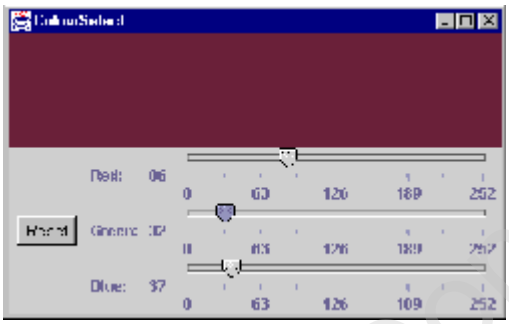
Exercise: 'Swing' ColorSelect

This exercise will give you a good appreciation of Java's 'Swing' GUI Toolkit. You will make use of

- JButton
- JFrame
- JSlider
- JPanel
- JLabel
- Gridlayout, GridBagLayout and BorderLayout layout managers
- Using AWT/Swing event handling

The Exercise

In this exercise you will produce a simple Java application that acts as a colour selector. You should (try and!) duplicate the layout shown here:



The following code should get you started.

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

public class ColourSelect extends JFrame
  {
  private final String RESET = "Reset";
  private LabelledSlider red,
                         green,
                         blue;
  private JPanel colorPanel;
  private JLabel label = new JLabel ("0 0 0");
  class LabelledSlider extends JPanel implements ChangeListener
    {
    private JLabel label;
    private JSlider slider;
    private String prefix;
    public LabelledSlider (String prefix, int val, int min, int max)
      {
      this.prefix = prefix;

      // set the layout to GridBagLayout
      // add a JLabel with GridBagConstraints weightx = gridwidth = 1
      // add a horizontal JSlider with ticks and labels painted in and other values
      //   from the parameters. Use constraints weightx = 5 and gridwidth = 10
      //   and anchor = GridBagConstraints.WEST and fill = GridBagConstraints.HORIZONTAL
      }
    public void setValue (int v)
      {
      label.setText (prefix + v);
      slider.setValue (v);
      }
    public int getValue ()
      {
      return (slider.getValue ());
      }
    public void stateChanged (ChangeEvent e)
      {
      setValue (slider.getValue ());
      }
```

Transerial Pty. Ltd. DonationWare

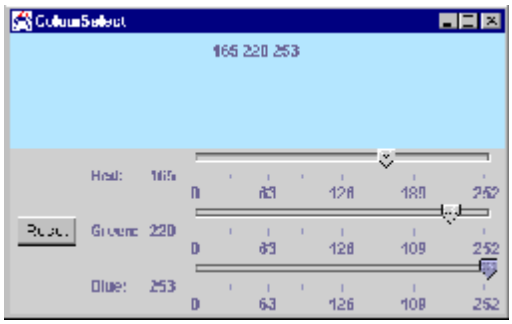
```
  public void addChangeListener (ChangeListener l)
    {
    slider.addChangeListener (l);
    }
  }

public ColourSelect ()
  {
  Container c = getContentPane ();
  setTitle ("ColourSelect");
  JPanel panel = new JPanel (new GridBagLayout ());
  GridBagConstraints gbc = new GridBagConstraints ();
  Button b = new Button (RESET);
  b.addActionListener
      (
      new ActionListener ()
        {
        public void actionPerformed (ActionEvent e)
          {
          redraw (0, 0, 0);
          }
        }
      );
  gbc.weightx = 1;
  gbc.gridwidth = 1;
  panel.add (b, gbc);
  ChangeListener al = new ChangeListener ()
    {
    public void stateChanged (ChangeEvent e)
      {
      // gather the red, green and blue slider values and redraw colorPanel
      }
    };
  JPanel p1 = new JPanel ();
  p1.setLayout (new GridLayout (3, 1));
  p1.add (red = new LabelledSlider ("Red:      ", 0, 0, 255));
  red.addChangeListener (al);
  p1.add (green = new LabelledSlider ("Green:   ", 0, 0, 255));
  green.addChangeListener (al);
  p1.add ( blue = new LabelledSlider ("Blue:      ", 0, 0, 255));
  blue.addChangeListener (al);
  gbc.weightx = 5;
  gbc.gridwidth = GridBagConstraints.REMAINDER;
  gbc.fill = GridBagConstraints.HORIZONTAL;
  gbc.anchor = GridBagConstraints.WEST;
  panel.add (p1, gbc);
  c.add (BorderLayout.SOUTH, panel);
  c.add (BorderLayout.CENTER, colorPanel = new JPanel ());
  colorPanel.add (label);
  redraw (0, 0, 0);
  }

private void redraw (int r, int g, int b)
  {
  red.setValue (r);
  green.setValue (g);
  blue.setValue (b);
  colorPanel.setBackground (new Color (r, g, b));
  colorPanel.repaint ();
  }

public static void main (String [] args)
  {
  ColourSelect f = new ColourSelect ();
  f.setSize (400, 250);
  f.show ();
  f.addWindowListener
    (
    new WindowAdapter ()
      {
      public void windowClosing (WindowEvent e)
        {
        System.exit (0);
        }
      }
```

Transcentia Pty. Ltd. DonationWare

```
        );
      }
  }
```

Other Work

A simple modification is to add a further JLabel within the colorPanel to show the actual colour being displayed, as shown below:



You will be able to add this extra functionality with very little extra work. All that is required is to:

- add a new JLabel instance to colorPanel
- add the following code to the ColorSelect constructor

```
ChangeListener labelListener = new ChangeListener ()
  {
  public void stateChanged (ChangeEvent e)
    {
    final String SP = " ";
    label.setText (red.getValue () + SP + green.getValue () + SP + blue.getValue ());
    }
  };
```

- establish the listener labelListener as a second listener (alongside al) for each JSlider

This shows that it is easy to have more than one listener for a component, and this can be a useful facility (although in this particular case, both listeners can easily be amalgamated into one).

Transentia Pty. Ltd. DonationWare