Project Report

Title: Chess Club

College

School of International Education

Leader Name & Student ID

Dhar Sudipta Sotra

202169990264

Collaborator (Name and Student ID)

Subodh Pokhrel (202169990044)

Ibrahim Alfaqih (202169990079)

Walid KH J.Suleiman (202169990015)

Teacher

Zhang Jing

Date:23rd December, 2022

1.Abstract

In this paper, we implement the standard chess game in JAVA, a popular object-oriented programming language. Our program has been developed and thoroughly tested on Windows. It allows two players to compete against each other. JAVA's object-oriented characteristics, which include abstraction and inheritance which greatly simplify development. At the end of the paper, we also discuss some areas that could be improved.

2. The Background of System

The chess game is a casual game that requires two players to play on an 8-by-8 chessboard. The game's principles are simple, but there are limitless ways to move the pieces, so each step is vital to the player. The chess game is a highly adversarial game that requires players to consolidate step by step and develop their logical thinking ability. Computers, a machine that can perform numerical calculations, logical operations, memory storage, and program operations, are already playing an indispensable role in this digital age. It can replace human thinking and deal with problems that human beings cannot solve. Computer provides chess game a brand new carrier, which not only makes it more convenient for people to play recreational games, train their intelligence and improve their chess skills, but also enables people to see the omnipotent compatibility of the computer field and the development vitality of other industries brought by computers. Even more it is helpful to promote research in the field of artificial intelligence. The crucial role the computer plays in the chess game is storing a large number of game records and remnants, acting as assistants in the game, helping players check the game, studying opponents, preparing and analyzing the situation before the game. The methods of analyzing chess games and documenting rules presented by the computer may help solve many practical problems such as economic management, military command and so on in the future.

Chess is often regarded as the game that is most commonly associated with intelligence and strategy. Science has in fact proven that chess players have more cognitive skill than non-chess players. Chess increases problem-solving skills. Legendary former World Champion Garry Kasparov once wrote: “Chess helps you to concentrate, improve your logic. It teaches you to play by the rules and take responsibility for your actions, how to problem solve in an uncertain environment.” As a result, if a player is able to defeat another player, we feel that Artificial Intelligence (hence AI) could simplify the wining opportunities by showing the next moves possibilities. Our game is entirely relied on encoded human knowledge; it is likely that Artificial Intelligence (hence AI) will be also involved. That is why JAVA programming language and Artificial

We chose the international chess game project because it's a traditional chess game which is known globally which also indicates how people apply their brains to play this game. At the Same time, it is a traditional game for many countries and its very popular. our goal is to promote this game and make it a worldwide playable application. The targeted clients for this project is the new generation students and anyone who is interested in international Chess games. The aim of this project is to introduce international and let our clients have a great experience using the game we built. we hope that the game will meet user's expectations and hopefully it will be promoted and shared inside and outside the community.
Intelligence (hence AI) are two approaches we chose.

3. Main functions

a) Two dimensional (2D) technology was added to give you an enthusiastic experience and it makes the shape of the game more realistic

b) Our game will have the ability to deal with humans as it is based on artificial intelligence, which will make the game challenging enough.

c) This project develops a self-learning Chess program using a crossover of the ideas of co-evolutionary approaches

d) The game has online mode using "free internet chess service" server with practice and puzzle board

Figure 1: Game main Function & List of Some Fundamental Functions.

4.Detailed Design

a) User interface

With the help of android studio, we built the main user interface and the starting user interface using XML. (Fig.2).

