Project 1: A Checkers Agent

Ramakrishna Oruganti

The checkers agent works submitted consists of the following files:

- MainGame.java
- AlphaBeta.java
- CheckersGame.java
- Move.java
- Pair.java

The project was developed using Eclipse IDE.

MainGame.java consists the main function which should be run to start the game. The user will input his move and can select if he wants to start the game or play second. The user will be given a list of moves from which he can choose the move he wants to play, by selecting the corresponding number for the move.

AlphaBeta.java consists of the minimax algorithm with alpha-beta pruning. It also has the evaluation function.

CheckersGame.java consists of printing the checkers board on the console and gather the list of legal moves both slide and jump moves.

The pieces are denoted by P on the board and when a piece turns into a king, it will be denoted by K on the board. The default search depth has been set to 15.

Below is the table for increasing search depth and time taken by the AI to make the move.

Max Search Depth	**Time taken per move (in seconds)**
5	0
10	0-2
15	45-60
20	> 180
25	> 300

As we can see for maximum search depth 5 and 10, the AI hardly takes any time as it has very less options to choose from. Hence, the moves it makes is not optimal as it does not have enough options and choses the move with the best evaluation value at that depth.

As the depth increases, the AI has more options to evaluate and choose from and it takes more time to decide and make a move. However, this means it makes better moves than when the depth is less.

Improving the evaluation function means that the AI makes better decisions regarding its moves and plays optimally. It takes into account the current status of the board and the number of pieces of the player and makes a decision about its next move.