CS 528 Mobile and Ubiquitous Computing
Lecture 01b: Introduction to Android

Emmanuel Agu

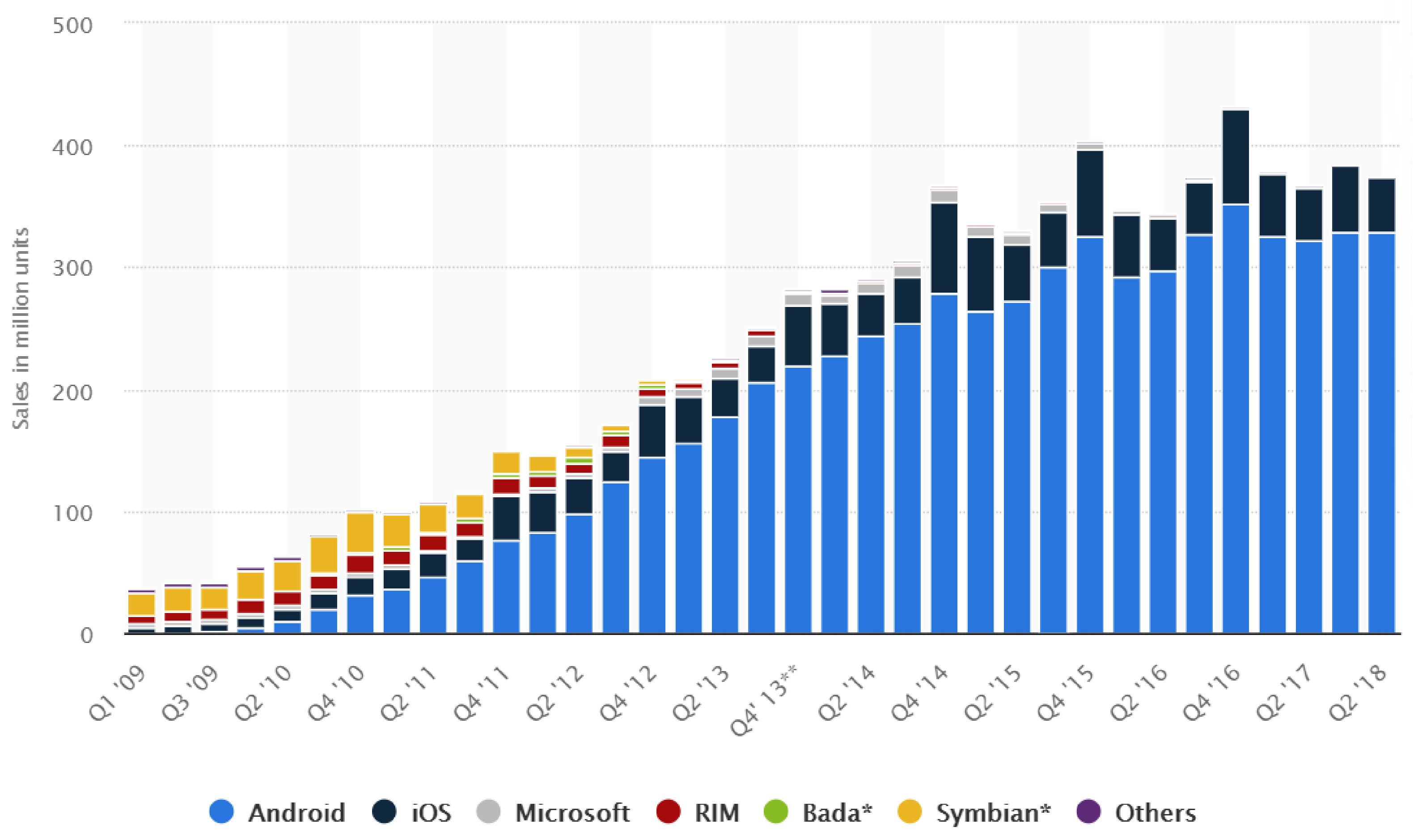
What is Android?

- Android is world's leading mobile operating system
  - Open source (https://source.android.com/setup/)

- **Google:**
  - Owns Android, maintains it, extends it
  - Distributes Android OS, developer tools, free to use
  - Runs Android app market

SmartPhone OS

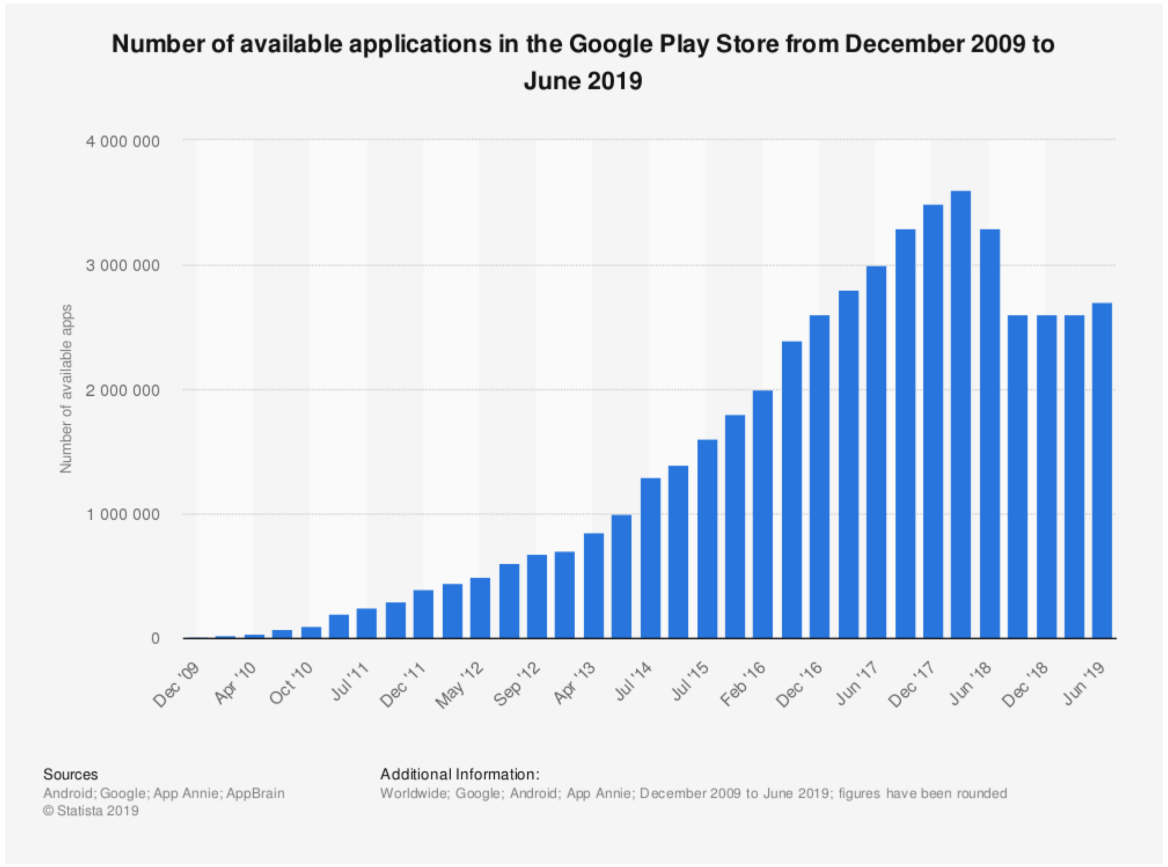
- Over 80% of all phones sold are smartphones
- Android share 86% worldwide



Source: Statista.com

Android Growth

- Over 2.5 billion Android devices, May 2019 (ref: Android police)
- 2.96 million apps on Google Play Android app market (ref: statista.com)
  - Games, organizers, banking, entertainment, etc



Android is Multi-Platform

Google Glass
(being redone)

In-car console

Smartwatch

Android runs on
all these devices

Smartphone

Tablet

Devices/Things
(e.g. Raspberry Pi)

Television

This Class: Focuses
Mostly on Smartphones!

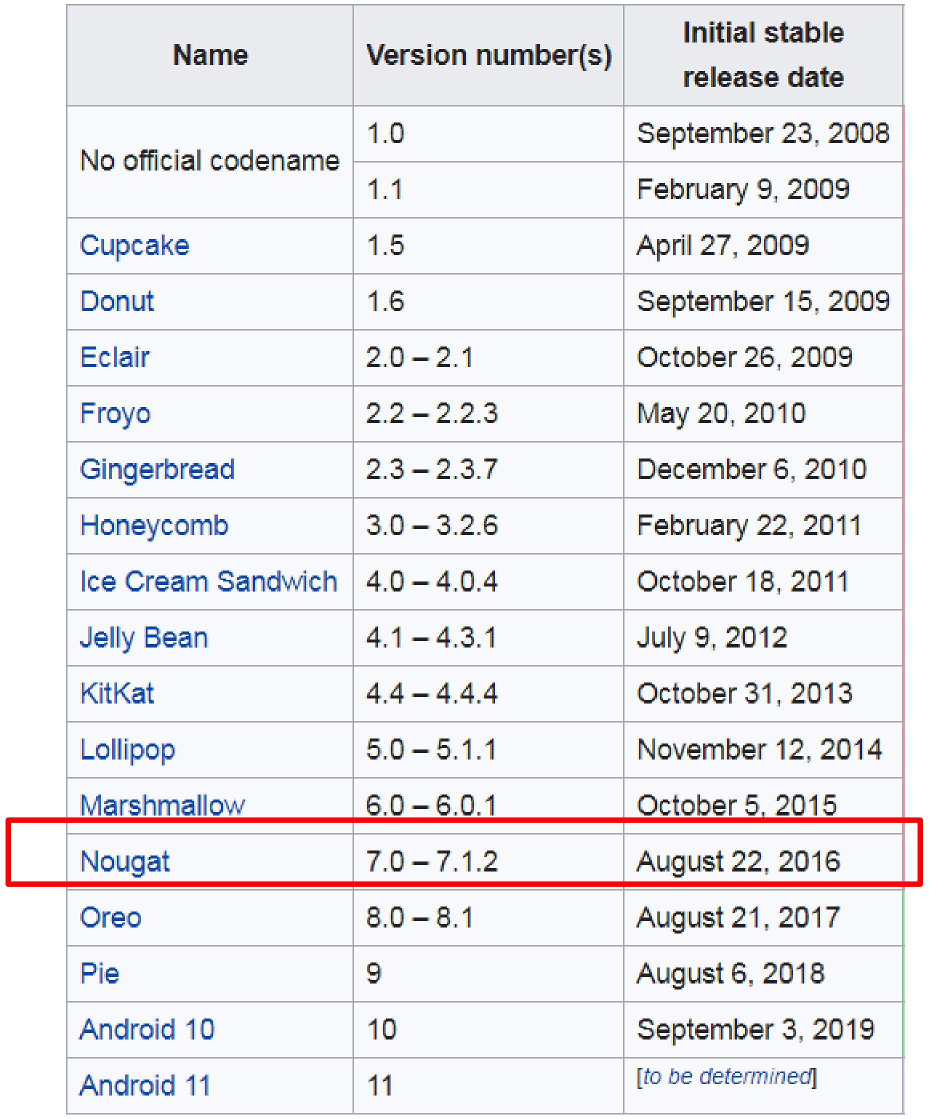
Why Android? Already has many Mobile Computing and Ubicomp Modules

