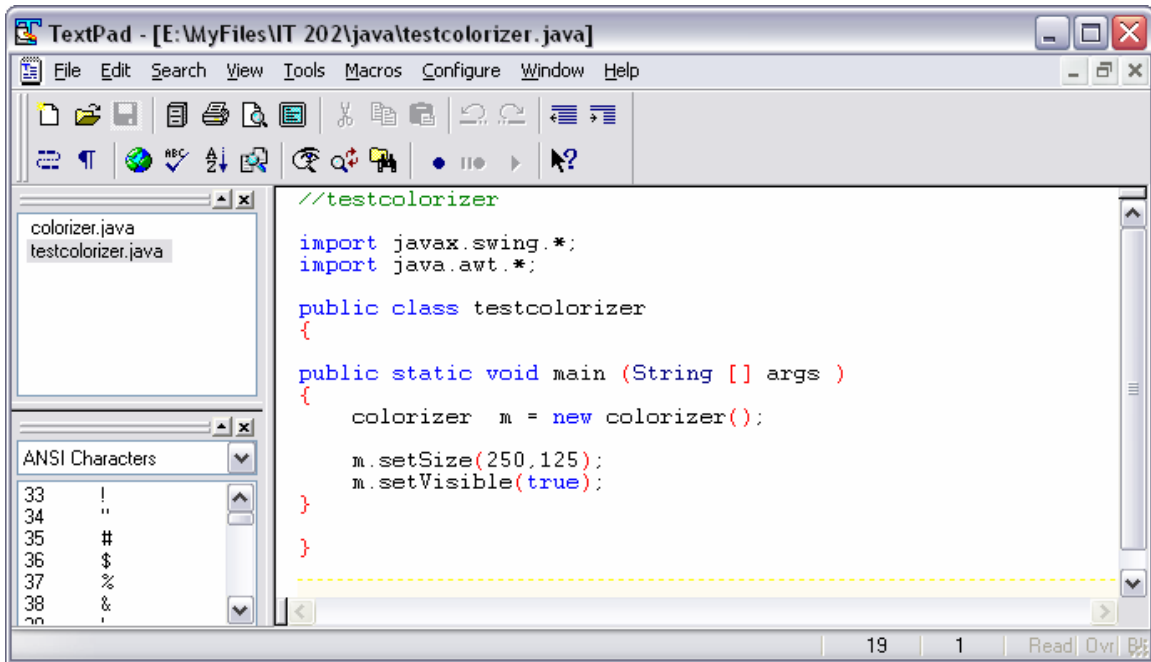


James Stump
Colormaker Assignment

Prof. McHugh
12-13-2004

First, I took the original Y.java and TestY.java, and copied it to colorizer.java and testcolorizer.java respectively.

The testcolorizer.java, is basically the same, but with different class and object names to reflect my program, like the following code:

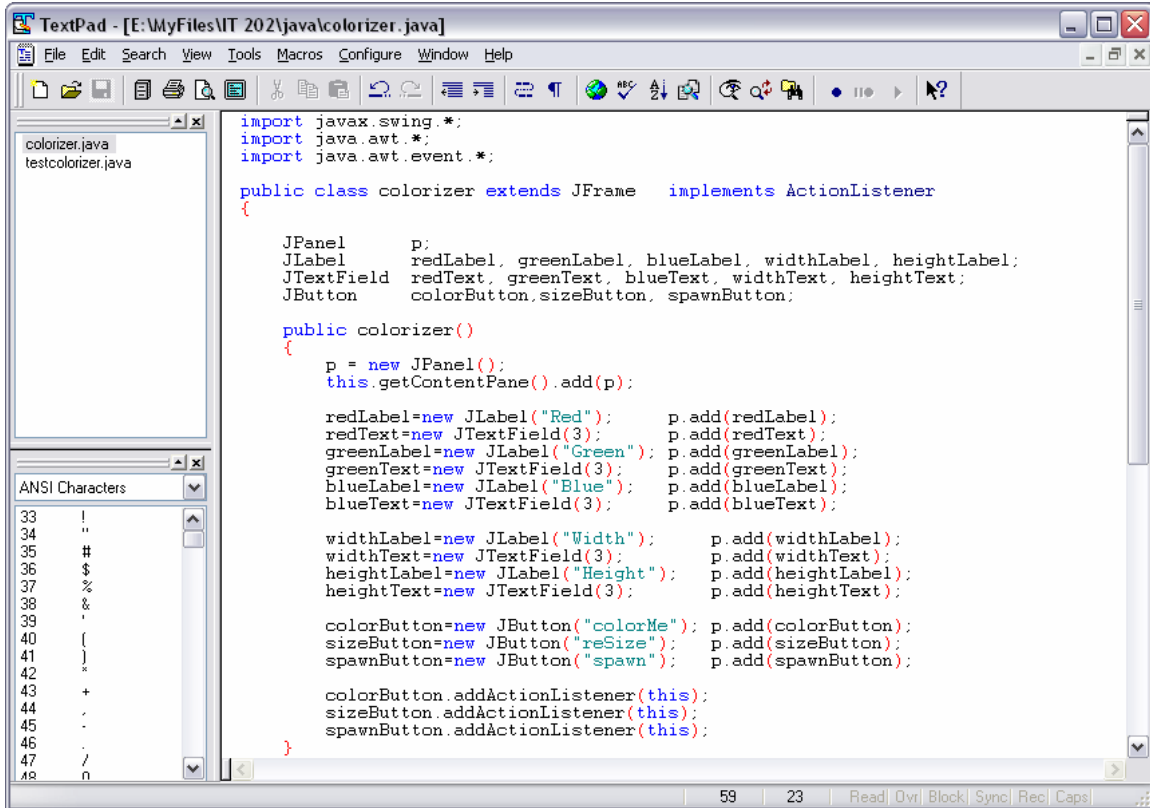


```
//testcolorizer
import javax.swing.*;
import java.awt.*;

public class testcolorizer
{
public static void main (String [] args )
{
    colorizer m = new colorizer();

    m.setSize(250,125);
    m.setVisible(true);
}
}
```

