TRBHUWAN UNIVERSITY

INSTITUTE OF ENGINEERING

THAPATHALI CAMPUS

PROJECT REPORT ON

COMPUTER CHESS

SUBMITTED BY:

MISHAN THAPA KSHETRI (THA076BCT019)

PRABIN BOHARA (THA076BCT026)

AYUSH BATALA (THA076BCT011)

SUBMITTED TO

DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING

THAPATHALI CAMPUS

KATHMANDU, NEPAL

October, 2021

CERTIFICATE OF APPROVAL

The undersigned certify that they have read and recommended to the **Department of Electronics and Computer Engineering, IOE, Thapathali Campus**, project work entitled "**Computer Chess**" submitted by **Ayush Batala, Mishan Thapa Kshetri, Prabin Bohara** in the fulfillment of project of OOP. The Project was carried out under special supervision and within the time frame prescribed by the syllabus.

We found the students to be hardworking, skilled and ready to undertake any related work to their field of study.

_____

Project Supervisor

Er. Saroj Shakya

Department of Electronics and Computer Engineering, Thapathali Campus

October, 2021

COPYRIGHT

ACKNOWLEDGEMENT

This is to acknowledge all those without whom this project would not have been reality. Firstly, we would like to thank our department of electronics and Computer engineering for providing us a chance to implement our knowledge in a project. Also, we would like to thank our lecturer Er. Saroj Shakya for his support and help for this project.

Special appreciation is extended towards Mr. Madhav Humagain for his guidance and encouragement. We would also like to take attention to the fact that we have learned a lot from internet

Eventually, we want to show our gratitude to all who helped us for the project.

MISHAN THAPA KSHETRI (THA076BCT019)

PRABIN BOHARA (THA076BCT026)

AYUSH BATALA (THA076BCT011)

Abstract

In this project, we will be designing and implementing a computer chess game. This chess game will have the feature of standard chess game played around the world. Each piece will follow the basic rules of chess. Here we will be designing a realistic virtual chess board and chess pieces by using a programming language C++. Along the use of basic algorithms in C++ and their libraries, we will be using SFML API for simple but powerful interactions with the users. Special moves of chess will also be implemented like castling, promotion. This game will be a platform for to players to compete. And this 2 players game is not limited in a single device. It will have the feature to play in single computer and to play through different computers. The network system is considered as LAN.

Keywords: API, SFML, Algorithm, Chess

Table of Contents

List of Figures

List of Tables

List of Abbreviations

et al.	And Others
AI	Artificial Intelligence
SFML	Simple and Fast Multimedia Library
GUI	Graphical User Interface
ACM	Association for Computing Machinery
API	Application Programming Interface
IT	Information Technology
i.e.	That is
LAN	Local Area Network
WAN	Wide Area Network
iOS	iPhone Operating System
PC	Personal Computer

1 INTRODUCTION

1.1 Background

Chess is a competitive board game played between two players. It is played on a square chessboard with 64 squares arranged in an eight-by-eight grid. At the start, each player (one controlling the white pieces, the other controlling the black pieces) controls sixteen pieces: one king, one queen, two rooks, two knights, two bishops, and eight pawns. The objective of the game is to checkmate the opponent's king, whereby the king is under immediate attack (in "check") and there is no way for it to escape. There are also several ways a game can end in a draw. Chess is an abstract strategy game and involves no hidden information. Played by millions all around the world, chess is originally believed to have originated in India around 7^{th} century and in the early 19^{th} century, it reached the current standard rules and appearance of pieces.

With the current development in computers and internet, humans are able to make the develop the chess into online game which can be played on display screens over the internet. Though most of the World Championship are played in physical chess board, chess application is also being used more day by day due to the advanced in technology and internet. So, we proposed an algorithm for computer program that plays chess according to result of our analysis. It follows most of the rules of physical board chess game. It can be played between two players using both same computer and different computers. It also keeps the timer for both players. It emulates real world scenario via GUI which we created using SFML and C++. We also tried to contribute with some new approaches to chess-playing programming.

1.2 Motivation

One of the goals of early computer scientists was to create a chess-playing machine. In 1997, Deep Blue became the first computer to beat the reigning World Champion in a match when it defeated Garry Kasparov. Today's chess engines are significantly stronger than even the best human players and have deeply influenced the development of chess theory. Online chess saw a spike in growth during the quarantines of the COVID-19 pandemic. And it involves vast engineering knowledge and algorithm for implementing basic rules of chess into the game. So, by doing research into different topic, we decided to choose a “Chess Application for Desktop” as our final project. It also helps us to gain more knowledge about GUI and networking.

Another important motivation was urge for the implementation of courses we have been undertaking during our study. This will enable us to challenge the knowledge we have gained at class and an opportunity strengthen them and learn something new on the way.

1.3 Problem Definition

Chess is the one of the most popular indoor, recreational and competitive game. It is a multiplayer strategical game. For developing the game into the simulated environment, it uses a huge amount of algorithm and knowledge of SFML and C++. We are willing to add networking, design and various others feature into our game. This makes the project challenging yet approachable. Hence, we decided to take this game as our project topic.

1.4 Project Objectives

The specific objectives of the project are listed below:

- To develop intuitive GUI to represent real world object into simulated environment using SFML and C++.
- To allow two different players to compete from two different devices which we will accomplished using networking.

1.5 Scope and Application

This project allows us to play chess using simulated environment. Everyone can use this application in the desktop. We can also play game via network. Later in the future we can extend this project and add various features to make it better application.

