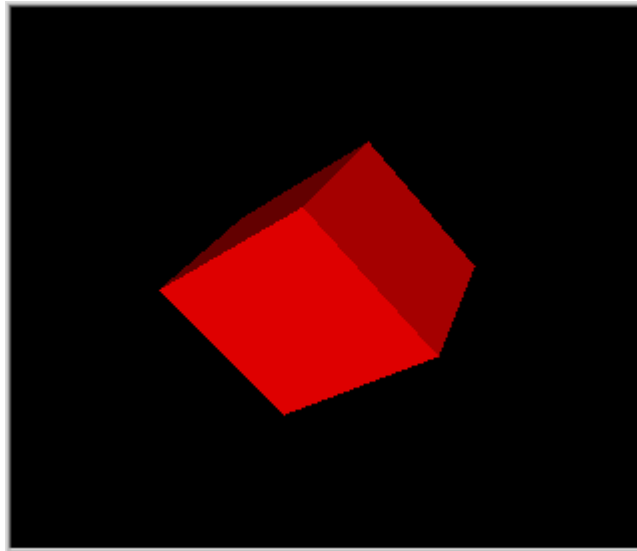


ΠΡΟΓΡΑΜΜΑ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ
ΤΜΗΜΑΤΟΣ ΕΦΑΡΜΟΣΜΕΝΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ &
ΠΟΛΥΜΕΣΩΝ

Development of Interactive 3D objects for data visualization



**Φοιτητές : Κουλεντιανός Δημήτριος AM : 947
Βγενόπουλος Ιωάννης AM : 1134**

Επιβλέπων Καθηγητής : Νικόλαος Βιδάκης

Περίληψη

Ο σκοπός της πτυχιακής αυτής , είναι η μελέτη και αξιοποίηση του Java3D API , για την ανάπτυξη μιας εφαρμογής η οποία χρησιμοποιεί την τεχνολογία αυτή , σε συνδυασμό με τη Java Swing και MySQL Data Bases , για την παρουσίαση 3D αντικειμένων και αλληλεπίδραση με αυτά .Κατά τη διάρκεια της μελέτης αυτής δημιουργήθηκε η ανάγκη χρησιμοποίησης ήδη υπαρχόντων αντικειμένων σε συνδυασμό με νέα αντικείμενα , για τη κατασκευή ενός περιβάλλοντος παρουσίασης τους .

Πιο συγκεκριμένα σε προγραμματιστικό επίπεδο στόχος μας είναι να συνδυάσουμε τη Java3D τη Java Swing και τις βάσεις δεδομένων σε MySQL , σε μία εφαρμογή η οποία έχει πλουραλισμό αντικειμένων , μεθόδων τόσο σε Java3D όσο και σε Java Swing όλα αυτά συνδεδεμένα με τη MySQL DataBase σαν μια πηγή δεδομένων.

Για την επίτευξη του στόχου αυτού , τα αντικείμενα που θα παρουσιάζει η εφαρμογή είναι κάποια βιβλία σε 3D μορφή τα οποία έχουμε σχεδιάσει όπως θα δούμε και παρακάτω. Επιλέξαμε κάτι προσιτό και χρηστικό για οποιονδήποτε , που αργότερα μπορεί να αναπτυχθεί πολύ περισσότερο. Ο χρήστης μπορεί να αλληλεπιδράσει με τα βιβλία με διάφορους τρόπους όπως πλοήγηση , άνοιγμα του PDF του κάθε βιβλίου και άλλα.

Abstract

The aim of this project , is the study and exploitation of Java3D API, for the development of an application which uses this technology, in combination with the Java Swing and MySQL Data Bases, for the 3D presentation of objects and interaction with them. At the duration of this study showed the need of utilization of already existing objects in combination with new ones, for the manufacture of their environment of presentation.

More concretely in programming level , the objective is to combine Java3D , Java Swing and MySQL databases, in an application which has pluralism of objects, methods so in Java3D as in Java Swing , and all these connected with the MySQL DataBase as a source of data.

For the achievement of this objective, the objects that will present the application are certain books in 3D form which we have designed as we will see below. Furthermore we selected something accessible and utilitarian for anyone, that can be developed wider. The user can interact with the books with various ways as piloting, opening the PDF of each book and other.

Πίνακας Περιεχομένων

Κεφάλαιο 1 Java3D API :	
1.1 Εισαγωγή στο 1 Java3D API	6
1.2 Scene Graph	6
1.2.1 Γενικά.....	7
1.2.2 Ιεραρχία κλάσεων.....	8
1.2.3 Simple Scene Graph.....	10
1.2.4 Simple Universe.....	11
1.3 Geometry.....	11
1.3.1 Shape3D	13
1.3.2 Node Components.....	14
1.3.3 Geometry Utility Classes	14
1.3.4 Geometry Classes.....	15
1.3.5 Appearance & Attributes.....	16
1.4 Background	18
1.4.2 Text2D	19
1.4.3 Text3D	20
1.5 Lights.....	22
1.5.1 Light types.....	22
1.6 Loaders.....	24
1.6.1 Γενικά.....	24
1.6.2 Loader Class.....	24
1.6.3 Behavior.....	25
1.6.4 Animation.....	28
1.6.7 Textures.....	30
1.7.1 Texturing.....	30
1.7.2 Texturing Creation.....	31
1.7.3 Technology Overview.....	34
Κεφάλαιο 2 Java Swing :	
2.1 Top-Level Swing Containers και Swing Components.....	38
2.2 Event Handlers.....	39
2.3 Swing και Threads.....	40
Κεφάλαιο 3 MySQL :	
3.1 Τι είναι η MySQL βάση δεδομένων.....	41
3.2 Οφέλη MySQL.....	43
3.3 Μειονεκτήματα MySQL.....	44
3.4 Χρήση της MySQL.....	44
Κεφάλαιο 4 : Application	
4.1 Πρόλογος.....	47
4.2 Tools.....	48
4.3 Application.....	51
4.3.1 Installation.....	51
4.3.2 Application Analysis.....	54
4.3.2.1 Panel 3D.....	54
4.3.2.1.1 Appearance Part.....	54
4.3.2.1.2 Functional Part.....	61
4.3.2.2 PDF Panel.....	75

4.3.2.3 My DataBase.....	78
4.3.2.4 ScreenShots.....	80
Κεφάλαιο 5 Βιβλιογραφία :	
5.1 E-Books	82
5.1 Sites.....	82

Κεφάλαιο 1^ο

ΤΟ JAVA3D API

Η Java στα γενικότερα πλαίσια ανάπτυξης και υποστήριξης νέων τεχνολογιών κινήθηκε προς την δημιουργία επιπλέον βιβλιοθηκών για τον εμπλουτισμό των εφαρμογών που μπορούν να αναπτυχθούν με την τεχνολογία Java . Στα πλαίσια αυτά δημιούργησε μια οικογένεια βιβλιοθηκών που ονομάστηκε Mediarproduct και περιέχει και το Java3D API . Οπότε ουσιαστικά το Java 3D αποτελεί ένα σύνολο εντολών υψηλού επιπέδου που βρίσκονται σε πλήρη συμφωνία και συνεργασία με τα υπόλοιπα πακέτα που διαθέτει η Java .

1.1 Εισαγωγή στο Java 3D API

Το Java3D API στην ουσία αποτελεί μια ιεραρχία κλάσεων που λειτουργούν σαν μια διεπαφή μεταξύ του προγραμματιστή και ενός πολύπλοκου συστήματος φωτορεαλιστικής απόδοσης (rendering) γραφικών και ήχων του OpenGL . Παρά την λειτουργικότητα του αυτή όμως το Java3D API είναι ξεκάθαρο στην χρήση του . Η φωτορεαλιστική απόδοση γίνεται αυτόματα και μάλιστα γίνεται χρήση των νημάτων (threads) της Java ώστε να γίνεται και παράλληλα . Κάθε εφαρμογή 3D στην Java αποτελείται τουλάχιστον μερικώς από αντικείμενα κλάσεων από την ιεραρχία της Java3D . Αυτή η συλλογή αντικειμένων αποτελεί το εικονικό σύμπαν (virtual universe) το οποίο τελικά αποδίδεται φωτορεαλιστικά . Το βασικό πακέτο είναι το javax.media.j3d το οποίο αποτελείται πάνω από 100 κλάσεις και παρέχει εκατοντάδες μεθόδους και πεδία . Οι κλάσεις αυτές είναι γνωστές και ως κλάσεις πυρήνα (core classes) του Java3D API .

Επιπρόσθετα υπάρχει το πακέτο com.sun.j3d.utils. το οποίο παρέχει τις utilities classes των οποίων η χρήση μπορεί να γλυτώσει τον προγραμματιστή από γραμμές κώδικα . Οι κλάσεις αυτές χωρίζονται σε 4 κατηγορίες : τους φορτωτές περιεχομένου (content loaders), τα βοηθήματα κατασκευής σκηνογραφήματος (scene graph construction aids), τις κλάσεις γεωμετρίας (geometry classes) και τις κλάσεις αξιοπιστίας (convenience classes) . Τέλος όλα τα προγράμματα Java3D χρησιμοποιούν το πακέτο java.awt που είναι υπεύθυνο για την δημιουργία παραθύρων και το javax.vecmath που είναι υπεύθυνο για την διαχείριση μαθηματικών πράξεων με πίνακες , διανύσματα κτλ.

Το API της Java3D υποστηρίζει τρεις διαφορετικούς τύπους φωτορεαλιστικής απόδοσης , ανάλογα με τις δυνατότητες που παρέχονται στους προγραμματιστές σε κάθε περίπτωση καθώς και σύμφωνα με τον απαιτούμενο χρόνο για την ολοκλήρωση

της διαδικασίας φωτορεαλιστικής απόδοσης . Έτσι ορίζονται οι παρακάτω επιλογές rendering :

- **Immediate mode** → το οποίο επιτρέπει μεγαλύτερη ευελιξία αλλά υστερεί σε χρόνο φωτορεαλιστικής απόδοσης. Στη συγκεκριμένη περίπτωση ο προγραμματιστής μπορεί να ορίσει επακριβώς γεωμετρία ή να χρησιμοποιήσει τμήματα από γράφημα σκηνής που θέλει να κάνει render . Η επιλογή αυτή δεν εκμεταλλεύεται καθόλου το δενδρικό γράφημα σκηνής .
- **Retained mode** → παρέχοντας παρόμοια ευελιξία στην διαχείριση και ελαφρώς καλύτερη ταχύτητα rendering. Παρέχεται πλήρης έλεγχος του γραφήματος σκηνής στον προγραμματιστή επιτρέποντας την επέκταση (create node) ή σμίκρυνση (delete node) του γραφήματος σκηνής παρατηρώντας ταυτόχρονα το αποτέλεσμα. Το API σε αυτήν την περίπτωση κάνει επιπλέον βελτιστοποιήσεις στην σκηνή για να βελτιώσει την ταχύτητα φωτοαπόδοσης .
- **Compiled-Retained mode** → στην περίπτωση αυτή εκτελούνται εκτενώς αλγόριθμοι για την μέγιστη βελτιστοποίηση του αποτελέσματος και την επιτάχυνση της φωτοαπόδοσης. Αυτό επιτυγχάνεται μεταγλωττίζοντας τμήμα ή ολόκληρο το γράφημα σκηνής σε Java3D .

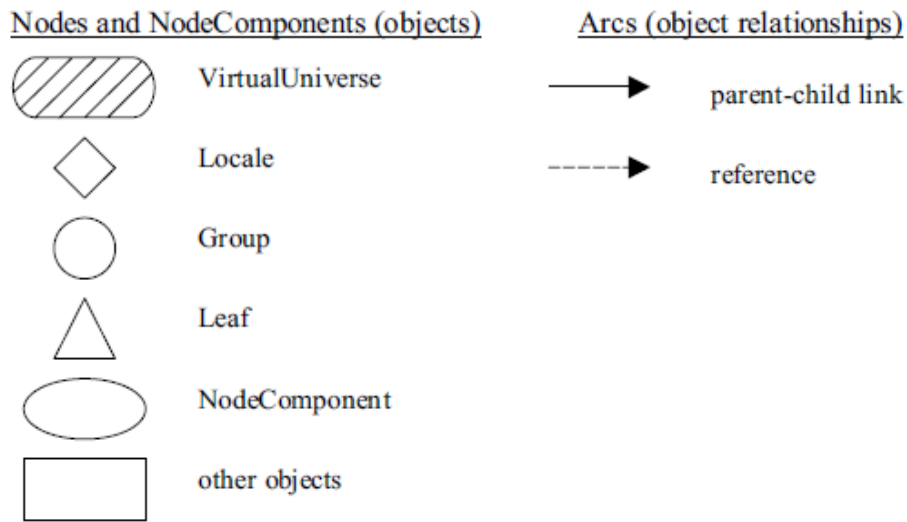
Άλλοι σημαντικοί στόχοι του Java3D API περιλαμβάνουν :

- Εκτενές σύνολο δυνατοτήτων για την δημιουργία τρισδιάστατων κόσμων, ξεκαθαρίζοντας τα κρίσιμα από τα μη κρίσιμα χαρακτηριστικά . Όσες δυνατότητες μπορούν να υποστηριχθούν σε ανώτερο επίπεδο API δεν περιλαμβάνονται στην Java3D .
- Σύνολο παραδειγμάτων αντικειμενοστραφούς προγραμματισμού υψηλού επιπέδου.
- Υποστήριξη runtime loaders . Με αυτό τον τρόπο η Java3D εξυπηρετεί ένα μεγάλο εύρος διαφορετικών file formats .

1.2 Το σκηνογράφημα (scene graph)

1.2.1 Γενικά

Το εικονικό σύμπαν δημιουργείται από το σκηνογράφημα . Το τελευταίο αποτελεί μια δενδρική δομή και περιέχει αντικείμενα κλάσεων του Java3D που καθορίζουν την γεωμετρία , τον φωτισμό , τον ήχο , την τοποθεσία , τον προσανατολισμό και την εμφάνιση των οπτικών και ηχητικών αντικειμένων. Το σκηνογράφημα αποτελείται από δυο ειδών συστατικά τους κόμβους(nodes) και τα βέλη (arcs) . Οι κόμβοι είναι τα αντικείμενα των κλάσεων της Java3D και τα βέλη δηλώνουν 2 τύπων σχέσεις μεταξύ των αντικειμένων. Την σχέση πατέρα-παιδιού και την αναφορά. Κατά την πρώτη κάθε κόμβος μπορεί να έχει πολλά παιδιά αλλά μόνο έναν πατέρα. Η αναφορά συνδέει τα συστατικά του κάθε κόμβου (node components) με αυτόν. Τα συστατικά των κόμβων ορίζουν τα χαρακτηριστικά της γεωμετρίας και εμφάνισης των τρισδιάστατων αντικειμένων που θα αποδοθούν φωτορεαλιστικά .



Εικόνα 1.1 : Τα σύμβολα στο σκηνογράφημα

Στην παραπάνω εικόνα φαίνονται τα κλασσικά σχήματα που χρησιμοποιούνται για την σχεδίαση ενός σκηνογραφήματος . Τα πρώτα δυο στην αριστερή στήλη αναπαριστούν αντικείμενα των συγκεκριμένων κλάσεων **Virtual Universe** και **Locale**, τα επόμενα αναπαριστούν αντικείμενα των κλάσεων **Group**, **Leaf**, **Node Component** και το τελευταίο αντικείμενα οποιασδήποτε άλλης κλάσης . Το βέλος δηλώνει σχέση πατέρα-παιδιού .

1.2.2 Η Ιεραρχία των κλάσεων στο Java3D API

Στην παρακάτω εικόνα φαίνεται η ιεραρχία των κλάσεων στο API . Στην κορυφή της ιεραρχίας βρίσκονται οι κλάσεις **Virtual Universe**, **Locale** και **View** αντικείμενα των οποίων περιέχονται υποχρεωτικά σε κάθε πρόγραμμα . Το υπόλοιπο κάθε σκηνογραφήματος αποτελείται από αντικείμενα της κλάσης **SceneGraphObjects** η οποία αποτελεί μια υπερκλάση για τις περισσότερες core και util κλάσεις . Υποκλάσεις αυτής είναι η **Node** και η **NodeComponent** . Τα περισσότερα αντικείμενα ενός σκηνογραφήματος είναι υποκλάσεις της **Node** η οποία είναι μια αφηρημένη (abstract) υποκλάση . Η **Node** έχει δυο υποκλάσεις την **Group** και την **Leaf** .

Η κλάση Group

Η κλάση αυτή είναι μία υπερκλάση για όλες τις κλάσεις που δηλώνουν τα χαρακτηριστικά τοποθεσίας και προσανατολισμού για τα εικονικά μας αντικείμενα . Πιο γνωστές υποκλάσεις της είναι η **TransformGroup** η οποία επιτρέπει την αλλαγή

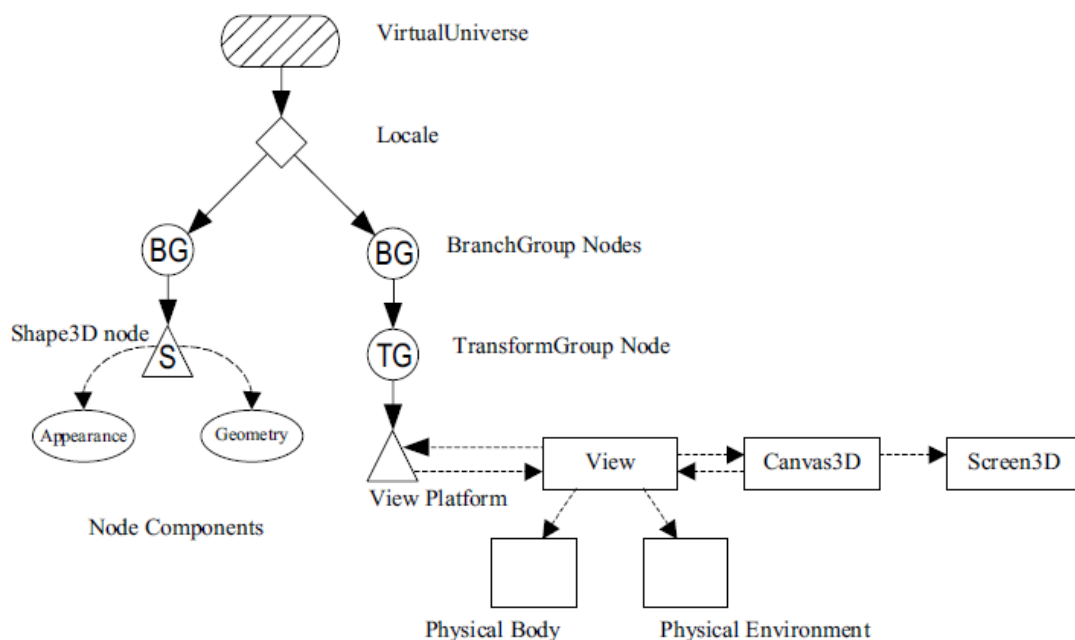
της θέσης και του προσανατολισμού των εικονικών αντικειμένων και η BranchGroup που επιτρέπει κατά την εκτέλεση να προστίθενται ή να αφαιρούνται παιδιά σε ένα κόμβο .

Η κλάση Leaf

Η κλάση αυτή είναι μια υπερκλάση για τις κλάσεις που καθορίζουν το σχήμα, την συμπεριφορά, τον ήχο, τον φωτισμό και άλλα χαρακτηριστικά των εικονικών μας αντικειμένων με πιο γνωστές υποκλάσεις την Shape3D που καθορίζει το σχήμα , την Behavior για την συμπεριφορά , την Light για τον φωτισμό και την Sound για τον ήχο . Τα αντικείμενα αυτών των κλάσεων δεν μπορούν να έχουν παιδιά αλλά μπορούν να έχουν αναφορές σε αντικείμενα NodeComponent .

Η κλάση NodeComponent

Αποτελεί μια υπερκλάση για τις κλάσεις που καθορίζουν την γεωμετρία, την εμφάνιση, το υλικό(material) και την υφή(texture) ενός αντικειμένου Shape3D . Τα αντικείμενα αυτής της κλάσης δεν αποτελούν τμήμα του σκηνογραφήματος αλλά συνδέονται με σχέση αναφοράς με τα φύλλα του . Μπορεί να συνδέονται με περισσότερα από ένα φύλλα .



Εικόνα 1.2 : Ένα τυπικό σκηνογράφημα μιας εφαρμογής Java3D

1.2.3 Δημιουργία ενός απλού σκηνογραφήματος

Στο tutorial του Java3D API της Sun αναφέρεται μια γενική συνταγή για την δημιουργία ενός απλού σκηνογραφήματος και κατά επέκταση μιας απλής εφαρμογής που αναπαριστά έναν τρισδιάστατο έγχρωμο κύβο . Αυτή αποτελείται από τους παρακάτω κανόνες :

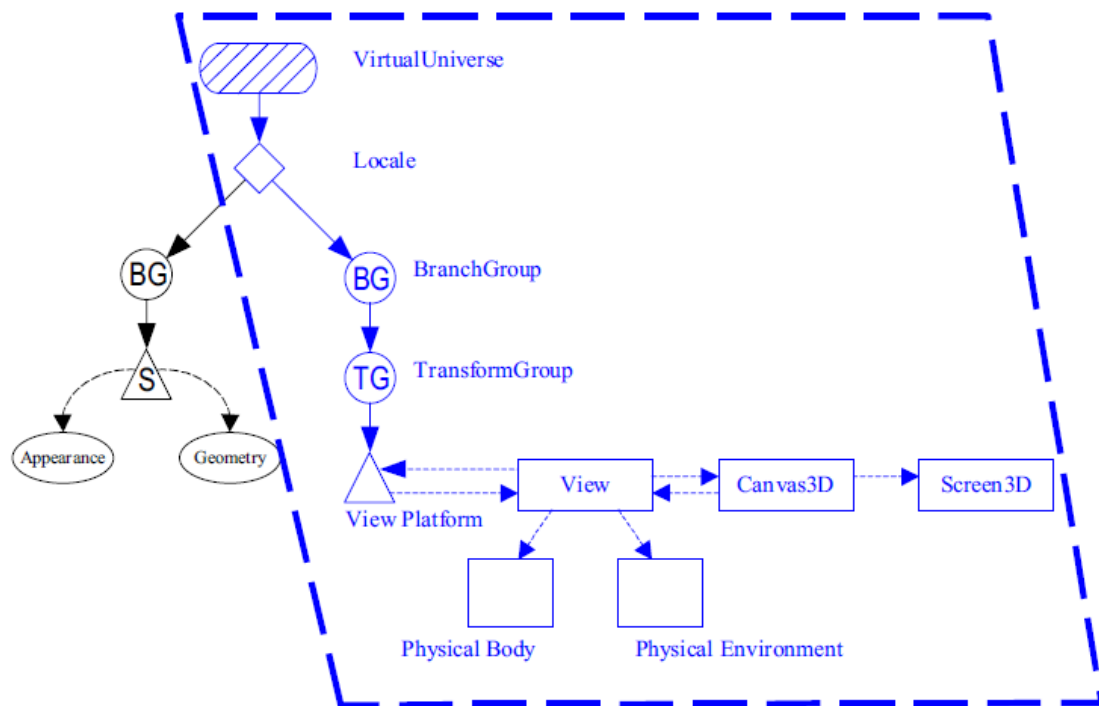
- A) Δημιούργησε ένα αντικείμενο Canvas3D.
- B) Δημιούργησε ένα αντικείμενο Virtual Universe.
- Γ) Δημιούργησε ένα αντικείμενο Locale και επικόλλησε το στο αντικείμενο Virtual Universe.
- Δ) Δημιούργησε ένα κλάδο View στο γράφημα (View Branch Graph).
 - α) Δημιούργησε ένα αντικείμενο View.
 - β) Δημιούργησε ένα αντικείμενο ViewPlatform.
 - γ) Δημιούργησε ένα αντικείμενο Physical.
 - δ) Δημιούργησε ένα αντικείμενο PhysicalEnvironment .
- E) Επικόλλησε τα αντικείμενα ViewPlatform , Physical , PhysicalEnvironment και Canvas3D στο αντικείμενο View.
- ΣΤ) Δημιούργησε κλάδους γραφήματος περιεχομένου.
- Z) Μετέφρασε (compile) όλου τους κλάδους.
- H) Εισάγετε τα υπογραφήματα (sub graphs)στο Locale.

Στην πραγματικότητα όλοι οι κανόνες της συνταγής αυτής δεν είναι απαραίτητο να τηρηθούν από τον χρήστη καθώς το API παρέχει την κλάση SimpleUniverse η οποία έρχεται να λύσει τα χέρια σε κάποιον αρχάριο και όχι μόνο . Η κλάση αυτή υλοποιεί τους κανόνες Β, Γ και Δ της συνταγής δίνοντας την άνεση του χρόνου και προσπάθειας στον χρήστη να ασχοληθεί με το περιεχόμενο . Έτσι πια η συνταγή απλοποιείται ως εξής :

- A) Δημιούργησε ένα αντικείμενο Canvas3D.
- B) Δημιούργησε ένα αντικείμενο SimpleUniverse και συνέδεσε το με αναφορά στο προηγούμενο αντικείμενο Canvas3D .Κατόπιν κάνε τις όποιες ρυθμίσεις θέλεις στο αντικείμενο της SimpleUniverse.
- Γ) Δημιούργησε κλάδους γραφήματος περιεχομένου.
- Δ) Μετέφρασε (compile) όλου τους κλάδους.
- E) Εισάγετε τα υπογραφήματα στο Locale της SimpleUniverse.

1.2.4 Η κλάση SimpleUniverse

Ένα αντικείμενο αυτής της κλάσης δημιουργεί μέσω του κατασκευαστή του ένα σκηνογράφημα το οποίο περιέχει ένα αντικείμενο της VirtualUniverse , της Locale καθώς και ένα κλάδο τύπου View στον οποίο περιέχονται οι πληροφορίες για το πώς ο θεατής βλέπει τον εικονικό κόσμο . Παρέχει δηλαδή όλα τα αναγκαία για να ξεκινήσει μια εφαρμογή . Παρακάτω παραθέτουμε το σκηνογράφημα της εφαρμογής με τον κύβο όπου τα τμήματα μέσα στο μπλε περίγραμμα είναι αυτά που έχουν δημιουργηθεί με την SimpleUniverse .



Εικόνα 1.3 : Σκηνογράφημα της εφαρμογής HelloJava . Με μπλε τα στοιχεία που δημιουργεί η SimpleUniverse .

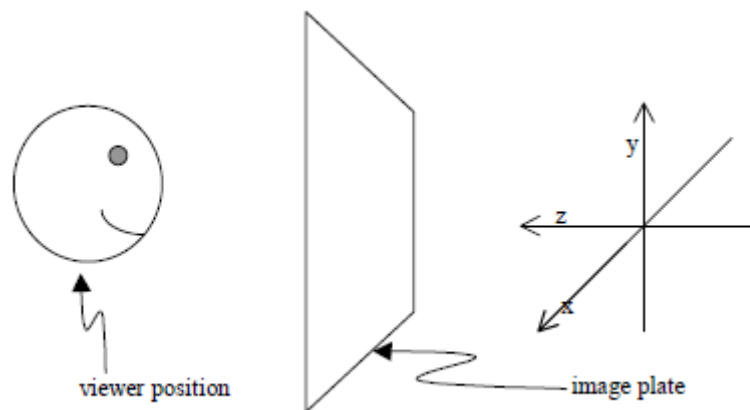
1.3 Δημιουργία της Γεωμετρίας ενός εικονικού κόσμου

Όπως αναφέραμε παραπάνω, μια περίπτωση κλάσης VirtualUniverse χρησιμεύει ως η ρίζα της scene graph σε όλα τα τρισδιάστατα προγράμματα της Java. Ο εικονικός

κόσμος virtual universe αναφέρεται συνήθως στο τρισδιάστατο virtual space όπου τα τρισδιάστατα αντικείμενα της Java υπάρχουν. Κάθε Locale αντικείμενο στον εικονικό κόσμο καθιερώνει ένα εικονικό παγκόσμιο καρτεσιανό ισότιμο σύστημα (**virtual world Cartesian Coordinate system**).

Ένα Locale αντικείμενο χρησιμεύει ως το σημείο αναφοράς για τα οπτικά αντικείμενα (visual objects) σε έναν εικονικό κόσμο (virtual universe). Με ένα Locale σε ένα SimpleUniverse, υπάρχει ένα ισότιμο σύστημα στον εικονικό κόσμο.

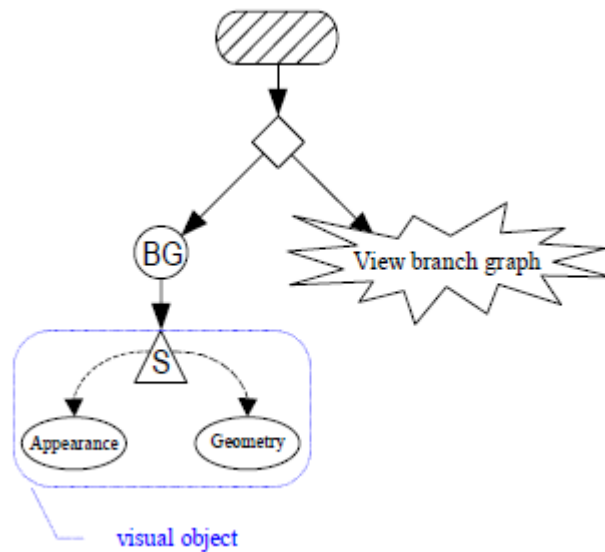
Το ισότιμο (**coordinate system**) σύστημα του τρισδιάστατου εικονικού κόσμου της Java είναι right-handed. Ο X-άξονας είναι θετικός στα δεξιά, ο Y-άξονας είναι θετικός επάνω, και ο Z-άξονας είναι θετικός προς την εμφάνιση, με όλες τις μονάδες στους μετρητές.



1.3.1 Η κλάση Shape3D

Ένα Shape3D scene graph node καθορίζει ένα οπτικό αντικείμενο (visual object). Το Shape3D είναι μια από τις υποκλάσεις της Leaf Class, επομένως, τα αντικείμενα Shape3D μπορούν μόνο να είναι φύλλα στο σκηνογράφημα. Το αντικείμενο Shape3D δεν περιέχει τις πληροφορίες για τη μορφή ή το χρώμα ενός οπτικού αντικειμένου. Αυτές οι πληροφορίες καταχωρούνται στα αντικείμενα NodeComponent αναφερόμενα στο αντικείμενο Shape3D. Ένα αντικείμενο Shape3D μπορεί να αναφερθεί σε ένα τμήμα κόμβων γεωμετρίας (*Geometry node component*) και ένα τμήμα κόμβων εμφάνισης (*Appearance node component*).

Στο παρακάτω σχήμα, το ορθογώνιο χρησιμοποιήθηκε για να αντιπροσωπεύσει το αντικείμενο ColorCube. Η απλή scene graph object στο σχήμα, εμφανίζει οπτικό αντικείμενο (visual object) που αντιπροσωπεύεται ως φύλλο Shape3D (τρίγωνο) και δύο NodeComponents (ovals) αντί του γενικού ορθογωνίου.



Ένα οπτικό αντικείμενο (visual object) μπορεί να καθοριστεί χρησιμοποιώντας ακριβώς ένα αντικείμενο Shape3D και ένα τμήμα κόμβων γεωμετρίας (**Geometry node component**) .

Προαιρετικά, το αντικείμενο Shape3D αναφέρεται σε ένα τμήμα κόμβων εμφάνισης (**Appearance node component**) επίσης. Οι constructors για Shape3D (που παρουσιάζεται στην επόμενη ομάδα δεδομένων αναφοράς) δείχνουν ότι ένα αντικείμενο Shape3D μπορεί να δημιουργηθεί χωρίς αναφορές τμημάτων κόμβων, με ακριβώς μια **Geometry node component** αναφορά, ή με τις αναφορές και στους δύο τύπους τμημάτων κόμβων.

