TCP2201 Object Oriented Analysis and Design

Lab 6 – More Swing Components and Listeners

|  |
| --- |
| **Lab outcomes** |

By the end of today’s lab, you should be able to

* Make use of new Java GUI Swing components – JColorChooser, JSlider
* Make use of new Java Swing listeners – ChangeListener, MouseListener, MouseMotionListener
* Implement mouse listeners to handle mouse interactivity
* Handle paint/paintComponent methods and the Graphics object

|  |
| --- |
| **Mouse listeners** |

Mouse activity can be handled with the two listeners MouseListener and MouseMotionListener. Both are interface objects and need to have all their methods overridden if they are “*implements”* in a class (there are other ways but not covered in TCP2201). The methods in the listeners are

MouseListener

void mouseClicked (MouseEvent e)

void mousePressed (MouseEvent e)

void mouseReleased (MouseEvent e)

void mouseEntered (MouseEvent e)

void mouseExited (MouseEvent e)

MouseMotionListener

void mouseMoved (MouseEvent e)

void mouseDragged (MouseEvent e)

To detect clicks, use the mousePressed (continuous press), mouseClicked (press+release) or mouseDragged (press+move) method. To detect buttons call the getModifiers() method from MouseEvent and compare with BUTTON1\_MASK, BUTTON2\_MASK and BUTTON3\_MASK for left, middle and right mouse buttons.

|  |
| --- |
| **Exercise 1: Simple Java drawing program** |

The following code presents the framework for a Java Swing application to create a drawing program.

|  |
| --- |
| import java.awt.\*;  import java.awt.event.\*;  import javax.swing.\*;  public class DrawingProgram extends JFrame {  public DrawingProgram(){  super("Painter");  JPanel toolbar = new JPanel(new FlowLayout(FlowLayout.LEFT));  toolbar.add(new Label("Drag mouse to draw"));  this.add(toolbar,BorderLayout.SOUTH);  setSize(800,600);  setVisible(true);  setDefaultCloseOperation(EXIT\_ON\_CLOSE);  }    public static void main(String[] a){  new DrawingProgram();  }  } |

The following items need to be added to complete the program to create a working drawing program

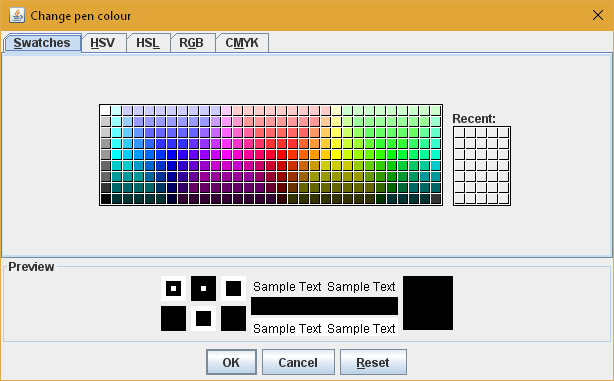
* JPanel object to keep track of the mouse coordinates in the program
* [Panel]jp = new JPanel();
* This.add(jp.BorderLayout.CENTER)

* Point object to keep track of the mouse coordinates in the program
* Private Point mousePnt = new Point(); <- should be declared outside the constructor for access by listener methods later.
* MouseMotionListener attached to the JPanel that triggers each time a ‘drag’ operation is performed
* Add ”implements MouseMotionListener” to class
* If a drag event is thrown, get the coordinates and call repaint
* Paint method to draw object to a JPanel subclass

Look in the [Java API](https://docs.oracle.com/javase/8/docs/api/index.html) for documentation of all these classes.

|  |
| --- |
| **Java Swing JColorChooser** |

The Swing package adds on JColorChooser to easily create a pane of controls allowing a user to manipulate and select a color based on a GUI element. By default, a JColorChooser pane will show options for ready-made colour swatches and HSB/RGB spectrum fields. The JColorChooser returns a Color object



There are multiple ways of *calling* the JColorChooser object into your application depending on how often you need the object.

Calling the showDialog method from JColorChooser

Color newcol = JColorChooser.showDialog(null, “title”, this.getBackground());

Passing a JColorChooser object to the JOptionPane

JColorChooser jcc = new JColorChooser();

JOptionPane.showMessageDialog(null,jcc);

Adding to a application content pane (i.e. container) using the add() methods

this.add(jcc, BorderLayout.END);

References:

<http://docs.oracle.com/javase/tutorial/uiswing/components/colorchooser.html>

<http://docs.oracle.com/javase/tutorial/uiswing/components/colorchooser.html#advancedexample>

|  |
| --- |
| **Exercise 2: Add colour** |

Modify the program from exercise 1 to let you choose what colour to draw with. You may want to implement the change with either a JButton or with mouse clicks identification.

|  |
| --- |
| **Java Swing JSlider** |

The JSlider is a new object class in the Swing package to help create components that allow user to easily enter a bounded numeric value. The JSlider replaces the ScrollBar from AWT and requires the ChangeListener to handle events that are triggered. Since the JSlider originates from the Swing package, the ChangeListener now comes from the javax.swing.event.\* package and must be imported in. To create a JSlider object, the code is as follows (other setups possible, see documentation) along with several options that can be set

JSlider sample = new JSlider(JSlider.HORIZONTAL, FPS\_MIN, FPS\_MAX, FPS\_INIT);

sample.setMajorTickSpacing(10);

sample.setMinorTickSpacing(1);

sample.setPaintTicks(true);

sample.setPaintLabels(true);

The ChangeListener interface object triggers a stateChange event that receives a ChangeEvent object thrown by the JSlider

public void stateChanged(ChangeEvent e) { }

The JSlider object usually returns an integer type object with its maximum and minimum values bounded during setup

Reference: <http://docs.oracle.com/javase/tutorial/uiswing/components/slider.html>

|  |
| --- |
| **Exercise 3: Change size** |

Modify the program from exercise 2 to let you choose the size of the line to draw with.