JavaFX HBox and JavaFX Accordion: Research Paper Module 11

My decision to research JavaFX HBox and JavaFX Accordion was based on my interest in them; I used VBox in my JavaFX program for this week, and wanted to learn about the differences between them, and Accordion looked interesting when I was looking into its function. They are both frameworks used in Java applications to help build user interfaces, and both contribute to the layout of the application’s format. These tools are not to build an application, but to help make the application visually appealing to the user. User interface functionality and design appeal can be easily overlooked, but the importance of it can affect things like user retention and accessibility.

JavaFX is used as a part of the scene.layout package. Its use is to provide a container that affects the horizontal layout. As you can imagine, VBox is a similar framework that affects the vertical layout of an application. HBox takes child nodes and arranges them into a horizontal row; essentially being able to list things side-by-side. This can be useful for setting up applications and creating a settings menu, search result pages, or anything else that may use large blocks of text or buttons that need to be lined up in a row.

One of the most notable features of HBox is spacing. HBox gives the flexibility to control the spaces between child nodes with ease, and this is one tool that helps give developers the freedom to make their application a design layout that they wish, or that is easily accessible by most people. HBox allows you to give a child node a specific alignment within the container, or you can also give your child nodes an overall padding if you’re not having to put in specific margins. This reflects the flexibility of this framework tool and shows how useful it can be for developers.

In my HBox example, I created an HBox with two buttons inside of it. I used margins and spacing to show how the elements within the HBox can be manipulated easily. I like using this method of styling your application because it feels very similar to HTML/CSS. I think having that knowledge of that syntax has made understanding JavaFX elements a lot easier than I originally expected.

Unlike HBox, JAVAFX Accordion is used with scene.control package. This is because while it does affect the way the application looks; it also requires user input as well. Accordion takes the content specified and lists it into a stack on clickable panels that can be both expanded or collapsed (like an accordion, obviously). Accordion is very versatile, and you use TitledPane to name/define each section. One that is defined, you can choose what content to input for when the accordion is expanded and what it will display when you do so. It also doesn’t open all panels at the same time; only one will be expanded at a time. This is to help keep a clean and organized appearance.

In my example of Accordion, I created a short list of panels that all say, “Click here”, and when you do, a short message pops up. This is a very simple display of the functionality of JavaFX Accordion, but it can by styled and structured to any developer's liking, even for more complex applications as well. When creating an accordion, it’s styled with padding and margins to ensure that it’s placed where the developer wants it in the application.

A very common use for Accordion panels is with settings sections of applications. It’s a great way to condense and organize an area that has a lot of information that can be broken down into smaller sub-sections and removing the other information that is no longer necessary when you’re trying to navigate to a specific part of the application. It’s also seen a lot in FAQ sections, and sometimes menus of fast-food applications.

In conclusion, within JavaFX there are a lot of tools that will take time and practice to learn and master, but with a strong understanding of HTML/CSS which seems to have similar concepts, I think it will be easy for someone with that knowledge to pick up quickly. These tools are great for allowing developers to organize and layout their applications in a way that will appeal to users and cater to those with accessibility issues. I know that these few topics are scratching the surface of what we can do with them, and I’m excited to continue to learn more!

\* Source: <https://jenkov.com/tutorials/javafx/hbox.html>

\* Source: <https://www.youtube.com/watch?v=NYGHL8N6Kc8>

\* Source: <https://www.geeksforgeeks.org/javafx-hbox-class/>

\* Source: <https://jenkov.com/tutorials/javafx/accordion.html>

\* Source: <https://www.tutorialspoint.com/javafx/javafx_accordion.htm>

\* Source: <https://www.geeksforgeeks.org/javafx-titledpane-class/>