Πανεπιστήμιο Πειραιώς

Τμήμα Ψηφιακών Συστημάτων

Πτυχιακή Εργασία

Ανάπτυξη παιχνιδιού σε πλατφόρμα Android.

Γκίκας Θεοδόσιος

Ε12035

Επιβλέπων καθηγητής

Καθηγητής Νικήτας-Μαρίνος Σγούρος

Περιεχόμενα

Περίληψη 2

1. Εισαγωγή 3

1.1 Ιστορία Βιντεοπαιχνιδιών 3

1.2 Ιστορία πλατφόρμας Android 5

1.3 Παρουσίαση παιχνιδιού 6

1.4 Βήματα Υλοποίησης Παιχνιδιού 9

1.5 Περίληψη Προγραμμάτων 10

2. Δημιουργία Στοιχείων Παιχνιδιού 12

2.1 Εισαγωγή 12

2.2 Ήχοι παιχνιδιού 12

2.3 Δημιουργία του παρασκηνίου 14

2.4 Δημιουργία του χαρακτήρα 14

2.5 Δημιουργία κίνησης του χαρακτήρα 15

2.6 Δημιουργία γραφικών διεπαφής χρήστη 16

2.7 Δημιουργία υπόλοιπων γραφικών 17

3. Σχεδιασμός Παιχνιδιού 20

3.1 Εισαγωγή 20

3.2 LibGDX framework 21

3.3 Μηχανή φυσικής Box2d 22

3.3.1 Συγκρούσεις (Collisions) 22

3.3.2 Σύσταση Χαρακτήρα 23

3.4 Διεπαφή χρήστη (UI) 24

3.4.1 Προβολή (Camera) 24

3.4.2 Διεπαφή με χρήση της Scene2d 24

3.4.3 Τα γραφικά του Scene2d 25

3.5 Δημιουργία της πίστας 25

3.6 Multiplayer 26

3.6.1 KryoNet Java Library 26

3.6.2 Χρήση της KryoNet 26

3.7 Δημιουργία αρχείου για το Google Play 27

4. Συμβουλές και Παρατηρήσεις 28

4.1 Εισαγωγή 28

4.2 Συμβουλές και Παρατηρήσεις 28

5. Πως ξεκίνησα το Game Development 29

Βιβλιογραφία 30

Περίληψη

Σκοπός της παρούσας πτυχιακής εργασίας είναι να αναλυθεί η διαδικασία δημιουργίας ενός βιντεοπαιχνιδιού σε πλατφόρμα Android. Ακόμα αναλύονται διάφορες δυσκολίες και θέματα που υπάρχουν στην δημιουργία παιχνιδιού για Android. Επίσης στα πλαίσια της εργασίας θα αναπτυχθεί ένα διδιάστατο adventure/platform παιχνίδι το οποίο θα ανέβει και στο Google Play Store.

Η εργασία χωρίζεται σε πέντε μέρη.

Στο πρώτο μέρος γίνεται μια σύντομη αναδρομή στην ιστορία των βιντεοπαιχνιδιών και της πλατφόρμας Android καθώς και την εξέλιξή τους με τα χρόνια. Επίσης γίνεται μια παρουσίαση του παιχνιδιού καθώς και μια σύντομη περιγραφή των βημάτων και των προγραμμάτων που χρησιμοποιήθηκαν για την υλοποίησή του.

Στο δεύτερο μέρος αναλύεται το πώς υλοποιείται το καλλιτεχνικό κομμάτι ενός βιντεοπαιχνιδιού. Δηλαδή τη δημιουργία των στοιχείων του παιχνιδιού (game assets), τα οποία είναι τα γραφικά και οι ήχοι του παιχνιδιού, με τον όρο γραφικά εννοούμε όλα αυτά που βλέπει ο χρήστης. Η ενότητα κλείνει με το θέμα του animation στα δισδιάστατα μοντέλα.

Στο τρίτο μέρος παρουσιάζεται η υλοποίηση του παιχνιδιού με την βοήθεια του libGDX framework και του Android Studio. Γίνεται ανάλυση στο πώς χρησιμοποιείται και το πώς δουλεύει η μηχανής φυσικής box2d του libGDX framework. Ακόμα εξηγείται το πώς φτιάχνεις το UI με το Scene2d στο libGDX framework. Τέλος αναφέρεται το πως δημιουργείται το αρχείο που πρέπει κάποιος να ανεβάσει στο Google Play Store.

Στο τέταρτο μέρος έχουμε συμβουλές και παρατηρήσεις από όλη την διάρκεια παραγωγής του παιχνιδιού. Τι να κάνει και τι να μην κάνει κάποιος.

Στο τελευταίο αναφέρονται λίγα λόγια για το πώς ξεκίνησα και ασχολήθηκα με την ανάπτυξη παιχνιδιών (game development).

1. Εισαγωγή

1.1 Ιστορία Βιντεοπαιχνιδιών

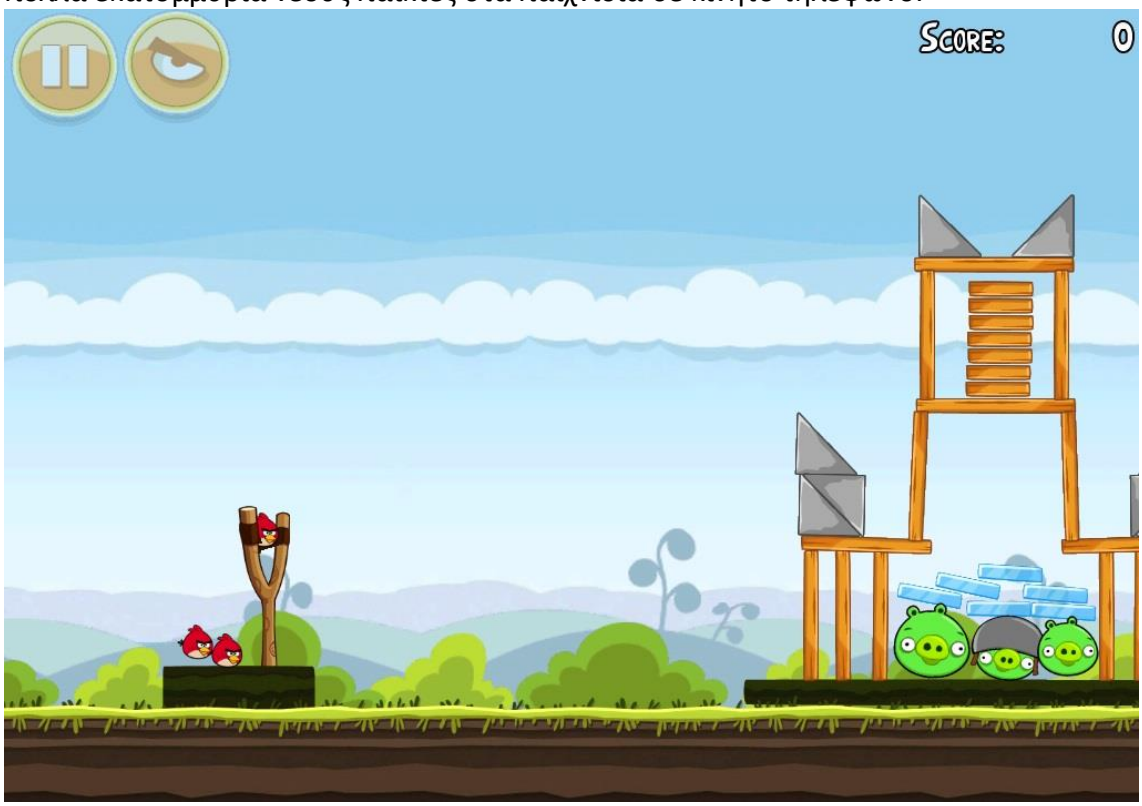
Ένα από τα πρώτα βιντεοπαιχνίδια στην ιστορία θεωρείται το «Tennis For Two» το οποίο δημιουργήθηκε το 1958 από τον Αμερικάνο φυσικό William Higinbotham για την ετήσια έκθεση του εθνικού εργαστηρίου του Brookhaven. Το παιχνίδι είχε προβληθεί στην οθόνη ενός παλμοσκοπίου (oscilloscope) και αναπαριστούσε ένα παιχνίδι τένις το οποίο απεικονιζόταν από τα πλάγια. Πρόκειται για ένα παιχνίδι για δύο παίκτες που ο καθένας από αυτούς έλεγχε την γωνία και το πότε θα χτυπήσει την μπάλα με ένα τηλεχειριστήριο. Παρόλο που τα γραφικά του ήταν πολύ απλοϊκά, είχε τεράστια επιτυχία στην έκθεση και απέδειξε την αγάπη που έχει ο κόσμος για τον συγκεκριμένο τρόπο ψυχαγωγίας. Η στιγμή όπου τα παιχνίδια εδραιώθηκαν στο ευρύ κοινό παγκοσμίως, ήταν την δεκαετία 70’-80’ με την εμφάνιση των arcades αλλά και των πρώτων οικιακών κονσόλων.

Εικόνα 1.1.1 : Το παιχνίδι Tennis For Two. Πηγή: en.wikipedia.org

1.1.2 Βιντεοπαιχνίδια σε Κινητά Τηλέφωνα

Το παλαιότερο γνωστό παιχνίδι σε κινητό τηλέφωνο ήταν μια παραλλαγή του γνωστού «Tetris» στην συσκευή Hagenuk MT-2000 του 1994. Το 1997, η Nokia ξεκίνησε το πολύ πετυχημένο φιδάκι «Snake». Το «Snake» έχοντας το προ-εγκατεστημένο στις περισσότερες κινητές συσκευές που κατασκεύαζε η Nokia, έγινε το πιο δημοφιλή παιχνίδι παγκοσμίως.
Μια παραλλαγή του παιχνιδιού «Snake» για το Nokia 6110, χρησιμοποιώντας τη θύρα υπέρυθρων, ήταν επίσης το πρώτο παιχνίδι δύο παικτών για κινητά τηλέφωνα.

Με την διάδοση των έξυπνων τηλεφώνων (Smartphones) στον 21ο αιώνα, τα οποία διαθέτουν ισχυρότερη υπολογιστική ισχύ τα παιχνίδια σε συσκευή κινητής τηλεφωνίας απέκτησαν μεγάλη δημοτικότητα. Το 2008 η Apple ξεκίνησε το App Store, η πρώτη αγορά κινητού περιεχομένου που λειτουργούσε απευθείας από έναν κάτοχο κινητής συσκευής. Επομένως Με την κυκλοφορία του App Store ο αριθμός των εξαιρετικά επιτυχημένων παιχνιδιών κινητής τηλεφωνίας πολλαπλασιάστηκε γρήγορα. Επιτυχίες όπως Angry Birds, Rolando, Flight Control, Doodle Jump εισήγαγαν πολλά εκατομμύρια νέους παίκτες στα παιχνίδια σε κινητό τηλέφωνο.



