

JavaFX: A RIA Solution To Reinvigorate Your Desktop

A Close Examination of JavaFX

JavaFX Compiler, Library and Use-Cases

29th July 2008

Peter Pilgrim

E-Channel Services,
Technical Lead

Lloyds TSB Corporate Markets, London



Lloyds TSB



Lecture Outline

- What is JavaFX? The Why Here & The Why Now?
- Part 1
 - Applets Came and then Left The Building
- Part 2
 - **JavaFX™ Languages Features**
 - **Syntax, User Interface, Animation**
- Part 3
 - **Integration with Java™ and Libraries**
 - What Lies Beneath?
- Summary




Charles Lowell, Drunk and Retired
Podcast, Episode 127

"I would almost call it avant garde."

JavaFX is a media focused, user interface, compiled and statically declarative language for the JVM

Why JavaFX?

- After conquering the server side. It's time to reinvigorate the client side of Java.
- Improvement by learning from mistakes from the past
- Research for developer interest beyond Java anyway
- Idea of JavaFX was born in the left field. **Non Sun Idea!**
- Christopher Oliver, a UI engineer, SeeBeyond 2005
- Sun buys See **Beyond**
- Oliver had a pet project F3 (*Form Follows Function*)



Part 1:

The Java Applet appeared one day and then vanished from these shores ...

1995: The Java Applet Arrived

- In 1995 Sun Demonstrated small Java applications embedded inside a HTML web page
- Java Applets were born
- Java Security Sandbox module
- Abstract Windows Toolkit
- Network Enabled
- Portable on machines with Java Plug-in Installed

And Then It Departed These Shores

- JVM/JRE had to be installed on the client machines
- Administration headaches, big unoptimised download
- Sun fought with Microsoft in the US courts **MSJVM**
- Plugin implementation was strongly coupled with the browser API
- Macromedia bought **FutureWave** and rolled about successive releases of **Flash** and **SWF**
- Then Microsoft entered the RIA scene with **SilverLight**

Technology Timeline Keeps On Moving On

- Java became the stronghold of the server-side
- Sun rolled out Java **WebStart**
- Sun attempted some media support with **JMF** and **JavaSound**
- Popularity of Java on User Interface decline in face of Flash, **AJAX** and other lighter weight solutions
- Programmers started to look beyond Java Language
- Swing was too hard much work to get great user interface despite the great work of **Haase** and **Guy** (*Authors of the Filthy Rich Clients book*)



*“Our goal with JavaFX is to deliver a
'Media Stack' for the Java platform ...
It's means five things: Audio, Video,
2D and 3D Graphics and Animation”*

Christopher Oliver

JavaFX Marks Return of Sun to User Interfaces

- Innovation probably cannot take place on the server-side for the rest of this decade.
- We have **Spring Framework** and can look forward to **Java EE 6** and modularisation (**OSGi**), but what other ideas are there other concurrency improvements?
- The real question is: How can we help *The Business* do their job better? We need to solve usability issues with Java on the UI if it is survive into the 2010s.

The Tale of the Consumer



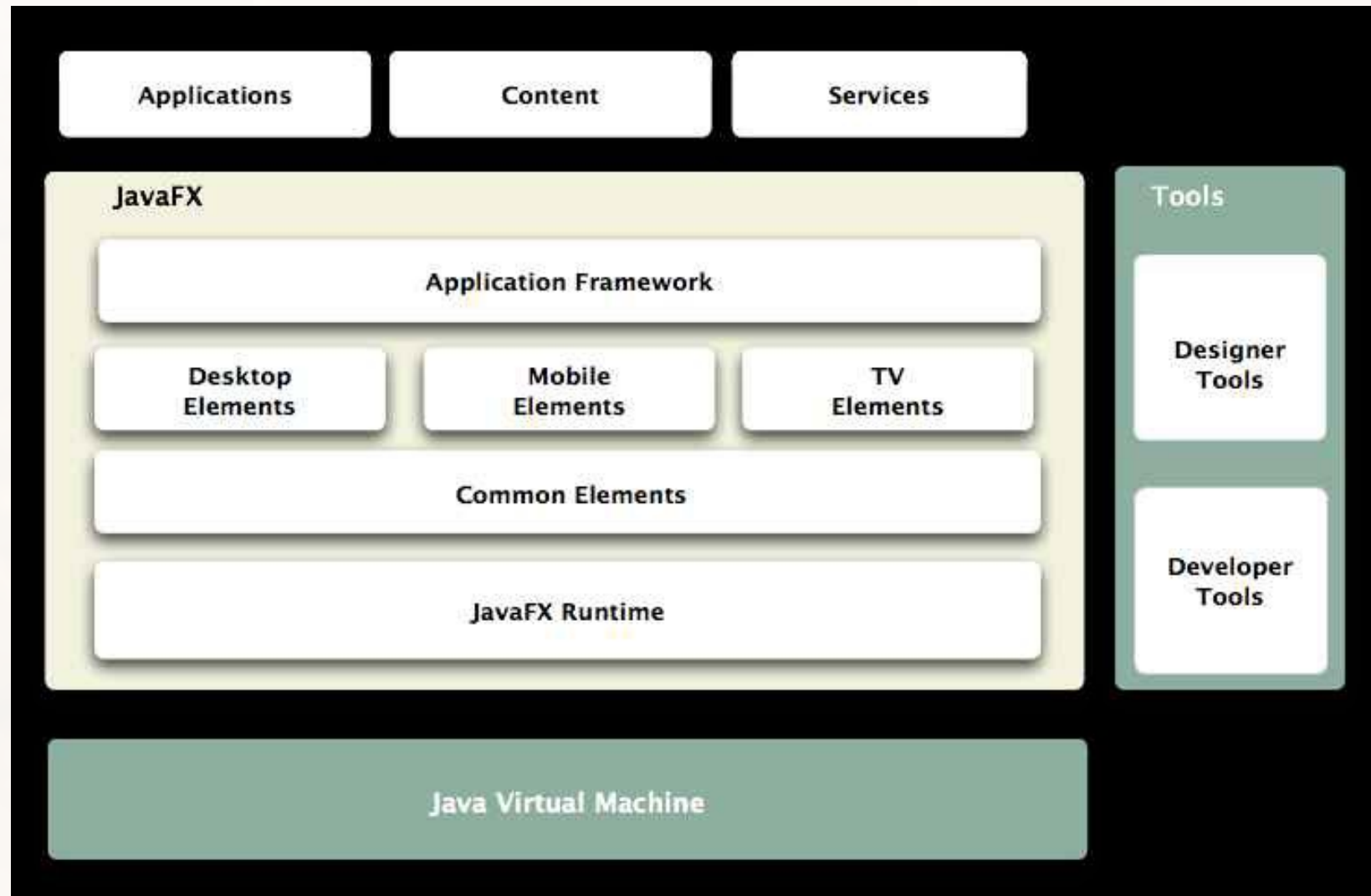
Internet Mobile Usage Will Increase,
Worldwide

