

# **Ανοικτό Πανεπιστήμιο Κύπρου**

*Σχολή Θετικών και Εφαρμοσμένων Επιστημών*

## **Μεταπτυχιακή Διατριβή στα Πληροφοριακά και Επικοινωνιακά Συστήματα**



**Κατασκευή Παιχνιδιού σε Κινητή Πλατφόρμα και Ενσωμάτωση  
Μηχανισμών Τεχνητής Νοημοσύνης**

**Ευθύμιος Σταμάτης**

**Επιβλέπων Καθηγητής  
Δημήτριος Καλλές**

**Μάιος 2014**

**Ανοικτό Πανεπιστήμιο Κύπρου**  
**Σχολή Θετικών και Εφαρμοσμένων Επιστημών**  
**Κατασκευή Παιχνιδιού σε Κινητή Πλατφόρμα και Ενσωμάτωση**  
**Μηχανισμών Τεχνητής Νοημοσύνης**

**Ευθύμιος Σταμάτης**

**Επιβλέπων Καθηγητής**  
**Δημήτριος Καλλές**

Η παρούσα μεταπτυχιακή διατριβή υποβλήθηκε  
προς μερική εκπλήρωση των απαιτήσεων για απόκτηση

μεταπτυχιακού τίτλου σπουδών  
στα Πληροφοριακά Συστήματα

από τη Σχολή Θετικών και Εφαρμοσμένων Επιστημών  
του Ανοικτού Πανεπιστημίου Κύπρου

**Μάιος 2014**

# Περίληψη

Η παρούσα μεταπτυχιακή διατριβή έχει ως στόχο την μετατροπή μίας εφαρμογής (RLGame) που κάνει χρήση τεχνολογίας τεχνητής νοημοσύνης και εκτελείται σε περιβάλλον windows χωρίς την δυνατότητα παιχνιδιού ανθρώπου - υπολογιστή, σε μία εφαρμογή (mRLGame) που εκτελείται σε περιβάλλον Android, κάνει χρήση γραφικών και υποστηρίζει παιχνίδι μεταξύ υπολογιστή και ανθρώπου.

Η παρούσα διατριβή εστιάζει στις περιοχές της Τεχνητής Νοημοσύνης και της Τεχνολογίας Λογισμικού και ειδικότερα στην ανάπτυξη παιχνιδιών που χρησιμοποιούν σκακίερα σε περιβάλλον Android και τεχνολογία Ενισχυτικής Μάθησης.

Στα πλαίσια της παρούσας διατριβής πραγματοποιήθηκαν τα ακόλουθα:

1. η μελέτη προηγούμενων υλοποιήσεων ώστε να γίνει κατανοητό πως λειτουργεί η ενισχυτική μάθηση και γενικότερα η τεχνητή νοημοσύνη όπως και η εκμάθηση νέων τεχνολογιών όπως η java και η ανάπτυξη εφαρμογών σε περιβάλλον Android και ειδικότερα παιχνιδιών.
2. η σχεδίαση και υλοποίηση της εφαρμογής mRLGame με χρήση της βιβλιοθήκης ενισχυτικής μάθησης RL-Lib προϊόν προγενέστερης μεταπτυχιακής διπλωματικής εργασίας, σε περιβάλλον Android με παράλληλη προσθήκη του γραφικού περιβάλλοντος, της επικοινωνίας γραφικού περιβάλλοντος και του χρήστη και της δυνατότητας παιχνιδιού μεταξύ ανθρώπου και υπολογιστή.
3. η δοκιμή της εφαρμογής σε διάφορες κινητές πλατφόρμες πάνω σε πραγματικό υλικό και σε προσομοιωτές όπου ελέγχτηκε το παιχνίδι σχετικά με την συμβατότητα του σε διαφορετικού τύπου υλικό, όπως και σχετικά με την ορθότητα υπολογισμού των κινήσεων του υπολογιστή παίκτη.

# Summary

*The goal of this dissertation is to convert a game application (RLGame) that makes use of reinforcement learning, is designed for a standard desktop platform and does not support games between human player and computer player to an Android game application (mRLGame) that uses graphics and allows games between a human player and a computer player.*

*This dissertation draws on Artificial Intelligence and Software Engineering.*

*In the framework of this dissertation the following activities where completed:*

- 1. Research of previous implementations of the game so as to learn how reinforcement learning is functioning (more generally, an introduction to some artificial intelligence was carried out). New technologies had to be mastered, like Java programming and Android application development.*
- 2. The new mRLGame Android application was designed and created using a reinforcement learning library (RL-Lib), which was the product of an earlier dissertation. To the new application a GUI had to be added to support the functionality between a human and a computer player.*
- 3. The new application was tested in various Android mobile platforms, using hardware or virtual devices, so as to test it for compatibility issues and to demonstrate the effectiveness of the artificial intelligence mechanism.*

## Ευχαριστίες

*Θα ήθελα να ευχαριστήσω αρχικά την οικογένεια μου για την αμέριστη συμπαράσταση στο έργο της συγγραφής αυτής της εργασίας και φυσικά τον κ. Καλλέ Δημήτριο, τον επιβλέποντα αυτού του έργου, ο οποίος πίστεψε στις δυνατότητες μου, με καθοδήγησε σωστά και έδειξε απεριόριστη κατανόηση στις δυσκολίες που αντιμετώπισα.*

# Περιεχόμενα

Περίληψη.....	i
Summary.....	ii
Ευχαριστίες .....	iii
Περιεχόμενα .....	iv
Κεφάλαιο 1 .....	1
Εισαγωγή.....	1
1.1 Σύντομη Εισαγωγή.....	1
1.2 Αντικείμενο της Διπλωματικής.....	2
1.2.1 Σύντομη Περιγραφή των Σταδίων Σχεδίασης του Συστήματος .....	3
1.2.2 Σύντομη Περιγραφή των Δυσκολιών στην Σχεδίαση και Υλοποίηση .....	3
1.2.3 Έλεγχος Καλής Λειτουργίας.....	4
1.3 Διάθρωση Διπλωματικής Διατριβής.....	4
Κεφάλαιο 2 .....	6
Κινητές Πλατφόρμες Υπολογιστικών Συστημάτων .....	6
2.1 Κινητές Πλατφόρμες.....	6
2.1.1 Προσωπικός Ψηφιακός Βοηθός (PDA).....	7
2.1.2 Έξυπνα Κινητά (Smartphones) .....	8
2.1.3 Ταμπλέτες (Tablets) .....	8
2.2 Λειτουργικά Συστήματα και Τεχνολογίες Κινητών Πλατφόρμων .....	10
2.2.1 Παλιότερα λειτουργικά συστήματα .....	10
2.2.2 Λειτουργικά συστήματα τελευταίας γενιάς .....	12
2.3 Το λειτουργικό σύστημα Android.....	13
2.3.1 Γενικά.....	13
2.3.2 Open Handset Alliance .....	15
2.3.3 Εφαρμογές – Play Store .....	15
2.3.4 Εκδόσεις του Android .....	16
2.3.5 Η Αρχιτεκτονική του Android.....	16
2.3.6 Επιλογή του Android.....	17
Κεφάλαιο 3 .....	19
Παιχνίδια Στρατηγικής και Πλατφόρμες Ανάπτυξης Παιχνιδιών .....	19

3.1	Παιχνίδια Στρατηγικής και Τεχνητή Νοημοσύνη .....	19
3.2	Πλατφόρμες Ανάπτυξης Παιχνιδιών Τεχνητής Νοημοσύνης .....	21
3.2.1	Περιγραφή του RL-Glue .....	22
3.2.2	Περιγραφή του ORTS (Open Real-Time Strategy) .....	22
3.2.3	Περιγραφή του Game Maker .....	23
3.2.4	Περιγραφή του General Game Playing (GGP).....	23
<b>Κεφάλαιο 4 .....</b>		<b>24</b>
<b>Το Παιχνίδι Στρατηγικής RLGame .....</b>		<b>24</b>
4.1	Περιγραφή του Παιχνιδιού.....	24
4.2	Κανόνες .....	25
4.3	Τεχνητή Νοημοσύνη.....	27
4.3.1	Ιδιότητες Παιχνιδιού .....	28
4.3.2	Εφαρμογή Ενισχυτικής Μάθησης.....	29
4.3.3	Εφαρμογή Νευρωνικών Δικτύων .....	30
4.4	Υπάρχουσες Υλοποιήσεις.....	31
4.4.1	Υπολογιστής εναντίον Υπολογιστή .....	31
4.4.2	Minimax εναντίον Υπολογιστή .....	32
4.4.3	Άνθρωπος εναντίον Υπολογιστή.....	32
<b>Κεφάλαιο 5 .....</b>		<b>33</b>
<b>Πρωτότυπη Βιβλιοθήκη για Παιχνίδια Ενισχυτικής Μάθησης - RLLib .....</b>		<b>33</b>
5.1	Γενικά.....	33
5.2	Αρχιτεκτονική.....	34
5.2.1	Πακέτο Βοηθητικών Κλάσεων.....	34
5.2.2	Πακέτο Μάθησης.....	36
<b>Κεφάλαιο 6 .....</b>		<b>38</b>
<b>Σχεδίαση και Υλοποίηση του mRLGame .....</b>		<b>38</b>
6.1	Προετοιμασία .....	38
6.1.1	Τεχνολογίες που χρειάστηκαν .....	39
6.1.2	Δοκιμές με υπάρχοντα κώδικα και η επιλογή του RLLib.....	39
6.2	Σχεδίαση και Υλοποίηση .....	40
6.2.1	Μετατροπή του αρχικού RLLib σε ένα Android Project .....	40
6.2.2	Σχεδίαση του GUI & των γραφικών.....	41
6.2.3	Σύνδεση του RLLib με τα Γραφικά.....	41

6.3	Αρχιτεκτονική .....	42
6.3.1	Παρουσίαση των Δραστηριοτήτων (Activities) .....	42
6.3.2	Παρουσίαση της δομής του java κώδικα .....	42
6.3.3	Αλληλεπίδραση Παιχνιδιού και RLLib .....	43
6.3.4	Περιγραφή των κλάσεων.....	44
6.3.5	Διαγράμματα κλάσεων .....	47
6.3.6	Τα Υπόλοιπα Αντικείμενα της Υλοποίησης .....	49
6.4	Βασικά Σημεία Σχεδίασης και Δυσκολίες Υλοποίησης.....	51
6.4.1	Δημιουργία της κλάσης Human Player .....	51
6.4.2	Η Δημιουργία της κλάσης GameStateUI.....	52
6.4.3	Καταστάσεις του γραφικού περιβάλλοντος .....	53
6.4.4	Αντιμετώπιση θεμάτων ανάλυσης οθόνης .....	54
6.4.5	Θέση Αποθήκευσης Αρχείων.....	55
<b>Κεφάλαιο 7 .....</b>		<b>56</b>
<b>Έλεγχος καλής Λειτουργίας και Προώθηση σε Τελικούς Χρήστες .....</b>		<b>56</b>
7.1	Δοκιμές σε φυσικές συσκευές .....	56
7.2	Δοκιμές σε εικονικές συσκευές.....	58
7.2.1	Εικονικές συσκευές του AVDM.....	58
7.2.2	Εικονική συσκευή με Android X86 σε Virtual Box .....	59
7.3	Δοκιμή καλής λειτουργίας τεχνητής νοημοσύνης.....	59
7.4	Προώθηση σε Τελικούς Χρήστες.....	62
<b>Κεφάλαιο 8 .....</b>		<b>63</b>
<b>Συμπεράσματα.....</b>		<b>63</b>
8.1	Απολογισμός .....	63
<b>Βιβλιογραφία .....</b>		<b>65</b>
<b>Παράρτημα Α.....</b>		<b>1</b>
<b>Γλωσσάρι.....</b>		<b>1</b>
A.1	Γλωσσάρι Ορολογιών .....	1
<b>Παράρτημα Β.....</b>		<b>1</b>
<b>Εργαλεία που χρησιμοποιήθηκαν και η δομή ενός Android Project.....</b>		<b>1</b>
B.1	Eclipse Android Development Tools .....	1
B2.	Star-UML.....	3
B3.	Η δομή ενός Android project.....	3



<b>Παράρτημα Γ .....</b>	<b>1</b>
<b>Εγχειρίδιο εγκατάστασης &amp; χρήσης.....</b>	<b>1</b>
Γ.1      Εγκατάσταση του εφαρμογής.....	1
Γ.2      Εκτέλεση της εφαρμογής .....	2

# Κεφάλαιο 1

## Εισαγωγή

Η παρούσα διατριβή εστιάζει στις περιοχές της Τεχνητής Νοημοσύνης και της Τεχνολογίας Λογισμικού σε κινητές πλατφόρμες. Ο απώτερος σκοπός της είναι η δημιουργία μίας εφαρμογής σε περιβάλλον Android που θα υλοποιεί το πρωτότυπο παιχνίδι στρατηγικής RLGame με την χρήση της βιβλιοθήκης ενισχυτικής μάθησης RL-Lib, η οποία αναπτύχθηκε στα πλαίσια παλιότερης μεταπτυχιακής διατριβής.

### 1.1 Σύντομη Εισαγωγή

Η μελέτη των παιχνιδιών στρατηγικής είναι ένα πεδίο που έχει τραβήξει εδώ και πολλά χρόνια την προσοχή των ερευνητών που ασχολούνται με εφαρμογές Τεχνητής Νοημοσύνης. Οι ερευνητές καταβάλουν προσπάθειες να δημιουργήσουν έξυπνα προγράμματα που θα είναι σε θέση να παίξουν το παιχνίδι ανταγωνιστικά και να είναι αποτελεσματικά ακόμη και εναντίον ανθρώπων παικτών.

Το RLGame είναι ένα παιχνίδι στρατηγικής που παίζεται ανάμεσα σε δύο παίκτες χρησιμοποιώντας πιόνια που κινούνται πάνω σε μία επιφάνεια τύπου «σκακιέρας». Το υπολογιστικό σύστημα εκτέλεσης πειραμάτων του παιχνιδιού έχει ήδη χρησιμοποιηθεί σε πολλές ερευνητικές προσπάθειες σχετικά με την εφαρμογή τεχνικών μάθησης στο πεδίο των παιχνιδιών στρατηγικής.

Ο βασικός στόχος της ερευνητικής ομάδας που υλοποίησε το σύστημα ήταν να δημιουργήσει υπολογιστικούς παίκτες που εκμεταλλεύονται τεχνικές Ενισχυτικής Μάθησης (*Reinforcement Learning*) και υλοποιήσεις Νευρωνικών Δικτύων (*Neural Networks*) στην προσπάθεια τους να παίξουν το παιχνίδι και να μάθουν/αναπτύξουν μια βέλτιστη στρατηγική.

Το σύστημα είχε αναπτυχθεί σε γλώσσα προγραμματισμού Java, ενώ το περιβάλλον εκτέλεσης πειραμάτων επέτρεπε τόσο τη μάθηση μέσω εκτέλεσης πειραμάτων ανάμεσα σε υπολογιστικούς παίκτες που χρησιμοποιούν κοινές τεχνικές ενισχυτικής μάθησης (υπολογιστής εναντίον υπολογιστή, *Self-play experiments*), ενώ δίνονταν ακόμη η δυνατότητα πειραμάτων παιχνιδιών υπολογιστικού παίκτη που χρησιμοποιεί τεχνικές ενισχυτικής μάθησης με υλοποίηση του αλγόριθμου min-max και η δυνατότητα πειραμάτων παιχνιδιών ανθρώπου με υπολογιστικό παίκτη που χρησιμοποιεί τεχνικές ενισχυτικής μάθησης.

Πριν την παρούσα διπλωματική εργασία δεν υπήρχε σχετική εμπειρία του γράφοντα με τα διάφορα εργαλεία και δεξιότητες που χρειάστηκαν για να ολοκληρωθεί όλο το έργο. Η εργασία ξεκίνησε με εκμάθηση της java, με παράλληλη εκμάθηση των τεχνολογιών τεχνικής νοημοσύνης και ειδικά με τις εργασίες που έχουν γίνει, papers και διπλωματικές, σχετικά με το RL-Game. Σε δεύτερο στάδιο χρειάστηκε να γίνει εκμάθηση της πλατφόρμας ανάπτυξης (ADT) του Android και πιο ειδικά στη σχεδίαση παιχνιδιών σε αυτό το περιβάλλον. Το τρίτο στάδιο περιλαμβάνει το συνδυασμό των παραπάνω γνώσεων με το αποτέλεσμα μίας μεταπτυχιακής διπλωματικής εργασίας [01] που δημιούργησε μία βιβλιοθήκη για παιχνίδια ενισχυτικής μάθησης με την ονομασία RL-Lib για την δημιουργία ενός παιχνιδιού σε Android που θα παίζει το παιχνίδι RLGame μεταξύ ανθρώπου και υπολογιστή το οποίο ονομάστηκε Mobile RLGame και σε συντομία mRLGame.

## 1.2 Αντικείμενο της Διπλωματικής

Στα πλαίσια της παρούσας διατριβής έγινε επανασχεδίαση της υλοποίησης του RLGame που είχε δημιουργηθεί σε προηγούμενη μεταπτυχιακή διπλωματική εργασία [01], ώστε να είναι δυνατή η εκτέλεση του σε περιβάλλον Android. Έγινε σχεδίαση του γραφικού περιβάλλοντος αλληλεπίδρασης με τον χρήστη (UI) και ο κώδικας της προηγούμενης υλοποίησης εμπλουτίστηκε με τις απαραίτητες κλάσεις, ώστε να υποστηρίζει παιχνίδι μεταξύ υπολογιστή και ανθρώπου καθώς και της διασύνδεσης με το περιβάλλον αλληλεπίδρασης του χρήστη, αφού η αρχική υλοποίηση δεν το υποστήριζε.

### **1.2.1 Σύντομη Περιγραφή των Σταδίων Σχεδίασης του Συστήματος**

Προτού καταστεί δυνατή η σχεδίαση του νέου συστήματος χρειάστηκε να γίνει εκμάθηση των τεχνολογιών που χρησιμοποιήθηκαν καθώς και η μελέτη των υπαρχουσών υλοποιήσεων ώστε να γίνει κατανοητή η δομή του RLGame και ειδικότερα της υπάρχουσας υλοποίησης [01] της οποίας η δομή χρησιμοποιήθηκε ως βάση για την ανάπτυξη της νέας εφαρμογής.

Η σχεδίαση του συστήματος ακολούθησε δύο παράλληλους δρόμους: ο πρώτος αφορούσε την ενσωμάτωση του υπάρχοντος κώδικα που ήταν γραμμένος σε java με την μορφή ενός Eclipse Project σε ένα Android Project και ο δεύτερος αφορούσε την σχεδίαση γραφικών σε ένα Android περιβάλλον και πιο ειδικά σε ένα Android παιχνίδι ο οποίος κατέληξε τελικά στη δημιουργία μίας Android εφαρμογής που εκτελούσε το παιχνίδι, χωρίς την ύπαρξη όμως του κώδικα που υλοποιούσε την λογική του παιχνιδιού, υλοποιώντας ουσιαστικά μόνο το γραφικό περιβάλλον με τον χρήστη να έχει την δυνατότητα να μετακινεί τα πιόνια επάνω στην σκακιέρα.

Οι δύο κατευθύνσεις ανάπτυξης στη συνέχεια ενσωματώθηκαν σε μία εφαρμογή που πλέον είχε το γραφικό μέρος αλλά και το τμήμα λογικής παιχνιδιού χωρίς όμως τα δύο τμήματα να επικοινωνούν μεταξύ τους. Στη συνέχεια έγινε η δημιουργία των κλάσεων που συνδέουν τα δύο τμήματα και της κλάσης παίκτη ανθρώπου που υλοποιεί τις κινήσεις του παίκτη στην σκακιέρα σε κίνηση στο υποσύστημα λογικής του παιχνιδιού.

### **1.2.2 Σύντομη Περιγραφή των Δυσκολιών στην Σχεδίαση και Υλοποίηση**

Τη μεγαλύτερη απαίτηση σε χρόνο είχε, όπως αποδείχθηκε, η προετοιμασία των προαπαιτούμενων, δηλαδή η εκμάθηση των τεχνολογιών και εργαλείων που χρησιμοποιήθηκαν καθώς και η μελέτη των προηγούμενων υλοποιήσεων όπως και αυτής πάνω στην οποία βασίστηκε η υλοποίηση αυτής της εφαρμογής.

Ένα θέμα το οποίο δεν είχε προβλεφθεί αρχικά αλλά τελικά χρειάστηκε μεγάλο μέρος του χρόνου είναι η σχεδίαση γραφικών στο Android και ειδικά γραφικών για παιχνίδια τα οποία θέλουν επανασχεδίαση της οθόνης σε τακτά χρονικά διαστήματα ή όποτε υπάρχει ανάγκη. Το ότι ένα παιχνίδι σκακιέρας δεν έχει την ανάγκη μετακίνησης πολλών αντικειμένων μέσα στην οθόνη εκτός φυσικά από τα πιόνια, οδήγησε στην επιλογή ανανέωσης της οθόνης όποτε υπήρχε η ανάγκη γι' αυτό. Η επιλογή αυτής της λύσης δημιούργησε τον προβληματισμό του προσδιορισμού της στιγμής επανασχεδίασης της οθόνης.

Η διασύνδεση των γραφικών του παιχνιδιού, με την λογική του παιχνιδιού ήταν το δυσκολότερο από τα στάδια υλοποίησης και το οποίο χρειάστηκε τη μεγαλύτερη προσπάθεια όπως επίσης και την δημιουργία του περισσότερου κώδικα. Η υλοποίηση επάνω στην οποία βασίστηκε η νέα εφαρμογή δεν παρουσίαζε καθόλου γραφικά και επίσης δεν υπήρχε πρόβλεψη για παιχνίδι μεταξύ ανθρώπου και υπολογιστή. Χρειάστηκε να δημιουργηθούν νέες κλάσεις που θα επιτρέπουν την διασύνδεση γραφικών και λογικής παιχνιδιού όπως επίσης και νέα κλάση που υλοποιεί τον άνθρωπο – παίκτη.

### **1.2.3 Έλεγχος Καλής Λειτουργίας**

Ο έλεγχος καλής λειτουργίας της νέας εφαρμογής περιλαμβάνει δύο σημεία ελέγχου. Ο πρώτος αφορά τον έλεγχο της σωστής εκτέλεσης της εφαρμογής σε διαφορετικά μοντέλα κινητών συσκευών όπως έξυπνα τηλέφωνα (smartphones) και υπολογιστές ταμπλέτες (tablets) και σε εικονικές πλατφόρμες όπως η πλατφόρμα Android-X86 [32] και ο προσομοιωτής που περιλαμβάνεται μαζί με τα εργαλεία ανάπτυξης λογισμικού του Android. Ο δεύτερος έλεγχος αφορά το σωστό υπολογισμό των κινήσεων από το μηχανισμό του παιχνιδιού για διαφορετικούς τύπους υπολογιστή παίκτη και στρατηγικές παιχνιδιού από τον άνθρωπο. Για αυτό τον έλεγχο παίχτηκαν διάφορα σετ παιχνιδιών ακολουθώντας για κάθε σετ διαφορετική στρατηγική προσέγγισης προς την βάση του αντιπάλου καταγράφοντας κάθε φορά την «αντίδραση» του παίκτη υπολογιστή. Στη συνέχεια αυτές οι κινήσεις αξιολογήθηκαν ως προς την δυνατότητα επανάληψης της «αντίδρασης» του παίκτη υπολογιστή στην ίδια στρατηγική από μεριάς του ανθρώπου - παίκτη.

## **1.3 Διάθρωση Διπλωματικής Διατριβής**

Η παρούσα διπλωματική διατριβή αποτελείται από οκτώ κεφάλαια. Το πρώτο κεφάλαιο αποτελεί την εισαγωγή της διπλωματικής διατριβής και σκοπό έχει να ενημερώσει για το τι πραγματεύεται η διατριβή, όπως και να κατατοπίσει σχετικά με την διάρθρωση της.

Στο δεύτερο κεφάλαιο γίνεται μία γενική αναφορά στις κινητές πλατφόρμες και μία πιο ειδική στο Android πάνω στο οποίο εκτελείται η εφαρμογή mRLGame. Το τρίτο κεφάλαιο κάνει μία γενική αναφορά στο παιχνίδι στρατηγικής και μία πιο ειδική στις πλατφόρμες ανάπτυξης παιχνιδιών που χρησιμοποιούν τεχνητή νοημοσύνη καθώς μία τέτοια πλατφόρμα με την ευρύτερη έννοια μπορούμε να θεωρήσουμε ότι είναι και το RL-Lib. Στο τέταρτο κεφάλαιο παρουσιάζεται το παιχνίδι RL-Game και οι παλιότερες υλοποιήσεις του, ενώ στο πέμπτο κεφάλαιο παρουσιάζεται η βιβλιοθήκη ενισχυτικής μάθησης RL-Lib.

Το έκτο κεφάλαιο παρουσιάζει την ανάπτυξη της εφαρμογής mRLGame και περιέχει τις τεχνολογίες που χρησιμοποιήθηκαν, τις γνώσεις που απαιτήθηκαν, τα στάδια σχεδίασης και υλοποίησης, την αρχιτεκτονική του κώδικα, τα βασικά σημεία της σχεδίασης και τις δυσκολίες που αντιμετωπίστηκαν.

Το έβδομο κεφάλαιο περιγράφει πώς έγινε ο έλεγχος καλής λειτουργίας της εφαρμογής με χρήση διαφόρων πραγματικών ή εικονικών συσκευών καθώς και της λειτουργίας του μηχανισμού τεχνητής νοημοσύνης. Επίσης περιγράφεται πώς μπορεί να εγκατασταθεί η εφαρμογή σε μία Android συσκευή και η ιστοσελίδα που δημιουργήθηκε για αυτή τη λειτουργία. Τέλος στο όγδοο κεφάλαιο γίνεται μία ανασκόπηση της εργασίας και πώς αυτή μπορεί να βοηθήσει μελλοντικές εργασίες με ίδια ή παραπλήσια κατεύθυνση.

