



open handset alliance

<http://www.android.com/>

# Mobile development

## Android overview

## Globally by the end of 2010 there will be 5.1 billion mobile subscriptions

That represents about 2 mobile subscriptions for every 3 people in the world

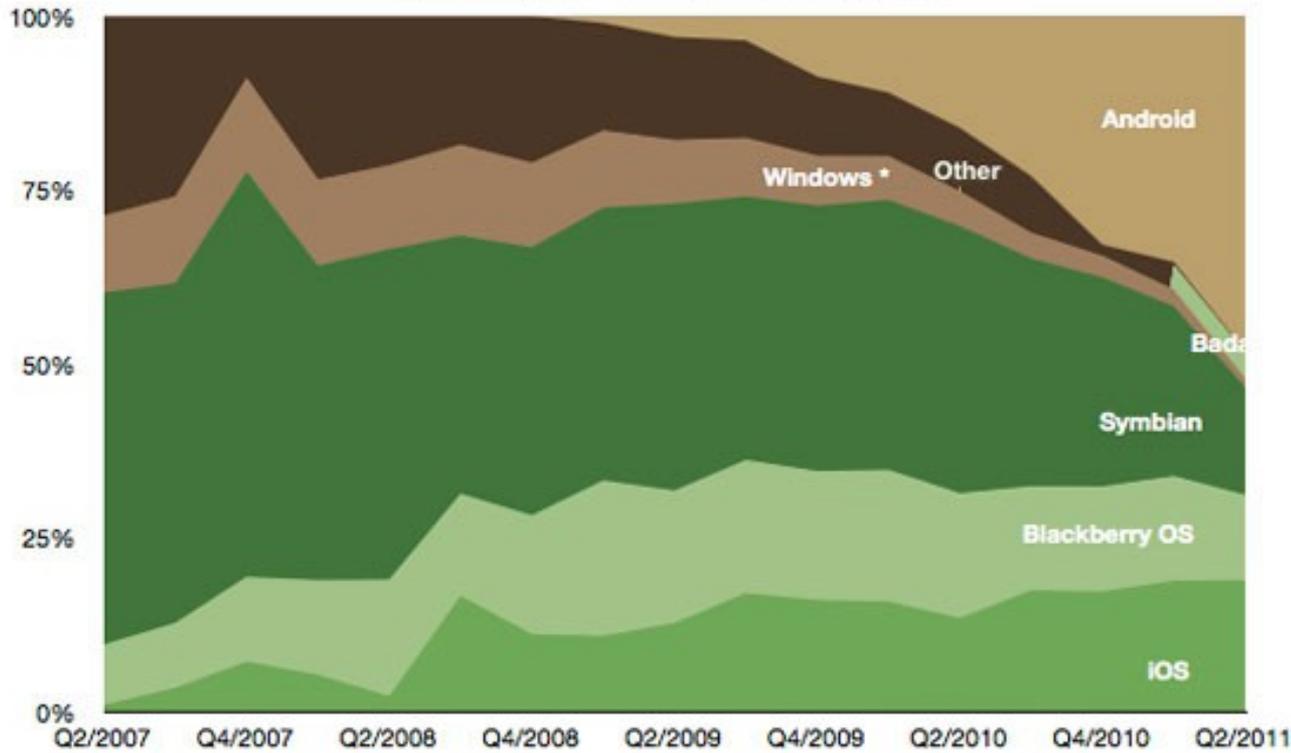
Vendor	2010 Unit Shipments	2010 Market Share	2009 Unit Shipments	2009 Market Share	Year-over-year Change
Nokia	453.0	32.6%	431.8	36.9%	4.9%
Samsung	280.2	20.2%	227.2	19.4%	23.3%
LG Electronics	116.7	8.4%	117.9	10.1%	-1.0%
ZTE	51.8	3.7%	26.7	2.3%	94.0%
Apple	47.5	3.4%	25.1	2.1%	89.2%
Others	439.4	31.6%	342.9	29.3%	28.1%
Total	1388.6	100.0%	1171.6	100.0%	18.5%

Total PC shipments in 2010, 346.2 million

In Q4 2010 it shipped 101 million smart phones

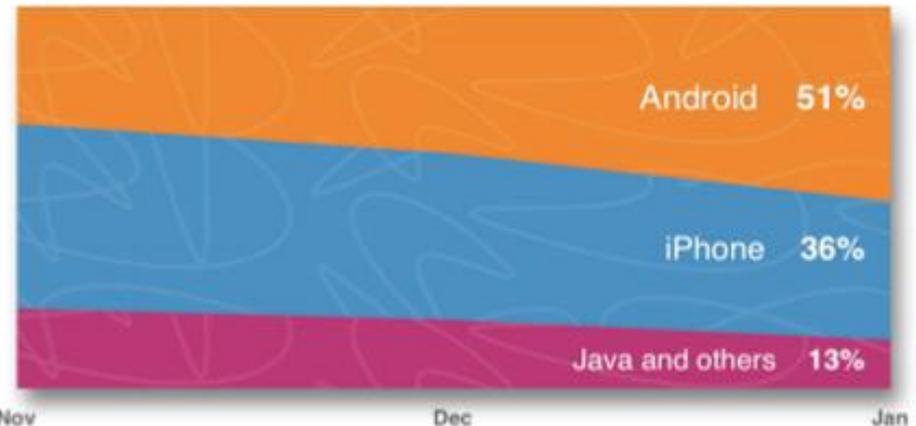
All shipments numbers are in million units  
according to International Data Corporation (IDC)

### Market shares of smartphone Platforms



Source: asymco.com

**Widespace mobile ad network**  
mobile platform share, Sweden, 2011



According to the CEO of the Swedish operator Telia Sonera  
In quarter 4 - 2010 were 90% of all mobile phones sold a smartphone!

<http://www.dagensmedia.se/nyheter/dig/article3081018.ece>

# Today's mobile "tele-puters"

- Sophisticated
  - Have capacity and functionality as a desktop computer (view the Mozilla Seabird movie or Motorola ATRIX docks promo)
  - GPS and other sensor devices built in, voice recognition...
  - Low cost unlimited data-plan
  - TV (dvh-h/mbms, dmb/dab)
  - Pay services (NFC)
- Examples
  - 3<sup>rd</sup> party applications (apps)
  - Cloud computing services
  - Advanced communications
  - OpenGL ES etc.
  - Location Based Services
  - VoIP, speech to text, ...



# Examples - LBS



**GAMING**  
Location-aware  
Interactive Gaming



**NAVIGATION & POINTS OF INTEREST**

Turn-by-turn Navigation  
City Guides  
Mobile Yellow Pages  
Traffic Reroute  
Weather



**PERSONAL SECURITY**  
Roadside Assistance  
Child Finders  
GeoFencing  
Protection of High-risk Personnel



**PEER-TO-PEER**  
Buddy Groups  
Dating  
Geo-marked Photo Sharing



**ENTERPRISE**  
Fleet Management  
Asset Monitoring  
Productivity



**COMMERCE**  
Mobile Coupons  
Customer Service



# Google™

## Introducing Google Latitude



Fred wants to hang out with his friends, and checks to see where they are.

[Learn more about Google Latitude](#)

[Watch a video](#)

## See where your friends are in real time!

Enjoy Google Latitude on your phone, computer, or both.

### Start using it on your phone

See your friends' locations and status messages and share yours with them.

Enter your number or visit [google.com/latitude](http://google.com/latitude) on your mobile web browser.

Send a link to my phone

Sweden ▾

▶ [Will it work with my phone?](#)

### View it on your computer

See your friends' locations and status messages on a full screen even without a compatible phone or data plan.

[Add Latitude to iGoogle »](#)



This service is free from Google; carrier charges may apply.

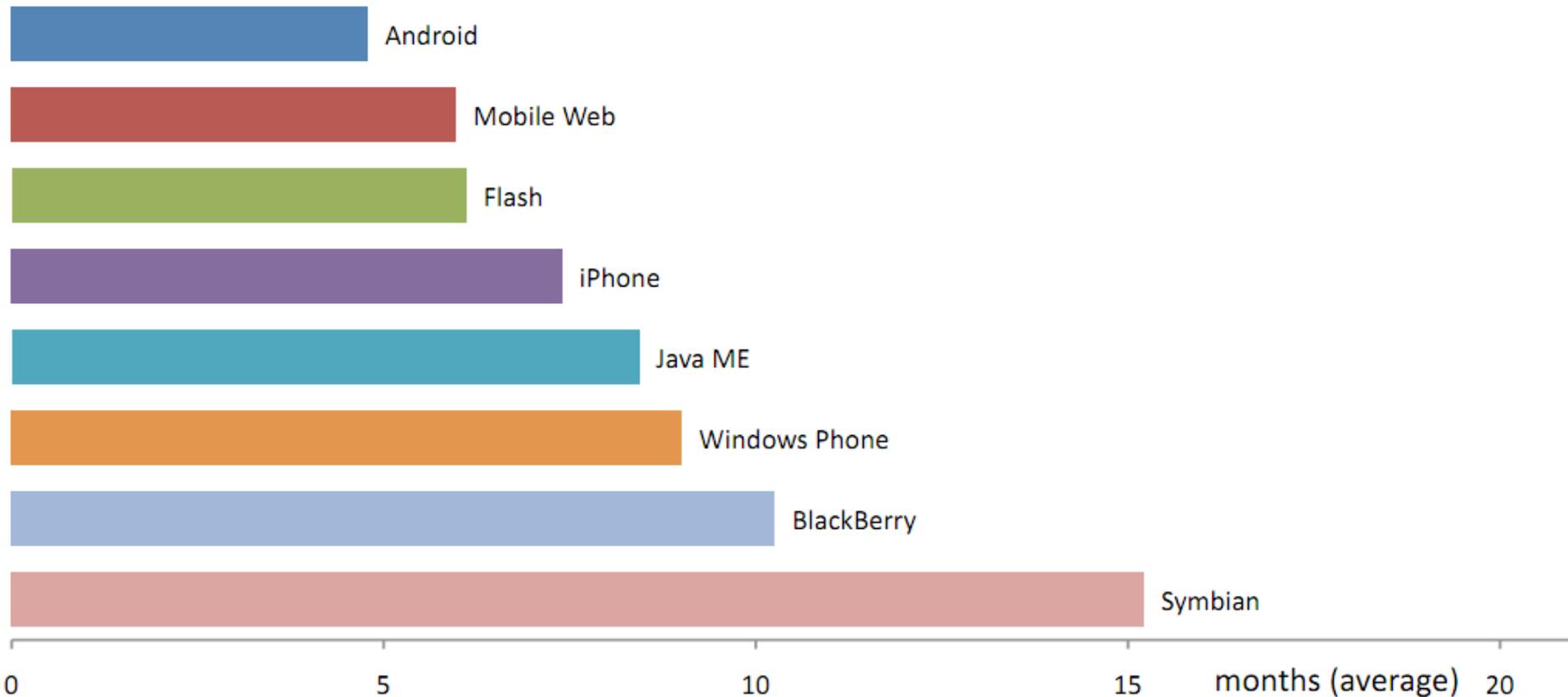
# Mobile development 1

- Others
  - Java(2) ME
  - Microbrowser based
  - Flash (Lite)
  - Python
  - BREW

