Assignment 2 Report

1. What heuristic did you use? Why?

   Consistent heuristic, all coded inside the `eval_frame()` function. This is done so that the heuristic function is simple and easy to program thus less bug prone.

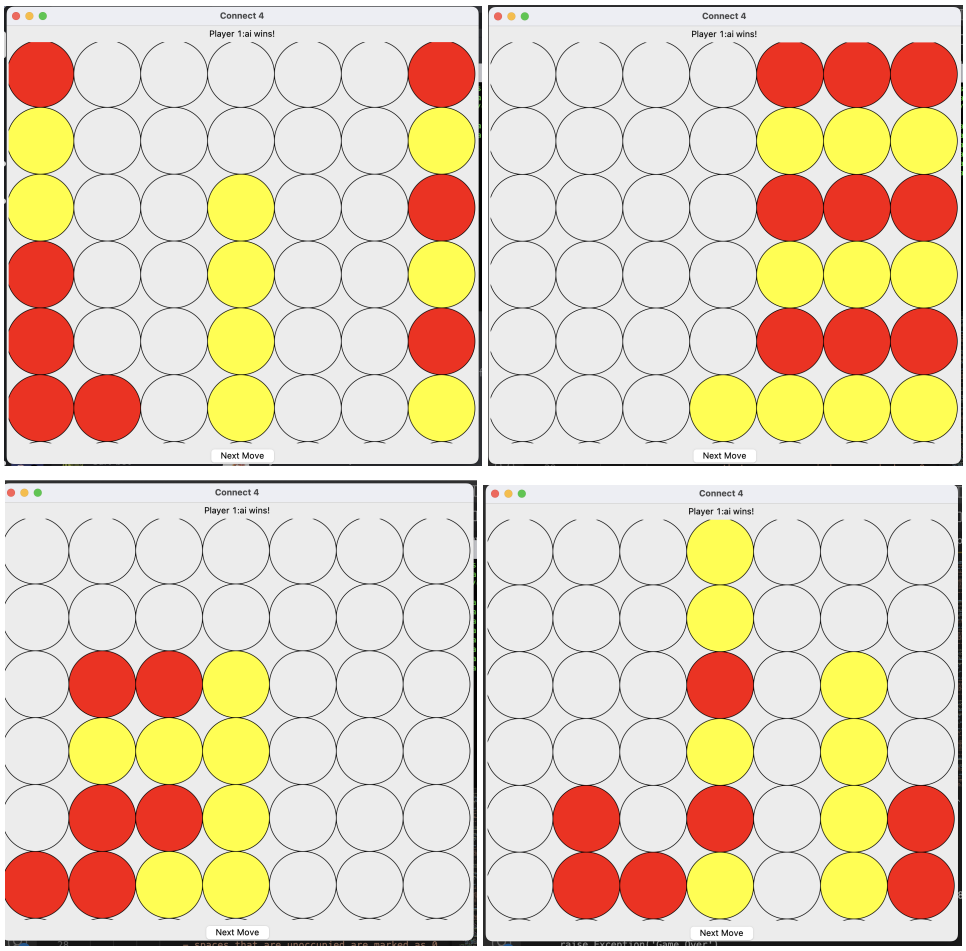
2. Describe how your algorithm performs given different time constraints. How much of the tree can you explore given 5 seconds per turn? 10 seconds? 3 seconds?

   Results vary machine by machine. On my desktop, 1 second is plenty for the entire tree to be explored with `MAX_DEPTH = 7`, while on my older MacBook, it struggled to explore the entire tree requiring 2 seconds at least. In both cases, as long as the tree is fully explored, the results appear to be independent of the time constraint given by running "`python3 ./ConnectFour.py <player1> <player2> -time <seconds>`", with `player1` having a better chance at winning.

3. Can you beat your algorithm?

   Yes, easily so.

4. If your algorithm plays itself, does the player that goes first do better or worse in general? Share some of the results. Running "`python3 ./ConnectFour.py ai ai`"yielded the following results (see next page). The first player appear to fair better than the other.


Connect 4
Player 1:ai wins!
Next Move
Connect 4
Player 1:ai wins!
Next Move
Connect 4
Player 1:ai wins!
Next Move
Connect 4
Player 1:ai wins!
Next Move