ΑΛΕΞΑΝΡΕΙΟ Τ.Ε.Ι. ΘΕΣΣΑΛΟΝΙΚΗΣ

ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΩΝ ΕΦΑΡΜΟΦΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Πτυχιακή Εργασία

Games Programming, 2D, 3D Simulations

Μούτσιος Νικόλαος

Επιβλέπων καθηγητής: Μπινιατίδης Συμεών

Θεσσαλονίκη 2010

Πρόλογος

Για πολλά χρόνια, οι προγραμματιστές ανέπτυσαν παιχνίδια για την Java πλατφόρμα χρησιμοποιώντας το standard API της. Ενώ κάποια από αυτά πέτυχαν χρησιμοποιώντας Java 2D, Java 3D και πλήθος άλλων APIs,κάποια προβλήματα της πλατφόρμας είχαν ως αποτέλεσμα πολλοί άνθρωποι να την παραβλέπουν και να μην την χρησιμοποιούν ως πλατφόρμα ανάπτυξης παιχνιδιών και όλα μαζί οδήγησαν στο συμπέρασμα ότι η Java δεν είναι κατάλληλη για την ανάπτυξη ποιοτικών παιχνιδιών. Ενώ σίγουρα η Java δεν είναι κατάλληλη για ανάπτυξη παιχνιδιών σε πολλές παιχνιδομηχανές, είναι στην πραγματικότητα αναγκαία για την ανάπτυξη παιχνιδιών σε προσωπικούς υπολογιστές, PDAs και κινητά τηλέφωνα.

Ένα από τα βασικά πλεονεκτήματα της Java έναντι των περισσοτέρων άλλων γλωσσών είναι η ανεξαρτησία του λειτουργικού συστήματος και πλατφόρμας. Τα προγράμματα που είναι γραμμένα σε Java τρέχουνε ακριβώς το ίδιο σε Windows, Linux, Unix και Macintosh (σύντομα θα τρέχουν και σε Playstation καθώς και σε άλλες κονσόλες παιχνιδιών) χωρίς να χρειαστεί να ξαναγίνει μεταγλώττιση (compiling) ή να αλλάξει ο πηγαίος κώδικας για κάθε διαφορετικό λειτουργικό σύστημα.

Το παρόν παίγνιο σχεδιάστηκε εξ' ολοκλήρου με Java. Είναι μία εφαρμογή για PC η οποία ως στόχο της έχει την ανάδειξη ενός δείγματος των τεχνολογιών που προσφέρονται από τα API που διαθέτει η Java ως τεχνολογία μέσα από ένα παιχνίδι προορισμένο για ψυχαγωγικούς σκοπούς. Είναι ένα κατά κύριο λόγο τρισδιάστατο παιχνίδι ενώ για το Γραφικό Περιβάλλον Χρήστη (GUI) χρησιμοποιήθηκε Swing.

Abstract

For many years, developers were writing games for the Java platform using its standard API. While some of them were successful using Java 2D,

Java 3D many problems of the platform caused many people to overlooking Java as a platform for developing high quality Games. While Java is not suitable for Game Development in many game consoles it is compelling in personal computers, PDAs and cellular phones.

One of the main advantages of Java against other languages is that its platform is indepented from the Operation System. Programes written in Java are running the same exact way in Windows, Linux or Macintosh without having to recompile them or to change its source code.

This Game is written exclusivly in Java. Its an application designed for a personal computer that serves a purpose of showing a sample of the Java 3D technology provided by its API, through entertainment. Its a three dimensional game while its GUI is written with Swing.

ΠΕΡΙΕΧΟΜΕΝΑ

ΚΕΦΑΛΑΙΟ 1: Εισαγωγή

Ανέκαθεν υπήρχε η θέληση και η περιέργεια για μια εισαγωγή στον μαγικό και χωρίς όρια κόσμο των γραφικών. Η εφαρμογή αυτή ξεκίνησε με αφορμή την εκπόνηση της πτυχιακής μου εργασίας και είναι η αρχή μιας πρώτης σοβαρής προσπάθειας στην εκμάθηση του Java 3D API και στις βασικές έννοιες των γραφικών. Η ανάπτυξη της θα συνεχιστεί με πειραματισμούς και υλοποιήσεις νέων πραγμάτων με στόχο την γνώση και την εμπειρία.

Στα κεφάλαια που ακολουθούν περιγράφεται το επιτραπέζιο παιχνίδι, όπως δημιουργήθηκε με τα περιεχόμενα και τους κανόνες του. Στη συνέχεια ακολουθεί μια περίληψη των τεχνολογιών που χρησιμοποιήθηκαν. Έπειτα η περιγραφή και ανάλυση των κλάσεων που σχεδιάστηκαν έτσι ώστε να υλοποιηθούν οι αντίστοιχες λειτουργίες καθώς και να σχεδιαστούν τα περιεχόμενα του παιχνιδιού (κτίρια, πιόνια κτλ.) σε 3D περιβάλλον και τέλος ακολουθούν προτάσεις για βελτίωση της εφαρμογής καθώς και σχέδια για την επέκταση της.

ΚΕΦΑΛΑΙΟ 2: Περιγραφή και Ανάλυση των κανόνων του παιχνιδιού

Το παιχνίδι αυτό καθ’ εαυτό υπάρχει εδώ και πολλά χρόνια ως επιτραπέζιο. Έχοντας το στυλ του κλασικού παιχνιδιού της Μονόπολης ο παίκτης έχει την δυνατότητα να αγοράζει και να χτίζει πολυτελή ξενοδοχεία.

2.1 Περιεχόμενα Παιχνιδιού

Το παιχνίδι αποτελείται από:

- Ένα ταμπλό.
- Τέσσερα πιόνια,ένα για κάθε παίκτη.
- Ένα ζάρι αριθμημένο και ένα με εικόνες για την άδεια κτισίματος.
- Χρήματα.
- Εικοσιπέντε κτίρια χωρισμένα σε οχτώ ξενοδοχειακές μονάδες μαζί με τους εξωτερικούς τους χώρους.
- Οχτώ κάρτες,μία για κάθε ξενοδοχείο.

2.2 Κανόνες Παιχνιδιού

Το παιχνίδι αποτελείται από 2 έως 4 παίκτες.Διαλέγει ο καθένας από ένα πιόνι,ανάλογα με το χρώμα της αρεσκείας του (κόκκινο, μπλέ, πράσινο, κίτρινο).

Στον κάθε παίκτη,αναλογεί ένα ποσό που ανέρχεται στις 12.000, το οποίο και μοιράζεται στην αρχή του παιχνιδιου.

Οι παίκτες τοποθετούν τα πιόνια τους στο ταμπλό.

Το ταμπλό,εκτός από τις 4 θέσεις-μία για κάθε παίκτη-αποτελείται κι από:

- Τη διαδρομή,την οποία θα ακολουθήσουν οι παίκτες.Η διαδρομή αυτή είναι που καθορίζει το τί μπορεί να αγοράσει ή να χτίσει ο κάθε παίκτης,ανάλογα με το κουτάκι στο οποίο θα βρεθεί ρίχνοντας το ζάρι.Η διαδρομή παρέχει στους παίκτες 2 check points,2 πιθανότητες για δωρεάν χτίσιμο κτιρίου και 3 δωρεάν παροχές εισόδου.
- Οχτώ ξενοδοχειακά συγκροτήματα,τα οποία αναλυτικά είναι:

    1. ***Boomerang:*** ένα κτίριο
    2. ***President:*** τέσσερα κτίρια
    3. ***Waikiki:*** πέντε κτίρια
    4. ***Taj Mahal:*** τρία κτίρια
    5. ***Safari:*** τρία κτίρια
    6. ***Fujiyama:*** τρία κτίρια
    7. ***L'Etoile:*** πέντε κτίρια
    8. ***Royal:*** τέσσερα κτίρια

    Σε όλα τα συγκροτήματα, εφόσον τελειώσει το χτίσιμο όλων των κτιρίων τους μπορούν να χτιστούν οι αντίστοιχοι εξωτερικοί χώροι. Υπάρχει ένας για κάθε ξενοδοχείο.

Αφού λοιπόν έχουν τοποθετηθεί τα πιόνια στο ταμπλό κι έχουν μοιραστεί τα χρήματα,οι παίκτες ρίχνουν το ζάρι.Όταν δύο ή περισσότεροι παίκτες φέρουν τον ίδιο αριθμό,ξαναρίχνουν προκειμένου να καθοριστει η σειρά με την οποία θα παίξει ο κάθε παίκτης.

Παίζει πρώτος αυτός που θα φέρει το μεγαλύτερο αριθμό.Όποιος,κατά τη διάρκεια του παιχνιδιού φέρει τον αριθμό 6,του δίνεται η δυνατότητα να ξαναπαίξει.

Στόχος του κάθε παίκτη είναι να αγοράσει κι να χτίσει,όσο το δυνατόν περισσότερα κτίρια, ολοκληρώνοντας τα ξενοδοχειακά συγκροτήματα.Η

αγορά ξεκινάει από το οικόπεδο.Άν ο παίκτης που αγόρασε το οικόπεδο δε χτίσει κανένα κτίριο,τότε κάποιος από τους υπόλοιπους παίκτες έχει τη δυνατότητα να αγοράσει το οικόπεδο μισοτιμής.Το συγκρότημα κατοχυρώνεται σε αυτόν που θα χτίσει έστω κι ένα κτίριό του.

Το κάθε ξενοδοχείο έχει από μία κάρτα.Στην κάρτα αυτή περιλαμβάνονται πληροφορίες για το χρηματικό ποσό που θα πρέπει να δώσει στον ιδιοκτήτη του συγκροτήματος ο παίκτης που θα βρεθεί στην εμβέλεια του οικοπέδου στο οποίο βρίσκεται το συγκρότημα,ακολουθώντας τη διαδρομή.Το χρηματικό ποσό καθορίζεται από τον αριθμό των κτιρίων που έχουν χτιστεί και τον αριθμό που θα φέρει ο παίκτης με το ζάρι.Όσο μεγαλύτερος είναι ο αριθμός του ζαριού και όσο περισσοτερα τα κτίρια που έχουν χτιστεί,τόσο μεγαλύτερο είναι το ποσό που πρέπει να καταβάλλει ο παίκτης.

Από το παιχνίδι αποχωρεί ο παίκτης που μένει χωρίς χρήματα. Νικητής θεωρείται ο παίκτης που θα μείνει τελευταίος.

ΚΕΦΑΛΑΙΟ 3: Τεχνολογίες που χρησιμοποιήθηκαν

3.1 Java 3D

Στα μέσα της χρονιάς 2003, ο Doug Twilleager είπε την τώρα διαβόητη φράση «Η Java 3D 1.4 είναι για την ώρα σε αναμονή».

Ο Doug Twilleager είναι ο chief architect του τμήματος Τεχνολογίας παιχνιδιών στην Sun, και ένας από τους σχεδιαστές της Java 3D.

Ένας πιθανός λόγος για την «αναμονή» είναι ότι η ανάπτυξη της Java3D προέρχεται από το Τμήμα γραφικών 3D υλικού. Καθώς οι κάρτες γραφικών από εταιρίες όπως η ATI και η nVidia ξεπέρασαν την τεχνολογία της Sun, το τμήμα άρχισε να γίνεται λιγότερο κερδοφόρο. Έγιναν περικοπές και σε μη κερδοφόρα projects, όπως η Java3D, δόθηκε μικρότερη προτεραιότητα. Τον Μάρτιο του 2004, ο Doug Twilleager επέστρεψε, αυτή τη φορά ανακοινώνοντας ότι η Sun έκανε την Java 3D διαθέσιμη μέσω δημόσιας χρήσης άδειας.

Η αναγέννηση της Java 3D είναι αποτέλεσμα της δουλειάς πολύ λίγων ανθρώπων-κλειδιά, μέσα σε αυτούς και ο Doug Twilleager, καθώς και η χρησιμοποίησή της σε μεγάλα projects όπως το Mars Rover και το Looking Glass. Η Java 3D πέρασε στο προχωρημένο τμήμα ανάπτυξης λογισμικού, ένα τμήμα μέσα στην Sun υποστηριζόμενο από υψηλότερου επιπέδου διοίκηση.

3.1.1 Java 3D API:

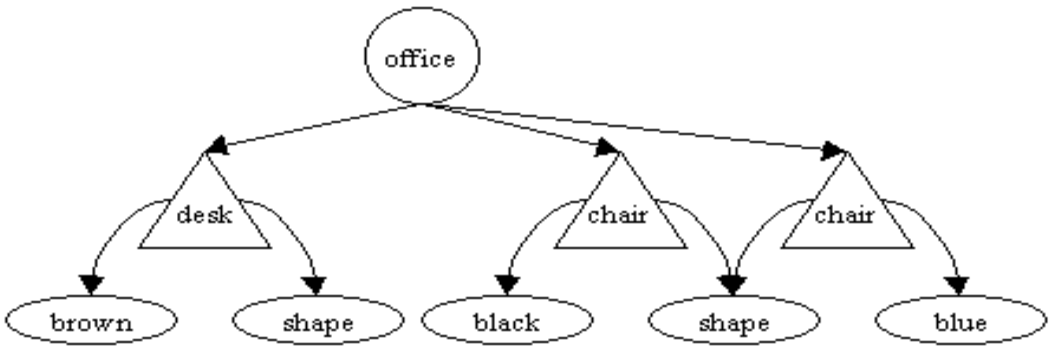
Το API της Java 3D προσφέρει μια συλλογή από υψηλού επιπέδου εργαλεία για την κατασκευή, απόδοση και χειραγώγιση του γράφου της σκηνής (scene graph) η οποία και απαρτίζεται από γεωμετρία, φως, ήχους και πολλά άλλα.

Υπάρχουν δύο παραλλαγές της Java 3D: η μία υλοποιημένη επάνω στην OpenGL, η άλλη επάνω στα γραφικά DirectX. Το χαμηλού επιπέδου API χειρίζεται την υψηλότερη απόδοση και τα διάφορα επίπεδα pixels, ενώ η 3D σκηνή, η λογική των εφαρμογών και οι αλληλεπιδράσεις των σκηνών αναπτύσσονται σε κώδικα Java. Αυτή η διπλή προσέγγιση ενθαρρύνει την μεταφορά των εφαρμογών, ανεξαρτησία υλικού και υψηλής ταχύτητας απόδοση.

3.1.2 Το Scene Graph

Το Scene Graph της Java 3D είναι ένα οδηγούμενος ακυκλικός γράφος (DAG), οπότε και υπάρχει μια σχέση γονιού-παιδιού ανάμεσα στους περισσότερους από τους κόμβους του και ο γράφος δεν περιέχει επαναλήψεις.

Είναι δυνατόν για τους κόμβους να μπορούν να διαμοιραστούν σε έναν γράφο, για παράδειγμα η αντιγραφή γεωμετρικών πληροφοριών ενός σχήματος. Ένα απλοποιημένο scene graph ενός γραφείου φαίνεται στην εικόνα 1. Το γκρουπ γραφείο περιέχει κόμβους για το γραφείο και δύο καρέκλες. Κάθε κόμβος χρησιμοποιεί άλλους κόμβους σχήματος και χρώματος καθώς οι πληροφορίες για το σχήμα της καρέκλας μοιράζεται.

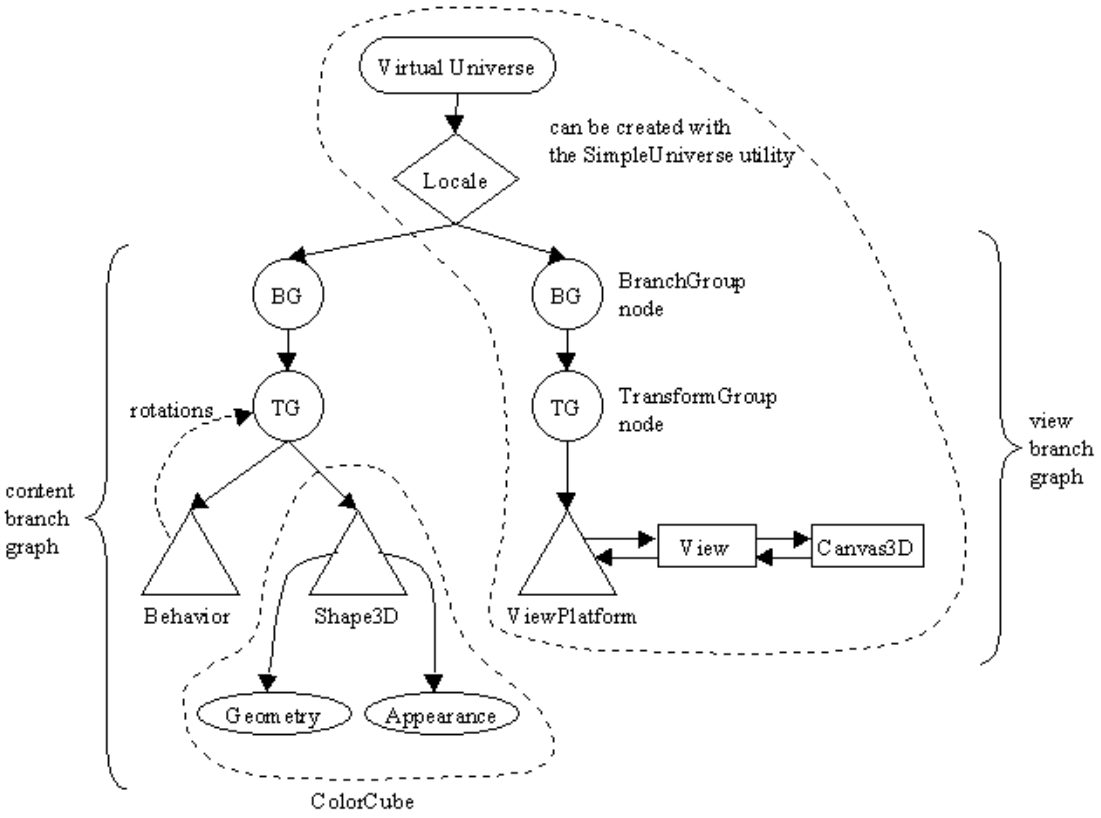


Εικόνα 3.1 : Το Scene Graph ενός γραφείου.

