

SE 210: Software Specification and Design I

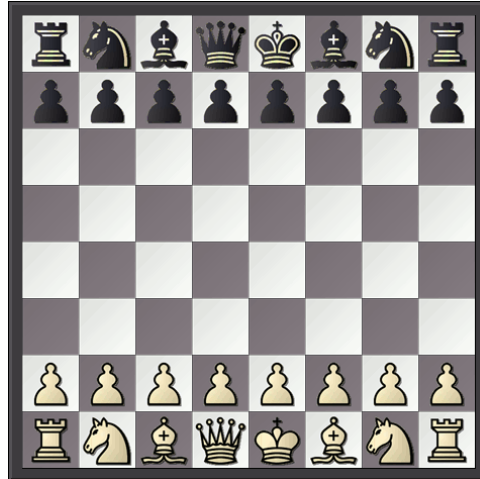
OnlineChess

Problem Analysis and Early Requirements

12 October 2016

Introduction:

OnlineChess is an online chess playing platform that allows chess players to square off in friendly or competitive games through the web browser. An online chess game is played between two players, either a computer or another user, on opposite sides of a chess board containing 64 squares of alternating colors. Each player has 16 pieces: 1 king, 1 queen, 2 rooks, 2 bishops, 2 knights, and 8 pawns. A reference to the layout of a chess board is listed below.



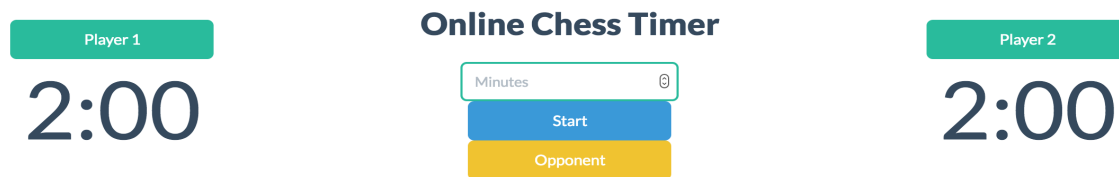
(Image of a Chessboard with corresponding pieces)

The goal of a game of *OnlineChess* is to checkmate the other king. Checkmate happens when the king is in a position to be captured by the other opponent and cannot escape from capture. Note that you cannot put your king in a position to be checked. Play continues until a King is checkmated or a draw occurs. The scenarios of a draw are the following.

1. Stalemate
The king is not in check but cannot move anywhere, and there are no other pieces left
2. Threefold repetition
A position has been repeated three times, such as players moving pieces back and forth
3. Mutual Agreement
A player may offer a draw to his opponent at any stage of a game. If the opponent accepts, the game is a draw.

Starting a Game:

Upon Starting a game, the user will be prompted to select an opponent and a time range of each turn. Default time range is two minutes. If a computer opponent is selected the User will have the ability to select a difficulty of either Easy, Medium, or Hard. An *OnlineChess* match will then be created. Each player will have the time limit selected to make a move.



(Example Timer for *OnlineChess*)

OnlineChess will randomly select each player's chess piece colors at the beginning of a match. The player with white chess pieces moves first. If a User is playing against a computer opponent, there is an option to save the current match to play at a later time or date. A user can have up to 3 saved progress matches to which they can play from. Play continues til there is a winner selected, or a User concedes or forfeits the match.

Important *OnlineChess* Definitions:

Check: To make a move that puts the opponent's King under direct attack.

Checkmate: a situation in which an opponent's king is in check and it cannot avoid being captured. This then brings the game to a victorious result.

En Passant: a method by which a pawn that is moved two squares can be captured by an opponent's pawn commanding the square that was passed

Stalemate: A situation in which a player's king is not in check, but that player can make no move. This then results is a stalemate, which is a draw.

Promotion: When a pawn reaches the other end of the board it can be changed for any other piece of its own colour, except the King

Castling: is a move in the game of chess involving a player's king and either of the player's original rooks. It is the only move in chess in which a player moves two pieces in the same move, and it is the only move aside from the knight's move where a piece can jump over another.

Chess Pieces and Their Moves:

Pawns: Pawns can only move forward. On their first move, they can move one or two squares.

Afterwards, they can move only one square at a time. They can capture an enemy piece by moving one square forward diagonally.