There are several applications of this project. Some of them are given below:

- Using this project, we can compete with our friend from two different computer
- It can be used to hold a chess tournament in school, home etc.
- It provides the virtual board to play chess during quarantine.
- It is the source of entertainment and knowledge.

2 LITERATURE REVIEW

With the advent of computers, interest in the development of chess playing programs was stimulated. As of today's, world, many computers chess games have been created with different algorithm concepts and graphics.

2.1 Shannon Paper

In March 1950 Claude Shannon of Bell Labs, published "programming a computer for playing chess". This was the first technical paper on computer chess; however, the paper was entirely theoretical. The paper by Shannon (1950) remains even to this day to be central importance. The minimax algorithm was first applied in a computer chess context in the landmark paper of Shannon. He also introduced the classification of chess playing programs into either type A or B. Type A (brute force) or B (selective search). But its effective implementation was still to be done. [1]

2.2 Chess by Bernstein

The first significant chess playing program was by Bernstein (1957) and ran on an IBM 704 computer. This was not a 'brute force' program as it only selected the best seven moves for consideration using heuristics based on chess lore. Compared to the sophisticated brute force programs of today which generate the full span of moves at the root, this is a very limited range of moves. [2]

2.3 Chess 4.5

The first program to achieve full width search and make 'brute force' appear a viable possibility was Chess 4.5. This program was developed for entry into the ACM 1973 Computer Chess contest by Slate and Atkin, using the experience they had gained in programming earlier selective search chess programs. The techniques used by Slate and Atkin (1977) are still in use today although they have been refined and improved over the years for making those techniques more effective. [3]

2.4 Chess.com

This platform for chess on today's world is considered as one of the best. Live online chess can be played against other users, match up against computer can be played. It is available on both android and iOS. It has hosted online tournaments including Titled Tuesdays, the PRO Chess League, the Speed Chess Championships and computer vs computer events. [4]

3 PROPOSED SYSTEM ARCHITECTURE

3.1 Flowchart of Proposed Model

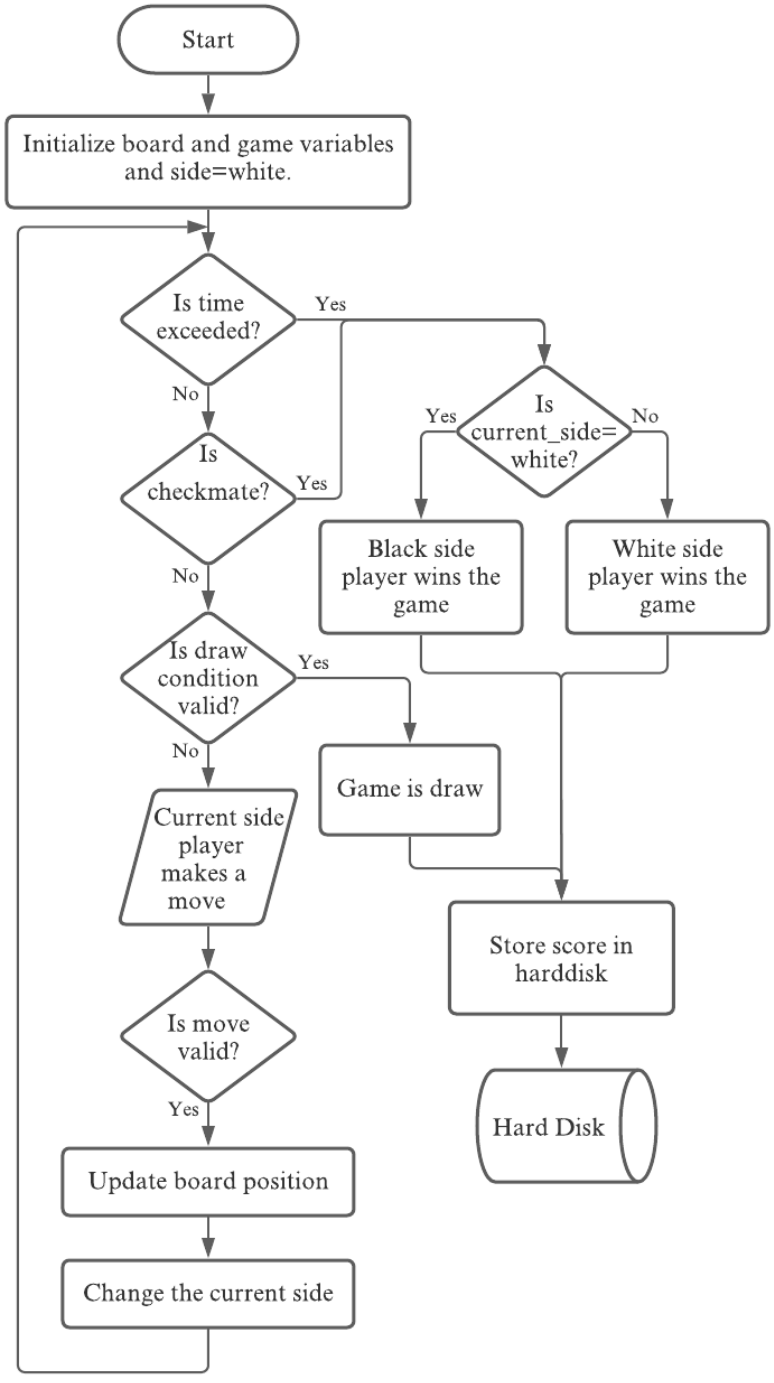


Figure 3.1-1: Flowchart of Proposed Model

3.2 Elaboration of Working Principle:

The sole purpose of our project is to make a chess game that is playable between two people either on a single PC or multiple PC.

