



CMPE 277

# Introduction to Android

Instructor: Charles Zhang

Fall 2017



# Outline

1. What is Android?
2. Why develop for mobile and Android?
3. Android versions
4. Android SDK features
5. Android Architecture
6. First Android Application - Hello World
7. Anatomy of an Android Application



Nexus 4, 5, 6, 7, 9, ...



# Pixel, Phone by Google

# What is Android?



Android is an operating system based on **Linux** with a **Java programming interface**. It provides tools, e.g. a compiler, debugger and a device emulator as well as its own Java Virtual machine (Dalvik Virtual Machine - **DVM**). Android is created by the Open Handset Alliance (**OHA**), led by Google.

# What is Android?

Watch:

[Google I/O 2008 -Dalvik Virtual Machine Internals](#)

## What is the Dalvik VM?



It is a virtual machine to...

- run on a slow CPU
- with relatively little RAM
- on an OS without swap space

Java language compiles to  
-> Dalvik byte-code which runs on  
-> Dalvik virtual machine  
-> Inside the Android OS (Linux-based)



Android used to use a special Java virtual machine (Dalvik) which is based on the **Apache Harmony** Java implementation. Dalvik uses **special bytecode**. Therefore you **cannot run standard Java bytecode** on Android. Android provides a tool "dx" which allows to convert Java Class files into "**dex**" (DalvikExecutable) files. Android applications are then packed into an **.apk**(Android Package) file.

# Android Runtime (ART)

- ❑ Android runtime (ART) is the managed runtime used by applications and some system services on Android
- ❑ ART and its predecessor Dalvik were originally created specifically for the Android project
- ❑ ART as the runtime executes the Dalvik Executable format and Dex bytecode specification
- ❑ ART and Dalvik are compatible for Dex bytecode, but not 100%

Source: <https://source.android.com>

# Why develop Mobile and Android?



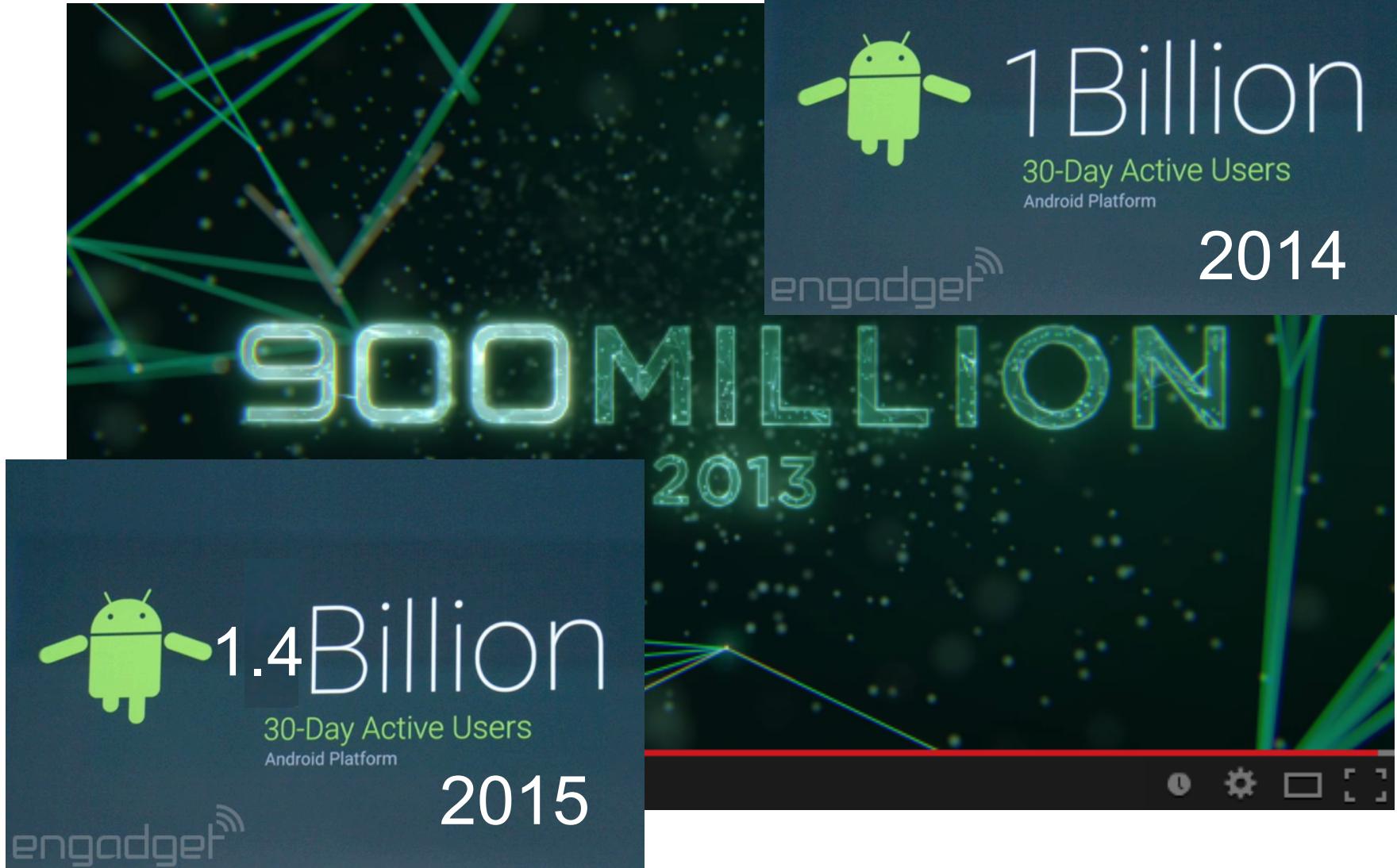
# The world's most popular mobile platform

*Easily optimize a single binary for phones, tablets, and other devices.*

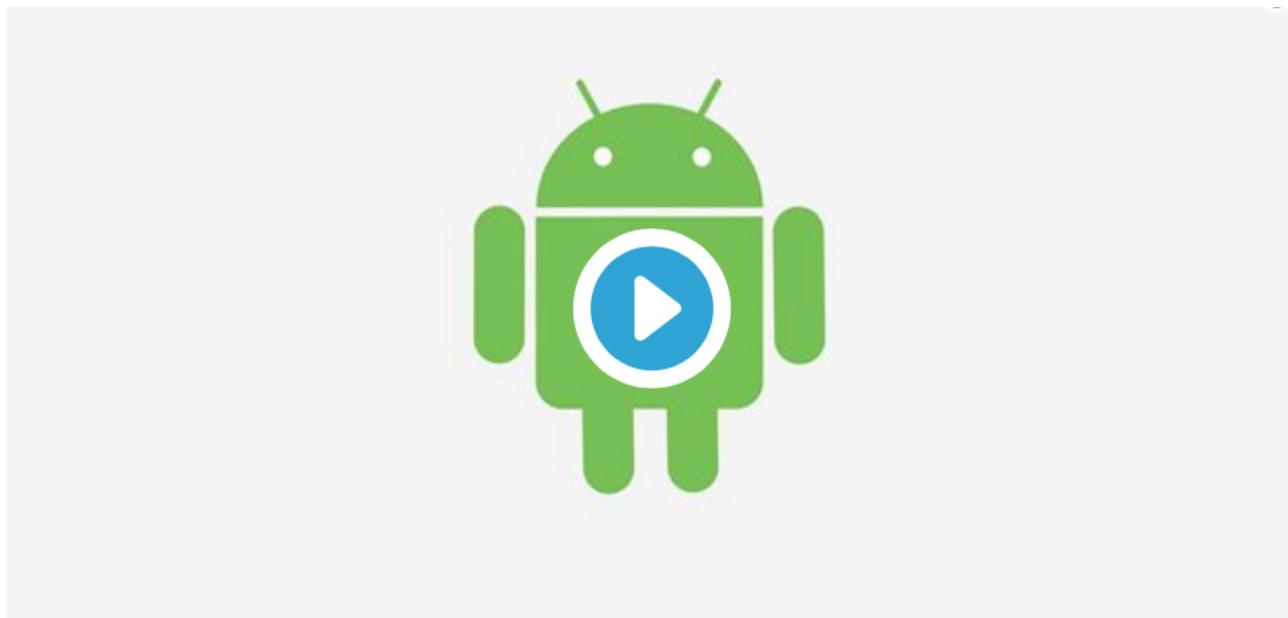
*1.5 billion downloads a month and growing. Get your apps in front of millions of users at Google's scale.*



# Why develop Mobile and Android?



# What about 2017?



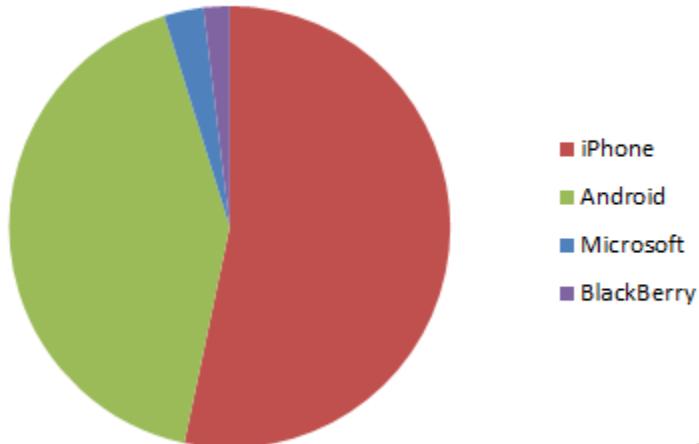
Thanks to developers and our partners around the world, there are now more than 2 billion monthly active Android devices.

#io17

10:09 AM - May 17, 2017

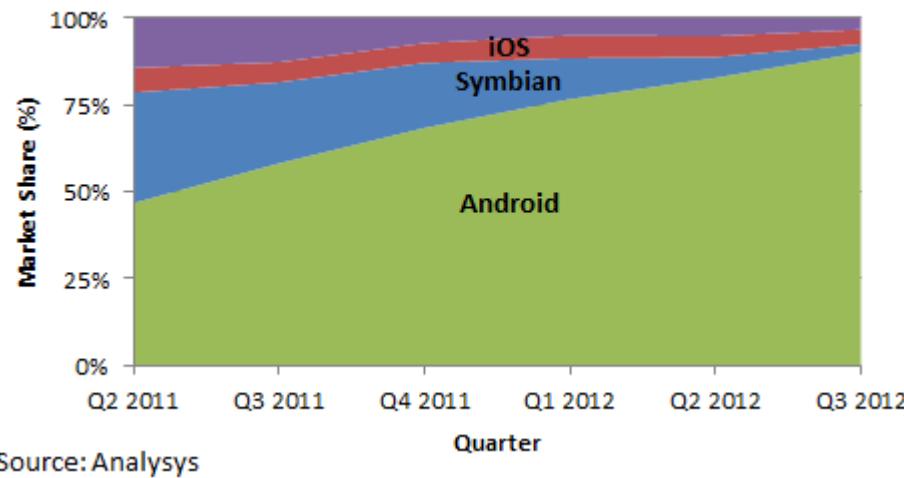
# The world's most popular mobile platform

U.S. Smartphone Market Share



Source: Kantar Worldpanel ComTech

Chinese Smartphone Market Share



Source: Analysys

# Newer shares for worldwide smartphone market

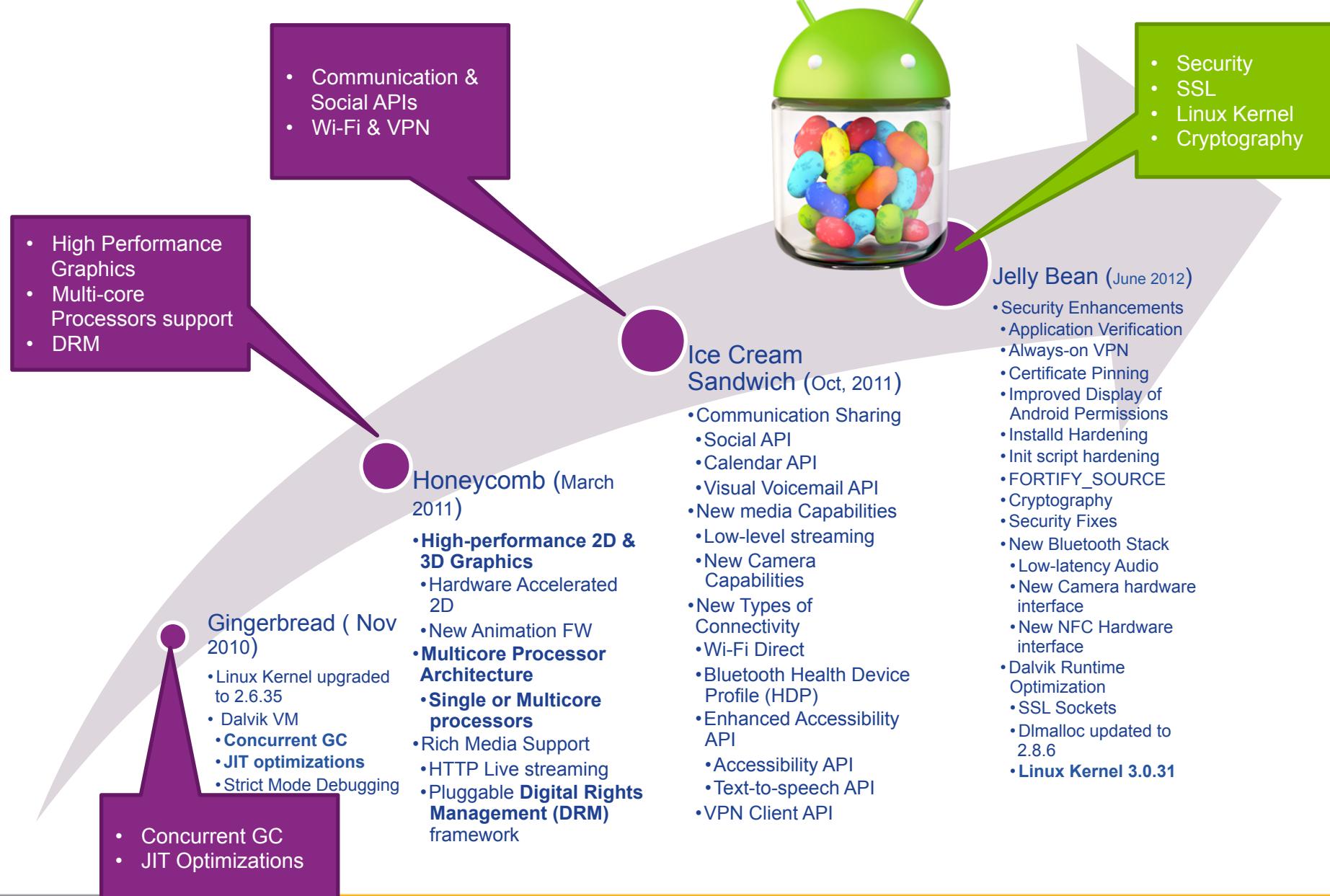
Period	Android	iOS	Windows Phone	Others
2015Q4	79.6%	18.7%	1.2%	0.5%
2016Q1	83.5%	15.4%	0.8%	0.4%
2016Q2	87.6%	11.7%	0.4%	0.3%
2016Q3	86.8%	12.5%	0.3%	0.4%

