

ΑΝΩΤΑΤΟ ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΚΡΗΤΗΣ

Σχολή Τεχνολογικών Εφαρμογών  
Τμήμα Εφαρμοσμένης Πληροφορικής και Πολυμέσων



Πτυχιακή Εργασία

**«Editor For Interactive 3D Objects »**

*Οικονομάκος Παναγιώτης*

**Επιβλέπων καθηγητής :** Δρ Βιδάκης Νικόλαος

**Επιτροπή Αξιολόγησης :** Βιδάκης Ν. , Αποστολάκης Σ , Μηλολιδάκης Ι

**Ημερομηνία παρουσίασης:** 19-10-2010

Ηράκλειο Οκτώμβριος 2010

## Περίληψη

Σκοπός της πτυχιακής αυτής ήταν η μελέτη ,η σχεδίαση και η ανάπτυξη μιας εφαρμογής οπου με την οποία ο χρήστης θα μπορεί να αναπτύξει , να τροποποιήσει και να επεξεργαστεί τρισδιάστατα αντικείμενα(3D Objects) με την χρήση του Java3D API της ORACLE ,στο τέλος η εφαρμογή θα δίνει την δυνατότητα στο χρήστη να παράγει τον τελικό κώδικα του 3D αντικειμένου ώστε ο χρήστης να μπορεί να το χρησιμοποιήσει σε άλλες εφαρμογές , για τον λόγο αυτό μελετήθηκαν ο σχεδιασμός και οι λειτουργία παρόμοιων εφαρμογών ώστε να καθοριστούν οι ανάγκες σχεδίασης και υλοποίησης της εφαρμογής.

Πιο συγκεκριμένα για την ανάπτυξη της εφαρμογής χρησιμοποιήθηκε η βιβλιοθήκη **Swing** της Java η οποία παρέχει στο χρήστη ένα graphical User Interface(GUI) για Java προγράμματα.Ετσι η εφαρμογή υλοποιείται κυρίως από της βιβλιοθήκες της Java – Java3D και λόγω συγκεκριμένων απαιτήσεων της εφαρμογής χρησιμοποιήθηκε επιπρόσθετα το Freemarker API.

Έτσι η εφαρμογή που αναπτύχθηκε δίνει την δυνατότητα στο χρήστη να επιλέξει ένα από τα primitives αντικείμενα του Java 3D API με σκοπό να παραχθεί ο Java κώδικας.Ο χρήστης έχει την δυνατότητα μέσω διαφόρων εργαλείων να επεξεργαστεί και να τροποποιήσει το αντικείμενο σε real time , η εφαρμογή παρέχει στο χρήστη τη δυνατότητα να αλλάξει το μέγεθος , το σχήμα , το χρώμα , την όψη των αντικειμένων καθώς και την θέση τους μέσα στο χώρο .

Επίσης ο χρήστης μπορεί να εμπλουτίσει τα αντικείμενα αυτά με κίνηση , εικόνα άλλα και με πολλές διαδραστικές λειτουργίες . Τέλος δίνετε η δυνατότητα στο χρήστη να αναπτύξει project το οποίο αποθηκεύεται στο work space του χρήστη και μπορεί να υποστηρίξει αρχεία Java,XML,TXT,XSD το project αυτό μπορεί να ανακτηθεί οποιαδήποτε στιγμή για περαιτέρω επεξεργασία..

## Abstract

Goal of this dissertation was to study, design and the development of an application by which any user will be able to develop, configure, and edit 3 dimensions object by using Java 3D API of ORACLE, at the end the application provides to user the ability to produce the final code of the mentioned 3D object in order the user to be able to use this produced java code in any other applications, due to this the design and the behavior of similar applications have been studied in order the needs of the our application for the development and for the implementation to be defined.

Specifically for the application's development, **Swing** library have been used which provides a graphical user interface (GUI) for Java programs. Additional the application is implemented by Java - Java 3D libraries and due to some specific application's requirements Freemarker API have been used.

The application provides to user an option to choose between one of the primitives object of Java3D API in order Java Code of selected primitive object to be produced. User is able to edit and change some parts of the 3D object in real time by using some application's tools , user is able to change the dimension , color, appearance and object's position into the space.

Last but not least user is able to embed the object with motion, images and many other interactive actions. Final user is able to develop a project which is stored into his work space and it can support JAVA ,XML ,TXT,XSD files , this project can be loaded when ever is necessary by the application.

# Πίνακας Περιεχομένων

<b>1. ΕΙΣΑΓΩΓΗ .....</b>	<b>7</b>
1.1. ΑΝΤΙΚΕΙΜΕΝΟ ΤΗΣ ΠΤΥΧΙΑΚΗΣ.....	7
1.2. ΟΡΓΑΝΩΣΗ ΤΟΥ ΤΟΜΟΥ .....	7
<b>2. ΧΡΗΣΙΜΟΙ ΟΡΙΣΜΟΙ .....</b>	<b>8</b>
2.1. ΤΙ ΕΙΝΑΙ JAVA.....	8
2.2. ΤΙ ΕΙΝΑΙ JAVA3D.....	9
2.3. ΤΙ ΕΙΝΑΙ SWING .....	10
2.4. ΓΡΑΦΙΚΟ ΠΕΡΙΒΑΛΛΟΝ ΧΡΗΣΤΗ(GUI).....	11
2.5. ΤΙ ΕΙΝΑΙ ΣΧΕΔΙΑΣΤΙΚΑ ΠΡΟΤΥΠΑ – DESIGN PATTERNS.....	12
2.6. ΤΙ ΕΙΝΑΙ FREEMARKER TEMPLATE.....	13
<b>3. ΤΕΧΝΟΛΟΓΙΑ.....</b>	<b>14</b>
3.1. SWING.....	14
3.1.1. Συστατικά του Swing.....	15
3.1.2. Μοντέλο Ριζικού Υποδοχέα.....	16
3.1.3. Πλαίσιο JFrame.....	17
3.1.4. Πλαίσιο JInternalFrame.....	18
3.1.5. Κουμπιά JButton.....	20
3.1.6. Ράβδος ολίσθησης - JSlider .....	22
3.1.7. Επιλογέας χρωμάτων JColorChooser.....	23
3.1.8. Μενού - JMenu –JMenuBar – JMenuItem.....	24
3.1.9. Γεγονότα - Swing Events.....	25
3.1.10. Συμπέρασμα χρήσης Swing.....	27
3.2. JAVA3D .....	28
3.2.1. Γράφος Σκηνής.....	29
3.2.2. Ιεραρχία Κλάσεων Java3D API.....	30
3.2.3. Η Κλάση Simple Universe.....	32
3.2.4. Η Κλάση Canvas3D.....	33
3.2.5. Η κλάση TransformGroup .....	33
3.2.6. Συμπεριφορές – Behaviors.....	34
3.2.7. Φωτισμός – Lights .....	34
3.2.8. Texturing.....	35
3.2.9. Primitive Objects .....	37
3.3. FREEMARKER TEMPLATES .....	39
<b>4. ΑΝΑΛΥΣΗ ΛΟΓΙΣΜΙΚΟΥ.....</b>	<b>41</b>
4.1. ΤΙ ΕΙΝΑΙ ΑΝΑΛΥΣΗ ΛΟΓΙΣΜΙΚΟΥ .....	41
4.1.1. Ανάλυση και Καταγραφή Απαιτήσεων.....	41
4.1.2. Σενάρια Χρήσης .....	42
4.1.3. Ανάπτυξη Λογισμικού.....	42
4.1.4. Έλεγχος Λογισμικού.....	42
4.2. JEDITOR3D1.1 CASE STUDY .....	43
4.2.1. Περιγραφή Εφαρμογής JEditor3D1.1 .....	43
4.2.2. Γενικό Σενάριο Χρήσης του JEditor3D1.1.....	43
4.2.3. Καταγραφή απαιτήσεων JEditor3D1.1.....	45
4.2.4. Η Διεπαφή JEditor3D1.1 .....	46
4.2.5. Επιπρόσθετα παράθυρα και συστατικά εφαρμογής.....	75
4.2.6. Περιγραφή παραγόμενου κώδικα .....	78
<b>5. ΣΥΜΠΕΡΑΣΜΑΤΑ .....</b>	<b>83</b>
<b>6. ΒΙΒΛΙΟΓΡΑΦΙΑ.....</b>	<b>83</b>

## Πίνακας Tables

Table 1 : Design Patterns .....	12
Table 2 : Πακέτα βιβλιοθήκης Swing .....	15
Table 3 : Κατασκευαστές Πλαισίου.....	18
Table 4 : Κατασκευαστές JInternalFrame.....	20
Table 5 : Κατασκευαστές JButton.....	21
Table 6 :Κατασκευαστές ράβδου ολίσθησης .....	23
Table 7 : Γεγονότα Swing - AWT.....	26
Table 8 : Κατασκευαστές κλάσης SimpleUniverse .....	32
Table 9:Κατασκευαστές κλάσης Canvas3D.....	33
Table 10: Κατασκευαστές κλάσης TransformGroup .....	33

## Πίνακας Εικόνων

Figure 1: Γράφος σκηνης .....	9
Figure 2: Swing κλάσεις.....	10
Figure 3 : Freemarker input-output .....	13
Figure 4 : Μοντέλο ριζικού υποδοχέα.....	17
Figure 5 : Πλαίσιο JFrame .....	18
Figure 6 : Πλαίσιο JInternal Frame .....	19
Figure 7: Κουμπί JButton.....	21
Figure 8 : Ράβδος ολίσθησης .....	22
Figure 9 : Επιλογέας χρωμάτων .....	24
Figure 10: Swing Μενού .....	25
Figure 11 : Σύμβολα γράφου σκηνης.....	29
Figure 12 :Γράφος σκηνης αντικιμένων.....	30
Figure 13 : Ιεραρχία κλάσεων Java3D .....	31
Figure 14:Texture σε κύβο.....	36
Figure 15: Box – Sphere (Κουτί και Σφαίρα αντίστοιχα).....	38
Figure 16: Cone – Cylinder (Κώνος και Κύλινδρος αντίστοιχα).....	38
Figure 17: Αρχική κατάσταση JEditor3D 1.1 .....	47
Figure 18: Μενού επιλογής File.....	48
Figure 19: Μενού επιλογής Window .....	48
Figure 20:Μενού επιλογής About .....	48
Figure 21: Projects Frame structure .....	49
Figure 22: Design View for 3D object.....	50
Figure 23: Source View for 3D object .....	51
Figure 24: Source for Java file .....	51
Figure 25:Design Tools.....	52
Figure 26:Source Tools .....	52
Figure 27:Cordinates Axis and Rotation Axis .....	52
Figure 28: Γενικές Ιδιότητες (General Properties).....	55
Figure 29 : Transform Explorer Attribute .....	55
Figure 30: Shape’s position after translation.....	56
Figure 31:Translation Sliders .....	56
Figure 32:Μεταβολή άξονα προσανατολισμού - περιστροφής.....	57
Figure 33: Ράβδοι σημείου αναφοράς του αντικειμένου .....	57
Figure 34: Άξονας περιστροφής.....	58
Figure 35: Ράβδοι περιστροφής του άξονα και της γωνίας του αντικειμένου .....	58
Figure 36: Uniform Scale .....	59
Figure 37: Scale on X and Z Axis .....	59
Figure 38: Ράβδοι κλίμακας μετατροπής του αντικειμένου.....	59
Figure 39: Interpolator Attributes .....	60
Figure 40: Rotator Interpolator .....	61
Figure 41: Alpha Configuration .....	62
Figure 42: Mouse Interaction tool.....	63
Figure 43: Coloring Attributes .....	64

Figure 44: Επιλογέας χρωμάτων – Color Chooser.....	65
Figure 45: Αλλαγή χρωμάτων σε σχήμα και φόντο.....	65
Figure 46:Material Attributes Tool.....	67
Figure 47:Light Attributes.....	69
Figure 48:Ambient – Directional – Point Light.....	70
Figure 49: Texture Image.....	70
Figure 50: Boundary Tool.....	71
Figure 51: Φιλτράρισμα.....	72
Figure 52: Κύβος εμπλουτισμένος με εικόνα.....	72
Figure 53: Κώνος εμπλουτισμένος με εικόνα.....	72
Figure 54: Texture Image Tool.....	73
Figure 55:Texture Attributes Tool.....	74
Figure 56:Cube in Modulate option.....	74
Figure 57:Cube in Blend option when Blend color is black.....	75
Figure 58: Παράθυρο δημιουργίας νέου Project.....	75
Figure 59: Παράθυρο δημιουργίας νέου αρχείου.....	76
Figure 60:Editor Pane Tool.....	77

# 1. Εισαγωγή

Στο κεφάλαιο αυτό παρουσιάζουμε μια γενική περιγραφή του αντικειμένου της πτυχιακής αυτής και την οργάνωση των κεφαλαίων του υπόλοιπου τόμου.

## 1.1. Αντικείμενο της πτυχιακής

Στην συγκεκριμένη πτυχιακή ασχοληθήκαμε με την ανάπτυξη μια εφαρμογής η οποία θα μπορεί να υποστηρίζει την ανάπτυξη τρισδιάστατων αντικειμένων όπου κατόπιν επεξεργασία τους θα είναι διαθέσιμη η παραγωγή του κώδικα του αντικειμένου με τέτοια μορφή ώστε να μπορεί να χρησιμοποιηθεί και σε άλλες εφαρμογές.

Η εφαρμογή μέσω διαφόρων εργαλείων μπορεί να αλλάξει το σχήμα το μέγεθος το χρώμα και την θέση ενός αντικειμένου στο χώρο. Επίσης προσφέρει την δυνατότητα να εμπλουτιστούν τα αντικείμενα αυτά με εικόνα , κίνηση και άλλες διαδραστικές ικανότητες , όπως μετακίνηση του αντικειμένου μέσω του ποντικιού , περιστροφή του αντικειμένου μέσω του ποντικιού καθώς και zoom in zoom out.

Τέλος η εφαρμογή δίνει την δυνατότητα να δημιουργηθεί ένα Project το οποίο θα μπορεί να υποστηρίζει αρχεία Java , xml , txt , xsd για περαιτέρω χρήση. Η εφαρμογή μπορεί να ανακτήσει το αναφερόμενο Project οποιαδήποτε στιγμή και δίνει την δυνατότητα στο χρήστη να γράψει σχετικό κώδικα μέσω αυτής.

Τέλος η εφαρμογή δίνει την δυνατότητα στο χρήστη να μπορεί να παρεμβαίνει στο αντικείμενο σε real time με τα διάφορα εργαλεία που έχει στην διαθεσιμότητα του.

## 1.2. Οργάνωση του τόμου

Η συνέχεια της πτυχιακής έχει οργανωθεί στα ακόλουθα κεφάλαια:

- Στο Κεφάλαιο 2 παρουσιάζονται χρήσιμες έννοιες και ορισμοί, σχετικά με μία Java εφαρμογή.
- Στο Κεφάλαιο 3, παρουσιάζεται η τεχνολογία που χρησιμοποιήθηκε για την ανάπτυξη της εφαρμογής.
- Στο Κεφάλαιο 4, παρουσιάζονται πληροφορίες για την σχεδίαση , ανάλυση και υλοποίηση της εφαρμογής
- Στο Κεφάλαιο 5, παρουσιάζονται τα συμπεράσματα της έρευνας και της υλοποίησης
- Στο Κεφάλαιο 6, παρουσιάζεται όλη η σχετική βιβλιογραφία στην οποία βασιστήκαμε για τη μελέτη και εγγραφή της πτυχιακής.

## 2. Χρήσιμοι ορισμοί

Προτού αναλύσουμε την εφαρμογή που αναπτύχθηκε και τις τεχνολογίες που χρησιμοποιήθηκαν, αναφέρουμε κάποιους ορισμούς ώστε να γίνουν πιο κατανοητά τα επόμενα κεφάλαια.

### 2.1. Τι είναι Java

Java είναι μια αντικειμενοστραφής γλώσσα προγραμματισμού η οποία μέσω των βιβλιοθηκών της προσφέρει στον χρήστη την ικανότητα να αναπτύξει επιτυχώς web based και stand alone εφαρμογές , καθώς και την υλοποίηση προγραμμάτων που απαιτούν υψηλού επιπέδου γλώσσες προγραμματισμού.

- **Εικονική μηχανή(Virtual Machine)**

Τα προγράμματα σε Java μεταγλωττίζονται μέσω του εργαλείου javac το οποίο με τη σειρά του παράγει αρχεία τύπου .class(bytecode). Όταν εκτελείτε ένα πρόγραμμα Java το Java Virtual Machine του υπολογιστή θα αναλάβει να διαβάσει τα αρχεία .class και να τα μεταφράσει σε γλώσσα και εντολές μηχανής.

Επίσης αυτή η εικονική μηχανή προσφέρει μεγαλύτερη ασφάλεια στο σύστημα γιατί είναι υπεύθυνη για την επικοινωνία χρήστη υπολογιστή.Δεν μπορεί να γραφτεί κώδικας ο οποίος να μπορεί να βλάψει τον υπολογιστή αλλά και δεν μπορεί να εκτελεστή κώδικας ο οποίος αποτελείτε από επικίνδυνο περιεχόμενο.

- **Garbage Collector**

Η Java υποστηρίζει την ύπαρξη του συλλέκτη απορριμμάτων , η έννοια αυτή που χρησιμοποιείτε από τον τομέα της πληροφορικής δηλώνει την ελευθέρωση τμημάτων μνήμης από δεδομένα που δεν χρειάζονται ή δεν χρησιμοποιούνται άλλο, η απελευθέρωση μνήμης γίνεται αυτόματα. Η εικονική μηχανή μόλις καταλάβει ότι ο σωρός(heap) της μνήμης έχει σχεδόν γεμίσει ενεργοποιεί τον συλλέκτη απορριμμάτων.

Έτσι ο προγραμματιστής δεν ασχολείται με την ελευθέρωση τμημάτων μνήμης.

- **Επιδόσεις**

Η Java παρόλο που η εικονική μηχανή της προσφέρει τόσα πολλά πλεονεκτήματα είναι ποιο αργή σε σχέση με άλλες γλώσσες προγραμματισμού. Ωστόσο γίνονται προσπάθειες για βελτιώσεις και μάλιστα οι τελευταίες εκδόσεις του Javac με τη χρήση της τεχνολογίας Hot Spot έχουν ανεβάσει της επιδόσεις.



## 2.2. Τι είναι Java3D

Java 3D είναι ένα application programming interface (API) βασισμένο σε διάγραμμα σκηνής(scene graph) το οποίο εκτελείτε τόσο σε OpenGL όσο και σε Direct3D τεχνολογίες. Η βιβλιοθήκη της Java 3D βασίζεται στην Java με αποτέλεσμα να μπορούν να συνδυαστούν με άλλες εφαρμογές βασισμένες στο API της Java.

Η Java3d δίνει την δυνατότητα ανάπτυξης τρισδιάστατων γραφικών όπου ακόμα στο σύνολό τους μπορούν να συντελέσουν και ένα εικονικό τρισδιάστατο κόσμο με υψηλή απεικόνιση. Η Java 3D δεν περιλαμβάνει μόνο την απεικόνιση και φωτοαπόδοση τρισδιάστατων γραφικών αλλά ενσωματώνει προγραμματισμό γραφικών βασισμένο στην αντικειμενοστραφή αρχή.

Ο γράφος σκηνής εμφανίζεται σαν μια δενδρική δομή που περιλαμβάνει ξεχωριστά διακριτά στοιχεία τα οποία είναι απαραίτητα για την απόδοση των τρισδιάστατων αντικειμένων όπως φαίνεται και στο Figure1..

Επίσης εκτός από την απόδοση αυτών των τρισδιάστατων γεωμετρικών σχημάτων η Java 3D παρέχει εργαλεία τα οποία δίνουν στον προγραμματιστή την ικανότητα να ενσωματώσει στην εφαρμογή του ήχο.

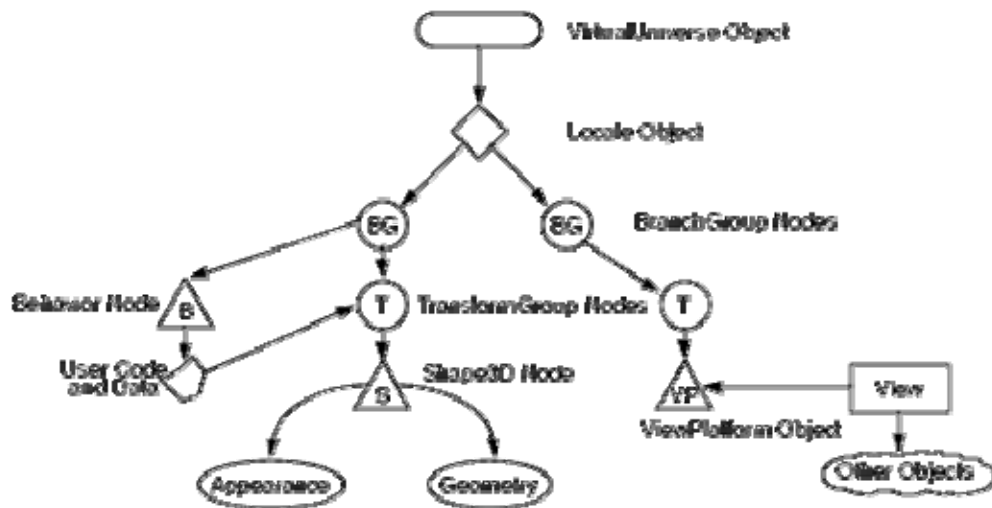


Figure 1: Γράφος σκηνής

- **Πλεονεκτήματα χρήσης**

Ορισμένα από τα πλεονεκτήματα που μπορεί να προσφέρει η χρήση της Java 3D αναφέρονται στα ίδια πλεονεκτήματα που προσφέρει η γενική χρήση της Java καθώς και ότι μπορεί να χρησιμοποιηθεί σε web based εφαρμογές βασισμένες στο API της Java και όχι μόνο. Δίνει την δυνατότητα στο προγραμματιστή να δημιουργήσει να τροποποιήσει και να επεξεργαστεί τρισδιάστατα αντικείμενα χρησιμοποιώντας πάντα το source code της εφαρμογής όπου αυτό δημιουργεί μια ευχέρεια στον προγραμματιστή καθώς ο κώδικας είναι πάντα ποιο αναγνώσιμος, συντηρίσιμος, επαναχρησιμοποιήσιμος και κυρίως εύκολος για να γραφτεί, επίσης κατόπιν συγκεκριμένων απαιτήσεων μπορεί να συνδυαστεί – συνεργαστεί με άλλες τεχνολογίες δίνοντας στον προγραμματιστή την ικανότητα δημιουργίας εφαρμογής υψηλών απαιτήσεων.

Επίσης η χρήση του γραφου σκηνής επιτρέπει στον προγραμματιστή να ορίσει να μετατρέψει και να επαναχρησιμοποιήσει όλα τα στοιχεία του τρισδιάστατου αντικειμένου με ταχύτητα και ευκολία.

### 2.3. Τι είναι Swing

Swing είναι μια βιβλιοθήκη της Java η οποία περιέχει πάρα πολλά συστατικά για την δημιουργία γραφικού περιβάλλοντος Graphical User Interface (GUI). Τέτοια συστατικά είναι κουμπιά ,μενού, γραμμές εργαλείων τμήματα παραθύρων , μπάρες κύλισης , ετικέτες και πολλά άλλα. Το Swing θα μπορούσε να οριστεί ως μια επέκταση της βιβλιοθήκης AWT - Abstract window toolkit και όχι ως αντικαταστάτη του.

Τα περισσότερα συστατικά του Swing είναι lightweight “ελαφρά” σε αντίθεση με αυτά του AWT που είναι heavyweight “βαριά” , αυτό συμβαίνει γιατί όλα τα συστατικά του Swing έχουν γραφτεί σε γλώσσα Java σε αντίθεση αυτών του AWT που έχουν γραφτεί με γηγενή κώδικα. Λόγο του παραπάνω τα συστατικά Swing έχουν την ικανότητα να μπορούν να αλλάζουν μορφή με δυναμικό τρόπο. Για παράδειγμα ο χρήστης μπορεί να τρέχει μια εφαρμογή σε windows και η εφαρμογή να έχει εμφάνιση όπως Motif της Sun.

Στο παρακάτω σχήμα παρουσιάζεται η βιβλιοθήκη Swing με μια σειρά από κλάσεις με τις οποίες ο χρήστης θα δουλέψει περισσότερο.

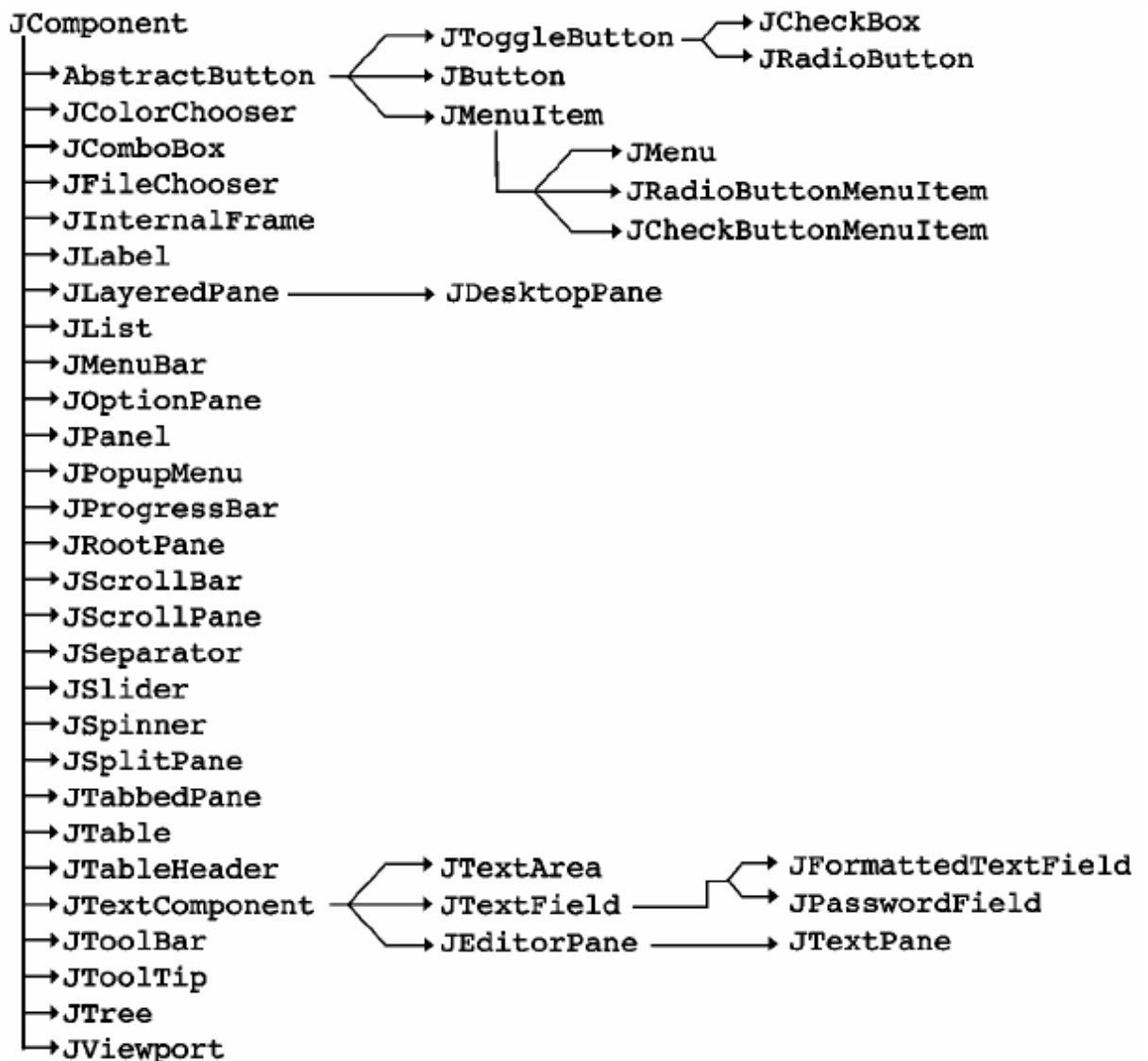


Figure 2: Swing κλάσεις

## 2.4. Γραφικό Περιβάλλον Χρήστη(GUI)

Γραφικό περιβάλλον Χρήστη – Graphical User Interface είναι ένα σύνολο γραφικών τα οποία εμφανίζονται στην οθόνη κάποιας συσκευής εξόδου και δίνουν την δυνατότητα στο χρήστη να επικοινωνεί – αλληλεπιδρά με τη συσκευή.

Τέτοια γραφικά μπορεί για παράδειγμα να είναι ένα κουμπί(button). Τα γραφικά αυτά παρέχουν κάποιες ενδείξεις στο χρήστη και δέχονται είσοδο(input) από τον χρήστη με την βοήθεια πάντα κάποιας συσκευής εισόδου όπως πληκτρολόγιο , ποντίκι, μικρόφωνο. Πλέον όλα τα σύγχρονα προγράμματα και λειτουργικά συστήματα παρέχουν στο χρήστη ένα γραφικό περιβάλλον καθότι είναι πιο εύχρηστο κατανοητό και γρηγορότερο από τον παλιό τρόπο της γραμμής εντολών.

Για την σωστή ανάπτυξη ενός GUI θα πρέπει να ακολουθηθούν κάποια βήματα ώστε το η επικοινωνία μεταξύ χρήστη υπολογιστή να είναι γρήγορη αμφίδρομη και διαδραστική ώστε να ικανοποιεί και τους πιο άπειρους χρήστες.

- Επικοινωνία με τον μελλοντικό χρήστη της εφαρμογής – προγράμματος ώστε να καταγραφούν οι απαιτήσεις του χρήστη σχετικά με το αντικείμενο της εφαρμογής αλλά και με τον τρόπο χρήσης που ίσως να ζητά ο συγκεκριμένος χρήστης.
- Έρευνα για παρόμοιες εφαρμογές προγράμματα με σκοπό την αποφυγή διαφόρων προβλημάτων αλλά και την βελτίωση κάποιων ήδη αναφερόμενων προσεγγίσεων
- Σχεδιασμός από τον προγραμματιστή και παρουσίαση στον χρήστη για πιθανές αλλαγές ή βελτιώσεις
- Σχεδιασμός ή βελτίωση των τελευταίων απαιτήσεων
- Ανάπτυξη εφαρμογής
- Σενάρια χρήσης για τεστ
- Παρουσίαση τελικού προϊόντος στον χρήστη

## 2.5. Τι είναι Σχεδιαστικά πρότυπα – Design Patterns

Στον προγραμματισμό ένα Design Pattern είναι μια επαναχρησιμοποιήσιμη λύση σε ένα σύνηθες εμφανιζόμενο πρόβλημα. Δηλαδή όταν ένα πρόβλημα εμφανίζεται συνέχεια τότε η γενική λύση σε αυτό το πρόβλημα μπορεί να χαρακτηριστεί ως Design Pattern. Ένα Design Pattern δεν είναι ένας ολοκληρωμένος σχεδιασμός ο οποίος υλοποιώντας τον μπορεί να μετατραπεί σε κώδικα. Είναι στην ουσία μια περιγραφή για το πως θα σχεδιαστή μια λύση ώστε να λυθεί το πρόβλημα, η λύση αυτή μπορεί να χρησιμοποιηθεί σε διάφορες περιπτώσεις ανεξάρτητα από την γλώσσα προγραμματισμού.

Το αντικειμενοστραφές σχεδιαστικό πρότυπο (object oriented design pattern) αναφέρεται σε συσχετίσεις και αλληλεπιδράσεις μεταξύ κλάσεων ή αντικειμένων χωρίς όμως να προσδιορίζει συγκεκριμένα ποιες κλάσεις ή ποια αντικείμενα εμπλέκονται. Γενικά για την σωστή υλοποίηση μιας εφαρμογής υψηλών και μη, απαιτήσεων η χρησιμοποίηση και ο συνδυασμός διαφόρων Design Patterns είναι απαραίτητος με σκοπό να αποφευχθούν διάφορα προβλήματα και να γίνει η εφαρμογή πιο γρήγορη, πιο εύελκτη και πιο συντηρήσιμη. Αρκετές φορές μπορεί να χρησιμοποιηθούν διαφορετικά Design Patterns για την λήψη του ίδιου προβλήματος. Συνήθως κάποια Patterns παρουσιάζουν ομοιότητες κάποια άλλα οδηγούν σε κάποιο άλλο σχεδιαστικό πρότυπο.

Τα συνήθως γνωστά και πιο κύρια Design Patterns μέχρι στιγμής είναι 23 στον αριθμό ενώ υπάρχουν και αρκετά άλλα τα οποία είναι λιγότερα σημαντικά.

Η εφαρμογή μας JEditor3D1.1 υλοποιεί δυο από τα βασικά Design patterns τα MVC (Model View Controller) και το Singleton, το πρώτο αναφέρεται σε στρατηγικές σχεδιαστικές τακτικές με σκοπό τον συνδυασμό γραφικού περιβάλλον GUI και Back End Programming, αντίστοιχα το δεύτερο αναφέρεται κυρίως στη σωστή διαχείριση των αντικειμένων της Java. Παρακάτω αναφέρονται ονομαστικά κάποια από τα πιο γνωστά και ευρέως χρησιμοποιήσιμα.

Creational Patterns	Structural Patterns	Behavioral	J2SEE Patterns	Misc
Abstract Factory	Adapter	Chain of Responsibility	MVC	Typesafe enum
Builder	Bridge	Command	Business Delegate	Restful WS
Factory Method	Composite	Interpreter	Composite Entity	
Prototype	Decorator	Mediator	Data Access Object	
Singleton	Facade	Memento	Font Controller	
	Flyweight	Observer	Intercepting Filter	
	Proxy	State	Service Locator	
		Strategy	Transfer Object	
		Template Method		
		Visitor		

Table 1 : Design Patterns

## 2.6. Τι είναι FreeMarker Template

Freemarker είναι μια μηχανή προτύπων η οποία μπορεί να παράγει ένα κείμενο σαν έξοδο. Δηλαδή είναι ένα γενικό εργαλείο το οποίο μπορεί και παράγει από HTML κείμενο μέχρι και αυτοδύναμο κώδικα βασισμένο πάντα σε πρότυπα (templates). Αυτή η μηχανή προτύπων είναι βασισμένη στην Java και εστιάζει κυρίως στην αρχιτεκτονική του MVC – Model View Controller.

Το FreeMarker είναι σχεδιασμένο για να χρησιμοποιείται κυρίως σε Web Based εφαρμογές για την παραγωγή HTML σελίδων.

Στο παρακάτω σχήμα ( Figure3) απεικονίζεται ένα σχεδιάγραμμα για την συσχέτιση Java Objects – Template αρχείων καθώς και το τελικό αποτέλεσμα.

- **Χρήση του FreeMarker**

Ένα απλό παράδειγμα χρήσης του freemarker template είναι όταν σε μια Web based εφαρμογή που χρησιμοποιεί σελίδες HTML πρέπει να χωριστεί το source code (Java programs) από το κομμάτι του σχεδιασμού της σελίδας ώστε η εφαρμογή να είναι λιγότερη περίπλοκη και περισσότερο συντηρήσιμη καθώς οι οντότητες – μέρη που την αποτελούν χωρίζονται και ομαδοποιούνται. Έτσι με αυτόν το τρόπο οι σχεδιαστές της σελίδας μπορούν να κάνουν αλλαγές χωρίς να χρειάζονται αλλαγές και στο source java code της εφαρμογής η να χρειάζεται μεταγλώττιση (re-compile) ο Java κώδικας με σκοπό να καταγραφούν οι αλλαγές.

- **Κύρια Στοιχεία**

Το κύριο χαρακτηριστικό της μηχανής Freemarker είναι ότι τα Java προγράμματα ετοιμάζουν τα δεδομένα που πρόκειται να εμφανιστούν και η μηχανή Freemarker κατασκευάζει το κείμενο το οποίο εμπεριέχει τα δεδομένα προς εμφάνιση.

Παρόλο που συνήθως συναντάτε σε Web Based εφαρμογές χρησιμοποιείται εξίσου συχνά και σε stand alone εφαρμογές.



Figure 3 : Freemarker input-output

## 3. Τεχνολογία

### 3.1. Swing

Όπως αναφέραμε και στο παραπάνω κεφάλαιο η εφαρμογή μας σχεδιάστηκε και υλοποιήθηκε πάνω στην τεχνολογία της Java και ειδικότερα χρησιμοποιήθηκε το πακέτο βιβλιοθηκών του Swing με σκοπό να αναπτυχθεί καταλλήλως το GUI.

Το πακέτο βιβλιοθήκης του Swing είναι ευρέως γνωστό και χρησιμοποιήσιμο τόσο σε Web Based εφαρμογές όσο και σε stand alone εφαρμογές όπως ο Editor (JE3D1.1). Το Swing δεν είναι ο αντικαταστάτης του AWT - Abstract Window Toolkit αλλά μια πετυχημένη βελτίωση και επέκταση.

