



ΤΕΧΝΟΛΟΓΙΚΟ
ΕΚΠΑΙΔΕΥΤΙΚΟ
ΙΔΡΥΜΑ ΚΡΗΤΗΣ

ΤΕΙ Ηρακλείου Τμήμα Εφαρμοσμένης Πληροφορικής και Πολυμέσων

Πτυχιακή Εργασία :

*«Διάδραση σε εικονικό περιβάλλον μέσω
γαντιού εικονικής πραγματικότητας»*

Εισηγητής: Αθανάσιος Μαλάμος

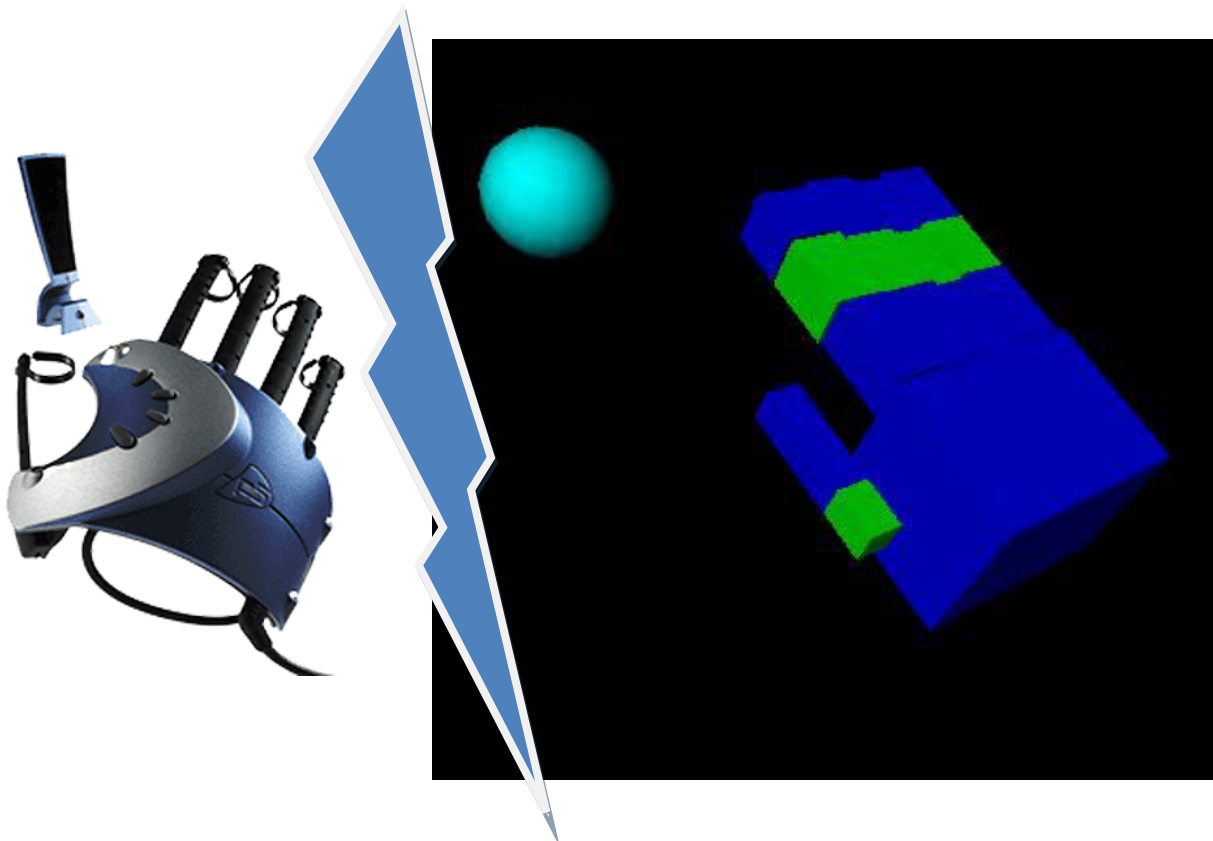
Στοιχεία σπουδαστών:

Αλεξάνδρου Αλέξανδρος Α.Μ. 1036
Βουλάμαντης Νικόλαος Α.Μ. 1053

Ημερομηνία: 19/12/2008



1. Εισαγωγικά
 - 1.1. Προηγούμενη δουλειά
 - 1.2. Τα εργαλεία που χρησιμοποιήθηκαν
 - 1.2.1. P5Glove
 - 1.2.2. NetBeans
 - 1.2.3. Xj3D
 - 1.2.4. Vitality Studio
 - 1.3. Προετοιμασία συστήματος
2. Τρισδιάστατο περιβάλλον προσομοίωσης
 - 2.1. Προσομοίωση του γαντιού σε τρισδιάστατο περιβάλλον
 - 2.2. Διάδραση σε τρισδιάστατο περιβάλλον
3. Επεξήγηση του κώδικα της πτυχιακής
 - 3.1. Ο κώδικας της Java για το χειρισμό του εικονικού γαντιού και την αλληλεπίδραση του με το τρισδιάστατο περιβάλλον
 - 3.2. Ο κώδικας του X3D για την δημιουργία του τρισδιάστατου περιβάλλοντος και του εικονικού μοντέλου του γαντιού
4. Παράρτημα
 - 4.1. Μελλοντικές προσθήκες και δυνατότητες χρήσης του εικονικού μοντέλου
 - 4.2. Επεξήγηση εννοιών
 - 4.3. Πηγές





ΤΕΙ Ηρακλείου Τμήμα Ε.Π.Π.

1. Εισαγωγικά

Η παρούσα πτυχιακή εργασία είναι μέρος μίας συλλογικής δουλειάς που έχει γίνει από πολλά άτομα προσθέτοντας ο κάθε ένας το δικό του κομμάτι στην ολοκλήρωση ενός συλλογικού project για την ανάπτυξη και αλληλεπίδραση με ένα περιβάλλον εικονικής πραγματικότητας.

Η πτυχιακή εργασία έχει ως σκοπό τις να δημιουργήσει ένα «εργαλείο» εικονικής πραγματικότητας και συγκεκριμένα ένα εικονικό μοντέλο γαντιού υπεύθυνο για την διάδραση και την αλληλεπίδραση του χρήστη με τρισδιάστατη διεπαφή (3D περιβάλλον) μέσω ενός εικονικού μοντέλου γαντιού. Δηλαδή την δημιουργία του αντίστοιχου κέρσορα του παραδοσιακού ποντικιού για υπολογιστεί αλλά για τις τρεις διαστάσεις.

Για το σκοπό αυτό χρησιμοποιήσαμε γάντι εικονικής πραγματικότητας (P5Glove) που θα μας επιτρέψει να έχουμε 3 άξονες κίνησης (x,y,z) καθώς και κίνηση σε κάθε εικονικό δάκτυλο ξεχωριστά.



Το εικονικό μοντέλο του γαντιού καθώς και το περιβάλλον δημιουργήθηκαν σε X3D και προβάλλονται στο χρήστη μέσω του browser Xj3D. Η σύνδεση του γαντιού (P5Glove) με το εικονικό μοντέλο του γίνεται μέσω κλάσης Java που διαχειρίζεται το αρχείο X3D.

Παρακάτω θα αναλύσουμε τα εργαλεία που χρησιμοποιήσαμε αλλά και τους λόγους που τα επιλέξαμε.



ΤΕΙ Ηρακλείου Τμήμα Ε.Π.Π.

1.1 Προηγούμενη δουλειά

Το P5Glove είναι ένα πολύ φτηνό γάντι εικονικής πραγματικότητας και για το λόγο αυτό πολύ διαδεδομένο σε όσους θέλουν να πειραματιστούν σε περιβάλλον εικονικής πραγματικότητας και όχι μόνο. Πολύ είναι αυτή που κατά καιρούς έχουν χρησιμοποιήσει το συγκεκριμένο περιφερικό άλλοτε για χειρισμό κάποιου μηχανικού μέρους και άλλοτε για χειρισμό κάποιου μοντέλου σε εικονικό περιβάλλον.

Μία πολύ χρήσιμη πηγή για μας αποτέλεσε το βιβλίο «Pro Java 6 3D Game Development» καθώς είχε πολλά παραδείγματα και έτοιμο κώδικα για το συγκεκριμένο γάντι (P5Glove).

Παλαιότερες πτυχιακές εργασίες σχετικά με το X3D του τμήματος σχετικά με τον σχεδιασμό τρισδιάστατου περιβάλλοντος και την αλληλεπίδραση μέσω παραδοσιακών συσκευών εισόδου όπως το ποντίκι και το πληκτρολόγιο.

Επίσης η πολύ καλή και δωρεάν προσπάθεια όλων των ανθρώπων που αναπτύσσουν των Xj3D browser και το API της εφαρμογής.

Τέλος πρέπει να σταθούμε στην προσπάθεια μερικών να βελτιώσουν τους drivers του P5Glove μετά το κλείσιμο της εταιρίας που το παρήγαγε (essential reality).



ΤΕΙ Ηρακλείου Τμήμα Ε.Π.Π.

1.2 Τα εργαλεία που χρησιμοποιήθηκαν

1.2.1 P5Glove: Το γάντι που χρησιμοποιούμε σαν συσκευή εισόδου για τη εργασία μας είναι το P5Glove της εταιρίας essential reality. Το γάντι διαθέτει ένα σύνολο αισθητήρων για ανίχνευση της κίνησης του στο χώρο (τρεις διαστάσεις) καθώς και τη κίνηση του κάθε δακτύλου του χεριού που το φοράει. Η εταιρία που το κατασκεύαζε έχει κλείσει οπότε μπορεί να το προμηθευτεί κάποιος μόνο από κάποιο μαγαζί στο internet (www.ebay.com) και μόνο από την Αμερική (δεν έχει γίνει ποτέ εισαγωγή στην Ευρώπη). Το συγκεκριμένο γάντι είναι από τα πιο φτηνά που μπορεί να βρει κάποιος στην αγορά. Αυτό είναι που το κάνει προσιτό σε κάποιον που απλώς θέλει να πειραματιστεί με μια διαφορετική συσκευή εισόδου και ιδανική λύση για εξοντωτικούς πειραματισμούς. Εμείς το προτιμήσαμε όχι μόνο για το χαμηλό κόστος του αλλά και γιατί δεν χιαζόμαστε κάποιο καλύτερο για την συγκεκριμένη εργασία.

1.2.2 Xj3D: Το Xj3D είναι ένα πρόγραμμα browser που ανέπτυξε το W3D Consortium και βασίζεται στην δημιουργία ενός σετ εργαλείων για την επεξεργασία VRML97 και X3D και είναι γραμμένο αποκλειστικά σε Java. Το Xj3D ήταν στην αρχή εξαρτημένο από την Java3D αλλά τώρα έχει επεκταθεί πέρα από αυτήν και μπορεί να υποστηρίξει και άλλες μεθόδους. Τον προτιμήσαμε έναντι των άλλων browsers γιατί προσφέρει την μεγαλύτερη συμβατότητα με την Java, είναι δωρεάν και μπορείς να βρεις το API και άλλες χρήσιμες πληροφορίες και παραδείγματα (tutorials) δωρεάν στο internet.

1.2.3 NetBeans: Το NetBeans είναι ένα ολοκληρωμένο πακέτο για την ανάπτυξη λογισμικού (χρησιμοποιώντας perl, php, python, ruby, java, c και c++), αλλά και ένα IDE χρησιμοποιώντας τη πλατφόρμα του NetBeans. Το NetBeans είναι ένα από τα πιο ολοκληρωμένα εργαλεία στον χώρο της Java. Υποστηρίζεται από την netbeans.org και είναι ελεύθερο προς “download” από τον διαδικτυακό τόπο: <http://www.netbeans.org>

1.2.4 Vivaty Studio: Το Vivaty είναι ένα δωρεάν πρόγραμμα δημιουργίας τρισδιάστατων μοντέλων όπως το 3DStudioMax και το AutoCAD το οποίο χρησιμοποιεί VRML και X3D για την αναπαράσταση των τρισδιάστατων γραφικών (<http://developer.vivaty.com>).



ΤΕΙ Ηρακλείου Τμήμα Ε.Π.Π.

1.3 Προετοιμασία συστήματος

Εκτός από την εγκατάσταση όλων των απαιτούμενων προγραμμάτων, για να μπορέσουν να λειτουργήσουν οι εφαρμογές που χρησιμοποιούμε όπως και το γάντι θα πρέπει να γίνουν μερικές ρυθμίσεις στον υπολογιστή πριν προσπαθήσουμε να τρέξουμε το πρόγραμμα. Επίσης επειδή ανοίγουμε τον browser μέσα από Java θα πρέπει να φορτώσουμε όλες της βιβλιοθήκες του στο project μας. Διαβάστε αναλυτικά τις οδηγίες που ακολουθούν και να γνωρίζεται ότι επειδή η τεχνολογία του X3D είναι σχετικά σε πρώιμο στάδιο αλλά και επειδή ο browser που χρησιμοποιούμε παρόλο που είναι γραμμένος σε Java (και πλεονεκτεί έναντι των υπολοίπων σε συμβατότητα όπως προαναφέραμε) έχει μερικές ασυμβατότητες και προβλήματα που είτε σχετίζονται με την έκδοση του προγράμματος, είτε με την έκδοση της Java, είτε με τις ρυθμίσεις του X3D. Το λειτουργικό που χρησιμοποιήσαμε είναι τα Windows XP GR service pack 2.

