Project – Chess GUI with Engine  
Analysis  
Harley Reeves-Martin

Contents

[Introduction: 3](#_Toc528922795)

[Description of the current system: 4](#_Toc528922796)

[Identification of the problems with the current system: 5](#_Toc528922797)

[Problem analysis techniques: 6](#_Toc528922798)

[Identification of the prospective users: 7](#_Toc528922799)

[Users’ needs: 8](#_Toc528922800)

[Acceptable limitations: 9](#_Toc528922801)

[Numbered general & specific objectives: 10](#_Toc528922802)

[Analysis data dictionary: 11](#_Toc528922803)

[Proposed object diagrams: 12](#_Toc528922804)

[Justification of chosen solution: 13](#_Toc528922805)

# References

Arena. (n.d.). *Arena Chess GUI for Linux and Windows*. Retrieved from playwitharena: www.playwitharena.com

Collaborative. (n.d.). *ELO Rating System*. Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Elo\_rating\_system

Erik Allebest, J. S. (n.d.). *Chess: Home*. Retrieved from Chess: https://www.chess.com/

Fédération Internationale des Échecs. (n.d.). FIDE: Laws of Chess. Retrieved from FIDE.

FIDE. (n.d.). Retrieved from FIDE Online Arena: https://arena.myfide.net/

Stefan-Meyer Kahlen (Shredder Chess). (2004, April). *UCI Protocol*. Retrieved from http://wbec-ridderkerk.nl/html/UCIProtocol.html

1.0.0 Introduction:

Chess is a popular activity and game that is known by many. It is commonly associated with the ability to think logically and abstractly, however currently is not encouraged onto youths, as many would hope.

Chess is an ancient game traditionally played with a physical chess set. The fact that many people do not own a physical chess set as well as a relative lack of support on digital platforms, has led to the game being less accessible than other, similar games. My solution aims to provide an approachable but complete digital implementation of the game of chess that introduces an opponent capable enough to allow a beginner to learn, whilst also allowing more experienced players to use the application recreationally.

My solution will provide a free alternative for those who do not have access to a chess set, whilst also providing additional support (that is otherwise not possible in physical sets) for beginners. The engine will provide a capable opponent for beginners who would otherwise be without someone else to play with; however, the strength of this artificial intelligence will be limited. Despite this, I aim for the rest of my solution to be suitable for advanced players.

My program will also be able to be used as a teaching aid, and therefore could be used in either independent chess clubs or school chess programmes.

I will use information-gathering techniques such as interviews and surveys with my target audiences in order to guide the implementation of my solution. Particularly, I will be able to interview the chess club ran at my school, as well as other chess clubs in the area. Additionally, I will be able to interview peers of mine who do not play chess, which I believe will allow me to give the best possible support to those with little to no knowledge of the game. Furthermore, I will use current documents and personal reviews to determine the best possible approaches to individual solutions.

* + 1. Description of the current system:
    2. Intro to topic:

Currently, the most common method of playing chess is using a physical chess-set. This however provides no support to those who do not know the rules of the game and requires you to find a second player to play the game with. Furthermore, such chess-sets are relatively expensive, costing anywhere from £20 to £150+.

As of 13/11/18, there are digital alternatives to using a physical board. This includes software such as Arena Chess GUI, FIDE Online Arena, or Chess.com. Arena Chess GUI is a software designed to allow users to play against some of the best engines developed currently, and carry out analysis of games, whereas FIDE Online Arena and Chess.com both offer engine-based play and online, player-vs-player play.

* + 1. Table of Key Terms:

|  |  |
| --- | --- |
| Term: | Description: |
| Pawn | A pawn is a piece in the game that can move one square forward, or two squares forward if it is still in its starting position. Pawns will also “Queen” if reaching the opposite side of the board from their starting position. |
| Queen | A queen is a piece with the combined movements of the Bishop and Rook; it is the strongest piece on the board excluding the King. |
| Knight, Bishop, Rook, King | For further information on the movements of pieces in chess, please refer to (Fédération Internationale des Échecs, pp. 4, 5, and 6) |
| Ply/Half-Move | A ply is when a turn is taken by either of the players. |
| Move | A move is played when a ply is completed by both the white and black side sequentially[[1]](#footnote-1). |
| Depth | Depth describes the number of half-moves ahead the engine has analysed from the current position. |
| ELO | ELO is a rating system used in calculating the relative skill level of a player in zero sum games, in this case Chess. For further information, refer to (Collaborative, n.d.) |
| ECO | This is the encyclopaedia of chess openings. It describes the name of the opening and the move order required to get there. |
| FEN | Forsyth-Edwards notation is a format used to store the position of a chessboard as a string. |
| PGN | Portable game notation is a format used to store the move order, and optionally the analysis of each move, along with any meta-data (such as opening played, event title etc.) after a game has been played. |
| File | A file is a column on the chessboard; files are labelled “A” through to “H”. |
| Rank | A rank is a row on a chessboard; rows are labelled “1” through “8”. |
| Algebraic Chess Notation | Chess notation describes moves made on the board algebraically. Generally, moves are represented algebraically in the format of the initial of the piece followed by the target tiles file and then ranks. For example, “Qg4” indicates that a queen moved to g4. For further information, please refer to citation[[2]](#footnote-2). |
| Line | A line is a sequence of moves usually referred to when analysing a position. |
| Variation | A variation is an alternative line of play in the opening, for example, the Berlin Defence is a side line of the Ruy Lopez opening. |
| Engine | An engine is a piece of software that can look at a chess position and heuristically determine a “best” move. My implementation will also allow the engine to directly interact with the chess GUIs to allow for games to be played against the user. |
| Engine Evaluation | An engine evaluation is a measurement of advantage given to one side by the computer engine. Positive values indicate and advantage for white, negative values indicate an advantage for black, and a value of “0” indicates that the engine believes the position is drawn. If there is a forced line that leads to checkmate, the engine may show this as “#n”, or “#-n” where “n” is the number of plies until checkmate. |

In order to fully understand the topic that my solution will address, it will be beneficial to understand how to play Chess, and to understand the principles of operation of a Chess engine. Topics such as algebraic notation, portable game notation and analysis notation will be useful but not essential, since my solution itself aims to teach the users these topics if they do not already know them.

My overall solution is intended to be used by Chess players of all experience; however, my engine is only targeted at players who are learning or want to learn how to play the game. My main target audience will be those in education, however I believe that it will be useful for private chess clubs also.

* 1. Identification of the problems with the current system:

I have found the following issues with using a physical chess set:  
 1. Chess sets are expensive.  
 2. Pieces are typically fragile and easy to lose.  
 3. Two players are required in the same location to play.  
 4. Rules must be enforced by the player, and so illegal moves may go unnoticed.  
 5. To save a game you must create a paper copy, which is easy to do incorrectly.  
 6. Computer analysis is not possible.  
 7. To play timed chess, you also require a timer.  
 8. It is difficult to keep count of scores, and to generate and maintain ELO ratings.

Moreover, I have found the following problems regarding the previously referenced digital solutions:

|  |  |
| --- | --- |
| Solution | Problem |
| Arena Chess GUI  (Arena, n.d.) | Unable to play online. |
|  | Engines provided by the software are generally extremely strong players, much stronger than any beginner is. |
|  | The user interface lacks proper customizability. |
|  | The user interface expects prior knowledge of Chess and Chess Engines and so it is hard to use for beginners. |
|  | There is a lack of a rating system. |
|  | Rules are not always enforced, i.e. moves can be taken back, etc. |
| FIDE Online Arena (FIDE, n.d.) | Access to all features require you to subscribe to their service. |
|  | Without a subscription, you do not have access to an analysis board. |
|  | Without a subscription, the number of games you can play in a day is limited. |
|  | Without a subscription, participation in tournaments is limited. |
|  | Without a subscription, the applications interface is not customizable. |
| Chess.com  (Erik Allebest, n.d.) | Without a subscription, the users access to all tools of the website is limited (tactics, lessons and videos). |
|  | Without a subscription, the user cannot play multiple live games at once. |
|  | The website requires an internet connection to be able to be used. |
|  | The website can lead to safety issues when used by kids, due to the range of online forums and chat rooms provided by the service. |

By nature, my program will avoid all of the problems associated with using a physical chess-set, and I aim to implement all of the equivalent features as the identified current solutions, as well as other important features. Furthermore, my solution will be completely free to use, something that I believe to be particularly important due to the target audience of the solution.

