Σκάκι

Εφαρμογή Παιχνιδιού 2 χρηστών

Γρυπάρης Ιάσονας
ΑΜ 1059505

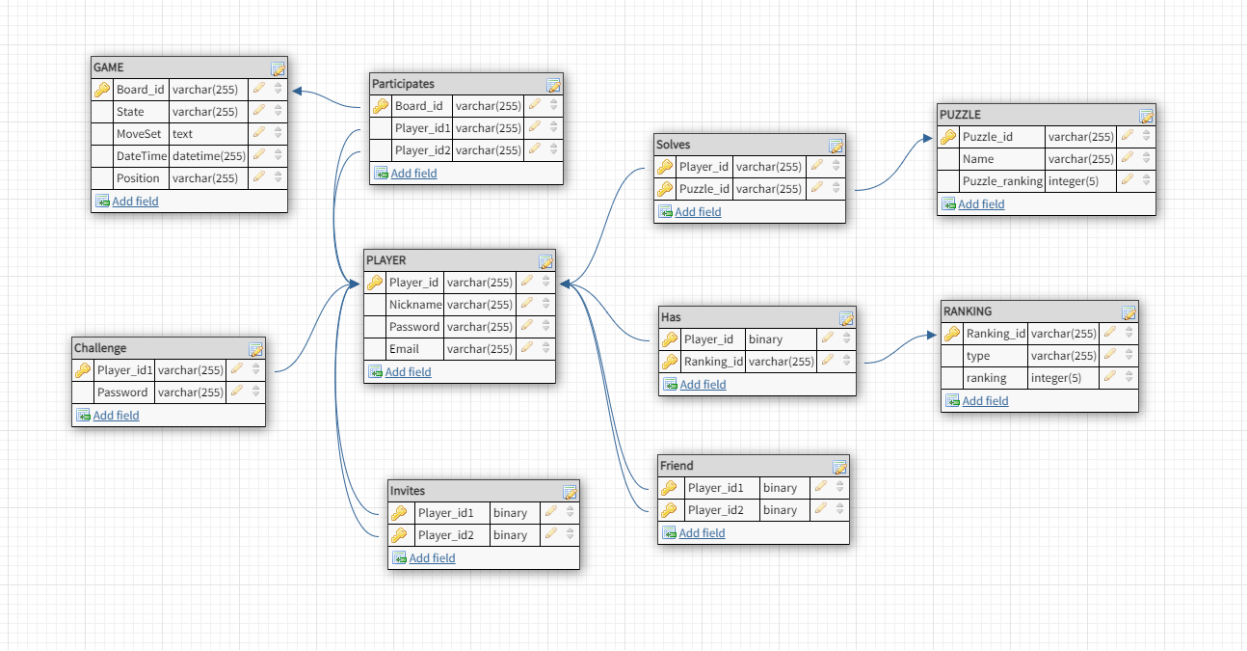
Λουκάς Λαμπίδης
ΑΜ 1059338

Περίληψη

Η περιγραφή του προβλήματος που ζητήθηκε να υλοποιηθεί είναι ένα παιχνίδι δύο παικτών. Το παιχνίδι που επιλέχθηκε είναι το σκάκι. Για την υλοποίηση του παιχνιδιού αυτού έγινε αρχικά μία ανάλυση μικρόκοσμου και καταγράφηκαν όλες οι μικροαλληλεπιδράσεις μεταξύ των στοιχείων του. Αφού, διαχωρίστηκαν οι αναγκαίες λειτουργίες σε client-side και server-side , αναπτύχθηκαν παράλληλα όλα τα κατάλληλα εργαλεία για την λειτουργεία της εφαρμογής. Στο τέλος της διαδικασίας ανάπτυξης έγινε η διασύνδεση όλων των εφαρμογών και δημοσιεύτηκε στην πλατφόρμα Heroku.

