatars. [http://www.freeavatars.org/virtual\\_models.html?Modelling\\_Avatar](http://www.freeavatars.org/virtual_models.html?Modelling_Avatar)
- [29] **Patrick Salamin, Daniel Thalmann, Frederic Vexo.** The Benefits of Third-Person Perspective in Virtual and Augmented Reality. [infoscience.epfl.ch/record/99021/files/VRST\\_06.pdf](http://infoscience.epfl.ch/record/99021/files/VRST_06.pdf)
- [30] **Thalmann, Daniel.** The Role of Virtual Humans in Virtual Environment Technology and Interfaces. <http://www.cise.ufl.edu/~lok/teaching/dcvef05/papers/EU.NSF.PDF>
- [31] **Χατζηπρίμου, Κλεοπάτρα.** Αποτίμηση και Επέκταση των νέων λειτουργικότητων της πλατφόρμας EVE - Διπλωματική Εργασία.
- [32] **Katerina MANIA, Alan CHALMERS.** A Classification for User Embodiment in Collaborative Virtual Environments. <http://www.cs.bris.ac.uk/Publications/Papers/1000310.pdf>
- [33] **Vivek Rajan, Satheesh Subramanian, Damin Keenan, Andrew Johnson, Daniel Sandin, Thomas DeFanti.** A Realistic Video Avatar System for Networked Virtual Environments.
- [34] **Jeremy Bailenson, James Blascovich.** BERKSHIRE ENCYCLOPEDIA OF HUMAN-COMPUTER INTERACTION. *Virtual Human Interaction Lab.* <http://vhil.stanford.edu/pubs/2004/bailenson-avatars.pdf>
- [35] The Journal of American Enterprise Institute. <http://www.american.com/archive/2007/july-0707/virtual-reality-2018avatars2019-are-now-real-enough-to-be-sued>
- [36] **Jeremy Bailenson, Dan Merget, Ralph Schroeder.** The Effect of Behavioural Realism and Form Realism of Real-Time Avatar Faces on Verbal Disclosure, Nonverbal Disclosure, Emotion Recognition and Copresence in Dyadic Interaction.
- [37] **Nikos Sarris, Michael Strintzis.** 3D modeling and animation: synthesis and analysis techniques for the human body. <http://books.google.gr/>.
- [38] H-Anim 2.0 Specification. [http://h-anim.org/Specifications/H-Anim200x/ISO\\_IEC\\_FCD\\_19774/](http://h-anim.org/Specifications/H-Anim200x/ISO_IEC_FCD_19774/).
- [39] **Qiming Wang, Sandy Ressler.** A Tool Kit to Generate 3D Animated CAESAR Bodies. [ovrt.nist.gov/people/sressler/2005-DHM-10.pdf](http://ovrt.nist.gov/people/sressler/2005-DHM-10.pdf).
- [40] **Kiss, Szilárd.** Visual Property Editing Tools for Web based Virtual Reality. <http://www2003.org/cdrom/papers/poster/p216/p216-kiss.html>.
- [41] Vizx3D Tutorial. <http://www.phamhi.com/tuto3/tuto3part2.htm>.