Ένας κόμβος-σχήμα χρησιμοποιεί το Shape3D (ή υποκατηγορίες του) ως «συσκευασία» κόμβων γεωμετρικών πληροφοριών (Geometry) ή

εμφάνισης(Appearance). Η κυρίως κλάση γεωμετρίας είναι η GeometryArray, η οποία χρησιμοποιείται για τον σχεδιασμό σημείων, γραμμών και πολυγώνων. Υπάρχουν πολλές κλάσεις Appearance, χρώματος, texture και διαφάνειας. Περιβαλλοντικοί κόμβοι χειρίζονται ρουτίνες όπως του φωτός, του ήχου και της συμπεριφοράς τα οποία αλλάζουν τον τεχνητό κόσμο.

Ομάδες κόμβων είναι «συσκευασίες» για άλλες ομάδες ή κόμβους φύλλων. Οι κόμβοι φύλλα είναι συνήθως σχήματα ή κόμβοι του περιβάλλοντος. Μια κλάση-ομάδα υποστηρίζει την τοποθέτηση και τον προσανατολισμό των παιδιών της. Για παράδειγμα, ένα BranchGroup επιτρέπει στα παιδιά του να προστίθενται ή να αφαιρούνται από το scene Graph σε χρόνο εκτέλεσης, ενώ ένα TransformGroup επιτρέπει την θέση και τον προσανατολισμό των παιδιών της να αλλάζει.



Εικόνα 3.2 : Το Scene Graph του Hello Universe.

Το VirtualUniverse είναι ο κόμβος κορυφή σε κάθε scene graph και αντιπροσωπεύει τον τεχνητό κόσμο με το σύστημα συντεταγμένων του. Το Locale παίζει το ρόλο της θέσης του scene graph μέσα στον τεχνητό κόσμο.

Κάτω από τον κόμβο Locale υπάρχουν πάντα δυο υπογράφοι.Το αριστερό κλαδί είναι κλαδί γράφος περιεχομένων(content branch graph), κρατάει προγραμματιστικό περιεχόμενο όπως η γεωμετρία, το φως, textures και το φόντο. Το content branch graph διαφέρει σημαντικά από εφαρμογή σε εφαρμογή.

Το δεξί κλαδί κάτω από το Locale είναι το κλαδί γράφος του οπτικού πεδίου (view branch graph) και καθορίζει την θέση, τον προσανατολισμό και την προοπτική του παρατηρητή όπως αυτός κοιτάει τον τεχνητό κόσμο από τον φυσικό. Ο κόμβος ViewPlatform κρατάει την θέση του παρατηρητή στον τεχνητό κόσμο, ο κόμβος View δηλώνει το πώς να μεταβληθεί αυτό που βλέπει ο παρατηρητής σε εικόνα του φυσικού κόσμου (μια 2D εικόνα στην οθόνη). Ο κόμβος Canvas3D είναι ένα Java GUI αντικείμενο που επιτρέπει μια 2D εικόνα να τοποθετηθεί μέσα σε μια Java εφαρμογή ή applet.

Το VirtualUniverse, το Locale και το view branch graph έχουν συχνά την ίδια δομή σε πολλές εφαρμογές, εφόσον πολλά προγράμματα χρησιμοποιούν ένα απλό Locale και βλέπουν τον τεχνητό κόσμο μέσω μιας 2D εικόνας σε μια οθόνη. Για αυτές τις εφαρμογές οι σχετικές κλάσεις μπορούν να δημιουργηθούν με το SimpleUniverse της Java 3D, ανακουφίζοντας τον προγραμματιστή από πολύ δουλειά σε επίπεδο κατασκευής του γράφου.

3.1.3 Τα δυνατά σημεία της Java 3D

Το Scene Graph της Java 3D έχει ως αποτέλεσμα ο προγραμματισμός να γίνεται πολύ πιο εύκολος για αρχάριους επειδή δίνει έμφαση στον σχεδιασμό της σκηνής παρά στην απόδοση, αποκρύπτοντας τις λεπτομέρειες των γραφικών σε χαμηλό επίπεδο. Ένα scene graph φυσιολογικά υποστηρίζει περίπλοκα γραφικά στοιχεία όπως 3D γεωμετρίες, συμπεριφορές, μεθόδους φωτός, διαλογή αντικειμένων και ανίχνευση συγκρούσεων. Σε επίπεδο εφαρμογής, το scene graph μπορεί να χρησιμοποιηθεί για την ομαδοποίηση σχημάτων με παρόμοιες ιδιότητες, να εκτελέσει view culling , level of detail selection, execution culling και κλάδεμα συμπεριφοράς – βελτιστοποιήσεις οι οποίες πρέπει να γραφούν απευθείας από τον προγραμματιστή σε χαμηλότερου επιπέδου APIs. Τέλος η Java 3D χρησιμοποιεί την διαδικασία

multithreading της Java για την εκτέλεση παράλληλων γράφων εκτέλεσης και απόδοσης οι οποίες αποτελούν πολύ χρήσιμες βελτιστοποιήσεις.

Απόδοση**.** Η Java3D είναι σχεδιασμένη με βάση την απόδοση, την οποία και επιτυγχάνει σε υψηλό επίπεδο μέσω βελτιστοποιήσεων του scene graph και σε χαμηλό επίπεδο με το γεγονός ότι χτίστικε επάνω από την OpenGL ή την DirectX.

Κάποιες βελτιστοποιήσεις του scene graph είναι διαθέσιμες μέσω capability bits, τα οποία δηλώνουν τι λειτουργίες μπορούν ή δεν μπορούν να εκτελεστούν σε χρόνο εκτέλεσης (για παράδειγμα απαγόρευση της κίνησης ενός αντικειμένου).

3.2 AUTODESK 3DS STUDIO MAX

Είναι ένα πακέτο λογισμικού για την δημιουργία και επεξεργασία 3D μοντέλων. Μεταξύ άλλων παράγει αρχεία του τύπου 3ds τα οποία και χρησιμοποιήθηκαν από αυτήν την εφαρμογή για την εύκολη και γρήγορη παραγωγή μοντέλων.

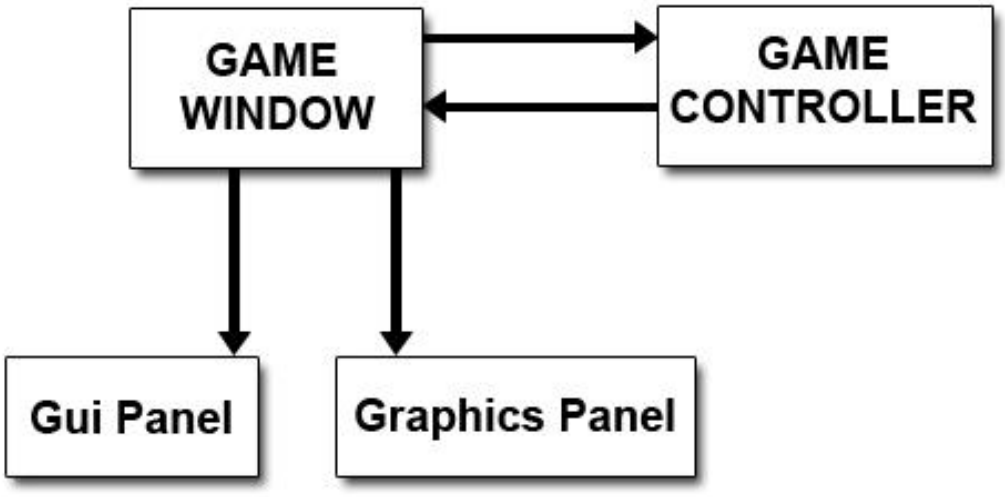
3.3 ADOBE PHOTOSHOP CS3

Είναι ένα επαγγελματικό λογισμικό για την επεξεργασία εικόνων το οποίο μπορεί να χρησιμοποιηθεί από αρχάριους μέχρι επαγγελματίες. Είναι κοινά αποδεκτό από πάρα πολλούς γραφίστες. Χρησιμοποιήθηκε για την δημιουργία των διαφόρων textures που «ντύνουν» τα 3D αντικείμενα της σκηνής του παιχνιδιού.

ΚΕΦΑΛΑΙΟ 4: Ανάλυση Κλάσεων

Στο κεφάλαιο αυτό θα γίνει μια προσπάθεια για μια όσο το δυνατόν αφαιρετική προσέγγιση στην περιγραφή των κλάσεων που αποτελούν το συγκεκριμένο παιχνίδι.

Το παρακάτω διάγραμμα απεικονίζει τις κυριότερες κλάσεις των διαφόρων ενοτήτων με τις συσχετίσεις τους.



Εικόνα 4.1 : Διάγραμμα BoardHotelGame.

Κατά την εκκίνηση του παιχνιδιού ο χρήστης καλείται να συμπληρώσει μια φόρμα στην οποία δηλώνει με πόσους παίκτες θα ξεκινήσει η παρτίδα, τα ονόματά τους και το χρώμα προτίμησής τους. Αφού συμπληρωθεί η φόρμα, μπορεί να ξεκινήσει το παιχνίδι.

Με το πάτημα του κουμπιού «start» δημιουργείται ένα GameWindow και ένας GameController. Το GameWindow είναι υπεύθυνο για την απεικόνιση του Γραφικού Περιβάλλοντος Χρήστη (GUI) το οποίο και αλληλεπιδρά με το παιχνίδι και του παραθύρου γραφικών στο οποίο απεικονίζεται το παιχνίδι αυτό καθεαυτό με όλα τα τρισδιάστατα αντικείμενά του. Οι λειτουργίες αυτές επιτυγχάνονται με την δημιουργία του GuiPanel και GraphicsPanel αντίστοιχα.΄

Ο GameController είναι ο κύριος υπεύθυνος και αυτός που συντονίζει την πορεία του παιχνιδιού. Ανάμεσα στις αρμοδιότητες του είναι και η διαδικασία επιλογής του ποιός παίκτης θα παίξει πρώτος, να αποφασίζει εάν ένας παίκτης μπορεί να αγοράσει ένα οικόπεδο, να χτίσει ένα ξενοδοχείο ή να δώσει εντολή στο GameWindow να ζωγραφίζει ή να καταστρέφει αντικείμενα από το γραφικό περιβάλλον (το χτίσιμο ενός ξενοδοχείου για παράδειγμα). Ο

GameController επίσης «ακούει» τις αποφάσεις που παίρνει ο χρήστης μέσω του GuiPanel.

Οι κλάσεις αυτές με τις λειτουργίες τους θα περιγραφούν αναλυτικότερα στις επόμενες ενότητες.

4.1 Η Κλάση GameWindow

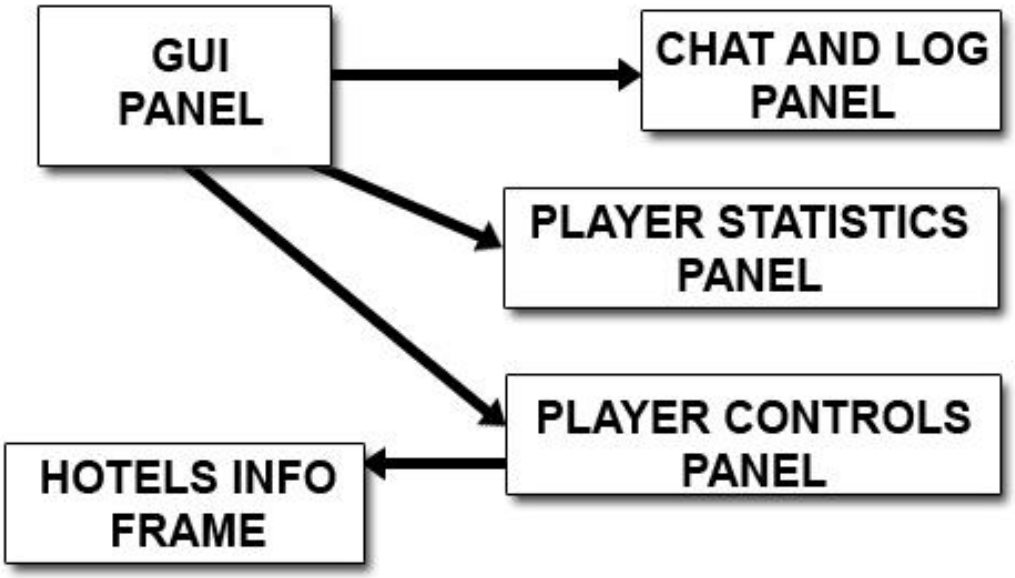
Σε αυτήν την ενότητα θα περιγραφεί αναλυτικότερα η κλάση GameWindow. Όπως αναφέρθηκε είναι υπεύθυνη για την εκτέλεση όλων των λειτουργιών που απεικονίζουν πληροφορίες ή αντικείμενα στην οθόνη και τα οποία σε τελική ανάλυση ανανεώνουν και εξελίσουν την πορεία του παιχνιδιού.

Το GameWindow είναι ουσιαστικά ένα Jpanel το οποίο περιέχει τα:

- GuiPanel
- GraphicsPanel

Περιέχει επίσης listeners που με τη βοήθειά τους ανανεώνει τα στατιστικά και τις διαθέσιμες κινήσεις του παίκτη που έχει τη σειρά και δίνει οδηγίες στο GraphicsPanel να σχεδιάσει ή να καταστρέψει αντικείμενα από την σκηνή.

4.1.1 Η Κλάση GuiPanel

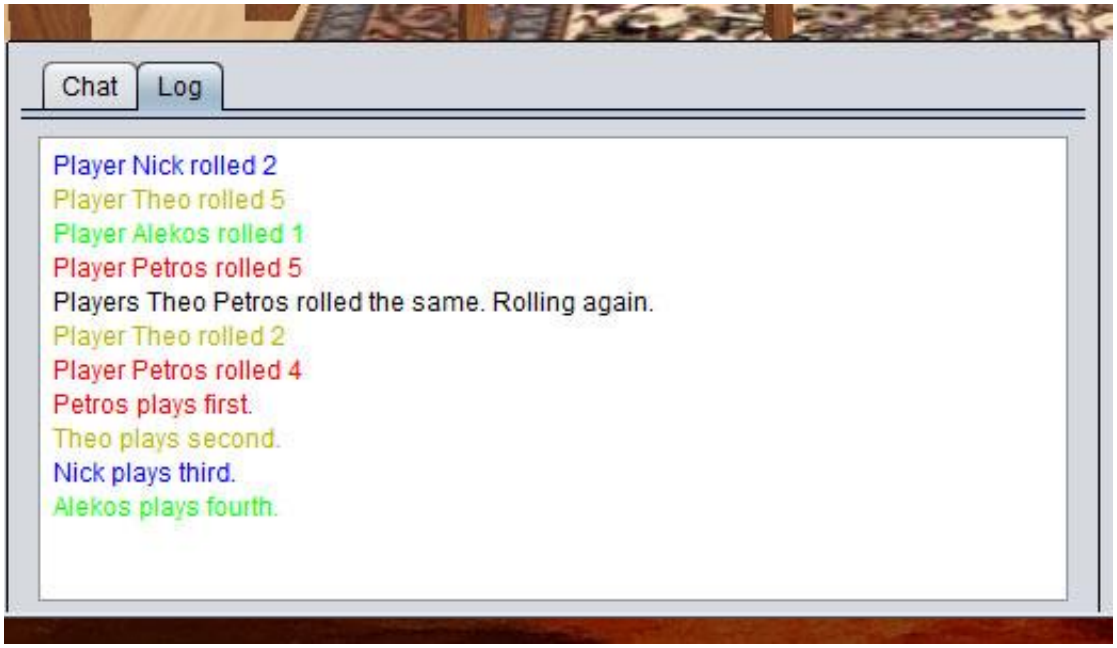


Εικόνα 4.2 : Διάγραμμα GuiPanel

Όπως φαίνεται και στο διάγραμμα «αριθμο» το GuiPanel περιέχει τα:

- ChatNLogPanel
- PlayerStatisticsPanel
- PlayerControlsPanel

4.1.1.1 Η Κλάση ChatNLogPanel:



Εικόνα 4.3 : Απεικόνιση του ChatNLogPanel.

Το Panel αυτό είναι μέλος του Γραφικού Περιβάλλοντος Χρήστη και τοποθετείται στα αριστερά του. Δημιουργεί 2 tabs τα οποία περιέχουν το ένα το ChatPanel και το άλλο το LogPanel.

Το ChatPanel θα χρησιμοποιηθεί για την διαδυκτιακή έκδοση του παιχνιδιού και θα επιτρέπει την συνομιλία μεταξύ των παικτών.

Το LogPanel χρησιμοποιείται για την εκτύπωση πληροφοριών που απορρέουν από την εξέλιξη του παιχνιδιού. Για παράδειγμα το ότι ο παίκτης 1 έχτισε το πρώτο κτίριο του ξενοδοχείου President. Τα μηνύματα εκτυπώνονται στο χρώμα του παίκτη για τον οποίο φέρουν την πληροφορία. Χρησιμοποιήθηκε ένα πεδίο JtextPane με την ικανότητα του να μπορεί να μορφοποιεί το κείμενό του ανάλογα. Το LogPanel «ακούει» την εξέλιξη του παιχνιδιού από τον LogListener και όποτε καταφθάνει ένα μήνυμα το εκτυπώνει με το κατάλληλο χρώμα.

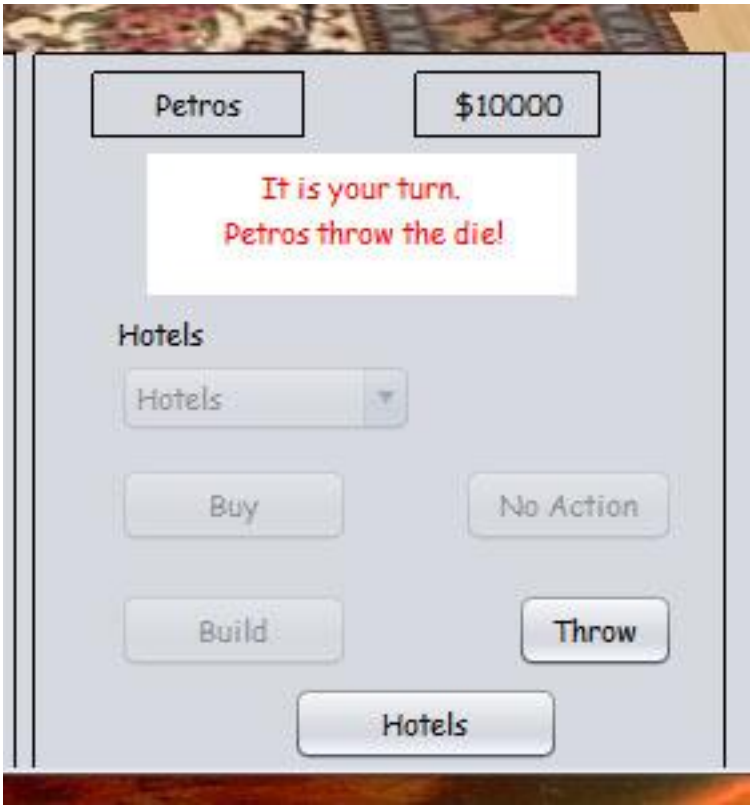
4.1.1.2 PlayerStatisticsPanel:

Εικόνα 4.4 : Απεικόνιση του PlayerStatisticsPanel.

