JogAmp: 2D/3D & Multimedia across Devices

Siggraph 2011 - Vancouver Convention Center
August 9, 2011

Presented by: Sven Gothel
Rami Santina
What is JogAmp?

JogAmp
JOGL – JOCL – JOAL ...

2D/3D & Multimedia across Devices
Example Usage

C3D – Visual Project Control

C3D Viewer
- Model Visualization
- Project Progress Update
- 4D Animation
- Report Generation
- Design Review

C3D Studio
- Scenario Creation
- Data Integration
- Vertical Application Dev.
- ...

C3D Mobile
- Model Visualization
- OnSite progress update
- ...

http://c3d.com
Looking back: 2010 - Roadmap

- OpenMAX (A/V)
  - HW Implementation (mobile)
  - SW Implementations (desktop)
    - LIMOA - LIM OpenMAX Implementation
    - Bellagio

- Text Rendering
  - Loop/Blinn alternative, GLSL, ..

- Generic UI

- Android Binding

- Linux ARM Binding

- API Maturity (currently: 300+ unit tests)

- AWT Parenting Enhancement

- Documentation & Tutorials
About US

- Open!
- BSD License
- Java Graphics, Audio, Media & Processing
  High Performance Bindings
- One Stop Community Platform
  - SCM, Bugtracking, Build Server, Mailinglist/Forum,..
- Commercial Support
- http://jogamp.org
Why Java?

• Availability:
  • Java, OpenGL, OpenCL, OpenAL, ..
  • Multiple Vendors
    – OpenJDK / IcedTea
    – Oracle JDK
    – IBM J9, ..
    – PhoneME
    – JamVM
    – CacaoVM
    – Dalvik
  • x86, arm, ppc, sh4, ..
  • GNU/Linux, BSD, MacOSX, MS Windows, QNX
Why Java?

- Managed Code
  - Common API for
    - Windowing
    - GLContext
    - Rendering
    - SwapBuffer
  - OpenGL Pipelining / Debugging / Trace
  - Access to vast number of API / Middleware
Continuity / Usage

• Usage http://jogamp.org
  • Ardor3D
  • C3D Studio http://c3d.com
  • Elflight Engine
  • Field/Processing
  • Gephi
  • NASA Worldwind
  • ...

2D/3D & Multimedia across Devices
Deployment

- Preinstalled Bundles
  - Modularized JARs
  - Android APKs
- Online / Cached
  - Applet
    - Classical
    - JNLP
  - Webstart (JNLP)
Continuity / Maturity

● Maturity
  ● Version 1
    - JSR-231
  ● Version 2
    - OpenGL Profiles (ES 1+2, GL 2 + 3 + 4)
    - Windowing Toolkit Abstraction
    - Continuity Build/Test Server http://jogamp.org/chuck/

● Community Contributions
  ● FreeBSD Port
  ● JOAL Fixes
  ● JOCL Project
  ● Bugzilla Entries and Test Cases
  ● Code Reviews
OpenGL Profiles
2D/3D & Multimedia across Devices
Windowing Toolkits

Native Window

Native Surface

X11 (Unix)  GDI (Windows)  Android  Coco (MacOSX)  SWT (SWT Canvas)  AWT (AWT Canvas)

GLX  WGL  EGL  CGL

GL

2D/3D & Multimedia across Devices
NEWT

• Seamless integration into the platform's
  • Creation/Destruction of top level and child windows
  • Multithreaded Access to Window Surface
  • Re - Parenting
  • Decorated- and Undecorated - Windows
  • Exclusive Full screen Mode
  • Screen Mode API
  • Event handling
NEWT Requirements *(todo)*

- Transparency *(API prepared)*
- Drag & Drop *(Pending)*
  - *Enhance SWT Parenting*
  - *Enhance Android Parenting*
New Stuff...
Graph API
Resolution Independent
Shapes and Curves
Resolution Independent Curve Rendering API

• Based on Paper:
  • R Santina, “Resolution Independent NURBS Curve Rendering using Programmable Graphics Pipeline”, to be presented in GraphiCon2011.

• **NOT** Loop/Blinn

• Patent Free

• Can Render Bezier, Bsplines, NURBS
Resolution Independent Curve Rendering API

• Why?