Παρόλα αυτά μέσω συζήτησης και έρευνας στο διαδίκτυο καταλήξαμε στην απόφαση η υλοποίηση να έχει γενικότερη εφαρμογή. Παιχνίδια με μεγάλο αριθμό παικτών παγκοσμίως όπως το σκάκι παρουσιάζουν εκτός από το αυτούσιο παιχνίδι και μεγάλο κομμάτι παραπλήσιων ασχολιών όπως τα puzzle σκακιού, διαφοροποιήσεις στους βασικούς κανόνες (αλλαγή χρόνου κλπ), βιβλιογραφία για εκπαίδευση και εξάσκηση και φυσικά μεγάλη κοινότητα. Οι ιστοσελίδες που φιλοξενούν τέτοια παιχνίδια, συνήθως φιλοξενούν και τα παραπλήσιά τους. Για αυτόν τον λόγο εκτός από το βασικό παιχνίδι, κατασκευάστηκε και υποδομή για να φιλοξενήσει και puzzles, βιβλιογραφία και σύνδεση μεταξύ των χρηστών. Όπως είναι φυσικό με την γενίκευση του προβλήματος δεν επαρκούσε ο χρόνος για την πλήρη υλοποίηση όλων των έξτρα λειτουργειών και έτσι κάποιες έξτρα λειτουργείες είναι απλά σκελετοί πάνω στους οποίους μπορούν να χτιστούν σε μεταγενέστερο χρόνο.

Βάση Δεδομένων



Η βάση δεδομένων είναι πολύ σημαντική για κάθε project καθώς μια σωστή υλοποίηση μειώνει αρκετά την πολυπλοκότητα τις διασύνδεσής της με την εφαρμογή. Οι βασικές οντότητες της βάσης είναι:

GAME: πίνακας του παιχνιδιού ο οποίος κρατάει την κατάσταση του, την θέση την κάθε χρονική στιγμή, τις κινήσεις που έχουν γίνει και την ημέρα δημιουργίας του παιχνιδιού.

PLAYER: οντότητα του παίχτη όπου κρατάει τα στοιχεία του.

RANKING: περιέχει όλες τις βαθμολογίες του χρήστη σε κάθε κατηγορία.

PUZZLE: το σύνολο των puzzles και τι βαθμολογία έχει το κάθε ένα.

Όπως φαίνεται και στο διάγραμμα, υπάρχουν έξτρα πίνακες στη βάση οι οποίοι συνδέουν τις οντότητες. Κάθε GAME το οποίο είναι ενεργό αποτελείται από 2 χρήστες. Οι αυτοσυσχετίσεις που έχει ο PLAYER αφοράνε τους φίλους, τις προσκλήσεις για φίλους και ο challenge για την δημιουργία κρυφού παιχνιδιού.

Η βάση δεδομένων έχει αποθηκευτεί στο Heroku και το αρχείο chess_db περιέχει τον κώδικα σε sql για την δημιουργία του. Η σύνδεση της εφαρμογής με την βάση επιτεύχθηκε με την βοήθεια της βιβλιοθήκης για postgress ‘pg’ της npm.

Μεθοδολογία

Ανάπτυξη του αντικειμένου Board

Όλη η υλοποίηση του παιχνιδιού σκάκι έγινε μέσα από το script Board.js το οποίο περιέχει το ομώνυμο αντικείμενο Board. Το αντικείμενο αυτό περιγράφει τους κανόνες του παιχνιδιού και κατασκευάζει την γραφική αναπαράσταση του. Τα βασικά δομικά στοιχεία του Board είναι τα αντικείμενα Piece και Move.

Το αντικείμενο Piece αποτελεί την βάση για την κατασκευή κάθε διαφορετικού πιονιού. Τα κοινά χαρακτηριστικά που περιγράφουν κάθε πιόνι της σκακιέρας είναι η θέση και το χρώμα. Επεκτείνοντας πάνω στην κοινή βάση κατασκευάζονται τα Pawn, Rook, Knight, Bishop, Queen και King. Κάθε ένα από αυτά τα αντικείμενα περιγράφουν το αντίστοιχο πιόνι. Ειδικότερα, τα αντικείμενα αυτά έχουν μία μέθοδο move() που επιβάλει την ορθότητα των κινήσεων για κάθε ένα με βάση τους κανόνες του παιχνιδιού. Επίσης, κάθε στιγμιότυπο των αντικειμένων αυτών περιέχει και διεύθυνση στην εικόνα με την οποία παρουσιάζεται πάνω στην σκακιέρα.

Όλα τα Piece αποθηκεύονται σε ένα αντικείμενο Pieceset το οποίο περιέχει μερικές αναφορές για την τοποθεσία των Pieces στην αρχική θέση και ένα array που περιέχει τα 32 πιόνια στην αρχική τους θέση.