Τοποθετείται στα δεξιά του Γραφικού Περιβάλλοντος Χρήστη. Δημιουργεί τόσα tabs, όσα οι παίκτες – 1 και με τη βοήθεια στοιχισμένων JLabel απεικονίζει πληροφορίες όπως τα χρήματα του κάθε παίκτη, τον αριθμό των ξενοδοχείων που έχει στην κατοχή του, τον αριθμό των κτιρίων που έχει χτίσει καθώς και τον αριθμό των βάσεων που έχει στη διάθεσή του. Συνεπώς, κάθε tab, τα οποία έχουν ξεχωριστό JPanel το καθένα, έχει σαν πεδίο τον Player για τον οποίο εμφανίζει στατιστικά. Όταν το παιχνίδι δεν παίζεται διαδυκτιακά, τα πεδία Player των tabs ανανεώνονται κυκλικά μετά το τέλος κάθε γύρου έτσι ώστε να δεχθούν τον παίχτη του οποίου η σειρά μόλις τελείωσε και να αποβάλλουν τον παίχτη που έχει τώρα σειρα. Αυτό συμβαίνει γιατί δεν υπάρχει η ανάγκη να εκτυπώνονται τα στατιστικά του εκάστοτε ενεργού παίκτη.

Τέλος, το PlayerStisticsPanel «ακούει» μέσω του CurrentPlayerChangedListener εάν κάποιος (και ποιος) παίκτης τελείωσε τη σειρά του. Όταν συμβεί αυτό μέσω της κλήσης της μεθόδου refresh() για κάθε tab ανανεώνονται οι τιμές των JLabel.

4.1.1.3 PlayerControlsPanel:



Εικόνα 4.5 : Απεικόνιση του PlayerControlsPanel.

Τοποθετείτε στη μέση του Γραφικού Περιβάλλοντος Χρήστη. Το πεδίο Player του ανανεώνεται κάθε φορά με τον ενεργό παίκτη και συνεπώς όλες οι λειτουργίες που εκτελούν τα πεδία του,αφορούν αυτόν. Είναι ένα JPanel που δίνει τη δυνατότητα στον χρήστη να αποφασίσει εάν θέλει να αγοράσει ένα οικόπεδο, να χτίσει ένα κτίριο, να αγοράσει μια είσοδο, να ρίξει τα ζάρια ή να μην εκτελέσει καμμία ενέργεια.

Για να εκτελέσει όλες αυτές τις λειτουργιες το PlayerControlsPanel διαθέτει τα εξής Jbutton με τους αντίστοιχους listeners τους :

- ***BuildButton:*** Στέλνει εντολή στον GameController να χτίσει το επόμενο στη σειρά κτίριο του ξενοδοχείου που είναι επιλεγμένο απο το HotelsComboBox component.
- ***ThrowButton:*** Στέλνει εντολή στον GameController να ρίξει το ζάρι με τα νούμερα ή το ζάρι για το χτίσιμο ενός κτιρίου.
- ***HotelsComboBox:*** Η λίστα αυτή δέχεται ως αντικείμενα ξενοδοχεία υποψήφια είτε για αγορά, είτε για χτίσιμο, είτε για την αγορά μιας εισόδου. Ο ενεργός χρήστης επιλέγει από τα διαθέσιμα ξενοδοχεία και στη συνέχεια πατάει το κουμπί που αντιστοιχεί στην επιθυμητή λειτουργία.
- ***BuyButton:*** Στέλνει εντολή στον GameController ότι ο ενεργός παίκτης αγόρασε το οικόπεδο ενός ξενοδοχείου.
- ***NoActionButton:*** Στέλνει εντολή στον GameController ότι ο παίκτης δεν επιθυμεί καμμία ενέργεια.
- ***HotelsButton:*** Εμφανίζει ένα καινούριο JFrame. Το νέο frame περιέχει ένα tab για κάθε ξενοδοχείο και εμφανίζει τις εξής πληροφορίες:
    - Την αξία αγοράς του οικοπέδου
    - Την αξία χτισίματος καθενός από τα κτίρια του
    - Την αξία του χτισίματος των εξωτερικών του χώρων
    - Την αξία αγοράς εισόδου
    - Την αξία διαμονής για 1 έως 6 ημέρες εάν κάποιος παίκτης χρειαστεί να πληρώσει κάποιον άλλον

Επιπλέον το PlayerControlsPanel «ακούει» από τους εξής listeners, τους οποίους και χρησιμοποιεί ο GameController για να ειδοποιεί αυτούς που ακούν:

- ***PlayerListener:*** Ειδοποίηση ότι η κατάσταση ενός Player έχει αλλάξει, πληρώθηκε από κάποιον άλλον παίκτη για παράδειγμα και άλλαξε το ποσό των χρημάτων του. Χρησιμοποιείται κυρίως για να ανανεώνει τα αντίστοιχα JLbel με τα καινούργια στατιστικά του παίκτη.
- ***CurrentPlayerChangedListener:*** Ειδοποίηση ότι ο ενεργός παίκτης έχει αλλάξει και ποιός είναι αυτός. Χρησιμοποιείται όταν το παιχνίδι δεν παίζεται διαδυκτιακά για να ενημερώσει το πεδίο Player του

PayerControlsPanel μαζί με τα αντίστοιχα JLabel. Ουσιαστικά δίνει τον έλεγχο του Panel στον εκάστοτε ενεργό παίκτη.

- ***PlayerMessageListener:*** Το PlayerControlsPanel διαθέτει ένα JTextPane μέσα στο οποίο καταφθάνουν μηνύματα που ειδοποιούν τον ενεργό παίκτη για την επόμενη απόφαση που πρέπει να πάρει. Για παράδειγμα, ότι μπορεί εάν θέλει να αγοράσει κάποιο οικόπεδο. Τα μηνύματα αυτά στέλνονται από τον GameController και συνήθως ακολουθούνται από την ενημέρωση των υποψήφιων ξενοδοχειακών επιλογών (εξηγείτε παρακάτω). Μαζί με τα μηνύματα αποστέλλεται και το αντίστοιχο PlayerAction που πρέπει να κάνει ο ενεργός παίκτης. Για παράδειγμα build. Ανάλογα με το PlayerAction που λαμβάνει ο PlayerControlsPanel ενεργοποιεί και απενεργοποιεί κάποια από τα JButton του. Για παράδειγμα για το Action build, απενεργοποιούνται τα πάντα εκτός από το κουμπί Build και No Action.
- ***CandidateHotelsListener:*** Ειδοποίηση από τον GameController που περιέχει τα ξενοδοχεία που είναι υποψήφια κάθε φορά για κάποια ενέργεια:
    - Για αγορά οικοπέδου
    - Για χτίσιμο του επομένου στη σειρά κτιρίου ή εξωτερικών χώρων
    - Για επιλογή θέσης στην οποία θα τοποθετηθεί μια είσοδος.

  Ο ενεργός παίκτης στη συνέχεια επιλέγει το ξενοδοχείο από τη λίστα και προχωρεί στην κίνηση του.

- ***BaseSelectedListener:*** Ειδοποίηση ότι ο ενεργός παίκτης έχει επιλέξει μια θέση για την τοποθέτηση εισόδου. Έρχεται από την κλάση MousePickBehavior η οποία παρακολουθεί το αριστερό click του ποντικιού του ενεργού παίκτη και επιστρέφει την αντίστοιχη θέση. Οι κλάσεις που υλοποιούν μια συγκεκριμένη συμπεριφορα – Behavior θα αναλυθούν αναλυτικότερα στην ενότητα του GraphicsPanel.

Τέλος, ο PlayerControlsPanel κρατάει τους δικούς του listeners:

- ***PlayerActionListener:*** Στέλνει ειδοποίηση στον GameController ότι ο ενεργός παίκτης έχει πατήσει ένα κουμπί και επομένως έχει

αποφασίσει για την επόμενή του κίνηση. Τα διαθέσιμα PlayerAction είναι:

- Build
- Throw
- Buy
- No Action

Ο GameController δέχεται το PlayerAction και εκτελεί την αντίστοιχη λειτουργία συνεχίζοντας έτσι τη ροή του παιχνιδιού.

- ***DisplayArrowsListener:*** Ειδοποιεί τον GameController να ενημερώσει το GraphicsPanel να ζωγραφίσει βέλη επάνω από τις θέσεις που μπορεί ο ενεργός παίκτης να αγοράσει και να τοποθετήσει μια είσοδο.

4.1.2 Η Κλάση GraphicsPanel

Είναι η κλάση μέσω της οποίας σχεδιάζονται και εκτυπώνονται όλα τα 3D αντικείμενα στην οθόνη. Ουσιαστικά είναι ένα JPanel το οποίο περιέχει το SimpleUniverse το οποίο με τη σειρά του περιέχει το Canvas3D (βλέπε ενότητα Java3D).

Με την κλήση του constructor της GraphicsPanel εκτελούνται οι παρακάτω λειτουργίες:

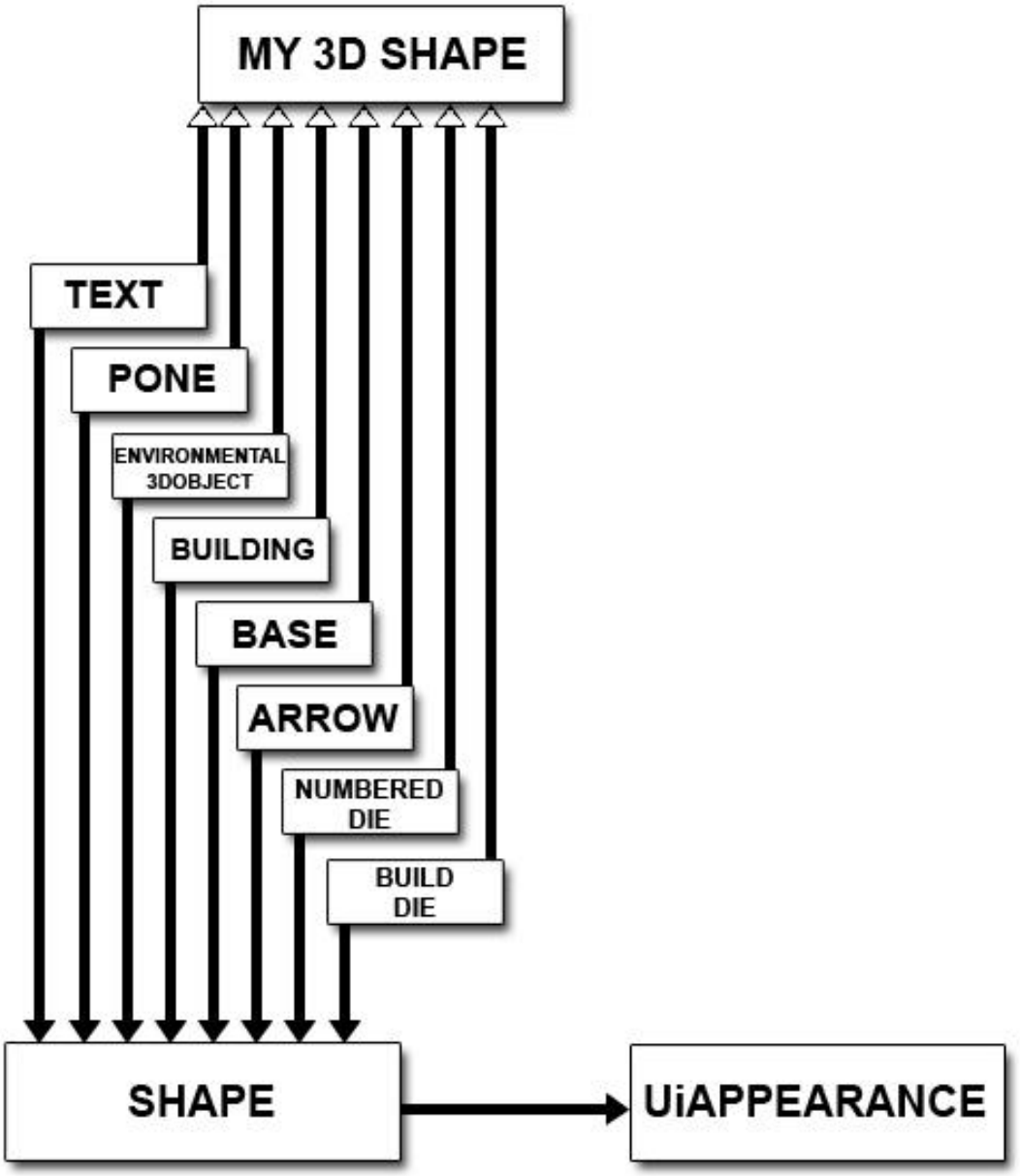
- Δήλωση των διαστάσεων του panel σε 1024x500. Αυτές οι τιμές έχουν σαν αποτέλεσμα ολόκληρη η εφαρμογή να τρέχει σε ανάλυση 1024x768. Τα υπόλοιπα 268 pixels του ύψους του παραθύρου της εφαρμογής κατανέμονται στο Γραφικό Περιβάλλον Χρήστη.
- Αρχικοποίηση του αντικειμένου Canvas3D canvas3d. Η κλάση Canvas3D παρέχει έναν καμβά ζωγραφικής για απεικόνιση 3D αντικειμένων.
- Αρχικοποίηση του αντικειμένου SimpleUniverse simpleUniverse το οποίο και περιέχει τον 3D καμβά. Η κλάση αυτή δημιουργεί όλα τα απαραίτητα αντικείμενα από την πλευρά του “view” ενός γράφου μιας

σκηνής. Ποιο συγκεκριμένα, δημιουργεί μια απλή ViewingPlatform και ένα αντικείμενο View με τις default τιμές τους. Στα πλαίσια αυτού του προγράμματος οι τιμές αυτές δεν χρειάστηκε να τροποποιηθούν.

- Κλήση της μεθόδου initUserPosition(). Η μέθοδος αυτή είναι υπεύθυνη για την τοποθέτηση του αντικειμένου View σε μια αρχική θέση. Αυτό πρακτικά σημαίνει από ποιό σημείο και γωνία κοιτάει ο χρήστης τον 3D κόσμο.
- Κλήση της μεθόδου init(). Με την κλήση της μεθόδου αυτής ξεκινάει η διαδικασία σχεδίασης όλων των 3D αντικειμένων που θα είναι ορατά από το ξεκίνημα της εφαρμογής. Η διαδικασία αυτή αναλύεται λεπτομερώς στις επόμενες ενότητες.

4.1.2.1 ΤΟ ΠΑΚΕΤΟ GRAPHICS.ENTITIES

Στην ενότητα αυτή θα περιγραφεί η διαδικασία σχεδίασης όλων των 3D αντικειμένων με στόχο την απεικόνισή τους στην οθόνη. Η διαδικασία αυτή μπορεί να γίνει με πολλούς τρόπους. Η περιγραφή που ακολουθεί όμως αφορά αυτή την εφαρμογή.



Εικόνα 4.6 : Διάγραμμα My3DShape.

Όπως φαίνεται και στο διάγραμμα «αριθμό» ,η ιεραρχία όλων των 3D αντικειμένων ξεκινάει με την κλάση My3DShape την οποία και κληρονομούν όλες οι κλάσεις του πακέτου graphics.entities.

Κάθε κλάση που αντιπροσωπεύει ένα 3D αντικείμενο και περιλαμβάνει τα εξής πεδία τα οποία και χρησιμοποιούνται για τις διάφορες λειτουργίες

εμφάνισης, τοποθέτησης, περιστροφής και κίνησης του αντικειμένου πάνω στη σκηνή:

- ***angleXPosition:*** Η γωνία περιστροφής πάνω στον άξονα των x τη στιγμή που το αντικείμενο πρωτοεμφανίζεται στην αρχική του θέση.
- ***angleYPosition:*** Η γωνία περιστροφής πάνω στον άξονα των y τη στιγμή που το αντικείμενο πρωτοεμφανίζεται στην αρχική του θέση.
- ***angleZPosition:*** Η γωνία περιστροφής πάνω στον άξονα των z τη στιγμή που το αντικείμενο πρωτοεμφανίζεται στην αρχική του θέση.
- ***scale:*** Η κλίμακα του αντικειμένου.
- ***shapes:*** Τα αντικείμενα shape από τα οποία αποτελείται το κτίριο. Η μέθοδος assembleShapes() έχει το ρόλο να συναρμολογήσει όλα τα shapes φτιάχνοντας το κτήριο.
- ***xPos:*** Η συντεταγμένη x της αρχικής θέσης του αντικειμένου.
- ***yPos:*** Η συντεταγμένη x της αρχικής θέσης του αντικειμένου.
- ***zPos:*** Η συντεταγμένη x της αρχικής θέσης του αντικειμένου.

4.1.2.2 Η ΚΛΑΣΗ BUILDING

Αντιπροσωπεύει ένα 3D αντικείμενο με το σχήμα ενός ολοκληρωμένου κτιρίου. Δημιουργείτε κατά την εκκίνηση της εφαρμογής και εμφανίζεται στη σκηνή από την αρχή του παιχνιδιού.

Η αρχική του θέση είναι σε κάποιο συγκεκριμένο σημείο επάνω σε ένα τραπέζι, έξω από το ταμπλό του παιχνιδιού. Τελική του θέση είναι η θέση χτισίματος πάνω στο ταμπλό στο αντίστοιχο ξενοδοχείο.

Ένα αντικείμενο building αναγνωρίζεται από τα πεδία του:

- name: Το όνομα του ξενοδοχείου στο οποίο ανήκει
- ordinal: Η σειρά του κτιρίου. Για παράδειγμα, το ξενοδοχείο Waikiki έχει πέντε κτίρια, το πρώτο κτίριο έχει τιμή ordinal = 0, το δεύτερο 1 κ.ο.κ.

