Ανοικτό Πανεπιστήμιο Κύπρου

Σχολή Θετικών και Εφαρμοσμένων Επιστημών

Μεταπτυχιακή Διατριβή
στα Πληροφοριακά και Επικοινωνιακά Συστήματα

Ανάπτυξη Διαδικτυακού Εξυπηρετητή για Παιχνίδια Στρατηγικής με Σύγχρονες Τεχνολογίες

Νίκος Δίκαρος

Επιβλέπων Καθηγητής
Δημήτρης Καλλές

Ιούνιος 2014

Ανοικτό Πανεπιστήμιο Κύπρου

Σχολή Θετικών και Εφαρμοσμένων Επιστημών

Ανάπτυξη Διαδικτυακού Εξυπηρετητή για Παιχνίδια Στρατηγικής με Σύγχρονες Τεχνολογίες

Νίκος Δίκαρος

Επιβλέπων Καθηγητής
Δημήτρης 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

Ιούνιος 2014

Περίληψη

Γίνεται χρήση σύγχρονων τεχνολογιών δικτυακού προγραμματισμού για την ανάπτυξη δικτυακής εφαρμογής στην οποία, μπορούν οι χρήστες από τον φυλλομετρητή (browser) τους να παίζουν ένα παιχνίδι στρατηγικής που παίζεται σε σκακιέρα με πούλια.

Υλοποιείται υποδομή για παιχνίδι ανθρώπου εναντίον ανθρώπου με web τεχνολογίες βασισμένες στη γλώσσα Java. Επίσης υλοποιείται κατανεμημένη αρχιτεκτονική, βάσει της οποίας, θα μπορούν ερευνητές του χώρου της τεχνητής νοημοσύνης να ενσωματώσουν προγράμματα (bots) που παίζουν αυτό το παιχνίδι, και τα οποία έχουν αναπτυχθεί με τεχνολογίες της προτιμήσεώς τους, στη λειτουργία της εφαρμογής. Η συνεργασία μεταξύ των bots και της εφαρμογής, θα γίνεται με τη χρήση HTTP αιτημάτων και μιας κοινής γλώσσας επικοινωνίας.

Επιλέγεται το βαθμολογικό σύστημα Glicko για την αξιολόγηση των επιδόσεων των παικτών (ανθρώπων και bots) και ενσωματώνεται στη διεξαγωγή των παιχνιδιών. Τέλος, προτείνεται ο αλγόριθμος Negamax με χρήση alpha-beta pruning και πινάκων μεταφοράς (transposition tables), σε συνδυασμό με μία ευριστική συνάρτηση εκτίμησης (evaluation function) για την κατασκευή ενός bot ,κάνοντας χρήση των δομών δεδομένων και των μεθόδων που δημιουργήθηκαν κατά την υλοποίηση της δικτυακής εφαρμογής.

Λέξεις κλειδιά: web-frameworks, παιχνίδια στρατηγικής,bots, συστήματα βαθμολογίας, Java-EE, Primefaces,Tomcat, MySQL,Hibernate

Summary

We use modern web programming technologies to develop a web application that enables users to play a strategy game played on a chessboard with checkers, via a webbrowser.

We Implemented an infrastructure for human vs human game with web technologies based in the Java language. We also implemented a distributed infrastructure with the use of which, artificial intelligence researchers can integrate programs (bots) playing this game (developed with technologies of their preference), within the server, using HTTP requests and a common communication language.

We integrated the Glicko rating system for assessing the performance of players (humans and bots) and proposed the NegaMax algorithm with alpha-beta pruning and transposition tables in conjunction with a heuristic evaluation function, in order to build a bot, using the data structures and methods we created during the server implementation.

Keywords: web-frameworks, strategy games, rating systems, Java-EE, Primefaces, bots, Tomcat, MySQL, Hibernate.

Ευχαριστίες

Στο σημείο αυτό αισθάνομαι την ανάγκη να ευχαριστήσω τον επιβλέποντα καθηγητή μου στην παρούσα διατριβή, Δημήτρη Καλλέ, για την αμέριστη συμπαράσταση και την ευελιξία που μου παρείχε στην εκπόνηση του θέματος.

Περιεχόμενα

Κεφάλαιο 1
Εισαγωγή

1.1 Αντικείμενο της διατριβής

Η παρούσα διατριβή αποσκοπεί στην δημιουργία ενός νέου web server όπου άνθρωποι και προγράμματα υπολογιστή (bots) θα μπορούν να παίζουν ένα παιχνίδι στρατηγικής που λέγεται RLGame.Το παιχνίδι παίζεται σε ένα ταμπλό παρόμοιο με σκακιέρα από δύο παίκτες που έχουν σκοπό να προωθήσουν τα πούλια τους στην βάση του αντιπάλου ώστε να κερδίσουν.

Προηγούμενες διατριβές έχουν ασχοληθεί με το παιχνίδι αυτό. Σε κάποιες απ' αυτές έγιναν προσπάθειες για τη δημιουργία ενός προγράμματος που έχει ανεπτυγμένη στρατηγική στο παιχνίδι, στα πλαίσια της τεχνητής νοημοσύνης, με χρήση μεθόδων μηχανικής μάθησης, νευρωνικών δικτύων και ενισχυτικής διδασκαλίας. Έχουν επίσης αναπτυχθεί, μία εφαρμογή γραφείου (desktop application) και μία δικτυακή εφαρμογή (web application) προκειμένου να μπορεί να υλοποιηθεί η διεξαγωγή του παιχνιδιού είτε μεταξύ ανθρώπου και υπολογιστή, είτε μεταξύ ανθρώπων.

Η παρούσα διατριβή, επεκτείνει τις προηγούμενες προσπάθειες για τη δημιουργία μίας διαδικτυακής εφαρμογής με σκοπό την διεξαγωγή παιχνιδιών RLGame είτε μεταξύ ανθρώπου

και υπολογιστή, είτε μεταξύ ανθρώπων. Σε συνέχεια των προηγούμενων προσπαθειών, γίνεται χρήση σύγχρονων τεχνολογιών δικτυακού προγραμματισμού, προκειμένου ο εξυπηρετητής, να παρέχει αφενός περισσότερες δυνατότητες και αφετέρου να μπορεί να είναι πιο επεκτάσιμος μελλοντικά, με λειτουργίες που θα υλοποιήσουν είτε ερευνητές του χώρου της τεχνητής νοημοσύνης, είτε web developers. Επίσης γίνεται μία αποτίμηση των χρησιμοποιούμενων τεχνολογιών και αξιολόγηση της καταλληλότητας τους για μελλοντική χρήση σε δικτυακές εφαρμογές.

Επειδή το RLGame είναι σχετικά καινούριο παιχνίδι, έχει ελάχιστα παιχτεί από ανθρώπους. Έτσι, η υπό κατασκευή εφαρμογή φιλοδοξεί να κεντρίσει το ενδιαφέρον του κοινού και να δημιουργηθεί μία βάση δεδομένων με παιχνίδια, τα οποία μπορούν να αποτελέσουν εξαιρετικό βοήθημα στην έρευνα για τη δημιουργία ενός στρατηγικά άρτιου προγράμματος (bot) το οποίο θα μπορεί να ανταγωνίζεται τους κορυφαίους παίκτες.

Παράλληλα, οι παίκτες παίζοντας περισσότερο θα έχουν την ευκαιρία να αναπτύξουν και τις στρατηγικές του παιχνιδιού. Για να δοθεί ένα επιπλέον κίνητρο σε αυτούς ώστε να δαπανήσουν «φαιά ουσία», υλοποιήθηκε στα πλαίσια της εργασίας και ένα σύστημα βαθμολογίας (rating) παρόμοιο με αυτά που χρησιμοποιούνται στη βαθμολόγηση παικτών σε παιχνίδια στρατηγικής όπως σκάκι και το τάβλι, που ονομάζεται Glicko rating.

Τέλος υλοποιείται μία κατανεμημένη αρχιτεκτονική, βάσει της οποίας, προγράμματα (bots) που θα δημιουργούνται από επίδοξους ερευνητές και τα οποία θα είναι κατασκευασμένα με τις τεχνολογίες της προτίμησής τους, θα μπορούν να συμμετέχουν στην λειτουργία του εξυπηρετητή μέσω HTTP requests από οποιαδήποτε διαδικτυακή τοποθεσία, και να ανταγωνίζονται με τους ανθρώπινους αντιπάλους, για την απόκτηση της καλύτερης δυνατής βαθμολογίας.

1.2 Δομή της διατριβής

Η διατριβή, πέραν της παρούσας εισαγωγής χωρίζεται σε πέντε ακόμη κεφάλαια.

Το 2ο κεφάλαιο ξεκινάει με την περιγραφή του παιχνιδιού και των κανόνων του. Στη συνέχεια παρουσιάζονται οι λειτουργίες που προσφέρει ο rlgameserver για την υλοποίηση του παιχνιδιού εναντίον ανθρώπου και η επέκταση για την λειτουργία στο παιχνίδι εναντίον υπολογιστή.

Παρουσιάζονται επίσης οι προσπάθειες που έχουν γίνει για να αναπτυχθεί ένας εξυπηρετητής και οιμέθοδοι τεχνητής νοημοσύνης που χρησιμοποιήθηκαν για να «εκπαιδευτεί» ο υπολογιστής στο παιχνίδι.

Στο 3ο κεφάλαιο γίνεται μία παρουσίαση των συστημάτων βαθμολόγησης (rating) παικτών στα παιχνίδια στρατηγικής και του συγκεκριμένου συστήματος που υλοποιήθηκε στα πλαίσια του rlgameserver.

Στο 4ο κεφάλαιο παρουσιάζονται οι τεχνολογίες υλοποίησης του rlgameserver. Γίνεται μία σύντομη παρουσίαση των πιο καθιερωμένων και γνωστών απ' αυτές που αποτελούν και τη βάση, και μία πιο εκτενής παρουσίαση των νεώτερων και λιγότερο γνωστών.

Στο 5ο κεφάλαιο παρουσιάζονται οι λεπτομέρειες και οι μεθοδολογίες που χρησιμοποιήθηκαν για την υλοποίηση των βασικών υπηρεσιών του rlgameserver με τη χρήση των προαναφερθέντων τεχνολογιών καθώς και οι μελλοντικές επεκτάσεις που κρίνονται σκόπιμες.

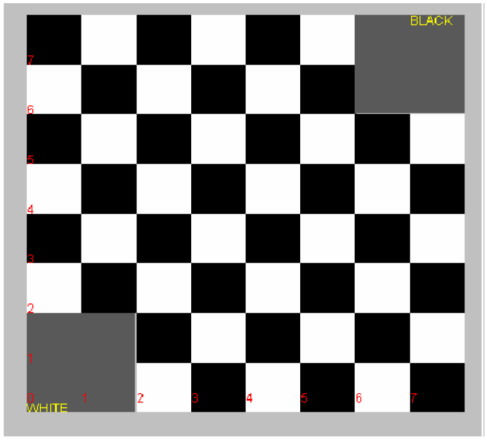
Στο 6Ο κεφάλαιο αναφέρονται τα συμπεράσματα που προέκυψαν κατά τη διαδικασία εκπόνησης της διατριβής.

Κεφάλαιο 2
Παρουσίαση των κανόνων του παιχνιδιού και των λειτουργιών του rlgameserver

Στο κεφάλαιο αυτό παρουσιάζονται οι κανόνες του παιχνιδιού και οι προσφερόμενες λειτουργίες του server.

2.1 Περιγραφή παιχνιδιού - κανόνες

Το παιχνίδι διεξάγεται σε μια τετραγωνική σκακιέρα διαστάσεων n x n και παίζεται μόνο από δύο παίκτες. Η βάση του πρώτου παίκτη στη σκακιέρα καταλαμβάνει την κάτω αριστερά γωνία της, και είναι μια τετραγωνική βάση διαστάσεων α x α όπου 2a+1<n. Ο άλλος παίκτης έχει τη βάση του έτσι ώστε να καταλαμβάνει στη σκακιέρα την απέναντι γωνία (πάνω δεξιά). Κατά σύμβαση ο πρώτος παίκτης ορίζεται ως "λευκός παίκτης" και ο δεύτερος ως "μαύρος παίκτης". Κάθε παίκτης φιλοξενεί στη βάση του από 1 έως β πούλια. Στην υλοποίηση του rlgameserver οι μεταβλητές έχουν τις εξής τιμές: n = 8, α = 2 και β = 10.



Εικόνα 2.1: Το ταμπλό του RLGame

Στόχος κάθε παίκτη είναι να βάλει ένα από τα ενεργά πούλια του στην αντίπαλη βάση προστατεύοντας συγχρόνως την δική του από τα αντίπαλα πούλια. Ο παίκτης που θα βάλει πρώτος κάποιο πούλι του στην αντίπαλη βάση θεωρείται νικητής. Επειδή οι κανόνες του παιχνιδιού προβλέπουν την απαλοιφή πουλίων όταν δεν έχουν νόμιμη κίνηση, αν κάποιος παίκτης απολέσει όλα του τα πούλια, ο αντίπαλος του θεωρείται αυτόματα νικητής.

Οι κινήσεις των πουλίων μπορούν να κατηγοριοποιηθούν ως εξής:

• Οι κινήσεις εξόδου από τη βάση: Η κάθε βάση θεωρείται ως ένα τετράγωνο και όχι ως μια συλλογή τετραγώνων, συνεπώς οποιοδήποτε πούλι της βάσης μπορεί να μετακινηθεί, με μια κίνηση, σε οποιοδήποτε από τα παρακείμενα στη βάση τετράγωνα που είναι ελεύθερα.

• Οι κινήσεις μετακίνησης από μία θέση σε μία άλλη: Κάθε πούλι μπορεί να μετακινηθεί σε οποιοδήποτε (πάνω, κάτω, δεξιά, αριστερά) γειτονικό ελεύθερο τετραγωνάκι της σκακιέρας, με την προϋπόθεση ότι η μέγιστη διαφορά του (είτε προς το x είτε προς y) από τη βάση του δεν μειώνεται (δεν επιτρέπονται δηλαδή κινήσεις προς τα πίσω).

Ένα πούλι που βρίσκεται σε ένα τετράγωνο με συντεταγμένες (x1, y1) και είναι πούλι "λευκού" παίκτη μπορεί να κινηθεί σε ένα μη κατειλημμένο τετράγωνο (x1, y2) ή (x2, y1)αν και μόνο αν

max (x1-α, y1-α) ≤ max (x1-α, y2-α) ή

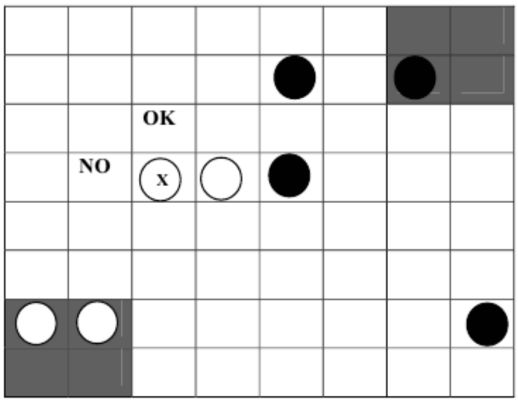
max (x2-α, y1-α) ≤ max (x2-α, y1-α)

Όμοια και για ένα πούλι "μαύρου" παίκτη που βρίσκεται σε τετράγωνο με συντεταγμένες(x1, y1) μπορεί να κινηθεί σε ένα μη κατειλημμένο τετράγωνο (x1, y2) ή (x2, y1) αν και μόνο αν

max (n-α-x1, n-α-y1) ≤ max (n-α-x1, n-α-y2) ή

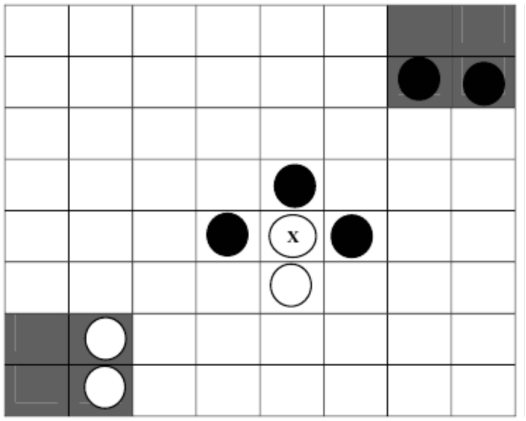
max (n-α-x1, n-α-y1) ≤ max (n-α-x2, n-α-y1)

Βάσει των κανόνων μετακίνησης, στην Εικόνα 2.2. Το λευκό πούλι Χ δεν μπορεί να μετακινηθεί στην θέση ΝΟ διότι παραβιάζεται ο κανόνας της αυξανόμενης μέγιστης διαφοράς του πιονιού από την βάση του [17].



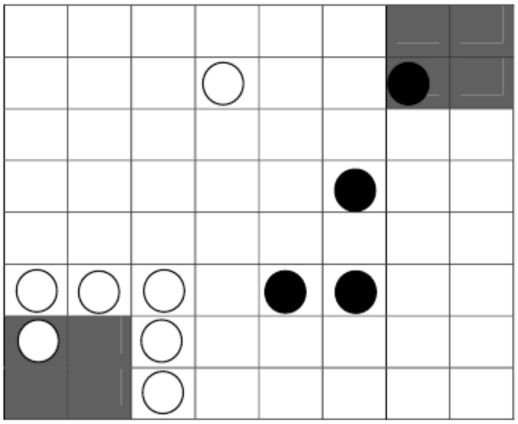
Εικόνα 2.2: Έγκυρες και μη έγκυρες κινήσεις γι το λευκό πούλι Χ

Το παιχνίδι επίσης προβλέπει και διαγραφές πουλίων. Όταν για ένα πιόνι όλες οι δυνατές επιτρεπτές θέσεις στα τετράγωνα της σκακιέρας, για την επόμενη κίνηση, είναι κατειλημμένες είτε από φίλια είτε από αντίπαλα πιόνια, το πιόνι αυτό δεν έχει νόμιμη κίνηση και απαλείφεται (Εικόνα 2.3).



Εικόνα 2.3: Απώλεια πουλίου για το οποίο δεν υπάρχει επιτρεπτή κίνηση

Συνέπεια αυτού είναι το ότι, όταν μία βάση "περικυκλωθεί" από πούλια (είτε φίλια είτε αντίπαλα) τότε όσα πιόνια φιλοξενεί εντός της απαλείφονται. Στην Εικόνα 2.4 για παράδειγμα, η βάση του "λευκού" παίκτη είναι περικυκλωμένη από πιόνια του ιδίου συνεπώς αυτομάτως όσα από τα υπόλοιπα πιόνια φιλοξενούνται στη βάση του θα απαλειφθούν.



Εικόνα 2.4: Απώλεια όλων των πουλίων της λευκής βάσης

2.2 Προσπάθειες για τη δημιουργία ενός online server για το RLGame

Στη διπλωματική του ο Γιάννος Δήμας [17] δημιούργησε ένα δικτυακό εξυπηρετητή, όπου οι συνδεδεμένοι χρήστες μπορούν να παίζουν μεταξύ τους, καθώς και εναντίον του υπολογιστή.

Η υλοποίηση του server έγινε με τεχνολογίες βασισμένες στη συναρτησιακή γλώσσα προγραμματισμού SCALA και στο πιο γνωστό web frameworkπου τη συνοδεύει, το LIFT. Η υλοποίηση του παιχνιδιού ανθρώπου εναντίον υπολογιστή γίνεται με την τεχνολογία των applets και χρησιμοποιείται το παραδοτέο της διπλωματικής εργασίας [16] το οποίο ήταν ένα java applet που λειτουργούσε εκτός web browser, με κάποιες τροποποιήσεις προκειμένου να εντάσσεται ομαλά στο περιβάλλον του browser.

Σαν συνέχεια της προσπάθειας αυτής, στην παρούσα διπλωματική γίνεται μία ανακατασκευή του δικτυακού αυτού server με δύο κύρια ζητούμενα. Το πρώτο ζητούμενο είναι να δοκιμαστούν νέες τεχνολογίες που εμφανίστηκαν τα τελευταία χρόνια στο χώρο της ανάπτυξης δικτυακών εφαρμογών, και να εξεταστεί το κατά πόσο προσφέρονται για τη δημιουργία ενός online εξυπηρετητή παιχνιδιών. Το δεύτερο ζητούμενο είναι η τροποποίηση της προηγούμενης υλοποίησης με τέτοιο τρόπο ώστε να διευκολύνεται η μελλοντική επέκταση του server με νέες δυνατότητες και η ενσωμάτωση bots που θα έχουν κατασκευαστεί από ανεξάρτητους ερευνητές, και τα οποία θα ανταγωνίζονται τους ανθρώπινους παίκτες, έχοντας τη δική τους βαθμολογία. Για το σκοπό αυτό, η λογική του παιχνιδιού υλοποιείται αποκλειστικά πλέον στην πλευρά του server όπου κρατούνται όλα τα δεδομένα των παιχνιδιών που είναι σε εξέλιξη ανά πάσα στιγμή, ενώ ο browser χρησιμεύει αποκλειστικά και μόνο σαν ένα interface μεταξύ παικτών, ή μεταξύ παικτών και bots.

2.3 Χρήση τεχνητής νοημοσύνης για την κατασκευή ενός ευφυούς πράκτορα (bot)

Έχουν γίνει προσπάθειες σε προηγούμενες διατριβές, για την εκπαίδευση ευφυών πρακτόρων (intelligent agents)στις στρατηγικές του παιχνιδιού. Νευρωνικά δίκτυα χρησιμοποιήθηκαν σε συνδυασμό με ενισχυτική μάθηση [10]. Αντί για την εκπαίδευση των νευρωνικών δικτύων σε παιχνίδι εναντίον του εαυτού τους, παρεμβλήθηκαν παιχνίδια εναντίον ανθρώπων, τα οποία επιταχύνουν την εκπαιδευτική διαδικασία. Σε μεταγενέστερη προσπάθεια, χρησιμοποιήθηκε ένας εκπαιδευτής (tutor) που χρησιμοποιεί τον αλγόριθμο minimax [12].

Στα πλαίσια της παρούσας εργασίας, φιλοδοξία είναι να δημιουργηθεί μία πλατφόρμα, πάνω στην οποία θα στηριχθούν μελλοντικοί ερευνητές για να κατασκευάσουν τους δικούς τους πράκτορες και επίσης να αποθηκευθεί μία αξιόλογη βάση δεδομένων με παιχνίδια (άνθρωπος

εναντίον ανθρώπου ή άνθρωπος εναντίον υπολογιστή) τα οποία θα μπορούν αργότερα να τεθούν και αυτά στην υπηρεσία της έρευνας πάνω στο παιχνίδι.

2.4 Λειτουργίες του rlgameserver

2.4.1 Δημόσιο chat

Είναι καθιερωμένη η δυνατότητα στους δικτυακούς εξυπηρετητέςπαιχνιδιών να μπορούν οι παίκτες να επικοινωνούν γραπτώς μεταξύ τους με τη μορφή chat. Orlgameserverδεν αποτελεί εξαίρεση στον κανόνα αυτό. Συγκεκριμένα οι χρήστες μπορούν να επικοινωνούν σε ένα δημόσιο chat room όπου όλοι βλέπουν τι γράφεται, ενώ υπάρχει δυνατότητα ένας παίκτης να στείλει ιδιωτικό μήνυμα (private message) σε κάποιον άλλο παίκτη, το οποίο δεν είναι ορατό από τους υπόλοιπους.