So, the game starts with the initial position of chess pieces on the board. After setting the flag for detecting the turn of sides (white or black), a player is only allowed to click or select his/her side's pieces which eventually shows all possible moves of the piece. There is always a system to block invalid moves of pieces which is defined by the "move validator" function. Different cases like check, En passant checkmate, castling is also checked for every move if possible. The constraint of time, which is pre-determined by the player, can be used to provide a justified time frame for both the parties. Once the move of a player is successful, the flag of which side to play is switched to the other one.

However, in the play between players sitting on a different machine could be functionalized with the concept of socket programming. Here, each PC will run a client executable controlled by one server. After generating a socket for each PC, it immediately sends a connection request to the server asking if any player is available. If the server response is positive, then the game is good to go. If both the parties agree for the start, then the chess board is initialized by the server sending the initial board array to both clients. Server then ignores the data from the black side player, and hence gives a chance for white player to move his/her piece. In a similar manner the server controls the movement of each side and eventually disconnects from each socket if the client chooses to quit the game.

Finally, the result is decided following the end game rules and the score is stored in the database of the server corresponding to the player.

3.3 Tools and Environment

- Visual Studio 2019
- SFML Library
- VS Code

4 METHODOLOGY

Before approaching to write the actual codes on chess programming, fundamentals of classical chess are mandatory. How the pieces on the board are placed, which pieces are allowed to attack how many/which squares, looking for checks and many other theories is discussed below:

4.1 Board Representation:

The arrangement of every chess piece on respective squares of the chessboard to easily manipulate functions like movement is simply termed as board representation.

It is also regarded as the data structure of chess programming. Its significance is considered when the pieces of the board need to be accessed. Generally, we come across two kinds of chess board representation i.e. off-set(array) and bitmaps.

4.1.1 Off-set Representation:

It is basically an array of elements in which each square of the chess board is represented by each element of the array. Array can be used in both one dimensional and two-dimensional format.

In one dimensional representation, each square of the board has values in incremental order despite the change in rank or file (row or column). Thus, switching row or column must be logically defined subjectively.

In two-dimensional representation, each square of the board corresponds to the pair of indexes of a 2D array. For example: board [1][5] represents b6 in the chess board below. Hence, this system of representation does not require separate logical implementation to identify the squares.

4.1.2 Bitmap Representation:

This way of representation of the board is mainly focused on the number of each piece on the board defined by different 64-bit variables. For example: ‘unsigned long int’ data type offers 64-bit space in all kinds of data models whereas ‘unsigned long int’ would work for LP64 data model. This data type is contained by every unique piece in the board (all 12 types). Each bit of data is equivalent to each square position of the board which is defined as ‘1’ if that piece is present there otherwise it is defined as ‘0’.

4.2 Piece Representation

Representing the piece in proper format makes sense when we need to clearly state which piece it is and what position it can move in. One of the ways to do this is by giving values to every unique piece like pawn=1, knight=2, bishop=3, rook=4, queen=5 and king=6. And using positive for white and negative for black will make this concept more logical. Furthermore, giving a score to each piece as pawn=1, knight=3, bishop=3, rook=5 and queen=9 can help in the calculation of material available for each side (white and black). This can even be useful for the 'live win prediction' algorithm.

4.3 Movement of Pieces

- **Pawn**

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.

- **Rook**

Rook can move any number of squares, up and down and side to side but cannot leap over other pieces.

- **Knight**

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.

- **Bishop**

Bishops can move any number of squares diagonally but cannot leap over other pieces.

- **Queen**

Queens can move any number of squares along ranks, files and diagonals but cannot leap over other pieces. Its movement is the combination of the movement of both bishop and rook.

- **King**

Kings can move one square at a time in any direction. The king is the most valuable piece — attacks on the king must be immediately countered.