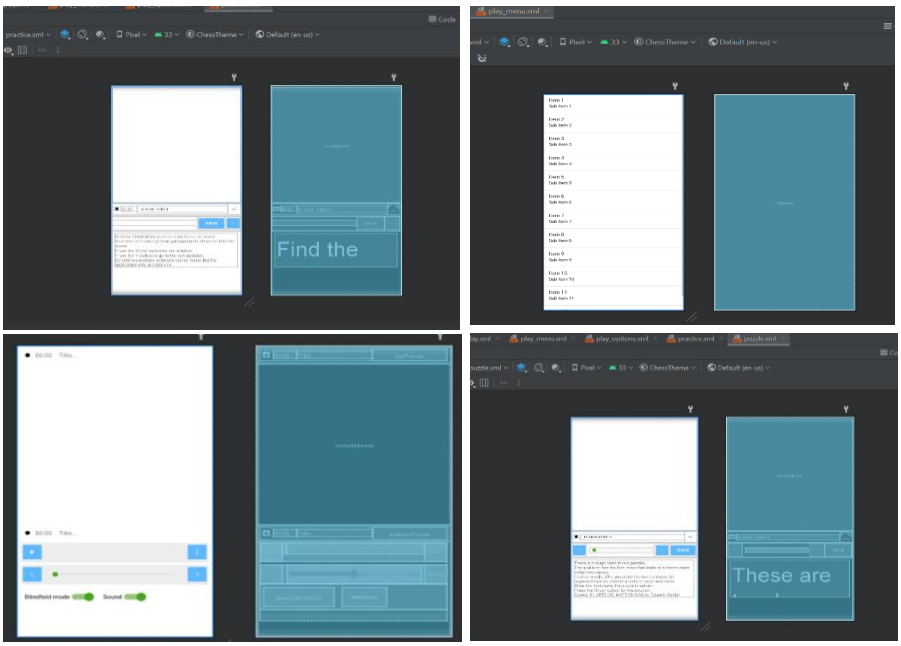


Figure 2 : Design User Interface Using Xml

b) Chess Board:

We modified the assets we already had for the Chess board and piece by changing the color texture settings by doing that it turned into a more realistic and authentic results. We designed The down part of the board with a brown color however the upper part had a normal brown and light brown tile. We can also change the color of the board from setting. In order to build the upper part, we had to duplicate the black and white tiles thirty-two times. All of the above details are shown in (Fig.3).

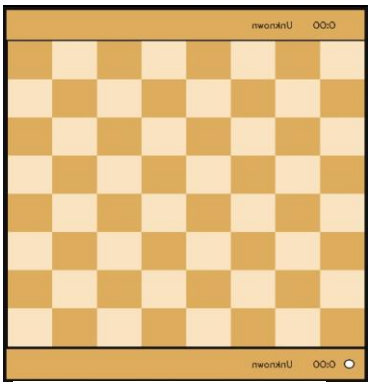


Figure 3: Chess Board

c) Chess Piece:

On the other hand the chess pieces was designed and customized after bringing the assets from the Internet we choose the basic color which is black and white to give a classic game view

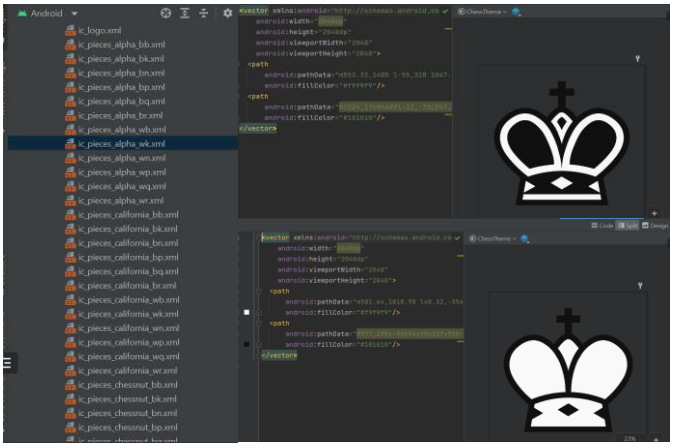


Figure 4: Chess Piece made by xml.

d) Game logo:

The game logo simulates the two of the most powerful pieces in the game of chess, and the yellow color was chosen in the background, in addition to that we used XML to build the icon in vector background.