Η διπλωματική εργασία περιλαμβάνει 3 παραρτήματα: το Α παράρτημα είναι ένα γλωσσάρι των τεχνικών όρων που χρησιμοποιούνται, το Β παράρτημα περιγράφει τα εργαλεία που χρησιμοποιήθηκαν και την δομή ενός Android Project και το Γ παράρτημα περιγράφει την διαδικασία εγκατάστασης της εφαρμογής, τις οδηγίες χρήσεως και πώς μπορεί να γίνει λήψη από την ειδικά κατασκευασμένη ιστοσελίδα και να εγκατασταθεί.

# Κεφάλαιο 2

## Κινητές Πλατφόρμες

### Υπολογιστικών Συστημάτων

Στις μέρες μας η χρήση κινητών υπολογιστικών συστημάτων, όπως έξυπνων τηλεφώνων (smartphones) & ταμπλετών (tablets), είναι ολοένα αυξανόμενη σε σχέση με την χρήση των κλασικών υπολογιστών. Αυτή η τάση δείχνει και το μέλλον που έρχεται και επιβάλλει την χρήση μικρών υπολογιστών με μεγάλη υπολογιστική ισχύ. Σε αυτό το κεφάλαιο παρουσιάζονται οι διάφορες κινητές πλατφόρμες και τα λειτουργικά συστήματα που εκτελούν. Επίσης παρουσιάζεται το λειτουργικό σύστημα Android και γιατί επιλέχθηκε αυτό για την ανάπτυξη της εφαρμογής mRLGame.

#### 2.1 Κινητές Πλατφόρμες

Μια κινητή πλατφόρμα (επίσης γνωστή ως κινητή συσκευή, συσκευή χειρός, υπολογιστής τσέπης ή απλά κινητό) είναι ένας υπολογιστής μικρού μεγέθους που αποτελείται από μία οθόνη αφής ή και ένα

πληκτρολόγιο. Η εισαγωγή δεδομένων σε αυτές τις συσκευές γίνεται κυρίως από την οθόνη αφής αλλά και από τους διάφορους αισθητήρες που έχουν όπως GPS, επιταχυνσιόμετρο κ.α.

### 2.1.1 Προσωπικός Ψηφιακός Βοηθός (PDA)

Ένας προσωπικός ψηφιακός βοηθός [02] (PDA), επίσης γνωστός ως υπολογιστής παλάμης (Palmtop Computer) [Εικόνα 2.1], είναι μια κινητή πλατφόρμα που λειτουργεί σαν προσωπικός διαχειριστής πληροφοριών. Τα σημερινά PDA έχουν την δυνατότητα να συνδέονται στο διαδίκτυο και μέσω της οθόνης αφής τους μπορεί ο χειριστής τους να ελέγχει πρόγραμμα περιήγησης στο παγκόσμιο ιστό. Η σύνδεση τους στο δίκτυο επιτυγχάνεται μέσω ασύρματων δικτύων (Wi-Fi).



Εικόνα 2.1 Προσωπικός Ψηφιακός Βοηθός (PDA) [02].

Ο όρος PDA χρησιμοποιήθηκε για πρώτη φορά τον Ιανουάριο του 1992 από τον πρόεδρο της εταιρίας Apple, John Sculley σε μια έκθεση ηλεκτρονικών συσκευών, την CES (Consumer Electronics Show) στο Las Vegas, και αναφερόταν στη συσκευή Apple Newton. Το 1992 η IBM παρουσίασε το πρώτο PDA που είχε την δυνατότητα να κάνει τηλεφωνικές κλήσεις, το IBM Simon το οποίο μπορεί να θεωρηθεί ως το πρώτο έξυπνο τηλέφωνο. Το 1996 η Nokia δημιούργησε το πρώτο κινητό τηλέφωνο με όλες τις λειτουργίες ενός PDA, το 9000 Communicator, το οποίο στη συνέχεια έγινε το πρώτο σε πωλήσεις παγκοσμίως. Με την είσοδο στην αγορά αυτής της συσκευής δημιουργήθηκε ουσιαστικά ο όρος PDA-τηλέφωνο που στην εποχή μας το ονομάζουμε smartphone. Σήμερα όλα τα PDA είναι ουσιαστικά και smartphones. Τα PDA χωρίς λειτουργίες τηλεφώνου περιορίζονται πια σε πολύ χαμηλές πωλήσεις και δημιουργούνται για να καλύψουν συγκεκριμένες ανάγκες, κυρίως στο τομέα της βιομηχανίας ή της αποθήκευσης (logistics).



### 2.1.2 Έξυπνα Κινητά (Smartphones)

Έξυπνο κινητό [03] (Smartphone) [Εικόνα 2.2], ονομάζεται ένα κινητό τηλέφωνο που προσφέρει αρκετά πιο ανεπτυγμένες πληροφοριακές δυνατότητες αλλά και συνδεσιμότητα από ότι ένα κοινό τηλέφωνο. Τα πρώτα έξυπνα τηλέφωνα συνδυάζαν τα χαρακτηριστικά ενός ψηφιακού βοηθού, με συσκευή αναπαραγωγής πολυμέσων, ψηφιακής κάμερας και συσκευής GPS. Τα τελευταίας γενιάς έξυπνα τηλέφωνα συνδυάζουν όλα τα παραπάνω και επιπρόσθετα έχουν τα χαρακτηριστικά ενός υπολογιστή με οθόνη αφής, πλοήγηση στο παγκόσμιο ιστό, ασύρματο δίκτυο (Wi-Fi) και υποστηρίζουν την εγκατάσταση εφαρμογών τρίτων κατασκευαστών. Τα smartphones χρησιμοποιούν ολοκληρωμένα λειτουργικά συστήματα τα οποία παρέχουν στους προγραμματιστές μια πλατφόρμα πάνω στην οποία μπορούν να αναπτύξουν τις εφαρμογές τους.



Εικόνα 2.2 Έξυπνο κινητό τηλέφωνο (Smartphone) [03].

Στις αρχές του 2013, για πρώτη φορά, οι πωλήσεις έξυπνων κινητών ξεπέρασαν παγκοσμίως εκείνες των κινητών τηλεφώνων παλαιότερου στυλ. Σύμφωνα με τελευταίες έρευνες (2014), το 90% των πωλήσεων παγκοσμίως αφορά έξυπνα τηλέφωνα [03].

### 2.1.3 Ταμπλέτες (Tablets)

Μια ταμπλέτα [04] (Tablet Computer) [Εικόνα 2.3], είναι ένας ηλεκτρονικός υπολογιστής μεγέθους συνήθως από 7 έως 11 ίντσες ο οποίος έχει ως βασικά του χαρακτηριστικά αυτά ενός κανονικού Η/Υ και συνήθως περιλαμβάνει ένα πλήρες λειτουργικό σύστημα. Μία ταμπλέτα έχει μια οθόνη αφής ως κύρια μέθοδο εισόδου δεδομένων, ενώ περιλαμβάνει ένα σετ από αισθητήρες, όπως επιταχυνσιόμετρο

και GPS, ενώ διαθέτει επίσης κάμερα και μικρόφωνο. Σε γενικές γραμμές θα μπορούσε κάποιος να πει ότι μία ταμπλέτα είναι ένα πιο μεγάλο σε μέγεθος έξυπνο τηλέφωνο, χωρίς όμως τις τυπικές λειτουργίες ενός τηλεφώνου, δηλαδή την πραγματοποίηση κλήσεων και την αποστολή και λήψη μηνυμάτων κειμένου (SMS).



Εικόνα 2.3 Υπολογιστής Ταμπλέτα (Tablet Computer) [04].

Ο όρος Tablet PC έγινε γνωστός από μια παρουσίαση της Microsoft το 2001 όπου και χρησιμοποιήθηκε για πρώτη φορά. Αυτός ο όρος όμως πια χρησιμοποιείται από μια ευρεία γκάμα μοντέλων και προϊόντων προσωπικών ηλεκτρονικών υπολογιστών αυτού του μεγέθους ακόμα και αν δεν χρησιμοποιούν το λειτουργικό σύστημα της Microsoft.

Οι ταμπλέτες χρησιμοποιούν εικονικά πληκτρολόγια και αναγνώριση γραφής έτσι ώστε να μπορούμε να εισάγουμε κείμενο, μέσω της οθόνης αφής αλλά οι περισσότερες μπορούν να δεχτούν και εξωτερικά πληκτρολόγιο για εισαγωγή κειμένου. Όλες οι ταμπλέτες έχουν δυνατότητα ασύρματης σύνδεσης στο διαδίκτυο μέσω Wi-Fi ή και μέσω κινητής τηλεφωνίας 3G – 4G. Το λογισμικό τους περιλαμβάνει εφαρμογές γραφείου, προγράμματα περιήγησης στο διαδίκτυο, παιχνίδια, αλλά από την στιγμή που τρέχουν ολοκληρωμένα λειτουργικά συστήματα, μπορούν ουσιαστικά να χρησιμοποιήσουν οποιοδήποτε πρόγραμμα υποστηρίζει το λειτουργικό τους.

Στις μέρες μας οι ταμπλέτες τείνουν να αντικαταστήσουν την χρήση υπολογιστών για την απλή πλοήγηση στο web, για την αναζήτηση πληροφοριών ή την χρήση τόπων κοινωνικής δικτύωσης. Ακόμη και επιχειρήσεις έχουν αρχίσει να κάνουν χρήση ταμπλετών στις συνεδριάσεις, τις εκδηλώσεις και τις εκθέσεις τους. Έρευνα του 2013 δείχνει ότι το 29% των καταναλωτών στις ΗΠΑ που διαθέτουν

κινητό έχουν και ταμπλέτα, σημειώνοντας πολύ θεαματική αύξηση σε σχέση με το 2011 που ήταν μόλις 5%.

## **2.2 Λειτουργικά Συστήματα και Τεχνολογίες Κινητών Πλατφόρμων**

Ένα λειτουργικό σύστημα για κινητές πλατφόρμες (γνωστό και ως Mobile OS) [05] είναι το λειτουργικό σύστημα που ελέγχει μια κινητή πλατφόρμα. Έχει τις ίδιες χαρακτηριστικές αρχές με ένα λειτουργικό σύστημα όπως τα Windows, το Mac OS, το Linux και άλλα, τα οποία ελέγχουν τους κλασικούς ηλεκτρονικούς υπολογιστές. Όμως αν και έχουν πολλά κοινά με τα προαναφερθέντα λειτουργικά, είναι κατά μια έννοια πιο ελαφρά, καθώς είναι φτιαγμένα ώστε να μπορούν να λειτουργούν με λιγότερους υπολογιστικούς πόρους, σχετίζονται περισσότερο με ασύρματες επικοινωνίες, με αναπαραγωγή αρχείων πολυμέσων και διαφορετικούς τρόπους εισαγωγής εντολών.

### **2.2.1 Παλιότερα λειτουργικά συστήματα**

Πολλά λειτουργικά συστήματα για κινητές πλατφόρμες έχουν παρουσιαστεί. Κάποια από αυτά έχουν εξαφανιστεί ενώ κάποια άλλα έχουν εξελιχτεί με νέα χαρακτηριστικά και δυνατότητες.

Ίσως το πρώτο λειτουργικό σύστημα για κινητή πλατφόρμα, με όλα τα χαρακτηριστικά που ένα τέτοιο πρέπει να έχει, είναι το Palm OS που έκανε τα πρώτα βήματα του στην αγορά το 1996. Αρχικά αναπτύχθηκε από την Palm και χρησιμοποιήθηκε σε PDA. Το Palm OS [06] [Εικόνα 2.4] σχεδιάστηκε με βάση την ευκολία χρήσης της οθόνης αφής και βασίστηκε στην αλληλεπίδραση του χρήστη με αυτήν μέσω ενός γραφικού περιβάλλοντος. Προσέφερε μια σουίτα με βασικές εφαρμογές με σκοπό την διαχείριση προσωπικών πληροφοριών. Αργότερα με την πάροδο των χρόνων, νέες εκδόσεις του λειτουργικού συστήματος υποστήριξαν και έξυπνα τηλέφωνα.

Τα Windows Mobile [07] [Εικόνα 2.4] είναι ένα λειτουργικό σύστημα για κινητές πλατφόρμες που αναπτύχθηκε από την Microsoft και χρησιμοποιείται σε έξυπνα τηλέφωνα και προσωπικούς ψηφιακούς βοηθούς. Η Microsoft το 2010 ανακοίνωσε ένα νέο λειτουργικό σύστημα για smartphones, τα Windows Phone 7, και παράλληλα σταμάτησε την υποστήριξη των κινητών τηλεφώνων που χρησιμοποιούν Windows Mobile.

Το Symbian OS [08] [Εικόνα 2.4] είναι λειτουργικό σύστημα για φορητές συσκευές και αποτελεί εξέλιξη του λειτουργικού συστήματος EPOC από την Psion. Το Symbian OS δημιουργήθηκε με τη γλώσσα προγραμματισμού C++ από τη Symbian Ltd. Πριν το 2009 το Symbian OS υποστήριζε διαφορετικά περιβάλλοντα χρήστη, όμως με την δημιουργία του Symbian Platform, το ίδιο έτος, τα 3 βασικά περιβάλλοντα χρήστη ενώθηκαν σε ένα, το οποίο εξαγοράστηκε από την Nokia και στην συνέχεια μετατράπηκε σε λογισμικό ανοικτού κώδικα. Μετά την εξαγορά του τμήματος κινητών συσκευών της Nokia από την Microsoft η ανάπτυξη του λειτουργικού σταμάτησε, καθώς όλες οι νέες συσκευές πλέον της Nokia θα έχουν λειτουργικό Windows Phone.



Εικόνα 2.4 Τα λειτουργικά συστήματα Palm OS [06], Windows Mobile [07] & Symbian OS [08].

### 2.2.2 Λειτουργικά συστήματα τελευταίας γενιάς

Τα λειτουργικά συστήματα για κινητές πλατφόρμες τα τελευταία χρόνια έχουν πραγματοποιήσει αλματώδη πρόοδο σε πολλούς τομείς ιδίως στην ευκολία χρήσης και την υποστήριξη πολλών διαφορετικών εφαρμογών. Η αυξανόμενη σημαντικότητα αυτών των συστημάτων στην παγκόσμια αγορά έχει καταστήσει ανταγωνιστικότητα την συγκεκριμένη κατηγορία λογισμικού και έχει κινηήσει το ενδιαφέρον ανάμεσα στους γίγαντες της ανάπτυξης λογισμικού, όπως η Google, η Microsoft και η Apple, με σκοπό την κατάκτηση του μεγαλύτερου μεριδίου στην αγορά.

Με την είσοδο του iPhone στην αγορά στο 2007, η Apple άλλαξε σε μεγάλο βαθμό την αγορά κινητών συσκευών και ως ένα βαθμό προκάλεσε την γέννηση νέων λειτουργικών συστημάτων για κινητές πλατφόρμες, μιας γενιά που δίνει μεγάλη προσοχή στην ευκολία και εμπειρία χρήσης και βασίζεται στην αλληλεπίδραση του με την συσκευή μέσω της οθόνης αφής. Τον Νοέμβριο του 2007 η Google ανακοίνωσε τη δημιουργία του λειτουργικού συστήματος Android [09]. Με το λανσάρισμα και του λειτουργικού συστήματος της Google, η αγορά των smartphones έχει αυξηθεί σημαντικά. Ενδεικτικά, τον Μάιο του 2010 τα smartphones που πουλήθηκαν έφτασαν το 17,3% της αγοράς κινητών τηλεφώνων παγκοσμίως. Αυτό οδήγησε τους καταναλωτές στην γνωριμία τους με τα διάφορα λειτουργικά συστήματα και τους κατασκευαστές και τις εταιρίες τηλεπικοινωνιών στη διαφήμιση των πλεονεκτημάτων των λειτουργικών συστημάτων τους.



Εικόνα 2.5: Firefox OS [11] & Ubuntu Phone [12].

Σύμφωνα με πρόσφατες μελέτες παρατηρείται μια μεγάλη αύξηση στο μερίδιο του λειτουργικού Android που αυτή την στιγμή φαίνεται να κατέχει την πρώτη θέση στην αγορά των smartphones.



Μέσα σε 3 χρόνια έχει καταφέρει να περάσει πρώτο, μπροστά και από το iOS [10] (iPhone) που κατείχε αυτή την θέση τα τελευταία χρόνια αλλά βλέπει το μερίδιο του στην αγορά να συρρικνώνεται συνεχώς στο ίδιο διάστημα. Πρόσφατα υπάρχει μία τάση, βασιζόμενη και στην ολοένα αυξανόμενη υπολογιστή ισχύ των κινητών, να εισέλθουν στην αγορά κατασκευαστές λογισμικού από τον χώρο του Linux (Ubuntu Phone) [11] αλλά και των web browsers (Firefox OS) [12] [Εικόνα 2.5]. Αυτή η μεγάλη κινητικότητα στον κλάδων του λογισμικού των κινητών πλατφορμών δικαιολογείται, αφού ο κλάδος είναι όλο και αυξανόμενος αντικαθιστώντας αργά μεν αλλά σταθερά μεγάλο μέρος των παραδοσιακών υπολογιστικών συστημάτων.

## 2.3 Το λειτουργικό σύστημα Android

### 2.3.1 Γενικά

Το Android [09] είναι ένα λειτουργικό σύστημα βασισμένο στον πυρήνα του Linux και έχει σχεδιαστεί κυρίως για συσκευές με οθόνη αφής, όπως έξυπνα τηλέφωνα και υπολογιστές ταμπλέτες. Αρχικά αναπτύχθηκε από την Android Inc, η οποία υποστηρίζονταν οικονομικά και αργότερα αγοράστηκε από την Google το 2005. Το Android παρουσιάστηκε το 2007 μαζί με την ίδρυση της Open Handset Alliance, μιας κοινοπραξία εταιριών υλικού, λογισμικού και τηλεπικοινωνιών που σκοπό έχει την προώθηση του Android. Η πρώτη δημόσια διαθέσιμη συσκευή που εκτελούσε το Android ήταν ένα έξυπνο τηλέφωνο, το HTC Dream που κυκλοφόρησε στις 22 Οκτωβρίου του 2008. Το λογότυπο για το λειτουργικό σύστημα Android είναι ένα ρομπότ σε χρώμα πράσινου μήλου και σχεδιάστηκε από τη γραφίστρια Irina Blok [Εικόνα 2.6].



Εικόνα 2.6: Το λογότυπο του Android

Το περιβάλλον επικοινωνίας με τον χρήστη (UI) του Android βασίζεται στον απευθείας χειρισμό κάνοντας χρήση κινήσεων στην οθόνη αφής που αντιστοιχούν σε φυσικές κινήσεις που θα έκανε ένας άνθρωπος όπως σύρσιμο, πάτημα και εκτεταμένο πάτημα για να χειριστεί αντικείμενα στην οθόνη.

Επιπλέον υλικό που υπάρχει στις συσκευές με Android, όπως επιταχυνσιόμετρο, γυροσκόπιο και αισθητήρας εγγύτητας, χρησιμοποιούνται από κάποιες εφαρμογές για να ανταποκριθούν σε επιπρόσθετους χειρισμούς του χρήστη, όπως η εναλλαγή της οθόνης από κατακόρυφο σε οριζόντιο προσανατολισμό ανάλογα με το πώς κρατά την συσκευή ο χρήστης [Εικόνα 2.7].



Εικόνα 2.7: Οθόνη εφαρμογών του Android [09].

Ο πηγαίος κώδικας του Android έχει εκδοθεί από την Google κάτω από την Apache License που επιτρέπει την ελεύθερη τροποποίηση του λογισμικού και την διάθεση του από κατασκευαστές συσκευών, τηλεπικοινωνιακούς παρόχους και από οποιονδήποτε ασχολείται ακόμη και ερασιτεχνικά με ανάπτυξη λογισμικού. Οι περισσότερες συσκευές με Android περιέχουν ένα συνδυασμό του Android και ειδικού λογισμικού που έχει δημιουργήσει ο κατασκευαστής της συσκευής.

Για κάθε κινητή πλατφόρμα σημαντικός παράγοντας είναι το πλήθος και η ποιότητα των εφαρμογών που υπάρχουν διαθέσιμες για λήψη. Η διάθεση των εφαρμογών για το Android γίνεται από την Google μέσω ειδικής εφαρμογής, το Play Store. Από τον Ιούλιο του 2013 το Android διαθέτει τη μεγαλύτερη βάση εφαρμογών που είναι διαθέσιμες για λήψη, οι οποίες ξεπέρασαν το 1 εκατομμύριο.

Το Android είναι πολύ δημοφιλές ανάμεσα στις εταιρείες τεχνολογίας που χρειάζονται ένα έτοιμο και χαμηλού κόστους, πλήρως παραμετροποιήσιμο λειτουργικό σύστημα. Παρόλο που αρχικά είχε σχεδιαστεί για έξυπνα κινητά τηλέφωνα και ταμπλέτες, έχει βρει το δρόμο του προς τηλεοράσεις, κονσόλες παιχνιδιών, ψηφιακές κάμερες και άλλες ηλεκτρονικές συσκευές.

Η ανοιχτού λογισμικού φύση του Android έχει δημιουργήσει μία μεγάλη κοινότητα από επαγγελματίες προγραμματιστές και μη, γεγονός που έχει ως αποτέλεσμα να υπάρχει στο διαδίκτυο πληθώρα αναφορών και λύσεων για κάθε προγραμματιστικό πρόβλημα το οποίο μπορεί να αντιμετωπίζει κάποιος. Παράλληλα έχουν συσταθεί εταιρίες που δημιούργησαν ολόκληρα APIs για παιχνίδια ή άλλες εφαρμογές, εμπλουτίζοντας το είδη πλούσιο API της Google.

Από τον Μάιο του 2012, το Android είναι το πιο δημοφιλές λειτουργικό για φορητές συσκευές φτάνοντας στο 3ο τέταρτο του 2013 το 81,6% των παγκόσμιων πωλήσεων σε έξυπνα τηλέφωνα. Το Σεπτέμβριο του 2013 οι συσκευές με Android έφτασαν το 1 Δισεκατομμύριο.

### **2.3.2 Open Handset Alliance**

Η Open Handset Alliance [13] είναι μια ομάδα 84 εταιριών τεχνολογίας και κινητής τηλεφωνίας που έχουν έρθει σε συμφωνία, για να επιταχύνουν την καινοτομία στον τομέα των κινητών και να προσφέρουν στους καταναλωτές μια πιο πλούσια και λιγότερο ακριβή εμπειρία χρήσης κινητών συσκευών. Μαζί έχουν αναπτύξει το Android™, την πρώτη ολοκληρωμένη, ανοικτή και δωρεάν κινητή πλατφόρμα. Η συμμαχία περιλαμβάνει εταιρίες κάθε είδους που σχετίζονται με κινητές συσκευές ή υπηρεσίες, όπως φορείς εκμετάλλευσης κινητής τηλεφωνίας, κατασκευαστές συσκευών, εταιρείες ημιαγωγών, εταιρείες λογισμικού αλλά και εταιρείες εμπορίας. Όλα τα μέλη της έχουν δεσμευτεί να κάνουν την πλατφόρμα μια εμπορική επιτυχία. Ορισμένες εταιρείες έχουν συμβάλει με δικής τους πνευματικής ιδιοκτησίας συστήματα στο Android ενώ άλλες εργάζονται για να είναι βέβαιο ότι τα ολοκληρωμένα κυκλώματα που παράγουν υποστηρίζουν την πλατφόρμα. Οι κατασκευαστές συσκευών και οι εταιρείες κινητής τηλεφωνίας εργάζονται για να αναπτύξουν συσκευές που βασίζονται στην πλατφόρμα.

### **2.3.3 Εφαρμογές – Play Store**

Πολύ σημαντικό κριτήριο για την επιλογή μίας κινητής συσκευής είναι το πλήθος και η ποιότητα των εφαρμογών που είναι διαθέσιμο για το λειτουργικό σύστημα που εκτελείται στην συσκευή. Μέχρι πρόσφατα τις περισσότερες εφαρμογές είχε το iOS που εκτελείται στα iPhone, iPad αλλά από το 2012 και μετά το Android αρχίζει να παίρνει το προβάδισμα, ενώ οι πιο γνωστές και σημαντικές εφαρμογές είναι διαθέσιμες και για τις 2 πλατφόρμες. Η διανομή των εφαρμογών Android γίνεται μέσω του Google Play Store [14] το οποίο ήταν παλιότερα γνωστό και ως Android Market.



Η υπηρεσία επιτρέπει στους χρήστες να περιηγηθούν και να κατεβάσουν μουσική, περιοδικά, βιβλία, ταινίες, τηλεοπτικά προγράμματα και εφαρμογές που είναι δημοσιευμένες εκεί με χρέωση ή δωρεάν. Οι χρήστες μπορούν επίσης να αγοράσουν Chromebooks και Google Nexus έξυπνες κινητές συσκευές μέσω του Google Play Store.

### 2.3.4 Εκδόσεις του Android

Η πρώτη έκδοση του Android παρουσιάστηκε το Νοέμβριο του 2007. Στις 23 Σεπτεμβρίου 2008 κυκλοφόρησε η πρώτη έξυπνη συσκευή, το HTC Dream G1, που λειτουργούσε με Android 1.0. Έπειτα ακολούθησε η έκδοση 1.1 το Φεβρουάριο του 2009, που επρόκειτο για την πρώτη αναβάθμιση της προηγούμενης έκδοσης, η οποία όμως δεν ήταν κάτι το επαναστατικό, καθώς πολλές λειτουργίες που σήμερα θεωρούνται απαραίτητες δεν υποστηρίζονταν, όπως η λειτουργία των κουμπιών αφής. Ωστόσο, για τα τότε δεδομένα, η έκδοση 1.1 ήταν ένα σημαντικό γεγονός. Έδειχνε ότι το Android επιδεχόταν αναβαθμίσεις, κάτι το οποίο άλλες πλατφόρμες δεν το υποστήριζαν, και ότι η όλη διαδικασία της αναβάθμισης ήταν μια εύκολη υπόθεση για το χρήστη.

Στον παρακάτω πίνακα παρουσιάζονται οι εκδόσεις του Android με τελευταία την 4.4.x με ονομασία kitkat σύμφωνα με την συνήθεια της Google να δίνει ονομασίες από γλυκά σε κάθε έκδοση του λειτουργικού [09].

