

Window System (Part 3): RAT: Really Awesome Toolkit

Due: Monday, May 19th, 2014, **8:30 am**, in groups of 2 students via L2P.

Description

You will finish off your window system in this assignment by building the **Really Awesome Toolkit** (RAT™). Your toolkit should support (at minimum) a label and a button widget.

Tasks

1. The Toolkit

There are many ways of structuring the class hierarchy of your widget toolkit. One possibility is as follows:

- `RATwidget` (*abstract class*)
 - `RATlabel`
 - * `RATbutton`

A `RATlabel` should be able to show some text on a colored background. A `RATbutton` should be able to respond to mouse events, and allow other objects to receive notifications when it is pressed. One way to do this is by using event listeners. Create an abstract class, e.g., `RATmouseListener`:

```
public interface RATmouseListener
{
    public void mouseClicked(parameters)
}
```

This interface is implemented by classes that need to do something when a button is pressed, and then registered with the button object. The way that you let the buttons know about relevant mouse events is left to you. It could be the job of the window manager, window, or window system. Again, think about what makes for the most elegant solution.

You do not have to do any special layout management for your widgets. It is acceptable to assume that the programmer is responsible for laying out the widgets manually in a way that they do not overlap.

2. HelloWorldGraphical Revisited

To demonstrate the functionality of your toolkit, rewrite the `HelloWorldGraphical` application from Assignment 1 (the basic one) using RAT.

Testing Your Understanding

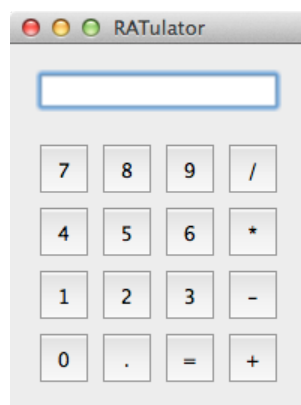
Answer the following questions:

1. List all of the classes in your window system implementation, and classify them as: Graphics and Event Library; Base Window System; Window Manager; UI Toolkit; or Application.
2. Compare and contrast the use of event listeners to standard callbacks in C.

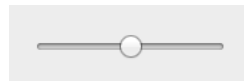
Extra Credit

Implement **one** of the following “extra features” for a combined bonus credit of up to 1.0:

- A basic (working) calculator program using RAT (instead of `HelloWorldGraphical`).
- A slider widget which lets one choose a value between 0.0 and 1.0 by dragging on the handle. Augment `HelloWorldGraphical` to use this to change the color of the label.



1. Calculator Program



2. Slider Widget

Be sure to document in your **README.pdf** which features you implemented.

Submission

Make sure that you only use functions that are available across platforms such that your submission runs on the lab machines (JDK8 – nothing else).

As usual, upload your submission to the L2P exercise course as a **ZIP archive** (**A4_GroupLetter.zip**).

Your assignment archive should include your source code and everything necessary to compile and run your program. Be sure to document your source code. Include a **PDF** document with the name **README.pdf** that contains:

- names and email addresses of all group members
- instructions on how to compile and run your source code
- answers to the questions (see *Testing Your Understanding*)
- non-obvious things you did in your code (if any)
- anything that you think makes your design particularly optimized and/or elegant

Be prepared to discuss your solution in the next lab.

Grading

The assignment will be graded on the following rough scale:

1.0	exceptional work that clearly went above and beyond what was given on the exercise
2.0	exercise was completed satisfactorily as per the assignment specification
3.0	exercise was completed, but has some problems
4.0	incomplete exercise
5.0	little or no effort was put into the exercise

Late assignments will *not* be graded.