JogAmp and Maven

http://jogamp.org/
Mark Raynsford
code@io7m.com

Maven

Maven

The JogAmp project currently distributes Jar files and .7z archives containing compiled code, source code, and documentation.

As with the majority of software projects, project artifacts must be manually downloaded and used in an IDE specific manner, or delivered over JNLP for applets.

Maven

Maven (http://maven.apache.org/) implements a package management system for Java.

Projects specify dependencies on packages in a central repository (http://search.maven.org/), and Maven automatically downloads the required artifacts and configures the build and/or runtime environment.

It's arguably the current de-facto standard for package management for Java. Other systems such as Gradle (http://gradle.org/) and Ivy (http://ant.apache.org/ivy) use Maven packages rather than create their own competing standards.

Maven

Naturally, it's the responsibility of a project to publish its own packages to the Central Repository. This can sometimes be a challenge if the project in question does not use Maven to build itself.

JogAmp

JogAmp

As of June 2012, the JogAmp project now publishes a full set of Maven packages.

Stable versions and release candidates are released to the Central Repository, and bleeding edge packages are published to a testing repository at http://jogamp.org/deployment/maven

Packages

Packages

The Maven packages produced are carefully organized for ease of use.

The addition of the dependencies shown to your project's Maven POM file is all that is required to use JOGL in a typical desktop application.

```
<dependency>
  <groupId>org.jogamp.gluegen</groupId>
  <artifactId>gluegen-rt-main</artifactId>
  <version>2.0.2</version>
</dependency>
<dependency>
  <groupId>org.jogamp.jogl</groupId>
  <artifactId>jogl-all-main</artifactId>
  <version>2.0.2</version>
</dependency>
```

Android

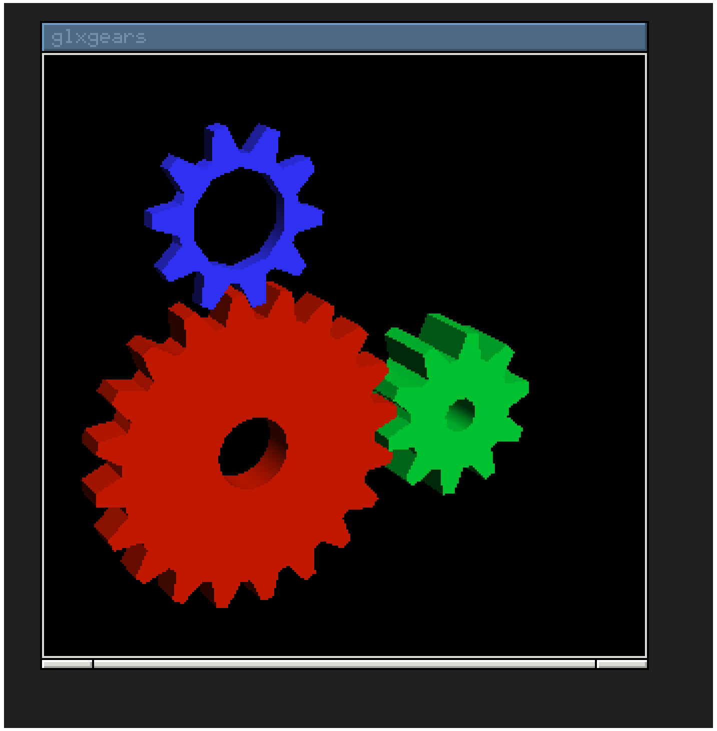
Android

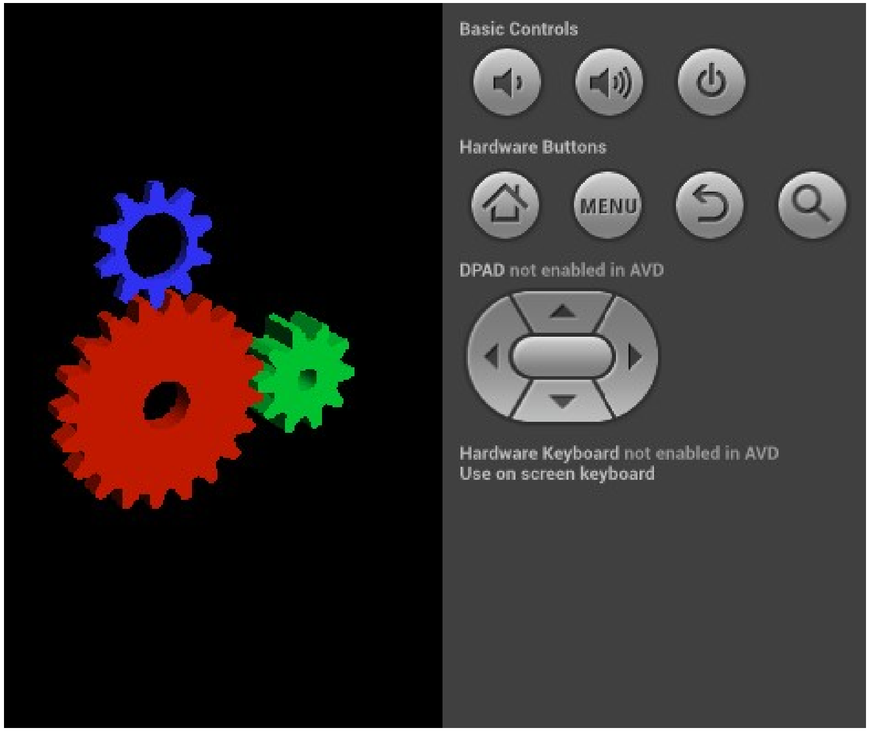
The JogAmp project provides an abstraction known as the **GLEventListener** that allows one to write applications in a renderer and display independent manner. That is, the programmer writes their program using **GLEventListener** and does not need to modify it to run on Android, or on other platforms that do not provide programming interfaces that look like typical Java programs.

Android

Writing a program that works unmodified on both traditional Java desktop/workstation systems and Android devices is simply a matter of providing trivial frontend code to the bulk of your application written using the **GLEventListener** abstraction.

The desktop code specifies dependencies on the standard JOGL packages, and the Android code specifies dependencies on the published JOGL Android packages.





Android

Beyond the initial separation of code into a desktop and Android frontend, the rest of the book-keeping and grunt work associated with creating Android apps is fully automated with the **maven-android-plugin**.

http://code.google.com/p/maven-android-plugin

Android

See the wiki for a complete example of building an application from scratch that works on both desktops and Android without modification.

http://jogamp.org/wiki/index.php/Maven_And_Android

Thanks!