**JAVAFX Readme**

Getting started with FX.

The start method is the heart of any JavaFX application. The method is called when the application is loaded, and a reference to the application's primary Stage is passed as a parameter. The start method creates the Scene object and displays it in the stage. Pictoral shown below.

@Override public void start(Stage primaryStage)

{

Group root = new Group();

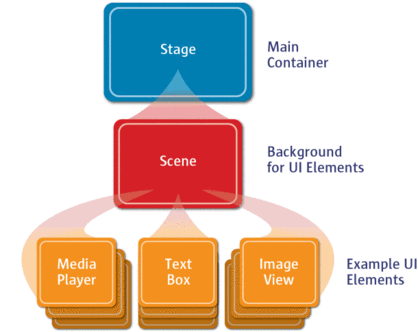
Scene scene = new Scene(root, 300, 250);

primaryStage.setScene(scene);

primaryStage.setTitle("The Click Me App");

primaryStage.show();

}



[Basic Layout panes](https://docs.oracle.com/javafx/2/layout/builtin_layouts.htm)

Layout panes in JavaFX let you manage the placement controls and shapes that appear in your scenes. Don't forget that a layout pane is itself a node, so you can nest layout panes within other layout panes to create complicated layouts.

**HBox**

HBox hbox1 = new HBox(10); // sets initial spacing

hbox1.getChildren().addAll(btn1, btn2, btn3);

hbox1.setAlignment(Pos.CENTER);

hbox1.setPadding(new Insets(10));

hbox1.setSpacing(10);

**VBox**

VBox vbox1 = new VBox(10); // sets initial spacing

vbox1.getChildren().addAll(btn1, btn2, btn3);

vbox1.setAlignment(Pos.CENTER);

vbox1.setPadding(new Insets(10));

vbox1.setSpacing(10);

**FlowPane**

FlowPane flow1 = new FlowPane(10, 10);

flow1.getChildren().addAll(btn1, btn2, btn3);

flow1.setAlignment(Pos.CENTER);

flow1.setOrientation(Orientation.VERTICAL);

flow1.setVgap(10);

flow1.setHgap(10);

flow1.setPrefWrapLength(5);

**BorderPane**

BorderPane border1 = new BorderPane();

border1.setCenter(btn1);

border1.setTop(btn2);

border1.setRight(btn3);

border1.setBottom(btn4);

border1.setLeft(btn5);

Basic Controls

JavaFX defines a plethora of controls that you use to interact with the user. The most *commonly* used are Label, Button, TextField, CheckBox, and RadioButton.

**Label**

Label lbl = new Label("This is a label");

**Button**

Button btn = new Button("Click Me!");

btn.setOnAction(e -> System.out.println("Click"));

**TextField**

TextField txtInput = new TextField("Prompt Text");

String input = txtInput.getText();

**CheckBox**

CheckBox chkOption = new CheckBox("Check Box Text");

if (chkOption.isSelected())

System.out.println("Option is selected");

**RadioButton (with ToggleGroup)**

RadioButton rdo1 = new RadioButton("Option 1");

RadioButton rdo2 = new RadioButton("Option 2");

rdo1.setSelected(true);

ToggleGroup options = new ToggleGroup();

options.getToggles().addAll(rdo1, rdo2);

if (rdo1.isSelected())

System.out.println("Option 1 is selected");

The Demo Files (breakdown). (**Java files with specific start() methods to trigger app are noted**)

> Switch (uses lambda functions to trigger events). **Trigger app from TwoScene.java file**.

<Includes programmatic CSS features>

> Login (jump from scene to scene upon a successful login). Includes Button UI event.

**Trigger LoginDemoApplication.java file**.

> PopUpDialog box (homebrewed!) Includes lambdas, programmatic FlowPane layout design

with HBox feature.

> JAVAFXTableView for rendering a “JTable” style output. Includes “customized” CSS file

(**styles.css**).  **Trigger DynamicTable.java file**.

> FXMenus Use of SceneBuilder and triggers for menu click events

Refs:

<http://www.oracle.com/technetwork/java/javase/overview/javafx-overview-2158620.html>

<http://docs.oracle.com/javase/8/javase-clienttechnologies.htm>

<https://docs.oracle.com/javafx/2/overview/jfxpub-overview.htm>

<http://www.javafxtutorials.com>

(Master-Detail view) <http://code.makery.ch/library/javafx-8-tutorial/>

Dialogs (aka popup boxes)!

<http://code.makery.ch/blog/javafx-dialogs-official/>

FX Chart City!!!

<http://docs.oracle.com/javase/8/javafx/user-interface-tutorial/charts.htm>

Totally cool text features

<https://docs.oracle.com/javase/8/javafx/user-interface-tutorial/text-settings.htm>

FX layouts

<https://docs.oracle.com/javafx/2/layout/builtin_layouts.htm#CHDGHCDG>

Simply CSS!

<http://www.guigarage.com/2016/02/javafx-and-css/>