Kiah Johnson

CS 510

Assignment 4

For this assignment, I was assigned to create a minimax algorithm that will make an AI agent play the game Othello. I was given the code to represent the board, handle movements, and determine whether the game is over, with the score being displayed. I was not allowed to use any external libraries or packages, except the built in standard libraries and NumPy. My program for this assignment implements 4 versions of the game: human player(s), random player(s), minimax agent, and minimax with alpha beta pruning agent. In addition to this, I have another minimax with alpha beta pruning agent that can return a solution within a given time.

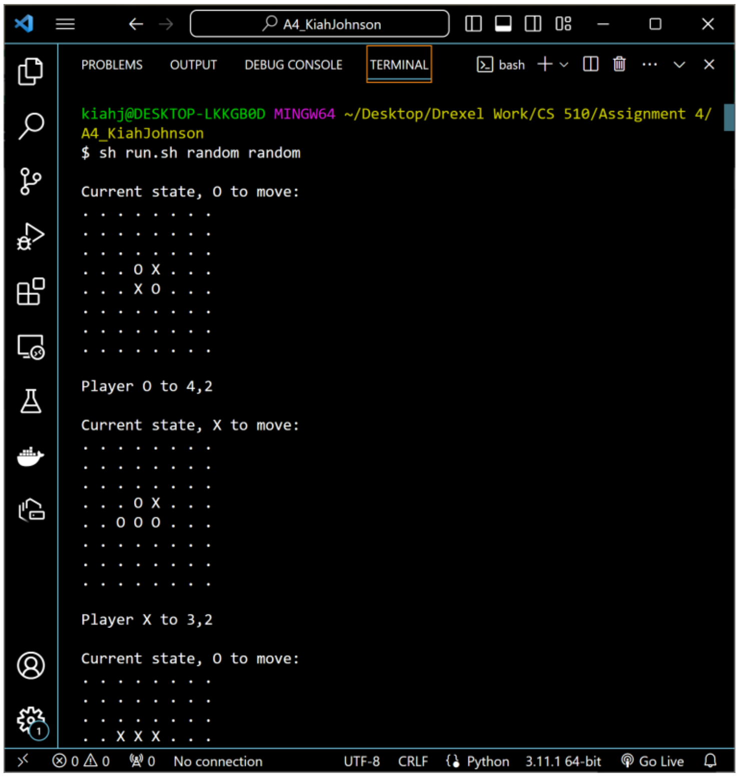
The first to be implemented is the human agent, which does not have a defined time complexity because it relies on human input. This means the time it takes the human to decide on a move depends on the human playing the game, which is unpredictable. The next agent to be implemented is the random agent, which has a time complexity of O(1), which means the runtime will always be the same, regardless of what choice is made beforehand. This is because this agent just randomly selects a move from the available list, so the time it takes to choose a move is constant, it doesn't take any additional work to decide on a move. The third agent is the minimax agent having the time complexity of O(b^d). Variable b is the branching factor and d is the depth of the search. This means that with this algorithm, the agent explores the entire game tree up to a certain depth, and the time grows exponentially with the depth of the search and the number of moves. The last two agents implemented, the alpha beta pruning and kj645 both have the time complexity of O(b^(d/2)). It has the same variables as described in the minimax agent, the only difference is that it reduces the minimax tree nodes by eliminating branches that cannot help the agent win the game.

As my final product, I turned in the project with the additions of my implemented agents which are RandomAgent, MinimaxAgent, AlphaBeta, and KJ645. Each class is described below:

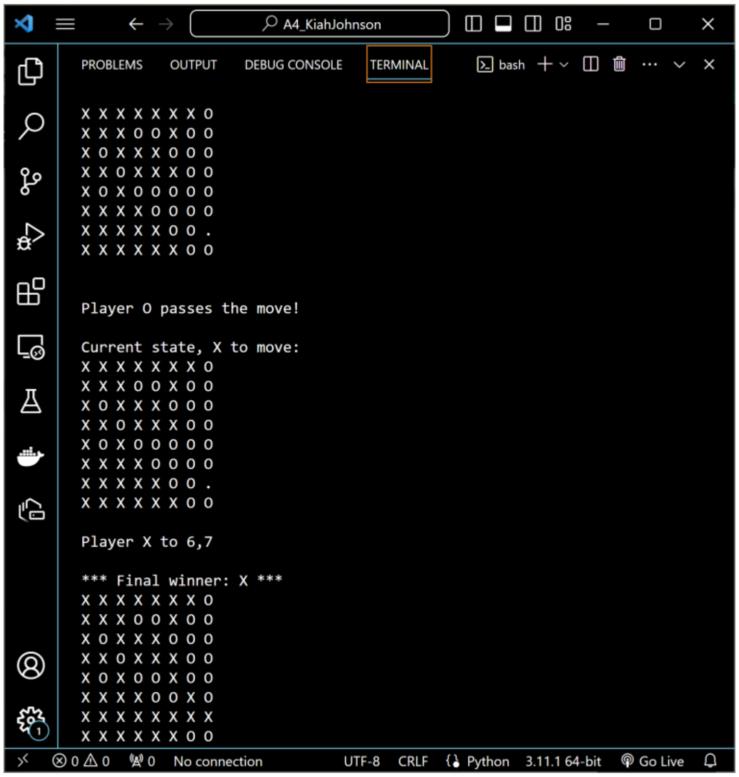
1. RandomAgent: implements an agent that plays Othello by choosing moves at random
2. MinimaxAgent: implements an agent that plays Othello using the standard minimax algorithm