• Resolution Independent Text Rendering
• GPU based - Fast
• Seamless integration into Renderer (Scenegraph,...)
• New User Interface – across devices

• http://jogamp.org/deployment/jogamp-current/jogl-test-applets.html
• http://www.youtube.com/watch?v=Rqsu46ifMaw

JOGL Graph API

- Outline → OutlineShapes → GLRegion
- Renderer
  - RegionRenderer
  - TextRenderer (same as RegionRenderer)
    - Helper methods for texts and fonts.

```java
outline.addVertex(x, y, z, w, onCurve);
```

```java
outlineShape.addOutline(outline);
outlineShape.addOutline(outline2);
region = GLRegion.create(outlineShape, getRenderModes());
region.render(gl, outlineShape,...);
```
JOGL Graph API

• Initializing:
  • Read Outlines (from font, svg, application, ...)
  • Modified Constrained Delaunay Triangulation
  • Generate Region

• Rendering:
  • VBO buffers
  • Realtime manipulation – weights
  • Transformation....
GPU based Resolution Independent UI

- Abstracted from the windowing toolkit
- Support multithreading
- Seamless integration into
  - A native window (HUD)
  - A custom Scenegraph (2D plane within 3D)
- High Quality rendering
- Super Fast
JOGL Graph.UI API

UIShape
- UITextShape
- RIButton
- RILabel
- UIGroup
- UITextBox
- UITextArea

Graph.curve API

UISceneController
- Add/removeShape
- GetSelected
- getActiveUI
- ...

GLEventListener
- MouseListener
UI Requirements (WIP)

• Generic UI Rendering
  • Rendering shall be performed using native rendering TKs (JOGL, ..)
  • Render primitives on an offscreen 2D plane to be
    - integrated into a custom 3D scenegraph
    - rendered as a HUD.

• Generic User Input
  • Input events should be delegated from the custom scenegraph to the UI input module.
JOGL on Embedded Devices
JOGL on Embedded Devices

- Development Env:
  - Beagleboard Devkit with ARM7I / PowerVR
    - Linux
    - Android
  - Platform based Unit tests
  - Continuous Integration with auto-builds.
  - Cross platform compilation/building
JOGL on Linux ARM
JOGL on Linux ARM

- EGL binding
- Tested with Ubuntu for embedded Devices.
- Demo!
JOGL on Android
JOGL Android Binding

• Why?
  • Short Development Cycles
  • No device specific development
  • Multitouch actions captured by Newt EDT
  • Same code compiled for all.

• Deployment:
  • adb install jogl.apk
  • adb install myFancyapplication.apk
  • Manual Daisy Chained ClassLoader.
JOGL Android Binding

• Details:
  • Enhanced EGL binding
  • Exposing GLES1 and GLES2 native profiles
  • GL2ES1 and GL2ES2 profiles for Desktop/Mobile
  • Using Android SDK/NDK
    - Requires SDK Level 9, Android 2.3 Gingerbread for NIO Surface access
  • Tested with:
    - Beagleboard with TI-rowboat gingerbread
    - Samsung Galaxy SII – Arm/Mali
    - Samsung Galaxy S
    - More soon with CI hook-up.
JOGL Android Binding

- Cross platform builds/tests with Linux host
- Scripts provided in source code repository
- NEWT Helper class (NewtActivity)
  - Android Surface / NEWT Window mapping
  - Android Input Event / NEWT translation
JOGL Android Binding

- [http://www.youtube.com/watch?v=VHxtVT4tWjM](http://www.youtube.com/watch?v=VHxtVT4tWjM)
Multitouch

- `com.jogamp.newt.MouseEvent` extended!
  - `e.getPointerCount()`
  - `e.getX(int index)`
  - `e.getY(int index)`
  - `e.getPointerId(int index)`
  - `e.getPressure(int index)`
Q&A

- What's Next?
- Why is neither Swing nor AWT recommended?
- What are the supported IDEs?
Thank You

• Sven Gothel
• Rami Santina
• Dominik aka DemoscenePassivist
• Wade Walker

• and all the many contributors!