ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΠΛΗ 311 – Τεχνητή Νοημοσύνη – 2022

Εαρινό εξάμηνο 2021-22 **2η Προγραμματιστική Εργασία** Παράδοση : 29/5/2022

Σημείωση: Καμμία ερώτηση δε θα απαντιέται μετά την 24η Μαϊου 2022

Ομαδική (max 2 άτομα/ομάδα)

Βάρος: 30% Βαθμού Μαθήματος

Παίζοντας το Παιχνίδι TUC-CHESS

1. Εισαγωγή

Στα πλαίσια της εργασίας προγραμματισμού θα σχεδιάσετε και θα υλοποιήσετε ένα πρόγραμμα το οποίο θα παίζει το παιχνίδι TUC-CHESS. Στο παρόν κείμενο υπάρχει ένας οδηγός με τους κανόνες του παιχνιδιού, καθώς και μια μικρή περιγραφή του κώδικα που σας δίνεται ως βάση.

Μετά την παράδοση των εργασιών, ανάλογα και με τον αριθμό τους, ενδέχεται να διεξαχθεί ένα πρωτάθλημα TUC-CHESS μεταξύ όλων των πρακτόρων. Σε μια τέτοια περίπτωση, θα δοθεί επιπλέον βαθμολογία (bonus), στον δεύτερο ίση με 1% επί του βαθμού μαθήματος, και στον πρώτο ίση 2% επί του βαθμού μαθήματος.

Είστε ελεύθεροι να χρησιμοποιήσετε όποια γλώσσα προγραμματισμού επιθυμείτε αρκεί να υπάρχει συμβατότητα επικοινωνίας με τον server του παιχνιδιού που θα σας δοθεί.

2. Το Παιχνίδι

Το παιχνίδι TUC-CHESS δημιουργήθηκε για τις ανάγκες της εργασίας προγραμματισμού του μαθήματος, και αποτελεί μια παραλλαγή του κλασικού παιχνιδιού σκάκι (του οποίου οι σημερινοί κανόνες ισχύουν με μικρές παραλλαγές από το 1475 μ.Χ. περίπου, ενώ εκτιμάται ότι υπάρχουν πάνω από 2000 παραλλαγές του σκακιού).

Το TUC-CHESS παίζεται σε μία σκακιέρα διαστάσεων 7×5 και υπάρχουν δύο παίκτες, ο λευκός και ο μαύρος, καθένας από τους οποίους έχει 7 πιόνια, 2 πύργους και 1 βασιλιά. Οι παίκτες παίζουν εναλλάξ και, κατά σύμβαση, η πρώτη κίνηση ανήκει στο λευκό παίκτη.

Κάθε πιόνι μπορεί να κινηθεί μόνο προς τη μεριά του αντιπάλου και μόνο κατά μία θέση, η οποία μπορεί να είναι είτε στην επόμενη γραμμή της ίδιας στήλης, εφόσον δεν υπάρχει κάποιος άλλος πεσσός (του ίδιου παίκτη ή του αντιπάλου) σε αυτή τη θέση, είτε σε κάποια διαγώνια θέση εφόσον υπάρχει κάποιος πεσσός του αντιπάλου σε αυτήν. Επισημαίνεται ότι **δεν** επιτρέπεται κατά την πρώτη κίνηση ενός πιονιού να προχωρήσει αυτό κατά δύο θέσεις, όπως συμβαίνει στο κλασικό σκάκι. Όταν ένα πιόνι φτάσει στην τελευταία γραμμή τότε απομακρύνεται από την σκακιέρα διαπαντός, και ο παίκτης στον οποίο ανήκει κερδίζει 1 πόντο. Η αιχμαλώτιση ενός πιονιού αυξάνει

τη βαθμολογία του παίκτη κατά 1 πόντο. Με τον όρο «αιχμαλώτιση» εννοούμε την απομάκρυνση του αντίπαλου πεσσού από τη σκακιέρα, και μπορεί να συμβεί όταν μία επιτρεπτή κίνηση του παίκτη είναι προς μία θέση στην οποία βρίσκεται ένας πεσσός του αντιπάλου.

Κάθε πύργος μπορεί να μετακινηθεί καθέτως ή οριζοντίως, όπως στο κλασικό σκάκι, αλλά μόνο κατά τρεις θέσεις. Φυσικά, δεν μπορεί να μετακινηθεί στη δεύτερη ή τρίτη κατά σειρά θέση μιας κατεύθυνσης αν η πρώτη ή η δεύτερη (αντίστοιχα) δεν είναι ελεύθερη (δεν υπάρχει κάποιος πεσσός, οποιουδήποτε παίκτη, στις ενδιάμεσες θέσεις). Η αιχμαλώτιση ενός πύργου αυξάνει τη βαθμολογία του παίκτη κατά 3 πόντους.

Κάθε βασιλιάς μπορεί να μετακινηθεί καθέτως ή οριζοντίως, όπως ο πύργος, αλλά μόνο κατά μία θέση. Η αιχμαλώτιση ενός βασιλιά αυξάνει τη βαθμολογία του παίκτη κατά 8 πόντους.

Επίσης, στο παιχνίδι υπάρχουν bonuses που απεικονίζονται ως δώρα. *Κατά την αρχική κατάσταση του παιχνιδιού υπάρχουν bonus σε όλες τις στήλες της μεσαίας γραμμής της σκακιέρας.* Ένα bonus δίνει σε έναν παίκτη που το αποκτά (μετακινείται σε αυτό) *1 πόντο με πιθανότητα 0.95* και *κανέναν πόντο με πιθανότητα 0.05*, ενώ μόλις αποκτηθεί (το bonus) απομακρύνεται από τη σκακιέρα. Μετά την κάθε κίνηση ενός παίκτη υπάρχει πιθανότητα *0.2* να εμφανιστεί στην σκακιέρα ένα νέο bonus (ισοπίθανα σε μια οποιαδήποτε ελεύθερη θέση).

Το παιχνίδι τερματίζεται όταν αιχμαλωτιστεί κάποιος βασιλιάς ή όταν οι μόνοι πεσσοί που έχουν απομείνει στο παιχνίδι είναι οι δύο βασιλιάδες ή όταν ξεπεραστεί το χρονικό όριο των 12 λεπτών από την εκκίνηση του server (πληροφορίες παρακάτω). Στόχος του κάθε παίκτη είναι να πετύχει μεγαλύτερη βαθμολογία από την αντίπαλο. Ο παίκτης με τη μεγαλύτερη βαθμολογία στο τέλος της παρτίδας ανακηρύσσεται νικητής. Αν οι παίκτες έχουν την ίδια βαθμολογία (έχουν συλλέξει τον ίδιο αριθμό πόντων) τότε η παρτίδα λήγει με ισοπαλία.

