

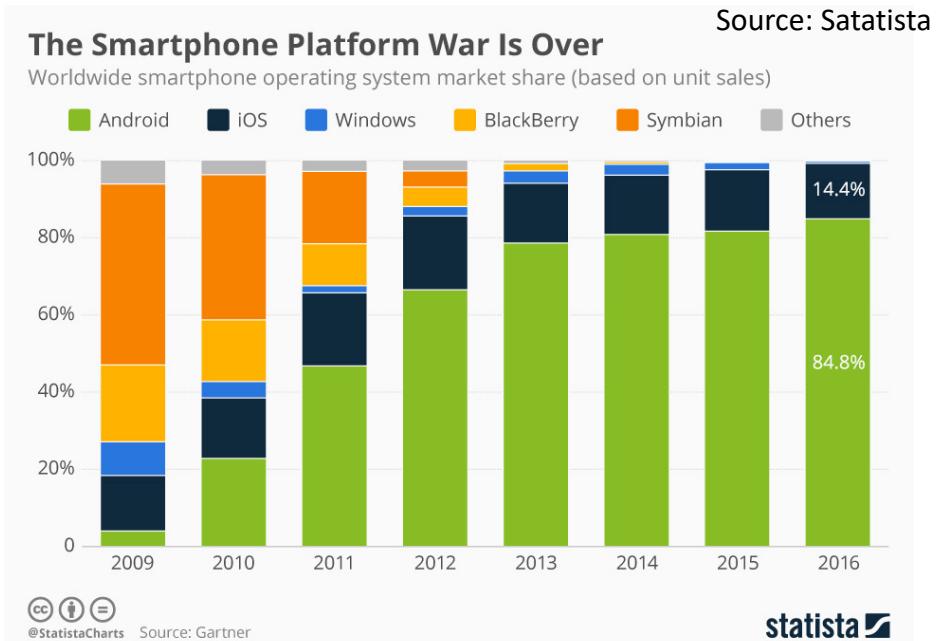
Introduction to Android

Android

- Android is world's most popular 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 Google Play store
- Open Handset Alliance: 80+ companies
 - Mobile operators: Vodafone, Telefonica, Telecom Italia, NTT Docomo, Sprint, ...
 - Handset manufacturers: Samsung, Sharp, Sony, HTC, Acer, Asus, Nec, ...
 - Semiconductor companies: ARM, Freescale, Nvidia, TI, ...
 - Software companies: Google, eBay, ...
 - Other companies: Accenture, Wind River

Android's history and present

- History:
 - 2005 Google buys Android Inc.
 - 2007 OHA is announced and Android is presented
 - 2008 Android SDK 1.0 is released
 - 2010 Nexus One is commercialized
 - 2012 800K+ Android devices activated every day
- Worldwide Smartphone Sales to End Users by Operating System in 2017 (Thousands of Units):

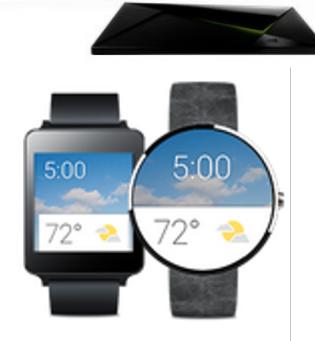
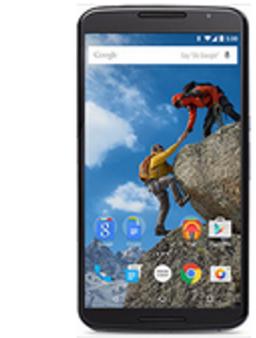


Operating System	2017 Units	2017 Market Share (%)	2016 Units	2016 Market Share (%)
Android	1,320,118.1	85.9	1,268,562.7	84.8
iOS	214,924.4	14.0	216,064.0	14.4
Other OS	1,493.0	0.1	11,332.2	0.8
Total	1,536,535.5	100.0	1,495,959.0	100.0

Source: Gartner (February 2018)

Multiple sizes and purposes

- A range of devices (form factors):
 - Smartphone (from the beginning)
 - Tablet (support introduced in Android 3.0, now considered same as smartphones)
 - TV (launch June 2014, Android 5.0)
 - Auto (launch June 2014, Android 5.0)
 - Wear (support introduced in Android 4.4)
 - Glass (experimental project, now only limited distribution with selected companies)



Supported platforms

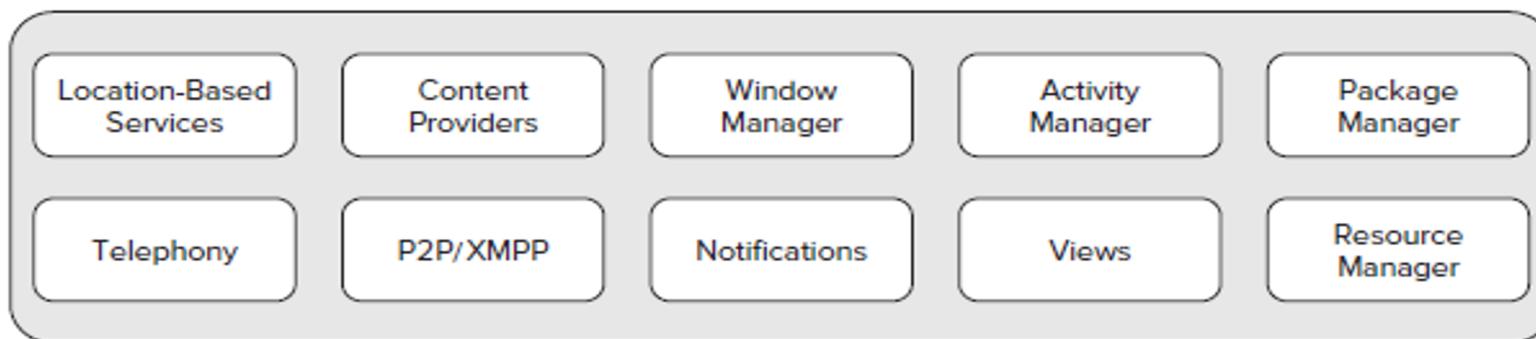
- Supported HW platforms:
 - Supports ARM, x86, MIPS architectures (ARM from the beginning)
 - Both 32bit and 64bit (the latter since Android 5.0)
 - ARM/x86/MIPS or 32/64, transparent at the application level (unless you need native (C) code)

Android

Application Layer



Application Framework



Libraries

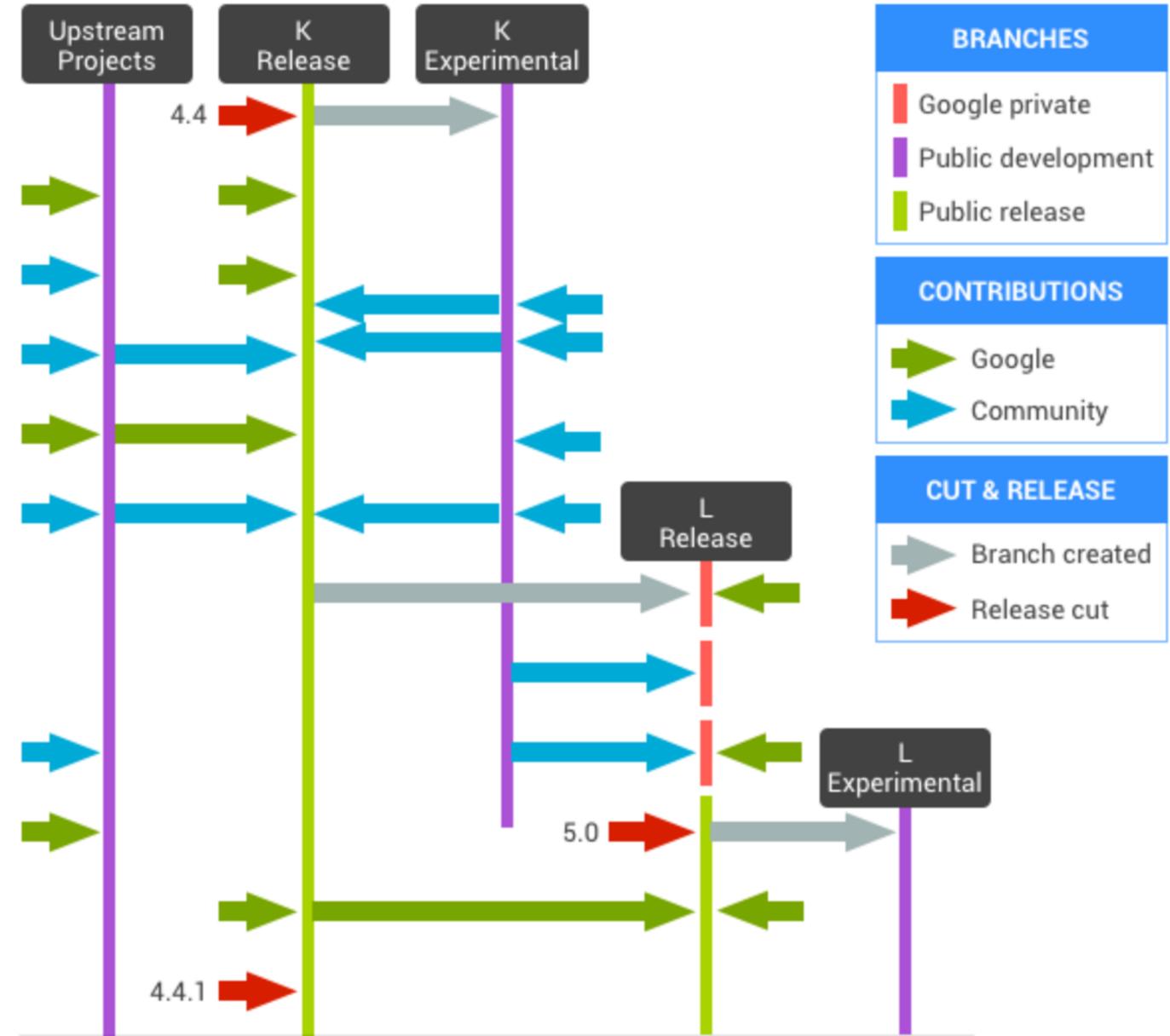


Linux Kernel



Android

- AOSP maintains the software
- Software is ported by OEMs to run on their own hardware



