Tic Tac Toe vs Artificial Intelligence Report

Project Statement

- The goal of this project was to simulate a tic tac toe game, and then simulate an AI playing against you as your opponent. I only realized later that I could make use of the minimax algorithm, to make the AI unbeatable, in that it either wins or forces a draw.

Algorithm Overview

- The main algorithm used in this project is the Minimax algorithm, which is a recursive process. It evaluates future game states, alternating between maximizing the AI's chances of winning, and then minimizing the user's chances.
- As it is a recursive algorithm, it consists of:
    - Base Cases
        - If the AI wins, it returns a score of +1
        - If the user wins, it returns a score of -1
        - If the game ends in a draw, it returns a 0
    - Recursive Cases
        - When it is the AI's turn, the algorithm looks at all possible moves and selects the one with the highest score.
        - When it is the user's turn, the algorithm looks at all possibles moves and selects the one with the lowest score.

Pseudocode

```
minimax(grid, depth, is_maximizing):
    if ai_win
        return 1
    if user_win
        return -1
    if draw
        return 0

    if is_maximizing:
        best_score = - infinity
        for move in valid_moves(grid):
            hypo_grid = play(grid, move, AI)
            score = minimax(hypo_grid, depth + 1, False)
            best_score = max(best_score, score)
        return best_score

    else:
        best_score = infinity   #
        for move in valid_moves(grid):
            hypo_grid = play(grid, move, user)
            score = minimax(hypo_grid, depth + 1, true)
            best_score = max(best_score, score)
        return best_score
```

Analysis

- This project was particularly relevant to chapter 5 of our class, in which we discussed adversarial games, and adversarial search. We went over multiple exercises on how to implement minimax, as well as many slides.
- Tic-tac-toe is a deterministic, zero-sum game, letting us make use of Minimax. Players alternate turns, and it involves a state-space search tree.
- Strengths
    - It will always win, or prevent the other player from winning (draw).
    - It shows off the minimax algorithm pretty well
- Weakness
    - It does not make use of alpha-beta pruning, which would improve the efficiency of the minimax algorithm

Summary

- Overall I’m really happy with this project. I started off without the minimax algorithm in mind, but after I developed a really simple algorithm, I realized we went over the perfect concept for designing an AI for tic tac toe, and decided to go with it. In my testing, the AI causes the player to not win 100% of the time. I think I could have made the input system from the user more intuitive, but it still works well.