Discussion-3: Introduction to JavaFX

Discussion Topic:

What are the benefits of using a Java Layout Manager within an application?
Note that there are different ways to set a panel's layout manager.
Provide an example of one of the ways to set this.

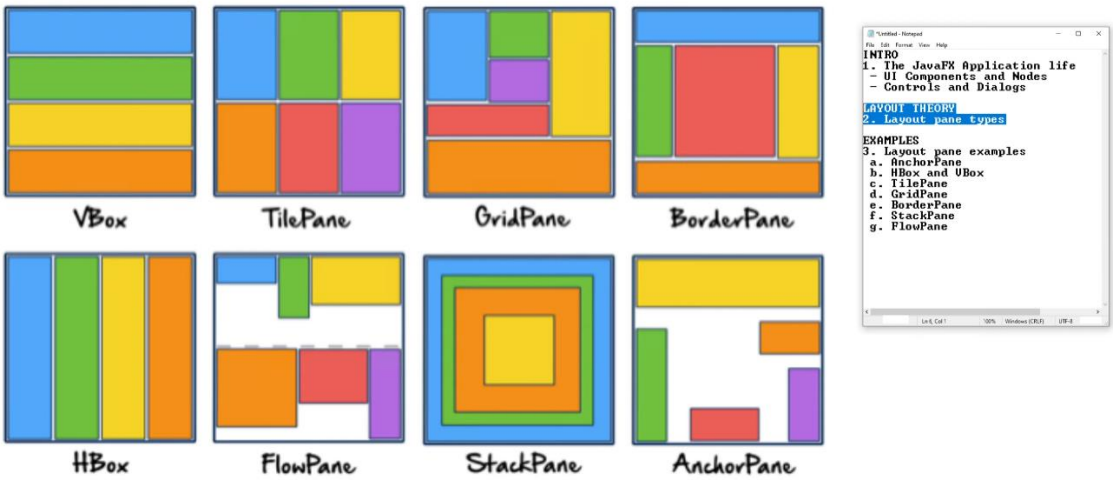
My Post:

Hello class,

First, I must point out that installing JavaFX is not an easy task.
The Java Layout Manager provides an easy way to develop graphical user interfaces (GUIs), particularly by offering tools to manage and organize GUI components. It is responsible for determining the dimensions and placement of components within a container (Oracle Docs. n.d.). While components can suggest their preferred sizes and alignments, the layout manager of the container ultimately decides the final size and position of these components. The Java Layout Manager provides a simpler approach to using panes (Gordon, 2013). It also facilitates the creation and management of standard layouts like rows, columns, stacks, tiles, and more. Additionally, when the window is resized, the layout pane automatically adjusts the positioning and size of its contained nodes based on their properties, it is responsive.

JavaFX offers a variety of layouts that can fit different GUI needs and functionalities. Layouts such as HBox, VBox, GridPane, BorderPane, StackPane, and FlowPane, see Figure 1.

