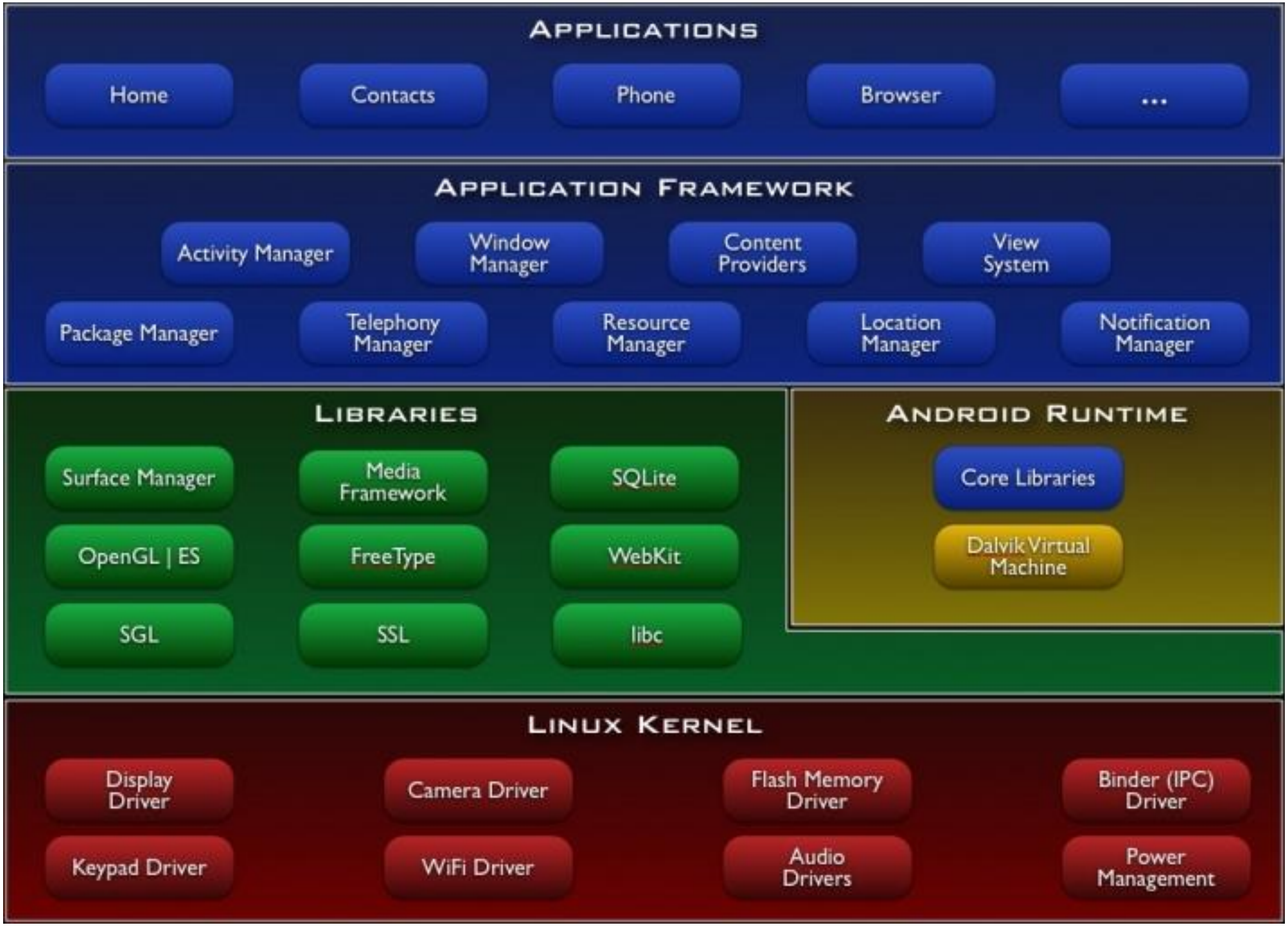
Mobile Operating System Architectures

Android Overview

What is Android?

- A software stack for mobile devices that includes
  - An operating system
  - Middleware
  - Key Applications
- Uses Linux to provide core system services
  - Security
  - Memory management
  - Process management
  - Power management
  - Hardware drivers



http://developer.android.com/guide/basics/what-is-android.html

Android Features

- **Application framework** enabling reuse and replacement of components
- **Integrated browser** based on the open source WebKit engine
- **Optimized graphics** powered by a custom 2D graphics library; 3D graphics based on the OpenGL ES 1.0 specification (hardware acceleration optional)
- **SQLite** for structured data storage
- **Media support** for common audio, video, and still image formats (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF)
- **GSM Telephony** (hardware dependent)
- **Bluetooth, EDGE, 3G, and WiFi** (hardware dependent)
- **Camera, GPS, compass, and accelerometer** (hardware dependent)
- **Rich development environment** including a device emulator, tools for debugging, memory and performance profiling, and a plugin for the Eclipse IDE

http://developer.android.com/guide/basics/what-is-android.html

A Short History Of Android

- 2001 Palm Kyocera 6035, combing PDA and phone
  - PDA = personal data assistant, PalmPilot
- 2003 - Blackberry smartphone released
- 2005
  - Google acquires startup Android Inc. to start Android platform.
  - Work on Dalvik VM begins
- 2007
  - Open Handset Alliance announced
  - Early look at SDK
  - June, iPhone released
- 2008
  - Google sponsors 1st Android Developer Challenge
  - T-Mobile G1 announced, released fall
  - SDK 1.0 released
  - Android released open source (Apache License)
  - Android Dev Phone 1 released

Pro Android by Hashimi & Komatineni (2009)

Short History cont.

- 2009
  - SDK 1.5 (Cupcake) after Alpha and Beta
    - New soft keyboard with “autocomplete” feature
  - SDK 1.6 (Donut)
    - Support Wide VGA
  - SDK 2.0/2.0.1/2.1 (Eclair)
    - Revamped UI, browser
- 2010
  - Nexus One released to the public
  - SDK 2.2 (Froyo)
    - Flash support, tethering
  - SDK 2.3 (Gingerbread)
    - UI update, system-wide copy-paste

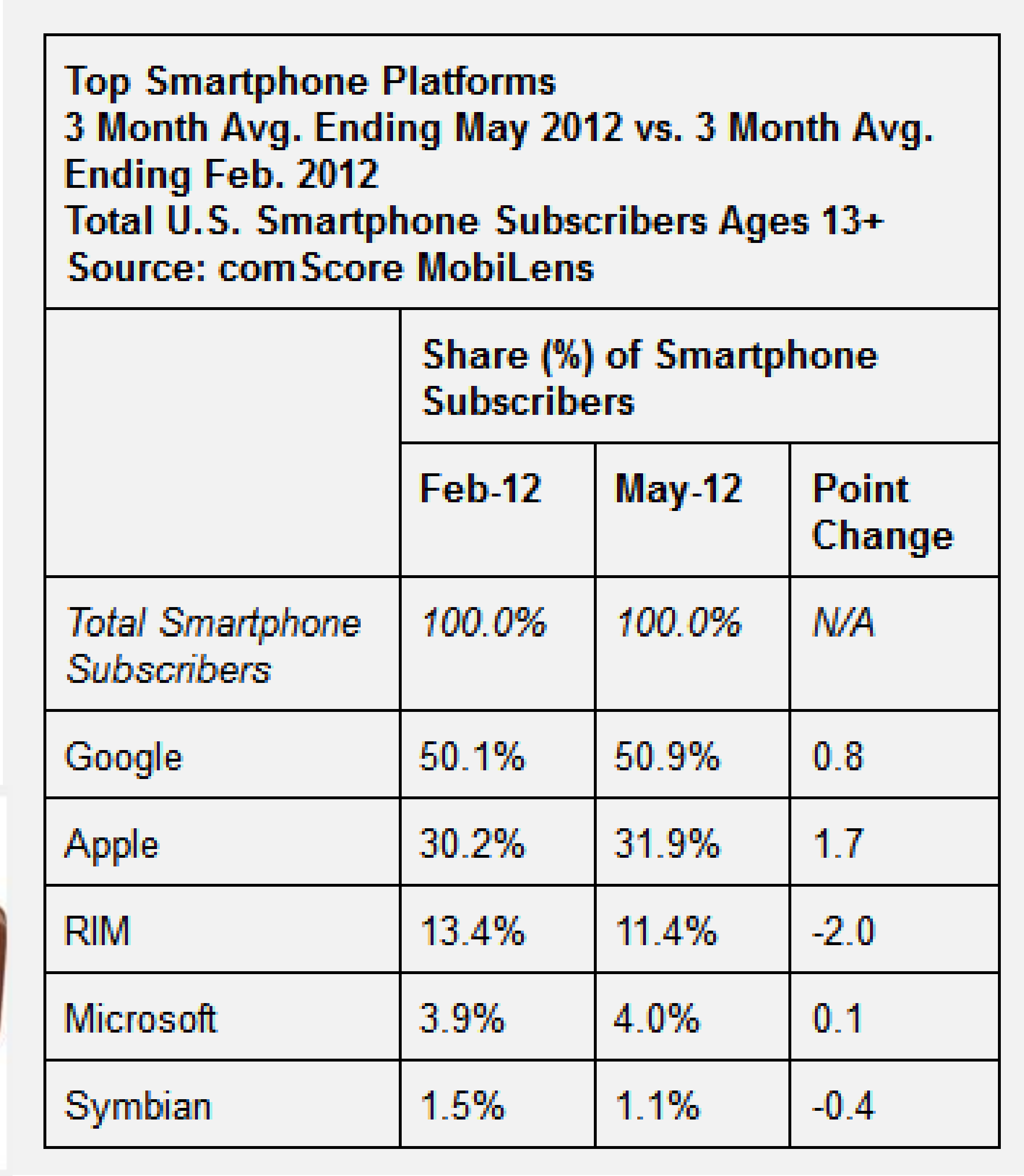
https://en.wikipedia.org/wiki/Android_version_history

Short History cont.

- 2011
  - SDK 3.0 (Honeycomb) for tablets only
    - New UI for tablets, support multi-core processors, fragments
  - SDK 3.1 and 3.2
    - Hardware support and UI improvements
  - SDK 4.0 (Ice Cream Sandwich)
    - For Q4, combination of Gingerbread Honeycomb

Short History cont.

- 2012
  - Android 4.1, "Jelly Bean" released in July
- 2013
  - Android 4.4, KitKat released October 31, 2013

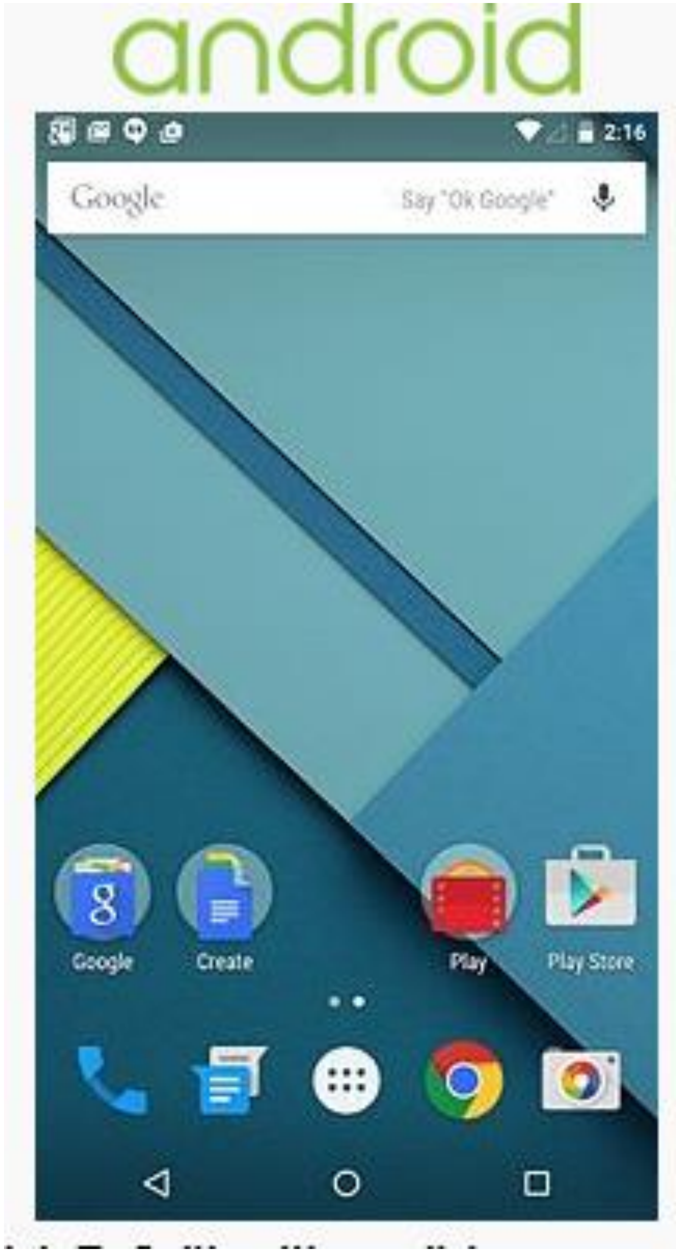


Top Smartphone Platforms
3 Month Avg. Ending May 2012 vs. 3 Month Avg. Ending Feb. 2012
Total U.S. Smartphone Subscribers Ages 13+
Source: comScore MobiLens

