Ανοικτό Πανεπιστήμιο Κύπρου

Σχολή Θετικών και Εφαρμοσμένων Επιστημών

Μεταπτυχιακή Διατριβή στα Πληροφοριακά Συστήματα

Συστήματα για Παιχνίδια Στρατηγικής: Αρχιτεκτονική Επανασχεδίαση με στόχο το Διαχωρισμό του Μηχανισμού Μάθησης από το Μηχανισμό Κινήσεων

Νικόλαος Γαβαλάς

Επιβλέπων Καθηγητής
Δημήτριος Καλλές

Σεπτέμβριος 2012

Ανοικτό Πανεπιστήμιο Κύπρου

Σχολή Θετικών και Εφαρμοσμένων Επιστημών

Συστήματα για Παιχνίδια Στρατηγικής: Αρχιτεκτονική Επανασχεδίαση με στόχο το Διαχωρισμό του Μηχανισμού Μάθησης από το Μηχανισμό Κινήσεων

Νικόλαος Γαβαλάς

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

Σεπτέμβριος 2012

Περίληψη

Ο σκοπός της παρούσας διατριβής είναι η τεχνική συντήρηση υπάρχουσας υλοποίησης του συστήματος διεξαγωγής παιχνιδιών του παιχνιδιού στρατηγικής RLGame.

Το RLGame είναι ένα παιχνίδι στρατηγικής που παίζεται ανάμεσα σε δύο παίκτες χρησιμοποιώντας πιόνια που κινούνται πάνω σε μία επιφάνεια τύπου «σκακιέρας». Η υλοποίηση του συστήματος έχει γίνει σε γλώσσα προγραμματισμού Java και χρησιμοποιεί τεχνικές Ενισχυτικής Μάθησης (*Reinforcement Learning*) και Νευρωνικών Δικτύων (*Neural Networks*) στην προσπάθεια βελτίωσης της τακτικής του.

Η παρούσα διατριβή εστιάζει στις περιοχές της Τεχνητής Νοημοσύνης και της Τεχνολογίας Λογισμικού.

Στα πλαίσια της παρούσας διατριβής πραγματοποιήθηκαν τα ακόλουθα:

1. η μελέτη σκοπιμότητας της επανασχεδίασης και επανυλοποίησης του RLGame με χρήση συστημάτων (πλατφορμών) περιγραφής και ανάπτυξης παιχνιδιών ανοιχτού κώδικα ή πλατφορμών/παραδειγμάτων μηχανικής μάθησης.

2. η επανασχεδίαση της αρχιτεκτονικής του συστήματος ή/και των δομών δεδομένων έτσι ώστε η υλοποίηση του παιχνιδιού (game logic) και η υλοποίηση της Ενισχυτικής Μάθησης να αποτελούν ξεχωριστά τμήματα

3. η ολοκλήρωση του υποσυστήματος μάθησης που προέκυψε από τον διαχωρισμό των τμημάτων του RLGame με υλοποίηση άλλου παιχνιδιού στρατηγικής.

Summary

The purpose of this thesis is the technical maintenance of the RLGame strategy game platform.

RLGame is a strategy, turns-based, game. It is a two-players game and has some similarities to checkers or chess since it played on a similar board. The RLGame platform has been implemented on Java language and it employs Reinforcement Learning and a Neural Networks implementation for approximating the value function, for the improvement of the tactic of the computer players.

The current thesis focuses on the disciplines of Artificial Intelligence and Software Engineering.

The tasks below were carried out for the purposes of this thesis:

1. Brief feasibility study for the redesign and development of the RLGame platform with the use of open source game engines or open source machine learning platforms/paradigms.

2. Redesign of the RLGame platform so that game logic implementation and reinforcement learning implementation are implemented in distinct modules.

3. Integration of the new RLGame reinforcement learning module with another strategy game's logic module.

Ευχαριστίες

Ευχαριστώ τον καθηγητή μου Δημήτρη Καλλέ για την καθοδήγηση και τη συμπαράσταση.

Ευχαριστώ την οικογένεια μου για την συμπαράσταση και την υπομονή που έδειξαν σε όλη τη διάρκεια του μεταπτυχιακού μου.

Περιεχόμενα

Κεφάλαιο 1
Εισαγωγή

Η παρούσα διατριβή εστιάζει στις περιοχές της Τεχνητής Νοημοσύνης και της Τεχνολογίας Λογισμικού και ο απώτερος σκοπός είναι η τεχνική συντήρηση υπάρχουσας υλοποίησης του συστήματος του πρωτότυπου παιχνιδιού στρατηγικής RLGame.

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 ανθρώπινων 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 υλοποιήσει 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 έχει αναπτυχθεί σε γλώσσα προγραμματισμού Java, ενώ το περιβάλλον εκτέλεσης πειραμάτων επιτρέπει τόσο τη μάθηση μέσω εκτέλεσης πειραμάτων ανάμεσα σε υπολογιστικούς παίκτες που χρησιμοποιούν κοινές τεχνικές ενισχυτικής μάθησης (υπολογιστής εναντίον υπολογιστή, Self-play experiments) ενώ δίνεται 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.

1.2 Διάθρωση Διπλωματικής Διατριβής

Η παρούσα διπλωματική διατριβή αποτελείται από 8 κεφάλαια. Το πρώτο κεφάλαιο είναι η εισαγωγή στην οποία παραθέτεται το αντικείμενο της διπλωματικής διατριβής. Το δεύτερο κεφάλαιο ασχολείται με το αντικείμενο των Παιχνιδιών Στρατηγικής και τη θεωρία της Μηχανικής Μάθησης.

Το τρίτο κεφάλαιο ασχολείται με την περιγραφή του παιχνιδιού, του κανόνες του, τις επιλογές τεχνικών τεχνητής νοημοσύνης που έχουν εφαρμοστεί σε αυτό καθώς και την περιγραφή των υλοποιήσεων του συστήματος εκτέλεσης πειραμάτων του παιχνιδιού RLGame. Το τέταρτο κεφάλαιο ασχολείται με τις πλατφόρμες ανάπτυξης παιχνιδιών όπως το GameMaker και πλατφόρμες/πρότυπα για την εκτέλεση πειραμάτων τεχνικής νοημοσύνης όπως η πλατφόρμα RL-Glue, το πρότυπο General Game Play ή η πλατφόρμα ORTS.

Το πέμπτο κεφάλαιο ασχολείται με την μεθοδολογία που ακολουθήθηκε στην παρούσα διπλωματική εργασία.

Το έκτο κεφάλαιο ασχολείται με την περιγραφή της νέας σχεδίασης του RLGame που πραγματοποιήθηκε στα πλαίσια της παρούσας διπλωματικής διατριβής και το έβδομο ασχολείται με την ολοκλήρωση του παιχνιδιού Connect4 με το υποσύστημα του μηχανισμού μάθησης που χρησιμοποιείται στο Rlgame.

Τέλος στο όγδοο κεφάλαιο παραθέτονται τα συμπεράσματα ενώ υπάρχουν και ένα παράρτημα με γλωσσάρι όρων και ένα με οδηγίες για τη χρήση της νέας υλοποίησης.

1.3 Αντικείμενο Τεχνικής Συντήρησης

Στα πλαίσια της παρούσας διατριβής διερευνήθηκε η επανασχεδίαση και επανυλοποίηση του RLGame με χρήση συστημάτων (πλατφορμών) περιγραφής και ανάπτυξης παιχνιδιών ανοιχτού κώδικα καθώς και η επανασχεδίαση της αρχιτεκτονικής του συστήματος ή/και των δομών δεδομένων έτσι ώστε η υλοποίηση του υποσυστήματος λογικής και εκτέλεσης του παιχνιδιού (game logic) και η υλοποίηση των τεχνικών της Ενισχυτικής Μάθησης να αποτελούν ξεχωριστά τμήματα..

1.3.1 Περιγραφή Ενεργειών Τεχνικής Συντήρησης

Η τεχνική συντήρηση του συστήματος παιχνιδιού στρατηγικής RLGame περιλαμβάνει:

- Την επανασχεδίαση της αρχιτεκτονικής του συστήματος
- Την νέα υλοποίηση που ακολουθεί την νέα αρχιτεκτονική αλλά κατά τα άλλα έχει την ίδια λειτουργικότητα με τις υπάρχουσες υλοποιήσεις του παιχνιδιού.
- Την ολοκλήρωση του υπάρχοντος μηχανισμού ενισχυτικής μάθησης με άλλη υλοποίηση παιχνιδιού (connect 4).
- Τον έλεγχο καλής λειτουργίας των νέων υλοποιήσεων μέσω εκτέλεσης πειραμάτων.
- Τεκμηρίωση της νέας υλοποίησης.

Επανασχεδίαση Αρχιτεκτονικής Συστήματος

Όπως αναφέρθηκε αρκετές φορές ως τώρα ο διαχωρισμός της υλοποίησης της λογικής του παιχνιδιού (game logic) και της υλοποίησης της Ενισχυτικής Μάθησης είναι η βασική ανάγκη η οποία επιδιώχθηκε κατά την επανασχεδίαση της αρχιτεκτονικής του συστήματος.

Στη νέα αρχιτεκτονική όπου ο μηχανισμός μάθησης έχει διαχωριστεί από την λογική του παιχνιδιού είναι εφικτή και εύκολη η εφαρμογή αλλαγών σε ένα υποσύστημα (π.χ. στο υποσύστημα Ενισχυτικής Μάθησης) χωρίς να χρειάζονται αλλαγές στο άλλο. ενώ ευνοείται και η ολοκλήρωση του μηχανισμού μάθησης με υλοποίηση άλλου παιχνιδιού.

Τέλος στα πλαίσια της επανασχεδίασης εξετάστηκαν και οι δυνατότητες ολοκλήρωσης του παιχνιδιού με πλατφόρμες ανάπτυξης παιχνιδιών όπως το Gamemaker ή πλατφόρμες διεξαγωγής πειραμάτων ενισχυτικής μάθησης όπως το RLGlue.

Υλοποίηση Νέας Αρχιτεκτονικής Συστήματος

Οι αλλαγές στην υλοποίηση του συστήματος ακολουθούν την νέα αρχιτεκτονική. Η νέα έκδοση του συστήματος, συνεχίζει να έχει την ίδια λειτουργικότητα με τις υπάρχουσες υλοποιήσεις του παιχνιδιού σε ότι αφορά τόσο τη λογική του παιχνιδιού όσο και την υλοποίηση των τεχνικών ενισχυτικής μάθησης.

Όπου κρίθηκε σκόπιμο και εφικτό η υλοποίηση προσαρμόστηκε έτσι ώστε να ακολουθεί καλές πρακτικές αντικειμενοστραφούς προγραμματισμού. Αυτό είχε σαν αποτέλεσμα να απλοποιηθεί σημαντικό κομμάτι κώδικα και να εισαχθούν πιο γενικές κλάσεις που επιτρέπουν για παράδειγμα τη χρήση του μηχανισμού μάθησης από παιχνίδια που έχουν περισσότερους από 2 υπολογιστικούς παίκτες.

Υλοποίηση άλλου Παιχνιδιού και Μηχανισμός Μάθησης

Στα πλαίσια της διερεύνησης των αποτελεσμάτων του διαχωρισμού του μηχανισμού μάθησης από το υποσύστημα της λογικής του παιχνιδιού κρίθηκε σκόπιμο να ελεγχθεί η λειτουργικότητα του μηχανισμού μάθησης κατά την ολοκλήρωση του με ένα άλλο παιχνίδι. Για το σκοπό αυτό επιλέχθηκε το παιχνίδι Connect4 (http://en.wikipedia.org/wiki/Connect Four) [14]. Το συγκεκριμένο παιχνίδι επιλέχθηκε γιατί είναι ένα παιχνίδι στρατηγικής μηδενικού αθροίσματος που παίζεται από δύο παίκτες με εναλλαγή σειράς. Ακόμη, είναι μικρότερης πολυπλοκότητας σε σχέση με το RLGame (αφού έχει μικρότερο αριθμό καταστάσεων), και η υλοποίηση της λογικής του παιχνιδιού είναι σχετικά απλή, ενώ είναι ένα παιχνίδι που έχει ήδη λυθεί με παραδοσιακές τεχνικές τεχνητής νοημοσύνης και υπάρχουν ξεκάθαρες στρατηγικές. Στα πλαίσια της παρούσας διατριβής υλοποιήθηκε το υποσύστημα λογικής παιχνιδιού του Connect4 και

ολοκληρώθηκε με την υλοποίηση του μηχανισμού ενισχυτικής μάθησης που προέκυψε από τον διαχωρισμό των υποσυστημάτων του RLGame.

Έλεγχος Καλής Λειτουργίας

Η πορεία της υλοποίησης των εκδόσεων του RLGame και του connect4 έγινε σε διαφορετικές φάσεις. Σε κάθε φάση υπήρχε έλεγχος καλής λειτουργίας των νέων υλοποιήσεων. Ο έλεγχος καλής λειτουργίας αποτελούταν από τον έλεγχο των επιμέρους υποσυστημάτων (unit testing) ενώ ακολουθούσε και έλεγχος του συστήματος μέσω εκτέλεσης πειραμάτων. Κριτήριο για την καλή λειτουργία του συστήματος αποτέλεσε η σύγκριση των αποτελεσμάτων των πειραμάτων με τα δημοσιευμένα αποτελέσματα πειραμάτων του RLGame.

Τεκμηρίωση

Στο παρόν κείμενο περιέχεται και τεκμηρίωση της νέας υλοποίησης Η τεκμηρίωση εστιάζει στην περιγραφή του διαχωρισμού της νέας υλοποίησης, στην λεπτομερή περιγραφή του υποσυστήματος μάθησης και τέλος στο εγχειρίδιο ολοκλήρωσης νέων παιχνιδιών με το υποσύστημα μηχανισμού μάθησης.

Κεφάλαιο 2
Παιχνίδια Στρατηγικής και Τεχνητή Νοημοσύνη

Αυτό το κεφάλαιο πραγματεύεται τα θέματα των παιχνιδιών στρατηγικής και τα πεδία μηχανικής μάθησης που βρίσκουν εφαρμογή στον τομέα των παιχνιδιών στρατηγικής.

2.1 Παιχνίδια Στρατηγικής και Θεωρία Παιχνιδιών

Τα παιχνίδια στρατηγικής έχουν παίξει ένα σημαντικό ρόλο στον ανθρώπινο πολιτισμό και τα παλαιότερα επιτραπέζια παιχνίδια στρατηγικής όπως το τάβλι ή το σκάκι παίζονται εδώ και χιλιάδες χρόνια. Φαίνεται ότι σε όλους τους πολιτισμούς οι άνθρωποι προσπαθούσαν να βρουν τρόπους να ανταγωνιστούν και να αναπτύξουν τις ικανότητες τους σε τέτοιου είδους πνευματικά παιχνίδια.

Από το ξεκίνημα της έρευνας της τεχνητής νοημοσύνης το πεδίο της θεωρίας παιχνιδιών βρίσκεται στο επίκεντρο των ερευνητικών προσπαθειών.

Η Θεωρία Παιχνιδιών έχει εφαρμογή σε πολλούς κλάδους της επιστήμης εκτός των παιχνιδιών στρατηγικής. Ξεκίνησε σαν κλάδος των οικονομικών με το βιβλίο των John von Neumann και Oskar Morgenstern Theory of Games and Economic Behaviour (Θεωρία Παιγνίων και Οικονομική Συμπεριφορά) πάνω σε παιχνίδια μηδενικού αθροίσματος[25]. Το κύριο αντικείμενό της είναι η ανάλυση των αποφάσεων σε καταστάσεις (παιχνίδια) στρατηγικής αλληλεπίδρασης (strategic interdependence) [25]. Με βάση τη θεωρία των παιχνιδιών, 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 κίνησής λαμβάνοντας υπόψη την παρούσα κατάστασή του, την κατάσταση του αντιπάλου, τις επιπτώσεις της κίνησής του και την ενδεχόμενη επόμενη κίνηση του αντιπάλου[6].

2.2 Παιχνίδια στρατηγικής και Τεχνητή Νοημοσύνη

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[1,2,6,24].

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: ανακάλυψαν αλγόριθμους όπως ο αλγόριθμος minimax με κλάδεμα που έκανε εφικτή ως ένα βαθμό, τη διερεύνησης 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 hardware[24].

Στη δεκαετία του 1990 η IBM, πρώτα με το Deep Though και στη συνέχεια με το Deep Blue, επένδυσε πολύ 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. Το σύστημα δεν μαθαίνει κάτι εμπειρικά.

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)[9]. Η πιο χαρακτηριστική επιτυχία της μεθόδου TD(λ) είναι το TD–Gammon για το παιχνίδι τάβλι που υλοποιήθηκε από τον Tesauro[8].

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[24].

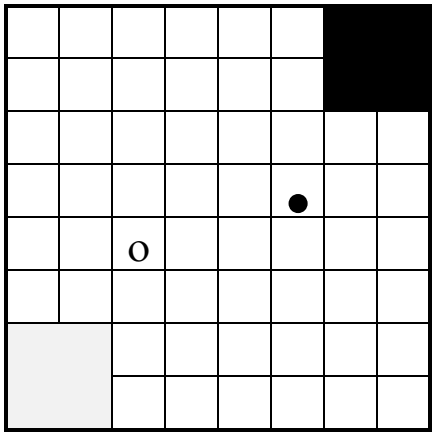
Κεφάλαιο 3
RLGame

Σε αυτό το κεφάλαιο παραθέτεται η περιγραφή του παιχνιδιού RLGame, και των επιλογών της ενισχυτικής μάθησης και των νευρωνικών δικτύων που χρησιμοποιούνται στο παιχνίδι.

3.1 Περιγραφή 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 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 αριστερά γωνία της σκακιέρας όπου βρίσκεται και η βάση του ενώ τα πιόνια του «μαύρου» εκκινούν από την πάνω δεξιά γωνία.



Εικόνα 3.1 : Σκακιέρα 8x8 με βάση 2x2

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

3.2 Κανόνες

Οι κανόνες του παιχνιδιού δεν έχουν αλλάξει από την αρχική παρουσίαση και περιγράφονται αναλυτικά στις εργασίες [01-06]. 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 πιόνι εκκινεί από την βάση του - που θεωρείται ένα ενιαίο τετράγωνο - επομένως κάθε πιόνι που εκκινεί από την βάση μπορεί να μετακινηθεί σε κάποιο ελεύθερο από τα παρακείμενα στη βάση τετράγωνα που είναι ελεύθερα.

Τα πιόνια 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:

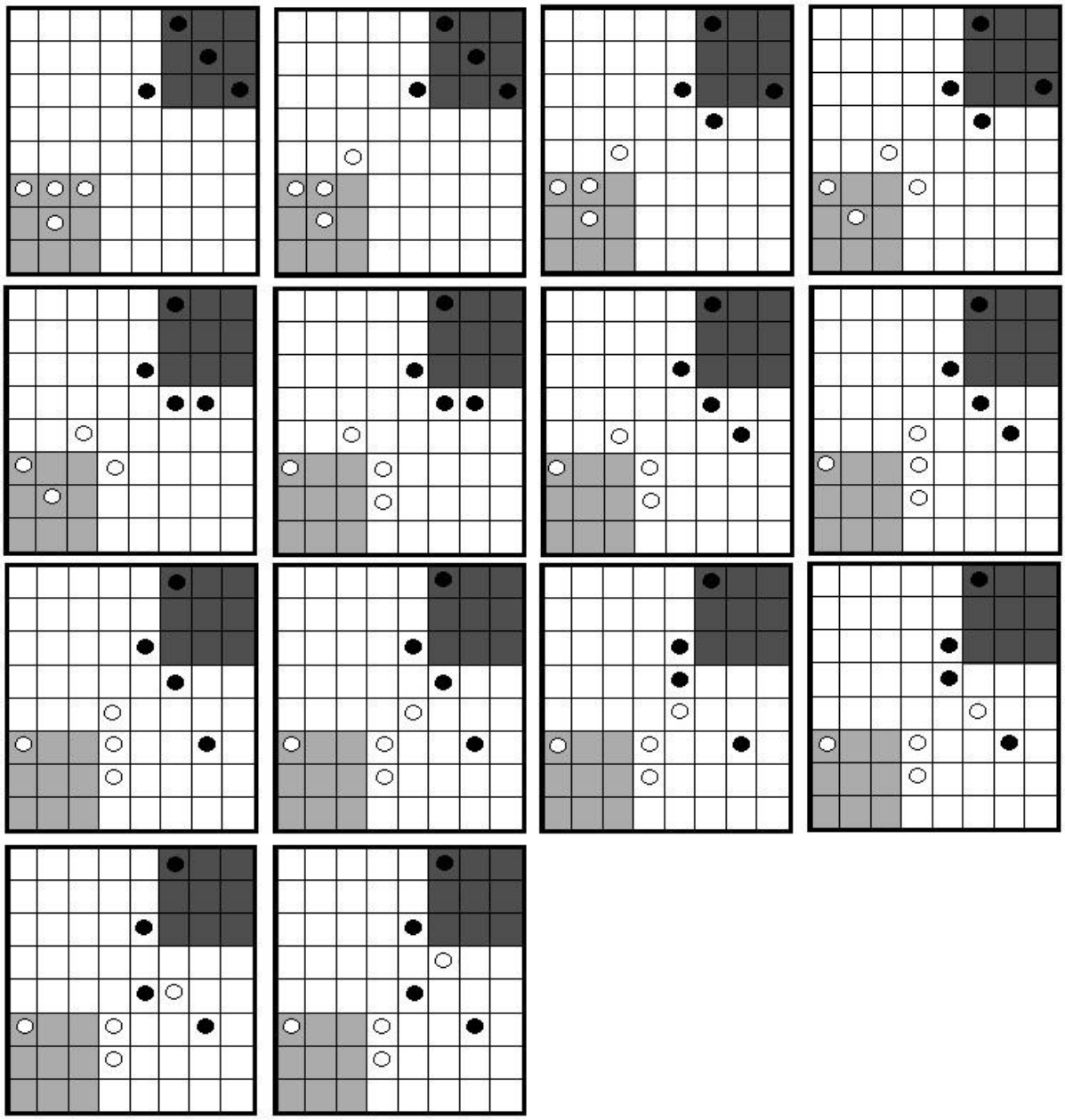
Έστω (x,y) θέση του πιονιού στην σκακιέρα, το πιόνι μπορεί να μετακινηθεί οριζόντια ή κάθετα κατά μία θέση (x,y') ή (x', y) αν και μόνο αν ισχύει:

- max(x - a, y - a) <= max(x - a, y' - a) ή max(x - a, y - a) <= max(x' - a, y - a) εάν το πιόνι ανήκει στον «λευκό» παίκτη ή αν ισχύει:

- max(n - a - x,n - a - y) <= max(n - a - x, n - a – y') ή max(n - a - x,n - a - y) <= 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 [06] παραθέτεται ένα παράδειγμα παιχνιδιού στο οποίο κερδίζει ο λευκός παίκτης.



Εικόνα 3.2 : Παράδειγμα παιχνιδιού RLGame όπου εκκινεί ο μαύρος και κερδίζει ο λευκός[6]

3.3 Επιλογές Τεχνητής Νοημοσύνης

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(λ) – Ενισχυτική Μάθηση χρονικών διαφορών όπως αυτή περιγράφεται στην εργασία του Sutton [09]. Ακόμη για κάθε παίκτη χρησιμοποιείται μία υλοποίηση Νευρωνικού Δικτύου (Neural Network) που χρησιμοποιείται για την κατά προσέγγιση αναπαράσταση του χώρου καταστάσεων του κάθε

παίκτη [07,03] 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 [01 - 06] που αποτελούν και την 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.

3.3.1 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 [20].

Ακόμη το σύνολο των καταστάσεων του RLGame είναι πεπερασμένο και εξαρτάται μόνο από το μέγεθος της σκακιέρας, 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.

3.3.2 Εφαρμογή Ενισχυτικής Μάθησης

Για τον μηχανισμό μάθησης, η ερευνητική ομάδα που δημιούργησε το RLGame αποφάσισε να χρησιμοποιήσει τεχνικές της Ενισχυτικής Μάθησης (Reinforcement Learning) [6]. 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/(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 (ε- 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 γ (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 (λ). Στην online έκδοση του αλγόριθμου η ανανέωση πραγματοποιείται σε κάθε βήμα. Ο παράγοντας παράληψης (forgetting factor) λ 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.

Ακόμη 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 υπολογισμός της βέλτιστης στρατηγικής του υπολογιστικού παίκτη που χρησιμοποιεί τον μηχανισμό μάθησης δεν είναι εφικτή η χρήση ενός πίνακα αντιστοίχησης 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, κατά προσέγγιση υπολογισμού της τιμής αξιολόγησης μία κατάστασης ικανοποιείται από την επιλογή των νευρωνικών δικτύων.

3.3.3 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 $2*(n^2-2*a^2+5)$ νευρώνες
- το κρυμμένο επίπεδο (hidden layer) που περιέχει n^2-2*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 $2 * (DIMBOARD^2 - 2 * DIMBASE^2 + 5)$. Όπως φαίνεται στον μονοδιάστατο πίνακα ορίζουμε τουλάχιστον 2 φορές τον αριθμό των διαθέσιμων θέσεων της σκακιέρας $DIMBOARD^2-2*DIMBASE^2$. 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 πιόνι 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.

3.4 Υπάρχουσες υλοποιήσεις

Η υλοποίηση του συστήματος έχει γίνει σε γλώσσα προγραμματισμού Java και η αρχική υλοποίηση έχει γίνει το 2001 και έχει παρουσιαστεί από τους κυρίους Καλλέ Δημήτρη, και Κανελλόπουλο Παναγιώτη [01]. 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/εκπαιδεύει κάποιο υπολογιστικό παίκτη, ή τέλος πειράματα στα οποία υλοποίηση του αλγόριθμου minimax παίζει εναντίον/εκπαιδεύει κάποιο υπολογιστικό παίκτη.