Εκτός από τα κοινά πεδία που έχουν όλες οι κλάσεις του πακέτου graphics.entities, η Building διαθέτει και τις παρακάτω:

- ***buildAngleXPosition:*** Η γωνία περιστροφής πάνω στον άξονα των x τη στιγμή που το αντικείμενο τοποθετείται στην θέση χτισίματος, επάνω στο ταμπλό.
- ***buildAngleYPosition:*** Η γωνία περιστροφής πάνω στον άξονα των y τη στιγμή που το αντικείμενο τοποθετείται στην θέση χτισίματος, επάνω στο ταμπλό.
- ***buildAngleZPosition:*** Η γωνία περιστροφής πάνω στον άξονα των z τη στιγμή που το αντικείμενο τοποθετείται στην θέση χτισίματος, επάνω στο ταμπλό.
- ***buildingBehavior:*** Η «συμπεριφορά» του αντικειμένου (behavior, βλέπε ενότητα Behaviors).
- ***builtXPos:*** Η συντεταγμένη x της θέσης χτισίματος.
- ***builtYPos:*** Η συντεταγμένη y της θέσης χτισίματος.
- ***builtZPos:*** Η συντεταγμένη z της θέσης χτισίματος.

Όλα τα αντικείμενα building δημιουργούνται από την κλάση Shapes3DEngine και αποθηκεύονται στην μνήμη σε ένα αντικείμενο τύπου Buildings το οποίο κληρονομεί την κλάση Vector.

Όταν στη συνέχεια δημιουργούνται τα αντικείμενα Hotel σε κάθε ένα από αυτά ανατίθενται τα δικά του κτίρια.

4.1.2.3 Η ΚΛΑΣΗ ENVIRONMENTAL3DOBJECT

Αντιπροσωπεύει ένα 3D αντικείμενο, μέλος του γύρω περιβάλλοντος που περικλύει το ταμπλό και τα κτίρια του παιχνιδιού. Ένα τέτοιο αντικείμενο είναι το τραπέζι πάνω στο οποίο τοποθετήθηκε ολόκληρο το παιχνίδι.

Έχει την δυνατότητα φόρτωσης ενός 3D μοντέλου το οποίο έχει αναπτυχθεί σε ένα τρίτο πρόγραμμα όπως το 3D Studio Max (βλέπε ενότητα 3D Studio Max). Την διαδικασία φόρτωσης ενός τέτοιου σχήματος αναλαμβάνει η κλάση Model3Dloader.

Ένας loader είναι ένα απαραίτητο εργαλείο για τη γρήγορη εισαγωγή 3D αντικειμένων όπως άνθρωποι, αντικείμενα και σκηνικά.

Γενικά η δουλειά ενός Loader είναι να διαβάσει κάποιο 3D model αρχείο, το οποίο έχει κατασκευαστεί από κάποιο τρίτο λογισμικό όπως το 3D Studio Max και να δημιουργήσει το αντίστοιχο σχήμα με βάση τους μηχανισμούς της java3D. Τέτοια αρχεία περιέχουν γεωμετρικές πληροφορίες για την μορφή των σχημάτων που περιέχουν καθώς και πληροφορίες για την εμφάνιση τους, χρώματα ή textures.

Η κλάση δημιουργεί ένα αντικείμενο τύπου Loader3DS και φορτώνει κάποιο model. Η θέση του αρχείου δίνεται με μορφή URL. Στην συνέχεια δημιουργείται ένα καινούριο BranchGroup, προστίθεται το καινούριο αντικείμενο και στη συνέχεια «προσκολλάται» στο κεντρικό BranchGroup.

4.1.2.4 Η ΚΛΑΣΗ PONE

Αντιπροσωπεύει ένα 3D αντικείμενο με σχήμα πιονιού. Όλα τα αντικείμενα Pone δημιουργούνται από την κλάση Shapes3DEngine και αποθηκεύονται στην μνήμη σε ένα αντικείμενο τύπου Pones το οποίο κληρονομεί την κλάση Vector.

Όταν στη συνέχεια δημιουργούνται τα αντικείμενα Player σε κάθε ένα από αυτά ανατίθεται το αντίστοιχο πιόνι ανάλογα με το χρώμα που διάλεξε ο παίκτης.

4.1.2.5 Η ΚΛΑΣΗ ARROW

Αντιπροσωπεύει ένα 3D αντικείμενο σε σχήμα βέλους. Χρησιμοποιείται για να δείξει μια θέση της διαδρομής του ταμπλό. Ο παίκτης στη συνέχεια μπορεί να διαλέξει αυτή τη θέση κάνοντας κλικ επάνω στο βελάκι. Στη συνέχεια μια είσοδος τοποθετείται σε εκείνη τη θέση.

Τα αντικείμενα αυτά δεν δημιουργούνται κατά την εκκίνηση της εφαρμογής αλλά όταν χρειάζεται. Στα αντικείμενα αυτά ανατίθονται δυο είδη συμπεριφοράς (behavior):

- MouseOverBehavior: Προσδίδει ένα mouse-over εφέ στο αντικείμενο.
- MousePickBehavior: Επιτρέπει την δυνατότητα επιλογής του αντικειμένου κάνοντας κλικ επάνω του.
- ArrowBehavior: Περιστρέφει το αντικείμενο γύρω από τον άξονα των y.

Οι κλάσεις Behavior θα αναλυθούν σε επόμενη ενότητα.

4.1.2.6 Η ΚΛΑΣΗ BASE

Αντιπροσωπεύει ένα 3D αντικείμενο σε σχήμα εισόδου.

Χρησιμοποιείται για να δηλώσει ότι κάποιος παίκτης τοποθέτησε μια είσοδο σε κάποια θέση. Δημιουργούνται και τοποθετούνται όταν ο παίκτης επιλέξει την επιθυμητή θέση.

4.1.2.7 Η ΚΛΑΣΗ TEXT

Αντιπροσωπεύει ένα 3D αντικείμενο σε σχήμα μιας λέξης.

Όλα τα αντικείμενα text δημιουργούνται κατά την εκκίνηση της εφαρμογής και κάθε ένα από αυτά απεικονίζει το όνομα ενός ξενοδοχείου.

Τοποθετούνται ακριβώς επάνω από τα οικόπεδα των ξενοδοχείων στο ταμπλό του παιχνιδιού. Στα αντικείμενα text ανατίθεται η ArrowsBehavior.

Επιτρέπει στο αντικείμενο να περιστρέφεται γύρω από τον άξονα των y καθώς χρωματίζεται με το χρώμα του παίκτη που το έχει αγοράσει.

4.1.2.8 Η ΚΛΑΣΗ MY3DSHAPE

Η My3Dshape παρέχει όλα τα απαραίτητα αντικείμενα που επιτρέπουν σε ένα 3D αντικείμενο να γίνει μέλος (εμφανιστεί) του καμβά καθώς και τους μηχανισμούς κίνησης και περιστροφής τους. Τα απαραίτητα αυτά αντικείμενα είναι τα εξής και αρχικοποιούνται με την κλήση του μοναδικού constructor της κλάσης:

- ***TransformGroup mainTG:*** Το TransformGroup που περιέχει όλα τα 3D σχήματα, τα οποία με τη σειρά τους σχηματίζουν ένα ολοκληρωμένο 3D σχήμα.
- ***TransformGroup scaleTG:*** Το TransformGroup που είναι υπεύθυνο για την κλίμακα σχεδίασης του 3D αντικειμένου. Περιέχει το mainTG.
- ***TransformGroup rotationTG:*** Το TransformGroup που είναι υπεύθυνο για την περιστροφή του 3D αντικειμένου γύρω από τον άξονα x, y και z. Περιέχει το scaleTG.
- ***TransformGroup moveTG:*** Το TransformGroup που είναι υπεύθυνο για την μετακίνηση του 3D αντικειμένου πάνω στον άξονα x, y ή z. Περιέχει το rotationTG.
- ***Transform3D xAxis:*** Παρέχει περιστροφή πάνω στον άξονα των x.
- ***Transform3D yAxis:*** Παρέχει περιστροφή πάνω στον άξονα των y.
- ***Transform3D zAxis:*** Παρέχει περιστροφή πάνω στον άξονα των z.
- ***Transform3D posT3d:*** Παρέχει κίνηση πάνω στον άξονα των x, y ή z.

Όλα τα nodes στην Java3D έχουν μια σχέση γονιού-παιδιού. Έτσι το scene graph του SimpleUniverse αυτής της εφαρμογής έχει ένα κεντρικό BranchGroup node και πάνω σε αυτό προστίθενται τα TransformGroup nodes που περιέχουν ένα σχήμα το καθένα (βλέπε ενότητα Java3D διάγραμμα «αριθμο»).

4.1.2.9 Η ΚΛΑΣΗ SHAPE

Γενικά, για τον σχεδιασμό 3D αντικειμένων, χρειάζεται να περιγράψουμε με κάποιο τρόπο τις γεωμετρικές ιδιότητες ενός σχήματος. Εναλλακτικά, είναι εφικτό να χωριστεί ένα μεγάλο σχήμα (όπως ένα κτίριο) σε επιμέρους μικρότερα, τοποθετημένα με το κατάλληλο τρόπο.

Αυτή είναι και η προσέγγιση που χρησιμοποιήθηκε σε αυτήν την εφαρμογή για τον σχεδιασμό των περισσοτέρων αντικειμένων.

Όλες οι γεωμετρικές πληροφορίες ενός σχήματος μαζί με άλλα χαρακτηριστικά που περιγράφουν την θέση τους μέσα στον 3D κόσμο ή την εμφάνιση τους είναι αποθηκευμένες σε ένα xml αρχείο με όνομα «Shapes3D.ini». Το πώς αποθηκεύονται όλες αυτές οι πληροφορίες στο xml αρχείο θα αναλυθεί σε επόμενη ενότητα.

Ένα αντικείμενο της κλάσης Shape αντιπροσωπεύει ένα 3D σχήμα. Υπάρχουν πολλοί τρόποι να δημιουργήσει κανείς ένα σχήμα στην java3D και αυτό εξαρτάται φυσικά από το είδος του σχήματος που πρέπει να κατασκευαστεί. Έτσι, υπάρχουν κάποια primitive αντικείμενα τα οποία χρησιμοποιήθηκαν παρέχουν έτοιμα σχήματα όπως η κλάση Box, Cylinder, Text3D ενώ για τη δημιουργία πιο πολύπλοκων σχημάτων χρησιμοποιήθηκε η QuadArray, LatheShape3D, CustomLatheShape3D και LatheQuad. Όλα τα παραπάνω είναι τα είδη σχημάτων που υποστηρίζονται από την εφαρμογή.

Τα πεδία ενός αντικειμένου Shape κρατούν γεωμετρικές πληροφορίες για το σχήμα καθώς και οδηγίες για την εμφάνιση τους και την θέση τους στον κόσμο και είναι τα εξής:

- String type
- float width
- float height
- float length
- float xPos

- float yPos
- float zPos
- float[] pointFparameters
- double[] latheShapeCoords
- double angleXPosition
- double angleYPosition
- double angleZPosition
- String texture
- String textureSideLeft
- String textureSideRight
- String textureSideFront
- String textureSideBack
- String text
- int fontSize
- Color3f lightColor
- Color3f darkColor
- TransformGroup moveTG
- TransformGroup scaleTG
- TransformGroup rotationTG
- Transform3D posT3d
- Transform3D xAxis
- Transform3D yAxis
- Transform3D zAxis

Η χρησιμότητα τους ανάλογα με το είδος του σχήματος που δημιουργείται θα εξηγηθεί στη συνέχεια.

Πριν εξηγηθεί η διαδικασία με την οποία δημιουργούνται τα επιμέρους σχήματα είναι σημαντικό να εξετασθεί ο τρόπος με τον οποίο δίνεται κάποιο χρώμα ή εικόνα (το λεγόμενο appearance ενός σχήματος).

4.1.2.10 Η ΚΛΑΣΗ UIAPPEARANCE

Κληρονομεί την κλάση Appearance. Ένα αντικείμενο appearance προσδιορίζει όλες τις καταστάσεις απεικόνισης ενός 3D αντικειμένου στον 3D κόσμο.

Για τις ανάγκες αυτής της εφαρμογής χρειάστηκε η δημιουργία ενός αντικειμένου material για το Appearance ενός σχήματος και, εάν χρειάζεται, η φόρτωση μιας εικόνας επάνω στο σχήμα με τη βοήθεια ενός Texture αντικειμένου.

Το material χρησιμοποιείται για να δημιουργήσει μια συγκεκριμένη εμφάνιση σ'ένα αντικείμενο που μπορεί και αντανακλά το φως που πέφτει επάνω του. Η κλάση UiAppearance μπορεί να δημιουργήσει δυο ειδών material:

- setColoredMaterial(): Καλείτε όταν δεν χρειάζεται να φορτώσουμε κάποια εικόνα στο αντικείμενό μας αλλά απλά να το χρωματίσουμε.
- setTextureMaterial(): Καλείτε όταν χρειάζεται να φορτώσουμε κάποια εικόνα στο αντικείμενό μας. Σε αυτήν την περίπτωση, η ένταση του φωτός που ανακλάται πάνω στο αντικείμενο είναι μικρή. Ένα Texture αντικείμενο δημιουργείται με την κλήση της μεθόδου loadTexture() και φορτώνεται η επιθυμητή εικόνα.

4.1.2.11 Η ΜΕΘΟΔΟΣ MAKEBOX

Επιστρέφει ένα αντικείμενο σε σχήμα κουτιού.

Οι διαστάσεις του ορίζονται από τα πεδία width, height και length.

Η θέση του στον χώρο ορίζεται από τα πεδία xPos, yPos και zPos.

Εάν πρέπει να περιστραφεί γύρω από κάποιο αξόνα χρησιμοποιούνται τα πεδία angleXPosition, angleYPosition και angleZPosition.

Τέλος για την εμφάνισή του, εάν πρέπει να πάρει κάποιο συγκεκριμένο χρώμα,τότε χρησιμοποιούνται τα πεδία darkColor και lightColor. Αλλιώς φορτώνεται μια εικόνα στο αντικείμενο.Το πεδίο texture δηλώνει ποιά θα είναι αυτή. Επιπλέον, εάν χρειάζεται οι διάφορες πλευρές του κουτιού να μην έχουν

την ίδια εικόνα, γίνεται η χρήση των πεδίων textureLeft, textureRight, textureFront και textureBack.

4.1.2.12 Η ΜΕΘΟΔΟΣ MAKECYLINDER

Περιστρέφει ένα αντικείμενο σε σχήμα κυλίνδρου.

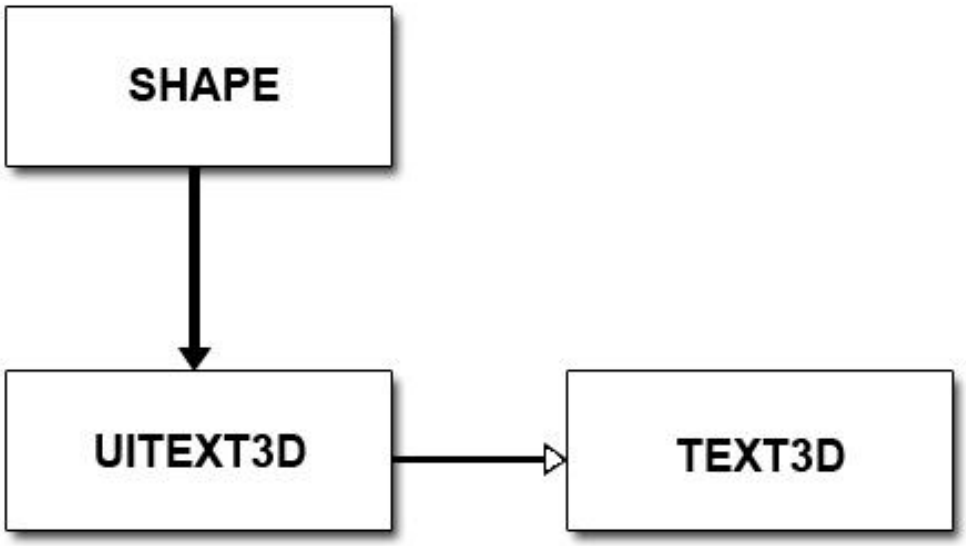
Οι διαστάσεις του ορίζονται από τα πεδία width και height τα οποία προσδιορίζουν την ακτίνα και το ύψος του κυλίνδρου αντίστοιχα.

Η θέση του στον χώρο ορίζεται από τα πεδία xPos, yPos και zPos.

Εάν πρέπει να περιστραφεί γύρω από κάποιο αξόνα χρησιμοποιούνται τα πεδία angleXPosition, angleYPosition και angleZPosition.

Τέλος για την εμφάνιση του, εάν πρέπει να πάρει κάποιο συγκεκριμένο χρώμα τότε χρησιμοποιούνται τα πεδία darkColor και lightColor. Αλλιώς φορτώνεται μια εικόνα στο αντικείμενο, το πεδίο texture δηλώνει ποια θα είναι αυτή.

4.1.2.13 Η ΜΕΘΟΔΟΣ MAKE3DTEXT



Εικόνα 4.7 : Διάγραμμα UiText3D.

Παίρνει ως παράμετρο μια λέξη ή κείμενο μέσω του πεδίου text και δημιουργεί ένα 3D σχήμα που απεικονίζει την λέξη ή το κείμενο αυτό. Το σχήμα μπορεί να έχει εμφάνιση η οποία,όπως και στα δυο προηγούμενα σχήματα,μπορεί να είναι είτε ένα απλό χρώμα είτε μια εικόνα.

Η θέση του στον χώρο ορίζεται από τα πεδία xPos, yPos και zPos.

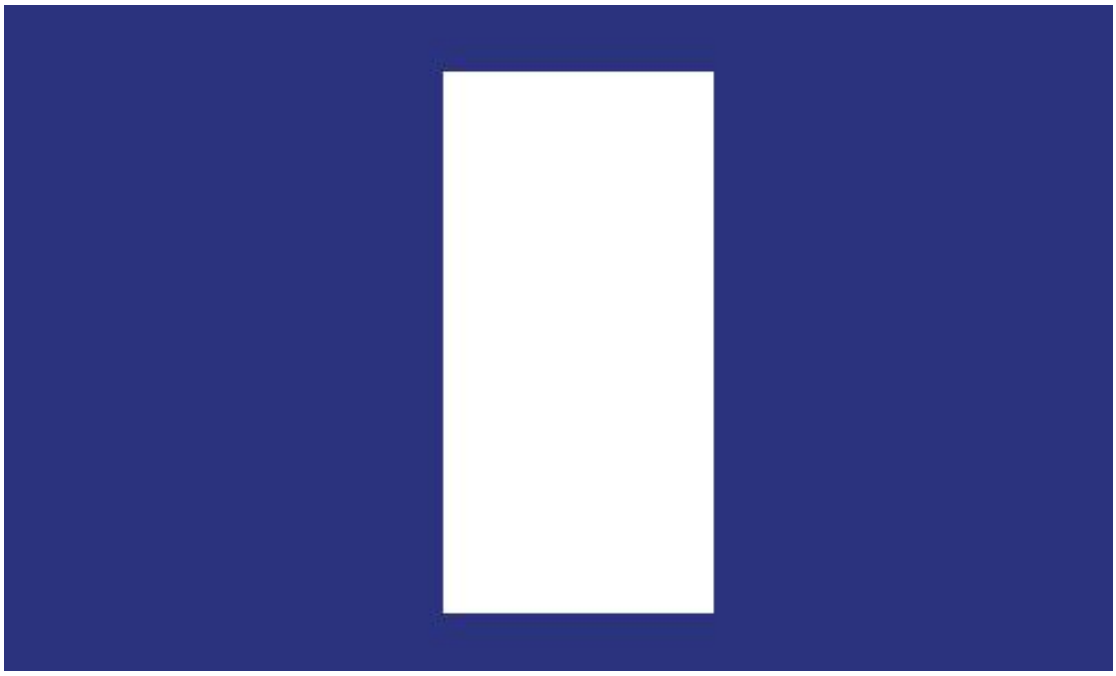
Εάν πρέπει να περιστραφεί γύρω από κάποιο αξόνα χρησιμοποιούνται τα πεδία angleXPosition, angleYPosition και angleZPosition.