Εικόνα 1.1.2 : Το παιχνίδι Angry Birds

1.2 Ιστορία πλατφόρμας Android

Το Android είναι λειτουργικό σύστημα για συσκευές κινητής τηλεφωνίας το οποίο τρέχει τον πυρήνα του λειτουργικού Linux. Αρχικά αναπτύχθηκε από την Google και αργότερα από την Open Handset Alliance. Επιτρέπει στους κατασκευαστές λογισμικού να συνθέτουν κώδικα με την χρήση της γλώσσας προγραμματισμού Java, ελέγχοντας την συσκευή μέσω βιβλιοθηκών λογισμικού ανεπτυγμένων από την Google. Το Android είναι κατά κύριο λόγο σχεδιασμένο για συσκευές με οθόνη αφής, όπως τα έξυπνα τηλέφωνα και τα τάμπλετ, με διαφορετικό περιβάλλον χρήσης για τηλεοράσεις (Android TV), αυτοκίνητα (Android Auto) και ρολόγια χειρός (Android Wear). Παρόλο που έχει αναπτυχθεί για συσκευές με οθόνη αφής, έχει χρησιμοποιηθεί σε κονσόλες παιχνιδιών, ψηφιακές φωτογραφικές μηχανές, συνηθισμένους Η/Υ και σε άλλες ηλεκτρονικές συσκευές.
Το Android είναι το πιο ευρέως διαδεδομένο λογισμικό στον κόσμο έχουν περισσότερες πωλήσεις από όλες τις συσκευές Windows, iOS και Mac OS X μαζί. Η πρώτη παρουσίαση της πλατφόρμας Android έγινε στις 5 Νοεμβρίου 2007 και η πρώτη εμπορική έκδοση ήταν το Android 1.0 που κυκλοφόρησε το Σεπτέμβριο του 2008. Από τον Απρίλιο του 2009, οι εκδόσεις του Android έχουν θέμα από την ζαχαροπλαστική στην κωδική ονομασία τους, και κυκλοφόρησαν σε αλφαβητική σειρά, εξαιρουμένων των εκδόσεων 1.0 και 1.1, που δεν τέθηκαν υπό συγκεκριμένα κωδικά ονόματα:

Κωδικό όνομα	Νούμερο έκδοσης	Ημερομηνία αρχικής κυκλοφορίας	Επίπεδο API
N/A	1.0	23 Σεπτεμβρίου 2008	1
	1.1	9 Φεβρουάριου 2009	2
Cupcake	1.5	27 Απριλίου 2009	3
Donut	1.6	15 Σεπτεμβρίου 2009	4
Eclair	2.0 – 2.1	26 Οκτωβρίου 2009	5–7
Froyo	2.2 – 2.2.3	20 Μαΐου 2010	8
Gingerbread	2.3 – 2.3.7	6 Δεκεμβρίου 2010	9–10
Honeycomb	3.0 – 3.2.6	22 Φεβρουάριου 2011	11–13
Ice Cream Sandwich	4.0 – 4.0.4	18 Οκτωβρίου 2011	14–15
Jelly Bean	4.1 – 4.3.1	9 Ιουλίου 2012	16–18
KitKat	4.4 – 4.4.4	31 Οκτωβρίου 2013	19–20
Lollipop	5.0 – 5.1.1	12 Νοεμβρίου 2014	21–22
Marshmallow	6.0 – 6.0.1	5 Οκτωβρίου 2015	23
Nougat	7.0 - 7.1.1	22 Αυγούστου 2016	24
Oreo	8.0 – 8.1	21 Αυγούστου 2017	26 – 27
Pie	9.0	6 Αυγούστου 2018	28

Πίνακας 1.2: Πηγή: el.wikipedia.org

1.3 Παρουσίαση παιχνιδιού

Το παιχνίδι που δημιουργήθηκε στα πλαίσια της πτυχιακής εργασίας είναι ένα διδιάστατο παιχνίδι εξόρυξης. Το παιχνίδι έχει δύο λειτουργίες, μία για έναν παίκτη και μία για δύο παίκτες μέσω δικτύου.
Ο κάθε παίκτης χειρίζεται έναν ανθρακωρύχο που ψάχνει για πολύτιμα ορυκτά σε ένα παλιό ορυχείο. Σκοπός είναι να βρει τα δύο διαμάντια και να κάνει την τύχη του πριν από τον άλλον (στην λειτουργία δυο παικτών) και χωρίς να εγκλωβιστεί στο ορυχείο. Το ορυχείο δημιουργείται τυχαία κάθε φορά που γίνεται ένα νέο παιχνίδι.
Μπορεί να πουλήσει τα ορυκτά που μαζεύει για να αγοράσει καλύτερη αξίνα για γρηγορότερο σκάψιμο ή μεγαλύτερο σακίδιο για να χωράει περισσότερα ορυκτά, επίσης μπορεί να αγοράσει σκάλες ή ακόμα και ενεργειακό ποτό ώστε να μην εγκλωβιστεί στο ορυχείο.
Στην λειτουργία για έναν παίκτη υπάρχει η δυνατότητα αποθήκευσης του παιχνιδιού για να συνεχιστεί σε κάποια άλλη χρονική στιγμή.

1.3.1 Χειρισμός παιχνιδιού

Για την χρήση όλων των μενού αρκεί ένα πάτημα στην επιλογή που θέλει ο χρήστης.
Εντός παιχνιδιού ο παίκτης μπορεί να χρησιμοποιήσει το touchpad,στην κάτω αριστερή γωνία της οθόνης, ως joystick για να μετακινήσει τον παίκτη ή να σκάψει αν ακουμπάει χώμα ή ορυκτά. Επίσης στην κάτω δεξιά γωνία υπάρχει ένα κουμπί για να τοποθετήσει σκάλα στο σημείο που βρίσκεται ο χαρακτήρας, ώστε να μπορεί να σκαρφαλώνει έξω από το ορυχείο ή να φτάσει κάποια ορυκτά. Ακόμα με την χρήση δύο δαχτύλων μπορεί να κάνει «pinch» για να μεγεθύνει (zoom-in) ή να μικρύνει(zoom-out) την σκηνή του παιχνιδιού.

Εικόνα 1.3.1 : Εδώ φαίνεται η κύρια διεπαφή του χρήστη.

Συγκεκριμένα στην εικόνα 1.3.1 διακρίνουμε στην πάνω αριστερή γωνία μια πράσινη μπάρα που δείχνει την ενέργεια του παίκτη, μια μπλε που δείχνει την εμπειρία του και πόσο θέλει ακόμα για να ανέβει επίπεδο και από κάτω της το βάθος που βρίσκεται αυτή την στιγμή σε μπλοκ. Πάνω δεξιά δείχνει πόσο χώρο έχει ακόμα στο σακίδιο του. Κάτω αριστερά είναι το touchpad (joystick) κίνησης, και κάτω δεξιά το κουμπί για τοποθέτηση σκάλας. Επίσης βλέπουμε ρωγμές από ένα μπλοκ χώμα που έχει αρχίσει να σκάβει.

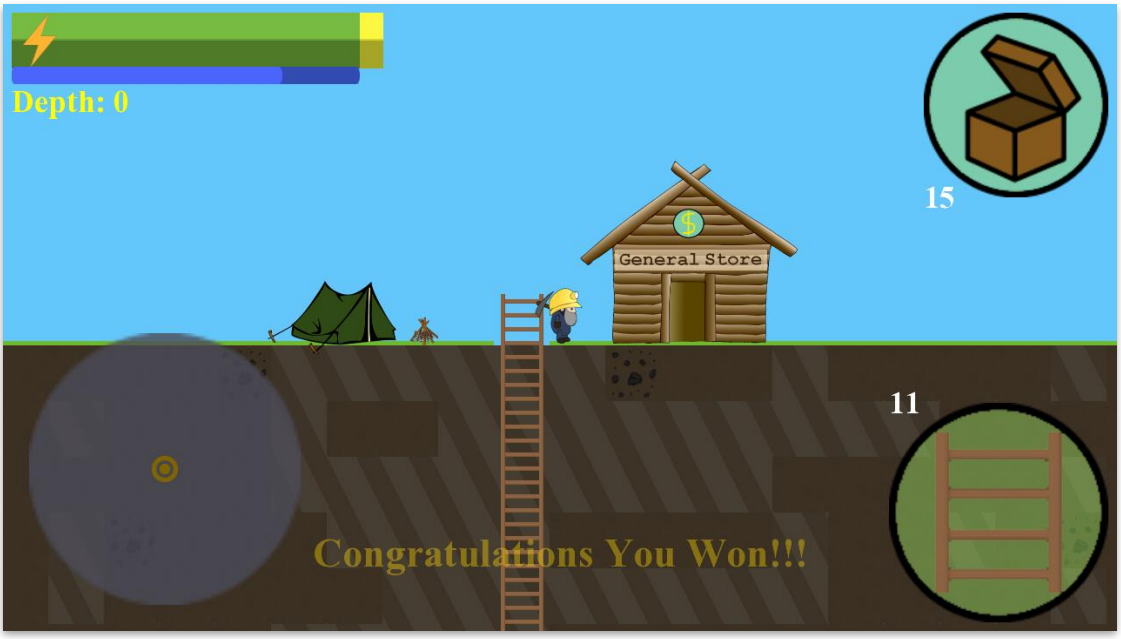
1.3.2 Σκηνές παιχνιδιού

Με το άνοιγμα του παιχνιδιού ξεκινάει αυτόματα η πρώτη σκηνή που είναι το βασικό μενού. Σε αυτό ο χρήστης μπορεί να διαλέξει την λειτουργία που θέλει να παίξει, για έναν παίκτη (singleplayer) ή για δύο (multiplayer) καθώς και την επιλογή ρυθμίσεις (settings) όπου μπορεί να αλλάξει το μέγεθος του κόσμου που θα δημιουργηθεί, το μέγεθος των κουμπιών και αν θέλει ήχο ή όχι στο παιχνίδι.
Αν ο χρήστης επιλέξει την λειτουργία για έναν παίκτη τότε του βγάζει επιλογή αν θέλει νέο παιχνίδι (new game) ή να φορτώσει (load game) το προηγούμενο ώστε να το συνεχίσει (αν το είχε αποθήκευση πρώτα).
Αν ο χρήστη επιλέξει την λειτουργία για δύο παίκτες του βγάζει επιλογή αν θέλει να δημιουργήσει αυτός το παιχνίδι (host game) ή να συνδεθεί σε παιχνίδι άλλου(join game).
Ο χάρτης του παιχνιδιού αποτελείται από 2 τμήματα. Το πρώτο είναι το τμήμα εκτός του ορυχείου (βλ. εικόνα 1.3.2). Περιέχει στα αριστερά την κατασκήνωση του παίκτη όπου μπορεί να κάνει αποθήκευση του παιχνιδιού του και στα δεξιά βρίσκεται το κατάστημα όπου μπορεί να αγοράσει σκάλες και να αναβαθμίσει την αξίνα του ή το σακίδιο του ώστε να μπορεί να κουβαλήσει περισσότερα ορυκτά πάνω του. Ανάμεσα από αυτά τα δύο βρίσκεται η είσοδος του ορυχείου.

Εικόνα 1.3.2 : Το τμήμα εκτός ορυχείου και η είσοδός σε αυτό.

Το δεύτερο τμήμα είναι το τμήμα μέσα στο ορυχείο όπου ο παίκτης μπορεί να βρει πολύτιμα ορυκτά για να μαζέψει, ενοχλητικούς βράχους να τον πλακώσουν ή δηλητηριώδη αέρα να τον πνίξουν (βλ. εικόνα 1.3.3). Καθώς πηγαίνει όλο και πιο βαθιά η σκηνή δυσκολεύει καθώς οι βράχοι και τα αέρια πληθαίνουν αλλά όσο πιο βαθιά πάει τόσο πιο πολύτιμα ορυκτά θα βρει.
Τέλος αν ο παίκτης καταφέρει να βρει τα διαμάντια και να βγει στην επιφάνεια θα δει μια σκηνή ανάλογη της εικόνας 1.3.4 όπου βλέπουμε screenshot από αληθινό παίκτη να έχει νικήσει το παιχνίδι.

Εικόνα 1.3.3 : Το τμήμα εντός ορυχείο



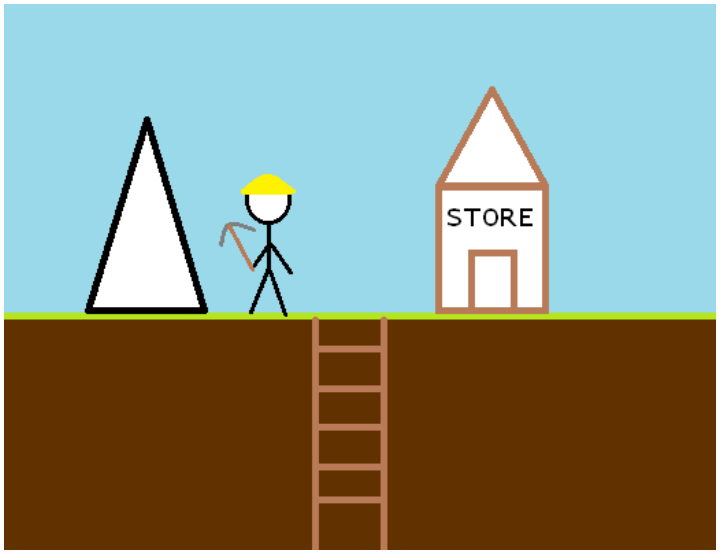
Εικόνα 1.3.4 : Τελική σκηνή νίκης από πραγματικό παίκτη.

1.4 Βήματα Υλοποίησης Παιχνιδιού

Παρακάτω αναφέρονται εν συντομία η σειρά των βημάτων που ακολουθήθηκαν για τον σχεδιασμό και την υλοποίηση του παιχνιδιού που δημιουργήθηκε για την παρούσα πτυχιακή.