Source: IDC, Nov 2016

# Android Versions

Platform Version	Version Code	Release Date
Android 1.0	BASE	October 2008: The original, first, version of Android.
Android 1.1	BASE_1_1	February 2009: First Android update, officially called 1.1.
Android 1.5	CUPCAKE	May 2009: Android 1.5.
Android 1.6	DONUT	September 2009: Android 1.6.
Android 2.0	ECLAIR	November 2009: Android 2.0
Android 2.0.1	ECLAIR_0_1	December 2009: Android 2.0.1
Android 2.1.x	ECLAIR_MR1	January 2010: Android 2.1
Android 2.2.x	FROYO	June 2010: Android 2.2
Android 2.3.2	GINGERBREAD	November 2010: Android 2.3
Android 2.3.1		
Android 2.3		
Android 2.3.4	GINGERBREAD_MR1	February 2011: Android 2.3.3.
Android 2.3.3		
Android 3.0.x	HONEYCOMB	February 2011: Android 3.0.
Android 3.1.x	HONEYCOMB_MR1	May 2011: Android 3.1.
Android 3.2	HONEYCOMB_MR2	June 2011: Android 3.2.
Android 4.0, 4.0.1, 4.0.2	ICE_CREAM SANDWICH	October 2011: Android 4.0.
Android 4.0.3, 4.0.4	ICE_CREAM_SANDWICH_MR1	December 2011: Android 4.0.3.
Android 4.1, 4.1.1	JELLY_BEAN	June 2012: Android 4.1.
Android 4.2	JELLY_BEAN_MR1	Android 4.2: Moar jelly beans!
Android 4.3	JELLY_BEAN_MR2	
Android 4.4	KITKAT	

# Android Platform Technologies



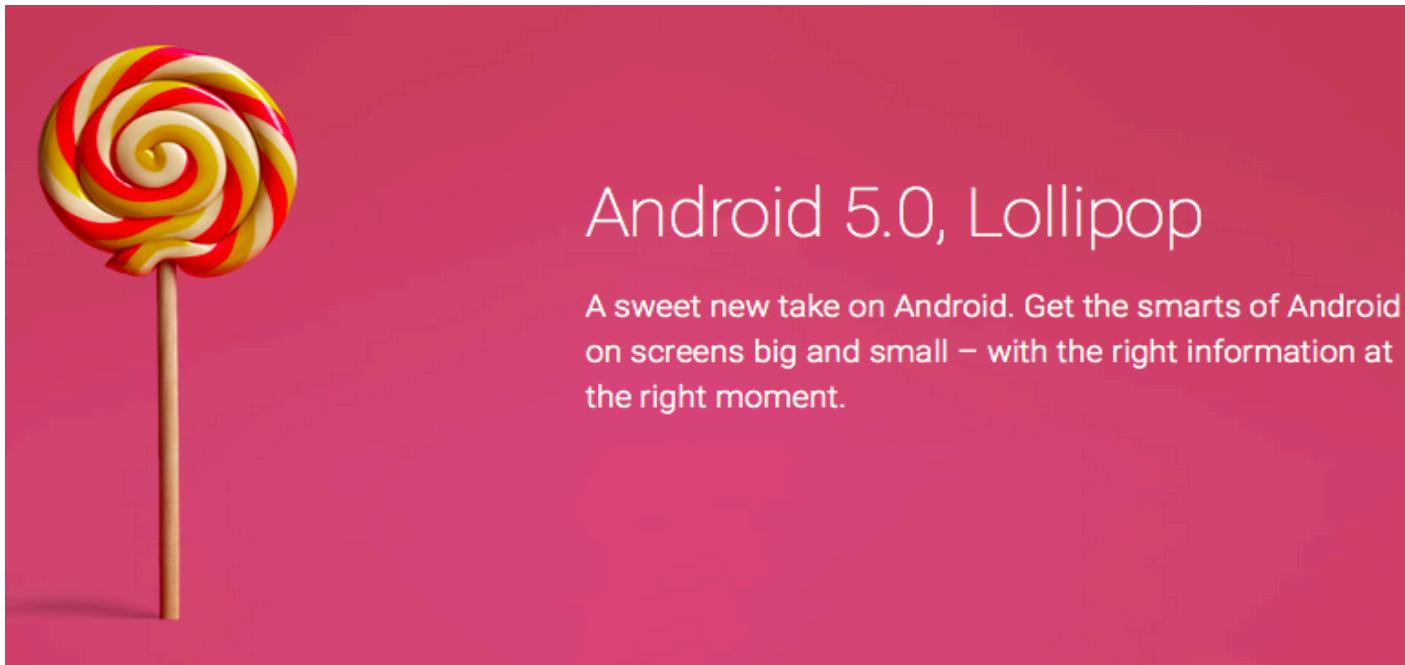
# Android KitKat

## Making Android for everyone

Android 4.4 is designed to run fast, smooth, and responsively on a much broader range of devices than ever before — including on millions of entry-level devices around the world that have as little as **512MB RAM**.



# Who is the newer kid in town?



## Android 5.0, Lollipop

A sweet new take on Android. Get the smarts of Android on screens big and small – with the right information at the right moment.

- Material Design
  - New 3D views, built-in transition, ripple animations
- Dalvik → Android Runtime (ART)
- Tools for building battery-efficient apps

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

# Who is the even newer kid in town?

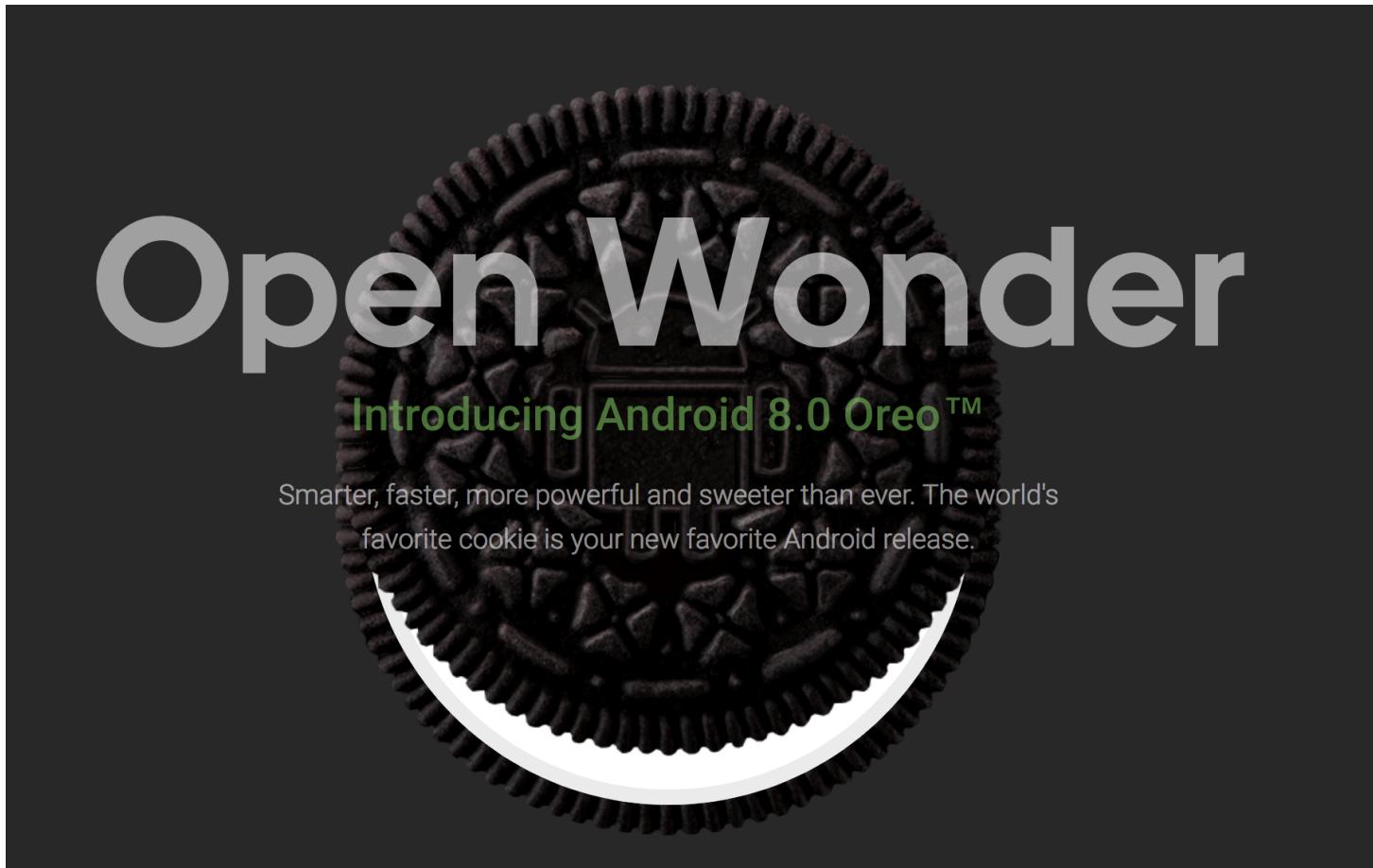


Android  
Nougat!

- Better battery life
- Faster charging
- User-driven privacy control
- Many more feature...

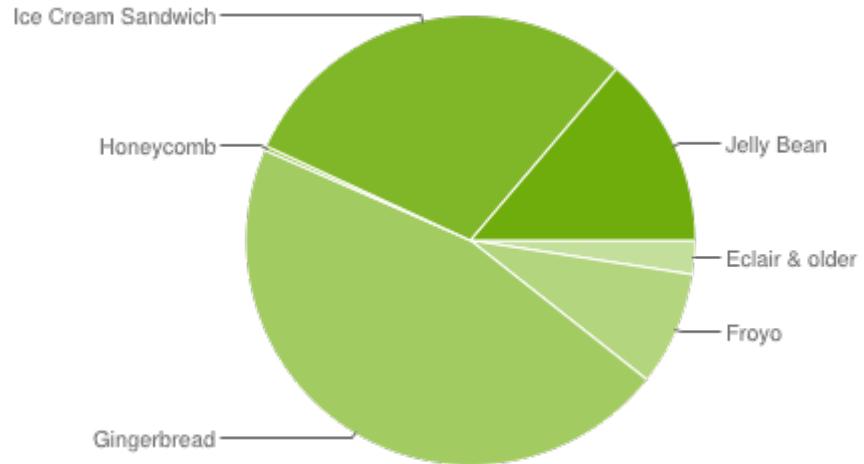
Source: <http://www.android.com>

# Android Oreo, 2017



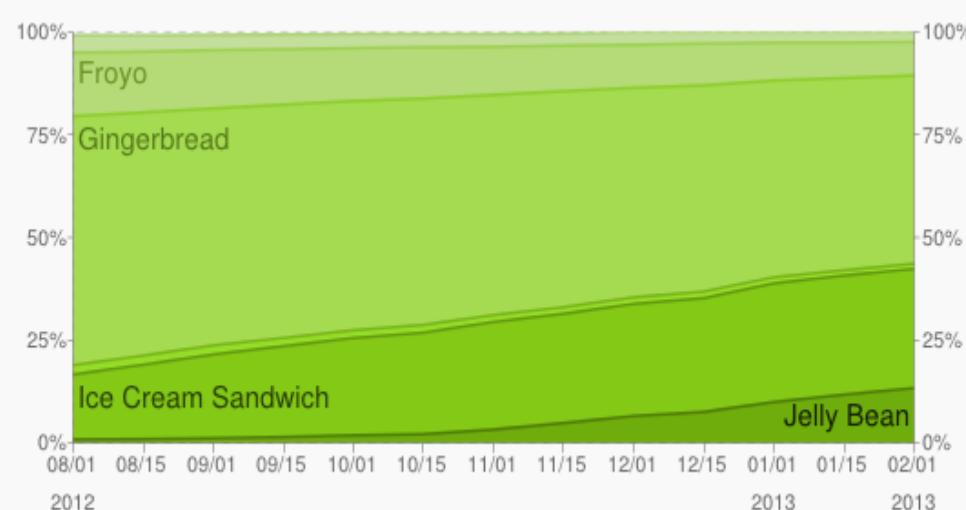
# Current Distribution

Platform Version	Version Code	Distribution
Android 1.6	DONUT	0.2%
Android 2.0	ECLAIR	2.2%
Android 2.2.x	FROYO	8.1%
Android 2.3.2	GINGERBREAD	0.2%
Android 2.3.1		
Android 2.3		
Android 2.3.4	GINGERBREAD_MR1	45.4%
Android 2.3.3		
Android 3.0.x	HONEYCOMB	0.3%
Android 3.2	HONEYCOMB_MR2	1.0%
Android 4.0, 4.0.1, 4.0.2	ICE_CREAM_SANDWI CH	29.0%
Android 4.1, 4.1.1	JELLY_BEAN	12.2%
Android 4.2	JELLY_BEAN_MR1	1.4%

