CSE 2200

SOFTWARE DEVELOPMENT LAB- III

Lab 1

COURSE TEACHER

MD. AMINUR RAHMAN, ASSISTANT PROFESSOR

FATHIMA MIRZA, LECTURER

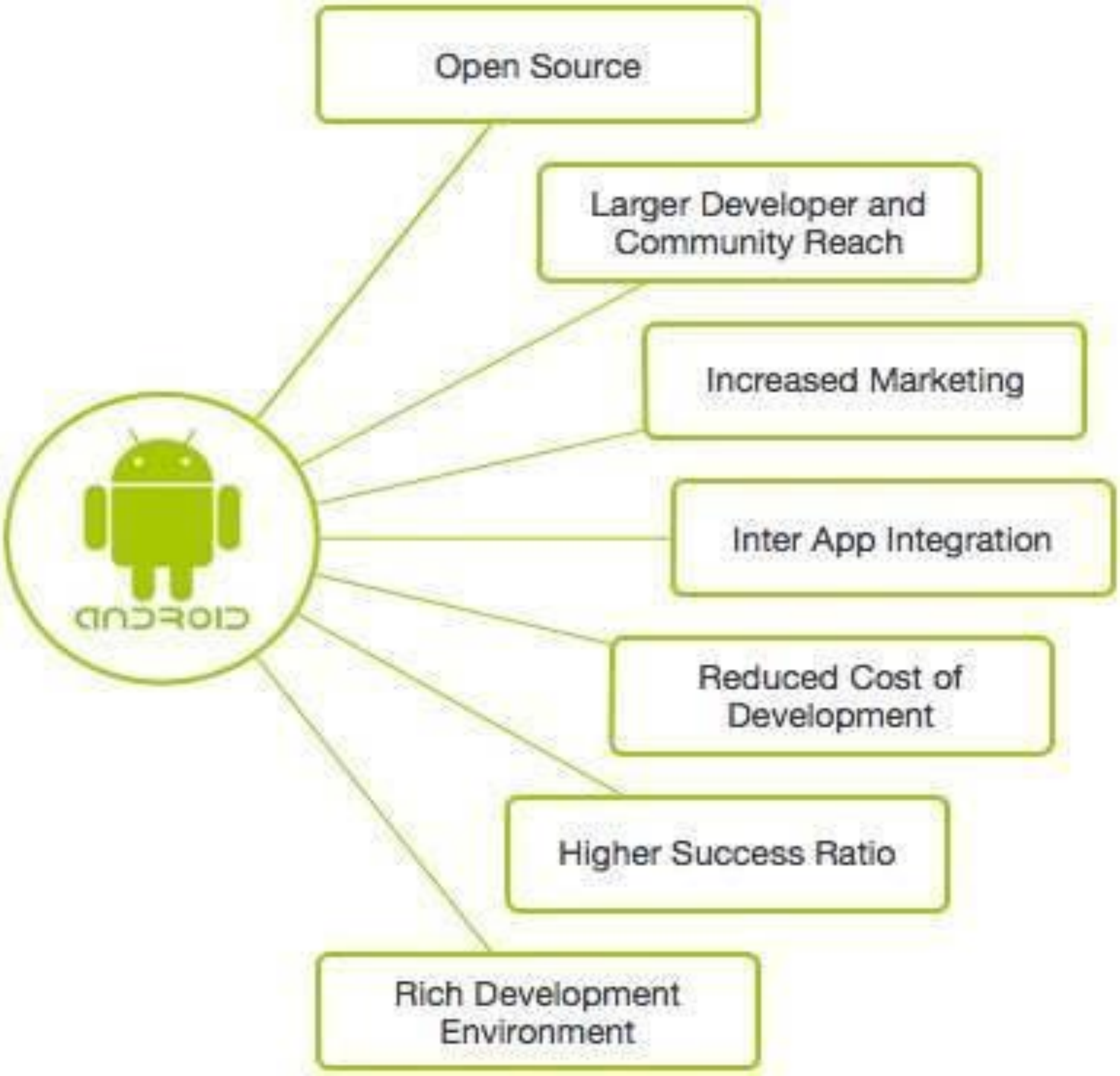
NOWSHIN NAWAR ARONY, LECTURER

Android is an open source and Linux-based operating system for mobile devices such as smartphones and tablet computers.

- Initially developed by Android Inc. by Andy Rubin, Rich Miner, Nick Sears, and Chris White in 2003.

- Which Google bought in 2005

- Now developed by an association of developers named Open Handset Alliance where the main contributor and commercial marketer is Google



Open Source

Larger Developer and Community Reach

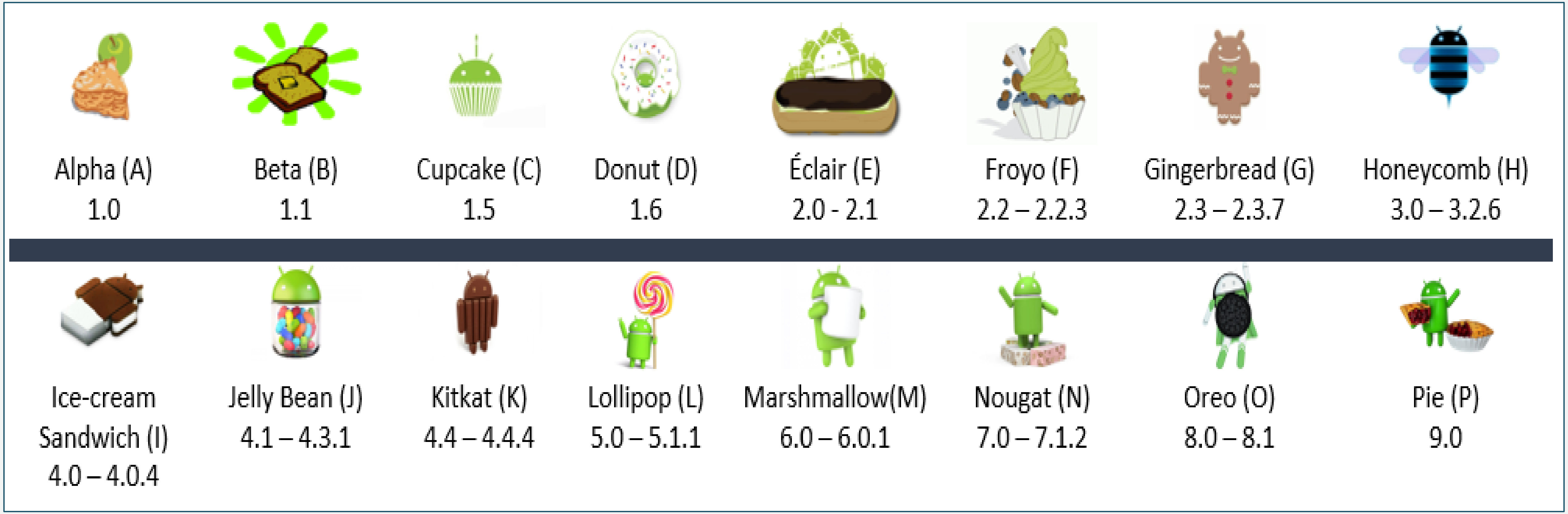
Increased Marketing

Inter App Integration

Reduced Cost of Development

Higher Success Ratio

Rich Development Environment



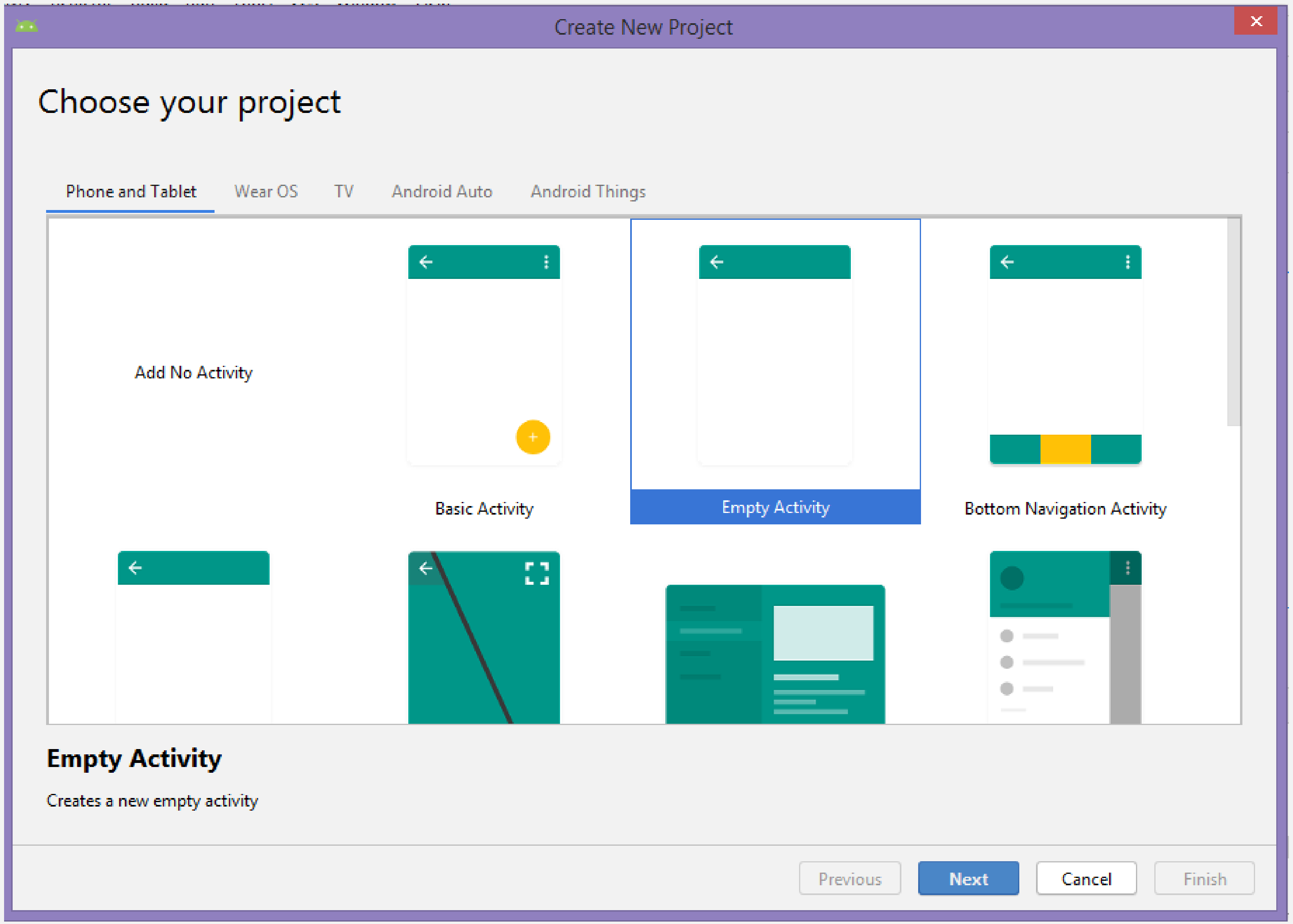
Alpha (A) 1.0	Beta (B) 1.1	Cupcake (C) 1.5	Donut (D) 1.6	Éclair (E) 2.0 - 2.1	Froyo (F) 2.2 – 2.2.3	Gingerbread (G) 2.3 – 2.3.7	Honeycomb (H) 3.0 – 3.2.6
Ice-cream Sandwich (I) 4.0 – 4.0.4	Jelly Bean (J) 4.1 – 4.3.1	Kitkat (K) 4.4 – 4.4.4	Lollipop (L) 5.0 – 5.1.1	Marshmallow(M) 6.0 – 6.0.1	Nougat (N) 7.0 – 7.1.2	Oreo (O) 8.0 – 8.1	Pie (P) 9.0

Latest Version: Android 10.0 (Q)
Released: September 3, 2019

Choosing Android Version

- Depending on your target customer.
- What features you want to integrate.
- An app developed in a lower version will run on higher versions. But if developed in an upper version won’t run on lower versions.

Creating your first application

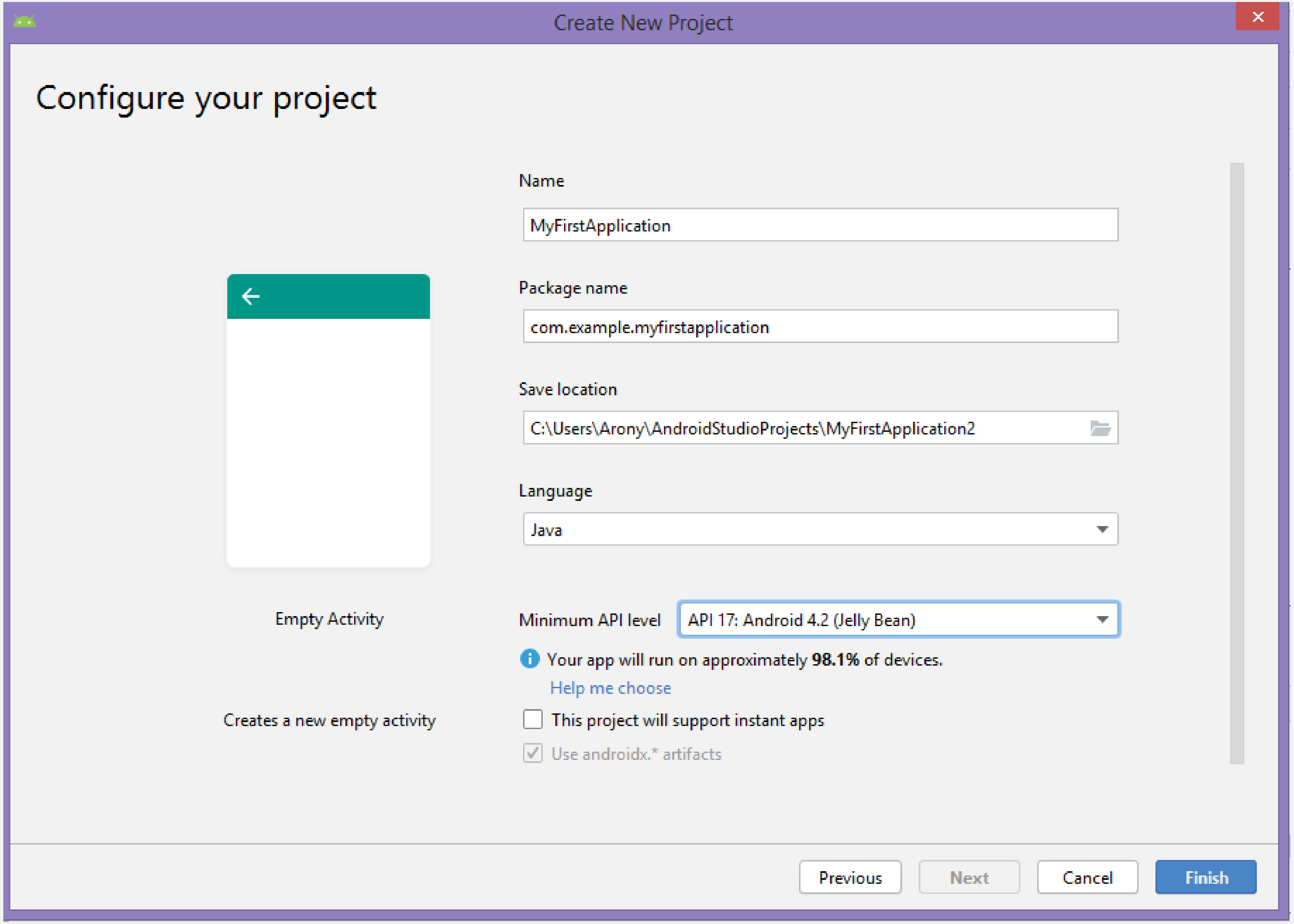


Create New Project

Choose your project

Empty Activity

Creates a new empty activity



Create New Project

Configure your project

Name

MyFirstApplication

Package name

com.example.myfirstapplication

Save location

C:\Users\Arony\AndroidStudioProjects\MyFirstApplication2

Language

Java

Minimum API level: API 17: Android 4.2 (Jelly Bean)

Your app will run on approximately **98.1%** of devices.

