

Lab01

Android Development Environment

EE5415

Mobile Apps Design and Development

Lab01 Objectives

- To provide Android Studio setup information
- To have experience on using Android Studio to build a simple app: “My Calendar”
 - Understand the various parts of an Android Project in Android Studio
 - Create your Android Virtual Device (AVD)
 - Run your developed My Calendar app on your created AVD

Computer Requirements

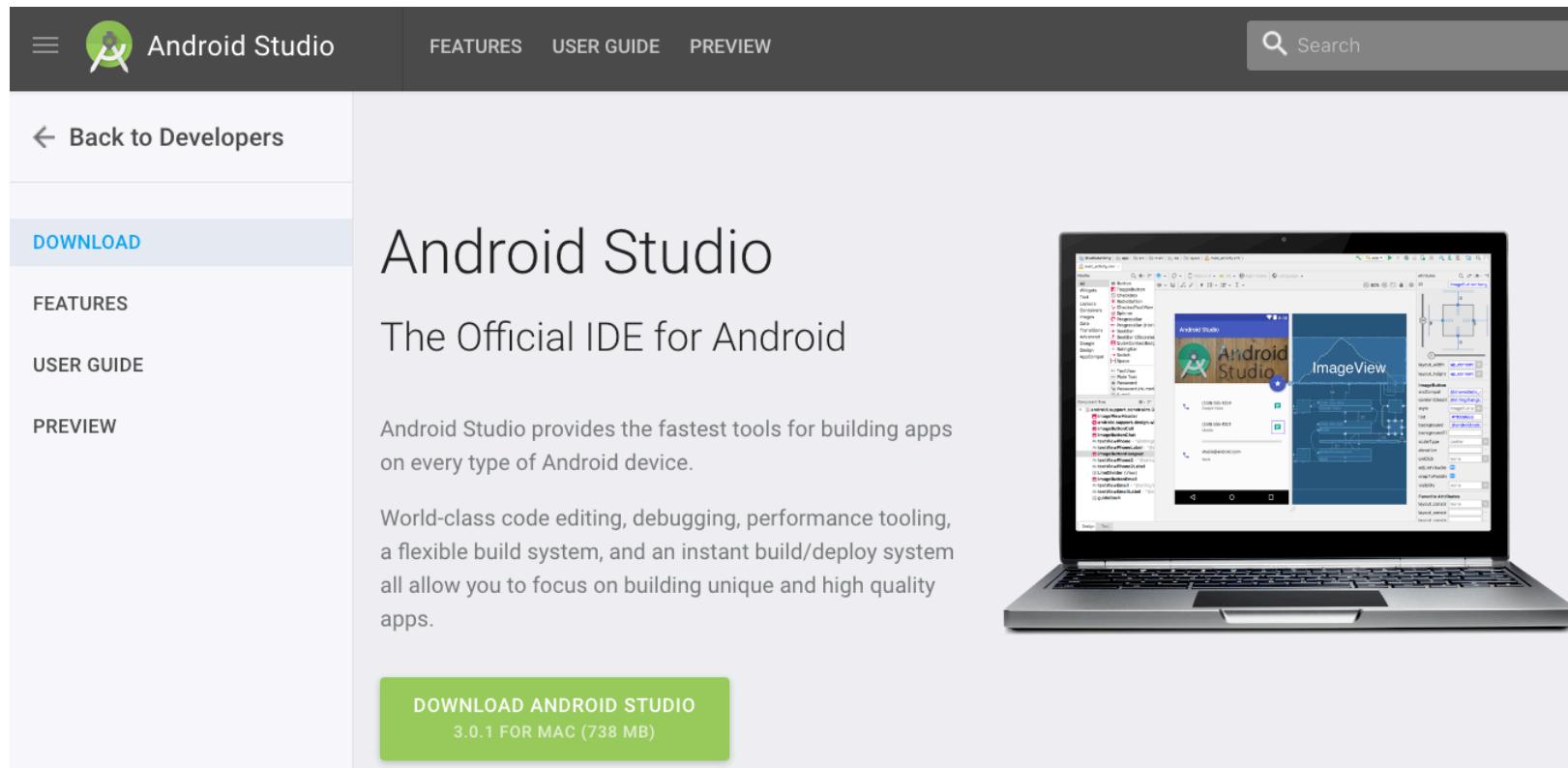
- Microsoft Windows 10/8/7/vista/2003
- Mac OS X 10.8.5 or above
- Linux (Tested on Ubuntu 14.04)
- *It is recommended to use Macbook Pro as it can be used for both Android and iOS application developments*

Software Requirements

- Setting up of Android Development Environment
 - Install Android Studio (version 3.5.3)
 - Install latest SDK components for Android 10 (Q)
 - API 29
 - Create Android Virtual Device (AVD)

Android Studio Installation

- Download and install Android Studio
 - <http://developer.android.com/sdk>
 - YouTube Tutorial: <https://www.youtube.com/watch?v=RVP5kJiwfDo>



What is Android Studio?

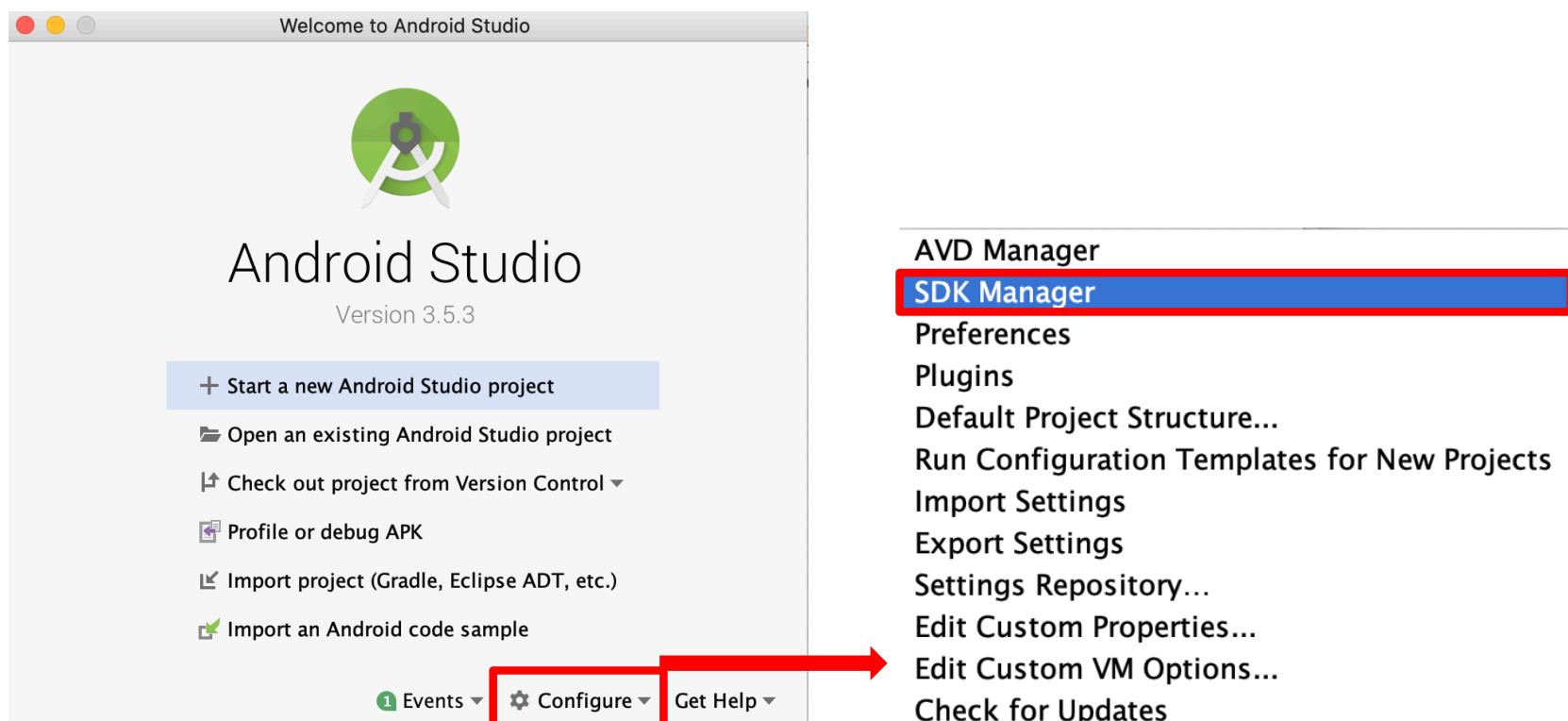
- Android Studio is the official IDE for Android application development starting from Dec. 2014
- Android Studio is based IntelliJ IDEA
 - IntelliJ IDEA is a leading Java development environment
 - Published by JetBrains
 - Community Edition (Free) for basic Java development tools
 - Ultimate Edition (Paid)
 - Android Studio is based on IntelliJ IDEA Community Edition
 - Android Studio is free

