CS 251 Intermediate Programming
Lab 8: GUI Layout Practice

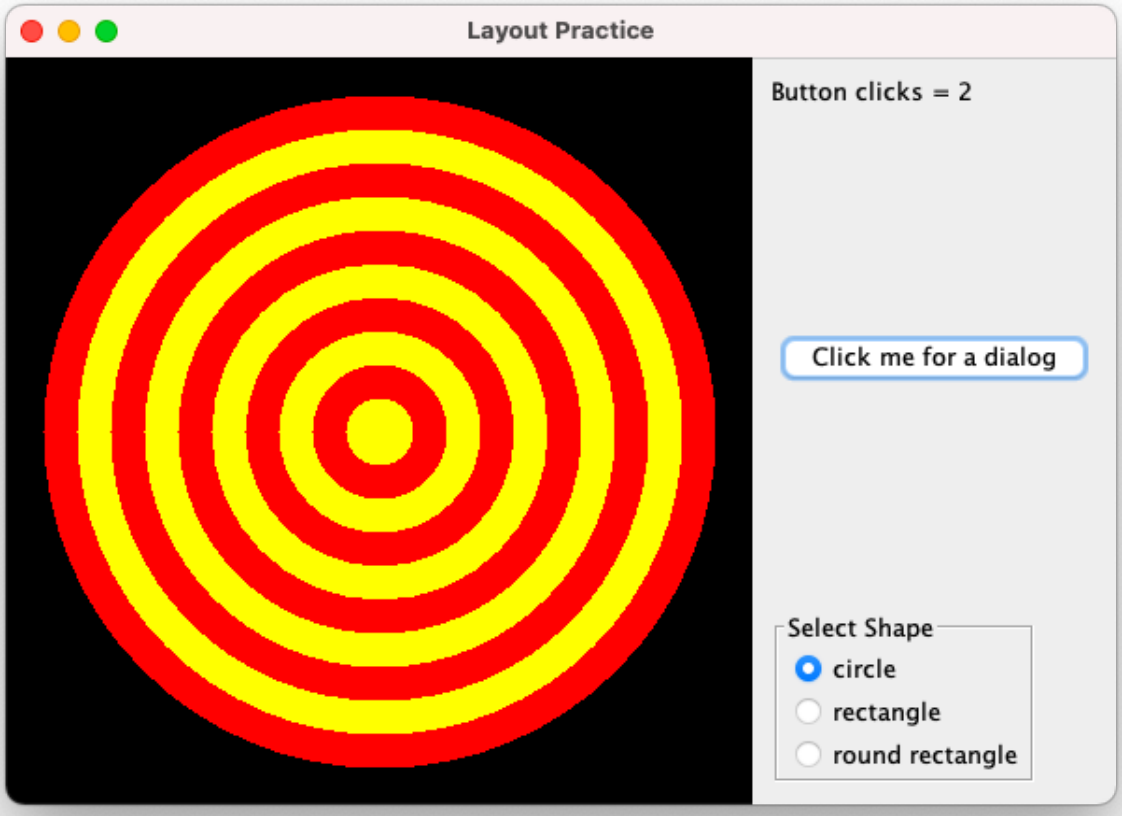
Brooke Chenoweth

Spring 2024

Please note: This is an individual assignment. It is designed for you to get used to get some experience writing a GUI class. This should further improve your understanding of how components and containers fit together and make a very powerful set of tools when building graphical interfaces. As you go through this exercise, you should pay attention to the inheritance structure that the classes in the `java.awt.*` packages and `javax.swing.*` packages belong to. It is a great example of object orientation in the working.

Problem Specification

For this assignment, you will create a simple GUI that looks something like the following.



Required Classes

The class that contains the `main` method should be named `LayoutPractice.java` so we will not have any question about which class to run to test your program. You are not limited to turning in just this file. If you prefer organize your GUI implementation with additional top-level classes, that is fine. Just make sure you turn in all the relevant files, and include a `LayoutPractice.java` to start the program.

GUI Elements

Your GUI should be sufficiently complicated, i.e., there should be at least a composite layout, and two customized `JPanels`.

The minimum required elements of the GUI are:

- A large panel on which a custom image is painted.[1] This doesn't have to be very

[1] Hint: override paintComponent

complicated, but should involve more than one shape[2] and more than one color.[3]

- A button that triggers a dialog[4] to be displayed. The dialog should display a message informing the user of how many times the button was clicked.
- A label that displays how many times the button was clicked.
- Some other control that will change the image displayed. In my example, I have a radio button group to select which shape is drawn. You could do that, or have a combo box to select from several options, a slider bar to select the size of something, another button that opens a color chooser dialog to change the color of your image, etc.

