App Development for Smart Devices

CS 495/595 - Fall 2013

Tamer Nadeem
Dept. of Computer Science

Course Logistics

Welcome to CS 495/595

- **Timings:** Monday 7:10pm to 9:50pm
- **Location:** Dragas 1117
- **Instructor:** Tamer Nadeem

  Ph.D from Univ. of Maryland, 2006

  Research in Networks, Dist Sys, Mobile Comp.

  Email: nadeem@cs.odu.edu

  Office: ECSB 3204
- **Office Hours:** Mon 1:00pm-2:30pm,
  or by appointment

Welcome to CS 495/595

- **Teaching Asst.:** Mostafa Uddin

  Email: muddin@cs.odu.edu

  Office: ECSB 3106

- **Office Hours:** Wednesday 10:00am-11:30am, or by appointment

Welcome to CS 495/595

- **Prerequisites:** Comfortable with Java

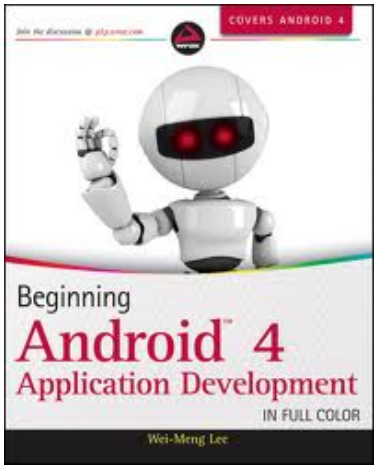
- **Grading:**
  - Participation: 10%
  - Midterm: 25%
  - Programming Assignments: 40%
  - Final Project: 25%

Welcome to CS 495/595

- **Class Webpage:**
  - http://www.cs.odu.edu/~cs495/
  - Please check course website frequently

- **Make up classes:**
  - Will be occasionally necessary due to travel
  - Fixed schedule versus case by case basis?

Welcome to CS 495/595



- **Text:**
  - Required:
    - Wei-Meng Lee, "*Beginning Android 4 Application Development*"
  - Recommended:
    - Reto Meier, "Professional Android 2 Application Development"
    - James Steele, Nelson To, "The Android Developer's Cookbook“
  - Android Developers (Dev Guide, Reference, etc.): http://developer.android.com/index.html

Welcome to CS 495/595

- **Academic Integrity / Honor Code:**
  - "I pledge to support the honor system of Old Dominion University. I will refrain from any form of academic dishonesty or deception, such as cheating or plagiarism. I am aware that as a member if the academic community, it is my responsibility to turn in all suspected violators of the honor system. I will report to Honor Council hearings if summoned.“
  - Please refer to ODU Honor Council’s webpage: http://orgs.odu.edu/hc/

Welcome to CS 495/595

- **Course Policy:**
  - Grading:
    - 90-100 A
    - 80-89 B
    - 70-79 C
    - 0-69 F
  - Late assignments are not accepted.
  - Attendance
  - Account & Email

Please refer to class webpage for more details.

Course Overview

This Course

- **Introduces fundamentals of application development for Android phones**

- **Goals of this course:**
  - Help you learn about mobile app development and best practices
  - Provide you with the tools, knowledge, and excuse to create a novel mobile app that helps solve a serious problem that strengthens your programming portfolio

- **Envisions new practical mobile applications/services**

Class Responsibilities

- **I will lead lectures**
  - You present 1-2 paper(s) in entire semester (25 minutes)
  - 2-3 students presentation per class
  - Some classes will include coding
- **For every class, read the readings list before the class**
- **Assignments should be on time**

Course Structure

- **1 mid term?, No Final Exam**
  - Tentative date of mid-term: Nov 11th
- **Semester-long class project**
  - In groups of 2 (max 3).
    - Individual projects are allowed by permission
  - Focus on this from early on
- **Class ends with a final project presentation & demo**

Participation / Presentation

- **Ask lots of questions. Period.**
  - I strongly encourage you to ask, disagree, debate
- **Class presentation**
  - You present 1-2 paper (25 minutes)
  - Check class schedule by next week for reading papers
  - Email me any paper you are interested in
  - Pick 4 open slots (check class schedule)
    - Earlier you pick, more options you have to choose from
    - Deadline is Sep 06, 2013
- **Email me your choice of paper (and date)**
  - Don’t worry about not knowing the topic of paper
    - By that time, you will know enough

Thoughts on Reading Assignments

- **Know why you are reading the paper**
  - Reading for absorbing concepts (class assignment)
    - Read fully, think, reread, ask, challenge
  - Reading for excitement (deciding project topic)
    - Read initial parts, don't try to understand everything, get a feel

Course Term Project

- **Initial proposal due Oct 10 after Fall break**
  - 1- page progress report every 2 weeks (due Friday night)
  - Final report + demo + presentation
- **Projects consist of:**
  - Application/Service identification
  - Solution design
  - Implementation
- **Discuss your thoughts and ideas with me**
  - They need not be cooked, and can have many flaws
  - Statistically, every 18 ideas lead to one decent idea
- **If you like an area/direction**
  - Search and Read many many related references

More on Projects

- **Project ideas take time … think now and then**
  - Spending 3 hours for 10 days better than 10 hours for 3 days
- **Find a project partner(s) early**
  - Search and discuss App/Services ideas
- **Everyone in the class will try/critique apps from other teams**
- **At end of the course we will vote for the Top App**
- **Possible Application/Service domains:**
  - Transportation
  - Education
  - Health
  - Energy
  - Smart Home

Labs/Facilities

- **Development Environment:**
  - Your laptop
  - ECSB 3104 (Open Research Lab)
  - SmartApp Lab (under construction)
- **Collaboration:**
  - Internet
  - BlackBoard Discussion (http://clt.odu.edu/bb/)
  - Friends/Colleagues

INTRODUCTION

Mobile Computing

- **Driven by technology and vision**
  - wireless communication technology
  - global infrastructure
  - device miniaturization
  - mobile computing platforms

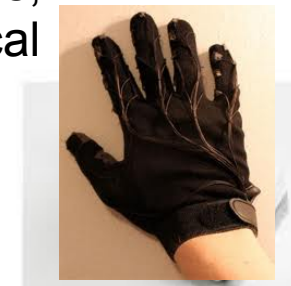
- **The field is moving fast**
- "People and their machines should be able to access information and communicate with easily and securely, in any medium each other or combination of media – voice, data, image, video, or multimedia – any time, anywhere, in a timely, cost-effective way.", ***Dr. G. H. Heilmeier, Oct 1992***
- "The mobile device will be the primary connection tool to the Internet for most people in the world in 2020.", ***PEW Internet and American Life Project, Dec. 2008***

Smart Devices

- A **smart device** is a device that is digital, active, computer networked, is user reconfigurable and that can operate to some extent autonomously.

- A **smart device** is a ubiquitous computing device: a device that exhibits some properties of ubiquitous computing including artificial intelligence.

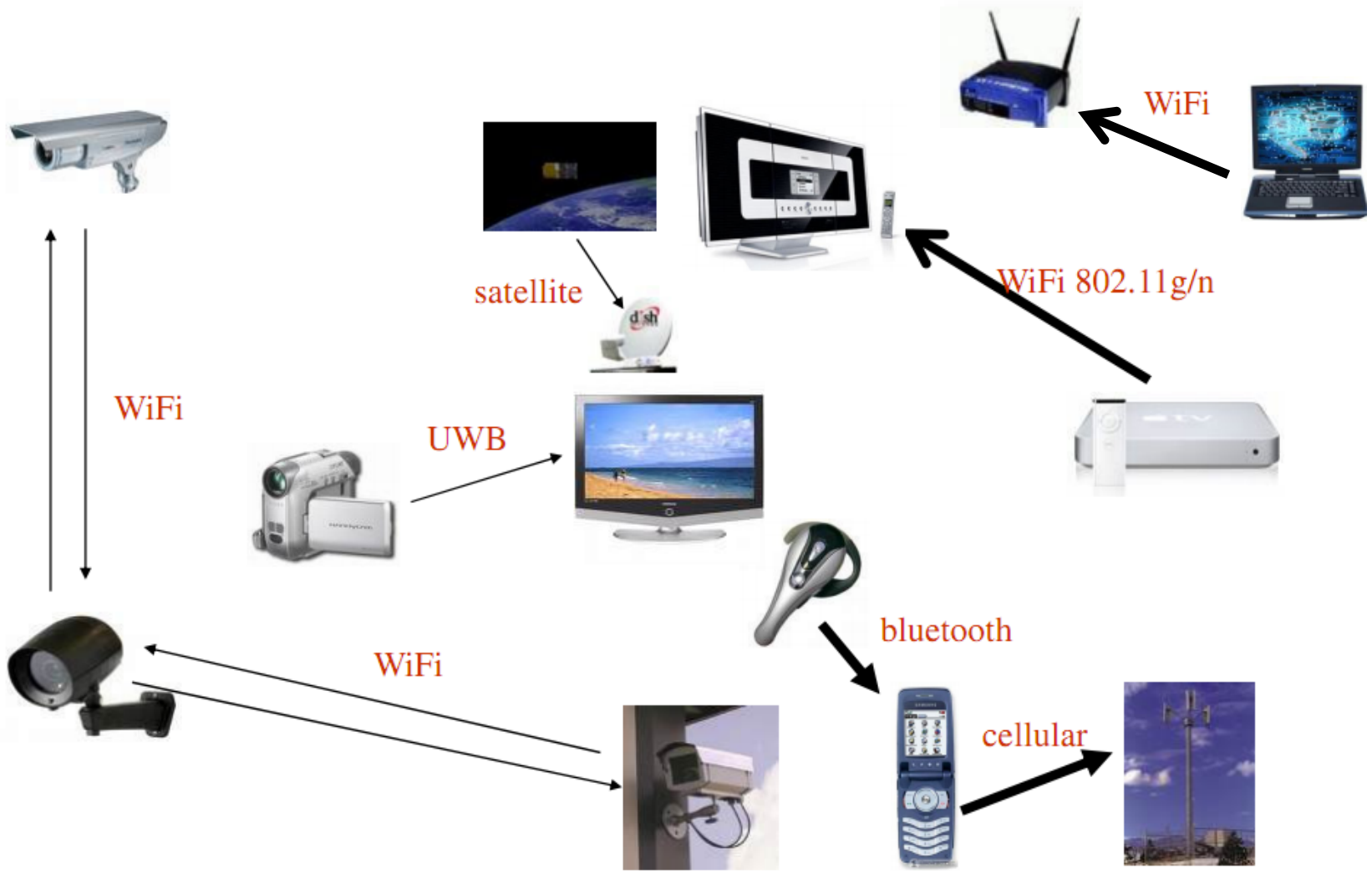
- **Mark Weiser** categorized ubiquitous devices:
  - **Tabs:** accompanied or wearable centimeter sized devices, e.g., smartphones, smart cards
  - **Pads:** hand-held decimeter-sized devices, e.g., laptops
  - **Boards:** meter sized interactive display devices, e.g., horizontal surface computers and vertical smart boards.