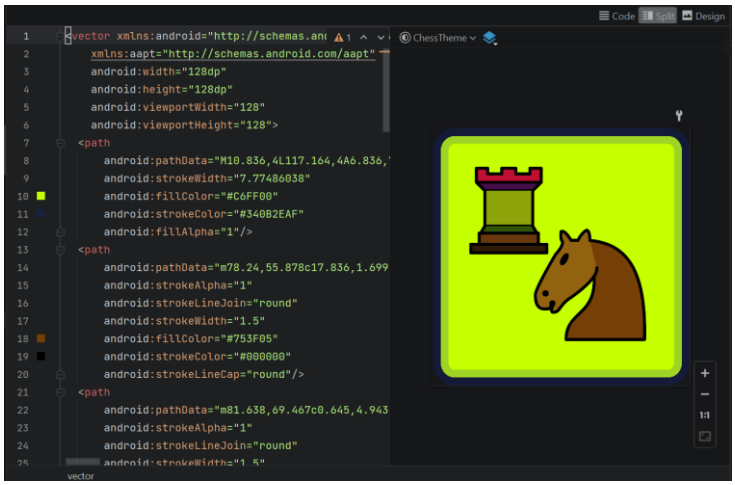


Figure 5:Chess game logo.

5. Process discussion:

The chessboard and the pieces are the most important components in a chess game. All of the features are impossible to execute without these two components. In the meantime, these two aspects are intertwined since the board must limit piece movement at all times, and the game cannot function without the pieces.

To summarize everything, the design and initialization of the chessboard and pieces should be prioritized. Before discussing any aspect of the game's design, we must first define the game's settings:

- ✓ The chessboard is two-dimensional and it is 8 inches by 8 inches.
- ✓ Players are divided into two camps, say white and black, with the white pieces moving first.
- ✓ The chessboard is arranged with the white pieces at the top and the black pieces at the bottom and vice versa.

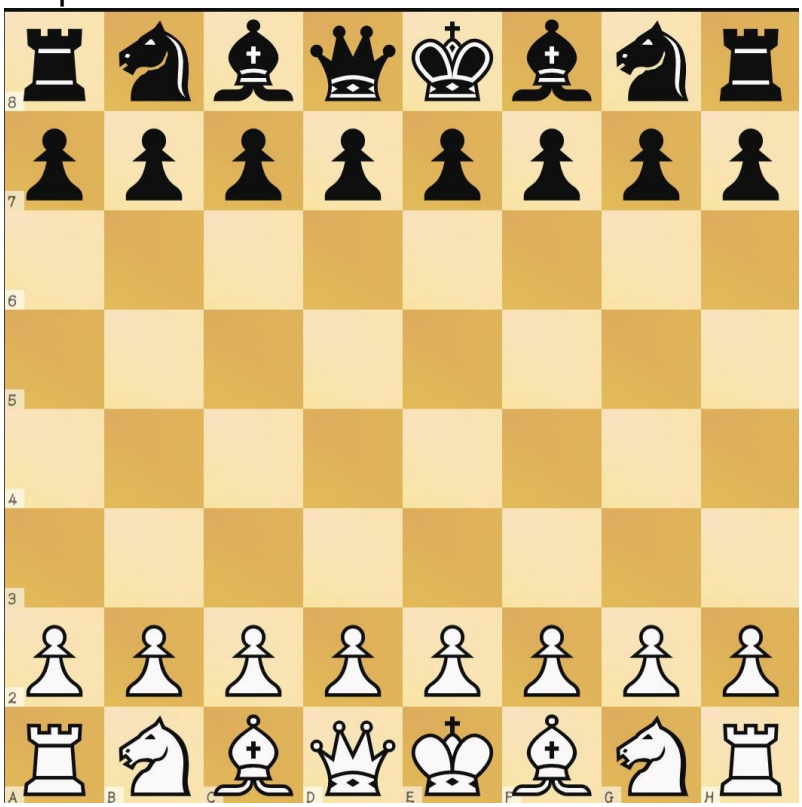


Figure 6 : Standard Chess Board Format.

- ✓ The king, queen, bishop, knight, rook, and pawn are the six types of pieces. One king, one queen, two bishops, two knights, two rooks, and eight pawns are given to each player at the start of the game. The white pawns are in the second row at the start of the game, whereas the black pawns are in the seventh row. Placing the rooks at the board's corners, the knights near the rooks, the bishops near the knights, and the queen near the bishop. The king's position is the rest square. (Fig.6) is an example of a standard chessboard. Following that, we must declare the movement rules for various chess pieces.