Shape3D Constructors

- **Shape3D()**
- **Shape3D(Geometry geometry)**
- **Shape3D(Geometry geometry Appearance appearance)**

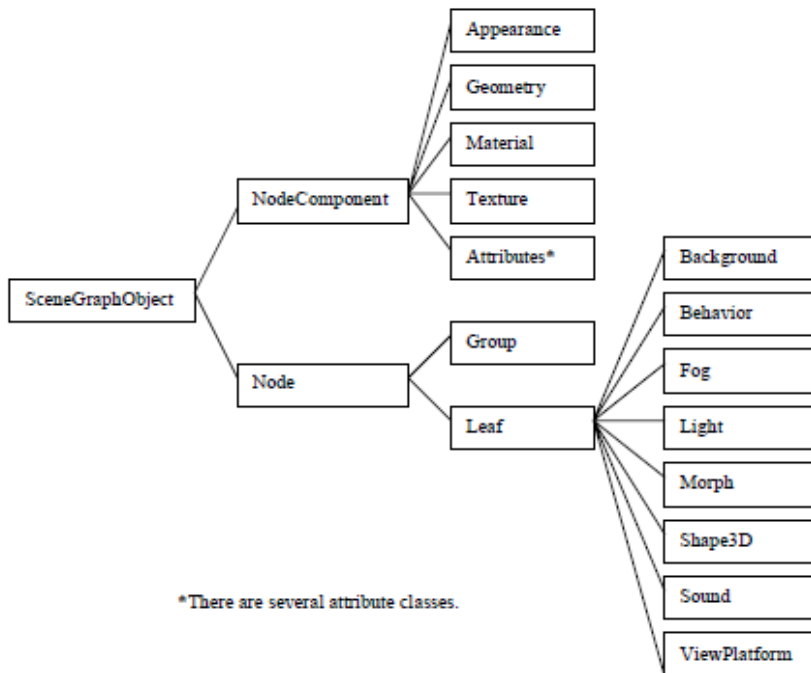
Εφ' όσον το αντικείμενο Shape3D δεν είναι live και compiled, οι τμηματικές κομβικές αναφορές μπορούν να αλλάξουν με τις μεθόδους που εμφανίζονται στην επόμενη ομάδα. Αυτές οι μέθοδοι μπορούν να χρησιμοποιηθούν στα live ή compiled αντικείμενα Shape3D εάν υπάρχει η δυνατότητα να έχουν οριστεί πρώτα.

Shape3D Methods

- **void setGeometry(Geometry geometry)**
- **void setAppearance(Appearance appearance)**

1.3.2 Node Components

Τα αντικείμενα NodeComponent περιέχουν την ακριβή προδιαγραφή των ιδιοτήτων ενός οπτικού αντικειμένου (visual object). Κάθε μια από τις διάφορες υποκλάσεις NodeComponent καθορίζει ορισμένες οπτικές ιδιότητες (visual attributes). Το σχήμα κάτω εμφανίζει μέρος της Java3D API ιεραρχίας που περιέχει την κλάση NodeComponent και τους απογόνους της.



1.3.3 Γεωμετρικές Utility κλάσεις

Αυτό το τμήμα αναφέρεται στις utility κλάσεις για τη δημιουργία του κιβωτίου (**box**), του κώνου (**cone**), του κυλίνδρου (**cylinder**), και των σφαιρών (**sphere**).

Τα γεωμετρικά στοιχεία (*primitive*) είναι ο δεύτερος ευκολότερος τρόπος να δημιουργηθεί περιεχόμενο σε έναν εικονικό κόσμο (virtual universe). Ο ευκολότερος τρόπος είναι να χρησιμοποιηθεί η κλάση ColorCube.

Οι *primitive* κλάσεις αυτές παρέχουν στον προγραμματιστή περισσότερη ευελιξία από ότι η κλάση ColorCube παρέχει.

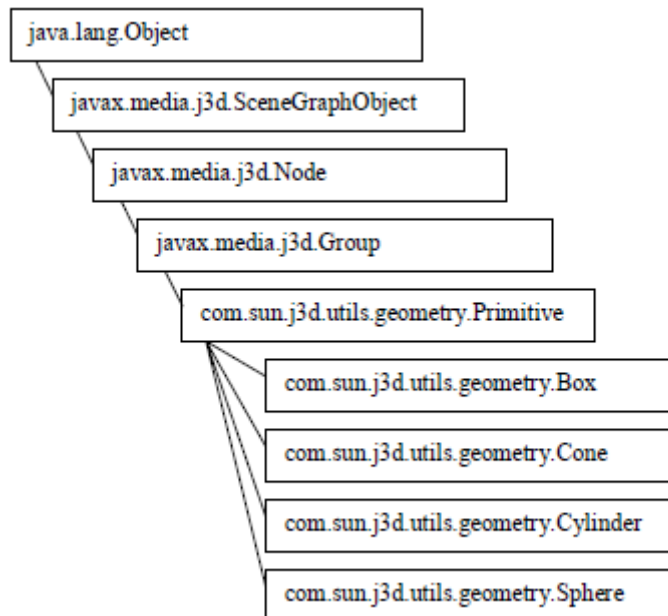
Ένα αντικείμενο ColorCube καθορίζει τη γεωμετρία και το χρώμα σε ένα Geometry node component.

Συνεπώς όλα σε ένα ColorCube καθορίζονται, εκτός από το μέγεθός του. Το μέγεθος ενός ColorCube προσδιορίζεται μόνο όταν δημιουργείται το αντικείμενο.

Ένα *primitive* αντικείμενο παρέχει περισσότερη ευελιξία με τον προσδιορισμό της μορφής χωρίς προσδιορισμό του χρώματος. Σε μια γεωμετρική primitive utility κλάση, ο προγραμματιστής δεν μπορεί να αλλάξει τη γεωμετρία (geometry), αλλά μπορεί να

αλλάζει την εμφάνιση (appearance). Οι *primitive* κλάσεις δίνουν στον προγραμματιστή την ευελιξία να έχουν τις πολλαπλές περιπτώσεις του ίδιου γεωμετρικού *primitive*.

Το κιβώτιο(Box), ο κύλινδρος(Cylinder), ο κώνος(Cone) και η σφαίρα(Sphere) utility κλάσεις, καθορίζονται από το *com.sun.j3d.utils.geometry* πακέτο.

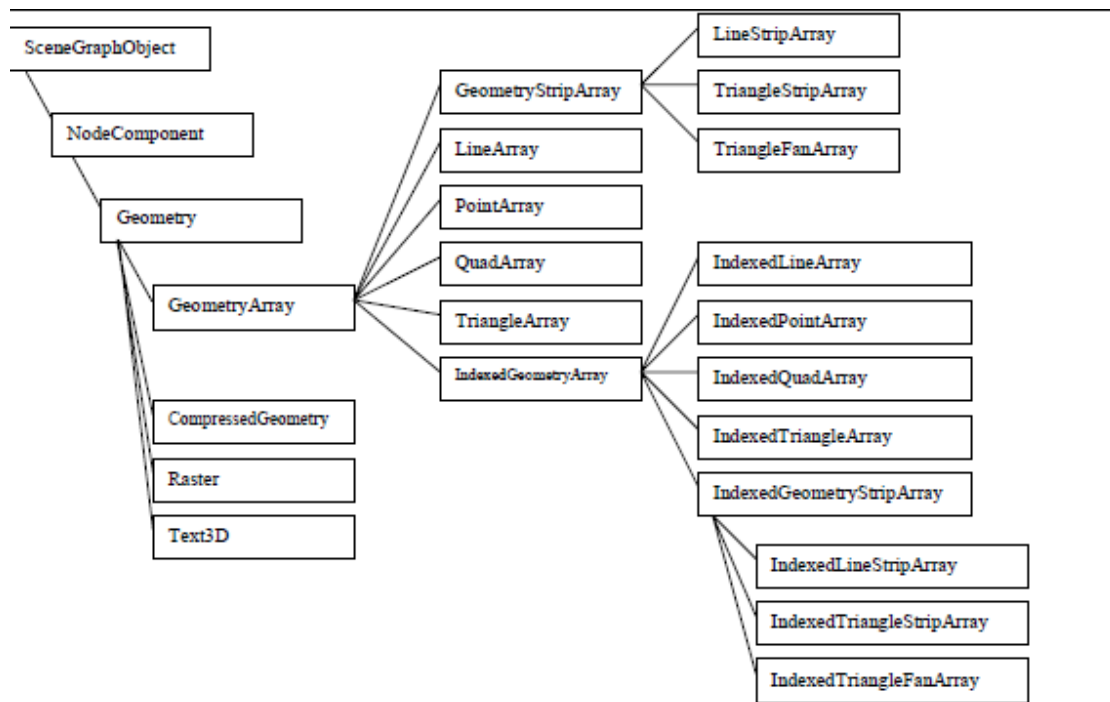


1.3.4 Γεωμετρικές κλάσεις (Geometry Classes)

Στην 3D computer graphics, όλα, από το απλούστερο τρίγωνο μέχρι το πιο περίπλοκο jumbo jet μοντέλο, διαμορφώνεται και δίνεται με τα βασισμένα στις κορυφές(vertex-based model) στοιχεία. Με την Java3D, κάθε αντικείμενο Shape3D πρέπει να καλέσει τη μέθοδό του `setGeometry()` αναφερόμενο σε ένα και μόνο ένα αντικείμενο γεωμετρίας (Geometry object).

Οι υποκλάσεις της γεωμετρίας εμπίπτουν σε τρεις ευρείες κατηγορίες:

- Η **Non-indexed vertex-based geometry** (κάθε φορά που δίνεται ένα οπτικό αντικείμενο(visual object), κάθε κορυφή μπορεί να χρησιμοποιηθεί μόνο μία φορά)
- Η **Indexed vertex-based geometry** (κάθε φορά που δίνεται ένα οπτικό αντικείμενο, οι κορυφές του μπορούν να επαναχρησιμοποιηθούν)
- **Άλλα οπτικά αντικείμενα(visual objects)** (οι κλάσεις `Raster`, `Text3D` και `CompressedGeometry`).



1.3.5 Appearances και Attributes

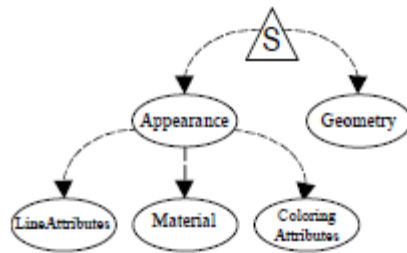
Τα Shape3D αντικείμενα μπορούν να αναφερθούν και σε ένα Geometry σε ένα Appearance αντικείμενο. Όπως συζητήθηκε προηγουμένως, το Geometry αντικείμενο προσδιορίζει τις πληροφορίες ανά-κορυφή ενός οπτικού αντικειμένου(visual object). Οι πληροφορίες ανά-κορυφή σε ένα Geometry αντικείμενο μπορούν να προσδιορίσουν το χρώμα των οπτικών αντικειμένων.

Τα στοιχεία σε ένα Geometry αντικείμενο είναι συχνά ανεπαρκή να περιγράψουν πλήρως πώς ένα αντικείμενο μοιάζει. Στις περισσότερες περιπτώσεις, απαιτείται επίσης και ένα Appearance αντικείμενο.

Ένα Appearance αντικείμενο δεν περιέχει τις πληροφορίες για το πώς το αντικείμενο Shape3D πρέπει να μοιάζει, αλλά ένα Appearance αντικείμενο ξέρει πού να βρεθούν τα στοιχεία εμφάνισης.

Ένα Appearance αντικείμενο(ήδη μια υποκλάση της NodeComponent) μπορεί να δηλώσει διάφορα αντικείμενα άλλων υποκατηγοριών της NodeComponent abstract κλάσης.

Επομένως, οι πληροφορίες που περιγράφουν την εμφάνιση ενός γεωμετρικού primitive λέγεται ότι είναι καταχωρημένες μέσα σε μια «δέσμη εμφάνισης» («**appearance bundle**») όπως αυτή εμφανίζεται παρακάτω :



Ένα Appearance αντικείμενο μπορεί να αναφερθεί σε διάφορες υποκλάσεις NodeComponent που ονομάζονται **appearance attribute objects**, περιλαμβάνοντας:

- PointAttributes
- LineAttributes
- PolygonAttributes
- ColoringAttributes
- TransparencyAttributes
- RenderingAttributes
- Material
- TextureAttributes
- Texture
- TexCoordGeneration

Attributes Class	Parameter	Default Value
ColoringAttributes	color	white (1, 1, 1)
	shade model	SHADE_GOURAUD
LineAttributes	line width	1.0
	line pattern	PATTERN_SOLID
	line antialiasing enable	false
PointAttributes	point size	1.0
	point antialiasing enable	false
PolygonAttributes	cull face	CULL_BACK
	backFaceNormalFlip	false

	polygon mode	POLYGON_FILL
	polygonOffset	0.0
	polygonOffsetFactor	0.0
RenderingAttributes	depthBufferEnable	true
	depthBufferWriteEnable	true
	alphaTestFunction	ALWAYS
	alphaTestValue	0.0
	visible	true
	ignoreVertexColors	false
	rasterOpEnable	false
	rasterOp	ROP_COPY
TextureAttributes	textureMode	REPLACE
	textureBlendColor	black (0, 0, 0, 0)
	transform	identity
	perspectiveCorrectionMode	NICEST
	textureColorTable	null
TransparencyAttributes	transparencyMode	NONE
	transparencyValue	0.0
	srcBlendFunction	SRC_BLEND_ALPHA
	dstBlendFunction	BLEND_ONE_MINUS_ALPHA

1.4 Δημιουργώντας το Φόντο

Για την δημιουργία του φόντου κάνουμε χρήση της κλάσης Background που είναι μια υποκλάση της Node. Η κλάση αυτή μας δίνει την δυνατότητα να διαλέξουμε ανάμεσα σε 3 τεχνικές , για την δημιουργία του φόντου, με χρήση ενός σταθερού χρώματος, με χρήση μιας εικόνας και με την χρήση ενός γεωμετρικού σχήματος.

1.4.1 Background

Τεχνική δημιουργίας φόντου με χρήση ενός χρώματος

Η τεχνική αυτή είναι μια απλή και με τα χειρότερα αισθητικά αποτελέσματα. Δημιουργούμε μία κλάση addBackground η οποία φτιάχνει ένα αντικείμενο Background και καλώντας τις μεθόδους της κλάσης setApplicationBounds(), setColor(int r, int g, int b) δημιουργεί το φόντο με τα δοθέντα όρια και στο χρώμα επιλέγουμε και τέλος εισάγει το αντικείμενο στο σκηνογράφημα.

Τεχνική δημιουργίας φόντου με χρήση μιας εικόνας

Κάνοντας χρήση αυτής της τεχνικής φορτώνουμε μια εικόνα ως ένα texture με ένα αντικείμενο της textureLoader η οποία είναι μια κλάση φορτωτής(loader) .

```
TextureLoader bgTexture = new TextureLoader(image,null);
```

Κατόπιν η Background μετατρέπει το Texture σε ένα ImageComponent2D.

```
Background back = new Background(bgTexture.getImage());
```

Στην συνέχεια κάνοντας χρήση της μεθόδου setImageScaleMode με την παράμετρο SCALE_FIT_MAX δίνουμε την εντολή να απλώσει η εικόνα στον μέγιστο βαθμό.

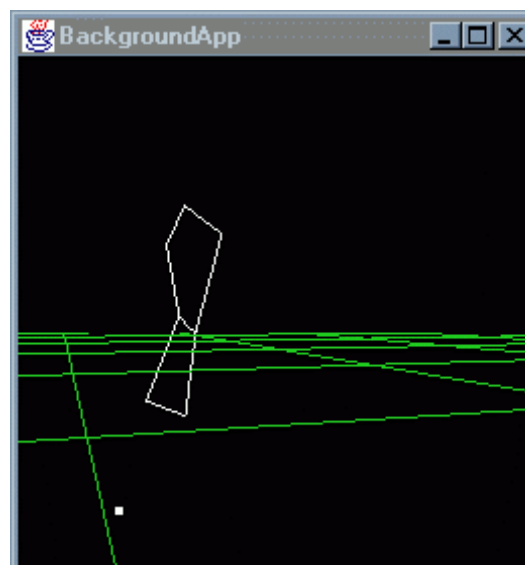
```
back.setImageScaleMode(Background.SCALE_FIT_MAX);
```

Αυτή η κίνηση θέλει προσοχή διότι μπορεί να προκαλέσει μεγάλη παραμόρφωση στην εικόνα . Θα μπορούσαμε να χρησιμοποιήσουμε και την SCALE_REPEAT η οποία παραθέτει απεικονίσεις της εικόνας κατ' επανάληψη . Η τεχνική αυτή έχει το μειονέκτημα ότι το φόντο μένει στάσιμο καθώς μετακινείται η οπτική γωνία του θεατή .

Τεχνική δημιουργίας φόντου με χρήση ενός γεωμετρικού σχήματος

Με την τεχνική αυτή δημιουργούμε ένα αντικείμενο Sphere το οποίο περικλείει όλο το σκηνικό που θα φτιάξουμε , έτσι ο θεατής είναι σαν να βρίσκεται μέσα σε μια εικονική σφαίρα .

Στην σφαίρα αυτή τοποθετούμε μια εικόνα σαν texture , αλλά την εικόνα αυτή αφού τη φορτώσουμε την περνάμε σε ένα αντικείμενο της κλάσης Appearance .



1.4.2 Text2D

Υπάρχουν δύο τρόποι να προστεθεί το κείμενο σε μια τρισδιάστατη σκηνή της Java. Ένας τρόπος χρησιμοποιήσουμε την κλάση Text2D και ο άλλος τρόπος χρησιμοποιήσουμε την κλάση Text3D. Προφανώς, μια σημαντική διαφορά είναι ότι τα αντικείμενα Text2D είναι δυσδιάστατα και τα αντικείμενα Text3D είναι

τρισεδιάστατα. Μια άλλη σημαντική διαφορά είναι πώς αυτά τα αντικείμενα δημιουργούνται.

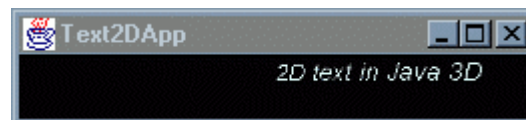
Τα Text2D αντικείμενα είναι ορθογώνια πολύγωνα με το κείμενο (text) να εφαρμόζεται ως βασική δομή (texture).

Τα Text3D αντικείμενα είναι τρισεδιάστατα γεωμετρικά αντικείμενα που δημιουργούνται ως εξώθηση του κειμένου.

Σαν υποκλάση του Shape3D, οι περιπτώσεις του Text2D μπορούν να είναι παιδιά των group objects. Για να τοποθετήσουμε ένα αντικείμενο Text2D σε μια τρισεδιάστατη σκηνή της Java, δημιουργούμε απλά το αντικείμενο Text2D και το προσθέτουμε στο σκηνογράφημα :

- 1) Δημιουργούμε ένα Text2d αντικείμενο
- 2) Το προσθέτουμε σε ένα σκηνογράφημα (scene graph)

Τα Text2D αντικείμενα εφαρμόζονται χρησιμοποιώντας ένα πολύγωνο και μια δομή(texture). Το πολύγωνο είναι διαφανές έτσι ώστε μόνο η δομή(texture) να είναι ορατή. Η δομή(texture) είναι της συμβολοσειράς κειμένων που τίθεται στον ονομασμένο χαρακτήρα(named typeface) με την προσδιορισμένη παράμετρο font. Οι χαρακτήρες που είναι διαθέσιμοι στο σύστημά , ποικίλουν. Χαρακτηριστικά, οι χαρακτήρες Courier, Helvetica, και TimesRoman είναι διαθέσιμοι μεταξύ των άλλων. Οποιοσδήποτε τύπος χαρακτήρων διαθέσιμος στο AWT, είναι διαθέσιμος σε μας για τις εφαρμογές Text2D (και Text3D).



1.4.3 Text3D

Ένας άλλος τρόπος να προστεθεί ένα κείμενο (text) σε έναν τρισεδιάστατο εικονικό κόσμο της Java είναι να δημιουργηθεί ένα αντικείμενο Text3D για το κείμενο. Όπου Text2D δημιουργεί το κείμενο με μια βασική δομή (texture), το Text3D δημιουργεί το κείμενο χρησιμοποιώντας τη γεωμετρία. Η γεωμετρία κειμένου (textual geometry) ενός αντικειμένου Text3D είναι μια εξώθηση του τύπου χαρακτήρων.

Η δημιουργία ενός αντικειμένου Text3D είναι λίγο πιο σύνθετη από ότι η δημιουργία ενός αντικειμένου Text2D. Το πρώτο βήμα είναι να δημιουργηθεί ένα αντικείμενο Font3D του επιθυμητού χαρακτήρα, του μεγέθους, και του ύφους των τύπων χαρακτήρων.

Κατόπιν ένα αντικείμενο Text3D για μια ιδιαίτερη συμβολοσειρά γίνεται χρησιμοποιώντας το αντικείμενο Font3D. Από όταν η Text3D κλάση είναι μια υποκλάση της Geometry, το Text3D αντικείμενο είναι ένα NodeComponent που σχετίζεται με ένα ή περισσότερα αντικείμενα Shape3D.

Παρακάτω είναι η διαδικασία εγγραφής των αντικειμένων Text3D σε ένα σκηνογράφημα (scene graph) :

- 1) Δημιουργούμε ένα Font3D αντικείμενο από το AWT Font.
- 2) Δημιουργούμε ένα Text3D για συμβολοσειρά χρησιμοποιώντας το αντικείμενο Font3D , προαιρετικά προσδιορίζοντας ένα σημείο αναφοράς.
- 3) Δήλωση του αντικειμένου από ένα αντικείμενο Shape3D που προστίθεται στο σκηνογράφημα (scene graph).

Το κείμενο (text) ενός αντικειμένου Text3D μπορεί να προσανατολιστεί με ποικίλους τρόπους. Ο προσανατολισμός προσδιορίζεται ως κατεύθυνση μονοπατιών. Οι επιλογές είναι δεξιά, αριστερά, επάνω, και κάτω.

Κάθε αντικείμενο Text3D έχει ένα σημείο αναφοράς. Το σημείο αναφοράς για ένα αντικείμενο Text3D είναι η προέλευση του αντικειμένου. Το σημείο αναφοράς για κάθε αντικείμενο καθορίζεται από το συνδυασμό του μονοπατιού και της ευθυγράμμισης του κειμένου.

Ο πίνακας παρακάτω παρουσιάζει τα αποτελέσματα του μονοπατιού και της προδιαγραμμένης ευθυγράμμισης στον προσανατολισμό του κειμένου και της τοποθέτησης του σημείου αναφοράς.

	ALIGN_FIRST (default)	ALIGN_CENTER	ALIGN_LAST
PATH_RIGHT (default)	●Text3D	Text3D●	Text3D●
PATH_LEFT	D3tXeT●	D3tXeT●	●D3tXeT
PATH_DOWN	●T e x t	T e ●x t	T e x ●t
PATH_UP	t x e ●T	t x ●e T	●t x e T

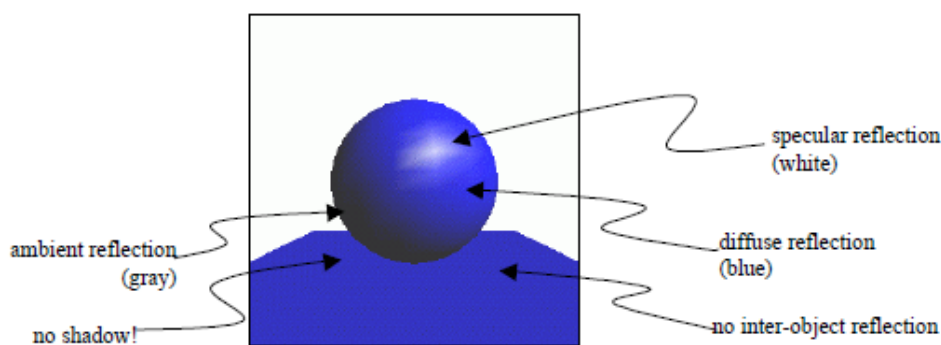
Τα αντικείμενα Text3D έχουν *normals*. Η προσθήκη μιας δέσμης εμφάνισης (*appearance bundle*) που περιλαμβάνει ένα Material αντικείμενο σε ένα αντικείμενο Shape3D αναφερόμενη στη γεωμετρία Text3D, θα επιτρέψει το φωτισμό για το αντικείμενο Text3D.

1.5 Φωτισμός του εικονικού κόσμου

Με τον όρο φωτισμός (lighting) στα τρισδιάστατα γραφικά εννοούμε στην ουσία τα ζητήματα που αφορούν την δημιουργία εικονικών πηγών φωτός και τον τρόπο με τον οποίο τα αντικείμενα ενός εικονικού κόσμου αντανακλούν το φως . Το με ποιο τρόπο αντανακλά το φως ένα εικονικό αντικείμενο επηρεάζεται από τις ιδιότητες του υλικού της επιφάνειάς του , και από την θέση και τον τύπο του εικονικού φωτισμού .

Μια βασική ιδιότητα που έχουν όλα τα τρισδιάστατα αντικείμενα είναι τα μοναδιαία διανύσματα . Τα διανύσματα αυτά προσδιορίζουν την κατεύθυνση προς την οποία αντανακλάται μια ακτίνα εικονικού φωτός που πέφτει πάνω σε κάθε σημείο του αντικειμένου . Για κάθε πλακίδιο αποθηκεύεται και ένα τέτοιο διάνυσμα . Όλα τα πλακίδια έχουν το ίδιο μοναδιαίο διάνυσμα, άρα το φως ανακλάται προς την ίδια κατεύθυνση σε οποίο σημείο του δαπέδου και να πέσει , αν το διάνυσμα είναι επίπεδο. Σε περίπτωση που έχουμε ανώμαλο έδαφος τα διανύσματα αυτά θα είχαν διαφορετική γωνία και έτσι θα μπορούσαν να δημιουργηθούν διαφορετικές ανακλάσεις και σκιές .

Ένας βασικός λόγος που χρησιμοποιούμε διάφορες τεχνικές φωτισμού , είναι για να μπορέσουμε πιο αποτελεσματικά να αποδώσουμε την αίσθηση του βάθους και τις σκιάσεις που δημιουργούνται σε ένα αντικείμενο .



1.5.1 Είδη φωτισμού στο JAVA3D API

Το μοντέλο φωτισμού στο JAVA3D API υποστηρίζει τέσσερις τύπους φωτισμού :

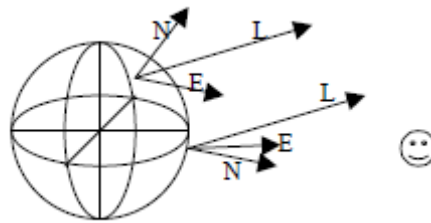
- Φωτισμός περιβάλλοντος (ambient light) :

Είναι ένας τύπος σταθερού φωτισμού χαμηλής έντασης που διαχέεται ομοιόμορφα σε όλο τον εικονικό κόσμο , χωρίς να έχει κάποια συγκεκριμένη κατεύθυνση . Οι επιφάνειες που φωτίζονται από τέτοιου είδους φωτισμό αντανακλούν προς όλες τις κατευθύνσεις .

- Κατευθυνόμενος φωτισμός (directional light) :

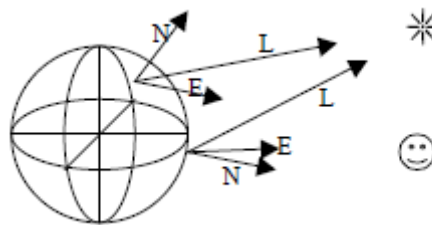
Ο φωτισμός αυτός μπορεί να περιγράψει ως άπειρες παράλληλες ακτίνες φωτός με συγκεκριμένη κατεύθυνση που ξεκινούν από το μείον άπειρο και καταλήγουν στο άπειρο . Δεν έχουν δηλαδή κάποια πηγή . Οι επιφάνειες που φωτίζονται από

τέτοιο φως αντανακλούν προς την κατεύθυνση που καθορίζεται από την γωνία που σχηματίζει η κατεύθυνση του φωτός με το κάθε σημείο , σε συνάρτηση με το μοναδιαίο διάνυσμα τους .



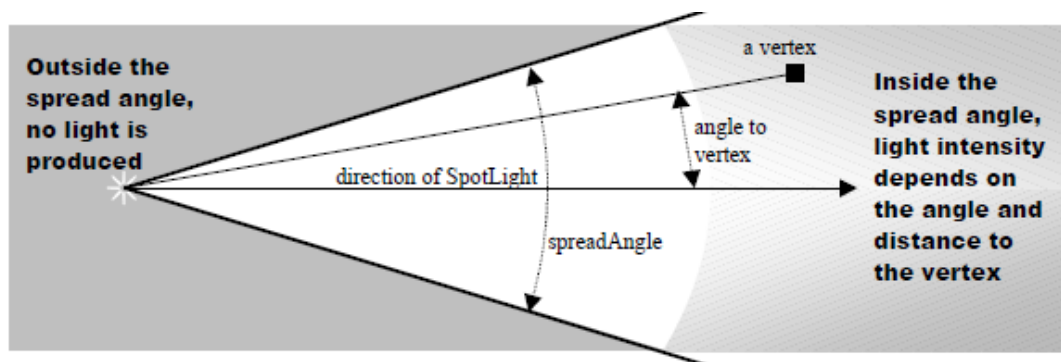
- Φωτισμός από σταθερό σημείο (Point light) :

Ο φωτισμός αυτός έχει μια σταθερή πηγή κάπου στον εικονικό κόσμο και αποστέλλει εικονικές ακτίνες προς όλες τις κατευθύνσεις όπως ο ήλιος . Επομένως μια επιφάνεια που φωτίζεται με τέτοιο φωτισμό δεν αντανακλάει προς την ίδια κατεύθυνση από όλα τα σημεία της .



- Φωτισμός προβολέα (Spot light) :

Ο φωτισμός αυτός είναι στην ουσία υποκατηγορία του προηγούμενου . Έχει μια πηγή αλλά η κατεύθυνση του περιορίζεται σε ένα πεδίο και η ένταση του εξαρτάται από τη απόσταση του αντικειμένου που φωτίζει . Αντίστοιχα με το φυσικό περιβάλλον είναι το φως που εκπέμπει ένας φακός ή ένας προβολέας .



Η κλάση υπεύθυνη για το φωτισμό είναι η Light . Υποκλάσεις αυτής είναι η AmbientLight , DirectionalLight και PointLight με υποκλάση την SpotLight .

