



**Εφαρμογή Java, παίγνιο για επιτραπέζιο υπολογιστή (java game application for PC) με δυνατότητα σύνδεσης σε δίκτυο**

**(online networked, multiplayer capable)**



**Πτυχιακή εργασία**

**Ιωαννίδης Ιωάννης**

**Επιβλέπων καθηγητής: Μπινιατίδης Συμεών**

**ΘΕΣΣΑΛΟΝΙΚΗ 2009**

© 2010 Alexander Technological Educational Institute of Thessaloniki © 2010 Ioannidis Ioannis Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with the Front-Cover Texts being LIST. A copy of the license is included in the section entitled “GNU Free Documentation License”.

## Περιεχόμενα

1. Πρόλογος.....	4
2. Γενική Περιγραφή Παιχνιδιού - Περίληψη.....	5
3. Game Overview - Summary.....	6
4. Έλεγχος-Κουμπιά Πλοήγησης.....	7
5. Στόχοι - Όραμα.....	7
6. Κυρίως Κορμός Παιχνιδιού.....	7
7. Περιγραφή Ανάλυση Κανόνων Παιχνιδιού.....	8
8. Περιγραφή Κλάσεων.....	12
9. Ανάλυση Κλάσεων.....	15
10. Συσχετισμός Κλάσεων – Δια - Λειτουργικότητα.....	41
11. Βοηθητικό Λογισμικό.....	57
12. Τεχνολογίες.....	62
13. Προτάσεις για Μελλοντική Επέκταση – Βελτίωση.....	71
14. Βιβλιογραφία-Πηγές-Αναφορές.....	74
15. Παράρτημα Α - GNU Free Documentation License.....	76

## Πρόλογος

Η Java ως γνωστόν ήδη εδώ και μία δεκαετία και παραπάνω έχει μπει μέσα στα σπίτια όσων διαθέτουν Ηλεκτρονικό Υπολογιστή και Internet. Ένα από τα βασικά πλεονεκτήματα της Java έναντι των περισσότερων άλλων γλωσσών είναι η ανεξαρτησία του λειτουργικού συστήματος και πλατφόρμας. Τα προγράμματα που είναι γραμμένα σε Java τρέχουν ακριβώς το ίδιο σε Windows, Linux, Unix και Macintosh (σύντομα θα τρέχουν και σε Playstation καθώς και σε άλλες κονσόλες παιχνιδιών) χωρίς να χρειαστεί να ξαναγίνει μεταγλώττιση (compiling) ή να αλλάξει ο πηγαίος κώδικας για κάθε διαφορετικό λειτουργικό σύστημα. Για να επιτευχθεί όμως αυτό χρειαζόταν κάποιος τρόπος έτσι ώστε τα προγράμματα γραμμένα σε Java να μπορούν να είναι «κατανοητά» από κάθε υπολογιστή ανεξάρτητα του είδους επεξεργαστή (Intel x86, IBM, Sun SPARC, Motorola) αλλά και λειτουργικού συστήματος (Windows, Unix, Linux, BSD, MacOS). Ο λόγος είναι ότι κάθε κεντρική μονάδα επεξεργασίας κατανοεί διαφορετικό κώδικα μηχανής. Ο συμβολικός (assembly) κώδικας που εκτελείται σε Windows είναι διαφορετικός από αυτόν που εκτελείται σε έναν υπολογιστή Macintosh. Η λύση

δόθηκε με την ανάπτυξη της Εικονικής Μηχανής (Virtual Machine ή VM ή EM στα ελληνικά).

Το παρόν παίγνιο σχεδιάστηκε εξ' ολοκλήρου με Java για τους παραπάνω λόγους.

Θα έλεγε κανείς λοιπόν 'ότι είναι μία εφαρμογή για PC η οποία ως στόχο της έχει την ανάδειξη ενός μικρού δείγματος των τεχνολογιών που προσφέρονται από τα συνεχώς αναπτυσσόμενα API που διαθέτει η Java ως τεχνολογία μέσα από ένα παιχνίδι προορισμένο για ψυχαγωγικούς σκοπούς.

Η εν λόγω εφαρμογή συνολικά είναι ένα κατά κύριο λόγο δισδιάστατο παιχνίδι πλατφόρμας όπου γίνεται συνδυασμός και τρισδιάστατων τεχνολογιών κυρίως στο αρχικό menu του παιχνιδιού όπως επίσης και κατά τη διάρκεια μέσω τεχνικών 3D απεικόνισης.

## Γενική Περιγραφή Παιχνιδιού - Περίληψη

Το Cartoon Land λοιπόν γενικά είναι ένα κατά κύριο λόγο δισδιάστατο παιχνίδι το οποίο βασίζεται στους χαρακτήρες του διάσημου τηλεοπτικού σόου South Park. Ο στόχος του χαρακτήρα cartoon που επιλέγεται αρχικά από το χρήστη είναι ευθύς και αρκετά απλός. Αναλόγως με το σε τι λειτουργία τίθεται το παιχνίδι, δηλαδή single player ή multiplayer mode, ο παίκτης προσπαθεί να διέλθει διαμέσου των επιπέδων του παιχνιδιού προσπαθώντας να τερματίσει χωρίς να χάσει όλες τις ζωές του και μαζεύοντας όσο το δυνατόν περισσότερα χρήματα. Αν και κάπως υλιστική η όλη προσέγγιση του στόχου του παίκτη ( σήμα κατατεθέν των Platform Games της δεκαετίας του 80 και 90 ), πολλές και διαφορετικές παραλλαγές θα μπορούσαν να γίνουν στη φιλοσοφία και το σενάριο ψυχαγωγίας του χρήστη που παίζει το παιχνίδι ως επέκταση ( βλέπε Προτάσεις για μελλοντική επέκταση - βελτίωση) .Αυτό φυσικά ξεφεύγει από τα όρια της συγκεκριμένης πτυχιακής η οποία ως κύριο στόχο έχει την ανάδειξη των δυνατοτήτων-τεχνολογιών της java μέσα από ένα απλό μη εμπορικά σχεδιασμένο παίγνιο. Τον ίδιο στόχο έχει ο εκάστοτε χρήστης και σε περίπτωση που επιλέξει να παίζει το

παιχνίδι σε τοπικό δίκτυο ( LAN ), με τη διαφορά ότι ουσιαστικά προσπαθεί να σκοτώσει όσο το δυνατόν περισσότερους αντιπάλους και να συλλέξει όσο το δυνατόν περισσότερα gadgets περνώντας την πίστα ώστε να αποκτήσει αυτός το μεγαλύτερο σκορ.

Όταν ο χρήστης φτάσει στο τέλος της τελευταίας πίστας εμφανίζεται το σκορ του, ο αριθμός των αντιπάλων που κατάφερε να σκοτώσει και τέλος γράφει η ίδια η εφαρμογή στον κατάλογο που βρίσκεται τα στατιστικά στοιχεία αυτά ώστε ο χρήστης να έχει πρόσβαση και αφού τελειώσει και βγει από το παιχνίδι.

### **Game Overview - Summary**

The Cartoon Land is generally a predominantly two-dimensional game based on the characters of the famous television show South Park.

The aim of the cartoon character originally selected by the user interface is straightforward and quite simple. Depending on the operation of the game, namely single player or multiplayer mode, the player tries to pass through the levels of the game trying to finish without losing any lives and collecting as much money as possible. Although this is somehow a materialistic approach of the overall objective of the player (trademark of Platform Games of the 80 and 90), many different variations could have been made in philosophy and entertainment scenario of a user playing the game as an extension (see Proposals for future expansion - improving). But this goes far beyond the bounds of this thesis project, which primarily aims to highlight opportunities, java technologies through a simple non-commercial game design game. The same goal takes place as far as the user is concerned even if the game is chosen to be played in local area network (LAN), except that it actually tries to kill as many enemies and collect as many gadgets coming through each track to become the highest scorer of the two users.

When the user reaches the end of the last track his score appears, the number of opponents who managed to kill and then the application itself writes on the same directory it's been the statistical data that the user has access at, and having finished by that and it exits game.

## Έλεγχος-Κουμπιά Πλοήγησης

Τα κουμπιά για το χειρισμό της εν λόγω εφαρμογής είναι τα εξής :

Αριστερά	Left	- Left Arrow
Δεξιά	Right	- Right Arrow
Πήδημα	Jump	- Space
Εκσφενδόνιση	Knife	- Left Control
Παύση	Pause	- P
Έξοδος	Escape	- Esc

## Στόχοι - Όραμα

Οι στόχοι της συγκεκριμένης εφαρμογής αφορούν καθαρά την παράδοση της ως πτυχιακή εργασία και επιπλέον τη μετεξέλιξη του συγκεκριμένου παιχνιδιού ( βλέπε Προτάσεις για μελλοντική επέκταση – βελτίωση ). Πολλές και ποικίλες θα μπορούσαν να είναι οι προσθήκες στη μετεξέλιξη του συγκεκριμένου παιχνιδιού και θα περιγραφούν στην ενότητα *βελτίωση* που αναλυτικά.

## Κυρίως Κορμός Παιχνιδιού

Ο κυρίως κορμός του συγκεκριμένου παιχνιδιού αποτελείται από δύο μέρη. Το πρώτο μέλος αποτελεί το Menu το οποίο είναι αυτό από το οποίο ο χρήστη μπορεί να επιλέξει μεταξύ τεσσάρων επιλογών :

START όπου αρχίζει το παιχνίδι

OPTIONS όπου συμπληρώνονται τα στοιχεία δικτύου

QUIT όπου γίνεται έξοδος από το παιχνίδι

HELP όπου εμφανίζεται το μενού βοήθειας

Το δεύτερο μέλος αποτελεί το κυρίως παιχνίδι το οποίο ακολουθεί το μενού και είναι και το τελευταίο σε σειρά ώστε να ολοκληρωθεί το παιχνίδι. Στο μέρος αυτό ο χρήστης-παίκτης παίζει το παιχνίδι προσπαθώντας να φτάσει στο τέλος.

Σε αυτά τα 2 μέλη έρχονται να προστεθούν και 2 ακόμη προεκτάσεις. Η μία είναι το Splash Screen που εμφανίζεται στην εκκίνηση του παιχνιδιού και η δεύτερη προέκταση είναι το ενδιάμεσο από τα επίπεδα του κυρίως παιχνιδιού όπου εμφανίζονται τα στατιστικά στοιχεία του επιπέδου από το οποίο μόλις διήλθε ο παίκτης. Τα στατιστικά αυτά στοιχεία περιλαμβάνουν το σκορ, τις ζωές που του απέμειναν και τους αντιπάλους που έχει σκοτώσει συνολικά.

Γίνεται αντιληπτό λοιπόν εύκολα από το κορμό του παιχνιδιού πως πρόκειται για μία απλή σχεδίαση του σεναρίου και της γενικότερης φιλοσοφίας του παιχνιδιού. Αυτό είναι επίσης κάτι το οποίο θα μπορούσε εύκολα σε μελλοντικές επανεκδόσεις του παιχνιδιού να αναπροσαρμοστεί σε μία φιλοσοφία οικολογικού τύπου για παράδειγμα ώστε ο παίκτης να μαζεύει φρούτα και να σώζει δένδρα.

## **Περιγραφή Ανάλυση Κανόνων Παιχνιδιού**

Στην ενότητα αυτή θα προσπαθήσουμε να εξηγήσουμε τους κανόνες παιχνιδιού όσον αφορά τη σχεδίασή του.



## Κανόνες Κίνησης

Οι γενικοί κανόνες κίνησης αφορούν την εμφάνιση του παίκτη στο εκάστοτε επίπεδο και την κίνηση του σε σχέση με τα υπόλοιπα αντικείμενα που βρίσκονται εκεί και επιπλέον την κίνηση των λοιπών ζωντανών αντικειμένων όπως των αντιπάλων και το gadgets της κάθε πίστας.

Ο παίκτης λοιπόν αποτελεί μία ξεχωριστή οντότητα που έχει τη δυνατότητα καταρχήν να πηδάει έπειτα από εντολή του χειριστή ή σε περίπτωση που σκοτώσει κάποιον αντίπαλο. Επίσης ο παίκτης μπορεί να κινηθεί αριστερά, δεξιά, να ανέβει και να σταθεί στα πλακάκια ( tiles ) που σχηματίζουν την πίστα. Ο παίκτης επιπλέον είναι ένα από τα αντικείμενα στο παιχνίδι που δεν είναι πετούμενα. Συνεπώς και διέπεται και αυτός όπως και τα υπόλοιπα μη πετούμενα αντικείμενα από το νόμο της βαρύτητας.

Τώρα ως προς τη γενική κίνηση των αντιπάλων του παίκτη. Αναλόγως το αν είναι πετούμενοι ή μη καθορίζεται διαφορετική ταχύτητα και διαφορά στη βαρύτητα.

Υπάρχει επίσης κίνηση αριστερά και δεξιά των υπάρξεων αυτών.

Όσον αφορά τα μαχαίρια που μπορεί να πετάξει ο παίκτης, αυτά κινούνται αριστερά, δεξιά και υπόκεινται στο νόμο της βαρύτητας.

Τέλος τα διάφορα gadgets των επιπέδων παραμένουν στο σημείο που καθορίζονται να βρίσκονται από την αρχή σύμφωνα με το χάρτη της εκάστοτε πίστας.

## Κανόνες Αλληλεπίδρασης

Γενικά τα πάντα σε ένα παιχνίδι είναι θέμα αλληλεπίδρασης των μεταξύ στοιχείων.

#### Παίκτης – Πλακάκια

Ο παίκτης αλληλεπιδρά με τα πλακάκια της πίστας ως εξής. Όταν βρεθεί να ακουμπά οριζοντίως ή καθέτως με πλακάκι εμποδίζεται η κίνηση προς την κατεύθυνση εκείνη.

#### Παίκτης – Αντίπαλοι

Ο παίκτης αλληλεπιδρά με τους αντίπαλους της πίστας ως εξής. Όταν βρεθεί να ακουμπά οριζοντίως με αντίπαλο σκοτώνεται και χάνει μία από τις ζωές του και εμφανίζεται ξανά από ψηλά.

Όταν βρεθεί να ακουμπά καθέτως με αντίπαλο τον σκοτώνει και πηδά αυτομάτως μία φορά.

#### Παίκτης – Gadgets

Ο παίκτης αλληλεπιδρά με τα gadgets της πίστας ως εξής. Όταν βρεθεί να ακουμπά οριζοντίως ή καθέτως με οποιοδήποτε gadget το «κατακτά» αποκτώντας πόντους από αυτό εξαφανίζοντάς το από το χάρτη και επίσης περνάει σε επόμενο επίπεδο παίρνοντας ένα συγκεκριμένο gadget.

#### Αντίπαλοι – Πλακάκια

Οι αντίπαλοι αλληλεπιδρούν με τα πλακάκια της πίστας ως εξής. Όταν βρεθούν να ακουμπούν οριζοντίως ή καθέτως με πλακάκι εμποδίζεται η κίνηση τους προς την κατεύθυνση εκείνη.

#### Αντίπαλοι – Gadgets

Οι αντίπαλοι αλληλεπιδρούν με τα Gadgets της πίστας ως εξής. Όταν βρεθούν να ακουμπούν οριζοντίως ή καθέτως με Gadget δεν υπάρχει καμία απολύτως τίποτα. Απλώς λοιπόν συνυπάρχουν στην ίδια πίστα.

#### Αντίπαλοι – Μαχαίρια

Οι αντίπαλοι αλληλεπιδρούν με τα Μαχαίρια της πίστας ως εξής. Όταν βρεθούν να ακουμπούν οριζοντίως ή καθέτως με μαχαίρια ο αντίπαλος σκοτώνεται και εξαφανίζεται από την πίστα.

## Κανόνες Εξωγενείς

Χρόνος

Χρήστης

### Χρόνος

Ο χρόνος αφορά σε έναν μετρητή ορισμένο από την ίδια την εφαρμογή στην αρχή κάθε επιπέδου μετρητής αυτός έχει σχεδιαστεί να αρχίζει από το 60" και να τερματίζει στο 0 μειούμενος κατά 1 συνεχώς από τη στιγμή που εμφανίζεται το κάθε επίπεδο. Κατά τη διάρκεια της συνεχούς μείωσης του μετρητή ο παίκτης μπορεί να κινηθεί κανονικά και να παίξει το παιχνίδι.

Στην περίπτωση που τελειώσει ο χρόνος πριν να φτάσει ο παίκτης στο τέλος της εκάστοτε πίστας το παιχνίδι τελειώνει και ο παίκτης χάνει όλες του τις ζωές και τους πόντους που περισυνέλεξε ως τότε.

### Χρήστης

Ο χρήστης είναι ο κύριος της εφαρμογής. Αυτό σημαίνει ότι επηρεάζει πιο σημαντικά από οποιοδήποτε άλλο παράγοντα την εξέλιξη του παιχνιδιού. Κατά πρώτον λοιπόν επηρεάζει με την πλοήγηση που κάνει στον παίκτη και τον οδηγεί διαμέσου της πίστας στο να σκοτώσει πιθανώς κάποιον αντίπαλο είτε να συλλέξει κάποιο sprite gadget. Επιπλέον επηρεάζει τη συνολική εξέλιξη του παιχνιδιού πατώντας κάνοντας παύση στο παιχνίδι σταματώντας το χρόνο και την οποιαδήποτε κίνηση στο εσωτερικό του επιπέδου, όπως επίσης και τη μουσική. Τέλος, κατά παρόμοιο τρόπο επηρεάζει βγαίνοντας από το κυρίως παιχνίδι στο μενού του παιχνιδιού. Έτσι, κάνει παύση της μουσικής του επιπέδου, του χρονομετρητή που μειώνεται και των κινήσεων που συμβαίνουν τη στιγμή εκείνη στο επίπεδο · παράλληλα όμως εισέρχεται στο μενού του παιχνιδιού απ' όπου μπορεί να πλοηγηθεί και να εκτελέσει κινήσεις όπως επανείσοδος στο παιχνίδι, βοήθεια, ολική έξοδος.

## Περιγραφή Κλάσεων

Στο ενότητα αυτή θα γίνει μία προσπάθεια να περιγραφούν οι κλάσεις που αποτελούν το συγκεκριμένο παιχνίδι όσο το δυνατόν πιο αφαιρετικά.

Τα πακέτα τα οποία αποτελούν και την πιο γενικότερη προσέγγιση ενοτήτων του παιχνιδιού είναι τα εξής :

Controller

Game

IO

Menu

ScreenGraphics

SendReceive

Sounds

Splash3d2

Sprites

Statistics

### Controller

Το πακέτο Controller περιέχει την τάξη Core και είναι αυτό που είναι υπεύθυνο για την κύρια “λούπα” που συνεχώς τρέχει κατά τη διάρκεια του παιχνιδιού. Ελέγχει την γενική πορεία του, τις όποιες παύσεις, την αρχικοποίηση των γραφικών σε πλήρη οθόνη, την φόρτωση των πηγών που είναι απαραίτητες

### Game

Το πακέτο Game είναι υπεύθυνο για την βασική κλάση MainGame η οποία διαχειρίζεται συνολικά το παιχνίδι. Εμπεριέχει όλες αυτές τις τάξεις η οποίες είναι κατά ένα τρόπο συνυπόλογες στη διεκπεραίωση της διαδικασίας αυτής όπως: η κλάση TileMap που περιέχει τα στοιχεία των επιπέδων – χαρτών απαραίτητα στην κύρια διαχείριση του παιχνιδιού, την κλάση Resource η οποία αναλαμβάνει το δύσκολο έργο της καθ' εαυτής φόρτωσης των πηγών και τέλος την κλάση

Renderer που είναι αυτή η οποία διεκπεραιώνει την απεικόνιση όλων των στοιχείων της εφαρμογής στη οθόνη στις σωστές θέσεις.