Τα προγράμματα που απαιτούνται για να τρέξει το project είναι τα εξής:

Η πλατφόρμα της Java : Χρειαζόμαστε την Standard Edition της τελευταίας έκδοσης του JDK της Java. (<http://java.sun.com>)

To NetBeans : Αναπτύσσεται από την netbeans.org στο site της οποίας μπορούμε να βρούμε την τελευταία έκδοση του εν λόγω προγράμματος. (<http://www.netbeans.org>)

To Vivaty Studio : (<http://developer.vivaty.com/>)

To περιβάλλον και ο browser Xj3D : (<http://www.xj3d.org>)

Η βιβλιοθήκη για την λειτουργία του γαντιού P5 : Εδώ πρέπει να σημειώσουμε ότι δεν αρκεί η απλή εγκατάσταση των drivers της συσκευής που την συνοδεύουν. Χρειαζόμαστε ποιό τελευταία έκδοση του P5DLL.dll και του CP5DLL.java. Τα παραπάνω αρχεία βρίσκονται στον διαδικτυακό τόπο του Yahoo! Group™ που δημιουργήθηκε ειδικά για το P5Glove και την ανάπτυξή του.

Ρύθμιση προγραμμάτων:

Οι ρυθμίσεις που γίνονται είναι συνήθως απλές αλλά αναγκαίες για την σωστή λειτουργία των προγραμμάτων. Τα προγράμματα που χρησιμοποιούνται έρχονται με autoseup και δεν χρειάζονται ειδικές ρυθμίσεις (εκτός από το μέρος εγκατάστασης) και μετά από μια μικρή εξοικείωση με τον τρόπο λειτουργίας τους δεν παρουσιάζουν ιδιαίτερες δυσκολίες. Αυτό που χρειάζεται προσοχής είναι το λογισμικό για το P5 glove. Το εν λόγω γάντι έρχεται με ένα λογισμικό το οποίο δεν είναι επαρκές για τις σύγχρονες ανάγκες των προγραμμάτων αφού υποστηρίζει μόνο θύρες COM και όχι USB που χρειαζόμαστε. Η εταιρία που το εμπορευόταν (essential reality) έχει κλείσει με αποτέλεσμα να μην υπάρχει επίσημη υποστήριξη προς το προϊόν. Τον ρόλο του service έχουν πάρει οι ίδιοι οι χρήστες του



ΤΕΙ Ηρακλείου Τμήμα Ε.Π.Π.

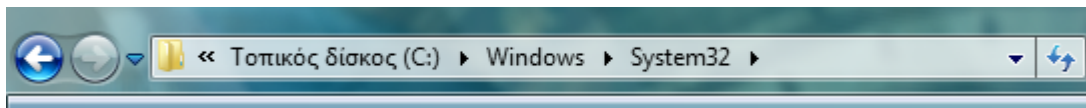
γαντιού ανά τον κόσμο. Σε ένα από τα Yahoo! GroupsTM όπως προαναφέρθηκε γίνεται η υποστήριξη λειτουργίας του P5 και από εκεί μπορεί κανείς να βρει τις τελευταίες εκδόσεις από διάφορα προγράμματα του P5 καθώς και την απαραίτητη νεότερη βιβλιοθήκη P5DLL.dll. Με αυτή την βιβλιοθήκη προστίθεται υποστήριξη για θύρες USB καθώς και αυξάνει την σωστή απόκριση του γαντιού. Επίσης στον ίδιο δικτυακό τόπο μπορούμε να βρούμε μια νεότερη έκδοση του CP5DLL.java.

Από το CP5DLL πρέπει να δημιουργήσουμε το CP5DLL.jar και θα το κάνουμε με τα παρακάτω βήματα:

- 1) Για να ανοίξουμε την «γραμμή εντολών» πατάμε: Έναρξη -> Εκτέλεση και στην φόρμα που θα εμφανιστεί γράφουμε «cmd» και πατάμε «OK»
- 2) Γράφουμε την εντολή: `javac -d CP5DLL.java` δημιουργώντας έτσι το `com.essentialreality` package

```
Microsoft Windows [Έκδοση 6.0.6001]
Πνευματικά δικαιώματα (c) 2006 Microsoft Corporation. Με επιφύλαξη κάθε νόμιμου
δικαιώματος.
C:\Users\Typhoun>cd .
C:\Users>cd .
C:\>javac -d CP5DLL.java
```

- 3) Στην συνέχεια με την εντολή: `jar cvf CP5DLL.jar com` δημιουργώντας το jar αρχείο CP5DLL.jar .
- 4) Αντιγράφουμε την βιβλιοθήκη P5DLL.dll στους φάκελους “system” και “system32” του συστήματος (C:\Windows\).

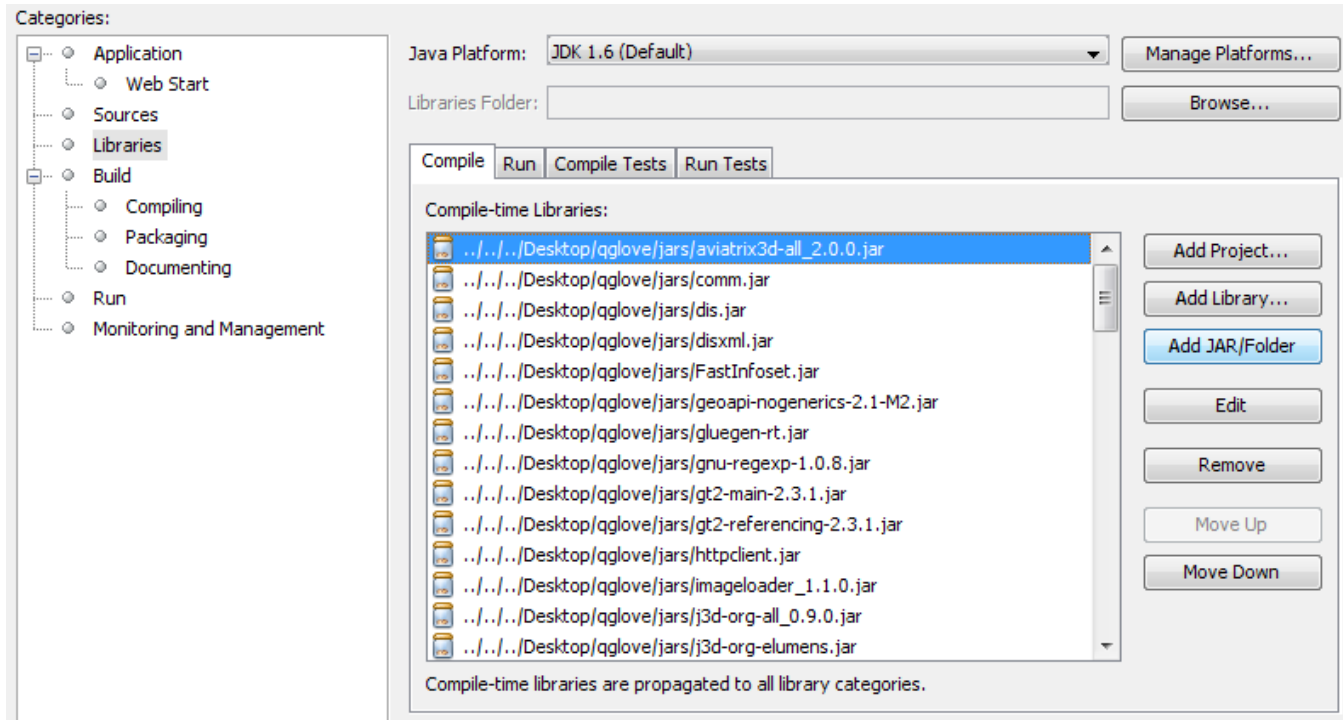


- 5) Ξεκινάμε το NetBeans και ανοίγουμε το υπάρχον project με τον πηγαίο κώδικα που έχουμε. Στα libraries κάνουμε προσθήκη τα απαιτούμενα jar αρχεία (jar αρχεία του x3d). Αυτό γίνεται κάνοντας δεξί κλικ στο όνομα του project και από εκεί επιλέγουμε το Properties και στην συνέχεια libraries. Πηγαίνουμε στην επιλογή “add jar” και επιλέγουμε τα jar αρχεία που βρίσκουμε στην βιβλιοθήκη του X3D καθώς και το CP5DLL.jar και πατάμε OK. Τώρα το project μας είναι έτοιμο να “τρέξει” το μόνο που έχουμε να κάνουμε είναι να επιλέξουμε “Run file” για τον κώδικά μας και θα δούμε να ανοίγει ο xj3d browser και να εμφανίζεται το μοντέλο του γαντιού μας.



ΤΕΙ Ηρακλείου Τμήμα Ε.Π.Π.

Όταν ανοίξουμε το NetBeans αφού επιλέξουμε «Open Project» για να εισάγουμε το project μας, κάνουμε διπλό κλικ πάνω στο όνομα που θα εμφανιστεί και μετά δεξή κλικ στην επιλογή «Libraries». Αν γίνει σωστά η διαδικασία θα εμφανιστεί η παρακάτω εικόνα.



Κάνοντας κλικ στο κουμπί «Add JAR/Folder», εντοπίζουμε τις απαραίτητες βιβλιοθήκες και πατάμε «OK».

Προβλήματα και ασυμβατότητες:

Η καλύτερη περίπτωση για να εγκαταστήσετε τα απαραίτητα προγράμματα χωρίς ιδιαίτερα προβλήματα είναι να εγκαταστήσετε τα Windows (έως και την έκδοση των Vista) μόνο με τους απαραίτητους drivers των συσκευών του συστήματος. Απενεργοποιήστε τα περιφερικά όπως ασύρματες κάρτες δικτύωσης και αποσυνδέστε όλες της συσκευές usb, αν θέλετε οπωσδήποτε κάποια, αφού έχετε καταφέρει να τρέξετε το project αρχίστε να συνδέεται διαδοχικά τις συσκευές μέχρι να προκύψει κάποιο πρόβλημα. Μην φορτώσετε τίποτε άλλο στο path της java και μην κάνετε εγκατάσταση σε αυτό το λειτουργικό κάποιο άλλο πρόγραμμα (γιατί μπορεί να δημιουργήσει πρόβλημα είτε στους drivers του γαντιού είτε στο NetBeans).

Καλό θα ήταν προτού προσπαθήσετε να τρέξετε το project να πάτε στο πίνακα ελέγχου και να κάνετε διπλό κλικ στο εικονίδιο P5Glove, μετά στο κάντε δεξί κλικ στο εικονίδιο του P5Glove στο system-tray των Windows και να επιλέξετε «Open P5 Panel». Στο παράθυρο που θα εμφανιστεί επιλέγουμε την καρτέλα «Mouse» και πατάμε στην επιλογή «Mouse OFF» στο κουμπί εφαρμογή.



ΤΕΙ Ηρακλείου Τμήμα Ε.Π.Π.

2. Τρισδιάστατο περιβάλλον προσομοίωσης

Την δεκαετία του 80' έχει αναπτυχθεί το τρισδιάστατο περιβάλλον στον τομέα τις πληροφορικής και των γραφικών. Το τρισδιάστατο περιβάλλον αυτό ήρθε σαν αντικαταστάτης του δισδιάστατου περιβάλλοντος το οποίο χρησιμοποιούνταν μέχρι τότε. Αυτό το βήμα ακολούθησε την τάση της βιομηχανίας παιχνιδιών να παράγει όλο και καλύτερα γραφικά. Στην ουσία τα τρισδιάστατα γραφικά ,ή απλούστερα 3D, είναι μια επεξεργασία 2D εικόνων μέσω της οποίας επιτυγχάνουμε να προσδώσουμε βάθος στο “αντικείμενο” που πλέον έχουμε. Η επεξεργασία αυτή συνίσταται από την εξομοίωση κάποιου φωτισμού (photorealistic rendering) ή στην εφαρμογή κάποιων προτύπων σχεδιασμού (non-photorealistic rendering). Έτσι αφού επεξεργαζόμαστε 2D εικόνες για την αναπαράσταση τρισδιάστατων γραφικών χρησιμοποιούμε και πολλές τεχνικές επεξεργασίας δισδιάστατων γραφικών.

Μία πρώτη προσπάθεια παρουσίασης τρισδιάστατων γραφικών υπήρξε μια “ηλεκτρονική” παρουσιάστρια ειδήσεων που παρουσίαζε το πρόγραμμά της στο internet για μια βραχεία περίοδο. Το μεγάλο ενδιαφέρον κυρίως για την εικονική πραγματικότητα ήταν το έναυσμα για την ανάπτυξη των τρισδιάστατων γραφικών και των τεχνολογιών που σχετίζονται με αυτά. Έχουν αναπτυχθεί τεχνολογίες που εκμεταλλεύονται τα τρισδιάστατα γραφικά, όπως το γάντι εικονικής πραγματικότητας που αναπτύσσουμε και εμείς. Η εικονική πραγματικότητα δεν είναι τίποτε άλλο από την προσπάθεια να αναπαραστήσουμε το πραγματικό περιβάλλον σε ένα εικονικό 3D περιβάλλον με όσο το δυνατόν μεγαλύτερη ακρίβεια.

Έχουν δημιουργηθεί συσκευές εικονικής πραγματικότητας όπως γάντια, γυαλιά και διάφορες άλλες συσκευές ανίχνευσης κίνησης, ενώ γίνεται προσπάθεια ώστε να εξομοιωθούν όλες οι ανθρώπινες αισθήσεις (π.χ. όσφρηση). Όσο περνάει ο καιρός, οι συσκευές αυτές βελτιώνονται, με συνέπεια την αύξηση της ακρίβειας των μετρήσεων αλλά και την αύξηση των δυνατοτήτων τους (σε επίπεδο εξομοίωσης).

Όσο η τεχνολογία βελτιώνεται, βελτιώνεται και η πιστότητα του 3D κόσμου. Νέες τεχνικές προσδίδουν νέα υφή και ομοιομορφία στον τρισδιάστατο κόσμο που πλέον φτάνει αρκετά κοντά στο πραγματικό. Η επόμενη πρόκληση είναι να καταφέρουμε την δημιουργία “πιστού” αντίγραφου του ανθρώπου, πράγμα το οποίο μέχρι τώρα δεν το έχουμε πετύχει παρά μόνο κατά προσέγγιση.





