marakana

marakana

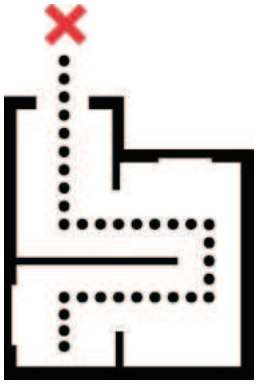
Android: A 9,000-foot Overview

Marko Gargenta
Marakana

marakana

Agenda

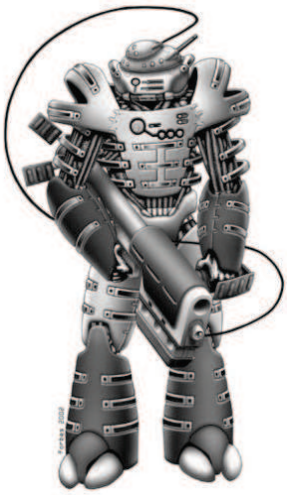
- Android History
- Android and Java
- The Stack
- Android SDK
- Hello World!
- Main Building Blocks
- Android User Interface
- Debugging
- Summary



marakana

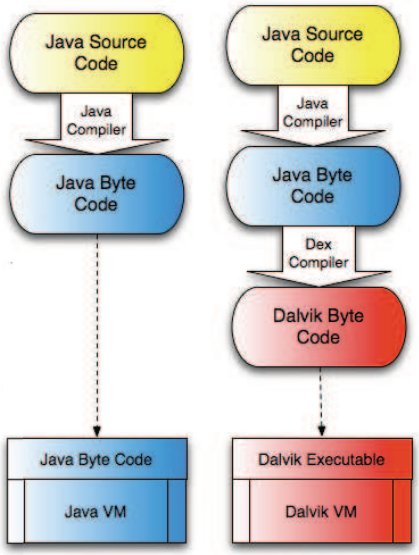
History

2005	Google buys Android, Inc. Work on Dalvik starts
2007	OHA Announced Early SDK
2008	G1 Announced SDK 1.0 Released
2009	G2 Released Cupcake, Donut, Eclair



Android and Java

Android Java =
Java SE –
AWT/Swing +
Android API



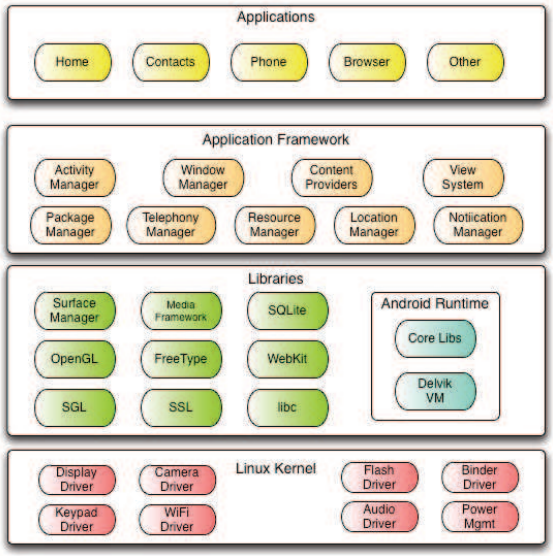
marakana

marakana

ANDROID STACK

marakana

The Stack



marakana

Linux Kernel

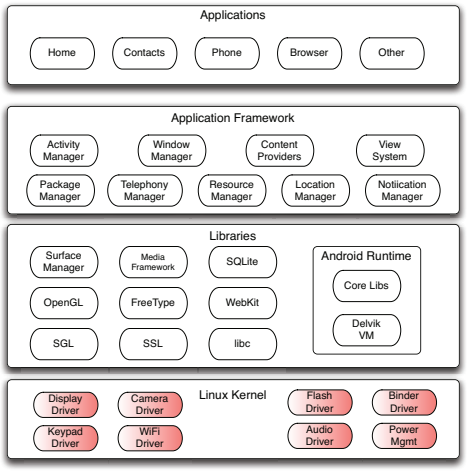
Android runs on Linux.