## IO

Το πακέτο IO είναι το πακέτο για όλες τις εισόδους και εξόδους από και προς την εφαρμογή. Περιέχει την κλάση Action που ορίζει μία οποιαδήποτε πράξη – κίνηση του παίκτη κατά τον σχεδιασμό του παιχνιδιού. Και την κλάση IOHandler που είναι ο διαχειριστής των κινήσεων αυτών αντιστοιχώντας τις σε κουμπιά του πληκτρολογίου

## Menu

Το πακέτο Menu είναι το πακέτο που αφορά το κύριο μενού της εφαρμογής. Διαθέτει τις κλάσεις που το αποτελούν, το ορίζουν, το αρχικοποιούν και το σχεδιάζουν. Όπως επίσης και όλες εκείνες τις λειτουργίες που είναι απαραίτητες για τη διαχείριση του.

## ScreenGraphics

Το πακέτο ScreenGraphics είναι το πακέτο που περιέχει τη βασική κλάση Sprite που αποτελεί τον αρχικό ορισμό οποιασδήποτε κινούμενης οντότητας στο εσωτερικό των επιπέδων του παιχνιδιού. Η κίνηση της κάθε οντότητας αυτής όμως σε βαθύτερη ανάλυση είναι υπόθεση της κλάσης Animation η οποία και ορίζει την ίδια την κίνηση τους εναλλάσσοντας εικόνες ( frames ) στις οποίες αντιστοιχεί κάποιος χρόνος σε χιλιοστά του δευτερολέπτου. Τέλος, στο πακέτο αυτό υπάρχει και η κλάση Screencontroler που ορίζει, καθορίζει και εμφανίζει τα οποιαδήποτε γραφικά σε πλήρη οθόνη.

## SendReceive

Το πακέτο SendReceive είναι το πακέτο που αναλαμβάνει για τη συνολική διεκπεραίωση της δικτυακής εργασίας που ως μοναδικό στόχο έχει την επικοινωνία των δύο σε αριθμό εφαρμογών που θα λειτουργούν σε πλαίσιο συνεργατικού παιχνιδιού ( multiplayer mode ). Η μία κλάση Sender είναι αυτή που αναλαμβάνει να στέλνει συνεχώς δεδομένα που αφορούν την εφαρμογή στη οποία και περιλαμβάνεται ώστε να γίνονται αντιληπτά από την άλλη. Και η κλάση Receiver είναι αυτή που αναλαμβάνει να εκτελέσει το έργο της

αποκωδικοποίησης των όσων λαμβάνονται από την άλλη εφαρμογή, όπως επίσης και της εκτελέσεως οποιασδήποτε ενεργείας απαραίτητης σύμφωνα με όσα έχει λάβει.

## Sounds

Το πακέτο Sounds είναι το πακέτο που ελέγχει τον ορισμό καταρχήν και το χειρισμό κατά δεύτερον των μουσικών κομματιών που «τρέχουν» κατά την εμφάνιση του αρχικού μενού της εφαρμογής, όπως και στο κυρίως παιχνίδι. Περιέχει την κλάση WavePlayer η οποία είναι και αυτή που αποτελεί το ίδιο το ρεύμα του ήχου.

## Splash3d2

Το πακέτο Splash3d2 είναι το πακέτο που διαθέτει τις δύο απαραίτητες κλάσεις για τη δημιουργία και την εκτέλεση της οθόνης εκκίνησης της εφαρμογής ( Splash Screen ). Στην τάξη Splash3d ορίζεται και αρχικοποιείται το ίδιο το background του Splash Screen. Έπειτα στην κλάση TestSplash3D κατασκευάζεται ο γράφος του Splash Screen με όλα τα στοιχεία που τον αποτελούν και καθορίζεται ο χρόνος διάρκειας της οθόνης εκκίνησης.

## Sprites

Το πακέτο Sprites είναι το πακέτο που διαθέτει όλες τις υποκλάσεις της βασικής υλοποίησης ενός Sprite. Περιέχει λοιπόν την τάξη Existence η οποία αφορά οποιαδήποτε ύπαρξη της εφαρμογής που έχει κάποια κίνηση ( animation ) και επηρεάζεται από τις υπόλοιπες υπάρξεις γύρω της. Από τις τάξεις όπως οι : Crawler, Fly , Player που αποτελούν το έμφυχο δυναμικό του παιχνιδιού, και οι : Bullet, Gadget που είναι οι άψυχες animations στο παιχνίδι, οι οποίες αλληλεπιδρούν όμως με τις υπόλοιπες.

## Statistics

Το πακέτο Statistics είναι το πακέτο που αφορά στην κράτηση και διαχείριση των στατιστικών στοιχείων του κάθε παίκτη. Η κλάση StatsCollector λοιπόν συνιστά στην λήψη των στατιστικών στοιχείων αυτών.

## Ανάλυση Κλάσεων

Όσον αφορά στην ανάλυση των κλάσεων που αποτελούν την εφαρμογή, θα επικεντρωθούμε στην συγκέντρωση όλων των ουσιαστικών στοιχείων μερών καθεμίας από αυτές επί του κώδικα και της λειτουργικότητας που προσφέρει γενικότερα.

1. Στο πακέτο Controller λοιπόν η κλάση Core αποτελεί όπως προείπαμε την κλάση που είναι υπεύθυνη για την κύρια “λούπα” που συνεχώς τρέχει κατά τη διάρκεια του παιχνιδιού. Παρακάτω θα περιγραφούν πιο αναλυτικά τα βασικότερα στοιχεία – μέρη της.
- Αρχικοποίηση των δυνατών αναλύσεων βάση σχεδιασμού του παιχνιδιού.

```
private static final DisplayMode POSSIBLE_MODES[] = {  
    new DisplayMode(300, 200, 16, 0), //here change  
    new DisplayMode(800, 600, 16, 0),  
    new DisplayMode(800, 600, 32, 0),  
    new DisplayMode(800, 600, 24, 0),  
    new DisplayMode(640, 480, 16, 0),  
    new DisplayMode(640, 480, 32, 0),  
    new DisplayMode(640, 480, 24, 0),  
    new DisplayMode(1024, 768, 16, 0),  
    new DisplayMode(1024, 768, 32, 0),  
    new DisplayMode(1024, 768, 24, 0),  
};
```

Στο σημείο αυτό διατίθενται οι πιθανές αναλύσεις που θα συγκριθούν μία προς μία με την τρέχουσα ανάλυση του συστήματος στο οποίο τρέχει η εφαρμογή έτσι ώστε αν ταιριάζουν να επιλεγεί η κατάλληλη προς εμφάνιση στο κυρίως παιχνίδι.

- Παροχή μεθόδων παύσης και συνέχισης της κύριας «λούπας» του παιχνιδιού.

### Pause,Unpause

- Υλοποίηση της run μεθόδου αρχικοποιεί την οθόνη και τα οποιαδήποτε αντικείμενα, όπως επίσης και καλεί την gameloop που είναι η κύρια «λούπα» του παιχνιδιού η οποία συνεχώς τρέχει.
  - Παροχή της Update και της draw μεθόδου που υλοποιούνται από την κλάση MainGame που διαχειρίζεται το παιχνίδι και κληρονομεί την Core κλάση και ολόκληρη τη συμπεριφορά της.
2. Στο πακέτο Game λοιπόν η κλάση MainGame αποτελεί τον βασικό διαχειριστή του παιχνιδιού.Παρακάτω θα περιγραφούν τα βασικά της σημεία.
- Αρχικοποίηση του εικονιδίου της μπάρας εργασίας.
  - Αρχικοποίηση του Splash Screen
  - Αρχικοποίηση του Menu
  - Αρχικοποίηση των Μουσικών του παιχνιδιού.
  - Αρχικοποίηση του Χρονομετρητή των επιπέδων
  - Αρχικοποίηση του Resource που φορτώνει όλες τις πηγές.
  - Αρχικοποίηση του Renderer που φορτώνει όλες τις πηγές.
  - Φόρτωση του πρώτου χάρτη.
  - Κλήση της μεθόδου connect η οποία αρχικοποιεί έναν αποστολέα Sender και ένα λήπτη Receiver για το δίκτυο σε multiplayer mode.
  - Αρχικοποίηση των κινήσεων του παίκτη και αντιστοίχισή τους σε κλειδιά του πληκτρολογίου.
  - Κλήση της io\_check μεθόδου για τον έλεγχο των εισόδων από το πληκτρολόγιο του χρήστη.
  - Υλοποίηση της draw μεθόδου που καλεί τον renderer για την απεικόνιση στην οθόνη



- Υλοποίηση της μεθόδου `gettilecollision` όπου εντοπίζεται η οποιαδήποτε σύγκρουση ενός `tile` με ένα `sprite` και επιστρέφεται η θέση της σύγκρουσης
- Υλοποίηση της μεθόδου `collision_found` όπου εντοπίζεται αν υπάρχει σύγκρουση ενός `sprite` με ένα άλλο ζωντανό.
- Υλοποίηση της μεθόδου `getSpriteCollision` όπου εντοπίζεται και επιστρέφεται το `sprite` το οποίο συγκρούεται με το επισυμένον `sprite`.
- Υλοποίηση της μεθόδου `update` γίνεται η διαχείριση της ενημέρωσης των κινήσεων των `Sprites` του παιχνιδιού, όπως επίσης και η αποστολή των απαιτούμενων για την ενημέρωση του συμπαίκτη στοιχείων στο δίκτυο σε λειτουργία `multiplayer`, και τέλος παράλληλα γράφονται σε αρχεία τα στατιστικά των παικτών.
- Υλοποίηση της μεθόδου `updateexistence` που αναλαμβάνει την επί θεωρητικού επιπέδου ενημέρωση των  $\chi$  και  $\psi$  σημείων των υπάρξεων στο παιχνίδι όπως και των καταστάσεων τους, των συγκρούσεών τους.
- Υλοποίηση της μεθόδου `updatebullet` που αναλαμβάνει την επί θεωρητικού επιπέδου ενημέρωση των  $\chi$  και  $\psi$  σημείων των μαχαιριών στο παιχνίδι όπως και των καταστάσεων τους, των συγκρούσεών τους.
- Υλοποίηση της μεθόδου `acquiregadget` που αναλαμβάνει την διαχείριση και εκτέλεση διαφόρων εκβάσεων του παιχνιδιού καθώς ο παίκτης αποκτά ένα οποιοδήποτε `gadget`, όπως για παράδειγμα η συνέχιση σε επόμενο επίπεδο ή η απόδοση πόντων.
- Υλοποίηση της `checkudp` όπου ελέγχεται αν έχει επιτευχθεί η επικοινωνία μεταξύ του δύο συμπαικτών σε λειτουργία `multiplayer` συνεργατικού παιχνιδιού.

Η κλάση `TileMap` είναι η επόμενη προς ανάλυση κλάση του πακέτου `Game`.

Η κλάση αυτή αποτελεί τον χάρτη της κάθε πίστας ορισμένο σε έναν δισδιάστατο πίνακα εικόνων. Εμπεριέχει ακόμη τα `sprites` που μπαίνουν στις πίστες σε μια λίστα.

Τα βασικά της στοιχεία είναι τα εξής:

- Ορίζει τον χάρτη εικόνων της κάθε πίστας, με συγκεκριμένο πλάτος και ύψος σε `tiles` – πλακάκια, βάση ενός αρχείου κειμένου κατά τη δημιουργία του.

- Ορίζει την λίστα των sprites του εκάστοτε χάρτη κατά τη δημιουργία του.
- Ορίζει τις μεθόδους που επιστρέφουν το πλάτος και το ύψος του χάρτη σε tiles – εικόνες του δισδιάστατου πίνακα που ορίζει τον χάρτη.
- Ορίζει τις μεθόδους τοποθέτησης και ανάκτησης θέσης σε tiles συγκεκριμένης εικόνας – tile.
- Ορίζει τις μεθόδους αναφοράς σε συγκεκριμένα sprites όπως αυτά των παικτών, των μαχαιριών,
- Ορίζει τις μεθόδους προσθήκης και διαγραφής ενός οποιουδήποτε sprite από τη λίστα των sprites του χάρτη.
- Ορίζει τις μεθόδους ανάκτησης των sprites, εκτός του παίκτη και των μαχαιριών, μέσω ενός επαναλήπτη που διατρέχει την λίστα των sprites αυτών. Όπως επίσης και την επιστροφή του αριθμού των sprites στη λίστα αυτή του χάρτη.

Η κλάση `Resourceer` είναι η επόμενη προς ανάλυση κλάση του πακέτου `Game`.

Η κλάση αυτή αποτελεί τον φορτωτή και ελεγκτή των πηγών του παιχνιδιού, όπως τα πλακάκια και τα sprites.

Τα βασικά της στοιχεία είναι τα εξής:

- Υλοποιεί την μέθοδο φόρτωσης των tiles του χάρτη μέσω της μεθόδου `loadTileImages` η οποία ελέγχει την ύπαρξη ορισμένων εικόνων των tiles και τις φορτώνει σε λίστα.
- Υλοποιεί την μέθοδο `loadExistenceSprites` η οποία προσθέτει στην πρώτη οριζόντια θέση ενός δισδιάστατου πίνακα τις εικόνες των sprites του χάρτη που αργότερα θα αποτελέσουν και τα animations.  
Στις επόμενες αντίστοιχες οριζόντιες θέσεις προσθέτονται οι ίδιες εικόνες αλλά σε διαφορετική γωνία ( ανάποδες στο  $x\chi'$  και αντίστοιχα στον  $y\gamma'$  ).

- Ορίζει επίσης τα animations των sprites. Σε αυτά οποία αναθέτει τις αντίστοιχες εικόνες για κάθε ένα από αυτά και κάποιο ορισμένο χρόνο σε χιλιοστά του δευτερολέπτου.
- Ορίζει τα sprites φορτώνοντας τους το αντίστοιχο animation το οποίο αποτελείται από τις αντίστοιχες εικόνες που έχουν προηγουμένως φορτωθεί, τόσο για τα sprites του παίκτη όσο και για τα υπόλοιπα όπως υπάρξεις, gadgets κ.α.
- Παρέχει ακόμη την μέθοδο κατασκευής, φόρτωσης, και επιστροφής του εκάστοτε χάρτη η οποία διαβάζει τον χάρτη σχεδιασμένο σε αρχείο κειμένου χαρακτήρων και προσθέτει - φορτώνει σε λίστα τα αντίστοιχα sprites και tiles.
- Περιέχει επιπλέον την μέθοδο που επιστρέφει τον αριθμό του χάρτη ο οποίος είναι φορτωμένος τη στιγμή αυτή.
- Τέλος, φορτώνει παίκτη και μαχαίρια ουσιαστικά δημιουργώντας ένα αντίγραφο sprite το οποίο κατ' επέκταση αντιγράφει και τα animation του sprite του οποίου καλείται να αποτελέσει αντίγραφο.

Η κλάση `Renderer` είναι η επόμενη προς ανάλυση κλάση του πακέτου `game`. Η κλάση αυτή ζωγραφίζει τον χάρτη των tiles στην οθόνη, όπως επίσης και όλα τα sprites και ένα οπίσθιο φόντο κεντραρισμένο γύρω από τη θέση που βρίσκεται ο παίκτης.

Τα βασικά της στοιχεία είναι τα εξής:

- Ορίζει το μέγεθος κάθε tile σε 64 pixels ώστε να γίνονται οι οποιοδήποτε υπολογισμοί βάση του νούμερου αυτού.
- Ορίζει το μέγεθος σε bits που απαιτείται, του μεγέθους 64 pixels του κάθε tile, είναι 6 bit.

- Ορίζει τις μεθόδους μετατροπής μιας θέσης tile σε θέση pixel και αντιστρόφως. Ωστε για παράδειγμα η αναφορά στο πρώτο tile να αντιστοιχεί στο εύρος τιμών pixel από 0 έως και 63 κοκ και αντίστροφα.
- Ορίζει το οπίσθιο φόντο στο παιχνίδι ως μία εικόνα.
- Υλοποιεί την βασική για τη διαδικασία της απόδοσης στην οθόνη των οποιονδήποτε γραφικών. Η μέθοδος αυτή θα μπορούσε να χαρακτηριστεί ως ένας προβολέας εικόνων βάσει υπολογισμένων θέσεων για αυτές. Αποδίδει στην οθόνη λοιπόν «τα πάντα» sprites , tiles , backgrounds αφού τα ανακτά από τον θεωρητικά ορισμένο χάρτη που έχει προηγουμένως οριστεί και στον οποίον έχουν φορτωθεί όλα αυτά μετατρέποντας τις θέσεις των tiles σε θέσεις pixels.
- Ζωγραφίζει επιπλέον το background των επιπέδων με έναν τρόπο που το κάνει να μοιάζει πως κινείται ( Συνυπολογίζοντας την απόσταση του παίκτη από τα άκρα του χάρτη και του μεγέθους της ίδιας της εικόνας του background του χάρτη επί της μετακίνησης του παίκτη).
- Ζωγραφίζει τον αντίστροφο χρονομετρητή.
- Ζωγραφίζει τις οποιεσδήποτε χρειαζόμενες πληροφορίες του παίκτη, όπως η ζωές που του έχουν απομείνει ή τα στατιστικά που εμφανίζονται μετά το τέλος κάθε επιπέδου

Όλα αυτά αποδίδονται όπως αναφέραμε και παραπάνω στην οθόνη βάση αυστηρού υπολογισμού θέσεις έχοντας υπόψη το mode ( single,multi player ), τη θέση του παίκτη, το μέγεθος σε pixels του χάρτη κ.α.

3. Στο πακέτο SendReceive λοιπόν δύο είναι οι βασικές για την εξυπηρέτηση της λειτουργικότητας του κλάσεις, η Sender και Receiver. Παρακάτω θα περιγραφούν αυτές και τα βασικά τους σημεία.

Η κλάση Sender αποτελεί τον βασικό αποστολέα πληροφορίας στο δίκτυο, σε λειτουργία συνεργατικού παιχνιδιού, στην παράλληλη εφαρμογή.

Τα βασικά της στοιχεία είναι τα εξής :

- Δημιουργία ενός datagram socket για αποστολή datagram packets μέσω αυτού.
- Ορισμός διεύθυνσης και πόρτας για την αποστολή πακέτων.
- Υλοποίηση μεθόδου `sendAway` για αποστολή δεδομένων μέσω `DataOutputStream`
- Γράψιμο χαρακτήρων στο `DataOutputStream`
- Αντιστοίχιση `DatagramPacket` σε συγκεκριμένη διεύθυνση και πόρτα.
- Αποστολή πακέτου μέσω του socket αυτού.
- Εμπεριέχει την μέθοδο `isActive` η οποία ελέγχει το κατά πόσο η εφαρμογή στην οποία θα κληθεί είναι active.  
Ο διαχωρισμός μεταξύ active και inactive είναι απαραίτητος για τη διεκπεραίωση ορισμένων ενδολειτουργιών.

Η κλάση `Receiver` αποτελεί τον βασικό λήπτη πληροφορίας στο δίκτυο, σε λειτουργία συνεργατικού παιχνιδιού, από την παράλληλη εφαρμογή.

Τα βασικά της στοιχεία είναι τα εξής :

- Ορισμός και κατασκευή του λήπτη πληροφορίας.
- Αρχικοποιείται ένα `DatagramSocket` με έναν αριθμό πόρτας στην οποία θα «ακούει» και θα λαμβάνει πληροφορίες η εφαρμογή.
- Αρχικοποιεί επιπλέον αναφορές στο βασικό διαχειριστή της εφαρμογής `MainGame`. Όπως επίσης και αναφορές στον παίκτη και στα υπόλοιπα `Sprite` αντικείμενα του παιχνιδιού.
- Υλοποιεί την μέθοδο `run`, η οποία «τρέχει» συνεχώς σε λειτουργία συνεργατικού παιχνιδιού και μέσω του `datagram socket` και της `receive` μεθόδου διαβάζει τα εισερχόμενα `datagram packets` και πράττει ανάλογα με το περιεχόμενό

τους όπως η ενημέρωση των σημείων και των ταχυτήτων των sprites του παίκτη και των μαχαιριών.

4. Στο πακέτο IO δύο είναι οι βασικές για την εξυπηρέτηση της λειτουργικότητας του κλάσεις, η Action και IOHandler. Παρακάτω θα περιγραφούν αυτές και τα βασικά τους σημεία.