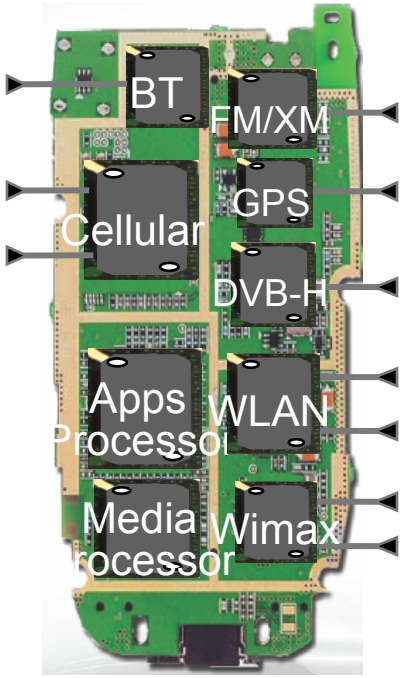
Mobile Devices

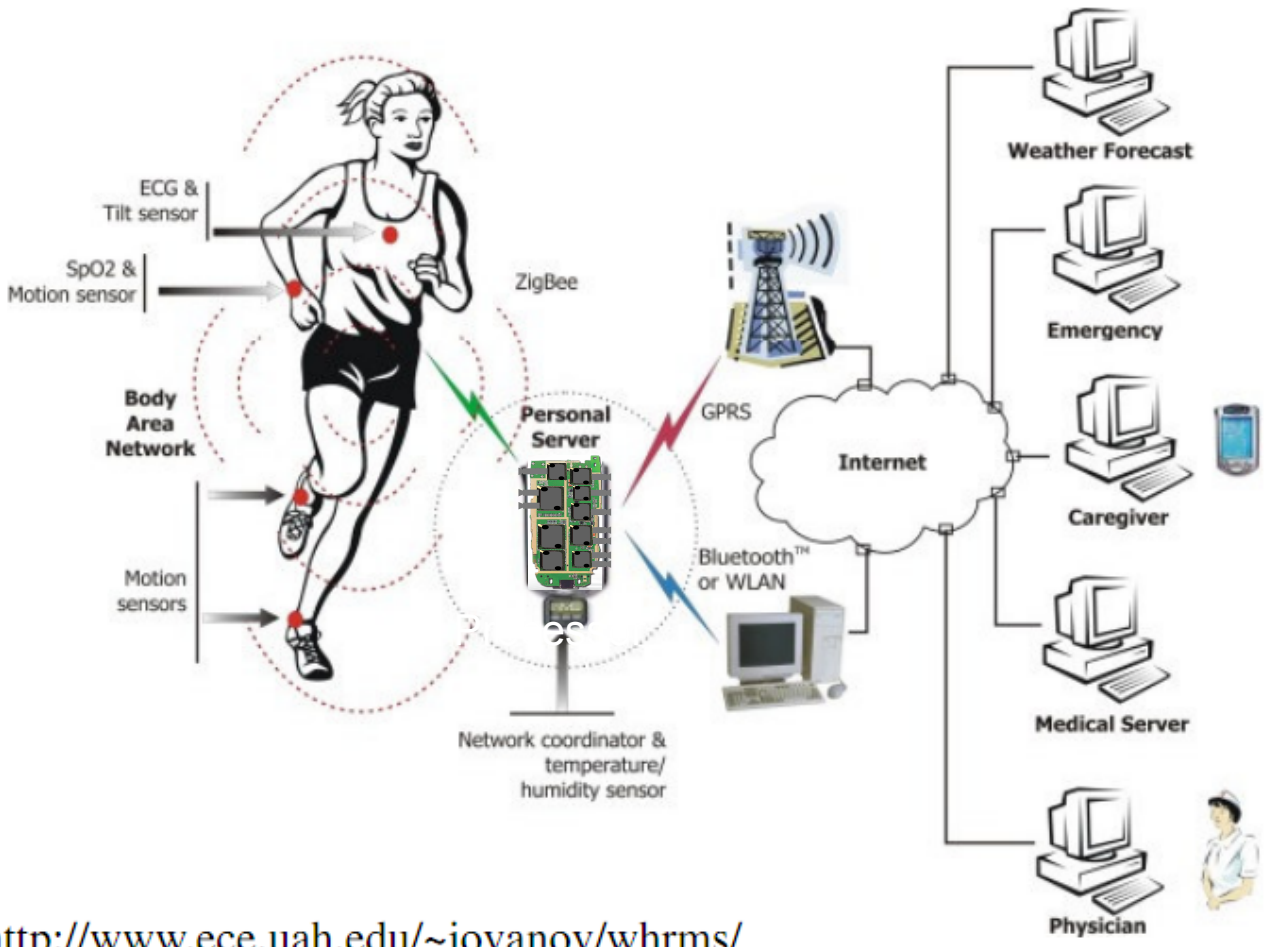
- Multi-purpose devices (e.g., personal office, mobile phone, camera, etc.)
- Mobility: loosely-bound vs. tightly-bound to users
- Personalized
- Operates as a single portal, e.g., a Web portal.
  - Internal application services
  - External services typically accessed local area wireless network
- Intermittent resource access
- A locus of control that resides in the smart device.
- Networked, distributed and transparently accessible.
- Context awareness

Smart Devices at Home/Office



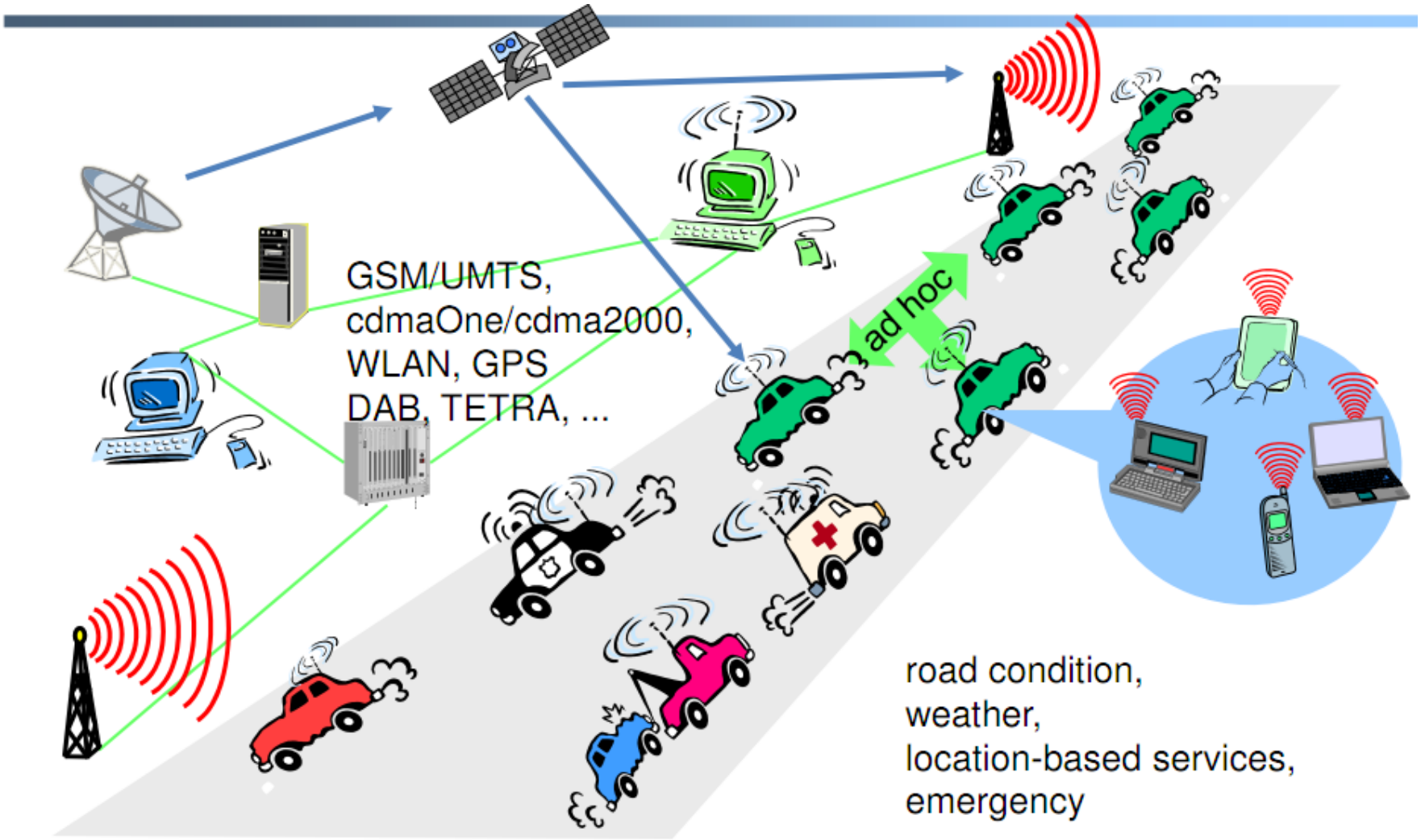
Smart Devices on Move





Source: http://www.ece.uah.edu/~jovanov/whrms/

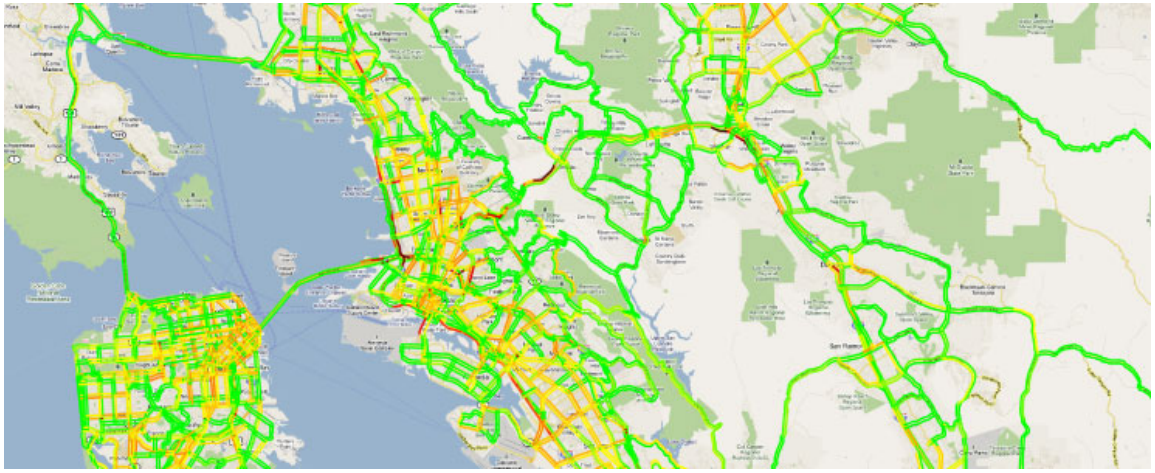
Smart Devices on Road

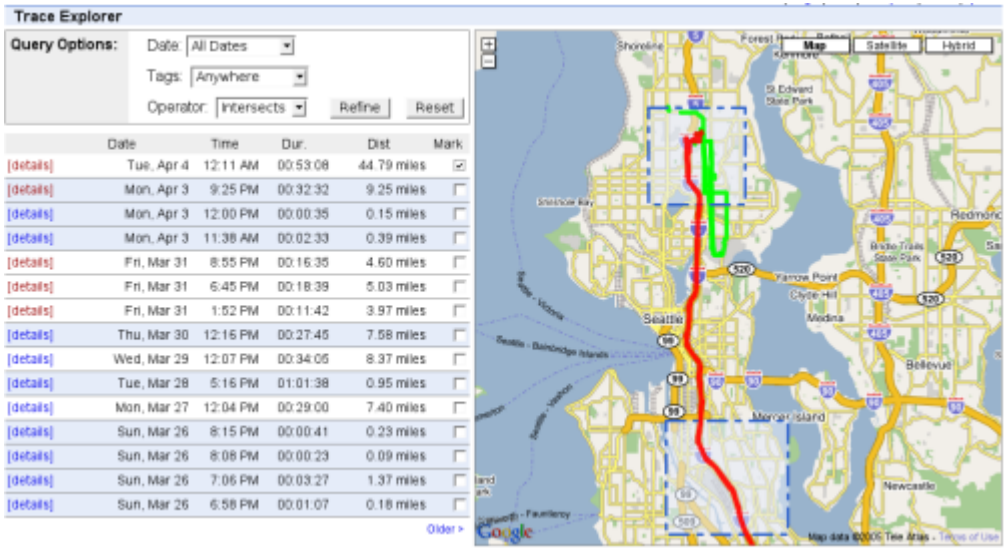


Smart Devices for Intelligent Transportation

Mobile Millennium Traffic in San Francisco and the Bay Area

Source:
http://traffic.berkeley.edu/



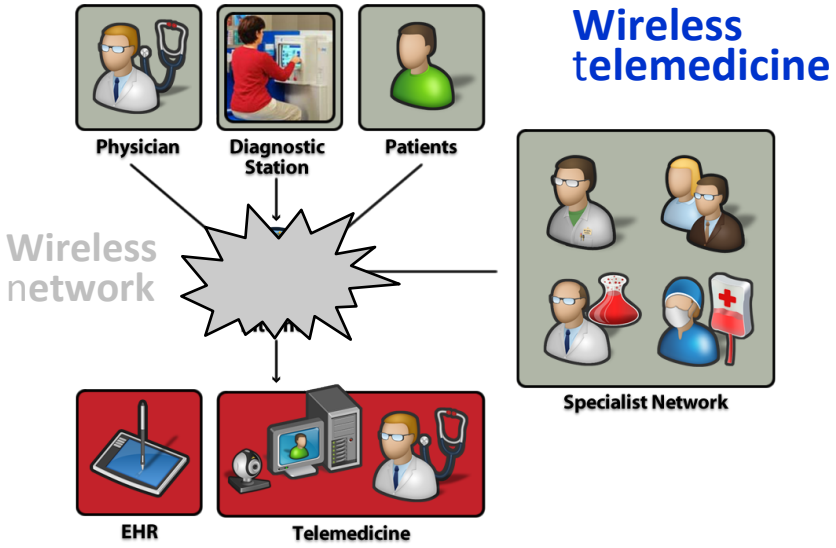


CarTel Project at MIT

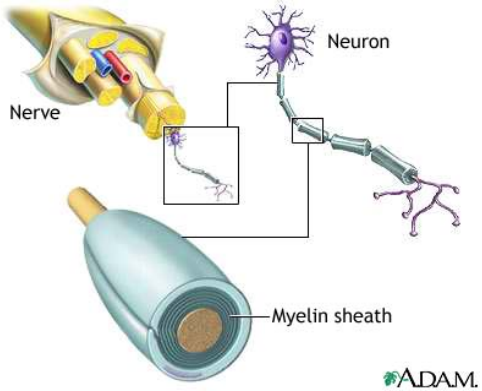
Source:
http://cartel.csail.mit.edu/doku.php

Smart Biomedical Systems

In-body smart d**evices**
-s**ensors/monitoring devices**
-d**rug delivery systems**
-m**edical robots**
-n**eural implants**