Android Studio Setup

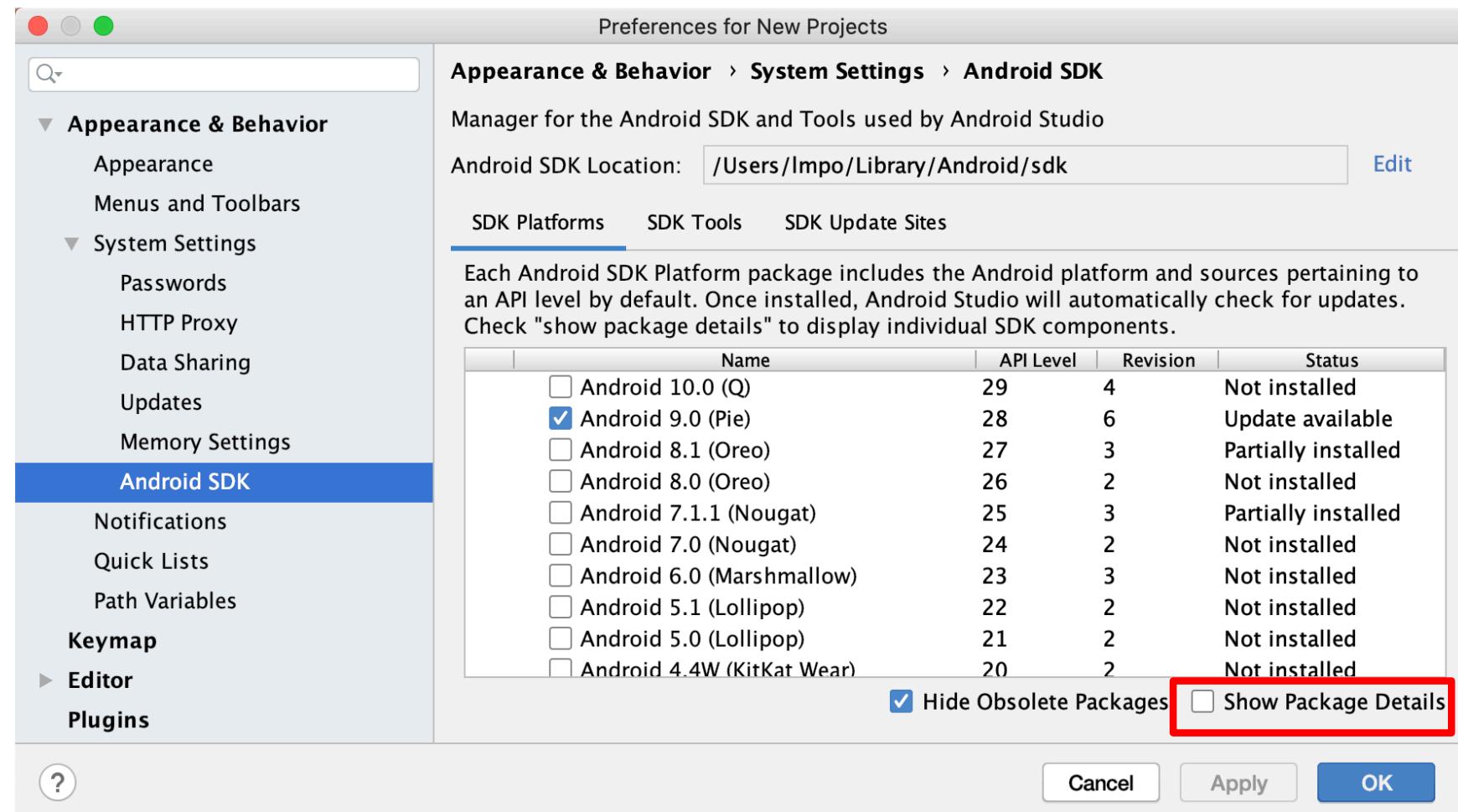
- Android Studio also come with Android Studio IDE, Android SDK tools, Android Platform and emulator system images.
- Before installing Android Studio, your system needs to have Java Development Kit (JDK) from Oracle.
 - <http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>
 - Java SE Development Kit 8 is recommended.
 - Check version by command: `java -version`

Adding SDK Packages

- By default, the Android SDK does not include everything you need to start your app development.
- Before starting your project development, there are a few packages you should add to your Android SDK.



Get the Latest SDK Platforms



Essential SDK Tools

Preferences for New Projects

Appearance & Behavior > System Settings > Android SDK

Manager for the Android SDK and Tools used by Android Studio

Android SDK Location: /Users/lmpo/Library/Android/sdk [Edit](#)

[Reset](#)

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 Le...	Revisi...	Status
▼	Android 10.0 (Q)			
↓	✓ Android SDK Platform 29	29	4	Not installed
↓	✓ Sources for Android 29	29	1	Not installed
	☐ Intel x86 Atom System Image	29	7	Not installed
	☐ Intel x86 Atom_64 System Image	29	7	Not installed
	☐ Google APIs Intel x86 Atom System Image	29	9	Not installed
	☐ Google APIs Intel x86 Atom_64 System Image	29	9	Not installed
↓	✓ Google Play Intel x86 Atom System Image	29	8	Not installed
	☐ Google Play Intel x86 Atom_64 System Image	29	8	Not installed
▼	Android O Preview			

Hide Obsolete Packages Show Package Details

?