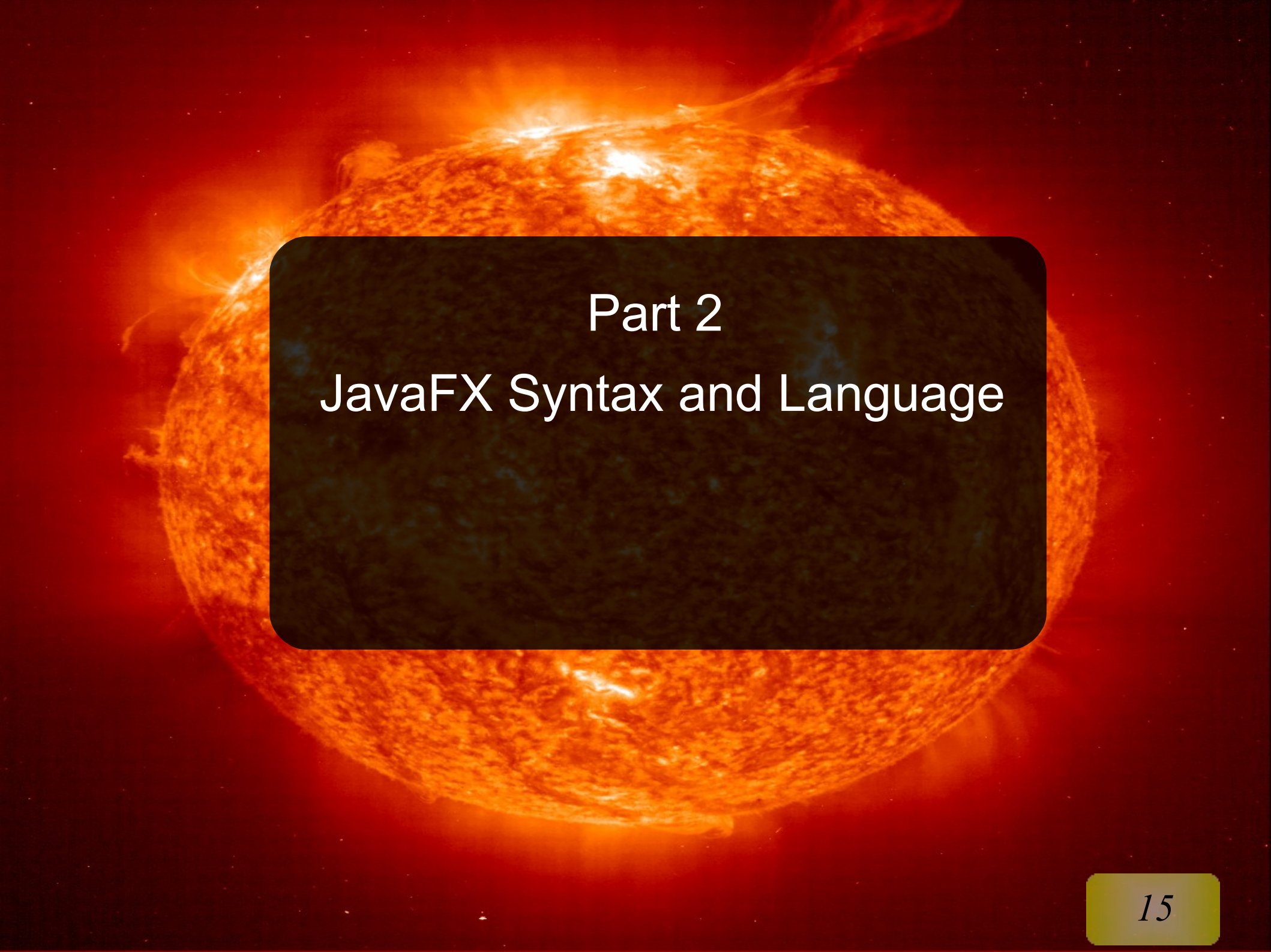
The Tale of the Prosumer



In the so-called developed world, the blurring of lines continues unabated.