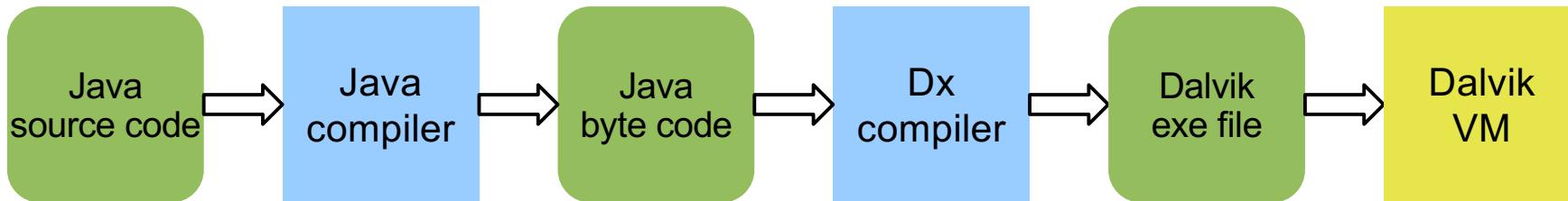
Android app development

- Development tools: free, no specific license is needed
- Distributed as APK file. Each file contains:
 - *dex* executable
 - resources (xml files, images, ...)
 - native libs
- Distribution
 - app stores: Google Play Store and others from third parties
 - the Web: just link the APK on a webpage
 - You can also send the APK by email



Java and Dalvik

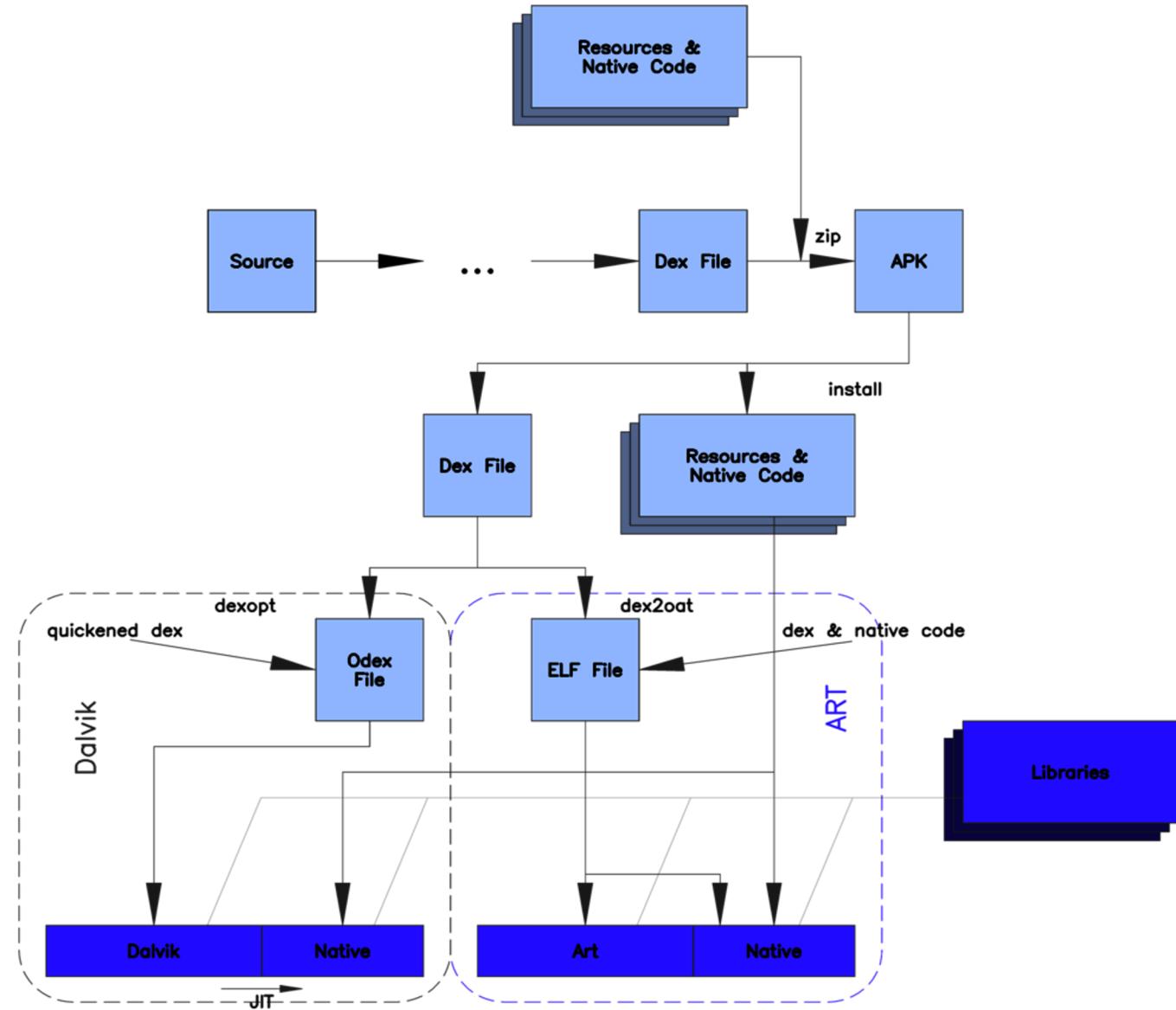
- Dalvik VM ensures developers never have to worry about a particular HW implementation
- The Dalvik VM executes Dalvik executable files, a format optimized to ensure minimal memory footprint
- Developers create *.dex* executables by transforming Java language compiled classes using the tools supplied within the SDK



- Source files: which version of Java? Approximately JSE without GUI, plus Android specific libraries

ART

- ART: Android Run-Time (default since 5.0)
- Dalvik vs ART
 - JIT vs AOT
 - ART faster (depends on application, some benchmarks 2x), less energy (no compilation)



Why Android?

- Runs on a large variety of devices and platforms
 - Easier to build pervasive applications
- Development tools are free of charge and available for different OSes
- Larger user base
- Programming language you already know (Java)
- Code is open source
 - Actually some components are proprietary (e.g. Maps, Play store)

Additional development problems

- Many versions of Android out there
 - Different APIs
 - Available hardware depends on devices (even if same API level)
 - Different UI models (some buttons have disappeared)
 - Different appearance of UI elements
 - Some vendors introduce their own customizations to the UI and may provide additional libraries
 - If you use native code, you have to provide compiled binaries for the processor architectures you want to support
- Several form factors: smartphones, tablets, TVs, smartwatches
- We will focus on smartphones

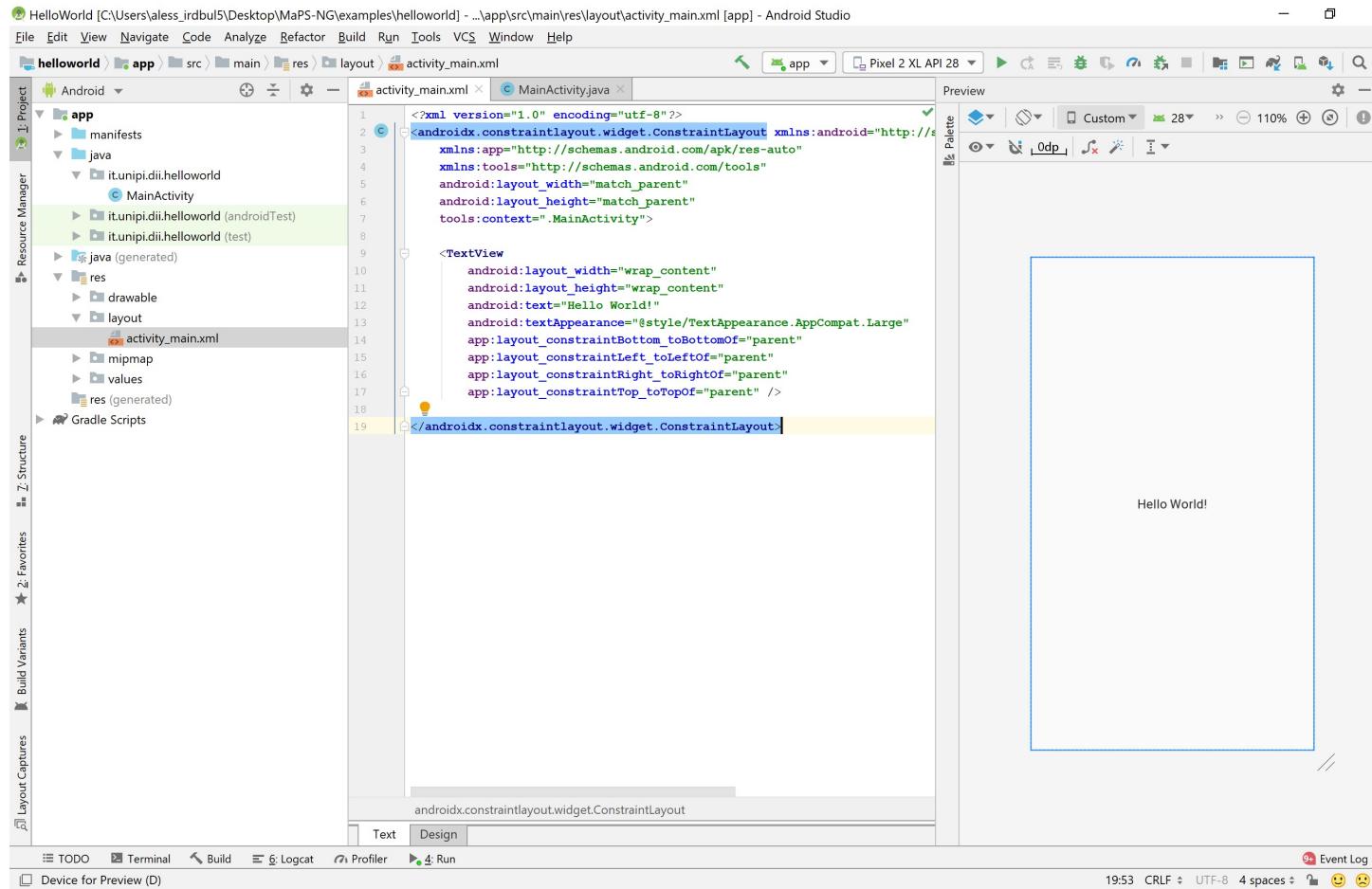
Android versions

- Version
- Codename
- API level

ANDROID PLATFORM VERSION	API LEVEL	CUMULATIVE DISTRIBUTION
4.1 Jelly Bean	16	
4.2 Jelly Bean	17	99,8%
4.3 Jelly Bean	18	99,5%
4.4 KitKat	19	99,4%
5.0 Lollipop	21	98,0%
5.1 Lollipop	22	97,3%
6.0 Marshmallow	23	94,1%
7.0 Nougat	24	89,0%
7.1 Nougat	25	85,6%
8.0 Oreo	26	82,7%
8.1 Oreo	27	78,7%
9.0 Pie	28	69,0%
10. Q	29	50,8%
11. R	30	24,3%

