Konan Le
CSE 140
Fall 2022

Assignment 2

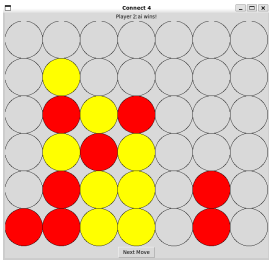
1. What heuristic did you use? Why?
    a. My heuristic function starts by initializing a board_utility value to 0. I use this as a representation of the entire board's goodness. **However, I wanted to encourage the AI to go for the choice with stronger opportunities for winning.** This could be said to be a board with many openings and different combos for multiple avenues of victory. So, my heuristic evaluates the "goodness" factor of every horizontal row, every vertical row, every diagonal, and adds it all together. In the end the algorithms will compare the various "goodness" levels against each other. The one with the highest "goodness" has more opportunities for the AI player to win because it implies that there are many pieces clustered together with openings for even more connections.
    b. **Implementation:** At first, I thought of just looking for a string match (e.g 111 or 222) amongst each row and returning a single value there. However, after a bit of tinkering and thinking about how I wanted to implement a heuristic, I came up with my above solution. So, I ended up coding a list of tuples (inspired by a kernel) corresponding to the player number and the enemy number, and assigned exponentially increasing values for friendly horizontal matches and exponentially decreasing values for enemy horizontal matches. The enemy values exist to make the AI afraid of moves that would allow the enemy to win. I made a similar list to check vertical scores, but this time it would just be the numbers building in one direction (right to left). This is because in a transpose matrix, the board becomes built from right to left. Afterward, I used the same method to check if game_completed in ConnectFour.py and simply go through the rows, columns and diagonals via a transpose of the numpy array, convert each row/column/diagonal into a string and see if the kernel can pinpoint anything inside, adding on to the board_utility variable everytime it finds a "good" connection.
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?

Konan Le
CSE 140
Fall 2022

a. I measured this by increasing the depth value until the set time limit was reached. I did not limit it to just 3, 5, 10 seconds, and instead increased the time until I did not receive a game over due to timeout.

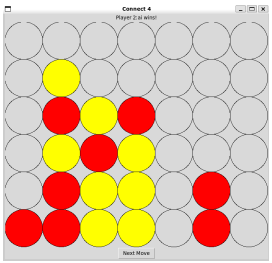
b. Alpha-Beta algorithm (tested with AI vs AI and Human vs AI)

   i. A 3 second time limit allows up to depth 5 for each move in AI vs AI. In Human vs AI, it also takes about 2-3 seconds.

1. 

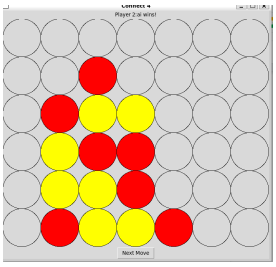
      a. Depth-5 game of AI vs AI

   ii. With a 5 second time limit, the AI will function up to depth 6 in AI vs AI. In Human vs AI, it will take between 5-6 seconds.

1. 

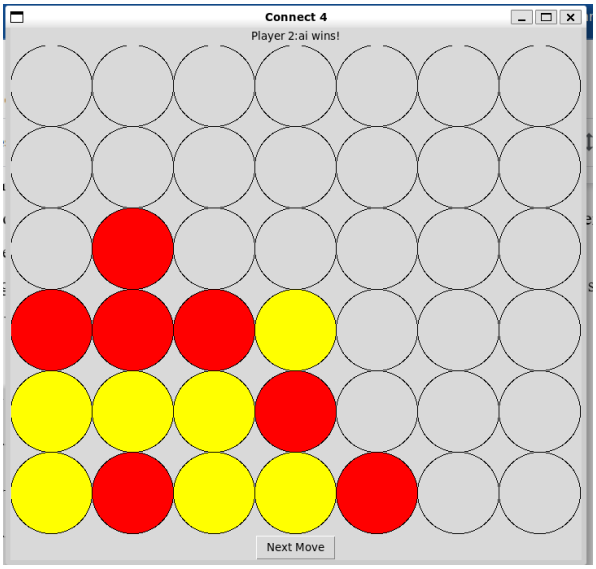
      a. Depth-6 game of AI vs AI, same result as depth 5.

   iii. Given depth 7, the algorithm fails a time limit of 10 seconds per turn. The AI will actually take 16-17 seconds to make each move at depth 7.