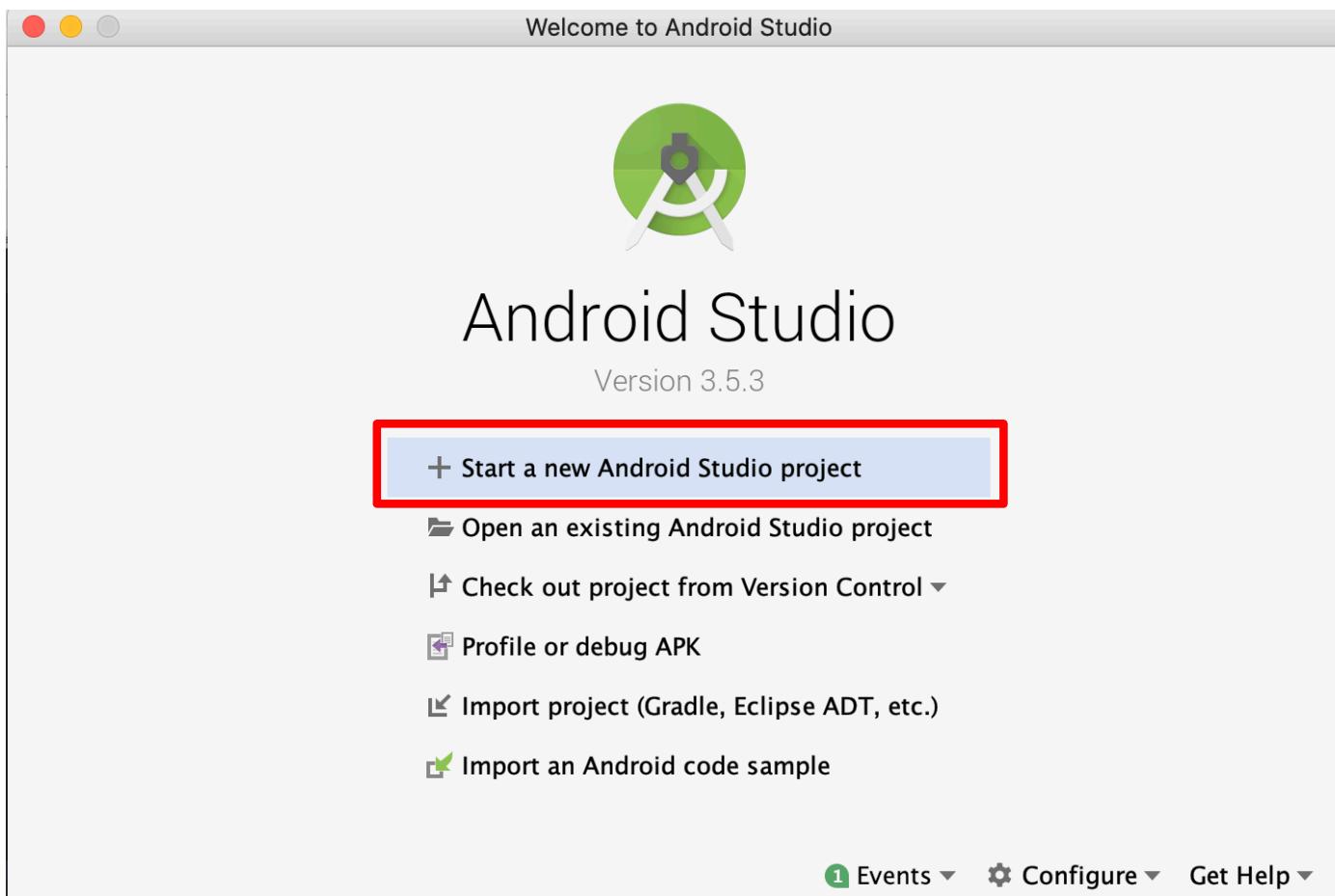
Cancel Apply OK

Suggested SDK Configurations

- **SDK Platform**
 - Open the Android 10.0 folder (the latest version) and select:
 - Android SDK Platform 29
 - Samples for SDK 29
 - Google APIs Intel x86 Atom_64 System Image
 - Google APIs Intel x86 Atom System Image
- **SDK Tools**
 - Android SDK Build Tools
 - Android SDK Platform Tools
 - Android SDK Tools
 - **Google USB Driver (PC only)**
 - Intel x86 Emulator Accelerator (HAXM installer)
 - Android Support Repository
 - Android Support Library

Create Your First Project in AS

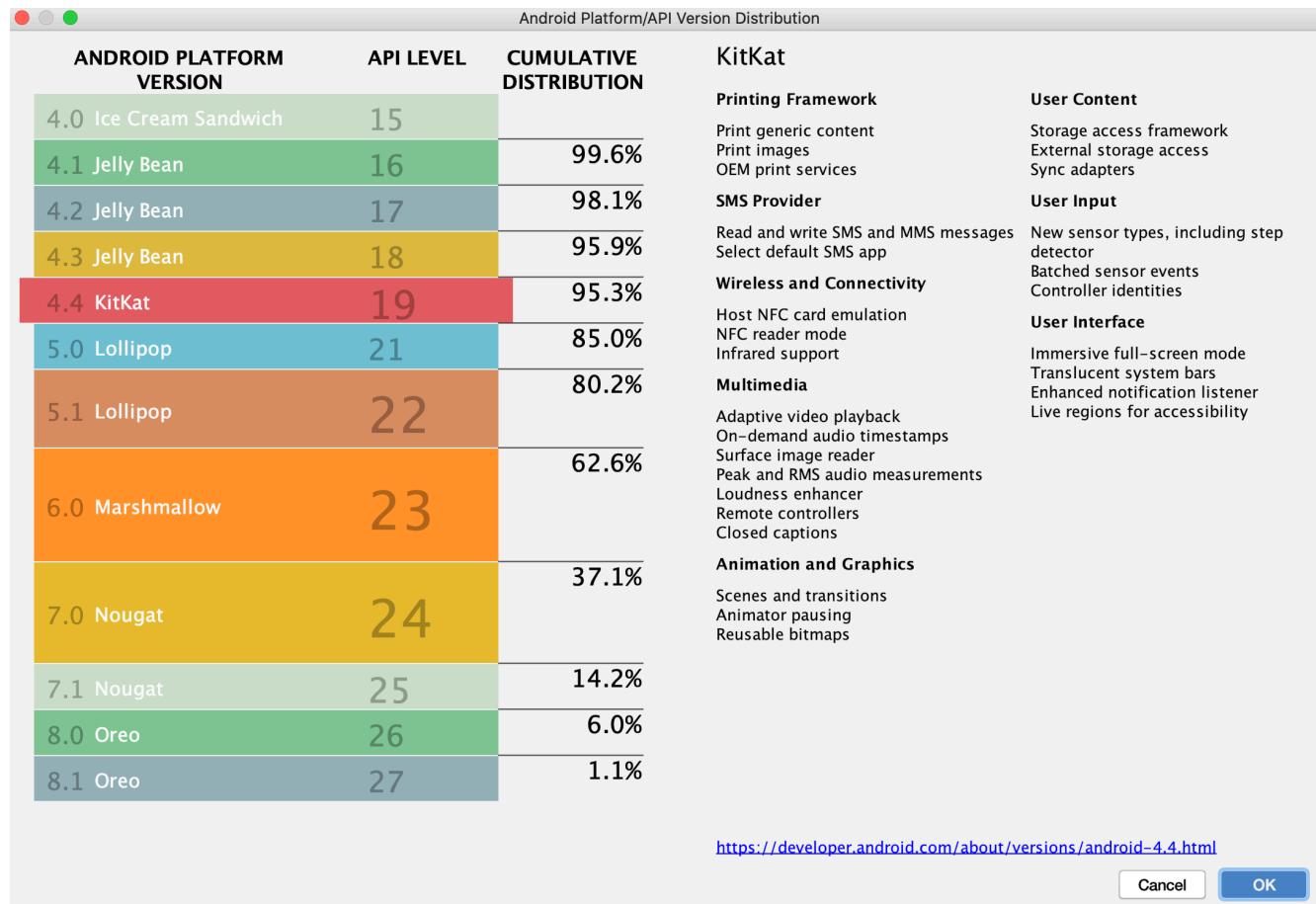
- In the AS welcome menu, click the “Start a new Android Studio project”



Create “My Calendar” Project

- Choose your project : “**Empty Activity**”
- Application name: **My Calendar**
- Package name: com.example.mycalendar
- Save location: /username/AndroidStudioProjects/MyCalendar
- Language: **Java** or **Kotlin**
- Minimum API level: API 19: Android 4.4 (KitKat)
- This configure will automatedly create an Empty Activity
 - Activity Name: **MainActivity**
 - Layout Name: **activity_main**

Platform Versions Distribution



- With minimum SDK set to API 19: Android 4.4 (KitKat), the developed app can **support about 95.3% of existing Android devices**.
- This is the reason why it is recommended to set minimum SDK to API 19 as the developed apps can support most of the Android devices and relatively new Android features.