3. AlphaBeta: implements an agent that plays Othello using minimax algorithm with alpha beta pruning
4. KJ645: implements an agent that plays Othello using minimax and algorithm with alpha beta pruning, just within a set time limit

Beginning/End result using random command:

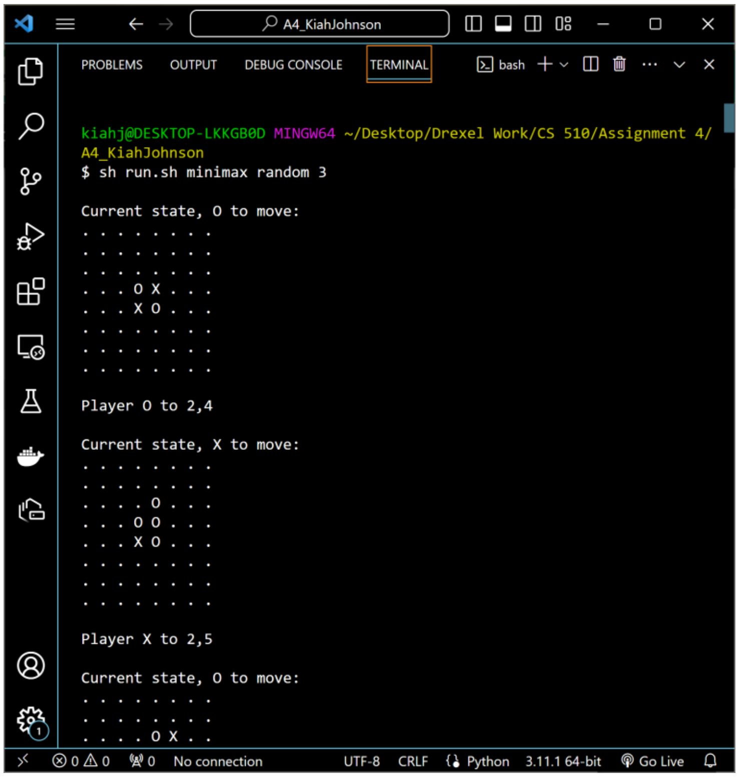


```
kiahj@DESKTOP-LKKGB0D MINGW64 ~/Desktop/Drexel Work/CS 510/Assignment 4/
A4_KiahJohnson
$ sh run.sh random random

Current state, O to move:
. . . . . . . .
. . . . . . . .
. . . . . . . .
. . . O X . . .
. . . X O . . .
. . . . . . . .
. . . . . . . .
. . . . . . . .

Player O to 4,2

Current state, X to move:
. . . . . . . .
. . . . . . . .
. . . . . . . .
. . . O X . . .
. . O O O . . .
. . . . . . . .
. . . . . . . .
. . . . . . . .

Player X to 3,2

Current state, O to move:
. . . . . . . .
. . . . . . . .
. . . . . . . .
. . X X X . . .
```