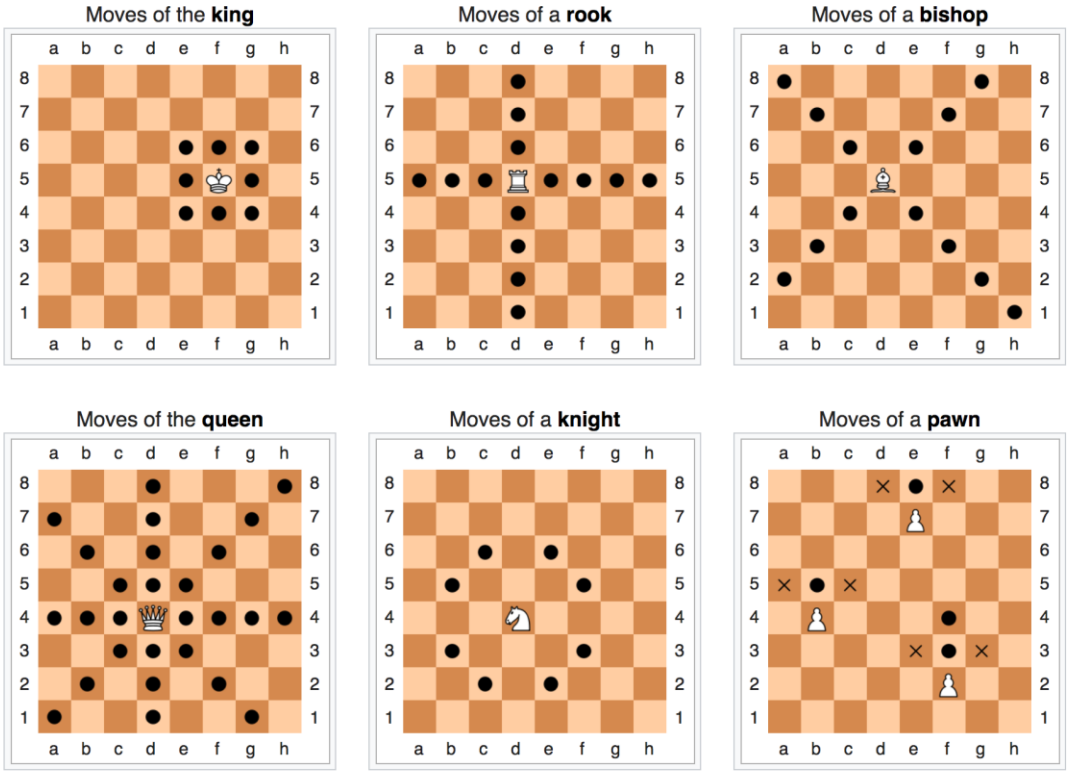


Figure 4.3-1: Movement of Pieces

4.4 Special Movement

4.4.1 Check and Checkmate

When king is under attack, it is said to be in check. A move in response to a check is legal only if it results in a position where the king is no longer in check. Castling is not a permissible response to a check.

When the king is in check, and there is no legal way to get it out of check, it is said to be in checkmate and the player getting checkmate loses the game. It is never legal for a player to make a move that puts or leaves the player's own king in check.

4.4.3 En passant

When a pawn makes a two-step advance from its starting position and there is an opponent's pawn on a square next to the destination square on an adjacent file, then the opponent's pawn can capture it En passant ("in passing"), moving to the square the pawn passed over. This can be done only on the turn immediately following the enemy pawn's two-square advance; otherwise, the right to do so is forfeited.

4.4.2 Castling

Castling involves rook and king. It can be done once in match by each player. Castling consists of moving the king two squares along the player's first rank toward a rook on the same rank, and then placing the rook on the last square that the king crossed.

Castling is permissible if the following conditions are met:

- Neither the king nor the rook has previously moved during the game.
- There are no pieces between the king and the rook.
- The king is not in check and will not pass through or land on any square attacked by an enemy piece.

Castling is still permitted if the rook is under attack, or if the rook crosses an attacked square.

4.4.4 Promotion

When a pawn advances to the eighth rank, as part of the move, it is promoted and must be exchanged for the player's choice of queen, rook, bishop, or knight of the same color. Usually, the pawn is chosen to be promoted to a queen, but in some cases, another piece is chosen; this is called underpromotion.

4.5 End of the Game

4.5.1 Win

A game can be won in the following ways:

- Checkmate: The king is in check and the player has no legal move.
- Resignation: A player may resign, conceding the game to the opponent. Most tournament players consider it good etiquette to resign in a hopeless position.
- Win on time: In games with a time control, a player wins if the opponent runs out of time, even if the opponent has a superior position, as long as the player has a theoretical possibility to checkmate the opponent were the game to continue.

4.5.2 Draw

There are several ways a game can end in a draw:

- Stalemate: If the player to move has no legal move, but is not in check, the position is a stalemate, and the game is drawn.
- Dead position: If neither player is able to checkmate the other by any legal sequence of moves, the game is drawn.
- Fifty-move rule: If during the previous 50 moves no pawn has been moved and no capture has been made, either player can claim a draw.
- Draw on time: In games with a time control, the game is drawn if a player is out of time and no sequence of legal moves would allow the opponent to checkmate the player.

4.5.3 Time Control

In competition, chess games are played with a time control. If a player's time runs out before the game is completed, the game is automatically lost (provided the opponent has enough pieces left to deliver checkmate). The duration of a game can varies according to the player's choice.

4.6 Playing Format

4.6.1 Person vs Person on Same Device

In this format of play, two players are allowed to enjoy the game of chess on a single pc. Each player is given a chance to play their move after the opponent’s move. Since the moves are triggered from the same device, the parameters like “who had just played the move, whose turn is next " is tracked with reference to a predefined flag. The significance of this flag is observed when the movement of the opponent side needs to be paralyzed and thus enabling the movement of the other side (white or black).

4.6.2 Person vs Person on Different Device

This format of play is characterized by the concept of networking. This system of play overcomes the drawback of confusion created for the turn of movement. Since both the players are from different nodes in a network, it can make clear judgement of which side is on play. The network system can be either LAN or WAN where the concept of WAN could be extended over the internet as well.

5 RESULT AND ANALYSIS

5.1 Result

Eventually, we have completed our project adding the basic rules of chess along with some decent features complimenting the game of chess. Our chess can be played normally between two players. It has all the functions like castling, promotion, etc. Possible moves of all the pieces is shown on clicking the particular piece. Illegal moves while king is on check is also forbidden. On the other hand, players can even play with different computers.

5.2 Homepage

This is the first page of our project which guides the user for different options. Player can either choose to play directly without changing anything or can choose the settings option to make some changes.