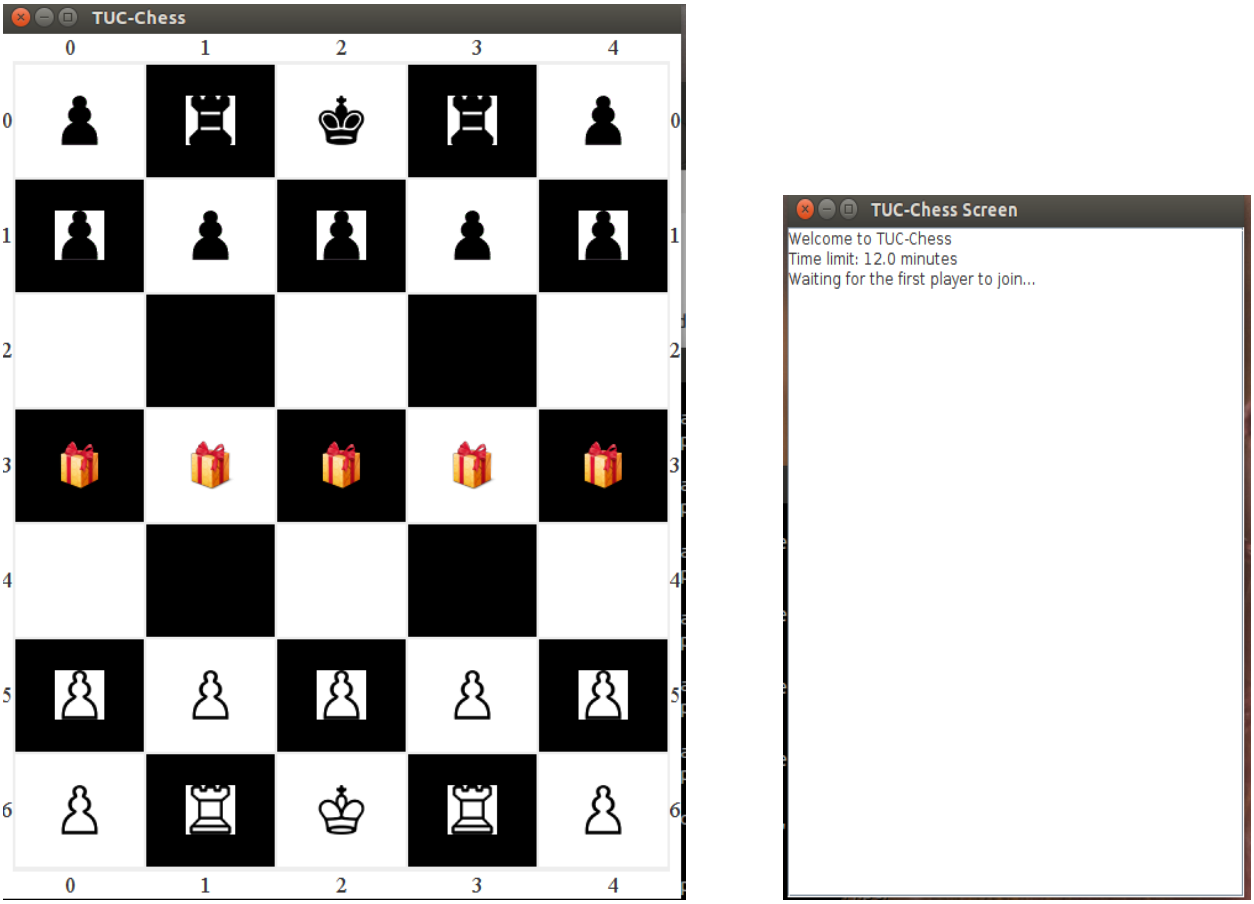
Επίσης, υπάρχει ένα γραφικό παράθυρο στο οποίο εμφανίζονται μηνύματα σχετικά με την εξέλιξη του παιχνιδιού, μεταξύ των οποίων και τα μηνύματα που λαμβάνει ο server από τους παίκτες. Ακόμα, οι γραμμές και οι στήλες είναι αριθμημένες έτσι ώστε οι πρώτες εξ αυτών να έχουν ως δείκτη το μηδέν.

Η αρχική κατάσταση του παιχνιδιού φαίνεται στην Εικόνα 1. Οι πεσσοί του λευκού παίκτη καταλαμβάνουν τις δύο κάτω γραμμές, δηλαδή τις 5 και 6. Στη γραμμή 5 υπάρχουν μόνο πιόνια, ενώ στη γραμμή 6 υπάρχουν με τη σειρά που αναφέρονται : πιόνι, πύργος, βασιλιάς, πύργος, πιόνι. Οι πεσσοί του μαύρου παίκτη καταλαμβάνουν τις δύο πάνω γραμμές, δηλαδή τις 0 και 1. Στη γραμμή 1 υπάρχουν μόνο πιόνια, ενώ στη γραμμή 1 υπάρχουν (όπως και για το λευκό παίκτη) με τη σειρά που αναφέρονται : πιόνι, πύργος, βασιλιάς, πύργος, πιόνι.

Στην Εικόνα 2 φαίνεται ένα στιγμιότυπο από μία παρτίδα ενός παιχνιδιού η οποία έχει ξεκινήσει αλλά δεν έχει τελειώσει, εξελίσσεται δηλαδή ακόμα. Όπως φαίνεται από το παράθυρο μηνυμάτων του παιχνιδιού (Εικόνα 2β) η τελευταία κίνηση έχει γίνει από το μαύρο παίκτη, το όνομα του οποίου είναι “client6”. Σε αυτό το σημείο της παρτίδας ο μαύρος παίκτης μετακίνησε έναν από τους πεσσούς του, συγκεκριμένα ένα πιόνι, από τη θέση 23 (γραμμή 2 – στήλη 3) στη θέση 33 (γραμμή 3 – στήλη 3).

Στην Εικόνα 3 φαίνεται ένα στιγμιότυπο από μία παρτίδα ενός παιχνιδιού η οποία έχει τερματιστεί. Η βαθμολογία του αγώνα ήταν 16-21, δηλαδή κέρδισε ο μαύρος παίκτης καθώς σε κάθε παρτίδα το πρώτο από τα δύο σκορ είναι του λευκού παίκτη ενώ το δεύτερο του μαύρου. Η

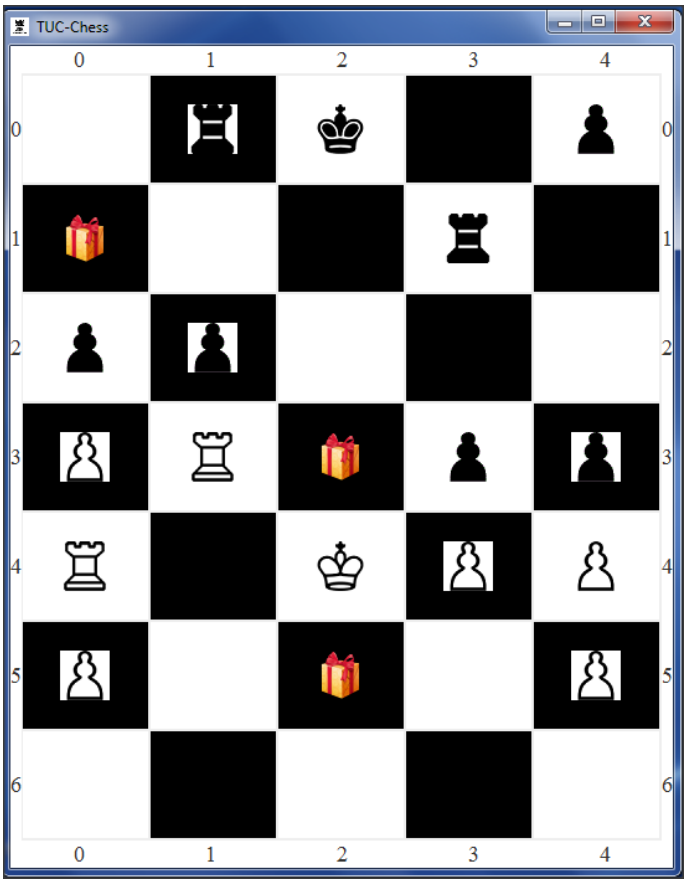
τερματική συνθήκη που ενεργοποιήθηκε ήταν αυτή της αιχμαλώτισης ενός βασιλιά (και όχι της παραμονής μόνο δύο βασιλιάδων στο παιχνίδι). Έτσι, στην τελευταία κίνηση της παρτίδας ο “client3” αιχμαλώτισε με έναν πύργο το βασιλιά του “client9” (λευκός παίκτης) κάνοντας την κίνηση 3031 (μετακίνηση του πεσσού από την γραμμή 3 και τη στήλη 0 στην γραμμή 3 και τη στήλη 1).