- Android for Mobile programmable modules
  - Audio/video playback, taking pictures, database, location detection, maps

- Android for Ubicomp programmable modules
  - Sensors (temperature, humidity, light, etc), proximity
  - Face detection, activity recognition, place detection, speech recognition, speech-to-text, gesture detection, place type understanding, etc
  - Machine learning, deep learning

Android Versions

- Class will use Android 7 (“Nougat”)
- Officially released December 5, 2016
- Latest version is Android 10, released Sept 3, 2019
- Below is Android version distribution as at Sept 1, 2020



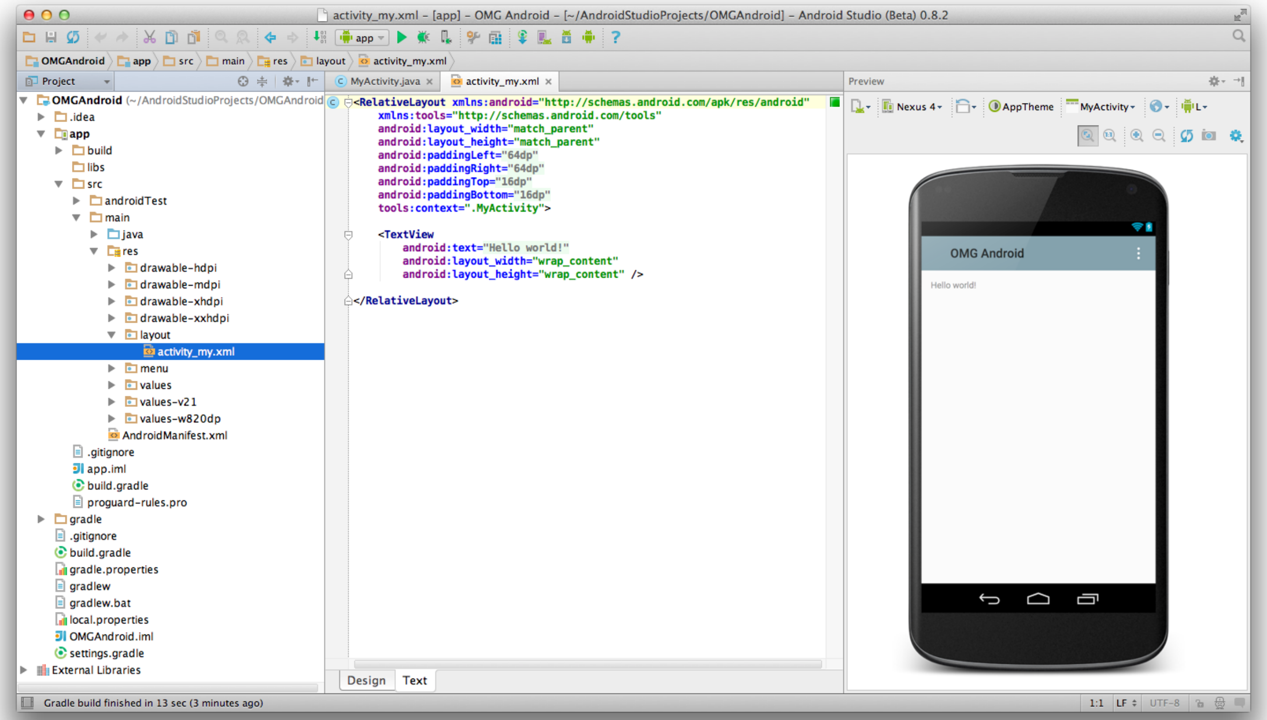
Name	Version number(s)	Initial stable release date
No official codename	1.0	September 23, 2008
	1.1	February 9, 2009
Cupcake	1.5	April 27, 2009
Donut	1.6	September 15, 2009
Eclair	2.0 – 2.1	October 26, 2009
Froyo	2.2 – 2.2.3	May 20, 2010
Gingerbread	2.3 – 2.3.7	December 6, 2010
Honeycomb	3.0 – 3.2.6	February 22, 2011
Ice Cream Sandwich	4.0 – 4.0.4	October 18, 2011
Jelly Bean	4.1 – 4.3.1	July 9, 2012
KitKat	4.4 – 4.4.4	October 31, 2013
Lollipop	5.0 – 5.1.1	November 12, 2014
Marshmallow	6.0 – 6.0.1	October 5, 2015
Nougat	7.0 – 7.1.2	August 22, 2016
Oreo	8.0 – 8.1	August 21, 2017
Pie	9	August 6, 2018
Android 10	10	September 3, 2019
Android 11	11	[to be determined]

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

Android Developer Environment

New Android Environment: Android Studio

- Old Android dev environment used **Eclipse + plugins**
- Google developed it’s own IDE called **Android Studio**
- Integrated development environment, cleaner interface, specifically for Android Development (e.g. drag and drop app design)
- In December 2014, Google announced it will stop supporting Eclipse IDE


activity_my.xml – [app] – OMG Android – [~/AndroidStudioProjects/OMGAndroid] – Android Studio (Beta) 0.8.2

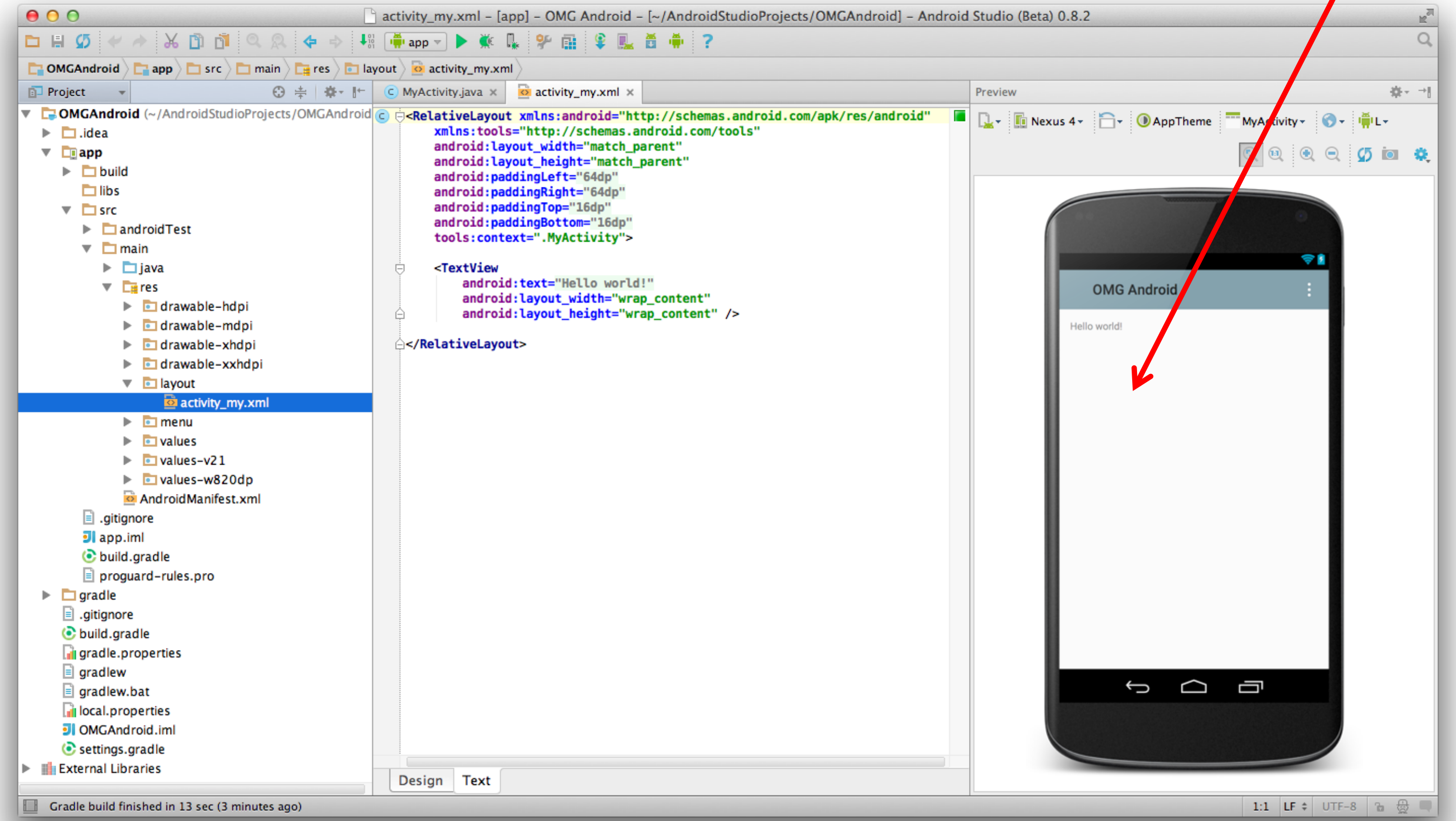
```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingLeft="64dp"
    android:paddingRight="64dp"
    android:paddingTop="16dp"
    android:paddingBottom="16dp"
    tools:context=".MyActivity">

    <TextView
        android:text="Hello world!"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

</RelativeLayout>
```