Linux provides as well as:

- Hardware abstraction layer
- Memory management
- Process management
- Networking

Users never see Linux sub system

The adb shell command opens Linux shell



Native Libraries

Bionic, a super fast and small GPL-based libc library optimized for embedded use

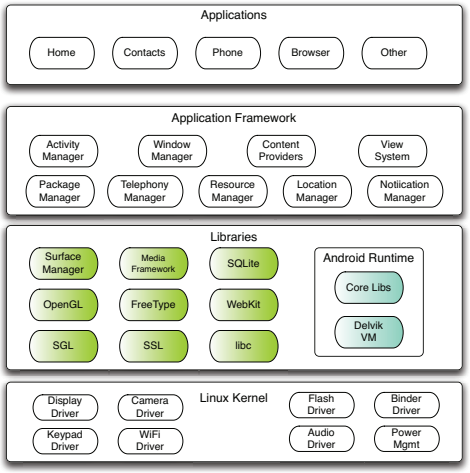
Surface Manager for composing window manager with off-screen buffering

2D and 3D graphics hardware support or software simulation

Media codecs offer support for major audio/video codecs

SQLite database

WebKit library for fast HTML rendering



marakana

Dalvik

Dalvik VM is Google's implementation of Java

Optimized for mobile devices

Key Dalvik differences:

- Register-based versus stack-based VM
- Dalvik runs .dex files
- More efficient and compact implementation
- Different set of Java libraries than SDK

Application Framework

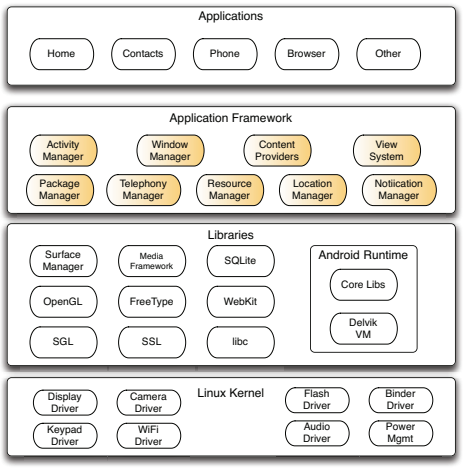
Activation manager controls the life cycle of the app

Content providers encapsulate data that is shared (e.g. contacts)

Resource manager manages everything that is not the code

Location manager figures out the location of the phone (GPS, GSM, WiFi)

Notification manager for events such as arriving messages, appointments, etc



Applications

Android SDK - What's in the box

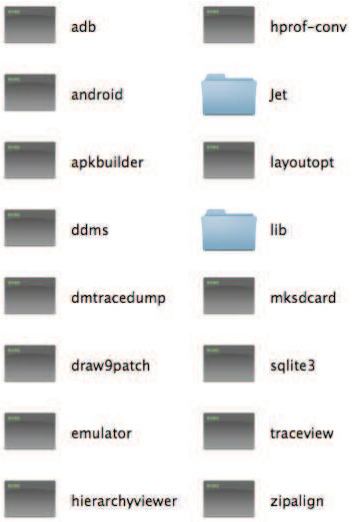
SDK

Tools
Docs
Platforms
- Data
- Skins
- Images
- Samples

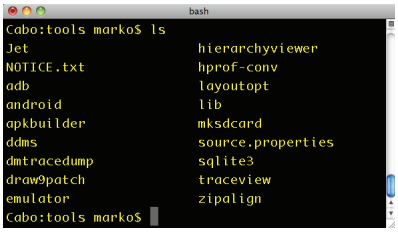
Add-ons
- Google Maps

marakana

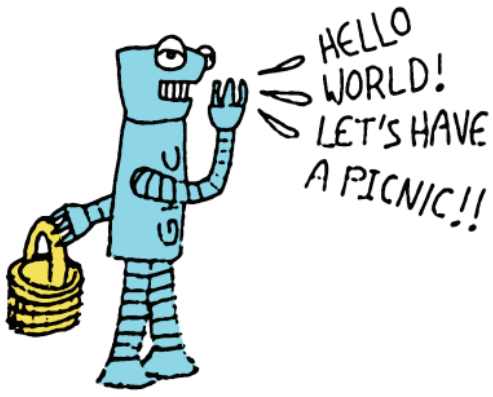
The Tools



Tools are important part of the SDK. They are available via Eclipse plugin as well as command line shell.



marakana



HELLO WORLD!

