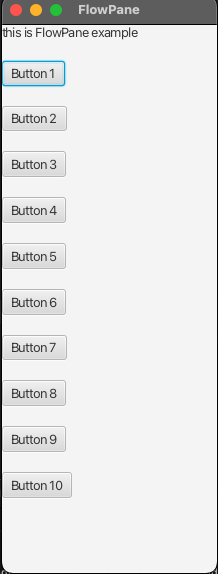
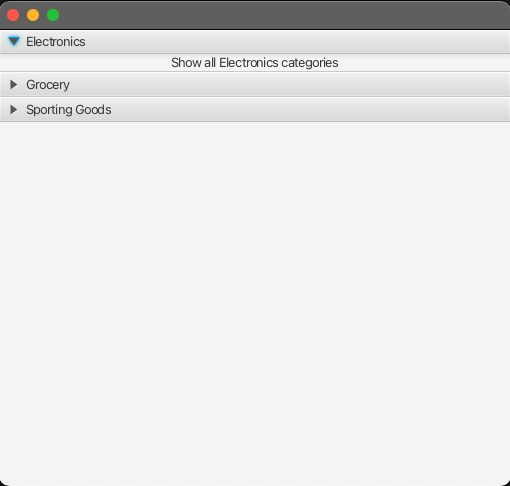
The JavaFX FlowPlane is a layout that allows wrapping along either the horizontal or vertical axis. According to geeks for geeks “Flowpane lays out its children in such a way that wraps at the flowpane’s boundary. A horizontal flowpane (the default) will layout nodes in rows, wrapping at the flowpane’s width. A vertical flowpane lays out nodes in columns, wrapping at the flowpane’s height.” This means that depending on whether you use the horizontal or vertical layout child objects on the page will either align in columns or rows and as the window gets smaller the elements will wrap around the chosen layout. This type of responsive design is crucial for applications that will be used on multiple screen sizes. Using something like this for a design paradigm allows an interface to be usable regardless of whether you are on a phone or a desktop by wrapping the elements along the scaling axis. The default orientation is horizontal wrapping, which makes a lot of sense because the typical space limitation on a mobile device is along the horizontal axis since people typically interact with their phones in landscape. This setting is controlled through the FlowPane Orientation property which can either be set inside of the constructor or with the setOrientation() method. With this if you created several buttons for the scene they would align themselves side by side or in a row. If there are too many buttons for the window size or if the window were to be resized along the x axis (made skinnier) the buttons would start to wrap around into another row below. In the case of a vertical FlowPlane if there were several buttons they would line up in a single column until they reached the bottom of the window at which point they would wrap around to start another column. In both examples this happens dynamically so adjusting the size of the window would alter the layout on the screen.



By default in a FlowPlane the child elements will be placed right next to each other with no spacing in between. To make these views more attractive the FlowPlane layout also includes methods for setting the horizontal and vertical gap between items. The gaps can be set either in the constructor or through the setHgap() or setVgap() methods.

The JavaFX accordion is a node structure which allows you to create lists that expand and contract much like the accordion instrument does in real life. According to Jakob Jenkov "The JavaFX Accordion control is a container control which can contain several sections internally, each of which can have their content expanded or collapsed.” The accordion class is great for designing hierarchical interfaces where there is a lot of information and it may be overwhelming to display all of the information at once. With the accordion class you can build a hierarchical list where you put the categories on a single level and can then expand those to see information below it. You see this design frequently on the web. Everytime you go to an online store if you browse for items it likely has an accordion style layout where you can select departments such as electronics which will then expand out to show you several options below that like tv’s, computers, video games, and so on. One interesting behavior that I found when researching the accordion layout is that you can only have one accordion expanded at a time. So if you are looking for an interface that lets you expand out multiple nodes simultaneously so according to Harold K on a stack exchange thread “The Accordion has an expandedPane property, that is a single TitledPane. There is no way for an Accordion to have multiple expanded panes. Instead, you can use multiple TitledPanes directly (inside a VBox or similar), to get the behavior you want. Unfortunately, this won't look just like an Accordion, because TitledPanes by default uses different styling. But with some custom CSS (look at caspian.css to see how accordions are styled) you could make it look just like the panes look in an Accordion.” So basically if you need multiple expanded areas you may be better served looking elsewhere. The accordion node is built out of Titled panes. The panes can be added through the getPanes().add() or getPanes.addAll() methods. It must then be added to the scene in order to be displayed.



**References**

Geeks for Geeks. (2018, September 12). *JavaFX | FlowPane Class*. geeks for geeks. Retrieved March 2, 2025, from https://www.geeksforgeeks.org/javafx-flowpane-class/

Jenkov, J. (2019, April 27). *JavaFX Accordion*. Jenkov.com. Retrieved March 1, 2025, from https://jenkov.com/tutorials/javafx/accordion.html

K, H. (2014, July 1). *How to expand multiple TitledPane of Accordion in javaFX?* Stack Overflow. Retrieved March 2, 2025, from https://stackoverflow.com/questions/24509804/how-to-expand-multiple-titledpane-of-accordion-in-javafx

TutorialsPoint. (n.d.). *JavaFX - Accordion*. TutorialsPoint. Retrieved March 2, 2025, from https://www.tutorialspoint.com/javafx/javafx\_accordion.htm