Gradle build finished in 13 sec (3 minutes ago)

Where to Run Android App

- Android app can run on:
  - Real phone (or device)
  - Emulator (software version of phone)

Emulated phone in Android Studio


activity_my.xml - [app] - OMG Android - [~/AndroidStudioProjects/OMGAndroid] - Android Studio (Beta) 0.8.2

```xml
<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"
    android:paddingLeft="64dp"
    android:paddingRight="64dp"
    android:paddingTop="16dp"
    android:paddingBottom="16dp"
    tools:context=".MyActivity">

    <TextView
        android:text="Hello world!"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

</RelativeLayout>
```

Gradle build finished in 13 sec (3 minutes ago)

Running Android App on Real Phone

- Need USB cord to copy app from development PC to phone

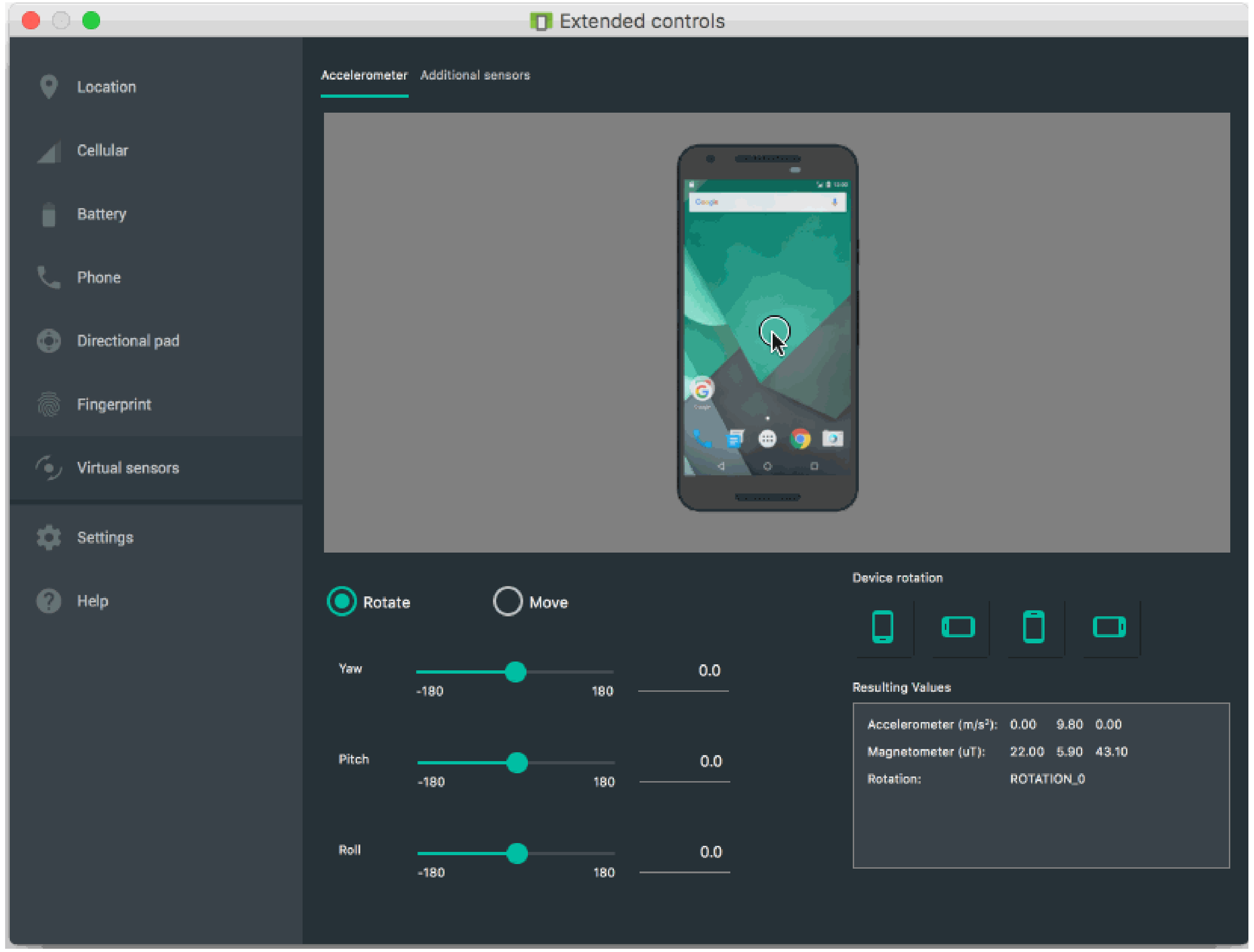
Emulator Pros and Cons (Vs Real Phone)

- Pros:
  - Conveniently test app on basic hardware by clicking in software
  - Easy to test app on many emulated devices (phones, tablets, TVs, etc), various screen sizes

- Cons:
  - Limited support, access to hardware, communications, sensors
  - E.g. GPS, camera, video recording, making/receiving phone calls, Bluetooth devices, USB devices, battery level, sensors, etc
  - Slower than real phone

New Support for Sensors

- Can now emulate **some** sensors (e.g. location, accelerometer), but still limited



Yaw -180 180 0.0

Pitch -180 180 0.0

Roll -180 180 0.0

Accelerometer (m/s²): 0.00 9.80 0.00
Magnetometer (uT): 22.00 5.90 43.10
Rotation: ROTATION_0

