**Table of Contents**

1. [**Game Overview**](#GameOverview)
2. [**Game Play Mechanics**](#GamePlayMechanics)
3. [**Controls**](#Controls)
4. [**Saving and loading**](#SavingAndLoading)
5. [**Menus and Screen Descriptions**](#Screens)
6. [**Game World**](#GameWorld)
7. [**Scripts title name**](#Scripts)
8. [**Future Features**](#FutureFeatures)

**Game Overview**

My project is a chess game! It is 2D and requires two players on the same machine, each will take turns using the mouse in order to play. The goal is to put the enemy team’s king into a checkmate, meaning the king is being attacked and cannot move out of danger with any valid move. Whoever puts the enemy king into checkmate first, wins!

**Game Play Mechanics**

Each piece has its own specific way that it is allowed to move:

Pawns can move only one space forward, unless they are in their starting position where they can move one or two spaces forward.

Knights move in an “L shape,” which consists of two spaces and then one space.

Bishops moves diagonally for any number of spaces.

Rooks move vertically or horizontally for any number of spaces.

The Queen can move in any direction for any number of spaces (except for the L shape move the knight can do).

The King can move one space in any direction, unless that move will actively put the King in danger of being captured.

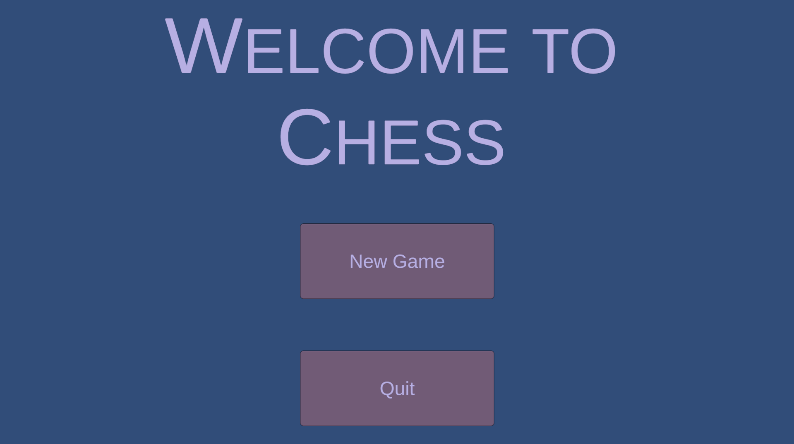
The player can castle on both King and Queen side, as well as promote their pawns to Queens when they reach the opposite side of the board with one.

**Controls**

The player will use the mouse to perform all actions in this game, such as moving their pieces, starting a new game, and quitting the game.

**Saving and loading**

It does not include saving, but you can load a new game from the main menu or from inside the game screen once you finish your last game. You do so from either pressing the “New Game” button on the main menu or clicking anywhere on the screen after the victory text appears.

1. **Menus and Screen Descriptions**

**A game of chess with a checkerboard

AI-generated content may be incorrect.**The main menu screen welcomes the player to the game and gives them two options. They can either click the new game button to start a new game of chess or the quit button to exit the application.

The game screen shows the player the chess board and all of the pieces so that they are able to play their game of chess.

**Game World**

There is a chess board populated with all the pieces in the correct starting positions. When a player clicks on the piece they want to move, purple squares will appear over any valid moves that they are allowed to make with that piece based on its movement rules.

There is text that indicates which player’s turn it is, and which color is allowed to move. There is also space to the left and right of the board where the captured pieces will appear to remind the players of how many pieces they are up or down.

**Scripts title name**

The “Game” script runs the game. It sets up the game, the initial positions of the pieces, and resets the logic of the arrays so that it will process the positions of the pieces internally. It checks for attacks, checks, and checkmates after every player’s turn by calling their respective functions and keeps the logic of the arrays updated.

The “GameManager” script oversees checking for each piece’s valid moves during each situation of the game. It is called by the Game script during each move and makes the calculations for those moves as well as moves allowed during check and checkmate.

The “GlobalGameManager” script handles the logic for displaying the captured pieces on the sides of the board.

The “MainMenuGame” script loads the Game scene when the new game button is pressed on the main menu.

The “MainMenuQuit” script handles the logic to exit the application when the quit button is pressed on the main menu.

The “MovePlate” script checks directly for logic revolving around the move plates (the purple squares that show valid moves and attacks). It checks for things like pawn promotion, potential checks (that haven’t occurred yet), and checks for if a winner exists after the previous move was completed.

**Future Features**

Future features that would be implemented include stalemate checking and En Passant. There would also be room to add an AI opponent instead of only offering the two-player option.