Όπως φαίνεται και στο διάγραμμα «αριθμό»,η μέθοδος δημιουργεί ένα αντικείμενο τύπου UiText3D το οποίο κληρονομεί την κλάση Text3D. Η μέθοδος createCustomText() προσδίδει μια συγκεκριμένη γραμματοσειρά καθώς και κεντρική στοίχιση. Για τις ανάγκες τις εφαρμογής ορίζεται μόνο ένας τύπος γραμματοσειράς, η Comic Sans MS, για να διατηρήσει το ίδιο στυλ σε όλο το εύρος του παιχνιδιού.

4.1.2.14 Η ΜΕΘΟΔΟΣ MAKEQUAD

Δημιουργεί ένα αντικείμενο τύπου QuadArray με τις κατάλληλες παραμέτρους για να ετιτευχθεί το κατάλληλο σχήμα.

Το αντικείμενο QuadArray ζωγραφίζει έναν πίνακα από σημεία ως ένα ξεχωριστό τετράπλευρο. Έτσι, για την περιγραφή ενός σχήματος quad,χρησιμοποιείται το πεδίο pointFparameters το οποίο περιέχει τις συντεταγμένες τεσσάρων σημείων τα οποία και σχηματίζουν το επιθυμητό τετράπλευρο.

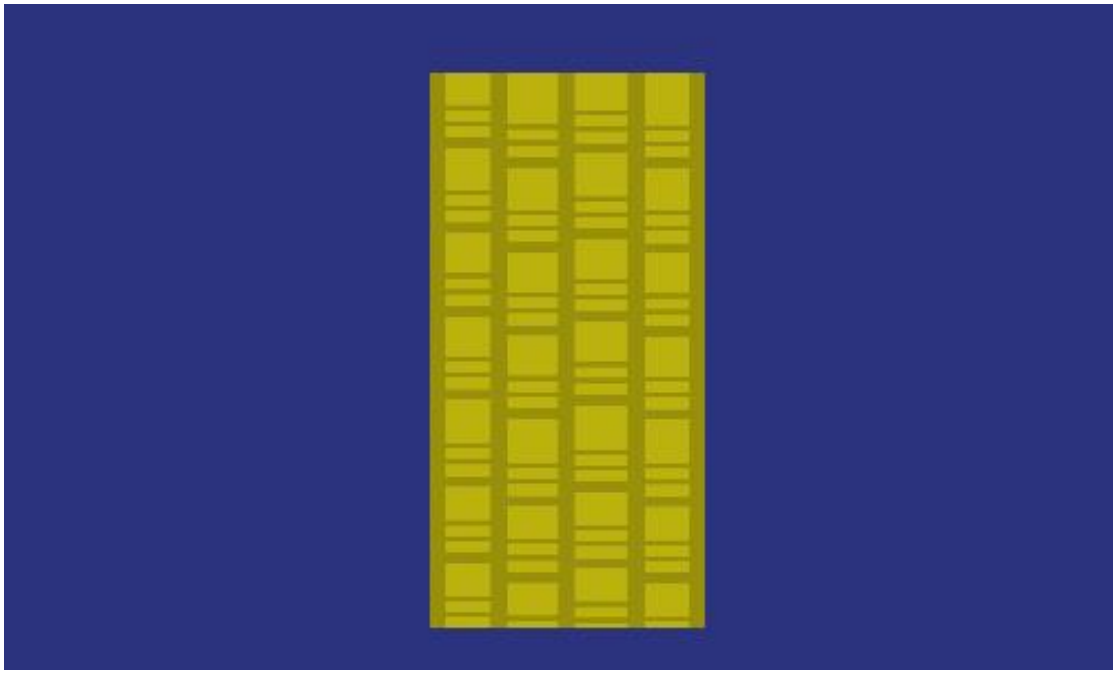


Εικόνα 4.8 : Απεικόνιση ενός Quad.

Με την ίδια λογική, όπως και στα προηγούμενα σχήματα του κυλίνδρου και του κουτιού μπορούμε να αποδώσουμε μια εμφάνιση στο τετράπλευρο, είτε κάποιο συγκεκριμένο χρώμα, είτε κάποια εικόνα (texture).

Εάν χρειάζεται να δοθεί κάποιο συγκεκριμένο χρώμα στο σχήμα τότε τα πράγματα είναι σχετικά απλά, δημιουργείτε το αντικείμενο UiAppearance με ένα ColoredMaterial και χρώμα αυτό που ορίζουν τα πεδία darkColor και lightColor.

Εάν όμως χρειαστεί η χρήση κάποιου texture,τότε τα πράγματα γίνονται πιο περίπλοκα. Καθώς το αντικείμενο QuadArray δεν είναι ένα ολοκληρωμένο σχήμα πρέπει να ορίσουμε επακριβώς πώς θέλουμε να εμφανιστεί η εικόνα μας πάνω στο τετράπλευρο.



Εικόνα 4.9 : Απεικόνιση ενός Quad με Texture.

Αυτές οι οδηγίες δίνονται μέσω του αντικειμένου TexCoord3f. Ορίζουμε τα τέσσερα σημεία που είναι και οι γωνίες της εικόνας. Η σειρά που ορίζονται αυτά τα σημεία παίζει σημαντικό ρόλο καθώς μπορεί το αποτέλεσμα να είναι να εμφανιστεί η εικόνα ανάποδα ή πλάγια. Στη συνέχεια περνάμε αυτά τα σημεία στο τετράπλευρο μας.

Τέλος, δημιουργείται ένα Shape3D το οποίο περιέχει τις γεωμετρικές πληροφορίες του τετραπλεύρου και την αντίστοιχη εμφάνιση.

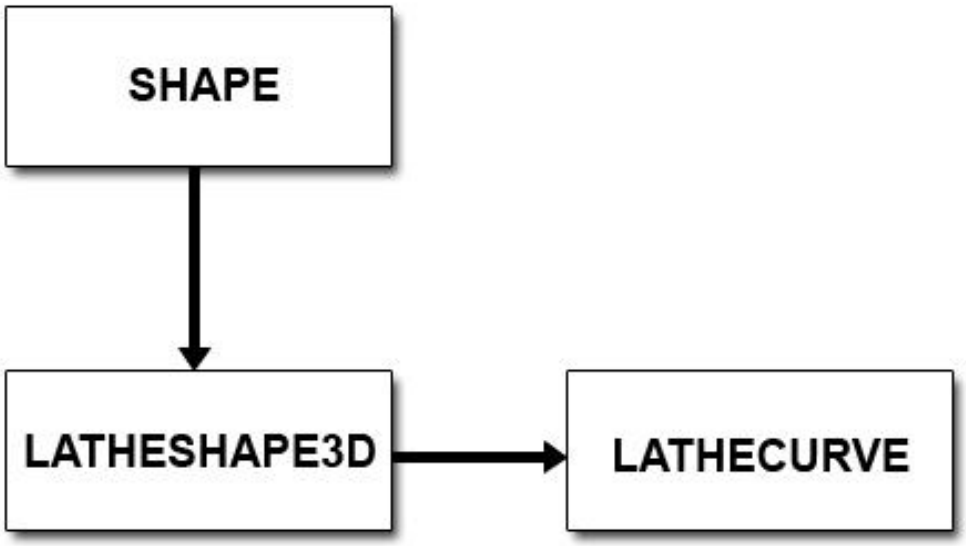
Ένα αντικείμενο Shape3D προσδιορίζει όλα τα γεωμετρικά σχήματα. Μπορεί να πάρει πληροφορίες για ένα ή περισσότερα γεωμετρικά σχήματα και μια εμφάνιση.

4.1.2.15 Η ΜΕΘΟΔΟΣ MAKEQUADARRAY

Εκτελεί παρόμοιες λειτουργίες με την μέθοδο makeQuad,με την μόνη διαφορά ότι επιστρέφει ένα QuadArray, τις γεωμετρικές ιδιότητες ενός τετραπλεύρου και όχι ένα ολοκληρωμένο σχήμα.

Χρησιμεύει στην περίπτωση που πρέπει να αποδώσουμε συγκεκριμένες ειδικές ιδιότητες σε ένα σχήμα,όπως για παράδειγμα για το βελάκι τοποθέτησης μιας εισόδου. Η διαδικασία αυτή θα αναλυθεί στην ενότητα :Η ΚΛΑΣΗ ARROW.

4.1.2.16 Η ΜΕΘΟΔΟΣ MAKELATHESHAPE3D



Εικόνα 4.10 : Διάγραμμα LatheShape3D.

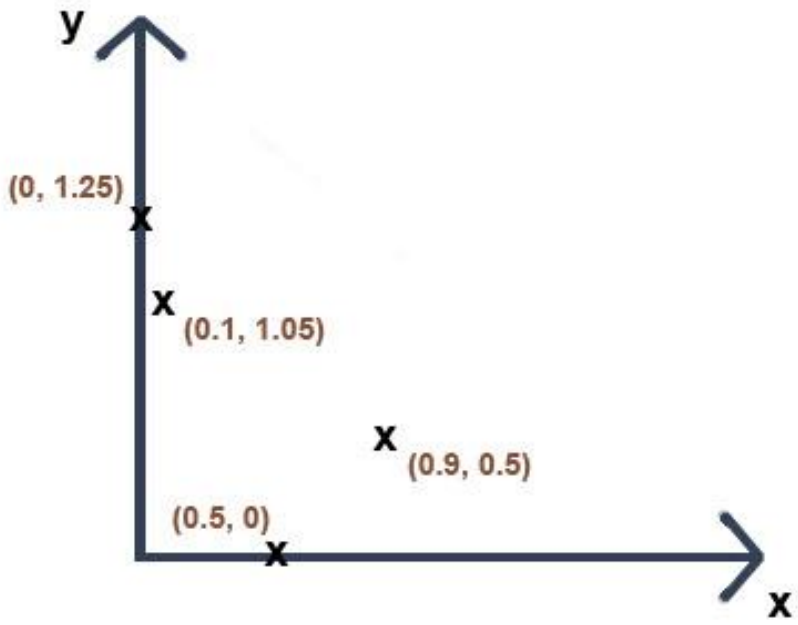
Δημιουργεί ένα κυκλικό καμπυλωτό σχήμα το οποίο προσδιορίζεται από ζευγάρια συντεταγμένων x και y.

Όπως φαίνεται και στο διάγραμμα «αριθμο» η μέθοδος δημιουργεί ένα αντικείμενο τύπου LatheShape3D και περνάει τις συντεταγμένες x και y. Αυτό με τη σειρά του δημιουργεί ένα LatheCurve.

Κύρια δουλειά του LatheCurve είναι να δημιουργεί μια καμπύλη γραμμή τύπου Hermite χρησιμοποιώντας τις συντεταγμένες x και y που δόθηκαν. Τα μαθηματικά που χρησιμοποιούνται για τον σχεδιασμό της καμπύλης είναι πολύ κοινά στον κόσμο των γραφικών και δεν θα αναλυθούν σε αυτό το σύγγραμα.

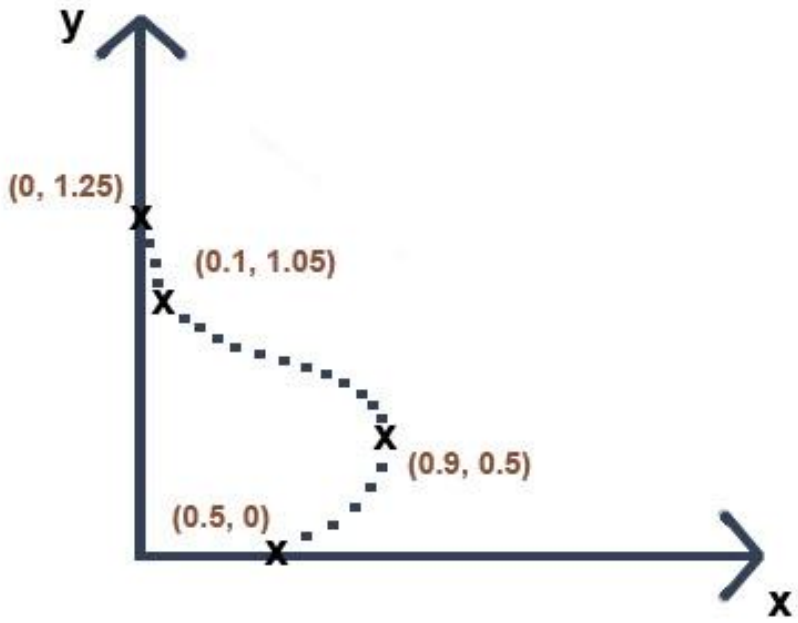
Στη συνέχεια παρουσιάζεται ένα παράδειγμα κατασκευής ενός σχήματος LatheShape3D, η σκεπή των κτιρίων του ξενοδοχείου Taj Mahal.

Οι συντεταγμένες επιλέγονται με βάση το επιθυμητό αποτέλεσμα. Η παρακάτω εικόνα παρουσιάζει τα ζευγάρια των συντεταγμένων που χρησιμοποιήθηκαν τα οποία και ανατίθενται στο πεδίο LatheShapeCoords.



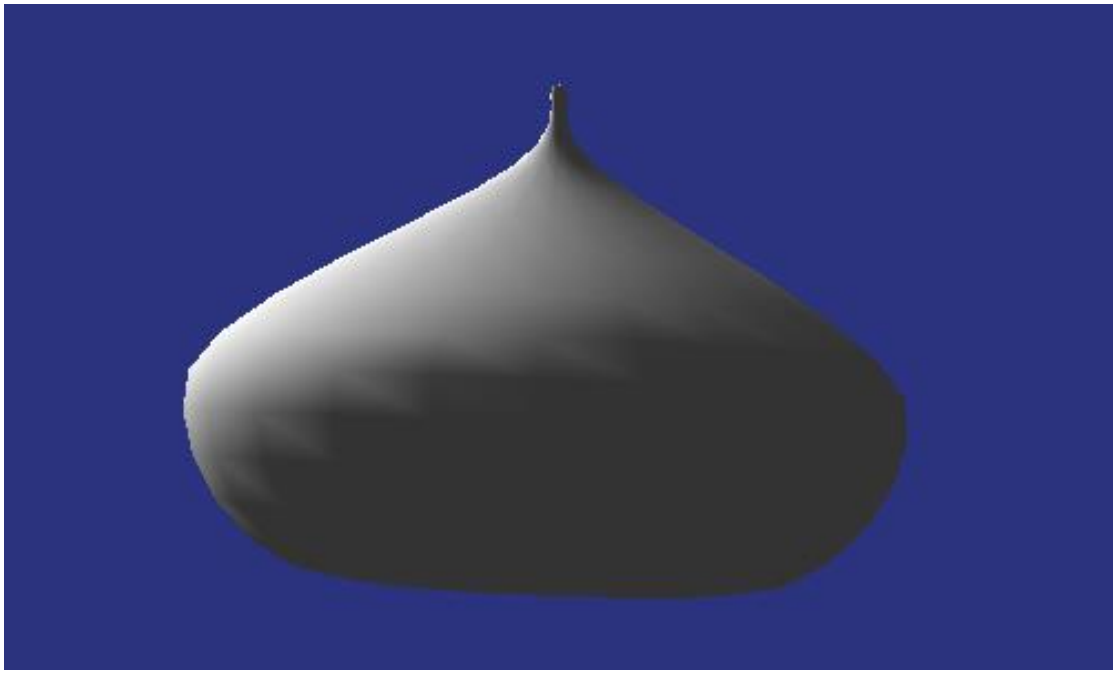
Εικόνα 4.11 : Σημεία σχεδιασμού μιας καμπύλης Hermite.

Στη συνέχεια δημιουργούνται τα απαραίτητα σημεία που σχηματίζουν την πραγματική καμπύλη. Τον ρόλο της δημιουργίας τους αναλαμβάνει το LatheCurve. Το αποτέλεσμα για το παράδειγμα μας δίνεται στην παρακάτω εικόνα.



Εικόνα 4.12 : Πρόσθετα σημεία σχεδιασμού της καμπύλης Hermite.

Για να δημιουργηθεί το τελικό σχήμα όμως μια καμπύλη δεν είναι αρκετή. Γι’ αυτό το λόγο δημιουργούνται τετράδες σημείων, οι οποιες με τη σειρά τους χρησιμοποιούνται για την κατασκευή quadArray το ένα δίπλα στο άλλο σε κυκλικό σχήμα φτιάχνοντας έτσι την επιθυμητή επιφάνεια. Το τελικό σχήμα παρουσιάζεται στην παρακάτω εικόνα.



Εικόνα 4.13 : Η οροφή των κτιρίων του Taj Mahal.

