Comp 4106

Assignment 2

Owen Craston

101037159

**Game State:**

This is a connect four implementations in python. The core functionality of the game revolved round a single entity, the board. The board is represented as a 2D array or matrix of the specified size. In the case of this implementation we are using a 6x7 board. The board has 6 rows that are each represented as a list of size 7 and the columns cute through each row list. The game begins with the random player making a move. As the name suggests, the random player is a player that chooses to move at the first valid random move it finds. This player plays against the AI which implements a minimax search algorithm with alpha beta pruning. The algorithm defends against other players winning moves (it blocks the other player from winning if it can) above all, otherwise it uses a scoring function to determine the maximum score in the tree node. The scoring heuristics vary in the versions of this game but they will be discussed further in this report.

**Heuristic 1:**

The first heuristics function is one that gives points for having multiple pieces in a row. This scoring algorithm works by parsing all the possible four consecutive slots in every direction and then passing it to a scoring algorithm. The scoring algorithm is simple as it simply adds five points to the score if there are three pieces in said window and two points for two pieces in the window.

**Heuristic 2:**

Placing pieces in the most center positions of the board as possible is ideal for winning in connect four. The reason for this is that the more pieces you have towards the center the more options for the player to play around these pieces. Because of this I built a heuristics function that gives highest score for placing pieces directly in the center (+= 10 points per item in center). The scoring of placing a piece on the next most center column (middle +- 1) is two less than the max score. This trend continues until you reach the first and last column which the player only gets four points per item in these columns.

Summary:

Heuristic one performs better than heuristic 2 since its scoring system is more important to the game. This is because placing pieces in the center does give the player more opportunities but the entire goal of the game is based on getting as many pieces in a row as possible (until we achieve the final state of four in a row) so using this to score moves by players is closer to measuring what the point of the game really is.

**Stats:**

Depth: 4

Heuristic 1:

Node count on initial turn: 281, 336, 349, 349, 244, average = 311.8

Heuristic 2:

Node count on initial turn: 623, 623, 623, 623, 623, average = 623

Initially, the first heuristic is about twice as fast as Heuristic 2.