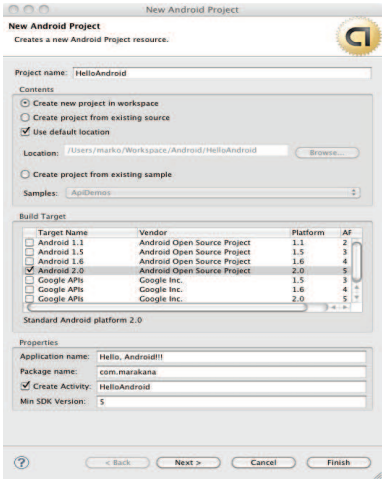
marakana

Create New Project

Use the Eclipse tool to create a new Android project.

Here are some key constructs:

Project	Eclipse construct
Target	minimum to run
App name	whatever
Package	Java package
Activity	Java class



The Manifest File

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.marakana"
    android:versionCode="1"
    android:versionName="1.0">
  <application android:icon="@drawable/icon"
        android:label="@string/app_name">
    <activity android:name=".HelloAndroid"
          android:label="@string/app_name">
      <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
      </intent-filter>
    </activity>
  </application>
  <uses-sdk android:minSdkVersion="5" />
</manifest>
```

marakana

The Layout Resource

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >
<TextView
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="@string/hello"
    />
</LinearLayout>
```

The Java File

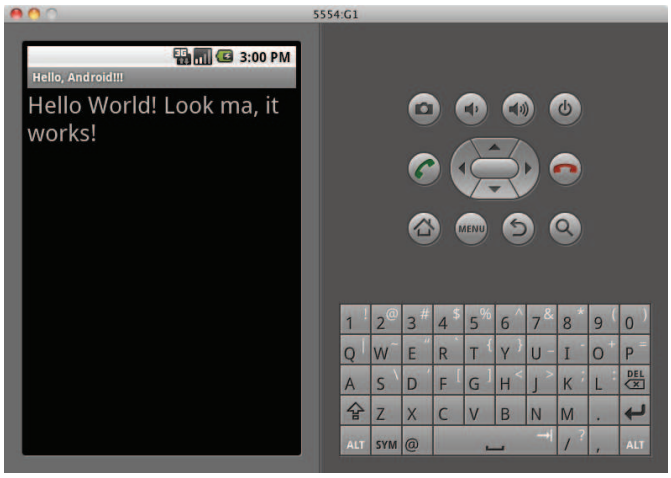
```
package com.marakana;

import android.app.Activity;
import android.os.Bundle;

public class HelloAndroid extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
    }
}
```