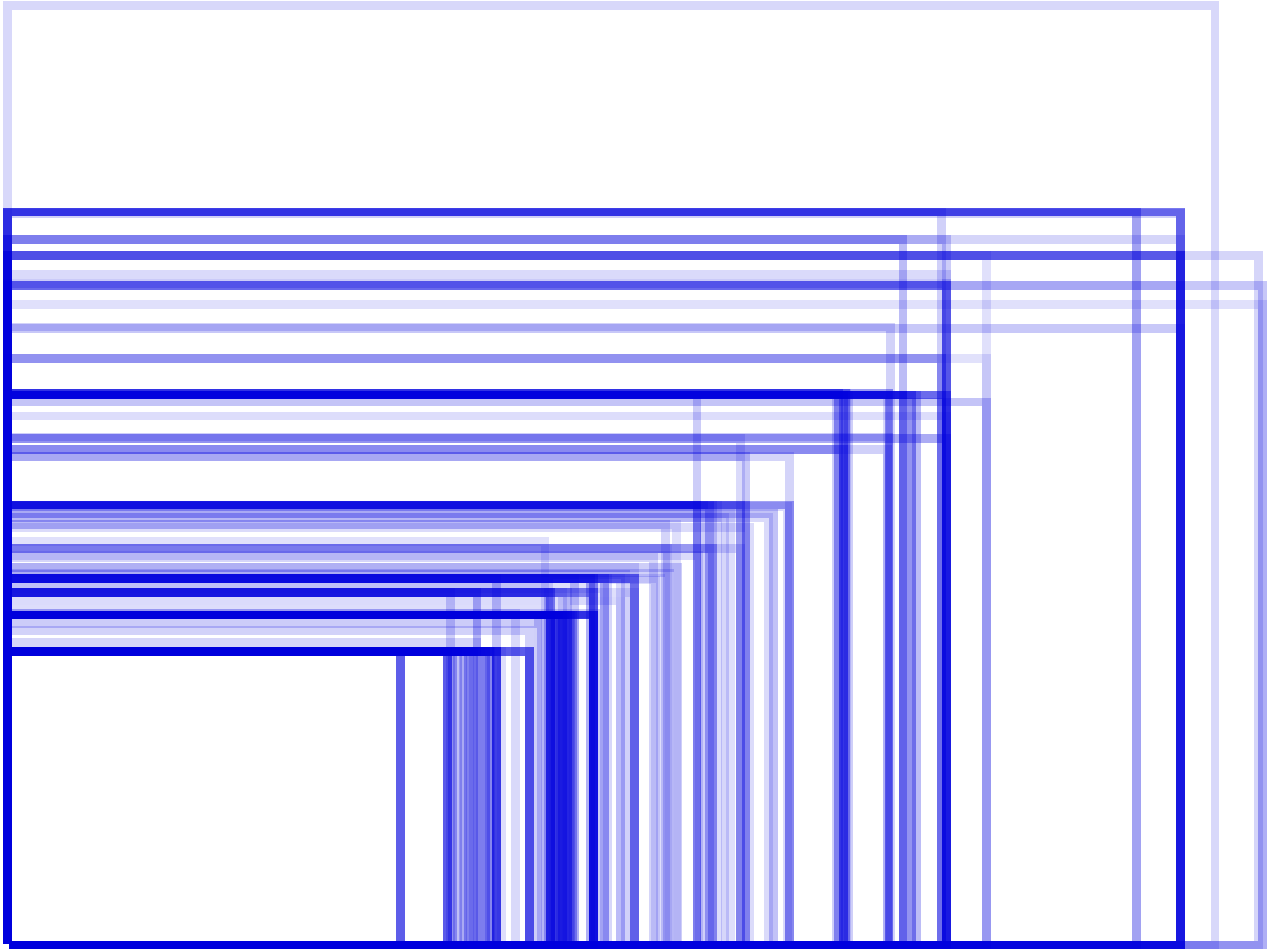
	Share (%) of Smartphone Subscribers		
	Feb-12	**May-12**	**Point Change**
Total Smartphone Subscribers	*100.0%*	*100.0%*	*N/A*
Google	50.1%	50.9%	0.8
Apple	30.2%	31.9%	1.7
RIM	13.4%	11.4%	-2.0
Microsoft	3.9%	4.0%	0.1
Symbian	1.5%	1.1%	-0.4

Short History (Getting Longer)

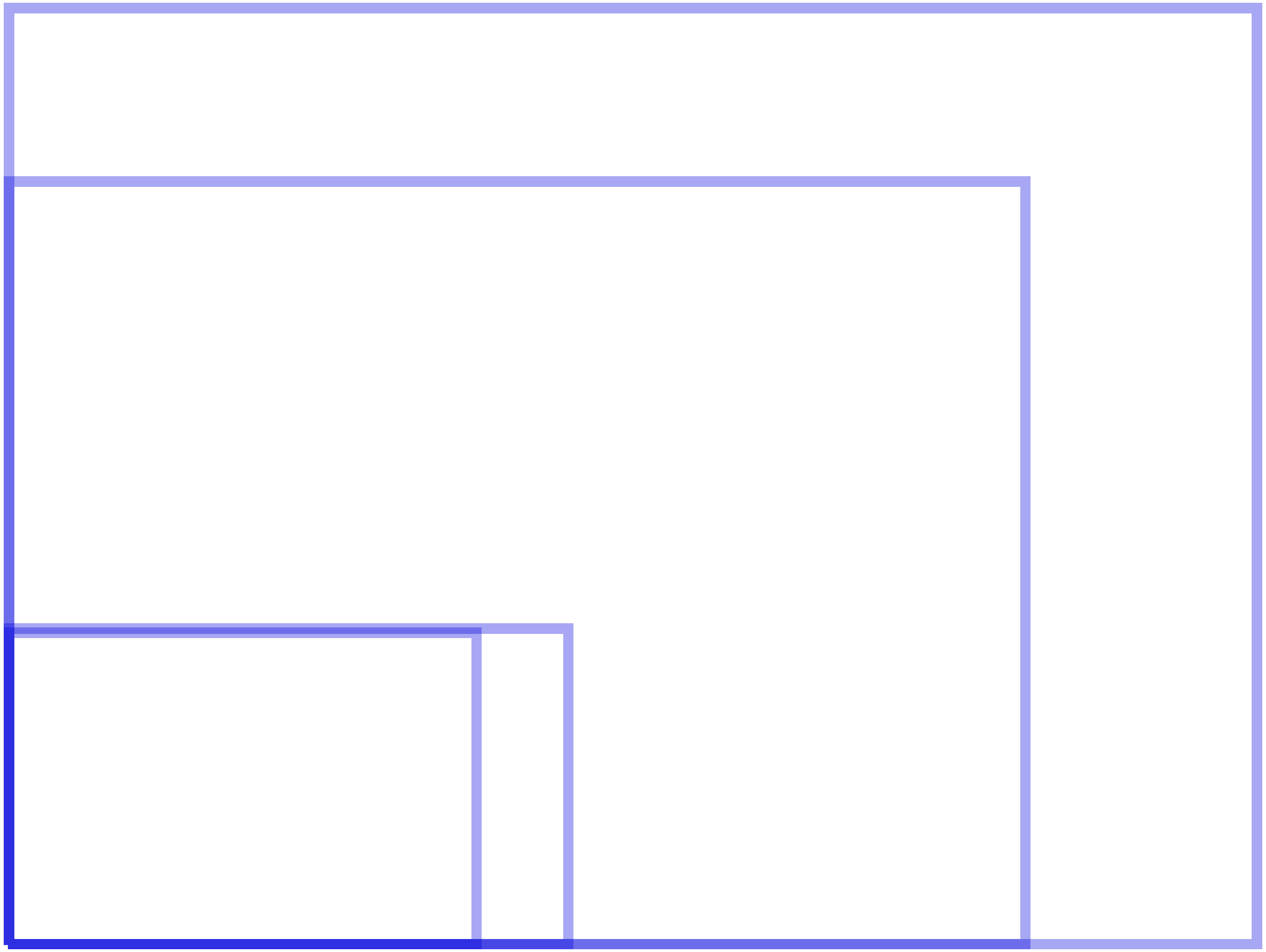
- November, 2014
  Android 5.0 Lollipop released.
  API level 21
  "Material Design"
- October, 2015
  Android 6.0
  Marshmallow
  API level 23