2.4.2 Command line interface

Εκτός από τη δυνατότητα επιτέλεσης λειτουργιών μέσω γραφικής διεπαφής χρήστη (GUI), σε πολλούς clients δικτυακώνπαιχνιδιών δίνεται η δυνατότητα οι χρήστες να επιτελούν διάφορες λειτουργίες μέσω της γραμμής εντολών. Στον rlgameserver υλοποιείται επίσης αυτή η δυνατότητα, και οι χρήστες μπορούν να καλέσουν και να αποδεχτούν πρόσκληση για παιχνίδι μέσω εντολών στην κονσόλα, ή να δουν τις βαθμολογίες των παικτών που συμμετέχουν, ταξινομημένες σε φθίνουσα σειρά. Παρέχεται επίσης η υποδομή, ούτως ώστε σε μελλοντική επέκταση του serverνα μπορούν εύκολα με μερικές γραμμές κώδικα να προστίθενται νέες λειτουργίες μέσω εντολών κονσόλας.

2.4.3 Υλοποίηση παιχνιδιού άνθρωπος εναντίον ανθρώπου

Η βασική προσφερόμενη λειτουργία του rlgameserver είναι η δυνατότητα 2 χρηστών να παίξουν ο ένας εναντίον του άλλου, αφού πρώτα απευθύνουν μία πρόσκληση για παιχνίδι ο ένας στον άλλο. Ο παίκτης που απευθύνει την πρόσκληση παίρνει τα άσπρα πούλια ενώ αυτός που την αποδέχεται παίρνει τα μαύρα.

2.4.4 Υλοποίηση παιχνιδιού Άνθρωπος εναντίον Υπολογιστή

Έχει υλοποιηθεί σχετική υποδομή προκειμένου, να μπορούν να ενσωματωθούν στον rlgameserver προγράμματα που έχουν κατασκευαστεί από τρίτους και τα οποία θα μπορούν να ανταγωνίζονται στο παιχνίδι εναντίον ανθρωπίνων αντιπάλων. Γίνεται η προσπάθεια ώστε όλοι όσοι φιλοδοξούν να κατασκευάσουν έναν ευφυή πράκτορα (intelligent agent ή bot), ανεξαρτήτως της γλώσσας προγραμματισμού και των τεχνολογιών που θα χρησιμοποιήσουν, να μπορούν να το ενσωματώσουν σχετικά εύκολα στον rlgameserver μέσω διαδικτύου και ανταλλαγής http αιτημάτων. Με τον τρόπο αυτό, τα bots θα μπορούν να βρίσκονται οπουδήποτε στο διαδίκτυο.

2.4.5 Παρακολούθηση βαθμολογίας

Όλοι οι παίκτες στον rlgameserver από τη στιγμή της εγγραφής τους και μετά διαθέτουν προσωπική βαθμολογία τύπου Glicko rating system. Μετά από κάθε παιχνίδι η βαθμολογία αυτή ανανεώνεται βάσει σχετικής φόρμουλας.

Κεφάλαιο 3
Τεχνολογίες Υλοποίησης

Στο κεφάλαιο αυτό γίνεται μια παρουσίαση των προγραμματιστικών τεχνολογιών που χρησιμοποιήθηκαν. Η παρουσίαση γίνεται από τις πιο χαμηλού επιπέδου (low level) προς τις πιο υψηλού επιπέδου (high level) τεχνολογίες, στην πλευρά του πελάτη και του εξυπηρετητή αντίστοιχα.

3.1 Τεχνολογίες ανάπτυξης web εφαρμογών

Τα τελευταία χρόνια, το παραδοσιακό μοντέλο εφαρμογών που τρέχουν μέσω διαδικτύου και το οποίο συνίστατο στην αρχιτεκτονική client-server, έχει αντικατασταθεί σε πολύ μεγάλο βαθμό με τις web εφαρμογές, στις οποίες το ρόλο του πελάτη(client) κατέχει ο φυλομετρητής (web browser). Τα πλεονεκτήματα που προσφέρει αυτό το μοντέλο είναι αρκετά (π.χ. οι χρήστες δεν χρειάζεται πια να εγκαθιστούν διάφορα λογισμικά στον υπολογιστή τους, αλλά αρκεί να διαθέτουν έναν browser τελευταίας τεχνολογίας).

Συνέπεια αυτού ήταν να αναπτυχθούν ραγδαία και οι τεχνολογίες που επιτρέπουν στους προγραμματιστές να αναπτύσσουν δυναμικές και διαδραστικές εφαρμογές σε φυλλομετρητές.

Επειδή στην αρχική του μορφή οπαγκόσμιος ιστός περιοριζόταν στην ανταλλαγή μορφοποιημένου σε HTML κειμένου ανάμεσα στον serverκαι τον browser, η HTML υπήρξε για πολλά χρόνια μία γλώσσα μορφοποίησης(markup language) με περιορισμένες δυνατότητες και όχι μία κανονική γλώσσα προγραμματισμού. Οι αυξανόμενες ανάγκες οδήγησαν στην δημιουργία ενός συνόλου τεχνολογιών οι οποίες χρησιμοποιούνται σε συνδυασμό με τον βασικό κορμό μιας ιστοσελίδας (δηλαδή την HTML). Κάποιες απ' αυτές τις τεχνολογίες, δημιουργήθηκαν για να προσθέσουν δυνατότητες στην πλευρά του server, ενώ κάποιες άλλες χρησιμοποιούνται κυρίως στην πλευρά του client.

3.2 Client side τεχνολογίες JavaScript (JS).

Η JavaScript (JS) είναι μια δυναμική γλώσσα προγραμματισμού. Χρησιμοποιείται συνήθως σαν τμήμα των φυλλομετρητών (web browsers) στους οποίους επιτρέπει μέσω client side scripts την αλληλεπίδραση με τον χρήστη, τον έλεγχο του browser, την ασύγχρονη επικοινωνία και την αλλοίωση του περιεχομένου του εγγράφου που εμφανίζεται. Έχει επίσης κερδίσει αρκετό έδαφος στον χώρο του server side programming μέσω της βιβλιοθήκης Node.js με τη χρήση της οποίας μπορεί να γραφτεί κώδικας για server side εφαρμογές.

Η σύνταξή της έχει επηρεαστεί από την C και χρησιμοποιεί πολλά ονόματα και συμβάσεις ονοματολογίας από την Java, αλλά πέραν αυτού οι 2 γλώσσες δεν έχουν κάποια άλλη σχέση. Είναι πολυπαραδειγματική (multi-paradigm) γλώσσα, η οποία υποστηρίζει τεχνικές αντικειμενοστραφούς (object oriented), δομημένου (imperative) και συναρτησιακού (functional) προγραμματισμού.

Η Javascript ξεκίνησε σαν scripting γλώσσα προγραμματισμού για σελίδες διαδικτύου για απλές περιπτώσεις επαλήθευσης (validation) δεδομένων φορμών πριν την αποστολή στονserver. Αρχικά εμφανίστηκε μόνο στον Netscape Navigator. Από τότε εξελίχθηκε σε ένα σημαντικό χαρακτηριστικό κάθε μεγάλου φυλλομετρητή. Δεν περιορίζεται πλέον για εφαρμογές επαλήθευσης δεδομένων αλλά χειρίζεται πλέον κάθε πτυχή του παραθύρου του browser και των περιεχομένων του. Ακόμη και η Microsoft που αρχικά είχε λανσάρει την δική της αντίστοιχη γλώσσα την VBScript, κατέληξε να περιλαμβάνει την δική της έκδοση της Javascript στον

Internet Explorer από τις πρώτες εκδόσεις του. Σήμερα σε συνδυασμό με τις νέες τεχνολογίες όπως το Ajax καθώς και βιβλιοθήκες που προσθέτουν δυνατότητες στη γλώσσα όπως η jQuery, η Javascript χρησιμοποιείται σχεδόν αποκλειστικά όσον αφορά τον client-sideπρογραμματισμό [26].

3.2.1 jQuery

Η jQuery είναι μία JavaScript βιβλιοθήκη πολλαπλής πλατφόρμας (cross platform),σχεδιασμένη για να απλοποιήσει την χρήση της Javascript στο σκριπτάρισμα της HTML στην πλευρά του client. Στις μέρες μας χρησιμοποιείται ευρύτατα, και είναι χαρακτηριστικό πως τη χρησιμοποιούν το 80% των 10.000 sites με την μεγαλύτερη επισκεψιμότητα παγκοσμίως.

Η jQuery είναι ελεύθερο και ανοιχτού κώδικα λογισμικό και η σύνταξή του είναι σχεδιασμένη ώστε να διευκολύνει την πλοήγηση εντός ενός HTML εγγράφου, την επιλογή και χρήση των DOM (DOCUMENT OBJECT MODEL) στοιχείων της σελίδας και στη συνέχεια τον χειρισμό αυτών των στοιχείων με την προσθήκη περιεχομένου, τον ορισμό χειριστών συμβάντων (event handlers), τη δημιουργία animations και την διαχείριση γεγονότων (events) και την ανάπτυξη Ajax εφαρμογών [27].

Παρακάτω παραθέτουμε μερικά παραδείγματα χρήσης της jQuery και τον αντίστοιχο κώδικα σε Javascript ώστε να γίνουν εμφανή τα χαρακτηριστικά και οι διαφορές μεταξύ της απλής JavaScriptκαι της jQuery.

1. Απλή επιλογή στοιχείων HTML

Αν υποθέσουμε ότι στον κώδικα της σελίδας HTML έχουμε το ακόλουθο στοιχείο (element):

```
<div id=”example”>Hello world!</div>
```

Για να επιλέξουμε το στοιχείο αυτό μέσω jQuery γράφουμε:

```
var elem = $('#example');
```

Ο αντίστοιχος κώδικας σε Javascript:

```
var elem = document.getElementById('example');
```

Η σύνταξη της εντολής σε jQuery είναι πιο συνοπτική. Επιπλέον δεν υπολείπεται πολύ της Javascript από θέμα ταχύτητας καθώς η jQuery χρησιμοποιεί εσωτερικά την μηχανή επιλογής CSS Sizzle[1sizz] η οποία όπου είναι δυνατό χρησιμοποιεί τις λειτουργίες του πυρήνα της Javascript για μεγαλύτερη αποδοτικότητα [17].

2. Κίνηση (animation) στοιχείων HTML

Έστω ότι θέλουμε να μετακινήσουμε το παραπάνω στοιχείο δεξιά κατά 100 pixels. Μέσω jQuery:

```
$('#example').animate({right:'100px'});
```

Ο αντίστοιχος κώδικας σε Javascriptείναι ο εξής:

```
var example = null; // object

function doMove() {

example.style.left = parseInt(example.style.right)+1+'px';

setTimeout(doMove,20); // call doMove in 20msec

}

function init() {

example = document.getElementById('example');

example.style.right = '0px'; // αρχική θέση -> 0px

doMove(); // startanimating

}

init();
```

Είναι δηλαδή εμφανές ότι η χρήση της jQuery προκαλεί μεγάλη διευκόλυνση στη συγγραφή και διαχείριση του client side κώδικα.

3.2.1 Ajax

Ajax είναι το ακρωνύμιο του όρου Asynchronous Javascript and XML και συνίσταται σε ένα γκρουπ συνεργαζόμενων μεθόδων που χρησιμοποιούνται στον browser για την δημιουργία διαδραστικών εφαρμογών διαδικτύου. Μέσω του Ajax οι διαδικτυακές εφαρμογές, μπορούν να αποστείλουν δεδομένα προς και να λάβουν σαν απάντηση δεδομένα από τον web server με ασύγχρονο τρόπο (στο υπόβαθρο) χωρίς να διαταράσσουν την εμφάνιση και την συμπεριφορά της σελίδας που εμφανίζεται στην οθόνη. Τα δεδομένα συνήθως λαμβάνονται με την χρήση του αντικειμένου XMLHTTPRequest της Javascript. Παρά την ονομασία η χρήση της XML δεν είναι απαραίτητη (συχνά χρησιμοποιείται JSON πρωτόκολλο ανταλλαγής δεδομένων) και οι αιτήσεις δεν είναι απαραίτητο να είναι ασύγχρονες [19].

Η Ajax δεν είναι μία τεχνολογία αλλά ένα σύνολο από τεχνολογίες που επιτυγχάνουν τον τελικό σκοπό. Χρησιμοποιεί ένα συνδυασμό HTML και CSS για να μορφοποιήσουν και να αλλάξει την εμφάνιση στοιχείων στην σελίδα. Η πρόσβαση στο DOM(Document Object Model) γίνεται μέσω Javascript για την δυναμική εμφάνιση και για να επιτρέψει στον χρήστη να αλληλεπιδράσει με την πληροφορία που εμφανίζεται. Η Javascript μέσω του XMLHTTPRequest object παρέχει την μέθοδο για ανταλλαγή δεδομένων ασύγχρονα μεταξύ browser και server για την αποφυγή επαναφόρτωσης της σελίδας HTML.

Ένα παράδειγμα χρήσης Ajax παρουσιάζεται ακολούθως με τη χρήση της μεθόδου GET:

```
// Το script από την πλευρά του client

// Αρχικοποίηση ενός αιτήματοςAjax

varxhr = newXMLHttpRequest();

xhr.open('get','send-ajax-data.php');
```

```
// Παρακολούθηση των αλλαγών στην κατάσταση του αιτήματος

xhr.onreadystatechange=function(){

      // Ready state 4 means the request is done

      if(xhr.readyState===4){

      // 200 is a successful return

      if(xhr.status===200){

      alert(xhr.responseText);// 'Εδώ είναι το κείμενο που επιστρέφεται.'

}else{

alert('Error: '+xhr.status);// Μήνυμα σε περίπτωση λάθους

}

}

}

// Αποστολή του αιτήματος στο server side script send-ajax-data.php

xhr.send(null);
```

send-ajax-data.php:

```
<?php

// Toscript που ‘ακούει’ στην πλευρά του server

// Ορισμός του τύπου περιεχομένου

header('Content-Type: text/plain');
```

```
// Αποστολή των δεδοομένων πίσω στον client

echo"This is the returned text.";

?>
```

Όπως βλέπουμε η αρχιτεκτονική λειτουργίας του Ajax είναι απλή. Δημιουργείται ένα αίτημα GET από την πλευρά του client, το οποίο στέλνεται ασύγχρονα, χωρίς υποβολή φόρμας προς το script send-ajax-data.php που 'περιμένει' στην πλευρά του server και το οποίο με τη σειρά του επιστρέφει ένα μήνυμα με μορφή κειμένου προς τον client, το οποίο εμφανίζεται σε ένα javascript alert σε περίπτωση που όλα πήγαν όπως έπρεπε και δεν μεσολάβησε κάποιο σφάλμα , το οποίο εντοπίζεται από τους μηχανισμούς του Ajax.

Η χρήση του Ajax απλοποιείται σημαντικά με τη χρήση της jQueryτο οποίο φαίνεται στον ακόλουθο κώδικα ο οποίος επιτελεί την ίδια εργασία με το client side script που δείξαμε παραπάνω [17].

```
$.get('send-ajax-data.php', function(data){

alert(data);

});
```

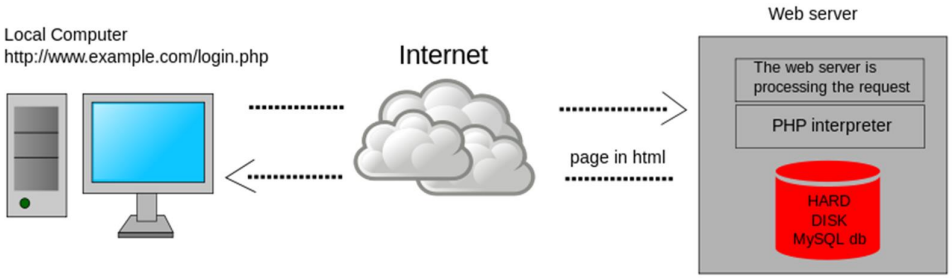
3.3 Server side τεχνολογίες

Το Server-side scripting είναι μια τεχνική που χρησιμοποιείται στον σχεδιασμό ιστοσελίδων και περιλαμβάνει την ενσωμάτωση scripts σε HTML κώδικα που οδηγεί στη δημιουργία ενός αιτήματος από την πλευρά του πελάτη προς τον εξυπηρετητή, όπου και ένα script διαχειρίζεται το αίτημα, προτού ο server απαντήσει στο αίτημα του client. Τα scripts αυτά, μπορούν να γραφτούν σε οποιαδήποτε από μια σειρά από διαθέσιμες server-side scripting γλώσσες. Το server-side scripting διαφέρει από το client-side scripting, όπου ενσωματωμένα scripts, όπως η JavaScript τρέχουν στον φυλλομετρητή.

Το Server-side scripting, συνήθως χρησιμοποιείται για να παρέχει μια διεπαφή και να περιορίσει την πρόσβαση των clients σε ιδιόκτητες βάσεις δεδομένων ή άλλες πηγές δεδομένων. Αυτά τα scripts μπορούν να συγκεντρώσουν τα χαρακτηριστικά του client για να τα χρησιμοποιήσουν στη διαμόρφωση της απάντησης που θα στείλει ο server. Τέτοια χαρακτηριστικά είναι οι απαιτήσεις του χρήστη, τα δικαιώματα πρόσβασης κ.λ.π. Το Server-side scripting επιτρέπει επίσης στον ιδιοκτήτη της ιστοσελίδας να μειώσει την πρόσβαση των χρηστών στον πηγαίο κώδικα των server-side scripts που μπορεί να είναι ιδιόκτητα και να έχουν από μόνα τους κάποια αξία. Το μειονέκτημα της χρήσης του server-side scripting είναι ότι η ιστοσελίδα του υπολογιστή του server θα πρέπει να παρέχει πόρους του συστήματος πριν από την αποστολή μιας σελίδας στον υπολογιστή-πελάτη για την προβολή μέσω του web browser.

Όταν ο server σερβίρει δεδομένα με ένα κοινώς αποδεκτό τρόπο, για παράδειγμα, σύμφωνα με τα πρωτόκολλα HTTP ή FTP, οι χρήστες μπορούν να επιλέξουν από μια σειρά από client προγράμματα (οι σύγχρονοι web browsers μπορούν να ζητήσουν και να λάβουν δεδομένα χρησιμοποιώντας και τα δύο αυτά πρωτόκολλα). Στην περίπτωση των πιο εξειδικευμένων εφαρμογών, οι προγραμματιστές μπορούν να γράψουν το δικά τους πρωτόκολλα (server, client και επικοινωνίας) που μπορούν να χρησιμοποιηθούν μόνο το ένα με το άλλο [30].

Το μοντέλο του server side scripting εικονίζεται στην Εικόνα 3.1.



Εικόνα 3.1: Αρχιτεκτονική του server-side scripting.

Στις μέρες μας οι προγραμματιστές μπορούν να επιλέξουν ανάμεσα σε μία πλειάδα γλωσσών προγραμματισμού (PHP, Java, Ruby, Python κ.λ.π) για να πραγματοποιήσουν το server side scripting, οι οποίες πλέον συνοδεύονται από τις δικές τους εξειδικευμένες τεχνολογίες και frameworks για τη διευκόλυνση της ανάπτυξης. Για της ανάγκες της παρούσας εργασίας, επιλέχθηκε η γλώσσα Java(και το συνεπακόλουθο οικοσύστημα τεχνολογιών).

3.3.1 Java EE

Η πλατφόρμα JavaEE (EnterpriseEdition) της Oracle , προσφέρει ένα API και ένα runtime environment για την ανάπτυξη και λειτουργία εταιρικού (enterprise) λογισμικού, συμπεριλαμβανομένων δικτυακών εφαρμογών, εφαρμογών τύπου webservices και άλλων μεγάλης κλίμακας, κατανεμημένων, ασφαλών και αξιόπιστων εφαρμογών που λειτουργούν μέσω διαδικτύου. Η JavaEE επεκτείνει την JavaStandard Edition.

Οι κυριότερες τεχνολογίες βάσει των οποίων η JavaEE επεκτείνει τη λειτουργικότητα της JavaSE και οι οποίες χρησιμοποιήθηκαν για την ανάπτυξη του εξυπηρετητή της παρούσας εργασίας είναι οι εξής:

`Javax.servlet.*`

Η προδιαγραφή servlet ορίζειένα σύνολο από APIs για την εξυπηρέτηση κυρίωςτων αιτημάτων τύπου HTTP. Περιλαμβάνει επίσης την JavaServer Pages(JSP).

`Javax.faces.*`

Αυτό το πακέτο ορίζει τη ρίζα του JavaServerFaces(JSF) API. Το JSFείναι μια τεχνολογία για την κατασκευή web διεπαφών χρήστη. Οι διεπαφές κατασκευάζονται με τη χρήση προκαθορισμένων XML tags αντί για τα γνωστά HTML, τα οποία συνιστούν έτοιμα components. Στον developer παρέχεται η δυνατότητα να δημιουργεί τα δικά του components, έτοιμα για μελλοντική χρήση.

`Javax.el.*`

Αυτό το πακέτο ορίζει τις κλάσεις και διεπαφές για την Java Expression Language. Η Expression Language(EL),είναι μια απλή γλώσσα που αρχικά σχεδιάστηκε για να ικανοποιήσει τις ιδιαίτερες ανάγκες των προγραμματιστών που αναπτύσσουν διαδικτυακές εφαρμογές. Χρησιμοποιείται κυρίως στις σελίδες JSF προκειμένου να υπάρχει πρόσβαση σε μεταβλητές και μεθόδους των backing beans αλλά μπορεί να χρησιμοποιηθεί σε ολόκληρη την πλατφόρμα.

`Javax.persistence.*`

Αυτό το πακέτο περιέχει τις κλάσεις και διεπαφές οι οποίες καθορίζουν τις συμβάσεις μεταξύ του persistence provider, των διαχειριζόμενων κλάσεων, και τους clients του Java Persistence API(JPA) [25].

3.3.2 Java Servlets

Ένα servlet είναι μία κλάση Java που κληρονομεί (inherits) και επαυξάνει (extends) τις δυνατότητες των προγραμμάτων εξυπηρετητών για την δημιουργία εφαρμογών που λειτουργούν με το προγραμματιστικό μοντέλο αίτησης και απάντησης (request – response model). Αν και τα servlets μπορούν να απαντήσουν σε κάθε λογής αίτηση χρησιμοποιούνται κατά κύριο λόγο για την υλοποίηση εφαρμογών που φιλοξενούνται σε διαδικτυακούς εξυπηρετητές (web servers).

Το servlet είναι μία κλάση της Java που συμμορφώνεται προς το Java Servlet API, ένα πρωτόκολλο με το οποίο μία Java κλάση μπορεί να απαντήσει σε αιτήσεις. Δεν είναι περιορισμένο στην χρήση κάποιου συγκεκριμένου πρωτοκόλλου επικοινωνίας αλλά το σύνηθες είναι να χρησιμοποιούν το πρωτόκολλο HTTP και για αυτό ο όρος servlet έχει αποκτήσει το νόημα "HTTP servlet". Συνεπώς ένας μηχανικός λογισμικού μπορεί να χρησιμοποιήσει ένα servlet για προσθέσει δυναμικό περιεχόμενο σε ένα web server χρησιμοποιώντας την πλατφόρμα Java. Το δυναμικό περιεχόμενο είναι τυπικά HTML αλλά μπορεί να είναι και άλλων τύπων, όπως XML. Αντίστοιχες με τα servlets τεχνολογίες παραγωγής δυναμικού διαδικτυακού περιεχομένου σε άλλες πλατφόρμες είναι η ASP.NET, και το CGI.