JavaFX Architecture





Part 2

JavaFX Syntax and Language

Java

```
JFrame f = new JFrame();
if (com.awt.sun.util.AWTUtilities.isTranslucencySupported()) {
    com.awt.sun.util.AWTUtilities.setTranslucency(true);
}
frame.setUndecorated(true);
JPanel panel = new JPanel() {
    protected void paintComponent(Graphics g) {
        Graphics2D gfx = (Graphics2D)g.create();
        gfx.setRenderingHints(RenderingHints.KEY_ANTIALIASING,
                               RenderingHints.VALUE_ANTIALIASING_ON);
        gfx.setColor(Color.RED);
        gfx.drawOval(0,0,100,100);
        gfx.dispose();
    }
}
frame.getRootPane().setContent(panel);
frame.setVisible(true);
```

JavaFX

```
Frame {  
    stage: Stage {  
        content: Circle {  
            centerX: 50  
            centerY: 50  
            radius: 50  
            fill: Color.RED  
        }  
        fill: null // turn off  
the background  
    }  
}
```


JavaFX Reserved Words

abstract, after, assert, **attribute**, **bind**, break,
class, continue, **def**, delete, false, for, **function**,
if, import, init, **insert**, let, package, private,
protected, public, readonly, return, super,
sizeof, static, this, throw, try, true, **var**, while,
after, and, as, **before**, by, **catch**, or, **and**, **not**, **between**
do, dur, else, exclusive, extends, finally,
first, from, **in**, bound, indexof, into, **inverse**,
lazy, **last**, on, or, **replace**, step, typeof,
with, **where**, instanceof, **override**, at, then, **tween**

JavaFX Declarative Statically Type Language

- A new **statically typed** scripting language for the JVM
- Leverages power of existing Java libraries
- **Declarative syntax** for User Interface
- Formalises **Binding** between UI components and UI models
- **Sequences** and **List Comprehensions**
- Basic closures support: **Anonymous Functions** with lexical scope
- A JVM language platform that blurs the distinction between the **web** and **desktop**
- Designed for **content designers** and **media engineers**

JavaFX Data Types

- Compiled JavaFX™ supports four primitive types
 - **String** (for java.lang.String)
 - **Boolean** (for java.lang.Boolean)
 - **Number** (for java.lang.Number)
 - **Integer** (for byte, short, int, long, BigInteger)
 - **Void** (for functions)
- Compiled JavaFX™ support objects
 - Object oriented type, single hierarchy like Java
 - No interfaces though, prefer composition instead

Declaring and Assigning FX Data Types

```
var x:Number = 0.98765;  
var y:Integer = 100;  
var name:String = "Peter Pilgrim"  
var flag:Boolean = true;  
  
var numbers:Number = [ 1,2,3,4,5];  
  
var num = 1; // Inferred type  
var fullName =  
    "Lewis Hamilton"; //ditto
```

Control Statements in FX

```
if ( x==y ) { /*..*/ }  
else if (x==z) { /*..*/ }  
else { /*...*/ }
```

```
while ( count != 100) { /*...*/ }
```

```
for (j in [0..10] ) { /* ... */ }
```

```
function vat( x:Number ): Number {  
    var t:Number = (17.5/100.0)*x;  
    return t;  
}
```

Declaring JavaFX Classes

- Create classes with JavaFX
 - Subclass existing Java classes and interfaces
 - Other JavaFX™ classes
- Compiled JavaFX
 - In-class definition of attributes and functions
 - Time-lines, animations and pointer syntax
 - Define lexically scoped variables

A Simple Class Definition

```
class OrderDetail {  
    private attribute orderId: Integer;  
    protected attribute orderName: String;  
    public attribute products: Order[];  
    function addOrder( order: Order ): void {  
        /*...*/  
    }  
    public function calcPrice(): Number {  
        /*...*/  
    }  
}
```

Sequences unique to JavaFX

- Sequences are dynamic arrays on a wildfire.
- Insertion
 - Before the first element
 - After the last element
 - Before the Nth indexed position
 - After the Nth indexed position
- Removal
 - In-class definition of attributes and functions
- Sequences are intimately associated with list comprehensions
 - List comprehension are supported with **where** and **for** keywords.

List Comprehension Example

```
var planets:String[] = ["Mercury", "Venus", "Earth",  
"Mars", "Saturn", "Neptune"];  
insert "Uranus" before planets[5];  
insert "Jupiter" after planets[3];  
insert "Pluto" into planets;
```

```
var somePlanets:String[];  
somePlanets =  
    for (planet in planets  
        where planet.length() > 5)  
        "{planet},";
```

```
System.out.println("The {sizeof somePlanets} planets  
with more than 5 characters are:    {somePlanets}");
```


Triggers and Binding

• Triggers

- replaces setters and getters
- Validate new and old values in the trigger “advice”
- Weak form of cross-cutting concern for UI model and property change “events”.