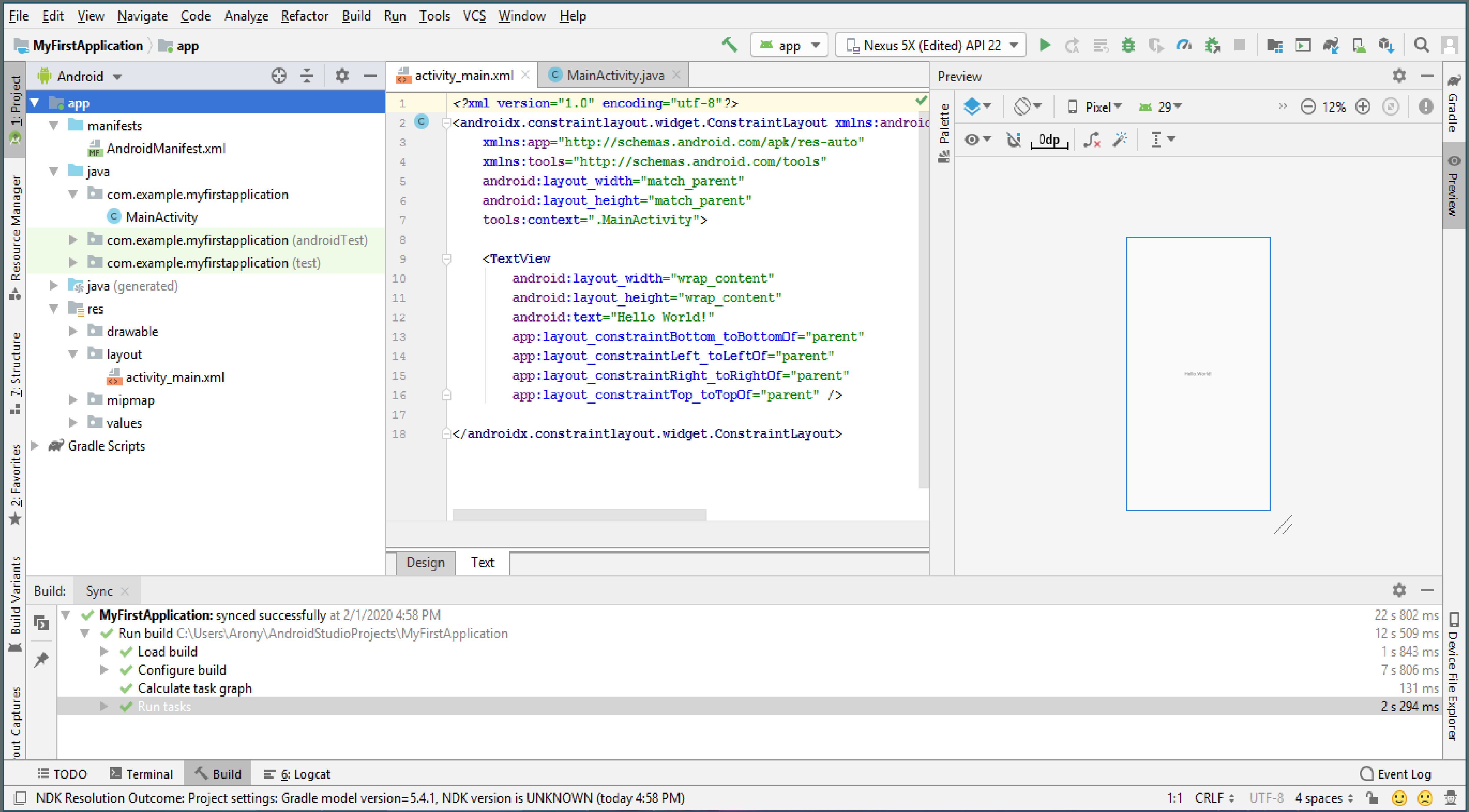
Help me choose

☐ This project will support instant apps

☑ Use androidx.* artifacts

Empty Activity

Creates a new empty activity



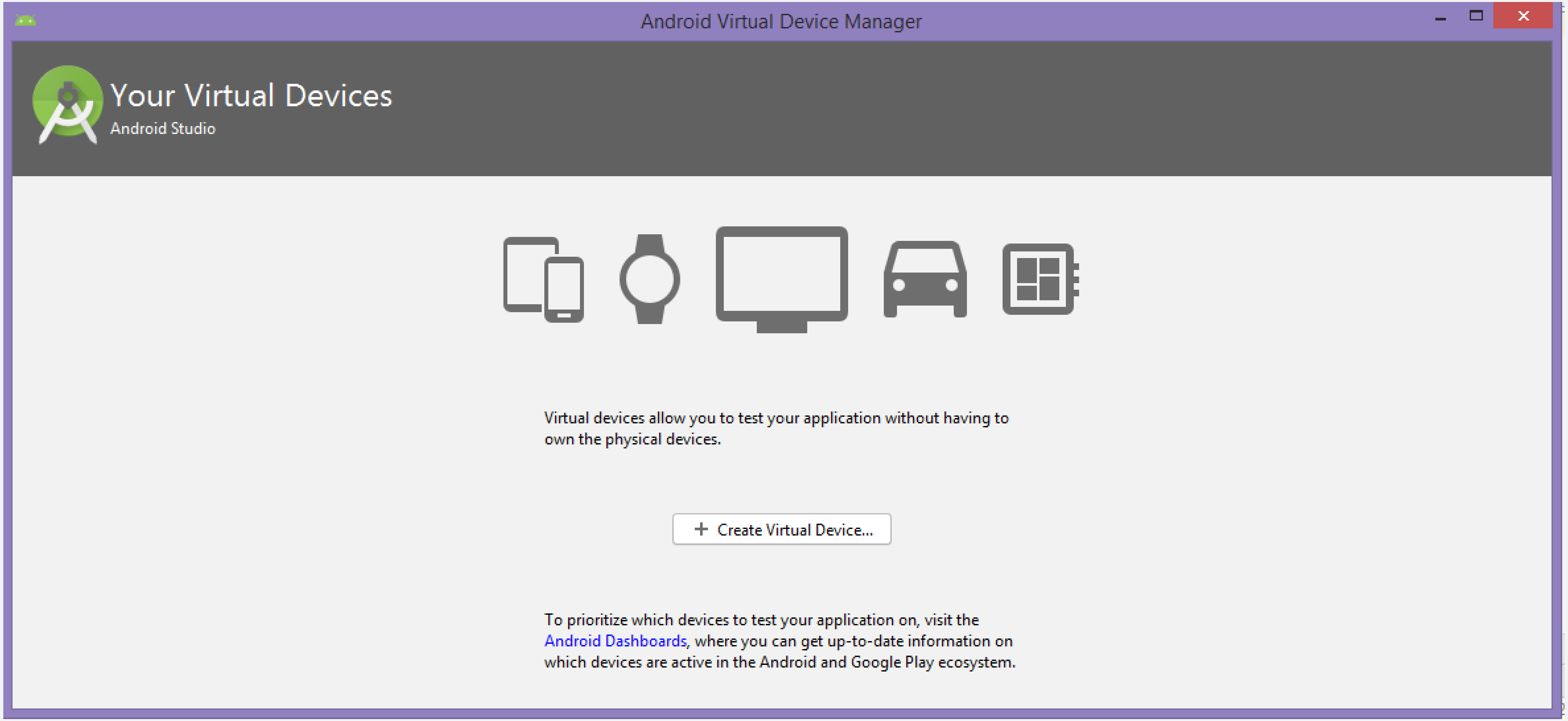
```xml
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:androic
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
 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

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

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

```
▼ ✓ MyFirstApplication: synced successfully at 2/1/2020 4:58 PM          22 s 802 ms
  ▼ ✓ Run build C:\Users\Arony\AndroidStudioProjects\MyFirstApplication    12 s 509 ms
    ▶ ✓ Load build                                                          1 s 843 ms
    ▶ ✓ Configure build                                                     7 s 806 ms
      ✓ Calculate task graph                                                   131 ms
    ▶ ✓ Run tasks                                                           2 s 294 ms
```

