Lab Report – Java GUI Implementation no Event Listening Programming Assignment

John Marshall
CSCI112 Spring 2023

Assignment Analysis and Design

In your own words, describe the problem including input and output. Briefly describe how you developed your code. Briefly describe what your code does and how it works – including anything different or unique or special that you did in your software. If the software is long or complicated, describe how it is organized.
Include a copy of any pseudocode or other design documents you used. If you worked with anyone else, asked anyone for help, or looked anything up, then mention it here. Include proper references to source material.

The purpose of this assignment was to create a Universal Remote using Java GUI. I started with briefly listing what the top user needs maybe and then creating a wireframe sticking to common conventions for Remotes. After that it was straight on to coding. With a Background in both UI and UX, this project was pretty straight forward. The code is designed to be a straight flow from the top of the remote to the bottom. The code basically repeats the process of

Wireframe included in folder as PDF

John Marshall PseduoCode
public class Main
  public static void main(String[] args)

        //first JFrame
remoteFrame = JFrame
remoteLayout = FlowLayout.CENTER
remoteFrame.setLayout remoteLayout

        //Contents Frame
remoteContentsFrame = JPanel
remoteContentsFrameLayout = BoxLayout(y_axis)
remoteContentsFrame.setLayout(remoteContentsFrameLayout)

        //On Button--
JPanel powerButtonContainer = JPanel
FlowLayout powerButtonLayout = FlowLayout(right)
powerButtonContainer.setLayout(powerButtonLayout)

```
        //radio button to show System Status
JRadioButton powerButton = "Power"
        //adding first panel to second
powerButtonContainer.add(powerButton)

    //Input---------------------------------------------------------------------
JPanel inputButtonContainer =  JPanel
FlowLayout inputButtonLayout =  FlowLayout(center)
inputButtonContainer.setLayout(inputButtonLayout)

    //JCombo to show users input Options
String[] inputs =  {"HDMI1","HDMI2","HDMI3","TV","Composite"}
JComboBox inputSelector =  JComboBox(inputs)
inputButtonContainer.add(inputSelector)


    //Channel Buttons-----------------------------------------------------------------

JPanel channelButtonContainer=  JPanel
FlowLayout channelButtonLayout =  FlowLayout(FlowLayout.CENTER)
channelButtonContainer.setLayout(channelButtonLayout)

    //Second JPanel
 JPanel channelButtonContentsContainer =  JPanel
BoxLayout channelButtonContentsLayout =  BoxLayout( BoxLayout.Y_AXIS)
channelButtonContentsContainer.setLayout(channelButtonContentsLayout)

    //Controller Buttons
JButton channelUp =  "CHNL ▲"
JButton channelDown = "CHNL ▼"
    //adding Buttons
channelButtonContentsContainer.add(channelUp)
channelButtonContentsContainer.add(channelDown)
    //adding Double Panel
channelButtonContainer.add(channelButtonContentsContainer)


        //VolumeSlider ----------------------------------------------------------------

JPanel volumeButtonContainer =  JPanel
FlowLayout volumeButtonLayout =  FlowLayout(center)
volumeButtonContainer.setLayout(volumeButtonLayout)
```

```
        //contents Panel
volumeButtonContentsContainer =  JPanel
BoxLayout  volumeButtonContentsLayout =  BoxLayout( BoxLayout.Y_AXIS)
volumeButtonContentsContainer.setLayout(volumeButtonContentsLayout)

        //creating volume Slider and Label
JLabel  volumeLabel =  JLabel("Volume")
JSlider volumeSlider =  JSlider

        //add label & buttons to panel
volumeButtonContentsContainer.add(volumeLabel)
volumeButtonContentsContainer.add(volumeSlider)

        //add contents panel to wrapper panel
volumeButtonContainer.add(volumeButtonContentsContainer)


         //Nav Buttons---------------------------------------------------------------------
JPanel arrowButtonContainer=  JPanel
BorderLayout arrowButtonLayout =  BorderLayout
arrowButtonContainer.setLayout(arrowButtonLayout)
         //nav gaps
arrowButtonLayout.setVgap(4)
arrowButtonLayout.setHgap(4)

         // Creating Buttons
JButton upArrow = ("↑")
JButton downArrow = ("↓")
JButton rightArrow = ("→")
JButton LeftArrow = ("←")
JButton enterButton = ("ENTER")

        //Adding Nav Buttons
arrowButtonContainer.add(NORTH)
arrowButtonContainer.add(SOUTH)
arrowButtonContainer.add(EAST)
arrowButtonContainer.add(WEST)
arrowButtonContainer.add(CENTER)



     //NumPAD---------------------------------------------------------------------
JPanel numPadContainer =  JPanel
GridLayout numPadLayout =  GridLayout(4,3)
```

```
      //numpad Gaps
numPadLayout.setVgap(4)
numPadLayout.setHgap(4)
numPadContainer.setLayout(numPadLayout)

          //creating and adding numpad buttons 1-9

          for (int i = 1 i <= 9 i++){
label = Integer.toString(i)
JButton numPadButton = (label)
numPadContainer.add(numPadButton)
      }
           //creating & Add 0 and spacer for Numpad layout
      spacer1 =  JPanel
      JButton numPad0 = ("0")
      numPadContainer.add(spacer1)
      numPadContainer.add(numPad0)

      //Content Adding to Main Frame-----------------------------------------------------------

      //elements being added to Contents Frame
      //spacers are "remoteContentsFrame.add(Box.createRigidArea( Dimension(0, 20)))"

      remoteContentsFrame.add(Box.createRigidArea( Dimension(0, 20)))
          //Power Button
remoteContentsFrame.add(powerButtonContainer)
remoteContentsFrame.add(Box.createRigidArea( Dimension(0, 20)))
          //Input Selector
remoteContentsFrame.add(inputButtonContainer)
remoteContentsFrame.add(Box.createRigidArea( Dimension(0, 20)))
          //Up and Down Buttons
remoteContentsFrame.add(channelButtonContainer)
remoteContentsFrame.add(Box.createRigidArea( Dimension(0, 20)))
           //volume Control
remoteContentsFrame.add(volumeButtonContainer)
remoteContentsFrame.add(Box.createRigidArea( Dimension(0, 20)))
           //Selectors
remoteContentsFrame.add(arrowButtonContainer)
remoteContentsFrame.add(Box.createRigidArea( Dimension(0, 20)))
          //NumberPad
remoteContentsFrame.add(numPadContainer)

      //Main Jframe adding & settings
remoteFrame.add(remoteContentsFrame)
```

```
remoteFrame.setTitle("Universal Remote")
remoteFrame.setSize(360, 620)
remoteFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE)
remoteFrame.setVisible(true)

  }//end main func
}//end main class
```