Στάδιο Ιδεών και Προτυποποίησης:

Δημιουργία μιας πρώτης έκδοσης του παιχνιδιού στην οποία τα γραφικά είναι απλά (placeholder art) κυρίως ζωγραφισμένα με το χέρι με βασικά χρώματα και χωρίς λεπτομέρειες. Επίσης φτιάχνονται οι βασικές λειτουργείες όπως το μενού και η κίνηση του παίκτη ώστε να μπορεί ο σχεδιαστής να καταλάβει πως δουλεύει το παιχνίδι.
Αναζήτηση εικόνων στο διαδίκτυο και μελέτη παρόμοιων παιχνιδιών για ιδέες.
Δημιουργία προσχέδιων αναφοράς του παιχνιδιού , τα οποία θα χρησιμοποιηθούν στην συνεχεία ως βάση για την δημιουργία του παιχνιδιού.



Εικόνα 1.4.1: Πρώτη σκηνή εκτός ορυχείου με placeholder γραφικά

Στάδιο Σχεδιασμού:

Δημιουργία του χαρακτήρα στο Adobe Illustrator με βάση τα placeholder γραφικά.
Επεξεργασία των εικόνων και textures που βρέθηκαν στο διαδίκτυο.
Δημιουργία των υφών (textures) για τα μοντέλα του παιχνιδιού όπως η σκάλα, το χώμα και τα ορυκτά.
Εισαγωγή κίνησης στον χαρακτήρα με την βοήθεια του Adobe Illustrator.

Στάδιο Υλοποίησης με τη βοήθεια του libGDX framework:

Σχεδιασμός της διεπαφής χρήστη (UI) με τη βοήθεια του Scene2d του framework.
Εισαγωγή φυσικής (Physics) με την χρήση της μηχανής Box2d του framework.
Προγραμματισμός των λειτουργιών του παιχνιδιού σε Java.

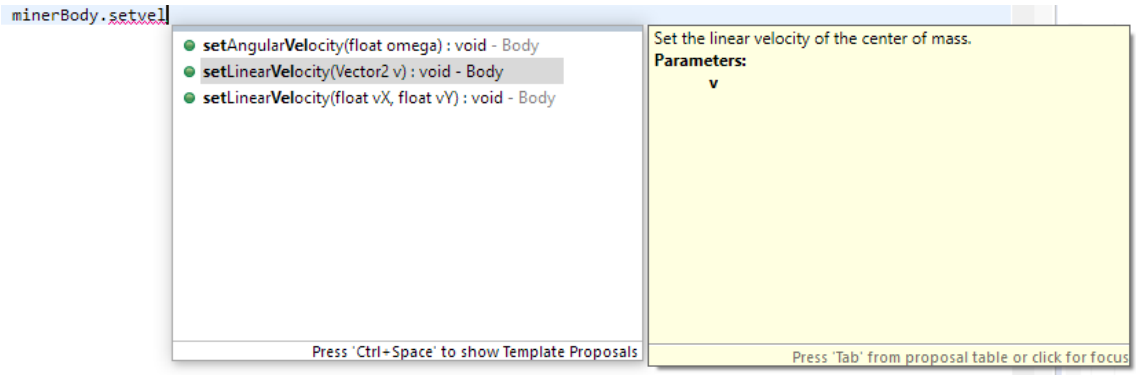
Όλα τα παραπάνω θα αναλυθούν σε βάθος στα επόμενα κεφάλαια.

1.5 Περίληψη Προγραμμάτων

Για την δημιουργία του τελικού παιχνιδιού χρησιμοποιήθηκαν διάφορα εργαλεία τα οποία περιγράφονται στην συνέχεια.

1.5.1 Eclipse IDE

Το Eclipse IDE είναι ένα ολοκληρωμένο περιβάλλον ανάπτυξης (integrated development environment, IDE) που χρησιμοποιείται στον Προγραμματισμό Υπολογιστών (Computer Programming). Κυρίως χρησιμοποιείται ως περιβάλλον ανάπτυξης προγραμμάτων. Ένα πολύ εύχρηστο εργαλείο του Eclipse είναι το content assist το οποίο βοηθάει την διαδικασία του προγραμματισμού διότι ο χρήστης δεν χρειάζεται να θυμάται ακριβώς τα ονόματα των μεθόδων και των μεταβλητών (βλ. εικόνα 1.4). Αυτό το πετυχαίνει ελέγχοντας το κείμενο που έγραψε ο χρήστης για ορθογραφικά λάθη και του δίνει προτάσεις για την συμπλήρωση της εντολής του κώδικα. Επειδή το Eclipse IDE δεν υποστηρίζει τις νέες εκδόσεις λογισμικού Android (από την 8 και άνω), για αυτό μετά χρησιμοποιήθηκε το Android Studio.

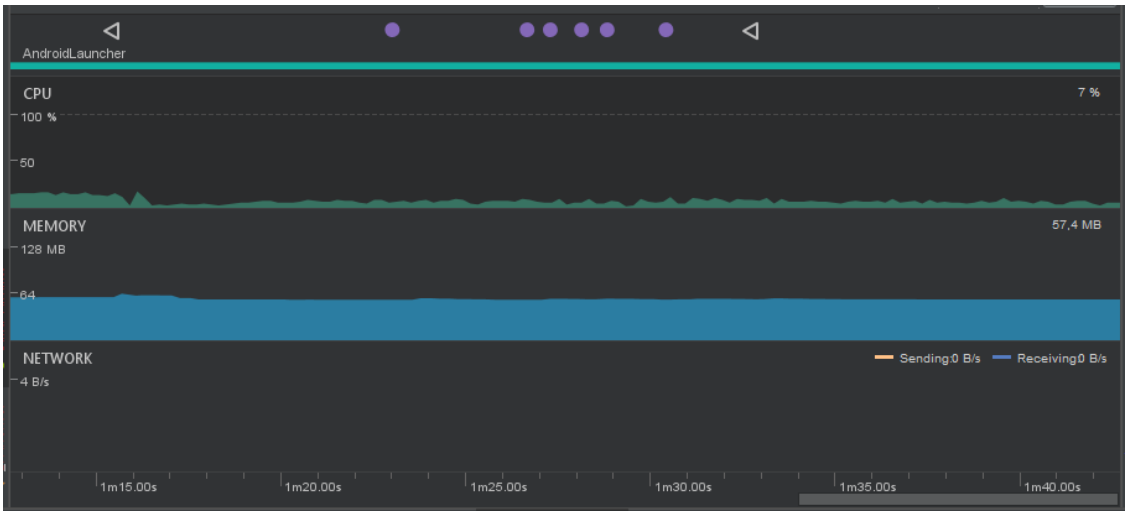


Εικόνα 1.4: Το σύστημα content assist προτείνει και εξηγεί τον κώδικα σύμφωνα με ότι έγραψε ο χρήστης.

1.5.2 Android Studio

Το Android Studio είναι το πλέον επίσημο ολοκληρωμένο περιβάλλον ανάπτυξης (integrated development environment, IDE) για το λογισμικό Android της Google το οποίο αντικατέστησε το Eclipse Android Development Tools. Ένα χρήσιμο εργαλείο του Android Studio είναι το Android Profiler (βλ. εικόνα 1.5) όπου ο χρήστης μπορεί να δει πόση μνήμη καταλαμβάνει η εφαρμογή του, πόσο επεξεργαστή και δίκτυο χρησιμοποιεί καθ' όλη την διάρκεια που είναι ανοικτή η εφαρμογή, καθώς και οι επαφές και τα πατήματα κουμπιών που έγιναν (τα σύμβολα στην κορυφή της εικόνας 1.5).

Μια νέα λειτουργία του Android Studio είναι το Instant Run για συσκευές με λογισμικό Android 5.0 και άνω, το οποίο μπορεί να τρέξει και να περάσει τις αλλαγές από τον κώδικα στην συσκευή χωρίς να χτίσει (Build) όλη την εφαρμογή και πολλές φορές χωρίς να κάνει επανεκκίνηση την εφαρμογή. Χρησιμοποιείται για πιο γρήγορες αλλαγές στην εφαρμογή, πολύ χρήσιμο στις δοκιμές κώδικα ειδικά για αλλαγές στην εμφάνιση.



Εικόνα 1.5: Το Android Profiler δείχνει πόσο μνήμη, επεξεργαστή και δίκτυο χρησιμοποιεί η εφαρμογή

1.5.3 Corel Paintshop Pro

Το Paintshop Pro της Corel είναι ένα πρόγραμμα επεξεργασίας εικόνων, γραφικών και διανυσματικών γραφικών. Χρησιμοποιήθηκε για την δημιουργία και διόρθωση των γραφικών (textures) του παιχνιδιού.

1.5.4 Adobe Illustrator

Το Adobe Illustrator είναι ένας επεξεργαστής διανυσματικών γραφικών που αναπτύχθηκε και κυκλοφορεί στην αγορά από την Adobe Systems. Χρησιμοποιήθηκε για την δημιουργία του χαρακτήρα και των Animation του.

1.5.5 Cool Edit Pro

Το Cool Edit Pro της Syntrillium Software είναι ένα πρόγραμμα επεξεργασίας ήχου, που το 2003 το αγόρασε η Adobe και μετονομάστηκε σε Adobe Audition. Αλλά χρησιμοποιήθηκε η παλιά έκδοση διότι είναι πιο απλή και κατανοητή. Χρησιμοποιήθηκε για την βελτίωση των ήχων του παιχνιδιού που βρέθηκαν/δημιουργήθηκαν. Για παράδειγμα επεξεργασία διάρκειας και έντασης ήχου.

1.5.6 LibGDX

Το LibGDX είναι μια πλατφόρμα ανάπτυξης παιχνιδιών ανοιχτού κώδικα.
Προσφέρει δυνατότητα δημιουργίας δισδιάστατων ή τρισδιάστατων παιχνιδιών σε διάφορες πλατφόρμες (Desktop, Android, iOS, HTML)

