ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΚΡΗΤΗΣ

ΤΜΗΜΑ ΕΦΑΡΜΟΣΜΕΝΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ & ΠΟΛΥΜΕΣΩΝ

Πτυχιακή Εργασία

Δυναμική δημιουργία και ενσωμάτωση εικονικών χώρων σε τεχνολογία X3D

Σπουδαστής: Καπετανάκης Κων/νος
Εισηγητής: Δρ. Αθανάσιος Μαλάμος

3/11/2009

1 Περιεχόμενα

2 Εισαγωγή 6
3 Περιγραφή των τεχνολογιών που χρησιμοποιήθηκαν 7
3.1 XML 7
3.2 SVG 8
3.3 JAVA 8
3.4 X3D 10
3.5 Επιπρόσθετα X3D Προφίλ 12
4 Μετατροπή SVG σε X3D 13
4.1 Αντιστοιχίες SVG σε X3D 15
4.2 Υποστηριζόμενα αντικείμενα για την μετατροπή 15
4.3 Παράδειγμα μετατροπής SVG σε X3D 18
5 Δημιουργία 3D Αντικειμένων 24
5.1 Χρήση Vivaty studio για την παραγωγή μοντέλου 24
6 Περιγραφή δωματίου και roomDescription.xml 28
7 Περιγραφή λειτουργίας προγράμματος 30
8 Java περιγραφή κλάσεων 31
8.1 BuildingPanelSC 31
8.2 MyX3DBrowser 33
8.3 WorldManager 33
8.4 Building 34
8.5 Message 35
8.6 MessageCreation 35
8.7 XMLHelper 35
8.8 GetSvg 36
9 Περιγραφή X3D σύνθετου μοντέλου 39
9.1 Μοντέλο καρέκλας 39
9.2 Μοντέλο λάμπας τοίχου 44
10 Class Diagrams 52
10.1 Interior Design 52
10.2 SVG to X3D converter 54
11 API - InteriorDesign 56
11.1 InteriorDesignJavadoc Class BuildingPanelSC 56

11.1.1 fields ... 57
11.1.2 browser ... 57
11.1.3 selfRef ... 57
11.1.4 mainScene ... 57
11.1.5 log ... 58
11.1.6 i ... 58
11.1.7 setBrowser ... 58
11.1.8 setFields ... 58
11.1.9 initialize ... 58
11.1.10 shutdown ... 58
11.1.11 eventsProcessed ... 58
11.1.12 readableFieldChanged ... 58
11.1.13 InsertFurniture ... 58
11.1.14 getValuesFromXPath ... 59
11.1.15 convert2Dom4j ... 59
11.2 Class MyX3DBrowser ... 59
11.2.1 root ... 60
11.2.2 readSVG ... 60
11.2.3 StringScene ... 60
11.2.4 MyX3DBrowser ... 60
11.2.5 main ... 60
11.2.6 convert2Dom4j ... 60
11.2.7 convert2w3c ... 60
11.2.8 getValuesFromXPath ... 60
11.3 Package interiorDesign ... 61
11.4 Class Building ... 61
11.4.1 buildingDocument ... 62
11.4.2 str ... 62
11.4.3 Building ... 62
11.4.4 prepareSphere ... 62
11.4.5 prepareScript ... 62
11.4.6 getBuildingNode ... 63
11.4.7 toString ... 63
11.4.8 getStringFromDocument ... 63
11.5 Package interiorDesign::msg ... 63

11.6 Package interiorDesign::xml ... 63
11.7 Class XMLHelper ... 64
11.7.1 log ... 64
11.7.2 loadXML ... 65
11.7.3 loadXML ... 65
11.7.4 saveXML ... 65
11.7.5 getEmptyDocument ... 65
11.8 Class Message ... 65
11.8.1 TYPE_TRANSLATION ... 66
11.8.2 TYPE_CREATION ... 66
11.8.3 type ... 66
11.8.4 objectID ... 66
11.8.5 getType ... 66
11.8.6 getObjectID ... 66
11.8.7 setType ... 67
11.8.8 setObjectID ... 67
11.9 Class MessageCreation ... 67
11.9.1 objectKind ... 68
11.9.2 translation ... 68
11.9.3 scaled ... 69
11.9.4 rotation ... 69
11.9.5 locked ... 69
11.9.6 MessageCreation ... 69
11.9.7 getObjectKind ... 69
11.9.8 setObjectKind ... 69
11.9.9 getTranslation ... 69
11.9.10 getRotation ... 69
11.9.11 getScaled ... 69
11.9.12 setTranslation ... 69
11.9.13 setRotation ... 69
11.9.14 setScaled ... 70
11.9.15 toString ... 70
11.9.16 isLocked ... 70
11.9.17 setLocked ... 70
12 SVG to X3D converter ... 70

12.1 Class GetSvg 70
12.1.1 document 71
12.1.2 scaling 71
12.1.3 final3D 72
12.1.4 wallHeight 72
12.1.5 StrokeWidth 72
12.1.6 floorPotition 72
12.1.7 GetSvg 72
12.1.8 setWallHeight 72
12.1.9 Get3D 72
12.1.10 addText 72
12.1.11 addCurve 72
12.1.12 addCircle 72
12.1.13 addRect 73
12.1.14 addLine 73
12.1.15 addElement 73
12.1.16 setScaling 73
12.1.17 createX3D 73
13 Πηγές 74

2 Εισαγωγή

Συχνά παρουσιάζεται η ανάγκη για την εξωτερίκευση σκέψεων μας για την περιγραφή κάποιου χώρου. Η εφαρμογή αυτής της πτυχιακής εργασίας επιτρέπει στο χρήστη να περιγράψει με λεπτομερή προσέγγιση το χώρο που επιθυμεί. Ξεκινώντας από την κάτοψη του δωματίου, η οποία αποτελείται από μία περιγραφή σε SVG (Scalable Vector Graphics) , υπολογίζεται η ακριβής θέση των τρισδιάστατων πλέον, αντικειμένων. Στη συνέχεια ενσωματώνονται στον κόσμο αυτά τα αντικείμενα-έπιπλα, με επιπρόσθετες ιδιότητες ώστε να επιτρέπεται στον χρήστη η μετακίνησή τους.

Για τον προγραμματισμό της παραπάνω εφαρμογής χρησιμοποιήθηκαν οι επόμενες γλώσσες προγραμματισμού.

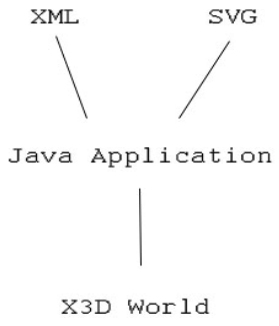
- JAVA
- XML
- SVG
- X3D

Η Java, δίνει στην εφαρμογή την ιδιότητα να λειτουργεί σε όλα τα λειτουργικά συστήματα.

XML, χρησιμοποιήθηκε για την αποθήκευση της περιγραφής του δωματίου, την προσπέλαση των αρχείων X3D (XML based), την μετατροπή από SVG σε X3D του κόσμου.

SVG, για την περιγραφή του δωματίου σε δύο διαστάσεις.

X3D, για την δημιουργία των γραφικών σε τρεις διαστάσεις και την ενσωμάτωση των αισθητήρων στα αντικείμενα για την μετακίνησή τους.



Εικόνα 1 κεντρική λογική

3 Περιγραφή των τεχνολογιών που χρησιμοποιήθηκαν

3.1 XML

Τα έγγραφα που μεταφέρονται στο Internet πρέπει να είναι εύκολα προσβάσιμα, μεταφέρσιμα και ευέλικτα. Οι γενικευμένες γλώσσες αποτελούν επιλογή για την δημιουργία τέτοιων εγγράφων. Η HTML όμως, που είναι η πιο διαδεδομένη γλώσσα για την μεταφορά εγγράφων μέσω Internet είναι περιοριστική και προβληματική. Τα προβλήματα που παρουσιάστηκαν χάθηκαν με την εμφάνιση της XHTML που αποτελεί ανασχεδιασμό της HTML με βάση την XML.

Στην πραγματικότητα, η XML είναι markup γλώσσα για έγγραφα που περιέχουν δομημένες πληροφορίες. Markup γλώσσα είναι ένας μηχανισμός που καθορίζει δομές σε ένα έγγραφο. Οι δομημένες πληροφορίες περιλαμβάνουν περιεχόμενο και κάποιες διευκρινίσεις για το ρόλο που παίζει το περιεχόμενο. Σχεδόν όλα τα έγγραφα έχουν την ίδια δομή.

Η XML είναι κάτι περισσότερο από markup language είναι metalanguage, δηλαδή μια γλώσσα που χρησιμοποιείται για να καθορίσει νέες markup γλώσσες.Η XML συμπληρώνει και δεν αντικαθιστά την HTML. Ενώ η HTML χρησιμοποιείται στη διατύπωση και την εμφάνιση των δεδομένων η XML αναπαριστά τη συναφή έννοια των δεδομένων.Στην HTML τα tags είναι προκαθορισμένα ενώ η XML παρέχει τη δυνατότητα να καθορίζουν οι χρήστες τα tags και τις δομημένες μεταξύ τους σχέσεις.

Τα XML έγγραφα δεν είναι πολύπλοκα αλλά απλά και πολύ αποτελεσματικά. Το διδακτικό υλικό της well-formed XML αναλύει τη δημιουργία των XML εγγράφων, η οποία είναι κατά κάποιο τρόπο ίδια με την HTML καθώς επιτρέπει τη μη δομημένη δημιουργία εγγράφου. Η valid XML είναι πιο σύνθετη. Απαιτεί την ύπαρξη ενός Document Type Definition πριν να γραφεί το έγγραφο αλλά παρέχει μια γενική δομή με βάση την οποία τη δημιουργούμε.
Η γλώσσα προγραμματισμού XML περιγράφει μια κατηγορία πληροφοριών (data objects) που καλούνται <u>XML έγγραφα</u> (documents) καθώς επίσης περιγράφει τμηματικά τη συμπεριφορά των προγραμμάτων που τα επεξεργάζονται.

Τα XML έγγραφα αποτελούνται από μονάδες αποθήκευσης που καλούνται *entities (οντότητες)*, οι οποίες περιέχουν πληροφορίες αναλυμένες ή μη. Οι αναλυμένες πληροφορίες αποτελούνται από *χαρακτήρες (characters)* οι οποίοι συνθέτουν *character data* και άλλοι οι οποίοι συνθέτουν *markup*.

Η μορφή markup κωδικοποιεί την περιγραφή της τελικής αποθήκευσης του εγγράφου καθώς και τη λογική δομή.
Ένα λογισμικό μοντέλο που καλείται **επεξεργαστής XML** χρησιμοποιείται να διαβάζει XML έγγραφα και παρέχει πρόσβαση στο περιεχόμενο και τη δομή τους. Υποτίθεται ότι ο επεξεργαστής XML λειτουργεί εκ μέρους ενός άλλου μοντέλου που καλείται **application (εφαρμογή)**. Αυτή η προδιαγραφή περιγράφει την απαιτούμενη συμπεριφορά του επεξεργαστή και συγκεκριμένα πως θα πρέπει να διαβάζει τα XML δεδομένα και ποιες πληροφορίες πρέπει να παρέχει στην εφαρμογή.
Πηγή: http://www.it.uom.gr/project/xml/Home%20Page.htm

3.2 SVG

Η SVG (Scalable Vector Graphics) είναι μία γλώσσα που χρησιμοποιείται για την περιγραφή 2D γραφικών σε XML. Η περιγραφή των γραφικών βασίζεται σε διανύσματα τα οποία είναι αποθηκευμένα σε μορφή XML. Αυτό έχει σαν αποτέλεσμα να μην αλλοιώνεται η ποιότητα της εικόνας κατά την μεγέθυνση ή την αλλαγή των διαστάσεων της. Οποιοδήποτε στοιχείο του αρχείου SVG έχει την ιδιότητα να γίνει κινούμενο. Επίσης λόγω της υποστήριξης που υπάρχει από το W3C ακολουθεί τα πρότυπα DOM και XSL.

Μπορούμε να επεξεργαστούμε με διάφορα εργαλεία τα αρχεία SVG, ακόμα και με το απλό Notepad. Τα αρχεία αυτά είναι μικρότερα από JPEG και GIF και μπορούν να συμπιεστούν περισσότερο. Οι παραγόμενες φωτογραφίες από SVG κώδικα μπορούν να αλλάξουν μέγεθος, να εκτυπωθούν με μέγιστη ανάλυση χωρίς να έχουμε απώλειες στην ποιότητα. ‘Άλλο ένα πλεονέκτημα είναι ότι το αντικείμενο κειμένου του SVG μπορεί να επιλεγεί (π.χ. για να γίνει αντιγραφή-επικόλληση) αλλά και να αναζητηθεί. Αυτό το καθιστά ιδανικό για τη δημιουργία χαρτών.

3.3 JAVA

Ένα από τα βασικά πλεονεκτήματα της Java έναντι των περισσότερων άλλων γλωσσών είναι η ανεξαρτησία του λειτουργικού συστήματος και πλατφόρμας. Τα προγράμματα που είναι γραμμένα σε *Java* τρέχουνε ακριβώς το ίδιο σε Windows, Linux, Unix και Macintosh (σύντομα θα τρέχουν και σε Playstation καθώς και σε άλλες κονσόλες παιχνιδιών) χωρίς να χρειαστεί να ξαναγίνει μεταγλώττιση (compiling) ή να αλλάξει ο πηγαίος κώδικας για κάθε διαφορετικό λειτουργικό σύστημα. Για να επιτευχθεί όμως αυτό χρειαζόταν κάποιος τρόπος έτσι ώστε τα προγράμματα γραμμένα σε Java να μπορούν να είναι «κατανοητά» από κάθε υπολογιστή ανεξάρτητα

του είδους επεξεργαστή (Intel x86, IBM, Sun SPARC, Motorola) αλλά και λειτουργικού συστήματος (Windows, Unix, Linux, BSD, MacOS). Ο λόγος είναι ότι κάθε κεντρική μονάδα επεξεργασίας κατανοεί διαφορετικό κώδικα μηχανής. Ο συμβολικός (*assembly*) κώδικας που εκτελείται σε Windows είναι διαφορετικός από αυτόν που εκτελείται σε έναν υπολογιστή Macintosh. Η λύση δόθηκε με την ανάπτυξη της *Εικονικής Μηχανής* (*Virtual Machine* ή VM ή ΕΜ στα ελληνικά).

Αφού γραφεί κάποιο πρόγραμμα σε Java τότε μεταγλωττίζεται μέσω του εργαλείου javac, το οποίο παράγει έναν αριθμό από αρχεία .class (=bytecode). Το bytecode είναι η μορφή που παίρνει ο πηγαίος κώδικας της Java όταν μεταγλωττιστεί. Όταν προσπαθήσουμε λοιπόν να εκτελέσουμε την εφαρμογή μας το Java Virtual Machine που πρέπει να είναι εγκατεστημένο στο μηχάνημά μας, θα αναλάβει να διαβάσει τα αρχεία .class και να τα μεταφράσει σε γλώσσα και εντολές μηχανής (assembly) που υποστηρίζει το λειτουργικό μας σύστημα και ο επεξεργαστής μας, έτσι ώστε να εκτελεστεί (να σημειώσουμε εδώ ότι αυτό συμβαίνει με την παραδοσιακή Εικονική Μηχανή (Virtual Machine) . Πιο σύγχρονες εφαρμογές της εικονικής Μηχανής μπορούν και μεταγλωττίζουν πολλά τμήματα bytecode απ' ευθείας σε εγγενή κώδικα (native code) με αποτέλεσμα να βελτιώνεται η ταχύτητα). Χωρίς αυτό δε θα ηταν δυνατή η εκτέλεση λογισμικού γραμμένου σε Java. Πρέπει να πούμε ότι η JVM είναι λογισμικό εξαρτημένο από την πλατφόρμα, δηλαδή για κάθε είδος λειτουργικού συστήματος και αρχιτεκτονικής επεξεργαστή υπάρχει διαφορετική έκδοση του. Έτσι υπάρχουν διαφορετικές JVM για Windows, Linux, Unix, Macintosh, κινητά τηλέφωνα, παιχνιδο-μηχανές κλπ.