1.6 Τεχνικές εισαγωγής τρισδιάστατων μοντέλων στην εφαρμογή

1.6.1 Γενικά

Στην αγορά κυκλοφορούν πάρα πολλά πακέτα σχεδίασης και δημιουργίας τρισδιάστατων μοντέλων , κάθε ένα από αυτά δημιουργεί τελικά ένα αρχείο σε διαφορετική μορφή και απαιτεί διαφορετική μεταχείριση για την εισαγωγή του σε μια εφαρμογή Java3D. Παρουσιάζονται παρακάτω τα πιο γνωστά πακέτα και τα είδη αρχείων που δημιουργούν :

File Format	Description
3DS	3D-Studio
COB	Caligari trueSpace
DEM	Digital Elevation Map
DXF	AutoCAD Drawing Interchange File
IOB	Imagine
LWS	Lightwave Scene Format
NFF	WorldToolKit NFF format
OBJ	Wavefront
PDB	Protein Data Bank
PLAY	PLAY
SLD	Solid Works (prt and asm files)
VRT	Superscape VRT
VTK	Visual Toolkit
WRL	Virtual Reality Modeling Language

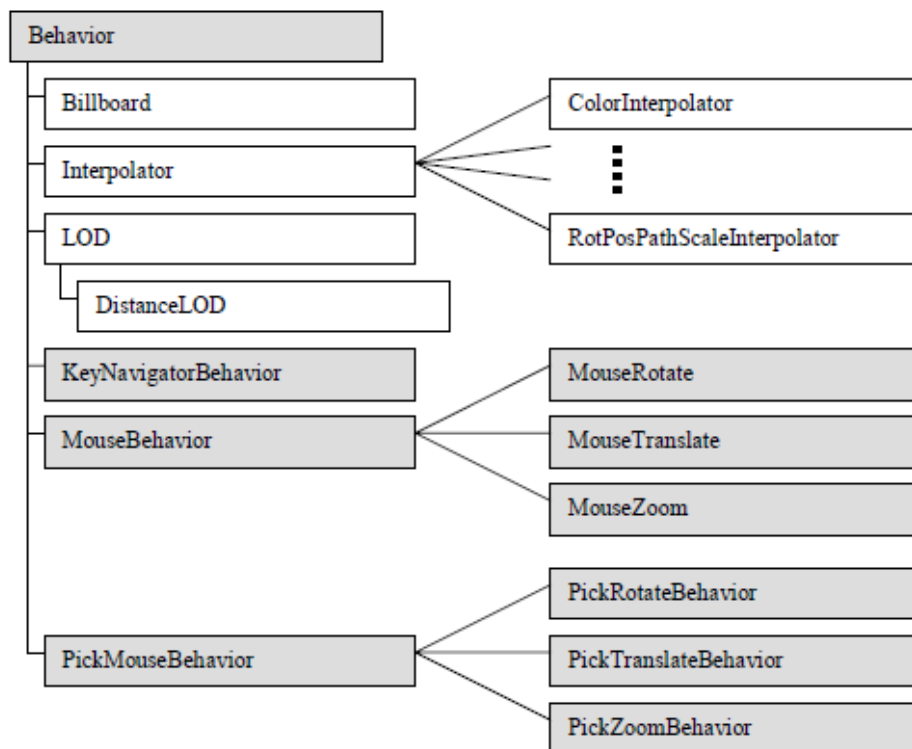
Για την εισαγωγή ενός τρισδιάστατου μοντέλου φτιαγμένου από κάποιο από τα παραπάνω μοντέλα , πρέπει να κάνουμε χρήση ενός interface της Java3D , του φορτωτή loader και της κλάσης scene .

1.6.2 Ο φορτωτής Loader

Ο φορτωτής αναλαμβάνει να διαβάσει ένα αρχείο ενός τρισδιάστατου μοντέλου και να δημιουργήσει αναπαραστάσεις Java3D επιλέγοντας τα συστατικά του μοντέλου που του έχουν ζητηθεί . Το Scene χρησιμοποιεί ένα αντικείμενο loader για να εξάγει τις πληροφορίες εκείνες που θα μπορέσουν να το καταστήσουν διαθέσιμο για αναπαράσταση σε μια εφαρμογή Java3D (κυρίως το BranchGroup του) αλλά και πληροφορίες σχετικές με τον φωτισμό , τον ήχο κτλ. Στο πακέτο Java3D utilities παρέχονται δυο υποκλάσεις της loader : η Lw3Dloader και η ObjectFile . παρέχεται επίσης και μια τρίτη υποκλάση η LoaderBase η οποία δίνει την δυνατότητα στον προγραμματιστή να δημιουργήσει τον δικό του loader .

1.6.3 Η κλάση Behavior

Οι κλάσεις **Behavior** χρησιμοποιούνται σε πολλές τρισδιάστατες εφαρμογές της Java και με πολλούς τρόπους. Είναι σημαντικό να γίνουν κατανοητά τα έργα και οι εκτιμήσεις προγραμματισμού της κλάσης **Behavior**. Αυτό το μέρος εξηγεί την κλάση **Behavior**, δίνει μια συνταγή για τον προγραμματισμό μιας συνηθισμένης κλάσης **Behavior**, και δίνει μια απλή εφαρμογή παραδείγματος χρησιμοποιώντας μια κλάση **Behavior**.



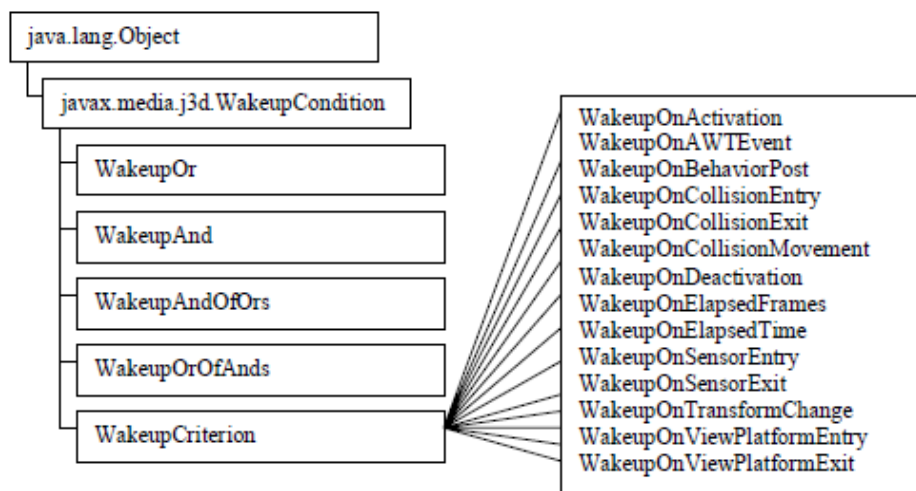
Μια συνηθισμένη κλάση **Behavior** εφαρμόζει τις μεθόδους έναρξης και processStimulus από την abstract behavior κλάση. Φυσικά, η κλάση **Behavior** έχει επίσης τουλάχιστον έναν constructor και μπορεί να έχει άλλες μεθόδους επίσης. Οι περισσότερες Behavior θα ενεργήσουν σε ένα scene graph object για να έχουν επιπτώσεις στη συμπεριφορά. Στον πίνακα πάνω, το αντικείμενο που μια **Behavior** ενεργεί επάνω αναφέρεται ως αντικείμενο της αλλαγής(object of change). Είναι μέσω αυτού του αντικειμένου, ή των αντικειμένων, όπου η **Behavior** έχει επιπτώσεις στον εικονικό κόσμο. Ενώ είναι δυνατό να υπάρξει μια **Behavior** που δεν έχει ένα object of change, όπως τα περισσότερα.

Η **Behavior** χρειάζεται μια αναφορά στο αντικείμενο αλλαγής(object of change) της για να είναι σε θέση να κάνει τις συμπεριφοριστικές αλλαγές. Ο constructor μπορεί να χρησιμοποιηθεί για να θέσει την αναφορά στο αντικείμενο της αλλαγής(object of change). Εάν όχι, μια άλλη μέθοδος στην κλάση **Behavior** πρέπει να καταχωρήσει αυτές τις πληροφορίες. Σε καθεμία περίπτωση, η αναφορά γίνεται τότε που ένα scene graph object κατασκευάζεται, το οποίο είναι ο πρώτος υπολογισμός της **Behavior**.

Η μέθοδος *initialization* καλείται όταν το scene graph object που περιέχει την κλάση Behavior γίνεται ζωντανό.

Η μέθοδος *initialization* είναι αρμόδια για τον καθορισμό του αρχικού γεγονότος ώθησης για τη συμπεριφορά και τον καθορισμό του αρχικού όρου των state μεταβλητών για τη συμπεριφορά. Η ώθηση προσδιορίζεται ως WakeupCondition αντικείμενο, ή ένας συνδυασμός WakeupCondition.

Η μέθοδος *processStimulus* καλείται όταν εμφανίζεται το γεγονός ώθησης που προσδιορίζεται για τη συμπεριφορά (behavior). Η μέθοδος *processStimulus* είναι αρμόδια για την απάντηση στο γεγονός (event). Δεδομένου ότι πολλά γεγονότα (events) μπορούν να κωδικοποιηθούν σε ένα ενιαίο αντικείμενο WakeupCondition (π.χ., ποικίλες ενέργειες πληκτρολογίων μπορούν να κωδικοποιηθούν στο WakeupOnAWTEvent), αυτό περιλαμβάνει την αποκωδικοποίηση του γεγονότος .



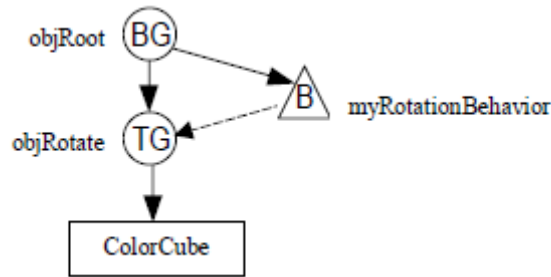
Η κλάση SimpleBehavior

Για να δημιουργήσουμε μια τέτοια κλάση **Behavior**, όλα αυτά που απαιτούνται είναι μια αναφορά στην TransformGroup (το object of change για αυτήν την κλάση), και μια μεταβλητή γωνίας. Η μεταβλητή γωνίας αλλάζει και η TransformGroup τίθεται στη γωνία. Δεδομένου ότι η συμπεριφορά(behavior) θα ενεργήσει στο αντικείμενο TransformGroup, το τι περιστρέφεται δεν είναι σημαντικό.

Για να δημιουργηθεί αυτή η κλάση δεν χρειάζεται τίποτα περισσότερο από τα τρία ουσιαστικά συστατικά προγραμματισμού : ένας constructor, η μέθοδος initialization, και η μέθοδος processStimulus. Ο constructor θα καταχωρήσει μια αναφορά στο TransformGroup object of change. Η μέθοδος initialization θέτει την αρχική ώθηση σε WakeOnAWTEvent, και θέτει τη γωνία περιστροφής σε μηδέν. Όπως αναφέρεται παραπάνω, το ερέθισμα σε μια συμπεριφορά προσδιορίζεται ως αντικείμενο WakeupCondition.

Η μέθοδος processStimulus δεν αποκωδικοποιεί το wakeup condition.

Είναι δυνατό να αποκωδικοποιηθεί το key press event για να καθορίσει ποιο πλήκτρο, ή ο συνδυασμός πλήκτρων, πιέστηκε.



Στα τρία βήματα της χρήσης μιας κλάσης Behavior, τα δύο πλέον πιθανά λάθη προγραμματισμού είναι:

- μην προσδιορίζοντας τα όρια ενός σχεδιασμού (σωστά), και
- μην προσθέτοντας μια συμπεριφορά στο σκηνογράφημα

KeyNavigatorBehavior και KeyNavigator κλάσεις

Το keyboard navigation utility εφαρμόζεται ως δύο κλάσεις. Στο χρόνο εκτέλεσης υπάρχουν δύο αντικείμενα. Το πρώτο αντικείμενο είναι το αντικείμενο KeyNavigatorBehavior, το δεύτερο είναι ένα αντικείμενο KeyNavigator. Η δεύτερη κλάση δεν είναι τεκμηριωμένη εδώ δεδομένου ότι ούτε ο προγραμματιστής ούτε ο χρήστης δεν πρέπει να ξέρει ότι η δεύτερη κλάση ή το αντικείμενο υπάρχει. Το αντικείμενο KeyNavigatorBehavior εκτελεί όλες τις χαρακτηριστικές λειτουργίες μιας κλάσης Behavior, εκτός από το ότι καλεί στο αντικείμενο KeyNavigator για να εκτελέσει τη λειτουργία processStimulus. Η κλάση KeyNavigator παίρνει το AWTEvent και το επεξεργάζεται κάτω στο key stroke level..Το KeyNavigator εφαρμόζει την κίνηση με την ταχύτητα.

Table 4-3 KeyNavigatorBehavior Movements

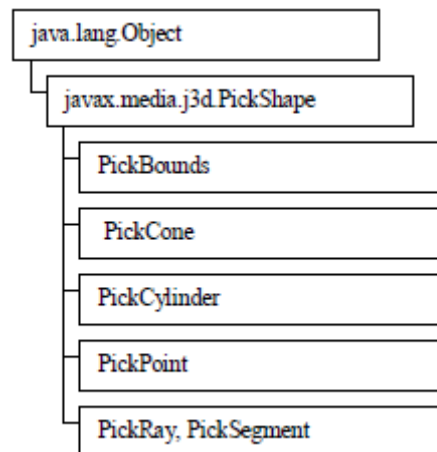
Key	MOVEMENT	Alt-key movement
←	rotate left	lateral translate left
→	rotate right	lateral translate right
↑	move forward	
↓	move backward	
PgUp	rotate up	translation up
PgDn	rotate down	translation down
+	restore back clip distance (and return to the origin)	
-	reduce back clip distance	
=	return to center of universe	

Utility κλάσεις για Mouse Interaction

MouseBehavior class	Action in Response to Mouse Action	Mouse Action
MouseRotate	rotate visual object in place	left-button held with mouse movement
MouseTranslate	translate the visual object in a plane parallel to the image plate	right-button held with mouse movement
MouseZoom	translate the visual object in a plane orthogonal to the image plate	middle-button held with mouse movement

PickShape κλάσεις

Αυτή η abstract κλάση δεν παρέχει ούτε constructors ούτε μεθόδους. Παρέχει την αφαίρεση (abstraction) για πέντε υποκατηγορίες: PickBounds, PickRay, PickSegment, PickPoint, και PickCone .



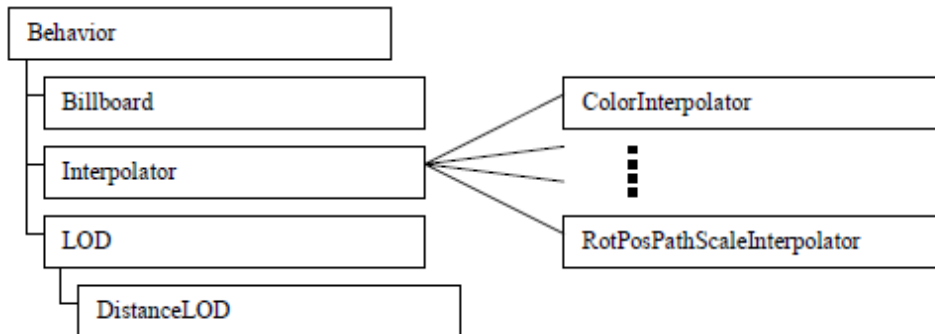
application	geometry		
	polygons appear large	polygons appear small	points and/or lines
general	PickRay	PickRay, PickBounds, PickCone, PickCylinder	PickBounds, PickCone, PickCylinder
accuracy	PickRay, PickSegment, PickPoint	PickRay, PickSegment, PickPoint,	PickRay, PickSegment, PickPoint
speed	PickRay	PickRay	PickRay

1.6.4 Animations

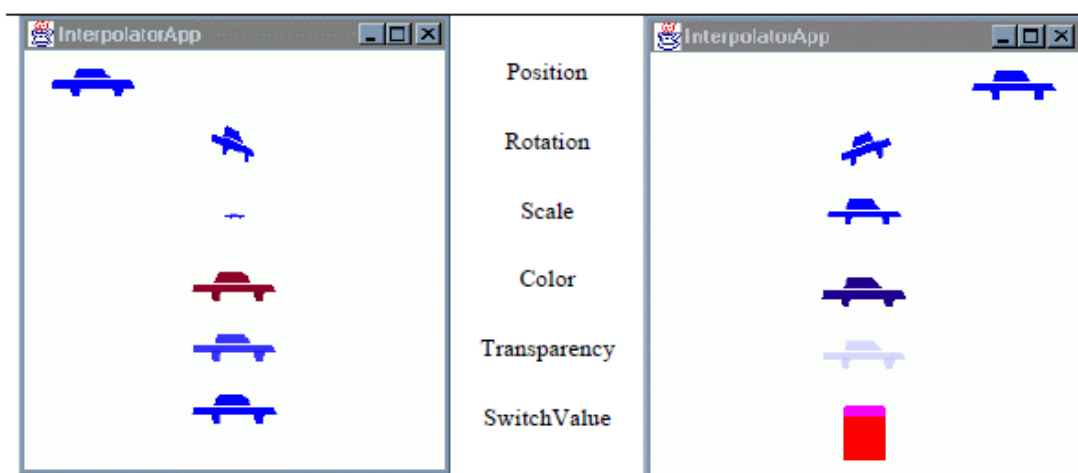
Όπως με την αλληλεπίδραση (interaction), τα animations στην Java3D εφαρμόζονται χρησιμοποιώντας το Behavior object . Οποιοδήποτε animation μπορεί να δημιουργηθεί χρησιμοποιώντας τα αντικείμενα Behavior. Εντούτοις, η Java 3D API παρέχει διάφορες κλάσεις χρήσιμες στη δημιουργία των animations χωρίς να πρέπει να δημιουργηθεί μια νέα κλάση. Δεν πρέπει να προκαλέσει καμιά έκπληξη ότι αυτές οι κλάσεις είναι βασισμένες στην κλάση Behavior.

Ένα σύνολο animations κλάσεων είναι γνωστό ως interpolators. Ένα Interpolator αντικείμενο, μαζί με ένα Alpha αντικείμενο, χειρίζεται κάποια παράμετρο ενός scene graph object για να δημιουργήσει ένα χρόνο-βασισμένο animation. Το Alpha αντικείμενο παρέχει το συγχρονισμό.

Ένα σύνολο animations κλάσεων ζωντανεύει τα οπτικά αντικείμενα σε απάντηση στις αλλαγές όψης. Αυτό το σύνολο κλάσεων περιλαμβάνει την κλάση OrientedShape3D, το Billboard και το Level of Detail (LOD) που οδηγούνται όχι από το πέρασμα του χρόνου, αλλά στη θέση ή τον προσανατολισμό της όψης.



Ένα Alpha αντικείμενο παράγει μια τιμή μεταξύ μηδέν και ένα, συμπεριλαμβάνοντας, ανάλογα με το χρόνο και παραμέτρους του Alpha αντικειμένου. Τα Interpolators είναι προσαρμοσμένα αντικείμενα συμπεριφοράς που χρησιμοποιούν ένα Alpha αντικείμενο για να παρέχουν τα animations των οπτικών (visual) αντικειμένων. Οι ενέργειες των Interpolator περιλαμβάνουν την αλλαγή της θέσης, του προσανατολισμού, του μεγέθους, του χρώματος, ή της διαφάνειας ενός οπτικού αντικειμένου. Όλες οι interpolator συμπεριφορές θα μπορούσαν να εφαρμοστούν με τη δημιουργία μιας κλάσης συμπεριφοράς (behavior) ωστόσο, η χρησιμοποίηση interpolator κάνει τη δημιουργία αυτών των animations πολύ πιο εύκολη. Οι Interpolator κλάσεις υπάρχουν και για άλλες ενέργειες, συμπεριλαμβανομένων μερικών συνδυασμών αυτών των ενεργειών.



1.7 Απόδοση Υφής (Textures)

Η εμφάνιση πολλών αντικειμένων του πραγματικού κόσμου εξαρτάται από τη σύστασή της (**texture**). Η σύσταση ενός αντικειμένου είναι πραγματικά η σχετικά λεπτομερής γεωμετρία στην επιφάνεια ενός αντικειμένου. Για να εκτιμήσουμε το παιχνίδι σύστασης (**texture**) της επιφάνειας στην εμφάνιση των αντικειμένων του πραγματικού κόσμου, θεωρούμε τον τάπητα. Ακόμα και όταν όλες οι ίνες ενός τάπητα έχουν το ίδιο χρώμα, ο τάπητας δεν εμφανίζεται ως σταθερό χρώμα λόγω της αλληλεπίδρασης του φωτός με τη γεωμετρία των ινών.

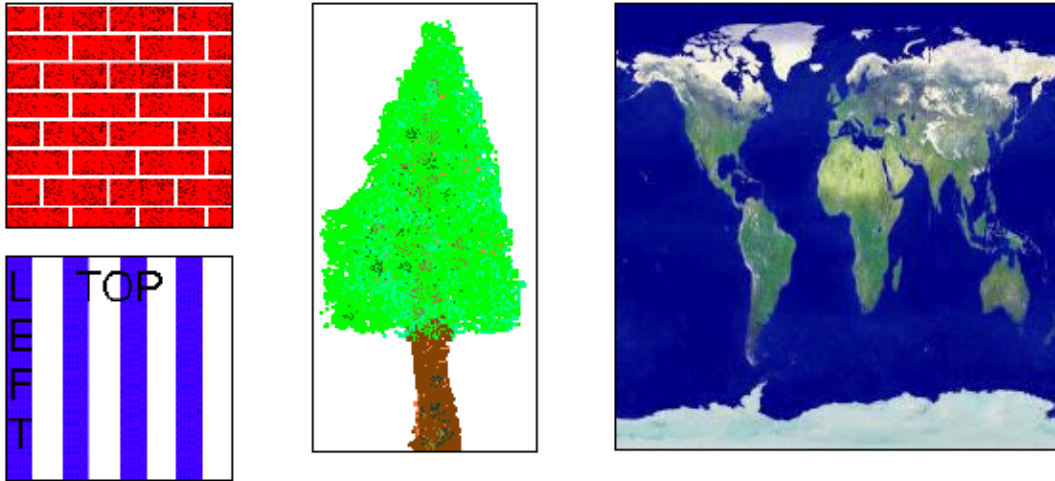
Ακόμα κι αν η Java3D είναι σε θέση να μοντελοποιήσει τη γεωμετρία των μεμονωμένων νημάτων του τάπητα, οι απαιτήσεις μνήμης και η απόδοση της παράστασης για ένα μεγέθους δωματίου κομμάτι του τάπητα που διαμορφώνεται με τέτοια λεπτομέρεια, θα καθιστούσαν ένα τέτοιο μοντέλο άχρηστο. Αφ' ετέρου, η κατοχή ενός επίπεδου πολυγώνου ενιαίου χρώματος δεν κάνει μια πειστική αντικατάσταση για τον τάπητα στη σκηνή.

1.7.1 Τι είναι το Texturing

Ο τάπητας μπορεί να είναι το ακραίο παράδειγμα από την άποψη της πολυπλοκότητας και της πυκνότητας της γεωμετρικής επιφάνειας, αλλά είναι μακριά από το μοναδικό αντικείμενο για το οποίο αντιλαμβανόμαστε τη σύσταση. Τα τούβλα, το σκυρόδεμα, το ξύλο, οι χορτοτάπητες, οι τοίχοι, και το χαρτί είναι ακριβώς μερικά από τα αντικείμενα για τα οποία η επίπεδη γεωμετρία (π.χ., πολύγωνα) μπορεί να χρησιμοποιηθεί για να αντιπροσωπεύσει τη γενική μορφή, αλλά όχι την ακριβή λεπτομέρεια. Ακριβώς όπως με τον τάπητα, το κόστος της σύστασης της επιφάνειας στα γεωμετρικά primitives, για αυτά τα αντικείμενα είναι αρκετά υψηλό.

Μια πιθανή εναλλακτική λύση της διαμόρφωσης του νήματος του τάπητα είναι να διαμορφωθεί ο τάπητας ως επίπεδο πολύγωνο με πολλές κορυφές, δίνοντας χρώματα στις κορυφές για να δώσουμε τις ποικιλία στο χρώμα. Εάν οι κορυφές είναι αρκετά στενές, τότε η εικόνα του τάπητα μπορεί να παραχθεί. Αυτό απαιτεί τη σημαντικά λιγότερη μνήμη από το μοντέλο που περιλαμβάνει τα νήματα του τάπητα, ωστόσο, το μοντέλο θα απαιτούσε ακόμα περισσότερη μνήμη για ένα δωμάτιο λογικού μεγέθους. Αυτή η ιδέα, της αντιπροσώπευσης της εικόνας του αντικειμένου σε μια επίπεδη επιφάνεια, είναι η βασική ιδέα του **texturing**. Εντούτοις, με το **texturing**, η γεωμετρία μπορεί να είναι πολύ απλή.

Το **texture**, αποκαλείται επίσης **texturing mapping**, είναι ένας τρόπος να προστεθεί η οπτική αφθονία (visual richness) μιας επιφάνειας χωρίς προσθήκη των γεωμετρικών λεπτομερειών. Η οπτική αφθονία παρέχεται από μια εικόνα, αποκαλούμενη επίσης **texture**, το οποίο δίνει την εμφάνιση της λεπτομερής επιφάνειας για το οπτικό αντικείμενο (visual object). Η εικόνα χαρτογραφείται πάνω στη γεωμετρία του οπτικού αντικειμένου. Κατά συνέπεια αυτός είναι ο όρος **texturing mapping**.



1.7.2 Απλή συνταγή δημιουργίας του texturing

- Βήμα 1 : Ετοιμάζουμε το texture image

Αυτή η συνταγή αρχίζει με ένα non-programming βήμα: «προετοιμάστε τα texture images». Η δημιουργία και η σύνταξη των texture images είναι κάτι που γίνεται κανονικά στα τρισδιάστατα προγράμματα της Java. Στην πραγματικότητα, τα περισσότερα texture images προετοιμάζονται προτού να αρχίσει το πρόγραμμα. Υπάρχουν δύο ουσιαστικές στοιχειώδεις εργασίες κατά την προετοιμασία των texture images:

- 1) που εξασφαλίζει αν οι εικόνες είναι αποδεκτών διαστάσεων, και
- 2) που εξασφαλίζουν αν οι εικόνες σώζονται σε μια μορφή αρχείου που μπορεί να διαβαστεί. Φυσικά η εικόνα θα μπορούσε να επιμεληθεί για να επιτύχει το επιθυμητό χρώμα, τη διαφάνεια και το χτίσιμο

Για την απόδοση της δραστηριότητας, η Java3D απαιτεί το μέγεθος του texture image να είναι μια μαθηματική δύναμη του δύο (1, 2, 4, 8, 16,...) σε κάθε διάσταση. Η αποτυχία να γίνει αυτός ο περιορισμός θα έχει σαν αποτέλεσμα σε ένα runtime error exception.

Εάν μια εικόνα δεν είναι αποδεκτών διαστάσεων, πρέπει να τροποποιηθεί (scaled ή cropped) για να καλύψει τις απαιτήσεις διάστασης προτού χρησιμοποιηθεί. Το image editing μπορεί να γίνει σε μια ευρεία ποικιλία των προγραμμάτων συμπεριλαμβανομένης της Java Advanced Imaging API.

Όσο οι μορφές αρχείου (file formats) είναι εμπλεκόμενες, ένα texture (image) μπορεί να καταχωρηθεί με οποιαδήποτε μορφή αρχείου εικόνας με τον όρο ότι υπάρχει μια μέθοδος για να την φορτώσει. Τα προγράμματα αυτού του κεφαλαίου φορτώνουν τα textures χρησιμοποιώντας την κλάση TextureLoader. Ένα αντικείμενο TextureLoader φορτώνει JPEG, GIF, και άλλες μορφές αρχείου.

- Βήμα 2^α : Φορτώνουμε το texture

Το επόμενο βήμα είναι να αποκτηθεί η έτοιμη εικόνα σε ένα image object. Αυτό είναι γνωστό σαν loading the texture(φορτώνοντας την σύσταση-δομή).

Το επόμενο βήμα είναι να αποκτηθεί η έτοιμη εικόνα σε ένα αντικείμενο εικόνας. Αυτό είναι γνωστό όπως φορτώνοντας τη σύσταση.

Οι συστάσεις μπορούν να φορτωθούν από τα αρχεία ή τα URLs χρησιμοποιώντας την ίδια βασική διαδικασία. Η φόρτωση μιας σύστασης μπορεί να ολοκληρωθεί με πολλές γραμμές κώδικα, ή με δύο γραμμές κώδικα που χρησιμοποιούν ένα αντικείμενο TextureLoader.

Σε κάθε τρόπο, το αποτέλεσμα είναι να αποκτηθεί η εικόνα σε ένα αντικείμενο ImageComponent2D. Το αποτέλεσμα αυτών των δύο γραμμών είναι να φορτωθεί η εικόνα από το αρχείο stripe.gif σε ένα αντικείμενο Image2DComponent που μπορεί να χρησιμοποιηθεί για να δημιουργήσει την απαραίτητη δέσμη εμφάνισης (appearance bundle) του βήματος 3.

1. TextureLoader loader = new TextureLoader(“stripe.gif”, this);
2. ImageComponent2D image = loader.getImage();

Πριν πάμε προς το βήμα 3 της συνταγής, ρίχνουμε μια προσεκτικότερη ματιά στη χρήση του αντικειμένου TextureLoader. Το δεύτερο όρισμα του κατασκευαστή προσδιορίζει ένα αντικείμενο που εξυπηρετεί τον *image observer*(παρατηρητής εικόνας). Η κλάση TextureLoader χρησιμοποιεί το πακέτο java.awt.image για τη φόρτωση των εικόνων. Αυτή το πακέτο φορτώνει τις εικόνες ασύγχρονα, το οποίο είναι ιδιαίτερα χρήσιμο όταν φορτώνεται μια εικόνα από ένα URL. Για την διευκόλυνση της φόρτωσης των ασύγχρονων εικόνων, τα συστατικά AWT είναι ικανοί image observers, που πρόκειται να παρατηρήσουν τη διαδικασία φόρτωσης της εικόνας. Ένας image observer μπορεί να ρωτηθεί για τις λεπτομέρειες της φόρτωσης της εικόνας.

- Βήμα 2β : Δημιουργούμε το appearance bundle

Για να χρησιμοποιηθεί ως texture για ένα visual object(οπτικό αντικείμενο), το texture image που φορτώνεται στο βήμα 2α πρέπει να τεθεί ως texture σε ένα Texture αντικείμενο, το οποίο χρησιμοποιείται έπειτα σε μια appearance bundle που δηλώνεται από το visual object.

Συγκεκριμένα, ένα αντικείμενο Texture2D κρατά το texture image. Η εικόνα ImageComponent2D που φορτώνεται στο βήμα 2^α είναι κεντρική στη δημιουργία των appearance bundles(δεσμών εμφάνισης) του βήματος 2β.

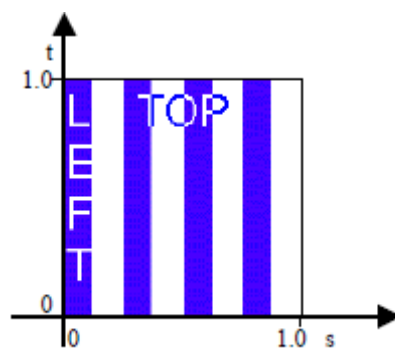
Στις παρακάτω γραμμές έχοντας φορτώσει το texture (γραμμή 1 και 2) η εικόνα είναι τοποθετημένη στο αντικείμενο Texture2D(γραμμή 4). Το αντικείμενο Texture2D προστίθεται στο αντικείμενο Appearance (γραμμή 6).

1. `TextureLoader loader = new TextureLoader("stripe.jpg", this);`
2. `ImageComponent2D image = loader.getImage();`
3. `Texture2D texture = new Texture2D();`
4. `texture.setImage(0, image);`
5. `Appearance appear = new Appearance();`
6. `appear.setTexture(texture);`

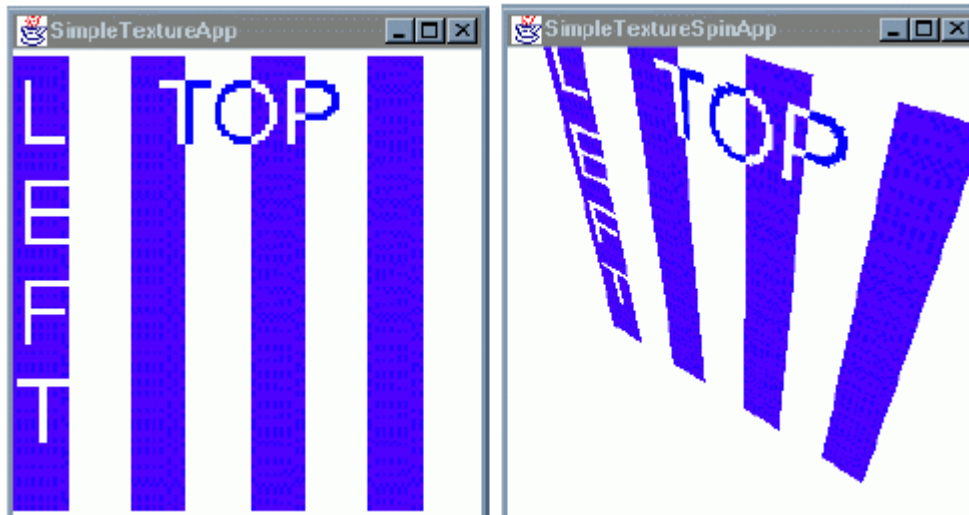
Το `appearance bundle` που δημιουργήθηκε παραπάνω στον κώδικα, θα μπορούσε να έχει άλλα τμήματα κόμβων (node components), ειδικότερα είναι το τμήμα κόμβων `TextureAttributes`.