Android Screen Sizes - August 2014

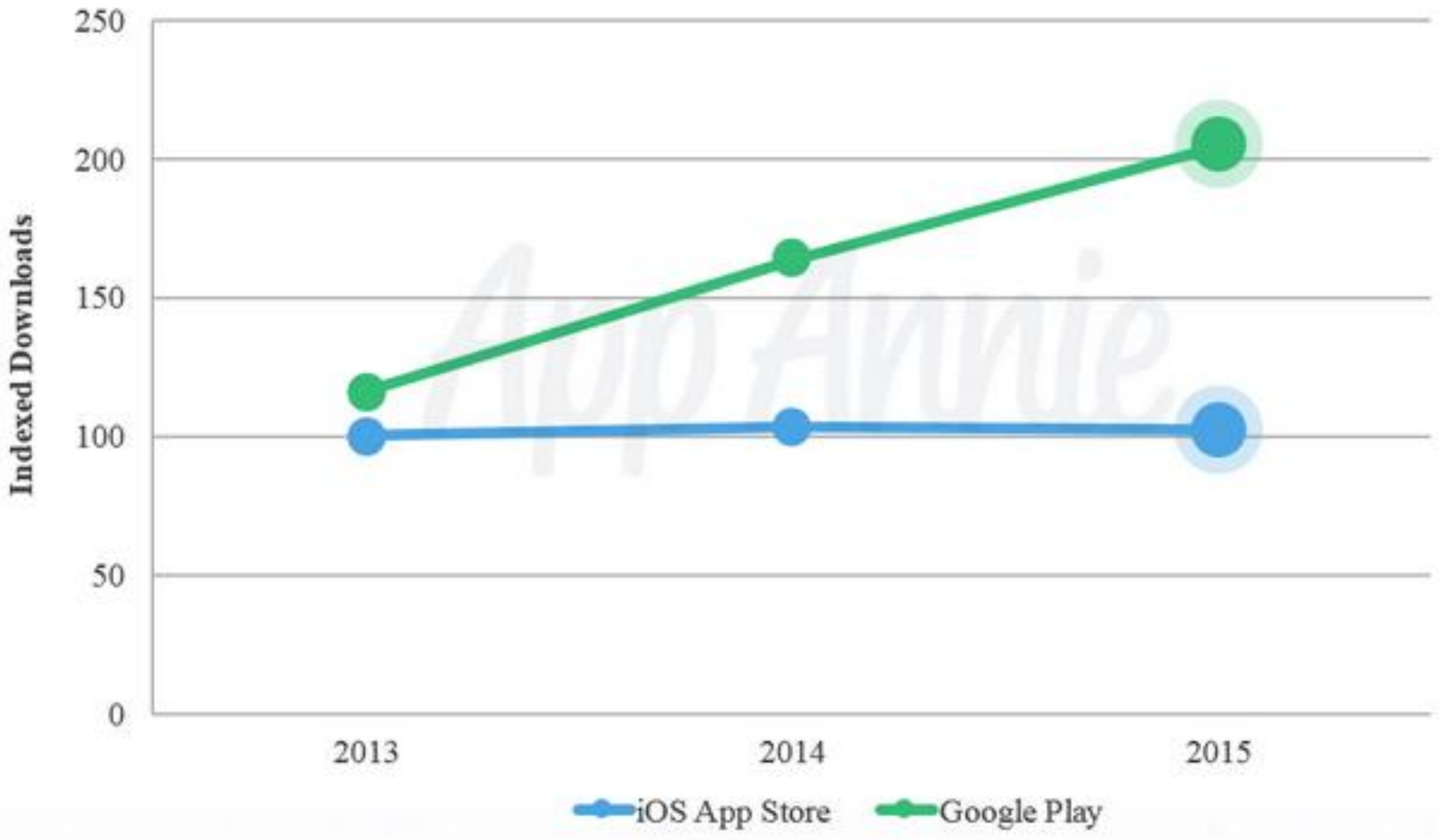


iOS Screen Sizes - August 2014



iPhone vs. Android

2015 App Downloads



https://www.appannie.com/

ANDROID DEVELOPMENT TOOLS

Setup Development Environment

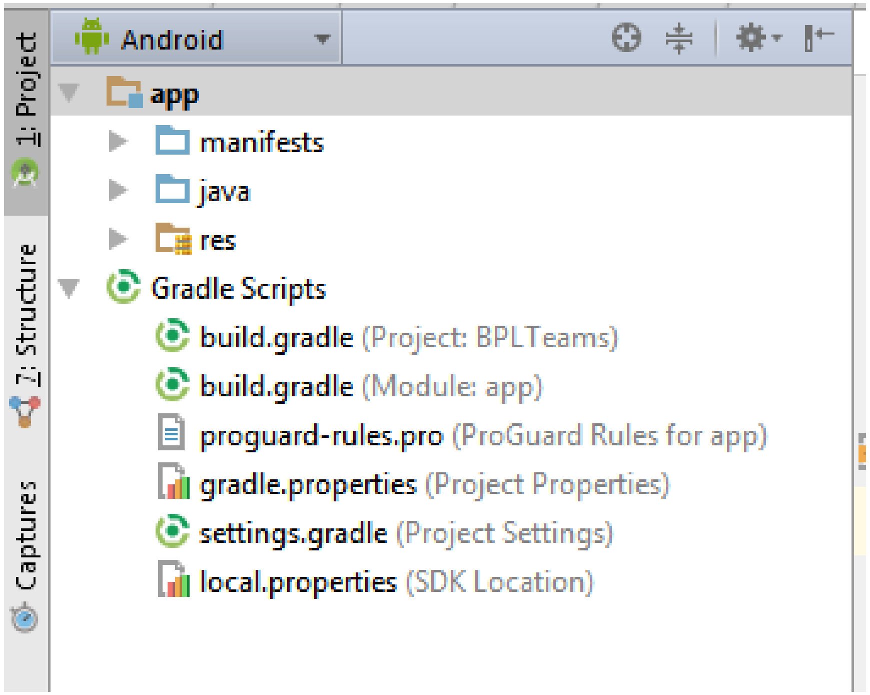
- Install JDK 8
- Install Android Studio
  - includes API level 23
- Use SDK manager to download lower API levels
  - I suggest down to 15
- Detailed install instructions available on Android site http://developer.android.com/sdk/installing.html

Elements of Android Projects

- ***Application Name***
  - seen by users on app chooser, app list, store
- ***Project Name***
  - in IDE, can be different, often directory
- ***Package Name***
  - Java package name, not using default package
- ***Minimum SDK Level***
  - how far back do you support, ~15 Jan 2016
- ***Target SDK Level***
  - device / api you had in mind for app, most recent?
- ***Theme***
  - look and feel of app, color scheme, various built in themes such as Theme, Holo, Material (Design)

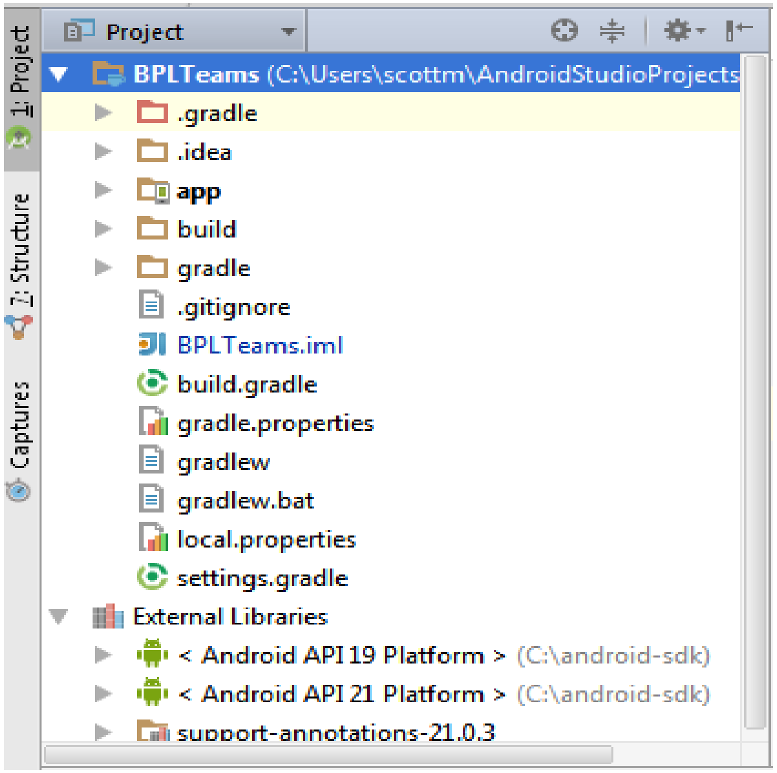
Android Projects

- Creating a project results in multiple files and resources being created



- app
  - manifests
  - java
  - res
- Gradle Scripts
  - build.gradle (Project: BPLTeams)
  - build.gradle (Module: app)
  - proguard-rules.pro (ProGuard Rules for app)
  - gradle.properties (Project Properties)
  - settings.gradle (Project Settings)
  - local.properties (SDK Location)

Android Project View



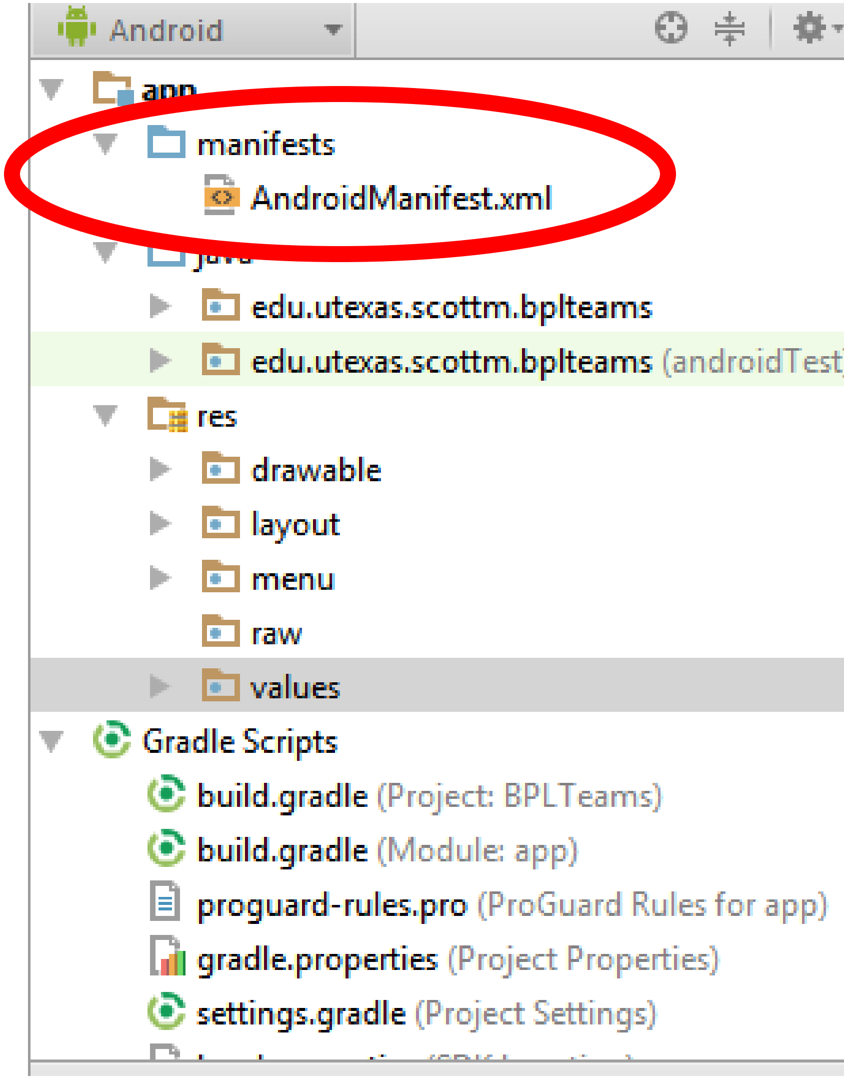
- BPLTeams (C:\Users\scottm\AndroidStudioProjects
  - .gradle
  - .idea
  - app
  - build
  - gradle
  - .gitignore
  - BPLTeams.iml
  - build.gradle
  - gradle.properties
  - gradlew
  - gradlew.bat
  - local.properties
  - settings.gradle
- External Libraries
  - < Android API 19 Platform > (C:\android-sdk)
  - < Android API 21 Platform > (C:\android-sdk)
  - support-annotations-21.0.3

Classic Project View

ANDROID PROJECT COMPONENTS

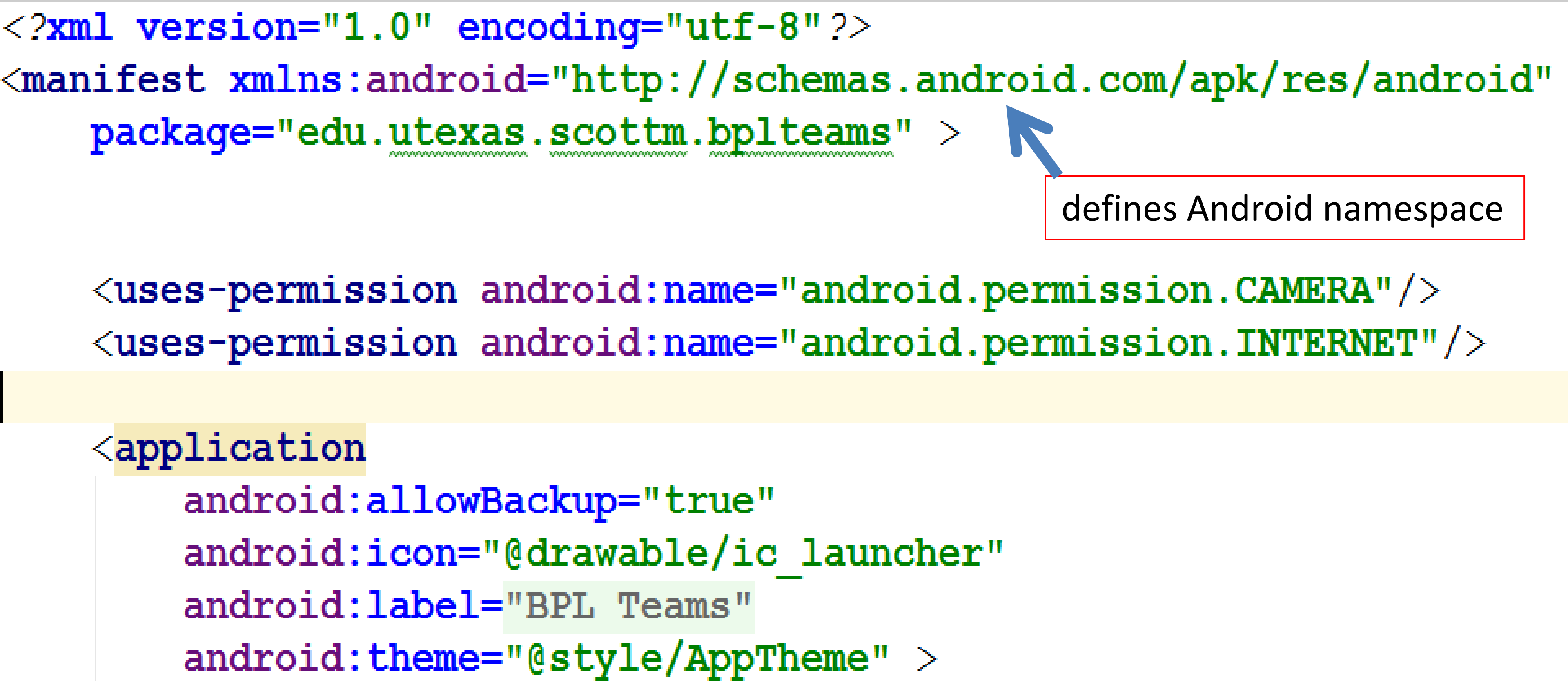
Android Projects - Components Manifest

- AndroidManifest.xml
- Like a table of contents for your app
- Main activity
- Target and min SDK
- Declare all the parts of your apps:
  - activities, services
- Request permissions
  - network, location, ...



```
app
  manifests
    AndroidManifest.xml
  java
    edu.utexas.scottm.bplteams
    edu.utexas.scottm.bplteams (androidTest)
  res
    drawable
    layout
    menu
    raw
    values
Gradle Scripts
  build.gradle (Project: BPLTeams)
  build.gradle (Module: app)
  proguard-rules.pro (ProGuard Rules for app)
  gradle.properties (Project Properties)
  settings.gradle (Project Settings)
```

