Ultimate Chess

V1.0

EECS 22L

University of California Irvine
Henry Samueli School of Engineering

Team Bug Busters

Chris Rodriguez: Presenter
Raymond Duong: Recorder
Sunshine Jennings: Reflector
Orion Serup: Reflector
Ian Poremba: Manager

Table of Contents:

Glossary

- **Pieces:**
  - **Bishop** - Each side has **two bishops** on a white and black tile. There are **two bishops** on each side located respectively on the **first and eighth row on the players own side, three spaces away from the corners**, next to the king and queen. The bishop can only **move diagonally** along the color it starts on. It may only capture pieces of the opposing team that are in a direct diagonal line with it, and may not capture a piece that has another piece of either color in between the piece it is trying to capture and itself. It also is **incapable of "jumping"** over a piece that is in its way. Typically known in value as being worth **three points**, although this is for reference as the game is not based on points.

  - **King** - Each player has **one king**, initially located on the **first and eighth row** of either team's side of the board. It starts **next to the queen**, for the starting side it is located to the right of the queen on the 4th space from the right side of the board. The opposing team's king lies on the same column directly on the opposite side of the board. The king may **move one square in any direction**, and under special circumstances has a special move called castling. It may capture any opposing piece that is within one space of it in any direction. If the **king is being threatened by an enemy piece it must be moved to safety or protected** by another piece. If there is **no way to save the king the opponent wins** the game.

  - **Knight** - Each player has **two knights** to start the game. They are located respectively on the **first and eighth row on the players own side of the board, two spaces away from the corners**, in between the rooks and the bishops. The knight **moves in an L shape** such that it moves two spaces horizontally or vertically followed by one space horizontally or vertically, but may not move in a straight line, such that if it moves two spaces vertically it must then move one space horizontally. It may capture any opposing piece that is on a tile it may move to. Additionally the knight **can leap over other pieces**. It is typically known in value as being worth **three points**, although this is for reference as the game is not based on points.

  - **Pawn** - Each player has **eight pawns** to start the game. They are **located respectively side by side along the second and seventh row** on the players own side. Pawns can **move either two spaces forward or one space forward** from its initial position. **After its initial move, it can only move one space forward** unless it is a capture move. On capturing moves it may either **capture an opposing piece one space diagonally** and in front of it or under special

circumstances it may perform an en passant. Typically known in value as being worth **one point**, although this is for reference as the game is not based on points.

- **Queen** - Each player has **one queen**, initially located on the **first and eighth row** of either team's side of the board. It starts **next to the king**. For the starting side, it is located to the left of the king on the 4th space from the left side of the board. The opposing team's queen lies on the same column directly on the opposite side of the board. queens **can move diagonally, horizontally, and vertically** any distance up to another piece. **It may not "jump" over any piece**. Each player gets only one queen. Typically known in value as being worth **nine points**, although this is for reference as the game is not based on points.

- **Rook -** Each player has **two rooks** to start the game. They are located respectively on the **first and eighth row on the players own side on the corners** next to the knights. Rooks **can move horizontally and vertically** on the board, along with the ability to be part of castling under certain circumstances. Typically known in value as being worth **five points**, although this is for reference as the game is not based on points.

Special moves:

- **Castle** - A special move in which under certain circumstances, the king may move two spaces, instead of his normal one space toward either rook and have the rook move to the opposite side one space over from the king. This may only be done if:
    - The king has not been moved during the game.
    - The intended rook has not been moved during the game.
    - There are no pieces in between the king and the rook.
    - The king is not in check or checkmate.
    - The rook is not being threatened by an opposing piece.
    - The tiles in between the rook are not being threatened by any piece of the opposing team.

- **En Passant** - A special pawn capture move which happens after one player moves a pawn two spaces forward from its initial position. On the next turn, an enemy pawn located on the same row next to the moved pawn can then "pass" through by moving diagonally one forward. Such passing results in the pawn being captured as well. This is intended so a pawn may not get out of being captured by moving two spaces, so it is treated as though it moved only one.

❑ Rules and Board States:

- ❑ **Check** - This occurs when the **king is being threatened** by a piece of the opposing team and has at least one possible move to get out of being threatened. The player in check **must make a legal move to get out** of check during his turn. Any other move that doesn't stop the king from being in check is invalid.

- ❑ **Checkmate -** This occurs when the **king is being threatened** by a piece of the opposing team and has **no possible moves** to get out of being threatened. Thus, the game is over and the **player in checkmate loses the game**.