Έκδοση	Κωδική ονομασία	Ημερομηνία	API level
4.4.x	<i>Kitkat</i>	31-Οκτ-13	19
4.3	<i>Jelly Bean</i>	24-Ιουλ-13	18
4.2.x	<i>Jelly Bean</i>	13-Νοε-12	17
4.1.x	<i>Jelly Bean</i>	9-Ιουλ-12	16
4.0.x	<i>Ice Cream Sandwich</i>	16-Δεκ-11	15
3.2	<i>Honeycomb</i>	15-Ιουλ-11	13
2.3-2.3.7	<i>Gingerbread</i>	9-Φεβ-11	10
2.2	<i>Froyo</i>	20-Μαΐ-10	8
2.0-2.1	<i>Eclair</i>	26-Οκτ-09	7
1.6	<i>Donut</i>	15-Σεπ-09	4

### 2.3.5 Η Αρχιτεκτονική του Android

Η αρχιτεκτονική του λειτουργικού συστήματος Android [Εικόνα 2.8] αποτελείται από 5 βασικές συνιστώσες [09]. Στην κορυφή βρίσκονται οι Εφαρμογές (Applications), και ακολουθούν το πλαίσιο Εφαρμογής (Application Framework), οι Βιβλιοθήκες (Libraries), ο Χρόνος Εκτέλεσης (Android

Runtime) και ο Πυρήνας Linux (Linux Kernel). Στην παρακάτω εικόνα παρουσιάζονται τα 4 διακριτά επίπεδα της αρχιτεκτονικής του Android [Εικόνα 2.8].



Εικόνα 2.8: Η Αρχιτεκτονική του Android [09].

### 2.3.6 Επιλογή του Android

Η επιλογή του Android ως το λειτουργικό σύστημα, στο οποίο αναπτύχθηκε και εκτελείται η εφαρμογή mRLGame, βασίστηκε στους παράγοντες που περιγράφονται παρακάτω:

Το Android αποτελεί το πιο διαδεδομένο λειτουργικό σύστημα για κινητές πλατφόρμες τόσο για έξυπνα τηλέφωνα αλλά και για υπολογιστές ταμπλέτες. Βασίζεται στον πυρήνα του Linux και είναι όπως και το Linux λειτουργικό σύστημα ανοιχτού πηγαίου κώδικα (Open source) και είναι πλήρως παραμετροποιήσιμο χωρίς να θέτει κανένα περιορισμό στον προγραμματιστή, εκτός φυσικά αυτών που προέρχονται από τη φιλοσοφία σχεδιασμού του. Επιπλέον βασίζεται στην γλώσσα προγραμματισμού Java η οποία είναι μοντέρνα, έχει αντικειμενοστραφή προσανατολισμό και δεν θέτει περιορισμούς με τις δυνατότητές της.

Η γλώσσα προγραμματισμού που χρησιμοποιείται από το Android, δηλαδή η Java, αποτελεί ένα ακόμη σημαντικό παράγοντα για την επιλογή του καθώς η βιβλιοθήκη RL-Lib στην οποία βασίστηκε η εφαρμογή είναι γραμμένη σε Java.

Οι παραπάνω παράγοντες αθροιστικά οδήγησαν στην επιλογή του Android ως την βέλτιστη πλατφόρμα για την ανάπτυξη μίας εφαρμογής με την ερευνητική βάση και τον προσανατολισμό του mRLGame.

# Κεφάλαιο 3

## Παιχνίδια Στρατηγικής και Πλατφόρμες Ανάπτυξης Παιχνιδιών

Σε αυτό το κεφάλαιο γίνεται μια περιγραφή των παιχνιδιών στρατηγικής και των εφαρμογών τεχνικής νοημοσύνης που έχουν χρησιμοποιηθεί σε αυτά τα παιχνίδια.

### 3.1 Παιχνίδια Στρατηγικής και Τεχνητή Νοημοσύνη

Παιχνίδι στρατηγικής [58] είναι ένα παιχνίδι όπου οι ικανότητες των παιχτών στην λήψη αποφάσεων, ακολουθώντας ένα δομημένο τρόπο σκέψης και έχοντας επίγνωση της τρέχουσας κατάστασης μπορούν να καθορίσουν το αποτέλεσμα Έχουν δε παίξει σημαντικό ρόλο στην ανάπτυξη του πολιτισμού καθώς οξύνουν την ικανότητα σκέψης και ανάλυσης προβλημάτων του ανθρώπου. Ο άνθρωπος παίζει παιχνίδια στρατηγικής από την αρχαιότητα με τα πιο γνωστά να είναι το σκάκι και το τάβλι. [61, 62].

Στα παιχνίδια στρατηγικής ο κύριος στόχος ενός παίκτη είναι η επιλογή της βέλτιστης στρατηγικής που θα ακολουθήσει κατά τη διάρκεια του παιχνιδιού. Με τον όρο στρατηγική εννοούμε την επιλογή της επόμενης κίνησης, λαμβάνοντας υπόψη την παρούσα κατάσταση του, την κατάσταση του αντιπάλου, τις επιπτώσεις της κίνησης του και την ενδεχόμενη επόμενη κίνηση του αντιπάλου [39].

Τα παιχνίδια στρατηγικής ενδείκνυνται για μελέτη λόγω της πολυπλοκότητάς τους και της έξυπνης στρατηγικής που χρειάζεται να ακολουθήσει κάποιος για να κερδίσει. Επιπλέον, οι είσοδοι του παιχνιδιού και τα κριτήρια αξιολόγησης είναι γνωστά, ενώ το περιβάλλον του παιχνιδιού, οι έγκυρες κινήσεις και οι κινήσεις τερματισμού μπορούν εύκολα να προσομοιωθούν σε έναν υπολογιστή. Η έρευνα σε αυτόν τον τομέα δεν εστιάζει μόνο στη δημιουργία ενός παίκτη που θα είναι ανταγωνιστικός εναντίον των ανθρώπων, αλλά χρησιμοποιεί το φιλικό περιβάλλον των παιχνιδιών για να ελέγξει την αποτελεσματικότητα νέων μεθόδων τεχνητής νοημοσύνης και μηχανικής μάθησης [39, 40, 41, 42].

Η έρευνα του Samuel στο παιχνίδι της Ντάμας και άλλων σύγχρονων του ερευνητών, όπως ο Shannon στο σκάκι έβαλαν τα θεμέλια της κοινής πορείας της έρευνας της τεχνητής νοημοσύνης με τα παιχνίδια στρατηγικής. Στα περίπου 60 χρόνια που έχουν μεσολαβήσει από τα πρώτα βήματα της έρευνας στον τομέα της τεχνητής νοημοσύνης, οι ερευνητές έχουν αναπτύξει μία πληθώρα τεχνικών στην προσπάθεια τους να δημιουργήσουν έξυπνα προγράμματα τα οποία να είναι σε θέση να παίξουν ανταγωνιστικά παιχνίδια στρατηγικής και σε ορισμένες περιπτώσεις έχουν καταφέρει να ανταγωνιστούν και να νικήσουν τους ανθρώπους - πρωταθλητές αρκετών παιχνιδιών.

Στη διάρκεια αυτών των χρόνων οι ερευνητές εφηύραν αλγόριθμους, όπως ο αλγόριθμος minimax με κλάδεμα που έκανε εφικτή ως ένα βαθμό, τη διερεύνηση του δένδρου των καταστάσεων του παιχνιδιού, δοκίμασαν τεχνικές εξαντλητικής διερεύνησης των πιθανών καταστάσεων εκμεταλλευόμενοι την όλο και περισσότερη υπολογιστική δύναμη που έφεραν οι εξελίξεις στο χώρο του υλικού των υπολογιστών [42].

Στη δεκαετία του 1990 η IBM, πρώτα με το Deep Thought και στη συνέχεια με το Deep Blue, επένδυσε πολλή προσπάθεια για να δημιουργήσει ένα πρόγραμμα που θα νικήσει τους καλύτερους σκακιστές του κόσμου. Ο στόχος επιτεύχθηκε όταν ο Deep Blue κέρδισε τον τότε πρωταθλητή σε μία σειρά αγώνων και η τεχνητή νοημοσύνη απέκτησε δημοσιότητα στο ευρύ κοινό. Το μειονέκτημα των προσπαθειών όπως αυτή του Deep Blue είναι ότι οι ικανότητες του συστήματος βασίζονται στην ικανότητα των προγραμματιστών του να ορίσουν αλγορίθμους για το συγκεκριμένο παιχνίδι και στη

δυνατότητα του ειδικά σχεδιασμένου hardware στην προσπάθεια διερεύνησης του δένδρου καταστάσεων.

Μία άλλη τάση στον τομέα της μηχανικής μάθησης που δεν γνώρισε αντίστοιχη δημοσιότητα με αυτή του Deep Blue, προσπαθεί να προσεγγίσει το πρόβλημα της υλοποίησης μίας βέλτιστης στρατηγικής με μια πιο γενική μέθοδο. Χρησιμοποιώντας τεχνικές Ενισχυτικής Μάθησης μέσω αλληλεπίδρασης με το περιβάλλον ένας πράκτορας μπορεί να προσεγγίσει/μάθει μία βέλτιστη τακτική μέσα από την εμπειρία που αποκτά κατά την διάρκεια αυτής της αλληλεπίδρασης. Ο πράκτορας επιλέγει κινήσεις και λαμβάνει από το περιβάλλον ανταμοιβές ή τιμωρίες ανάλογα με το πόσο καλή ήταν η κίνηση που επέλεξε. Κατόπιν ο πράκτορας προσπαθεί να προσαρμόσει τη στρατηγική του στην προσπάθεια να επιτύχει μακροπρόθεσμη αύξηση του συνόλου των ανταμοιβών του.

Το 1988 ο Sutton διατύπωσε τη μέθοδο TD(λ) για τη Μάθηση χρονικών διαφορών (Temporal Difference Learning)[19]. Η πιο χαρακτηριστική επιτυχία της μεθόδου TD(λ) είναι το TD-Gammon για το παιχνίδι τάβλι που υλοποιήθηκε από τον Tesauro[20].

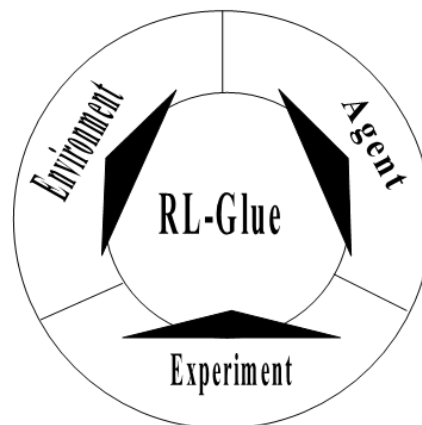
Χρησιμοποιώντας τεχνικές της ενισχυτικής μάθησης (Reinforcement Learning) και ύστερα από 1,5 εκατομμύρια παιχνίδια με τον εαυτό του το TD-Gammon κατάφερε να όχι μόνο να έχει φτάσει σε επίπεδο ανάλογο των πρωταθλητών στο τάβλι, αλλά να κάνει γνωστές νέες στρατηγικές για κινήσεις στο ξεκίνημα του παιχνιδιού[18].

## **3.2 Πλατφόρμες Ανάπτυξης Παιχνιδιών Τεχνητής Νοημοσύνης**

Έχουν γίνει πολλές προσπάθειες να δημιουργηθεί μία μηχανή τεχνητής νοημοσύνης που θα χρησιμοποιείται από όλα τα παιχνίδια στρατηγικής. Ενδεικτικές υλοποιήσεις είναι: το RL-Glue, το ORTS, το GameMaker και το General Game Player, οι οποίες διαφέρουν ως προς την φιλοσοφία προσέγγισης, με το RL-Glue και το ORTS να προσφέρουν εγγενής υποστήριξη τεχνητής νοημοσύνης ενώ το GameMaker και General Game Player δεν προσφέρουν λειτουργίες τεχνητής νοημοσύνης, αλλά αφήνουν σε αυτόν που θα τα χρησιμοποιήσει να αναλάβει την υλοποίηση αυτής της λειτουργίας.

### 3.2.1 Περιγραφή του RL-Glue

Το RL-Glue χαρακτηρίζεται ως ένα πρότυπο πακέτο λογισμικού για την εκτέλεση πειραμάτων ενισχυτικής μάθησης. Όπως αναφέρεται στο άρθρο των Tanner και White [21], το RL-Glue ορίζει ένα γενικό τρόπο με τον οποίο ερευνητές μπορούν να εκτελέσουν πειράματα τα οποία εμπλέκουν διαφορετικά περιβάλλοντα και διαφορετικούς πράκτορες τεχνητής νοημοσύνης. Κατά τους δημιουργούς του RL-Glue οι ερευνητές στο αντικείμενο της ενισχυτικής μάθησης τείνουν να υλοποιούν τα προγράμματα των πειραμάτων τους σε διάφορες γλώσσες και με διάφορα μη συμβατά εργαλεία. Αυτό το γεγονός αποτελεί τροχοπέδη στις συνέργειες μεταξύ των μελών της ερευνητικής κοινότητας. Έχει παρατηρηθεί δε ότι μερικές φορές είναι δύσκολο ή αδύνατο να αναπαράγει ακριβώς ένας ερευνητής το πείραμα ενός άλλου ερευνητή. Προσπαθώντας να καλύψει το κενό που υπάρχει στο πεδίο της προτυποποίησης των υλοποιήσεων των πειραμάτων ενισχυτικής μάθησης, η ερευνητική ομάδα του RL-Glue έχει δημιουργήσει ένα πρωτόκολλο επικοινωνίας μεταξύ των κύριων οντοτήτων που μετέχουν σε ένα πείραμα ενισχυτικής μάθησης. Η φιλοσοφία λειτουργίας του RL-Glue φαίνεται παρακάτω [Εικόνα 3.1].



Εικόνα 3.1. Φιλοσοφία λειτουργίας RL-Glue

### 3.2.2 Περιγραφή του ORTS (Open Real-Time Strategy)

Το ORTS (Open Real-Time Strategy) είναι ένα περιβάλλον προγραμματισμού και εκτέλεσης παιχνιδιών στρατηγικής, πραγματικού χρόνου. Η προσπάθεια του ORTS καθοδηγείται από τον καθηγητή του πανεπιστημίου Alberta Michael Buro. Η ομάδα του ORTS δημιούργησε μία πλατφόρμα δημιουργίας και εκτέλεσης παιχνιδιών RTS ανοιχτού κώδικα με στόχο την παροχή ενός περιβάλλοντος όπου ερευνητές θα μπορούν να ελέγξουν προβλήματα τεχνητής νοημοσύνης σχετικά με τα παιχνίδια στρατηγικής

πραγματικού χρόνου και να θέσει μία νέα πρόταση στη βιομηχανία RTS παιχνιδιών για μία εναλλακτική αρχιτεκτονική για ένα RTS παιχνίδι.

### **3.2.3 Περιγραφή του Game Maker**

Το GameMaker είναι μια πλατφόρμα δημιουργίας παιχνιδιών που απευθύνεται σε ένα ευρύ κοινό καθώς ο χρήστης του μπορεί να δημιουργήσει κάποιο απλό παιχνίδι χρησιμοποιώντας τις επιλογές του UI. Δημιουργήθηκε αρχικά από τον καθηγητή Mark Overmars, και πλέον έχει κοινότητα ενεργών χρηστών που υλοποιούν τα παιχνίδια τους με το Gamemaker. Το GameMaker δεν είναι λογισμικό ανοιχτού κώδικα αλλά η βασική έκδοση του παρέχεται δωρεάν ενώ η εμπορική έκδοση η οποία έχει παραπάνω λειτουργικότητα διατίθεται έναντι κάποιου αντιτίμου. Το λογισμικό έχει υλοποιηθεί σε περιβάλλον προγραμματισμού Delphi και είναι διαθέσιμο μόνο για λειτουργικό σύστημα Microsoft Windows και Mac OS. Τέλος υπάρχει και μια νέα έκδοση που δημιουργεί παιχνίδια που μπορούν να εκτελεστούν σε οποιοδήποτε browser μπορεί να υποστηρίξει HTML5 [22,23].

### **3.2.4 Περιγραφή του General Game Playing (GGP)**

Ένα σύστημα χαρακτηρίζεται σαν General Game Playing (GGP) όταν μπορεί να χρησιμοποιήσει/διαβάσει την περιγραφή των καταστάσεων και τους κανόνες του παιχνιδιού που επικοινωνούνται στη γλώσσα λογικής GDL (Game Description Language) και μπορεί να σχεδιάσει τη στρατηγική έτσι ώστε να παίξει αποτελεσματικά το παιχνίδι [24]. Το GGP στοχεύει στη δημιουργία πρακτόρων που μπορούν να παίξουν οποιοδήποτε παιχνίδι (πεπερασμένων καταστάσεων, όπου δεν υπάρχει κρυμμένη πληροφορία και είναι μηδενικού αθροίσματος) αρκεί να τους δοθούν οι κανόνες του παιχνιδιού κωδικοποιημένοι σε GDL. Το εργαστήριο λογικής του πανεπιστημίου Standford έχει ενεργό ρόλο στην έρευνα για τα συστήματα General Game Playing.



# Κεφάλαιο 4

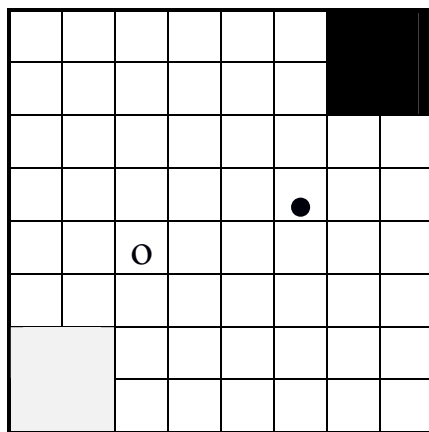
## Το Παιχνίδι Στρατηγικής RLGame

Σε αυτό το κεφάλαιο γίνεται περιγραφή του παιχνιδιού στρατηγικής RLGame, των κανόνων του παιχνιδιού, των επιλογών τεχνητής νοημοσύνης καθώς και παρουσίαση των υπαρχόντων υλοποιήσεων.

### 4.1 Περιγραφή του Παιχνιδιού

Το RLGame είναι ένα παιχνίδι στρατηγικής που παίζεται ανάμεσα σε δύο παίκτες χρησιμοποιώντας πιόνια που κινούνται πάνω σε μία επιφάνεια τύπου «σκακιέρας». Η σκακιέρα μοιάζει με αυτή που παίζονται το σκάκι ή η ντάμα με την διαφορά ότι στην πάνω δεξιά και στην κάτω αριστερά γωνία ορίζονται 2 τετραγωνικές βάσεις. Κάθε βάση ανήκει σε έναν παίκτη και είναι η αφετηρία για τα πιόνια του.

Το τυπικό παιχνίδι παίζεται σε επιφάνεια σκακιέρας 8x8 με βάση 2x2 [Εικόνα 4.1] και ο κάθε αντίπαλος έχει στην διάθεση του 10 πιόνια και οι παίκτες καλούνται «λευκός» και «μαύρος». Τα πιόνια του «λευκού» παίκτη εκκινούν πάντα από την κάτω αριστερή γωνία της σκακιέρας όπου βρίσκεται και η βάση του, ενώ τα πιόνια του «μαύρου» εκκινούν από την πάνω δεξιά γωνία.



Εικόνα 4.1: Σκακιέρα 8x8 με βάση 2x2

Στόχος του κάθε παίκτη είναι να καταφέρει να τοποθετήσει ένα πiónι στην αντίπαλη βάση ενώ παράλληλα πρέπει να προστατεύει τη δικιά του βάση από τα πiónια του αντιπάλου. Αυτός που θα καταφέρει να βάλει πρώτος κάποιο πiónι του στη βάση του αντιπάλου θεωρείται νικητής, ενώ το παιχνίδι λήγει και στην περίπτωση που κάποιος παίκτης δεν έχει πια πiónια στην διάθεση του (σε αυτή την περίπτωση ο αντίπαλος θεωρείται νικητής). Στο παιχνίδι RLGame δεν προβλέπεται ισόπαλο αποτέλεσμα.

## 4.2 Κανόνες

Οι κανόνες του παιχνιδιού περιγράφονται αναλυτικά στις εργασίες [15,16,17,25,]. Στην προηγούμενη παράγραφο αναφέραμε τα συστατικά του παιχνιδιού και τις τελικές καταστάσεις του παιχνιδιού. Σε αυτή την παράγραφο θα αναφερθούμε στις κινήσεις των πιονιών και στις περιπτώσεις που επισύρουν χάσιμο πιονιών.

Κάθε πiónι εκκινεί από την βάση του, που θεωρείται ένα ενιαίο τετράγωνο, επομένως κάθε πiónι που εκκινεί από την βάση μπορεί να μετακινηθεί σε κάποιο ελεύθερο από τα παρακείμενα στη βάση τετράγωνα που είναι ελεύθερα.

Τα πiónια μπορούν να κινηθούν οριζόντια και κάθετα στα ελεύθερα τετράγωνα της σκακιέρας (1 τετράγωνο την φορά). Η μόνη προϋπόθεση είναι να μην μειώνεται η μέγιστη διαφορά του από τη βάση του, συνεπώς δεν επιτρέπονται οι κινήσεις προς τα πίσω.

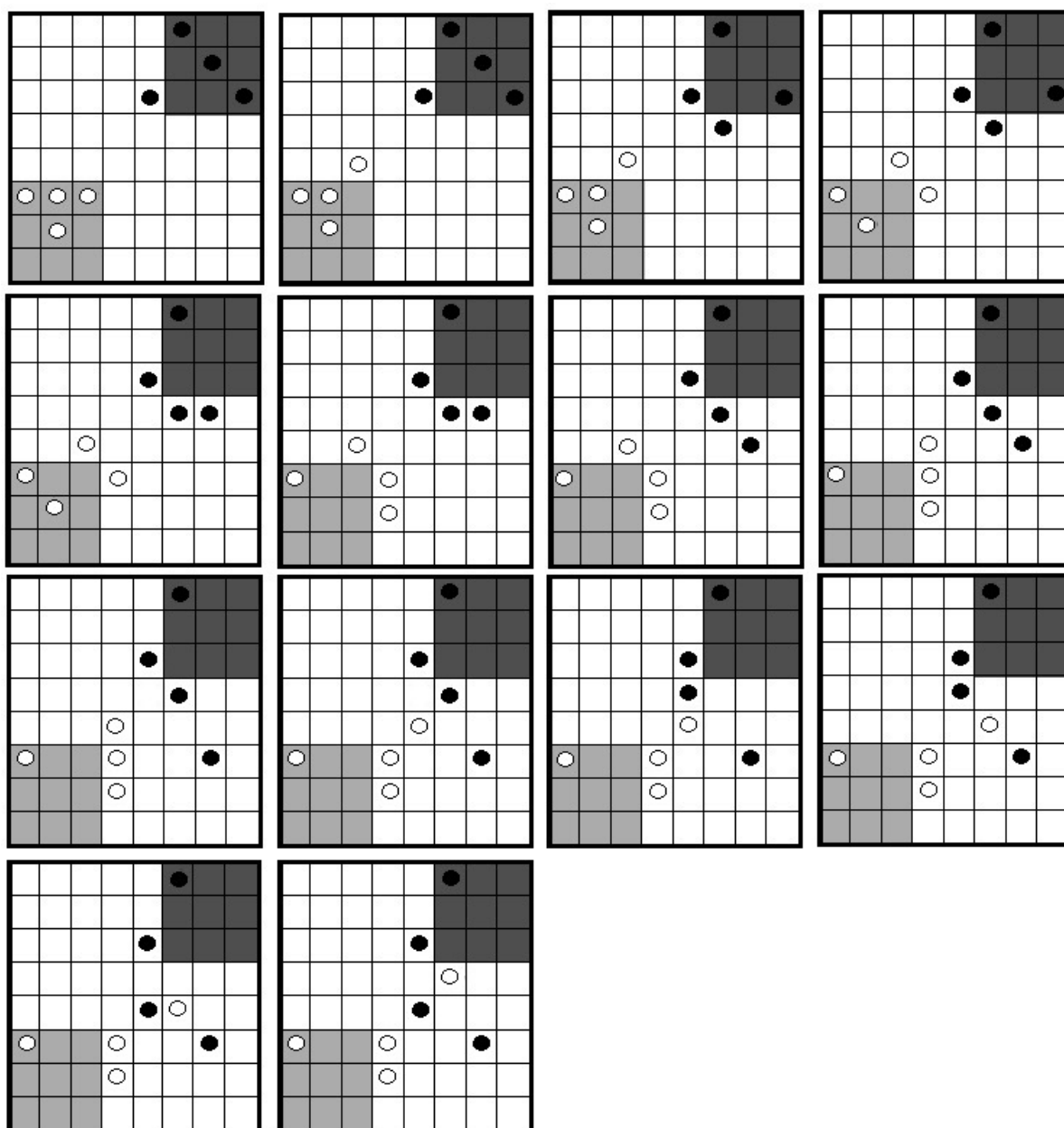
Η τελευταία προϋπόθεση αποτυπώνεται στους ακόλουθους ορισμούς:

Έστω  $(x,y)$  θέση του πιονιού στην σκακιέρα, το πιόνι μπορεί να μετακινηθεί οριζόντια ή κάθετα κατά μία θέση  $(x,y')$  ή  $(x',y)$  αν και μόνο αν ισχύει:

- $\max(x - a, y - a) \leq \max(x - a, y' - a)$  ή  $\max(x - a, y - a) \leq \max(x' - a, y - a)$  εάν το πιόνι ανήκει στον «λευκό» παίκτη ή αν ισχύει:
- $\max(n - a - x, n - a - y) \leq \max(n - a - x, n - a - y')$  ή  $\max(n - a - x, n - a - y) \leq \max(n - a - x', n - a - y)$  εάν το πιόνι ανήκει στο «μαύρο» παίκτη.

Στο παιχνίδι προβλέπονται καταστάσεις που επισύρουν απώλεια πιονιών. Ένα πιόνι διαγράφεται όταν όλες οι νόμιμες κινήσεις είναι προς κατειλημμένες θέσεις. Οι θέσεις μπορεί να είναι κατειλημμένες είτε από πιόνια αντιπάλου είτε από πιόνια του ίδιου παίκτη.