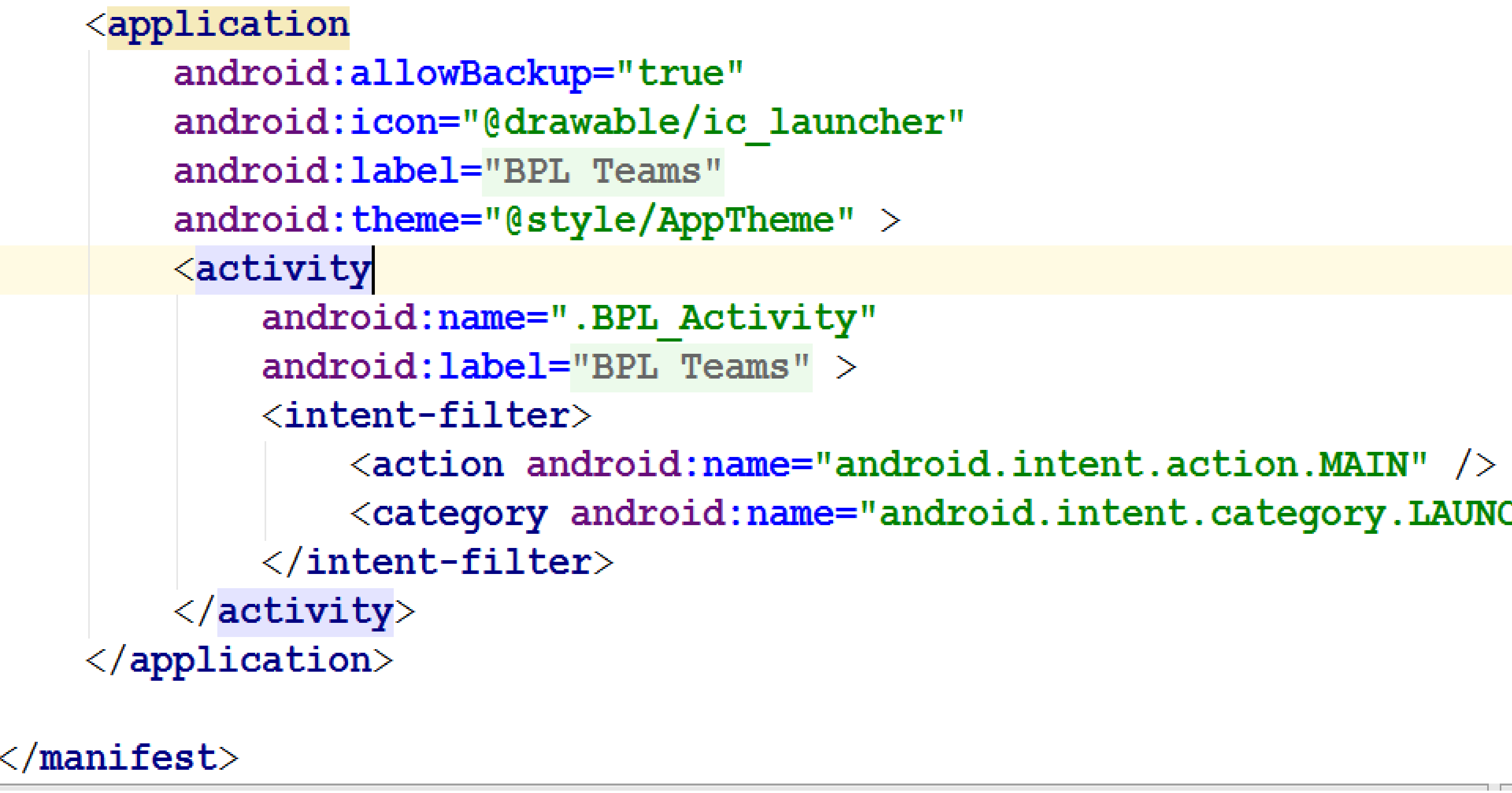
Android Manifest - Sample



```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="edu.utexas.scottm.bplteams" >

    <uses-permission android:name="android.permission.CAMERA"/>
    <uses-permission android:name="android.permission.INTERNET"/>

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="BPL Teams"
        android:theme="@style/AppTheme" >
```

defines Android namespace

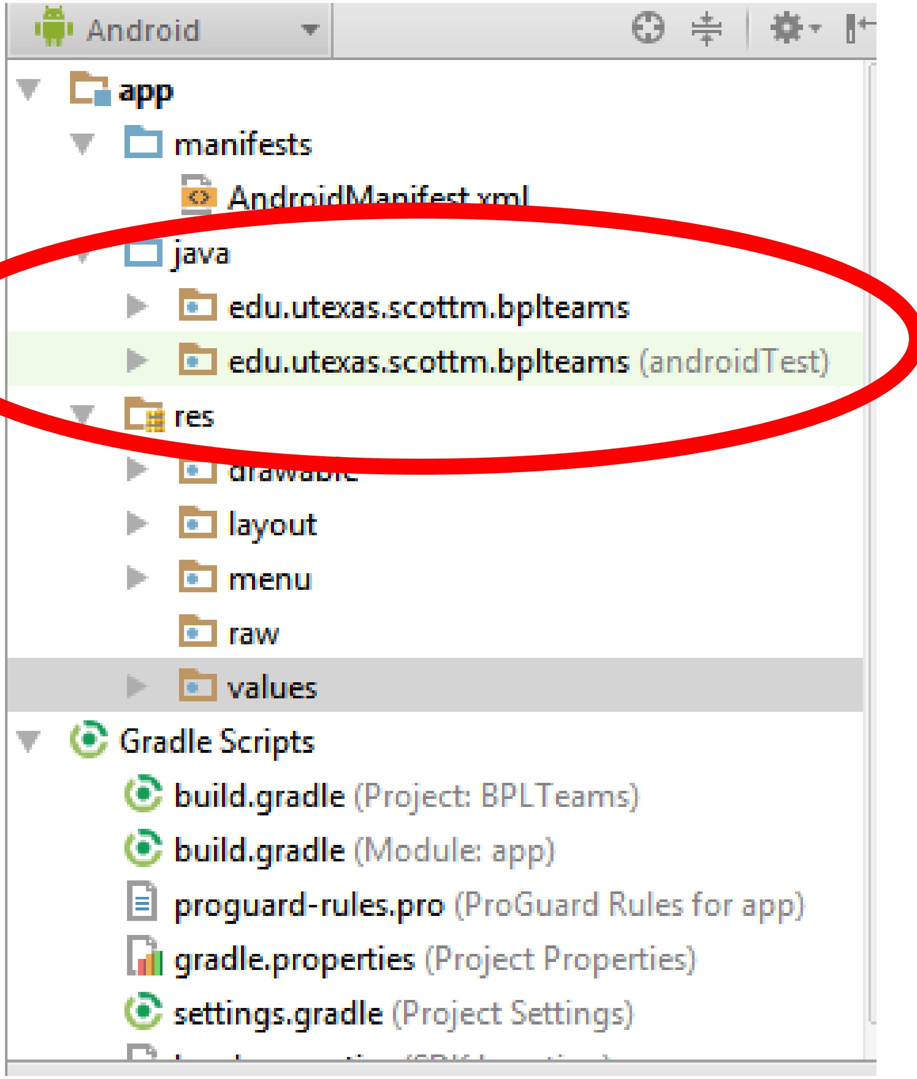
Android Manifest - Sample



```
    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="BPL Teams"
        android:theme="@style/AppTheme" >
        <activity
            android:name=".BPL_Activity"
            android:label="BPL Teams" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNC
            </intent-filter>
        </activity>
    </application>

</manifest>
```

Android Projects - Components
Java Source Code

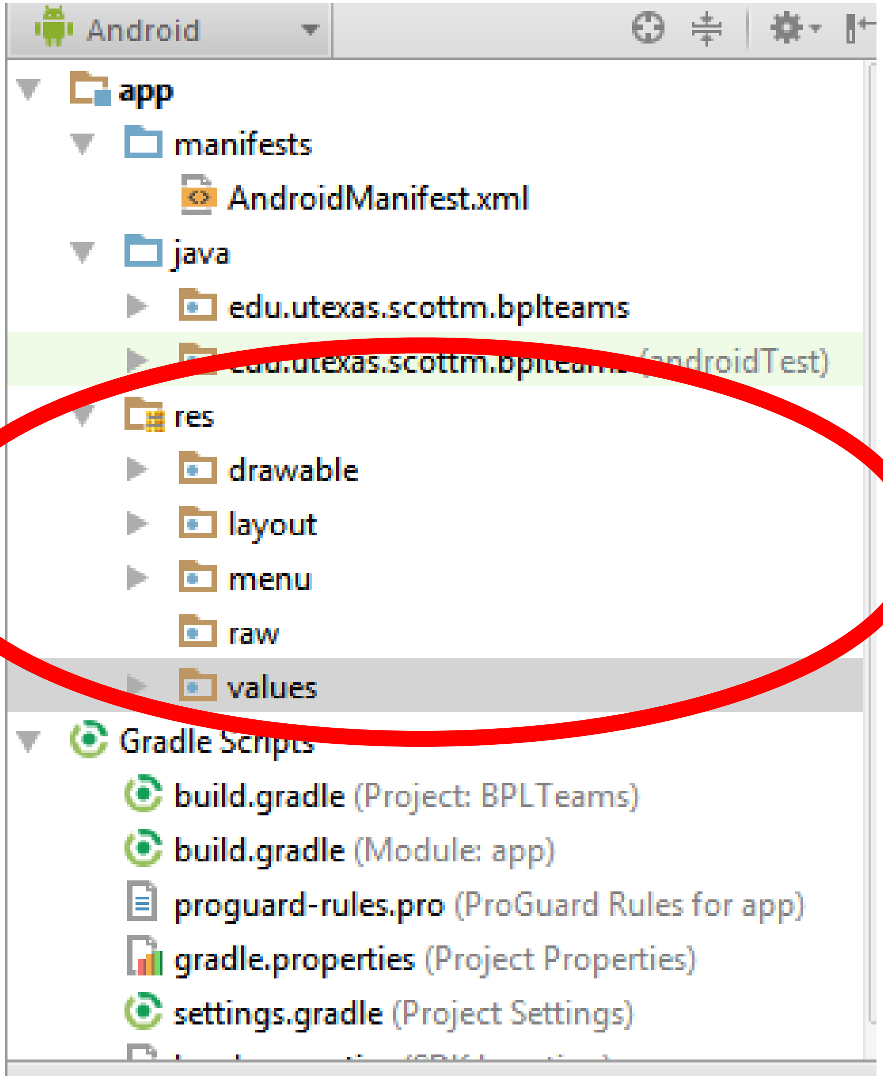
- Source Code:
- In java directory in Android Project View
- Actually in src directory on system



```
Android
app
  manifests
    AndroidManifest.xml
  java
    edu.utexas.scottm.bplteams
    edu.utexas.scottm.bplteams (androidTest)
  res
    drawable
    layout
    menu
    raw
    values
Gradle Scripts
  build.gradle (Project: BPLTeams)
  build.gradle (Module: app)
  proguard-rules.pro (ProGuard Rules for app)
  gradle.properties (Project Properties)
  settings.gradle (Project Settings)
```

