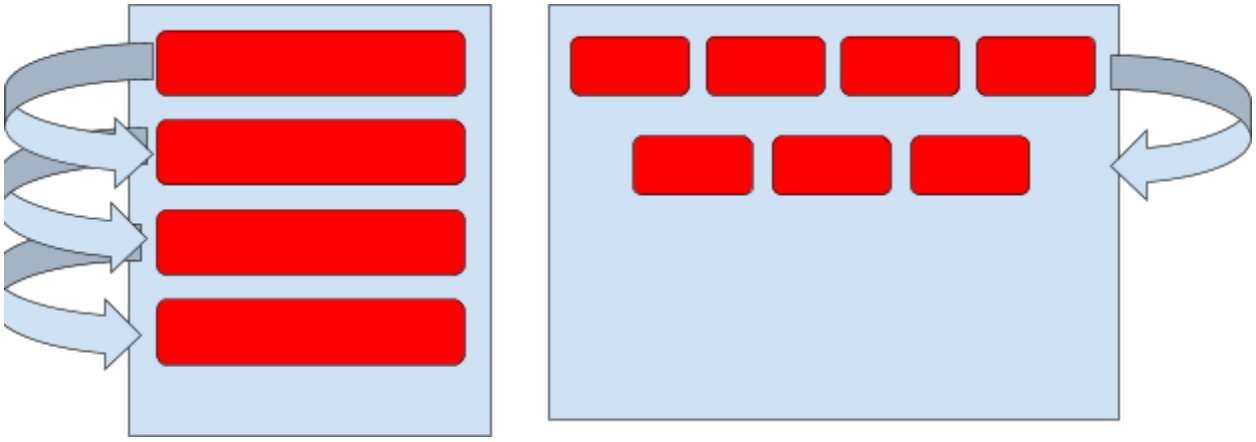
Jason Palmeri
Assignment 7
03/26/2022
CSD 405

JavaFX FlowPane/GridPane

When I started taking a look at the listed topics for JavaFX two of these really stood out to me, having come from a web development background aligning elements in a flexible way where you can resize the window, and keep the items in the view is very important to modern development. So the first topic that was interesting was the JavaFX FlowPane, and looking at it I immediately thought about CSS Flexbox. The ideology of flexboxes is that you have multiple items in a container, and each item takes up an even amount of space so that the items do not overflow off the screen. Geeksforgeeks explains JavaFX's FlowPane Class as a class that "Lays out its children in such a way that wraps at the flowpan'es boundary. A Horizontal flowpane will layout nodes in rows, wrapping at the flowpan's width. A vertical flowpane lays out nodes in columns, wrapping at the flowpane's height" (andrew1234).



In this example on the left you can see a vertical flowpane, where the container is the blue square, as we add more items (red rectangles) they flow from top to bottom, in a column pattern, and on the right we have the horizontal flowpane, where adding more items will make them flow to the next line in a row pattern, and these are based on height and width respectively.
Initiating a flowpane is as simple as creating a new instance of the class:
FlowPane myFlow = new FlowPane();
The FlowPane class contains a variety of setters and getters that allow you to edit how the FlowPane appears in your program. Geeksforgeeks has a table with some commonly used methods and what they do:

Method	Explanation
setAlignment(Pos v)	Sets the alignment of the pane
setHgap(double v)	Sets the horizontal gap of the pane

setOrientation(Orientation o)	Sets the orientation of the pane
setRowValignment(double v)	Sets the value of the rowValignment property
setVgap(double v)	Sets the vertical gap of the flow pane

Each of these methods has a getter version as well, and if you want to create an instance of the FlowPane with some of these values already set there is a plethora of constructors for the FlowPane class to instantly create something like a FlowPane with a Hgap and orientation. I would use FlowPane for things like lists, or showing a bunch of “Card” objects that will outscale the size of the window.

The other topic that looked similar to things I’ve seen in web development is the JavaFX GridPane, and similarly to the FlowPane it allows you to create a layout for the items in your application. GridPane can be simplified as being a table with 9 cells, and each cell can contain different items, like buttons and input fields.

Title		
	Input Fields	
	Submit	

In the above example you can see the 9 cells, with the first row containing a title, and 2 being empty white space. The next row contains a set of input fields with whitespace around to make the illusion of centering the inputs. And finally the submit button is bottom middle centered aligning on the right. JavaFX’s GridPane is similar to TilePane with the difference being GridPane allows you to change the size of each individual cell.Jenkov.com explains how scalable GridPane is by saying “The number of rows and columns in a GridPane depends on the components added to it. When you add a component to a GridPane you tell in what cell (row, column) the component should be inserted, and how many rows and columns the component should span” (Jenkov). Creating a GridPane is the same as creating any instance of a class:
GridPane myGrid = new GridPane();
And adding items to the grid can be done using the .add() method, which takes a few different fields:
myGrid.add(itemName, 0, 0, 1, 1);
The first field is the item you want to add to the list, so for here were adding the “Title” item. The second and third parameter is the column index and row index of the cell we want to display our item, so 0,0 is the top left cell. The fourth and fifth field sets the column span and row span, so we only want it to take up 1 row span and 1 column span. If we gave it a row span of 2, it would overflow into the next cell over, and be on top of the input fields cell like this:

	Title	
	Input Fields	
	Submit	

Notice that the top row has only 2 cells now, and the 0,0 cell is taking up 2 row spans. The Gridspan class is great for creating layouts like login windows, and generally anything that you want in a column/row scenario.

Citations

andrew1234. (2018, September 12). *JavaFX: Flowpane class*. GeeksforGeeks. Retrieved March 26, 2022, from https://www.geeksforgeeks.org/javafx-flowpane-class/

Jenkov, J. (2020, April 29). *JavaFX GridPane*. Tutorials. Retrieved March 26, 2022, from http://tutorials.jenkov.com/javafx/gridpane.html#:~:text=A%20JavaFX%20GridPane%20is%20a,will%20have%20the%20same%20width

TutorialsPoint. (n.d.). *JavaFX - FlowPane*. JavaFX - Flowpane. Retrieved March 26, 2022, from https://www.tutorialspoint.com/javafx/layout_flowpane.htm

TutorialsPoint. (n.d.). *JavaFX - Layout GridPane*. JavaFX - layout GridPane. Retrieved March 26, 2022, from https://www.tutorialspoint.com/javafx/layout_gridpane.htm