Στο ακόλουθο σχήμα που δημοσιεύτηκε στην εργασία [15] παρατίθεται ένα παράδειγμα παιχνιδιού στο οποίο κερδίζει ο λευκός παίκτης [Εικόνα 4.2] με την ιδιαιτερότητα ότι οι βάσεις είναι  $3 \times 3$ .



Εικόνα 4.2 : Παράδειγμα παιχνιδιού RLGame όπου εκκινεί ο μαύρος και κερδίζει ο λευκός[15]

## 4.3 Τεχνητή Νοημοσύνη

Οι τεχνικές μάθησης που έχουν ακολουθηθεί, έτσι ώστε οι υπολογιστικοί παίκτες του παιχνιδιού να βελτιώσουν την τακτική τους, βασίζονται στις αρχές της Ενισχυτικής Μάθησης (Reinforcement Learning) και πιο συγκεκριμένα στη μέθοδο TD( $\lambda$ ) – Ενισχυτική Μάθηση χρονικών διαφορών όπως αυτή περιγράφεται στην εργασία του Sutton [19]. Ακόμη για κάθε παίκτη χρησιμοποιείται μία υλοποίηση Νευρωνικού Δικτύου (Neural Network) που χρησιμοποιείται για την κατά προσέγγιση

αναπαράσταση του χώρου καταστάσεων του κάθε παίκτη κατά τον υπολογισμό της συνάρτησης που εκτιμά την ποιότητα κάθε κατάστασης (value function).

Οι επιλογές των τεχνικών μάθησης του RLGame, έχουν περιγραφεί διεξοδικά στις εργασίες [15 - 17, 25 - 27] που αποτελούν και τη βάση για τις αναφορές που γίνονται σε αυτό το κεφάλαιο. Στις ακόλουθες παραγράφους περιγράφονται οι ιδιότητες του παιχνιδιού που το καθιστούν κατάλληλο για την εφαρμογή των επιλεγμένων τεχνικών Ενισχυτικής Μάθησης με χρήση Νευρωνικών Δικτύων καθώς και οι μηχανισμοί Ενισχυτικής μάθησης που χρησιμοποιούνται.

#### 4.3.1 Ιδιότητες Παιχνιδιού

Στο παιχνίδι στρατηγικής RLGame υπάρχει διαθέσιμη η πλήρης πληροφορία της κατάστασης του παιχνιδιού, καθώς δεν υπάρχει κρυμμένη πληροφορία και οι δυο παίκτες μπορούν, να παρατηρήσουν τις θέσεις των πιονιών ανά πάσα στιγμή.

Πρόκειται για ένα παιχνίδι στρατηγικής με εναλλαγή σειράς για τους παίκτες (turn based) όπου ο κάθε παίκτης έχει τη δυνατότητα σε μια καθορισμένη χρονική περίοδο να αναλύσει τις καταστάσεις του παιχνιδιού και να επιλέξει την κίνηση που θεωρεί ότι θα τον οδηγήσει στο επιθυμητό αποτέλεσμα. Κατόπιν η σειρά εναλλάσσεται και ο αντίπαλος καλείται να επιλέξει την δική του κίνηση.

Οι παίκτες δρουν ανταγωνιστικά με σκοπό τη νίκη και το RLGame αποτελεί ένα παιχνίδι μηδενικού αθροίσματος (zero-sum). Συνεπώς με βάση τη θεωρία παιχνιδιών στο τέλος του παιχνιδιού ότι κερδίζει ο νικητής έχει χαθεί από τον αντίπαλο και το άθροισμα των ανταμοιβών και των ποινών είναι πάντα μηδενικό [63].

Το σύνολο των καταστάσεων του RLGame είναι πεπερασμένο και εξαρτάται μόνο από το μέγεθος της σκακιάρας, το μέγεθος των βάσεων, τον αριθμό των πιονιών και των κανόνων των κινήσεων που διέπουν το παιχνίδι. Τέλος, η τελική κατάσταση του παιχνιδιού οδηγεί πάντα σε νίκη ενός παίκτη καθώς οι τελικές καταστάσεις είναι πλήρως ορισμένες – ο νικητής θα καταφέρει να εισβάλει στην βάση του αντιπάλου ή θα προκαλέσει μία τελική κατάσταση στην οποία ο αντίπαλος θα χάσει όλα του τα πιόνια.

### 4.3.2 Εφαρμογή Ενισχυτικής Μάθησης

Για τον μηχανισμό μάθησης, η ερευνητική ομάδα που δημιούργησε το RLGame αποφάσισε να χρησιμοποιήσει τεχνικές της Ενισχυτικής Μάθησης (Reinforcement Learning). Κεντρικό ρόλο στο μηχανισμό μάθησης έχει το σύστημα αμοιβών που λαμβάνει από το περιβάλλον με βάση τις ακόλουθες συνθήκες:

- Αν η κατάσταση είναι τελική, ο νικητής επιβραβεύεται με +100 και ο άλλος παίκτης τιμωρείται με -100.
- εάν η κατάσταση δεν είναι τελική, υπολογίζεται κάθε φορά η διαφορά των πιονιών των δύο αντιπάλων (απόρροια της απαλοιφής των πιονιών που δεν έχουν διαθέσιμες κινήσεις) και πολλαπλασιάζεται με τον παράγοντα  $1/(\text{αρχικό πλήθος πιονιών})$ .

Ο μηχανισμός μάθησης χρησιμοποιεί την αποδιδόμενη αμοιβή (reward) για την εκπαίδευσή του. Ο σκοπός του είναι να αξιολογεί τις πιθανές κινήσεις ανάλογα με την αμοιβή που θα του επιφέρουν. Ο μηχανισμός δεν επιλέγει πάντα την κίνηση με την μεγαλύτερη αναμενόμενη ανταμοιβή. Έχει προβλεφθεί να επιλέγει και τυχαίες κινήσεις στην προσπάθεια του να εξερευνήσει νέες κινήσεις (exploration) που πιθανόν να οδηγούν σε μεγαλύτερη μελλοντική αμοιβή. Ο μηχανισμός μάθησης ακολουθεί μια άπληστη στρατηγική ( $\epsilon$ - greedy policy) που στο 90% των περιπτώσεων (όρισμα του προγράμματος με προεπιλεγμένη τιμή 90%) το σύστημα επιλέγει εκείνες τις κινήσεις που έχουν τη μεγαλύτερη αναμενόμενη αμοιβή με βάση την γνώση που ήδη κατέχει (exploitation) και στο υπόλοιπο 10% των περιπτώσεων το σύστημα επιλέγει τυχαίες κινήσεις (exploration).

Για την παράμετρο του ρυθμού μείωσης  $\gamma$  (discount rate parameter) που καθορίζει την αξία των μελλοντικών αμοιβών, ο μηχανισμός μάθησης έχει προεπιλεγμένη τιμή 0,95 που επιλέχθηκε προκειμένου το σύστημα να υιοθετήσει μια μακροπρόθεσμη στρατηγική λαμβάνοντας σοβαρά υπόψη του τις μελλοντικές αμοιβές.

Ο μηχανισμός μάθησης χρησιμοποιεί την online έκδοση του αλγόριθμου μάθησης χρονικών διαφορών (temporal difference learning algorithm) TD ( $\lambda$ ). Στην online έκδοση του αλγόριθμου η ανανέωση πραγματοποιείται σε κάθε βήμα. Ο παράγοντας παράληψης (forgetting factor)  $\lambda$  έχει προεπιλεγμένη τιμή 0,5. Η παράμετρος αυτή καθορίζει τη διάρκεια ισχύος της ανάθεσης πίστωσης (credit assignment) και η τιμή επιλέχθηκε με σκοπό το κάθε λάθος να επηρεάσει την αξιολόγηση μόνο των τελευταίων 6-7 κινήσεων του παιχνιδιού.

Ο μηχανισμός χρησιμοποιεί ίχνη καταλληλότητας που ανανεώνονται σε κάθε βήμα. Κρίθηκε σκόπιμο να χρησιμοποιηθούν ίχνη αντικατάστασης (replacing traces) και όχι ίχνη συσσώρευσης (accumulating traces) εξαιτίας γνωστών μειονεκτημάτων που έχουν τα τελευταία με πιο σημαντικό το πρόβλημα ότι μια επαναλαμβανόμενη λάθος κίνηση εμποδίζει τη μάθηση καθώς το ίχνος της κακής κίνησης αυξάνεται.

Για τον υπολογισμό της βέλτιστης στρατηγικής του υπολογιστικού παίκτη που χρησιμοποιεί τον μηχανισμό μάθησης δεν είναι εφικτή η χρήση ενός πίνακα αντιστοίχισης καταστάσεων και τιμών αξιολόγησης. Αυτό οφείλεται στο γεγονός ότι όσο μεγαλύτερες είναι οι διαστάσεις της σκακιέρας ή ο αριθμός των παικτών τόσο περισσότερες είναι οι πιθανές καταστάσεις του παιχνιδιού. Η ανάγκη για γενίκευση, του κατά προσέγγιση υπολογισμού της τιμής αξιολόγησης μίας κατάστασης, ικανοποιείται από την επιλογή των νευρωνικών δικτύων.

#### 4.3.3 Εφαρμογή Νευρωνικών Δικτύων

Στο παιχνίδι χρησιμοποιήθηκαν δύο νευρωνικά δίκτυα. Έχει ορισθεί ξεχωριστό δίκτυο για κάθε παίκτη για το λόγο ότι ο χώρος των καταστάσεων του ενός παίκτη δεν έχει κοινά στοιχεία με το χώρο των καταστάσεων του άλλου παίκτη. Για την εκπαίδευση του κάθε νευρωνικού χρησιμοποιείται ο απλούστερος δυνατός (vanilla) backpropagation αλγόριθμος. Το δίκτυο που χρησιμοποιείται στο RLGame ακολουθεί αρχιτεκτονική 3 επιπέδων. Πιο συγκεκριμένα, το κάθε νευρωνικό έχει (με  $n$ : η διάσταση της σκακιέρας,  $a$ : η διάσταση της βάσης):

- το επίπεδο εισόδου (input layer) που περιέχει  $2 \cdot (n^2 - 2 \cdot a^2 + 5)$  νευρώνες
- το κρυμμένο επίπεδο (hidden layer) που περιέχει  $n^2 - 2 \cdot a^2 + 5$  νευρώνες
- το επίπεδο εξόδου (output layer) που περιέχει 1 νευρώνα.

Ως είσοδος στο νευρωνικό χρησιμοποιήθηκε ένας μονοδιάστατος πίνακας τιμών που αναπαριστά την κατάσταση του παιχνιδιού. Ο πίνακας είναι δυαδικός (με τιμές 0 και 1) και έχει μήκος  $2 \cdot (\text{DIMBOARD}^2 - 2 \cdot \text{DIMBASE}^2 + 5)$ . Όπως φαίνεται στον μονοδιάστατο πίνακα ορίζουμε τουλάχιστον 2 φορές τον αριθμό των διαθέσιμων θέσεων της σκακιέρας  $\text{DIMBOARD}^2 - 2 \cdot \text{DIMBASE}^2$ . Αυτό συμβαίνει γιατί η αποτύπωση των πιονιών του λευκού και του μαύρου παίκτη ορίζονται σε διαφορετική ομάδα κελιών του δυαδικού πίνακα. Για τον λευκό παίκτη οι θέσεις στη σκακιέρα έχουν την τιμή 1 όταν ένα πiónι του λευκού βρίσκεται σε αυτή τη θέση ενώ όλα τα άλλα στοιχεία έχουν τιμή 0. Το αντίστοιχο συμβαίνει στα

κελιά που ορίζουν τη σκακιέρα του μαύρου παίκτη. Ακόμη για κάθε παίκτη χρησιμοποιούνται 4 νευρώνες οι οποίοι αποτυπώνουν το ποσοστό που είναι ακόμη μέσα στη βάση. Τέλος ορίζεται 1 νευρώνας για κάθε παίκτη που παίρνει την τιμή 1 όταν αυτός ο παίκτης είναι ο νικητής.

Καταρχήν ακολουθείται η συνήθης πρακτική για την αρχικοποίηση των βαρών ενός νευρωνικού δικτύου και τα βάρη αρχικοποιούνται με τυχαίες μικρές τιμές. Σε κάθε αλλαγή κατάστασης, το νευρωνικό ενεργοποιείται με κάποια είσοδο και ξεκινάει η φάση της ενεργοποίησης της έμπροσθεν διασποράς (activation forward propagation phase) όπου υπολογίζονται οι τιμές του κρυμμένου επιπέδου και του επιπέδου εξόδου με τη βοήθεια μιας σιγμοειδούς συνάρτησης ενεργοποίησης (sigmoid activation function). Κατόπιν ακολουθεί η φάση της προς τα πίσω διασποράς του λάθους (error backward propagation phase) στην οποία υπολογίζεται το λάθος για το επίπεδο εξόδου. Με βάση το λάθος εξόδου τα βάρη των συνδέσεων του νευρωνικού (κρυμμένο και εισαγωγής) ανανεώνονται.

## 4.4 Υπάρχουσες Υλοποιήσεις

Η υλοποίηση του συστήματος έχει γίνει σε γλώσσα προγραμματισμού Java και η αρχική υλοποίηση έχει γίνει το 2001 [16]. Οι υπάρχουσες υλοποιήσεις μπορούν να υποστηρίξουν πειράματα κατά τα οποία υπολογιστικοί παίκτες που ακολουθούν ίδιες τεχνικές μάθησης και ίδιες ρυθμίσεις συναγωνίζονται σε μία σειρά παιχνιδιών (Self Play πειράματα), πειράματα στα οποία οι υπολογιστικοί παίκτες με διαφορετική δυναμική, με διαφορετικές επιμέρους ρυθμίσεις ή με διαφορετικό επίπεδο εκπαίδευσης, συναγωνίζονται σε μία σειρά παιχνιδιών, πειράματα στα οποία κάποιος άνθρωπος παίζει εναντίον ή εκπαιδεύει κάποιο υπολογιστικό παίκτη, ή τέλος πειράματα στα οποία υλοποίηση του αλγόριθμου minimax παίζει εναντίον/εκπαιδεύει κάποιο υπολογιστικό παίκτη.

### 4.4.1 Υπολογιστής εναντίον Υπολογιστή

Αυτή είναι η πιο παλιά μορφή του συστήματος RLGame στην οποία έχουν βασιστεί σειρά πειραμάτων που αναφέρονται στις δημοσιεύσεις [15-17, 25-27]. Το πρόγραμμα εκτελεί σειρά πειραμάτων ανάμεσα σε 2 υπολογιστικούς παίκτες οι οποίοι χρησιμοποιούν όμοιες τεχνικές ενισχυτικής μάθησης και νευρωνικών δικτύων. Στην αρχική έκδοση ο χρήστης μπορούσε να εκτελέσει πειράματα χωρίς να μπορεί να επέμβει στις ρυθμίσεις των παικτών (έπρεπε να επαναπρογραμματίσει το πρόγραμμα). Σε πιο πρόσφατες εκδόσεις δόθηκε η δυνατότητα στον χρήστη να εκτελέσει το πρόγραμμα με ορίσματα που διαφοροποιούν τις ρυθμίσεις των 2 παικτών. Στην έκδοση που παρέλαβα μπορούσαν να ορισθούν



τα ακόλουθα ορίσματα για κάθε παίκτη: Τιμή ρυθμού μείωσης  $\gamma$ , τιμή παράγοντα παράληψης  $\lambda$ , τιμή ανταμοιβής στην περίπτωση νίκης, τιμή e-greedy πολιτικής.

#### **4.4.2 Minimax εναντίον Υπολογιστή**

Αυτή η μορφή του παιχνιδιού RLGame έχει προστεθεί αργότερα στο σύστημα διεξαγωγής πειραμάτων RLGame. Σε αυτή την έκδοση βασίζονται πειράματα που αναφέρονται στη δημοσίευση [27]. Το πρόγραμμα εκτελεί σειρά πειραμάτων ανάμεσα σε έναν minimax παίκτη (ο λευκός είναι πάντα ο minimax) και έναν υπολογιστικό παίκτη ο οποίος χρησιμοποιεί τον μηχανισμό ενισχυτικής μάθησης. Σε αυτή την έκδοση μπορούν να ορισθούν τα ακόλουθα ορίσματα για κάθε παίκτη: Τιμή ρυθμού μείωσης  $\gamma$ , τιμή παράγοντα παράληψης  $\lambda$ , τιμή ανταμοιβής στην περίπτωση νίκης. Η τιμή e-greedy πολιτικής ορίζεται μόνο για τον μαύρο παίκτη ενώ το βάθος που χρησιμοποιείται από τον minimax λευκό παίκτη είναι constant του προγράμματος.

#### **4.4.3 Άνθρωπος εναντίον Υπολογιστή**

Αυτή η μορφή του παιχνιδιού RLGame έχει προστεθεί από την δημοσίευση [17]. Το πρόγραμμα αναλαμβάνει την διεξαγωγή παιχνιδιών ανάμεσα σε έναν άνθρωπο (λευκός) και έναν υπολογιστικό παίκτη ο οποίος χρησιμοποιεί τον μηχανισμό ενισχυτικής μάθησης. Η έκδοση αυτή χρησιμοποιήθηκε σε πειράματα που είχαν σκοπό να μετρήσουν την απόδοση του υπολογιστικού παίκτη εναντίον ανθρώπων, ενώ χρησιμοποιήθηκε και σε πειράματα όπου ο άνθρωπος αναλάμβανε να εκπαιδεύσει καταρχήν ένα νευρωνικό δίκτυο σε μια βέλτιστη τακτική και κατ' επέκταση τον αντίπαλο υπολογιστικό παίκτη σε παιχνίδι ενάντια σε έναν δυνατό παίκτη.

# Κεφάλαιο 5

## Πρωτότυπη Βιβλιοθήκη για Παιχνίδια Ενισχυτικής Μάθησης - RLLib

Σε αυτό το κεφάλαιο περιγράφεται η δομή και η λειτουργία της βιβλιοθήκης για την υλοποίηση παιχνιδιών ενισχυτικής μάθησης RLLib..

### 5.1 Γενικά

Η βιβλιοθήκη RLLib αναπτύχθηκε στα πλαίσια Μεταπτυχιακής Διπλωματικής Διατριβής [01] που ως στόχο είχε τον διαχωρισμό του μηχανισμού του παιχνιδιού από τον Μηχανισμό ενισχυτικής μάθησης στο παιχνίδι RLGame.

## 5.2 Αρχιτεκτονική

Η υλοποίηση του RLlib αποτελείται από 2 διαφορετικές λογικές ομαδοποιήσεις που αντιπροσωπεύουν μια διαφορετική όψη του συστήματος. Οι λογικές αυτές ομαδοποιήσεις που στην ονοματολογία της Java καλούνται πακέτα είναι οι εξής :

Όνομα Πακέτου	Περιγραφή Πακέτου
org.rlgame.common	<b>Πακέτο βοηθητικών κλάσεων</b> Το πακέτο αυτό περιέχει τις κοινές βοηθητικές κλάσεις που χρησιμοποιούνται στην αμφίδρομη επικοινωνία του υποσυστήματος λογικής παιχνιδιού και του υποσυστήματος μάθησης.
org.rlgame.ai	<b>Πακέτο Μάθησης</b> Το πακέτο αυτό περιέχει τις κλάσεις του υποσυστήματος μάθησης.

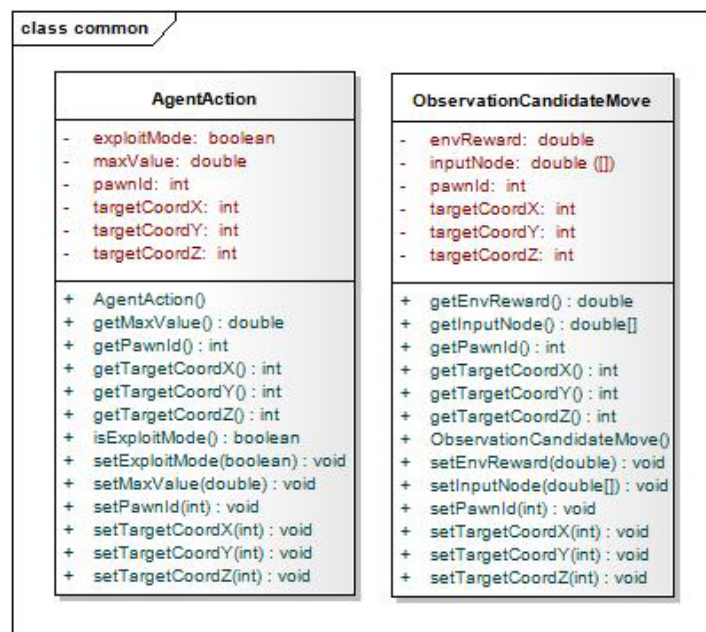
### 5.2.1 Πακέτο Βοηθητικών Κλάσεων

Στο πακέτο βοηθητικών κλάσεων του μηχανισμού μάθησης έχουν ορισθεί οι 2 κλάσεις που χρησιμοποιούνται για την επικοινωνία της παρατήρησης του περιβάλλοντος και την επιλεγμένη ενέργεια του πράκτορα ενισχυτικής μάθησης. Ο σκοπός των κλάσεων αυτών είναι να ορίσουν αντικείμενα που απλά επικοινωνούν κάποιες τιμές και οι μόνες μέθοδοι που έχουν δημιουργηθεί είναι αυτές που θέτουν ή επιστρέφουν αυτές τις τιμές

Κλάση	Περιγραφή
ObservationCandidateMove.java	Κλάση που χρησιμοποιείται για την επικοινωνία των πιθανών επόμενων κινήσεων του υπολογιστικού παίκτη ενισχυτικής μάθησης. Στην κλάση αυτή έχουν ορισθεί μεταβλητές που χρησιμοποιούνται από τον μηχανισμό μάθησης, όπως ο δυαδικός πίνακας που αντιστοιχεί στην κατάσταση και θα χρησιμοποιηθεί σαν είσοδος για το νευρωνικό δίκτυο, και η ανταμοιβή του περιβάλλοντος που χρησιμοποιείται κατά την επιλογή της κίνησης από τον μηχανισμό μάθησης. Ακόμη έχουν ορισθεί οι

	<p>ακόλουθες τιμές που δεν χρησιμοποιούνται από τον μηχανισμό μάθησης στην διαδικασία επιλογής κίνησης αλλά επικοινωνούνται μόνο για να επιστραφούν στο αντικείμενο της επιλεγμένης κίνησης στην περίπτωση που αυτή η κατάσταση επιλεγεί από τον μηχανισμό μάθησης. Οι τιμές αυτές είναι : Αριθμός πιονιού, συντεταγμένη X, συντεταγμένη Y, συντεταγμένη Z (για μελλοντική χρήση σε παιχνίδι που θα χρειάζεται και την διάσταση του ύψους).</p>
AgentAction.java	<p>Κλάση που χρησιμοποιείται για την επικοινωνία της επιλεγμένης κίνησης από τον πράκτορα ενισχυτικής μάθησης. Οι τιμές που επικοινωνούνται είναι οι ακόλουθες: Αριθμός πιονιού, συντεταγμένη X, συντεταγμένη Y, συντεταγμένη Z, τιμή αξιολόγησης της κίνησης, σημαία εάν η κίνηση είναι προϊόν εκμετάλλευσης γνώσης ή αναζήτησης.</p>

Στο ακόλουθο διάγραμμα κλάσεων παρουσιάζονται οι κλάσεις του πακέτου βοηθητικών κλάσεων [Εικόνα 5.1].



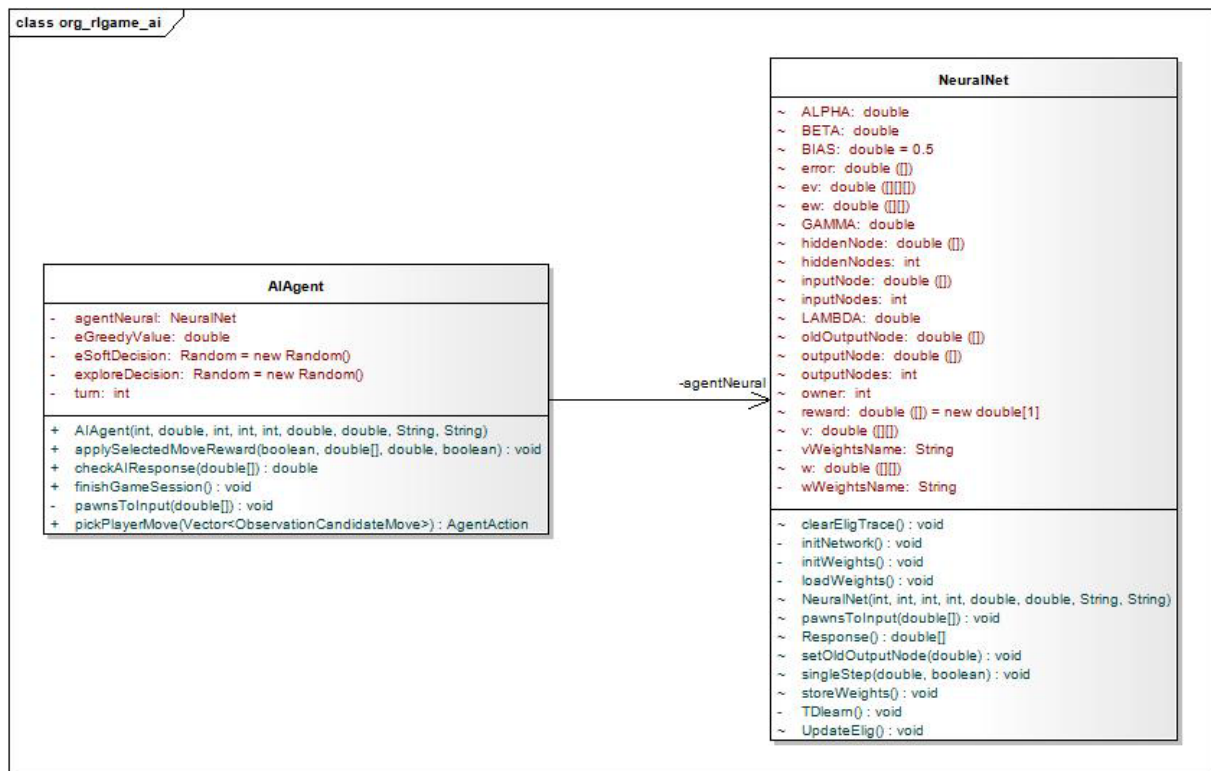
Εικόνα 5.1 : Διάγραμμα κλάσεων του πακέτου βοηθητικών κλάσεων

### 5.2.2 Πακέτο Μάθησης

Στο πακέτο του μηχανισμού μάθησης έχουν ορισθεί οι δύο κλάσεις που χρησιμοποιούνται για την υλοποίηση των αλγορίθμων της ενισχυτικής μάθησης και του Νευρωνικού δικτύου.