Οτιδήποτε θέλει να κάνει ο προγραμματιστής (ή ο χρήστης) γίνεται μέσω της εικονικής μηχανής. Αυτό βοηθάει στο να υπάρχει μεγαλύτερη ασφάλεια στο σύστημα γιατί η εικονική μηχανή είναι υπεύθυνη για την επικοινωνία χρήστη - υπολογιστή. Ο προγραμματιστής δεν μπορεί να γράψει κώδικα ο οποίος θα έχει καταστροφικά αποτελέσματα για τον υπολογιστή γιατί η εικονική μηχανή θα τον ανιχνεύσει και δε θα επιτρέψει να εκτελεστεί. Από την άλλη μεριά ούτε ο χρήστης μπορεί να κατεβάσει «κακό» κώδικα από το δίκτυο και να τον εκτελέσει. Αυτό είναι ιδιαίτερα χρήσιμο για μεγάλα κατανεμημένα συστήματα όπου πολλοί χρήστες χρησιμοποιούν το ίδιο πρόγραμμα συγχρόνως.

Ακόμα μία ιδέα που βρίσκεται πίσω από τη *Java* είναι η ύπαρξη του συλλέκτη απορριμμάτων (*Garbage Collector*). Συλλογή απορριμμάτων είναι μία κοινή ονομασία που χρησιμοποιείται στον τομέα της πληροφορικής για

να δηλώσει την ελευθέρωση τμημάτων μνήμης από δεδομένα που δε χρειάζονται και δε χρησιμοποιούνται άλλο. Αυτή η απελευθέρωση μνήμης στη Java είναι αυτόματη και γίνεται μέσω του συλλέκτη απορριμμάτων. Υπεύθυνη για αυτό είναι και πάλι η εικονική μηχανή η οποία μόλις «καταλάβει» ότι ο σωρός (heap) της μνήμης (στη Java η συντριπτική πλειοψηφία των αντικειμένων αποθηκεύονται στο σωρό σε αντίθεση με τη C++ όπου αποθηκεύονται κυρίως στη στοίβα - stack) κοντεύει να γεμίσει ενεργοποιεί το συλλέκτη απορριμμάτων. Έτσι ο προγραμματιστής δε χρειάζεται να ανησυχεί για το πότε και αν θα ελευθερώσει ένα συγκεκριμένο τμήμα της μνήμης, ούτε και για δείκτες (pointers) που αναφέρονται σε άδειο χώρο μνήμης. Αυτό είναι ιδιαίτερα σημαντικό αν σκεφτούμε ότι ένα μεγάλο ποσοστό κατάρρευσης των προγραμμάτων οφείλονται σε λανθασμένο χειρισμό της μνήμης.

Παρόλο που η εικονική μηχανή προσφέρει όλα αυτά (και όχι μόνο) τα πλεονεκτήματα, η *Java* είναι πιο αργή σε σχέση με άλλες προγραμματιστικές γλώσσες υψηλού επιπέδου (high-level) όπως η C και η C++. Έχει αποδειχτεί ότι η C++ μπορεί να είναι αρκετές φορές γρηγορότερη από την Java. Ωστόσο γίνονται προσπάθειες από τη Sun για τη βελτιστοποίηση της εικονικής μηχανής, ενώ υπάρχουν και άλλες υλοποιήσεις της εικονικής μηχανής από διάφορες εταιρίες (όπως IBM) οι οποίες μπορεί σε κάποια σημεία να προσφέρουν καλύτερα και σε κάποια άλλα χειρότερα αποτελέσματα. Επιπλέον με την καθιέρωση των μεταγλωττιστών JIT (Just In Time), οι οποίοι μετατρέπουν το bytecode απ' ευθείας σε γλώσσα μηχανής, η διαφορά ταχύτητας από τη C++ έχει μικρύνει κατά πολύ. Οι τελευταίες εκδόσεις του javac με τη χρήση της τεχνολογίας Hot Spot καταφέρνει να παρέχει αξιόλογες επιδόσεις που πλησιάζουν ή και ξεπερνούν σε μερικές περιπτώσεις εγγενή κώδικα.
Πηγή: http://el.wikipedia.org/wiki/Java

3.4 X3D

eXtensible 3D είναι ένα είδος αρχείων βασισμένα σε XML, που αναπαριστά τρισδιάστατα γραφικά. Αποτελεί την βελτίωση της VRML. Μας δίνει την δυνατότητα να περιγράψουμε μία τρισδιάστατη σκηνή με αντικείμενα τύπου XML.

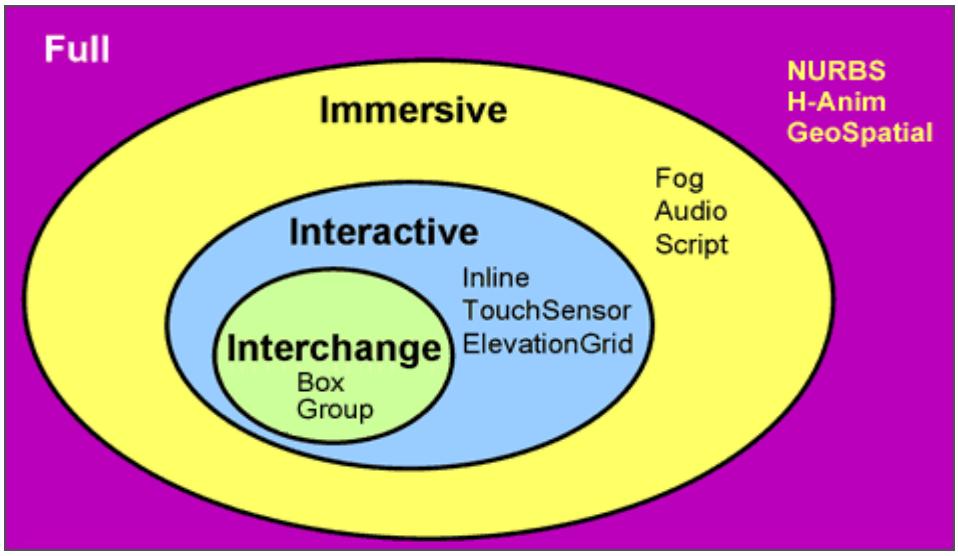
Το X3D έχει επιλογές για προφίλ ανάλογα με το σκοπό που θέλει να πετύχει ο προγραμματιστής και τις ιδιότητες που τον ενδιαφέρουν και μπορεί να χρησιμοποιήσει. Τα προφίλ που προσφέρει είναι :

- **Interchange**
  Είναι το βασικό προφίλ για την επικοινωνία μεταξύ των εφαρμογών. Υποστηρίζει γεωμετρικά σχήματα, υφές, βασικό φωτισμό και κίνηση.

- **Interactive**
  Ενεργοποιεί βασικές ιδιότητες αλληλεπίδρασης με το τρισδιάστατο περιβάλλον προσθέτοντας διάφορους αισθητήρες που χρησιμοποιούνται για την περιήγηση και αλληλεπίδραση του χρήστη στον κόσμο. Προσθέτει ακόμα και ρολόι και περισσότερες ιδιότητες φωτισμού.

- **Immersive**
  Ενεργοποιεί πλήρως τα τρισδιάστατα γραφικά και την αλληλεπίδραση του χρήστη με αισθητήρες, συμπεριλαμβάνοντας την ενσωμάτωση ήχου, ανίχνευση συγκρούσεων 3D αντικειμένων, ομίχλης, και τη δυνατότητα scripting (μικρά κομμάτια κώδικα σε JavaScript) .

- **Full**
  Συμπεριλαμβάνει όλα τα nodes που υπάρχουν, επιπρόσθετα NURBS,H-Anim και Geospatial components.



Εικόνα 2 X3D προφίλ
Πηγή www.w3d.org

3.5 Επιπρόσθετα X3D Προφίλ

- **MPEG-4 Interactive** είναι μία μικρή αρχική έκδοση ενός αλληλεπιδραστικού προφίλ σχεδιασμένο για αναμετάδοση, μικρές συσκευές χειρός και κινητά τηλέφωνα.

- **CDF (CAD Distillation Format)** είναι υπό ανάπτυξη για την ενεργοποίηση της μετάφρασης δεδομένων από CAD σε ένα είδος ανοικτού κώδικα για την προβολή διαδραστικών περιεχομένων.

Μερικά από τα πλεονεκτήματα της χρήσης του X3D εκτός από το μικρό μέγεθος των αρχείων που το καθιστά ιδανικό για δικτυακές εφαρμογές, είναι η δυνατότητα λειτουργίας του σε όλες τις πλατφόρμες ανεξάρτητα από το λειτουργικό σύστημα που το φιλοξενεί, η δυναμικότητα που προσφέρει για την επέκταση του σε συνδυασμό με την διαπεραστικότητα που μπορεί να έχει με τον χρήστη. Μπορεί εκτός από την απεικόνιση τρισδιάστατων αντικειμένων να αναπαράγει αρχεία βίντεο και ήχου.

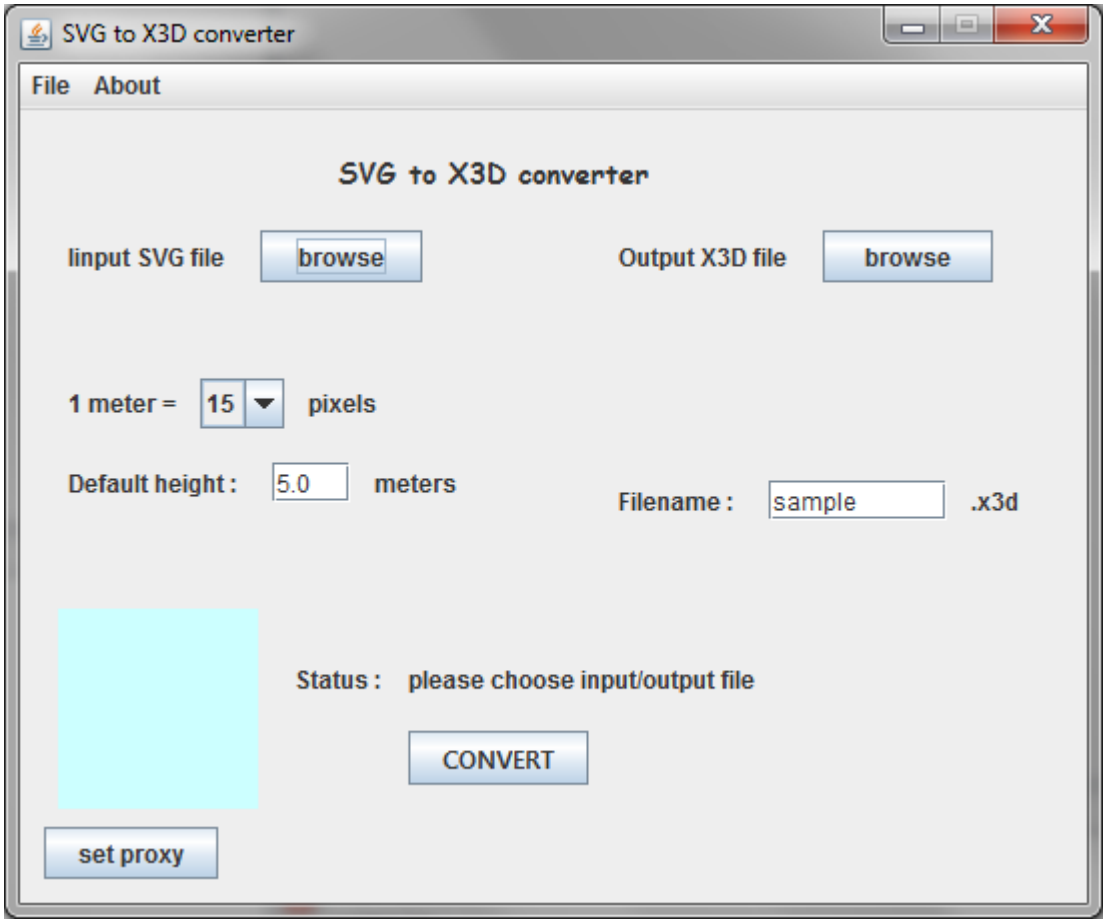
Ο κόσμος που παράγει το X3D δεν έχει καμία έλλειψη σε σχέση με αντίστοιχους κόσμους από άλλα προγράμματα όπως το Maya και το 3DS max. Μπορούμε και εδώ να έχουμε κάμερες αισθητήρες εξωτερικών συσκευών αλλά και κίνησης μέσα στον κόσμο. Με κατάλληλα script μπορούμε να εκμεταλλευτούμε τις εισόδους και εξόδους των αισθητήρων και να δημιουργήσουμε μια πολύ φιλική προς το χρήστη τρισδιάστατη εφαρμογή. Ένα πολύ μεγάλο πλεονέκτημα του X3D είναι η υποστήριξη που προσφέρει για διαδικτυακές εφαρμογές ώστε να μπορούν οι χρήστες να μοιράζονται αντικείμενα σε πραγματικό χρόνο.

Οι δύο τύποι αισθητήρων που χρησιμοποιήθηκαν για την πτυχιακή είναι:
A) PlaneSensor
B) CylinderSensor

Ο πρώτος χρησιμοποιείται για την μετατόπιση των αντικειμένων στο χώρο, ενώ ο δεύτερος για την περιστροφή γύρο από το κέντρο του άξονά τους.

4 Μετατροπή SVG σε X3D

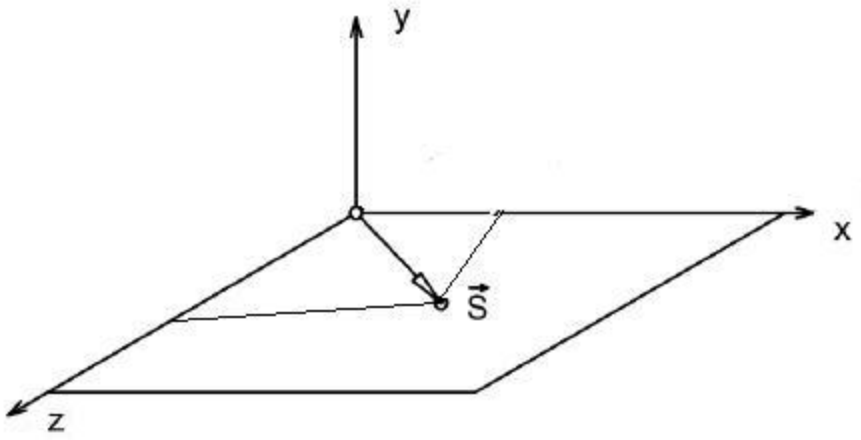
Το πρώτο βήμα στην πτυχιακή εργασία ήταν η δημιουργία μιας πρωτότυπης εφαρμογής, η οποία μετατρέπει το δεδομένο SVG που αναπαριστά ένα δισδιάστατο δωμάτιο, στο αντίστοιχο τρισδιάστατο σε X3D. Τα δύο συστήματα αυτά εκτός από την διαφορά της τρίτης διάστασης έχουν και διαφορετικό σύστημα αξόνων.



Εικόνα 3 X3D to SVG converter

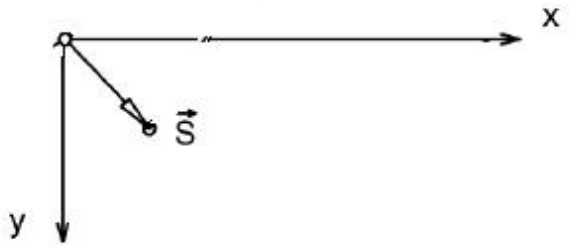
Στον τρισδιάστατο κόσμο ένα μοντέλο έχει 3 τιμές που περιγράφουν σε μέτρα που θα τοποθετηθεί το κέντρο του. Οι τρείς τιμές αυτές είναι ύψος, βάθος και η οριζόντια απόσταση από το κέντρο των αξόνων. Οι αποστάσεις αυτές μετρούνται σε μέτρα. Όπως φαίνεται στην παρακάτω εικόνα το ύψος στο X3D είναι η τιμή του άξονα-y.

