JavaFX Example

The Green-Red Button

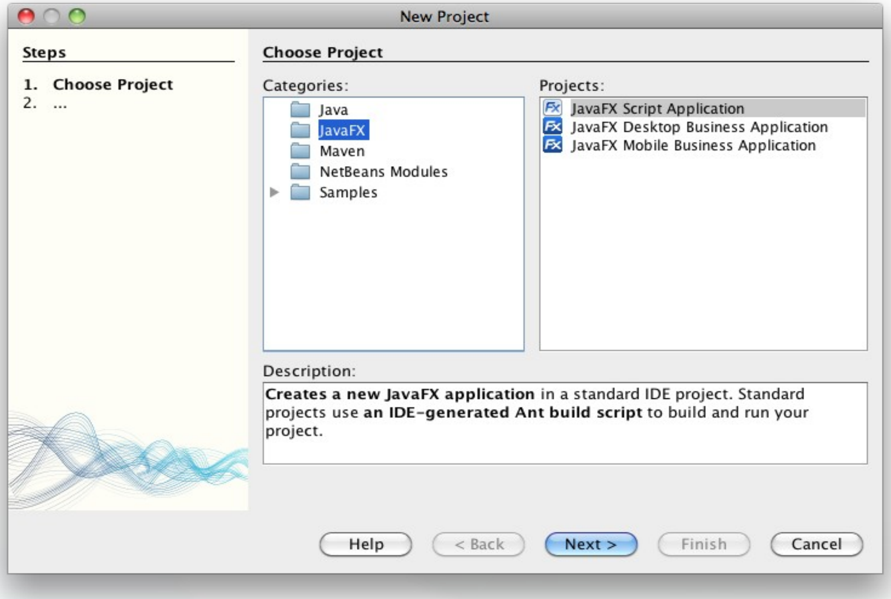
Goals

- Show bind
- Show effects
- Show trigger
- Show CustomNode
- Show translate
- Challenge to you

crisp

Create Application

- Start Netbeans
- Create a new JavaFX application


New Project

Steps

1. Choose Project
2. ...

Choose Project

Categories:

- Java
- JavaFX
- Maven
- NetBeans Modules
- Samples

Projects:

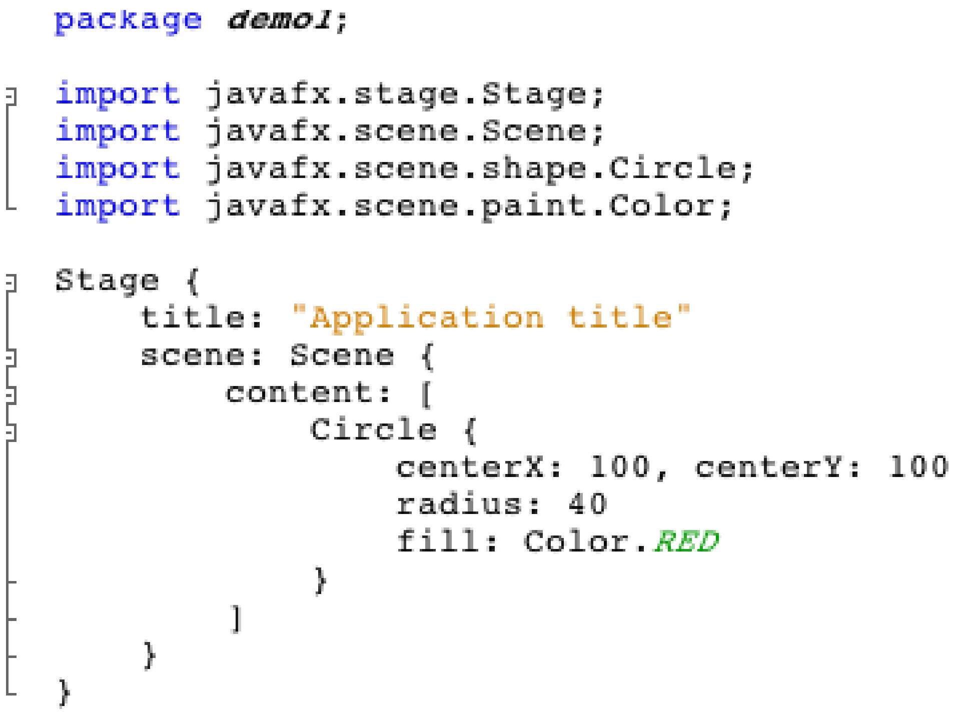
- JavaFX Script Application
- JavaFX Desktop Business Application
- JavaFX Mobile Business Application

Description:

Creates a new JavaFX application in a standard IDE project. Standard projects use **an IDE-generated Ant build script** to build and run your project.