Κλάση	Περιγραφή
AIAgent.java	Η κλάση του πράκτορα ενισχυτικής μάθησης. Η κλάση έχει ορισμένες δημόσιες μεθόδους που χρησιμοποιούνται από τα αντικείμενα των κλάσεων των παικτών ενισχυτικής μάθησης και Minimax. Κατά την δημιουργία κάποιου αντικειμένου αυτής της κλάσης ο καλών περνάει ορίσματα με τις ρυθμίσεις του πράκτορα και του νευρωνικού δικτύου. Ο πράκτορας δημιουργεί το αντικείμενο του νευρωνικού δικτύου και κατόπιν χρησιμοποιεί μεθόδους του για να εξάγει την αξιολόγηση μιας κίνησης, για να εισάγει μια κατάσταση και την τρέχουσα αξιολόγηση της κ.λπ.
NeuralNet.java	Κλάση που χρησιμοποιείται για την υλοποίηση των δομών και των αλγορίθμων του νευρωνικού δικτύου. Η κλάση αυτή έχει ορισθεί με εμβέλεια πακέτου και μπορεί να χρησιμοποιηθεί μόνο από κλάσεις του πακέτου org.rlgame.ai, συνεπώς χρησιμοποιείται μόνο από την κλάση του πράκτορα ενισχυτικής μάθησης. Η επιλογή αυτή έγινε για να γίνει σαφές ότι η NeuralNet είναι μία εσωτερική κλάση και ο μηχανισμός μπορεί να χρησιμοποιηθεί μόνο μέσω των δημόσιων κλήσεων της κλάσης του πράκτορα ενισχυτικής μάθησης.

Στο ακόλουθο διάγραμμα κλάσεων παρουσιάζονται οι κλάσεις του πακέτου μάθησης.



Εικόνα 5.2 : Διάγραμμα κλάσεων του πακέτου μάθησης

# Κεφάλαιο 6

## Σχεδίαση και Υλοποίηση του mRLGame

Σε αυτό το κεφάλαιο περιγράφεται η υλοποίηση της εφαρμογής mRLGame δηλαδή της έκδοσης του RLGame για κινητές πλατφόρμες. Γίνεται αναλυτική παρουσίαση των τεχνολογιών που χρειάστηκε να κατακτηθούν, των βημάτων που ακολουθήθηκαν μέχρι την τελική υλοποίηση, των προβλημάτων που αντιμετωπίστηκαν καθώς και των λύσεων που τελικά εφαρμόστηκαν.

### 6.1 Προετοιμασία

Για την υλοποίηση της εφαρμογής χρειάστηκε μεγάλος χρόνος προετοιμασίας αρχικά για να γίνουν κατανοητές οι τεχνολογίες που χρειάζονταν και στην συνέχεια για την μελέτη των υπαρχόντων υλοποιήσεων του RLGame και ειδικά του κώδικα της υλοποίησης του RL-Lib [01] που περιγράφεται στο Κεφάλαιο 5.

### 6.1.1 Τεχνολογίες που χρειάστηκαν

Η αρχική υλοποίηση του RLGame αλλά και αυτή του RL-Lib είναι γραμμένες σε γλώσσα προγραμματισμού java οπότε αρχικά χρειάστηκε να γίνει εξοικείωση του γράφοντα με την Java και στη συνέχεια να γίνει μελέτη και ανάλυση της δομής των αντικειμένων στο υπάρχοντα κώδικα των προηγούμενων υλοποιήσεων.

Για την ανάπτυξη εφαρμογών σε Android η Google έχει αναπτύξει και διανέμει δωρεάν ένα πρόσθετο (plug-in) για το Eclipse το οποίο ονομάζεται Android Development Tools (ADT) και που υπήρξε και το εργαλείο για την ανάπτυξη της εφαρμογής. Ο χρόνος που χρειάστηκε για την εκμάθηση του Android παράλληλα με την Java ξεπέρασε τους 4 μήνες [59, 60] καθώς το Android διαθέτει ένα αρκετά περίπλοκο API και κάθε εργασία μέσα σε αυτό γίνεται με κλήσεις σε αντίστοιχες κλάσεις του API [34 - 38]. Επίσης χρειάστηκε και η εκμάθηση της γλώσσας XML καθώς αυτή χρησιμοποιεί το Android για να σχεδιάσει την δομή ενός παραθύρου – δραστηριότητας (Activity) της εφαρμογής.

Το επόμενο σκαλί για την ανάπτυξη της εφαρμογής ήταν η χρήση γραφικών μέσα σε μία Android εφαρμογή. Μία εφαρμογή που χρησιμοποιεί τα γνώριμα χειριστήρια που ξέρουμε σε ένα κινητό Android όπως κουμπιά, μενού κλπ μπορεί σχετικά εύκολα να σχεδιαστεί με χρήση XML, όμως η σχεδίαση γραφικών είναι μία τελείως διαφορετική υπόθεση η οποία από μόνη της αποτελεί ένα πολυδιάστατο αντικείμενο. Μετά από αρκετή έρευνα χρησιμοποιήθηκε η standard βιβλιοθήκη canvas που έχει το Android της οποίας χρειάστηκε η εκμάθηση ώστε να γίνει δυνατή η σχεδίαση των γραφικών του παιχνιδιού.

### 6.1.2 Δοκιμές με υπάρχοντα κώδικα και η επιλογή του RLLib

Για την κατανόηση της εσωτερικής λειτουργίας του παιχνιδιού μελετήθηκαν όλες οι προηγούμενες υλοποιήσεις του παιχνιδιού. Αρχικά πάρθηκε η απόφαση της μελέτης και χρήσης, ως αφετηρίας, της δομής που χρησιμοποιήθηκε σε μία αρχική υλοποίηση της εφαρμογής RLGame [15] κυρίως γιατί υλοποιούσε παιχνίδι μεταξύ υπολογιστή και ανθρώπου όπως κάνει και το mRLGame.

Στη συνέχεια έγινε κατανοητό, όσο προχωρούσε παράλληλα και η εκμάθηση του Android, ότι η δομή που χρησιμοποιούσε στην εφαρμογή για windows δεν ήταν δυνατό να εφαρμοστεί σε Android λόγω της διαφορετικής φιλοσοφίας του που οφείλεται στον προσανατολισμό του προς κινητές πλατφόρμες. Μπορεί για παράδειγμα μία δραστηριότητα (Activity) στο Android να μοιάζει με την χρησιμότητα που έχει ένα παράθυρο στα windows όμως δεν έχει καμία άλλη ομοιότητα, αφού για



παράδειγμα στο Android μπορεί να υπάρχουν πολλές ανοιχτές δραστηριότητες αλλά να εκτελείται μόνο μία, ενώ οι υπόλοιπες να βρίσκονται σε κατάσταση παύσης. Επιπλέον η εργασία με γραφικά είναι επίσης τελείως διαφορετική καθώς στο android για την σχεδίαση γραφικών δύο διαστάσεων υπάρχει η κλάση canvas που θα περιγραφεί παρακάτω η οποία έχει διαφορετικό τρόπο χειρισμού των γραφικών από ότι οι αντίστοιχες κλάσεις της Java για windows.

Οι παραπάνω δυσκολίες σε συνδυασμό με την μελέτη της υλοποίησης άλλης διπλωματικής εργασίας [01] που υλοποιούσε ξεχωριστά την λογική παιχνιδιού και την τεχνητή νοημοσύνη χρησιμοποιώντας τελικά πιο ξεκάθαρη δομή κώδικα, με οδήγησαν στην υιοθέτηση του RL-Lib ως βάσης για το νέο παιχνίδι. Ο υπάρχων κώδικας δεν υποστήριζε το παιχνίδι μεταξύ ανθρώπου και υπολογιστή αλλά μόνο υπολογιστή και υπολογιστή. Παρ' όλα αυτά, η σχεδίαση του ως βιβλιοθήκης τον κάνει ιδανικό για την υλοποίηση του mRLGame αλλά και κάθε άλλου παιχνιδιού που χρησιμοποιεί Reinforcement Learning.

## 6.2 Σχεδίαση και Υλοποίηση

### 6.2.1 Μετατροπή του αρχικού RLLib σε ένα Android Project

Αρχικά ο κώδικας της υλοποίησης του RLLib που ήταν στην μορφή ενός Eclipse Java Project μετατράπηκε σε ένα Eclipse Android Project. Για να γίνει αυτό δημιουργήθηκε ένα νέο Android Project και κάθε κλάση της αρχικής υλοποίησης αντιγράφηκε μέσα στο νέο Project.

Για να μπορεί να εκτελεστεί μία εφαρμογή που κάνει έξοδο στην γραμμή εντολών μέσα στο Android και να μπορούμε να δούμε αποτελέσματα έγινε χρήση μίας λειτουργίας του Android που ονομάζεται LogCat το οποίο στην πραγματικότητα είναι ένα Log μέσα από το οποίο βλέπουμε μηνύματα της εφαρμογής και της Java VM σε πραγματικό χρόνο. Κάθε εντολή εξόδου προς της οθόνη μετατράπηκε σε εντολή εξόδου προς το LogCat.

Επιπλέον χρειάστηκε να γίνουν αλλαγές στην θέση όπου αποθηκεύονται τα αρχεία που δημιουργεί η εφαρμογή και αυτό γιατί στο Android δεν μπορεί κάποια εφαρμογή να γράψει σε όποιο φάκελο θέλει. Εδώ να σημειώσουμε ότι Android είναι μία Linux υλοποίηση και χρησιμοποιεί την ίδια δομή του συστήματος αρχείων με αυτό. Τα αρχεία που δημιουργεί η εφαρμογή δημιουργούνται στην εξωτερική SD Card για λόγους εξοικονόμησης μνήμης.

Το επόμενο βήμα ήταν η εκκίνηση της εφαρμογής, που στο Android γίνεται από μία δραστηριότητα (Activity), με τον τρόπο που είχε σχεδιαστεί να ξεκινάει αρχικά. Χρειάστηκαν επιπλέον κάποιες αλλαγές σε βιβλιοθήκες που κάνει χρήση το παιχνίδι και τελικά η εφαρμογή λειτούργησε φυσικά παίζοντας παιχνίδια μεταξύ υπολογιστή και υπολογιστή όπως είχε σχεδιαστεί με τα αποτελέσματα να είναι ορατά μόνο στο LogCat και στα αρχεία που δημιουργεί η εφαρμογή για κάθε παιχνίδι. Αυτό ήταν ενθαρρυντικό στοιχείο καθώς έδειξε ότι ο κώδικας του RL-Lib είναι συμβατός και μπορεί να εκτελεστεί στο Android χωρίς περισσότερες αλλαγές.

### 6.2.2 Σχεδίαση του GUI & των γραφικών

Η σχεδίαση της κάθε οθόνης γίνεται με χρήση αρχείων XML όπου περιγράφεται το κάθε χειριστήριο ενώ με βάση το όνομα του γίνεται και η σύνδεση του με τον java κώδικα. Η σχεδίαση γραφικών έγινε με την χρήση της βιβλιοθήκης για 2D γραφικά του Android που ονομάζεται canvas.

Η σχεδίαση γραφικών μέσα στον canvas γίνεται με χρήση εντολών που σχεδιάζουν σχήματα όπως γραμμές και κύκλους και στα αρχικά στάδια της υλοποίησης δημιουργήθηκε μία έκδοση του παιχνιδιού όπου ο παίχτης μπορούσε να μετακινεί τα πιόνια με χρήση της αφής χωρίς όμως να χρησιμοποιεί κανόνες ή κάποια λογική παιχνιδιού.

### 6.2.3 Σύνδεση του RLLib με τα Γραφικά

Το τελευταίο στάδιο, το οποίο αποδείχτηκε και το πιο δύσκολο, ήταν η σύνδεση του υποσυστήματος τεχνητής νοημοσύνης δηλαδή του RL-Lib, με το υποσύστημα λογικής παιχνιδιού και με τα γραφικά. Αυτό είχε δύο σημεία δυσκολίας, αρχικά γιατί η υλοποίηση του RL-Lib είχε σχεδιαστεί να λειτουργεί για παιχνίδι μεταξύ υπολογιστή και υπολογιστή και όχι μεταξύ υπολογιστή και ανθρώπου οπότε ο κώδικας για την διαχείριση του ανθρώπου παίχτη έπρεπε να σχεδιαστεί από την αρχή όπως επίσης και η διασύνδεση του γραφικού μέρους του παιχνιδιού με την υπόλοιπη λειτουργία του.

## 6.3 Αρχιτεκτονική

### 6.3.1 Παρουσίαση των Δραστηριοτήτων (Activities)

Κάθε οθόνη που εμφανίζεται στο Android, την οποία θα μπορούσε κάποιος να παρομοιάσει με ένα παράθυρο στα windows, είναι και μία δραστηριότητα (activity). Οι δραστηριότητες που δημιουργήθηκαν για αυτή την εφαρμογή είναι οι παρακάτω:

Δραστηριότητα (Activity)	Περιγραφή
SplashScreenActivity	Εμφανίζει μία αρχική (splash) οθόνη με πληροφορίες σχετικά με το πρόγραμμα η οποία κλείνει μόνη της μετά από 5 δευτερόλεπτα και ξεκινάει την δραστηριότητα MenuActivity.
MenuActivity	Η κεντρική οθόνη της εφαρμογής όπου ο χρήστης μπορεί να επιλέξει το τύπο AI του υπολογιστή παίκτη και να ξεκινήσει το παιχνίδι. Επιπλέον μπορεί να δει τις οδηγίες του παιχνιδιού ή να κλείσει το παιχνίδι.
GameView	Είναι η βασική δραστηριότητα και αυτή που ουσιαστικά παίζει το παιχνίδι.
InstructionActivity	Εμφανίζει τις οδηγίες του παιχνιδιού οι οποίες ουσιαστικά είναι μια html σελίδα που περιέχεται μέσα στο παιχνίδι.

### 6.3.2 Παρουσίαση της δομής του java κώδικα

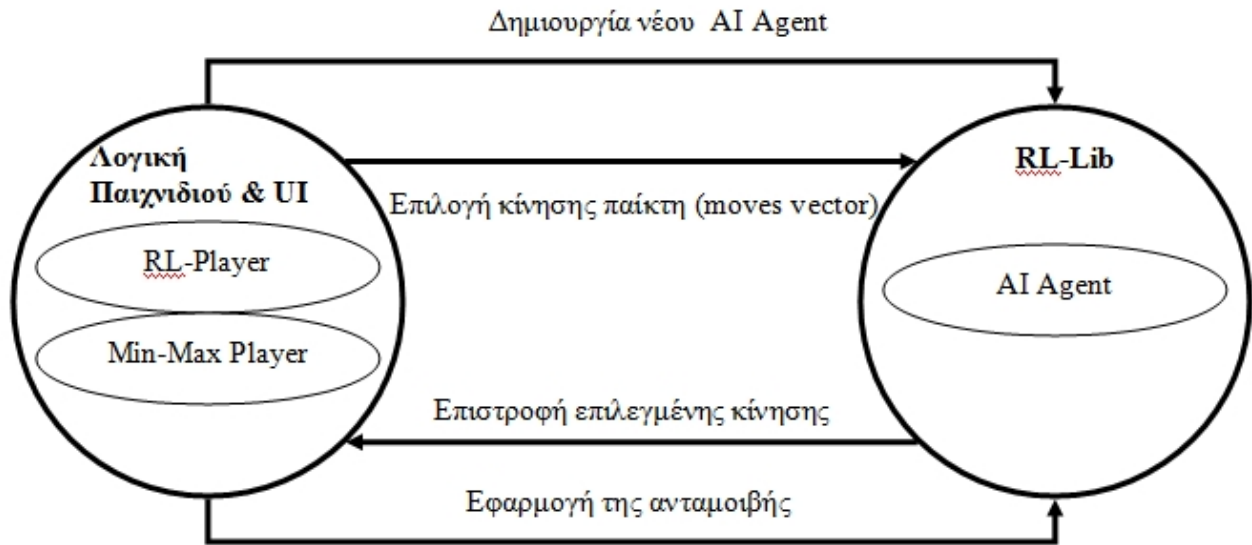
Η εφαρμογή αποτελείται από 4 πακέτα που το καθένα από αυτά περιέχει κλάσεις σχετικές μεταξύ τους. Τα πακέτα που αποτελούν την εφαρμογή είναι τα παρακάτω:

<b>mRLGame</b>	org.ouc.gameplay	Περιέχει τις κλάσεις που σχετίζονται με τους κανόνες και γενικά για την διαχείριση του παιχνιδιού, χωρίς όμως το γραφικό περιβάλλον
	org.ouc.mrlgame	Περιέχει τις κλάσεις που διαχειρίζονται ό,τι έχει σχέση με το γραφικό περιβάλλον και την διασύνδεση με το υπόλοιπο παιχνίδι όπως και τις κλάσεις που διαχειρίζονται τις Δραστηριότητες. Επίσης κρατάει την τρέχουσα κατάσταση του παιχνιδιού και αποθηκεύει σε αρχεία ό,τι χρειάζεται. Μπορούμε να πούμε είναι το κέντρο ελέγχου της υλοποίησης.
<b>RL-Lib</b>	org.rlgame.ai	Το πακέτο αυτό περιέχει τις κοινές βοηθητικές κλάσεις που χρησιμοποιούνται στην αμφίδρομη επικοινωνία του υποσυστήματος λογικής παιχνιδιού και του υποσυστήματος μάθησης.
	org.rlgame.common	Το πακέτο αυτό περιέχει τις κλάσεις του υποσυστήματος μάθησης.

Τα δύο τελευταία πακέτα αποτελούν το RL-Lib το οποίο έχει περιγραφεί αναλυτικά σε προηγούμενο κεφάλαιο.

### 6.3.3 Αλληλεπίδραση Παιχνιδιού και RL-Lib

Όπως έχουμε ήδη αναφέρει το παιχνίδι αντιμετωπίζει το RL-Lib ως μία βιβλιοθήκη ενισχυτικής μάθησης (reinforcement learning) από την οποία ζητάει την επόμενη κίνηση, την λαμβάνει και της επιστρέφει την ανταμοιβή γι' αυτή την κίνηση. Το παρακάτω διάγραμμα παρουσιάζει την επικοινωνία μεταξύ των δύο αντικειμένων του παιχνιδιού [Εικόνα 6.4].



Εικόνα 6.4: Αλληλεπίδραση Παιχνιδιού και RL-Lib

Στο τμήμα λογικής παιχνιδιού γίνεται η δημιουργία του AI Agent που υπολογίζει τις κινήσεις του υπολογιστή παίκτη ενώ ο τύπος του AI Agent έχει να κάνει με την επιλογή του χρήστη από το μενού. Στο υποσύστημα της μάθησης ζητείται να γίνει επιλογή της κίνησης του παίκτη υπολογιστή καθώς του δίνεται ως πληροφορία η τρέχουσα κατάσταση του παιχνιδιού. Το υποσύστημα επιστρέφει την κίνηση που υπολόγισε και αφού γίνει η κίνηση λαμβάνει την ανταμοιβή.

### 6.3.4 Περιγραφή των κλάσεων

Οι κλάσεις που περιέχει το πακέτο `org.ouc.gameplay` και σχετίζονται με το `gameplay` είναι οι παρακάτω:

Κλάση	Περιγραφή
GameState.java	Η κλάση που αποτυπώνει την κατάσταση του παιχνιδιού. Η κλάση τηρεί την κατάσταση των πιονιών των 2 αντιπάλων και την κατάσταση της σκακιέρας.
IPlayer.java (Interface)	Πρότυπο (Interface) κλάσης του παίκτη. Σε αυτό το πρότυπο στηρίζονται οι κλάσεις των παικτών που ακολουθούν εκτός του <code>HumanPlayer</code> που δεν μπορούσε να ακολουθήσει το πρότυπο λόγω διαφορετικής φιλοσοφίας σχεδίασης.
MMPlayer.java	Η κλάση του Min-Max παίκτη. Ο αλγόριθμος Minimax υλοποιείται σε

	<p>αυτή την κλάση. Πρέπει να υπογραμμιστεί ότι η υλοποίηση του Minimax αλγόριθμου της εξερεύνησης του δένδρου καταστάσεων αποτελεί μέρος του υποσυστήματος της λογικής του παιχνιδιού. Η κλάση όμως χρησιμοποιεί τις υπηρεσίες του πράκτορα ενισχυτικής μάθησης για την αξιολόγηση των καταστάσεων του παιχνιδιού. Κάθε Minimax παίκτης δημιουργεί ένα αντικείμενο τύπου πράκτορα και χρησιμοποιεί τη μέθοδο checkAIResponse με όρισμα τον δυαδικό πίνακα είσοδο του νευρωνικού για να παίρνει την αξιολόγηση της τρέχουσας κατάστασης. Για κάθε επιλεγμένη κίνηση ο Minimax παίκτης καλεί την μέθοδο applySelectedMoveReward όπου επικοινωνεί στον πράκτορα την ανταμοιβή του περιβάλλοντος για την επιλεγμένη κίνηση.</p>
Pawn.java	<p>Δημιουργία – διαχείριση πιονιών. Έχει ενσωματωμένη λογική/κανόνων κινήσεων.</p>
HumanPlayer.java	<p>Η κλάση που υλοποιεί τον άνθρωπο παίκτη. Αυτός ο τύπος παίκτη δεν υπήρχε στην υλοποίηση που χρησιμοποιήθηκε ως βάση αυτής τη υλοποίησης.</p>
RandomPlayer.java	<p>Η κλάση του παίκτη τυχαίας επιλογής. Η κλάση ακολουθεί/υλοποιεί το πρότυπο IPlayer.java. Ο τύπος του παίκτη προστέθηκε στην παρούσα υλοποίηση.</p>
RLPlayer.java	<p>Η κλάση του παίκτη Ενισχυτικής Μάθησης. Η κλάση ακολουθεί/υλοποιεί το πρότυπο IPlayer.java. Η κλάση χρησιμοποιεί τις υπηρεσίες του πράκτορα ενισχυτικής μάθησης. Κάθε RL παίκτης δημιουργεί ένα αντικείμενο τύπου πράκτορα. Ο RL παίκτης καλεί την μέθοδο επιλογής κίνησης του πράκτορα περνώντας σαν όρισμα μία λίστα με αντικείμενα τύπου ObservationCandidateMove. Τα αντικείμενα της λίστας επικοινωνούν το σύνολο των επιτρεπόμενων κινήσεων που μπορεί να κάνει ο παίκτης (ο RL παίκτης χρησιμοποιεί μεθόδους της κλάσης κατάστασης του παιχνιδιού για να πάρει τις διαθέσιμες κινήσεις). Μάλιστα για κάθε πιθανή κίνηση ο μηχανισμός επικοινωνεί τόσο τον δυαδικό πίνακα είσοδο του νευρωνικού όσο και την ανταμοιβή του περιβάλλοντος. Ο RL παίκτης λαμβάνει την επιλεγμένη κίνηση από τον πράκτορα και αφού ενημερώσει την κατάσταση του παιχνιδιού, καλεί την μέθοδο</p>

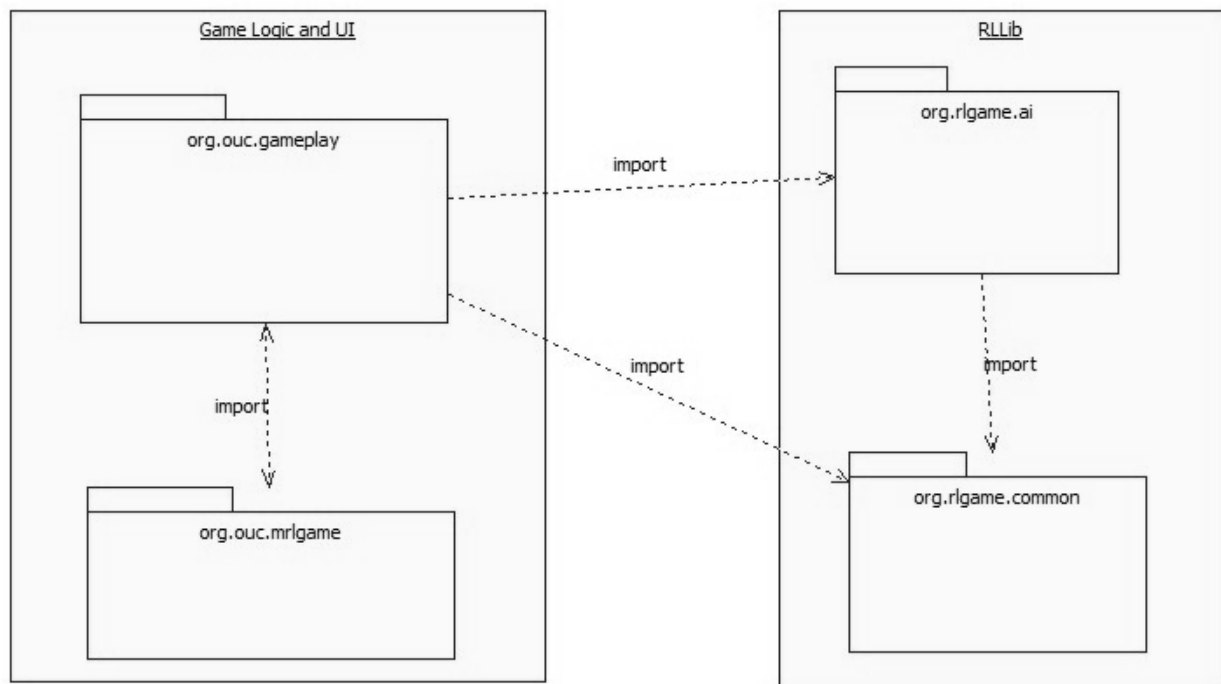
	applySelectedMoveReward όπου επικοινωνεί στον πράκτορα την ανταμοιβή του περιβάλλοντος για την επιλεγμένη κίνηση.
Square.java	Η κλάση που αναπαριστά το τετράγωνο της σκακιέρας

Οι κλάσεις που περιέχει το πακέτο org.ouc.mrlgame και σχετίζονται με βασικά θέματα του παιχνιδιού όπως ρυθμίσεις και θέσεις αρχείων, τις δραστηριότητες και την κατάσταση του παιχνιδιού, είναι οι παρακάτω:

Κλάση	Περιγραφή
History.java	Κλάση επιφορτισμένη με την καταγραφή των εξαγόμενων αρχείων αποτελεσμάτων των παιχνιδιών και των κινήσεων των παικτών.
Settings.java	Κλάση στην οποία ορίζονται οι καθολικές στατικές μεταβλητές οι οποίες καθορίζουν κάποιες παραμέτρους της εφαρμογής όπως ποιος θα είναι ο προκαθορισμένος τύπος παίκτη υπολογιστή.
GameStateUI.java	Η κλάση αυτή αποτυπώνει την κατάσταση του παιχνιδιού σε μορφή που εξυπηρετεί την σχεδίαση της κατάστασης στη σκακιέρα. Η κλάση αυτή ενημερώνεται από την κλάση GameState.java.
GameView.java	Η κλάση περιέχει της λειτουργίες της δραστηριότητας GameView η οποία είναι αυτή που υλοποιεί τα γραφικά και ελέγχει το παιχνίδι αφού διαχειρίζεται της κινήσεις του παίκτη ανθρώπου στην οθόνη αφής.
InstructionActivity.java	Η κλάση περιέχει της λειτουργίες της Activity InstructionActivity και εμφανίζει της οδηγίες χρήσεως.
MenuActivity.java	Η κλάση περιέχει της λειτουργίες της Activity MenuActivity και εμφανίζει το μενού επιλογών.
SplashScreenActivity.java	Η κλάση περιέχει της λειτουργίες της Activity SplashScreenActivity και εμφανίζει την εισαγωγική οθόνη.

