

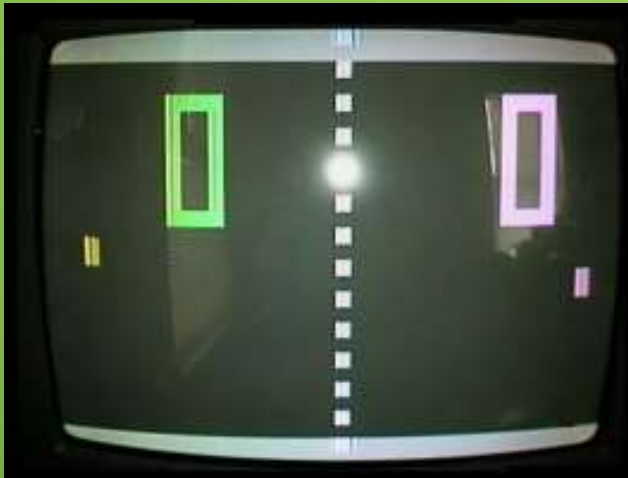


CSE 219 COMPUTER SCIENCE III

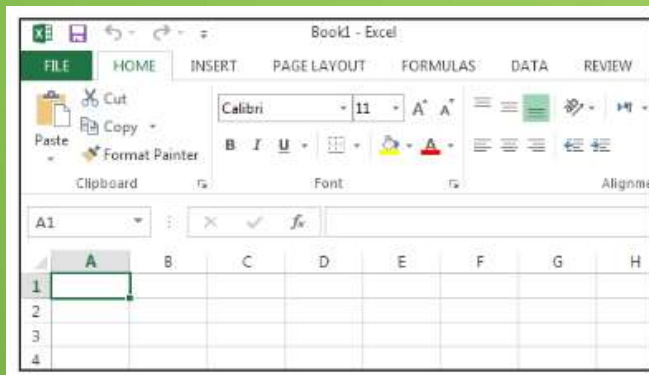
GRAPHICAL USER INTERFACES WITH JAVAFX

SLIDES COURTESY: PROF. RICHARD MCKENNA (SBU)

GUI EXAMPLES



MORE RECENT GUIs





GUI

Graphical User Interface (GUI)

- provides user-friendly human interaction

One typically uses frameworks for building GUIs

JavaFX (as of Java8, 2014)