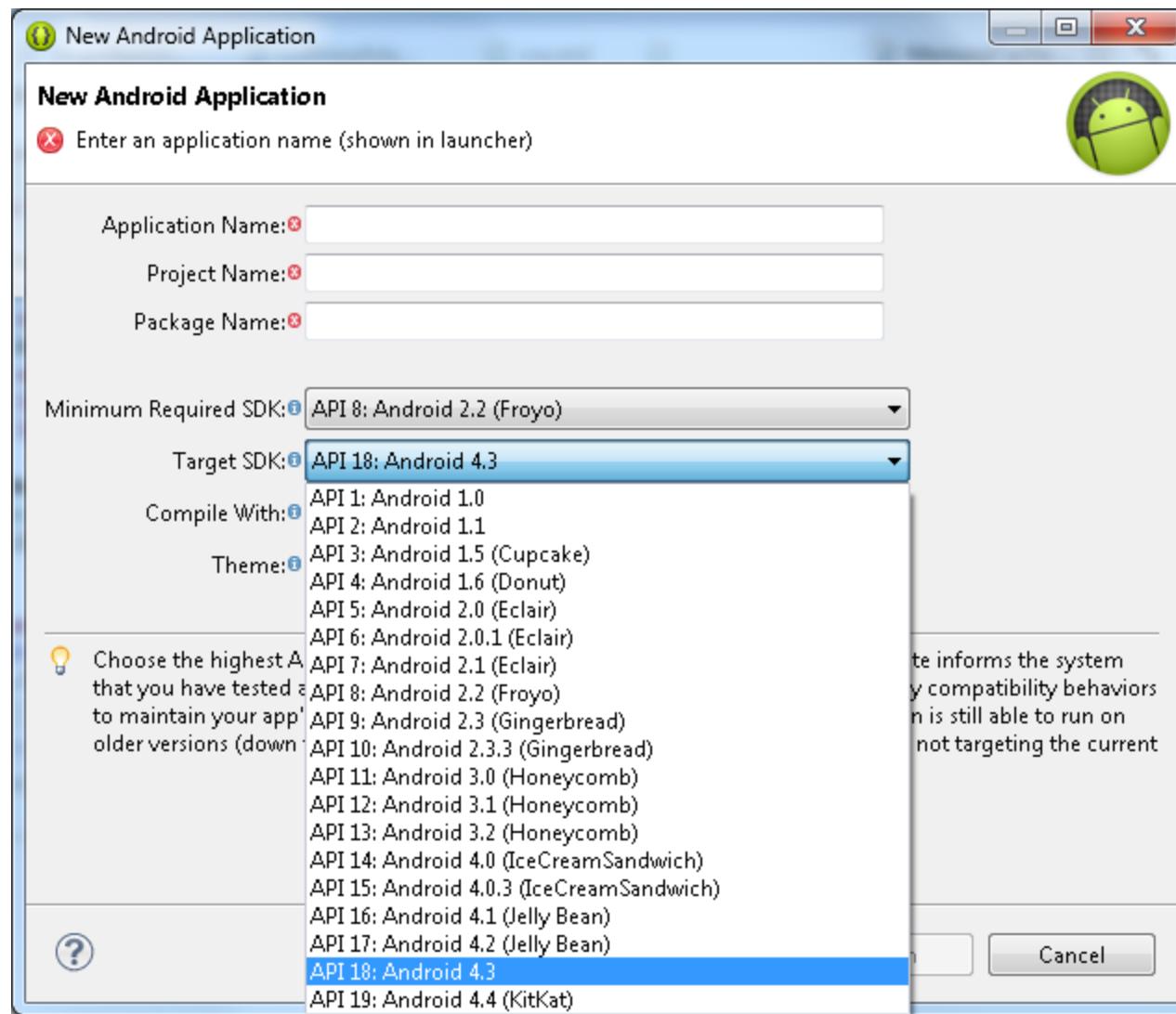


Data collected during a 14-day period ending on February 4, 2013

Last historical dataset collected during a 14-day period ending on February 1, 2013



If you develop your application for the version that is at the very top of the chart, then your application is compatible with 100% of active devices (and all future versions), because all Android APIs are forward compatible. Or, if you develop your application for a version lower on the chart, then it is currently compatible with the percentage of devices indicated on the y-axis, where the line for that version meets the y-axis on the right.



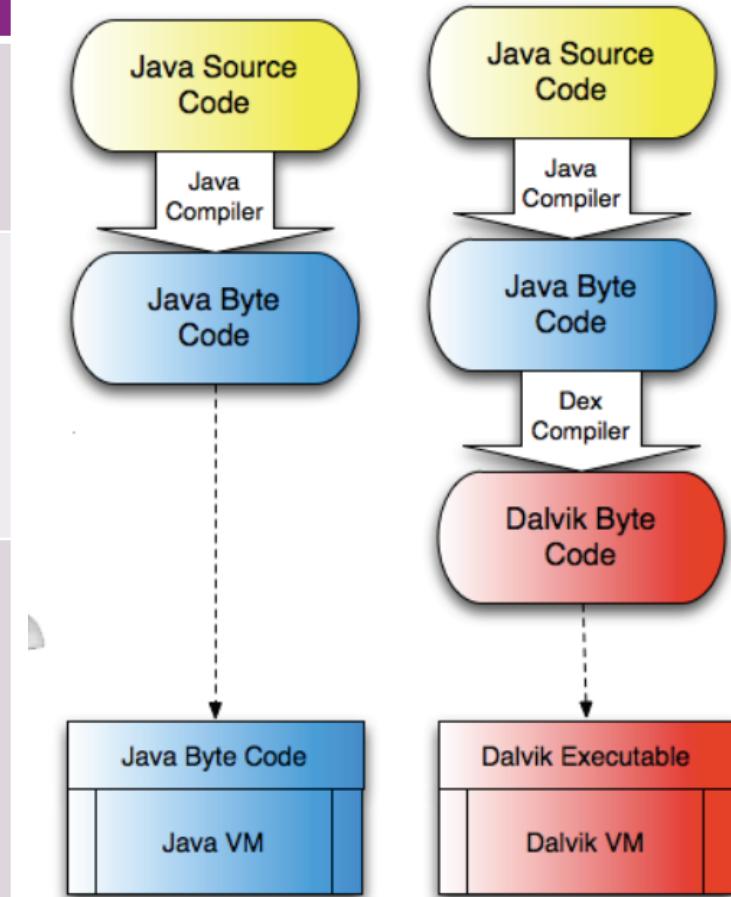
# Android Architecture



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

# Java SDK vs. Android SDK

Java SDK	Android SDK
Lot of CPU Speed – Unlimited Memory Allocations	Low Speed CPU & Very little Memory
Write Code, Compile, Run & Debug on Same Machine	Code is written and Compile on PC. Emulator or Actual Device to Run the App.
Need not worry about task priorities OS will take care	Phone functionality is priority #1 on a call between when there is a call, one application would be closed or paused by the device.
Key board and mouse, large screen size,	



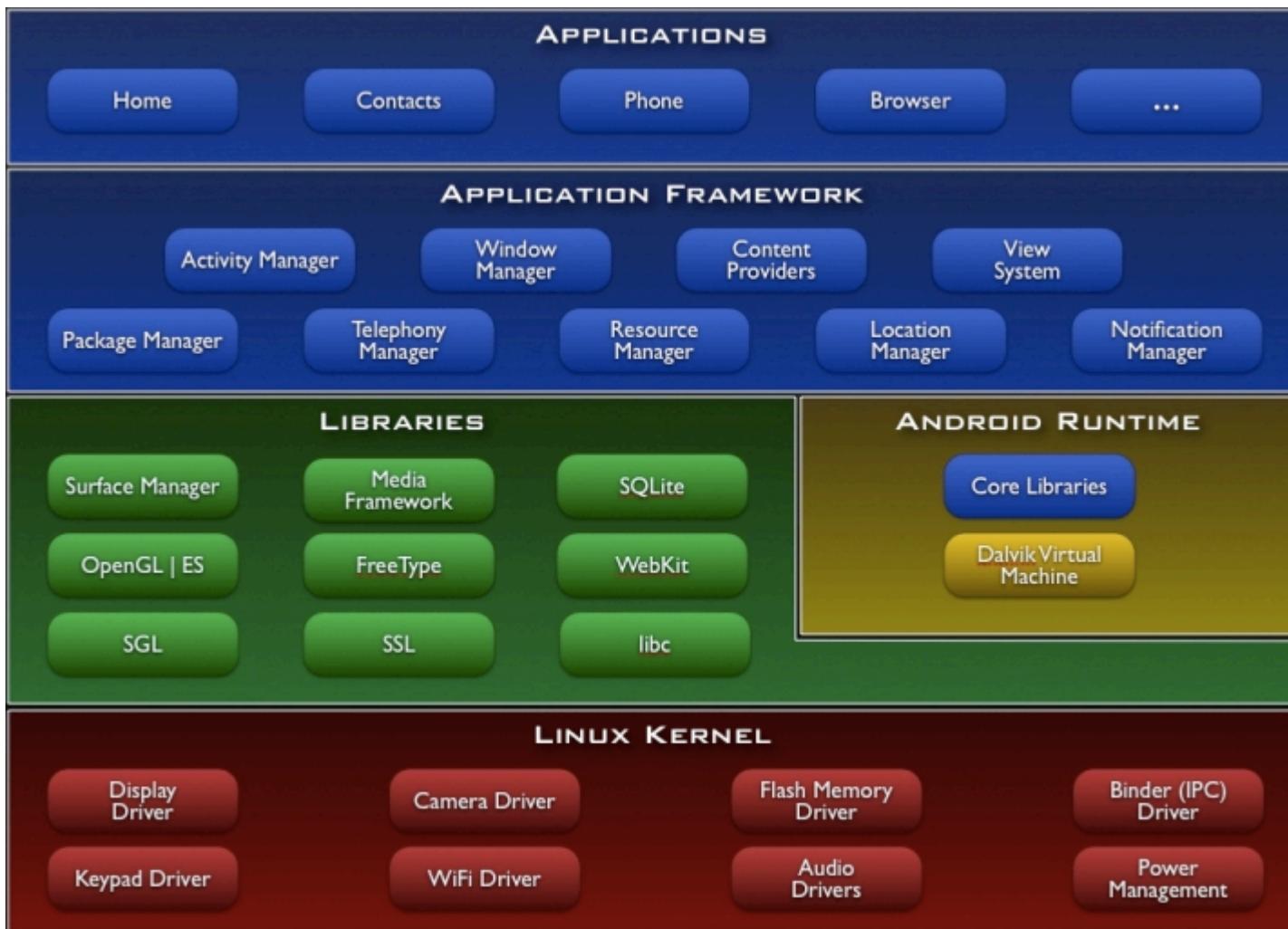
# Introduction to Android Features

Android is made up of several necessary and dependent parts, including the following:

- **A hardware reference design** that describes the capabilities required for a mobile device to support the software stack.
- **A Linux operating system kernel** that provides low-level interface with the hardware, memory management, and process control, all optimized for mobile devices.
- **Open-source libraries** for application development, including SQLite, WebKit, OpenGL, and a media manager.
- **A run time used to execute and host Android applications**, including the Dalvik virtual machine and the core libraries that provide Android-specific functionality. The run time is designed to be small and efficient for use on mobile devices.
- **An application framework** that agnostically exposes system services to the application layer, including the window manager and location manager, content providers, telephony, and sensors.
- **A user interface framework** used to host and launch applications.
- **Preinstalled applications** shipped as part of the stack.
- **A software development kit** used to create applications, including tools, plug-ins, and documentation.