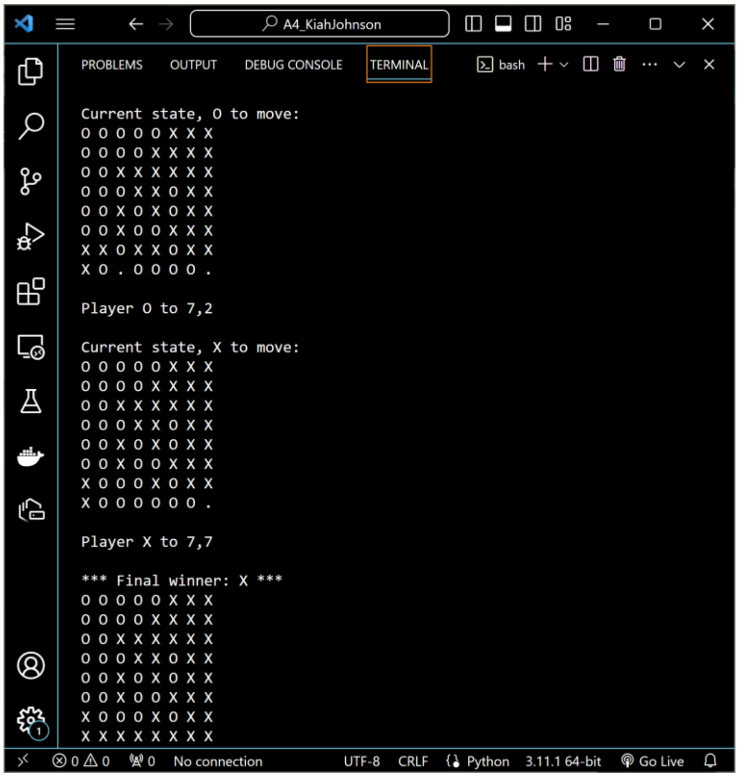


```
X X X X X X X O
X X X O O X O O
X O X X X O O O
X X O X X X O O
X O X O O O O O
X X X X O O O O
X X X X X O O .
X X X X X X O O


Player O passes the move!

Current state, X to move:
X X X X X X X O
X X X O O X O O
X O X X X O O O
X X O X X X O O
X O X O O O O O
X X X X O O O O
X X X X X O O .
X X X X X X O O

Player X to 6,7

*** Final winner: X ***
X X X X X X X O
X X X O O X O O
X O X X X O O O
X X O X X X O O
X O X O O X O O
X X X X O O X O
X X X X X X X X
X X X X X X O O
```

Beginning/End result using minimax command:

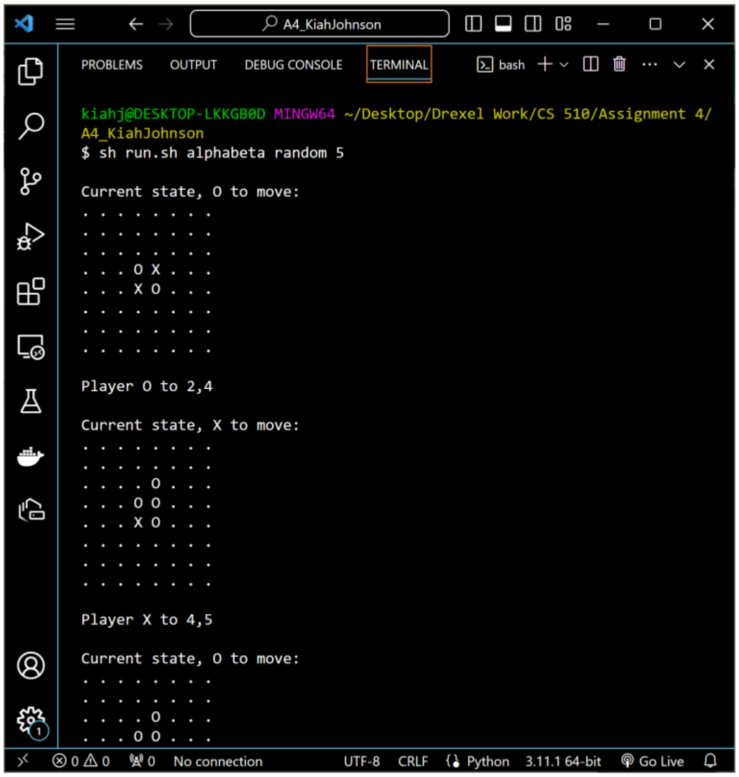


```
kiahj@DESKTOP-LKKGB0D MINGW64 ~/Desktop/Drexel Work/CS 510/Assignment 4/
A4_KiahJohnson
$ sh run.sh minimax random 3

Current state, O to move:
. . . . . . . .
. . . . . . . .
. . . . . . . .
. . . O X . . .
. . . X O . . .
. . . . . . . .
. . . . . . . .
. . . . . . . .

Player O to 2,4

Current state, X to move:
. . . . . . . .
. . . . . . . .
. . . . O . . .
. . . O O . . .
. . . X O . . .
. . . . . . . .
. . . . . . . .
. . . . . . . .

Player X to 2,5

Current state, O to move:
. . . . . . . .
. . . . . . . .
. . . . O X . .
```