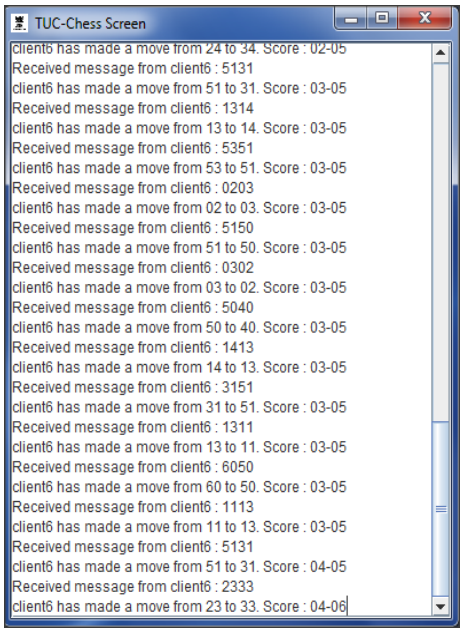
α. Αρχική κατάσταση της σκακιέρας.

β. Η οθόνη μηνυμάτων του παιχνιδιού κατά την αρχική του κατάσταση.

Εικόνα 1. Αρχική κατάσταση του παιχνιδιού.

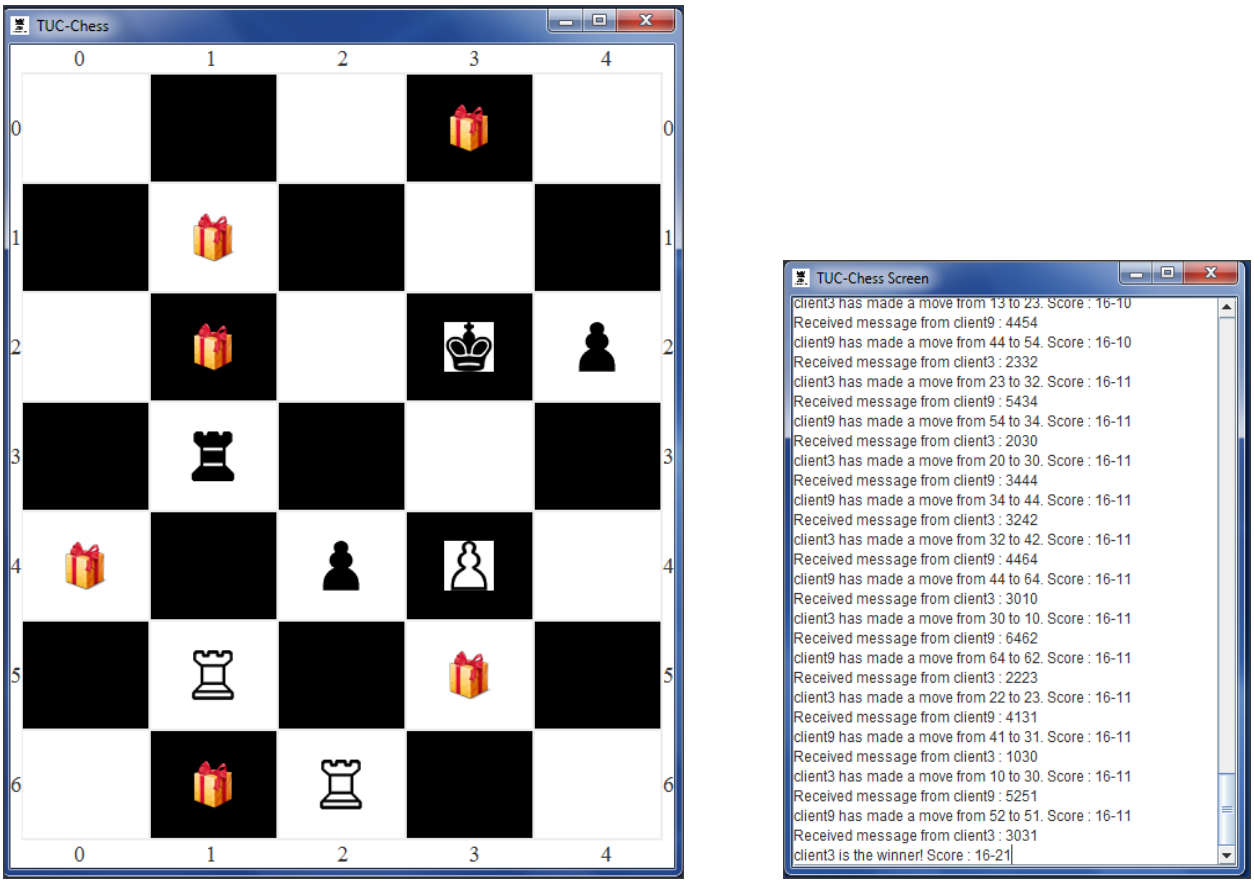


α. Κατάσταση σκακιέρας ενώ το παιχνίδι έχει ξεκινήσει αλλά δεν έχει τερματιστεί.



β. Η οθόνη μηνυμάτων του παιχνιδιού σε αυτή την κατάσταση.

Εικόνα 2. Παιχνίδι σε εξέλιξη.



α. Κατάσταση σκακιέρας όταν το παιχνίδι έχει τερματιστεί.

β. Η οθόνη μηνυμάτων του παιχνιδιού κατά τον τερματισμό του.

Εικόνα 3. Τερματισμός παιχνιδιού.

Συνοψίζοντας, οι κανόνες έχουν ως εξής :

- Οι παίκτες παίζουν εναλλάξ και ο λευκός παίζει πρώτος.
- Ο λευκός παίκτης μετακινεί τους λευκούς πεσσούς και ο μαύρος τους μαύρους.