Assignment Code

Include the code for your assignment unless otherwise directed by the assignment or by your instructor, which will be a copy of your Python project submitted with the report. You can put the report and the Python project all in one submission. In the report, either tell the reader that it is attached file or include the code.

```
/*
John Marshall
CSCI112 Universal Remote Assignment
3/20/2023
*/

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

public class Main {
  public static void main(String[] args) {

      //first JFrame ------------------------------------------------------------
      JFrame remoteFrame = new JFrame();
      FlowLayout remoteLayout = new FlowLayout(FlowLayout.CENTER);
      remoteFrame.setLayout(remoteLayout);


      //Contents Frame----------------------------------------------------------------
      //Second wrapper so contents can always be center within the remoteFrame
      JPanel remoteContentsFrame = new JPanel();
```

```
    BoxLayout remoteContentsFrameLayout = new BoxLayout(remoteContentsFrame,
BoxLayout.Y_AXIS);
    remoteContentsFrame.setLayout(remoteContentsFrameLayout);

    //On Button----------------------------------------------------------------
    JPanel powerButtonContainer = new JPanel();
    FlowLayout powerButtonLayout = new FlowLayout(FlowLayout.RIGHT);
    powerButtonContainer.setLayout(powerButtonLayout);
    //radio button to show System Status
    JRadioButton powerButton = new JRadioButton("Power");
    //adding first panel to second
    powerButtonContainer.add(powerButton);

    //Input---------------------------------------------------------------------
    JPanel inputButtonContainer = new JPanel();
    FlowLayout inputButtonLayout = new FlowLayout(FlowLayout.CENTER);
    inputButtonContainer.setLayout(inputButtonLayout);

    //JCombo to show users input Options
    String[] inputs = new String[] {"HDMI1","HDMI2","HDMI3","TV","Composite"};
    JComboBox inputSelector = new JComboBox(inputs);
    inputButtonContainer.add(inputSelector);


    //Channel Buttons-----------------------------------------------------------------
    //channel Controls Double Wrapped to center buttons
    JPanel channelButtonContainer= new JPanel();
    FlowLayout channelButtonLayout = new FlowLayout(FlowLayout.CENTER);
    channelButtonContainer.setLayout(channelButtonLayout);

    //Second JPanel
    JPanel channelButtonContentsContainer = new JPanel();
    BoxLayout channelButtonContentsLayout = new
BoxLayout(channelButtonContentsContainer, BoxLayout.Y_AXIS);
    channelButtonContentsContainer.setLayout(channelButtonContentsLayout);

    //Controller Buttons
    JButton channelUp = new JButton("CHNL ▲");
    JButton channelDown = new JButton("CHNL ▼");

    //adding Buttons
    channelButtonContentsContainer.add(channelUp);
    channelButtonContentsContainer.add(channelDown);
    //adding Double Panel
```

```
        channelButtonContainer.add(channelButtonContentsContainer);


        //VolumeSlider ----------------------------------------------------------------
        //Volume Slider Double wrapped
        JPanel volumeButtonContainer = new JPanel();
        FlowLayout volumeButtonLayout = new FlowLayout(FlowLayout.CENTER);
        volumeButtonContainer.setLayout(volumeButtonLayout);

        //volume Second JPanel
        JPanel volumeButtonContentsContainer = new JPanel();
        BoxLayout  volumeButtonContentsLayout = new
    BoxLayout(volumeButtonContentsContainer, BoxLayout.Y_AXIS);
        volumeButtonContentsContainer.setLayout(volumeButtonContentsLayout);

        //creating volume Slider and Label
        JLabel  volumeLabel = new JLabel("Volume");
        JSlider volumeSlider = new JSlider();
        //add label&buttons to panel
        volumeButtonContentsContainer.add(volumeLabel);
        volumeButtonContentsContainer.add(volumeSlider);

        //add contents panel to wrapper panel
        volumeButtonContainer.add(volumeButtonContentsContainer);


