**CS 342 Report for (Lab 3)**

**Code**

//Programmer: Fayaz Khan

//Assignment: Lab 3

//Date: September 10, 2015

//Description: Event Handling of GUI components: buttons, radio buttons, checkboxes,

//drop down menus

**import** java.awt.FlowLayout;

**import** java.awt.Graphics;

**import** java.awt.event.\*;//used for ActionListner

**import** javax.swing.\*;

**import** java.awt.Color;

**public** **class** Lab03 **extends** JApplet **implements** ActionListener , ItemListener

{

**private** **final** **int** APPLETWIDTH = 600; //Width for the applet

**private** **final** **int** APPLETHEIGHT = 500; //Height for the applet

**private** **static** **final** String[] ***LOCATION*** = {"","Random", "Top-Left"}; // Used for ComboBox

**private** JPanel firstPanel; //consist of 2 Radio buttons linked and a text box

**private** JRadioButton drawSquare; //1 of 2 R button labeled Draw square

**private** JRadioButton writeMessage; //2 of 2 R button labeled Write message

**private** JTextField textMessage; //Text box next to write message R button

**private** JPanel secondPanel; //consist of Label, ComboBox, CheckBox and button

**private** JLabel whereDraw; //"Select where to draw: "

**private** JComboBox locationMenu; // String[] LOCATION will be used for this

**private** JCheckBox drawColor; //Check box for "Draw in color"

**private** JButton drawButton; //button for "Draw it!"

**private** String message; //For the message box

**private** **boolean** bMessage = **false**; //Boolean for the radio button message

**private** **boolean** bCombo = **false**; //Boolean for the drop down menu

**private** **boolean** bSquare = **false**; //Boolean for the square button message

**private** **boolean** bButton = **false**; //Boolean for the square button message

**private** **int** locSelectIndex=0; //To determine which LOCATION is selected

**private** **int** ranX; //random x cordinate for the square

**private** **int** ranY; //random y cordinate for the square

@Override

**public** **void** init()

{

setLayout(**new** FlowLayout());

ButtonGroup radioLink; // group to link radio buttons

drawSquare = **new** JRadioButton("Draw square", **false**); // create radio buttons

writeMessage = **new** JRadioButton("Write message", **false**);

add(drawSquare); // add them to GUI

add(writeMessage);

drawSquare.addItemListener(**this**);

writeMessage.addItemListener(**this**);

radioLink = **new** ButtonGroup(); // logically link buttons

radioLink.add(drawSquare);

radioLink.add(writeMessage);

textMessage = **new** JTextField(18);

add(textMessage);

textMessage.addActionListener(**this**); //ActionListenerf for text

firstPanel = **new** JPanel(); // create panel

firstPanel.add(drawSquare);

firstPanel.add(writeMessage);

firstPanel.add(textMessage);

add(firstPanel);

whereDraw = **new** JLabel("Select where to draw: ");

add(whereDraw);

**KHAN 2**

locationMenu = **new** JComboBox(***LOCATION***);

locationMenu.setMaximumRowCount(4);

add(locationMenu);

locationMenu.addItemListener(**this**);

drawColor = **new** JCheckBox("Draw in color");

drawColor.addItemListener(**this**);

add (drawColor);

drawButton = **new** JButton("Draw It!");

drawButton.addActionListener(**this**);

add(drawButton);

secondPanel= **new** JPanel();

secondPanel.add(whereDraw);

secondPanel.add(locationMenu);

secondPanel.add(drawColor);

secondPanel.add(drawButton);

add(secondPanel);

}

@Override

**public** **void** paint(Graphics g) // Display results

{

**super**.paint(g);

**if**(bMessage== **true** && bCombo == **true** && bButton == **true**)

{

g.drawString(message, 60, 100);

}

**if**(bSquare== **true** && bCombo == **true** && bButton == **true** )

{

**if**(locationMenu.getSelectedIndex()== 1)//RANDOM

{

ranX = (**int**) (Math.*random*() \*((APPLETWIDTH-65)+1)+65);

ranY = (**int**) (Math.*random*() \*((APPLETHEIGHT-165)+165));

//JOptionPane.*showMessageDialog*(**null**,"Draw Random SQUARE"+ APPLETWIDTH+ " "+ ranY);

g.drawRect(ranX, ranY, 100, 100);

}

**if**(locationMenu.getSelectedIndex()== 2)//Top-Left

{

//JOptionPane.*showMessageDialog*(**null**,"Draw Top-left SQUARE");

g.drawRect(10, 75, 100, 100);

}

}

bButton = **false**;

}

**public** **void** actionPerformed (ActionEvent e)

{

**if**(e.getSource()== drawButton)

{

bButton = **true**;

}

**if**(bMessage== **true** && bCombo == **true**)

{

message = textMessage.getText();

}

repaint();

}

**KHAN 3**

**public** **void** itemStateChanged(ItemEvent e)

{

**if**(e.getSource()== drawColor && e.getStateChange() == ItemEvent.***SELECTED***)

{

JOptionPane.*showMessageDialog*(**null**,"Color ON");

}

**if**(e.getSource()== drawColor && e.getStateChange() == ItemEvent.***DESELECTED***)

{

JOptionPane.*showMessageDialog*(**null**,"Color OFF");

}

**if**(e.getSource() == writeMessage && e.getStateChange()== ItemEvent.***SELECTED*** )

{

bMessage = **true**;

bSquare = **false**;

}

**if**(e.getSource() == locationMenu && e.getStateChange()== ItemEvent.***SELECTED*** )

{

bCombo = **true**;

}

**if**(e.getSource() == drawSquare && e.getStateChange()== ItemEvent.***SELECTED*** )

{

bMessage = **false**;

bSquare = **true**;

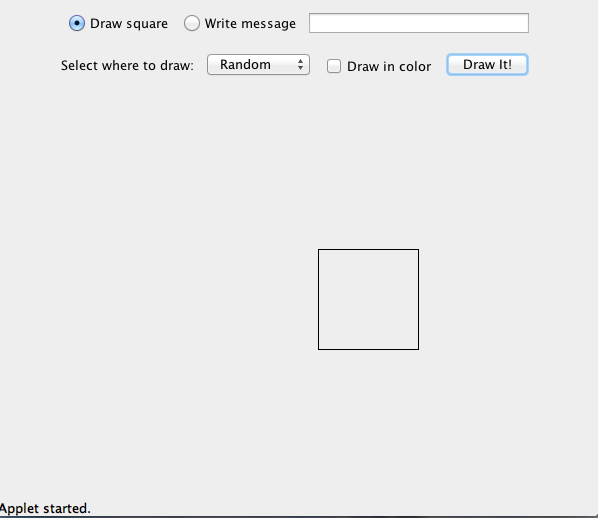
}

repaint();

}

}

**Sample Runs**



**KHAN 4**