- replaces Swing/AWT

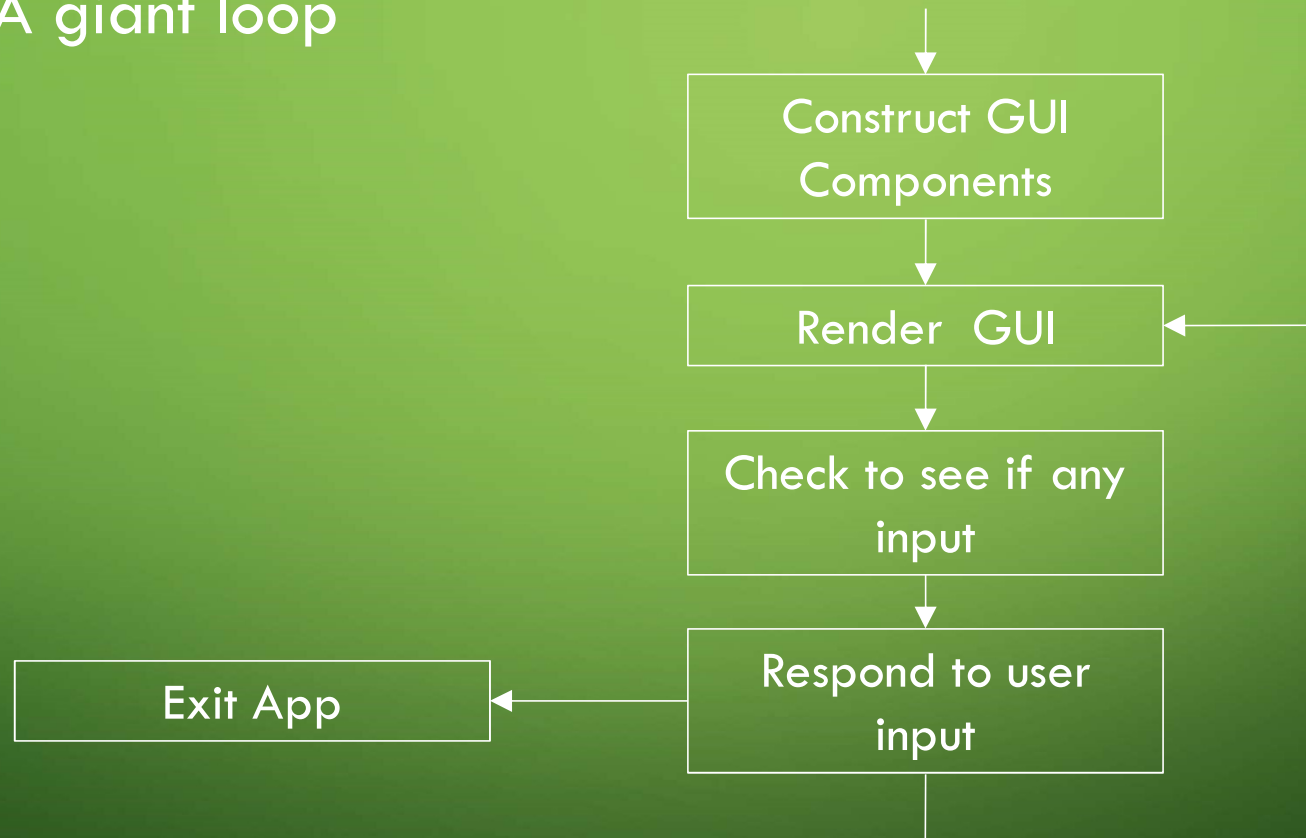
Why JavaFX better than Swing

- <https://blog.idrsolutions.com/2015/01/5-reasons-javafx-better-swing-developing-java-pdf-viewer/>

You'll spend much of your life learning new frameworks

HOW DO GUIs WORK?

A giant loop



EXAMPLE, A MOUSE CLICK ON A BUTTON

Operating System recognizes mouse click

- determines which window it was inside
- notifies that program

Program runs in loop

- checks input buffer filled by OS
- if it finds a mouse click:
 - determines which component in the program
 - if the click was on a relevant component
 - respond appropriately according to handler

GUI LOOK VS. BEHAVIOR

Look

- physical appearance
- custom component design
- containment
- layout management

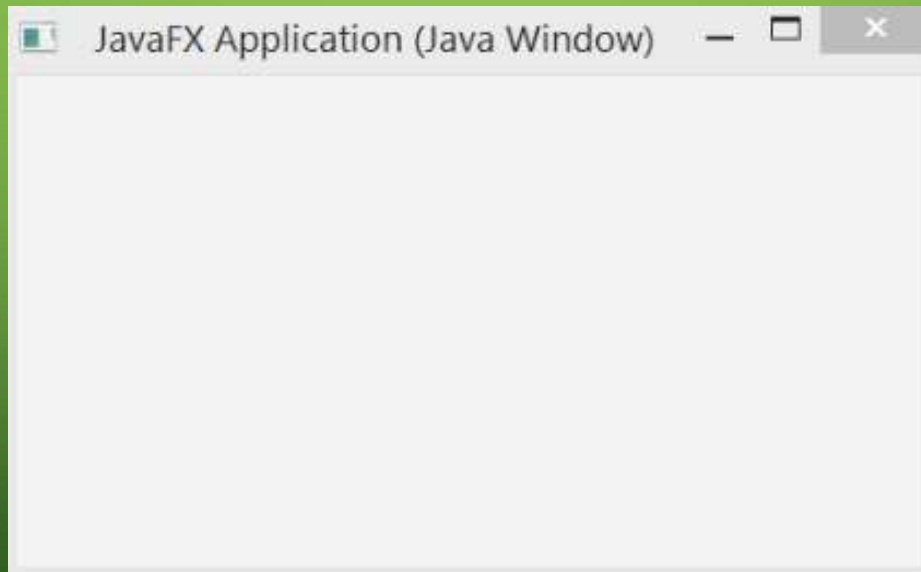
Behavior

- interactivity
- event programmed response

WHAT DOES A GUI FRAMEWORK DO FOR YOU?

