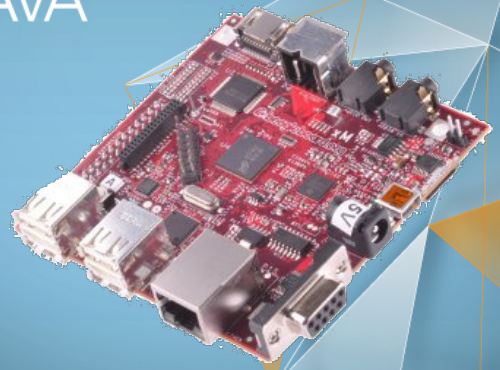

JavaOne™

JavaOne™

JavaFX on Smart Embedded Devices

David Hill
JavaFX Embedded Lead

MAKE THE FUTURE JAVA



ORACLE

Program Agenda

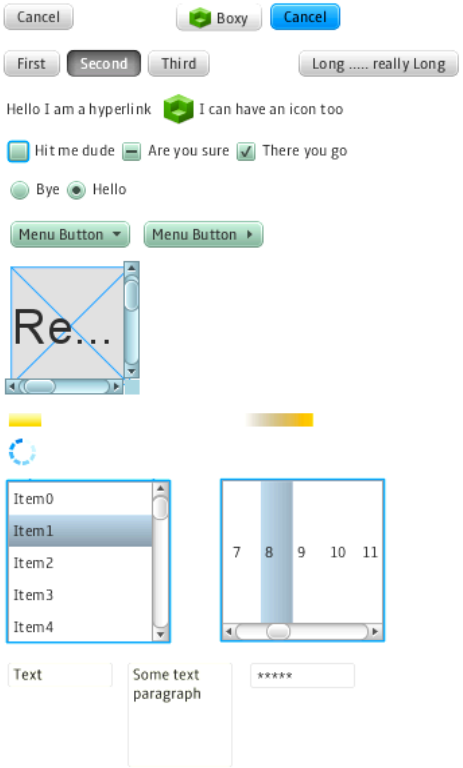
- JavaFX
- JavaFX Embedded
- Targeted Platforms
- Rendering Paths
- Design Considerations
- Demos

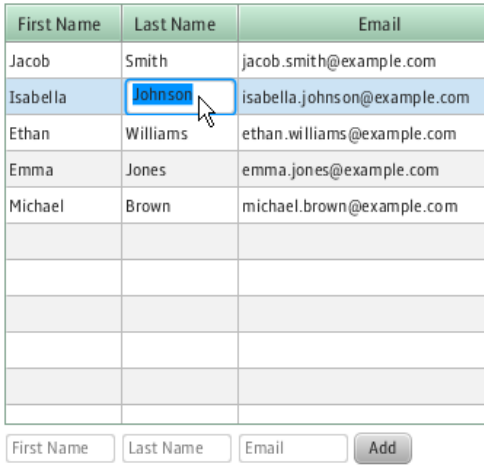
JavaFX

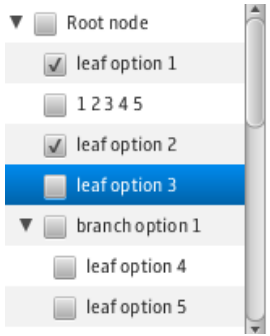
- A rich graphical application framework with built in support for animation, and effects using Java API
- Development tool support
- A graphics engine supporting GPUs
- Full integration with Java SE 7 for desktop