<http://developer.android.com/about/dashboards/index.html>

Configure your project

Create New Project

Configure your project

Name
My Calendar

Package name
com.example.mycalendar

Save location
/Users/Impo/AndroidStudioProjects/MyCalendar

Language
Java or Kotlin

Empty Activity
Creates a new empty activity

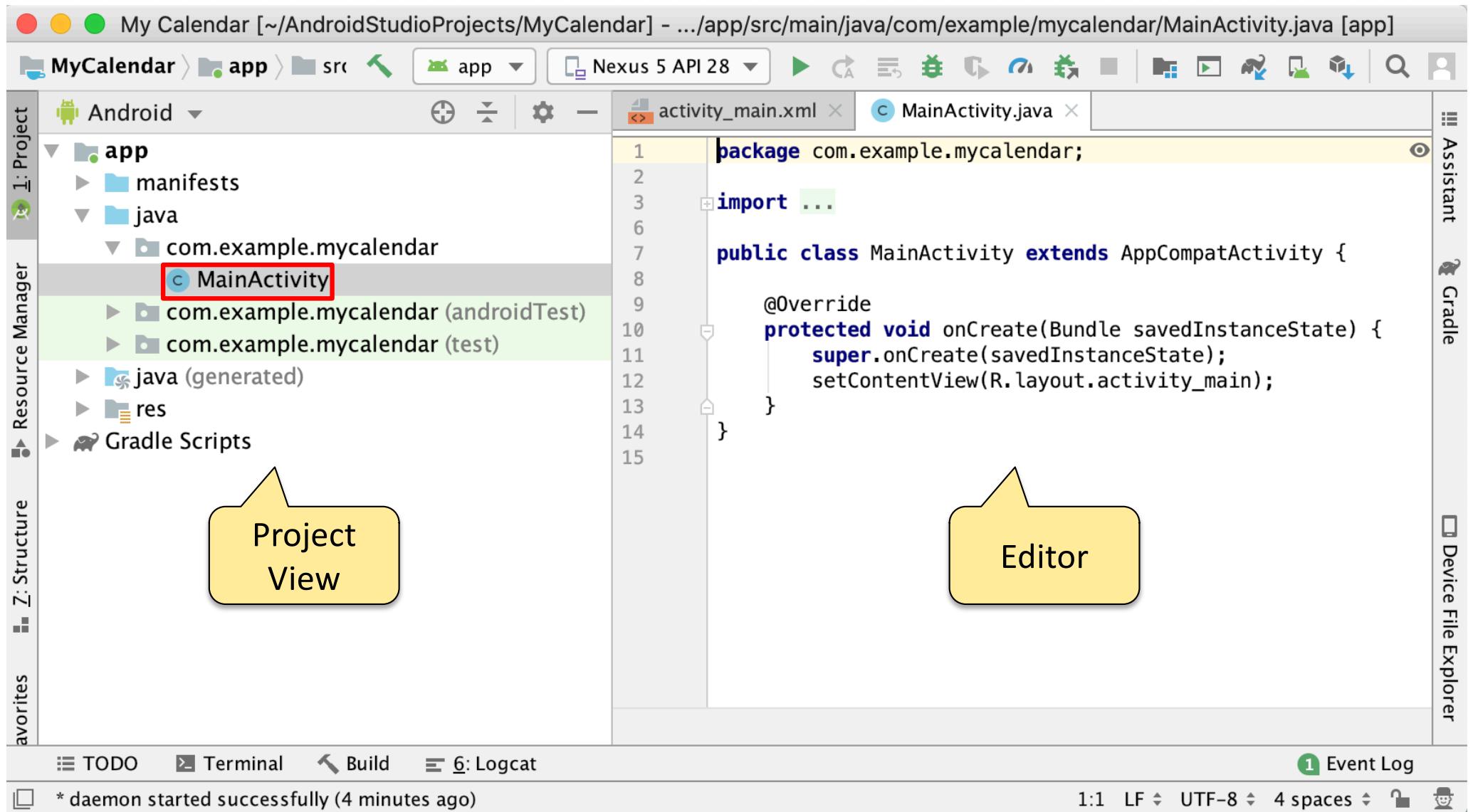
Minimum API level
API 19: Android 4.4 (KitKat)

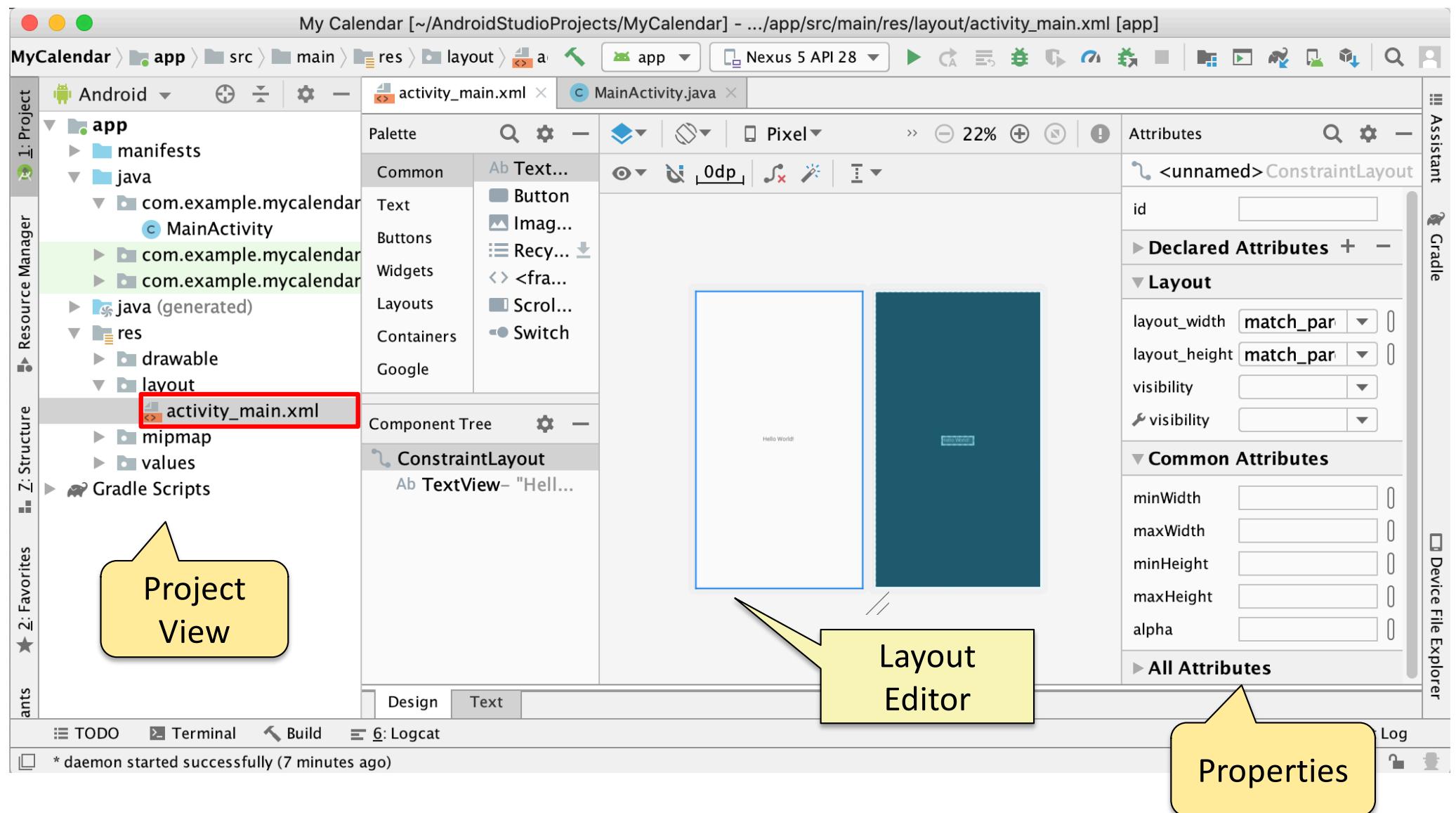
i Your app will run on approximately 95.3% of devices.
[Help me choose](#)

