JavaFX™ 1.0 SDK

Richard Bair
Martin Brehovsky
Sun Microsystems
Overall Presentation Goal

Dig into the JavaFX 1.0 SDK
Our qualifications

- Richard and Martin are on the JavaFX SDK team
- Martin lead the team responsible for the JavaFX Production Suite
- Richard is one of the API leads for JavaFX SDK
JavaFX is the new client stack for graphical Java applications across all devices.
Agenda

- Introduction to JavaFX
- (Brief) JavaFX Script
- Scene Graph
- Animations and Transitions
- Q & A
Introduction to JavaFX™

- Common APIs across devices
- Scales from small devices to powerful desktops
- Brings rich media and graphical APIs to Java
- Simplifies building graphical consumer and enterprise applications
JavaFX Script

- Expression language
- Declarative and Procedural
- Integrates with Java
- Loosely based on JavaScript
JavaFX Script Explained

- JavaFX Script source files are called “scripts”
- Everything* in JavaFX Script is an expression
- All blocks are expressions
- The last line is the result
Primitive Data Types

- Boolean
- Integer
- Number
- String
- Duration
- Primitives cannot be null*
// A sequence of one can omit the brackets
var names:String[] = "Richard";

// Empty sequences are not null
var names:String[];
println(sizeof names); // prints 0

// elements are accessed by index
var names = ["Jasper", "Richard", "Martin"];
var martin = names[2];

// and you can populate a sequence using a for loop
var hellos = for (i in [1..3]) { "Hello #{i}" }
Modifying Sequences

// Inserting items into a sequence
var names: String[] = "Richard";
insert "Jasper" into names;

// Inserting before a certain index
insert "Martin" before names[1];

// Inserting after a certain index
insert "Duke" after names[1];

// Deleting from the sequence
delete "Duke" from names;
Operators

- + - / *
- ++ --
- *= /= += -=
- and or not
- = == !=
- mod
Flow Control

- if (booleanExpression) then a else b
- if (booleanExpression) a else b
- if (booleanExpression) { a } else { b }
- while (booleanExpression) { ... }
- for (i in sequence) { ... }
- Can get index of item “i” by “indexof i”
- break
- continue
**Bind**

- **bind** is a way to tie the value of one variable to the value of an expression.
- Binding must be defined when the variable is initialized.
- Bindings are statically compiled.
- Bound variables cannot be set manually.
public class Distance extends {
  public var x1:Number;
  public var x2:Number;
  // Whenever x2 or x1 changes, distance will be updated
  // Binding can be used for invariants
  public-read var distance:Number = bind x2 - x1;
}
Object Literals

- Concise declarative syntax for object creation
- Similar to JavaScript
- Combine with binding for maximum effect
- variable: initial-value
  - initial-value is an expression
// creates a Rectangle
// x: 10 is not an assignment, it is an initialization!
var rect = Rectangle {
    x: 10
    y: 10
    width: 100
    height: 100
}
// creates a Rectangle with a Color for its fill
var rect = Rectangle {
    x: 10
    y: 10
    width: 100
    height: 100
    fill: Color {
        red: 1
        green: 0
        blue: 0
    }
}
// A variation that allows me to reference the color later
var color: Color;
var rect = Rectangle {
    x: 10
    y: 10
    width: 100
    height: 100
    fill: color = Color {
        red: 1
        green: 0
        blue: 0
    }
}
Scene Graph

- Describes the graphics and controls in a scene
- Each node in the graph has a single parent
- Special “group” nodes have zero or more children
- “leaf” nodes have no children
- Graph is set on a Scene
- Scene is set in a Stage
Scene Graph

Diagram showing a scene graph with nodes for Image, Group, Circle, and Line.
Scene

- Canvas upon which the Scene Graph is displayed
- Can set multiple CSS Stylesheets
- Can set background color (or set to null)
- Can set canvas width / height
Stage

- Top-level container for the scene
- Contains only one Scene
- Can set Stage width / height
- Potentially represented by:
  - JFrame on desktop
  - JApplet on web page
  - SVG player on mobile
Custom Node

- Primary method of Scene Graph encapsulation
- All other nodes are not extendable
- Simply override the create() method
Shapes - Building Blocks