marakana

Running on Emulator



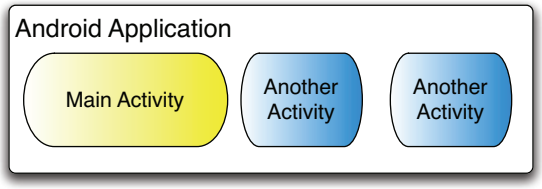
marakana

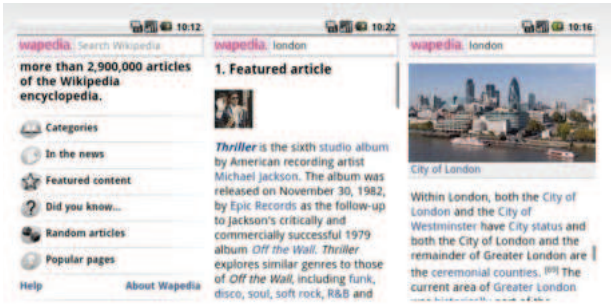
MAIN BUILDING BLOCKS

marakana

Activities

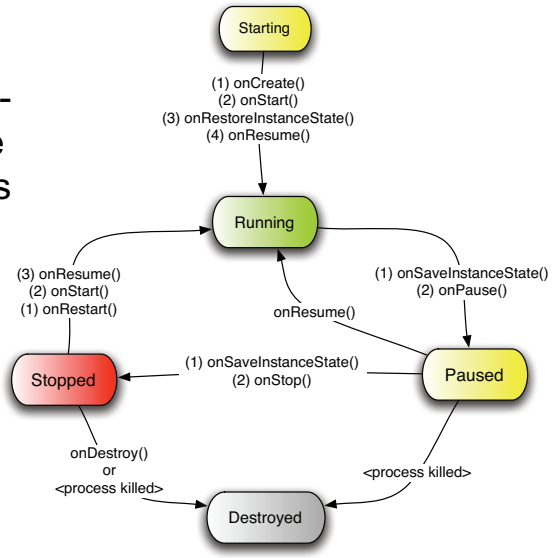
Activity is to an application what a web page is to a website. Sort of.





Activity Lifecycle

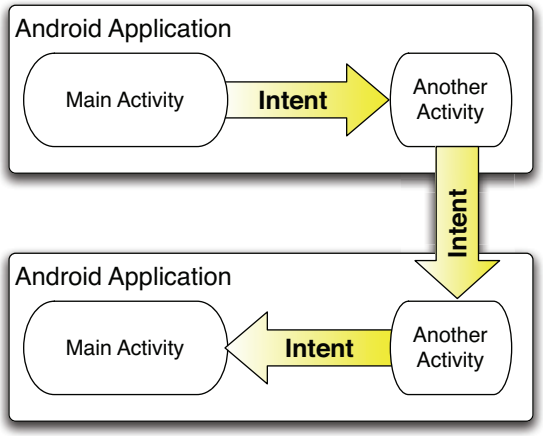
Activities have a well-defined lifecycle. The Android OS manages your activity by changing its state. You fill in the blanks.



marakana

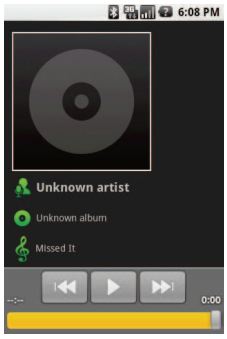
Intents

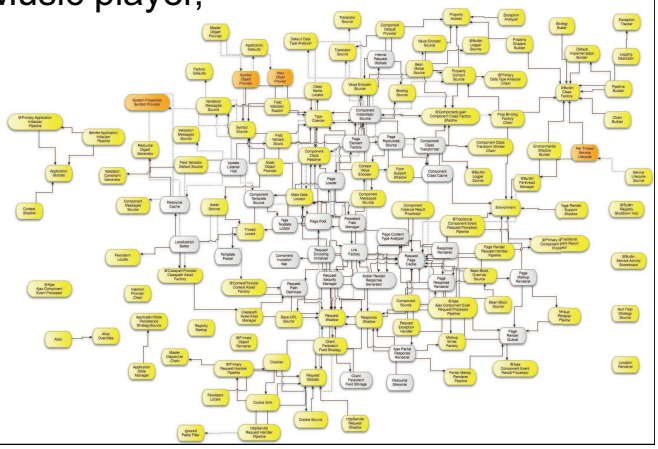
Intents are to Android apps what hyperlinks are to websites. They can be implicit and explicit. Sort of like absolute and relative links.



Services

A service is something that can be started and stopped. It doesn't have UI. It is typically managed by an activity. Music player, for example

