

JavaFX 1

Introduction to JavaFX

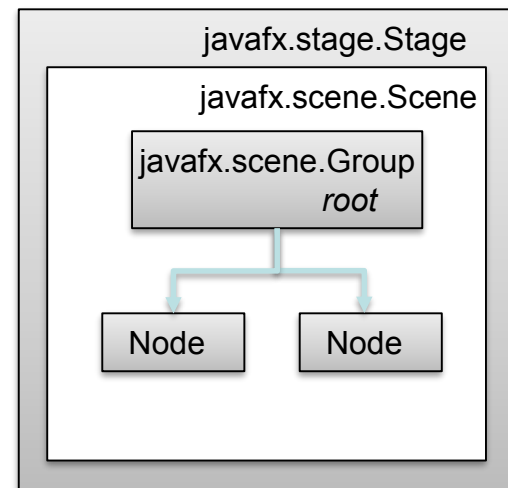
X1

JavaFX

- Designed for rich client applications
 - Graphics, UI's, video, audio, etc.
- Java APIs
 - *Not* to be confused with JavaFX 1.x, which is a scripting language, not a Java API
 - Java 8-10, integrated with JDK
 - Java 11-15, JavaFX is separate, and must be installed separately
- Replaces Swing
- JavaFX HelloWorld

JavaFX

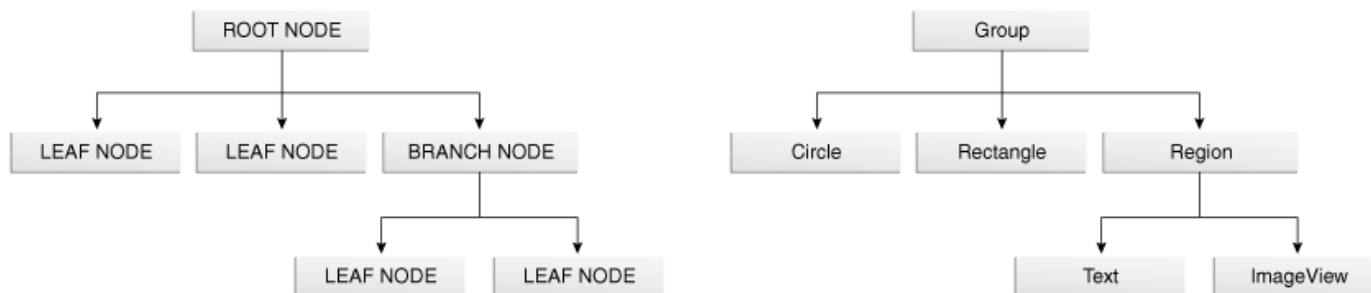
- **Extend** `javafx.application.Application`
 - Override the `start()` method
- Stage: the window
- Scene: container for a scene graph
- Node: object or group of objects in scene
- Pane: organizer of nodes in scene graph:
`FlowPane`, `TilePane`, `GridPane`, `HBox`, `VBox`, **etc.**



Java FX Scene Graph

Tree of nodes, with a single 'branch' at the root

- branch (may have children e.g. Group, Region)
- leaf (may not have children e.g. Rectangle, Circle)



Copyright Oracle (<http://docs.oracle.com/javafx/2/scenegraph/jfxpub-scenegraph.htm>)

Nodes and Properties

Can set node properties programmatically:

```
Text message = new Text("Hello");  
message.setFont(Font.font("Tahoma",  
    FontWeight.NORMAL, 40));  
message.setFill(Color.RED);
```

or declaratively using FXML / CSS:

```
#text {  
    -fx-font-family: Tahoma, sans-serif;  
    -fx-font-style: normal;  
    -fx-font-size: 40;  
    -fx-fill: red;  
}
```

JavaFX 2

X2

JavaFX and Event Handling

Event Handling

Event handling is another control flow construct.

- Branches (a conditional or switch selects control flow)
- Loops (a loop repeats control flow)
- Methods (a method call nests control flow)
- Events (the occurrence of event changes control flow)
 - Event handling in UIs
 - Exception handling (later)

Events and Passing Code in Java

An event handler executes some code when a certain event occurs.

Q: How do I pass *code* as an argument in Java?

A: Lambda expressions (since Java 8), see unit J09

Events in JavaFX

Events are instances of `javafx.event.Event`

- Event properties:

- Event type
- Source
- Target

- Event handlers

- Pass the Lambda expression. For example:

```
scene.setOnKeyTyped(event -> { ...your code... })
```

COMP1140

JavaFX Transformations

X3

JavaFX Transformations

JavaFX Transformations

2D and 3D

- Our focus is 2D, but JavaFX supports 3D

Translation

- Shifts a node from one position to another

Rotation

- Rotates a node around some point



Australian
National
University

COMP1140

JavaFX 4

X4

JavaFX Animation

The Game Loop

Classically, a game will have at its heart a loop like this:

```
while (notEndOfGame) {  
    updateGameState();  
    renderGame();  
}
```

The frequency of the loop is referred to as the **frame rate**, and can determine the “smoothness” of the game play. 35mm film runs at 18”/sec, which is 24FPS. Video games often run at about 30-60FPS.

A Game Loop in JavaFX

JavaFX takes care of rendering (completely), and will update at about 60FPS when possible.

The `AnimationTimer` class can be used to get a call back every frame.

```
new AnimationTimer() {  
    @Override  
    public void handle(long now) {  
        updateGameState(now);  
    }  
}.start();
```



Australian
National
University

COMP1140 JavaFX 5

JavaFX and Junit

X5

JavaFX and JUnit

Unfortunately this is a messy juxtaposition

- Both have their own special execution environment which are incompatible.
- I've written a class `JFXTest` to get you started
 - Tests need to be in a class that extends `JFXTest`
 - Tests passed to `runJFXTest()` via anonymous class.
 - Use `assertJFXTrue()` rather than `assertTrue()`
 - You must call `endOfJFXTest()` at the end of each test
 - Results gathered and tested afterwards