Το Swing αποτελείται από πάρα πολλά συστατικά τα οποία είναι υπεύθυνα για την δημιουργία γραφικού περιβάλλοντος , τέτοια συστατικά μπορεί να είναι : κουμπιά , πίνακες , ετικέτες , παράθυρα , μπάρες κύλισης , μενού , γραμμή εργαλείων , πεδία για την υποδοχή κειμένου και πολλά άλλα. Το Swing εκτός από τα συστατικά για την ανάπτυξη του γραφικού περιβάλλοντος αποτελείται και από άλλα πακέτα τα οποία δίνουν την δυνατότητα στον προγραμματιστή να συνδυάσει το γραφικό περιβάλλον με διαδραστικές λειτουργίες μεταξύ χρήστη υπολογιστή. Ποιο συγκεκριμένα με την χρησιμοποίηση κάποιων πακέτων δίνεται η δυνατότητα στα γραφικά συστατικά να παράγουν κάποιες ενέργειες όπου τις ενέργειες αυτές της διαχειρίζονται άλλα πακέτα με σκοπό την καταγραφή του συμβάντος και την επεξεργασία του. Οπότε το Swing παρέχει μια μεγάλη ποικιλία από πακέτων τα οποία μπορούν να ολοκληρώσουν τις απαιτήσεις μια εφαρμογής τόσο Web based όσο και stand alone.

Στον παρακάτω πίνακα αναφέρονται κάποια από τα πακέτα τα οποία χρησιμοποιούνται για την ολοκλήρωση μια Java εφαρμογής.

Πακέτο	Περιγραφή
javax.swing	Είναι το αρχικό πακέτο το οποίο περιέχει όλες της κλάσεις για την ανάπτυξη και επικοινωνία των GUI συστατικών
javax.swing.text	Περιέχει κλάσεις και συστατικά για την υποδοχή και επεξεργασία κειμένου.
javax.swing.filechooser	Περιέχει κλάσεις και συστατικά για την εμφάνιση και διαχείριση του συστατικού διαλόγου για την επιλογή αρχείων και φακέλων.
javax.swing.colorchooser	Περιέχει κλάσεις και συστατικά για την εμφάνιση και διαχείριση του συστατικού διαλόγου για την επιλογή χρωμάτων.
javax.swing.table	Περιέχει κλάσεις και λειτουργίες για την εμφάνιση και διαχείριση δεδομένων με την μορφή πίνακα
javax.swing.undo	Περιέχει κλάσεις για την διαχείριση της λειτουργίας αναίρεσης και επανάληψης λειτουργιών.
javax.swing.tree	Περιέχει κλάσεις για την δημιουργία και διαχείριση του συστατικού δενδρικού νήματος.

javax.swing.border	Περιέχει κλάσεις και λειτουργίες για την δημιουργία πλαισίων γύρο από τα συστατικά του Swing.
javax.swing.event	Περιέχει κλάσεις και λειτουργίες οι οποίες ορίζουν τα γεγονότα – συμβάντα και τους ακροατές σχετικά με τα γραφικά συστατικά του Swing.
javax.swing.plaf	Περιέχει κλάσεις και λειτουργίες που σχετίζονται με την μορφή ενός GUI.Ο προγραμματιστής μπορεί να αλλάξει το look and feel ενός γραφικού συστατικού.

**Table 2 :** Πακέτα βιβλιοθήκης Swing

### 3.1.1. Συστατικά του Swing

Παρακάτω αναφέρονται τα κύρια συστατικά της βιβλιοθήκης Swing καθώς και όλες οι υποκλασεις της κλάσης JComponent.

- **Abstract Action**  
Είναι η ανώτερη κλάση κουμπιών. Υποκλάσεις της κλάσης Abstract Button είναι οι JButton JMenuItem και JToggleButton. Η πρώτη αποτελεί τα κλασικά και βελτιωμένα κουμπιά που εμφανίζονται σε κάθε Java εφαρμογή , η δεύτερη αποτελεί το μενού και όλα τα συστατικά που συσχετίζονται με το μενού και η τρίτη τα ραδιοπλήκτρα.
- **JcomboBox**  
Είναι ένα συστατικό που συνδυάζει κείμενο και πτυσσόμενη λίστα.
- **JInternalFrame**  
Είναι πλαίσιο (window) το οποίο μπορεί να τοποθετηθεί σε άλλον υποδοχέα.
- **JLabel**  
Είναι ετικέτα που μπορεί να περιέχει κείμενο εικόνα ή ακόμα και τα δύο.
- **JlayerPane**  
Είναι ένα πάνελ το οποίο περιέχει πολλά επίπεδα.
- **Jlist**  
Είναι μια λίστα που μπορεί να χρησιμοποιηθεί σε πολλές περιπτώσεις.
- **JMenuBar**  
Είναι μια γραμμή μενού που προσθέτετε σε υποδοχέα.
- **JPanel**  
Είναι ένα συστατικό στο οποίο συνήθως τοποθετούνται κουμπιά λίστες , πίνακες και πολλά άλλα διαδραστικά συστατικά.

- **JPopupMenu**  
Είναι ένα αναδύομενο συστατικό το οποίο μπορεί να υποδεχθεί κείμενο και γραφικά.
- **JProgressBar**  
Είναι μια ράβδος προόδου που απεικονίζει την πρόοδο μιας διεργασίας σε ποσοστό % του συνολικού μήκους της.
- **JScrollBar**  
Είναι μια ράβδος κύλισης
- **JSlider**  
Είναι ένα συστατικό ολίσθησης
- **JTable**  
Είναι ένας πίνακας ο οποίος μπορεί να φιλοξενήσει στα κελιά του πολλά άλλα συστατικά του Swing όπως κείμενο εικόνα και άλλα.
- **JToolBar**  
Είναι μια εργαλειοθήκη η οποία μεταφέρεται με το ποντίκι
- **JTree**  
Γραφικό συστατικό για την παρουσίαση πληροφορίας σε μορφή δένδρου.

### 3.1.2. Μοντέλο Ριζικού Υποδοχέα

Το μοντέλο του ριζικού υποδοχέα χρησιμοποιείται από τις κλάσεις JFrame , JDialog , JWindow , JApplet και JInternalFrames και ορίζει ποια θα είναι η ιεραρχία των συστατικών που θα τοποθετηθούν σε έναν από τους παραπάνω υποδοχείς. Δεν είναι δυνατόν να τοποθετήσουμε ένα κουμπί κατευθείαν σε έναν υποδοχέα JFrame. Το μοντέλο του ριζικού υποδοχέα περιλαμβάνει κάποιες αρχές για το πως και που μπορούμε να τοποθετήσουμε γραφικά συστατικά του Swing. Γενικά υπάρχει μια ιεραρχία συστατικών τα οποία το ένα υποδέχεται το άλλο με σκοπό να ολοκληρωθεί το γραφικό περιβάλλον. Εάν αυτή η ιεραρχία δεν ακολουθηθεί τότε κατά την μεταγλώττιση θα εμφανίζονται μηνύματα λάθους.

Παρακάτω αναφέρουμε μερικές από τις αρχές που διέπουν το μοντέλο του ριζικού υποδοχέα.

- Κάθε εφαρμογή που χρησιμοποιεί τα πακέτα βιβλιοθηκών του Swing πρέπει να διαθέτει έναν ριζικό υποδοχέα , αν έχουμε μια stand alone εφαρμογή τότε σαν ριζικός υποδοχέας ορίζεται ένα JFrame. Αν η εφαρμογή μας είναι βασισμένη στο Web τότε σαν ριζικός υποδοχέας ορίζεται ένα JApplet , μέσα σε κάθε έναν ριζικό υποδοχέα ορίζεται μια ιεραρχία υποδοχής των γραφικών συστατικών του Swing.
- Κάθε ριζικός υποδοχέας απαρτίζεται από επιμέρους χώρους. Ο κύριος χώρος ενός ριζικού υποδοχέα ονομάζεται Root Pane και σαν κλάση αναφέρεται ως JRootPane.
- Πάνω στον ριζικό χώρο Root Pane ορίζεται ένας άλλος χώρος που ονομάζεται Layerd Pane και σαν κλάση αναφέρεται ως JLayerdPane. Σε αυτόν τον χώρο μπορούν να



τοποθετηθούν αντικείμενα σε διάφορα επίπεδα

- Πάνω στο LayerdPane υπάρχει ο χώρος περιεχομένων Content Pane ο οποίος είναι ο κύριος χώρος τοποθέτησης συστατικών.Εδώ σε αυτόν το χώρο είναι δυνατό να τοποθετηθούν JPanel's όπου με την σειρά τους μπορούν να φιλοξενήσουν κουμπιά μπάρες κύλισης πίνακες , ετικέτες και άλλα γραφικά συστατικά του Swing.
- Επίσης πάνω στο LayerdPane εκτός από τον χώρο περιεχομένων μπορεί και να προστεθεί μια γραμμή μενού η οποία θα τοποθετηθεί στο πάνω μέρος του διαστρωματομένου χώρου και στο κάτω ο χώρος περιεχομένων.
- Στο τέλος υπάρχει ένα επίπεδο που ονομάζεται διαφανής χώρος και Glass Pane και είναι υπεύθυνος για τον χειρισμό συμβάντων που σχετίζονται με τα αντικείμενα που απαρτίζεται.

Στο παρακάτω σχήμα διακρίνεται η ιεραρχία των υποδοχών.

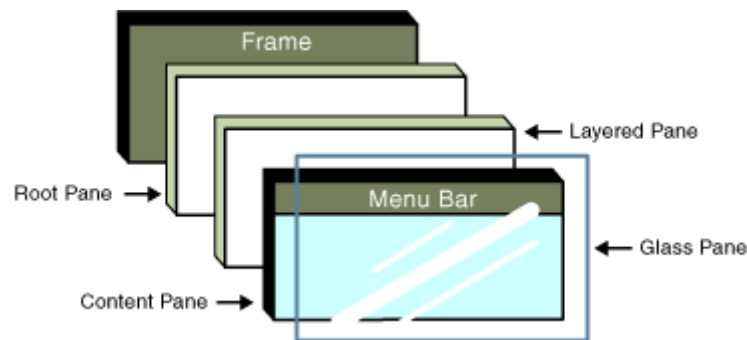


Figure 4 : Μοντέλο ριζικού υποδοχέα

### 3.1.3. Πλαίσιο JFrame

Το πλαίσιο JFrame είναι το βασικό συστατικό και ο αρχικός υποδοχέας σε μία γραφική παραθυριακή εφαρμογή.Στο συστατικό JFrame μπορούμε να τοποθετήσουμε πολλαπλά συστατικά του γραφικού περιβάλλοντος που διέπει την βιβλιοθήκη Swing.Για να γίνει η τοποθέτηση αυτή θα πρέπει να καλέσουμε τον χώρο περιεχομένων Content Pane.Στον χώρο αυτόν εκτελούνται και άλλες διεργασίες όπως καθορισμός του φόντου και διάταξη περιεχομένων.Το JFrame προσφέρει ουσιαστικά ότι προσφέρει και ένα παράθυρο , κουμπιά ελαχιστοποίησης , μεγιστοποίησης κλείσιμο ,μετακίνηση και άλλα πολλά.

Ένα JFrame σαν ριζικός υποδοχέας μπορεί να φιλοξενήσει και άλλος ριζικούς υποδοχείς όπως για παράδειγμα ένα συστατικό διαλόγου (JDialog).

Στο παρακάτω σχήμα παρουσιάζεται η δημιουργία ενός απλού παραθύρου JFrame.

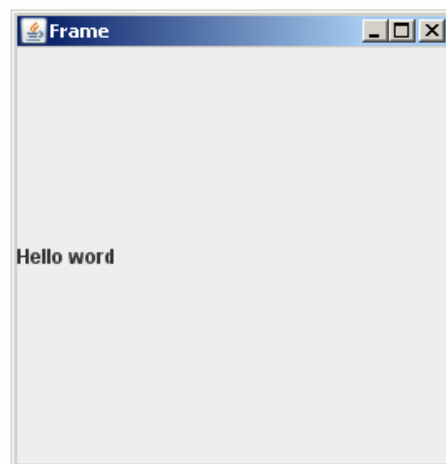
```

import java.awt.Dimension;
import javax.swing.JFrame;
import javax.swing.JLabel;

public class HelloWorld {

    public static void main(String [] args) {
        JFrame frame = new JFrame("Frame");
        JLabel label = new JLabel("Hello word");
        frame.getContentPane().add(label);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(new Dimension(200, 200));
        frame.setVisible(true);
    }
}

```



**Figure 5 :** Πλαίσιο JFrame

Παρατηρώντας τον κώδικα βλέπουμε ότι για την δημιουργία ενός απλού παραθύρου χρειάζεται μόνο να δημιουργήσουμε ένα αντικείμενο της κλάση Frame εκτελώντας την παρακάτω εντολή `JFrame = new JFrame("Title")` , στη συνέχεια δημιουργούμε μια ετικέτα και της περνάμε ένα κείμενο προς εμφάνιση τοποθετώντας την στο Content Pane. Τέλος δίνουμε διαστάσεις στο παράθυρο και του δίνουμε την εντολή να εμφανιστεί.

- **Κατασκευαστές(Constructor)**

Constructor	Description
<code>JFrame()</code>	Constructs a new frame that is initially invisible.
<code>JFrame(GraphicsConfiguration gc)</code>	Creates a Frame in the specified GraphicsConfiguration of a screen device and a blank title.
<code>JFrame(String title)</code>	Creates a new, initially invisible Frame with the specified title.
<code>JFrame(String title, GraphicsConfiguration gc)</code>	Creates a JFrame with the specified title and the specified GraphicsConfiguration of a screen device.

**Table 3 :** Κατασκευαστές Πλαισίου

### 3.1.4. Πλαίσιο `JInternalFrame`

Το πλαίσιο `JInternalFrame` έχει την όψη ενός κανονικού πλαισίου `JFrame` με την ικανότητα όμως ότι μπορεί να τοποθετηθεί μέσα σε ένα άλλο πλαίσιο με την χρήση του `JDesktopPane` ο οποίος είναι ένας υποδοχέας που μπορεί να φιλοξενήσει τα πλαίσια αυτά. Οι διαφορές ενός `JInternalFrame` με ένα κανονικό πλαίσιο `Frame` είναι ότι το `JInternalFrame` δεν παράγει γεγονότα παραθύρου (Window Events) όπως κάνει το `JFrame` και ότι χρειάζεται ένα υποδοχέα για να εμφανιστεί.

- **Χαρακτηριστικά**

Κύρια χαρακτηριστικά ενός `JInternalFrame` είναι ότι μπορεί να κινείται μέσα σε ένα `JDesktopPane` και να υποστηρίζει τις παρακάτω λειτουργίες `resizing` , `iconifying` , `closing` , `maximizing`.

Για να χρησιμοποιήσεις σωστά τα `JInternalFrames` πλαίσια πρέπει να ορίσεις το μέγεθος του πλαισίου και να ορίσεις το σημείο τοποθέτησης του πλαισίου. Επίσης για να τοποθετήσεις συστατικά στο `JInternalFrame` θα πρέπει να καλέσεις τον χώρο περιεχομένων (`Content Pane`) του `JInternalFrame`. Τέλος θα πρέπει να τοποθετήσεις το πλαίσιο μέσα σε έναν υποδοχέα όπως το `JDesktopPane` που προαναφέραμε.

Στο παρακάτω σχήμα παρουσιάζεται η δημιουργία ενός απλού `JInternalFrame`.

```
import java.awt.Dimension;
import javax.swing.JDesktopPane;
import javax.swing.JFrame;
import javax.swing.JInternalFrame;
import javax.swing.JLabel;

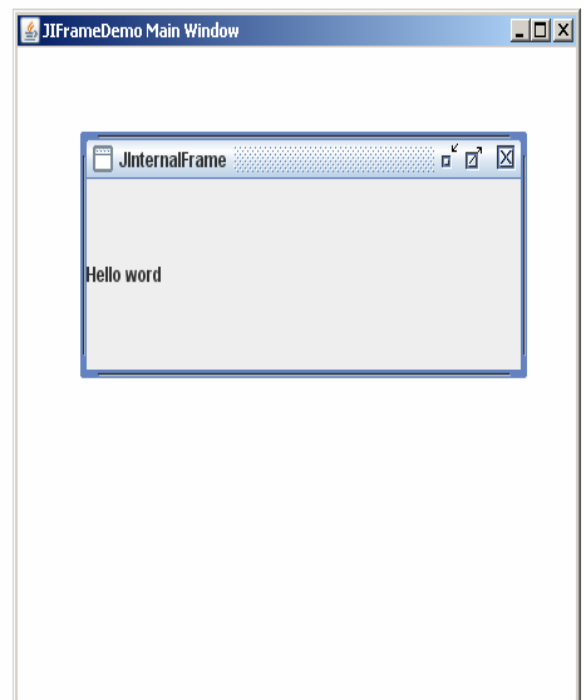
public class HelloWorld {

    public static void main(String[] a) {
        final JFrame frame = new JFrame("JIFrameDemo Main Window");
        frame.setSize(new Dimension(400, 400));

        JDesktopPane dtp = new JDesktopPane();
        frame.setContentPane(dtp);

        JInternalFrame interFrame = new JInternalFrame("JInternalFrame", true,true, true, true);
        JLabel reader = new JLabel("Mail Reader Would Be Here");
        interFrame.setContentPane(reader);
        interFrame.setSize(400, 300);
        interFrame.setLocation(50, 50);
        interFrame.setVisible(true);
        dtp.add(interFrame);

        frame.setVisible(true);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}
```



**Figure 6 :** Πλαίσιο `JInternal Frame`

Όπως προκύπτει και από το σχήμα στην αρχή δημιουργείται ένα απλό πλαίσιο τύπου `JFrame` στη συνέχεια δημιουργείται ένας υποδοχέας τύπου `JDesktopPane` για να υποδεχτεί το `JInternal Frame` πλαίσιο στη συνέχεια τοποθετείται ο υποδοχέας αυτός πάνω στον υποδοχέα του `JFrame` και τέλος αφού έχουμε δημιουργήσει και αρχικοποιήσει το `JInternalFrame` αντικείμενο το περνάμε στο `JDesktop Pane` ώστε να παράγουμε το επιθυμητό αποτέλεσμα. Το πλαίσιο `JInternal Frame` το δημιουργήσαμε δημιουργώντας ένα αντικείμενο της κλάσης `JInternalFrame` εκτελώντας τη παρακάτω εντολή.

```
JInternalFrame internalFrame = new JInternalFrame();
```

- **Κατασκευαστές (Constructors)**

<b>Constructor</b>	<b>Description</b>
JInternalFrame()	Creates a non-resizable, non-closable, non-maximizable, non-iconifiable JInternalFrame with no title.
JInternalFrame(String title)	Creates a non-resizable, non-closable, non-maximizable, non-iconifiable JInternalFrame with the specified title.
JInternalFrame(String title, boolean resizable)	Creates a non-closable, non-maximizable, non-iconifiable JInternalFrame with the specified title and resizability.
JInternalFrame(String title, boolean resizable, boolean closable)	Creates a non-maximizable, non-iconifiable JInternalFrame with the specified title, resizability, and closability.
JInternalFrame(String title, boolean resizable, boolean closable, boolean maximizable)	Creates a non-iconifiable JInternalFrame with the specified title, resizability, closability, and maximizability.
JInternalFrame(String title, boolean resizable, boolean closable, boolean maximizable, boolean iconifiable)	Creates a JInternalFrame with the specified title, resizability, closability, maximizability, and iconifiability.

**Table 4 :** Κατασκευαστές JInternalFrame

### 3.1.5. Κουμπιά JButton

Τα κουμπιά στο Swing προέρχονται από την κλάση JButton η οποία είναι υποκλάση της κλάσης AbstractButton. Τα κουμπιά JButton στο Swing μπορούν να υποδεχτούν κείμενο εικόνα , κείμενο και εικόνα μαζί καθώς μπορεί ο προγραμματιστής να μεταβάλει το μέγεθος το χρώμα καθώς και την όψη τους. Επίσης κάθε κουμπί στο Swing παράγει κάποια Action Events (Γεγονότα) ύστερα από κάθε εντολή του χρήστη : mouse pressed , mouse released και άλλα. Για τα γεγονότα αυτά έχει οριστεί ένας ακροατής όπου είναι μια διεργασία υπεύθυνη για το τι πρέπει να συμβεί μόλις παραχθεί ένα γεγονός. Για να επιτευχθεί αυτό χρειάζεται ο ορισμός ενός ActionListener και με την βοήθεια της μεθόδου addActionListener περνάμε τον ακροατή στο κουμπί.

Στο παρακάτω σχήμα παρουσιάζεται η δημιουργία ενός απλού JButton.

```

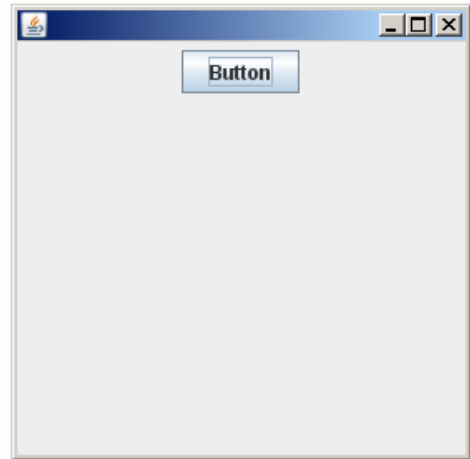
import java.awt.Dimension;
import java.awt.HeadlessException;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JPanel;

public class SwingButton extends JFrame {

    public SwingButton() throws HeadlessException {
        super();
        JPanel panel = new JPanel();
        JButton button = new JButton("Button");
        panel.add(button);
        this.getContentPane().add(panel);
        this.setVisible(true);
        this.setDefaultCloseOperation(EXIT_ON_CLOSE);
        this.setSize(new Dimension(300, 300));
    }

    public static void main(String []args) {
        SwingButton sw = new SwingButton();
    }
}

```



**Figure 7:** Κουμπί JButton

Σύμφωνα με τον παραπάνω κώδικα για να κατασκευαστή ένα κουμπί στο Swing πρέπει να δημιουργήσουμε ένα αντικείμενο της κλάσης JButton , αυτό γίνεται εκτελώντας την παρακάτω εντολή:

```
JButton button = new JButton("Title")
```

Στη συνέχεια περνάμε το κουμπί σε ένα JPanel το οποίο στη συνέχεια το περνάμε στο JFrame. Σε κάθε κουμπί της κλάσης JButton είναι δυνατό να του ορίσουμε και έναν ακροατή γεγονόςτος κάτι που θα μελετήσουμε παρακάτω.

- **Κατασκευαστές (Constructors)**

<b>Constructors</b>	<b>Description</b>
JButton()	Creates a button with no set text or icon.
JButton(Action a)	Creates a button where properties are taken from the Action supplied
JButton(Icon icon)	Creates a button with an icon.
JButton(String text)	Creates a button with text.
JButton(String text, Icon icon)	Creates a button with initial text and an icon.

**Table 5 :** Κατασκευαστές JButton

### 3.1.6. Ράβδος ολίσθησης - JSlider

Στη βιβλιοθήκη Swing μας δίνεται η δυνατότητα να εμπλουτίσουμε το γραφικό μας περιβάλλον με ράβδους ολίσθησης. Σε μια ράβδο ολίσθησης μπορούμε να ορίσουμε τον προσανατολισμό αν θα είναι οριζόντιος ή κάθετος καθώς και μέγιστη τιμή αρχική τιμή και ελάχιστη τιμή. Με την ράβδο ολίσθησης μπορούμε να μεταβούμε από την ελάχιστη στην μέγιστη τιμή χρησιμοποιώντας την λαβή που σου παρέχει το συστατικό αυτό. Μια ράβδο ολίσθησης παράγει γεγονότα αλλαγής κατάστασης ChangeEvent γιαυτο και ο ακροατής με τον οποίο την συνδέουμε ονομάζεται ChangeListener.

Για να δημιουργήσουμε μια ράβδο ολίσθησης θα πρέπει να δημιουργήσουμε ένα αντικείμενο της κλάσης JSlider εκτελώντας την παρακάτω εντολή:

```
JSlider slider = new JSlider();
```

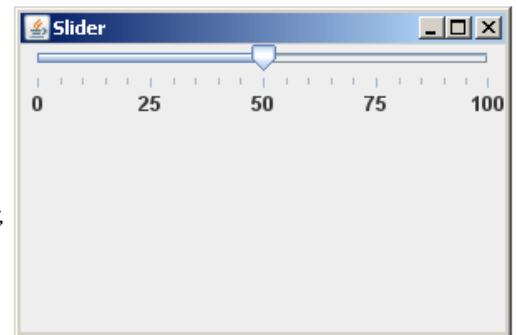
Στο παρακάτω σχήμα παρουσιάζεται μια ράβδος ολίσθησης με ελάχιστη τιμή 0 και μέγιστη τιμή 100.

```
import java.awt.BorderLayout;
import javax.swing.JFrame;
import javax.swing.JSlider;

public class SwingSlider {
    public static void main(String args[]) {
        JFrame frame = new JFrame("Slider");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        JSlider slider = new JSlider();
        slider.setMinorTickSpacing(5);
        slider.setMajorTickSpacing(25);
        slider.setPaintTicks(true);
        slider.setSnapToTicks(true);
        slider.setPaintLabels(true);
        slider.setMaximum(100);
        slider.setMinimum(0);

        frame.add(slider, BorderLayout.NORTH);
        frame.setSize(300, 200);
        frame.setVisible(true);
    }
}
```



**Figure 8 :** Ράβδος ολίσθησης

Όπως φαίνεται και στο παραπάνω σχήμα για να δημιουργήσουμε μία ράβδο ολίσθησης δημιουργήσαμε ένα αντικείμενο της κλάσης JSlider εκτελώντας την εντολή : JSlider slider = new JSlider() και στην συνέχεια του αρχικοποιήσαμε κάποιες τιμές με σκοπό να ορίσουμε μέγιστη και ελάχιστη τιμή καθώς και να προβάλουμε σκίαση και να δημιουργήσουμε μικρές και μεγάλες υποδιαιρέσεις της ράβδου.

- **Κατασκευαστές (Constructors)**

<b>Constructors</b>	<b>Description</b>
JSlider()	Creates a horizontal slider with the range 0 to 100 and an initial value of 50.
JSlider(BoundedRangeModel brm)	Creates a horizontal slider using the specified BoundedRangeModel.
JSlider(int orientation)	Creates a slider using the specified orientation with the range 0 to 100 and an initial value of 50.
JSlider(int min, int max)	Creates a horizontal slider using the specified min and max with an initial value equal to the average of the min plus max.
JSlider(int orientation, int min, int max, int value)	Creates a slider with the specified orientation and the specified minimum, maximum, and initial values.
JSlider(int min, int max, int value)	Creates a horizontal slider using the specified min, max and value.

Table 6 :Κατασκευαστές ράβδου ολίσθησης

### 3.1.7. Επιλογέας χρωμάτων JColorChooser

Το Swing αποκλειστικά παρέχει ένα συστατικό για την επιλογή χρωμάτων συστατικό αυτό καλείται σαν modal και μόλις ο χρήστης επιλέξει χρώμα με έναν απο τους τρεις τρόπους ο επιλογέας χρωμάτων σου επιστρέφει το χρώμα το οποίο διάλεξες σαν αντικείμενο της κλάσης Color. Οι τρόποι για την επιλογή χρώματος είναι τρεις και είναι οι εξής:

- Στον πρώτο τρόπο εμφανίζονται μικρά τετραγωνάκια με χρώματα στα οποίο ο χρήστης καλείτε να επιλέξει ένα χρώμα της αρέσκειας του.
- Ο δεύτερος τρόπος είναι ο τρόπος HSB (Hue/Saturation/Bightness)
- Ο τρίτος τρόπος είναι ο τρόπος RGB , δηλαδή ο χρήστης διαλέγει ένα χρώμα αναμιγνύοντας τις τιμές των χρωμάτων κόκκινου ,πράσινου και μπλε οι οποίες κυμαίνονται από 0 μέχρι 256.

Το να καλέσουμε προγραμματιστικά έναν επιλογέα χρωμάτων είναι πάρα πολύ απλό. Απλά εκτελούμε την μέθοδο showDialog() της κλάσης JColorChooser όπως παρακάτω:

```
JColorChooser.showDialog(MainFrameView.getInstance(), "Background Color", new Color(Color.BLUE));
```

Οι τιμές που πρέπει να περάσουμε στην μέθοδο είναι σε ποιο παράθυρο ανήκει ο επιλογέας τον τίτλο του επιλογέα καθώς και ποια είναι η αρχική τιμή του χρώματος.

Στο παρακάτω σχήμα παρουσιάζεται ένας επιλογέας χρωμάτων.

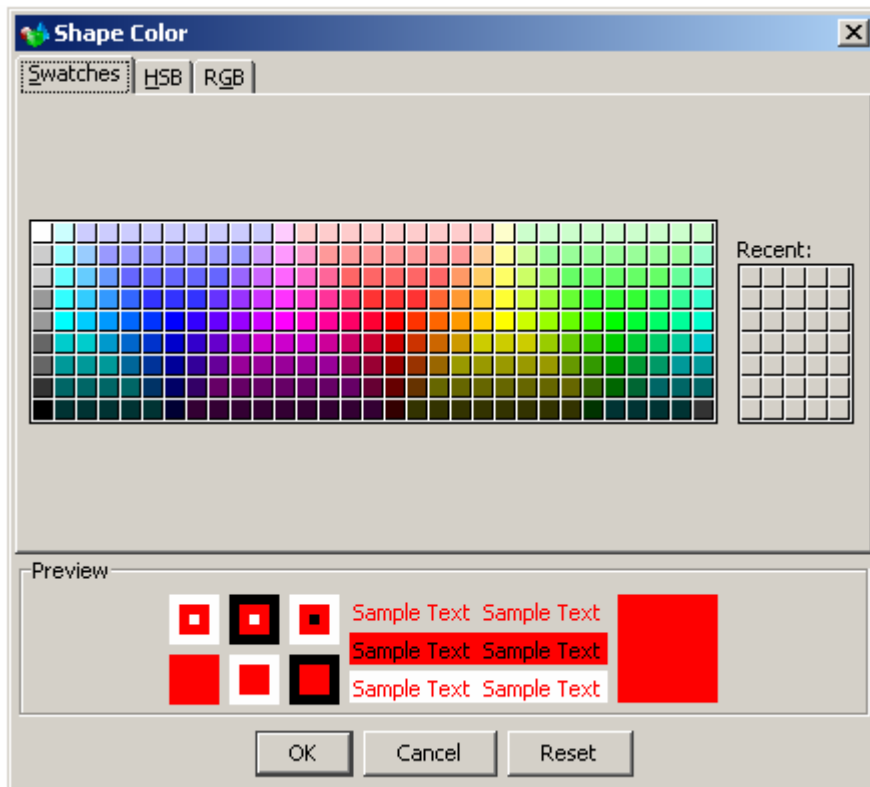


Figure 9 : Επιλογέας χρωμάτων

### 3.1.8. Μενού - JMenu –JMenuBar – JMenuItem

Το μενού σε μία εφαρμογή αποτελεί κύριο συστατικό λόγω της λειτουργικότητάς του καθώς και όλες της ευκολίες που παρέχει στο χρήστη.

Για την κατασκευή ενός μενού σε μια εφαρμογή χρειάζεται η χρησιμοποίηση των κλάσεων JMenu –JMenuBar – JMenuItem.

Η πρώτη είναι υπεύθυνη για τον χειρισμό της γραμμής μενού η δεύτερη είναι υπεύθυνη για τον χειρισμό των μενού που περιέχουν στοιχεία μενού ενώ η τρίτη είναι υπεύθυνη για τον χειρισμό των στοιχείων που υπάρχουν στα μενού.



Στο παρακάτω σχήμα παρουσιάζεται ένα μενού με υποσυστατικά.

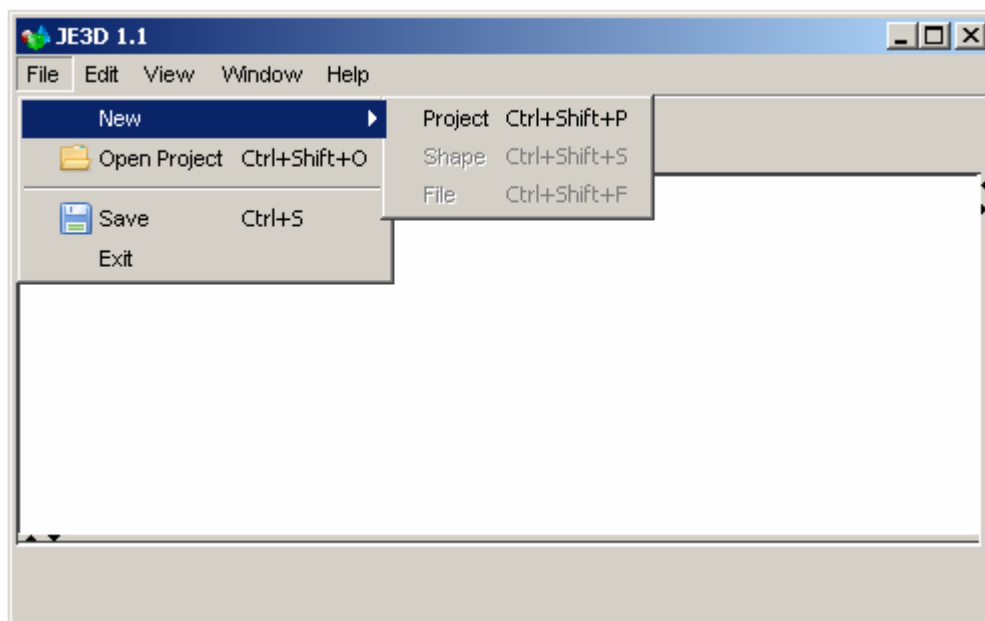


Figure 10: Swing Μενού

Για να δημιουργήσουμε ένα απλό μενού καλούμε πρώτα την κλάση `JMenuBar` η οποία κατασκευάζει μια γραμμή μενού και στην συνέχεια της περνάμε ένα μενού καλώντας την κλάση `JMenu`.

Αν θέλουμε να περάσουμε και υποσυστατικά στο μενού τότε καλούμε την κλάση `JMenuItem` Και περνάμε ένα αντικείμενο αυτής στο `JMenu`. Υπάρχει η δυνατότητα να βάλουμε ένα `menuItem` μέσα σε ένα άλλο. Τα `MenuItems` παράγουν γεγονότα δράσης τα λεγόμενα `Action Events` και λόγω αυτού ορίζουμε σε κάθε `menuItem` έναν ακροατή `Action Listener` ο οποίος καταγράφει το γεγονός και εκτελεί τις αντίστοιχες ενέργειες που του έχουμε ορίσει.

### 3.1.9. Γεγονότα - Swing Events

Το Swing αποτελείται από συστατικά γραφικού περιβάλλοντος τα οποία εκτελούν κάποιες ενέργειες του χρήστη. Κάθε φορά που ο χρήστης εκτελεί μια ενέργεια σε ένα από τα συστατικά του Swing το τρέχον συστατικό παράγει ένα γεγονός (Event) το γεγονός αυτό για να αξιοποιηθεί πρέπει να χειρισθεί από κάποιον ακροατή ώστε ο τελευταίος να εκτελέσει την κατάλληλη μέθοδο για τον χειρισμό του event. Στο Swing τα συστατικά δεν παράγουν όλα τον ίδιο τύπο Event καθότι έχουν σχεδιαστεί για να εκτελούν διαφορετικές λειτουργίες. Ένα συστατικό μπορεί να παράγει περισσότερα από ένα γεγονότα και αυτό εξαρτάτε από την ενέργεια του χρήστη. Για παράδειγμα ένα πλαίσιο `JFrame` μπορεί να παράγει δύο ξεχωριστά γεγονότα ανάλογα με την τρέχον κατάστασή του, αν το πλαίσιο ανοίξει ή κλείσει τότε παράγει `WindowEvents` αν μικρύνει ή μεγαλώσει τότε παράγει `ComponentEvent`, και για τα δύο αυτά γεγονότα έχει οριστεί ο κατάλληλος ακροατής ο οποίος με την σειρά του εκτελεί τις αντίστοιχες μεθόδους.

Στον παρακάτω πίνακα αναφέρονται μερικά από τα Events της βιβλιοθήκης AWT και της βιβλιοθήκης Swing τα οποία συναντώνται περισσότερο στις εφαρμογές Java. Η βιβλιοθήκη Swing καθότι είναι επέκταση του πακέτου AWT κληρονομεί και τα Events του τελευταίου.