Then I set up the JPanel by utilizing the syntax from the Y.java file, by adding this code.



The screenshot shows a TextPad window titled "TextPad - [E:\MyFiles\IT 202\java\colorizer.java]". The window contains the following Java code:

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

public class colorizer extends JFrame implements ActionListener
{
    JPanel      p;
    JLabel      redLabel, greenLabel, blueLabel, widthLabel, heightLabel;
    JTextField  redText, greenText, blueText, widthText, heightText;
    JButton      colorButton, sizeButton, spawnButton;

    public colorizer()
    {
        p = new JPanel();
        this.getContentPane().add(p);

        redLabel=new JLabel("Red");      p.add(redLabel);
        redText=new JTextField(3);      p.add(redText);
        greenLabel=new JLabel("Green");  p.add(greenLabel);
        greenText=new JTextField(3);    p.add(greenText);
        blueLabel=new JLabel("Blue");    p.add(blueLabel);
        blueText=new JTextField(3);     p.add(blueText);

        widthLabel=new JLabel("Width");  p.add(widthLabel);
        widthText=new JTextField(3);     p.add(widthText);
        heightLabel=new JLabel("Height"); p.add(heightLabel);
        heightText=new JTextField(3);    p.add(heightText);

        colorButton=new JButton("colorMe"); p.add(colorButton);
        sizeButton=new JButton("reSize");  p.add(sizeButton);
        spawnButton=new JButton("spawn");  p.add(spawnButton);