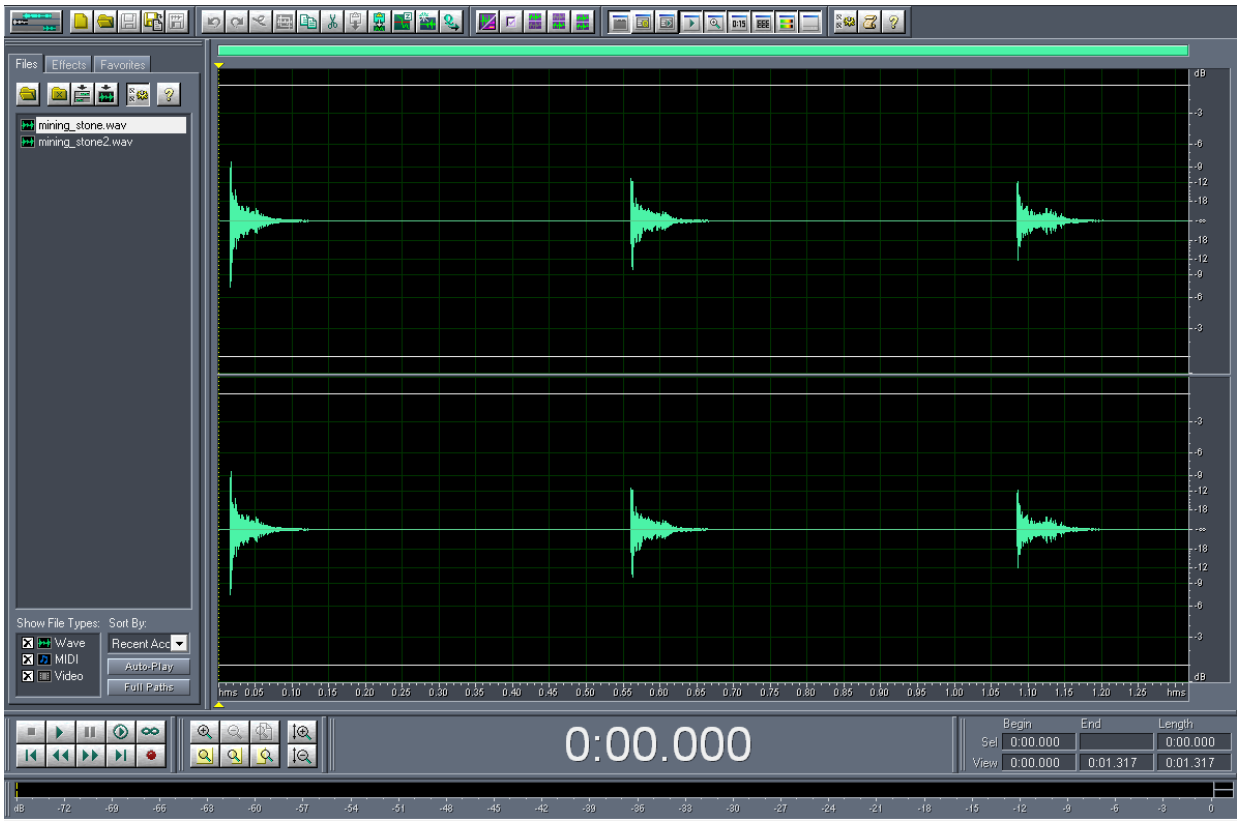
2. Δημιουργία Στοιχείων Παιχνιδιού

2.1 Εισαγωγή

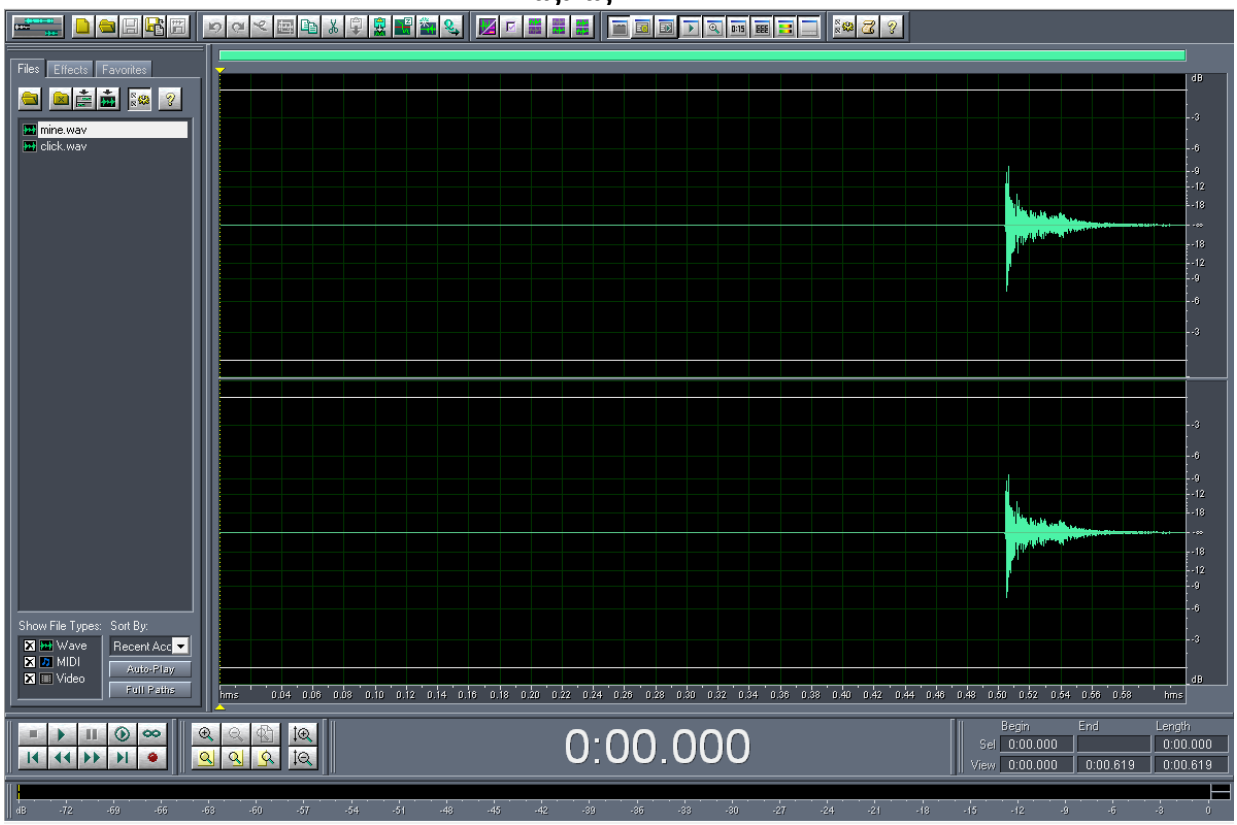
Για την δημιουργία οποιουδήποτε βιντεοπαιχνιδιού χρειάζονται σχεδιαστικά εργαλεία για να δημιουργηθούν όλα αυτά που βλέπει ο χρήστης, τα λεγόμενα γραφικά παιχνιδιού. Στην συγκεκριμένη περίπτωση, σε ένα διδιάστατο παιχνίδι τα γραφικά συνήθως αποτελούνται από εικόνες, όμως μπορεί να είναι και παραγόμενα από τον κώδικα.

2.2 Ήχοι παιχνιδιού

Ένα εξίσου σημαντικό ερέθισμα πέρα από την εικόνα είναι και ο ήχος. Για αυτό προστέθηκαν ήχοι, για τις διάφορες δράσεις του χαρακτήρα όπως όταν ο χαρακτήρας σκάβει ή τοποθετεί μια σκάλα. Μια μέθοδος για δημιουργία ηχητικών εφέ είναι η ηχογράφηση τέτοιων ήχων, για παράδειγμα τα χτυπήματα του χεριού στο γραφείο. Όμως για το συγκεκριμένο παιχνίδι χρειαζόμασταν χτύπημα σε πέτρα (ο οποίος είναι πιο οξύς) οπότε έγινε με την δεύτερη μέθοδο, η οποία είναι η αναζήτηση δωρεάν ηχητικών εφέ στο διαδίκτυο.
Για τον ήχο της αξίνας που χτυπάει ένας βράχο χρησιμοποιήθηκε ο παρακάτω ήχος (https://freesound.org/people/ryanconway/sounds/240801/) από το site freesound.org, ο οποίος δεν ταίριαζε απόλυτα διότι είχε τρία χτυπήματα μαζί (βλ. εικόνα 2.1). Οπότε εδώ χρησιμοποιήθηκε το Cool Edit Pro για να «κόψουμε» τον ήχο ώστε να κρατήσουμε μόνο το ένα χτύπημα και να το δυναμώσουμε λίγο (βλ. εικόνα 2.2). Ακόμα αφήσαμε λίγο «κενό» στην αρχή για να ταιριάξει ο ήχος με το animation του χαρακτήρα.



Εικόνα 2.1: Cool Edit Pro (πάνω αριστερό κανάλι κάτω δεξί) αρχική μορφή του ήχου της αξίνας



Εικόνα 2.2: Cool Edit Pro τελική μορφή του ήχου της αξίνας

Για την τοποθέτηση της σκάλας και το πάτημα κουμπιού εντός παιχνιδιού βρέθηκε ένα ήχος από πάτημα κουμπιού από το ίδιο site.

2.3 Δημιουργία του παρασκηνίου

Συνήθως όλα τα γραφικά ενός παιχνιδιού είναι αποθηκευμένα σε κάποια εικόνα. Όμως αυτό δεν είναι απαραίτητο, ειδικά όταν πρόκειται για απλά πράγματα όπως ένας φόντο παρασκηνίου. Το παρασκήνιο (βλ. εικόνα 2.3) δημιουργήθηκε μόνο με χρήση κώδικα. Αναλυτικότερα, χρησιμοποιήθηκαν εντολές του είδους, πάρε αυτό το χρώμα και ζωγράφισε ένα παραλληλόγραμμο από αυτές τις συντεταγμένες (x1, y1) ως εκείνες (x2, y2). Σημαντικό ρόλο έχει η σειρά που ήταν γραμμένες αυτές οι εντολές διότι με αυτή την σειρά δημιουργείται το βάθος (layers) των σχημάτων. Συγκεκριμένα πρώτα ζωγραφίζεται ο ουρανός και μετά το γρασίδι ώστε να φαίνεται μπροστά από τον ουρανό το γρασίδι. Επίσης πρώτα ζωγραφίζεται το σκούρο καφέ στο χώμα έπειτα γίνονται πλάγιες παχιές γραμμές (παραλληλόγραμμα) με το πιο ανοιχτό καφέ όπως φαίνεται στην εικόνα 2.3. Όμως οι πλάγιες ανοιχτές γραμμές προεξείχαν από το σκούρο καφέ για αυτό και ζωγραφίστηκε το σκούρο γκρι που φαίνεται στις κάτω άκρες δεξιά και αριστερά.

Εικόνα 2.3: Φόντο δημιουργημένο εξολοκλήρου από τον κώδικα

2.4 Δημιουργία του χαρακτήρα

Ο χαρακτήρας δημιουργήθηκε από την αρχή στο Adobe illustrator. Αρχικά όλος ο χαρακτήρας δημιουργήθηκε με διανύσματα (vector) τα οποία έχουν πολλά θετικά σε αντίθεση με τα pixel γραφικά (raster) τα οποία είναι απλός, ζωγραφισμένες κουκίδες. Οι περισσότερες συσκευές όπως οθόνες και εκτυπωτές υποστηρίζουν raster γραφικά οπότε στο τέλος τα vector μετατρέπονται σε raster που είναι πιο απλά στην χρήση. Ένα διανυσματικό γραφικό είναι για παράδειγμα μια γραμμή που ξεκινάει από το σημείο 0,0 και πάει στο 3,4 με πάχος 2 πόντων και χρώμα κίτρινο. Οπότε τα διανυσματικά κουβαλάνε πολύ πληροφορία και για αυτό και επεξεργάζονται εύκολα χωρίς να αλλοιωθεί η ποιότητα του γραφικού. Ένας από τους μεγαλύτερους λόγους να επιλέξεις vector γραφικά για να σχεδιάσεις κάτι είναι ότι γίνονται μετατροπή σε raster σε ότι

μέγεθος θέλουμε χωρίς αλλοίωση της εικόνας και δεύτερον αλλάζεις πανεύκολα το χρώμα ενός vector γραφικού. Όπως φαίνεται και στην εικόνα 2.4 ο χαρακτήρας αποτελείται από διανύσματα τα οποία αποτελούνται από πολλά σημεία που έχουν κατεύθυνση. Ακόμα και το κεντρικό σώμα του χαρακτήρα που φαίνεται απλό αποτελείται από 17 σημεία.

Ο χαρακτήρας ξεκίνησε να γίνεται από το κέντρο προς τις άκρες. Δηλαδή έγινε πρώτα ένα οβάλ για το σώμα, κ άλλα 2 για τα πόδια. Έπειτα ένα στρογγυλό κράνος και η αξίνα που ήταν πιο εύκολα. Μετά το πρόσωπο, η γενειάδα και το χέρι. Σειρά είχαν οι λεπτομέρειες όπως το φαναράκι στο κράνος και αλλαγές στα σχήματα ώστε να μην είναι ομοιόμορφα σχήματα. Τέλος μπήκαν τα χρώματα και έγινε εξαγωγή σε απλό αρχείο εικόνας στις διαστάσεις που θέλαμε.

Εικόνα 2.4: Το Adobe Illustrator με επιλεγμένα κάποια layers του χαρακτήρα στα οποία φαίνονται τα σημεία των διανυσμάτων που δημιουργούν τα σχήματα. Αριστερά φαίνονται όλα τα layers του χαρακτήρα.

2.5 Δημιουργία κίνησης του χαρακτήρα

Οι κινήσεις (animation) στα δισδιάστατα γραφικά είναι συνήθως πολλές διαφορετικές εικόνες που δείχνονται με μια σειρά η μία μετά την άλλη. Ένας άλλος τρόπος είναι να είναι ξεχωριστά αποθηκευμένα τα κομμάτια που θέλουμε να κινούνται (πχ. Χέρια, πόδια) ώστε να μπορούσε να τα διαχειριστεί μεμονωμένα ο κώδικας και να τους δώσει κίνηση.

Στην συγκεκριμένη περίπτωσή χρησιμοποιήθηκε το θετικό στοιχείο των διανυσματικών γραφικών να μετακινούνται χωρίς αλλοίωση και το ότι κάθε κομμάτι του χαρακτήρα είναι ξεχωριστό layer. Με αυτά μετακινώντας μόνο τα πόδια όπως φαίνεται στην εικόνα

2.5 δημιουργήθηκαν σιγά σιγά οι πέντε διαφορετικές εικόνες για την κίνηση του χαρακτήρα.