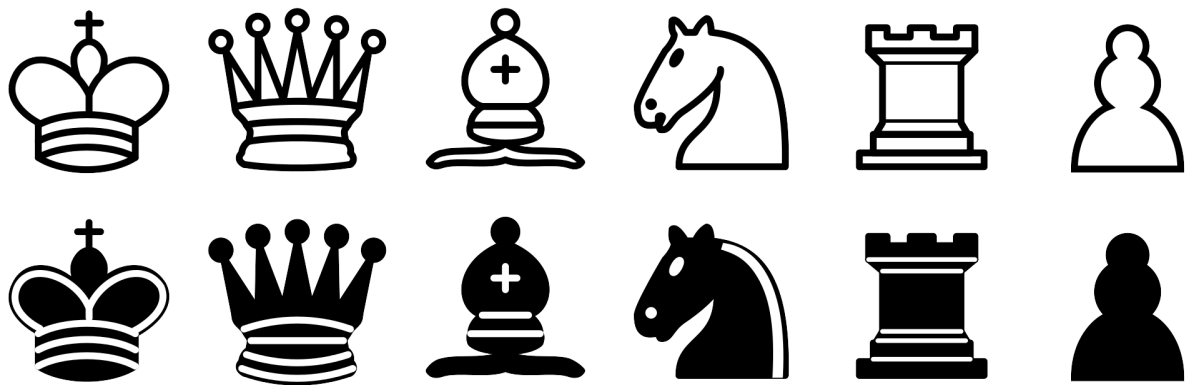
Bishops: Bishops can move any number of squares diagonally.

Knights: Knights can move only in an L-shape, one square up and two over, or two squares over and one down, or any such combination of one-two or two-one movements in any direction.

Rooks: Rooks can move any number of squares, up and down and side to side.

Queens: Queens can move any number of squares along ranks, files and diagonals.

Kings: Kings can move one square at a time in any direction.



King Queen Bishop Knight Rook Pawn

Application Structure - High Level Arch.:

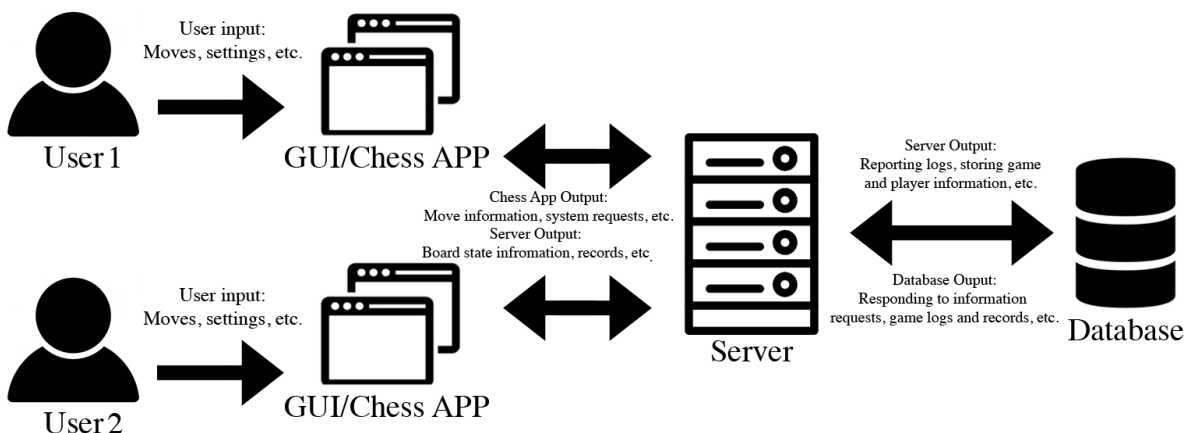
The software will be divided into two components: client side and server side. The objective is to facilitate functional communication across clients as to enable a game of chess. The program will enable the creation of games across separate computers as well as against a bot.

Client Side:

The user will interface with a web GUI which interfaces with the program talking to the server hosting the chess program. A home screen will be the face of the program and allow a user to start: an offline game, random online game, request another account to play, or continue existing games. In addition, a user may send a request to replay moves of a past game. Once a game mode or reload request has been processed, the user is directed to a chess board interface. Either chosen or randomly decided, the user will appear in the back of either a set of black or white pieces. The user will have control of piece moves during their respective turns and can assign a preemptive move during their opponent's turn. At any point, a piece will highlight all possible legal moves on the board. Once a game has been decided ended or paused, the user is directed back to the home screen.

Server Side:

The server will contact the database at select instances throughout the running of the software. From the home screen the server will aggregate: the current paused games of the user, all past games of the user, the user's rank, and the available players. When a match is requested the server will create a game with the appropriate opponent for the user based on the game type. The server will check for the correctness of moves and determine the state of the game as either: unstarted, active, paused, or ended. After each move a record will be sent and stored in a database. If a request to the home screen has been requested the server will store current values of the game and aggregate all the necessary information for the home screen.