Provides ready made visible, interactive, customizable components

- you wouldn't want to have to code your own window



PRIMARY JAVAFX CLASSES

javafx.application.Application

- entry point for JavaFX applications

javafx.stage.Stage

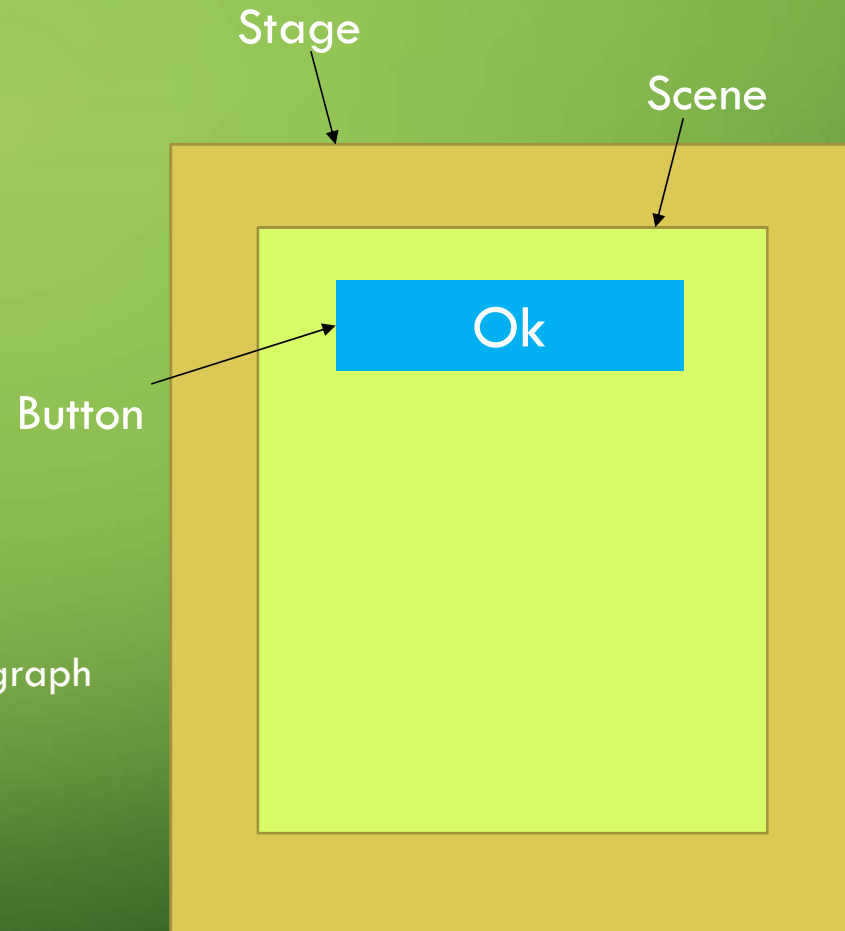
- top level JavaFX container

javafx.scene.Scene

- container for all content in a scene graph

javafx.scene.Node

- base class for scene graph nodes



IT ALL STARTS WITH APPLICATION

`javafx.application.Application`

When one gets created:

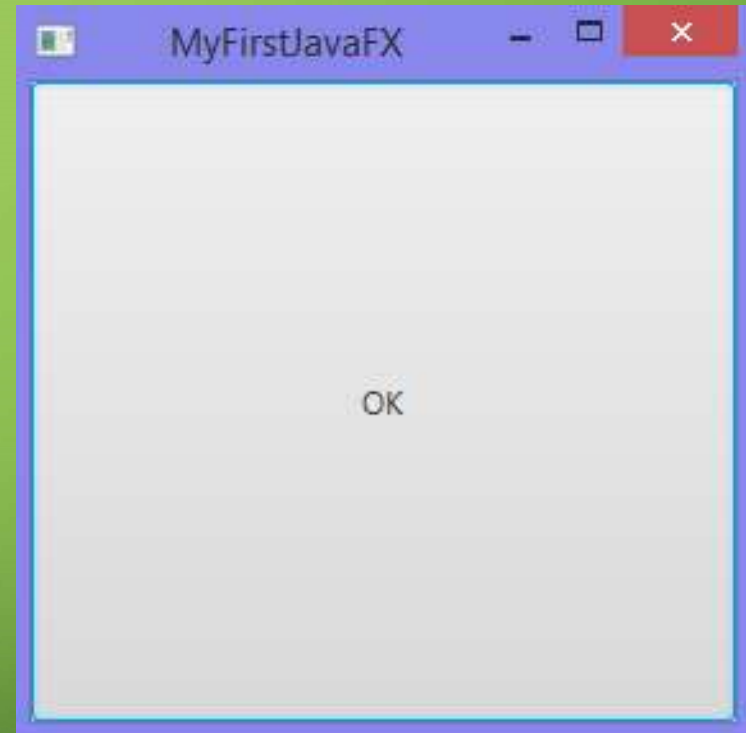
- JavaFX creates an application thread for running the application start method, processing input events, and running animation timelines.