Για να εγκατασταθούν και να εκτελεστούν τα servlets χρειάζονται ένα εξυπηρετητή που ονομάζεται Servlet ή Web Container. Το Servlet API καθορίζει πλήρως τις επιτρεπόμενες αλληλεπιδράσεις μεταξύ servlet και web container. Στην ουσία ο web container είναι το τμήμα του web server που αλληλεπιδρά με τον web server. Ο web container είναι υπεύθυνος για την διαχείριση του κύκλου ζωής των servlets, για την αντιστοίχηση ενός URL σε ένα συγκεκριμένο servlet για την εξασφάλιση των απαραίτητων δικαιωμάτων πρόσβασης από την πλευρά του αιτούντος (requester). Για τις ανάγκες του rlgameserver χρησιμοποιήθηκε ως servletcontainer ο Apache Tomcat.

Πιο συγκεκριμένα το servlet είναι ένα αντικείμενο που δέχεται μία αίτηση και παράγει μία απάντηση που βασίζεται σε αυτή την αίτηση. Το βασικό πακέτο της Java το javax.servlet.http καθορίζει αντικείμενα Java που αντιστοιχούν σε αιτήσεις και απαντήσεις καθώς και αντικείμενα

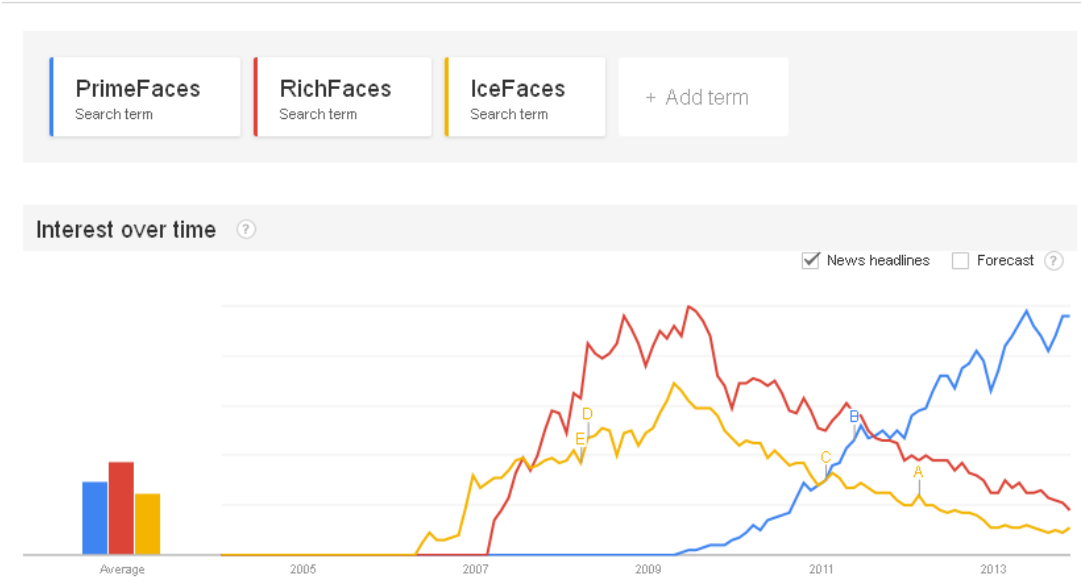
που αντικατοπτρίζουν την παραμετροποίηση και το περιβάλλον εκτέλεσης ενός servlet. Οι διαδικτυακές εφαρμογές τύπου servlet πακετάρονται σε *.war αρχεία πριν αναπτυχτούν σε ένα web container σαν διαδικτυακές εφαρμογές [17].

3.3.3 Primefaces

Για να διευκολυνθούν οι προγραμματιστές web εφαρμογών που χρησιμοποιούν JavaServerFaces έχουν δημιουργηθεί βιβλιοθήκες που παρέχουν έτοιμα components (widgets) τα οποία μπορούν να χρησιμοποιηθούν από τους προγραμματιστές σε συνδυασμό με τα βασικά componentsπου προσφέρει η βιβλιοθήκη JSF για τη δημιουργία διεπαφών χρήστη κατά την κατασκευή μίας web εφαρμογής.

Οι κυριότερες βιβλιοθήκες της αγοράς αυτή τη στιγμή για τις JSF σελίδες, θεωρούνται οι Primefaces, Richfaces, Icefaces.

Η βιβλιοθήκη Primefaces είναι η πιο δυναμικά εξελισσόμενη τα τελευταία χρόνια και αυτό καταμαρτυρεί και η πορεία των αναζητήσεων στην μηχανή Google έτσι όπως φαίνεται από την Εικόνα 3.2, και το χαρακτηριστικό αυτό σε συνδυασμό με τα άλλα πλεονεκτήματά της που αναλύονται ακολούθως, επιλέχθηκε για την υλοποίηση του rlgameserver στα πλαίσια της παρούσας εργασίας.



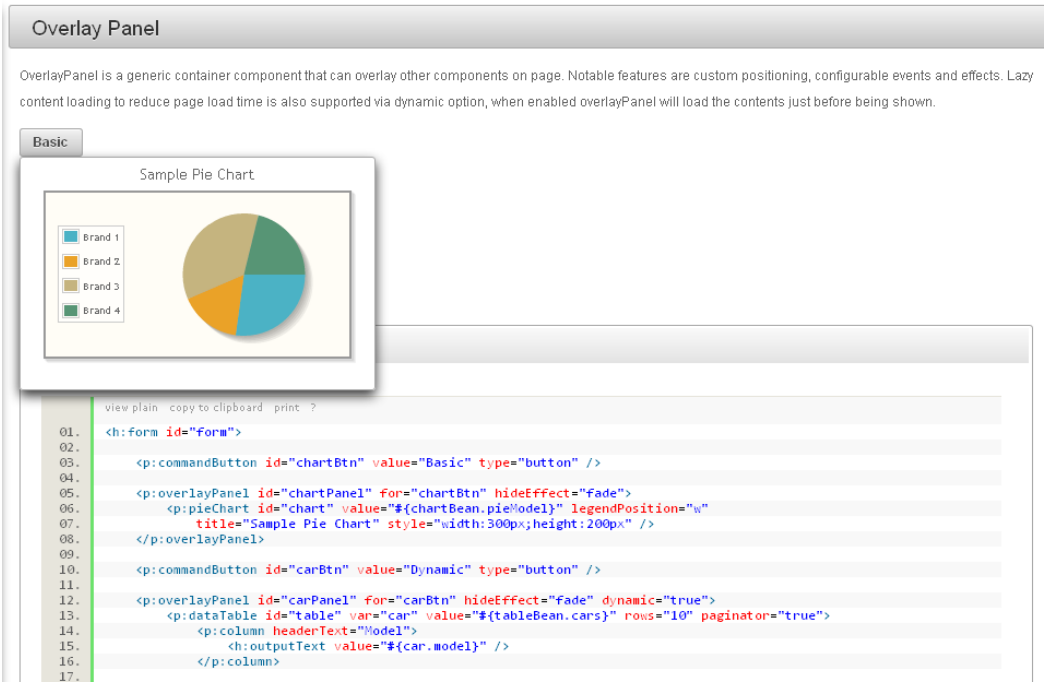
Εικόνα 3.2: Ο δείκτης αναζητήσεων στο google για τα τρία κυριότερα JSF component libraries. Πηγή: http://www.google.com/trends

Όπως φαίνεται από την εικόνα τα PrimeFaces τα τελευταία χρόνια, έχουν καταλάβει το μεγαλύτερο κομμάτι της αγοράς που μέχρι πρότινος κατείχαν οι 2 άλλες βιβλιοθήκες

Στον ιστοτόπο http://www.primefaces.org/showcase, μπορεί κανείς να δει τα εργαλεία που παρέχουν τα PrimeFaces. Προσφέρεται μια πλειάδα από components για τον εύκολο και γρήγορο σχεδιασμό των user interfaces (datatables, menus, charts, multimedia, draganddrop, googlemaps κ.λ.π), για την μερική ανανέωση των σελίδων μέσω AJAX. Καθώς και τη λειτουργικότητα AjaxPush η οποία χρησιμοποιείται στα πλαίσια της παρούσας εργασίας για την υλοποίηση του παιχνιδιού μεταξύ δύο ανθρώπων και η αρχιτεκτονική της οποίας αναλύεται στο σχετικό κεφάλαιο.

Η βιβλιοθήκη χαρακτηρίζεται από το γεγονός ότι είναι πολύ ελαφριά (ο χρήστης πρέπει να ενσωματώσει στην εφαρμογή του ένα και μόνο jar αρχείο), οι ρυθμίσεις(configurations)που πρέπει να γίνουν είναι ελάχιστες, ενώ υπάρχει πολύ καλή τεκμηρίωση, κάτι που είναι πολύ σημαντικό. Στη σελίδα showcase δίνονται έτοιμα παραδείγματα για τη χρήση του κάθε component, μαζί με τον αντίστοιχο κώδικα και υπάρχει επίσης διαθέσιμο στην κεντρική ιστοσελίδα το επίσημο documentation πεντακοσίων (500) σελίδων.

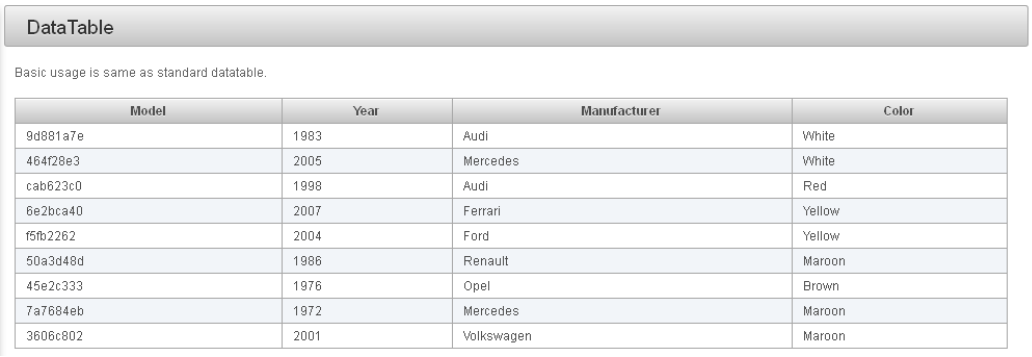
Στην Εικόνα 3.3 παρουσιάζεται ένα από τα components που προσφέρει η βιβλιοθήκη, σε συνδυασμό με τον κώδικα που δείχνει τον τρόπο χρήσης του έτσι όπως φαίνεται στη σελίδα του showcase.



Εικόνα 3.3: Παράδειγμα χρήσης ενός component των PrimeFaces. Πηγή: http://www.primefaces.org/showcase/.

Η χρήση των primefaces σε μία web σελίδα είναι πολύ απλή και σε πλήρη αντιστοιχία με τη χρήση των JavaServerFaces. Ηβιβλιοθήκη διαθέτει τις δικές της ετικέτες XML(custom tags)και σε συνδυασμό με την el (expression language) επικοινωνούν με ένα Java backing Bean, δηλαδή μία κλάση Javaη οποία τρέχει στον server και προσδίδει δυναμική λειτουργία στην σελίδα.

Παρακάτω αναλύεται ενδεικτικά η λειτουργία του `<p:dataTable>` το οποίο είναι ένα component της βιβλιοθήκης για την εύκολη και ευπαρουσίαστη απεικόνιση πινάκων δεδομένων σε μία σελίδα.



DataTable

Basic usage is same as standard datatable.

Model	Year	Manufacturer	Color
9d881a7e	1983	Audi	White
464f28e3	2005	Mercedes	White
cab623c0	1998	Audi	Red
6e2bca40	2007	Ferrari	Yellow
f5fb2262	2004	Ford	Yellow
50a3d48d	1986	Renault	Maroon
45e2c333	1976	Opel	Brown
7a7684eb	1972	Mercedes	Maroon
3606c802	2001	Volkswagen	Maroon

Εικόνα 3. 4: Παράδειγμα εμφάνισης του component <p:dataTable>. Πηγή: http://www.primefaces.org/showcase/

Στην Εικόνα 3.4 φαίνεται οπτικά το αποτέλεσμα μίας χρήσης του `<p:dataTable>` και στην Εικόνα 3.5 φαίνεται ο κώδικας της σελίδας.

```
view plain  copy to clipboard  print  ?
01. <h:form>
02.     <p:dataTable var="car" value="#{tableBean.carsSmall}">
03.         <p:column headerText="Model">
04.             <h:outputText value="#{car.model}" />
05.         </p:column>
06.
07.         <p:column headerText="Year">
08.             <h:outputText value="#{car.year}" />
09.         </p:column>
10.
11.         <p:column headerText="Manufacturer">
12.             <h:outputText value="#{car.manufacturer}" />
13.         </p:column>
14.
15.         <p:column headerText="Color">
16.             <h:outputText value="#{car.color}" />
17.         </p:column>
18.     </p:dataTable>
19. </h:form>
20.
```

Εικόνα 3.5: Ο κώδικας της JSF σελίδας με τη χρήση του component <p:dataTable>. Πηγή: http://www.primefaces.org/showcase/

Παρατηρούμε πως όλος ο πίνακας είναι ένα αντικείμενο `<p:dataTable>` ενώ οι στήλες του, είναι εμφωλευμένα αντικείμενα του τύπου `<p:column>` που επίσης προσφέρει η βιβλιοθήκη.

Παρατηρούμε επίσης ότι το value του datatable έχει την τιμή `#{tableBean.carsSmall}` . Ο συμβολισμός αυτός ανήκει στην expression language και έχει το νόημα ότι στον πίνακα αποδίδεται ένα object που ονομάζεται `carsSmall`το οποίο βρίσκεται στην κλάση `tableBean` η οποία δεν είναι παρά ένα Java Bean το οποίο τρέχει στον server. Στο object αυτό αποδίδεται ένα alias μέσω του attribute var. Το alias αυτό όπως παρατηρούμε έχει την τιμή car.

Μέσω αυτού, τα αντικείμενα `<p:column>` αποκτούν πρόσβαση στα μέλη του αντικειμένου `carsSmall`. Έτσι το πρώτο αντικείμενο p:column περιέχει ένα αντικείμενο `<h:outputText>` το οποίο έχει value `#{car.model}` δηλαδή το μοντέλο του εκάστοτε αυτοκινήτου.

Τα δεδομένα προέρχονται όπως είδαμε από την κλάση `tableBean` της οποίας ο κώδικας φαίνεται παρακάτω

```
08. import org.primefaces.examples.domain.Car;
09.
10. public class TableBean implements Serializable {
11.
12.     static {
13.         colors = new String[10];
14.         colors[0] = "Black";
15.         colors[1] = "White";
16.         colors[2] = "Green";
17.         colors[3] = "Red";
18.         colors[4] = "Blue";
19.         colors[5] = "Orange";
20.         colors[6] = "Silver";
21.         colors[7] = "Yellow";
22.         colors[8] = "Brown";
23.         colors[9] = "Maroon";
24.
25.         manufacturers = new String[10];
26.         manufacturers[0] = "Mercedes";
27.         manufacturers[1] = "BMW";
28.         manufacturers[2] = "Volvo";
29.         manufacturers[3] = "Audi";
30.         manufacturers[4] = "Renault";
31.         manufacturers[5] = "Opel";
32.         manufacturers[6] = "Volkswagen";
33.         manufacturers[7] = "Chrysler";
34.         manufacturers[8] = "Ferrari";
35.         manufacturers[9] = "Ford";
36.     }
37.
38.     private final static String[] colors;
39.
40.     private final static String[] manufacturers;
41.
42.     private List<Car> carsSmall;
43.
44.     public TableBean() {
45.         carsSmall = new ArrayList<Car>();
46.
47.         populateRandomCars(carsSmall, 9);
48.     }
49.
50.     private void populateRandomCars(List<Car> list, int size) {
51.         for(int i = 0 ; i < size ; i++)
52.             list.add(new Car(getRandomModel(), getRandomYear(), getRandomManufacturer(), getRandomColor()));
53.     }
54.
55.     public List<Car> getCarsSmall() {
56.         return carsSmall;
57.     }
58.
59.     private int getRandomYear() {
60.         return (int) (Math.random() * 50 + 1960);
61.     }
```

Εικόνα 3. 6: Ο κώδικας του backing bean. Πηγή: http://www.primefaces.org/showcase/

Όπως βλέπουμε Στην κλάση `tableBean` περιέχεται ένα ArrayList αντικείμενο με το όνομα `CarsSmall` , το οποίο δημιουργείται με τον constructor της κλάσης και του δίνονται τυχαίες τιμές οι οποίες δημιουργούνται από τη μέθοδο `populateRandomCars()` την οποία καλεί ο constructor. Από το αντικείμενο αυτό μέσω του τρόπου που είδαμε παίρνει η σελίδα τις τιμές και τις εμφανίζει στον πίνακα.

Εδώ παρατηρούμε ότι τα δεδομένα του παραδείγματος που βρίσκονται μέσα στον Java κώδικα της κλάσης, θα μπορούσαν όμως κάλλιστα να καλούνται από μία εξωτερική πηγή όπως μία βάση δεδομένων.

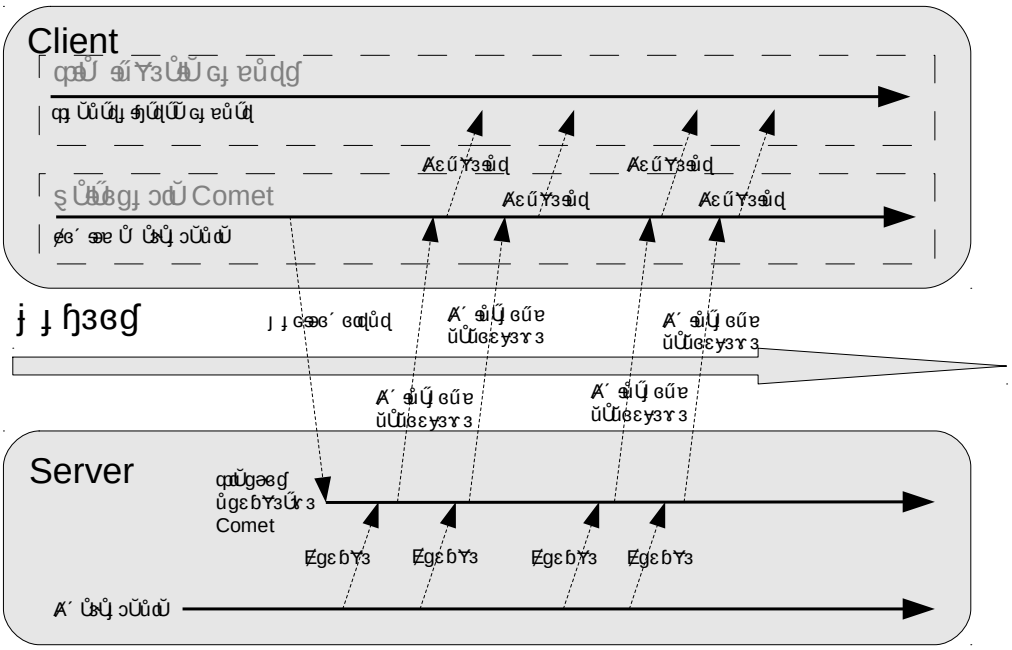
3.3.4 Comet

Με την τεχνική Comet, ένα HTTP αίτημα "παρακρατείται" και παραμένει σε εκκρεμότητα ώστε να επιτρέπει στον web server να "σπρώξει" δεδομένα προς τον browser χωρίς ο browser να το έχει αιτηθεί ρητά. Εκεί που το μοντέλο Ajax αυξάνει την διαδραστική εμπειρία για ένα πελάτη (browser) κάθε φορά, μέσω Comet γίνεται το ίδιο για πολλαπλούς πελάτες. Ο όρος Comet είναι ένας γενικός όρος που περιλαμβάνει διάφορες εναλλακτικές τεχνικές για την επίτευξη του στόχου. Όλες αυτές οι μέθοδοι βασίζονται σε χαρακτηριστικά που περιλαμβάνονται εξ' ορισμού στους browsers, όπως η Javascript παρά σε επιπλέον πρόσθετα (plug-ins που χρειάζονται εγκατάσταση). Η τεχνική Comet είναι γνωστή και με πολλούς άλλους όρους όπως: Ajax Push, Reverse Ajax, Two-way web, HTTP Streaming και HTTP server push ανάμεσα σε άλλες [17].

Στα πλαίσια του rlgameserver μέσω του Comet είναι δυνατό μόλις συμβεί μία κίνηση από τον παίκτη Α ο web server να αποστείλει την κίνηση μόλις εκδηλωθεί στον αντίπαλο παίκτη Β. Μέσω Comet είναι δυνατή η δημιουργία εφαρμογών που βασίζονται σε συμβάντα (event-based). Όταν ένα συμβάν εκδηλωθεί στον server, ο server αφού εκτελέσει τις σχετικές επεξεργασίες μπορεί να μεταφέρει δεδομένα προς τον/τους πελάτες χωρίς αυτό να έχει ζητηθεί μέσω αίτησης.

Τα Primefaces υλοποιούν το Comet μέσω του γνωστού Atmosphere framework και ονομάζουν την τεχνολογία αυτή PrimePush. Οτρόπος με τον οποίο χρησιμοποιείται για να υλοποιηθούν το chat και το παιχνίδι ανθρώπου εναντίον ανθρώπου θα εξηγηθεί αναλυτικά στο αντίστοιχο κεφάλαιο.

Η αρχιτεκτονική του Comet παρουσιάζεται σχηματικά στην Εικόνα 3.7.



Εικόνα 3. 7: Μοντέλο λειτουργίας Comet (Δήμας 2011)

3.4 Βάση Δεδομένων MySQL

Η MySQLείναι ένα πασίγνωστο Σύστημα Διαχείρισης Σχεσιακών Βάσεων Δεδομένων (RDMBS), και στα πλαίσια της παρούσας διπλωματικής επιλέχθηκε για την αποθήκευση των δεδομένων που αφορούν τον server παιχνιδιού που κατασκευάστηκε.

3.5 Hibernate

Το Hibernate ORM (Hibernate εν συντομία) είναι μία Object Relational mapping βιβλιοθήκη για τη γλώσσα Java που παρέχει ένα framework για την αντιστοίχηση (mapping) ενός αντικειμενοστραφούς μοντέλου δεδομένων (object oriented) σε μία παραδοσιακή σχεσιακή βάση δεδομένων. Το Hibernate επιλύει διάφορα προβλήματα αναντιστοιχίας που εμφανίζονται μεταξύ των 2 αυτών μοντέλων, αντικαθιστώντας τις απευθείας προσβάσεις στη βάση δεδομένων με υψηλού επιπέδου (high level) συναρτήσεις που διαχειρίζονται αντικείμενα της Java.

Το Hibernate είναι ελεύθερο λογισμικό, και διατίθεται υπό την GNU Lesser General Public License [24].