Το αντικείμενο Move εκφράζει μία πιθανή κίνηση ενός Piece σε κάποια τοποθεσία της σκακιέρας. Αποτελείται από μία αναφορά σε αντικείμενο Piece και ένα αλφαριθμητικό που περιγράφει μία θέση στην σκακιέρα. Κατά την διάρκεια του παιχνιδιού κάθε επίσημη κίνηση προστίθεται σε αντικείμενο Moveset το οποίο περιέχει ένα array από Move.

Κατά την αρχικοποίηση του αντικειμένου Board προστίθενται σε αυτό ένα στιγμιότυπο Pieceset και ένα στιγμιότυπο Moveset. Αυτά, σε συνεργασία με τις μεθόδους του Board, περιγράφουν την κατάσταση του παιχνιδιού σε κάθε στιγμή , για την ομαλή λειτουργία του.

Μέθοδοι:

Initialize

Το αντικείμενο Board περιέχει πολλές μεθόδους για την εκτέλεση της σχεδιασμένης λειτουργείας. Η πιο βασική είναι η μέθοδος initialize. Η μέθοδος initialize εκτελεί την αρχικοποίησή του Board. Πρώτα, ζητά από τον εξυπηρετητή το χρώμα του παίκτη μέσω ενός fetch request. Ανάλογα με το χρώμα κατασκευάζει το grid από divisions που αντιστοιχούν στην σκακιέρα και τα εμφανίζει στην σελίδα. Στα divisions αυτά προστίθεται ένας EventListener για να μπορούν να εντοπιστούν τα click στην σκακιέρα. Αφού ολοκληρωθεί η τοποθέτηση των division της σκακιέρας τοποθετούνται και τα πιόνια. Η θέση αυτών έχει αντιστοίχιση με το id των divisions. Με την ολοκλήρωση της κατασκευής της σκακιέρας και την τοποθέτηση των πιονιών όλες η περεταίρω ενέργειες είναι αποτέλεσμα του EventListener κάθε division.

InitializeSolo

Η initializeSolo είναι μία διαφοροποίηση της initialize που χρησιμοποιείται για την αρχικοποίηση Board που θα χρησιμοποιηθεί μόνο από ένα client. Παράδειγμα τέτοιας λειτουργείας είναι η σκακιέρα στο Academy - Self-Analyzing όπου ένας χρήστης μπορεί να κάνει διαδοχικά κινήσεις και για τα δύο χρώματα.

Initialize EventListener

Όταν αναγνωριστεί κάποιο κλικ σε σημείο της σκακιέρας ελέγχεται αν υπάρχει στο σημείο αυτό τοποθετημένο Piece, δηλαδή αν ο χρήστης επέλεξε ένα πιόνι να κινήσει. Αν επέλεξε

ένα αυτό τίθεται σαν επιλεγμένο πιόνι και αναμένεται η επιλογή τοποθεσίας. Όταν ο χρήστης επιλέξει με δεύτερο κλικ την τοποθεσία όπου θέλει να μετακινήσει το πιόνι καλείται η μέθοδος validate του Board. Η μέθοδος αυτή ελέγχει αν η κίνηση που επιλέχθηκε είναι δυνατή βάσει των κανόνων, η λειτουργεία της θα περιγραφεί παρακάτω αναλυτικά. Αν η κίνηση είναι δυνατή ελέγχονται αρχικά κάποιο ειδικοί κανόνες όπως το castling ή το promotion και στην συνέχεια εκτελείται η μεταφορά του πιονιού στην θέση που επιλέχθηκε. Αν στην θέση αυτή υπάρχει εχθρικό πιόνι αυτό αφαιρείται από την σκακιέρα. Τέλος, ελέγχεται η πιθανότητα checkmate ή stalemate με της ανάλογες μεθόδους πριν σταλθεί στον εξυπηρετητή η κίνηση με μέθοδο post και μπει το Board σε κατάσταση αναμονής για την κίνηση του αντιπάλου. Η κατάσταση αυτή είναι ένα συχνά επαναλαμβανόμενο fetch request στον εξυπηρετητή για την επόμενη κατάσταση την οποία καθορίζει ο άλλος χρήστης. Η διαδικασία αυτή επαναλαμβάνεται μετά την επιστροφή του fetch request για την κίνηση του αντιπάλου.