<b>Ακροατές</b>	<b>Γεγονότα - Events</b>
ActionListener	Action events
AdjustmentListener	Adjustment events
ComponentListener	Component (move, size, hide, show) events
ContainerListener	Container (ad, remove component) events
FocusListener	Focus (gain, loss) events
ItemListener	Item events
MouseListener	Mouse buttons events
MouseMotionListener	Mouse motion events
MouseWheelListener	Mouse wheel events
TextListener	Text events
WindowFocusListener	Window focus events (new focus management framework)
WindowListener	Window events (non focus related)
WindowStateListener	Window state events
ChangeListener	Change events
DocumentListener	Text document events
TreeExpansionListener	Tree expand/collapse events
TreeModelListener	Tree model data events
TreeWillExpandListener	Tree expand/collapse pending events

Table 7 : Γεγονότα Swing - AWT

Για να χρησιμοποιήσει και να διακηρηστεί κανείς την παραγωγή και καταγραφή ενός γεγονόσ θα πρέπει να ορίσει έναν ακροατή σε ένα γραφικό συστατικό το οποίο μπορεί να παράγει γεγονότα. Παρακάτω φαίνεται ένα απλό παράδειγμα χρήσης πηγής - ακροατή με την υλοποίηση ενός κουμπιού. Πρώτα κατασκευάζουμε το κουμπί το οποίο είναι και η πηγή του γεγονόσ και στη συνέχεια του περνάμε έναν ακροατή στο κουμπί. Ο ακροατής αυτός υλοποιεί κάποιες μεθόδους όπου ανάλογα τον τύπο της ενέργειας εκτελείται και η κατάλληλη μέθοδος που στο παράδειγμα μας είναι η `actionPerformed`.

```
 JButton button = new JButton("Action");
  button.addActionListener( new ActionListener() {

      @Override
      public void actionPerformed(ActionEvent e) {
          System.out.println("Button Pressed");
      }
  });
```

### 3.1.10. Συμπέρασμα χρήσης Swing

Γενικά το Swing χρησιμοποιείται ευρέως στην αγορά για να καλύψει τόσο ανάγκες για web εφαρμογές όσο και για Stand alone εφαρμογές. Στο εμπόριο υπάρχουν πολλά προγράμματα τα οποία μπορούν να υποστηρίξουν την υλοποίηση προγραμμάτων σε Java Swing τα οποία δίνουν κίνητρα στους προγραμματιστές να αναπτύξουν τις εφαρμογές τους με Swing

## 3.2. Java3D

Το Java3D API είναι ένα πακέτο κλάσεων και συστατικών το οποίο προσφέρει στον προγραμματιστή την δυνατότητα να αναπαραστήσει τρισδιάστατα γεωμετρικά σχήματα χρησιμοποιώντας υψηλού επιπέδου εργαλεία παρέχοντας του την ικανότητα να τροποποιήσει την όψη και το σχήμα των γεωμετρικών σχημάτων , να αλλάξει τις συντεταγμένες τους , την θέση τους στο χώρο , όπως επίσης μπορεί να επεξεργαστεί και να τροποποιήσει το χρώμα το φόντο και γενικά την εμφάνισή τους.Επίσης του παρέχεται η ικανότητα να ενσωματώσει ήχο στο τρισδιάστατο αντικείμενο καθώς και άλλες διαδραστικές ικανότητες όπως κίνηση και αλληλεπίδραση με το ποντίκι.Όλες αυτές οι λειτουργίες και τα συστατικά οργανώνονται σε ένα γράφημα σκηνης με την μορφή δομής δένδρου.

Τέλος η Java3D παρέχει ένα πλήθος από χαρακτηριστικά τα οποία κάνουν το API ποιο προσιτό και ενδιαφέρον για χρήση.

Παρακάτω αναφέρουμε μερικά από τα χαρακτηριστικά αυτά.

- **Συμπεριφορά - Behavior**

Η Java3D υποστηρίζει πολλαπλούς τύπους συμπεριφοράς συμπεριλαμβάνοντας animation , κίνηση , ανίχνευση σύγκρουσης (collision detection) και μετασχηματισμό μιας εικόνας σε μία άλλη (morphing).

- **Ομίχλη – Fog**

Η Java3D υποστηρίζει συστατικά ομίχλης (Fog) τα οποία περιορίζουν τους θεατές να διακρίνουν ορισμένα αντικείμενα σε μία σκηνή.Για παράδειγμα η ομίχλη επιτρέπει να δημιουργηθεί ένα πραγματικό μοντέλο καταιγίδας σε ένα τρισδιάστατο παιχνίδι.

- **Γεωμετρία Geometry**

Η Java3D στηρίζεται στη τρισδιάστατη πρωτογενές γεωμετρία με σκοπό να δημιουργεί γεωμετρικά σχήματα.Η Java3D μπορεί να φωτοαποδόση σκηνές που έχουν δημιουργηθεί από υπάρχοντα κατασκευαστικά εργαλεία όπως 3d Studio Max.

- **Φωτισμός – Lights**

Η Java3D προσφέρει εργαλεία με τα οποία μπορούμε να φωτίσουμε τα τρισδιάστατα αντικείμενα. Επίσης υποστηρίζει διαφορετικές φόρμες από φωτισμό και προσφέρει χειρισμό για την ένταση , το χρώμα του φωτισμού και την κατεύθυνση.

- **Ήχος – Sound**

Ένα μοναδικό χαρακτηριστικό της Java3D είναι ότι υποστηρίζει την ενσωμάτωση ήχου.

- **Texture**

Η Java3D προσφέρει texture mapping για να συνάπτει εικόνες πάνω στα 3D αντικείμενα.

### 3.2.1. Γράφος Σκηνής

Ένα “εικονικό σύμπαν” Virtual Universe δημιουργείται από ένα γράφο σκηνής. Ένας γράφος σκηνής κατασκευάζεται χρησιμοποιώντας αντικείμενα από τις κλάσεις του Java3D πακέτου. Ένας γράφος σκηνής αποτελείται από αντικείμενα τα οποία προσδιορίζουν την γεωμετρία τον ήχο τον φωτισμό την τοποθεσία τον προσανατολισμό και την εμφάνιση ενός τρισδιάστατου αντικειμένου.

Η πιο συνηθισμένη σχέση μεταξύ των αντικειμένων ενός γραφήματος είναι η σχέση “πατέρα - παιδιού” ένα γκρουπ δομών μπορεί να έχουν περισσότερα από ένα παιδιά αλλά μόνο έναν πατέρα. Πάντα υπάρχει μόνο ένα μονοπάτι από την ρίζα μέχρι κάθε παιδί.

Κάθε κόμβος ορίζει μια κατάσταση πληροφορίας για το αντικείμενο που πρόκειται να παρουσιαστεί. Τέτοιες πληροφορίες μπορεί να είναι ο προσανατολισμός του αντικειμένου η ο φωτισμός η οποιαδήποτε άλλη πληροφορία έχουμε αναφέρει παραπάνω.

Στο παρακάτω σχήμα παρουσιάζονται τα σύμβολα που διέπουν ένα γράφημα σκηνής.

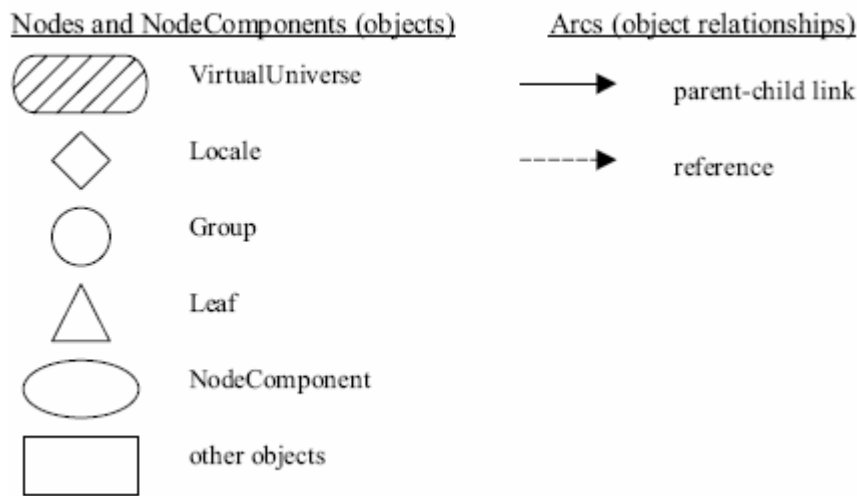


Figure 11 : Σύμβολα γράφου σκηνής

Κάθε ένα από τα παραπάνω σύμβολα αντιπροσωπεύει ένα μοναδικό αντικείμενο όταν χρησιμοποιείται από τον γράφο σκηνής. Τα δύο πρώτα σύμβολα VirtualUniverse και Locale αντιπροσωπεύουν αντικείμενα των ομώνυμων κλάσεων. Τα τρία επόμενα σύμβολα αντιπροσωπεύουν αντικείμενα των κλάσεων Group , Leaf και NodeComponet αντίστοιχα.

Παρακάτω επισυνάπτεται ένα γράφημα σκηνή το οποίο αποτελείται από τρία τρισδιάστατα αντικείμενα.

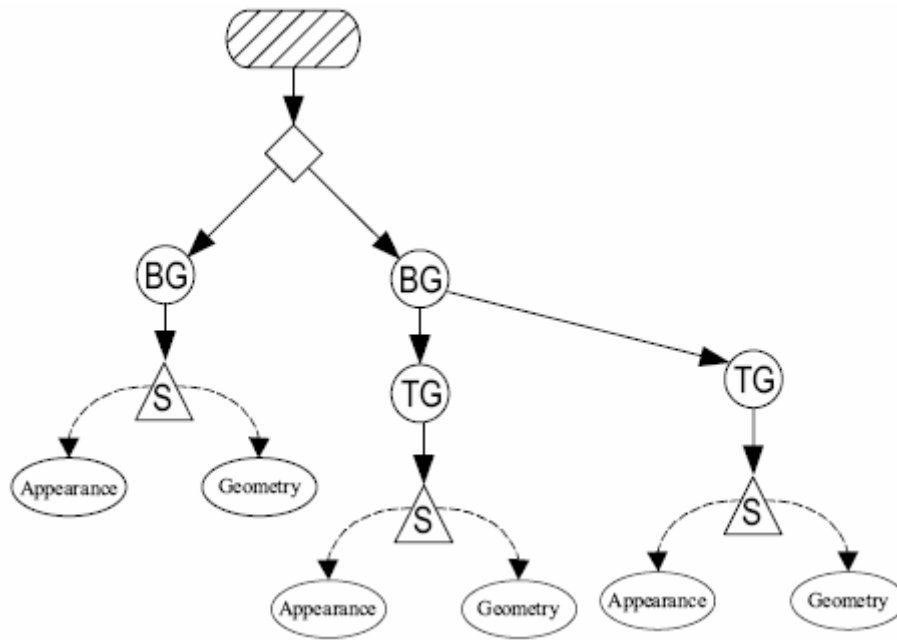


Figure 12 :Γράφος σκηνής αντικειμένων

### 3.2.2. Ιεραρχία Κλάσεων Java3D API

Ένας γράφος σκηνής (scene graph) αποτελείται κυρίως από αντικείμενα των κλάσεων VirtualUniverse , Local και ScreenGraphObjects.

Η κλάση ScreenGraphObject περιέχει δύο υποκλάσεις την Node και NodeComponent.Οι υποκλάσεις της κλάσης Node παρέχουν τα περισσότερα αντικείμενα σε ένα γράφο σκηνής.Ένα αντικείμενο της κλάσης Node είναι ή ένα Group Node ή ένα Leaf Node.Οι κλάσεις Group και Leaf είναι υπερκλάσεις σε ένα μεγάλο αριθμό υποκλάσεων του Java3D API.Παρακάτω παρουσιάζονται κάποια χαρακτηριστικά των κλάσεων αυτών.

- **Κλάση Node**

Η κλάση Node είναι μια abstract υπερκλάση των κλάσεων Leaf και Group.Η κλάση Node προσδιορίζει κάποιες κοινές σημαντικές μεθόδους για τις υποκλάσεις της.Οι υποκλάσεις της κλάσης Node συγκροτούν κατά ένα μεγάλο μέρος το γράφο σκηνής.

- **Κλάση Group**

Η κλάση Group είναι μία υπερκλάση η οποία χρησιμοποιείται για να ορίσει την τοποθεσία και τον προσανατολισμό των οπτικών αντικειμένων σε ένα virtual universe.

Υποκλάσεις της κλάσης Group είναι η TransformGroup και η BranchGroup στις οποίες θα αναφερθούμε παρακάτω.

- **Κλάση Leaf**

Η κλάση Leaf είναι μια υπερκλάση η οποία χρησιμοποιείται για να ορίσει το σχήμα τον ήχο και την συμπεριφορά των οπτικών αντικειμένων σε ένα virtual universe.Μερικές από τις υποκλάσεις της κλάσης Leaf είναι οι Shape3D , Light, Sound,Behavior.Τα αντικείμενα αυτά δεν έχουν παιδιά.

- **Κλάση NodeComponent**

Η κλάση αυτή είναι μια υπερκλάση η οποία χρησιμοποιείται για να ορίσει την γεωμετρία , την εμφάνιση , το texture και τέλος όλες τις ιδιότητες που σχετίζονται με το material ενός Shape3D αντικειμένου.

Παρακάτω συνάπτεται ένα σχήμα με την ιεραρχία των κλάσεων στο Java3D API.

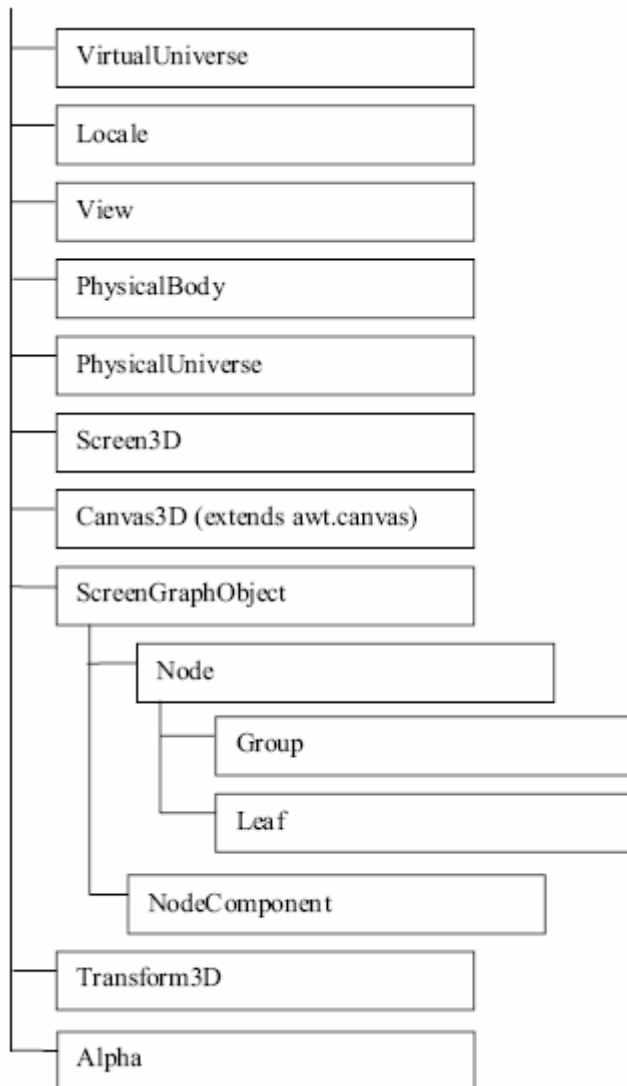


Figure 13 : Ιεραρχία κλάσεων Java3D

### 3.2.3. Η Κλάση Simple Universe

Η κλάση Simple Universe δημιουργεί αυτόματα ένα γρήγορο και εύκολο προς τον χρήστη περιβάλλον ώστε να κάνει τα προγράμματα της Java3D να εκτελεστούν με επιτυχία. Η χρησιμότητα της κλάσης αυτής είναι ότι δημιουργεί όλα τα απαραίτητα αντικείμενα που σχετίζονται με την προβολή του αντικειμένου σε ένα γράφο σκηνής. Πιο συγκεκριμένα η κλάση αυτή δημιουργεί μια τοποθεσία συμβάντος (Locale) , μία μοναδική πλατφόρμα προβολής (ViewingPlatform) καθώς και ένα αντικείμενο Viewer αρχικοποιώντας τα με τις αρχικές τους τιμές.

Με την χρήση της κλάσης αυτής ο προγραμματιστής κερδίζει πολύτιμο χρόνο στην υλοποίηση του γράφου σκηνής καθώς δεν ασχολείται με το κομμάτι που συσχετίζεται με το "View" του γράφου σκηνής, έτσι ο γράφος γίνεται πιο απλούστερος συντηρίσιμος και προσπελάσιμος.

Για την δημιουργία ενός αντικειμένου της κλάσης SimpleUniverse εκτελούμε την παρακάτω εντολή SimpleUniverse universe = new SimpleUniverse(). Έτσι με τον τρόπο αυτό δημιουργήθηκε μία τοποθεσία συμβάντος Locale , πλατφόρμα προβολής ViewingPlatform και ένας Viewer με τις αρχικές τιμές τους.

Στον παρακάτω πίνακα αναφέρονται οι κατασκευαστές της κλάσης SimpleUniverse.

Constructors	Description
SimpleUniverse()	Creates a locale, a single ViewingPlatform, and a Viewer object (both with their default values).
SimpleUniverse(Canvas3D canvas)	Creates a locale, a single ViewingPlatform (with default values), and a Viewer object.
SimpleUniverse(Canvas3D canvas, int numTransforms)	Creates a locale, a single ViewingPlatform, and a Viewer object The Viewer object uses default values for everything but the canvas.
SimpleUniverse(HiResCoord origin, int numTransforms, Canvas3D canvas, java.net.URL userConfig)	Deprecated. use ConfiguredUniverse constructors to read a configuration file
SimpleUniverse(HiResCoord origin, int numTransforms, Canvas3D canvas, java.net.URL userConfig, LocaleFactory localeFactory)	Deprecated. use ConfiguredUniverse constructors to read a configuration file
SimpleUniverse(int numTransforms)	Creates a locale, a single ViewingPlatform, and a Viewer object (with default values).
SimpleUniverse(ViewingPlatform viewingPlatform, Viewer viewer)	Creates the "view" side of the scene graph
SimpleUniverse(ViewingPlatform viewingPlatform, Viewer viewer, LocaleFactory localeFactory)	Creates the "view" side of the scene graph.

Table 8 : Κατασκευαστές κλάσης SimpleUniverse



### 3.2.4. Η Κλάση Canvas3D

Η κλάση Canvas3D παρέχει έναν καμβά σχεδίασης για τρισδιάστατες φωτοαποδόσεις. Η κλάση Canvas3D είναι μια επέκταση της κλάσης Canvas του πακέτου AWT Abstract Window Toolkit.

Κάθε αντικείμενο της κλάσης Canvas3D είναι μια επέκταση της κλάσης Canvas συμπεριλαμβάνοντας όμως αρκετές νέες πληροφορίες όπως το μέγεθος του καμβά σε pixels και την τοποθεσία του καμβά πάλι σε pixels.

Όλα τα αντικείμενα του Canvas3D περιέχουν μια αναφορά σε αντικείμενα κλάσης Screen3D και επειδή η τελευταία προσδιορίζει το μέγεθος ενός pixel σε φυσικές μονάδες το API της Java3D μετατρέπει το μέγεθος ενός καμβά από pixels σε μέτρα. Επίσης μπορεί να ορίσει την φυσική θέση και προσανατολισμό ενός καμβά στον φυσικό κόσμο.

Ένας καμβάς τοποθετείται σε ένα αντικείμενο της κλάσης SimpleUniverse με σκοπό να πάρει μέρος στο γράφο σκηνής. Στην Java3D κάθε τρισδιάστατο σχήμα για να φωτοαποδοθεί χρειάζεται έναν καμβά.

Για να δημιουργήσουμε ένα αντικείμενο της κλάσης Canvas3D εκτελούμε την ακόλουθη εντολή

```
Canvas3D canvas3d = new Canvas3D(SimpleUniverse.getPreferredConfiguration());
```

Στον παρακάτω πίνακα αναφέρονται οι κατασκευαστές της κλάσης Canvas3D.

Constructor	Description
Canvas3D(java.awt.GraphicsConfiguration graphicsConfiguration)	Constructs and initializes a new Canvas3D object that Java 3D can render into.
Canvas3D(java.awt.GraphicsConfiguration graphicsConfiguration, boolean offScreen)	Constructs and initializes a new Canvas3D object that Java 3D can render into.

Table 9: Κατασκευαστές κλάσης Canvas3D

### 3.2.5. Η κλάση TransformGroup

Η κλάση TransformGroup είναι μία επέκταση της κλάσης Group και περιέχει πληροφορίες για την χωρική μετατροπή των παιδιών του μέσω ενός αντικειμένου της κλάσης Transform3D.

Η μετατροπή αυτή αναφέρεται κυρίως στην θέση στον προσανατολισμό και στην κλίμακα μεγέθους όλων των αντικειμένων που έχουν σαν πατέρα το εκάστοτε TransformGroup.

Κάθε αντικείμενο της κλάσης TransformGroup αρχικοποιείται σύμφωνα με το δοθέν αντικείμενο της κλάσης Transform3D. Κάθε αντικείμενο της κλάσης αυτής περιέχει όλες τις πληροφορίες σχετικά με την θέση, τον προσανατολισμό και την κλίμακα σαν μια πληροφορία δεδομένων. Για να εφαρμοστέ η πληροφορία αυτή στο γράφημα σκηνής θα πρέπει να περάσουν οι τιμές αυτές στο αντικείμενο της κλάσης TransformGroup.

Για να δημιουργήσουμε ένα αντικείμενο της κλάσης TransformGroup χρειάζεται να εκτελέσουμε την παρακάτω εντολή TransformGroup group = new TransformGroup();

Στον παρακάτω πίνακα αναφέρονται οι κατασκευαστές της κλάσης TransformGroup.

Constructor	Description
TransformGroup()	Constructs and initializes a TransformGroup using an identity transform.
TransformGroup(Transform3D t1)	Constructs and initializes a TransformGroup from the Transform passed.

Table 10: Κατασκευαστές κλάσης TransformGroup

### 3.2.6. Συμπεριφορές – Behaviors

Με τον όρο Behavior στο API της Java3D εννοούμε κάθε ενέργεια προσδιοριζόμενη από τον χρήστη η οποία συμπεριλαμβάνεται στον γράφο σκηνής. Τέτοιες ενέργειες θα μπορούσαν να προέλθουν από το ποντίκι ή το πληκτρολόγιο οι οποίες μπορούν να επηρεάσουν και να ελέγξουν το τρισδιάστατο αντικείμενο κάθε φορά. Παρακάτω παρατάσσονται ορισμένες συμπεριφορές που χρησιμοποιήσαμε στην εφαρμογή μας οι οποίες συσχετίζονται με την λειτουργία του ποντικιού.

- **MouseRotate**

Η συμπεριφορά αυτή δίνει την ικανότητα στο χρήστη να ελέγξει την περιστροφή του τρισδιάστατου αντικειμένου στο χώρο μεταβάλλοντας μόνο την θέση του ποντικιού.

- **Mouse Translate**

Η συμπεριφορά αυτή δίνει την ικανότητα στο χρήστη να ελέγξει την τοποθεσία του αντικειμένου σε συντεταγμένες (X,Y). Εκτελείται με σύρσιμο του ποντικιού πατώντας το δεξί κουμπί του ποντικιού σταθερό.

- **Mouse Wheel Zoom**

Η συμπεριφορά αυτή δίνει στον χρήστη την ικανότητα να ελέγξει την μεγέθυνση ή συρρίκνωση του αντικειμένου στον Z άξονα μόνο. Για να εκτελεστή η λειτουργία επιτυχώς χρησιμοποιούμε το δεύτερο κουμπί του ποντικιού.

- **Mouse Zoom**

Η συμπεριφορά αυτή παράγει το ίδιο ακριβώς αποτέλεσμα με την προηγούμενη με την διαφορά ότι εκτελείται μόνο με κύλιση του ποντικιού και ταυτόχρονο πάτημα του δευτέρου κουμπιού.

Το Java3D API δίνει την ικανότητα να χρησιμοποιηθούν οι παραπάνω συμπεριφορές όλες μαζί ή κάθε μια ξεχωριστά ή συνδυασμός αυτών.

### 3.2.7. Φωτισμός – Lights

Το API της Java 3D προσφέρει κλάσεις και εργαλεία για τον χειρισμό και την ενσωμάτωση φωτισμού τόσο σε όλη την σκηνή όσο και σε συγκεκριμένα σημεία που επιλέγει ο προγραμματιστής. Σε μία σκηνή τρισδιάστατων γραφικών αντικειμένων ο φωτισμός προσφέρει να μετάγει την σκηνή από επίπεδη και μη κατανοητή σε ατμοσφαιρική εικονική πραγματικότητα οι οποία προσφέρει βάθος στα τρισδιάστατα γεωμετρικά σχήματα. Η Java3D προσφέρει 4 βασικούς τύπους φωτισμού οι οποίοι μπορούν να χρησιμοποιηθούν για τον φωτισμό μιας σκηνής μεμονωμένα ή να συνδυαστούν μεταξύ τους.

Το API προσφέρει κατάλληλες μεθόδους για κάθε ένα από τους τέσσερις φωτισμούς ώστε να γίνει δυνατός ο ορισμός ή η τροποποίηση της έντασης του φωτός, να επιλεγεί το κατάλληλο το χρώμα φωτός και να οριστεί ο προσανατολισμός.

Οι τέσσερις τύποι φωτός που ολοκληρώνουν το Java3D API είναι οι εξής : Ambient Light, Directional Light, Point Light, Spot Light. Παρακάτω αναφέρουμε κάποια χαρακτηριστικά αυτών.

- **Ambient Light**  
Το Ambient Light εφαρμόζεται σε όλη την σκηνή και η ένταση του είναι εξίσου ίδια προς όλες τις κατευθύνσεις.
- **Directional Light**  
Το Directional Light προσεγγίζει να οριστεί με πολύ μακρινές πηγές φωτός όπως ο ήλιος. Το Directional Light σε αντίθεση με το Ambient Light παρέχει φωτεινότητα μόνο σε μία κατεύθυνση την οποία ορίζει ο χρήστης. Η κατεύθυνση αυτή μπορεί να οριστεί δίνοντας τις επιθυμητές τιμές στις συντεταγμένες X, Y, Z του χώρου. Επίσης στο Directional Light όπως και στο Ambient Light μπορείς να ορίσεις το χρώμα του φωτός καθώς και την ένταση του.
- **Point Light**  
Το Point Light ακτινοβολεί από μέσα προς τα έξω από ένα δοθέν σημείο στον τρισδιάστατο χώρο της σκηνής. Το Point Light όπως και το Directional Light μπορεί να ορίσει το χρώμα του φωτός και την ένταση. Επίσης το Point Light παρέχει στο χρήστη μέθοδο υπεύθυνη για την αραίωση της έντασης του χρώματος συναρτήσει της απόστασης.
- **Spot Light**  
Το Spot Light είναι ένας φωτισμός ο οποίος δρα από ένα δοθέν σημείο προς μια δοθέν κατεύθυνση

### 3.2.8. Texturing

Με τον όρο Texturing στο API της Java 3D εννοούμε όλα τα συστατικά, τα εργαλεία και τις ενέργειες που ακολουθούμε για να εμπλουτίσουμε ένα γεωμετρικό τρισδιάστατο σχήμα με εικόνες. Ο σκοπός που γίνεται αυτό είναι για να αποδώσουμε σε ένα τρισδιάστατο σχήμα ένα φυσικό ρόλο. Για παράδειγμα αν μια σφαίρα την εμπλουτίσουμε με την εικόνα ενός χάρτη της γης τότε η σφαίρα αυτή μπορεί να θεωρηθεί σαν μια υδρόγειο σφαίρα. Ένα άλλο παράδειγμα είναι όταν σε ένα ορθογώνιο κουτί το εμπλουτίσουμε με την εικόνα μια πόρτας τότε το κουτί αυτό θα μοιάζει με πόρτα.

Τα βήματα που ακολουθούνται για την ολοκλήρωση της προσθήκης μιας εικόνας σε ένα γεωμετρικό σχήμα είναι τα εξής:

#### 1. Προετοιμασία της εικόνας

Για να εμπλουτιστεί ένα τρισδιάστατο σχήμα με μια εικόνα θα πρέπει το μέγεθος της εικόνας να είναι δύναμη του 2, επίσης θα πρέπει να τα αρχεία να είναι αποθηκευμένα σε format που να μπορεί να διαβαστεί από το API όπως GIF, JPG.

#### 2. Φόρτωμα της εικόνας

Το φόρτωμα (Load) μια εικόνας μπορεί να γίνει με τους γνωστούς τρόπους που σου παρέχει η Java για την προσπέλαση αρχείων, παρόλα αυτά το API της Java3D σου δίνει την δυνατότητα να φορτώσεις μια εικόνα εκτελώντας μόνο δύο εντολές:

```
TextureLoader loader = new TextureLoader("stripe.gif", this);
ImageComponent2D image = loader.getImage();
```

Στον Constructor της κλάσης TextureLoader περνάμε το URL του αρχείου και στο δεύτερο πεδίο έναν image observer για να πετύχουμε ασύγχρονο φόρτωμα εικόνας.

### 3. Ανάθεση εικόνας στην εμφάνιση του αντικειμένου

Μετά το φόρτωμα της εικόνας για να περάσουμε την εικόνα στο αντικείμενο μας θα πρέπει να την περάσουμε σε ένα αντικείμενο της κλάσης Appearance η οποία είναι υπεύθυνη για την εμφάνιση του αντικειμένου. Έτσι εκτελούμε τον παρακάτω κώδικα ο οποίος περνάει την εικόνα σε ένα αντικείμενο της κλάσης Texture2D.

```
TextureLoader loader = new TextureLoader("stripe.jpg", this);
Texture2D texture = new Texture2D();
texture.setImage(0, image);
Appearance appear = new Appearance();
appear.setTexture(texture);
```

### 4. Προσδιορισμός συντεταγμένων

Τέλος πρέπει να ορίσουμε την τοποθέτηση του αντικειμένου πάνω στο γεωμετρικό σχήμα, όπου αυτό γίνεται με τον προσδιορισμό των συντεταγμένων του αντικειμένου της εικόνας. Πρέπει να αναφέρουμε ότι στα primitive αντικείμενα που μας παρέχει το API της Java 3D δεν χρειάζεται να ορίσουμε συντεταγμένες καθότι παρέχεται μια επιλογή με την οποία καθορίζονται αυτόματα οι συντεταγμένες για την τοποθέτηση της εικόνας. Όπως στο παρακάτω παράδειγμα.

```
Sphere sphere = new Sphere(1.0f, Primitive.GENERATE_TEXTURE_COORDS,
appear)
```

Στο παρακάτω σχήμα παρουσιάζεται ένας κύβος στον οποίο έχει τοποθετηθεί μια εικόνα τύπου JPG.

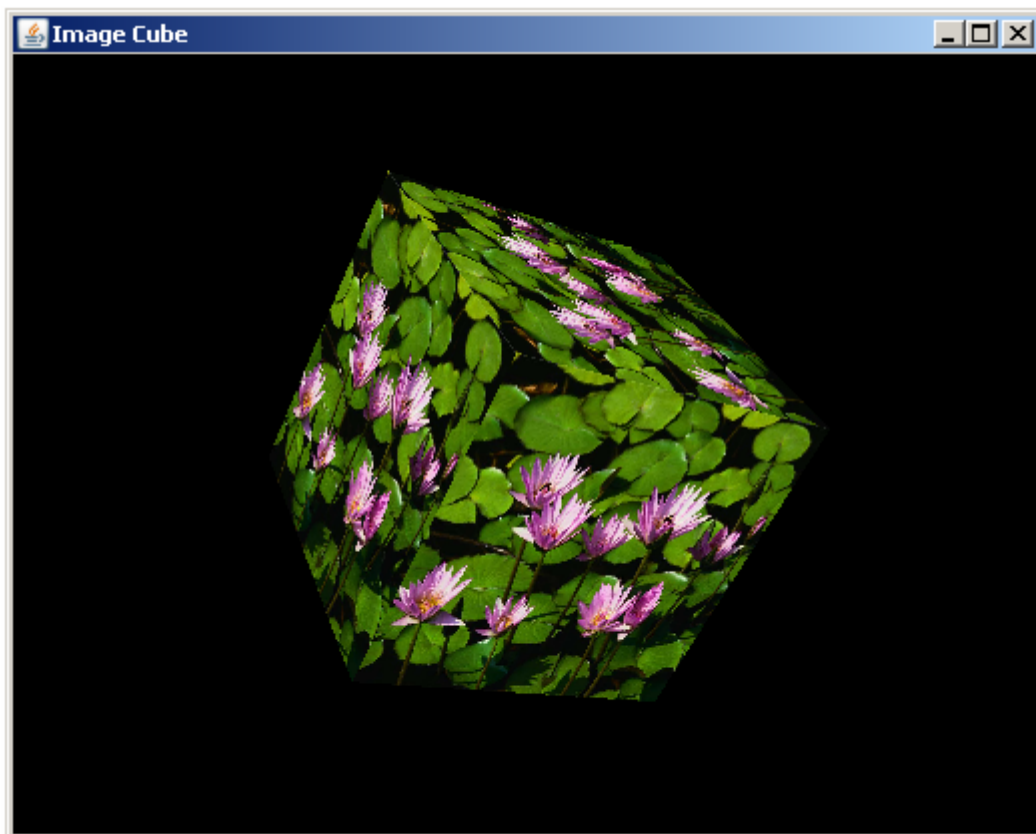


Figure 14: Texture σε κύβο.

### 3.2.9. Primitive Objects

Το API της Java 3D παρέχει στο χρήστη την ικανότητα να δημιουργήσει γρήγορα τρισδιάστατα αντικείμενα καλώντας κάποιες κλάσεις που βρίσκονται στο `utils` πακέτο του API και ποιο συγκεκριμένα στο `com.sun.j3d.utils.geometry`. Για την δημιουργία αυτών το γεωμετρικών σχημάτων χρειάζεται να περάσουμε σαν input κάποια στοιχεία όπως για παράδειγμα για την δημιουργία μιας σφαίρας χρειάζεται να περάσουμε την ακτίνα της σφαίρας και κάποιες σταθερές. Τα primitive αντικείμενα που μπορούμε να δημιουργήσουμε μέσω του API είναι τα εξής :

- **Sphere**  
Για την κατασκευή μιας γεωμετρικής σφαίρας χρειάζεται να δώσουμε την ακτίνα και την ανάλυση (resolution) της σφαίρας.
- **Cone**  
Για την κατασκευή ενός κώνου χρειάζεται να ορίσουμε την ακτίνα και το ύψος του κώνου.  
Ο κώνος έχει σαν κεντρικό άξονα τον άξονα Y.
- **Box**  
Για την κατασκευή ενός τρισδιάστατου κουτιού πρέπει να ορίσουμε το ύψος το πάχος και το μήκος.
- **Cylinder**  
Για την κατασκευή ενός Κυλίνδρου χρειάζεται να ορίσουμε την ακτίνα και το ύψος του κυλίνδρου. Ο κεντρικός άξονας του κυλίνδρου ορίζεται ο Y άξονας.

Το API της Java3D σου παρέχει τα παραπάνω primitive αντικείμενα τα οποία μπορούν να μετατραπούν σε οποιοδήποτε γεωμετρικό σχήμα με την χρησιμοποίηση των καταλλήλων εργαλείων που σου παρέχει το API.

Στα παρακάτω σχήματα παρουσιάζονται τα primitive αντικείμενα που παρέχει το API της Java3D.