- Βήμα 3 : Καθορισμός των `TextureCoordinates`

Στην περίπτωση που φορτώσουμε το **texture** σε ένα **appearance bundle**, ο προγραμματιστής προσδιορίζει επίσης, την τοποθέτηση του texture στη γεωμετρία μέσω της λεπτομερής προδιαγραφής των **texture coordinates**. Οι λεπτομερείς προδιαγραφές των *texture coordinates* γίνονται ανά γεωμετρική κορυφή. Κάθε *texture coordinate* προσδιορίζει ένα σημείο από το texture που εφαρμόζεται στην κορυφή. Με την λεπτομερή προδιαγραφή μερικών σημείων της εικόνας που εφαρμόζεται στις κορυφές της γεωμετρίας, η εικόνα θα περιστραφεί, έπειτα θα τεντωθεί, θα συμπιεστεί και θα αναπαραχθεί για να εφαρμόσει.



Τα `Texture Coordinates` είναι ορισμένα στις **s (οριζόντιες)** και **t (κάθετες)** διαστάσεις του texture image, στο διάστημα μεταξύ του 0,0 και 1,0 .



1.7.3 Επισκόπηση χρήσης της τεχνολογίας

Σύμφωνα με την παραπάνω περιγραφή είναι φανερό ότι σκοπός της τεχνολογίας Java 3D είναι να παρέχει ένα σύνολο βιβλιοθηκών υψηλού επιπέδου για τρισδιάστατα γραφικά. Η λογική ανάπτυξης του Java 3D είναι αρκετά ανοιχτή επιτρέποντας μεγάλο εύρος εφαρμογών που να αξιοποιούν τόσο της Java 3D όσο και τις γενικότερες Java βιβλιοθήκες. Επίσης η δυνατότητα για συνεργασία με άλλες τεχνολογίες και πρότυπα καθιστά ακόμα πιο εύκολη την ανάπτυξη εφαρμογών για τους προγραμματιστές, στην περίπτωση της υποστηριζόμενης τεχνολογίας ανήκει το πρότυπο VRML97.

Όπως υποστηρίζεται στον ιστότοπο της Java το API γραφικών μπορεί να χρησιμοποιηθεί σε εφαρμογές CAD, για δικτυακές εφαρμογές, για παιχνίδια, για Data Visualizations κ.α. Οι υλοποιήσεις που έχουν πραγματοποιηθεί με την τεχνολογία εκτείνονται περισσότερο προς εφαρμογές CAD, Data Visualizations και δικτυακές εφαρμογές. Η Sun Microsystems, η οποία βρίσκεται πίσω από την ανάπτυξη της Java, ήταν από τις πρώτες εταιρείες που υλοποίησαν 3D Desktop χρησιμοποιώντας Java και Java 3D για το Visualization.



Το σύστημα Glass της Sun Microsystems, αποτελεί ένα από τα πρώτα 3D Desktop OS, έχει υλοποιηθεί με Java 3D.

The World Leader in Advanced Visualization

Demonstration Locations

Please contact FakeSpace to arrange a visit to any of our visualization demonstration facilities:

- FakeSpace Systems Technology Center**
Corporate Headquarters
Marshalltown, Iowa, USA
We would be happy to provide you with a tour and demonstration and discuss how your company can benefit from using our visualization technology.
- FakeSpace Houston Office**
Houston, Texas, USA
Visit our Houston Office to view a demonstration of the FakeSpace Interaction Engine (FIE), a universal interaction interface and integrated computer hardware and software solution that acts as a connectivity hub between visualization displays, interaction devices and computing platforms.
- Purdue University, Evision Center**
West Lafayette, Indiana, USA
Purdue's Evision Center incorporates a FLEX visualization theater as well as two displays. FakeSpace's software (CONVERT) capabilities also installed to strengthen Purdue's status as a Dassault System's PLM development center. Please contact FakeSpace to arrange a visit.
- Virtual Environments Technologies Center**
London, Ontario, Canada
FakeSpace Systems Inc. is a founding member of this Center, part of the 118001 Research Council of Canada (NRC). An immersive room, 6x24' PowerWall™ and two 6' x 6' PowerWall utilized by the NRC and by local design and manufacturing corporations.

Η εταιρεία FakeSpace που ασχολείται τόσο με την παροχή λογισμικού όσο και υλικού για Visualizations είναι μια από τις εταιρείες που χρησιμοποιεί Java 3D.

HSSPalign: P12931_1qcf-A | Sequence SRC_HUMAN Aligned onto 3D Structure 1QCF

Description of SRC_HUMAN Proto-oncogene tyrosine-protein kinase SRC (BC 2.7.1.112) (P60-SRC) (C-SRC).

Features of SRC_HUMAN Extracted from SwissProt/SPTREMBL.

Description of 1QCF crystal structure of hck in complex with a src family- selective tyrosine kinase inhibitor

Sequence Similarity: High 62% identical residues. SRC_HUMAN is very likely to adopt a structure similar to 1QCF.

Εικόνα από το πρόγραμμα SRS 3D βιοχημείας της εταιρείας Lion AG για την οπτικοποίηση μοριακών αλυσίδων που χρησιμοποιεί Java 3D.

Παρά το γεγονός ότι υπάρχουν αρκετές χρήσεις Java 3D σε εφαρμογές, κυρίως visualizations επιστημονικών δεδομένων, στις περισσότερες περιπτώσεις τα προγράμματα αυτά είναι αρκετά παλιά.

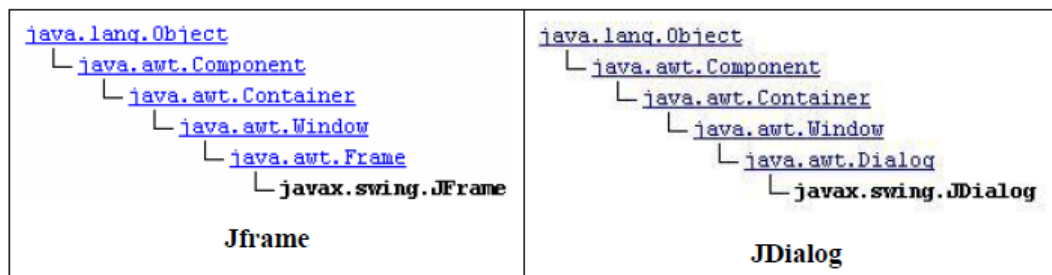
Κεφάλαιο 2^ο

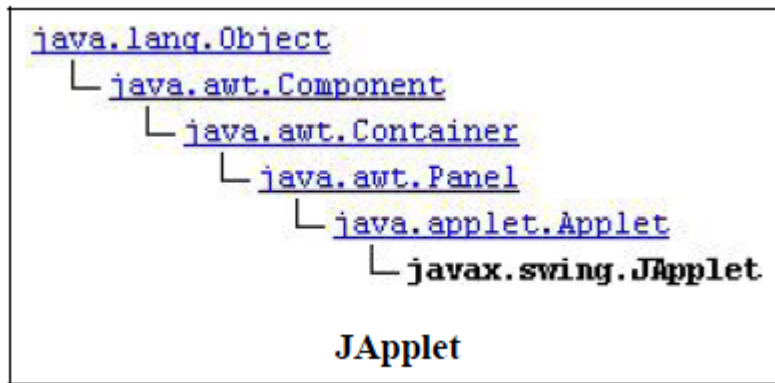
Java Swing

Τα πακέτα που αρχίζουν με το πρόθεμα javax.swing παρέχουν ευέλικτα και ισχυρά εργαλεία ανάπτυξης GUI. Ιδιαίτερα το πακέτο javax.swing αναπτύχθηκε κυρίως λόγω των ανεπαρκειών του Abstract Windows Toolkit (AWT). Παραδείγματος χάριν, η κλάση JButton που ορίζεται σε αυτό, υπερτερεί της AWT Button κλάσης επιτρέποντας την ύπαρξη όχι μόνο απλού κείμενου, αλλά και εικόνων στα κουμπιά.

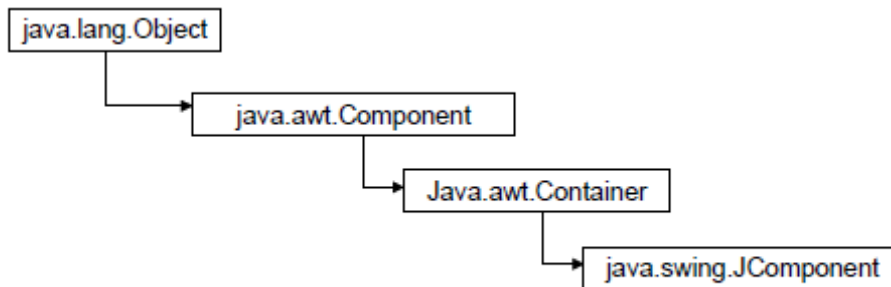
2.1 Top-Level Swing Containers και Swing Components

Κάθε γραφική διεπαφή τύπου Swing πρέπει να έχει τουλάχιστον ένα top-level Swing container. Ένα top-level Swing container παρέχει την απαραίτητη υποστήριξη που χρειάζονται τα Swing components για την εμφάνισή τους και την διαχείριση των γεγονότων που αυτά παράγουν. Υπάρχουν τρία top-level Swing containers: το JFrame, το JDialog, και (για applets) το JApplet. Κάθε JFrame αντικείμενο δημιουργεί ένα κύριο γραφικό παράθυρο, κάθε JDialog αντικείμενο δημιουργεί ένα δευτερεύον παράθυρο (δηλ παράθυρο που εξαρτάται από κάποιο άλλο παράθυρο). Κάθε JApplet αντικείμενο δημιουργεί την περιοχή εμφάνισης ενός applet στο παράθυρο του Web Browser. Η ιεραρχία των top-level Swing containers δίνεται παρακάτω:





Με εξαίρεση τα top-level containers, πχ το JFrame, όλα τα Swing components είναι υποκλάσεις της κλάσης JComponent. Η ιεραρχία της JComponent φαίνεται στο παρακάτω σχήμα:



Συνήθως, όταν υλοποιείται μια GUI εφαρμογή σε Swing, δημιουργούμε ένα JFrame αντικείμενο και επιλέγουμε συγκεκριμένο πλάνο ή σχέδιο (layout) για αυτό. Κατόπιν τοποθετούμε ένα ή περισσότερα JPanels στο JFrame. Τα JPanels διαθέτουν επίσης επιλογές layout όπως και τα JFrames. Στη συνέχεια, μπορούμε να προσθέτουμε και άλλα αντικείμενα τύπου Component.

2.2 Διαχειριστές Γεγονότων - Event Handlers

Για να υπάρξει κάποιο αποτέλεσμα κάνοντας κλικ σε ένα κουμπί χρειάζεται να προσθέσουμε σε αυτό το component (JButton αντικείμενο) τον κατάλληλο ακροατή γεγονότων (ButtonListener). Έτσι όταν ο χρήστης πατήσει το κουμπί, ο ακροατής προκαλείται και καλούνται οι κατάλληλες μέθοδοί του (διαχειριστές γεγονότων). Η εργασία αυτή γίνεται με τον ίδιο τρόπο που ορίζονται οι ακροατές γεγονότων στα AWT Component.

Τα Swing components παράγουν διάφορους τύπους γεγονότων (events). Ο παρακάτω πίνακας περιέχει τα πιο συνηθισμένα από αυτά :

Some Events and Their Associated Event Listeners	
Act that Results in the Event	Listener Type
User clicks a button, presses Enter while typing in a text field, or chooses a menu item	ActionListener
User closes a frame (main window)	WindowListener
User presses a mouse button while the cursor is over a component	MouseListener
User moves the mouse over a component	MouseMotionListener
Component becomes visible	ComponentListener
Component gets the keyboard focus	FocusListener
Table or list selection changes	ListSelectionListener
Any property in a component changes such as the text on a label	PropertyChangeListener

2.3 Swing και Threads

Η μετάβαση από τα AWT στα Swing GUI παρουσιάζει κάποιο μικρό πρόβλημα σε σχέση με τα Threads. Όλες οι AWT κλάσεις είναι συμβατές με τα threads (thread safe) ενώ οι swing κλάσεις δεν είναι.

Η Java επιτυγχάνει τον συγχρονισμό των threads που έχουν πρόσβαση σε κάποιο κοινό αντικείμενο. Οι AWT μέθοδοι συγχρονίζονται την στιγμή της αρχικοποίησης του αντίστοιχου αντικειμένου τους στην οθόνη. Έτσι εμποδίζονται πιθανά αδιέξοδα (deadlocks) που επιφέρουν το «κρέμασμα» εφαρμογών την στιγμή της εμφάνισης γραφικών αντικείμενων (πχ. buttons, panels, κλπ) στην οθόνη.

Ωστόσο όπως αναφέρθηκε και προηγουμένως οι swing κλάσεις δεν είναι thread-safe.

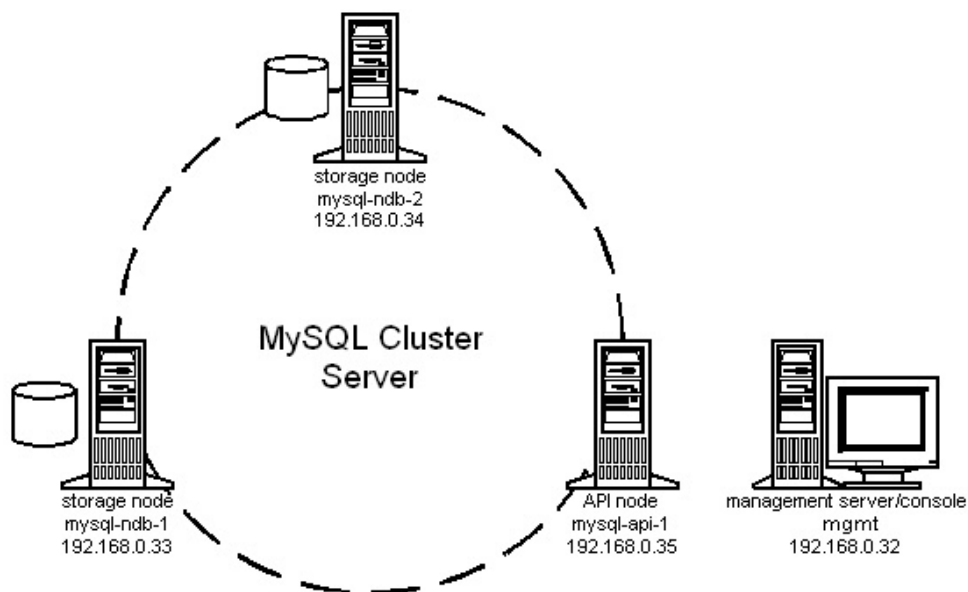
Ο κώδικας διαχείρισης γεγονότων στις Swing εφαρμογές εκτελείται μέσα από μια συγκεκριμένη thread η οποία ονομάζεται *event-dispatching thread*. Με αυτό τον τρόπο επιτυγχάνεται ο κώδικας ενός event handler να τελειώνει πριν την εκτέλεση του κώδικα ενός άλλου event handler. Για την αποφυγή πιθανού αδιεξόδου (deadlock), είναι απαραίτητο και τα Swing components να δημιουργούνται, τροποποιούνται και ερωτούνται για την κατάστασή τους μέσα από την *event-dispatching thread*.

Κεφάλαιο 3^ο

Βάση δεδομένων MySQL

3.1 Τι είναι η MySQL βάση δεδομένων

MySQL είναι ένα ανοικτό λογισμικό βάσεων δεδομένων πηγής. Είναι μέρος μιας μεγάλης και αυξανόμενης οικογένειας του λογισμικού ανοιχτού κώδικα συμπεριλαμβανομένου Linux, Apache, και των πέσος Φιλιππίνων γλωσσών προγραμματισμού και Perl. Εντούτοις, αντίθετα από πολλά ανοικτά προϊόντα πηγής όπου τα πνευματικά δικαιώματα είναι κύρια από τους μεμονωμένους συντάκτες ή τους διανομείς, η MySQL είναι κύριο και υποστηρίζεται από μια ενιαία εταιρία για-κέρδους, MySQL ab, το οποίο κρατά τα πνευματικά δικαιώματα στο μεγαλύτερο μέρος του codebase. Αναπτύσσουν και διατηρούν το σύστημα, πωλούν την υποστήριξη, τις συμβάσεις υπηρεσιών, και τα ιδιόκτητος-χορηγημένα άδεια αντίγραφα MySQL.



2/9/2005 - LOD Consulting
consult@lod.com

Το MySQL είναι μια βάση δεδομένων σχέσης, η οποία είναι μια συλλογή των αντικειμένων ή των πινάκων που οργανώνονται στις προκαθορισμένες κατηγορίες ή των περιγεγραμμένων πινάκων από τους οποίους τα στοιχεία μπορούν να προσεγγιστούν ή να συγκεντρωθούν εκ νέου σε πολλούς διαφορετικούς τρόπους

χωρίς να πρέπει να αναδιοργανωθούν οι πίνακες βάσεων δεδομένων. Αυτή η μορφή είναι πάρα πολύ δημοφιλής με περισσότερα από 100 εκατομμύρια αντίγραφα του λογισμικού που μεταφορτώνεται κατά τη διάρκεια ζωής της MySQL. Η δημοτικότητα προέρχεται από το γεγονός ότι οι βάσεις δεδομένων είναι εύκολο να δημιουργηθούν και να έχουν πρόσβαση. Επιπλέον, είναι εύκολο να επεκταθούν, μια νέα κατηγορία στοιχείων μπορεί να προστεθεί χωρίς απαίτηση ότι όλες οι υπάρχουσες εφαρμογές τροποποιούνται.

Κατά δημιουργία μιας σχεσιακής βάσης δεδομένων, την μπορείτε όχι μόνο να επεκτείνετε τα στοιχεία, μπορείτε να καθορίσετε τη δικτυακή γειτονιά των πιθανών τιμών σε μια στήλη στοιχείων και οποιουσδήποτε πρόσθετους περιορισμούς που θέλετε να εφαρμόσετε στην αξία των στοιχείων σας. Αυτό καθιστά τις πληροφορίες σας σημαντικά πολυτιμότερες και φιλικές προς το χρήστη.

3.2 Οφέλη MySQL

- Ευκολία - - χρήση. Μια βασική γνώση του SQL και χειρισμού βάσεων δεδομένων απαιτείται, εντούτοις η MySQL είναι ένα λογισμικό λογικό και εύχρηστο. Αυτό σημαίνει ότι δεν είναι απαραίτητο να ξοδέψουμε τόσο εκπαιδευτικό υπάρχον προσωπικό χρόνου και χρημάτων είτε τους υπεύθυνους για την ανάπτυξη μίσθωσης.
- Ασφάλεια. Δεν υπάρχουν πολλές βάσεις δεδομένων που μπορούν να συγκριθούν με την ασφάλεια που η MySQL προσφέρει. Περιλαμβάνει τα στερεά στρώματα ασφαλείας δεδομένων που προστατεύουν τη ευαίσθητη πληροφορία μας. Τα δικαιώματα μπορούν να τεθούν ως στόχος να επιτρέψουν μερικά ή όλα τα προνόμια είτε στα άτομα είτε στις ομάδες. Οι κωδικοί πρόσβασης κρυπτογραφούνται. Η πρόσβαση στη βάση δεδομένων μας μπορεί να ελεγχθεί από μια απομακρυσμένη μηχανή που μπορεί εντελώς να ελέγξει τι ένας χρήστης μπορεί να εμφανίσει. Το σύστημα μπορεί να κλειδωθεί από ακόμα περισσότερους μέσω του λειτουργικού συστήματος.
- Ικανότητα και μνήμη. Το MySQL μπορεί να χειριστεί σχεδόν οποιοδήποτε ποσό στοιχείων από μόνο μερικές σειρές σε τουλάχιστον 50 εκατομμύρια σειρές ή περισσότερες. Το όριο μεγέθους προκαθορισμένων αρχείων είναι περίπου 4 MB. Διαχειρίζεται τη μνήμη πολύ καλά, το οποίο σημαίνει ότι λειτουργεί αποτελεσματικά και γρήγορα χωρίς τη συντριβή ή καταγραφή των πληροφοριών μας. Επιπλέον, ο κεντρικός υπολογιστής MySQL έχει εξεταστεί λεπτομερώς για να αποτρέψει τις διαρροές μνήμης.
- Ανίκητη ευελιξία. Τρεξίματα MySQL σε πολλά λειτουργικά συστήματα, συμπεριλαμβανομένου Novell Netware, Windows, Linux και Unix.
- Υποστηρίζει διάφορες διαπροσωπείες ανάπτυξης συμπεριλαμβανομένου [JDBC](#), ODBC, και των γλωσσών σεναριογραφιών συμπεριλαμβανομένων των πέσος Φιλιπίνων και Perl. Αυτό επιτρέπει σε μας να δημιουργήσουμε τις λύσεις βάσεων δεδομένων που τρέχουν σε όλα τα σημαντικά λειτουργικά συστήματα, συμπεριλαμβανομένου Linux, UNIX, και των Windows.

- Περισσότερα από ένα πρόσωπα μπορούν να έχουν πρόσβαση στις επιχειρησιακές πληροφορίες σε έναν χρόνο. Αυτό σημαίνει ότι μια μεγάλη ομάδα ανθρώπων μπορεί να έχει πρόσβαση στις πληροφορίες αμέσως αλλά μπορούν επίσης να υπάρξουν διάφορες συνδέσεις συγχρόνως.

3.3 Μειονεκτήματα MySQL

Οι πρώρες εκδόσεις MySQL στερήθηκαν πολλά πρότυπα χαρακτηριστικά γνωρίσματα συστημάτων διαχείρισης σχεσιακής βάσης δεδομένων (RDBMS), συνήθως με το στόχο τις εμπορικές συναλλαγές εγγυημένης ακρίβειας για την ταχύτητα. Πολλές (αλλά όχι ακόμα όλοι) τέτοιες κριτικές έχουν θεραπευθεί στις πιο πρόσφατες εκδόσεις του λογισμικού, συμπεριλαμβανομένων των συναλλαγών και των συγγενικών περιορισμών ακεραιότητας.

Μερικοί λένε ότι ο κατάλογος χαρακτηριστικών γνωρισμάτων MySQL αφήνει κάτι για να επιδιωχτεί, αλλά οι θαυμαστές του ορκίζονται ότι μειωμένα καθορισμένα μέσα χαρακτηριστικών γνωρισμάτων που πρέπει μόνο να επεκτείνουμε, να διαμορφώσουμε και να διατηρήσουμε ότι χρειαζόμαστε .

Υπάρχουν επίσης μερικά κοινά σφάλματα όπως από τη μνήμη ή ένα πακέτο πάρα πολύ μεγάλο που αφορούν γενικά τη διαμόρφωση του κεντρικού υπολογιστή τα οποία είναι ασυνήθιστα και είναι αρκετά εύκολο να θεραπευθούν.

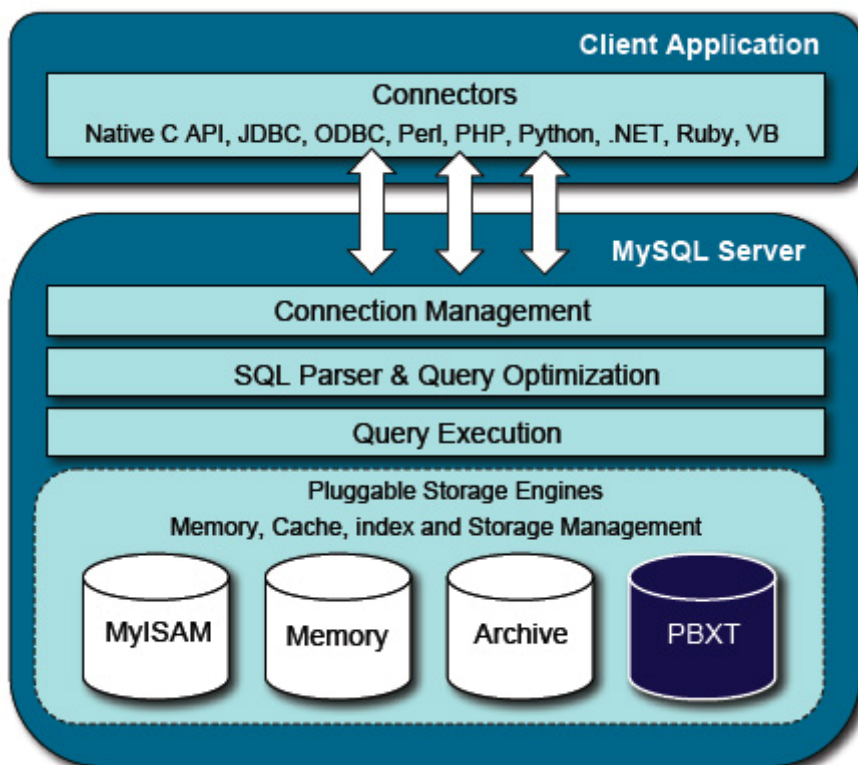
3.4 Χρήση της MySQL

Μια **βάση δεδομένων** (database) αποτελείται από έναν ή περισσότερους **πίνακες** (tables), ο καθένας από τους οποίους περιέχει μια λίστα από κάποια πράγματα .Για μια βάση δεδομένων πελατών (clients), είναι φυσικό να ξεκινήσουμε μ' έναν πίνακα με όνομα clients που θα περιέχει μια λίστα από στοιχεία πελατών. Ο κάθε πίνακας σε μια βάση δεδομένων περιέχει μια ή περισσότερες **στήλες** (columns) ή **πεδία** (fields), όπου η κάθε στήλη περιέχει μια συγκεκριμένη πληροφορία για τον κάθε πελάτη που υπάρχει στην βάση δεδομένων (database). Ο πίνακας clients μπορεί να περιέχει στήλες για τον κωδικό ενός πελάτη (ID), για το όνομά του (Name) καθώς και για την ημερομηνία γέννησής του (Date). Το κάθε τι που αποθηκεύουμε σ' αυτόν τον πίνακα λέμε ότι αποτελεί μια **γραμμή** (row) ή μια **εγγραφή** (record) του πίνακα. Για παράδειγμα, ας δούμε τον παρακάτω πίνακα :

ID	Name	Date
1	Κώστας	15/09/1986
2	Μαρία	23/10/1987

Εκτός από τις στήλες για το όνομα του πελάτη (Name) και την ημερομηνία γέννησής του (Date), υπάρχει και μια στήλη με όνομα ID, ο σκοπός της οποίας είναι να εκχωρήσει έναν μοναδικό αριθμό στον κάθε πελάτη έτσι ώστε να έχουμε έναν εύκολο τρόπο αναφοράς σ' αυτόν και να μπορούμε να τον ξεχωρίσουμε από τους άλλους

πελάτες. Σαν επισκόπηση, το παραπάνω είναι ένας πίνακας τριών στηλών που περιέχει δύο γραμμές ή καταχωρήσεις. Η κάθε γραμμή του πίνακα περιέχει έναν κωδικό (ID) αναγνώρισης του πελάτη, το όνομά του (text) καθώς και την ημερομηνία γέννησής του (date). Με βάση αυτήν την βασική ορολογία, είμαστε έτοιμοι να αρχίσουμε να χρησιμοποιούμε την MySQL.



Το standard interface για να δουλέψουμε με τις βάσεις δεδομένων της MySQL είναι να συνδεθούμε με το λογισμικό του MySQL server και να δίνουμε μία εντολή την φορά. Για να κάνουμε αυτήν την σύνδεση με τον server, θα χρειασθούμε το *πρόγραμμα πελάτη* (client program) της MySQL. Στο Linux, το πρόγραμμα αποκαλείται **mysql** και βρίσκεται εξ ορισμού στον κατάλογο /usr/local/mysql/bin, ενώ στα Windows, το πρόγραμμα αποκαλείται **mysql.exe** και βρίσκεται εξ ορισμού στον

κατάλογο C:\mysql\bin. Υπάρχουν δύο τρόποι για να μπορέσουμε να συνδεθούμε με τον MySQL server. Ο πρώτος είναι να χρησιμοποιήσουμε το telnet για να συνδεθούμε (log into) στον server του Web host που μας φιλοξενεί και να δώσουμε την εντολή mysql από εκεί. Ο δεύτερος είναι να φορτώσουμε (download) και να εγκαταστήσουμε το λογισμικό πελάτη (client software) της MySQL από το site <http://www.mysql.com/> στον δικό μας υπολογιστή και να το χρησιμοποιήσουμε για να συνδεθούμε με τον MySQL server. Όποια μέθοδο κι αν επιλέξουμε και όποιο λειτουργικό σύστημα χρησιμοποιούμε, θα καταλήξουμε σε μια γραμμή εντολών (command line), έτοιμοι να εκτελέσουμε το πρόγραμμα πελάτη της MySQL για να συνδεθούμε στον MySQL server.

Πρέπει να γράψουμε τα εξής `mysql -h <hostname> -u <username> -p`

Θα πρέπει να αντικαταστήσουμε το *<hostname>* με το όνομα του host ή την IP διεύθυνση του υπολογιστή στον οποίο εκτελείται ο MySQL server. Αν εκτελούμε το πρόγραμμα πελάτη στον ίδιο υπολογιστή με τον server, μπορούμε να παραλείψουμε το τμήμα *-h <hostname>* της εντολής αντί να γράψουμε *-h localhost*, για παράδειγμα. Το *<username>* πρέπει να είναι το δικό μας όνομα χρήστη στην MySQL. Αν εγκαταστήσαμε εμείς οι ίδιοι τον MySQL server, αυτό θα είναι το *root*, ενώ αν χρησιμοποιούμε τον MySQL server του Web host που μας φιλοξενεί, αυτό θα πρέπει να είναι το όνομα χρήστη της MySQL που μας έχει δοθεί. Το όρισμα *-p* λέει στο πρόγραμμα να ζητήσει από μας τον κωδικό εισόδου (password), το οποίο θα συμβεί μόλις δώσουμε την παραπάνω εντολή. Αν έχουμε εγκαταστήσει εμείς οι ίδιοι τον MySQL, αυτό το password θα είναι το root password που επιλέξαμε εμείς, ενώ αν χρησιμοποιούμε τον MySQL server του Web host που μας φιλοξενεί, αυτό θα πρέπει να είναι το password της MySQL που μας έχει δοθεί. Αν τα γράψαμε όλα σωστά, το πρόγραμμα πελάτη της MySQL θα παρουσιάσει τον εαυτό του και θα εμφανίσει την γραμμή εντολών της MySQL, ως εξής : *mysql>*

Τώρα, ο MySQL server είναι σε θέση να παρακολουθεί περισσότερες από μία βάσεις δεδομένων, που αυτό σημαίνει ότι Web host μπορεί να στήσει έναν μόνο MySQL server για να χρησιμοποιηθεί από πολλούς από τους συνδρομητές του. Έτσι, το επόμενο βήμα μας θα πρέπει να είναι να επιλέξουμε την βάση δεδομένων με την οποία θα δουλέψουμε. Πρώτα απ' όλα, θα δούμε μια λίστα των βάσεων δεδομένων που υπάρχουν στον τρέχοντα server. Δίνουμε την επόμενη εντολή και μετά ENTER.

mysql> SHOW DATABASES;

Η MySQL θα εμφανίσει μια λίστα με τις βάσεις δεδομένων που υπάρχουν στον server, ως εξής :

```
Database  
mysql  
test  
rows in set (0.11 sec)
```

Αυτό ήταν ένα παράδειγμα για να καταλάβουμε τον τρόπο λειτουργίας της **Βάσης Δεδομένων MySQL** .

