IDS 401

**Fall 2015**

# Assignment 1

**Due Friday 11/27/2015**

**180 points**

# Instructions

### Deadline

In order to receive full credit, this assignment must be submitted by Friday, November 27th at 11:59pm. Submitting your assignment early is recommended, in case problems arise with the submission process. Late submissions will be accepted (but penalized 15pts) up to one week after the submission deadline. After that, assignments will not be accepted.

### Assignment

Extend your text-based Connect Four application to create a GUI-based application allowing customization of the game board and victory conditions. Submit your code via Blackboard, as a single .java file. The file name should be your last name, first name, and GuiApp run together:

EX: LundquistDougGuiApp.java.

The required functionality is:

1. Provide GUI objects for players to:
   1. Enter Player 1’s name
   2. Enter Player 2’s name
   3. Have the option to randomize play order prior to game start
   4. Make moves (still dropping pieces into columns as in Connect Four)
   5. Specify the dimensions of the board: rectangular with:
      1. Between 4 and 12 rows
      2. Between 4 and 15 columns
      3. Specify the number of placements in a row required for victory: At least 3 and less than the number of rows or columns
2. Players make moves by using mouse and clicking on squares (assume “hotseat”, where both players take turns using the mouse)
3. Display the game board in a JFrame object, both before the game and updated after each move. Output should include:
   1. Which squares are unclaimed
   2. Player 1 moves shown as solid red circles
   3. Player 2 moves shown as solid yellow circles
   4. Borders around each square and the entire board for easier board-reading
4. After each move, the game should check for victory or a tie
5. Game should conclude with a dialog box including:
   1. Announcement of result and congratulations to winner (if any)
   2. Set of GUI objects for selecting repeat play by the two players, with the options of randomized, reversed, or repeated play order
   3. GUI object for selecting new game (with changes to board, players, or victory requirements)
6. Attempting to close the window should:
   1. Before or after a game: Close the window and exit the game.
   2. During a game: Provide a dialog box asking if the player really wants to quit and options for closing the window (and aborting the game) or not.

### Program Code Constraints

There should be only one class, GuiApp. Each of the required functionalities should be implemented as separate methods (as far as possible) and operate by passing parameters and/or returning values (*e.g.*, inputting player 1’s name should call a method that returns whatever String the user entered). Comments should be provided for the class and for each variable and method, explaining their usage. You can assume players only enter valid inputs for game parameters but the program should explicitly prevent impossible moves (*i.e.*, dropping a piece into a full column). Additionally, any relevant deficiencies in gameplay from the previous assignment should be corrected.