# Architecture

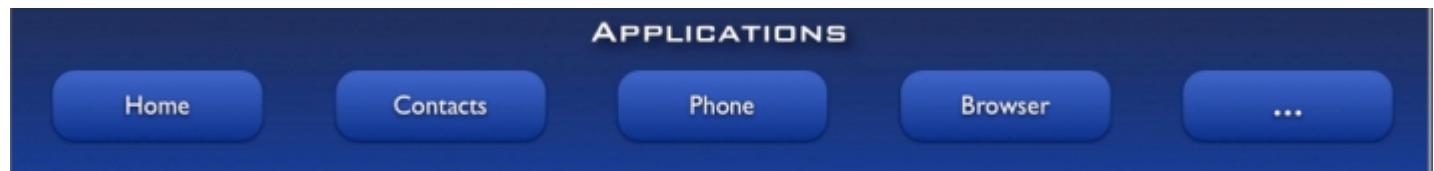


Source: <http://code.google.com/p/androidteam/wiki/AndroidSystemArch>

# Core Android Applications

Android will ship with a set of core applications:

- Client
- Email Client
- Phone
- SMS Application
- Maps
- Browser
- Other applications



# Application Framework

Expose various capabilities of the Android OS to application developers:

Views: A rich and extensible set of views that can be used to build an applications including lists, grids, text boxes, buttons, and even an embedded browser.

java.lang.Object  
↳ android.view.View

This class represents the basic building block for user interface components. A View occupies a rectangular area on the screen and is responsible for drawing and event handling. View is the base class for widgets, which are used to create interactive UI components (buttons, text fields, etc.). The ViewGroup subclass is the base class for layouts, which are invisible containers that hold other Views (or other ViewGroups) and define their layout properties.



# Application Framework

Content Providers that enable applications to access data from other applications (such as Contacts), or to share their own data

`java.lang.Object`

↳ `android.content.ContentProvider`

Content providers are one of the primary building blocks of Android applications, providing content to applications. They encapsulate data and provide it to applications through the single ContentResolver interface. A content provider is only required if you need to share data between multiple applications. For example, the contacts data is used by multiple applications and must be stored in a content provider. If you don't need to share data amongst multiple applications you can use a database directly via SQLiteDatabase.



# Content Providers

Android ships with Content Providers:

1. Browser – Stores data such as browser bookmarks, browser history, and so on
2. CallLog – Stores data such as missed calls, call details
3. Contacts – Stores Contact details
4. Media Storage – Stores Media files such as video, audio
5. Settings – Stores the device's settings and preferences

Content://media/internal/images



# Application Framework

A Resource Manager, providing access to non-code resources such as localized strings, graphics, and layout files.

1. Internationalization & Unicode support
2. High Quality 3-D Graphics

[http://news.cnet.com/8301-1023\\_3-57391152-93/android-apps-max-size-shoots-from-50mb-to-4gb/](http://news.cnet.com/8301-1023_3-57391152-93/android-apps-max-size-shoots-from-50mb-to-4gb/)



# Application Framework

A Notification Manager that enables all applications to display custom alerts in status bar.

`java.lang.Object`

↳ `android.app.NotificationManager`

Class to notify the user of events that happen. This is how you tell the user that something has happened in the background.



# Application Framework

An Activity Manager that manages the lifecycle of applications and provides common navigation back stack

Class	Description
<b>ActivityManager.MemoryInfo</b>	Information you can retrieve about the available memory through <a href="#"><u>getMemoryInfo(ActivityManager.MemoryInfo)</u></a> .
<b>ActivityManager.ProcessErrorStatelnfo</b>	Information you can retrieve about any processes that are in an error condition.
<b>ActivityManager.RecentTaskInfo</b>	Information you can retrieve about tasks that the user has most recently started or visited.
<b>ActivityManager.RunningAppProcessInfo</b>	Information you can retrieve about a running process.
<b>ActivityManager.RunningServiceInfo</b>	Information you can retrieve about a particular Service that is currently running in the system.
<b>ActivityManager.RunningTaskInfo</b>	Information you can retrieve about a particular task that is currently "running" in the system.



# Libraries

Android includes a set of C/C++ libraries used by various components of the Android system. These capabilities are exposed to developers through the Android application framework. Some of the core libraries are listed below:

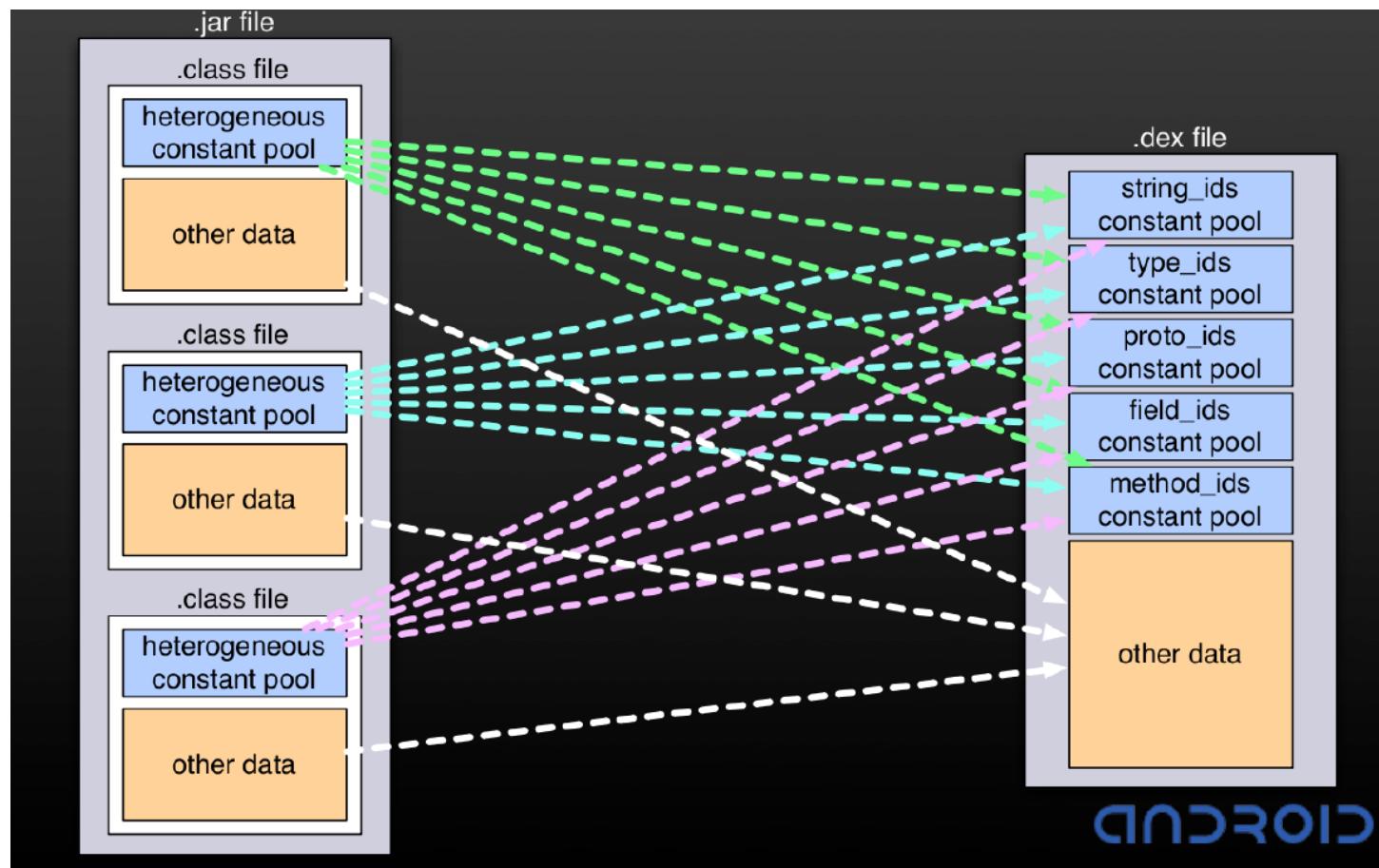
**System C library** - a BSD-derived implementation of the standard C system library (libc), tuned for embedded Linux-based devices

1. **BSD license**
2. **Small Size and Fast Code Paths**
3. **Very fast Custom pThread Implementation**
4. **Built-in support for Android specific services**
5. **System Properties, Log Capabilities)**
6. **Doesn't Support some POSIX features**



# Dex File Anatomy

In order to save storage space, .dex files only contain unique data. If multiple class files share the same string, this string would only exist once in the .dex file and the multiple occurrences are just pointers to this one string. The same mechanism is used for method names, constants and objects which results in smaller files with much internal “pointing”.



Source: Analysis of the Android Architecture by Studienarbeit Von

# DVM in Android Operating System

Android operating system is based on a Linux kernel with native open---source C libraries. The Dalvik Virtual Machine (DVM) that Android uses is designed to manage multiple instances on a single device. The DVM uses the kernel's process and memory management functionalities. The DVM executes Dalvik executable files (.dex), which help efficient memory usage.

The .DEX file has **constant pool** that stores all literal constant values like string constants, attributes, methods, classes, interfaces, etc. used in the class. Unlike the .class file, these values stored in the common pool are referred by their index in the constant pool and not specifying thought all the classes, thus **eliminating the duplicate** of constants. This helps in conserving the memory. The constant pool in a .DEX file is a Strings constant pool, Type/Class constant pool, Filed constant pool, and method constant pool.

Source: Analysis of the Android Architecture by Studienarbeit Von

# Libraries

SQLite supports database management support

WebKit for Web Browsing purposes

SGL: underlying 2D graphics engine

FreeType: Bitmap and vector font rendering



# Android Runtime

Provides core runtime libraries that enables the developers to write Apps using Java Programming Language.

Dalvik virtual machine enables every Android application to run in its own process.

1. Optimized for low memory requirements
2. Designed to allow multiple VM instances to run at once
3. Relying on Underlying OS for process isolation, memory management and threading support
4. Execute Dalvik Executable (DEX) files which are zipped into an APK file



# Android Runtime

Each subsequent application is launched in it's own process



# Linux Kernel

This is the kernel on which Android is based. This layer contains all the low-level device drivers for the various hardware components of an Android device.

1. Android is built on the Linux kernel, but Android is not Linux
2. No native windowing system
3. No glibc support
4. Does not include the full set of standard Linux utilities
5. Standard Linux 2.6.24 Kernel (L uses 3.14)
6. Patch of “kernel enhancements” to support Android

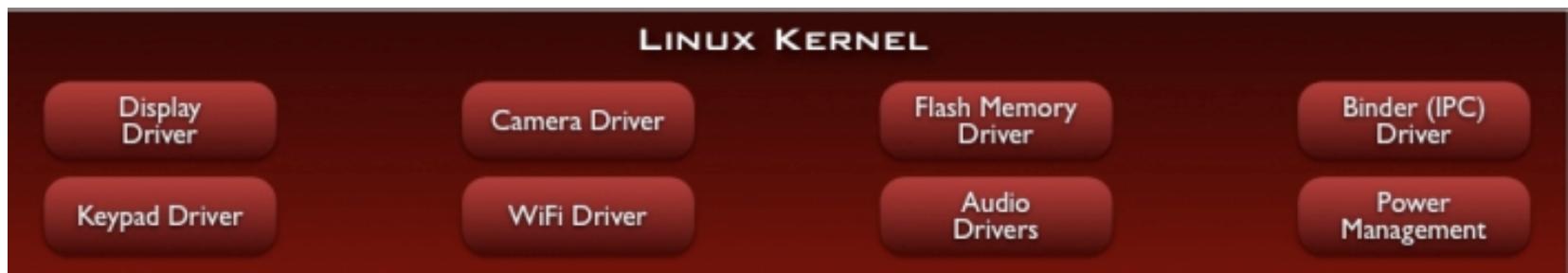


# Why Linux Kernel?

1. Great memory and process management
2. Permissions-based security model
3. Proven driver model
4. Support for shared libraries
5. It's already open source!

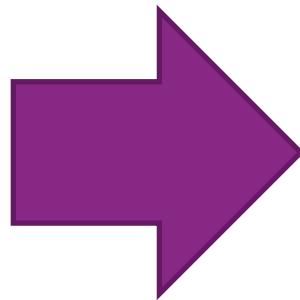
Read:

[Runtime Walkthrough - Android Anatomy and Physiology](#)