```
NDK Resolution Outcome: Project settings: Gradle model version=5.4.1, NDK version is UNKNOWN (today 4:58 PM)    1:1  CRLF  UTF-8  4 spaces
```

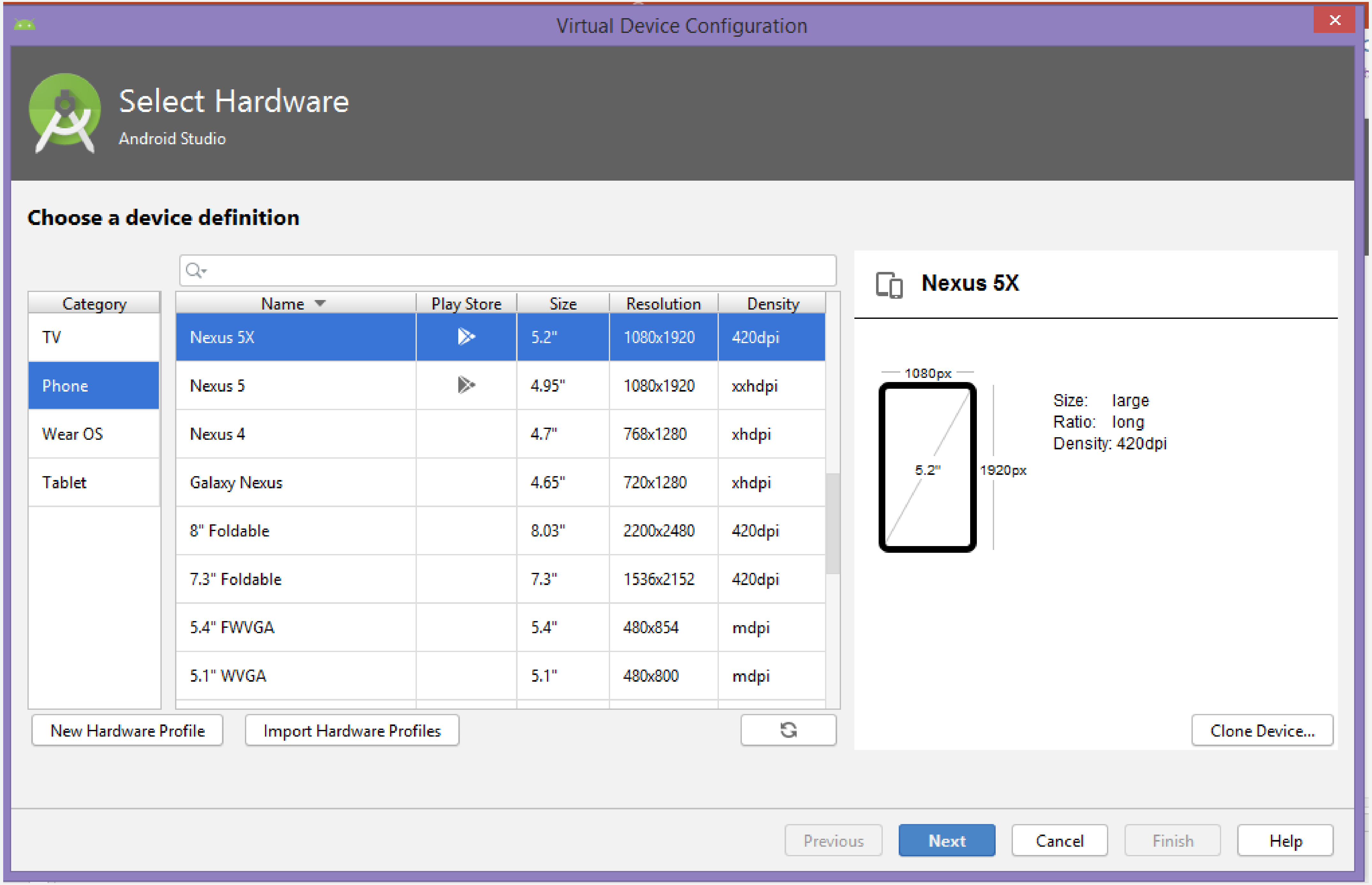
Creating a Virtual Device

- Tools -> AVD Manager



Virtual devices allow you to test your application without having to own the physical devices.

To prioritize which devices to test your application on, visit the Android Dashboards, where you can get up-to-date information on which devices are active in the Android and Google Play ecosystem.



Select Hardware

Android Studio

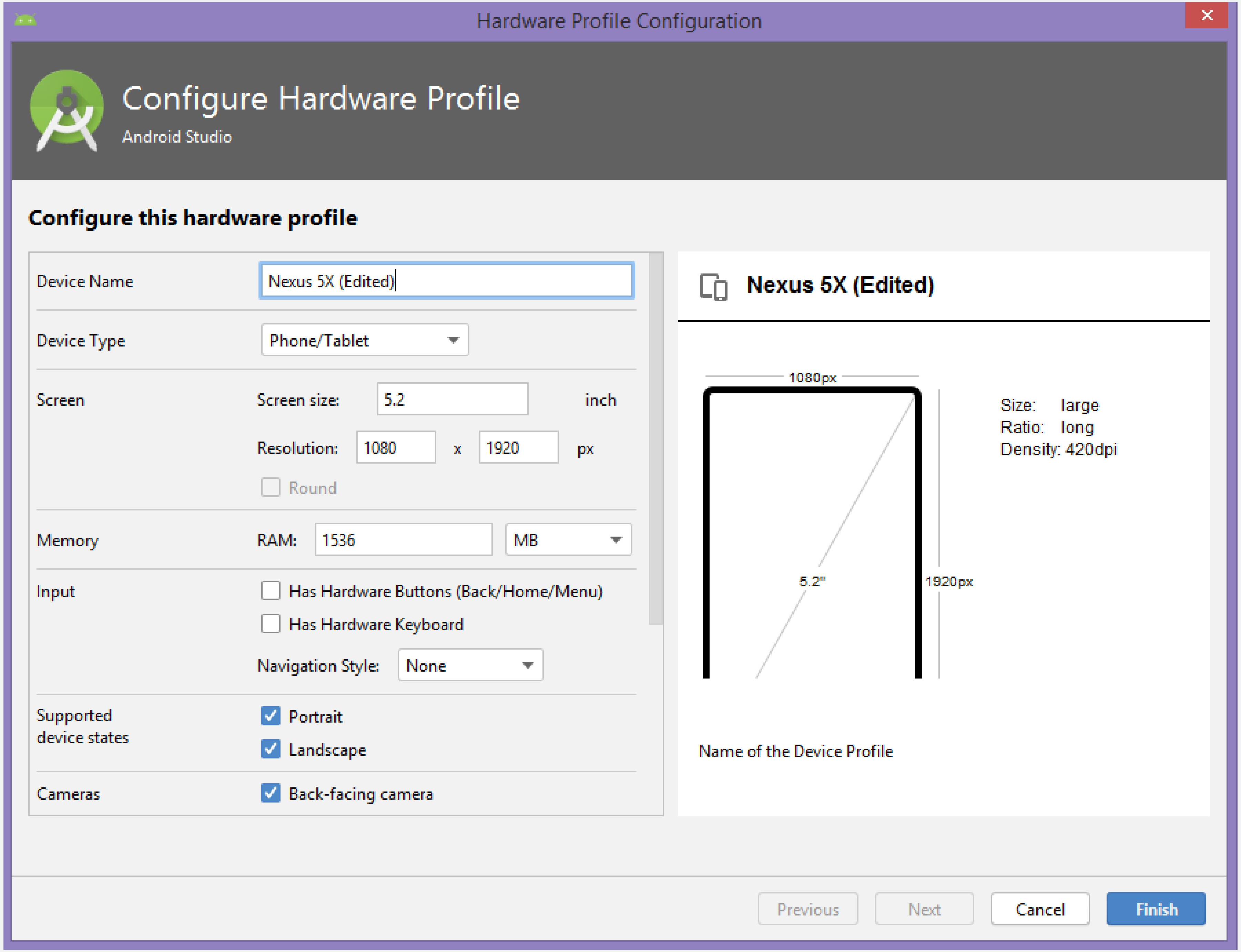
Choose a device definition

Category
TV
Phone
Wear OS
Tablet

Name ▼	Play Store	Size	Resolution	Density
Nexus 5X	▶	5.2"	1080x1920	420dpi
Nexus 5	▶	4.95"	1080x1920	xxhdpi
Nexus 4		4.7"	768x1280	xhdpi
Galaxy Nexus		4.65"	720x1280	xhdpi
8" Foldable		8.03"	2200x2480	420dpi
7.3" Foldable		7.3"	1536x2152	420dpi
5.4" FWVGA		5.4"	480x854	mdpi
5.1" WVGA		5.1"	480x800	mdpi

Nexus 5X

Size: large
Ratio: long
Density: 420dpi



Hardware Profile Configuration

Configure Hardware Profile

Android Studio

Configure this hardware profile

Device Name: Nexus 5X (Edited)

Device Type: Phone/Tablet

Screen

Screen size: 5.2 inch

Resolution: 1080 x 1920 px

□ Round

Memory

RAM: 1536 MB

Input

□ Has Hardware Buttons (Back/Home/Menu)

□ Has Hardware Keyboard

Navigation Style: None

Supported device states

☑ Portrait

☑ Landscape

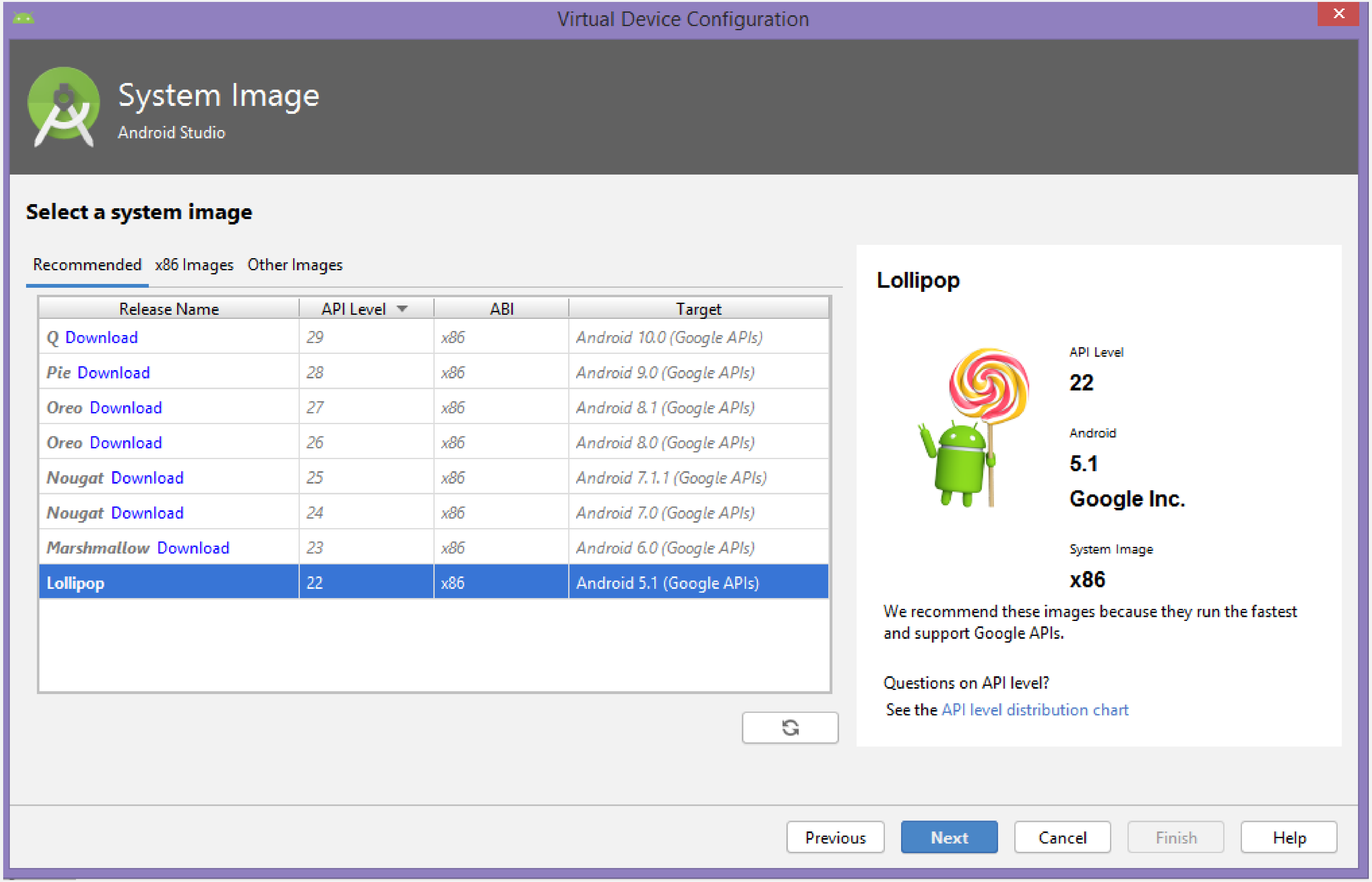
Cameras

☑ Back-facing camera

Nexus 5X (Edited)

Size: large
Ratio: long
Density: 420dpi

Name of the Device Profile



Select a system image

Release Name	API Level	ABI	Target
Q Download	*29*	*x86*	*Android 10.0 (Google APIs)*
Pie Download	*28*	*x86*	*Android 9.0 (Google APIs)*
Oreo Download	*27*	*x86*	*Android 8.1 (Google APIs)*
Oreo Download	*26*	*x86*	*Android 8.0 (Google APIs)*
Nougat Download	*25*	*x86*	*Android 7.1.1 (Google APIs)*
Nougat Download	*24*	*x86*	*Android 7.0 (Google APIs)*
Marshmallow Download	*23*	*x86*	*Android 6.0 (Google APIs)*
Lollipop	22	x86	Android 5.1 (Google APIs)

Lollipop

API Level

22

Android

5.1

Google Inc.