Validate

Η μέθοδος validate που προαναφέρθηκε ελέγχει όλους τους κανόνες για την επικύρωση την επιλεγμένης κίνησης. Αρχικά, καλεί την μέθοδο isPlayersTurn η οποία συγκρίνει το χρώμα του παίκτη με το χρώμα που πρέπει να κάνει την επόμενη κίνηση. Έπειτα, ελέγχει αν το πιόνι που επιλέχθηκε έχει το σωστό χρώμα με την μέθοδο isColorsTurn. Ακόμα, ελέγχει αν η κίνηση αυτή μπορεί να γίνει από το πιόνι με την μέθοδο move του Piece. Αν είναι νόμιμη κίνηση εξετάζεται αν υπάρχει στην διαδρομή κάποιο άλλο πιόνι που να εμποδίζει αυτήν την κίνηση με την μέθοδο notBlocked. Τέλος, ελέγχεται αν ο βασιλιάς απειλείται με την μέθοδο isUnderCheck. Αν τηρούνται όλες η συνθήκες η μέθοδος επιστρέφει true αφού πρώτα προσθέσει την κίνηση στο Moveset και αλλάξει το χρώμα που παίζει με την μέθοδο colourChange.

isPlayersTurn , isColorsTurn

Οι δύο μέθοδοι πραγματοποιούν ένα πολύ απλό έλεγχο ισότητας μεταξύ του χρώματος του παίκτη και της αναμενόμενης κίνησης και του χρώματος του πιονιού και της αναμενόμενης κίνησης αντίστοιχα.

notBlocked, notBlockedCheck

Η μέθοδος notBlocked ελέγχει για πιόνια στην διαδρομή προς τον προορισμό. Ο τρόπος με τον οποίο ελέγχονται τα πιόνια εξαρτάται από το είδος τους. Η μέθοδος διαφοροποιεί την διαδικασία υπολογισμού των ενδιάμεσων τετραγώνων ανάλογα με τους κανόνες κίνησης του πιονιού. Η μέθοδος notBlockedCheck αποτελεί μία διαφοροποίηση της notBlocked η οποία δεν ελέγχει για πιόνι στον προορισμό το οποίο προϋπήρχε εκεί. Η μέθοδος αυτή χρησιμεύει για τον έλεγχο κίνησης του βασιλιά όταν απειλείται. Όταν ο βασιλείας προσπαθεί να κινηθεί σε τετράγωνο που υπάρχει πιόνι που τον απειλεί πρέπει να μπορεί να ελεγχθεί αν άλλα εχθρικά πιόνια μπορούν να μεταβούν στην νέα θέση του. Η ύπαρξη το φιλικού πιονιού στην θέση αυτή δεν θα επέστρεφε την σωστή τιμή με την μέθοδο notBlocked για αυτό κατασκευάστηκε η notBlockedCheck.

isUnderCheck

Η μέθοδος isUnderCheck ελέγχει αν ο βασιλιάς απειλείται. Αν πράγματι υπάρχει κάποιο εχθρικό πιόνι το οποίο μπορεί μετακινηθεί στην θέση του βασιλιά (Piece.move και notBlockedCheck) τότε οι μόνες κινήσεις που επιτρέπονται είναι αυτές που είτε απομακρύνουν τον βασιλιά από την απειλή, είτε εξουδετερώνουν την απειλή, είτε εμποδίζουν την απειλή.

colourChange

Η μέθοδος colourChange αλλάζει το αναμενόμενο χρώμα για την επόμενη κίνηση.

stalemate, checkmate, draw

Μετά από κάθε κίνηση ελέγχεται αν το παιχνίδι έχει φτιάσει σε κατάσταση statemate ή checkmate . Η μέθοδος stalemate ελέγχει αν οποιοδήποτε από τα υπολειπόμενα pieces μπορεί να πραγματοποιήσει οποιαδήποτε κίνηση. Η μέθοδος checkmate ελέγχει αν υπάρχει οποιαδήποτε κίνηση των φιλικών πιονιών που να λύνει την κατάσταση απειλής του βασιλιά. Η μέθοδος draw ελέγχει πόσα piece υπάρχουν στο board για να τηρείται το sufficient material condition.

