John Bennett

CMIS 242

Final Project

5/11/12

In this project we were required to store color/hex value pairs in a collection, then create a radio button for each color. The GUI would then have radio buttons display the hex value and when clicked, would change the background color.

I chose HashMap as my collection to store the values. The key requirement was storing up to 20 pairs of colors, and so the first thought was how much code this was going to take. So far in class, we have learned to use three lines for creating one button and an additional 4 lines for each listener:

* Create the JRadioButton (1 line)
* Add the button to a group (1 line)
* Add the button to the JPanel (1 line)
* Add a listener for the button (4 lines- assuming an anonymous listener)

To add 20 color pairs to a GUI would mean that one would have to create a file with 160 lines (if you include the HashMap creation) of redundant code, where the only thing that would change would be the name of the button variable that was created.

I decided to attempt to use the for-each iteration techniques used in Module 5 Section 3 to assist with making the code more dynamic and much shorter. I started by using the HashMap of values to create add anonymous JRadioButton objects that were referenced by the Keys from the color HashMap:

(Red>FF0000) (Red > new JRadioButton)

I attempted to make a simple List or array of JRadioButtons, and then use their positions in the array as links to the button objects, but I had problems with creating a for loop that would add the buttons to groups and the JPanel. I decided to use a HashMap for this reason, and also, the button and color pair would be easily linked together, having the same Key.

The end solution utilizes two for-each loops, one that creates a new button and adds it to the JPanel, and the other that assigns each JRadioButton to a group, and then assigns an ItemListener to the button.

The difficult part was getting the hex value passed to the listener to change the background color. I found that creating a constructor with an argument was the best way to do so. I declared an inner class called RadioListener which implements ItemListener, and the constructor receives the String for the hex value from the color HexMap.

To handle the action from the listener, I created a method that uses the Hex value string, then converts it from a String to an Integer of R/G/B values for use with the Color class. Variables r,g, and b were created and then the substring method was used to grab each pair of values to convert to Integers. The parseInt method allows for an additional argument for base 16, to handle letters up to F in the String. The r/g/b variables were then passed to the setBackgroundColor method using Color(r, g, b).

The application requires 4 files, using Module 3 Section 4, Example 12 as reference:

**-JohnBennettFinalProject.java**

-**GeneralFrame.java**, which contains the outer frame

-**RadioButtonPanel.java**, which contains all the code for the radio buttons

-**ColorPanel.java**, which contains the code for the background color panel

Here is a screen shot of the final application:



