Zachariah King
5/11/25
CSD 402
Assignment 11.2

JavaFX Topics Report

To start, JavaFX is a media-enhanced packet in the Java world that allows engineers to plan out and create deluxe applications and programs for clients that work well across multiple environments. Like Java, JavaFX has had multiple releases. It's basically another collection of classes and media productions that use regular Java code. It has all the key User Interface controls necessary to make a full-bodied application with stylish enhancements made possible by languages like CSS. This combined with JavaFX's potential to use audio/visual elements like graphics and animations really make JavaFX a key player in modernizing the portrayal of the information it covers.

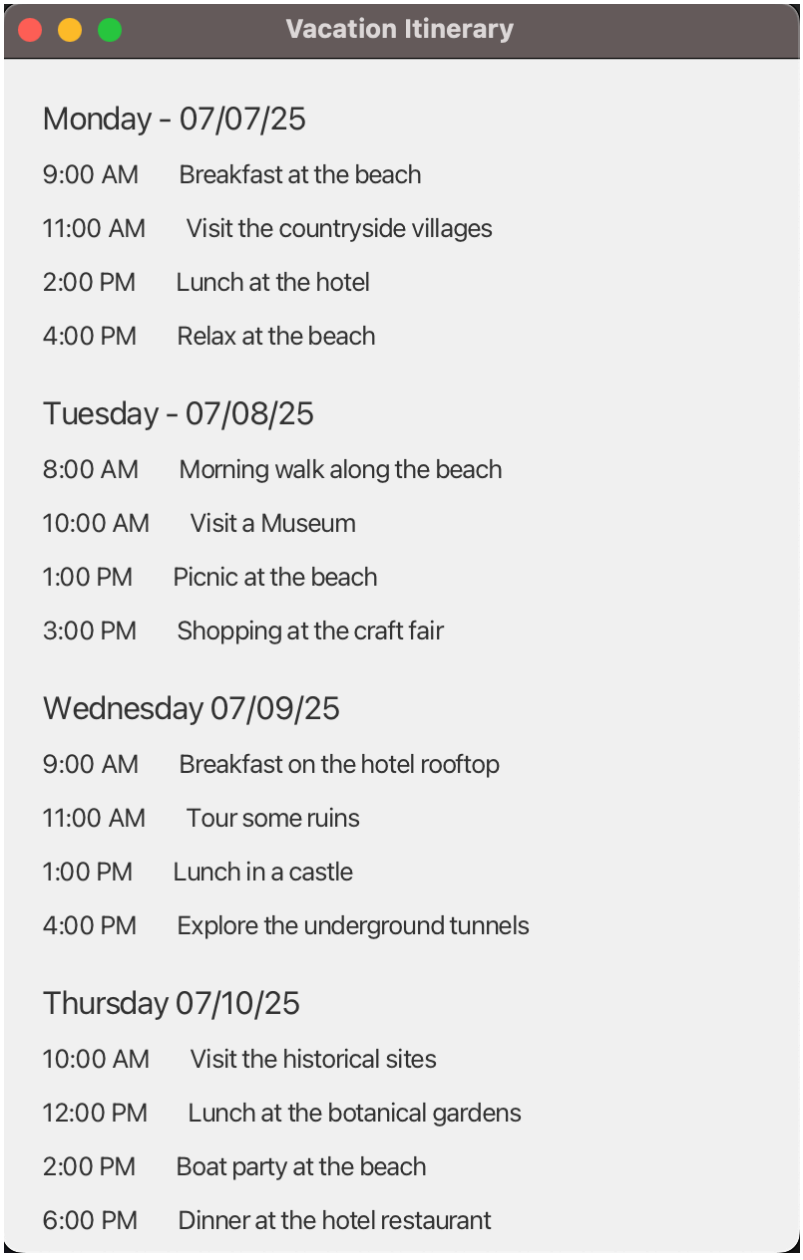
The first ability of JavaFX to discuss is the HBox. To put it simply, this setup tool is used to horizontally lay out elements or nodes in a single row, from left to right. It's like using a table to make sure things are set up side-by-side. Being a grid setup, it comes from the javafx.scene.layout package. Given how the elements are placed alongside each other in the HBox, they're good for organizing buttons, form labels, text boxes, and other elements. Using one comes with a simple process of initializing it and then setting its properties to clean it up and make it presentable. This means adding some spacing and padding values so that elements are not squished. Then comes the content by using methods like getChildren().add() or addAll(). More formatting comes from aligning the HBox appropriately by using setAlignment(). Although HBoxes have similarities with tables such as in HTML, this concept is Java's and more consistently upheld within different environments. These are widely used for toolbars, horizontal labels/input fields on forms, or graphics that sit next to each other such as on e-commerce sites. The HBox is direct and usually is not used alone. It could be accompanied by some of the other topics on the assignment list such as the VBox.

Speaking of the devil, the JavaFX VBox is the HBox's fraternal twin brother. They're fraternal because although they may be of the same DNA, they do not look alike. The Vbox is the Vertical Box in JavaFX. This one is also a container for a GUI that sets up its child elements in a single vertical column rather than a single horizontal row, like its twin. It comes from the same parent package, but it just prioritizes top-to-bottom depiction instead. Here, spacing is also very important so elements look nicely

organized, as well as the alignment and padding properties. It should be mentioned that if desired, there is a method that can be used to customize each separate child element's spacing properties. That static method is VBox.setMargin(Node child, Insets value) where you would replace child and value with the desired changes. VBoxes can also adapt or not adapt to its environment by its vgrow setting. If one wants their VBox to grow and fill its container, they must make sure the vgrow priority setting is set to ALWAYS. If it's set to NEVER, the VBox will not add any vertical space between its elements. Rather than lining up things side-by-side, VBoxes are for stacking things on top of each other, but just because the layout is different does not mean it's made for entirely different content. It still often is utilized for buttons, labels, text boxes, and other GUI pieces. Like its sibling, it is a flexible, irreplaceable arrangement organizer within the JavaFX library that's crucial for vertically aligning UI elements in a presentable and straightforward way.

The concept of combining these two JavaFX capabilities to create more complex and capable layouts is what makes JavaFX so powerful. For example, VBoxes and HBoxes can be used together to make layouts with elements organized both horizontally and vertically, where each layout is more appropriate. One could be creating an outline for an itinerary and need to use a parent VBox with VBox sections for each day of the trip with embedded HBoxes that contain a time slot and its planned activity. This is a semi-simpler way than using other JavaFX classes such as the other topics in the assignment instructions. An example of this can be found in my attached Java file and screenshot.

Zachariah King
5/11/25
CSD 402
Assignment 11.2



Zachariah King
5/11/25
CSD 402
Assignment 11.2

References

JavaNexus. (2024, September 16). *Mastering VBox and HBox: Common Layout Issues in JavaFX 20*. Java Tech Blog. https://javanexus.com/blog/mastering-vbox-hbox-layout-issues-javafx-20

Jenkov, J. (2020, February 5). *JavaFX HBox*. Jenkov.com Tech & Media Labs - Resources for Developers, IT Architects and Technopreneurs. https://jenkov.com/tutorials/javafx/hbox.html

Jenkov, J. (2021, September 3). *JavaFX VBox*. Jenkov.com Tech & Media Labs - Resources for Developers, IT Architects and Technopreneurs. https://jenkov.com/tutorials/javafx/vbox.html