• Binding

- Avoids JavaBeans properties, events, event publishing event listeners boilerplate!
- One keyword **bind**
- Can also be bidirectional! (**with inverse**)

Example of Bind Usage – A Menu List

```
class CcyPairModel {  
    attribute ccyPairs: String[] ;  
    attribute selIndex: Integer;  
}  
var model: CcyPairModel {  
    ccyPairs: ["EURUSD", "EURGBP", "USDJPY" ]; };
```

```
Frame { /*...*/  
    listCcys: List {  
        selectedIndex: bind model.selIndex;  
        items: bind for  
            (ccyPair in model.ccyPairs)  
            ListItem {  
                text: ccyPair  
            } // creates [ListItem "EurUsd",...]  
    }  
}
```

Example of Trigger Usage

```
class Trade {  
    attribute quantity: Integer = 10  
    on replace {  
        if ( quantity <=0 ) {  
            throw new Exception(  
                "Illegal trade quantity: {quantity}"); }  
        }  
    }
```

```
var trade: Trade = { };
```

```
trade.quantity = 10; // OK
```

```
trade.quantity = -1; // whoops!
```


Printing and String Substitution

```
var s1 = 'Hello';  
var s2 = "World";  
var s3 = "{s1} {s2} Peter";  
  
var answer = false;  
var s4 = "Der Antwoert is {if {answer} "Ja" else  
    "Nein, Danke"} "; // Prints 'Nein, Danke'  
  
Var x =33.4184152;  
System.out.println{"x={%6.3f x}"}; // 33.418  
  
import java.util.Date;  
var now = new Date();  
var timeStr = "{%tH:%tM:%tS now}";
```

Animation and Timelines

- Timelines handles the animation in JavaFX.
- Timelines are first-class citizen in the language along with the duration time constants (1s, 10s)
- They can have one or more **KeyFrames**
- Methods: **start()**, **stop()**, **pause()**, **resume()**
- Properties: **autoReverse**, **repeatCount**, **toggle**
- BigDeal: *Timelines are nestable!*

Time Example

```
var xpos: Integer;
var t1 =
    repeatCount: Timeline.INDEFINITE
    Timeline {
        autoReverse: true
        keyFrames: [
            KeyFrame {
                time: 0s
                values: xpos => 200
            },
            KeyFrame {
                time: 1s
                values: xpos => 400 tween Interpolator.LINEAR
            }
        ]
    };
t1.start();
```




Part 3

What Lies Beneath?

In order to implement any reasonable design, one must have good firm foundations

Scene Graph

- SceneGraph – a hierarchical representation of graphics node in a scene.
- Ideas from 3D Computer Graphics, PHIGS, Open-GL but applied to 2D Windows as well
- “Wobbly Windows” are certainly possible if not probable
- Affine transformations of nodes and components
- Swing components can be contained as components
- JavaFX will feature a set new UI elements based on SceneGraph components

Scene Graph Nodes

- SGNODE is the root object classes
- Subclasses can be SGImage, SGShape, SGText, SGComponent, SGFilter, SGGroup or filter
- Filters can be transform, composite, clips
- Nodes can received events like AWT Events
- A scene graph is very powerful model of a landscapes