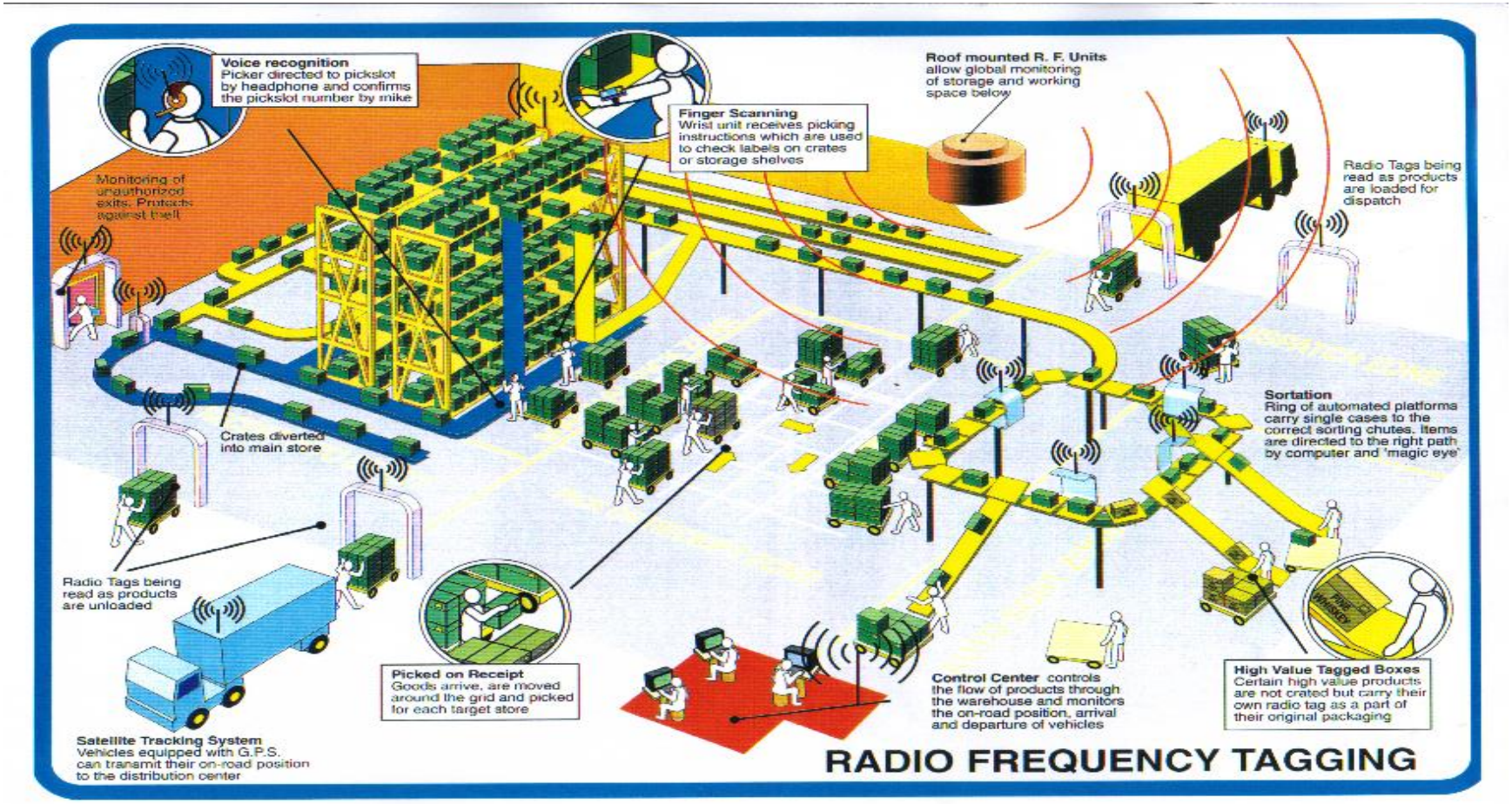


Wireless telemedicine

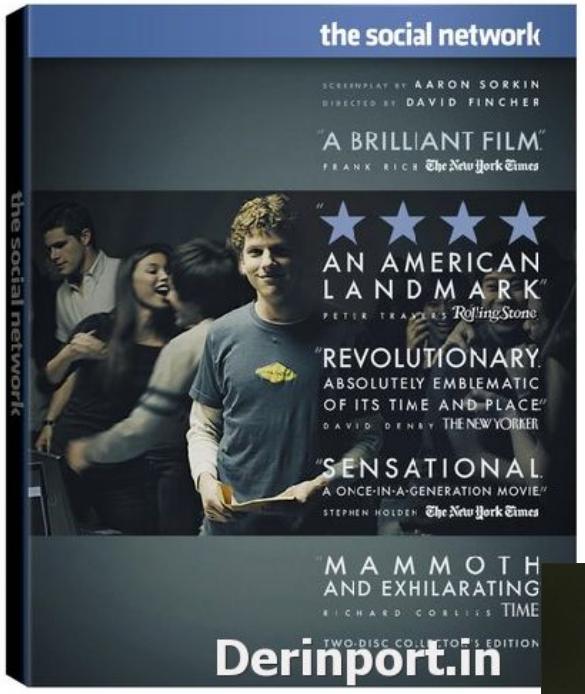


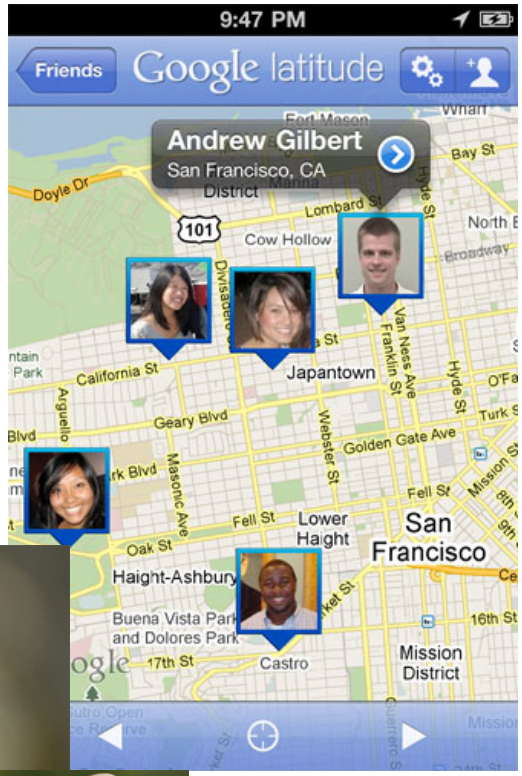
Recovery from nerve damage

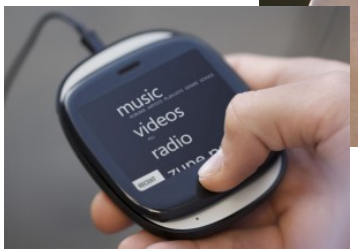
Smart Devices in Industry



Smart Devices and Mobile Social Networking







Microsoft KIN

Smartphones

Trend: everything in one small device

Smartphones - Overview

- **A smartphone** is a high-end mobile phone
- **Combines the functions of**
  - mobile phone
  - personal digital assistant (PDA)
  - portable media players
  - camera phones
  - high-resolution touchscreens
  - web browsers
  - GPS navigation
  - Wi-Fi and mobile broadband access
  - etc.
- **Feature phone vs. Smartphone**
  - Feature phone
    - proprietary firmware & limited platforms
  - Smartphone
    - open and complete mobile operating system
    - tightly integrate with the user interface and phone features
    - relies on a more powerful application programming interface (API)

Smartphones - History

Early Years:

IBM Simon (1992)

Nokia 9000 (1996)
Nokia 9110i (1998)
Nokia 9110i (2000)

Ericsson GS88 (1997)

Symbian:

Ericsson R380 (2000)
Ericsson P800 (2002)

Nokia 9210 (2000)
Nokia 9500 (2005)
Nokia E90 (2007)

Nokia N95 (2007)
Nokia N8 (2010)

Smartphones - History

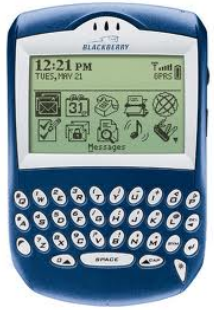
Palm, Windows, BlackBerry:

Palm Kyocera 6035 (2001)

Windows CE Pocket PC (2002)
Windows Phones 7 (2007)

Palm OS Treo (2002)

RIM BlackBerry (2002)



iPhone:

- iPhone (2007)
- iPhone 3G (2008)
- iPhone 4 (2010)
- iPhone 4S (2011)
- iPhone 5 (2012)

Android:

- Android G1 – HTC Dream (2008)
- Nexus One (2010)
- Nexus S (2011)
- Galaxy Note (2013)

Smartphones - Statistics

Worldwide Mobile Communications Device Open OS Sales to End Users by OS (thousands of units)				
OS	**2010**	**2011**	**2012**	**2015**
Symbian	111,577	89,930	32,666	661
Market Share (%)	37.6	19.2	5.2	0.1
Android	67,225	179,873	310,088	539,318
Market Share (%)	22.7	38.5	49.2	48.8
Research in Motion	47,452	62,600	79,335	122,864
Market Share (%)	16.0	13.4	12.6	11.1
iOS	46,598	90,560	118,848	189,924
Market Share (%)	15.7	19.4	18.9	17.2
Microsoft	12,378	26,346	68,156	215,998
Market Share (%)	4.2	5.6	10.8	19.5
Other Operating Systems	11,417.4	18,392.3	21,383.7	36,133.9
Market Share (%)	3.8	3.9	3.4	3.3
Total Market	**296.647**	**467,701**	**630,476**	**1,104,898**