Στην παρακάτω εικόνα βλέπουμε ένα παράδειγμα ενός διανύσματος. Η αρχή του είναι το κέντρο των αξόνων, δηλαδή x=0, y=0 και z=0. Ενώ το τέλος του διανύσματος είναι το σημείο S.Δεν είναι αρκετό όμως να πάρουμε από το SVG την αρχή και το τέλος του διανύσματος καθώς το X3D έχει σαν όρισμα του αντίστοιχου αντικειμένου, το κέντρο του κουτιού και τις μοίρες για τον προσανατολισμό.



Εικόνα 4 X3D σύστημα συντεταγμένων

Το παραπάνω διάνυσμα προέκυψε μετά από τον υπολογισμό θέσης και προσανατολισμού του αντίστοιχου διανύσματος S από τον δισδιάστατο κόσμο όπως φαίνεται στην παρακάτω εικόνα.



Εικόνα 5 SVG σύστημα συντεταγμένων

Το σύστημα συντεταγμένων του SVG έχει σαν αρχή αξόνων την πάνω αριστερή γωνία της οθόνης. Οι άξονες εκτίνονται θετικά όπως φαίνεται στην παραπάνω εικόνα. Οι αποστάσεις μετρούνται όπως τις καθορίσει ο σχεδιαστής. Εμείς επιλέξαμε pixels.

Μία ακόμα σημαντική διαφορά είναι η τιμή για την περιστροφή. Στο SVG την δηλώνουμε σε μοίρες ενώ στο X3D σε ακτίνια.

Λαμβάνοντας υπόψη όλα τα παραπάνω στον αλγόριθμο που δημιουργεί τον τρισδιάστατο κόσμο βασισμένο στον δισδιάστατο με το οποίο το τροφοδοτούμε έχουμε ένα ακριβές και λεπτομερές αποτέλεσμα.

4.1 Αντιστοιχίες SVG σε X3D

SVG	***X3D***
Line	Box
rect	Rectangle2D
circle	cylinder
Path	IndexedFaceSet
Text	Text
15 pixel	1 meter

Μία ακόμα διαφορά σε αυτά τα δύο συστήματα είναι ο αρχικός προσανατολισμός των σχημάτων. Στο X3D η μηδενική περιστροφή είναι όταν το αντικείμενο είναι κάθετο στον άξονα x ενώ στο SVG είναι παράλληλο με τον άξονα x.

4.2 Υποστηριζόμενα αντικείμενα για την μετατροπή

Τα χρώματα που υποστηρίζονται είναι:

- Μαύρο
- Μπλε
- Πράσινο
- Κίτρινο
- Κόκκινο

Το αντικείμενο **g :**

translation(x,y)

rotation στη μορφή *rotate*(x)
η αλλαγή του κέντρου περιστροφής δεν υποστηρίζεται

Οι τιμές που υπάρχουν στο translate προστίθενται στα ανάλογα (x,y) πεδία τοποθέτησης στα αντικείμενα παιδιά του g.

```
<g transform="translate(130.0,35.0) rotate(-13.346603332976507)">
```

Για την προσθήκη παραθύρων ή πόρτας θα χρειαστεί να προσθέσουμε ένα attribute id με τιμή window ή door αντίστοιχα. Το παιδί θα αποτελεί εξαίρεση στο σχεδιασμό που περιγράφεται παρακάτω και στη θέση του ενσωματώνεται στον κόσμο ένα αντικείμενο πόρτας ή παραθύρου.

Το αντικείμενο **Line** από το SVG αντιστοιχεί με το αντικείμενο **Box** στο X3D **:**

Stroke δίνει τιμή για την επιλογή χρώματος του αντικειμένου.

Stroke-width δίνει τιμή για το πάχος της γραμμής το οποίο το διαιρούμε δια 6 και παίρνει θέση στην μεταβλητή x του size : *(stroke-width / 6),y,z*

x1,x2,y1,y2 για τον υπολογισμό της θέσης, του μήκους και της περιστροφής με τον παρακάτω μαθηματικό τύπο.

width=(x2 - x1)^2 + (y2 – y1)^2

vertical_lline1 = x2 - x1

vertical_line2 = y2 - y1

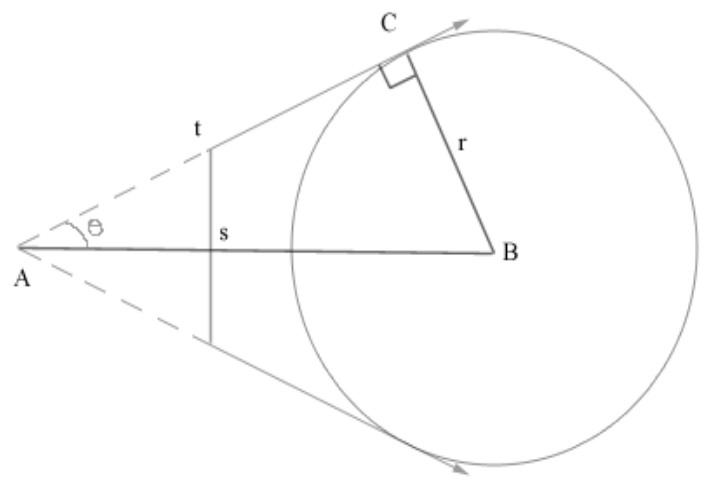
theta = java.lang.Math.atan2(vertical_line2,vertical_line1)-pi/2
[χρησιμοποιούμε την atan2 για να μας επιστρέφονται και οι αρνητικές τιμές]

theta = theta + java.lang.Math.toRadians(rotation) [αν έχουμε rotation από αντικείμενο g]

*finalX1=x1 + (width / 2) * cos(theta)*

*finalY1=y1 + (width / 2) * sin(theta)*

```
<line stroke="black" stroke-width="1" x1="0" x2="233.0" y1="0"
y2="59.0"/>
```



$s^2 = t^2 + r^2$ Theta =atan(r/t)

Το αντικείμενο **rect** από το SVG αντιστοιχεί με το αντικείμενο **Rectangle2D** στο X3D **:**

```
<rect fill="none" height="78.0" stroke="black" stroke-width="1"
width="66.0" x="0" y="0"/>
```

Το αντικείμενο **circle** από το SVG αντιστοιχεί με το αντικείμενο **Cylinder** στο X3D **:**

```
<circle cx="0" cy="0" fill="none" r="22.80350850198276" stroke="black"
stroke-width="1"/>
```

αν η τιμή της ακτίνας είναι μικρότερη από 1 μέτρο τότε θεωρούμε πως το σχέδιο δηλώνει την ύπαρξη κολόνας οπότε επιτρέπουμε στο top του κυλίνδρου να γίνει render. Διαφορετικά θεωρούμε τον χώρο σαν κυκλικό δωμάτιο και αφήνουμε ανοικτή την πάνω πλευρά.

Το αντικείμενο **Path** από το SVG αντιστοιχεί με το αντικείμενο **IndexedFacedSet** στο X3D **:**

```
<path d="M0,0 C52.99314880371094,-62.99314880371094
92.99314880371094,-62.99314880371094 145.98629760742188,0.0"
fill="none" stroke="black" stroke-width="1"/>
```

Λόγω της περιπλοκότητας της δομής αυτού του αντικειμένου έχουν υλοποιηθεί μόνο οι καμπύλες της παραπάνω μορφής.

Το αντικείμενο **Text** από το SVG αντιστοιχεί με το αντικείμενο **Text** στο X3D **:** Για το αντικείμενο αυτό υπολογίζεται η γραμματοσειρά και το μέγεθος γραμμάτων καθώς και η θέση που θα τοποθετηθούν οι λέξεις.

```
<text fill="black" font-family="Arial" font-size="12" x="0" y="0">
  ENTER TEXT </text>
```

4.3 Παράδειγμα μετατροπής SVG σε X3D

Στο τέλος της πρώτης φάσης της πτυχιακής δημιουργήθηκε μία stand alone εφαρμογή για την μετατροπή SVG σε X3D.

1) Αρχικά επιλέγουμε το αρχείο SVG που θέλουμε να μετατρέψουμε, κάνοντας κλικ στο πρώτο κουμπί browse. Ακριβώς από κάτω θα εμφανιστεί η διαδρομή του αρχείου που επιλέξαμε.

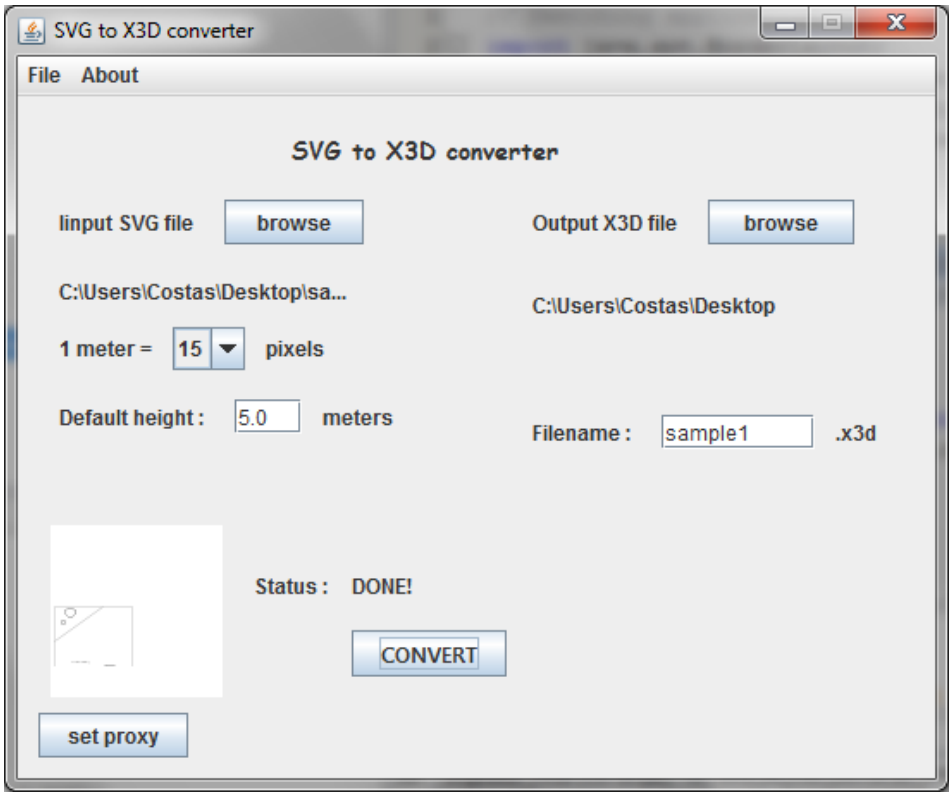
2) Στο πλαίσιο δίπλα από το STATUS βλέπουμε μία προεπισκόπηση του αρχείου που επιλέξαμε και στη συνέχεια επιλέγουμε το φάκελο και το όνομα του αρχείου που θέλουμε να δημιουργηθεί, με το δεύτερο κουμπί browse.

3) Στη συνέχεια μπορούμε να ορίσουμε την κλίμακα pixel/meters. Πόσα pixels στον δισδιάστατο χώρο αντιστοιχούν σε ένα μέτρο στον τρισδιάστατο.

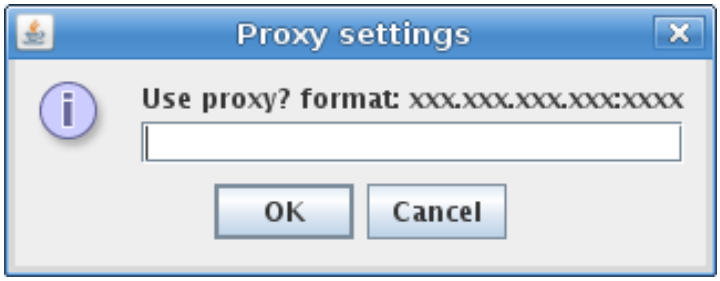
4) Τέλος γράφουμε το ύψος των τοίχων που θα έχει το δωμάτιο σε μέτρα, στο αντίστοιχο πεδίο.

Αν χρειαστεί έχουμε την δυνατότητα να ορίσουμε Proxy server πατώντας το κουμπί **set proxy** στην κάτω μεριά της εφαρμογής. Στο αντίστοιχο πεδίο που θα εμφανιστεί πληκτρολογούμε τον proxy σε μορφή :

xxx.xxx.xxx.xxx:xxxx (μετά το χαρακτήρα ¨:¨ πληκτρολογούμε την θύρα)



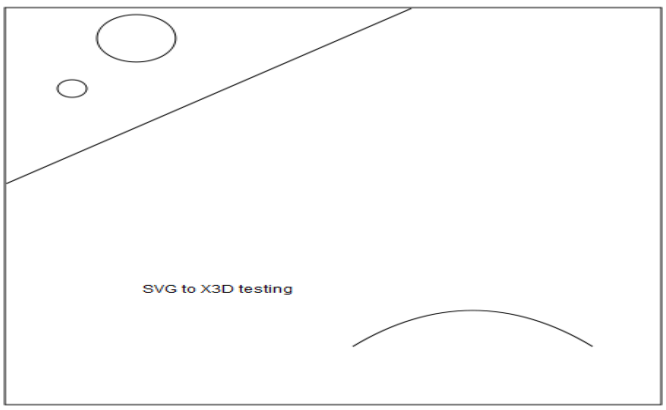
Εικόνα 6 αρχική οθόνη



Εικόνα 7 ρύθμιση proxy

Στα παρακάτω πλαίσια μπορούμε να δούμε το πηγαίο SVG αρχείο και το αντίστοιχο σχήμα που δημιουργεί σε δύο διαστάσεις.