- Τα πιόνια μετακινούνται πάντα προς τη μεριά του αντιπάλου, είτε ένα βήμα μπροστά αν δεν υπάρχει κάποιος πεσσός σε εκείνη τη θέση είτε ένα βήμα διαγωνίως (αριστερά ή δεξιά) αν υπάρχει κάποιος πεσσός του αντιπάλου σε εκείνη τη θέση.
- Οι πύργοι μετακινούνται μέχρι και κατά τρεις θέσεις προς οποιαδήποτε κάθετη και οριζόντια κατεύθυνση (μη διαγώνια). Αν στις ενδιάμεσες θέσεις κάποιας κατεύθυνσης υπάρχει πεσσός τότε δεν επιτρέπεται η μετακίνηση.
- Οι βασιλιάδες μετακινούνται προς οποιαδήποτε κάθετη και οριζόντια κατεύθυνση κατά μία θέση.
- Τα πιόνια που φτάνουν στην τελευταία γραμμή (γραμμή 0 για τα λευκά και γραμμή 6 για τα μαύρα) απομακρύνονται από τη σκακιέρα, κερδίζοντας 1 πόντο.
- Όλοι οι πεσσοί που αιχμαλωτίζονται απομακρύνονται από τη σκακιέρα διαπαντός.
- Η αιχμαλώτιση ενός πιονιού αυξάνει τη βαθμολογία του παίκτη κατά 1 πόντο.
- Η αιχμαλώτιση ενός πύργου αυξάνει τη βαθμολογία του παίκτη κατά 3 πόντους.
- Η αιχμαλώτιση ενός βασιλιά αυξάνει τη βαθμολογία του παίκτη κατά 8 πόντους.
- Ένα bonus δίνει, στον παίκτη που μετακινείται στη θέση που αυτό βρίσκεται, 1 πόντο με πιθανότητα 0.95 και 0 πόντους με πιθανότητα 0.05.
- Μετά την κίνηση ενός παίκτη ένα νέο bonus εμφανίζεται σε κάποια από τις θέσεις στις οποίες δεν υπάρχει πεσσός ή bonus, με πιθανότητα 0.2.
- Για οποιαδήποτε κίνηση ABCD ισχύει ότι A, C ∈ {0, 1, 2, 3, 4, 5, 6} (7 γραμμές) και B, D ∈ {0, 1, 2, 3, 4} (5 στήλες).
- Κάθε παρτίδα λήγει είτε όταν ένας από τους δύο βασιλιάδες αιχμαλωτιστεί ή όταν στο παιχνίδι δεν έχει μείνει κανένας άλλος πεσσός εκτός των δύο βασιλιάδων ή όταν ξεπεραστεί το χρονικό όριο των 12 λεπτών.
- Για την αποστολή της κίνησή σας, όταν είναι η σειρά σας να παίξετε, μην ξεπερνάτε τα **4 δευτερολέπτα.**

3. Διαδικαστικά

Η διαδικασία που πρέπει να ακολουθήσετε για να ολοκληρώσετε την εργασία είναι η παρακάτω :

- Διαβάστε προσεκτικά το παρόν κείμενο και τον κώδικα που το συνοδεύει.
- Βεβαιωθείτε ότι μπορείτε να τρέξετε τον κώδικα και να κατανοήσετε τη λειτουργία του.

Α Μέρος (αξία: περίπου 20% βαθμού μαθήματος, δεδομένης της απόδοσής σας στο Γ μέρος)

- Υλοποιήστε τον αλγόριθμο αναζήτησης minimax.
- Υλοποιήστε μια καλή συνάρτηση αξιολόγησης και μια μέθοδο αποκοπής.
- Βελτιώστε την αναζήτηση υλοποιώντας α-β pruning.
- Βελτιώστε περαιτέρω την αναζήτηση με singular extensions, forward pruning, κλπ.

Β Μέρος (αξία: περίπου 10% βαθμού μαθήματος, δεδομένης της απόδοσής σας στο Γ μέρος)

- Υλοποιήστε τον αλγόριθμο αναζήτησης MCTS (Monte Carlo Tree Search)

- Δοκιμάστε την απόδοση του αλγορίθμου για διαφορετικό αριθμό iterations.

Γ Μέρος

- Συγκρίνετε τις δύο διαφορετικές υλοποιήσεις, και περιγράψτε τα αποτελέσματα σας.
- Βεβαιωθείτε ότι το πρόγραμμα σας συνεργάζεται με τον server χωρίς προβλήματα.
- Παραδώστε την εργασία σας μέχρι την ημερομηνία παράδοσης.
- Παραδώστε μια σύντομη αναφορά (**8 σελίδες max – μαζί με τη βιβλιογραφία**) για τις τεχνικές που χρησιμοποιήσατε και υλοποιήσατε. Στην αναφορά πρέπει να περιλαμβάνεται και η περιγραφή των επιλογών των evaluation functions και cutoff search μεθόδων που χρησιμοποιήσατε. Επίσης να αναφέρετε τις επιλογές σας σχετικά με τον αλγόριθμο MCTS, και παρέχεται περιγραφή των συγκρίσεων/αποτελεσμάτων σας.

4. Εκπόνηση

Α Μέρος

Ο αλγόριθμος αναζήτησης minimax και το α-β pruning καθώς και άλλες τεχνικές για παιχνίδια έχουν καλυφθεί εκτενώς στο μάθημα. Επίσης, τα βιβλία του μαθήματος περιέχουν όλες τις λεπτομέρειες και τον ψευδοκώδικα των αλγορίθμων αναζήτησης που χρειάζεται για την εργασία.

Σκεφτείτε τον τρόπο με τον οποίο θα δημιουργήσετε το δέντρο της αναζήτησης. Η μέθοδος που θα ακολουθήσετε μπορεί να επηρεάσει σημαντικά την ταχύτητα του παίκτη σας. Προφανώς, δεν θα υπάρχει αρκετός χρόνος για να ψάξει το πρόγραμμα σας ολόκληρο το δέντρο αναζήτησης για την καλύτερη κίνηση. Έτσι, θα πρέπει να ψάχνετε για την καλύτερη δυνατή κίνηση σε λογικά πλαίσια πραγματικού χρόνου.

Ακολουθούν κάποια σχόλια και προτάσεις για την εκπόνηση της εργασίας.

- Βεβαιωθείτε ότι μετά την υλοποίηση του α-β pruning η αναζήτησή σας καταλήγει ακριβώς στις ίδιες τιμές με τον minimax. Θα ήταν σκόπιμο να έχετε ένα μηχανισμό για να ενεργοποιείτε και να απενεργοποιείτε το α-β pruning κατά βούληση.
- Ο απλός αλγόριθμος α-β pruning εξετάζει τα παιδιά κάθε κόμβου με τη σειρά δημιουργίας, αλλά μπορείτε να υλοποιήσετε κάτι πιο έξυπνο. Όπως ξέρετε, η σειρά με την οποία γίνονται επεκτάσεις κόμβων έχει μεγάλη επιρροή στην αποτελεσματικότητα του κλαδέματος.
- Μια αρχική (απλή) συνάρτηση αξιολόγησης κατάστασης (state) που μπορείτε να χρησιμοποιήσετε σε συνδυασμό με κάποια μέθοδο αποκοπής είναι η παρακάτω :