JavaFX

Standard UI Controls

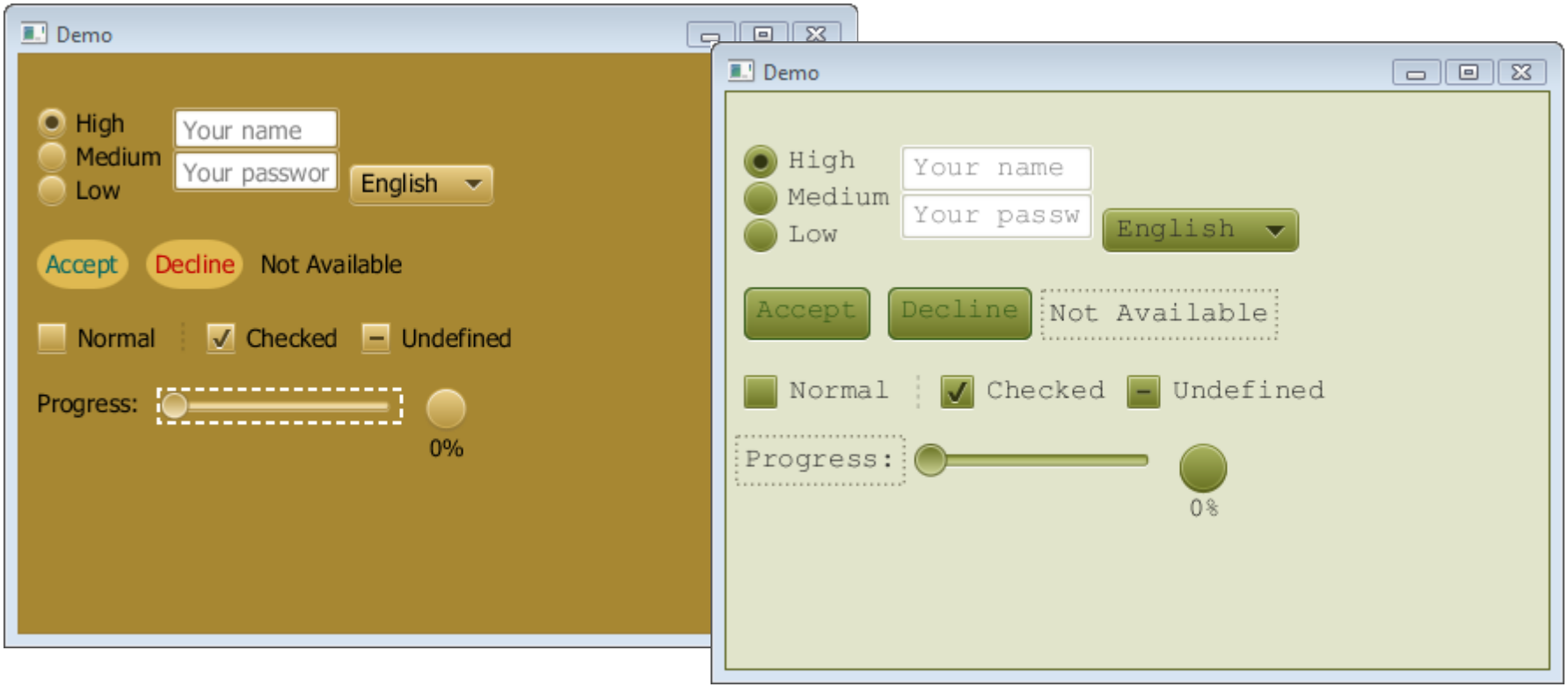






JavaFX

Visually Configurable – Using CSS



Program Agenda

- JavaFX
- JavaFX Embedded
- Targeted Platforms
- Rendering Paths
- Design Considerations
- Demos

JavaFX Embedded

Markets

- Industrial automation
- Home automation
- Home entertainment systems
- Medical devices
- Automotive
- Retail/Info Kiosks

JavaFX Embedded

What I am not talking about today

- For the purposes of this talk, I am not considering JavaFX Embedded for general purpose tablets or mobile phones

JavaFX Embedded

Features

- A proper subset of JavaFX
- No dependency on AWT/Swing
- Common Java and JavaFX development tools
  - Netbeans
  - Scene Builder
  - To name just a few …

JavaFX Embedded

Rendering

- Hardware accelerated and software only rendering
- Support for platform media decoders

JavaFX Embedded

Input

- Traditional mouse and keyboard
-

JavaFX Embedded

Targeted Footprint

-
  - Java Embedded SE (Compact 1 Profile) ~10MB *
    -

* For more details, check out CON4538: Java Embedded Goes Modular

JavaFX Embedded

Optional Footprint

- Additional footprint for optional features like

  Additional footprint for optional features like

  – Media

JavaFX Embedded

Porting Layer

- Only a small porting layer
  - Window Manager
  - Input (Key, Mouse, Touch)
  - Rendering (EGL/OpenGL ES)
- Most code is common with desktop

JavaFX Embedded

Features Not Planned for SE 8

- System Menu support
- Swing or SWT related nodes
- Drag and Drop support
- WebView (future release)

JavaFX Embedded

Availability

- JavaFX Embedded Developer Preview for ARM
  - http://jdk7.java.net/fxarmpreview/index.html
- JavaFX Embedded Developer Preview for ARM
  - http://jdk7.java.net/fxarmpreview/index.html
- Integrated with Java SE Embedded 8 release

Program Agenda

- JavaFX
- JavaFX Embedded
- Targeted Platforms
- Rendering Paths
- Design Considerations
- Demos

Targeted Platform

Overview

- Linux 3.x
- ARM
- And depending on the application
  - Integrated EGL/ GLESv2 support
  - Integrated touch support
  - Integrated GStreamer with hardware decoding

Targeted Platform

So many platforms to choose from...

- Many chip vendors, with different GPUs and different features (hard float, etc.)
- Many OS versions and distros
- Many “favorite” development boards

Targeted Platform

And so many EGL drivers

- Issues obtaining good hardware drivers
  - OpenGL ES2 Drivers vary in availability
  - Support for Media drivers
  - Support for Hardware Cursors
- Differing levels of community support

Targeted Platform

And we picked...

