CS 333-Project 1
FALL2022

Ozyegin University

Description

In this project you will implement a game called “connect 4” by Haspro.

Game Rules and description

- In this game players alternately place pieces on a vertical board 7 columns across and 6 rows high.
- Each player uses pieces of a particular color, and the goal is to be the first to obtain 4 pieces in a horizontal, vertical, or diagonal line.
- Because the board is vertical, pieces inserted in a column always drop to the lowest unoccupied row of that column. As soon as a column contains 6 pieces, it is full and no other piece can be placed in the column.
- Both players begin with 21 identical pieces, and the first player to achieve a line of four connected pieces wins the game. If all 42 pieces are played and no player has places 4 pieces in a row, the game is drawn.

Implementation

- You are to implement a DFS or BFS to choose the move. You are free to choose the algorithm (DFS or BFS). Choose only one.
- It is a step by step game between a user and your program.
- You must set up your program so that search depth/level can be a user-provided parameter and you must run your program with different levels of lookahead. your program runs it must ask the user to enter a number that sets the search depth/level, for example:

  “””

  Enter the search depth:

  2

  “””

- The game should run step by step, In the first step the user chooses the move by entering the column number that they want to put the piece in (for example when all the columns

have empty spaces the move can be a number between 1 to 7 that represents the column number). When it is the user turn the program should show the state of the game to user ask:

“””

pick a move (a number between 1-7)
2

“””

Now it’s the program turn to pick and execute a move. And it should continue until the game finishes.

- **You must print on the screen the state of the game at each step**. (every time after the user move and every time after the program move. Show empty spots with “#” and show the user and program moves by “X” and “O” respectively.

  For example the state of the game is shown as:

  “””

  # # # # # # #

  # # # # # # #

  # # # # # # #

  # # # # # # #

  # O # X # # #

  # X X O # # #

  “””

- You should implement an evaluation function that estimates how good a particular game state is to tell your program which move to pick.
- You will also need to keep track of pieces in a row; when you make a move that results in 4 pieces in a row, your program should declare that who is the winner
- Finally, if you make an illegal move, then you lose the game

Technical Detail

- It is your responsibility to make sure that your code runs without any errors. In the case of any errors will get 0 points.
- **Use of any additional libraries is not permitted in this project.** You must implement everything yourself. Use of additional libraries will result in your code not running and you getting 0 points.

Report

The next step is to write a report and explain your work.

In the report you should write whatever we need to know about your work. In particular, which search method did you choose (DFS or BFS), your strategy, evaluation function, …

Submit your report in pdf format.

Submission

You must turn in two files in a **zipped folder.**

1. First file should be your code named:
   NAME_SURNAME_StudentNO.java
   Or
   NAME_SURNAME_studentNO.cpp
2. Second file should be your report named:
   REPORT_ NAME_SURNAME_StudentNO.pdf

Grading

It is very important to state that each submission must have both the code and the report, submissions that lack any of these two files will not be evaluated and will receive 0 in total.

- Your code (60 pts)
- Your report (40 pts)
- In this assignment we are not focusing on the optimal strategy for the game. However your evaluation function will be evaluated based on creativity and functionality. If you execute a suitable strategy you can get **up to 20pts bonus**. If you want to get this bonus make sure to explain your evaluation function and strategy well in your report

Important Notes

- This is an individual assignment, hence sharing your code will be considered cheating
- you can most probably find code online for this game, but it is very likely that code will be more complicated than what you are required to implement for the assignment. While it is OK to look for strategies online, copying code that you find online will be considered cheating.
- Your codes will be compared with codes available online and also with other submissions.

BEST OF LUCK 😊

In case of any questions email me via negin.amirshirzad@ozu.edu.tr