  V(state) = | my (points + pieces) | - | opponent's (points + pieces) |

- Μπορείτε να χρησιμοποιήσετε οποιαδήποτε τεχνική βελτίωσης βρείτε διαθέσιμη στο διαδίκτυο ή σε βιβλία αρκεί να εξηγήσετε στην αναφορά σας τι κάνει, πώς το κάνει, γιατί την επιλέξατε και πώς την προσαρμόσατε στον κώδικα σας.
- Στην αναφορά σας περιγράψτε τις ιδιαιτερότητες του κώδικά σας και τις τυχόν προεκτάσεις ή έξυπνες λύσεις που υλοποιήθηκαν. Θα πρέπει απαραίτητα να υπάρχουν αναφορές των πηγών σας, **τυφλή αντιγραφή (plagiarism) οδηγεί σε μηδενισμό στο μάθημα.**

Β Μέρος

Ο MCTS είναι ένας αλγόριθμος αναζήτησης ο οποίος χρησιμοποιείται κατά κόρον στην υλοποίηση πρακτόρων για board games. Το πιο γνωστό παράδειγμα πράκτορα που χρησιμοποιεί MCTS είναι ο Alpha Go, ο οποίος είναι πρωταθλητής στο παιχνίδι Go. Ένα κλασσικό άρθρο με τίτλο "A Survey of Mone Carlo Tree Search Methods", από τον Browne και συν-συγγραφείς, και το οποίο περιγράφει την τεχνική MCTS, τις αρχές της, και παραλλαγές της, μπορεί να βρεθεί εδώ: http://www.incompleteideas.net/609%20dropbox/other%20readings%20and%20resources/MCTS-survey.pdf

Ο αλγόριθμος, για να πάρει μία απόφαση ενέργειας από μια τρέχουσα κατάσταση, χτίζει προοδευτικά ένα δένδρο αναζήτησης (υπό αντιπαλότητα) από την τρέχουσα κατάσταση, μέσω διαδοχικών επαναλήψεων εντός των οποίων χρησιμοποιούνται στάδια διάσχισης και επέκτασης του δένδρου αναζήτησης με επιλογή ενεργειών και αντίστοιχη προσθήκη κόμβων, καθώς και στάδια *προσομοιώσεων* μελλοντικών κινήσεων (μέσω λχ εντελώς τυχαίας επιλογής ενεργειών) ως ότου φτάσουμε σε έναν κόμβο-φύλλο. Η διαδικασία αυτή παράγει διαδοχικές εκτιμήσεις αξίας για τους κόμβους του δένδρου, οι οποίες αναθεωρούνται μέσω μελλοντικών επαναλήψεων. Λεπτομέρειες μπορείτε να βρείτε στο παραπάνω άρθρο, και ο γενικός αλγόριθμος θα σας εξηγηθεί αναλυτικά σε φροντιστήρια του μαθήματος.

Μερικά σχόλια και προτάσεις για την εκπόνηση του Β μέρους:

- Υλοποιήστε αρχικά την απλή συνάρτηση Upper Confidence Bound for Trees (UCT), με την οποία επιλέγεται ως καλύτερη ενέργεια από τον τρέχοντα κόμβο το "καλύτερο παιδί" (best child) του τρέχοντος κόμβου. Συγκεκριμένα, το «καλύτερο παιδί» (καλύτερη ενέργεια) είναι αυτό (αυτή) με την *μεγαλύτερη* UCT αξία ανάμεσα σε όλα τα παιδιά του τρέχοντος κόμβου:

  $$UCT = V_i + 2\,C\sqrt{\frac{\ln N}{n_i}}$$

  όπου V_i είναι το *μέσο reward* του παιδιού i του συγκεκριμένου κόμβου, N είναι ο αριθμός των φορών που ο *τρέχον* κόμβος έχει δεχτεί επίσκεψη, C είναι μια σταθερά την οποία μπορείτε να θέσετε ίση με 1/ sqrt(2) αν οι αμοιβές είναι κανονικοποιημένες στο [0 ,1], και τέλος n_i είναι ο αριθμός των φορών που ο κόμβος-παιδί i έχει δεχτεί επίσκεψη.
- Πειραματιστείτε με διαφορετικό αριθμό iterations και αν έχετε χρόνο δοκιμάστε και διαφορετικές συναρτήσεις αντί για τη UCT.

Γενικά και για το Α και για το Β μέρος ισχύουν τα ακόλουθα:

- Μπορείτε να χρησιμοποιήσετε οποιαδήποτε τεχνική βελτίωσης βρείτε διαθέσιμη στο διαδίκτυο ή σε βιβλία αρκεί να εξηγήσετε στην αναφορά σας τι κάνει, πώς το κάνει, γιατί την επιλέξατε και πώς την προσαρμόσατε στον κώδικα σας. Ακόμα αναφέρετε τις πηγές σας στην τελική αναφορά.

- Στην αναφορά σας περιγράψτε τις ιδιαιτερότητες του κώδικά σας και τις τυχόν προεκτάσεις ή έξυπνες λύσεις που υλοποιήθηκαν. Θα πρέπει απαραίτητα να υπάρχουν αναφορές των πηγών σας, **τυφλή αντιγραφή (plagiarism) οδηγεί σε μηδενισμό στο μάθημα**.
- Τέλος αναφέρετε αποτελέσματα, συμπεράσματα και ιδέες για το πως θα μπορούσατε να έχετε καλύτερα αποτελέσματα.

5. Υλοποίηση

Μπορείτε να γράψετε τον κώδικα σας ως συνέχεια του κώδικα TUC-CHESS που σας δίνεται (ενότητα Υλικό Εργαστηρίου στο courses). Εκεί υπάρχει υλοποιημένη όλη η απαραίτητη λειτουργικότητα του παιχνιδιού, ώστε να ελαττωθεί ο φόρτος υλοποίησης. Συγκεκριμένα περιλαμβάνεται ο κώδικας ενός βασικού παίκτη (client) και του server του παιχνιδιού σε Java.

Ο κώδικας παρέχει τους βασικούς μηχανισμούς για τη διατήρηση της λειτουργικότητας του παιχνιδιού κατά την εξέλιξη. Ο server ελέγχει για κάθε κίνηση που λαμβάνει τόσο αν αυτή προέρχεται από τον παίκτη του οποίου είναι η σειρά να παίξει όσο και αν αυτή είναι έγκυρη. Δεν επιτρέπονται αλλαγές στο πρωτόκολλο επικοινωνίας. Αν αλλάξετε τον τρόπο επικοινωνίας, η συμμετοχή σας στο τουρνουά δεν θα είναι εφικτή.

5.1 Υποδομή

Η επικοινωνία του server και των clients επιτυγχάνεται μέσω του πρωτοκόλλου UDP (User Datagram Protocol). Περισσότερα για αυτόν τον τρόπο επικοινωνίας θα παρουσιαστούν σε φροντιστήριο του μαθήματος. Θα ήταν χρήσιμο να ελέγξετε εκ των προτέρων μόνοι σας πώς επιτυγχάνεται η επικοινωνία server και client στον κώδικα που σας δίνεται. Η επικοινωνία τους βασίζεται αυστηρά στην ανταλλαγή αλφαριθμητικών. Η επικοινωνία είναι ανεξάρτητη της γλώσσας προγραμματισμού, κι έτσι όποια κι αν επιθυμείτε να χρησιμοποιήσετε δεν πρέπει να έχετε πρόβλημα. Ωστόσο αν αυτή είναι η Java τότε μπορείτε να χρησιμοποιήσετε τον κώδικα του client που σας δίνεται για την υλοποίηση της επικοινωνίας με τον server.