Η κύρια λειτουργία του Hibernate είναι η αντιστοίχηση κλάσεων της Java σε πίνακες βάσεων δεδομένων (και από τύπους δεδομένων της Javaσε τύπους δεδομένων της SQL). Παρέχει επίσης λειτουργίες διευκόλυνσης για την δημιουργία ερωτημάτων και την ανάκτηση δεδομένων από τις βάσεις. Παράγει SQL κλήσεις και απαλλάσσει τον προγραμματιστή από την χειροκίνητη διαχείριση και μετατροπή των ανακτηθέντων δεδομένων. Οι εφαρμογές που χρησιμοποιούν Hibernate είναι εύκολα μεταφερόμενες ανάμεσα στις υποστηριζόμενες βάσεις δεδομένων, με μικρή επιβάρυνση στην απόδοση.

3.6 Apache Tomcat

Ο Apache Tomcat (ή απλά Tomcat, και πρώην Jakarta Tomcat) είναι ένας web server ανοιχτού κώδικα και servlet container, που αναπτύχθηκε από την Apache Software Foundation (ASF). Ο Tomcat εφαρμόζει τις προδιαγραφές των Java Servlet και JavaServerPages(JSP) απότη Sun Microsystems, και παρέχει έναν HTTP web server αμιγώς κατασκευασμένο σε Java. Στην απλούστερη δυνατή του υλοποίηση, ο Tomcat τρέχει σε μία και μόνο διεργασία του λειτουργικού συστήματος. Η διαδικασία εκτελείται από μια εικονική μηχανή Java(JVM). Κάθε HTTP request από έναν browser προς τον Tomcat επεξεργάζεται σε ξεχωριστό thread του process [20].

Ο Tomcat περιλαμβάνει εργαλεία για τις ρυθμίσεις και τη διαχείριση του, αλλά μπορεί επίσης να ρυθμιστεί με επεξεργασία αρχείων ρυθμίσεων τύπου XML.

Η επιλογή του Tomcat έγινε με κριτήριο το γεγονός οτι είναι ελαφρύτερος και πιο απλός από τους υπόλοιπους JavaEE application servers όπως είναι ο glassfish και ο JBOSS, εφόσον δεν υλοποιεί ολόκληρη την αρχιτεκτονική της JAVA EE καθώς και για το γεγονός ότι θεωρείται μία ώριμη και δοκιμασμένη επιλογή. Οι επιπλέον λειτουργικότητες από το JavaEE stack που χρειάστηκαν στα πλαίσια του rlgameserver όπως για παράδειγμα τα JavaServerFaces, ενσωματώθηκαν υπό τη μορφή αυτόνομης βιβλιοθήκης (.jar) στο projectπου κατασκευάστηκε.

Κεφάλαιο 4
Συστήματα Βαθμολόγησης (rating) παικτών σε τεχνικά παίγνια

4.1 Σύστημα βαθμολογίας ELO

Για να μπορεί να μετρηθεί η συγκριτική ικανότητα των παικτών σε παιχνίδια όπως το σκάκι, έχουν δημιουργηθεί κάποια συστήματα βαθμολογίας, με πιο γνωστό απ' αυτά το σύστημα ELO που αντλεί το όνομά του από τον Ούγγρο καθηγητή φυσικής Aprad Elo.

Το σύστημα αυτό εφευρέθηκε αρχικά σαν μία βελτιωμένη μορφή βαθμολόγησης παικτών στο σκάκι και σήμερα χρησιμοποιείται επίσης σε πολλά άλλα παιχνίδια.

Η διαφορά στη βαθμολογία μεταξύ δύο παικτών εξυπηρετεί σαν εκτιμήτρια του αποτελέσματος ενός παιχνιδιού που θα λάβει χώρα μεταξύ τους. Εάν δύο παίκτες με ίδια βαθμολογία παίξουν ο ένας εναντίον του άλλου, αναμένεται να έχουν 50% πιθανότητα νίκης ο καθένας. Ένας παίκτης του οποίου η βαθμολογία είναι 100 πόντοι μεγαλύτεροι απ' αυτήν του αντιπάλου του αναμένεται να κερδίζει 64% και αν η διαφορά είναι 200 πόντοι το αναμενόμενο σκορ για τον πιο ισχυρό αναμένεται να είναι 76%.

Η βαθμολογία ELO αναπαρίσταται από έναν αριθμό ο οποίος αυξάνεται και ελαττώνεται ανάλογα με το αν ο παίκτης χάνει ή κερδίζει σε παιχνίδια μεταξύ παικτών ενταγμένων στο βαθμολογικό σύστημα. Μετά από κάθε παιχνίδι, ο νικητής παίρνει πόντους από τον χαμένο. Ο συνολικός αριθμός πόντων που χάνονται ή κερδίζονται μετά από ένα παιχνίδι, καθορίζεται από τη διαφορά στη βαθμολογία μεταξύ νικητή και χαμένου. Σε μία σειρά παιχνιδιών μεταξύ ενός υψηλόβαθμου κι ενός χαμηλόβαθμου παίκτη, ο υψηλόβαθμος αναμένεται να κερδίσει τα περισσότερα απ' αυτά. Τις φορές που θα κερδίζει ο υψηλόβαθμος, θα αφαιρούνται μόνο λίγοι πόντοι από τον χαμηλόβαθμο. Ωστόσο εάν ο χαμηλόβαθμος πετύχει μια νίκη, τότε θα αφαιρεθούν πολλοί πόντοι απ τον αντίπαλό του και θα δοθούν σε αυτόν. Ο χαμηλόβαθμος παίκτης θα κερδίσει θα κερδίσει επίσης μερικούς πόντους απ' τον υψηλόβαθμο στην περίπτωση της ισοπαλίας. Αυτό καθιστά το βαθμολογικό σύστημα αυτοδιορθούμενο. Δηλαδή ένας παίκτης του οποίου η βαθμολογία είναι χαμηλότερη συγκριτικά με την αντικειμενική του ικανότητα στο παιχνίδι, μακροπρόθεσμα να επιτύχει περισσότερες νίκες απ' όσες προβλέπεται βάσει της βαθμολογίας του, και ως εκ τούτου θα ανέβει σιγά-σιγά βαθμολογικά μέχρις ότου η βαθμολογία του να αντικατοπτρίζει την πραγματική του ικανότητα.

Ο Elo δημιούργησε το σύστημά του κάνοντας την βασική υπόθεση πως η απόδοση του κάθε παίκτη σε κάθε παιχνίδι είναι μία μεταβλητή που ακολουθεί κανονική κατανομή. Αν και, ένας παίκτης μπορεί να αποδώσει καλύτερα ή χειρότερα από παιχνίδι σε παιχνίδι, ο Elo θεώρησε πως η μέση τιμή των αποδόσεων ενός οποιουδήποτε παίκτη μεταβάλλεται αρκετά αργά συναρτήσει του χρόνου. Ο Elo θεώρησε ως πραγματική ένδειξη της ικανότητας ενός παίκτη τη μέση τιμή αυτής της τυχαίας μεταβλητής.

Στατιστικοί έλεγχοι που έγιναν εκ των υστέρων υπέδειξαν πως η απόδοση στο σκάκι σχεδόν σίγουρα δεν ακολουθεί κανονική κατανομή, αφού οι πιο αδύναμοι παίκτες, έχουν υψηλότερη πιθανότητα επιτυχίας σε σχέση με αυτή που προβλέπει το μοντέλο του Elo. Έτσι η Αμερικανική Σκακιστική ομοσπονδία (USCF) και μερικά σκακιστικά sitesχρησιμοποιούν για τον υπολογισμό του ELO μία φόρμουλα που βασίζεται στην λογιστική κατανομή. Η FIDE εξακολουθεί να

χρησιμοποιεί την κανονική κατανομή. Και οι δύο αυτές κατανομές λειτουργούν αρκετά καλά για έναν μεγάλο αριθμό παιχνιδιών [22].

4.2 Σύστημα βαθμολογίας Glicko

Τα βαθμολογικά συστήματα Glicko rating system και Glicko-2 rating system εφευρέθηκαν από τον Mark Glickman σαν μία βελτιωμένη έκδοση του Συστήματος ELO με αρχικό σκοπό να χρησιμοποιηθούν σαν ένα σύστημα βαθμολογίας στο σκάκι. Η βασική συνεισφορά του Glickman ήταν ο παράγοντας αξιοπιστίας «ratings reliability", ο οποίος συμβολίζεται με RD από τα αρχικά της φράσης Ratings Deviation.

Ο RD μετράει την ακρίβεια της βαθμολογίας ενός παίκτη. Για παράδειγμα, ένας παίκτης με βαθμολογία 1500 και RD 50 η πραγματική του αξία μπορεί να κυμαίνεται από 1400 έως και 1600 με ένα διάστημα εμπιστοσύνης 95%. Για να υπολογιστεί αυτό το εύρος, προστίθεται και αφαιρείται το διπλάσιο του RD από τη βαθμολογία του παίκτη. Μετά από ένα παιχνίδι, το πόσο μεταβάλλεται η βαθμολογία ενός παίκτη εξαρτάται από το RD του. Όσο μικρότερο είναι το RD τόσο μικρότερη είναι και η αλλαγή. Το ίδιο το RD ελαττώνεται μετά από ένα παιχνίδι, αλλά αυξάνεται αργά μετά από περιόδους που ο παίκτης είναι ανενεργός [23].

4.3 Υλοποίηση του Glicko rating για τις ανάγκες του rlgameserver.

Η ύπαρξη ενός συστήματος αξιολόγησης των παικτών στο rlgame μπορεί να προσφέρει αρκετά, τόσο στην αύξηση του ενδιαφέροντος του κοινού για το ίδιο το παιχνίδι, αφού εντείνει την προσπάθεια των παικτών για εκμάθηση του παιχνιδιού και βελτίωση της βαθμολογίας τους, όσο και στο να κατανοήσουμε κάποιες πτυχές του παιχνιδιού, όπως για παράδειγμα το πόση τεχνική περιέχει.

Όσο μεγαλύτερη είναι η βαθμολογική διαφορά μεταξύ του πρώτου και του τελευταίου βαθμολογικά παίκτη, τόσο πιο τεχνικό θεωρείται ένα παιχνίδι. Ενδεικτικά αναφέρεται πως στο σκάκι οι βαθμολογίες ποικίλλουν από μικρότερες του 1000 έως και 2800. Αντίστοιχα στο τάβλι, επειδή εμπεριέχεται και ο παράγοντας τύχη, το δυνατό εύρος των βαθμολογιών είναι αρκετά μικρότερο. (Για παράδειγμα στην Ελληνική Ομοσπονδία Backgammon οι βαθμολογίες

κυμαίνονται από 1394 έως και 1711 ανάμεσα σε παίκτες που έχουν συμπληρώσει ένα ελάχιστο αριθμό 400 παρτίδων).

Θα ήταν λοιπόν αρκετά ενδιαφέρον να παρατηρηθεί το εύρος βαθμολογιών που θα διαμορφωθεί στο παιχνίδι rlgame. Επισημαίνεται εδώ, πως όπως ακριβώς ένας άνθρωπος έχει βαθμολογία, έτσι μπορεί να έχει και ένα bot το οποίο παίζει το παιχνίδι αυτό. Θα μπορούσαν λοιπόν στον rlgameserver να συμμετέχουν εκτός από ανθρώπους και προγράμματα υπολογιστή με τη δική τους βαθμολογία το καθένα, και η δυνατότητα αυτή θα μπορούσε να αποτελέσει ένα είδος testing για άτομα τα οποία ασχολούνται με τεχνητή νοημοσύνη και machine learning οι οποίοι αποφάσισαν να εφαρμόσουν σχετικές τεχνικές (π.χ. με χρήση νευρωνικών δικτύων, ενισχυτικής μάθησης ή άλλων μεθόδων) για τη δημιουργία ενός προγράμματος που παίζει το παιχνίδι αυτό.

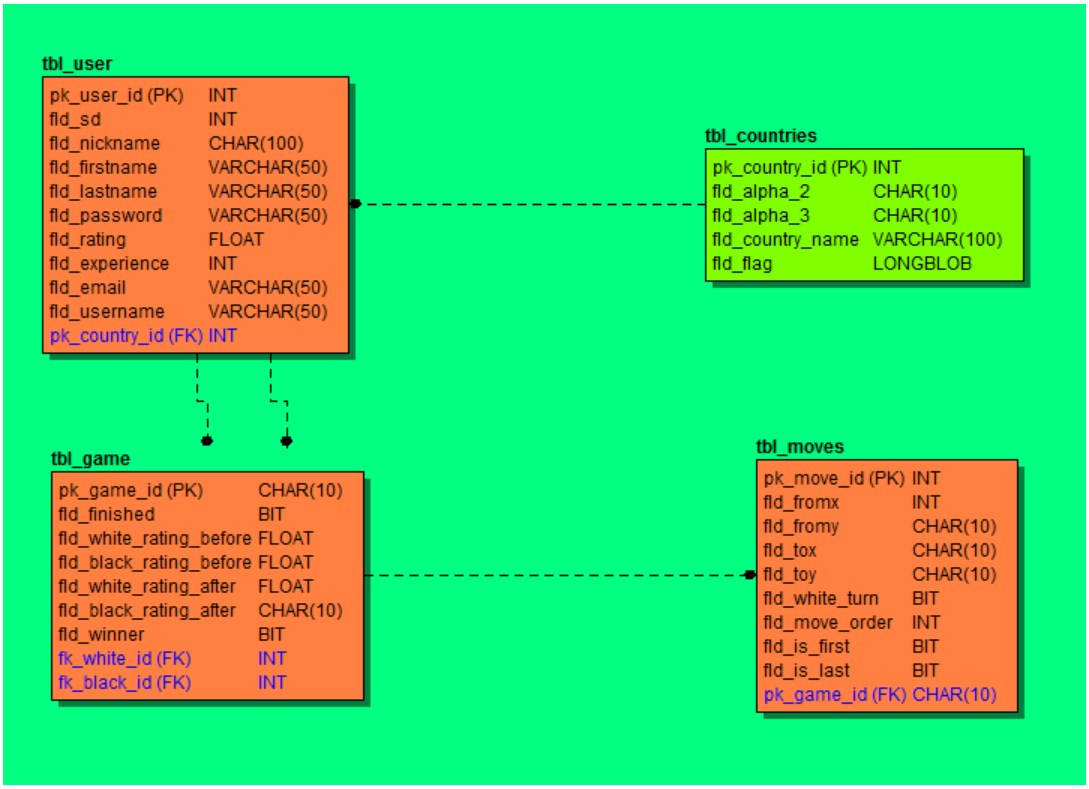
Για να ενσωματωθεί η δυνατότητα του rating στις δυνατότητες του rlgameserver χρησιμοποιήθηκε έτοιμη κλάση java που έχει αναπτυχθεί από τους Alex Nicolaou και Jay Steele. Η κλάση έχει ως μέλη 2 μεταβλητές τύπου double, τις rating και RD οι οποίες αρχικοποιούνται στις τιμές 1500 και 350, εκτός και αν δίνονται διαφορετικές τιμές στον constructor της κλάσης.

Κεφάλαιο 5

Υλοποίηση του rlgameserver

5.1Δημιουργία της βάσης δεδομένων

Για τις ανάγκες σε αποθήκευση δεδομένων του εξυπηρετητή παιχνιδιού, στήθηκε ένας MySQL server και δημιουργήθηκε μία σχεσιακή βάση δεδομένων. Για τις ανάγκες διαχείρισης της βάσης δεδομένων χρησιμοποιήθηκε το διαχειριστικό περιβάλλον phpmyadmin το οποίο τρέχει σε έναν apache2 web server κάνοντας χρήση της scripting γλώσσας PHP.Τοσχεσιακό μοντέλο της εν λόγω βάσης φαίνεται στην Εικόνα 5.1.



Εικόνα 5.1: Το σχεσιακό σχήμα της βάσης δεδομένων

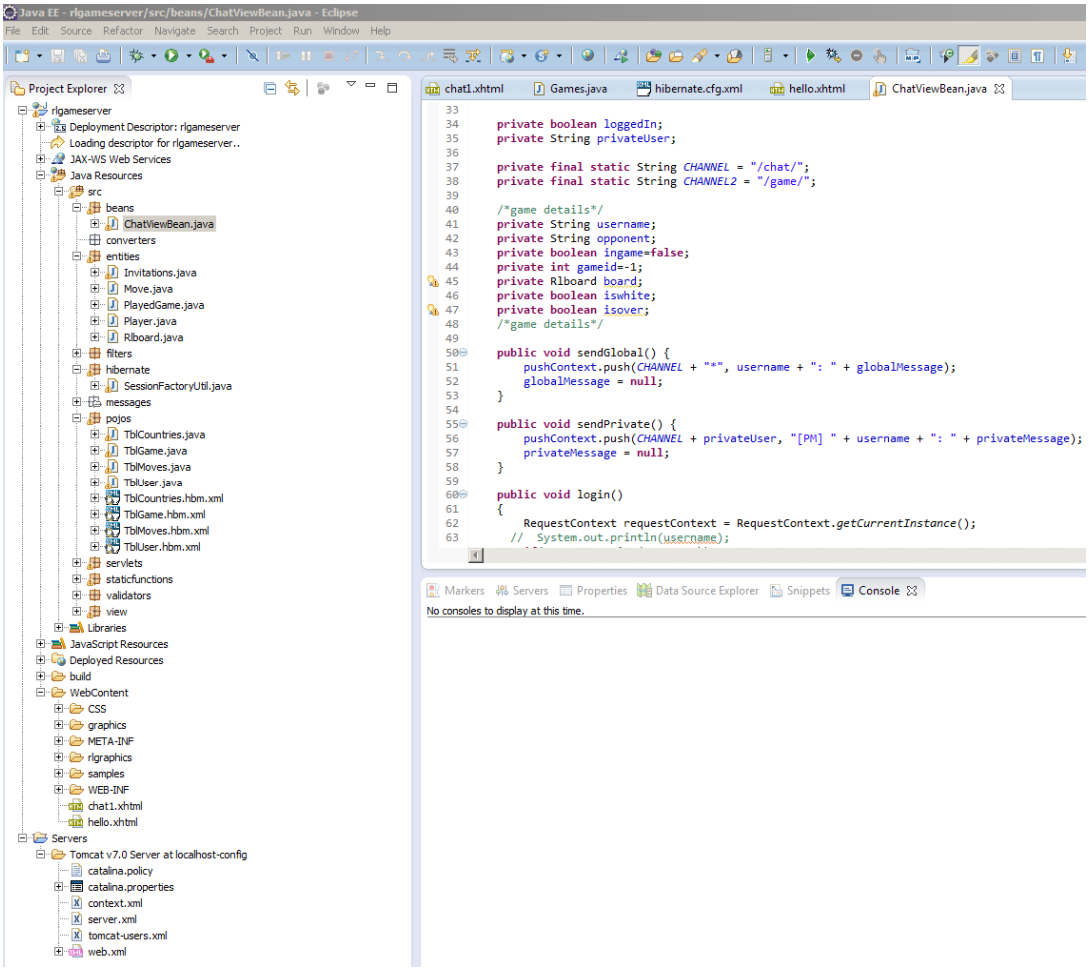
Η βάση αποτελείται από 4 πίνακες. Στον tbl_user αποθηκεύονται τα στοιχεία του παίκτη (όνομα, επίθετο, nickname, τα στοιχεία της βαθμολογίας του στο παιχνίδι κ.α.)

Ο πίνακας tbl_countrieseίναι ένας βοηθητικός πίνακας στον οποίο έχει αποθηκευθεί μία λίστα των κρατών του κόσμου ούτως ώστε να υπάρχει μελλοντικά η δυνατότητα να καταχωρείται και αυτό το στοιχείο που αφορά τους παίκτες στη βάση δεδομένων.

Στον πίνακα tbl_game αποθηκεύονται τα στοιχεία όλων των παιχνιδιών που έχουν παιχτεί (ποιοι ήταν οι αντίπαλοι, ποιος ο νικητής κ.λ.π.), ενώ στον tbl_moves αποθηκεύονται όλες οι κινήσεις που παίχτηκαν στα πλαίσια κάποιου συγκεκριμένου παιχνιδιού, προκειμένου όλα τα παιχνίδια να μπορούν να ανασκευαστούν από την πρώτη μέχρι την τελευταία κίνηση και να χρησιμοποιηθούν για ερευνητικούς ή διάφορους άλλους σκοπούς.

5.2 Δημιουργία Java Web Project με τη χρήση του Eclipse IDE

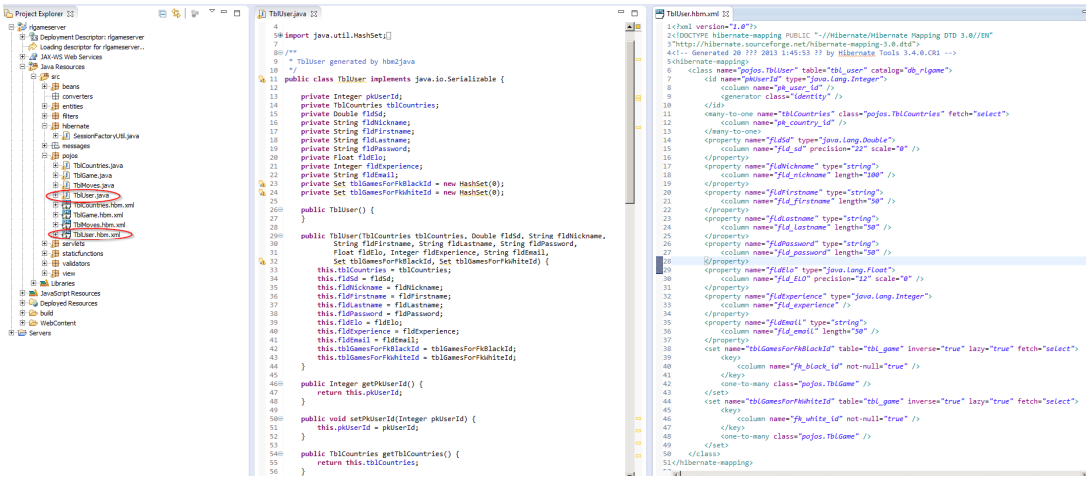
Η συγγραφή του κώδικα του rlgameserver έγινε με τη βοήθεια του ολοκληρωμένου περιβάλλοντος (IDE) Eclipse που παρέχει αρκετές διευκολύνσεις στη διαδικασία της ανάπτυξης δικτυακών εφαρμογών (όπως codecompletion, projectbuild, deployment κ.α.). Η δόμηση του κώδικα στα πλαίσια ενός Java Web projectφαίνεται στην Εικόνα 5.2.



Εικόνα 5. 2: Δόμηση του κώδικα στο Eclipse

5.3 Δημιουργία του ενδιάμεσου layerπρόσβασης με τη χρήση του HibernateUtilities

Όπως αναφέρθηκε, η επικοινωνία , η ανάκτηση και αποθήκευση δεδομένων από και προς τη βάση επιτυγχάνεται μέσω ενός ενδιάμεσου layer κλάσεων java τις οποίες χρησιμοποιεί το hibernate και οι οποίες αποκαλούνται για συντομία POJOS (Plain Old Java Objects). Οι κλάσεις αυτές αντιστοιχίζονται με τους πίνακες της βάσης δεδομένων μέσω ειδικών xml αρχείων που έχουν την επέκταση .hbm (hibernate mapping). Άπαξ και γίνει η αντιστοίχηση αυτή, ο προγραμματιστής κάνοντας χρήση των δυνατοτήτων του hibernate μπορεί να παίρνει αυτόματα τα δεδομένα της βάσης σαν αντικείμενα POJOS και δεν χρειάζεται να ασχολείται με τη μετατροπή των resultsets σε objects. Ένα παράδειγμα για το πως γίνεται mapped ο πίνακας tbl_user της βάσης δεδομένων σε POJO μέσω ενός .hbm.xml αρχείου φαίνεται στην παρακάτω Εικόνα 5.3.