Java FX Architecture and Project Scene Graph

JavaFX Script Software

Project Scene Graph

Effects

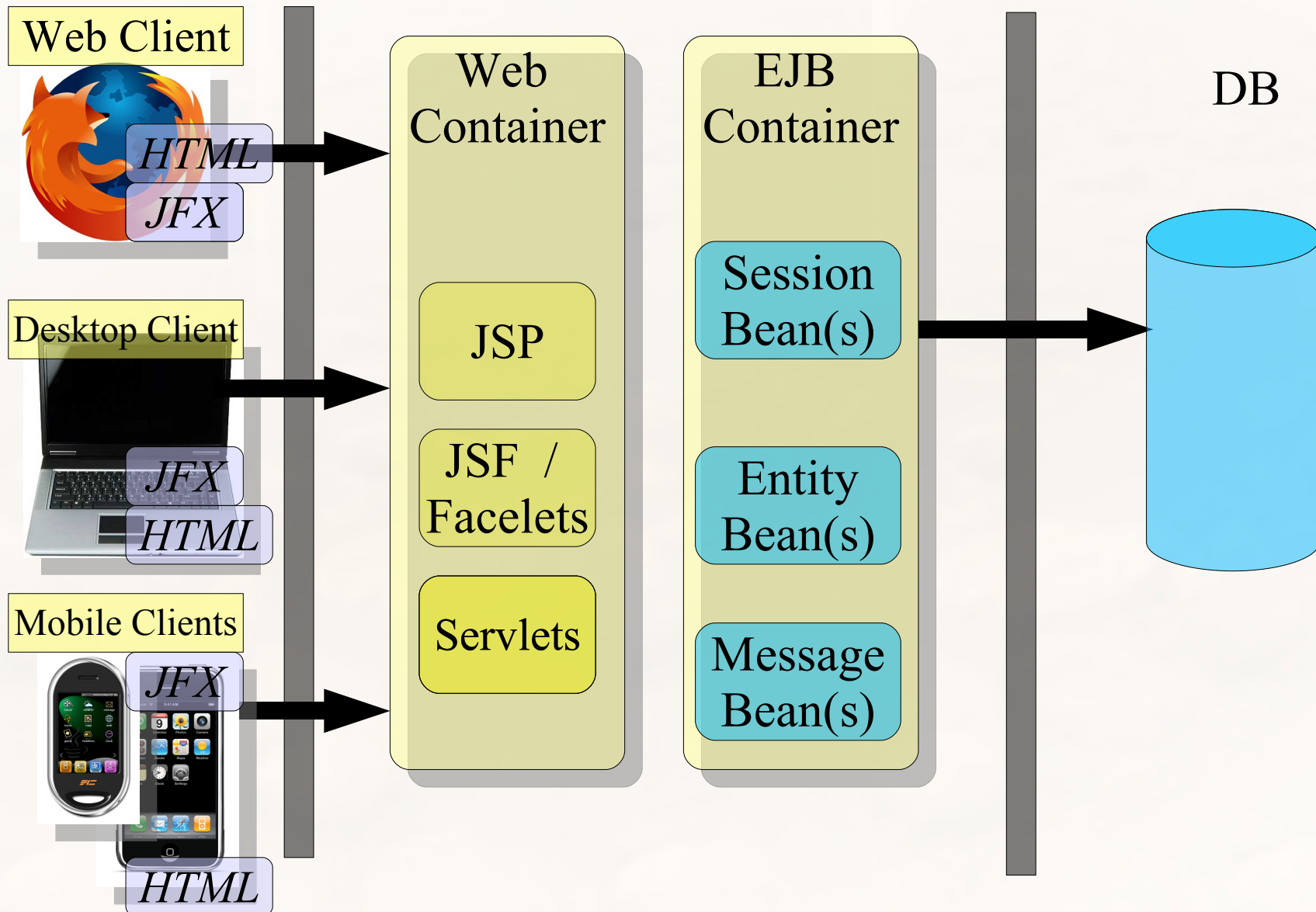
Java 2D

Graphics Hardware

“Consumer JRE”

- Java Runtime Environment 10 Update 6
- Quickstarter – warm versus cold JVM start up
- Java Kernel - modularisation
- Deployment toolkit – great for administrators
- Unified Browser and Desktop Experience – draggable Applets
- Graphics Acceleration on the Windows platform
 - “Someone please write to Steve Jobs” (Mac OS/X)
- Nimbus – new cross platform look and feel

Sample JavaFX Architecture



JavaFX People on Compiled Language

● Compiler Development

- Brian Goetz, Sun
- Per Bothner, Sun
- Richard Bair, Sun, Driving FX Compiler Changes
- Robert Field, Sun Java Compiler
- Kevin Rushford, Sun
- Christopher Oliver, Compiler, UI & Visual Processing
- Petr Nejedly, Sun

JavaFX People Working on GUI Tools

- “Reprise” FX Tools
 - Amy Fowler, Sun
 - Shannon Hickey, Sun
 - Joshua Marinacci, Sun
 - Brent Christian, Sun, GUI JavaFX
 - Silveira Neto, JavaFX Blogger

JavaFX People on the External

- Early Adopters (Outsiders)
 - James Weaver, David Iverson, et al
 - Tom Ball, ex-Sun, recently joined Google, ex FX Compiler Team Manager
 - Kim Topley, External, Author of Early Prentice Hall Swing Books
 - Michael Aziz, External
 - Jo Voordeckers, BeJUG JavaFX Version of Parleys.com

Tom Ball, Java FX Compiler Engineer



Tom Ball, recently joined Google. He was a Distinguished Engineer at Sun Microsystems, working on Java language tools.

<http://weblogs.java.net/blog/tball/>

Joshua Marinacci, JavaFX Designer Tools



Joshua Marinacci is currently a staff engineer for Sun Microsystems working on designer tools for JavaFX. He previously worked on NetBeans and core Java SE.

<http://weblogs.java.net/blog/joshy/>



Executive Summary

*“Indeed, Form does ever follow
Function”,*

*Louis H. Sullivan, Architect, 1856-1924,
The Father of Mordenism*

JavaFX SDK Preview Availability

- Java SE 6 Update 10
<https://jdk6.dev.java.net/6u10ea.html>
- NetBeans 6.1 and JavaFX Plugin
 - Available now and on 31st July 2008
- JavaFX SDK Preview 31st July 2008
 - <http://openjfx.java.sun.com>
 - <http://www.javaafx.com>
- Desktop SDK 1.0 later this year, Mobile SDK in 2009