Android Projects - Components Resources

- Resources or the res directory
- non source code resources for the app
- packaged up with app
- large role and use in development of app



```
app
    manifests
        AndroidManifest.xml
    java
        edu.utexas.scottm.bplteams
        edu.utexas.scottm.bplteams (androidTest)
    res
        drawable
        layout
        menu
        raw
        values
Gradle Scripts
    build.gradle (Project: BPLTeams)
    build.gradle (Module: app)
    proguard-rules.pro (ProGuard Rules for app)
    gradle.properties (Project Properties)
    settings.gradle (Project Settings)
    local.properties (SDK Location)
```

Resource Directories

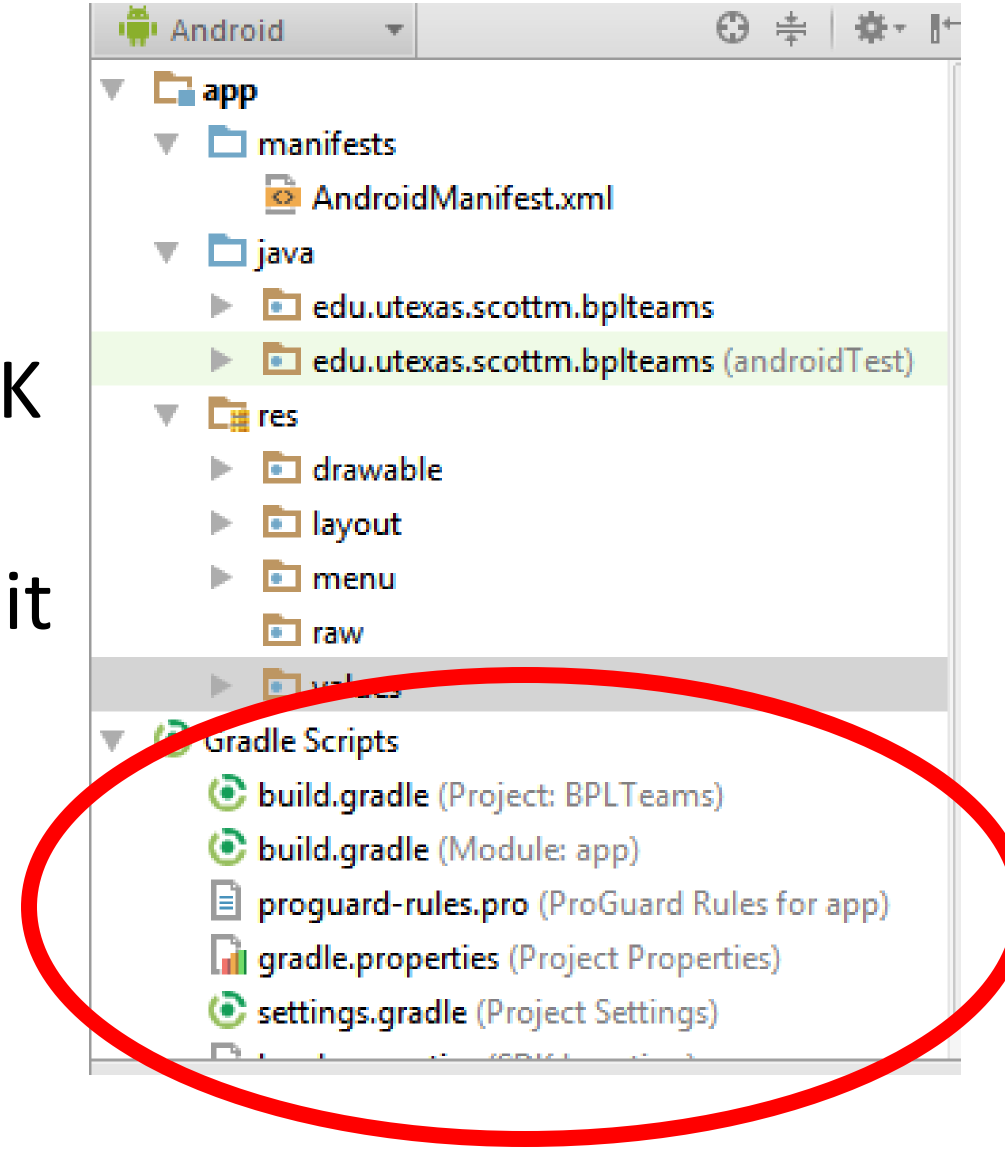
- res/drawable for graphic images such as png, jpeg
- res/layout for xml files that define the layout of user interfaces inside the app
- res/menu for xml based menu specifications
- res/values for lists of strings, dimensions, colors, lists of data
- res/raw for other kinds of files such as audio clips, video clips, csv files, raw text
- res/xml for other general purpose xml files

Gradle

- .apk files, Android Package Kit
  - Android executables
- Development environment takes, source code, manifest, libraries, resources, etc and packages them together in an APK
- some things known and set
- some things variable and configurable
- Gradle

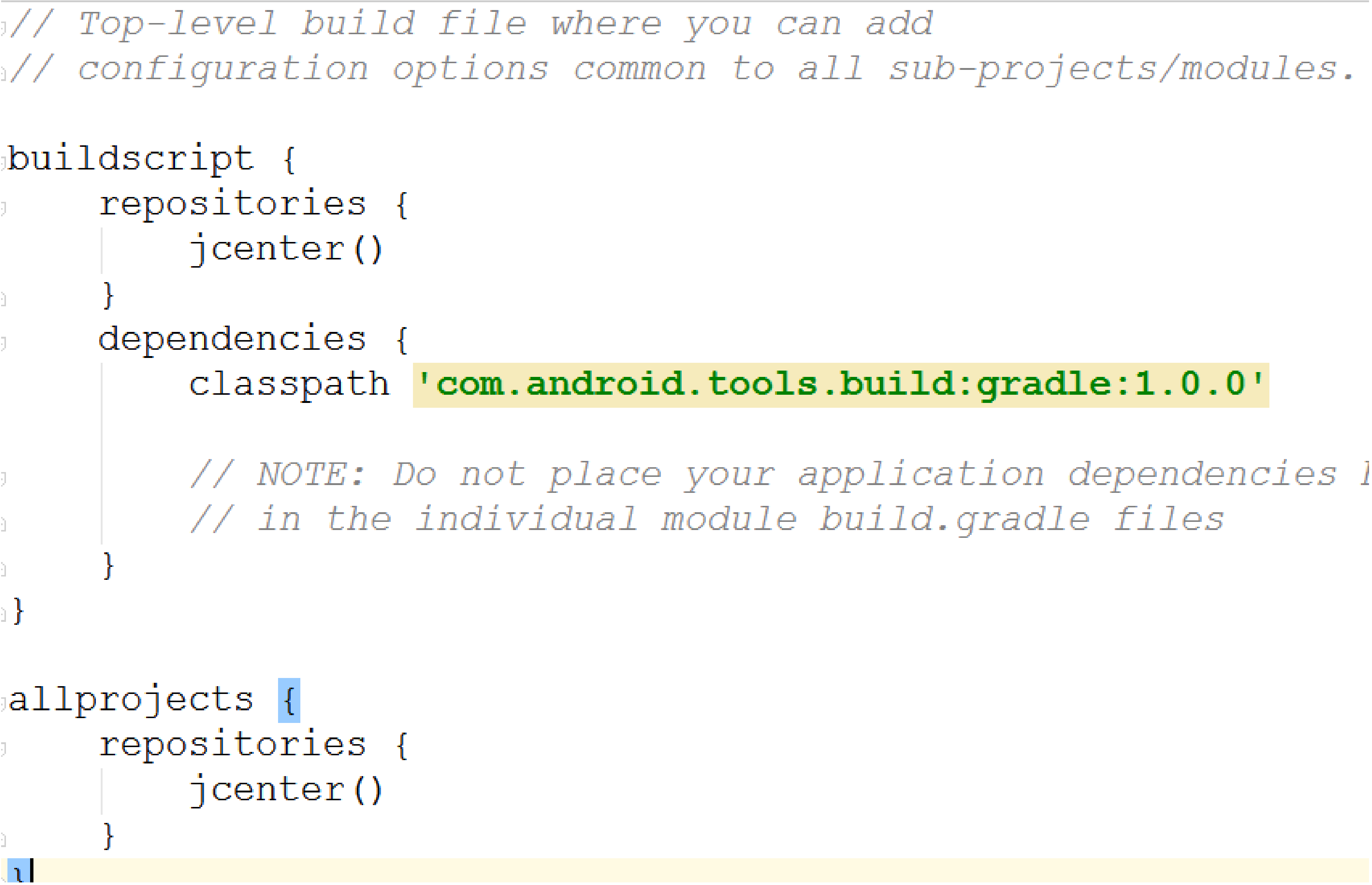
Gradle

- Gradle is the build engine that Android Studio uses to convert your project into an APK
- What needs to be created and how to do it
- Like
  - make for C/C++
  - Ant/Maven for Java
- build.gradle file



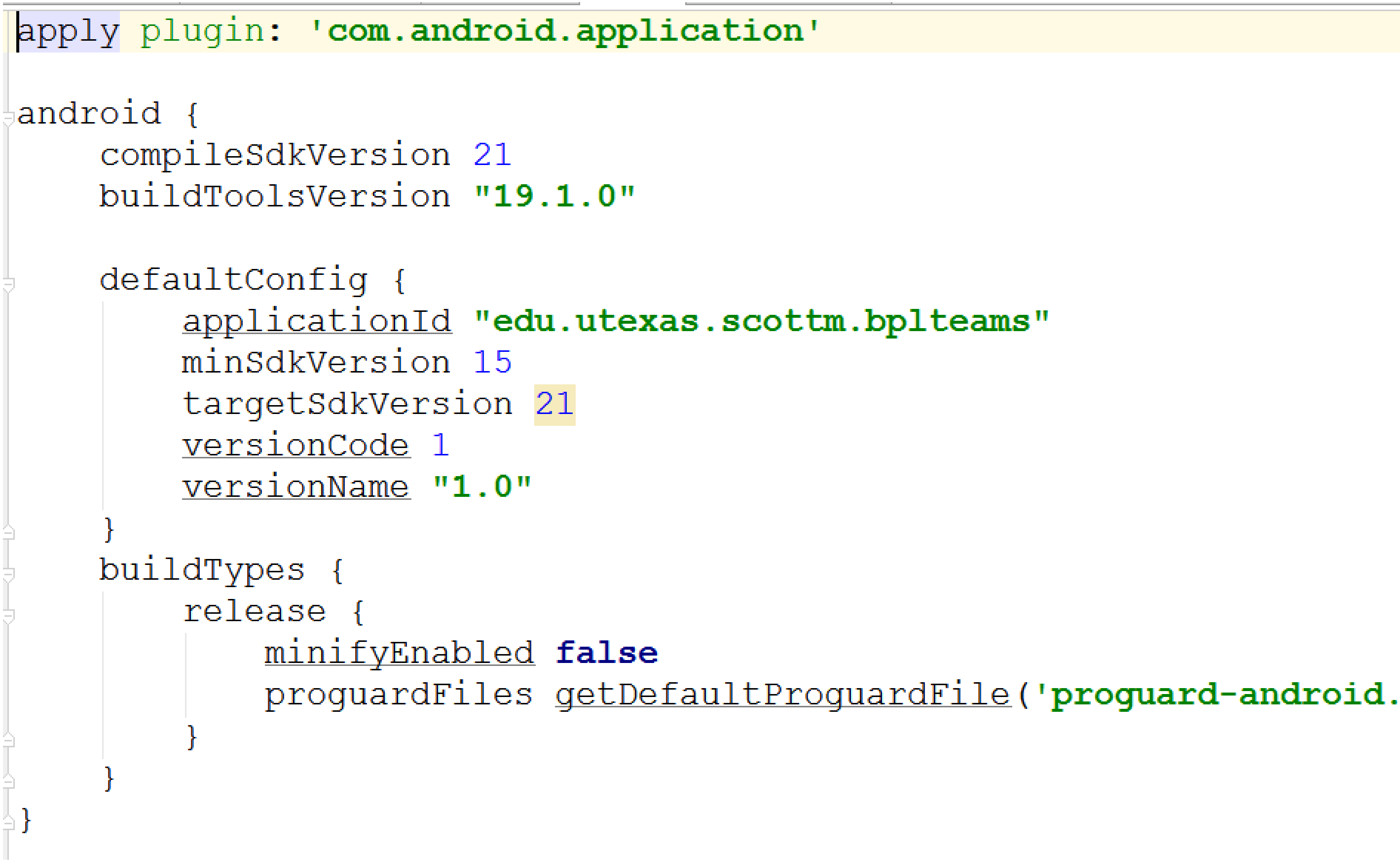
- app
  - manifests
    - AndroidManifest.xml
  - java
    - edu.utexas.scottm.bplteams
    - edu.utexas.scottm.bplteams (androidTest)
  - res
    - drawable
    - layout
    - menu
    - raw
    - values
- Gradle Scripts
  - build.gradle (Project: BPLTeams)
  - build.gradle (Module: app)
  - proguard-rules.pro (ProGuard Rules for app)
  - gradle.properties (Project Properties)
  - settings.gradle (Project Settings)