3.4.1 Υπολογιστής εναντίον Υπολογιστή

Αυτή είναι η πιο παλιά μορφή του συστήματος RLGame στην οποία έχουν βασιστεί σειρά πειραμάτων που αναφέρονται στις δημοσιεύσεις [01-06]. 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 χρήση 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 γ, τιμή παράγοντα παράληψης λ, τιμή ανταμοιβής στην περίπτωση νίκης, τιμή e-greedy πολιτικής.

3.4.2 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 [04]. 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 γ, τιμή παράγοντα παράληψης λ, 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.

3.4.3 Άνθρωπος εναντίον Υπολογιστή

Αυτή η μορφή του παιχνιδιού RLGame έχει προστεθεί από την δημοσίευση [02]. 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 4
Πλατφόρμες Δημιουργίας Παιχνιδιών

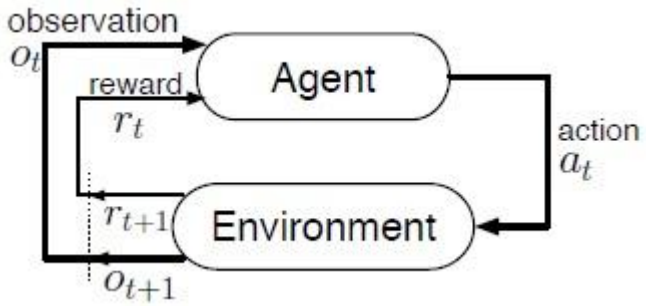
Σε αυτό το κεφάλαιο παραθέτεται η περιγραφή των πλατφορμών δημιουργίας παιχνιδιών και εκτέλεσης πειραμάτων ενισχυτική μάθησης που εξετάστηκαν στα πλαίσια της παρούσας διατριβής. Στο κεφάλαιο αυτό γίνεται αναφορά σε τέσσερεις πλατφόρμες/πρότυπα: RL-Glue, ORTS, GameMaker, General Game Player.

4.1 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 [10], 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

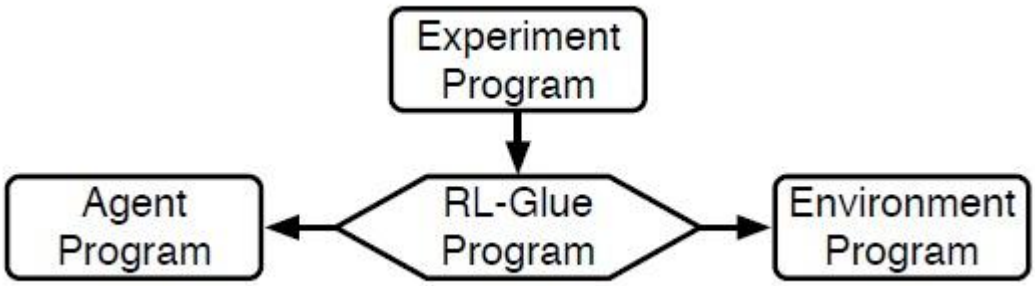
των μελλών της ερευνητικής κοινότητας. Έχει παρατηρηθεί δε ότι μερικές φορές είναι δύσκολο ή αδύνατο να αναπαράγει ακριβώς ένας ερευνητής το πείραμα ενός άλλου ερευνητή. Προσπαθώντας να καλύψει το κενό που υπάρχει στο πεδίο της προτυποποίησης των υλοποιήσεων των πειραμάτων ενισχυτικής μάθησης, η ερευνητική ομάδα του RL-Glue έχει δημιουργήσει ένα πρωτόκολλο επικοινωνίας μεταξύ των κύριων οντοτήτων που μετέχουν σε ένα πείραμα ενισχυτικής μάθησης.

Σε ένα πείραμα ενισχυτικής μάθησης το περιβάλλον (πείραμα και υποσύστημα εφαρμογής κανόνων που διέπουν το περιβάλλον) επικοινωνεί στον πράκτορα παρατηρήσεις της υπάρχουσας κατάστασης του περιβάλλοντος και αποδίδει ανταμοιβές ως αποτέλεσμα των ενεργειών που επιλέγει ο πράκτορας ενισχυτικής μάθησης. Η ροή αυτών των δράσεων αποδίδεται από το ακόλουθο σχεδιάγραμμα των Sutton & Barto [7]



Εικόνα 4.1 : Αλληλεπιδράσεις πράκτορα και περιβάλλοντος σε ένα σύστημα ενισχυτικής μάθησης[7,10]

Το πρωτόκολλο RL-GLue έχει προσπαθήσει να τυποποιήσει το σύστημα που περιγράφεται από το παραπάνω διάγραμμα. με στόχευση στα online προβλήματα ενισχυτικής μάθησης που αφορούν 1 πράκτορα. Το πρωτόκολλο έχει ορίσει 4 οντότητες : πείραμα, RL-Glue, περιβάλλον, πράκτορας και έχει περιγράψει πως αυτές οι οντότητες οργανώνονται σε προγράμματα και έναν τυποποιημένο τρόπο με τον οποίο επικοινωνούν/ αλληλεπιδρούν[10]. Στο ακόλουθο σχεδιάγραμμα φαίνονται οι αλληλεπιδράσεις των 4 οντοτήτων:



Εικόνα 4.2 : Αρχιτεκτονική RL-Glue[10]

Το πρόγραμμα του πειράματος είναι υπεύθυνο για τον τρόπο εκτέλεσης του πειράματος. Το πρόγραμμα του περιβάλλοντος είναι υπεύθυνο για την εκτέλεση των επιλεγμένων ενεργειών του πράκτορα, δημιουργεί τις παρατηρήσεις του περιβάλλοντος και αποδίδει ανταμοιβές ως αποτέλεσμα των επιλογών του πράκτορα. Το πρόγραμμα του πράκτορα αποτελεί την υλοποίηση του αλγόριθμου μάθησης και του μηχανισμού επιλογής ενεργειών. Το πρόγραμμα RL-Glue λαμβάνει εντολές από το πρόγραμμα του πειράματος και ενεργεί ως διαμεσολαβητής της επικοινωνίας ανάμεσα στο πρόγραμμα του πράκτορα και στο πρόγραμμα περιβάλλοντος.

Το RL-Glue υποστηρίζει τη συνέργεια των 4 οντοτήτων σε ένα πρόγραμμα και την επικοινωνία μεταξύ τους μέσω κλήσεων συναρτήσεων (εσωτερική αρχιτεκτονική - internal mode) ή την ύπαρξη 4 ξεχωριστών προγραμμάτων : πείραμα, περιβάλλον, πράκτορας και RL-Glue πρόγραμμα που μπορεί να εκτελούνται στον ίδιον ή σε διαφορετικούς υπολογιστές. Σε αυτή την εξωτερική αρχιτεκτονική (external mode) τα 3 πρώτα προγράμματα ανοίγουν συνδέσεις TCP/IP (socket connections) με το RL-Glue πρόγραμμα και ανταλλάσουν μηνύματα με αυτό. Η εσωτερική αρχιτεκτονική υποστηρίζεται από τις υλοποιήσεις προγραμμάτων πειράματος, περιβάλλοντος, και πράκτορα σε Java και C/C++, ενώ η εξωτερική αρχιτεκτονική υποστηρίζεται από τις υλοποιήσεις σε C/C++, Java, Python, Lisp, and Matlab. Το RL-Glue παρέχει υλοποιήσεις προγραμμάτων πελατών που υλοποιούν τα κομμάτια της επικοινωνίας για όλες τις γλώσσες προγραμματισμού που προαναφέραμε. Ακόμη ορίζει τις μεθόδους που έχει το κάθε πρόγραμμα οπότε ο ερευνητής ασχολείται μόνο με το να κωδικοποιήσει την λογική της κάθε οντότητας στη γλώσσα προγραμματισμού που γνωρίζει, χωρίς να τον απασχολούν οι λεπτομέρειες της επικοινωνίας μεταξύ των υποσυστημάτων.

Από την πλευρά του ερευνητή που θέλει να δοκιμάσει την αποτελεσματικότητα του πράκτορα μάθησης που έχει υλοποιήσει, με διάφορες υλοποιήσεις περιβάλλοντος και πειραμάτων που

μπορεί να είναι κωδικοποιημένα σε άλλες γλώσσες, τα οφέλη που μπορεί να έχει από το RL-Glue είναι προφανή. Βέβαια για να έχει νόημα μια τέτοια επένδυση σε χρόνο θα πρέπει να υπάρχει ένας ικανός αριθμός ερευνητών που θα επιλέξουν να βασίσουν τις υλοποιήσεις τους στο RL-Glue. Το γεγονός ότι υπάρχουν δημοσιεύσεις, κοινότητα χρηστών, βιβλιοθήκη προγραμμάτων για το RL-Glue, αλλά και διαγωνισμοί που έχουν πραγματοποιηθεί με το RL-Glue, συνηγορεί ότι η ερευνητική κοινότητα αντιλαμβάνεται τις ανάγκες και εξετάζει την πρόταση του RL-Glue.

Το RL-Glue θέτει περιορισμούς στον τρόπο επικοινωνίας των προγραμμάτων είτε λόγω του γεγονότος ότι όλη η επικοινωνία περνάει μέσω του RL-Glue προγράμματος είτε λόγω της επιλογής των δομών οι οποίες μπορούν να περάσουν στα μηνύματα. Γενικά χρησιμοποιούνται πίνακες απλών τύπων όπως int, double, char και ενώ στις περισσότερες περιπτώσεις των προβλημάτων οι δομές αυτές μπορούν να αποτυπώσουν την πλήρη πληροφορία, η αλλαγή υπαρχόντων προγραμμάτων με σκοπό να γίνουν συμβατά με το RL-Glue μπορεί να δυσκολέψει τον ερευνητή.

Για τις ανάγκες της παρούσας διπλωματικής διατριβής πειραματίστηκα με το RLGlue 3.0. Στην έκδοση αυτή τα προγράμματα πελάτες (codecs) που είναι υλοποιημένα σε διάφορες γλώσσες προγραμματισμού διαχωρίστηκαν από το κύριο RL-Glue project που είναι υλοποιημένο σε C/C++. Ακόμη στην έκδοση 3.0, το 2008, το project αναβαθμίστηκε από ένα περιφερειακό έργο του πανεπιστημίου Alberta σε ένα πρόγραμμα ανοιχτού κώδικα με τη φιλική για τους ερευνητές άδεια Apache 2.0.

Για τις ανάγκες της έρευνας μου χρησιμοποίησα τον Java codec. Η λογική της οργάνωσης των επιμέρους προγραμμάτων του RL-Glue είναι σχετικά απλή : Το προγράμματα περιβάλλοντος και του πράκτορα υλοποιούν συγκεκριμένα java interfaces και ο ερευνητής καλείται να αποτυπώσει τη λογική της οντότητας σε κάποιες προσχεδιασμένες μεθόδους (με σαφείς εισόδους και εξόδους). Για παράδειγμα στο περιβάλλον, σημαντικές μέθοδοι είναι η env_start και env_step μέθοδοι. Η πρώτη καλείται από το RL-Glue στην αρχή ενός επεισοδίου και επιστρέφει την παρατήρηση της τρέχουσας κατάστασης του περιβάλλοντος. Η δεύτερη καλείται ξανά από το RL_Glue και τροφοδοτείται με την επιλογή ενέργειας του πράκτορα, κατόπιν επιστρέφει την παρατήρηση της τρέχουσας κατάστασης του περιβάλλοντος, το reward που αποδίδεται από το περιβάλλον για την ενέργεια που επιλέχθηκε και μία σημαία για το εάν φτάσαμε σε τελική κατάσταση. Στον πράκτορα σημαντικές μέθοδοι είναι η agent_start και agent_step και agent_end μέθοδοι. Η πρώτη καλείται από το RL-Glue και τροφοδοτείται μόνο με την πρώτη παρατήρηση του περιβάλλοντος, Η μέθοδος επιστρέφει την επόμενη ενέργεια Η δεύτερη καλείται ξανά από το

RL_Glue και τροφοδοτείται με την παρατήρηση της τρέχουσας κατάστασης του περιβάλλοντος, και την ανταμοιβή που αποδίδεται από το περιβάλλον για την προηγούμενη ενέργεια που επιλέχθηκε. Η μέθοδος επιστρέφει την ενέργεια που έχει επιλέξει ο αλγόριθμος. Τέλος η τρίτη καλείται μόνο εάν φτάσουμε σε τελική κατάσταση και το μοναδικό της όρισμα είναι η ανταμοιβή του περιβάλλοντος.

Το πρόγραμμα του πειράματος εκκινεί το πείραμα, ορίζει τις λεπτομέρειες εκτέλεσης του πειράματος, αναλύει το αποτέλεσμα του πειράματος και τερματίζει το πείραμα. Για το σκοπό αυτό καλεί συγκεκριμένες στατικές μεθόδους του RL-Glue:

- RL_init : αρχικοποιεί τον πράκτορα και το περιβάλλον

- RL_start: καλεί τις agent_start & env_start μεθόδους

- RL_step : καλεί την env_step μέθοδο με όρισμα την επιλεγμένη από τον πράκτορα ενέργεια, η μέθοδος επιστρέφει την παρατήρηση, ανταμοιβή και σημαία τερματισμού. Στην περίπτωση μη τερματικής συνθήκης καλείται η agent_step μέθοδος με ορίσματα την παρατήρηση και την ανταμοιβή που επέστρεψε η env_step. Στην περίπτωση τερματικής συνθήκης η agent_end μέθοδος καλείται με όρισμα την ανταμοιβή.

- RL_episode : Στον αντίποδα της μεθόδου RL_step όπου το πρόγραμμα του πειράματος έχει πρόσβαση σε όλα τα δεδομένα που παράγονται στη διάρκεια της εκτέλεση του πειράματος, το RL-Glue προσφέρει την RL_episode μέθοδο για τις περιπτώσεις που δεν χρειάζεται όλη αυτή η πληροφορία σε επίπεδο βήματος. Η RL_episode μέθοδος εκτελεί τα βήματα και επιστρέφει τον έλεγχο στο πείραμα. Κατόπιν το πείραμα μπορεί να λάβει τον αριθμό των συνολικών βημάτων και της συνολικής ανταμοιβής μέσω των στατικών μεθόδων RL_num_steps και RL_return. –

Η κύρια δυσκολία που συνάντησα με το RL-Glue είναι ότι δεν έχει φτιαχτεί για να υποστηρίζει πειράματα με πολλαπλούς πράκτορες που συναγωνίζονται με σκοπό τη νίκη σε ένα παιχνίδι στρατηγικής μηδενικού αθροίσματος όπου οι παίκτες/πράκτορες παίζουν εκ περιτροπής. Είναι σαφές τόσο στην δημοσίευση όσο και στο FAQ ότι το RL-Glue έχει σχεδιαστεί να υποστηρίζει πειράματα που εμπλέκουν ένα και μόνο πράκτορα[10, 22, 23].

Συνεπώς στην περίπτωση που ερευνώ δεν μπορώ να χρησιμοποιήσω το πρωτόκολλο RL-Glue με τον τρόπο που προτείνουν οι δημιουργοί του και προσπάθησα να βρω τρόπους να παρακάμψω τη βασική ροή που υποστηρίζει το RL-Glue. Για το σκοπό αυτό προχώρησα σε μια πιλοτική υλοποίηση του παιχνιδιού στρατηγικής connect4 χρησιμοποιώντας το RL-Glue java codec. Μπόρεσα να εξομοιώσω το παιχνίδι μεταξύ 2 αντιπάλων κάνοντας τις ακόλουθες παραδοχές: έπρεπε στο πρόγραμμα του περιβάλλοντος να τηρώ τη σειρά επιλογής και να επιστρέφω τη σειρά στα πλαίσια της παρατήρησης που πέρναγα στο RL-Glue. Το πρόγραμμα πράκτορας έπρεπε να αναγνωρίζει τη σειρά επιλογής και να εκτελεί τον αλγόριθμο επιλογής κίνησης, εκ περιτροπής για τον κάθε ένα παίκτη. Στην πιλοτική εφαρμογή χρησιμοποίησα την υλοποίηση του παιχνιδιού connect4 αντί του RLGame γιατί ήταν πιο εύκολο να προσαρμόσω την επικοινωνία της κατάστασης/παρατήρησης του παιχνιδιού στις δομές που επιτάσσει το RL-Glue (πίνακες int, double,char). Ο λόγος ήταν ότι οι δομές που χρειάζονται για την αποτύπωση των καταστάσεων του connect4 είναι έτσι και αλλιώς πιο λιτές από αυτές του RLGame ενώ το connect4 το προγραμμάτιζα από την αρχή και έτσι ήταν πιο εύκολη η προσαρμογή του. Στη μεριά του πράκτορα χρησιμοποίησα το υποσύστημα της ενισχυτική μάθησης του RLGame. Το υποσύστημα βασιζόταν στην έκδοση alpha (Ιανουάριος 2012, λεπτομέρειες στο κεφάλαιο 4) και χρειάστηκε να τροποποιηθεί περαιτέρω για να προσαρμοστεί στις δομές που επιτάσσει το RL-Glue.

Το συμπέρασμα ήταν ότι η παράκαμψη που ακολούθησα με αφήνει να παίζω τα παιχνίδια με 2 παίκτες αλλά αναγκαστικά προϋποθέτει ότι τα προγράμματα του πειράματος, περιβάλλοντος και πράκτορα έχουν φτιαχτεί πολύ συγκεκριμένα για να υποστηρίξουν την παράκαμψη του πρωτοκόλλου που εφάρμοσα. Ακόμη στο επίπεδο του RL-Glue προγράμματος που δρομολογεί όλη την επικοινωνία, οι υπηρεσίες που δίνονται όπως μετρητές του αριθμού των βημάτων ή του συνόλου των ανταμοιβών υπολογίζονται αθροιστικά και για τους 2 παίκτες. Συνεπώς πρέπει να μετράω τις κινήσεις ανά παίκτη ή κάποιο στατιστικό σχετικό με τις ανταμοιβές του παίκτη στο επίπεδο του πειράματος χρησιμοποιώντας κλήσεις της RL_step.

Το βασικό πλεονέκτημα του RL_Glue είναι ότι δίνει υπηρεσίες έτσι ώστε διαφορετικές υλοποιήσεις προγραμμάτων περιβάλλοντος και πρακτόρων που ακολουθούν το RL-Glue πρωτόκολλο να μπορούν εύκολα να συνεργαστούν στο ίδιο πείραμα. Στην περίπτωση που είμαστε διατεθειμένοι να γράψουμε όλα τα προγράμματα με τον τρόπο που περιγράφηκε προηγουμένως μπορούμε να εκτελέσουμε το πείραμα μάθησης του connect4 πάνω από το RL-Glue πρωτόκολλο. Παρόλα αυτά η παρέκκλιση που πρέπει να κάνουμε από τη βασική ροή του πρωτοκόλλου όπως και το γεγονός ότι δεν υπάρχουν υλοποιήσεις παιχνιδιών στρατηγικής 2

παικτών στο RL-Glue συνεπάγεται ότι δεν μπορούμε να εκμεταλλευτούμε τα πλεονεκτήματα του RL-Glue και να δοκιμάσουμε τον μηχανισμό μάθησης με υπάρχουσες υλοποιήσεις παιχνιδιών στρατηγικής.

Η δική μου εμπειρία με το RL-Glue είναι θετική καθώς η λιτή και ξεκάθαρη αρχιτεκτονική του μου έδωσε ιδέες για την προσέγγιση που ακολούθησα μετέπειτα στον τελικό διαχωρισμό του μηχανισμού μάθησης από το RLGame. Θεωρώ όμως με βάση τα στοιχεία που παρέθεσα παραπάνω ότι είναι προτιμότερο να μην βασίσω την υλοποίηση της νέας αρχιτεκτονικής του RLGame στο πρωτόκολλο RL-Glue.

4.2 ORTS

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.

Το ORTS είναι υλοποιημένο σε γλώσσα προγραμματισμού C++ και έχει τα ακόλουθα υποσυστήματα :

- Το υποσύστημα του διακομιστή όπου ο χρήστης δίνει τον ορισμό του παιχνιδιού μέσω scripts που περιγράφουν τα δομικά στοιχεία του παιχνιδιού και τις αλληλεπιδράσεις μεταξύ των αντικειμένων του παιχνιδιού. Ο διακομιστής στέλνει στοιχεία για την κατάσταση που είναι ορατή στον παίκτη και λαμβάνει και εκτελεί ενέργειες από όλους του μετέχοντες παίκτες.

- Το υποσύστημα του πελάτη που συνδέεται στον διακομιστή και μέσω του οποίου δημιουργείται η δράση του παιχνιδιού. Υπάρχει η δυνατότητα γραφικού περιβάλλοντος στον πελάτη (όταν ένας άνθρωπος παίζει το παιχνίδι), ή ενός υπολογιστικού παίκτη που αναπαριστά εσωτερικά την κατάσταση του παιχνιδιού.

Τόσο ο προσανατολισμός του ORTS στα παιχνίδια στρατηγικής πραγματικού χρόνου, όσο και το γεγονός ότι η σχετική με αυτό προσπάθεια φαίνεται να βρίσκεται σε ύφεση (δεν υπάρχουν πρόσφατες δημοσιεύσεις, δεν υπάρχει πρόσφατη ανανέωση της έκδοσης), με οδήγησαν στην απόφαση ότι το ORTS δεν είναι ιδανική, προς το παρόν επιλογή για ολοκλήρωση με το RLGame.

4.3 GameMaker

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 οποιοδήποτε browser μπορεί να υποστηρίξει HTML5 [12,13].

Το βασικό πλεονέκτημα του GameMaker είναι το γραφικό περιβάλλον προγραμματισμού που παρέχει. Οι σχεδιαστές του έχουν μεριμνήσει έτσι ώστε να κατηγοριοποιήσουν και να ορίσουν τις δομικές κλάσεις ενός παιχνιδιού και να ορίσουν και τα πιθανά γεγονότα για κάθε κλάση αντικειμένων. Για κάθε ορισμένο γεγονός, ο προγραμματιστής μπορεί να ορίσει τις ενέργειες που επιθυμεί. Με αυτό τον τρόπο ο χρήστης μπορεί να αρχίσει να δημιουργεί τη ροή του παιχνιδιού του ορίζοντας τα δικά του αντικείμενα και να αποδώσει στα αντικείμενα συμπεριφορές επιλέγοντας για παράδειγμα τον τρόπο που θα κινείται το αντικείμενο από τις διαθέσιμες επιλογές μετακίνησης ή ορίζοντας μια ενέργεια που θα λάβει χώρα σε περίπτωση κάποιου συγκεκριμένου γεγονότος (π.χ. σύγκρουση με άλλο αντικείμενο).

Το GameMaker έρχεται με μια ενσωματωμένη scripting γλώσσα την GML. Ο χρήστης μπορεί να κωδικοποιήσει την λογική του παιχνιδιού σε αυτή την γλώσσα (συνήθως ο κώδικας αυτός γράφεται στις μεθόδους που έχουν ανατεθεί από το GameMaker στα γεγονότα).

Μία μεγάλη γκάμα παιχνιδιών έχει αναπτυχθεί στο GameMaker όπως παιχνίδια δράσης, παιχνίδια πρώτου προσώπου κ.λ.π.

Στα πλαίσια της παρούσας διπλωματικής διατριβής εξετάστηκε το κατά πόσο είναι εφικτό να δημιουργηθεί έκδοση του RLGame με το GameMaker. Για το κομμάτι του παιχνιδιού το γεγονός ότι η λογική πρέπει να αποτυπωθεί σε κώδικα GML δημιουργεί ένα δίλλημα για τον εάν πρέπει να επενδυθεί χρόνος στην εξοικείωση με μια scripting γλώσσα η οποία χρησιμοποιείται μόνο σε ένα λογισμικό κλειστού κώδικα. Για το κομμάτι μάθησης. η αποτύπωση σε GML δεν μπορεί να είναι αποδεκτή επιλογή λόγω της πολυπλοκότητας του κώδικα και την ανάγκη για εισόδους εξόδους από αρχεία. Λαμβάνοντας υπόψη ότι η υπάρχουσα υλοποίηση του υποσυστήματος μάθησης έχει βασιστεί στη Java, για να μπορέσω να χρησιμοποιήσω το υποσύστημα της ενισχυτικής μάθησης, πρέπει να χρησιμοποιήσω τον extension μηχανισμό του GameMaker. Σε αυτό το σενάριο θα έπρεπε να και να μπορέσω να δημιουργήσω ένα Windows DLL (Dynamic Link Library) το οποίο θα χρησιμοποιείται από τον κώδικα του GameMaker. Αυτό το dll μπορεί να χρησιμοποιεί μέσω JNI (Java Native Interface) τις μεθόδους του μηχανισμού μάθησης ή το dll θα ανοίγει TCP συνδέσεις (sockets) με τον μηχανισμό μάθησης και θα ανταλλάσει μηνύματα. Η δεύτερη επιλογή συνεπάγεται ότι στην υλοποίηση του μηχανισμού μάθησης θα πρέπει να προστεθεί η λειτουργικότητα της επικοινωνίας μέσω μηνυμάτων.

Για να συνοψίσω τα συμπεράσματα για την εφικτότητα της ολοκλήρωσης του RLGame με το GameMaker το υποσύστημα μηχανισμού κινήσεων και οι κανόνες του παιχνιδιού θα μπορούσαν να κωδικοποιηθούν σε GML. Η χρήση όμως του extension μηχανισμού ο οποίος κάνει δυνατή τη χρήση Native interfaces (dll) είναι διαθέσιμη μόνο στην standard έκδοση (η οποία διατίθεται κατόπιν πληρωμής άδειας χρήσης). Τέλος το γεγονός ότι το λογισμικό είναι κλειστού κώδικα, και τρέχει μόνο σε κάποια λειτουργικά συστήματα αποτελεί μειονέκτημα στην επιλογή του ως εργαλείου για κάποιο ερευνητικό έργο.