Demo: Android Studio

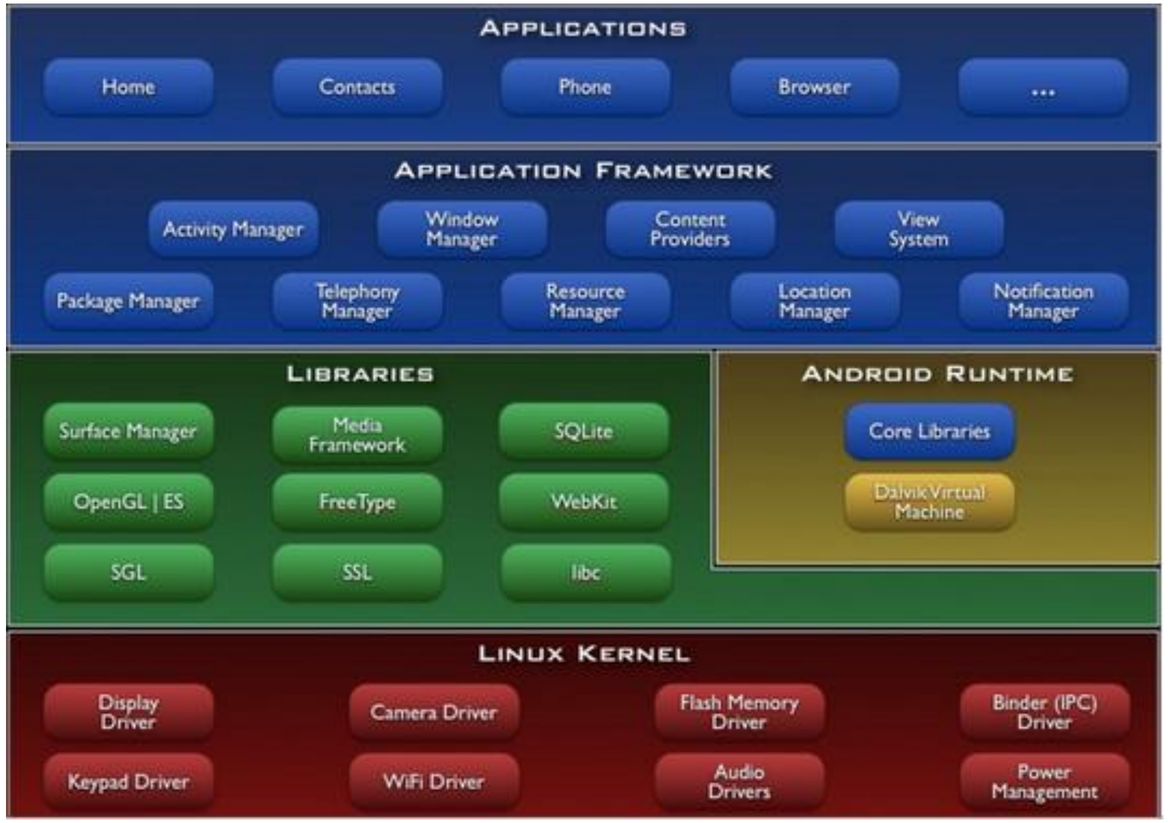
Android Software Framework

Android Functionality as Apps

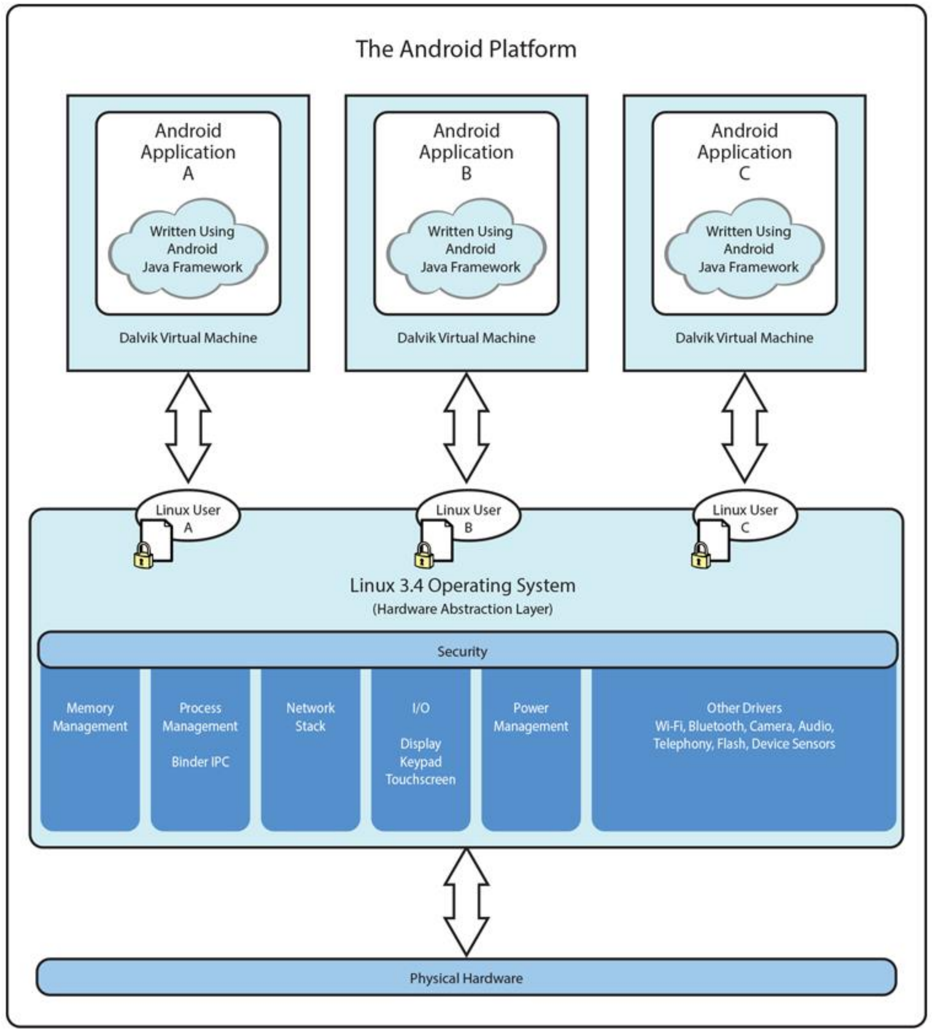
- Android functionality: collection of mini-applications (apps)
- Even phone “hardware” components dialer, keyboard, etc

Android Software Framework

- **OS:** Linux kernel, drivers
- **Apps:** programmed & UI in Java
- **Libraries:** OpenGL ES (graphics), SQLite (database), etc



Android Software Framework



- Each Android app runs in its own security sandbox (VM, minimizes complete system crashes)
- Android OS multi-user Linux system
- Each app is a different user (assigned unique Linux ID)
- Access control: only process with the app’s user ID can access its files

Ref: Introduction to Android Programming, Annuzzi, Darcey & Conder

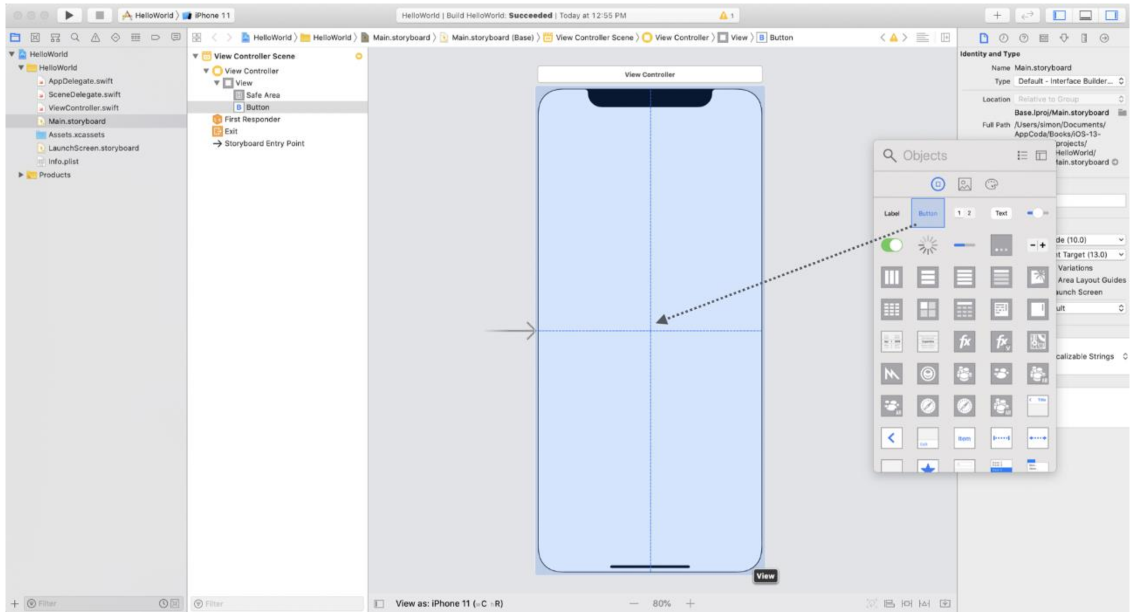
Android Programming Languages

- Two main languages to program Android
  1. Java-based (Native) programming + XML:
     - We will focus on that in this class
  2. Kotlin:
     - New alternative way, Higher level, easier?
     - We will give overview of Kotlin later in class
     - Google is encouraging developers to switch to kotlin

Other Mobile Development Frameworks

iOS App Development

- Download Xcode (iOS programming IDE)
- Free to program iOS apps
- But to publish app on app store, need to buy $99/yr membership
  - More regulated than Android
  - A human checks all iOS apps before publishing them
- iOS apps programmed in Swift language



Other Mobile Development Frameworks

- Lots of cross-platform frameworks
- Idea: write code in “some” language that gets compiled to Android or iOS

$$\begin{tikzcd}
 & \boxed{\text{Mobile Dev. Framework}} \arrow[dl] \arrow[dr] & \\
\text{Android} & & \text{iOS}
\end{tikzcd}$$

Xamarin: .NET Microsoft framework, code in C#

PhoneGap: Program mobile code in HTML, CSS

Corona SDK: Rapid game development

Other Mobile Development Frameworks

- Some framework just for UI development

Flutter

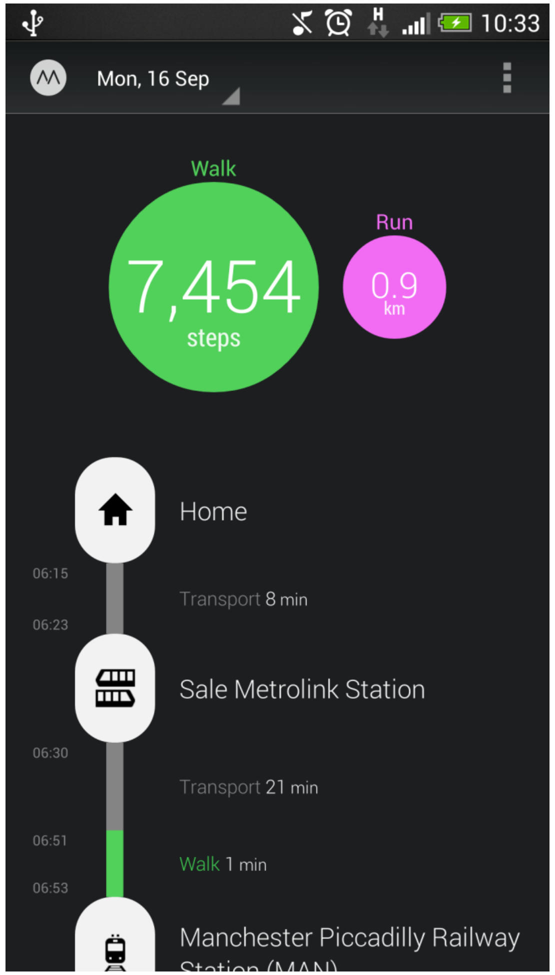
Ionic

React

Android Apps: Big Picture

UI Design using XML

- UI design code (XML) separate from the program (Java)
- Why? Can modify UI without changing Java program
- **Example:** Shapes, colors can be changed in XML file without changing Java program
- UI designed using either:
  - Drag-and drop graphical (WYSIWYG) tool or
  - Programming Extensible Markup Language (XML)
- **XML:** Markup language, both human-readable and machine-readable''



Android App Compilation

- Android Studio compiles code, data and resource files into **Android PacKage (filename.apk)**.
  - .apk is similar to .exe on Windows

- Apps download from Google Play, or copied to device as **filename.apk**

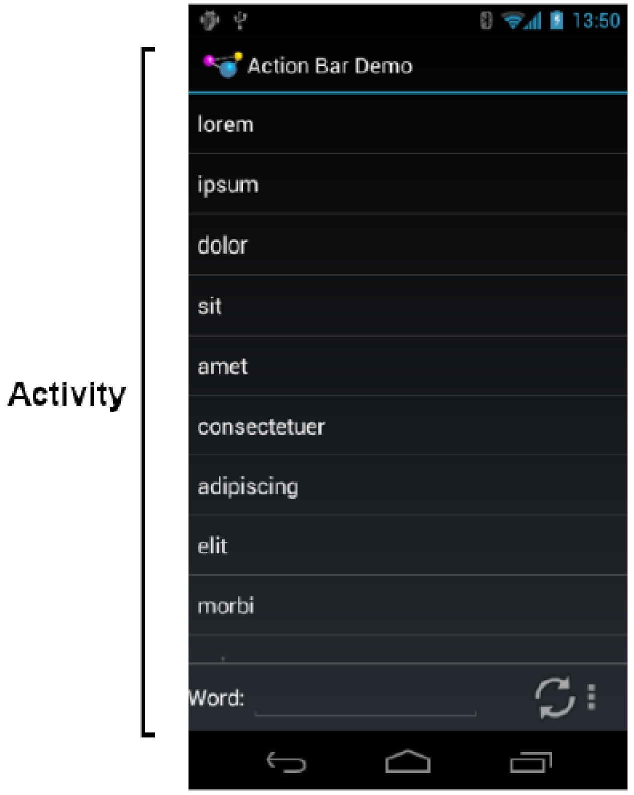
- Installation = installing **apk file**