Smartphones - Statistics

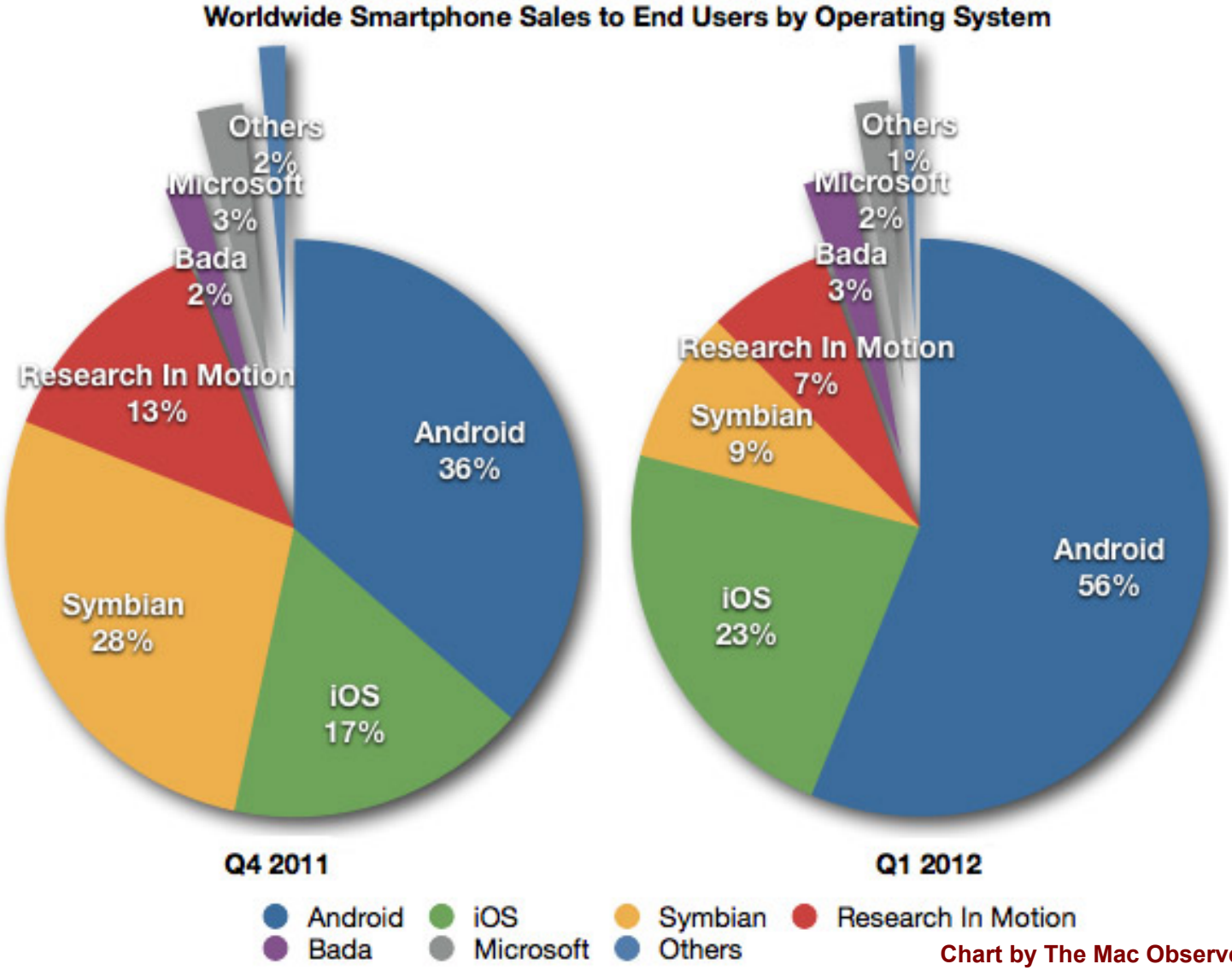


Chart by The Mac Observer, from Gartner data

Tablets - Statistics

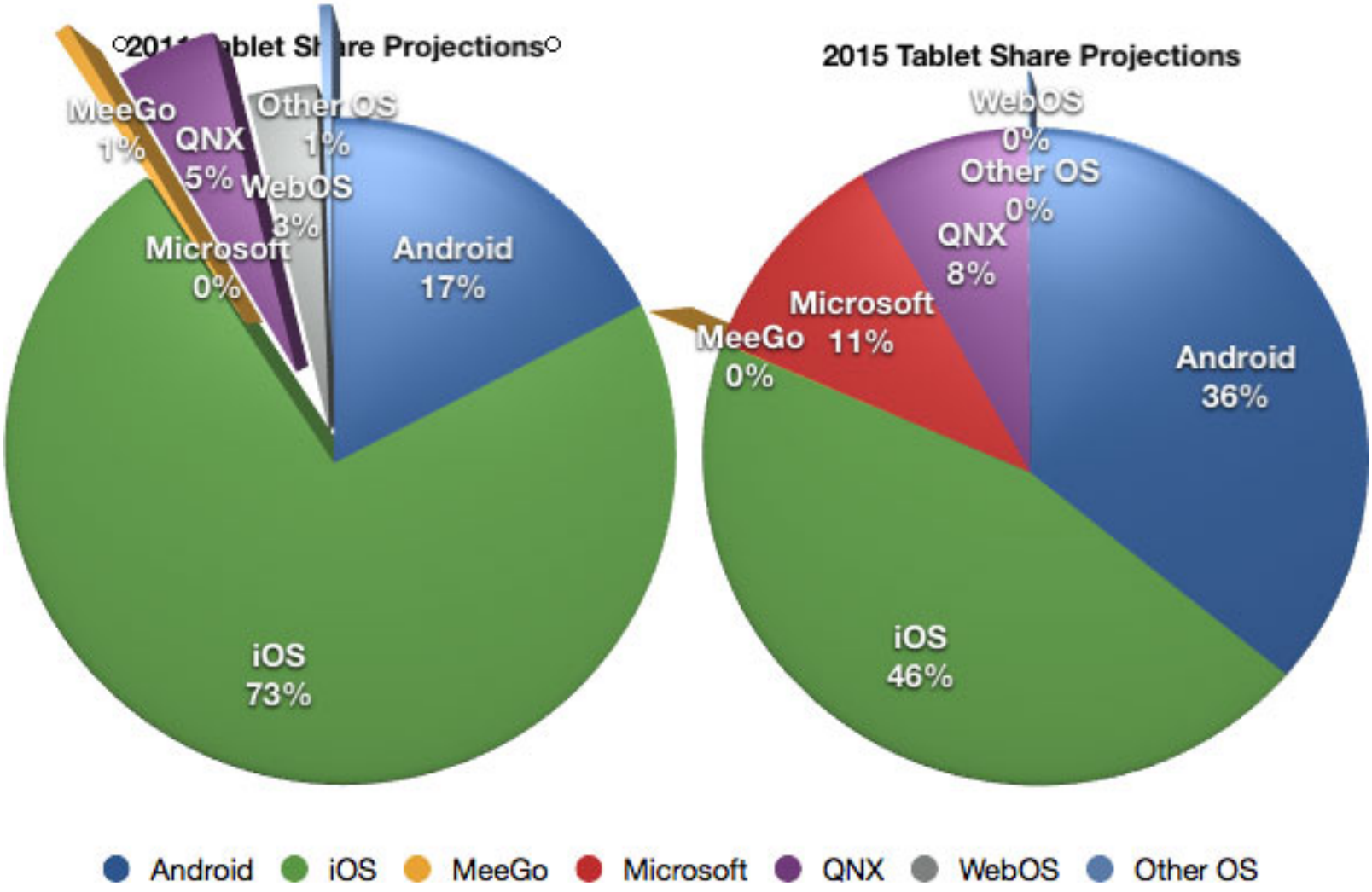
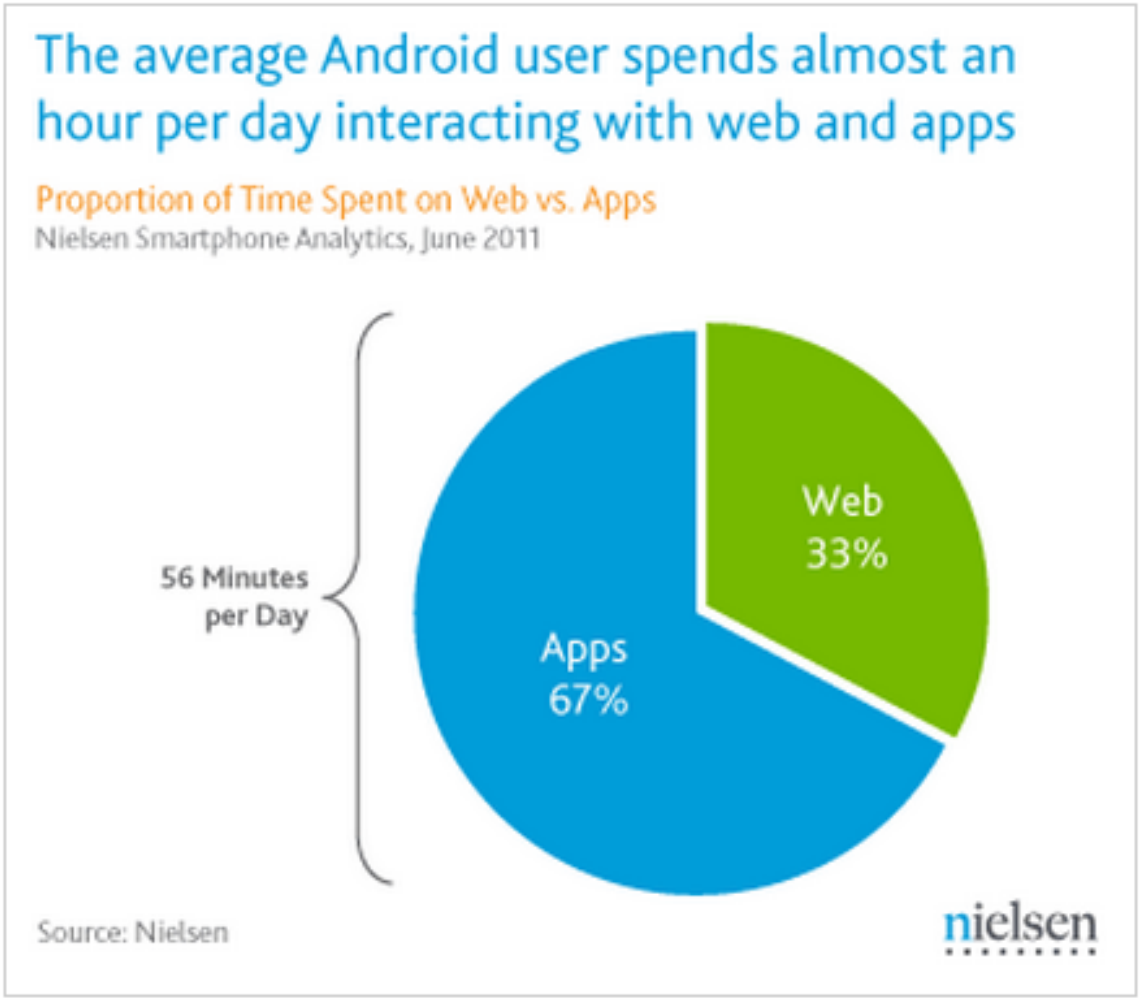


Chart by The Mac Observer, from Gartner data

Mobile Applications

- **What are they?**
  - Any application software that is developed for small low-power handheld devices such as personal digital assistants, enterprise digital assistants or mobile phones.
- **Users on mobile phone's**
  - Typically check the news, weather, email, or their social networks
  - Often have a choice between the mobile web version or a specially-created mobile app.
- **Mobile App Types**
  - Web apps: run in a web browser
    - HTML, JavaScript, Flash, server-side components, etc.
  - Native: compiled binaries for the device
    - Not cross-platform, but more interesting options

Web App vs. Native App



Native Development Environments

- **Options**
  - Java ME
  - .NET Compact Framework (C++, C#, VB.NET) for
  - Windows Mobile
  - Qualcomm's BREW (C or C++)
  - Symbian (C++)
  - BlackBerry (Java)
  - Android (Java)
  - iPhone (Objective-C)

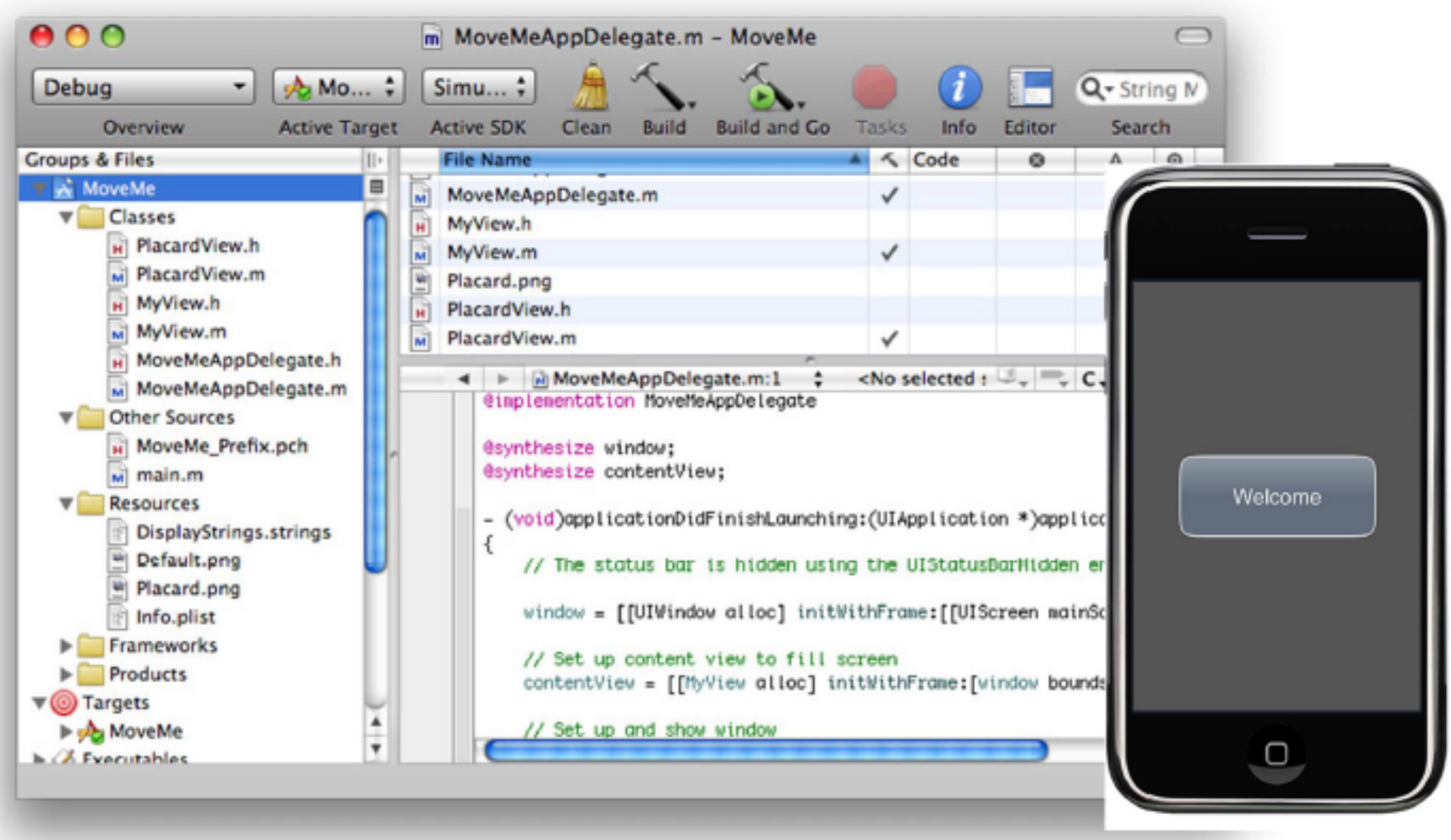
Is having so many choices and so much industry turmoil/competition a good thing?

Development Environments

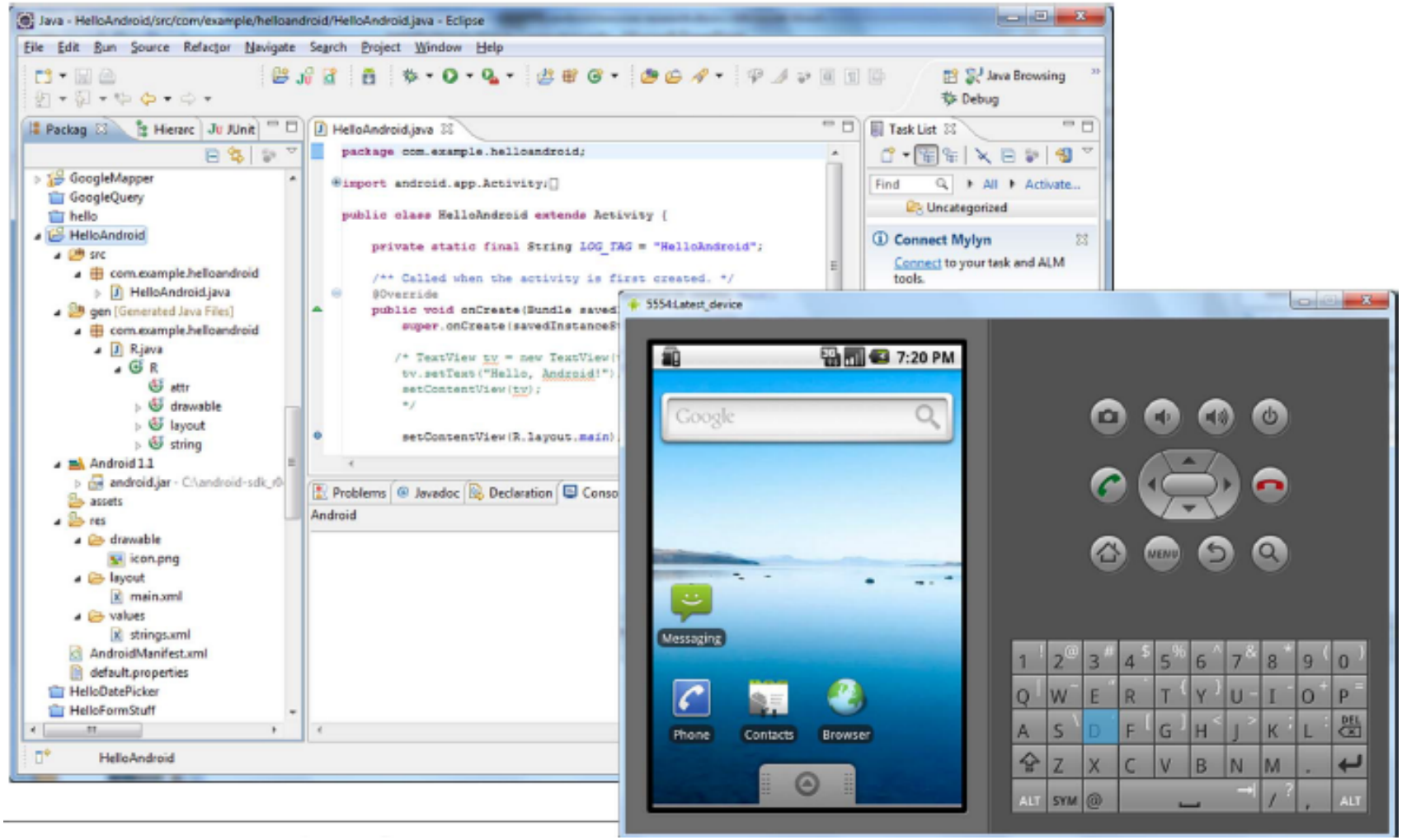
- **Most platforms have an SDK that you can download and build against**
- **Every platform has an emulator that you can use to test your apps**
- **Most emulators are configurable to match a variety of mobile devices**
  - Various screen sizes, memory limitations, tablets, etc.
  - In practice, emulators quite limited