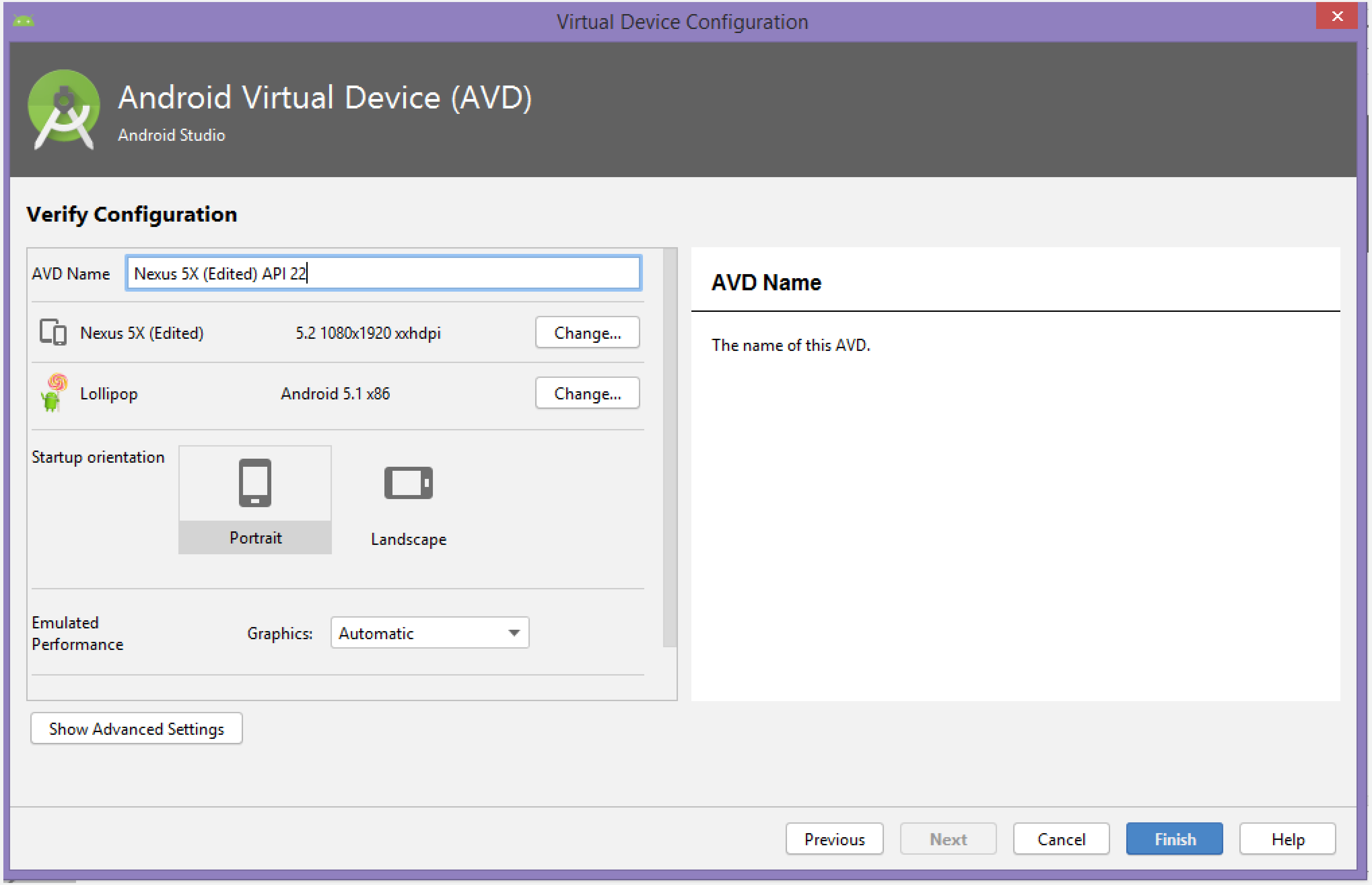
System Image

x86

We recommend these images because they run the fastest and support Google APIs.

Questions on API level?

See the API level distribution chart



Virtual Device Configuration

Android Virtual Device (AVD)

Android Studio

Verify Configuration

AVD Name: Nexus 5X (Edited) API 22

Nexus 5X (Edited) 5.2 1080x1920 xxhdpi

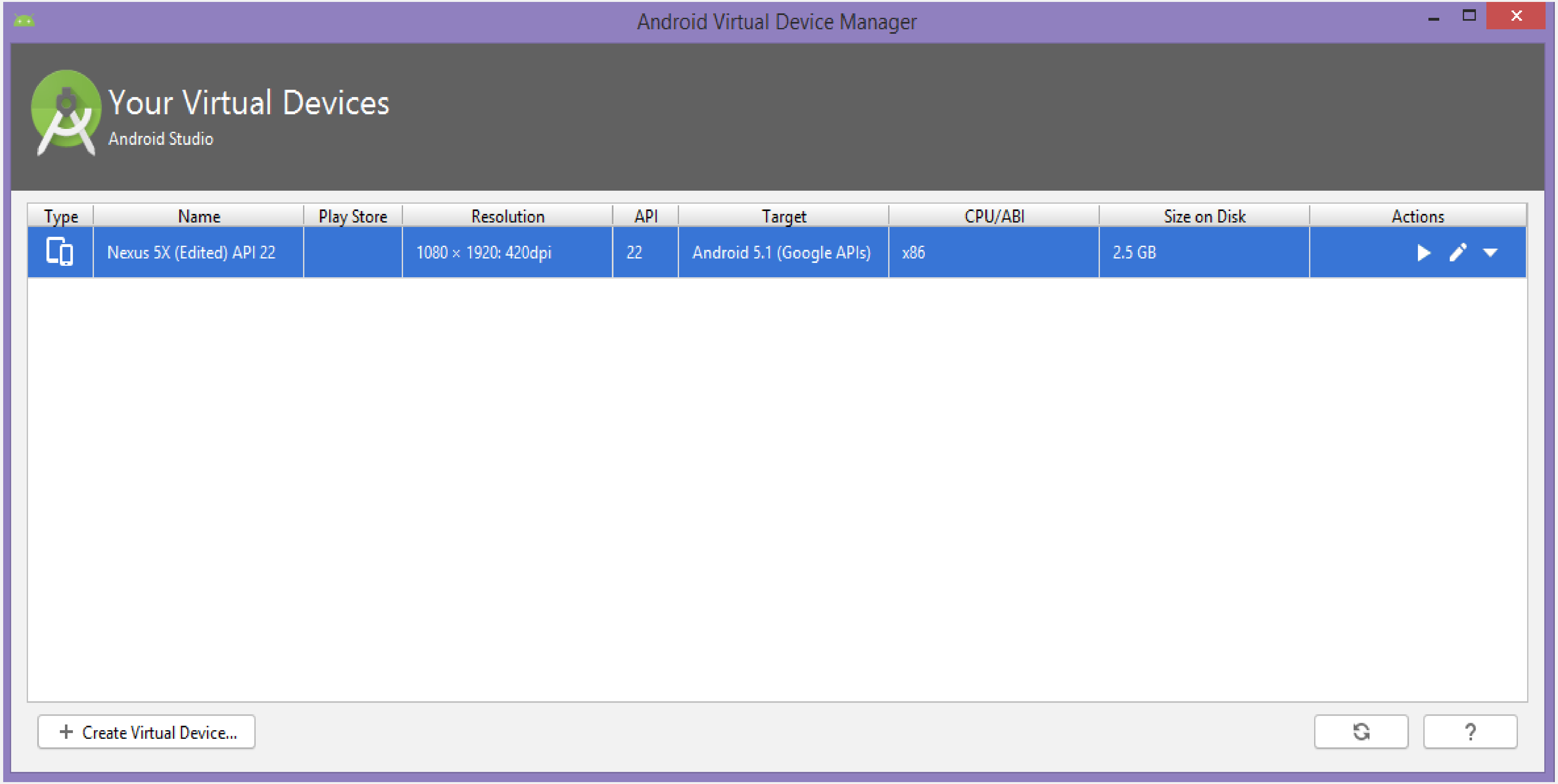
Lollipop Android 5.1 x86

Startup orientation: Portrait

Emulated Performance Graphics: Automatic

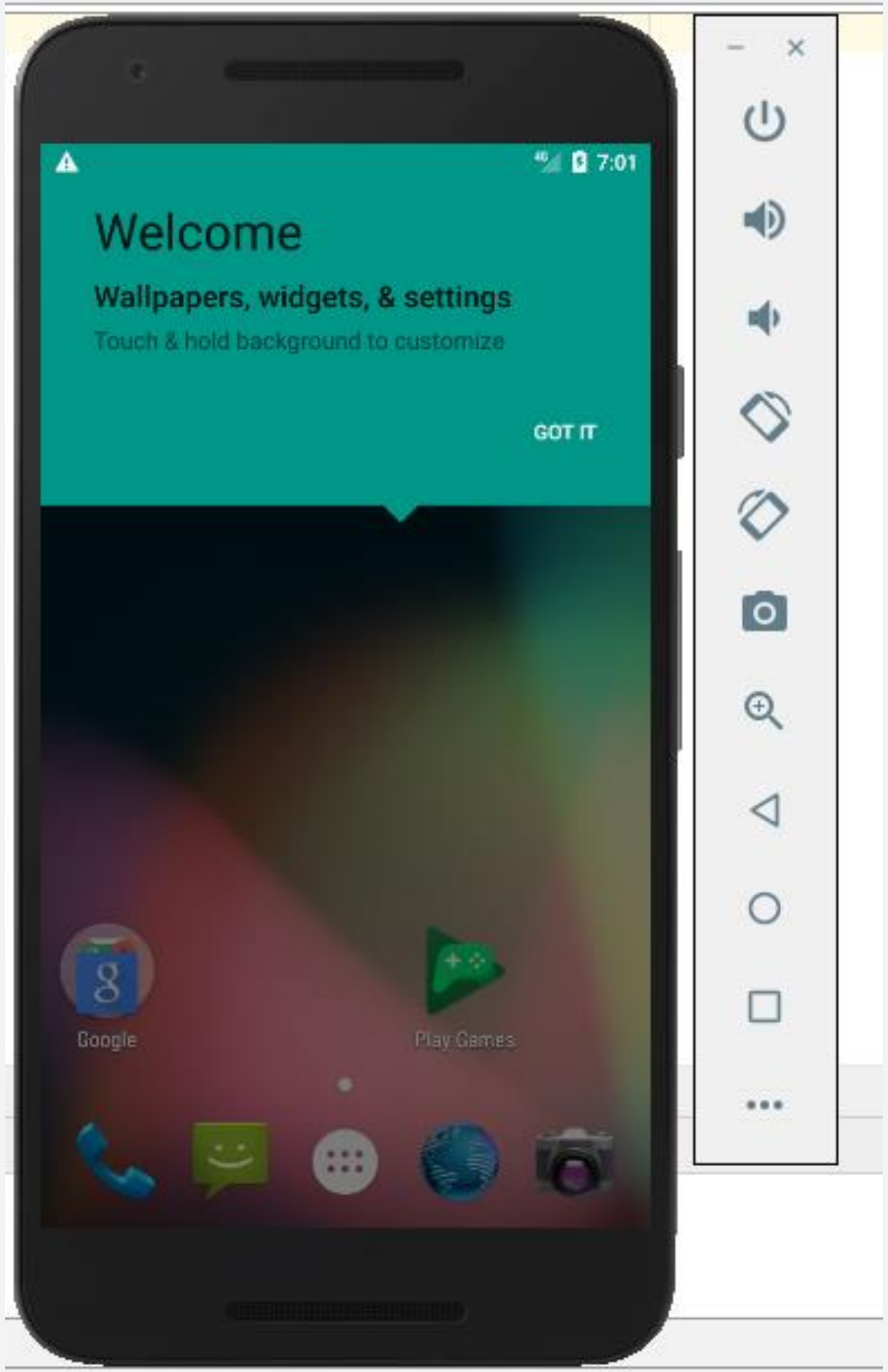
AVD Name

The name of this AVD.



Type	Name	Play Store	Resolution	API	Target	CPU/ABI	Size on Disk	Actions
	Nexus 5X (Edited) API 22		1080 × 1920: 420dpi	22	Android 5.1 (Google APIs)	x86	2.5 GB	▶ ✎ ▾

The Virtual Device is Now Ready



7:01

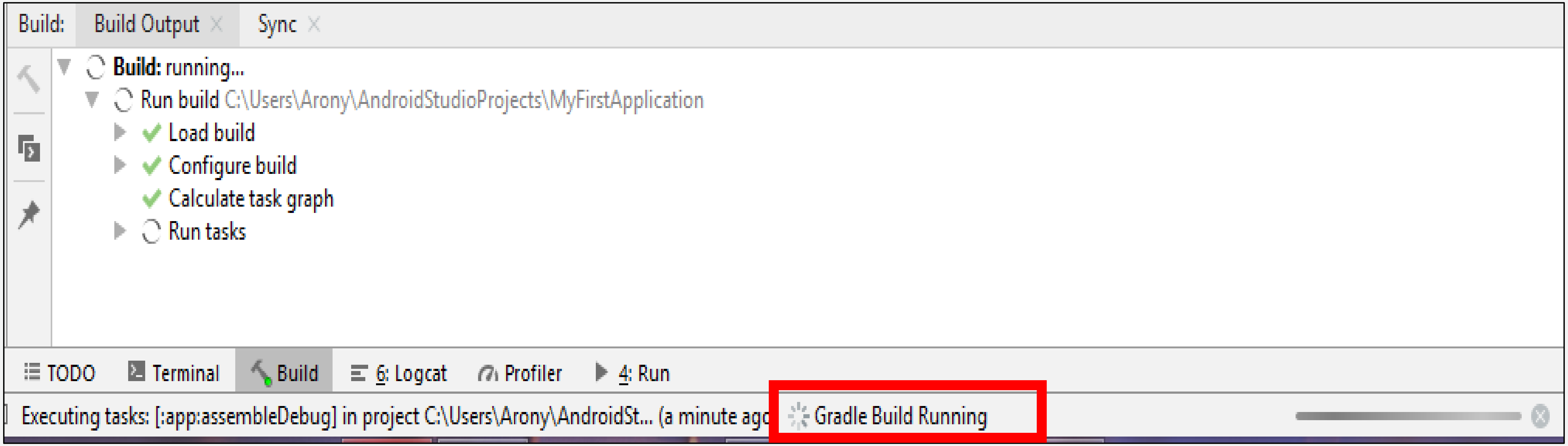
Welcome

Wallpapers, widgets, & settings

Touch & hold background to customize

GOT IT

Executing the project



Build: Build Output × Sync ×

Build: running...
Run build C:\Users\Arony\AndroidStudioProjects\MyFirstApplication
✓ Load build
✓ Configure build
✓ Calculate task graph
Run tasks

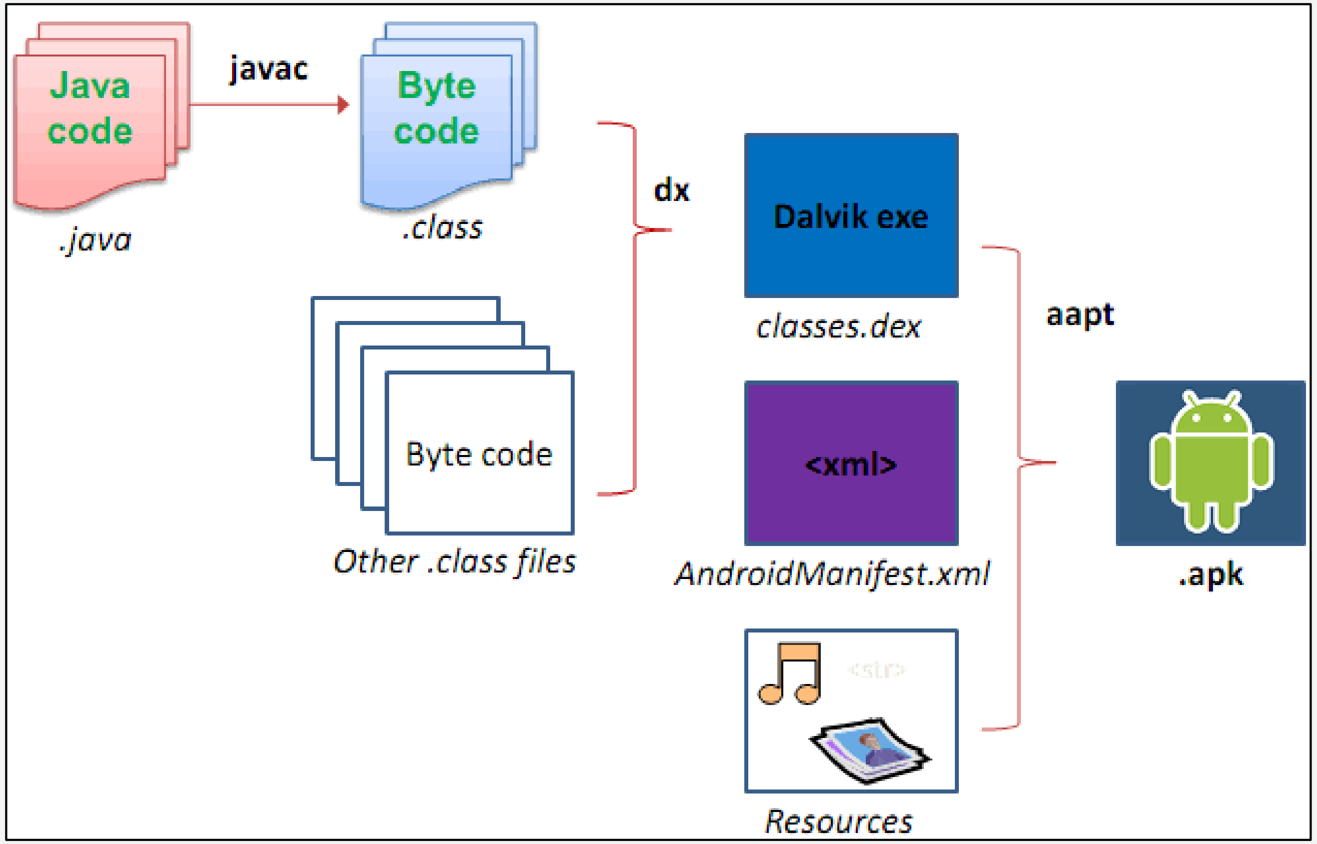
TODO Terminal Build 6: Logcat Profiler 4: Run

Executing tasks: [:app:assembleDebug] in project C:\Users\Arony\AndroidSt... (a minute ago

Gradle Build Running

Gradle

- *Gradle* is an open-source build automation tool that is designed to be flexible enough to build almost any type of software.



APK

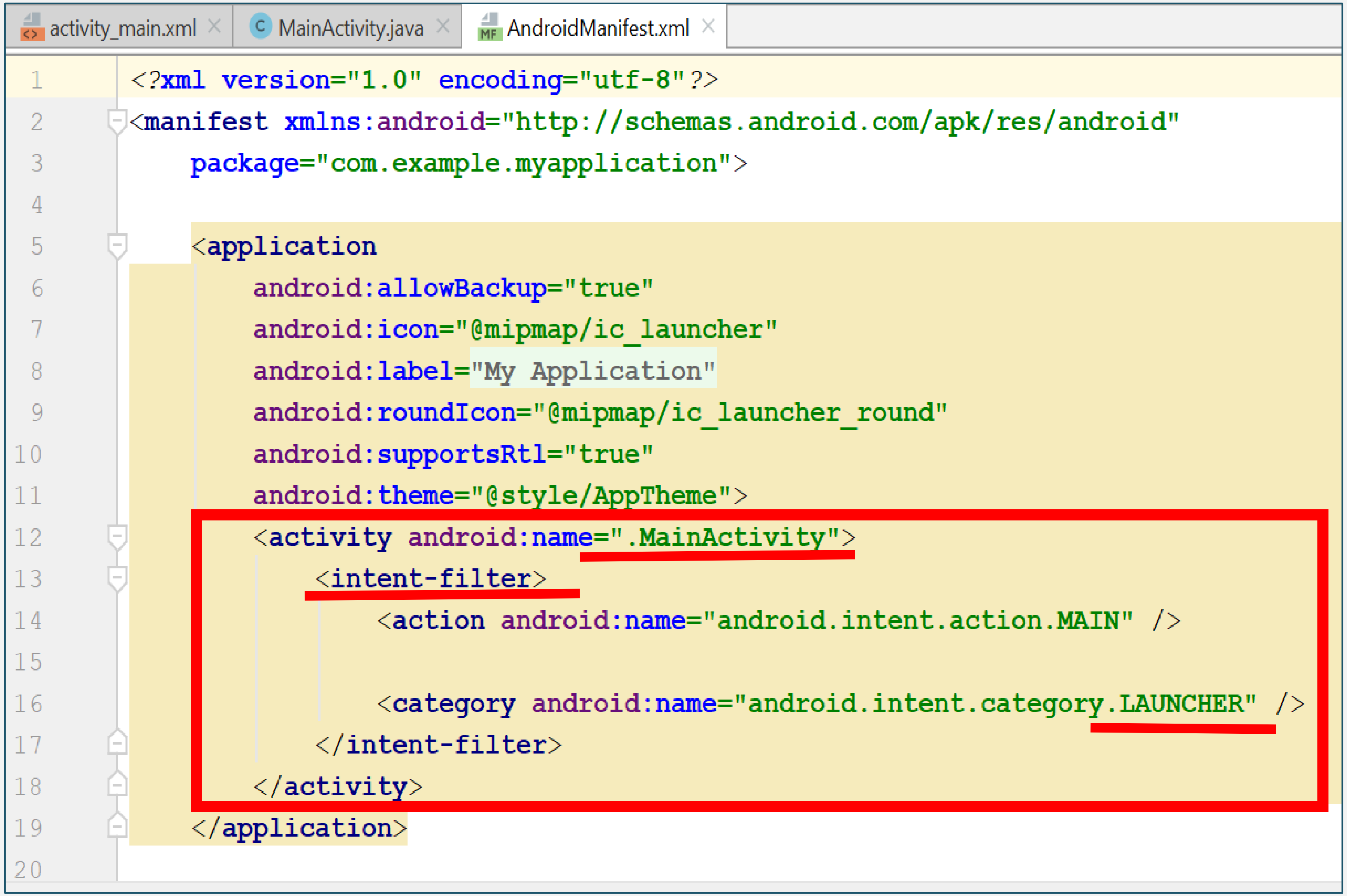
APK stands for ***“Android application package***”. Basically, it is an archive file that contains every component needed for an Android application to install.

The Application that run on android devices have the extension .apk

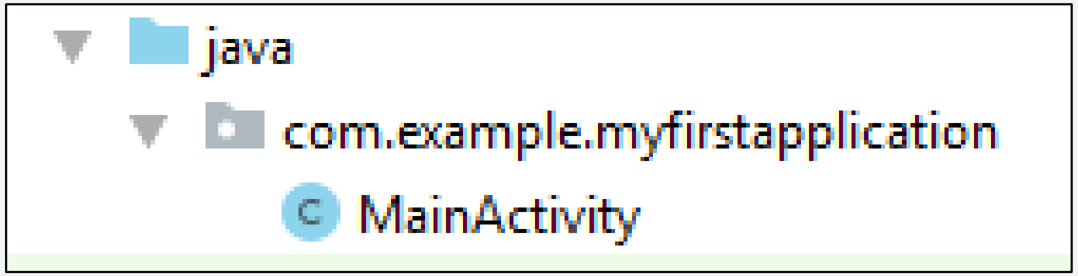
Android File Structure

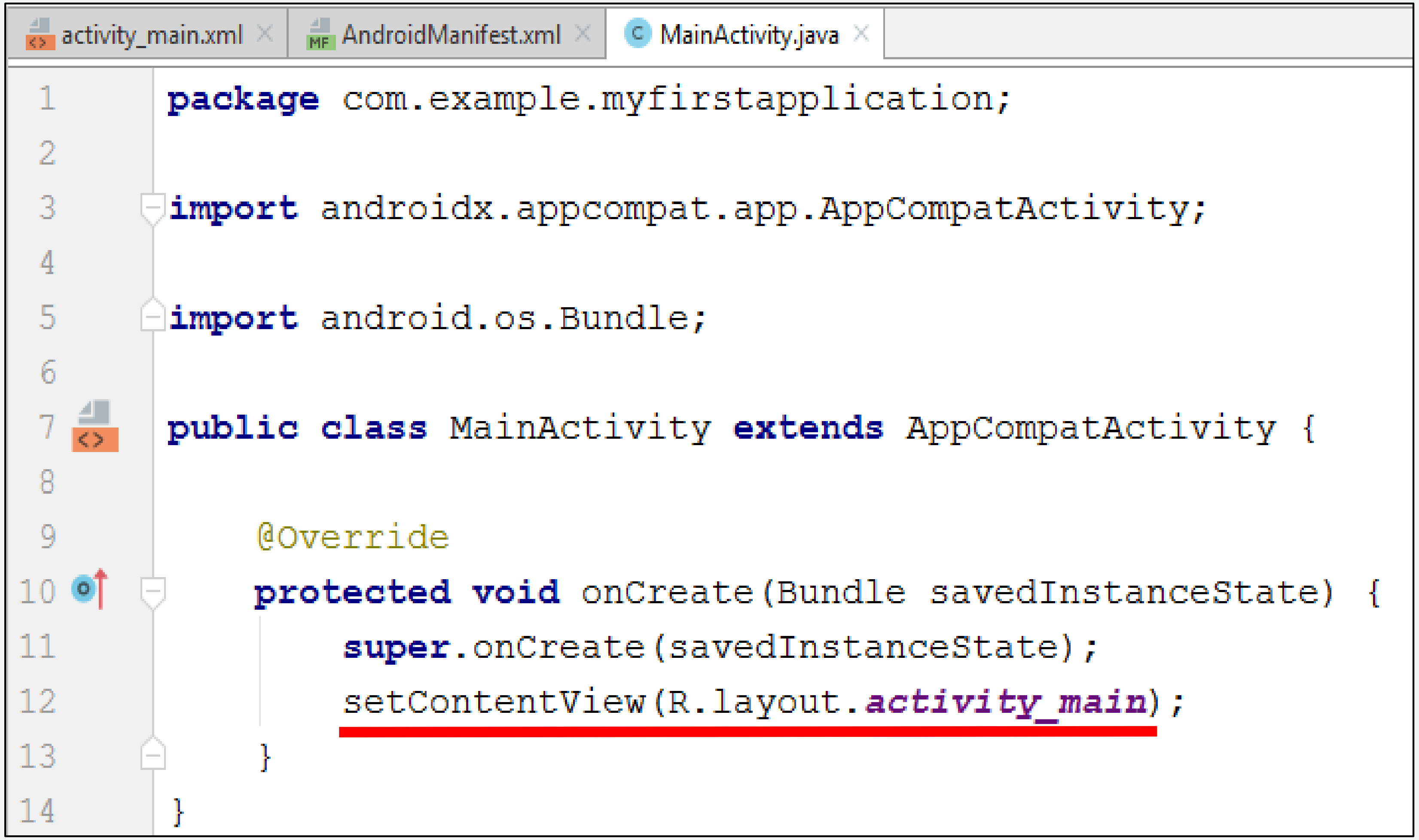
- Manifest
  - AndroidManifest.xml

When project is run, execution starts from here.



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

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="My Application"
        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>