4.4 General Game Player

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[11]. 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. Τα παιχνίδια που είναι αποδεκτά για το GGP πρέπει να έχουν τις ακόλουθες ιδιότητες :

- πεπερασμένο σύνολο καταστάσεων
- η δράση να διαδραματίζεται σύγχρονα (υπάρχει η έννοια του χρόνου και της ορισμένης περιόδου που έχει ο κάθε παίκτης για να αποφασίσει την επόμενη ενέργεια)
- προκαθορισμένο αριθμό παικτών
- κάθε παίκτης έχει έναν πεπερασμένο αριθμό πιθανών κινήσεων σε κάθε βήμα
- κάθε τερματική κατάσταση έχει μια συγκεκριμένη τιμή στόχο για κάθε παίκτη

Όπως αναφέρθηκε και στον ορισμό η GDL (Game Definition Language) είναι η γλώσσα λογικής που χρησιμοποιείται για την περιγραφή των πεπερασμένων καταστάσεων και των κανόνων των παιχνιδιών του GGP.

Τα συστήματα General Game Playing αποτελούνται από:

- Κάποιον GameMaster που είναι ειδικοί διακομιστές που χρησιμοποιούνται για να διαθέτουν τις περιγραφές των παιχνιδιών να τηρούν κεντρικά την επίσημη τρέχουσα κατάσταση του παιχνιδιού, να συλλέγουν τις κινήσεις των παικτών να ελέγχουν την ορθότητα των κινήσεων και να αποφασίζουν τον/τους νικητές. Οι διακομιστές χρησιμοποιούνται για την ανάπτυξη και το έλεγχο νέων παικτών από τα μέλη της GGP κοινότητας όπως και στους ετήσιους GGP διαγωνισμούς που προκηρύσσει ο οοργανισμός AAAI.

- Τους υπολογιστικούς παίκτες (General Game Players) που πρέπει να είναι σε θέση να διαβάσουν την περιγραφή οποιουδήποτε παιχνιδιού σε γλώσσα λογικής GDL και να χρησιμοποιήσουν αυτή τη γνώση για να παίξουν το παιχνίδι ανταγωνιστικά, να υλοποιούν το πρωτόκολλο που έχει περιγραφεί για επικοινωνία με κάποιον GameMaster, να διατηρούν τοπικά την κατάσταση του παιχνιδιού έχοντας σαν input την περιγραφή του παιχνιδιού τις κινήσεις που επιλέγουν και τις κινήσεις του/των αντιπάλων ή το μήνυμα τερματισμού που επικοινωνεί ο GameMaster.

Κατά συνέπεια ένας τέτοιος υπολογιστικός παίκτης δεν μπορεί και δεν έχει νόημα να έχει λογική και αλγορίθμους που είναι προσαρμοσμένοι στις ανάγκες ενός συγκεκριμένου παιχνιδιού. Ο υπολογιστικός παίκτης πρέπει να μπορεί να προσαρμόζει την τακτική του ανάλογα με το παιχνίδι που καλείται να παίξει και που η μόνη πληροφορία που έχει για αυτό προέρχεται από την περιγραφή του παιχνιδιού.

Στο κέντρο του ενδιαφέροντος των ερευνητών που ασχολούνται με το GGP είναι η εφαρμογή και η ολοκλήρωση ποικίλων τεχνολογιών του τομέα της τεχνητής νοημοσύνης όπως αναπαράσταση γνώσης, λογική, μάθηση, λήψη αποφάσεων.

Για την παρούσα διατριβή και το σύστημα διεξαγωγής πειραμάτων του RLGame το GGP έχει ενδιαφέρον σε δύο βασικούς άξονες :

- Την εφικτότητα αποτύπωσης της λογικής του παιχνιδιού σε γλώσσα GDL και την δημοσίευση του παιχνιδιού σε κάποιο GGP GameMaster.

- Την εφικτότητα δημιουργίας ενός General Game Player ο οποίος θα μπορεί να χρησιμοποιήσει τον υπάρχον μηχανισμό ενισχυτικής μάθησης.

Για το πρώτο ζητούμενο το RLGame καλύπτει όλες τις απαιτήσεις που θέτονται για τα GGP παιχνίδια. Οπότε το μόνο που χρειάζεται είναι η περιγραφή του παιχνιδιού στην γλώσσα λογικής GDL. Για το δεύτερο ζητούμενο, οι ιδιότητες του μηχανισμού ενισχυτικής μάθησης (δεν έχει κάποιο αλγόριθμο για κάποιο συγκεκριμένο παιχνίδι, η γνώση αποκτάται από την εμπειρία) του RLGame τον κατατάσσουν σαν μια πιθανή επιλογή για το υποσύστημα μάθησης και επιλογής κινήσεων ενός GGP παίκτη. Παρόλα αυτά το υποσύστημα μάθησης είναι ένα μόνο μέρος των ιδιοτήτων που πρέπει να έχει ένας GGP παίκτης. Όπως αναφέρθηκε και παραπάνω ένας GGP παίκτης πρέπει να έχει την δυνατότητα να χρησιμοποιεί την GDL περιγραφή του παιχνιδιού για να αντιλαμβάνεται τους κανόνες του παιχνιδιού, την αρχική κατάσταση, την επόμενη κατάσταση που προκύπτει μετά από μια κίνηση του ή μετά από μια κίνηση αντιπάλου κ.λ.π.

Για την παρούσα διατριβή που ο στόχος της είναι η τεχνική συντήρηση του συστήματος RLGame, κρίθηκε ότι δεν ήταν σκόπιμη η περαιτέρω διερεύνηση κάποιου από τα δύο σημεία ενδιαφέροντος σχετικά με το RLGame και το GGP.

Κεφάλαιο 5
Στάδια Επανασχεδίασης και Υλοποίησης

Σε αυτό το κεφάλαιο γίνεται η προσπάθεια να αποδοθεί η πορεία της διαδικασίας επανασχεδίασης και αναδιάταξης των υλοποιήσεων των προηγούμενων ερευνητικών προσπαθειών του συστήματος παιχνιδιού RLGame.

5.1 Εισαγωγή

Η εργασία αναδιάταξης και επανασχεδιασμού των υλοποιήσεων του συστήματος παιχνιδιού RLGame είναι μία δουλειά που πραγματοποιήθηκε σε στάδια. Για να προχωρήσω στα στάδια της υλοποίησης έπρεπε να εκτελέσω αρχικά στάδια αναδιάταξης της υπάρχουσας υλοποίησης με σκοπό όχι την αλλαγή της αρχιτεκτονικής αλλά την καλύτερη κατανόηση από μέρους μου της υπάρχουσας υλοποίησης και των εννοιών. Η διαδικασία που ακολούθησα για να κατανοήσω το αρχικό μοντέλο μοιάζει πολύ με τις τακτικές «refactor to understand» (αναδιάταξη κώδικα με σκοπό την καλύτερη κατανόηση) και «step through the execution» (κατανόηση της αλληλεπίδρασης των αντικειμένων ενός συστήματος μέσω ελέγχου σε πρόγραμμα εντοπισμού σφαλμάτων) που περιγράφει το σύγγραμμα [26]. Η τελική αρχιτεκτονική προέκυψε στην

πορεία αυτών των 4 σταδίων που περιγράφω σε αυτό το κεφάλαιο. Κάθε στάδιο χωριζόταν σε επιμέρους φάσεις και ο στόχος ήταν να υπάρχει πάντα μια τρέχουσα λειτουργική έκδοση. Στο τέλος δε κάθε σταδίου γινόταν λεπτομερής έλεγχος και αποτίμηση του αποτελέσματος και της προσπάθειας και προσαρμοζόντουσαν επιμέρους ζητούμενα/απαιτήσεις.

Στα ακόλουθα υπο-τμήματα περιγράφω τα βασικά στάδια της υλοποίησης.

5.1.1 Υπάρχουσες υλοποιήσεις και απαιτήσεις νέας έκδοσης

Τα ζητούμενα της παρούσας διπλωματικής διατριβής είναι η δημιουργία μίας έκδοσης του παιχνιδιού RLGame όπου ο μηχανισμός μάθησης θα είναι διαχωρισμένος από το υπόλοιπο παιχνίδι, η μελέτη σκοπιμότητας ολοκλήρωσης του παιχνιδιού με πλατφόρμες δημιουργίας παιχνιδιών και η ολοκλήρωση του μηχανισμού με άλλο παιχνίδι.

Το σύστημα διεξαγωγής πειραμάτων του RLGame έχει ήδη 3 διαφορετικές εκδόσεις για παιχνίδια μεταξύ υπολογιστικών παικτών που χρησιμοποιούν τεχνικές μηχανικής μάθησης, παιχνίδια μεταξύ υλοποίησης του αλγόριθμου minimax και υπολογιστικού παίκτη που χρησιμοποιεί τεχνικές μηχανικής μάθησης και τέλος παιχνίδια που παίζει ένας άνθρωπος εναντίον υπολογιστικού παίκτη που χρησιμοποιεί τεχνικές μηχανικής μάθησης. Η αρχική απαίτηση ήταν αυτές να μεταφερθούν στην νέα αρχιτεκτονική.

Στην πορεία της εργασίας προέκυψε ότι οι δύο πρώτες εκδόσεις μπορούσαν να ενοποιηθούν σε ένα πρόγραμμα ενώ επιλέχθηκε να μην προχωρήσει η αναβάθμιση της έκδοσης παιχνιδιού άνθρωπου εναντίον υπολογιστή καθώς η desktop υλοποίηση του γραφικού περιβάλλοντος ήταν πολύ παλιά ενώ νεότερη διαδικτυακή έκδοση που έχει ήδη υλοποιηθεί και δεν ήταν στα αντικείμενα της παρούσας διπλωματικής διατριβής μπορεί να τροποποιηθεί στο μέλλον και να χρησιμοποιήσει τον διαχωρισμένο μηχανισμό μάθησης.

Στα πλαίσια της μελέτης σκοπιμότητας ολοκλήρωσης του παιχνιδιού με πλατφόρμες δημιουργίας παιχνιδιών αποφασίστηκε να προχωρήσουμε σε μια πιλοτική έκδοση του μηχανισμού μάθησης και του παιχνιδιού connect4 με το RL-Glue. Το αποτέλεσμα της διερεύνησης ήταν ότι δεν ήταν σκόπιμο να βασιστεί η υλοποίηση του συστήματος RLGame που παραδόθηκε στα πλαίσια της παρούσας διπλωματικής διατριβής στο RL-Glue.

Τέλος για τον στόχο της ολοκλήρωση του μηχανισμού με άλλο παιχνίδι προχωρήσαμε στην υλοποίηση του connect4 με τον μηχανισμό μάθησης και με τον αλγόριθμο minimax.

Στην πορεία των επιμέρους παραδοτέων προέκυπτε η δυνατότητα για βελτιώσεις στο πρόγραμμα όπως δυνατότητα ενοποίησης των εκδόσεων υπολογιστή εναντίον υπολογιστή και αλγόριθμου minimax εναντίον υπολογιστή και η επιλογή της μορφής του παιχνιδιού μέσω ρυθμίσεων. Παράλληλα με αυτές τις βελτιώσεις προέκυπταν ανάγκες για χρήσιμες προσθήκες όπως η δυνατότητα για παιχνίδι μεταξύ 2 minimax παικτών. Έγινε προσπάθεια οι περισσότερες από αυτές τις βελτιώσεις να ενσωματωθούν στα προγράμματα.

5.1.2 Αρχιτεκτονική υπάρχουσας υλοποίησης

Σε αυτή την παράγραφο παραθέτονται κάποια γνωρίσματα της αρχιτεκτονικής των υλοποιήσεων του RLGame:

- οι κλάσεις έχουν ορισθεί στο προεπιλεγμένο κενό πακέτο

- Στην κλάση του παίκτη και της κατάστασης του παιχνιδιού επαναλαμβάνονται μέθοδοι που εκτελούν τις ίδιες ενέργειες για τον λευκό και μαύρο παίκτη

- Η λογική του μηχανισμού μάθησης είναι αναμεμιγμένη με την λογική του παιχνιδιού στις κλάσεις του παίκτη και της κατάστασης του παιχνιδιού ενώ στην κλάση του Νευρωνικού υπάρχει η μέθοδος που προετοιμάζει την είσοδο του νευρωνικού και χρησιμοποιεί τις καθολικές μεταβλητές του προγράμματος αλλά και τις κλάσεις του πιονιού και του τετραγώνου της σκακιέρας.

Στην παρούσα μορφή η υλοποίηση παρεκκλίνει από βασικές αρχές του αντικειμενοστραφούς προγραμματισμού όπως φαίνεται στο δεύτερο σημείο που αναφέρθηκε στην παραπάνω λίστα ενώ η λογική του μηχανισμού μάθησης είναι δεμένη με την λογική του παιχνιδιού.

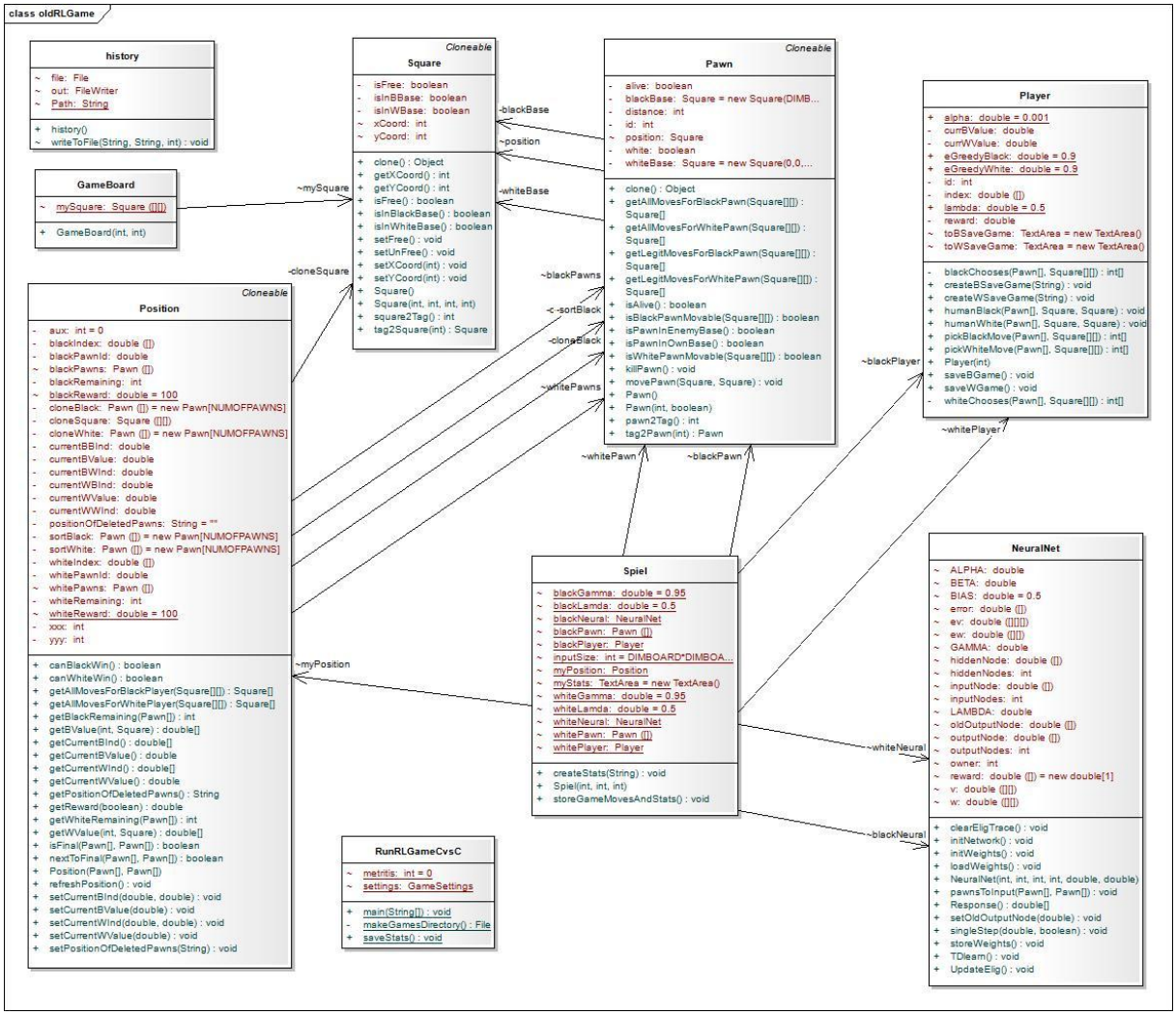
Εκτός της δομής του κώδικα με δυσκόλεψε και η έλλειψη σχολίων που θα με βοηθούσαν να κατανοήσω καλύτερα τον κώδικα και τις επιλογές που είχαν γίνει από τους μηχανικούς που είχαν ασχοληθεί με την υλοποίηση. Βέβαια αυτή ήταν μία αναμενόμενη δυσκολία. Η παρούσα υλοποίηση υφίσταται για περισσότερα από 10 χρόνια ενώ αλλαγές έχουν ενσωματωθεί

κατά τη διάρκεια αυτών των 10 χρόνων από αρκετούς σπουδαστές/ερευνητές γεγονός που έχει οδηγήσει σε πιο σύνθετη/περίπλοκη και δυσκολότερη στη συντήρηση κωδικοποίηση.

Στον ακόλουθο πίνακα ακολουθεί μία αναφορά στις κλάσεις της έκδοσης υπολογιστή εναντίον υπολογιστή.

Κλάση	**Περιγραφή**
Common.java (Interface)	Περιλαμβάνει κάποιες καθολικές μεταβλητές που είναι προσπελάσιμες από τις άλλες κλάσεις του συστήματος.
GameBoard.java	Κλάση που αναπαριστά την σκακιέρα του παιχνιδιού.
GameSettings.java	Κλάση επιφορτισμένη με την συλλογή των ρυθμίσεων του πειράματος από γραφικό περιβάλλον
history.java	Κλάση επιφορτισμένη με την καταγραφή των εξαγόμενων αρχείων αποτελεσμάτων των παιχνιδιών και των κινήσεων των παικτών.
NeuralNet.java	Κλάση επιφορτισμένη με την δημιουργία του και διαχείριση ενός νευρωνικού δικτύου.
Pawn.java	Δημιουργία – διαχείριση πιονιών. Έχει ενσωματωμένη λογική/κανόνων κινήσεων
Player.java	Δημιουργία – διαχείριση παίκτη. Υλοποιεί μεγάλο κομμάτι της λογικής του πράκτορα της ενισχυτική μάθησης.
Position.java	Κλάση που είναι επιφορτισμένη με την διαχείριση της κατάστασης του παιχνιδιού. Κομμάτι της λογικής μηχανισμού μάθησης του παιχνιδιού βρίσκεται σε αυτή την κλάση.
RunRLGameCvsC.java	Η κλάση που εκτελεί το πείραμα.
Spiel.java	Κλάση δημιουργίας και εκτέλεσης ενός παιχνιδιού.
Square.java	Κλάση των τετραγώνων της σκακιέρας.

Στο ακόλουθο διάγραμμα κλάσεων από το οποίο έχουν απαλειφθεί το πρότυπο Common που υλοποιείται από σχεδόν όλες τις κλάσεις και η κλάση GameSettings παρουσιάζονται οι κλάσεις της εφαρμογής για το παιχνίδι RLGame υπολογιστή εναντίον υπολογιστή.



Εικόνα 5.1 : Διάγραμμα κλάσεων της έκδοσης υπολογιστή εναντίον υπολογιστή

5.1.3 Στάδιο ένα - Ιανουάριος 2012

Τον Ιανουάριο του 2012 παραδόθηκε η έκδοση alpha του RLGame όπου παρουσιάζεται η πρώτη εκδοχή της αναδιάταξης του RLGame με την λογική του παιχνιδιού και την υλοποίηση της Ενισχυτικής Μάθησης να αποτελούν ξεχωριστά τμήματα. Η έκδοση αφορούσε μόνο στη μορφή του παιχνιδιού υπολογιστή εναντίον υπολογιστή.

Η σχεδίαση και υλοποίηση της έκδοσης alpha του νέου συστήματος πραγματοποιήθηκε με την εφαρμογή τεχνικών συστηματικής αναδιάταξης του υπάρχοντος κώδικα (παράτημα Α1 όρος refactoring). Οι αλλαγές πραγματοποιήθηκαν σε διαδοχικά στάδια τα οποία ολοκληρωνόταν με την εκτέλεση πειραμάτων που είχαν στόχο την επιβεβαίωση της ορθότητας των αλλαγών.

Όπως αναφέρθηκε και στην ενδιάμεση αναφορά η έκδοση alpha έχει ελεγχθεί με περίπου 10000 χιλιάδες παιχνίδια υπολογιστή εναντίον υπολογιστή (5000 παιχνίδια ανά σετ νευρωνικών). Ο έλεγχος περιελάμβανε την επισκόπηση των κινήσεων επιλεγμένων παιχνιδιών στο playback εργαλείο που διαθέτει η ομάδα προγραμμάτων του RLGame. Ο έλεγχος έδειξε ότι δεν είχε προκληθεί κάποιο πρόβλημα στον μηχανισμό κινήσεων. Για τα αποτελέσματα της μάθησης ελέγχθηκαν τα αποτελέσματα των παιχνιδιών (νίκες ανά παίκτη και αριθμός κινήσεων). Στα αποτελέσματα εντοπιστήκαν στοιχεία κοινά με τα αντίστοιχα αποτελέσματα μάθησης της προηγούμενης υλοποίησης.

Κάποια βασικά γνωρίσματα της έκδοσης alpha αναφέρονται στην ακόλουθη λίστα:

- Εισαγωγή διακριτών πακέτων για την διάκριση των υποσυστημάτων της λογικής παιχνιδιού και του μηχανισμού μάθησης.

- Αναδιάταξη (Refactoring) βασικών κλάσεων που περιείχαν λογική του παιχνιδιού και του μηχανισμού μάθησης.

- Αναδιάταξη (Refactoring) κλάσεων ως προς την ονοματολογία, την εμβέλειας μεθόδων και ιδιοτήτων, και αλλαγή στατικών μεθόδων και ιδιοτήτων σε δυναμικές.

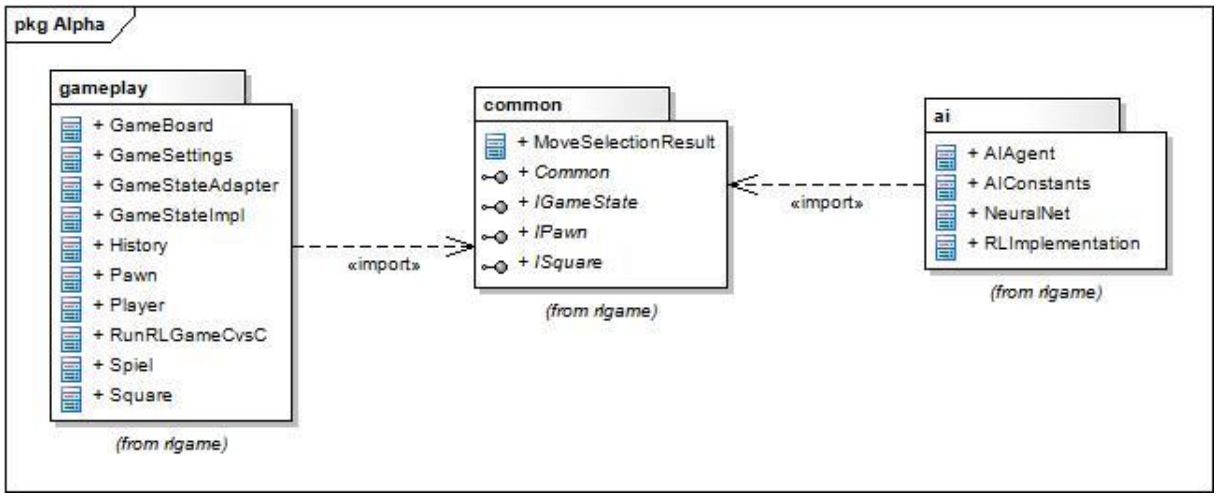
- Εισαγωγή προτύπων κλάσεων (Interfaces) και χρήση Προτύπων Σχεδίασης Λογισμικού (Design Patterns)΄

    - Χρήση του προτύπου σχεδίασης λογισμικού Adapter στην κλάση της κατάστασης του παιχνιδιού (παράρτημα Α1). Ουσιαστικά έχει ορισθεί ένα πρότυπο IGameState που χρησιμοποιεί τα IPawn και ISquare που είναι γνωστά στον πράκτορα και στο παιχνίδι. Τα πρότυπα αυτά ορίζουν μόνο τις δημόσιες μεθόδους που πρέπει να είναι διαθέσιμες στον μηχανισμό μάθησης. Στην μεριά του παιχνιδιού έχουν ορισθεί η κλάση GameStateImpl που χρησιμοποιεί το σύνολο των δημόσιων μεθόδων που ορίζονται στις κλάσεις Pawn και Square. Για να μπορέσει το υποσύστημα παιχνιδιού να περάσει το αντικείμενο της κλάσης GameStateImpl στον μηχανισμό μάθησης ορίζει την κλάση του αντάπτορα GameStateAdapter που υλοποιεί το πρότυπο IGameState. Ο αντάπτορας έχει ιδιωτική αναφορά στο αντικείμενο της κλάσης GameStateImpl και οι κλήσεις του μηχανισμού μάθησης προωθούνται στις μεθόδους του αντικειμένου αναφοράς

και οι απαντήσεις επιστρέφονται στον μηχανισμό μάθησης στη μορφή που περιμένει.

- Χρήση των προτύπων σχεδίασης λογισμικού, Singleton και Facade στην κλάση του πράκτορα Ενισχυτική Μάθησης (AIAgent) (παράρτημα Α1). Δηλαδή θα δημιουργηθεί μόνο ένα αντικείμενο από την κλάση του πράκτορα (Singleton) η οποία διαχειρίζεται το μηχανισμό μάθησης και για τους 2 παίκτες ενώ η κλάση λειτουργεί και σαν μια πρόσοψη (Facade) που κρύβει το γεγονός ότι οι μέθοδοι υλοποιούνται στην πραγματικότητα από τις κλάσεις RLImplementation και NeuralNet (όπως θα φανεί και στα σχεδιαγράμματα των εικόνων 5.2 και 5.3 οι κλάσεις RLImplementation και NeuralNet και οι μέθοδοι τους έχουν ορισθεί σαν δημόσιες, για πιο σωστή αποτύπωση του Facade προτύπου σχεδίασης θα έπρεπε να έχουν ορισθεί με εμβέλεια package).

