AI Homework 4

Members:

- Songxi Chen
- Jim Yang

Four-in-a-row AI with minimax algorithm

Heuristic Implementation

- **Check for win**

We have our board setup as a two-dimensional array. Nested for-loops are used to check whether a player has won. For example, the horizontal test only checks for columns from index 0 to 2 because you can't start at column index 3 to get 4-in-row. For the vertical test it is similar where we check rows from index 0 to 1 because starting at row index 2 will not get 4-in-row when checking downwards.

- **Heuristic evaluation functions**

In the same heuristic function we have our heuristic evaluations. In each for loop a heuristic variable is incremented or decremented by a certain value(200,80,150…) based on the number of sides open. Every situation that one-side-open or two-side-open could appear on the board is checked. By checking downwards or upwards to see if the player is equal to "X", "Y", or "N"(N is the spot that has not been placed by either player.

Minimax implementation

A depth is specified for the number of moves we want to see in the future. The minimax will switch between max and min values when *max_min_player* is true or false. If it is currently at a terminal state then there is nothing for the minimax to do. Because we have reached a win condition or the board is currently full. A *legal_moves* function is used to get all the possible moves for that player. The *legal_moves function* is used again to find the possible moves

for the opponent. From the list of possible moves we want the one with the minimum heuristic and then continue to find the max for the opponent.

Results

List of player 1 moves:
Total number of nodes generated:

List of player 2 moves:
Total number of nodes generated: