```
package ui.layouts.GridPane;

import javafx.application.Application;
import javafx.event.ActionEvent;
import javafx.event.EventHandler;
import javafx.geometry.HPos;
import javafx.geometry.Pos;
import javafx.geometry.Rectangle2D;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.control.Label;
import javafx.scene.control.TextField;
import javafx.scene.layout.BorderPane;
import javafx.scene.layout.GridPane;
import javafx.scene.layout.HBox;
import javafx.scene.layout.VBox;
import javafx.stage.Screen;
import javafx.stage.Stage;

/**
 * Demo1. Entry point into demonstration application.
 */
public class Demo1 extends Application {

    private BorderPane layout;
    private Scene scene;
    private TextField txtFirstName, txtLastName;

    @Override
    public void start(Stage stage) {

        //Create BorderPane layout manager.
        layout = new BorderPane();  //This is the "root node".

        //Give Root Node a CSS ID Attribute
        layout.setId("appContainer");

        //Set Scene Properties.
        setSceneProperties();

        //Build Demo App Layout
        buildLeft();
        buildTop();
```

```
        //Set a few properties of our Application Window
        stage.setScene(scene);
        stage.setTitle("Grid Pane Demo");
        stage.show();
    }

    /**
     * main. Application Entry Point.
     * @param args
     */
    public static void main(String[] args) {
        launch();
    }

    /**
     * buildLeft. This method builds the Left Region of BorderPane.
     * This is VBox containing all buttons.
     */
    private void buildLeft() {

        BorderPane leftLayout = new BorderPane();

        // Create a faux border-right effect using a Label.
        Label divider = new Label();
        divider.setId("divider1");
        divider.setPrefWidth(1);
        divider.setMinHeight(Screen.getPrimary().getBounds().getHeight());
        leftLayout.setRight(divider);

        //Place all demonstration buttons in a Vercial Box.
        VBox buttonBox = new VBox();

        //Set Alignment of Buttons in VBox Container.
        buttonBox.setAlignment(Pos.TOP_CENTER);

        //Give VBox a CSS ID
        buttonBox.setId("buttonMenuContainer");

        //Create some vertical spacing b/n buttons
        buttonBox.setSpacing(10);

        //Add Demonstration Buttons
```

```
        Button btnExample1 = new Button();

        //Set Button Text
        btnExample1.setText("Example 1");

        //Set All Buttons to the same size.
        btnExample1.setMaxWidth(Double.MAX_VALUE);

        //Add Click Event.
        btnExample1.setOnAction(new EventHandler<ActionEvent>() {

            @Override
            public void handle(ActionEvent event) {
                System.out.println("Example 1 Button Clicked.");
                layout.setCenter(example1());
            }
        });

        //Create Button 2
        Button btnExample2 = new Button();
        btnExample2.setText("Useless Button");
        btnExample2.setMaxWidth(Double.MAX_VALUE);
        btnExample2.setOnAction(new EventHandler<ActionEvent>() {

            @Override
            public void handle(ActionEvent event) {
                System.out.println("Example 2 Button Clicked.");
            }
        });

        //Create Button 3
        Button btnExample3 = new Button();
        btnExample3.setText("Useless Button");
        btnExample3.setMaxWidth(Double.MAX_VALUE);
        btnExample3.setOnAction(new EventHandler<ActionEvent>() {

            @Override
            public void handle(ActionEvent event) {
                System.out.println("Example 3 Button Clicked.");
            }
        });

        buttonBox.getChildren().addAll(btnExample1, btnExample2, btnExample3);
```

```
    //Add VBox to leftLayout.
    leftLayout.setCenter(buttonBox);

    //Place into Application.
    layout.setLeft(leftLayout);


}

/**
 * buildTop. Create a Title Bar.
 */
private void buildTop() {

    BorderPane topLayout = new BorderPane();

    //Add CSS Style ID.
    topLayout.setId("topLayoutContainer");

    // Create a faux border-bottom effect using a Label.