Κεφάλαιο 4:Application

4.1Πρόλογος

Στα παραπάνω κεφάλαια κάναμε μια αναλυτική επισκόπηση όλων των τεχνολογιών που χρησιμοποιήσαμε για την υλοποίηση της εφαρμογής μας. Παρακάτω θα αναλύσουμε την εφαρμογή μας . Η ανάλυση αυτή περιέχει τις λειτουργίες της εκτενώς, τη διασύνδεση με τη βάση δεδομένων , screenshots από την εφαρμογή , τα εργαλεία που χρησιμοποιήθηκαν για την υλοποίηση της και γενικά μια αναλυτική παρουσίαση της εφαρμογής τόσο σε επίπεδο χρήστη όσο και σε επίπεδο προγραμματιστικό.

Στόχος της εφαρμογής σε επίπεδο χρήστη είναι η παρουσίαση κάποιων βιβλίων σε 3D μορφή και interaction με αυτά όπως πλοήγηση , άνοιγμα του κάθε PDF και άλλα. Σε προγραμματιστικό επίπεδο στόχος μας είναι να συνδυάσουμε τη Java3D τη Java Swing και βάσεις σε MySQL , σε μία εφαρμογή η οποία έχει πλουραλισμό αντικειμένων , μεθόδων τόσο σε Java3D όσο και σε Java Swing όλα αυτά συνδεδεμένα με τη MySQL DataBase σαν μια πηγή δεδομένων.

Αλλά καλύτερα ας τα δούμε πιο αναλυτικά παρακάτω όπου αρχικά θα αναφερθούμε στα εργαλεία που χρησιμοποιήθηκαν για την ανάπτυξη της εφαρμογής θα συνεχίσουμε με την ανάλυση της ίδιας της εφαρμογής και τέλος θα παρουσιάσουμε κάποιες εικόνες από την εφαρμογή.

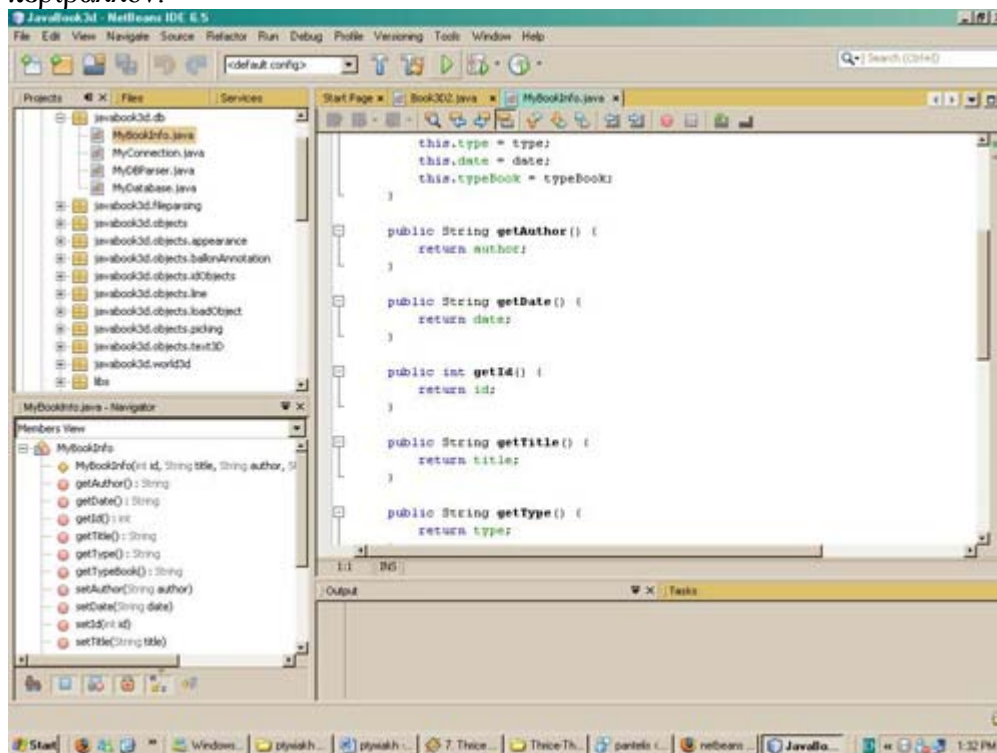
4.2 Εργαλεία

Για την υλοποίηση μιας τέτοιας εφαρμογής πέρα από ένα IDE για τον προγραμματισμό σε java και ένα Query Browser για τη MySQL χρειάστηκαν και άλλα εργαλεία τα οποία είναι εξίσου σημαντικά για το εμφανισιακό κομμάτι της εφαρμογής την κατασκευή των βιβλίων κλπ . Παρακάτω παραθέτουμε τα εργαλεία που χρησιμοποιήθηκαν , μια μικρή περιγραφή για το καθένα και που χρησιμοποιήθηκε .

NetBeans IDE 6.5

Το NetBeans είναι ένα Open-Source IDE (integrated development environment) της Sun Microsystems της εταιρίας δηλαδή που έφτιαξε τη java.

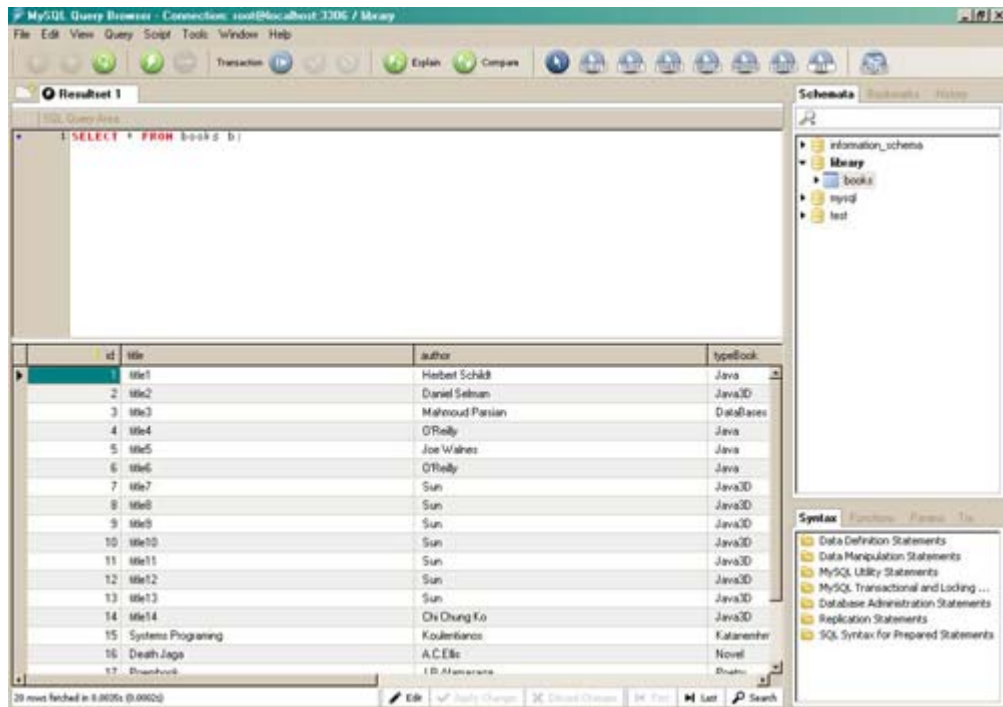
Όλη μας η εφαρμογή είναι γραμμένη σε αυτό το περιβάλλον.



Μπορεί κανείς εύκολα να προμηθευτεί το NetBeans από τη σελίδα της sun www.netbeans.org .

MySQL GUI Tools 5.0

Εργαλεία διαχείρισης της MySQL database σε Graphical Users Interface και όχι σε γραμμή εντολών. Προφανώς για κάποιον που θέλει να γράψει τα πάντα ή κάποια με εντολές δεν αλλάζει κάτι , απλά είναι μεγάλη διευκόλυνση να διαχειρίζεται κανείς τη βάση δεδομένων σε γραφικό περιβάλλον παρά σε γραμμή εντολών.



Η Βάση Δεδομένων θα αναλυθεί σε παρακάτω ενότητα .

3D Studio max

Το 3DS Max είναι το τρίτο ευρέως χρησιμοποιούμενο 3D animation πρόγραμμα δημιουργίας περιεχομένου από επαγγελματίες ,σύμφωνα με την έκθεση Roncarelli .Έχει πολλές δυνατότητες modeling, μια ευέλικτη αρχιτεκτονική plug-in και μια παράδοση στην πλατφόρμα Microsoft Windows. Χρησιμοποιείται πιο πολύ από Game Developers, στην τηλεόραση και σε studio αρχιτεκτονικής παρουσίασης.

Στην εφαρμογή μας το χρησιμοποιήσαμε για τη σχεδίαση του 3D βιβλίου μας όπως φαίνεται παρακάτω.

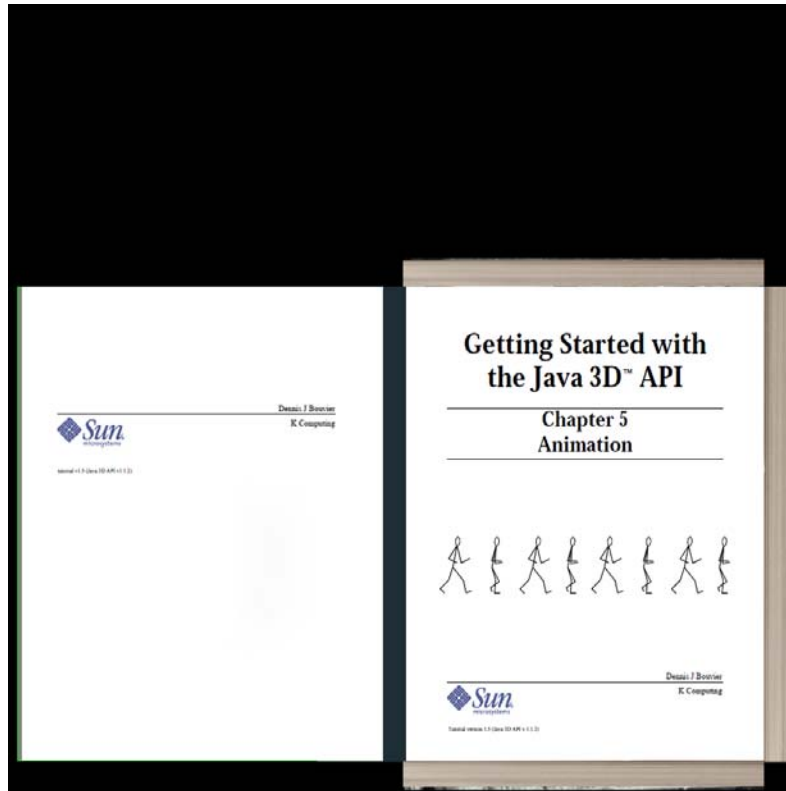


Αυτό είναι το αρχικό σχήμα του βιβλίου χωρίς εξώφυλλα. Με την κατάλληλη επεξεργασία που θα δούμε στο παρακάτω εργαλείο τοποθετούμε τα εξώφυλλα και όταν φορτώνουμε το βιβλίο στην εφαρμογή μας φαίνεται κανονικά ντυμένο.

Adobe Photoshop CS3

Το Photoshop είναι το γνωστό σε όλους μας πρόγραμμα επεξεργασίας εικόνων. Έχει πάρα πολλές επιλογές και είναι για πολλούς το καλύτερο πρόγραμμα επεξεργασίας εικόνων. Χρησιμοποιείται από επαγγελματίες φωτογράφους και web designers , μέχρι και απλούς χρήστες.

Εμείς χρησιμοποιήσαμε αρχικά το εργαλείο αυτό για να « ντύσουμε » το βιβλίο μας και ύστερα για το φόντο της εφαρμογής και την επεξεργασία των φωτογραφιών του κάθε βιβλίου. Μετά το Photoshop το βιβλίο έγινε όπως παρακάτω και αυτό το κάνουμε για κάθε βιβλίο που θέλουμε να προσθέσουμε στην εφαρμογή μας .



4.3 Η εφαρμογή

4.3.1:Εγκατάσταση Εφαρμογής

Για να τρέξει κάποιος την εφαρμογή στο δικό του υπολογιστή χρειάζεται να κάνει κάποια configurations με συγκεκριμένα βήματα . Παρακάτω θα παραθέσουμε την εγκατάσταση σε λειτουργικό Microsoft Windows XP από την αρχή με δεδομένο ότι κάποιος δεν έχει τις απαραίτητες γνώσεις.

Αρχικά εγκαθιστούμε το jdk-6u7-windows-i586-p το οποίο περιέχει και το Jre 1.6. Αυτό μπορεί κανείς να το βρει εύκολα στη σελίδα <http://java.sun.com/javase/downloads/index.jsp> . Είναι ένα .exe αρχείο με εύκολη εγκατάσταση.

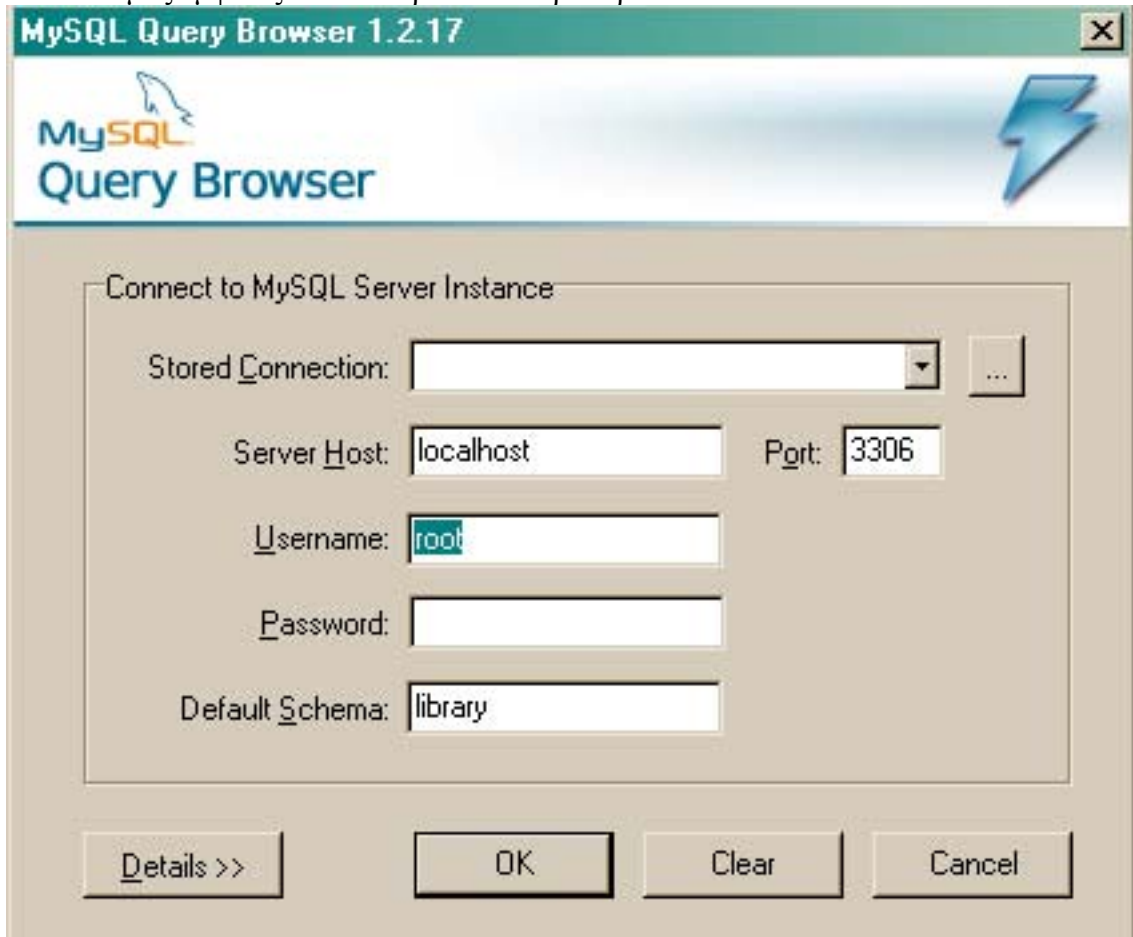
Αφού έχουμε εγκαταστήσει το jdk εγκαθιστούμε και τις βιβλιοθήκες της java3D και αυτές με ένα αρχείο .exe το οποίο ονομάζεται j3d-1_5_2-windows-i586 και μπορεί κανείς να βρει στη σελίδα <https://java3d.dev.java.net/binary-builds.html>

Έπειτα μπορούμε να εγκαταστήσουμε το NetBeans IDE 6.5.Και αυτό το βρίσκουμε εύκολα στη σελίδα www.netbeans.org.

Αφού τελειώσει και η εγκατάσταση αυτή εγκαθιστούμε τη MySql από το αρχείο mysql-essential-5.1.34-win32 . Εδώ έχουμε μια ιδιαίτερη εγκατάσταση. Αφού τρέξουμε κανονικά το .exe αυτό επιλέγουμε Next την πρώτη φορά Complete installation στη δεύτερη σελίδα μετά Install . Με το τελείωμα της εγκατάστασης επιλέγουμε το Configure now πατάμε Next και μετά Configure

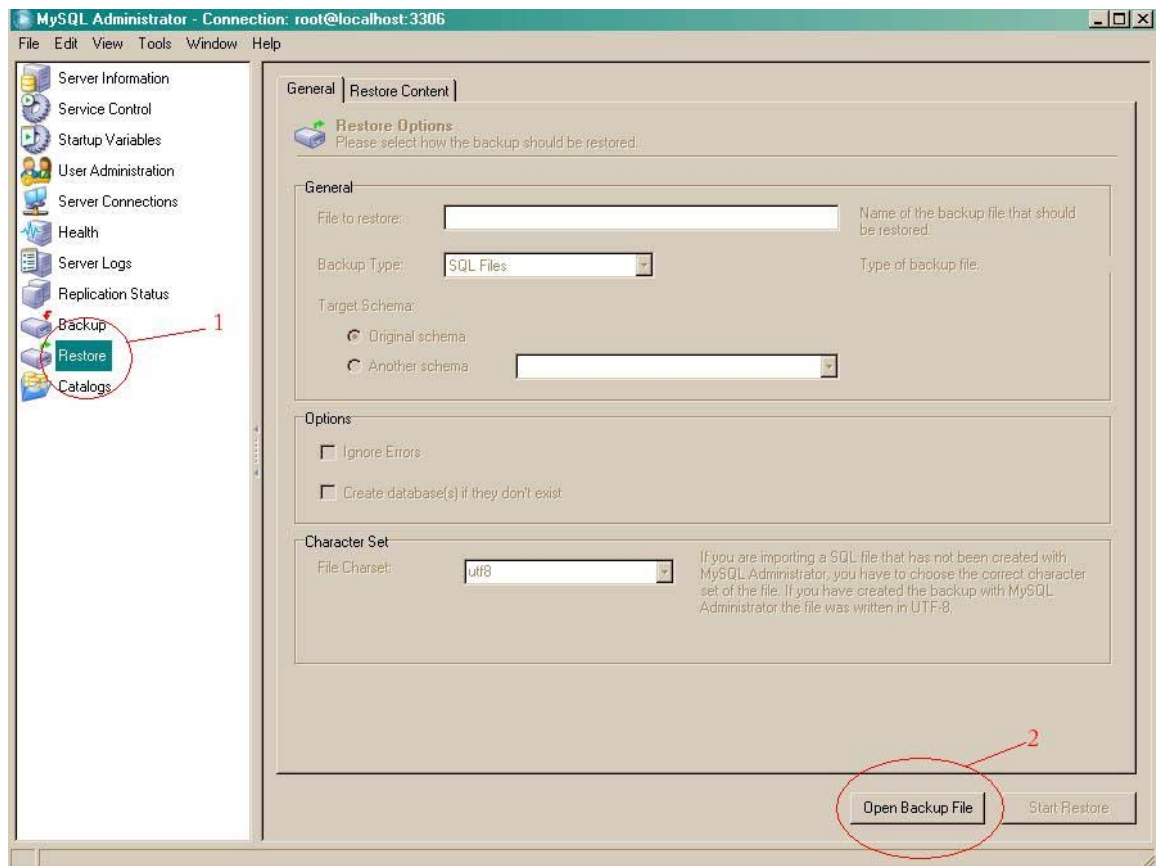
Instance και μετά δίνουμε τον κωδικό “vortex” και execute .Αφού τελειώσει το execution πατάμε finish.

Όταν τελειώσουμε με την εγκατάσταση του MySql Server συνεχίζουμε με την εγκατάσταση των MySQL GUI Tools 5.0 που κατεβάζουμε από τη σελίδα <http://dev.mysql.com/downloads/gui-tools/5.0.html> . Η εγκατάσταση είναι απλή δεν έχει καμία δυσκολία . Όταν τελειώσει τρέχουμε το Query Browser όπου μας εμφανίζεται το παρακάτω παράθυρο.



Συμπληρώνουμε τα πεδία όπως στην εικόνα και στον κωδικό βάζουμε “vortex” .Μετά θα μας ρωτήσει αν θέλουμε να κατασκευάσει το library και εμείς επιλέγουμε OK. Αφού τελειώσει και αυτό μπαίνουμε στον browser. Τώρα όμως έχουμε μια κενή Βάση.

6. Τώρα χρειαζόμαστε τη βάση που έχουμε φτιάξει να τη φορτώσουμε εδώ. Κλείνουμε το Query Browser και ανοίγουμε το MySql Administrator. Δίνουμε πάλι κωδικούς κλπ και μας εμφανίζεται η παρακάτω οθόνη:

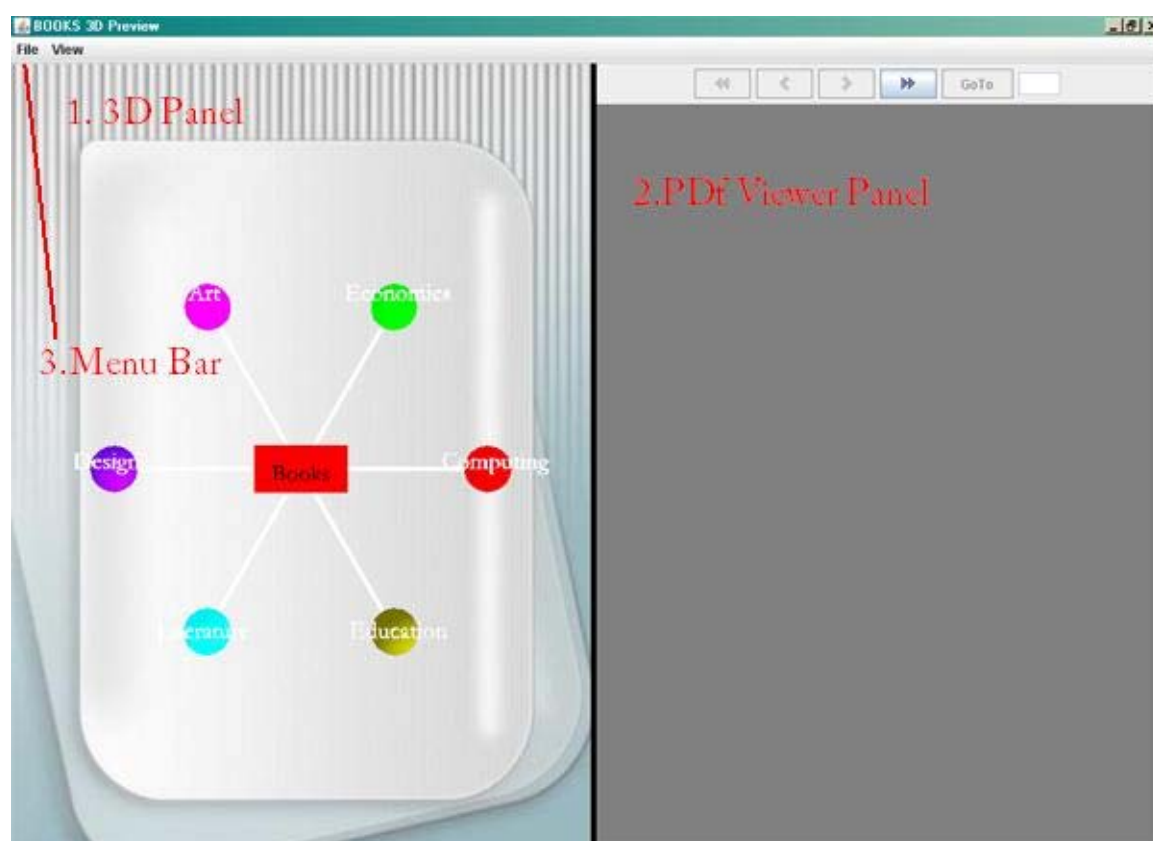


Επιλέγουμε το Restore και μετά Open Backup File.Μας εμφανίζεται ένα File Chooser και πάμε και κάνουμε Open το Backup της Βάσης μας(το οποίο έχουμε δημιουργήσει και υπάρχει μαζί με το Project) .Μετά πατάμε Start Restore.Αφού τελειώσει το restore ανοίγουμε το Query Browser και επιλέγουμε το library δεξιά και βλέπουμε ότι έχει το books από κάτω. Αν διπλοκλικάρουμε το books και πατήσουμε το execute query εμφανίζεται η βάση μας.

7. Αφού τελειώσουν όλα αυτά ανοίγουμε το NetBeans και πατάμε open Project και επιλέγουμε το project μας και πατάμε run και τρέχει κανονικά .

4.3.2 : Application Analysis

Η εφαρμογή τρέχει όπως είπαμε πιο πάνω και αφού τρέξει εμφανίζεται το παρακάτω JFrame:



4.3.2.1 Panel 3D

Το 3D Panel θα το χωρίσουμε σε 2 μέρη , στο εμφανισιακό και στο λειτουργικό

4.3.2.1.1 Εμφανισιακό Μέρος

Αριστερά είναι το 3D Panel μας με τις κατηγορίες βιβλίων που περιέχονται στη βάση δεδομένων .Ανάλογα με το πόσες διαφορετικές (πεδίο type) εγγραφές έχει στη βάση (δλδ πόσες διαφορετικές κατηγορίες) χωρίζεται και ο κύκλος . Αυτό γίνεται ως εξής: Μετράμε τις διαφορετικές κατηγορίες και παίρνουμε το πλήθος τους – στην προκειμένη 6 - ,το διαιρούμε με τις συνολικές μοίρες του κύκλου και βρίσκουμε τη γωνία μεταξύ 2 κατηγοριών .Αυτό γίνεται δυναμικά δλδ αν τώρα προσθέσουμε νέες κατηγορίες στο πεδίο

type και ξανατρέξουμε την εφαρμογή θα αλλάξει αυτόματα .Οι γραμμές σχεδιάζονται επίσης δυναμικά με βάση αυτό που είπαμε πριν . Παρακάτω θα παραθέσουμε τον κώδικα για το πώς γίνεται ο έλεγχος για τις νέες κατηγορίες ο οποίος βρίσκεται στην κλάση CreateObjects .

Όπως φαίνεται και από την εικόνα κάθε κατηγορία έχει και διαφορετικό χρώμα . Και αυτό γίνεται δυναμικά . Το background έχει φτιαχτεί με το Photoshop και εμφανίζεται με το ξεκίνημα της εφαρμογής.

Οι γραμμές κατασκευάζονται και αυτές στην CreateObjects στην παρακάτω μέθοδο :

```
private void addLine(Vector3d translation) {
    CreateLineArguments lineArguments = new CreateLineArguments(new Point3d
    MyLine myLine = new MyLine(null, lineArguments.getMyLine(), lineArgument
    rootBG.addChild(myLine);
}
```

Η μέθοδος αυτή χρησιμοποιεί την CreateLineArguments κλάση η οποία κατασκευάζει γραμμές με τα ορίσματα που τις δίνουμε για τη θέση της γραμμής το πάχος της γραμμής κλπ. Αφού κατασκευαστεί η γραμμή με βάση τα στοιχεία που έχει το vector που περάσαμε σαν όρισμα στην addLine κατασκευάζουμε instance της δικιάς μας κλάσης MyLine η οποία έχει σκοπό να δώσει Id (identification για να ξεχωρίζει η μία από την άλλη) στην κάθε γραμμή που κατασκευάζεται για τη μετέπειτα χρησιμοποίησή της

```
private String[] getTypesCategories() {
    Vector<String> typesCategories = new Vector<String>();
    for (int i = 1; i < booksInfos.length; i++) {
        boolean isAlreadyAdded = false;
        for (int j = 0; j < typesCategories.size(); j++) {
            String type = (String) (typesCategories.get(j));
            if (type.equals(booksInfos[i].getType())) {
                isAlreadyAdded = true;
            }
        }
        if (!isAlreadyAdded) {
            typesCategories.add(booksInfos[i].getType());
        }
    }
    return convertVectorToStringArray(typesCategories);
}
```

Ο κώδικας αυτός ελέγχει και επιστρέφει πίνακα με τις διαφορετικές κατηγορίες του πεδίου type της βάσης δεδομένων. Αυτό θα χρησιμοποιηθεί μετά για να γίνει έλεγχος και μετά να τοποθετηθούν οι κατηγορίες

id	title	author	typeBook	date	type
5	title5	Joe Walnes	Java	4564	Computing
6	title6	O'Reilly	Java	435	Computing
7	title7	Sun	Java3D	1999	Computing
8	title8	Sun	Java3D	1999	Computing
9	title9	Sun	Java3D	1999	Computing
10	title10	Sun	Java3D	2000	Computing
11	title11	Sun	Java3D	2000	Computing
12	title12	Sun	Java3D	2000	Computing
13	title13	Sun	Java3D	2000	Computing
14	title14	Ch Chung Ko	Java3D	2000	Computing
15	Systems Programming	Koustenianos	Katalenimena	2009	Education
16	Death Jaga	A.C.E.B.	Novel	2004	Literature
17	Poembook	J.R.Atemasaga	Poetry	1999	Literature
18	Home Design	asf90	Architecture	2000	Design
19	mlia1	mliaos	painter	1999	Art
20	mlia2	mliaos	finance	2006	Economic

Μετά ο παρακάτω κώδικας τοποθετεί τις κατηγορίες τις μπάλες τις γραμμές και τα 3D Texts στις θέσεις που βλέπουμε όπως ακριβώς είπαμε και αυτός βρίσκεται στην κλάση CreateObjects. Όπως είπαμε κατασκευάζει και τοποθετεί τις μπάλες κάθε μία με ξεχωριστό χρώμα ανάλογα με την κατηγορία. Επίσης

```

int n = types.length;
double currentAngle, stepAngle, radius = scale * 4;
stepAngle = 2 * Math.PI / n;
int count=0;
for (int i = 0; i < n; i++) {
    Vector3d translation = new Vector3d();
    currentAngle = stepAngle * i;
    translation.x = Math.cos(currentAngle) * radius;
    translation.z = Math.sin(currentAngle) * radius;
    Transform3D transform3D = new Transform3D();
    transform3D.setScale(scale);
    Transform3D transform3DTranslation = new Transform3D();
    transform3DTranslation.setTranslation(translation);
    transform3D.mul(transform3DTranslation);

    TransformGroup transformGroupRoot = new TransformGroup(transform3D);
    transformGroupRoot.setCapability(TransformGroup.ALLOW_CHILDREN_EXTEND);
    transformGroupRoot.setCapability(TransformGroup.ALLOW_CHILDREN_WRITE);
    transformGroupRoot.setCapability(TransformGroup.ALLOW_CHILDREN_READ);
    transformGroupRoot.setCapability(TransformGroup.ALLOW_TRANSFORM_WRITE);
    transformGroupRoot.setCapability(TransformGroup.ALLOW_TRANSFORM_READ);

    SphereId sphere = new SphereId("type" + i, (float) (scale / 2));
    Appearance app = new AppearanceObject().createAppearance(colors[count]);