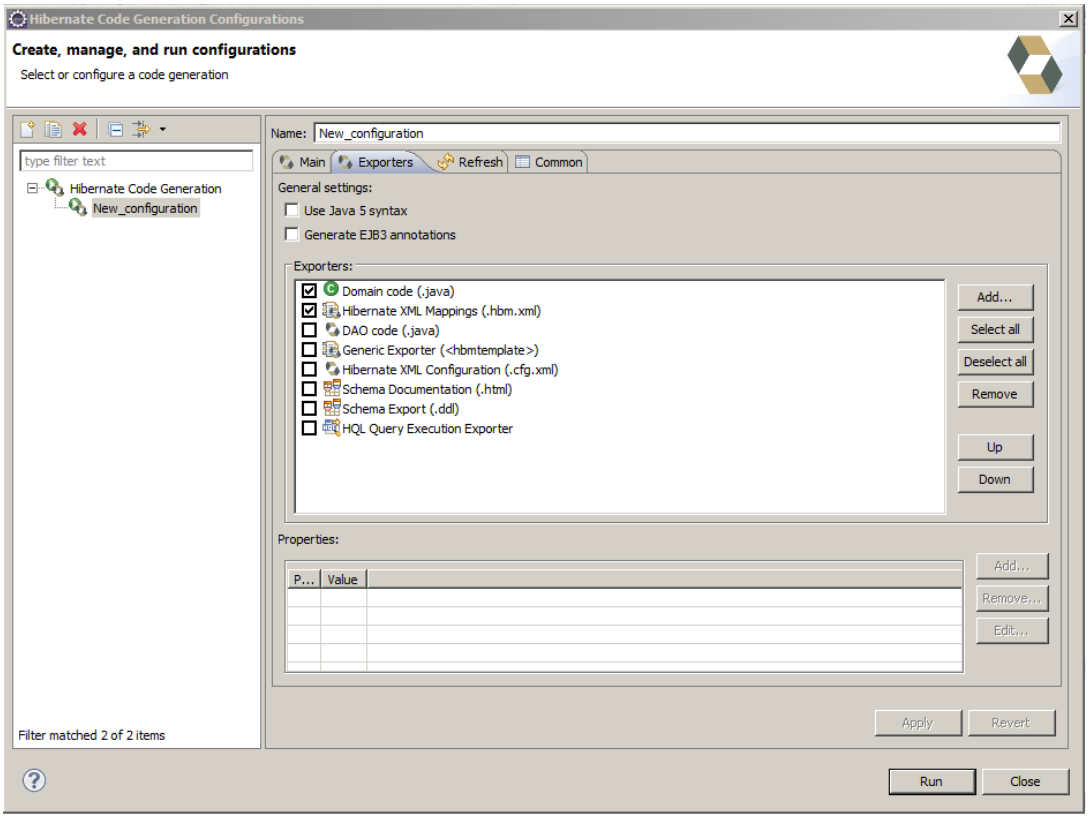


Εικόνα 5. 3: Απεικόνιση πίνακα βάσης δεδομένων σε κλάση Java μέσω αρχείου .hbm.xml

Όπως φαίνεται στην εικόνα, κάθε πεδίο του πίνακα της βάσης, απεικονίζεται σε έναν αντίστοιχο τύπο δεδομένων Java με παραμέτρους που καθορίζονται στο .hbm.xml αρχείο. Παρατηρούμε επίσης πως οι συσχετίσεις μεταξύ πινάκων μεταφράζονται σε συσχετίσεις κλάσεων Java. Για

παράδειγμα βλέπουμε πως η κλάση TblUser που δημιουργήθηκε ως απεικόνιση του πίνακα tbl_user περιέχει ως μέλος της δύο σετ από αντικείμενα της κλάσης tblGame που εκφράζουν την 1 προς n συσχέτιση του πίνακα tbl_user με τον πίνακα tbl_game (αφού ένας παίκτης συμμετείχε σε n παιχνίδια ως λευκός και σε n παιχνίδια ως μάυρος).

Επειδή ο κώδικας για την συγγραφή των Pojos και των αρχείων απεικόνισης .hbm.xml είναι αρκετός και επιρρεπής σε λάθη, το Hibernate με τη χρήση των Hibernate Utilities μας δίνει την δυνατότητα να τον παράξουμε αυτόματα. Για τις ανάγκες τις εργασίας το Hibernate Utilities χρησιμοποιήθηκε σαν plugin του eclipse και ο κώδικας παράχθηκε με μερικά κλικ όπως φαίνεται στην Εικόνα 5.4.



Εικόνα 5. 4: Αυτόματη παραγωγή κώδικα (POJOS, hbm.xml) με τη χρήτη των Hibernate Utilities

Για να δούμε ένα παράδειγμα χρήσης του Hibernate ας υποθέσουμε ότι θέλουμε να προσθέσουμε έναν νέο παίκτη στη βάση δεδομένων υπό την προϋπόθεση πως αυτός δεν

υπάρχει ήδη σε αυτήν. Για να γίνει αυτό υλοποιήθηκαν 2 μέθοδοι java με τη χρήση των δυνατοτήτων του Hibernate οι οποίες παρατίθενται ακολούθως:

```
public Boolean searchPlayerInDB(String player)

{

	Session session = SessionFactoryUtil.getSessionFactory().getCurrentSession();

	session.getTransaction().begin();

	List us= session.createQuery("from TblUser").list();

	for (Iterator it = (Iterator) us.iterator() ; it.hasNext();  )

	{

		TblUser user= (TblUser) it.next();

		if(player.equals(user.getFldNickname()))

		{

			this.players.add(user);

			return true;

		}

	}

	session.getTransaction().commit();

	return false;

}
```

Η μέθοδος `searchPlayerInDB()` δέχεται ως όρισμα ένα αλφαριθμητικό το οποίο αντιστοιχεί στο ψευδώνυμο (nickname) με το οποίο επιχείρησε κάποιος παίκτης να εισέλθει στον rlgameserver. Η μέθοδος λαμβάνει το Hibernate Session που έχει δημιουργηθεί και ξεκινάει μία συναλλαγή με τη βάση (Transaction). Με ένα query τύπου HQL (Hibernate Query Language) λαμβάνουμε σε μια λίστα όλες τις εγγραφές του πίνακα, οι οποίες μετατρέπονται με ένα απλό typecasting σε αντικείμενα τύπου TblUser που έχουμε ορίσει. Πλέον είναι πολύ απλό να γίνει η αναζήτηση του ονόματος που ψάχνουμε με τη χρήση μεθόδων της Java. Η μέθοδος επιστρέφει true αν ο παίκτης υπάρχει ήδη στη βάση δεδομένων, και false αν το όνομα του παίκτη δεν εντοπιστεί.

```
Public void addPlayer(String player)

    {

       if(searchPlayerInDB(player)==false)

       {

              TblUser us= new TblUser();

              us.setFldElo((float) 1500);

              us.setFldExperience(0);

              us.setFldSd(350.00);

              us.setFldNickname(player);



       Session session = SessionFactoryUtil.getSessionFactory().getCurrentSession();

              session.getTransaction().begin();

              session.save(us);

              session.getTransaction().commit();
```

```
            this.players.add(us);

        }

        else

        {}

    }
```

Σε συνέχεια της προηγούμενης μεθόδου, η `addPlayer()` καλεί την `searchPlayerInDB()` και αν αυτή επιστρέψει false, που σημαίνει ότι δεν έχει εντοπίσει παίκτη με το συγκεκριμένο ψευδώνυμο στη βάση δεδομένων, τότε δημιουργεί ένα νέο αντικείμενο τύπου TblUser και ξεκινάει ένα hibernate transaction κατά τη διάρκεια του οποίου με την απλή εντολή

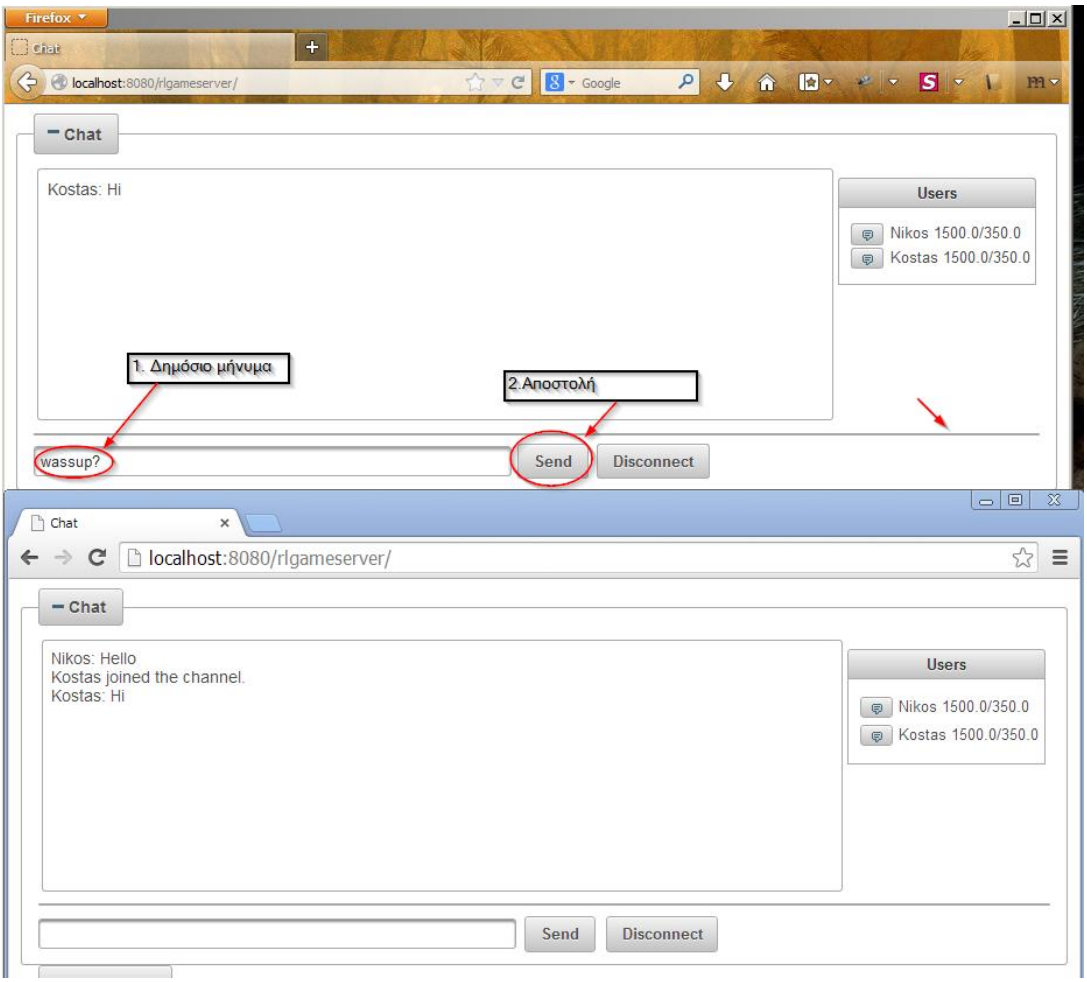
```
session.save(us);
```

Σώζονται τα στοιχεία του νεοδημιουργηθέντα παίκτη στη βάση. Παρατηρούμε ότι ο κώδικας που γράφτηκε στις 2 μεθόδους είναι σχεδόν αμιγώς κώδικας java με ελάχιστο εμφωλευμένο κώδικα γλώσσας διατύπωσης ερωτημάτων (Query Language).

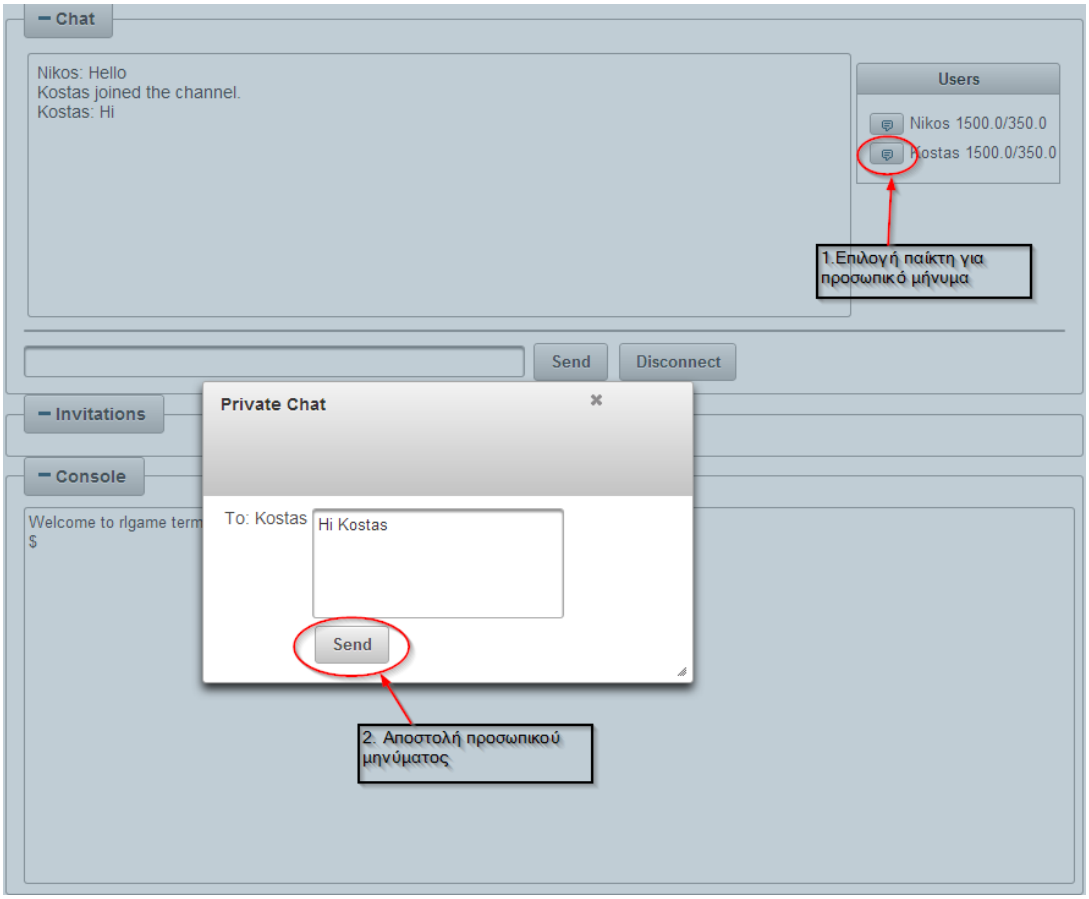
5.4 Δημιουργία της κύριας ιστοσελίδας με τη χρήση Primefaces και Backing Beans

5.4.1 Υποδομή για δημόσιο και ιδιωτικό chat.

Είναι κοινός τόπος για τους δικτυακούς servers παιχνιδιών να προσφέρουν τη δυνατότητα επικοινωνίας μεταξύ των παικτών τους. Στην περίπτωση του rlgameserver υλοποιήθηκε η δυνατότητα επικοινωνίας των παικτών με chat έτσι όπως αυτό έχει υλοποιηθεί στο demo showcase των PrimeFaces. Οι παίκτες μπορούν να στέλνουν μηνύματα τα οποία μπορεί να δει οποιοσδήποτε βρίσκεται συνδεδεμένος στην εφαρμογή τη στιγμή εκείνη καθώς και προσωπικά μηνύματα ο ένας προς τον άλλον στα οποία δεν έχουν πρόσβαση οι υπόλοιποι παίκτες, όπως φαίνεται στις Εικόνες 5.5 και 5.6.



Εικόνα 5. 5: Δημόσιο chat στα πλαίσια του rlgameserver



Εικόνα 5. 6: Αποστολή προσωπικού μηνύματος

Το παράδειγμα του chat είναι ιδανικό για να εξηγηθεί και η τεχνολογία PrimePush μέσω της οποίας υλοποιούνται όλες οι ασύγχρονες κλήσεις από τον server στους clients (δηλαδή η αποστολή μηνυμάτων προς τους clients χωρίς να έχουν προηγηθεί requests απ' αυτούς). Όταν ο χρήστης πατάει το κουμπί Send για να αποστείλει το μήνυμα στους συμπαίκτες του, όπως γνωρίζουμε ο client του στέλνει ένα HTTP requeststον server, ο server την παραλαμβάνει και αποθηκεύει το περιεχόμενο του μηνύματος σε μία μεταβλητή ενός backing bean (δηλαδή μίας Java κλάσης που τρέχει στον server και παραλαμβάνει όλα τα αιτήματα που αποστέλλονται από τη συγκεκριμένη σελίδα). Απομένει όμως το κομμάτι της αποστολής του μηνύματος στους παραλήπτες του, δηλαδή σε άλλους clients στο δίκτυο. Αυτή η τελευταία λειτουργία πρέπει να γίνει ασύγχρονα καθώς οι clients αυτοί δεν έχουν στείλει κάποιο αίτημα στον server. Τη λύση του προβλήματος αυτού δίνει η τεχνολογία PrimePush που υλοποιούν τα PrimeFaces, και η οποία συμπεριλαμβάνεται στην ομπρέλα τεχνικών που ονομάζουμε Comet. Ας

παρακολουθήσουμε όλη τη διαδικασία αποστολής του μηνύματος μέσα από τα αντίστοιχα τμήματα του κώδικα στον client και στον server.

```
<p:dialogwidgetVar="pChat"header="Private Chat" modal="true"
        showEffect="fade"hideEffect="fade">
        <h:panelGridid="privateChatContainer"columns="2"
                columnClasses="vtop,vtop">
                <p:outputLabelfor="pChatInput"
                        value="To: #{chatviewbean.privateUser}"/>
                <p:inputTextareaid="pChatInput"
                        value="#{chatviewbean.privateMessage}" rows="5"cols="30"/>
                <p:spacer/>
                <p:commandButtonvalue="Send"
                        actionListener="#{chatviewbean.sendPrivate}"
                        oncomplete="pChat.hide()"/>
        </h:panelGrid>
</p:dialog>
```

Βλέπουμε πως στην πλευρά του client το chat υλοποιείται με ένα σύνολο Primefaces components που είναι εμφωλευμένα σε ένα container. Ο χρήστης γράφει το μήνυμά του μέσα στο component τύπου `<p:inputTextArea>` και αποστέλλει το μήνυμα πατώντας ένα κουμπί που είναι ένα component τύπου `<p:commandButton>`. Αυτό με τη σειρά του, μέσω του attribute actionListener στέλνει ένα μήνυμα στον server το οποίο παραλαμβάνει η μέθοδος `sendPrivate()` που ανήκει στο backing bean `chatviewbean` και της οποίας ο κώδικας ακολουθεί.

```
publicvoid sendPrivate()

{

	pushContext.push(CHANNEL + privateUser, "[PM] " + username + ": " +
		privateMessage);

	privateMessage = null;

}
```

Όπως βλέπουμε το μόνο που κάνει η συνάρτηση είναι να καλεί το μηχανισμό PrimePush μέσω της κλήσης της μεθόδου `push()` στην οποία αποδίδονται ως ορίσματα το κανάλι επικοινωνίας με τον χρήστη στο οποίο επιδιώκουμε να στείλουμε το μήνυμα, το όνομά μας και το περιεχόμενο του μηνύματος που στείλαμε το οποίο μέσω του προηγούμενου αιτήματος έχει αποθηκευθεί στην μεταβλητή `privateMessage`.

Ομηχανισμός primePush έχει δημιουργήσει ένα κανάλι επικοινωνίας για κάθε χρήστη που έχει εισέλθει στο server, την στιγμή που έκανε login, όπως φαίνεται από την μέθοδο `login()`που ακολουθεί:

```
publicvoid login()

{

	RequestContext requestContext = RequestContext.getCurrentInstance();

	//  System.out.println(username);

	if(users.contains(username))

	{

		//System.out.println(username);

		loggedIn = false;
```

```
            FacesContext.getCurrentInstance().addMessage(null, new
            FacesMessage(FacesMessage.SEVERITY_ERROR, "Username taken", "Try with
            another       username."));

            requestContext.update("growl");

        }

        else

        {

            users.addPlayer(username);

            pushContext.push(CHANNEL + "*", username + " joined the channel.");

            requestContext.execute("subscriber.connect('/" + username + "')");

            requestContext.execute("subscriber2.connect('/" + username + "')");

            loggedIn = true;

        }

}
```

Η μέθοδος `login()` μεταξύ άλλων, χρησιμοποιεί το utility `requestContext` που προσφέρουν τα Primefaces. Το utility αυτό, δίνει τη δυνατότητα στο backing bean ενός client να προκαλέσει από τη μεριά του server ανανέωση κάποιου component στον client, ή να τρέξει κάποιον javascript κώδικα. Στην προκειμένη περίπτωση η requestContext δίνει την εντολή να εκτελεστεί η javascript συνάρτηση `subscriber2.connect()` με όρισμα το όνομα του παίκτη που μόλις έχει εισέλθει. Η συνάρτηση αυτή επιδρά σε ένα Primefaces component που λέγεται `<p:socket>` και το οποίο βρίσκεται στον client για το άνοιγμα του καναλιού επικοινωνίας με τον server.

```
<p:socketonMessage="handleMessage"channel="/chat"autoConnect="false"

        widgetVar="subscriber">

        <p:ajaxevent="message"update=":form1:users"/>
```

```
        <p:ajax event="message" update=":form2:gamecontainer" />

</p:socket>
```

Όπως βλέπουμε στον κώδικα του component, μόλις δεχτεί κάποιο μήνυμα από το κανάλι επικοινωνίας του οποίο το όνομα δίνεται από το attribute channel καλεί μία javascript συνάρτηση της οποίας το όνομα ορίζεται από το event `onMessage`. Ταυτόχρονα, μέσα στο component έχουν εμφωλευτεί δύο components τύπου p:ajax τα οποία προκαλούν την ασύγχρονη ανανέωση των οριζόμενων στοιχείων (που στην περίπτωση αυτή είναι το `form1:users` δηλαδή το containerπου περιλαμβάνει το chat)για να αποτυπωθούν οι αλλαγές, δηλαδή να εμφανιστεί το ληφθέν μήνυμα στον client.

Η συνάρτηση javascript που τρέχει κατά το eventonMessageφαίνεται παρακάτω.

```
function handleMessage(data){

        var chatContent = $(PrimeFaces.escapeClientId('form1:public'));

        chatContent.append(data + '<br />');

        PrimeFaces.scrollTo('form2:gamecontainer');

        //keep scroll

        chatContent.scrollTop(chatContent.height());

}
```

Όπως βλέπουμε η συνάρτηση παίρνει σαν όρισμα το κείμενο του μηνύματος και κάνει το κάνει append στο υπόλοιπο σώμα κειμένου που έχει εμφανιστεί μέχρι εκείνη τη στιγμή στον client. Κατόπιν όπως είδαμε μέσω του `p:ajax` το κομμάτι της σελίδας που αφορά το chat ανανεώνεται για να εμφανιστούν οι αλλαγές.