Platform	Language(s)	Remarks
Android	Java, C, C++	Open Source OS (based on Linux) <a href="http://developer.android.com">developer.android.com</a>
bada	C, C++	Samsung's mobile platform running on Linux or RealTime OS <a href="http://developer.bada.com">developer.bada.com</a>
BlackBerry	Java, Web Apps	Java ME compatible, extensions enable tighter integration <a href="http://blackberry.com/developers">blackberry.com/developers</a>
BlackBerry Tablet OS (QNX)	ActionScript, C++, HTML, CSS, JavaScript	Java announced <a href="http://blackberry.com/developers">blackberry.com/developers</a>
iOS	Objective-C, C	Requires Apple Developer Account <a href="http://developer.apple.com/iphone">developer.apple.com/iphone</a>
MeeGo	Qt, C++, others	Intel and Nokia guided open source OS (will be replaced by Tizen) <a href="http://meego.com/developers">meego.com/developers</a>
Symbian	C, C++, Java, Qt, Web Apps, others	Currently the longest running of all smartphone OSs <a href="http://www.forum.nokia.com/symbian">www.forum.nokia.com/symbian</a>
webOS	HTML, CSS, JavaScript, C	Supports widget style programming, (based on Linux), probably dead since it has been abandoned by HP <a href="http://developer.palm.com">developer.palm.com</a>
Windows Mobile	C#, C	.NET CF or Windows Mobile API, most devices ship with Java ME compatible JVM <a href="http://developer.windowsmobile.com">developer.windowsmobile.com</a>
Windows Phone	C#, VB.NET	Silverlight, XNA frameworks <a href="http://create.msdn.com">create.msdn.com</a>

# Mobile development 2

Average time required to master each platform



Source: Mobile Developer Economics 2010 and Beyond. Produced by VisionMobile. Sponsored by Telefonica Developer Communities. June 2010. Licensed under Creative Commons Attribution 3.0 License. Any use or remix of this work must retain this notice.

# Mobile Dev. 3

[http://en.wikipedia.org/wiki/Mobile\\_programming](http://en.wikipedia.org/wiki/Mobile_programming)  
for even more info

**A very good read: "Mobile Developers Guide To The Galaxy"**

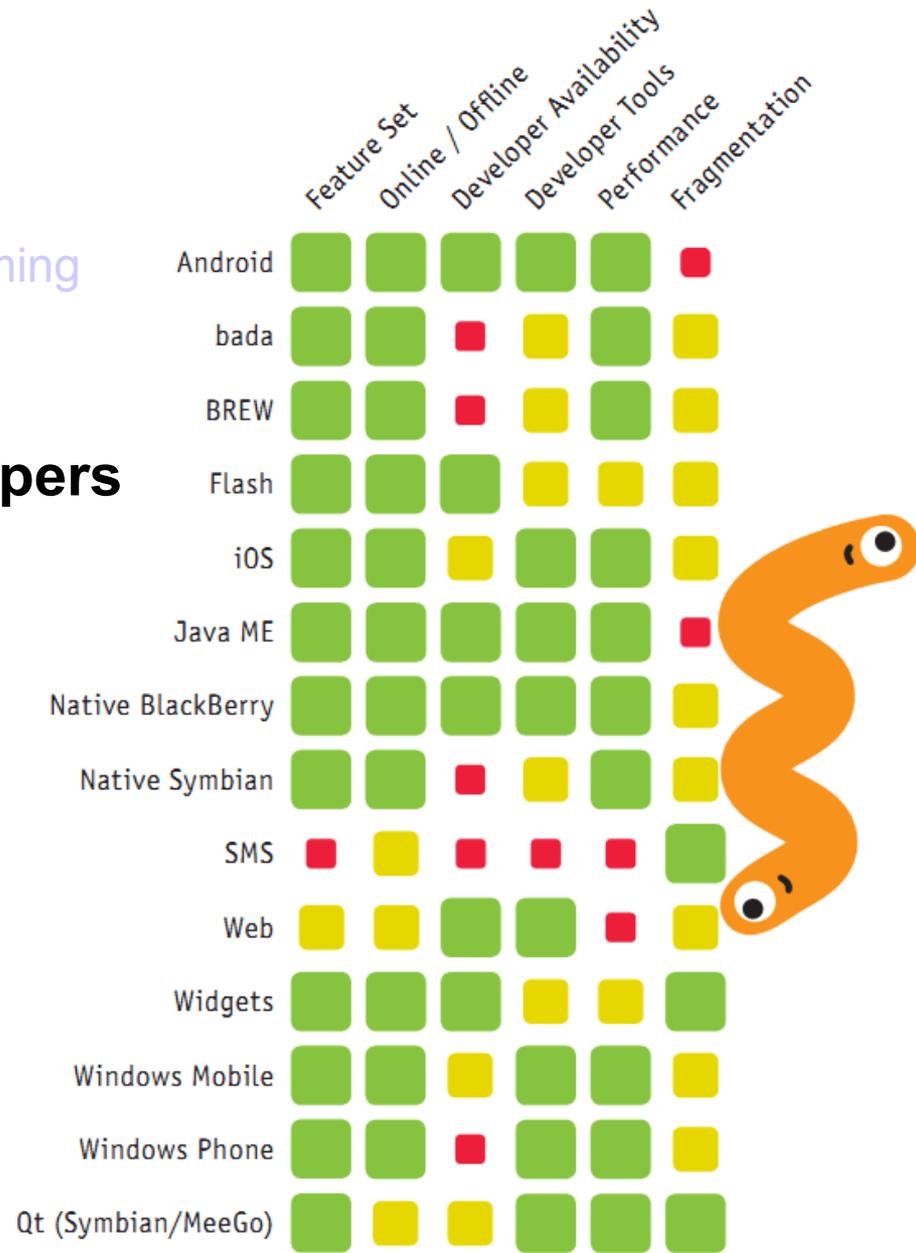
<http://www.enough.de/products/mobile-developers-guide/>

Android compared to other development environments and platforms

**Missing platforms**

HP - webOS

... ?



Green indicates good coverage or support, yellow for limited and red for bad coverage of the respective topic

# What is Android?



- A free, open source mobile platform framework
- Android is not a device, or a product, or limited to phones
  - A hardware reference design describing the minimum requirements to support the stack
- Android brings Internet-style innovation and openness to mobile phones
- Contains a set of pre-installed key applications (a fully functioning smart phone)
- Software Development Kit (tools, plug-ins, and documentation)
- The online Android Dev Guide
  - <http://developer.android.com/guide/>

# Android history



- July 2005, Google buy Android Inc. of Andy Rubin
- Jan 2007, Apple release Iphone with the help of Google
- Nov 2007, Open Handset Alliance is created
- Oct 2008, Android is open source (Apache license)
- Apr 2009, Android 1.5 (API level 3 ) Cupcake is **released**
- Sep 2009, Android 1.6 (API level 4) Donut is ...
- Jan 2010, Android 2.1 (API level 7) Eclair is ...
- May 2010, Android 2.2 (API level 8) Froyo is ...
- Dec 2010, Android 2.3 (API level 9) Gingerbread is ...
- Feb 2011, Android 3.0 (API level 11) Honeycomb is ...
- Oct 2011, Android 4.0 (API level 14) Ice Cream Sandwich is released

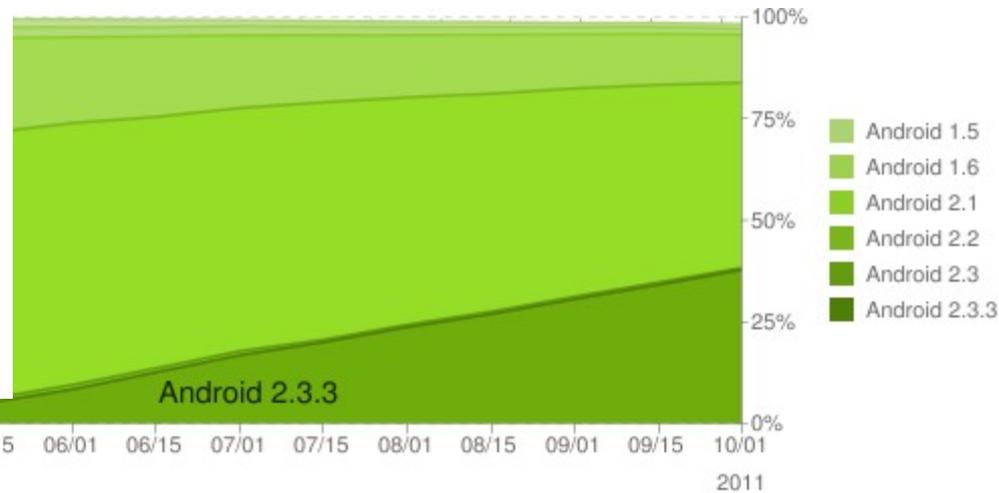
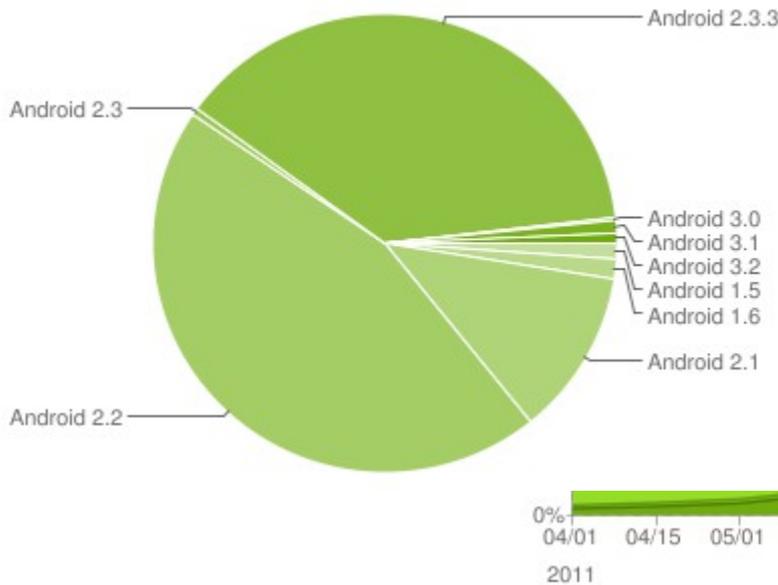
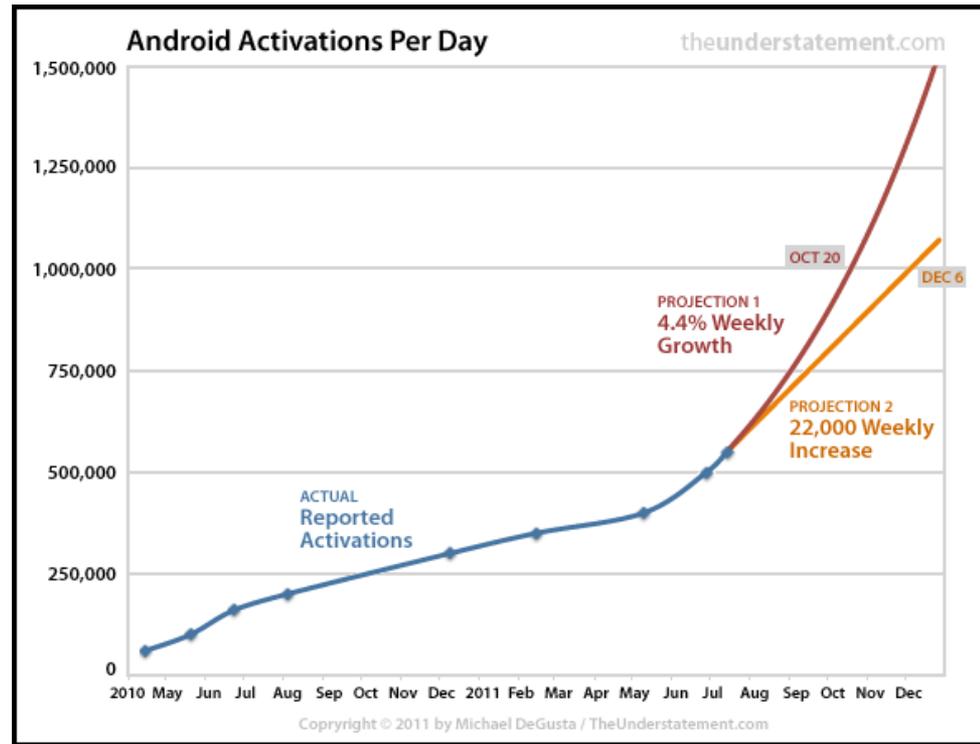
# Android tablets