So what do we do?

- Override App's start method!

OUR FIRST JAVAFX APP

```
public class MyFirstJavaFX extends Application {  
    @Override  
    public void start(Stage primaryStage) {  
        Button btOK = new Button("OK");  
        Scene scene = new Scene(btOK, 200, 250);  
        primaryStage.setTitle("MyFirstJavaFX");  
        primaryStage.setScene(scene);  
        primaryStage.show();  
    }  
    public static void main(String[] args) {  
        launch(args);  
    }  
}
```



LOTS OF INHERITED PROPERTIES

java.lang.Object

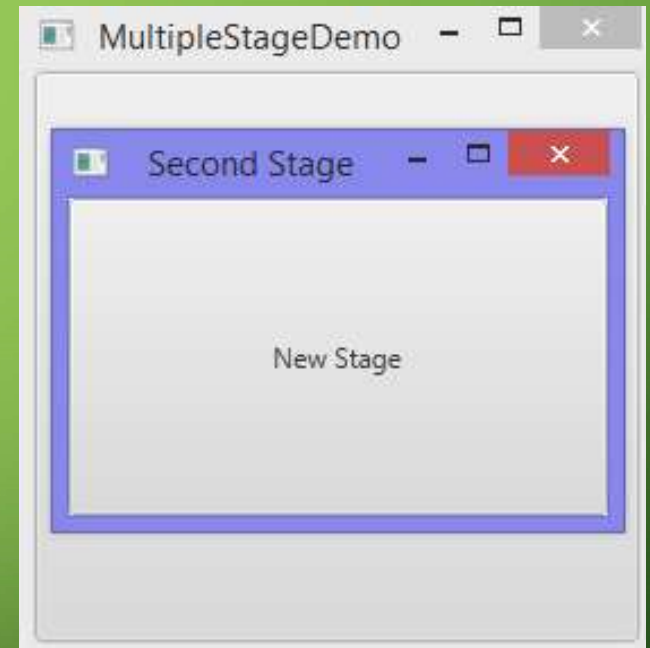
- javafx.stage.Window
- javafx.stage.Stage