- ❑ **Draw** - When a draw occurs the game being played ends and neither player is deemed the winner. This may happen under 4 circumstances.
    - ❑ Both players agree to call a draw and end the game.
    - ❑ There is a stalemate.
    - ❑ The fifty-move rule is invoked.
    - ❑ There is a lack of mating material.

- ❑ **Fifty-Move Rule -** If neither a pawn has been moved nor a capture has been made in the last 50 moves. Either player can declare the match a draw

- ❑ **Lack of Mating Material** - This results in a draw and occurs when both sides are left with one of the following.
    - ❑ Only a king and one knight
    - ❑ Only a king and one bishop
    - ❑ Only king

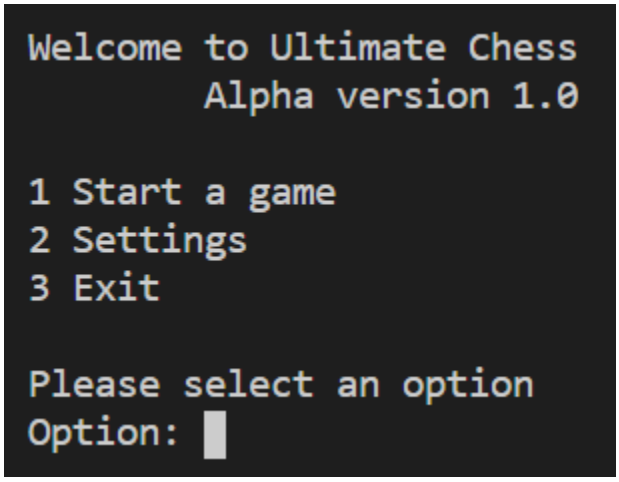
- ❑ **Stalemate -** If a player has **no legal moves** but is **not in check** then there is a stalemate and the game is declared a draw.

1 Computer Chess

1.1 Usage Scenario:

Beginning of Game

When Ultimate Chess is started it will **first show the main menu** that will provide the game modes as specified in section 3.1.



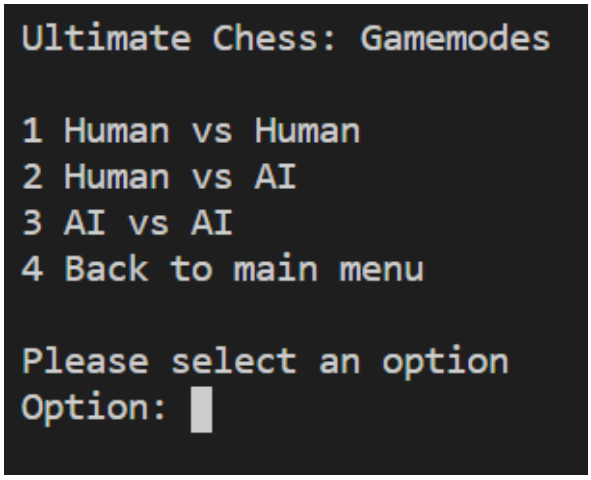


Fig.1 and Fig. 2 Game Menu with game mode options

Then the **player(s) can choose sides** and the game board will appear. An example of a chess board is shown below.

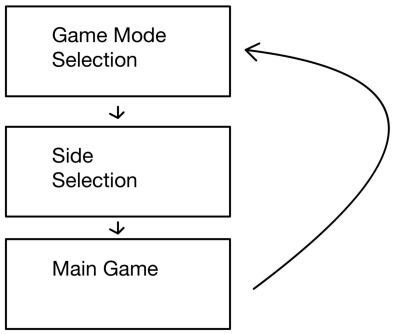


Fig.3 Diagram Showing the Flow of the Game

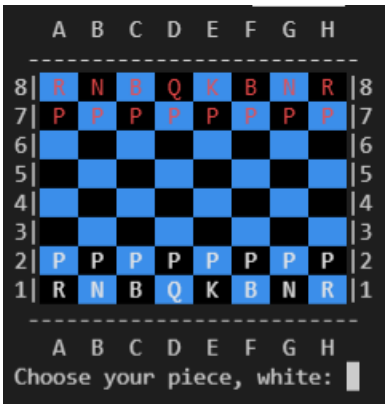
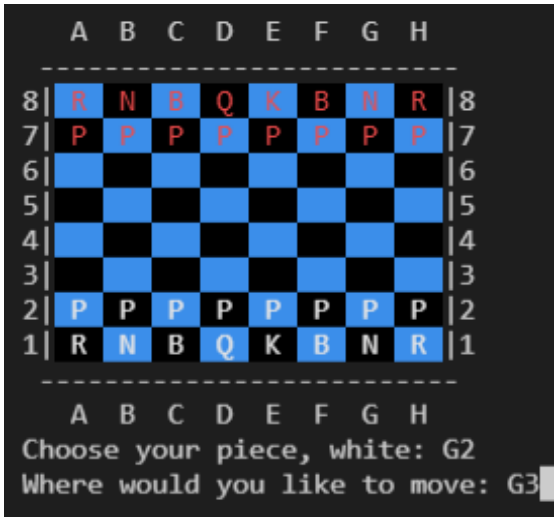