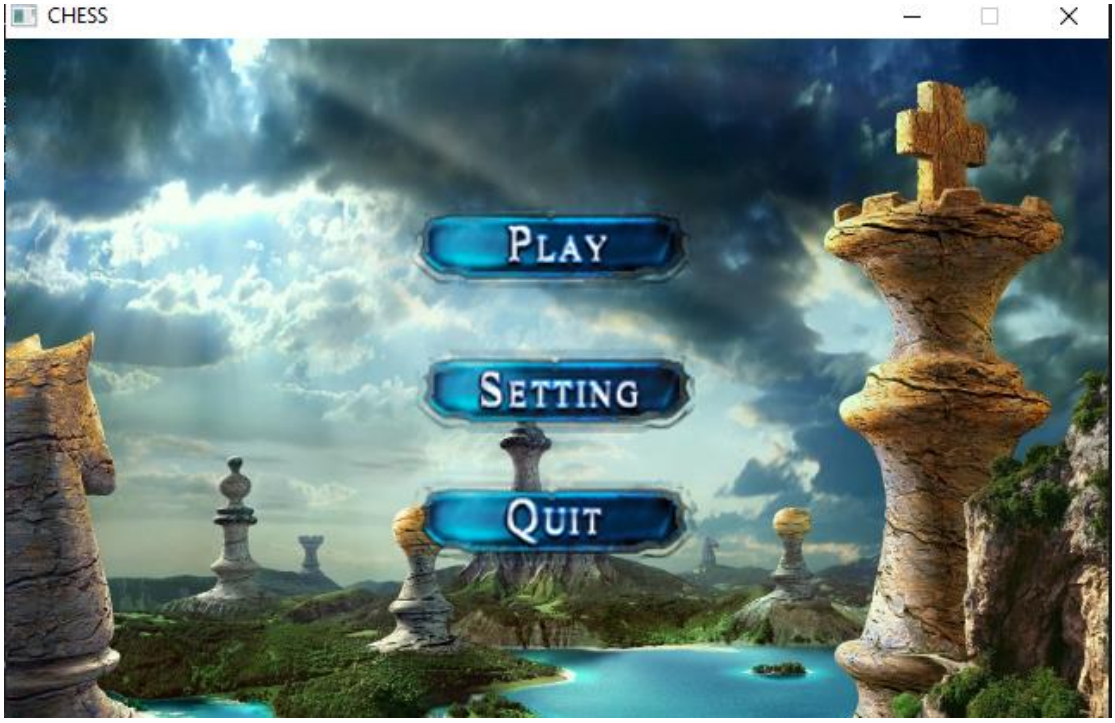


Figure 5.2-1: Homepage

5.3 Modes

There are two modes in this game. One is to play in a single computer and other is to play from two different computers. By clicking one of the modes, we can enter the main game section.

Figure 5.3-1: Modes

5.4 Main Game

Below is the UI of the game after choosing to play the game. Two players can the game one being the black and other being the white.