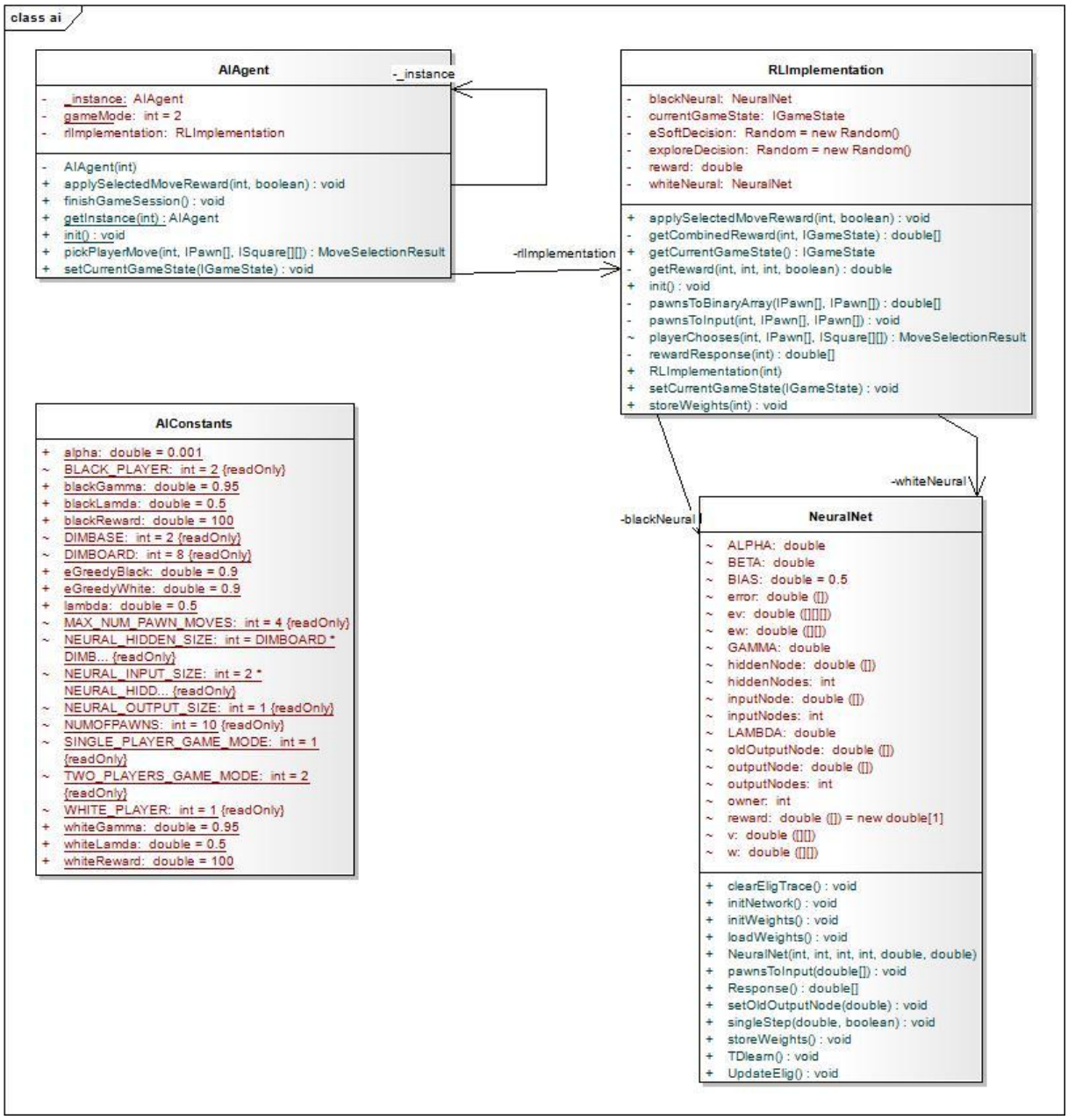
Στο σχεδιάγραμμα της εικόνας 5.2 παρουσιάζονται τα πακέτα της έκδοσης alpha. Όπως φαίνεται το πακέτο λογικής του παιχνιδιού και του μηχανισμού χρησιμοποιούν κάποιες κοινές κλάσεις και πρότυπα έτσι ώστε να μπορούν να αλληλεπιδρούν.



Εικόνα 5.2 : Γενική άποψη της έκδοσης Aplha του συστήματος (Διάγραμμα πακέτων προγράμματος)

Το παιχνίδι χρησιμοποιεί μόνο τις δημόσιες μεθόδους της κλάσης AIAgent στις οποίες και θέτει αντικείμενα που υλοποιούν το πρότυπο IGameState (κατάσταση παιχνιδιού η οποία με τη σειρά της αναφέρεται σε αντικείμενα που υλοποιούν τα πρότυπα IPawn και ISquare). Εν συνεχεία ο

μηχανισμός μάθησης χρησιμοποιεί δημόσιες μεθόδους που έχουν ορισθεί στα πρότυπα που αναφέρθηκαν.



Εικόνα 5.3 : Διάγραμμα κλάσεων του πακέτου μάθησης της έκδοσης Aplha του συστήματος

Ανοιχτά θέματα σε αυτή τη σχεδίαση ήταν ο περαιτέρω διαχωρισμός των υποσυστημάτων με σκοπό το υποσύστημα μάθησης να μην καλεί μεθόδους αντικειμένων του πακέτου λογικής παιχνιδιού καθώς και η επιλογή της οντότητας που θα εκτελούσε την μετατροπή της κατάστασης του παιχνιδιού στον δυαδικό πίνακα που θα χρησιμοποιούταν από το Νευρωνικό Δίκτυο ως είσοδος.

Σε αυτή την έκδοση ο μηχανισμός ήταν ακόμη πολύ δεμένος με το RLGame και δεν θα μπορούσε να χρησιμοποιηθεί σε άλλο παιχνίδι χωρίς περαιτέρω προγραμματισμό.

5.1.4 Στάδιο δύο – Μάρτιος 2012

Σε αυτό το στάδιο ασχολήθηκα με την ολοκλήρωση του μηχανισμού μάθησης με το connect4 και με πειραματική έκδοση Connect4 με το RL-Glue.

Μετά την παράδοση της έκδοσης Alpha διερευνήθηκε η σκοπιμότητα ολοκλήρωση του RLGame με το RL-Glue. Εξαιτίας περιορισμών στο πρωτόκολλο, προχώρησα σε μια πιλοτική υλοποίηση του παιχνιδιού στρατηγικής connect4 χρησιμοποιώντας το RL-Glue.

Στην έκδοση αυτή χρησιμοποιήθηκε το υποσύστημα της ενισχυτική μάθησης του RLGame. Το υποσύστημα βασιζόταν στην έκδοση alpha (Ιανουάριος 2012) και χρειάστηκε να τροποποιηθεί περαιτέρω για να προσαρμοστεί στις δομές που επιτάσσει το RL-Glue ενώ χρησιμοποιήθηκε η υλοποίηση του παιχνιδιού connect4 αντί του RLGame γιατί ήταν πιο εύκολο να προσαρμόσω την επικοινωνία της κατάστασης/παρατήρησης του παιχνιδιού στις δομές που επιτάσσει το RL-Glue (πίνακες int, double,char).

Παράλληλα με την έκδοση αυτή υλοποιούσα και την έκδοση του Connect4 για παιχνίδι υπολογιστή εναντίον υπολογιστή χρησιμοποιώντας ξανά το υποσύστημα μάθησης της έκδοσης Alpha. Το υποσύστημα μάθησης έπρεπε να προσαρμοστεί για να ολοκληρωθεί με το Connect4 αφού ακόμη χρησιμοποιούσε δομές που ήταν πολύ δεμένες με την υλοποίηση του RLGame(π.χ. την κλάση του πιονιού και την κλάση του τετραγώνου της σκακιέρας που δεν χρησιμοποιούνται στο connect4). Οι δύο εκδόσεις παραδόθηκαν τον Μάρτιο 2012 και ο έλεγχος τους βασίστηκε σε μελέτη των κινήσεων του παιχνιδιού για να εντοπιστούν τυχόν λάθη στο υποσύστημα λογικής του παιχνιδιού ενώ εκτελέστηκαν και πειράματα για να φανεί εάν συγκλίνει ο αριθμός νικών των παικτών.

Η ενασχόληση με το connect4 και το RLGlue με βοήθησαν να δω ανάγκες περαιτέρω γενίκευσης που χρειαζόταν ο μηχανισμός μάθησης αλλά και μία διαφορετική λογική για τον τρόπο που μπορεί να επικοινωνηθεί η πληροφορία ανάμεσα στο περιβάλλον και στον πράκτορα μάθησης.

Μετά το πέρας της πιλοτικής εφαρμογής αποφασίστηκε ότι ήταν προτιμότερο οι υλοποιήσεις να μη βασιστούν στο RL-Glue ενώ εκκρεμούσε η υλοποίηση του αλγόριθμου minimax και η επόμενη έκδοση του μηχανισμού μάθησης.

5.1.5 Στάδιο τρία – Μάιος 2012

Τον Μάιο του 2012 παραδόθηκε η έκδοση 1.1 του connect4 με τον μηχανισμό μάθησης. Στην έκδοση αυτή παραδόθηκε και η νέα έκδοση του πιο γενικού και διαχωρισμένου μηχανισμού μάθησης.

Η έκδοση 1.1 του connect4 υποστήριζε παιχνίδι υπολογιστή εναντίον υπολογιστή και minimax εναντίον υπολογιστή ενώ εισήγαγε και έναν παίκτη τυχαίας επιλογής . Η έκδοση του μηχανισμού είχε πάρει πλέον την τελική της μορφή (αποτυπώνεται στο επόμενο κεφάλαιο). Η έκδοση αυτή εισήγαγε την ενοποίηση του minimax και του μηχανισμού μάθησης σε ένα πρόγραμμα, με δυνατότητα επιλογής του τύπου παίκτη μέσα από ορίσματα που δίνονται κατά τη διάρκεια της εκτέλεσης, ενώ δόθηκε και δυνατότητα παιχνιδιού minimax εναντίον minimax.

Σε ότι αφορά τον μηχανισμό μάθησης το υποσύστημα αναδιατάχθηκε πλήρως καθώς απαλείφτηκαν κλάσεις, και αντικαταστάθηκαν τα πρότυπα σχεδίασης που αναφέρθηκαν στην έκδοση alpha του σταδίου 1. Το υποσύστημα πλέον δεν καλεί οποιαδήποτε μέθοδο του υποσυστήματος λογικής παιχνιδιού. Αυτό έγινε εφικτό με την αλλαγή στο τρόπο επικοινωνίας της παρατήρησης από το υποσύστημα λογικής παιχνιδιού (περιβάλλον) στον πράκτορα μάθησης. Σε αυτή την έκδοση, στην παρατήρηση που επικοινωνεί τις επιτρεπόμενες κινήσεις επικοινωνούνται ο δυαδικός πίνακας που χρησιμοποιείται από το νευρωνικό δίκτυο και η ανταμοιβή από το περιβάλλον (που χρησιμοποιείται στη διαδικασία επιλογής κινήσεων του μηχανισμού). Ακόμη για κάθε παίκτη ορίζεται ένα διαφορετικό αντικείμενο πράκτορα και από την μεριά του μηχανισμού είναι εφικτό να υποστηριχθούν παραπάνω από 2 παίκτες για ένα παιχνίδι. Κατά την δημιουργία του αντικειμένου του πράκτορα ο παίκτης επικοινωνεί και όλες τις μεταβλητές παραμέτρους που χρειάζονται στην εκτέλεση των αλγορίθμων Ενισχυτικής Μάθησης και των Νευρωνικών Δικτύων. Τέλος σε αυτή την έκδοση ο μηχανισμός μάθησης διαχωρίστηκε σαν ένα ξεχωριστό πρόγραμμα και οι κλάσεις του μηχανισμού ομαδοποιήθηκαν σε ένα ξεχωριστό Java jar που μπορεί να χρησιμοποιηθεί σαν βιβλιοθήκη από υλοποιήσεις παιχνιδιών στρατηγικής.

5.1.6 Στάδιο τέσσερα – Ιούλιος 2012

Τον Ιούλιο του 2012 παραδόθηκε η έκδοση 1.2 του RLGame με τον μηχανισμό μάθησης. Στην έκδοση αυτή το RLGame υποστήριζε παιχνίδι υπολογιστή εναντίον υπολογιστή και minimax εναντίον υπολογιστή ενώ εισήγαγε και έναν παίκτη τυχαίας επιλογής. Όπως και στην περίπτωση του connect4, η τελική έκδοση του RLGame εισήγαγε την ενοποίηση του minimax και του μηχανισμού μάθησης σε ένα πρόγραμμα, με δυνατότητα επιλογής του τύπου παίκτη μέσα από ορίσματα που δίνονται κατά τη διάρκεια της εκτέλεσης του προγράμματος. Ακόμη υπάρχει δυνατότητα παιχνιδιού minimax εναντίον minimax.

Στην έκδοση αυτή χρησιμοποιήθηκε η έκδοση του σταδίου 3 του πιο γενικού και διαχωρισμένου μηχανισμού μάθησης. Οι μόνες αλλαγές που έγιναν στον μηχανισμό αφορούσαν διορθώσεις σφαλμάτων και αλλαγές στην εμβέλεια της κλάσης του Νευρωνικού δικτύου.

Κεφάλαιο 6
Νέα Αρχιτεκτονική RLGame

Σε αυτό το κεφάλαιο περιγράφεται η νέα υλοποίηση όπου η λογική του παιχνιδιού είναι διαχωρισμένη από τον μηχανισμό μάθησης. Παρατίθεται λεπτομερής περιγραφή της τελικής αρχιτεκτονικής, και παρουσιάζονται θέματα υλοποίησης όπως και στοιχεία του ελέγχου διασφάλισης ποιότητας μέσω εκτέλεσης πειραμάτων με την νέα έκδοση.

6.1 Νέα Υλοποίηση

Στην τελική έκδοση του RLGame ο μηχανισμός εκτέλεσης του παιχνιδιού είναι διαχωρισμένος από τον μηχανισμό μάθησης.

Στην έκδοση αυτή το RLGame υποστηρίζει παιχνίδι υπολογιστή εναντίον υπολογιστή και minimax εναντίον υπολογιστή ενώ εισήχθηκε και ο παίκτης τυχαίας επιλογής για χρήση σε πειράματα.

Ακόμη στην η τελική έκδοση του RLGame εισήχθηκε η ενοποίηση του minimax και του μηχανισμού μάθησης σε ένα πρόγραμμα, και δόθηκε η δυνατότητα επιλογής του τύπου παίκτη μέσα από ορίσματα κατά τη διάρκεια της εκτέλεσης όπως και η δυνατότητα παιχνιδιού minimax εναντίον minimax.

Τέλος στην εκτελέσιμη μορφή του το σύστημα αποτελείται από το εκτελέσιμο Java jar του RLGame που έχει εξάρτηση από το Java jar του μηχανισμού μάθησης.

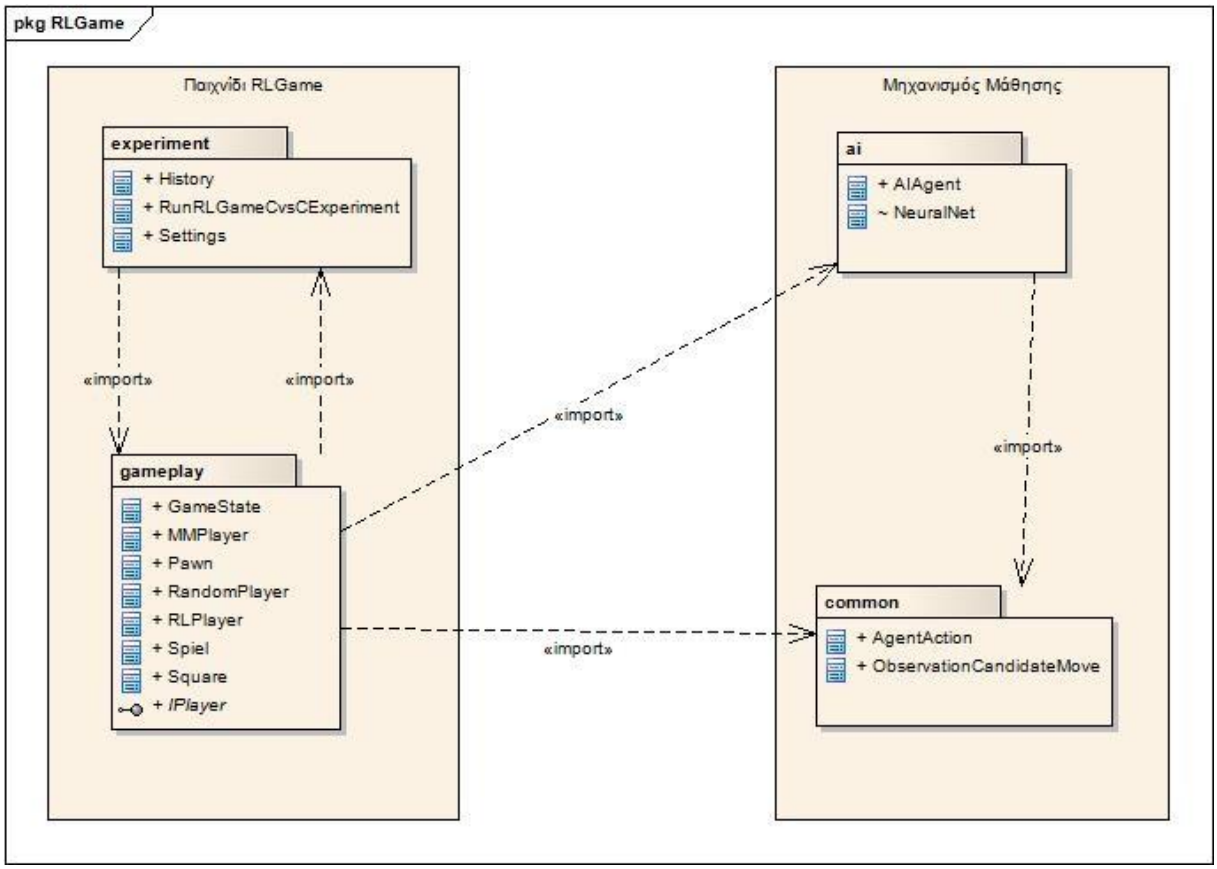
6.2 Αρχιτεκτονική

Ακολουθώντας πρακτικές αντικειμενοστραφούς προγραμματισμού και συμβάσεων της γλώσσας προγραμματισμού Java, η υλοποίηση χωρίστηκε σε 4 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.gameplay	Το πακέτο αυτό περιέχει τις κλάσεις που περιγράφουν την λογική του παιχνιδιού, που εφαρμόζουν τους κανόνες, εκτελούν τις κινήσεις και ελέγχουν την κατάσταση του παιχνιδιού.
org.rlgame.experiment	Το πακέτο αυτό περιέχει τις κλάσεις για την εκτέλεση του πειράματος.
org.rlgame.common	Το πακέτο αυτό περιέχει τις κοινές βοηθητικές κλάσεις που χρησιμοποιούνται στην αμφίδρομη επικοινωνία του υποσυστήματος λογικής παιχνιδιού και του υποσυστήματος μάθησης.
org.rlgame.ai	Το πακέτο αυτό περιέχει τις κλάσεις του υποσυστήματος μάθησης.

Στο σχεδιάγραμμα της εικόνας 6.1 φαίνονται τα πακέτα που περιγράψαμε στον παραπάνω πίνακα. Στο σχεδιάγραμμα φαίνεται η ομαδοποίηση των πακέτων στα υποσυστήματα λογικής παιχνιδιού και μηχανισμού μάθησης. Ακόμη οι διακεκομμένες γραμμές με την ετικέτα import συμβολίζουν τις εξαρτήσεις των κλάσεων των πακέτων, δηλαδή το γεγονός ότι χρησιμοποιούν κλάσεις από άλλα πακέτα.. Το σύστημα που φαίνεται στο λογικό διάγραμμα έχει διαχωριστεί και φυσικά σε δύο ξεχωριστά προγράμματα. Το πρόγραμμα του μηχανισμού μάθησης μπορεί να θεωρηθεί ως μια βιβλιοθήκη/api (Application Programming Interface) η οποία χρησιμοποιείται από το πρόγραμμα του παιχνιδιού RLGame. Στην εκτελέσιμη μορφή του το σύστημα

αποτελείται από το εκτελέσιμο jar του RLGame που έχει εξάρτηση από το jar του μηχανισμού μάθησης.



Εικόνα 6.1 : Γενική άποψη του νέου συστήματος (Διάγραμμα πακέτων προγράμματος)

Το πακέτο εκτέλεσης του πειράματος περιέχει την κλάση που εκτελεί το πρόγραμμα (πείραμα), λαμβάνει τις παραμέτρους εκτέλεσης, αρχικοποιεί το υποσύστημα της εκτέλεσης και λογικής του παιχνιδιού και καταγράφει τα στατιστικά στοιχεία του πειράματος σε αρχεία.

Το πακέτο εκτέλεσης και λογικής του παιχνιδιού δημιουργεί τα αντικείμενα που ορίζουν την κατάσταση του παιχνιδιού, τα δύο αντικείμενα των παικτών που θα παίξουν το παιχνίδι και αυτά με την σειρά τους (στην περίπτωση παικτών ενισχυτικής μάθησης) δημιουργούν το αντίστοιχο αντικείμενο του πράκτορα ενισχυτικής μάθησης που θα επιλέξει τις κινήσεις αυτού του υπολογιστικού παίκτη. Κατά τη διάρκεια της εκτέλεσης του παιχνιδιού το κάθε αντικείμενο παίκτη καλεί μεθόδους του αντίστοιχου αντικειμένου πράκτορα μάθησης περνώντας σαν όρισμα ή λαμβάνοντας απάντηση με κάποιο αντικείμενο από τις βοηθητικές κλάσεις του πακέτου βοηθητικών κλάσεων που ορίζονται στο υποσύστημα μάθησης.

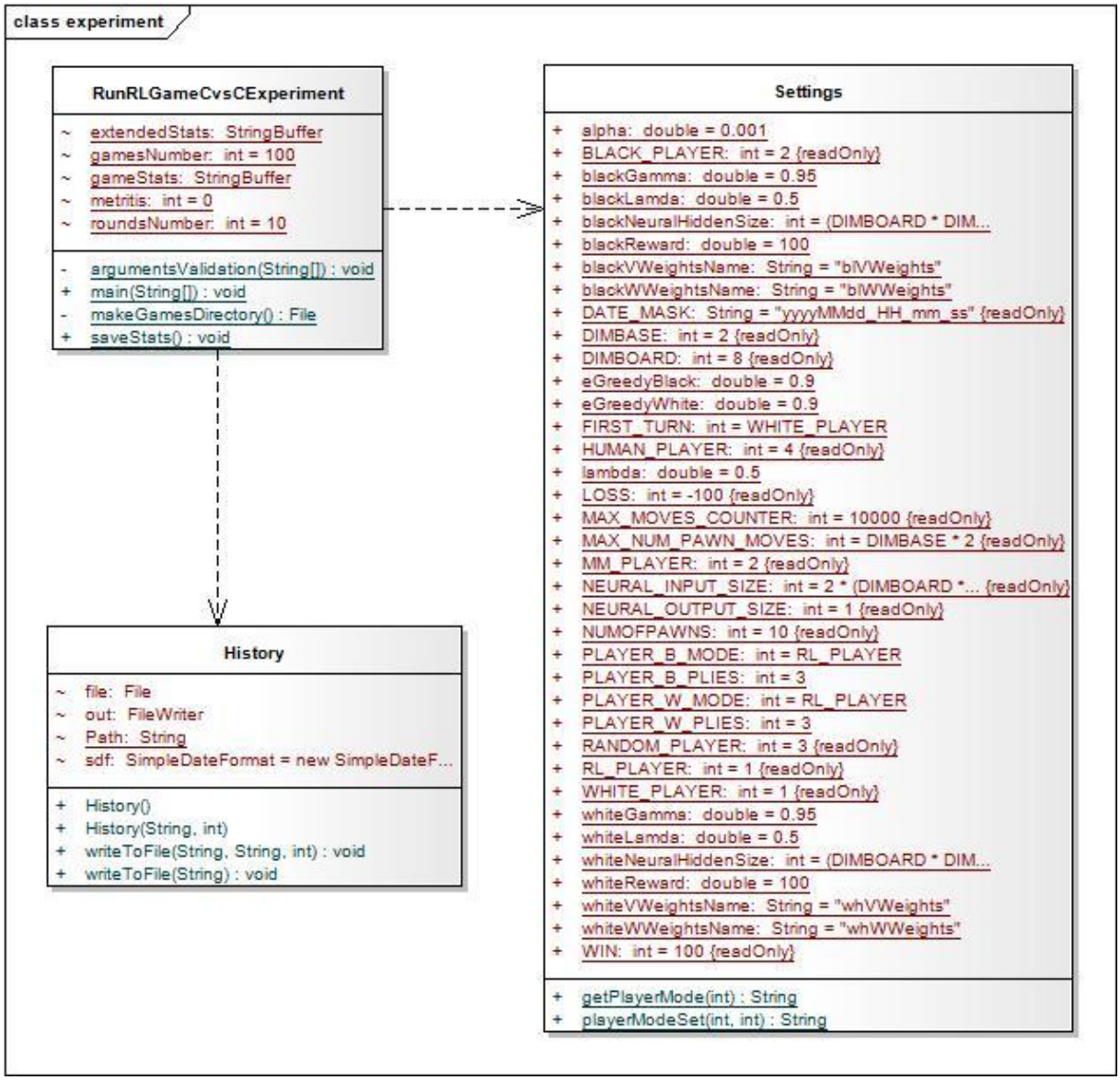
Τέλος στο πακέτο μάθησης, κάθε αντικείμενο πράκτορα που αποκτά υπόσταση αρχικοποιεί το αντίστοιχο νευρωνικό δίκτυο που θα χρησιμοποιηθεί για τον υπολογισμό της τιμής αξιολόγησης της κάθε πιθανής επόμενης κίνησης του παίκτη που έχει δημιουργήσει και χρησιμοποιεί τις υπηρεσίες αυτού του πράκτορα.

