DS2020– Artificial Intelligence

Lab 3

Due on 3/3/2024 11.59pm

Instructions: Upload to your moodle account one zip file containing the following. Please do not submit hardcopy of your solutions. Late submission is not allowed without prior approval of the instructor. You are expected to follow the honor code of the course while doing this homework. This lab should be completed **in pairs**.

1. A neatly formatted PDF document with your answers for each of the questions in the homework. You can use latex, MS word or any other software to create the PDF.

2. Viva will be conducted to check the contribution of each member of the team towards the assignment.

3. Include a separate folder named as 'code' containing the scripts for the homework along with the necessary data files.

4. Include a README file explaining how to execute the scripts.

5. Name the ZIP file using the following convention rollnumberrollnumberhwnumber.zip

---

ConnectFour

This lab assignment is adopted from the modelAI assignment[1].

The goal of the assignment is to implement an agent to play the game of Connect 4. Your agent will use alpha-beta pruning and expectimax search to select the next move given the current board state.

There are two main files:

- ConnectFour.py
- Player.py

ConnectFour.py contains all functions of the game. Player.py contains all types of players that can participate in the game:

- AIPlayer (Your Implementation)
- RandomPlayer (Chooses from valid columns with equal probability)
- HumanPlayer (You)

You need to implement all of the following functions:

- def get_alpha_beta_move(self, board)
- def get_expectimax_move(self, board)
- def evaluation_value(self, board)

These functions serve as a high-level abstraction for interacting with a player. You will likely need to implement other functions in the class to make your code modular and readable. **Note that it will likely be too expensive to explore the entire game tree at the beginning of the game so a depth-limited search is a good place to start.**

To play the game you need to run “python ConnectFour.py arg1 arg2” where arg1 and arg2 are one of AI, random, or human.

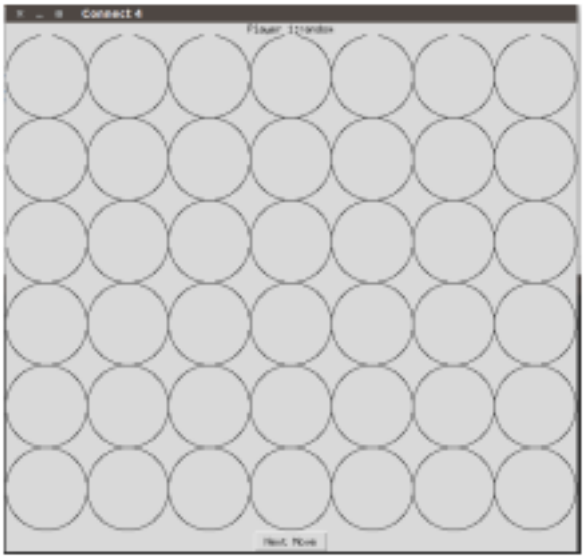
For example if you wanted to play against a random player you would run “python ConnectFour.py human random”

If you wanted your AI to play itself, you would run “python ConnectFour.py ai ai”

ConnectFour.py takes one optional argument --time that is an integer. It is the value used to limit the amount of time in seconds to wait for the AI player to make a move. The default value is 5 seconds.

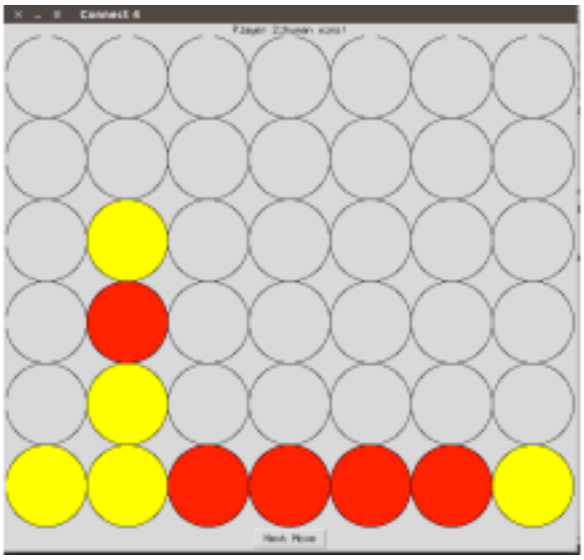
Note: A human player has to enter their move into the terminal.

Here is the description of the GUI included in the game:



The top text displays the player making the next move. The well-drawn circles in the middle are the places on the game board. The next move button does exactly what you expect, it moves the game forward by one move. When the game is over the top text will change to the name of the winning player and 'wins!'

Shown below is an example of a Random player:



In this lab you will submit the following:

- Player.py: this should include your alpha-beta search algorithm and expectimax search implementations in the AIPlayer class
- A pdf that addresses the following questions:
    - Describe the heuristic that was used, along with the rationale behind using it.
    - Given different time constraints, how will your algorithm perform? How much of the tree can you explore given 3, 5, or 10 seconds per turn?
    - If the AI player plays against itself, does the player who goes first do better or worse in general? Illustrate it through some results.

Submit the all the code and the pdf as a single zip file.

Note: The TA was successful at executing the code. In case you face any issues and have found a solution, please share it in the Moodle forum for other's benefit.

[1] http://modelai.gettysburg.edu/2020/connect4/