5.4.2 Command line interface

Πολλοί online servers παιχνιδιών εκτός από την παραδοσιακή γραφική διεπαφή με το χρήστη στην οποία ο χρήστης αλληλεπιδρά με την εφαρμογή μέσω clicks του ποντικιού , διαθέτουν

στους χρήστες τους και μία διεπαφή μέσω γραμμής εντολών με όλα τα πλεονεκτήματα που αυτή προσφέρει. Στην περίπτωση του rlgame αποφασίστηκε κάποιες από τις λειτουργίες του server να υλοποιηθούν μέσω γραμμής εντολών (με ενδεχόμενες μελλοντικές προεκτάσεις τις αντίστοιχες υλοποιήσεις σε μορφή γραφικής διεπαφής). Οι λειτουργίες που μέχρι στιγμής έχουν υλοποιηθεί στη γραμμή εντολών είναι η αποστολή και αποδοχή πρόσκλησης για παιχνίδι προς και από έναν παίκτη, και η εμφάνιση της λίστας των παικτών του server με αύξουσα σειρά ανάλογα με τη βαθμολογία τους στο παιχνίδι.

Για την υλοποίηση της γραμμής εντολών σε περιβάλλον web browser τα Primefaces Προσφέρουν την απλή λύση του component `<p:terminal>` . Ο client κώδικάς του που είναι πολύ απλός είναι ο εξής:

```
<p:terminalcommandHandler="#{chatviewbean.handleCommand}"

            height="300px"prompt="$"

            welcomeMessage="Welcome to rlgame terminal"

/>
```

Όπως βλέπουμε τα Primefaces κάνουν όλη τη δουλειά ως προς το εμφανισιακό κομμάτι της κονσόλας στα πλαίσια του browser και ο χρήστης ορίζει μόνο κάποιες παραμέτρους, όπως είναι το μήνυμα καλωσορίσματος, το ύψος της κονσόλας σε pixel και η μέθοδος του backing beanπου θα κάνει parse τις εντολές που δίνει ο χρήστης.

Στην περίπτωσή μας η μέθοδος αυτή είναι η `handleCommand()` που περιγράφεται παρακάτω.

```
public String handleCommand(String command, String[] params)

{

      if(command.equals("greet"))

            return"Hello " + params[0];



      elseif(command.equals("date"))
```

```
        return new Date().toString();

elseif (command.equals("invite"))

{

        if (users.contains(params[0]))

        {

                if(invitations.contains(new Invitations(username, params[0])))

                {

                        return"you have already invited" + params[0];

                }

                else

                {

                        Invitations inv = new Invitations(username,params[0]);

                        invitations.addInvitation(inv);

                        pushContext.push(CHANNEL + "*", username + " invites
                        "+params[0]);

                        return command+" succesful";

                }

        }

        else

        {

                return "there is no player with such name logged in at the moment";
```

```
        }
 }


elseif(command.equals("accept"))
 {
        RequestContext request = RequestContext.getCurrentInstance();
        if(invitations.contains(new Invitations(params[0], username)))
        {
                //ingame=true;
                this.setIngame(true);
                this.setIswhite(false);
                this.opponent= new String(params[0]);
                games.addNewGame(params[0], username);
                gameid=games.getgameid(username);
                request.update("form");
                pushContext.push(CHANNEL + "*", username + " accepts invitation of
                "+params[0]);
                pushContext.push(CHANNEL + params[0],username+" accepted your
                invitation!");
                invitations.removeInvitations(params[0], username);
                return username+" accepts invitation of "+ params[0];
        }
```

```
            else

                    return params[0]+" hasn't invited "+username;

        }

        elseif (command.equals("ranking"))

        {

                return users.rankings();

        }

        else

                return command + " not found";

}
```

Όπως παρατηρούμε η συνάρτηση αυτή λαμβάνει σαν δεδομένο την εντολή που έγραψε ο χρήστης μαζί με τις παραμέτρους που την ακολουθούν και εκτελεί τις ανάλογες ενέργειες και εμφανίζει την αντίστοιχη απάντηση στην κονσόλα του χρήστη. Η υλοποίηση επιπλέον εντολών από τον προγραμματιστή είναι κάτι παραπάνω από απλή. Στην Εικόνα 5.7 φαίνεται η εκτέλεση της εντολής ranking η οποία εμφανίζει όλους τους παίκτες με βαθμολογική σειρά.

Εικόνα 5.7: Η γραμμή εντολών του rlgameserver

5.4.3. Το tableau και η λογική του παιχνιδιού στον server

Θα περιγράψουμε εδώ τις μεθόδους που χρησιμοποιήθηκαν για να υλοποιηθεί η βασική λειτουργικότητα του serverπου είναι το παιχνίδι ανθρώπου εναντίον ανθρώπου. Κατ' αρχάς, από πλευράς σχεδιαστικής στον client του παίκτη το tableau δημιουργήθηκε με τη βοήθεια Primefaces components. Η κίνηση των πουλίων πραγματοποιείται με drag and drop, ενώ η απεικόνιση των πουλίων γίνεται με τη βοήθεια UTF-8 χαρακτήρων που απεικονίζουν τα πούλια. Τα χρώματα στο tableau δημιουργήθηκαν με τη βοήθεια της τεχνολογίας CSS3 που επιτρέπει την κάλυψη επιφανειών με ένα χρώμα σε διαβαθμισμένες αποχρώσεις (gradient). Ο κώδικας που υλοποιεί ένα απλό τετράγωνο του tableau στον client παρατίθεται ακολούθως:

```
<p:column>
        <p:panelid="sq47" styleClass="black">
        <p:outputLabel rendered="#{chatviewbean.board.board[4][7]==1}"
                id="wpawn47"value="&#9920;"styleClass="checker_black">
        </p:outputLabel>
        <p:outputLabel rendered="#{chatviewbean.board.board[4][7]==2}"
                id="bpawn47"value="&#9922;"styleClass="checker_black">
        </p:outputLabel>
        <p:draggable for="wpawn47"
                rendered="#{chatviewbean.board.board[4][7]==1 and chatviewbean.iswhite}"
                revert="true">
        </p:draggable>
        <p:draggable
                for="bpawn47"
                rendered="#{chatviewbean.board.board[4][7]==2 and !chatviewbean.iswhite}"
```

```
            revert="true"></p:draggable>

        <p:droppabletolerance="intersect">

        <p:ajax listener="#{chatviewbean.ondrop}"/>

        </p:droppable>

        </p:panel>

</p:column>
```

Παρατηρούμε πως το τα τετράγωνα του tableau αποτελούνται από components του τύπου <p:panel> εμφωλευμένα μέσα σε components <p:column>. Το attribute id χαρακτηρίζεται από τις συντεταγμένες του τετραγώνου στα πλαίσια της σκακιέρας (Το κάτω αριστερά τετράγωνο έχει συντεταγμένες 0, 0 και το πάνω δεξιά έχει συντεταγμένες 7, 7). Το attribute styleClass καθορίζει μία κλάση CSS3 η οποία είναι υπεύθυνη για τη μορφοποίηση του τετραγώνου με τα ζητούμενα χρώματα και διαβαθμίσεις όπως φαίνεται παρακάτω.

Ένα τετράγωνο μπορεί να περιέχει ένα μαύρο ή ένα άσπρο πούλι ή να είναι κενό. Αυτό επιτυγχάνεται με components του τύπου <p:outputLabel> εμφωλευμένα μέσα στο <p:panel> τα οποία εμφανίζονται ή όχι μέσω του attribute rendered κάτι που καθορίζεται με τη χρήση της expression language (el) ανάλογα με την τιμή μιας μεταβλητής που αντιστοιχεί στο συγκεκριμένο τετράγωνο και υπάγεται σε μία δομή η οποία αναπαριστά ολόκληρη τη θέση του παιχνιδιού και η οποία βρίσκεται στον server και συγκεκριμένα την κρατάει το backing beantης σελίδας.

Για κάθε ένα από τα δύο πιθανά πούλια που μπορεί να βρίσκονται επάνω στο τετράγωνο, χρησιμοποιούμε και ένα component του τύπου <p:draggable> το οποίο δίνει τη δυνατότητα στο χρήστη να σηκώσει και να σύρει το πούλι σε κάποιο άλλο τετράγωνο. Τα components αυτά εμφανίζονται (rendered) ανάλογα με το αν εμφανίζεται και το αντίστοιχο πούλι στο τετράγωνο.

Τέλος μέσα στο τετράγωνο εμφωλεύεται και ένα component του τύπου <p:droppable> το οποίο δίνει τη δυνατότητα στο χρήστη να σύρει στο συγκεκριμένο τετράγωνο ένα πούλι από κάποιο άλλο τετράγωνο. Όταν πραγματοποιείται μία τέτοια ενέργεια από τον παίκτη, τότε στέλνεται από τον client ένα ασύγχρονο μήνυμα μέσω AJAX στον serverπροκειμένου να γίνουν οι απαραίτητες ενέργειες και να ενημερωθεί και το tableau του αντίπαλου παίκτη. Αυτό γίνεται με

τη χρήση ενός component <p:ajax> εμφωλευμένου μέσα στο <p:droppable>. Όπως βλέπουμε εκείνη τη στιγμή στο backing beantης σελίδας τρέχει η μέθοδος onDrop η οποία παρατίθεται ακολούθως:

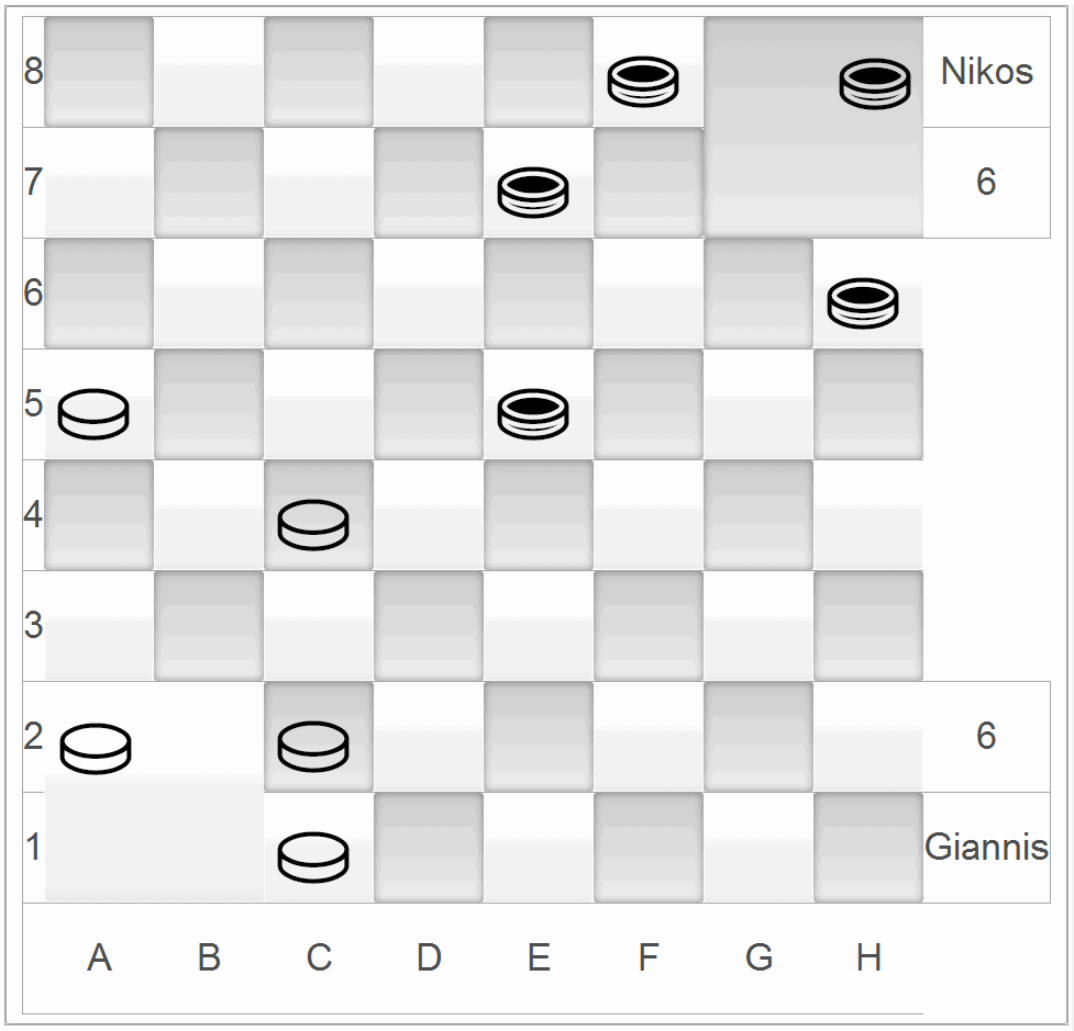
```
public void ondrop(DragDropEvent ddEvent)

{

	//Parse ddEvent and get source and destination square

	String draggedId = ddEvent.getDragId();

	Object data = ddEvent.getData();

	String from = ddEvent.getDragId();

	String to = ddEvent.getDropId();

	from = from.substring(11);

	to= to.substring(8);

	int fromx = Integer.parseInt(from.substring(0,1));

	int fromy = Integer.parseInt(from.substring(1));

	int tox = Integer.parseInt(to.substring(0,1));

	int toy = Integer.parseInt(to.substring(1));

	//Create a new Move object from source and destination square

	Move move = new Move();

	move.setFromX(fromx);

	move.setFromY(fromy);

	move.setToX(tox);
```

```
move.setToY(toy);



//apply the move to the game

games.getPlayedgames().get(gameid).getBoard().applyMove(move);

//remove the game and update players ratings if it is over

int  over = 0;

if(this.getBoard().isGameover())

{

        users.updatePlayerRating(username, opponent);

        if (iswhite)

        {

                this.games.removeGame(username, opponent);

        }

        else

        {

                this.games.removeGame(opponent, username);

        }

        over = 1;

pushContext.push(CHANNEL + "*", username +" won a match against " + opponent+"!");



}
```

```
        //Push changes to clients

        pushContext.push(CHANNEL2 + opponent, over+","+username );

        pushContext.push(CHANNEL2 + username, over+","+username );

}
```

Όπως βλέπουμε η μέθοδος δέχεται ως όρισμα από τον client ένα αντικείμενο τύπου DragDropEvent, το οποίο έχουν δημιουργήσει τα Primefaces και απ' αυτό εξάγονται όλες οι απαραίτητες πληροφορίες, όπως είναι το τετράγωνο απ' το οποίο έσυρε το πούλι ο παίκτης και το τετράγωνο στο οποίο το εναπόθεσε. Απ' το αντικείμενο αυτό η μέθοδος κάνει parsing των δεδομένων αυτών και δημιουργεί ένα αντικείμενο του τύπου Move, το οποίο δίνεται ως όρισμα στην μέθοδο `applyMove()`, η οποία ανήκει στην κλάση Rlboard που κρατάει όλη την εικόνα του tableau μία δεδομένη στιγμή του παιχνιδιού. Η `applyMove()` πραγματοποιεί όλες τις αλλαγές που έγιναν με την κίνηση που έκανε ο παίκτης και ενημερώνει την εικόνα του tableau. Κατόπιν, με χρήση της τεχνολογίας primePush όπως είδαμε στα προηγούμενα, οι αλλαγές στέλνονται στους 2 clients προκειμένου να ενημερωθούν. Αν από την κίνηση προέκυψε νικητής, τότε οι αντίπαλοι ενημερώνονται για το τέλος του παιχνιδιού, και το παιχνίδι τους αφαιρείται από τη λίστα των παιχνιδιών που τηρούνται στον server, ενώ ταυτόχρονα ενημερώνονται και οι βαθμολογίες τους.

Η μορφή του tableauστα πλαίσια ενός παιχνιδιού φαίνεται στην Εικόνα 5.8.



Εικόνα 5.8: Το tableau του rlgameserver

5.5 Μοντελοποίηση της λογικής του παιχνιδιού στην πλευρά του εξυπηρετητή

Ένας από τους σκοπούς του server που αναπτύχθηκε στην παρούσα διπλωματική, είναι να είναι κατά το δυνατόν «χτισμένος» με τη λογική των αυτόνομων τμημάτων (modularity) ούτως ώστε τμήματά του να μπορούν να χρησιμοποιηθούν σε μελλοντικές προσπάθειες είτε για ανάπτυξη κάποιου αλγορίθμου τεχνητής νοημοσύνης, είτε για χρησιμοποίηση του περιβάλλοντος διεπαφής χρήστη στην πλευρά του web browser σε ενδεχόμενες επεκτάσεις.

Για το σκοπό αυτό, η λογική του παιχνιδιού και η μοντελοποίηση του σε δομές δεδομένων, έγινε σε μία ξεχωριστή κλάση java η οποία εδράζεται στην πλευρά του server. Συγκεκριμένα η κλάση

RLboard κρατάει όλα τα στοιχεία μίας θέσης του tableau για ένα στιγμιότυπο του παιχνιδιού. Τα στοιχεία αυτά είναι οι θέσεις των πουλίων στο tableau, τα εναπομείναντα πούλια στη βάση του κάθε παίκτη, ο παίκτης που είναι η σειρά του να παίξει, εάν το παιχνίδι έχει κερδηθεί από κάποιον παίκτη κ.λ.π. όπως φαίνεται παρακάτω:

```
public static final int EMPTY = 0;

public static final int WHITE = 1;

publicstaticfinalintBLACK = 2;

public static final int OPPONENT_INVADED = -1;

public static final int FINISHED = 2;

public static final int LEGAL = 0;

public static final int ILLEGAL = 1;

public int[][] board = newint[8][8];

private int whitebase;

private int blackbase;

private int whitecheckerscaptured;

private int blackcheckerscaptured;

private int turn;

private Boolean gameover;

privateintwinner;
```

Η μέθοδος `initBoard()`όπως υποδηλώνει το όνομά της, αρχικοποιεί τις τιμές έτσι όπως είναι στο ξεκίνημα του παιχνιδιού μεταξύ δύο παικτών

```
publicvoid initBoard()

{

	this.whitebase=10;

	this.blackbase=10;

	this.turn=WHITE;

	this.blackcheckerscaptured=0;

	this.whitecheckerscaptured=0;

	gameover=false;

	for (int x=0; x<8; x++)

	{

		for(int y=0; y<8; y++)

		{

			/*if(!((x<2 && y<2) || (x>5 && y>5 )))*/

			{ board[x][y]=EMPTY;}

			}

	}

}
```

Η κλάση συνοδεύεται από διάφορες μεθόδους που εκτελούν τις διάφορες λειτουργίες του παιχνιδιού. Για παράδειγμα υπάρχει η μέθοδος `applymove()` η οποία οδηγεί από ένα στιγμιότυπο σε ένα άλλο, λαμβάνοντας ως όρισμα μία κίνηση, δηλαδή τις συντεταγμένες ενός τετραγώνου αφετηρίας και ενός τετραγώνου προορισμού. Όταν ο παίκτης στον client του κάνει

κάποια κίνηση με drag and drop, η κίνηση αποστέλλεται στον server, και ο server αναλαμβάνει να τρέξει την `applyMove()`, για το στιγμιότυπο (instance) της κλάσης που απεικονίζει το tableau του συγκεκριμένου παιχνιδιού που παίζει ο παίκτης. Μόλις το tableau ενημερωθεί, οι αλλαγές μεταφέρονται στον client του αντιπάλου με τη διαδικασία του PrimePush που είδαμε.

Η κλάση RLBoard μπορεί να χρησιμοποιηθεί και για διάφορες άλλες εργασίες σχετικές με το παιχνίδι, όπως για παράδειγμα η δημιουργία ενός bot. Προς το παρόν η υλοποίηση της αφορά τη μορφή του παιχνιδιού, όπου η σκακιέρα είναι διαστάσεων 8x8 και οι βάσεις των αντιπάλων είναι διαστάσεων 2x2. Προφανώς μία μελλοντική υλοποίηση μπορεί να δέχεται αυτά τα μεγέθη με παραμετρική μορφή, και να έχει έτσι τη δυνατότητα να χρησιμοποιηθεί για τις ανάγκες όλων των δυνατών παραλλαγών του παιχνιδιού.

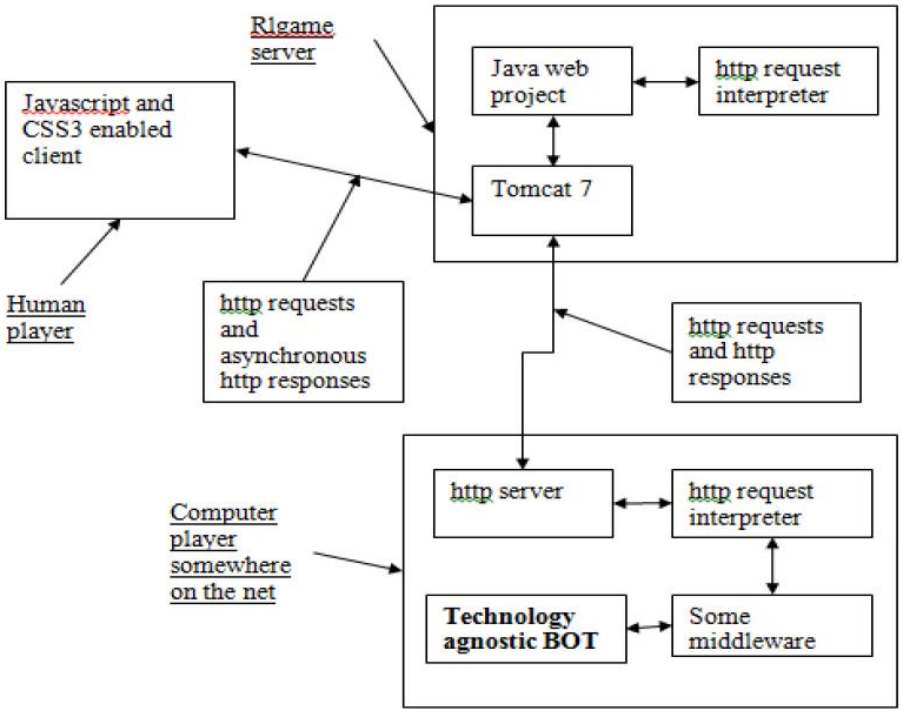
5.6 Υποδομή για παιχνίδι ανθρώπου εναντίον υπολογιστή

Ο rlgameserver πέραν όλων των άλλων έχει σκοπό να διευκολύνει και τους επίδοξους ερευνητές τεχνητής νοημοσύνης στο να τεστάρουν τα προγράμματά τους εναντίον ανθρωπίνων αντιπάλων. Για να επιτευχθεί αυτό, χρειάζεται μία υποδομή που θα διευκολύνει κάποιον που έχει κατασκευάσει ένα bot, να το θέσει σε λειτουργία στα πλαίσια του rlgameserver και να μπορούν έτσι οι χρήστες που μπαίνουν στον server, να ανταγωνιστούν εναντίον του και να ελέγχονται μέσω του βαθμολογικού συστήματος οι δυνατότητές και των δύο.

Υλοποιήσαμε μία κατανεμημένη υποδομή, βάσει της οποίας, ένα bot θα μπορεί να βρίσκεται οπουδήποτε στο διαδίκτυο και να ανταλλάσσει πληροφορίες με τον rlgameserver μέσω HTTPrequests (GET ή POST).

