Amanda Wedergren

July 18, 2025

Module 11.2 Assignment

JavaFX Gridpane is a useful component in JavaFX that creates a layout that arranges child components into a flexible grid of columns and rows. (Jenkov) Since it is flexible, nodes can be placed within the grid wherever desired, able to span multiple rows and columns. Children may overlap freely within the rows and columns, their order defined by the order of the child list. There are also constraints that can be placed on the child’s placement within the grid. Since the Gridpane will automatically adjust the grid to accommodate the content that is provided, the total number of rows and columns do not need to be specified up front. It is also customizable, allowing for padding and borders that will affect the way the content is laid of within the Gridpane. Most commonly, JavaFX Gridpane is used to create responsive user interfaces. (Point) Dynamic adjustments of the grid happen based on the size of the components that are created. When using the Gridpane class in JavaFX, multiple properties are available which help with alignment of the pane. In order to use Gridpane, the application would need to set the constraints of the layout for the children and then add the children to the gridpane instance. Static setter methods are used to set the constraints on the children. (Oracle, 2025) Rows and columns will automatically be sized to fit their content. For example, a column will be wide enough to accommodate the widest child or tall enough to fit the tallest child. Constraints can also be added to explicitly control the size of the rows and/or columns in the Gridpane. These constraints can also be added by percentage of the available Gridpane space, further allowing customization to fit the needs of the application.

Examples of using JavaFX Gridpane would be for things like a login screen for a website account or an entry for a contest or signing up for promotional offers. These are all things that can be created using JavaFX’s Gridpane functionality. I am also including a program example code using JavaFX Gridpane to create a window that allows users to enter their name and email address to sign up for promotional offers from a website.

import javafx.application.Application;

import javafx.scene.Scene;

import javafx.scene.control.Button;

import javafx.scene.control.Label;

import javafx.scene.layout.GridPane;

import javafx.stage.Stage;

public class GridPaneExample extends Application {

    @Override

    public void start(Stage primaryStage) {

        GridPane grid = new GridPane();

        // Add some controls to the grid

        grid.add(new Label("Name:"), 0, 0);

        grid.add(new Button("Enter Name"), 1, 0);

        grid.add(new Label("Email:"), 0, 1);

        grid.add(new Button("Enter Email"), 1, 1);

        // Set some spacing

        grid.setHgap(10);

        grid.setVgap(10);

        Scene scene = new Scene(grid, 300, 150);

        primaryStage.setTitle("JavaFX GridPane Example");

        primaryStage.setScene(scene);

        primaryStage.show();

    }

    public static void main(String[] args) {

        launch(args);

    }

}

I also chose JavaFX Scrollpane. JavaFX Scrollpane is a type of control within Java that allows a scrollable view of contents. (Point) The user is able to scroll the content vertically or horizontally, using scroll bars. When a component is too large or can change sizes dynamically, Scrollpane is a helpful tool to be able to view the entire component when the viewport is otherwise limited. This functionality is very helpful to make sure that all parts of the element can be viewed, even when it is larger than the screen being used. When creating a Scrollpane, the setContent method would be used to define the node that is used for the content of the Scrollpane that is being created. Only one node is able to be specified. If you are wanting to create a scroll view that is able to be used with more than one component, then a layout container or the Group class would need to be utilized. Policies can also be set up for the Scrollpane to set the horizontal and vertical scroll bars or components can be resized within the Scrollpane to match the width or height. An example of using a Scrollpane in an application would be when an image is much larger than the screen, requiring scroll bars to be present on the vertical and horizontal sides in order to view the entire image. I have also included some example coding to show how to utilize the Scrollpane method with Java and a selected image.

import javafx.application.Application;

import javafx.scene.Scene;

import javafx.scene.control.ScrollPane;

import javafx.scene.image.Image;

import javafx.scene.image.ImageView;

import javafx.stage.Stage;

public class GridPaneExample extends Application {

    @Override

    public void start(Stage primaryStage) {

        // Load an image (replace with your image path)

        Image image = new Image("file:Family\_2019.jpg");

        ImageView imageView = new ImageView(image);

        imageView.setPreserveRatio(true);

        imageView.setFitWidth(600); // Optional: set max width

        // Create a ScrollPane for the image

        ScrollPane scrollPane = new ScrollPane(imageView);

        scrollPane.setPannable(true);

        scrollPane.setFitToWidth(true);

        scrollPane.setFitToHeight(true);

        Scene scene = new Scene(scrollPane, 600, 400);

        primaryStage.setTitle("JavaFX ScrollPane Image Example");

        primaryStage.setScene(scene);

        primaryStage.show();

    }

    public static void main(String[] args) {

        launch(args);

    }

}

Both of these components allow for further customization of the Java program code, adding increased usability and a better user experience of the website. Gridpane allows information and content to be organized in a user friendly way, while the Scrollpane component allows larger elements to still be utilized and enables smooth scrolling through the content of the site or application. These components can even be utilized together in different places in the site or application, to create a more user-friendly experience and accessible content.

References

GridPane (javafx 8). (2025, July 15). https://docs.oracle.com/javase/8/javafx/api/javafx/scene/layout/GridPane.html

Jenkov, J. (n.d.). *JavaFX GridPane*. Jenkov.com Tech & Media Labs - Resources for Developers, IT Architects and Technopreneurs. https://jenkov.com/tutorials/javafx/gridpane.html

Point, T. (n.d.-a). *JavaFX GridPane layout*. Tutorialspoint. https://www.tutorialspoint.com/javafx/layout\_gridpane.htm

Point, T. (n.d.-b). *JavaFX Scrollpane*. Tutorialspoint. https://www.tutorialspoint.com/javafx/javafx\_scrollpane.htm