1.3.0 Problem analysis techniques:

Interview(s):

In my interviews I asked a series of questions pertaining to the user experience and design of my program. I asked for suggestions as to what users would like to see in a chess program. Each interviewee’s self-described experience in chess is shown beside their name, I will use this to prioritize suggestions that I believe need to be made over trivial or less-important features.

| Who? | Suggestions: |
| --- | --- |
| Alex Patey  (Advanced) | **What would you like to see in a basic chess program?**  There should be more than just the regular chess variant, specifically I enjoy Bughouse and Crazyhouse.  **What are the important characteristics of a chess engine?**  The engine should be fast and should allow you to pick a level to play against. It should also be capable of beating stronger players. It should also play known openings to make the games realistic.  **What would you consider important about the user-interface of the solution?**  You could add indicators that show the current material difference in the game: so, if I capture a piece, there is some sort of visual indication that the piece has been taken from the board. You could also create an evaluation bar used to show who is currently winning the game.  **Would you need to use the program to learn how to play chess?**  No. |
| Oliver Stevens  (No Experience) | **What would you like to see in a basic chess program?**  You should be able to create timed games, including pre-set times. There should be a help page to refer to when playing that describes how pieces move etc.  **What are the important characteristics of a chess engine?**  Move generation must be fast and the engine must not be too difficult to win against.  **What changes would you like to see be made to the user-interface?**  You should make the interface interesting and allow the user to select graphics to be used.  **Would you need to use the program to learn how to play chess?**  Yes. |
| Jed Hamilton  (Beginner) | **What would you like to see in a basic chess program?**  Implement a feature that allows the player to get a hint on the best move from the computer, either by telling the user the best move, or by telling what piece the best move moves etc. There should also be a leader board feature that allows the users to compete.  **What are the important characteristics of a chess engine?**  Make the engine make random moves like a human.  **What would you consider important about the user-interface of the solution?**  The interface should be scalable so that it can be used on any sized screen comfortably.  **Would you need to use the program to learn how to play chess?**  Yes. |
| Craig Smith  (Beginner) | **What would you like to see in a basic chess program?**  Allow the user to make notes when playing a game. Allow games progress to be saved so you can come back to a game later if you do not want to finish it in one session.  **What are the important characteristics of a chess engine?**  I think it is important that you can change the difficulty of the engine, and it is also important that the engine operates quickly.  **What would you consider important about the user-interface of the solution?**  I would prefer the user-interface to be simplistic and clean, without unnecessary complications.  **Would you need to use the program to learn how to play chess?**  Yes. |
| Dan (Advanced) | **What would you like to see in a basic chess program?**  I think it is important that you can set up timed games, as well as organise matches for two people or the engine to play. For example, you should be able to set up a best of 3 match etc.  **What are the important characteristics of a chess engine?**  The engine should be quick to find a move and imitate a human player by making random choices between moves, as to reduce the predictableness of the engine.  **What would you consider important about the user-interface of the solution?**  You could add indicators that show the current material difference in the game: so, if I capture a piece, there is some sort of visual indication that –that piece has been taken from the board. You could also create a bar used to show who is currently winning the game.  **Would you need to use the program to learn how to play chess?**  No. |

Observations:

Other digital services allow for the user to create an account to save progress and allow the user to export games in PGN and FEN file formats.

Survey(s):

I will distribute surveys to members of my school’s chess club to gauge who would be interested in the solution whilst also collecting their opinions on what makes a chess program interesting and useful.

Current Documents:

Books regarding chess are extremely available however many require a good understanding of the game already, and do not teach new players how to play the game. Furthermore, whilst there are books that teach people how to play chess, it does not allow users to practice playing against another player, and I believe that a digital GUI and engine will prove more beneficial to beginners.

1.4.0 Identification of the prospective users:

The primary prospective user of my solution will be those wanting to learn the game and therefore by extension, those who want to teach others how to play the game. This will be primarily those in education, and so my application will be designed to support its use in small chess clubs, therefore, it will also be suitable to be used by private chess clubs also.

Since my primary set of prospective users have no assumed knowledge of chess, aspects of the solution will also assume no prior knowledge of chess (importantly the engine). This is important as it will allow me to develop the solution within the given time constraints.