java.lang.Object

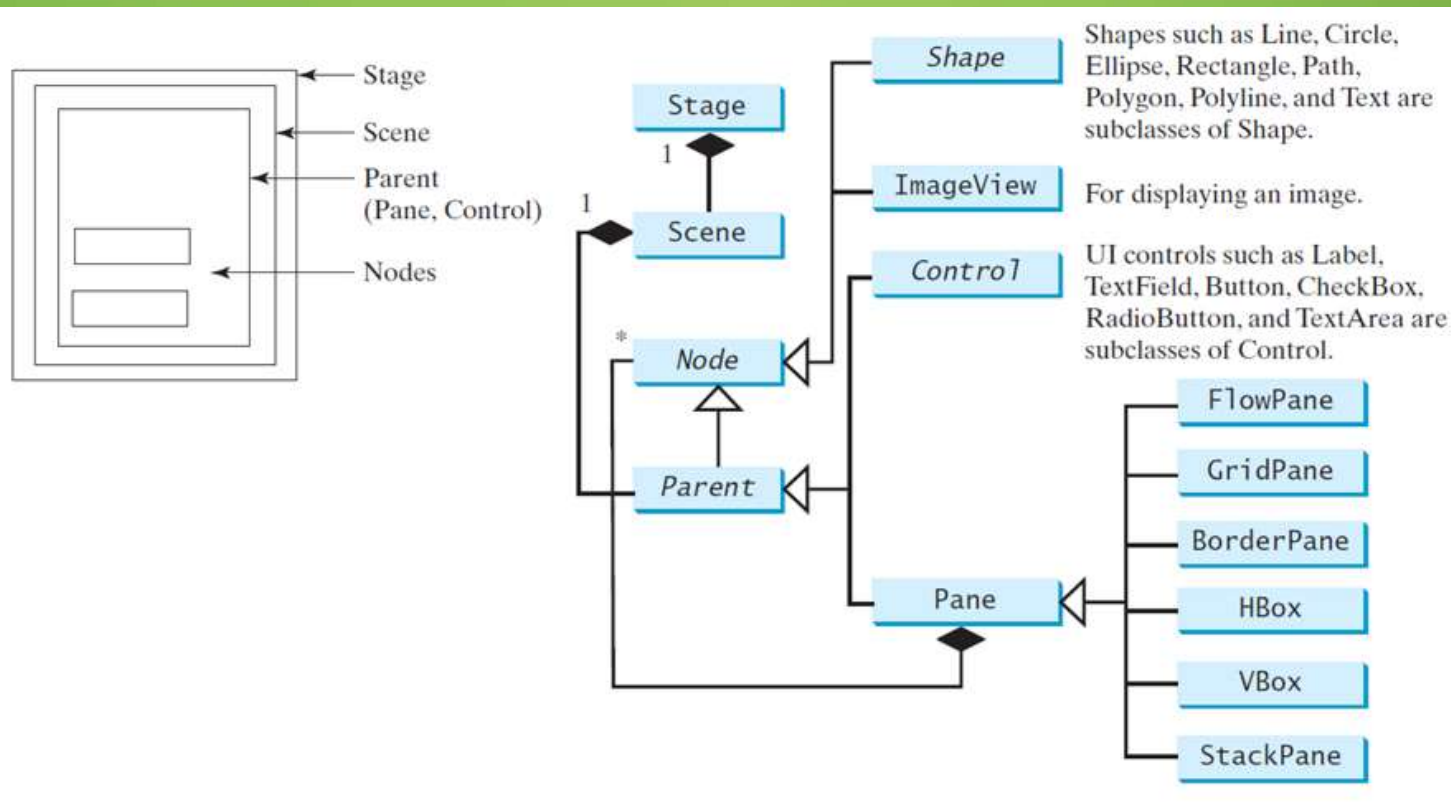
- javafx.scene.Node
- javafx.scene.Parent
- javafx.scene.layout.Region
- javafx.scene.layout.Pane

MULTISTAGE DEMO

```
public class MultipleStageDemo extends Application {  
    @Override  
    public void start(Stage primaryStage) {  
        Scene scene = new Scene(new Button("OK"), 200, 250);  
        primaryStage.setTitle("MultipleStageDemo");  
        primaryStage.setScene(scene);  
        primaryStage.show();  
  
        Stage stage = new Stage();  
        stage.setTitle("Second Stage");  
        stage.setScene(new Scene(new Button("New Stage"), 100, 100));  
        stage.show();  
    }  
}
```

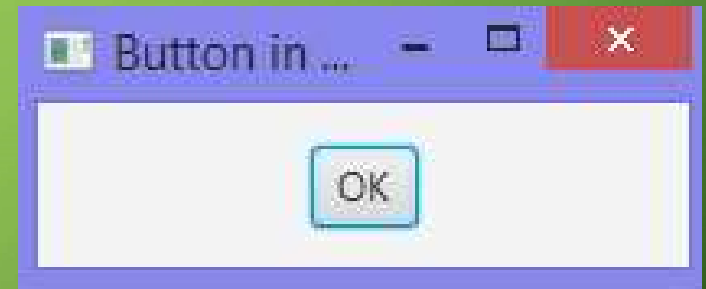


PANES, UI CONTROLS, AND SHAPES



A BUTTON IS A CONTROL

```
public class ButtonInPane extends Application {  
    @Override  
    public void start(Stage primaryStage) {  
        StackPane pane = new StackPane();  
        pane.getChildren().add(new Button("OK"));  
        Scene scene = new Scene(pane, 200, 50);  
        primaryStage.setTitle("Button in a pane");  
        primaryStage.setScene(scene);  
        primaryStage.show();  
    }  
    public static void main(String[] args) {  
        launch(args);  
    }  
}
```

