Connect4 AI - Assignment #2

-Submitted by Nandha Ramakrishnan
(Student ID: 1851265)

- 1) - I used a heuristic algorithm(evaluation function) where the current player is rewarded based on a sliding window(diagonal check included as well) checking for similar connected components. Only the windows with the same player number and empty slots are checked. The sliding window checks for connected components of the same player in all directions - vertical, horizontal, positive, and negative sloped diagonals. The player is rewarded with a score based on the number of connected components in the sliding window. It is awarded the highest score if the window is filled with a number similar to the player number. The score is being deducted when it finds something similar happening to the opponent to train the AI player based on the chances of the opponent winning as well. I chose this heuristic because it seems really intuitive and easy to implement. But obviously, this is not the best heuristics.

Player/ Window Size of 4 based evaluation	**Player 1**	**Player 2**
4 similar player components	current_score+5000	current_score-500
3 similar player components + 1 empty component	current_score+500	current_score-50
2 similar player components + 2 empty component	current_score+50	current_score-5

- 2) - Considering 3 seconds per turn, I'm able to explore up to a Depth of 4. Similarly, considering 5 seconds per turn, the algorithm explores up to a Depth of 5. With a time limit of 10 seconds per turn, the algorithm can reach up to a Depth of 6.

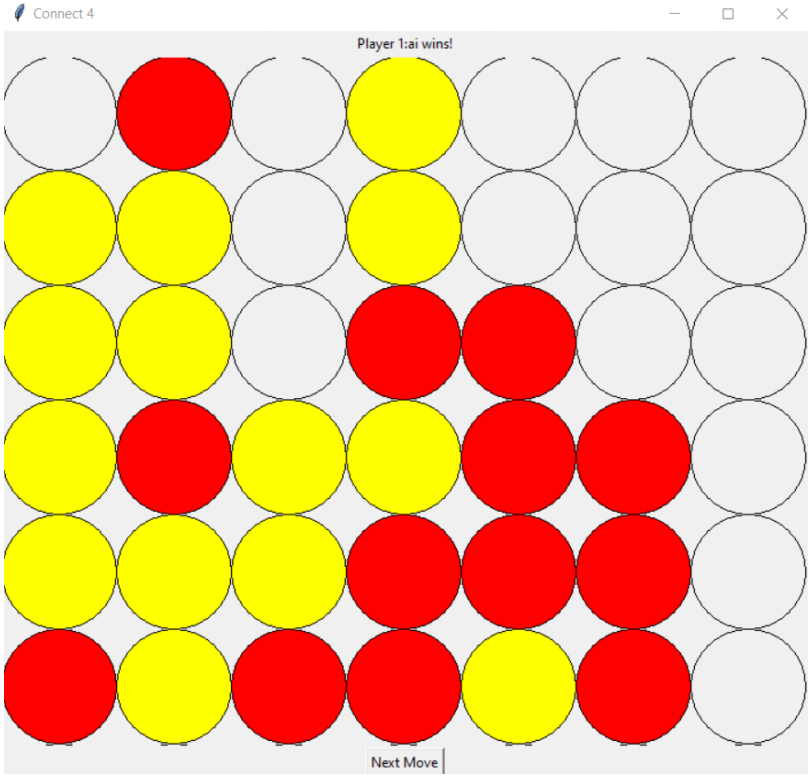
- 3) - Yes, I can beat the current implementation by using a better heuristics or evaluation function. In most cases, the AI should fill the middle columns first because they provide the highest chance of winning, I can introduce new scoring techniques for those moves. I recently came across MIT's Victor AI heuristics which can definitely be implemented for an effective AI player. Based on the current algorithm(AI vs me) - I'm able to win.

- 4) - When two AI players play, the probability that the first player wins is always high based on the heuristics that I used because it is considered to be the maximizer. The AI does not make moves in a complete manner as this is not the best heuristics. When the AI is played against a random player(when expectimax is implemented), it almost always wins. When the game is played between human and AI(since the heuristics is a simple one), human wins!

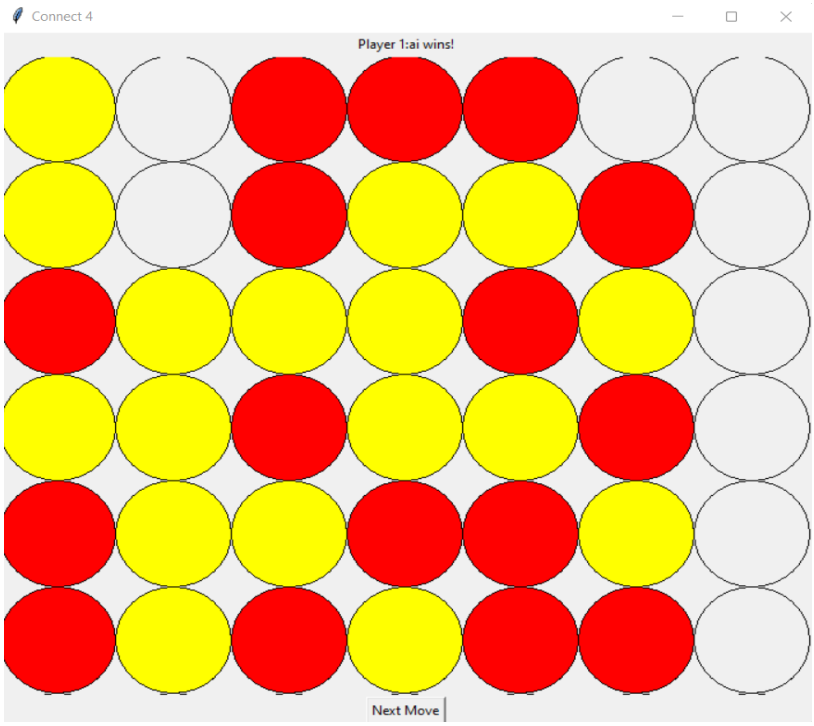
OBSERVATIONS:

AI vs AI:

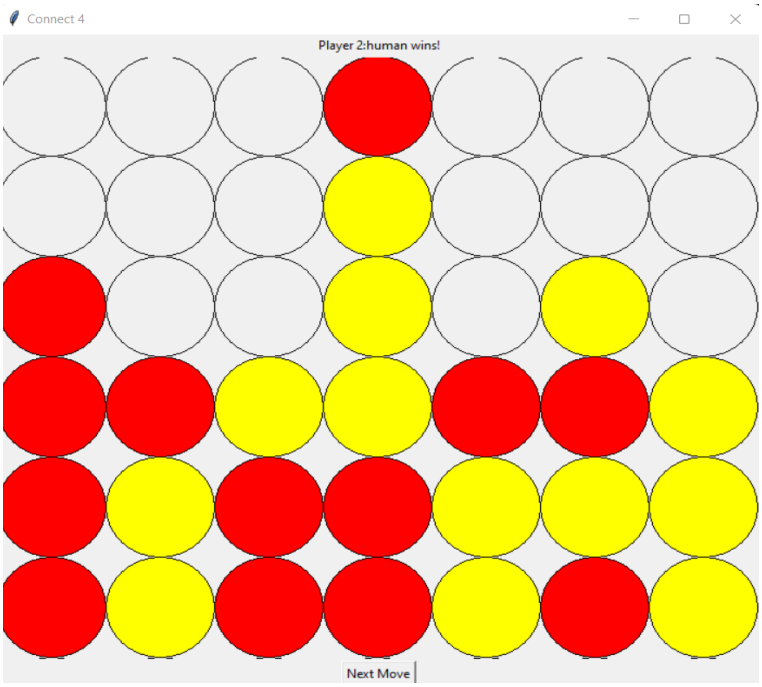
With DEPTH = 6:



With DEPTH = 5:

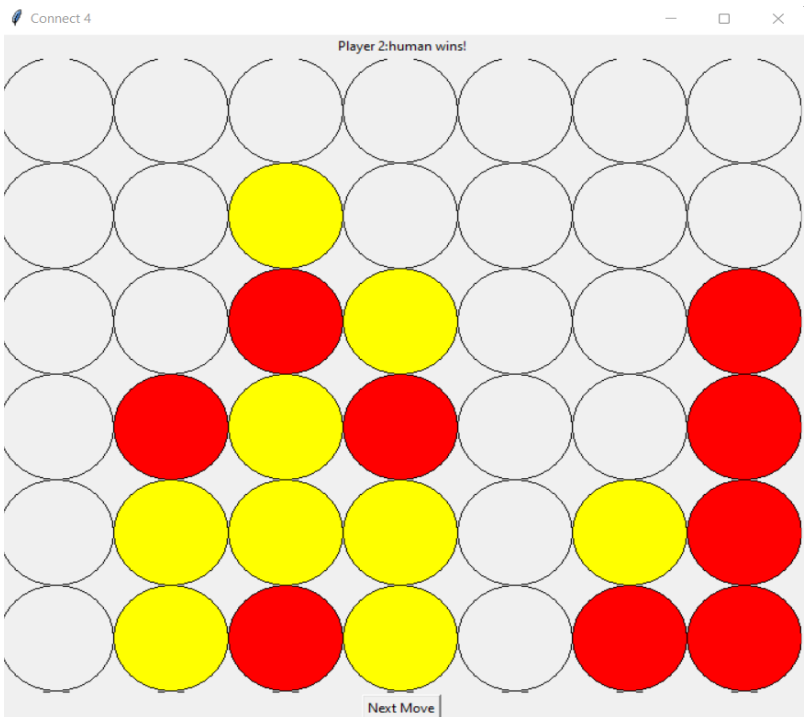


AI vs Human: I WON!!! (DEPTH = 6)



The AI performs better and is competitive at Depth = 6. For lesser depths, the performance is comparatively less as it less number of states are seen and evaluated. I can win easily when the depth is much less compared to the present depth.

At DEPTH = 5 - Easy win!!



Random vs AI:

Traversing from DEPTH = 3, the best(making it tough for AI to win) I could make AI and random to work based on expectimax,