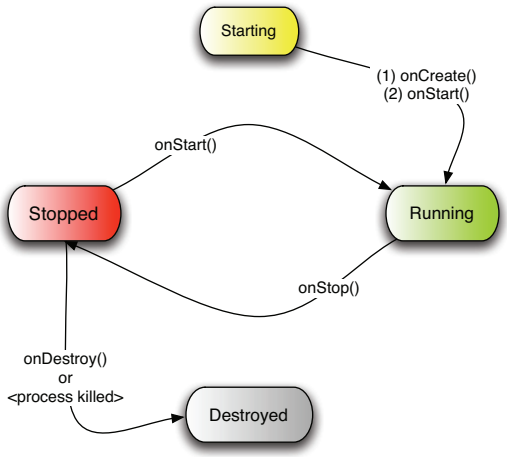




marakana

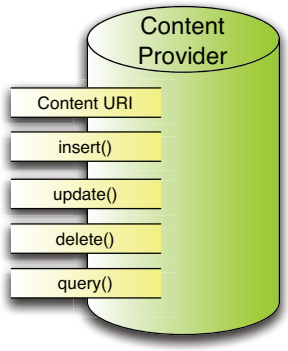
Service Lifecycle

Service also has a lifecycle, but it's much simpler than activity's. An activity typically starts and stops a service to do some work for it in the background. Such as play music, check for new tweets, etc.



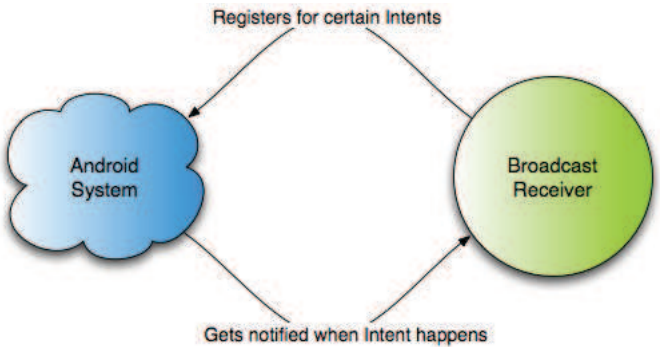
Content Providers

Content Providers share content with applications across application boundaries.
Examples of built-in Content Providers are: Contacts, MediaStore, Settings and more.



marakana

Broadcast Receivers



An Intent-based publish-subscribe mechanism. Great for listening system events such as SMS messages.

marakana

ANDROID USER INTERFACE

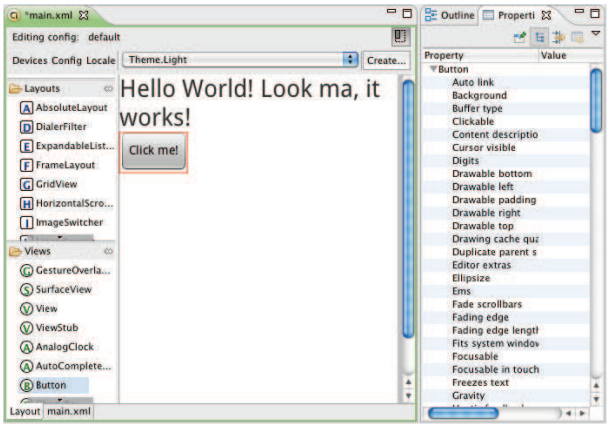
marakana

Two UI Approaches

Procedural	Declarative
You write Java code Similar to Swing or AWT	You write XML code Similar to HTML of a web page

You can mix and match both styles. Declarative is preferred: easier and more tools

XML-Based User Interface



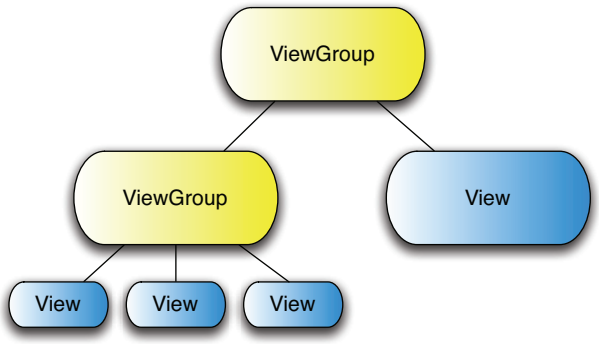
Use WYSIWYG tools to build powerful XML-based UI. Easily customize it from Java. Separate concerns.

marakana

Dips and Sps

px (pixel)	Dots on the screen
in (inches)	Size as measured by a ruler
mm (millimeters)	Size as measured by a ruler
pt (points)	1/72 of an inch
dp (density-independent pixel)	Abstract unit. On screen with 160dpi, 1dp=1px
dip	synonym for dp and often used by Google
sp	Similar to dp but also scaled by users font size preference

Views and Layouts



ViewGroups contain other Views but are also Views themselves.