```

Όπως είδαμε στην αρχική σελίδα υπάρχει ένα παραλληλόγραμμα το οποίο γράφει Books. Αυτό φτιάχνεται επίσης στην CreateObjects και πιο συγκεκριμένα στη μέθοδο addCenterBox της οποίας ο κώδικας είναι ο παρακάτω. Εδώ φτιάχνεται το αντικείμενο αυτό από ένα αντικείμενο τύπου Box και καθορίζουμε όλες τις ιδιότητες του αντικειμένου αυτού. Θέση, χρώμα, 3D text, μέγεθος τα πάντα καθορίζονται εδώ. Το αντικείμενο αυτό δεν είναι interactive.

```

private void addCenterBox() {
    Box boxCenter = new Box((float) (2 * scale), 0, (float) (scale), new
        rootBG.addChild(boxCenter);
    Transform3D transform3D = new Transform3D();
    transform3D.setTranslation(new Vector3d(0, 0, scale / 2));
    Transform3D transform3DrotX = new Transform3D();
    transform3DrotX.rotX(-Math.PI / 2);
    transform3D.mul(transform3DrotX);
    Transform3D transform3Dscale = new Transform3D();
    transform3Dscale.setScale(scale);
    transform3D.mul(transform3Dscale);
    TransformGroup textTG = new TransformGroup(transform3D);
    textTG.addChild(new MyText3D().getText3D(null, "Books"));

    Transform3D transform3D1 = new Transform3D();
    transform3D1.setTranslation(new Vector3d(0, scale / 10, 0));
    TransformGroup tg = new TransformGroup(transform3D1);
    tg.addChild(textTG);
    tg.addChild(boxCenter);
    rootBG.addChild(tg);
}

```

Η κλάση CreateObjects φτιάχνει ουσιαστικά και τοποθετεί όλα τα αντικείμενα στις θέσεις που βλέπουμε .Η κλάση αυτή χρησιμοποιείται στην κλάση που φτιάχνουμε το JFrame. Παραθέτουμε τον κώδικα από την κλάση του JFrame

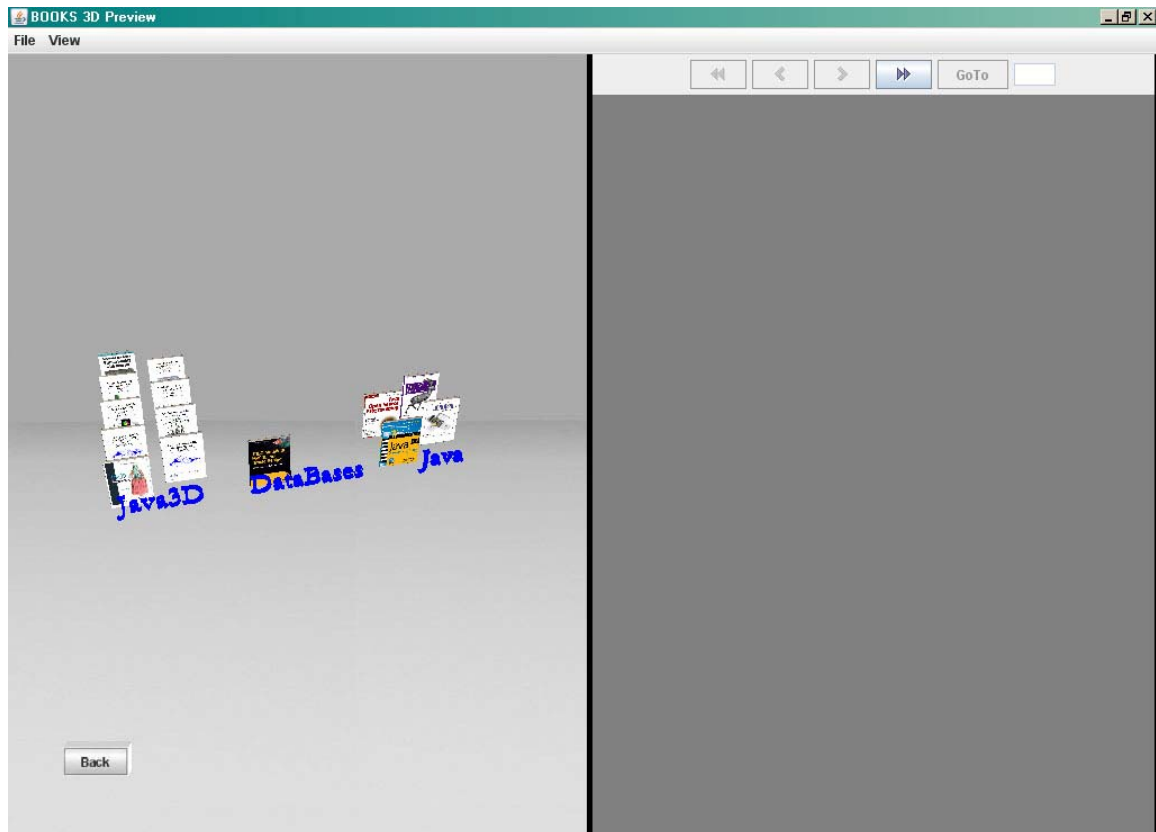
```

private void loadCategories() {
    URL backgroundImage = Resources.getResource("icons/1.jpg");
    TextureLoader backgroundTexture = new TextureLoader(backgroundImage, null);
    simpleUniverse.getBackground().setImage(backgroundTexture.getImage());
    if (createObjects != null && createObjects.getRootBG().isLive()) {
        simpleUniverse.getLocale().removeBranchGraph(createObjects.getRootBG());
    }
    if (backButtonBG != null && backButtonBG.isLive()) {
        simpleUniverse.getLocale().removeBranchGraph(backButtonBG);
    }
    defaultPos = new Point3d(0, 50, 1);
    setDefaultCameraPos();
    createObjects = new CreateObjects(booksInfos);
    BranchGroup rootBG = createObjects.setTypesPositions();
    categoryDetails = new BranchGroup[createObjects.getTypes().length];
    for (int i = 0; i < categoryDetails.length; i++) {
        categoryDetails[i] = new BranchGroup();
    }
    simpleUniverse.getLocale().addBranchGraph(rootBG);
}

```

Εδώ φορτώνουμε τις κατηγορίες έτσι όπως περιγράψαμε πριν, τοποθετούμε το background τη θέση της κάμερας κλπ.

Μπαίνοντας σε μία από τις κατηγορίες για παράδειγμα στην κατηγορία εμφανίζονται τα βιβλία όπως στην παρακάτω εικόνα. Επίσης εμφανίζεται ένα κουμπί Back το οποίο είναι ένα Box με texture το οποίο μας πηγαίνει στην αρχική σελίδα.



Όπως βλέπουμε από την εικόνα τα βιβλία χωρίζονται σε υποκατηγορίες. Και αυτός ο χωρισμός γίνεται από τη βάση δεδομένων ως εξής : Υπάρχει ένα πεδίο που ονομάζεται typeBook το οποίο περιέχει τις υποκατηγορίες που ανήκουν τα βιβλία .

id	title	author	typeBook	date	type
5	title5	Joe Walnes	Java	4564	Computing
6	title6	O'Reilly	Java	435	Computing
7	title7	Sun	Java3D	1999	Computing
8	title8	Sun	Java3D	1999	Computing
9	title9	Sun	Java3D	1999	Computing
10	title10	Sun	Java3D	2000	Computing
11	title11	Sun	Java3D	2000	Computing
12	title12	Sun	Java3D	2000	Computing
13	title13	Sun	Java3D	2000	Computing
14	title14	Chi Chung Ko	Java3D	2000	Computing
15	Systems Programing	Kouleptianos	Katanemhmena	2009	Education
16	1 ^η κατηγοries poy	A.C.Ellis	Novel	2004	Literature
17	1 ^η emfanizontai	J.R.Afamasaga	Poetry	1888	Literature
18	1 ^η sthn eikona	asdf98	Architecture	2000	Design
19		mitsos	painters	1899	Art
20		mitsosos	finance	2006	Economics

Η ταξινόμηση γίνεται ως εξής: Όπως βλέπουμε και στη βάση κάθε εγγραφή έχει ένα id ένα typeBook και ένα type .Με βάση τα typeBook και type γίνεται ο χωρισμός. Το id το χρησιμοποιούμε για να ξεχωρίζουμε τα βιβλία μεταξύ τους. Γενικά κάθε αντικείμενο της εφαρμογής μας έχει id. Το id των books το τραβάμε από τη βάση. Συγκεκριμένα όποια typeBooks ανήκουν στο type εμφανίζονται μέσα σε αυτό. Ο έλεγχος αυτός γίνεται και πάλι μέσα στην CreateObjects και συγκεκριμένα στην setTypesBookPos η οποία (CreateObjects)καλείται ύστερα μέσα στο

JFrame(Books3D class). Εδώ φαίνεται ο κώδικας που κάνει την ταξινόμηση μέσα σε κάθε υποκατηγορία

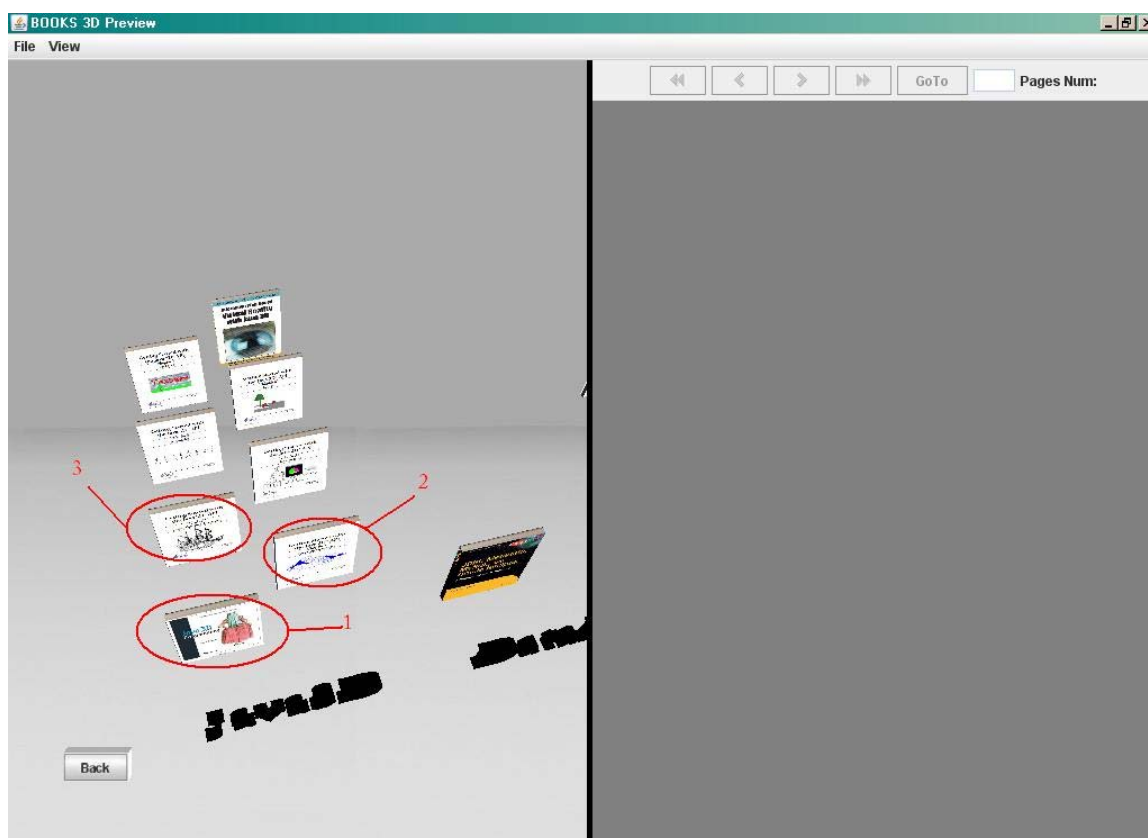
```
private void setTypeBookPos(String[] types, String typeGroup) {
    System.out.println(types.length);
    object3d = new LoadObj().loadObj();
    scaleBooks = 5f / types.length;
    if (scaleBooks > 1) {
        scaleBooks = 1;
    }
    System.out.println("scaleBooks = " + scaleBooks);
    Vector3d translationRoot = new Vector3d(-types.length / 2 * scaleBooks * 5, 0, 0);
    Transform3D transform3DRoot = new Transform3D();
    transform3DRoot.setTranslation(translationRoot);
    TransformGroup rootTransformGroup = new TransformGroup(transform3DRoot);
    rootBG.addChild(rootTransformGroup);
    for (int j = 0; j < types.length; j++) {
        int bookPerCategory = 0;
        Vector3d translation = new Vector3d(j * scaleBooks * 5, 0, 0);
        Transform3D transform3D = new Transform3D();
        transform3D.setTranslation(translation);
        TransformGroup tg = new TransformGroup(transform3D);
        transform3D = new Transform3D();
        transform3D.setTranslation(new Vector3d(0, -0, 1.5 * scaleBooks));

        Transform3D scaleTransform3D = new Transform3D();
        scaleTransform3D.setScale(scaleBooks);
        transform3D.mul(scaleTransform3D);

        TransformGroup tgTitle = new TransformGroup(transform3D);
        tgTitle.addChild(new MyText3D().getText3D(null, types[j], new AppearanceObject().createAppearance(new C
        tg.addChild(tgTitle);
    }
    for (int i = 0; i < booksInfos.length; i++) {
        if (booksInfos[i].getType().equals(typeGroup) && booksInfos[i].getTypeBook().equals(types[j])) {
            Vector3d translationBook = new Vector3d();
            if (bookPerCategory % 2 == 1) {
                translationBook = new Vector3d(scaleBooks, 0, -scaleBooks * bookPerCategory);
            } else {
                translationBook = new Vector3d(-scaleBooks, 0, -scaleBooks * bookPerCategory);
            }
            Transform3D transform3DrotY = new Transform3D();
            transform3DrotY.rotY(-Math.PI / 2);
            Transform3D transform3DBook = new Transform3D();
            transform3DBook.setTranslation(translationBook);
            transform3DBook.mul(transform3DrotY);
            Transform3D transform3Scale = new Transform3D();
            transform3Scale.setScale(scaleBooks);
            transform3DBook.mul(transform3Scale);
            TransformGroup transformGroup = new TransformGroup(transform3DBook);
            transformGroup.setCapability(TransformGroup.ALLOW_TRANSFORM_WRITE);
            transformGroup.setCapability(TransformGroup.ALLOW_TRANSFORM_READ);
            transformGroup.setCapability(TransformGroup.ENABLE_PICK_REPORTING);
            BranchGroup bg = new BranchGroup();
            bg.addChild(transformGroup);
            PickRotateBehavior pickRotateBehavior = new PickRotateBehavior(bg, canvas3D, new BoundingSphere
            rootBG.addChild(pickRotateBehavior);
            NodeId book3D = new NodeId("obj " + i, (Shape3D) object3d.cloneNode(true)); //cloneNode
            book3D.setAppearance(new AppearanceObject().getBookImageApp(i)); // fortunel ta texture images s
            transformGroup.addChild(book3D);
            tg.addChild(bg);
            bookPerCategory++;
        }
    }
}
```

Σε αυτή τη μέθοδο γίνονται οι έλεγχοι για τις διάφορες κατηγορίες (type) και μετά οι έλεγχοι για τις υποκατηγορίες και μετά για όλα τα βιβλία που ανήκουν στην κάθε υποκατηγορία υπολογίζεται η θέση τους και μετά κατασκευάζονται με την εικόνα που αντιστοιχεί στο κάθε ένα και εμφανίζονται ντυμένα. Επίσης το κάθε Book με τη βοήθεια της κλάσης NodeId γίνεται interactive σε όλες τις πλευρές του. Αυτό γίνεται για όλες τις υποκατηγορίες μέχρι να κατασκευαστούν και τοποθετηθούν όλα τα βιβλία .

Η διάταξη των βιβλίων γίνεται ως εξής. Αφού γίνει ο έλεγχος για την κατηγορία (type) και την υποκατηγορία (typeBook) κάνουμε ένα έλεγχο για το αν είναι μονός η ζυγός ο αριθμός που βρίσκεται στη συγκεκριμένη κατηγορία και αναλόγως τοποθετούμε το βιβλίο. Αν είναι μονός τοποθετείται στη θέση 1 και αυξάνεται ένας μετρητής για τον επόμενο έλεγχο. Αν το επόμενο βιβλίο είναι στην ίδια κατηγορία και υποκατηγορία έχει ζυγό αριθμό και τοποθετείται με βάση τον έλεγχο που κάνουμε (και τον αυξημένο κατά ένα μετρητή) στη θέση 2. Ο μετρητής αυξάνεται πάλι και το επόμενο βιβλίο μπαίνει στο ίδιο επίπεδο στην ίδια σειρά αλλά πιο μέσα στη θέση 3 και αυτό συνεχίζεται για όλα τα βιβλία της ίδιας κατηγορίας μέχρι να τελειώσουν και συνεχίζεται στην επόμενη υποκατηγορία. Η διαδικασία αυτή παίρνει τις εγγραφές από τη βάση δεδομένων. Από εκεί παίρνει τις υποκατηγορίες και κατηγορίες των βιβλίων και κάνει τους ελέγχους.



Επίσης παρατηρούμε ότι το background έχει αλλάξει. Όλα αυτά γίνονται μέσα στο JFrame(Book3D() class) και πιο συγκεκριμένα στη μέθοδο loadBooks

```

private void loadBooks(String typeName) {
    URL backgroundImage = Resources.getResource("icons/2.jpg");
    TextureLoader backgroundTexture = new TextureLoader(backgroundImage, null);
    simpleUniverse.getBackground().setImage(backgroundTexture.getImage());
    hideAllCategoriesDetails();
    simpleUniverse.getLocale().removeBranchGraph(createObjects.getRootBG());
    createObjects = new CreateObjects(booksInfos);
    createObjects.setTypesBookPositions(typeName, simpleUniverse.getCanvas3D());
    BranchGroup rootBG = createObjects.getRootBG();
    bookDetails = new BranchGroup[booksInfos.length];
    bookSphere = new BranchGroup[booksInfos.length];
    for (int i = 0; i < bookDetails.length; i++) {
        bookDetails[i] = new BranchGroup();
        bookSphere[i] = new BranchGroup();
    }
    defaultPos = new Point3d(-5, 15, 20); // allazei to deutero camera
    setDefaultCameraPos();
    simpleUniverse.getLocale().addBranchGraph(rootBG);
    addBackButton();
}
}

```

Η μέθοδος αυτή φορτώνει το καινούριο background ,το οποίο όπως και το προηγούμενο όπως και όλες εικόνες και τα PDF βρίσκονται στο resources, για το οποίο θα μιλήσουμε εκτενώς αργότερα.

Μετά διώχνει την 1^η σελίδα (κατηγορίες) και φορτώνει τη δεύτερη τις υποκατηγορίες , κάνει load τα βιβλία καλώντας τις μεθόδους που περιέχονται στην CreateObjects και τα τοποθετεί στο Canvas3D. Επίσης βάζει την κάμερα στη default θέση που επιλέγουμε εμείς και μπορούμε από εκεί να αλλάξουμε εύκολα .Η default θέση είναι η θέση στην οποία βρίσκεται η κάμερα μόλις μπαίνουμε στις υποκατηγορίες το ίδιο ισχύει και για τις κατηγορίες.

Επίσης προσθέτει και το κουμπί που φαίνεται στην εικόνα καλώντας τη μέθοδο addBackButton :

```

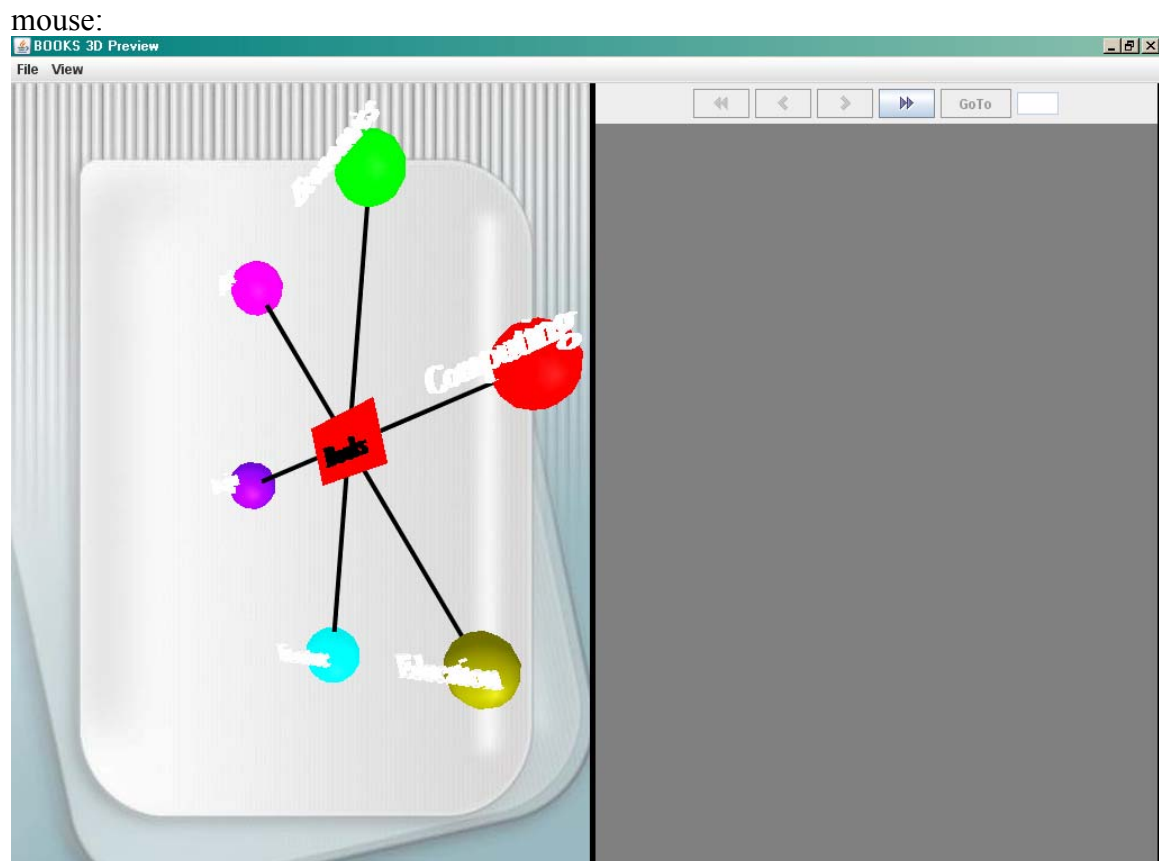
private void addBackButton() {
    backButtonBG = new BranchGroup();
    backButtonBG.setCapability(BranchGroup.ALLOW_DETACH);
    Transform3D transform3D = new Transform3D();
    transform3D.setTranslation(new Vector3d(-0.2, -0.3, -0.7)); //8esh tou button
    TransformGroup backButtonTG = new TransformGroup(transform3D);
    URL imageURL = Resources.getResource("icons/back.jpg");
    BoxId box = new ButtonId().myBoxButton("butt1", imageURL, 65, 28);
    backButtonTG.addChild(box);
    backButtonBG.addChild(backButtonTG);
    simpleUniverse.getViewTransformGroup().addChild(backButtonBG);
}
}

```

Η μέθοδος αυτή κατασκευάζει το κουμπί που βλέπουμε χρησιμοποιώντας ένα image που βρίσκεται στο resources στο icons και με την κλάση BoxId ButtonId δίνουμε τα χαρακτηριστικά της εμφάνισης του Button και γίνεται και interactive.

4.3.2.1.2 Λειτουργικό Μέρος

Αφού τελειώσαμε με το εμφανισιακό μέρος του 3D της εφαρμογής μας πάμε να δούμε και το πιο σημαντικό ίσως κομμάτι , τις λειτουργίες της .Αφού τρέξουμε την εφαρμογή εμφανιστούν οι κατηγορίες των βιβλίων έτσι όπως είδαμε πριν έχουμε τη δυνατότητα της πλοήγησης μέσα στο περιβάλλον αυτό γίνεται με όλα τα κουμπιά του

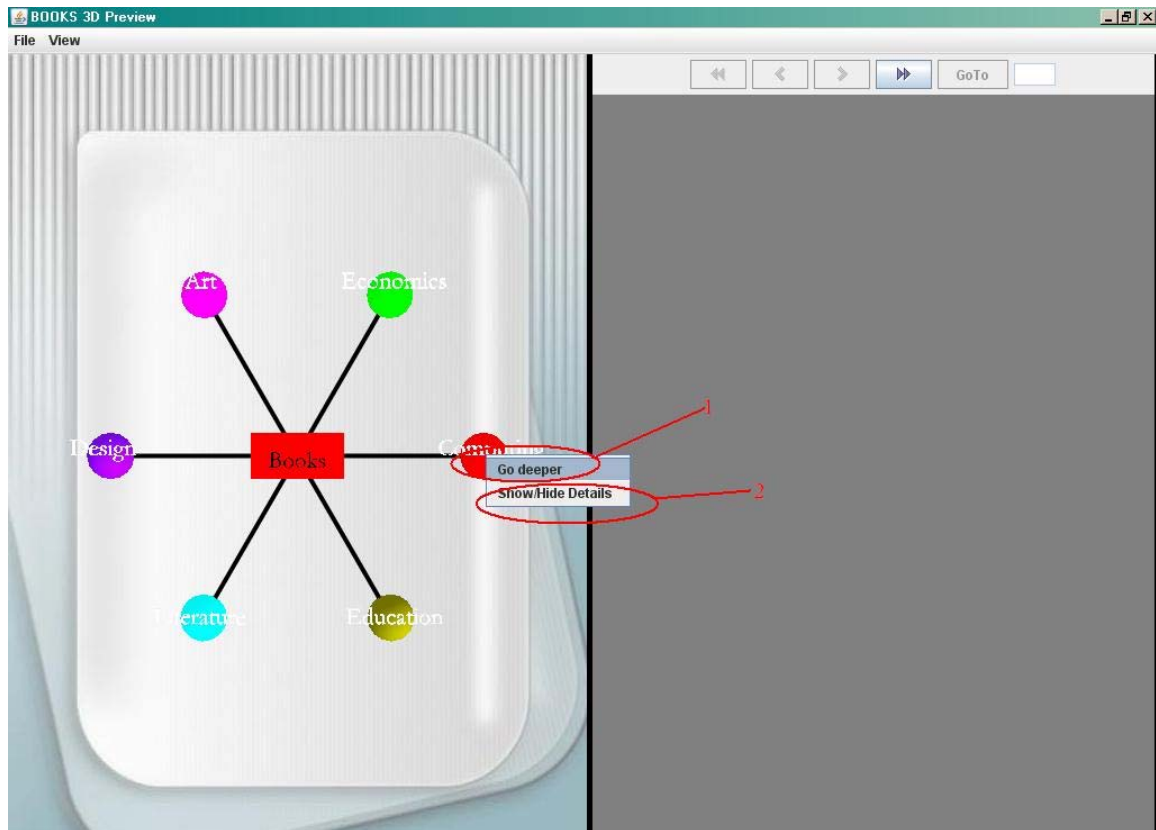


Με το αριστερό κουμπί γυρνάμε την κάμερα με το ένα της σημείο σταθερό Πατώντας τη ροδέλα και κουνώντας πάνω κάτω το ποντίκι έχουμε τη δυνατότητα zoom in-out.

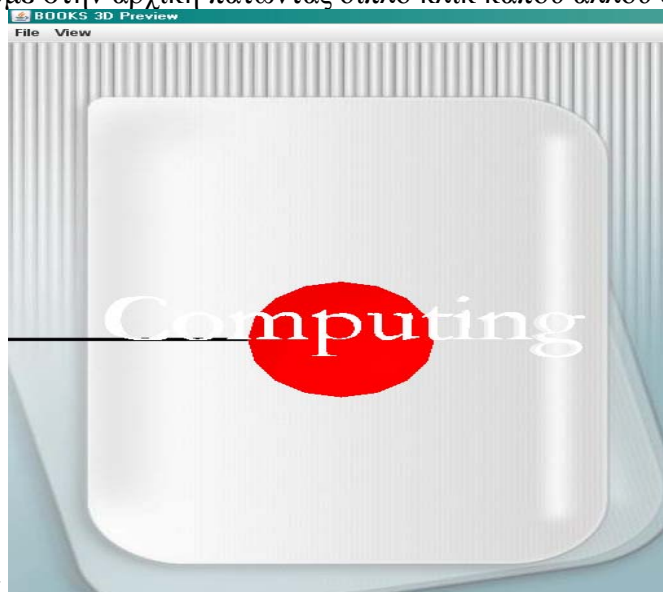
Με το δεξί κουμπί πάλι γυρνάμε την κάμερα όχι με σταθερό σημείο.

Γενικά έχουμε μια καλή πλοήγηση μέσα στο 3D περιβάλλον λόγω του μεγέθους του Panel3D είναι λίγο ευαίσθητο το ποντίκι. Τα πάντα που έχουν να κάνουν με τα πατήματα του ποντικιού και τις ενέργειες που γίνονται , βρίσκονται στη μέθοδο `mouseClicked`.

Συνεχίζουμε με το ίδιο επίπεδο και πάμε πάνω από μία οποιαδήποτε κατηγορία και πατάμε δεξί κλικ. Εμφανίζεται ένα menu με 2 επιλογές “GoDeeper” και “show/hide Details ”.

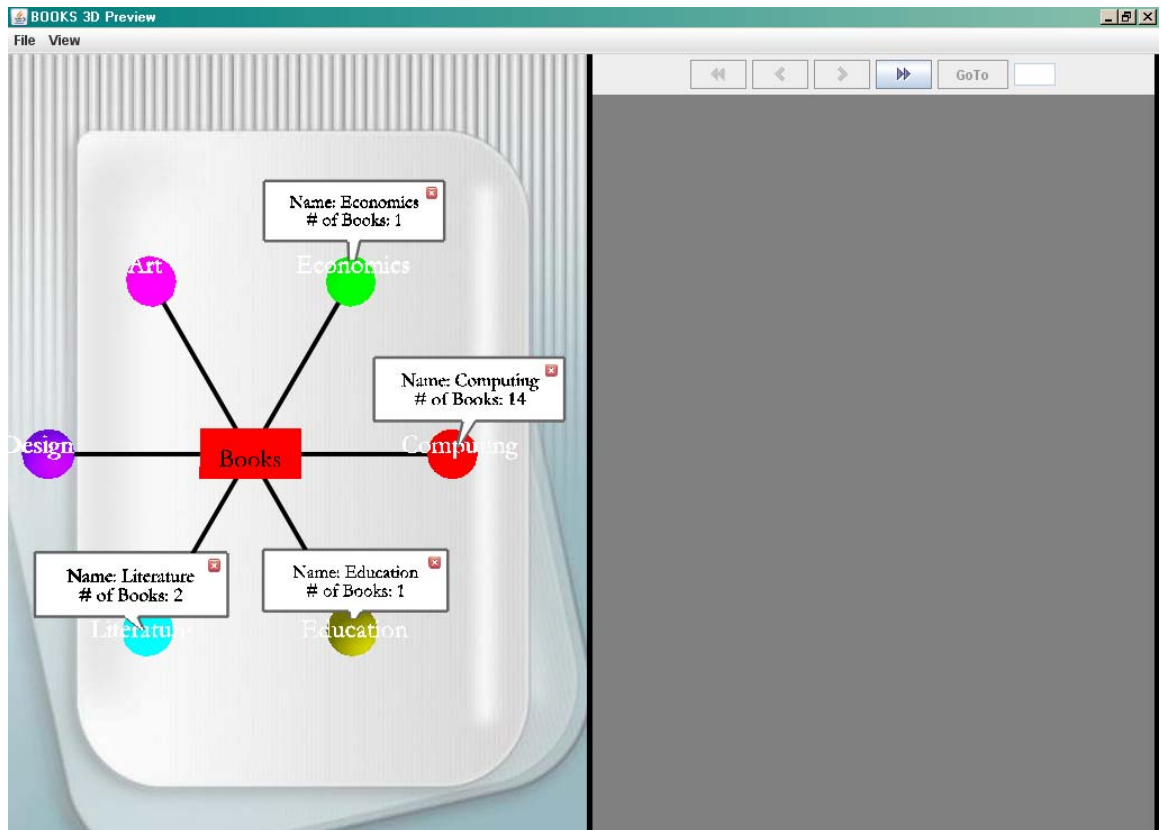


Επιλογή “GoDeeper” : κάνει zoom στην κατηγορία και μπορούμε να επιστρέψουμε στην αρχική πατώντας διπλό κλικ κάπου αλλού εκτός



κατηγορίας

Επιλογή “Show/Hide Details”: Μας εμφανίζει ένα Balloon Annotation με πληροφορίες για την εκάστοτε κατηγορία .Οι πληροφορίες είναι το όνομα της κατηγορίας και ποσά βιβλία περιέχει. Το Balloon Annotation κλείνει με το κόκκινο “x” που έχει ή πατώντας πάλι δεξί κλικ και την επιλογή “show/hide Details”.