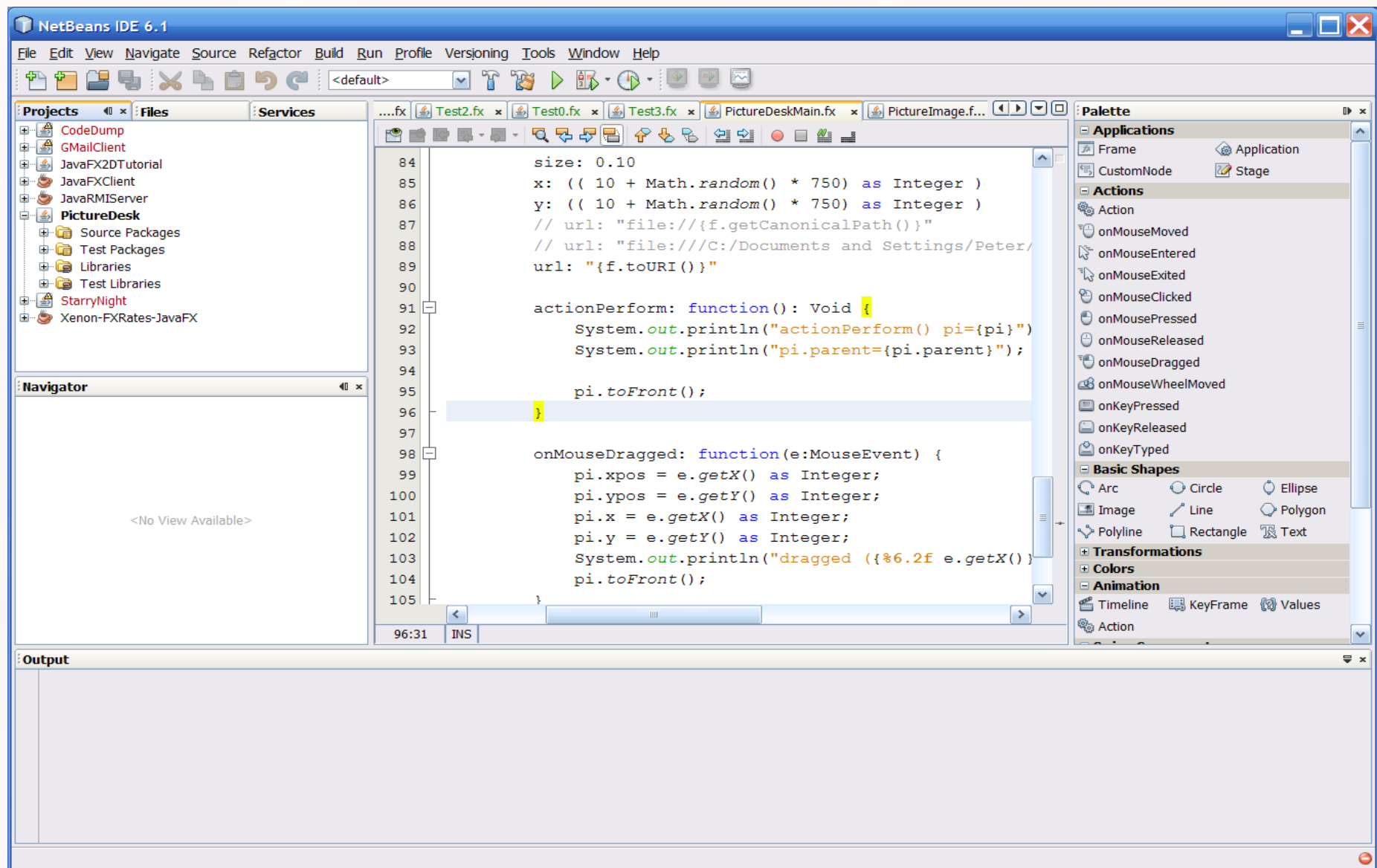
SDK Preview Highlights

- Preview 1 release with Binaries for Windows and MacOS
- JavaFX Documentation
- Tutorials
- Hardware Accelerated OpenGL libraries for Windows
- Java Media Components 1.0
- NetBeans 6.1 Java FX Plugin
- Project Nile
- Reports that Preview SDK Works on Linux
 - <http://learnjavafx.typepad.com/weblog/2008/08/using-tr>

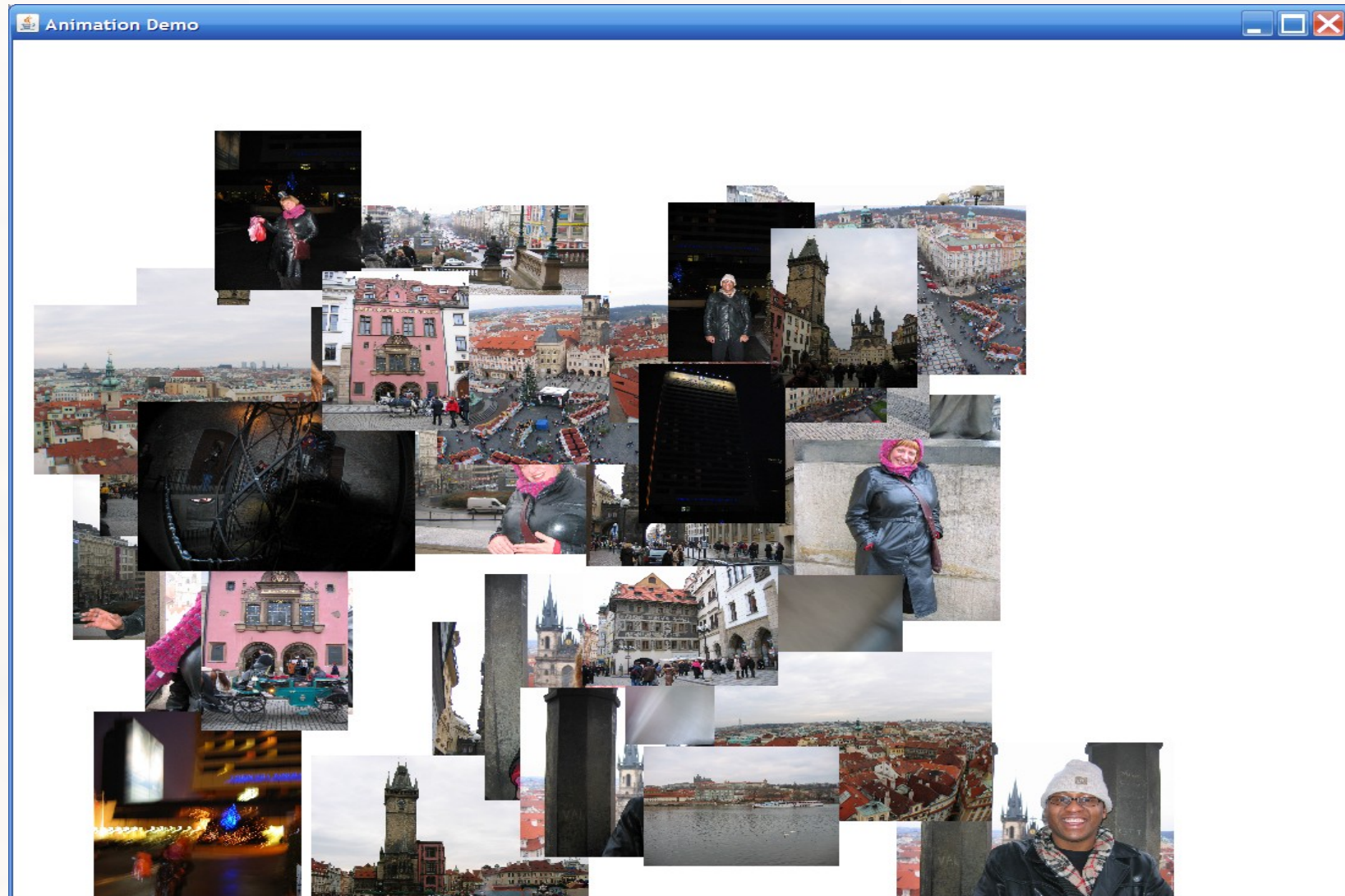
Project Nile (Designer + Developer)