6.2.1 Πακέτο Εκτέλεσης Πειραμάτων

Στο πακέτο εκτέλεσης πειραμάτων έχουν ορισθεί οι ακόλουθες κλάσεις:

Κλάση	**Περιγραφή**
History.java	Κλάση επιφορτισμένη με την καταγραφή των εξαγόμενων αρχείων αποτελεσμάτων των παιχνιδιών και των κινήσεων των παικτών.
`RunRLGameCvsCExperiment`.java	Η κλάση που εκτελεί το πείραμα και είναι υπεύθυνη για την λήψη των παραμέτρων εκτέλεσης, και την αρχικοποίηση παικτών του παιχνιδιού και της κλάσης που εκτελεί το παιχνίδι.
Settings.java	Κλάση στην οποία ορίζονται οι καθολικές στατικές μεταβλητές του προγράμματος και οι στατικές μεταβλητές που μπορεί να μεταβληθούν ανάλογα με τα ορίσματα εκτέλεσης. Αυτή η κλάση χρησιμοποιείται από τις κλάσεις του πακέτου πειράματος και από τις κλάσεις του πακέτου λογικής και εκτέλεσης παιχνιδιού.

Στο ακόλουθο διάγραμμα κλάσεων παρουσιάζονται οι κλάσεις του πακέτου πειράματος. Οι διακεκομμένες γραμμές συμβολίζουν τις εξαρτήσεις της κλάσης RunRLGameCvsCExperiment (η κλάση καλεί στατικές μεθόδους των άλλων 2 κλάσεων του πακέτου).



Εικόνα 6.2 : Διάγραμμα κλάσεων του πακέτου πειράματος

6.2.2 Πακέτο Λογικής Παιχνιδιού

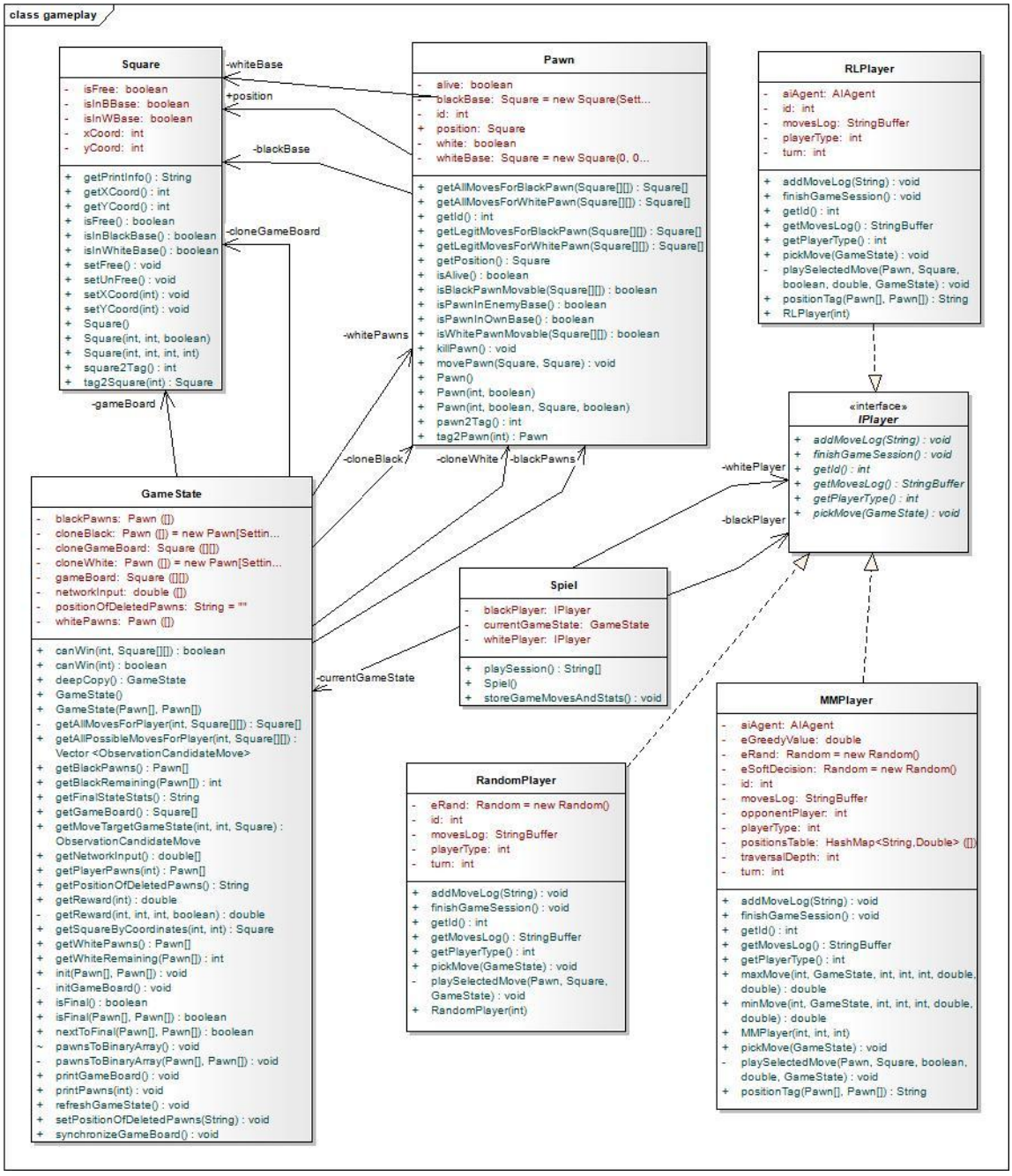
Στο πακέτο λογικής παιχνιδιού έχουν ορισθεί οι ακόλουθες κλάσεις:

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.
MMPlayer.java	Η κλάση του Min-Max παίκτη. Η κλάση ακολουθεί/υλοποιεί το πρότυπο IPlayer.java. Ο αλγόριθμος Minimax υλοποιείται σε 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 μεθόδου 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.
Pawn.java	Δημιουργία – διαχείριση πιονιών. Έχει ενσωματωμένη λογική/κανόνων κινήσεων.
RandomPlayer.java	Η κλάση του παίκτη τυχαίας επιλογής. Η κλάση ακολουθεί/υλοποιεί το πρότυπο IPlayer.java. Ο τύπος του παίκτη προστέθηκε στην παρούσα υλοποίηση.
RLPlayer.java	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 χρησιμοποίει 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 applySelectedMoveReward όπου επικοινωνεί στον πράκτορα την ανταμοιβή του περιβάλλοντος για την επιλεγμένη κίνηση.
Spiel.java	Η κλάση εκτέλεσης ενός παιχνιδιού/επεισοδίου
Square.java	Η κλάση που αναπαριστά το τετράγωνο της σκακιέρας

Στο ακόλουθο διάγραμμα κλάσεων παρουσιάζονται οι κλάσεις του πακέτου λογικής του παιχνιδιού. Οι συνεχόμενες γραμμές συμβολίζουν τις σχέσεις των κλάσεων (associations) ενώ οι διακεκομμένες συμβολίζουν το γεγονός ότι οι τρεις διαφορετικές κλάσεις παικτών υλοποιούν το πρότυπο (interface) IPlayer.



Εικόνα 6.3 : Διάγραμμα κλάσεων του πακέτου λογικής παιχνιδιού

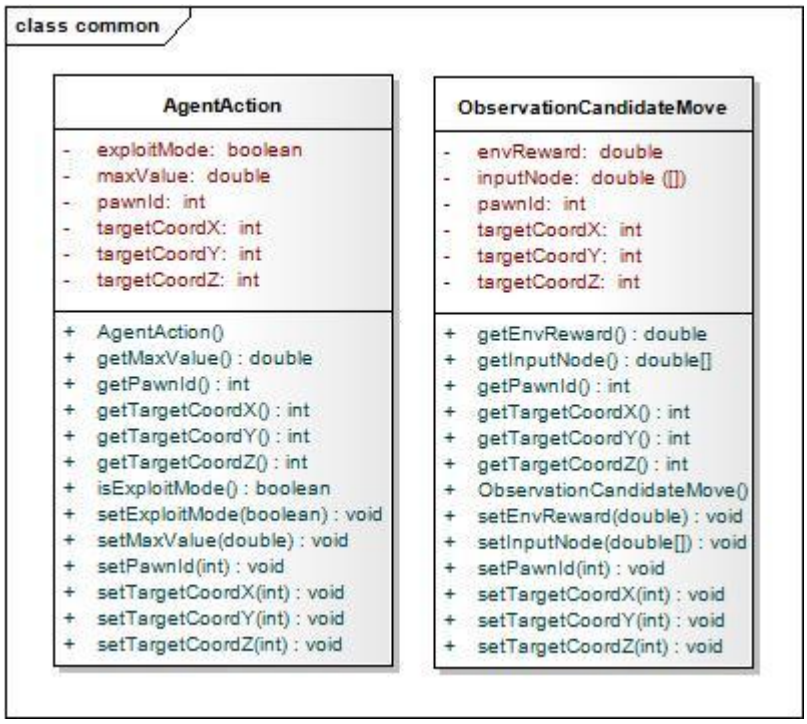
6.2.3 Πακέτο Βοηθητικών κλάσεων

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 (Στις υλοποιήσεις του RLGame και του Connect4 αποστέλλεται η ανταμοιβή του περιβάλλοντος στην παρατήρηση, ένα άλλο πρόγραμμα που θα χρησιμοποιήσει τον μηχανισμό θα αποφασίσει εάν θα περνάει πάντα την τιμή 0 έτσι ώστε να λάβει επιλογή που θα προκύπτει μόνο από την αξιολόγηση του Νευρωνικού Δικτύου). Ακόμη έχουν ορισθεί οι 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).
AgentAction.java	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.

Στο ακόλουθο διάγραμμα κλάσεων παρουσιάζονται οι κλάσεις του πακέτου βοηθητικών κλάσεων.



Εικόνα 6.4 : Διάγραμμα κλάσεων του πακέτου βοηθητικών κλάσεων

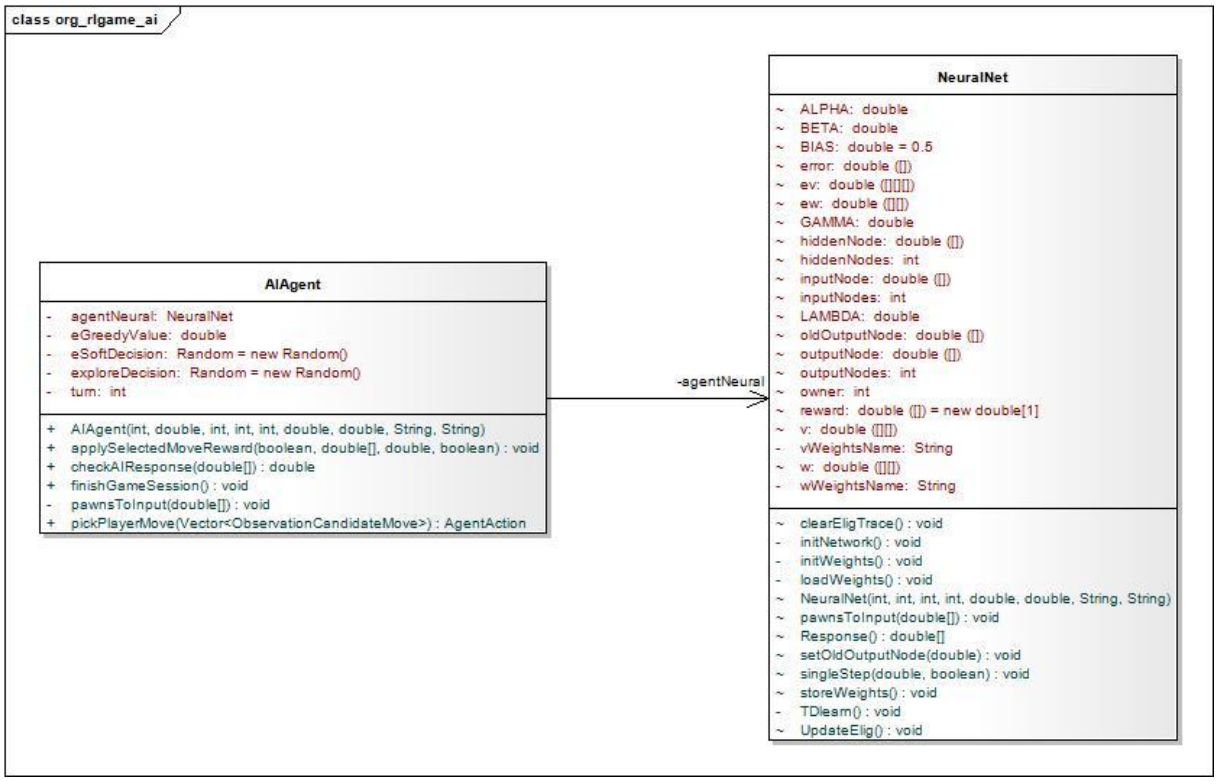
6.2.4 Πακέτο Μάθησης

Στο πακέτο του μηχανισμού μάθησης έχουν ορισθεί οι 2 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 6.5 : Διάγραμμα κλάσεων του πακέτου μάθησης

6.3 Αξιολόγηση Διαχωρισμού Υποσυστημάτων

Όπως έχει περιγραφεί και παραπάνω στην υπάρχουσα υλοποίηση του RLGame η υλοποίηση της λογικής του παιχνιδιού και του μηχανισμού μάθησης ήταν ενοποιημένες καθώς λειτουργικότητα του μηχανισμού μάθησης βρισκόταν στις κλάσεις του παίκτη και της κατάστασης του παιχνιδιού, ενώ η κλάση του νευρωνικού περιείχε, χρησιμοποιούσε αντικείμενα της λογικής παιχνιδιού στη διαδικασία δημιουργίας του δυαδικού πίνακα που δέχεται σαν είσοδο η υλοποίηση του Νευρωνικού δικτύου.

Όπως έχει αναφερθεί παραπάνω η νέα υλοποίηση πραγματοποιήθηκε σε διαφορετικές φάσεις.

Στην έκδοση alpha που παραδόθηκε τον Ιανουάριο 2012, η λογική του παιχνιδιού και ο μηχανισμός μάθησης αποτελούσαν ξεχωριστά υποσυστήματα αλλά υπήρχε η ανάγκη αμφίδρομης επικοινωνίας αφού το υποσύστημα μάθησης έπρεπε να χρησιμοποιήσει μεθόδους του υποσυστήματος εκτέλεσης του παιχνιδιού. Για παράδειγμα ο μηχανισμός μπορούσε να χρησιμοποιήσει μεθόδους του αντικειμένου που αναπαριστά την τρέχουσα κατάσταση του παιχνιδιού ή γνώριζε/χρησιμοποιούσε interfaces των υλοποιήσεων των αντικειμένων του πιονιού και του τετραγώνου της σκακιέρας.

Στην τελική έκδοση που παραδόθηκε στο τέλος της διπλωματικής ο μηχανισμός μάθησης είναι πλέον διαχωρισμένος από την υπόλοιπη υλοποίηση. Το υποσύστημα του μηχανισμού μάθησης μπορεί να θεωρηθεί ως μία βιβλιοθήκη που παρέχει ένα σύνολο δημόσιων μεθόδων (api) και μπορεί να χρησιμοποιηθεί σε κάποιο άλλο Java πρόγραμμα. Το υποσύστημα του μηχανισμού μάθησης δεν έχει οποιαδήποτε γνώση για το παιχνίδι ή τις λεπτομέρειες της υλοποίησης και δεν καλεί οποιαδήποτε μέθοδο αντικειμένων του υποσυστήματος εκτέλεσης του παιχνιδιού.

Πρέπει να αναφερθεί όμως, ότι ο μηχανισμός περιμένει από το υποσύστημα παιχνιδιού ότι θα τροφοδοτηθεί μόνο με τις επιτρεπόμενες κινήσεις και ότι για κάθε κίνηση θα λάβει τον δυαδικό πίνακα εισόδου του νευρωνικού (χρησιμοποιείται στην αναζήτηση της αξιολόγησης της κατάστασης) καθώς και την ανταμοιβή που δίνει το περιβάλλον σε αυτή την κατάσταση (Ο μηχανισμός μάθησης χρησιμοποιεί αυτή την πληροφορία στην φάση της εκμετάλλευσης της υπάρχουσας γνώσης - exploitation). Ακόμη σε αυτή την έκδοση έγινε δυνατό να δημιουργηθούν ξεχωριστά προγράμματα για τα 2 υποσυστήματα τα οποία και πακετάρονται σε ξεχωριστά Java jar.

Το γεγονός ότι ο μηχανισμός αποτελεί πλέον ξεχωριστό υποσύστημα/βιβλιοθήκη που μπορεί να χρησιμοποιηθεί από διάφορες υλοποιήσεις παιχνιδιών συνεπάγεται έναν ικανοποιητικό βαθμό διαχωρισμού των υποσυστημάτων. Βέβαια όπως αναφέρεται στην ακόλουθη λίστα, ένα πρόγραμμα παιχνιδιού όταν ολοκληρώνεται με τον μηχανισμό μάθησης έχει τις εξής εξαρτήσεις:

- Το jar του μηχανισμού μάθησης και η υλοποίηση της κλάσης του πράκτορα ενισχυτικής μάθησης όσο και των βοηθητικών κλάσεων, χρειάζεται να είναι διαθέσιμη κατά την μεταγλώττιση του προγράμματος του παιχνιδιού

- Το jar του μηχανισμού μάθησης χρειάζεται να είναι διαθέσιμο κατά την εκτέλεση του προγράμματος του παιχνιδιού

- Το υποσύστημα του παιχνιδιού καλεί τον πράκτορα να επιλέξει κίνηση επικοινωνώντας την παρατήρηση του περιβάλλοντος και λαμβάνει σαν απάντηση την επιλεγμένη κίνηση. Τόσο η παρατήρηση όσο και επιλεγμένη κίνηση επικοινωνούνται με αντικείμενα του πακέτου βοηθητικών κλάσεων του μηχανισμού μάθησης.

6.4 Εκτέλεση πειραμάτων με τη νέα υλοποίηση

Για την επιβεβαίωση της ορθότητας της νέας έκδοσης εκτελέστηκε μια σειρά πειραμάτων τόσο σε παιχνίδι υπολογιστή εναντίον υπολογιστή όσο και σε παιχνίδι minimax εναντίον υπολογιστή. Τα πειράματα που παραθέτονται σε αυτή την παράγραφο, έχουν εκτελεστεί με την τελική έκδοση του RLGame. Στις ακόλουθες παραγράφους αναφέρονται τα αποτελέσματα αυτών των πειραμάτων.

6.4.1 Πειράματα σε παιχνίδι υπολογιστή εναντίον υπολογιστή

Η έκδοση αυτή ελέγχθηκε καταρχήν μέσω πειραμάτων σε σύνολο 50000 παιχνιδιών υπολογιστή εναντίον υπολογιστή με ένα ζεύγος νευρωνικών δικτύων που δημιουργήθηκαν για αυτό το πείραμα. Καταρχήν δεν υπήρξε κάποιο σφάλμα κατά την διάρκεια εκτέλεσης των πειραμάτων ενώ τα αποτελέσματα των παιχνιδιών μοιάζουν με τα δημοσιευμένα αποτελέσματα της περίπτωσης εκπαίδευσης μόνο μέσω ατομικού παιχνιδιού σε μη εκπαιδευμένο νευρωνικό δίκτυο[1].

Οι συνολικές νίκες είναι 27007 για τον λευκό και 22993 για τον μαύρο ενώ ο μέσος αριθμός κινήσεων είναι 217,94 κινήσεις.

Στον παρακάτω πίνακα φαίνονται τα αποτελέσματα ανά σετ 5000 παιχνιδιών.

Σετ 5000 παιχνιδιών	**Παίκτης**	**Νίκες**	**Κινήσεις**	**Αριθμός ενεργών πιονιών λευκού**	**Αριθμός ενεργών πιονιών μαύρου**	**Πιόνια λευκού στη βάση**	**Πιόνια μαύρου στη βάση**
1	Aspros	4820	48,19	8,72	8,57	3,98	2,29
1	mavros	180	142,64	6,43	8,43	1,42	0,77
2	Aspros	4780	60,02	8,18	8,48	2,76	1,98
2	mavros	220	176,88	5,7	8,33	0,84	0,41
3	Aspros	1887	281,73	6,66	6,55	1,2	0,19
3	mavros	3113	243,12	7,3	7,28	1,72	0,18
4	Aspros	2162	269,71	6,37	6,96	1,35	0,07
4	mavros	2838	254,13	6,33	7,21	1,47	0,17
5	Aspros	2285	275,74	6,29	6,91	1,3	0,01
5	mavros	2715	275,76	6,05	6,95	1,12	0
6	Aspros	2015	263,23	6,28	7,01	1,43	0,01
6	mavros	2985	239,66	6,45	7,37	1,6	0
7	Aspros	2088	267,44	6,29	6,88	1,39	0
7	mavros	2912	240,89	6,29	7,24	1,56	0,01
8	Aspros	2021	262,52	6,17	7,1	1,41	0,01
8	mavros	2979	232,19	6,35	7,53	1,65	0,01
9	Aspros	2040	257,08	6,24	7,14	1,54	0
9	mavros	2960	229,09	6,41	7,6	1,64	0
10	Aspros	2909	281,4	5,76	6,99	1,19	0
10	mavros	2091	283,51	5,68	6,78	0,93	0

Κατόπιν θέλησα να συγκρίνω το τωρινό πρόγραμμα με την προηγούμενη υλοποίηση υπολογιστή εναντίον υπολογιστή που μου δόθηκε. Η ιδέα ήταν να ελέγξω αντίγραφα των εκπαιδευμένων νευρωνικών που ανέφερα παραπάνω, και στις 2 περιπτώσεις με το σενάριο με ε-greedy policy = 1.0 (καθόλου εξερεύνηση). Το αποτέλεσμα ήταν ότι τα παιχνίδια και στις 2 περιπτώσεις δεν έφταναν σε τελική κατάσταση μετά από 10000 κινήσεις (Στα πειράματα με ε-

greedy policy = 0.9 τέλειωναν εξαιτίας της εξερεύνησης). Κατόπιν εκπαίδευσα 2 νέα νευρωνικά με 2000 παιχνίδια και εκτέλεσα το ίδιο πείραμα για 100 παιχνίδια. Αυτή τη φορά το αποτέλεσμα των νικών ήταν πανομοιότυπο καθώς και τις 100 φορές κέρδισε ο λευκός σε 13 κινήσεις. Μάλιστα, όταν έτρεξα το πείραμα έχοντας διορθώσει ένα σφάλμα του παλιού προγράμματος κατά τη συμπλήρωση κάποιων επιμέρους τιμών στην είσοδο του νευρωνικού, οι κινήσεις των παικτών και στις 2 περιπτώσεις ήταν ακριβώς οι ίδιες. Με βάση τα παραπάνω τα αποτελέσματα του ελέγχου κρίνονται ικανοποιητικά.

6.4.2 Πειράματα σε παιχνίδι minimax εναντίον υπολογιστή

Η έκδοση 1.3 του RLGame ελέγχθηκε μέσω πειραμάτων σε σύνολο 3500 παιχνιδιών minimax με βάθος αναζήτησης 5 (Λευκός) εναντίον υπολογιστή (Μαύρος) σε σκακιέρα 6x6 και βάση 2x2 με ένα ζεύγος νευρωνικών δικτύων που δημιουργήθηκαν για αυτό το πείραμα και με τις προεπιλεγμένες ρυθμίσεις. Δεν υπήρξε κάποιο σφάλμα κατά την διάρκεια εκτέλεσης των πειραμάτων.

Οι συνολικές νίκες είναι 1388 για τον λευκό και 2612 για τον μαύρο ενώ ο μέσος αριθμός κινήσεων είναι 8,64 κινήσεις.

Τα αποτελέσματα αυτά ήταν μη αναμενόμενα και πραγματοποιήθηκε περαιτέρω έλεγχος. Στον έλεγχο αυτό εντοπίστηκε ότι ο minimax παίκτης είχε αυτή την απόδοση λόγω ενός σφάλματος που δημιουργήθηκε κατά την μεταφορά της υλοποίησης του minimax αλγορίθμου του RLGame στην νέα ενοποιημένη έκδοση. Πιο συγκεκριμένα, μία συνθήκη είχε απαλειφτεί και ο max παίκτης δεν μπορούσε να καταλάβει τις καταστάσεις στις οποίες έχανε (ενώ καταλάβαινε τις περιπτώσεις που μπορί να έχανε επειδή είχε χάσει όλα τα πιόνια δεν μπορούσε να καταλάβει τις περιπτώσεις που έχανε επειδή ο αντίπαλος min είχε βάλει ένα πιόνι στην βάση του).

Σε πειράματα που πραγματοποιήθηκαν με την έκδοση 1.4 στις συνθήκες που επιλέχθηκαν και στο πείραμα της έκδοσης 1.3, ο λευκός παίκτης ήταν κατά κανόνα ο νικητής με την βοήθεια του minimax αλγορίθμου και αναζήτηση σε βάθος 5.

Καταρχήν τα νευρωνικά αρχικοποιήθηκαν με ένα σετ 1000 παιχνιδιών υπολογιστή εναντίον υπολογιστή. Σε αυτή την περίπτωση τα αποτελέσματα ήταν ισορροπημένα καθώς οι αντίπαλοι είχαν περίπου τις ίδιες νίκες.