  You have a lot of flexibility here. I just want some additional control component so that you will have to hook up another listener and appropriately update your image. It doesn't have to be particularly fancy, but it needs to make some sort of noticeable change to the image so we can tell that it's working.

These elements should each be part of your GUI, but the layout is up to you. If you, for example, want to put the label and the button in a horizontal line across the top instead of at the right as in my example, you can, of course, do that. You can be as fancy as you want. Feel free to experiment with other components as long as the required functionality is still obviously there.

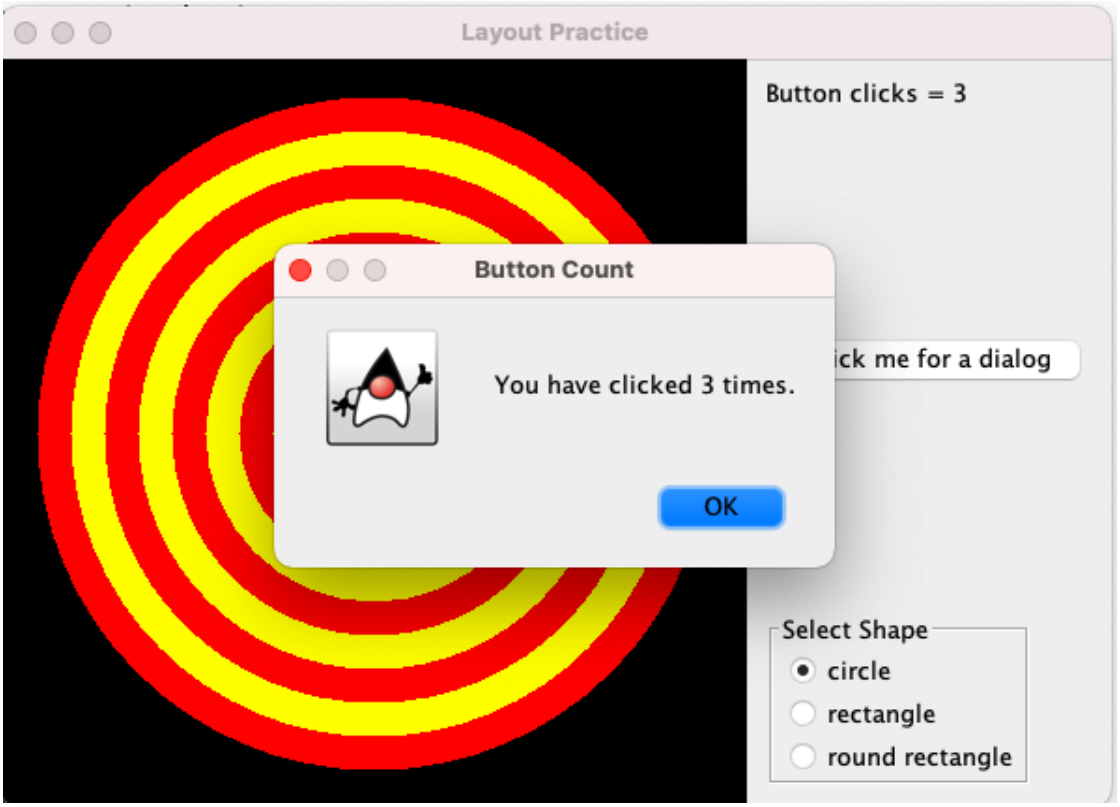
There need not be many actions implemented, but the one that is required is that the label is updated and the dialog is displayed when you click on the button.

Have fun with the assignment! Feel free to experiment with additional GUI elements.

[2]This can be more than one of the same shape, such as the concentric circles in my example, or you might get fancy and actually put shapes and lines together into a picture.

[3]If you chose to load and display an image from a file, that will count as one shape/color.

[4]Hint: use one of the JOptionPane methods to make the dialog.



Turning in your assignment

Once you are done with your assignment, use Canvas to turn in the java file(s) that you have created.[5] Note that each source code file should still contain a header comment with your name, class, and other related information. Failure to properly comment your files may lead to point deductions. *Please make sure to use proper Javadoc comments in all your files.*

[5] Also include any necessary sound and image files if you chose to get really fancy and load additional resources.