Basic Architecture:

Operating Environment:

The *OnlineChess* client operating environment should be the web browser. This will allow anyone who visits the web site to play. The end user should be able to use all of the site's features as long as he or she has a modern web browser. That includes browsers like Firefox, Google Chrome, Safari, and Opera. In order to make mobile usage easier, a mobile optimized version of the site should be made. In order to simplify login, OAuth should be used to allow users to log in with their Google or Facebook. That way, users do not need to remember their passwords to use the site.

There are a number of features needed for the server side to make it fit well into the operating environment of the customer. The customer owns several servers, which are used for hosting its sites. Since some of the servers are Windows, and some of them are Linux, the *OnlineChess* server application should be platform independent, written in a language that supports both Windows and Linux operating systems. Otherwise, the customer would be limited in where the server application could be run. Since the customer buying this software uses Oracle databases in many of its business functions, the server side application should be able to use the Oracle database as a backend. This will allow for easier system administration, since the customer already employs Oracle database experts.

Development Environment:

The development environment will support the development of the online Chess program as a web application. We are expecting to support a development team of at least 5 developers/testers.

The environment will support the development, test and integration of all components of the application.

The supported user platforms will be any of the following web browsers: Google Chrome, Safari, Opera and Firefox.

The application will be tested and be fully functional and compatible across all supported browsers (Google Chrome, Safari, Opera and Firefox).

Each developer will have an iMac computer for development The iMac was selected because it has a rich development environment for web applications and built in Unix shell. A Windows PC will also be part of the environment for compatibility testing in a Windows environment.

Any of the computers can function as web servers for integration testing.

Source code control will be managed using "Git".

Problem and issue tracking will be managed using Atlassian Jira.

Hardware:

5 - iMacs (iMac 27" 4.0 GHz quad core, 1TB HDD)

1 - File Server (2 TB) (to hold shared files, Git repository, Jira)

1 - Windows PC (for testing on Chrome/Safari/Firefox/Opera hosted on Windows)

High Speed internet access for all computers.

Software:

Web Browsers: Chrome, Safari and Firefox

JavaScript Frameworks: Angularjs, react

Web Server: Apache

Source Code Control: Git

Issue Management: Atlassian Jira

Editors: Wrangler

Office Suite: Microsoft Office Professional Edition

Capabilities:

The user interface should be intuitive to use and familiar to the average chess player. In order to accomplish this, the client application should use conventions, like moving the white king to G2 to castle kingside. To enhance usability, the user should be able to drag and drop pieces to make moves, and submit moves without having to refresh the web page. A frontend framework like AngularJS or React should be used to accomplish this. There is currently a wide variety of mobile chess apps, so a mobile optimized version of the site should be made. A “mobile optimized” version of the site will make various tradeoffs to increase usability. In a mobile optimized version, the pieces can be controlled with taps rather than dragging and dropping, and in a “mobile optimized” version, the board will take up a greater portion of the screen.

Both the client side and the server side should enforce the standard chess rules on the end users. The client-side safeguards should help the users to know which moves are legal and illegal, and the server-side safeguards should prevent the clients from cheating on each other. The server side will need to handle the player matchup functionality so that, when users are entering into games with non-friends, they are matched with players of similar skill levels. There are a number of different chess ranking systems, but the most commonly used one is the Elo rating system. Chess.com, for example, uses the Elo rating system to set up matches.

Conclusion:

This project has collectively enhanced skills among individuals comprised in this team. The primary take away of this project is the understanding of technical documentation and the processes behind it. This *OnlineChess* project taught many members of the team how to research technically. Whether it was experiencing a couple of online chess games or polling friends for in-game advice, we all gained new knowledge of chess and as such wrote accurately. Following this, requirements defining was integral to our overall goal. After laying out all the initial requirements, the team divided the assignment into respective sections to optimize our efficiency. As such, each member specialized in a section and stayed within line of the overall requirements. Several rounds of comments and critiques kept our sections sharp and overall quality high. All in all, our team started out as a bunch of Software Engineering students who had not known each other previously, and became a competent team capable of technical requirements writing.

Reference:

By Using the International Rule: How Many Queens Are in Board? "What Are the Few Rules of Chess Which Are Not Commonly Known?" - *Quora*. N.p., n.d. Web. 09 Oct. 2016.

Blksheep. "How to Play Chess." *Instructables.com*. N.p., 03 Mar. 2008. Web. 07 Oct. 2016.

"Rules of Chess." *Wikipedia*. Wikimedia Foundation, n.d. Web. 07 Oct. 2016.

"Chess.com." *How to Play Chess: Rules & Basics* -. N.p., n.d. Web. 07 Oct. 2016.

"Castling." *Wikipedia*. Wikimedia Foundation, n.d. Web. 09 Oct. 2016.

"The Best Free, Adless Chess Server." *Lichess.org*. N.p., n.d. Web. 09 Oct. 2016.