# Android version distribution and activations

There is around 190 million Android devices running in the world just now – oct. 2011

It cost \$25 (one time fee) to publish Apps on the Android Market



<http://developer.android.com/resources/dashboard/platform-versions.html>

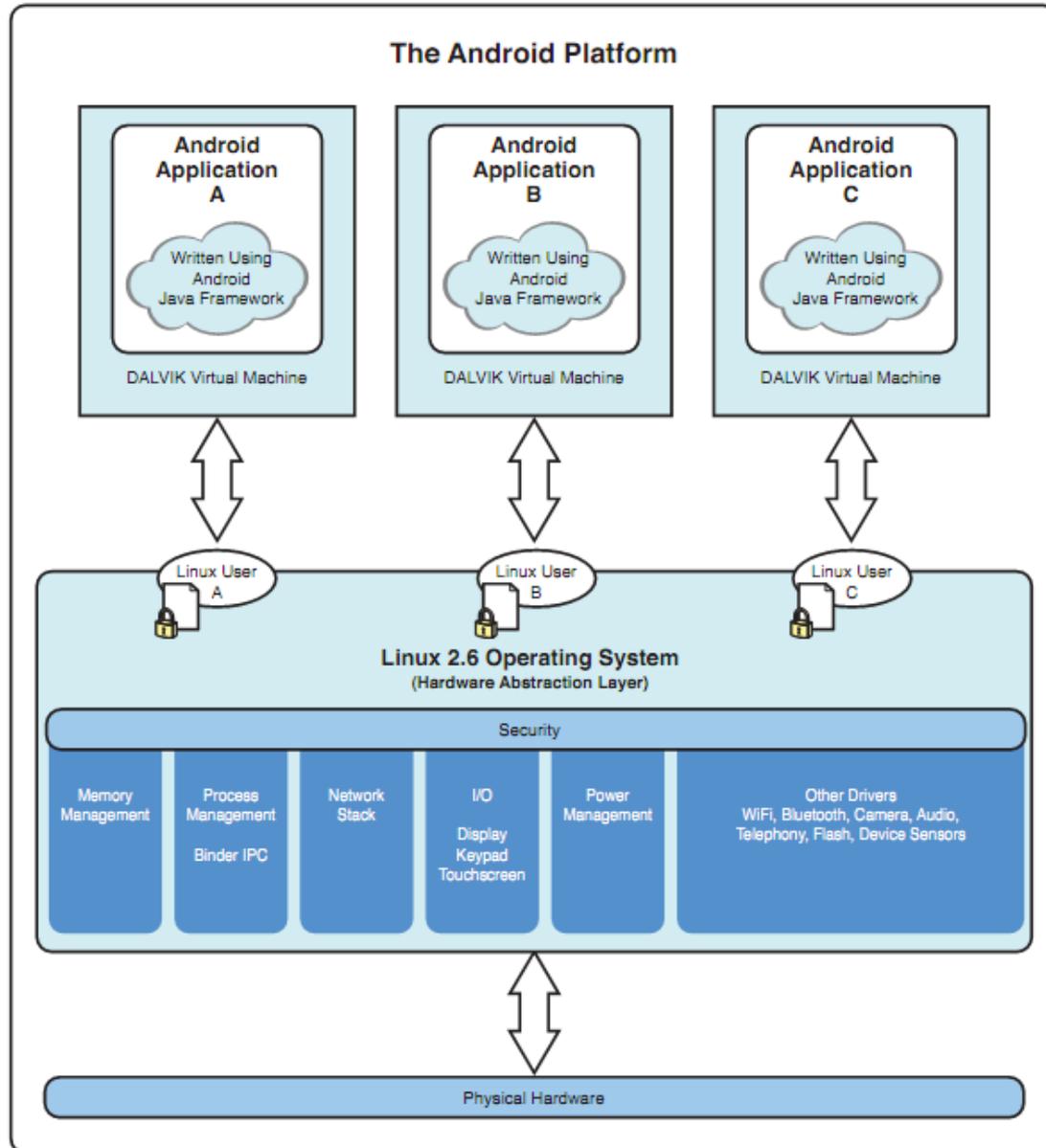
# Android operating system 1



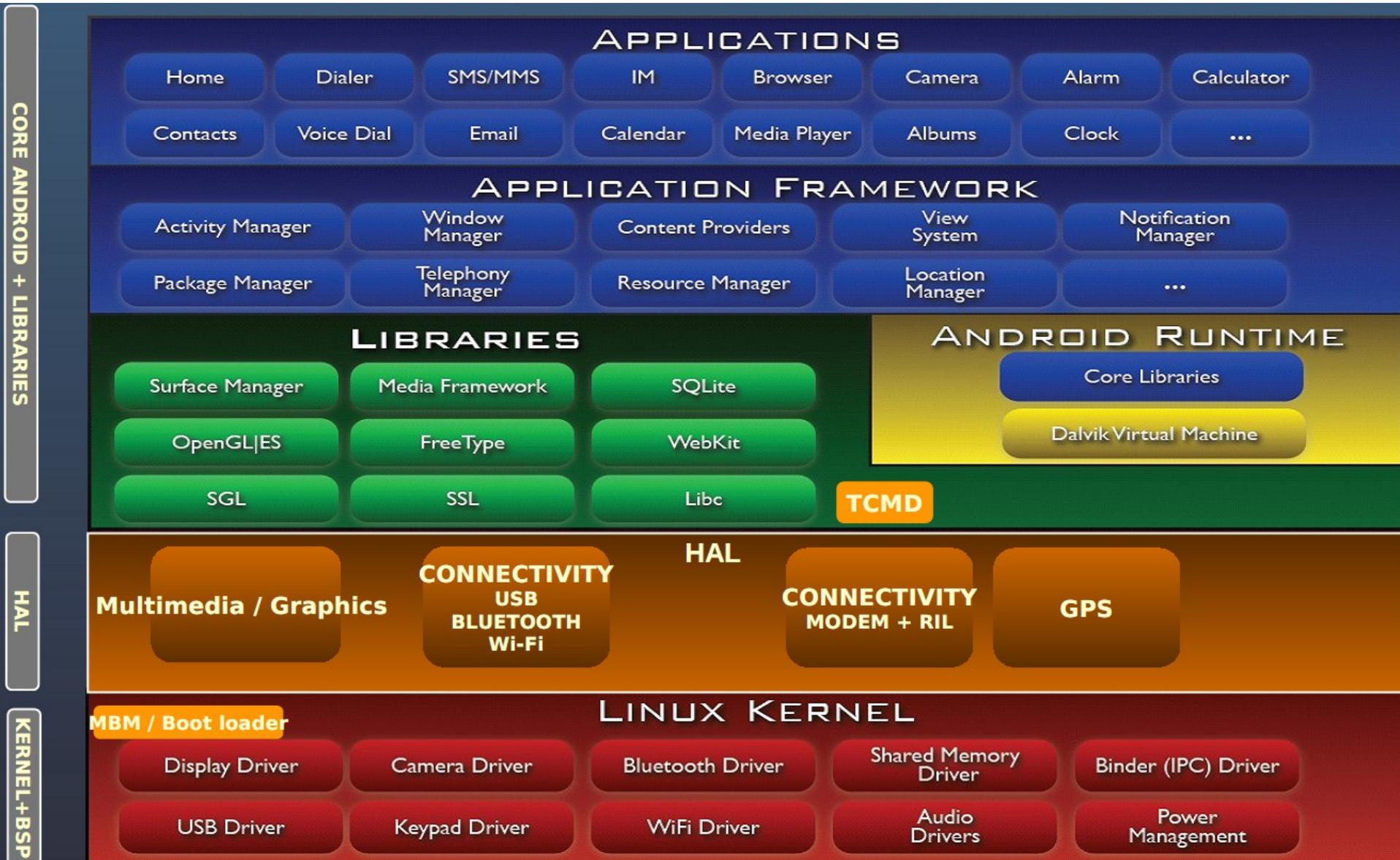
- Android is a Linux based, multiprocess, multithreaded OS
  - By default each App runs in its own isolated process with its own Java Virtual Machine
  - Each App is assigned a unique Linux user ID
  - Apps (tasks) can share the same user ID to see each other's files and share the same VM
- Open source libraries
- Android run time environment
  - The DVM (Dalvik Virtual Machine) uses APK (Android Package) files which are created in the DEX (Dalvik Executable) binary format
  - Register based instead of stack based VM with JIT support
- An application framework SDK with a Java 5 SE programming interface
  - Android NDK for native code (C/C++) via JNI (Java Native Interface)
- Some pre-installed key applications

<http://developer.android.com/videos>

# Android application model



# Android stack (architecture)



# Android operating system 2



- Android-supported hardware shares some common features due to the nature of the operating system
  - Phones, tablets, televisions, cars, cameras, ...?
- The Android OS is organized into the following images
  - Bootloader - Initiates loading of the boot image during startup
  - Boot image - Kernel and RAMdisk (root file system tree)
  - System image - Android operating system platform and apps
  - Data image - User data saved across power cycles
  - Recovery image - Files used for rebuilding or updating the system
  - Radio image - Files of the radio stack
- These images are stored on nonvolatile flash memory, so they are protected when the device powers down. The flash memory is used like read-only memory (hence, some call it ROM), but can it be rewritten as necessary. For example, with FOTA (Firmware Over-The-Air) Android operating system updates

# "Old" generation phones



- A couple of popular phones, note that the ROM (Flash) is divided into several partitions (images) mounted in the RAMdisk tree
  - System, data, etc.