Παίκτης	Νίκες	Κινήσεις	Αριθμός ενεργών πιονιών λευκού	Αριθμός ενεργών πιονιών μαύρου	Πιόνια λευκού στη βάση	Πιόνια μαύρου στη βάση
aspros	506	14,54	9,69	8,31	6,77	4,32
mavros	494	16,24	9,65	9,05	6,85	4,46

Ακολούθησε έλεγχος σε σύνολο 2500 παιχνιδιών minimax με βάθος αναζήτησης 5 (Λευκός) εναντίον υπολογιστή (Μαύρος) σε σκακιέρα 6x6 και βάση 2x2 με το ζεύγος νευρωνικών δικτύων που αναφέρθηκε παραπάνω. Τα αποτελέσματα ήταν αυτή την φορά αναμενόμενα καθώς με την βοήθεια του minimax ο λευκός κέρδισε συντριπτικά περισσότερα παιχνίδια. Τα συνοπτικά αποτελέσματα και τα αποτελέσματα ανά σετ των 500 παιχνιδιών παρουσιάζονται στους ακόλουθους (2) πίνακες.

Παίκτης	Νίκες	Κινήσεις	Αριθμός ενεργών πιονιών λευκού	Αριθμός ενεργών πιονιών μαύρου	Πιόνια λευκού στη βάση	Πιόνια μαύρου στη βάση
aspros	2390	15,69	9,1	9,38	5,41	4,66
mavros	110	21,85	7,19	9,5	3,51	4,07

Σετ 500 παιχνιδιών	Παίκτης	Νίκες	Κινήσεις	Αριθμός ενεργών πιονιών λευκού	Αριθμός ενεργών πιονιών μαύρου	Πιόνια λευκού στη βάση	Πιόνια μαύρου στη βάση
1	aspros	472	15,12	9,2	9,43	5,62	4,76
1	mavros	28	19,32	7,5	9,75	3,61	4,57
2	aspros	474	15,77	9,06	9,44	5,41	4,72
2	mavros	26	22,65	7,65	9,42	4,04	3,96
3	aspros	483	15,97	8,97	9,36	5,24	4,58
3	mavros	17	25,35	5,71	8,94	2,12	3,29
4	aspros	477	15,54	9,12	9,32	5,36	4,61
4	mavros	23	21,43	7,13	9,57	3,52	4,13

5	aspros	484	16,04	9,16	9,34	5,42	4,62
5	mavros	16	21,88	7,56	9,69	3,94	4,12

Για επιβεβαίωση των συμπερασμάτων του ελέγχου εκτέλεσα και μία σειρά 3500 παιχνιδιών υπολογιστή εναντίον υπολογιστή χρησιμοποιώντας αντίγραφα των αρχικοποιημένων νευρωνικών. Σε αυτή τη μορφή παιχνιδιού ο μαύρος παίκτης κέρδισε τα περισσότερα παιχνίδια.

Παίκτης	**Νίκες**	**Κινήσεις**	**Αριθμός ενεργών πιονιών λευκού**	**Αριθμός ενεργών πιονιών μαυρού**	**Πιόνια λευκού στη βάση**	**Πιόνια μαύρου στη βάση**
aspros	1254	15,62	9,66	8,1	6,46	3,94
mavros	2246	14,56	9,67	9,14	6,58	4,59

Κεφάλαιο 7
Connect 4 και Μηχανισμός Μάθησης RLGame

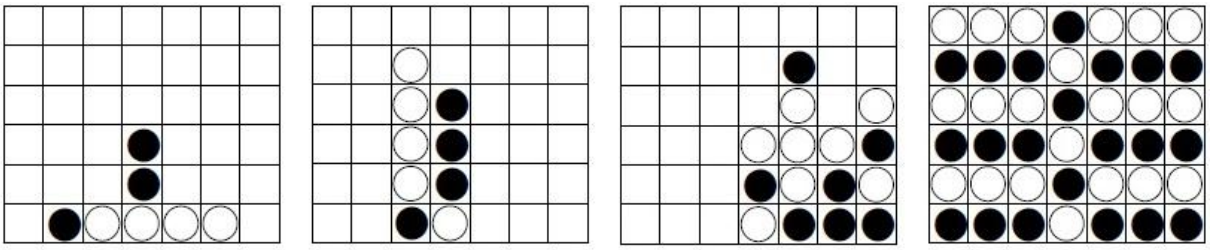
Στα πλαίσια της μελέτης χρησιμότητας του μηχανισμού μάθησης που προέκυψε από τον διαχωρισμό των υποσυστημάτων που περιγράφηκε στα προηγούμενα κεφάλαια, επιλέχθηκε να γίνει ολοκληρωθεί/δοκιμαστεί ο μηχανισμός μάθησης με το παιχνίδι στρατηγικής connect4. Σε αυτό το κεφάλαιο δίνεται μια σύντομη περιγραφή του παιχνιδιού Connect 4 και περιγράφεται η ολοκλήρωση του παιχνιδιού με τον μηχανισμό μάθησης που χρησιμοποιείται στο RLGame.

7.1 Παιχνίδι Στρατηγικής Connect4

Το παιχνίδι στρατηγικής Connect4 είναι ένα παιχνίδι 2 παικτών με εναλλαγή σειράς. Η κύρια έκδοση του παιχνιδιού παίζεται σε ένα κάθετο παραλληλόγραμμο πλαίσιο επτά κολώνων και έξι γραμμών και έχει συνολικά 42 θέσεις. Ο κάθε παίκτης έχει 21 πιόνια τα οποία μπορεί να τοποθετήσει στο πλαίσιο του παιχνιδιού πετώντας τα από την κορυφή του πλαισίου. Με αυτό τον τρόπο, τα πιόνια τοποθετούνται στην χαμηλότερη ελεύθερη θέση της κολώνας που επέλεξε ο παίκτης. Ο στόχος του κάθε παίκτη είναι να συνδέσει τέσσερα συνεχόμενα πιόνια με οποιοδήποτε τρόπο (κάθετα, οριζόντια ή διαγώνια). Ο πρώτος παίκτης που το καταφέρνει είναι

και ο νικητής του παιχνιδιού. Εάν συμπληρωθούν και οι 42 θέσεις του πλαισίου και δεν υπάρξει νικητής τότε το παιχνίδι είναι ισόπαλο [14,21].

Στην ακόλουθη εικόνα που είναι συρραφή εικόνων της εργασίας [21] φαίνονται 4 παραδείγματα τελικών καταστάσεων του παιχνιδιού. Στις πρώτες τρεις εικόνες ο λευκός παίκτης κερδίζει (έχοντας κάνει μία σειρά από τέσσερα πιόνια σε διάταξη οριζόντια, κάθετη, διαγώνια) ενώ στην τέταρτη εικόνα το αποτέλεσμα είναι ισόπαλο.



Εικόνα 7.1 : Connect4, παραδείγματα τελικών καταστάσεων παιχνιδιού [21]

7.1.1 Ιδιότητες Παιχνιδιού

Στο παιχνίδι στρατηγικής Connect4 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.

Το σύνολο των καταστάσεων του RLGame είναι πεπερασμένο και εξαρτάται μόνο από το μέγεθος του πλαισίου του παιχνιδιού, τον αριθμό των πιονιών και των κανόνων που διέπουν το παιχνίδι.

Οι παίκτες δρουν ανταγωνιστικά με σκοπό τη νίκη και το Connect4 αποτελεί ένα παιχνίδι μηδενικού αθροίσματος (zero-sum) στο οποίο υπάρχει η πιθανότητα νίκης ενός από τους δύο αντιπάλους και η πιθανότητα ισοπαλία.

7.1.2 Βέλτιστη στρατηγική για το Connect4

Σε αντίθεση με το RLGame το connect4 είναι ένα παιχνίδι στρατηγικής που έχει λυθεί. Τον Οκτώβριο του 1988, το παιχνίδι λύθηκε σχεδόν ταυτόχρονα από δύο ανεξάρτητους ερευνητές. Ο James D. Allen δημοσίευσε μία μαθηματική λύση του παιχνιδιού που ενώ ο Victor Allis στα

πλαίσια της διπλωματικής του διατριβή [21] δημοσίευσε μία σειρά από κανόνες που οδηγούν στη λύση του παιχνιδιού.

Με τέλειο παιχνίδι και από τους δύο παίκτες, ο παίκτης που παίζει πρώτος και εκκινεί από την μεσαία κολώνα νικάει πάντα. Ξεκινώντας από τις 2 θέσεις που είναι δίπλα στην μεσαία κολώνα ο πρώτος παίκτης επιτρέπει στον δεύτερο να φτάσει την ισοπαλία. Ενώ ξεκινώντας από τις τέσσερις ακραίες κολώνες επιτρέπει στον δεύτερο παίκτη να νικήσει [14].

7.2 Νέα Υλοποίηση

Στην τελική έκδοση του προγράμματος του παιχνιδιού Connect4 ο μηχανισμός εκτέλεσης του παιχνιδιού που υλοποιήθηκε στα πλαίσια της παρούσας διπλωματικής διατριβής, είναι διαχωρισμένος από τον νέο μηχανισμό μάθησης του RLGame που περιγράφηκε στο προηγούμενο κεφάλαιο.

Στην έκδοση αυτή το πρόγραμμα Connect4 υποστηρίζει παιχνίδι μεταξύ των τριών τύπων παικτών που έχουν ορισθεί στο υποσύστημα λογικής παιχνιδιού: υπολογιστικός πράκτορας μάθησης, minimax, παίκτης τυχαίας επιλογής. Η επιλογή του τύπου παιχνιδιού γίνεται με την βοήθεια ορισμάτων που θέτονται κατά τη διάρκεια της εκτέλεσης. Όπως και στο RLGame, στην εκτελέσιμη μορφή του το σύστημα αποτελείται από το εκτελέσιμο Java jar του Connect4 που έχει εξάρτηση από το Java jar του μηχανισμού μάθησης.

Σχεδόν το σύνολο των προεπιλεγμένων τιμών που ισχύουν για το RLGame και έχουν περιγραφεί στο κεφάλαιο τρία ισχύουν και για το connect4. Στην ακόλουθη λίστα παραθέτονται κάποιες επιλογές που αφορούν στην έκδοση του παιχνιδιού connect4:

- Η έκδοση υποστηρίζει μόνο παιχνίδι σε σκακιέρα 7x6
- Η τιμή ανταμοιβής του περιβάλλοντος στην περίπτωση της ισοπαλίας είναι 0
- Για τα επίπεδα του Νευρωνικού Δικτύου έχουν ορισθεί τα ακόλουθα :
    - Ο δυαδικός πίνακας που δίνεται σαν input στο νευρωνικό έχει 90 θέσεις. Για κάθε θέση της σκακιέρας έχουν ορισθεί 2 θέσεις στον δυαδικό πίνακα που αντιστοιχούν σε κάθε παίκτη. Εάν η θέση είναι κατειλημμένη από πιόνι του

παίκτη το κελί συμπληρώνεται με την τιμή 1 αλλιώς παίρνει την τιμή 0. Τέλος έχουν προβλεφθεί 3 ακόμη θέσεις για κάθε πιθανό αποτέλεσμα: ισοπαλία, νίκη του λευκού, νίκη του μαύρου παίκτη (και 3 θέσεις μένουν κενές).

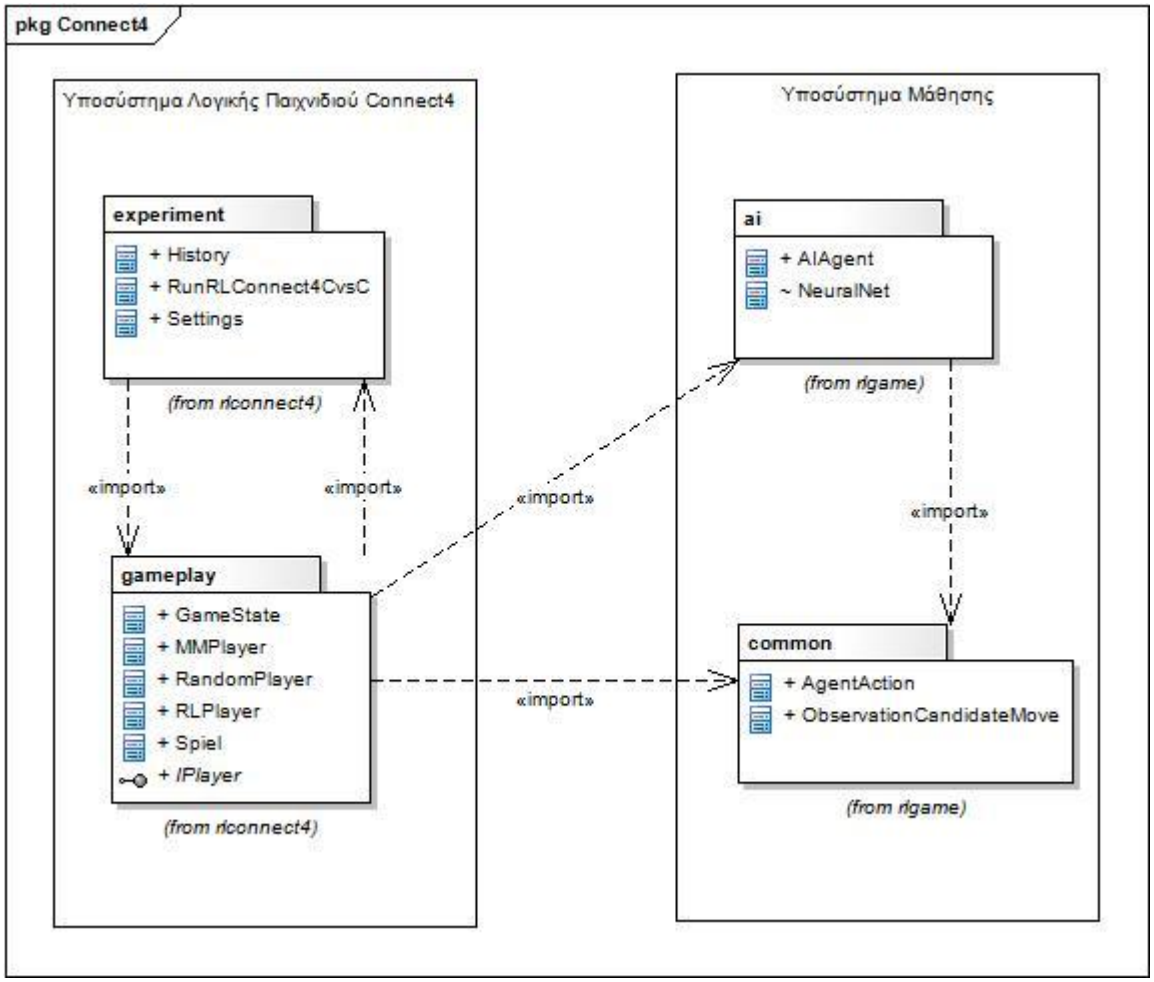
- Η προεπιλεγμένη τιμή για το κρυμμένο επίπεδο έχει ορισθεί να είναι ο αριθμός νευρώνων του επιπέδου εισόδου / 2. Στην τελική έκδοση αυτή η τιμή μπορεί να αλλάξει κατά την εκτέλεση του προγράμματος, μέσω ορισμάτων που έχουν προβλεφθεί. Η τιμή μπορεί να είναι διαφορετική για κάθε παίκτη.

7.3 Αρχιτεκτονική

Σε αυτό το τμήμα θα αναφερθούμε μόνο στο υποσύστημα λογικής παιχνιδιού του connect4 καθώς ο μηχανισμός μάθησης έχει ήδη περιγραφεί στο προηγούμενο κεφάλαιο. Ο μηχανισμός λογικής παιχνιδιού χωρίστηκε σε 2 διαφορετικές λογικές ομαδοποιήσεις:

Όνομα Πακέτου	**Περιγραφή Πακέτου**
org.rlconnect4.gameplay	Το πακέτο αυτό περιέχει τις κλάσεις που περιγράφουν την λογική του παιχνιδιού, που εφαρμόζουν τους κανόνες, εκτελούν τις κινήσεις και ελέγχουν την κατάσταση του παιχνιδιού.
org. rlconnect4.experiment	Το πακέτο αυτό περιέχει τις κλάσεις για την εκτέλεση του πειράματος.

Στο σχεδιάγραμμα της εικόνας 7.1 φαίνονται τα πακέτα που περιγράψαμε στο παραπάνω πίνακα καθώς και αυτά του μηχανισμού μάθησης. Στο σχεδιάγραμμα φαίνεται η ομαδοποίηση των πακέτων στα υποσυστήματα λογικής παιχνιδιού και μηχανισμού μάθησης. Ακόμη οι διακεκομμένες γραμμές με την ετικέτα import συμβολίζουν τις εξαρτήσεις των κλάσεων των πακέτων, δηλαδή το γεγονός ότι χρησιμοποιούν κλάσεις από άλλα πακέτα.. Το σύστημα που φαίνεται στο λογικό διάγραμμα έχει διαχωριστεί και φυσικά σε δύο ξεχωριστά προγράμματα. Το πρόγραμμα του μηχανισμού μάθησης μπορεί να θεωρηθεί ως μια βιβλιοθήκη/api (Application Programming Interface) η οποία χρησιμοποιείται από το πρόγραμμα του παιχνιδιού Connect4. Στην εκτελέσιμη μορφή του το σύστημα αποτελείται από το εκτελέσιμο jar του Connect4 που έχει εξάρτηση από το jar του μηχανισμού μάθησης.



Εικόνα 7.1 : Γενική άποψη του νέου συστήματος (Διάγραμμα πακέτων προγράμματος)

Το πακέτο εκτέλεσης του πειράματος περιέχει την κλάση που εκτελεί το πρόγραμμα (πείραμα), λαμβάνει τις παραμέτρους εκτέλεσης, αρχικοποιεί το υποσύστημα της εκτέλεσης και λογικής του παιχνιδιού και καταγράφει τα στατιστικά στοιχεία του πειράματος σε αρχεία.

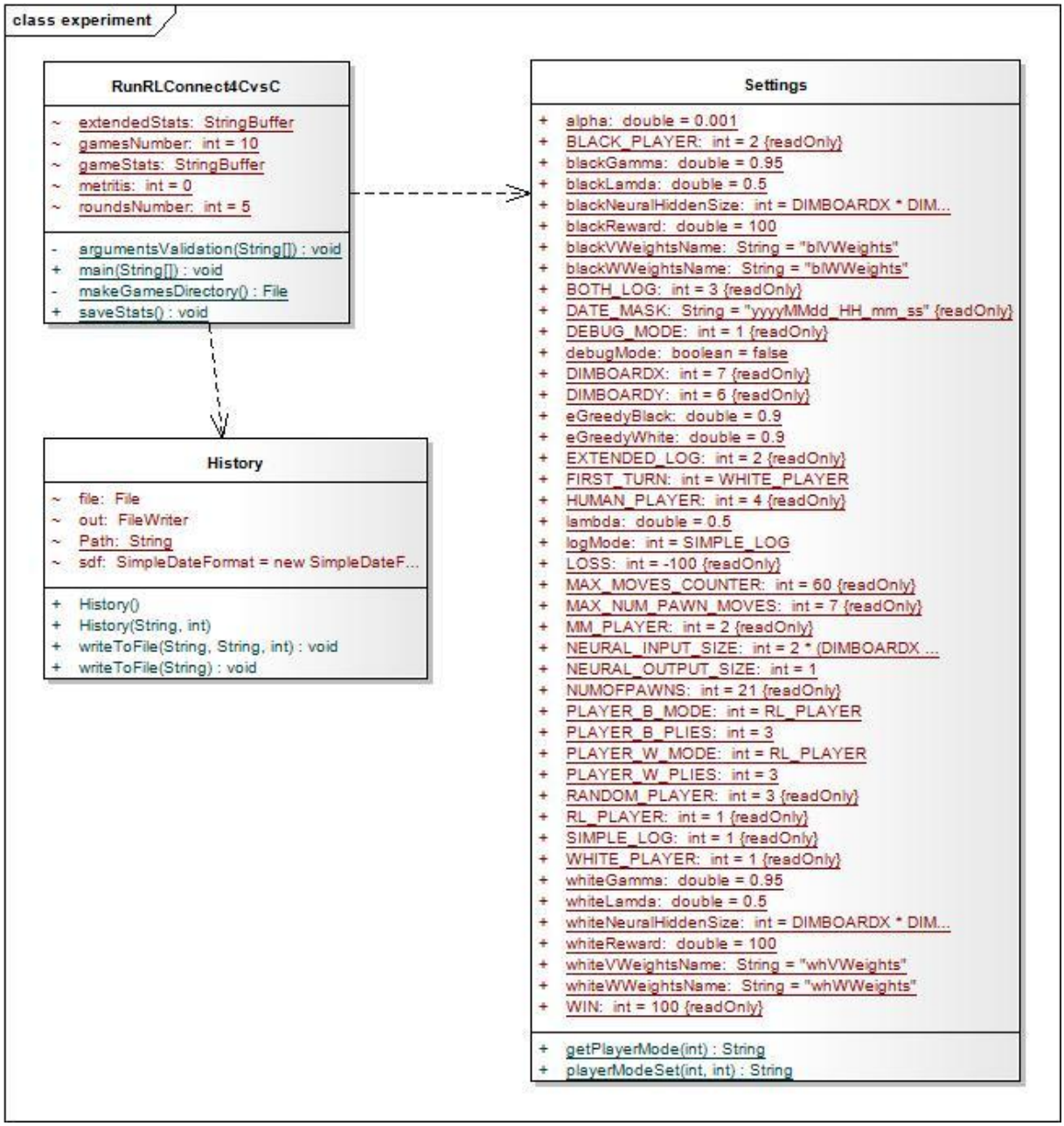
Το πακέτο εκτέλεσης και λογικής του παιχνιδιού δημιουργεί τα αντικείμενα που ορίζουν την κατάσταση του παιχνιδιού, τα δύο αντικείμενα των παικτών που θα παίξουν το παιχνίδι και αυτά με την σειρά τους (στην περίπτωση παικτών ενισχυτικής μάθησης) δημιουργούν το αντίστοιχο αντικείμενο του πράκτορα ενισχυτικής μάθησης που θα επιλέξει τις κινήσεις αυτού του υπολογιστικού παίκτη. Κατά τη διάρκεια της εκτέλεσης του παιχνιδιού το κάθε αντικείμενο παίκτη καλεί μεθόδους του αντίστοιχου αντικειμένου πράκτορα μάθησης περνώντας σαν όρισμα ή λαμβάνοντας απάντηση με κάποιο αντικείμενο από τις βοηθητικές κλάσεις του πακέτου βοηθητικών κλάσεων που ορίζονται στο υποσύστημα μάθησης.

7.3.1 Πακέτο Εκτέλεσης Πειραμάτων

Στο πακέτο εκτέλεσης πειραμάτων έχουν ορισθεί οι ακόλουθες κλάσεις:

Κλάση	**Περιγραφή**
History.java	Κλάση επιφορτισμένη με την καταγραφή των εξαγόμενων αρχείων αποτελεσμάτων των παιχνιδιών και των κινήσεων των παικτών.
RunRLConnect4CvsC.java	Η κλάση που εκτελεί το πείραμα και είναι υπεύθυνη για την λήψη των παραμέτρων εκτέλεσης, και την αρχικοποίηση παικτών του παιχνιδιού και της κλάσης που εκτελεί το παιχνίδι.
Settings.java	Κλάση στην οποία ορίζονται οι καθολικές στατικές μεταβλητές του προγράμματος και οι στατικές μεταβλητές που μπορεί να μεταβληθούν ανάλογα με τα ορίσματα εκτέλεσης. Αυτή η κλάση χρησιμοποιείται από τις κλάσεις του πακέτου πειράματος και από τις κλάσεις του πακέτου λογικής και εκτέλεσης παιχνιδιού.

Στο ακόλουθο διάγραμμα κλάσεων παρουσιάζονται οι κλάσεις του πακέτου πειράματος. Οι διακεκομμένες γραμμές συμβολίζουν τις εξαρτήσεις της κλάσης RunRLConnect4CvsC (η κλάση καλεί στατικές μεθόδους των άλλων 2 κλάσεων του πακέτου).