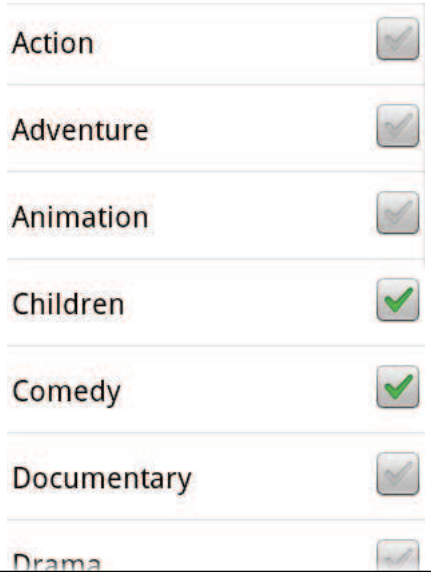
marakana

Common UI Components

Android UI includes many common modern UI widgets, such as Buttons, Tabs, Progress Bars, Date and Time Pickers, etc.

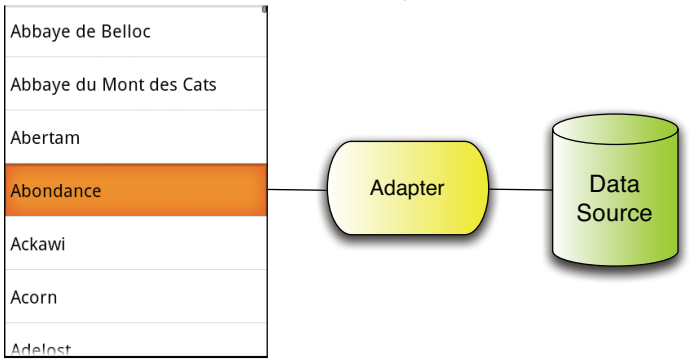
Selection Components

Some UI widgets may be linked to zillions of pieces of data. Examples are ListView and Spinners (pull-downs).



marakana

Adapters



To make sure they run smoothly, Android uses Adapters to connect them to their data sources. A typical data source is an Array or a Database.

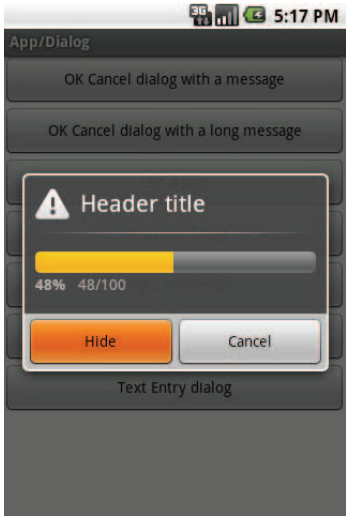
marakana

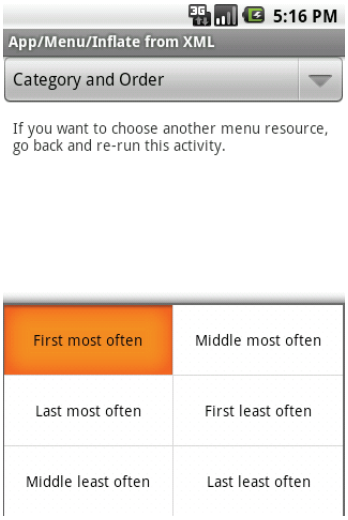
Complex Components

Certain high-level components are simply available just like Views. Adding a Map or a Video to your application is almost like adding a Button or a piece of text.

marakana

Menus and Dialogs

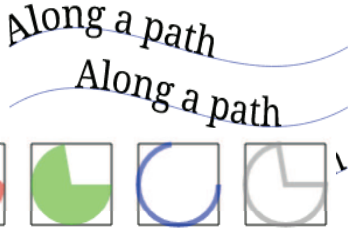




Graphics & Animation

Android has rich support for 2D graphics.
You can draw & animate from XML.
You can use OpenGL for 3D graphics.