promote

Η μέθοδος promote καλείται όταν ένα πιόνι φτάσει στην κατάλληλη θέση και δημιουργεί ένα κατάλληλο μενού επιλογής πιονιού για να εκτελεστεί το promotion. Αφού επιλεχθεί το είδος πιονιού κατασκευάζεται η νέα κατάσταση της σκακιέρας και στέλνεται στον αντίπαλο

transformToForsythEdwardsNotation, extractPosArrayFromForsythEdwardsNotation

Η Forsyth–Edwards Notation είναι μία συνήθης γραφή της κατάστασης μίας σκακιέρας. Η γραφή αυτή παρουσιάζει όλα τα πιόνια και της θέσεις του με ένα αλφαριθμητικό π.χ. rnbqkbnr/pppppppp/8/8/8/8/PPPPPPPP/RNBQKBNR όπου κάθε τμήμα μεταξύ «/» είναι μία γραμμή της σκακιέρας ,ξεκινώντας από την γραμμή οκτώ μέχρι την πρώτη, και τα σύμβολα αντιπροσωπεύουν το πιόνι. Με κεφαλαίο τα λευκά και με πεζό τα μαύρα. Τα σύμβολα αντιπροσωπεύουν Rook, Knight, Bishop, Queen, King, Pawn αντίστοιχα. Οι αριθμοί περιγράφουν τον αριθμό κενών τετραγώνων.

Το notation αυτό χρησιμοποιήθηκε για ταχεία μεταφορά της κατάστασης στον εξυπηρετητή και για οικονομία χώρου στην βάση. Οι μέθοδοι αυτοί μετατρέπουν από την κατάσταση του Pieceset προς Forsyth–Edwards Notation και αντίστροφα πριν την αποστολή κατάστασης και μετά από την παραλαβή της επόμενης κατάστασης.

clearBoard, setPosition ,setPositionInactive

Οι μέθοδοι clearBoard, setPosition ,setPositionInactive αποτελούν τις μεθόδους επεξεργασίας απεικόνισης της κατάστασης της σκακιέρας. Η clearBoard αφαιρεί όλα τα πιόνια από την σελίδα και χρησιμοποιείται με κάθε κλήση της setPosition για να καθαρίσει την σκακιέρα πριν τοποθετηθεί η επόμενη κατάσταση. Η μέθοδος setPosition λαμβάνει ένα αλφαριθμητικό Forsyth–Edwards Notation το οποίο μετατρέπει σε pieceset με τις προαναφερθείσες μεθόδους και τοποθετεί τα πιόνια στις θέσεις που περιγράφονται. Τέλος, η setPositionInactive αποτελεί μία μέθοδο αρχικοποίησης σκακιέρας που δεν θα αλλάξει η θέση της π.χ. οι εικόνες στο Academy και κατασκευάζει την εμφάνιση της σκακιέρας χωρίς την λειτουργικότητα και προσθέτει τα πιόνια στις θέσεις που περιγράφονται από το αλφαριθμητικό εισόδου.

Server

Το μοντέλο του server χωρίστηκε σε κομμάτια ώστε να γίνεται καλύτερα η διαφοροποίηση των λειτουργιών του. Όλα τα dependences βρίσκονται στο αρχείο package.json ενώ η έναρξη του server γίνεται με το αρχείο start.js. Η διαδρομή που ακολουθεί κάποιο αίτημα είναι

1. Περνά στο app.js το οποίο συνέχεια ακούει στο port της εφαρμογής.
2. Από εκεί περνά στο router.js όπου γίνεται η δρομολόγηση του αιτήματος
3. Το αίτημα ύστερα περνά στον controller.
4. Αν χρειαστεί ο controller καλεί με την σειρά του τις μεθόδους του model το οποίο κάνει την σύνδεση με την βάση και εξάγει τα κατάλληλα δεδομένα.
5. Ακολουθείται η αντίστροφη διαδρομή μέχρι τον controller όπου στέλνει το response στον client

Η αρχικοποίηση του server γίνεται στο αρχείο app.js το οποίο φορτώνει όλες τις βιβλιοθήκες που χρειάζονται για την εφαρμογή . Επίσης για κάθε χρήστη που συνδέεται ξεκινά ένα session και στέλνει ένα cookie με το οποίο και αναγνωρίζει τα αιτήματα του χρήστη αυτού. Θέτει επίσης την render engine ώστε να αναγνωρίζει τα handlebars με το επιλεγμένο extension. Τέλος, μεταφέρει κάθε request που έρχεται στο αρχείο routes.js από οπού διαχωρίζονται τα αιτήματα του client.

Ο routes.js για την ορθή επεξεργασία του αιτήματος χρησιμοποιεί middlewares. Το πρώτο middleware σε κάθε σελίδα ελέγχει αν ο client έχει συνδεθεί. Τα επόμενα εκτελούν τις ανάλογες πράξεις για να παράξουν το response object που θα σταλθεί στον client.