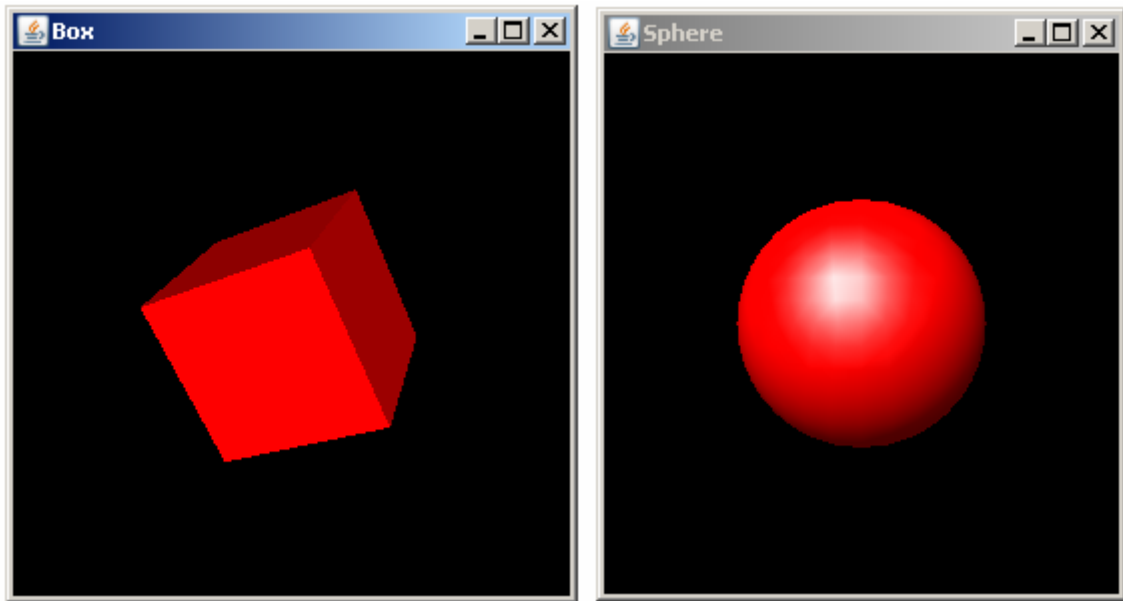


Figure 15: Box – Sphere (Κουτί και Σφαίρα αντίστοιχα)

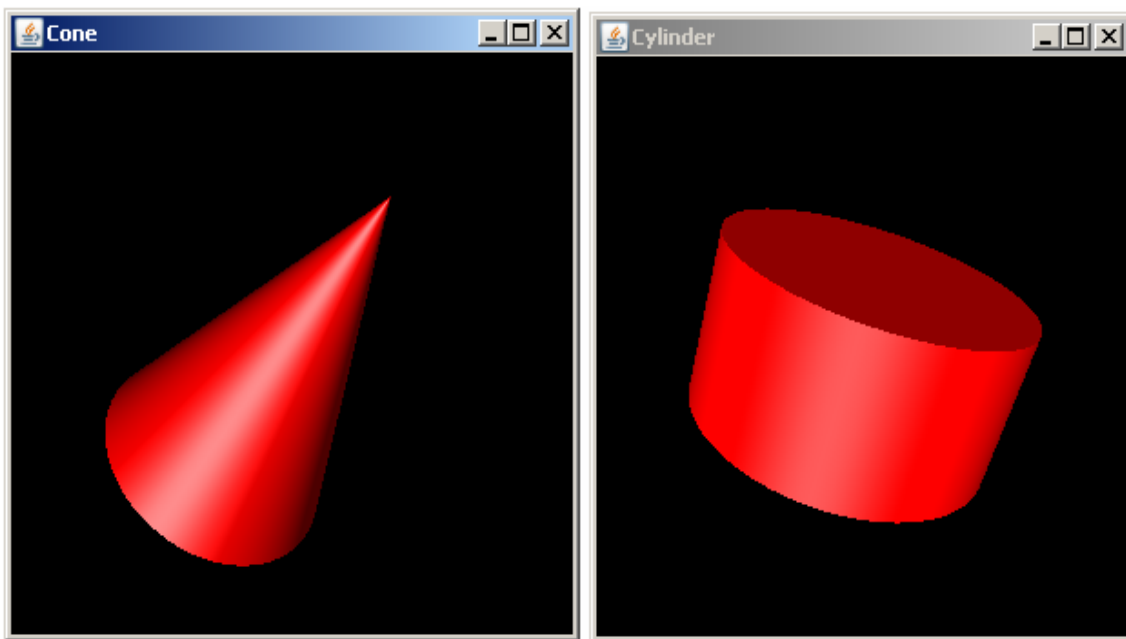


Figure 16: Cone – Cylinder (Κώνος και Κύλινδρος αντίστοιχα)

### 3.3. Freemarker Templates

Όπως έχουμε αναφέρει και σε παραπάνω κεφάλαιο η FreeMarker Template είναι μία μηχανή προτύπων η οποία παράγει σαν έξοδο ένα κείμενο. Η μηχανή αυτή είναι βασισμένη κυρίως στην Java και χρησιμοποιείται για την παραγωγή HTML σελίδων και για την παραγωγή αυτόματου κώδικα όπως την χρησιμοποιήσαμε και εμείς στην εφαρμογή μας. Για να επιτευχθεί αυτό γίνεται ένας συνδυασμός Java κώδικα και αρχείων freemarker template.

Για παράδειγμα ένα αρχείο template με format file.ftl περιέχει ένα πρότυπο κειμένου με διάφορες μεταβλητές οι οποίες θα αρχικοποιηθούν από τα αντικείμενα της Java. Έτσι στο ένα μέρος της εφαρμογής μας οργανώνουμε το πρότυπο (module) του κώδικα που θα παραχθεί ορίζοντας του κάποιες μεταβλητές οι οποίες θα πάρουν δυναμικά τιμές από τα αντικείμενα του source κώδικα. Με την λειτουργία αυτή κάνουμε τον μηχανισμό κατασκευής Java κώδικα πιο απλό. Παρακάτω παρουσιάζονται κάποια παραδείγματα για το πως δουλεύει ο μηχανισμός αυτός.

Έστω ότι θέλουμε να εμφανίσουμε την παρακάτω HTML σελίδα η οποία καλωσορίζει ένα χρήστη και εμφανίζει το όνομά του, στη συνέχεια αναζητά τυχαία σε μία βάση δεδομένων και του παρουσιάζει ένα δώρο. Όπως είναι προφανές εδώ το όνομα του χρήστη και το δώρο είναι δυναμικά πεδία και αλλάζουν συνεχώς. Για τον λόγο αυτό δημιουργούμε ένα πρότυπο αρχείου .ftl το οποίο θα περιέχει το ίδιο κείμενο με κάποιες αλλαγές σύμφωνα με το συντακτικό της γλώσσας ftl.

#### Αρχικό στατικό κείμενο

```
<html>
<head>
  <title>Welcome!</title>
</head>
<body>
  <h1>Welcome Maria!</h1>
  <p>Our latest product:
  <a href="gift/gidt.html">car</a>!
</body>
</html>
```

#### Τελικό δυναμικό κείμενο ftl.

```
<html>
<head>
  <title>Welcome!</title>
</head>
<body>
  <h1>Welcome ${user}!</h1>
  <p>Our latest product:
  <a href="${dataGift.url}">${dataGift.name}</a>!
</body>
</html>
```

Όπως παρατηρούμε στο δυναμικό κείμενο ftl τα δυναμικά πεδία αντικαταστάθηκαν από το παρακάτω συντακτικό `${...}` με τον όρο αυτό δηλώνουμε ότι μέσα στα άγκιστρα ακολουθεί δυναμική μεταβλητή η οποία θα πάρει τιμές από τα αντικείμενα της Java. Η τιμή που θα περάσει στα άγκιστρα έχει οριστεί από την μεταβλητή “user”.

Μέσα στα άγκιστρα μπορείς να περάσεις από τιμές μέχρι και αντικείμενα της Java. Ο τρόπος για να γίνει αυτό είναι πολύ απλός. Πρώτα ορίζεις έναν φάκελο οπου θα περιέχει τα αρχεία ftl και στη συνέχεια περνάς σε ένα `HashMap` όλες τις μεταβλητές και τα αντικείμενα που θες να μεταφέρεις στο ftl. Ποιο συγκεκριμένα έστω ότι θέλαμε στο παραπάνω παράδειγμα να περάσουμε στη δυναμική μεταβλητή user το όνομα Nick τότε θα έπρεπε να κατασκευάσουμε ένα `HashMap` και να περάσουμε σαν τιμή το όνομα Nick στο κλειδί user.

```
Map<String , String> params = new HashMap<String , String>();
params.put(“user” , “Nick”);
```

Με τον ίδιο τρόπο μπορούμε να περάσουμε και αντικείμενα αλλάζοντας βέβαιά τον τύπο του `HashMap`. Όταν περνάμε ένα αντικείμενο σε ένα ftl αρχείο το συντακτικό της μηχανής freemarker σου δίνει την δυνατότητα να προσπελάσεις και τις μεθόδους του αντικειμένου. Στο παραπάνω δυναμικό κείμενο ftl υπάρχει η εντολή `${dataGift.url}` από το συντακτικό καταλαβαίνουμε ότι στο `HashMap` έχει περαστεί σαν value ένα αντικείμενο με το όνομα dataGift το αντικείμενο αυτό έχει κάποιες μεθόδους την `getUrl()` και την `getName()` , η γλώσσα ftl σου δίνει την ικανότητα να προσπελάσεις μια μέθοδο ενός αντικειμένου αναφέροντας το αντικείμενο (“.”) και το όνομα της μεθόδου χωρίς όμως τα αρχικά get και set. Για παράδειγμα:

```
public class NewClass {
    String name;
    String url;

    public String getURL() {
        return url;
    }

    public String getName() {
        return name;
    }
}
```

```
NewClass dataGift = new NewClass ();
params.put(“dataGift” , dataGift);
```

Έτσι με τον παραπάνω τρόπο μπορούμε να προσπελάσουμε και αντικείμενα οποιοδήποτε τύπου όπως λίστες. Το συντακτικό της γλώσσας ftl υποστηρίζει και τη χρήση επαναλαμβανόμενων βρόχων for καθώς και την χρήση συνθηκών if – else – else if /switch-case -break.

Στην εφαρμογή μας JE3D1.1 κάνουμε χρήση της μηχανής ftl για να δημιουργήσουμε το δυναμικό κείμενο κώδικα που παράγεται μετά την επεξεργασία των τρισδιάστατων αντικειμένων.



## 4. Ανάλυση Λογισμικού

### 4.1. Τι είναι ανάλυση λογισμικού

Με τον όρο ανάλυση λογισμικού εννοούμε κάθε λειτουργία - διαδικασία που εφαρμόστηκε για την ανάπτυξη και ολοκλήρωσης ενός λογισμικού. Τέτοιες λειτουργίες μπορεί να είναι η ανάλυση του λογισμικού και των απαιτήσεων , η σχεδίαση του λογισμικού , η τεκμηρίωση της σχεδίασης , η υλοποίηση , και η έρευνα για πιθανές μετατροπές ή αναβαθμίσεις του λογισμικού.

Με την ανάλυση του λογισμικού αναλύονται και καταγράφονται οι απαιτήσεις του λογισμικού , με την σχεδίαση αναπτύσσονται σχεδιαστικές λύσεις οι οποίες υλοποιούνται για την περαιτέρω ανάπτυξη του λογισμικού , στη συνέχεια με την τεκμηρίωση γίνεται ο έλεγχος αν η σχεδίαση του λογισμικού καλύπτει τις απαιτήσεις που καταγράφηκαν στην αρχή, αν το λογισμικό καλύπτει τις απαιτήσεις τότε η διαδικασία μεταφέρεται στο επόμενο βήμα που είναι η ανάπτυξη του λογισμικού με την βοήθεια διάφορων τεχνολογιών και εργαλείων. Τέλος το τελευταίο στάδιο είναι η έρευνα για πιθανή μετατροπή του λογισμικού ώστε αυτό να αντεπεξέλθει στις μελλοντικές απαιτήσεις που μπορεί να προκύψουν. Όταν ολοκληρωθεί η έρευνα ξεκινούν πάλι τα βήματα για την υλοποίηση των εξελικτικών αλλαγών του λογισμικού.

#### 4.1.1. Ανάλυση και Καταγραφή Απαιτήσεων

Όταν πρόκειται να αναπτυχθεί ένα λογισμικό ή μια εφαρμογή πρώτα θα πρέπει να καταγραφούν και να αναλυθούν οι απαιτήσεις του λογισμικού ή της εφαρμογής. Οι απαιτήσεις που μπορεί να έχει ένα λογισμικό μπορεί να είναι τεχνικές δηλαδή σε ποια συστήματα θα εφαρμοστεί ή ποιες τεχνολογίες θα πρέπει να ακολουθήσει ώστε να είναι συντηρήσιμο γρήγορο και αξιόπιστο και οι λειτουργικές που ορίζονται από το ποιος είναι ο σκοπός του λογισμικού ποιος χρήστης ή ποια συστήματα θα χρησιμοποιούν αυτό το λογισμικό και ποιες ανάγκες θα καλύψει.

Στην πρώτη κατηγορία των τεχνικών απαιτήσεων οι τεχνικοί (προγραμματιστές , τεχνικοί δικτύων κτλ) είναι το αρμόδια πρόσωπα για να αναλύσουν και να καταγράψουν οτιδήποτε χρειάζεται για την ανάπτυξη του λογισμικού.

Στην δεύτερη κατηγορία που είναι η κατηγορία των λειτουργικών απαιτήσεων του λογισμικού σύνηθες πρόσωπα για την αναφορά των απαιτήσεων είναι οι χρήστες και κάποιες φορές ο γενικός σκοπός χρήσης του λογισμικού.

Για να επιτευχθεί η σωστή καταγραφή ο εκάστοτε υπεύθυνος για την ανάπτυξη του λογισμικού θα πρέπει να έρθει σε επαφή με τους χρήστες για μια συνέντευξη αλλά και με ανθρώπους που γνωρίζουν καλά τι θα διαχειρίζεται το λογισμικό ή η εφαρμογή.

Όταν καταγραφούν όλες οι απαιτήσεις τότε αναλύεται το σύστημα και σχεδιάζονται λύσεις για το πως τεχνικά το σύστημα θα καλύψει τις λειτουργικές απαιτήσεις των χρηστών αλλά και γενικά όλου του σκοπού που πρόκειται να εξυπηρετήσει.

Συνήθως οι απαιτήσεις μπορεί να αλλάξουν πολλές φορές κατά την διάρκεια της ανάπτυξης ενός λογισμικού αλλά δεν θα παρεκκλίνουν πολύ από τις αρχικές απαιτήσεις γιαυτό πριν ξεκινήσει η τεχνική ανάπτυξη μιας εφαρμογής – λογισμικού θα πρέπει να έχουν παρουσιαστεί διάφορα δείγματα ανάπτυξης και σεναρίων χρήσης για την τεκμηρίωση της σχεδίασης.

#### **4.1.2. Σενάρια Χρήσης**

Τα σενάρια Χρήσης ενός λογισμικού αναφέρονται στην καταγραφή δραστηριοτήτων που αφορούν το λογισμικό και την οντότητα που θα χρησιμοποιεί το λογισμικό. Έτσι με τον τρόπο αυτό γίνεται λεπτομερής καταγραφή των λειτουργιών ενός λογισμικού από το input του χρήστη μέχρι το αποτέλεσμα που θα παράγει το λογισμικό. Ένα σενάριο χρήσης μπορεί να καλύπτει όλο το εύρος των λειτουργιών της εφαρμογής ή μόνο ένα μέρος αυτής.

Με τη καταγραφή των σεναρίων χρήσης σε μια εφαρμογή γίνεται ευκολότερο να προσδιοριστούν πιθανά προβλήματα αλληλεπίδρασης και επικοινωνίας που μπορεί να εμφανιστούν μεταξύ του χρήστη και της διεπαφής.

Με την καταγραφή των σεναρίων σε μια διεπαφή ορίζονται οι ρόλοι που έχουν οι χρήστες αλλά και ελέγχεται η αποδοτικότητα του συστήματος. Απόρροια των παραπάνω είναι να σχεδιαστούν νέες λύσεις για πιθανά προβλήματα που προέκυψαν ή να βελτιωθούν κάποιες υπάρχουσες δραστηριότητες με σκοπό να κάνουν την εφαρμογή ή το λογισμικό πιο εύχρηστο και προσιτό στο χρήστη. Με την ανάλυση των σεναρίων χρήσης πολλές φορές επηρεάζεται το GUI του λογισμικού (αν απαρτίζεται από GUI), δηλαδή καταγράφονται οι χρόνοι και οι κινήσεις – βήματα που χρειάζεται ένας χρήστης για να ολοκληρώσει το σενάριο. Αν οι χρόνοι και οι κινήσεις - βήματα δεν καλύπτουν τις προδιαγραφές τότε μπορεί να αλλαχτεί και να βελτιωθεί το GUI με σκοπό να είναι πιο εύχρηστο στο χρήστη.

Κατά την δημιουργία ενός σεναρίου χρήσης θα πρέπει να ορίζεται ο στόχος και ο σκοπός που εξετάζεται καθώς και οι εμπλεκόμενοι.

#### **4.1.3. Ανάπτυξη Λογισμικού**

Αφού ολοκληρωθούν οι έρευνες για την καταγραφή των απαιτήσεων και τον σχεδιασμό σεναρίων αναπτύσσεται το λογισμικό με την χρήση διαφόρων εργαλείων και τεχνολογιών. Κατά την ανάπτυξη του λογισμικού θα πρέπει να γίνεται έλεγχος για την κάλυψη των απαιτήσεων στο στάδιο που βρίσκεται το λογισμικό καθώς και για την αποφυγή πιθανών τεχνικών προβλημάτων που μπορεί να προκύψουν ύστερα από συνεχές σενάρια χρήσης του λογισμικού. Αν κατά την ανάπτυξη εμφανιστεί κάποιο σχεδιαστικό λάθος ή προκύψουν νέες επιπρόσθετες απαιτήσεις τότε σταματάει η ανάπτυξη και εκτελούνται τα αρχικά στάδια που αναφέραμε πιο πάνω.

Σκοπός στην ανάπτυξη του λογισμικού είναι να ολοκληρωθεί το τεχνικό κομμάτι της εφαρμογής μετατρέποντας την πιο αποδοτική, γρήγορη, εύχρηστη και αξιόπιστη.

#### **4.1.4. Έλεγχος Λογισμικού**

Ο έλεγχος του λογισμικού εμφανίζεται κάθε φορά που ένα κομμάτι ή ολόκληρη η εφαρμογή – λογισμικό είναι έτοιμη προς παραγωγή. Ο έλεγχος πρέπει να γίνεται από πολλούς χρήστες και να έχει αρχή και τέλος. Για παράδειγμα για τον έλεγχο μιας εφαρμογής που αποθηκεύει τα στοιχεία ενός χρήστη σε μια βάση δεδομένων ο έλεγχος θα πρέπει να γίνει και στο GUI της εφαρμογής αλλά και στην βάση δεδομένων. Ένα λογισμικό – εφαρμογή είναι έτοιμα μόνο αν έχουν περάσει επιτυχώς τον έλεγχο πιστότητας, αποδοτικότητας ευχρηστίας αλλά και τον τεχνικό έλεγχο.

## 4.2. JEditor3D1.1 case study

Η εφαρμογή εστιάζει στη ανάπτυξη ενός εργαλείου το οποίο θα προσφέρει στον χρήστη την ικανότητα ανάπτυξης και επεξεργασίας τρισδιάστατων αντικειμένων χωρίς να σπαταλήσει πολύ χρόνο. Επίσης η εφαρμογή δίνει την δυνατότητα στο χρήστη να εργαστεί με αρχεία που αναφέρονται σε γλώσσες προγραμματισμού και μη. Επίσης κύριος σκοπός της εφαρμογής είναι να παράγει στο χρήστη ένα αρχείο Java το οποίο να αποτελεί ένα module βασισμένο στο τρισδιάστατο αντικείμενο που ανέπτυξε και χρησιμοποίησε.

Τέλος ο χρήστης θα μπορεί να τροποποιήσει το αρχείο αυτό να το εμπλουτίσει και να το συμπεριλάβει στις εφαρμογές του.

### 4.2.1. Περιγραφή Εφαρμογής JEditor3D1.1

Ο Editor JEditor3D1.1 ο οποίος αναπτύχθηκε κυρίως από τεχνολογίες της Java και Java3D έχει ως σκοπό να εξυπηρετήσει τα παρακάτω:

- Δημιουργία και αποθήκευση αρχείων στο χώρο εργασίας του χρήστη. Τα αρχεία αυτά ορίζονται ως τα αρχεία που έχει δημιουργήσει ο χρήστης κατά την χρησιμοποίηση του editor.
- Δημιουργία και επεξεργασία τρισδιάστατων αντικειμένων.
- Αυτόματη δημιουργία Java source κώδικα του αντικειμένου που δημιούργησε ο χρήστης.
- Παροχή διαφόρων εργαλείων για την επεξεργασία και μετατροπή των τρισδιάστατων αντικειμένων.
- Υποστήριξη αρχείων Java , XML , TXT
- Υποστήριξη ξεχωριστού περιβάλλοντος για την σχεδίαση και την ανάγνωση του κώδικα του αντικειμένου.
- Παροχή ομαδοποιημένων πληροφοριών για την τρέχουσα κατάσταση του τρισδιάστατου αντικειμένου.
- Παροχή editor για την επεξεργασία αρχείων Java , xml , txt.

### 4.2.2. Γενικό Σενάριο Χρήσης του JEditor3D1.1

Ένα γενικό σενάριο χρήσης του editor θα μπορούσε να είναι το παρακάτω.

Ας υποθέσουμε ότι ένας χρήστης θέλει να αναπτύξει τρισδιάστατες εφαρμογές βασισμένες στο Java3D API. Χρησιμοποιώντας την εφαρμογή για πρώτη φορά δημιουργεί ένα project σε ένα workspace το οποίο έχει επιλέξει αυτός , για να γίνει αυτό επιλέγει από το μενού την εντολή File->New Project ή πληκτρολογεί Ctrl -Shift-P. Αφού δώσει το όνομα του project και πατήσει ok δημιουργούνται στο χώρο εργασίας του οι αντίστοιχοι κενοί φάκελοι στους οποίους θα αποθηκεύσει τα επόμενα αρχεία που θα δημιουργήσει.

Στη συνέχεια ο χρήστης επιλέγει να δημιουργήσει ένα νέο αρχείο είτε κάνοντας δεξί κλικ στο δένδρο αρχείων που δημιουργήθηκε είτε πληκτρολογεί Ctrl-Shift-F. Έτσι εμφανίζεται ένα modal παράθυρο στο οποίο ο χρήστης μπορεί να επιλέξει τον τύπο του

αρχείου δηλαδή αν θέλει να επιλέξει ένα Java αρχείο ή ένα xml , txt ή να δημιουργήσει ένα τρισδιάστατο αντικείμενο. Ο χρήστης παρατηρεί ότι αν επιλέξει ένα από τα τρία αρχεία Java ,xml , txt το μόνο που χρειάζεται να συμπληρώσει είναι το όνομα του αρχείου, ενώ αν επιλέξει την επιλογή Shape τότε εμφανίζεται μια λίστα με Primitive αντικείμενα από την οποία ο χρήστης μπορεί να επιλέξει ένα από αυτά δίνοντας πάλι ένα όνομα στο αρχείο. Έστω ότι την πρώτη φορά επιλέγει να δημιουργήσει ένα απλό αρχείο Java τότε δημιουργείται ένα αρχείο Java στο χώρο εργασίας του χρήστη και ανοίγει ένα tab κειμένου στον editor έτοιμο για επεξεργασία επίσης εμφανίζεται και το αρχείο με την κατάληξη .java στο δενδρικό νήμα. Ο χρήστης εδώ έχει την ικανότητα να γράψει κώδικα ο οποίος στην συνέχεια θα αποθηκευτεί στο αντίστοιχο αρχείο που βρίσκεται στον φάκελο εργασίας του. Η αποθήκευση των αλλαγών που κάνει ο χρήστης πάνω σε ένα αρχείο γίνεται με το πάτημα του κουμπιού Save στην μπάρα εργαλείων.

Έστω ότι ο χρήστης την δεύτερη φορά που επιχειρεί να δημιουργήσει ένα αρχείο επιλέγει την κατηγορία Shape. Με την επιλογή αυτή ο χρήστης καλείται να επιλέξει ένα από τα Primitive αντικείμενα που παρέχει το API καθώς να δώσει μια ονομασία στο αρχείο που πρόκειται να δημιουργήσει. Όταν πατήσει OK στο παράθυρο διαλόγου τότε δημιουργείται ένα καινούργιο tab στο κυρίως παράθυρο της εφαρμογής με την ονομασία του αρχείου που δόθηκε , επίσης δημιουργείται και το σχετικό αρχείο Java στο φάκελο εργασίας του χρήστη με τον σχετικό αυτοδημιούργητο source κώδικα. Εδώ ο χρήστης παρατηρεί ότι στο κεντρικό frame της εφαρμογής είναι ενεργοποιημένα δυο κουμπιά το Design View και το Source View. Το πρώτο κουμπί χρησιμοποιείται για σχεδιασμό την ανάπτυξη και την επεξεργασία του αντικειμένου και όταν είναι επιλεγμένο τα tabs που σχετίζονται με τα εργαλεία επεξεργασίας του αντικειμένου στο κάτω δεξιά frame είναι ενεργοποιημένα ώστε ο χρήστης να τα χρησιμοποιήσει. Καθώς ο χρήστης έχει επιλέξει το Design View μπορεί και κάνει σε πραγματικό χρόνο αλλαγές πάνω στο τρισδιάστατο αντικείμενο. Έστω ότι θέλει να αλλάξει το μέγεθος , την τοποθέτηση , ή τον προσανατολισμό του αντικειμένου θα χρησιμοποιήσει το tab transformer explorer το οποίο σχετίζεται με τα παραπάνω. Καθώς ο χρήστης μετακινεί τυχαία έναν από τους διαθέσιμους ράβδους ολίσθησης οι αλλαγές του χρήστη παρατηρούνται και στο αντικείμενο χωρίς ο χρήστης να πρέπει να κάνει αποθήκευση των αλλαγών αλλά και στον πίνακα Γενικές Ιδιότητες ο οποίος παρουσιάζει ολοκληρωμένα όλα τα στοιχεία του αντικειμένου και τις τιμές που αυτά έχουν από την δημιουργία μέχρι την τελική επεξεργασία του αντικειμένου. Έστω στη συνέχεια ότι ο χρήστης θέλει να εφαρμόσει διαδραστικές λειτουργίες στο τρισδιάστατο αντικείμενο παρατηρεί ότι ο editor προσφέρει το tab MouseInteractions το οποίο σχετίζεται με διαδραστικές συμπεριφορές μεταξύ ποντικιού και 3D αντικειμένου. Στο tab παρουσιάζονται όλες οι δραστηριότητες που μπορεί να έχει το ποντίκι με το τρισδιάστατο αντικείμενο. Επίσης ο χρήστης παρατηρεί ότι πάλι μπορεί να εφαρμόσει και να ελέγξει μια δραστηριότητα σε πραγματικό χρόνο τσεκάροντας το κουμπί ενεργοποίηση ελέγχου. Στη συνέχεια ο χρήστης παρατηρεί ότι υπάρχει σχετικό tab για την εφαρμογή κίνησης(κυκλικής κυρίως) στο τρισδιάστατο αντικείμενο αλλά και την αλλαγή της εμφάνισης του αντικειμένου. Για την κίνηση ο χρήστης μπορεί να ενεργοποιήσει ή να απενεργοποιήσει την εφαρμογή κίνησης στο αντικείμενο και να ορίσει κάποιες τιμές που θα προσδιορίσουν πόσο χρόνο θα κρατήσει η κίνηση αυτή στις πόσες μοίρες θα περιστρέφεται το αντικείμενο τότε θα ξεκινήσει να περιστρέφεται και άλλες. Εδώ ο χρήστης έχει την ικανότητα με κάποια κουμπιά ελέγχου να ελέγξει σε πραγματικό χρόνο την κατάσταση της κίνησης του αντικειμένου(ο έλεγχος γίνεται με τα κουμπιά Start ,Pause ,Stop). Για την αλλαγή εμφάνισης του αντικειμένου ο χρήστης μπορεί να χρησιμοποιήσει το σχετικό tab και να εφαρμόσει νέα χρώματα στο αντικείμενο με την χρήση του επιλογέα χρωμάτων (Color Chooser), επίσης σε αυτό το tab ο χρήστης ενεργοποιεί ή απενεργοποιεί το φωτισμό σε ένα αντικείμενο. Στη συνέχεια ο χρήστης μπορεί να επιλέξει κατάλληλο φωτισμό με την χρήση του tab Light Attributes καθώς και να ορίσει τον προσανατολισμό και το χρώμα του φωτισμού με τα σχετικά εργαλεία που του παρέχονται. Κάθε αλλαγή χρωμάτων φωτισμού ή συντεταγμένων που κάνει ο χρήστης καταγράφεται στο παράθυρο των γενικών ιδιοτήτων του

τρισδιάστατου αντικειμένου. Τέλος έστω ότι ο χρήστης αποφασίζει ότι θέλει να εμπλουτίσει το αντικείμενο του με μία εικόνα , χρησιμοποιεί το tab Texture Image όπου μπορεί να επιλέξει ένα αρχείο εικόνας σε οποιοδήποτε φάκελο του υπολογιστή του με τον επιλογέα αρχείων.Επίσης σε αυτό το tab δίνεται η ικανότητα στο χρήστη να επιλέξει ένα από τα διαθέσιμα φλιταρίσματα Τέλος αν ο χρήστης θέλει να επεξεργαστεί περισσότερο την εικόνα μπορεί να χρησιμοποιήσει το tab Texture Attributes.

Αφού ο χρήστης ολοκληρώσει την επεξεργασία του αντικειμένου επιλέγει το κουμπί Source View και παρατηρεί ότι τα Tabs που σχετίζονται με την διαχείριση του τρισδιάστατου αντικειμένου στο frame κάτω δεξιά απενεργοποιήθηκαν και ενεργοποιείται το tab source output το οποίο σχετίζεται με πληροφορίες σχετικά με source κώδικα.Επίσης τώρα ο χρήστης μπορεί να διαβάσει το αρχείο Java που δημιουργήθηκε αυτόματα και αποθηκεύτηκε στο φάκελο εργασιών του.Ο χρήστης μπορεί μόνο να διαβάσει τον παραγόμενο κώδικα και όχι να τον μετατρέψει.Ο χρήστης μπορεί οποιαδήποτε στιγμή να επιλέξει το source view και να ελέγξει τον κώδικα του τρισδιάστατου αντικειμένου ανεξάρτητα αν έχει ολοκληρώσει την επεξεργασία.

Τέλος ο χρήστης παίρνει το module που δημιούργησε ο editor και το μετατρέπει ή το ενσωματώνει σε δικές του εφαρμογές.

#### **4.2.3. Καταγραφή απαιτήσεων JEditor3D1.1**

Η καταγραφή απαιτήσεων είναι μία από τις σημαντικότερες διεργασίες κατά την ανάπτυξη μιας εφαρμογής.Παρακάτω αναφέρονται ορισμένες απαιτήσεις που καταγράφηκαν κατά τον σχεδιασμό και ορισμένες που εμφανίστηκαν κατά την υλοποίηση.

- 1. Προσδιορισμός διαστάσεων αντικειμένου**  
Ο χρήστης θα έχει την ικανότητα να μετατρέψει τις διαστάσεις των τρισδιάστατων γεωμετρικών σχημάτων.
- 2. Προσδιορισμός τοποθεσίας και προσανατολισμού αντικειμένου**  
Ο Editor θα πρέπει να μπορεί να υποστηρίζει την επεξεργασία της θέσης και του προσανατολισμού του αντικειμένου.
- 3. Εμφάνιση δεδομένων πάνω στο σχήμα**  
Ο χρήστης θα έχει την ικανότητα να μπορεί να εμπλουτίσει το σχήμα με δεδομένα όπως εικόνα.
- 4. Μέθοδοι διάδρασης του αντικειμένου**  
Ο Editor θα μπορεί να εμπλουτίσει το τρισδιάστατο αντικείμενο με διαδραστικές μεθόδους όπως για παράδειγμα διαδραστική επικοινωνία μέσω του ποντικιού.
- 5. Επεξεργασία εμφάνισης του αντικειμένου**  
Ο Editor θα πρέπει να υποστηρίζει εργαλεία για την επεξεργασία της εμφάνισης του τρισδιάστατου αντικειμένου όπως αλλαγή χρωμάτων φωτισμού και αλλαγή χρώματος φόντου.
- 6. Εμπλουτισμός αντικειμένου με κίνηση**  
Ο Editor θα πρέπει να μπορεί να υποστηρίζει εργαλεία για να εμπλουτίσει το αντικείμενο με κίνηση.

7. **Παράθυρο για την σχεδίαση του αντικειμένου και εμφάνιση του κώδικα.**  
Ο Editor θα πρέπει να υποστηρίζει σε ένα παράθυρο την εναλλαγή μεταξύ Design View και Source View ώστε να είναι ποιο εύχρηστο στο χρήστη.
8. **Παράθυρο διαχείρισης αρχείων**  
Ο Editor θα πρέπει να υποστηρίζει την διαχείριση αρχείων με των οποίων θα δουλεύει.
9. **Συλλογή πληροφοριών αντικειμένου**  
Ο Editor θα μπορεί να συλλέγει όλες τις πληροφορίες που σχετίζονται με το τρισδιάστατο αντικείμενο και να τις εμφανίζει με σκοπό να ενημερώνεται ο χρήστης για την τρέχων κατάσταση του αντικειμένου.

#### 4.2.4. Η Διεπαφή JEditor3D1.1

Η διεπαφή μας χαρακτηρίζεται από την δημιουργία και επεξεργασία τρισδιάστατων αντικειμένων οπότε η διεπαφή που κατασκευάστηκε θα πρέπει να είναι εύχρηστη προς τον χρήστη και λειτουργική.

Παρακάτω παρουσιάζεται η διεπαφή με όλα τα λειτουργικά της μέρη για την ανάπτυξη και επεξεργασία τρισδιάστατων αντικειμένων.

- **Περίληπτική περιγραφή της διεπαφής**

Η διεπαφή αποτελείται από ένα Main frame το οποίο είναι χωρισμένο σε τέσσερα τμήματα από τρεις διαφορετικούς Sliders. Το πρώτο τμήμα είναι υπεύθυνο για την προβολή και διαχείριση των αρχείων που πρόκειται να δημιουργήσει ο χρήστης. Το δεύτερο τμήμα είναι το κύριο τμήμα της απόδοσης του τρισδιάστατου αντικειμένου το οποίο προβάλλει το τρισδιάστατο αντικείμενο στον χρήστη. Το τμήμα αυτό μπορεί επίσης να φιλοξενήσει και αρχεία που περιέχουν κώδικα προς επεξεργασία ή ανάγνωση. Δηλαδή είναι το κύριο τμήμα στο οποίο ο χρήστης μπορεί να παρακολουθήσει το τρισδιάστατο αντικείμενο αλλά και να γράψει κώδικα στον παρεχόμενο editor.

Το τρίτο τμήμα περιέχει έναν πίνακα ο οποίος συλλέγει πληροφορίες σχετικά με την τρέχον κατάσταση του αντικειμένου. Οι πληροφορίες αυτές ομαδοποιούνται και εμφανίζονται με την σειρά για την καλύτερη πληροφόρηση του χρήστη.

Τέλος το τέταρτο τμήμα περιέχει όλα τα εργαλεία για την επεξεργασία και ανάπτυξη του τρισδιάστατου αντικειμένου.

Επίσης η διεπαφή αποτελείται και από μία γραμμή εργαλείων για την γρηγορότερη προσπέλαση ενεργειών από το χρήστη και από το σύνθηρες εμφανιζόμενο μενού.

Στο παρακάτω σχήμα παρουσιάζεται η διεπαφή στην αρχική της κατάσταση.

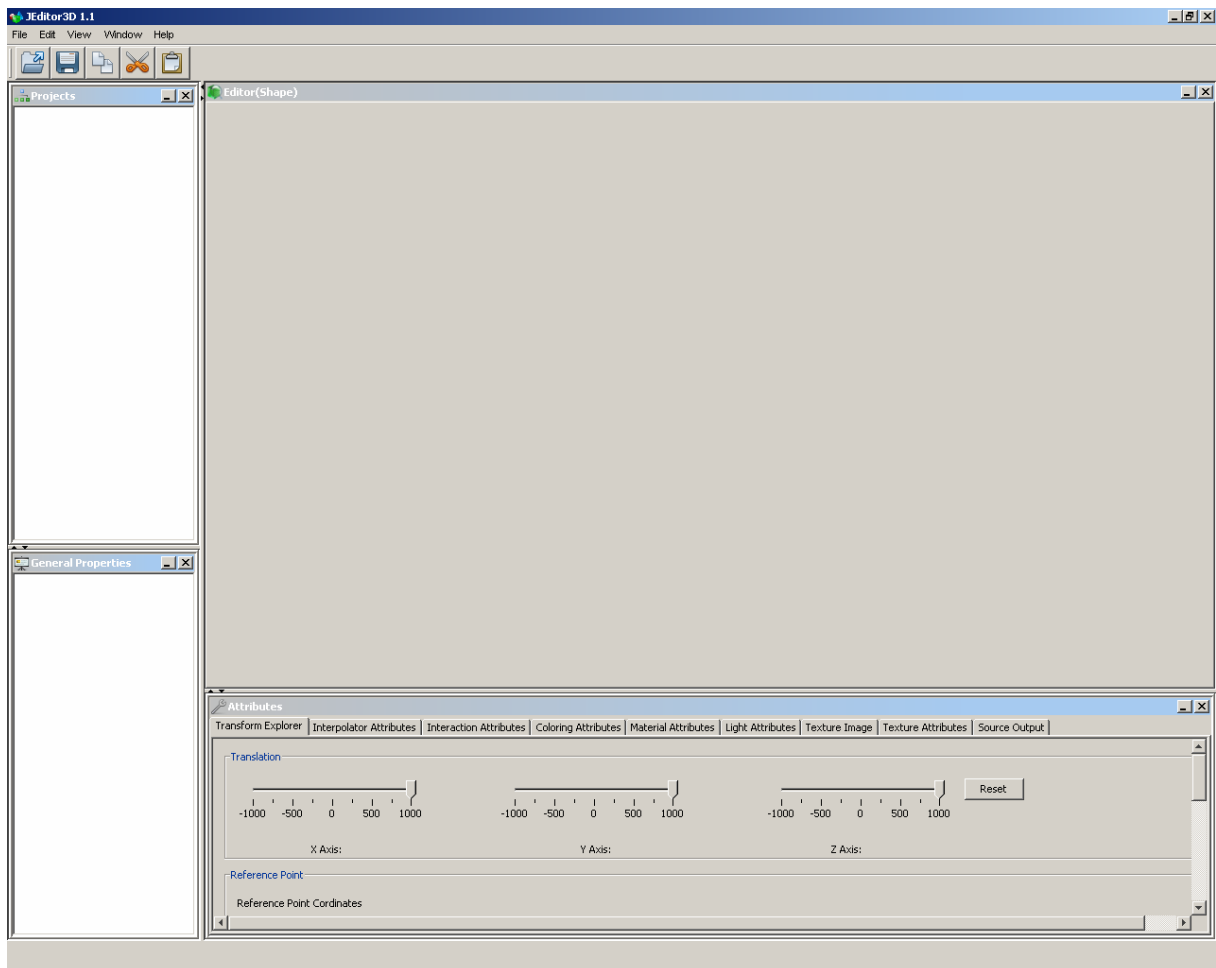


Figure 17: Αρχική κατάσταση JEditor3D 1.1

Όπως φαίνεται και από το σχήμα το πρώτο τμήμα ονομάζεται Projects και σχετίζεται με την παρουσίαση και διαχείριση αρχείων, το δεύτερο τμήμα ονομάζεται Editor(στην παρένθεση μπαίνει το mode που έχει επιλέξει ο χρήστης πχ Source-Shape) το οποίο είναι το κυρίως παράθυρο ανάπτυξης του αντικειμένου. Το τρίτο τμήμα ονομάζεται γενικές ιδιότητες(General Properties) και σχετίζεται με την συλλογή και εμφάνιση πληροφοριών του αντικειμένου. Το τέταρτο τμήμα ονομάζεται Attributes (Χαρακτηριστικά) και σχετίζεται με τα εργαλεία ανάπτυξης των αντικειμένων.