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 20010904//EN" "http://www.w3.org/TR/2001/REC-
SVG-20010904/DTD/svg10.dtd">
<svg height="900px" viewBox="0 400 800 100" width="800px" xml:space="preserve"
xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink"
zoomAndPan="disable">
 <g id="line" transform="translate(25.0,224.0) rotate(0.0)">
  <line stroke="black" stroke-width="1" x1="0" x2="277.0" y1="0" y2="-199.0"/>
 </g>
 <g id="circle" transform="translate(113.92582403567252,58.92582403567252) rotate(0.0)">
  <circle cx="0" cy="0" fill="none" r="26.92582403567252" stroke="black" stroke-width="1"/>
 </g>
 <g id="circle" transform="translate(70.0,116.0) rotate(0.0)">
  <circle cx="0" cy="0" fill="none" r="10.0" stroke="black" stroke-width="1"/>
 </g>
 <g id="curve" transform="translate(262.0,409.0) rotate(0.0)">
  <path d="M0,0 C54.667683919270836,-54.667683919270836 109.33536783854167,-
54.667683919270836 164.0030517578125,0.0" fill="none" stroke="black" stroke-width="1"/>
 </g>
 <g id="text" transform="translate(106.0,348.0) rotate(0.0)">
  <text fill="black" font-family="Arial" font-size="12" x="0" y="0">
   SVG to X3D testing
  </text>
 </g>
 <g id="rectangle" transform="translate(24.0,24.0) rotate(0.0)">
  <rect fill="none" height="451.0" stroke="black" stroke-width="1" width="449.0" x="0" y="0"/>
 </g>
</svg>
```



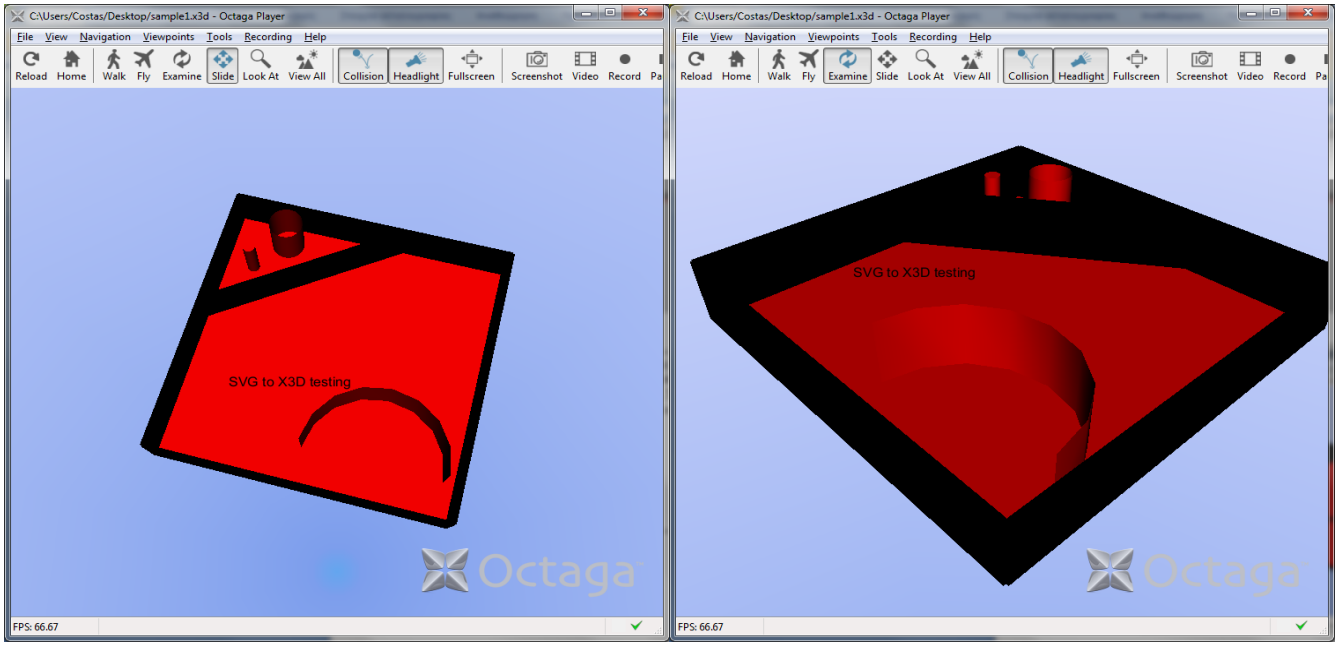
Εικόνα 8 SVG αρχείο

Στα παρακάτω πλαίσια είναι ο κώδικας που δημιουργεί η εφαρμογή και η προβολή του στον κατάλληλο player (Octaga).

```
<?xml version="1.0" encoding="UTF-8"?>
<X3D profile="Immersive">
<Scene>
<Background DEF="background" groundAngle=".03491 1.5708" groundColor=".39216 .65098
.93333 .4902 .61961 .92549 .90196 .90196 1" skyColor=".39216 .65098 .93333 .4902 .61961 .92549
.90196 .90196 1"/>
<Script DEF="SC" directOutput="TRUE" mustEvaluate="TRUE"
url="../build/classes/BuildingPanelSC.class"/>
<Viewpoint DEF="DefaultCam" description="DefaultCam" fieldOfView="0.785"
jump="TRUE" orientation="-1 -0 0 .579" position="17.50242 18.08865 50.61155"/>
<Transform translation="10.899999999999999 0 8.3">
<Transform rotation="0 -1 0 -2.1937724590539247" scale="1 1 1">
<Shape>
<Appearance containerField="appearance">
<Material ambientIntensity="0.200" diffuseColor="0 0 0" shininess="0.200"/>
</Appearance>
<Box containerField="geometry" size="0.3333333333333333 5.0 22.738122662661095"
solid="false"/>
</Shape>
</Transform>
</Transform>
<Transform>
<Transform rotation="1 0 0 1.5707963267948966" scale="1 1 1"
translation="16.566666666666666 -2.51 16.633333333333333">
<Shape>
<Rectangle2D size="29.933333333333334 30.066666666666666" solid="false"/>
<Appearance containerField="appearance">
<Material ambientIntensity="0.200" diffuseColor="1 0 0" shininess="0.200"/>
</Appearance>
</Shape>
</Transform>
</Transform>
<Transform translation="16.566666666666666 0 1.6">
<Transform rotation="0 -1 0 -1.5707963267948966" scale="1 1 1">
<Shape>
<Appearance containerField="appearance">
<Material ambientIntensity="0.200" diffuseColor="0 0 0" shininess="0.200"/>
</Appearance>
<Box containerField="geometry" size="0.3333333333333333 5.0 29.933333333333334"
solid="false"/>
</Shape>
</Transform>
</Transform>
<Transform translation="1.600000000000001 0 16.633333333333333">
<Transform rotation="0 -1 0 0.0" scale="1 1 1">
<Shape>
<Appearance containerField="appearance">
```

<Box containerField="geometry" size="0.3333333333333333 5.0 30.066666666666666" solid="false"/>
</Shape>
</Transform>
</Transform>
<Transform translation="16.566666666666666 0 31.666666666666668">
<Transform rotation="0 -1 0 -1.5707963267948966" scale="1 1 1">
<Shape>
<Appearance containerField="appearance">
<Material ambientIntensity="0.200" diffuseColor="0 0 0" shininess="0.200"/>
</Appearance>
<Box containerField="geometry" size="0.3333333333333333 5.0 29.933333333333334" solid="false"/>
</Shape>
</Transform>
</Transform>
<Transform translation="31.533333333333335 0 16.633333333333333">
<Transform rotation="0 -1 0 0.0" scale="1 1 1">
<Shape>
<Appearance containerField="appearance">
<Material ambientIntensity="0.200" diffuseColor="0 0 0" shininess="0.200"/>
</Appearance>
<Box containerField="geometry" size="0.3333333333333333 5.0 30.066666666666666" solid="false"/>
</Shape>
</Transform>
</Transform>
<Transform>
<Transform rotation="0 1 0 1.5707963267948966" scale="1 1 1" translation="7.595054935711501 0 3.928388269044835">
<Shape>
<Cylinder height="5.0" radius="1.7950549357115013" side="TRUE" solid="FALSE" top="FALSE"/>
<Appearance containerField="appearance">
<Material ambientIntensity="0.200" diffuseColor="1 0 0" shininess="0.200"/>
</Appearance>
</Shape>
</Transform>
</Transform>
<Transform>
<Transform rotation="0 1 0 1.5707963267948966" scale="1 1 1" translation="4.666666666666667 0 7.733333333333333">
<Shape>
<Cylinder height="5.0" radius="0.6666666666666666" side="TRUE" solid="FALSE" top="TRUE"/>
<Appearance containerField="appearance">
<Material ambientIntensity="0.200" diffuseColor="1 0 0" shininess="0.200"/>
</Appearance>
</Shape>
</Transform>
</Transform>
<Transform rotation="0 0.0 0" translation="7.066666666666666 0 23.2">
<Shape>
<Text string="SVG to X3D testing">

```
<ScreenFontStyle family="Arial" justify="BEGIN" pointSize="12" style="PLAIN"/>
</Text>
<Appearance containerField="appearance">
<Material ambientIntensity="0.200" diffuseColor="0 0 0" shininess="0.200"/>
</Appearance>
</Shape>
</Transform>
<Transform>
<Transform rotation="0 0.0 0" scale="7.289024522569445 2.5 7.289024522569445"
translation="22.933435058593748 0 27.266666666666666">
<Shape>
<IndexedFaceSet containerField="geometry" coordIndex=" 0 1 2 -1    1 0 3 -1    3 4 1 -1    4 3 5 -1    5 6 4 -
1    6 5 7 -1    7 8 6 -1    8 7 9 -1    10 11 12 -1    11 10 13 -1    13 14 11 -1    14 13 15 -1    15 16 14 -1    16
15 17 -1    17 2 16 -1    2 17 0 -1" creaseAngle="0.524" solid="false">
<Coordinate containerField="coord" point=" 0 1 -1    .38268 -1 -.92388    0 -1 -1    .38268 1 -.92388
.70711 -1 -.70711    .70711 1 -.70711    .92388 -1 -.38268    .92388 1 -.38268    1 -1 0    1 1 0    -1 1 -0    -
.92388 -1 -.38268    -1 -1 -0    -.92388 1 -.38268&#9;    -.70711 -1 -.70711    -.70711 1 -.70711&#9;    -
.38268 -1 -.92388    -.38268 1 -.92388"/>
</IndexedFaceSet>
<Appearance containerField="appearance">
<Material ambientIntensity="0.200" diffuseColor="1 0 0" shininess="0.200"/>
</Appearance>
</Shape>
</Transform>
</Transform>
</Scene>
</X3D>
```

Εικόνα 9 αρχείο X3D

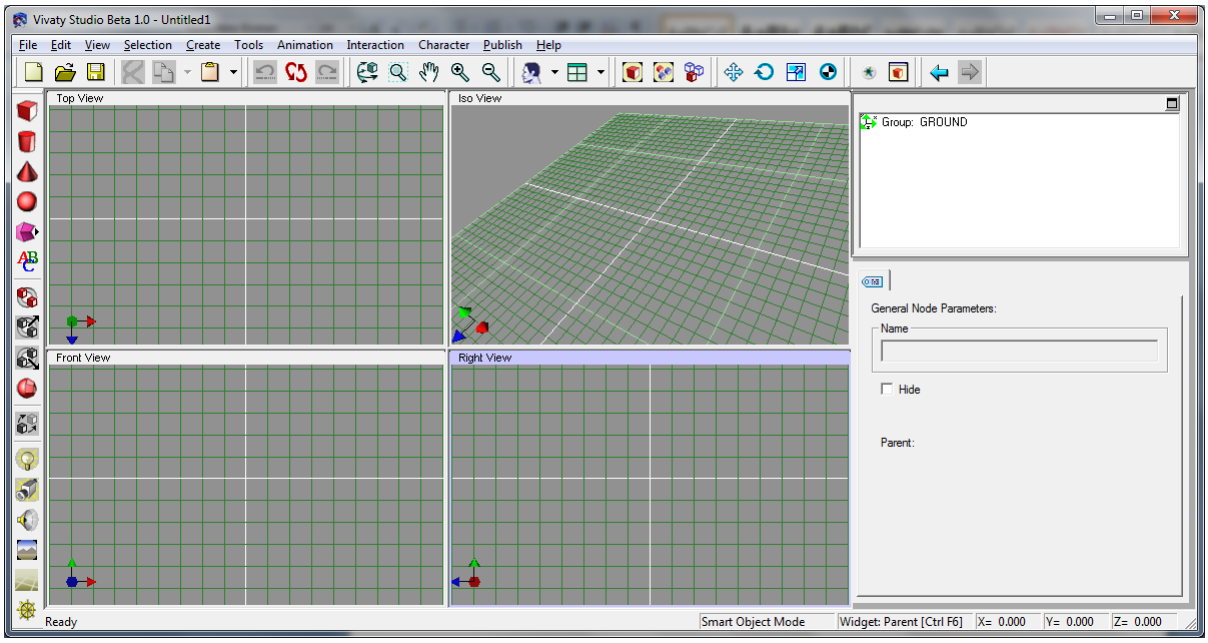
5 Δημιουργία 3D Αντικειμένων

Στη δεύτερη φάση της πτυχιακής εργασίας χρησιμοποιήθηκε ο Xj3D για το API που προσφέρει. Ξεκινώντας από τις βιβλιοθήκες του ανοίγουμε έναν java browser ορίζοντας αρχική σκηνή αυτή που μόλις παρήγαμε από την εφαρμογή της πρώτης φάσης. Στη συνέχεια χρησιμοποιώντας 3DStudio Max και Vivaty studio μπορούμε να δημιουργήσουμε τρισδιάστατα έπιπλα και αντικείμενα, για το δωμάτιο, που θέλουμε να τοποθετήσουμε. Το κάθε αντικείμενο πρέπει να περιλαμβάνει κάποια εξωτερικά Transform που περιγράφονται παρακάτω, στα οποία ενσωματώνονται δυναμικά οι αισθητήρες και οι τιμές για το μέγεθος και την μετατόπιση του αντικειμένου.

Τα αντικείμενα του δωματίου καθώς και ο χώρος του δωματίου περιγράφονται σε ένα κεντρικό XML έγγραφο που βρίσκεται στο models/settings και ονομάζεται roomdescription.xml.

Τα μοντέλα των αντικειμένων βρίσκονται στο φάκελο models ενώ τα απαραίτητα textures τοποθετούνται στον ομώνυμο υπο-φάκελο.
Πρέπει να ληφθεί υπόψη κατά τη δημιουργία του X3D μοντέλου για την σωστή διαδρομή που θα καταχωρηθεί στο αντίστοιχο tag του αρχείου.

5.1 Χρήση Vivaty studio για την παραγωγή μοντέλου

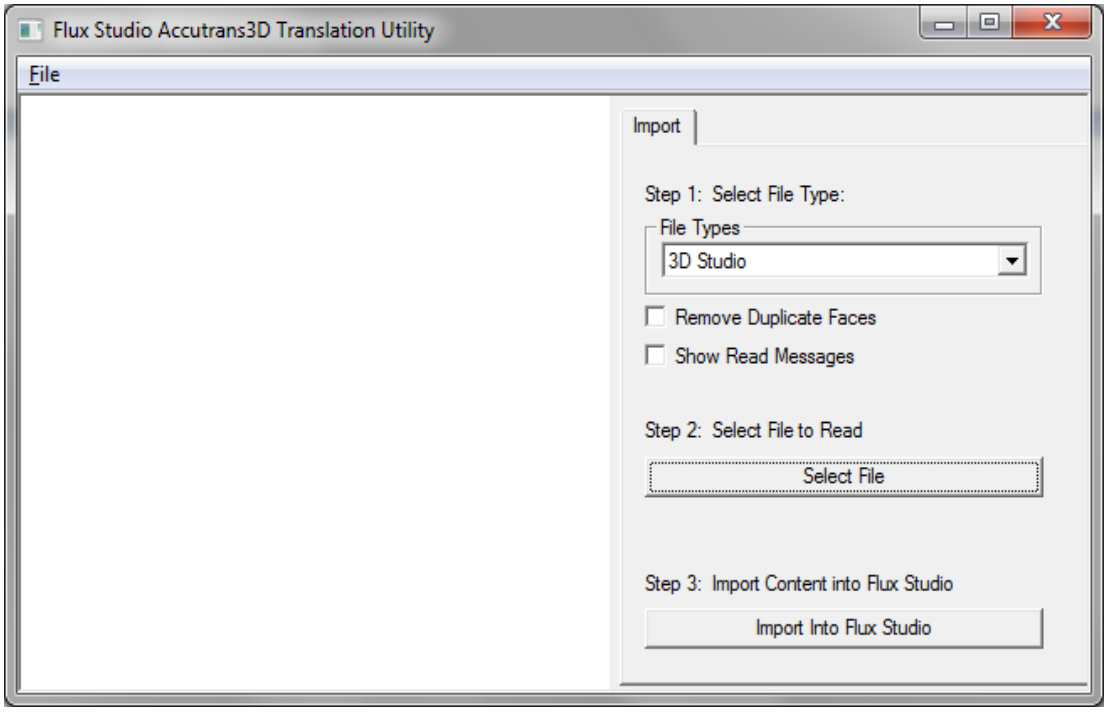


Εικόνα 10 Vivaty Studio (Flux)

1) Επιλέγουμε **File → import other Format**
2) Στη συνέχεια διαλέγουμε τον τύπο του αρχείου που θέλουμε να εισάγουμε, στην αριστερή στήλη του παραθύρου που άνοιξε.
3) Με την επιλογή **Select File** επιλέγουμε το αρχείο με το μοντέλο που θέλουμε να πάρουμε τον X3D κώδικά για τη δημιουργία του.

Προτείνεται η ενεργοποίηση της επιλογής **remove duplicate faces**

4) Ολοκληρώνουμε την διαδικασία εισαγωγής με την επιλογή του **Import into flux studio**.



Εικόνα 11 import 3DS to vivaty

5) Για να εξάγουμε τον κώδικα που δημιουργεί το αντικείμενο που βλέπουμε τώρα στις 4 οθόνες, αφού ¨καθαρίσουμε¨ το μοντέλο από περιττά αντικείμενα που μπορεί να έχει, όπως κάμερες, φώτα, παρασκήνιο ή ότι άλλο θέλουμε να αφαιρέσουμε διορθώνουμε τη διαδρομή που έχει για τα textures επιλέγοντας κάθε αντικείμενο ξεχωριστά, κάτω δεξιά πάμε στην καρτέλα **texture** και συμπληρώνουμε ή διορθώνουμε το πρώτο πεδίο κάτω από τον τίτλο **texture**.

6) Για να πάρουμε τον κώδικα που δημιουργεί το μοντέλο προσέχουμε να μην έχουμε επιλεγμένο κάποιο κομμάτι του μοντέλου (κάνοντας κλικ στο κενό του κόσμου, σε μία από τις τέσσερις οθόνες) και

επιλέγουμε **File → export text to clipboard** και κάνουμε paste τον κώδικα που μας κράτησε, σε ένα άδειο έγγραφο.

7) Σβήνουμε τα περιττά tags από τον κώδικα που μας έδωσε το vivaty και κρατάμε μόνο το **<Shape>** και όλα τα παιδιά του.
8) Ανοίγουμε με ένα notepad το αρχείο που θα αποτελεί το τελικό μας μοντέλο. Προσθέτουμε τον κώδικα του shape ανάμεσα στα παρακάτω tags.

<u>Αν το αντικείμενο είναι για το πάτωμα</u>

