Foundations of AI
Yunyu Guo
Assignment 6
Due October 28

This assignment is to write a solver for the game of Tic Tac Toe using the Minimax algorithm.

- Questions specific to this assignment:
  - In `solve_opponent_move()`, why do we return the opposite (negative) of the score used in the base case of `solve_my_move()`?

    Because the player's win is the opponent's loss, if player wins, they get 1 reward, which is -1 for the opponent.

  - Why is there no winner if both sides are playing optimally?

    The 3x3 board is small enough that every possible threat can be blocked. If one player plays optimally, the opponent can always play defensively to prevent the player from creating a wining line.

```
None|None|None
None|None|None
None|None|None

X's turn

Current board:
Player.X|None|None
None|None|None
None|None|None

O's turn

Current board:
Player.X|Player.O|None
None|None|None
None|None|None

X's turn

Current board:
Player.X|Player.O|Player.X
None|None|None
None|None|None

O's turn

Current board:
Player.X|Player.O|Player.X
Player.O|None|None
None|None|None

X's turn

Current board:
Player.X|Player.O|Player.X
Player.O|Player.X|None
None|None|None

O's turn

Current board:
Player.X|Player.O|Player.X
Player.O|Player.X|None
Player.O|None|None

X's turn

Current board:
Player.X|Player.O|Player.X
Player.O|Player.X|Player.X
```

```
O's turn

Current board:
Player.X|Player.O|Player.X
Player.O|Player.X|Player.X
Player.O|None|Player.O

X's turn

Final board:
Player.X|Player.O|Player.X
Player.O|Player.X|Player.X
Player.O|Player.X|Player.O

It's a draw
```

- This pseudocode includes alpha-beta pruning. What would be the pseudocode for implementing this same algorithm, but without alpha-beta pruning?

function minimax(node, maximizingPlayer):
If node is a terminal node: node.value = score
Else if maximizingPlayer:
node.value = -inf
For each child of the node:
node.value = max(node.value, minimax(child, false))
Else
node.value = inf
For each child of the node:
node.value = min(node.value, minimax(child, true))
Return node.value

- The usual questions:
  - How long did this assignment take you? (1 sentence)
    1 day

  - Whom did you work with, and how? (1 sentence each)
    - Discussing the assignment with others is encouraged, as long as you don't share the code.

    I discuss with a student regarding the solve_my_move method, in the requirements, it says "alpha should be updated to be the smallest best_move value found so far (using the cumulative sum pattern)". We think it should be the maximum(alpha, best_move.value), while alpha means "the minimum score that player can get through that path".

  - Which resources did you use? (1 sentence each)
    - For each, please list the URL and a brief description of how it was useful.
      Course slides

  - A few sentences about:
    - What was the most difficult part of the assignment?
      What is the alpha? I think it should be the max(α, node.value)

    - What was the most rewarding part of the assignment?
      Practice using the minimax algorithm and alpha beta pruning

    - What did you learn doing the assignment?
      Use minimax to determine the best move to take on every turn, and the value we will end up with. Use the alpha beta pruning to reduce the search space.

- Constructive and actionable suggestions for improving assignments, office hours, and class time are always welcome.