GUI development with JavaFX Script

Alexander Scherbatiy

JavaFX Overview

JavaFX is a new family of products based on Java technology designed to enable consistent user experiences

JavaFX includes:

- **JavaFX Script**

- JavaFX Mobile

- Enhanced JRE

- Tools support

JavaFX Script Language

- UI oriented
  - > easier, less code Swing development
  - > easy Java 2D graphics
- Full-featured programming language
  - > object-oriented
  - > declarative syntax
  - > data binding
  - > triggers, array queries and more
- Java Integration

HelloWorld.fx


```
import javafx.ui.*;

Frame{
    width: 300
    height: 100
    title: "JavaFX Hello World!"
    content: SimpleLabel  {
        text: "Hello World!"
        icon: Image {
            url: "{__DIR__}/resources/dukeWaveRed.gif"
        }
    }
    visible: true
}
```

Full Swing support

- All main Swing widgets supported
  - > Components
  - > Layouts
  - > Menus
  - > etc...
- Declarative syntax for UI definition
- Simple event model
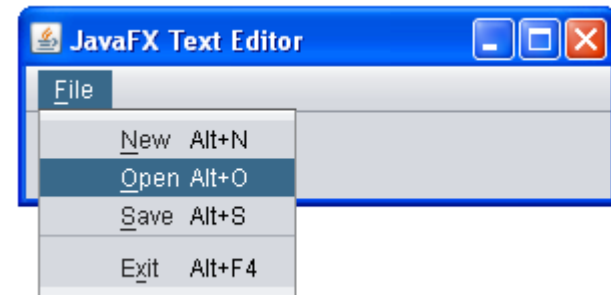- Possibility to extend existing widgets

Declarative Syntax for UI Definition

Form Follows Function concept


```
Frame{
    width: 300
    height: 100
    title: "JavaFX Text Editor"
    menubar: MenuBar {
        menus:  Menu {
            text: "File"
            mnemonic: F
            items: [
            MenuItem {
                text          : "New"
                mnemonic  : N
                accelerator : { modifier: ALT, keyStroke: N }
                action        : operation() {}
            },
```
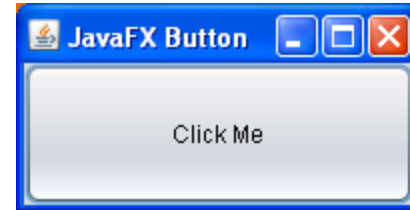
Simple Event Model

```
Frame {
    title: "JavaFX Button"
    content: Button {
        text: "Click Me"
        action: operation(){
            System.out.println("Button is pressed");
        }
    }
}
```

Full Java 2D Support

- Complete support for
  - > shapes
  - > transformations
  - > filters
  - > animation
- Easy graphics manipulation, events and animation

Hello World 2D

```
Frame {
    title: "JavaFX Hello World!"
    content: Canvas{
        content: [
        Star {
            filter     : ShapeBurst
            points     : 12
        },
        Text{
            filter     : ShadowFilter
            font       : new Font("Verdana", "ITALIC", 24)
            content : "Hello World!"
            fill        : red
        },
        ]
    }
}
```

Animation Sample


```
var rotationValue =
        bind [0..360] dur 2000 continue if true;

Frame{
  content: Canvas{
    content: Star {
      //define star attributes here
      transform: bind rotate(rotationValue, 100, 100)
    }
  }
}
```

JavaFX Script Language Features

- Data Binding
  - > variables/attributes binding
  - > operations binding
- Triggers
  - > object creation
  - > attribute value change
  - > arrays modifications
- Enhanced Arrays Support
  - > arrays queries
  - > declarative syntax for arrays operations

