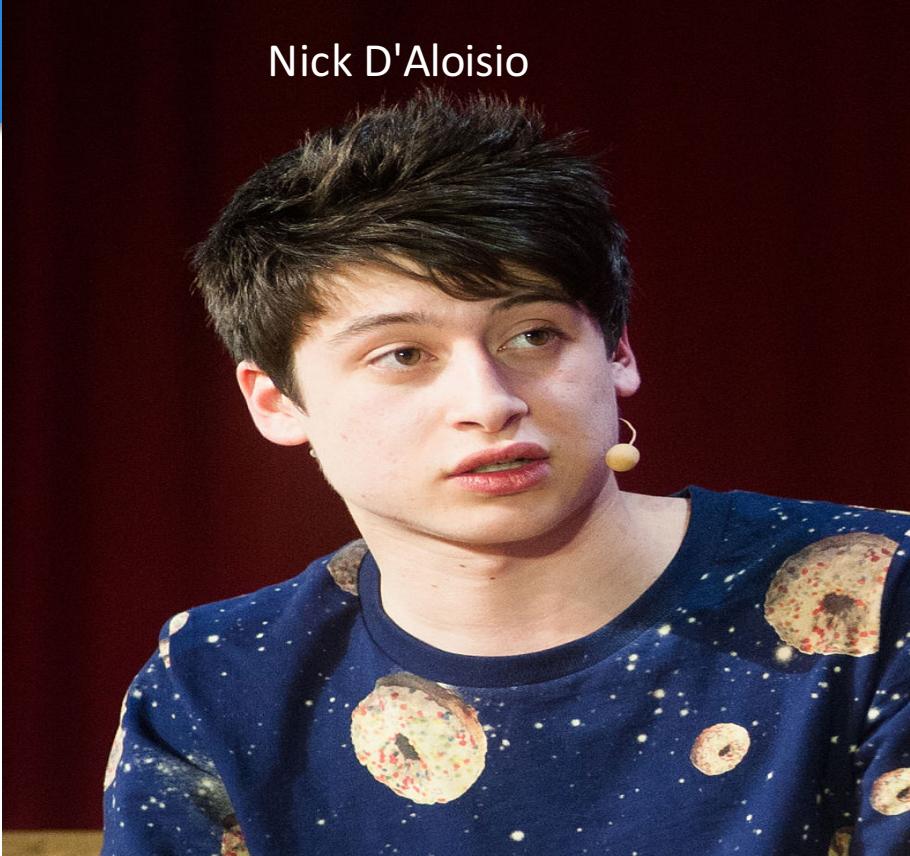


Introduction

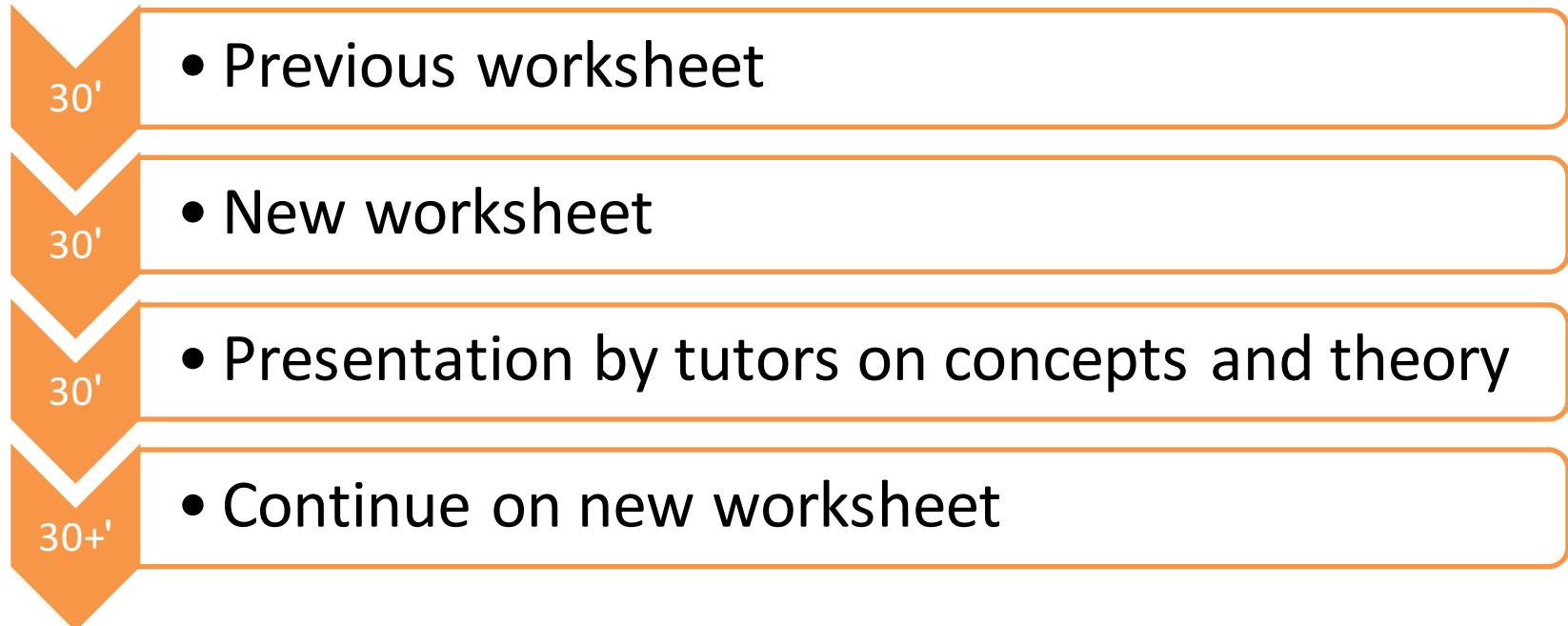
Nick D'Aloisio



- Born 1995
- Created app called **Trimit** in 2011
- Sold to Yahoo for \$30 million US dollars in 2013

http://en.wikipedia.org/wiki/Nick_D'Aloisio

Weekly Lab Format

- 
- Previous worksheet
30'
 - New worksheet
30'
 - Presentation by tutors on concepts and theory
30'
 - Continue on new worksheet
30+'

Contents

- Module structure
- Why mobile apps?
- First bite on Android
- Tools and resources
- Setting up IDE
- Hello, Android!
- Android anatomy
- Coding the interface
- Coding the behaviour
- Create-modify-reuse

Module structure

Requirements

- Existing knowledge and programming experience

Outcomes

- Demonstrate familiarity with the **Java** Programming language and the Android Studio.
- **Design** applications suitable for Android devices.
- Use the Android **software development kit** and emulator to develop applications for the Android platform.
- Make use of the main modes of **interaction** available on a smartphone platform.

Module structure

1. Android Studio
2. The Java language
3. XML and gradle
4. Simple views and controls
5. Composit views
6. Data persistence
7. Testing
8. Wearables
9. Graphics and animation
10. Location services and Maps
11. Multimedia

Module structure – assessment

- Assignment sheet on Moodle!
- Also at
<https://github.com/covcom/387COM/tree/master/assignment>
- Deadline: 18 December 2015

Why mobile apps?

A mobile app is a computer program designed to run on smartphones, tablet computers and other mobile devices.



Why mobile apps?

Transformative Devices

- Ubiquitous and compact
- long battery life
- Environment-aware
 - location (coarse/fine), angle, acceleration
- Multi-channel communication
 - cellular, Wi-Fi, Bluetooth, voice, data

Why mobile apps?

Reaching Customers

- Disruptive technology
- Personalized devices
- Always online
- Packed with advanced hardware
 - Immediacy
 - Short bursts
 - Not tied to ‘business hours’

Why mobile apps?

Changing Business Processes

- Processes fit current technology
- As technology changes, processes can be radically transformed
- Ubiquitous personal technology
 - Always on connectivity
 - BYOD
 - Advanced hardware (sensors, etc.)