```
<?xml version="1.0" encoding="UTF-8"?>
<X3D profile='Immersive' >

<Scene>
<WorldInfo   DEF='SceneWorldInfo'   title="   info="  />

<Transform DEF="bar_chair1"   translation='0 0 0'>

<Transform   DEF='floor'   rotation="1 0 0 0"
translation='0 0 0'>

<Transform    DEF='Box01_XFORM'  translation='0 0 0'    scale='1 1 1'>
```

<Εδώ ο κώδικας του shape που πήραμε από το vivaty>

```
  </Transform>

 </Transform>
 <PlaneSensor DEF="sensor"/>

</Transform>
 </Scene>
</X3D>
```

<u>Αν το αντικείμενο είναι για τον τοίχο</u>

```
<?xml version="1.0" encoding="UTF-8"?>
<X3D profile='Immersive' >

<Scene>
<WorldInfo   DEF='SceneWorldInfo'   title="   info="  />

<Transform DEF="lamp"  translation='0 0 0'>

<Transform   DEF='wall'    rotation="0 0 0 1.570796"
translation='0 0 0'>

<Transform    DEF='Box01_XFORM'
  translation='0 0 0'    scale='25 25 25'  center="-2.5051 4.15977 16.005">
<Transform DEF="bringitonwall" center="-2.5051 4.15977 16.005" rotation="1 0 0 1.570796">
```

<Εδώ ο κώδικας του shape που πήραμε από το vivaty>

```
   </Transform>
   </Transform>

  </Transform>
 <PlaneSensor DEF="sensor"/>

</Transform>
 </Scene>
</X3D>
```

9) Σώζουμε το έγγραφο κειμένου με το όνομα που θέλουμε να δώσουμε στο μοντέλο και την κατάληξη .x3d

6 Περιγραφή δωματίου και roomDescription.xml

Στο xml που περιέχει την περιγραφή του δωματίου (models/settings/ roomDescription.xml) , στη διαδρομή *//room/model3D/path* μπορούμε να ορίσουμε ένα αρχείο X3D για να φορτωθεί ως δωμάτιο στον κόσμο, ή στη θέση *//room/shape* να ενσωματώσουμε τον SVG κώδικα με την κάτοψη του δωματίου το οποίο θα μετατραπεί σε 3D όπως είδαμε στην πρώτη φάση της πτυχιακής.

```
<room MoodCharacterization="mood">
        <model3D>
            <coordinates>x,y,z virtual world</coordinates>
            <path>room.x3d</path>
        </model3D>
        <shape>SVG CODE</shape>
</room>
```

Με παρόμοιο τρόπο τοποθετούνται και τα αντικείμενα στη θέση που περιγράφει το XML.

Τα αντικείμενα που θέλουμε να είναι τοποθετημένα στο πάτωμα του δωματίου *//floor/furniture/model3D/path*

Και η θέση x,y πάνω στο πάτωμα *//floor/furniture/model3D/coordinates*

```
<floor>
        <model3D>
            <coordinates>10, 2</coordinates>
            <path>chair.x3d</path>
        </model3D>
        <shape>SVG something</shape>
        <dimensions>number,number,number-|x|,|y|,|z|  bounding box</dimensions>
        <RelativePosition ReferenceObject="another object">left right above below across infront behind
center top bottom abut</RelativePosition>
        <material>
            <materialType materialIndexName="material index name" materialIndexURL="URL of
Material Index">code</materialType>
            <pattern patternIndexName="style index name" patternIndexURL="URL of XML of pattern
Index">pattern index ID that expresses the details shapes</pattern>
        </material>
        <ColorPalette ColorPaletteIndexName="palette index name" ColorPaletteIndexURL="URL of
palette Index">code</ColorPalette>
        </floor>
```

Αντίστοιχα τα αντικείμενα που τοποθετούνται στον τοίχο του δωματίου βρίσκονται στο *//wall/furniture/model3D/path*

Και για την x,y θέση πάνω στον τοίχο *//wall/furniture/model3D/coordinates*

```
<wall>
        <model3D>
            <coordinates>10, 2</coordinates>
            <path>wall-lamp.x3d</path>
        </model3D>
        <shape>SVG something</shape>
        <dimensions>number,number,number-|x|,|y|,|z|  bounding box</dimensions>
        <RelativePosition ReferenceObject="another object">left right above below across infront
behind center top bottom abut</RelativePosition>
        <material>
            <materialType materialIndexName="material index name" materialIndexURL="URL of
Material Index">code</materialType>
            <pattern patternIndexName="style index name" patternIndexURL="URL of XML of
pattern Index">pattern index ID that expresses the details shapes</pattern>
        </material>
        <ColorPalette ColorPaletteIndexName="palette index name" ColorPaletteIndexURL="URL
of palette Index">code</ColorPalette>
</wall>
```

7 Περιγραφή λειτουργίας προγράμματος

Το project είναι γραμμένο σε Netbeans, και περιλαμβάνει τις απαραίτητες βιβλιοθήκες στον υπο-φάκελο libs. Απαραίτητη προϋπόθεση είναι η τοποθέτηση jars στον φάκελο ext που βρίσκεται στη θέση ¨*C:\Java\jdk1.6.0_16\jre\lib\ext*¨

Τα jars συμπεριλαμβανομένων των αρχικών που προϋπήρχαν στο φάκελο είναι :

- Dnsns.jar
- Gluegen-rt.dll
- Gluegen-rt.jar
- J3dcore.jar
- J3dutils.jar
- Jinput.jar
- Jinput-dx8.dll
- Jinput-raw.dll
- Jinput-wintab.dll
- Jogl.dll
- Jogl.jar
- Jogl_awt.dll
- Jogl_cg.dll
- Locatedata.jar
- Log4j.jar
- Odejava.dll
- Odejava.jar
- Sunjce_provider.jar
- Sunmscapi.jar
- Sunpkcs11.jar
- Vecmath.jar

8 Java περιγραφή κλάσεων

1. <default package>

8.1 BuildingPanelSC

Ενεργοποιείται με την κλήση που κάνει το script του X3D όταν ο Browser ανοίξει το αρχείο και εμφανίσει το δωμάτιο. Ακολουθεί τα interfaces **3DScriptImplementation** και **X3DFieldEventListener** που προσφέρουν τις μεθόδους που ενεργοποιούνται από τον Browser όταν γίνεται η αναπαραγωγή του κόσμου.

*public void **setBrowser**(Browser browser)*
*public void **setFields**(X3DScriptNode externalView, Map fields)*
*public void **initialize**()*
*public void **eventsProcessed**()*

Οι τέσσερις παραπάνω μέθοδοι εξασφαλίζουν την ένωση του X3D με την Java εφαρμογή τραβώντας αντικείμενα από τον 3D κόσμο όπως το script και τα Fields τα οποία χρησιμοποιούμε στη συνέχεια για τις λειτουργίες τις τοποθέτησης, μετακίνησης και περιστροφής των αντικειμένων.

*public void **InsertFurniture**(WorldManager myWorldManager)*

Αυτή η μέθοδος διαβάζοντας τα κατάλληλα παιδία στο XML που περιγράφει το δωμάτιο καλεί τις επόμενες κλάσεις δίνοντας ορίσματα για το ποια αντικείμενα θέλουμε να δημιουργήσουμε και σε ποιο σημείο. Για τη σωστή λειτουργία της μεθόδου ελέγχουμε κάθε αντικείμενο αν έχει τις δικές του συντεταγμένες στο χώρο που πρόκειται να τοποθετηθεί. Στη συνέχεια για κάθε αντικείμενο που βρίσκουμε ορίζουμε το scale rotation και translation. Οι τιμές που δεν προέρχονται από το XML και θέλουμε να τις ορίσουμε θα πρέπει πάντα να υπολογίζονται ανάλογα με τις τιμές που έχουν τα αντικείμενα (τα headers και tails που τοποθετήσαμε σε κάθε μοντέλο κατά την δημιουργία του). *Σε περίπτωση που το μέγεθος του μοντέλου είναι κατά πολύ μεγαλύτερο για το δωμάτιο θα πρέπει να κάνουμε scale χρησιμοποιώντας το vivaty studio, ακλουθώντας ξανά την διαδικασία για το export, γιατί αν συρρικνώσουμε το αντικείμενο δίνοντας scale στο transform που ορίζει αυτή η μέθοδος, τότε συρρικνώνουμε και την επιρροή του touch*

Sensor με αποτέλεσμα να καθιστά δύσκολη την μετακίνηση του από τον χρήστη.

public ArrayList<String> ***getValuesFromXPath****(Element root, String path)*

Κατά τη λειτουργία της παραπάνω μεθόδου καλείται η getValuesFromXPath για να μας επιστρέψει ένα πίνακα που περιέχει τις τιμές που ζητήσαμε να δούμε από το XML.

public org.dom4j.Document ***convert2Dom4j****(org.w3c.dom.Document element)*

Για μετατροπή των XML από w3c.dom.Document σε Dom4j, το οποίο μας δίνει δυνατότητα χρήσης XPath.

```
public ArrayList<String> getValuesFromXPath(Element root,String path){
   ArrayList<String> values=new ArrayList<String>();
 XPath xp=DocumentHelper.createXPath(path);
       java.util.List results=xp.selectNodes(root);
       for(Iterator iter=results.iterator();iter.hasNext();){
       Element element=(Element) iter.next();

          values.add(element.getTextTrim());
       }

       return values;
}
```

Έχουμε δύο πίνακες modelValue και positionValue που περιέχουν το αντικείμενο και τη θέση στον κόσμο αντίστοιχα. Οι δύο πίνακες πρέπει να έχουν το ίδιο πλήθος στοιχείων για να είναι σωστή η αντιστοιχία. Φροντίζουμε ανάλογα σε όλες τις περιπτώσεις, τα αντικείμενα, να έχουν διαφορετικό ID ξεκινώντας με τη λέξη wall ή floor (ανάλογα που θα τοποθετηθεί, το όνομα του μοντέλου και ένας αύξον αριθμός από το μετρητή της while που τρέχει εκείνη τη στιγμή.)

8.2 MyX3DBrowser

Η αρχή της εφαρμογής είναι η εκτέλεση της main που περιέχεται εδώ. Ξεκινάει διαβάζοντας το XML με την περιγραφή (roomDescription.xml). Σε περίπτωση που υπάρχει αρχείο X3D για δωμάτιο, φορτώνεται αυτό στον browser που ανοίγει, διαφορετικά διαβάζει το Shape tag που περιέχει τον κώδικα SVG με την κάτοψη του δωματίου και καλεί τις κατάλληλες μεθόδους για την μετατροπή του σε X3D. Το αποτέλεσμα αυτό είναι και ο χώρος του δωματίου που θα εμφανιστεί στον χρήστη.

Όμοια με παραπάνω έχουμε τις μεθόδους για την μετατροπή του Dom4j σε w3c.dom και αντίστροφα, όπως και την μέθοδο getValuesFromXPath για την εκτέλεση των XPath στο τρέχον Document.

2. interiorDesign

8.3 WorldManager

Στην κλάση αυτή γίνεται το κτίσιμο των αντικειμένων. Αρχικοποιούμε τιμές για rotation scale και translation ώστε να αρχικοποιήσουμε την θέση των αντικειμένων στον κόσμο.

public synchronized void ***enqueueNewBuilding****(String type, String ID,*

float[] translation, float[] scale, float[] rotation)
Για την εισαγωγή νέας κατασκευής στην ουρά καλούμε αυτή τη μέθοδο με ορίσματα το όνομα του αρχείου X3D που βρίσκεται στο φάκελο models, το id που θέλουμε να δώσουμε , translation scale και rotation.
Στην αρχή της μεθόδου διαβάζουμε το αντίστοιχο πεδίο από το xml με την περιγραφή και ανάλογα τοποθετούμε τα αντικείμενα.

Για την μοναδικότητα των DEF πεδίων που δημιουργούμε σε επόμενο στάδιο χρησιμοποιούμε το αντικείμενο : VMID myID = new VMID();
Γνωρίζοντας από την **WorldManager** ότι το id κάθε αντικειμένου ξεκινάει με wall ή floor ανάλογα με την αρχική τοποθέτησή του, ορίζουμε το αρχικό rotation να είναι μηδέν αν το αντικείμενο είναι για να τοποθετηθεί στο πάτωμα, αλλιώς το περιστρέφουμε έτσι ώστε να ταιριάζει στον τοίχο που πρέπει να τοποθετηθεί.

Αφού περάσουμε αυτά τα ορίσματα δημιουργούμε ένα αντικείμενο MessageCreation και το δίνουμε σαν όρισμα στην **queueEvent** για να ¨δηλώσουμε¨ την επιθυμία μας για το κτίσιμο του αντικειμένου.
Στη συνέχεια γίνεται η ¨προσφορά¨ τοπικά : this.**pendingEvents**.offer(msg) οπότε η **processLocalEvents** μόλις ανιχνεύσει κάποιο αντικείμενο στην ουρά συνεχίζει τη διαδικασία ελέγχοντας εάν το αντικείμενο αυτό έχει δημιουργηθεί, αν όχι, προωθεί το μήνυμα στην addNewBuilding.

*private synchronized void **addNewBuilding**(MessageCreation msg)*
Στο σημείο αυτό παίρνουμε από το μήνυμα το id και το όνομα του μοντέλου για το οποίο θέλουμε να χτίσουμε και να ενσωματώσουμε το X3D στον τρέχον κόσμο.

Αρχικά δημιουργούμε ένα αντικείμενο Building με όρισμα το μήνυμα που μόλις λάβαμε για να διαβάσουμε το αρχείο με το μοντέλο που θέλουμε, και αρχίζουμε πάνω σε αυτό να ¨χτίζουμε¨ ένα αντίγραφο του μοντέλου προσθέτοντας δικά μας αντικείμενα όπως το κουμπί που θα εμφανίζεται πάνω από το αντικείμενο για την περιστροφή του, αλλά και Translations όπου θα τοποθετήσουμε δυναμικά CylinderSensors. Αναλυτικότερα θα τα δούμε στην περιγραφή της κλάσης αυτής.

Αφού ολοκληρωθούν οι διαδικασίες με το μοντέλο το τοποθετούμε στην σκηνή και προχωράμε στην προσθήκη ROUTES τώρα που γνωρίζουμε τα DEF των nodes που μόλις προσθέσαμε. Τα ROUTES αυτά είναι για την μεταφορά των τιμών μετακίνησης και περιστροφής από τον αισθητήρα στο αντικείμενο, και να του δώσει drag and drop ιδιότητες.

3. interiorDesign.buildings

8.4 Building

Όπως αναφέρθηκε παραπάνω η κλάση αυτή δημιουργεί το μοντέλο για το οποίο θέλουμε να ¨χτίσουμε¨ το τελικό X3D που θα ενσωματώσουμε στον κόσμο. Κατά τη δημιουργία του αντικειμένου της κλάσης, χρησιμοποιώντας τον XMLHelper που θα περιγράψουμε παρακάτω, διαβάζουμε από τον φάκελο models το αρχικό X3D, με όνομα αυτό που περάσαμε σαν όρισμα στον constructor της κλάσης. Στο σημείο αυτό διαμορφώνουμε τα transform του αντικειμένου για να φέρουμε τους αισθητήρες στη θέση που θέλουμε για να γίνεται η περιστροφή και η μετακίνηση στους σωστούς άξονες.

Επίσης προσθέτουμε ένα script node στο αντικείμενο για να έχουμε πρόσβαση στα πεδία που θέλουμε να αλλάζουν τιμές από τον αντίστοιχο αισθητήρα κάθε στιγμή.

Τέλος προσθέτουμε το διάφανο αντικείμενο που θα βρίσκεται πάνω από το αντικείμενο και θα περιλαμβάνει τον αισθητήρα περιστροφής, τοποθετημένος έτσι ώστε να δίνει τιμές κατάλληλες για την περιστροφή του αντικειμένου.

4. interiorDesign.msg

8.5 Message

Η abstract κλάση αυτή περιλαμβάνει τους getters και setters για την δημιουργία του μηνύματος που περιγράψαμε νωρίτερα.

8.6 MessageCreation

Εδώ υπάρχουν τα υπόλοιπα ορίσματα του μηνύματος για τα οποία έχουμε πρόσβαση από εξωτερικά πακέτα της εφαρμογής (public). Η τιμή του setLocked μπορεί να κληθεί και να ορίσει αν ο αισθητήρας μετακίνησης θα είναι ενεργοποιημένος ή όχι.

5. interiorDesign.xml

8.7 XMLHelper

Η κλάση αυτή χρησιμοποιείται για την προσπέλαση κάποιου xml αρχείου. Το Document που δημιουργεί είναι τύπου w3c.dom και η διαδρομή που θα ακολουθήσει ξεκινάει από το σημείο που βρίσκεται το project. Σε περίπτωση που θέλουμε να σώσουμε κάποιο xml είτε για να δούμε τη δομή του είτε για να αποθηκεύσουμε δεδομένα, μπορούμε να καλέσουμε από την κλάση αυτή την μέθοδο saveXML.

6. svg2x3d

8.8 GetSvg

Από την πρώτη φάση της πτυχιακής χρησιμοποιούμε την κλάση GetSvg. Στην περίπτωση που το XML με την περιγραφή ορίζει SVG αντικείμενα για την περιγραφή του δωματίου, παίρνουμε τα αντικείμενα αυτά χρησιμοποιώντας τις μεθόδους **convert2Dom4j** και **convert2w3c**