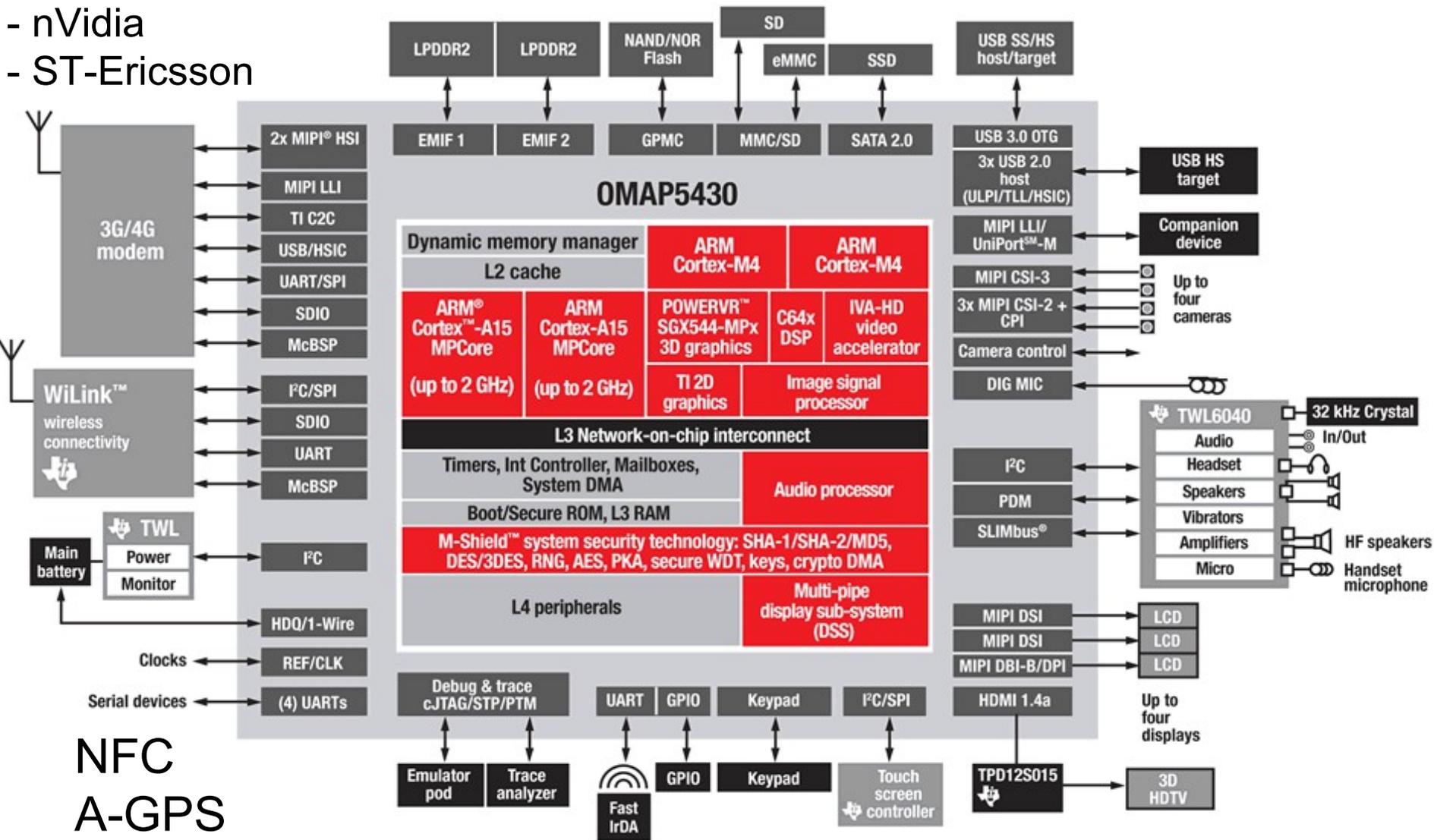
Model	MPU	RAM/ ROM	Screen	Other Features
Samsung Galaxy S Pro (August 2010)	1-GHz Samsung Hummingbird	512MB/ 2GB	AMOLED 480x800 hdpi	CDMA/1xEV-DO, 802.16, FM radio slide out keyboard BT3.0, 802.11b/g/n, 5-MP camera 0.3MP front-facing camera, AGPS
Nexus One / HTC Passion (January 2010)	1-GHz QCOM Snapdragon	512MB/ 512MB	AMOLED 480x800 hdpi	GSM/UMTS Trackball, dual microphones BT2.0, 802.11a/b/g/n, 5-MP camera AGPS, geotagging

At least 5 MF of SoC

- Samsung
- Qualcomm
- Texas Instruments
- nVidia
- ST-Ericsson

"Current" generation of system-on-chip (SoC) boards

# TI OMAP5430 SoC



# User input methods



- Touch screens
  - **Resistive** - Two resistive material layers sit on top of a glass screen. When a finger, stylus, or any object applies pressure, the two layers touch together and the location of the touch can be determined
  - **Capacitive** - A charged material layer is overlaid on a glass screen. When a finger or any conductive object touches the layer, some charge is drawn off, changing the capacitance, which is measured to determine the location of the touch.
  - **Surface Acoustic Wave** - This uses a more advanced method that sends and receives ultrasonic waves
- Alternative methods
  - **D-pad** (directional pad) - An up-down-right-left type of joystick
  - **Trackball** - A rolling ball acting as a pointing device that is similar to a mouse
  - **Trackpad** - A special rectangular surface acting as a pointing device
  - **Keyboard** - As it sounds, a minimal keyboard for phones