Figure 1
Example of JavaFX Layouts

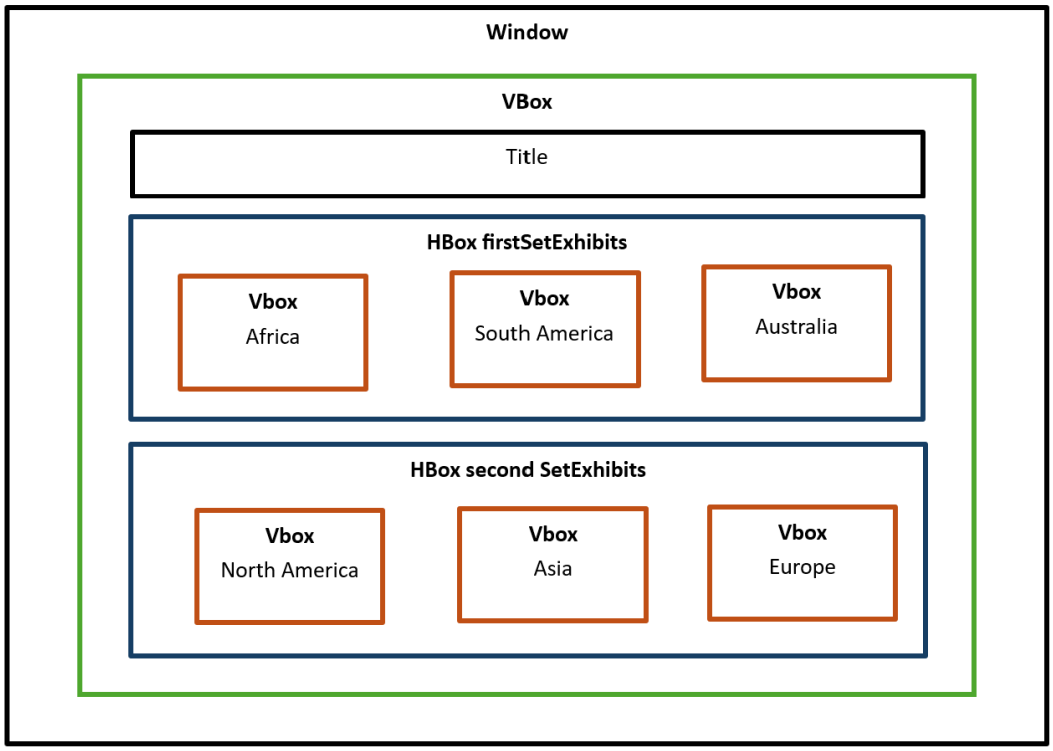


Note: from "3/10 - Introduction and overview of JavaFX panes or GUI containers for layout" by JavaHandsOnTeaching (2021)

The program below displays the animals found in various exhibits of a Zoo using the JavaFx VBox and HBox layouts, see Figure 2 to see how the different layout panes are positioned.

Figure 2

Zoo Layout Panes



Java File

```
import javafx.application.Application;
import javafx.geometry.Pos;
import javafx.scene.Scene;
import javafx.scene.control.Label;
import javafx.scene.layout.HBox;
import javafx.scene.layout.VBox;
import javafx.stage.Stage;

public class Main extends Application {
	@Override
	public void start(Stage primaryStage) {

		// Title for the window
		Label title = new Label("Zoo Exhibits");
		title.getStyleClass().add("title");

		// Create main VBox layout
		VBox mainLayout = new VBox(20);
		// Center align the contents of the VBox (title and Horizontal boxes for
the two
		// sets of exhibits)
		mainLayout.setAlignment(Pos.CENTER);
```