Η κλάση Action αποτελεί μία κλάση αφηρημένη για τις εισόδους του χρήστη στην εφαρμογή. Οι κινήσεις του χρήστη αντιστοιχούνται σε κουμπιά του πληκτρολογίου μέσω της IOHandler κλάσης. Τα βασικά στοιχεία της Action είναι τα εξής :

- Ορίζει τις διάφορες καταστάσεις των Actions όπως Normal, Initial, Released, Pressed, Waiting for Release
- Δομεί μία κίνηση και της αναθέτει μία συμπεριφορά και ένα όνομα.
- Υλοποιεί τη μέθοδο reset που αναθέτει την κατάσταση released για μία κίνηση ώστε να μοιάζει πως δεν έγινε – πατήθηκε καμιά φορά.
- Υλοποιεί τη μέθοδο tap η οποία κάνει την κίνηση να γίνει εικονικά, σα να πατήθηκε το πλήκτρο που της αντιστοιχεί, και έπειτα απελευθερώνει εικονικά το πλήκτρο.
- Υλοποιεί τη μέθοδο press που αναλαμβάνει να καλέσει μία φορά την press(1), δηλαδή «πάτα 1 φορά το πλήκτρο».

- Υλοποιεί τη μέθοδο `press(1)`, η οποία αφού ελέγξει εάν η κατάσταση της συγκεκριμένης κίνησης είναι σε αναμονή για απελευθέρωση ( `waiting for release` ) έπειτα θέτει στον αριθμό φορών που πατήθηκε το πλήκτρο +1 και επίσης θέτει την κατάσταση της κίνησης σε `pressed`.
- Υλοποιεί τη μέθοδο `release`, η οποία θέτει την κατάσταση της κίνησης σε `released`.
- Υλοποιεί τη μέθοδο `isPressed` που καλεί την `getAmount` ώστε να ελέγξει το κατά πόσο έχει ή όχι πατηθεί το πλήκτρο της κίνησης από την τελευταία φορά που ελέγχθηκε.
- Υλοποιεί τη μέθοδο `getAmount` η οποία αρχικά ελέγχει τον αριθμό φορών που πατήθηκε το πλήκτρο, σε δεύτερο στάδιο ελέγχει αν η κατάσταση της κίνησης στο μεταξύ έχει μεταβληθεί σε `released` ώστε να θέσει τον αριθμό αυτό σε 0 και τέλος, ελέγχει αν η συμπεριφορά της κίνησης έχει κατάσταση `initial press` ώστε να θέσει τον αριθμό των φορών που πατήθηκε το πλήκτρο σε 0 και την κατάσταση σε `waiting for release` και επιστρέφει τον αριθμό των φορών αυτών.

Η κλάση `IOHandler` αποτελεί μία κλάση η οποία διαχειρίζεται την είσοδο από διάφορα `events` του πληκτρολογίου. Τα `events` αυτά αντιστοιχούνται στις κινήσεις `Actions`. Τα βασικά στοιχεία της `IOHandler` είναι τα εξής :

- Ορίζεται ένας πίνακας με τα `Actions`.
- Αρχικοποιείται ο `IOHandler` με ένα `component`, όπως η οθόνη και αγκιστρώνεται ένας `listener` στο `component` αυτό.

- Υλοποιείται η `mapToKey` μέθοδος, η οποία αντιστοιχεί ένα `Action` σε ένα κουμπί μέσω του κωδικού του.
- Υλοποιείται η `clearMap` μέθοδος που διακόπτει κάθε σχέση μεταξύ κουμπιών και του `action` για το οποίο καλείται.
- Υλοποιείται η `getMaps` μέθοδος που επιστρέφει τα ονόματα των κουμπιών της συγκεκριμένης κίνησης – `action`.
- Υλοποιείται η `resetAllActions` μέθοδος που κάνει τις κινήσεις όλες να μην έχουν πατηθεί.
- Υλοποιείται η `getKeyName` μέθοδος που επιστρέφει το όνομα του κουμπιού που πατείτε.
- Υλοποιείται η `getKeyCode` μέθοδος που επιστρέφει τον κωδικό του κουμπιού που πατείτε.

5. Στο πακέτο `ScreenGraphics` τρεις είναι οι βασικές για την εξυπηρέτηση της λειτουργικότητας του κλάσεις, η `Animation` και `ScreenControler`, `Sprite`. Παρακάτω θα περιγραφούν αυτές και τα βασικά τους σημεία.

Η κλάση `Animation` αποτελεί μία κλάση η οποία διαχειρίζεται έναν αριθμό από εικόνες `frames` και το χρόνο εμφάνισης καθενός από αυτά. Τα βασικά στοιχεία της `Animation` είναι τα εξής :

- Τα βασικά στοιχεία που ενδιαφέρουν στην κλάση αυτή είναι η λίστα με τα διαθέσιμα εμφανιζόμενα `frames`, ένας δείκτης ώστε να μπορούμε να γνωρίζουμε σε ποιο `frame` βρισκόμαστε καθώς διατρέχεται η λίστα, ο χρόνος εμφάνισης ενός συγκεκριμένου `frame` και τέλος ο συνολικός χρόνος εμφάνισης ενός σετ από `animations`.
- Δομεί ένα `animation` με μία συγκεκριμένη λίστα από εικόνες `frames` και αναθέτει στο `animation` αυτό ένα συγκεκριμένο χρόνο



διαρκείας. Επιπλέον μετά τον ορισμό του το προκαλεί να ξεκινήσει με κλήση της μεθόδου start.

- Υλοποιεί την start μέθοδο, η οποία θέτει δείκτη τρέχοντος frame της λίστας του animation στην αρχή και επίσης το χρόνο εκτέλεσης του animation στο 0.
- Υλοποιεί την μέθοδο clone που δημιουργεί και επιστρέφει ένα νέο animation το οποίο έχει για λίστα frame του τη λίστα του animation για το οποίο καλείται και συνολική διάρκεια αντιστοίχως.
- Υλοποιεί την μέθοδο addFrame που προσθέτει σε επόμενη θέση της λίστας των frames του animation ένα animated frame και αυξάνει το συνολικό χρόνο εκτέλεσης του animation.
- Υλοποιεί την update μέθοδο, η οποία ελέγχει το μέγεθος της λίστας του animation και αν είναι περισσότερα από 1 frames τότε αναλόγως προθέτει το χρόνο που έχει περάσει ( δηλαδή κάποια χιλιοστά του δευτερολέπτου ) στο χρόνο εκτέλεσης του animation. Έπειτα ελέγχει εάν ο χρόνος που έχει περάσει ξεπερνάει τον συνολικό ορισμένο χρόνο εκτέλεσης του animation ώστε να θέσει χρόνο και δείκτη frame σε 0 πάλι ώστε να αρχίσει από την αρχή. Τέλος, ελέγχει εάν ο χρόνος εκτέλεσης που έχει περάσει ξεπερνάει τον καθορισμένο χρόνο εκτέλεσης ενός frame από το animation και προσαυξάνει το δείκτη στη λίστα των frames του animation.
- Υλοποιεί την getFrame που αναλαμβάνει να επιστρέψει ένα συγκεκριμένο frame από τη λίστα των frames του animation.
- Υλοποιεί την getImage που αναλαμβάνει να επιστρέψει μία εικόνα που αντιστοιχεί στο συγκεκριμένο frame από τη λίστα των frames του animation.
- Υλοποιεί την removeAnimation που αναλαμβάνει να διαγράψει τη λίστα των frames του animation για το ποίο καλείται και θέτει το χρόνο εκτέλεσης του animation ίσο με το 0.

- Τέλος, ενθυλακώνουμε στην Animation κλάση μία εσωτερική κλάση που αφορά στο οποιοδήποτε Frame της λίστας το οποίο εμφανίζεται μέσω της AnimationFrame. Στην κλάση αυτή δημιουργείται για κάθε frame μία εικόνα και ένας καθορισμένος χρόνος εκτέλεσης, δηλαδή εμφάνισης της εικόνας του frame αυτού.

Η κλάση ScreenControler αποτελεί μία κλάση η οποία διαχειρίζεται τη ρύθμιση και την εμφάνιση των γραφικών της εφαρμογής στην οθόνη. Τα βασικά στοιχεία της ScreenControler είναι τα εξής :

- Ορίζει και αρχικοποιεί την εξ' ορισμού συσκευή απεικόνισης του συστήματος.
- Υλοποιεί μέθοδο επιστροφής όλων των διαθέσιμων displays του συστήματος.
- Υλοποιεί μέθοδο ελέγχου των διαθέσιμων displays με αυτά που ορίσαμε εμείς ως σχεδιαστές για την εφαρμογή.
- Υλοποιεί μέθοδο επιστροφής του πρώτου διαθέσιμου από τη λίστα των διαθέσιμων displays.
- Υλοποιεί μέθοδο εισόδου και αρχικοποίησης γραφικών σε πλήρη οθόνη μέσω νήματος και ορισμού 2 buffers για το frame που θα απεικονιστεί στην οθόνη.
- Υλοποιεί μέθοδο δημιουργίας και επιστροφής των γραφικών της οθόνης.
- Υλοποιεί μέθοδο ενημέρωσης της οθόνης μέσω κλήσης του επομένου διαθέσιμου buffer φορτωμένο με γραφική πληροφορία.
- Υλοποιεί μέθοδο επιστροφής παραθύρου που βρίσκεται σε κατάσταση πλήρους οθόνης.

- Υλοποιεί μέθοδο επιστροφής πλάτους παραθύρου.
- Υλοποιεί μέθοδο επιστροφής ύψους παραθύρου.
- Υλοποιεί μέθοδο αποδέσμευσης οθόνης θέτοντας το βασικό παράθυρο πλήρους απεικόνισης σε null.
- Υλοποιεί μέθοδο δημιουργίας εικόνας γραφικών προς εμφάνιση συμβατή με το υπάρχον σύστημα.

Η κλάση Sprite αποτελεί μία βασική κλάση η οποία δημιουργεί ένα sprite με κάποιο ορισμένο animation ενθυλακωμένο μέσα του. Τα βασικά στοιχεία της Sprite είναι τα εξής :

- Ορισμός ενός sprite με συγκεκριμένο animation.
- Υλοποιεί μέθοδο ενημέρωσης του animation του sprite όπως επίσης και των θέσεων του σε x και y άξονα βάση της ταχύτητας του. ( θέση = ταχύτητα \* χρόνο )
- Υλοποιεί μέθοδο ανάθεσης τιμής x στο sprite.
- Υλοποιεί μέθοδο ανάθεσης τιμής y στο sprite.
- Υλοποιεί μέθοδο επιστροφής τιμής x του sprite.
- Υλοποιεί μέθοδο επιστροφής τιμής y του sprite.
- Υλοποιεί μέθοδο επιστροφής τιμής πλάτους του sprite.
- Υλοποιεί μέθοδο επιστροφής τιμής ύψους του sprite.
- Υλοποιεί μέθοδο ανάθεσης τιμής ταχύτητας x σε pixel ανά millisecond στο sprite.
- Υλοποιεί μέθοδο ανάθεσης τιμής ταχύτητας y σε pixel ανά millisecond στο sprite.
- Υλοποιεί μέθοδο επιστροφής τιμής ταχύτητας x σε pixel ανά millisecond από sprite.
- Υλοποιεί μέθοδο επιστροφής τιμής ταχύτητας y σε pixel ανά millisecond από sprite.

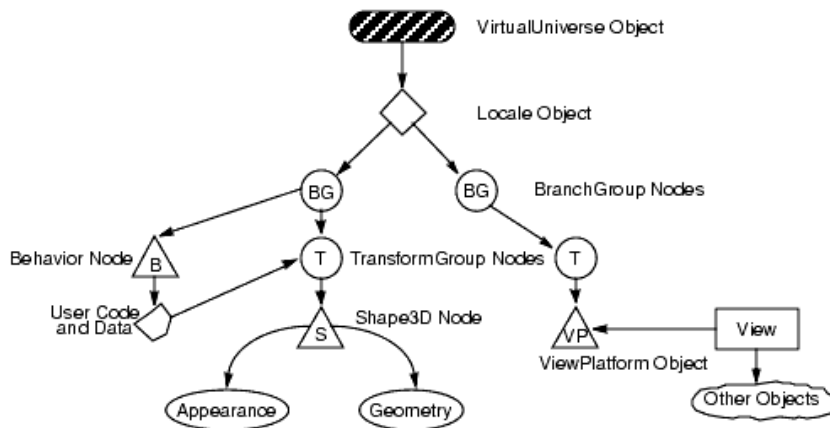
- Υλοποιεί μέθοδο επιστροφής της εικόνας του sprite τη στιγμή αυτή.
- Υλοποιεί μέθοδο δημιουργίας και επιστροφής ενός sprite όμοιο με το υπάρχον ως οντότητα.

6. Στο πακέτο Menu τρεις είναι οι βασικές για την εξυπηρέτηση της λειτουργικότητας του κλάσεις, η Menu3d και HelpForm, OptionsFrm. Παρακάτω θα περιγραφούν αυτές και τα βασικά τους σημεία.

Η κλάση Menu3d αποτελεί μία βασική κλάση η οποία είναι υπεύθυνη για τη δημιουργία του μενού της εφαρμογής. Τα βασικά στοιχεία της Menu3d είναι τα εξής :

- Αρχικά και εντελώς αφαιρετικά, είναι ένα frame το οποίο «τρέχει» κάνοντας χρήση μίας run μεθόδου νήματος για την εκτέλεση συγκριμένου μέρους κώδικα και εμφανίζεται στην οθόνη.
- Καθώς τρέχει είναι active έως ότου κάποιο άλλο νήμα ζητήσει να εκτελεσθεί.
- Διατίθεται και κάποιες ενσωματωμένες συμπεριφορές ( built-in behaviors ) τα οποία βρίσκονται στον κώδικα για περαιτέρω βελτίωση της εφαρμογής στο μέλλον.
- Ορίζει την μέθοδο loadscene φόρτωσης 3d μοντέλων με κατάληξη .obj, τα οποία φορτώνονται εξ' ολοκλήρου στο scene που επιστρέφεται.

- Ορίζει την μέθοδο `getscene` η οποία επιστρέφει ένα `transformgroup`, δηλαδή μία μορφοποίηση, αποτελούμενο από τα `transformgroups` που αναλύονται και αντιστοιχούνται βάση του `scene` που προηγουμένως φορτώθηκε.
- Υλοποίηση της μεθόδου δημιουργία του γράφου της όλης σκηνής `createscenegraph`. Σε αυτή τη μέθοδο επιστρέφεται ένα βασικό `branchgroup` το οποίο αποτελείται από επιμέρους `branchgroups`, τα οποία είναι κατασκευασμένα με συγκεκριμένη μορφοποίηση που είναι κάτι που επαφίεται στα `transformgroups`, τα οποία με τη σειρά τους συναποτελούνται από άλλα `transformgroups` το καθένα από τα οποία διαθέτει κάποιο `transform3d`, κάποιο `appearance`, κάποιο σχήμα, κάποιο `rotation` ( χαρακτηρισικά εμφάνισης ). Παρακάτω διατίθεται σχήμα κατανόησης του `Java3d World` ως μία απλή αναφορά.



Εικόνα 1. Το μοντέλο του προγραμματιστικού γράφου του `Java3d API`.

( βλέπε Τεχνολογίες )

- Ορίζει ένα virtual universe στο οποία προσθέτει έναν canvas3d με διάφορους key listeners και συγκεκριμένη μορφοποίηση, όπως επίσης και το scene που προηγουμένως έχει φτιαχτεί και το εμφανίζει.

Η κλάση OptionsFrm αποτελεί μία κλάση η οποία είναι υπεύθυνη για τη δημιουργία και την εμφάνιση της φόρμας του μενού της εφαρμογής που εισάγονται τα όποια στοιχεία δικτύου όπως η ip address του συμπαίκτη και η πόρτα του, ώστε να του στέλνονται τα δεδομένα από την τρέχουσα εφαρμογή. Τα βασικά στοιχεία της OptionsFrm είναι τα εξής :

- Ορίζει μία φόρμα βασισμένη σε ένα frame με ορισμένα στοιχεία, όπως τίτλο, κουμπιά, πλαίσια κειμένου κ.α.
- Υλοποιεί συγκεκριμένους key listeners οι οποίοι αφορούν παράδειγμα την επιβεβαίωση των στοιχείων που εισήχθη στη φόρμα.

Η κλάση HelpForm αποτελεί μία κλάση η οποία είναι υπεύθυνη για τη δημιουργία και την εμφάνιση της φόρμας του μενού της εφαρμογής που εμφανίζεται ως βοήθεια του χρήστη ως προς την πλοήγηση στο κυρίως παιχνίδι όπως επίσης και για την κατανόηση του στόχου του παιχνιδιού. Τα βασικά στοιχεία της HelpForm είναι τα εξής :

- Ορίζει τη φόρμα θέτοντας κατάλληλα χαρακτηριστικά ( ύψος, εμφάνιση κλπ. ).

- Υλοποιεί την μέθοδο `paint` που αναλαμβάνει να «ζωγραφίσει» το εσωτερικό της φόρμας κάνοντας χρήση διαστάσεων γραφικών
- Τέλος, προσαρτά και υλοποιεί έναν `key listener` στη φόρμα για την έξοδο – απόκρυψη της πατώντας το πλήκτρο `ESC`.

7. Στο πακέτο `Splash3d2` δύο είναι οι βασικές για την εξυπηρέτηση της λειτουργικότητας του κλάσεις, η `Splash3d` και `TestSplash3D`. Παρακάτω θα περιγραφούν αυτές και τα βασικά τους σημεία.

Η κλάση `Splash3d` αποτελεί μία κλάση η οποία είναι υπεύθυνη για τη δημιουργία και την εμφάνιση του `splash screen` αντικειμένου και ειδικότερα της δημιουργίας του `background` και του `canvas3d` της οθόνης εκκίνησης διαμέσου μίας σκηνής ( `scene` ) που απαρτίζεται από `branchgroups` βασισμένα στο `Java3d API`. Τα βασικά στοιχεία της `Splash3d` είναι τα εξής :

- Αρχικοποιεί, δημιουργεί και ορίζει τον `canvas3d`.
- Αρχικοποιεί και δημιουργεί ένα `screen capture` .
- Δημιουργεί ένα `simple universe` ορισμένο κατά τον `canvas3d`.
- Προσαρτά σε `simple universe` το `screen capture` ως οπίσθιο φόντο ( `background` ).
- Δημιουργεί ένα `view` ( θέαση ) του `simple universe` για την εμφάνιση του `splash screen` ολόκληρο.

Η κλάση TestSplash3D αποτελεί μία κλάση η οποία είναι υπεύθυνη για τη δημιουργία μίας σκηνής ( scene ) ως ένα branchgroup δέντρο του Java3d API, όπως επίσης και της αρχικοποίησης της οθόνης εκκίνησης προσθέτοντας τη σκηνή που δημιουργήθηκε σε αυτό. Τα βασικά στοιχεία της Splash3d είναι τα εξής :

- Ορισμός και δημιουργία σκηνής ( scene ).
- Υλοποίηση της createScene που αναλαμβάνει τη δημιουργία και επιστροφή ενός branchgroup κόμβου, ο οποίος αποτελείται από transform groups συνδυασμένα μεταξύ τους με κατάλληλο τρόπο και αφού έχουν προστεθεί σε αυτά τα κατάλληλα transform3d's, appearances και text geometry που κατασκευάζουν την ολότητα τους ως προς το σχήμα, το χρώμα, θέση στην οθόνη. Σε αυτό το τελικό branchgroup ρίζα που επιστρέφεται γίνεται επιπλέον προσθήκη συγκεκριμένου φωτισμού, όπως επίσης και των rotators για την κύλιση του οριζοντίως ή καθέτως επί της τελικής οθόνης του χρήστη .
- Τέλος, προσαρτά στο splash screen το οποίο προηγουμένως έχει ορίσει έναν timer που διαρκεί 5 δευτερόλεπτα ώστε να η εμφάνισή του να διαρκέσει μονάχα το χρόνο αυτό και αφού πυροδοτηθεί το γεγονός λήξης του χρόνου να εξαφανιστεί από την οθόνη του χρήστη.





*Εικόνα 2. Σύλληψη εικόνας ( print screen ) κατά την εκκίνηση του παιχνιδιού .*

8. Στο πακέτο Sounds υπάρχει η μία και μοναδική βασική κλάση για την εξυπηρέτηση της λειτουργικότητας του, η WavePlayer. Παρακάτω θα περιγραφούν αυτή και τα βασικά της σημεία.