Τέλος η γραμμή εργαλείων και το μενού.

## 1. Μενού

Το μενού στην διεπαφή μας αποτελείται από το File , Edit , View , Window , Help ,About.

- **File**

Από τη επιλογή του μενού επιλογών File μπορούμε να ανοίξουμε ένα υπάρχον project με την επιλογή open , μπορούμε να αποθηκεύσουμε τις αλλαγές που έχουμε κάνει με την επιλογή Save , μπορούμε να κλείσουμε τον editor με την επιλογή exit και τέλος μπορούμε να δημιουργήσουμε ένα καινούργιο project με την επιλογή New – Project.

Παρακάτω παρουσιάζεται μια εικόνα με τις επιλογές του μενού επιλογής File.

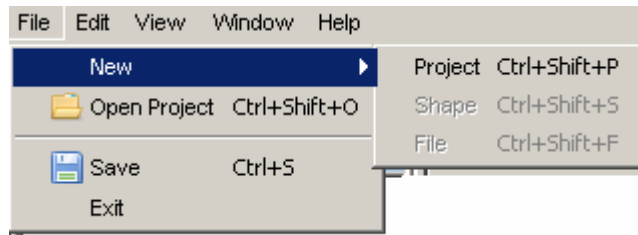


Figure 18: Μενού επιλογής File

- **Window**

Από την επιλογή Window ο editor δίνει την δυνατότητα στο χρήστη να επαναφέρει τα τέσσερα παράθυρα που αναφέραμε παραπάνω στον editor. Συνήθως η επιλογή αυτή χρησιμοποιείται όταν ο χρήστης έχει κλείσει κάποια παράθυρα για να δώσει έμφαση σε κάποιο άλλο παράθυρο ή γιατί την τρέχουσα στιγμή δουλεύει λιγότερο με κάποια παράθυρα και δεν θέλει να του πιάνουν χώρο.

Παρακάτω παρουσιάζεται μια εικόνα με τις επιλογές του μενού επιλογής Window.

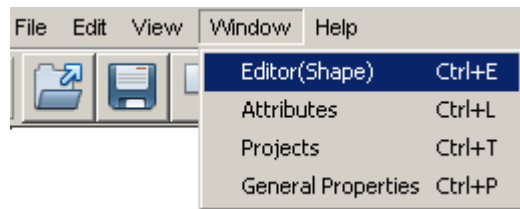


Figure 19: Μενού επιλογής Window

- **Help**

Από την επιλογή Help - about ο editor εμφανίζει μια εικόνα με το χαρακτηριστικό γνώρισμα της εφαρμογής μας.

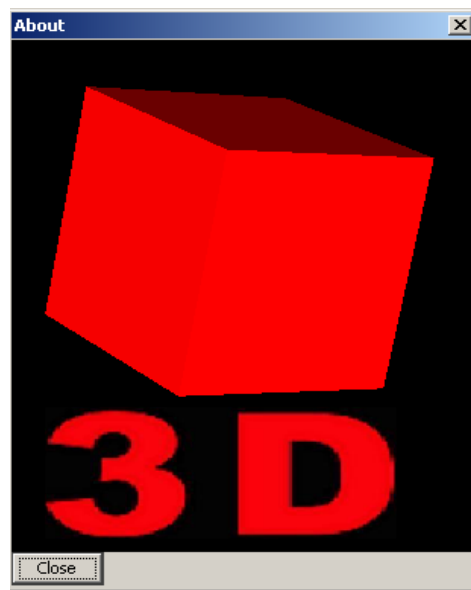


Figure 20: Μενού επιλογής About

Λειτουργίες για τις υπόλοιπες επιλογές δεν έχουν υλοποιηθεί ακόμα. Αλλά έχουν προστεθεί για μελλοντική χρήση.



## 2. Projects Frame

Το παράθυρο αυτό αποτελείται από ένα `JInternalFrame` το οποίο έχει τοποθετηθεί μέσα σε ένα `DesktopPane`. Εδώ όπως και στα άλλα τμήματα χρησιμοποιήσαμε ένα `JInternalFrame` για την καλύτερη διαχείριση των συστατικών που θα τοποθετήσουμε μέσα σε αυτό.

Σκοπός του παραθύρου αυτού είναι να είναι υπεύθυνο για την εμφάνιση και διαχείριση των αρχείων του project που δημιουργεί ο χρήστης. Τα αρχεία αυτά έχουν δημιουργηθεί και αποθηκευτεί στον φάκελο εργασίας του χρήστη.

Κύριο συστατικό του παραθύρου αυτού είναι ότι αποτελείται από το συστατικό `JTree` το οποίο εμφανίζει τα δεδομένα σε δομή δένδρου. Το δένδρο υποστηρίζει διαφορετικά icons σχετικά με τον τύπο του αρχείου. Επίσης υποστηρίζει λειτουργίες όπως `delete` και `paste` (για εικόνες μόνο). Επίσης δίνει την δυνατότητα εμφάνισης popup για την δημιουργία και διαγραφή αρχείων, φακέλων

Παρακάτω παρουσιάζεται ένα project το οποίο εμφανίζεται με την μορφή δένδρου.

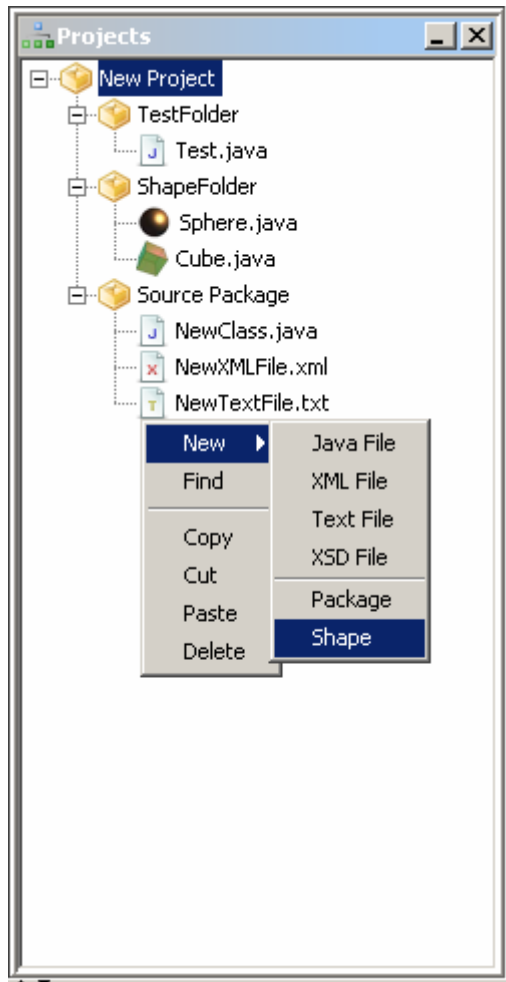


Figure 21: Projects Frame structure

### 3. Editor Shape Frame

Το Frame αυτό αποτελείται από ένα JInternalFrame το οποίο έχει εμπλουτιστεί με ένα JTabbedPane το οποίο είναι υπεύθυνο για την εμφάνιση του περιεχομένου των αρχείων που δουλεύει ο χρήστης. Κάθε αρχείο που καλείται από το χρήστη να ανοίξει τοποθετείται σε αυτό το frame με την μορφή ενός tab, ο χρήστης έχει την δυνατότητα να μετακινηθεί από το ένα tab στο άλλο καθώς και να κλείσει το κάθε tab μεμονωμένα ή όλα μαζί.

Το περιεχόμενο των αρχείων μπορεί να είναι από ένας απλός κώδικας Java μέχρι ένα τρισδιάστατο αντικείμενο όπου σε αυτήν την περίπτωση ο χρήστης έχει την ικανότητα να επιλέξει μεταξύ Design View και Source View. Όταν το αρχείο σχετίζεται με την ανάπτυξη τρισδιάστατου αντικείμενου το Frame παρέχει δύο επίπεδα επεξεργασίας το Design View και το Source View.

Αν το αρχείο που εμφανίζεται στο tab είναι ένα απλό αρχείο κειμένου τότε εμφανίζεται σε έναν editor κειμένου με την επιλογή Source View ενεργοποιημένη.

Αν το αρχείο αποτελεί ένα τρισδιάστατο αντικείμενο τότε οι επιλογές Source και Design View είναι διαθέσιμες για να επιλέξει ο χρήστης. Αν θέλει να επεξεργαστεί το τρισδιάστατο αντικείμενο τότε θα πρέπει να πατήσει Design View ώστε το αντικείμενο να εμφανιστεί στην οθόνη και να ενεργοποιηθούν τα κατάλληλα εργαλεία επεξεργασίας. Αν θέλει να διαβάσει τον παραγόμενο κώδικα του αντικείμενου αυτού τότε ενεργοποιεί το Source View. Οι εναλλαγές μεταξύ Design και Source View μπορούν να γίνουν οποιαδήποτε στιγμή.

Παρακάτω παρουσιάζεται το σχετικό frame.

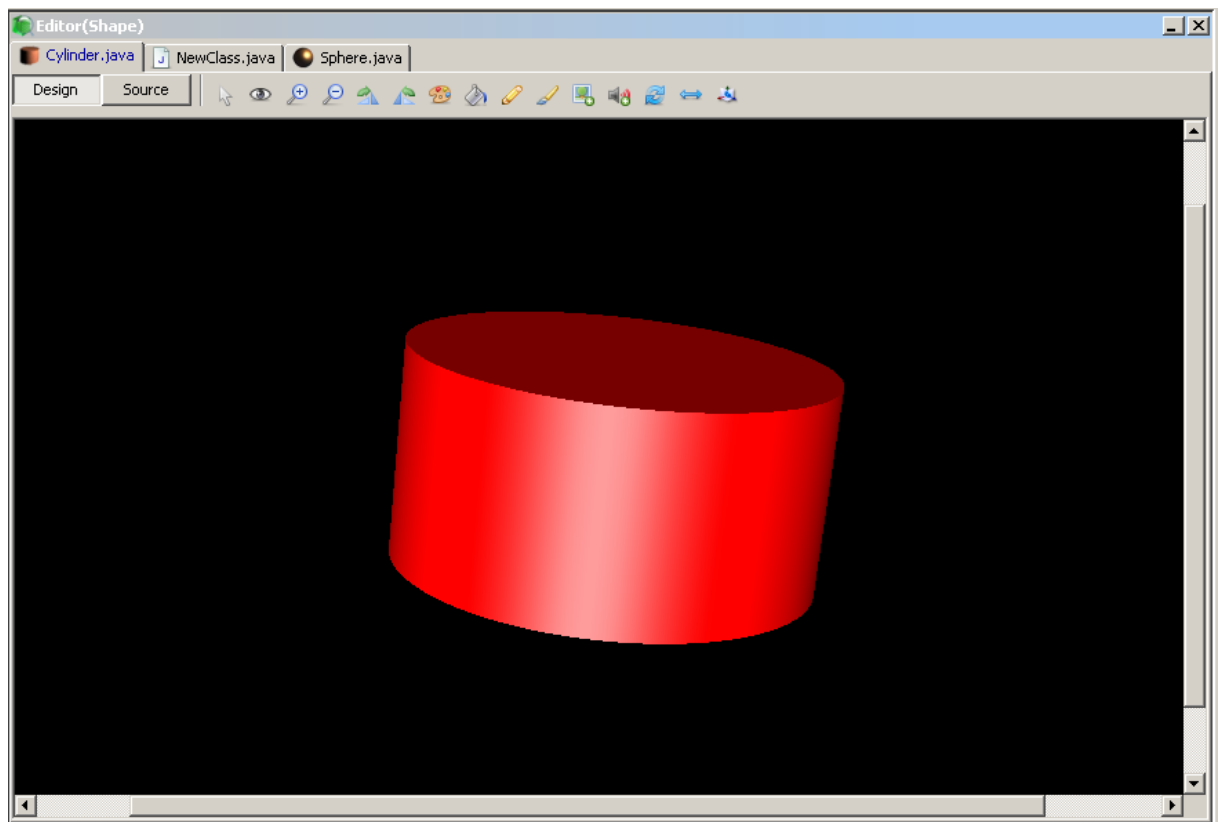


Figure 22: Design View for 3D object

The screenshot shows an IDE window titled "Editor(Source)" with tabs for "Cylinder.java", "NewClass.java", and "Sphere.java". The "Source" view is active, displaying the following Java code for the Cylinder class:

```

36
37
38 public Cylinder() throws Exception {
39     initComponents();
40 }
41
42
43
44
45 /*
46  * Initializes calss components
47  */
48 private void initComponents() throws Exception {
49     GraphicsConfiguration config = SimpleUniverse.getPreferredConfiguration();
50
51     moduleCanvas = new Canvas3D(config);
52     moduleCanvas.setBackground(Color.BLACK);
53
54     moduleUniverse = new SimpleUniverse(moduleCanvas);
55     moduleView = moduleUniverse.getViewer().getView();
56     moduleView.setFrontClipDistance(0.01f);
57
58
59     viewingPlatform = moduleUniverse.getViewingPlatform();
60     viewingPlatform.setNominalViewingTransform();
61
62     moduleBranchGroup = new BranchGroup();
63     moduleBranchGroup.setCapability(BranchGroup.ALLOW_DETACH);
64
65
66     //Initialize Coloring Attributes
67     coloringAttributes = new ColoringAttributes();
68     coloringAttributes.setColor(new Color3f(1.0f, 0.0f, 0.0f));
69     coloringAttributes.setShadeModel(3);
70
71     //Initialize Material Attributes

```

Figure 23: Source View for 3D object

The screenshot shows an IDE window titled "Editor(Shape)" with tabs for "Cylinder.java", "NewClass.java", and "Sphere.java". The "Source" view is active, displaying the following Java code for the NewClass class:

```

1  /*
2  * This Code is auto generated.
3  *
4  */
5
6
7
8
9 public class NewClass {
10
11     public NewClass() throws Exception {
12         initComponents();
13     }
14
15
16
17     /*
18     * Initializes calss components
19     *
20     */
21     private void initComponents() throws Exception {
22         JButton button = new JButton();
23
24         int k=0;
25         for(int i=0; i < |max|; i++) {
26             }
27
28     }
29
30 }

```

Figure 24: Source for Java file

Επιπρόσθετα εκτός από τις δύο κύριες επιλογές Design και Source κάθε tab περιέχει ένα σύνολο εργαλείων το οποίο βοηθά το χρήστη στην επεξεργασία των αρχείων του Παρακάτω εμφανίζονται τα εργαλεία αυτά.

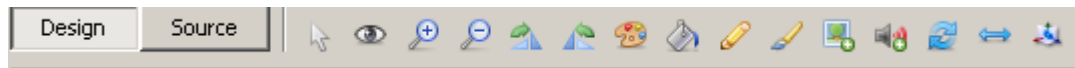






Figure 25:Design Tools



Figure 26:Source Tools

Στο Design View τα εργαλεία που χρησιμοποιούνται είναι για την ενεργοποίηση των συντεταγμένων του χώρου σε XYZ συντεταγμένες, για την ενεργοποίηση του τόξου προσανατολισμού που ορίζει σε ποια κατεύθυνση – κλίση θα ορίζεται το σχήμα, ο image chooser για την επιλογή και πρόσθεση εικόνας στο αντικείμενο και το refresh κουμπί που τοποθετεί το σχήμα στην προγραμματισμένη και προσδιορισμένη θέση του στον χώρο.

-  κουμπί για την ενεργοποίηση των συντεταγμένων XYZ.
-  κουμπί για την ενεργοποίηση του άξονα προσανατολισμού-περιστροφής.
-  κουμπί για την εισαγωγή εικόνας στο τρισδιάστατο αντικείμενο.
-  κουμπί για την επαναφορά του αντικειμένου στην αρχική του θέση.

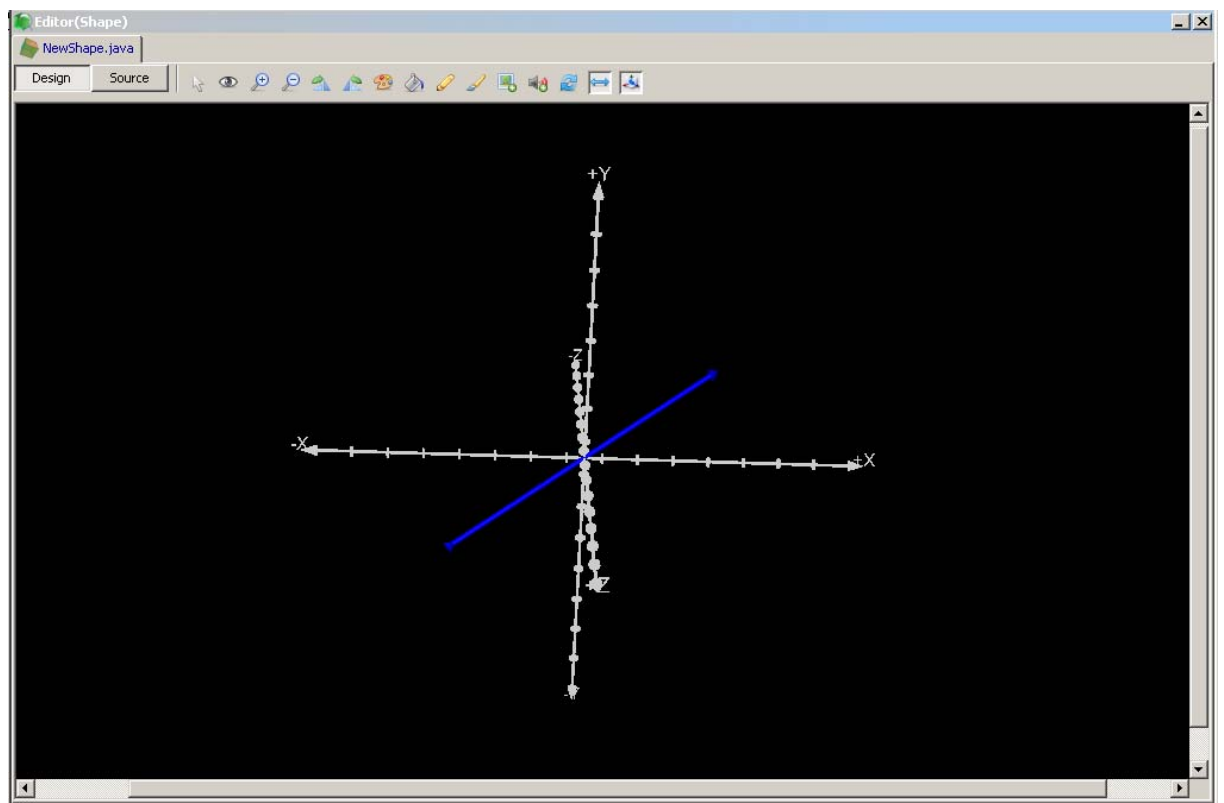


Figure 27:Cordinates Axis and Rotation Axis

Λειτουργίες για τα υπόλοιπα κουμπιά δεν έχουν υλοποιηθεί ακόμα.Αλλα έχουν προστεθεί για μελλοντική χρήση.

#### **4. General Properties Frame**

Το τμήμα αυτό της εφαρμογής αποτελείται από ένα JInternal Frame στο οποίο έχουμε τοποθετήσει έναν πίνακα τύπου JTable ο οποίος είναι υπεύθυνος για την συλλογή και εμφάνιση πληροφοριών που σχετίζονται με το τρισδιάστατο αντικείμενο. Ο πίνακας αποτελείται από δύο στήλες και από αυξηνόμενες δυναμικές γραμμές.Η πρώτη στήλη του πίνακα ορίζει το όνομα της μεταβλητής – παραμέτρου που μεταβάλλεται και ο δεύτερος ορίζει την τιμή που έχει η παράμετρος αυτή κάθε φορά.

Ο πίνακας αυτός ομαδοποιεί τις πληροφορίες ανάλογα με την πηγή προέλευσης τους.Για παράδειγμα κατά την επεξεργασία των διαστάσεων και της θέσης του αντικειμένου ο πίνακας αντιλαμβάνεται ότι οι πληροφορίες αυτές προέρχονται από το Transform Explorer tab και δημιουργεί μία επικεφαλίδα με τίτλο την πηγή προέλευσης τους εμφανίζοντας από κάτω το όνομα της παραμέτρου - ιδιότητας που μεταβλήθηκε και την τιμή που πήρε.

Ο πίνακας αυτός δημιουργήθηκε με σκοπό να ενημερώνει τον χρήστη για την τρέχον κατάσταση του αντικειμένου κάθε φορά ανεξάρτητα σε ποιο κομμάτι του αντικειμένου δουλεύει ο χρήστης.

Ο πίνακας αυτός ενεργοποιείται μόνο για αρχεία που σχετίζονται με την ανάπτυξη και επεξεργασία τρισδιάστατων αντικειμένων και όχι για αρχεία ανάπτυξης κώδικα ή κειμένου όπως απλά αρχεία Java,xml,txt.

Παρακάτω παρουσιάζεται ο πίνακας αυτός

Property	Value
<b>Transform Explorer</b>	
Scaling	(1.0, 1.0, 1.0)
Rotation	(1.0, 0.0, 0.0, 0.0)
Translation	(0.0, 0.0, 0.0)
Reference Point	(0.0, 0.0, 0.0)
<b>Interpolator</b>	
Loop Count	-1
Trigger Time	0
PhaseDelayDuration	0
Alpha At One Duration	0
Alpha At Zero Duration	0
Decreasing Alpha Duration	0
Decreasing Alpha Rump Duration	0
Increasing Alpha Duration	4000
Increasing Alpha Rump Duration	0
Rotator Enable	false
Maximum Angle	360.0
Minimum Angle	0.0
Alpha Mode	INCREASING_ENABLE
<b>Mouse Interaction</b>	
Translate Enable	false
Translate X Factor	0.02
Translate Y Factor	0.02
Rotate Enable	false
Rotate X Factor	0.03
Rotate Y Factor	0.03
Zoom Enable	false
Zoom Factor	0.04
Wheel Zoom Enable	false
Wheel Zoom Factor	0.1
<b>Coloring Attributes</b>	
Shape Color	(1.0, 0.0, 0.0)
Background Color	(0.0, 0.0, 0.0)
Shade Model	SHADE_GOURAND
<b>Material Attributes</b>	
Ambient Color	(1.0, 0.0, 0.0)
Specular Color	(1.0, 1.0, 1.0)
Diffuse Color	(1.0, 0.0, 0.0)
Emissive Color	(0.0, 0.0, 0.0)
Shininess	20.0
Light Enabled	false
<b>Light Attributes</b>	
Light Category	Ambient Light
Ambient Light Color	(0.3019608, 0.3019608, 0.3019608)
Directional Light Color	(1.0, 1.0, 1.0)
Directional Direction	(0.0, 0.0, -0.1)
Point Light Color	(1.0, 1.0, 1.0)
Point Position	(-4.0, 8.0, 16.0)
Point Attenuation	(1.0, 0.0, 0.0)
<b>Texture Image</b>	
Boundary S Mode	WARP
Boundary T Mode	WARP
Boundary Color	(0.0, 0.0, 0.0, 1.0)
Magnification Filter	Base Level Point
Minification Filter	Base Level Point

Figure 28: Γενικές Ιδιότητες (General Properties)

Όπως παρατηρούμε και από την εικόνα υπάρχουν επικεφαλίδες που χρωματίζονται με ανοικτό γκρι οι οποίες δηλώνουν ότι οι παράμετροι που ακολουθούν προέρχονται από την επεξεργασία του συγκεκριμένου τομέα του τρισδιάστατου αντικειμένου. Για κάθε αλλαγή που κάνει ο χρήστης στο τρισδιάστατο αντικείμενο ενημερώνεται και ο πίνακας Γενικών Ιδιοτήτων σε πραγματικό χρόνο. Έτσι ο χρήστης έχει την ευχέρεια κάθε φορά που ελέγχει τον αυτόματα παραγόμενο κώδικα να ελέγχει τις τιμές των παραμέτρων μέσω του πίνακα. Ο πίνακας αυτός είναι αρκετά συνδεδεμένος με το frame Attributes όπου το Frame αυτό είναι υπεύθυνο για την επεξεργασία και μεταβολή των παραμέτρων του τρισδιάστατου αντικειμένου.

## 5. Attributes Frame

Το frame αυτό αποτελείται από ένα JInternalFrame στο οποίο έχει τοποθετηθεί ένα συστατικό τύπου JTabbedPane. Το Frame αυτό παρέχει στο χρήστη εργαλεία για την μεταβολή και επεξεργασία των τρισδιάστατων αντικειμένων. Το Frame αυτό αποτελείται από 9 tabs εκ των οποίων τα 8 πρώτα σχετίζονται με τα εργαλεία ανάπτυξης τρισδιάστατου αντικειμένου ενώ το τελευταίο έχει σχεδιαστεί για να παράγει πληροφορίες σχετικά με το αρχείο που περιέχουν κώδικα ή κείμενο. Παρακάτω αναφέρονται ονομαστικά τα εργαλεία αυτά.

- **Transform Explorer**

Το εργαλείο αυτό είναι υπεύθυνο για την μεταβολή της θέσης, του προσανατολισμού, του σημείου αναφοράς και τις κλίμακας μεγέθους του αντικειμένου. Για την επίτευξη των παραπάνω έχουν οριστεί διάφοροι ράβδοι κύλισης η οποίοι παρέχουν μεταβολή των παραπάνω παραμέτρων σε πραγματικό χρόνο χωρίς ο χρήστης να αναγκαστεί να πατήσει OK ή Apply μετά από κάθε αλλαγή.

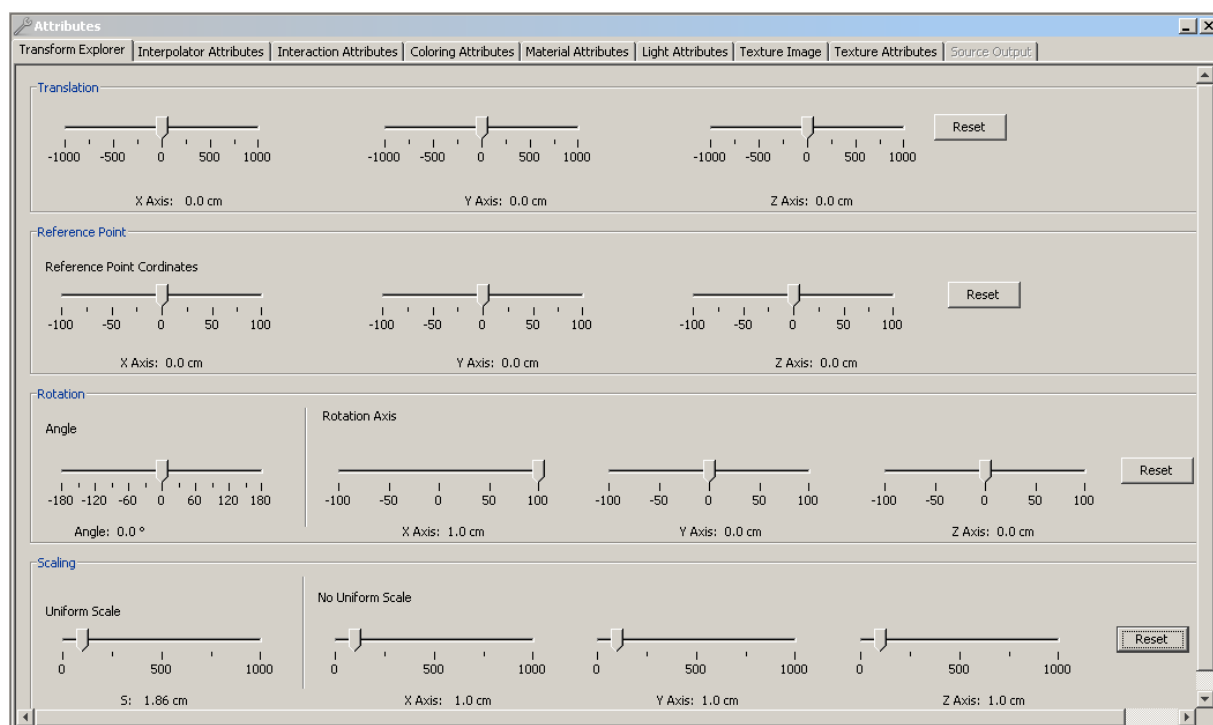


Figure 29 : Transform Explorer Attribute

### **Θέση αντικειμένου (Translation)**

Η μεταβολή της θέσης του αντικειμένου πραγματοποιείται σε συντεταγμένες XYZ και για το σκοπό αυτό ο editor παρέχει στο χρήστη τρεις ράβδους κύλισης μία για κάθε συντεταγμένη. Η μέγιστη τιμή που μπορεί να πάρει η ράβδος είναι 1000 ενώ η ελάχιστη είναι -1000. Κάθε φορά που μεταβάλλεται μία από τις τιμές των συντεταγμένων η τιμή της μεταβολής που παράγεται από την ράβδο διαιρείται με το 100 ώστε η μέγιστη τιμή που μπορεί να πάρει το αντικείμενο στο χώρο είναι από -10cm μέχρι +10 cm. Η μέγιστη και η ελάχιστη τιμή οριστήκανε στα 10 , -10 κατόπιν έρευνας και πολλαπλών ελέγχων(tests).Επίσης ένα κουμπί Reset έχει προστεθεί για την επαναφορά των τιμών των παραμέτρων στην αρχική τιμή τους.Παρακάτω επισυνάπτεται η θέση ενός κύβου(Figure 30) κατόπιν μεταβολής των συντεταγμένων θέσης(Figure 31)

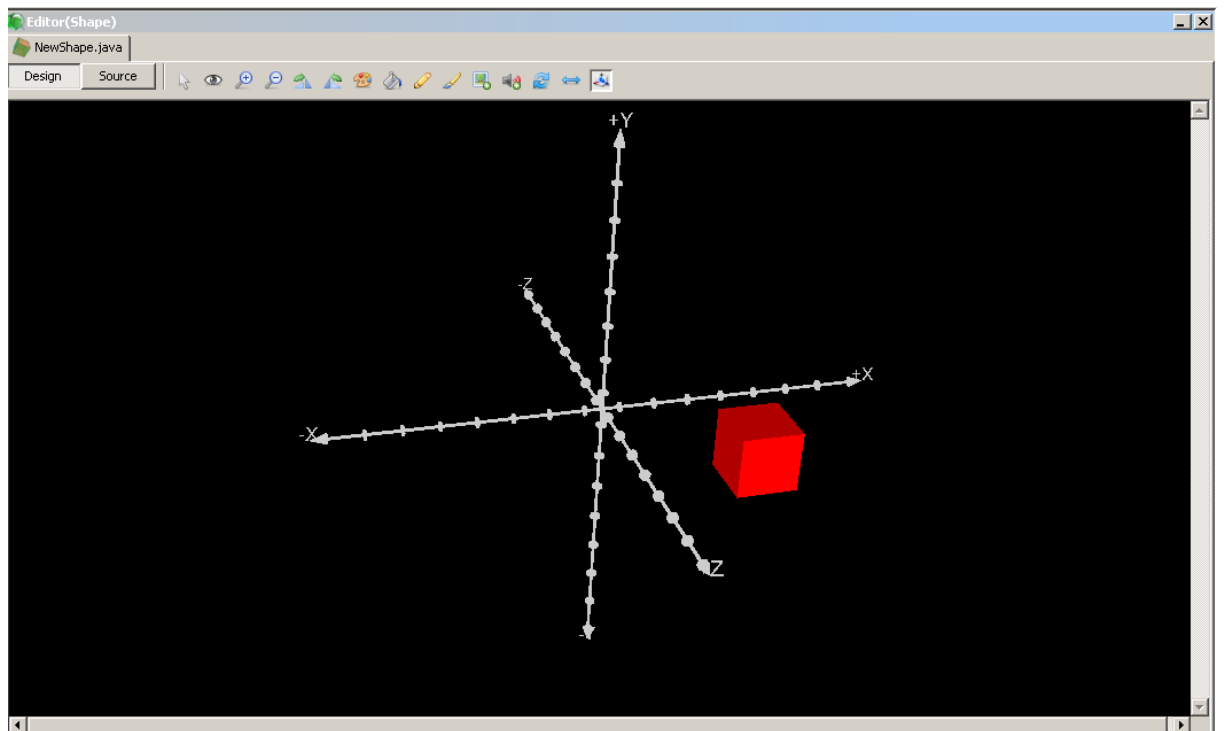


Figure 30: Shape's position after translation

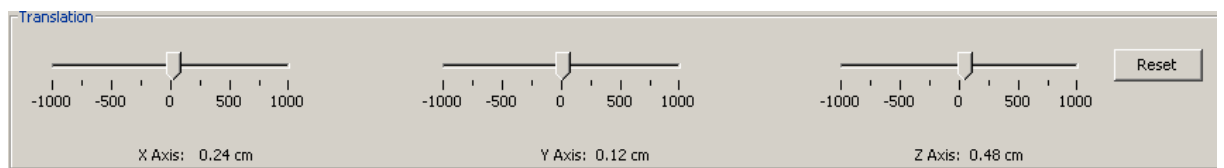


Figure 31: Translation Sliders

### **Σημείο αναφοράς αντικειμένου(Reference Point)**

Το εργαλείο αυτό αποτελείται από τρεις ράβδους κύλισης και σχετίζεται με την μετατροπή του σημείου αναφοράς του αντικειμένου. Οι μέγιστες τιμές της ράβδους κύλισης είναι από -10,10. Κάθε φορά που μεταβάλλεται μία από τις τιμές των συντεταγμένων η τιμή της μεταβολής που παράγεται από την ράβδο διαιρείται με το 10 ώστε η μέγιστη τιμή του σημείου αναφοράς του τρισδιάστατου αντικειμένου να είναι από -1cm έως +1 cm. Το σημείο αναφοράς παίζει χαρακτηριστικό ρόλο στην



περιστροφική κίνηση του αντικειμένου καθώς το αντικείμενο δεν περιστρέφεται γύρω από την αρχική του θέση που είναι  $(0,0,0)$  αλλά γύρω από την θέση που έχουμε ορίσει με τους ράβδους κύλισης. Ένα σημαντικό εργαλείο που μας βοηθάει να ελέγξουμε οπτικά την μεταβολή του σημείου αναφοράς του αντικειμένου είναι ο άξονας προσανατολισμού-περιστροφής που αναφέραμε παραπάνω. Παρακάτω επισυνάπτεται μια εικόνα με την μεταβολή του σημείου αναφοράς του αντικειμένου.