2.1 Προσομοίωση του γαντιού σε τρισδιάστατο περιβάλλον

Για να κατανοήσουμε πλήρως την προσομοίωση του γαντιού, θα πρέπει να εξετάσουμε το ανθρώπινο χέρι που φοράει το γάντι P5Glove. Ένα ανθρώπινο χέρι διαθέτει 27 μικρά οστά, συμπεριλαμβανομένων 8 οστών σε σχήμα βότσαλου στον καρπό, όλα συνδεδεμένα με ένα δίκτυο συνδέσμων και δεκάδες μικροσκοπικούς μύες που κινούν τις αρθρώσεις. Οι επιστήμονες έχουν παρατηρήσει ότι ο εγκέφαλος συνδέεται με τα χέρια και με περισσότερες νευρικές συνδέσεις από οποιοδήποτε άλλο μέρος του σώματος και έτσι οι κινήσεις που κάνουμε με αυτά δίνουν ισχυρές ενδείξεις για τη συναισθηματική μας κατάσταση. Άρα είναι εύκολο να αντιληφτούμε την περιπλοκότητα που μπορεί να έχουν οι κινήσεις του χεριού μας καθώς και το πόσο μεγάλη ακρίβεια μπορούν να πετύχουν με την σωστή εξάσκηση.

Σε τι όμως μας χρησιμεύουν οι πληροφορίες σχετικά με το ανθρώπινο χέρι, όταν έχουμε να κάνουμε με την εξομοίωση της κινήσεις ενός γαντιού; Στο ότι για να εξομοιώσει ένα γάντι τις κινήσεις ενός χεριού χρειάζεται έναν αισθητήρα για κάθε πιθανή κίνηση (άρθρωση του χεριού). Έτσι μπορούμε να καταλήξουμε στο συμπέρασμα ότι για να καταφέρουμε να εξομοιώσουμε σωστά σχεδόν όλες τις κινήσεις ενός χεριού χρειαζόμαστε πάρα πολύ καλή μηχανή εισόδου. Στην περίπτωσή μας το P5Glove διαθέτει αισθητήρια μόνο για τη αναγνώριση του χεριού στο χώρο και για το λύγισμα των δακτύλων (μόνο κάθετη κίνηση). Αυτός ήταν και ένας από τους λόγους που μας οδήγησαν να εξομοιώσουμε μοντέλο γαντιού και όχι χεριού. Επίσης αν θέλαμε να έχουμε ένα αρκετά ρεαλιστικό μοντέλο χεριού θα έπρεπε να σχεδιάσουμε ένα πολύπλοκο μοντέλο, που θα ήταν δύσκολο να εξομοιώσουμε ακόμα και μια απλή κίνηση.

Προβλήματα επεξεργαστικής ισχύς γενικά δεν αντιμετωπίσαμε όσο άφορα το εικονικό μοντέλο του χεριού, κάποια «σπασίματα» στην κίνηση οφείλονται σε αδυναμία του browser.

Anatomy of the Hand





ΤΕΙ Ηρακλείου Τμήμα Ε.Π.Π.

2.2 Διάδραση σε τρισδιάστατο περιβάλλον

Το 3D περιβάλλον που χρησιμοποιούμε είναι σχεδιασμένο σε X3D (VRML πρότυπο) και ουσιαστικά είναι πειραματικό. Δηλαδή φτιάξαμε ένα κόσμο μόνο για να πειραματιστούμε με το γάντι, γι' αυτό και είναι λιτός έχοντας μόνο το μοντέλο του γαντιού και μερικές σφαίρες (πειραματικά αντικείμενα αλληλεπίδρασης). Τον κόσμο μας δεν το διέπουν κανόνες φυσικής (π.χ. βαρύτητα). Ας δούμε όμως κάποια γενικά στοιχεία για τους εικονικούς αυτούς κόσμους και τη εννοούμε όταν αναφερόμαστε σε διάδραση μέσα σε περιβάλλον 3D.

Όταν αναφερόμαστε στην διάδραση σε 3D περιβάλλον εικονικής πραγματικότητας, εννοούμε ότι οι χρήστες θα μπορούν να διεκπεραιώνουν κάποιες εργασίες σε τρισδιάστατο περιβάλλον το οποίο εξομοιώνει τον πραγματικό κόσμο και τις εργασίες που ο χρήστης θα έκανε εκεί. Οι διαδράσεις αυτές μπορούν να αποτυπωθούν σαν ομάδες από κατεύθυνση κίνησης, εικονικό κόσμο και ελέγχους συστήματος. Η διάδραση σε έναν τρισδιάστατο κόσμο μπορεί να γίνει με διάφορα μέσα, ποντίκι, γάντι εικονικής πραγματικότητας (όπως στην περίπτωση μας), αισθητήρες σώματος κ.α. Το ποντίκι σαν συσκευή κατάδειξης και επιλογής μπορεί να επιλέξει ένα εικονικό σώμα και να το μεταφέρει μέσα στον χώρο. Αυτή είναι μία μορφή απλής διάδρασης με τον τρισδιάστατο κόσμο και προσφέρει περιορισμένες λειτουργίες. Το επόμενο βήμα είναι το γάντι εικονικής πραγματικότητας το οποίο μέσω πολλαπλών αισθητήρων, οι οποίοι βρίσκονται πάνω στο κυρίως σώμα του γαντιού που αποτελούνται συνήθως από αισθητήρες υπερύθρων (για την αποτύπωση των κινήσεων του χεριού) και αισθητήρες αντιστάσεως (για την αποτύπωση των κινήσεων των δακτύλων), μεταφέρουν τις κινήσεις του χεριού σε ένα τρισδιάστατο και ψηφιακό μοντέλο γαντιού (όπως στην περίπτωση μας) και έτσι δίνουν την δυνατότητα στο τρισδιάστατο αυτό μοντέλο να κάνει κινήσεις στον χώρο και να “πιάσει”, όπως και το πραγματικό χέρι στην πραγματικότητα, κάποιο αντικείμενο και να το μεταφέρει στον χώρο.

Αυτό που αποτελεί την επιτομή της τεχνολογίας στον τομέα της μεταφοράς της ανθρώπινης κίνησης στο τρισδιάστατο μοντέλο είναι η “στολή αισθητήρων” που ουσιαστικά είναι αυτό που η φράση υποδηλώνει, μια στολή που της έχουν προσθέσει διάφορους αισθητήρες. Η στολή αυτή πιάνει όλες τις κινήσεις του σώματος κάνοντας το τρισδιάστατο μοντέλο του σώματος να κινείται όπως το κανονικό σώμα. Αυτή η μέθοδος χρησιμοποιείται κατά κόρων στην βιομηχανία παραγωγής αθλητικών παιχνιδιών, όπως μπάσκετ, ποδόσφαιρο, τένις, γκολφ και λοιπά.

Το πρόβλημα στον τομέα αυτόν έγκειται στο ότι οι αισθητήρες δεν είναι 100% έγκυροι και ακριβείς. Η έλλειψη ακρίβειας λειτουργεί σαν ανασταλτικός παράγοντας στην σωστή αποτύπωση της ανθρώπινης κίνησης στο τρισδιάστατο μοντέλο αφού μπορεί η παρουσία πολλαπλών αισθητήρων να καταστήσει τους αισθητήρες ανίκανους να πάρουν σωστή μέτρηση ή να μεταφράσουν λάθος μία κίνηση του σώματος και να την αποδώσουν σαν μια διαφορετική κίνηση στο τρισδιάστατο μοντέλο. Την εξάλειψη αυτού του προβλήματος ασάφειας (η οποία όσο τέλεια να είναι θα έχει πάντα ένα ποσοστό λάθους) έρχεται να καταφέρει το calibration των αισθητήρων. Πρέπει να ρυθμιστεί ίση ευαισθησία σε όλους τους αισθητήρες (αυξομειώνοντας την ευαισθησία των αισθητήρων), ώστε να παρουσιαστεί



ΤΕΙ Ηρακλείου Τμήμα Ε.Π.Π.

με τον ένα ή τον άλλο τρόπο ένα ομοιόμορφο αποτέλεσμα που να πλησιάζει ικανοποιητικά στο πραγματικό. Στο project μας συγκεκριμένα ο χώρος κίνησης δεν έχει πρακτικά όρια αλλά και αν εξετάσουμε προσεκτικά το γάντι θα δούμε ότι ο χώρος μέσα στο οποίο κινείται έτσι ώστε να μην βγαίνει έξω από την εμβέλεια του γαντιού, είναι συγκεκριμένος. Άρα για να το καταλάβουμε καλύτερα ας πάρουμε το παράδειγμα του ποντικιού και να εξετάσουμε τη συμβαίνει σε διδιάστατα περιβάλλοντα κίνησης. Τα θεωρητικά όρια κίνησης του ποντικιού είναι τα όρια του mouse-pad, αλλά και τα όρια τις οθόνης μας. Έτσι όταν χρειαζόμαστε σε μία εφαρμογή (π.χ. σχεδίασης) ακρίβεια, ελαττώνουμε την ταχύτητα του και όταν θέλουμε να κινηθούμε από την μία άκρη της οθόνης στην άλλη αστραπιαία (π.χ. σε κάποιο παιχνίδι) τότε αυξάνουμε την ταχύτητα. Κάτι αντίστοιχο λοιπόν συμβαίνει και με το μοντέλο του γαντιού. Η ταχύτητα του είναι ανάλογη του χώρου κίνησης της συσκευής εισόδου (P5Glove), αλλά και του μεγέθους του κόσμου μέσα στον οποίο θέλουμε να κινήσουμε το εικονικό γάντι.



Κάτι ανάλογο συμβαίνει και με την αλληλεπίδραση με κάποιο άλλο μοντέλο που υπάρχει στο κόσμο. Το γάντι μας για να αλληλεπιδράσει με ένα άλλο μοντέλο στον εικονικό κόσμο πρέπει να γνωρίζουμε την θέση που έχει το αντικείμενο που θέλουμε να αλληλεπιδράσουμε και την θέση που έχει το μοντέλο του γαντιού κάθε στιγμή. Γνωρίζοντας τις δύο θέσεις, εμείς θέτουμε ένα όριο τιμής που καθορίζει πόσο κοντά είναι τα δύο μοντέλα. Αν το όριο που θέσαμε ξεπεραστεί τότε τα δύο μοντέλα θα αλληλεπιδράσουν σύμφωνα με της εντολές που τους έχουμε δώσει (π.χ. στην περίπτωση μας το γάντι θα πιάσει την σφαίρα αν και μόνο αν είναι κοντά της αλλά και αν είναι λυγισμένα όλα τα δάκτυλά μας). Τα πράγματα θα ήταν πολύ πιο εύκολα αν υπήρχε κάποιο αισθητήριο, όπως αυτό που τοποθετείτε σ' ένα αντικείμενο και αλληλεπιδρά με τον κέρσορα του ποντικιού όταν αυτός περάσει από πάνω



ΤΕΙ Ηρακλείου Τμήμα Ε.Π.Π.

του, αλλά δυστυχώς μέχρι στιγμής δεν υπάρχει κάποιος έτοιμος που να μπορούμε να χρησιμοποιήσουμε σε περιβάλλον X3D και να επαληθεύεται όταν συγκρούονται δύο αντικείμενα (υποστηρίζονται μόνο δύο: του ποντικιού που προαναφέραμε και του viewpoint).

Μπορούμε να βάλουμε κανόνες φυσικής αλλά και υπερφυσικής στο κόσμο μας, εξαρτάται από το τη θέλουμε να κάνουμε. Ένα ενδιαφέρον project που αναπτύσσεται από μία ομάδα αμερικάνων προγραμματιστών με το όνομα «oblong», ουσιαστικά είναι ένα τρισδιάστατο περιβάλλον προσομοίωσης των πληροφοριών και των δυνατοτήτων ενός λειτουργικού συστήματος με υποστήριξη οθονών, αλλά το πιο ενδιαφέρον είναι ότι η συσκευή αλληλεπίδρασης είναι ένα (ή παραπάνω) γάντι εικονικής πραγματικότητας. Το μειονέκτημα του συγκεκριμένου project είναι ότι το μοντέλο αλληλεπίδρασης είναι ένας σχεδόν δισδιάστατος δίσκος και γι' αυτό το λόγο χρησιμοποιούν αναγνώριση προτύπων ώστε να αναγνωρίζουν την κίνηση και να την μεταφράζουν σε εντολή συστήματος. Ποιο απλά το μειονέκτημα τους είναι ότι δεν έχουν σχεδιάσει ένα μοντέλο γαντιού-χειριού!