Η κλάση WavePlayer αποτελεί μία κλάση η οποία είναι υπεύθυνη για τη δημιουργία και την υλοποίηση της αναπαραγωγής της μουσικής στο μενού και στο κυρίως παιχνίδι της εφαρμογής μέσω ενός νήματος που θα τρέχει για ορισμένο χρόνο. Τα βασικά στοιχεία της WavePlayer είναι τα εξής :

- Ορίζει και το Wave Player, το όνομα της διαδρομής του αρχείου που αναπαραχθεί.
- Δημιουργεί το μηχανισμό παύσης του νήματος ( lock object )
- Ειδοποιεί τα υπόλοιπα νήματα ότι είναι έτοιμο να ξεκινήσει να εκτελείται.
- Υλοποιεί τη βασική μέθοδο run που «τρέχει» το νήμα στη οποία συμβαίνουν διάφορα όπως:

- I. Ορισμός file αντικειμένου του αρχείου μουσικής.
  - II. Ορισμός `AudioInputStream` που εισάγει το αρχείο ήχου προς διαχείριση και αναπαραγωγή ως ένα ρεύμα ήχου.
  - III. Ορισμός `AudioFormat`. Η `AudioFormat` είναι η κλάση που καθορίζει μια συγκεκριμένη διάταξη των δεδομένων σε ένα ρεύμα. Εξετάζοντας τις πληροφορίες που είναι αποθηκευμένες στο ακουστικό σήμα μπορεί να ερμηνεύσει τα κομμάτια του ρεύματος στα δυαδικά δεδομένα ήχου.
  
  - IV. Μια πηγή γραμμή ( `source data line` ) δεδομένων ως μια γραμμή δεδομένων όπου δεδομένα μπορούν να γραφτούν βάση του `audioformat` του ρεύματος ήχου. Λειτουργεί ως πηγή για τον μίξερ ήχου της εφαρμογής. Η εφαρμογή γράφει ήχο σε μορφή `byte` σε μια πηγή γραμμής δεδομένων, η οποία χειρίζεται το `buffering` των `byte` και τα παραδίδει στους μίξερ. Το μίξερ στη συνέχεια μπορεί να παραδώσει το μίγμα σε μια θύρα εξόδου που οδηγεί σε μια συσκευή εξόδου ήχου, σε μια κάρτα ήχου.
  - V. Τέλος, υλοποιεί και τη μέθοδο που προκαλεί την παύση του `wave player` μέσω τη νήματος περιμένει.
9. Στο πακέτο `Statistics` υπάρχει μία κλάση για την εξυπηρέτηση της λειτουργικότητας του, η `StatsCollector`. Παρακάτω θα περιγραφούν αυτή και τα βασικά της σημεία.

Η κλάση `StatsCollector` αποτελεί μία κλάση η οποία είναι υπεύθυνη για τη συλλογή και μόνο των στατιστικών στοιχείων του παίκτη, ο

οποίος περνιέται ως παράμετρος κατά τον ορισμό της. Τα βασικά στοιχεία της StatsCollector είναι τα εξής :

- Ορίζει και αρχικοποιεί μία αναφορά στα ως τώρα στατιστικά που έχει ο εκάστοτε παίκτης. Αυτό, αντιστοιχώντας τις ζωές, τους πόντους, τους «σκοτωμούς» ( kills ) κ.α. σε τοπικές μεταβλητές για μετέπειτα χρήση.
- Ορίζει και υλοποιεί τη μέθοδο για την επιστροφή του αριθμού των ζώων.
- Ορίζει και υλοποιεί τη μέθοδο για την επιστροφή του αριθμού των πόντων.
- Ορίζει και υλοποιεί τη μέθοδο για την επιστροφή του αριθμού των «σκοτωμών» ( kills ).
- Ορίζει και υλοποιεί τη μέθοδο για την επιστροφή του αριθμού της πίστας – επίπεδο στο οποίο βρισκόμαστε.

10. Στο πακέτο Sprites έξι σε αριθμό είναι οι βασικές για την εξυπηρέτηση της λειτουργικότητας του κλάσεις, η Existence, Player, Crawler, Fly, Bullet, Gadget. Παρακάτω θα περιγραφούν αυτές και τα βασικά τους σημεία.

Η κλάση Existence αποτελεί μία κλάση η οποία αποτελεί τον πυλώνα της επέκτασης της Sprite κλάσης που προηγουμένως έχει οριστεί στο πακέτο ScreenGraphics και άρα αποτελείται ουσιαστικά από 4 animations. Είναι ουσιαστικά ένα Sprite το οποίο επηρεάζεται από τη βαρύτητα και μπορεί να αλληλεπιδράσει κατά τη διάρκεια του κυρίως παιχνιδιού. Τα βασικά στοιχεία της Existence είναι τα εξής :

- Ορίζει και αρχικοποιεί μία ύπαρξη στην οποία αναθέτει 4 animations και μία κατάσταση normal φυσική.

- Υλοποιεί τη μέθοδο `getSpeed` που επιστρέφει την ταχύτητα της ύπαρξης.
- Υλοποιεί τη μέθοδο `wakeUp` που αναθέτει μία ταχύτητα στην ύπαρξη μία φορά κατά την πρώτη εμφάνιση στο χάρτη ώστε να αρχίσει να κινείται.
- Υλοποιεί τη μέθοδο που επιστρέφει την κατάσταση στην οποία βρίσκεται η συγκεκριμένη ύπαρξη σε ορισμένη στιγμή.
- Υλοποιεί τη μέθοδο που αναθέτει μία κατάσταση στη συγκεκριμένη ύπαρξη σε ορισμένη στιγμή.
- Υλοποιεί τη μέθοδο `isAlive` που ελέγχει αν η κατάσταση στην οποία βρίσκεται η συγκεκριμένη ύπαρξη σε ορισμένη στιγμή είναι η `normal` – και άρα τη ύπαρξη είναι σίγουρα ζωντανή ( όχι `dead` ).
- Υλοποιεί τη μέθοδο που επιστρέφει `false` γενικότερα για το αν η συγκεκριμένη ύπαρξη «πετάει» σε ορισμένη στιγμή. Σε επόμενη επέκταση της `Existence` υπερφορτώνεται η εν λόγω μέθοδος και υλοποιείται διαφορετικά.
- Υλοποιεί τη μέθοδο που αντιστρέφει την κίνηση της ύπαρξης στον άξονα `X` μετά από εντοπισμό σύγκρουσης με κάποιο πλακάκι οριζοντίως – εμπόδιο του επιπέδου.
- Υλοποιεί τη μέθοδο που σταματά την κίνηση της ύπαρξης στον άξονα `Y` μετά από εντοπισμό σύγκρουσης με κάποιο πλακάκι καθέτως – εμπόδιο του επιπέδου.
- Υλοποιεί τη μέθοδο `reappears` που υπερφορτώνεται από την κλάση `player` αργότερα ώστε να ξαναεμφανίζει με συγκεκριμένο τρόπο τον παίκτη αφού σκοτωθεί.
- Υλοποιεί τη μέθοδο που θέτει ένα συγκεκριμένο `id` στην ύπαρξη ως ταυτότητα ορίζοντας κατά ένα τρόπο τη μοναδικότητά της στο παιχνίδι.
- Υλοποιεί τη μέθοδο που επιστρέφει το `id` το οποίο έχει ανατεθεί στη συγκεκριμένη ύπαρξη.

- Υλοποιεί τη μέθοδο `update` που αναλαμβάνει να εκτελέσει την λειτουργία της ενημέρωσης του `animation` της ύπαρξης. Αρχικά, επιλέγεται το κατάλληλο `animation` πρώτον βάση της κατεύθυνσης της ύπαρξης και έπειτα βάση της κατάστασης στην οποία βρίσκεται. Επιπλέον, ελέγχει αν το `animation` που επιλέγεται είναι διαφορετικό από το ήδη υπάρχον ώστε να το «αρχίσει» ή να το «συνεχίσει», ως προς την ενημέρωση του. Τέλος, υλοποιείται ακόμα ένας έλεγχος ως προς το χρόνο ο οποίος έχει περάσει από την ώρα που η ύπαρξη έχει επέλθει σε κατάσταση `dying` ώστε να συνεχίσει σε `dead`.

Η κλάση `Player` αποτελεί μία κλάση η οποία υλοποιεί τον παίκτη της εφαρμογής. Είναι η οντότητα εκείνη που επεκτείνει με τη σειρά της οποιαδήποτε ύπαρξη ( `Existence` ) και επεκτείνει τα βασικά της χαρακτηριστικά περαιτέρω, όπως για παράδειγμα ο παίκτης μπορεί και πηδάει. Τα βασικά στοιχεία της `Player` είναι τα εξής :

- Βασικά χαρακτηριστικά του παίκτη είναι η ταχύτητα που πηδάει, η ένδειξη ότι βρίσκεται στη γη, τα στατιστικά του όπως η ζωές, οι πόντοι κλπ.
- Ορίζει και αρχικοποιεί μία ύπαρξη με συγκεκριμένο `animation`.
- Υπερφορτώνει τις μεθόδους που αφορούν στην σύγκρουση οριζοντίως ή καθέτως με κάποιο πλακάκι - εμπόδιο της πίστας ( `tile` ).
- Υπερφορτώνει την μέθοδο ανάθεσης `y` τιμής ως κάτι το ξεχωριστό από την απλουστευμένη εκδοχή απλής ανάθεσης τιμής. Αυτό διότι η συμπεριφορά του παίκτη καλείται να εμφανίσει ένα `sprite` που υπόκειται στους νόμους της βαρύτητας. Συνεπώς σε κάθε καινούργια ανάθεση τιμής `y` γίνεται μία σύγκριση με το τρέχων `y` και αν το καινούργιο `y`

είναι μεγαλύτερο τότε επιδρά στο sprite ο νόμος της βαρύτητας.

- Υλοποιείται η μέθοδος που καθορίζει το «πήδημα» του παίκτη ως κίνηση απαραίτητη για τον διαχωρισμό του από τα υπόλοιπα sprites. Γίνεται λοιπόν, έλεγχος του κατά πόσον ο παίκτης βρίσκεται στη γη, και δεύτερον κατά πόσον ο χρήστης έχει δώσει εντολή για άμεση ενέργεια εξύψωσης του παίκτη και έπειτα τίθεται η κατάλληλη αρνητική ταχύτητα στον παίκτη ώστε να κινηθεί αρνητικά.
- Υλοποιείται η μέθοδος επανεμφάνισης του παίκτη κατά τον απότομο θάνατό του σε περίπτωση μετωπικής σύγκρουσης με sprite είτε τύπου fly είτε τύπου crawler για τα οποία θα επεκταθούμε παρακάτω. Ο παίκτης χάνει μία ζωή και επανέρχεται από το πάνω μέρος της πίστας στο παιχνίδι έχοντας μία normal κατάσταση ή αλλιώς εξαφανίζεται από την πίστα ( game over ).
- Υλοποιείται η μέθοδος `get_Id` για την επιστροφή της μοναδικής ταυτότητας του παίκτη.
- Υλοποιείται η μέθοδος `set_Id` για την ανάθεση της μοναδικής ταυτότητας του παίκτη.
- Υλοποιείται η μέθοδος `get_Level` για την επιστροφή του τρέχοντος επιπέδου του παίκτη.
- Υλοποιείται η μέθοδος `set_Level` για την ανάθεση του τρέχοντος επιπέδου του παίκτη.
- Υλοποιείται η μέθοδος `get_Money` για την επιστροφή των πόντων του παίκτη.
- Υλοποιείται η μέθοδος `set_Money` για την ανάθεση των πόντων του παίκτη.
- Υλοποιείται η μέθοδος `get_Kills` για την επιστροφή του αριθμού των sprites που «σκότωσε » ο παίκτης.
- Υλοποιείται η μέθοδος `set_Kills` για την ανάθεση του αριθμού των sprites που «σκότωσε » ο παίκτης.

- Υλοποιείται η μέθοδος `get_Life` για την επιστροφή του αριθμού των ζωών που έχει ο παίκτης.
- Υλοποιείται η μέθοδος `set_Life` για την ανάθεση του αριθμού των ζωών που έχει ο παίκτης.

Η κλάση `Bullet` αποτελεί μία κλάση η οποία υλοποιεί τα μαχαίρια που διαθέτει ο παίκτης σε κάθε πίστα. Είναι η οντότητα εκείνη που επεκτείνει με τη σειρά της οποιαδήποτε ύπαρξη ( `Existence` ) και επεκτείνει τα βασικά της χαρακτηριστικά περαιτέρω, όπως ο παίκτης. Η συμπεριφορά που έχει επεκτείνει μόνο λίγα επιπλέον πράγματα.

Τα βασικά στοιχεία της `Bullet` είναι τα εξής :

- Ορίζει ένα «μαχαίρι» με ένα συγκεκριμένο animation, δηλαδή μία λίστα από frames – εικόνες οι οποίες ορίζονται προς εμφάνιση για συγκεκριμένο χρόνο.
- Υλοποιεί την ανάθεση ταχύτητας στο `sprite` αυτό κατά τρόπο παρόμοιο με αυτόν του παίκτη για παράδειγμα, απλώς υπερφορτώνεται για λόγους διαφορετικότητας της τιμής της ταχύτητας.

Η κλάση `Crawler` αποτελεί μία κλάση η οποία υλοποιεί το ένα από τα δύο `sprite` που είναι ικανά να «σκοτώσουν» τον παίκτη στα επίπεδα σε περίπτωση μετωπικής σύγκρουσης. Είναι επίσης μία οντότητα που επεκτείνει με τη σειρά της οποιαδήποτε ύπαρξη ( `Existence` ) και επεκτείνει τα βασικά της χαρακτηριστικά περαιτέρω, όπως ο παίκτης. Η συμπεριφορά που έχει επεκτείνει μόνο λίγα επιπλέον πράγματα.

Τα βασικά στοιχεία της `Crawler` είναι τα εξής :

- Ορίζει ένα «ερπετό» με ένα συγκεκριμένο animation, δηλαδή μία λίστα από frames – εικόνες οι οποίες ορίζονται προς εμφάνιση για συγκεκριμένο χρόνο και αποτελούν το ερπετό αυτό.
- Υλοποιεί την ανάθεση ταχύτητας στο sprite αυτό κατά τρόπο παρόμοιο με αυτόν του παίκτη και του bullet για παράδειγμα, απλώς υπερφορτώνεται για λόγους διαφορετικότητας της τιμής της ταχύτητας ( πιο αργή ταχύτητα ).

Η κλάση Fly αποτελεί μία κλάση η οποία υλοποιεί το δεύτερο από τα δύο sprite που είναι ικανά να «σκοτώσουν» τον παίκτη στα επίπεδα σε περίπτωση μετωπικής σύγκρουσης. Είναι επίσης μία οντότητα που επεκτείνει με τη σειρά της οποιαδήποτε ύπαρξη ( Existence ) και επεκτείνει τα βασικά της χαρακτηριστικά περαιτέρω, όπως ο παίκτης. Η συμπεριφορά που έχει επεκτείνει μόνο λίγα επιπλέον πράγματα.

Τα βασικά στοιχεία της Fly είναι τα εξής :

- Ορίζει ένα «πετούμενο» με ένα συγκεκριμένο animation, δηλαδή μία λίστα από frames – εικόνες οι οποίες ορίζονται προς εμφάνιση για συγκεκριμένο χρόνο και αποτελούν το πετούμενο αυτό.
- Υλοποιεί την ανάθεση ταχύτητας στο sprite αυτό κατά τρόπο παρόμοιο με αυτόν του crawler για παράδειγμα, απλώς υπερφορτώνεται για λόγους διαφορετικότητας της τιμής της ταχύτητας ( πιο γρήγορη ταχύτητα ).



- Υπερφορτώνει επιπλέον την μέθοδο ελέγχου της κατάσταση της ύπαρξης αυτής ως προς το αν πετάει ή όχι και ταυτόχρονα της κατάστασης του ( dead or alive), διότι πετάει αν και μόνο αν είναι ζωντανό και αντιστρόφως - πέφτει και σκοτώνεται - εξαφανίζεται.

Η κλάση Gadget αποτελεί μία κλάση η οποία αποτελεί τον εξωτερικό φλοιό των οντοτήτων των gadget που «κατακτά» ο παίκτης διερχόμενος των επιπέδων. Στο εσωτερικό της κρύβονται κλάσεις όπως η Shine και η Target που απλώς ορίζονται διαφορετικά για να ξεχωρίζουν μεταξύ τους μονάχα λόγω τύπου. Τα βασικά στοιχεία της Gadget είναι τα εξής :

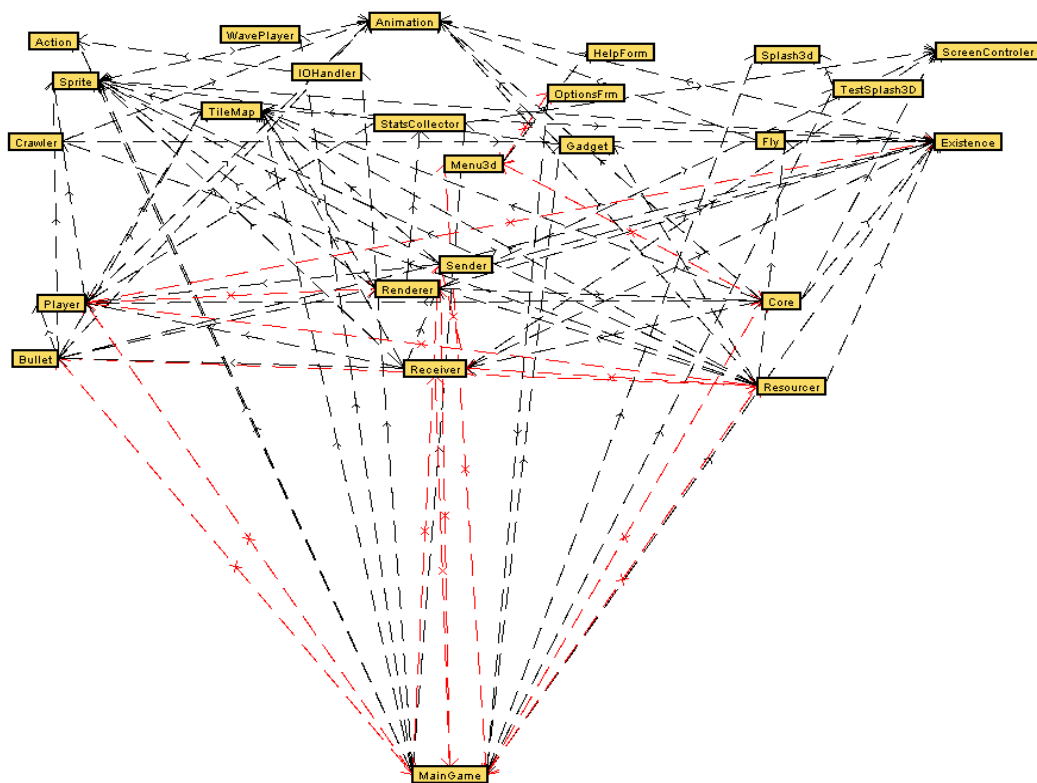
- Ορισμός και δημιουργία ενός sprite με ένα συγκεκριμένο animation, δηλαδή μία λίστα από frames – εικόνες οι οποίες ορίζονται προς εμφάνιση για συγκεκριμένο χρόνο και αποτελούν το «επιπόνημα» αυτό.
- Υπερφόρτωση της clone μεθόδου της Object κλάσης που δημιουργεί αντιγράφοι και επιστρέφει ένα νέο animation το οποίο έχει για λίστα frame του τη λίστα του animation για το οποίο καλείται και συνολική διάρκεια αντιστοίχως.
- Τέλος, ενθυλακώνει και ορίζει τις κλάσεις που αναφέρθηκαν παραπάνω ( Shine, Target ).

