Ali Rahman

Checkers A.I.

Instructions:

The program was compiled and tested on Windows 10 using g++ Version 7.3.0 running on Cygwin. In order to compile the files, there is a provided makefile to generate Game.exe which is the program to play the game in any form. No arguments are necessary to run the program.

Implementation:

The program is broken up into 7 files which are a combination of classes, header files, and the main program. The game is run through Game.cpp which specifies the details of the game such as the time limit, board, number of players, and which player goes first. The move and piece classes and header files are simple structures that represent the information used to determine the legal moves, board state, and heuristic. The bulk of the code is within the CheckerBoard class which contains the definitions for the functions that represent the construction of the game, display of the board on the UI, the heuristic, updating the game board, checking multi-jumps and determining the legal moves.

Features:

The program allows the user to specify if two, one or zero humans are going to play the game, whether the computer or the human player goes first if only one human is playing, and a time limit that’ll determine the difficulty of the computer

Details:

The heuristic for the AI is done in two stages, early/mid game and late game depending on the player with the advantage.

In order to load a board, there is a specified format to use which is to specify the index of square on the board with a 0 for empty, 1 for a player 1 piece, 2 for a player 2 piece, 3 for a player 1 king, and 4 for a player 2 king. The general format is to use the dark squares which are specified by non-zero numbers in the testboard.txt file (See provided testboard.txt file). If an invalid number is specified, the program will not run correctly, and if an invalid position is selected for a piece (a white square), the board will still be displayed but the game will not run correctly.

The game program checks for certain error handling but has certain pre-defined default values for aspects of the game such as the game board, number of players/computers, which player goes first, and time limit for the AI. These values are listed when the program prompts the user, but if an invalid parameter is specified, the default value is used.

The board itself is displayed using ascii characters where Player 1 is red, Player 2 is white, and kings are identified with an O on the piece.

I had tried implementing a heuristic that would satisfy a victory in the two kings advantage with a double corner but it does not win, even though it should not lose given the advantage.