```
private Element createX3D(Document svgdoc) throws
TransformerConfigurationException, TransformerException,
FileNotFoundException, IOException {
try {
        db = dbf.newDocumentBuilder();
        document = db.newDocument();

      } catch (ParserConfigurationException e) {
        e.printStackTrace();
      }

      document.setXmlVersion("1.0");
      document.setTextContent("X3D");

      Element X3D = document.createElement("X3D");
      X3D.setAttribute("profile", "Immersive");
…
}
   public void setWallHeight(Double wallHeight) {
      this.wallHeight = wallHeight;
      floorPotition = (wallHeight / 2) + 0.01;
   }

 public Element Get3D(Document doc) {
      try {
     return  final3D= createX3D(doc);
      } catch (Exception ex) {
        ex.printStackTrace();
      }
      return null;
   }
```

```
public org.dom4j.Document convert2Dom4j( org.w3c.dom.Document element)
throws ParserConfigurationException {


  org.w3c.dom.Document doc1 = element;


  // Convert w3c document to dom4j document
  DOMReader reader = new DOMReader();
  org.dom4j.Document doc2 = reader.read( doc1);

  return doc2;
}
   public org.w3c.dom.Document convert2w3c( String element) throws
DocumentException, ParserConfigurationException, IOException {
        DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
      DocumentBuilder builder = factory.newDocumentBuilder();

org.w3c.dom.Document doc2 = null;
       try {
          doc2 = builder.parse(new InputSource(new StringReader(element)));
       } catch (SAXException ex) {
          Logger.getLogger(MyX3DBrowser.class.getName()).log(Level.SEVERE,
null, ex);
       }
  return doc2;
```

Το X3D που χρησιμοποιούμε για τον χώρο του δωματίου θα πρέπει να περιλαμβάνει κάποια nodes που έχουμε φροντίσει να υπάρχουν τα κατά τη μετατροπή του από SVG.

1) Profile για τη δήλωση του προφίλ που θα χρησιμοποιήσουμε

```
<X3D profile="Immersive">
```

2) Background για την ευπαρουσίαστη προβολή του δωματίου

```
<Background DEF="background" groundAngle=".03491 1.5708"
groundColor=".39216 .65098 .93333 .4902 .61961 .92549 .90196
.90196 1" skyColor=".39216 .65098 .93333 .4902 .61961 .92549
.90196 .90196 1"/>
```

3) Viewpoint για την ενεργοποίηση της κάμερας στη θέση που θέλουμε.

```
<Viewpoint DEF="DefaultCam"
description="DefaultCam"
jump="TRUE"
fieldOfView="0.785"
position='17.50242 18.08865 50.61155'
orientation="-1 -0 0 .579" />
```

4) Script για να γίνεται η κλήση της BuildingPanelSC και να ξεκινάει η διαδικασία εισαγωγής αντικειμένων στον κόσμο.

```
Script DEF="SC" directOutput="TRUE"
mustEvaluate="TRUE"
url="../build/classes/BuildingPanelSC.class"/>
```

5) Για την ακριβή τοποθέτηση του δωματίου σε σχέση με την κάμερα μπορούμε να περικλύσουμε τα αντικείμενα που αποτελούν το δωμάτιο σε ένα εξωτερικό transform δίνοντας τις ανάλογες τιμές στο πεδίο transform.

```
<Transform translation="x y">
```

9 Περιγραφή X3D σύνθετου μοντέλου

Δύο απλά μοντέλα σε X3D περιγράφονται παρακάτω σαν σημείο αναφοράς για τη δημιουργία αντίστοιχων μοντέλων, ένα που τοποθετείται στον τοίχο και ένα στο πάτωμα.

9.1 Μοντέλο καρέκλας

```
<?xml version="1.0" encoding="UTF-8"?>
<X3D profile='Immersive' >

<Scene>
<WorldInfo   DEF='SceneWorldInfo'   title="   info="  />

<Transform DEF="bar_chair1"   translation='0 0 0'>

<Transform   DEF='floor'   rotation="1 0 0 0"
translation='0 0 0'>

<Transform    DEF='Box01_XFORM'  translation='0 0 0'    scale='1 1 1'>

<Shape DEF='bar-chair'>
 <Appearance
  containerField='appearance'>
  <Material DEF='material0'
   containerField='material'
   ambientIntensity='0.486'
   shininess='0.400'
   diffuseColor='.48628 .48628 .48628'
   specularColor='.89804 .89804 .89804'/>
 </Appearance>
 <IndexedFaceSet DEF='bar-chair_Geo'
  containerField='geometry'
  creaseAngle='0.000'
  coordIndex='
    0 1 2 -1
  0 2 3 -1
    0 3 4 -1
    0 4 5 -1
    0 5 6 -1
    0 6 7 -1
    0 7 8 -1
    0 13 14 -1
    0 14 15 -1
…
```

```
…
   239 238 241 -1
   240 239 241 -1
   225 240 241 -1'>
  <Coordinate DEF='bar-chair_Coord'
   containerField='coord'
   point='
    .00049 -2.57062 -.00195
    .00049 -2.57062 -55.88379
    21.38477 -2.57062 -51.62988
…
```

```
…
-5.07959 143.98737 -.00195
    -4.69287 143.98737 -1.94727
    -3.5918 143.98737 -3.59473
    -1.94385 143.98737 -4.69629
    .00049 143.98737 -.00195'/>
 </IndexedFaceSet>
</Shape>
<Shape DEF='bar-chair-_1'>
 <Appearance
  containerField='appearance'>
  <Material DEF='material1'
   containerField='material'
   ambientIntensity='0.129'
   shininess='0.160'
   diffuseColor='.12941 .12941 .12941'
   specularColor='.83922 .83922 .83922'/>
 </Appearance>
 <IndexedFaceSet DEF='bar-chair-_1_Geo'
  containerField='geometry'
  creaseAngle='0.000'
  coordIndex='
    30 28 26 -1
    30 26 24 -1
    30 24 22 -1
    30 22 20 -1
    30 20 18 -1
    30 18 16 -1
…
```

```
…
  26 29 27 -1
    28 30 31 -1
    28 31 29 -1'>
  <Coordinate DEF='bar-chair-_1_Coord'
   containerField='coord'
   point='
    5.08032 143.98737 -.00195
    5.08032 145.25736 -.00195
    4.6936 143.98737 -1.94727
    4.6936 145.25736 -1.94727
…
```

```
…
3.59253 143.98737 3.58984
    3.59253 145.25736 3.58984
    4.6936 143.98737 1.94141
    4.6936 145.25736 1.94141'/>
 </IndexedFaceSet>
</Shape>
<Shape DEF='bar-chair-_2'>
 <Appearance
  containerField='appearance'>
  <Material DEF='material2'
   containerField='material'
   ambientIntensity='0.534'
   shininess='0.030'
   diffuseColor='.52549 .53333 .56078'
   specularColor='.44314 .44314 .44314'/>
 </Appearance>
 <IndexedFaceSet DEF='bar-chair-_2_Geo'
  containerField='geometry'
  creaseAngle='0.000'
  coordIndex='
    1 0 13 -1
    1 13 14 -1
    2 1 14 -1
…
```

```
…
  119 115 117 -1'>
  <Coordinate DEF='bar-chair-_2_Coord'
   containerField='coord'
   point='
    .00049 68.80338 46.22461
…
```

```
…
  3.06104 58.49771 4.05078
    40.53516 69.2865 25.6875
    3.38452 56.76273 3.78516
    41.09155 67.61847 25.55567'/>
 </IndexedFaceSet>
</Shape>
<Shape DEF='bar-chair-_3'>
 <Appearance
  containerField='appearance'>
  <Material DEF='material3'
   containerField='material'
   ambientIntensity='0.432'
   shininess='0.010'
   diffuseColor='.58039 .41176 .19608'
   specularColor='.89804 .89804 .89804'/>
 </Appearance>
 <IndexedFaceSet DEF='bar-chair-_3_Geo'
  containerField='geometry'
  creaseAngle='0.000'
  coordIndex='
    0 1282 264 -1
    264 723 0 -1
    1 722 264 -1
    264 1282 1 -1
…
```

```
…
   263 1428 721 -1
   721 1430 263 -1'>
  <Coordinate DEF='bar-chair-_3_Coord'
   containerField='coord'
   point='
    -2.11316 256.16935 -50.89184

…

  -47.36115 171.06814 -22.25268'/>
 </IndexedFaceSet>
</Shape>

   </Transform>

  </Transform>
 <PlaneSensor DEF="sensor"/>

</Transform>
 </Scene>
</X3D>
```

Η δομή της παραπάνω καρέκλας περιλαμβάνει το header και tail που περιγράψαμε νωρίτερα, ενώ στη μέση του κώδικα, υπάρχει ένα node τύπου Shape το οποίο περιέχει 3 IndexedFacedSet, τα οποία περιγράφουν τα 3 κομμάτια της καρέκλας, κάθισμα, κορμός, βάση.
Για κάθε ένα από αυτά έχουμε ένα μεγάλο σετ συντεταγμένων που περιγράφουν τα τρίγωνα που το αποτελούν. Με παρόμοιο τρόπο δομείται και το αντικείμενο του τοίχου.

9.2 Μοντέλο λάμπας τοίχου

```
<?xml version="1.0" encoding="UTF-8"?>
<X3D profile='Immersive' >

<Scene>
<WorldInfo   DEF='SceneWorldInfo'   title="   info="  />

<Transform DEF="lamp"  translation='0 0 0'>

<Transform   DEF='wall'    rotation="0 0 0 1.570796"
translation='0 0 0'>

<Transform    DEF='Box01_XFORM'
  translation='0 0 0'    scale='25 25 25'  center="-2.5051 4.15977 16.005">
<Transform DEF="bringitonwall" center="-2.5051 4.15977 16.005" rotation="1 0 0 1.570796">


<Shape DEF='carpyen_01'>
 <Appearance
  containerField='appearance'>
  <ImageTexture
   containerField='texture'
   url='
    "texture/yellowhenna2.jpg"'/>
  <Material DEF='material0'
   containerField='material'
   ambientIntensity='0.711'
   shininess='0.400'
   diffuseColor='.96863 .84706 .58823'
   specularColor='.98431 .92157 .78431'/>
 </Appearance>
 <IndexedFaceSet DEF='carpyen_01_Geo'
  containerField='geometry'
   solid='false'
  creaseAngle='0.000'
  coordIndex='
    29 0 91 -1
    30 91 5 -1
    0 102 1 -1
    91 1 92 -1

…
```

```
…
  526 660 644 -1
   644 643 526 -1'
 texCoordIndex='
   220 250 246 -1
   217 246 247 -1
…
```

```
…

   6 12 15 -1'>
  <Coordinate DEF='carpyen_01_Coord'
   containerField='coord'
   point='
    -2.61292 6.82136 16.10632
    -2.59198 6.73521 16.09479
    -2.61292 6.93795 16.11178

…

  -2.5886 9.69466 14.57745
    -2.75007 9.69466 14.58592
    -2.5886 5.52836 13.56252
    -2.5886 5.54804 13.45084'/>
  <TextureCoordinate
   containerField='texCoord'
   point='
    1 .994
    1 .997
…

  2.012 1.662
    2.012 1.674'/>
 </IndexedFaceSet>
</Shape>
<Shape DEF='carpyen_01_1'>
 <Appearance
  containerField='appearance'>
  <ImageTexture
   containerField='texture'
   url='
    "texture/yellowhenna2.jpg"'/>
```

```
  <Material DEF='material1'
   containerField='material'
   ambientIntensity='0.588'
   shininess='0.100'
   diffuseColor='.58823 .58823 .58823'
   specularColor='.89804 .89804 .89804'/>
 </Appearance>
 <IndexedFaceSet DEF='carpyen_01_1_Geo'
  containerField='geometry'
  creaseAngle='0.000'
  coordIndex='
    0 5 6 -1
    6 7 15 -1
…

39 40 2 -1
    41 43 40 -1'
  texCoordIndex='
    49 52 54 -1
    54 53 55 -1

…

46 11 12 -1
    46 32 47 -1
    36 41 32 -1'>
  <Coordinate DEF='carpyen_01_1_Coord'
   containerField='coord'
   point='
    -2.54785 2.23228 13.9214
    -2.54658 4.99308 15.95495
…

  -2.63063 2.47969 13.67399
    -2.6691 4.58725 16.26287'/>
  <TextureCoordinate
   containerField='texCoord'
   point='
    .97 .534