- ✓ A king can only move one square in any direction, even diagonally. It is not possible for a king to pass through other pieces. Furthermore, the king cannot advance into a square controlled by the enemy's piece; otherwise, the foul is committed. Other pieces will "check" the king, hence the particular judgement for king should be coded.

- ✓ A queen can travel in any direction, even diagonally, for any number of spaces. Other pieces are not allowed to pass through a queen.
- ✓ A bishop can only travel diagonally, but in any direction. Other pieces are not allowed to pass past a bishop.
- ✓ A knight can move in a two-by-one or one-by-two L-shape. Only the knight is unaffected by other pieces. The knight is the only piece that is unaffected by the presence of other pieces (i.e., it can move through other pieces to get to an open square)
- ✓ A rook can move any number of squares on the board, but only in a straight line that is not diagonal. A rook is not allowed to pass through other pieces.
- ✓ A pawn can only move forward, but only in certain directions, towards the opponent's side of the board. A pawn can advance one or two squares on its first move in the game; on subsequent moves, a piece can only advance one square. A pawn is not allowed to pass through other pieces. Furthermore, the pawn is not permitted to utilize this forward motion to capture the opponent's piece. Instead, the pawn advances one square diagonally forward in order to capture a piece and remove it off the board.

6. Board and Chess game classes:

Set up a chessboard, make chess pieces, start games, move pieces, analyze games, and end games—all of these processes must be abstracted into class object behavior. Because these functions must be implemented on the chess board. We develop a Board class to contain all of them. The name of the Board class (Board Package) implies that it can be used in any game that takes place on a board, with chess being one of them.

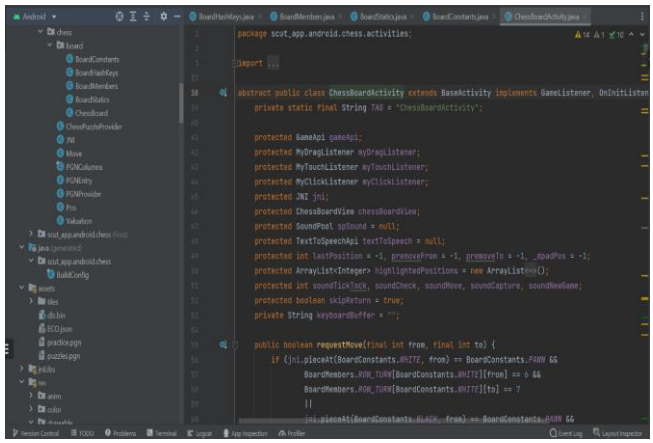


Figure 7: Chess Board Configuration.

We may easily inherit from the created Board class if we want to write other board games in the future (e.g., Chinese chess, checkers). This encapsulates the qualities of object-oriented languages including abstraction, inheritance.

The Board class is designed as follows, with only a few of its most significant methods and characteristics shown in (Fig.7 & 8)