Custom



marakana

Multimedia

AudioPlayer lets you simply specify the audio resource and play it.

VideoView is a View that you can drop anywhere in your activity, point to a video file and play it.

XML:

```
<VideoView
   android:id="@+id/video"
   android:layout_width="fill_parent"
   android:layout_height="fill_parent"
   android:layout_gravity="center" />
```

Java:

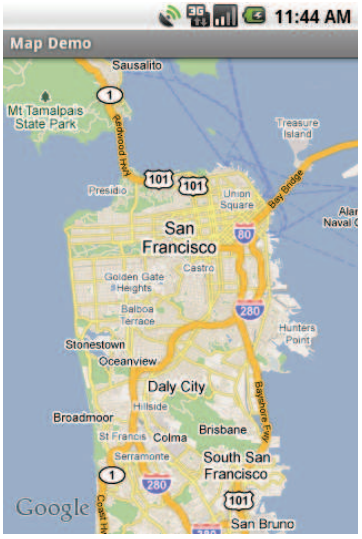
```
player = (VideoView) findViewById(R.id.video);
player.setVideoPath("/sdcard/samplevideo.3gp");
player.start();
```

marakana

Google Maps

Google Maps is an add-on in Android. It is not part of open-source project.

However, adding Maps is relatively easy using **MapView**.



XML:

```
<com.google.android.maps.MapView
android:id="@+id/map"
android:clickable="true"
android:layout_width="fill_parent"
android:layout_height="fill_parent"
android:apiKey="0EfLSgdSCWIN…A"
/>
```

marakana

DEBUGGING ANDROID APPS

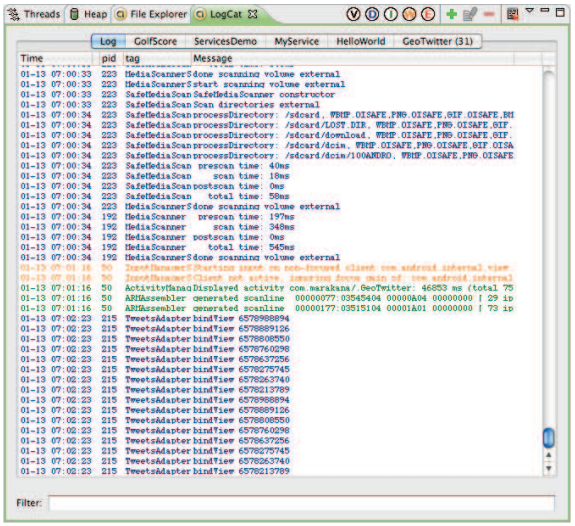
marakana

LogCat

The universal, most versatile way to track what is going on in your app.

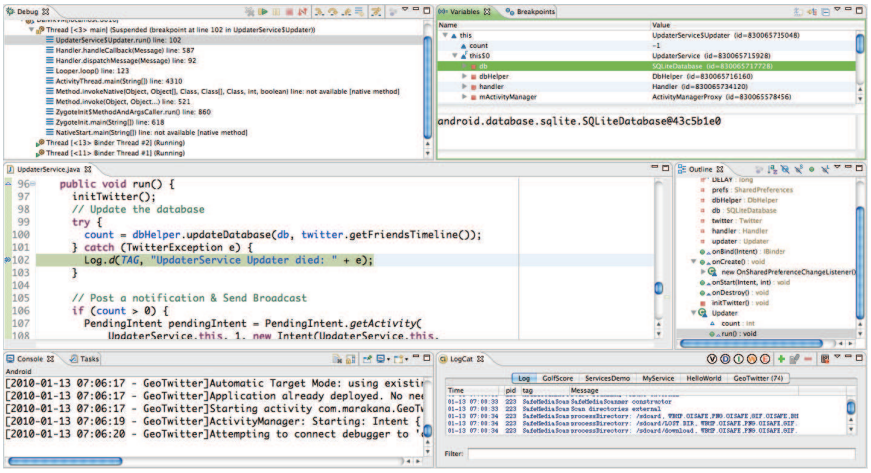
Can be viewed via command line or Eclipse.