This project will support instant apps

Use androidx.* artifacts

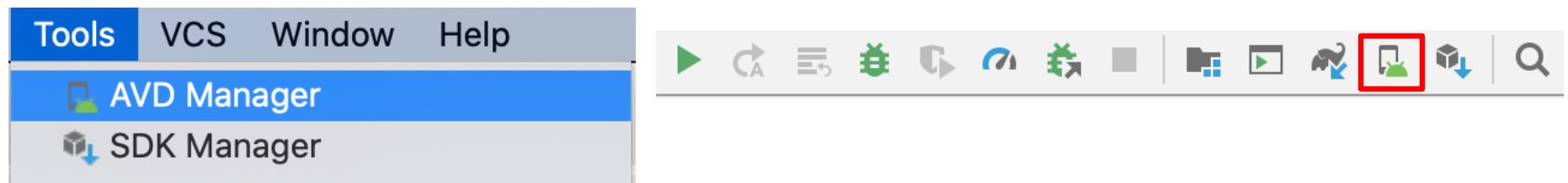
Cancel Previous Next **Finish**



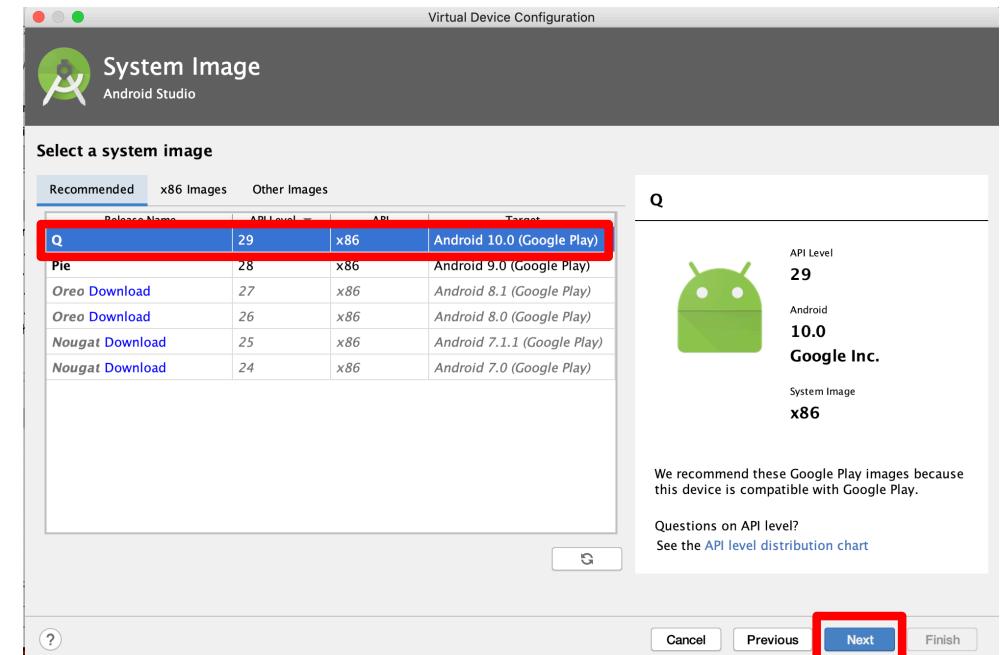
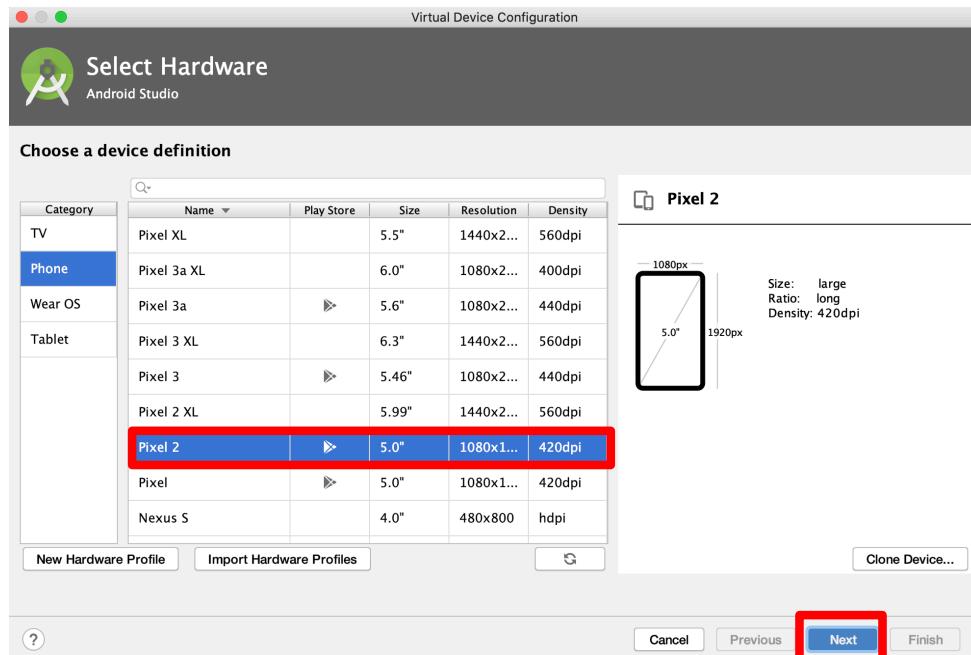
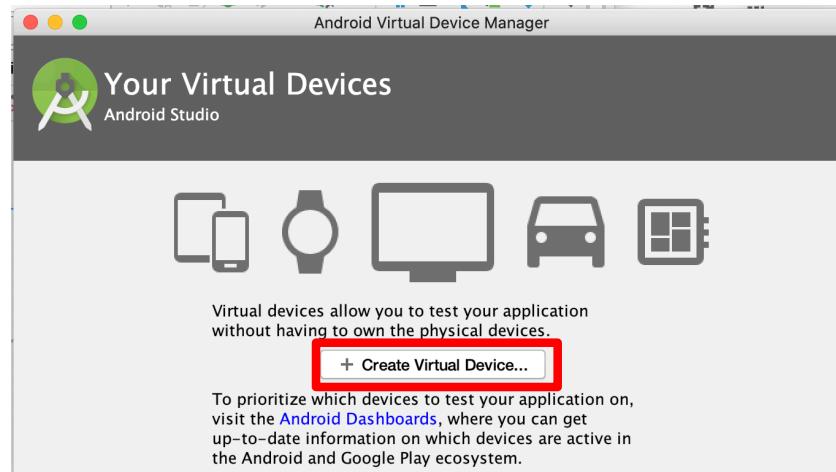