        colorButton.addActionListener(this);
        sizeButton.addActionListener(this);
        spawnButton.addActionListener(this);
    }
}
```

I then compiled and ran the code using these two commands:



The screenshot shows a Command Prompt window titled "Command Prompt - java testcolorizer". The window contains the following commands:

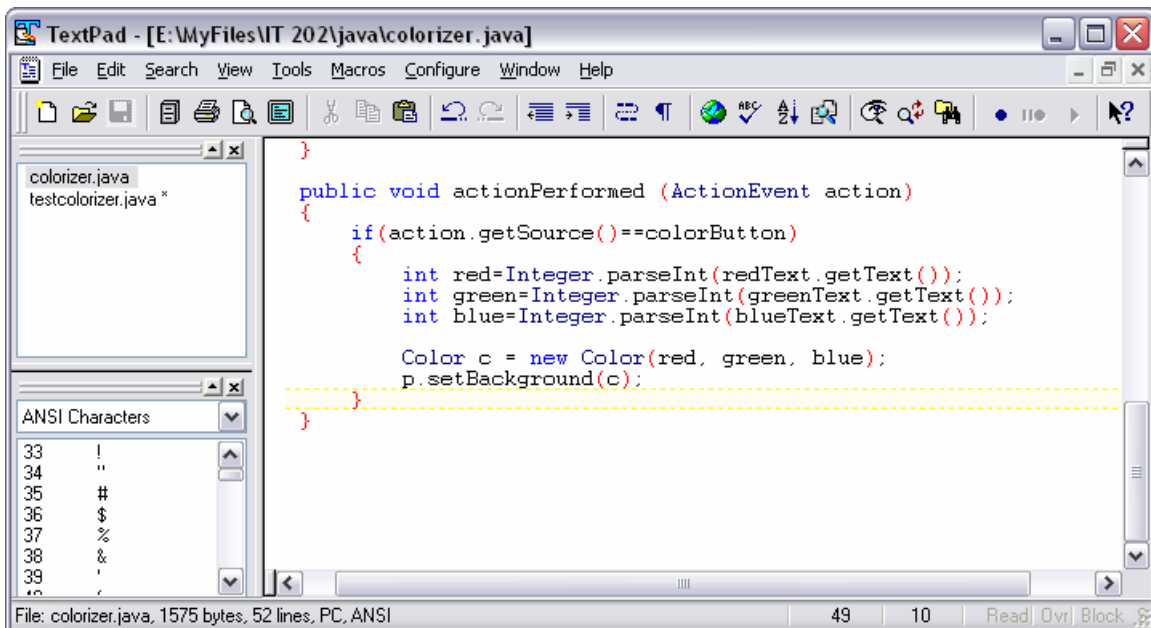
```
E:\MyFiles\IT 202\java>javac testcolorizer.java
E:\MyFiles\IT 202\java>java testcolorizer
```

After running the code, the program popped up like the screen shot below.

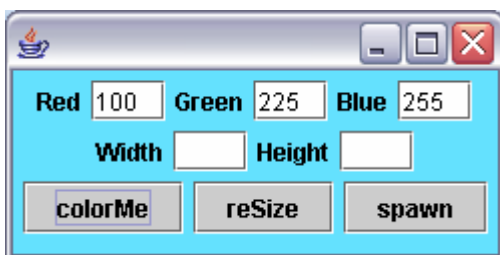


After I got the GUI working, I went and wrote the code that would be executed when the buttons are pressed.

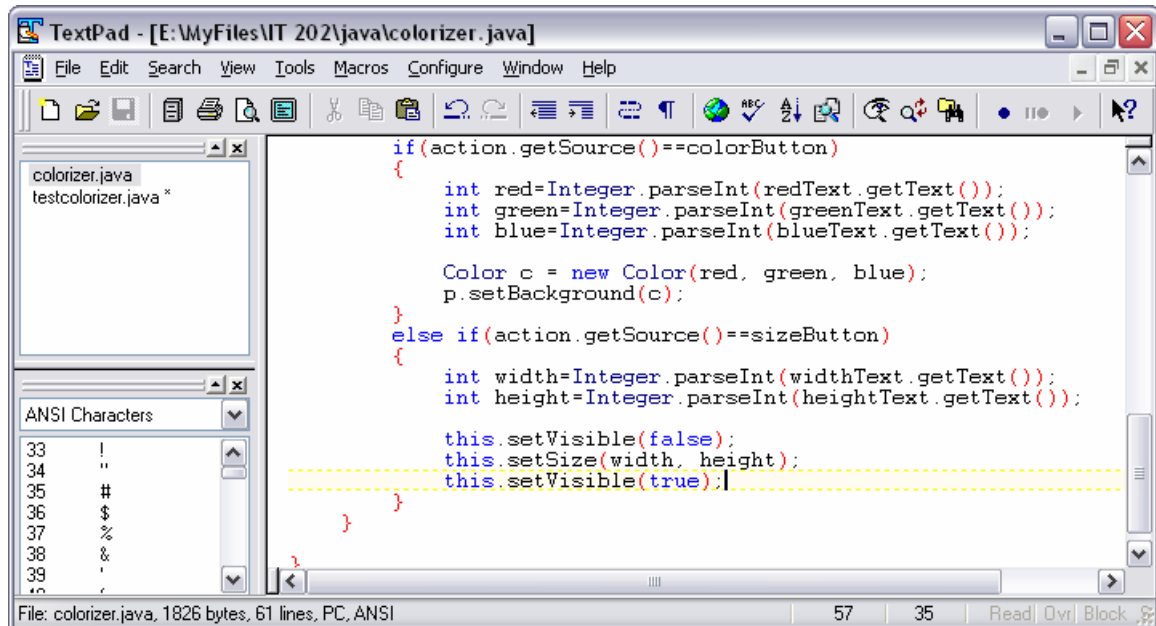
First I started with the background color, and to do that, I added this code:



Then I compiled and ran the code. When the program opened, I put values in for Red, Green, and Blue, and pressed the “colorMe” button, which set the background color to those values, like the screen shot below.



Once that worked, I moved on to the resize, which I got to work using the following code:

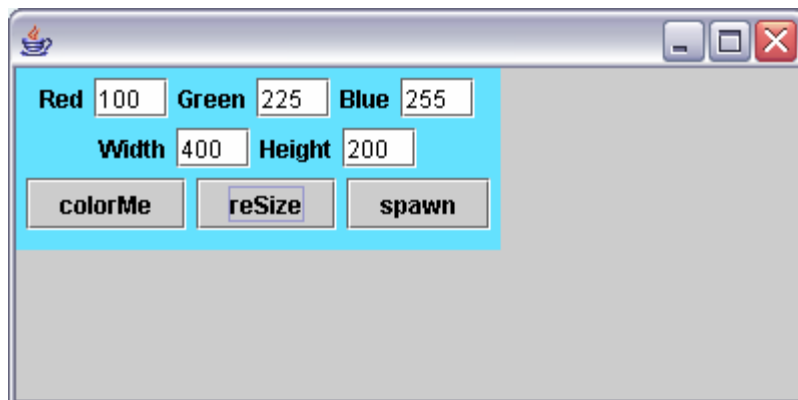