sample build.gradle file - PROJECT



```
// Top-level build file where you can add
// configuration options common to all sub-projects/modules.

buildscript {
    repositories {
        jcenter()
    }
    dependencies {
        classpath 'com.android.tools.build:gradle:1.0.0'

        // NOTE: Do not place your application dependencies
        // in the individual module build.gradle files
    }
}

allprojects {
    repositories {
        jcenter()
    }
}
```

sample build.gradle file - MODULE / APP

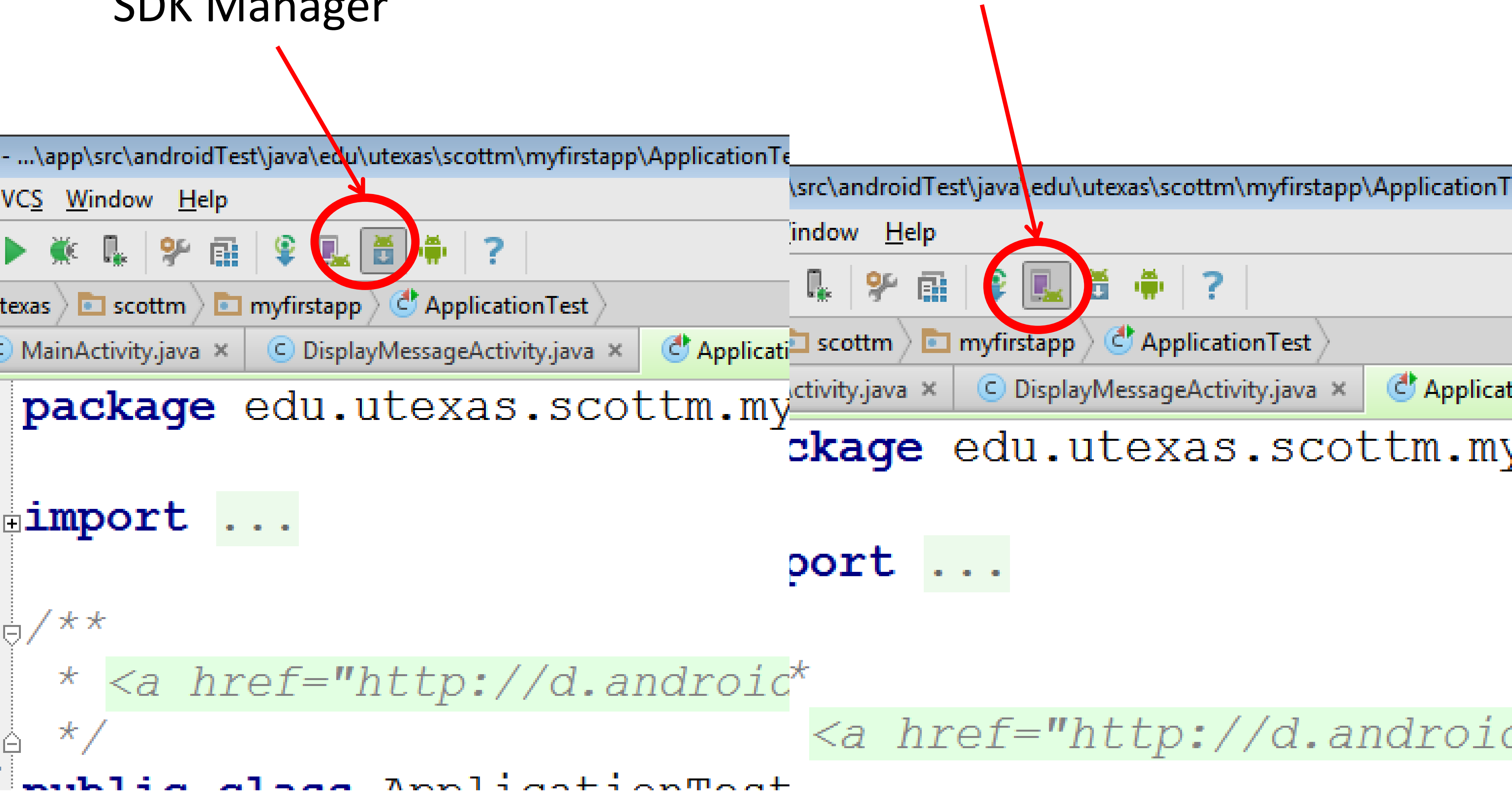


```
apply plugin: 'com.android.application'

android {
    compileSdkVersion 21
    buildToolsVersion "19.1.0"

    defaultConfig {
        applicationId "edu.utexas.scottm.bplteams"
        minSdkVersion 15
        targetSdkVersion 21
        versionCode 1
        versionName "1.0"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.
        }
    }
}
```

EMULATORS

SDK Manager

AVD Manager


- ...\app\src\androidTest\java\edu\utexas\scottm\myfirstapp\ApplicationTe
VCS Window Help
?
texas
scottm
myfirstapp
ApplicationTest
MainActivity.java ×
DisplayMessageActivity.java ×
Applicati
\src\androidTest\java\edu\utexas\scottm\myfirstapp\ApplicationT
indow Help
?
scottm
myfirstapp
ApplicationTest
ctivity.java ×
DisplayMessageActivity.java ×
Applicat

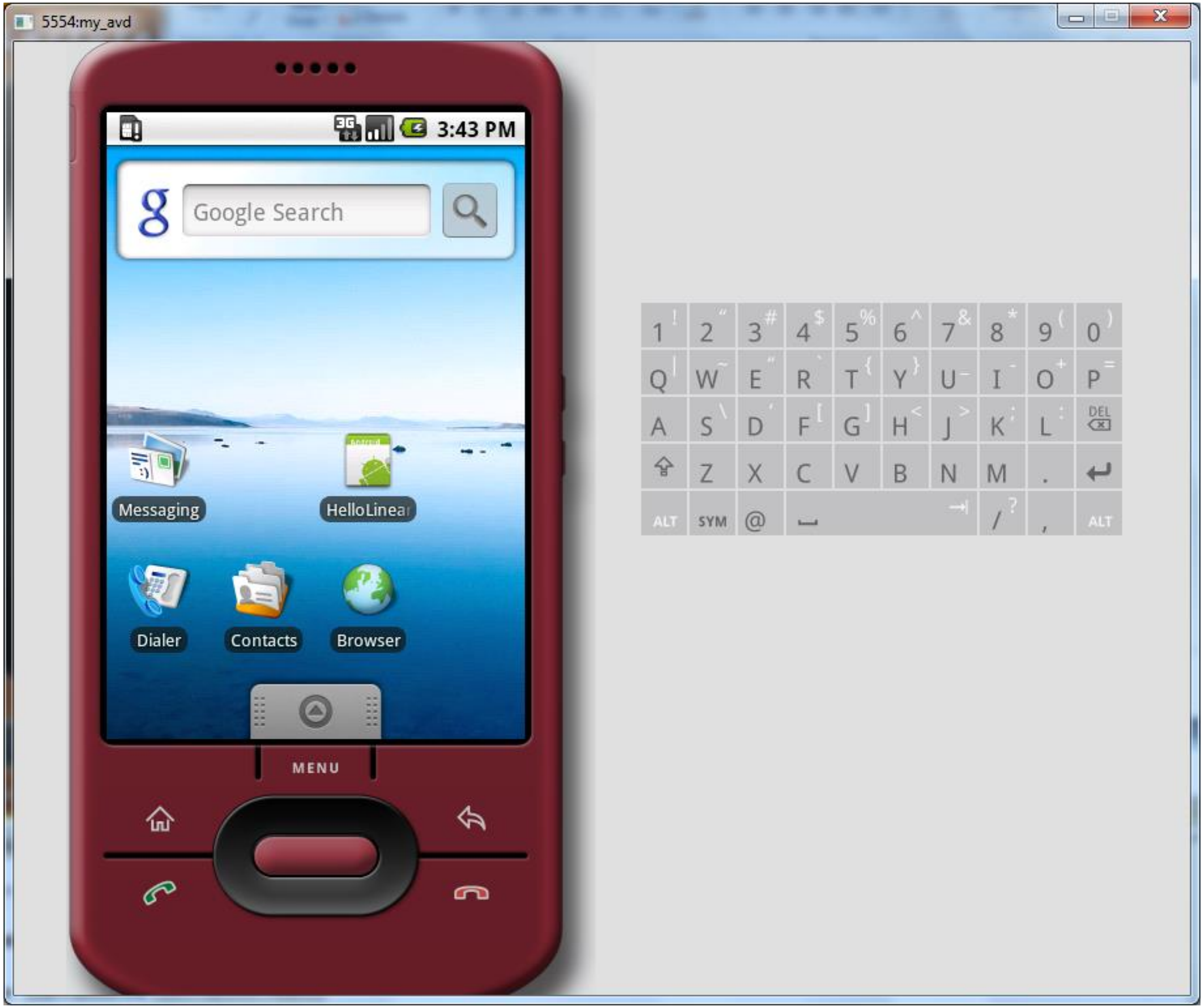
```
package edu.utexas.scottm.my

import ...

/**
 * <a href="http://d.androic
 */
public class ApplicationTest
```

```
ckage edu.utexas.scottm.my

port ...

*
<a href="http://d.androic
```