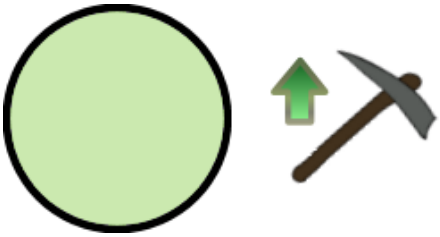
Εικόνα 2.5: Έχοντας επιλέξει στο Illustrator μόνο το ένα πόδι του χαρακτήρα

Όπως βλέπουμε στην εικόνα 2.6 δεν είναι ολοκληρωμένη η κίνηση του χαρακτήρα, δεν χρειάζεται όμως. Χρησιμοποιώντας την τεχνική ping pong παίζονται με την σειρά οι εικόνες και μόλις φτάσει στην τελευταία παίζονται με την σειρά προς τα πίσω. Έτσι έχουμε μια ολοκληρωμένη κίνηση έχοντας μόνο τα μισά καρέ.

Εικόνα 2.6: Όλες οι εικόνες για την κίνηση του χαρακτήρα

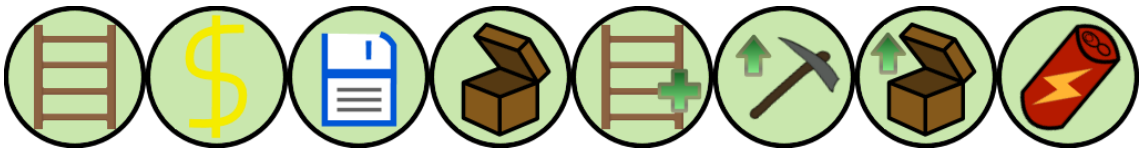
2.6 Δημιουργία γραφικών διεπαφής χρήστη

Όλα τα γραφικά για την διεπαφή του χρήστη δημιουργήθηκαν από την αρχή με την βοήθεια του προγράμματος Paintshop Pro της Corel. Τα κουμπιά του μενού και το joystick είναι από κατεβασμένο θέμα (skin) του Scene2d.
Αρχικά για την δημιουργία των κουμπιών (για τις αγορές, την τοποθέτηση σκάλας κτλ.) δημιουργήθηκε ένας βασικός κύκλος με ελαφρύ λαχανί γέμισμα ως βάση (βλ. εικόνα 2.7 αριστερά). Έπειτα δημιουργήθηκαν ένα-ένα σαν ξεχωριστά εικόνα το εσωτερικό κάθε κουμπιού (βλ. εικόνα 2.7 δεξιά).



Εικόνα 2.7: Το αρχικό layer για τα κουμπιά της διεπαφής (αριστερά).
Το εσωτερικό ενός κουμπιού (δεξιά)

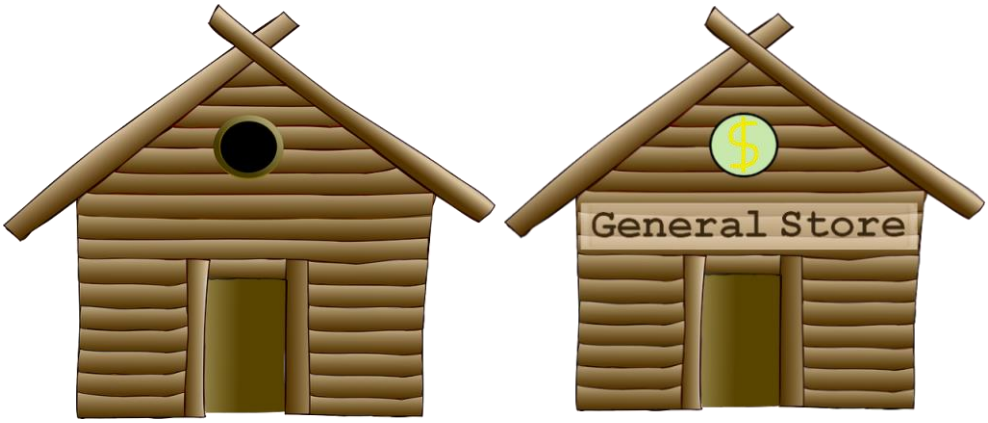
Το κύριο εργαλείο που χρησιμοποιήθηκε από το Paintshop για την δημιουργία των εικόνων είναι το pen tool. Με την βοήθεια του φτιάχτηκαν διανυσματικά γραφικά. Με πολλή ευκολία γίνονται οι καμπύλες όπως φαίνονται στην εικόνα 2.8, το δολάριο, η δισκέτα, το κουτί και το αναψυκτικό. Αφού φτιάχτηκε το κάθε κουμπί τοποθετήθηκαν το καθένα πάνω στην αρχική βάση που δημιουργήθηκε. Έτσι έγινε το τελικό αποτέλεσμα που βλέπουμε στην εικόνα 2.8.



Εικόνα 2.8: Όλα τα κουμπιά της διεπαφής του χρήστη

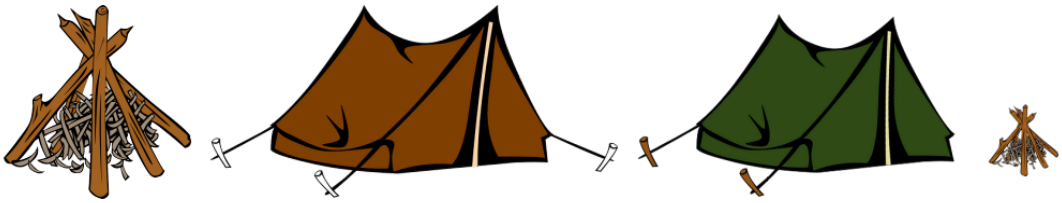
2.7 Δημιουργία υπόλοιπων γραφικών

Για τα υπόλοιπα γραφικά του παιχνιδιού βρέθηκαν δωρεάν εικόνες στο διαδίκτυο όπου μετά από επεξεργασία ταίριαξαν στο υπόλοιπο παιχνίδι. Εδώ επίσης, χρησιμοποιήθηκε το Paintshop Pro της Corel για την επεξεργασία των εικόνων.
Συγκεκριμένα, για το κατάστημα του παιχνιδιού βρέθηκε μια καλύβα όπου μεταμορφώθηκε σε κατάστημα μαζί με το γραφικό αγοράς που είχε φτιαχτεί από την διεπαφή του χρήστη (βλ. εικόνα 2.9). Φτιάχτηκε λίγο η εικόνα προστέθηκε μια πινακίδα και άλλαξε τελείως το παραθυράκι της καλύβας.



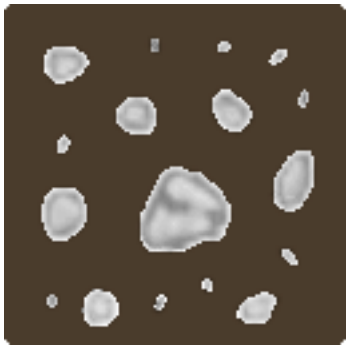
Εικόνα 2.9: Το κατάστημα (δεξιά) και η αρχική καλύβα (αριστερά)

Για την σκηνή του χαρακτήρα βρέθηκαν δύο εικόνες οι οποίες ενώθηκαν και βγήκε το τελικό αποτέλεσμα (βλ. εικόνα 2.10). Πιο αναλυτικά, αρχικά χρησιμοποιήθηκε το straighten tool πάνω στην σκήνη για να την «ισιώσει» ώστε να μην φαίνεται τόσο η προοπτική και για να ταιριάζει πιο πολύ με το κατάστημα. Έπειτα χρησιμοποιήθηκε το color changer για να αλλάξει από το καφέ της χρώμα στο πράσινο. Έπειτα μικραίνουμε κατάλληλα την εικόνα με τα ξύλα ώστε να ταιριάξει μπροστά από την σκηνή.

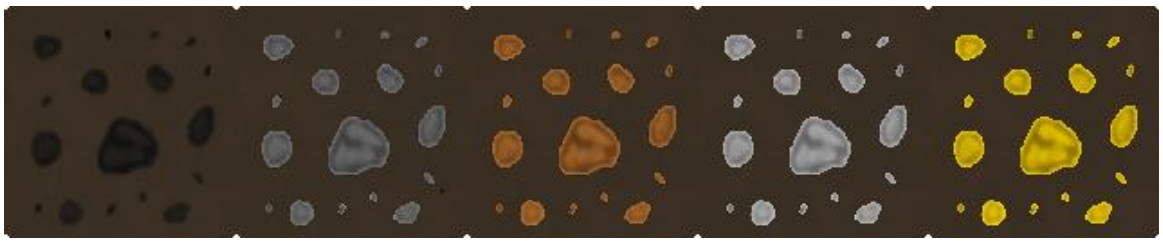


Εικόνα 2.10: Η τελική σκηνή στα δεξιά και οι δύο αρχικές εικόνες στα αριστερά

Τελικά απέμειναν τα γραφικά για τα ορυκτά που σκάβει ο παίκτης. Αυτά φτιάχτηκαν με ιδέες από γραφικά παρόμοιων παιχνιδιών. Αρχικά είχαμε ένα απλό καφέ τετράγωνο με άσπρα ακανόνιστα σχήματα όπως βλέπουμε στην εικόνα 2.11. Χρησιμοποιώντας το color changer του Paintshop αλλάζαμε το άσπρο χρώμα προς όλα τα χρώματα των ορυκτών που θέλαμε. Έπειτα βρήκαμε μια υφή από πέτρα (texture) και την κάναμε blend στην υπάρχουσα εικόνα και έχουμε τα τελικά ορυκτά (βλ. εικόνα 2.12).

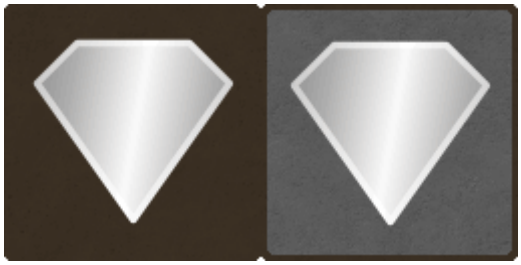


Εικόνα 2.11: Το βασικό ορυκτό



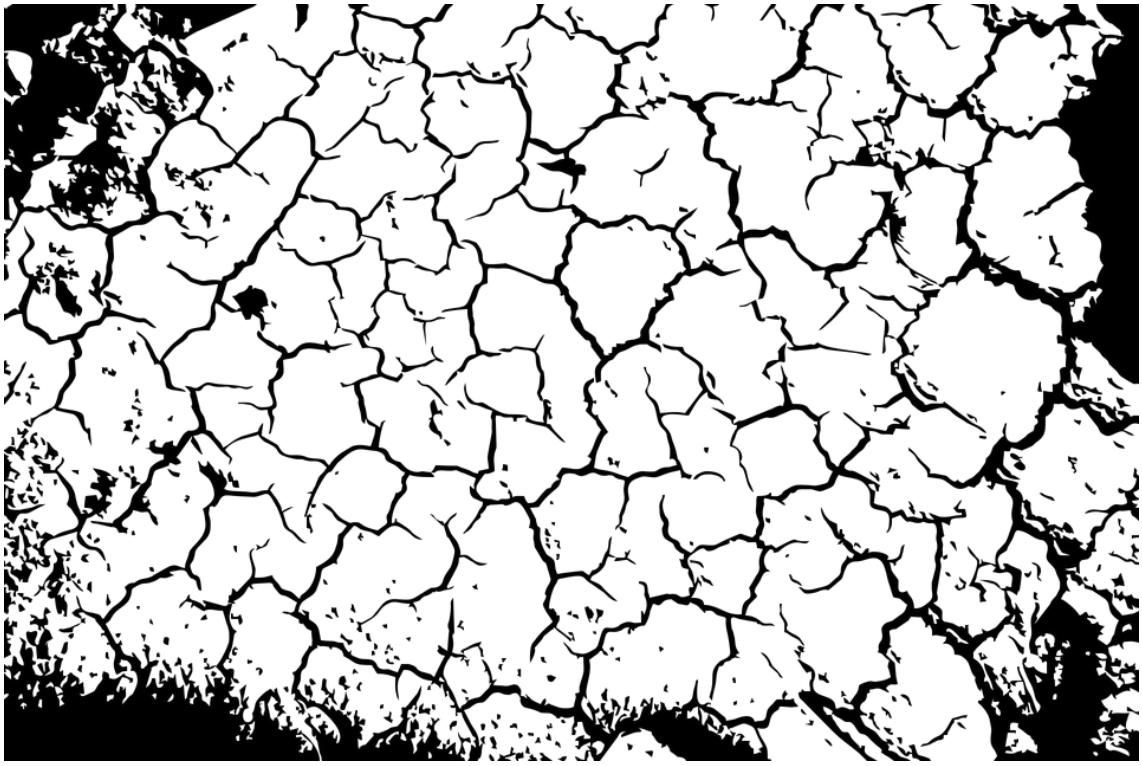
Εικόνα 2.12: Τελικά ορυκτά

Τα διαμάντια δημιουργήθηκαν από την αρχή με το pen tool ως διανυσματικά γραφικά με gradient γέμισμα κοιτώντας κάποια παρόμοια εικόνα στο διαδίκτυο. Έπειτα μπήκαν πάνω στο χώμα και στον βράχο αντίστοιχα όπως φαίνεται στην εικόνα 2.13.