Τα bots θα πρέπει να συνεργάζονται (ή να παρέχουν τα ίδια αυτή τη λειτουργικότητα) me έναν HTTP server ο οποίος ανταλλάσσει αιτήματα με τον rlgamserver (για παράδειγμα όταν ένας παίκτης κάνει μία κίνηση, ο rlgameserver στέλνει ένα HTTP αίτημα με την εικόνα του tableau εκείνη τη στιγμή, και τον παίκτη που πρέπει να κάνει την κίνηση κωδικοποιημένα σαν GET ή POST παράμετρους. Το αίτημα αυτό παραλαμβάνει ο HTTP server του bot και επικοινωνεί με το τμήμα τεχνητής νοημοσύνης του προγράμματος, το οποίο παράγει μία κίνηση για τη θέση αυτή και ο server αναλαμβάνει να την στείλει πίσω στον rlgameserver). Ιδανικά, θα θέλαμε οι προγραμματιστές να είναι σε θέση να αναπτύξουν τα bots τους σε όποια γλώσσα και τεχνολογία

αυτοί προτιμούν. Είναι όμως απαραίτητη μία κοινή γλώσσα επικοινωνίας με τον rlgameserver. Για να τυποποιήσουμε αυτή την επικοινωνία χρειάζεται μία μέθοδος, η οποία θα παραλαμβάνει τα HTTP Αιτήματα και θα τα κάνει parse. Από εκεί και πέρα το μόνο που θα έχει να κάνει ο προγραμματιστής, θα είναι να κατασκευάσει κάποιον ενδιάμεσο (middleware) κώδικα ο οποίος προσαρμόζει τα inputs και outputs του bot του με τα inputs και τα outputs που χρειάζεται να παραλάβει ο rlgameserver μέσω των HTTP requests (δηλαδή να δίνονται κοινά ονόματα παραμέτρων και οι αντίστοιχες τιμές τους). Η αρχιτεκτονική της κατανεμημένης υποδομής επικοινωνίας που υλοποιήσαμε, δίνεται στην Εικόνα 5.9.



Εικόνα 5.9: Κατανεμημένη αρχιτεκτονική για παιχνίδι ανθρώπου εναντίον υπολογιστή (Δίκαρος, Καλλές 2014).

Για τις ανάγκες της παρούσας διατριβής, και για να εξασφαλίζεται η όσο το δυνατόν απλούστερη δομή της αρχιτεκτονικής χρησιμοποιήσαμε έναν απλό http serverπου κατασκευάστηκε με τη χρήση των δυνατοτήτων της java. Αυτός αναλαμβάνει το ρόλο της επικοινωνίας μέσω GET requestsμε τον rlgameserver, δεχόμενος ένα στιγμιότυπο του tableau, κωδικοποιημένο σε ένα request. Κατόπιν, ο server μεταβιβάζει το request σε μία μέθοδο (middleware) η οποία το αποκωδικοποιεί, και δημιουργεί απ' αυτό ένα στιγμιότυπο της κλάσης RLBoard. Το instance

αυτό με χρήση κατάλληλων μεθόδων (που υλοποιούν αλγορίθμους τεχνητής νοημοσύνης), μπορεί να επιστρέψει μία προτεινόμενη κίνηση, η οποία μέσω middleware, κωδικοποιείται σε http απάντηση και αποστέλλεται με τον απλό Java server που δημιουργήσαμε, πίσω στον rlgame server. Εκεί μία άλλη μέθοδος την παραλαμβάνει, την αποκωδικοποιεί, και τη χρησιμοποιεί στα πλαίσια της κλάσης RLBoard για να ενημερώσει την κατάσταση του παιχνιδιού και να στείλει την απάντηση στον client του χρήστη.

5.7 Προτεινόμενες επεκτάσεις ως προς τις δυνατότητες του rlgameserver.

Υπάρχουν πολλές δυνατότητες που μπορούν να προστεθούν στον rlgameserver, ούτως ώστε να τον καταστήσουν εφάμιλλο με τους εμπορικούς δικτυακούς εξυπηρετητές στους οποίους ο κόσμος μπορεί να παίζει παιχνίδια στρατηγικής (τάβλι, σκάκι κ.λ.π). Ενδεικτικά παραθέτουμε τις ακόλουθες, ως κατευθύνσεις σε όσους αποφασίσουν να επεκτείνουν την προσπάθεια αυτή.

- Υλοποίηση περισσότερων command line δυνατοτήτων.

Επί της παρούσης έχουν υλοποιηθεί κάποιες δυνατότητες μέσω του command line interface, όπως είναι η αποστολή και αποδοχή πρόσκλησης και η εμφάνιση των βαθμολογιών των παικτών. Επιπλέον δυνατότητες που μπορούν να υλοποιηθούν, είναι για παράδειγμα η εμφάνιση του ιστορικού των παιχνιδιών του χρήστη, του ιστορικού των αντιπάλων του παίκτη, η εμφάνιση των στοιχείων κάποιου αντιπάλου και η εκτέλεση ερωτημάτων (queries) για θέματα όπως το ποιοι παίκτες με παρόμοια βαθμολογία βρίσκονται αυτή τη στιγμή συνδεδεμένοι κ.λ.π.

- Υλοποίηση επιπλέον δυνατοτήτων μέσω του web interface

Οι προαναφερθείσες δυνατότητες μπορούν να υλοποιηθούν πέραν της κονσόλας, και μέσω γραφικού περιβάλλοντος. Επίσης το γραφικό περιβάλλον μπορεί να εμπλουτιστεί με γραφήματα ή πίνακες που αφορούν την πορεία των παικτών στο παιχνίδι. Τα Primefaces προσφέρουν components με τη χρήση των οποίων μπορούν να ενταχθούν γραφήματα τα στοιχεία των οποίων θα συλλέγονται από τη βάση δεδομένων του rlgameserver, στο web browser περιβάλλον.

- Δυνατότητα για σταμάτημα ενός παιχνιδιού στη μέση και συνέχισή του (adjourn) σε κάποια άλλη μελλοντική στιγμή.

- Δυνατότητα για χρονικό περιορισμό στα παιχνίδια

Οι παίκτες θα μπορούν να θέσουν ένα προκαθορισμένο χρονικό περιθώριο μέσα στο οποίο θα πρέπει να έχουν εκτελέσει τις κινήσεις τους αλλιώς θα χάνουν λόγω χρόνου.

- Δυνατότητα για αλλαγή της εμφάνισης του tableau και των χρωμάτων, ανάλογα με τις προτιμήσεις του παίκτη.

5.8 Έλεγχος και αποσφαλμάτωση του εξυπηρετητή

Είναι προφανές ότι ο rlgameserver, όπως και όλες οι εφαρμογές λογισμικού, χρειάζονται εκτενή έλεγχο προκειμένου να εντοπιστούν και να διορθωθούν οι δυσλειτουργίες (bugs). Από τη φύση της, μία εφαρμογή παιχνιδιού, έχει ανάγκη από παικτικό κοινό για να τη δοκιμάσει, προτού φτάσει σε μια ώριμη κατάσταση παραγωγικής λειτουργίας. Για το σκοπό αυτό, μία δοκιμαστική έκδοση του rlgameserver έχει ανέβει σε εικονικό εξυπηρετητή (virtual machine) που εδράζεται στο cloud service OKEANOS. Οι χρήστες μπορούν να εισέρχονται ελεύθερα εισάγοντας ένα ψευδώνυμο (nickname) και να παίξουν δοκιμαστικά παιχνίδια με φίλους τους.

Η δικτυακή διεύθυνση στην οποία μπορούν να συνδεθούν οι χρήστες είναι η:

http://83.212.105.28:8080/rlgameserver/

Στο προσεχές διάστημα, αναμένεται να ενσωματωθεί και μία φόρμα αναφοράς σφαλμάτων και παρατηρήσεων που θα αφορούν σε βελτιώσεις είτε λειτουργικές είτε αισθητικές, προκειμένου να γίνει πιο συστηματική η διαδικασία.

Κεφάλαιο 6
Επίλογος

Η διατριβή εστίασε σε μεγαλύτερο βαθμό στην διαδικασία ανάπτυξης ενός διαδικτυακού εξυπηρετητή παιχνιδιών με ταυτόχρονη μελέτη σχετικών τεχνολογιών, και σε μικρότερο βαθμό στις δυνατότητες που υπάρχουν για την αξιοποίησή του στον τομέα της τεχνητής νοημοσύνης.

Ως προς το πρώτο σκέλος, διερευνήθηκαν διεξοδικώς οι δυνατότητες που παρέχονται από διάφορες δικτυακές (και όχι μόνο) τεχνολογίες και αποκτήθηκε μία σημαντική τεχνογνωσία ως προς την αρχιτεκτονική ανάπτυξης διαδραστικών και ασύγχρονων δικτυακών εφαρμογών. Η διερεύνηση αυτή έγινε λαμβάνοντας υπόψη το γεγονός, ότι έχει πρόσφατα γίνει αντίστοιχη απόπειρα ανάπτυξης δικτυακού εξυπηρετητή για το RLGame με την χρήση των τεχνολογιών SCALA και LIFT. Προσπαθήσαμε να εντοπίσουμε ποιες είναι οι σχετικές τεχνολογίες που εμφανίζουν δυναμική και προοπτική ως προς τη χρήση τους, οι οποίες και χρησιμοποιήθηκαν στην παρούσα αναπτυξιακή προσέγγιση.

Τα PrimeFaces ήταν το κυρίως web framework που αξιοποιήθηκε για την ανάπτυξη του εξυπηρετητή και οι εντυπώσεις που αποκομίστηκαν είναι θετικές. Θετικές αφενός γιατί είναι πολύ εύκολο να ενταχθεί στα πλαίσια ενός Java web project με ελάχιστες ρυθμίσεις, και αφετέρου γιατί προσφέρει πάρα πολλές δυνατότητες μέσω των components του, για τις οποίες οι προγραμματιστές δικτυακών εφαρμογών είναι συχνά αναγκασμένοι να χάνουν αρκετό χρόνο, είτε ανακαλύπτοντας βιβλιοθήκες που παρέχουν τις αντίστοιχες δυνατότητες, είτε υλοποιώντας τες μόνοι τους. Επίσης τα components που παρέχονται από τη βιβλιοθήκη αποδείχτηκαν ευχάριστα οπτικά (και σε αυτό συνεισφέρει και το γεγονός ότι υπάρχει ομοιομορφία μεταξύ τους), ενώ η εμφάνισή τους διαθέτει αρκετές δυνατότητες παραμετροποίησης μέσω του theming framework που παρέχεται.

Τα μειονεκτήματα της βιβλιοθήκης εστιάζονται κυρίως σε δυσλειτουργίες(bugs) που οφείλονται στο γεγονός ότι είναι σχετικά καινούρια. Οι δημιουργοί της εν τούτοις, συνεχίζουν να δίνουν στη δημοσιότητα νέες εκδόσεις, όπου πολλά από τα προβλήματα που υπάρχουν επιλύονται. Χαρακτηριστικό παράδειγμα είναι η λειτουργία του PrimePush (στην οποία βασίστηκε η ασύγχρονη αποστολή μηνυμάτων από τον serverπρος τους clients) η οποία χρειαζόταν έκδοση του Apache Tomcat 7.0.35 ή μεταγενέστερη για να λειτουργεί σωστά, καθώς η λειτουργικότητα αυτή υλοποιήθηκε πολύ πρόσφατα.

Είναι κοινός τόπος πλέον ότι η HTML5 αποτελεί το μέλλον του παγκόσμιου ιστού, και θα θέλαμε να δούμε, πως οι δυνατότητες που παρέχει μπορούν να διευκολύνουν, να εμπλουτίσουν και να βελτιώσουν τον rlgameserver. Επειδή τα PrimeFaces υποστηρίζουν τη συγγραφή HTML5, θα μπορούσαμε να πούμε ότι υπήρξαν μία καλή επιλογή και γι αυτόν τον επιπλέον λόγο.

Ως προς το σκέλος της έρευνας πάνω σε θέματα τεχνητής νοημοσύνης, υπήρξε ένας προβληματισμός για το ποιος θα είναι ο πιο εύκολος τρόπος να ενταχθούν ευφυείς πράκτορες (bots) στα πλαίσια του εξυπηρετητή, και η πρόταση που δόθηκε είναι η ανταλλαγή HTTP αιτημάτων μέσω του διαδικτύου μεταξύ αυτού και των bots. Η πρόταση διατυπώθηκε με γνώμονα την μέγιστη δυνατή απλότητα της κατανεμημένης αρχιτεκτονικής, δεδομένου ότι είναι αρκετά πιο απλό να ανταλλάσσεται ένα αίτημα με HTTP πρωτόκολλο ανάμεσα σε εξυπηρετητή και bot, παρά να δημιουργηθεί υποδομή για να εντάσσεται ολόκληρο το bot (το οποίο μπορεί να έχει δημιουργηθεί με τη χρήση διαφόρων τεχνολογιών) μέσα στον εξυπηρετητή. Επίσης προτιμήθηκε η απλή μορφή http αιτημάτων και απαντήσεων (μέσω GET ή POST) αντί για μια αντίστοιχη υλοποίηση μέσω τεχνολογιών web services.

Στο παρελθόν έχουν γίνει απόπειρες ανάπτυξης bots με τη χρήση νευρωνικών δικτύων, ενισχυτικής μάθησης και Minimax. Στα πλαίσια της επέκτασης της λειτουργίας του rlgameserver και της παροχής διευκολύνσεων προς τους ερευνητές, προτείνεται η δημιουργία ενός νέου απλού bot, το οποίο αξιοποιεί την υποδομή μοντελοποίησης που δημιουργήθηκε για τις ανάγκες υλοποίησης του παιχνιδιού ' άνθρωπος εναντίον ανθρώπου'. Το νέο αυτό bot θα χρησιμεύσει και σαν απόδειξη της εφικτότητας (proof of concept) για την διαδικασία της διαδικτυακής συνεργασίας μεταξύ ενός bot και του rlgameserver. Σαν κατάλληλη μεθοδολογία προτείνεται ο αλγόριθμος Negamax με alpha beta pruningκαι πίνακες μεταφοράς (transposition tables) καθώς και μιας συνάρτησης εκτίμησης (evaluation function) η οποία χρησιμοποιεί απλά ευριστικά (heuristics) του παιχνιδιού. Μετά την ανάπτυξη αυτών, θα είναι πολύ εύκολο για κάποιον προγραμματιστή να χρησιμοποιήσει τις δομές δεδομένων που αναπτύχθηκαν για να εφαρμόσει τους δικούς του αλγορίθμους, ή και να επαυξήσει τις δυνατότητες του υπάρχοντος αλγορίθμου (π.χ. με επιπλέον ευριστικά) ενώ θα είναι πολύ απλή πλέον η συνεργασία του bot που θα αναπτύξει με τον rlgameserver, χωρίς να χρειάζονται πολύπλοκοι ενδιάμεσοι κώδικες (middleware) προσαρμογής των εισόδων και των εξόδων (inputs και outputs).

Βιβλιογραφία

[01] Bauer, C. King, G. "Hibernate in Action: Practical Object/Relational Mapping". Manning, (2004).

[02] Bauer, C. King, G. "Java Persistence with Hibernate". Manning, (2007).

[03] Brittain, J. Darwin, I. "Tomcat: The Definitive Guide". O' Reilly, (2007).

[04] Çivici, Ç . "PrimeFaces User's Guide 3.5". (2013)

[05] Dikaros, N. Kalles, D. "Developing a game server for humans and bots". 8th Hellenic Conference on Artificial Intelligence, Ioannina, Greece,(2014).

[06] DuBois, P. "MySQL (5th Edidion) (Developer's Library)". Addison -Wesley (2008).

[07] Geary, D. Horstmann, C. "Core JavaServer Faces (Sun Core Series)". Prentice-Hall, (2010).

[08] Kalles, D., Kanellopoulos, P. On Verifying Game Design and Playing Strategies using Reinforcement Learning. ACM Symposium on Applied Computing, special track on Artificial Intelligence and Computation Logic, Las Vegas (2001).

[09] Kalles, D., Ntoutsi, E."Interactive Verification of Game Design and Playing Strategies". IEEE International Conference on Tools with Artificial Intelligence, Washington D.C. (2002).

[10] Kalles, D. Measuring Expert Impact on Learning how to Play a Board Game. In 4th IFIP Conference on Artificial Intelligence Applications and Innovations, Athens, Greece (2007).

[11] Kalles, D. "Player co-modelling in a strategy board game: discovering how to play fast". Cybernetics and Systems 39(1), (2008).

[12] Kalles, D., Kanellopoulos, P. "A Minimax Tutor for Learning to Play a Board Game". In 18th European Conference on Artificial Intelligence, workshop on Artificial Intelligence in Games, Patras, Greece, (2008).

[13] Katz, M. Shaikovsky, I . "Practical RichFaces 2nd Edition". Apress,(2011) .

[14] Sawyer McFarland, D. "JavaScript & jQuery: The Missing Manual". O' Reilly, (2011).

[15] Varaksin, O. Caliskan, M. "PrimeFaces CookBook". PACKT Publishing, (2013).

[16] Βλάση, Α. "Συστήματα για παιχνίδια στρατηγικής". Διπλωματική εργασία. Ελληνικό Ανοικτό Πανεπιστήμιο, Πάτρα (2008).

[17] Δήμας, Γ. "Ανάπτυξη και λειτουργία Δικτυακού Εξυπηρετητή παιχνιδιών". Διπλωματική εργασία. Ελληνικό Ανοικτό Πανεπιστήμιο, Πάτρα (2011).

[18] Φύκουρας, Η. "Συστήματα για παιχνίδια στρατηγικής: Ανάπτυξη και λειτουργία Δικτυακής Υπηρεσίας Διεξαγωγής Παιχνιδιών". Διπλωματική εργασία. Ελληνικό Ανοικτό Πανεπιστήμιο, Πάτρα (2008).

[19] Wikipedia contributors. "Ajax (programming)". http://en..wikipedia.org/wiki/Ajax_(programming). Wikipedia The Free Encyclopedia (Accessed May 2014).

[20] Wikipedia contributors. "Apache Tomcat". http://en.wikipedia.org/wiki/Apache_Tomcat. Wikipedia The Free Encyclopedia (Accessed May 2014).

[21] Wikipedia contributors. "Comet (programming)". http://en.wikipedia.org/wiki/Comet_(programming). Wikipedia The Free Encyclopedia (Accessed May 2014).

[22] Wikipedia contributors. "Elo rating system". http://en.wikipedia.org/wiki/Elo_rating_system. Wikipedia The Free Encyclopedia (Accessed May 2014).

[23] Wikipedia contributors. "Glicko rating system". http://en.wikipedia.org/wiki/Glicko_rating_system. Wikipedia The Free Encyclopedia (Accessed May 2014).

[24] Wikipedia contributors. "Hibernate Java". http://en.wikipedia.org/wiki/Hibernate_%28Java%29. Wikipedia The Free Encyclopedia (Accessed May 2014).

[25] Wikipedia contributors. "Java Platform, Enterprise Edition". http://en.wikipedia.org/wiki/Java_EE. Wikipedia The Free Encyclopedia (Accessed May 2014).

[26] Wikipedia contributors. "JavaScript". http://en.wikipedia.org/wiki/JavaScript. Wikipedia The Free Encyclopedia (Accessed May 2014).

[27] Wikipedia contributors. "jQuery". http://en.wikipedia.org/wiki/Jquery. Wikipedia The Free Encyclopedia (Accessed May 2014).

[28] Wikipedia contributors. "MySQL". http://en.wikipedia.org/wiki/MySQL. Wikipedia The Free Encyclopedia (Accessed May 2014).

[29] Wikipedia contributors. "Negamax". http://en.wikipedia.org/wiki/MySQL. Wikipedia The Free Encyclopedia (Accessed May 2014).

[30] Wikipedia contributors. "Server side scripting". http://en.wikipedia.org/wiki/Server-side_scripting. Wikipedia The Free Encyclopedia (Accessed May 2014).

[31] PrimeTek. "PrimeFaces Ultimate JSF Framework". http://www.primefaces.org/ (Accessed May 2014).

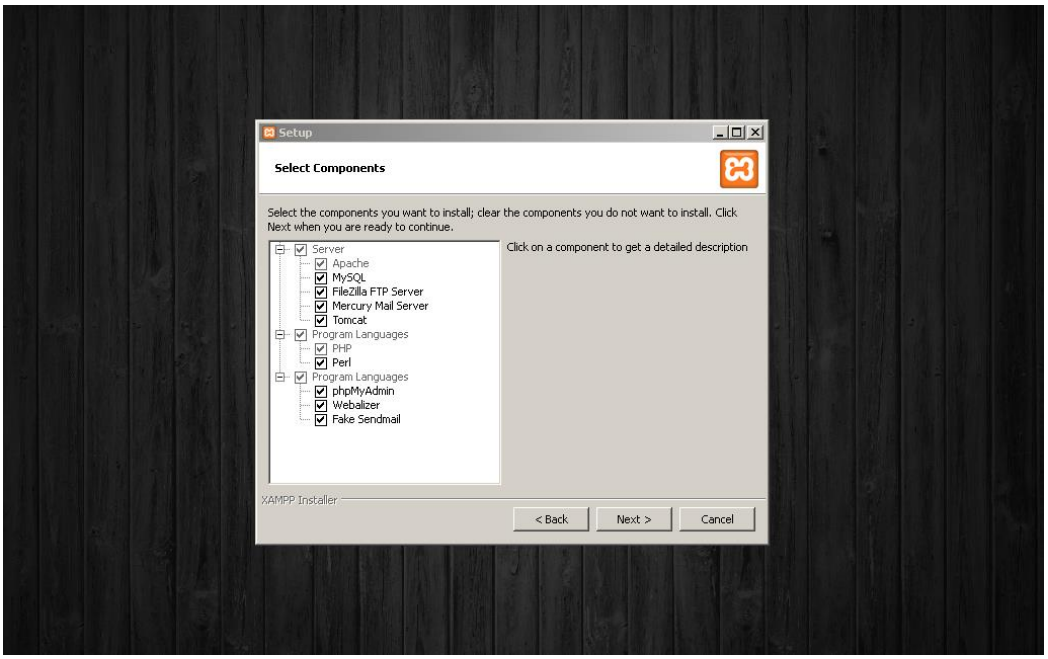
Παράρτημα Α
Οδηγίες χρήσης για τη λειτουργία του rlgameserver

Για τη λειτουργία του rlgameserver χρειάζονται μία βάση δεδομένων MySQL και ένας Apache Tomcat Server έκδοσης 7.0.35 ή μεταγενέστερης. Στα πλαίσια της εργασίας ο κώδικας που δημιουργήθηκε έγινε με τη βοήθεια του περιβάλλοντος IDE Eclipse. Επίσης για τη διαχείριση της MySQL χρησιμοποιήθηκε το γραφικό διαχειριστικό περιβάλλον phpMyAdmin το οποίο λειτουργεί σε περιβάλλον web browser και η λειτουργία του απαιτεί την ύπαρξη εγκατεστημένων εκδόσεων της γλώσσας PHP και του Apache http server.