```

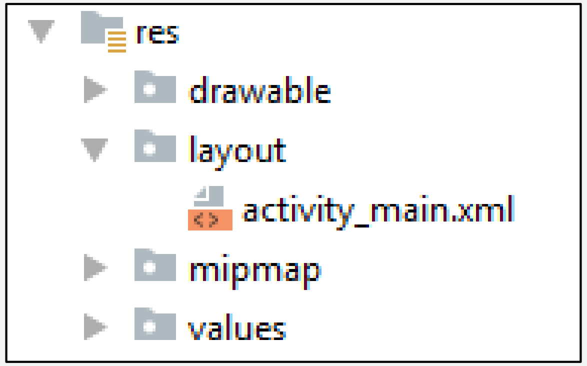


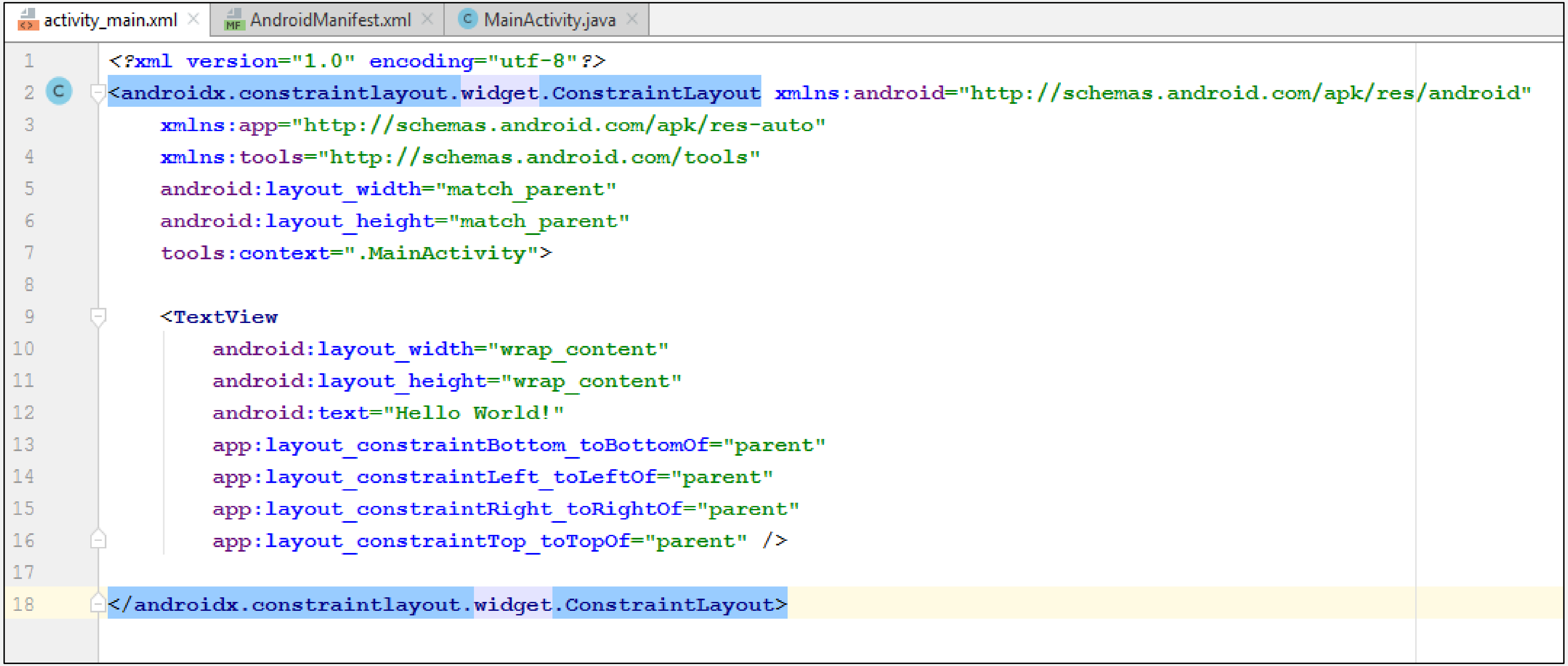


```java
package com.example.myfirstapplication;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}
```

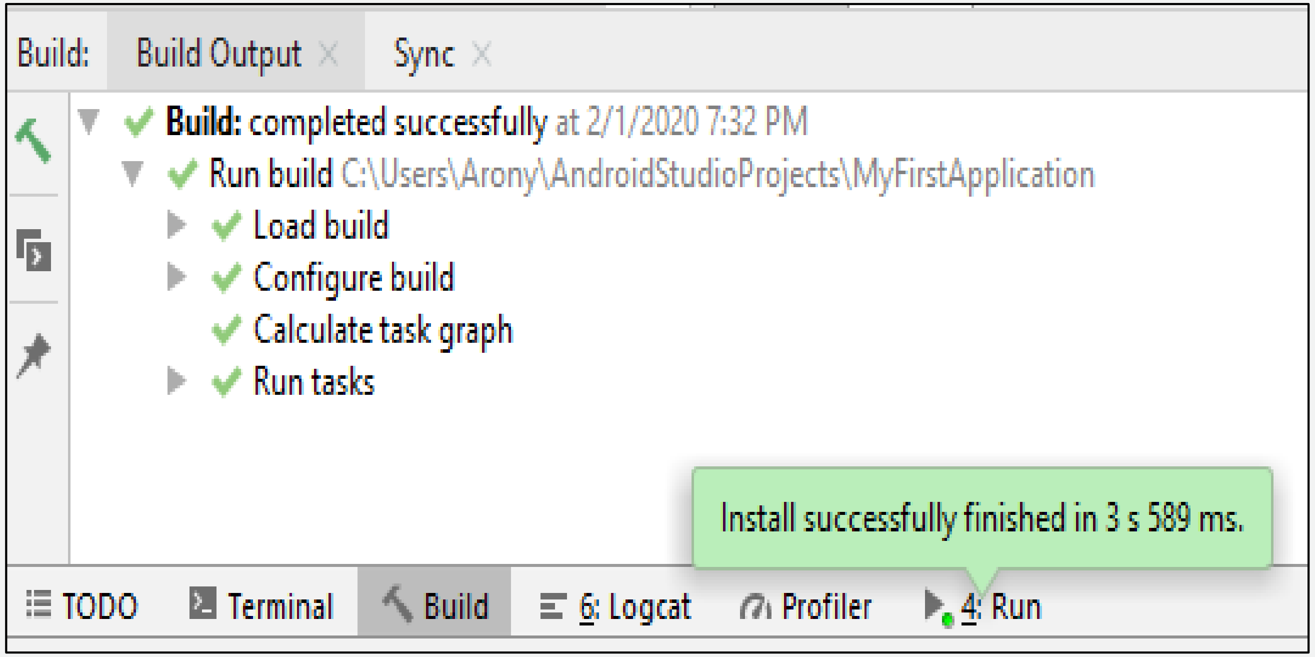
From AndroidManifest.xml MainActivity.java is called, where the onCreate() method executes.