### Συσχετισμός Κλάσεων – Δια - Λειτουργικότητα

Στην ενότητα αυτή θα προσπαθήσουμε να περιγράψουμε τον τρόπο με τον οποίο συνδυάζονται οι κλάσεις μεταξύ τους και πως λειτουργούν η καθεμία από μόνη της αλλά και συνολικά όλες μαζί. Σύμφωνα με ένα απλό class diagram της γνωστής σε όλους μας UML μπορούμε να διακρίνουμε αρχικά της οντότητες της εφαρμογής μας, όπως επίσης και το συνολικό σχήμα των τάξεων και ένα οπτικό

συνδυασμό τους. Για το λόγω αυτό αρχικά θα παρουσιαστεί και θα αναλυθεί το διάγραμμα κλάσεων της εφαρμογής.

## Διάγραμμα Κλάσεων



Εικόνα 3. Διάγραμμα κλάσεων (print screen) του παιχνιδιού.

Στο παραπάνω διάγραμμα παρουσιάζονται είκοσι πέντε κλάσεις συνδεδεμένες μεταξύ τους είτε με σύνθεση, είτε με κληρονομικότητα. Φυσικά το εν λόγω διάγραμμα απεικονίζει την πιο γενική εικόνα του σχήματος της εφαρμογής που πρέπει να γίνει αντιληπτή από τον αναγνώστη. Αυτό που φαίνεται επιπλέον με μία πρώτη ματιά στο διάγραμμα κλάσεων είναι ότι υπάρχει μία κεντρική κλάση «διαχειριστής των πάντων» θα μπορούσαμε να πούμε ( MainGame ).

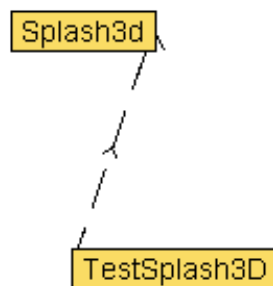
Ας πάρουμε όμως από το πιο ειδικό στο πιο γενικό τις κλάσεις ανά θεματική ενότητα και ας δούμε πως ενθυλακώνονται στις πιο λειτουργικές κλάσεις.

## Ενότητα SplashScreen

Στην ενότητα αυτή συνδυάζονται βασικά οι κλάσεις `Splash3d`, `Testsplash3d` και μετέπειτα συνδυάζονται με την `MainGame` από την οποία και καλούνται για να εκτελεστούν. Όσον αφορά στο συνδυασμό της `Splash3d` με την `Testsplash3d` θα λέγαμε ότι η πρώτη δημιουργεί το υπόβαθρο, δηλαδή ένα παράθυρο με διαφανές οπίσθιο φόντο και είναι έτοιμο να δεχθεί και να ενθυλακώσει στο εσωτερικό του `canvas3d` έναν γράφο από αντικείμενα.

Η `Testsplash3d` από την πλευρά δημιουργεί τη σκηνή αυτή ( γράφος 3d αντικειμένων – γραμμάτων ) και αναλαμβάνει την εισαγωγή αυτής στον `canvas3d`, όπως επίσης και την προσάρτηση ενός χρονομετρητή ως προς τη διάρκεια εμφάνισης του `Splash Screen`.

Τέλος, η `Testsplash3d` καλείται προς εκτέλεση από την `MainGame` κατά την εκκίνηση του παιχνιδιού.

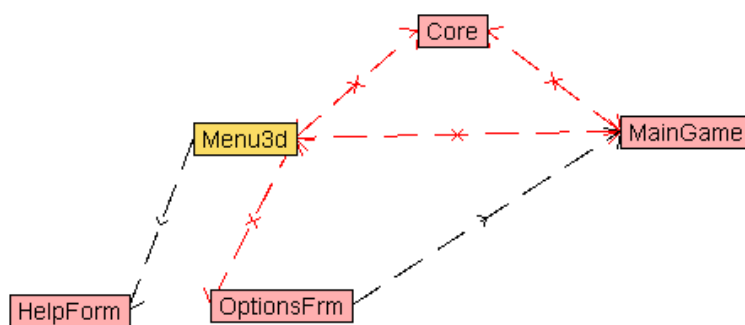


Εικόνα 4. Υποενότητα `Splash Screen` Διαγράμματος κλάσεων του παιχνιδιού .

## Ενότητα Menu

Στην ενότητα αυτή συνδυάζονται βασικά οι κλάσεις Menu3d, OptionsFrm, HelpForm και μετέπειτα συνδυάζονται με την MainGame από την οποία και καλούνται για να εκτελεστούν. Όσον αφορά στο συνδυασμό της OptionsFrm και της HelpForm με την Menu3d θα λέγαμε ότι η είναι δύο φόρμες που έχουν συγκεκριμένη λειτουργία η μία (OptionsFrm) για την εισαγωγή των στοιχείων του δικτύου ip και port number του συμπαίκτη σε συνεργατικό παιχνίδι και η άλλη (HelpForm) αποτελεί μία βοήθεια προς το χρήστη της εφαρμογής ως προς το «τί» είναι το παιχνίδι που θα παίξει όπως επίσης και υποδείξεις των πλήκτρων πλοήγησης στο παιχνίδι. Η Menu3d αναλαμβάνει την ενθουλάκωσή τους στο εσωτερικό της. Η Menu3d είναι και η πιο βασική κλάση της ενότητας αυτής διότι δημιουργεί εμφανίζει και διαχειρίζεται το μενού του παιχνιδιού αποτελούμενο από τρισδιάστατα γραφικά.

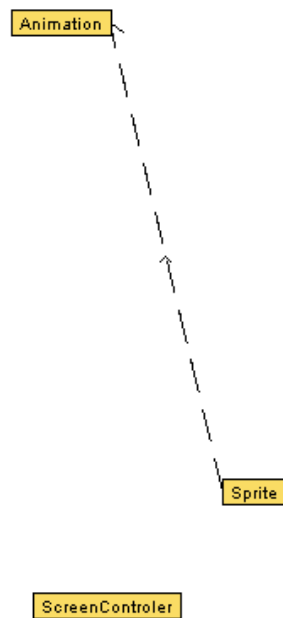
Τέλος, η Menu3d καλείται προς εκτέλεση από την MainGame κατά την εκκίνηση του παιχνιδιού και αφότου τελειώσει ο χρόνος εκτέλεσης του splash screen.



Εικόνα 5. Υποενότητα Menu Διαγράμματος κλάσεων του παιχνιδιού .

## Ενότητα ScreenGraphics

Στην ενότητα αυτή συνδυάζονται βασικά οι κλάσεις Animation, Sprite, ScreenControler και μετέπειτα συνδυάζονται με την MainGame από την οποία και καλούνται για να εκτελεστούν, όπως και από άλλες κλάσεις που αφορούν τη φόρτωση των sprites και την απεικόνισή τους στο πλαίσιο της οθόνης. Συνοπτικά λοιπόν θα μπορούσαμε να περιγράψουμε την Animation ως τη λίστα που υλοποιεί τα animations με την απόδοση συγκεκριμένου δείκτη σε κάθε εικόνα και κάποιο χρόνο σε αυτή και όλες μαζί να υλοποιούν το animation. Η κλάση αυτή επεκτείνεται από την Sprite που είναι αυτή που κάθε φορά ενσωματώνει στην κατασκευή ενός οποιουδήποτε Sprite ένα animation σε αυτό. Επιπλέον το ενημερώνει ως προς τις θέσεις του στους άξονες της οθόνης X και Y και εκτελεί λειτουργίες όπως η ανάθεση και η ανάκτηση ταχύτητας, θέσης, εικόνας του animation που ενσωματώνει, όπως και αντιγραφή του. Τέλος στην ενότητα αυτή προστίθεται και η κλάση που αφορά στον τρόπο εμφάνισης του παιχνιδιού στην οθόνη. Είναι η κλάση «ΕΛΕΓΧΟΣ» ( ScreenControler ) για τις δυνατότητες απεικόνισης βάση του συστήματος στο οποίο τρέχει η εφαρμογή και εφαρμόζει σύγκριση σύμφωνα με τη σχεδίαση των διαθέσιμων τρόπων απεικόνισης από την πλευρά του παιχνιδιού και διαλέγει την πρώτη διαθέσιμη από το τρέχων σύστημα. Επιπλέον θέτει σε πλήρη εμφάνιση ένα frame με συγκεκριμένο πλάτος και ύψος. Αναθέτει ένα συγκεκριμένο display mode σε αυτό. Δημιουργεί μια νέα στρατηγική για multi-buffering σχετικά με το frame αυτό. Το Multi-buffering είναι χρήσιμο για την απόδοση των επιδόσεων. Προσπαθεί να δημιουργήσει την καλύτερη στρατηγική που διατίθεται με τον αριθμό των buffers που παρέχονται ( στην περίπτωσή μας 2 σε αριθμό ). Υλοποιεί την εμφάνιση των γραφικών της οθόνης σε σχέση με τη στρατηγική που παρατίθεται και επίσης την δημιουργία buffer εικόνας συμβατής με το σύστημα την οποία και αναλαμβάνει να ενημερώνει.



Εικόνα 6. Υποεπότητα ScreenGraphics Διαγράμματος κλάσεων του παιχνιδιού

## Ενότητα IO

Στην ενότητα αυτή συνδυάζονται βασικά οι κλάσεις Action και IOHandler και μετέπειτα συνδυάζονται με στην MainGame από την οποία και καλούνται για να εκτελεστούν. Έτσι λοιπόν θα λέγαμε πως η Action δημιουργεί καθώς ορίζεται «κινήσεις» με συγκεκριμένο όνομα και συγκεκριμένες συμπεριφορές – καταστάσεις όπως και τις μεθόδους που αφορούν στη διαφοροποίηση των καταστάσεων από τις οποίες μπορεί να διέλθει μία «κίνηση» π.χ. :

- Pressed
- Released
- Waiting for Release
- Detect One Press Only

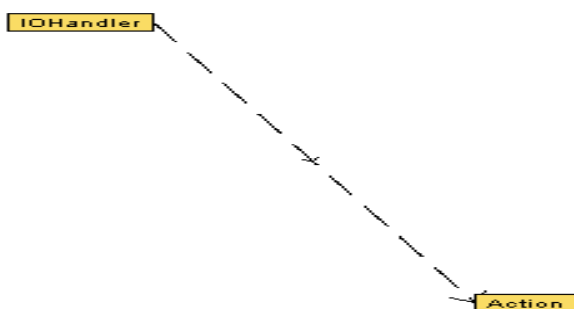
Τέλος διαθέτει και μία πολύ σημαντική μέθοδο που αφορά στην επιστροφή του αριθμού των φορών που πατιέται ένα κουμπί. Έτσι επιτυγχάνεται ο έλεγχος επί

τούτου και ρυθμίζονται κατά ορισμένο τρόπο οι κινήσεις. Δίνοντας ένα πιο κατανοητό παράδειγμα εξετάζουμε τον τρόπο με τον οποίο ο παίκτης πετάει τα μαχαίρια. Αναλυτικότερα και εξετάζοντας παράλληλα την άμεσα συνδεδεμένη στην Action κλάση IOHandler θα δείξουμε τη δια- λειτουργικότητα μεταξύ του ορισμού κάποιας κίνησης και του διαχειριστή της. Η κλάση IOHandler υλοποιεί τη διεπαφή Keylistener και τις μεθόδους του και ορίζεται διαχειριστής για κάποιο συγκεκριμένο στοιχείο, όπως η οθόνη για το οποίο θα διαχειρίζεται γεγονότα και αντιστοιχεί κινήσεις σε έναν πίνακα με τους κωδικούς των κουμπιών.

Όταν πατηθεί το αντίστοιχο κουμπί που πυροδοτεί την εκτόξευση ενός μαχαιριού. Γίνεται `press(1)` και αυξάνεται κατά ένα ο αριθμός των φορών που πατήθηκε το κουμπί συνεχώς `amount++` διότι απλώς δεν εμποδίζεται και καθώς τρέχει ο επεξεργαστής και έτσι το ένα πάτημα κουμπιού «δεν είναι μόνο ένα» αλλά όσοι και οι κύκλοι του επεξεργαστή. Συνεπώς μοιάζει σα να πατιέται συνεχώς έως ότου κληθεί η `release` του κουμπιού που το θέτει σε κατάσταση `released` που έτσι κι αλλιώς το εμποδίζει να μοιάζει ότι πατιέται.

Γι' αυτό είναι απαραίτητος ο λόγος ύπαρξης ορισμού μίας κίνησης με κατάσταση `detect one press only`. Αντιθέτως, αν ορίσουμε για παράδειγμα την «αριστερή κίνηση» του παίκτη κατ' αυτόν τον τρόπο, τότε δεν το αφήνουμε να κάνει «smooth move» καθώς το κουμπί μας είναι παρατεταμένα πατημένο αλλά πρέπει να το ξαναπατάμε συνεχώς προκειμένου να κινηθεί αριστερά.

Βλέπουμε λοιπόν ότι όλη η κινησιολογία είναι καθαρά θέμα σχεδιασμού του τρόπου συμπεριφοράς των οντοτήτων μας σε σχέση με το πώς πλοηγούμαστε ως χρήστες στη εφαρμογή και επιπλέον θέμα συνδυασμού της σχεδίασης αυτής με τον υπόλοιπο «συμπεριφοριστικό» κώδικα της εφαρμογής

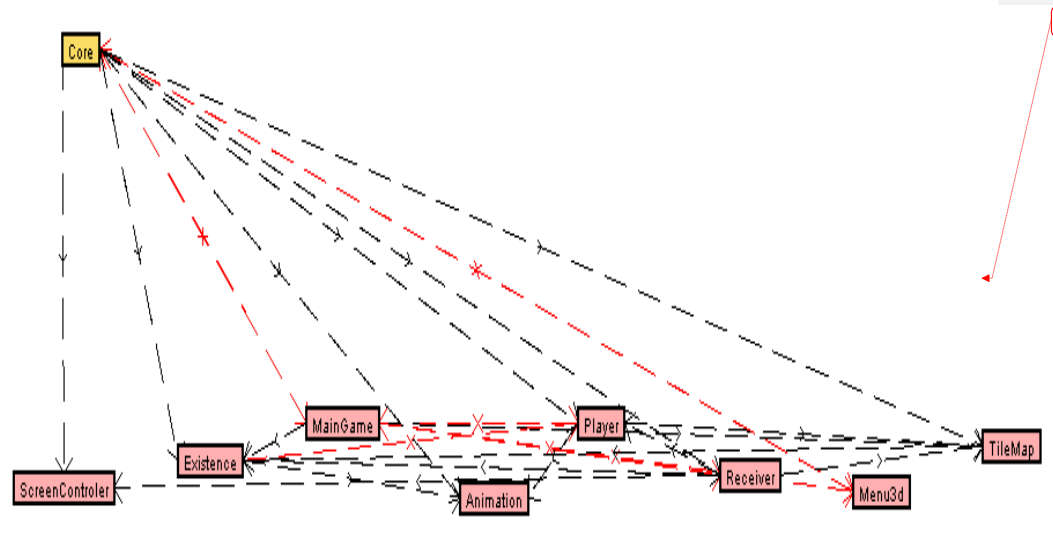


Εικόνα 6. Υποενότητα ScreenGraphics Διαγράμματος κλάσεων του παιχνιδιού

### Ενότητα ControllerController

Στην ενότητα αυτή η μοναδική κλάση που λαμβάνει χώρα είναι η Core. Είναι η κλάση που είναι υπεύθυνη για την κύρια «λούπα» του παιχνιδιού. Η ίδια κληρονομείται και από τον κεντρικό διαχειριστή ( MainGame ), για λόγους υλοποίησης μεθόδων όπως η update που αναλαμβάνει τη ενημέρωση του animation βάση του χρόνου ο οποίος έχει περάσει. Ορίζει της διαθέσιμες εμφανίσεις βάση ανάλυσης του παιχνιδιού ( display modes ), όπως επίσης και τις μεθόδους για την παύση ή συνέχιση του παιχνιδιού. Διαθέτει την run που υλοποιεί μέσω της init την αρχικοποίηση των γραφικών αντικειμένων στην οθόνη και την gameloop που αρχίζει να «τρέχει» το κυρίως παιχνίδι, δηλαδή ενημερώνει και ζωγραφίζει τα γραφικά μέσω της update και της draw. Οι τελευταίες υλοποιούνται αργότερα στην MainGame και είναι δύο από τις πιο βασικές μεθόδους στη λειτουργία του παιχνιδιού. Η τελευταία λειτουργία μέσω της lazilyExit είναι αυτή που εμπεριέχει την υλοποίηση ενός νήματος το οποίο αναλαμβάνει να τρέξει για να κάνει κλήση System.exit(0) με καθυστέρηση 2 δευτερολέπτων ( χρειάζεται για να προλάβουν να τερματιστούν τα νήματα του ήχου). Παρακάτω εμφανίζεται η Core με όλους τους έμμεσους συσχετισμούς της με άλλες κλάσεις.

- Formatted: Greek
- Formatted: Greek
- Formatted: Greek
- Formatted: Greek
- Formatted: Greek
- Formatted: Greek
- Formatted: Greek
- Formatted: Greek
- Formatted: Greek
- Formatted: Greek
- Formatted: Greek
- Formatted: Greek
- Formatted: Greek
- Formatted: Greek
- Formatted: Greek
- Formatted: Greek
- Formatted: Greek
- Formatted: Greek
- Formatted: Greek
- Formatted: Greek
- Formatted: Greek



- Formatted: Centered

Εικόνα 7. Υποενότητα Controller Διαγράμματος κλάσεων του παιχνιδιού

- Formatted: Greek



## Ενότητα Game

Στην ενότητα αυτή η μοναδική τέσσερις είναι οι βασικές κλάσεις που συνδυάζονται φέροντας εις πέρας τη λειτουργικότητα της, `TileMap`, `Renderer`, `Resource`, `MainGame`. Η `TileMap` ορίζει έναν χάρτη – επίπεδο – πίστα μέσω ενός δισδιάστατου πίνακα εικόνων όπου κάθε πλακάκι της πίστας ( `tile` ) είναι και μία αναφορά σε μια εικόνα του πίνακα. Επιπλέον διαθέτει και όλα τα `Sprites` που λαμβάνουν χώρα στο παιχνίδι όπως και μεθόδους για τη διαχείριση τους. Η `Resource` είναι η κλάση που αναλαμβάνει να «φορτώσει» όλα τα `sprites` και τα `tiles` ως εικόνες και ως `animations` και έπειτα να τα αναθέσει στο εκάστοτε χάρτη που ζητείται να φορτωθεί. Η `Renderer` είναι αυτή η οποία αναλαμβάνει να ζωγραφίσει τον χάρτη στη οθόνη μέσω της `draw` μεθόδου που καλείται από τη `MainGame` την οποία καλεί στην κύρια «λούπα» η `Core`. Όλη η «τοποθεσιολογία» των «πάντων» ανάγεται στην λογική συνέχεια του κώδικα (`active – inactive application`) κλπ όπως πχ η σχεδίαση του `background`. Ο βασικότερος υπολογισμός βάση του οποίου σχεδιάζεται η απεικόνιση της κίνησης στην οθόνη αφορά τον παίκτη και κατά πόσο μετακινήθηκε σε σχέση με το μήκος οθόνης, το μήκος του χάρτη, και το που βρίσκεται ο ίδιος. Όταν φτάσει στο μέσω περίπου της οθόνης μετράνε αρνητικά τα `pixels` τα οποία μετακινήθηκε. Για παράδειγμα όταν ο παίκτης βρίσκεται σε οποιοδήποτε `pixel` της οθόνης ζωγραφίζεται σε εκείνο ( 1000 ) μείον τη μετατόπισή του ( 700 ), άρα στο  $x=300$  όσον αφορά την τελική απεικόνιση. Κατά παρόμοιο τρόπο ζωγραφίζονται στην οθόνη και τα υπόλοιπα αντικείμενα που αποτελούν μέρος του παιχνιδιού. Τέλος η `MainGame` είναι η κλάση που αναλαμβάνει τη αρχικοποίηση και τον έλεγχο μέσω κώδικα και λογικής τα όλων των υπολοίπων κλάσεων προς εκτέλεση με κατάλληλη χρονική σειρά για παράδειγμα. Επίσης είναι υπεύθυνη για την αρχικοποίηση του δικτύου σε περίπτωση συνεργατικού παιχνιδιού. Ακόμη, υλοποιεί όπως προείπαμε την `update` μέθοδο της `Core` όπου γίνεται η αριθμητική ενημέρωση όπως και η ενημέρωση των καταστάσεων των αντικειμένων της εφαρμογής. Είναι υπεύθυνη για τη διαχείριση των νημάτων της μουσικής, για τον εντοπισμό των διαφόρων συγκρούσεων των θέσεων μεταξύ των `sprites` ή των πλακιδίων (`tiles`) των επιπέδων κ.α.