Εικόνα 7.2 : Διάγραμμα κλάσεων του πακέτου πειράματος

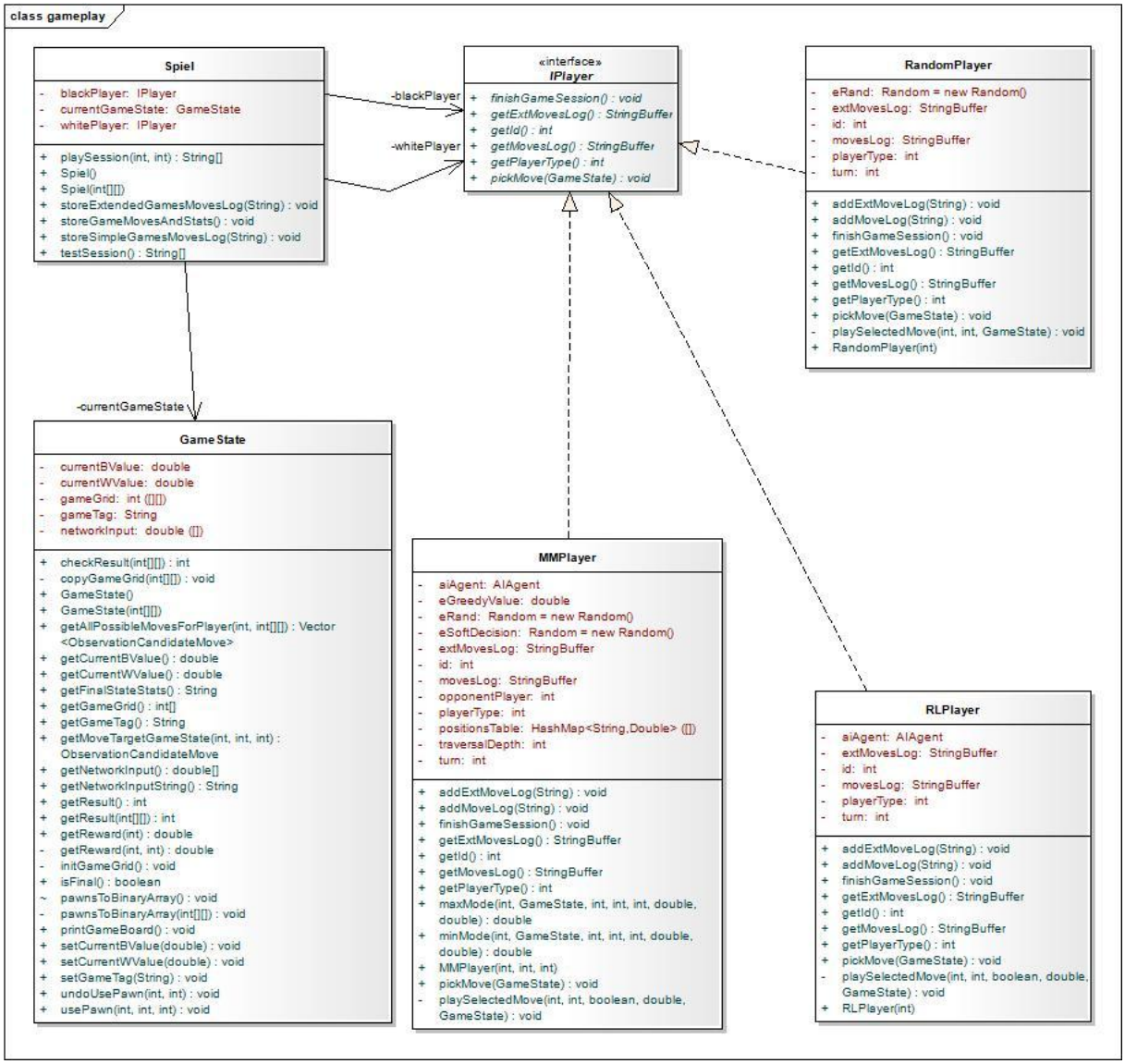
7.3.2 Πακέτο Λογικής Παιχνιδιού

Στο πακέτο λογικής παιχνιδιού έχουν ορισθεί οι ακόλουθες κλάσεις:

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 του πλαισίου του connect4

IPlayer.java (Interface)	Πρότυπο (Interface) κλάσης του παίκτη. Σε αυτό το πρότυπο στηρίζονται οι κλάσεις των παικτών που ακολουθούν.
MMPlayer.java	Η κλάση του Minimax παίκτη. Η κλάση ακολουθεί/υλοποιεί το πρότυπο IPlayer.java. Ο αλγόριθμος Minimax υλοποιείται σε αυτή την κλάση.
RandomPlayer.java	Η κλάση του παίκτη τυχαίας επιλογής. Η κλάση ακολουθεί/υλοποιεί το πρότυπο IPlayer.java.
RLPlayer.java	Η κλάση του παίκτη Ενισχυτικής Μάθησης. Η κλάση ακολουθεί/υλοποιεί το πρότυπο IPlayer.java. Η κλάση χρησιμοποιεί τις υπηρεσίες του πράκτορα ενισχυτικής μάθησης.
Spiel.java	Η κλάση εκτέλεσης ενός παιχνιδιού/επεισοδίου

Έχουν ορισθεί αντίστοιχες κλάσεις με αυτές του RLGame και οι διαφορές εντοπίζονται μόνο στο γεγονός ότι έχουν απαλειφθεί οι κλάσεις του πιονιού και του τετραγώνου της σκακιέρας καθώς στο Connect4 αυτές οι πληροφορίες της κατάστασης του παιχνιδιού μπορούν να συντηρηθούν σε έναν απλό δισδιάστατο πίνακα int όπου τα πιόνια των παικτών συμβολίζονται με 1 και 2 ανάλογα με τον παίκτη. Στο ακόλουθο διάγραμμα κλάσεων παρουσιάζονται οι κλάσεις του πακέτου λογικής του παιχνιδιού. Οι συνεχόμενες γραμμές συμβολίζουν τις σχέσεις των κλάσεων (associations) ενώ οι διακεκομμένες συμβολίζουν το γεγονός ότι οι τρεις διαφορετικές κλάσεις παικτών υλοποιούν το πρότυπο (interface) IPlayer.



Εικόνα 7.3 : Διάγραμμα κλάσεων του πακέτου λογικής παιχνιδιού

7.4 Πειραματική υλοποίηση με το RL-Glue

Όπως έχει αναφερθεί στα κεφάλαια 4 και 5 το παιχνίδι connect4 χρησιμοποιήθηκε σε μία πειραματική έκδοση που αφορούσε στην ολοκλήρωση του παιχνιδιού connect4 με τον μηχανισμό μάθησης του RLGame με χρήση του συστήματος και πρωτοκόλλου RL-Glue. Η πιλοτική έκδοση, υλοποιήθηκε στα πλαίσια της μελέτης σκοπιμότητας χρήσης του συστήματος RL-Glue σε παιχνίδια στρατηγικής 2 παικτών που χρησιμοποιούν τον μηχανισμό ενισχυτικής μάθησης του RLGame.

7.5 Εκτέλεση πειραμάτων με την νέα υλοποίηση

Για την επιβεβαίωση της ορθότητας της νέας έκδοσης εκτελέστηκε μια σειρά πειραμάτων τόσο σε παιχνίδι υπολογιστή εναντίον υπολογιστή όσο και σε παιχνίδι minimax εναντίον υπολογιστή. Τα πειράματα που παραθέτονται σε αυτή την παράγραφο, έχουν εκτελεστεί με την τελική έκδοση του Connect4. Στις ακόλουθες παραγράφους αναφέρονται τα αποτελέσματα αυτών των πειραμάτων.

7.5.1 Πειράματα σε παιχνίδι υπολογιστή εναντίον υπολογιστή

Η έκδοση αυτή ελέγχθηκε μέσω πειραμάτων σε σύνολο 50000 παιχνιδιών υπολογιστή εναντίον υπολογιστή με ένα ζεύγος νευρωνικών δικτύων που δημιουργήθηκαν για αυτό το πείραμα. Δεν υπήρξε κάποιο σφάλμα κατά τη διάρκεια εκτέλεση των πειραμάτων ενώ τα αποτελέσματα των παιχνιδιών παρατίθενται παρακάτω.

Οι συνολικές νίκες είναι 32891 για τον λευκό και 17096 για τον μαύρο ενώ ο συνολικός μέσος αριθμός κινήσεων είναι 8,02 κινήσεις.

Σετ 5000 παιχνιδιών	**Νίκες Λευκού**	**Μέσος όρος κινήσεων λευκού (νίκες)**	**Νίκες Μαύρου**	**Μέσος όρος κινήσεων μαύρου (νίκες)**	**Ισοπαλία**
1	3387	8,42	1603	8,33	10
2	3315	8,03	1683	8,25	2
3	3238	7,82	1762	8,28	0
4	3194	7,86	1806	8,14	0
5	3248	7,83	1752	8,09	0
6	3261	7,8	1738	8,15	1
7	3312	7,92	1688	8,17	0
8	3344	7,86	1656	8,16	0
9	3310	7,82	1690	8,2	0
10	3282	7,9	1718	8,21	0

Οι παρατηρήσεις από αυτό το σετ παιχνιδιών είναι ότι ο λευκός που ξεκινάει πρώτος έχει προβάδισμα ενώ ανάμεσα στα σετ παρατηρούμε σε μικρότερο βαθμό (αφού ο λευκός κερδίζει πάντα περισσότερα παιχνίδια) τις εναλλαγές στην αποτελεσματικότητα των παικτών.

7.5.2 Πειράματα σε παιχνίδι minimax εναντίον υπολογιστή

Η έκδοση αυτή ελέγχθηκε μέσω πειραμάτων σε σύνολο 5000 παιχνιδιών minimax (λευκός παίκτης) εναντίον υπολογιστή (μαύρος παίκτης) με ένα ζεύγος νευρωνικών δικτύων που δημιουργήθηκαν για αυτό το πείραμα με τις προεπιλεγμένες τιμές των ρυθμίσεων και τον minimax να ψάχνει σε βάθος 3. Δεν υπήρξε κάποιο σφάλμα κατά τη διάρκεια εκτέλεση των πειραμάτων ενώ τα αποτελέσματα των παιχνιδιών παρατίθενται παρακάτω.

Οι συνολικές νίκες είναι 4282 για τον λευκό και 708 για τον μαύρο ισοπαλία 10 ενώ ο συνολικός μέσος αριθμός κινήσεων είναι 10,16 κινήσεις.

Στον παρακάτω πίνακα φαίνονται τα αποτελέσματα ανά σετ 500 παιχνιδιών.

Σετ 5000 παιχνιδιών	**Νίκες Λευκού**	**Μέσος όρος κινήσεων λευκού (νίκες)**	**Νίκες Μαύρου**	**Μέσος όρος κινήσεων μαύρου (νίκες)**	**Ισοπαλία**
1	384	10,99	111	12,34	5
2	388	10,79	111	12,71	1
3	441	9,85	59	11,63	
4	439	9,55	61	11,52	
5	443	9,88	56	12,04	1
6	435	9,75	64	11,61	1
7	446	9,51	54	11,98	
8	431	9,54	68	12,18	1
9	431	9,52	68	11,47	1
10	444	9,3	56	11,09	

Οι παρατηρήσεις από αυτό το σετ παιχνιδιών είναι ότι όπως αναμενόταν, ο λευκός που ξεκινάει πρώτος και έχει την βοήθεια του minimax έχει μεγαλύτερο προβάδισμα από το προβάδισμα του λευκού σε παιχνίδι υπολογιστή εναντίον υπολογιστή.

Κεφάλαιο 8
Συμπεράσματα

8.1 Απολογισμός

Στα πλαίσια της παρούσας διατριβής διερευνήθηκε καταρχήν η επανασχεδίαση της αρχιτεκτονικής του συστήματος διεξαγωγής πειραμάτων RLGame έτσι ώστε η υλοποίηση του παιχνιδιού και η υλοποίηση της Ενισχυτικής Μάθησης να αποτελούν ξεχωριστά τμήματα. Σε δεύτερο επίπεδο ζητήθηκε η ολοκλήρωση του υποσυστήματος μάθησης που προέκυψε από τον διαχωρισμό των τμημάτων του RLGame με υλοποίηση άλλου παιχνιδιού στρατηγικής. Όπως έχει περιγραφεί και στα προηγούμενα κεφάλαια στα τελικά παραδοτέα παραδόθηκε νέα υλοποίηση στην οποία ο μηχανισμός μάθησης έχει διαχωριστεί πλήρως από την υλοποίηση του παιχνιδιού. Ο μηχανισμός μάθησης έχει πλέον ορισθεί με συγκεκριμένες διεπαφές/μεθόδους και μπορεί να χρησιμοποιηθεί από παιχνίδια που εμπλέκουν 1 ή περισσότερους παίκτες. Ο μηχανισμός χρησιμοποιείται από το RLGame και επιπλέον από μία νέα υλοποίηση του παιχνιδιού connect4. Το connect4 χρησιμοποιήθηκε για να επιβεβαιωθεί το δεύτερο ζητούμενο, να αποδειχθεί δηλαδή η δυνατότητα ολοκλήρωσης του μηχανισμού με άλλο παιχνίδι στρατηγικής.

Τέλος το τρίτο ζητούμενο ήταν η μελέτη της δυνατότητας/σκοπιμότητας της επανασχεδίασης και επανυλοποίησης του RLGame με χρήση συστημάτων (πλατφορμών) περιγραφής και ανάπτυξης παιχνιδιών ανοιχτού κώδικα. Στα πλαίσια αυτού του ζητούμενου μελετήθηκαν το RL-

Glue, το GameMaker, η πλατφόρμα ORTS και το παράδειγμα General Game Playing. Για διαφορετικούς λόγους που περιγράφηκαν στο αντίστοιχο κεφάλαιο δεν κρίθηκε σκόπιμο να ακολουθηθεί κάποιο από αυτά τα παραδείγματα στην νέα υλοποίηση. Παρόλο που το RL-Glue δεν χρησιμοποιήθηκε για το τελικό παραδοτέο, οι πειραματισμοί μου με το αυτό μου έδωσαν ιδέες για το πως μπορώ να διαχωρίσω το RLGame και να κάνω ένα πιο γενικό υποσύστημα ενισχυτικής μάθησης.

Το βασικό συμπέρασμα ήταν ότι για να δοκιμάσουμε ένα άλλο παιχνίδι με το μηχανισμό μάθησης του RLGame είναι προτιμότερο να το κάνουμε με την υλοποίηση του υποσυστήματος μάθησης που παραδόθηκε στα πλαίσια της παρούσας διπλωματικής διατριβής. Στο παραδοτέο, το υποσύστημα του παιχνιδιού χρησιμοποιεί τις υπηρεσίες του μηχανισμού μάθησης. Οποιαδήποτε υλοποίηση παιχνιδιού σε Java ή γλώσσα JVM μπορεί να προσαρμοστεί για να χρησιμοποιεί αυτές τις υπηρεσίες, και να αποκτήσει έναν υπολογιστικό πράκτορα που κάνει χρήση των τεχνικών ενισχυτικής μάθησης που χρησιμοποιούνται στο RLGame.

Βιβλιογραφία

[01] 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

[02] 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

[03] 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

[04] 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

[05] D. Kalles and I. Fykouras. "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

[06] Ντούτση Ε. «Μοντελοποίηση και βελτίωση της ανθρώπινης ικανότητας σε παιχνίδια στρατηγικής» Διπλωματική Εργασία Πανεπιστήμιο Πατρών, τμήμα Μηχανικών Η/Υ και Πληροφορικής, (2001).

[07] R.S. Sutton & A.G. Barto. "Reinforcement Learning - An Introduction", MIT Press, Cambridge, Massachusetts, 1998.

[08] G. Tesauro. "Temporal Difference Learning and TD-Gammon", Communications of the ACM, March 1995/Vol. 38, No 3.

[09] R.S. Sutton. "Learning to Predict by the Methods of Temporal Differences", Machine Learning 3: 9-44, 1988.

[10] Brian Tanner and Adam White. RL-Glue: Language-Independent Software for Reinforcement-Learning Experiments. Journal of Machine Learning Research, 10(Sep):2133--2136, 2009.

[11] http://games.stanford.edu/competition/misc/aaai.pdf

[12] http://www.gamemaker.nl

[13] http://en.wikipedia.org/wiki/Game_Maker

[14] http://en.wikipedia.org/wiki/Connect_Four

[15] http://en.wikipedia.org/wiki/Code_refactoring

[16] http://en.wikipedia.org/wiki/Design_pattern_%28computer_science%29

[17] http://en.wikipedia.org/wiki/Singleton_pattern

[18] http://en.wikipedia.org/wiki/Adapter_pattern

[19] http://en.wikipedia.org/wiki/Facade_pattern

[20] http://en.wikipedia.org/wiki/Zero%E2%80%93sum_game

[21] Allis, V. A Knowledge-based Approach of Connect-Four. Master Thesis, Department of Mathematics and Computer Science, Vrije Universiteit, Amsterdam, The Netherlands.

[22] https://rl-glue.googlecode.com/svn/trunk/docs/Glue-Overview.tex

[23] http://glue.rl-community.org/Home/rl-glue#TOC-Frequently-Asked-Questions

[24] Jacek Mandziuk, "Knowledge-Free and Learning-Based Methods in Intelligent Game Playing", Springer, 2009

[25] http://en.wikipedia.org/wiki/Game_theory

[26] S. Demeyer, S. Ducasse, O. Nierstrasz, "Object-Oriented Reengineering Patterns", http://www.iam.unibe.ch/~scg/OORP/, 01/06/2008

Παράρτημα Α
Γλωσσάρι

Α.1 Γλωσσάρι Όρων Τεχνολογίας Λογισμικού

Στο παράρτημα παραθέτονται περιγραφές κάποιων όρων σχετικών με την τεχνολογία λογισμικού που ενδεχομένως έχουν αναφερθεί στην παρούσα διπλωματική διατριβή. Κατά την εκπόνηση της διπλωματικής διατριβής ασχολήθηκα με τις έννοιες/τεχνικές που αναφέρονται σε αυτό το παράρτημα και έκρινα σκόπιμο να τις συμπεριλάβω σε αυτό το παράρτημα ακόμα και εάν κάποιες από αυτές τις τεχνικές δεν χρησιμοποιήθηκαν στα τελικά παραδοτέα.

Refactoring: Ο όρος refactoring [15] για τον οποίο χρησιμοποιώ τον ελληνικό όρο αναδιάταξη/αναδιάρθρωση αναφέρεται στη συστηματική αναδιάρθρωση υπάρχοντος κώδικα με στόχο την αλλαγή στην εσωτερική δομή του κώδικα, χωρίς να αλλάζει η εξωτερική συμπεριφορά της υλοποίησης. Η συστηματική αυτή τεχνική βασίζεται σε μια σειρά μικρών αλλαγών που δεν επηρεάζουν την εξωτερική συμπεριφορά του υπάρχοντος συστήματος. Μια σειρά από μικρές αλλαγές μπορεί να επιφέρει μεγάλη βελτίωση στην εσωτερική δομή του συστήματος ενώ βασική αρχή είναι το σύστημα να παραμένει λειτουργικό μετά το πέρας κάθε μικρής αλλαγής.

Design Patterns: Ο όρος Design Patterns (Πρότυπα Σχεδίασης Λογισμικού) [16] χρησι**μ**οποιείται στην τεχνολογία λογισμικού για κάποιες γενικής φύσης λύσεις σε προβλήματα σχεδίασης λογισμικού. Τα περισσότερα πρότυπα αυτά σχέδια είναι συνυφασμένα με τον αντικειμενοστραφή προγραμματισμό. Τα πρότυπα αυτά έχουν ταξινομηθεί σε

- πρότυπα δημιουργίας, που δίνουν λύσεις σχετικά με τους μηχανισμούς δημιουργίας των αντικειμένων
- πρότυπα δόμησης που δίνουν λύσεις στην υλοποίηση των σχέσεων μεταξύ των αντικειμένων
- πρότυπα συμπεριφοράς που δίνουν λύσεις στην επικοινωνία μεταξύ των αντικειμένων

Facade: Σε αυτό το πρότυπο μία κεντρική κλάση υλοποιεί μεθόδους οι οποίες αφορούν σε μεθόδους και συμπεριφορές περισσότερων κλάσεων. Η façade κλάση είναι στην ουσία μια πρόσοψη η οποία κρύβει από τον πελάτη των μεθόδων της το πώς υλοποιείται μια μέθοδος (Ο πελάτης ξέρει μόνο τις μεθόδους την πρόσοψης) [19].

Singleton: Με αυτή τη σχεδίαση εξασφαλίζουμε ότι μόνο ένα αντικείμενο θα δημιουργηθεί για μια συγκεκριμένη κλάση. Κατόπιν διάφορες κλάσεις θα μπορούν να χρησιμοποιούν το μοναδικό αυτό αντικείμενο[17].

Adapter: Αυτή η σχεδίαση δίνει λύσεις στην επικοινωνία 2 κλάσεων που δεν θα μπορούσαν να επικοινωνήσουν λόγω ασυμβατότητας των υπογραφών και των αντικειμένων που αναμένονται από την κάθε μια. Η κλάση του adapter δίνει στην κλάση πελάτη το αντικείμενο και τις υπογραφές που χρειάζεται κρύβοντας το αντικείμενο και τις υπογραφές της κλάσης που απαντάει[18].

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.

Java Interface: Ένα Java interface ορίζει ένα πρότυπο μία κλάσης χωρίς να ορίζει κάποια υλοποίηση. Οι κλάσεις που υλοποιούν το πρότυπο υποχρεούνται να παρέχουν τις υλοποιήσεις των μεθόδων των οποίων οι υπογραφές έχουν ορισθεί στο πρότυπο. Τα πρότυπα αυτά

χρησιμοποιούνται για να ορίσουν μία γενική δομή ή ένα συμβόλαιο την οποία μπορούν να χρησιμοποιήσουν άλλα προγράμματα. Τα προγράμματα πελάτες μπορούν να στηρίξουν την λογική τους στο συμβόλαιο του Java interface χωρίς να γνωρίζουν κάτι για την κλάση που ικανοποιεί τις κλήσεις τους.

Java Jar: Το Jar είναι μία μορφή πακεταρίσματος των κλάσεων που αποτελούν ένα πρόγραμμα ή των κλάσεων μίας βιβλιοθήκης που μπορεί να χρησιμοποιηθεί από κάποιο άλλο πρόγραμμα. Ένα jar είναι ουσιαστικά ένα συμπιεσμένο αρχείο που περιέχει τη δομή των αρχείων των κλάσεων. Το jar περιέχει και ένα ειδικό αρχείο το META_INF/MANIFEST.MF στο οποίο μπορεί να ορισθεί η εκτελέσιμη κλάση του jar ή οι κλάσεις/jars που πρέπει να είναι ορατά σε αυτό.

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 την βιβλιοθήκη κλάσεων στα προγράμματα τους.

Παράρτημα Β
Τεχνικό Εγχειρίδιο Υλοποίησης

Β.1 Οδηγίες Δημιουργίας Εκτελέσιμου Προγράμματος

Τα προγράμματα που παραδόθηκαν με την παρούσα διπλωματική διατριβή έχουν δημιουργηθεί σε γλώσσα προγραμματισμού JAVA. Ο κώδικας είναι συμβατός μέχρι την JAVA έκδοση 1.5 ενώ τα εκτελέσιμα των παραδοτέων έχουν μεταγλωττιστεί σε περιβάλλον JAVA SDK 1.6. Ο κώδικας των προγραμμάτων έχει δημιουργηθεί σε περιβάλλον Eclipse. Το Eclipse είναι μια εφαρμογή ανοιχτού κώδικα που χρησιμοποιείται για τη συγγραφή και εκτέλεση κώδικα σε διάφορες γλώσσες προγραμματισμού. .Οι εφαρμογές αυτής της κατηγορίας ονομάζονται Ολοκληρωμένα Περιβάλλοντα Ανάπτυξης (Integrated Development Environment) και δίνουν στο προγραμματιστή ένα ολοκληρωμένο περιβάλλον για την διαχείριση τη μεταγλώττιση, τον έλεγχο και την εκτέλεση προγραμμάτων. Στο Eclipse τα προγράμματα οργανώνονται σε projects στα οποία υπάρχει τόσο ο κώδικα όσο και πληροφορίες για την οργάνωση του προγράμματος, εξαρτήσεις από άλλα προγράμματα ή βιβλιοθήκες, ρυθμίσεις μεταγλώττισης, ελέγχου και εκτέλεσης. Τα παραδοτέα προγράμματα έχουν δοθεί με την μορφή eclipse java project (δηλαδή

εκτός από τον κώδικα και τα εκτελέσιμα επικοινωνούν και τα αρχεία του που κρατούν τις πληροφορίες οργάνωσης του project). Σε αυτή την παράγραφο θα αναφερθούν τα βήματα για την εισαγωγή των προγραμμάτων στο περιβάλλον του eclipse, την μεταγλώττιση και την εκτέλεση των προγραμμάτων.

Ο κώδικας είναι οργανωμένος στα ακόλουθα έργα (projects):

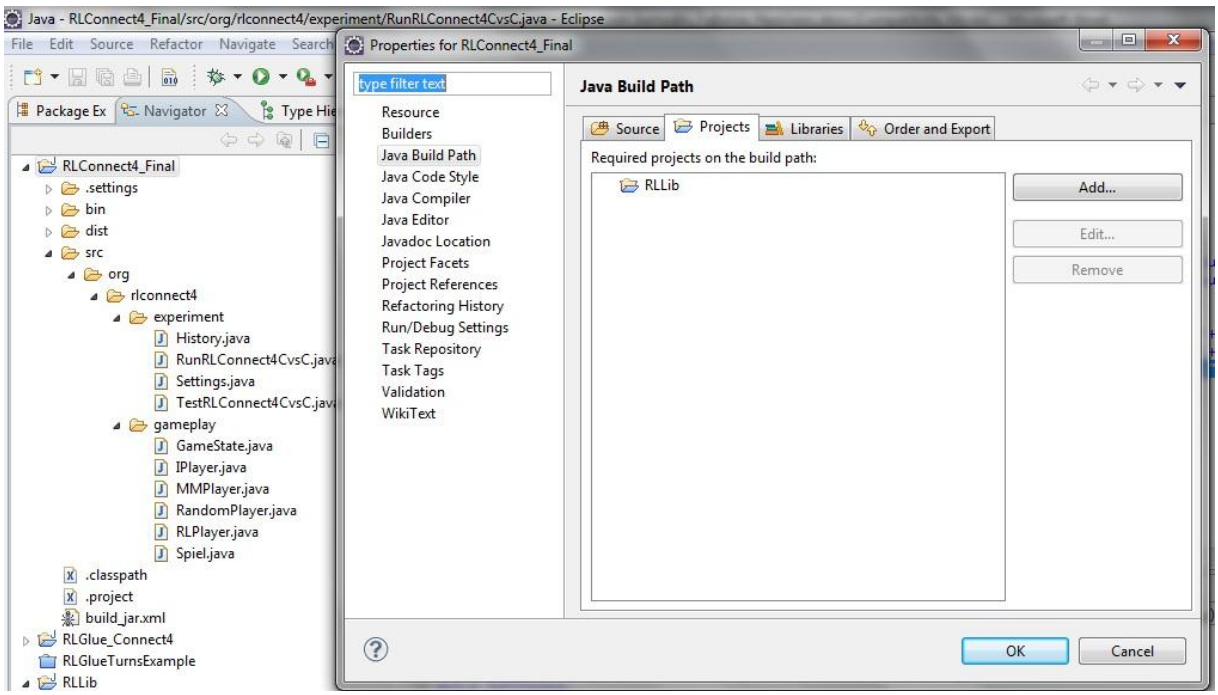
- RLGame_Final: πρόγραμμα εκτέλεσης πειραμάτων RLGame. Το πρόγραμμα χρησιμοποιεί/εξαρτάται από το πρόγραμμα/βιβλιοθήκη RLLib. Το RLLib χρειάζεται τόσο κατά την εκτέλεση των πειραμάτων όσο και κατά την μεταγλώττιση (compile) του προγράμματος.