Τα αιτήματα χωρίζονται σε δύο κατηγορίες, στα αιτήματα για login και στα αιτήματα για το παιχνίδι. Για αυτό το λόγο έχουν δημιουργηθεί 2 controllers αντίστοιχα. Ο controller για το login είναι υπεύθυνος να αναγνωρίζει αν ο χρήστης είναι συνδεδεμένος ή αν συνδεθεί εκείνη την ώρα να ανανεώνει τις μεταβλητές του session. Παρομοίως πράττει και σε περίπτωση που γίνει register κάποιος καινούργιος χρήστης όπου στέλνει αίτημα και στο μοντέλο της βάσης. Ο chess-controller από την άλλη έχει 2 λειτουργίες. Η μια είναι να φορτώνει στον client τις διάφορες σελίδες και η δεύτερη ύστερα από επικοινωνία με το μοντέλο της βάσης να επιστρέφει κατάλληλα response στον χρήστη.

Το μοντέλο της βάσης δεδομένων μάς είναι η βασική διεπαφή που έχουμε με την βάση. Η σύνδεση γίνεται με βάση τα χαρακτηριστικά που υπάρχουν στο script db.heroku-pg.js. Στο chess-model-heroku-pg βρίσκονται όλες οι συναρτήσεις που αλληλοεπιδρούν με τη βάση. Οι στόχοι επικοινωνίας είναι εγγραφή, τροποποίηση, διαγραφή και επιλογή κάποιων στοιχείων από την βάση.

Οι σελίδες που στέλνονται στο χρήστη είναι παραγόμενες από την μηχανή render που συνδέει τα διάφορα partial handlebars, από τον φάκελο views, και κατασκευάζει την κάθε ζητούμενη σελίδα. Για κάθε σελίδα έχουν γραφεί και χρησιμοποιηθεί πληθώρα script και stylesheet. Κάποια από αυτά κοινά με άλλες σελίδες και κάποια μοναδικά για αυτήν. Για να αποφευχθεί η συνεχείς επανάληψη του head με τα κοινά resources, αυτά έχουν προστεθεί σε ένα φάκελο public από όπου η μηχανή handlebars τα χρησιμοποιεί ως static αρχεία.

Οι σελίδες που φορτώνονται στον client βρίσκονται στον φάκελο views και είναι σε μορφή handlebars ενώ τα αρχεία CSS, JavaScript, json και png που χρειάζονται για να λειτουργήσουν βρίσκονται στον φάκελο public οργανωμένα σε υποφακέλους.

Αξιολόγηση

Η ανάλυση του προβλήματος χώρισε την υλοποίηση του προβλήματος σε τρία διασυνδεόμενα μέρη. Αρχικά, είναι προφανής η ανάγκη scripts που εκτελούνται στον χρήστη για την επικύρωση των ενεργειών του παιχνιδιού. Αναγκαία είναι επίσης η ύπαρξη μόνιμου και αξιόπιστου εξυπηρετητή καθώς το παιχνίδι βασίζεται σε επικοινωνία δύο χρηστών που δεν έχουν γνώση ο ένας για τον άλλο. Τέλος, για την αποθήκευση του δεδομένων είναι πολύ χρήσιμη μία καλά ορισμένη βάση η οποία να έχει και δυνατότητα επέκτασης σε περίπτωση γενίκευσης της εφαρμογής.

Όσον αφορά την client side υλοποίηση εκτιμούμε ότι έχουμε υλοποιήσει ένα σύνολο εργαλείων που αντιμετωπίζουν το πλείστο των απαιτήσεων. Η απόδοση τους μετά από εκτενής χρήση δεν φαίνεται να παρουσιάζει επιβάρυνση στην ταχύτητα πλοήγησης του χρήστη. Ακόμη, το πρωτότυπο σύστημα αντικειμένων που κατασκευάστηκε έχει ειδίκευση για διαφορετικές εφαρμογές σαν προσπάθεια βελτιστοποίησης της εφαρμογής. Ακόμη, έχουν γίνει ενέργειες στον ορισμό όλων των αντικειμένων ώστε να υπάρχουν εργαλεία για επέκταση.