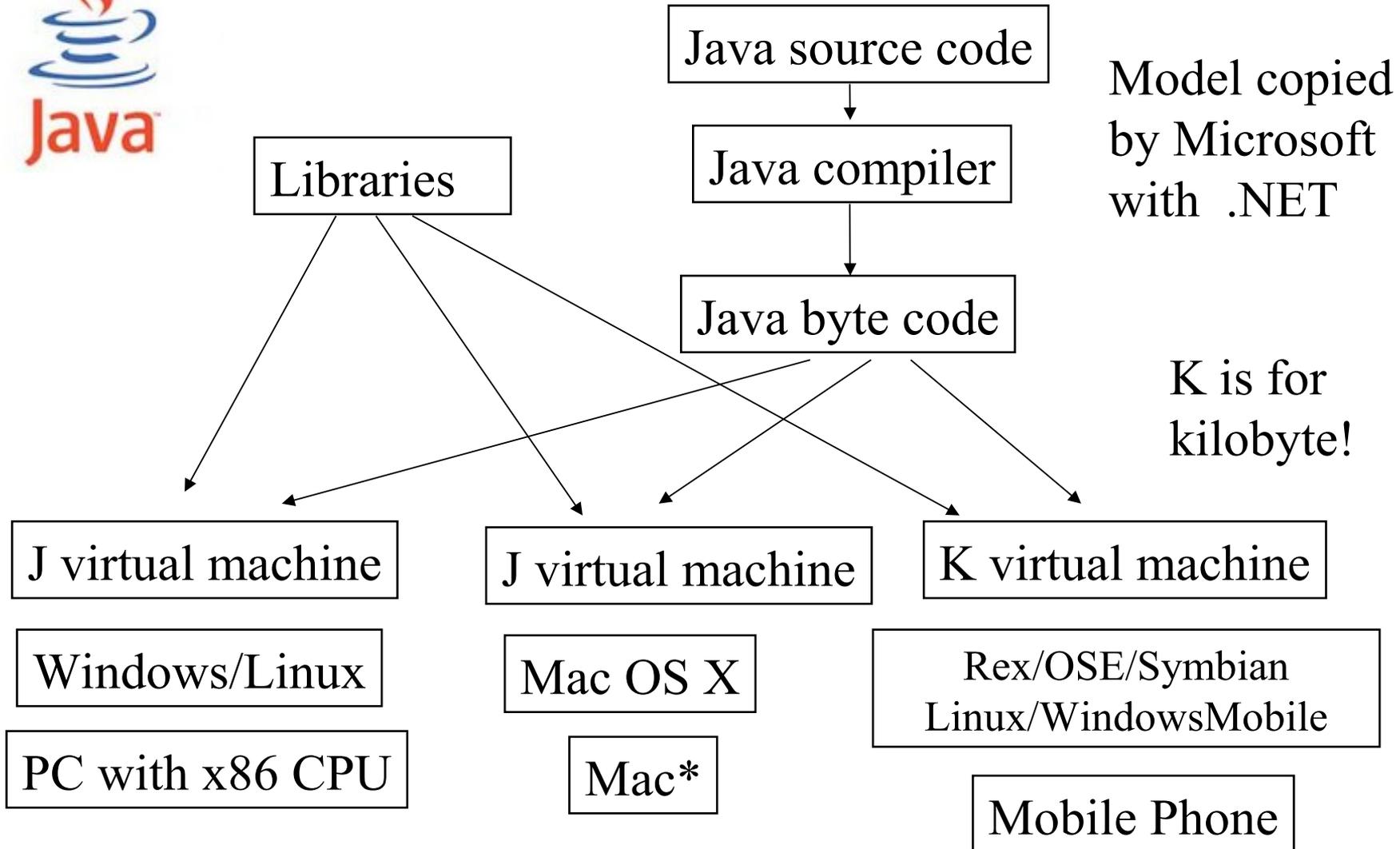
# Sensors in Android



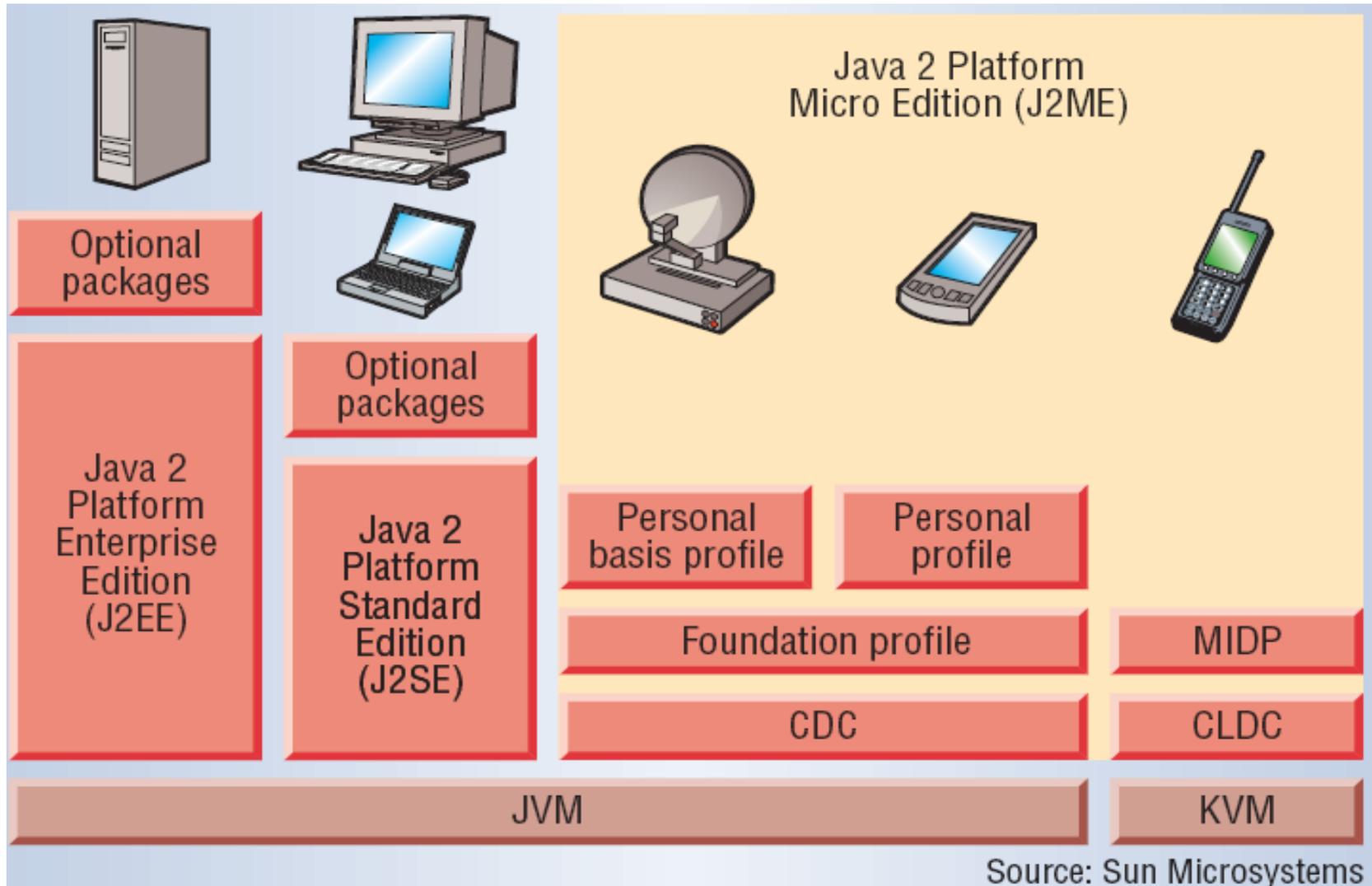
- Apart from the obvious as microphone and one or two cameras most phones contain
- Three axis accelerometer to measure gravity
- Three axis magnetometer to measure magnetic fields
- Temperature sensor for measuring ambient temperature
- GPS (Global Position System)
- Barometer (air pressure)
  - Increase the GPS performance and possibly predict weather
- Light sensor to optimize the display - conserve energy
- Proximity sensor - used when talking
- Gyroscope sensor to allow more accurate movement in 3D than an accelerometer
  - Allows the calculation of orientation and rotation



# The Java model



# Java platforms





# Whats needed to develop?

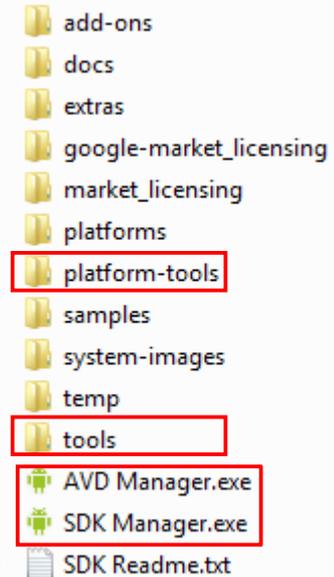


- The Android "tool chain"
  - Java SE JDK/JRE 5, 6, 7 (1.5.x – 1.7.x) or newer
  - Android SDK and tools
    - Build, test and debug the Android applications
  - Eclipse IDE or other Java IDE
    - Eclipse IDE for Java Developers 3.7.x (Indigo)
    - Android Development Tools (ADT) plugin for Eclipse
- Debug certificate
  - [drive]:\Users\\.android\debug.keystore
- Various other extra/special tools (not needed in our programming labs) as Ant, advanced testing tools etc.
  - Fastboot, Proguard, Hierarchy Viewer, etc.
  - Monkeyrunner for stress tests, TraceView for profiling etc.
- The first Android lab may have more info (system setup)

# Android SDK and tools folders



- SDK Manager and AVD Manager
- The Android Emulator
  - A virtual implementation of an Android phone (QEMU)
  - AVD (Android Virtual Device)
- Dalvik Debug Monitoring Service (DDMS)
  - Manage and control the Apps, GPS, SMS, calls, etc.
  - LogCat (see logged messages), file explorer, etc.
- Android Debug Bridge (ADB)
  - Manage the state of an emulator instance or Android-powered device
    - Copy files, install compiled application packages, run shell commands etc.
- Traceview
  - Graphical analysis tool for viewing the trace logs from the Android application, i.e. debug your application and profile its performance
- SQLite command tool
- MkSDCard
  - Creates a SD Card disk image to be used for external storage
- Zipalign (done automatically by ADT in Eclipse)
  - Optimize the created and signed .apk file, aligning the bytes to 4



# Eclipse with Android SDK - Java



The screenshot shows the Eclipse IDE interface. The Package Explorer on the left displays the project structure for 'SimpleTracker', including the 'src' directory with 'SMS.java'. The main editor window shows the code for 'SMS.java'. The toolbar at the top has several icons highlighted with red boxes: the Android icon, the Run icon, and the Java icon. Arrows point from these icons to text labels: 'AVD/SDK manager' points to the Android icon, and 'Window > Open Perspective' points to the Java icon. The bottom of the IDE shows a yellow sticky note with a list of bullet points.

AVD/SDK manager

Window > Open Perspective

- R.java - This file is automatically generated by Android Developer Tools and "connects" the visual resources to the Java source code, we never touch it ourselves
- res - Externalization of strings and graphics many other resource types
- layout - Contains the visual elements or resources in xml, for use by Android activities
- AndroidManifest.xml - This file contains all general information about the application

# Eclipse with Android SDK - DDMS



DDMS - SimpleTracker/src/se/du/simpletracker/SMS.java - Eclipse

File Edit Source Navigate Search Project Refactor Run Window Help

Java DDMS Debug PyDev

Threads Heap Allocation Tracker File Explorer Emulator Control

Devices

Name	State	API Level
304D1976013BF64E	Online	2.3.5
se.fl.nally.snow	18930	8600
emulator-5554	Online	api14 [4.0.1, d...
system_process	79	8601
com.android.sy	142	8602
com.android.in	156	8603
com.android.pl	171	8604
com.android.la	185	8605
com.android.se	211	8606
android.proces	239	8607
com.android.ca	258	8608
com.android.d	292	8609
com.android.co	309	8610
android.proces	331	8611
com.android.m	340	8612
com.android.pi	359	8613
com.android.er	389	8614
com.android.ex	407	8615

Telephony Status

Voice: home Speed: Full

Data: home Latency: None

Telephony Actions

Incoming number: 5556

Message: SMS sant!

Send Hang Up

Location Controls

Manual GPX KML

Decimal

Longitude -122,084095

Latitude 37,422006

Send

LogCat Method Purpose

- Log.e() Log errors
- Log.w() Log warnings
- Log.i() Log informational messages
- Log.d() Log Debug messages
- Log.v() Log Verbose messages

LogCat Console

Saved Filters All messages (no filters)

Search for messages. Accepts Java regexes. Prefix with pid, app; tag; or text: to limit scope.

L...	Time	PID	Application	Tag	Text
I	10-20 07:53:1...	185	com.android.launcher	dalvikvm-heap	Grow heap (frag case) to 7.931MB for 153680-byte allocation
D	10-20 07:53:1...	185	com.android.launcher	dalvikvm	GC_FOR_ALLOC freed 2K, 7% free 8048K/8583K, paused 62ms
D	10-20 07:53:1...	79	system_process	dalvikvm	GC_CONCURRENT freed 496K, 8% free 9199K/9927K, paused 7ms+15ms
W	10-20 07:53:1...	331	android.process.media	MediaScanner	Error opening directory '/mnt/sdcard/.android_secure/', skipping: Perm...
V	10-20 07:53:1...	331	android.process.media	MediaScanner	pruneDeadThumbnailFiles... android.database.sqlite.SQLiteCursor@410b79d0
V	10-20 07:53:1...	331	android.process.media	MediaScanner	/pruneDeadThumbnailFiles... android.database.sqlite.SQLiteCursor@410b79d0

LogCat verbose

# Eclipse with Android SDK - Debug



● Remember to start with the debug button!

● Debug controls

● Breakpoint

```
private void sendSMS(boolean binary)
{
    String phoneNo = txtPhoneNo.getText().toString();
    String message = txtMessage.getText().toString();

    if(phoneNo.length() > 0 && message.length() > 0)
    {
        if(binary)
            sendBinarySMS(phoneNo, message.getBytes());
        else
            sendTextSMS(phoneNo, message);
    }
    else

```

Name	Value
this	SMS (id=830019590464)
message	"" (id=830019702040)
phoneNo	"23423" (id=830019701952)
binary	false

Android

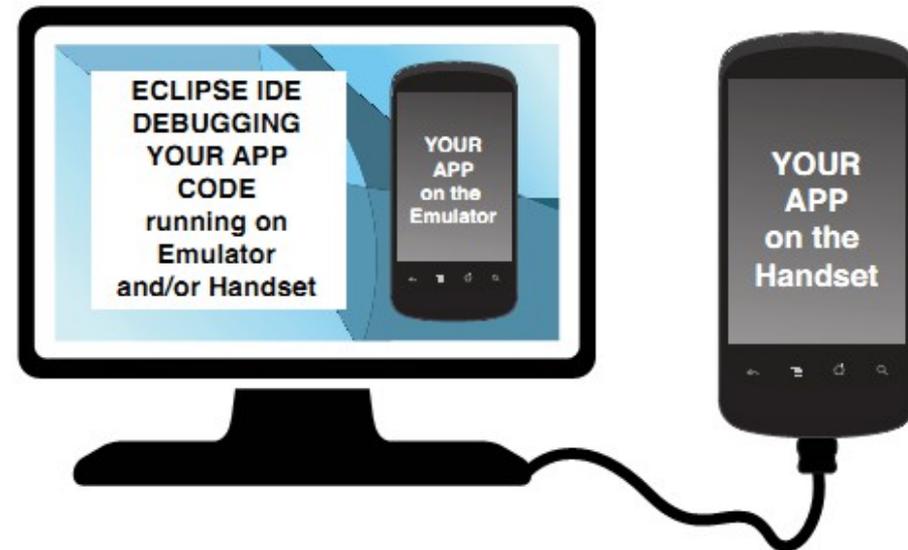
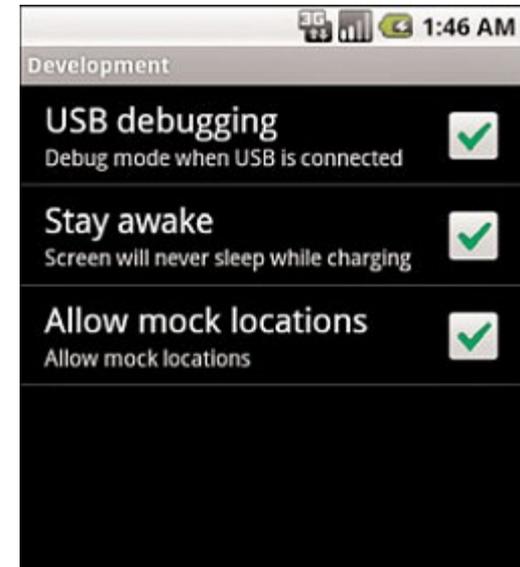
```
[2011-10-20 10:08:31 - SMS] Android Launch!
[2011-10-20 10:08:31 - SMS] adb is running normally.
[2011-10-20 10:08:31 - SMS] Performing se.du.sms.SMS activity launch
[2011-10-20 10:08:31 - SMS] Automatic Target Mode: Several compatible targets. Please select a target device.
[2011-10-20 10:08:37 - SMS] Application already deployed. No need to reinstall.
[2011-10-20 10:08:37 - SMS] Starting activity se.du.sms.SMS on device emulator-5554
[2011-10-20 10:08:39 - SMS] ActivityManager: Starting: Intent { act=android.intent.action.MAIN cat=[android.intent.category.LAUNCHER] cmp=se.du.sms/.SMS }
[2011-10-20 10:08:40 - SMS] Attempting to connect debugger to 'se.du.sms' on port 8616
```