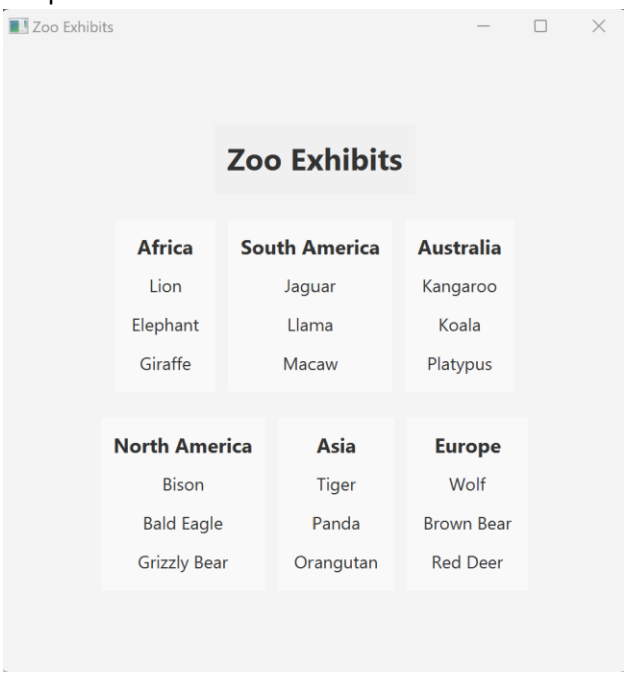
```
            // Horizontal boxes for the two sets of exhibits
            HBox firstSetExhibits = new HBox(10);
            firstSetExhibits.setAlignment(Pos.CENTER); // Center align the contents
of the HBox
            firstSetExhibits.getChildren().add(createExhibitSection("Africa",
"Lion", "Elephant", "Giraffe"));
            firstSetExhibits.getChildren().add(createExhibitSection("South America",
"Jaguar", "Llama", "Macaw"));
            firstSetExhibits.getChildren().add(createExhibitSection("Australia",
"Kangaroo", "Koala", "Platypus"));

            HBox secondSetExhibits = new HBox(10);
            secondSetExhibits.setAlignment(Pos.CENTER); // Center align the contents
of the Exhibit HBox
            secondSetExhibits.getChildren()
                        .add(createExhibitSection("North America", "Bison", "Bald
Eagle", "Grizzly Bear"));
            secondSetExhibits.getChildren().add(createExhibitSection("Asia",
"Tiger", "Panda", "Orangutan"));
            secondSetExhibits.getChildren().add(createExhibitSection("Europe",
"Wolf", "Brown Bear", "Red Deer"));

            // Add the title and horizontal sets to the main layout
            mainLayout.getChildren().addAll(title, firstSetExhibits,
secondSetExhibits);

            // Create a Scene
            Scene scene = new Scene(mainLayout, 500, 500);

            // Load the CSS file

      scene.getStylesheets().add(getClass().getResource("application.css").toExterna
lForm());

            // Set the scene on the primary stage
            primaryStage.setTitle("Zoo Exhibits");
            primaryStage.setScene(scene);
            primaryStage.show();
      }

      private VBox createExhibitSection(String continent, String... animals) { //
String... passes multiple string an

                                          // array of strings not a set size
            VBox exhibitSection = new VBox(5);
            exhibitSection.setAlignment(Pos.CENTER); // Center align the exhibit
section labels
            exhibitSection.getStyleClass().add("exhibit-section");

            // Title label for the continent
            Label continentLabel = new Label(continent);
            continentLabel.getStyleClass().add("continent-label");
            exhibitSection.getChildren().add(continentLabel);
```

```
            // Vertical box to hold animal labels
            VBox animalsBox = new VBox(5);
            animalsBox.setAlignment(Pos.CENTER); // Center align the animal labels
            for (String animal : animals) {
                  Label animalLabel = new Label(animal);
                  animalLabel.getStyleClass().add("animal-label");
                  animalsBox.getChildren().add(animalLabel);
            }

            // Add the VBox to the section
            exhibitSection.getChildren().add(animalsBox);

            return exhibitSection;
      }

      public static void main(String[] args) {
            launch(args);
      }
}
```

CSS file:

```
.title {
     -fx-font-size: 24px;
     -fx-font-weight: bold;
     -fx-padding: 10px;
     -fx-background-color: #f0f0f0;
}

.main-content-title {
     -fx-font-size: 20px;
     -fx-font-weight: bold;

     -fx-padding: 5px;
     -fx-background-color: #d0d0d0;
}

.exhibit-section {
     -fx-padding: 10px;
     -fx-background-color: #f9f9f9;
     -fx-border-radius: 5px;
}

.continent-label {
     -fx-font-size: 16px;
     -fx-font-weight: bold;
}

.animal-label {
     -fx-font-size: 14px;
     -fx-border-radius: 3px;
     -fx-padding: 3px;

}
```

Output:



-Alex

References:

Gordon, J. (2013, June). *JavaFX: Working with layouts in JavaFX* [PDF]. Oracle Docs. https://docs.oracle.com/javafx/2/layout/jfxpub-layout.pdf

JavaHandsOnTeaching (2021, June 19). *3/10 - Introduction and overview of JavaFX panes or GUI containers for layout* [Video]. YouTube. https://www.youtube.com/watch?v=GH-3YRAuHs0&t=905s

Oracle Docs. (n.d). *Using layout managers*. The Java™ Tutorials. Oracle. https://docs.oracle.com/javase%2Ftutorial%2Fuiswing%2F%2F/layout/using.html