COM S 1270 - 'Four In Sequence!'

Please see the 'Course Schedule' on the Syllabus for this assignment's due date.

Assignment Objective

The objective of the assignment is to give you practice working with/ fixing/ adapting code made by AI (in this case, ChatGPT), and getting an incomplete project into a functioning state.

The activity for this assignment will give you practice creating a simple AI agent for a version of a pre-existing boardgame, 'Connect Four,' which will be in a pre-coded 'start file' file called `fourInSequence.py` (**NOTE**: Do *not* change the name of the start file). This will likely be a difficult assignment as it will require logical thinking considering multiple exact cases. Meaning, the game will not work correctly unless all of the cases are correctly accounted for.

If you have never played 'Connect Four' before, please take a look at the following resources:

- https://en.wikipedia.org/wiki/Connect_Four
- https://papergames.io/en/connect4

Instructions

The 'algorithm' (rule set) of the game is thus:

- The user selects a game between one or two players.
    - A one player game has the user play against the computer.
    - A two player game has two users play against one another.
- Each player takes turns 'dropping' their 'pieces' into the board.
    - The goal is for one player to connect four pieces in an unbroken sequence.
        - A sequence can be horizontal, vertical, or diagonal.
- The game ends when one player has... FOUR IN SEQUENCE!

You should study the file(s) provided, and notice the various 'TODO' comments contained therein. These comments indicate places in the file that you should modify the code to be your own. These 'TODO' comments are the places in the script where code/ student work will be evaluated for the final grade on the assignment. **DO NOT PLACE CODE OUTSIDE OF THESE TODO STATMENT AREAS.**

NOTE: IF YOU DO NOT USE THE PROVIDED 'START FILE,' YOU WILL BE GIVEN A ZERO (0) ON THE ASSIGNMENT. DO NOT ATTEMPT TO PROGRAM YOUR OWN VERSION OF THE GAME!

Your work should be your own, completely original effort.

To help with debugging, you may temporarily place the code found in `debugging.txt` at the top of the `main()` function in your code. You can tell where to put it in the 'start file' by way of the `def main():` function in `debugging.txt`. This code you temporarily add will allow you to immediately test any board state with any function that takes the game board as input. This will prevent you from having to play through multiple rounds of the game just to test whatever function you wish to test. You should just configure the board however you like, and then pass in that data structure to the function to be tested. Just be sure to remove this temporary debugging code before submitting your final version.

As a note - 'Player 1' should always be the 'human' player in '1 player mode.' Obviously, in '2 player mode,' both players are 'human' players.

Other Requirements:

Additionally, you do *not* need to compensate for the crashes that occur when the user enters a value of the incorrect type. (E.g., The program asks for an integer, but the user enters a letter instead.)

It is required for you to write your name, the date you started working on your script, and a short explanation of what your code does at the top of your file. For example:

```
# Matthew Holman             7-19-2024
# Assignment #4
#
# This is a version of the classic boardgame 'Connect Four' - here called 'Four in Sequence!'
```

Do *not* attempt to 'double dip' by placing your 'header' printout information at the top of your file - the header and the initial comment block are different things.

Your work should be your own, completely original effort. Meaning - you should not just copy my code explanation above, but rather you should try to come up with one of your own.

The actual Python script itself can be programmed in any way (excluding cheating ala ChatGPT) so long as the output resembles that below. This includes the use of the initial 'header' when starting the program (i.e. the program title, who it is by, the student's class/ section, etc.). Please note that the 'header' will be strictly enforced with this assignment. Be sure to include the title, your name, and your class/ section. As long as you use the pre-written 'start file,' everything should appear correctly.

Example Output

```
Four In Sequence!

By: Matthew Holman
[COM S 127 1]

Choice? [p]lay, [i]nstructions, [q]uit: i

#### The 'algorithm' of the game is thus:
- The user selects a game between one or two players.
    - A one player game has the user play against the computer.
    - A two player game has two users play against one another.
- Each player takes turns 'dropping' their 'pieces' into the board.
    - The goal is for one player to connect four pieces in an unbroken sequence.
        - A sequence can be horizontal, vertical, or diagonal.
- The game ends when one player has... FOUR IN SEQUENCE!

Choice? [p]lay, [i]nstructions, [q]uit: p
Number of Players? [1] / [2]: 1
######################################################################

~~ Starting New Round ~~

.......
.......
.......
.......
.......
.......
0123456

Player 1, please select a column between (0-6): 0
.......
.......
.......
.......
.......
X......
```

```
0123456

~~ End Of Player 1 (X) Turn ~~

Player 2, please select a column between (0-6): 4
.......
.......
.......
.......
.......
X...O..
0123456

~~ End Of Player 2 (O) Turn ~~

Player 1, please select a column between (0-6): 3
.......
.......
.......
.......
.......
X..XO..
0123456

~~ End Of Player 1 (X) Turn ~~

Player 2, please select a column between (0-6): 4
.......
.......
.......
.......
....O..
X..XO..
0123456

~~ End Of Player 2 (O) Turn ~~

Player 1, please select a column between (0-6): 2
.......
.......
.......
.......
....O..
X.XXO..
0123456

~~ End Of Player 1 (X) Turn ~~

Player 2, please select a column between (0-6): 1
.......
.......
.......
.......
....O..
XOXXO..
0123456

~~ End Of Player 2 (O) Turn ~~

Player 1, please select a column between (0-6): 4
.......
.......
.......
....X..
....O..
XOXXO..
0123456
```

```
~~ End Of Player 1 (X) Turn ~~

Player 2, please select a column between (0-6): 0
.......
.......
.......
....X..
O...O..
XOXXO..
0123456

~~ End Of Player 2 (O) Turn ~~
```

etc.

Files Provided

`fourInSequence.py`

`debugging.txt`

Special Notes

NOTE: This assignment will be even more difficult than the previous assignments. However:

- Completing this assignment may require you to start your work 'before the last minute.' Please plan accordingly.
- Your script **CANNOT** crash under any circumstances except for those circumstances noted below. Any portions of your code where the script crashes will receive a zero (0) for that aspect of the assignment.
    - Understand, when the word 'crash' is used in the instruction above, what this means is 'crashing under expected use cases given the level of knowledge attained in the class.'
        - You do not currently have the ability to enforce user input types (although you will later in the semester). As such, if you ask for an integer as input (e.g., 1), and the user provides a letter (e.g., 'a'), and it crashes, then this is *not* a problem *yet*.

NOTE: You are turning in your CODE - NOT the OUTPUT of your code.

- Your code will be run by the TAs, and the output of those runs, along with the code itself, is what will be evaluated.

NOTE: Assignments turned in in any other format other the specified file types will not be accepted.

- Screenshots of code **will not** be accepted.
- `.sln` files are **not** code files - they contain **no** Python code and **will not** be accepted.
- `.zip`, `.rar`, `.tar.gz`, and other compressed files **will not** be accepted unless otherwise specified.
- If your submission is not in a `.py` file, when so specified, or if the submission is not accompanied by **all** the files needed to run the submission, the submission will not be graded.
    - **THIS WILL LEAD TO YOU RECEIVING A ZERO (0) ON THE ASSIGNMENT.**
    - **You will *NOT* be allowed to re-submit your work after the final deadline in this case.**

NOTE: You have the ability (and responsibility) to 'double check' that your work/ submission is correct when you turn it in.

- If you accidentally turn in the wrong submission, you will **not** be allowed to resubmit it after the final deadline.
- Please note, you can submit your work as many times as you like *before* the final deadline.