Homework #3—Minimax

In class—We will go over the code: game.py, play.py, and alphaBetaPruning.py for Tic-Tac-Toe. Please .make sure you understand these files

A. Run the game when the search depth (DEPTH) is 9 and also where it is 2. Is there a difference in the runtime and the moves that are generated during the different searches?

B. It is generally accepted that the best opening move is the center square. When the search depth is 2 the computer agent plays the center square but when the depth is 9 it takes the corner. Why?

For Homework #3:

Base yourself on the provided code to implement Othello (also known as Reversi) on an 8X8 board. Your agent should play against a human opponent using the base code with Minimax and Alpha-Beta Pruning.

In order to implement this agent, you will need to do the following:

1. In the provided code there the value function is missing and therefor the agent plays poorly. You need to write an intelligent, heuristic-based value function that takes as its input the current state of the board, as is defined in a comment in the game.py file, and returns a value. Positive values are good for the agent, and negative value are good for the human-player.
2. In the provided code the first player always plays first. According to the rules of the game, black always plays first. Write the function whoIsFirst which takes as its input the game state, as is defined in a comment in game.py, and asked the human player if they want to be black or white. Based on this response, the game will choose if the human goes first or second. Note that this decision will impact the implementation as X's and O's will need to be black and white based on the user's decision. This is currently not implemented, but the whoIsFirst function does have a comment where this code will hook in and after you do so, feel free to erase this comment and otherwise this file does not need to be changed.
3. There is no condition for ending the game in the sample code. Complete the function isFinished which accepts a state as input to address this point as is described in a comment in game.py. The code will need to check if the game ended and return True if that is the case.

Unless otherwise noted, you only need to make changes to the file game.py. This code should work seamlessly with the alphaBetaPruning file and also with the play file after you have made the edits described above in item #2.

Make sure you fully comment your code. In your comments include a description about the heuristic you use and the justification for its logic. Once you are done, submit the game.py file and push it to Git by November 6th at 11:59 PM.

Grade Breakdown:
50% -- Implementation of the value function
20% -- Implementation of the whoIsfirst function
15% -- Implementation of the isFinished function
15% -- Documentation