Στην μεριά του εξυπηρετητή έχουμε υλοποιήσει όλες τις λειτουργίες ώστε να επιτυγχάνεται η σύνδεση των δύο χρηστών χωρίς προβλήματα και καλύπτοντας πιθανά λάθη του χρήστη (πχ εκτελώντας κίνηση πριν παίξει ο αντίπαλος). Η σύνδεση με την βάση καλύπτει όλες τις ανάγκες και ανανεώνεται αυτόματα με κάθε κίνηση του client. Οι έξτρα υλοποιήσεις που προσπαθήσαμε να κάνουμε είναι το σημείο που υστερεί η εφαρμογή καθώς δεν είναι ολοκληρωμένες λόγω έλλειψης χρόνου.

Τέλος, πιστεύουμε ότι έχει επιτευχθεί λύση του προβλήματος δύο παικτών καθώς αντιμετωπίστηκαν τα προβλήματα προγραμματισμού κανόνων παιχνιδιού, σύνδεσης δύο χρηστών χωρίς αυτοί να έχουν γνώση ο ένας για τον άλλο, ασύγχρονη μεταφορά δεδομένων μεταξύ χρηστών, ορθή σύνδεση και ανανέωση με την βάση και «κλείδωμα» της κατάστασης καθώς αναμένεται απάντηση.

Πέρα από το βασικό πρόβλημα η προσπάθεια γενίκευσης που επιχειρήσαμε παρέμεινε στα αρχικά στάδια κατασκευής του βασικού σκελετού της εφαρμογής. Πιστεύουμε ότι είναι μία καλή αρχή για να αντιμετωπιστεί το πρόβλημα της γενίκευσης αλλά λόγω έλλειψής χρόνου δεν ήταν εφικτό να την τελειώσουμε. Καταφέραμε όμως να προετοιμάσουμε την βάση για υποστήριξη τέτοιων λειτουργειών και να κατασκευάσουμε ένα στατικό κομμάτι αυτών για την παρουσίαση του σχεδιασμού μας.

Δεδομένα

Στην υλοποίηση της εφαρμογής χρειάστηκαν κάποια δεδομένα τα οποία πήραμε από διάφορες πηγές. Για τα openings και τα endings οι πηγές από τις οποίες πήραμε τα δεδομένα ήταν οι:

- https://en.wikipedia.org/wiki/Chess_opening
- https://www.chess.com/openings
- https://en.wikipedia.org/wiki/Chess_endgame

Επιπροσθέτως, χρησιμοποιήσαμε εικόνες για τα πιόνια της σκακιέρας και icons για την κάθετη μπάρα επιλογών. Οι σύνδεσμοι φαίνονται παρακάτω

- https://commons.wikimedia.org/wiki/Category:PNG_chess_pieces/Standard_transparent

- https://fontawesome.com/v5.15/icons?d=gallery&p=2

Κύριες Ενέργειες

Ενέργειες	**Άτομο**
Κατασκευή Σκελετού Ιστοσελίδας	Γρυπάρης Ιάσονας
Καλλωπισμός Ιστοσελίδας	Γρυπάρης Ιάσονας, Λαμπίδης Λουκάς
Δημιουργία Βάσης	Λαμπίδης Λουκάς
Κατασκευή chatbox	Γρυπάρης Ιάσονας
Κατασκευή Server	Λαμπίδης Λουκάς
Κατασκευή board object	Γρυπάρης Ιάσονας
Διασύνδεση Βάσης-Server-Client	Λαμπίδης Λουκάς
Διασύνδεση Server-Client	Γρυπάρης Ιάσονας, Λαμπίδης Λουκάς

Χρονοδιάγραμμα

Βήμα	**Χρονική Περίοδος**
Κατασκευή Σκελετού Ιστοσελίδας	6 Απριλίου – 20 Απριλίου
Καλλωπισμός Ιστοσελίδας	6 Απριλίου - 6 Ιουνίου
Δημιουργία Βάσης	1 Μαΐου – 30 Μαΐου
Κατασκευή chatbox	25 Απριλίου – 20 Μαΐου
Κατασκευή Server	20 Μαΐου - 6 Ιουνίου
Κατασκευή board object	10 Μαΐου - 6 Ιουνίου
Διασύνδεση Βάσης-Server-Client	20 Μαΐου - 6 Ιουνίου
Διασύνδεση Server-Client	25 Μαΐου - 6 Ιουνίου
Σύνταξη έκθεσης	4 Ιουνίου – 6 Ιουνίου

Οδηγίες χρήσης

Μεταβαίνοντας στην εφαρμογή ο χρήστης ζητείται να συνδεθεί σε λογαριασμό ή να δημιουργήσει καινούργιο. Για την δημιουργία λογαριασμού ο χρήστης χρειάζεται να πληκτρολογήσει ένα όνομα χρήστη, ένα κωδικό και μία διεύθυνση ηλεκτρονικού ταχυδρομείου. Για να συνδεθεί στο λογαριασμό του πληκτρολογεί το όνομα χρήστη του και τον κωδικό του.