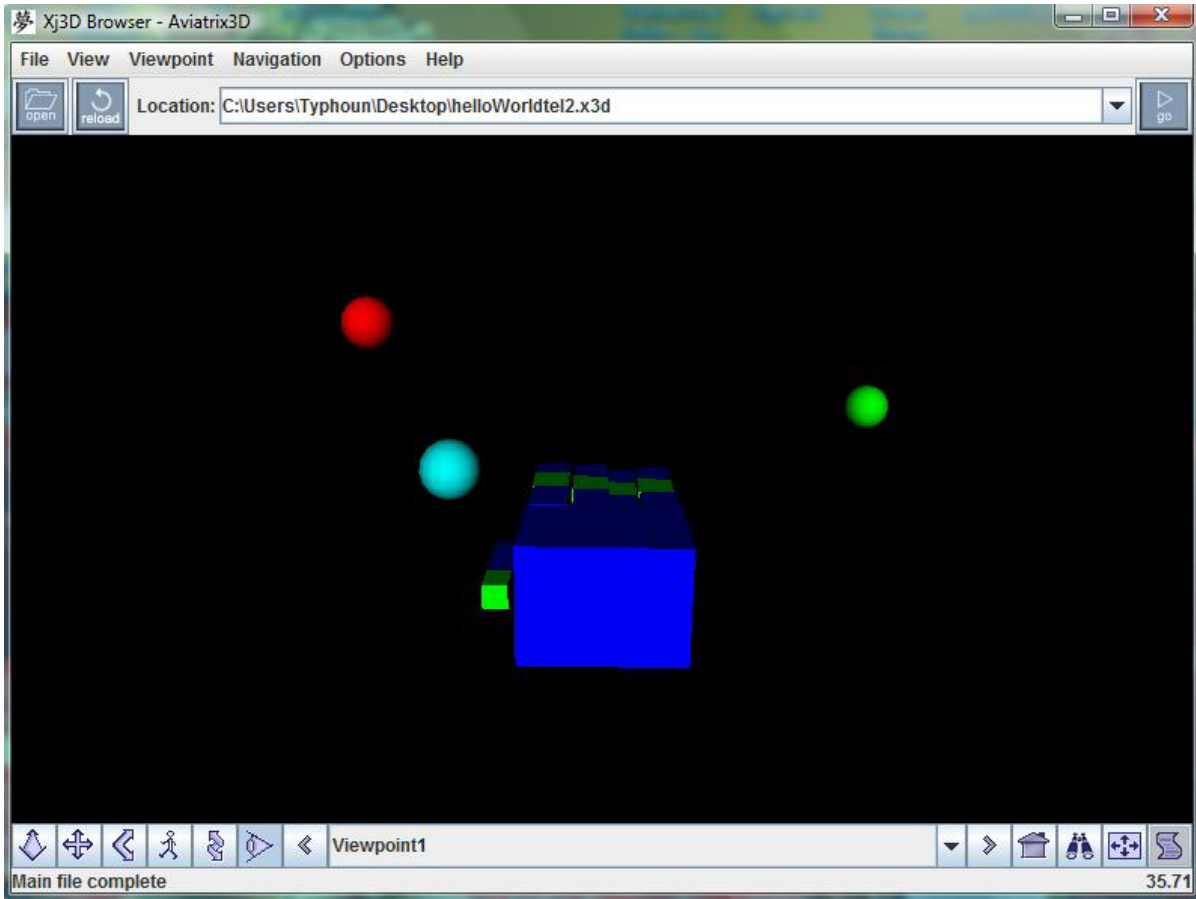
Στο μέλλον οι συσκευές εισόδου θα δίνουν την δυνατότητα στο να σχεδιαστούν ακόμα πιο περίπλοκα περιβάλλοντα με περισσότερες δυνατότητες αλληλεπίδρασης. Χαρακτηριστικά κάποιοι υποστηρίζουν ότι το αισθητήριο που θα αλλάξει τα δεδομένα στον τομέα της διάδρασης στο τρισδιάστατο περιβάλλον είναι οι αισθητήρες που θα ανιχνεύουν σκέψεις από τον ίδιο τον εγκέφαλο του ανθρώπου μέσω ηλεκτροδίων που είναι προσκολλημένα στην περιοχή του μετώπου και των κροτάφων του.



ΤΕΙ Ηρακλείου Τμήμα Ε.Π.Π.

3 Επεξήγηση του κώδικα της πτυχιακής

Παρακάτω θα αναφερθούμε εξηγήσουμε το κώδικα που χρησιμοποιήσαμε, είτε αυτόν που πήραμε έτοιμο, είτε αυτόν που γράψαμε εμείς.





ΤΕΙ Ηρακλείου Τμήμα Ε.Π.Π.

3.1 Ο κώδικας της Java για το χειρισμό του εικονικού γαντιού και την αλληλεπίδραση του με το τρισδιάστατο περιβάλλον

Η κλάση που χρησιμοποιήσαμε για να καλέσουμε και να διαχειριστούμε το X3D είναι η «Ganti.java», παρακάτω επεξηγούμε τμηματικά το κώδικα.

```
package com.essentialreality;

import java.io.IOException;
import java.util.Enumeration;
import java.util.Vector;
import javax.comm.PortInUseException;
import javax.comm.UnsupportedCommOperationException;
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.HashMap;
import org.omg.CORBA_2_3.portable.OutputStream;
import org.web3d.x3d.sai.BrowserFactory;
import org.web3d.x3d.sai.ExternalBrowser;
import org.web3d.x3d.sai.SFRotation;
import org.web3d.x3d.sai.X3DComponent;
import org.web3d.x3d.sai.X3DField;
import org.web3d.x3d.sai.X3DNode;
import org.web3d.x3d.sai.X3DScene;
import com.essentialreality.Ganti.TimerListener;
import javax.comm.CommPortIdentifier;
import javax.comm.SerialPort;
import org.web3d.x3d.sai.SFVec3f;
import java.lang.Math;
```

Το package παίρνει το όνομά του από την βιβλιοθήκη CP5DLL. Στη συνέχεια εισάγουμε τις απαραίτητες βιβλιοθήκες.

```
public class Ganti extends JFrame {
    static CP5DLL gloves = null;
    Timer timer = null;
    Timer timer1 = null;
    Timer timer2 = null;
    Timer timer3 = null;
    static Enumeration portList;
    static CommPortIdentifier portId;
    static String messageString = "Hello, world!\n";
    static SerialPort serialPort;
    static OutputStream outputStream;
    ExternalBrowser x3dBrowser;
    int currentServoPosition[] = {0, 0, 0, 0, 0, 0, 0, 0, 0};
    int desiredServoPosition[] = {127, 127, 127, 127, 127, 127, 127, 127, 127};
    X3DScene myScene = null;
    private final float p = 3.14f;
    private static int counter = 0;
    Vector rollVec = new Vector();
    Vector pitchVec = new Vector();
```



ΤΕΙ Ηρακλείου Τμήμα Ε.Π.Π.

```
Vector yawVec = new Vector();
X3DScene mainScene;
int xZero, yZero, zZero;

float[] val = new float[3];
float a0 = 0;
float a1 = 0;
float a2 = 0;
float s0 = 0;
float s1 = 0;
float s2 = 0;
float v0 = 0;
float v1 = 0;
float v2 = 0;
float maxa0 = 0;
float mina0 = 10;
float maxs0 = 0;
float mins0 = 10;
float maxv0 = 0;
float minv0 = 10;
float maxa1 = 0;
float mina1 = 10;
float maxs1 = 0;
float mins1 = 10;
float maxv1 = 0;
float minv1 = 10;
float maxa2 = 0;
float mina2 = 10;
float maxs2 = 0;
float mins2 = 10;
float maxv2 = 0;
float minv2 = 10;
float[] trans1 = new float[3];
float[] trans2 = new float[3];
float[] trans3 = new float[3];
float[] tran1 = new float[3];
float[] tran2 = new float[3];
float[] tran3 = new float[3];
int simea = 0;
```

Παραπάνω ορίζουμε τις εξωτερικές μεταβλητές τις οποίες θα χρησιμοποιήσουμε παρακάτω.

```
public Ganti(){
    super("P5 Glove");
    setSize(1000,600);
    setVisible(true);
    setDefaultCloseOperation(EXIT_ON_CLOSE);
}
```

Δημιουργούμε τον constructor της κλάσης και ορίζουμε τις ιδιότητες του JFrame.

```
public X3DScene interfaceX3D(){
    HashMap requestedParameters = new HashMap();
    Container contentPane = getContentPane();
    X3DComponent x3dComp =
    BrowserFactory.createX3DComponent(requestedParameters);
```



ΤΕΙ Ηρακλείου Τμήμα Ε.Π.Π.

```
JComponent x3dPanel = (JComponent)x3dComp.getImplementation();
contentPane.add(x3dPanel, BorderLayout.CENTER);
x3dBrowser = x3dComp.getBrowser();
mainScene = x3dBrowser.createX3DFromURL(new String[]
{"file:///c:/helloWorldtel2.x3d"});
x3dBrowser.replaceWorld(mainScene);
return mainScene;
}
```

Δημιουργούμε το interface του X3D και του ορίζουμε τις ιδιότητές του.

```
class TimerListener implements ActionListener {
    int val1 = 0;
    int val2 = 0;
    int val3 = 0;
    int val4 = 0;
    int val5 = 0;
    int val6 = 0;
    int valtr1 = 0;
    int valtr2 = 0;
    int valtr3 = 0;

    public void actionPerformed (ActionEvent e) {
        gloves.m_P5Devices[0].update();
        val1 = (new Float(gloves.m_P5Devices[0].m_fx)).intValue();
        if(gloves.m_P5Devices[0].m_byButtons[2]) {
            xZero = val1;
        }
        val1 = val1 - xZero;
        if (val1 < -1024) {
            val1 = -1024;
        }else if (val1 > 1024) {
            val1 = 1024;
        }

        int a1 = val1;
        val2 = (new Float(gloves.m_P5Devices[0].m_fy)).intValue();
        if(gloves.m_P5Devices[0].m_byButtons[2]) {
            yZero = val2;
        }
        val2 = val2 - yZero;
        if(val2 < -1024) {
            val2 = -1024;
        }else if (val2 > 1024) {
            val2 = 1024;
        }

        int a2 = val2;
        val3 = (new Float(gloves.m_P5Devices[0].m_fz)).intValue();
        if(gloves.m_P5Devices[0].m_byButtons[2]) {
            zZero = val3;
        }
        val3 = val3 - zZero;
        if(val3 < -1024) {
            val3 = -1024;
        }else if (val3 > 1024) {
            val3 = 1024;
        }
    }
}
```



ΤΕΙ Ηρακλείου Τμήμα Ε.Π.Π.

```
    }  
  
    int a3 = val3;  
    val[0] = Float.parseFloat(Integer.toString(a1)+1)/256;  
    val[1] = Float.parseFloat(Integer.toString(a2)+1)/256;  
    val[2] = Float.parseFloat(Integer.toString(a3)+1)/256;  
    applyTranform(val,myScene);  
    }  
}
```

Δημιουργούμε έναν `actionListener` για την κίνηση του γαντιού στους άξονες (x, y, z). Στη αρχή ορίζουμε την μεταβλητή κίνησης και θέτουμε όριο στις τιμές που μπορεί να πάρει ο κάθε άξονας (όρια κόσμου 3D). Στην συνέχεια κάνουμε βελτιστοποίηση των τιμών, ώστε να έχουμε όσο πιο ομαλή και ακριβής κίνηση γίνεται.

```
class TimerListener1 implements ActionListener{  
    float normalization_coif=1.0f;  
  
    public void actionPerformed (ActionEvent e) {  
        gloves.m_P5Devices[0].update();  
  
        float valtr11 = (new  
Float(gloves.m_P5Devices[0].m_froll)).floatValue();  
        valtr11=normalization_coif*valtr11;  
        if(valtr11 < (-90*normalization_coif)) {  
            valtr11 = 0;  
        } else if (valtr11 > (90*normalization_coif)) {  
            valtr11 = 0;  
        }  
  
        float valtr22 = (new  
Float(gloves.m_P5Devices[0].m_fpitch)).floatValue();  
        valtr22=normalization_coif*valtr22;  
        if (valtr22 < (-90*normalization_coif)) {  
            valtr22 = 0;  
        } else if ((valtr22 > 90*normalization_coif)) {  
            valtr22 = 0;  
        }  
  
        float valtr33 =(new  
Float(gloves.m_P5Devices[0].m_fyaw)).floatValue();  
        valtr33=normalization_coif*valtr33;  
        if (valtr33 < (-90*normalization_coif)) {  
            valtr33 = 0;  
        } else if ((valtr33 > 90*normalization_coif)) {  
            valtr33 = 0;  
        }  
  
        if(counter<40){  
            rollVec.add(counter,Float.toString((valtr11)));  
            pitchVec.add(counter,Float.toString((valtr22)));  
            yawVec.add(counter,Float.toString((valtr33)));  
            counter++;  
        }  
  
        if (counter>=40){
```




ΤΕΙ Ηρακλείου Τμήμα Ε.Π.Π.

```
float frtel=0;
float prtel=0;
float yrtel=0;
float maxroll;
float minroll;
float maxpitch;
float minpitch;
float maxyaw;
float minyaw;
int a=0;
int b = 0;
int d=0;
int c=0;
int e1=0;
int f=0;
int i;
int k=0;

while(k<rollVec.size()) {
    frtel=frtel+Float.parseFloat((String)rollVec.get(k))/2;
    k++;
}

frtel=frtel/counter;
int kp=0;
while(kp<pitchVec.size()) {
    prtel=prtel+Float.parseFloat((String)pitchVec.get(kp))/2;
    kp++;
}

prtel=prtel/counter;

int kpc=0;
while(kpc<yawVec.size()) {
    yrtel=yrtel+Float.parseFloat((String)yawVec.get(kpc))/2;
    kpc++;
}

yrtel=yrtel/counter;
if(java.lang.Math.abs(frtel)>90){
    System.out.println("frtel111111111111="+frtel);
}
if(java.lang.Math.abs(prtel)>90){
    System.out.println("prtel111111111111="+prtel);
}
if(java.lang.Math.abs(yrtel)>90){
    System.out.println("yrtel111111111111="+yrtel);
}

float
angle=(java.lang.Math.abs(frtel)+java.lang.Math.abs(prtel)+java.lang.Math.abs(yrtel))*p
/180;

float rooltel=((frtel)*p/180)/angle;
float pitchtel=((prtel)*p/180)/angle;
float yawtel=((yrtel)*p/180)/angle;

System.out.println("frtel="+frtel+" prtel="+prtel+"
yrtel="+yrtel+" rooltel="+rooltel+" pitchtel="+pitchtel+" yawtel="+yawtel+"
angle="+angle);
```



ΤΕΙ Ηρακλείου Τμήμα Ε.Π.Π.

```
rollVec.removeElementAt(0);
pitchVec.removeElementAt(0);
yawVec.removeElementAt(0);
counter--;

float [] vec=new float[4];
vec[0]=pitchtel;
vec[1]=-yawtel;
vec[2]=-rooltel;
vec[3]=angle;

applyRottation(vec,myScene);
}

float thumb = gloves.m_P5Devices[0].m_byBendSensor_Data[0];
if (thumb < 0) {
    thumb = 0;
} else if (thumb > 64) {
    thumb = 64;
}

float index = gloves.m_P5Devices[0].m_byBendSensor_Data[1];
if (index < 0) {
    index = 0;
} else if (index > 64) {
    index = 64;
}

float middle= gloves.m_P5Devices[0].m_byBendSensor_Data[2];
if (middle < 0) {
    middle = 0;
} else if (middle > 64) {
    middle = 64;
}

float ring = gloves.m_P5Devices[0].m_byBendSensor_Data[3];
if (ring < 0) {
    ring = 0;
} else if (ring > 64) {
    ring = 64;
}

float pinky = gloves.m_P5Devices[0].m_byBendSensor_Data[4];
if (pinky < 0) {
    pinky = 0;
} else if (pinky > 64) {
    pinky = 64;
}

applyRottationFingers(thumb,index,middle,ring,pinky,myScene);
}
}
```

Δημιουργούμε έναν `addListener` για να ανιχνεύσουμε την περιστροφική κίνηση που εκτελεί το γάντι σε κάθε έναν από τους άξονες (x, y, z). Στην αρχή κοινοποιεί τις τιμές $\pm 90^\circ$ και στην συνέχεια αποθηκεύει τις τιμές σε ένα `Vector` (για κάθε άξονα). Τέλος τις



ΤΕΙ Ηρακλείου Τμήμα Ε.Π.Π.