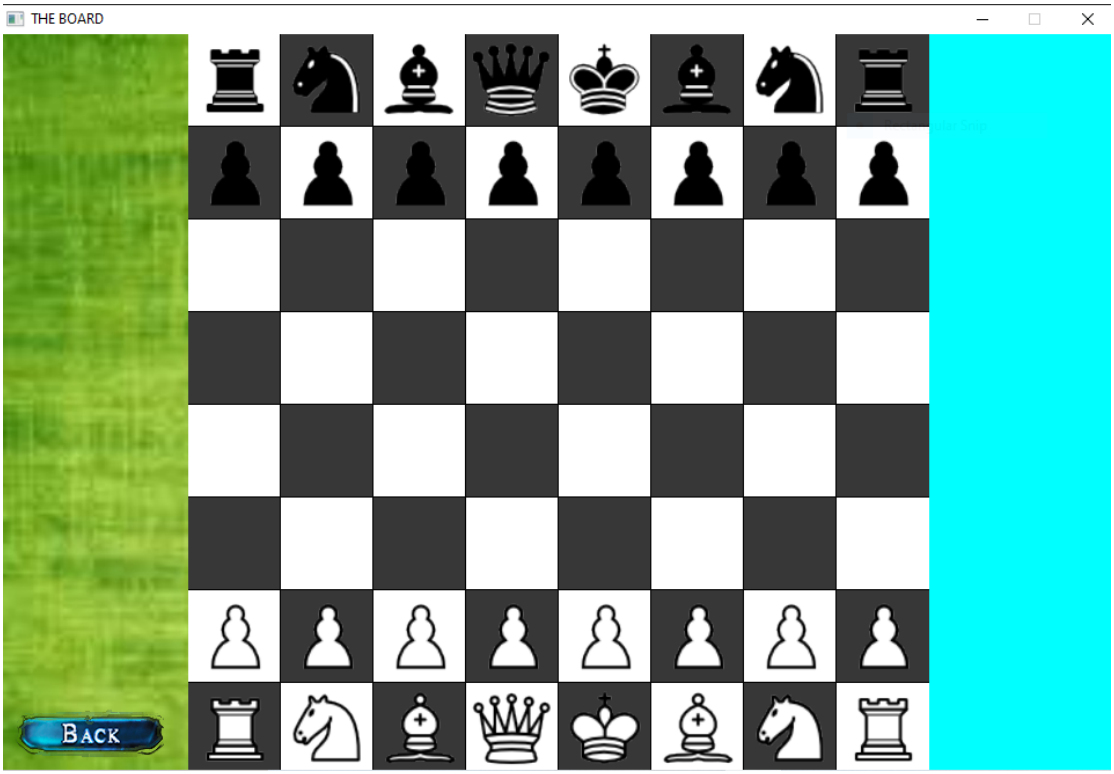


Figure 5.4-1: Main Game

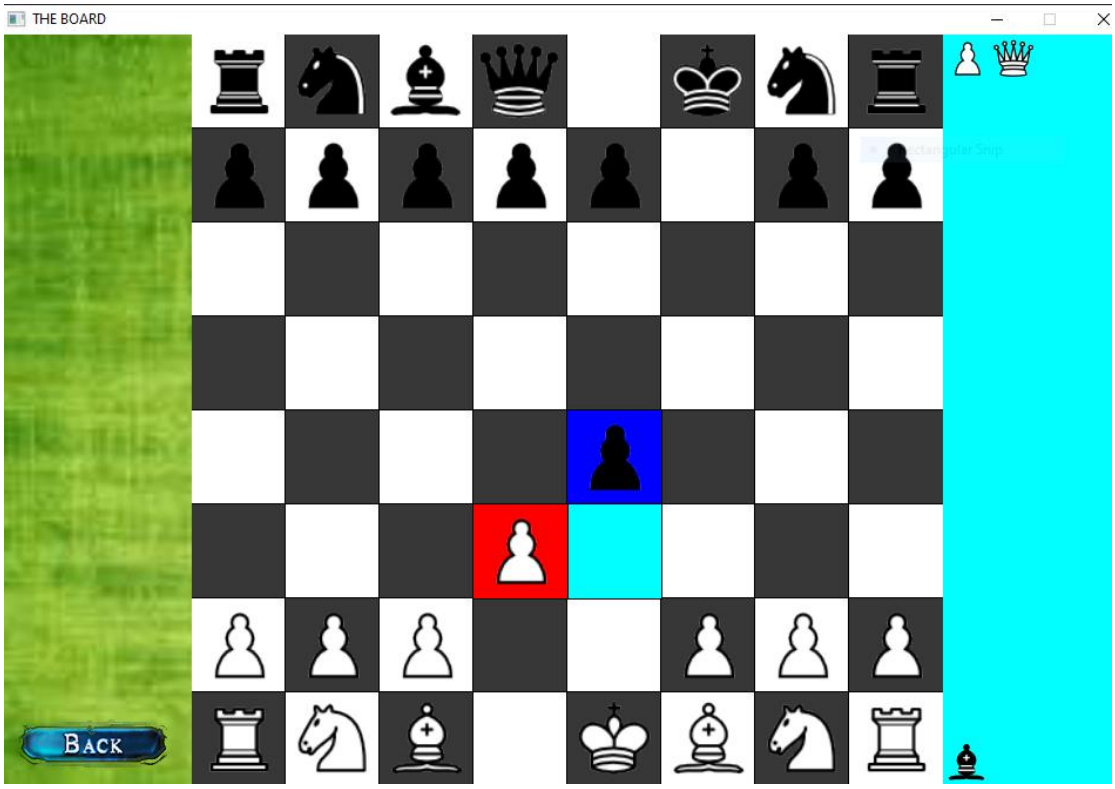


Figure 5.4-2: Main Game

Offline

It is played two players on the same computer. In this mode, two players are given chance to play their move turn by turn using same hardware resource.

Online

In this mode of play, two players play from different device enjoying from their individual computer. A server is setup and two computers are connected to the server and game is controlled by the main server and played from two individual nodes working as a client.