Development environment

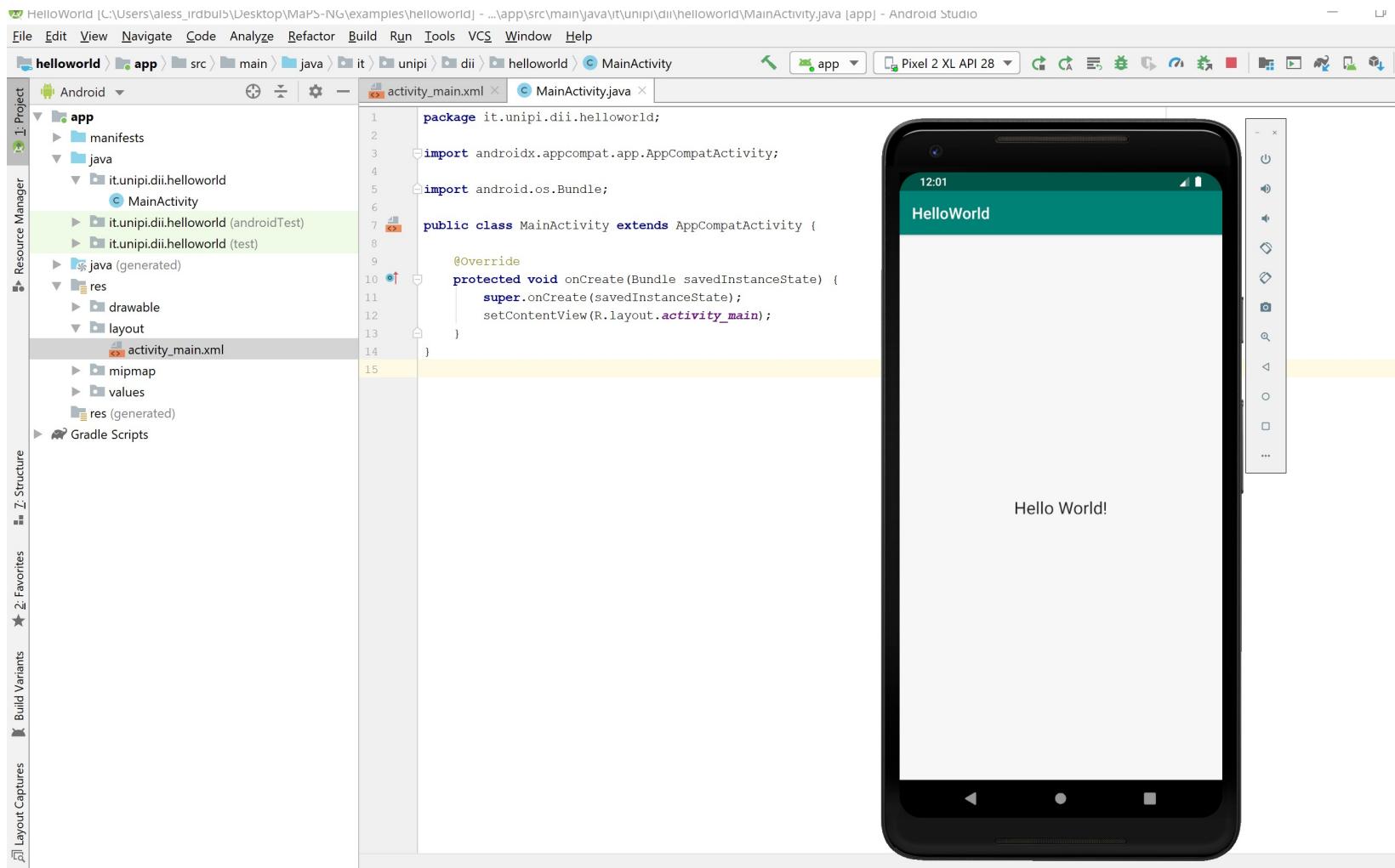
- Android Studio, based on IntelliJ IDEA
- Compiles to executables (.dex), packages files (.apk), and deploys to phone



Old IDE: Eclipse +
plugins,
cumbersome to
set-up, many
bugs, abandoned

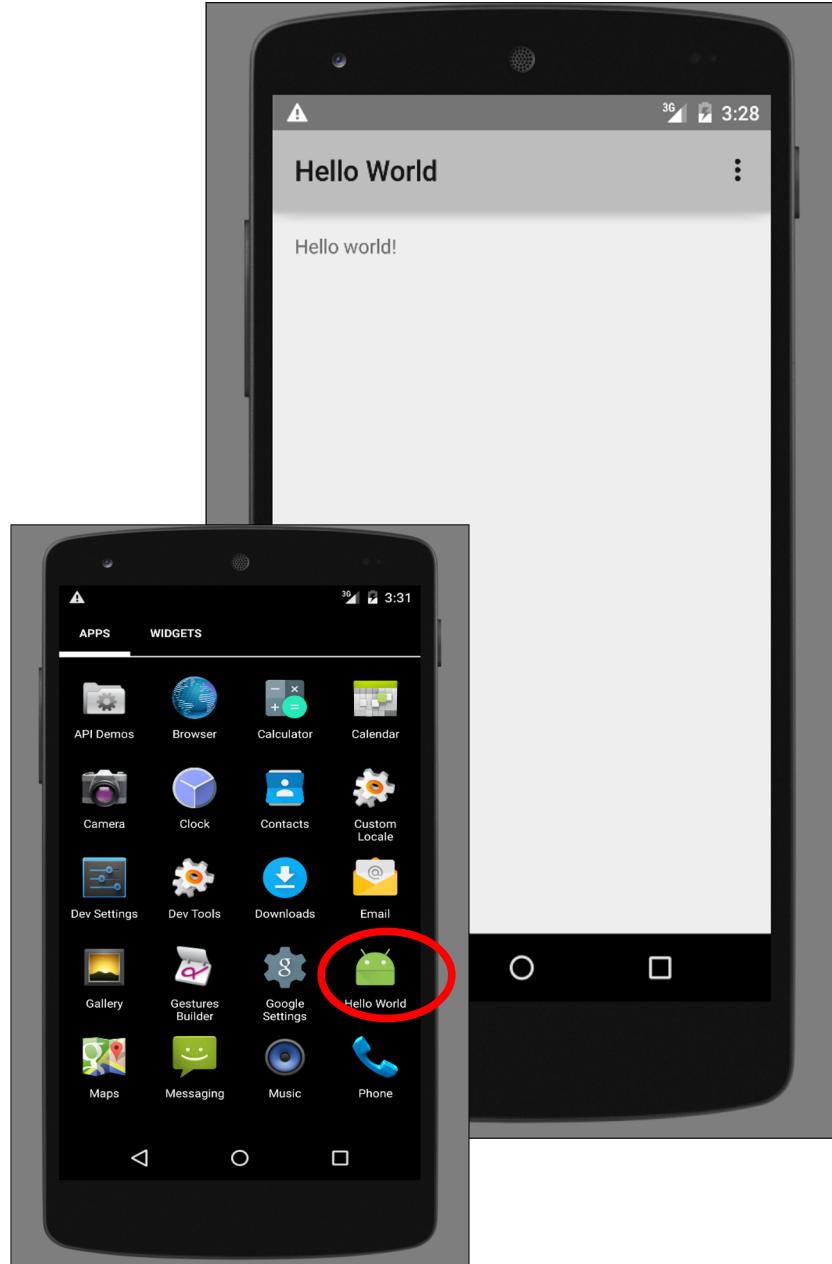
Running apps

- Apps can be executed
 - On a real device
 - On the emulator provided by Android Studio
- Real devices must be enabled:
 - *Settings > About phone* and tap *Build number* 7 times
 - *Settings > Developer options* enable *USB debugging*



