Mancala Final Report

In this project, our goal was to implement the game of Mancala with the provided ruleset to evaluate the performance of 3 different types of players against a random player across 100 games. The three types of player we implemented are random, Minimax AI, and Alpha-Beta AI.

Our code builds off of our work from Homework 6, where we implemented a simple Mancala game. We implemented our own minimax algorithm that determines, using a given utility function, the best move to make, as well as an alpha-beta algorithm that uses minimax values to determine the best move in a shorter amount of time (by pruning). For extra credit, we also modified the utility function to improve the success rate of the AI player using alpha-beta pruning.

To test our code, we used print statements to determine what player was playing and what move they were making, as well as printing out whole games, which was especially useful when checking the capture implementation. We also initially did not set a seed for our code, allowing us to see varying values of win rates for each player. By determining if these values were high or low, we were able to determine whether or not our code was working correctly.

We have submitted a .ipynb file where our code is included as well as written explanations as to what is happening. We have subtitles on our document indicating what test is being run. So, if each cell were run in the .ipynb file, the results will be printed, and an explanation of these results can be found below.

Overall, we learned how important utility functions are, as well as how to implement minimax and alpha-beta for a two player game. Since we implemented our own minimax and alpha-beta algorithms, we both gained a much better understanding of each algorithm and how it works, as well as how it returns the best move for our AI mancala player. We also saw the importance of the utility function come into play, as during our testing phase we saw interesting win and tie ratios, especially when we started messing around with the level of plies. We also gained practice writing a utility function, as we made our own for the extra credit problem, which took into account the previous utility function, and gave weight to the proportion of moves for each player. This project helped us understand the fundamentals in creating an AI player that learns from the game using concepts we have learned in class: utility functions, minimax, and alpha-beta pruning.