- Sun going to deliver a JavaFX tool set later in the year.
- Photoshop and Illustrator plug-ins for designers
- FXD File Format, FX Viewer, SVG Converter
- Tools for Designer / Developer Collaboration

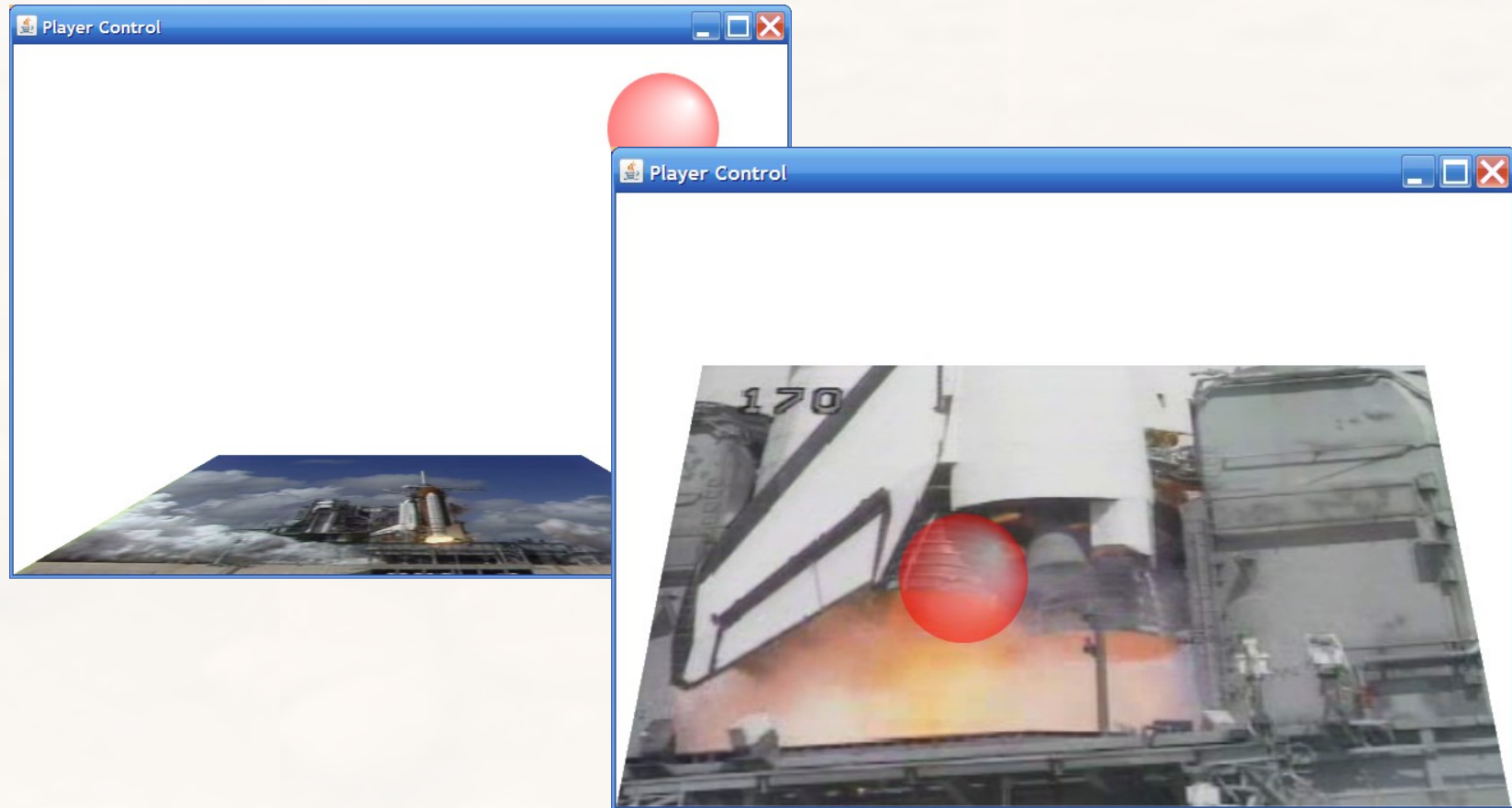
Net Beans JavaFX Plugin



JavaFX Picture Desk: My Naïve Example



JavaFX Play AVI, WMV, MPG Video Files



URL References

● Open JavaFX

<http://openjfx.dev.java.net>

● Open JavaFX™ Compiler

<http://openjfx-compiler.dev.java.net>

● NetBeans 6.1

<http://www.netbeans.org>

● JavaFX™ Portal Preview SDK (31/07/2008)