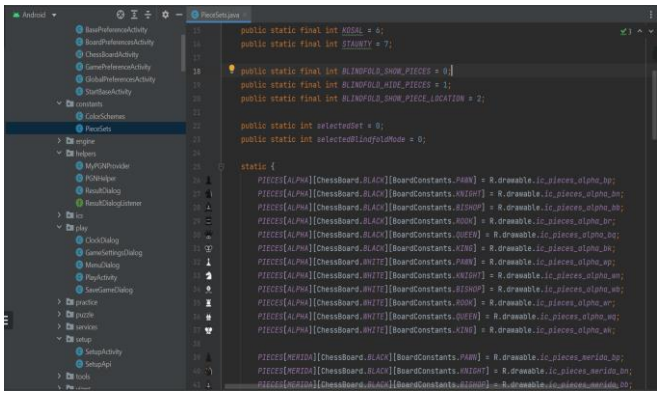
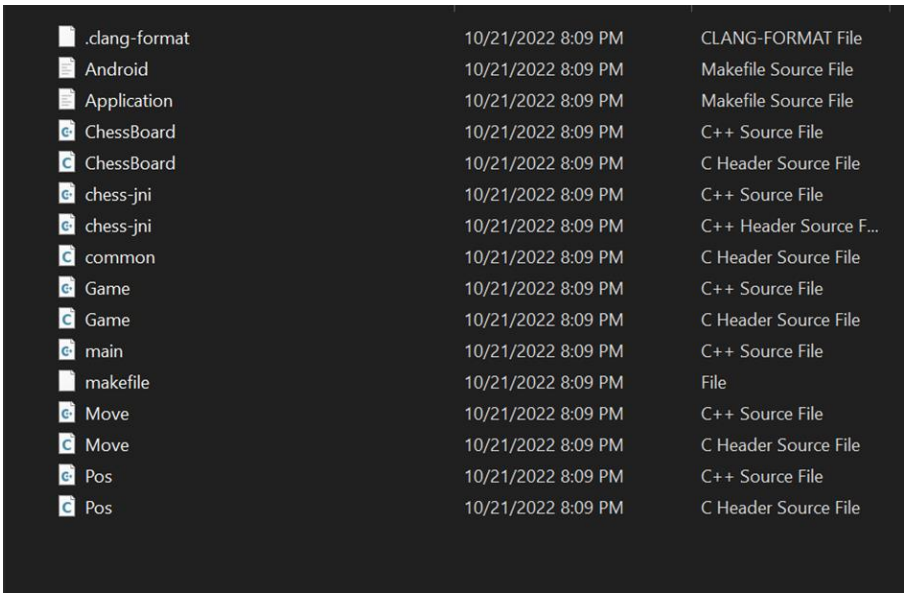


Figure 8:Chess Pieces

Unreal Engine :

It should be mentioned that we have used the unreal engine 4 to combine our JAVA code with graphical interface (XML) ,2D models, initialization, and create environment for android user interface. To optimize our code and we use unreal engine plugin in android studio, then implement some C++ code into our game. we will list the important functions below (Fig.9)



.clang-format	10/21/2022 8:09 PM	CLANG-FORMAT File
Android	10/21/2022 8:09 PM	Makefile Source File
Application	10/21/2022 8:09 PM	Makefile Source File
ChessBoard	10/21/2022 8:09 PM	C++ Source File
ChessBoard	10/21/2022 8:09 PM	C Header Source File
chess-jni	10/21/2022 8:09 PM	C++ Source File
chess-jni	10/21/2022 8:09 PM	C++ Header Source F...
common	10/21/2022 8:09 PM	C Header Source File
Game	10/21/2022 8:09 PM	C++ Source File
Game	10/21/2022 8:09 PM	C Header Source File
main	10/21/2022 8:09 PM	C++ Source File
makefile	10/21/2022 8:09 PM	File
Move	10/21/2022 8:09 PM	C++ Source File
Move	10/21/2022 8:09 PM	C Header Source File
Pos	10/21/2022 8:09 PM	C++ Source File
Pos	10/21/2022 8:09 PM	C Header Source File

Figure 9 : C++ code

It should be noted that in the game we used C++ to optimize our setup, gameplay, player and game board functions. On the other hand, we used visual code studio to combine C++ functionality with our project .

Movement of Chess Game:

According to the rules specified before for each piece, the implementation for movements in each derived class is not difficult:

- Pawn (Base Chess Pawn) can only walk correctly in three situations: move forward one square, move forward two squares only at its starting

position, move diagonally-forward one square to capture a piece of the other side. If none of these happens, then an error will be returned.

- For rook (Base Chess Rook), after receiving the starting position and the ending position, we will check whether the horizontal land vertical coordinates of the starting position and the ending position are the same or not. If no one is the same, it indicates that the moving track of rook is not a straight line, and then an error will be returned.
- Knights (Base Chess Knights) are L-shaped moves, so we only need to consider two cases: 1: move one grid in horizontal way and then walk two grid vertically 2: move two grids horizontally and then walk one grid vertically. Return an error if player violates the rules above.