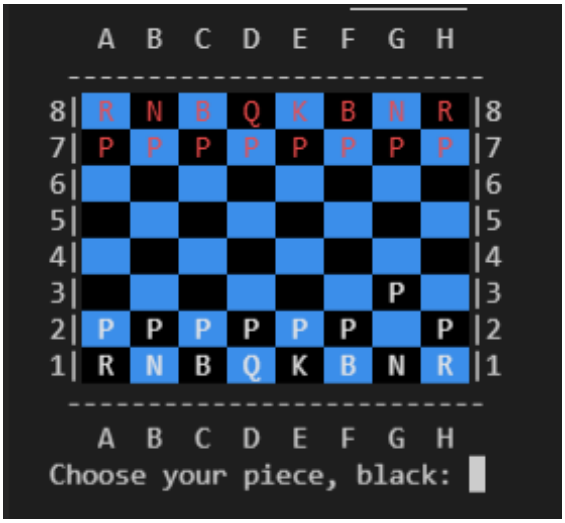
Fig.4 Default Chess Board Use Will See When a Game is Started

Middle of Game

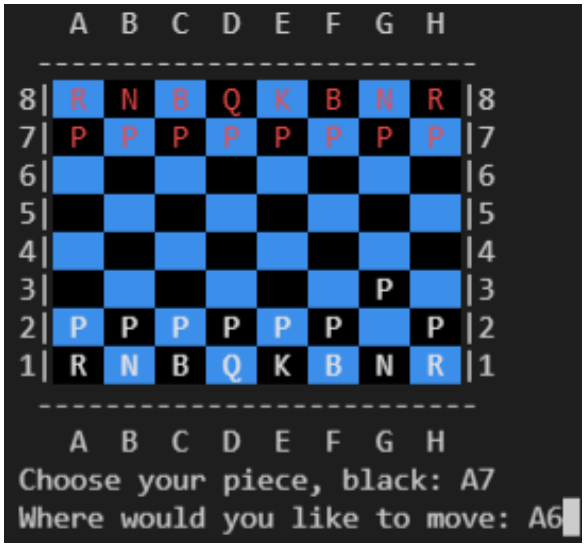
After the game has begun, **players alternate their turns** moving one of their pieces. To make a move the program **prompts the user** for an **initial position** followed by a prompt for a **final location** of **a selected piece**. If the **move is invalid**, the player will get an **error message** and be prompted for a new location to move to. Just like any real chess game, Ultimate Chess abides by all chess rules. An example scenario can be seen below



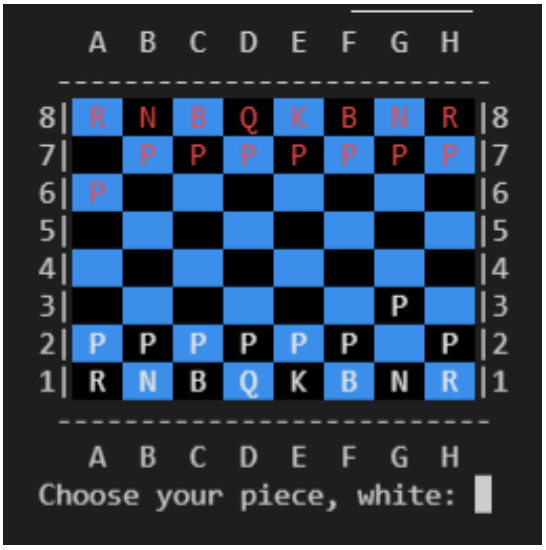
White Player first makes a move



White move made
Move Reflected on Board



Black Player first makes a move



Black move made
Move Reflected on Board

End of Game

The chess game ends when there is a **draw** triggered by the **50-move rule**, a **stalemate**, an **inadequate amount of checking pieces**, or a **checkmate**.