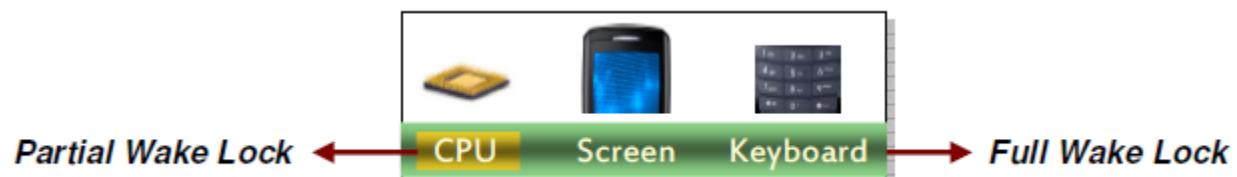


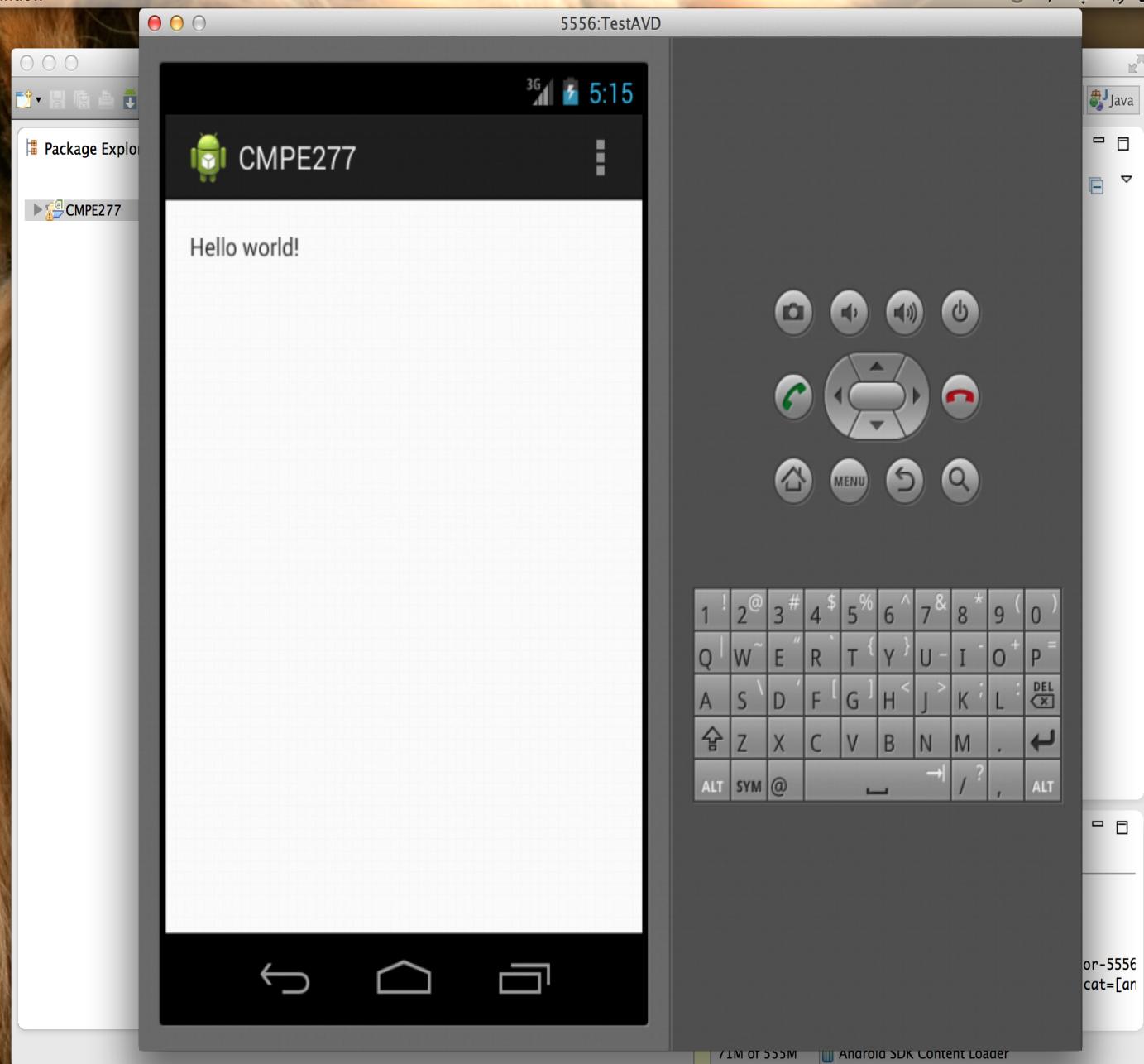
# Linux Drivers

1. Display Drivers –
2. Keypad Driver
3. Camera Driver
4. Wi Fi Driver
5. Flash Memory Driver
6. Audio Driver
7. Power Management:



Hardware Abstract  
Layer (HAL)