- Bishop (Base Chess Bishop) can only move diagonally, so we only need to ensure that the number of squares that move horizontally after the movement equals to the number of squares that move vertically. If the player moves in other ways, an error will be returned.
- Queen (Base Chess Queen) can move in a straight line or in a diagonal line. This is just a combination of rook's rule and bishop's rule.
- The moving direction of king is the same as that of the queen, but king can only move one grid each time, so we need to ensure that the distance between the ending position and the starting position in each direction is one square. Otherwise, an error will be returned.(Fig.10)

```
Move.java
package scut_app.chess;

public class Move {
    private static final int MASK_POS = 0x3F;
    private static final int MASK_BOOL = 1;
    private static final int SHIFT_TO = 6;
    private static final int SHIFT_EP = 13;
    private static final int SHIFT_OO = 14;
    private static final int SHIFT_OOO = 15;
    private static final int SHIFT_HIT = 16;
    private static final int SHIFT_FIRSTPAWN = 17;
    private static final int SHIFT_CHECK = 18;
    private static final int SHIFT_PROMOTION = 19;
    private static final int SHIFT_PROMOTIONPIECE = 20;
    public static final int makeMove(final int from, final int to) {
        return from | (to << SHIFT_TO);
```

Figure 10:Chess Piece Movement.

Application & Functions

Before running the software, the system must verify the main activity and then launch the engine using main manifest which will connect all the asset of this software and run the program in android. In the following flow chart, a brief cycle will describe the full steps of the system functionality (Fig.11).