Android Virtual Device Manager

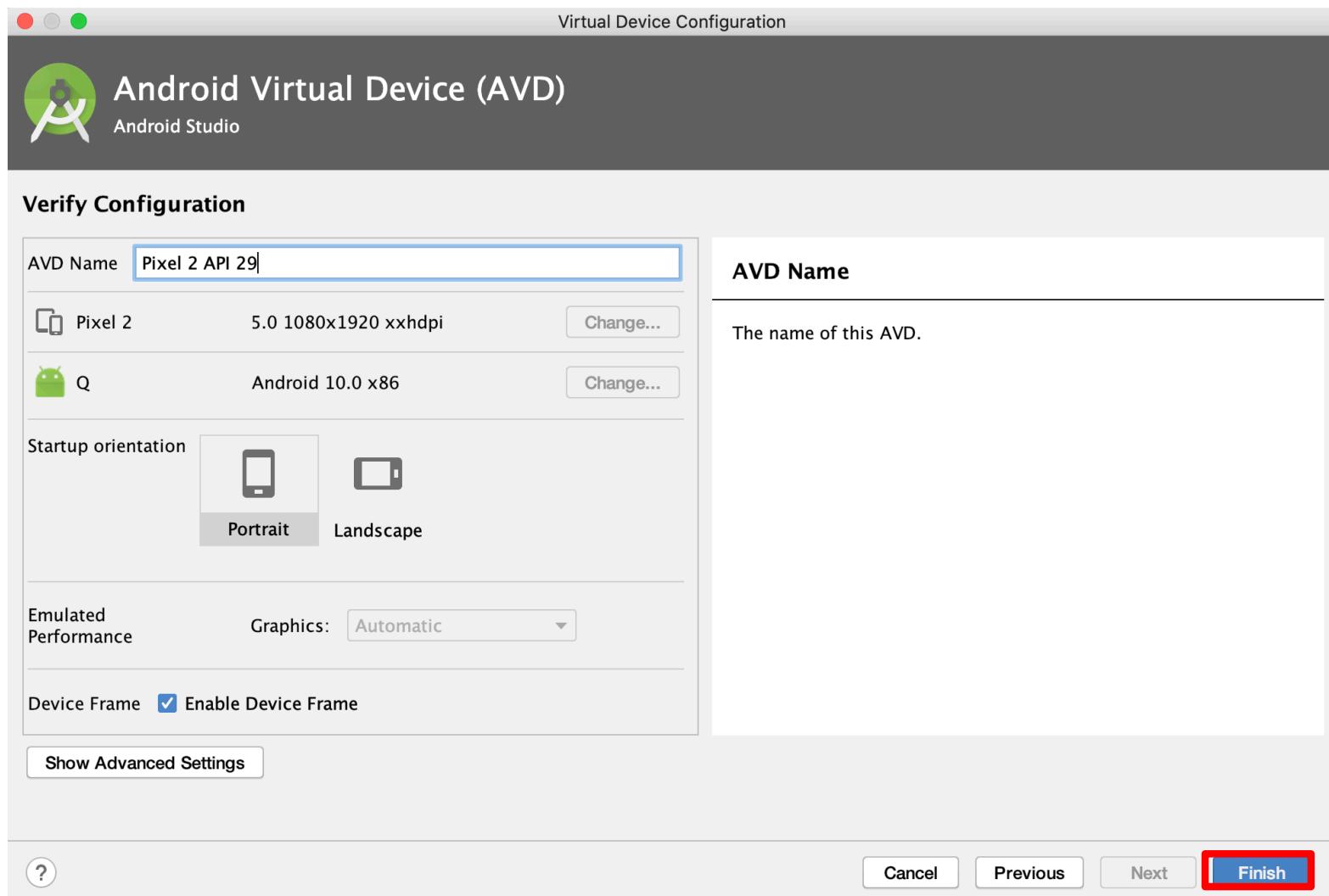
- Android Studio provides virtual devices for all shapes and sizes
 - Android Studio comes pre-configured with an optimized emulator image.
 - The updated and streamlined Virtual Device Manager provides pre-defined device profiles for common Android devices.
- You can launch the AVD Manager in Android Studio by selecting **Tools > AVD Manager**, or click the AVD Manager icon in the toolbar.



Create Your Android Virtual Device (AVD)