1.2 Goals:

The main objective of the program is to:

- Provide and display a fully functioning chess game capable of "**human vs human**," "**human vs AI**," and "**AI vs AI**" matches.
- For the game to have a functioning AI.
- Have additional features such as:
  - Being able to choose a side
  - Keeping a readable log of all moves in a text file for a single game
  - A computer player being able to make its moves under 1 minute
  - Settings menu where user is able to navigate through customization features and gameplay preferences
  - Customization for board and game pieces

1.3 Features:

- Main menu in which users can decide on game modes which includes **one player against an AI**, **two players** against each other, or **two AIs** against each other.
- Ability to **choose sides** for each player(s).
- Ability to customize game board and piece color.
- **Log of all the moves** that have been performed during the game.
- Option to **export game log** to a text file after a game.
- Option to quit the game at any point by typing "QQ"

2 Installation

2.1 System Requirements:

Computer or Server running **Linux CentOS 6.10**
Minimum Requirements:
Minimum Disk Space/Recommended - **1 GB / 5 GB**
Minimum Memory Requirement - **6273MB**

For More On Linux CentOS 6 Requirements:
https://wiki.centos.org/About/Product

2.2 Setup and Requirements:

Download all files from Github repository for the chess game. **Run "make"** in the Linux command line and then **run ./bin/Ultimate_Chess** to begin playing.

2.3 Uninstalling:

Run "make clean" to remove the object files and the executable.
Run "make uninstall" to delete all files.

```
make UltimateChess
make[1]: Entering directory '/home/ray246/Team14'
gcc -std=c99 -Wall -O3 -DNDEBUG -march=native -fomit-frame-pointer -c src/Player.c -o bin/Player.o
gcc -std=c99 -Wall -O3 -DNDEBUG -march=native -fomit-frame-pointer -c src/Board.c -o bin/Board.o
gcc -std=c99 -Wall -O3 -DNDEBUG -march=native -fomit-frame-pointer -c src/Settings.c -o bin/Settings.o
gcc -std=c99 -Wall -O3 -DNDEBUG -march=native -fomit-frame-pointer -c src/main.c -o bin/main.o
gcc -std=c99 -Wall -O3 -DNDEBUG -march=native -fomit-frame-pointer -c src/Moves.c -o bin/Moves.o
gcc -std=c99 -Wall -O3 -DNDEBUG -march=native -fomit-frame-pointer -c src/Menu.c -o bin/Menu.o
gcc -std=c99 -Wall -O3 -DNDEBUG -march=native -fomit-frame-pointer -c src/Gameplay.c -o bin/Gameplay.o
gcc -std=c99 -Wall -O3 -DNDEBUG -march=native -fomit-frame-pointer -c src/AI.c -o bin/AI.o
gcc -std=c99 -Wall -O3 -DNDEBUG -march=native -fomit-frame-pointer -c src/Game.c -o bin/Game.o
gcc -std=c99 -Wall -O3 -DNDEBUG -march=native -fomit-frame-pointer -c src/GameData.c -o bin/GameData.o
gcc -std=c99 -Wall -O3 -DNDEBUG -march=native -fomit-frame-pointer -c src/AIGameplay.c -o bin/AIGameplay.o
gcc -std=c99 -Wall -O3 -DNDEBUG -march=native -fomit-frame-pointer -c src/MoveList.c -o bin/MoveList.o
gcc -std=c99 -Wall -O3 -DNDEBUG -march=native -fomit-frame-pointer -c src/MoveValidation.c -o bin/MoveValidation.o
gcc bin/Player.o bin/Board.o bin/Settings.o bin/main.o bin/Moves.o bin/Menu.o bin/Gameplay.o bin/AI.o bin/Game.o bi
n/GameData.o bin/AIGameplay.o bin/MoveList.o bin/MoveValidation.o -flto -lpthread -o UltimateChess
make[1]: Leaving directory '/home/ray246/Team14'
```

Example Make Command Line

```
ray246@DESKTOP-7S8A263:~/Team14$ make clean
rm -f bin/*.o
rm -f bin/UltimateChess
```

Example Make clean/uninstall

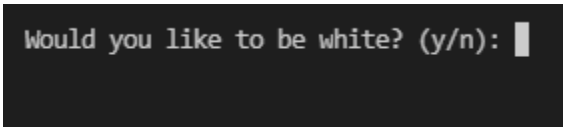
3 Chess Program Functions and Features

3.1 Game Mode Selection and Choosing Player Sides:

Ultimate Chess Game will provide the following game modes:

- ❑ **Player Vs AI**
- ❑ **Player Vs Player**
- ❑ **AI Vs AI**

If the user selects to play against a computer AI, it will prompt whether the user wishes to play the white side.