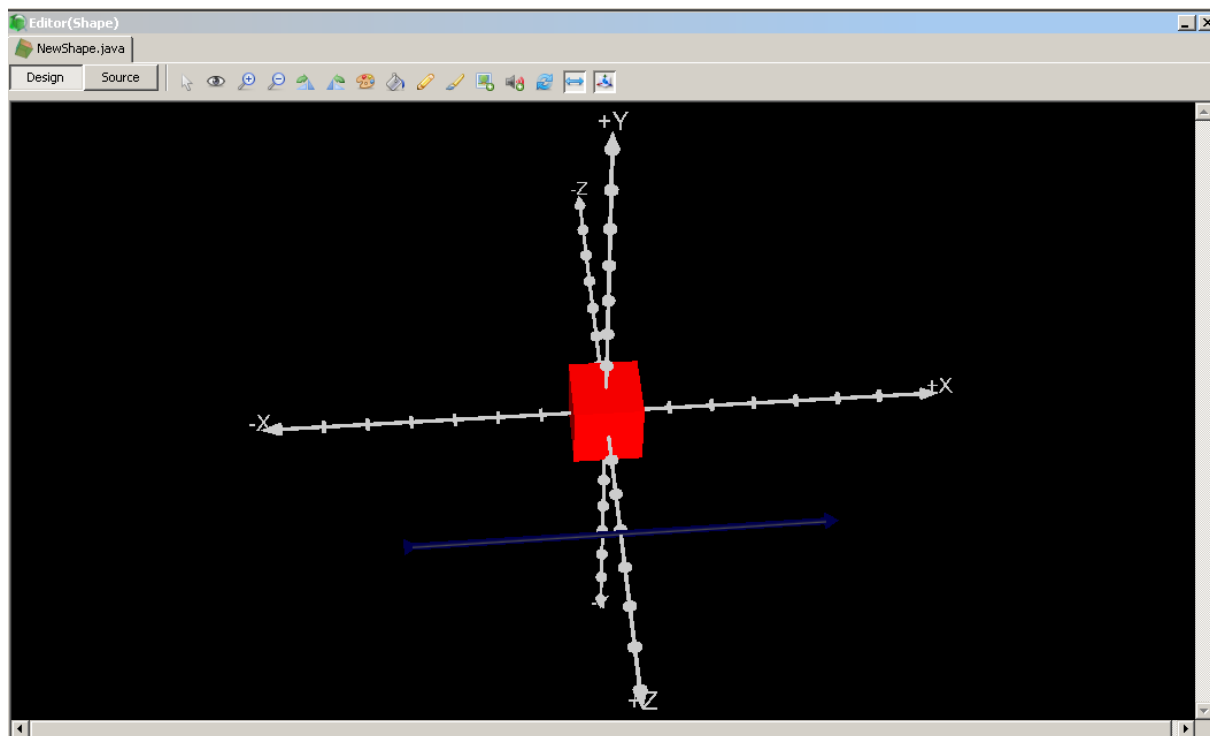


Figure 32:Μεταβολή άξονα προσανατολισμού - περιστροφής

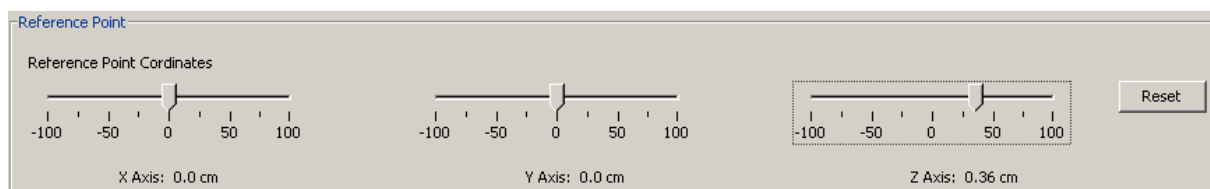


Figure 33: Ράβδοι σημείου αναφοράς του αντικειμένου

### **Περιστροφή(Rotation)**

Το εργαλείο αυτό αποτελείται από τέσσερις ράβδους κύλισης η πρώτη ράβδος ονομάζεται Angle και σχετίζεται με την μεταβολή της γωνίας περιστροφής του αντικειμένου. Έχει σαν μέγιστες τιμές από -180 μέχρι +180 μοίρες. Οι υπόλοιπες τρεις σχετίζονται με τις συντεταγμένες του άξονα περιστροφής του τρισδιάστατου αντικειμένου. Η μέγιστη τιμή των ράβδων αυτών είναι 100 και η ελάχιστη -100. Κάθε φορά που μεταβάλλεται μία από τις τιμές των συντεταγμένων η τιμή τις μεταβολής που παράγεται από την ράβδο διαιρείται με το 100 έτσι ώστε η μέγιστη τιμή και η ελάχιστη τιμή που μπορεί να πάρει ο άξονας περιστροφής να είναι +1 και -1 αντίστοιχα. Για παράδειγμα αν οι συντεταγμένες του άξονα περιστροφής είναι  $(1,0,0)$  τότε η περιστροφή πραγματοποιείται στον +X άξονα και από 0 μέχρι 180 μοίρες η περιστροφή γίνεται δεξιόστροφα. Αν ο χρήστης θέλει να περιστρέφεται το αντικείμενο

ως προς τον +Y τότε θα πρέπει να ορίσει τις συντεταγμένες ως εξής (0,1,0). Τέλος και σε αυτό το εργαλείο υπάρχει ένα reset κουμπί που επαναφέρει τις αρχικές ρυθμίσεις. Παρακάτω επισυνάπτεται μια εικόνα του άξονα περιστροφής και του αντικειμένου κατόπιν μεταβολής της γωνίας περιστροφής.

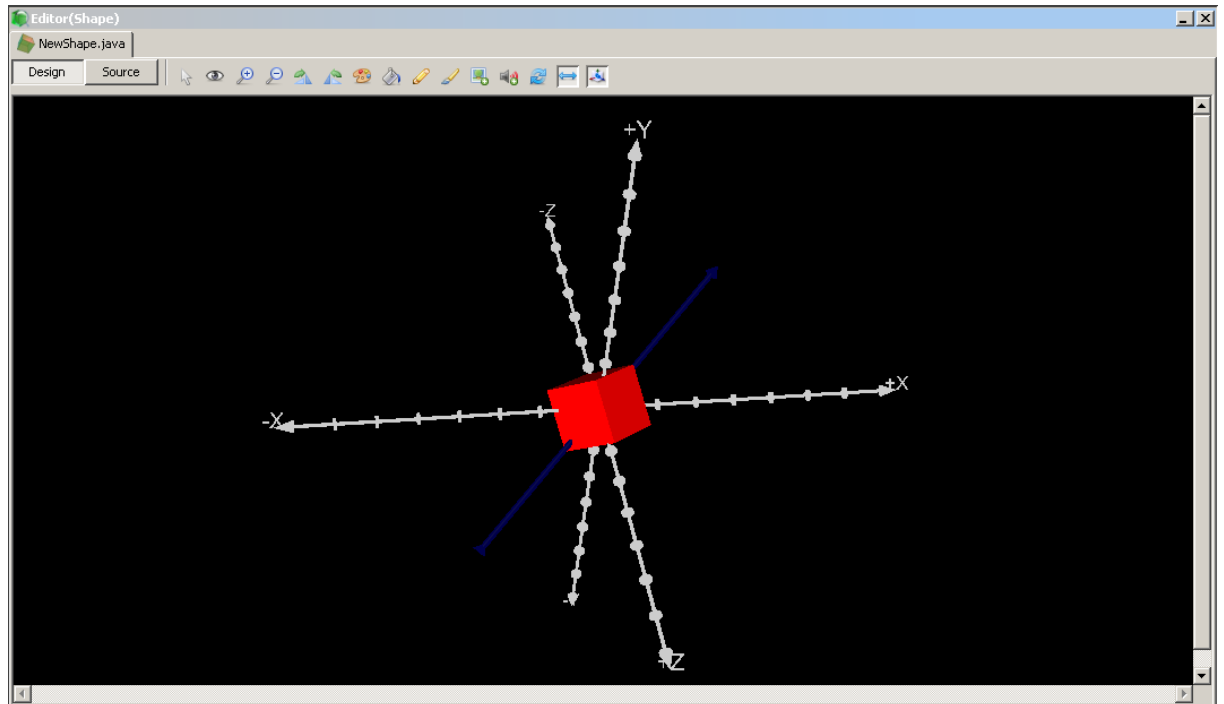


Figure 34: Άξονας περιστροφής

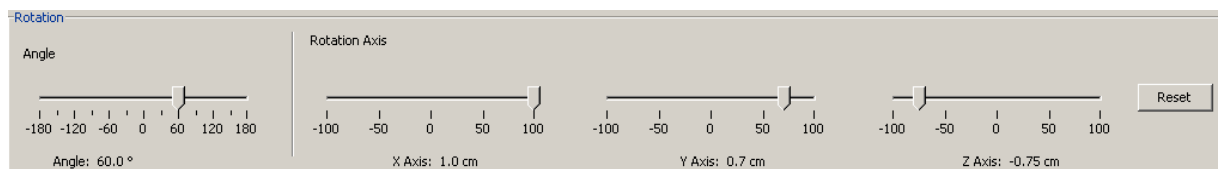


Figure 35: Ράβδοι περιστροφής του άξονα και της γωνίας του αντικειμένου

### **Κλίμακα(Scaling)**

Το εργαλείο αυτό αποτελείται από τέσσερις ράβδους οι οποίες μπορούν να μεταβάλουν την κλίμακα του αντικειμένου. Πολλές φορές η μεταβολή της κλίμακας του αντικειμένου θεωρείται σαν μεταβολή του μεγέθους του αντικειμένου γιατί το API της Java3D δεν σου παρέχει την δυνατότητα να μεταβάλλεις το μέγεθος του αντικειμένου μετά την δημιουργία του. Έτσι όλοι οι προγραμματιστές της Java3D όταν θέλουν να μεταβάλουν το μέγεθος του αντικειμένου μετά την δημιουργία του μεταβάλουν την κλίμακά του.

Η πρώτη ράβδος του εργαλείου αυτού που ονομάζεται Uniform Scale χρησιμοποιείται για να μεταβάλει το μέγεθος του αντικειμένου ισόποσα σε όλες τις συντεταγμένες (X,Y,Z). Οι επόμενες τρεις ράβδοι μεταβάλουν το μέγεθος του αντικειμένου ξεχωριστά για κάθε άξονα. Τέλος και σε αυτό το εργαλείο υπάρχει ένα reset κουμπί που επαναφέρει τις αρχικές ρυθμίσεις.

Παρακάτω επισυνάπτονται δυο εικόνες μεταβολής του μεγέθους του αντικειμένου.

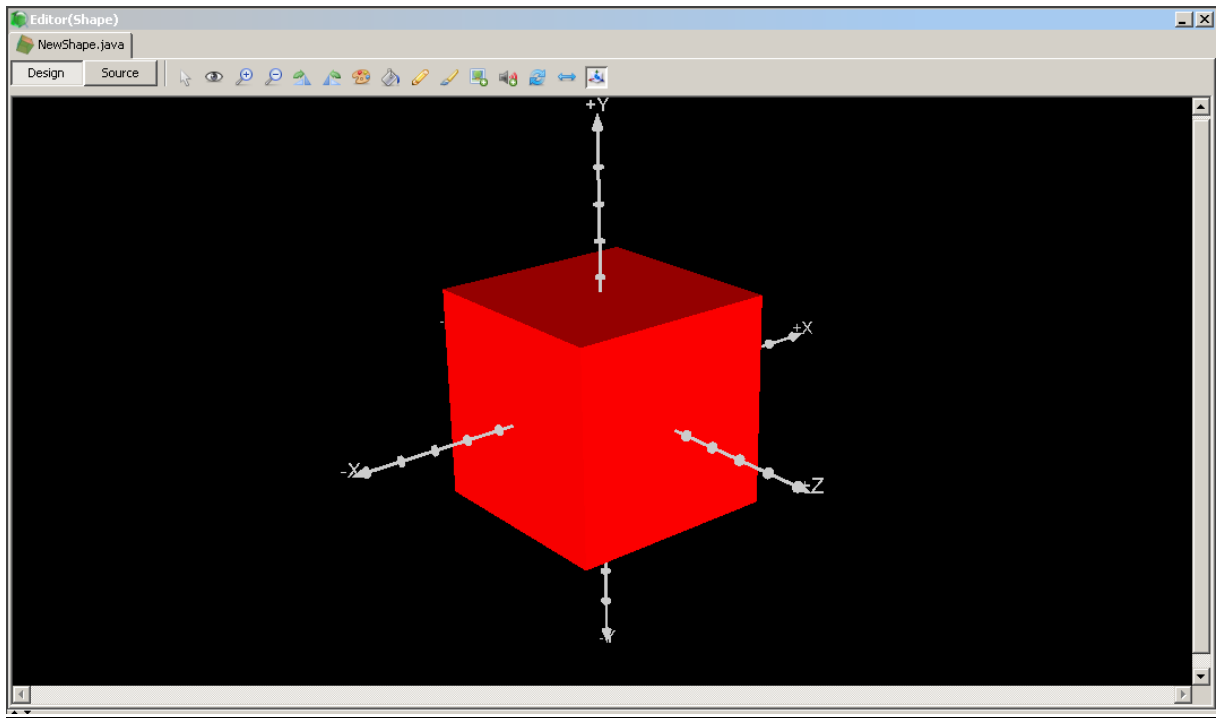


Figure 36: Uniform Scale

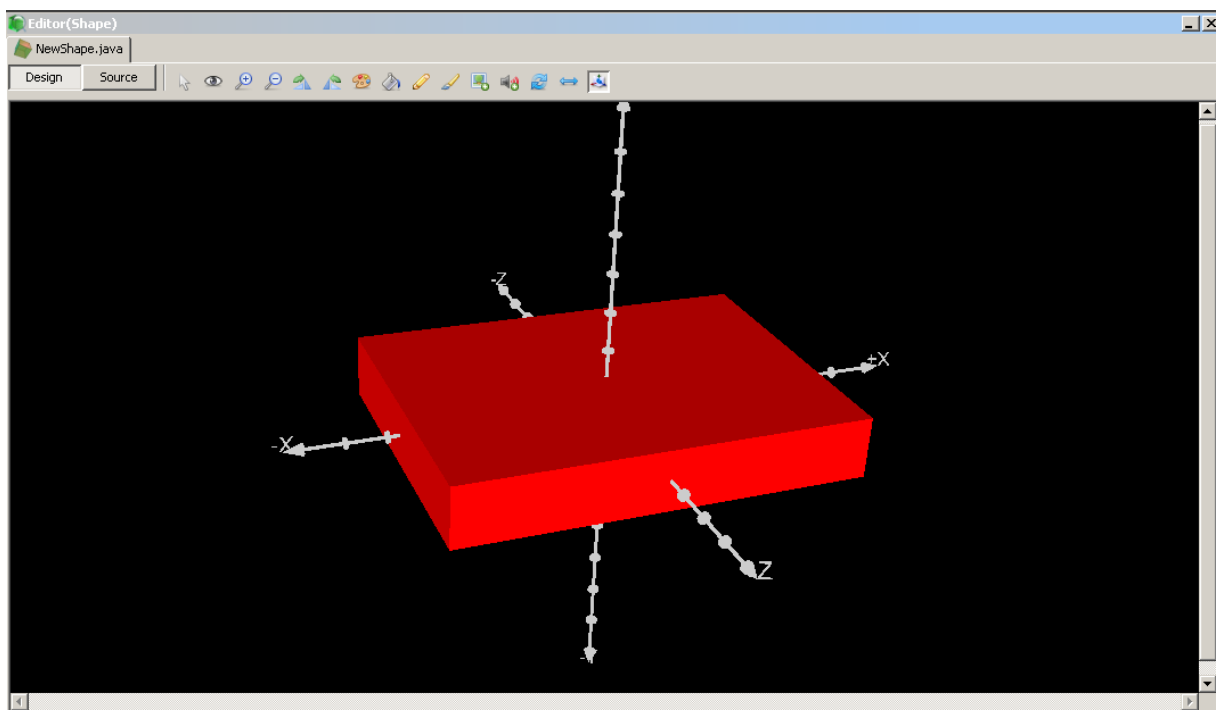


Figure 37: Scale on X and Z Axis

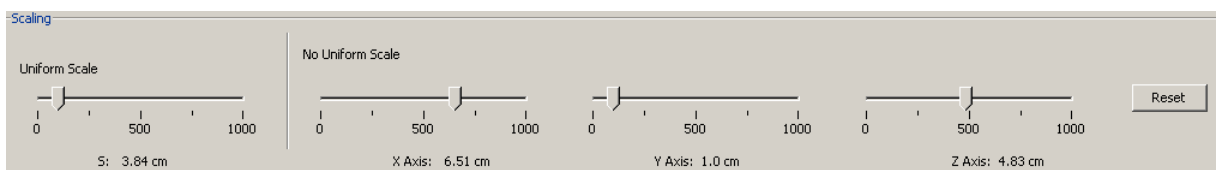


Figure 38: Ράβδοι κλίμακας μετατροπής του αντικειμένου

- **Interpolator Attributes**

Το εργαλείο αυτό σχετίζεται με τον εμπλουτισμό περιστρεφόμενης κίνησης του τρισδιάστατου αντικειμένου. Για να επιτευχθεί αυτό έχει χρησιμοποιηθεί η συμπεριφορά RotatorInterpolator του Java3D API η οποία σε συνδυασμό με τις παραμέτρους του αντικειμένου Alpha περιστρέφει ένα τρισδιάστατο αντικείμενο. Το εργαλείο αυτό χωρίζεται σε δύο τμήματα. Το πρώτο τμήμα προσδιορίζει τις παραμέτρους του αντικειμένου RotatorInterpolator που είναι η μέγιστη και η ελάχιστη γωνία περιστροφής ενώ το δεύτερο τμήμα ορίζει τις παραμέτρους Alpha οι οποίες σχετίζονται με το ποια χρονική στιγμή θα ξεκινήσει η περιστροφή, πόσες περιστροφές θα πραγματοποιήσει το αντικείμενο και για ποια χρονικά διαστήματα το αντικείμενο θα μεταβαίνει από την τιμή ένα στην τιμή μηδέν όταν μειώνεται η διάρκεια του Alpha και αντίστροφα από την τιμή μηδέν στην τιμή ένα όταν αυξάνεται η διάρκεια του Alpha καθώς παρέχεται η δυνατότητα στο χρήστη να μπορεί να επιλέξει αν το αντικείμενο θα μεταβαίνει από την τιμή ένα στην τιμή μηδέν ή από την τιμή μηδέν στην τιμή ένα.

Παρακάτω επισυνάπτεται το σχετικό εργαλείο.

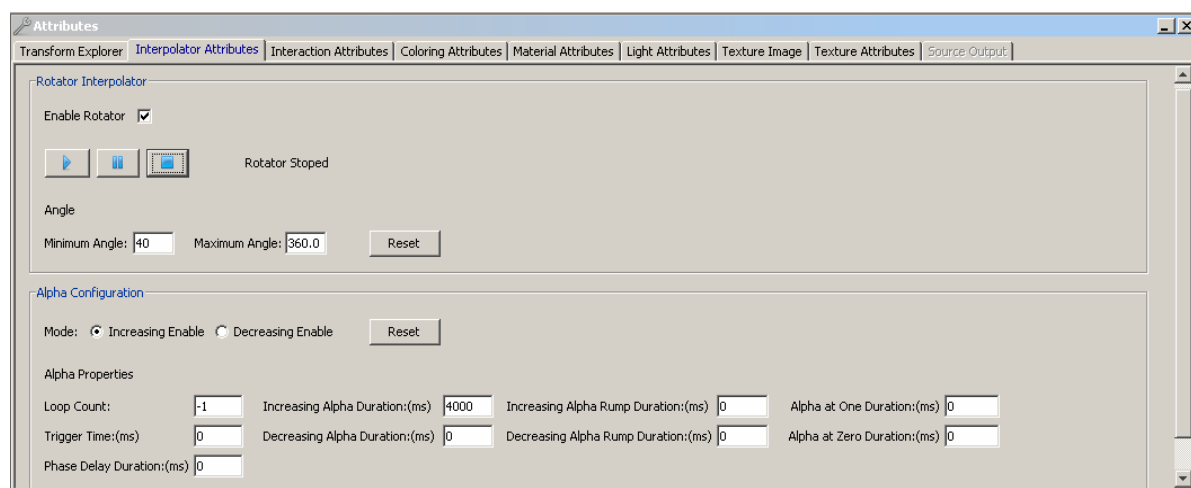


Figure 39: Interpolator Attributes

### **Rotator Interpolator**

Το τμήμα αυτό αποτελείται από ένα CheckBox οπου επιλέγοντάς το ενεργοποιείται η συμπεριφορά περιστρεφόμενης κίνησης στο τρισδιάστατο αντικείμενο το οποίο έχει σαν απόρροια να εμπλουτίζεται ο αυτόματα παραγόμενος κώδικας με τις κατάλληλες εντολές. Στη συνέχεια αποτελείται από τρία κουμπιά ελέγχου τα οποία σχετίζονται με την εκκίνηση, παύση και τερματισμό της κίνησης του τρισδιάστατου αντικειμένου. Τα κουμπιά αυτά έχουν προστεθεί ώστε να μπορεί ο χρήστης να ελέξει την κίνηση του αντικειμένου σε πραγματικό χρόνο οποιαδήποτε στιγμή. Τέλος υπάρχουν δύο ακόμα πεδία τα οποία προσδιορίζουν την μέγιστη και την ελάχιστη γωνία περιστροφής. Δηλαδή αν η ελάχιστη γωνία είναι 40 μοίρες και η μέγιστη γωνία είναι 360 μοίρες τότε μια περιστροφή ξεκινά από τις 40 και ολοκληρώνεται στις 360 μοίρες. Πάντα ο άξονας περιστροφής ορίζεται από το παραπάνω εργαλείο (Transform Explorer - Rotation) όπως αναφέραμε.

Όπως και στα υπόλοιπα εργαλεία έτσι και σε αυτό υπάρχει το κουμπί Reset το οποίο επαναφέρει τα πεδία στις αρχικές τιμές τους.

Παρακάτω επισυνάπτεται το σχετικό εργαλείο

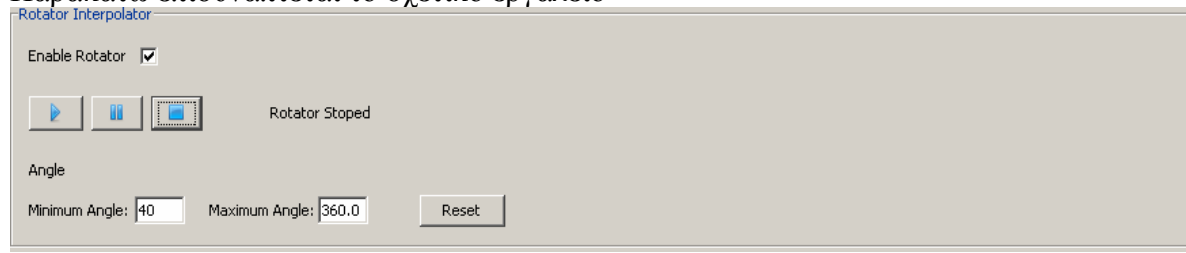


Figure 40: Rotator Interpolator

### **Alpha Configuration**

Το αντικείμενο Alpha παράγει μία τιμή κάθε φορά από 0.0 μέχρι και 1.0 και αυτή η τιμή ονομάζεται τιμή Alpha. Η παραγόμενη τιμή Alpha αλλάζει κάθε φορά και εξαρτάται από τις παραμέτρους που έχουμε ορίσει στο Alpha Configuration Panel.

Όταν η τιμή Alpha έχει την τιμή μηδέν ή ένα σημαίνει ότι το αντικείμενο ολοκλήρωσε μια ολόκληρη περιστροφή μεταξύ της μέγιστης και ελάχιστης γωνίας που του έχουμε ορίσει. Το αντικείμενο Alpha αποτελείται κυρίως από τέσσερις φάσεις : Increasing Alpha , Alpha at one , Decreasing Alpha , Alpha at Zero. Οι δύο πρώτες φάσεις ορίζουν την μεταβολή του Alpha από μηδέν σε ένα ενώ οι επόμενες δύο από ένα σε μηδέν.

Την κατάσταση μεταβολής του Alpha την ορίζει το πεδίο Mode το οποίο ορίζει αν θα έχουμε αύξηση ή μείωση του Alpha. Όταν το Alpha μεταβαίνει από 0 σε 1 (Increasing) τότε το αντικείμενο μας κινείται δεξιόστροφα ενώ όταν το Alpha μεταβαίνει από 1 σε 0 (Decreasing) τότε το αντικείμενο μας κινείται αριστερόστροφα.

Παρακάτω παρουσιάζονται οι παραμέτροι του Alpha.

- **Loop Count**  
Προσδιορίζει το πόσες φορές θα εκτελεστεί η περιστροφική κίνηση. Η τιμή -1 δηλώνει άπειρο πλήθος εκτελέσεων.
- **Trigger Time**  
Είναι ο χρόνος σε milliseconds που προσδιορίζει πόσος χρόνος θα περάσει για να εκτελεστεί η κίνηση μετά την ενεργοποίηση της.
- **Phase Delay Duration**  
Είναι ο χρόνος σε milliseconds που ορίζει πόσο χρονικό διάστημα θα περάσει μετά την ενεργοποίηση του Trigger Time ώστε να εκτελεστεί η κίνηση. Στην ουσία είναι άλλη μία καθυστέρηση μεταξύ Trigger Time και εκτέλεση κίνησης.
- **Mode**  
Η παράμετρος αυτή ορίζει την κατάσταση μεταβολής του Alpha. Υπάρχουν δύο διαθέσιμες καταστάσεις , η αύξηση του Alpha INCREASING\_ENABLE και η μείωση του Alpha DECREASING\_ENABLE. Όταν έχει επιλεγθεί η αύξηση του Alpha τότε τα πεδία που σχετίζονται με την μείωση του Alpha δεν λαμβάνονται υπόψη. Όταν έχει επιλεγθεί η μείωση του Alpha τότε τα πεδία που

σχετίζονται με την αύξηση του Alpha δεν λαμβάνονται υπόψιν.

- **Increasing Alpha Duration**  
Το πεδίο αυτό ορίζει το χρονικό διάστημα που απαιτείται για να μεταβεί η τιμή του Alpha από την τιμή μηδέν στην τιμή 1.
- **Increasing Alpha Ramp Duration**  
Το πεδίο αυτό ορίζει το χρονικό διάστημα κατά το οποίο το Alpha αυξάνεται από την αρχή του Increasing Alpha Duration και μειώνεται κατά το τέλος του Increasing Alpha Duration..
- **Alpha at One Duration**  
Το πεδίο αυτό ορίζει την χρονική περίοδο που το Alpha μένει στην τιμή ένα.
- **Decreasing Alpha Duration**  
Το πεδίο αυτό ορίζει το χρονικό διάστημα που απαιτείται για να μεταβεί η τιμή του Alpha από την τιμή 1 στην τιμή 0.
- **Decreasing Alpha Rump Duration**  
Το πεδίο αυτό ορίζει το χρονικό διάστημα κατά το οποίο το Alpha αυξάνεται από την αρχή του Decreasing Alpha Duration και μειώνεται κατά το τέλος του Decreasing Alpha Duration.
- **Alpha at Zero Duration**  
Το πεδίο αυτό ορίζει την χρονική περίοδο που η τιμή του Alpha παραμένει στο μηδέν.

Παρακάτω επισυνάπτεται το σχετικό εργαλείο

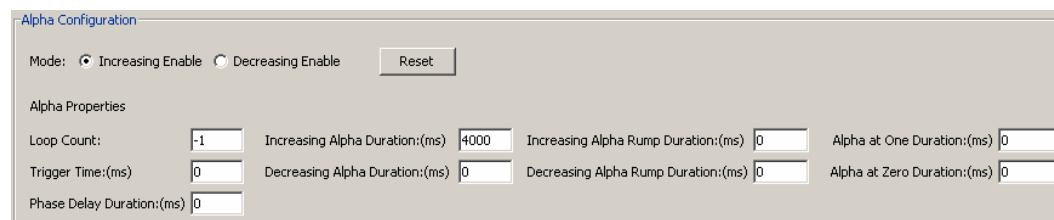


Figure 41: Alpha Configuration

- **Interaction Attributes**

Το εργαλείο αυτό σχετίζεται με τον εμπλουτισμό του τρισδιάστατου αντικειμένου με διαδραστικές μεθόδους – λειτουργίες. Οι διαδραστικές λειτουργίες σχετίζονται με την επικοινωνία του τρισδιάστατου αντικειμένου με το ποντίκι. Ο Editor παρέχει την ικανότητα στον χρήστη να επικοινωνεί και να ελέγχει το τρισδιάστατο αντικείμενο μέσω του ποντικιού. Το εργαλείο αυτό αποτελείται από τέσσερα τμήματα. Το πρώτο τμήμα σχετίζεται με την μετακίνηση του αντικειμένου μέσα στο χώρο. Το δεύτερο τμήμα σχετίζεται με την περιστροφή του αντικειμένου μέσα στο χώρο και το τρίτο και τέταρτο τμήμα με το Zoom – μεγέθυνση του αντικειμένου μέσα στο χώρο. Ο χρήστης έχει την ικανότητα να ενεργοποιήσει όλες τις λειτουργίες μαζί ή την κάθε μια μεμονωμένα μέσω του κουμπιού Enabled που βρίσκεται στην κάθε διαδραστική λειτουργία.. Τέλος παρέχεται στον χρήστη η επιλογή να εφαρμόσει τις διαδραστικές λειτουργίες στο αντικείμενο σε πραγματικό χρόνο επιλέγοντας το πεδίο Enable Mouse Interaction Test και πατώντας Apply.

Παρακάτω παρουσιάζονται τα τμήματα αυτά λεπτομερώς.

The image shows a software interface for mouse interaction settings. It is organized into several sections, each with an 'Enabled' checkbox and a 'Reset' button. The sections are: 'Enable Mouse Interaction Test' (checkbox unchecked, 'Apply' button), 'Mouse Translate' (checkbox unchecked, 'Translate X Factor' 0.02, 'Translate Y Factor' 0.02, 'Reset' button), 'Mouse Rotate' (checkbox unchecked, 'Rotation X Factor' 0.03, 'Rotation Y Factor' 0.03, 'Reset' button), 'Mouse Zoom' (checkbox unchecked, 'Zoom Factor' 0.04, 'Reset' button), and 'Mouse Wheel Zoom' (checkbox unchecked, 'Wheel Zoom Factor' 0.1, 'Reset' button).

Figure 42: Mouse Interaction tool

- **Mouse Translate**

Η συμπεριφορά αυτή δίνει την ικανότητα στο χρήστη να ελέξει την τοποθεσία του αντικειμένου σε συντεταγμένες (X,Y). Εκτελείται με σύρσιμο του ποντικιού πατώντας το δεξί κουμπί του ποντικιού σταθερό. Επίσης με τα πεδία Translate X Factor και Translate Y Factor παρέχεται στο χρήστη η δυνατότητα να ορίζει αυτός πόσα εκατοστά θα κινείται στον X και Y άξονα αντίστοιχα το αντικείμενο με κάθε σύρσιμο του ποντικιού. Τέλος με το κουμπί Enabled ενεργοποιεί ή απενεργοποιεί την λειτουργία αυτή.

- **Mouse Rotate**

Η συμπεριφορά αυτή δίνει την ικανότητα στο χρήστη να ελεξει την περιστροφή του τρισδιάστατου αντικειμένου στο χώρο σε συντεταγμένες (X,Y) πατώντας αριστερό κλικ και μεταβάλλοντας μόνο την θέση του ποντικιού. Επίσης με τα πεδία Rotate X Factor και Rotate Y Factor παρέχεται στο χρήστη η δυνατότητα να ορίζει αυτός την περιστροφή του αντικειμένου στον X και Y άξονα αντίστοιχα με κάθε σύρσιμο του ποντικιού.

- **Mouse Zoom**

Η συμπεριφορά αυτή δίνει στον χρήστη την ικανότητα να ελέξει την μεγέθυνση του αντικειμένου στον Z άξονα μόνο. Για να εκτελεστή η λειτουργία επιτυχώς χρησιμοποιούμε το δεύτερο κουμπί του ποντικιού και μεταβάλλουμε το ποντίκι μπροστά ή πίσω. Επίσης με το πεδίο Zoom Factor παρέχεται στο χρήστη η δυνατότητα να ορίζει αυτός πόσα εκατοστά θα αυξάνεται ή θα μειώνεται η μεγέθυνση ή η συρρίκνωση του αντικειμένου.

- **Mouse Wheel Zoom**

Η συμπεριφορά αυτή παράγει το ίδιο ακριβώς αποτέλεσμα με την

προηγούμενη με την διαφορά ότι εκτελείται μόνο με κύλιση του ποντικιού και ταυτόχρονο πάτημα του δευτέρου κουμπιού.

Τέλος σε κάθε από τα παραπάνω εργαλεία παρέχεται το κουμπί Reset το οποίο επαναφέρει τα πεδία στις αρχικές τους τιμές.

- **Coloring Attributes**

Το εργαλείο αυτό παρέχει στο χρήστη την δυνατότητα να ορίσει το χρώμα του τρισδιάστατου αντικειμένου, το χρώμα του φόντου στο οποίο βρίσκεται το αντικείμενο και το μοντέλο επισκίασης του αντικειμένου. Το εργαλείο αποτελείται από τρία τμήματα το Shape Coloring Attributes που σχετίζεται με το χρώμα του τρισδιάστατου αντικειμένου, το Background Coloring Attributes το οποίο σχετίζεται με το χρώμα του φόντου και το Shade που σχετίζεται με την σκίαση του αντικειμένου. Για να ορίσει ο χρήστης καθένα από τα παρακάτω χρώματα υπάρχει το κουμπί Set το οποίο ανοίγει έναν επιλογέα χρωμάτων όπου από εκεί ο χρήστης καλείται να επιλέξει ένα χρώμα της αρέσκειας του. Κάθε φορά που ο χρήστης θα επιλέγει ένα χρώμα αυτό θα εμφανίζεται αριστερά του κουμπιού Set με την μορφή χρωματισμένης ετικέτας ενώ δεξιά του κουμπιού Set θα εμφανίζονται οι τιμές του χρώματος σε RGB red, green, blue.

Για να ορίσει ο χρήστης το μοντέλο σκίασης του αντικειμένου θα πρέπει να επιλέγει μεταξύ των επιλογών Shade Flat, Shade Gouraound, Nicest, Fastest. Τέλος σε κάθε από τα τρία τμήματα του εργαλείου υπάρχει ένα Reset κουμπί το οποίο επιστρέφει τις παραμέτρους των χρωμάτων και της σκίασης στην αρχική τους τιμή.