βελτιστοποιεί και ανανεώνει την κίνηση του χεριού.

Το τελευταίο κομμάτι κώδικα αναφέρεται στην κίνηση των δακτύλων. Πρώτα διαβάζει τις τιμές για κάθε δάκτυλο που του στέλνει το γάντι και στην συνέχεια φιλτράρει τις τιμές σε συγκεκριμένο εύρος τιμών.

```
public class moveServos extends Thread{
    public void run() {
        byte[] servoCommand = new byte[3];
        servoCommand[0] = (byte)0xff;
        int tmpVal;
        portList = CommPortIdentifier.getPortIdentifiers();

        while (portList.hasMoreElements()) {
            portId = (CommPortIdentifier) portList.nextElement();
            if (portId.getPortType() == CommPortIdentifier.PORT_SERIAL) {
                if (portId.getName().equals("COM1")) {
                    try {
                        serialPort = (SerialPort)
                            portId.open("RobotGlove", 2000);
                    } catch (PortInUseException e) {
                        System.out.println("Port in Use "+e);
                    }
                    try {
                        outputStream = (OutputStream)
serialPort.getOutputStream();
                    } catch (IOException e) {
                        System.out.println("getOutputStream error "
+e);
                    }
                    try {
                        serialPort.setSerialPortParams(9600,
                            SerialPort.DATABITS_8,
                            SerialPort.STOPBITS_1,
                            SerialPort.PARITY_NONE);
                    } catch (UnsupportedCommOperationException e) {
                        System.out.println("setSerialPortParams
Failure" +e);
                    }
                    try {
                        serialPort.setFlowControlMode(serialPort.FLOWCONTROL_NONE);
                    } catch (UnsupportedCommOperationException e) {
                        System.out.println("setSerialPortParams
Failure" +e);
                    }
                }
            }
        }

        while (true) {
            servoCommand[1] = (byte) 0x05;
            if (desiredServoPosition[5] == currentServoPosition[5]) {
                currentServoPosition[5] = desiredServoPosition[5];
            } else if (desiredServoPosition[5] > currentServoPosition[5]) {
                currentServoPosition[5] = currentServoPosition[5] + 1;
            } else {
                currentServoPosition[5] = currentServoPosition[5] - 1;
            }
        }
    }
}
```



ΤΕΙ Ηρακλείου Τμήμα Ε.Π.Π.

```
}
servoCommand[2] = (byte)currentServoPosition[5];
try {
    outputStream.write(servoCommand);
} catch (IOException e) {
    System.out.println("Serial port write error "+e);
    System.exit(0);
}
try {
    outputStream.flush();
} catch (IOException e) {
    System.out.println("Serial Port flush error "+e);
}

servoCommand[1] = (byte) 0x00;
if (desiredServoPosition[0] == currentServoPosition[0]) {
    currentServoPosition[0] = desiredServoPosition[0];
} else if (desiredServoPosition[0] > currentServoPosition[0]) {
    currentServoPosition[0] = currentServoPosition[0] + 1;
} else {
    currentServoPosition[0] = currentServoPosition[0] - 1;
}
servoCommand[2] = (byte)currentServoPosition[0];
try {
    outputStream.write(servoCommand);
} catch (IOException e) {
    System.out.println("Serial port write error "+e);
    System.exit(0);
}
try {
    outputStream.flush();
} catch (IOException e) {
    System.out.println("Serial Port flush error "+e);
}

servoCommand[1] = (byte) 0x03;
if (desiredServoPosition[3] == currentServoPosition[3]) {
    currentServoPosition[3] = desiredServoPosition[3];
} else if (desiredServoPosition[3] > currentServoPosition[3]) {
    currentServoPosition[3] = currentServoPosition[3] + 1;
} else {
    currentServoPosition[3] = currentServoPosition[3] - 1;
}
servoCommand[2] = (byte)currentServoPosition[3];
try {
    outputStream.write(servoCommand);
} catch (IOException e) {
    System.out.println("Serial port write error "+e);
    System.exit(0);
}
try {
    outputStream.flush();
} catch (IOException e) {
    System.out.println("Serial Port flush error "+e);
}

servoCommand[1] = (byte) 0x04;
if (desiredServoPosition[4] == currentServoPosition[4]) {
    currentServoPosition[4] = desiredServoPosition[4];
```



ΤΕΙ Ηρακλείου Τμήμα Ε.Π.Π.

```
} else if (desiredServoPosition[4] > currentServoPosition[4]) {
    currentServoPosition[4] = currentServoPosition[4] + 1;
} else {
    currentServoPosition[4] = currentServoPosition[4] - 1;
}
servoCommand[2] = (byte)currentServoPosition[4];
try {
    outputStream.write(servoCommand);
} catch (IOException e) {
    System.out.println("Serial port write error "+e);
    System.exit(0);
}
try {
    outputStream.flush();
} catch (IOException e) {
    System.out.println("Serial Port flush error "+e);
}

servoCommand[1] = (byte) 0x02;
desiredServoPosition[2] = 255 - desiredServoPosition[2];
if (desiredServoPosition[2] == currentServoPosition[2]) {
    currentServoPosition[2] = desiredServoPosition[2];
} else if (desiredServoPosition[2] > currentServoPosition[2]) {
    currentServoPosition[2] = currentServoPosition[2] + 1;
} else {
    currentServoPosition[2] = currentServoPosition[2] - 1;
}
servoCommand[2] = (byte)currentServoPosition[2];
try {
    outputStream.write(servoCommand);
} catch (IOException e) {
    System.out.println("Serial port write error "+e);
    System.exit(0);
}
try {
    outputStream.flush();
} catch (IOException e) {
    System.out.println("Serial Port flush error "+e);
}
}
}
```

Αυτή η κλάση είναι υπεύθυνη για την επικοινωνία του γαντιού με τον υπολογιστή μέσω της θύρας usb. Αυτό το κομμάτι του κώδικα το πήραμε έτοιμο, επειδή παραμένει το ίδιο για κάθε πρόγραμμα που θέλει να επικοινωνεί με το γάντι.

```
private void applyTranform(float[] val, X3DScene myScene1 ) {
    X3DNode mat = myScene1.getNamedNode("main");
    if (mat == null) {
        System.out.println("aaaaaaaaaaaaaaaaaaaaaaCouldn't find material named: MAT");
    }
    SFVec3f a=(SFVec3f)mat.getField("translation");
    a.setValue(val);
}
```




ΤΕΙ Ηρακλείου Τμήμα Ε.Π.Π.

Το παραπάνω κομμάτι κώδικα είναι υπεύθυνο για ορισμό του translation του group του γαντιού (όλα τα μοντέλα αντικειμένων που απαρτίζουν το γενικό μοντέλο του γαντιού) στον γενικό κόσμο.

```
public class geometryValues extends Thread{
    public void run(){
        while(true){
            try{
                X3DNode xeri = myScene.getNamedNode("myObject");
                SFVec3f xeriSca = (SFVec3f)xeri.getField("scale");
                SFVec3f xeriTra = (SFVec3f)xeri.getField("translation");
                float[] sizeXeri = new float[3];
                float[] translationXeri = new float[3];
                xeriSca.getValue(sizeXeri);
                xeriTra.getValue(translationXeri);

                float[] sizeIndex1 = new float[3];
                float[] tranIndex1 = new float[3];
                float[] centerIndex1 = new float[3];
                X3DNode index1 = myScene.getNamedNode("index1");
                SFVec3f ind1Tra =
(SFVec3f)index1.getField("translation");
                SFVec3f ind1Geo = (SFVec3f)index1.getField("scale");
                SFVec3f ind1Center = (SFVec3f)index1.getField("center");
                ind1Geo.getValue(sizeIndex1);
                ind1Tra.getValue(tranIndex1);
                ind1Center.getValue(centerIndex1);

                X3DNode sphere2 = myScene.getNamedNode("dad_Sphere2");
                SFVec3f sphere2Tra =
(SFVec3f)sphere2.getField("translation");

                sphere2Tra.getValue(tran2);

                X3DNode sphere1 = myScene.getNamedNode("dad_Sphere1");
                SFVec3f sphere1Tra =
(SFVec3f)sphere1.getField("translation");
                sphere1Tra.getValue(tran1);

                float[] sizeSphere3 = new float[3];

                X3DNode sphere3 = myScene.getNamedNode("dad_Sphere3");
                SFVec3f sphere3Tra =
(SFVec3f)sphere3.getField("translation");
                sphere3Tra.getValue(tran3);

                float index =
gloves.m_P5Devices[0].m_byBendSensor_Data[1];
                float middle=
gloves.m_P5Devices[0].m_byBendSensor_Data[2];
                float ring =
gloves.m_P5Devices[0].m_byBendSensor_Data[3];
                float pinky =
gloves.m_P5Devices[0].m_byBendSensor_Data[4];

                a0 = Math.abs(tran2[0]-val[0]);
                a1 = Math.abs(tran2[1]-val[1]);
            }
        }
    }
}
```



ΤΕΙ Ηρακλείου Τμήμα Ε.Π.Π.

```
a2 = Math.abs(tran2[2]-val[2]);

s0 = Math.abs(tran3[0]-val[0]);
s1 = Math.abs(tran3[1]-val[1]);
s2 = Math.abs(tran3[2]-val[2]);

v0 = Math.abs(tran1[0]-val[0]);
v1 = Math.abs(tran1[1]-val[1]);
v2 = Math.abs(tran1[2]-val[2]);

if(simea == 0){
    while((a0<=4) && (a1<=4) && (a2<=10) && (index>40)
&& (middle>40) && (ring>40) && (pinky>40)){
        trans2[0] = val[0];
        trans2[1] = val[1]-1;
        trans2[2] = val[2]-2;
        sphere2Tra.setValue(trans2);
        simea = 1;
        index =
gloves.m_P5Devices[0].m_byBendSensor_Data[1];
        middle=
gloves.m_P5Devices[0].m_byBendSensor_Data[2];
        ring =
gloves.m_P5Devices[0].m_byBendSensor_Data[3];
        pinky =
gloves.m_P5Devices[0].m_byBendSensor_Data[4];
        if((index<40) && (middle<40) && (ring<40) &&
(pinky<40)){
            simea = 0;
            break;
        }
    }
    while((s0<=4) && (s1<=4) && (s2<=10) && (index>40)
&& (middle>40) && (ring>40) && (pinky>40)){
        trans3[0] = val[0];
        trans3[1] = val[1]-1;
        trans3[2] = val[2]-2;
        sphere3Tra.setValue(trans3);
        simea = 1;
        index =
gloves.m_P5Devices[0].m_byBendSensor_Data[1];
        middle=
gloves.m_P5Devices[0].m_byBendSensor_Data[2];
        ring =
gloves.m_P5Devices[0].m_byBendSensor_Data[3];
        pinky =
gloves.m_P5Devices[0].m_byBendSensor_Data[4];
        if((index<40) && (middle<40) && (ring<40) &&
(pinky<40)){
            simea = 0;
            break;
        }
    }
    while((v0<=4) && (v1<=4) && (v2<=10) && (index>40)
&& (middle>40) && (ring>40) && (pinky>40)){
        trans1[0] = val[0];
        trans1[1] = val[1]-1;
        trans1[2] = val[2]-2;
        sphere1Tra.setValue(trans1);
```



ΤΕΙ Ηρακλείου Τμήμα Ε.Π.Π.

```
        simea = 1;
        index =
gloves.m_P5Devices[0].m_byBendSensor_Data[1];
        middle=
gloves.m_P5Devices[0].m_byBendSensor_Data[2];
        ring =
gloves.m_P5Devices[0].m_byBendSensor_Data[3];
        pinky =
gloves.m_P5Devices[0].m_byBendSensor_Data[4];
        if((index<40) && (middle<40) && (ring<40) &&
(pinky<40)){
                simea = 0;
                break;
        }
    }else{
        simea = 0;
    }

    if(gloves.m_P5Devices[0].m_byButtons[0])
        xeriTra.setValue(translationXeri);
    }catch(Exception ex){
        System.out.println("Exception!!!! : "+ex);
    }
}
}
}
```

Η παραπάνω κλάση παίρνει την θέση του γαντιού και των αντικειμένων από το περιβάλλον, βρίσκει την απόσταση που έχει το χέρι από το κάθε αντικείμενο, ελέγχει αν τα δάχτυλα έχουν κλίσει ώστε αν είναι κοντά σε κάποιο αντικείμενο να το πιάσει (το αντικείμενο που αλληλεπιδρά μαζί του, ακολουθεί την κίνηση του γαντιού μέχρι να ανοίξουν τα δάχτυλα του γαντιού).