Formatted: Greek

Formatted: Greek

Formatted: Greek

Formatted: Greek

Formatted: Greek

Formatted: Greek

Formatted: Greek

Formatted: Greek

Formatted: Greek

Formatted: Greek

Formatted: Greek

Formatted: Greek

Formatted: Greek

Formatted: Greek

Formatted: Greek

Formatted: Greek

Formatted: Greek

Formatted: Greek

Formatted: Greek

Formatted: Greek

Formatted: Greek

Formatted: Greek

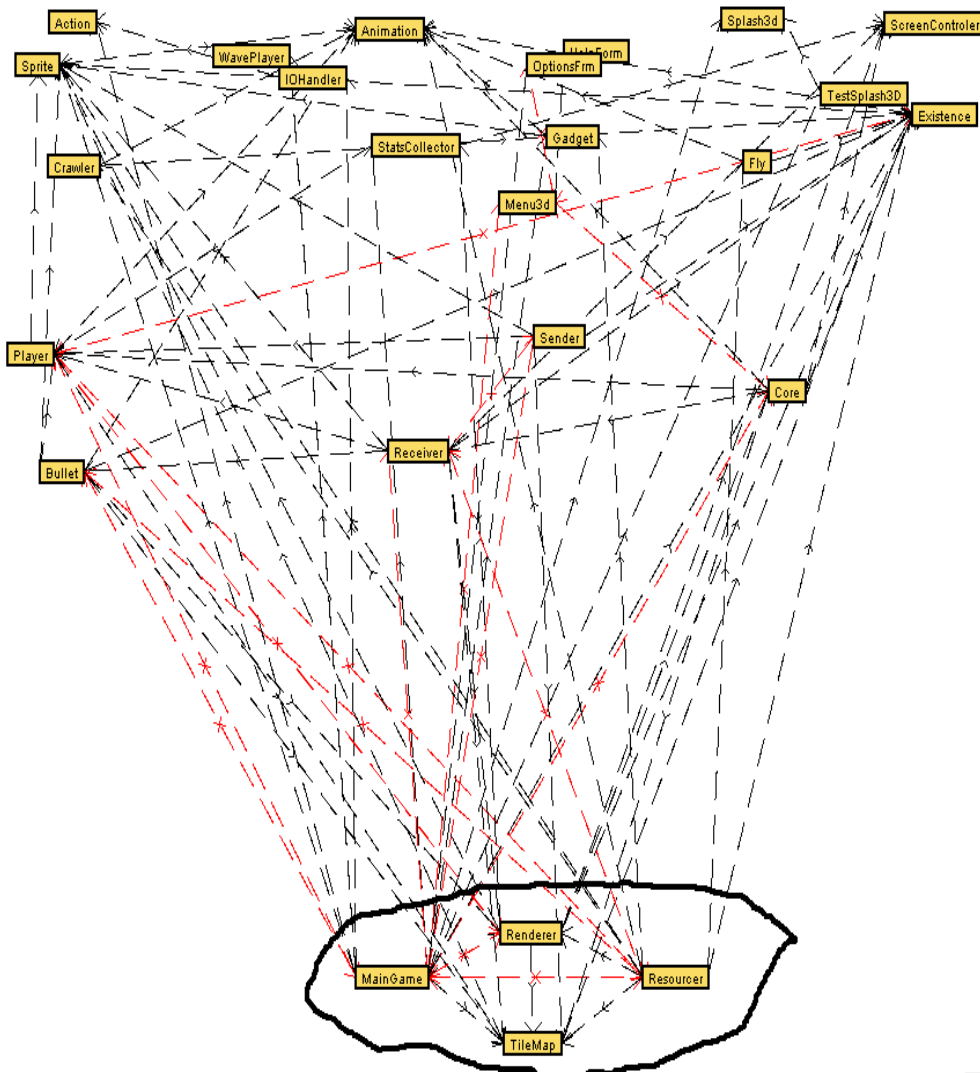
Formatted: Greek

Formatted: Greek

Formatted: Greek

Formatted: Greek

Formatted: Greek



Formatted: Centered

Formatted: Greek

**Εικόνα 8. Υποεξάντληση Game Διαγράμματος κλάσεων του παιχνιδιού**

## Ενότητα Sprites

Στην ενότητα αυτή η μοναδική έξι είναι οι βασικές κλάσεις που συνδυάζονται για να εξυπηρετήσουν τη λειτουργικότητα του πακέτου (Existence, Player, Bullet, Crawler, Fly, Gadget).

Η κλάση Existence επεκτείνει την Sprite του πακέτου ScreenGraphics και αποτελεί μία οποιαδήποτε ύπαρξη του παιχνιδιού που μπορεί να εξαφανιστεί από την οθόνη πεθαίνοντας και υπόκειται στους νόμους της βαρύτητας που σημαίνει ότι όταν αυτό δε συγκρούεται καθέτως με κάποιο πλακίδιο της εκάστοτε πίστας «πέφτει». Επιπλέον διαθέτει έξι καταστάσεις normal, dying, dead, game over, waiting. Κάθε μία για τον έλεγχο της ύπαρξης μέσω κώδικα. Επίσης η μεταβλητή που παρακολουθεί, σύμφωνα με το χρόνο που έχει περάσει από τον τελευταίο έλεγχο, το statetime δηλαδή πόσο χρόνο βρίσκεται η ύπαρξη στην κατάσταση αυτή.

Ορίζει τη συμπεριφορά μίας γενικής ύπαρξης ως προς το «πώς» θα συμπεριφέρεται την πρώτη φορά που εμφανίζεται στην οθόνη, αν πετάει, αν είναι ζωντανό, «πώς» όταν συγκρούεται οριζοντίως ή καθέτως με κάποιο πλακίδιο, ποιο animation να φορτώσει προς απεικόνιση βάση της κατάστασης και της ταχύτητας κατεύθυνσης του.

Επέκταση της κλάσης αυτής αποτελεί και η επόμενη κλάση της ενότητας αυτής που είναι η Player. Στην κλάση αυτή περιγράφεται η συμπεριφορά του παίκτη όπου διαθέτει επιπλέον τα στατιστικά στοιχεία του παίκτη όπως η ζωές του, οι πόντοι που έχει στην κατοχή του, οι «σκοτωμοί» που κατάφερε να κάνει και τέλος το επίπεδο στο οποίο βρίσκεται την κάθε στιγμή. Υλοποιεί το «πήδημα» του παίκτη με συγκεκριμένη αρνητική ταχύτητα από το έδαφος και τέλος την επανεμφάνιση του παίκτη σε περίπτωση σύγκρουσης του με άλλο sprite existence που το σκοτώνει ελέγχοντας τον αριθμό των ζώων του, την κατάσταση στην οποία βρίσκεται και τον θέτει είτε στο ίδιο χ αλλά τον «ρίχνει» από ψηλά ξανά μειώνοντας τον αριθμό των ζώων του κατά ένα είτε τον εξαφανίζει από την οθόνη σε περίπτωση κατάστασης game over.

Formatted: Greek

Formatted: Greek

Formatted: Greek

Formatted: Greek

Formatted: Greek

Formatted: Greek

Formatted: Greek

Formatted: Greek

Formatted: Greek

Formatted: Greek

Formatted: Greek

Formatted: Greek

Formatted: Greek

Formatted: Greek

Formatted: Greek

Formatted: Greek

Η επόμενη κλάση είναι η Bullet που αφορά στη συμπεριφορά των μαχαιριών που διαθέτει ο παίκτης σε κάθε πίστα – επίπεδο. Επεκτείνει λοιπόν την κλάση existence και η μόνη διαφορά έγκειται στη ταχύτητα με την οποία κινείται το μαχαίρι όταν ο χρήστης το πετάξει στο κυρίως παιχνίδι.

Formatted: Greek

Formatted: Greek

Οι κλάσεις Crawler και Fly επεκτείνουν με τη σειρά τους την Existence και διαφέρουν επίσης μονάχα στην ταχύτητα. Επιπλέον η Fly ορίζει στο εσωτερικό της την μονοσήμαντη έννοια της «ζωντανής υπόστασης» μίας Fly ύπαρξης συναρτήσεως του αν πετάει. Δηλαδή ένα Fly όταν ελέγχεται αν πετάει ουσιαστικά ελέγχεται το είναι ζωντανό ( 2 σε ένα ).

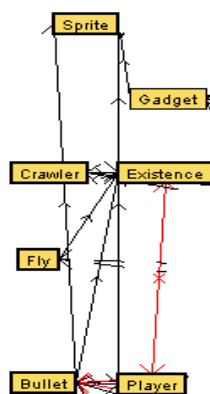
Formatted: Greek

Formatted: Greek

Τέλος, η κλάση Gadget αφορά όλα τα Sprites που καλείται να «κατακτήσει» ο παίκτης κατά τη διέλευση του από τα επίπεδα του κυρίως παιχνιδιού και να αποκτήσει πόντους μέσω αυτών ή να περάσει σε επόμενο επίπεδο. Η gadget κλάση επεκτείνει απευθείας τη συμπεριφορά ενός απλού sprite αντικειμένου και διαθέτει δύο ενθυλακωμένες κλάσεις : την shine και την target. Η πρώτη αφορά σε οποιοδήποτε gadget εκτός του στόχου - gadget που πρέπει να κατακτηθεί ώστε να περάσει ο παίκτης σε επόμενο επίπεδο, κάτι για το οποίο η target κλάση είναι υπεύθυνη. Ουσιαστικά ορίζονται διαφορετικά για να μπορούν να διαφοροποιηθούν μεταξύ τους κατ' απαίτηση του σχεδιαστή του παιχνιδιού.

Formatted: Greek

Παρακάτω παρουσιάζεται το διάγραμμα κλάσεων της ενότητας αυτής προς απεικόνιση και κατανόηση του τρόπου με τον οποίο συνδέονται και αλληλεπιδρούν.



**Εικόνα 9.** Υποεπότητα Sprites Διαγράμματος κλάσεων του παιχνιδιού

## Ενότητα Statistics

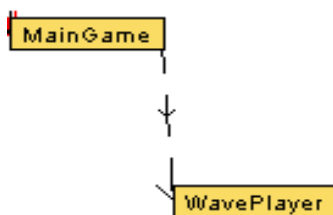
Στην ενότητα αυτή η μοναδική κλάση εξυπηρέτησης της λειτουργικότητας είναι η StatsCollector. Η μοναδική δουλειά που έχει να κάνει είναι να αναθέτει και να επιστρέφει στατιστικά στοιχεία για τον παίκτη. Καθώς ορίζεται λοιπόν ένας StatsCollector για έναν παίκτη αποθηκεύει εσωτερικά τον αριθμό των ζωνών του παίκτη, τους πόντους, τους «σκοτωμούς» που έχει κάνει, τον αριθμό του επίπεδου – πίστας στο οποίο βρίσκεται. Όλα αυτά όσον αφορά εκείνη τη στιγμή που ορίζεται. Χρήση του μπορεί γίνεται από τον κεντρικό διαχειριστή της εφαρμογής την Main Game όπου τον καλεί να συλλέξει τα στατιστικά στοιχεία αυτά. Ουσιαστική κλήση της κλάσης αυτής δε γίνεται στην έκδοση αυτή του παιχνιδιού. Υπάρχει για μελλοντική χρήση σε ενδεχόμενη επέκταση του παιχνιδιού. Στην τρέχουσα έκδοση υπάρχει τα στατιστικά ενθυλακώνονται στην κλάση του παίκτη και γίνεται αναφορά σε αυτά κατά τη μετάβαση σε επόμενα επίπεδα δείχνοντας την μέχρι τώρα πορεία του παίκτη, όπως επίσης και στο σημείο όπου η Main Game γράφει σε αρχείο τα στατιστικά αυτά. Το αρχείο αυτό βρίσκεται στον τρέχοντα κατάλογο της εφαρμογής.

```
try {  
  
    out1 = new BufferedWriter(new FileWriter(player1_stats));  
  
    if(((Player)player1).get_id()==1)//just a true condition  
    {  
        out1.write("player"+((Player)player1).get_id());  
        out1.newLine();  
        out1.write("Money"+((Player)player1).get_Money());  
        out1.newLine();  
        out1.write("Kills"+((Player)player1).get_Kills());  
        out1.newLine();  
        out1.write("Life"+((Player)player1).get_life());  
        out1.newLine();  
        out1.write("Level"+((Player)player1).get_Level());  
        out1.newLine();  
    }  
    out1.close();
```

*Απόσπασμα κώδικα από το σημείο εγγραφής σε αρχείο των στατιστικών του παίκτη.*

## Ενότητα Sounds

Στην ενότητα αυτή η μοναδική κλάση εξυπηρέτησης της λειτουργικότητας είναι η WavePlayer η οποία αναλαμβάνει να υλοποιήσει ένα νήμα που αφορά στην αναπαραγωγή της μουσικής είτε του «μενού» είτε του κυρίως παιχνιδιού. Ορίζεται με το όνομα αρχείου .wav που θα αναπαραχθεί και συγχρονίζοντας όλα τα υπόλοιπα threads ότι είναι έτοιμο να αρχίσει να τρέχει. Η κύρια συμπεριφορά της κλάσης αυτής υλοποιείται από την run μέθοδο που τρέχει για το thread της κλάσης καθώς καλείται. Περιγράφοντας λοιπόν τη λειτουργικότητα της όταν καλείται προς εκτέλεση από τη Main Game είτε αρχικά στο «μενού» είτε αργότερα στο κυρίως παιχνίδι διαβάζεται ένα αρχείο το οποίο σε επόμενο στάδιο γίνεται ρεύμα ήχου από το οποίο αποσπώνται πληροφορίες σχετικά με τη μορφή του και γίνεται περεταίρω διαχείριση σχετικά με την ένταση του για παράδειγμα. Έπειτα, αναλαμβάνει μία πηγή γραμμή ( source data line ) δεδομένων ως μια γραμμή δεδομένων όπου δεδομένα μπορούν να γραφτούν βάση του audioformat του ρεύματος ήχου. Λειτουργεί ως πηγή για τον μίξερ ήχου της εφαρμογής. Η εφαρμογή γράφει ήχο σε μορφή byte σε μια πηγή γραμμής δεδομένων, η οποία χειρίζεται το buffering των byte και τα παραδίδει στους μίξερ. Το μίξερ στη συνέχεια μπορεί να παραδώσει το μίγμα σε μια θύρα εξόδου που οδηγεί σε μια συσκευή εξόδου ήχου, σε μια κάρτα ήχου. Τέλος, υλοποιείται η διαχείριση της παύσης του νήματος που τρέχει μέσω της wait μεθόδου.



**Εικόνα 10.** Υποενότητα Sounds Διαγράμματος κλάσεων του παιχνιδιού

## Ενότητα **SendReceive**

Στην ενότητα αυτή για την εξυπηρέτηση της λειτουργικότητας υπάρχει η Sender και η Receiver. Οι δύο αυτές κλάσεις είναι υπεύθυνες για τη δικτύωση δύο τοπικά συνδεδεμένων υπολογιστών και τη μεταξύ τους ανταλλαγή των απαιτούμενων από την εφαρμογή δεδομένων ώστε να επιτυγχάνεται αλληλοενημέρωση των στοιχείων της μίας στην άλλη και αντιστρόφως.

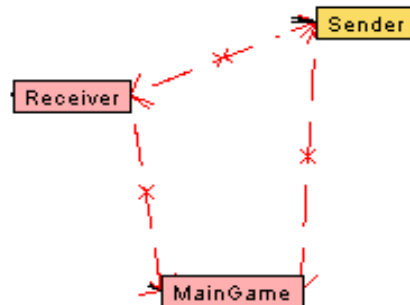
Περιγράφοντας γενικά τη διαλειτουργικότητα των κλάσεων αυτών σε συνδυασμό και με την Main Game θα μπορούσαμε να πούμε ότι σε περίπτωση επιλογής από το μενού συνεργατικού παιχνιδιού με τη συμπλήρωση των στοιχείων δικτύου ( ip address και port number ), τότε από τη Main Game υλοποιείται η μέθοδος connect που αναλαμβάνει να συνδέσει την τοπική ορισμένη στο «μενού» πόρτα της εφαρμογής με την πόρτα μίας ip που ορίζεται ως η «απέναντι» εφαρμογή. Ουσιαστικά με την κλήση της connect αρχικοποιείται ένας αποστολέας μέσω της sender και ένας παραλήπτης μέσω της receiver, ο οποίος παραλήπτης αρχίζει να τρέχει και να περιμένει να λάβει δεδομένα συνεχώς.

Εξηγώντας τη συμπεριφορά της Sender αρχικά θα λέγαμε ότι ορίζεται βάσει μίας ip και μίας port που αναλαμβάνει να στείλει δεδομένα και επίσης βάση των Sprites των οποίων τα στοιχεία θα στέλνει στη απέναντι εφαρμογή. Τέλος, ένα datagramsocket ορίζεται και αρχικοποιείται για την αποστολή των δεδομένων μέσω datagrampackets τα οποία ενθυλακώνουν πίνακες από byte που περιέχουν τα δεδομένα προς αποστολή (ένα dataoutputstream αδειάζει τα δεδομένα του σε ένα byteArrayoutputstream με το οποίο είναι και άμεσα συνδεδεμένο). Τέλος, διαθέτει και μία μέθοδο ελέγχου της κατάστασης της εφαρμογής ( active, inactive ).

Όσον αφορά το παραλήπτη της εφαρμογής ( Receiver ) αποτελεί ένα ακόμα νήμα το οποίο τρέχει συνεχώς «περιμένοντας» δεδομένα προς αποκωδικοποίηση. Ορίζεται κατά τρόπο παρόμοιο με τον αποστολέα, δηλαδή βάση ενός port number στο οποίο λαμβάνει δεδομένα από τον αποστολέα της απέναντι εφαρμογής και όλων των αντίστοιχων sprites

ορισμένων στην τοπική εφαρμογή για τροποποίηση των θέσεων τους σύμφωνα με το «τί» λαμβάνει και ένα `datagramsocket` αρχικοποιημένο σε στην πόρτα αυτή ώστε να λαμβάνει δεδομένα. Έπειτα ορίζει στη `run` μέθοδο του νήματος ένα `datagrampacket` με συγκεκριμένο μέγεθος της τάξεως του 1 Kbit δεδομένων προς ανάγνωση και καλεί σε μία συνεχώς επαναλαμβανόμενη «λούπα» την `receive` μέθοδο του `datagramsocket`, η οποία περιμένει να διαβάσει δεδομένα, αν υπάρχουν στη συγκεκριμένη πόρτα για την οποία ορίστηκε το `datagramsocket`. Μέχρι να διαβάσει δεδομένα «περιμένει» στο σημείο εκείνο του κώδικα. Καθώς γίνει λήψη δεδομένων, το πακέτο δεδομένων εναποτίθεται σε ένα ρεύμα εισόδου το οποίο με τη σειρά του εναποτίθεται σε ένα ρεύμα εισόδου δεδομένων από το οποίο και τελικά γίνεται η αποκωδικοποίηση βάση σχεδίασης της συμπεριφοράς της εφαρμογής. Για παράδειγμα εάν η εφαρμογή αρχίσει να λαμβάνει δεδομένα ( `connectioncreated = true` ) τότε ο `sender` της εφαρμογής στέλνει στην απέναντι εφαρμογή ότι αυτός είναι ο `active` ( γενική παραδοχή ) και έπειτα ο `receiver` της απέναντι αποκωδικοποιώντας την κατάσταση αυτή αυτό – ορίζεται ως `inactive`. Η διαδικασία που μόλις περιγράφηκε αφορά στην αρχικοποίηση του δικτύου μεταξύ των δύο συνεργατικών εφαρμογών. Κατά τη συνέχεια της αποκωδικοποίησης η κάθε η `Receiver` ανάλογα με το αν είναι `active` ή `inactive` ενημερώνει τις θέσεις και την κίνηση των αντίστοιχων `sprite objects` στα οποία διαθέτει αναφορές βάση των οποίων ορίστηκε. Δηλαδή η `active` ενημερώνει τα `sprites` του παίκτη 1 και η `inactive` του παίκτη 2 για παράδειγμα.