Figure 43: Coloring Attributes



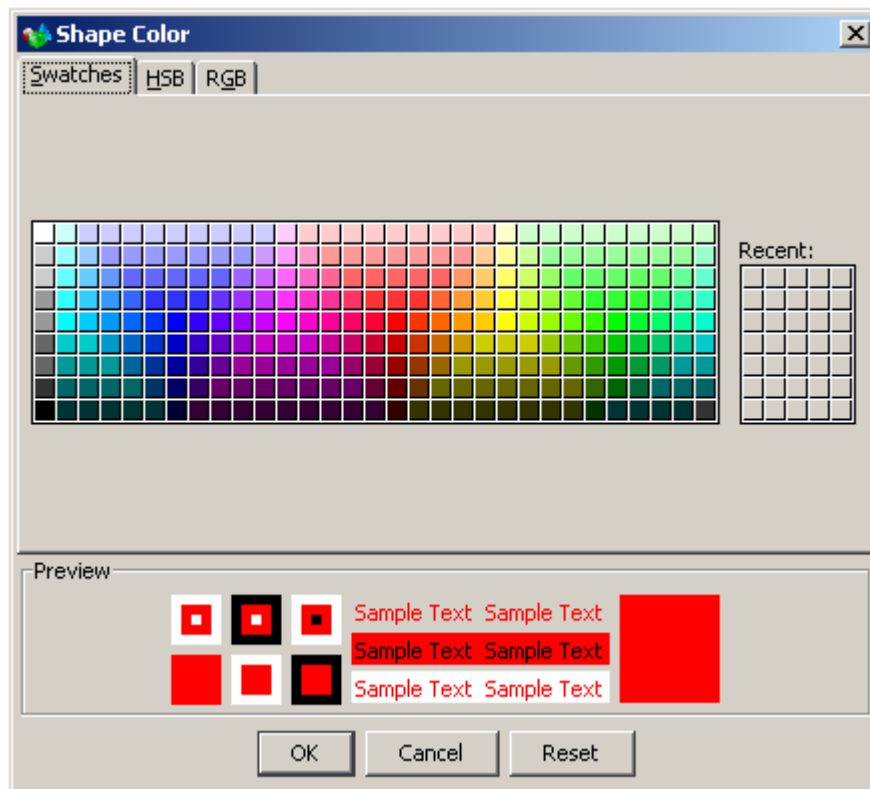


Figure 44: Επιλογέας χρωμάτων – Color Chooser

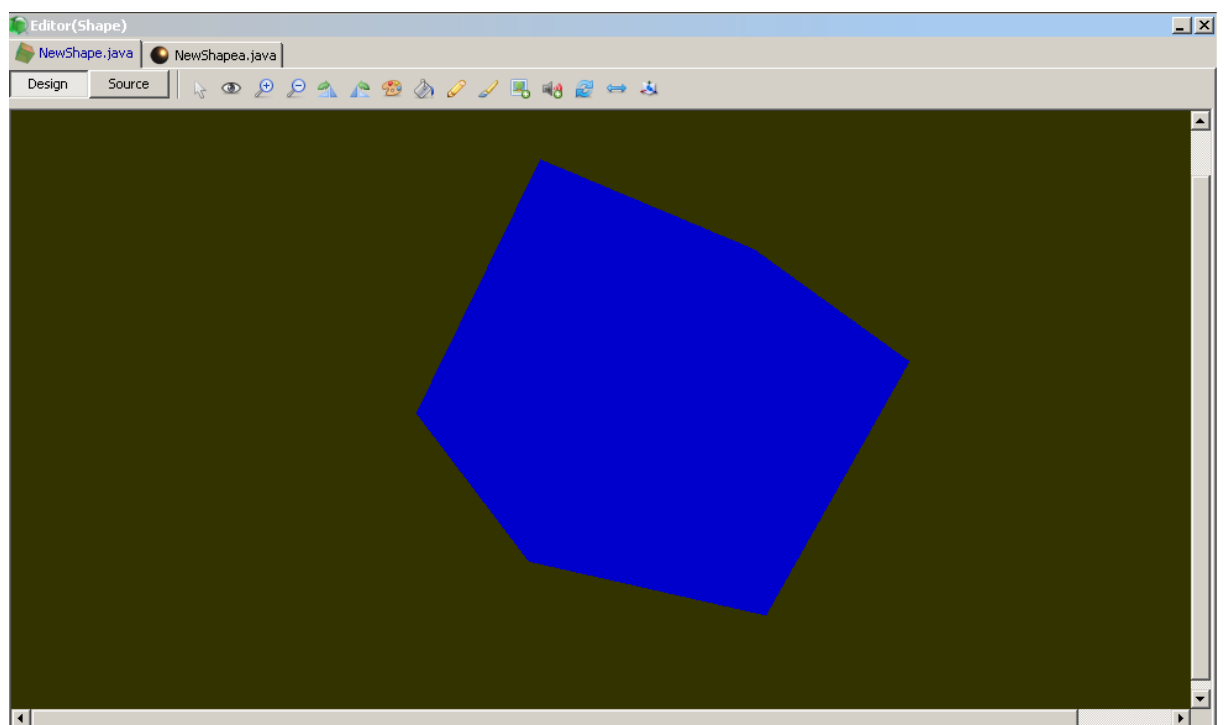


Figure 45: Αλλαγή χρωμάτων σε σχήμα και φόντο

- **Material Attributes**

Το εργαλείο αυτό ορίζει τα χρώματα της επιφάνειας που αποτελούν ένα τρισδιάστατο αντικείμενο και επιτρέπει στο χρήστη να ενεργοποιήσει ή να απενεργοποιήσει του φωτισμού ενός αντικειμένου καθώς και να ελέγξει την τιμή της φωτεινότητας του φωτισμού. Τα χρώματα της επιφάνειας ενός αντικειμένου αναμειγνύονται μεταξύ τους επηρεασμένα άλλοτε από την θέση και το χρώμα του φωτισμού του αντικειμένου και άλλοτε ανεξάρτητα. Το εργαλείο αυτό αποτελείται από έξι ξεχωριστά τμήματα. Το πρώτο τμήμα (Light Category) παρέχει την δυνατότητα στο χρήστη να ενεργοποιήσει ή να απενεργοποιήσει τον φωτισμό σε ένα αντικείμενο, κάθε φορά που ενεργοποιεί ο χρήστης τον φωτισμό αναφέρεται ο τύπος του φωτισμού σε μορφή ετικέτας κάτω από το πεδίο ενεργοποίησης του φωτισμού.

Κάθε φορά που το πεδίο Enabled είναι επιλεγμένο εμπλουτίζεται ο αυτόματα παραγόμενος κώδικας με τις κατάλληλες εντολές συμπεριλαμβάνοντας και τον τύπο φωτισμού που θα παρουσιάσουμε στο επόμενο εργαλείο.

Τα επόμενα τέσσερα τμήματα ορίζουν τα χρώματα επιφάνειας ενός αντικειμένου τα οποία είναι: Ambient Color , Diffuse Color , Specular Color, Emissive Color. Για να ορίσει ο χρήστης καθένα από τα παραπάνω χρώματα υπάρχει το κουμπί Set το οποίο ανοίγει έναν επιλογέα χρωμάτων όπου από εκεί ο χρήστης καλείται να επιλέξει ένα χρώμα της αράς του. Κάθε φορά που ο χρήστης θα επιλέγει ένα χρώμα αυτό θα εμφανίζεται αριστερά του κουμπιού Set με την μορφή χρωματισμένης ετικέτας ενώ δεξιά του κουμπιού Set θα εμφανίζονται οι τιμές του χρώματος σε RGB red , green , blue. Τα χρώματα επιφάνειας του αντικειμένου Ambient και Emissive συνδυάζονται ανεξάρτητα από τον προσανατολισμό την θέση και το χρώμα του φωτισμού σε μία σκηνή, ενώ τα χρώματα Diffuse και Specular συνδυάζονται με το χρώμα του κάθε φωτισμού.

Τέλος το τελευταίο τμήμα (Shininess Color) αποτελείται από μία ράβδο κύλισης η οποία ορίζει την φωτεινότητα του φωτός που έχει οριστεί για το αντικείμενο. Η ράβδος παίρνει τιμές από 0 που μέχρι 128 που είναι και η μέγιστη φωτεινότητα.

Όπως σε κάθε εργαλείο που περιγράψαμε παραπάνω έτσι και σε αυτό υπάρχει σε κάθε τμήμα ένα κουμπί Reset το οποίο χρησιμοποιείται για να επαναφέρει τις παραμέτρους του εργαλείου στην αρχική τους τιμή.

Παρακάτω επισυνάπτεται το σχετικό εργαλείο.

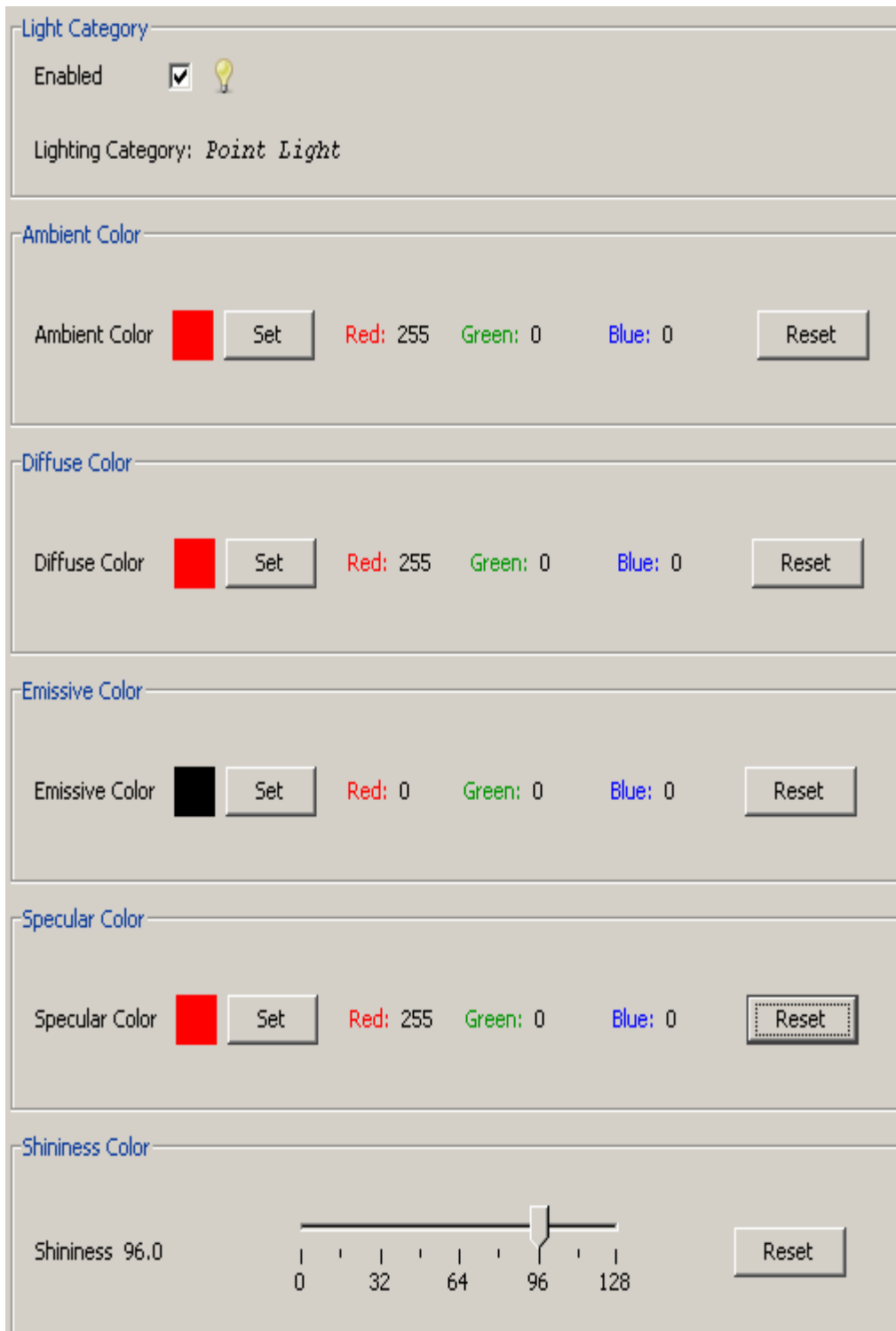


Figure 46:Material Attributes Tool

- **Light Attributes**

Το εργαλείο αυτό σχετίζεται με τον εμπλουτισμό της σκηνής του τρισδιάστατου αντικειμένου με φωτισμό. Το εργαλείο αυτό αποτελείται από τέσσερα τμήματα τα οποία ορίζουν την κατηγορία του φωτισμού της σκηνής, την θέση και τον προσανατολισμό του φωτισμού και τέλος το χρώμα του φωτισμού.

Το πρώτο τμήμα ονομάζεται Light Category και είναι υπεύθυνο για την επιλογή του φωτισμού της σκηνής. Ο χρήστης έχει την ικανότητα να επιλέξει έναν φωτισμό κάθε φορά. Κάθε φορά που επιλέγει έναν φωτισμό ο φωτισμός αυτός εφαρμόζεται στην σκηνή σε πραγματικό χρόνο μόνο και μόνο αν το πεδίο (Enabled – Material Attributes) είναι επιλεγμένο.

Το δεύτερο τμήμα ονομάζεται Ambient Light και σχετίζεται με τον φωτισμό Ambient. Στην κατηγορία αυτή ο χρήστης μπορεί να αλλάξει μόνο το χρώμα του φωτισμού καθώς η κατηγορία αυτή δεν υποστηρίζει θέση και προσανατολισμό φωτισμού ο φωτισμός απλώνεται σε όλη την σκηνή.

Το τρίτο τμήμα ονομάζεται Directional Light και σχετίζεται με το φωτισμό Directional Light. Στην κατηγορία αυτή ο χρήστης έχει την δυνατότητα να ορίσει τον προσανατολισμό του φωτισμού σε συντεταγμένες (X, Y, Z) μέσω των αντίστοιχων ράβδων κύλισης. Οι μέγιστες τιμές που μπορεί να πάρει η κάθε ράβδος είναι από -1000 μέχρι +1000. Κάθε φορά που μεταβάλλεται μία από τις τιμές των συντεταγμένων η τιμή τις μεταβολής που παράγεται από την ράβδο διαιρείται με το 100 ώστε η μέγιστη τιμή που μπορεί να οριστεί για τον προσανατολισμό του φωτός στην σκηνή να είναι από -10cm μέχρι +10 cm. Επίσης στην κατηγορία αυτή ο χρήστης έχει την δυνατότητα να μεταβάλει το χρώμα του αντικειμένου με τον γνωστό τρόπο που έχουμε προαναφέρει.

Το τέταρτο τμήμα ονομάζεται Point Light και σχετίζεται με τον φωτισμό Point Light. Στην κατηγορία αυτή ο χρήστης μπορεί να ορίσει το σημείο φωτισμού σε συντεταγμένες (X, Y, Z) μέσω των αντίστοιχων ράβδων κύλισης. Οι μέγιστες τιμές που μπορεί να πάρει η κάθε ράβδος είναι από -2000 μέχρι +2000. Κάθε φορά που μεταβάλλεται μία από τις τιμές των συντεταγμένων η τιμή τις μεταβολής που παράγεται από την ράβδο διαιρείται με το 100 ώστε η μέγιστη τιμή που μπορεί να οριστεί για την θέση του φωτός στην σκηνή να είναι από -20cm μέχρι +20 cm. Επίσης ο χρήστης μπορεί να ορίσει και την εξασθένηση του φωτός η οποία ορίζεται από την τιμή των συντεταγμένων του χώρου (X, Y, Z) μέσω των αντίστοιχων ράβδων κύλισης. Οι μέγιστες τιμές που μπορεί να πάρει η κάθε ράβδος είναι από -200 μέχρι +200. Κάθε φορά που μεταβάλλεται μία από τις τιμές των ράβδων κύλισης η τιμή τις μεταβολής που παράγεται από την ράβδο διαιρείται με το 10 ώστε η μέγιστη τιμή που μπορεί να οριστεί για να είναι από -20cm μέχρι +20 cm.


Παρακάτω επισυνάπται το σχετικό εργαλείο.

**Light Category**

Light Category  Ambient  Directional  Point  Spot


---

**Ambient Light**

Ambient Color   Red: 77 Green: 77 Blue: 77

---

**Directional Light**


Directional Color   Red: 255 Green: 255 Blue: 255

Light Direction

X: -0.0 Y: -0.0 Z: 0.1

---

**Point Light**

Point Color   Red: 255 Green: 255 Blue: 255

Light Position

X: -4.0 Y: 8.0 Z: 16.0

Attenuation

X: 1.0 Y: 0.0 Z: 0.0

**Figure 47:Light Attributes**

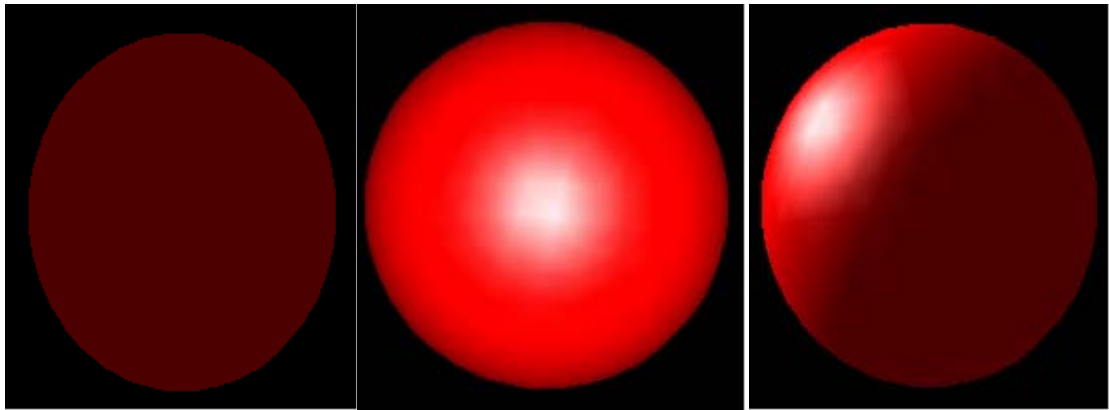


Figure 48: Ambient – Directional – Point Light

- **Texture Image Tool**

Το εργαλείο αυτό σχετίζεται με τον εμπλουτισμό και την επεξεργασία του αντικειμένου με εικόνα. Αποτελείται από τρία τμήματα που ορίζουν την επιλογή της εικόνας που πρόκειται να τοποθετηθεί στο τρισδιάστατο αντικείμενο την επεξεργασία των συνόρων μεταξύ τρισδιάστατου αντικειμένου και εικόνας καθώς και την επιλογή φιλτραρίσματος της εικόνας για καλύτερη απόδοση. Το πρώτο τμήμα ονομάζεται Texture Image, το δεύτερο τμήμα ονομάζεται Boundary και το τρίτο τμήμα ονομάζεται

**Texture Image**

Το πρώτο τμήμα ονομάζεται Texture Image και είναι υπεύθυνο για την επιλογή και εμφάνιση της εικόνας στο τρισδιάστατο αντικείμενο. Αποτελείται από ένα πεδίο επιλογής που ενεργοποιεί και απενεργοποιεί τον εμπλουτισμό του αντικειμένου με την σχετική εικόνα και από το κουμπί Browse το οποίο εμφανίζει έναν επιλογέα αρχείων φιλτραρισμένο να εμφανίζει μόνο αρχεία εικόνας jpg, png, jpeg.

Όταν ο χρήστης επιλέξει μία εικόνα από τον επιλογέα τότε η εικόνα εμφανίζεται δίπλα από το πεδίο επιλογής μαζί με το path της εικόνας. Έτσι ο χρήστης κάθε φορά ξέρει ποια εικόνα πρόκειται να χρησιμοποιηθεί ασχέτως αν έχει ενεργοποιηθεί ο εμπλουτισμός του αντικειμένου με την εικόνα αυτή. Παρακάτω επισυνάπτεται το τμήμα αυτό του εργαλείου.

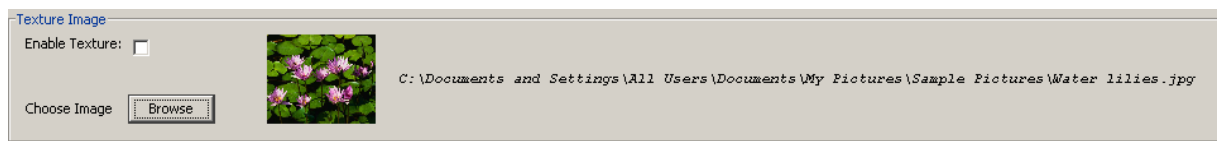


Figure 49: Texture Image

**Boundary**

Το δεύτερο τμήμα του εργαλείου αυτού ονομάζεται Boundary και σχετίζεται με την χαρτογράφηση (mapping) που πρόκειται να ακολουθηθεί όταν η συντεταγμένες του texture που έχουν υπολογιστεί αυτόματα ξεπερνούν την κλίμακα [0,1] του χώρου της εικόνας. Ποιό συγκεκριμένα όταν ένα αντικείμενο πρόκειται να εμπλουτιστεί με μια εικόνα δημιουργούνται κάποιες συντεταγμένες τις οποίες καλείται να καλύψει η εικόνα. Έτσι το API κάνει ένα σχετικό mapping της εικόνας με τις συντεταγμένες αυτές. Αν σε μία περίπτωση οι συντεταγμένες είναι μεγαλύτερες από το εύρος που πρόκειται να καλύψει η εικόνα τότε το εργαλείο αυτό δίνει κάποιες επιλογές για την

λύση του προβλήματος αυτού. Οι επιλογές που παρέχει το εργαλείο αυτό είναι να κάνει “wrap” ή “clamp” στην εικόνα τόσο στο οριζόντιο επίπεδο όσο και στο κάθετο. Η επιλογή Wrap επαναλαμβάνει την εικόνα στα σημεία που δεν έχουν καλυφτεί από αυτήν ενώ η επιλογή Clamp συμπληρώνει το ακάλυπτο σημείο του αντικειμένου με το χρώμα από το άκρο της εικόνας. Για την οριζόντια επιλογή Wrap ή Clamp χρησιμοποιείται το πεδίο της εφαρμογής Boundary S Mode. Για την κάθετη επιλογή Wrap ή Clamp χρησιμοποιείται το πεδίο της εφαρμογής Boundary T Mode. Επίσης το τμήμα αυτό του εργαλείου σου προσφέρει την δυνατότητα να ορίσεις εσύ το χρώμα όταν έχεις επιλέξει την επιλογή χαρτογράφησης Clamp. Για την επιλογή του χρώματος αυτού ο editor παρέχει το πεδίο Boundary Color το οποίο ορίζεται όπως κάθε πεδίο επιλογής χρώματος στην εφαρμογή. Τέλος και σε αυτό το εργαλείο υπάρχει το κουμπί Reset το οποίο επαναφέρει το πεδία στις αρχικές τιμές τους. Παρακάτω επισυνάπτεται το τμήμα αυτό του εργαλείου.

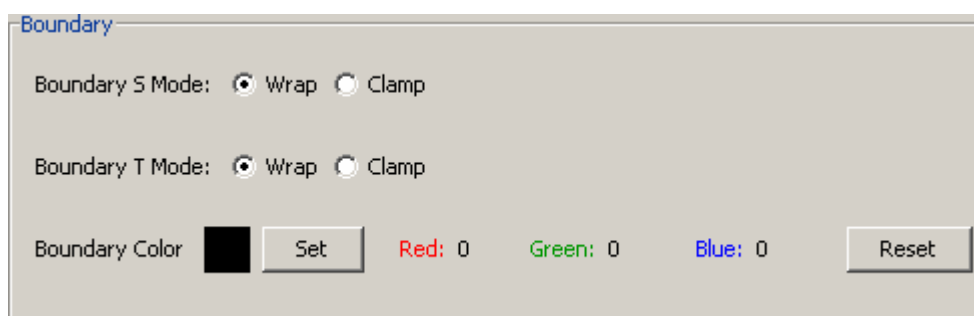


Figure 50: Boundary Tool

### **Filtering(Φιλτράρισμα)**

Το εργαλείο αυτό είναι υπεύθυνο για το χειρισμό περιπτώσεων όπου το μέγεθος ενός pixel είναι αντίστοιχο με το μέγεθος πολλαπλών texel ή μικρότερο του ενός texel. Ποιο συγκεκριμένα κατά τον υπολογισμό των Texture Coordinates γίνεται ένα σχετικό mapping μεταξύ pixels και texels, δηλαδή ένα pixel να αντιστοιχεί σε ένα texel. Όμως στην παραπάνω περίπτωση δημιουργούνται κάποιες ατέλειες στην απόδοση της εικόνας. Έτσι το εργαλείο αυτό παρέχει κάποιες επιλογές για τον χειρισμό των παραπάνω περιπτώσεων. Οι επιλογές αυτές είναι το Magnification Filter και το Minification Filter.

Το Magnification Filter ορίζει τι πρέπει να γίνει όταν ένα pixel είναι μικρότερο από ένα texel. Σε αυτήν την περίπτωση μπορεί να γίνει δειγματοληψία σημείου (point sampling) όπου στην οποία επιλέγεται το χρώμα του κοντινότερου texel ή υπολογίζεται το χρώμα μέσω των γειτονικών texels.

Το Minification Filter ορίζει τι πρέπει να γίνει όταν ένα pixel είναι μεγαλύτερο από ένα texel. Το πρόβλημα εδώ είναι ότι ένα pixel μπορεί να αποτελείται μόνο από ένα χρώμα. Σε αυτήν την περίπτωση μπορεί να γίνει δειγματοληψία σημείου (point sampling) όπου στην οποία επιλέγεται το χρώμα του κοντινότερου texel ή το χρώμα των γειτονικών texels. Οι πιθανές επιλογές που μπορούν να πάρουν τα παραπάνω φίλτρα είναι FASTEST όπου επιλέγεται ο γρηγορότερος τρόπος απόδοσης, NICEST όπου επιλέγεται ο ωραιότερος τρόπος απόδοσης, BASE\_LEVEL\_POINT όπου επιλέγεται το κοντινότερο texel, BASE\_LEVEL\_LINEAR όπου επιλέγεται η διγραμμική παρεμβολή μεταξύ τεσσάρων γειτονικών texels και τέλος οι επιλογές MULTI\_LEVEL\_LINEAR και MULTI\_LEVEL\_POINT.

Τέλος και αυτό το τμήμα του εργαλείου περιέχει ένα κουμπί Reset για την επαναφορά των τιμών των παραμέτρων στην αρχική τους τιμή.

Παρακάτω επισυνάπτεται το σχετικό εργαλείο.

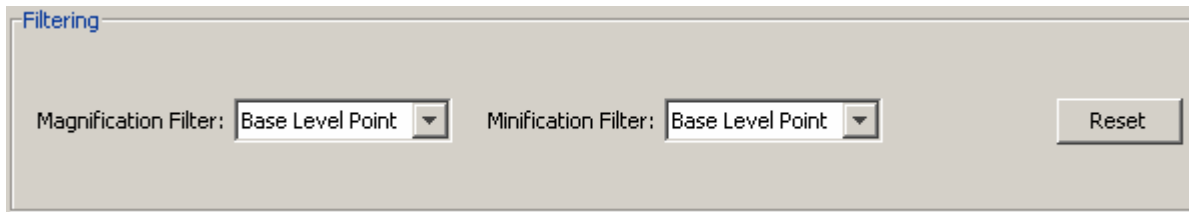


Figure 51: Φιλτράρισμα

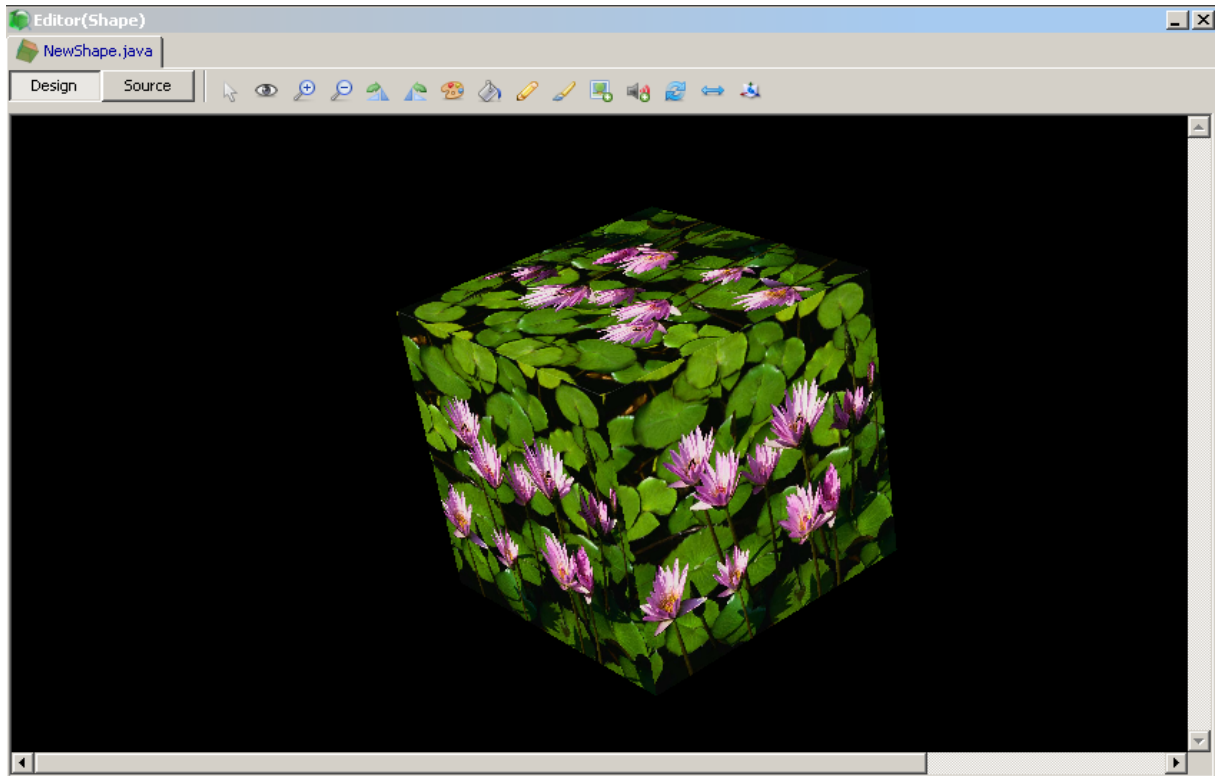


Figure 52: Κύβος εμπλουτισμένος με εικόνα

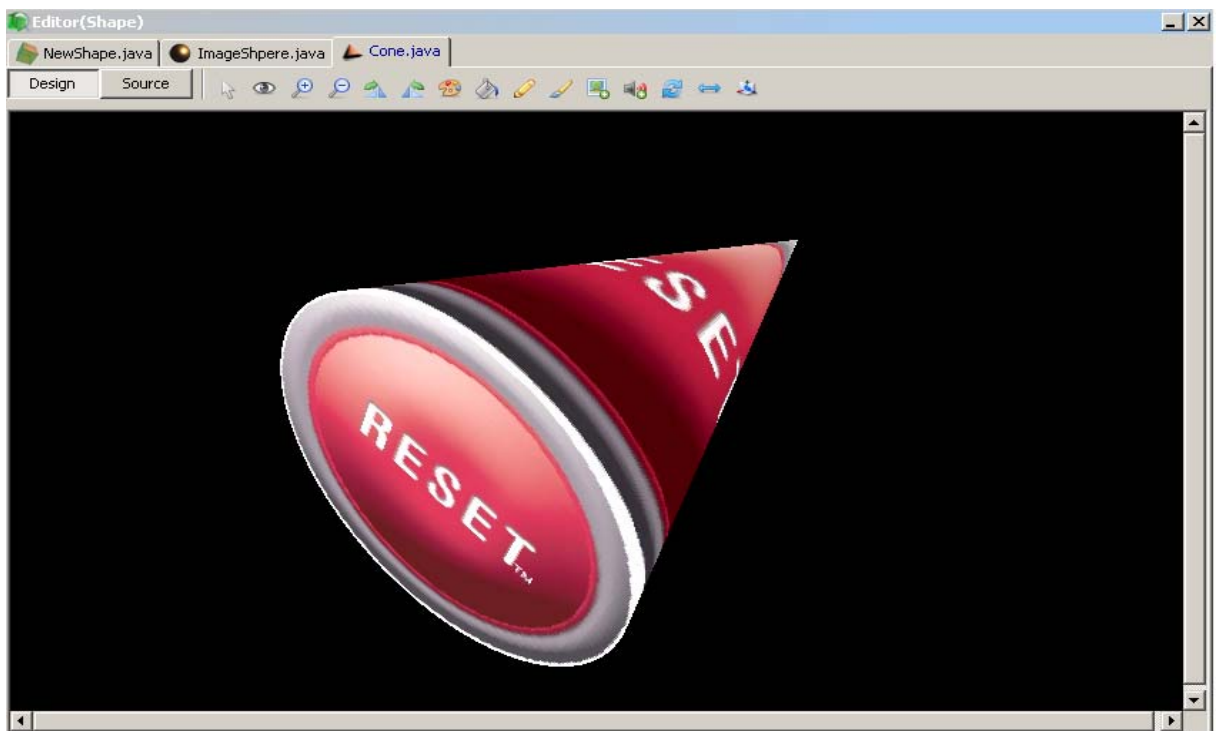


Figure 53: Κώνος εμπλουτισμένος με εικόνα



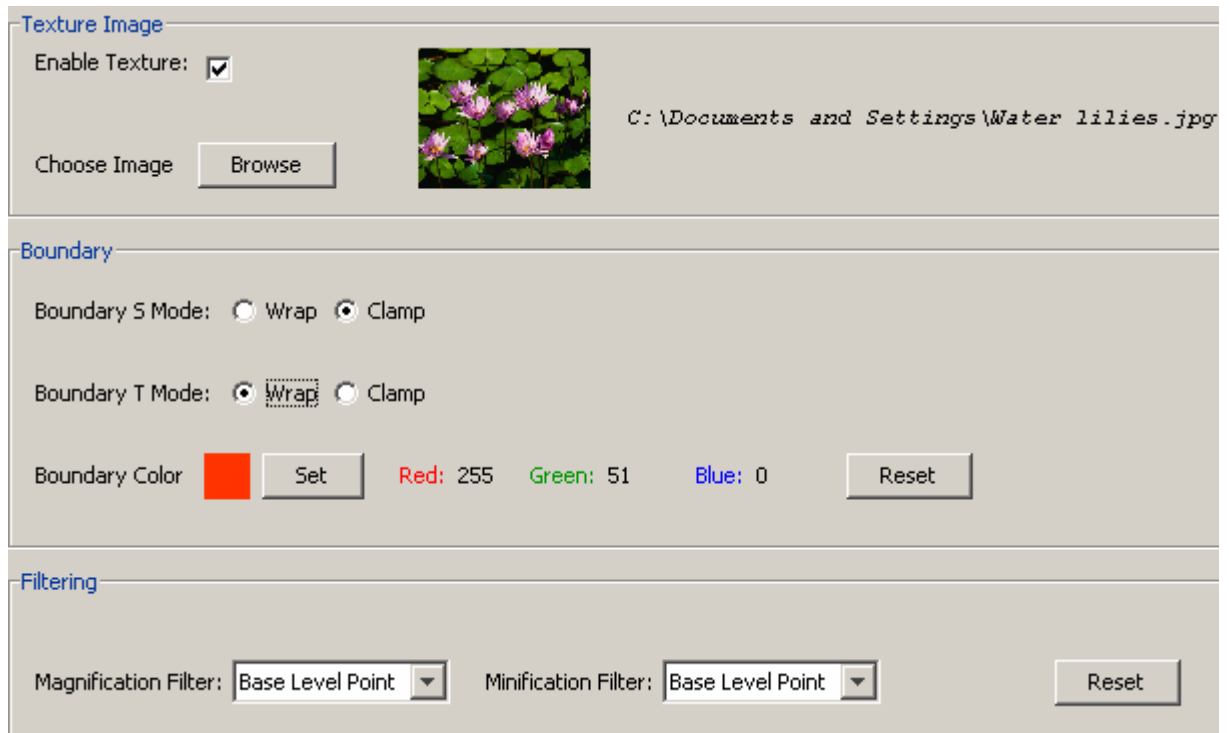


Figure 54: Texture Image Tool

- **Texture Attribute Tool**

Το εργαλείο αυτό σχετίζεται με τον συνδυασμό του non-texture χρώματος και του texture χρώματος. Ποιό συγκεκριμένα το εργαλείο αυτό παρέχει την δυνατότητα στο χρήστη να συνδυάσει τα χρώματα του αντικειμένου με τα χρώματα της εικόνας που το περιβάλλει. Επίσης το εργαλείο αυτό παρέχει στον χρήστη την δυνατότητα να βελτιώσει την απόδοση της εικόνας όταν αυτή παρατηρείται από τα άκρα της. Το εργαλείο αυτό αποτελείται από τρία τμήματα τα οποία είναι: Το πρώτο τμήμα ονομάζεται Texture Mode και αποτελείται από τέσσερα πεδία επιλογής ανάμειξης του χρώματος της εικόνας με αυτό του αντικειμένου. Οι πιθανές επιλογές είναι Modulate, Replace, Decal και Blend.

#### Replace

Σε αυτήν την κατηγορία το Texture (εικόνα) παρέχει το χρώμα για ένα pixel.

#### Modulate

Σε αυτήν την κατηγορία το χρώμα του Texture συνδυάζεται με αυτό του αντικειμένου.

#### Decal

Σε αυτήν την κατηγορία το χρώμα του Texture εφαρμόζεται σαν επιγραφή στο αντικείμενο.

#### Blend

Σε αυτήν την κατηγορία το χρώμα του αντικειμένου αναμειγνύεται με αυτό του Texture. Επίσης στην κατηγορία αυτή μπορεί να οριστεί νέο χρώμα ανάμειξης από το εργαλείο Blend Color.

Το δεύτερο τμήμα ονομάζεται Blend Color και είναι υπεύθυνο για τον ορισμό του χρώματος που πρόκειται να αναμιχθεί με το χρώμα της εικόνας. , όταν η επιλογή του Texture Mode είναι Blend. Η επιλογή του χρώματος γίνεται με τον ίδιο τρόπο όπως η επιλογή χρωμάτων στα άλλα εργαλεία της εφαρμογής.

Τέλος το τελευταίο τμήμα ονομάζεται Perspective και ορίζει την ποιότητα της απόδοσης της εικόνας όταν αυτή παρατηρείται από τα άκρα της. Οι διαθέσιμες επιλογές εδώ είναι NICE , FASTEST.

Παρακάτω επισυνάπτεται το σχετικό εργαλείο.