HelloWorld

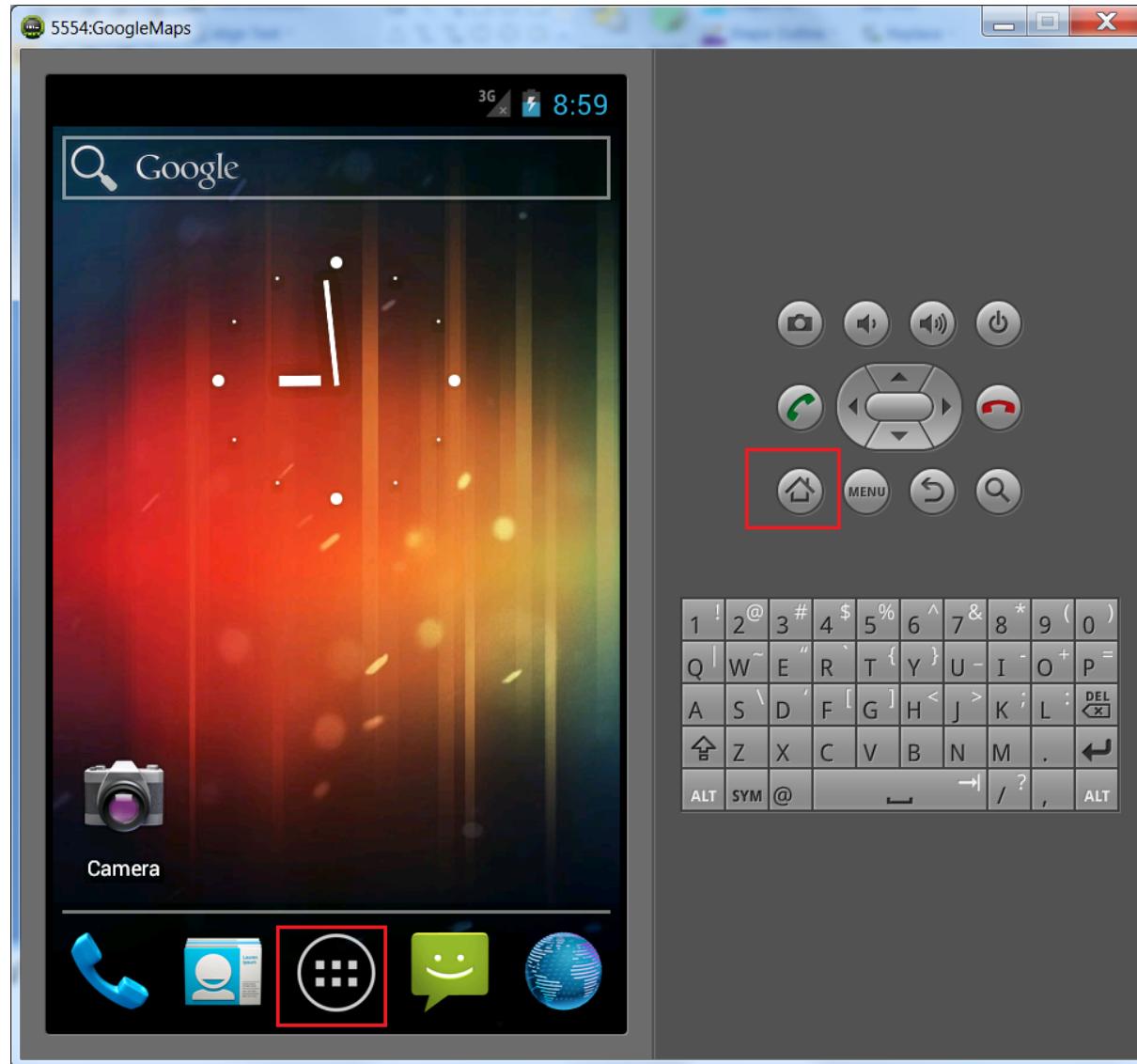


Wizard101

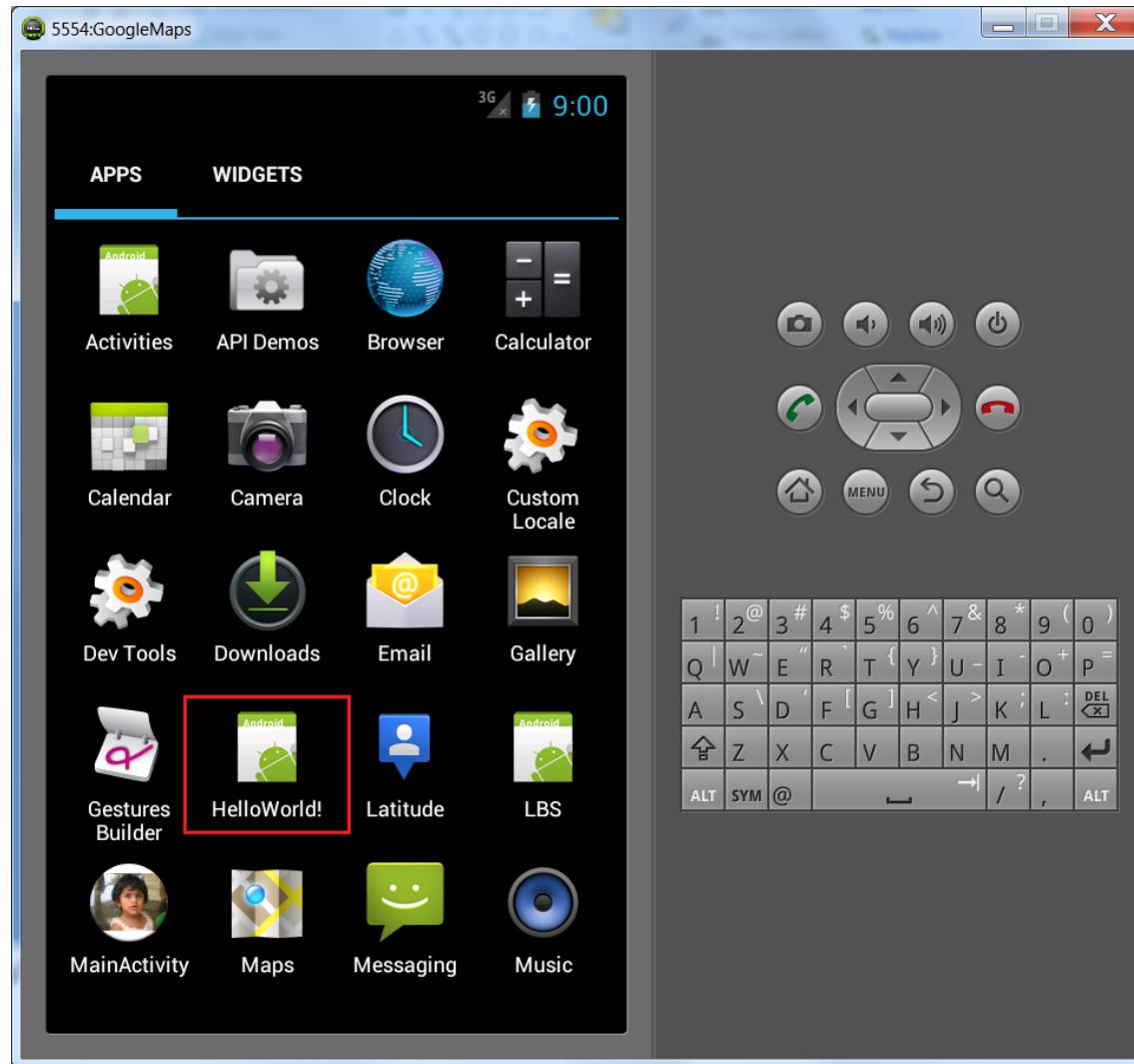


Applications

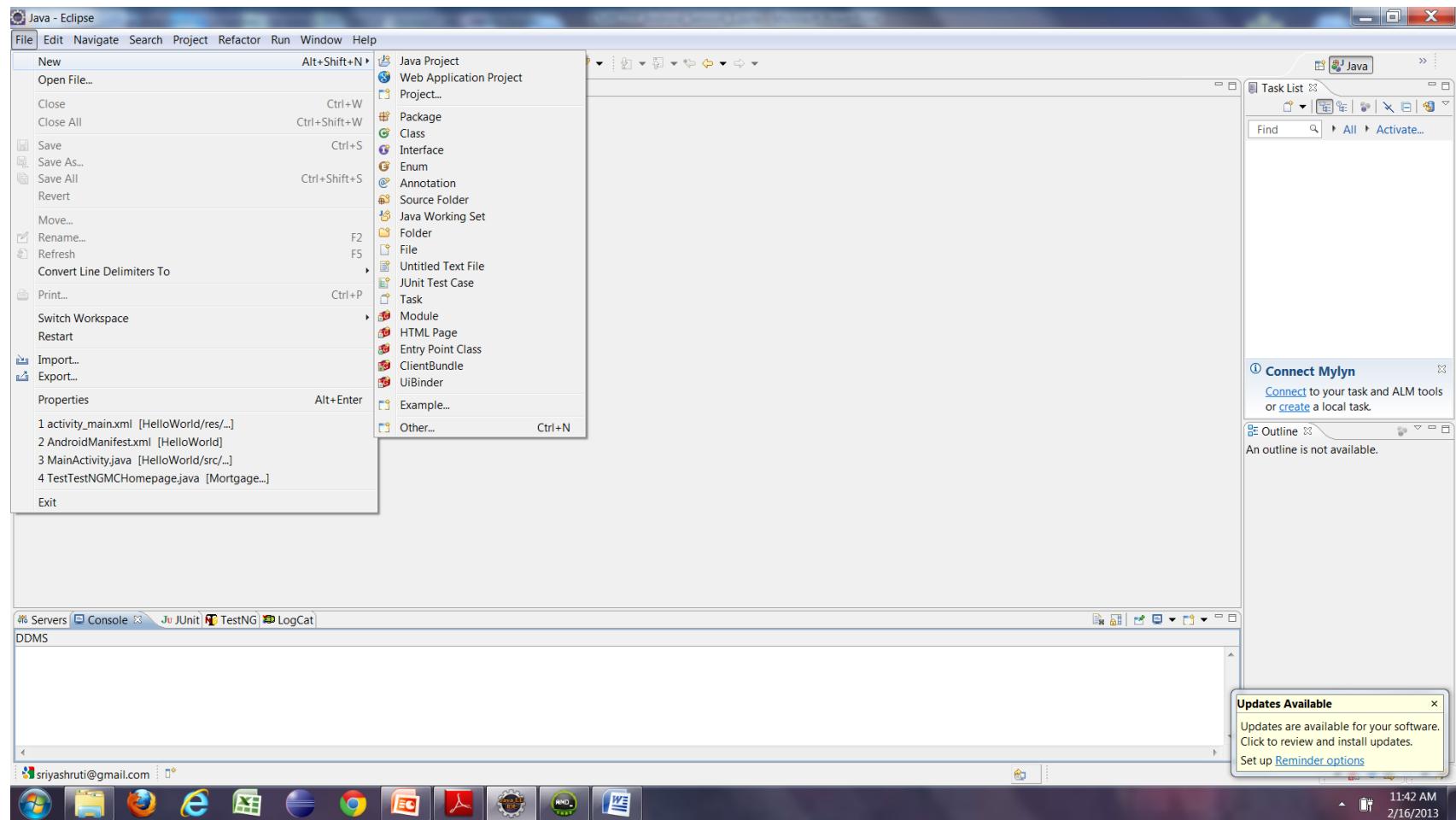
# Android Home Screen



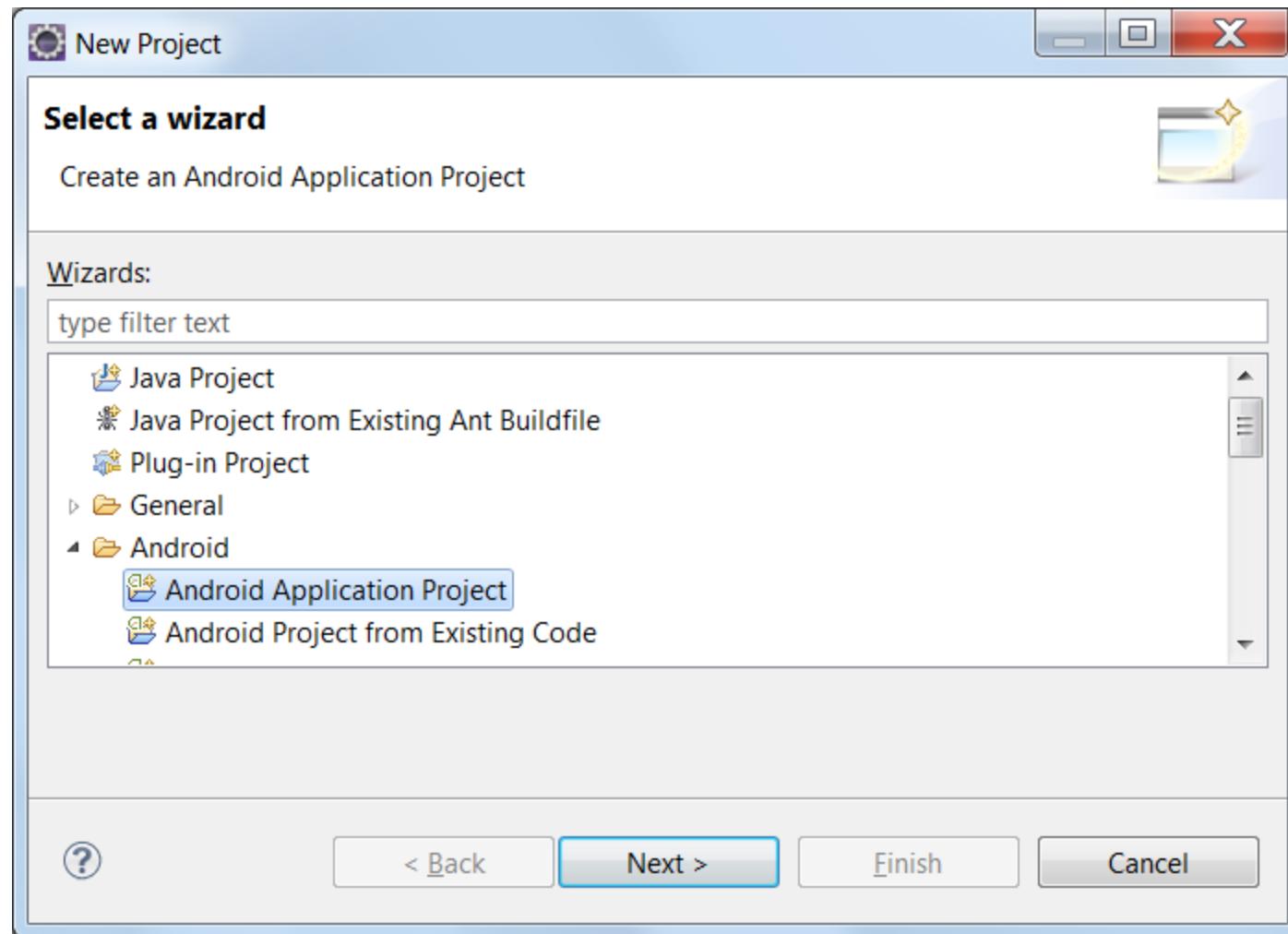
# Application Launcher



# Creating First Android Application



# Creating Your First Android Application



Quick Access

Java

Package Explorer

CMPE277

## Android Virtual Device (AVD) Manager

Android Virtual Devices Device Definitions

List of existing Android Virtual Devices located at /Users/chandrasekarvuppala/.android/avd

AVD Name	Target Name	Platform	API Level	CPU/ABI	
TestAVD	Android 4.3	4.3	18	ARM (armeabi-v7a)	Create...

Create new Android Virtual Device (AVD)

AVD Name:

Nexus 7 (2012) (7.0", 800 × 1280: tvdpi)

Nexus 5 (4.95", 1080 × 1920: xxhdpi)

Nexus 4 (4.7", 768 × 1280: xhdpi)

Nexus 10 (10.055", 2560 × 1600: xhdpi)

Nexus 7 (7.02", 1200 × 1920: xhdpi)

Galaxy Nexus (4.65", 720 × 1280: xhdpi)

Nexus S (4.0", 480 × 800: hdpi)

Nexus One (3.7", 480 × 800: hdpi)

Android TV (720p) (1280 × 720: tvdpi)

Android TV (1080p) (1920 × 1080: xhdpi)

Android TV (720p) (1280 × 720: tvdpi)

Android TV (1080p) (1920 × 1080: xhdpi)

Android Wear Round (320 × 320: hdpi)

Android Wear Square (280 × 280: hdpi)

Android Wear Round (320 × 320: hdpi)

Android Wear Square (280 × 280: hdpi)

Android TV (720p) (1280 × 720: tvdpi)

Android TV (1080p) (1920 × 1080: xhdpi)

Android Wear Round (320 × 320: hdpi)

Android Wear Square (280 × 280: hdpi)

10.1" WXGA (Tablet) (1280 × 800: mdpi)

7" WSVGA (Tablet) (1024 × 600: mdpi)

5.4" FWVGA (480 × 854: mdpi)

5.1" WVGA (480 × 800: mdpi)

4.7" WXGA (1280 × 720: xhdpi)

4.65" 720p (Galaxy Nexus) (720 × 1280: xhdpi)

4" WVGA (Nexus S) (480 × 800: hdpi)

3.7" FWVGA slider (480 × 854: hdpi)

3.7" WVGA (Nexus One) (480 × 800: hdpi)

3.4" WQVGA (240 × 432: ldpi)

3.3" WQVGA (240 × 400: ldpi)

3.2" QVGA (ADP2) (320 × 480: mdpi)

Device:

Target:

CPU/ABI:

Keyboard:

Skin:

Front Camera:

Back Camera:

Memory Options:

Internal Storage:

SD Card:

Emulation Options:

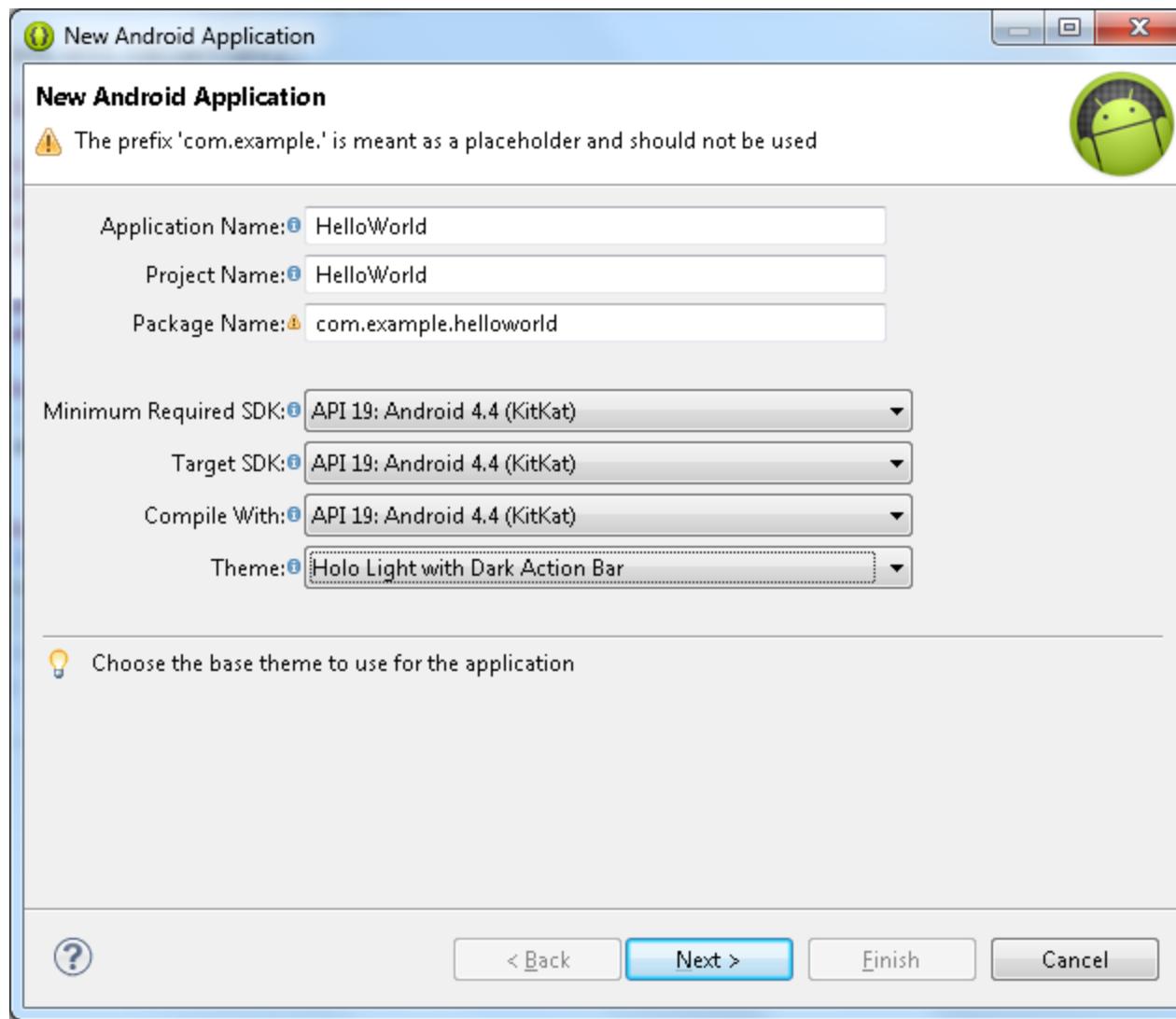
 Override the ex... AVD Name can...