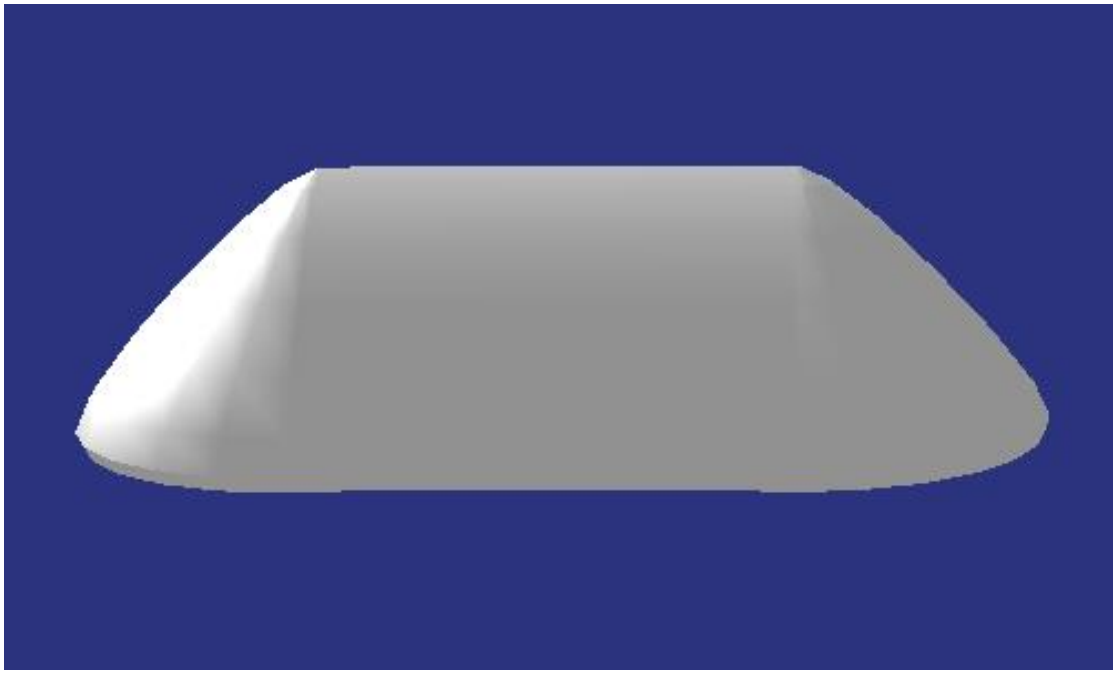
Όπως και στα υπόλοιπα σχήματα, η θέση στην οποία θα τοποθετηθούν στον τρισδιάστατο χώρο ορίζεται από τα πεδία xPos, yPos και zPos, καθώς και η περιστροφή τους στους τρεις άξονες από τα angleXPosition, angleYPosition και angleZPosition.

Το ίδιο ισχύει και για την εμφάνιση τους.

4.1.2.17 Η ΜΕΘΟΔΟΣ MAKECUSTOMLATHESHAPE3D

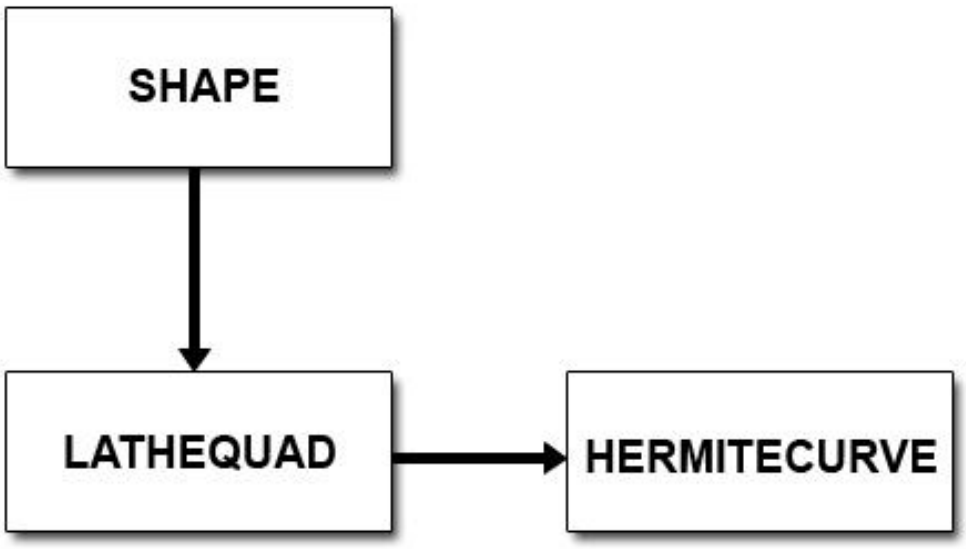
Κατασκευάζει ένα καμπυλωτό σχήμα παρόμοιο του σχήματος που κατασκευάζει και η makeLatheShape3D() με μόνη διαφορά το ότι δεν εκτυλίσεται κυκλικά αλλά σε μια ευθεία πάνω στον άξονα των x.

Ένα παράδειγμα τέτοιου σχήματος είναι η σκεπή των κτιρίων του ξενοδοχείου waikiki. Το σχήμα φαίνεται στην παρακάτω εικόνα και απαρτίζεται από τέσσερα αντικείμενα, δυο LatheShape3D και 2 CustomLatheShape3D.



Εικόνα 4.14 : Η οροφή των κτιρίων του Waikiki.

4.1.2.18 Η ΜΕΘΟΔΟΣ MAKELATHEQUAD



Εικόνα 4.15 : Διάγραμμα LatheQuad.

Δημιουργεί ένα τετράπλευρο με τις δυο του πλευρές αντί για ευθεία γραμμή να είναι μια καμπύλη τύπου Hermite η οποία προσδιορίζεται από το πεδίο LatheShapeCoords.

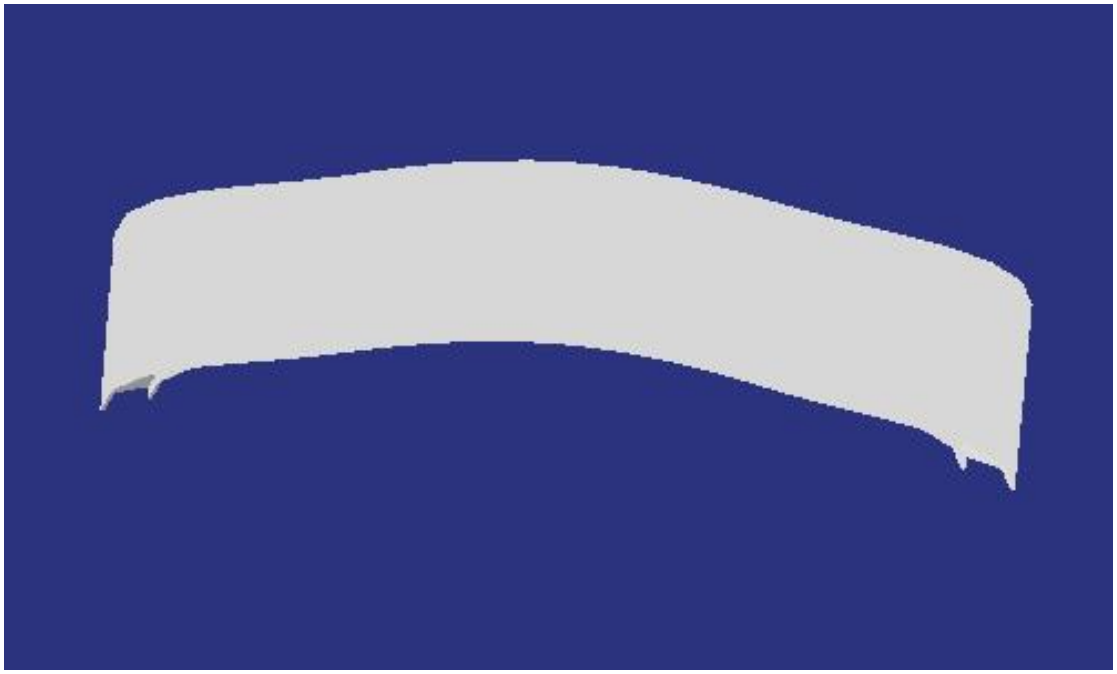
Όπως φαίνεται και στο διάγραμμα «αριθμο» η μέθοδος δημιουργεί ένα αντικείμενο τύπου LatheQuad και περνάει τις συντεταγμένες x και y. Αυτό με τη σειρά του δημιουργεί ένα HermiteCurve.

Κύρια δουλειά του HermiteCurve είναι να δημιουργεί μια καμπύλη γραμμή τύπου Hermite χρησιμοποιώντας τις συντεταγμένες x και y που δόθηκαν.

Χαρακτηριστικό παράδειγμα χρήσης αυτού του σχήματος είναι η οροφή του κτιρίου του ξενοδοχείου Boomerang. Η διαδικασία σχηματισμού της καμπύλης από τα ζευγάρια συντεταγμένων είναι παρόμοια με αυτή του σχήματος LatheShape3D.

Έτσι, αφού σχηματιστούν οι δυο παρόμοιες καμπύλες,οι οποίες θα αποτελέσουν τις δυο πλευρές του τετράπλευρου, τότε δημιουργούνται πολλά μικρότερα τετράπλυερα που γεμίζουν όλη την επιφάνεια ανάμεσα από τις δυο καμπύλες.

Το τελικό σχήμα που προκύπτει παρουσιάζεται στην παρακάτω εικόνα.



Εικόνα 4.16 : Το κτίριο του ξενοδοχείου Boomerang.

4.1.2.19 Η ΚΛΑΣΗ SHAPES3DENGINE

Είναι υπεύθυνη για την δημιουργία όλων των 3D αντικειμένων. Περιέχει static μεθόδους οι οποίες χρησιμοποιούν έναν xml reader για να διαβάσουν ομάδες αντικειμένων και τα χαρακτηριστικά τους από το αρχείο shapes3D.ini . Πριν ξεκινήσουν να διαβάζουν οι μέθοδοι το shapes3D.ini αρχικοποιούν ένα εργαλείο, το αντικείμενο CharacteristicsReader. Ουσιαστικά είναι ένας xml parser ο οποίος διαβάζει ένα xml αρχείο και επιστρέφει το αντίστοιχο Document. Στην συνέχεια ο CharacteristicsReader επιστρέφει το στοιχείο ρίζα (root element) του αρχείου.

Οι μέθοδοι λοιπόν κατασκευής των 3D αντικειμένων χρησιμοποιούν αυτό το root element και ψάχνουν, η καθεμία για διαφορετικά, xml αντικείμενα. Τα xml αντικείμενα είναι οργανωμένα σε τμήματα(blocks), που περιέχουν διαφορετικού είδους αντικείμενα το καθένα. Για παράδειγμα το block <buildings> περιέχει αντικείμενα <building>. Η δομή του xml αρχείου για τα κτίρια παρουσιάζεται στην παρακάτω εικόνα.

Εικόνα 4.17 : Παράδειγμα ενός αποθηκευμένου κτιρίου στο αρχείο Shapes.ini

Τα διάφορα parameters που μπορούν να έχουν (και να είναι αναγνωρίσιμα από τον xml parser) τα blocks τύπου shape είναι και τα πεδία ενός αντικειμένου Shape και περιγράφονται στην ενότητα Η ΚΛΑΣΗ SHAPE. Όσον αφορά τα parameters του ίδιου του αντικειμένου, θα περιγραφούν στις ενότητες που περιγράφουν το αντίστοιχο αντικείμενο.

Με τον ίδιο τρόπο που περιγράφεται στην εικόνα, αποθηκεύονται όλων τον ειδών τα αντικείμενα: environmentalObjects, Arrows, Pones, Bases.

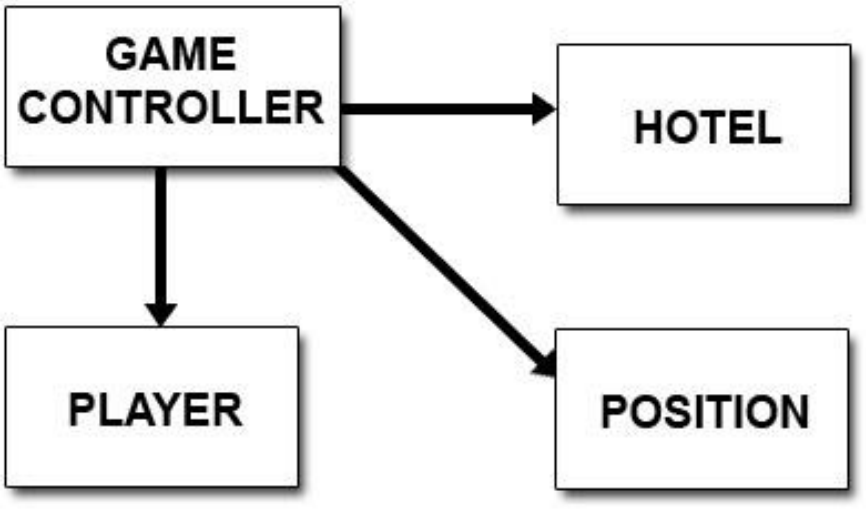
Οι μέθοδοι που χρησιμοποιεί η Shapes3Dengine είναι οι εξής:

- ***createEnvironmentalObjects:*** Συλλέγει όλα τα <environmentalObjects> αντικείμενα από το αρχείο. Για κάθε ένα από αυτά δημιουργεί ένα αντικείμενο Environmental3DObject και το τοποθετεί σε ένα αντικείμενο Vector.

- ***createBuildings:*** Συλλέγει όλα τα <buildings> αντικείμενα από το αρχείο. Για κάθε ένα από αυτά δημιουργεί ένα αντικείμενο Building και το τοποθετεί σε ένα αντικείμενο Vector.
- ***createPones:*** Συλλέγει όλα τα <Pones> αντικείμενα από το αρχείο. Για κάθε ένα από αυτά δημιουργεί ένα αντικείμενο Pone και το τοποθετεί σε ένα αντικείμενο Vector.
- ***loadArrowShape:*** Συλλέγει όλες τις γεωμετρικές πληροφορίες για την δημιουργία ενός αντικειμένου με σχήμα βέλους.
- ***loadBaseShape:*** Συλλέγει όλες τις γεωμετρικές πληροφορίες για την δημιουργία ενός αντικειμένου με σχήμα εισόδου.

create3Dtext: Δημιουργεί ένα UiText3D αντικείμενο από μια συγκεκριμένη λέξη. Για την δημιουργία του σχήματος αυτού δεν χρειάζονται περαιτέρω πληροφορίες εφόσον η java3D παρέχει ένα ολοκληρωμένο σχήμα.

4.2 GAME CONTROLLER

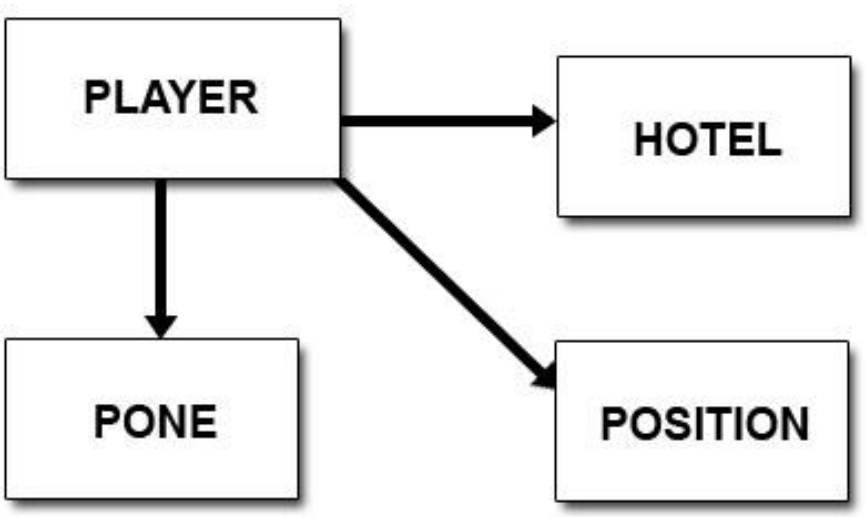


Εικόνα 4.18 : Διάγραμμα GameController.

Είναι η κλάση ελέγχου η οποία συντονίζει όλο το παιχνίδι. Είναι υπεύθυνη για την κατάσταση (state) του παιχνιδιού και για τη εξέλιξή του. Κύρια πεδία της είναι αντικείμενα του πακέτου game.entities :

- Player
- Hotel
- Position

4.2.1 Η Κλάση Player



Εικόνα 4.19 : Διάγραμμα Player.

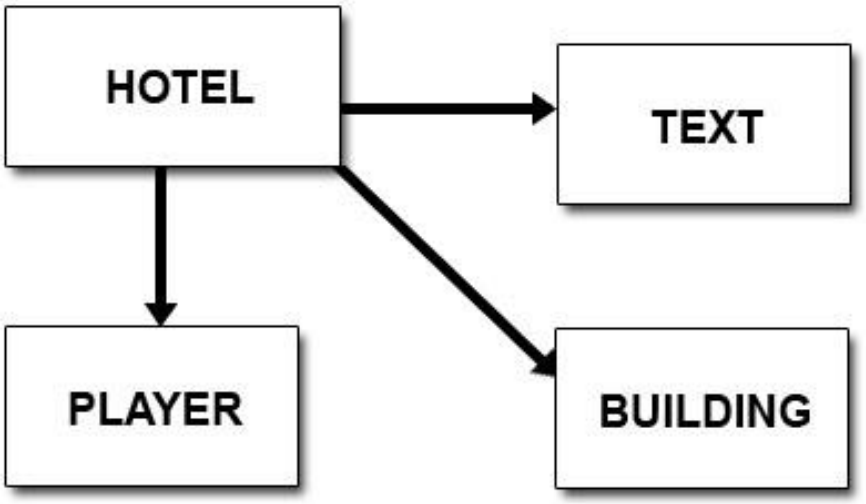
Αναπαριστά έναν παίκτη. Κυριότερα πεδία του:

- ***String name:*** Το όνομα του παίκτη
- ***Int noney:*** Τα λεφτά του παίκτη
- ***Int numberOfBases:*** Ο αριθμός των εισόδων που έχει αγοράσει και τοποθετήσει στα ξενοδοχεία του ο παίκτης
- ***Int lastRoll:*** Η τελευταία ζαριά του παίκτη. Χρησιμοποιείται για διάφορες λειτουργίες όπως είναι ο υπολογισμός της θέσης του παίκτη μετά από την κίνηση του για να ελέγξει εάν έχει περάσει από κάποιο checkpoint.

- ***Pone pone:*** Το πιόνι του παίκτη. Αυτή η κλάση ανήκει στο πακέτο graphics.entities καθώς παριστάνει το 3D αντικείμενο με το σχέδιο ενός πιονιού με το αντίστοιχο χρώμα.
- ***Vector<Hotel> Hotels:*** Είναι ένα Vector που περιέχει όλα τα ξενοδοχεία που έχει αγοράσει ο παίκτης. Η κλάση Hotel περιγράφεται στη συνέχεια.
- ***Position position:*** Είναι η τωρινή θέση του παίκτη πάνω στην διαδρομή του παιχνιδιού.

