
FRED GROTT

• fred.grott@gmail.com

 mobilebytes.wordpress.com



at

Application Demos

HomeScreen Demo

Install pk and hold home button down and install as temp homescreen.

[HomeZ.apk](#)

[HomeZ Video](#)

Objective

My obsession is Android Application Development, not just the look and feel going so far as to develop my own User Interface Android Application Widget library but also the development process itself and the development tools that help developers produce great android application products.

FOSS Projects

Free and Open Source projects that Fred Grott has contributed such areas as javaME(J2ME), JAAVEE(J2EE), IDEs,etc.

SynclastUI(javaME UI toolkit)

Antenna ANT Tasks for MIDP development(javaME)

Android OS and Android SDK

Eclipse MTJ(before it became Pulsar)

WURFL (**W**ireless **U**niversal **R**esource **F**ile. a **M**obile **D**evice capabilities database)

Ed Burnette

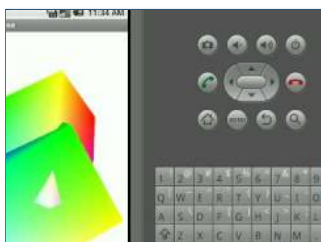
Ed Burnette has written several dev articles overing both server side development and mobile development and writes a dev column at ZDNET called **Ed Burnette's Dev Connection**. He is also a developer at SAS. **Hello,Android** is his top selling android development book, now in its 3rd edition.

Mark Murphy

Mark Murphy is the Author of [Busy Coder's Guide to Android Development](#) which is one of the top selling Android Programming books.

Several of my Android Dev techniques are finding its way into new editions of Mark Murphy's Android Books.

Sample Android Application Video



OpenGL Cube



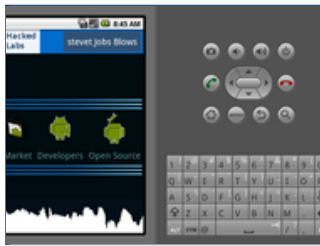
HomeZ HomeScreen Example



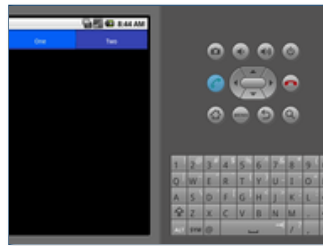
OpenGL Tunnel Effect



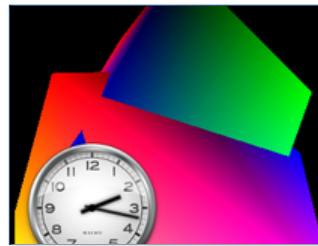
Widget UI



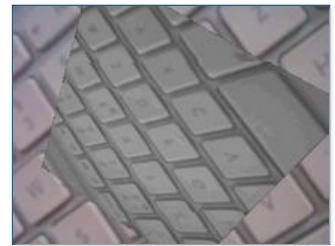
Custom Status/Title



Custom UI Components



OpenGL



Camera API

Summary

My android development and spec services can be requested by filling out one of the forms linked at:

<http://mobilebytes.wordpress.com/contract-services/hiring-a-consultant/>

Although Android device unit sales are 160,000 units per day compared to Apple's iPhone unit sales of 130,00 per day, Android OS and Android SDK are rather still rough and new. My experience in both contributing to android OS and Android SDK while building Android Java Applications gives you the added edge in implementing that look and feel you want in your android Application.

It starts out with special libraries to do such things a customized Title and Status bars. Special XML and graphics to brand such things as UI highlights as your brand and your android application. It takes Android source knowledge of Android internals, something that no Android Training course covers. My experience in contributing to Android SDK and Android OS allows me to customize any UI component for Android Application use.

It continues with behind the scenes coding to have such features as GPS, sensor orientation via the accelerometer, using SQLite for application databases or preferences, bluetooth features, camera features, WebKit embedded in the application as a webview, etc.

Server Side

Developing Server Side Applications is a balance between access control, usability, and speed of information flow across a firm's infrastructure. Over the years the differences between enterprise applications and Web applications has become smaller and less differentiated. It is no longer enough to have just exactly one tool. My Server Side Development Tools include the following: LAMP Linux(Ubuntu, Centos, gNewSense) Apache HTTPD, LightHTTPD MySQL/PostgreSQL/Apache Derby Php/Python/Perl/Ruby MVC Frameworks(CakePHP, SymfonyPHP, Django-Python, ruby on Rails) JavaEnterprise(JavaEE) SQL-MySQL, PostgreSQL, Apache Derby, SQLite Hibernate ORM Spring IoC Servers-Jboss, Apache Geronimo, Jetty, Tomcat, Glassfish MVC Frameworks-Groovy/Grails, Seam, wicket(AJX etc) Generally, its LAMP and or Groovy/Grails/Seam that is used for rapid application development(RAD) rather than the traditional java enterprise development and server stacks.

Mobile Web Development

Mobile web development is a very fragmented beast and is one of the reasons why skilled developers contribute to the WURFL project, such as myself, so that we have an accurate database of web capabilities of every device. The tool kit frameworks often used are: Google Web Toolkit(Webkit browsers and those mobile browsers that come close to full HTML5 support) PhoneGap, QuickConnect, and Rhomobile provide a different approach to mobile web RIA with the Webkit's own webview in the native language of the mobile platform becoming the container whereas the application is coded in html, CSS, and javascript. Mobile Widgets are based on W3C run-times for mobile platforms such as javaME-Symbian, Windows Mobile and some android versions namely those that are OPhone based. Mobile widgets usually consume web services through rss, atom, soap, etc. The major difference between Mobile Widgets and WebKit Webview application approach is that Mobile Widgets cannot access all native device functions. Its benefits like the Webkit WebView approach is that the most of the application code is the same for all mobile platforms that can deploy W3C Mobile Web Widgets with the widget browser based runtime providing a subset of device functions that can be accessed using javascript. The main feature of all these approaches is to get the following levels of compatibility the same on every mobile device: CSS level 1, Level 2 HTML5 Javascript and AJAX.

Android

Android Java is made up of parts of java from several places not just javaSE and parts of javaME but also some small parts of JavaEE as well. Thus, while the main thrust is JavaSE including threading, annotations, unit testing,

etc there is also or example use of simple messaging inter-application wise and application to application borrowed from JavaEE. Android like other frameworks such as Mozilla applies a XUI strategy of XML-izing the User Interface. This means you can state the UI components in XML form and also program wise state them as code instead depending upon the need. Also the android skills include how to use Google APIs such as the Google MAP API. Experienced in extending several APIs for greater functionalities such as Webkit, UI widgets, the unit testing mock object framework, etc. APIs include: Android Core SurfaceManager OpenGL Webkit MediaFramework SQLite Application Components including: Activities Services Broadcast Receivers Content Providers Intents A form of RPCs implemented using Android IDL

Apr 2008 - Present

Android Developer

Android Developer

Develop Android applications using several APIs including Multimedia, Telephony, Intents, SQLite, activity-to-activity messaging, application-to-application messaging, etc. Also developed application using OpenGL. Webkit WebView, etc.

Also, small outside contributor to the Android OS project. Thus far my contributions have been in the unit testing area of the SDK.

Purdue University

Completed course work in computer science, mathematics, physics, chemistry, etc associated with the Molecular Biology field of study

A.A.S.

CommonWealth Business College

Completed an Associates in Applied Science in the fields of Computer Science and Business with a 3.95 GPA.