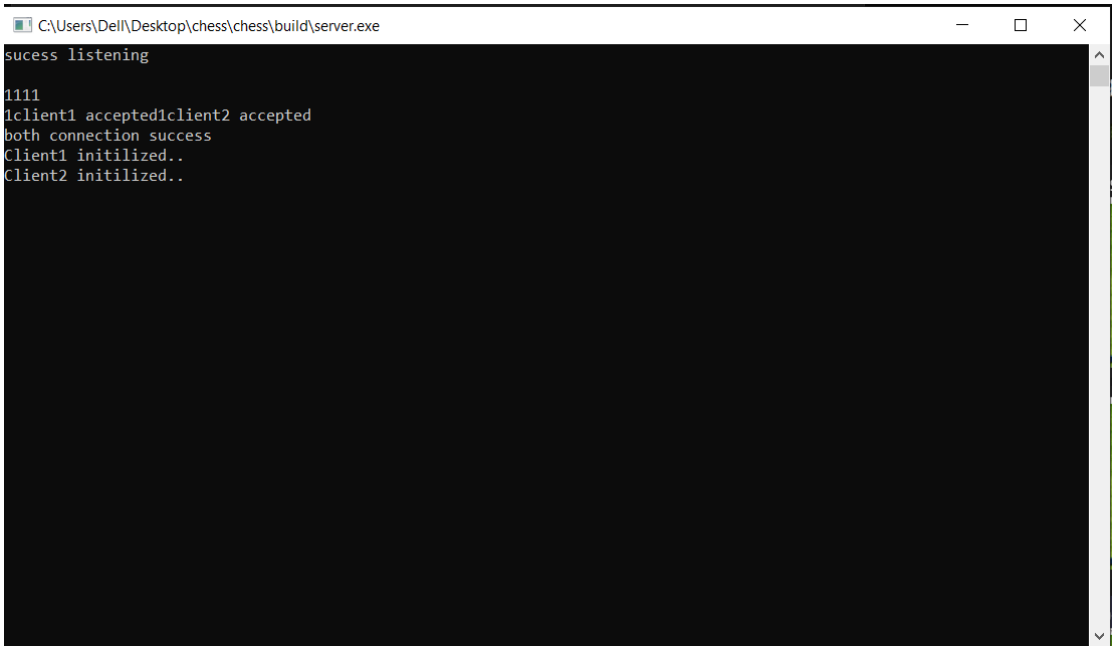


Figure 5.4-3: Server side

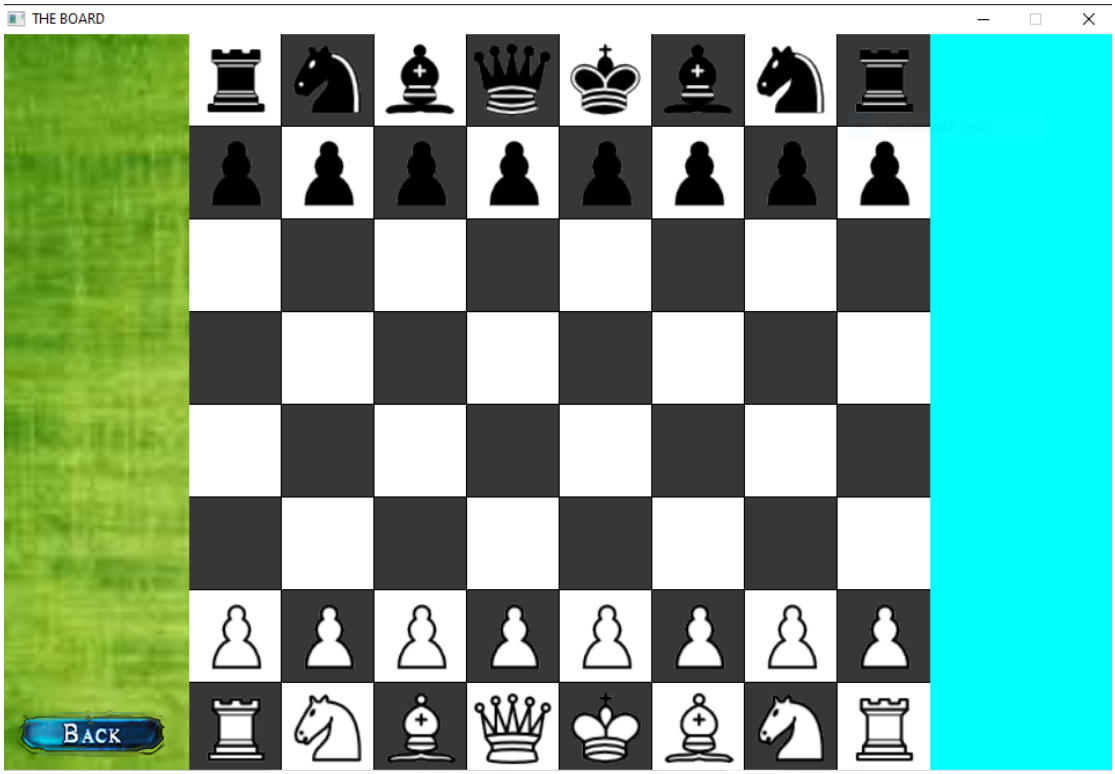


Figure 5.4-4: PC1(client1) view

Figure 5.4-5: PC2(client2) view

5.5 Settings

In this section, players can be able to change turn the music and sound on and off as per their interest.

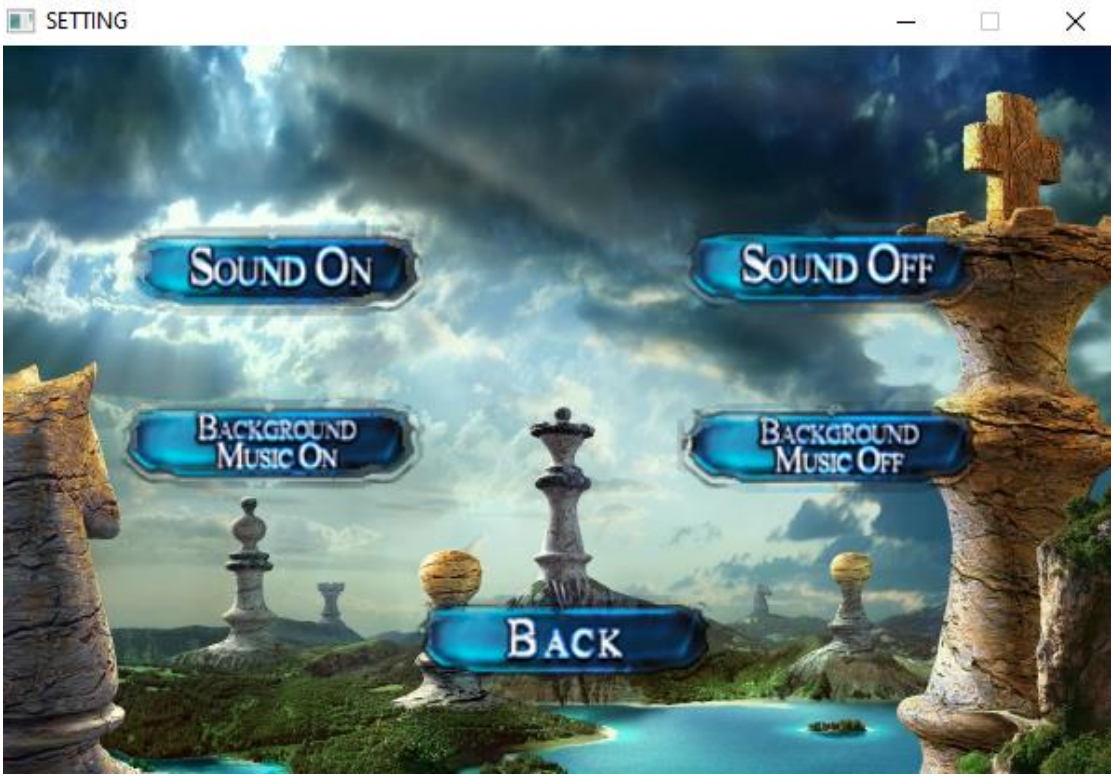


Figure 5.5-1: Setting

5.6 Analysis

Form some testing, our project seems to be in a good condition for two ordinary players to play. The properties of the real world chess, which is kept in this project works fine. Although on testing, it shows some inaccuracy in case of checkmate. Bringing down to number, for the functions in the project, it is accurate for every functions except for the checkmate.