Εικόνα 2.13: Τα δύο διαμάντια που πρέπει να βρει ο παίκτης

Ακόμα για την δημιουργία των ρωγμών στα ορυκτά όταν πάει να σπάσει ένα, επεξεργάστηκε η εικόνα 2.14. Χρειαζόμασταν τρεις εικόνες στο ίδιο μέγεθος με το ορυκτό ώστε να δείχνουμε πότε κοντεύει να σπάσει το ορυκτό. Μία με λίγες ρωγμές, μία με περισσότερες και μία με ακόμα πιο πολλές. Έτσι κόπηκαν κομμάτια από την αρχική εικόνα και δημιουργήθηκαν οι τρεις μικρότερες εικόνες (βλ. εικόνα 2.15), οι οποίες ζωγραφίζονται πάνω από το ορυκτό που κοντεύει να σπάσει ο παίκτης.



Εικόνα 2.14: Μια εικόνα με ρωγμές που βρέθηκε στο διαδίκτυο

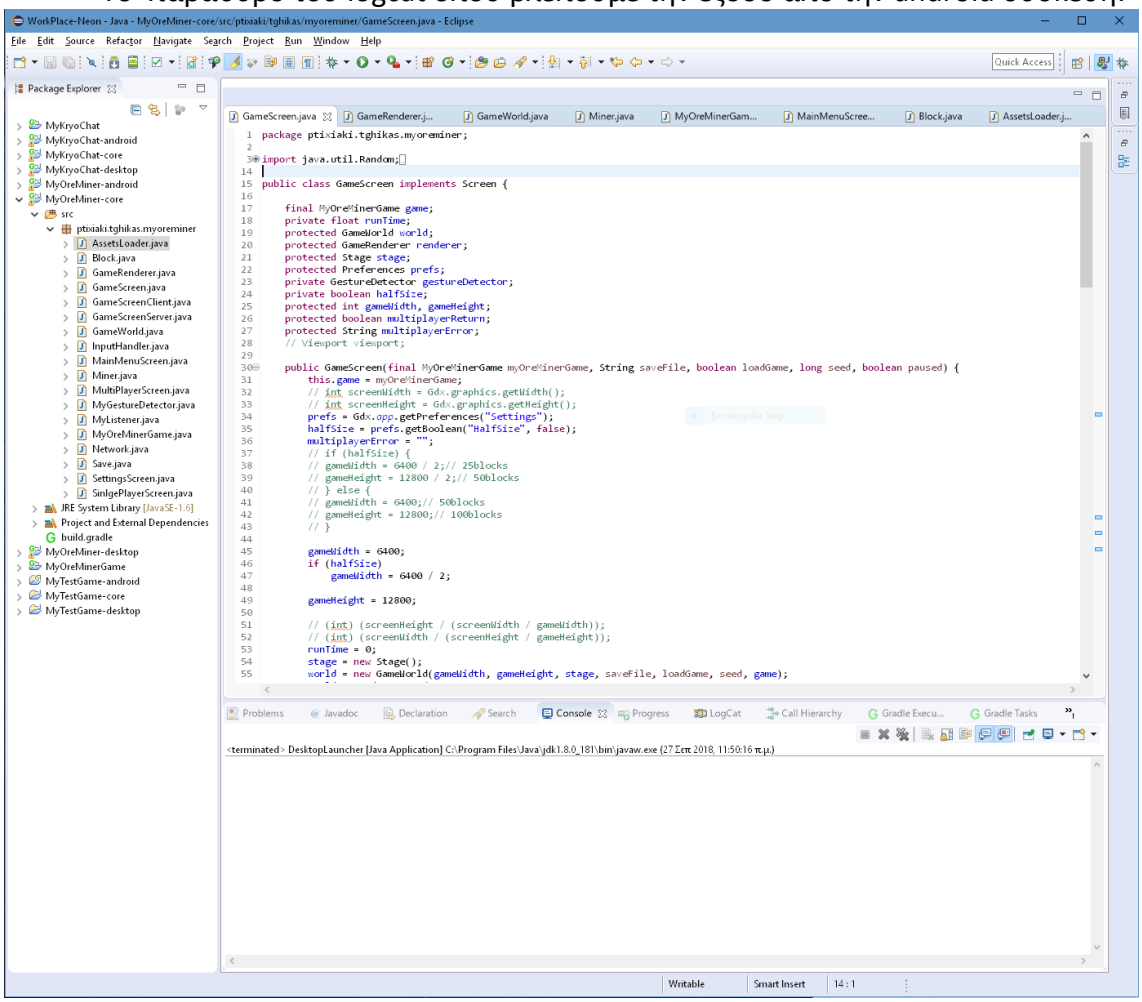
Εικόνα 2.15: Οι τρεις μικρότερες εικόνες για τις ρωγμές

3. Σχεδιασμός Παιχνιδιού

3.1 Εισαγωγή

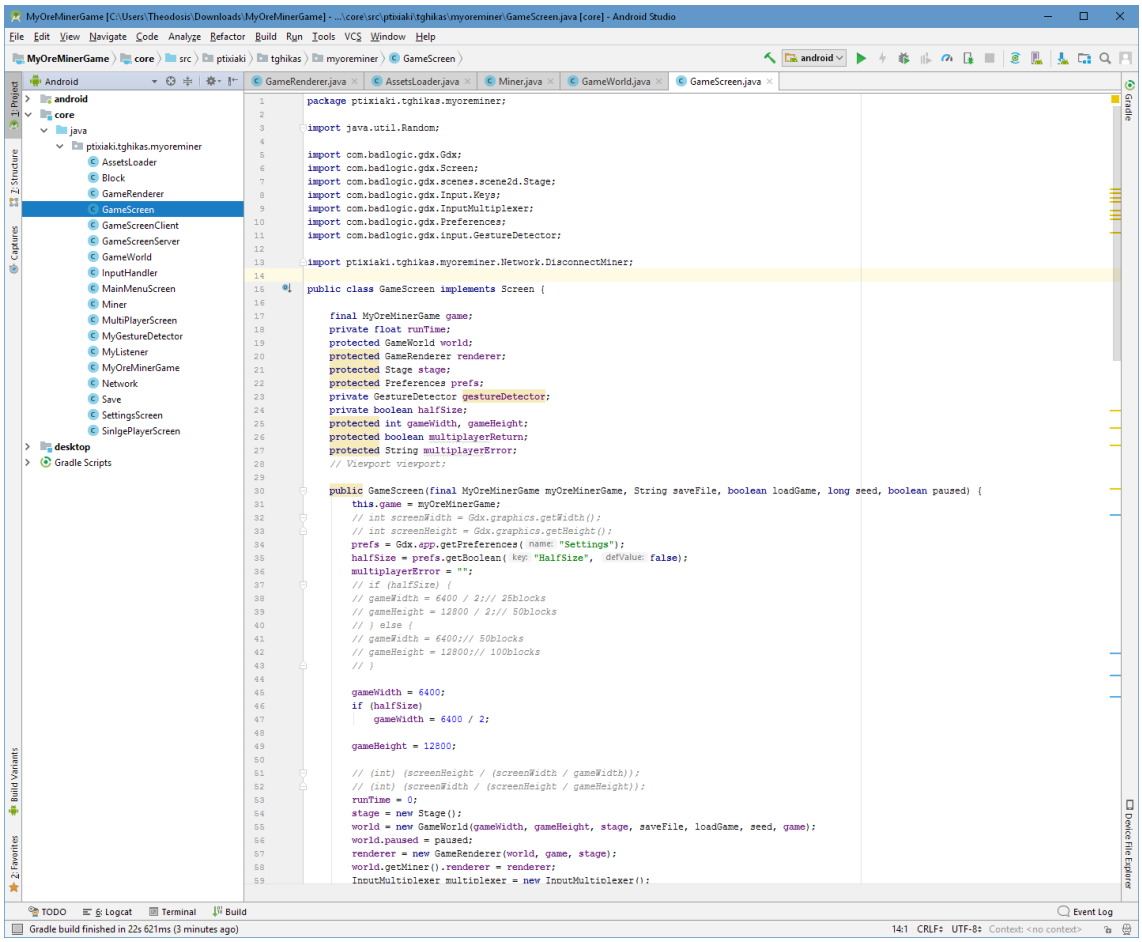
Αρχικά το βιντεοπαιχνίδι άρχισε να σχεδιάζεται στο ολοκληρωμένο περιβάλλον του Eclipse με την γλώσσα προγραμματισμού Java (www.java.com). Χρησιμοποιώντας το αρχείο εγκατάστασης του libGDX Framework libgdx.badlogicgames.com/nightlies/dist/gdx-setup.jar δημιουργήθηκε το project για το Eclipse με ότι ρυθμίσεις και βιβλιοθήκες χρειαζόμασταν, όπως την βιβλιοθήκη box2d. Στην εικόνα 3.1 φαίνεται το περιβάλλον του Eclipse. Το περιβάλλον αποτελείται από πολλά παράθυρα αλλά 5 είναι τα βασικά.

- Το παράθυρο εξερευνητή πακέτου (package explorer) που βασικά μας δείχνει τα αρχεία, τους φακέλους και τις βιβλιοθήκες που χρησιμοποιούνται σε επίπεδο πακέτου.
- Το κεντρικό παράθυρο του κώδικα όπου γράφουμε τον κώδικα μας.
- Το παράθυρο με τα προβλήματα που μας βγάζει αν υπάρχουν σφάλματα στον κώδικα μας.
- Το παράθυρο της κονσόλας που βλέπουμε την έξοδο όταν τρέχουμε το project.
- Το παράθυρο του logcat όπου βλέπουμε την έξοδο από την android συσκευή.



Εικόνα 3.1: Το περιβάλλον Eclipse

Έπειτα αφού βγήκε νέα έκδοση android που δεν υποστήριζε πλέον το ADT (android developer tools) plugin του eclipse, το project έπρεπε να προχωρήσει στο Android Studio, το οποίο ήταν πολύ εύκολο διότι το android studio έχει να κάνει εισαγωγή (import) project από το eclipse. Στην εικόνα 3.2 βλέπουμε το περιβάλλον Android Studio. Το Android Studio αποτελείται από παρόμοια παράθυρα με του Eclipse με την προσθήκη του Android Profiler όπου δείχνει πόσους πόρους καταναλώνει η εφαρμογή στην συσκευή Android.



Εικόνα 3.2: Το περιβάλλον Android Studio

3.2 LibGDX framework

Χρησιμοποιήθηκε το libGDX framework (libgdx.badlogicgames.com) διότι είναι ένα cross-platform εργαλείο για την δημιουργία βιντεοπαιχνιδιών σε Java. Πέρα από το βασικό setup του για την δημιουργία του project έχει και μερικά έξτρα εργαλεία που μπορούν να κατέβουν από εδώ https://libgdx.badlogicgames.com/tools.html. Ένα από τα εργαλεία που χρησιμοποιήθηκε ήταν το Hiero, το οποίο μετατρέπει αρχεία με γραμματοσειρές σε bitmap font τα οποία είναι η επιλεγμένη γραμματοσειρά αποθηκευμένη σε εικόνα για να την χρησιμοποιήσει το libGDX framework. Ένα άλλο εργαλείο που χρησιμοποιήθηκε ήταν το Texture packer το οποίο δέχεται σαν είσοδο ένα φάκελο με εικόνες και τις ενώνει όλες μαζί σε μία εικόνα όσο μπορεί καλύτερα και παράγει και ένα αρχείο που διαβάζει το framework ώστε να ξέρει που βρίσκεται κάθε ενωμένη εικόνα στο νέο αρχείο. Το libGDX αλληλεπιδρά με την κάρτα γραφικών μέσω

της OpenGL (Open Graphics Library), η οποία συνθέτει την εικόνα και μετά την ζωγραφίζει στην οθόνη, αυτό γίνεται για κάθε εικόνα που πρέπει να δείξει στην οθόνη, η σύνθεση αυτή είναι ακριβή διαδικασία (παίρνει ώρα). Οπότε έχοντας όλα τα γραφικά σε μία εικόνα, συνθέτει μια φορά και ζωγραφίζει όλα τα γραφικά. Ακόμα βοηθάει η εικόνα αυτή να είναι δύναμη του δύο το οποίο είναι προεπιλεγμένη ρύθμιση στο texture packer να δημιουργεί τέτοιες διαστάσεις.
Επειδή το libGDX είναι cross-platform framework ο κώδικας γράφεται μια φορά και τρέχει σε πολλές συσκευές. Στο συγκεκριμένο έργο της πτυχιακής, εκτός από android είχε επιλεχτεί να τρέχει και σε java desktop application. Οπότε πολλές δοκιμές έγιναν στο java application που έτρεχε πολύ πιο γρήγορα το debugging από ότι σε μια android συσκευή. Αυτό είναι πολύ βολικό για να δοκιμάζεις την λογική και τους μηχανισμούς του παιχνιδιού αν δουλεύουν σωστά. Προτιμήθηκε το libGDX framework διότι όλα γράφονται με κώδικα οπότε υπάρχει ο πλήρης έλεγχος.