Launching SMS

# Run or debug on-device



- You need to enable your device to install Android applications other than those from the Android Market
  - Menu > Settings > Applications > check (enable) the option called Unknown Sources
- Menu > Settings > Applications > Development
  - USB Debugging: This setting enables you to debug your applications via the USB connection.
  - Stay Awake: This convenient setting keeps the phone from sleeping in the middle of your development work, as long as the device is plugged in.
  - Allow Mock Locations: This setting enables you to send mock location information to the phone for development purposes and is very convenient for applications using location-based services (LBS).



# SDK and AVD manager



Android SDK Manager

SDK Path: C:\android-sdk-windows\

Name	API	Rev.	Status
Tools			
Android SDK Tools		14	Installed
Android SDK Platform-tools		8	Installed
Android 4.0 (API 14)			
Documentation for Android SDK	14	1	Installed
SDK Platform	14	1	Installed
Samples for SDK	14	1	Installed
ARM EABI v7a System Image	14	1	Installed
Google APIs by Google Inc.	14	1	Installed
Android 3.2 (API 13)			
Android 3.1 (API 12)			
Android 3.0 (API 11)			
Android 2.3.3 (API 10)			
SDK Platform	10	2	Installed
Samples for SDK	10	1	Installed
Dual Screen APIs by KYOCERA Corporation	10		Not installed
EDK 1.1 by Sony Ericsson Mobile Communications	10		Not installed
Google APIs by Google Inc.	10	2	Installed
Android 2.2 (API 8)			
SDK Platform	8	3	Installed
Samples for SDK	8		Not installed
Dual Screen APIs by KYOCERA Corporation	8		Not installed
Real3D by LGE	8		Not installed
GALAXY Tab by Samsung Electronics.	8		Not installed
Google APIs by Google Inc.	8	2	Installed
Android 2.1 (API 7)			
Android 1.6 (API 4)			
Android 1.5 (API 3)			
Extras			
Android Compatibility package	4		Installed
Google Admob Ads Sdk package	3		Installed
Google Market Billing package	1		Installed
Google USB Driver package	4		Installed
Google Webdriver package	1		Installed
Google Market Licensing package	1		Installed
Market Licensing package	1		Installed

Show:  Updates/New  Installed  Obsolete [Select New/Updates](#)

Sort by:  API level  Repository [Deselect All](#)

Done loading packages.

Install 6 packages...  
Delete packages...

Android Virtual Device Manager

List of existing Android Virtual Devices located at C:\Users\hjo\.android\avd

AVD Name	Target Name	Platform	API Level	CPU/ABI
myavd1	Google APIs (Google Inc.)	2.3.3	10	ARM (armeabi)
myavd2	Google APIs (Google Inc.)	2.3.3	10	ARM (armeabi)
api14	Android 4.0	4.0		ARM (armeabi-v7a)

✓ A valid Android Virtual Device.  
✗ An Android Virtual Device that fa

Edit Android Virtual Device (AVD)

Name: api14

Target: Android 4.0 - API Level 14

CPU/ABI: ARM (armeabi-v7a)

SD Card:

Size: 20 MiB  
 File: Browse...

Snapshot:  Enabled

Skin:  Built-in: HVGA  
 Resolution: x

Hardware:

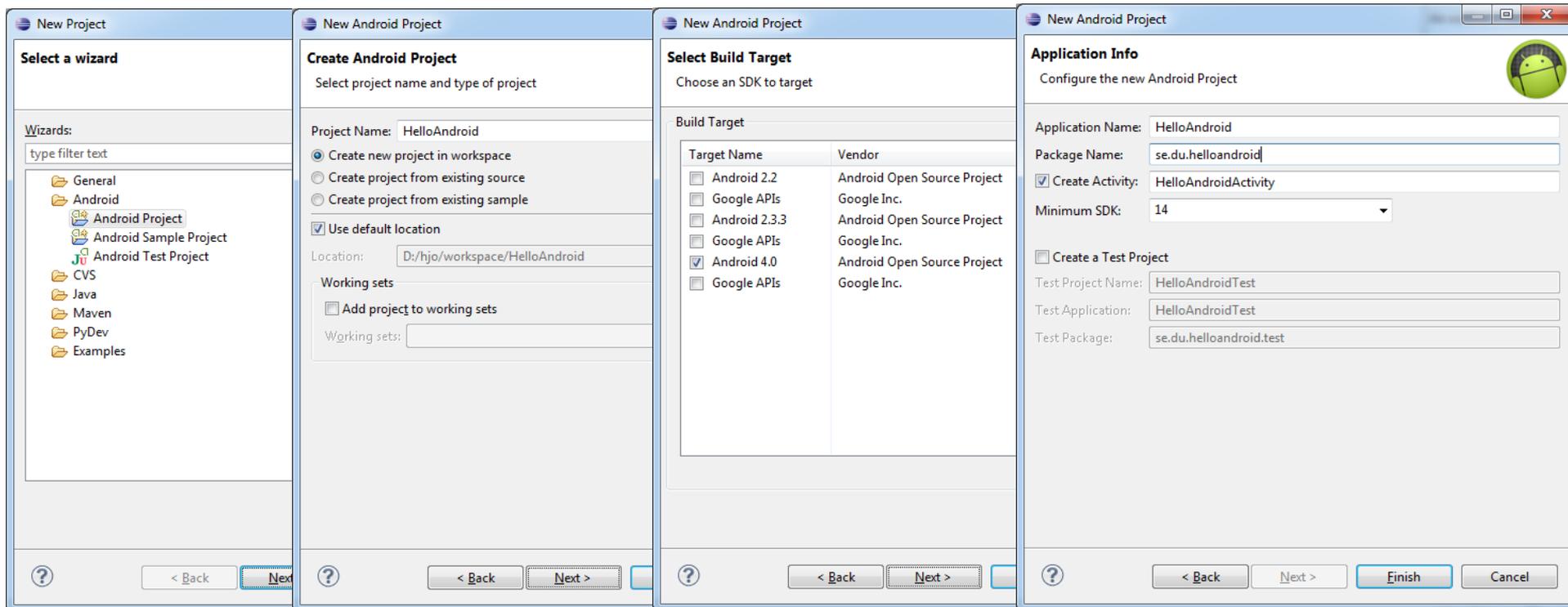
Property	Value	New...
Abstracted LCD density	160	Delete
Max VM application hea...	24	
Device ram size	256	

Override the existing AVD with the same name

Edit AVD Cancel

# Steps to create your first app

- Assuming you have a working development installation
- In Eclipse select File > New Project > Android Project
- Give a project name and select a build target (an AVD must exist)
- Enter application info details - give a package name



The image displays four sequential screenshots of the Eclipse IDE's 'New Project' wizard, illustrating the steps to create a new Android project.

**Screenshot 1: Select a wizard**  
The 'Wizards' list is shown, with 'Android Project' selected.

**Screenshot 2: Create Android Project**  
The 'Project Name' is 'HelloAndroid'. The 'Create new project in workspace' option is selected. The 'Use default location' checkbox is checked, and the 'Location' is 'D:/hjo/workspace/HelloAndroid'.

**Screenshot 3: Select Build Target**  
The 'Build Target' table is shown, with 'Android 4.0' selected.

Target Name	Vendor
<input type="checkbox"/> Android 2.2	Android Open Source Project
<input type="checkbox"/> Google APIs	Google Inc.
<input type="checkbox"/> Android 2.3.3	Android Open Source Project
<input type="checkbox"/> Google APIs	Google Inc.
<input checked="" type="checkbox"/> Android 4.0	Android Open Source Project
<input type="checkbox"/> Google APIs	Google Inc.

**Screenshot 4: Application Info**  
The 'Application Info' dialog is shown, with the following details:

- Application Name: HelloAndroid
- Package Name: se.du.helloandroid
- Create Activity: HelloAndroidActivity
- Minimum SDK: 14
- Create a Test Project
- Test Project Name: HelloAndroidTest
- Test Application: HelloAndroidTest
- Test Package: se.du.helloandroid.test



# Hello World

ANDROID



```
// HelloWorldMIDlet.java
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;

public class HelloWorldMIDlet extends MIDlet
{
    private Display display;
    private Alert warning;

    public HelloWorldMIDlet()
    {
        warning = new Alert("Hi!", "Hello World!", null, null);
        display = Display.getDisplay(this);
    }

    // Lifecycle methods
    public void destroyApp(boolean unconditional) { }
    public void pauseApp() { }
    public void startApp() {
        display.setCurrent(warning);
    }
}
```

```
package se.du.helloandroid;
import android.app.Activity;
import android.os.Bundle;
import android.widget.TextView;

public class HelloAndroidActivity extends Activity {
    private TextView mTV1;
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);

        // alt.1 via layout in XML
        setContentView(R.layout.main);
        mTV1 = (TextView) findViewById(R.id.textView1);
        mTV1.setText(R.string.hello_android);

        // alt.2 via Java code
        TextView tv = new TextView(this);
        tv.setText("Hello, Android");
        setContentView(tv);
    }
    // the other lifecycle methods
    // use @Override for all
    public void onStart() {super.onStart();}
    public void onRestart() {super.onRestart();}
    public void onResume() {super.onResume();}
    public void onPause() {super.onPause();}
    public void onStop() {super.onStop();}
    public void onDestroy() {super.onDestroy();}
}
```