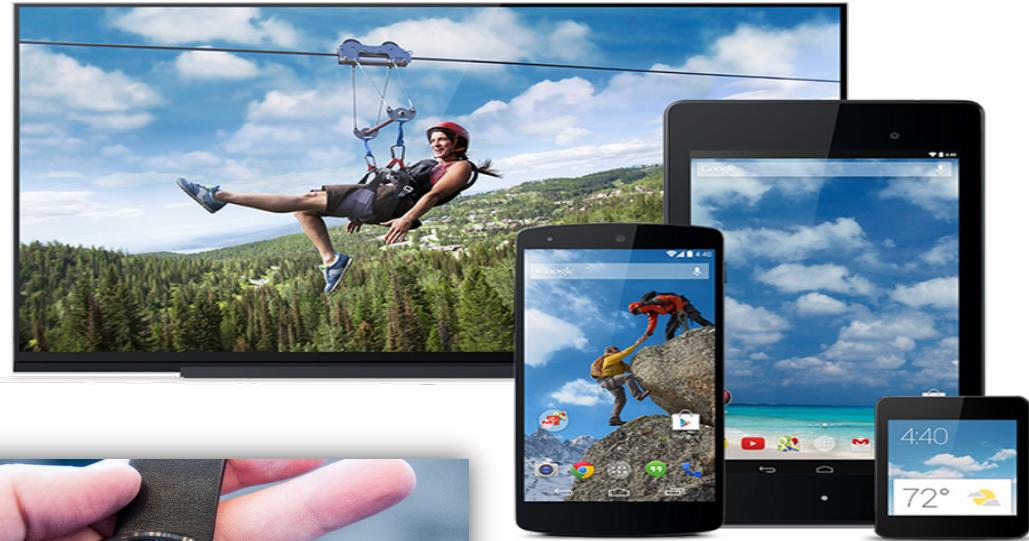
Why mobile apps?

Making Money

- App Store
 - distribution, returns and payment
- Monetisation
 - one-off fee, in-app advertising (view/click), subscription, in-app purchase

First bite on Android

- Complete
- Open
- Free



First bite on Android

Android is:

- A free, open-source operating system for embedded devices
- An open-source development platform for creating applications
- Devices, particularly mobile phones, that run the Android operating system and the applications created for it

“The first truly open and comprehensive platform for mobile devices. It includes an operating system, user-interface and applications – all of the software to run a mobile phone but without the proprietary obstacles that have hindered mobile innovation.”

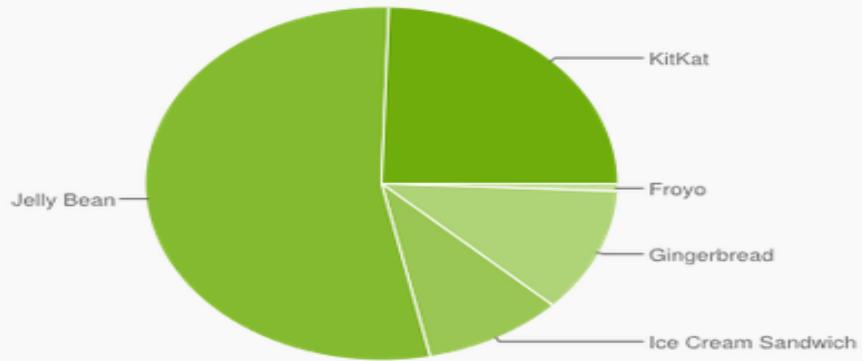
– Andy Rubin, Where’s My Gphone?

First bite on Android



First bite on Android

Version	Codename	API	Distribution
2.2	Froyo	8	0.7%
2.3.3 - 2.3.7	Gingerbread	10	11.4%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	9.6%
4.1.x	Jelly Bean	16	25.1%
4.2.x		17	20.7%
4.3		18	8.0%
4.4	KitKat	19	24.5%



*Data collected during a 7-day period ending on September 9, 2014.
Any versions with less than 0.1% distribution are not shown.*

Tools and resources

- Apps can be broadly classified into four distinct categories: native apps, generic mobile apps, dedicated web apps, and hybrid apps
- Native applications are usually developed using higher level programming languages, such as Java for Android, Objective-C for iOS