Τα βασικά στοιχεία που είναι απαραίτητα να γνωρίζετε είναι τα εξής :

- Η IP διεύθυνση του server είναι αυτή του local host.
- Ο server δέχεται μηνύματα στη θύρα (port) **9876.**
- Τα πακέτα δεδομένων που στέλνει και δέχεται ο server μπορούν να έχουν μέγεθος μέχρι και 200 byte (παραπάνω από αρκετά στην περίπτωσή μας).

5.2 Πρωτόκολλο Επικοινωνίας

Όταν ο server ξεκινά να "τρέχει" περιμένει να λάβει ένα μήνυμα από τον πρώτο παίκτη, το οποίο θα περιέχει ένα αλφαριθμητικό με το όνομά του. Ως απάντηση ο server στέλνει στον πρώτο παίκτη

το αλφαριθμητικό “PW”, γνωστοποιώντας του με αυτόν τον τρόπο ότι είναι ο λευκός παίκτης. Έτσι, μία σύμβαση του παιχνιδιού είναι ότι ο παίκτης ο οποίος επικοινωνεί πρώτος με τον server θα είναι και ο λευκός, έχοντας την πρώτη κίνηση του παιχνιδιού. Στη συνέχεια ο server περιμένει να λάβει ένα μήνυμα από το δεύτερο παίκτη, το οποίο θα περιέχει ένα αλφαριθμητικό με το όνομά του. Ως απάντηση ο server στέλνει στο δεύτερο παίκτη το αλφαριθμητικό “PB”, γνωστοποιώντας του με αυτό τον τρόπο ότι είναι ο μαύρος παίκτης. Έτσι, ένας client κατά την εκκίνηση του πρέπει να στείλει στον server ένα μήνυμα το οποίο θα περιέχει ένα αλφαριθμητικό με το όνομά του και στη συνέχεια να περιμένει να λάβει μία απάντηση από αυτόν, η οποία θα είναι είτε το αλφαριθμητικό “PW” είτε το αλφαριθμητικό “PB”. Ελέγχοντας αν ο πρώτος χαρακτήρας ισούται με το “P”, ο client μπορεί να γνωρίζει αν το μήνυμα που έλαβε περιέχει την πληροφορία για το χρώμα των πεσσών του, ελέγχοντας αν ο δεύτερος χαρακτήρας ισούται με “W” ή “B” (δε γίνεται να στείλει ο server κάτι διαφορετικό από αυτά ως πρώτο μήνυμα).

Αφού ο server λάβει πληροφορίες για τα ονόματα των παικτών, καθορίσει το χρώμα τους και αποθηκεύσει τα στοιχεία των διευθύνσεών τους (μέσω των μηνυμάτων που έχει δεχτεί – δε χρειάζεται οι clients να στείλουν κάτι περισσότερο για αυτά), μετρά αντίστροφα τρία δευτερόλεπτα (η αντίστροφη μέτρηση φαίνεται στην οθόνη του παιχνιδιού) κι έπειτα στέλνει και στους δύο παίκτες ένα μήνυμα το οποίο περιέχει το αλφαριθμητικό “GB”. Ο client έχοντας δεχτεί ένα μήνυμα, το οποίο περιέχει ένα αλφαριθμητικό, πρέπει να εξετάσει αν ο πρώτος χαρακτήρας ισούται με “G”. Αν αυτό ισχύει τότε πρέπει να εξετάσει αν ο δεύτερος χαρακτήρας ισούται με “B”. Αν και αυτό ισχύει τότε σημαίνει ότι η παρτίδα έχει ξεκινήσει και ότι ο server περιμένει από το λευκό παίκτη να του στείλει μία κίνηση. Σε αυτή την περίπτωση ο λευκός παίκτης θα πρέπει να αποφασίσει και να στείλει την κίνησή του στον server, ενώ ο μαύρος παίκτης μπορεί να περιμένει μέχρι να έρθει η σειρά του. Αφού ο server εξετάσει την εγκυρότητα του παίκτη και της κίνησης, ελέγχει την περίπτωση που το παιχνίδι έχει τερματιστεί. Αν αυτό είναι αληθές τότε στέλνει στους δύο clients ένα μήνυμα της μορφής “GE*ABCD*”, με κάθε χαρακτήρα του “*ABCD*” να είναι ένας ακέραιος εκφρασμένος σε αλφαριθμητική τιμή, εκφράζοντας την τελική βαθμολογία του παιχνιδιού. Π.χ. Το “GE2217” δηλώνει ότι το παιχνίδι έληξε και η βαθμολογία είναι 22-17 (με τον πρώτο αριθμό να δηλώνει τη βαθμολογία του άσπρου παίκτη και το δεύτερο του μαύρου) και οπότε νικητής είναι ο άσπρος παίκτης, ενώ το “GE0915” δηλώνει ότι το παιχνίδι έληξε και νικητής είναι ο μαύρος παίκτης με βαθμολογία 09-15. Έτσι, όταν ένας client λάβει ένα μήνυμα και ο πρώτος χαρακτήρας του αλφαριθμητικού που περιέχεται σε αυτό είναι το “G” τότε ο δεύτερος θα είναι είτε το “B” είτε το “E”, με την πρώτη περίπτωση να δηλώνεται ότι η παρτίδα έχει ξεκινήσει (και η πρώτη κίνηση πρέπει να γίνει από τον άσπρο παίκτη) και με τη δεύτερη περίπτωση να δηλώνεται ότι το παιχνίδι έχει λήξει (και στους επόμενους τέσσερις χαρακτήρες του αλφαριθμητικού βρίσκεται η βαθμολογία).