Because of the large range of ages of those in education, my solution will need to cater to both younger and older users, and it therefore must suit an appropriate range of sophistication.

Secondarily, my program is directed at those who already know how to play chess. As described above, some aspects of my solution will not be suitable for this set of users, despite this however, I aim for the rest of the solution to be able to be used by players of any skill level.

1.5.0 Users’ needs:

1. A chess user interface that will allow the user to play games of chess locally.
2. An extension that will allow the users to play across a local-area network.
3. An intuitive interface that will aid the user in learning the rules of chess if they are a beginner.
4. Ability to export games played on the application to the universal PGN file format.
5. Ability to import games from the PGN file format, in order to analyse and watch games through the application.
6. Ability to import and export FEN strings of positions as an alternative to PGN files.
7. An engine for beginners to learn to play without having to have another player present.
8. Customization of the interface to make the game engaging and interesting.
9. The user should be able to set up an account.
10. The user should be able to generate an ELO rating based on their record in the application.

1.6.0 Constraints and Acceptable limitations:

This is a list of the limitations associated with my solution, accompanied by suitable justifications.

Hardware issues:

1. I do not have access to a server that would be able to host games for users. Therefore, the online feature will only allow for peer-to-peer connections on a LAN.

Other than this, most computers will be able to run my program with sufficient responsiveness. In particular, at my school the chess club will use HP all-in-one computers which will handle the program without any problem.

Software Issues:

1. My engine will not interface to the UCI (Stefan-Meyer Kahlen (Shredder Chess), 2004), meaning that the engine can only be used inside of my application, and is not compatible with other applications such as Chess Arena etc.
2. By extension, my application can only utilize my bespoke engine (and not others like Stockfish, or AlphaZero), since my application cannot interface the universal chess engine.
3. The solution will be written on a Windows operating system, specifically for a windows operating system. There is no way of interpreting a Visual Basic.NET source file into an executable file that could be used on other operating systems such as MacOS or Linux. Therefore, my program will only run on computers running the Windows operating system.

User Skills:

1. The program assumes that the user has a very basic knowledge of the game of chess; the user must understand the goal of the game.

Budget Issues:

1. I was unable to purchase an existing table-base for my engine. This has led to me having to create my own table-base, which will likely be less accurate than otherwise possible.

Limit to scope:

1. Due to time constraints, the sophistication of the engine will be limited, and only a basic decision rule algorithm will be implemented.

1.7.0 Numbered general & specific objectives:

General Objectives:

You must be able to play games of chess on my application, over local-area networks and locally to the machine.

You must be able to play against a bespoke engine, at a strength suitable for beginners.

You must be able to import - and export games to the application for analysis.

Specific Objectives:

The program should…

|  |  |
| --- | --- |
|  | be able to generate pseudo legal moves for a pawn. |
|  | … a knight. |
|  | … a bishop. |
|  | … a rook. |
|  | … a queen. |
|  | … a king. |
|  | be able to store and use a move object with the board & pieces. |
|  | be able to derive a subset of legal moves from a set of pseudo legal moves. |
|  | be able to setup the standard position of a chess board. |
|  | be able to make any legal move on a board. |
|  | be able to determine the state of any given board (i.e. whether someone has won, if it is a draw, etc). |
|  | be able to generate a map of which tiles are attacked by which alliance for any given board. |
|  | be able to store and then analyse any board to determine whether the previous move was legal (a ghost board). |
|  | be able to setup a multiplayer game over any local area network, given that the user has all necessary permissions to open the connection. |
|  | be able to add time constraints to games. |
|  | be able to represent a board graphically. |
|  | be able to represent a piece graphically. |
|  | allow the user to input moves using a graphical user interface. |
|  | be able to graphically display a move list containing the moves played in the current game. |
|  | be able to graphically display any previous positions whilst in a game via interaction with the move list. |
|  | be able to graphically display a clock if one is being used in the game. |
|  | be able to graphically show all legal moves for a selected piece in a given position. |
|  | allow the user to start a new game at any time. |
|  | allow the user to flip the board so that black can also be shown on the bottom of the board. |
|  | allow the user to undo a move (if they have enabled cheats in their preferences). |
|  | allow the user to show an evaluation of the position in a message box (if they have enabled cheats). |
|  | allow the user to export an already played game in PGN. |
|  | allow the user to export a position in FEN. |
|  | allow the user to import and analyse an already played game from PGN. |
|  | allow the user to import a position from FEN. |
|  | allow the user to select and store settings for the application, storing preferences for things such as: tile colour, enabling cheats, etc. |
|  | be able to traverse the game tree to any given depth. |
|  | be able to determine the best move in any given position. |
|  | be able to determine which player has the advantage in any given position. |
|  | be able to play audio files (of .wav format) from within the program. |
|  | be able to setup an account. |
|  | be able to store a record and ELO rating associated with an account. |
|  | be able to store an account in its own file (“.acc” extension). |
|  | be able to store a table of accounts (“.dsv” extension). |
|  | be able to allow the user to login to their account. |
|  | be able to secure the account using a username/password system. |
|  | be able to hash a password to ensure that the users’ data is secure. |
|  | be able to obfuscate an account file in order to preserve the integrity of the data. |
|  | be able to obfuscate an account file in order to hide the password hash associated with their account. |
|  | be able to show when a user is logged on, and appropriately update attributes as the user plays. |