        //Nav Buttons----------------------------------------------------------------
        JPanel arrowButtonContainer= new JPanel();
        BorderLayout arrowButtonLayout = new BorderLayout();
        arrowButtonContainer.setLayout(arrowButtonLayout);
        //nav gaps
        arrowButtonLayout.setVgap(4);
        arrowButtonLayout.setHgap(4);

        // Creating Buttons
        JButton upArrow = new JButton("↑");
        JButton downArrow = new JButton("↓");
        JButton rightArrow = new JButton("→");
        JButton LeftArrow = new JButton("←");
        JButton enterButton = new JButton("ENTER");

        //Adding Nav Buttons
        arrowButtonContainer.add(upArrow, BorderLayout.NORTH);
        arrowButtonContainer.add(downArrow, BorderLayout.SOUTH);
```

```
arrowButtonContainer.add(rightArrow, BorderLayout.EAST);
arrowButtonContainer.add(LeftArrow, BorderLayout.WEST);
arrowButtonContainer.add(enterButton, BorderLayout.CENTER);



//NumPAD---------------------------------------------------------------------
JPanel numPadContainer = new JPanel();
GridLayout numPadLayout = new GridLayout(4,3);
//numpad Gaps
numPadLayout.setVgap(4);
numPadLayout.setHgap(4);
numPadContainer.setLayout(numPadLayout);

//creating and adding numpad buttons 1-9
for (int i = 1; i <= 9; i++){
   String label = Integer.toString(i);
   JButton numPadButton = new JButton(label);
   numPadContainer.add(numPadButton);
}
//creating & Add 0 and spacer for Numpad layout
JPanel spacer1 = new JPanel();
JButton numPad0 = new JButton("0");
numPadContainer.add(spacer1);
numPadContainer.add(numPad0);

//Content Adding to Main Frame-----------------------------------------------------------

//elements being added to Contents Frame
//spacers are "remoteContentsFrame.add(Box.createRigidArea(new Dimension(0, 20)))"

remoteContentsFrame.add(Box.createRigidArea(new Dimension(0, 20)));
//Power Button
remoteContentsFrame.add(powerButtonContainer);
remoteContentsFrame.add(Box.createRigidArea(new Dimension(0, 20)));
//Input Selector
remoteContentsFrame.add(inputButtonContainer);
remoteContentsFrame.add(Box.createRigidArea(new Dimension(0, 20)));
//Up and Down Buttons
remoteContentsFrame.add(channelButtonContainer);
remoteContentsFrame.add(Box.createRigidArea(new Dimension(0, 20)));
//volume Control
remoteContentsFrame.add(volumeButtonContainer);
remoteContentsFrame.add(Box.createRigidArea(new Dimension(0, 20)));
```

```
        //Selectors
        remoteContentsFrame.add(arrowButtonContainer);
        remoteContentsFrame.add(Box.createRigidArea(new Dimension(0, 20)));
        //NumberPad
        remoteContentsFrame.add(numPadContainer);

        //Main Jframe adding & settings
        remoteFrame.add(remoteContentsFrame);
        remoteFrame.setTitle("Universal Remote");
        remoteFrame.setSize(360, 620);
        remoteFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        remoteFrame.setVisible(true);


    }//end main func
}//end main class
```

Assignment Testing

Describe how you tested this program to verify that it runs correctly. Assignment Evaluation
Briefly describe what you learned from this project. What things did you struggle with? What was easy? Give your opinions of the process, including what you liked about the project and any suggestions you have for improving the project.

All testing was done by simply running program and visually checking against wireframe

Conclusion:

On concluding this project I thought it was a good introduction to building UIs with the java GUI. It gave me multiple opportunities to try out a lot of different frame layouts and input types. I actually ended up gearing my design to utilize as many input types as I could manage as opposed to strictly sticking to what might be considered a better UX design. For example the buttons around my navigation group probably should have been organized slightly differently to avoid missing intended buttons, however i want to try using a borderlayout instead of more of the same techniques i had already used.