Όπως είπαμε και πριν ότι έχει να κάνει με το ποντίκι είναι στη μέθοδο `mouse clicked`. Ο κώδικας που ακολουθεί και είναι μέσα στη `mouse clicked` κατασκευάζει τα αντικείμενα του `right click menu` και ουσιαστικά τους δίνει «ζωή» με τον `action listener`.

Αρχικά ελέγχουμε πιο κουμπί πατήθηκε και ύστερα αφού αναιρέσουμε ότι επιλογές μενού (ομοίως και με του βιβλίου μετά) είχαμε κάνει επιλεγούμε το εκάστοτε αντικείμενο με βάση το `Id`.

Φτιάχνουμε το 1^ο `menu item` και του λέμε όταν πατιέται να μεταφέρεται η κάμερα στο σημείο που έχουμε ορίσει μέσα στη μέθοδο `moveCameraInFrontOfTypeObject`. Το προσθέτουμε στο μενού και συνεχίζουμε με το άλλο.

Φτιάχνουμε το 2^ο `menu item` και του λέμε πάλι όταν πατιέται να εμφανίζει τις λεπτομέρειες της κάθε κατηγορίας με τη μέθοδο `showHideCategoryDetails`.

```

// add new menuItem
menuItem = new JMenuItem("Go deeper");
menuItem.addActionListener(new ActionListener() {

    public void actionPerformed(ActionEvent e) {
        moveCameraInFrontOfTypeObject();
    }

});
popup.add(menuItem);

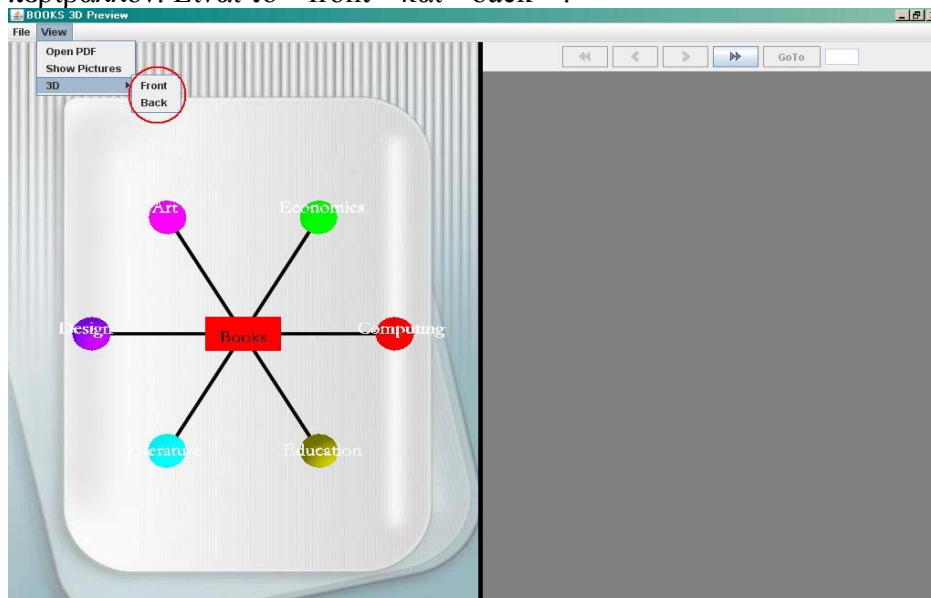
// add new menuItem
menuItem = new JMenuItem("Show/Hide Details");
menuItem.addActionListener(new ActionListener() {

    public void actionPerformed(ActionEvent e) {
        showHideCategoryDetails(intId);
    }

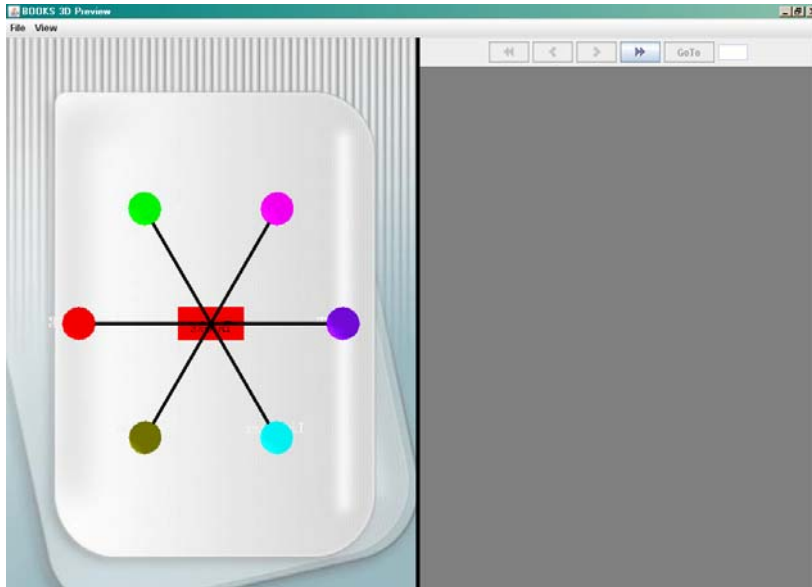
});

```

Στο μενού “View” που βλέπουμε στο πάνω μέρος της εφαρμογής μας έχουμε 2 επιλογές που μπορούμε να χρησιμοποιήσουμε στην παρούσα φάση για το 3D περιβάλλον. Είναι το “front” και “back”.



Η επιλογή back μας δείχνει τις κατηγορίες από την κάτω μεριά ενώ η front το επαναφέρει στο αρχικό point of view.



Αφού κατασκευαστούν όλα τα αντικείμενα του menu bar όπως δείχνει ο παρακάτω κώδικας προσθέτουμε στα Menu Items Action με τους ActionListeners.

```
private void setMenuBar () {
    mb = new JMenuBar ();
    //MENUS
    fileMenu = new JMenu ("File");
    pdfMenu = new JMenu ("PDF");
    j3dMenu = new JMenu ("3D");
    viewMenu = new JMenu ("View");
    //MENU ITEMS

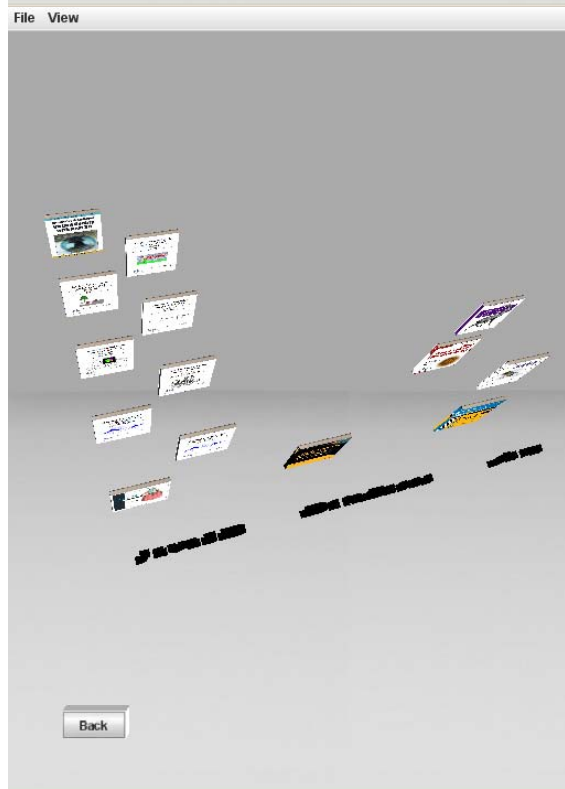
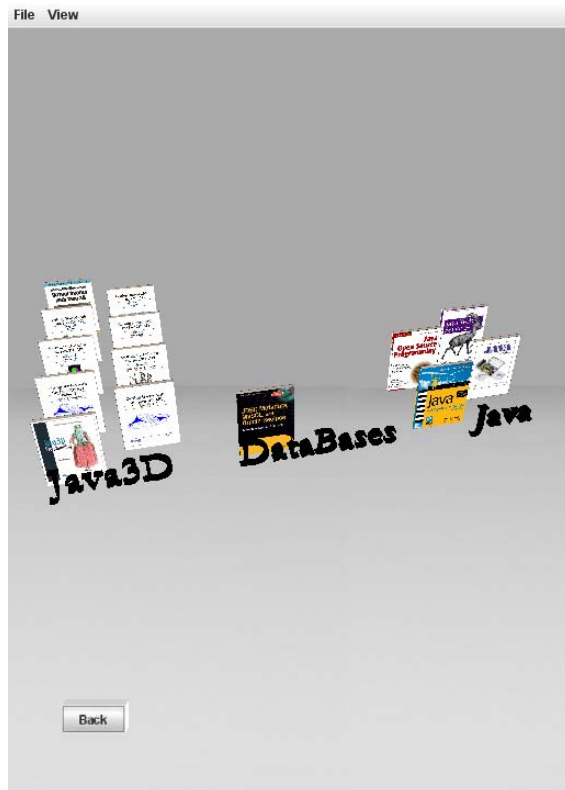
    openPdfItem = new JMenuItem ("Open PDF");
    frontViewItem = new JMenuItem ("Front");
    backViewItem = new JMenuItem ("Back");
    pictureItem = new JMenuItem ("Show Pictures");
    exitItem = new JMenuItem ("Exit");
    j3dMenu.add (frontViewItem);
    j3dMenu.add (backViewItem);

    //SET THE ITEMS
    setJMenuBar (mb);

    //viewMenu.addSeparator ();
    viewMenu.add (openPdfItem);
    //viewMenu.addSeparator ();
    viewMenu.add (pictureItem);
    viewMenu.add (j3dMenu);
    fileMenu.add (exitItem);
```

Συγκεκριμένα τώρα για το front και back menu items χρησιμοποιούμε τις μεθόδους setDefaultCameraPos -για το front (η οποία τοποθετεί την κάμερα στην αρχική της θέση σε αυτή δλδ που είναι όταν ξεκινάει η εφαρμογή)- και simpleUniverse.setCameraPosition(-defaultPos.x, -defaultPos.y, -defaultPos.z) η οποία βάζει την κάμερα ακριβώς αντίθετα .Με τις μεθόδους αυτές μας δίνεται η δυνατότητα να αλλάζουμε τις default θέσεις της κάμερας ανάλογα με τις επιθυμίες μας.

Με διπλό κλικ πάνω σε μία κατηγορία μπαίνουμε στο δεύτερο επίπεδο της εφαρμογής μας , στα βιβλία . Εδώ γίνεται η 3D αναπαράσταση των βιβλίων που έχουμε κατασκευάσει και υποθηκεύσει στα Resources.



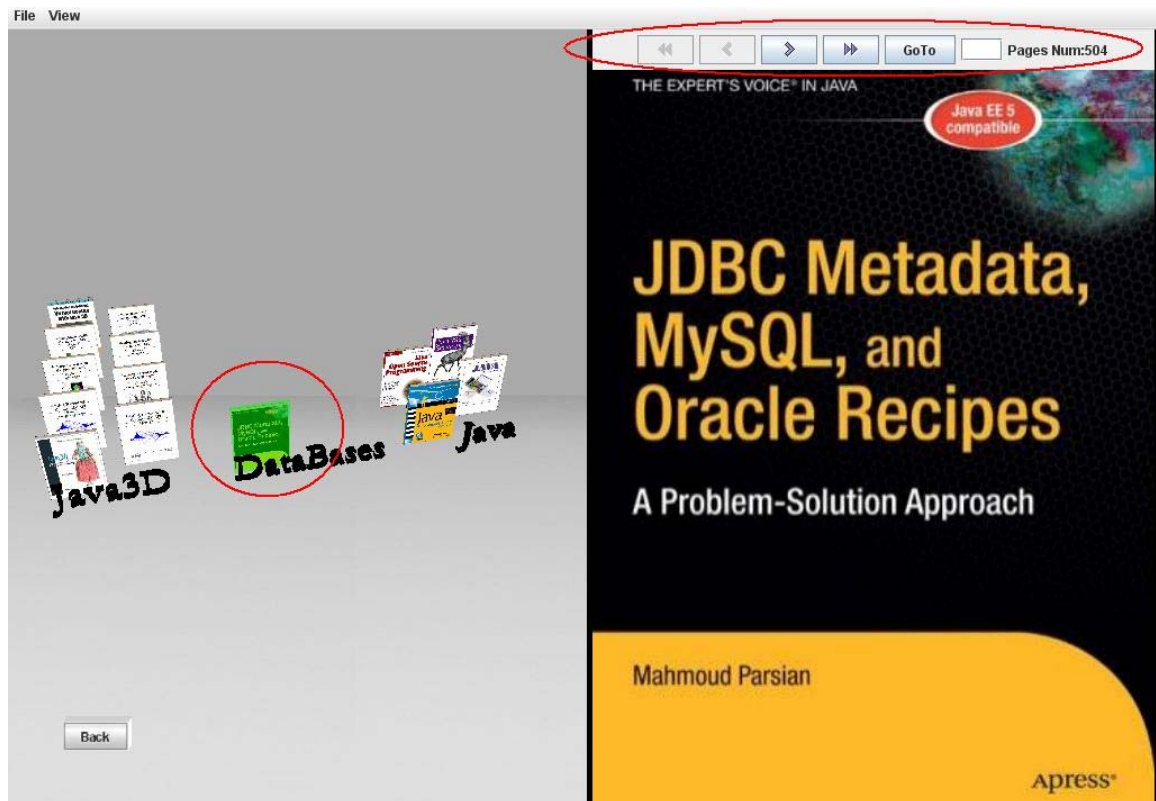
Όπως είναι εμφανές και στις εικόνες είναι και σε αυτό το επίπεδο δυνατή η πλοήγηση με τον ίδιο τρόπο που περιγράψαμε προηγουμένως.

Το κουμπί “Back” στο οποίο είχαμε αναφερθεί παραπάνω μας δίνει τη δυνατότητα να μεταφερθούμε στο προηγούμενο επίπεδο , στις κατηγορίες . Κατασκευάζεται στην κλάση Book3D και γίνεται add στο περιβάλλον μας στην ίδια κλάση .Το παρακάτω block κώδικα προσθέτει το button όπου ακριβώς είναι θεμιτό και του δίνει σχήμα , id , texture, color , μέγεθος κ.α. τα οποία ορίζονται στην κλάση ButtonId .Επίσης στη μέθοδο αυτή το προσθέτουμε στο δικό μας περιβάλλον αφού του έχουμε δώσει όλα αυτά τα χαρακτηριστικά τα οποία θέλουμε.

```
private void addBackButton() {
    backButtonBG = new BranchGroup();
    backButtonBG.setCapability(BranchGroup.ALLOW_DETACH);
    Transform3D transform3D = new Transform3D();
    transform3D.setTranslation(new Vector3d(-0.2, -0.3, -0.7)); //8esh του
    TransformGroup backButtonTG = new TransformGroup(transform3D);
    URL imageURL = Resources.getResource("icons/back.jpg");
    BoxId box = new ButtonId().myBoxButton("butt1", imageURL, 65, 28);
    backButtonTG.addChild(box);
    backButtonBG.addChild(backButtonTG);
    simpleUniverse.getViewTransformGroup().addChild(backButtonBG);
}
```

Ερχόμαστε τώρα στο interaction του χρήστη με τα books. Όταν ο χρήστης είναι στο επίπεδο των βιβλίων έχει δυνατότητες αλληλεπίδρασης με τα βιβλία τις οποίες θα παρουσιάσουμε παρακάτω .Αρχικά θα μιλήσουμε για το τι μπορεί να κάνει ο χρήστης με το ποντίκι και ύστερα θα δούμε τις δυνατότητες που έχει από το μενού.

Πατώντας πάνω σε ένα βιβλίο αυτό γίνεται highlighted όπως φαίνεται στην παρακάτω εικόνα . Αυτόματα εμφανίζεται το αντίστοιχο PDF στο διπλανό panel , το οποίο χρησιμοποιείται για να μπορεί κάποιος να διαβάσει το επιλεγμένο book. Μαζί με το PDF εμφανίζονται τα buttons για το διάβασμα του PDF και ο αριθμός σελίδων του PDF.



Το highlight του βιβλίου γίνεται στη μέθοδο mouse Clicked όπου καλεί τη μέθοδο bookSelection κάθε φορά που το ποντίκι κλικάρει πάνω σε ένα βιβλίο .Με το που κλικάρουμε πάνω σε ένα βιβλίο γίνονται οι ακόλουθες ενέργειες:

```

if (prosema.equals("obj ")) {
    if (e.getClickCount() == 2) {
        hideAllBookDetails();
        moveCameraInFrontOfBookObject();
    }
    crowdPDFImages = 0;
    countPDFImages = 0;
    deselectAll();
    noBook();
    selectBook(intId);
    addImages(intId);
}

```

- Deselect ότι ήταν από πριν επιλεγμένο,
- noBook το οποίο κάνει disabled τα κουμπιά του PDF Viewer
- γίνεται new book selection με βάση το id του book , με τη selectBook που ακολουθεί
- προσθήκη εικόνων του επιλεγμένου book με βάση το id πάλι
- εμφάνιση του επιλεγμένου PDF στο PDF Panel και enable τα κουμπιά για πλοήγηση
- selectBook

```

private void selectBook(int id) {
    if (bookSphere[id].isLive()) {
        simpleUniverse.getLocale().removeBranchGraph(bookSphere[id]);
    } else {
        Transform3D transform3DBook = getLocalToWorldObject();
        TransformGroup tg = new TransformGroup(transform3DBook);
        com.sun.j3d.utils.geometry.Box box = new com.sun.j3d.utils.geometry.Box(0.1f, 1.05f, 0.8f, new Appearance());
        for (int i = 0; i < 6; i++) {
            NodeId node = new NodeId("obj " + id, box.getShape(i));
            tg.addChild(node);
        }

        BranchGroup bg = new BranchGroup();
        bg.setCapability(BranchGroup.ALLOW_DETACH);
        bg.addChild(tg);
        bookSphere[id] = bg;
        simpleUniverse.getLocale().addBranchGraph(bookSphere[id]);
    }
}

```

Γίνεται έλεγχος για το αν το επιλεγμένο book είναι ήδη επιλεγμένο και αν δεν είναι επιλέγεται και φτιάχνεται γύρω του ένα box με transparency (να είναι διαφανές για να φαίνεται το book) και με πράσινο χρώμα με μέγεθος λίγο μεγαλύτερο από το book για να το εμπεριέχει.

Αφού γίνουν όλες αυτές οι ενέργειες πάμε στην εμφάνιση του PDF που γίνεται με τη μέθοδο setPDF αν υπάρχουν data για το PDF αλλιώς πετάει error.

```

getCrowdPDF(intId);
if (crowdPDFImages > 0) {
    // set PDFViewer
    setPDF(intId, 0);
    elegxosPDFButtons();
} else {
    JOptionPane.showMessageDialog(panel, "No Book", "ERROR", JOptionPane.ERROR_MESSAGE);
}

```

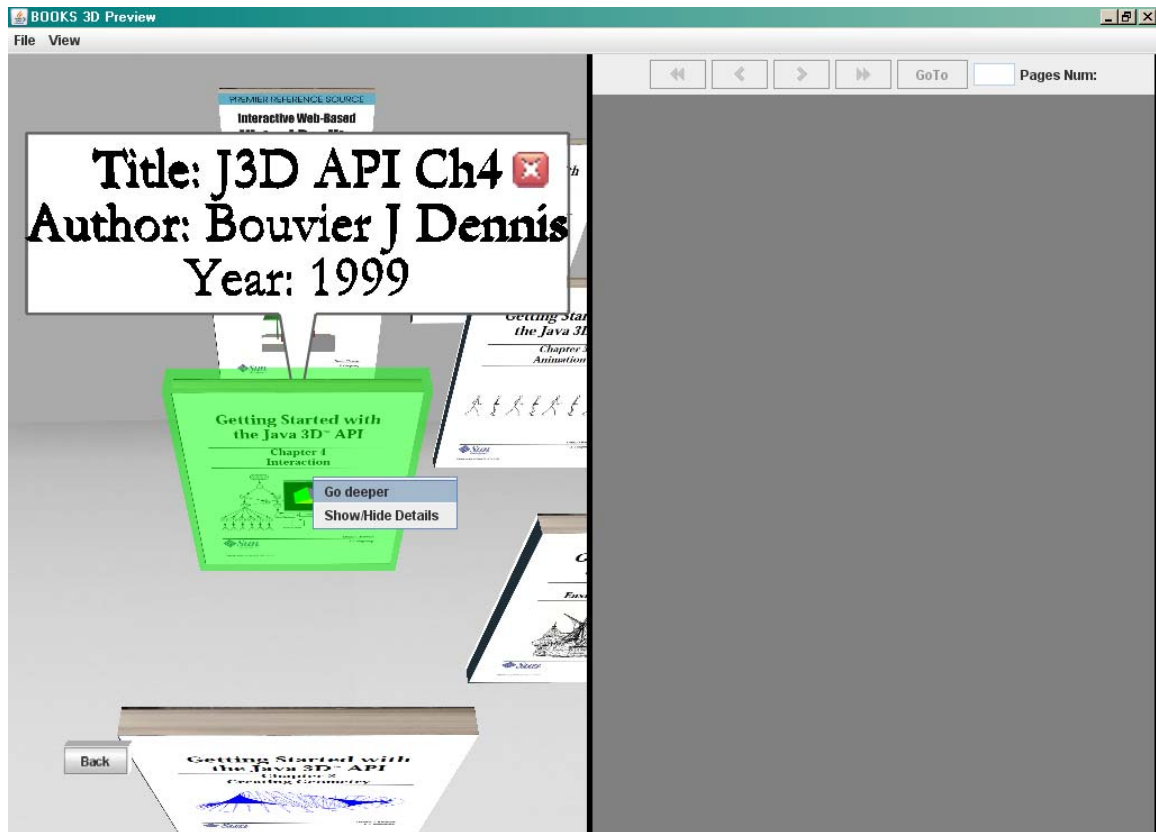
Η getCrowdPDF μετράει τις σελίδες του PDF με βάση πάλι το id μετά γίνεται ο έλεγχος που είπαμε και καλείται η setPDF .Η setPDF με βάση το PDF (αριθμός) και το pageNumber εμφανίζει την αντίστοιχη σελίδα του PDF.

Πέρα από το διάβασμα του PDF έχουμε και το Right – Click menu του επιπέδου αυτού. Εδώ έχουμε τις εξής επιλογές :

“ Go Deeper ”

“ Show/Hide Details “

Και οι 2 αυτές επιλογές εμφανίζονται στην παρακάτω εικόνα .



Το Go Deeper μας κάνει zoom στο book ενώ το show hide details εμφανίζει το annotation με τις πληροφορίες του book τις οποίες παίρνει από τη βάση δεδομένων. Η επιλογή goDeeper μπορεί να αντικαταστεί αν πατήσουμε διπλό κλικ πάνω στο book αν πατήσουμε διπλό κλικ κάπου αλλού εκτός book την ώρα που είναι σε κατάσταση zoom μας επαναφέρει στην αρχική κατάσταση του δευτέρου επιπέδου.

```
// add new menuItem
menuItem = new JMenuItem("Go deeper");
menuItem.addActionListener(new ActionListener() {

    public void actionPerformed(ActionEvent e) {
        hideAllBookDetails();
        moveCameraInFrontOfBookObject();
    }
});
```

Η μέθοδος που γίνεται το zoom στο book είναι η `moveCameraInFrontOfBookObject`

```
private void moveCameraInFrontOfBookObject() {
    Transform3D transform3D = getLocalToWorldObject();
    Vector3d translation = new Vector3d();
    transform3D.get(translation);
    -0.4302278, -3.0020974E-4, -8.491321E-4, 0.9027199
    Transform3D transform3D1 = new Transform3D(new Quat4f(-0.4302278f, -3.00
    simpleUniverse.setTransform3D(transform3D1);
```

Και μας μεταφέρει μπροστά από μια συγκεκριμένη θέση που του ορίζουμε εμείς με το αντικείμενο `Transform3D`.

Η επιλογή show/hide Details τώρα κάνει πάλι deselect all και select το συγκεκριμένο book που κάνουμε δεξί κλικ. Με τη μέθοδο showhideDetails και με βάση το id εμφανίζει τις πληροφορίες του επιλεγμένου book που παίρνει από τη βάση δεδομένων.

```
menuItem = new JMenuItem("Show/Hide Details");
menuItem.addActionListener(new ActionListener() {

    public void actionPerformed(ActionEvent e) {
        deselectAll();
        selectBook(intId);
        // hideAllBookDetails();
        showHideBookDetails(intId);
    }
});
```

Εδώ έχουμε το κομμάτι που τραβάει τα στοιχεία από τη βάση δεδομένων. Παίρνει τις πληροφορίες από τον πίνακα booksInfos -αντικείμενο της κλάσης MyBookInfo το οποίο έχει κατασκευαστεί στον constructor της Book3D- και μας επιστρέφει τις πληροφορίες από τη βάση δεδομένων με τις μεθόδους που φαίνονται στον παρακάτω κώδικα. Αφού πάρει τις πληροφορίες αυτές τις τοποθετεί στο Annotation.

```
" + id, booksInfos[id].getTitle(), booksInfos[id].getAuthor(), booksInfos[id].getBalloon(dimensionX, dimensionY));
anchGroup();
Group.ALLOW_DETACH);
emGroup);
Branch(bookDetails[id]);
```

Το Balloon Annotation με τις πληροφορίες κλείνει είτε πατώντας πάλι δεξί κλικ και hide/show details ή πατώντας το κουμπάκι “x” το οποίο φτιάχεται από την κλάση MyCloseIcon οπότε εκεί καθορίζονται τα πάντα γύρω από το κουμπάκι αυτό. Η κλάση αυτή χρησιμοποιείται από την κλάση BalloonAnnotation η οποία κατασκευάζει το annotation και γίνεται ενεργό.

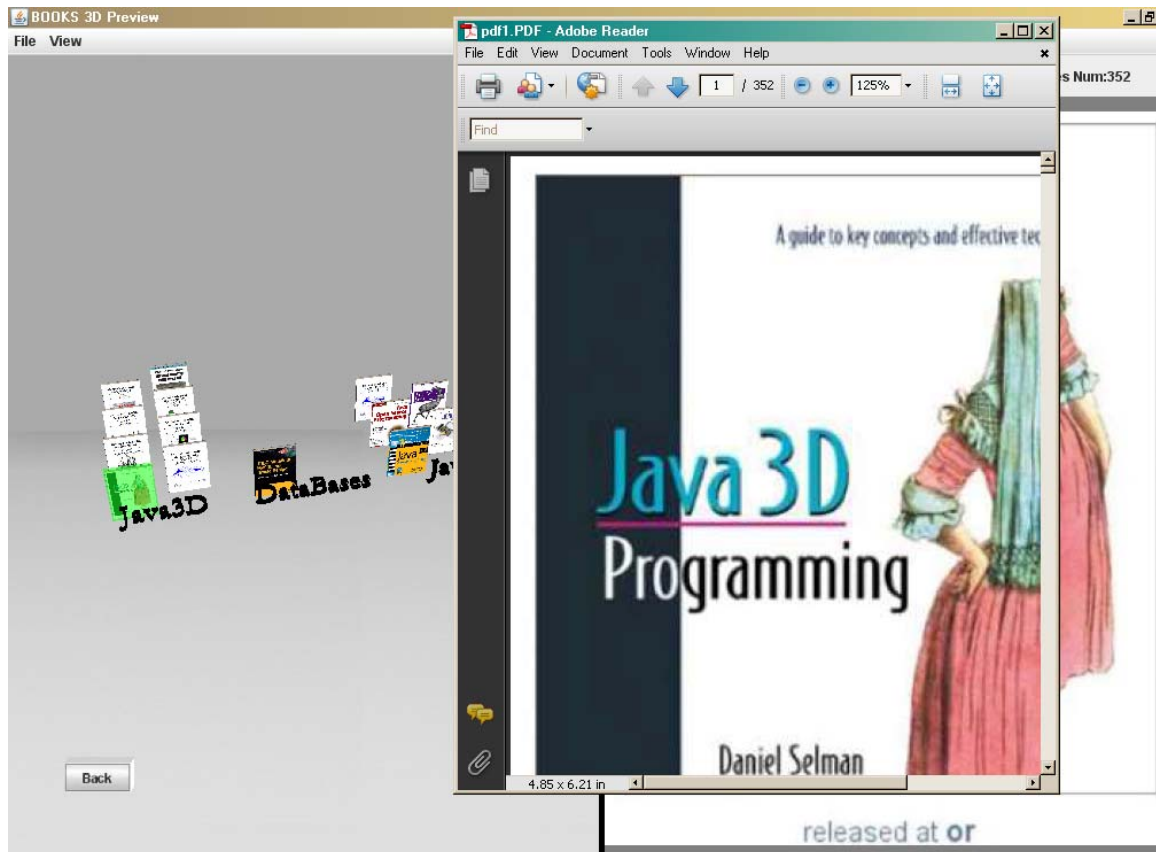
Αφού τελειώσαμε τώρα και με το menu του ποντικιού ας πάμε να δούμε το menu bar. Το menu bar έχει τις εξής επιλογές File και View. Το File αυτό που κάνει μόνο είναι ένα exit το οποίο σε βγάζει από την εφαρμογή. Το View menu έχει 3 επιλογές :

OpenPDF: Έχοντας επιλέξει ένα PDF έχουμε τη δυνατότητα με αυτή την επιλογή να το ανοίξουμε με το πρόγραμμα εμφάνισης PDF Acrobat Reader. Αυτή την επιλογή την έχει ο χρήστης για την περίπτωση που θέλει να επεξεργαστεί το PDF και να το διαβάσει με ένα πρόγραμμα γνωστό σε όλους.

ShowPictures: Πάλι έχοντας επιλέξει ένα PDF έχουμε τη δυνατότητα να ανοίξουμε ένα dialog το οποίο μας δίνει τη δυνατότητα να δούμε κάποιες φωτογραφίες από το επιλεγμένο PDF τις οποίες έχουμε επεξεργαστεί και τοποθετήσει στα Resources της εφαρμογής. Όλες οι εικόνες είναι της μορφής .jpg.

3D: Στο menu 3D έχουμε 3 επιλογές το Front view το back view και το back books view. Το Front μας πάει στο default view έτσι όπως ήταν το επίπεδο των books όταν μπήκαμε. Το back view μας πηγαίνει από κάτω από τα books και το back book view μας δείχνει τα βιβλία από πίσω.