Activities

- Activity? 1 Android screen or dialog box
- Apps
  - Have at least 1 activity that deals with UI
  - Entry point, similar to **main( )** in C
  - Typically have multiple activities (screens)

- Example: A camera app
  - **Activity 1:** to focus, take photo, launch activity 2
  - **Activity 2:** to view photo, save it

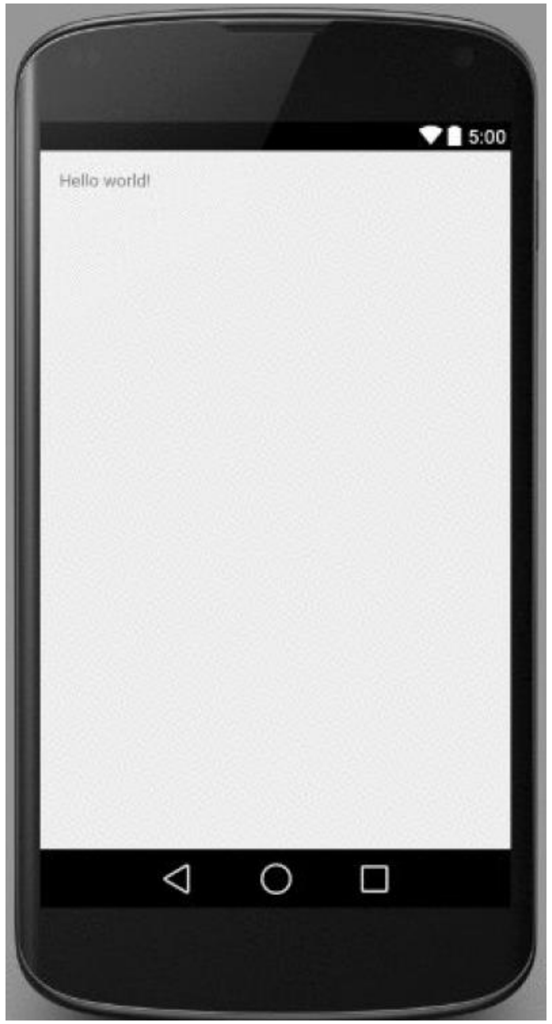
- Activities
  - independent of each other
  - E.g. Activity 1 can write data, read by activity 2
  - App Activities derived from Android's **Activity** class



Our First Android App

3 Files in “Hello World” Android Project

- **Activity_my.xml:** XML file specifying screen layout
- **MainActivity.Java:** Java code to define behavior, actions taken when button clicked (intelligence)
- **AndroidManifest.xml:**
  - Lists all screens, components of app
  - Analogous to a table of contents for a book
  - E.g. Hello world program has 1 screen, so AndroidManifest.xml has 1 item listed
  - App starts running here (like main( ) in C)
- **Note:** Android Studio creates these 3 files for you



Execution Order

Next: Samples of AndroidManifest.xml
Hello World program

Start in AndroidManifest.xml
Read list of activities (screens)
Start execution from Activity tagged Launcher

Create/execute activities (declared in java files)
E.g. MainActivity.Java

Format each activity using layout
In XML file (e.g. Activity_my.xml)

Inside “Hello World” AndroidManifest.xml

This file is written using xml namespace and tags and rules for android

Your package name

Android version

List of activities (screens) in your app

```
<?xml version="1.0"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
  package="com.commonsware.android.skeleton"
  android:versionCode="1"
  android:versionName="1.0">

  <application>
    <activity
      android:name="Now"
      android:label="Now">
      <intent-filter>
        <action android:name="android.intent.action.MAIN"/>

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

</manifest>
```

One activity (screen) designated LAUNCHER. The app starts running here

Execution Order

Start in AndroidManifest.xml
Read list of activities (screens)
Start execution from Activity
tagged Launcher

↓

Next → Create/execute activities
(declared in java files)
E.g. MainActivity.Java

↓

Format each activity using layout
In XML file (e.g. Activity_my.xml)

Example Activity Java file (E.g. MainActivity.java)

Package declaration →

Import needed classes →

My class inherits from Android activity class →

Initialize by calling onCreate( ) method of base Activity class →

```java
package com.commonsware.empublite;

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

public class EmPubLiteActivity extends Activity {
  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);
  }
}
```

Note: Android calls your Activity's onCreate method once it is created

Use screen layout (design) declared in file main.xml

Execution Order

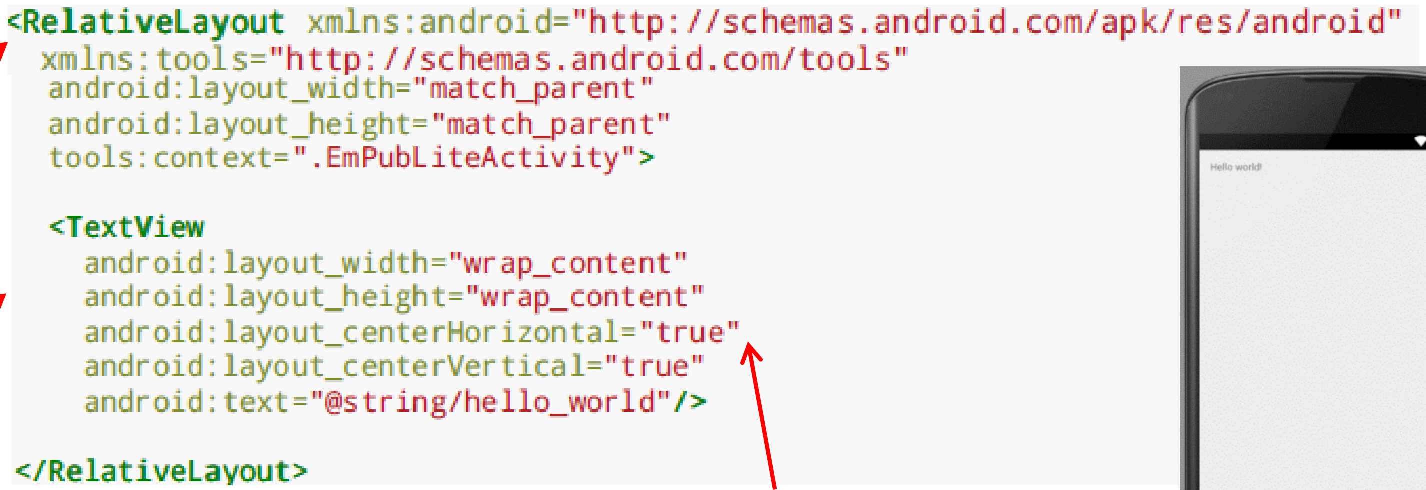
Start in AndroidManifest.xml
Read list of activities (screens)
Start execution from Activity tagged Launcher

Create/execute activities (declared in java files)
E.g. MainActivity.Java

Next → Format each activity using layout In XML file (e.g. Activity_my.xml)

Simple XML file Designing UI

- After choosing the layout, then widgets added to design UI
- XML Layout files consist of:
  - UI components (boxes) called **Views**
  - Different types of views. E.g
    - **TextView:** contains text,
    - **ImageView:** picture,
    - **WebView:** web page
  - **Views** arranged into layouts or **ViewGroups**

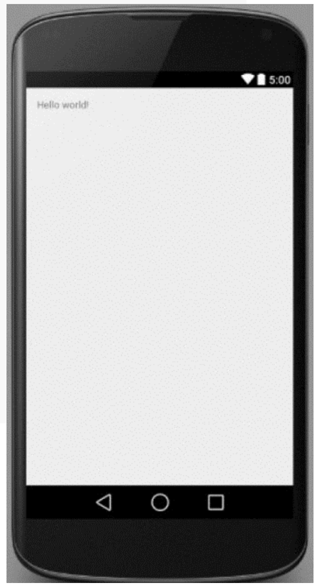


```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:tools="http://schemas.android.com/tools"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   tools:context=".EmPubLiteActivity">

   <TextView
     android:layout_width="wrap_content"
     android:layout_height="wrap_content"
     android:layout_centerHorizontal="true"
     android:layout_centerVertical="true"
     android:text="@string/hello_world"/>

</RelativeLayout>
```