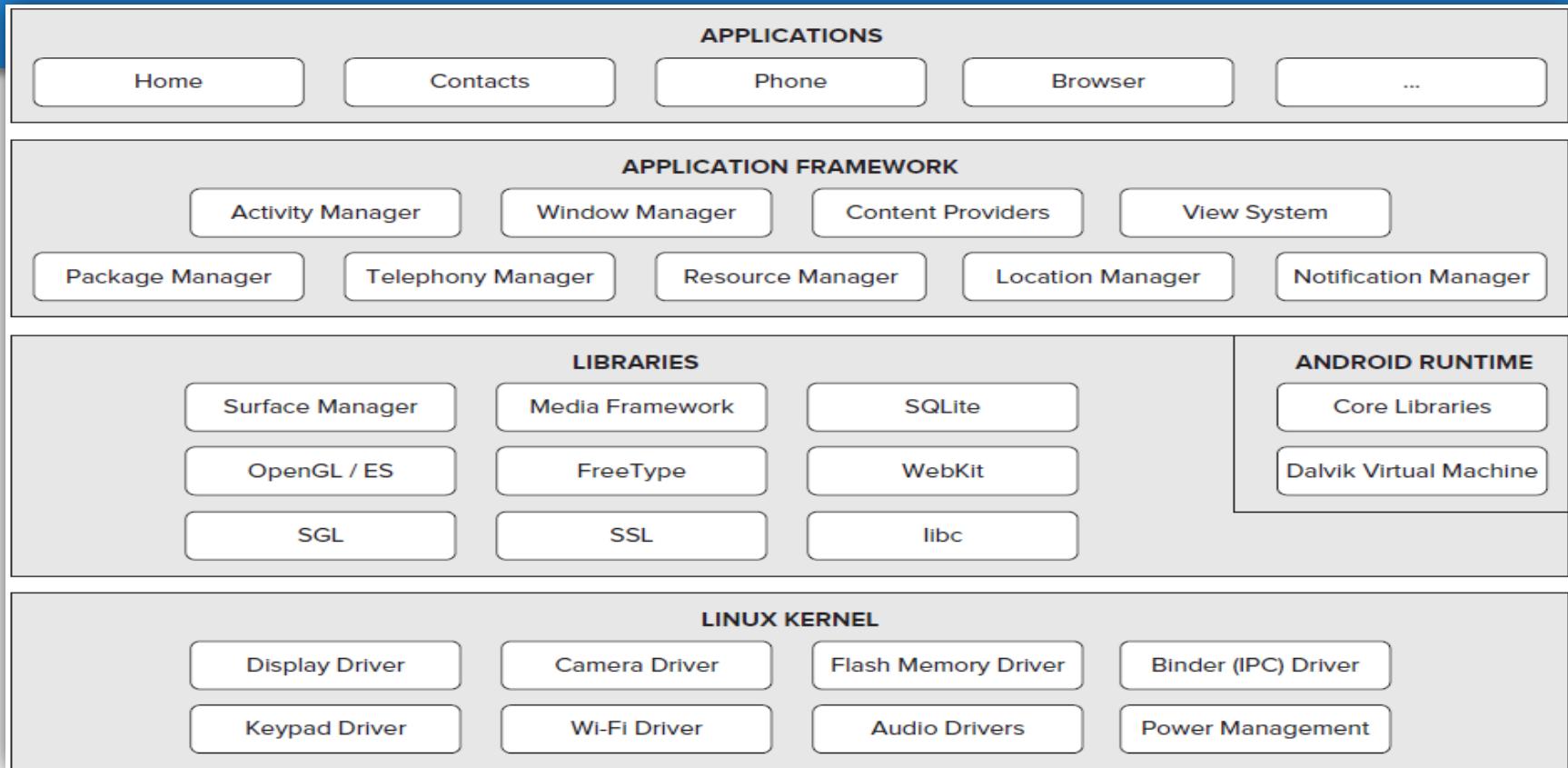


xamarin



Phone**Gap**

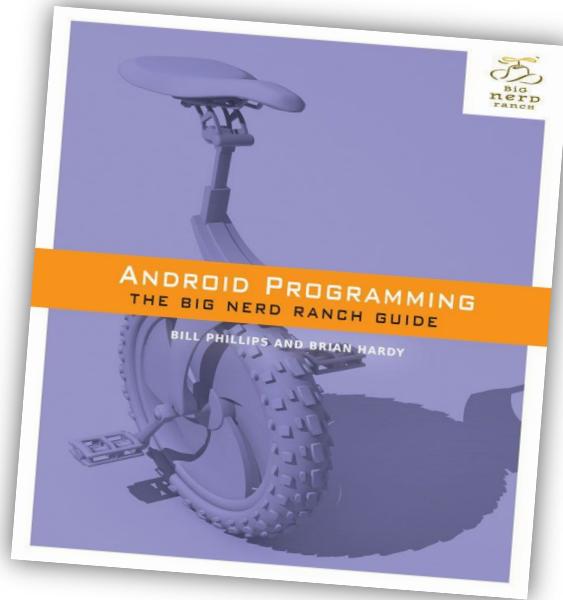
Tools and resources



Tools and resources

- Android Studio
- Android SDK Tools
- Android Platform-tools
- A version of the Android platform
- A version of the Android system image for the emulator

Tools and resources



Tools and resources

Online resources:

- Stack Overflow
 - www.stackoverflow.com
- Google Android Training
 - <http://developer.android.com/training/index.html>
- Android Discuss
 - <http://groups.google.com/group/android-discuss>

Setting up IDE

Step 1: install Java

- ✗ JRE (Java Runtime Environment)
 - Libraries
 - Java Virtual Machine
 - Components to run applets and java applications
- ✓ JDK (Java Development Toolkit)
 - Contains complete JRE
 - Compilers
 - Debuggers

The screenshot shows the Stack Overflow homepage. At the top, there's a navigation bar with links for Questions, Tags, Users, Badges, and Unans. Below the navigation bar, a banner states: "Stack Overflow is a question and answer site for professional and enthusiast programmers. It's 100% registration required." A search bar is located at the top right. A large, bold, black header at the bottom asks, "What is the difference between JDK and JRE?"

The screenshot shows the Java website. At the top, there's a red header with the Java logo, a search bar, and links for Download and Help. Below the header, a section titled "Verify Java and Find Out-of-Date Versions" contains instructions: "Check to ensure that you have the recommended version of Java installed on your Windows computer and identify any versions that are out of date and should be uninstalled." On the left, there's a "Help Resources" sidebar with links like "What is Java?", "Remove Older Versions", and "Disable Java". At the bottom right, there's a red button labeled "Agree and Continue".

Setting up IDE

Step 2: install Android Studio

The screenshot shows the 'Tools' section of the Android Developers website. On the left, a sidebar menu under 'Download' includes links for 'Installing the SDK', 'Adding SDK Packages', 'Android Studio', 'Workflow', 'Tools Help', 'Build System', 'Support Library', 'Revisions', 'NDK', 'ADK', and 'Eclipse with ADT'. The main content area features the Android Studio logo and a list of included components: 'Android Studio IDE', 'Android SDK tools', 'Android 5.0 (Lollipop) Platform', and 'Android 5.0 emulator system image with Google APIs'. A large green button says 'Download Android Studio for Mac'. To the right, there's a section titled 'The official Android IDE' showing a laptop displaying the Android Studio interface with an emulator running a login screen.

Setting up IDE

Step 2: (alternative): IntelliJ

1. Download and install the Android SDK
2. Configure Android platforms
3. Download and install IntelliJ IDEA

<https://confluence.jetbrains.com/display/IntelliJIDEA/Prerequisites+for+Android+development>



Setting up IDE

Step 2 (alternative): the hard way

- Eclipse + ADT plugin
- Android SDK Tools
- Android Platform-tools
- A version of the Android platform
- A version of the Android system image for the emulator



Setting up IDE

Step 2: Eclipse != IntelliJ

- An Eclipse workspace is similar to a project in IntelliJ IDEA
- An Eclipse project maps to a module in IntelliJ IDEA
- An Eclipse project-specific JRE maps to a module JDK in IntelliJ IDEA
- An Eclipse user library maps to a global library in IntelliJ IDEA

Setting up IDE

Step 3: update SDK

- From Eclipse (with ADT), select Window > Android SDK Manager.
- From Android Studio, select Tools > Android > SDK Manager.
- On Windows, double-click the SDK Manager.exe file at the root of the Android SDK directory.
- On Mac or Linux, open a terminal and navigate to the tools/ directory in the Android SDK, then execute android sdk.

<http://developer.android.com/tools/help/sdk-manager.html>

Android SDK Manager

[Packages](#) [Tools](#)

SDK Path: C:\Program Files (x86)\Android\android-sdk

Packages

Name	API	Rev.	Status
<input checked="" type="checkbox"/>  Tools			
<input checked="" type="checkbox"/>  Android SDK Tools	16		 Installed
<input checked="" type="checkbox"/>  Android SDK Platform-tools	10		 Not installed
<input checked="" type="checkbox"/>  Android 4.0.3 (API15)			
<input checked="" type="checkbox"/>  Documentation for Android SDK	15	1	 Not installed
<input checked="" type="checkbox"/>  SDK Platform	15	2	 Not installed
<input checked="" type="checkbox"/>  Samples for SDK	15	1	 Not installed
<input checked="" type="checkbox"/>  ARM EABI v7a System Image	15	1	 Not installed
<input checked="" type="checkbox"/>  Google APIs by Google Inc.	15	1	 Not installed
<input checked="" type="checkbox"/>  Sources for Android SDK	15	1	 Not installed
<input type="checkbox"/>  Android 4.0 (API14)			
<input type="checkbox"/>  Android 3.2 (API13)			
<input type="checkbox"/>  Android 3.1 (API12)			
<input type="checkbox"/>  Android 3.0 (API11)			

Show: Updates/New Installed Obsolete Select [New](#) or [Updates](#)[Install 8 packages...](#)Sort by: API level Repository[Deselect All](#)[Delete 1 package...](#)

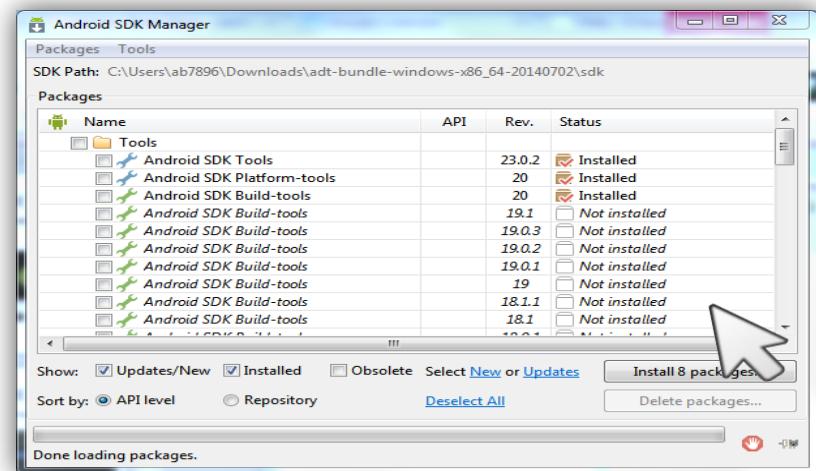
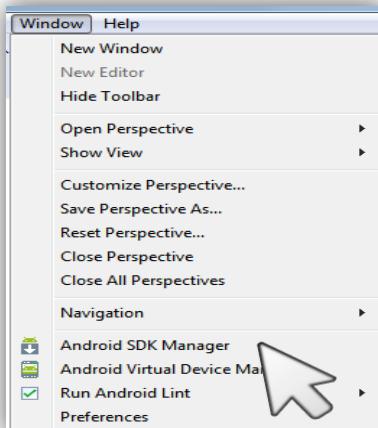
Done loading packages.

