

CECS 277 – Lecture 15 – GUI Applications

Swing Components and the Action Listener – You can add functionality to your application by adding buttons, textboxes and other components. The ActionListener is an event handler for when these elements are clicked on or modified. Each of the components has several methods associated with them (check the Java API for details).

Layouts – There are several types of layouts you can use to help you with the placement of your GUI components. Some allow you to set sizes, others ignore all settings.

Tutorials for layouts: <https://docs.oracle.com/javase/tutorial/uiswing/layout/visual.html>.

BorderLayout	Breaks up the window into 5 areas: top, bottom, left, right, center. Fields and buttons placed in each area expand to fill that area.
BoxLayout	Places components in a single column.
FlowLayout	Places components in a single row.
GridLayout	Components are equal size and displayed on a grid.
GroupLayout	Used by creating horizontal and vertical groups of components, similar to using Box and Flow layouts simultaneously.
SpringLayout	Define spacing of components based on other components.
null	The position of each component is set using setBounds().

Button – Buttons are created and tied to the ActionListener. They are initialized with text, and/or an image. The button sends an event to the actionPerformed() when clicked.

Label – A label is a non-interactive block of text. Labels are used give titles to sections of your window. Labels can use plain text, images, and html formatting.

TextField – a text field is a one line text area that allows the user to type in it. The value can be returned as a string. It is initialized with a string for the default text, and the width, given in the number of characters. A text field may be associated with the ActionListener and so an event will occur when the Enter key is pressed.

CheckBox – Allows the user to select multiple options. A checkbox can be bound to the ActionListener, but they are usually handled using a submit button to trigger an event in the actionPerformed() methods, there the states of the checkboxes can be tested.

RadioButton – Radio buttons are similar to check boxes, except that they are combined into a group and only one can be selected at a time.

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
public class Panel extends JPanel implements ActionListener{
    private JTextField tfInfo;
    private JLabel lblColor, lblShapes;
    private JCheckBox cbRed, cbBlue;
    private ButtonGroup shapes;
    private JRadioButton rbCircle, rbSquare;
    private JButton btnSubmit;
```

```

public Panel() {
    setLayout( new FlowLayout( FlowLayout.CENTER ) );
    tfInfo = new JTextField( "Color & Shapes", 15 );
    lblColor = new JLabel( "Colors:" );
    cbRed = new JCheckBox( "Red" );
    cbBlue = new JCheckBox( "Blue" );
    lblShapes = new JLabel( "Shapes:" );
    shapes = new ButtonGroup();
    rbCircle = new JRadioButton( "Circle" );
    rbSquare = new JRadioButton( "Square" );
    btnSubmit = new JButton( "Submit" );
    btnSubmit.addActionListener( this );
    add( tfInfo );
    add( lblColor );
    add( cbRed );
    add( cbBlue );
    add( lblShapes );
    add( rbCircle );
    add( rbSquare );
    add( btnSubmit );
    shapes.add( rbCircle );
    shapes.add( rbSquare );
}

public void actionPerformed((ActionEvent a) {
    if( a.getSource() == btnSubmit ) {
        if( cbRed.isSelected() && cbBlue.isSelected() ) {
            if( rbCircle.isSelected() ) {
                tfInfo.setText( "Purple Circle" );
            } else if( rbSquare.isSelected() ) {
                tfInfo.setText( "Purple Square" );
            }
        } else if( cbRed.isSelected() ) {
            if( rbCircle.isSelected() ) {
                tfInfo.setText( "Red Circle" );
            } else if( rbSquare.isSelected() ) {
                tfInfo.setText( "Red Square" );
            }
        } else if( cbBlue.isSelected() ) {
            if( rbCircle.isSelected() ) {
                tfInfo.setText( "Blue Circle" );
            } else if( rbSquare.isSelected() ) {
                tfInfo.setText( "Blue Square" );
            }
        }
    }
}
}

```