TV,  
Wearables

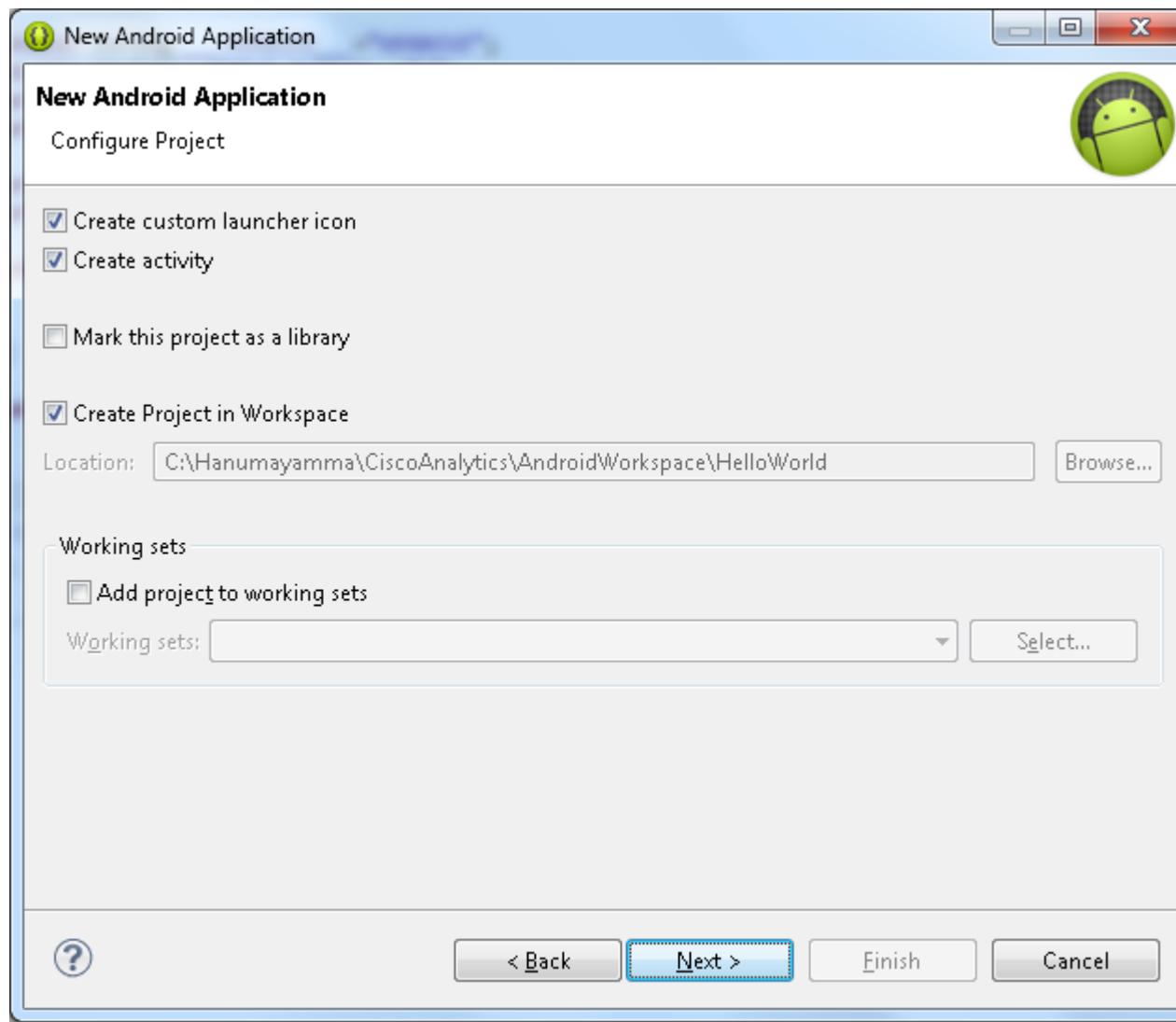
5556

=[ar

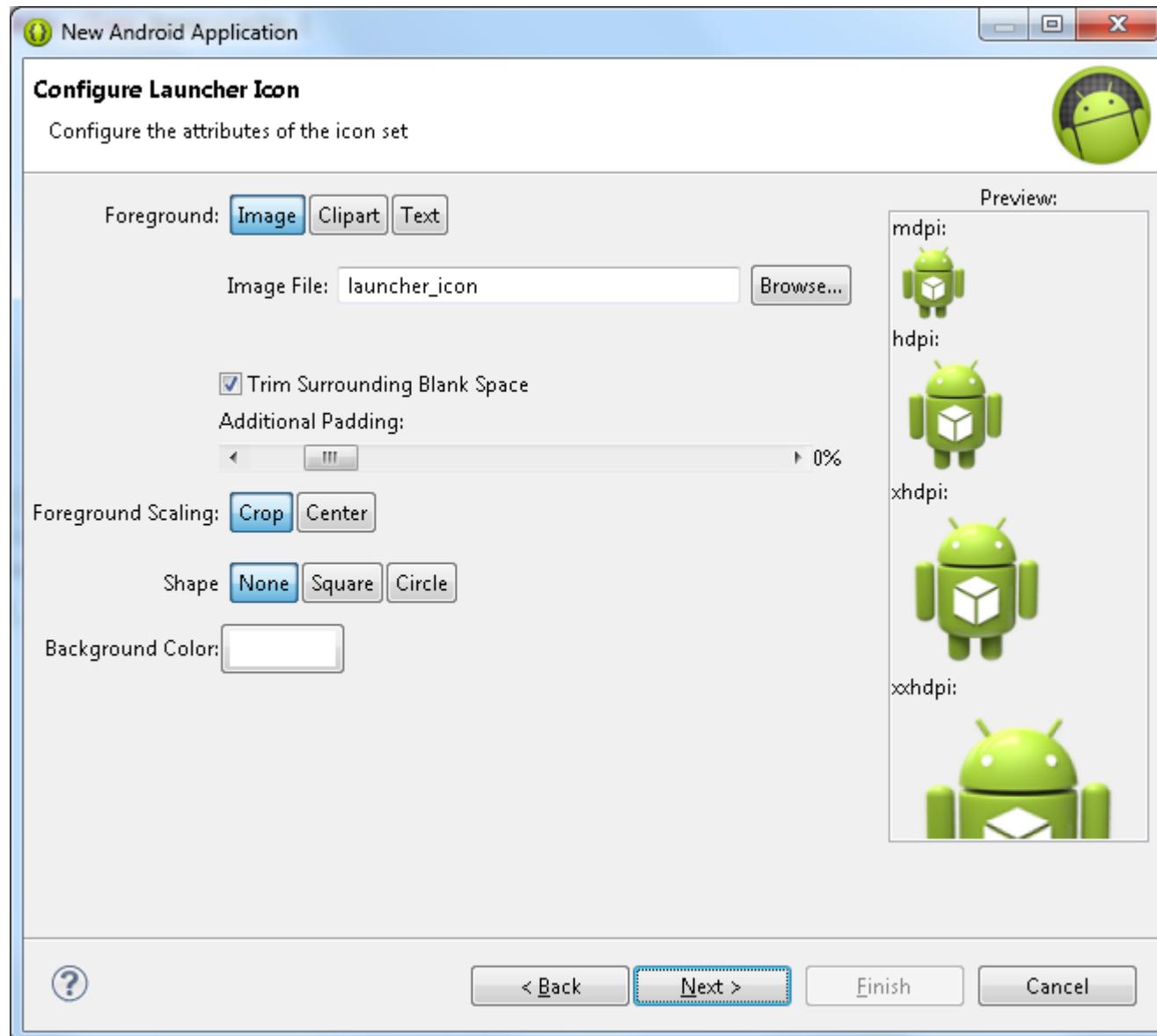
# Creating Your First Android Application



# Creating Your First Android Application



# Creating Your First Android Application



# Creating Your First Android Application

New Android Application

Create Activity

Select whether to create an activity, and if so, what kind of activity.

Create Activity

**Blank Activity** Selected

Fullscreen Activity

Master/Detail Flow

New Android Application

**Blank Activity**

Creates a new blank activity, with an action bar and optional navigational elements such as tabs or horizontal swipe.

Activity Name:

Layout Name:

Navigation Type:

None

Fixed Tabs + Swipe

Scrollable Tabs + Swipe

Dropdown

The type of navigation to use for the activity

?

?

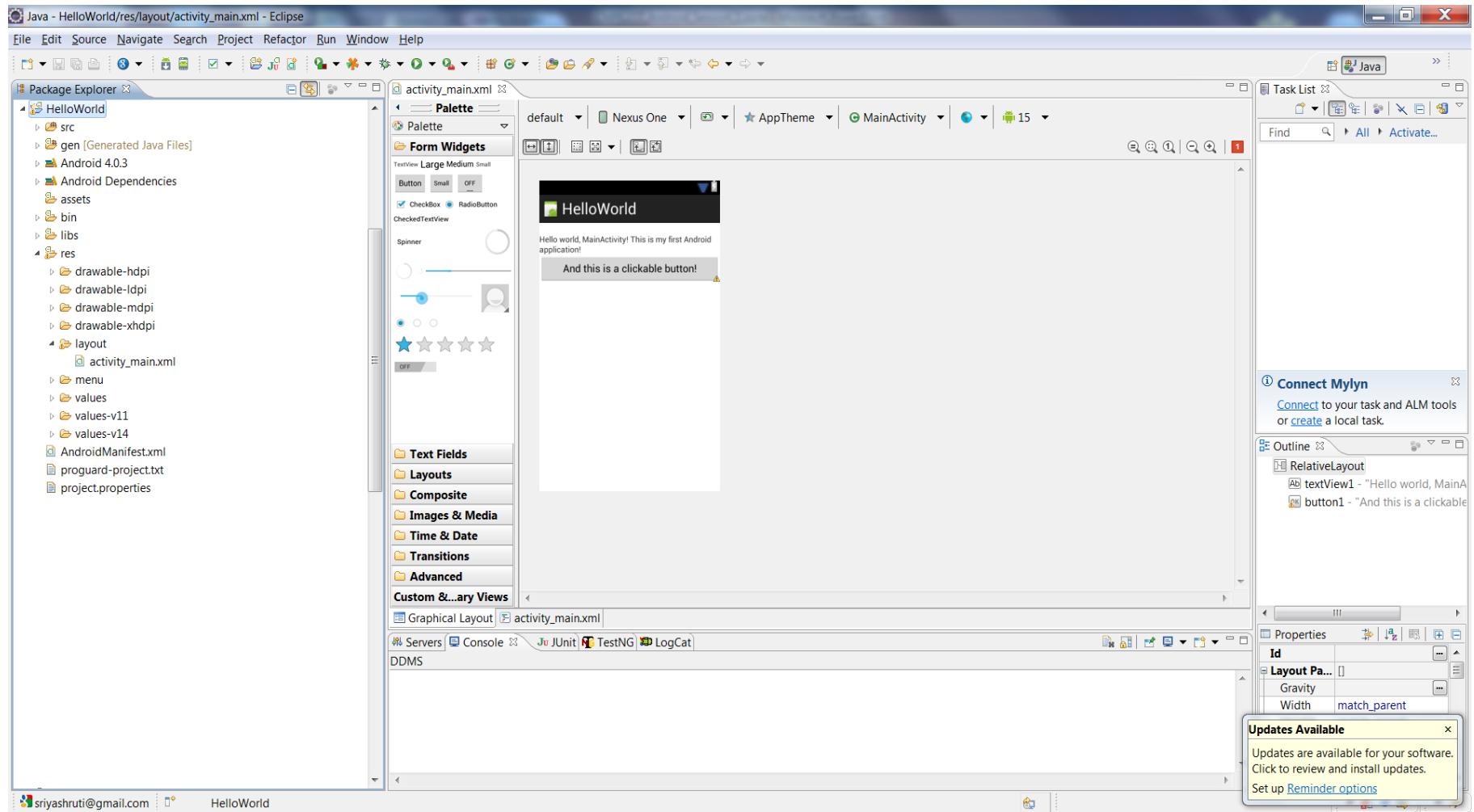
< Back

Next >

Finish

Cancel

# Creating Your First Android Application



# Creating Your First Android Application

Android Virtual Device Manager

Android Virtual Devices | Device Definitions

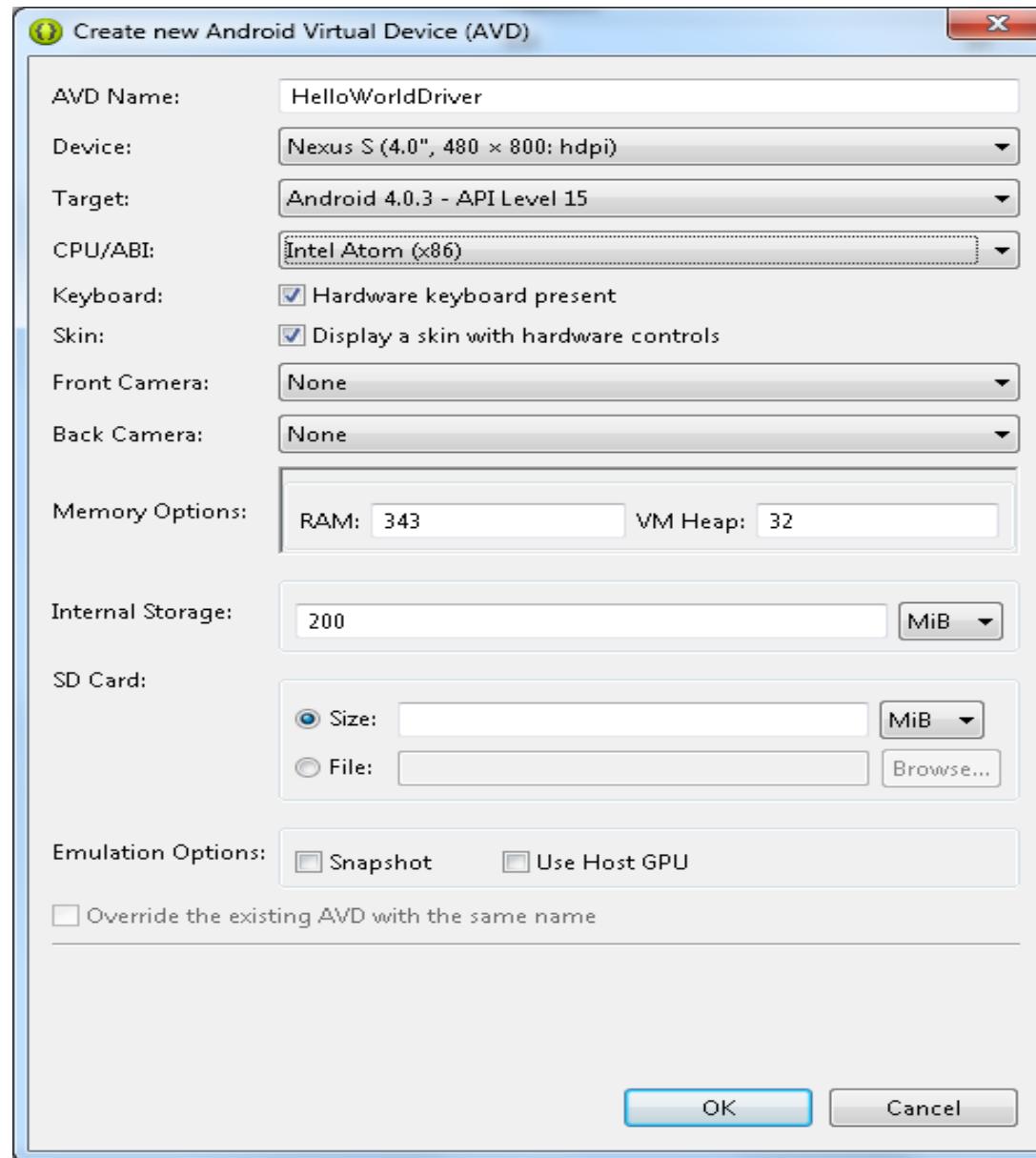
List of existing Android Virtual Devices located at C:\Users\cvuppala\.android\avd