Επιπλέον, η κλάση Player περιέχει μεθόδους για διάφορες λειτουργίες όπως η αύξηση ή μείωση των χρημάτων του, έλεγχοι για το αν έχει περάσει κάποιο checkpoint.

4.2.2 Η Κλάση Hotel



Εικόνα 4.20 : Διάγραμμα Hotel.

Η κλάση αυτή αναπαριστά ένα ξενοδοχείο. Τα κυριότερα πεδία της είναι:

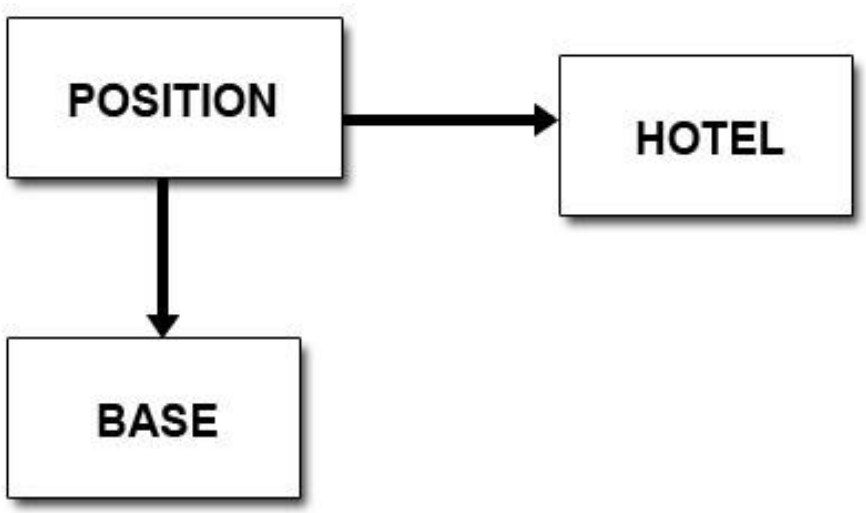
- ***String name:*** Το όνομα του ξενοδοχείου

- ***Player player:*** Ο ιδιοκτήτης του ξενοδοχείου.
- ***Vector<Building>:*** Είναι ένα Vector που περιέχει όλα τα κτίρια του ξενοδοχείου. Καθώς η κλάση Building αναπαριστά ένα 3D αντικείμενο με το σχήμα του αντίστοιχου κτιρίου, ανήκει στο πακέτο graphics.entities.
- ***Int fieldCost:*** Το κόστος του οικοπέδου.
- ***Boolean isFieldBought:*** Δηλώνει εάν το οικόπεδο του ξενοδοχείου έχει αγοραστεί από κάποιον παίκτη. Η αναγκαιότητα αυτού του πεδίου βρίσκεται στον κανόνα που λέει ότι εάν κάποιος παίκτης έχει στην κατοχή του μόνο το οικόπεδο του ξενοδοχείου και δεν έχει χτίσει κάποιο κτίριό του, τότε κάποιος άλλος παίκτης μπορεί να αγοράσει το οικόπεδο μισοτιμής.
- ***Int baseCost:*** Το κόστος μιας εισόδου για αυτό το ξενοδοχείο.
- ***Boolean hasOutskirts:*** Δηλώνει εάν στο ξενοδοχείο έχουν χτιστεί οι εξωτερικοί χώροι. Το πεδίο αυτό χρησιμοποιείται για να ελέγξει εάν ένας παίκτης μπορεί να χτίσει πάνω στο ξενοδοχείο. Για παράδειγμα, εάν η τιμή του πεδίου είναι true τότε δεν υπάρχει κάτι άλλο που μπορεί να χτίσει ο παίκτης για αυτό το ξενοδοχείο.
- ***Int outskirtsCost:*** Το κόστος των εξωτερικών χώρων.
- ***Int[] outskirtsCheckinCost:*** Πίνακας που κρατάει το κόστος της βραδιάς του ξενοδοχείου εφόσον έχουν χτιστεί και οι εξωτερικοί χώροι. Το μήκος του πίνακα είναι πάντα 6 εφόσον οι περισσότερες μέρες που μπορεί να «καθήσει» ένας παίκτης είναι 6.
- ***Point3f labelCoords:*** Περιέχει τις συντεταγμένες που θα πρέπει να ζωγραφιστεί το αντικείμενο Text επάνω από το οικόπεδο του ξενοδοχείου.
- ***Text tet3d:*** Είναι το 3D αντικείμενο με το όνομα του ξενοδοχείου που ζωγραφίζεται επάνω από το οικόπεδο του. Ανήκει στο πακέτο graphics.entities.
- ***String sidePosition:*** Μπορεί να πάρει 2 τιμές. Είτε «left» είτε «right» και δηλώνουν σε ποιά πλευρά της διαδρομής βρίσκεται το ξενοδοχείο. Το πεδίο αυτό χρησιμοποιείται από το GraphicsPanel και δίνει οδηγίες

σε ποιά πλευρά θα πρέπει να ζωγραφιστεί η είσοδος που αγοράζει ο παίκτης.

Η κλάση Hotel δημιουργεί 8 static αντικείμενα που αναπαριστούν τα 8 ξενοδοχεία που έχει το παιχνίδι. Παράλληλα ανατίθενται και οι default τιμές των πεδίων τους όπου υπάρχουν.

4.2.3 Η Κλάση Position



Εικόνα 4.21 : Διάγραμμα Position.

Η κλάση αυτή αναπαριστά ένα κουτάκι/θέση πάνω στη διαδρομή του παιχνιδιου. Τα κυριότερα πεδία της είναι :

- ***Int Ordinal:*** Ο αριθμός της θέσης πάνω στη διαδρομή.

- ***Boolean isOccuppied:*** Δηλώνει εάν στην θέση αυτή υπάρχει ήδη κάποιος άλλος παίκτης. Το πεδίο αυτό χρησιμοποιείται για την αποφυγή της κατάστασης κατά την οποία 2 πιόνια βρίσκονται στην ίδια θέση. Εάν η τιμή αυτή είναι true και κάποιο πιόνι σταματήσει σε αυτήν, τότε προχοράει στο επόμενο.
- ***String positionType:*** Δηλώνει τον τύπο της θέσης. Μπορεί να πάρει μια από τις παρακάτω τιμές:
    - BUILD
    - BUY
    - FREE BUILDING
    - FREE BASE
    - START
    - PARKING
- ***Vector<Hotel> hotels:*** Είναι ένα Vector που περιέχει τα ξενοδοχεία τα οποία ο παίκτης μπορεί να αγοράσει εάν βρίσκεται σε θέση τύπου BUY καθώς και τα ξενοδοχεία για τα οποία κάποιος παίκτης μπορεί να τοποθετήσει μια είσοδο.
- ***Vector3f positionCoords:*** Οι συντεταγμένες της θέσης. Είναι ακριβώς το κέντρο της θέσης πάνω στο οποίο σταματάει ένα πιόνι.
- ***Vector3f baseCoordsLeft:*** Οι συντεταγμένες της θέσης πάνω στις οποίες μπορεί να τοποθετηθεί μια είσοδος στα αριστερά της. Χρησιμοποιείται για την τοποθέτηση μιας εισόδου που προορίζεται για ξενοδοχείο στην αριστερή πλευρά της διαδρομής.
- ***Vector3f baseCoordsRight:*** Οι συντεταγμένες της θέσης πάνω στις οποίες μπορεί να τοποθετηθεί μια είσοδος στα δεξιά της. Χρησιμοποιείται για την τοποθέτηση μιας εισόδου που προορίζεται για ξενοδοχείο στην δεξιά πλευρά της διαδρομής.
- ***Double baseAngleRight:*** Η τιμή της γωνίας που θα πρέπει να στρίψει η είσοδος πάνω στον άξονα y για να τοποθετηθεί σωστά πάνω στην θέση, όταν βρίσκεται στα δεξιά της.
- ***Double baseAngleLeft:*** Η τιμή της γωνίας που θα πρέπει να στρίψει η είσοδος πάνω στον άξονα y για να τοποθετηθεί σωστά πάνω στην θέση, όταν βρίσκεται στα αριστερά της.

- ***Base base:*** Η είσοδος εάν έχει τοποθετηθεί κάποια. Αλλιώς το πεδίο αυτό είναι null. Χρησιμοποιείται επίσης στον έλεγχο της θέσης για το εάν έχει ήδη τοποθετηθεί κάποια είσοδος σε αυτήν.

Η κλάση Position δημιουργεί 31 static αντικείμενά της τα οποία παριστάνουν και τις 31 θέσεις πάνω στην διαδρομή του παιχνιδιού. Επίσης αρχικοποιεί κάποιες default τιμές των αντικειμένων όπου χρειάζεται.

4.2.4 Οι κυριότερες μέθοδοι της κλάσης GameController

Εφόσον περιγράφηκαν τα κυριότερα πεδία του GameController θεωρείται σκόπιμο σε αυτό το σημείο να περιγραφούν κάποιες μεθόδοι της κλάσης οι οποίες και εκτελούν μερικές από τις σημαντικότερες λειτουργίες του παιχνιδιού. Επιπλέον, η διαδικασία αυτή θα βοηθήσει τον αναγνώστη να κατανοήσει πως ο GameController ευθύνεται για την ομαλή ροή του παιχνιδιού. Οι σημαντικότερες λοιπόν μεθόδοι είναι οι εξής:

- ***initializePlayerPosition() :*** Τοποθετεί τα πιόνια των παικτών στην αρχική τους θέση.
- ***fixPlayerOrder():*** Η διαδικασία με την οποία αποφασίζεται ποιός παίκτης θα παίξει πρώτος, δεύτερος κτλ. Η διαδικασία αυτή μπορεί να περιγραφεί ως εξής:
    1. Ρίχνουν όλοι οι παίκτες το ζάρι.
    2. Βρίσκεται ο παίκτης ή παίκτες με την μεγαλύτερη ζαριά
    3. Εάν είναι ένας, τότε τοποθετείτε στην αρχή του Vector
    4. Εάν είναι πολλοί, τότε ξαναρίχνουν το ζάρι και εκτελείτε ξανά το βήμα 2
    5. Έλεγχος των υπόλοιπων ρίψεων και ταξινόμηση των υπόλοιπων παικτών σύμφωνα με την ζαριά τους
    6. Εάν υπάρχουν 2 ή περισσότεροι παίκτες με την ίδια ζαριά τότε εκτελείτε ξανά το βήμα 2.
    7. Τερματισμός του αλγόριθμου όταν έχουν ταξινομηθεί όλοι οι παίκτες.

- ***playRound():*** Είναι η κεντρικότερη μέθοδος και σηματοδοτεί την αρχή ενός γύρου από τη στιγμή που η σειρά περνάει στον επόμενο παίκτη. Μέσα από αυτήν εκτελούνται με αυτήν την χρονική σειρά οι μεθόδοι:
    - throwDie()
    - moveCurrentPlayer()
    - checkIfPlayerPassedCheckPoint()

  Στη συνέχεια γίνεται ο έλεγχος για το τι τύπου θέση έχει σταματήσει το πιόνι του ενεργού παίκτη και ανάλογα καλείται η μέθοδος buildBuilding(), buyHotelField(), buildBuilding().

- ***throwDie(currentPlayer):*** Καλείτε από την playRound() και χρησιμοποιεί την κλάση NumberedDie για την παραγωγή ενός τυχαίου αριθμού από το 1 μέχρι το 6. Αυτή η τιμή τίθεται στο πεδίο lastRoll του ενεργού Player.
- ***moveCurrentPlayer():*** Υπολογίζεται η νέα θέση στην οποία πρέπει να μετακινηθεί το πιόνι του ενεργού παίκτη σύμφωνα με το lastRoll του. Εάν η θέση εκείνη είναι κατηλλειμένη, τότε προχωράει ακόμα μια θέση μπροστά.
- ***checkIfPlayerPassedCheckPoint():*** Ελέγχει εάν ο παίκτης, σύμφωνα με την τελευταία και την καινούργια θέση του, έχει περάσει απ'το checkPoint ,από το οποίο λαμβάνει $2000 bonus, ή απ'το checkPoint από το οποίο μπορεί να αγοράσει μια είσοδο για κάθε ένα ξενοδοχείο που έχει στην κατοχή του. Εάν έχει περάσει από το δεύτερο checkPoint τότε καλείτε η buyBase().
- ***payPlayer():*** Από τους πρώτους ελέγχους που γίνονται στην playRound() είναι εάν ο παίκτης προχώρησε σε μια θέση στην οποία έχει τοποθετήσει μια είσοδο κάποιος άλλος παίκτης. Εάν ναι, τότε υποχρεώνεται να ρίξει το ζάρι για ακόμα μια φορά. Η ζαριά του δηλώνει το πόσες μέρες θα μείνει στο αντίστοιχο ξενοδοχείο, όπου αυτό μεταφράζεται στο ποσό που πρέπει να δώσει ο ενεργός παίκτης στον κάτοχο της εισόδου. Τέλος, αφαιρείτε το ποσό από τα χρήματα του ενεργού παίκτη και προστίθενται στον κάτοχο της εισόδου.

- ***buildBuilding():*** Εάν ο παίκτης έχει σταματήσει πάνω σε μια θέση τύπου Build, μπορεί να χτίσει εάν φυσικά έχει στην κατοχή του κάποιο οικόπεδο. Οι ενέργειες που εκτελούνται από την μέθοδο είναι οι εξής:
    - Έλεγχος για το αν ο παίκτης έχει στην κατοχή του οικόπεδα.
    - Έλεγχος για το αν ο παίκτης έχει χρήματα να χτίσει στα ξενοδοχεία του.
    - Δίνει εντολή στο PlayerControlsPanel να εκτυπώσει το κατάλληλο μήνυμα που ενημερώνει το χρήστη ότι θα πρέπει να αποφασίσει εάν θα χτίσει ή όχι. Σε αυτό το σημείο σταματάει η εκτέλεση του προγράμματος και το σύστημα περιμένει την απάντηση του παίκτη μέσω των κουμπιών του PlayerControlsPanel.
    - Αφού αποφασίσει ότι θέλει να χτίσει ο χρήστης, μια επιπλέον πληροφορία στέλνεται :θα πρέπει να ρίξει το ζάρι για την άδεια χτισίματος κτιρίου. Ανάλογα με το τι θα φέρει το ζάρι η μέθοδος ενημερώνει το GraphicsPanel ότι θα πρέπει να ζωγραφίσει πάνω στο οικόπεδο του ξενοδοχείου το αντίστοιχο κτίριο. Αφαιρείται από τα χρήματα του παίκτη το κόστος του κτιρίου.
    - Εάν δεν μπορεί να χτίσει ο παίκτης τότε καλείτε η nextPlayer() η οποία δίνει την σειρά στον επόμενο παίκτη και η playRound() ξεκινάει από την αρχή.
- ***buyHotelField():*** Εάν ο παίκτης έχει σταματήσει σε μια θέση τύπου Buy τότε μπορεί να αγοράσει κάποιο από τα οικόπεδα που βρίσκονται γύρω από αυτήν. Οι ενέργειες που εκτελούνται από την μέθοδο είναι οι εξής:
    - ‘Ελεγχος για το αν τα οικόπεδα είναι ήδη αγορασμένα και χτισμένα από κάποιον άλλον παίκτη. Εάν ναι, τότε καλείτε η nextPlayer().
    - Εάν μπορεί να αγοράσει κάποιο από τα οικόπεδα τότε στέλνει το αντίστοιχο μήνυμα στον PlayerControlsPanel και σταματάει την εκτέλεση του προγράμματος περιμένοντας την απάντηση του παίκτη μέσα από τα κουμπιά του PlayerControlsPanel.

- Εάν αποφασίσει να αγοράσει το οικόπεδο τότε η αξία του αφαιρείται από τα χρήματά του.

- ***buildBuilding():*** Εάν ο παίκτης έχει σταματήσει σε μια θέση τύπου Build τότε μπορεί να χτίσει σε κάποιο από τα ξενοδοχεία που έχει στην κατοχή του. Το ίδιο ισχύει εάν έχει σταματήσει σε θέση δωρεάν κτιρίου. Φυσικά γίνονται οι κατάλληλοι έλεγχοι για να διαπιστωθεί εάν ο παίκτης μπορεί να χτίσει ακόμα και αν έχει ξενοδοχεία στην κατοχή του. Μπορεί να έχει χτίσει έως και τους εξωτερικούς χώρους οπότε και να μην υπάρχει κάτι άλλο να χτίσει για κάποιο ξενοδοχείο. Οι ενέργειες που εκτελούνται από την μέθοδο είναι οι εξής:
  - Έλεγχος για το αν ο παίκτης μπορεί να χτίσει σε κάποιο ξενοδοχείο. Εάν όχι τότε καλείτε η netPlayer().
  - ‘Ελεγχος για το αν ο παίκτης έχει αρκετά χρήματα για να χτίσει σε κάποιο ξενοδοχείο που έχει στην κατοχή του. Εάν δεν έχει το κατάλληλο ποσό τότε καλείτε η nextPlayer().
  - Εάν μπορεί να χτίσει, τότε τα υποψήφια ξενοδοχεία στέλνονται στο PlayerControlsPanel μαζί με κάποιο αντίστοιχο μήνυμα το οποίο ρωτάει το χρήστη εάν και για ποιό ξενοδοχείο θέλει να χτίσει. Παράλληλα σταματάει η εκτέλεση του προγράμματος περιμένοντας την απάντηση του παίκτη.
  - Εάν επιλέξει ο παίκτης να χτίσει τότε καλείτε μέσω δεύτερου μηνύματος να ρίξει το ζάρι για την άδεια χτισίματος.
  - Εάν τύχει οτιδήποτε άλλο εκτός από «κόκκινο» το οποίο και σημαίνει ότι δεν μπορεί να χτίσει, τότε στέλνεται η κατάλληλη ειδοποίηση στο GraphicsPanel να ζωγραφίσει το κτίριο στην κατάλληλη θέση. Στη συνέχεια αφαιρείται η αξία χτισίματος του κτιρίου από τα χρήματα του παίκτη και καλείται η nextPlayer().
  - Εάν έχει τύχει κόκκινο τότε καλείτε η nextPlayer().
- ***buyBase():*** Καλείτε όταν ο παίκτης περάσει το checkPoint που του επιτρέπει να αγοράσει μια είσοδο ή εάν τύχει στην θέση δωρεάν εισόδου. Οι ενέργειες που εκτελεί είναι οι εξής:

- Έλεγχος για το αν ο παίκτης έχει τη δυνατότητα να τοποθετήσει μια είσοδο σε κάποιο σημείο. Εάν όχι, τότε καλείτε η nextPlayer().
- Εάν μπορεί να αγοράσει είσοδο τότε στέλνεται η λίστα με τα υποψήφια ξενοδοχεία στην PlayerControlsPanel μαζί με το αντίστοιχο μήνυμα που ενημερώνει τον παίκτη ότι μπορεί να αγοράσει είσοδο για αυτά τα ξενοδοχεία. Παράλληλα, σταματάει η εκτέλεση του προγράμματος και στέλνεται ειδοποίηση στην GraphicsPanel να ζωγραφίσει βελάκια επάνω από τις θέσεις στις οποίες μπορεί ο παίκτης να τοποθετήσει βάση για το ξενοδοχείο που έχει επιλεγμένο. Εάν τα υποψήφια ξενοδοχειά είναι πάνω από ένα και ο παίκτης επιλέγει μεταξύ αυτών, τότε στέλνεται ειδοποίηση στην Graphicspanel να καταστρέφει τα βελάκια στις παλίες τοποθεσίες και να τα ξαναζωγραφίζει στις καινούριες.
- Αφού διαλέξει ο παίκτης μια θέση κάνοντας click στο αντίστοιχο βελάκι τότε ζωγραφίζεται η είσοδος, αφαιρούνται χρήματα από τον παίκτη, το αντικείμενο είδοσος (base) τίθεται στην θέση (position) και καλείτε η nextPlayer().