Basic Shapes
- Arc
- Circle
- Ellipse
- Line
- Path
- Polygon
- Rectangle

Common Variables
- stroke
- strokeWidth
- fill
- smooth
Create an App (Part 1)
Colors

- 150+ built in colors (all the HTML & CSS built in values)
  - Color.web(“#aabbcc”)
  - Color.web(“blue”)
  - Color.rgb(128, 222, 21)
Linear Gradients

- startX, startY, endX, endY
  - Define the direction of the gradient
- On the unit square
- Stops define each step in the gradient. Each stop
  - Has an offset between 0...1
  - Has a Color
Create an App (Part 2)
Images

- ImageView node shows images
- Image represents an in memory Image
- Image can load images in FG thread or BG thread
- Both ImageView and Image can scale
  - Preserve ratio
  - Fit within a specific width/height
  - When fit on Image level, keeps smaller image in memory
Create an App (Part 3)
Text

- x, y, TextOrigin
- By default, text positioned such that (x, y) is left baseline
- Fonts can be specified on Text
- Favor “fill” over “stroke”
- Supports multiline text
- Use alignment to align multiline text
- To center text, compute the center via layout bounds
Used for text input

Use CSS to style the TextBox

“text” is changed to reflect the actual text in the TextBox

“value” is changed when the text is “committed” via ENTER, TAB, etc

“action” function is invoked when ENTER pressed

“columns” specifies the preferred width based on the font size and number of characters to display
Effects

- Effects are placed on Nodes
- Many standard built in
  - DropShadow
  - ColorAdjust
  - GaussianBlur
  - Glow
  - Reflection
- more...
Create an App (Part 4)
JavaFX supports both visual and audio media
Cross platform JavaFX Media file (fxm, mp3)
Also plays native formats (mov, wmv)
Media class represents a media file
MediaPlayer plays a Media file
MediaView is the Node which displays the Media
No built in Movie playing Control
Animation

- Animation is a key feature of every rich graphics application platform
- There are two supported animation types in JavaFX
  - Keyframe Animations
  - Transitions
Keyframe Animation

- A general purpose animation mechanism for JavaFX
- Uses a concept of a timeline and keyframes specifying values at given times
- Built in the language syntax - can animate any variable
tween keyword and custom interpolators

- Long syntax vs short syntax
- autorepeat + autoreverse behavior

- Actions

- Nested timelines
Keyframe Animation Example

var simpleTimeline = Timeline {
  keyFrames: [
    KeyFrame {
      time: 0s
      values: [
        node1.translateX => 50
        node1.rotate => 0
      ]
    },
    KeyFrame {
      time: 3s
      values: [
        node1.translateX => 800 tween Interpolator.EASEBOTH
        node1.rotate => 720 tween Interpolator.EASEBOTH
      ]
    }
  ]
}
Transitions

- “Precanned”, single purpose animations
- Translation/Rotation/Scale/Fade
- “by” values
- Animation along a path
- Containers:
  - Parallel
  - Sequential
- Pause Transition
var complexTransition = ParallelTransition { content: [  
  SequentialTransition { node: node2  content: [  
    PauseTransition { duration: 200ms },  
    ParallelTransition { duration: 600ms content: [  
      TranslateTransition { duration: 500ms byY: 100 byX: 100},  
      ScaleTransition {duration: 200ms byX: -0.15 byY: -0.15},  
      RotateTransition {duration: 500ms byAngle: -180}  
    ]},  
  ]},  
]}
SequentialTransition { node: node3  content: [  
  PauseTransition { duration: 400ms },  
  ParallelTransition { duration: 600ms content: [  
    TranslateTransition { duration: 500ms byY: -100 byX: 100},  
    ScaleTransition {duration: 200ms byX: -0.15 byY: -0.15},  
    RotateTransition {duration: 500ms byAngle: 180}  
  ]},  
]}
]
Create an App (Part 5)
Summary

JavaFX 1.0 SDK provides the APIs necessary for building creative applications

Future updates to JavaFX will come regularly and quickly

1.1 release around February 2009

1.5 release around June 2009

JavaFX 1.0 Production Suite allows you to use professional graphics in your apps
Concluding statement

Try JavaFX Today
Thanks for your attention!

http://javafx.com