Setting up IDE

Step 3: update SDK

For every version going back to Android 2.2 (Froyo):

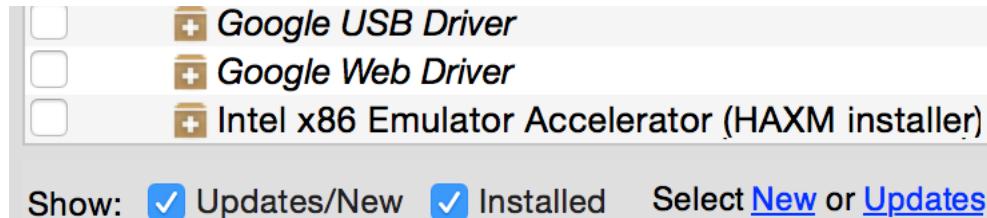
- The SDK Platform
- An emulator system image
- The Google APIs



Setting up IDE

Step 3: update SDK

The Intel Emulator



Stack Overflow is a question and answer site for professional and enthusiast programmers. Registration required.

Why is the Android emulator so slow?

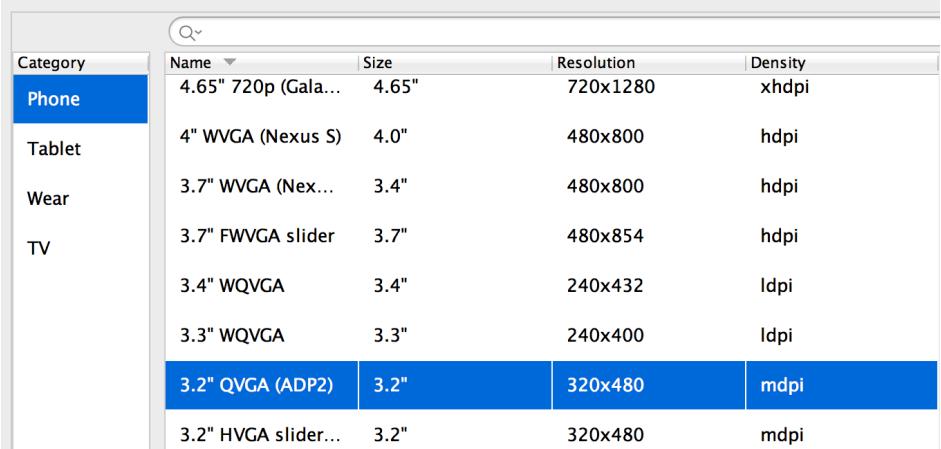
▲ 1631 I have a 2.67 GHz Celeron processor, 1.21 GB of RAM on understanding is that the Android emulator should start fast not. I have followed all the instructions in setting up the IDE successfully in starting the emulator quickly but that is very rare.

★ 854 Even if it starts and loads the home screen, it is very sluggish on a Samsung Galaxy S II running Ganymede.

Setting up IDE

Step 4: create Android virtual devices (AVDs)

- AVD Name: MyTestDevice
- Device: 3.2 QVGA
- Target: Android 5.0 (x86)



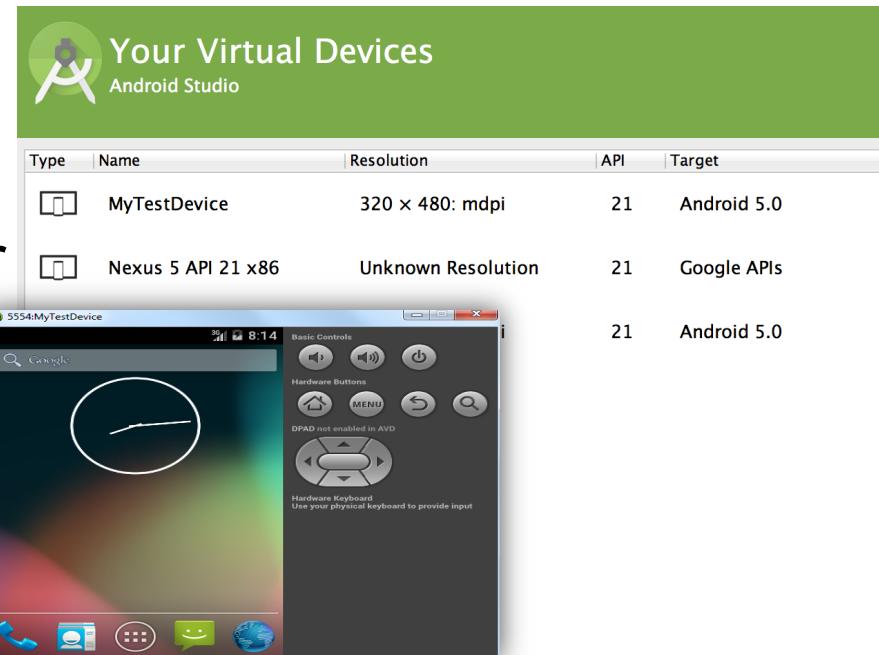
The screenshot shows the Android Virtual Device Manager window. On the left, there is a sidebar with categories: Phone (selected), Tablet, Wear, and TV. The main area displays a table of virtual device configurations. The columns are: Name, Size, Resolution, and Density. The table contains the following data:

Category	Name	Size	Resolution	Density
Phone	4.65" 720p (Gala...	4.65"	720x1280	xhdpi
Tablet	4" WVGA (Nexus S)	4.0"	480x800	hdpi
Wear	3.7" WVGA (Nex...	3.4"	480x800	hdpi
TV	3.7" FWVGA slider	3.7"	480x854	hdpi
	3.4" WQVGA	3.4"	240x432	ldpi
	3.3" WQVGA	3.3"	240x400	ldpi
	3.2" QVGA (ADP2)	3.2"	320x480	mdpi
	3.2" HVGA slider...	3.2"	320x480	mdpi

Setting up IDE

Step 4: create Android virtual devices (AVDs)

- 💡 Start with one of the smaller devices because it's easier to scale up when developing the user interface than to scale down.
- 💡 AVDs can be quite slow, give it time.



Setting up IDE

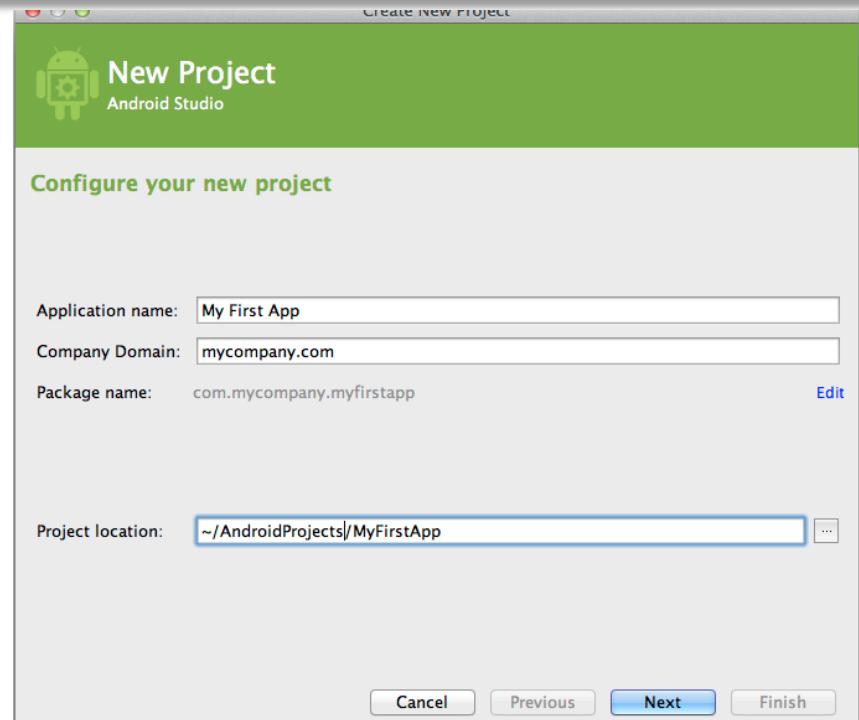
Step 5 (optional): testing device

- Verify that your application is "debuggable" in your manifest or build.gradle file.
- Enable USB debugging on your device.
- Set up your system to detect your device.

```
    android {  
        buildTypes {  
            debug {  
                debuggable true  
            }  
        }  
    }
```

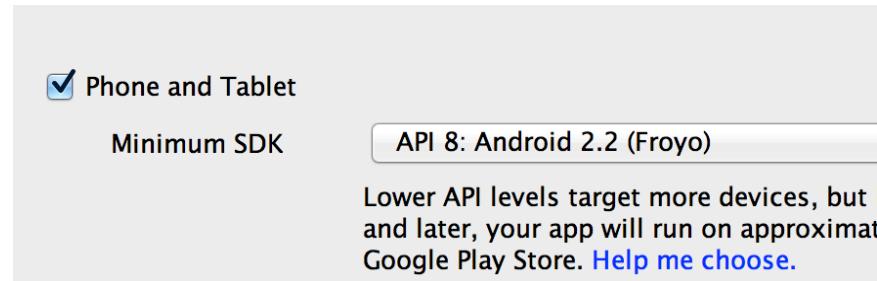
Hello, Android!