1.8.0 Analysis data dictionary:

For my solution to function as intended, I will need to store the following data-items externally:

1. Accounts
2. Positions
3. Position Evaluations
4. A transposition table consisting of both hashes of a position as well as its corresponding evaluation and best move.
5. A configuration file detailing the user’s preferences.

1.9.0 Proposed object diagrams:

Interface

Account

Move List GUI

Clock GUI

Board GUI

Ghost Board

Game GUI

Clock

Board

Tile

Attack Map

Game

Piece

King

Queen

Bishop

Knight

Pawn

1.10.0 Justification of chosen solution:

Concerning the chess interface and engine side of my solution, special purpose software such as FIDE online arena and Chess.Com exists as competition to my solution. Despite this, I have chosen to create a bespoke solution in Visual Basic for the following reasons.

Firstly, none of the mentioned existing solutions were designed to be used as teaching aids, and as such, lack a variety of the features that I have deemed essential to the solution.

Secondly, the previously mentioned existing special purpose software choices both require a paid subscription in order to access all features of the software. By creating a bespoke solution, I will not have to implement a subscription payment model. I believe that leaving the program free to use is essential for the program to reach my target audience since firstly, students will likely not have an income to be able to pay for a subscription, and secondly most schools do not have a budget allocated to enable them to purchase the program. Finally, since my program is intended to try to interest people in Chess, logically, someone who has not played the game before and is not necessarily interested in the game to begin with will not purchase the program which defeats the purpose of my program.

Thirdly, existing solutions generally lack full customizability. Chess.com does not allow full customizability of the depictions of pieces or tile colours. Furthermore, FIDE online arena again requires you to pay the subscription in order to access their customization options. My bespoke solution will allow the user to customize as many aspects of the interface as possible. Because my target audience is mainly lower schoolchildren and that my solution is a teaching aid, I believe that it is important for my solution to be as engaging as possible.

Due to these problems associated with using existing special purpose software, I have decided that writing a bespoke solution will be most appropriate despite related limitations (as previously listed in “1.6.0 Constraints & Acceptable Limitations”).

I will be writing my bespoke solution in Visual Basic (a part of the .NET framework) for the following reasons:

1. Visual basic is free to use, the binary files generated will be able to be ran on any Windows computer that runs Windows 7 or later editions of Windows.
2. Visual Basic is a part of the .NET framework, and so it is possible to easily integrate other solutions (such as those written in C#) with the project in the future, should I choose to expand it.
3. I have experience with writing programs in Visual Basic, allowing me to write code that is more efficient.
4. Visual basic (along with other .NET framework languages) make use of the garbage collector, this will make memory management in my solution significantly easier than if I were to write the program in C++, etc.

1. Plies and moves are commonly interchangeable, however in the context of this project, these are the definitions that I will be using. [↑](#footnote-ref-1)
2. For further information on algebraic chess notation, refer to: https://en.wikibooks.org/wiki/Chess/Algebraic\_notations [↑](#footnote-ref-2)