Data Binding

```
function max (a, b) =  if a < b then b else a;

var x = 10;
var y = 20;

var maxXY = bind max(x,y);

System.out.println("max( {x}, {y} ) = {maxXY}");   //   max( 10, 20 ) = 20

x = 30;

System.out.println("max( {x}, {y} ) = {maxXY}");   //   max( 30, 20 ) = 30

y = 40;

System.out.println("max( {x}, {y} ) = {maxXY}");   //   max( 30, 40 ) = 40
```
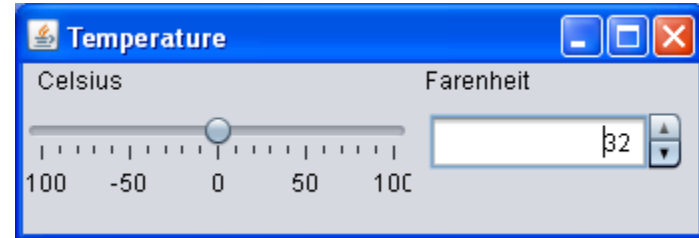
Bidirectional Data Binding


```
var temperature = 0;

    BorderPanel {
        center: FlowPanel {
            content:
                Slider {  value: bind ( temperature - 32 ) / 4.5  }
        }
        right: FlowPanel {
            content: Spinner { value: bind temperature }
        }
    }
```

Triggers

```
class MyClass {
    attribute num: Number;
}

trigger on new MyClass{
    System.out.println("Class Initialization");
}

trigger on MyClass.num [oldValue] = newValue {
    System.out.println("Just replaced {oldValue} with {newValue}");
}

var myClass = MyClass{ num: 10 };

// Output:
// Class Initialization
// Just replaced 0 with 10
```

Arrays Support

```
var x = [ -1, 0, 2 ];

insert 3 into x;                    // x = [ -1, 0, 2, 3]
insert 1 after  x [ n | n == 0 ];   // x = [ -1, 0, 1, 2, 3]
delete x [ 0 ];                     // x = [ 0, 1, 2, 3]

for( n in x )  {   System.out.println(n); }

function factors(n) {
    return select i from i in [ 1 .. (n/2) ] where n % i == 0;
}
 //  factors ( 12 ) = [ 1, 2, 3, 4, 6 ]
```
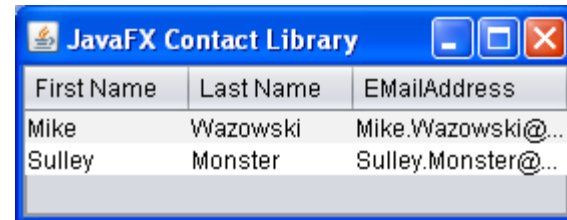
Arrays & Data Binding


```
var selectedRow = 0;

var contacts  = [ Contact {
            firstName: "Mike"
            lastName: "Wazowski"
            eMailAddress: "Mike.Wazowski@monster.com"
    },  ];

Table{
   cells: bind foreach(p in contacts)[
      TableCell { text: bind p.firstName      },
      TableCell { text: bind p.lastName       },
      TableCell { text: bind p.eMailAddress }
   ]
   selection: bind selectedRow
}
```
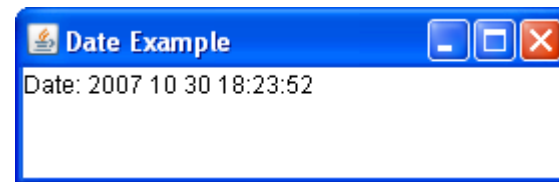
Direct calls to Java API

- Seamless JavaFX Script to Java integration


```
import java.util.Date;

var date = new Date();

Frame {
    title: "Date Example"
    content: Label {
        text: "Date: {date format as <<yyyy MM dd HH:mm:ss>>}"
    }
}
```

JavaFX Interpreter

- Will be open sourced
- Java to JavaFX calls: use Java Scripting API

JavaFX Compiler

- Open Source
- Translates JavaFX Script code into JVM class files (bytecode)
- Leverages and extends the JDK's javac compiler capabilities
- Development is in progress

JavaFX Script Language Support

- Sun tools:
  - > JavaFX Script Plugin for NetBeans
  - > JavaFX Script Plugin for Eclipse
  - > FX Pad
- Third party tools:
  - > JFX Builder
  - > other...

Resources

- https://openjfx.dev.java.net
- https://openjfx-compiler.dev.java.net
- http://jfx.wikia.com
- http://ru.jfx.wikia.com

Tools

- http://javafx.netbeans.org

Aliases

- users@openjfx.dev.java.net
- users-ru@openjfx.dev.java.net

Questions & Answers

Alexander Scherbatiy

Alexandr.Scherbatiy@Sun.COM