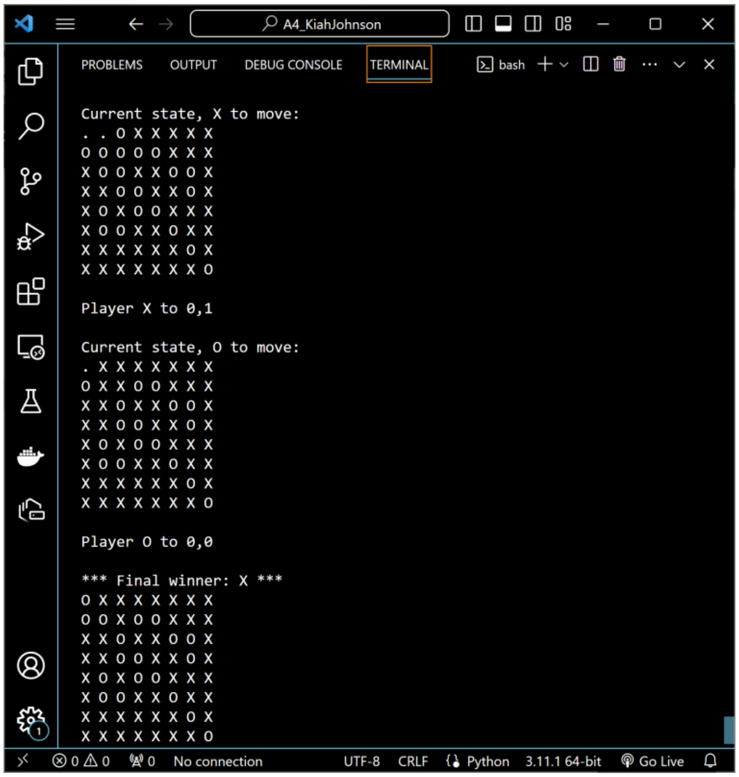


```
Current state, O to move:
O O O O O X X X
O O O O X X X X
O O X X X X X X
O O O X X O X X
O O X O X O X X
O O X O O X X X
X X O X X O X X
X O . O O O O .

Player O to 7,2

Current state, X to move:
O O O O O X X X
O O O O X X X X
O O X X X X X X
O O O X X O X X
O O X O X O X X
O O X O O X X X
X O O O X O X X
X O O O O O O .

Player X to 7,7

*** Final winner: X ***
O O O O O X X X
O O O O X X X X
O O X X X X X X
O O O X X O X X
O O X O X O X X
O O X O O X X X
X O O O X O X X
X X X X X X X X
```

Beginning/End result using alphabeta command:

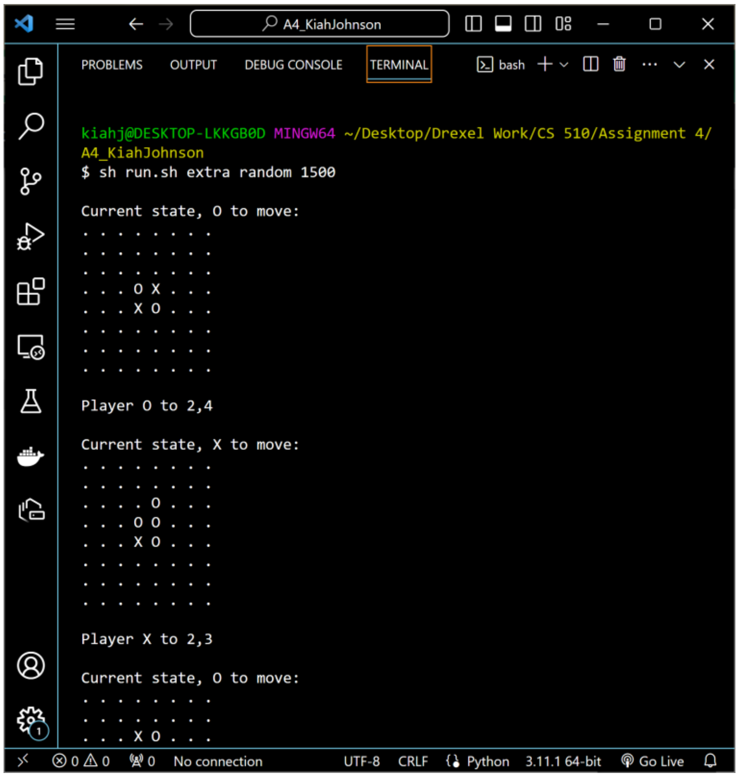


```
kiahj@DESKTOP-LKKGB0D MINGW64 ~/Desktop/Drexel Work/CS 510/Assignment 4/
A4_KiahJohnson
$ sh run.sh alphabeta random 5

Current state, O to move:
. . . . . . . .
. . . . . . . .
. . . . . . . .
. . . O X . . .
. . . X O . . .
. . . . . . . .
. . . . . . . .
. . . . . . . .

Player O to 2,4

Current state, X to move:
. . . . . . . .
. . . . . . . .
. . . . O . . .
. . . O O . . .
. . . X O . . .
. . . . . . . .
. . . . . . . .
. . . . . . . .

Player X to 4,5

Current state, O to move:
. . . . . . . .
. . . . . . . .
. . . . O . . .
. . . O O . . .
```