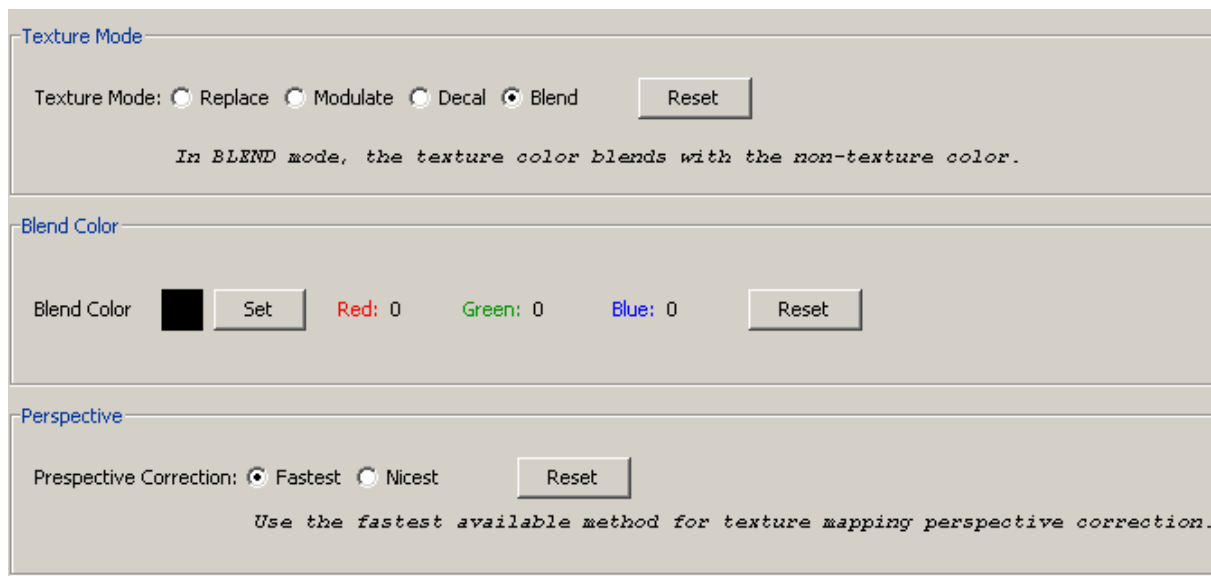


Figure 55:Texture Attributes Tool

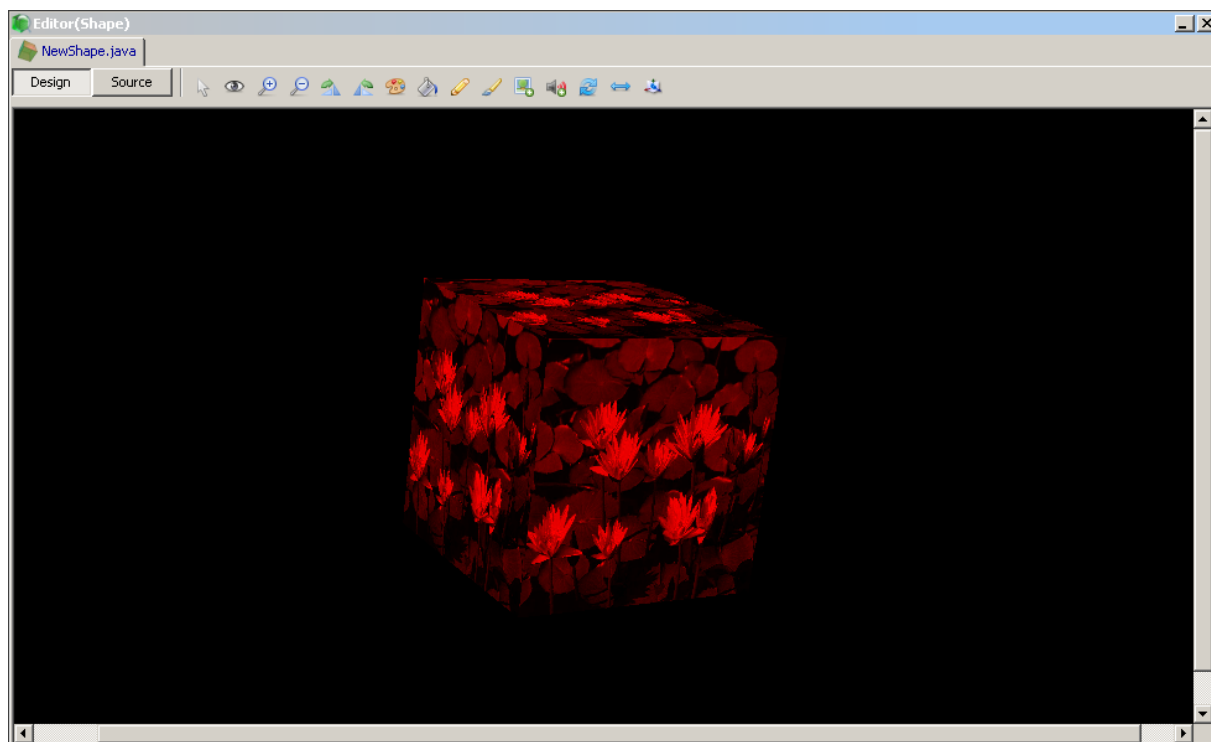


Figure 56:Cube in Modulate option

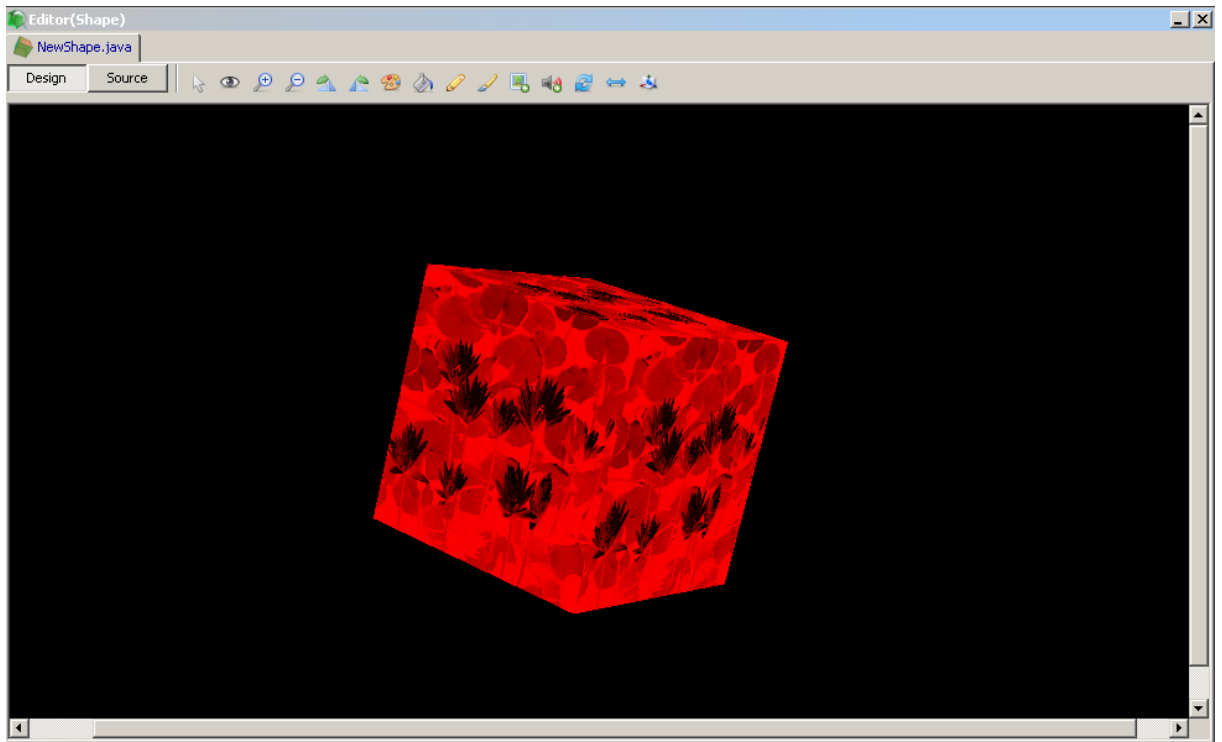


Figure 57:Cube in Blend option when Blend color is black

#### 4.2.5. Επιπρόσθετα παράθυρα και συστατικά εφαρμογής

Για την ολοκλήρωση της η εφαρμογή αποτελείται και από κάποια επιπρόσθετα παράθυρα διαλόγου τα οποία ορίζουν βασικές οντότητες της εφαρμογής.

- **Παράθυρο δημιουργίας Project**

Το παράθυρο αυτό δίνει την δυνατότητα στο χρήστη να δημιουργήσει ένα Project δίνοντας του ένα όνομα. Αποτελείται από δύο κουμπιά Create και Cancel που ορίζουν την απόφαση του χρήστη.Επίσης αποτελείται από ένα πεδίο κειμένου στο οποίο ο χρήστης θα προσδιορίζει το όνομα του Project. Παρακάτω επισυνάπτεται το σχετικό παράθυρο.

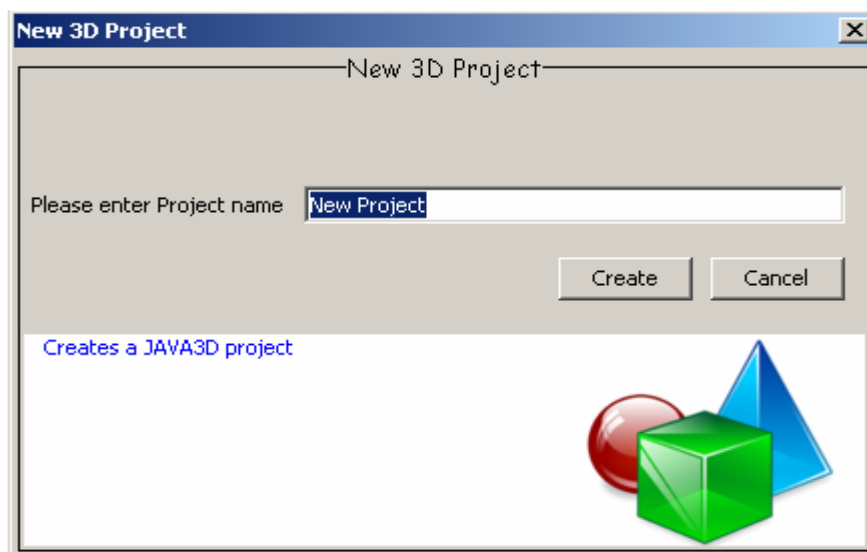


Figure 58: Παράθυρο δημιουργίας νέου Project

- **Παράθυρο δημιουργίας νέου αρχείου**

Το παράθυρο αυτό παρέχει την δυνατότητα στο χρήστη να δημιουργήσει νέα αρχεία. Αποτελείται από τέσσερα πεδία.

Το πρώτο πεδίο σου παρέχει την δυνατότητα επιλογής μεταξύ των διαθέσιμων τύπων αρχείου που υποστηρίζει η εφαρμογή. Τέτοια αρχεία είναι Java , txt , xml και τέλος ο τύπος Shape όπου με αυτήν την επιλογή ενεργοποιείται το επόμενο πεδίο που σχετίζεται με την επιλογή ενός από των διαθέσιμων Primitive αντικειμένων. Το τρίτο πεδίο σχετίζεται με την εισαγωγή του ονόματος του αρχείου και το τέταρτο πεδίο παρέχει στον χρήστη μια περιγραφή της κάθε επιλογής του , τόσο για τα Primitive αντικείμενα όσο και για τα υπόλοιπα αρχεία. Τέλος δυο κουμπιά Create και Cancel ορίζουν την απόφαση του χρήστη.

Παρακάτω παρουσιάζεται το σχετικό παράθυρο.

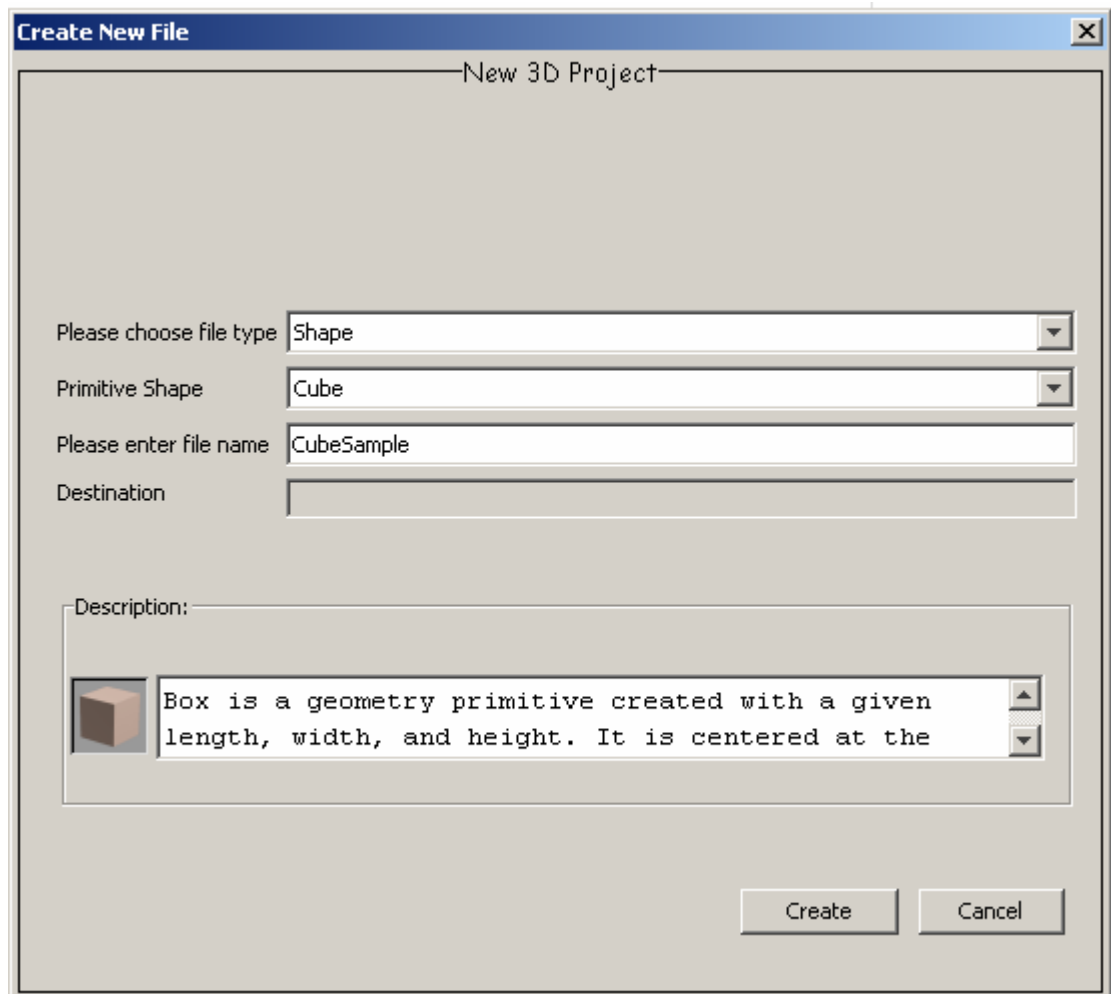


Figure 59: Παράθυρο δημιουργίας νέου αρχείου

- **Ρομπρ συστατικό επεξεργασίας κειμένου**

Το συστατικό αυτό αποτελεί ένα σημαντικό εργαλείο για την επεξεργασία κειμένου όταν ο τύπος του αρχείου που έχει επιλέξει ο χρήστης είναι Java,XML,TXT.

Ύστερα από έρευνα στο διαδίκτυο αποφασίστηκε να εμπλουτίσουμε την εφαρμογή με το εργαλείο jsyntaxpane-0.9.5-b29 το οποίο παρέχει στον Editor κειμένου της εφαρμογής την ικανότητα να αναγνωρίζει τουλάχιστον τα αρχεία Java,XML,txt υποστηρίζοντας τις συντακτικές ιδιαιτερότητες που διέπουν το καθένα.Το εργαλείο αυτό δίνει την δυνατότητα στο χρήστη να γράψει γρήγορα και εύκολα κώδικα εμπλουτίζοντας τον με διάφορα highlights και να επεξεργάζεται το κείμενο με διάφορες λειτουργίες όπως για παράδειγμα cut,copy paste,find,go to line,select all ,undo,redo,indent,unindent.Επίσης παρέχει στον χρήστη την δυνατότητα να γραφεί κώδικα χρησιμοποιώντας συντομογραφίες. Για παράδειγμα πληκτρολογώντας sout και tab εμφανίζει την εντολή System.out.println("#{p:text}");.

Τέλος το εργαλείο αυτό παρέχει την δυνατότητα εμφάνισης της αρίθμησης των γραμμών.

Παρακάτω επισυνάπτεται το σχετικό εργαλείο.

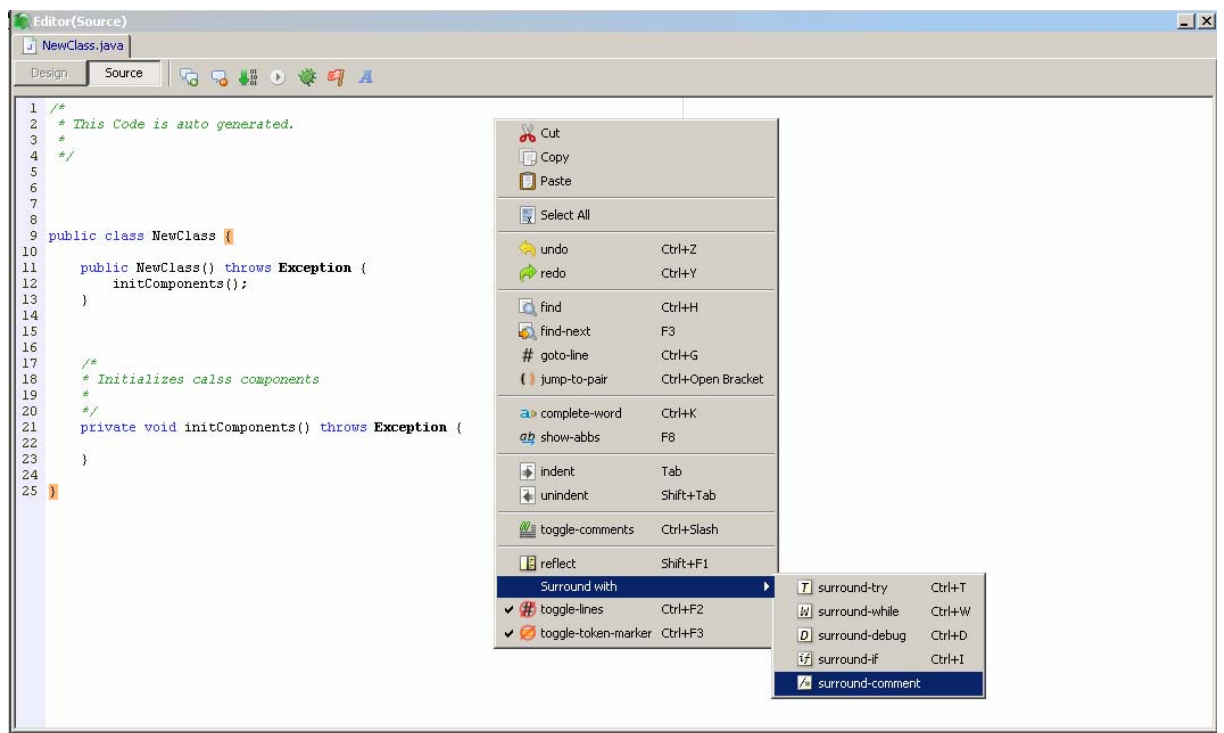


Figure 60:Editor Pane Tool

#### 4.2.6. Περιγραφή παραγόμενου κώδικα

Όπως αναφέραμε και σε προηγούμενα κεφάλαια ο editor δημιουργεί και αποθηκεύει στο workspace του χρήστη ένα αρχείο java που σχετίζεται με το αντικείμενο που δημιούργησε και επεξεργάστηκε. Το αρχείο αυτό περιέχει όλες τις πληροφορίες και τις εντολές για να το χρησιμοποιήσει ο χρήστης σε μία δικιά του εφαρμογή ώστε να αναπαράγει το αντικείμενο που έφτιαξε μέσω των εργαλείων του editor.

Παρακάτω παρουσιάζεται ο κώδικας αυτός χωρισμένος σε επιμερους τμήματα..

```
1  /*
2     * This Code is auto generated.
3  */
4  import com.sun.j3d.utils.behaviors.mouse.*;
5  import com.sun.j3d.utils.image.TextureLoader;
6  import com.sun.j3d.utils.geometry.*;
7  import com.sun.j3d.utils.universe.SimpleUniverse;
8  import com.sun.j3d.utils.universe.ViewingPlatform;
9  import java.awt.Color;
10 import java.awt.GraphicsConfiguration;
11 import javax.media.j3d.*;
12 import javax.vecmath.*;
13
14
15 public class NewShape {
16
17     private Canvas3D moduleCanvas;
18     private SimpleUniverse moduleUniverse;
19     private ViewingPlatform viewingPlatform;
20     private View moduleView;
21     private BranchGroup moduleBranchGroup;
22     private ColoringAttributes coloringAttributes;
23     private Background moduleBackground;
24     private Material    moduleMaterial;
25     private TransformGroup moduleScaleTG;
26     private TransformGroup moduleObjectTG;
27     private TransformGroup moduleShapeTG;
28     private TextureAttributes moduleTextureAttributes;
29     private Appearance appearance;
30     private AmbientLight moduleAmbientLight;
31     private MouseTranslate moduleMouseTranslateBehavior;
32     private TransformGroup moduleRotatorTG;
33     private Alpha moduleInterpolatorAlpha;
34     private RotationInterpolator moduleRotatorInterpolator;
35     private DirectionalLight moduleDirectionalLight;
36     private TextureLoader textureLoader;
37     private Texture2D moduleTexture2D;
38
39
40     public NewShape() throws Exception {
41         initComponents();
42     }
```

```

46      /*
47      * Initializes calss components
48      */
49      private void initComponents() throws Exception {
50
51          GraphicsConfiguration config = SimpleUniverse.getPreferredConfiguration();
52          moduleCanvas = new Canvas3D(config);
53          moduleCanvas.setBackground(Color.BLACK);
54          moduleUniverse = new SimpleUniverse(moduleCanvas);
55          moduleView = moduleUniverse.getViewer().getView();
56          moduleView.setFrontClipDistance(0.01f);
57          viewingPlatform = moduleUniverse.getViewingPlatform();
58          viewingPlatform.setNominalViewingTransform();
59          moduleBranchGroup = new BranchGroup();
60          moduleBranchGroup.setCapability(BranchGroup.ALLOW_DETACH);
61
62
63          //Initialize Coloring Attributes
64          coloringAttributes = new ColoringAttributes();
65          coloringAttributes.setColor(new Color3f(1.0f, 0.0f, 0.0f));
66          coloringAttributes.setShadeModel(3);
67
68          //Initialize Material Attributes
69          moduleMaterial = new Material();
70          moduleMaterial.setEmissiveColor(new Color3f(0.0f, 0.0f, 0.0f));
71          moduleMaterial.setSpecularColor(new Color3f(1.0f, 1.0f, 1.0f));
72          moduleMaterial.setAmbientColor(new Color3f(1.0f, 0.0f, 0.0f));
73          moduleMaterial.setDiffuseColor(new Color3f(1.0f, 0.0f, 0.0f));
74          moduleMaterial.setShininess(20.0f);
75          moduleMaterial.setLightingEnable(true);
76
77          //Initialize Texture Image
78          textureLoader = new TextureLoader("C:\\Water lilies.jpg", null);
79          moduleTexture2D = (Texture2D)textureLoader.getTexture();
80          moduleTexture2D.setBoundaryColor(new Color4f(0.0f, 0.0f, 0.0f, 1.0f));
81          moduleTexture2D.setMagFilter(2);
82          moduleTexture2D.setMinFilter(2);
83          moduleTexture2D.setBoundaryModeS(3);
84          moduleTexture2D.setBoundaryModeT(3);
85          moduleTexture2D.setEnable(true);

```

```

88 //Initialize Texture Attributes
89 moduleTextureAttributes = new TextureAttributes();
90 moduleTextureAttributes.setTextureBlendColor(new Color4f(0.0f, 0.0f, 0.0f, 1.0f));
91 moduleTextureAttributes.setTextureMode(4);
92 moduleTextureAttributes.setPerspectiveCorrectionMode(1);
93
94 //Initialize Appearance Module
95 appearance = new Appearance();
96 appearance.setColoringAttributes(coloringAttributes);
97 appearance.setTextureAttributes(moduleTextureAttributes);
98 appearance.setMaterial(moduleMaterial);
99 appearance.setTexture(moduleTexture2D);
100
101
102 moduleScaleTG = new TransformGroup();
103 Transform3D scaleTG3D = new Transform3D();
104 scaleTG3D.set(1 / 1.0f);
105 moduleScaleTG.setTransform(scaleTG3D);
106 moduleBranchGroup.addChild(moduleScaleTG);
107
108 moduleObjectTG = new TransformGroup();
109 moduleScaleTG.addChild(moduleObjectTG);
110 moduleObjectTG.setCapability(TransformGroup.ALLOW_TRANSFORM_READ);
111 moduleObjectTG.setCapability(TransformGroup.ALLOW_TRANSFORM_WRITE);
112 moduleShapeTG = new TransformGroup();
113 moduleShapeTG.setCapability(TransformGroup.ALLOW_TRANSFORM_WRITE);
114 moduleShapeTG.setCapability(TransformGroup.ALLOW_TRANSFORM_READ);
115 moduleShapeTG.addChild(new Cylinder(0.13f,0.13f,
116 Cylinder.GENERATE_NORMALS|Cylinder.GENERATE_TEXTURE_COORDS , 200, 200, appearance));
117
118 Transform3D transform3D = new Transform3D();
119 Vector3f translationVector3f = new Vector3f(0.0f, 0.0f, 0.0f);
120 Vector3d scaleVector3d = new Vector3d(1.0f, 1.0f, 1.0f);
121 AxisAngle4f axisAngle4f = new AxisAngle4f(1.0f, 0.0f, 0.0f, 0.0f);
122 transform3D.set(translationVector3f);
123 transform3D.setScale(scaleVector3d);
124 transform3D.setRotation(axisAngle4f);
125 moduleShapeTG.setTransform(transform3D);

```



```

118 Transform3D transform3D = new Transform3D();
119 Vector3f translationVector3f = new Vector3f(0.0f, 0.0f, 0.0f);
120 Vector3d scaleVector3d = new Vector3d(1.0f, 1.0f, 1.0f);
121 AxisAngle4f axisAngle4f = new AxisAngle4f(1.0f, 0.0f, 0.0f, 0.0f);
122 transform3D.set(translationVector3f);
123 transform3D.setScale(scaleVector3d);
124 transform3D.setRotation(axisAngle4f);
125 moduleShapeTG.setTransform(transform3D);
126
127 moduleRotatorTG = new TransformGroup();
128 moduleRotatorTG.setCapability(TransformGroup.ALLOW_TRANSFORM_READ);
129 moduleRotatorTG.setCapability(TransformGroup.ALLOW_TRANSFORM_WRITE);
130 BoundingSphere bounds = new BoundingSphere(new Point3d(0.0, 0.0, 0.0),100.0);
131
132 //Initialize transform for Interpolator Transform Axis
133 Transform3D moduleRotatorTrans3d = new Transform3D();
134 Vector3f interpolatorTranslatVec = new Vector3f(0.0f, 0.0f, 0.0f);
135 Vector3f interpolatorAxisVector = new Vector3f(0.0f, 0.0f, -1.0f);
136 AxisAngle4f InterpolatorAxisAngle = new AxisAngle4f(interpolatorAxisVector , 1.571f);
137
138 moduleRotatorTrans3d.set(InterpolatorAxisAngle);
139 moduleRotatorTrans3d.setTranslation(interpolatorTranslatVec);
140
141
142
143 //Initialize Alpha
144 moduleInterpolatorAlpha = new Alpha();
145 moduleInterpolatorAlpha.setLoopCount(-1);
146 moduleInterpolatorAlpha.setMode(1);
147 moduleInterpolatorAlpha.setAlphaAtOneDuration(0);
148 moduleInterpolatorAlpha.setAlphaAtZeroDuration(0);
149 moduleInterpolatorAlpha.setDecreasingAlphaDuration(0);
150 moduleInterpolatorAlpha.setDecreasingAlphaRampDuration(0);
151 moduleInterpolatorAlpha.setIncreasingAlphaDuration(4000);
152 moduleInterpolatorAlpha.setIncreasingAlphaRampDuration(0);
153 moduleInterpolatorAlpha.setPhaseDelayDuration(0);
154 moduleInterpolatorAlpha.setTriggerTime(0);
155
156 //Initialize Rotation Interpolator
157 moduleRotatorInterpolator = new RotationInterpolator(moduleInterpolatorAlpha, moduleRotatorTG);
158 moduleRotatorInterpolator.setTransformAxis(moduleRotatorTrans3d);
159 moduleRotatorInterpolator.setMaximumAngle(6.2831855f);
160 moduleRotatorInterpolator.setMinimumAngle(0.0f);
161 moduleRotatorInterpolator.setSchedulingBounds(bounds);
162 moduleRotatorInterpolator.setEnabled(true);
163 moduleRotatorTG.addChild(moduleRotatorInterpolator);
164 moduleRotatorTG.addChild(moduleShapeTG);
165 moduleObjectTG.addChild(moduleRotatorTG);
166
167
168 //Initialize Mouse Translate Behavior
169 moduleMouseTranslateBehavior = new MouseTranslate();
170 moduleMouseTranslateBehavior.setTransformGroup(moduleObjectTG);
171 moduleMouseTranslateBehavior.setSchedulingBounds(new BoundingSphere());
172 moduleMouseTranslateBehavior.setFactor(0.02d , 0.02d);
173 moduleBranchGroup.addChild(moduleMouseTranslateBehavior);
174
175 //Initialize Module Background
176 moduleBackground = new Background();
177 moduleBackground.setApplicationBounds(new BoundingSphere(new Point3d(), 100.0f));
178 moduleBackground.setColor(new Color3f(0.0f, 0.0f, 0.0f));
179
180
181 //Initialize Module Light
182 moduleDirectionalLight = new DirectionalLight();
183 moduleDirectionalLight.setDirection(new Vector3f(0.1f, 0.1f, -1.1f));
184 moduleDirectionalLight.setColor(new Color3f(1.0f, 1.0f, 1.0f));
185 moduleDirectionalLight.setInfluencingBounds(new BoundingSphere(new Point3d(), 1000.0f));
186 moduleBranchGroup.addChild(moduleDirectionalLight);
187
188 moduleBranchGroup.compile();
189 moduleUniverse.addBranchGraph(moduleBranchGroup);

```

Από την γραμμή 1 μέχρι την γραμμή 12 γίνονται import όλα τα πακέτα του Java3d που πρόκειται να χρησιμοποιήσουμε.

Από την γραμμή 17 έως 37 δηλώνονται όλες οι μεταβλητές και ο τύπος των αντικειμένων που πρόκειται να χρησιμοποιήσουμε.

Από την γραμμή 51 μέχρι την γραμμή 60 στήνεται το SimpleUniverse καθώς και δημιουργείται ο καμβάς που πρόκειται να αποδώσει το τρισδιάστατο αντικείμενο. Επίσης δημιουργείται και ο γράφος σκηνης(γραμμή 59) στον οποίο θα ενσωματωθούν όλα τα αντικείμενα που περιβάλλουν το τρισδιάστατο αντικείμενο.

Από την γραμμή 64 μέχρι την γραμμή 66 ορίζεται το χρώμα του αντικειμένου και το μοντέλο σκίασης.

Από την γραμμή 69 μέχρι την γραμμή 75 ορίζονται τα χρώματα που αποτελούν την εμφάνιση της επιφάνειας του αντικειμένου. Επίσης ορίζεται η φωτεινότητα και αν ο φωτισμός είναι ενεργοποιημένος ή απενεργοποιημένος. Οι τιμές για αυτές τις παραμέτρους ορίστηκαν από το αντίστοιχο εργαλείο Material Attributes.

Από την γραμμή 78 μέχρι την γραμμή 85 στήνεται το Texture Image και φορτώνεται η σχετική εικόνα. Επίσης δίνονται τιμές στις παραμέτρους του Texture. Οι τιμές αυτές προέρχονται από το εργαλείο Texture Image.

Από την γραμμή 89 μέχρι την γραμμή 92 δημιουργούνται οι ιδιότητες του Texture οι οποίες προέρχονται από το εργαλείο Texture Attributes.

Από την γραμμή 95 μέχρι την γραμμή 99 δημιουργείται η εμφάνιση του αντικειμένου και αναθέτονται στην εμφάνιση αυτή όλα τα παραπάνω.

Από την γραμμή 108 μέχρι την γραμμή 116 δημιουργείται ένα αντικείμενο TransformGroup το οποίο τοποθετείται στο γράφο σκηνης και το οποίο θα ορίζει το τρισδιάστατο αντικείμενο που είναι ένας κύλινδρος.

Από την γραμμή 118 μέχρι την γραμμή 125 ορίζεται ένα Transform3D το οποίο παίρνει τις σχετικές τιμές για την θέση τον προσανατολισμό και την κλίμακα του αντικειμένου από το εργαλείο Transform Explorer.

Από την 167 μέχρι την 125 ορίζεται και αρχικοποιείται ένα TransformGroup για την περιστρεφόμενη κίνηση RotatorInterpolator καθώς και ορίζονται η τιμές του Alpha από το σχετικό εργαλείο RotatorInterpolator Attributes.

Από την γραμμή 169 μέχρι την γραμμή 173 ορίζονται οι εντολές για τον εμπλουτισμό του τρισδιάστατου αντικειμένου με διαδραστικές μεθόδους και συγκεκριμένα με την MouseTranslate.

Από την γραμμή 175 μέχρι την γραμμή 186 ορίζεται το χρώμα του φόντου και ο φωτισμός.

Από την γραμμή 188 μέχρι την γραμμή 189 γίνεται compile ο γράφος σκηνης και τοποθετείται στο SimpleUniverse για να αποδοθεί.

## 5. Συμπεράσματα

Κύριο στοιχείο της εφαρμογής αυτής ήταν η ανάπτυξη και επεξεργασία τρισδιάστατων αντικειμένων με σκοπό την αυτόματη παραγωγή του κώδικα που αντιπροσωπεύει το αντικείμενο. Ο Editor παράγει κώδικα ο οποίος αποτελεί ένα module με σκοπό να χρησιμοποιηθεί από τον χρήστη σε περαιτέρω εφαρμογές. Για την επίτευξη των παραπάνω χρησιμοποιήθηκαν τεχνολογίες της Java και συγκεκριμένα η βιβλιοθήκη Swing η οποία σχετίζεται με την δημιουργία των γραφικών συστατικών του αντικειμένου. Επίσης για την δημιουργία των τρισδιάστατων αντικειμένων χρησιμοποιήθηκε το Java3D API. Για την παραγωγή του αυτόματου κώδικα χρησιμοποιήθηκε η τεχνολογία Freemarker Template. Τέλος εφαρμόστηκαν διάφορα Design Patterns όπως το MVC και το Singleton. Για την ανάπτυξη της εφαρμογής έγινε έρευνα σε παρόμοιες εφαρμογές και υλοποιήθηκαν πολλαπλοί έλεγχοι στην συμπεριφορά και στην επεξεργασία των τρισδιάστατων αντικειμένων.

Στην ανάλυση και καταγραφή των απαιτήσεων του λογισμικού έγιναν πολλαπλοί έλεγχοι ώστε ο editor να είναι εύχρηστος και να παρέχει στον χρήστη γρήγορη και ακριβής επεξεργασία των τρισδιάστατων αντικειμένων.

Επίσης ο παραγόμενος κώδικας είναι έτοιμος προς χρησιμοποίηση χωρίς ο χρήστης να κάνει καμία αλλαγή. Επίσης είναι οργανωμένος με τέτοιο τρόπο ώστε να μπορεί να συντηρηθεί και να εμπλουτιστεί με περισσότερα στοιχεία.

Το API ης Java3D προσφέρει στον προγραμματιστή την ευχέρεια να αναπτύξει τρισδιάστατους κόσμους χρησιμοποιώντας κατευθείαν το source code του API. Αυτό δίνει την ικανότητα στον προγραμματιστή να κατανοήσει καλύτερα την τεχνολογία και να μπορεί να παρέβη οποτεδήποτε και οπουδήποτε στο αντικείμενο. Επίσης το API είναι στηριγμένο στην Java κάτι που του επιτρέπει να συνεργαστεί και με άλλες υπάρχουσες εφαρμογές.

Τέλος η βιβλιοθήκη του Swing παρέχει στον προγραμματιστή πληθώρα συστατικών για την ανάπτυξη γραφικών εφαρμογών υψηλού επιπέδου.

## 6. Βιβλιογραφία

Μέρος υλικού που παρουσιάστηκε στην πτυχιακή αντλήθηκε από τις παρακάτω πηγές.

Γιώργος Διακέας / Εισαγωγή στην Java 2 / Κλειδάριθμος / Αθήνα 2003

H.M Deitel , P.J Deitel , S.E Santry / Advanced Java 2 Platform How to program / Prentice Hall / New Jersey 2002

Daniel Selman / Java 3D Programming / Manning

<http://freemarker.sourceforge.net>

<http://download.oracle.com>

<http://en.wikipedia.org>

<http://code.google.com/p/jsyntaxpane>

