

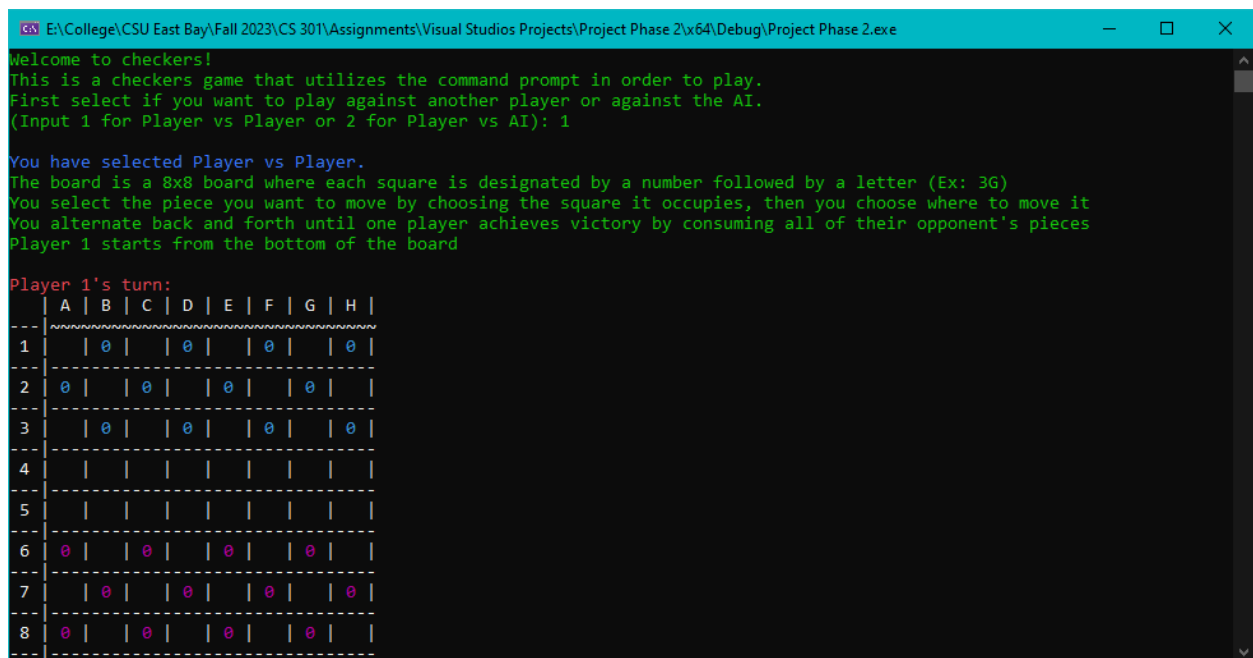
Final Report

Layers completed: 2 layers have been completed

Project Summary: Our project is to design a checkers game that takes user input to select and move their desired pieces on the board in order to win using the command prompt. The user will get to choose if they want to face off against another player on the board or against an AI. They will be prompted to choose the checker piece that they want to move and where to move it. The AI will be using a random number generator in order to make a random move that is available to it. The AI could be improved so that it will always choose the option to jump over its opponent's piece if available

User Interaction with the program:

First off, we ask the user to choose their mode. Only player vs player mode has been implemented so the user would input 1 to start the program. The program would then print out the board and ask for input (Screenshot below).



```
E:\College\CSU East Bay\Fall 2023\CS 301\Assignments\Visual Studios Projects\Project Phase 2\64\Debug\Project Phase 2.exe
Welcome to checkers!
This is a checkers game that utilizes the command prompt in order to play.
First select if you want to play against another player or against the AI.
(Input 1 for Player vs Player or 2 for Player vs AI): 1

You have selected Player vs Player.
The board is a 8x8 board where each square is designated by a number followed by a letter (Ex: 3G)
You select the piece you want to move by choosing the square it occupies, then you choose where to move it
You alternate back and forth until one player achieves victory by consuming all of their opponent's pieces
Player 1 starts from the bottom of the board

Player 1's turn:
  | A | B | C | D | E | F | G | H |
  |---|
1 |   |   |   |   |   |   |   |   |
  |---|
2 | 0 |   |   |   |   |   |   |   |
  |---|
3 |   | 0 |   |   |   |   |   |   |
  |---|
4 |   |   |   |   |   |   |   |   |
  |---|
5 |   |   |   |   |   |   |   |   |
  |---|
6 | 0 |   |   |   |   |   |   |   |
  |---|
7 |   | 0 |   |   |   |   |   |   |
  |---|
8 | 0 |   |   |   |   |   |   |   |
  |---|
```

The user interacts with the program by inputting the position of the piece they want to move. For example: they would input the row number followed by the column letter. They would then input the position of where they want to move it too. The board is printed out

once again and asks for input from player 2. It continues in this fashion until a player has won and a message gets printed out (Screenshot below).

```
E:\College\CSU East Bay\Fall 2023\CS 301\Assignments\Visual Studios Projects\Project Phase 2\64\Debug\Project Phase 2.exe
Pick the checker piece you would like to move by inputting the row number and column letter(Ex: 3G): 7d
Please choose which square you want to move the piece to: 4f

Player 2's turn:
  A | B | C | D | E | F | G | H |
  ~~~~~
1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
  ~~~~~
2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
  ~~~~~
3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
  ~~~~~
4 | 0 | 0 | 0 | 0 | 0 | x | 0 |
  ~~~~~
5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
  ~~~~~
6 | x | 0 | x | 0 | x | 0 | x | 0 |
  ~~~~~
7 | 0 | x | 0 | 0 | 0 | x | x | 0 |
  ~~~~~
8 | x | 0 | x | 0 | x | 0 | x | 0 |
  ~~~~~
Pick the checker piece you would like to move by inputting the row number and column letter(Ex: 3G):
```

What has proved to be harder (or easier):

- We encountered several challenges and design revisions. One of the significant aspects that was harder than expected was the Piece object related information. In phase one of our project, we focused on implementing the move function for moving the piece. However, as the development process progressed we realized that additional functionalities were required. To address these challenges we implemented the following new functions: `vector<Piece> Piece::updatePiece`, `Piece::returnID`, `int Piece::returnRow`, `int Piece::returnColumn` and `string Piece::returnLabel`.
- We also came across the issue of the checkerboard looking too plain and the spaces being indistinguishable from one another. This required us to search for a method to colorize the board and make it look more presentable.
- Another challenge we encountered was carrying the different data into different files and making sure it was updated correctly in the main function to reflect those changes