Declare Layout

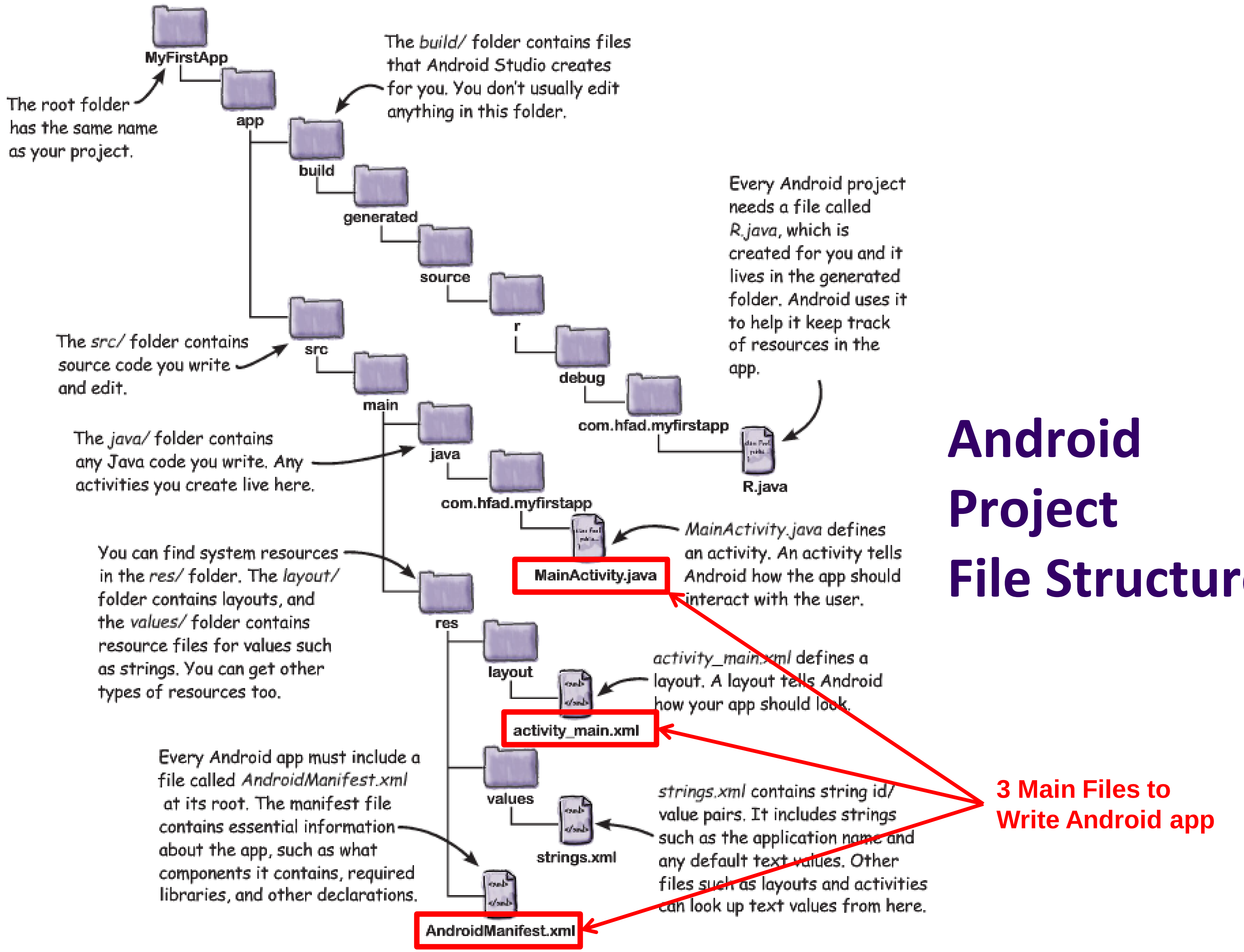
Add widgets

Widget properties
(e.g. center contents
horizontally and vertically)



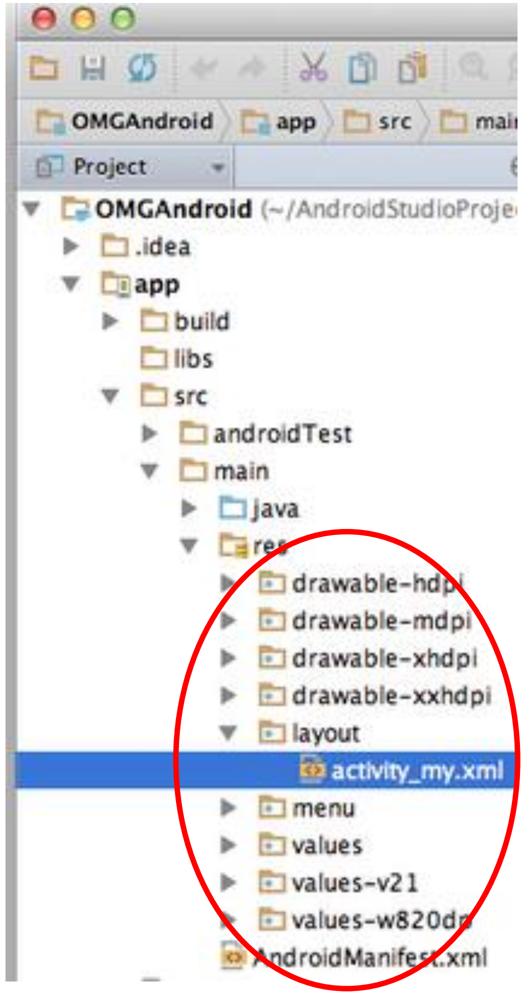
Android Files

Android Project File Structure


MyFirstApp
The root folder has the same name as your project.
app
The build/ folder contains files that Android Studio creates for you. You don't usually edit anything in this folder.
build
generated
source
r
debug
com.hfad.myfirstapp
R.java
Every Android project needs a file called R.java, which is created for you and it lives in the generated folder. Android uses it to help it keep track of resources in the app.
The src/ folder contains source code you write and edit.
src
main
The java/ folder contains any Java code you write. Any activities you create live here.
java
com.hfad.myfirstapp
MainActivity.java
MainActivity.java defines an activity. An activity tells Android how the app should interact with the user.
You can find system resources in the res/ folder. The layout/ folder contains layouts, and the values/ folder contains resource files for values such as strings. You can get other types of resources too.
res
layout
activity_main.xml
activity_main.xml defines a layout. A layout tells Android how your app should look.
values
strings.xml
strings.xml contains string id/value pairs. It includes strings such as the application name and any default text values. Other files such as layouts and activities can look up text values from here.
Every Android app must include a file called AndroidManifest.xml at its root. The manifest file contains essential information about the app, such as what components it contains, required libraries, and other declarations.
AndroidManifest.xml
3 Main Files to Write Android app

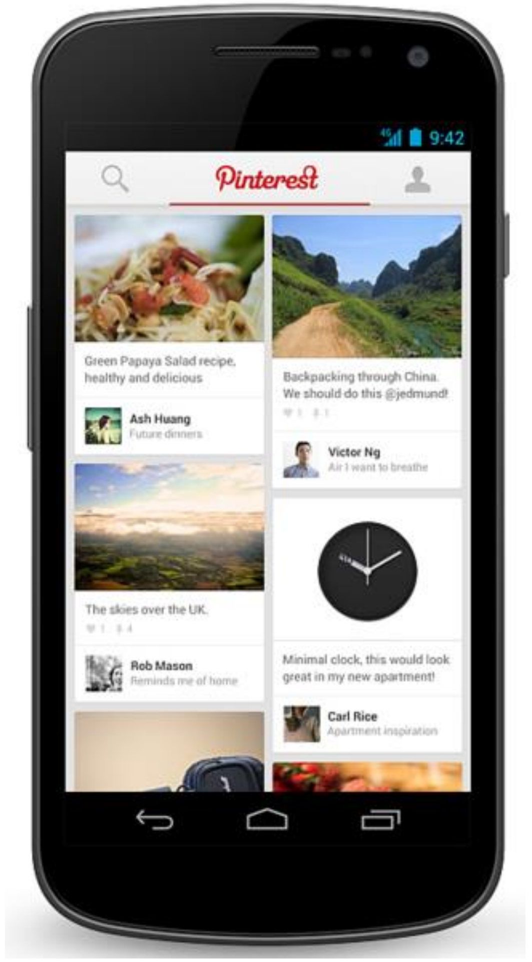
Files in an Android Project

- **res/** (resources) folder contains static resources you can embed in Android screen (e.g. pictures, string declarations, etc)
- **res/menu/:** XML files for menu specs
- **res/drawable-xyz/:** images (PNG, JPEG, etc) at various resolutions
- **res/raw:** general-purpose files (e.g. audio clips, mpeg, video files, CSV files
- **res/values/:** strings, dimensions, etc



Concrete Example: Files in an Android Project

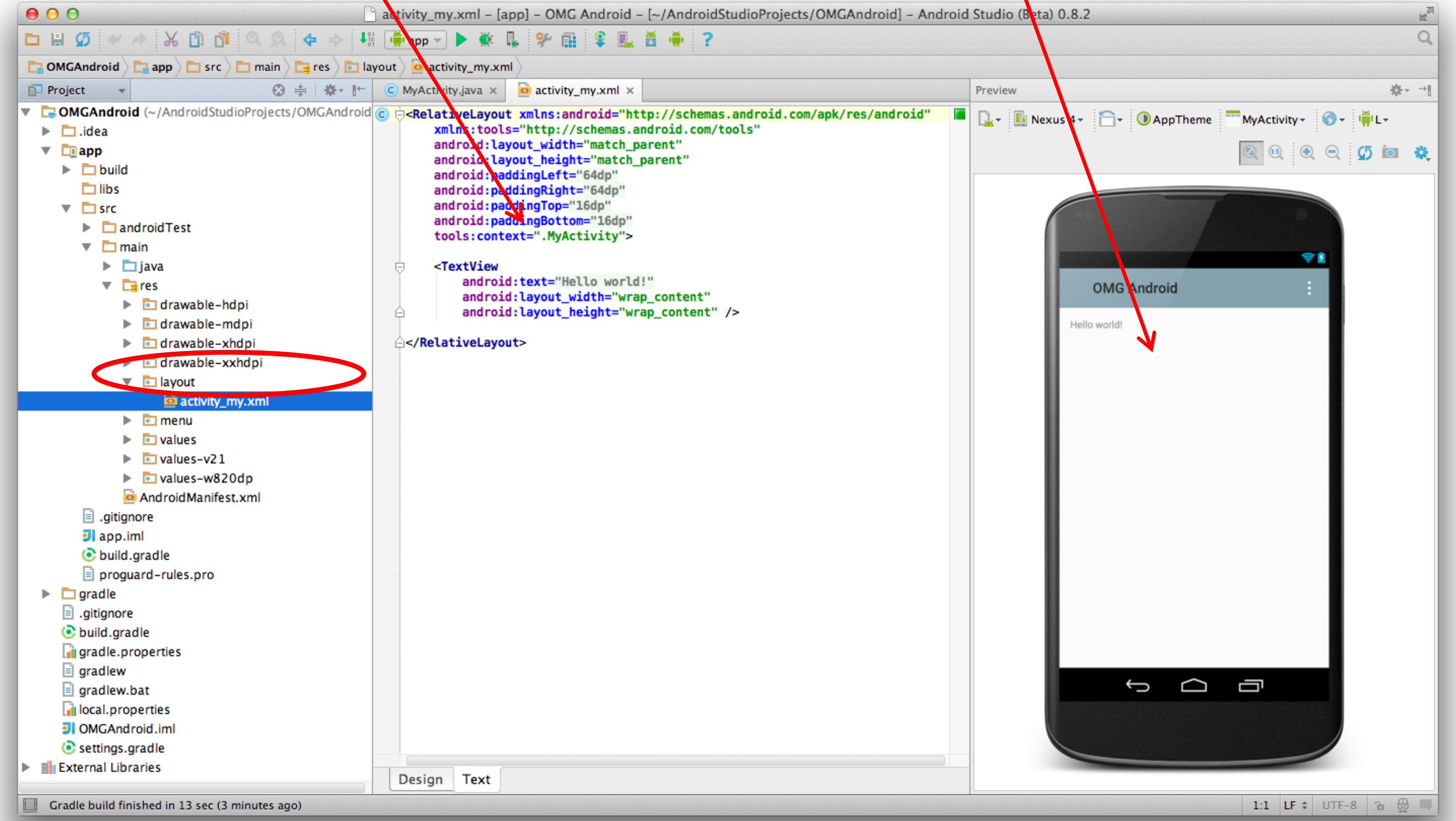
- **res/layout:** layout, dimensions (width, height) of screen cells are specified in XML file here
- **res/drawable-xyz/:** The images stored in jpg or other format here
- **java/:** App’s response when user clicks on a selection is specified in java file here
- **AndroidManifext.XML:** Contains app name (Pinterest), list of app screens, etc