It is seen that for someone new to this game can also play along in this game as every steps which can be moved by a piece is calculated is shown. So it can a basic program for beginners.

6 CONCLUSION AND FUTURE ENHANCEMENT

6.1 Conclusion

In summary, this is a chess program where two players can play chess using the computer either online or offline as per the requirement. The conventional chess board is visualized using computer programming and optimal GUI is used for the better user experience.

This project was started by implementing the rules of Chess where features of individual pieces was deployed accordingly. A set of objectives were set to develop an ideal computer chess game. Chess pieces shows all the possible moves and doesn't show any square if its movement is not possible. It shows the pieces which were eliminated on the side of the board as a captured pieces. It can also be played via two computers using networking.

However, there were some limitations we have to afford for this project due to some constraints and time limitation.

6.2 Limitation

- Login system for the users is not available.
- Playing against computer (bot) is not implemented.
- Time limiting is not added.
- We cannot end this game by draw.

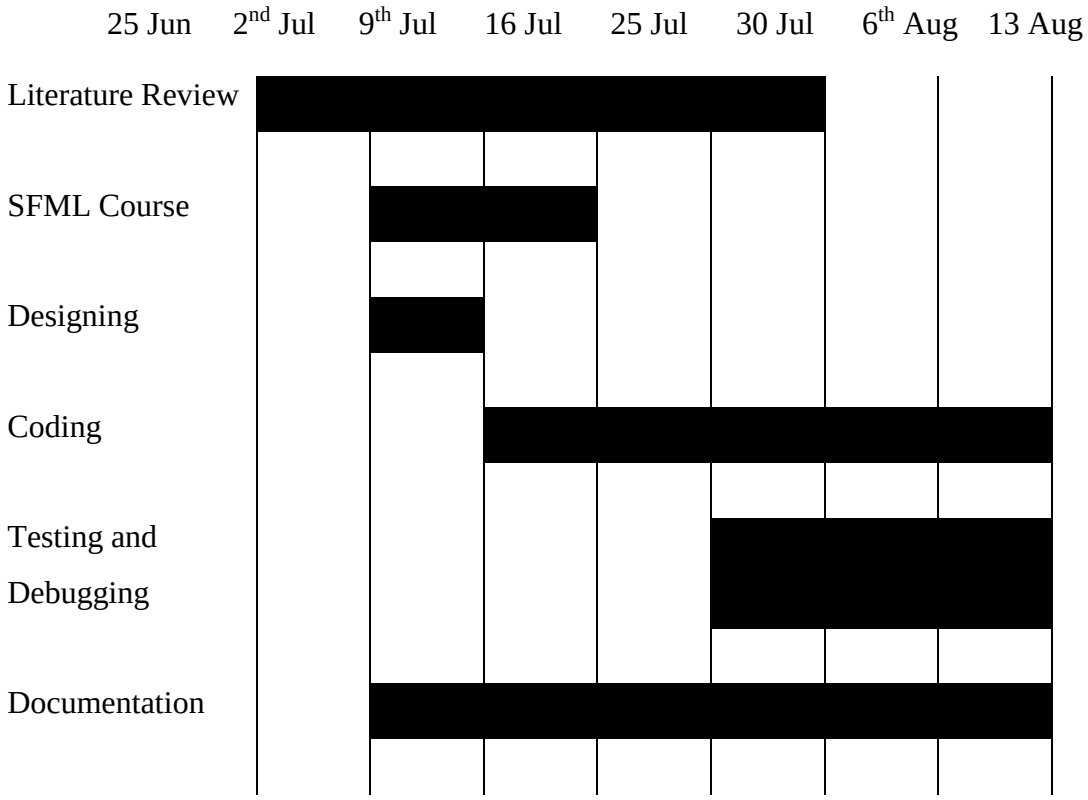
6.3 Future Enhancement

- Ending the game by draw between the players could be added.
- A timer to set the time for players could be added.
- AI could be added.
- Different sound effects like moves, promotion, check, checkmate, etc. can be added.

7 APPENDICES

Appendix A: Time Estimation

Table 7-1: Gantt chart



Appendix B: System Requirements

Table 7-2: System Requirements

Items	**Requirements**
OS	Windows 7/8/10/11
Processor	Above Intel i3
RAM	More than 500 MB

References

[1] D. H and D. A, "The Historical Development of Computer Chess," AAAI Technical Report WS-97-04, Park Square, 1997.

[2] P. S. P and A. N, "Literature Survey of Chess Engines," Mumbai,India, 2017.

[3] D. S and A. N, "Literature Review of Chess Studies," November, 2014.

[4] "Wikipedia," [Online]. Available: https://en.wikipedia.org/wiki/Chess.com. [Accessed 22 07 2021].

[5] D. R. M. (. Hyatt., "Crafty 25.2," 29 October 2016. [Online]. Available: https://craftychess.com.

[6] M. L, "Chess Programming Wiki," 26 October 2007. [Online]. Available: www.chessprogramming.org..

[7] "Documentation of SFML 2.5.1," [Online]. Available: https://www.sfml-dev.org/documentation/2.5.1/. [Accessed 5 July 2021].