3.3 Μηχανή φυσικής Box2d

Για να αλληλεπιδρά ο χαρακτήρας με τον υπόλοιπο κόσμου του παιχνιδιού πρέπει κάπως να λαμβάνεται υπόψιν το πότε συγκρούεται με κάποιο αντικείμενο, τα λεγόμενα collisions. Για αυτό χρησιμοποιήθηκε η box2d, η οποία είναι μια μηχανή φυσικής ανοιχτού κώδικα για παιχνίδια (box2d.org). Χωρίς αυτήν θα έπρεπε να γραφτεί κώδικας που να ελέγχει αν ο χαρακτήρας συγκρούστηκε με κάποιο αντικείμενο. Με το box2d ορίζουμε έναν κόσμο με βαρύτητα. Μέσα σε αυτόν τον κόσμο δημιουργούμε τα αντικείμενα (σώματα) που θέλουμε, όσο περίπλοκα και να είναι. Δηλαδή ένα σώμα μπορεί να αποτελείται από πολλά μέρη.
Μερικές παραμέτρους που μπορούμε ορίσουμε για κάθε αντικείμενο είναι:

- το σχήμα του,
- την πυκνότητα του,
- την ελαστικότητα του,
- αν θέλουμε να είναι σταθερό αντικείμενο ή όχι,
- αν θέλουμε να είναι αισθητήρας.

Σύμφωνα με αυτές τις παραμέτρους η μηχανή φυσικής box2d υπολογίζει τι πρέπει να γίνει όταν το ένα αντικείμενο συγκρουστεί με ένα άλλο. Ακόμα σε κάθε αντικείμενο (σώμα) υποστηρίζεται η προσθήκη οποιουδήποτε δεδομένου θέλει ο προγραμματιστής.

3.3.1 Συγκρούσεις (Collisions)

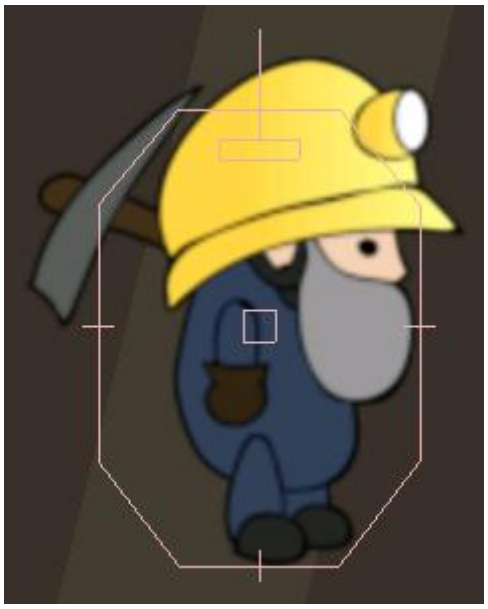
Όταν ένα αντικείμενο οριστεί ως σταθερό δεν μετακινείται με οποιαδήποτε σύγκρουση μόνο μέσω κώδικα. Συνήθως τέτοια αντικείμενα μπορεί να είναι οι άκρες της πίστας του παιχνιδιού όπου δεν θέλουμε τίποτα να πέσει έξω από αυτές.
Αν ένα αντικείμενο ή μέρος αντικειμένου οριστεί ως αισθητήρας (sensor) δεν προκαλεί μια φυσική σύγκρουση αλλά ένα συμβάν (event) σύγκρουσης , στο οποίο πρέπει να γραφτεί κώδικας για το τι θα συμβεί.
Εφόσον η μηχανή φυσικής υπολογίζει μόνη της τις συγκρούσεις των αντικείμενων πως μπορούμε να ορίσουμε τι θα συγκρουστεί με τι?
Εδώ έρχονται να λύσουν το πρόβλημα οι μάσκες.

Σε κάθε αντικείμενο (σώμα) του κόσμου, ορίζεται μια κατηγορία και μια μάσκα. Συγκεκριμένα η κατηγορία αναφέρεται στην κατηγορία που είναι το αντικείμενο και η μάσκα αναφέρεται με ποιες κατηγορίες αντικειμένων μπορεί να συγκρουστεί αυτό. Οπότε έτσι ορίζεται η σκάλα να μην συγκρούεται με τον χαρακτήρα αλλά να συγκρούεται με το χώμα και τα ορυκτά. Όμως πως ο χαρακτήρας ανεβαίνει την σκάλα? Έχοντας προσθέσει έναν αισθητήρα (sensor) στην σκάλα που έχει για μάσκα την κατηγορία του χαρακτήρα , όπως φαίνεται στην εικόνα 3.3. Έτσι όταν ο χαρακτήρας ακουμπήσει τη σκάλα δημιουργείται ένα συμβάν που έχει προγραμματιστεί να επιτρέπει στον χαρακτήρα να ανεβαίνει την σκάλα. Στην πραγματικότητα όσο ο χαρακτήρας είναι πάνω στην σκάλα αφαιρείται η επίδραση της βαρύτητας στον χαρακτήρα.

Εικόνα 3.3: Στο debug του κόσμου φαίνονται αντικείμενα με τα σχήματά τους. Η σκάλα έχει δύο μέρη, το πιο μικρό στο εσωτερικό είναι ο αισθητήρας που όταν το ακουμπάει ο χαρακτήρας του επιτρέπει να ανέβει πάνω. Το πράσινο περίγραμμα σημαίνει ότι το αντικείμενο αυτό είναι σταθερό.

3.3.2 Σύσταση Χαρακτήρα

Ο χαρακτήρας αποτελείται από πολλά μέρη για να λειτουργήσει όπως φαίνεται στην εικόνα 3.4. Αναλυτικότερα υπάρχει ένα κεντρικό οκτάγωνο που είναι το κύριο σώμα του χαρακτήρα που «συγκρούεται» με τα άλλα αντικείμενα (σώματα) του κόσμου. Οι τέσσερεις μικρές προεξοχές που φαίνονται πάνω κάτω αριστερά δεξιά είναι αισθητήρες που βοηθάνε στο σκάψιμο. Με αυτές ξέρει το πρόγραμμα πιο τετραγωνάκι (ορυκτό, χώμα) ακουμπάει ο χαρακτήρας ώστε να σκάψει προς τις αντίστοιχες πλευρές. Το μικρό τετραγωνάκι στο κέντρο είναι ένας ακόμα αισθητήρας που χρησιμοποιείται για να ελέγξει αν μπορεί να τοποθετήσει κάποιο αντικείμενο ο παίκτης στην θέση που βρίσκεται π.χ. μια σκάλα. Με άλλα λόγια να μην μπορεί να τοποθετήσει σκάλα μέσα σε άλλη σκάλα. Το μικρό παραλληλόγραμμο στο κεφάλι του χαρακτήρα εξυπηρετεί έναν συγκεκριμένο μηχανισμό του παιχνιδιού, τον έλεγχο αν «πλακώνει» κάποιος βράχος τον χαρακτήρα.



Εικόνα 3.3: Τα διάφορα μέρη του χαρακτήρα.

3.4 Διεπαφή χρήστη (UI)

Με την διεπαφή χρήστη εννοούμε όλα αυτά με τα οποία ο χρήστης μπορεί να αλληλοεπιδράσει με το παιχνίδι. Είτε αυτό είναι ένα κουμπί που πατάει στην οθόνη ο χρήστης είτε αυτό είναι η μπάρα ζωής του χαρακτήρα. Στο libGDX η διεπαφή φτιάχνεται με την βοήθεια της βιβλιοθήκης Scene2d.

3.4.1 Προβολή (Camera)

Καταρχήν το libGDX χρειάζεται μια προβολή για να ζωγραφίσει όλα αυτά που θέλει ο προγραμματιστής να φαίνονται στην οθόνη. Για αυτό φτιάχνεται ένα αντικείμενο τύπου Camera με συγκεκριμένες διαστάσεις. Αυτή η κάμερα μπορεί να είναι με προοπτική (perspective) ή ορθογραφική (orthographic). Η perspective κάμερα είναι για να απεικονίζει βάθος που χρειάζεται για τα τρισδιάστατα παιχνίδια. Όμως στο συγκεκριμένο παιχνίδι που δημιουργήθηκε χρειάστηκε μια orthographic κάμερα αφού είναι διδιάστατο και δεν χρειαζόμαστε βάθος. Έτσι με την ορθογραφική κάμερα όλα φαίνονται να είναι στο ίδιο βάθος. Οπότε το αντικείμενο που είναι υπεύθυνο για να ζωγραφίσει τα γραφικά χρησιμοποιεί την προβολή (κάμερα) για να τα ζωγραφίσει ώστε να φαίνονται στην οθόνη.

3.4.2 Διεπαφή με χρήση της Scene2d

Το Scene2d είναι ένα γράφημα δισδιάστατης σκηνής για την κατασκευή εφαρμογών και διεπαφών (UIs) χρησιμοποιώντας μια ιεραρχία από Actors. Εδώ χρησιμοποιήθηκε μόνο για την δημιουργία της διεπαφής με τον χρήστη. Για να χρησιμοποιηθεί το Scene2d χρειάζεται να δημιουργηθεί τουλάχιστον ένα αντικείμενο τύπου stage (σκηνή). Η σκηνή καταλαμβάνει ολόκληρη την οθόνη. Actors είναι όλα τα widget και UI

αντικείμενα που έχει το Scene2d, είτε αυτά είναι ένα κουμπί είτε ένας πίνακας για στοίχιση άλλων Actor. Με τον όρο ιεραρχία εννοείται ότι ένα actor μπορεί να έχει άλλα actors μέσα του, π.χ. υπάρχει Actor τύπου group που μπαίνουν άλλα Actors «μέσα» του και χρησιμοποιείται για ομαδοποίηση των Actor, αυτό βολεύει όταν θέλουμε να κάνουν πολλά actors την ίδια δράση (action). Όλα τα Actors υποστηρίζουν κάποια Actions. Τα actions χρησιμοποιούνται για τον χειρισμό των Actors. Δηλαδή όταν θέλουμε να προσθέσουμε κάποιο εφέ σε κάποιο actor ή να μετακινηθεί κάποιο actor χρησιμοποιούμε τα actions.

3.4.3 Τα γραφικά του Scene2d

Το Scene2d χρησιμοποιεί Skins για τα γραφικά του, με άλλα λόγια θέματα. Ο χρήστης μπορεί να φτιάξει το δικό του θέμα ή να χρησιμοποιηθεί κάποιο έτοιμο και δωρεάν. Στην σελίδα github.com/czyzby/gdx-skins μπορούν να βρεθούν skins για το scene2d καθώς οδηγίες και εργαλεία για την δημιουργία νέου skin. Για το παρόν παιχνίδι χρησιμοποιήθηκε και τροποποιήθηκε το θέμα flat-earth, αλλάχτηκαν και προστέθηκαν τα κάποια χρώματα.
Το skin συνήθως αποτελείται από τέσσερα αρχεία:

- Ένα αρχείο εικόνας .png με όλα τα γραφικά (κουμπιά, γράμματα, joystick κ.α.)
- Ένα αρχείο κειμένου .atlas με οδηγίες για το που βρίσκεται κάθε γραφικό στην εικόνα
- Ένα αρχείο κειμένου .json με οδηγίες για το τι χρώματα πρέπει να ζωγραφιστούν
- Ένα αρχείο κειμένου .fnt με οδηγίες για την γραμματοσειρά

Οδηγίες εξυπακούεται ότι χρειάζεται το πρόγραμμα για να διαβάσει το skin και να το χρησιμοποιήσει το Scene2d.