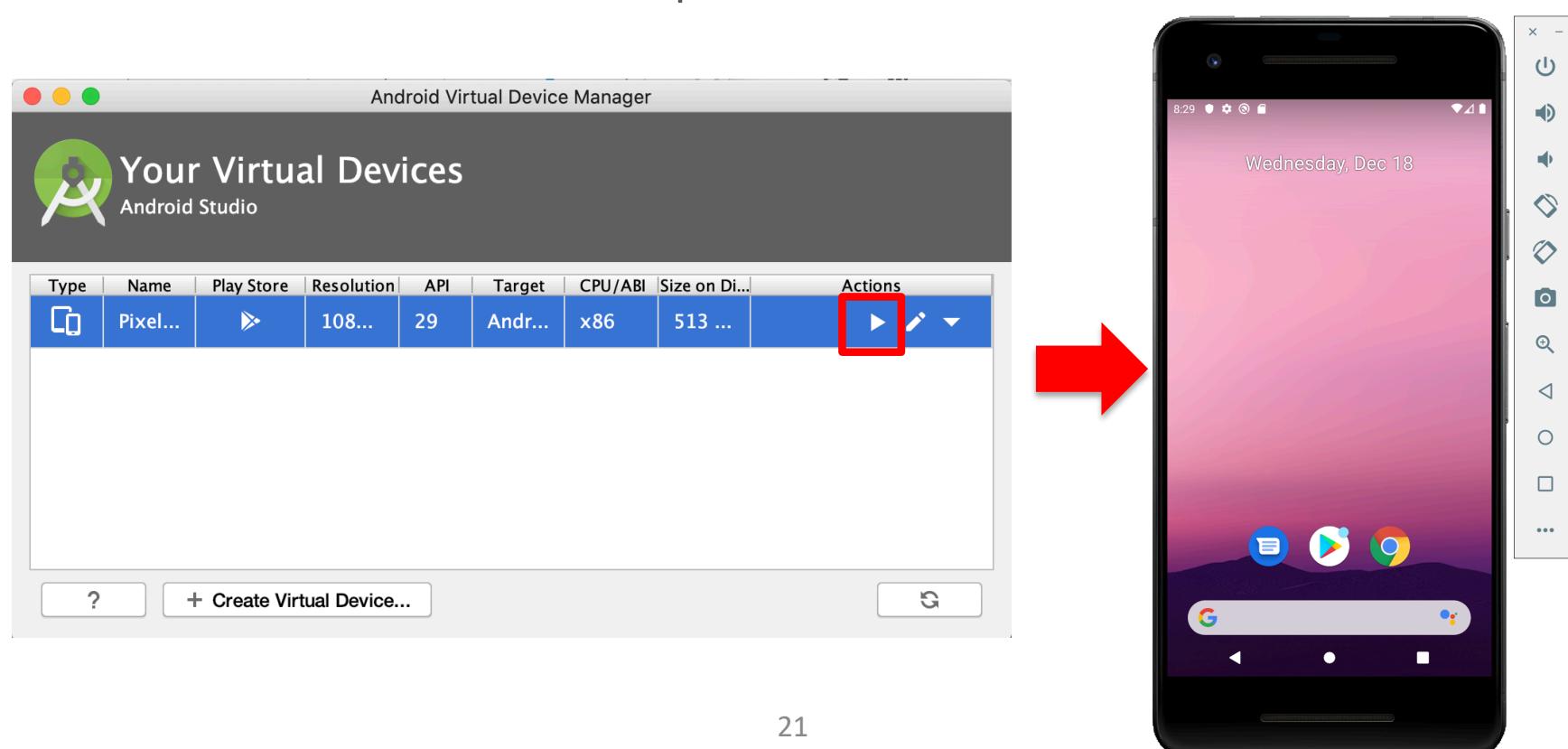


AVD Advanced Settings



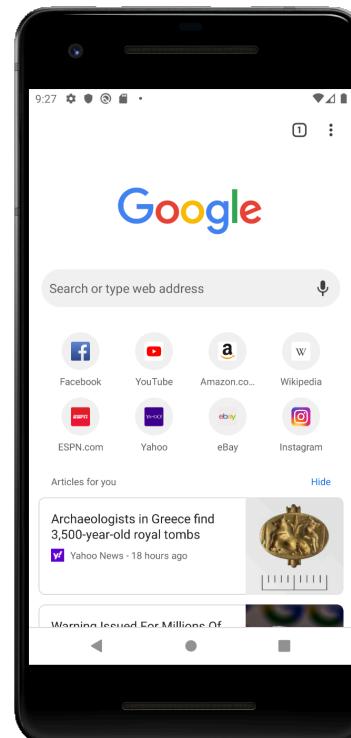
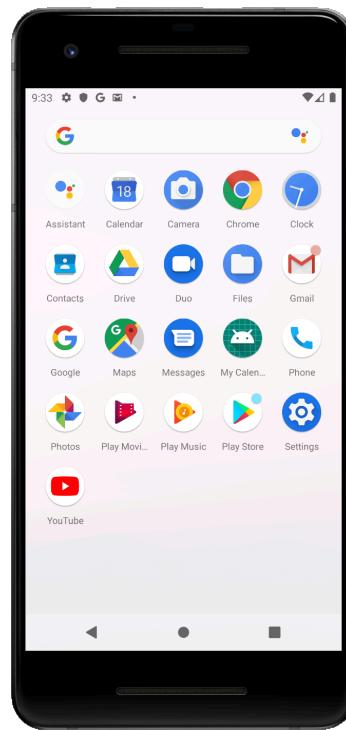
Efficient Virtual Device Configurations

- To have fast virtual device for your app development:
 - To select **System images** for Intel Architecture:
 - Intel Atomx86
 - Intel Atomx86_64
 - To enable **Use Host GPU** option for Emulated Performance

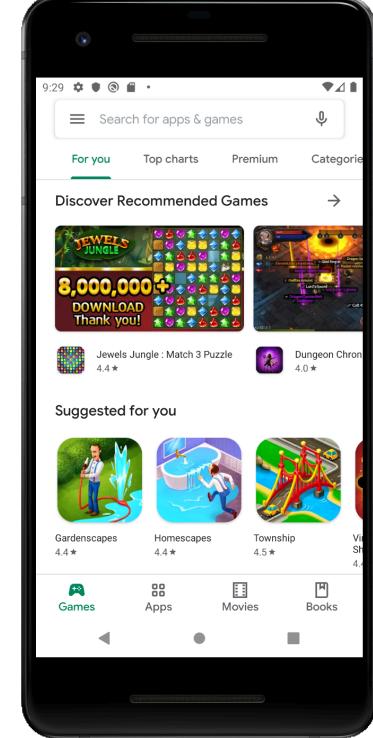


Launch the AVD

- Select Virtual Device, click “Start...” button, then [Launch Options]windows will appear
- Click [Launch] button to start the AVD