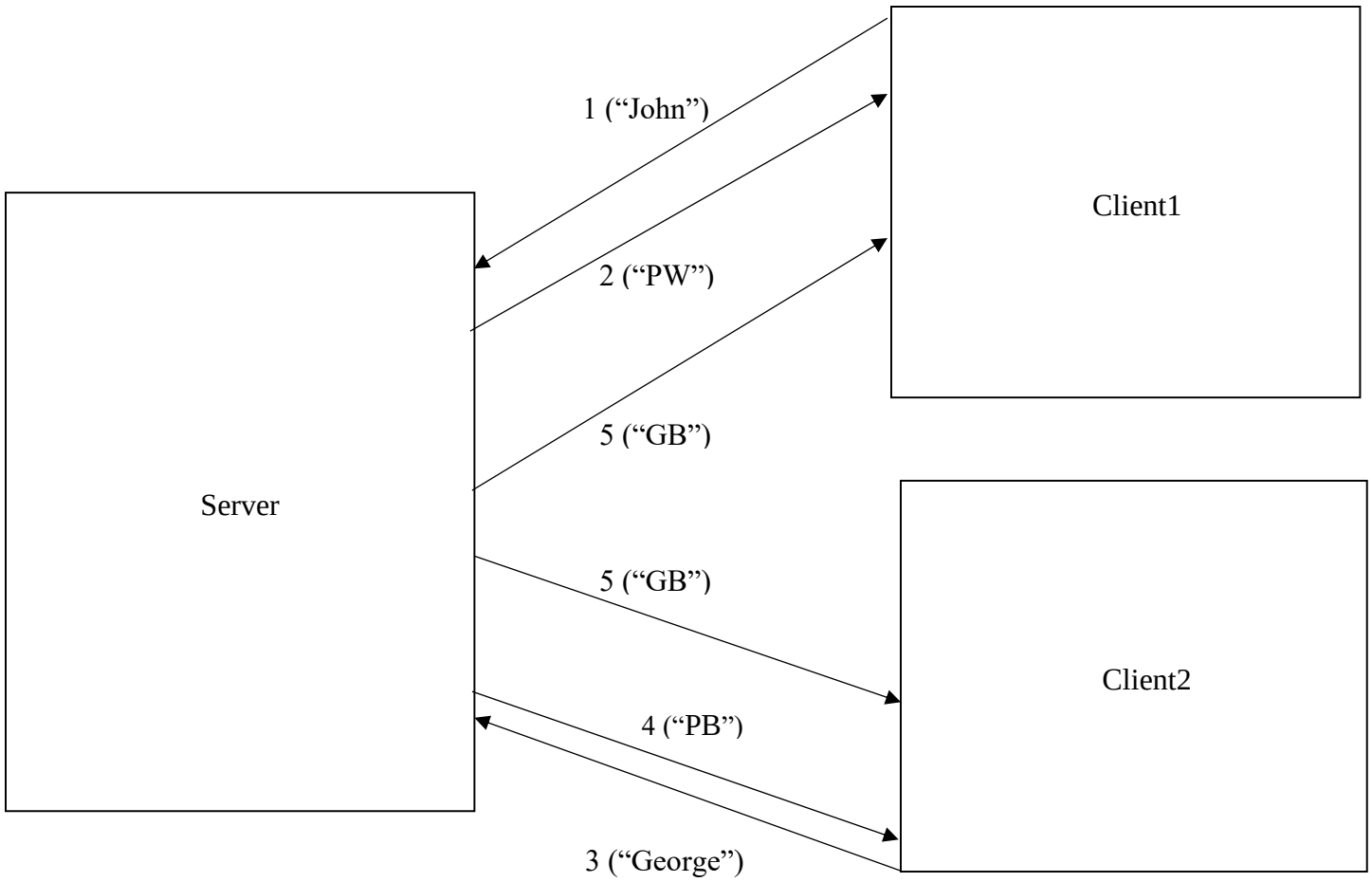
Όπως αναφέρθηκε στη δεύτερη ενότητα οι παίκτες στέλνουν στον server ένα αλφαριθμητικό το οποίο αναπαριστά την κίνησή τους (εκτός του πρώτου εξερχόμενου μηνύματός τους το οποίο περιέχει το όνομα του client). Για οποιαδήποτε κίνηση ABCD ισχύει ότι A, C ∈ {0, 1, 2, 3, 4, 5, 6} (7 γραμμές) και B, D ∈ {0, 1, 2, 3, 4} (5 στήλες). Έτσι, για τη μετακίνηση από την τρέχουσα θέση, έστω γραμμή 2 και στήλη 1, στη θέση που θέλει να βρεθεί, έστω γραμμή 2 και στήλη 3, ο client πρέπει στείλει στον server ένα μήνυμα το οποίο να περιέχει το αλφαριθμητικό “2123”. Αφού ο server ελέγξει την κίνηση (και εφόσον είναι έγκυρη), την πραγματοποιεί στη σκακιέρα του παιχνιδιού και στέλνει στους δύο clients ένα μήνυμα της μορφής “T*ABCDEFGHIJK*”, όπου :

- *A* ∈ {0,1} και δηλώνει τον παίκτη ο οποίος είναι ο επόμενος που πρέπει να παίξει, το μηδέν(0) είναι για τον λευκό παίκτη και το ένα(1) για τον μαύρο.
- *B, D* ∈ {0, 1, 2, 3, 4, 5, 6} και *C, E* ∈ {0, 1, 2, 3, 4}. Το *BCDE* δηλώνει την κίνηση που πραγματοποιήθηκε (προφανώς από τον παίκτη που δεν είναι ο *A*).
- F ∈ {0, 1, 2, 3, 4, 5, 6, 9} και G και ∈ {0, 1, 2, 3, 4, 9}. Το *FG* δηλώνει τη θέση του bonus που εμφανίστηκε. Αν έχει την τιμή "99" τότε δεν έχει εμφανιστεί κάποιο νέο bonus στη σκακιέρα.
- Το *HIJK* δηλώνει την τρέχουσα βαθμολογία της παρτίδας, με τον τρόπο που αναφέρθηκε προηγουμένως.

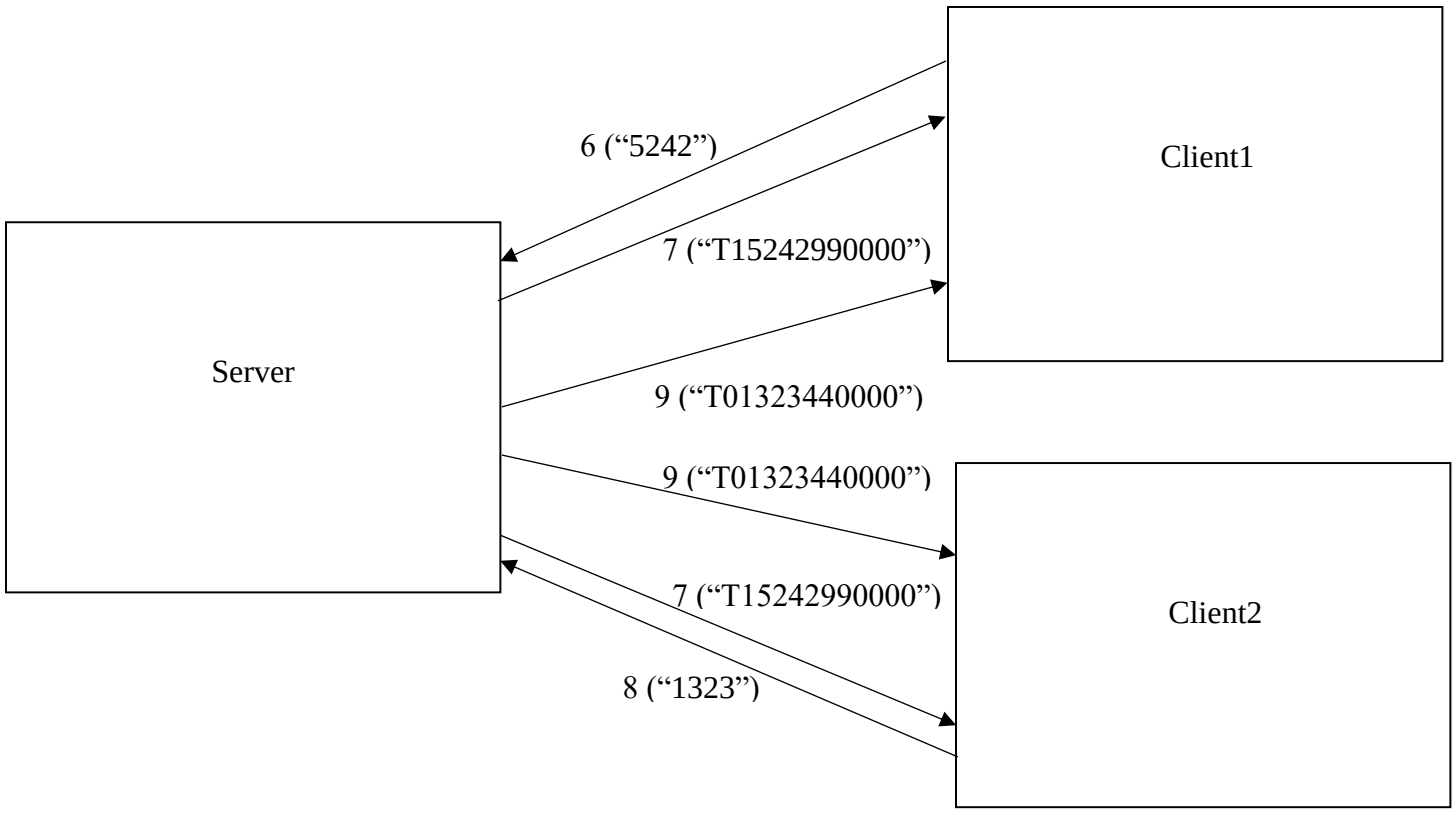
Έτσι, ένα μήνυμα που θα μπορούσε να στείλει ο server στους clients είναι το "T10102991109", το οποίο περιέχει τις εξής πληροφορίες :