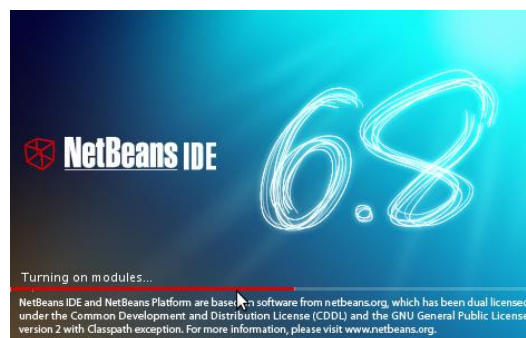
**Εικόνα 11.** Υποενότητα SendReceive Διαγράμματος κλάσεων του παιχνιδιού

## Βοηθητικό Λογισμικό

Το λογισμικό που χρησιμοποιήθηκε για την περάτωση της εν λόγω εφαρμογής του επιτραπέζιου παιχνιδιού ποικίλει. Παρακάτω θα γίνει μία σύντομη αναφορά σε όλα τα λογισμικά πακέτα που χρησιμοποιήθηκαν όπως επίσης και η ξεχωριστή προσφορά του καθενός από αυτά στο συνολικό αποτέλεσμα.

- NetBeans 6.8

Αρχικά, θα ήταν άτοπο να μην αναφερθούμε στο βασικό εργαλείο ανάπτυξης του εφαρμογής, το NetBeans 6.8. Ένα δωρεάν, ανοιχτού κώδικα ενσωματωμένο περιβάλλον ανάπτυξης για την ανάπτυξη λογισμικού. Το εν λόγω όλα τα εργαλεία που χρειάζονται για τη δημιουργία επαγγελματικών εφαρμογών με τη γλώσσα Java, καθώς και η C / C + +, PHP, JavaScript, Groovy, και Ruby. Είναι εύκολο να εγκατασταθεί, να χρησιμοποιηθεί κατευθείαν και να τρέχει σε πολλά λειτουργικά συστήματα όπως τα Windows, Linux, Mac OS X και Solaris.



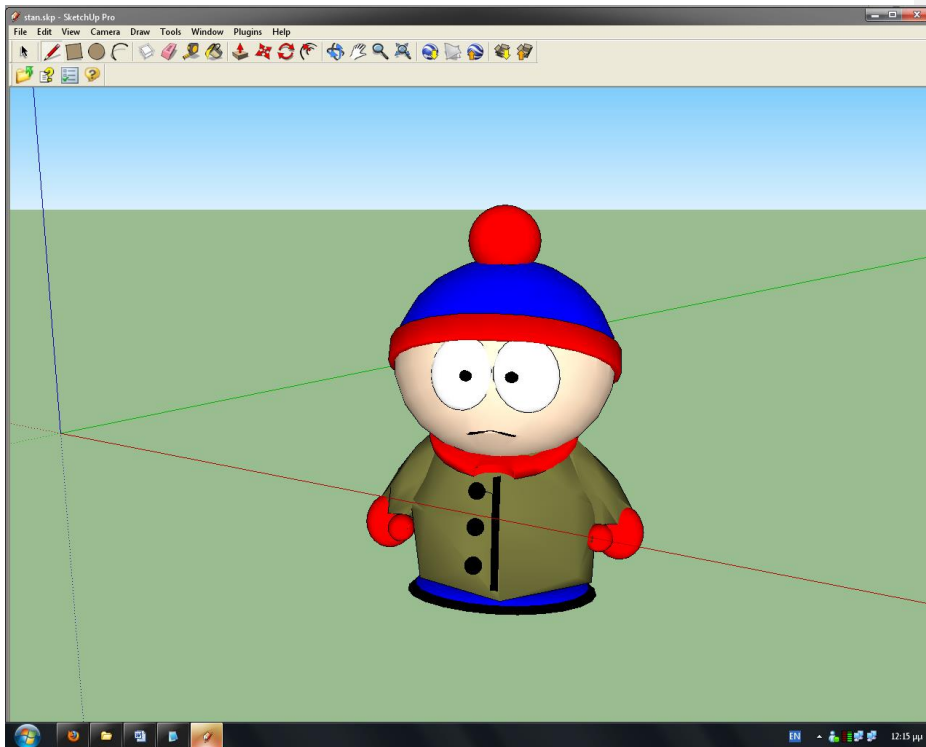
**Εικόνα 12\_** Ολοκληρωμένο Περιβάλλον Ανάπτυξης Λογισμικού

- Google SketchUp 7

Το SketchUp είναι ένα πρόγραμμα 3D μοντελοποίησης σχεδιασμένο για τους αρχιτέκτονες, πολιτικούς μηχανικούς, σκηνοθέτες, σχεδιαστές παιχνιδιών, και τα συναφή επαγγέλματα. Περιλαμβάνει επίσης στοιχεία για τη διευκόλυνση της τοποθέτησης των μοντέλων στο Google Earth. Είναι σχεδιασμένο για να είναι ευκολότερο στη χρήση από άλλα προγράμματα 3D.

Ένα χαρακτηριστικό του είναι το SketchUp 3D Warehouse που επιτρέπει στους χρήστες SketchUp έρευνα για τα μοντέλα που γίνονται από άλλους και να συμβάλλει μοντέλα. Η χρήση του στο project είναι ουσιαστική και απαρτίζεται από την εξαγωγή εικόνων τρισδιάστατου περιεχομένου από μοντέλα 3d που προσφέρει το Google 3d Warehouse. Μέσω του Google Sketchup έγινε διαχείριση των μοντέλων αυτών ώστε να εξαχθούν από αυτό εικόνες χρήσιμες για την αναπαράσταση των sprite αντικειμένων στο εσωτερικό του παιχνιδιού. Με τη χρήση του λογισμικού αυτού τα sprites στην εφαρμογή απεικονίζονται με όλους τους «φωτισμούς» και την τρισδιάστατη μορφή που παρέχει το ίδιο το 3d μοντέλο από το οποίο προέρχεται κάθε εικόνα - frame. Κατά αυτόν τον τρόπο δίνεται στη εφαρμογή μία αίσθηση τρισδιάστατης απεικόνισης και πιο όμορφου περιεχομένου του κυρίως παιχνιδιού. Επιπλέον το εν λόγω λογισμικό χρησιμοποιείται και για την εξαγωγή .obj αντικειμένων 3d μοντέλων τα οποία σε επόμενο στάδιο εισάγονται στο μενού της εφαρμογής αυτοτελή και με 3d απεικόνιση. Κάθε ένα από τα .obj που εισάγονται

ενσωματώνονται στον κώδικα μέσω της ονοματολογίας που διαθέτουν τα «πλέγματα» - meshes από τα οποία αποτελούνται.

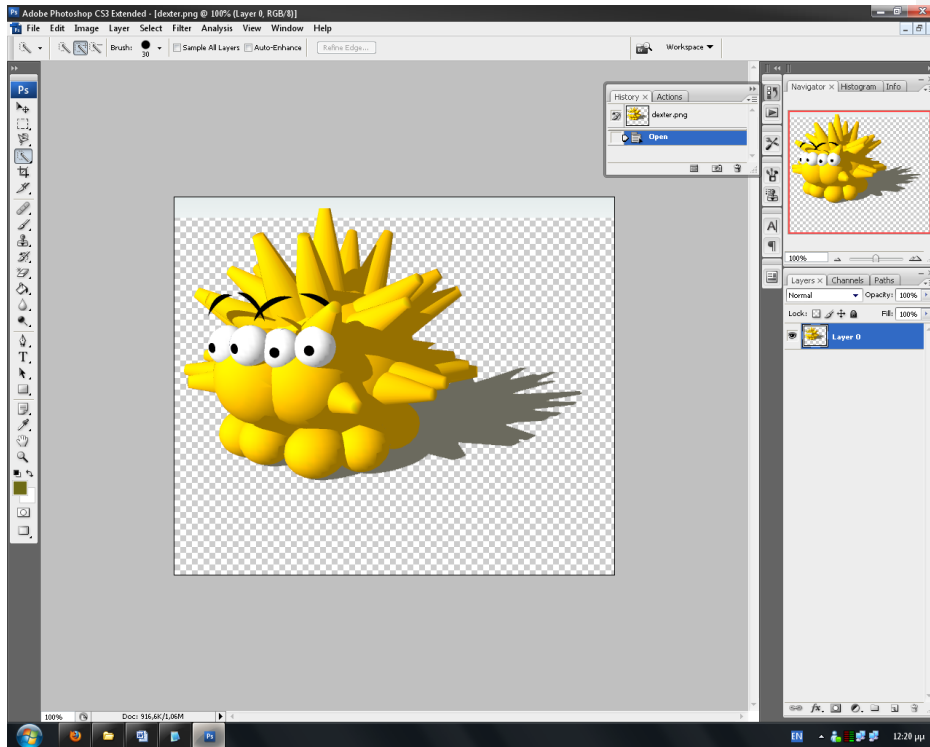


**Εικόνα 13.** Google SketchUp 7 pro – Πρόγραμμα 3D Μοντελοποίησης

- Photoshop

Το Adobe Photoshop είναι ένα πρόγραμμα επεξεργασίας γραφικών και παρέχεται από την Adobe Systems.

Νέα χαρακτηριστικά : περιλαμβάνει βελτιωμένο interface, βελτιώθηκε Camera Ακατέργαστη, καλύτερο έλεγχο των επιλογών εκτύπωσης, βελτιωμένη υποστήριξη PDF, και την καλύτερη διαχείριση με το Adobe Bridge, Φωτεινότητα / Αντίθεση προσαρμογή. Το εν λόγω λογισμικό χρησιμοποιήθηκε εκτεταμένα στη διάρκεια του project για τη διαχείριση, την επεξεργασία των εικόνων πλέον που προήλθαν από τα 3d μοντέλα του Google Sketch Up, τη προσαρμογή τους σε ορισμένο μέγεθος κλπ.



**Εικόνα 14.** Adobe Photoshop CS3 – Πρόγραμμα επεξεργασίας γραφικών

- Blender

Το Blender είναι μια εφαρμογή γραφικών 3D είναι διαθέσιμο ως ελεύθερο λογισμικό υπό την άδεια GNU General Public License.

Μπορεί να χρησιμοποιηθεί για την μοντελοποίηση, προσομοιώσεις και τη δημιουργία διαδραστικών 3D εφαρμογών, συμπεριλαμβανομένων και παιχνιδιών. Στην εφαρμογή μας χρησιμοποιήθηκε για την αποθήκευση εικόνων 3d μοντέλων ελεύθερης χρήσης και την παραγωγή από αυτά δισδιάστατων εικόνων και αποθήκευσής τους στα resources του project ως αναφορά για τα sprite αντικείμενα που επρόκειτο να ενσωματωθούν στο παιχνίδι.



**Εικόνα 15.** Blender – Πρόγραμμα επεξεργασίας 3D Μοντέλων

- Ζωγραφική των Windows Xp

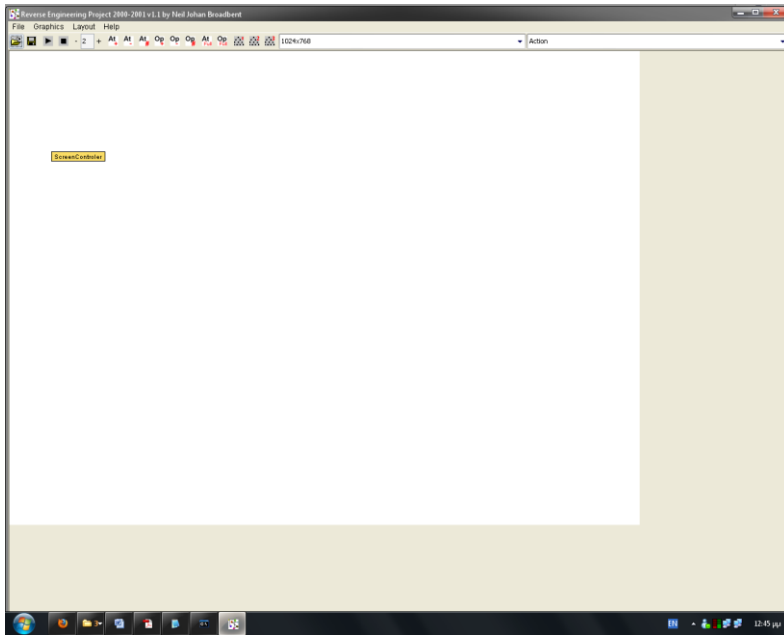
Η Ζωγραφική των βοηθημάτων των Windows Xp προσέφερε βασική βοήθεια στην απλή τροποποίηση και επεξεργασία των οποιονδήποτε εικόνων χρειάστηκαν κατά τη διάρκεια ανάπτυξης του συγκεκριμένου project.



**Εικόνα 16.** Ζωγραφική των Windows XP – Πρόγραμμα επεξεργασίας εικόνων

- Reverse Engineering Project

Το λογισμικό αυτό βοήθησε βασικά στην εξαγωγή διαγραμμάτων κλάσεων σε αρκετές φάσεις της εφαρμογής για την οπτική απεικόνιση της σύνδεσης μεταξύ των κλάσεων και του λογικού σχεδιαγράμματος του όλου project



17. Παράδειγμα απεικόνισης κλάσης στο Reverse Engineering Project

## Τεχνολογίες

Οι τεχνολογίες που χρησιμοποιήθηκαν κατά τη σχεδίαση και ανάπτυξη του συγκεκριμένου παιχνιδιού ποικίλουν. Παρκάτω θα περιγράψουμε τις βασικότερες αυτών :

### Java 3D

Java 3D είναι ένα βασισμένο σε ένα γράφημα ( scene graph ), 3D API για την πλατφόρμα Java. Τρέχει πάνω από είτε OpenGL ή Direct3D. Από την έκδοση 1.2, Java 3D έχει αναπτυχθεί στο πλαίσιο της κοινοτικής διαδικασίας Java. Σε σύγκριση με άλλες λύσεις, Java 3D δεν είναι μόνο ένα περιβάλημα γύρω

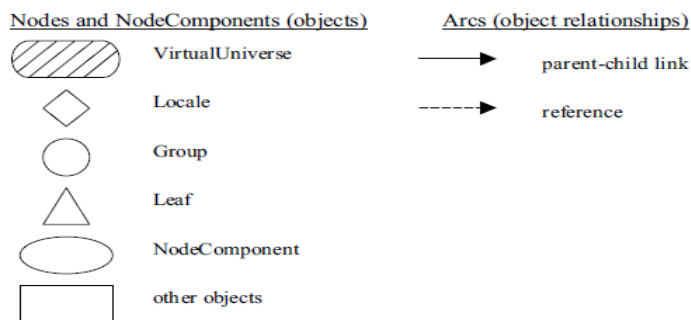
από αυτά τα API γραφικών, αλλά μια διεπαφή που συμπυκνώνει τον προγραμματισμό γραφικών χρησιμοποιώντας μια πραγματική, αντικειμενοστραφή έννοια. Εδώ μια σκηνή είναι κατασκευασμένη με μία σκηνή γράφημα ( scene graph ) που είναι μια αναπαράσταση των αντικειμένων που πρέπει να απεικονιστούν. Αυτό το γράφημα σκηνή είναι δομημένο σαν ένα δέντρο που περιέχει αρκετά στοιχεία που είναι απαραίτητα για την εμφάνιση των αντικειμένων. Επιπλέον, Java 3D προσφέρει εκτεταμένη υποστήριξη spatialized ήχου. Πολλές σελίδες πυκνού περιεχομένου θα μπορούσαν να γραφτούν για τη συγκεκριμένη τεχνολογία η οποία χρησιμοποιείται κατά βάση στην οθόνη εκκίνησης και στο κεντρικό μενού της εφαρμογής, αυτό όμως ξεπερνάει τα όρια της πτυχιακής εργασίας αυτής και για το λόγω αυτό θα γίνει μία περιληπτική αναφορά στα επιμέρους στοιχεία της μέσω εικόνων.

## Χαρακτηριστικά

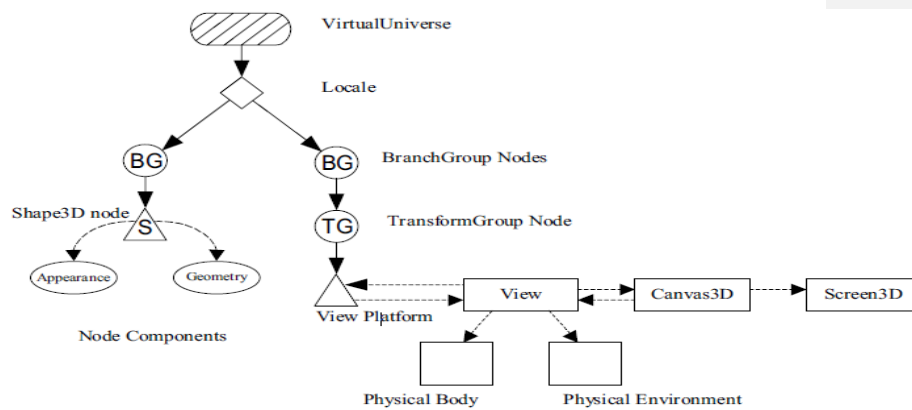


18. Παράδειγμα απεικόνισης γραφικών με Java 3D API

Πολυνηματική δομή σκηνής-γράφου



19. Παράδειγμα Συμβολισμοί Java 3D API γράφου



20. Παράδειγμα Java 3D API γράφου

Το βασικό περίγραμμα της Java 3D ανάπτυξη του προγράμματος αποτελείται από επτά βήματα :

1. Δημιουργία του Canvas3D αντικειμένου
2. Δημιουργία του VirtualUniverse αντικειμένου
3. Δημιουργία του Locale αντικειμένου, και προσάρτησή του στο VirtualUniverse αντικείμενο
4. Δημιουργία του View γράφου
  - a. Δημιουργία του View αντικειμένου
  - b. Δημιουργία του ViewPlatform αντικειμένου
  - c. Προσάρτηση του ViewPlatform και του Canvas3D αντικειμένου στο View object
5. Δημιουργία περιεχομένου στα branch graph(s)
6. Compilation των branch graph(s)



7. Εισαγωγή των sub graphs στο Locale ( addbranchgraph στο simpleuniverse - makes it live )

### Platform independent

Οποιαδήποτε εφαρμογή ανεπτυγμένη με το Java3d API δύναται να «τρέξει» σε οποιοδήποτε λειτουργικό σύστημα.

Generic Real-time API

Το Real-time API έχει να κάνει με τη απεικόνιση των γραφικών των εικόνων και των μετρικών των υπολογισμών στον τρισδιάστατο χώρο σε συγκεκριμένο χρόνο ώστε να εμφανίζονται αληθοφανή.

Υποστήριξη retained, compiled-retained, and immediate mode rendering

Με το rendering εννοούμε την απόδοση των γραφικών στην οθόνη και του τρόπου με τον οποίο επιτυγχάνεται αυτό. Στη συγκεκριμένη περίπτωση υπάρχει ένα μοντέλο το οποίο διαθέτει όλη την απεικόνιση και το μόνο που γίνεται είναι απλώς η ενημέρωση του μοντέλου αυτού και όχι η επανασχεδίαση και επαναπόδοσή του στην οθόνη.

Υποστήριξη hardware-accelerated JOGL, OpenGL and Direct3D renderers (depending on platform)

3D spatial ήχος

Importers για πολλές μορφές τρισδιάστατης απεικόνισης, όπως 3DS, OBJ, VRML, X3D

## **Java 2D API**

Το Java 2D είναι ένα API για τη σύνταξη γραφικών δύο διαστάσεων χρησιμοποιώντας τη γλώσσα προγραμματισμού Java.