### 6.3.5 Διαγράμματα κλάσεων

Το παιχνίδι αποτελείται από δύο διακριτά τμήματα: από την μια μεριά είναι το τμήμα της λογικής παιχνιδιού και διαχείρισης των γραφικών και από την άλλη το τμήμα τεχνητής νοημοσύνης που ουσιαστικά είναι το RL-Lib. Το τμήμα λογικής παιχνιδιού και διαχείρισης αποτελείται από τα πακέτα `org.ouc.gameplay` και το `org.ouc.mrlgame` ενώ το RL-Lib από τα `org.rlgame.ai` και `org.rlgame.common`. Η διασύνδεση των δύο τμημάτων φαίνεται παρακάτω [Εικόνα 6.1].



Εικόνα 6.1: Διασύνδεση RL-Lib και λογικής παιχνιδιού

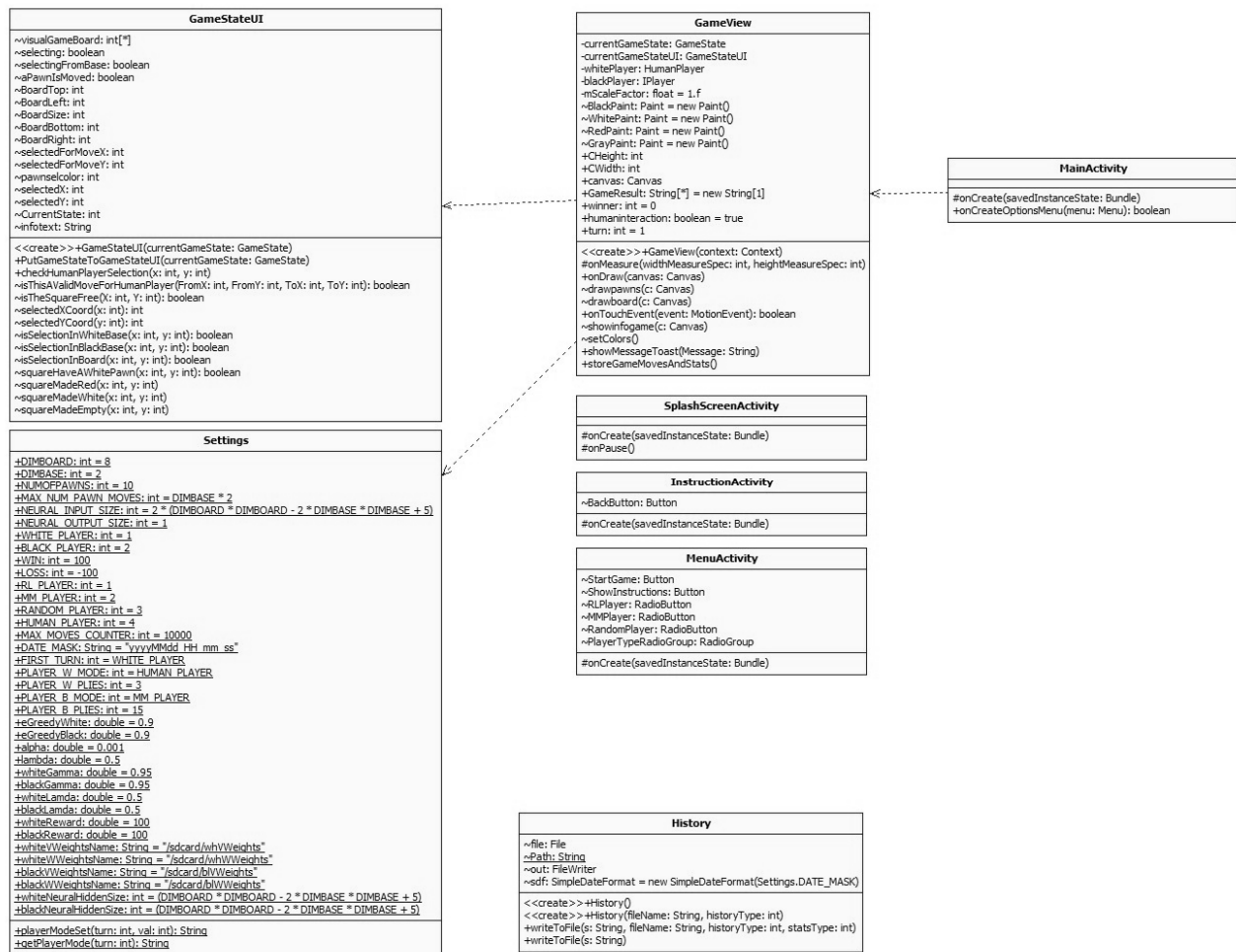


Το διάγραμμα κλάσεων του java πακέτου org.ouc.gameplay είναι το παρακάτω [Εικόνα 6.2].



Εικόνα 6.2: Διάγραμμα κλάσεων του java πακέτου org.ouc.gameplay

Το διάγραμμα κλάσεων του java πακέτου org.ouc.mrlgame είναι το παρακάτω [Εικόνα 6.3].



Εικόνα 6.3: Διάγραμμα κλάσεων του java πακέτου org.ouc.mrlgame

### 6.3.6 Τα Υπόλοιπα Αντικείμενα της Υλοποίησης

Στην δημιουργία του project χρησιμοποιήθηκαν και άλλα αντικείμενα, όπως κάποια γραφικά ενώ επιπλέον κάθε δραστηριότητα (Activity) του android μπορεί να συνοδεύεται από κάποιο xml αρχείο που ουσιαστικά περιέχει τον τρόπο που θα διαταχθούν τα αντικείμενα, όπως κουμπιά και τετράγωνα επιλογής, στην οθόνη. Επιπλέον κάθε εφαρμογή στο Android έχει ένα αρχείο manifest.xml όπου περιγράφονται όλες οι δραστηριότητες που περιέχει η εφαρμογή, από ποια δραστηριότητα ξεκινάει η εκτέλεση και άλλες πληροφορίες, όπως ποιά είναι τα δικαιώματα που έχει η εφαρμογή επάνω στην συσκευή στην οποία εκτελείται.

Τα διάφορα βοηθητικά αρχεία της εφαρμογής είναι:

Αρχείο	Περιγραφή
mrlgame.png	Το εικονίδιο της εφαρμογής
splash.png	Η εικόνα με το λογότυπο του πανεπιστημίου που εμφανίζεται στην αρχή της εφαρμογής.
splashscreen.xml	Το xml αρχείο της activity splashscreen
menu.xml	Το xml αρχείο της activity menu
instructions.xml	Το xml αρχείο της activity instructions
activity_main.xml	Το xml αρχείο της activity main
Gameinstr.html	Html αρχείο που περιέχει τις οδηγίες του παιχνιδιού και εμφανίζεται στην instructions activity

Ο κώδικας του AndroidManifest.xml παρουσιάζεται παρακάτω:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="org.ouc.mrlgamewithrllib"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk
        android:minSdkVersion="8"
        android:targetSdkVersion="17" />
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>

    <application
        android:allowBackup="true"
        android:icon="@drawable/mrlgame"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name="org.ouc.mrlgame.SplashScreenActivity"
            android:screenOrientation="landscape"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity
            android:name="org.ouc.mrlgame.MenuActivity"
            android:screenOrientation="landscape"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="org.ouc.mrlgamewithrllib.MENUACTIVITY" />
                <category android:name="android.intent.category.DEFAULT" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

```

</activity>
<activity
    android:name="org.ouc.mrlgame.MainActivity"
    android:screenOrientation="landscape"
    android:label="@string/app_name" >
    <intent-filter>
        <action android:name="org.ouc.mrlgamewithrllib.MAINACTIVITY" />
        <category android:name="android.intent.category.DEFAULT" />
    </intent-filter>
</activity>
<activity
    android:name="org.ouc.mrlgame.InstructionActivity"
    android:label="@string/app_name"
    android:screenOrientation="landscape" >
    <intent-filter>
        <action android:name="org.ouc.mrlgamewithrllib.INSTRUCTIONACTIVITY" />
        <category android:name="android.intent.category.DEFAULT" />
    </intent-filter>
</activity>
</application>
</manifest>

```

## 6.4 Βασικά Σημεία Σχεδίασης και Δυσκολίες Υλοποίησης

Σε αυτό το τμήμα περιγράφονται τα βασικά σημεία της υλοποίησης, τα πιο σημαντικά προβλήματα και ο τρόπος που αντιμετωπίστηκαν.

### 6.4.1 Δημιουργία της κλάσης Human Player

Όπως έχει αναφερθεί σε προηγούμενο σημείο της εργασίας το παιχνίδι πάνω στο οποίο βασίστηκε η υλοποίηση αυτής της έκδοσης [01] δεν παρείχε την δυνατότητα παιχνιδιού μεταξύ υπολογιστή και ανθρώπου αλλά μόνο μεταξύ υπολογιστή και υπολογιστή. Κατά συνέπεια η δημιουργία μίας κλάσης που θα υλοποιούσε τις λειτουργίες του ανθρώπου – παίκτη ήταν απαραίτητη.

Η δομή της κλάσης είναι η παρακάτω [Εικόνα 6.5]

HumanPlayer
<pre>-id: int -playerType: int -turn: int -movesLog: StringBuffer +lastmove: String</pre>
<pre>&lt;&lt;create&gt;&gt; +HumanPlayer(ident: int) +getId(): int +getPlayerType(): int +getMovesLog(): StringBuffer +pickMove(passedGameState: GameState) +pickMove(passedGameState: GameState, FromX: int, FromY: int, ToX: int, ToY: int) -playSelectedMove(chosenPawn: Pawn, targetSquare: Square, passedGameState: GameState) +finishGameSession() +addMoveLog(s: String) ~getWhitePawnFromCoords(passedGameState: GameState, FromX: int, FromY: int): Pawn ~isThereAWhitePawn(passedGameState: GameState, X: int, Y: int): boolean</pre>

Εικόνα 6.5: Η κλάση HumanPlayer που υλοποιεί τον άνθρωπο - παίκτη

Οι βασικές λειτουργίες αυτής της κλάσης είναι η λήψη από το γραφικό περιβάλλον της κίνησης του ανθρώπου παίκτη, από κάποιο τετράγωνο σε κάποιο άλλο τετράγωνο επάνω στη σκακιέρα, η μετατροπή των συντεταγμένων αυτών σε στοιχεία που μπορεί η λογική του παιχνιδιού να καταλάβει (δηλαδή, χρησιμοποιώντας τις κλάσεις pawn και square) και η εισαγωγή της κίνησης εντός της κατάστασης του παιχνιδιού (GameState).

#### 6.4.2 Η Δημιουργία της κλάσης GameStateUI

Τις λειτουργίες που αφορούν την τρέχουσα κατάσταση του παιχνιδιού τις υλοποιεί η κλάση GameState. Για να γίνει δυνατόν να γεφυρωθούν το GameState που περιέχει την τρέχουσα κατάσταση του παιχνιδιού με χρήση των αντικειμένων pawn & square με το γραφικό περιβάλλον που καταλαβαίνει τις θέσεις των πιονιών σε συντεταγμένες X και Y επάνω στη σκακιέρα, χρειάστηκε η δημιουργία μίας νέας κλάσης που θα περιέχει όλες τις λειτουργίες που αφορούν την τρέχουσα κατάσταση του γραφικού περιβάλλοντος και θα φροντίζει αυτά τα δύο να συμβαδίζουν.

Η δομή της κλάσης GameStateUI είναι η παρακάτω [Εικόνα 6.6].

GameStateUI
<pre>~visualGameBoard: int[*] ~selecting: boolean ~selectingFromBase: boolean ~aPawnIsMoved: boolean ~BoardTop: int ~BoardLeft: int ~BoardSize: int ~BoardBottom: int ~BoardRight: int ~selectedForMoveX: int ~selectedForMoveY: int ~pawnselcolor: int ~selectedX: int ~selectedY: int ~CurrentState: int ~infotext: String</pre>
<pre>&lt;&lt;create&gt;&gt; +GameStateUI(currentGameState: GameState) +PutGameStateToGameStateUI(currentGameState: GameState) +checkHumanPlayerSelection(x: int, y: int) ~isThisAValidMoveForHumanPlayer(FromX: int, FromY: int, ToX: int, ToY: int): boolean ~isTheSquareFree(X: int, Y: int): boolean ~selectedXCoord(x: int): int ~selectedYCoord(y: int): int ~isSelectionInWhiteBase(x: int, y: int): boolean ~isSelectionInBlackBase(x: int, y: int): boolean ~isSelectionInBoard(x: int, y: int): boolean ~squareHaveAWhitePawn(x: int, y: int): boolean ~squareMadeRed(x: int, y: int) ~squareMadeWhite(x: int, y: int) ~squareMadeEmpty(x: int, y: int)</pre>

Εικόνα 6.6: Η κλάση GameStateUI που αναλαμβάνει την διασύνδεση γραφικής απεικόνισης και λογικής παιχνιδιού.

Οι λειτουργίες της κλάσης είναι η αντιγραφή της κατάστασης του παιχνιδιού σε μορφή κατάλληλη που να μπορεί να αποτυπωθεί στη σκακιέρα, να ελέγξει αν μία κίνηση του ανθρώπου παίκτη είναι σωστή, και άλλες που αφορούν το γραφικό περιβάλλον.

### 6.4.3 Καταστάσεις του γραφικού περιβάλλοντος

Ένα πρόβλημα που δημιουργήθηκε κατά την υλοποίηση της εφαρμογής και έχει να κάνει με την φιλοσοφία σχεδίασης γραφικών στο Android είναι ότι σε κάθε κίνηση του παίκτη τα γραφικά σχεδιάζονται από την αρχή χωρίς να υπάρχει η δυνατότητα να σταματήσει η διαδικασία σχεδίασης μέχρι να ολοκληρωθεί πλήρως. Μερικές φορές αυτό δεν είναι επιθυμητό, όπως όταν ο παίχτης έχει επιλέξει ένα πιόνι στην σκακιέρα για μετακίνηση αλλά δεν έχει ακόμη δηλώσει σε πιο τετράγωνο θέλει να μετακινηθεί. Για υπερνικηθεί αυτό το πρόβλημα χρειάστηκε εντός της κλάσης GameView (που

ελέγχει το γραφικό περιβάλλον), στην διαδικασία OnDraw (που ενεργοποιείται κάθε φορά που σχεδιάζονται τα γραφικά), να υλοποιηθούν καταστάσεις με βάση τις οποίες το γραφικό περιβάλλον ενημερώνεται από το υποσύστημα κατάστασης του παιχνιδιού (GameState -> GameStaUI) ή όχι, καταχωρείται η κίνηση του παίκτη ή απλώς δεν γίνεται καμία κίνηση στο παιχνίδι γιατί ο παίκτης ακύρωσε την κίνηση του.

Κατάσταση (state)	Περιγραφή
1	Ο παίκτης άνθρωπος έχει επιλέξει ένα πιόνι στην σκακιέρα (έχει γίνει κόκκινο) αλλά δεν έχει ολοκληρώσει την κίνηση του.
2	Ο παίκτης άνθρωπος έχει ολοκληρώσει την κίνηση του
3	Ενδιάμεση κατάσταση η οποία σχεδιάζει ξανά τα γραφικά για να ενημερωθεί η σκακιέρα από την κατάσταση του παιχνιδιού
4	Είναι σειρά του παίκτη – υπολογιστή να επιλέξει κίνηση
5	Το παιχνίδι έχει λήξει και δεν δέχεται καμία κίνηση στο ταμπλό

#### 6.4.4 Αντιμετώπιση θεμάτων ανάλυσης οθόνης

Το Android είναι ένα λειτουργικό σύστημα που εκτελείται σε διαφόρων τύπων συσκευές με διαφορετικής ανάλυσης οθόνες που κυμαίνονται από 320X240 pixels μέχρι και full HD 1920X1024 pixels. Η ύπαρξη θεμάτων σχετικά με το σχεδιασμό γραφικών σε ένα τέτοιο περιβάλλον είναι φυσικό να υπάρχουν.

Το παιχνίδι για να λειτουργήσει και να εμφανίζονται όλα τα στοιχεία του πρέπει να έχει στη διάθεση του μία οθόνη με ανάλυση 480X480 pixels. Για την βέλτιστη λειτουργία σε tablets επιλέχθηκε η εκτέλεση του παιχνιδιού να γίνεται σε οριζόντιο προσανατολισμό (landscape view) χωρίς να έχει την δυνατότητα ο χρήστης να αλλάζει τον προσανατολισμό κάτι που κάνουν πολλά παιχνίδια στο Android. Η ανάλυση του παιχνιδιού στον οριζόντιο άξονα καλύπτεται από σχεδόν όλες τις συσκευές που κυκλοφορούν στο εμπόριο αλλά με τον κατακόρυφο άξονα σε κάποιες παλιές συσκευές υπήρχε πρόβλημα καθώς είχαν μικρότερη ανάλυση. Αυτό υπερνικήθηκε με την χρήση της λειτουργίας του scroll view όπου η οθόνη δεν εμφανίζεται ολόκληρη αλλά ο χρήστης μπορεί με το δάκτυλο του να μετακινηθεί μέσα στην οθόνη ώστε να εμφανίσει όλα τα περιεχόμενά της τμηματικά.

Επίσης η εφαρμογή κατά την εκκίνηση ελέγχει την ανάλυση της οθόνης και αν κάποια διάσταση της είναι μικρότερη από 480 pixels εμφανίζει σχετικό μήνυμα ότι η βέλτιστη ανάλυση για αυτή την εφαρμογή είναι 480X480 pixels ή μεγαλύτερη.

#### **6.4.5 Θέση Αποθήκευσης Αρχείων**

Το Android χρησιμοποιεί την ίδια οργάνωση αρχείων με αυτή του Linux και καμία από τις εφαρμογές (εκτός του λειτουργικού) δεν εκτελείται με δικαιώματα διαχειριστή (root). Η εφαρμογή χρειάζεται να αποθηκεύσει και να διαβάσει αρχεία από την συσκευή. Επιλέχθηκε η αποθήκευση των αρχείων να γίνεται στον αρχικό κατάλογο της εσωτερικής μνήμης της συσκευής. Να σημειωθεί ότι δεν θα ήταν δυνατό να γίνει εγγραφή των αρχείων αυτών στον φάκελο που είναι εγκατεστημένη η εφαρμογή καθώς δεν είναι δυνατό να γράψει μία εφαρμογή εκεί για λόγους ασφαλείας που θέτει το Android. Η εγγραφή αρχείων στην εσωτερική μνήμη της συσκευής είναι και το μοναδικό δικαίωμα που ζητάει η εφαρμογή επάνω στην συσκευή.



# Κεφάλαιο 7

## Έλεγχος καλής Λειτουργίας και Προώθηση σε Τελικούς Χρήστες

### 7.1 Δοκιμές σε φυσικές συσκευές

Η δοκιμή σε πραγματικές συσκευές εκτός από προσομοιωτές ήταν απαραίτητη για να δοκιμαστεί η εφαρμογή και να φανεί πως λειτουργεί και πως ανταποκρίνεται σε διαφορετικό υλικό όσον αφορά τις απαιτήσεις της σε μνήμη και υπολογιστική ισχύ, σε διαφορετικές αναλύσεις οθόνης και διαφορετικές εκδόσεις του Android

Οι πραγματικές συσκευές στις οποίες δοκιμάστηκε η εφαρμογή είναι οι παρακάτω:

#### 1. Sony Xperia Tipo ST21i (Smart Phone)

Επεξεργαστής: Single Core 800 MHz

Μνήμη Ram: 512 MB

Οθόνη: 3,2 "

Ανάλυση Οθόνης: 480 X 320 pixels

Έκδοση Android: 4.0.4

Εμπειρία χρήσης: Η συγκεκριμένη συσκευή είναι μία χαμηλού κόστους συσκευή με χαμηλή ανάλυση οθόνης (μικρότερη των προδιαγραφών της εφαρμογής) και σχετικά αργό επεξεργαστή. Για να λειτουργήσει η εφαρμογή γίνεται χρήση της λειτουργίας Scroll View που εμφανίζει το περιεχόμενο της οθόνης τμηματικά. Η εφαρμογή λειτούργησε κανονικά χωρίς να παγώνει ή να αργεί να ανταποκριθεί.

## **2. Huawei U8650 (Smart phone)**

Επεξεργαστής: Single Core 600 MHz

Μνήμη Ram: 256 MB

Οθόνη: 3,5 "

Ανάλυση Οθόνης: 480 X 320 pixels

Έκδοση Android: 3.2.2

Εμπειρία χρήσης: Η συσκευή χρησιμοποιήθηκε ως παράδειγμα παλιάς και αργής συσκευής καθώς διαθέτει πολύ αργό επεξεργαστή και την ελάχιστη μνήμη που μπορεί να υπάρχει σε συσκευή Android. Παράλληλα η έκδοση του Android είναι αρκετά παλιά. Η εφαρμογή λειτούργησε κανονικά χωρίς να παγώνει ή να αργεί να ανταποκριθεί.

## **3. Samsung Galaxy Tab II 7.0" P3100 (Tablet)**

Επεξεργαστής: Dual Core 1 GHz

Μνήμη Ram: 1024 MB

Οθόνη: 7 "

Ανάλυση Οθόνης: 1024 X 600 pixels

Έκδοση Android: 4.2.2

Εμπειρία χρήσης: Η συσκευή χρησιμοποιήθηκε ως παράδειγμα συσκευής tablet και επίσης συσκευής με σχετικά καλά χαρακτηριστικά, επεξεργαστή και μνήμης RAM. Λόγω ανάλυσης της οθόνης η σκακιέρα χωράει ολόκληρη στην οθόνη. Επίσης διαθέτει σχετικά νέα έκδοση του Android

## 7.2 Δοκιμές σε εικονικές συσκευές

Εκτός από τις δοκιμές σε πραγματικές συσκευές δοκιμές έγιναν σε εικονικές συσκευές (Virtual Machines). Οι δοκιμές σε εικονικές συσκευές εξασφαλίζει ότι θα γίνει δυνατή η δοκιμή σε μεγαλύτερο εύρος συσκευών από αυτές που είναι δυνατόν κάποιος να έχει στην κατοχή του και επιπλέον είναι πολύ πιο εύχρηστες κατά την δημιουργία της εφαρμογής καθώς εκτελούνται σε ένα παράθυρο εντός του ίδιου λειτουργικού συστήματος που είναι και το λογισμικό ανάπτυξης της εφαρμογής.

### 7.2.1 Εικονικές συσκευές του AVDM

Ο AVDM (Android Virtual Device Manager) είναι τμήμα του ADT και μπορεί να δημιουργήσει εικονικές συσκευές που καλύπτουν όλο το φάσμα των επεξεργαστών, μνήμης RAM, αναλύσεων οθόνης και εκδόσεων του Android

Το πιο σημαντικό χαρακτηριστικό των AVD (Android Virtual Device) που δημιουργεί ο AVDM είναι ότι μπορεί κάποιος προγραμματιστής να δοκιμάσει τα χαρακτηριστικά των τελευταίων εκδόσεων του Android πριν ακόμη βγούνε συσκευές που το έχουν εγκατεστημένο.

Η εφαρμογή δοκιμάστηκε σε όλες τις εκδόσεις του Android από την έκδοση 2.2, που είναι και η παλιότερη που υποστηρίζει μία AVD μέχρι και την τελευταία 4.4 και λειτούργησε κανονικά χωρίς κανένα πρόβλημα.

### 7.2.2 Εικονική συσκευή με Android X86 σε Virtual Box

Το Android X86 [31, 32] είναι μία μη επίσημη έκδοση του Android που μπορεί να εκτελείται σε υπολογιστές με επεξεργαστές AMD και Intel X86 και όχι σε υπολογιστές βασισμένους σε RISC ARM επεξεργαστές, όπως είναι όλες οι συσκευές που εκτελούν την κανονική έκδοση του.

Για πιο εύκολη διανομή της εφαρμογής το Android X86 εγκαταστάθηκε σε μία εικονική μηχανή (VM) που δημιουργήθηκε με το πρόγραμμα Virtualization της Oracle που ονομάζεται Virtual Box [29]. Η εικονική μηχανή μαζί με εγκατεστημένο το mRLGame μπορούν να βρεθούν στην ιστοσελίδα της εφαρμογής [33]. Περισσότερα σχετικά με αυτό υπάρχουν στο Παράρτημα Γ της παρούσας διπλωματικής.