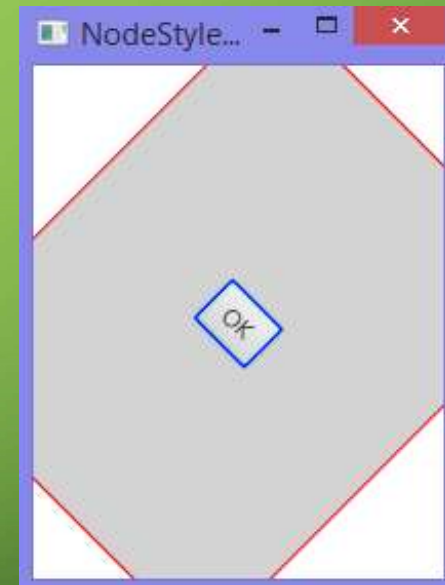


STACKPANE

`/* The StackPane layout pane places all of the nodes within a single stack with each new node added on top of the previous node. This layout model provides an easy way to overlay text on a shape or image and to overlap common shapes to create a complex shape. */`

```
public class NodeStyleRotateDemo extends Application {  
    @Override  
    public void start(Stage primaryStage) {  
        StackPane pane = new StackPane();  
        Button btOK = new Button("OK");  
        btOK.setStyle("-fx-border-color: blue;");  
        pane.getChildren().add(btOK);  
        pane.setRotate(45);  
        pane.setStyle("-fx-border-color: red;  
                      -fx-background-color: lightgray;");  
        Scene scene = new Scene(pane, 200, 250);  
        primaryStage.setTitle("NodeStyleRotateDemo");  
        primaryStage.setScene(scene);  
        primaryStage.show();  
    }  
}
```

CSS ROTATION!



LAYOUT PANES

JavaFX provides many types of panes for organizing nodes in a container.

<i>Class</i>	<i>Description</i>
Pane	base class for layout panes. it contains the <code>getChildren()</code> method for returning a list of nodes in the pane
StackPane	places the nodes on top of each other in the center of the pane
FlowPane	places the nodes row-by-row horizontally or column-by-column vertically
GridPane	places the nodes in the cells in a two-dimensional grid
BorderPane	places the nodes in the top, right, bottom, left, and center regions
HBox	places the nodes in a single row
VBox	places the nodes in a single column

FLOW PANE

```
public class ShowFlowPane extends Application {  
    @Override  
    public void start(Stage primaryStage) {  
        FlowPane pane = new FlowPane();  
        pane.setPadding(new Insets(11, 12, 13, 14));  
        pane.setHgap(5);  
        pane.setVgap(5);  
        pane.getChildren().addAll(new Label("First Name:"),  
                                   new TextField(),  
                                   new Label("MI:"),  
                                   new TextField(),  
                                   new Label("Last Name:"),  
                                   new TextField());  
        Scene scene = new Scene(pane, 210, 150);  
        primaryStage.setTitle("ShowFlowPane");  
        primaryStage.setScene(scene);  
        primaryStage.show();  
    }  
}
```




```
public class ShowGridPane extends Application {
```

```
    @Override
```

```
    public void start(Stage primaryStage) {
```

```
        GridPane pane = new GridPane();
```

```
        pane.setAlignment(Pos.CENTER);
```

```
        pane.setHgap(5.5);
```

```
        pane.setVgap(5.5);
```

```
        pane.add(new Label("First Name:"), 0, 0);
```

```
        pane.add(new TextField(), 1, 0);
```

```
        pane.add(new Label("MI:"), 0, 1);
```