9:42
Pinterest
Green Papaya Salad recipe, healthy and delicious
Ash Huang
Future dinners
Backpacking through China. We should do this @jedmund!
Victor Ng
Air I want to breathe
The skies over the UK.
Rob Mason
Reminds me of home
Minimal clock, this would look great in my new apartment!
Carl Rice
Apartment inspiration

Editting in Android Studio

Editting Android

- Can edit apps in:
  - **Text View:** edit XML directly
  - **Design View:** or drag and drop widgets unto emulated phone


activity_my.xml – [app] – OMG Android – [~/AndroidStudioProjects/OMGAndroid] – Android Studio (Beta) 0.8.2
OMGAndroid › app › src › main › res › layout › activity_my.xml
Project
MyActivity.java
activity_my.xml
Preview
Nexus 4
AppTheme
MyActivity
L
OMG Android
Hello world!
Design
Text
Gradle build finished in 13 sec (3 minutes ago)

```
OMGAndroid (~/AndroidStudioProjects/OMGAndroid)
  .idea
  app
    build
    libs
    src
      androidTest
      main
        java
        res
          drawable-hdpi
          drawable-mdpi
          drawable-xhdpi
          drawable-xxhdpi
          layout
            activity_my.xml
          menu
          values
          values-v21
          values-w820dp
        AndroidManifest.xml
    .gitignore
    app.iml
    build.gradle
    proguard-rules.pro
  gradle
  .gitignore
  build.gradle
  gradle.properties
  gradlew
  gradlew.bat
  local.properties
  OMGAndroid.iml
  settings.gradle
External Libraries
```

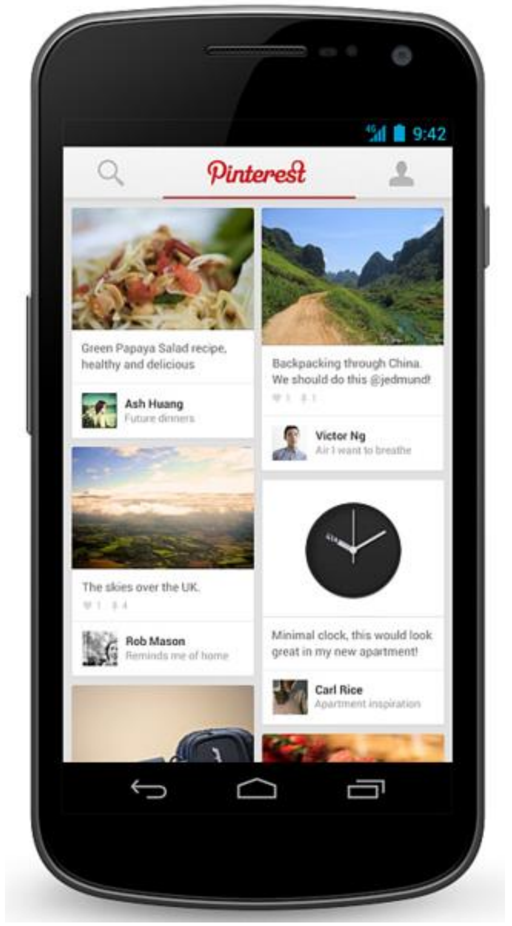
```xml
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingLeft="64dp"
    android:paddingRight="64dp"
    android:paddingTop="16dp"
    android:paddingBottom="16dp"
    tools:context=".MyActivity">

    <TextView
        android:text="Hello world!"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

</RelativeLayout>
```

Android UI Design in XML

Recall: Files Hello World Android Project

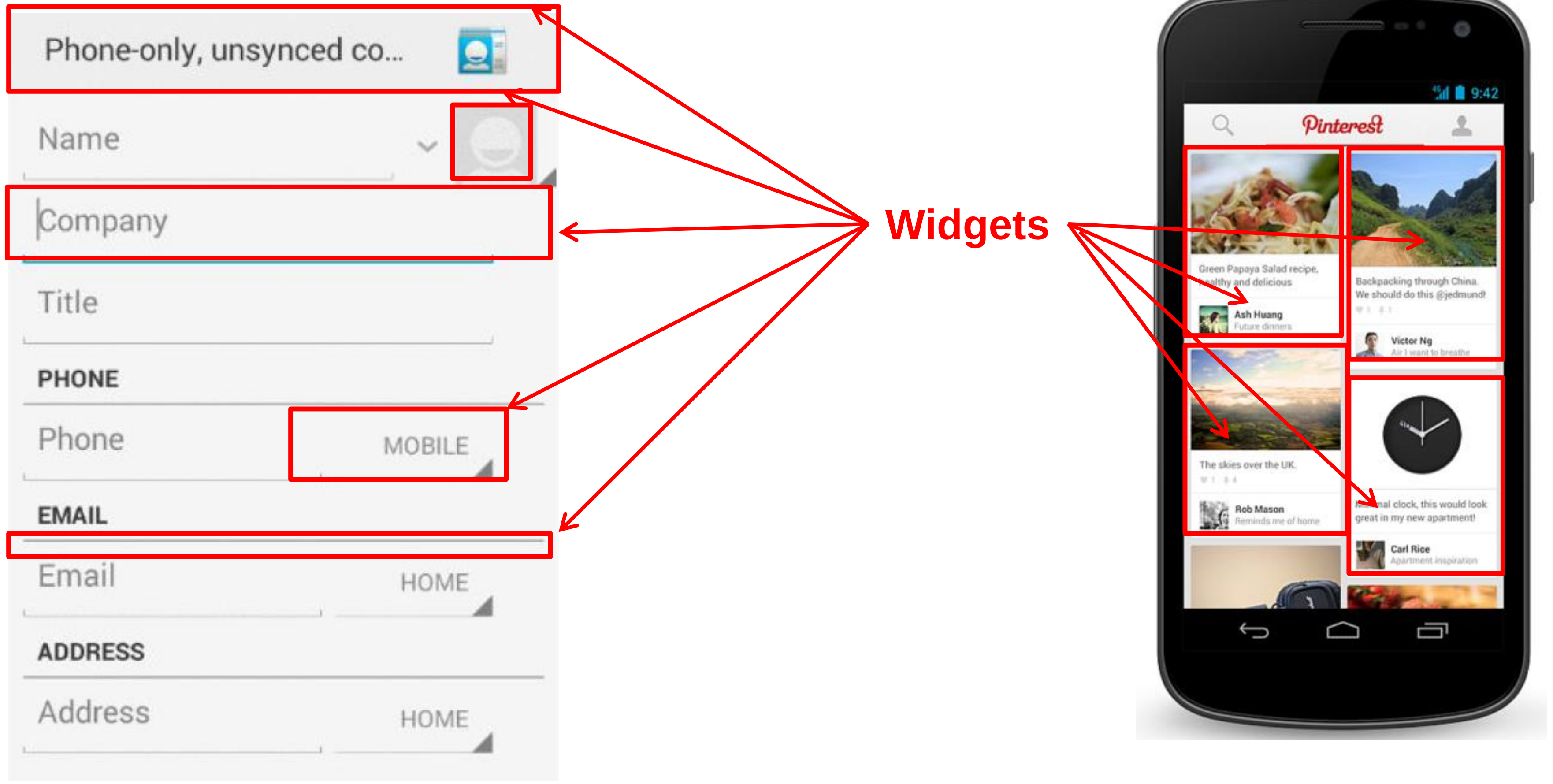
XML file used to design Android UI

- 3 Files:
  - **Activity_main.xml:** XML file specifying screen layout
  - **MainActivity.Java:** Java code to define behavior, actions taken when button clicked (intelligence)
  - **AndroidManifest.xml:**
    - Lists all app components and screens
    - Like a table of contents for a book
    - E.g. Hello world program has 1 screen, so AndroidManifest.xml has 1 item listed
    - App starts running here (a bit like main( ) in C), launching activity with a tag “LAUNCHER”


9:42
Pinterest
Green Papaya Salad recipe, healthy and delicious
Ash Huang
Backpacking through China. We should do this @jedmund!
Victor Ng
The skies over the UK.
Rob Mason
Minimal clock, this would look great in my new apartment!
Carl Rice