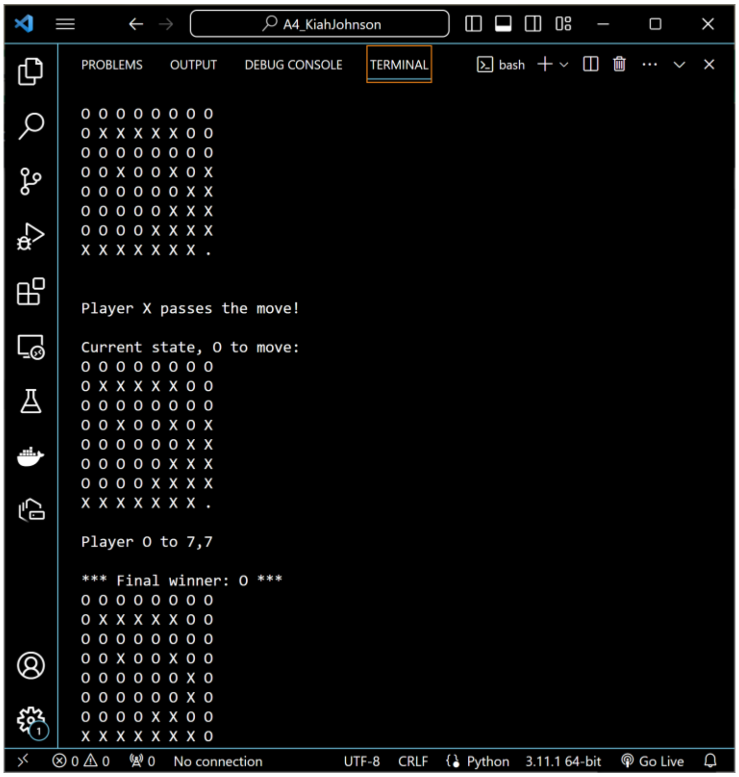


```
Current state, X to move:
. . O X X X X X
O O O O O X X X
X O O X X O O X
X X O O X X O X
X O X O O X X X
X O O X X O X X
X X X X X X O X
X X X X X X X O

Player X to 0,1

Current state, O to move:
. X X X X X X X
O X X O O X X X
X X O X X O O X
X X O O X X O X
X O X O O X X X
X O O X X O X X
X X X X X X O X
X X X X X X X O

Player O to 0,0

*** Final winner: X ***
O X X X X X X X
O O X O O X X X
X X O X X O O X
X X O O X X O X
X O X O O X X X
X O O X X O X X
X X X X X X O X
X X X X X X X O
```

Beginning/End result using extra command:



```
kiahj@DESKTOP-LKKGB0D MINGW64 ~/Desktop/Drexel Work/CS 510/Assignment 4/
A4_KiahJohnson
$ sh run.sh extra random 1500

Current state, O to move:
. . . . . . . .
. . . . . . . .
. . . . . . . .
. . . O X . . .
. . . X O . . .
. . . . . . . .
. . . . . . . .
. . . . . . . .

Player O to 2,4

Current state, X to move:
. . . . . . . .
. . . . . . . .
. . . . O . . .
. . . O O . . .
. . . X O . . .
. . . . . . . .
. . . . . . . .
. . . . . . . .

Player X to 2,3

Current state, O to move:
. . . . . . . .
. . . . . . . .
. . . X O . . .
```



```
O O O O O O O O
O X X X X X O O
O O O O O O O O
O O X O O X O X
O O O O O O X X
O O O O O X X X
O O O O X X X X
X X X X X X X .


Player X passes the move!

Current state, O to move:
O O O O O O O O
O X X X X X O O
O O O O O O O O
O O X O O X O X
O O O O O O X X
O O O O O X X X
O O O O X X X X
X X X X X X X .

Player O to 7,7

*** Final winner: O ***
O O O O O O O O
O X X X X X O O
O O O O O O O O
O O X O O X O O
O O O O O O X O
O O O O O O X O
O O O O X X O O
X X X X X X X O
```