Inside the method, setContentView runs the view activity_main and the view can be seen on the screen



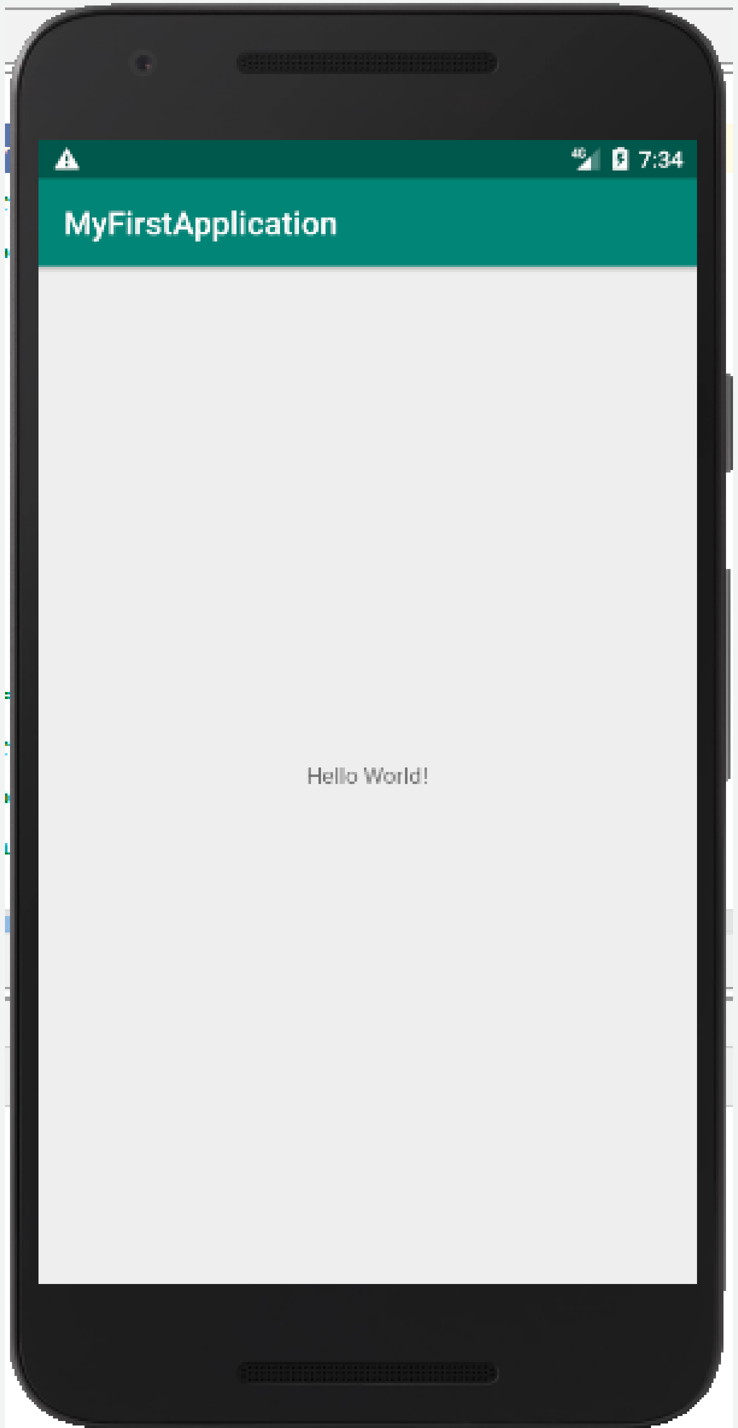


```xml
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"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

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

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



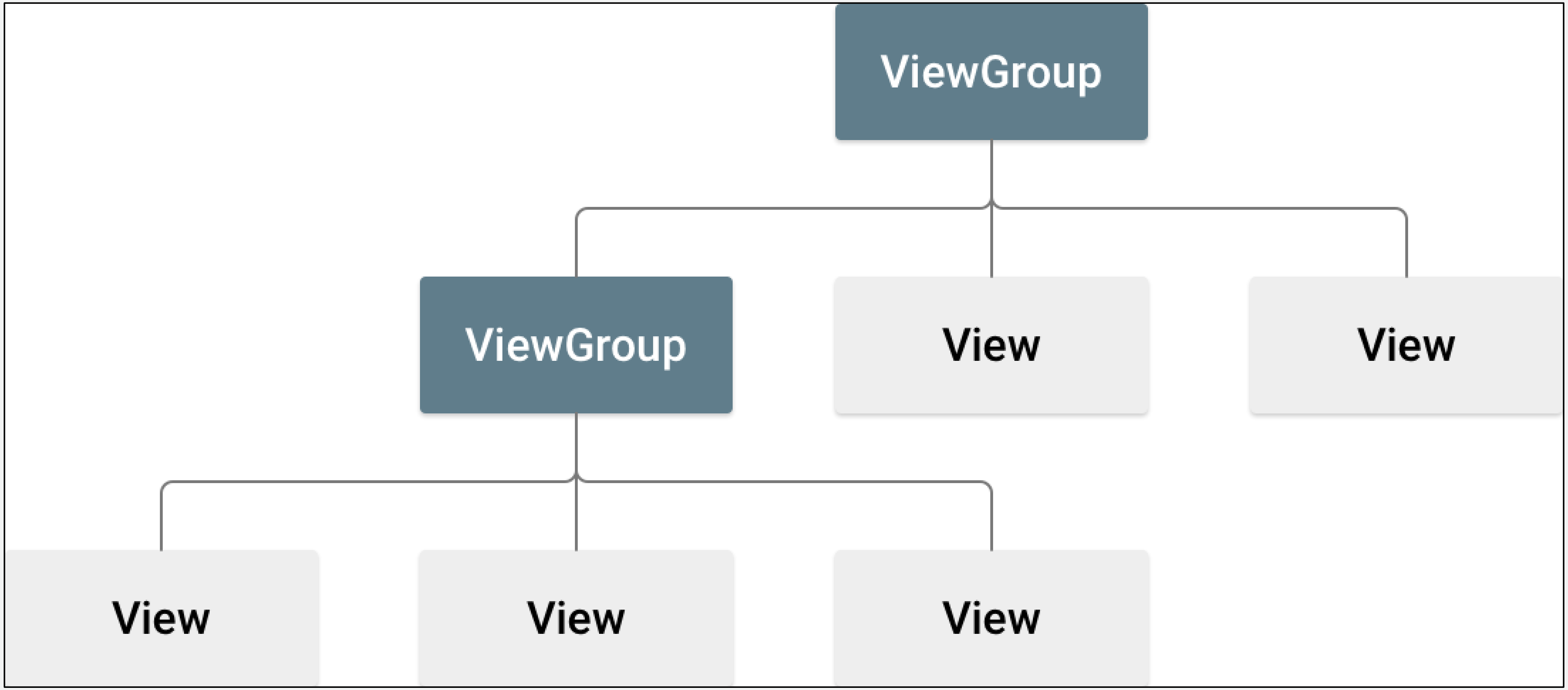
```
Build: completed successfully at 2/1/2020 7:32 PM
  Run build C:\Users\Arony\AndroidStudioProjects\MyFirstApplication
    Load build
    Configure build
    Calculate task graph
    Run tasks
```

Install successfully finished in 3 s 589 ms.



When executing completes without any error, you'll see the interface on your device

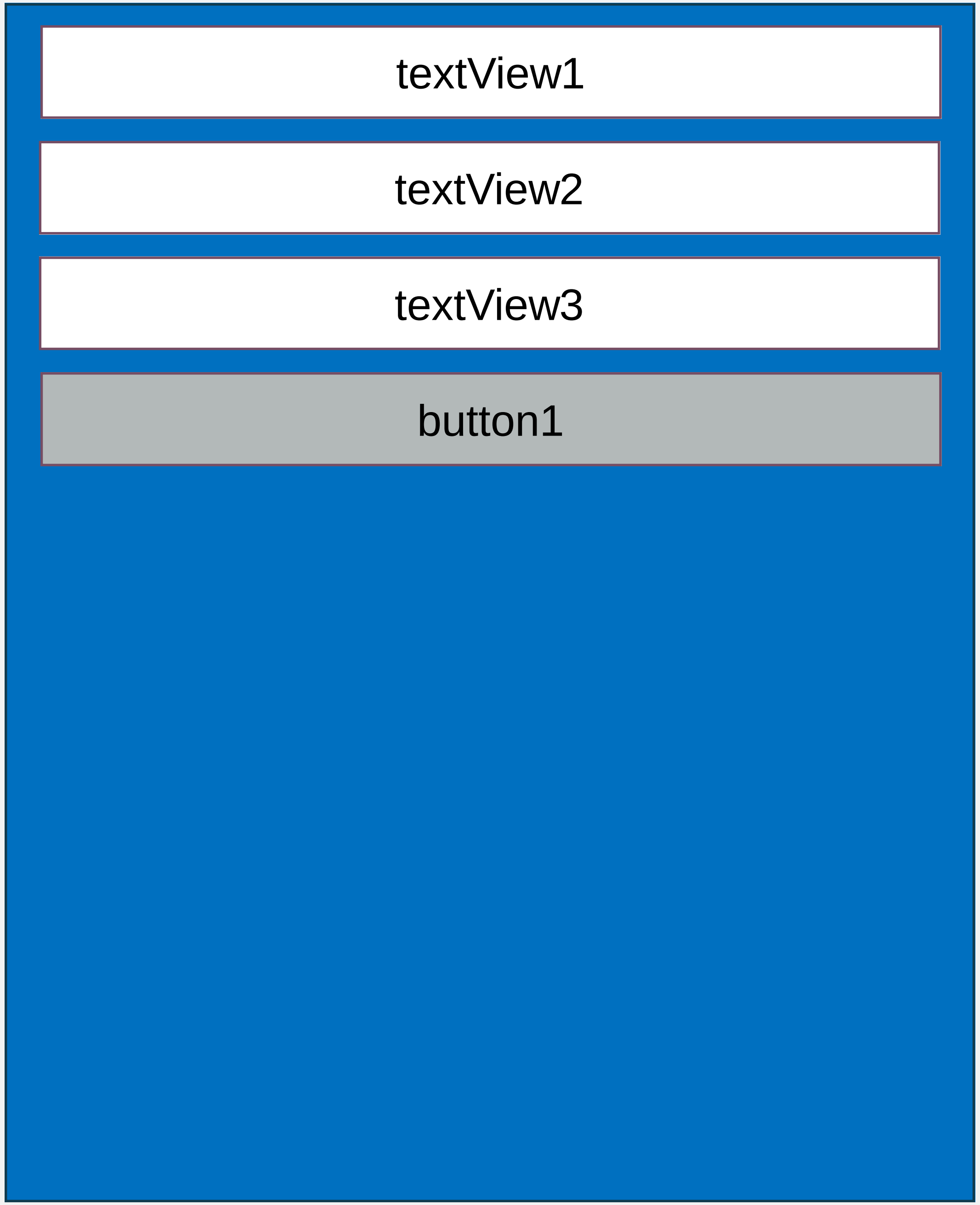
Layouts



Layouts

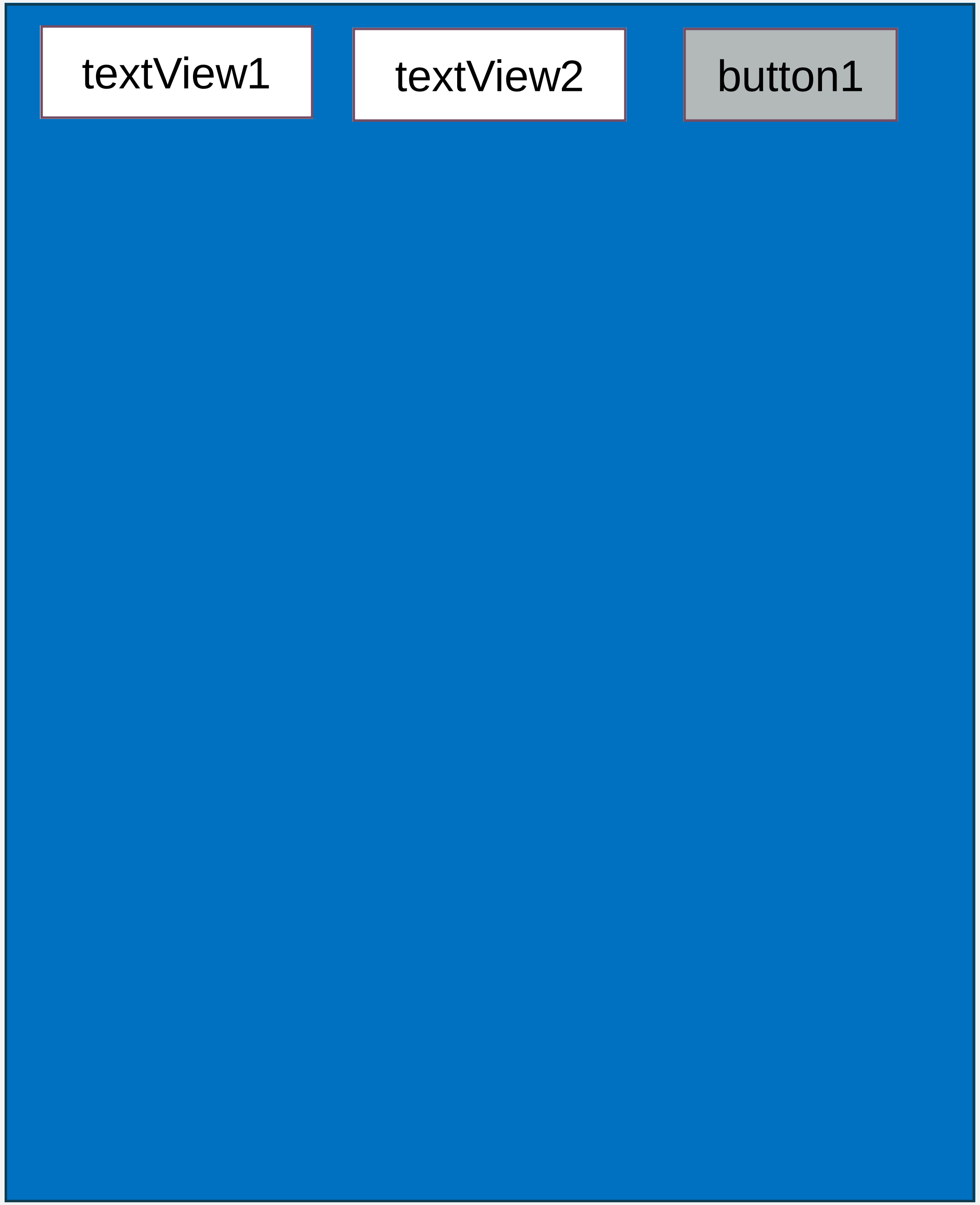
- A ViewGroup is an invisible container that defines the layout structure. That is, how the components in an activity will be placed. For example: LinearLayout, RelativeLayout etc.

- A view is something that the user can interact with. (touch/click/type). For example: button, textView etc.



- Here the **blue** background box is a viewGroup where textview and button is structured in a single column.

- And the textView and button is a view.



- Here the **blue** background box is a viewGroup where textview and button is structured in a single row.