Create a Circle

- Create a Circle by dragging from palette
- Set the color to red
- Watch the preview



```
package demo1;

import javafx.stage.Stage;
import javafx.scene.Scene;
import javafx.scene.shape.Circle;
import javafx.scene.paint.Color;

Stage {
    title: "Application title"
    scene: Scene {
        content: [
            Circle {
                centerX: 100, centerY: 100
                radius: 40
                fill: Color.RED
            }
        ]
    }
}
```

Add state

- Add a state variable at script level

```
var state = -1;
```

- BIND the color of the circle to the state

```
fill: bind if (state > 0) Color.GREEN else Color.RED
```

- Drag a mouse action to the circle

```
onMouseClicked: function (e: MouseEvent): Void {
   state = state * -1;
}
```

- Run (shift-F6). Verify that the color changes

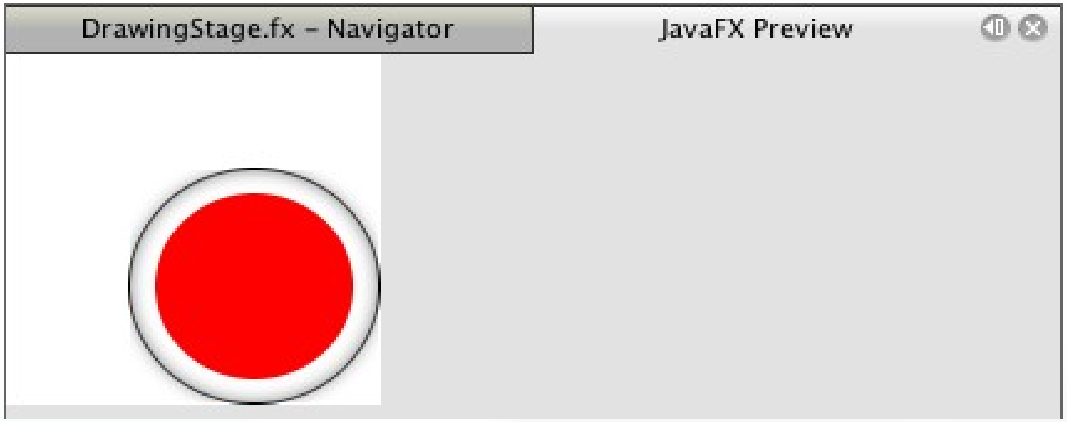
Decorate

- Add a grey circle behind (before) the current circle
- Let the grey circle have a larger radius (50)
- Add a black circle before the grey
- Let the black circle have a radius one unit larger than the grey
- Save and watch the preview

Add Glow Effect

- Add a glow effect to the grey circle

```
effect: Glow {
            level: 1
    }
```



Create a Custom Node

- Create a button class that extends CustomNode
    - Use Group to join circles into a button

```
class Button extends CustomNode {
    var state = -1;
    override protected function create(): Node {
        Group {
            content: [
                Circle { … }
                Circle { … }
                Circle { … }
            ]
} }
```

Instances

- Insert instances of the new class into the scene
- Insert a second instance of a button
- Translate the second button to the right

```
Stage {
    title: "Application title"
    scene: Scene {
        content: [
                Button{}
                Button{ translateX:110 }
                ]
    }
}
```

Trigger

- Insert a trigger to the state

```
var state = -1 on replace oldValue {
  println("Value changed from {oldValue} to {state}")
};
```

- Value changed from 0 to -1
- Value changed from 0 to -1
- Value changed from -1 to 1
- Value changed from -1 to 1
- Value changed from 1 to -1

Challenge

- Create a bin-hex-dec converter
  - It should be possible to change any number and the other should follow
  - Use layout help from Panel, Vbox and HBox

- Use Radial Gradient on the buttons