Η Java 2D API παρέχει ακόλουθες δυνατότητες:

Ένα ενιαίο μοντέλο απόδοσης για οθόνες και εκτυπωτές

Ένα ευρύ φάσμα γεωμετρικών primitives

Ανίχνευση «χτυπήματος» σε σχήματα, κείμενο και εικόνες

Μοντέλο σύνθεσης  
Ενισχυμένη υποστήριξη χρώματος  
Εκτύπωση εγγράφων

Η Java 2D API ενισχύει τα γραφικά, το κείμενο και τις δυνατότητες απεικόνισης του Abstract Windowing Toolkit (AWT). AWT ήταν το αρχικό πακέτο εργαλείων για τη δημιουργία διεπαφών χρήστη και γραφικά σε Java. Για λόγους συμβατότητας, Java 2D είναι τεχνικά υπερσύνολο της εργαλειοθήκης AWT.

Java 2D είναι μια ισχυρή τεχνολογία. Μπορεί να χρησιμοποιηθεί για να δημιουργήσει πιο πλούσια user interfaces, παιχνίδια, κινούμενα σχέδια, εφαρμογές πολυμέσων ή διάφορα ειδικά εφέ.

## Sprite animation

Η απεικόνιση και γενικότερα η υλοποίηση των animations του παιχνιδιού είναι μία διαδικασία που ως τεχνολογία δεν είναι ιδιαίτερα πολύπλοκη. Αρχικά, ορίζεται ένας πίνακας με εικόνες. Επιπλέον ορίζεται και το συνολικό χρονικό διάστημα απεικόνισης ολόκληρου του πίνακα και τέλος οι εικόνες φορτώνονται η μία μετά την άλλη, καθώς ο πίνακας του συγκεκριμένου κάθε φορά animation καλείται, βάση ενός index – δείκτη που συνεχώς αυξάνεται φθάνοντας το άνω όριο του πίνακα των frames και πάλι από την αρχή.

## Threads

Ένα νήμα είναι ένα νήμα της εκτέλεσης σε ένα πρόγραμμα. Η Java Virtual Machine επιτρέπει σε μια εφαρμογή να έχει πολλαπλά threads εκτέλεσης που τρέχουν ταυτόχρονα.

Κάθε νήμα έχει προτεραιότητα. Τα νήματα με μεγαλύτερη προτεραιότητα εκτελούνται κατά προτίμηση σε σχέση με εκείνα με χαμηλότερη προτεραιότητα.

Όταν μια Java Virtual Machine ξεκινάει, υπάρχει συνήθως ένα ενιαίο νήμα μη

δαίμονας (το οποίο συνήθως καλεί τη μέθοδο που ονομάζεται κύρια – main). Η Java Virtual Machine συνεχίζει να εκτελεί τα νήματα μέχρι ένα από τα παρακάτω λόγους:

Η μέθοδος εξόδου exit της τάξης Runtime έχει κληθεί και ο security manager επιτρέπει την έξοδο από το πρόγραμμα.

Όλα τα νήματα που δεν είναι daemon έχουν πεθάνει, είτε με την επιστροφή από την κλήση προς τη μέθοδο run είτε ρίχνοντας μια εξαίρεση

Υπάρχουν δύο τρόποι για να δημιουργήσετε ένα νέο νήμα της εκτέλεσης. Ο ένας είναι να δηλώσει μια τάξη να είναι μια υποκατηγορία της Thread. Αυτή η υποκατηγορία θα πρέπει να υπερφορτώνει του την μέθοδο run της κλάσης Thread και να επεκτείνει τη συμπεριφορά της γενικότερα.

```
class PrimeThread extends Thread {
    long minPrime;
    PrimeThread(long minPrime) {
        this.minPrime = minPrime;
    }

    public void run() {
        // compute primes larger than minPrime
        . . .
    }
}
```

.....  
.....

```
PrimeThread p = new PrimeThread(143);
p.start();
```

#### *21. Παράδειγμα Thread με επέκταση κλάσης THREAD*

Ο άλλος τρόπος για τη δημιουργία ενός νήματος είναι να δηλωθεί μια κλάση που να υλοποιεί το Runnable interface. Έπειτα η κλάση αυτή πρέπει να υλοποιεί τη run μέθοδο για την εκτέλεση ενός αποσπάσματος κώδικα. Ακολουθεί παράδειγμα χρήσης :

```
class PrimeRun implements Runnable {
    long minPrime;
```

```

    PrimeRun(long minPrime) {
        this.minPrime = minPrime;
    }

    public void run() {
        // compute primes larger than minPrime
        . . .
    }
}

.....

....

PrimeRun p = new PrimeRun(143);
new Thread(p).start();

```

### 22. Παράδειγμα Thread υλοποίηση Runnable Interface

```

public WavePlayer(String wavfile) {
    filename = wavfile;

    pausedLock = new Object();
    // notify threads in pool it's ok to start
    synchronized (this) {
        notifyAll(); //everybody waiting on me..stare at me!!
    }
}

...

...

// if paused, wait until unpaused
    synchronized (pausedLock) {
        if (paused) {
            try {
                pausedLock.wait();
            }
            catch (InterruptedException ex) {
                return;
            }
        }
    }
}

```

### 23. Παράδειγμα Thread υλοποίηση με επέκταση Thread από τον κώδικα του Wave Player

## Udp Network

Το δίκτυο μεταξύ των 2 εφαρμογών που συνεργάζονται ώστε να παίξουν μαζί υλοποιείται από έναν Αποστολέα και ένα Παραλήπτη. Το πρωτόκολλο που χρησιμοποιείται για την επίτευξη ομαλότητας και όχι επαναποστολών χαμένων πακέτων που θα επιβάρυναν αρνητικά τη θεσιολογία των παικτών, χρησιμοποιείται το **User Datagram Protocol (UDP)**. Διάφορα προγράμματα χρησιμοποιούν το πρωτόκολλο UDP για την αποστολή σύντομων μηνυμάτων (γνωστών και ως datagrams) από τον έναν υπολογιστή στον άλλον μέσα σε ένα [δίκτυο υπολογιστών](#).

Ένα από τα κύρια χαρακτηριστικά του UDP είναι ότι δεν εγγυάται αξιόπιστη [επικοινωνία](#). Τα πακέτα UDP που αποστέλλονται από έναν [υπολογιστή](#) μπορεί να φτάσουν στον παραλήπτη με λάθος σειρά, διπλά ή να μην φτάσουν καθόλου εάν το δίκτυο έχει μεγάλο φόρτο. Αντιθέτως, το πρωτόκολλο [TCP](#) διαθέτει όλους τους απαραίτητους μηχανισμούς ελέγχου και επιβολής της αξιοπιστίας και συνεπώς μπορεί να εγγυηθεί την αξιόπιστη επικοινωνία μεταξύ των υπολογιστών. Η έλλειψη των μηχανισμών αυτών από το πρωτόκολλο UDP το καθιστά αρκετά πιο γρήγορο και αποτελεσματικό, τουλάχιστον για τις εφαρμογές εκείνες που δεν απαιτούν αξιόπιστη επικοινωνία.

### Source port

Η πόρτα του αποστολέα από την οποία προήλθε το πακέτο. Εάν ο παραλήπτης επιθυμεί να στείλει κάποια απάντηση, θα πρέπει να την στείλει στην πόρτα αυτήν. Το συγκεκριμένο πεδίο δεν είναι υποχρεωτικό και στις περιπτώσεις που δεν χρησιμοποιείται θα πρέπει να έχει την τιμή μηδέν.

### Destination port

Η πόρτα του παραλήπτη στην οποία θα πρέπει να παραδοθεί το πακέτο.

### Internet Address

Η διεύθυνση ip του μηχανήματος η οποία σε συνδυασμό με μία πόρτα υλοποιεί τη λειτουργία της αποστολής των δεδομένων μέσω udp πρωτοκόλλου. Αντιθέτως, η λειτουργία της λήψης δεδομένων στηρίζεται στη συνεχή ανάγνωση δεδομένων από byte array μίας πόρτας του τοπικού μηχανήματος.

## Tile engine

Είναι μία τεχνική αναπαράστασης γραφικών σε υπολογιστές βασισμένη σε αρχεία κειμένου τα οποία εμπεριέχουν τις ταυτότητες των εικόνων που στη συνέχεια θα αποτελέσουν το χάρτη. Ο τυπικός χάρτης από πλακάκια αποτελείται από έναν 2-διαστάσεων πίνακα / μήτρα που περιέχει αναφορές σε αντικείμενα πλακάκια. Αυτό το αντικείμενο πλακάκι περιέχει πληροφορίες όπως ο τύπος του εδάφους, καθώς και την ταυτότητα του γραφικού που πρέπει να αποδοθεί στην οθόνη κάθε φορά που το συγκεκριμένο πλακάκι είναι να ζωγραφιστεί. Η προσέγγιση αυτή επιτρέπει απλά, οπτικά δεδομένα χαρτών, αφήνοντας τους σχεδιαστές να δημιουργούν ολόκληρους κόσμους με ένα απλό επεξεργαστή κειμένου αν και οι περισσότεροι.

Έτσι για κάθε χάρτη φορτώνονται τα αντίστοιχα πλακάκια – εικόνες που λόγω της θέσης τους μέσα στο αρχείο κειμένου που αποτελεί την αναφορά στον απεικονιζόμενο στην οθόνη χάρτη, εμφανίζονται και με την ανάλογη σειρά στο δισδιάστατο χώρο καθώς αποδίδονται στην οθόνη.

## IO Handling

Όσον αφορά τη διαχείριση των διαφόρων εισόδων του χρήστη της εφαρμογής έχει δημιουργηθεί μία αντιστοιχία πλήκτρων - κωδικών σε ονόματα κινήσεων με ορισμένη συμπεριφορά και κατάσταση (GameAction), τα οποία σε επόμενο στάδιο αντιστοιχούνται σε «πίνακα κινήσεων» διαμέσου του οποίου προσπελαύνονται και υλοποιούνται (IOHandler).

Για παράδειγμα :

Ορίζουμε την κίνηση

- `moveLeft = new Action("moveLeft");`

Την Αντιστοιχούμε σε έναν πίνακα

- ```
input = new IOHandler(
screen.getFullScreenWindow());

input.mapToKey(moveLeft, KeyEvent.VK_LEFT);
```

Ελέγχουμε αν πατήθηκε

- `if (moveLeft.isPressed())`

Υλοποίηση της press

- ```
public void keyPressed(KeyEvent e) {
    Action action = getKeyAction(e);
    if (action != null) {
        action.press();
        e.consume();
    }
}
```

## Προτάσεις για Μελλοντική Επέκταση – Βελτίωση

Στην ενότητα αυτή πολλά μπορούν να ειπωθούν βάση του τι ακριβώς θα αποζητούσε ο καθένας ως επέκταση. Η διαφορά λοιπόν, έγκειται στο «γούστο» του καθενός από εμάς λοιπόν. Φυσικά όταν μιλάμε για υποκειμενικότητα ως προς την επέκταση - βελτίωση περιοριζόμαστε πάντα στ όρια της ίδιας φύσης του παιχνιδιού. Πρόκειται λοιπόν για επεκτάσεις πάνω σε ένα δισδιάστατο παιχνίδι σχεδιασμένο για επιτραπέζιο ηλεκτρονικό υπολογιστή με προσθήκες τρισδιάστατων γραφικών και διαχείριση των εικόνων, των animation όπως και της απόδοσης στην οθόνη (rendering) με τρόπο τέτοιο ώστε να μοιάζουν τρισδιάστατα κλπ.

- Κάτι που επώθησε αρχικά στην αναφορά αυτή και θα μπορούσε να αποτελεί βελτίωση του παιχνιδιού σε επόμενη έκδοσή του είναι το «σενάριο». Όπως έχουμε αναφέρει το σενάριο του παιχνιδιού είναι αρκετά

υλιστικό βάση του ότι ο παίκτης περνά διαμέσου των επιπέδων «περισυλλέγοντας» και «κερδίζοντας» πόντους όπως επίσης και «σκοτώνοντας» τα διάφορα sprites αντιπάλους που βρίσκονται στο δρόμο του. Αυτό είναι σίγουρα κάτι που μπορεί να αλλάξει και δε δόθηκε ιδιαίτερη έμφαση σε αυτό διότι στα πλαίσια της εν λόγω πτυχιακής βάση δόθηκε στην όλη ανάπτυξη και σωστή λειτουργικότητα του παιχνιδιού και όχι στην εμπορικότητα που θα μπορούσε να έχει βάση του σεναρίου του. Για παράδειγμα θα μπορούσε ο παίκτης να διέρχεται και πάλι μέσω των επιπέδων κάνοντας κάτι το «οικολογικό», όπως το να «σβήνει φωτιές» που ανάβουν οι αντίπαλοι ή θα «κατακτά» ένα gadget το οποίο θα του δίνει οδηγίες για το πώς θα συνεχίσει στο επίπεδο, τι θα πρέπει να πάρει και με ποια σειρά ώστε να προχωρήσει ομαλά και επιτυχώς στο επόμενο επίπεδο κλπ.

- Μία δεύτερη εξέλιξη – βελτίωση θα μπορούσε να γίνει ως προς τα γραφικά. Με αυτό εννοείται μία περαιτέρω προσθήκη τρισδιάστατων αντικειμένων στο κυρίως παιχνίδι αλλάζοντας φυσικά όχι το βασικό κορμό της όλης κινησιολογίας και διαχείρισης αλλά μόνο της προσθήκης για παράδειγμα στο οπίσθιο φόντο των επιπέδων ενός καμβά τρισδιάστατης απεικόνισης αντικειμένων εμποδίων της πίστας κλπ . Κάτι που σίγουρα θα άλλαζε μετά από αυτή την προσθήκη θα ήταν και ο εντοπισμός των collisions γενικότερα διότι δεν θα αφορούσε τη σύγκριση των θέσεων των εικόνων αλλά τη σύγκριση των εικόνων με τη τοπολογία των αντικειμένων του καμβά κλπ.



- Μία επιπλέον βελτίωση που λόγω του περιορισμένου χρόνου δε υπάρχει στην παρούσα έκδοση του παιχνιδιού και θα ήταν ωραίο να προστεθεί είναι οι ήχοι που θα μπορούσαν να αναπαράγονται κατά τη διάρκεια του παιχνιδιού αυτού καθώς ο παίκτης θα ερχόταν σε επαφή με κάποιον αντίπαλο για παράδειγμα ή ακόμα όταν θα έπαιρνε κάποιο gadget sprite αντικείμενο που θα βρισκόταν στο δρόμο του ή όταν θα έχανε ζωή. Θα μπορούσε να υλοποιηθεί με πολλούς τρόπους, όπως για παράδειγμα η αναπαραγωγή midi αρχείων ήχου ή ακόμα και mp3.
- Επέκταση ή καλύτερα διαφορετικός τρόπος σχεδίασης του παιχνιδιού θα μπορούσε να γίνει αν αντί του java 3d API γινόταν χρήση game engines που συνεργάζονται με τις ήδη υπάρχουσες βιβλιοθήκες και προσφέρουν έναν δικό τους τρόπο σκέψης λειτουργίας και λειτουργικότητας των πραγμάτων στο παιχνίδι απεικονιζόμενα στο game engine API and documentation. Μερικά παραδείγματα αυτών είναι η JGame, JMonkeyEngine, G Java 2d Graphics library κ.α.
- Τέλος, ίσως θα μπορούσε να γίνει μία επέκταση όσον αφορά και το τον τρόπο με τον οποίο χειρίζεται ο χρήστης το παιχνίδι. Αυτό αφορά δηλαδή τον πιθανό χειρισμό από κάποιο «εξωτερικό χειριστήριο» του εμπορίου. Για την επίτευξη του σκοπού αυτού πρέπει να γίνουν προσθήκες στις κινήσεις που έχουν οριστεί ως τώρα και μάλιστα η περαιτέρω προσθήκη και αντιστοιχία των κινήσεων αυτών στους κωδικούς που αντιστοιχούν τα πλήκτρα του «εξωτερικού χειριστηρίου» .

## Βιβλιογραφία-Πηγές-Αναφορές

- <http://java.sun.com/developer/onlineTraining/java3d/>
  - <http://www.javalobby.org/java/>
  - <http://www.java2s.com/Open-Source/Java-Document/6.0-JDK-Modules/java-3d/com.sun.j3d.utils.geometry.htm>
  - <http://www.java2s.com/Code/Java/3D/LightBug.htm>
  - <https://jogl-demos.dev.java.net/>
  - <http://fivedots.coe.psu.ac.th/~ad/jg2/ch15/index.html>
  - <http://robot.unipv.it/index.php/didattica/grafica-3d/54>
  - <http://www.3dcafe.com/>
  - <http://www.javacooperation.gmxhome.de/BildschirmflackernEng.html>
  - <http://zetcode.com/tutorials/javagamestutorial/>
  - <http://www.13thmonkey.org/~boris/jgame/>
  - <http://www.apl.jhu.edu/~hall/java/Java2D-Tutorial.html>
  - <http://www.java3d.org/>
  - <http://home.earthlink.net/~kduling/Java3D/First/>
  - <http://zetcode.com/tutorials/javagamestutorial/animation/>
  - <http://www.ambrosine.com/resource.html>
  - <http://java.dzone.com/news/3d-model-interaction-java-3d>
  - <http://home.cogeco.ca/~ve3ll>
  - <http://www.captain.at/howto-java-application-template-1.php>
  - <http://www.javagalaxy.com/Java.jsp>
  - <http://www.java-tips.org/other-api-tips/java3d/how-to-use-text2d-class-in-java3d.html>
  - <http://www.developer.com/java/other/article.php/3622246/Understanding-Lighting-in-the-Java-3D-API.htm#Run%20the%20program>
  - [http://java.sun.com/javase/technologies/desktop/java3d/forDevelopers/J3D\\_1\\_2\\_API/j3dguide/Intro.doc.html](http://java.sun.com/javase/technologies/desktop/java3d/forDevelopers/J3D_1_2_API/j3dguide/Intro.doc.html)
- 
- Java 3D Programming Daniel Selman
  - CoreTechniquesandAlgorithmsinGameProgramming Daniel Sánchez-Crespo Dalmau
  - Programming Games for Series 60 JUUSO KANNER
  - Programming Game AI by Example Mat Buckland
  - Game Interface Design by Brent Fox

- A.LaMothe - Tricks of the Windows Game Programming Gurus
- Sams\_Teach\_Yourself\_Game\_Programming\_CD
- Introduction to 3D Game Programming with DirectX 9.0 - fly
- Addison Wesley - Software Engineering and Computer Games
- CMP Books - 2004 - Game Design Workshop. Designing, Prototyping, and Playtesting Games - ISBN 1578202221
- DirectX.3D.Graphics.Programming.Bible.eBook-EEEn
- J.P.Flynt, O.Salem - Software Engineering for Game Developers
- Multiplayer Game Programming
- Wordware Publishing - 2002 - 3D Math Primer For Graphics And Game Development – Fly
- windows\_game\_programming\_guru
- OReilly.AI.for.Game.Developers.Jul.2004.eBook-DDU
- Wordware Publishing - 2002 - 3D Math Primer For Graphics And Game Development – Fly
- Java Network Programming by O'Reilly
- Java Game Programming For Dummies by John Willey
- Developing Games in Java by New Riders Publishing
- Developing Online Games - An Insider's Guide by New Riders Publishing
- Teach Yourself Internet Game Programming with Java In 21 Days by Sams Publishing
- Black Art of Java Game Programming by Waite Group Press
- Java 1.4 Game Programming by Wordware Publishing
- Developing\_Games\_In\_Java\_\_2003

## Παράρτημα Α - GNU Free Documentation License

Version 1.3, 3 November 2008

Copyright © 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc.

<<http://fsf.org/>>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

### 0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

### 1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML

using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text. The "publisher" means any person or entity that distributes copies of the Document to the public.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

## 2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute.

However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

### 3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

#### 4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.



J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same

entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

#### 5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

#### 6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

## 7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

## 8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

## 9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

#### 10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

#### 11. RELICENSING

"Massive Multiauthor Collaboration Site" (or "MMC Site") means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A "Massive Multiauthor Collaboration"

(or "MMC") contained in the site means any set of copyrightable works thus published on the MMC site.

"CC-BY-SA" means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

"Incorporate" means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is "eligible for relicensing" if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright (C) YEAR YOUR NAME.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.

A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with ... Texts." line with this:

with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.