IDE - integrated development environment that provides tools to allow a developer to write, test and deploy applications into the target platform environment.

xCode IDE & iPhone Emulator



Eclipse and Android Emulator



Smart Phone – the good

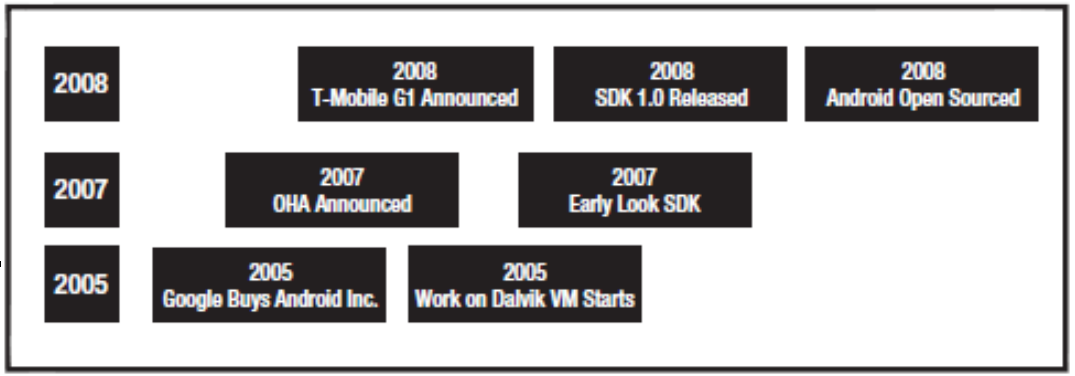
- **Always with the user**
- Increasingly powerful devices
- Typically GPS capable
- Typically have accelerometer
- Designed for communication
- 2+ types of wireless connections
- Many apps are free or low-cost

Smart Phone – the not-so-good

- **Limited battery life**
- Limited processor speed
- Limited RAM
- Limited, unreliable, and slow network access
- Limited screen size
- Limited permanent storage capacity
- Limited or awkward input
  - (none great: soft keyboard, phone keypad, touch screen, stylus, speech)
- Inconsistent platforms across devices
- High costs associated with data transfer

Android

Android



- Android, Inc. founded in Palo Alto, California in October 2003
- Google acquired Android Inc. in August 2005
- Developed a mobile device platform powered by the Linux kernel
- Google marketed the platform to handset makers and carriers on the premise of providing a flexible, upgradable system
- On November 2007, the Open Handset Alliance, a consortium of several companies (e.g., Broadcom, Google, HTC, Intel, etc. unveiled itself). The goal is to develop open standards for mobile devices.
- Open Handset Alliance unveiled their first product, Android, a mobile device platform built on the Linux kernel version 2.6
- Android OS (open source) released in October 2008

Why Android

- Simple and powerful SDK
- No licensing fees
- Excellent documentation, and a thriving developer community
- From commercial perspective
  - Requires no certification for becoming an Android developer
  - Provides the Android Market for distribution and monetization of your application
  - Has no approval process for application distribution
  - Gives you total control over your brand and access to the user's home screen

Android Version

Version	Codename	API	Distribution
1.6	Donut	4	0.2%
2.1	Eclair	7	1.9%
2.2	Froyo	8	7.6%
2.3 - 2.3.2	Gingerbread	9	0.2%
2.3.3 - 2.3.7		10	44%
3.1	Honeycomb	12	0.3%
3.2		13	0.9%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	28.6%
4.1	Jelly Bean	16	14.9%
4.2		17	1.6%

- **Initial:** 1.5 (Cupcake) (Apr 2009) , 1.6 (Donut) (Sep'09)
- **2.0/2.1 (Eclair)** (Oct'09/Jan'10)**:** new web browser, new user interface, support for HTML5, Geolocation API, enhanced camera features / voice controls, 5 homescreens, animated backgrounds.
- **2.2 (Froyo)** (May'10)**:** speed improvement, Chrome v8 JavaScript engine, Wi-Fi tethering, Adobe Flash support
- **2.3 (Gingerbread)** (Dec'10)**:** Near Field Communication
- **3.0 (Honeycomb)** (Feb'11)**:** tablet-oriented release, supports multicore processors, hardware acceleration for graphics
- **3.1 (Honeycomb)** (May'11)**:** directly transfer content from USB devices
- **3.2 (Honeycomb)** (July'11)**:** adds several new capabilities for users and developer (e.g., providing developers with more precise control over the UI)
- **4.0 (Ice Cream Sandwich)** (Oct'11)**:** combination of Gingerbread and Honeycomb
- **4.1-4.3 (Jelly Bean)** (Jul'12)**:** improve user interface (4.1), Bluetooth Low Energy support (4.3)

http://en.wikipedia.org/wiki/Android_version_history

Features and Specifications I

- Platform is adaptable to larger, VGA, 2D graphics library, 3D OpenGL graphics library
- **Storage** - SQLite, a lightweight relational database
- **Connectivity** - supports connectivity technologies including GSM/ EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth, Wi-Fi, LTE, NFC and WiMAX.
- **Messaging** – SMS, MMS, threaded text messaging, Push Messaging service.
- **Multiple language support**
- **Web browser** - based on the open-source WebKit layout engine, coupled with Chrome's V8 JavaScript engine.
- **Java support** – no Java Virtual Machine, Dalvik executables and run on Dalvik

Features and Specifications II

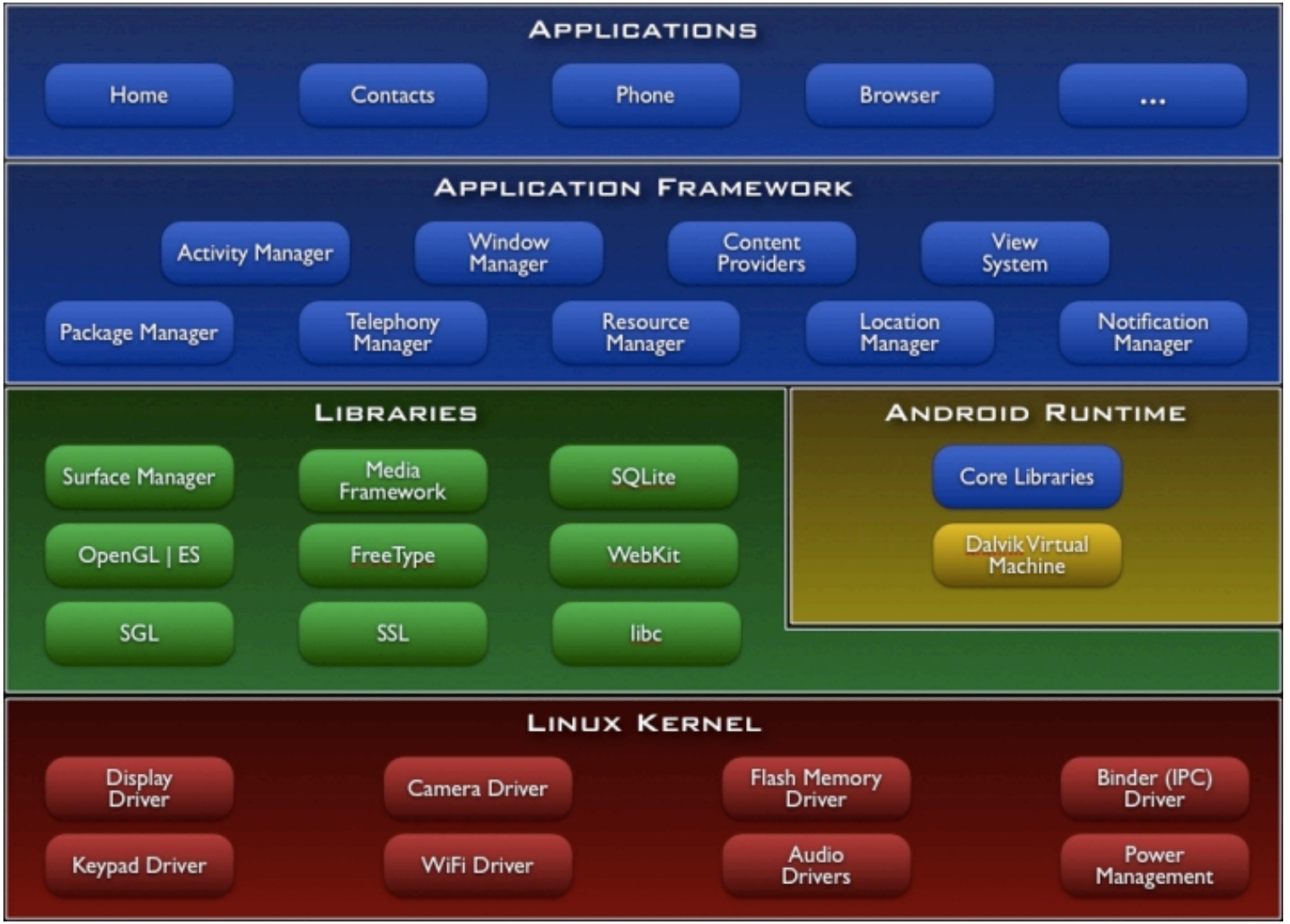
- **Media support** - audio/video/still media formats: WebM, H.263, H.264, MPEG-4 SP, WAV, JPEG, PNG, GIF, BMP, etc.
- **Streaming media support** - RTP/RTSP streaming (3GPP PSS, ISMA), HTML5 <video> tag, Adobe Flash Streaming (RTMP), HTTP Dynamic Streaming, Apple HTTP Live Streaming
- **Additional hardware support** - video/still cameras, touchscreens, GPS, accelerometers, gyroscopes, magnetometers, dedicated gaming controls, proximity and pressure sensors, thermometers, accelerated 2D bit blits and accelerated 3D graphics
- **Multi-touch**
- **Bluetooth** - supports A2DP, AVRCP, sending files (OPP), accessing the phone book (PBAP), voice dialing and sending contacts between phones. Keyboard, mouse and joystick (HID)

Features and Specifications III

- **Video calling** – no native video calling, but some handsets have a customized version of the operating system that supports it. Video calling through Google Talk is available in Android 2.3.4 and later. Skype 2.1 offers video calling in Android 2.3, including front camera support.
- **Multitasking**
- **Voice based features** - Google search through voice and voice actions for calling, texting, navigation, etc.
- **Tethering**

Google describes Android as: *"The first truly open and comprehensive platform for mobile devices, all of the software to run a mobile phone but without the proprietary obstacles that have hindered mobile innovation."*

Android Architecture



Android SDK Features

- **No licensing, distribution, or development fees**
- Wi-Fi hardware access
- GSM, EDGE, and 3G networks for telephony or data transfer, allowing you to make or receive calls or SMS messages, or to send and retrieve data across networks
- Comprehensive APIs for location-based services such as GPS
- Full multimedia hardware control including playback and recording using the camera and microphone
- APIs for accelerometer and compass hardware
- IPC message passing
- Shared data stores
- An integrated open source WebKit-based browser
- Full support for app that integrate Map controls as part of their user interface
- Peer-to-peer (P2P) support using Google Talk
- Mobile-optimized hardware-accelerated graphics including a path-based 2D graphics library and support for 3D graphics using OpenGL ES
- Media libraries for playing/recording a variety of audio/video or image formats
- **An application framework that encourages reuse of application components and the replacement of native applications**

Application Types

- **Foreground:** An application that's only useful when it's in the foreground and is effectively suspended when it's not visible.
- **Background:** An application with limited interaction that, apart from when being configured, spends most of its lifetime hidden. Examples of this include call screening applications or SMS auto-responders.
- **Intermittent:** Expects some interactivity but does most of its work in the background. Often these applications will be set up and then run silently, notifying users when appropriate. A common example would be a media player.
- **Widgets:** Interactive visual components that users can add to their home screens. Usually used to display information such as battery levels, weather forecasts, or the date and time.