- BeagleBoard xM Rev. C1
  - TI DM3730
  - Imagination Technologies PowerVR SGX530
- OpenEmbedded Linux
- Available drivers for OpenGL ES2, Media

Targeted Platform

some of the list we are “monitoring”

- Raspberry Pi
  - Broadcom BCM2835 with VideoCore IV GPU
- Freescale I.MX53
  - Freescale ARM Cortex™-A8 with AMD Z430 GPU
- PandaBoard
  - Dual-core ARM® Cortex™-A9 with SGX540 GPU
- Kontron M2m
  - Intel Atom with GMA 500

Program Agenda

- JavaFX
- JavaFX Embedded
- Targeted Platforms
- Rendering Paths
- Design Considerations
- Demos

Three rendering paths for Embedded JavaFX

- OpenGL ES2, EGL and Frame buffer
- Software Rendering with DirectFB
- OpenGL ES2, EGL and X11 (future)

Rendering Paths

OpenGL ES2, EGL and Frame buffer

- Direct Frame buffer access
- No X11 or Window Manager overhead
- Preferred by most embedded customers
- Requires native driver support
- Fastest, lowest overhead for accelerated rendering

Rendering Paths

OpenGL ES2, EGL and Frame buffer (continued)

- We provide “window management”
- Single JavaFX Application supported with
  - Multiple Stages/Windows
  - Partial or full screen
  - Programmatic reordering, resizing of windows
- Input direct from /dev/input
  - Mouse, Keyboard, Touch

Rendering Paths

Software Rendering with DirectFB

- DirectFB to access the Frame buffer
  - Provides window management
  - Mouse/Keyboard support
- JavaFX software rendering
  - Not based on AWT
- Slower rendering compared to EGL
  - Better suited for simpler control devices

Rendering Paths

OpenGLES2, EGL and X11

- Requires X11 Server and Window manager
  - Input provided by X11 Server
  - Limitations with X11 add to overhead
    - Shaped rendering for rounded windows
- OpenGL ES2 using EGL
- Not planned for the Java SE Embedded 8 release.

Rendering Paths

Media support

- Media uses GStreamer
  - Allows using vender supplied hardware assisted codex

Program Agenda

- JavaFX
- JavaFX Embedded
- Targeted Platforms
- Rendering Paths
- Design Considerations
- Demos

Design Considerations

My desktop App should just run, right ???

- Yes, it probably will, but will it be usable ?
- Embedded platforms have
  - Slower CPU/GPU
  - Less Memory
  - Likely smaller display
  - Possible Touch screen

Design Considerations

Keep it simple if you can

- Complex Scene graphs will slow rendering
  - Enabling caching can help, but will also take extra memory
- Free up unused resources
  - Time versus memory trade off

Design Considerations

Effects

- JavaFX has a rich set of visual effects
- Effects have different performance costs, mostly depending on platform GPU

Design Considerations

Think Big

- Large images use up precious memory
  - Think about your 12 Megapixel camera !
  - Use JavaFX features to load pre-scaled images

Design Considerations

Think Small

- Smaller screens often need
  - Larger text
  - Bigger overall controls

Program Agenda

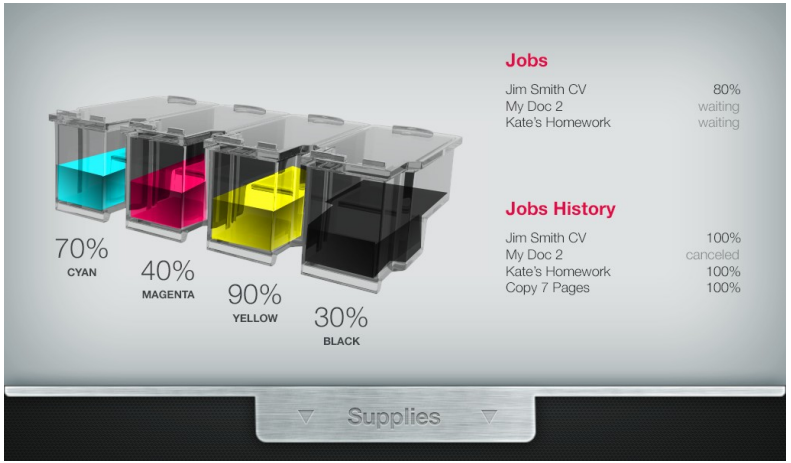
- JavaFX
- JavaFX Embedded
- Targeted Platforms
- Rendering Paths
- Design Considerations
- Demos

Demo

JavaOne Scheduler

Demo

Printer Interface



The preceding is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.


JavaOne™