Logs can be generated both from SDK Java code, or low-level C code via Bionic libc extension.



marakana

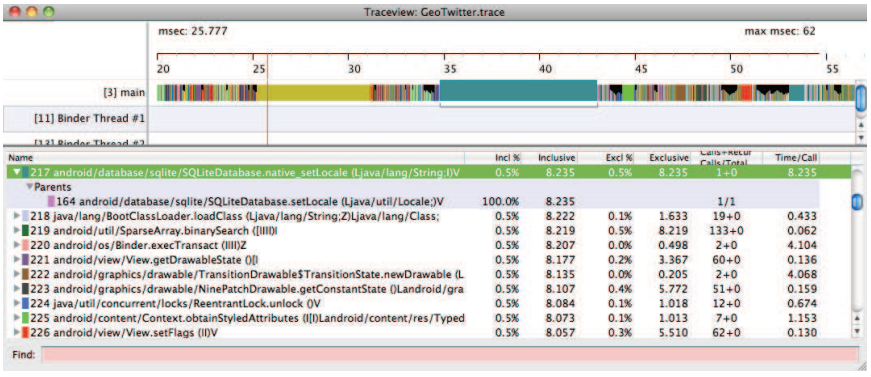
Debugger



Your standard debugger is included in SDK, with all the usual bells & whistles.

marakana

TraceView



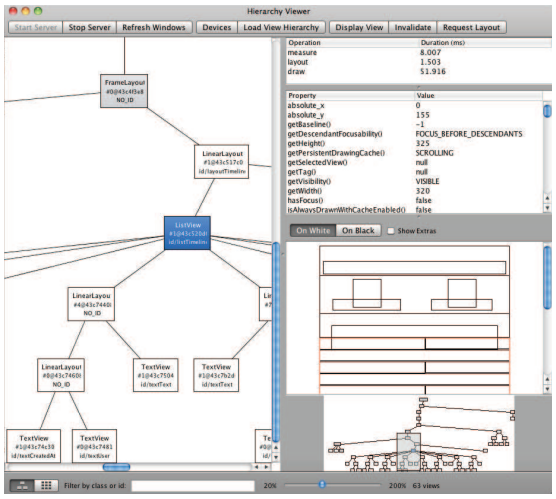
TraceView helps you profile you application and find bottlenecks. It shows execution of various calls through the entire stack. You can zoom into specific calls.

marakana

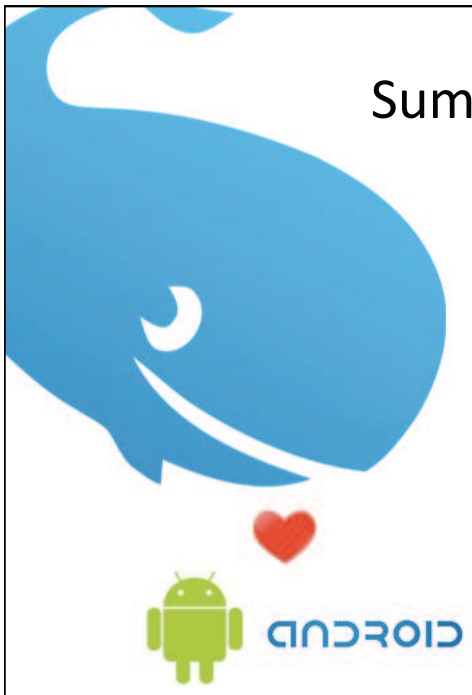
Hierarchy Viewer

Hierarchy Viewer helps you analyze your User Interface.

Base UI tends to be the most “expensive” part of your application, this tool is very useful.



Summary



Android is based on Java.

You write Java code, but technically don’t run it.

Java is augmented with XML, mostly for UI purposes.

Android Java is mostly based on Java SE with replacement UI libraries.

Marko Gargenta, Marakana.com
marko@marakana.com
+1-415-647-7000