Application Good behavior

- Is well behaved
- Switches seamlessly from background to foreground
- Is polite (e.g., stealing focus)
- Presents a consistent user interface
- Is responsive

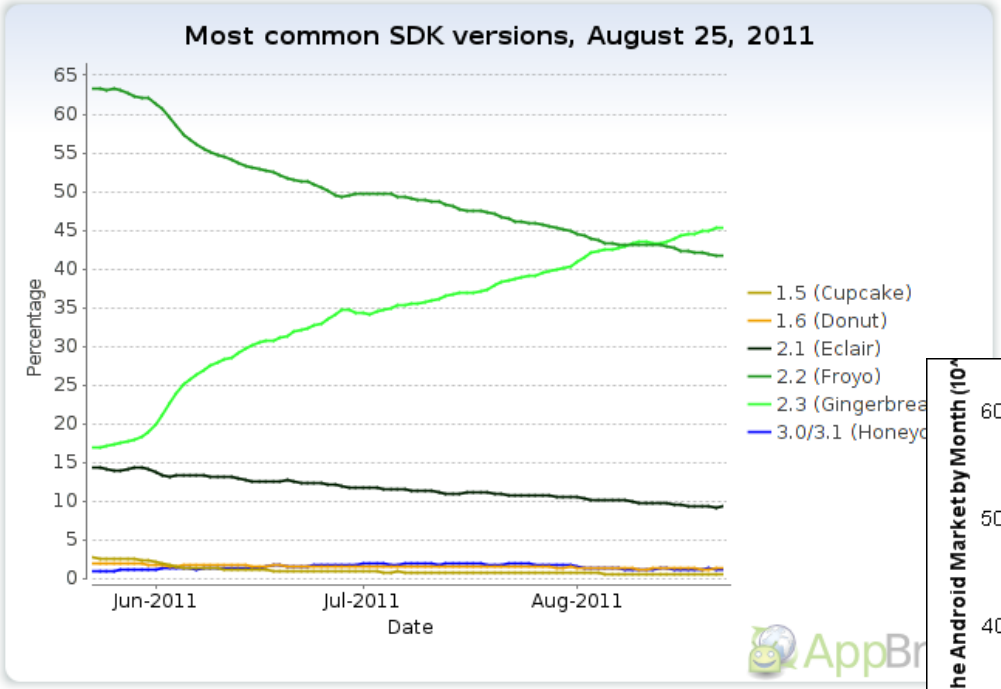
Behavior police - process assassin

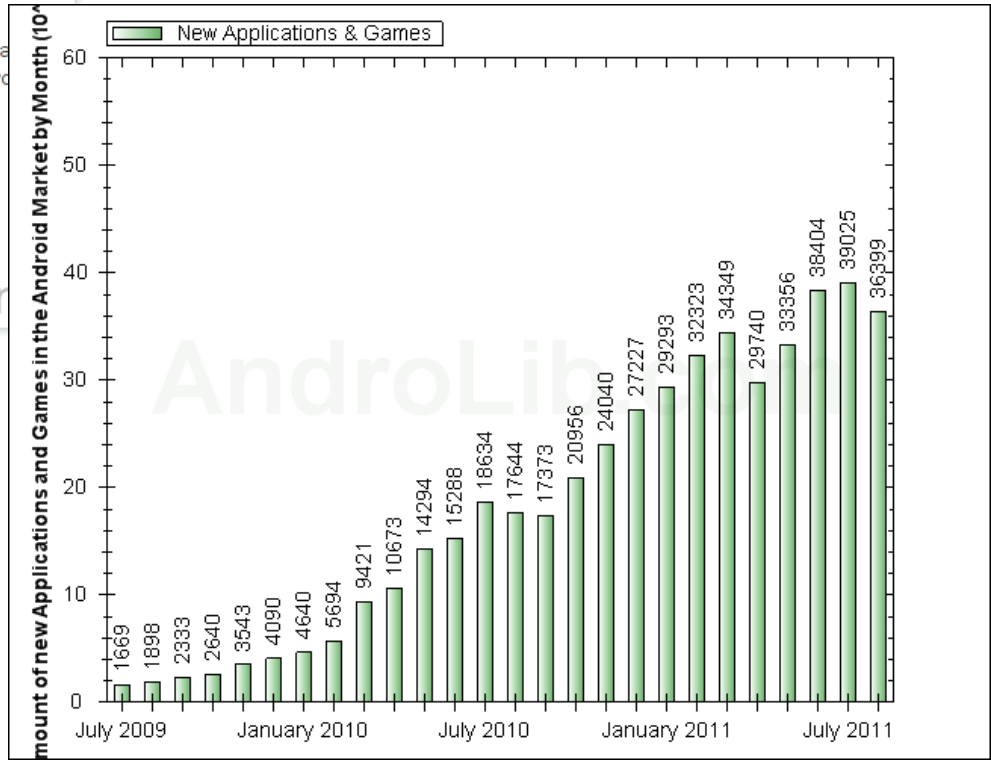
- **Two conditions monitored**
  - Must respond to any user action (e.g., key press) within 5s
  - A BroadcastReceiver must return from its OnReceive handler within 10s

Android Market (https://market.android.com/apps)

- Online software store developed by Google for Android devices
- As of December 2010 there were about 200,000 games, applications and widgets available on the Android Market.
- At end of June 2011 Google said there had been more than 6 billion Android apps installed
- The operating system itself is installed on 130 million total devices.
- Only devices that comply with Google's compatibility requirements are allowed to preinstall Google's closed-source Android Market app and access the Market.
- The Market filters the list of applications presented by the Market app to those that are compatible with the user's device,

Android Market





Android Tools

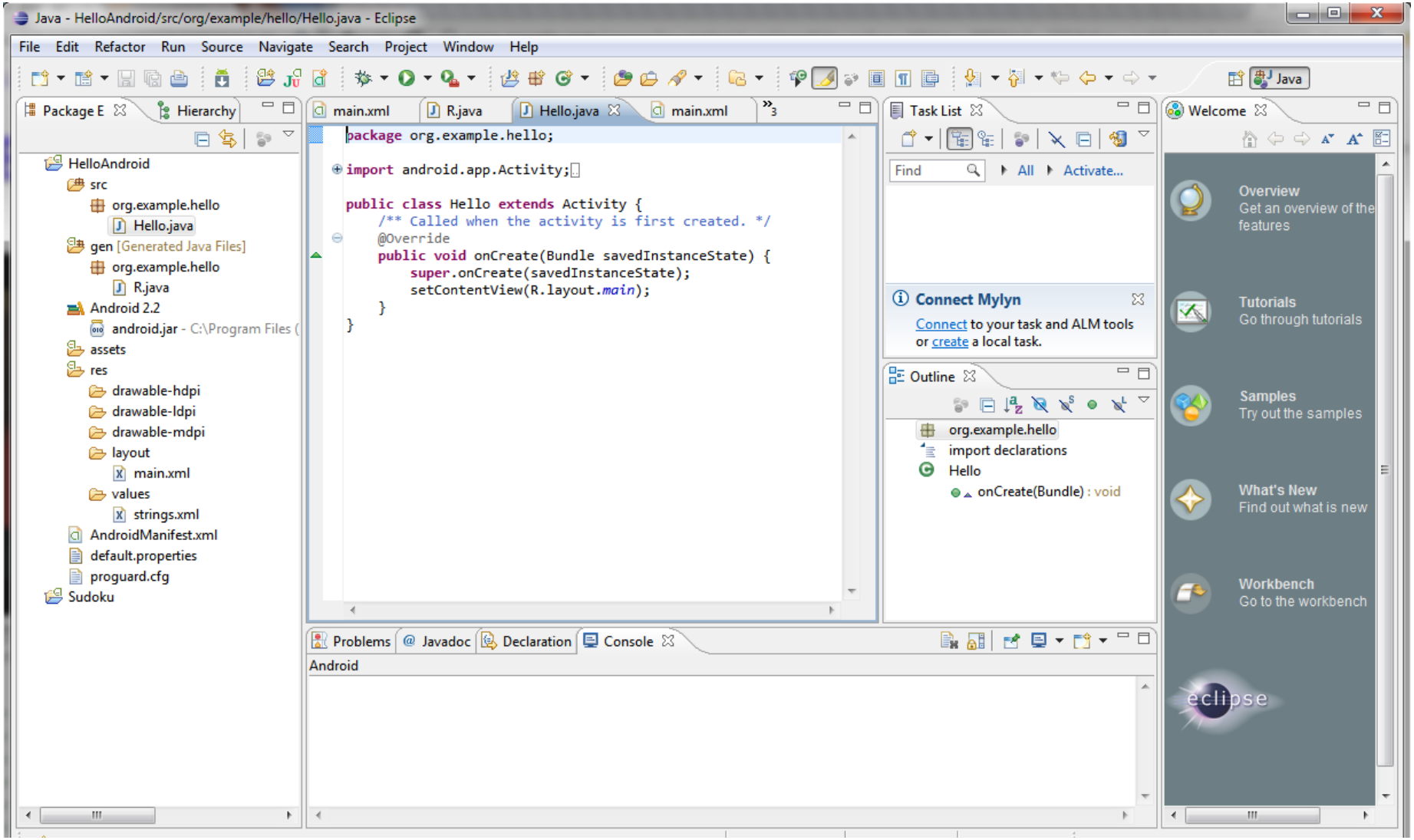
Objective

- Understand Android Tools
- Setup Android Development Environment
- Create HelloWorld Application
- Understand HelloWorld Structure
- Familiarize with Android Application Types and Components
- Understand the Role of Android Activity
- Know how to Build and Run Android Application
- Introduce Debug and Publish

Developing for Android

- Eclipse
- Android SDK
- Android Development Tools (ADT)
- Android Virtual Devices (AVD) / Emulator
- Dalvik Debug Monitor Services (DDMS)
- The Android Debug Bridge (ADB)

Eclipse


Java - HelloAndroid/src/org/example/hello/Hello.java - Eclipse
File Edit Refactor Run Source Navigate Search Project Window Help
Java
Package E
Hierarchy
HelloAndroid
src
org.example.hello
Hello.java
gen [Generated Java Files]
org.example.hello
R.java
Android 2.2
android.jar - C:\Program Files (
assets
res
drawable-hdpi
drawable-ldpi
drawable-mdpi
layout
main.xml
values
strings.xml
AndroidManifest.xml
default.properties
proguard.cfg
Sudoku
main.xml
R.java
Hello.java
main.xml
package org.example.hello;
import android.app.Activity;
public class Hello extends Activity {
/** Called when the activity is first created. */
@Override
public void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.main);
}
}
Task List
Find
All
Activate...
Connect Mylyn
Connect to your task and ALM tools or create a local task.
Outline
org.example.hello
import declarations
Hello
onCreate(Bundle) : void
Welcome
Overview
Get an overview of the features
Tutorials
Go through tutorials
Samples
Try out the samples
What's New
Find out what is new
Workbench
Go to the workbench
eclipse
Problems
Javadoc
Declaration
Console
Android

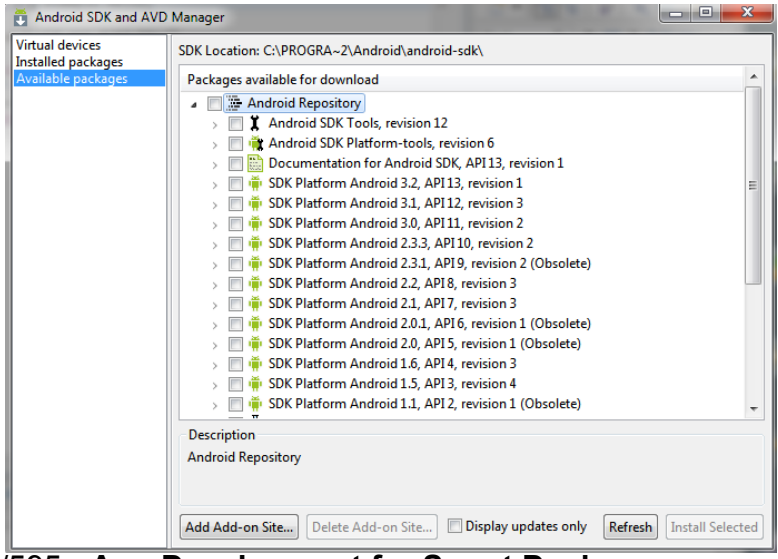
Eclipse

- **Eclipse IDE for Java Developers**
  - Open http://www.eclipse.org/
  - Select "Downloads"
  - Select Eclipse IDE for Java Developers, 149MB
  - Select Windows32

- **Extract zip file to c:\eclipse**
- **Installation Tutorial:**
  - http://archive.eclipse.org/technology/phoenix/europa/EclipseIdeForJavaDevelopers/
- **"Test First Development" Tutorial:**
  - http://archive.eclipse.org/technology/phoenix/europa/TestFirstWithEclipse/