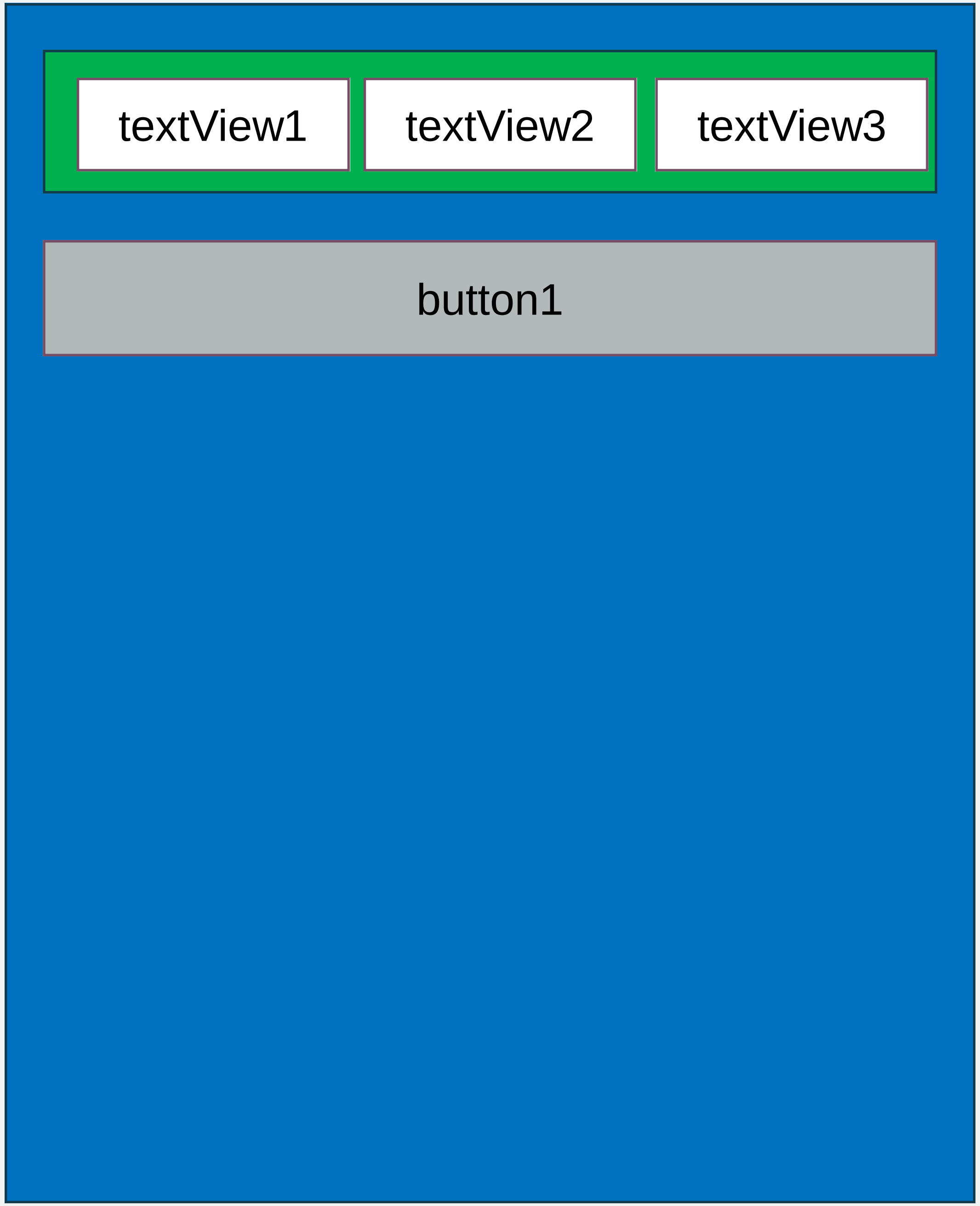
Layout Types

- LinearLayout
  - a view group that aligns all children in a single direction, vertically or horizontally.
- RelativeLayout
  - a view group that displays child views in relative positions.
- TableLayout
  - a view that groups views into rows and columns.

Layout Types

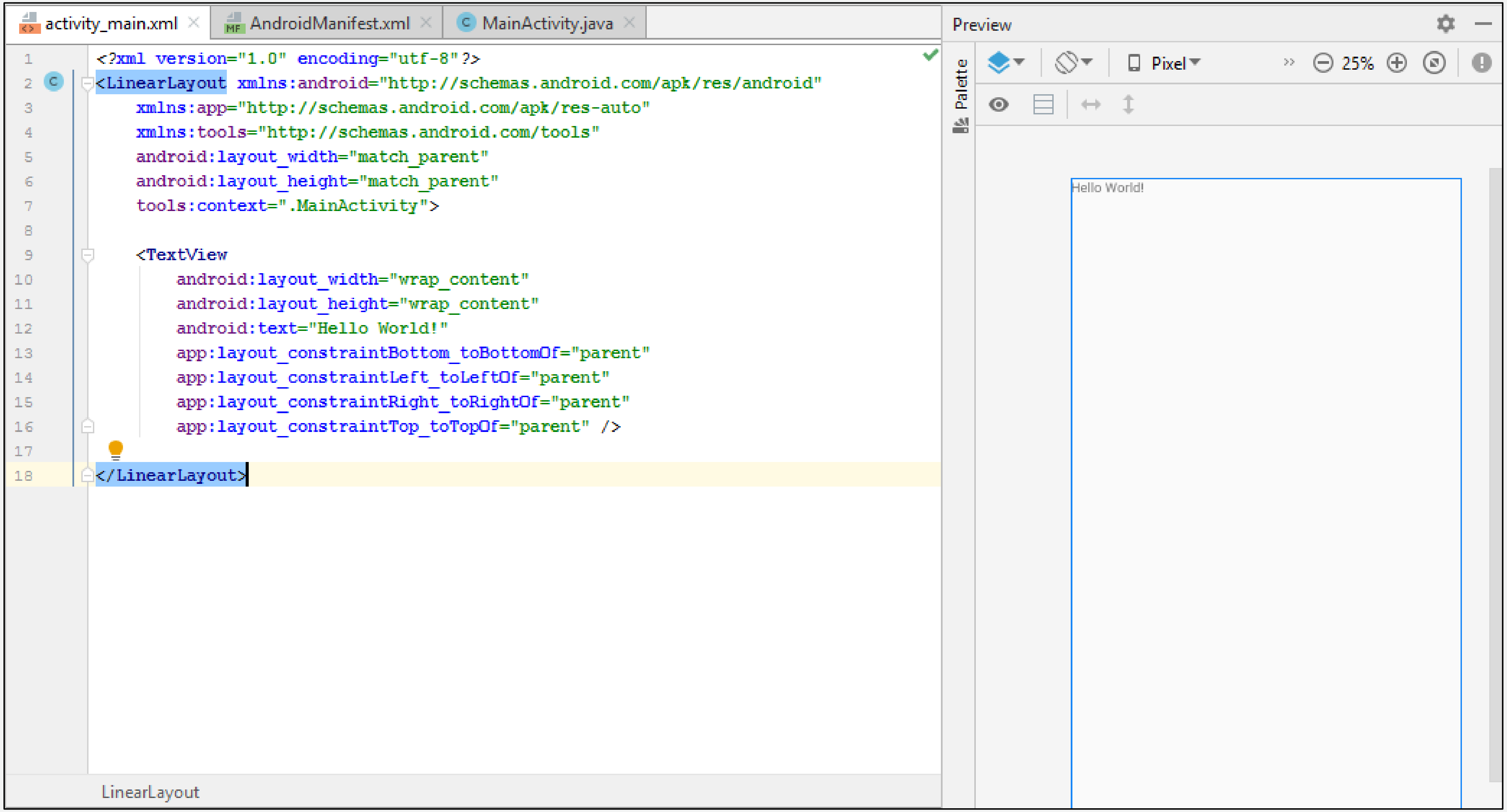
- AbsoluteLayout
  - enables you to specify the exact location of its children.
- The FrameLayout
  - a placeholder on screen that you can use to display a single view.
- ListView
  - a view group that displays a list of scrollable items.
- GridView
  - a ViewGroup that displays items in a two-dimensional, scrollable grid.

Multiple viewGroups can be placed inside a viewGroup. That is, inside a LinearLayout another LinearLayout can be placed.

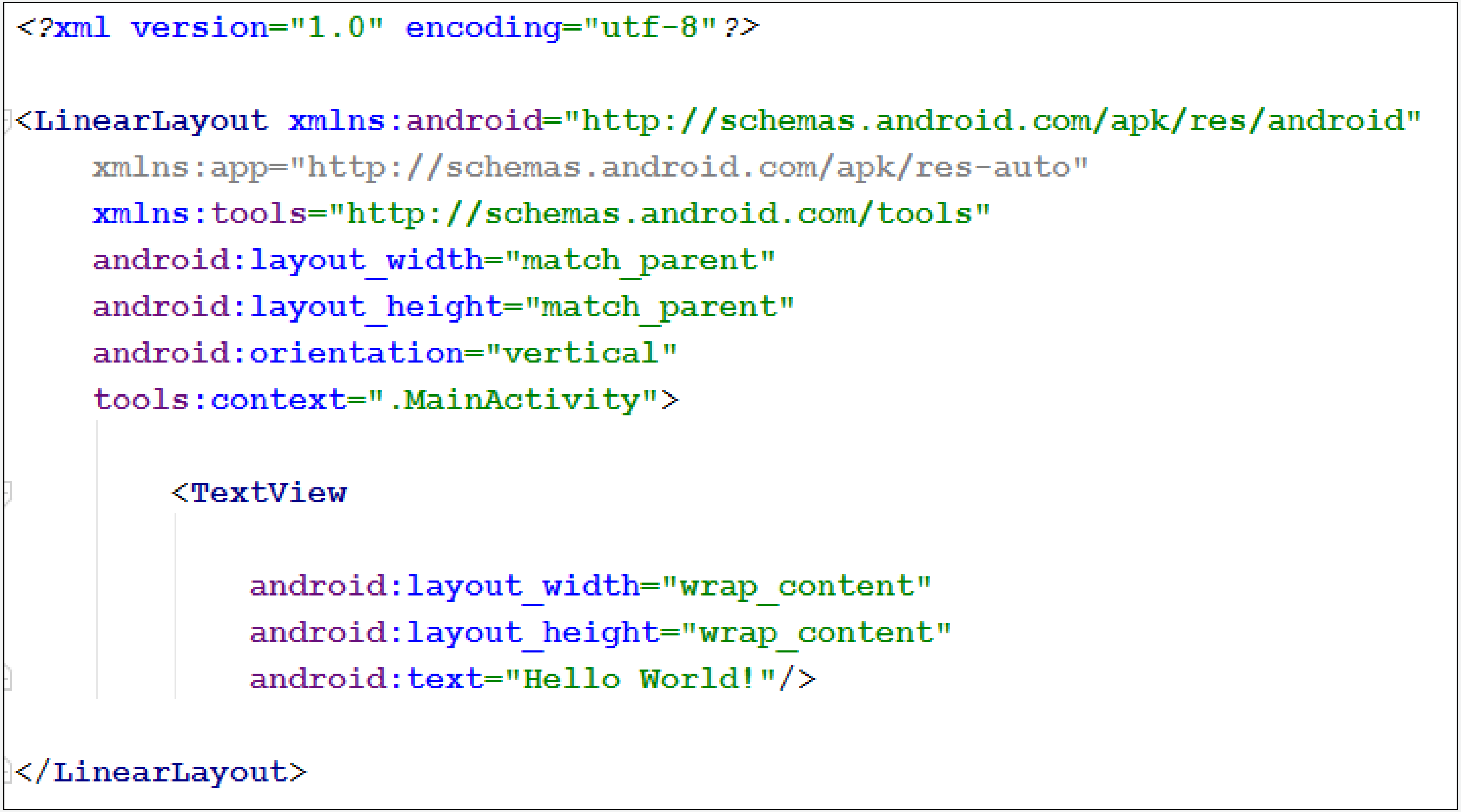


- Here the **blue** box is a LinearLayout with a vertical orientation (one column).
- Inside which the green box is another LinearLayout with a horizontal orientation (one row).

Changing the Layout to LinearLayout



```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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!"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</LinearLayout>
```



```xml
<?xml version="1.0" encoding="utf-8"?>

<LinearLayout 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"
    android:orientation="vertical"
    tools:context=".MainActivity">