Με την επιτυχή σύνδεση ο χρήστης ανακατευθύνεται στην αρχική σελίδα. Στην σελίδα αυτή ο χρήστης έχει πρόσβαση στα ενεργά παιχνίδια του τα οποία παρουσιάζονται στην οριζόντια μπάρα καθώς και στο ημερήσιο puzzle ακριβώς από κάτω. Μέσω στην πλαϊνής μπάρας στο αριστερό μέρος της οθόνης ο χρήστης μπορεί να μεταβεί στην σελίδα για νέο παιχνίδι, στα puzzle, στο academy, προσθήκη φίλου και στις ρυθμίσεις του, δίνεται ακόμη η δυνατότητα να αλλάξει τη χρωματική παλέτα της εφαρμογής .

Για την εύρεση καινούργιου παιχνιδιού ο χρήστης χρειάζεται να πατήσει το κουμπί “Search for a game” στην σελίδα που ανακατευθύνεται και να περιμένει μέχρι να βρεθεί αντίπαλος. Όταν βρεθεί αντίπαλος θα εμφανιστεί το κουμπί “Join”. Πατώντας το κουμπί αυτό ανακατευθύνεται στην σελίδα του

παιχνιδιού. Για να παίξει το παιχνίδι χρειάζεται να πατήσει πάνω στο πιόνι που επιθυμεί να κινήσει και στην συνέχεια να πατήσει στο σημείο που θέλει να το μετακινήσει. Ύστερα, αναμένει για την κίνηση του αντιπάλου και επαναλαμβάνει τα ίδια βήματα μέχρι το τέλος του παιχνιδιού.

Για την είσοδο στο Academy ο χρήστης πατά το κουμπί στην μπάρα στο αριστερό μέρος της οθόνης. Με την ενέργεια αυτή ανακατευθύνεται στο Academy με πρώτη σελίδα που παρουσιάζεται την σελίδα με τις στρατηγικές ανοιγμάτων. Σε αυτήν την σελίδα ο χρήστης μπορεί να δει παραδείγματα διάσημων στρατηγικών ανοίγματος καθώς και να διαβάσει μερικά λόγια για την λογική πίσω από το καθένα. Στο πάνω μέρος της σελίδας εμφανίζονται τρεις επιλογές για το χρήστη. Η πρώτη είναι όπως αναφέρθηκε η στρατηγική ανοίγματος. Η δεύτερη “Self-Analyzing” παρέχει στο χρήστη την δυνατότητα να χρησιμοποιήσει μία σκακιέρα όπου μπορεί να κάνει κινήσεις και για τους δύο παίκτες. Με την χρήση αυτού ο χρήστης μπορεί να μελετήσει διάσημες παρτίδες ή ακόμα και τις δικές του για να εντοπίσει τα λάθη του. Η τρίτη επιλογή “Endings” παρουσιάζει στον χρήστη στρατηγικές κλεισίματος του παιχνιδιού.

Πατώντας πάνω στον χρήστη στη οριζόντια μπάρα στην αρχική οθόνη ο χρήστης μπορεί να μεταβεί στο προφίλ του όπου μπορεί δει στατιστικά στοιχεία για τα παιχνίδια του.

Οι επιλογές Puzzles , Add Friend και Settings δεν έχουν υλοποιηθεί ακόμη και σε σκελετική φάση όπως το υπόλοιπο όμως κατά την μελέτη είχε καταγραφή η ιδέα σχεδιασμού. Για τα Puzzles είχε σχεδιαστεί διεπαφή παρόμοια με αυτή του Academy όπου ο χρήστης θα μπορούσε να επιλέξει την κατηγορία puzzle που θα ήθελε. Για τα Settings ο χρήστης θα μπορούσε να αλλάξει κάποιες βασικές ρυθμίσεις του λογαριασμού του με την συνηθισμένη μορφή πάνελ. ,ενώ στο Add Friend θα πρόσθετε φίλους ώστε να πραγματοποιούνται παρτίδες μεταξύ τους