Running

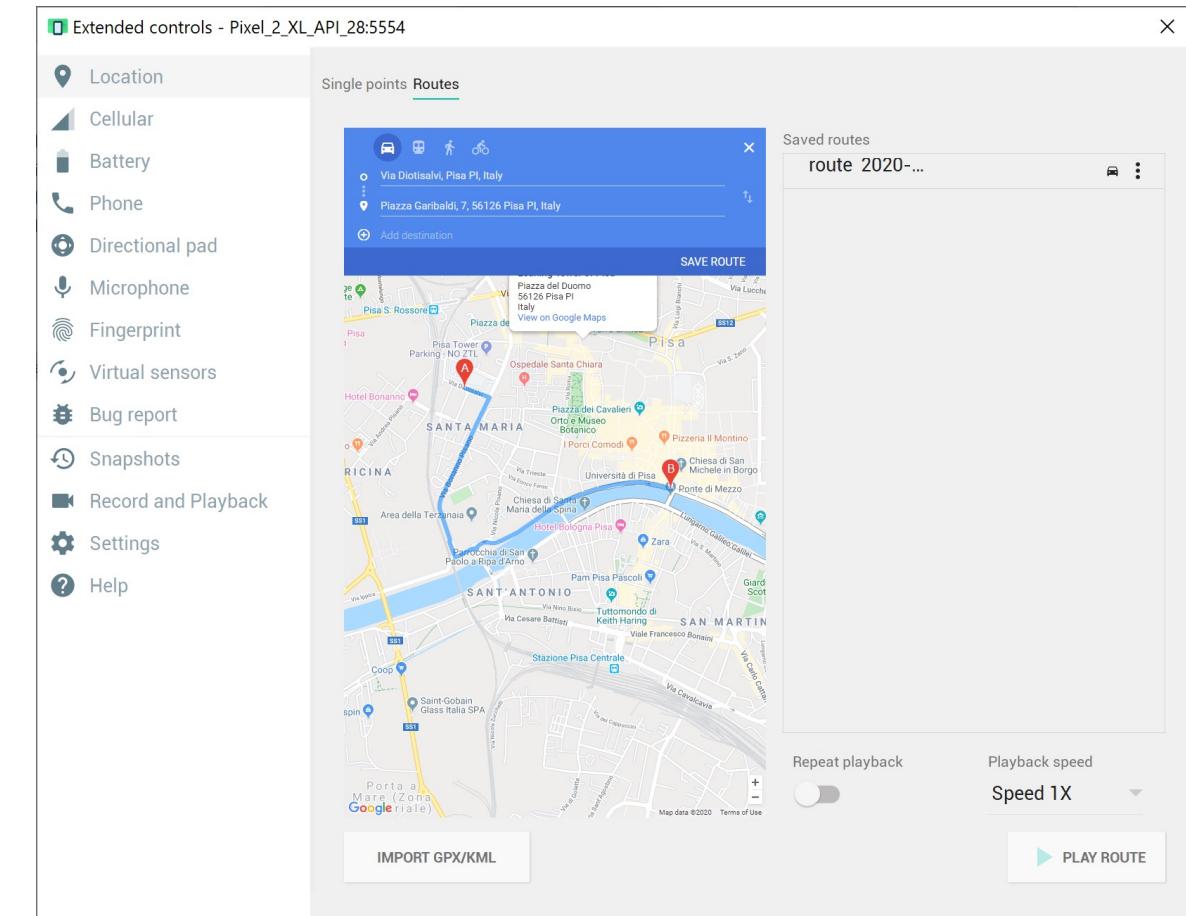
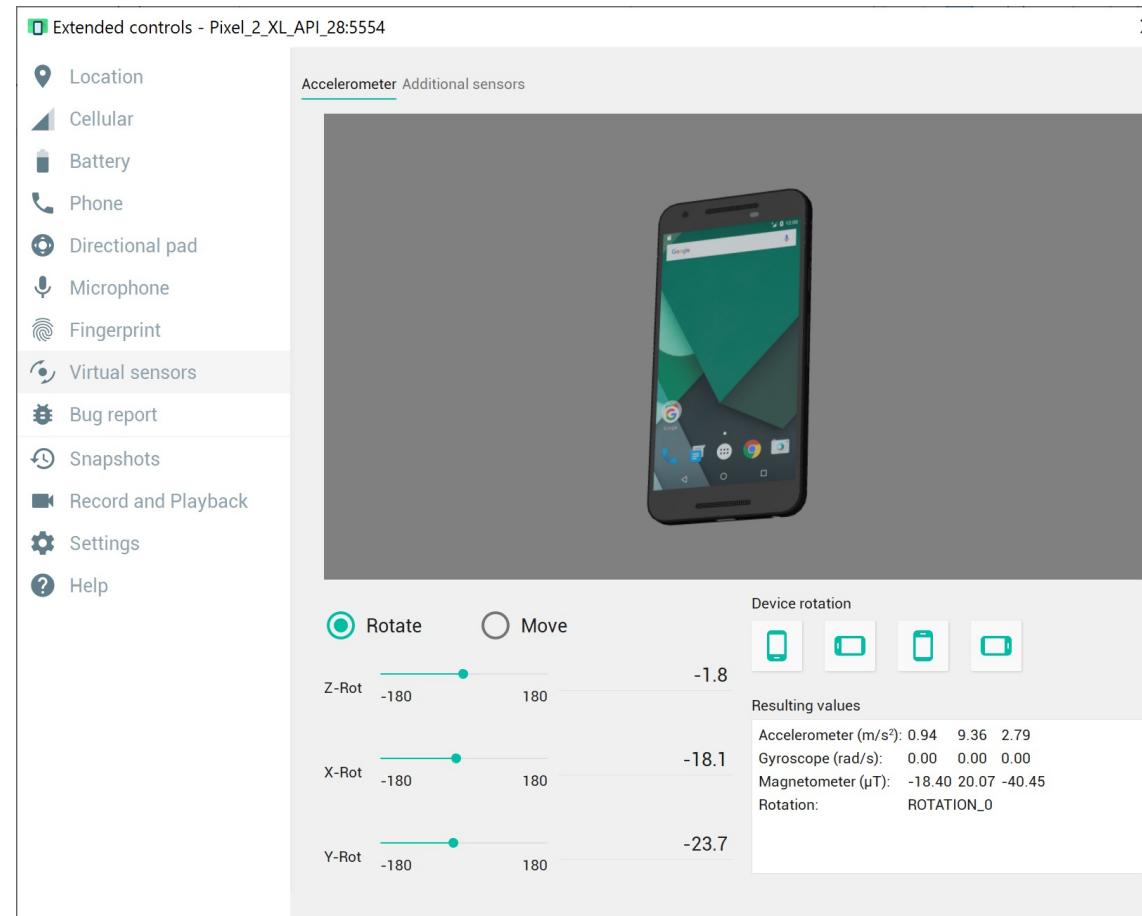
- The emulator boots like a real device, then your application is started
 - Keep the emulator running to save time
- If you store something on the persistent memory of the emulator, it will be persistent
- You can “navigate” within the emulator like a real device



Emulator Pros and Cons (Vs Real Phone)

- **Pros:**
 - Convenient execution of apps within the development environment
 - Easy to test app on various emulated devices (phones, tablets, TVs, etc), various screen sizes
- **Cons:**
 - Slower than real phone
 - Once there was limited support for GPS, camera, video recording, making/receiving phone calls, Bluetooth devices, USB devices, battery level, sensors, etc. Now much better
- Emulator is OK for this class (in case you don't have a real Android smartphone)

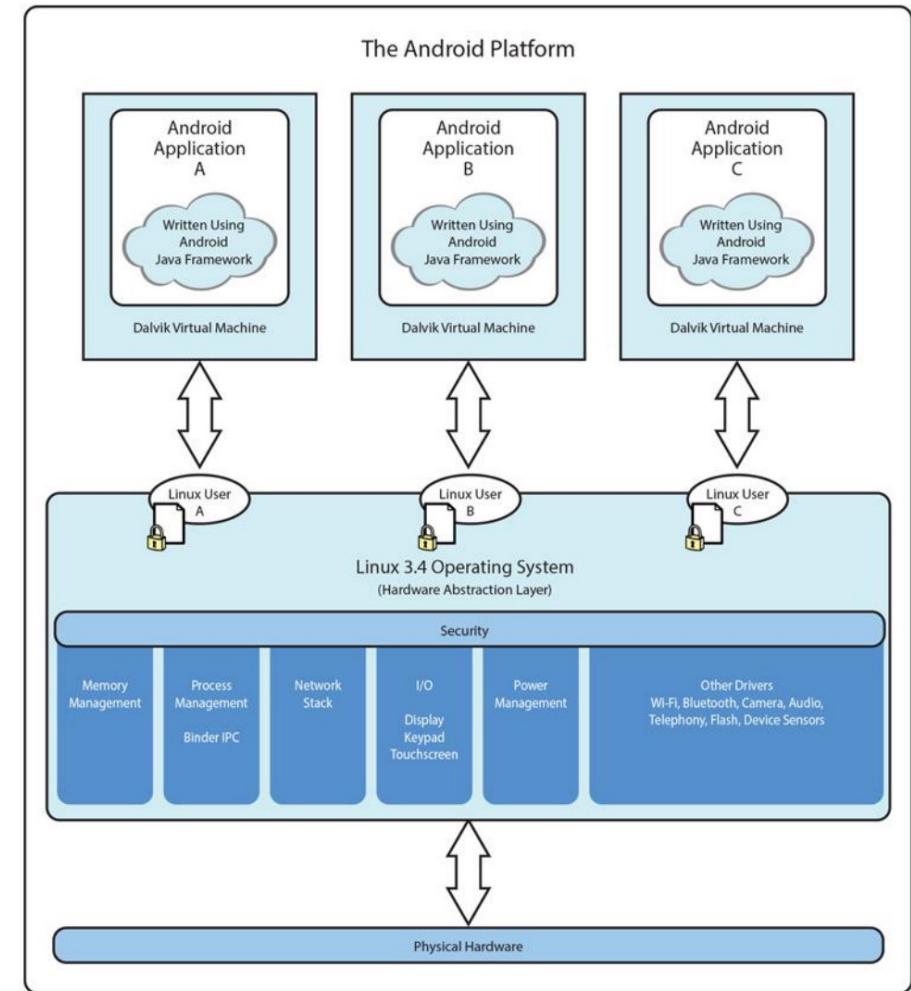
Support for artificial sensor readings



Demo

Isolation between apps

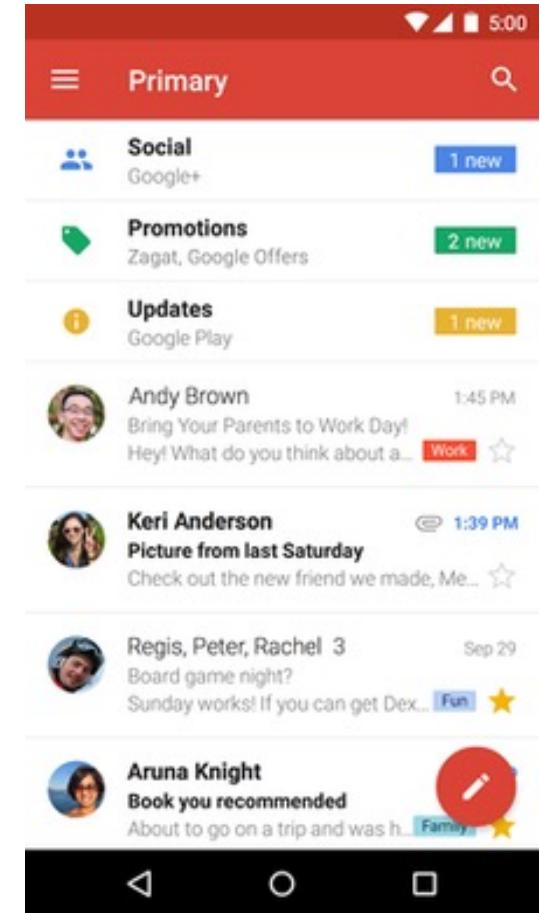
- Android is a multi-user Linux system
- Each application is a different user, ID assigned by the system
- Access control: only process with same ID can access files
- Each process has its own VM, so an application's code runs in isolation from other applications
- Every application runs in its own Linux process.
 - Started when any of the app components needs to be executed
 - Shut down when no longer used