Αναλυτικά: το πρώτο κομμάτι του κώδικα διαβάσει τις τιμές θέσης του κάθε αντικειμένου. Το επόμενο κομμάτι κώδικα διαβάσει τις τιμές των δακτύλων, στην συνέχεια υπολογίζει τις αποστάσεις μεταξύ του γαντιού και των αντικειμένων. Έπειτα κάνει έλεγχο ώστε να διασφαλιστεί ότι θα πιάσει μόνο ένα αντικείμενο τη φορά. Έτσι αν η σημαία-μεταβλητή δεν έχει τιμή «0» τότε συνεχίζει στον επόμενο έλεγχο, αν η τιμή της σημαία-μεταβλητή είναι ίση με «1» τότε το γάντι κρατάει ήδη ένα αντικείμενο και δεν μπορεί να πιάσει άλλο. Ο επόμενος βρόχος ελέγχει αν το γάντι είναι κοντά σε ένα από τα αντικείμενα και αν ταυτόχρονα όλα τα δάχτυλα του γαντιού είναι κλειστά. Εάν επαληθευτεί ο προαναφερθείς βρόχος το αντικείμενο που θα αλληλεπιδράσει με το γάντι η κίνηση του θα γίνει ίδια με την κίνηση του γαντιού. Ο μόνος τρόπος για να βγούμε από αυτή την κατάσταση είναι να αλλάξει μία από τις μεταβλητές των δακτύλων (γι' αυτό τις ξαναδιαβάζουμε μέσα στο βρόχο, ώστε να ανανεώνονται οι τιμές τους).

Το τελευταίο κομμάτι κώδικα μας επιτρέπει να μηδενίσουμε (επανατοποθετήσουμε) το γάντι στην αρχική θέση με το πάτημα του κουμπιού «C» του P5Glove. Αυτό μας είναι ιδιαίτερα χρήσιμο σε περίπτωση που το γάντι είτε ξεφύγει από την οπτική μας γωνία, είτε το χέρι μας βγει εκτός εμβέλειας του δέκτη του P5Glove.



ΤΕΙ Ηρακλείου Τμήμα Ε.Π.Π.

```
private void applyRottation(float [] vec,X3DScene myScene) {
    X3DNode mat = myScene.getNamedNode("main");
    if (mat == null) {
        System.out.println("aaaaaaaaaaaaaaaaCouldn't find material named: MAT");
    }
    SFRotation c=(SFRotation)mat.getField("rotation");
    c.setValue(vec);
}
```

Το παραπάνω κομμάτι κώδικα είναι υπεύθυνο για ορισμό του rotation του group του γαντιού (όλα τα μοντέλα αντικειμένων που απαρτίζουν το γενικό μοντέλο του γαντιού) στον γενικό κόσμο.

```
private void applyRottationFingers(float thumb, float index, float middle, float
ring, float pinky, X3DScene myScene) {
    float [] th=new float[4];
    X3DNode matthum = myScene.getNamedNode("thumb");
    SFRotation thum=(SFRotation)matthum.getField("rotation");
    th[0]=0;
    th[1]=0;
    th[2]=-1;
    th[3]=((45/32)*thumb*p)/180;
    thum.setValue(th);
    float [] indx=new float[4];
    X3DNode matind = myScene.getNamedNode("index0");
    SFRotation ind=(SFRotation)matind.getField("rotation");
    indx[0]=-1;
    indx[1]=0;
    indx[2]=0;
    indx[3]=((45/32)*index*p)/180;
    ind.setValue(indx);
    float [] indx1=new float[4];
    X3DNode matind1 = myScene.getNamedNode("index1");
    SFRotation ind1=(SFRotation)matind1.getField("rotation");
    indx1[0]=-1;
    indx1[1]=0;
    indx1[2]=0;
    indx1[3]=((90/32)*index*p)/180;
    ind1.setValue(indx1);
    float [] middl=new float[4];
    X3DNode matmiddle = myScene.getNamedNode("middle0");
    SFRotation mid=(SFRotation)matmiddle.getField("rotation");
    middl[0]=-1;
    middl[1]=0;
    middl[2]=0;
    middl[3]=((45/32)*middle*p)/180;
    mid.setValue(middl);
    float [] middl1=new float[4];
    X3DNode matmind1 = myScene.getNamedNode("middle1");
    SFRotation mid1=(SFRotation)matmind1.getField("rotation");
    middl1[0]=-1;
    middl1[1]=0;
    middl1[2]=0;
    middl1[3]=((90/32)*middle*p)/180;
    mid1.setValue(middl1);
    float [] rig=new float[4];
```



ΤΕΙ Ηρακλείου Τμήμα Ε.Π.Π.

```
X3DNode matrig= myScene.getNamedNode("ring0");
SFRotation r=(SFRotation)matrig.getField("rotation");
rig[0]=-1;
rig[1]=0;
rig[2]=0;
rig[3]=((45/32)*ring*p)/180;
r.setValue(rig);
float [] rig1=new float[4];
X3DNode matrig1= myScene.getNamedNode("ring1");
SFRotation r1=(SFRotation)matrig1.getField("rotation");
rig1[0]=-1;
rig1[1]=0;
rig1[2]=0;
rig1[3]=((90/32)*ring*p)/180;
r1.setValue(rig1);
float [] pink=new float[4];
X3DNode matpink= myScene.getNamedNode("pinky0");
SFRotation pi=(SFRotation)matpink.getField("rotation");
pink[0]=-1;
pink[1]=0;
pink[2]=0;
pink[3]=((45/32)*pinky*p)/180;
pi.setValue(pink);
float [] pink1=new float[4];
X3DNode matpink1= myScene.getNamedNode("pinky1");
SFRotation p1=(SFRotation)matpink1.getField("rotation");
pink1[0]=-1;
pink1[1]=0;
pink1[2]=0;
pink1[3]=((90/32)*pinky*p)/180;
p1.setValue(pink1);
double ansin=((45/32)*index*p)/180;
float [] intr=new float[3];
matind1= myScene.getNamedNode("index1");
SFVec3f pti=(SFVec3f)matind1.getField("translation");
intr[0]=-0.251670f;
intr[1]=Float.parseFloat(Double.toString(-
(java.lang.Math.sin(2*ansin)*1.30354/2 + java.lang.Math.sin(ansin)*0.672)));
intr[2]=Float.parseFloat(Double.toString(-1.63238-(1.34401-
1.34401*(java.lang.Math.sin(ansin)))/2));
pti.setValue(intr);
double anmid=((45/32)*middle*p)/180;
float [] inmi=new float[3];
matmind11= myScene.getNamedNode("middle1");
SFVec3f pmid=(SFVec3f)matmind11.getField("translation");
inmi[0]=-0.251670f;
inmi[1]=Float.parseFloat(Double.toString(-
(java.lang.Math.sin(2*anmid)*1.30354/2 + java.lang.Math.sin(anmid)*0.672)));
inmi[2]=Float.parseFloat(Double.toString(-1.63238-(1.34401-
1.34401*(java.lang.Math.sin(anmid)))/2));
pmid.setValue(inmi);
double anrig=((45/32)*ring*p)/180;
float [] inri=new float[3];
matrig1= myScene.getNamedNode("ring1");
SFVec3f prig=(SFVec3f)matrig1.getField("translation");
inri[0]=-0.251670f;
inri[1]=Float.parseFloat(Double.toString(-
(java.lang.Math.sin(2*anrig)*1.30354/2 + java.lang.Math.sin(anrig)*0.672)));
inri[2]=Float.parseFloat(Double.toString(-1.63238-(1.34401-
```



ΤΕΙ Ηρακλείου Τμήμα Ε.Π.Π.

```
1.34401*(java.lang.Math.sin(anrig))/2));
    prig.setValue(inri);
    double anpink=((45/32)*pinkyp)/180;
    float [] inpi=new float[3];
    matpink1= myScene.getNamedNode("pinkyl");
    SFVec3f ppin=(SFVec3f)matpink1.getField("translation");
    inpi[0]=-0.251670f;
    inpi[1]=Float.parseFloat(Double.toString(-
(java.lang.Math.sin(2*anpink)*1.30354/2 + java.lang.Math.sin(anpink)*0.672)));
    inpi[2]=Float.parseFloat(Double.toString(-1.63238-(1.34401-
1.34401*(java.lang.Math.sin(anpink))/2)));
    ppin.setValue(inpi);
    float rtel=-2.222870f;
    float [] thumpp=new float[3];
    matthum= myScene.getNamedNode("thumb");
    SFVec3f thumbbbb=(SFVec3f)matthum.getField("translation");
    thumpp[0]=rtel-rtel*(thumb/64);
    thumpp[1]=rtel*(thumb/64);
    thumpp[2]=-2;
    thumbbbb.setValue(thumpp);
}
```

Ο παραπάνω κώδικας παίρνει τις τιμές του rotation του κάθε δακτύλου και υπολογίζει με μαθηματικές πράξεις την κλίση του κάθε μέρους των δακτύλων. Δηλαδή το κάθε δάκτυλο αποτελείται από 3 μέρη (μοντέλο αντικειμένου), το κάθε ένα από αυτά θα στραφεί με διαφορετική γωνία ώστε η συνολική κίνηση του δακτύλου να προσεγγίζει την κίνηση που κάνει το ανθρώπινο δάκτυλο όταν λυγίζει.

```
public void init(){
    timer = new Timer(10, new TimerListener());
    timer.setInitialDelay(10000);
    timer.start();

    timer1 = new Timer(10, new TimerListener1());
    timer1.setInitialDelay(10030);
    timer1.start();

    System.out.println("Starting moveServo Thread");
    moveServos msThread = new moveServos();
    msThread.start();
    System.out.println("Thread Started");
    myScene=interfaceX3D();

    System.out.println("Starting Geometric Thread");
    geometryValues gmThread = new geometryValues();
    gmThread.start();
    System.out.println("Thread Started");
}
```

Στην παραπάνω κλάση αρχίζουν την λειτουργία τους τέσσερις διαφορετικές διεργασίες. Η πρώτη ορίζει την κίνηση του χεριού, η δεύτερη την κλίση, η τρίτη την επικοινωνία με το γάντι, και η τέταρτη είναι υπεύθυνη για την αλληλεπίδραση του γαντιού με τον κόσμο.



ΤΕΙ Ηρακλείου Τμήμα Ε.Π.Π.

```
public static void main(String[] args){
    Ganti glove = new Ganti();
    gloves = new CP5DLL();
    gloves.P5_Init();
    gloves.P5_SetForwardZ(-1);
    gloves.P5_SetFilterMode((int)gloves.P5_FILTER_DEADBAND |
(int)gloves.P5_FILTER_AVERAGE);
    gloves.P5_SetPrediction(1);
    gloves.P5_SetFilterAmount(20, (float)14.0);
    System.out.println(gloves.m_nNumP5);
    System.out.println("m_nNumP5=" + gloves.m_nNumP5+"");
    System.out.println("VendorID = " + gloves.m_P5Devices[0].VendorID + ",
ProductID = " + gloves.m_P5Devices[0].ProductID + ", Version = " +
gloves.m_P5Devices[0].Version + "");
    System.out.println("ProductString = " +
gloves.m_P5Devices[0].ProductString + ", ProductID = " +
gloves.m_P5Devices[0].ManufacturerString + ", Version = " +
gloves.m_P5Devices[0].SerialNumString + "");
    System.out.println("m_nDeviceID = " + gloves.m_P5Devices[0].m_nDeviceID +
"");
    System.out.println("m_nMajorRevisionNumber = " +
gloves.m_P5Devices[0].m_nMajorRevisionNumber + ", m_nMinorRevisionNumber = " +
gloves.m_P5Devices[0].m_nMinorRevisionNumber+"");
    System.out.println("m_nGloveType = " +
gloves.m_P5Devices[0].m_nGloveType+"");
    for(int i=0; i<3; i++) {
        gloves.m_P5Devices[0].updateFull();
        int skipCounter = 0;
        while(gloves.m_P5Devices[0].m_fx == 0.0) {
            gloves.m_P5Devices[0].updateFull();
            skipCounter++;
        }
        System.out.println("m_fx = " + gloves.m_P5Devices[0].m_fx+"");
        System.out.println("m_fyaw= " + gloves.m_P5Devices[0].m_fyaw);
        System.out.println("m_byBendSensor_Data=" +
gloves.m_P5Devices[0].m_byBendSensor_Data[0]);
        System.out.println("skipCounter=" + skipCounter);
    }
    glove.init();
}
```

Η «main» ξεκινάει το πρόγραμμά και μας δίνει πληροφορίες για το γάντι.



ΤΕΙ Ηρακλείου Τμήμα Ε.Π.Π.

3.2 Ο κώδικας του X3D για την δημιουργία του τρισδιάστατου περιβάλλοντος και του εικονικού μοντέλου του γαντιού

Παρακάτω θα δούμε τον κώδικα του X3D που αποθηκεύσαμε όλα τα γραφικά του τρισδιάστατου κόσμου μας.