# Project > Build and Run



Java - HelloAndroid/src/se/du/helloandroid/HelloAndroidActivity.java - Eclipse

File Edit Source Navigate Search Project Refactor Run Window Help

Package Explorer

- HelloAndroid
  - src
    - se.du.helloandroid
      - HelloAndroidActivity.java
  - gen [Generated Java Files]
    - se.du.helloandroid
  - Android 4.0
    - android.jar - C:\android-sdk-w
  - assets
  - bin
    - res
      - classes.dex
      - HelloAndroid.apk
      - resources.ap\_
  - res
    - drawable-hdpi
      - ic\_launcher.png
    - drawable-ldpi
      - ic\_launcher.png
    - drawable-mdpi
      - ic\_launcher.png
    - layout
      - main.xml
    - values
      - strings.xml
  - AndroidManifest.xml
  - proguard.cfg
  - project.properties
- HttpDownload
- info24

```
package se.du.helloandroid;

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

public class HelloAndroidActivity extends Activity {
    private TextView mTV1;
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        // alt.1 via layout in XML
        setContentView(R.layout.main);
        mTV1 = (TextView) findViewById(R.id.textView1);
        mTV1.setText(R.string.hello_android);
    }

    // alt.2 via Java code
    TextView tv = new TextView(this);
    tv.setText("Hello, Android");
    setContentView(tv);
}

@Override
public void onStart() {super.onStart()}

@Override
public void onResume() {super.onResume()}

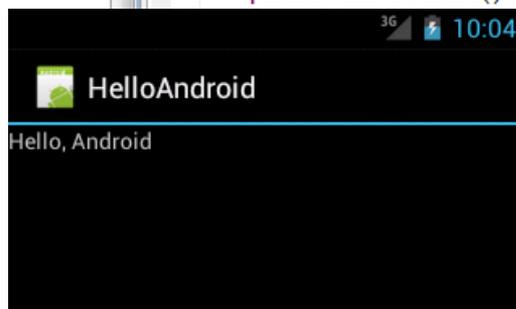
@Override
public void onRestart() {super.onRestart()}

@Override
public void onStop() {super.onStop()}

@Override
public void onDestroy() {super.onDestroy()}

@Override
public void onPause() {super.onPause()}

@Override
public void onRestoreInstanceState(Bundle savedInstanceState) {super.onRestoreInstanceState(savedInstanceState)}
```



Run As

Select a way to run 'HelloAndroid':

- Android Application
- Android JUnit Test
- Java Applet
- Java Application
- JUnit Test

Description

Runs an Android Application

OK Cancel

Android Device Chooser

Select a device compatible with target Android 4.0.

Choose a running Android device

Serial Number	AVD Name	Target	Debug	State
304D1976013BF64E	N/A	2.3.5		Online
emulator-5554	api14	Google APIs (...)	Yes	Online

Launch a new Android Virtual Device

AVD Name	Target Name	Platform	API Level	CPU/ABI	Details...
--	No AVD available	--	--	--	Start...

Refresh Manager...

OK Cancel

# AndroidManifest.xml



The screenshot displays an IDE window with two panes. The left pane shows the raw XML code of the `AndroidManifest.xml` file. The right pane shows the graphical configuration interface for the manifest.

**XML Code (Left Pane):**

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="se.du.helloandroid"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk android:minSdkVersion="14" />

    <application
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name" >
        <activity
            android:label="@string/app_name"
            android:name=".HelloAndroidActivity" >
            <intent-filter >
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

**Configuration Panel (Right Pane):**

- Android Manifest**
  - Manifest General Attributes**

Defines general information about the AndroidManifest.xml

Package	se.du.helloandroid	Browse...
Version code	1	
Version name	1.0	Browse...
Shared user id		Browse...
Shared user label		Browse...
Install location		
  - Manifest Extras**
    - Uses Sdk
    - Buttons: Add..., Remove..., Up, Down
  - Attributes for Uses Sdk**
    - The `uses-sdk` tag describes the SDK features that the containing package must be running on to operate correctly.

Min SDK version	14	Browse...
Target SDK version		Browse...
Max SDK version		
- Exporting**

To export the application for distribution, you have the following options:

  - Use the [Export Wizard](#) to export and sign an APK
  - Export an [unsigned APK](#) and sign it manually
- Links**

The content of the Android Manifest is made up of three sections. You can also edit the XML directly.

  - A Application:** Activities, intent filters, providers, services and receivers.
  - P Permission:** Permissions defined and permissions used.
  - I Instrumentation:** Instrumentation defined.
  - XML Source:** Directly edit the AndroidManifest.xml file.
  - Documentation:** Documentation from the Android SDK for AndroidManifest.xml.

# [res] layout/main.xml



Editing config: default    Any locale    Android 4.0    Create...

3.2in HVGA (ADP2)    Portrait    Normal    Day time    Theme

Palette

Form Widgets

TextView Large Medium Small Button

OFF     CheckBox

RadioButton     CheckedTextView

Spinner

Progress indicators and stars are also visible in the palette.

Id = @+id/textView1

Property	Value
Sound effects enabled	
Style	
Tag	
<b>Text</b>	<b>@string/hello</b>
Text all caps	
Text appearance	
Text color	
Text color highlight	
Text color hint	
Text color link	
Text cursor drawable	
Text direction	
Text edit no paste	
Text edit paste	
Text edit side no paste	
Text edit side paste	
Text edit suggestion	
Text is selectable	
Text scale X	
Text select handle	
Text select handle	
Text select handle	
Text size	
Text style	
Transform pivot X	
Transform pivot Y	
Translation X	
Translation Y	
Typeface	
Vertical scrollbar position	
Width	
Visibility	
Misc	
Layout gravity	
Layout height	wrap_content
Layout margin	
Layout margin bottom	
Layout margin end	

Graphical Layout    main.xml

# [res] layout/main.xml and values/strings.xml



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

    <TextView
        android:id="@+id/textView1"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/hello_android" />
</LinearLayout>
```

The screenshot shows the Android Studio IDE interface. The top bar displays the current file as 'strings.xml'. Below the toolbar, the 'Android Resources (default)' panel is active. On the left, the 'Resources Elements' list contains three items: 'hello (String)', 'app\_name (String)', and 'hello\_android (String)'. The 'hello\_android (String)' item is selected. To the right of this list are buttons for 'Add...', 'Remove...', 'Up', and 'Down'. On the right side of the panel, the 'Attributes for hello\_android (String)' section is visible. It contains a description of strings and two input fields: 'Name\*' with the value 'hello\_android' and 'Value\*' with the value 'Hello, Android'.

```
<?xml version="1.0" encoding="utf-8"?>
<resources>

    <string name="hello">Hello World, HelloAndroidActivity!</string>
    <string name="app_name">HelloAndroid</string>
    <string name="hello_android">Hello, Android</string>

</resources>
```

# Difference C# vs. Java?

- Marginal differences

- CLR vs. JVM
- ASP .NET vs Servlets
- WinForms vs. Applets

<http://www.25hoursaday.com/CsharpVsJava.html>

```
using library;
class Hello {
    public static void Main() {
        Console.WriteLine("Hello world"); }
}
```

```
import library;
class Hello {
    public static void main(String args[]) {
        System.out.println("HelloWorld"); }
}
```

- Inheritance

- Multiple inheritance is not supported by either language

- Operators and control flow

- Operator overloading and goto (removed in java) otherwise similar

- Exception handling

- Not as forced in C# as in Java

A Java programmer just do this:

```
public synchronized void methodName() {...}
```

- Multithreading

- More simplified in C#, must synchronize with monitor and mutex class

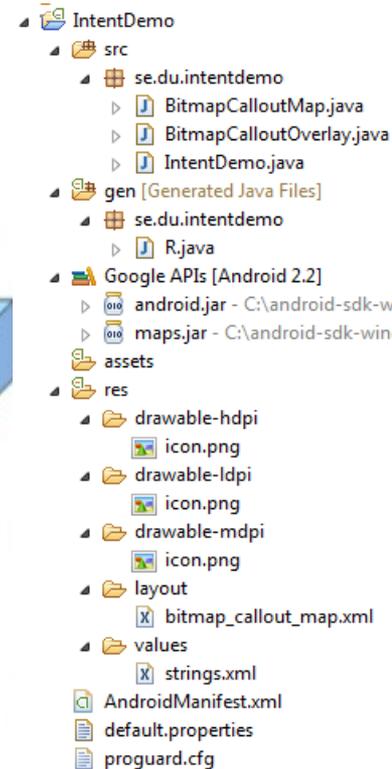
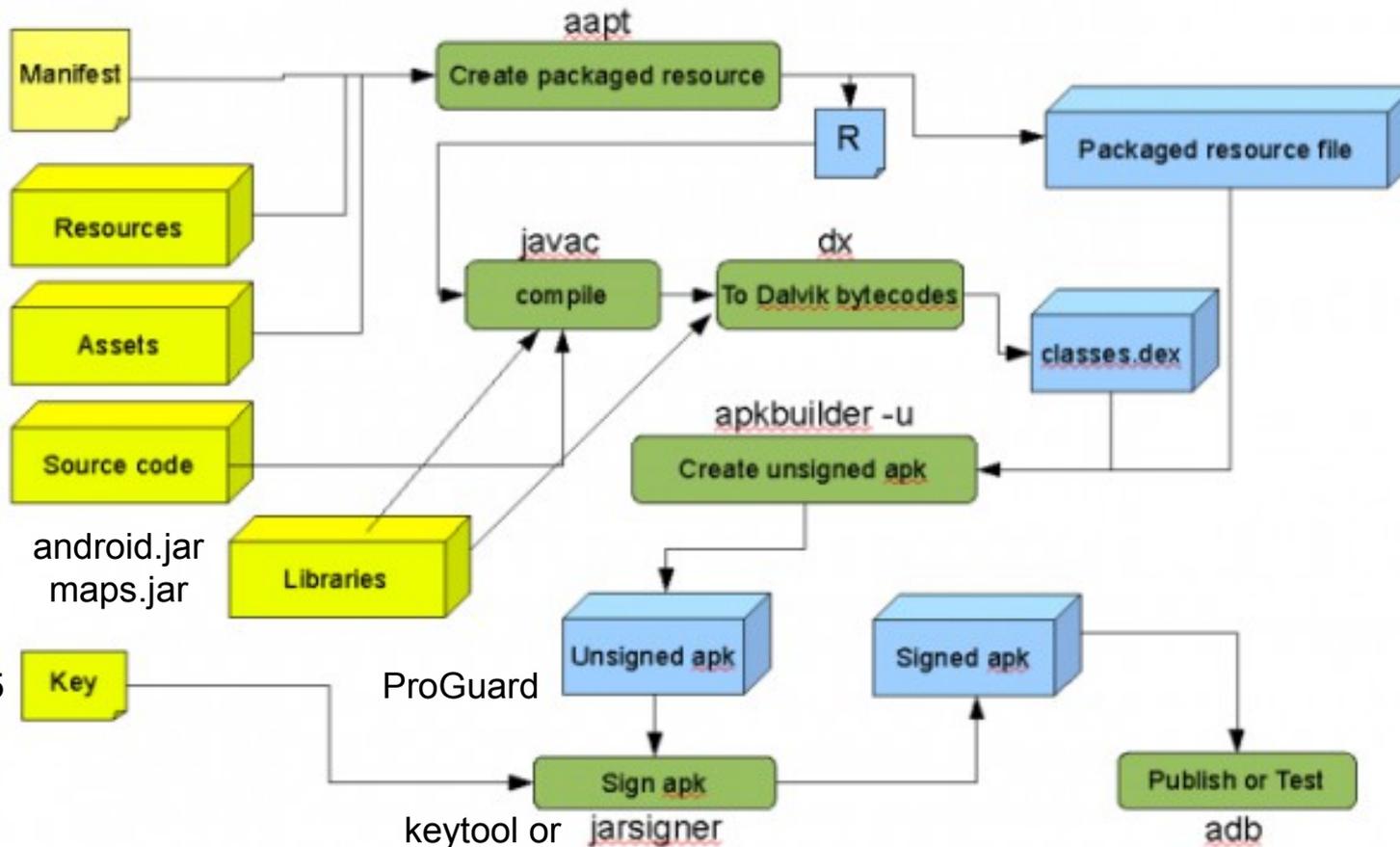
- Read more...

- [http://en.wikipedia.org/wiki/Comparison\\_of\\_C\\_Sharp\\_and\\_Java](http://en.wikipedia.org/wiki/Comparison_of_C_Sharp_and_Java)

# Android application package (apk) build process



- Can be done by hand from the command line
- Android Asset Packaging Tool (aapt)



# ProGuard



**Android SDK tools revision 12-14 has problem with ProGuard**  
<http://code.google.com/p/android/issues/detail?id=18359>

Welcome to ProGuard, version 4.4

ProGuard is a free class file shrinker, optimizer, obfuscator, and preverifier.

With this GUI, you can create, load, modify, and save ProGuard configurations. You can then process your code right away, or you can run ProGuard from the command line using your saved configuration.

With the ReTrace part of this GUI you can de-obfuscate your stack traces.

ProGuard and ReTrace are written and maintained by Eric Lafortune.

Distributed under the GNU General Public License.  
Copyright (c) 2002-2009.

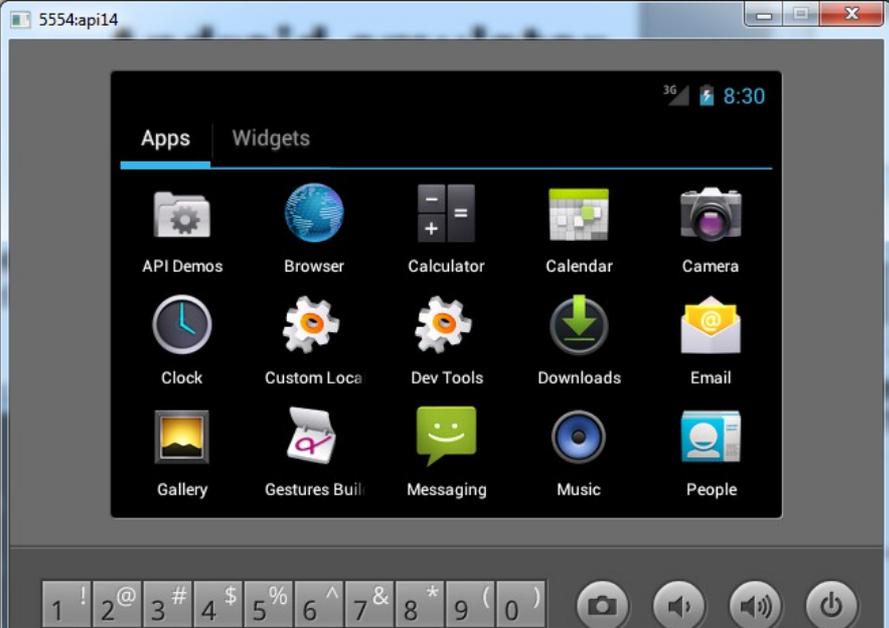
The ProGuard tool shrinks, optimizes, and obfuscates your code by removing unused code and renaming classes, fields, and methods with semantically obscure names. The result is a smaller sized .apk file that is more difficult to reverse engineer.

Enable ProGuard with the proguard.config property in the <project\_root>/default.properties file.  
proguard.config=proguard.cfg

<http://developer.android.com/guide/developing/tools/proguard.html>

Load configuration... Next

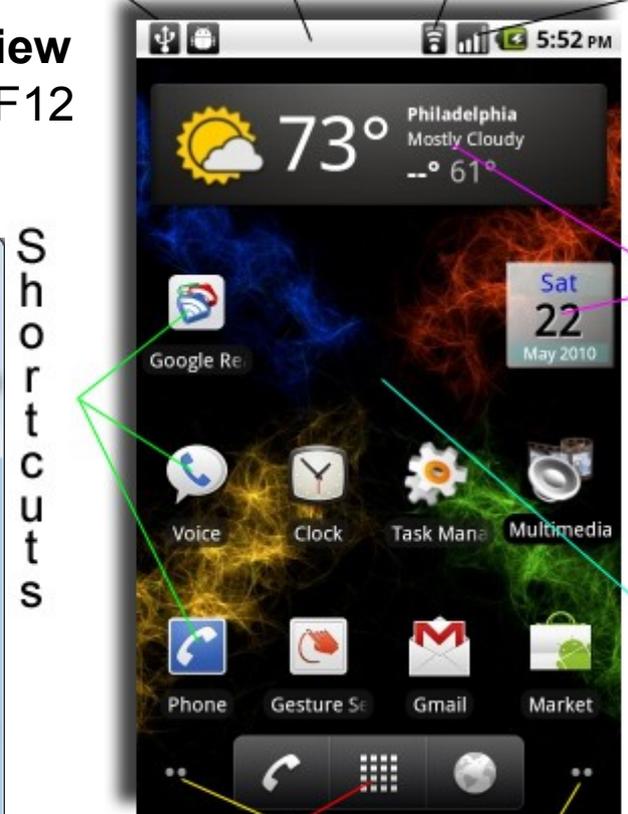
# Android emulator



Rotate view  
Ctrl-F11/F12



Notification Bar  
WIFI Strength  
Cell Signal Strength



Shortcuts

Widgets  
Home screen

App Drawer

Home Screen indicator

# Emulator Control

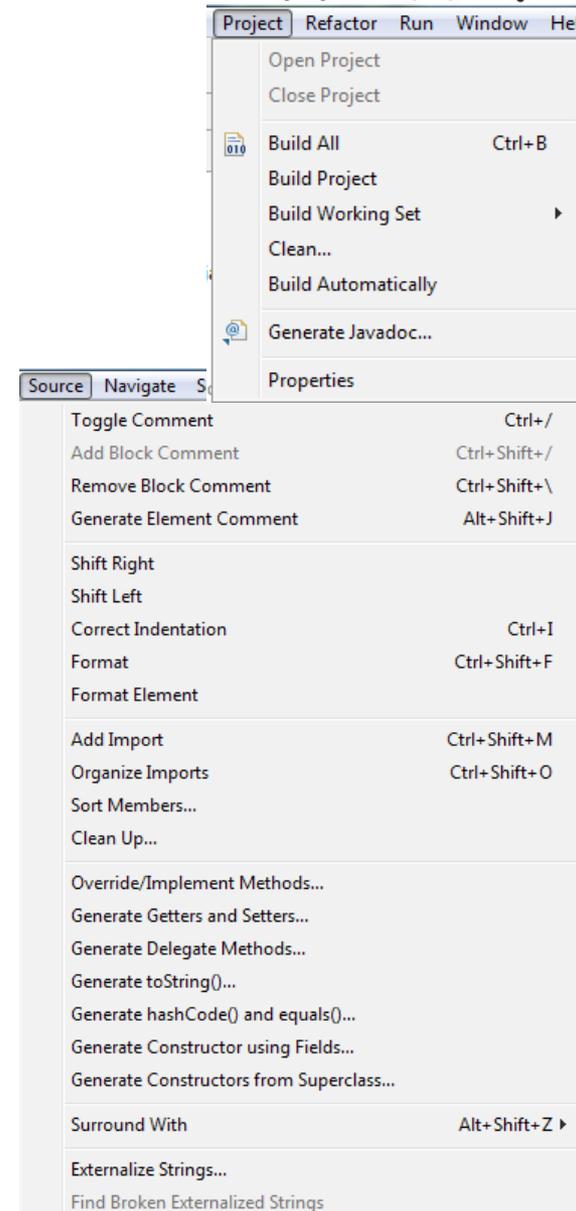


Key	Emulated Function
Escape	Back button
Home	Home button
F2, PageUp	Menu button
Shift-F2, PageDown	Start button
F3	Call/Dial button
F4	Hangup/EndCall button
F5	Search button
F7	Power button
Ctrl-F3, Ctrl-KEYPAD_5	Camera button
Ctrl-F5, KEYPAD_PLUS	Volume up button
Ctrl-F6, KEYPAD_MINUS	Volume down button
KEYPAD_5	DPAD center
KEYPAD_4, KEYPAD_6	DPAD left, DPAD right
KEYPAD_8, KEYPAD_2	DPAD up, DPAD down
F8	Toggle cell network on/off
F9	Toggle code profiling (when <b>-trace</b> set)
Alt-ENTER	Toggle fullscreen mode
Ctrl-T	Toggle trackball mode
Ctrl-F11, KEYPAD_7	Rotate screen orientation to previous or next layout
Ctrl-F12, KEYPAD_9	

# Eclipse and Android SDK



- The emulator only needs to be started once!
- Eclipse IDE for Java Developers 32/64 bit and Java JDK 32/64 bit must match
- Java not found?
  - `eclipse.exe -vm "C:\Program Files\Java\jre6\bin\javaw"`.
- Sometimes import `android.R;` is wrongly imported into your source especially when copy and pasting code.
- To force IntelliSense press `Ctrl + Space` at certain places for a pop-up list with syntax and method proposals.
- Try out some of the SDK samples in `C:\android-sdk-windows\samples\API-LEVEL` if you want.
  - You need to create a new project from existing sample in the New Android Project dialogue in this case.
- Android SDK environment variables
  - `Path=c:\android-sdk-windows\tools;c:\android-sdk-windows\platform-tools;c:\Program Files\Java\jdk1.6.0_26\bin`
  - `"ANDROID_SDK_HOME"="H:"`



# Web resources



- Java, Eclipse, Android, and Web Programming Tutorials
  - <http://www.vogella.de/>
- Android Developer Pages
  - <http://developer.android.com/>
- Stack Overflow Discussion Threads
  - <http://stackoverflow.com/>
- <http://www.swedroid.se/>
- <http://www.anddev.org/>
- [http://en.wikipedia.org/wiki/Android\\_%28operating\\_system%29](http://en.wikipedia.org/wiki/Android_%28operating_system%29)
- Bygg din egen Android-app
  - <http://www.idg.se/2.1085/1.327186/bygg-din-egen-android-app>