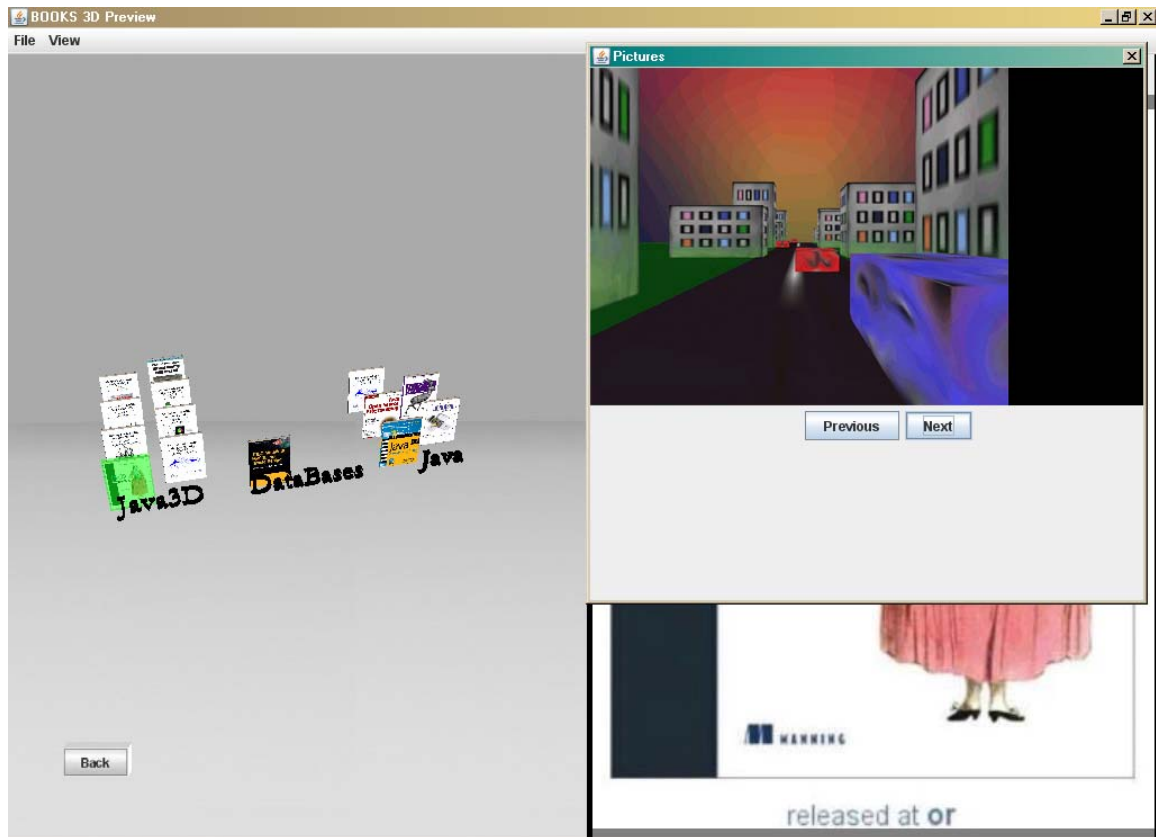
PDF Open



Η επιλογή OpenPDF υλοποιείται στον Action Listener του menu item Open PDF (στην κλάση Book3D), το οποίο καλεί την παρακάτω μέθοδο η οποία ανατρέχει στα Resources και με βάση το εκάστοτε PDF με το Id το ανοίγει κιάλας με την `Runtime.getRuntime().exec("rundll32 url.dll,FileProtocolHandler " + zhtoumeno);`

```
private void openPdfFile(int currentId) {
    try {
        String folderName = "bookPDFs/";
        String name = "pdf" + currentId + ".pdf";
        System.out.println(folderName + name);
        String zhtoumeno = Resources.getResource(folderName + name).toString();
        System.out.println(zhtoumeno);
        Runtime.getRuntime().exec("rundll32 url.dll,FileProtocolHandler " + zhtoumeno);
    } catch (IOException ex) {
        Logger.getLogger(Book3D2.class.getName()).log(Level.SEVERE, null, ex);
    }
}
```

Show Pictures



Με την επιλογή Show Pictures ανοίγει όπως είπαμε ένα JDialog και μας δείχνει τις φωτογραφίες του επιλεγμένου βιβλίου τις οποίες βρίσκει στο resources με βάση το Id του επιλεγμένου book .Και αυτό γίνεται στην κλάση Book3D με τη μέθοδο setImagePanel. Με τη μέθοδο αυτή κατασκευάζουμε το Panel που βλέπουμε στην εικόνα.

Το panel αυτό αποτελείται από 2 μικρότερα panels ένα για τις εικόνες και ένα για τα κουμπιά. Με τα κουμπιά αλλάζουμε φωτογραφίες. Για να εμφανιστούν οι φωτογραφίες πρέπει να έχουμε επιλέξει ένα βιβλίο. Τότε πάλι στη μέθοδο mouseClicked καλείται η addImages (όπως είχαμε πει και σε προηγούμενη αναφορά στην επιλογή βιβλίου) που παίρνει ως όρισμα το id του επιλεγμένου book, ανατρέχει στον φάκελο του book στα resources και εμφανίζονται οι photos όταν επιλέξουμε το Show Pictures.

Η addImages που φαίνεται από κάτω παίρνει ως όρισμα το id του επιλεγμένου book και με βάση αυτό ανατρέχει στον αντίστοιχο φάκελο στα resources και με τη βοήθεια της κλάσης fileparsing διαβάζει τις εικόνες που είναι στο φάκελο. Αν κάποιος θέλει να προσθέσει εικόνες δεν έχει παρά να κάνει copy paste τις εικόνες που θέλει στον αντίστοιχο φάκελο του PDF.

```

private void addImages(int currentId) {
    String folderName = "bookImages/images" + currentId + "/";
    URL folderURL = Resources.getResource(folderPath);
    System.out.println(folderURL.getPath());

    String folderPath = "build/classes/resources/Resources/" + folderName;
    ReadPath f = new ReadPath();
    ArrayList fileList = f.okpath(folderPath);
    int n = fileList.size();
    if (n > 0) {
        images = new URL[n];
        for (int i = 0; i < n; i++) {
            String name = folderName + fileList.get(i);
            images[i] = Resources.getResource(name);
            if (images[i] == null) {
                System.err.println(i + ": error! Picture not found! URL = " + name);
                System.exit(0);
            }
        }
        ImageIcon image = new ImageIcon(images[0]);
        jLabelImage.setIcon(image);
        if (n > 1) {
            jButtonNext.setEnabled(true);
        }
    } else {
        System.err.println("Auto to biblió den exei fotografies.");
    }
}

```

Στην εικόνα από κάτω φαίνεται η ενέργεια που γίνεται όταν επιλέγουμε από το μενού το Show Pictures. Κατασκευάζεται ένα JDialog item στο οποίο προσθέτουμε το panel με τις εικόνες .Και το κάνουμε ορατό.

```

pictureItem.addActionListener(new ActionListener() {

    public void actionPerformed(ActionEvent j) {
        JDialog pictures = new JDialog();
        pictures.setSize(500, 500);
        pictures.setTitle("Pictures");
        pictures.add(jPanelImages);
        pictures.setVisible(true);
    }
}

```

3D

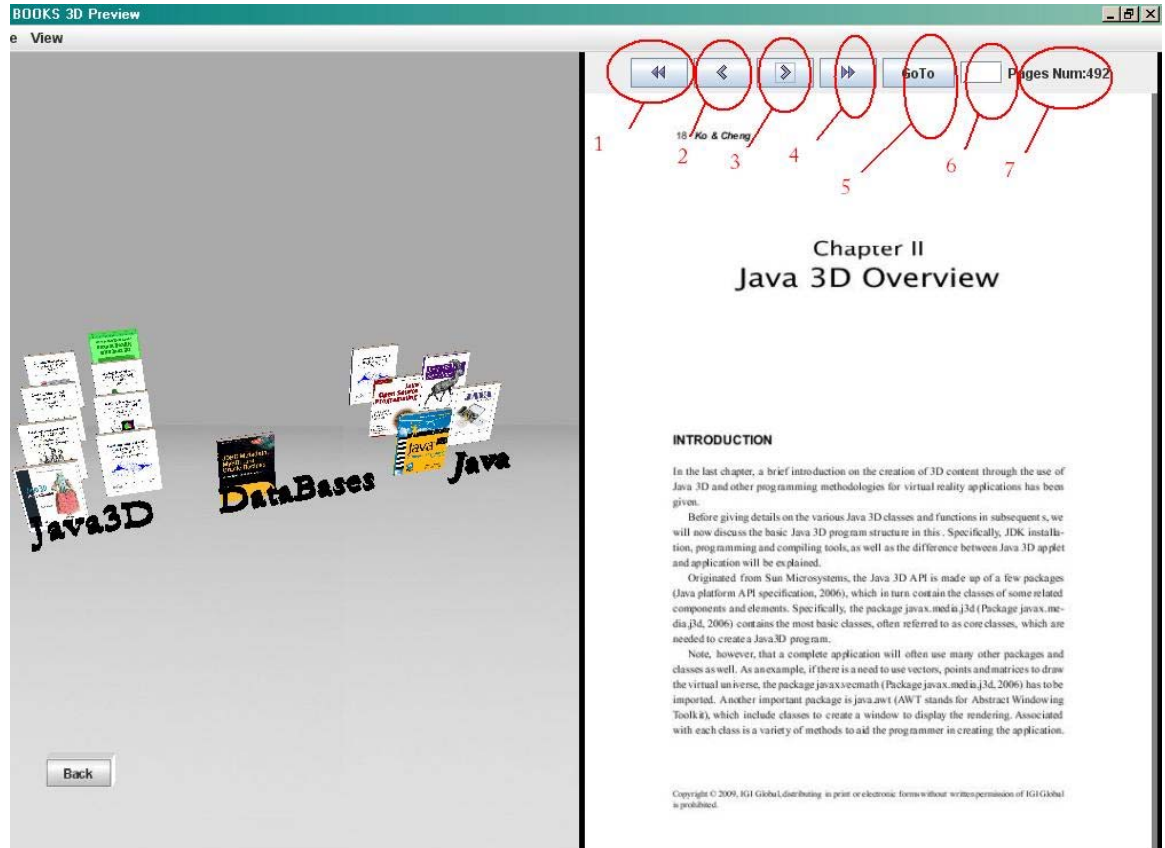
Εδώ κάνουμε το ίδιο με αυτό που είχαμε πει σε προηγούμενη αναφορά μας στο μενού αυτό. Με τα actions των Menu Items αλλάζουμε τη θέση της κάμερας όπως ακριβώς είχαμε πει.

4.3.2.2 PDF Panel

Το PDF panel είναι αυτό που εμφανίζονται οι σελίδες των PDF , όταν επιλέγεται ένα PDF. Οι σελίδες των PDF εμφανίζονται ως εικόνες. Αυτό γίνεται με 2 τρόπους. Ο ένας είναι όταν χρησιμοποιούμε ένα jar αρχείο το PDFRenderer το οποίο διαβάζει τα αρχεία PDF και τα μετατρέπει αυτόματα σε εικόνες. Αυτός ο τρόπος έχει προβλήματα , όταν ανοίγουμε πολλά διαφορετικά PDF δεν κλείνει τα προηγούμενα για ανοίξει τα επόμενα οπότε έχουμε πρόβλημα μνήμης της εφαρμογής.

Η λύση που βρέθηκε είναι μια κλάση που χρησιμοποιήσαμε η οποία ανοίγει το PDF και σώζει κάθε σελίδα του σαν εικόνα .Οπότε βάλαμε τις εικόνες αυτές στα Resources και τις εμφανίζουμε έτσι.

Πάμε να δούμε όμως τις λειτουργίες του.



Με το button αυτό πηγαίνουμε στην πρώτη σελίδα του PDF. Γίνεται enabled false όταν βρισκόμαστε στην πρώτη σελίδα..

Με το 2^ο πηγαίνουμε σε πίσω σελίδες. Γίνεται και αυτό enabled false όταν βρισκόμαστε στην πρώτη σελίδα .

Μας πηγαίνει στην επόμενη σελίδα. Γίνεται και αυτό enabled false όταν βρισκόμαστε στην τελευταία σελίδα .

Μας πηγαίνει στην τελευταία σελίδα. Γίνεται και αυτό enabled false όταν βρισκόμαστε στην τελευταία σελίδα .

Το κουμπί GoTo συνδυάζεται με το JTextField Παίρνει αυτό που εισάγουμε στο πεδίο το κάνει integer και μας πηγαίνει στη σελίδα που λέει ο αριθμός .Σε περίπτωση που δώσουμε λάθος σελίδα μας πετάει error.

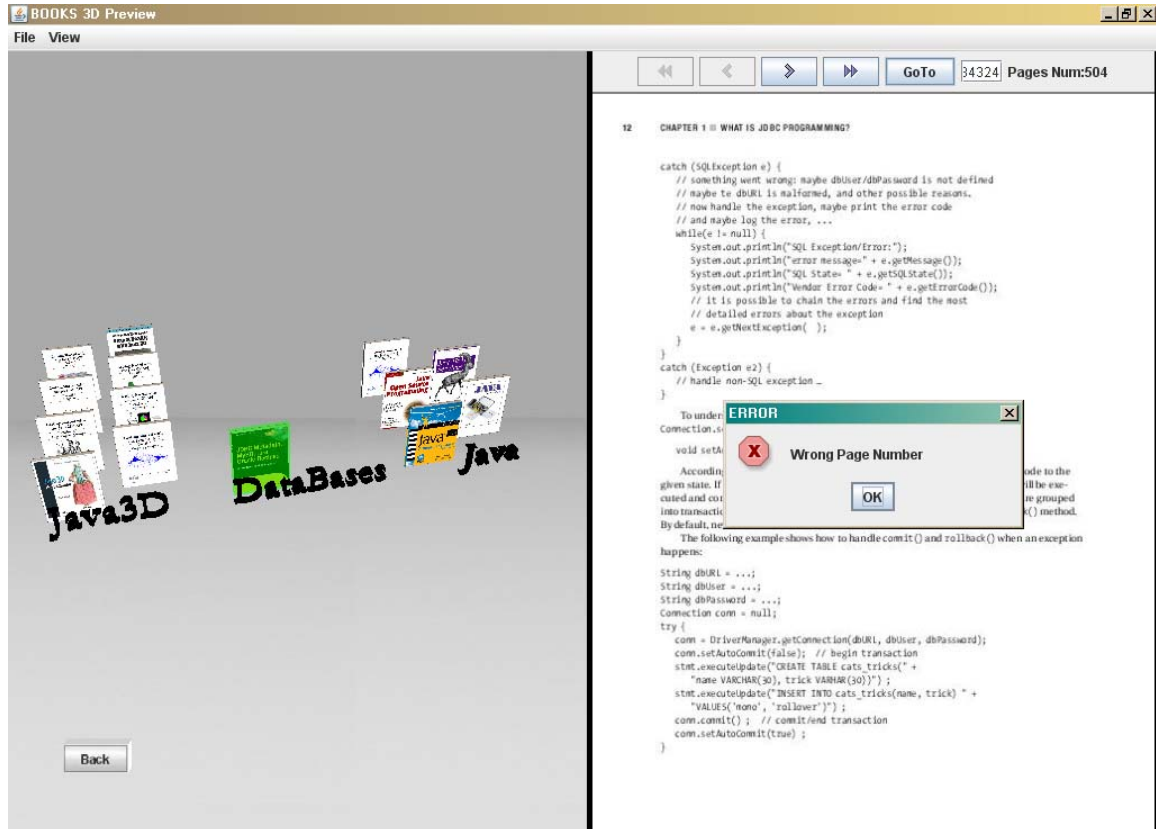
JTextField που δίνουμε τον αριθμό.

Είναι ένα JLabel που όταν επιλέξουμε book μας εμφανίζει τον αριθμό των σελίδων του επιλεγμένου book. Αν δεν έχουμε επιλεγμένο book τότε δεν έχει κανένα αριθμό.

Enabled false



ERROR



Το PDF Viewer κατασκευάζεται στην Book3D κλάση. Αποτελείται από 2 Panels, το ένα για τα buttons και το άλλο για να εμφανίζεται το PDF. Η μέθοδος `mouseClicked` καλεί την `setPDF` η οποία παίρνει ως όρισμα το `id` του book και μας εμφανίζει τις εικόνες του επιλεγμένου book.

```
private void setPDF(int bookId, int bookPage) {
    System.out.println("bookId = " + bookId + " bookPage = " + bookPage);
    String filepath = "imagesPDFs/pdf" + bookId + "/image" + bookPage + ".jpg";
    URL fileURL = Resources.getResource(filepath);
    System.out.println(fileURL.toString());
    ImageIcon image = new ImageIcon(fileURL);
    currentImagePDF = bookPage;
    jLabelPDF.setIcon(image);
    System.out.println("asfdas" + crowdPDFImages);
    int i = crowdPDFImages;
    if (i == 0) {
        pagesLabel.setText("Pages Num=none");
    } else {
        pagesLabel.setText("Pages Num:" + i);
    }
}
```

Περίπτωση που χρησιμοποιούμε το PDF Renderer.

```

private void setPDFViewer(int id) {

    try {
        pagepanel = new PagePanel();
        pagepanel.setSize(300,300);

        File file = new File("src/resources/Resources/bookPDFs/pdf" + id + ".pdf");
        RandomAccessFile raf = new RandomAccessFile(file, "r");
        FileChannel channel = raf.getChannel();
        ByteBuffer buf = channel.map(FileChannel.MapMode.READ_ONLY, 0, channel.size());
        pdffile = new PDFFile(buf);
    } catch (IOException ex) {
        ex.printStackTrace();
    }

    page = pdffile.getPage(count);
    pagepanel = new PagePanel();
    panel.add(pagepanel, BorderLayout.CENTER);

    System.out.println("num pages" + pdffile.getNumPages());
    pagepanel.setSize(500, 700);
    pagepanel.useZoomTool(true);

    // pdffile.
    pagepanel.showPage(page);
    //pdffile.stop(pagenumber);

    pageslabel = new JLabel("num pages:" + pdffile.getNumPages());
}

```

Ο έλεγχος για τα κουμπιά –αν είναι ενεργά η όχι – γίνεται στον Action Listener κάθε κουμπιού ώστε σε διαφορετικές καταστάσεις να αλλάζει το αν είναι ενεργά η όχι.

Το Panel κατασκευάζεται από τη μέθοδο setPDFPanel και στις 2 περιπτώσεις.

4.3.2.3 My Data Base

Όπως αναφέραμε και δείξαμε πιο πριν , τα στοιχεία των books που παρουσιάζονται στην εφαρμογή είναι αποθηκευμένα σε μια MySQL βάση δεδομένων. Στη συνέχεια θα δούμε πιο αναλυτικά τα στοιχεία της βάσης δεδομένων , τη διασύνδεση της βάσης με την εφαρμογή και πως παίρνουμε και χρησιμοποιούμε τα στοιχεία αυτά.

Μερος1: Με τη βοήθεια του MySQL Browser μπορούμε να δούμε τα στοιχεία της βάσης μας και να επέμβουμε σε αυτά προσθέτοντας ή αφαιρώντας εγγραφές από τα πεδία της βάσης η ακόμα και προσθαφαίρεση πεδίων της βάσης. Παρακάτω φαίνεται η βάση μας και τα στοιχεία της τα οποία και θα αναλύσουμε.

id	title	author	typeBook	date	type
1	Beginners Guide	Heribert Schildt	Java	2005	Computing
2	Java3D Program..	Daniel Selman	Java3D	2003	Computing
3	JDBC Metadata	Mahmoud Parsian	Databases	2006	Computing
4	2D Graphics	O'Reilly	Java	1999	Computing
5	Java	Joe Walnes	Java	2004	Computing
6	Web Services	O'Reilly	Java	2002	Computing
7	J3D API Ch1	Bouvier J Dennis	Java	1999	Computing
8	J3D API Ch2	Bouvier J Dennis	Java3D	1999	Computing
9	J3D API Ch3	Bouvier J Dennis	Java3D	1999	Computing
10	J3D API Ch4	Bouvier J Dennis	Java3D	1999	Computing
11	J3D API Ch5	Bouvier J Dennis	Java3D	1999	Computing
12	J3D API Ch6	Bouvier J Dennis	Java3D	1999	Computing
13	J3D API Ch7	Bouvier J Dennis	Java3D	1999	Computing
14	Java3D V R	Chi Chung Ko	Java3D	2008	Computing
15	Systems Program..	Darrel Ince	Katanemhmena	2009	Education
16	Death Jaga	A.C.Ellis	Novel	2004	Literature
17	Poembook	J.R.Afamasaga	Poetry	1888	Literature

Έχουμε τη βάση library με πίνακα books ο οποίος περιέχει τα παρακάτω πεδία id: Είναι το πρωτεύων κλειδί της βάσης μας. Με βάση αυτό γίνεται ο διαχωρισμός των βιβλίων και είναι ένα πεδίο τύπου INT.

title: Είναι το πεδίο που αποθηκεύεται ο τίτλος του βιβλίου. Είναι πεδίο τύπου VARCHAR , ουσιαστικά αποθηκεύονται Strings.

author: Είναι το πεδίο που αποθηκεύεται ο συγγραφέας του βιβλίου. Είναι πεδίο τύπου VARCHAR , και εδώ αποθηκεύονται ως Strings οι συγγραφείς

typeBook: Είναι το πεδίο που αποθηκεύεται η υποκατηγορία του βιβλίου. Είναι πεδίο τύπου VARCHAR. Με βάση αυτό το πεδίο γίνεται ο χωρισμός των κατηγοριών στο επίπεδο 2 της εφαρμογής που είπαμε πριν.

date: Είναι το πεδίο που αποθηκεύεται η ημερομηνία που εκδόθηκε το βιβλίο. Είναι πεδίο τύπου VARCHAR και αυτό.

type: Είναι το πεδίο που αποθηκεύεται η κατηγορία του βιβλίου. Είναι πεδίο τύπου VARCHAR και με βάση αυτό γίνεται η κατηγοριοποίηση στο επίπεδο 1 της εφαρμογής.

Μερος2: Η εφαρμογή μας περιέχει ένα πακέτο τεσσάρων κλάσεων τις οποίες χρησιμοποιούμε για τη διασύνδεση με τη βάση μας και για να πάρουμε τα στοιχεία της βάσης. Πάμε να δούμε ακριβώς πως γίνονται όλα αυτά.

Αρχικά έχουμε μια κλάση **MyDatabase** η οποία κατασκευάζει τη βάση δεδομένων η οποία όμως χρησιμοποιείται από την κλάση **MyConnection** και τη **Book3D**. Δε την χρησιμοποιούμε για την κατασκευή ή επεξεργασία της βάσης γιατί το MySQL query Browser είναι πιο ευέλικτο στην επεξεργασία της βάσης και πιο κατανοητό από απλούς χρήστες.

MyConnection τη χρησιμοποιούμε για να επιστρέψουμε τα στοιχεία για τη σύνδεση με τη βάση δεδομένων. Αποθηκεύουμε τα απαραίτητα στοιχεία για τη σύνδεση με τη βάση και τα επιστρέφουμε με μία μέθοδο. Επίσης έχουμε μεθόδους για να επιστρέφονται –σε περίπτωση που χρειαστεί- τα στοιχεία της βάσης ξεχωριστά.

MyBookInfo κλάση η οποία παίρνει ως ορίσματα στον constructor της int id, String title, String author, String type, String typeBook, String date και τα επιστρέφει. Έτσι εμείς παίρνουμε μετά τα δεδομένα που θέλουμε από τη βάση.

MyDBParser κλάση με την οποία γίνεται η σύνδεση στη βάση δεδομένων και ύστερα διαβάζουμε τη βάση και παίρνουμε τα στοιχεία της. Αυτά αποθηκεύονται σε ένα πίνακα τύπου **MyBookInfo**. Κάθε εγγραφή του πίνακα είναι και ένα στοιχείο άρα και ένα βιβλίο της βάσης δεδομένων. Αφού γίνει αυτό μετά έχουμε μία μέθοδο η οποία επιστρέφει αυτόν τον πίνακα γεμάτο. Στην εφαρμογή μας μετά κατασκευάζουμε ένα αντικείμενο της **MyBookInfo** και ένα **dbParser** και ύστερα `booksInfos=dbParser.getBookInfos()`; Μας επιστρέφεται ένας πίνακας αντικειμένων τύπου **MyBookInfo** με τα data της βάσης μας.

Τα στοιχεία μετά όπως είπαμε και σε προηγούμενες αναφορές τα χρησιμοποιούμε όπου αυτά είναι χρήσιμα .

4.3.2.4 ScreenShots

Image1

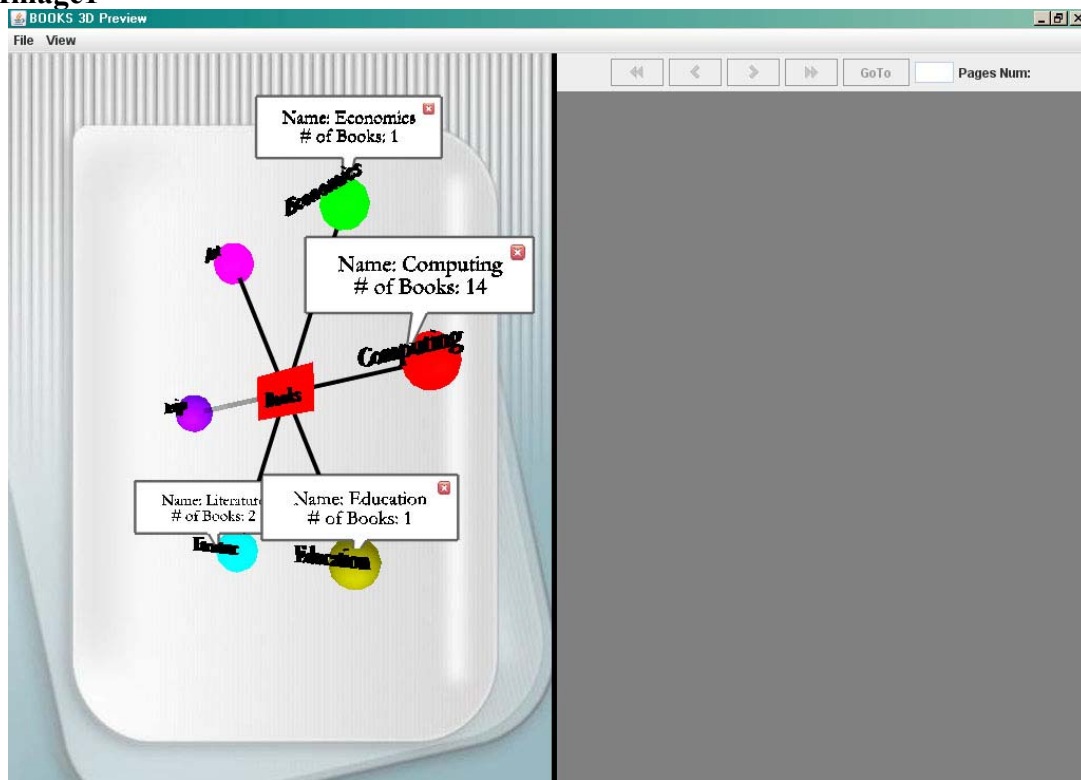


Image2

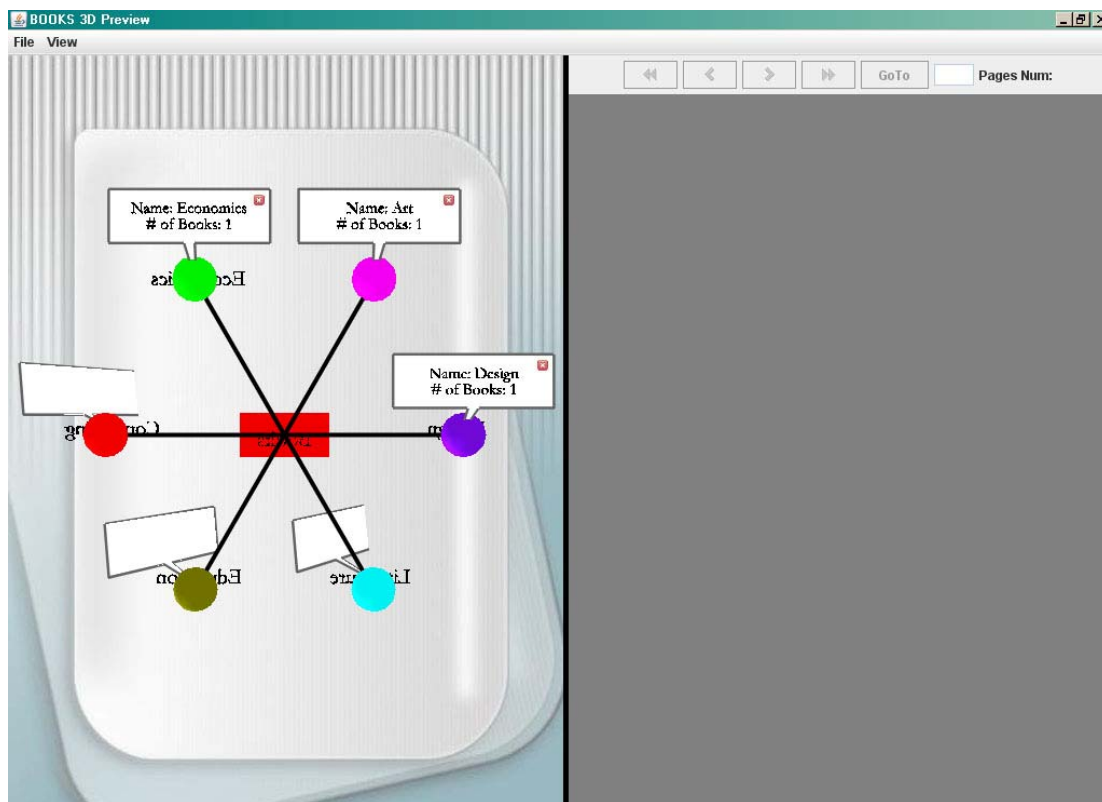


Image3

BOOKS 3D Preview

File View

GoTo 34 Pages Num:482

12 Chapter 2

- A customer should be able to visit the store's Web site and browse through the pets. A customer should be able either to select a category to view a pet or to perform a text-based search for a pet.
- If a customer finds a desirable pet, the customer should be able to add it to a shopping cart. When the customer is satisfied, he or she should be able to go to a checkout screen and place an order for all the pets in the store.

To keep the example application simple, this book does not go into the details of what happens once the order is placed.

Examining the Architecture and Technologies

The core platform used for development is JDK 1.4.1 with the *Servlet 2.3* and *SP 1.2* APIs from the *J2EE 1.3* standard. It's worth noting that PetSear doesn't use all J2EE features such as JMS or EJBs. In all cases, we aim to use the simplest tool for the job.

Figure 2.1 shows all these frameworks, along with their places and relationships in the big puzzle of the architecture.

NOTE The technologies used throughout this book are only several of many possible recommendations. If you prefer to use other technologies, such as Jakarta Struts instead of WebWork, go ahead and do so. The beauty of Open Source Java is that it's very easy to mix and match. All applications should use the best combination to fit the stated needs, and we don't agree with a concrete prescribed technology set.

Looking at the Architecture

The core functionality of the application is made up of a collection of services. Low-level services provide system functionality, such as indexing a document or persisting on object. Higher-level services provide business logic, such as maintaining the pet inventory or processing the shopping cart on checkout.

Services are layered on top of each other to build the application. Each service follows the facade design pattern and encapsulates the complexities of logic and external APIs behind a very simple interface.

Layering services together in this way is known as a *service-oriented architecture* and, at the expense of more classes, simplifies the code greatly throughout the application by organizing it into modules that specialize in only doing one thing and doing it well. Code is cleaner and contains less duplication. This increases the maintainability of the application in the long run.

Back

Image4

BOOKS 3D Preview

File View

GoTo Pages Num:492

Premier Reference Source

Interactive Web-Based Virtual Reality

Pictures

sphere_edit

Previous Next

Back

Κεφάλαιο 5^ο

Βιβλιογραφία

Για την υλοποίηση της εφαρμογής πέρα από κάποια e-books –μερικά από τα οποία παρουσιάζονται στην εφαρμογή- χρησιμοποιήσαμε και ιστοσελίδες με open source οι οποίες παρουσιάζονται παρακάτω μαζί με τα e-books

5.1 E-Books

- Java3D Sun API tutorial all chapters 1-7
- IGI Global Interactive Web Based Virtual Reality with Java.3D
- Java3D Programming Daniel Selman
- O'Reilly Java 2D Graphics

5.2 Sites

- <http://en.wikipedia.org>
- <http://java.sun.com/>
- <http://java.sun.com/docs/books/tutorial/uiswing/>
- <http://dev.mysql.com/>
- <http://www.mysql.com>
- <http://www.java2s.com/>
- <http://download.java.net/media/java3d/>
- <http://www.netbeans.org/features>
- <http://java.sun.com/javase/technologies/desktop/java3d/>
- <https://java3d.dev.java.net/>
- <http://www.java3d.org/>