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

Developing apps: UI + logic

- 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



HelloWorld

Create New Project X

Choose your project

Phone and Tablet Wear OS TV Android Auto Android Things

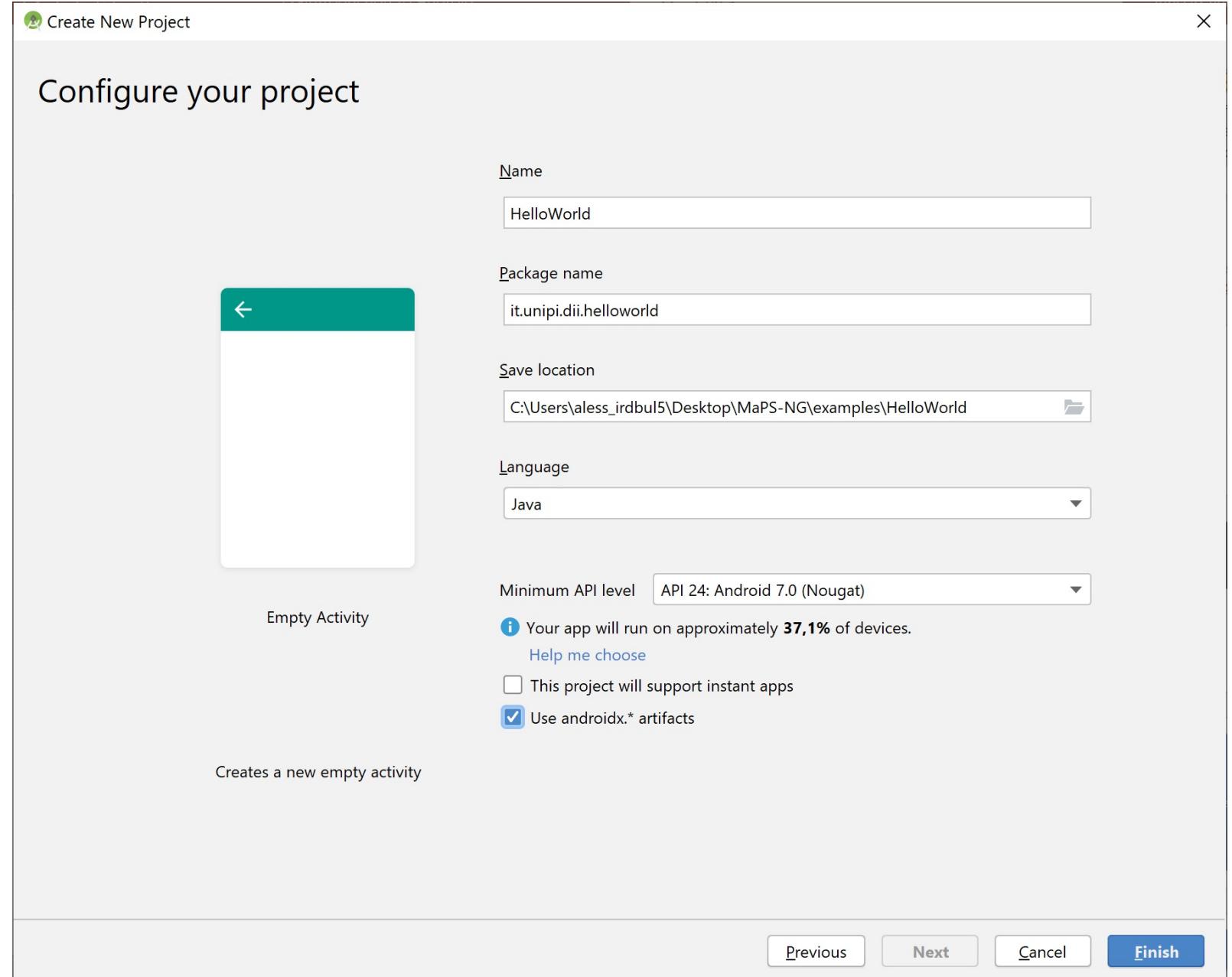
The screenshot shows the 'Create New Project' dialog with the 'Phone and Tablet' tab selected. It displays eight activity templates arranged in two rows of four:

- Add No Activity**: A simple activity template with a yellow '+' button.
- Basic Activity**: A basic activity template.
- Empty Activity**: An empty activity template.
- Bottom Navigation Activity**: An activity template featuring a bottom navigation bar.
- Fragment + ViewModel**: An activity template with a fragment and view model.
- Fullscreen Activity**: A fullscreen activity template.
- Master/Detail Flow**: A master/detail flow activity template.
- Navigation Drawer Activity**: An activity template with a navigation drawer.

Empty Activity
Creates a new empty activity

[Previous](#) [Next](#) [Cancel](#) [Finish](#)

HelloWorld

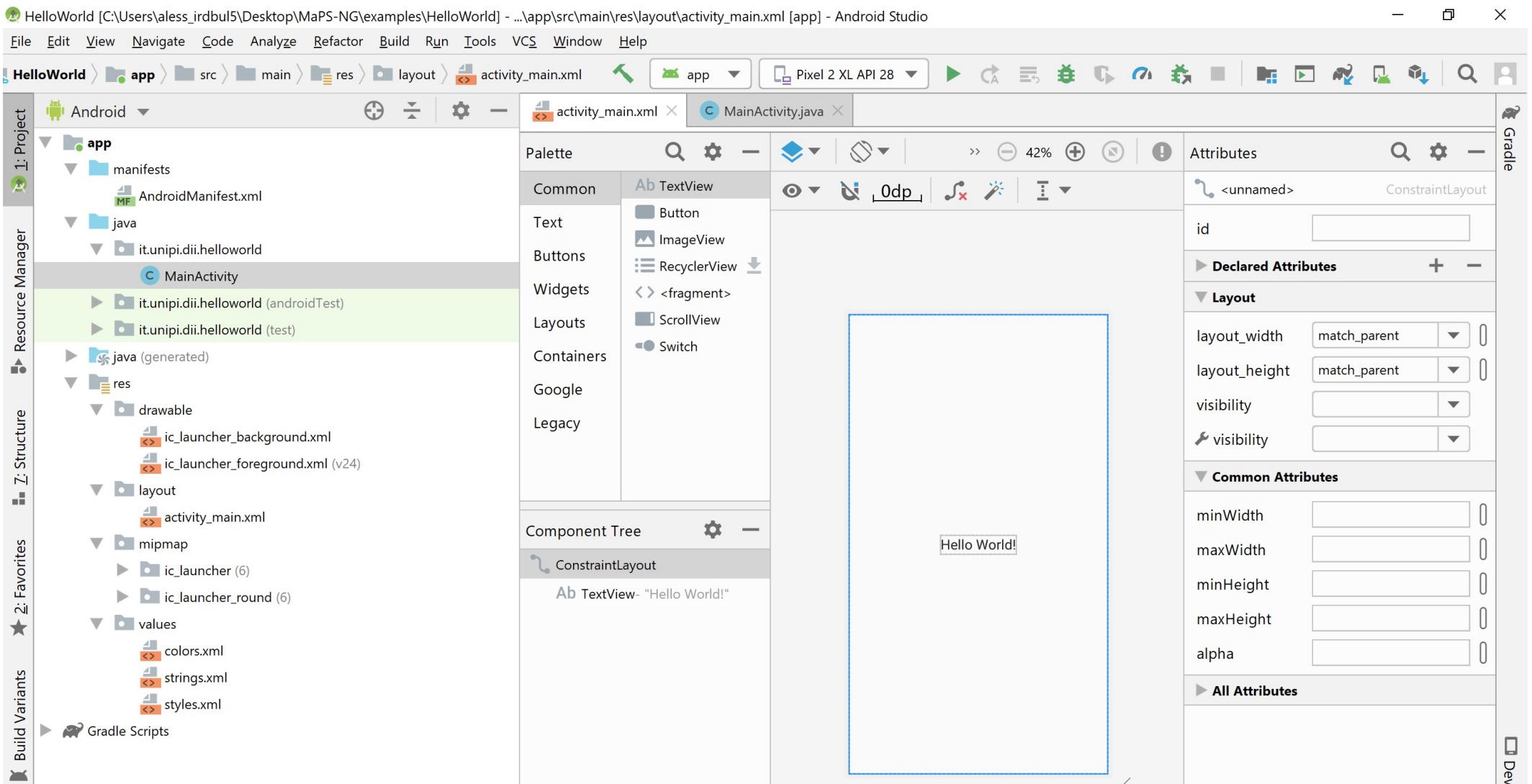


HelloWorld

The screenshot shows the Android Studio interface with the following details:

- Title Bar:** HelloWorld [C:\Users\ales... - ...app\src\main\java\it\unipi\dii\helloworld>MainActivity.java [app] - Android Studio
- Toolbar:** File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
- Project Navigators:** Project (1: Project), Resource Manager, Structure, Favorites, Build Variants, Gradle Scripts.
- Toolbars:** Standard toolbar with icons for file operations, search, and device selection.
- Editor:** The code editor displays the `MainActivity.java` file under the `activity_main.xml` tab. The code is as follows:package it.unipi.dii.helloworld;
import ...
public class MainActivity extends AppCompatActivity {
 @Override
 protected void onCreate(Bundle savedInstanceState) {
 super.onCreate(savedInstanceState);
 setContentView(R.layout.activity_main);
 }
}
- Device:** Pixel 2 XL API 28 selected in the top right.
- Gradle:** A green checkmark icon is visible on the right side of the editor area.

HelloWorld



HelloWorld: most important files

- The three most important files are
 - **MainActivity.java**: Java code, defines behavior, e.g. actions executed when button clicked
 - **activity_main.xml**: file that describes the UI in XML
 - **AndroidManifest.xml**:
 - Includes a list of all application components
 - In this case there is only one component
 - Specifies entry point of execution
- These files are automatically created

HelloWorld: MainActivity.java

- *Activity* is, basically, a screen
- An app may have many separate activities
 - but the user interacts with them one at a time
- *onCreate()* method:
 - will be called by the Android system when your *Activity* is started
 - where you should perform all initialization and UI setup

```
package it.unipi.dii.helloworld;  
  
import . . .  
  
public class MainActivity extends AppCompatActivity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
    }  
}
```

HelloWorld: activity_main.xml

- **Layout:** a container of UI elements
- **TextView:** a non-editable text box
- Behavior of layouts and appearance of widgets is defined by means of properties

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!"
        android:textAppearance="@style/TextAppearance.AppCompat.Large"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

HelloWorld: AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="it.unipi.dii.helloworld">