- RLConnect4_Final: πρόγραμμα εκτέλεσης πειραμάτων RLConnect4. Το πρόγραμμα χρησιμοποιεί/εξαρτάται από το πρόγραμμα/βιβλιοθήκη RLLib. Το RLLib χρειάζεται τόσο κατά την εκτέλεση των πειραμάτων όσο και κατά την μεταγλώττιση (compile) του προγράμματος.

- RLLib: πρόγραμμα μηχανισμού μάθησης.

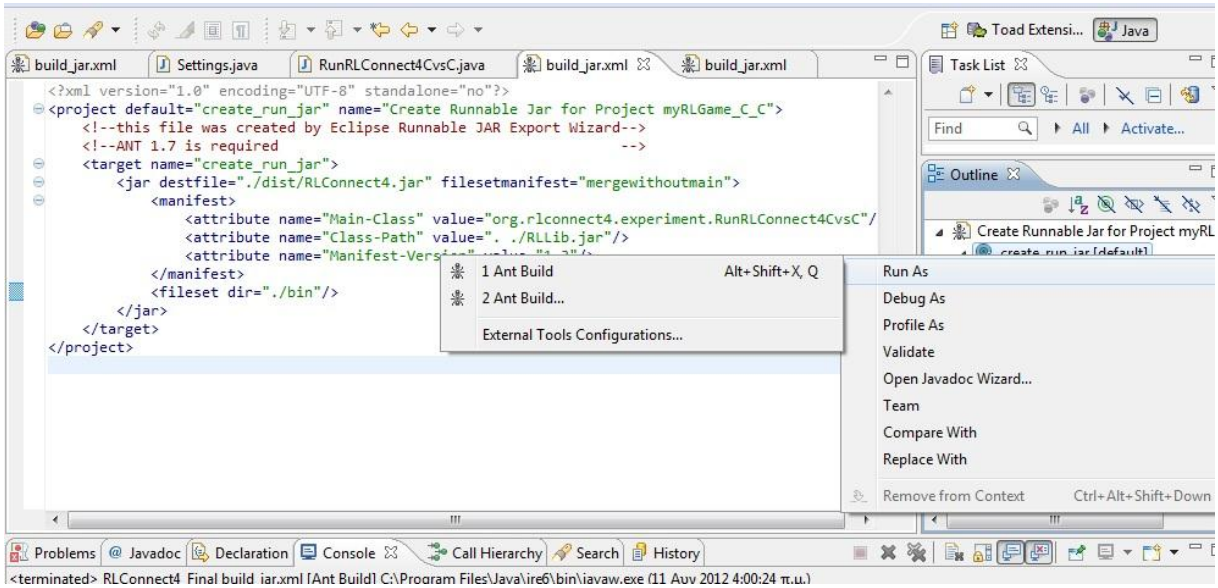
- RLGlue_Connect4: πειραματική υλοποίηση του connect 4 με την έκδοση alpha του διαχωρισμένου μηχανισμού μάθησης.

Το κάθε ένα από τα παραπάνω προγράμματα μπορεί να αντιγραφεί στο σύστημα και να εισαχθεί στο περιβάλλον του eclipse μέσω της επιλογή File-> Import Project. Η διαδικασία είναι δομημένη σε μια σειρά βημάτων και σχετικά αυτοματοποιημένη. Οι μόνες επιπλέον ρυθμίσεις που μπορεί να χρειάζονται είναι επιβεβαίωση της εξάρτησης του RLGame_Final και του RLConnect4_Final με το project του RLLib και η αλλαγή του μονοπατιού στο οποίο θα δημιουργηθεί το συμπιεσμένο αρχείο jar (εκτελέσιμο ή βιβλιοθήκη) του προγράμματος.



Εικόνα Β.1 : Eclipse διάλογος ρυθμίσεων project

Μέσα από τις επιλογές του Eclipse ο προγραμματιστής μπορεί να εκτελέσει το πρόγραμμα, να αλλάξει μαζικά ονοματολογία και δομή του προγράμματος με τη βοήθεια των επιλογών του refactoring, και να πακετάρει τον εκτελέσιμο στη μορφή jar με τη βοήθεια του apache Ant make/build εργαλείου (build_jar.xml).



Εικόνα Β.2 : Επιλογή δημιουργίας εκτελέσιμου jar μέσω του Apache Ant xml script

Β.2 Οδηγίες Ολοκλήρωσης Νέου Παιχνιδιού με τον μηχανισμό Μάθησης

Στην παρούσα μορφή, η βιβλιοθήκη του μηχανισμού μάθησης μπορεί να χρησιμοποιηθεί από υλοποίηση άλλου παιχνιδιού ή πειράματος που εκτελείται σε περιβάλλον Java JVM (Java, .Scala, Groovy κ.λ.π.).

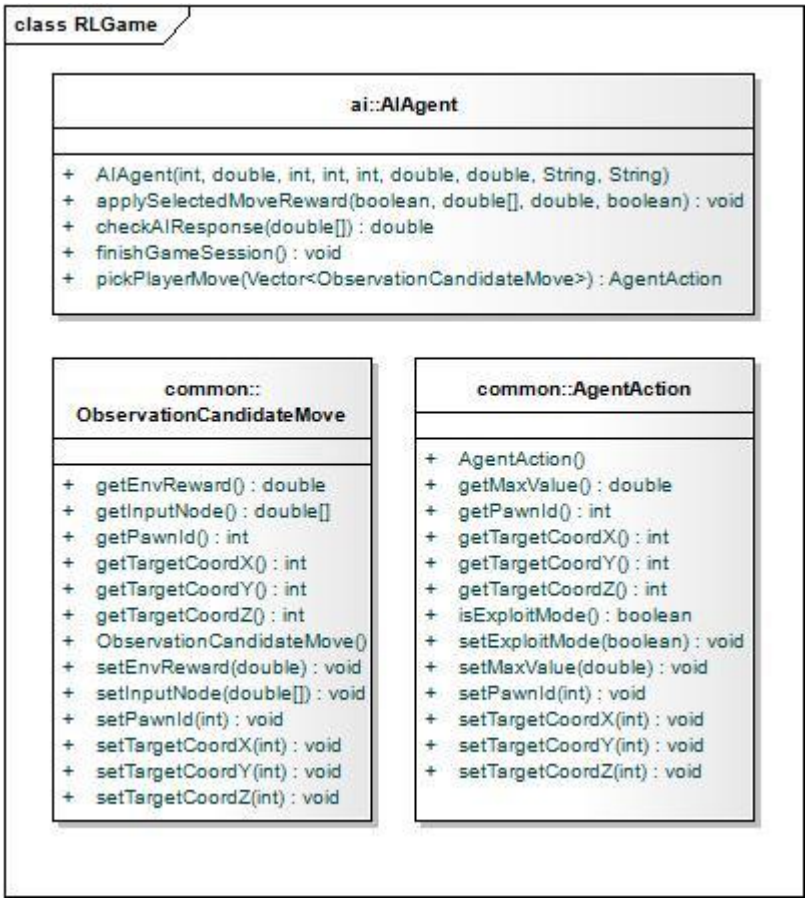
Σε αυτό το παράρτημα περιγράφεται το API της βιβλιοθήκης μάθησης και ο τρόπος που μπορούν να κληθούν από ένα άλλο πρόγραμμα. Όπως έχει περιγραφεί και στο κεφάλαιο 6 το API αποτελείται από 4 κλάσεις οι οποίες πακετάρονται στο RLLib.jar:

Δομή Μηχανισμού

Κλάση	**Περιγραφή**
AIAgent.java	Η κλάση του πράκτορα ενισχυτικής μάθησης. έχει ορισμένες δημόσιες μεθόδους που χρησιμοποιούνται από το πρόγραμμα πελάτης που χρησιμοποιεί τον μηχανισμό. Κάθε αντικείμενο πράκτορα που αποκτά υπόσταση δημιουργεί και ένα αντικείμενο της κλάσης του Νευρωνικού Δικτύου. 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.
ObservationCandidateMove.java	Δημόσια 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 (Στις υλοποιήσεις του RLGame και του Connect4 αποστέλλεται η ανταμοιβή του περιβάλλοντος στην παρατήρηση, ένα άλλο πρόγραμμα που θα χρησιμοποιήσει τον μηχανισμό θα αποφασίσει εάν θα περνάει πάντα την τιμή 0 έτσι ώστε να λάβει επιλογή που θα προκύπτει μόνο από την αξιολόγηση του Νευρωνικού Δικτύου). Ακόμη έχουν ορισθεί οι 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 (διαθέσιμη για μελλοντική χρήση σε παιχνίδι που χρειάζεται και Τρίτη διάσταση (π.χ. ύψος ή θέση στην στοίβα).
AgentAction.java	Δημόσια 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.
NeuralNet.java	Κλάση με εμβέλεια πακέτου(org.rlgame.ai) που χρησιμοποιείται για την υλοποίηση των δομών και των αλγορίθμων του νευρωνικού δικτύου. Η κλάση αυτή 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 του API (της κλάσης του πράκτορα ενισχυτικής μάθησης).

Στο ακόλουθο σχεδιάγραμμα φαίνονται οι δημόσιες μέθοδοι των κλάσεων της βιβλιοθήκης τις οποίες θα πρέπει να χρησιμοποιήσει κάποιο πρόγραμμα που επιλέγει να χρησιμοποιήσει την βιβλιοθήκη του μηχανισμού.



Εικόνα Β.3 : Δημόσιες κλάσεις και μέθοδοι του RLLib API

Εξαρτήσεις προγραμμάτων παιχνιδιών

Ένα πρόγραμμα παιχνιδιού όταν ολοκληρώνεται με τον μηχανισμό μάθησης έχει τις εξής εξαρτήσεις:

- Το jar του μηχανισμού μάθησης και η υλοποίηση της κλάσης του πράκτορα ενισχυτικής μάθησης όσο και των βοηθητικών κλάσεων της παρατήρησης και της επιλεγμένης ενέργειας, χρειάζεται να είναι διαθέσιμη κατά την μεταγλώττιση του προγράμματος του παιχνιδιού
- Το jar του μηχανισμού μάθησης χρειάζεται να είναι διαθέσιμο κατά την εκτέλεση του προγράμματος του παιχνιδιού
- Το υποσύστημα του παιχνιδιού καλεί τον πράκτορα να επιλέξει κίνηση επικοινωνώντας την παρατήρηση του περιβάλλοντος και λαμβάνει σαν απάντηση την επιλεγμένη κίνηση. Τόσο η παρατήρηση όσο και επιλεγμένη κίνηση επικοινωνούνται με αντικείμενα του πακέτου βοηθητικών κλάσεων του μηχανισμού μάθησης.

Επιλογές ρυθμίσεων προγραμμάτων παιχνιδιών

Ένα πρόγραμμα παιχνιδιού όταν ολοκληρώνεται με τον μηχανισμό μάθησης πρέπει να αποφασίσει για τα εξής:

- Το ποσοστό εκμετάλλευσης γνώσης που θα χρησιμοποιηθεί από τον πράκτορα στην επιλογή κίνησης (ε-greedy policy). Στο υπολειπόμενο ποσοστό ο πράκτορας θα εξερευνεί νέες κινήσεις (θα επιλέγει τυχαία μία από τις δυνατές επόμενες κινήσεις).
- Τον αριθμό των νευρώνων στα τρία επίπεδο του Νευρωνικού Δικτύου
    - Αριθμός νευρώνων και δομή δυαδικού πίνακα του επιπέδου εισόδου. Το πρόγραμμα του παιχνιδιού πρέπει να αποφασίσει για τον τρόπο που η κατάσταση του παιχνιδιού μεταφέρεται στον δυαδικό πίνακα που περιμένει το Νευρωνικό σαν είσοδο.
    - Αριθμό νευρώνων κρυμμένου επιπέδου.
    - Αριθμός νευρώνων επιπέδου εξόδου.
- τιμή για την παράμετρο του ρυθμού μείωσης γ (discount rate parameter) που καθορίζει την αξία των μελλοντικών αμοιβών.
- τιμή για τον παράγοντα παράληψης (forgetting factor) λ που χρησιμοποιείται από την online έκδοση του αλγόριθμου μάθησης χρονικών διαφορών (temporal difference learning algorithm) TD (λ).
- Την εμβέλεια των ανταμοιβών του περιβάλλοντος.
- Ονοματολογία αρχείων στα οποία αποθηκεύονται τα βάρη του Νευρωνικού.

Περιγραφή Μεθόδων

Σε αυτή την παράγραφο δίνεται η σειρά των ενεργειών που εκτελεί το πρόγραμμα πελάτης.

- Καταρχήν το αντικείμενο του πράκτορα πρέπει να δημιουργηθεί από το πρόγραμμα πελάτη με την κλήση του ακόλουθου constructor method. Η μέθοδος καλείται με τα εξής ορίσματα : turn = παίκτης (πλέον χρησιμοποιείται μόνο σαν ετικέτα, ο πράκτορας και το νευρωνικό δεν χρησιμοποιούν αυτή την πληροφορία στη λογική τους), eGreedyValue=ποσοστό εκμετάλλευσης γνώσης (χρησιμοποιείται από τον πράκτορα

στην επιλογή κίνησης), input = αριθμός νευρώνων επιπέδου εισόδου (όπως όλα τα επόμενα ορίσματα χρησιμοποιείται από το νευρωνικό), hidden = αριθμός νευρώνων κρυμμένου επιπέδου, output= αριθμός νευρώνων επιπέδου εξόδου, gamma=τιμή γ, lambda= τιμή λ, vWeightsName =Όνομα αρχείου (Το αρχείο θα δημιουργηθεί στον φάκελο εκτέλεσης), wWeightsName=Όνομα αρχείου

- **public AIAgent(int turn, double eGreedyValue, int input, int hidden, int output, double gamma, double lambda, String vWeightsName, String wWeightsName);**

- Κατόπιν το πρόγραμμα πελάτης μπορεί να καλεί την μέθοδο επιλογής κίνησης. Η κλήση γίνεται με όρισμα μία δομή τύπου vector (λίστα) με τις επιτρεπτές κινήσεις. Η κάθε κίνηση επικοινωνείται με ένα αντικείμενο της κλάσης ObservationCandidateMove ενώ η απάντηση της κλήσης επιστρέφεται με ένα αντικείμενο της κλάσης AgentAction.

  - **public AgentAction pickPlayerMove(Vector<ObservationCandidateMove> movesVector);**

- Μετά από κάθε κίνηση που επιλέγει ο πράκτορας, το πρόγραμμα πελάτης εκτελεί την κίνηση και καλεί τη μέθοδο απόδοσης της ανταμοιβή του περιβάλλοντος. Η μέθοδος καλείται με τα εξής ορίσματα : exploitMode= εάν η επιλογή της κίνησης ήταν αποτέλεσμα εκμετάλλευσης, networkInput = δυαδικός πίνακας/είσοδος του νευρωνικού δικτύου που αναπαριστά την κατάσταση του παιχνιδιού μετά την κίνηση, environmentReward=ανταμοιβή του περιβάλλοντος, isFinal=σημαία τερματικής κατάστασης.

  - **public void applySelectedMoveReward(boolean exploitMode, double [] networkInput, double environmentReward, boolean isFinal);**

- Όταν τελειώνει ένα επεισόδιο το πρόγραμμα πελάτης καλεί τη μέθοδο που διαχειρίζεται την αποθήκευση των βαρών του Νευρωνικού στα αντίστοιχα αρχεία.

  - **public void finishGameSession();**

- Τέλος υπάρχουν περιπτώσεις που το πρόγραμμα πελάτης μπορεί να χρειάζεται την απάντηση του Νευρωνικού για μία συγκεκριμένη κατάσταση (Στο RLGame & Connect4 χρησιμοποιούνται στην περίπτωση του minimax παίκτη). Σε αυτή την περίπτωση καλεί την ακόλουθη μέθοδο

    - **public double checkAIResponse(double [] networkInput);**

B.3 Οδηγίες Εκτέλεσης Πειραμάτων

Τα προγράμματα που έχουν παραδοθεί στα πλαίσια της διπλωματικής εργασίας δεν έχουν κάποιο UI και οι ρυθμίσεις τους δίνονται κατά την εκτέλεση του προγράμματος από την γραμμή εντολών. Στις ακόλουθες παραγράφους περιγράφονται οι τρόποι κλήσης των προγραμμάτων RLGame και Connect4.

Πρόγραμμα RLGame

Για το RLGame έχουν παραδοθεί 2 διαφορετικές εκδόσεις για σκακιέρα μεγέθους 8x8 με βάση μεγέθους 2x2 και σκακιέρα μεγέθους 6x6 με βάση μεγέθους 2x2. Τα εκτελέσιμα σε μορφή jar είναι τα : RLGameBoard8.jar και RLGameBoard6.jar. Το jar του παιχνιδιού πρέπει να βρίσκεται στον ίδιο φάκελο με το jar του μηχανισμού μάθησης (RLLib.jar) κατά την εκτέλεση του πειράματος. Όλα τα αρχεία και οι φάκελοι που δημιουργούνται κατά την εκτέλεση του προγράμματος βρίσκονται κάτω από τον φάκελο εκτέλεσης. Στην ακόλουθη λίστα δίνονται 2 εναλλακτικές κλήσεις του προγράμματος και κατόπιν παρατίθενται οι διαθέσιμες ρυθμίσεις του RLGame.

- Παιχνίδι 8x8 μεταξύ παικτών ενισχυτικής μάθησης για πέντε γύρους των 100 παιχνιδιών

    - java -jar RLGameBoard8.jar -r 5 -g 100

- Παιχνίδι 6x6 μεταξύ λευκού minimax με βάθος αναζήτησης 5 και μαύρου παίκτη ενισχυτικής μάθησης για 10 γύρους των 50 παιχνιδιών

    - java -jar RLGameBoard6.jar -r 10 -g 50 -pw 2 -dw 5

Προϋπάρχουσες ρυθμίσεις

-h : Επιλογή Βοήθειας

-r : Επιλογή αριθμού γύρων

-g : Επιλογή αριθμού παιχνιδιών που θα εκτελεστούν ανά γύρο

-wg, -bg: τιμή για την παράμετρο του ρυθμού μείωσης γ (discount rate parameter) του λευκού(**-wg**), μαύρου (**-bg**) παίκτη

-wl, -bl: τιμή για τον παράγοντα παράληψης (forgetting factor) λ του λευκού (**-wl**), μαύρου (**-bl**) παίκτη

-wr, -br: Η τιμή που της ανταμοιβής του περιβάλλοντος σε περίπτωση νίκης (σε περίπτωση ήττας αποδίδεται η αντίστοιχη αρνητική τιμή) του λευκού (**-wr**), μαύρου (**-br**) παίκτη

-we, -be: Η τιμή του ποσοστού εκμετάλλευσης γνώσης που θα χρησιμοποιηθεί από τον λευκό (**-we**), μαύρο (**-be**) παίκτη στην επιλογή κίνησης (ε-greedy policy).

Νέες ρυθμίσεις στην παραδοτέα έκδοση

-ft: Επιλογή του παίκτη που θα κάνει την πρώτη κίνηση, 1 - λευκός, 2 – μαύρος. Στην απουσία ρύθμισης πρώτος ξεκινάει ο λευκός.

-pw, -pb: Επιλογή τύπου παίκτη για τον λευκό (**-pw**) και μαύρο παίκτη(**-pb**) 1 – Παίκτης Ενισχυτικής Μάθησης, 2 – Αλγόριθμος Minimax, 3 – Τυχαίας επιλογής, 4 – Άνθρωπος (δεν υποστηρίζεται στην υπάρχουσα έκδοση). Στην περίπτωση απουσίας ρύθμισης η προεπιλογή είναι 1 - Παίκτης Ενισχυτικής Μάθησης.

-dw, -db: Επιλογή του βάθους αναζήτησης του Minimax για τον λευκό (**-dw**) και μαύρο παίκτη(**-db**). Στην περίπτωση απουσίας ρύθμισης η προεπιλογή είναι 3.

-hw, -hb: Μέγεθος κρυφού επιπέδου Νευρωνικού Δικτύου για τον λευκό (**-hw**) και μαύρο παίκτη(**-hb**). Στην περίπτωση απουσίας ρύθμισης η προεπιλογή είναι αυτή που περιγράφεται στο κεφάλαιο 3.

-lm: Επιλογή τύπου αρχείων στατιστικών: 1 - Στατιστικά στη μορφή των αρχείων των παλιών υλοποιήσεων του RLGame (προεπιλογή), 2 - Στατιστικά στη μορφή των αρχείων με περισσότερες πληροφορίες από αυτά των παλιών υλοποιήσεων, 3 - Και οι δύο τύποι αρχείων.

-dm: Επιλογή είδους σχόλιων : 0 – Κανένα (προεπιλογή), 1 - Σχόλια ελέγχου σφαλμάτων

Πρόγραμμα Connect4

Για το Connect4 έχει παραδοθεί η έκδοση του παιχνιδιού 6x7 (6 γραμμών και 7 κολώνων). Το εκτελέσιμο σε μορφή jar είναι το : RLConnect4.jar. Το jar του παιχνιδιού πρέπει να βρίσκεται στον ίδιο φάκελο με το jar του μηχανισμού μάθησης (RLLib.jar) κατά την εκτέλεση του πειράματος. Όλα τα αρχεία και οι φάκελοι που δημιουργούνται κατά την εκτέλεση του προγράμματος βρίσκονται κάτω από τον φάκελο εκτέλεσης. Στην ακόλουθη λίστα δίνονται 2 εναλλακτικές κλήσεις του προγράμματος και κατόπιν παρατίθενται οι διαθέσιμες ρυθμίσεις του Connect4.

- Παιχνίδι μεταξύ παικτών ενισχυτικής μάθησης για πέντε γύρους των 100 παιχνιδιών

    - java -jar RLConnect4.jar -r 5 -g 100

- Παιχνίδι μεταξύ λευκού minimax με βάθος αναζήτησης 3 (προεπιλεγμένη τιμή) και μαύρου παίκτη ενισχυτικής μάθησης για 10 γύρους των 50 παιχνιδιών

    - java -jar RLConnect4.jar -r 10 -g 50 -pw 2

Ρυθμίσεις

-h : Επιλογή Βοήθειας
-r : Επιλογή αριθμού γύρων
-g : Επιλογή αριθμού παιχνιδιών που θα εκτελεστούν ανά γύρο
-wg, -bg: τιμή για την παράμετρο του ρυθμού μείωσης γ (discount rate parameter) του λευκού(**-wg**), μαύρου (**-bg**) παίκτη
-wl, -bl: τιμή για τον παράγοντα παράληψης (forgetting factor) λ του λευκού (**-wl**), μαύρου (**-bl**) παίκτη
-wr, -br: Η τιμή που της ανταμοιβής του περιβάλλοντος σε περίπτωση νίκης (σε περίπτωση ήττας αποδίδεται η αντίστοιχη αρνητική τιμή) του λευκού (**-wr**), μαύρου (**-br**) παίκτη
-we, -be: Η τιμή του ποσοστού εκμετάλλευσης γνώσης που θα χρησιμοποιηθεί από τον λευκό (**-we**), μαύρο (**-be**) παίκτη στην επιλογή κίνησης (ε-greedy policy).
-ft: Επιλογή του παίκτη που θα κάνει την πρώτη κίνηση, 1 - λευκός, 2 – μαύρος. Στην απουσία ρύθμισης πρώτος ξεκινάει ο λευκός.
-pw, -pb: Επιλογή τύπου παίκτη για τον λευκό (**-pw**) και μαύρο παίκτη(**-pb**) 1 – Παίκτης Ενισχυτικής Μάθησης, 2 – Αλγόριθμος Minimax, 3 – Τυχαίας επιλογής, 4 – Άνθρωπος (δεν

υποστηρίζεται στην υπάρχουσα έκδοση). Στην περίπτωση απουσίας ρύθμισης η προεπιλογή είναι 1 - Παίκτης Ενισχυτικής Μάθησης.

-dw, -db: Επιλογή του βάθους αναζήτησης του Minimax για τον λευκό (**-dw**) και μαύρο παίκτη(**-db**). Στην περίπτωση απουσίας ρύθμισης η προεπιλογή είναι 3.

-hw, -hb: Μέγεθος κρυφού επιπέδου Νευρωνικού Δικτύου για τον λευκό (**-hw**) και μαύρο παίκτη(**-hb**). Στην περίπτωση απουσίας ρύθμισης η προεπιλογή είναι αυτή που περιγράφεται στο κεφάλαιο 7.

Β.4 Ευρετήριο Εκδόσεων

Στη παρούσα παράγραφό παραθέτεται το ευρετήριο των εκδόσεων των προγραμμάτων που έχουν παραδοθεί στα πλαίσια της διπλωματικής εργασίας καθώς γίνονται αναφορές σε αυτές τις εκδόσεις στο κείμενο της διπλωματικής διατριβής.

RLGame

- **Έκδοση 1.1 (alpha) – Ιανουάριος 2012 :** Εκτεταμένη αναδιάταξη του προγράμματος υπολογιστή εναντίον υπολογιστή
- **Έκδοση 1.2 – Ιούλιος 2012 :** Ενοποιημένη έκδοση παίκτη ενισχυτικής και minimax
- **Έκδοση 1.3 – Αύγουστος 2012 :** Διορθώσεις σφαλμάτων
- **Έκδοση 1.4 Σεπτέμβριος 2012 :** προσθήκη τυχαίας επιλογής κίνησης ανάμεσα στις κινήσεις που έχουν την μεγαλύτερη αξιολόγηση

Connect4

Έκδοση 1.0 – Μάρτιος 2012 : Πρώτη έκδοση που χρησιμοποιεί τον μηχανισμό μάθησης

Έκδοση 1.2 – Μάιος 2012 : Ενοποιημένη έκδοση παίκτη ενισχυτικής και minimax

Έκδοση 1.2 – Ιούνιος 2012 : Ο minimax χρησιμοποιεί το e-greedy policy

Έκδοση 1.3 – Αύγουστος 2012 : Διορθώσεις σφαλμάτων

Έκδοση 1.4 Σεπτέμβριος 2012 : προσθήκη τυχαίας επιλογής κίνησης ανάμεσα στις κινήσεις που έχουν την μεγαλύτερη αξιολόγηση