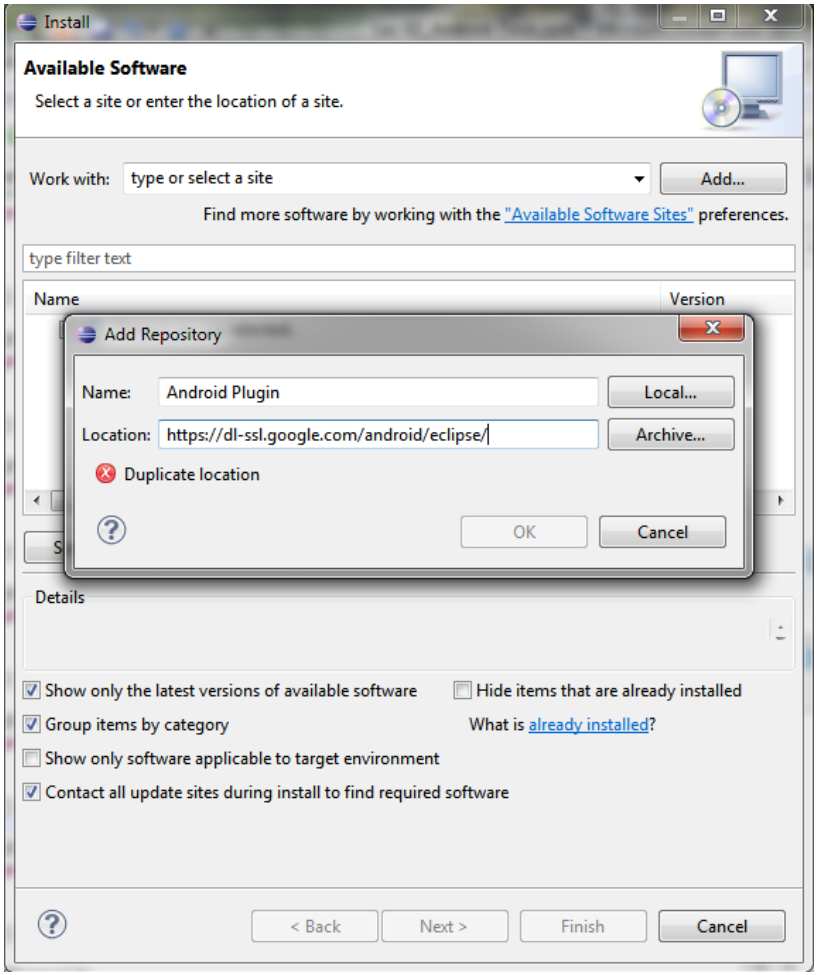
Android SDK

- **Check Your Development Computer**
  - **Supported Operating Systems**
    - Windows XP (32-bit), Vista (32- or 64-bit), or Windows 7 (32- or 64-bit)
    - Mac OS X 10.5.8 or later (x86 only)
    - Linux (tested on Ubuntu Linux, Lucid Lynx)
  - **Supported Development Environments**
    - JDK SE 6.0 – 32bit version (including JRE)
    - Eclipse IDE for Java Developers - Indigo version – 32bit version
    - Android Development Tools (ADT) plugin
- **SDK Installation**
  - **Android SDK starter package**
    - installer_r12-windows.zip
  - **Android SDK Components**
    - Run Android SDK setup
    - From “Available packages” window, select all items and then click install



ADT plugin

- Start Eclipse, then select Help > Install New Software
- Click Add, in the top-right corner.
- In the Add Repository dialog that appears, enter "ADT Plugin" for the Name and the following URL for the Location: https://dl-ssl.google.com/android/eclipse/
- Click OK
- Wait and you should see "Developer Tools" in the Available Software dialog, select the checkbox next to Developer Tools and click Next.
- In the next window, you'll see a list of the tools to be downloaded. Click Next.
- Click Finish.



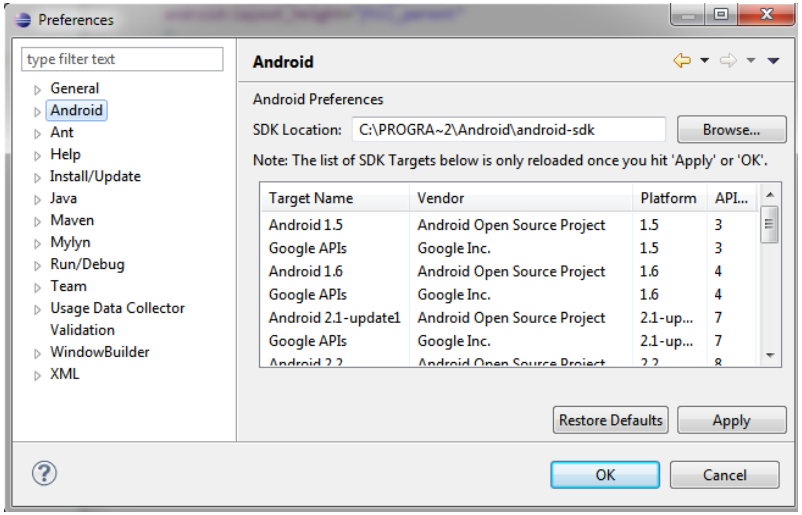
Configure ADT plugin

Modify your ADT preferences in Eclipse to point to the Android SDK directory:

1) Select Window > Preferences... to open the Preferences panel (Mac OS X: Eclipse > Preferences).

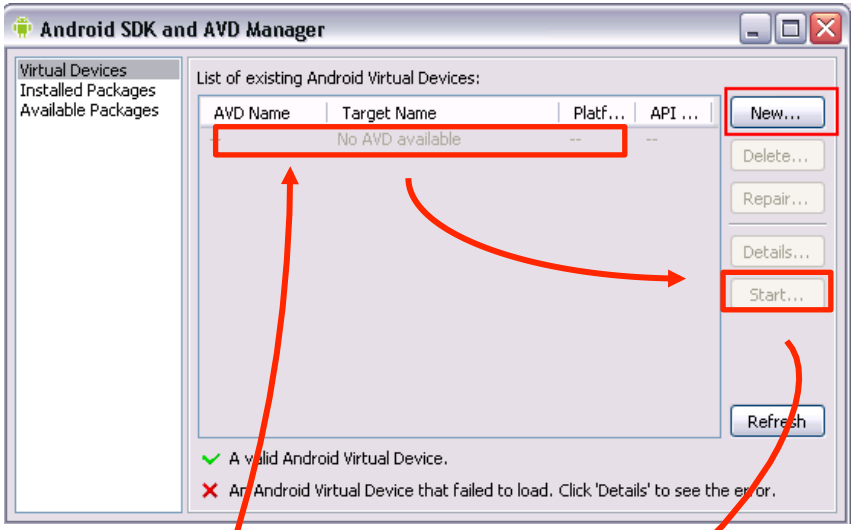
2) Select Android from the left panel.

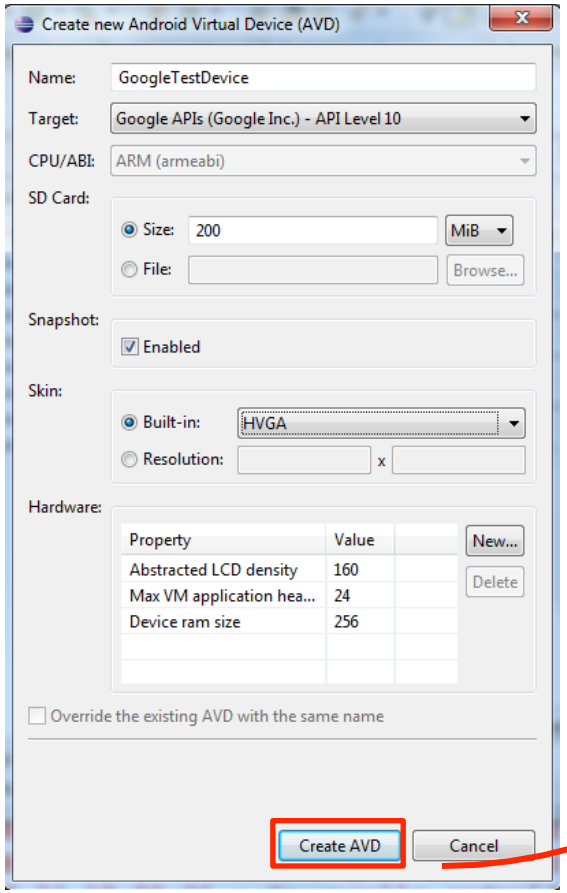
3) You may see a dialog asking whether you want to send usage statistics to Google. If so, make your choice and click Proceed. You cannot continue with this procedure until you click Proceed.

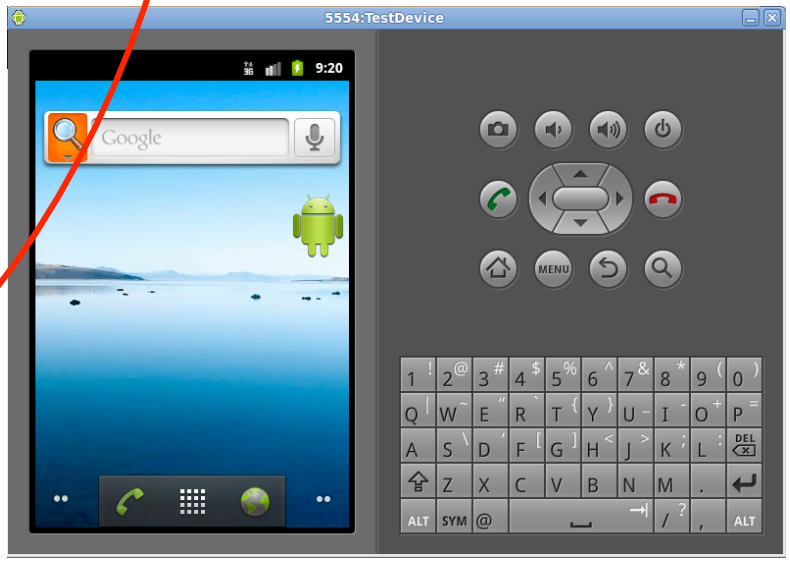


4) For the SDK Location, locate your downloaded SDK directory. Click Apply, then OK. (***Spaces in path causes an error, if you are using Windows64. For example, C:\Program Files(x86)\Android\android-sdk → C:\PROGRA~2\Android\android-sdk***)

Android Emulator Device







Android Emulator Hardware Option

Characteristic	Description
Device ram size	The amount of physical RAM on the device, in megabytes. Default value is "96".
Touch-screen support	Whether there is a touch screen or not on the device. Default value is "yes".
Trackball support	Whether there is a trackball on the device. Default value is "yes".
Keyboard support	Whether the device has a QWERTY keyboard. Default value is "yes".
DPad support	Whether the device has DPad keys. Default value is "yes".
GSM modem support	Whether there is a GSM modem in the device. Default value is "yes".
Camera support	Whether the device has a camera. Default value is "no".
Maximum horizontal camera pixels	Default value is "640".
Maximum vertical camera pixels	Default value is "480".
GPS support	Whether there is a GPS in the device. Default value is "yes".
Battery support	Whether the device can run on a battery. Default value is "yes".
Accelerometer	Whether there is an accelerometer in the device. Default value is "yes".
Audio recording support	Whether the device can record audio. Default value is "yes".
Audio playback support	Whether the device can play audio. Default value is "yes".
SD Card support	Whether the device supports insertion/removal of virtual SD Cards. Default value is "yes".
Cache partition support	Whether we use a /cache partition on the device. Default value is "yes".
Cache partition size	Default value is "66MB".
Abstracted LCD density	Sets the generalized density characteristic used by the AVD's screen. Default value is "160".
Trackball support	Whether there is a trackball present.

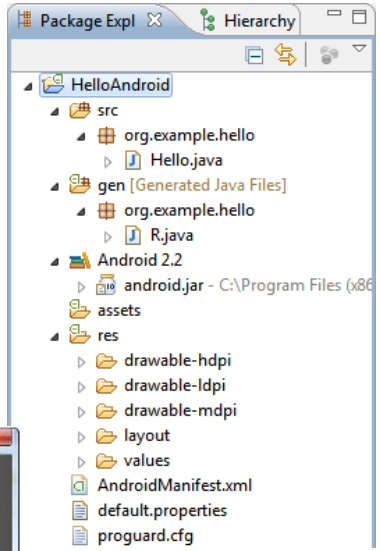
Android Emulator Controls

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 **-trace** 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	