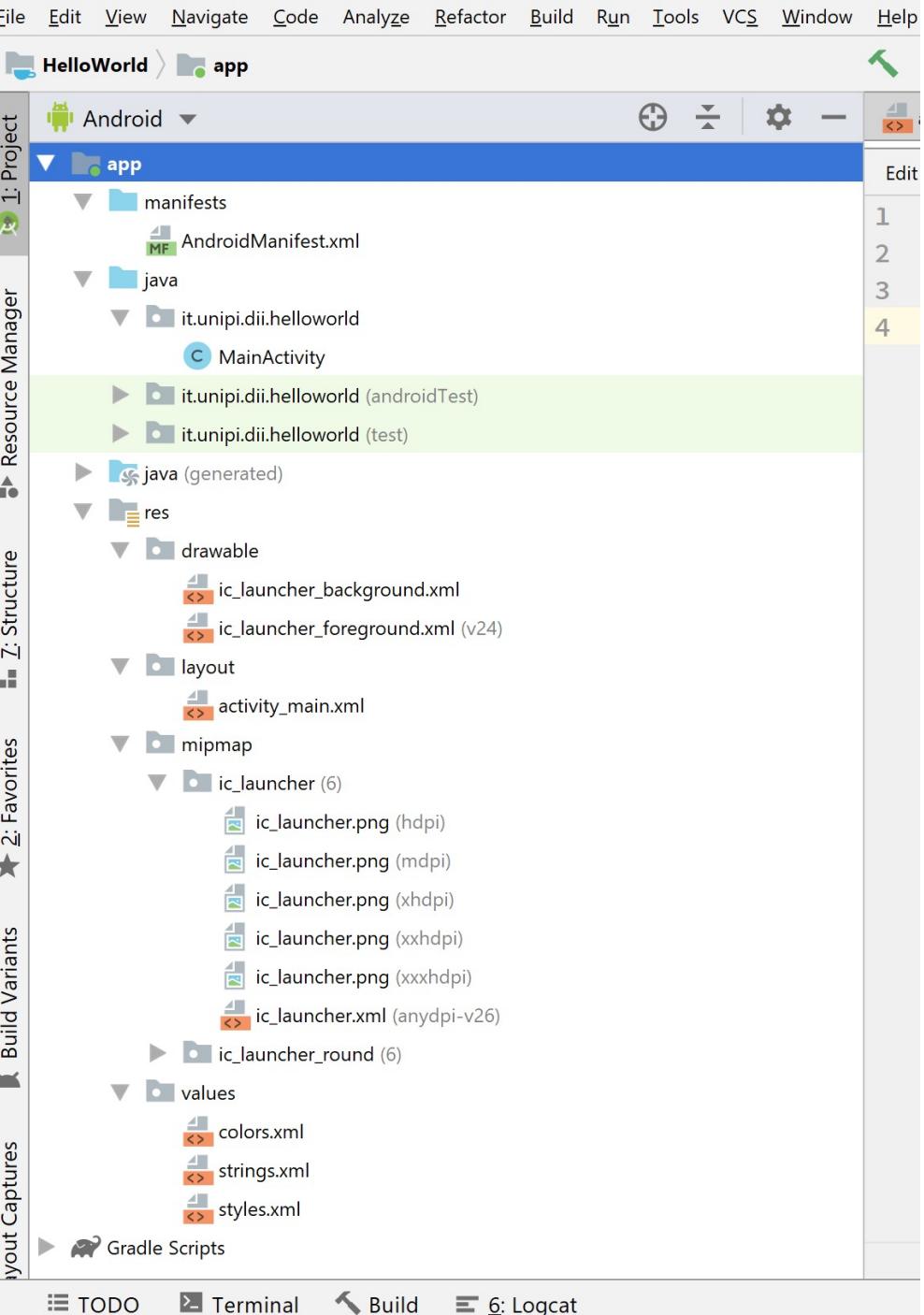
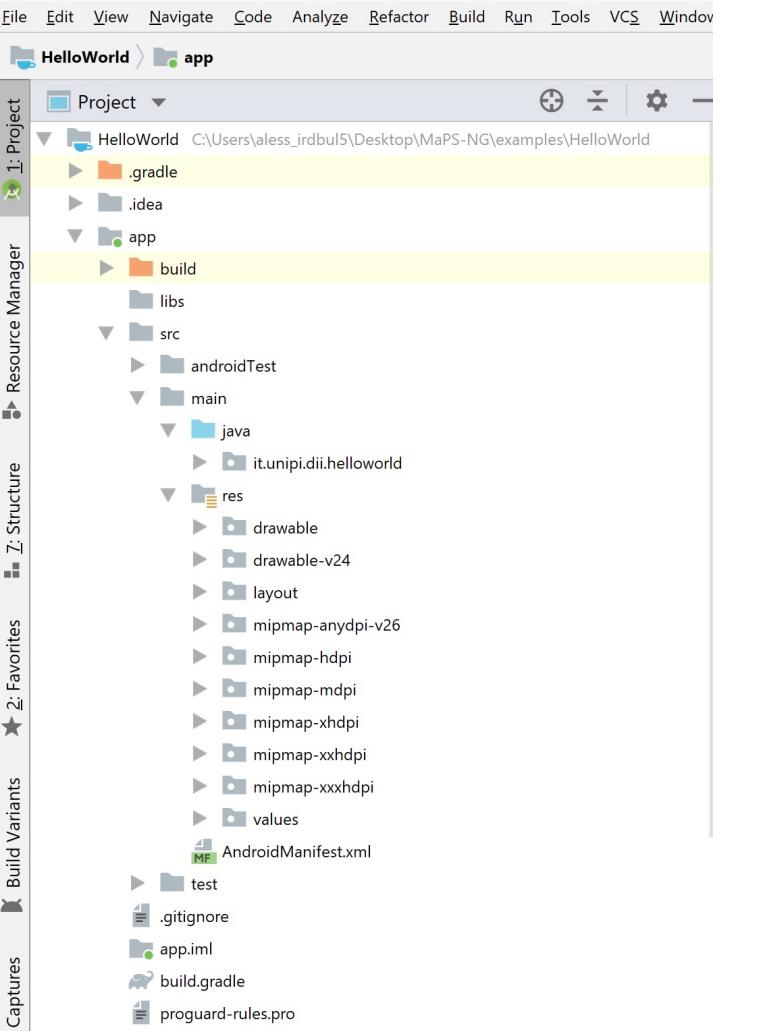
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