Widgets

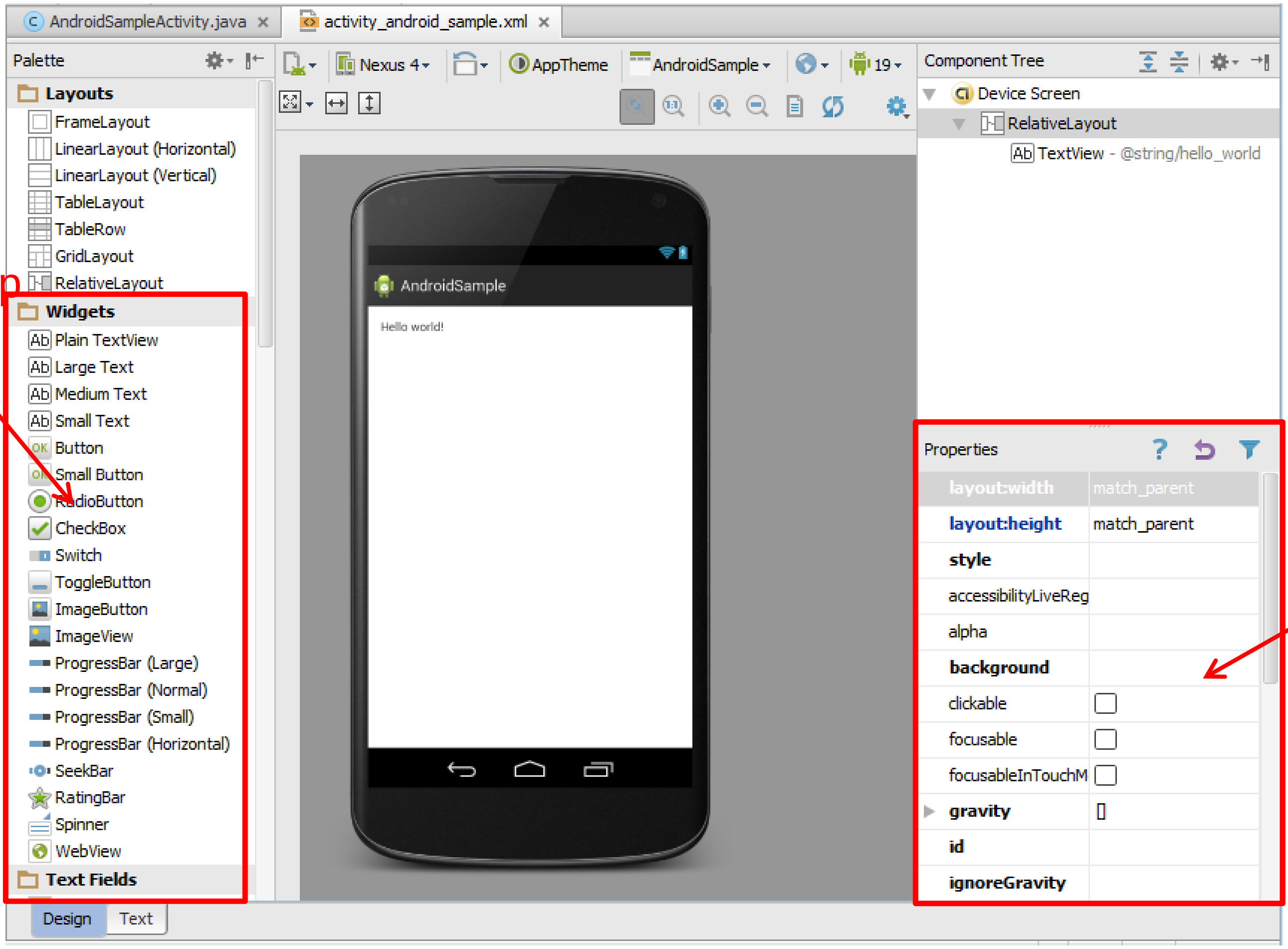
- ***Android UI design involves arranging widgets on a screen***
- **Widgets?** Rectangles containing texts, image, etc
- **Screen design:** Pick widgets, specify attributes (dimensions, margins, etc)



Design Option 1: Drag and Drop Widgets

- Drag and drop widgets in Android Studio Design View
- Edit widget properties (e.g. height, width, color, etc)

Drag and drop button or any other widget or view

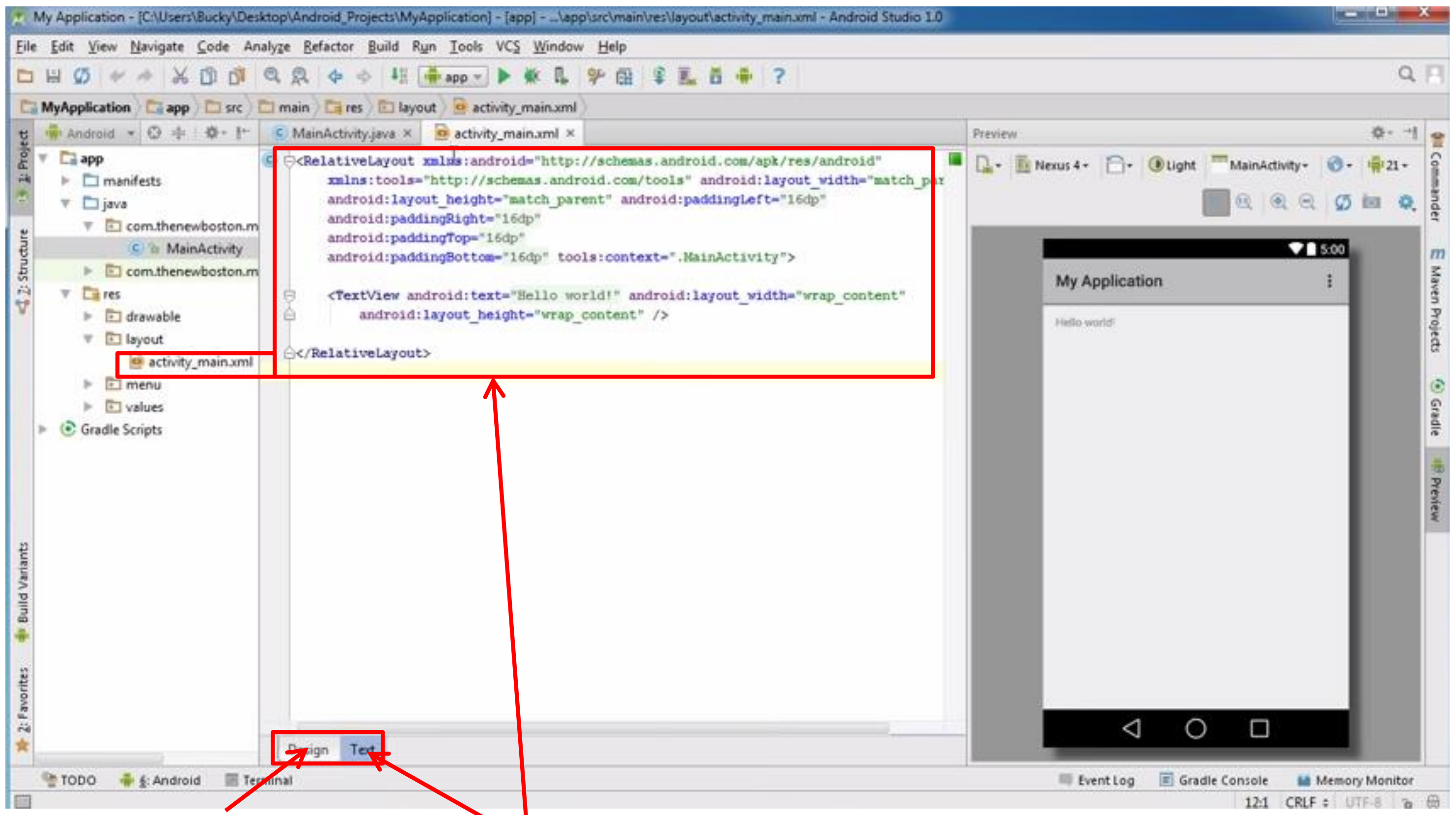


layout:width	match_parent
layout:height	match_parent
style	
accessibilityLiveReg	
alpha	
background	
clickable	☐
focusable	☐
focusableInTouchM	☐
gravity	[]
id	
ignoreGravity	

Edit widget properties

Design Option 2: Edit XML Directly

- **Text view:** Directly edit XML file defining screen (activity_main.xml)
- **Note:** dragging and dropping widgets in design view auto-generates corresponding XML in Text view


My Application - [C:\Users\Bucky\Desktop\Android_Projects\MyApplication] - [app] - ...\app\src\main\res\layout\activity_main.xml - Android Studio 1.0
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
MyApplication app src main res layout activity_main.xml
MainActivity.java activity_main.xml
app manifests java com.thenewboston.m MainActivity com.thenewboston.m res drawable layout activity_main.xml menu values Gradle Scripts
Preview
Nexus 4 Light MainActivity 21
My Application
Hello world!
Design Text
TODO 6: Android Terminal Event Log Gradle Console Memory Monitor

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools" android:layout_width="match_par
    android:layout_height="match_parent" android:paddingLeft="16dp"
    android:paddingRight="16dp"
    android:paddingTop="16dp"
    android:paddingBottom="16dp" tools:context=".MainActivity">

    <TextView android:text="Hello world!" android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

</RelativeLayout>
```

Drag and drop widget **Edit XML**

HW0: Android Setup/Getting Going

HW0: Tutorials from YouTube Android Development Tutorials 1-8 by Bucky Roberts

- **Tutorials 1 & 2 (Optional):** Installing Java, Android Studio on your own machine
  - **Tutorial 1:** Install Java (Android studio needs this at least ver. 1.8)
  - **Tutorial 2:** Install Android Studio

- **Tutorial 3:** Setting up your project
  - How to set up a new Android Project, add new Activity (App screen)

- **Tutorial 4:** Running a Simple App
  - How to select, run app on a virtual device (AVD)

- **Tutorial 5:** Tour of Android Studio Interface
  - Intro to Android Studio menus, toolbars and Drag-and-drop widget palette

References

- Android App Development for Beginners videos by Bucky Roberts (thenewboston)
- Ask A Dev, Android Wear: What Developers Need to Know, https://www.youtube.com/watch?v=zTS2NZpLyQg
- Ask A Dev, Mobile Minute: What to (Android) Wear, https://www.youtube.com/watch?v=n5Yjzn3b_aQ
- Busy Coder's guide to Android version 4.4
- CS 65/165 slides, Dartmouth College, Spring 2014
- CS 371M slides, U of Texas Austin, Spring 2014