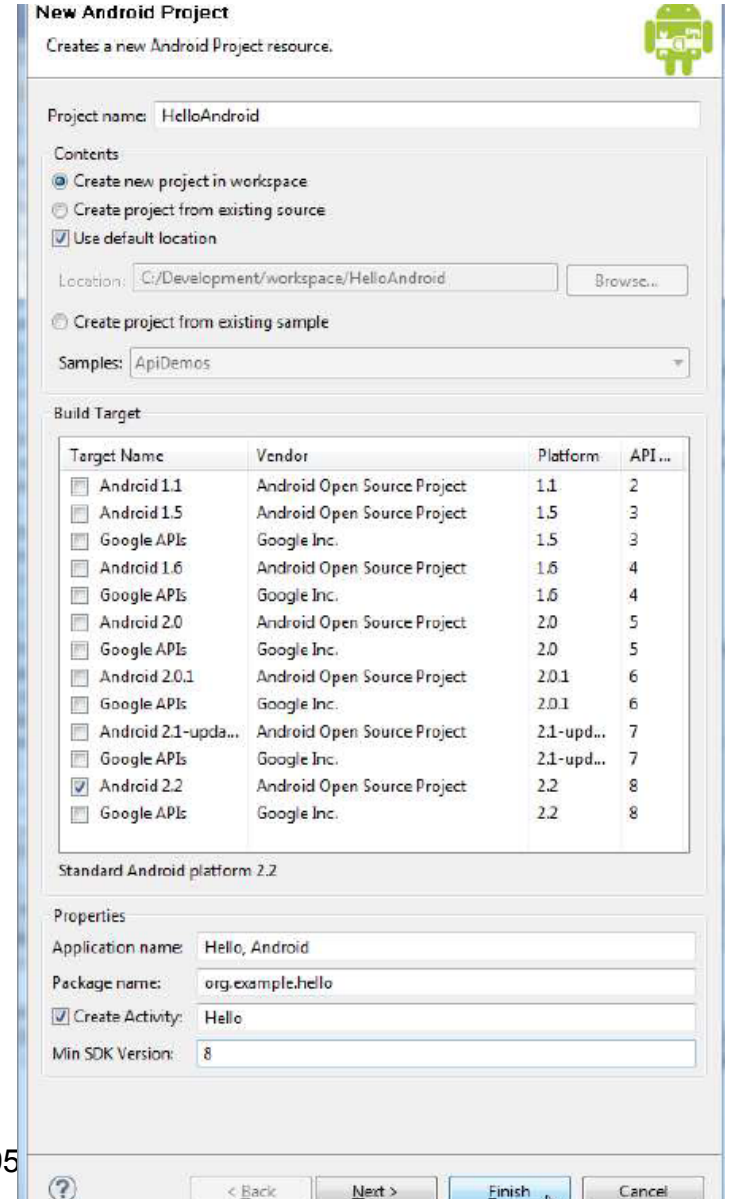
Windows Version

Your First Android Project

- Select File -> New -> Project… -> Android -> Android Project and create the Android project

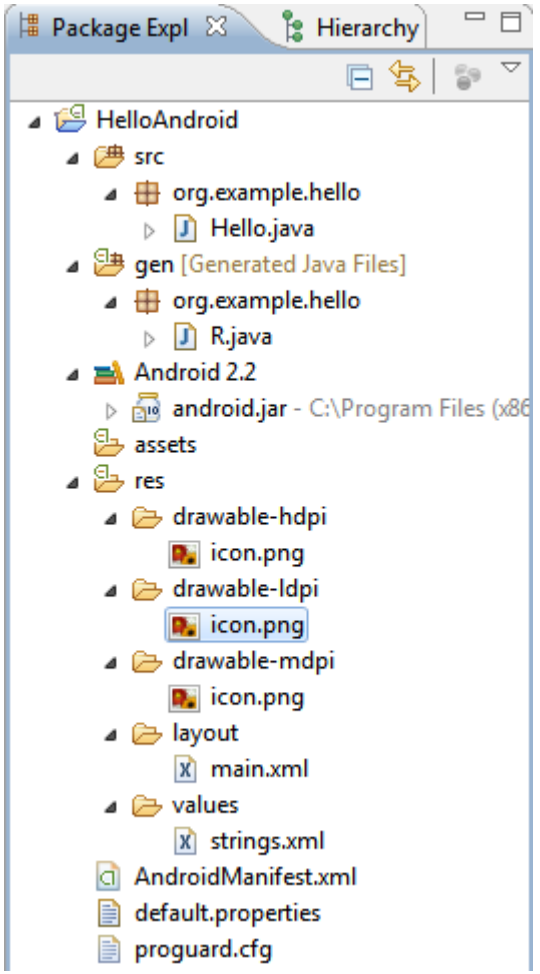


- right-click the project and select Run As > Android Application



Project Structure

- **src/** - Java packages. Each package can have multiple .java files representing different classes.
- **res/layout/** - XML files that specify the layout of each screen.
- **res/values/** - XML files used as references by other files.
- **res/drawable-hdpi/**, **res/drawable-mdpi/**, and **res/drawable-ldpi/** - high, medium, and low dots-per-inch resolution pictures.
- **res/color, res/menu, res/anim**
- **assets/** - additional non-media files.
- **AndroidManifest.xml** specifies the project to the Android OS.
- **Auto-generated files (do not modify):**
  - **gen/** contains auto-generated code. Class **R.java** generated by **Android Asset Packaging Tool (aapt).**
  - **default.properties** contains project settings.



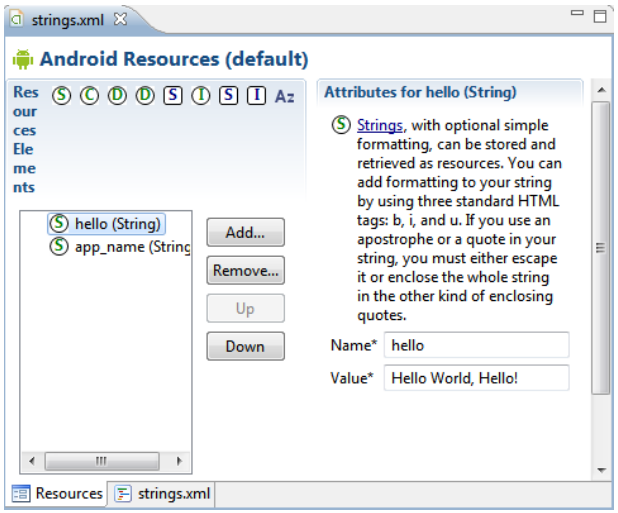
Project Structure

Hello.java

```
package org.example.hello;

import android.app.Activity;

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

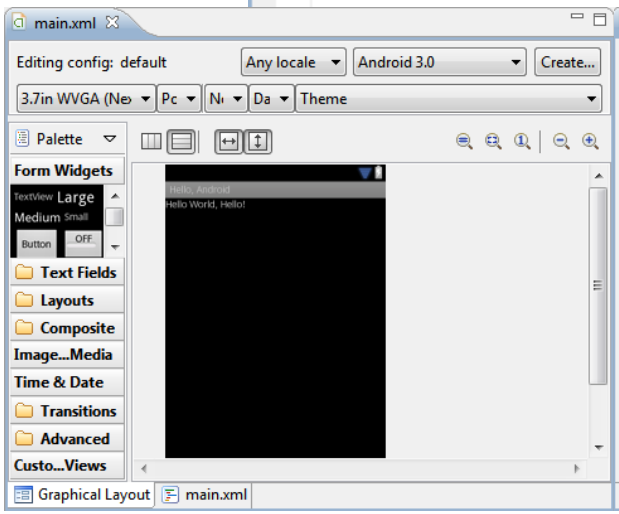
R.java

```
/* AUTO-GENERATED FILE.  DO NOT MODIFY.

package org.example.hello;

public final class R {
    public static final class attr {
    }
    public static final class drawable {
        public static final int icon=0x7f020000;
    }
    public static final class layout {
        public static final int main=0x7f030000;
    }
    public static final class string {
        public static final int app_name=0x7f040001;
        public static final int hello=0x7f040000;
    }
}
```



strings.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="hello">Hello World, Hello!</string>
    <string name="app_name">Hello, Android</string>
</resources>
```



main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/
    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>
```

Graphical Layout | main.xml

Resources, Package, Manifest

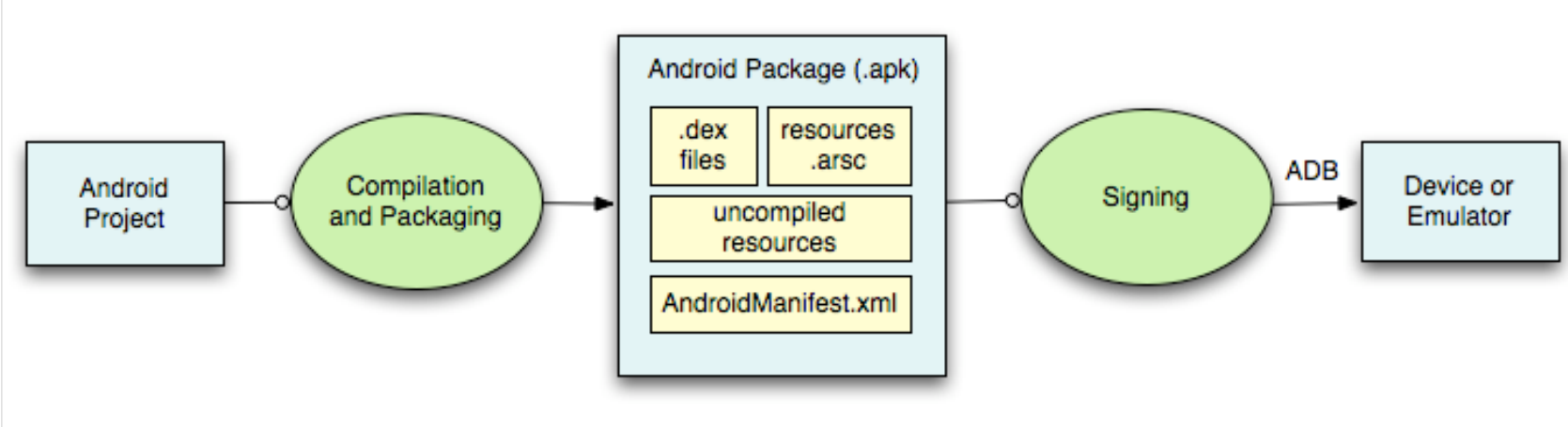
Resource	Reference in Java	Reference in XML
res/layout/main.xml	`R.layout.main`	`@layout/main`
res/drawable-hdpi/icon.png	`R.drawable.icon`	`@drawable/icon`
`@+id/home_button`	`R.id.home_button`	`@id/home_button`
`<string name="hello">`	`R.string.hello`	`@string/hello`

HelloAndroid Manifest

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
      package="org.example.hello"
      android:versionCode="1"
      android:versionName="1.0">
    <uses-sdk android:minSdkVersion="8" />

    <application android:icon="@drawable/icon" android:label="@string/app_name">
        <activity android:name=".Hello"
                  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>
</manifest>
```

Building Android Project

- **Eclipse outputs an .apk file automatically to the bin folder of the project**
- Contains all information to run the application, such as compiled .dex files, a binary version of AndroidManifest.xml, compiled resources (resources.arsc) and uncompiled resource files.
- Eclipse automatically builds application, enables debugging and signs the .apk with a debug key, by default.



Running

- **Running on the emulator**
  - To run (or debug) your application, select **Run** > **Run** (or **Run** > **Debug**) from the Eclipse menu bar.
  - Be certain to create multiple AVDs upon which to test your application.

- **Enable Running on a device**
  - Ensure application is debuggable by setting ***android:debuggable="true"*** of the<application> element in AndroidManifest.xml.
  - Set up the device to allow installation of non-Market applications. On the device, go to **Settings > Applications** and enable **Unknown sources**.
  - Enable USB Debugging on the device. On the device, **Settings > Applications > Development > USB debugging**.
  - For Windows, use **Device Manager** to install a **USB driver for adb** tool (<android-sdk>\extras\google\usb_driver)

Manual Application Installation

- **Manual Installation using adb**
  - App Package: <workspace>\<Project Name>\bin\<Project Name>.apk
  - Need adb tool located at <sdk>/platform-tools/
  - Modify Device settings as in “Running on a Device”
  - Connect the device to your host machine through USB
  - In Command Prompt, type: adb [-d | -e] install <path>/<file>.apk

- **Manual Uninstall**
  - On device, go to Applications Menu ->Settings->Manage Applications.
  - Select the application which you want to Uninistall, and click on the ‘Uninstall’ button.

Questions?

Assignment #1: Warming UP

- **Due Sun Sep 1st, 11:59pm**

Requirement #1:

- Select three high-quality apps from the iPhone or Android Market that are related to one of the following application domains:
  - Increase driving safety
  - Traffic monitoring
  - Enhance education experience
  - Monitor/support personal health
  - Monitor/save energy consumption
  - Support smart environments
- Pick applications that are innovative in some way (e.g., use new phone functionality, easy-to-use, popular, or take an approach that is very different from competitor apps.
- Critically evaluate the three apps

Assignment #1: Warming UP

- For each app, create a one-page pdf write-up that includes the following:
  - Your name and the name of the app evaluated.
  - A summary paragraph that identifies key features of the app.
  - A paragraph or bullet list describing what makes this app better than competitors. Why did you select it?
  - A list of positive characteristics (e.g. high-quality graphics, fun, indispensable tool). Try to be as precise as possible about what makes it good. For example, don't just say it is addictive; try to explain why it is addictive. Don't just say that it "looks professional." Explain what makes it look that way.
  - A list of negative characteristics (e.g. force close, slow, confusing menu titles). What could be done better? Be detailed
  - A paragraph identifying the target audience for the app. Who might use this?
  - A paragraph discussing what additional features and functionalities you would add to enhance it.

- **Few samples will be selected "randomly" to be presented and discussed in upcoming classes (be prepared).**

Assignment #1: Warming UP

Requirement #2:

- Send me an ordered list of only 3 recent reports/articles/papers you mostly liked about phone applications/services/features!
  - Again use instincts/guts
  - Your presentation paper could be among this list
  - Where to search:
    - Internet
    - Conferences: ACM PhoneSense, Mobicom, MobiSys, HotMobile, IEEE SECON, IEEE PerCom
    - Magazines

- **Submission Format:**
  - Send all pages in a single PDF file
  - Have a cover page with your name and your email
  - Send the file as an attachment to me
  - Make the subject line of the email: **cs495_assignment_1**

Recommended Assignment

- **Refresh your Java programming**
  - http://docs.oracle.com/javase/tutorial/java/
    - http://docs.oracle.com/javase/tutorial/java/javaOO/index.html
- **Have your Android development environment setup**
- **Programming: try out and explore "Hello World"**
- **Look for announcement about the reading papers. Send me your choice of 4 papers by Sep 06, 2013**