3.5 Δημιουργία της πίστας

Η δημιουργία της πίστας γίνεται τυχαία την στιγμή που γίνεται ένα νέο παιχνίδι. Θα μπορούσε να χρησιμοποιηθεί κάποιος αλγόριθμός θορύβου ή κάποιος απλός αλγόριθμος πιθανοτήτων. Εδώ χρησιμοποιήθηκε ένας αλγόριθμος παραγωγής τυχαίου αριθμού από την βιβλιοθήκη libGDX. Αναλυτικότερα, θέλουμε να παραχθούν μπλοκς για την πίστα μας με πλάτος(x) επί το ύψος(y). Επειδή θέλουμε να αλλάζει η πιθανότητα εμφάνισης του κάθε μπλοκ ανάλογα με το βάθος (ύψος) του θα βρίσκεται, δημιουργούμε τα μπλοκς γραμμή-γραμμή. Έχοντας έναν τυχαίο αριθμό στο εύρος [0.0-1.0) ανάλογα το βάθος που είμαστε αλλάζουμε την πιθανότητα εμφάνισης του μπλοκ που θέλουμε. Π.χ. αν χωρίσουμε νοητά όλο το ύψος της πίστας σε έξη ίσα μέρη στο πρώτο που θα είναι κοντά στην επιφάνεια έχει μεγάλη πιθανότητα εμφάνισης χώματος και άνθρακα, ενώ στο τελευταίο μέρος δεν υπάρχει πιθανότητα εμφάνισης άνθρακα διότι υπάρχουν πιο πολύτιμα ορυκτά όπως χρυσάφι και σίδηρος. Αλλά όσο πιο βαθιά είναι τόσο μεγαλύτερη πιθανότητα υπάρχει να εμφανιστεί βράχος ή δηλητηριώδη αέρια οπότε είναι και πιο επικίνδυνα.

3.6 Multiplayer

Για τη λειτουργία για πολλούς παίκτες (Multiplayer) χρησιμοποιήθηκε το Server-Client μοντέλο, στο οποίο μια συσκευή κάνει τον Host (Server) και οι άλλες κάνουν join (Client). Για αυτήν την λειτουργία χρησιμοποιήθηκε η βιβλιοθήκη KryoNet.

3.6.1 KryoNet Java Library

Η KryoNet είναι μια βιβλιοθήκη java (https://github.com/EsotericSoftware/kryonet) που παρέχει μια καθαρή και απλή διεπαφή προγραμματισμού εφαρμογών (API) για αποτελεσματική επικοινωνία δικτύου client/server με τα πρωτοκολλά TCP και UDP.
Η KryoNet βιβλιοθήκη είναι ιδανική για εφαρμογές client/server είναι πολύ αποδοτική και κατάλληλη για παιχνίδια.
Το KryoNet χρησιμοποιεί τη δικιά της βιβλιοθήκη σειριοποίησης (serialization), την Kryo (https://github.com/EsotericSoftware/kryo) για την αυτόματη και αποδοτική μεταφορά αντικειμένων στο δίκτυο.

3.6.2 Χρήση της KryoNet

Η χρήση της βιβλιοθήκης είναι πολύ απλή. Χρειάζεται να δημιουργηθεί ένα αντικείμενο Server και ένα Client. Έπειτα πρέπει και στα δυο αυτά αντικείμενα να γίνει εγγραφή (register) των κλάσεων των αντικειμένων που θα μεταφερθούν στο δίκτυο με την ίδια σειρά στον κώδικα. Το βήμα αυτό χρειάζεται γιατί αυτά τα αντικείμενα πρέπει σειριοποιηθούν πριν μεταφερθούν στο δίκτυο και λόγω της αποδοτικότητας της βιβλιοθήκης σειριοποίησης (Kryo) πρέπει να εγγραφτούν με την ίδια σειρά οι κλάσεις. Μετά ο server κάνει bind σε μια θύρα (port) TCP και σε μια UDP. Μετά ο client πρέπει να συνδεθεί με χρήση των ίδιων θυρών. Ο client μπορεί να συνδεθεί είτε γνωρίζοντας την IP του server είτε μπορεί να κάνει αναζήτηση για τον server, αν βρίσκονται στο ίδιο τοπικό δίκτυο (LAN – WLAN/WIFI), όπως γίνεται και στο παρόν παιχνίδι. Εφόσον συνδεθούν οι δύο συσκευές αρχίζουν και στέλνουν αντικείμενα των κλάσεων που προαναφέρθηκαν. Στο παρόν παιχνίδι με το που συνδεθεί ο client στέλνει στον server τον χαρακτήρα του, έπειτα του απαντά ο server με ότι πληροφορία χρειάζεται ο client για να δημιουργήσει τον ίδιο κόσμο με τον server και τον δικό του χαρακτήρα φυσικά. Συγκεκριμένα του στέλνει έναν αριθμό (seed) για τον αλγόριθμο δημιουργίας του χάρτη και το αν είναι να δημιουργήσει μισό ή ολόκληρο κόσμο, τις συντεταγμένες (x,y) και το όνομα του χαρακτήρα. Εφόσον γίνουν αυτά οι παίκτες είναι έτοιμοι να αρχίσουν το σκάψιμο. Κάθε φορά που μετακινείται ένας παίκτης στέλνει στον άλλο τις συντεταγμένες (x,y) του και το αν μετακινήθηκε προς τα δεξιά ώστε να αλλάξει το γραφικό του χαρακτήρα να κοιτάει την σωστή πλευρά. Ακόμα όταν κάποιος παίκτης σκάψει κάποιο τετραγωνάκι (χώμα, ορυκτό) στέλνει το ID στον άλλον παίκτη ώστε να καταστραφεί και σε αυτόν. Το ίδιο γίνεται και όταν τοποθετείται μια σκάλα ή κάτι άλλο, στέλνεται οι συντεταγμένες (x,y) και το είδος του αντικειμένου π.χ. σκάλα. Όταν αποσυνδεθεί κάποιος παίχτης στέλνει σήμα στον άλλον ότι αποσυνδέθηκε για να κλείσει το παιχνίδι.

3.7 Δημιουργία αρχείου για το Google Play

Για να ανέβει το βιντεοπαιχνίδι στο Google Play (Store) πρέπει να δημιουργηθεί ένα Signed APK (Android PacKage). Στην πραγματικότητα είναι το παραγόμενο πακέτο android ψηφιακά υπογεγραμμένο με πιστοποιητικό από τον προγραμματιστή. Το πιστοποιητικό δεν χρειάζεται να το υπογράψει μια αρχή πιστοποίησης. Η δημιουργία του Signed APK στο Android Studio βρίσκεται στο πάνω μενού Build-> Generate Signed APK. Όπου ακολουθώντας τις οδηγίες ο προγραμματιστής θα δημιουργήσει ένα πιστοποιητικό αν δεν έχει και θα υπογράψει με αυτό την εφαρμογή του. Για ποιο λόγο όμως χρειάζεται το πιστοποιητικό?

Το Android χρησιμοποιεί το πιστοποιητικό για να αναγνωρίσει τον συντάκτη (προγραμματιστή) της εφαρμογής και να δημιουργήσει σχέσεις εμπιστοσύνης μεταξύ των εφαρμογών. Δηλαδή, για να γίνει αναβάθμιση μιας εφαρμογής πρέπει να είναι υπογεγραμμένη με το ίδιο πιστοποιητικό που εγκαταστάθηκε. Ακόμα οι εφαρμογές που έχουν υπογραφτεί με το ίδιο πιστοποιητικό μπορούν αν θέλουν να εκτελούνται στην ίδια διαδικασία (process) ώστε να αντιμετωπίζονται ως ενιαία εφαρμογή. Με αυτό τον τρόπο ο προγραμματιστής μπορεί να αναπτύσσει την εφαρμογή του σε ενότητες και να τις ενημερώνει ανεξάρτητα.

4. Συμβουλές και Παρατηρήσεις

4.1 Εισαγωγή

Κλείνοντας αναφέρονται κάποιες συμβουλές και παρατηρήσεις σχετικά με προβλήματα που βρέθηκαν και αντιμετωπίστηκαν στην πορεία υλοποίησης του βιντεοπαιχνιδιού.

4.2 Συμβουλές και Παρατηρήσεις

- Αν θέλετε να δουλεύει σε Android το παιχνίδι θα χρειαστεί οπωσδήποτε το Android Studio οπότε ξεκινήστε απευθείας σε αυτό το περιβάλλον
- Το παρόν παιχνίδι δεν δημιουργήθηκε και για iOS διότι για να κάνει compile σε iOS χρειάζεται έναν υπολογιστή Mac και γενικά είναι πιο ακριβή η ανάπτυξη σε πλατφόρμα iOS
- Πάντα να παίζεται παρόμοια παιχνίδια με αυτό που δημιουργείτε για να παίρνεται ιδέες, κανένα παιχνίδι πλέον δεν είναι τελείως πρωτότυπο
- Μην παρακολουθείτε δύο ή περισσότερα μαθήματα για το ίδιο ακριβώς θέμα, διότι το πιο πιθανόν είναι να έχουν άλλη προσέγγιση για το πώς πρέπει να δομηθεί ο κώδικας και στο τέλος να γίνει ένα χάος
- Όταν έχει βραδιάσει και έχετε κουραστεί μην προσπαθείτε να λύσετε κάποιο πρόβλημα στον κώδικα, το επόμενο πρωί θα βρεθεί πολύ πιο γρήγορα η λύση
- Όταν πρόκειται για παιχνίδι με πολλούς να δημιουργείτε πάντα μια οθόνη π.χ. Lobby menu, πριν την εισαγωγή στο κυρίως παιχνίδι για ανταλλαγή ρυθμίσεων και πληροφοριών όπως μέγεθος πίστας.
- Ακόμα αν σκέφτεστε για peer-to-peer multiplayer στο παιχνίδι καλό είναι να ρίξετε μια ματιά εδώ https://www.factorio.com/blog/post/fff-147

5. Πως ξεκίνησα το Game Development

Όλοι παίζαμε παιχνίδια όταν είμασταν παιδιά απλώς κάποιοι δεν σταμάτησαν ποτέ, προφανώς είμαι ένας από αυτούς. Στο δεύτερο εξάμηνο της σχολής ξεκινήσαμε να μαθαίνουμε αντικειμενοστραφής προγραμματισμό σε γλώσσα Java και επειδή μου άρεσε το έψαξα πιο πολύ και βρήκα μια ωραία σελίδα την www.kilobolt.com, η οποία έχει μαθήματα για το πώς φτιάχνεται ένα βιντεοπαιχνίδι σε πλατφόρμα Android. Ξεκίναγε με τα βασικά της γλώσσας Java, μετά την λογική και δημιουργία ενός βιντεοπαιχνιδιού σε Java applet και τέλος την μετατροπή του σε Android application. Σε κάποιο από τα μαθήματα της σελίδας kilobolt.com χρησιμοποιούσε το libGDX framework και έτσι το ανακάλυψα.

Εφόσον μου άρεσαν πολύ τα βιντεοπαιχνίδια τι καλύτερο από το να δημιουργώ δικά μου παιχνίδια. Έτσι ξεκίνησα να ασχολούμαι με το Game Development. Στην εικόνα 5.1 φαίνεται το πρώτο μου παιχνίδι με την βοήθεια της σελίδας kilobolt.com.

Έπειτα άρχισα να παρακολουθώ την ανάπτυξη ενός νέου παιχνιδιού το factorio (www.factorio.com). Οι προγραμματιστές γράφουν κάθε παρασκευή ένα blog για το πώς προχωράει η ανάπτυξη του παιχνιδιού τους, το οποίο blog είναι πολύ ενδιαφέρον γιατί έχει και τεχνικά θέματα. Οπότε έμαθα κάποια πράγματα από εκεί.

Εικόνα 5.1: Βιντεοπαιχνίδι σε Java Applet

Βιβλιογραφία

Εισαγωγικά:

en.wikipedia.org/wiki/Tennis_for_Two

en.wikipedia.org/wiki/Mobile_game

el.wikipedia.org/wiki/Android

el.wikipedia.org/wiki/Ιστορία_εκδόσεων_του_Android

Βιβλιοθήκες που χρησιμοποιήθηκαν:

libgdx.badlogicgames.com

github.com/EsotericSoftware/kryonet

Χρήσιμα links:

freesound.org

pixabay.com

github.com/czyzby/gdx-skins

romannurik.github.io/AndroidAssetStudio/icons-launcher.html

romannurik.github.io/AndroidAssetStudio/index.html