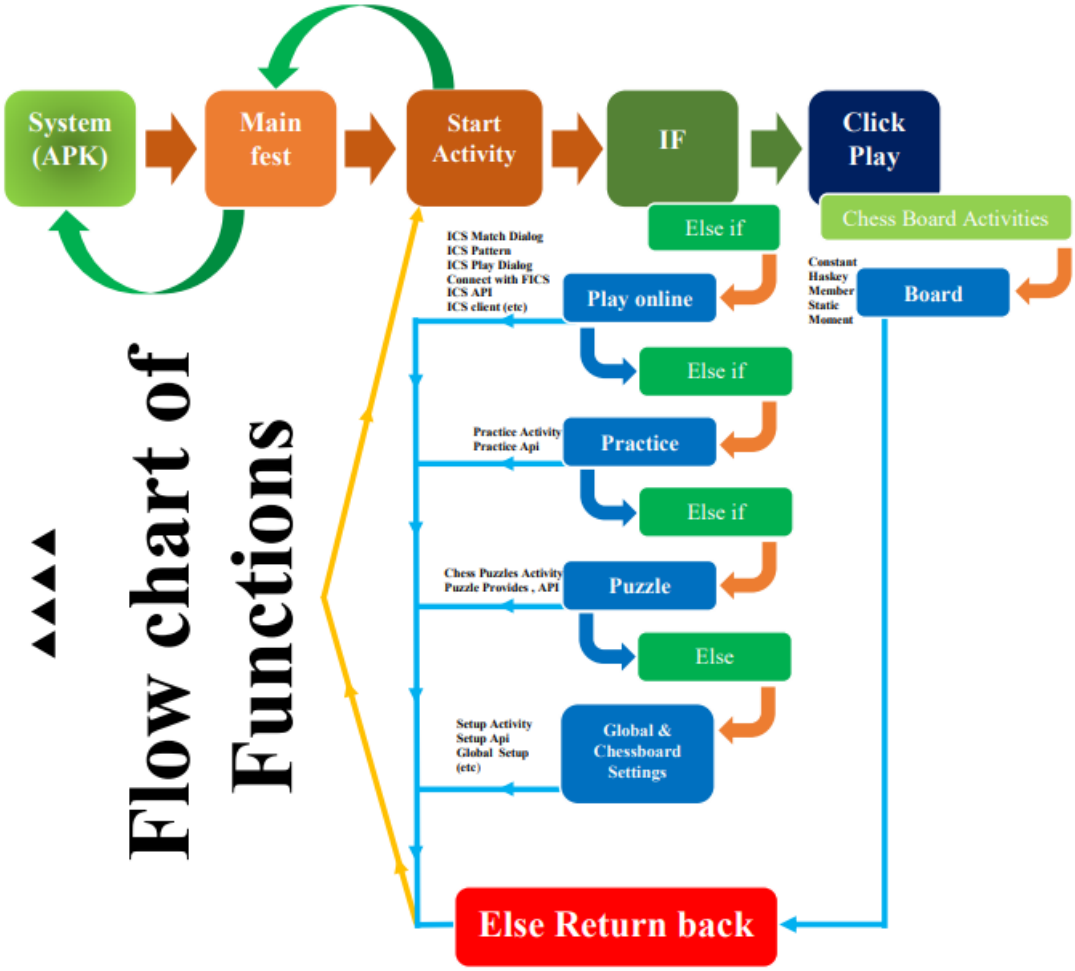


Figure 11: Flow Chart of Application.(Chess game)

7.Key Features for System:

Our program has a number of important features. And these qualities necessitate a thorough understanding of computer graphics algorithms. To save the project file as gradle, we must first determine the project file's data structure. The next sections go over the specific characteristics.

Online gaming mode

Using free internet chess server, we managed to make the game more intractable by playing with different users and challenge each other. There is a verity of Game modes which will make the game more exiting for the user.

Artificial Intelligence and User Interface:

Many of chess club features are based on Artificial Intelligence (hence AI). When two users play against each other a possible move appears by highlighting the chess tile for users who are intermediate level in chess. It can also determine if a player got checked by the opponent showing the word “check” on the top side of the screen. In addition, a move counter calculator was added to the game to show the player how many moves he made. A position detector was built with the help of Artificial Intelligence (hence AI) which can help the player to know the position of the chess piece. Unlike many chess games chess club has a nice feature that includes a classic friendly user interface at the beginning of the game.

8.Duty Assignments

We had multiple meetings for each assignment, which included programming, design, and research, to express our thoughts and profit from the provided observations to be applied to the project, in addition to having passion and motivation to work on this project. So we decided to work as a group for each task so we can gather more information and seek help from each other if facing any problems or issues. Sharing perspective played a big role to fill in the gaps in information and assist one another.

9.Programming Progress:

Phases of Mission	Period	Planned Completion	Actual Completion
Collection of Data for our project	2022.12.04 -2022.12.07	Finished assets, blueprint, texture and briefly studied the rules and ideas of chess game.	Studied the rules and ideas by finding important resources for the game.
Research on platform software's.	2022.12.8-2022.12.10	We trained to get better using android studio and unreal engine .	Main Screen background, icon and logo are designed by android studio and programming file was created and combined with the main project.
Programming journey Begins	2022.12.11-2022.12.14	Completion of all the .C++ files for chess game	All C++ files are created and some of them are included by Java program itself.
Design user interface and main screen interface and finalizing The game	2022.12.15-2022.12.19	Finishing user interface design, and implement all the functions.	Program built and ready for the testing phase.
Testing and debugging	2022.12.19-2022.05.21	Building the project file and testing the apk	Bugs corrected and the game is running smoothly and ready for delivery

10.Game Testing

During the game testing, using a physical device (android version 13) we faced some major errors which included gradle, play button, string, game launching and a error regarding the icon implementing. Therefore, we made a great effort to debug and coordinate some of the error functions. we worked together to overcome the bugs by changing some codes and making some research about it. Which eventually help it us have a solid knowledge about programing and learn from our mistakes. Eventually the functions and codes are executed correctly and have met the expectation. Screenshots of the working application will be shown below (Fig.12)