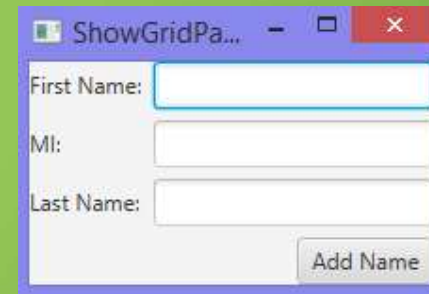
```
        pane.add(new TextField(), 1, 1);
```

```
        pane.add(new Label("Last Name:"), 0, 2);
```

```
        pane.add(new TextField(), 1, 2);
```

```
    }
```

GRID PANE



```
        Button btAdd = new Button("Add Name");
```

```
        pane.add(btAdd, 1, 3);
```

```
        GridPane.setHalignment(btAdd, HPos.RIGHT);
```

```
        Scene scene = new Scene(pane);
```

```
        primaryStage.setTitle("ShowGridPane");
```

```
        primaryStage.setScene(scene);
```

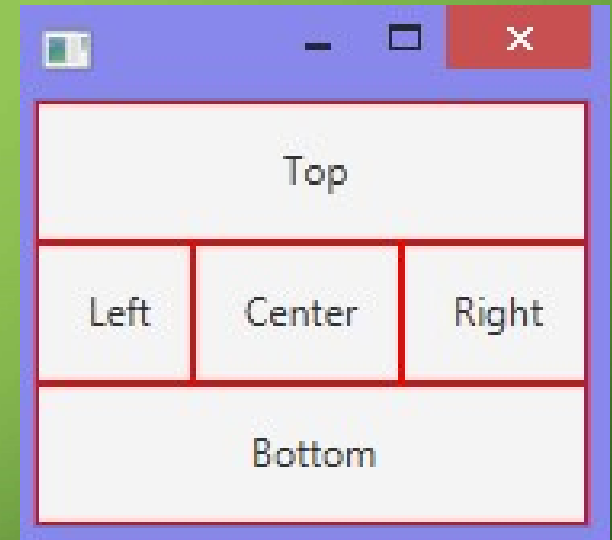
```
        primaryStage.show();
```

LET'S MAKE A CUSTOM PANE

```
class CustomPane extends StackPane {  
    public CustomPane(String title) {  
        getChildren().add(new Label(title));  
        setStyle("-fx-border-color: red");  
        setPadding(new Insets(11.5, 12.5, 13.5, 14.5));  
    }  
}
```

```
public class ShowBorderPane extends Application {  
    @Override  
    public void start(Stage primaryStage) {  
        BorderPane pane = new BorderPane();  
        pane.setTop(new CustomPane("Top"));  
        pane.setRight(new CustomPane("Right"));  
        pane.setBottom(new CustomPane("Bottom"));  
        pane.setLeft(new CustomPane("Left"));  
        pane.setCenter(new CustomPane("Center"));  
        Scene scene = new Scene(pane);  
        primaryStage.setScene(scene);  
        primaryStage.show();  
    }  
}
```

BORDERPANE



```
public class ShowVBox extends Application {  
    @Override  
    public void start(Stage primaryStage) {  
        VBox vBox = new VBox(15);  
        vBox.getChildren().add(new Label("Courses"));  
        Label[] courses = {new Label("CSE114"),  
            new Label("CSE214"), new Label("CSE219"), new Label("CSE308")  
        };  
        for (Label course: courses) {  
            vBox.getChildren().add(course);  
        }  
        Scene scene = new Scene(vBox);  
        primaryStage.setScene(scene);  
        primaryStage.show();  
    }  
}
```

VBOX




```
public class ShowHBox extends Application {
```

```
    @Override
```

```
    public void start(Stage primaryStage) {
```

```
        HBox hBox = new HBox(15);
```

```
        hBox.getChildren().add(new Label("Courses"));
```

```
        Label[] courses = {new Label("CSE114"),
```

```
            new Label("CSE214"),new Label("CSE219"),new Label("CSE308")
```

```
        };
```

```
        for (Label course: courses) {
```

```
            hBox.getChildren().add(course);
```

```
        }
```