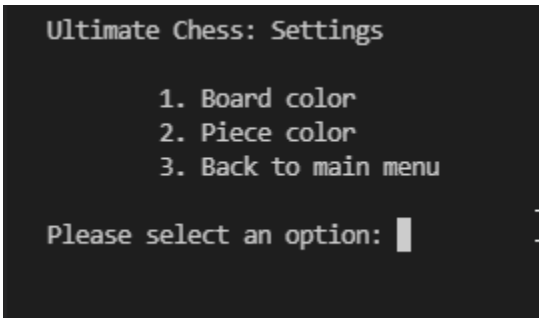
User chooses whether to play the white side or not

3.2 Multithreading Capabilities

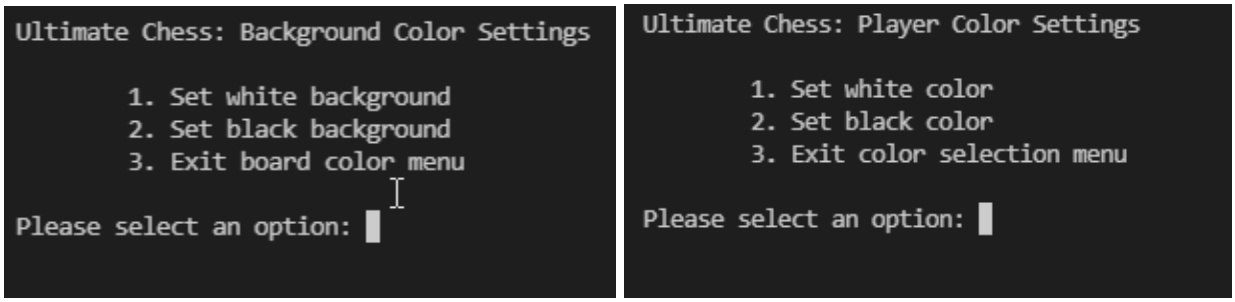
Our chess game utilizes **multithreading** which allows our program to **make most of the available CPU resources**. Multithreading allows for **multiple threads** to execute independently while **sharing their process resources**, this leads to a faster and more responsive program especially on multiprocessor machines.

3.3 Color Changing Support

Our program provides fully functioning color customization options for both the board and the pieces. Players can navigate to the settings menu where they can choose from a variety of different board and piece colors.



Player will choose whether they want to change the board or pieces color



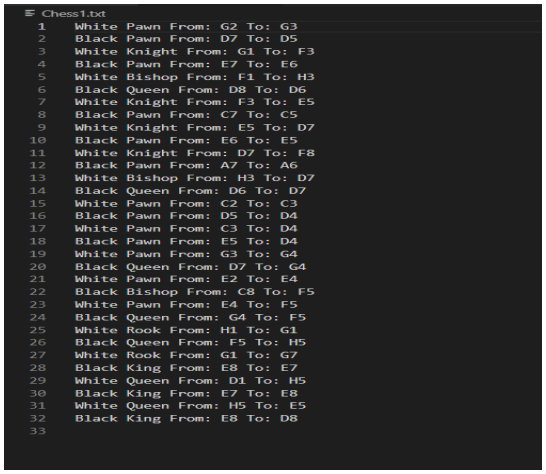
Depending on whether user decides to change board or piece color, they will have the ability to switch for both the white and black side

Players have 17 different colors to choose from

3.4 Game Log:

Ultimate Chess will have a game log that keeps track of all moves that can be seen while in game. After a game, the user(s) will have the ability to export the game log to a text file. The game log will have the following format:

[Piece color] [Piece Type] from: [Start position] To: [End Position]



Game Log example of a game played

3.5 AI:

AI will have the capability to make its moves under 1 minute and will look for a random valid move when it is the AI's turn.

Back Matter

Copyright

Error messages

Error Message	Reasoning
"Invalid piece, try again"	❏ If the player selects a position on the board that is not occupied by one of his pieces, this message will be prompted
"Invalid selection, try again"	❏ Choosing an invalid piece when promoting a pawn

“Invalid move, try again”	❑ If a player Selects a location that is not a valid move for the selected piece, this message will be prompted.

Index