AVD Name	Target Name	Platform	API Level	CPU/ABI
✓ DevMac	Android 4.4	4.4	19	ARM (armeabi-v7a)
✓ QB	Android 4.4	4.4	19	ARM (armeabi-v7a)

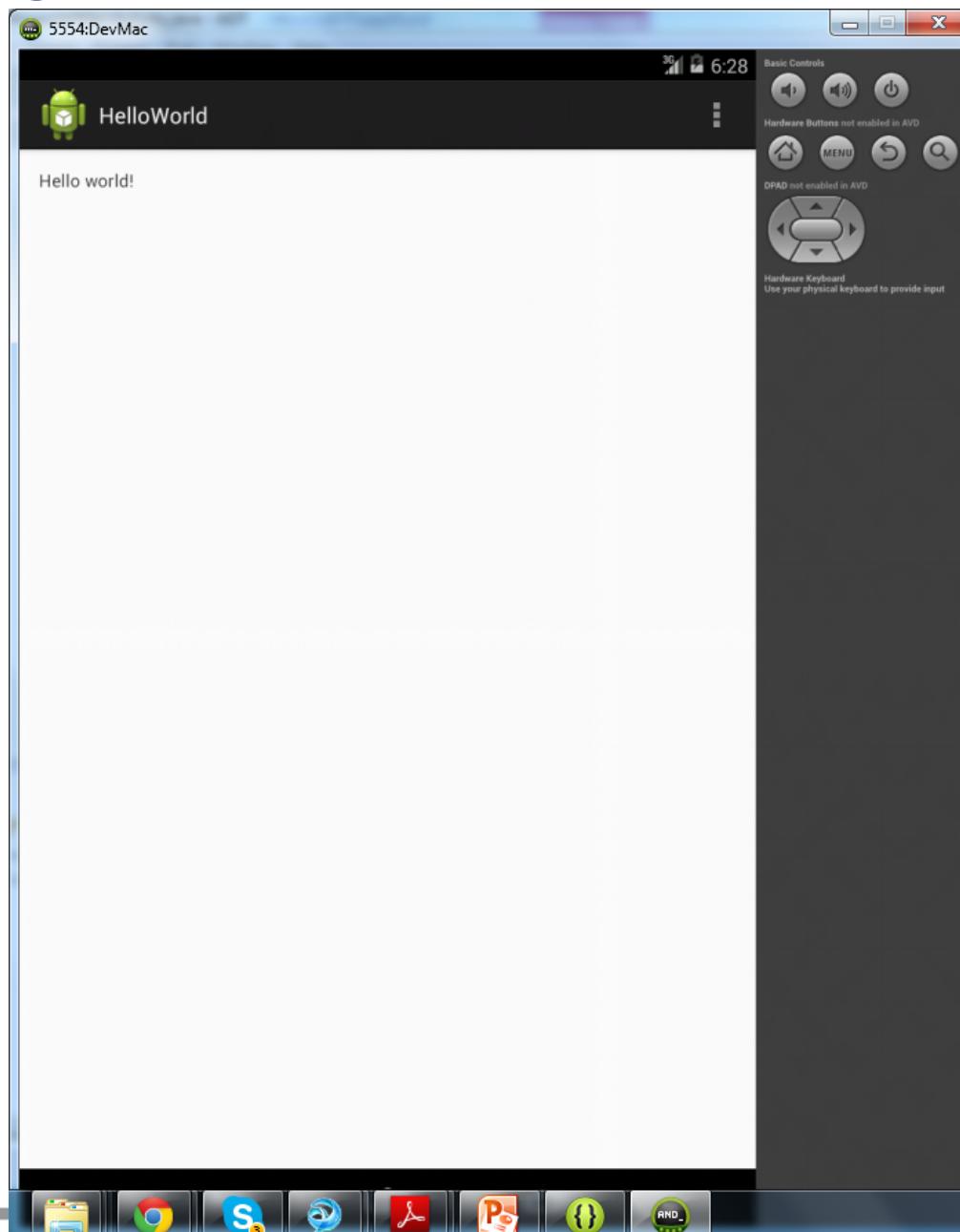
New... Edit... Delete... Repair... Details... Start... Refresh

✓ A valid Android Virtual Device. ⚡ A repairable Android Virtual Device.  
✗ An Android Virtual Device that failed to load. Click 'Details' to see the error.

# Creating Your First Android Application



# Creating Your First Android Application

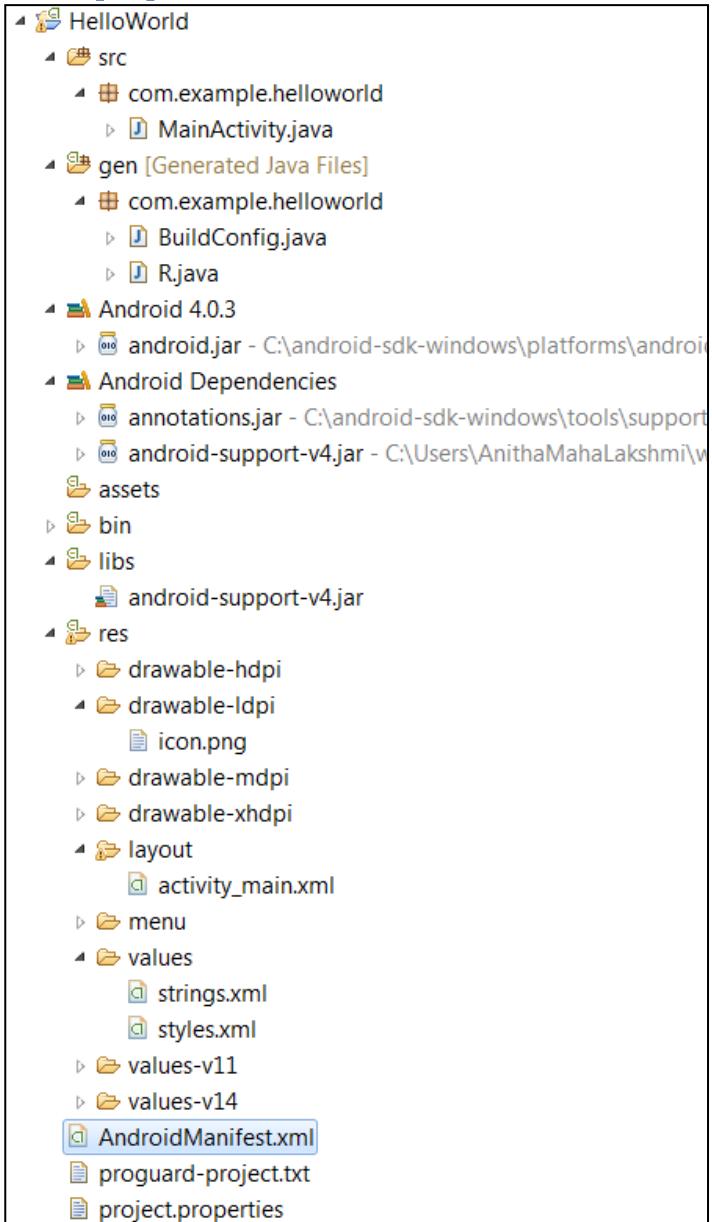


# Basic Concepts in Android Application

- Activity - A screen in the Android application
- Services- Background activities without UI
- Content Provider - provides data to applications, Android contains a SQLite DB which can serve as data provider
- Broadcast Receivers— receives system messages, can be used to react to changed conditions in the system
- Intents— allow the application to request and/or provide services . For example the application call ask via an intent for a contact application. Application register itself via an Intent Filter. Intends are a powerful concept as they allow to create loosely coupled applications.

# Anatomy of An Android Application

- src – contains the .Java source files for your project.
- gen - Contains the R.java file, a compiler-generated file that references all the resources found in your project.  
**You should not modify this file.**
- Android 4.0.3 - This item contains one file, android.jar, which contains all the class libraries needed for an Android application.
- Android Dependencies – Contains Android dependencies files.
- assets — This folder contains all the assets used by your application, such as HTML, text files, databases, etc.
- res — This folder contains all the resources used in your application. It also contains a few other subfolders: `drawable-<resolution>`, `layout`, and `values`.
- AndroidManifest.xml — This is the manifest file for your Android application. Here you specify the permissions needed by your application, as well as other features (such as intent-filters, receivers, etc.)



# Activity Main.XML

## Activity\_main.xml

The Activity\_main.xml file defines the user interface (UI) of your application.

**Activity – A screen in the Android Application**

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
    xmlns:tools="http://schemas.android.com/tools"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent" >  
  
    <TextView  
        android:id="@+id/textView1"  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
        android:layout_alignParentLeft="true"  
        android:layout_alignParentRight="true"  
        android:layout_alignParentTop="true"  
        android:layout_marginTop="20dp"  
        android:text="@string/hello_world"  
        tools:context=".MainActivity" />  
  
    <Button  
        android:id="@+id/button1"  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
        android:layout_alignParentLeft="true"  
        android:layout_alignParentRight="true"  
        android:layout_below="@+id/textView1"  
        android:text="And this is a clickable button!" />  
  
    </RelativeLayout>
```

# Android Manifest.XML

Within the definition for this activity, there is an element named <intent-filter>:

- The action for the intent filter is named android.intent.action.MAIN to indicate that this activity serves as the entry point for the application.
- The category for the intent-filter is named android.intent.category.LAUNCHER to indicate that the application can be launched from the device's Launcher icon.

## AndroidManifest.xml

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
    package="com.example.helloworld"  
    android:versionCode="1"  
    android:versionName="1.0" >  
  
<uses-sdk  
    android:minSdkVersion="11"  
    android:targetSdkVersion="15" />  
  
<application  
    android:icon="@drawable/icon"  
    android:label="@string/app_name"  
    android:theme="@style/AppTheme" >  
    <activity  
        android:name=".MainActivity"  
        android:label="@string/title_activity_main" >  
        <intent-filter>  
            <action  
                android:name="android.intent.action.MAIN" />  
            <category  
                android:name="android.intent.category.LAUNCHER" />  
        </intent-filter>  
    </activity>  
    </application>  
  
</manifest>
```

# Thank you!

Note: the slides are based on the original work of Prof. Chandra Vuppalapati

# References

- Introductions to Processing on Android Devices,  
[http://dl.acm.org.libaccess.sjlibrary.org/citation.cfm?  
id=1900520.1900545&coll=DL&dl=ACM&CFID=180845267&CFTOKEN=15732211](http://dl.acm.org.libaccess.sjlibrary.org/citation.cfm?id=1900520.1900545&coll=DL&dl=ACM&CFID=180845267&CFTOKEN=15732211)
- Android Anatomy And Physiology – androidteam - [https://docs.google.com/viewer?  
a=v&q=cache:iDzu7bHR1dMJ:androidteam.googlecode.com/files/Anatomy-  
Physiology-of-an-  
Android.pdf+android+anatomy&hl=en&gl=us&pid=bl&srcid=ADGEESh0P\\_UCHSGz  
nPMNmQVveQ834EMhfWG1wiPNLqNoVHvPMLiZituwcLAz64qkHZw5s2JbLrr1NV  
jWAXhvs\\_FA8tTe6fK0eeLDXm93rblryYhXqd3eJKZaXkROI-  
iiul415WrWYLU&sig=AHIEtbSJI\\_7uKIT5ugxW7\\_SuGpo7fnI7mQ](https://docs.google.com/viewer?a=v&q=cache:iDzu7bHR1dMJ:androidteam.googlecode.com/files/Anatomy-Physiology-of-an-Android.pdf+android+anatomy&hl=en&gl=us&pid=bl&srcid=ADGEESh0P_UCHSGznPMNmQVveQ834EMhfWG1wiPNLqNoVHvPMLiZituwcLAz64qkHZw5s2JbLrr1NVjWAXhvs_FA8tTe6fK0eeLDXm93rblryYhXqd3eJKZaXkROI-iiul415WrWYLU&sig=AHIEtbSJI_7uKIT5ugxW7_SuGpo7fnI7mQ)
- Beginning Android Application Development, Wei-Meng Lee, 2011 Wiley Publishing
- Android Development Team – <http://www.android.com/about/>
- To learn how to design notifications and their interactions, read the Android Design Guide - <http://developer.android.com/design/patterns/notifications.html>
- the Status Bar Notifications developer guide -[http://developer.android.com/guide/  
topics/ui/notifiers/notifications.html](http://developer.android.com/guide/topics/ui/notifiers/notifications.html)
- Learning About Android Graphics Subsystem -  
<http://developer.mips.com/2012/04/11/learning-about-android-graphics-subsystem/>
- Android Developer Guide -  
<http://developer.android.com/guide/components/index.html>