Assignment 11
CSc 210 Fall 2017
Programs Due November 20th, 8:00 pm MST

Introduction

We have begun looking at GUIs and how to create them using JavaFX. For this assignment, you will build upon your Shape Drawing program that prints shapes to the screen. In this final rendition of our shape program, you will ditch the REPL and create a GUI that allows users to draw shapes onto the screen using visual interactions.

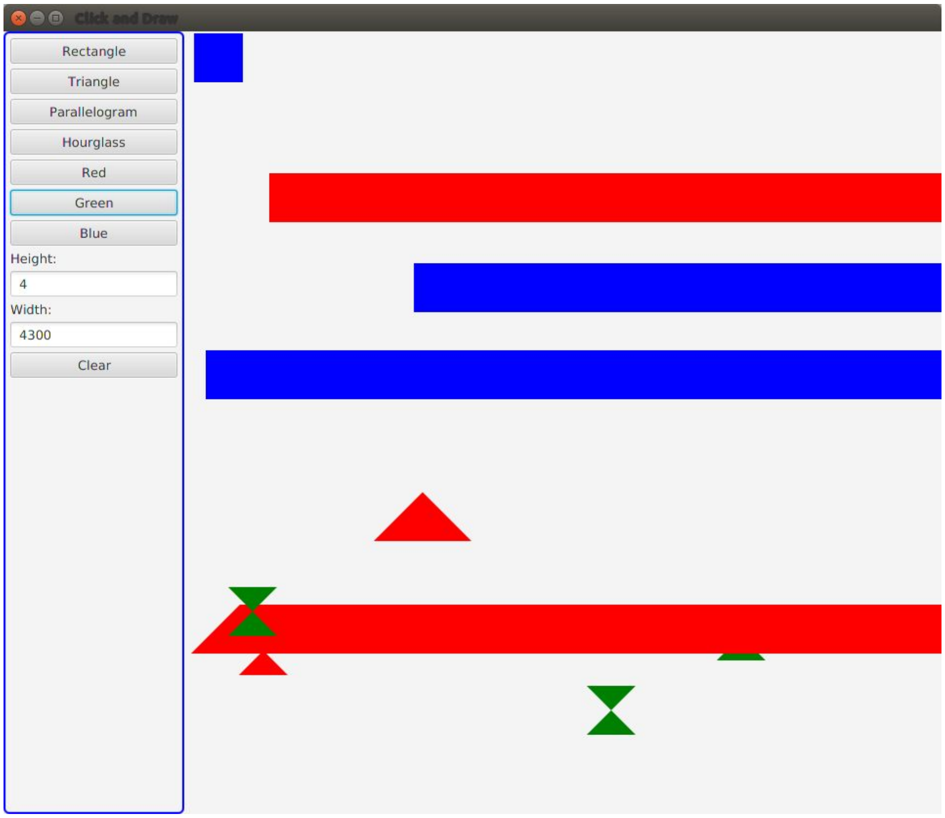
You will turn in your code via the github classroom. Here is the link to create your repository: https://classroom.github.com/a/xKElOeRX

For each Java program assigned, you will turn in a .java file for that program. We should be able to run "javac *file*.java", followed by "java *file*". Your program should compile without warnings and run on the test machine (Lectura)

Your Java programs should adhere to the style guidelines. Here is the link: https://www2.cs.arizona.edu/classes/cs210/fall17/StyleGuidelines.pdf

Drawing Shapes (75 points)

As stated before, this program will draw various shapes on the screen via interactions with a GUI. In class, we have been discussing how to use event listeners to detect when different events, such as button presses, text box updates, and mouse clicks, happen in a GUI. You will use these tools to implement your solution to this assignment. Below is an image of what your GUI might look like (Note: This is not a strict template. Feel free to be creative! However, **all of the functionality offered here must be present in your solution**).



Height:
4

Width:
4300

Your solution must have the 8 buttons and two text boxes pictured here on the left side of your window. You may design them in any way you like. As mentioned in the section talking about extra credit below, you can add to this design in any way you like. You can add buttons, labels, text boxes, drop down menus, check boxes, or anything else you think would be fun.

You should minimally implement the following functionality:

- When the “**main canvas**” is clicked (the area in the GUI to the right of the bordered box in the image above), your program will display a shape on the screen; the top left corner of the shape’s bounding box will be the location of the click. The main canvas will be 800x800 pixels in size.

- Upon clicking one of the “**shape buttons**,” the current shape being drawn upon the main canvas will be updated to the shape on the button.

- When clicking the “**color buttons**,” the color of the shape being drawn will update to the color on the button. Implement buttons for red, green, and blue.

- The **text boxes for “width” and “height”** will take integers as input to define the size of the bounding box for the shape being drawn. You may use the pressing of the enter key to save the new width or height. If the entered text is not an integer, your program should not quit, but handle this anomaly gracefully.

- When the “**clear button**” is pressed, all shapes will be removed from the GUI and the main canvas will be blank.

The code for this program will share many components with the last assignment. We recommend the following steps for implementation:

1. Start with the code from assignment 10. Remove the ShapeREPL code.

2. Create your GUI layout without any listeners. Figure out how to layout your canvas, buttons, and text boxes. Then create this GUI in JavaFX and get it to appear on the screen.

3. Once you have the GUI built, add event listeners for the various components.

Finally, there will be an extra credit opportunity on this assignment. For an additional 15% of the grade, you may implement a “wow factor” in your program. This will be subjectively graded by your SL. If you decide to add a wow factor to this program, make sure to let your SL know you did and what it is. Some ideas for wow factors are:

1. Implement **several** additional shapes (i.e. add in a circle, octagon, and star)
2. Add in an opacity slider for the shapes drawn.
3. Create an undo button that will remove one shape at a time, just like pressing Ctrl+Z.
4. Add an erase tool that allows you to click on a shape and erase it.

The wow factor should show that you put significant effort into implementing your feature and went a step beyond what was expected.

When you are done, you should turn in at least one file called `Drawing.java` as well as all of the files for your shape classes. (`Triangle.java`, `Rectangle.java`, etc.). `Drawing.java` should contain your main method.

Miscellaneous

In order to receive a grade, your repo must contain only source .java files in the root directory of your repo.

This assignment will be submitted through github classroom. Make sure all the code you wish to submit is in your repository when the due date arrives.

Note: Your output must match what is defined here in the spec. We will give you a small selection of test cases so you can make sure you have the right format. Do not print extraneous output, you may lose a lot of points (this includes prompts when reading in input!). **These test cases can be found in your repo.**

Remember, do not cheat! Refer to the syllabus and first lecture for more information.