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

</manifest>
```

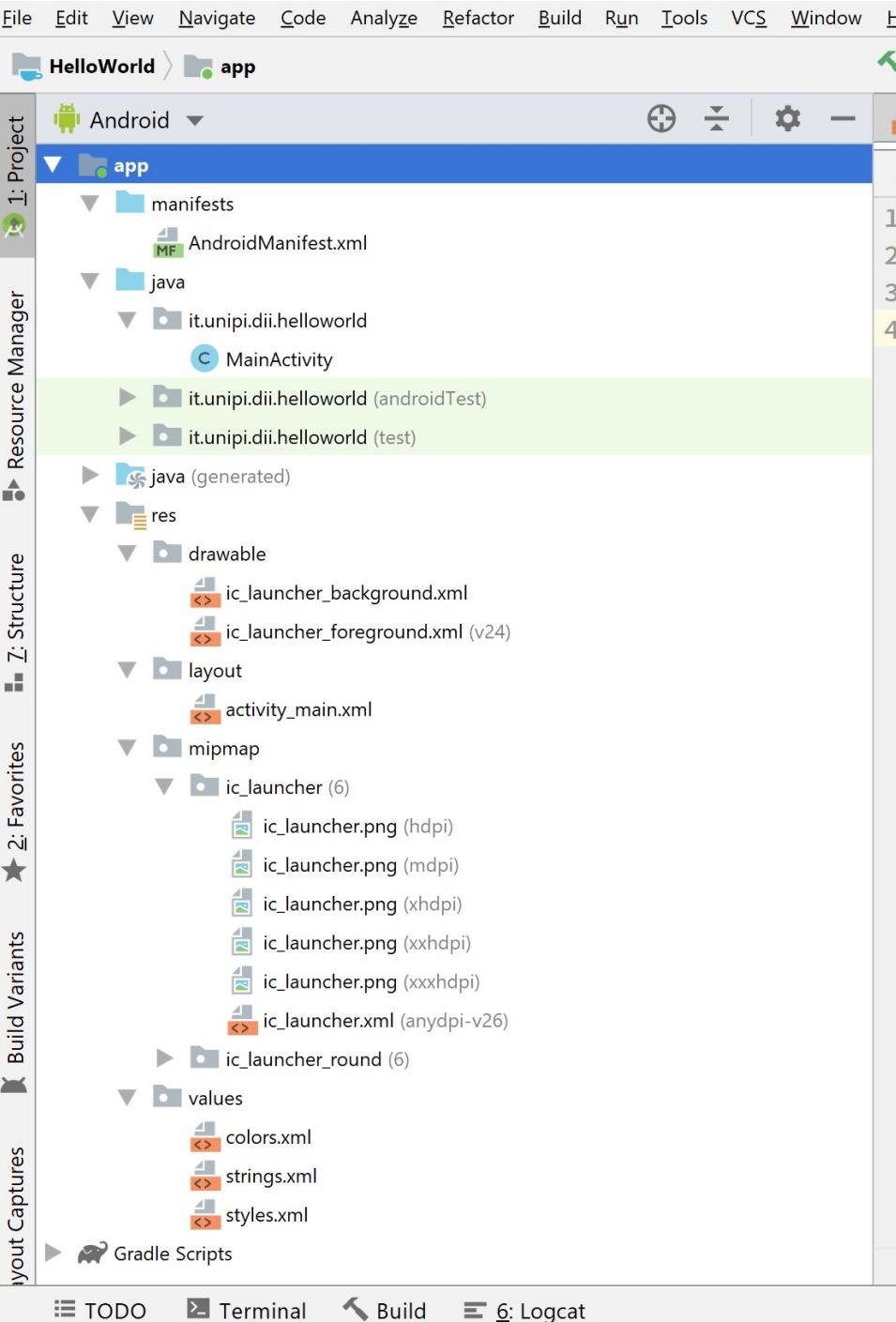
Project structure

- **java**: contains all Java source files
- **res**: contains all resources



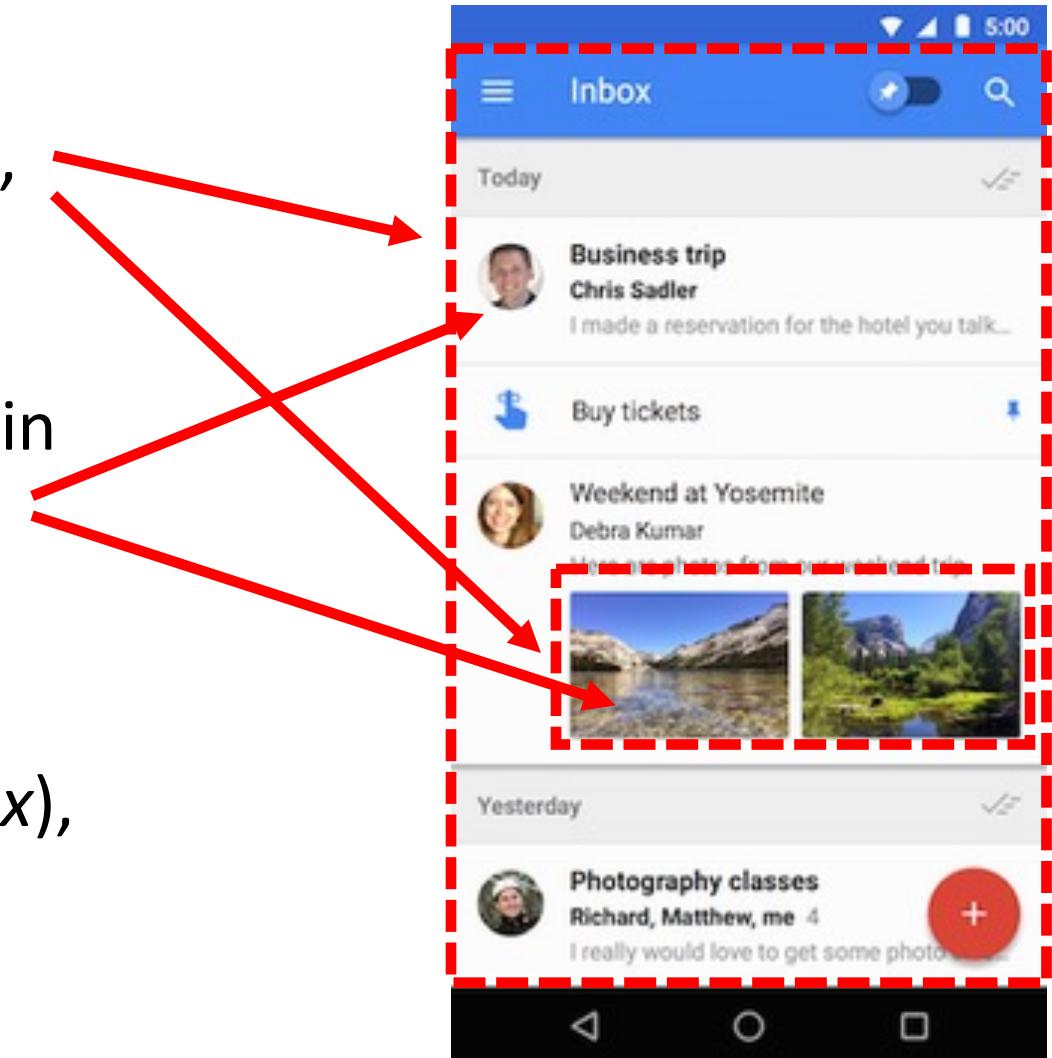
The *res* folder

- **res/** (resources) folder contains static resources you can embed in Android screen (e.g. images, 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



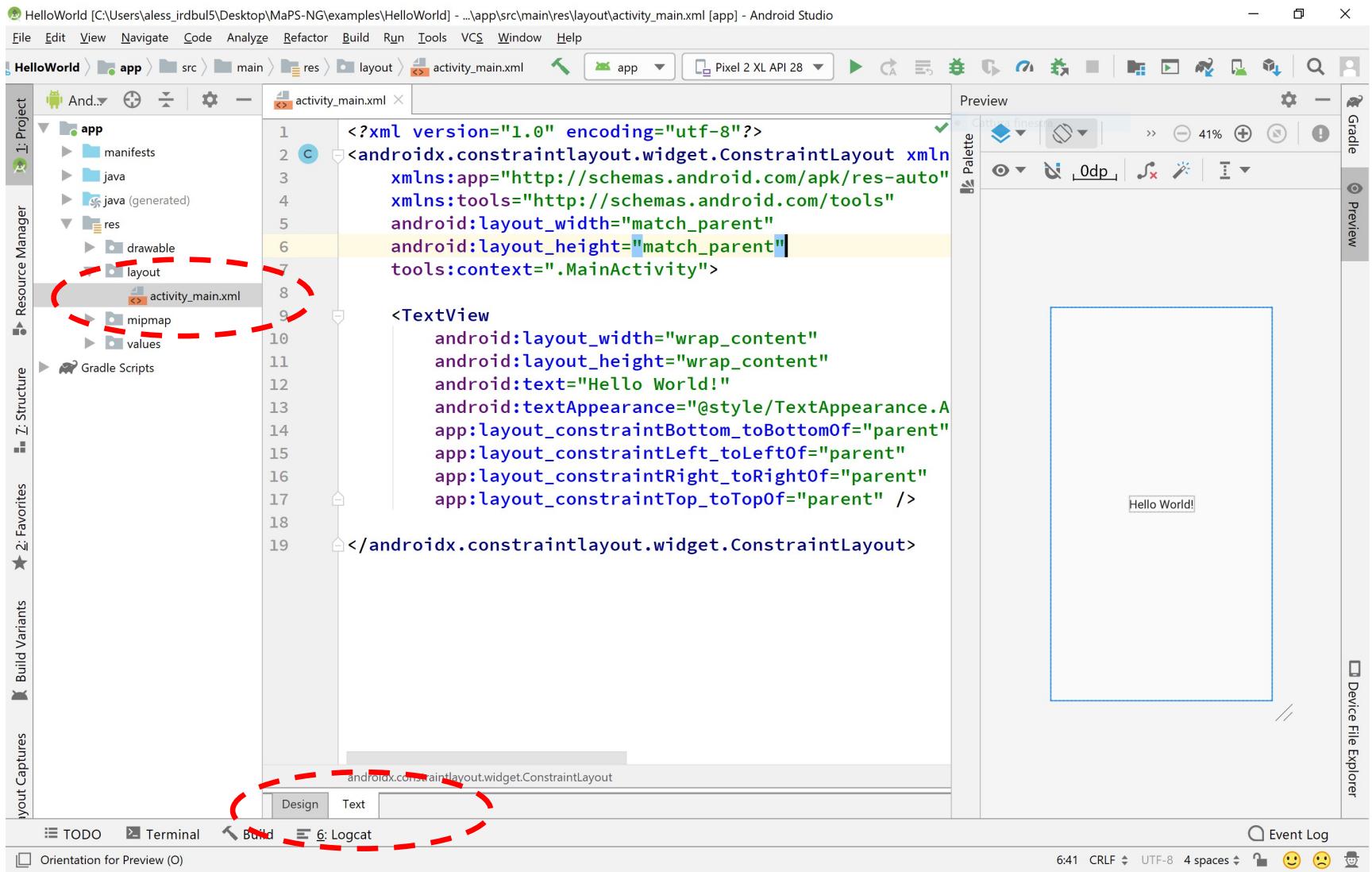
Files in Android project: an example

- **res/layout/**: layout, dimensions (width, height) of screen cells are specified in XML files here
- **res/drawable-xyz/**: The images stored in jpg or other format here
- **java/**: code to be executed when user clicks on a button
- **AndroidManifest.xml**: app name (*Inbox*), list of app screens, etc



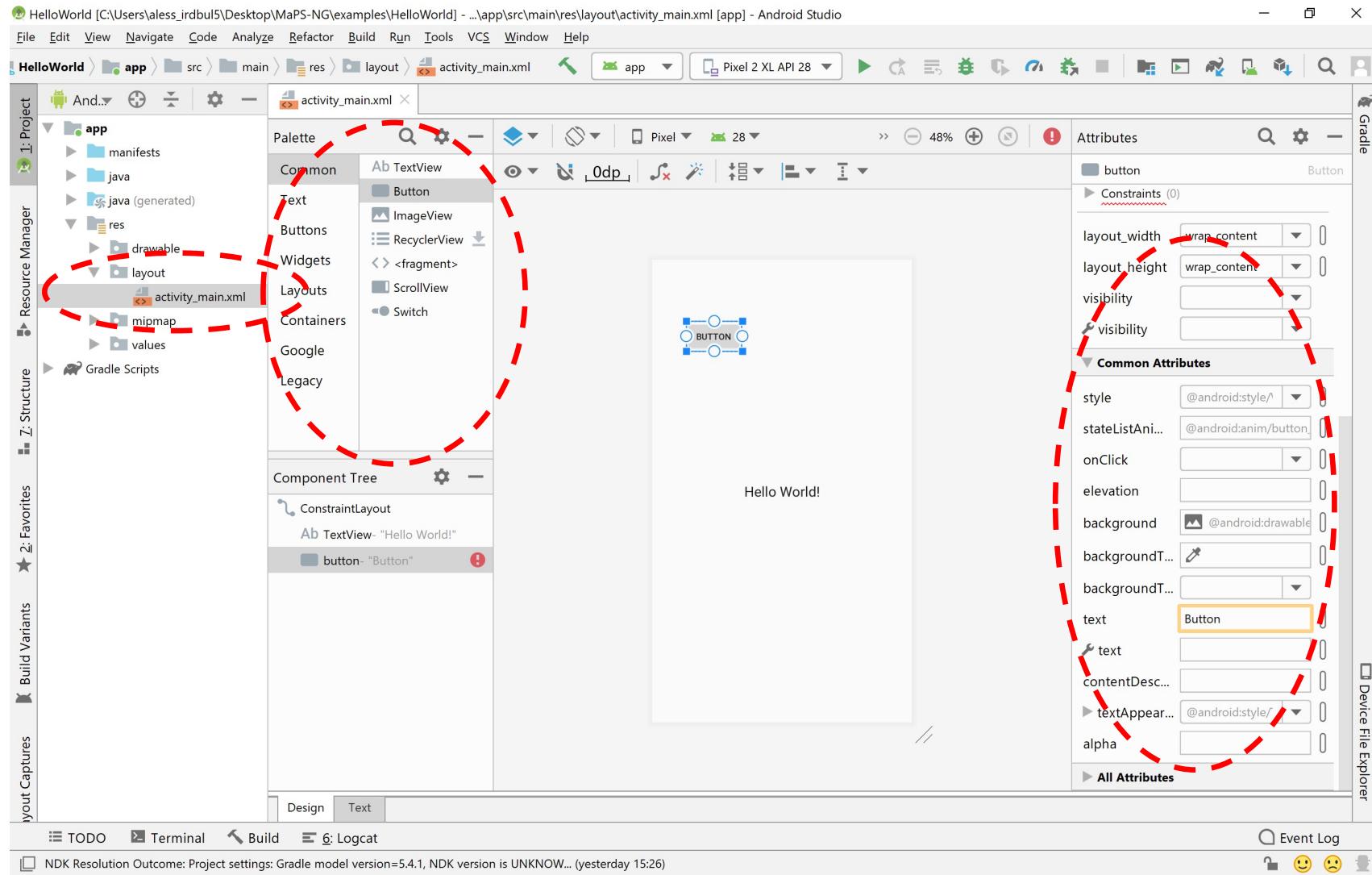
AndroidStudio: creating GUI

- First possibility:
edit XML files



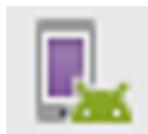
AndroidStudio: creating GUI

- Second possibility:
drag&drop
 - XML code is automatically generated



AVDs

- You can create as many AVDs as you want (e.g. corresponding to different real devices)
- Use the AVD manager



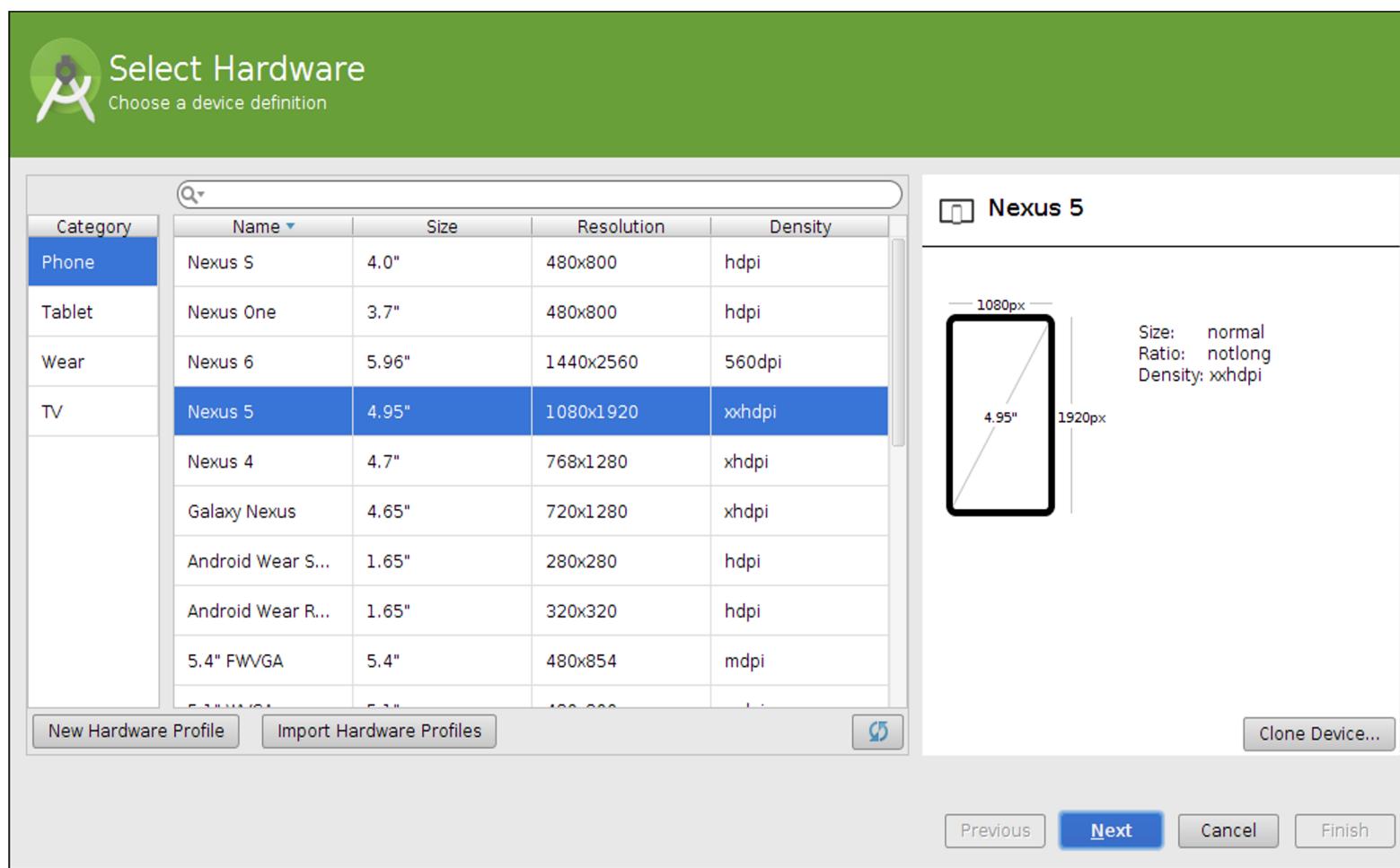
Your Virtual Devices
Android Studio

Type	Name	Resolution	API	Target	CPU/ABI	Size on Disk	Actions
Tablet	AVD for Nexus One ...	480 x 800: hdpi	10	Google APIs	arm	77 MB	
Watch	Android Wear Roun...	320 x 320: hdpi	21	Android 5.0.1	x86	1 GB	
Tablet	AVD for 10.1in WXG...	1280 x 800: mdpi	16	Android 4.1.2	arm	57 MB	
Tablet	MyAndroidKitKat	768 x 1280: xhdpi	19	Google APIs	arm	466 MB	
Watch	Android Wear Roun...	320 x 320: hdpi	21	Android 5.0.1	arm	650 MB	
Tablet	Nexus 5 API 21 x86	1080 x 1920: xxhdpi	21	Google APIs	x86	1 GB	

[+ Create Virtual Device...](#)

AVDs

- When creating an AVD, you can select category, device type, resolution



AVDs

- ... system image
(API level,
processor
architecture)

System Image
Select a system image

Release Name	API Level ▾	ABI	Target
Lollipop	21	x86	Google APIs (Google Inc.) - google_
Lollipop Download	21	armeabi-v7a	Android SDK Platform 5.0
Lollipop Download	21	x86_64	Android SDK Platform 5.0
Lollipop Download	21	x86	Android SDK Platform 5.0
Lollipop Download	21	armeabi-v7a	System Image armeabi-v7a with Go
Lollipop Download	21	x86_64	System Image x86_64 with Google
KitKat	19	armeabi-v7a	Android 4.4.2
KitKat	19	armeabi-v7a	Google APIs (Google Inc.)
KitKat Download	19	x86	Android SDK Platform 4.4.2
Jelly Bean Download	18	armeabi-v7a	Android SDK Platform 4.3
Jelly Bean Download	18	x86	Android SDK Platform 4.3
Jelly Bean	17	armeabi-v7a	Android 4.2.2
Jelly Bean Download	17	x86	Android SDK Platform 4.2
Jelly Bean Download	17	mips	Android 4.2.1

Show downloadable system images

Lollipop

API Level
21

Android
5.0.1

Google Inc.

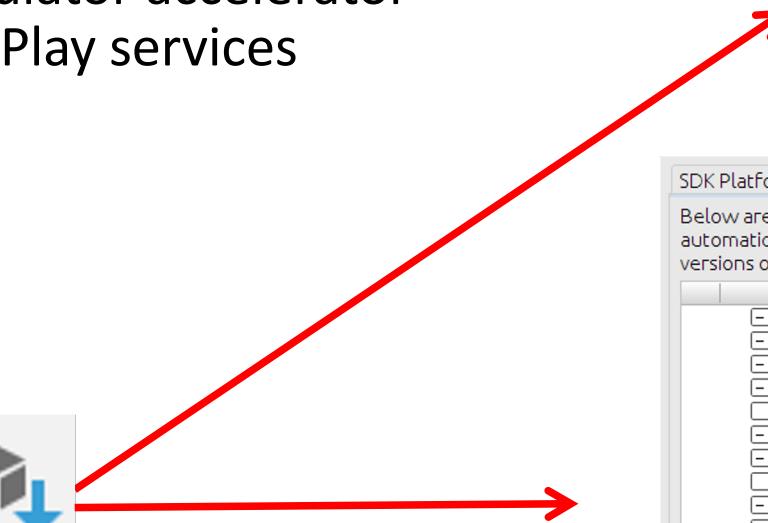
System Image
x86

? - See documentation for Android 5 APIs

Previous Next Cancel Finish

Android SDK Manager