…
```

```
…
2.002 .459
    2.033 .286'/>
 </IndexedFaceSet>
</Shape>
<Shape DEF='carpyen_01_2'>
 <Appearance
  containerField='appearance'>
  <ImageTexture
   containerField='texture'
   url='
    "texture/yellowhenna2.jpg"'/>
  <Material DEF='material2'
   containerField='material'
   ambientIntensity='0.667'
   shininess='0.100'
   diffuseColor='.67843 .66667 .63922'
   specularColor='.89804 .89804 .89804'/>
 </Appearance>
 <IndexedFaceSet DEF='carpyen_01_2_Geo'
  containerField='geometry'
  creaseAngle='0.000'
  coordIndex='
    260 252 258 -1
    258 259 260 -1
…

  428 436 353 -1'
  texCoordIndex='
    259 238 242 -1
    242 255 259 -1

…

414 424 27 -1
    14 414 27 -1
    14 27 29 -1'>
  <Coordinate DEF='carpyen_01_2_Coord'
   containerField='coord'
   point='
    -2.54658 2.4677 13.31151
    -2.63063 5.07805 16.27236
    -2.82069 2.15347 13.31151
    -2.77279 2.54444 13.60924
…
```

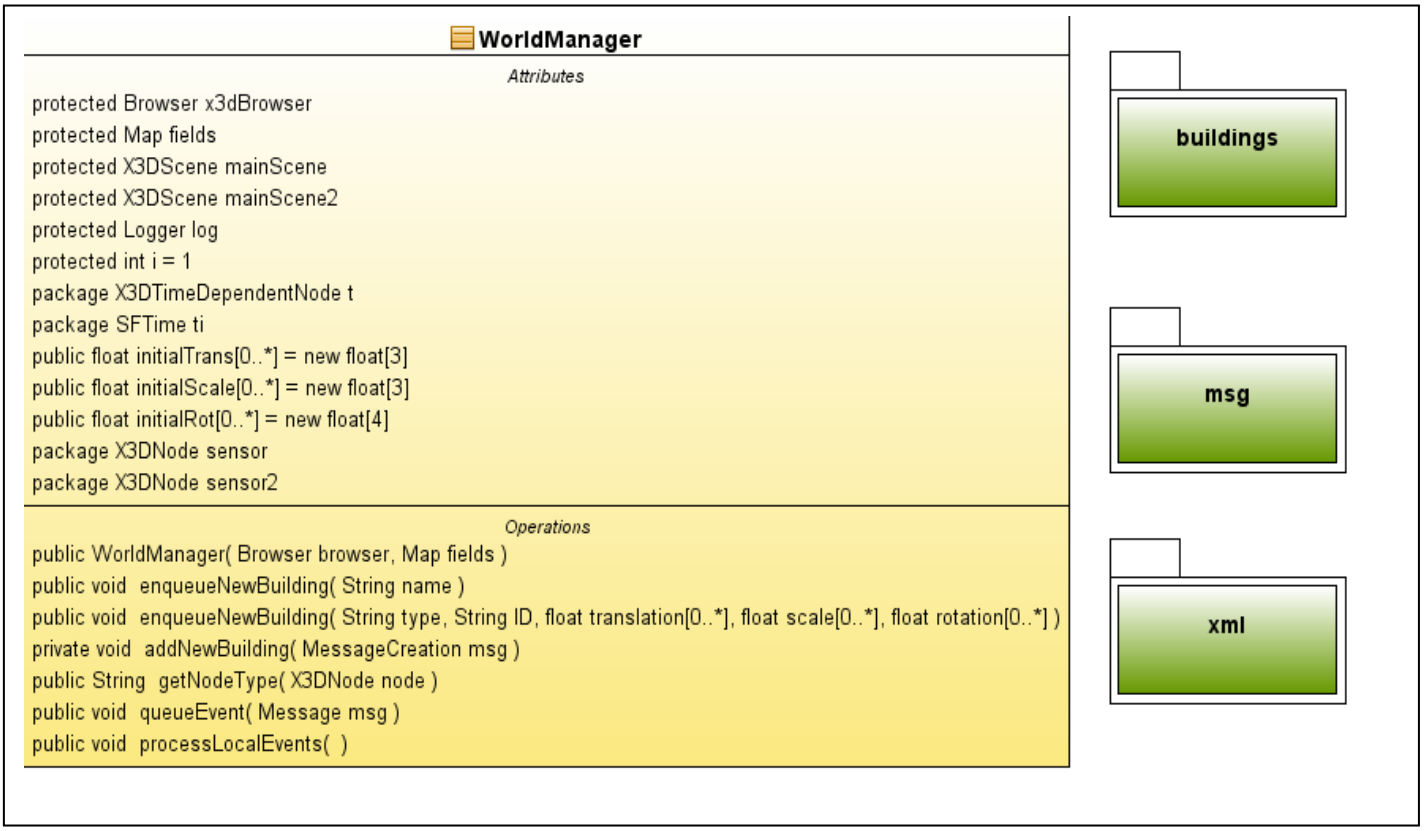
```
…

-2.5886 2.23002 13.92367
    -2.54785 1.92734 13.61645'/>
  <TextureCoordinate
   containerField='texCoord'
   point='
    .959 .402
    .96 .381

…


  2.474 .92
    2.485 .88
    2.515 .88'/>
 </IndexedFaceSet>
</Shape>
<Shape DEF='carpyen_gi'>
 <Appearance
  containerField='appearance'>
  <Material DEF='material3'
   containerField='material'
   ambientIntensity='0.078'
   shininess='0.100'
   diffuseColor='.2902 .2902 .2902'
   specularColor='.89804 .89804 .89804'/>
 </Appearance>
 <IndexedFaceSet DEF='carpyen_gi_Geo'
  containerField='geometry'
  creaseAngle='0.000'
  coordIndex='
    51 0 1 -1
    1 41 51 -1

…

   161 125 151 -1'>
  <Coordinate DEF='carpyen_gi_Coord'
   containerField='coord'
   point='
    -2.5886 4.94661 15.90351
…
```

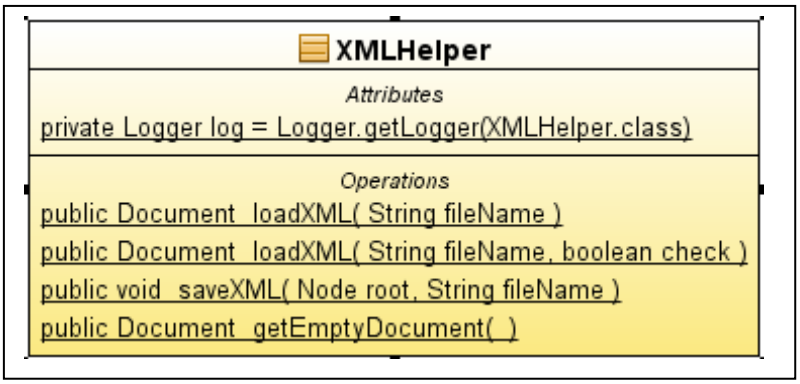
```
…

-2.82759 7.46599 16.31132
   -2.67989 7.46599 16.41864'/>
 </IndexedFaceSet>
</Shape>
<Shape DEF='carpyen_gi_1'>
 <Appearance
  containerField='appearance'>
  <Material DEF='material4'
   containerField='material'
   ambientIntensity='1.000'
   shininess='0.100'
   diffuseColor='1 1 1'
   specularColor='1 1 1'/>
 </Appearance>
 <IndexedFaceSet DEF='carpyen_gi_1_Geo'
  containerField='geometry'
  creaseAngle='0.000'
  coordIndex='
    237 236 1 -1
    2 1 0 -1


 130 265 186 -1
    130 237 2 -1'>
  <Coordinate DEF='carpyen_gi_1_Coord'
   containerField='coord'
   point='
    -2.68866 7.40309 16.31222
    -2.64033 7.40309 16.33224

…

  -2.5886 7.2635 16.01141'/>
 </IndexedFaceSet>
</Shape>
   </Transform>
   </Transform>

  </Transform>
 <PlaneSensor DEF="sensor"/>

</Transform>
 </Scene>
</X3D>
```

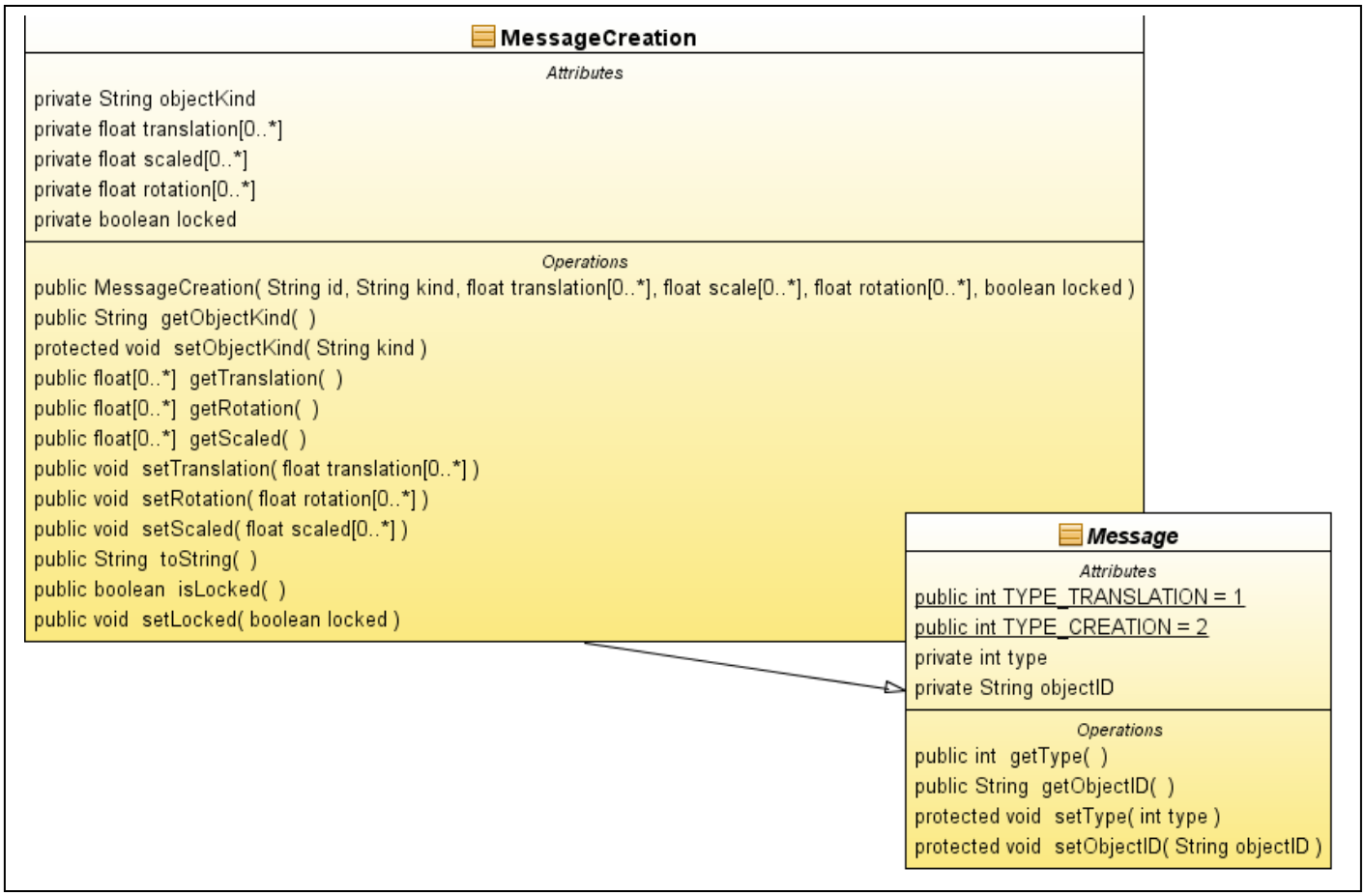
Στο παράδειγμα αυτό εκτός από τα headers και tails μπορούμε να δούμε και την χρήση των textures, που τοποθετήσαμε στο κάλυμμα. Παρατηρούμε πως η περιγραφή της τοποθέτησης του texture είναι ίδια με την περιγραφή των τριγώνων που αποτελούν το σχήμα του IndexedFaceSet.

10 Class Diagrams

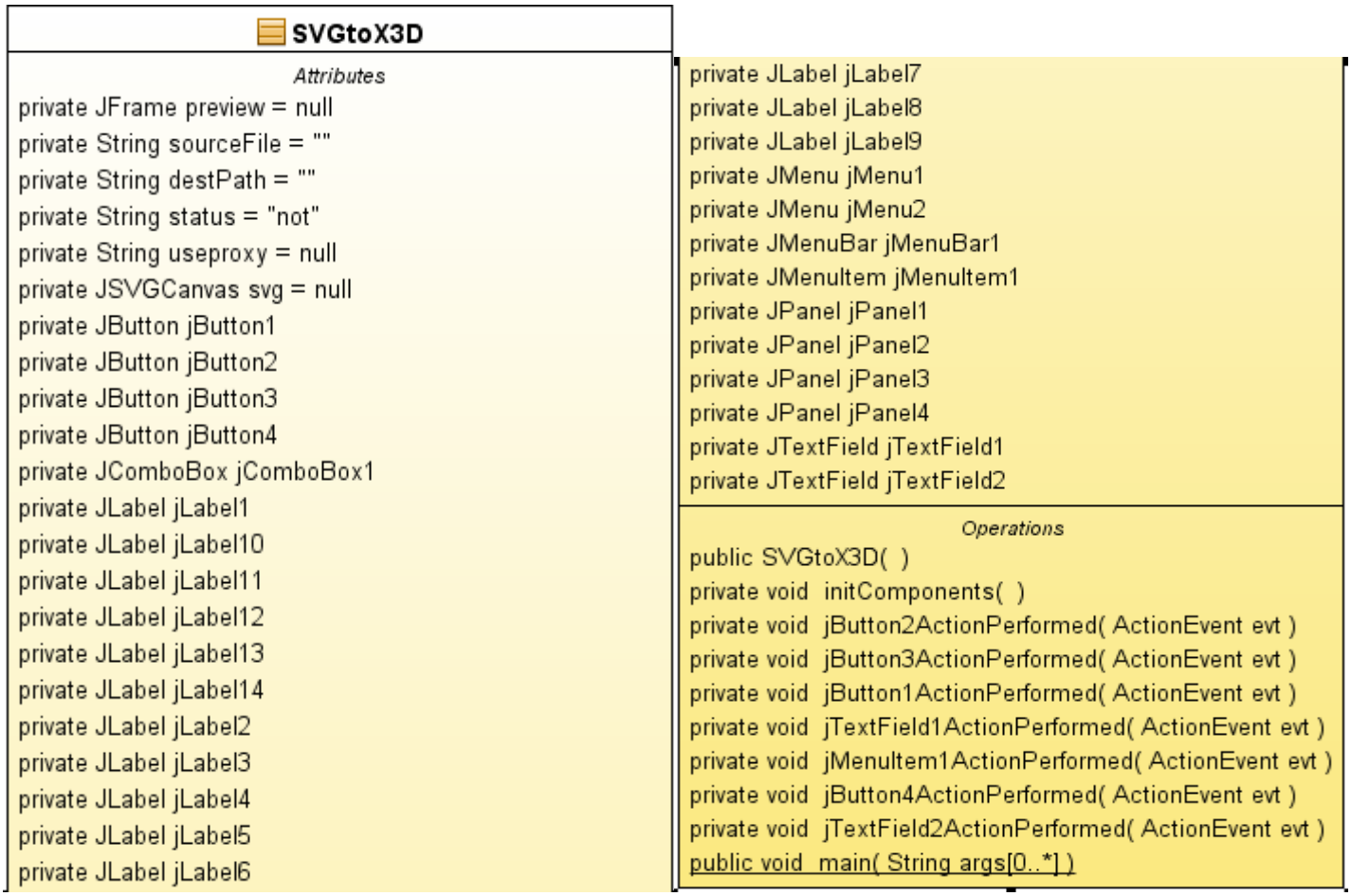
10.1 Interior Design






MessageCreation
Attributes
private String objectKind
private float translation[0..*]
private float scaled[0..*]
private float rotation[0..*]
private boolean locked
Operations
public MessageCreation(String id, String kind, float translation[0..*], float scale[0..*], float rotation[0..*], boolean locked)
public String getObjectKind( )
protected void setObjectKind(String kind)
public float[0..*] getTranslation( )
public float[0..*] getRotation( )
public float[0..*] getScaled( )
public void setTranslation(float translation[0..*])
public void setRotation(float rotation[0..*])
public void setScaled(float scaled[0..*])
public String toString( )
public boolean isLocked( )
public void setLocked(boolean locked)
Message
Attributes
public int TYPE_TRANSLATION = 1
public int TYPE_CREATION = 2
private int type
private String objectID
Operations
public int getType( )
public String getObjectID( )
protected void setType(int type)
protected void setObjectID(String objectID)

10.2 SVG to X3D converter



GetSvg

Attributes

private Document document = null
public int scaling
public Double wallHeight
public Double StrokeWidth
public Double floorPotition
private String srcpath
private String destpath

Operations

public GetSvg( )
public void setWallHeight(Double wallHeight)
public void setDestpath(String destpath)
public void setSrcpath(String srcpath)
public void run( )
private void writeX3dFile(String destpath, Document svgdoc)
private void addText(String Color, Double rotation, Double transformX, Double transformY, Double centerX, Double centerY, String fontf, String fonts, int m, Element Scene, String text)
private void addCurve(String Color, Element Scene, int m, Double rotation, Double Mx, Double My, Double C1x, Double C1y, Double C2x, Double C2y, Double C3x, Double C3y, Double transformX, Double transformY)
private void addCircle(String Color, Double rotation, Double transformX, Double transformY, Double centerX, Double centerY, Double radius, int m, Element Scene)
public void addRect(String Color, Double transformX, Double transformY, Double x1, Double y1, Double rotation, Double width, Double height, int m, Element Scene)
public void addLine(Double transformX, Double transformY, Double x1, Double x2, Double y1, Double y2, Double rotation, int m, String Color, Element Scene)
public void addElement(String element, Double transformX, Double transformY, Double cx, Double cy, Double rotation, Double width, Double height, int m, Element Scene)
public void setScaling(int scaling)

svgfilter

Attributes

Operations

public boolean accept(File f)
public String getDescription( )

11 API - InteriorDesign

11.1 InteriorDesignJavadoc

Class BuildingPanelSC

BuildingPanelSC

All Implemented Interfaces:

X3DScriptImplementation, X3DFieldEventListener

All Dependency Suppliers:

Implementation X3DScriptImplementation

Implementation X3DFieldEventListener

All Known Associations:

Navigable_Aggregation: WorldManager

