CMPSC-221: OOP for Web

Week 9 Homework: Connect 4 - Fall 2019 (100 points)
Due: Monday, October 28th at 11:59pm

Background

According to Wikipedia Connect 4 is, "a two-player connection game in which the players first choose a color and then take turns dropping one colored disc from the top into a seven-column, six-row vertically suspended grid. The pieces fall straight down, occupying the lowest available space within the column. The objective of the game is to be the first to form a horizontal, vertical, or diagonal line of four of one's own discs."

Objectives

By completing this assignment, students should be able to:

- Write graphical programs using buttons and labels from JavaFX
- Explain how to build programs using JavaFX along with inheritance and/or interfaces

Assignment

This week, you will work to create a simple two player version of Connect 4. Your game should be able to determine a winner or if the game results in a tie. You must also have a button that will reset the game so it can be played again.

Statement.txt (5 points)

As with all assignments, please make sure to submit a completed `statement.txt`.

Design (25 points)

1. What are five classes from JavaFX you will need to complete this assignment? What do you expect to use each class for?

2. We have to be able to track whether a spot is marked as Red, Black, or unmarked so we can figure out when someone wins or if there is a tie. How might we do this without having to do lots of string comparisons?

3. Is there a way we could perhaps use inheritance or interfaces when designing our board spaces to make determining a winner easier? If so, explain it. If not, explain why not.

4. Write an English language algorithm for resetting the board. Be sure to explain what you have to do (and in what order) in order to reset the board.

Implementation (60 points)

Write a program that will let two users sitting right next to each other play a game of Connect 4 that meets the following criterion:

- The game should be able to be reset so it can be played multiple times.
- The game ends when one person has won or there is a tie.
- The winner should be printed to the command line.
- The user should be able to select a column and have their token (red or black) occupy the lowest available space on the column. (Note: This doesn't have to be be fancy, a button at the top of the row is fine.)

Style (10 points)

As with all assignments, ten points of your grade are connected to how well you follow the style guidelines detailed on Canvas.

Notes

- A sample layout (minus the reset button) for tic-tac-toe can be found on page 806 of your book. You do not have to use their X and O images to get a good grade on this assignment. Your reset button can be wherever you want it to be, but it must be visible to the user as to where this button is.

- Red should go first.

- Unlike past assignments, I am fine with the main class being the board, but the board class must be called `Board.java`.

- Once a space is taken, you have to make sure that it cannot be claimed by the other player. For example, if Red picks a spot Black cannot come in and take it.

Contest

This assignment will also have a contest as part of it. The person who creates the best GUI as determined by the CMPSC-221 staff will get one of two prizes of their choice.

Submission

Please submit all files prior to the deadline listed above. Your main class should be called `Board.java`.

Implementation Grading

20 points: The board is set up correctly.

5 points: The board is laid out in a clean and easy to understand format.

5 points: Game alternates between placing Red and Black pieces depending on whose turn it is.

10 points: The game determines a winner or a tie and prints it to the console.

10 points: The reset button successfully resets the board.

10 points: The algorithm to figure out the winner is well thought out and easy to understand.