Connect-4

CSE 1342 - Programming Concepts - Summer-2 2018
Due: July 23 at 11pm.

Overview:

Games are fun... super fun! In this assignment, you're going to implement the game of Connect-4[1]. For our version of Connect-4, we'll be using a 9 rows x 7 columns grid. Your version of the game should be text-based and run in a terminal.

Implementation Details:

The Connect 4 game board should be implemented as a 2-D array of characters (or other data types). Initially, the board is filled with spaces in all the possible token areas, but will be filled by the players with Os and *s during play. You are free to choose your own tokens for play. Notice in the implementation at the end, Xs and Os were used. During play, one of the players will drop a token down one column by choosing a particular column. The game implementation must calculate what row to place the token. The board will then need to be checked for a win. If there is a winner, state the winner and then end the game. Otherwise continue the game with the next player.

Your implementation needs to be modular. Your design decisions will be reflected in your grade. Remember that functions should perform one task. For this game, some examples might be a function to display the game board, a function to check for horizontal winning, etc.

Error Checking:

Your program should not allow the user to enter an invalid column number. If the user enters an invalid column number, your program should ask the user to re-enter a column number. If the user chooses a column that is already full, your program needs to alert the user and ask him/her to choose a different column.

Due Date:

This project will be due on July 23 at 11:00pm. You should submit your zipped file with your source code in it. Your code should be:

- properly formatted
- properly commented
- properly decomposed

Submit <lastname>P03.zip to Slack.

[1]If you're unfamiliar with Connect Four, have a look at http://en.wikipedia.org/wiki/Connect_Four.

How your project will be graded:

Criteria	**Points Possible**
Correctness	50
Clarity of code	20
Comments	10
User Interface	10
Formatting	10