- SDK Manager allows you to install build-tools, system images, Android APIs, Google proprietary APIs, extras
- Extras include Google USB driver (Win), x86 emulator accelerator (Win), Google Play services



SDK Platforms SDK Tools SDK Update Sites				
Each Android SDK Platform package includes the Android platform and sources pertaining to an API level by default. Once installed, Android Studio will automatically check for updates. Check "show package details" to display individual SDK components.				
Name	API Level	Revision	Status	
<input type="checkbox"/> Android 6.0	23	2	Not installed	
<input checked="" type="checkbox"/> Android 5.1.1	22	2	Update available	
<input checked="" type="checkbox"/> Android 5.0.1	21	2	Update available	
<input type="checkbox"/> Android 4.4.W.2	20	2	Not installed	
<input checked="" type="checkbox"/> Android 4.4.2	19	4	Update available	
<input type="checkbox"/> Android 4.3.1	18	3	Not installed	
<input type="checkbox"/> Android 4.2.2	17	3	Update available	
<input checked="" type="checkbox"/> Android 4.1.2	16	5	Update available	
<input type="checkbox"/> Android 4.0.3	15	5	Not installed	
<input checked="" type="checkbox"/> Android 4.0	14	4	Installed	
<input checked="" type="checkbox"/> Android 2.3.3	10	2	Update available	
<input type="checkbox"/> Android 2.2	8	3	Not installed	

Show Package Details

SDK Platforms SDK Tools SDK Update Sites		
Below are the available SDK developer tools. Once installed, Android Studio will automatically check for updates. Check "show package details" to display available versions of an SDK Tool.		
Name	Version	Status
<input type="checkbox"/> Android SDK Build Tools		Update Available: 23.0.2
<input type="checkbox"/> Android SDK Tools 24.1.2	24.1.2	Update Available: 24.4.1
<input type="checkbox"/> Android SDK Platform-Tools 22	22.0.0	Update Available: 23.1.0
<input type="checkbox"/> Documentation for Android SDK	1	Update Available: 1
<input type="checkbox"/> CPU Debugging tools	1.0.3	Not installed
<input type="checkbox"/> Android Support Repository, rev 14	14.0.0	Update Available: 26
<input type="checkbox"/> Android Support Library, rev 22.1.1	22.1.1	Update Available: 23.2.0
<input type="checkbox"/> Android Auto Desktop Head Unit emulator	1.1.0	Not installed
<input type="checkbox"/> Google Play services, rev 23	23.0.0	Update Available: 29
<input type="checkbox"/> Google Repository, rev 16	16.0.0	Update Available: 24
<input type="checkbox"/> Google Play APK Expansion Library	3.0.0	Not installed
<input type="checkbox"/> Google Play Billing Library	5.0.0	Not installed
<input type="checkbox"/> Google Play Licensing Library	2.0.0	Not installed
<input type="checkbox"/> Android Auto API Simulators	1.0.0	Not installed
<input type="checkbox"/> Google Web Driver	2.0.0	Not installed
<input type="checkbox"/> Android NDK	1.0.0	Not installed
<input type="checkbox"/> LLDB	2.1.2589848	Not installed

References

- <http://developer.android.com>
- CS 528 Mobile and Ubiquitous Computing, WPI