In this week’s lab, you will be using some of what we have talked about in class regarding Graphical User Interface programming to modify an existing GUI program to incorporate some new components. If you want some additional background on Java GUI programming, you can check out the Java Tutorial:

http://java.sun.com/docs/books/tutorial/uiswing/index.html

On the class assignments page, you will find the program ColorChanger. Here is what this program looks like:

![ColorChanger GUI](image)

The user enters either one of the sixteen color names recognized by the Color class (“red,” “blue,” “magenta,” etc. – see the API docs for the full list) or a hexadecimal string in the format shown above where the first two digits represent the red component of the color, the next two the green component, and the last two the blue component. Clicking the “Change Color” button displays the corresponding color. “Quit now” terminates the application. The code for this application is quite short, so have a look at it and familiarize yourself with how it works.

Since human beings don’t think too well in hexadecimal, I’d like for you to modify the program so that it accepts color descriptions as three decimal numbers ranging from 0 to 255 inclusive, one number each for the red, green and blue components of the color, and then displays the requested color when the “Change Color” button is clicked. Your first
job is to modify this application so it behaves in this way. Here is a picture of how it should look:

![ColorChanger](image)

You can call this class ColorChanger2. To implement this change, you will be adding some JLabels and JTextFields to the topmost panel and modifying the actionPerformed method’s response to the “Change Color” button. For that job, you will need to know how to grab the current text from a JTextField, convert the string representation of a number to an actual int (Integer class is helpful there) and create a new color from decimal R, G and B values (see the Color class documentation for this).

That’s pretty good, but if you really want to explore the color space, it gets tiresome typing in all those numbers. The second task is to add up and down arrow buttons for each of the three text fields that let you “nudge” the corresponding value up or down by 1. Clicking the up or down button for a text field should check the resulting value to see that it is in range, update the text field if so, and change the color immediately. Here’s how that version should look:
For this change, you’ll need to know how to create ImageIcon objects from an image file (see the ImageIcon class documentation). You will find up and down arrow images on the class assignments web page. You’ll need to create a JButton that displays an image rather than a textual label (See JButton documentation, and note that the ImageIcon class implements the Icon interface). To get the buttons to size themselves to the image, you’ll need to use the setPreferredSize method on your JButtons. The icons are 16 x 16 pixels. You’ll also be adding code to the actionPerformed method for the new buttons. And, you’ll need to know how to change the text of a JTextField. You can call this version of the class ColorChanger3.

Because this is the first try for most of you at doing Swing programming, I don’t expect you to finish these two tasks during lab. Your two program files are due Monday by 11:59 PM

Extra credit:

“Nudging” by one with each click can really tire your mouse finger out. Changing one of the RGB values by one doesn’t result in much visible change to the color. Enhance ColorChanger3 by adding a new text field for the “nudge” factor and use this number as the value to increase or decrease the value each time one of the arrow buttons is clicked. Sorry, no pictures of how this one should look…