    Label divider = new Label();
    divider.setId("divider2");
    divider.setMaxHeight(1);
    divider.setMinHeight(1);
    divider.setMinWidth(Screen.getPrimary().getBounds().getWidth());
    topLayout.setBottom(divider);

    //Create an HBox to hold title.
    //We use the HBox to set text alignment to LEFT, MIDDLE
    HBox titleBox = new HBox();
    titleBox.setAlignment(Pos.TOP_LEFT);
    titleBox.setId("titleBox");

    //Create title.
    Label title = new Label();
    title.setText("GridPane Demo");
    title.setId("appTitle");

    //Add Title label to titleBox
    titleBox.getChildren().add(title);

    //Add Title Box (with label) to topLayout
```

```
        topLayout.setCenter(titleBox);

        //Add topLayout (a BorderPane Manager) to App Layout.
        layout.setTop(topLayout);

    }

    /**
     * setSceneProperties. This method sets the app to almost full size. It
also
     * is where CSS style sheet is attached.
     */
    private void setSceneProperties()
    {
        //The percentage values are used as multipliers for screen
width/height.
        double percentageWidth = 0.98;
        double percentageHeight = 0.90;

        //Calculate the width / height of screen.
        Rectangle2D screenSize = Screen.getPrimary().getBounds();
        percentageWidth *= screenSize.getWidth();
        percentageHeight *= screenSize.getHeight();

        //Create a scene object. Pass in the layout and set
        //the dimensions to 98% of screen width & 90% screen height.
        this.scene = new Scene(layout, percentageWidth, percentageHeight);

        //Add CSS Style Sheet (located in same package as this class).
        String css =
this.getClass().getResource("Demo1.css").toExternalForm();
        scene.getStylesheets().add(css);

    }

    /**
     * example1. This method just creates a simple GridPane with 2
     * rows and 2 columns. This example demonstrates the use of
     * showing gridLines.
     * @return
     */
    private VBox example1() {
```

```
//Create a container to fill 100% space in Center Region of
//App BorderPane (layout).
VBox exContainer = new VBox();
exContainer.setId("exContainer");

//Create a new GridPane.
GridPane gridPane = new GridPane();

//Turn on GridLines so we can see what is going on.
gridPane.setGridLinesVisible(true);

//Give the GridPane an ID for CSS Styles.
gridPane.setId("gridPane_Example1");

//Add some spacing between each control.
//Comment the next 2 lines out to see what happens when this is
//not explicitly set. It will remove the padding you specified.
gridPane.setHgap(5);
gridPane.setVgap(5);

//Add a description of what we are doing to GridPane.
//This description starts in Row 0, Col 0 and spans
//2 columns and one row.
Label label = new Label("Turn on the grid lines to see results.");
gridPane.add(label, 0,0,2,1);

//Add A Label. The label starts in Col 0, Row 1 and does not
//span any columns or rows.
gridPane.add(new Label("First Name"), 0, 1);

//Add a TextField. The textfield starts in Col 1, Row 1 and
//does not span any columns or rows.
txtFirstName = new TextField();
txtFirstName.setId("txtFirstName");
gridPane.add(txtFirstName, 1,1);

//Add Last Name label in Col 0, Row 2
gridPane.add(new Label("Last Name"), 0,2);

//Add Last Name Text Field in Col 1, Row 2.
txtLastName = new TextField();
txtLastName.setId("txtLastName");
gridPane.add(txtLastName, 1,2);
```

```
        //Add a Submit Button.
        Button submitButton = new Button("Submit");
        submitButton.setOnAction(new EventHandler<ActionEvent>() {

            @Override
            public void handle(ActionEvent event) {
                System.out.printf("Submit Button Clicked. Hi there %s %s",
                        txtFirstName.getText(), txtLastName.getText());
            }
        });
        gridPane.add(submitButton, 1,3);

        //Align the Submit Button to Right.
        gridPane.setHalignment(submitButton, HPos.RIGHT);

        //Add GridPane to container.
        exContainer.getChildren().add(gridPane);

        //Return Container
        return exContainer;
    }
}
```