Η εφαρμογή δοκιμάστηκε εκτενώς και σε αυτό το περιβάλλον και λειτούργησε χωρίς κανένα πρόβλημα και μάλιστα λόγω της μεγαλύτερης υπολογιστικής ισχύος του υπολογιστή που εκτελούσε την εικονική μηχανή του Android X86 εκτελέστηκε με μεγαλύτερη ταχύτητα σε σχέση με τις κανονικές Android συσκευές που δοκιμάστηκε.

## 7.3 Δοκιμή καλής λειτουργίας τεχνητής νοημοσύνης

Για να δοκιμαστεί η σωστή λειτουργία της τεχνητής νοημοσύνης έγιναν σερ από παιχνίδια χρησιμοποιώντας ως αντίπαλο του ανθρώπου τον παίκτη ενισχυτικής μάθησης και τον min-max παίκτη. Για την διευκόλυνση της ανάλυσης των δεδομένων έγινε χρήση μίας λειτουργίας της εφαρμογής που αποθηκεύει κάθε παιχνίδι σε αρχείο απλού κειμένου στην μνήμη της συσκευής.

Συνολικά παίχτηκαν 60 παιχνίδια τα οποία χωρίστηκαν στις παρακάτω κατηγορίες με βάση τον τύπο του παίκτη υπολογιστή και την στρατηγική που εφαρμόστηκε από την μεριά του ανθρώπου παίκτη. Τα 30 πρώτα παιχνίδια έγιναν με τον υπολογιστή να χρησιμοποιεί τον παίκτη ενισχυτικής μάθησης και τα υπόλοιπα 30 με χρήση το παίκτη min-max.

Τα 30 παιχνίδια που αντιστοιχούν σε κάθε τύπο παίκτη χωρίστηκαν ανάλογα με στρατηγική επίθεσης από αριστερά, κέντρο και δεξιά και ανάλογα με πόσο πιόνια βγήκαν από την βάση του λευκού παίκτη. Τα παιχνίδια χωρίστηκαν στις παρακάτω περιπτώσεις:

- Στρατηγική επίθεσης από δεξιά:
  - Με πέντε πιόνια : 5 παιχνίδια
  - Με δέκα πιόνια: 5 παιχνίδια
- Στρατηγική επίθεσης από αριστερά:
  - Με πέντε πιόνια: 5 παιχνίδια
  - Με δέκα πιόνια : 5 παιχνίδια
- Στρατηγική επίθεσης από το κέντρο:
  - Με πέντε πιόνια: 5 παιχνίδια
  - Με δέκα πιόνια : 5 παιχνίδια

Τα αποτελέσματα παρατίθενται στον ακόλουθο πίνακα όπου για κάθε παιχνίδι δίνεται το πλήθος των κινήσεων που έγιναν και ο νικητής.

Πιόνια	Στρατηγική	Τύπος Υπολογιστή Παίκτη	Παχνίδι 1	Παχνίδι 2	Παχνίδι 3	Παχνίδι 4	Παχνίδι 5	Νίκες Λευκού	Νίκες Μαύρου
5	Από δεξιά	RL-Player	33	48	39	36	35	5	0
5	Από αριστερά	RL-Player	47	52	59	25	44	5	0
5	Από το κέντρο	RL-Player	54	48	50	51	46	5	0
10	Από δεξιά	RL-Player	61	55	54	58	70	5	0
10	Από αριστερά	RL-Player	95	81	38	70	38	3	2
10	Από το κέντρο	RL-Player	71	80	80	85	80	4	1
5	Από δεξιά	Min-Max Player	46	43	50	51	47	5	0
5	Από αριστερά	Min-Max Player	47	42	45	37	37	5	0
5	Από το κέντρο	Min-Max Player	45	42	39	41	34	5	0
10	Από δεξιά	Min-Max Player	86	86	99	82	64	5	0
10	Από αριστερά	Min-Max Player	75	75	70	85	80	5	0
10	Από το κέντρο	Min-Max Player	81	87	94	79	86	4	1

Παρατηρούμε ότι το παιχνίδι είχε διαφορετική αντίδραση σε κάθε συνδυασμό αριθμού πιονιών και στρατηγικής γεγονός που δείχνει πως ο μηχανισμός τεχνητής νοημοσύνης τουλάχιστον αντιδρά με μη

μόνιμο τρόπο. Επιπλέον παρατηρούμε ότι το παιχνίδι λειτουργεί καλύτερα όταν ο λευκός παίκτης (άνθρωπος) αποφασίζει να αναπτύξει και τα 10 πιόνια του στη σκακιέρα αφού με αυτή την τακτική ο υπολογιστής κατάφερε να κερδίσει 4 παιχνίδια σε αντίθεση με όταν ο λευκός παίκτης ανέπτυξε μόνο 5, αφού σε αυτή την περίπτωση δεν καταφέρει να αποσπάσει καμία νίκη.

Οφείλουμε να παρατηρήσουμε ότι ο μαύρος παίκτης (υπολογιστής) δεν καταφέρνει να είναι ανταγωνιστικός απέναντι στο λευκό παίκτη (άνθρωπος) αφού οι νίκες του είναι μόλις 4 στα 60 παιχνίδια. Αυτό δεν είναι κάτι μη αναμενόμενο καθώς στον μηχανισμό υπολογισμού των κινήσεων του μαύρου παίκτη δεν υπάρχει η έννοια της στρατηγικής στην επιλογή των κινήσεων ώστε να επιτίθεται προς την βάση του αντιπάλου ή να αμύνεται της δικής του.

## 7.4 Προώθηση σε Τελικούς Χρήστες

Για την προώθηση της εφαρμογής επιλέχθηκε η δημιουργία ιστοσελίδας [33] ειδικά για την εφαρμογή ενώ δεν χρησιμοποιήθηκε το Android Play Store.

Το Android Play Store υπήρξε η αρχική σκέψη για να μπορεί η εφαρμογή να βρει κοινό για να την δοκιμάσει όμως τελικά μετά από ανάγνωση της σύμβασης της Google σχετικά με τις υποχρεώσεις και τις ευθύνες κάποιου που ανεβάζει μία εφαρμογή στο Play Store ακόμη και αν η εφαρμογή είναι δωρεάν πάρθηκε η απόφαση ότι η μορφή της σύμβασης ξεφεύγει από τον καθαρά ακαδημαϊκό χαρακτήρα της εφαρμογής οπότε και κάνει το Play Store ακατάλληλο για την διανομή της εφαρμογής.

Στην ιστοσελίδα [33] της εφαρμογής μπορεί να βρει κάποιος το παιχνίδι σε μορφή αρχείου .apk που είναι κατάλληλο να κατέβει απευθείας στην Android συσκευή του και σε μορφή εικονικής μηχανής που περιέχει το Android X86 με εγκατεστημένο το mRLGame ώστε να μπορεί κάποιος να δοκιμάσει την εφαρμογή χωρίς να έχει μία συσκευή με Android στην κατοχή του. Περισσότερα υπάρχουν στο Παράρτημα Γ της διπλωματικής.

# Κεφάλαιο 8

## Συμπεράσματα

### 8.1 Απολογισμός

Στα πλαίσια της παρούσης μεταπτυχιακής διατριβής αρχικά μελετήθηκαν οι υπάρχουσες υλοποιήσεις του πρωτότυπου παιχνιδιού RLGame και ειδικά η εφαρμογή RL-Lib που αναπτύχθηκε στα πλαίσια προγενέστερης μεταπτυχιακής διπλωματικής εργασίας [01] η οποία αποτέλεσε και την βάση για την υλοποίηση της παρούσας εφαρμογής. Σε δεύτερη φάση χρειάστηκε να γίνει εκμάθηση της γλώσσας προγραμματισμού Java που χρησιμοποιεί το Android και εμβάθυνση στο προγραμματισμό στο συγκεκριμένο περιβάλλον και πιο ειδικά σε παιχνίδια που χρησιμοποιούν σκακιέρα.

Ο υπάρχων κώδικας πάνω στον οποίο βασίστηκε η εφαρμογή χρειάστηκε να τροποποιηθεί για να μπορεί να εκτελεστεί στην κινητή πλατφόρμα και επίσης προστέθηκε το γραφικό περιβάλλον της σκακιέρας και οι απαραίτητες κλάσεις για την αλληλεπίδραση χρήστη, γραφικού περιβάλλοντος και λογικής παιχνιδιού.

Το παιχνίδι δοκιμάστηκε σε διάφορες πλατφόρμες, φυσικές και εικονικές, τόσο όσον αφορά στην ομαλή εκτέλεση σε διαφορετικά συστήματα όσο και στην ορθότητα της λειτουργίας της μηχανής τεχνητής νοημοσύνης (όσο αυτό ήταν δυνατό να δοκιμαστεί προκειμένου να αποκλειστούν προφανή λάθη).



Οι επεμβάσεις στο κώδικα που τεκμηριώνεται σε αυτή την διατριβή κάνουν την δημιουργία άλλων υλοποιήσεων, παρόμοιων παιχνιδιών, που χρησιμοποιούν κάποιο είδος σκακιέρας, πολύ πιο απλή καθώς η δομή του προγράμματος μπορεί να παραμείνει ίδια και να γίνουν αλλαγές εσωτερικά στις κλάσεις λογικής παιχνιδιού και στο σχεδιασμό των γραφικών ενώ στο υποσύστημα τεχνητής νοημοσύνης δεν απαιτείται καμία αλλαγή.

## Βιβλιογραφία

- [01] Ν. Γαβαλάς, «Συστήματα για Παιχνίδια Στρατηγικής: Αρχιτεκτονική Επανασχεδίαση με στόχο το Διαχωρισμό του Μηχανισμού Μάθησης από το Μηχανισμό Κινήσεων». Μεταπτυχιακή Διπλωματική Διατριβή για το Ανοικτό Πανεπιστήμιο Κύπρου, 2012.
- [02] [http://en.wikipedia.org/wiki/Personal\\_digital\\_assistant](http://en.wikipedia.org/wiki/Personal_digital_assistant)
- [03] [http://en.wikipedia.org/wiki/Smart\\_phone](http://en.wikipedia.org/wiki/Smart_phone)
- [04] [http://en.wikipedia.org/wiki/Tablet\\_computer](http://en.wikipedia.org/wiki/Tablet_computer)
- [05] [http://en.wikipedia.org/wiki/Mobile\\_OS](http://en.wikipedia.org/wiki/Mobile_OS)
- [06] [http://en.wikipedia.org/wiki/Palm\\_os](http://en.wikipedia.org/wiki/Palm_os)
- [07] [http://en.wikipedia.org/wiki/Windows\\_mobile](http://en.wikipedia.org/wiki/Windows_mobile)
- [08] [http://en.wikipedia.org/wiki/Symbian\\_os](http://en.wikipedia.org/wiki/Symbian_os)
- [09] [http://en.wikipedia.org/wiki/Android\\_os](http://en.wikipedia.org/wiki/Android_os)
- [10] <http://en.wikipedia.org/wiki/IOS>
- [11] <http://en.wikipedia.org/wiki/Linux>
- [12] [http://en.wikipedia.org/wiki/Firefox\\_OS](http://en.wikipedia.org/wiki/Firefox_OS)
- [13] <http://www.openhandsetalliance.com/>
- [14] [http://en.wikipedia.org/wiki/Google\\_Play](http://en.wikipedia.org/wiki/Google_Play)
- [15] Ε. Ντούτση, «Μοντελοποίηση και βελτίωση της ανθρώπινης ικανότητας σε παιχνίδια στρατηγικής» Διπλωματική Εργασία Πανεπιστήμιο Πατρών, τμήμα Μηχανικών Η/Υ και Πληροφορικής, 2001.

- [16] D. Kalles and P. Kanellopoulos (2001), "On Verifying Game Designs and Playing Strategies using Reinforcement Learning", Proceedings of the 2001 ACM Symposium on Applied Computing, p. 6-11
- [17] D. Kalles and E. Ntoutsi (2002), "Interactive Verification of Game Design and Playing Strategies" Proceedings of the 14th IEEE International Conference on Tools with Artificial Intelligence, Washington D.C., p. 425 - 430
- [18] Jacek Mandziuk, "Knowledge-Free and Learning-Based Methods in Intelligent Game Playing", Springer, 2009
- [19] R.S. Sutton. "Learning to Predict by the Methods of Temporal Differences", Machine Learning 3: 9-44, 1988.
- [20] G. Tesauro. "Temporal Difference Learning and TD-Gammon", Communications of the ACM, March 1995/Vol 38, No 3.
- [21] Brian Tanner and Adam White. RL-Glue: Language-Independent Software for Reinforcement-Learning Experiments. Journal of Machine Learning Research, 10(Sep):2133--2136, 2009.
- [22] <http://www.gamemaker.nl>
- [23] [http://en.wikipedia.org/wiki/Game\\_Maker](http://en.wikipedia.org/wiki/Game_Maker)
- [24] <http://games.stanford.edu/competition/misc/aaai.pdf>
- [25] D. Kalles and I. Fykyouras. "EXAMPLES AS INTERACTION: ON HUMANS TEACHING A COMPUTER TO PLAY A GAME", International Journal on Artificial Intelligence Tools, 2010, Vol. 16, No. 6 pp. 717-732
- [26] D. Kalles. "Player Co-modeling in a Strategy Board Game: Discovering how to Play Fast", Cybernetics and Systems, Vol 39, No. 1, pp. 1 - 18, 2008
- [27] D. Kalles, P. Kanellopoulos (2008), "A minimax tutor for learning to play a board game", Proceedings of the AI in Games Workshop, 18th European Conference on Artificial Intelligence, Patras, Greece

- [28] R.S. Sutton & A.G. Barto. "Reinforcement Learning - An Introduction", MIT Press, Cambridge, Massachusetts, 1998.
- [29] <https://www.virtualbox.org/>
- [30] <http://el.wikipedia.org/wiki/XML>
- [31] <http://en.wikipedia.org/wiki/Android-x86>
- [32] <http://www.android-x86.org/>
- [33] <http://www.tech.edu.gr/index.php/el/android/mrgame>
- [34] Mario Zechner, Robert Green, "Beginning Android 4 Games Development", Apress, 2011
- [35] Donn Felker, Joshua Dobbs, "Android Application Development For Dummies" Wiley Publishing Inc, 2011.
- [36] WEI-MENG LEE, "Beginning Android 4 Application Development" John Wiley & Sons Inc, 2012.
- [37] Mark L. Murphy, "The Busy Coder's Guide to Advanced Android Development", Commons Ware, 2011.
- [38] Jason Morris, "Android User Interface Development, Beginner's Guide" Packt Publishing, 2011.
- [39] [http://en.wikipedia.org/wiki/Code\\_refactoring](http://en.wikipedia.org/wiki/Code_refactoring)
- [40] [http://en.wikipedia.org/wiki/Design\\_pattern\\_%28computer\\_science%29](http://en.wikipedia.org/wiki/Design_pattern_%28computer_science%29)
- [41] [http://en.wikipedia.org/wiki/Singleton\\_pattern](http://en.wikipedia.org/wiki/Singleton_pattern)
- [42] [http://en.wikipedia.org/wiki/Adapter\\_pattern](http://en.wikipedia.org/wiki/Adapter_pattern)
- [43] <http://developer.android.com/reference/android/app/Activity.html>
- [44] <http://developer.android.com/tools/help/logcat.html>

- [45] <http://el.wikipedia.org/wiki/Java>
- [46] <http://el.wikipedia.org/wiki/Linux>
- [47] [http://el.wikipedia.org/wiki/Κάρτα\\_μνήμης](http://el.wikipedia.org/wiki/Κάρτα_μνήμης)
- [48] <http://el.wikipedia.org/wiki/GUI>
- [49] <http://developer.android.com/reference/android/graphics/Canvas.html>
- [50] <http://developer.android.com/tools/sdk/eclipse-adt.html>
- [51] <http://developer.android.com/tools/help/emulator.html>
- [52] <https://www.virtualbox.org/>
- [53] <http://developer.android.com/reference/android/widget/ScrollView.html>
- [54] [http://en.wikipedia.org/wiki/Open\\_Virtualization\\_Format](http://en.wikipedia.org/wiki/Open_Virtualization_Format)
- [55] <https://www.eclipse.org/>
- [56] <http://developer.android.com/sdk/index.html>
- [57] <http://staruml.sourceforge.net/en/>
- [58] [http://en.wikipedia.org/wiki/Strategy\\_board\\_game](http://en.wikipedia.org/wiki/Strategy_board_game)
- [59] <http://www.engr.uconn.edu/~steve/Cse293/ModODLAndroid.pdf>
- [60] Android Programmer's Guide, J.F. DiMarzio, McGraw-Hill, 2008
- [61] <http://el.wikipedia.org/wiki/Σκάκι>
- [62] <http://el.wikipedia.org/wiki/Τάβλι>
- [63] [http://en.wikipedia.org/wiki/Zero-sum\\_game](http://en.wikipedia.org/wiki/Zero-sum_game)

# Παράρτημα Α

## Γλωσσάρι

### A.1 Γλωσσάρι Ορολογιών

**API (Application Programming Interface):** Ένα API είναι μια συλλογή/βιβλιοθήκη από όλες τις δημόσιες μεθόδους και ιδιότητες που ανήκουν σε μία ομάδα κλάσεων. Το API ορίζει τον τρόπο με τον οποίο οι προγραμματιστές μπορούν να χρησιμοποιήσουν τη λειτουργικότητα που προσφέρει αυτή η βιβλιοθήκη κλάσεων στα προγράμματα τους.

**Java Package:** Στην γλώσσα Java οι κλάσεις είθισται να ομαδοποιούνται σε λογικές οντότητες που ονομάζονται πακέτα. Οι κλάσεις που βρίσκονται στο ίδιο πακέτο μπορούν να χρησιμοποιήσουν τις δημόσιες ή δηλωμένες με εμβέλεια πακέτου μεθόδους άλλων κλάσεων (χωρίς να χρειάζεται η εντολή `import`). Η κάθε κλάση χαρακτηρίζεται μοναδικά από το πλήρες όνομα της που περιέχει και το όνομα (διαδρομή) του πακέτου.

**XML (Extensible Markup Language):** είναι μία γλώσσα σήμανσης, που περιέχει ένα σύνολο κανόνων για την ηλεκτρονική κωδικοποίηση κειμένων. Η XML σχεδιάστηκε δίνοντας έμφαση στην απλότητα, τη γενικότητα και τη χρησιμότητα στο Διαδίκτυο. Είναι μία μορφοποίηση δεδομένων κειμένου, με ισχυρή υποστήριξη Unicode για όλες τις γλώσσες του κόσμου. Αν και η σχεδίαση της XML εστιάζει στα

κείμενα, χρησιμοποιείται ευρέως για την αναπαράσταση αυθαίρετων δομών δεδομένων, που προκύπτουν για παράδειγμα στις υπηρεσίες ιστού [30].

**Activity (Android Activity):** Σχεδόν όλες οι Activities αλληλεπιδρούν με το χρήστη και φροντίζουν για τη δημιουργία ενός παραθύρου όπου μπορούν να τοποθετηθούν τα αντικείμενα με τα οποία αλληλεπιδρά ο χρήστης. Ενώ οι Activities συχνά παρουσιάζονται στο χρήστη ως παράθυρα σε πλήρη οθόνη, μπορούν επίσης να χρησιμοποιούνται με άλλους τρόπους: ως πλωτά παράθυρα ή ενσωματωμένες στο εσωτερικό μιας άλλης δραστηριότητας [43].

**LogCat (Android Logging System):** Είναι η εντολή με την οποία γίνεται καταγραφή των συμβάντων του Android. Λόγω αυτής της εντολής το σύστημα καταγραφής έχει πάρει το ίδιο όνομα (logcat) [44].

**Java VM (Java Virtual Machine):** Ο κώδικας byte είναι η μορφή που παίρνει ο πηγαίος κώδικας της Java όταν μεταγλωττίζεται. Όταν πρόκειται να εκτελεστεί η εφαρμογή σε ένα μηχάνημα, το Java Virtual Machine που πρέπει να είναι εγκατεστημένο σε αυτό θα αναλάβει να διαβάσει τα αρχεία .class. Στη συνέχεια τα μεταφράζει σε γλώσσα μηχανής που να υποστηρίζεται από το λειτουργικό σύστημα και τον επεξεργαστή, έτσι ώστε να εκτελεστεί [45].

**Linux:** Η ονομασία Linux, είναι ένας γενικός όρος αναφοράς σε λειτουργικά συστήματα που βασίζονται στον πυρήνα Linux. Η αρχιτεκτονική του Linux είναι βασισμένη στις αρχές του λειτουργικού Unix αλλά έχει αναπτυχθεί εκ του μηδενός και δεν περιλαμβάνει κώδικα από το Unix. Η ανάπτυξη του Linux είναι χαρακτηριστικό παράδειγμα εθελοντικής συνεργασίας από διαδικτυακές κοινότητες, ενώ όλο το έργο είναι ανοικτού κώδικα και ελεύθερα προσβάσιμο από όλους για αντιγραφή, τροποποίηση ή αναδιανομή χωρίς περιορισμό [46].

**SD Card (Smart Media):** Κάρτα μνήμης για την αποθήκευση ψηφιακών δεδομένων. Χρησιμοποιείται πολύ συχνά σε συσκευές όπως ψηφιακές φωτογραφικές μηχανές, κινητά τηλέφωνα, υπολογιστές κλπ. Οι κάρτες μνήμης είναι μικρές σε μέγεθος, επανεγγράψιμες και διατηρούν τα δεδομένα τους χωρίς την ανάγκη για τροφοδοσία με ρεύμα [47].

**GUI (Γραφικό Περιβάλλον Χρήστη):** είναι ένα σύνολο γραφικών στοιχείων, τα οποία εμφανίζονται στην οθόνη κάποιας ψηφιακής συσκευής (π.χ. Η/Υ) και χρησιμοποιούνται για την αλληλεπίδραση του χρήστη με τη συσκευή αυτή. Παρέχουν στον τελευταίο, μέσω γραφικών, ενδείξεις και εργαλεία προκειμένου αυτός να φέρει εις πέρας κάποιες επιθυμητές λειτουργίες [48].

**Canvas (Android Canvas):** Κλάση του API του Android που διαχειρίζεται τις draw κλήσεις για σχεδίαση αντικειμένων μέσα σε μία δραστηριότητα του Android (Activity) [49].

**ADT (Android Development Tools):** Είναι ένα πρόσθετο (plugin) για το περιβάλλον ανάπτυξης εφαρμογών Eclipse που του δίνει την δυνατότητα να χρησιμοποιηθεί ως εργαλείο για την ανάπτυξη Android εφαρμογών [50].

**Android Emulator (Simulator):** Το πακέτο λογισμικού με τα εργαλεία για την ανάπτυξη Android εφαρμογών περιλαμβάνει και μία εφαρμογή που προσομοιώνει την λειτουργία μίας φορητής – εικονικής Android συσκευής [51].

**Virtual Box (Oracle Virtual Box):** Είναι ένα ανοιχτού κώδικα λογισμικό για την δημιουργία εικονικών μηχανών (virtual machines) με επεξεργαστές X86. Χρησιμοποιήθηκε για την εκτέλεση του Android X86 μίας έκδοσης του Android για X86 επεξεργαστές [52].

**Scroll View (Android):** Είναι ένα αντικείμενο που περιέχει μία δραστηριότητα του Android και επιτρέπει τον χρήστη να μετακινεί τα περιεχόμενα την οθόνης ώστε να είναι δυνατό να προβάλλονται σε αυτή αντικείμενα μεγαλύτερα από την φυσική διάσταση της οθόνης [53].

**Open Virtualization Format (OVF):** Είναι ένα ανοιχτού κώδικα στάνταρτ για την αποθήκευση και διανομή μίας εικονικής μηχανής [54].