```
public class BuildingPanelSC
implements X3DScriptImplementation, X3DFieldEventListener
```

Properties	
Alias	BuildingPanelSC
Visibility	public
Final	☐
Transient	☑
Abstract	☐
Leaf	☐

Attribute Summary	
Map	**fields** A mapping for fieldName(String) to an X3DField object
Browser	**browser**
X3DScriptNode	**selfRef**
X3DScene	**mainScene** The main scene
Logger	**log** Log4j system
int	**i** Building Count

Operation Summary

void	**setBrowser**(Browser browser)
void	**setFields**(X3DScriptNode externalView, Map fields)
void	**initialize**() Called from the browser to initialize the status of the script
void	**shutdown**() Notification that this script instance is no longer in use by the scene graph and should now release all resources.
void	**eventsProcessed**() Notification that all the events in the current cascade have finished processing.
void	**readableFieldChanged**(X3DFieldEvent evt) ---------- Methods from the X3DFieldEventListener interface.
void	**InsertFurniture**(WorldManager myWorldManager)
String[0..*]	**getValuesFromXPath**(Element root, String path)
Document	**convert2Dom4j**(Document element)

Attribute Detail

11.1.1 fields

private Map **fields**

A mapping for fieldName(String) to an X3DField object

11.1.2 browser

private Browser **browser**

11.1.3 selfRef

private X3DScriptNode **selfRef**

11.1.4 mainScene

private X3DScene **mainScene**

The main scene

11.1.5 log

`private Logger log`

Log4j system

11.1.6 i

`protected int i`

Building Count

Operation Detail

11.1.7 setBrowser

`public void setBrowser(Browser browser)`

11.1.8 setFields

`public void setFields(X3DScriptNode externalView, Map fields)`

11.1.9 initialize

`public void initialize()`

Called from the browser to initialize the status of the script

---

11.1.10 shutdown

`public void shutdown()`

Notification that this script instance is no longer in use by the scene graph and should now release all resources.

11.1.11 eventsProcessed

`public void eventsProcessed()`

Notification that all the events in the current cascade have finished processing.

11.1.12 readableFieldChanged

`public void readableFieldChanged(X3DFieldEvent evt)`

---------- Methods from the X3DFieldEventListener interface. ----------

---

11.1.13 InsertFurniture

`public void InsertFurniture(WorldManager myWorldManager)`

11.1.14 getValuesFromXPath

```
public String[0..*] getValuesFromXPath(Element root, String path)
```

11.1.15 convert2Dom4j

```
public Document convert2Dom4j(Document element)
```

11.2 Class MyX3DBrowser

MyX3DBrowser

```
public class MyX3DBrowser
```

Properties	
Alias	MyX3DBrowser
Visibility	public
Final	☐
Transient	☑
Abstract	☐
Leaf	☐

Attribute Summary	
Element	**root**
boolean	**readSVG**
String	**StringScene**

Constructor Summary
MyX3DBrowser()

Operation Summary	
void	**main**(String args[0..*])

Document	**convert2Dom4j**(Document element)
Document	**convert2w3c**(String element)
String[0..*]	**getValuesFromXPath**(Element root, String path)

Attribute Detail

11.2.1 root

protected Element **root**

11.2.2 readSVG

private boolean **readSVG**

11.2.3 StringScene

private String **StringScene**

Constructor Detail

11.2.4 MyX3DBrowser

public **MyX3DBrowser**()

Operation Detail

11.2.5 main

public static void **main**(String args[0..*])

11.2.6 convert2Dom4j

public Document **convert2Dom4j**(Document element)

11.2.7 convert2w3c

public Document **convert2w3c**(String element)

11.2.8 getValuesFromXPath

public String[0..*] **getValuesFromXPath**(Element root, String path)

11.3 Package interiorDesign

Properties	
Alias	interiorDesign
Visibility	public

Class Summary	
WorldManager	Maintains the status of the actual world.

Package Summary	
buildings	
msg	
xml	

11.4 Class Building

`interiorDesign::buildings::Building`

```
public class Building
```

Properties	
Alias	Building
Visibility	public
Final	☐
Transient	☑
Abstract	☐
Leaf	☐

Attribute Summary	
`Document`	**`buildingDocument`** Document that contains the building elements

String	**str**

Constructor Summary

Building(MessageCreation msg)

Operation Summary

Node	**prepareSphere**(String id)
Node	**prepareScript**(String id)
Node	**getBuildingNode**()
String	**toString**()
String	**getStringFromDocument**(Document doc) method to convert Document to String

Attribute Detail

11.4.1 buildingDocument

protected Document **buildingDocument**

Document that contains the building elements

11.4.2 str

public String **str**

Constructor Detail

11.4.3 Building

public **Building**(MessageCreation msg)

Operation Detail

11.4.4 prepareSphere

protected Node **prepareSphere**(String id)

11.4.5 prepareScript

protected Node **prepareScript**(String id)

11.4.6 getBuildingNode

```
public Node getBuildingNode()
```

11.4.7 toString

```
public String toString()
```

11.4.8 getStringFromDocument

```
public String getStringFromDocument(Document doc)
```

method to convert Document to String

11.5 Package interiorDesign::msg

Properties	
Alias	msg
Visibility	public

Class Summary	
Message	
MessageCreation	

11.6 Package interiorDesign::xml

Properties	
Alias	xml
Visibility	public

Class Summary	
XMLHelper	

11.7 Class XMLHelper

interiorDesign::xml::XMLHelper

public class **XMLHelper**

Properties	
Alias	XMLHelper
Visibility	public
Final	☐
Transient	☑
Abstract	☐
Leaf	☐

Attribute Summary	
Logger	**log**

Operation Summary	
Document	**loadXML**(String fileName)
Document	**loadXML**(String fileName, boolean check)
void	**saveXML**(Node root, String fileName)
Document	**getEmptyDocument**()

Attribute Detail

11.7.1 log

private Logger **log**

Operation Detail

11.7.2 loadXML

```
public static Document loadXML(String fileName)
```

11.7.3 loadXML

```
public static Document loadXML(String fileName, boolean check)
```

11.7.4 saveXML

```
public static void saveXML(Node root, String fileName)
```

11.7.5 getEmptyDocument

```
public static Document getEmptyDocument()
```

11.8 Class Message

interiorDesign::msg::Message

All Implemented Interfaces:

Serializable

All Known Subclasses:

MessageCreation

All Dependency Suppliers:

Implementation Serializable

All Known Associations:

Navigable_Aggregation: WorldManager

```
public class Message
implements Serializable
```

Properties	
Alias	Message
Visibility	public
Final	☐
Transient	☑
Abstract	☑
Leaf	☐

Attribute Summary	
int	**TYPE_TRANSLATION** Message of type "Translation" (When we move a building)
int	**TYPE_CREATION** Message of type "Creation" (When we create a building)

int	**type** Kind of message
String	**objectID** Unique id of the building referenced

Operation Summary

int	**getType**()
String	**getObjectID**()
void	**setType**(int type)
void	**setObjectID**(String objectID)

Attribute Detail

11.8.1 TYPE_TRANSLATION

public int **TYPE_TRANSLATION**

Message of type "Translation" (When we move a building)

11.8.2 TYPE_CREATION

public int **TYPE_CREATION**

Message of type "Creation" (When we create a building)

11.8.3 type

private int **type**

Kind of message

11.8.4 objectID

private String **objectID**

Unique id of the building referenced

Operation Detail

11.8.5 getType

public int **getType**()

11.8.6 getObjectID

public String **getObjectID**()

11.8.7 setType

```
protected void setType(int type)
```

11.8.8 setObjectID

```
protected void setObjectID(String objectID)
```

11.9 Class MessageCreation

Message
└ **interiorDesign::msg::MessageCreation**

All Implemented Interfaces:

Serializable

All Dependency Suppliers:

Implementation Serializable

```
public class MessageCreation
extends Message
implements Serializable
```

Properties	
Alias	MessageCreation
Visibility	public
Final	☐
Transient	☑
Abstract	☐
Leaf	☐

Attribute Summary	
String	**objectKind**
float	**translation**[0..*]
float	**scaled**[0..*]
float	**rotation**[0..*]
boolean	**locked**

	To create objects that you cannot move

Constructor Summary

MessageCreation(String id, String kind, float translation[0..*], float scale[0..*], flo

Operation Summary

String	**getObjectKind**()
void	**setObjectKind**(String kind)
float[0..*]	**getTranslation**()
float[0..*]	**getRotation**()
float[0..*]	**getScaled**()
void	**setTranslation**(float translation[0..*])
void	**setRotation**(float rotation[0..*])
void	**setScaled**(float scaled[0..*])
String	**toString**()
boolean	**isLocked**()
void	**setLocked**(boolean locked)

Attribute Detail

11.9.1 objectKind

private String **objectKind**

11.9.2 translation

private float **translation**[0..*]

11.9.3 scaled

```
private float scaled[0..*]
```

11.9.4 rotation

```
private float rotation[0..*]
```

11.9.5 locked

```
private boolean locked
```

To create objects that you cannot move

Constructor Detail

11.9.6 MessageCreation

```
public
MessageCreation(String id, String kind, float translation[0..*], float
scale[0..*], float rotation[0..*], boolean locked)
```

Operation Detail

11.9.7 getObjectKind

```
public String getObjectKind()
```

11.9.8 setObjectKind

```
protected void setObjectKind(String kind)
```

11.9.9 getTranslation

```
public float[0..*] getTranslation()
```

11.9.10 getRotation

```
public float[0..*] getRotation()
```

11.9.11 getScaled

```
public float[0..*] getScaled()
```

11.9.12 setTranslation

```
public void setTranslation(float translation[0..*])
```

11.9.13 setRotation

```
public void setRotation(float rotation[0..*])
```

11.9.14 setScaled

```
public void setScaled(float scaled[0..*])
```

11.9.15 toString

```
public String toString()
```

11.9.16 isLocked

```
public boolean isLocked()
```

11.9.17 setLocked

```
public void setLocked(boolean locked)
```

12 SVG to X3D converter

12.1 Class GetSvg

svg2x3d::GetSvg

```
public class GetSvg
```

Properties	
Alias	GetSvg
Visibility	public
Final	☐
Transient	☑
Abstract	☐
Leaf	☐

Attribute Summary	
Document	**document**
int	**scaling**
Element	**final3D**

Double	**wallHeight**
Double	**StrokeWidth**
Double	**floorPotition**

Constructor Summary

GetSvg()

Operation Summary

void	**setWallHeight**(Double wallHeight)
Element	**Get3D**(Document doc)
void	**addText**(String Color, Double rotation, Double transformX, Double transformY, Do
void	**addCurve**(String Color, Element Scene, int m, Double rotation, Double Mx, Double
void	**addCircle**(String Color, Double rotation, Double transformX, Double transformY,
void	**addRect**(String Color, Double transformX, Double transformY, Double x1, Double y
void	**addLine**(Double transformX, Double transformY, Double x1, Double x2, Double y1,
void	**addElement**(String element, Double transformX, Double transformY, Double cx, Dou
void	**setScaling**(int scaling)
Element	**createX3D**(Document svgdoc)

Attribute Detail

12.1.1 document

private Document **document**

12.1.2 scaling

public int **scaling**

12.1.3 final3D

public Element **final3D**

12.1.4 wallHeight

public Double **wallHeight**

12.1.5 StrokeWidth

public Double **StrokeWidth**

12.1.6 floorPotition

public Double **floorPotition**

Constructor Detail

12.1.7 GetSvg

public **GetSvg**()

Operation Detail

12.1.8 setWallHeight

public void **setWallHeight**(Double wallHeight)

12.1.9 Get3D

public Element **Get3D**(Document doc)

12.1.10 addText

private void
addText(String Color, Double rotation, Double transformX, Double transf
ormY, Double centerX, Double centerY, String fontf, String fonts, int m
, Element Scene, String text)

12.1.11 addCurve

private void
addCurve(String Color, Element Scene, int m, Double rotation, Double Mx
, Double My, Double C1x, Double C1y, Double C2x, Double C2y, Double C3x
, Double C3y, Double transformX, Double transformY)

12.1.12 addCircle

private void
addCircle(String Color, Double rotation, Double transformX, Double tran

sformY, Double centerX, Double centerY, Double radius, int m, Element S
cene)

12.1.13 addRect

public void
addRect(String Color, Double transformX, Double transformY, Double x1,
Double y1, Double rotation, Double width, Double height, int m, Element
Scene)

12.1.14 addLine

public void
addLine(Double transformX, Double transformY, Double x1, Double x2, Dou
ble y1, Double y2, Double rotation, int m, String Color, Element Scene)

12.1.15 addElement

public void
addElement(String element, Double transformX, Double transformY, Double
cx, Double cy, Double rotation, Double width, Double height, int m, El
ement Scene)

12.1.16 setScaling

public void **setScaling**(int scaling)

12.1.17 createX3D

private Element **createX3D**(Document svgdoc)

13 Πηγές

http://en.wikipedia.org/wiki/X3D
http://www.web3d.org/x3d/
http://www.w3.org/Graphics/SVG/
http://en.wikipedia.org/wiki/XML
http://java.com/en/