Α.1 Εγκατάσταση σε περιβάλλον windows

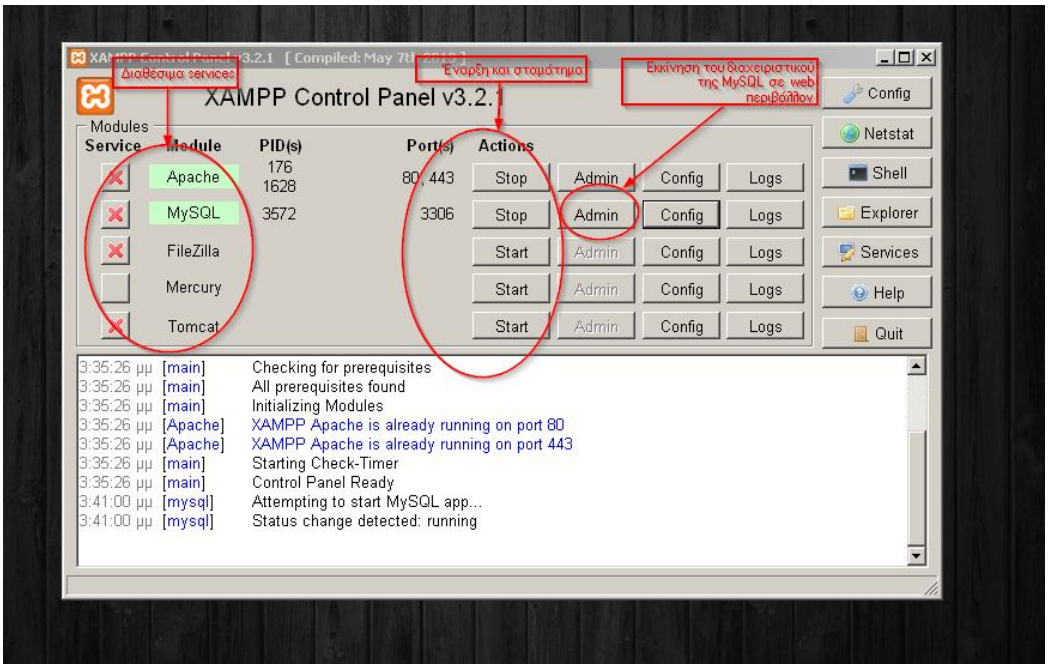
Μία πολύ καλή λύση για να εγκατασταθούν όλα τα παραπάνω με τις ελάχιστες δυνατές ρυθμίσεις και παραμετροποιήσεις είναι η εγκατάσταση του λογισμικού XAMPP. Το λογισμικό διατίθεται ελεύθερα και μπορεί να κατέβει από τη σελίδα

https://www.apachefriends.org/download.html



Εικόνα Α-1: εγκατάσταση του XAMPP

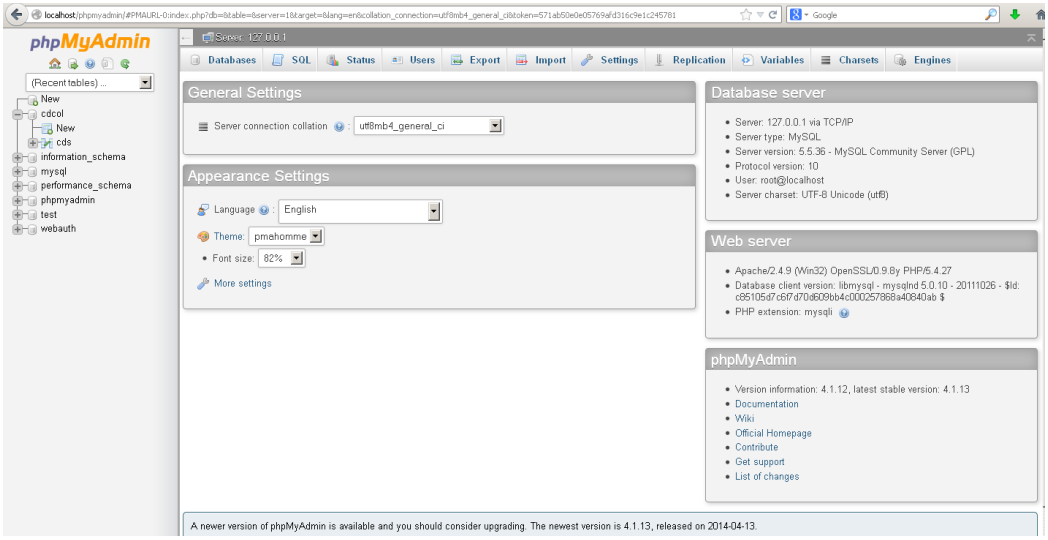
Όπως βλέπουμε στην Εικόνα Α-1 εγκαθιστώντας το XAMPPεπιλέγουμε να εγκαταστήσουμε Apache, MySQL, Tomcat και PHP. Όταν ολοκληρωθεί η εγκατάσταση έχουμε πρόσβαση στο διαχειριστικό περιβάλλον της εφαρμογής το οποίο φαίνεται ακολούθως στην Εικόνα Α-2.



Εικόνα Α-2: Διαχειριστικό περιβάλλον του XAMPP

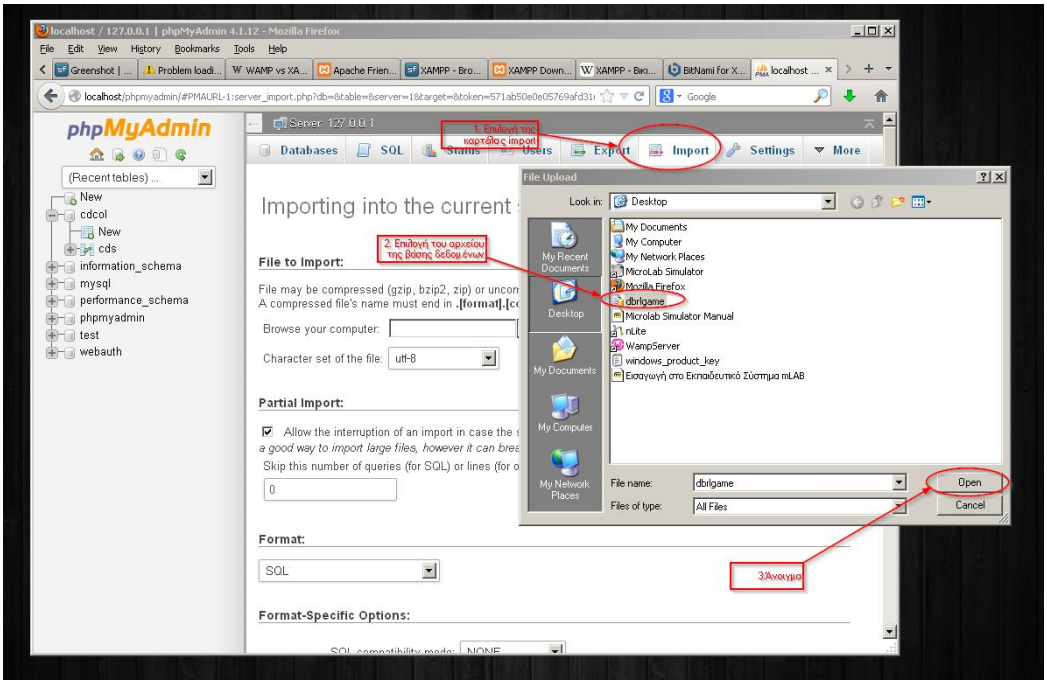
Όπως βλέπουμε στο διαχειριστικό περιβάλλον του XAMPP μπορούμε να εκκινήσουμε και να σταματήσουμε κατά βούληση τα διαθέσιμα services και τα αντίστοιχα διαχειριστικά τους. Επίσης, έχουμε εποπτεία στα ports που τρέχουν τα services, στα αρχεία ρυθμίσεων και τα logs τους.

Ανοίγοντας το διαχειριστικό περιβάλλον της MySQL από το αντίστοιχο κουμπί βλέπουμε την παρακάτω οθόνη της Εικόνας Α-3 στον browser.



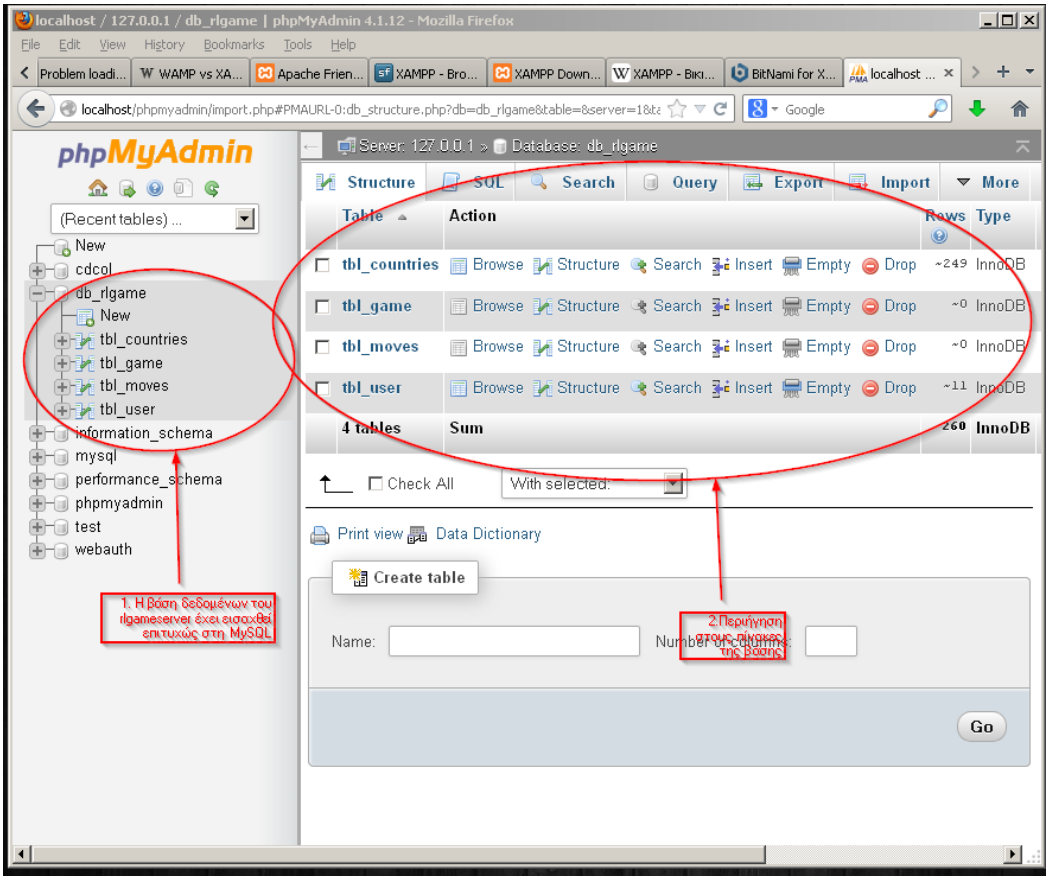
Εικόνα Α-3: Διαχειριστικό περιβάλλον phpMyAdmin

Στο phpMyAdmin μπορούμε πολύ εύκολα να κάνουμε importτη βάση δεδομένων του rlgameserver που έχει παραδοθεί με τη μορφή SQL αρχείου (και γενικότερα να εκτελέσουμε οποιαδήποτε διαχειριστική λειτουργία αφορά τον MySQLserver). Η διαδικασία εισαγωγής της βάσης φαίνεται στην Εικόνα Α-4.



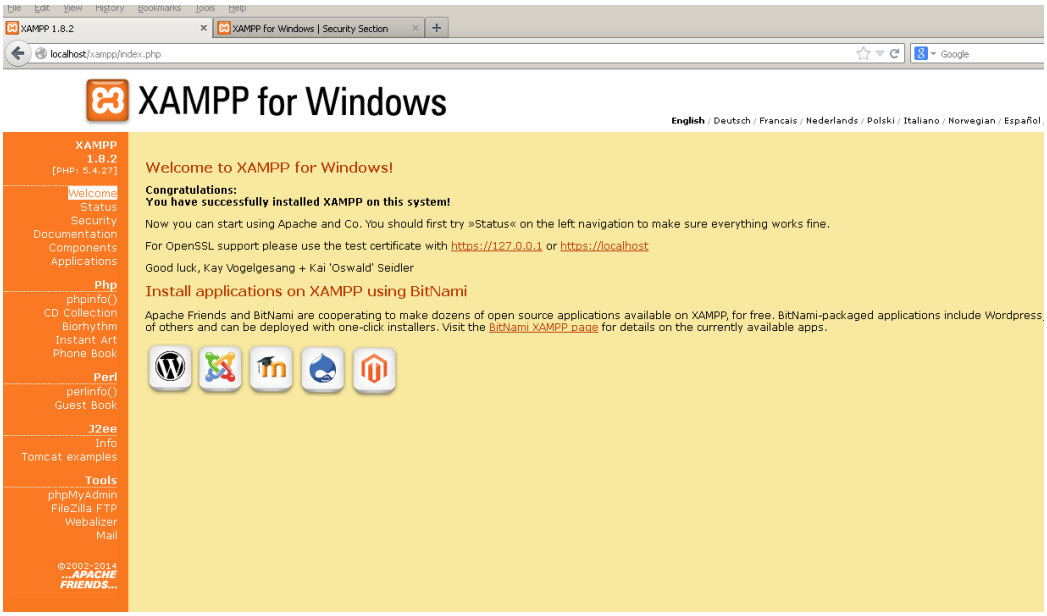
Εικόνα Α-4: Εισαγωγή βάσης δεδομένων από αρχείο στο phpMyAdmin

Αν όλα έχουν πάει καλά μπορούμε να δούμε τη βάση δεδομένων του rlgameserver και να περιηγηθούμε στους πίνακές της όπως φαίνεται παρακάτω. Επειδή το XAMPP έγκαθιστά την MySQL χωρίς κωδικό πρόσβασης, είναι καλύτερα να δημιουργήσουμε έναν κωδικό, τον οποίο θα χρησιμοποιήσουμε και στο connection String μέσω του οποίου θα αποκτήσει πρόσβαση ο rlgameserver στη βάση δεδομένων. Για να θέσουμε έναν κωδικό ανοίγουμε τον browser και βάζουμε τη διεύθυνση localhost και εμφανίζεται άλλο ένα διαχειριστικό εργαλείο του XAMPP όπως φαίνεται στην Εικόνα Α-5.



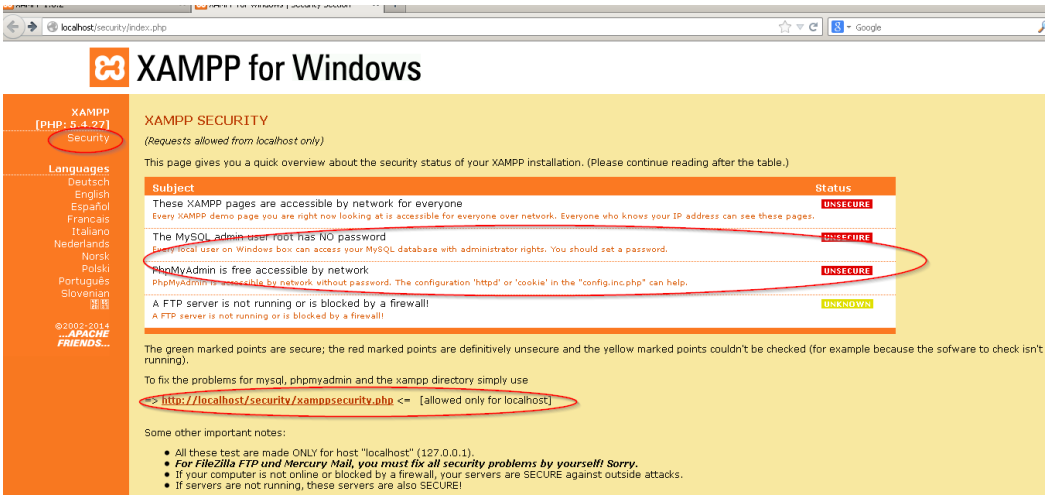
Εικόνα Α-5: Περιήγηση σε βάση δεδομένων στο phpMyAdmin

Επειδή το XAMPP έγκαθιστά την MySQLχωρίς κωδικό πρόσβασης, είναι καλύτερα να δημιουργήσουμε έναν κωδικό, τον οποίο Θα χρησιμοποιήσουμε και στο connection String μέσω του οποίου θα αποκτήσει πρόσβαση ο rlgameserver στη βάση δεδομένων. Για να θέσουμε έναν κωδικό ανοίγουμε τον browser και βάζουμε τη διεύθυνση localhost και εμφανίζεται άλλο ένα διαχειριστικό εργαλείο του XAMPPόπως δείχνει η Εικόνα Α-6.



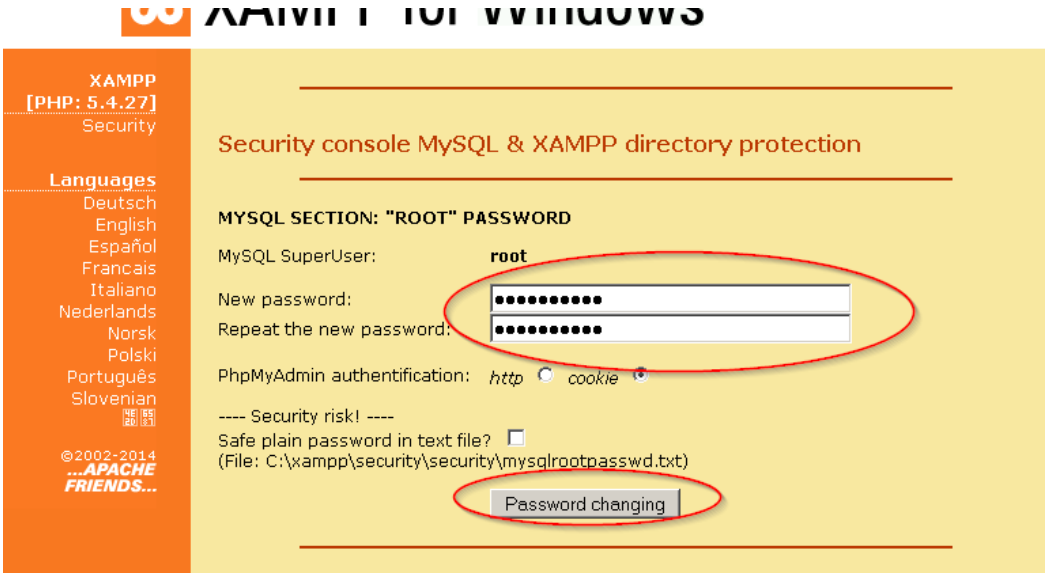
Εικόνα Α-6: web διαχειριστικό εργαλείο του XAMPP

Επιλέγουμε τον σύνδεσμο security και οδηγούμαστε στην παρακάτω σελίδα όπου ειδοποιούμαστε πως δεν έχουν οριστεί κωδικοί πρόσβασης για τον root user σε MySQL και phpMyAdmin, και επιλέγουμε τον αντίστοιχο σύνδεσμο για την επίλυση αυτού του προβλήματος.



Εικόνα Α-7: Επιλογή ρυθμίσεων ασφαλείας της MySql

Στη σελίδα που οδηγούμαστε, συμπληρώνουμε τα πεδία για την δημιουργία του νέου κωδικού πρόσβασης και πατάμε το κουμπί για την επιβεβαίωση των αλλαγών.



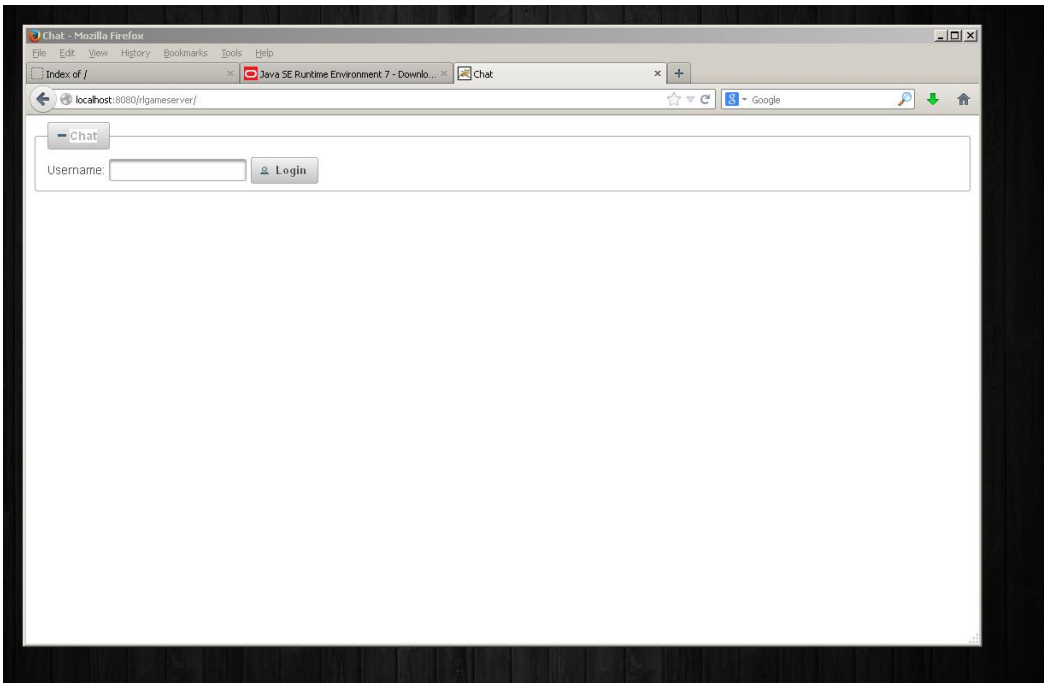
Εικόνα Α-8: Αλλαγή κωδικού πρόσβασης για την MySql

Για να χρησιμοποιήσουμε απευθείας το .warfile που έχει παραδοθεί και να μην κάνουμε build τον κώδικα από την αρχή, επιλέγουμε τον κωδικό 123456.

Κατόπιν παίρνουμε το .warfile (rlgameserver.war) και το τοποθετούμε μέσα στο document root του ApacheTomcat, το οποίο αν η εγκατάσταση του XAMPP έχει γίνει σωστά βρίσκεται στη διαδρομή **C:\xampp\tomcat\webapps.** Για να λειτουργήσει ο Tomcat πρέπει να έχουμε εγκατεστημένο στον υπολογιστή και ένα Java Runtime Environment(JRE) ή ένα Java Development Kit (JDK). Σε περίπτωση που αυτό δεν υπάρχει μπορούμε να το κατεβάσουμε από τον ιστοτόπο:

http://www.oracle.com/technetwork/java/javase/downloads/jre7-downloads-1880261.html

Αφού τοποθετήσουμε το .war file επανεκκινούμε τον Tomcat, πηγαίνουμε στη σελίδα localhost:8080/rlgameserver. Αν όλα έχουν γίνει επιτυχώς τότε θα δούμε την αρχική σελίδα του rlgamesever, όπως φαίνεται στην Εικόνα Α-9.



Εικόνα Α-9: Σελίδα καλωσορίσματος του rlgameserver

Γράφουμε κάποιο nickname στο πεδίο εισόδου στον server, πατάμε το κουμπί login, και θα οδηγηθούμε στην κυρίως οθόνη του rlgameserver για να ξεκινήσουμε τη διεξαγωγή κάποιου παιχνιδιού.