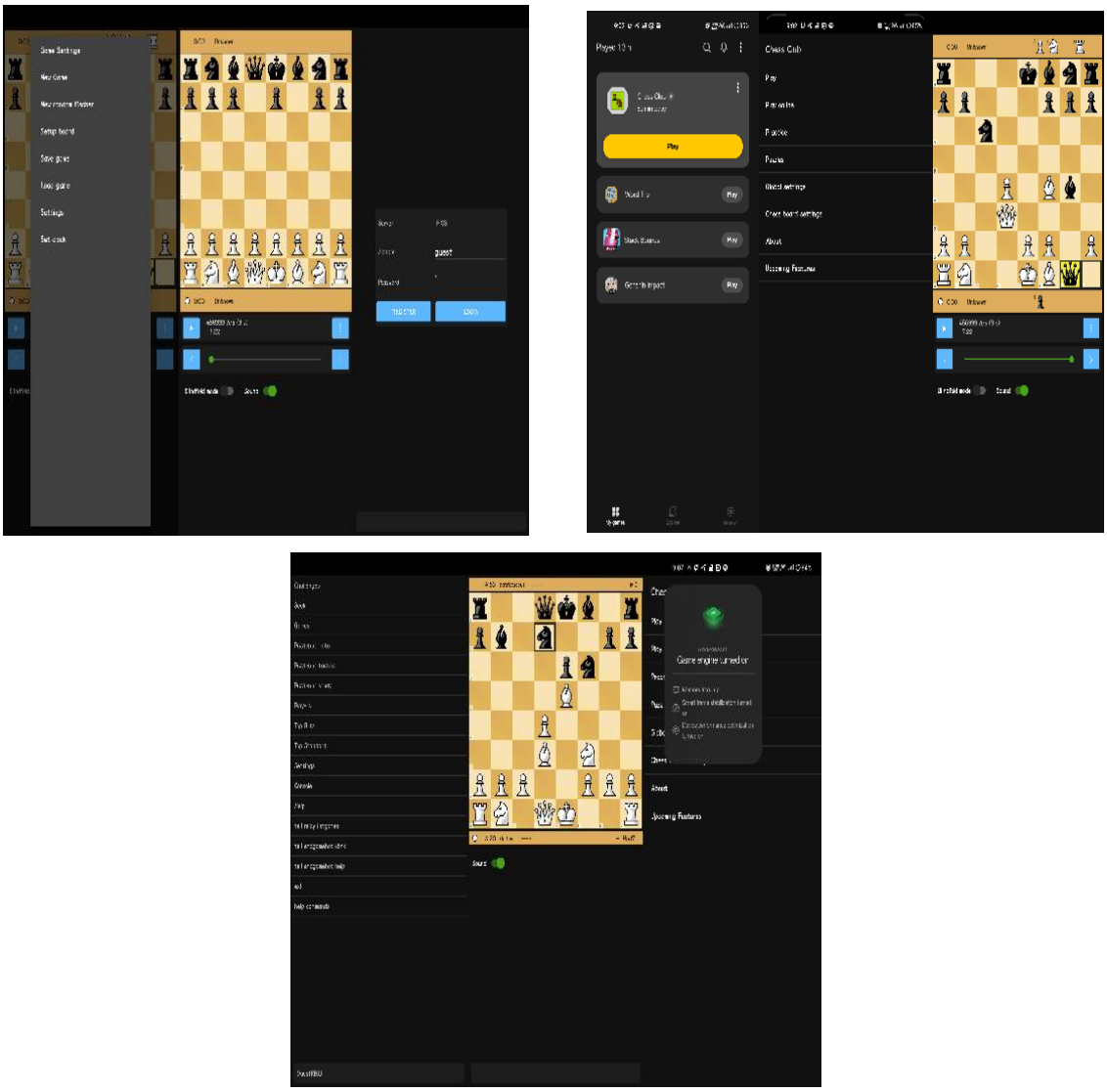


Figure 12: Showing the Game & Option.

Conclusion

In this paper, we show how we used JAVA to create the typical chess game for two players. The object-oriented language's characteristics made it easier to construct this project. We benefited a lot from the lectures and lab tasks given by professor Jing. Following the typical procedure, we create some high level abstractions of the game before implementing the specifics for each component. We learned a lot about debugging from this section. For example, via practice, we may immediately identify the incorrect code and generate a new effective concept. This knowledge will be extremely useful in our further research. Meanwhile, there are other parts of our work that may be improved. For example, we believe that we need to focus more on the graphical interface to provide a more realistic depiction of the scenario. In the future, we will also work on adding background music. On the other side, we will employ additional AI and machine learning techniques to develop a virtual opponent capable of challenging an actual player. At the end we would like to thank our professor Jing for giving us her time to share her ideas and knowledge about android studio and java programing language which helped us to develop this game.