- ***nextPlayer():*** Δίνει την σειρά στον επόμενο παίκτη.

Όλες οι ενέργειες και οι αποφάσεις που εκτελεί ο GameController γράφονται στο log και στέλνοται στο LogPanel με το αντίστοιχο χρώμα του κάθε ενεργού παίκτη. Η εκτέλεση του προγράμματος σταματάει με τη βοήθεια της μεθόδου wait(condition) και συνεχίζει από εκείνο το σημείο με την κλήση της wakeUp(). Η παύση επιτυγχάνεται με το αντικείμενο : Object wait που λειτουργεί ως Thread και την εκτέλεση των εντολών

```
wait.wait();
```

και

```
wait.notify();
```

για την παύση και την συνέχεια της εκτέλεσης αντίστοιχα.

Τέλος, παρατίθενται οι listeners που χειρίζεται και ενημερώνει η κλάση :

- PlayerListener
- PlayerMessageListener
- CandidateHotelsListener
- CurrentPlayerChangedListener

Όλοι οι παραπάνω listeners στέλνουν πληροφορίες στην PlayerControlsPanel ενώ με την χρήση του PlayerActionListener ο GameController ακούει τις αποφάσεις (πάτημα κουμπιών) του ενεργού παίκτη από την PlayerControlsPanel.

4.3 ΤΟ ΠΑΚΕΤΟ GRAPHICS.BEHAVIOR

Περιέχει όλες τις κλάσεις behavior. Είναι αυτές που δίνουν «ζωή» στα αντικείμενα. Είναι υπεύθυνες για την κίνηση και την περιστροφή των αντικειμένων (animation) καθώς και για την προσθήκη συγκεκριμένων ικανοτήτων των αντικειμένων.

Στην java3D η αλληλεπίδραση και η κίνηση καθορίζονται με τη χρήση της κλάσης Behavior. Είναι μια abstract κλάση που προσφέρει έναν μηχανισμό ένταξης κώδικα για την αλλαγή του scene graph. Η κλάση Behavior και οι απόγονοί της είναι σύνδεσμοι στον κώδικα του προγραμματιστή ο οποίος προσφέρει αλλαγές στα γραφικά και τους ήχους του virtual universe.

Ο σκοπός ενός αντικειμένου Behavior στο scene graph είναι να το αλλάζει, ή να αλλάζει αντικείμενα μέσα σε αυτό ως απάντηση σε κάποιο ερέθισμα. Ερέθισμα μπορεί να είναι το πάτημα ενός κουμπιού, η κίνηση του ποντικιού, η σύγκρουση αντικειμένων, η πάροδος ενός συγκεκριμένου χρονικού διαστήματος, κάποιο άλλο γεγονός ή και συνδυασμός αυτών. Οι αλλαγές αυτές μπορεί να είναι η προσθήκη αντικειμένων στο scene graph ή η διαγραφή τους, η αλλαγή κάποιων χαρακτηριστικών τους, η αλλαγή της θέσης τους ή συνδυασμός όλων αυτών.

Οι κλάσεις behavior αναπτύσονται και χρησιμοποιούνται σε μια εφαρμογή για να προσφέρουν κάποια αλληλεπίδραση με το χρήστη ή για την κίνηση (animation) ενός αντικειμένου.

Το παρακάτω παράδειγμα παρουσιάζει την διαφορά των δυο αυτών όρων:

Έστω εφαρμογή στην οποία η κάμερα κινείται ως αποτέλεσμα του πατήματος κουμπιών ή της κίνησης του ποντικιού. Αυτού του είδους η κίνηση είναι μια αλληλεπίδραση καθώς είναι άμεσο αποτέλεσμα χειρισμού του χρήστη. Άλλα αντικείμενα μπορεί να επηρεαστούν όμως από την κίνηση της κάμερας. Σε αυτήν την περίπτωση τέτοιες κινήσεις είναι animations.

Σε αυτήν την εφαρμογή αναπτύχηκαν και τα δυο είδη behavior.

4.3.1 INTERACTION

4.3.1.1 Η ΚΛΑΣΗ KEYBEHAVIOR

Κληρονομεί την κλάση Behavior. Το ερέθισμα που προκαλεί την «αντίδρασή» της είναι το πάτημα ενός κουμπιού. Σκοπός της είναι η μετακίνηση ενός αντικειμένου πάνω στους τρεις άξονες ή η περιστροφή του γύρω από τους τρεις άξονες. Τα κουμπιά που υποστηρίζονται από την Behavior:

- ***UP_KEY:*** Μετακινεί το αντικείμενο πάνω στον άξονα των z με βήμα +step. Δηλαδή το απομακρύνει από τον παρατηρητή προς το βάθος της σκηνής.
- ***DOWN_KEY:*** Μετακινεί το αντικείμενο πάνω στον άξονα των z με βήμα -step. Δηλαδή το φέρνει πιο κοντά στα μάτια του παρατηρητή.
- ***LEFT_KEY:*** Μετακινεί το αντικείμενο πάνω στον άξονα των x με βήμα -step. Δηλαδή προς τα αριστερά.
- ***RIGHT_KEY:*** Μετακινεί το αντικείμενο πάνω στον άξονα των x με βήμα +step. Δηλαδή προς τα δεξιά.
- ***U:*** Μετακινεί το αντικείμενο πάνω στον άξονα των y με βήμα +step. Δηλαδή προς τα επάνω.

- ***D:*** Μετακινεί το αντικείμενο πάνω στον άξονα των y με βήμα -step. Δηλαδή προς τα κάτω.
- ***Y:*** Περιστρέφει το αντικείμενο γύρω από τον άξονα των y.
- ***X:*** Περιστρέφει το αντικείμενο γύρω από τον άξονα των x.
- ***Z:*** Περιστρέφει το αντικείμενο γύρω από τον άξονα των z.

Ο κώδικας που υλοποιεί όλη την παραπάνω διαδικασία δεν προορίζεται για κάποια λειτουργία που μπορεί να κάνει ο χρήστης κατά τη διάρκεια του παιχνιδιού. Αναπτύχθηκε καθαρά για λόγους σχεδιασμού της εφαρμογής.

Καθώς μετακινείτε το εκάστοτε αντικείμενο, τυπώνει τις καινούργιες συντεταγμένες του. Έτσι βρέθηκαν όλες οι συντεταγμένες στις οποίες χρειάζεται να βρίσκονται κάποια αντικείμενα κατά την εξέλιξη του παιχνιδιού. Για παράδειγμα οι συντεταγμένες χτισίματος ενός κτιρίου, η μετακίνηση των πιονιών στις θέσεις της διαδρομής ή η τοποθέτηση εισόδων στα πλαϊνά των θέσεων.

4.3.1.2 Η ΚΛΑΣΗ MOUSEOVERBEHAVIOR

Δίνει ως αποτέλεσμα τον χρωματισμό ενός αντικειμένου εάν το ποντίκι βρίσκεται από πάνω του. Στο παράδειγμά μας, όταν ένας παίκτης τοποθετήσει το ποντίκι επάνω από ένα βελάκι, τότε αυτό χρωματίζεται κόκκινο ως ένδειξη ότι εάν κάνει κλικ σε αυτήν την θέση, η νέα του είσοδος θα τοποθετηθεί εκεί.

Κληρονομεί την κλάση Behavior. Δέχεται ως παράμετρο ένα canvas3D και ένα branchgroup αυτού μαζί με όλα τα αντκείμενα που βρίσκονται μέσα σε αυτό. Είναι μια υπολογιστικά βαριά συμπεριφορά καθώς ερέθισμά της είναι η κίνηση του ποντικιού. Παρατηρεί την κίνηση του και όταν αυτό σταματήσει σε μια θέση τότε υπολογίζει τις συντεταγμένες κάτω από τις οποίες βρίσκεται.

Στη συνέχεια περνάει τις συντεταγμένες στο canvas3D και επιλέγει το κοντινότερο αντικείμενο του branchgroup ,εάν υπάρχει. Εάν όντως υπάρχει

αυτό επιστρέφεται σε ένα αντικείμενο PickResult. Τέλος, αλλάζουμε εμφάνιση στο αντικείμενο προσθέτοντας του ένα coloredMaterial με χρώμα κόκκινο.

4.3.1.3 Η ΚΛΑΣΗ MOUSEPICKBEHAVIOR

Επιτρέπει την επιλογή ενός αντικειμένου από την σκηνή. Κληρονομεί την κλάση PickMouseBehavior. Το ερέθισμα της είναι το αριστερό κλικ του ποντικιού.

Όπως και η MouseOverBehavior, δέχεται ένα branchgroup με τα αντικείμενα τα οποία περιέχει. Στη συνέχεια, μέσω της μεθόδου updateScene ελέγχει εάν στις συντεταγμένες που βρισκόταν το ποντίκι την ώρα που πατήθηκε το αριστερό του κλικ βρίσκονται πολύ κοντά ή επάνω από κάποιο αντικείμενο του branchgroup που αναγνωρίζει η συμπεριφορά.

Εάν όντως υπάρχει κάποιο αντικείμενο κάτω από το ποντίκι τότε αυτό αποθηκεύεται σε ένα αντικείμενο τύπου PickResult. Στο δικό μας παράδειγμα, το branchgroup της συμπεριφοράς περιέχει την ομάδα από βέλη που εμφανίζεται κάθε φορά που ένας παίκτης χρειάζεται να διαλέξει μια θέση για να τοποθετήσει την είσοδό του.

Εφόσον επιλέξει ένα βελάκι, αναγνωρίζεται η θέση πάνω από την οποία βρισκόταν το βελάκι και στην συνέχεια αποκόπτεται το branchgroup(detach) από τη σκηνή, δηλαδή δεν είναι πια ορατά από τον παρατηρητή. Τέλος δημιουργείται ένα αντικείμενο Base και τοποθετείται σε αυτήν την θέση.

4.3.2 ANIMATION

Όπως και με την αλληλεπίδραση, animation στην java3D επιτυγχάνεται με behavior αντικείμενα. Ένα σετ τέτοιων κλάσεων ονομάζονται Interpolators. Ένα αντικείμενο interpolator μαζί με ένα Alpha χειραγωγεί κάποιες παραμέτρους του scene graph για να δημιουργήσει κάποιο animation με βάση

το χρόνο. Το αντικείμενο Alpha παρέχει την χρονική διάρκεια, καθώς και την χρονική στιγμή την οποία θα ξεκινήσει το animation.

Υπάρχουν πολλά είδη Interpolator και καθένα προσφέρει συγκεκριμένου τύπου animation ή και συνδυασμό τους. Μερικά από αυτά είναι:

- ColorInterpolator
- PathInterpolator
- PositionPathIntepolator:
    - PositionPathInterpolator
    - RotationPathInterpolator
    - RotPosPathInterpolator
    - RotPosScalePathInterpolator
- RotationInterpolator
- ScaleInterpolator
- SwitchValueInterpolator
- TransparencyInterpolator
- TCBSplinePathInteporlator

Σε αυτήν την εφαρμογή χρησιμοποιήθηκαν οι ColorInterpolator, PositionPathInterpolator kai RotationInterpolator.

4.3.2.1 Η ΚΛΑΣΗ PONEBEHAVIOR

Ο ρόλος αυτής της συμπεριφοράς είναι να μετακινεί τα πιόνια των παικτών τόσες θέσεις όσες έτυχε το ζάρι. Χρησιμοποιεί ένα αντικείμενο PositionPathInterpolator ο οποίος μετακινεί τα πιόνια πάνω στον άξονα των x.

Το χρονικό διάστημα που χρειάζεται το πιόνι για να μετακινηθεί από μια θέση στην επόμενη είναι το ένα δευτερόλεπτο. Έτσι για μια κίνηση πέντε θέσεων θα περάσουν πέντε δευτερόλεπτα.

Το interpolator αντικείμενο δέχεται τις θέσεις στις οποίες θα πρέπει να μετακινήσει το αντικείμενο και το χρονικό διάστημα που θα χρειαστεί μέσω ενός αντικειμένου Alpha.

4.3.2.2 Η ΚΛΑΣΗ TEXTBEHAVIOR

Ρόλος της είναι να περιστρέφει ένα text αντικείμενο γύρω από τον άξονα των y και ταυτόχρονα να το χρωματίζει σύμφωνα με το χρώμα κάποιου παίκτη. Η συμπεριφορά τίθεται σε εφαρμογή μόλις ένας παίκτης αγοράσει ένα οικόπεδο.

Για να πετύχει η κλάση αυτού του είδους το animation χρησιμοποιεί δυο interpolators, ColorInterpolator και RotationInterpolator.

Ο χρωματισμός του αντικειμένου ξεκινάει με το αντικείμενο να είναι σε άσπρο χρώμα και σταδιακά να παίρνει το χρώμα του παίκτη. Ο χρόνος που χρειάζεται για αυτή την διαδικασία είναι πέντε δευτερόλεπτα και ορίζεται με ένα alpha αντικείμενο. Ο χρόνος πλήρης περιστροφής του αντικειμένου είναι τα τριάντα δευτερόλεπτα. Αυτό έχει ως αποτέλεσμα μια αργή κίνηση.

4.3.2.3 Η ΚΛΑΣΗ ARROWBEHAVIOR

Είναι υπεύθυνη για την περιστροφή ενός βέλους γύρω από τον άξονα των y. Χρησιμοποιεί έναν RotationInterpolator παρόμοιο με αυτού της TextBehavior.

ΚΕΦΑΛΑΙΟ 5: Προτάσεις για βελτίωση και ανάπτυξη

Στο κεφάλαιο αυτό πολλά μπορούν να ειπωθούν. Η εφαρμογή αυτή αναπτύχθηκε χωρίς προηγούμενη εμπειρία πάνω στην java3D και επομένως πάρα πολλές δυνατότητες παρέμειναν ανεκμετάλλευτες. Οι δυνατότητες του java3D API περιορίζονται μόνο στην φαντασία και στις ικανότητες αυτού που το χειρίζεται.

Δεδομένης αυτής της κατάστασης, η εφαρμογή μπορεί να χρησιμοποιηθεί για περαιτέρω εκπαίδευση και πειραματισμό με στόχο συγκεκριμένα πάντα αποτελέσματα.

Μερικά από αυτά είναι:

Υποστήριξη multiplayer μέσω διαδικτύου: Δυο έως τέσσερεις παίκτες θα μπορούν να συνδέονται μεταξύ τους και να παίζουν το παιχνίδι ο καθένας από τον δικό του υπολογιστή.

Βελτίωση του GUI του παιχνιδιού: Αυτή τη στιγμή γίνεται μόνο χρήση της Swing για την κατασκευή του GUI. Καθώς το API της java3D δεν παρέχει κάποιο μηχανισμό δημιουργίας η επόμενη καλύτερη λύση είναι ο σχεδιασμός του με τη χρήση της java2D.

Προσθήκη ήχου: Ο ήχος σε ένα παιχνίδι είναι αν μη τι άλλο απαραίτητος. Πάρα πολλές πολυμεσικές εφαρμογές «χάνουν» στις συνολικές εντυπώσεις εξαιτίας κακής διαχείρισης των ηχητικών εφέ τους. Πολλά σημεία του παιχνιδιού θα μπορούσαν να περιέχουν ήχο προσθέτοντας έτσι την συνολική εμπειρία των παικτών.

Βελτίωση του φωτισμού της σκηνής

Βελτίωση των textures και των αντικειμένων της σκηνής: Σε αυτό το κομμάτι της εφαρμογής δεν υπάρχει κάποιο όριο και ο στόχος είναι ένας, να δείχνουν και να «δένουν» τα αντικείμενα της σκηνής καλύτερα τόσο μεταξύ τους όσο και με βάση μιας συγκεκριμένης «ατμόσφαιρας» που θα κυριαρχεί στο παιχνίδι. Περιθώριο βελτίωσης θα υπάρχει πάντα καθώς οι συνδυασμοί είναι ατελείωτοι με μόνο περιορισμό την φαντασία και τις ικανότητες του προγραμματιστή.

Βελτίωση του animation: Η κίνηση ενός αντικειμένου του δίνει ζωή. Όσο πιο φυσική και ομαλή είναι τόσο πιο ωραίο το αποτέλεσμα. Σ’ αυτόν τον τομέα υπάρχει χώρος για σημαντικές βελτιώσεις και προσθήκες. Για παράδειγμα, το χτίσιμο των κτιρίων, αντί να τοποθετούνται από τη μια θέση στην άλλη,θα μπορούσαν να ακολουθούν μια διαδρομή σε συνεργασία με κάποιο εφέ σκόνης ή η αποσυναρμολόγηση των επιμέρους σχημάτων και

επανασυναρμολόγησή τους στην κατάλληλη τοποθεσία. Η κίνηση που ακολουθούν τα πιόνια θα μπορούσε να βελτιωθεί έτσι ώστε να μην περνάει ένα πιόνι μέσα από ένα άλλο.

Βιβλιογραφία - Αναφορές – Πηγές

- http://www.macs.hw.ac.uk/~nkt/graphics/B%20Java3D%20Animation%20Slides.pdf
- http://www.fastgraph.com/makegames/3drotation/
- http://fivedots.coe.psu.ac.th/~ad/jg/
- http://www.turbosquid.com/3d
- http://www.the3dstudio.com/
- http://programmedlessons.org/VectorLessons/vectorIndex.html