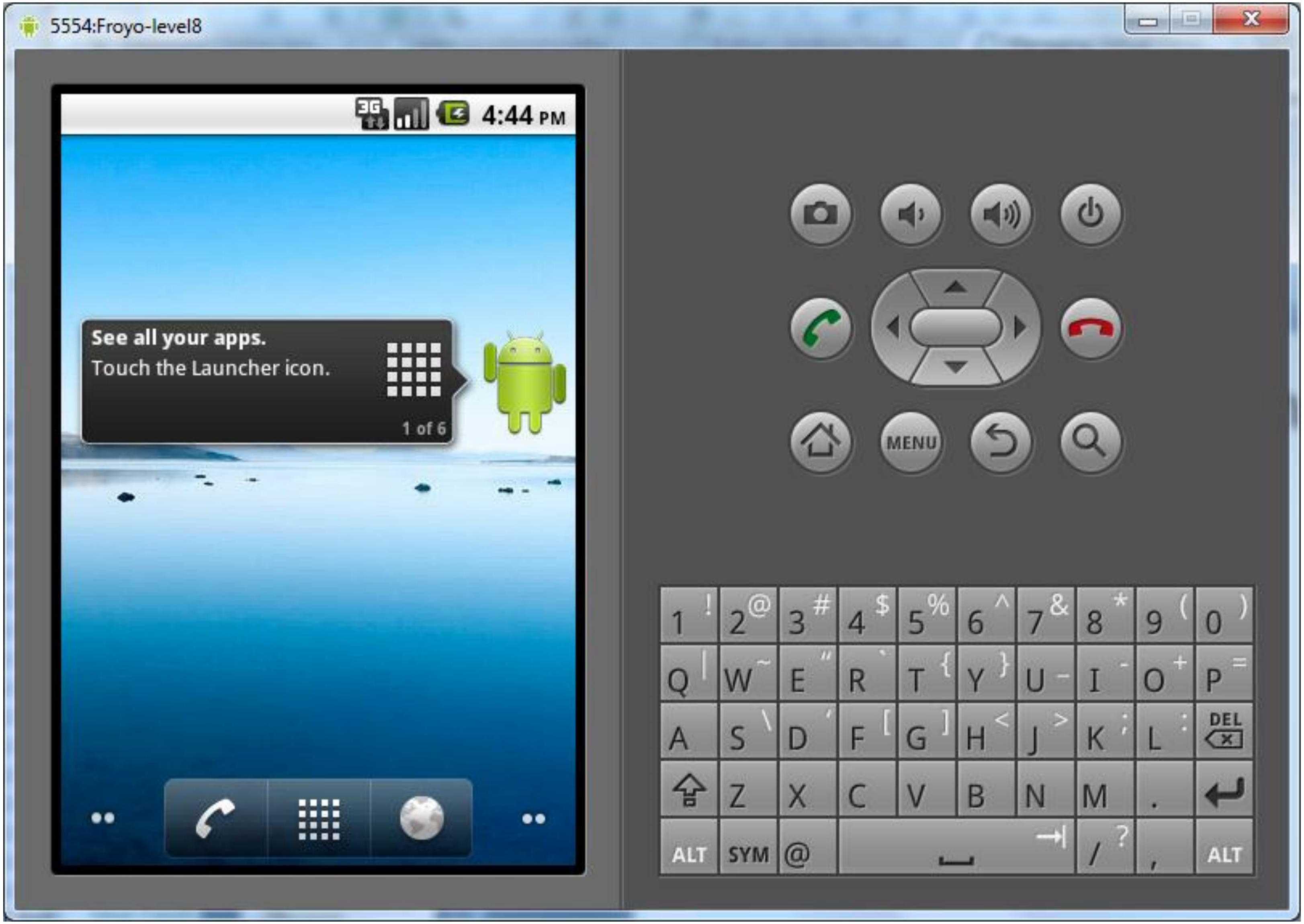
Android Emulator or AVD

- Emulator is useful for testing apps but is not a substitute for a real device
- Emulators are called **Android Virtual Devices** (AVDs)
- Android SDK and AVD Manager allows you to create AVDs that target any Android API level
- AVD have configurable resolutions, RAM, SD cards, skins, and other hardware

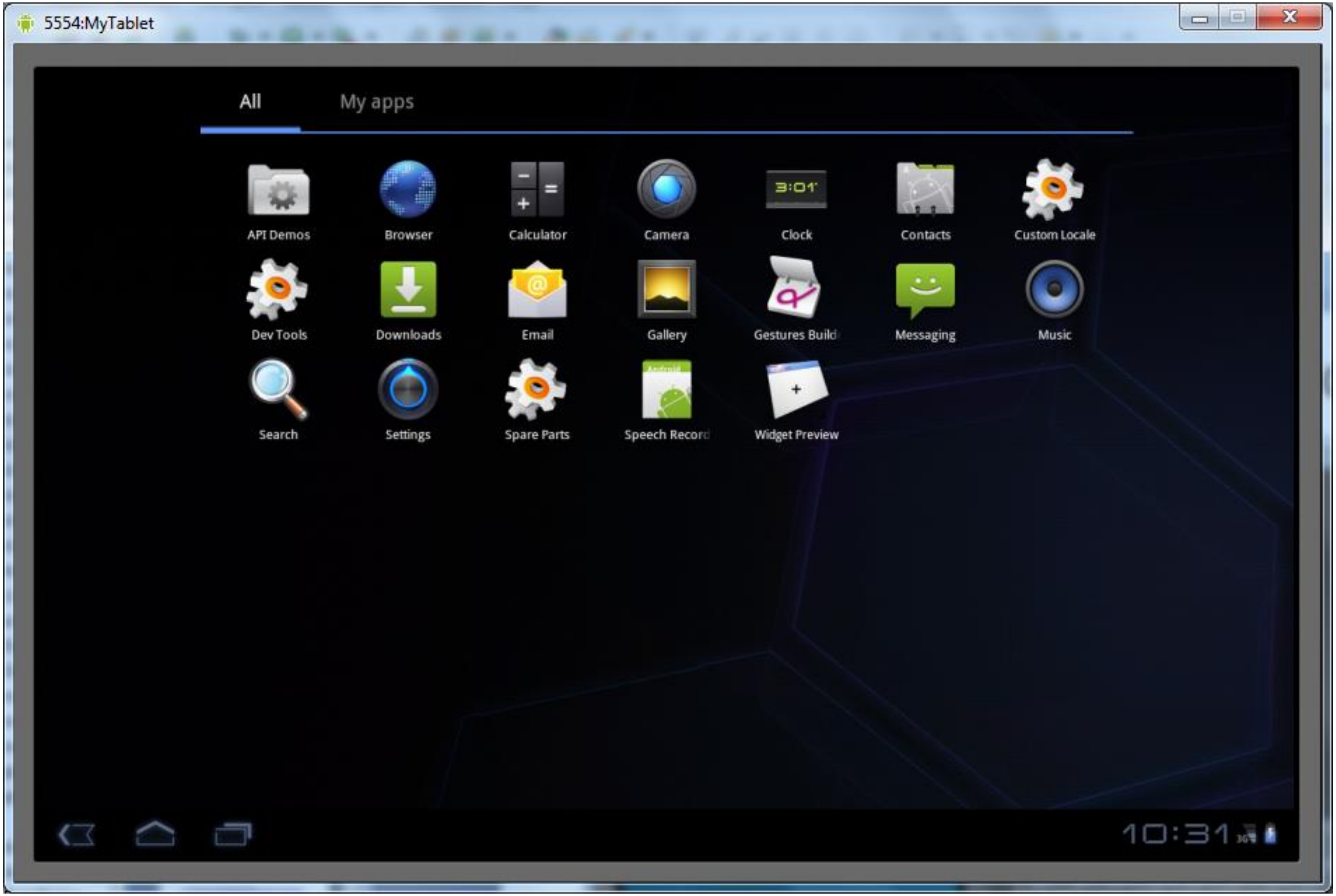
Android Emulator: 1.6



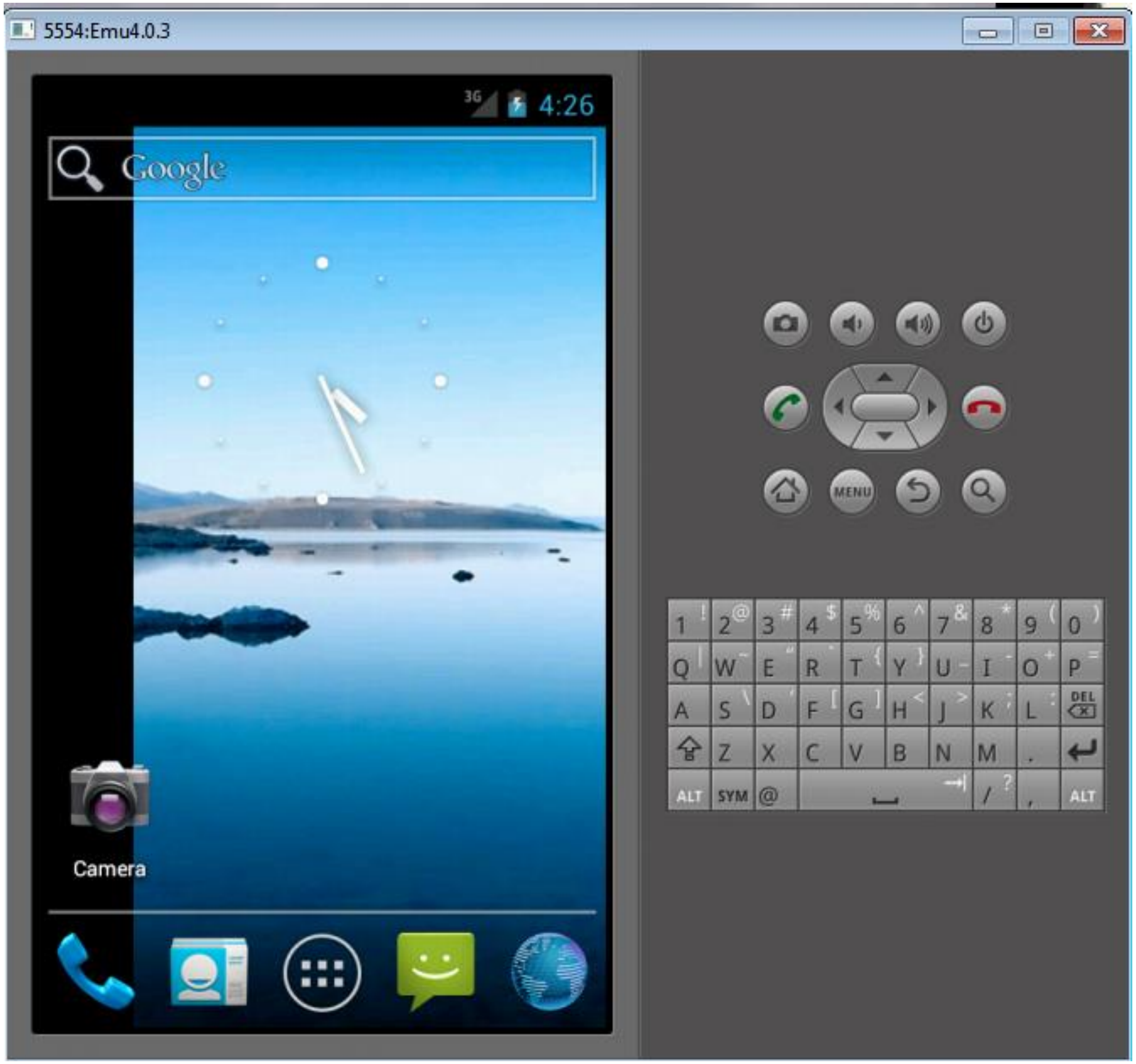
Android Emulator: 2.2



Android Emulator: 3.0



Android Emulator: 4.0



Emulator Basics

- Host computer’s keyboard works
- Host’s mouse acts as finger
- Uses host’s Internet connection
- Other buttons work: Home, Menu, Back, Search, volume up and down, etc.
- Ctrl-F11 toggle landscape → portrait
- Alt-Enter toggle full-screen mode
- More info at http://developer.android.com/guide/developing/devices/emulator.html

Emulator Limitations

- No support for placing or receiving actual phone calls
  - Simulate phone calls (placed and received) through the emulator console
- No support for USB connections
- No support for camera/video capture (input)
- No support for device-attached headphones
- No support for determining connected state
- No support for determining battery charge level and AC charging state
- No support for determining SD card insert/eject
- No support for Bluetooth
- No support for simulating the accelerometer
  - Use OpenIntents's Sensor Simulator

That's why we need the dev phones and tablets!