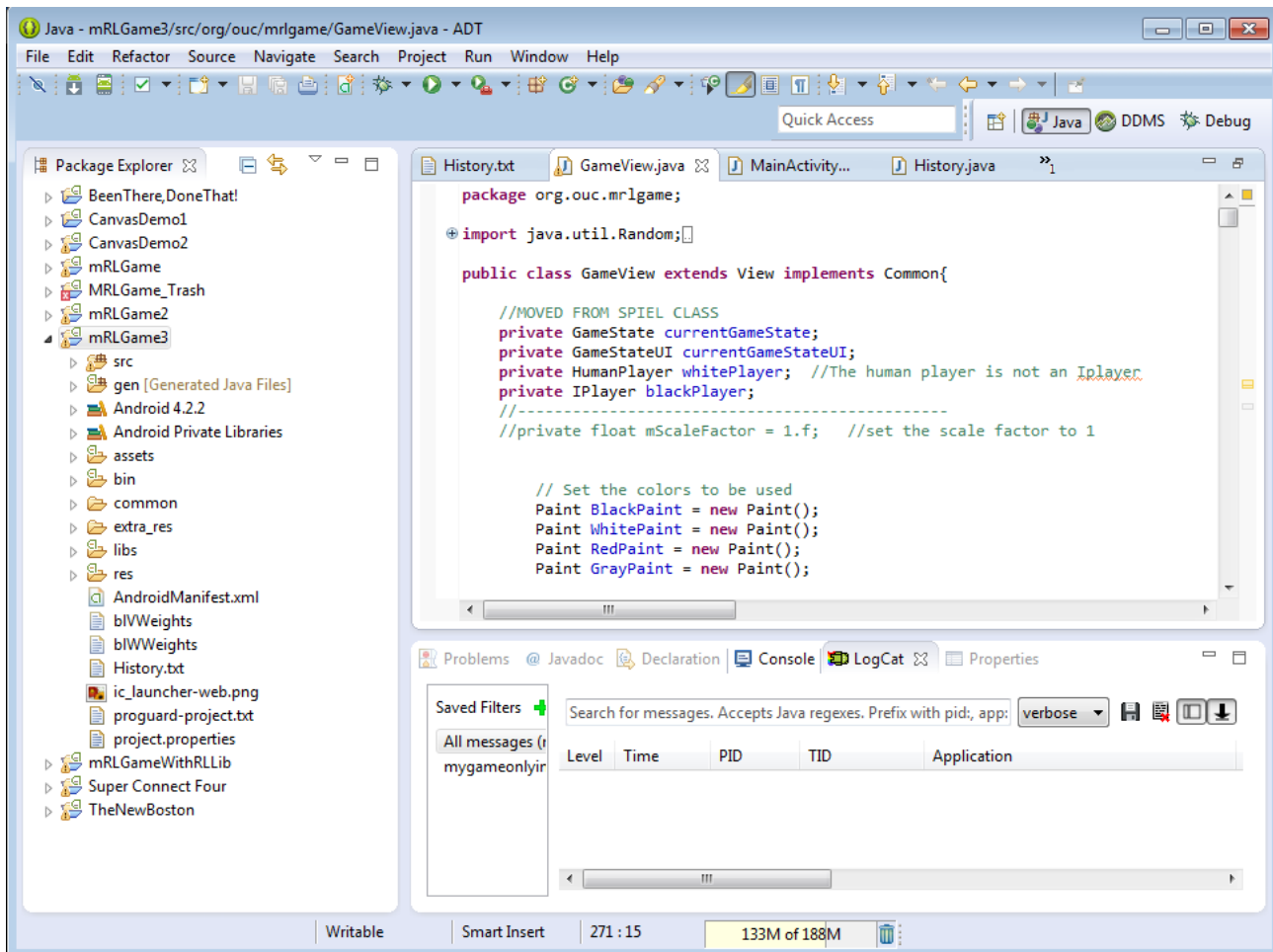


# Παράρτημα Β

## Εργαλεία που χρησιμοποιήθηκαν και η δομή ενός Android Project

### B.1 Eclipse Android Development Tools

Το εργαλείο που χρησιμοποιήθηκε για την ανάπτυξη της εφαρμογής είναι το Eclipse [55], το οποίο είναι από τα πιο καλά εργαλεία για την ανάπτυξη java εφαρμογών. Για να είναι δυνατή η ανάπτυξη Android εφαρμογών στο Eclipse προστέθηκε το ADT (Android Development Tools) [50] που είναι ένα plug-in και προσθέτει στο Eclipse αυτή την δυνατότητα. Επίσης υπάρχει η δυνατότητα τις λήψης [56] του έτοιμου πακέτου που περιέχει ό, τι χρειάζεται κάποιος για αναπτύξει εφαρμογές σε Android που περιέχει το Eclipse με το ADT plugin, το Android SDK, Platform Tools και την εφαρμογή εικονικής Android μηχανής (Emulator) [Εικόνα Β.1.1].



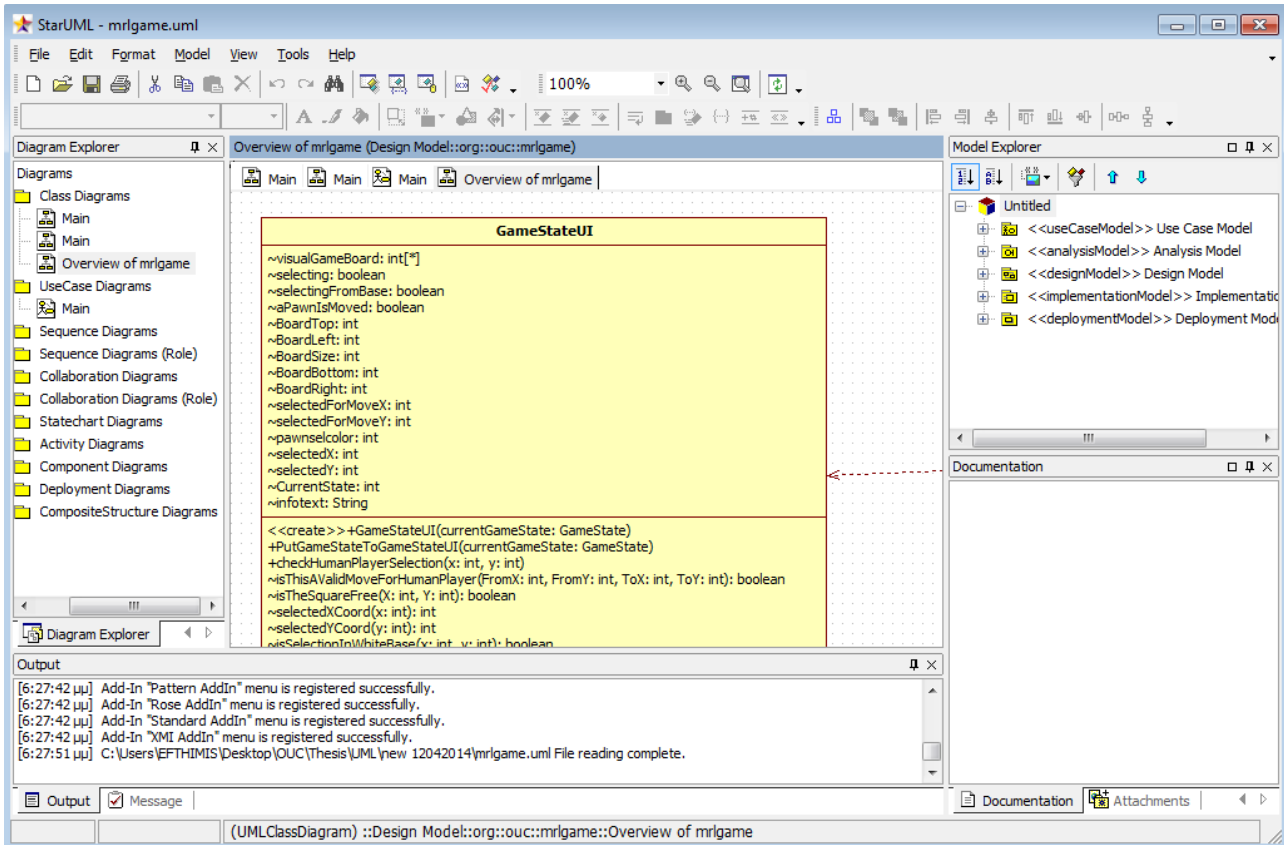
Εικόνα Β.1.1: Το περιβάλλον ανάπτυξης του Eclipse με το ADT plugin [55].

Το περιβάλλον ανάπτυξης εκτός από την ανάπτυξη της εφαρμογής κατά την εκτέλεση της «προωθεί» την εφαρμογή σε κάποια Android συσκευή, πραγματική ή εικονική. Επιπλέον ο προγραμματιστής έχει στην διάθεση του εργαλεία όπως το logcat [44] όπου βλέπει μηνύματα από την Java Virtual Machine σχετικά με την πορεία εκτέλεσης της εφαρμογής αλλά μπορεί και να στείλει δικά του μηνύματα. Να σημειωθεί ότι το logcat ενημερώνεται σε πραγματικό χρόνο.

Παράλληλα υπάρχει η δυνατότητα πρόσβασης στο σύστημα αρχείων της συσκευής την ώρα που γίνεται η εκτέλεση όπως επίσης και προβολή όλων των διεργασιών που τρέχουν επάνω στη συσκευή σε πραγματικό χρόνο.

## B2. Star-UML

Για την δημιουργία των UML διαγραμμάτων κλάσεων χρησιμοποιήθηκε η εφαρμογή StarUML [57]. Η εφαρμογή αυτή είναι ανοιχτού κώδικα και το πολύ βολικό χαρακτηριστικό της είναι ότι μπορεί να «διαβάζει» τον πηγαίο κώδικα σε java και να παράγει το UML διάγραμμα κλάσεων [Εικόνα B.2.1].



Εικόνα B.2.1: Η εφαρμογή StarUML [57].

## B3. Η δομή ενός Android project

Ένα Android project αποτελείται από τα παρακάτω στοιχεία.

**AndroidManifest.xml:** Το πλέον βασικό αρχείο σε ένα Android Project. Περιέχει την περιγραφή των Activities, από ποιο Activity ξεκινάει η εκτέλεση, ποια δικαιώματα έχει η εφαρμογή επάνω στη συσκευή, για ποια έκδοση του Android API έχει σχεδιαστεί η εφαρμογή κ.ά.

**Φάκελος src:** Περιέχει όλες τις κλάσεις της εφαρμογής σε γλώσσα Java, δηλαδή τον πηγαίο κώδικα. Οι κλάσεις είναι οργανωμένες σε πακέτα.

**Φάκελος gen:** Generated Java library. Είναι αποκλειστικά για εσωτερική χρήση του Android

**Φάκελος res (resources):** Περιέχει όλα τα βοηθητικά αρχεία όπως εικόνες, αρχεία XML για τον σχεδιασμό των Activities κ.α. Οι παρακάτω φάκελοι: Drawable, Layout & Values περιέχονται σε αυτό το φάκελο.

**Φάκελος drawable:** Περιέχει όλα τα αρχεία γραφικών. Υπάρχουν διάφοροι τύποι του φακέλου drawable όπως drawable-mdpi, drawable-hdpi που σχετίζονται με τις διαφορετικές αναλύσεις οθόνης που θα εκτελείται η εφαρμογή.

**Φάκελος layout:** Περιέχει τα XML αρχεία που περιγράφουν τον σχεδιασμό των Activities όπως κουμπιά, πλαίσια κειμένου κ.α.

**Φάκελος values:** Περιέχει XML αρχεία στα οποία αποθηκεύονται διάφορες αλφαριθμητικές τιμές όπως τίτλοι χειριστηρίων, ετικέτες κ.α.

# Παράρτημα Γ

## Εγχειρίδιο εγκατάστασης & χρήσης

### Γ.1 Εγκατάσταση του εφαρμογής

Για την εγκατάσταση του προγράμματος θα πρέπει να έχετε μία συσκευή που να εκτελεί το λειτουργικό σύστημα Android με συμβατές να είναι όλες οι εκδόσεις του από την έκδοση 2.2 και μετά ή κάποιο προσομοιωτή. Υπάρχει επίσης το project Android X86 το οποίο είναι μία έκδοση του Android που μπορεί να εκτελείται σε υπολογιστές συμβατούς με X86.

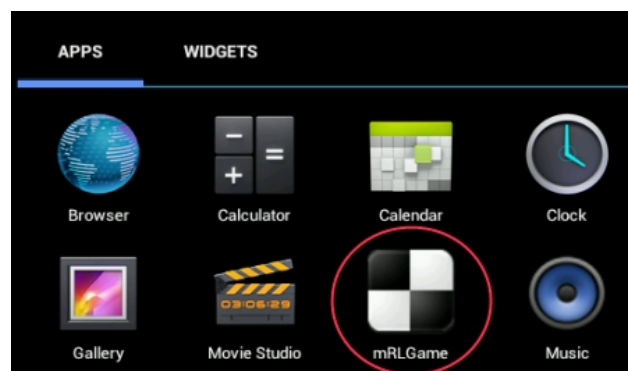
Για να επιτρέψει η συσκευή την εγκατάσταση της εφαρμογής θα πρέπει να έχει ενεργοποιηθεί από τις ρυθμίσεις της συσκευής η ρύθμιση που επιτρέπει την εγκατάσταση εφαρμογών που δεν έχουν «κατεβεί» από το Android Play Store το οποίο σε κάποιες εκδόσεις αναφέρεται ως Android Market.

Στη συνέχεια πρέπει μέσα από την Android συσκευή σας να επισκεφτείτε την ιστοσελίδα της εφαρμογής [33] με την χρήση της εφαρμογής φυλλομετρητή ιστού της συσκευής σας και να επιλέξετε τον σύνδεσμο για να κατέβει η εφαρμογή. Η συσκευή σας θα σας ρωτήσει αν θέλετε να προχωρήσετε σε εγκατάσταση και εσείς πρέπει να επιλέξετε «Ναι». Μετά από αυτό το εικονίδιο της εφαρμογής [Εικόνα Γ.2.1] θα έχει τοποθετηθεί στις εφαρμογές σας.

Ένας εύκολος τρόπος για να εκτελέσετε την εφαρμογή χωρίς να χρειαστείτε μία συσκευή με Android ούτε να κατεβάσετε και να εγκαταστήσετε το Android X86 είναι να κατεβάσετε την έτοιμη εικονική μηχανή (VM) από την ιστοσελίδα του mRLGame [33] που είναι σε format OVF και μπορεί να εκτελεστεί στα περισσότερα συστήματα εικονικών μηχανών ενώ προτείνεται το Virtual Box [29] της Oracle. Στην εικονική μηχανή είναι ήδη κατεβασμένη και εγκατεστημένη η εφαρμογή οπότε και δεν χρειάζονται περαιτέρω εργασίες.

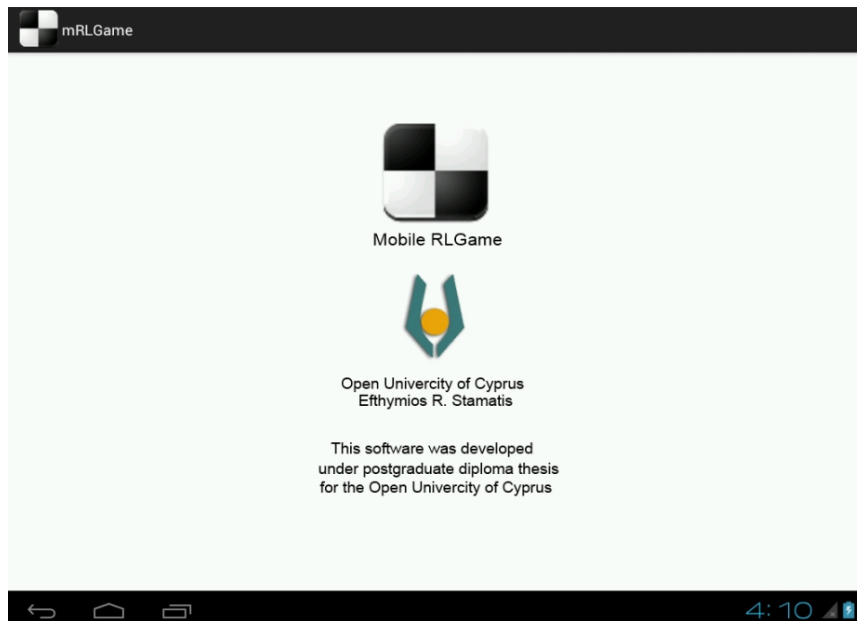
## Γ.2 Εκτέλεση της εφαρμογής

Η εφαρμογή εκτελείται από το εικονίδιο της [Εικόνα Γ.2.1], που εμφανίζεται σε κόκκινο κύκλο, πατώντας πάνω σε αυτό. Προϋπόθεση είναι να έχει γίνει η εγκατάσταση της εφαρμογής, όπως περιγράφηκε παραπάνω.



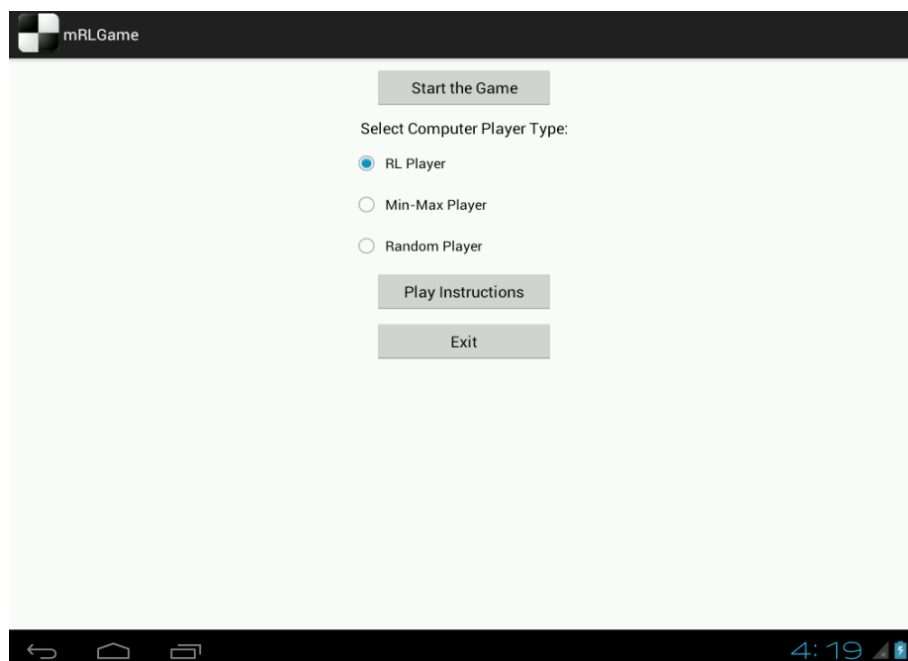
Εικόνα Γ.2.1: Το εικονίδιο της εφαρμογής mRLGame

Μετά την εκτέλεση της εφαρμογής εμφανίζεται η αρχική οθόνη, η οποία έχει την έννοια της ενημερωτικής οθόνης για το πρόγραμμα που ξεκίνησε η εκτέλεση γνωστή ως splash screen [Εικόνα Γ.2.2].



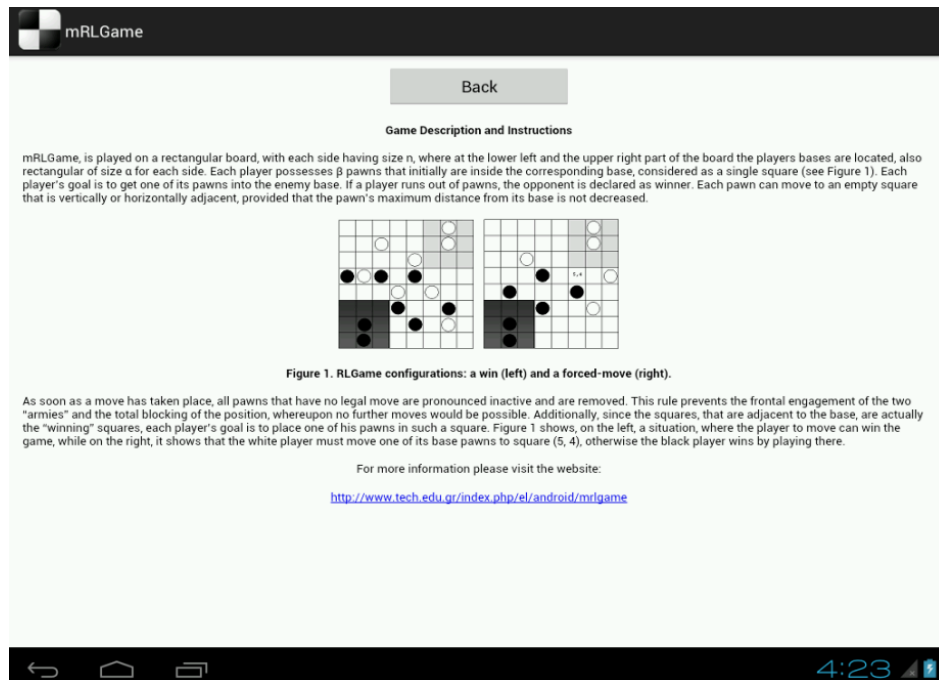
Εικόνα Γ.2.2: Η αρχική οθόνη της εφαρμογής mRLGame

Η αρχική οθόνη μένει για 5 δευτερόλεπτα στην οθόνη και μετά εμφανίζει αυτόματα την οθόνη επιλογών [Εικόνα Γ.2.3]. Σε αυτή την οθόνη ο χρήστης μπορεί να επιλέξει τον τύπο του παίκτη υπολογιστή ο οποίος μπορεί να είναι: RL-Player (παίκτης ενισχυτικής μάθησης), Min-Max Player (παίκτης που χρησιμοποιεί τον αλγόριθμο Min-Max) και Random Player (παίκτης με τυχαία επιλογή κινήσεων) και να ξεκινήσει το παιχνίδι. Επίσης μπορεί να δει την οθόνη με τους κανόνες του παιχνιδιού και να τερματίσει την εφαρμογή.



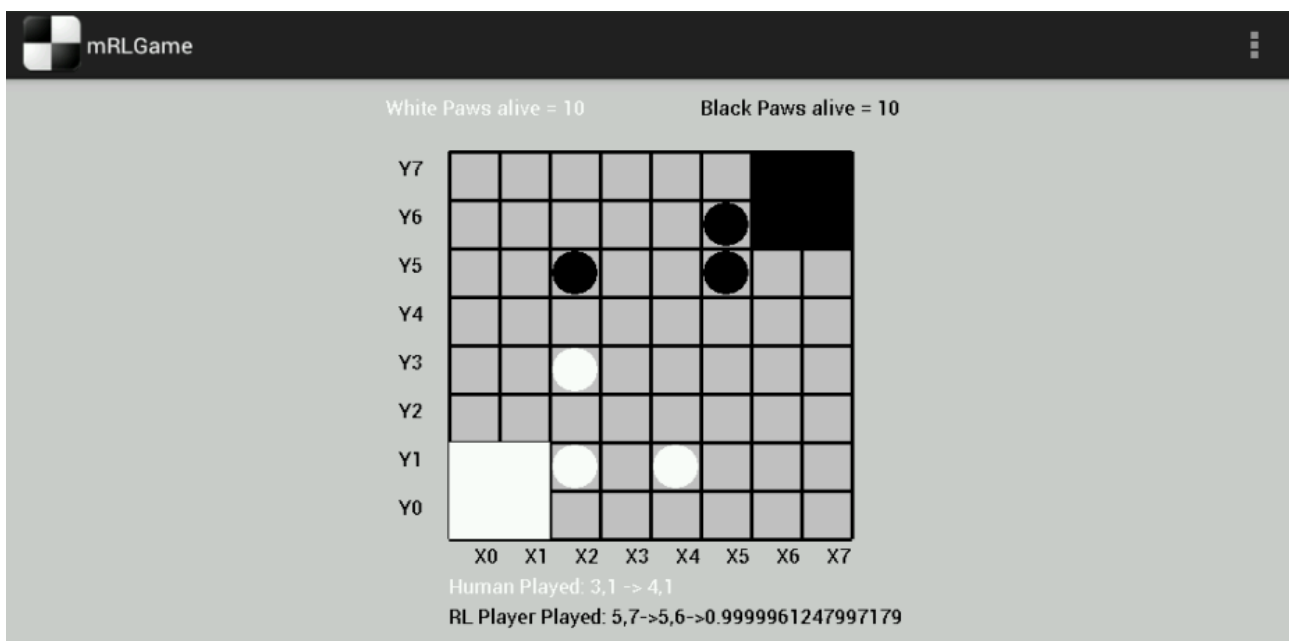
Εικόνα Γ.2.3.: Η οθόνη επιλογών της εφαρμογής mRLGame

Η οθόνη εμφάνισης των οδηγιών παιχνιδιού έχει την μορφή της παρακάτω εικόνας [Εικόνα Γ.2.4.]. Στην οθόνη υπάρχει και υπερσύνδεση για την ιστοσελίδα της εφαρμογής.



Εικόνα Γ.2.4.: Η οθόνη εμφάνισης των οδηγιών της εφαρμογής mRLGame

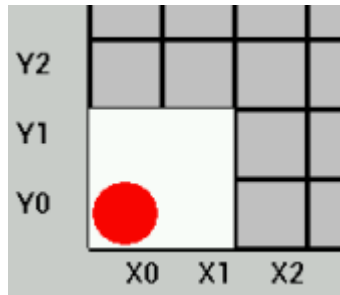
Η οθόνη παιχνιδιού είναι και η βασική οθόνη της εφαρμογής [Εικόνα Γ.2.5.]. Σε αυτή εμφανίζεται η σκακιέρα με τους 2 παίκτες. Ο άσπρος παίκτης είναι ο άνθρωπος και παίζει πάντα πρώτος ενώ ο υπολογιστής παίκτης είναι πάντα ο μαύρος παίκτης. Στην οθόνη εμφανίζονται πληροφορίες όπως πόσα πιόνια είναι ενεργά για κάθε παίκτη όπως και η τελευταία κίνηση του.





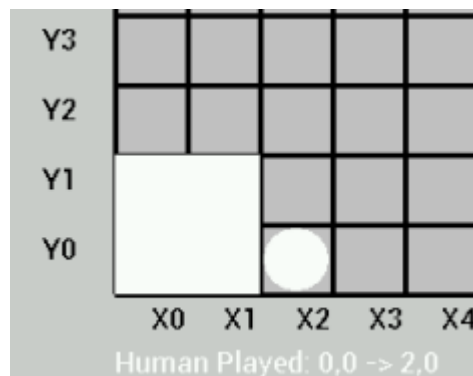
Εικόνα Γ.2.5.: Η σκακιέρα του παιχνιδιού.

Το παιχνίδι ξεκινάει με την επιλογή του ανθρώπου παίκτη με κλικ μέσα στην λευκή περιοχή. Ένα πιόνι από τα δέκα που έχει στη διάθεση του επιλέγεται και γίνεται κόκκινο [Εικόνα Γ.2.6.].



Εικόνα Γ.2.6: Επιλογή εντός της λευκής βάσης

Στη συνέχεια ο παίκτης πρέπει να επιλέξει ένα όμορο στη βάση τετράγωνο για να μετακινηθεί εκεί το πιόνι [Εικόνα Γ.2.7]. Το πιόνι μετακινήθηκε από τη θέση 0,0 (που είναι η βάση) στη θέση 2,0.



Εικόνα Γ.2.7: Μετακίνηση πιονιού από τη βάση στη θέση 2,0.

Κάθε μετακίνηση πιονιού γίνεται σε δύο χρόνους. Πρώτα επιλέγεται το πιόνι που θέλουμε να μετακινήσουμε το οποίο και γίνεται κόκκινο και στη συνέχεια επιλέγεται το τετράγωνο στο οποίο θέλουμε να μετακινηθεί το πιόνι. Αν η κίνηση δεν είναι σωστή βάσει των κανόνων του παιχνιδιού που περιγράφηκαν στο κεφάλαιο 4 ή το τετράγωνο δεν είναι άδειο η κίνηση δεν γίνεται και το πιόνι παραμένει κόκκινο μέχρι να επιλεγεί μία επιτρεπόμενη κίνηση. Ο χρήστης μπορεί οποιαδήποτε στιγμή να ακυρώσει την επιλογή του, επιλέγοντας εκτός της σκακιέρας και να επιλέξει ένα άλλο πιόνι για να μετακινήσει.