- **Application Name** is the app name that appears to users.
- **Company domain** provides a qualifier that will be appended to the package name; Android Studio will remember this qualifier for each new project you create.
- **Package Name** is the package namespace for your app (as packages in the Java).

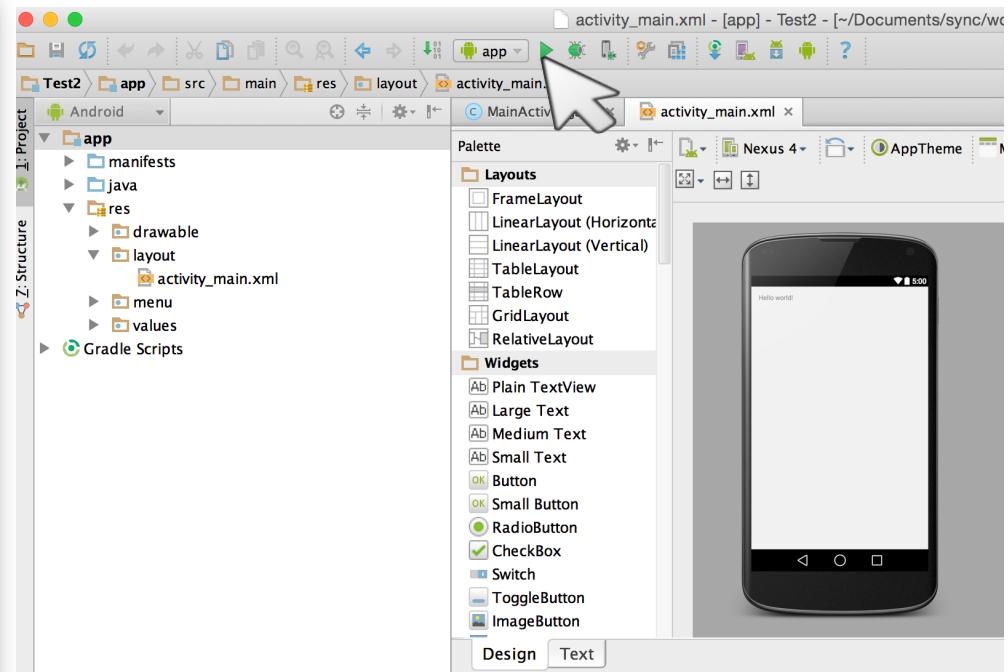
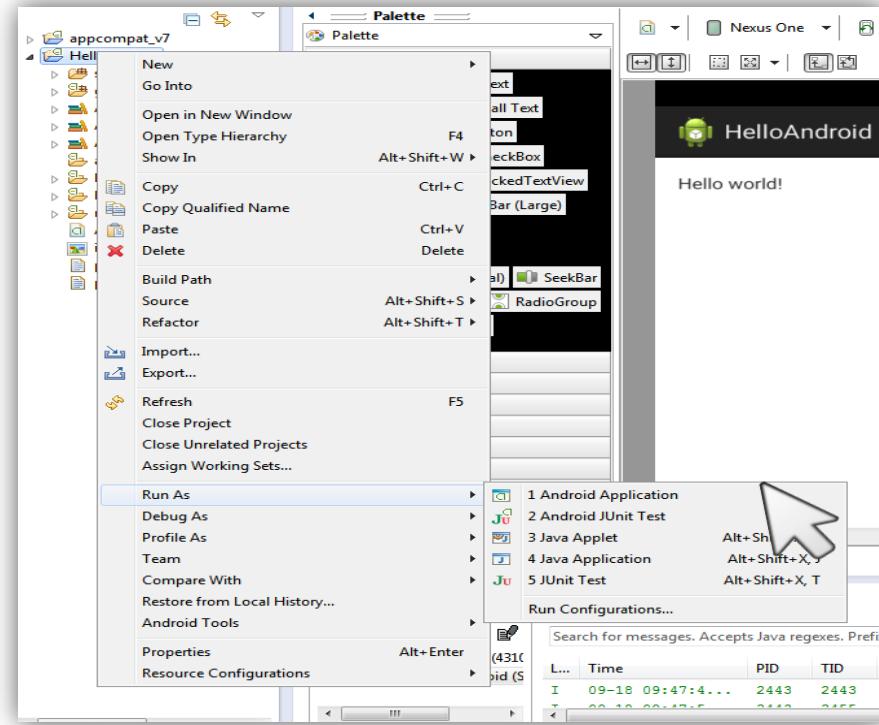


Hello, Android!

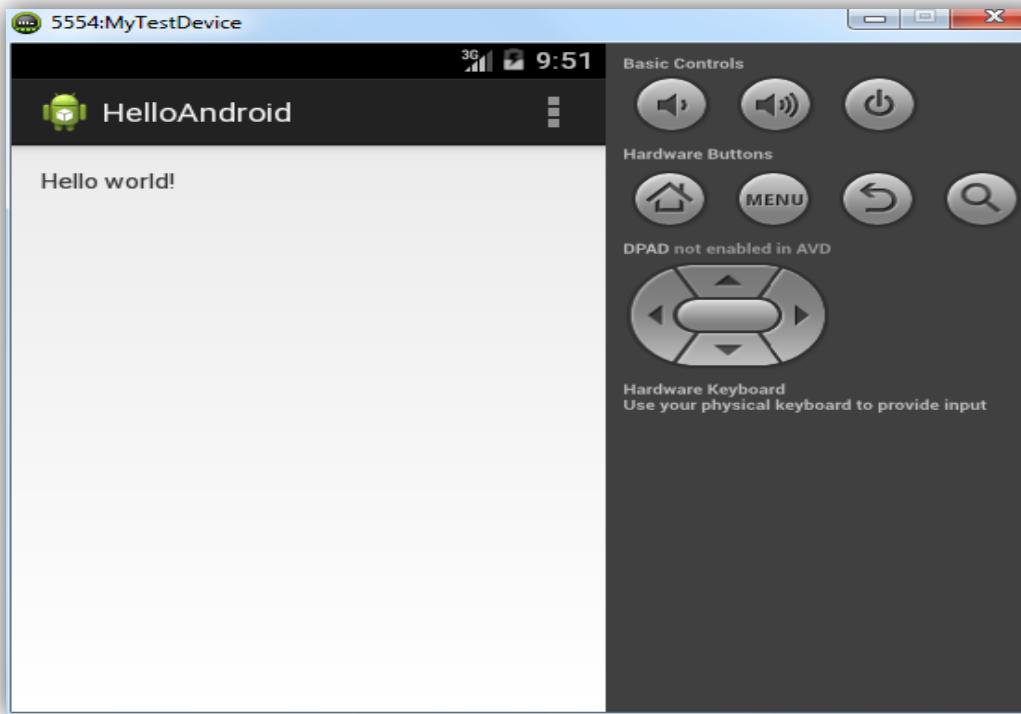
- **Minimum Required SDK** is the lowest version of Android that your app supports, indicated using the API level.
- **Target SDK** indicates the highest version of Android (also using the API level) with which you have tested with your application.
- **Compile With** is the platform version against which you will compile your app.



Hello, Android!

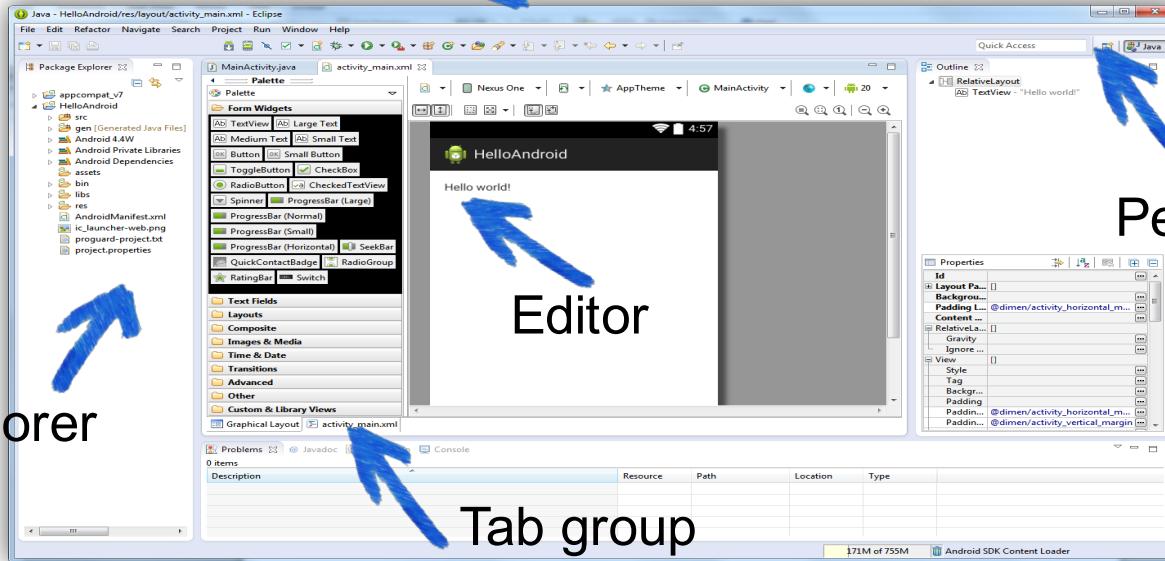


Hello, Android!



Hello, Android!

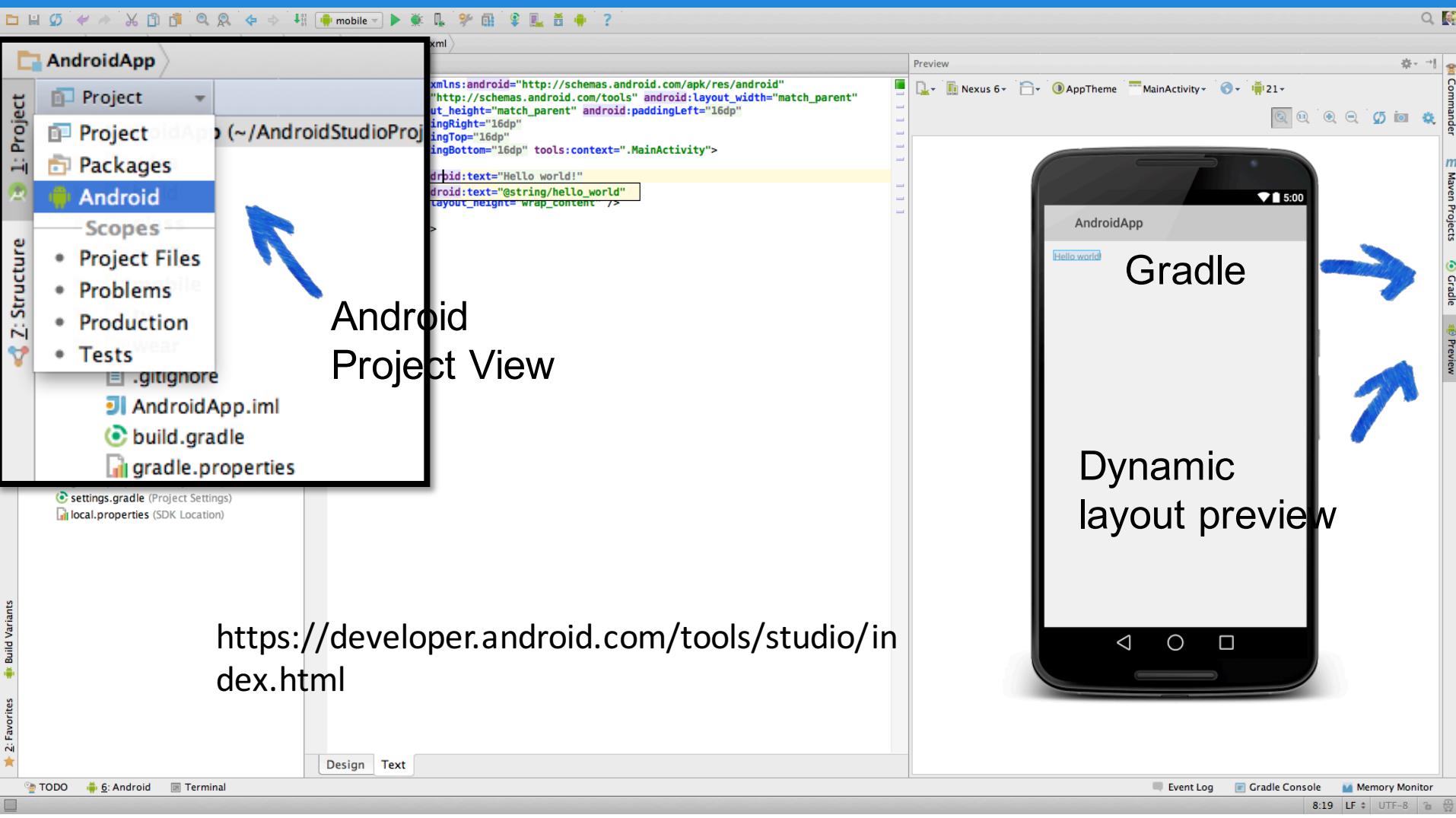
Workbench window



Package Explorer

Tab group

Perspectives



Android Project View

Gradle

Dynamic layout preview

<https://developer.android.com/tools/studio/index.html>

Android anatomy

- **manifest.xml**

This is the manifest file for your Android application. Here you specify the permissions needed by your application.

- **java**

Contains the .java source files for your project.

- **res**

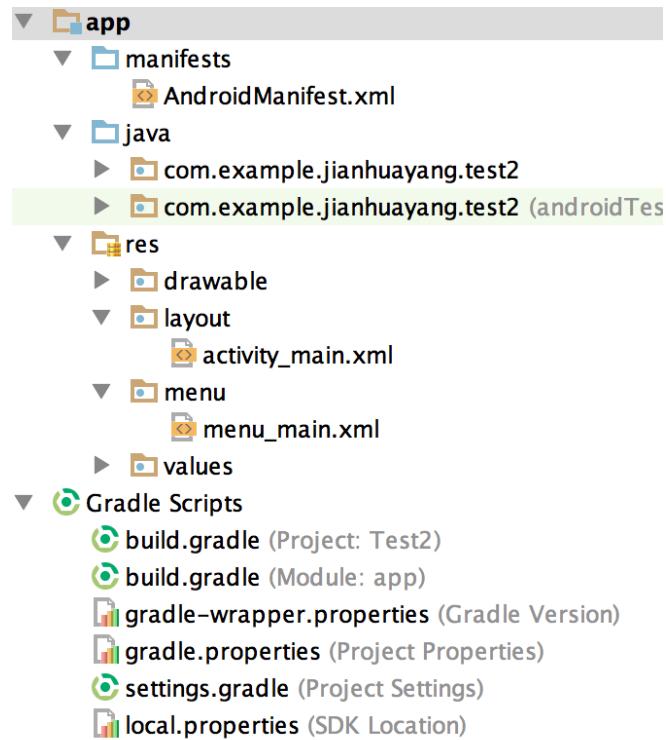
This folder contains all the resources used in your application.

- **Gradle**

Customizable properties for the build system.

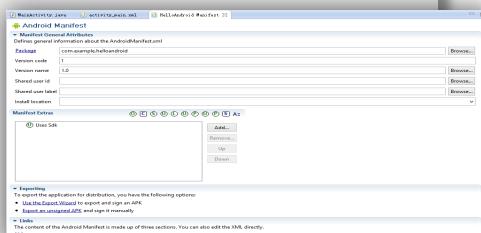
- **gen**

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 anatomy

```
1. <?xml version="1.0" encoding="utf-8"?>
2. <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3.   package="com.example.helloandroid"
4.   android:versionCode="1"
5.   android:versionName="1.0" >
6.
7. <uses-sdk
8.   android:minSdkVersion="8"
9.   android:targetSdkVersion="21" />
10.
11. <application
12.   android:allowBackup="true"
13.   android:icon="@drawable/ic_launcher"
14.   android:label="@string/app_name"
15.   android:theme="@style/AppTheme" >
16.   <activity
17.     android:name=".MainActivity"
18.     android:label="@string/app_name" >
19.     <intent-filter>
20.       <action android:name="android.intent.action.MAIN" />
21.
22.       <category android:name="android.intent.category.LAUNCHER" />
23.     </intent-filter>
24.   </activity>
25. </application>
26.
27. </manifest>
```



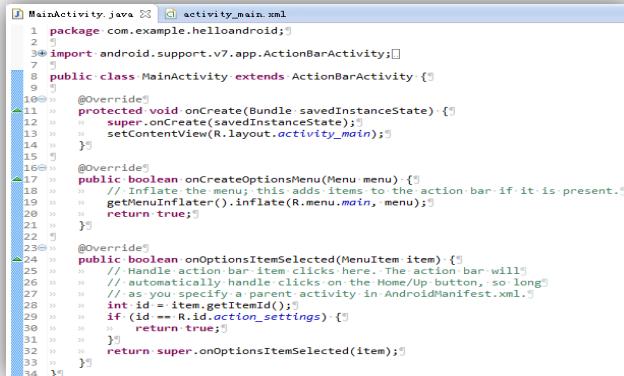
Manifest:

- App's components
- User permissions the app requires
- Minimum API Level required by the app
- Hardware and software features used or required by the app
- API libraries the app needs to be linked against

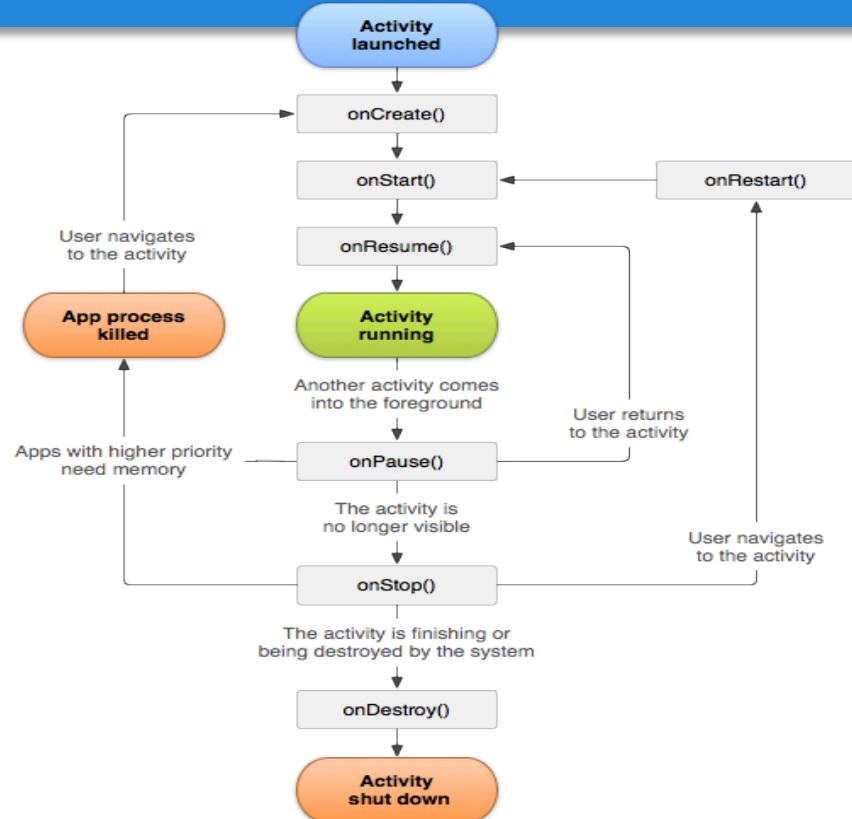
Android anatomy

Activity :

- An activity is an instance of Activity, a class in the Android SDK.
- An activity is responsible for managing user interaction with a screen of information.



```
MainActivity.java
1 package com.example.helloandroid;
2
3 import android.support.v7.app.ActionBarActivity;
4
5 public class MainActivity extends ActionBarActivity {
6
7     @Override
8     protected void onCreate(Bundle savedInstanceState) {
9         super.onCreate(savedInstanceState);
10        setContentView(R.layout.activity_main);
11    }
12
13    @Override
14    public boolean onCreateOptionsMenu(Menu menu) {
15        // Inflate the menu; this adds items to the action bar if it is present.
16        getMenuInflater().inflate(R.menu.main, menu);
17        return true;
18    }
19
20    @Override
21    public boolean onOptionsItemSelected(MenuItem item) {
22        // Handle action bar item clicks here. The action bar will
23        // automatically handle clicks on the Home/Up button, so long
24        // as you specify a parent activity in AndroidManifest.xml.
25        int id = item.getItemId();
26        if (id == R.id.action_settings) {
27            return true;
28        }
29        return super.onOptionsItemSelected(item);
30    }
31
32
33
34 }
```



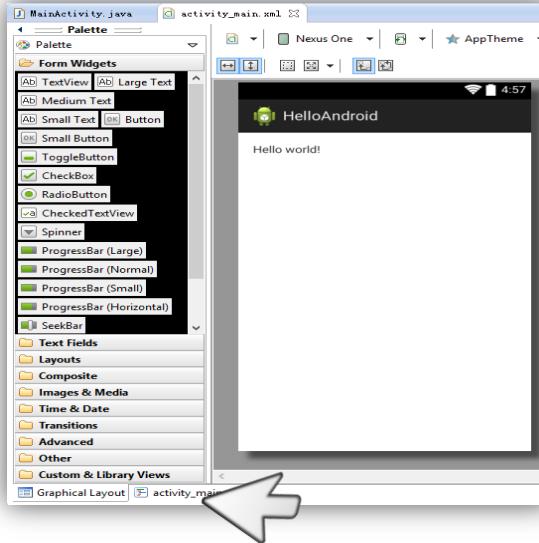
<http://developer.android.com/reference/android/app/Activity.html>

<http://developer.android.com/guide/components/fundamentals.html>

Android anatomy

Layout:

- A layout defines a set of user interface objects and their position on the screen.
- A layout is made up of definitions written in XML.

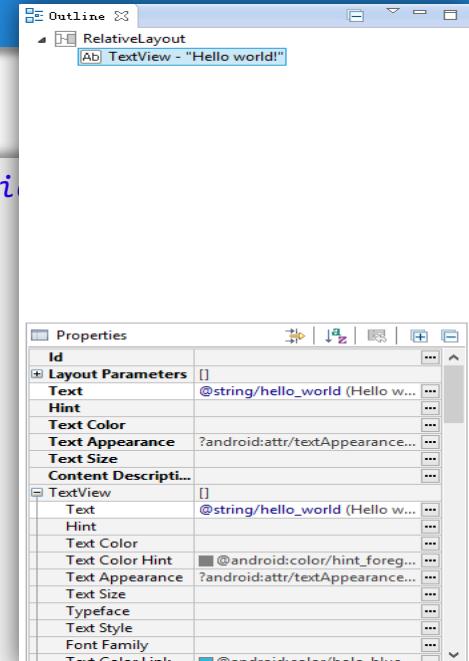


```
1 <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
2     android:layout_width="match_parent"
3     android:layout_height="match_parent">
4     
11    <TextView
12        android:layout_width="wrap_content"
13        android:layout_height="wrap_content"
14        android:text="@string/hello_world" />
15
16 </RelativeLayout>
```

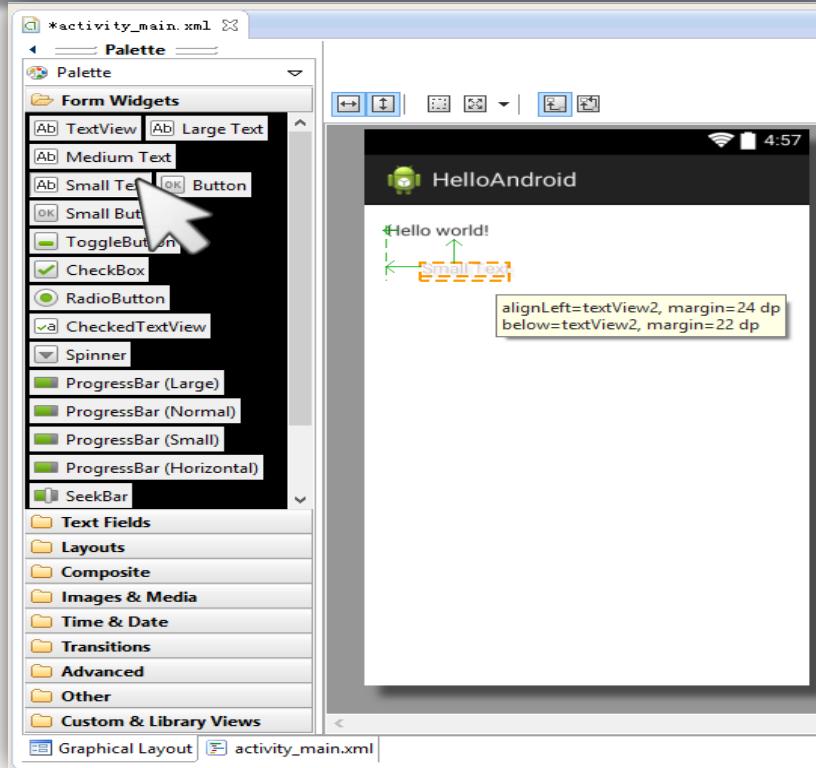
Coding the interface

Auto-generated layout

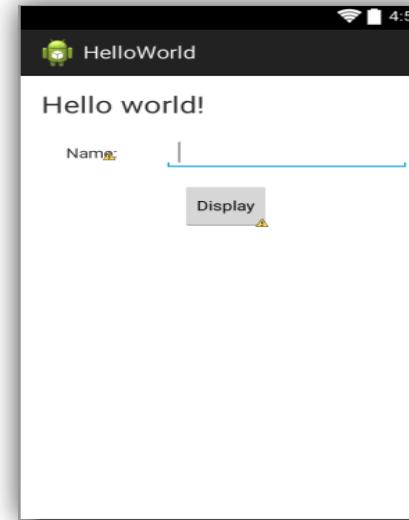
```
1. <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
2.     xmlns:tools="http://schemas.android.com/tools"
3.     android:layout_width="match_parent"
4.     android:layout_height="match_parent"
5.     android:paddingBottom="@dimen/activity_vertical_margin"
6.     android:paddingLeft="@dimen/activity_horizontal_margin"
7.     android:paddingRight="@dimen/activity_horizontal_margin"
8.     android:paddingTop="@dimen/activity_vertical_margin"
9.     tools:context="com.example.helloandroid.MainActivity" 
10.
11.    <TextView
12.        android:layout_width="wrap_content"
13.        android:layout_height="wrap_content"
14.        android:text="@string/hello_world" />
15.
16.</RelativeLayout>
```



Coding the interface



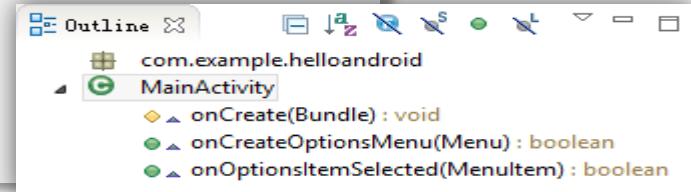
- SmallText
- PersonName
- Button



Coding the behaviour

Auto-generated activity

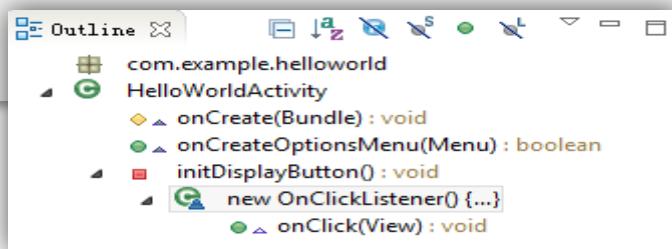
```
1. public class MainActivity extends ActionBarActivity {  
2.  
3.     @Override  
4.     protected void onCreate(Bundle savedInstanceState) {  
5.         super.onCreate(savedInstanceState);  
6.         setContentView(R.layout.activity_main);  
7.     }  
8.  
9.     @Override  
10.    public boolean onCreateOptionsMenu(Menu menu) {  
11.        getMenuInflater().inflate(R.menu.main, menu);  
12.        return true;  
13.    }  
14.  
15.    @Override  
16.    public boolean onOptionsItemSelected(MenuItem item) {  
17.        int id = item.getItemId();  
18.        if (id == R.id.action_settings) {  
19.            return true;  
20.        }  
21.        return super.onOptionsItemSelected(item);  
22.    }  
23.}
```



Coding the behaviour

setOnClickListener

```
1. private void initDisplayButton() {  
2.     Button displayButton = (Button) findViewById(R.id.buttonDisplay);  
3.     displayButton.setOnClickListener(new OnClickListener() {  
4.  
5.         @Override  
6.         public void onClick(View arg0) {  
7.             EditText editName = (EditText) findViewById(R.id.editTextName);  
8.             TextView textDisplay = (TextView) findViewById(R.id.textViewDisplay);  
9.             String nameToDisplay = editName.getText().toString();  
10.            textDisplay.setText("Hello " + nameToDisplay);  
11.  
12.        }  
13.  
14.    });  
15.}
```



Create-modify-reuse

- Code examples from “Learning Mobile App Development”
- Downloadable at <https://github.com/LearningMobile/BookApps>
- Import



The screenshot displays several windows from the Eclipse IDE:

- Import Dialog:** Shows the 'Select' tab with the message 'Create new projects from an archive file or directory.' and a 'Select an import source:' dropdown. Under 'General', 'Existing Projects into Workspace' is selected.
- Properties Dialog:** For the project 'HelloWorld'. The 'Android' tab is selected, showing the 'Project Build Target' table:

Target Name	Vendor	Platform	API ...
<input checked="" type="checkbox"/> Android 4.4W	Android Open Source Project	4.4W	20
<input type="checkbox"/> Android L (Preview)	Android Open Source Project	L	L
- Problems View:** Shows 4 errors, 0 warnings, and 0 others. The errors listed are:
 - The project was not built since its build path is incomplete. Cannot resolve file reference.
 - The type java.lang.Object cannot be resolved. It is indirectly referenced from required API package android.os.
 - Unable to resolve target 'android-17'
 - Unable to resolve target 'android-17'
- Package Explorer:** Shows the project structure with 'HelloAndroid' as the root. It includes 'HelloWorld' which contains 'src', 'gen [Generated Java Files]', 'Android 4.4W', 'Android Private Libraries', 'bin', 'libs', 'res', 'AndroidManifest.xml', 'ic_launcher-web.png', 'proguard-project.txt', and 'project.properties'.