CSC171 — Homework 8

Events and Graphics

The goal of this assignment is to give you experience with handling events in graphical applications.

For the following questions, you need to define a "canvas" class that extends `JComponent`. You also need a class that tests your canvas by creating a `JFrame`, adding the canvas to it, and preparing the frame (set size, set window closing behavior, make it visible). You should be able to answer all questions in one class, but make sure that whatever you do, it is clear to the TAs.

Questions

1. Write a canvas class that extends `JComponent`. Make it implement the `MouseListener` interface. In the `MouseListener` methods, just print the event that is passed as argument to the constructor. In the canvas class constructor, have it add itself as its own `MouseListener`.

2. Add code to make your canvas draw a small filled circle on itself when it receives a `mouseClicked` event, at the location of the event: use `MouseEvent.getX()` and `MouseEvent.getY()`.

3. Have your canvas implement the `java.awt.event.MouseMotionListener` interface and add itself as its own `MouseMotionListener`. Make your program draw small filled circles as the mouse is dragged. Congratulations: you've recreated the basics of MacPaint!

4. Extend your program to work as follows: When the mouse is first pressed, draw a small circle as above. As the mouse is dragged, draw a line connecting the original point to the current position of the mouse. Erase the previous line first so that your canvas doesn't get covered in black. When the mouse is released, leave the last line on the canvas. Think about what you need to keep track of between events to make this work. This needs to be part of the state of your canvas...

5. Have your canvas implement the `java.awt.event.KeyListener` interface, add itself as its own `KeyListener`, and call `setFocusable(true)` in order to receive keyboard focus. When a key is typed, your program should draw the corresponding character (see `KeyEvent.getKeyChar()`) on the canvas at the location of the event. If another key is typed without the mouse being clicked, then draw the next character to the right of the previous one as if you were typing in a text field. Again, think about what state you need to maintain to do this. It doesn't have to be perfect (you can hard-code the width of the characters if you like).

6. Let the user change the color used for painting by pressing a number key. Let the different numeric keys correspond to different colors and switch the drawing color depending on which key is pressed. If you want to use the numeric keypad for this, you might look into `KeyEvent.getKeyLocation()` and `KeyEvent.KEY_LOCATION_NUMPAD`, as well as the `VK_NUMPAD` constants.

Grading Scheme

Equal weight for each part.

Doesn't compile or is trivial	$< 50\%$
Compiles and is non-trivial	$\geq 50\%$
Complete and correct with good style and comments	100%
Incomplete, incorrect, bad style, no comments	$< 100\%$

Submission Requirements

Your submission **MUST** include a file named "`README.txt`" with your name, your NetID, the assignment number, and your lab section. This file should explain anything we need to know about how to build and run your project. In particular, be sure to explain how to run what parts of your submission for each question in the assignment.

Submit your solution as a single ZIP archive to BlackBoard before the deadline.

Late homeworks will not be graded and will receive a grade of 0.

All assignments and activities associated with this course must be performed in accordance with the University of Rochester's Academic Honesty Policy.