**Julio Pochet Edmead**

**Module 11 – Programming Assignment**

**JavaFX Layouts: VBox and GridPane**

JavaFX has become a go-to framework for building user interfaces in modern Java applications. While older tools like AWT and Swing laid the foundation, JavaFX offers cleaner syntax, better graphics, and a more modern development experience. In this paper, I’m focusing on two of the most common layout managers in JavaFX: ‘VBox’ and ‘GridPane’. I’ll explain how each one works, show code examples, and share why they’re both useful depending on what kind of UI you're building.

**VBox – Vertical Simplicity**

`VBox` stands for “Vertical Box” and that’s exactly what it does—it stacks items from top to bottom. If you’re building a layout where one component needs to sit right below another (like a label, followed by a button), `VBox` is a perfect fit. It saves time compared to manually positioning elements and keeps your code clean.

In my example, I created a `VBox` with three components: a label and two buttons. I added a spacing of 10 pixels, so they don’t look crammed together. Here's the line that adds them:

```java

VBox vbox = new VBox(10);

vbox.getChildren().addAll(label, btn1, btn2);

```

This is great for quick interfaces like login forms or menu screens. You can also align the entire column of content to the left, right, or center, which gives you flexibility without overcomplicating the layout.

According to Oracle’s documentation (2024), VBox is one of the simplest ways to build vertically aligned GUIs, especially when you don’t need multiple rows or columns.

**GridPane – Rows and Columns Done Right**

When vertical stacking isn’t enough, and you need more structure, like in a form or calculator, `GridPane` is the tool for the job. This layout lets you place elements inside a grid by specifying which row and column they belong to. It works like a table and is ideal when you need side-by-side alignment.

In my `GridPane` example, I used two labels and two text fields for “Name” and “Age”, followed by a submit button. Each label is placed in column 0, and its matching text field is in column 1. For example:

```java

grid.add(nameLabel, 0, 0);

grid.add(nameField, 1, 0);

```

This gives a clean and aligned form layout. You can also control padding and spacing between rows and columns, which is super useful when you want your app to look neat and professional.

TutorialsPoint (n.d.) explains that `GridPane` is especially useful for arranging controls in a structured way, like forms or settings panels. It allows developers to define the layout clearly and update it easily without breaking alignment.

**When to Use Each One**

Both `VBox` and `GridPane` serve different purposes, and choosing the right one depends on what you’re building. If you need something quick and simple, like stacking buttons or showing a list of options, `VBox` is ideal. It keeps things clean and readable, and it’s easy for beginners to learn.

If you're working on a layout where labels need to line up with input fields (or you’re building something like a calculator), `GridPane` gives you the control you need. It’s a little more complex, but the structure is worth it when you need exact positioning.

I also found that combining these layouts gives the best results in many cases. For example, you can place a `GridPane` inside a `VBox` if you want a header label on top and a structured form underneath it.

**Conclusion**

JavaFX has made GUI design much simpler than older frameworks, and `VBox` and `GridPane` are two of its most important layout tools. `VBox` is great for vertical stacking and quick designs, while `GridPane` offers structure and precision. Both layouts are useful in different situations, and learning how to use each one effectively is a big step toward becoming a confident Java developer. These layouts make it easier to build responsive, clean-looking user interfaces without needing complex positioning logic.

Understanding when to use `VBox` versus `GridPane` comes down to the layout you’re aiming for. If you keep that in mind, JavaFX becomes a powerful and enjoyable framework to work with.

**References**

Oracle. (2024). **JavaFX Layouts – VBox and GridPane.** <https://docs.oracle.com/javase/8/javafx/layout-tutorial>

TutorialsPoint. (n.d.). **JavaFX - GridPane Layout.** <https://www.tutorialspoint.com/javafx/javafx_gridpane_layout.htm>