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

Το αρχείο του X3D ξεκινάει πάντα με την δήλωση της έκδοσης της xml που χρησιμοποιεί και ακολουθεί η κωδικοποίηση των χαρακτήρων του αρχείου, όπως ακριβώς και σ' ένα συνηθισμένο αρχείο xml. Στην συνέχεια έχουμε την δήλωση του profil του X3D που χρησιμοποιεί το αρχείο. Κάθε profil έχει δικές του δυνατότητες. Με το scene υποδηλώνουμε ότι ξεκινάει η δημιουργία του εικονικού μας κόσμου. Στην συνέχεια έχουμε την δημιουργία των αντικειμένων που θα είναι στον εικονικό μας κόσμο.

```
<Transform DEF='main' translation='0 0 0' rotation='0 0 0 0'>  
<Transform DEF='myObject'  
translation='.46717 .09133 .06846'  
rotation='0 0 0 0'  
scale='4.10862 3 3'  
center='0 0 0'>  
<Shape>  
<Appearance>  
<Material diffuseColor='0 0 1' />  
</Appearance>  
<Box  
containerField='geometry'  
size='1 1 1' />  
</Shape>  
</Transform>
```

Με το translation δηλώνουμε την θέση του αντικειμένου μας στο τρισδιάστατο χώρο (x, y, z). Με το rotation δηλώνουμε την κλίση που έχει ως προς τους άξονες το αντικείμενο, το πεδίο scale δίνει την αναλογία του μεγέθους του αντικειμένου, το center δίνει το κέντρο του αντικειμένου, το shape φτιάχνει το αντικείμενο (και του προσθέτει κάποιες ιδιότητες), στο πεδίο appearance δηλώνουμε το material (χρώμα). Το box υποδηλώνει ότι το σχήμα είναι παραλληλεπίπεδο και με το size του δίνουμε μέγεθος στους άξονες. Για να σταματήσουμε να επηρεάζουμε το συγκεκριμένο αντικείμενο, κλείνουμε τα πεδία shape και transform που είχαμε ανοίξει παραπάνω. Το ίδιο γίνεται και για τα υπόλοιπα στοιχεία του χεριού, το μόνο που αλλάζει είναι το όνομα, η θέση, το χρώμα και το μέγεθός τους.

```
<Transform DEF='thumb'  
translation='-2.22287 -.28112 -.55688'  
rotation='0 0 0 0'>  
<Transform DEF='thumb0'
```



```
translation='0 0 0'  
rotation='0 0 0 0'  
<Shape>  
<Appearance>  
<Material diffuseColor='0 1 0' />  
</Appearance>  
<Box  
  containerField='geometry'  
  size='.64199 .64199 1.45821' />  
</Shape>  
</Transform>  
  
<Transform DEF='thumb1'  
  translation='0 0 -1.26008'  
  rotation='0 0 0 0'  
<Shape>  
<Appearance>  
<Material diffuseColor='0 0 1' />  
</Appearance>  
<Box  
  size='.64976 .64976 2.0534' />  
</Shape>  
</Transform>  
</Transform>  
  
<Transform DEF='index'  
  translation='-.79817 1.12867 -2.74134'  
  rotation='0 0 0 0'  
  
<Transform DEF='index0'  
  translation='-.25167 0 -1.05244'  
  rotation='0 0 0 0'  
<Shape>  
<Appearance>  
<Material diffuseColor='0 1 0' />  
</Appearance>  
<Box  
  containerField='geometry'  
  size='1 1 1.34401' />  
</Shape>  
</Transform>  
  
<Transform DEF='index1'  
  translation='-.25167 0 -2.28791'  
  rotation='0 0 0 0'  
  center='0 0 0'  
  scale='1 1 1'  
<Shape>  
<Appearance>  
<Material diffuseColor='0 0 1' />  
</Appearance>  
<Box  
  containerField='geometry'  
  size='1 1 1' />  
</Shape>  
</Transform>  
  
<Transform DEF='index2'  
  translation='-.20591 0 .45758'
```



```
rotation='0 0 0 0'>
<Shape>
<Appearance>
  <Material diffuseColor='0 0 1' />
</Appearance>
<Box
  containerField='geometry'
  size='1 1 1.63238' />
</Shape>
</Transform>
</Transform>

<Transform DEF='middle'
  translation='.3295 1.05899 -2.75726'
  rotation='0 0 0 0'>

  <Transform DEF='middle0'
translation='-.25167 0 -1.05244'
  rotation='0 0 0 0'>
    <Shape>
      <Appearance>
        <Material diffuseColor='0 1 0' />
      </Appearance>
      <Box size='1 1 1.34401' />
    </Shape>
  </Transform>

  <Transform DEF='middle1'
  translation='-.25167 0 -2.28791'
rotation='0 0 0 0'>
    <Shape>
      <Appearance>
        <Material diffuseColor='0 0 1' />
      </Appearance>
      <Box
        containerField='geometry'
        size='1 1 1.30354' />
    </Shape>
  </Transform>

  <Transform DEF='middle2'
  translation='-.20591 0 .45758'
  rotation='0 0 0 0'>
    <Shape>
      <Appearance>
        <Material diffuseColor='0 0 1' />
      </Appearance>
      <Box size='1 1 1.63238' />
    </Shape>
  </Transform>
</Transform>

  <Transform DEF='ring'
translation='1.13439 .8841 -2.75979'
  rotation='0 0 0 0'>

  <Transform DEF='ring0'
translation='-.25167 0 -1.05244'
  rotation='0 0 0 0'>
```



```
<Shape>
<Appearance>
<Material diffuseColor='0 1 0' />
</Appearance>
<Box
  containerField='geometry'
  size='1 1 1.34401' />
</Shape>
</Transform>

<Transform DEF='ring1'
  translation='-.25167 0 -2.28791'
rotation='0 0 0 0'>
<Shape>
<Appearance>
<Material diffuseColor='0 0 1' />
</Appearance>
<Box
  containerField='geometry'
  size='1 1 1.30354' />
</Shape>
</Transform>

<Transform DEF='ring2'
  translation='-.20591 0 .45758'
rotation='0 0 0 0'>
<Shape>
<Appearance>
<Material diffuseColor='0 0 1' />
</Appearance>
<Box size='1 1 1.63238' />
</Shape>
</Transform>
</Transform>

<Transform DEF='pinky'
translation='2.18646 .98458 -2.79046'
rotation='0 0 0 0'>

<Transform DEF='pinky0'
translation='-.25167 0 -1.05244'
rotation='0 0 0 0'>
<Shape>
<Appearance>
<Material diffuseColor='0 1 0' />
</Appearance>
<Box
  containerField='geometry'
  size='1 1 1.34401' />
</Shape>
</Transform>

<Transform DEF='pinky1'
translation='-.25167 0 -2.28791'
rotation='0 0 0 0'>
<Shape>
<Appearance>
<Material diffuseColor='0 0 1' />
</Appearance>
```



ΤΕΙ Ηρακλείου Τμήμα Ε.Π.Π.

```
<Box
  containerField='geometry'
  size='1 1 1.30354' />
</Shape>
</Transform>

<Transform DEF='pinky2'
  translation='-.20591 0 .45758'
  rotation='0 0 0 0'>
  <Shape>
  <Appearance>
  <Material diffuseColor='0 0 1' />
  </Appearance>
  <Box
    containerField='geometry'
    size='1 1 1.63238' />
  </Shape>
  </Transform>
</Transform>
</Transform>
```

Με το τελευταίο transform κλείνει και το γενικό transform με το όνομα main που είχε ανοίξει στις πρώτες γραμμές. Έτσι καθαρίζουμε ότι τα αντικείμενα που απαρτίζουν το χέρι είναι στο ίδιο group αντικειμένων και κινούνται μαζί στο χώρο.

Ο κώδικας που ακολουθεί είναι για την δημιουργία των άσφαιρων-αντικειμένων αλληλεπίδρασης που υπάρχουν διάσπαρτα στο κόσμο. Τα αντικείμενα αυτά έχουν την ίδια δομή με τα αντικείμενα box του group του μοντέλου του γατιού με μερικές αλλαγές. Στο πεδίο appearance-material έχουμε επιλογές για το πως θα εμφανίζονται οι σφαίρες όπως το shininess και το ambientIntensity, που καθορίζουν το πόσο γυαλιστερό και πόσο διάφανο είναι το αντικείμενο. Επίσης ένα σημείο που διαφέρει σε σχέση με τα box είναι ότι τα sphere έχουν ακτίνα (radius) και όχι size και scale όπως τα box.

```
<Transform DEF='dad_Sphere2'
  translation='12.5 1.0 -18.5'>
  <Shape DEF='Sphere2'
    containerField='children'>
  <Appearance
    containerField='appearance'>
  <Material DEF='Green'
    containerField='material'
    ambientIntensity='0.200'
    shininess='0.200'
    diffuseColor='0 1 0' />
  </Appearance>
  <Sphere DEF='GeoSphere2'
    containerField='geometry'
    radius='1.000' />
  </Shape>
</Transform>

<Transform DEF='dad_Sphere1'
  translation='-8.48985 5.087736 -12.61817'>
  <Shape DEF='Sphere1'
```




```
containerField='children'>
<Appearance
containerField='appearance'>
  <Material DEF='Red'
    containerField='material'
    ambientIntensity='0.200'
    shininess='0.200'
    diffuseColor='1 0 0' />
  </Appearance>
  <Sphere DEF='GeoSphere1'
    containerField='geometry'
    radius='1.000' />
  </Shape>
</Transform>

<Transform DEF='dad_Sphere3'
translation='-4.48985 1.087736 -6.61817'>
  <Shape DEF='Sphere3'
    containerField='children'>
    <Appearance
      containerField='appearance'>
        <Material DEF='Cyan'
          containerField='material'
          ambientIntensity='0.200'
          shininess='0.200'
          diffuseColor='0 1 1' />
        </Appearance>
        <Sphere DEF='GeoSphere3'
          containerField='geometry'
          radius='1.000' />
        </Shape>
      </Transform>
```

Στο πεδίο viewpoint ορίζουμε ποια θα είναι η οπτική μας γωνία σε σχέση με τον κόσμο.

```
<Viewpoint DEF='Viewpoint1'
description='Viewpoint1'
jump='true'
fieldOfView='0.785'
position='.35302 8.30661 19.1383'
orientation='- .999 0 .048 .283' />
</Scene>
</X3D>
```

Κλείνοντας το scene, σταματάει και η δημιουργία της σκηνής μας και το X3D κλείνει το αρχείο που είναι έτοιμο για χρήση.



4. Παράρτημα

4.1 Μελλοντικές προσθήκες και δυνατότητες χρήσης του εικονικού μοντέλου

Το project που αναπτύσσουμε θα χρησιμοποιηθεί σαν πρωτότυπο για μελλοντική εργασία και εφαρμογή του κώδικα μας σε ένα μεγαλύτερο project. Το project θα χρησιμοποιεί το γάντι σε περιβάλλον εικονικής πραγματικότητας και θα εκτελεί σε αυτό τις λειτουργίες αλληλεπίδρασης που θα του έχουν προσθέσει. Γενικά ο τομέας στον οποίον δουλεύουμε έχει πολύ μέλλον και μπορούν να γίνουν πολλές προσθήκες τόσο σε επίπεδο software όσο και σε επίπεδο hardware.

Για παράδειγμα πρώτα να δούμε τι μπορεί να γίνει με την συσκευή αλληλεπίδρασης που χρησιμοποιούμε. Θα μπορούσαμε να βελτιώσουμε την απόκριση του γαντιού, πειραματιζόμενοι με τα κατάλληλα calibration των αισθητήρων, ώστε να πάρουμε καλύτερα αποτελέσματα για το περιβάλλον που το χρησιμοποιούμε. Από την άλλη υπάρχουν συσκευές γαντιών οι οποίες είναι πολύ καλύτερες από την συγκεκριμένη που χρησιμοποιούμε και δίνουν περισσότερες δυνατότητες και καλύτερες μετρήσεις. Μπορεί κατ' επέκταση να προστεθούν κάποια γυαλιά εικονικής πραγματικότητας, τα οποία θα δίνουν την δυνατότητα στον χρήστη να αλληλεπιδράσει στον χώρο σαν να είναι ο ίδιος “μέσα” στον τρισδιάστατο εικονικό περιβάλλον, καθώς και πλήθος άλλων περιφερικών συσκευών που κάνουν την διάδραση ακόμα πιο αληθοφανή.

Το “θέμα” με τα τρισδιάστατα γραφικά είναι τόσο “ζεστό” και υπάρχει τόση πολύ θέληση για επέκταση προς αυτή την κατεύθυνση ώστε όσο περνάει ο καιρός γίνονται νέες ανακαλύψεις για καλύτερα οπτικά αποτελέσματα (όπως εφέ, σκίαση, κίνηση, διαφάνειες κ.α.) και γρηγορότερη επεξεργασία γραφικών.



4.2 Επεξήγηση εννοιών

Java: Η Java είναι μια αντικειμενοστραφής γλώσσα προγραμματισμού, η οποία έχει δημιουργηθεί και αναπτύσσεται από την Sun Microsystems. Είναι φτιαγμένη με την ιδιότητα να “τρέχει” σε διαφορετικά λειτουργικά συστήματα χωρίς περιορισμούς. Αυτή της η φορητότητα της δίνει την δυνατότητα να εγκατασταθεί σε διάφορες συσκευές όπως palmtops, κινητά τηλέφωνα, web servers και high-end μηχανήματα. Το πρόγραμμα που δίνει στη Java αυτή της την δυνατότητα είναι το Java Virtual Machine (JVM). Το JVM είναι υπεύθυνο για το compilation και την εκτέλεση του κώδικα, επίσης το JVM δεν έχει κάποια εξάρτηση από hardware και αυτός είναι ο βασικός λόγος που δίνει στην Java την δυνατότητα της να τρέχει σε διαφορετικά λειτουργικά περιβάλλοντα χωρίς κάποιο πρόβλημα. Αυτό που πρέπει να προσεχτεί εδώ είναι ότι για να μπορέσει να τρέξει κάποιο πρόγραμμα γραμμένο σε Java πρέπει το σύστημα στο οποίο θα τρέξουμε το πρόγραμμα να έχει το δικό του JVM. Τα αρχεία «.jar» είναι τα αντίστοιχα «.exe» των windows δηλαδή εκτελούνται με διπλό κλικ εφόσον υπάρχουν οι απαιτούμενες βιβλιοθήκες εγκατεστημένες στο σύστημα.

