JavaFX is a java platform for developers to create graphical user interfaces, or GUI’s. The overall concept of how these interfaces are put together can be compared to a Broadway musical. There are many elements that make up the background of each scene, and each scene is then put on the stage at the right time for everything to appear the way it is supposed to. JavaFX is similar, elements are created and added to a scene, and then the scene is added to the stage to be displayed to the user. Within JavaFX are organizational controls that help with the overall layout of the scene, controlling the alignment, position, and sizing of each interface element. While there are numerous different layouts to choose from, the ones we will look at are HBox and GridPane.

The HBox layout is pretty straightforward, it is an option that arranges interface elements in a horizontal line. The HBox code allows developers to specify spacing between elements and the overall alignment of the elements being included. The HBox container is an ideal choice for creating simple interface structures, such as toolbars or headers, situations where elements need to be arranged on a horizontal line. “If the HBox has a border and/or padding set, then the contents will be layed out within those insets.” (GeeksForGeeks.com). An example of the HBox layout is included in the attached Java Application.

GridPane on the other hand, organizes the interface elements in a grid or matrix fashion, providing for a more complex and flexible layout option. Like a spreadsheet, the GridPane layout consists of multiple rows and columns, with the developer able to specify the size and behavior of the rows and columns, as well as the ability to control size and positioning of each individual element on the grid. GridPane also includes a option to make grid lines visible, which can help with creating the overall interface and debugging. GridPane layouts are an excellent way to create forms, such as a user login form, with each label and input held within the grid layout. An example of the GridPane Layout is also included in the attached Java Application.

Another important feature of GridPane and HBox is their flexibility, with the ability to automatically adjust element size and positioning. This flexibility is achieved by the developer using layout constraints, which are essentially a set of rules that the JavaFX uses to determine element size and positioning. These rules/constraints can be different for different rows/columns, providing more flexibility. Additionally, this automatic flexibility allows developers to create dynamic interface structures, interfaces that can adapt to different user inputs or various screen sizes.

It is also important to remember that as Java is inherently an object oriented language, this orientation extends to JavaFX as well. Each layout design, each element, scene, and stage are individual objects of classes. The built in classes within the Java and JavaFX library are used to create objects of the interface elements. It follows then that each of these objects are able to use the methods defined in their class. Increasing functionality, these layout designs such as HBox and GridPane, are all written as subclasses from the JavaFX Scene class, inheriting the ability to adjust sizes, positions, and even colors of the interface object the developer creates. (Oracle)

In conclusion, JavaFX includes a wide variety of layout options for a developer to choose from. Each layout available has differing features, functionality, and purposes. The HBox layout keep each interface element in a horizontal line, perfect for toolbars. The GridPane layout hold each interface element in a grid pattern, with columns and rows. GridPane provides a more flexible layout. The layout designs available allow JavaFX developers to create applications that are user-friendly with great flexibility and powerful features.

# References

GeeksForGeeks.com. (n.d.). *JavaFX - HBox*. Retrieved from GeeksForGeeks.com: https://www.geeksforgeeks.org/javafx-hbox-class/

Oracle. (n.d.). *Oracle*. Retrieved from Oracle - JavaFX - HBox: https://docs.oracle.com/javafx/2/api/javafx/scene/layout/HBox.html