<http://www.javaafx.com/>

● Peter Pilgrim Blog

http://jroller.com/peter_pilgrim/

● JavaFX Documentation

<http://openjfx.java.sun.com/current-build/doc/index.html>

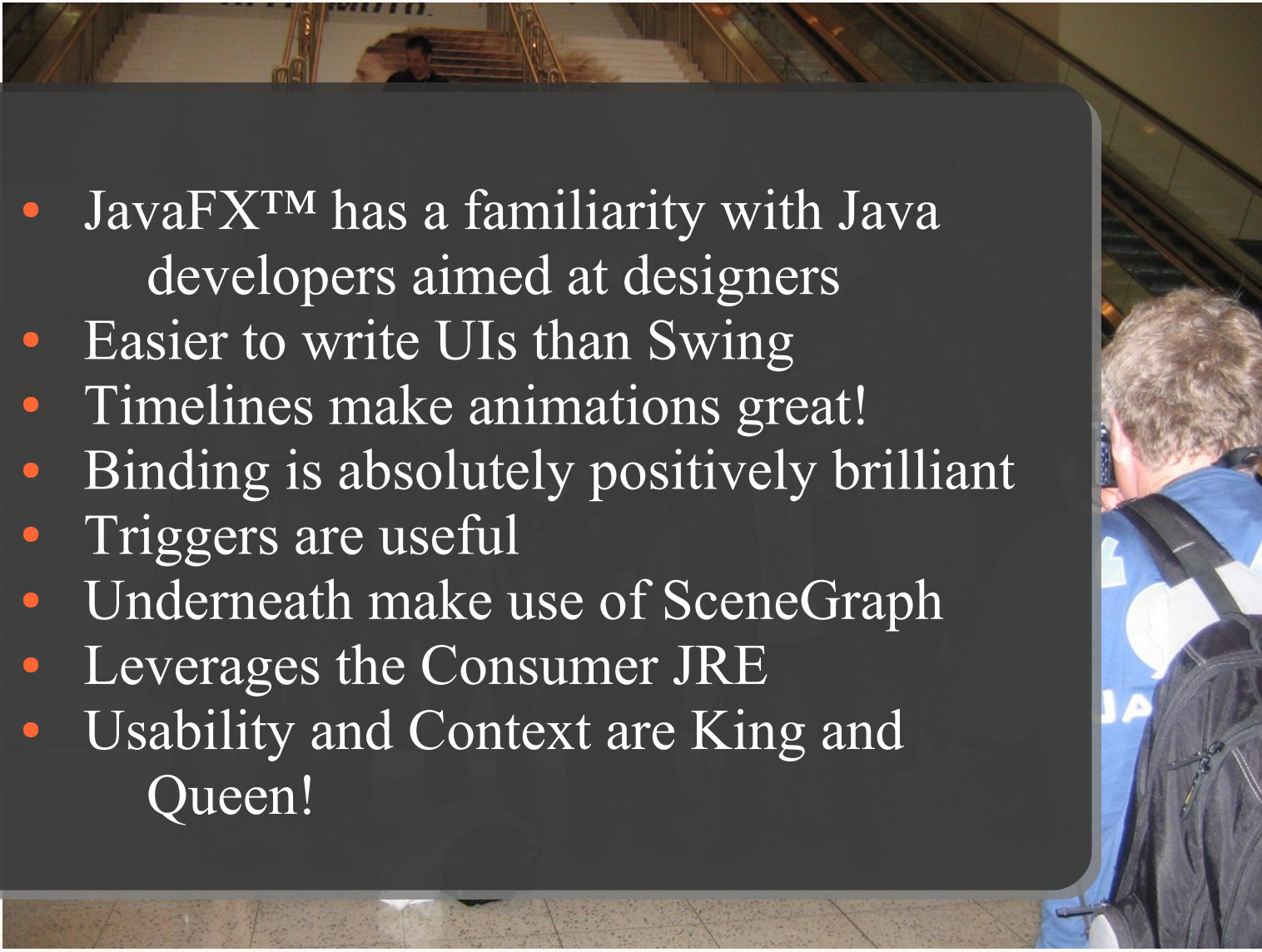
● “Consumer JRE” / Draggable Applet

http://java.sun.com/developer/technicalArticles/javase/6u10_applets/

● JavaFX™ is not JavaScript

http://weblogs.java.net/blog/joshy/archive/2007/09/javaafx_javaafx_s.html

Executive Summary

- 
- The background of the slide features a photograph. On the left, a person is seen from behind, walking up a wide staircase with a metal railing. On the right, a person with curly hair, wearing a blue t-shirt and a black backpack, is looking down at a mobile device. The scene appears to be an indoor public space, possibly a transit station or a large building lobby.
- JavaFX™ has a familiarity with Java developers aimed at designers
 - Easier to write UIs than Swing
 - Timelines make animations great!
 - Binding is absolutely positively brilliant
 - Triggers are useful
 - Underneath make use of SceneGraph
 - Leverages the Consumer JRE
 - Usability and Context are King and Queen!

Thank You All For Listening

Q & A

JavaFX Script To Target Mobile Web