1. 

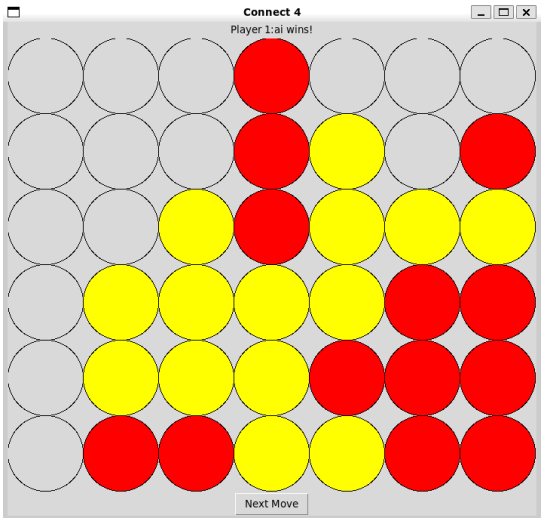
      a. Depth 7 with AI vs AI.

Konan Le
CSE 140
Fall 2022

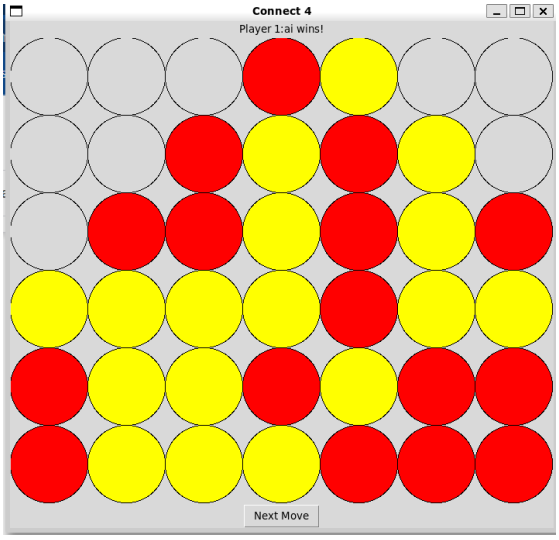
3. Can you beat your algorithm?
    a. After testing my skills against my AI, I found that I can only beat it occasionally, and whenever I did, it would put up a good fight. I would estimate my win rate against it to be about ~40%. If the AI takes a lead, then it goes straight for the victory, or attempts to smother me until I can only make a bad move. And of course, it analyzes the whole board and sees opportunities that I do not recognize myself.
4. If your algorithm plays itself, does the player that goes first do better or worse in general? Share some of the results.
    a. In general, I use deterministic, static values for my heuristic and calculations, so AI vs AI will only give one output. In that case, **Player 2 will win**. In the case of AI vs random, the AI would tend to win regardless of whether it was the first player, which makes sense because it is making moves with intuition.
    b. 

        i. AI vs AI, player 2 wins. Regardless of how many times you run it, this same result will follow.

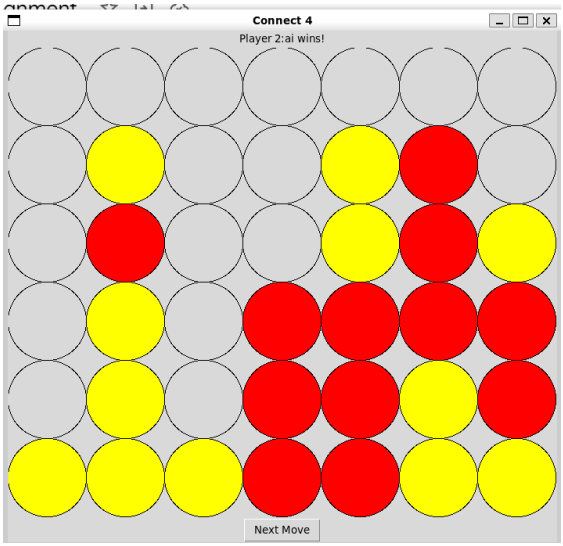
Konan Le
CSE 140
Fall 2022

c. 

i. AI vs random player, AI is player 1.

d. 

i. Another AI vs random player, AI is player 1.

e. 

i. Random vs AI, AI still wins.