- Η παρτίδα δεν έχει τελιώσει.
- Είναι η σειρά του μαύρου παίκτη να παίξει.
- Έχει πραγματοποιηθεί (προφανώς από τον άσπρο παίκτη) η κίνηση από τη θέση 01(γραμμή 0 – στήλη 1) στη θέση 02(γραμμή 0 – στήλη 2).
- Δεν έχει δημιουργηθεί κάποιο νέο bonus στη σκακιέρα.
- Η τρέχουσα βαθμολογία είναι 11-09 (άρα κερδίζει ο άσπρος παίκτης).

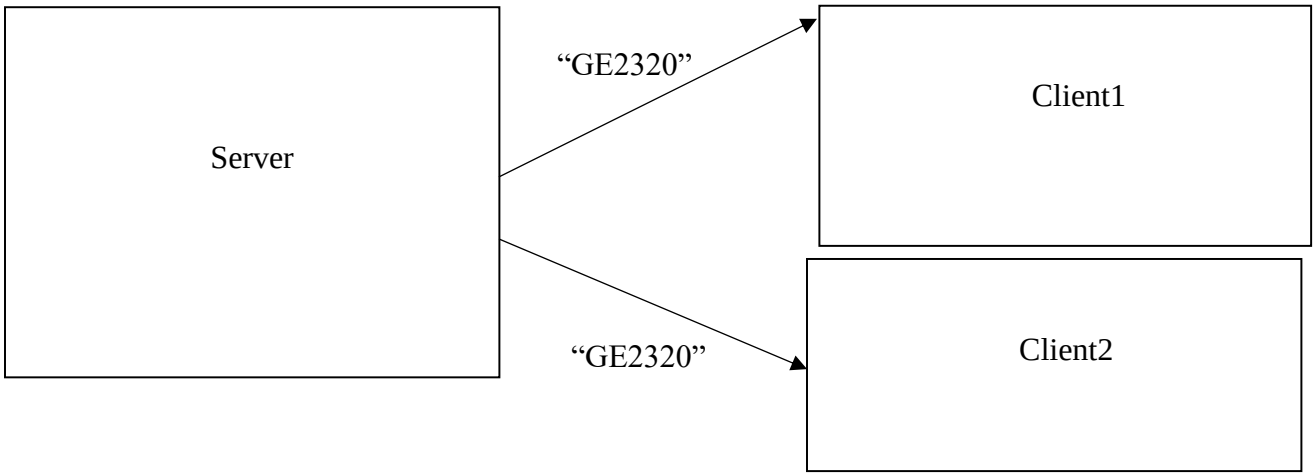
Το πρωτόκολλο επικοινωνίας φαίνεται και διαγραμματικά παρακάτω. Η φορά των τόξων προσδιορίζει τη φορά των μηνυμάτων, τα οποία είναι αριθμημένα έτσι ώστε να ορίζουν μια χρονική σειρά (το πρώτο μήνυμα είναι το 1) και με το περιεχόμενο των παρενθέσεων να ορίζει το περιεχόμενο του μηνύματος (προφανώς στα μηνύματα 1 και 3 το περιεχόμενο θα μπορούσε να είναι κάποιο άλλο έγκυρο αλφαριθμητικό).



Εικόνα 4. Επικοινωνία μεταξύ server και clients για να οριστεί η έναρξη του παιχνιδιού.



Εικόνα 5. Επικοινωνία μεταξύ server και clients κατά την εξέλιξη του παιχνιδού.



Εικόνα 6. Αποστολή μηνύματος λήξης του παιχνιδιού.

5.3 Client

Ο client που σας δίνεται είναι σχεδιασμένος για να επικοινωνεί με τον server (στο πρωτόκολλο UDP η επικοινωνία των sockets πραγματοποιείται χωρίς σύνδεση). Για να τρέξει ο client απλά γράφουμε :

```
java -jar tuc-chess-client.jar
```

Επίσης, υπάρχει η δυνατότητα για πέρασμα μιας παραμέτρου στον client, η οποία προσδιορίζει πόσο θα καθυστερήσει να στείλει στον server την ενέργειά του, αφού την αποφασίσει, π.χ.

```
java -jar tuc-chess-client.jar 1000
```

Η παραπάνω κλήση θα έχει ως αποτέλεσμα να καθυστερεί ο client την αποστολή της ενέργειας του στον server κατά 1000 ms, δηλαδή κατά ένα δευτερόλεπτο. Προσοχή! Δεν πρέπει να δοθεί ποτέ ως όρισμα η τιμή μηδέν, καθώς θα έχει ως αποτέλεσμα την αδυναμία αποστολής των ενεργειών του client στον server. Αν δε δώσετε όρισμα, τότε η καθυστέρηση έχει οριστεί να είναι 10 ms.

5.4 Server

Η δημιουργία των γραφικών παραθύρων (της σκακιέρας και της οθόνης του παιχνιδιού) γίνεται αυτόματα με την έναρξη της λειτουργίας του server. Για να επιτευχθεί αυτή γράφουμε :

```
java -jar tuc-chess-server.jar
```

6. Επίλογος

Όπως θα έχετε διαπιστώσει η εργασία είναι αρκετά ανοικτή με την έννοια ότι σας δίνει την ελευθερία να κάνετε πολλές αυθαίρετες επιλογές και να επικεντρωθείτε στα σημεία που σας ενδιαφέρουν περισσότερο. Επίσης, σας δίνει τη δυνατότητα για μελλοντικές επεκτάσεις με μεθόδους μηχανικής μάθησης ώστε ο παίκτης να βελτιώνει την απόδοσή του σταδιακά. Θα είμαστε στη διάθεσή σας για ό,τι προκύψει. Ξεκινήστε σήμερα κιόλας, βάλτε το μυαλό και τη φαντασία σας να δουλέψουν και διασκεδάστε ασκώντας τη δημιουργικότητά σας!

Καλή επιτυχία!!!