3D: Το 3D (στα ελληνικά τρισδιάστατο περιβάλλον) είναι μια τρισδιάστατη αναπαράσταση εικόνων στον υπολογιστή μέσω διαφόρων τεχνικών απεικόνισης. Στην επιστήμη υπολογιστών χρησιμοποιείται για να αναπαραστήσει ένα τρισδιάστατο περιβάλλον και διάφορα αντικείμενα. Χρήσεις βρίσκει στα τρισδιάστατα παιχνίδια αλλά και στην εικονική πραγματικότητα, πεδίο το οποίο αναπτύσσουμε παρακάτω.

Εικονική πραγματικότητα: Η εικονική πραγματικότητα είναι εφαρμογή του τρισδιάστατου χώρου στον υπολογιστή η οποία επιτρέπει μια αλληλεπίδραση στον 3D χώρο η οποία μεταφράζεται σε μετακίνηση αντικειμένων στον εικονικό χώρο, περιήγηση και αντιγραφή κινήσεων του ανθρώπου και μεταφορά τους στο εικονικό μοντέλο του στον υπολογιστή.

Java Standard Edition – Java Enterprise Edition: Το Java JDK (αναλύεται παρακάτω) έρχεται σε μερικές εκδόσεις που είναι για διαφορετικές, ή αλληλεπικαλυπτόμενες δουλειές.

1. Java Standard Edition: Είναι το πιο απλό JDK που μπορεί κάποιος να κατεβάσει και έχει όλα τα απαραίτητα εργαλεία για την δημιουργία εφαρμογών σε Java.
2. Java Enterprise Edition: Είναι το ίδιο JDK με το Standard Edition με την μόνη διαφορά ότι εμπεριέχει στο όλο “πακέτο”, υποστήριξη για MySQL και το Glass Fish που είναι ένας server χειρισμού servlets.
3. Java Micro Edition: Είναι ένα εργαλείο που έχει υποστήριξη εφαρμογών για κινητά τηλέφωνα και γενικά “κινητές εφαρμογές”.
4. JavaFX: Είναι ένα εργαλείο για να δημιουργείς πλούσιες εφαρμογές Internet.

Java JDK: Το JDK (Java Development Kit) είναι ένα προϊόν της Sun Microsystems το οποίο προορίζεται για αυτούς που αναπτύσσουν τα προγράμματά τους σε Java. Από την πρώτη στιγμή που ξεκίνησε η Java το JDK είναι το ποιο πολυχρησιμοποιημένο πρόγραμμα



ΤΕΙ Ηρακλείου Τμήμα Ε.Π.Π.

Java SDK (Software Development Kit). Είναι ελεύθερο πρόγραμμα από το 2006. Τα περιεχόμενα του JDK είναι τα εξής:

5. java:- είναι το πρόγραμμα που φορτώνει της εφαρμογές Java. Δουλειά του είναι να μεταφράζει τα αρχεία “.class” που βγάζει σαν έξοδο ο javac compiler (μεταγλωττιστής).
6. javac:- είναι ο μεταγλωττιστής που μετατρέπει το κώδικα java σε java bytecode (εντολές τις οποίες καταλαβαίνει η Java Virtual Machine).
7. jar:- είναι μια λειτουργία που ενώνει διάφορες συσχετιζόμενες βιβλιοθήκες της ίδιας κλάσης σε ένα “.jar” αρχείο. Επίσης μπορεί να διαχειριστεί τα διάφορα “.jar” αρχεία.
8. javadoc:- δημιουργεί αρχεία τεκμηρίωσης αυτόματα από τα σχόλια που υπάρχουν στον πηγαίο κώδικα του προγράμματος.
9. jdb:- ο debugger (διορθωτής) της Java. Δουλειά του είναι να βρίσκει λάθη στον κώδικα και να προτείνει λύσεις στον χρήστη.
10. javap:- είναι ο υπεύθυνος για την αποσύνθεση των αρχείων “.class” για την περεταίρω χρήση τους.
11. appletviewer:- χρησιμοποιείτε για να εκτελεστούν και να διορθωθούν οι εφαρμογές applet χωρίς την χρήση του περιηγητή.
12. javah:- γεννήτρια C headers και stubs για την γραφή “native” μεθόδων.
13. extcheck:- είναι μια λειτουργία η οποία ελέγχει τυχόν συγκρούσεις από jars.
14. apt:- εργαλείο επεξεργασίας σχολιασμών.
15. Υπάρχουν και κάποια εργαλεία τα οποία βρίσκονται σε πειραματικό στάδιο : jhat, jstack, jstat, jstatd, jinfo, jmap, idlj, policytool, VisualVM.

Μαζί με το JDK έρχεται και το Java Runtime Environment που έχει την δυνατότητα να εκτελεί προγράμματα σε java. Εμπεριέχει ένα Java Virtual Machine και διάφορες βιβλιοθήκες jar που είναι χρήσιμες για τους προγραμματιστές.

Java SDK: Το SDK είναι ένα σετ εργαλείων τα οποία χρησιμεύουν στους προγραμματιστές για την δημιουργία εφαρμογών για συγκεκριμένο λειτουργικό σύστημα ή hardware. Εδώ στην περίπτωση του Java SDK έχουμε μια σειρά εργαλείων που χρησιμεύουν για την δημιουργία εφαρμογών σε java.

C: Γλώσσα προγραμματισμού που δημιουργήθηκε από την Bell Telephone Laboratories το 1972 και είναι μια γλώσσα γενικού σκοπού, δομημένη, και η ευρεία χρήση της είναι για την δημιουργία εφαρμογών και προγράμματα συστήματος (λειτουργικό σύστημα κ.α.)

COM: Η θύρα επικοινωνίας COM είναι η αρχική, συναντάτε ακόμα, μορφή της serial-port και είναι ένας διάυλος που χρησιμοποιήθηκε για τους IBM συμβατούς υπολογιστές.

Universal Serial Bus (USB): Στον τομέα της πληροφορικής ένας Γενικός Σειριακός Διάυλος (Universal Serial Bus (USB)) είναι ένα αναγνωρισμένο πρωτόκολλο σειριακής



ΤΕΙ Ηρακλείου Τμήμα Ε.Π.Π.

σύνδεσης συσκευών σε έναν υπολογιστή. Ο δίαυλος USB έχει σχεδιαστεί έτσι ώστε να υποστηρίζει την ταυτόχρονη σύνδεση πολλαπλών συσκευών και παράλληλα δίνει στις συσκευές αυτές διάφορες ιδιότητες όπως: "τοποθέτηση και άμεση λειτουργία" (plug and play), δυνατότητας αφαίρεσης της συσκευής χωρίς να χρειάζεται να επανεκκινήσεις τον υπολογιστή, μπορεί να τροφοδοτήσει συσκευές χαμηλής κατανάλωσης χωρίς την ανάγκη για εξωτερική τροφοδοσία και τέλος δίνει την δυνατότητα σε κάποιες συσκευές να λειτουργήσουν χωρίς να χρειάζονται τους drivers του κατασκευαστή τους.

Java Servlets: Οι servlets είναι προγράμματα φτιαγμένα σε Java. Παρέχουν δυναμικό περιεχόμενο σε έναν Web Server ο οποίος στηρίζεται σε μια πλατφόρμα Java. Το περιεχόμενο που δημιουργείται συνήθως είναι HTML αλλά μπορεί να συναντηθούν και άλλες μορφές όπως XML. Με απλά λόγια ένα servlet είναι ένα object το οποίο λαμβάνει ένα request και εμφανίζει στον browser την απάντηση για το συγκεκριμένο request.

HTML: Είναι τα αρχικά για την HyperText Markup Language και είναι η κύρια γλώσσα που χρησιμοποιείται στις ιστοσελίδες. Μας δίνει την δυνατότητα να περιγράψουμε το πώς είναι δομημένο το κείμενο σε ένα αρχείο. Η HTML είναι δομημένη σε tags που έχουν αυτή την μορφή: "<tag>". Τα αρχεία HTML έχουν την κατάληξη ".html" ή ".htm". Μέσα στον κώδικα της HTML μπορεί να μπει κώδικας από άλλες παρόμοιες γλώσσες όπως την Javascript.

XML: Τα αρχικά XML ανταποκρίνονται στην Extensible Markup Language. Η XML είναι μια γενικού σκοπού ιδιότητα η οποία χρησιμεύει για να γίνουν τροποποιήσιμες mark-up γλώσσες, σε αυτό τον τομέα βοηθάει η ελευθερία που δίνει η XML στον χρήστη να δημιουργεί δικά του στοιχεία. Στόχος της γλώσσας είναι η δομημένη αναπαράσταση των δεδομένων για χρήση κυρίως στο internet.

X3D: Το X3D είναι ένα είδος αρχείου (ISO standard) βασισμένο σε XML για να αναπαραστήσει τρισδιάστατα γραφικά και διάδοχος της VRML (Virtual Reality Modeling Language). Το X3D έχει την ιδιότητα να κωδικοποιεί τρισδιάστατες «σκηνές» χρησιμοποιώντας XML συντακτικό όπως έκανε το αντίστοιχο (Open Inventor) συντακτικό της VRML.

Path: Ένα path (διαδρομή αρχείου) είναι μια γενική μορφή ονόματος για ένα αρχείο ή κάποιο κατάλογο η οποία υποδηλώνει μια μοναδική τοποθεσία σε ένα σύστημα αρχείων. Το path δείχνει σε μια τοποθεσία στο σύστημα αρχείων ακολουθώντας την δενδροειδή ιεράρχηση των καταλόγων και εκφράζεται σαν μία ακολουθία χαρακτήρων στην οποία τα στοιχεία του path διαχωρίζονται από έναν ειδικό χαρακτήρα υποδηλώνοντας έτσι τον κάθε κατάλογο. Αυτός ο ειδικός χαρακτήρας είναι συνήθως η πλάγια γραμμή "/", την ανάποδη πλάγια "\" ή μία άνω-κάτω τελεία ":" αν και μερικά υπολογιστικά συστήματα κάνουν χρήση διαφορετικών ειδικών χαρακτήρων. Υπάρχουν δυο διαφορετικά είδη διαδρομών. Η "σχετική διαδρομή" η οποία είναι μια διαδρομή σχετική με τον κατάλογο τον οποίο χρησιμοποιεί η



ΤΕΙ Ηρακλείου Τμήμα Ε.Π.Π.

εκάστοτε εφαρμογή, και η πλήρης ή απόλυτη διαδρομή ή οποία δίνει μια τοποθεσία χωρίς να παίρνει υπ' όψιν τον κατάλογο που χρησιμοποιεί η εφαρμογή αλλά παρουσιάζει την διαδρομή της τοποθεσίας σε σχέση με τον αρχικό κατάλογο.

Device Driver: Ένας "οδηγός" (driver) επιτρέπει την επικοινωνία της συσκευής με τον υπολογιστή στον οποίο αυτή είναι συνδεδεμένη. Όταν ένα πρόγραμμα καλεί μια "ρουτίνα" διεργασιών στον "οδηγό", ο "οδηγός" δίνει τις απαραίτητες εντολές στην συσκευή. Όταν η συσκευή δίνει δεδομένα πίσω στον "οδηγό" αυτός με την σειρά του μπορεί να εκκινήσει διεργασίες στο πρόγραμμα το οποίο ζήτησε την αρχική επικοινωνία με την συσκευή. Οι "οδηγοί" των συσκευών είναι απόλυτα εξαρτώμενοι από το υλικό για το οποίο φτιαχτήκαν και είναι συγκεκριμένοι για κάθε λειτουργικό σύστημα.

Mousepad: Το mousepad είναι μια επιφάνεια πάνω στην οποία χειριζόμαστε το "ποντίκι" του υπολογιστή. Είναι ιδιαίτερα χρήσιμο αφού αυξάνει την απόκριση και την ακρίβεια της κίνησης του "ποντικιού".

Viewpoint: Το viewpoint είναι μια ιδιότητα του X3D μέσω της οποίας καθορίζουμε την οπτική γωνία που θα έχει ο χρήστης στον κόσμο τον οποίο έχει δημιουργήσει. Το viewpoint έχει διάφορες ιδιότητες όπως θέση στον χώρο και κλίση οπτικής γωνίας. Μπορεί να μένει στατικό στον χώρο ή ακόμα και να κινείται μαζί με τις κινήσεις που κάνει ο χρήστης.



ΤΕΙ Ηρακλείου Τμήμα Ε.Π.Π.

4.3 Πηγές

Το βιβλίο «Pro Java 6 3D : Game Development» του Andrew Davison από τις εκδόσεις Apress του 2007.

Οι πηγές μας ήταν κυρίως μέσω του διαδικτύου. Παραθέτουμε τα site που αναζητήσαμε πληροφορίες καθώς και τα site που μπορείτε να βρείτε χρήσιμα προγράμματα.

www.netbeans.org : το επίσημο site του προγράμματος.

www.xj3d.org : το site του browser που χρησιμοποιούμε.

www.wikipedia.com : online εγκυκλοπαίδεια.

www.google.com : μηχανή αναζήτησης.

www.vivaty.com : το site του προγράμματος.

java.sun.com : το site της Sun Microsystems για την γλώσσα προγραμματισμού java.

www.web3d.org : το site του οργανισμού που είναι υπεύθυνος για το X3D.

www.oblong.com : το blog που ασχολείται με το project oblong.