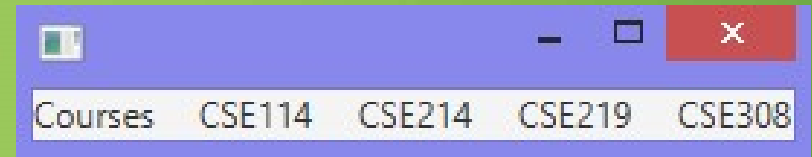
```
        Scene scene = new Scene(hBox);
```

```
        primaryStage.setScene(scene);
```

```
        primaryStage.show();
```

```
    }
```

HBOX

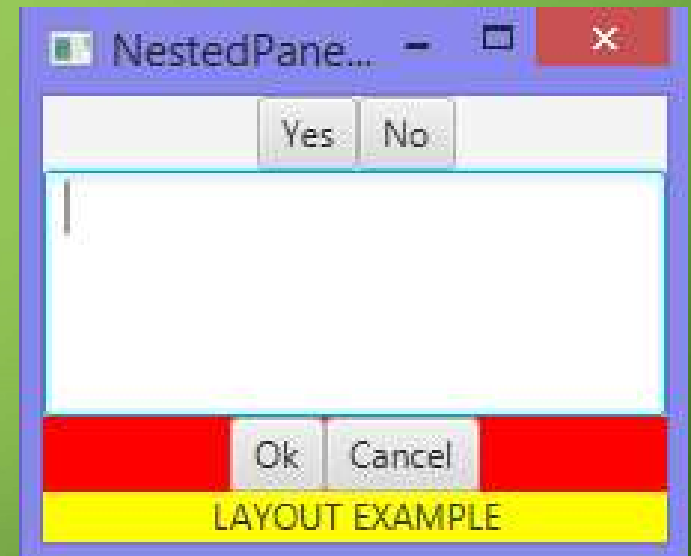


HOW ABOUT NESTED PANES?

...

```
FlowPane northPane = new FlowPane();  
northPane.setAlignment(Pos.CENTER);  
northPane.getChildren().add(new Button("Yes"));  
northPane.getChildren().add(new Button("No"));
```

...

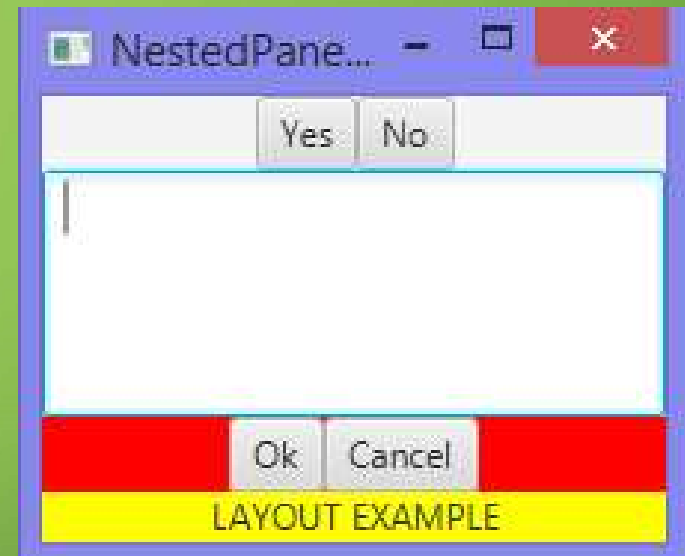


NESTED PANES

...

```
BorderPane southPane = new BorderPane();  
FlowPane northSouthPane = new FlowPane();  
northSouthPane.setAlignment(Pos.CENTER);  
northSouthPane.setStyle("-fx-background-color: red");  
northSouthPane.getChildren().add(new Button("Ok"));  
northSouthPane.getChildren().add(new Button("Cancel"));  
FlowPane southSouthPane = new FlowPane();  
southSouthPane.setAlignment(Pos.CENTER);  
southSouthPane.setStyle("-fx-background-color: yellow");  
southSouthPane.getChildren().add(new Label("LAYOUT EXAMPLE"));  
southPane.setTop(northSouthPane);  
southPane.setBottom(southSouthPane);
```

...



NESTED PANES

...

```
BorderPane pane = new BorderPane();  
pane.setTop(northPane);  
pane.setCenter(new TextArea());  
pane.setBottom(southPane);  
Scene scene = new Scene(pane, 210, 150);
```

...

