Tabari Harvey, CSD 402, Module-11 Programming Assignment

I will talk about JavaFX BorderPane and JavaFX ScrollPane. To start this off, what is JavaFX? “JavaFX is a set of packages and APIs for developing programs with graphical user interfaces, 3D graphics, etc. A graphical user interface, or GUI, enables the user to interface with a program using graphical components, such as windows, buttons, text boxes, etc.” (Ferguson Nov 8, 2022)

When JavaFX was created it was to include, Model-View-Controller which is a design pattern that is built into three different components. All of the components are for different parts of the application. The model represents the data and the logic for the business, this is meant to show a small framework of what it is designed to be, it will also respond to what the controller is requesting. The controller is the connection between the model and view, all it does is tell the model what to do before it goes to the view. Finally, the view is where all of the logic from the application goes. This is where all of the data from the model, pushed through the controller all for the user to see the final product. The model format of MVC is of similar nature to SQL, with the items and viewers but it specifies the components more. “BorderPane is a part of JavaFX. BorderPane class lays its children in top, bottom, center, left and, right positions. BorderPane lays out each child set in the five positions regardless of the child’s visible property value unmanaged children are ignored. BorderPane class inherits Pane class.” (https://www.geeksforgeeks.org/javafx-borderpane-class/) This is setting up the border for the UI to be well-structured. When you’re designing your website or application it helps to have a good interface for the users to adapt well towards it. BorderPane could be used to make a menu bar, a status bar, a section for main content, or it could be for navigation. BorderPane has 12 different methods that help form the border. It starts with, getAlignment(Node C) and it returns the alignment of the node, setAlignment(Node c, Pos v) sets the alignment of the node c to v. These following methods are pretty self-explanatory. GetBottom(), getCenter(), getLeft(), getRight(), getTop(), setBottom(), setCenter(), setLeft(), setRight(), finally setTop(). When you use BorderPane, the thought of different devices needs to be in the forefront. There are constraints that can be applied when it is set on the child. Alignment is the constraint, javafx.geometry.Pos is the position, and the description is, the alignment of the child within its area of the border pane. The second constraint is margin, and the type is javafx.geometry.Insets. The description is the margin space around the outside of the child. JavaScrollPane, “Is a control that provides a scrollable viewport of its contents. It allows the user to scroll the content vertically or horizontally by using scroll bars. It is used to display a component that is large or one whose size can change dynamically when the screen viewport is limited.” (<https://www.tutorialspoint.com/javafx/javafx_scrollpane.htm>) When you go through an application or a website, the page can be pretty big but, it is determined on what device you are looking at it through. Typically on a PC it will be stretched vertically and, in order to see more going to the top or bottom there is a side bar. This side bar can be clicked and moved up or down to specifically see more of the page. Following my comments on the application adapting to the screen type, this happens through the values getting mapped into the bounds that are contained in the node. To configure the policies of the scrollbar to fit whatever screen you would use on all surfaces is (java.io.Serializable, java.lang.Comparable<ScrollPane.ScrollBarPolicy>). The constructors that apply to ScrollPane are ScrollPane(), which makes a new ScrollPane. ScrollPane(Node) which creates a new ScrollPane as well. There are 17 methods that go with ScrollPane and they are contentProperty(), createDefaultSkin(), fitToHeightProperty(), fitToWidthProperty(), getContent(), getClassCssMetaData(), getControlCssMetaData(), getHbarPolicy(), getHmax(), getHmin(), getHvalue(), getMinViewportHeight(), getMinViewportWidth(), getPrefViewportHeight(), getPrefViewportWidth(), getVbarPolicy(), and getViewportBounds(). There are four simple steps as to how the ScrollPane is initiated. Starts with creating a node to display the ScrollPane. The second step is inisitating the ScrollPane class, the third step is setting up the contents of the ScrollPane, and the fourth step is launching the application.

A code example for BorderPane is:

import javafx.application.Application;

import javafx.collections.ObservableList;

import javafx.scene.Scene;

import javafx.scene.control.TextField;

import javafx.scene.layout.BorderPane;

import javafx.stage.Stage;

public class BorderPaneExample extends Application {

@Override

public void start(Stage stage) {

BorderPane bPane = new BorderPane();

bPane.setTop(new TextField("Top"));

bPane.setBottom(new TextField("Bottom"));

bPane.setLeft(new TextField("Left"));

bPane.setRight(new TextField("Right"));

bPane.setCenter(new TextField("Center"));

Scene scene = new Scene(bPane);

stage.setTitle("BorderPane Example");

stage.setScene(scene);

stage.show();

}

public static void main(String args[]){ launch(args);

}

}

A code Example for ScrollPane is: import javax.swing.\*;  
import java.awt.\*;  
  
public class SimpleJScrollPaneExample {  
 public static void main(String[] args) {  
 JFrame frame = new JFrame("Simple JScrollPane Example");  
 frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);  
 frame.setSize(300, 200);  
  
 *// Create a JPanel to hold a list of labels*  
 JPanel panel = new JPanel();  
 panel.setLayout(new BoxLayout(panel, BoxLayout.Y\_AXIS));  
  
 *// Add some content to the panel*  
 for (int i = 0; i < 10; i++) {  
 panel.add(new JLabel("Label " + (i + 1)));  
 }  
  
 *// Wrap the panel in a JScrollPane*  
 JScrollPane scrollPane = new JScrollPane(panel);  
  
 *// Set the vertical and horizontal scroll bar policies*  
 scrollPane.setVerticalScrollBarPolicy(JScrollPane.VERTICAL\_SCROLLBAR\_ALWAYS);  
 scrollPane.setHorizontalScrollBarPolicy(JScrollPane.HORIZONTAL\_SCROLLBAR\_AS\_NEEDED);  
  
 *// Add the scrollPane to the frame*  
 frame.add(scrollPane);  
  
 *// Make the frame visible*  
 frame.setVisible(true);  
 }  
}

Works Cited

GeeksforGeeks. (2021, September 1). *JavaFX: Borderpane class*. https://www.geeksforgeeks.org/javafx-borderpane-class/

Ferguson, N. (2022, November 9). *JavaFX Basics*. Medium. https://medium.com/@teamtechsis/javafx-basics-8fea407f060e

Kaalel, & Follow. (2024, July 8). *MVC Framework introduction*. GeeksforGeeks. <https://www.geeksforgeeks.org/mvc-framework-introduction/>

Borderpane (javafx 8). (2015, February 10). https://docs.oracle.com/javase/8/javafx/api/javafx/scene/layout/BorderPane.html

Borderpane (javafx 8). (2015a, February 10). <https://docs.oracle.com/javase/8/javafx/api/javafx/scene/layout/BorderPane.html>

*JavaFX Scrollpane*. Tutorialspoint. (n.d.). https://www.tutorialspoint.com/javafx/javafx\_scrollpane.htm