```
if(action.getSource()==colorButton)
{
    int red=Integer.parseInt(redText.getText());
    int green=Integer.parseInt(greenText.getText());
    int blue=Integer.parseInt(blueText.getText());

    Color c = new Color(red, green, blue);
    p.setBackground(c);
}
else if(action.getSource()==sizeButton)
{
    int width=Integer.parseInt(widthText.getText());
    int height=Integer.parseInt(heightText.getText());

    this.setVisible(false);
    this.setSize(width, height);
    this.setVisible(true);
}
}
```

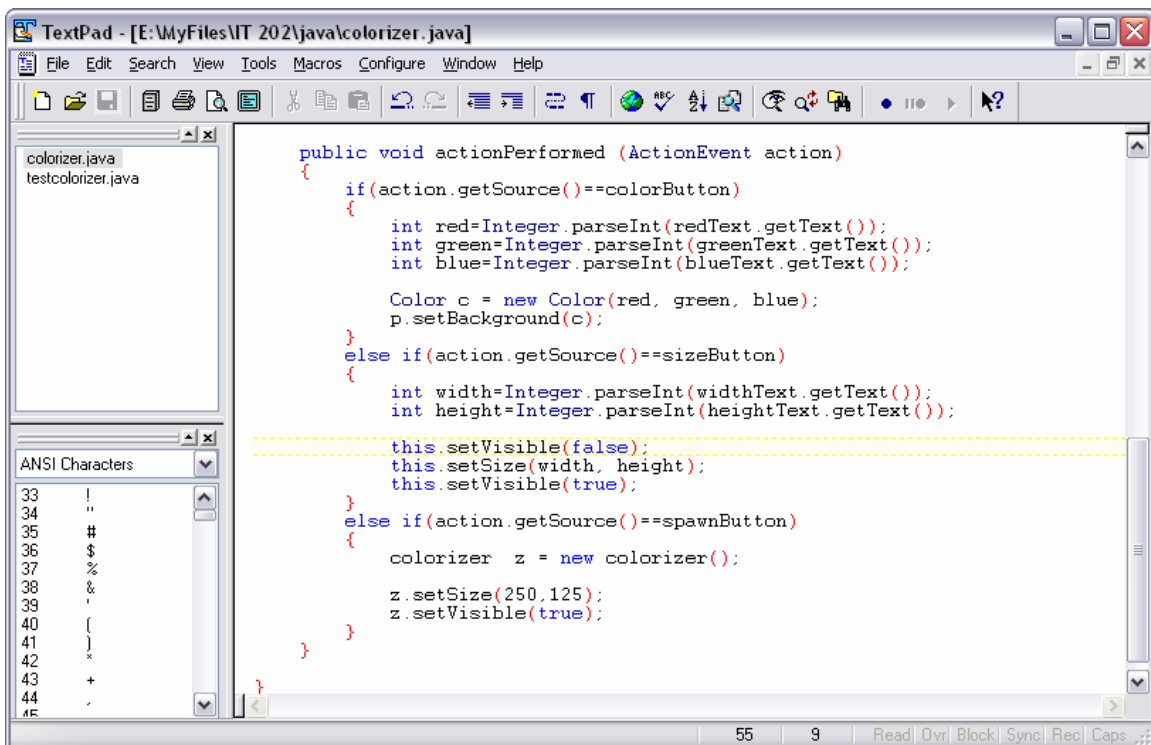
Notice that I placed `this.setVisible(false)`, before `this.setSize(width, height)`, and `this.setVisible(true)`, after it. This forces the window to be refreshed, and fixes the bug that would cause the screen to resize, but the contents, as well background color would not fill the new space. Thus, it would look like this:



Instead though, with those lines in there, it looks like this:



The final function that I needed to get working was the spawn function. This was accomplished by simply copying the code from the main function in testcolorizer, into the place where it would be launched when the spawn button was pressed, which looks like this:



After compiling and running it again, and then pressing the spawn button, another, completely separate window will appear, like the following picture:

