In JavaFX, there are a very wide range of tools that a person could use for creating a Graphical User Interface. One of the more basic – albeit incredibly important – programming tools is the HBox class. This class’s primary function is to lay out your page in neat, customizable horizontal columns. This is contrasted with VBox, which accomplishes the same task, but in the vertical rows. With these two tools, almost any page can be made. The children of the Hbox class can be of any data type, whether it is text, a number value, another element of JFX, or anything else that is required to make it function. While they are capable of tackling any GUI problem, they are not necessarily the most efficient option that is available, as they do come with their own various drawbacks. Firstly, is the processing power required to create a page entirely out of boxes. When you create a box, you need to individually add each child into the system. Doing that ten times is easy, but if you want to make a creative and visually appealing page, it is going to require much more computing power adding in the dozens or even hundreds of cells required. Another issue is that each box has to be evenly spaced out, or entirely touching. This hinders its usefulness as it relates to making more advanced and creative pages than a simple, minesweeper-esque application would be.

The FlowPlane is similar to the box commands in that it is also used to lay out simple children across the page, but it does so with a greater respect for the size of the scene as well as is more adaptable if an aspect of the scene is changed. The FlowPlane class works by wrapping it around the established frame of the pane as necessary. This allows the individual elements to not get crunhed smaller than is comfortably visible or expanded to be too big if any changes are made to the scene or stage’s sizes. FlowPanes are also much more efficient on a processing level, as you do not need to manually add the children in individual functions, but instead can simply add them as parameters when creating the FlowPane itself. Admittedly, that does weaken the flexibility and use cases of the tool, as the ability to added children after declaration allows for much more dynamic systems, even if it comes with some more processing overhead. It mostly just comes down to which system would be more beneficial for any given circumstance.

Both of these tools are incredibly useful building blocks for creating a Graphical User Interface using JavaFX, and because of their striking similarities in many way I felt that they would be good options for writing this paper. They both accomplish the same general task of displaying several cells of text, prompts, buttons etc. but they each complete that task in separate ways. The FlowPanes allow for a more natural, easy to read layout that wraps where it needs to and does not interfere with the readability of the page, all while having a very fast processing speed. On the other hand, the HBox and VBox systems allow for dynamic addition and subtraction of children to make the page much more active and lively, at the cost of having less flexibility in the way of placement on the page due to requiring even spacing and forcing all segments to be on the row or column. Alongside the higher processing overhead from the additional steps of building each individual box, the FlowPane seems to have many more uses than HBox, though that is not to say they are utterly useless.