        <TextView

            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Hello World!"/>

</LinearLayout>
```

Setting Height and Width

You can specify width and height with exact measurements but more often, you will use one of these constants to set the width or height −

•**android:layout_width= “wrap_content**”   (same for height)

- tells your view to size itself to the dimensions required by its content.

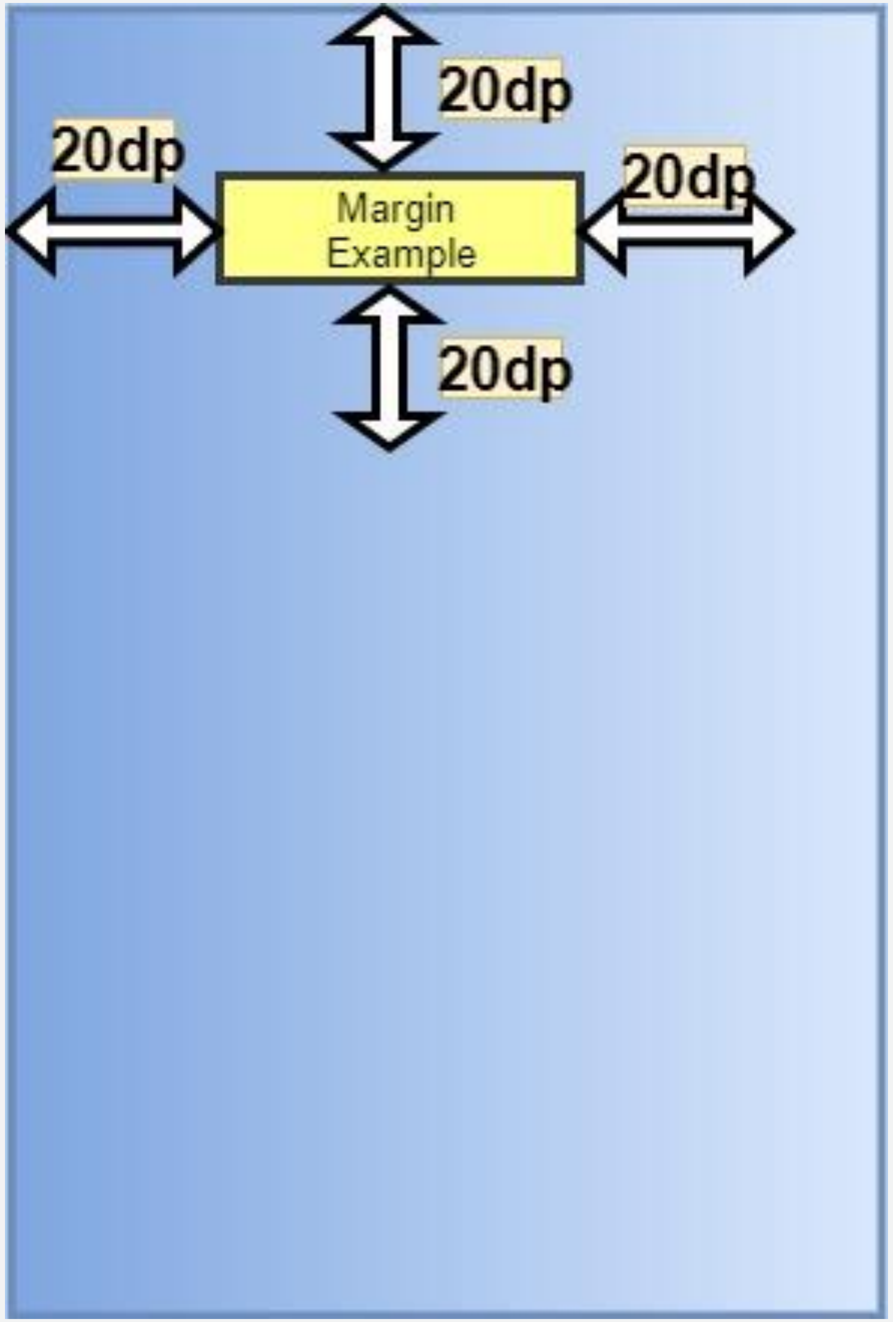
•**android:layout_width= “match_parent”** ”   (same for height)

- tells your view to become as big as its parent view

Margin

android:layout_margin="20dp”

Margin means to push outside, i.e the rectangle pushes its surrounding contents from itself by the dimension specified in the margin attribute.



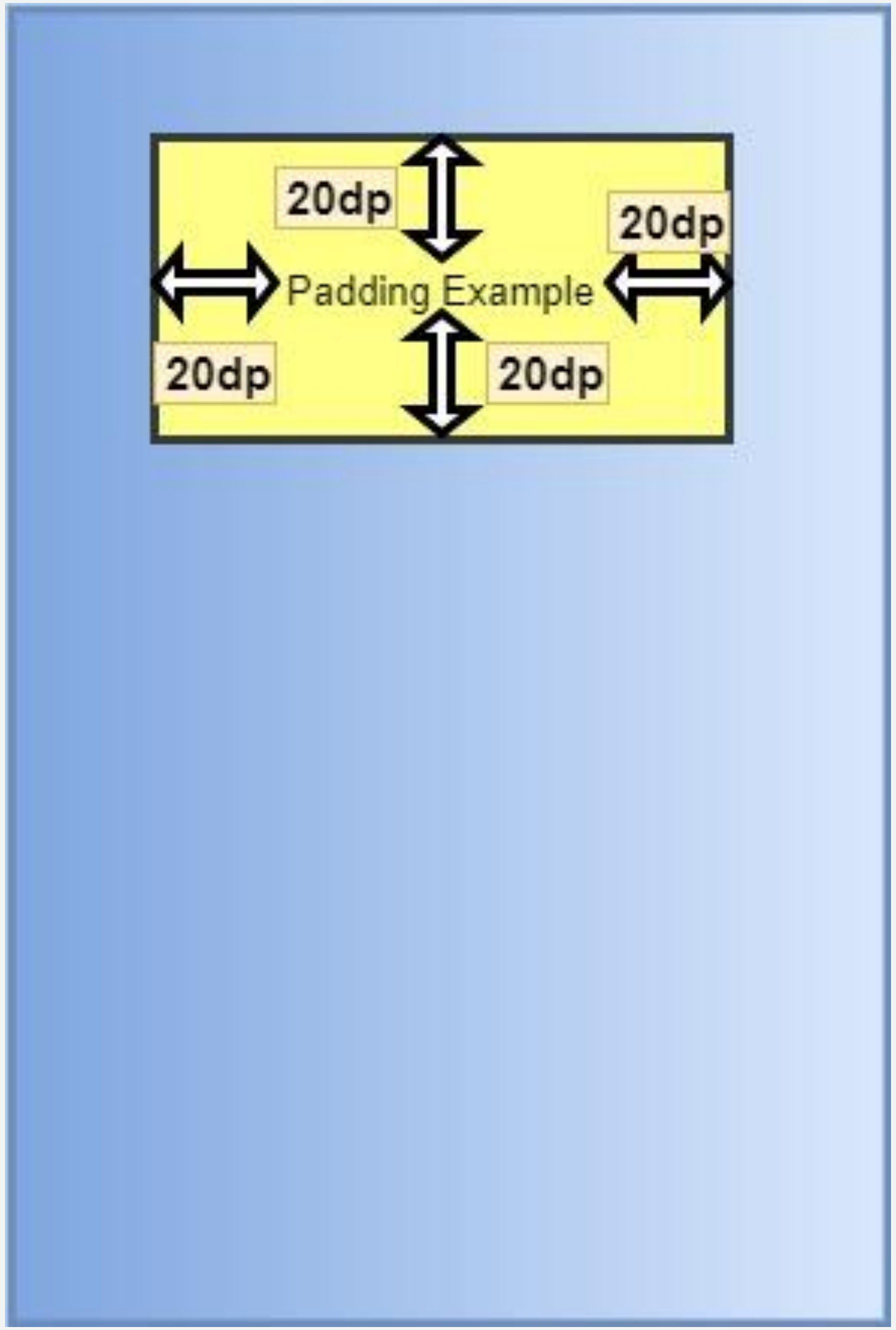
Margin can be added for all sides separately

```
android:layout_marginRight="20dp"
android:layout_marginLeft="20dp“
android:layout_marginTop="20dp“
android:layout_marginBottom="20dp"
```

Padding

android:padding="20dp"

Padding means to push inside, i.e the rectangle pushes its contents from itself by the dimension specified in the padding attribute towards its center. Padding can be considered as margin but inside the View.

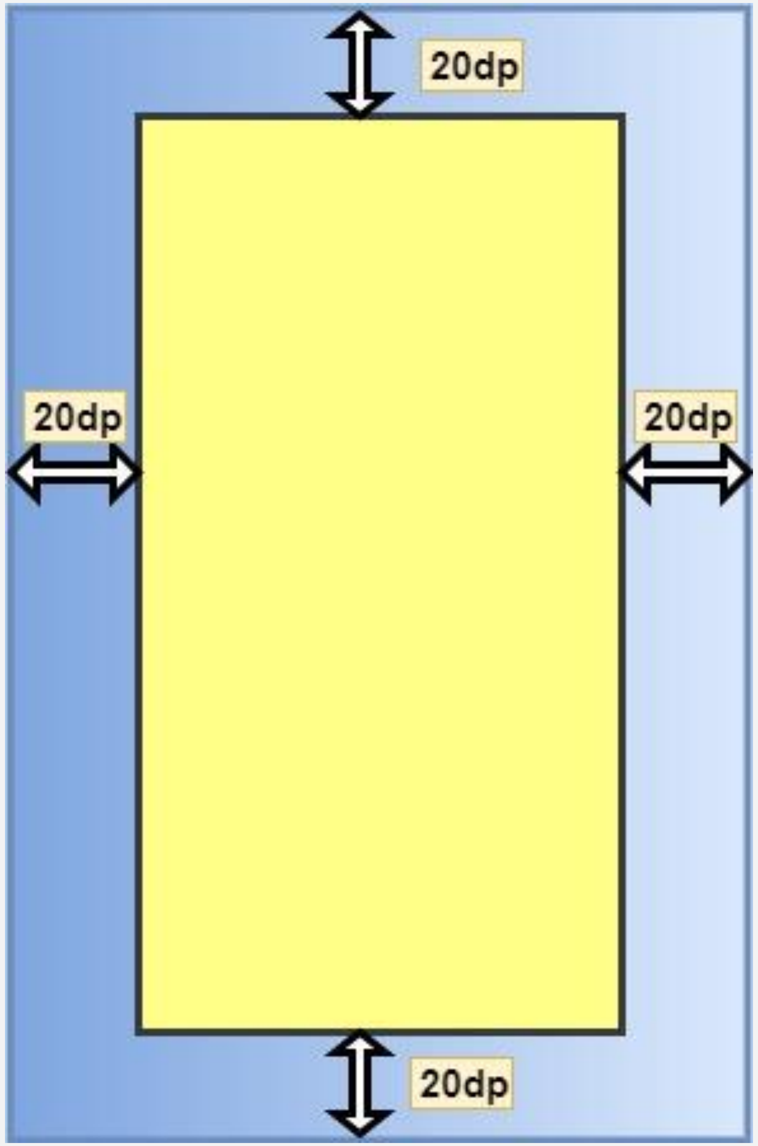


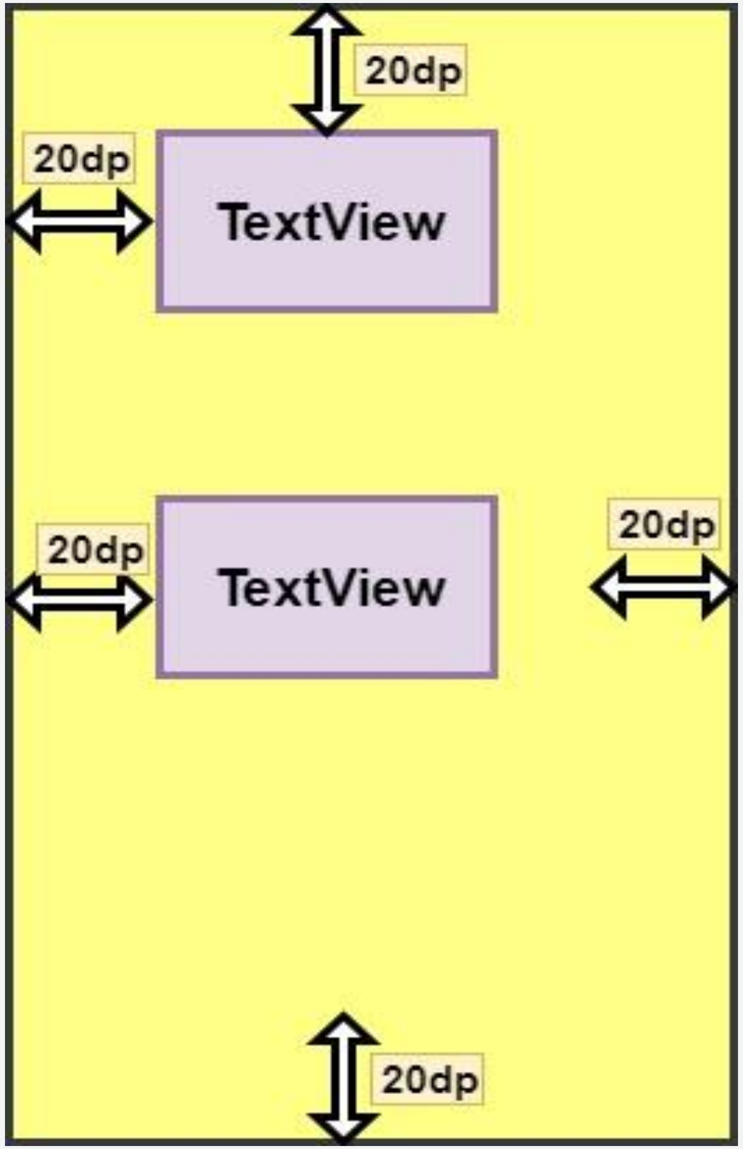
Margin can be added for all sides separately

```
android:paddingRight ="20dp"
android:paddingLeft ="20dp"
android:paddingTop ="20dp"
android:paddingBottom ="20dp"
```

Margin and Padding with Layout

Margin





Padding