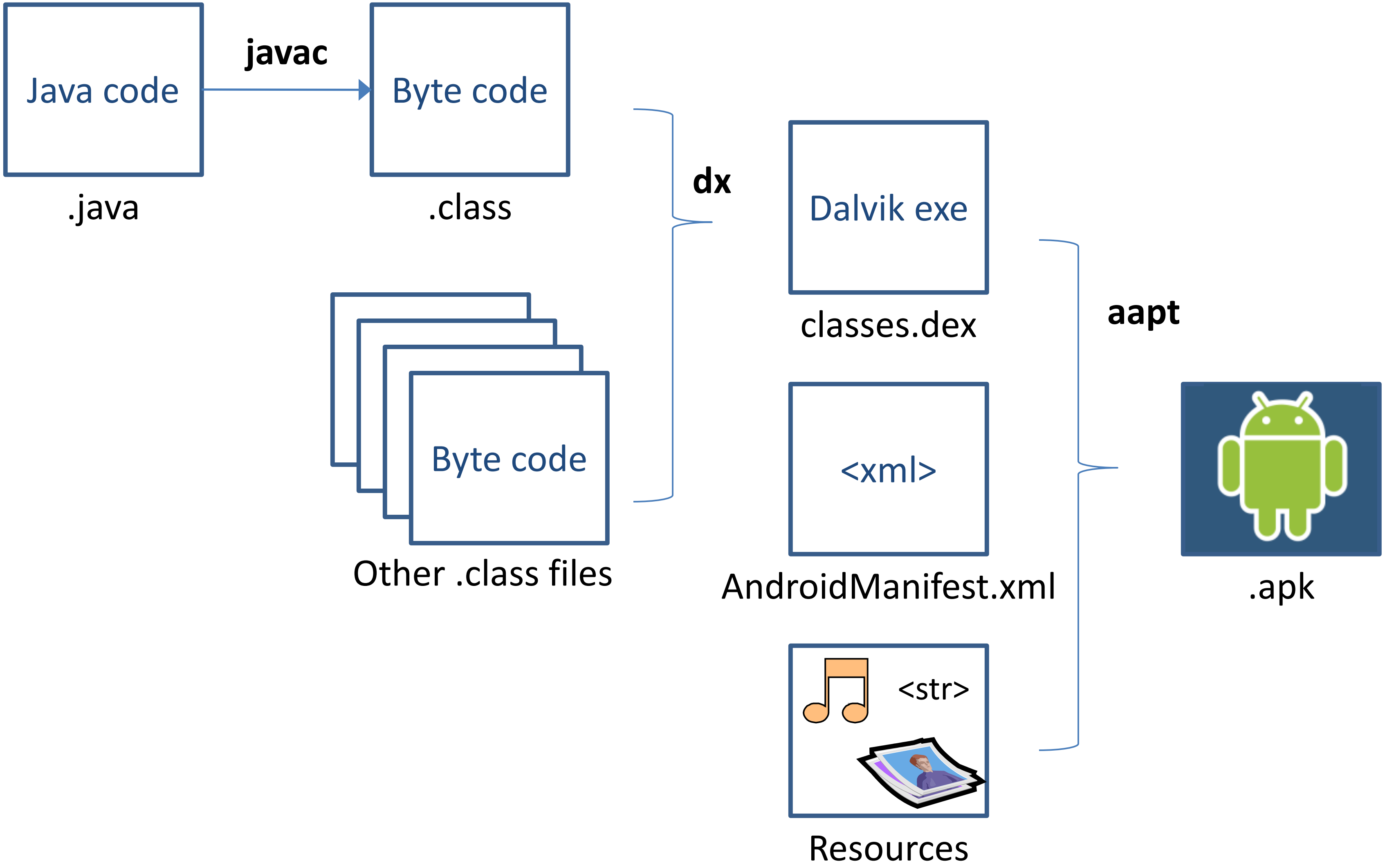
Android Runtime: Dalvik VM

- Subset of Java developed by Google
- Optimized for mobile devices (better memory management, battery utilization, etc.)
- Dalvik runs .dex files that are compiled from .class files
- Introduces new libraries
- Does not support some Java libraries like AWT, Swing
- http://developer.android.com/reference/packages.html

Applications Are Boxed

- By default, each app is run in its own Linux process
  - Process started when app's code needs to be executed
  - Threads can be started to handle time-consuming operations
- Each process has its own Dalvik VM
- By default, each app is assigned unique Linux ID
  - Permissions are set so app's files are only visible to that app

Producing an Android App



Java code

javac

Byte code

.java

.class

dx

Dalvik exe

classes.dex

Byte code

Other .class files

aapt

<xml>

AndroidManifest.xml

.apk

<str>

Resources

Other Dev Tools

- Android Debug Bridge
- Part of SDK
- command line tool to communicate with an emulator or connected Android device
  - check devices attached / running
  - install apk's, **A**ndroid **P**ac**K**age files, "executables", can find samples on places besides Google Play (security?)
  - and more!

  http://developer.android.com/guide/developing/tools/adb.html

Dalvik Debug Monitor Server

- DDMS
- debugging tool
- "provides, screen capture on the device, thread and heap information on the device, logcat, process, and radio state information, incoming call and SMS spoofing, location data spoofing, and more."
- can interact with DDMS via Android Studio

Getting Active Through Activities

Application Anatomy

There are 4 types of application components/building blocks:

Activities

1. Activity provides user interface
2. Usually represents a single screen
3. Can contain one or more views
4. Extends the Activity base class

Services

1. No user interface
2. Runs in background
3. Extends the Service base class

BroadcastReceiver

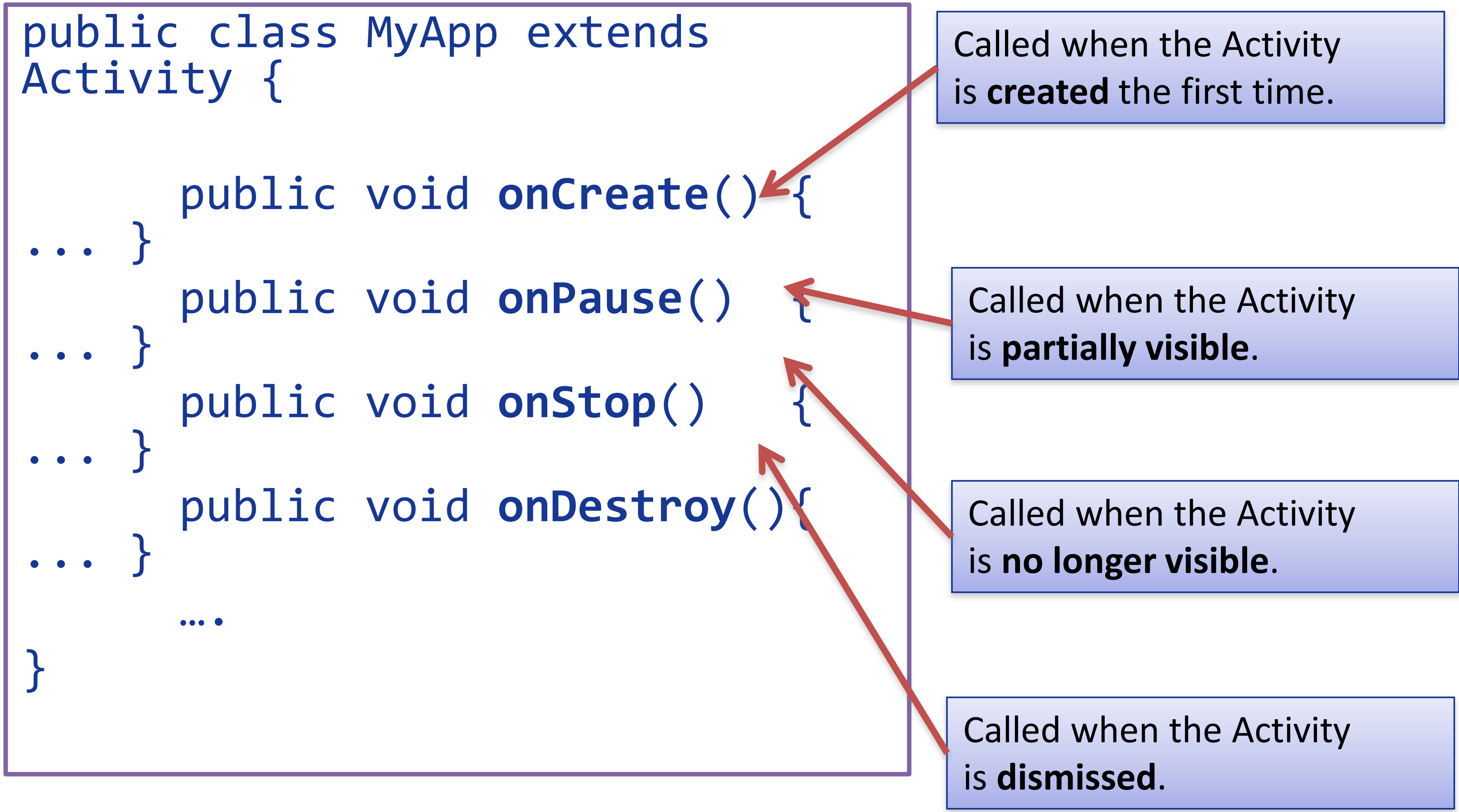
1. Receives and Reacts to broadcast Intents
2. No UI but can start an Activity
3. Extends the BroadcastReceiver base class

ContentProviders

1. Makes application data available to other apps [data sharing]
2. Uses SQLite database as storage
3. Extends the ContentProvider base class

Getting Active Through Activities

Activity



```
public class MyApp extends
Activity {

        public void onCreate() {
... }
        public void onPause()  {
... }
        public void onStop()   {
... }
        public void onDestroy(){
... }
        ….
}
```

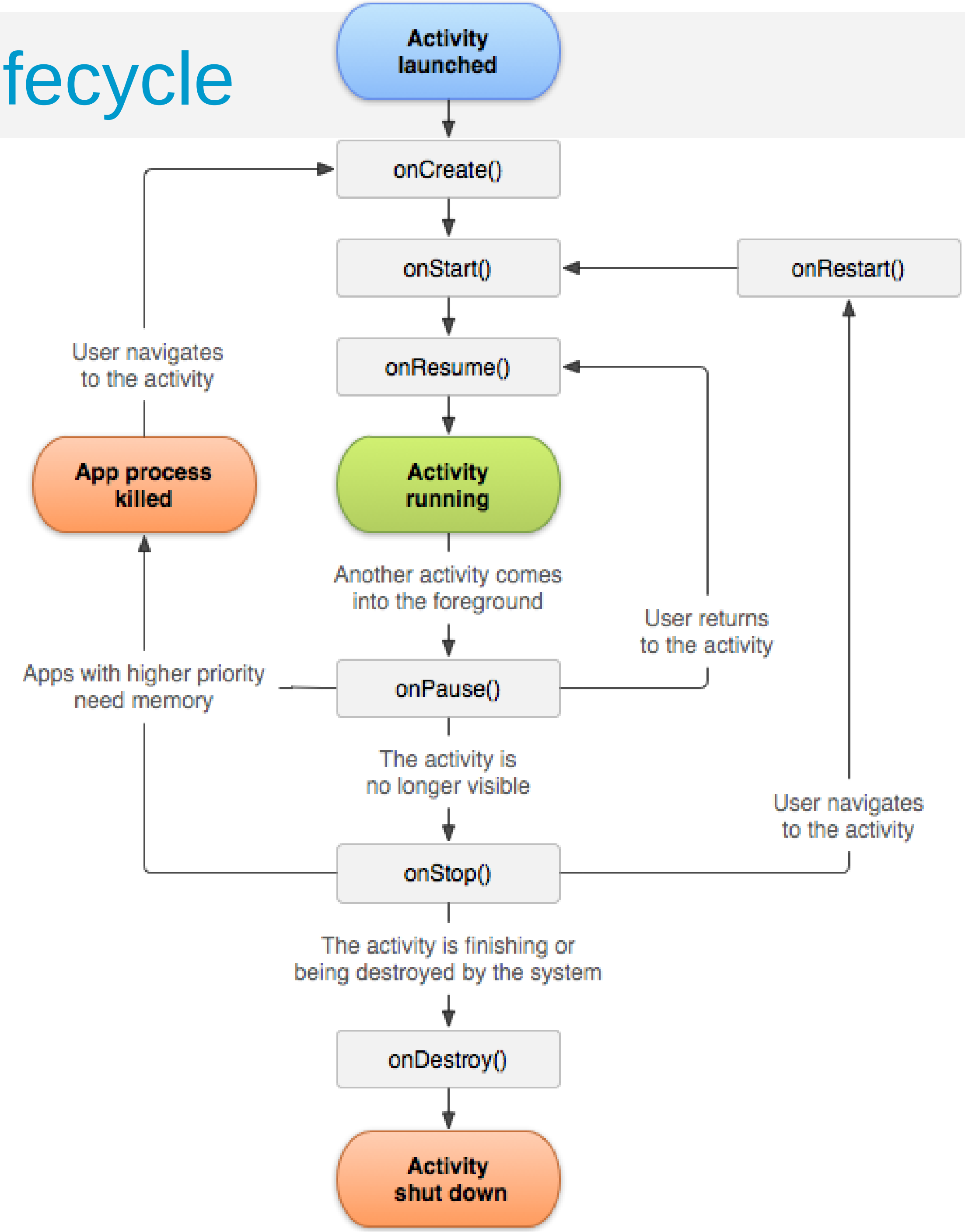
Called when the Activity is **created** the first time.

Called when the Activity is **partially visible**.

Called when the Activity is **no longer visible**.

Called when the Activity is **dismissed**.

Activity Lifecycle


Activity launched
onCreate()
onStart()
onRestart()
User navigates to the activity
onResume()
App process killed
Activity running
Another activity comes into the foreground
User returns to the activity
Apps with higher priority need memory
onPause()
The activity is no longer visible
User navigates to the activity
onStop()
The activity is finishing or being destroyed by the system
onDestroy()
Activity shut down