Chrome

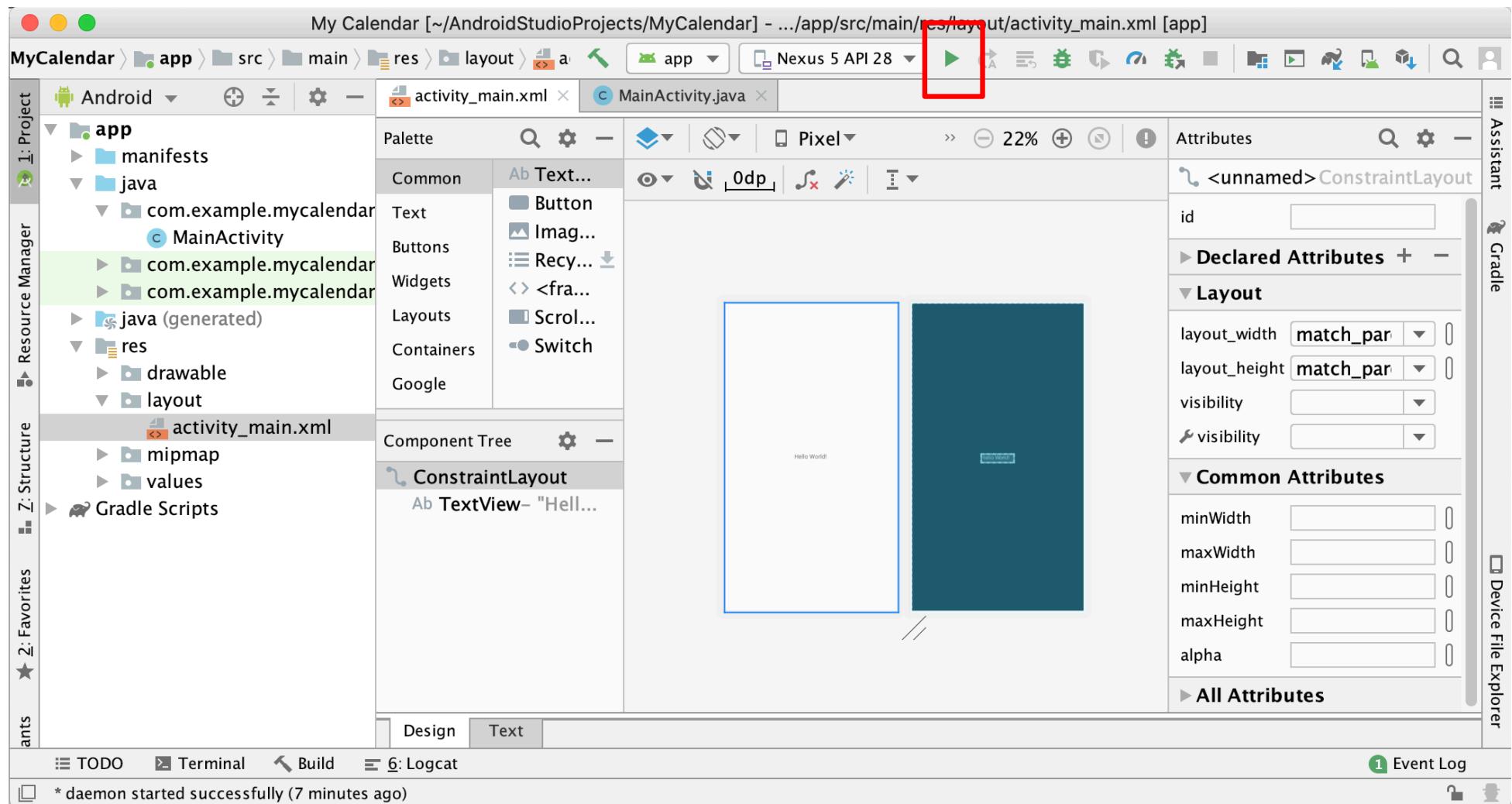


Play Store

Pros and Cons of using AVD

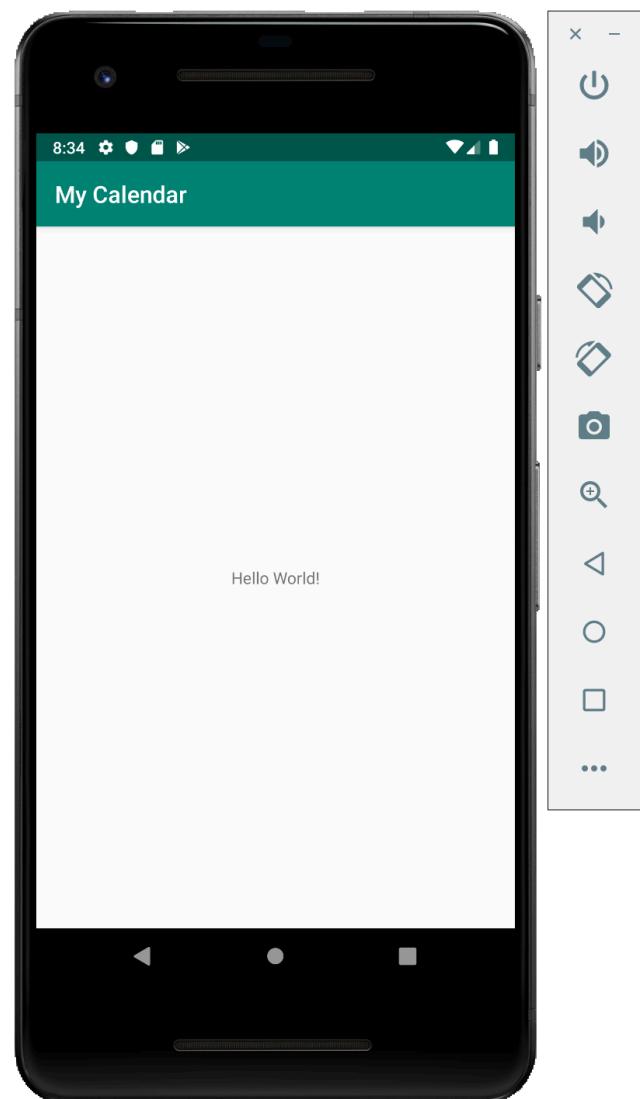
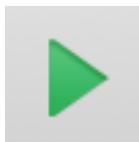
- **Pros**
 - Doesn't require an actual phone
 - Hardware is reconfigurable
- **Cons**
 - Can be very slow
 - Some features unavailable
 - E.g. No support for Bluetooth or USB connections
 - Performance / user experience can be misleading

Run the “My Calendar” App

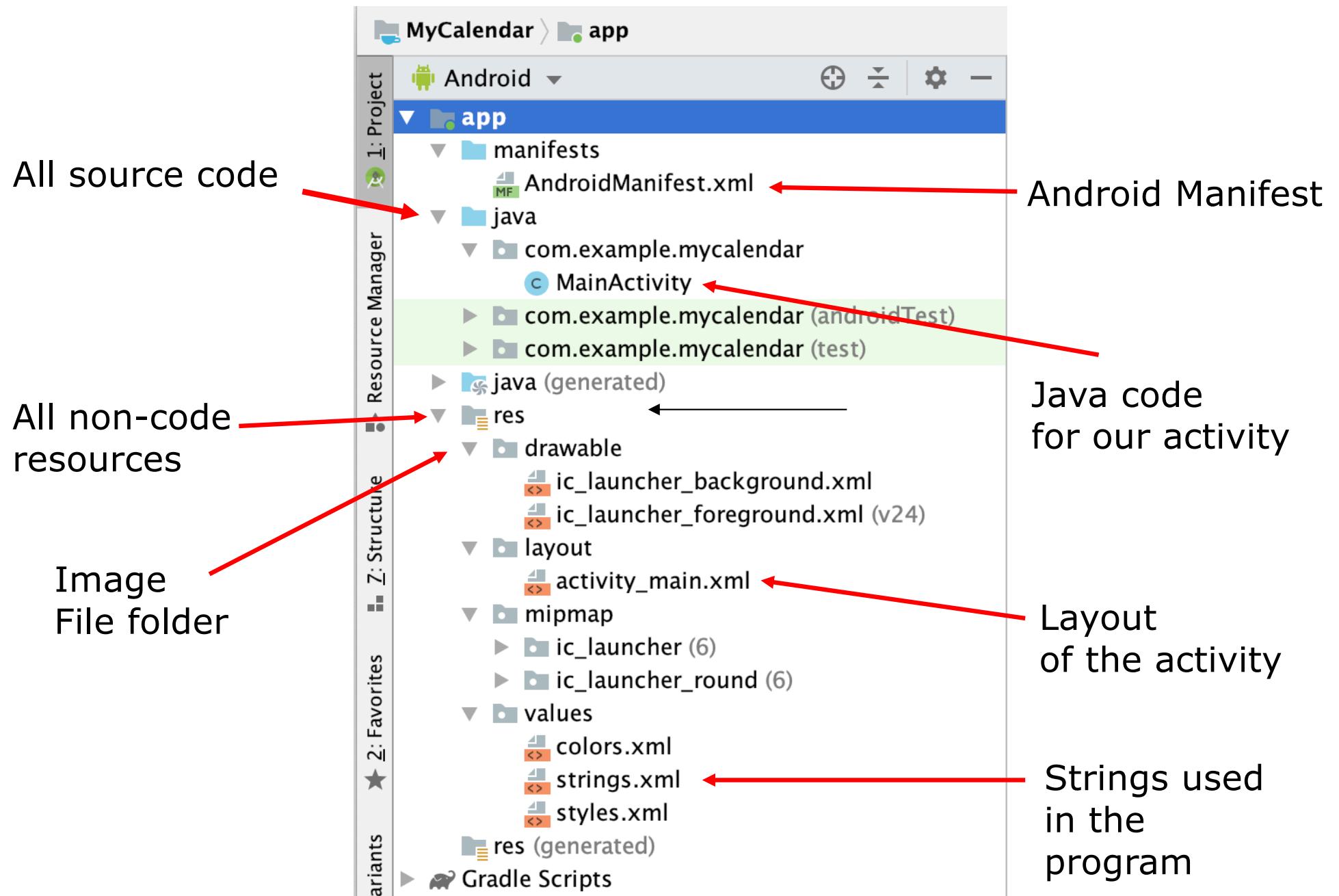


Run Your App on Your AVD

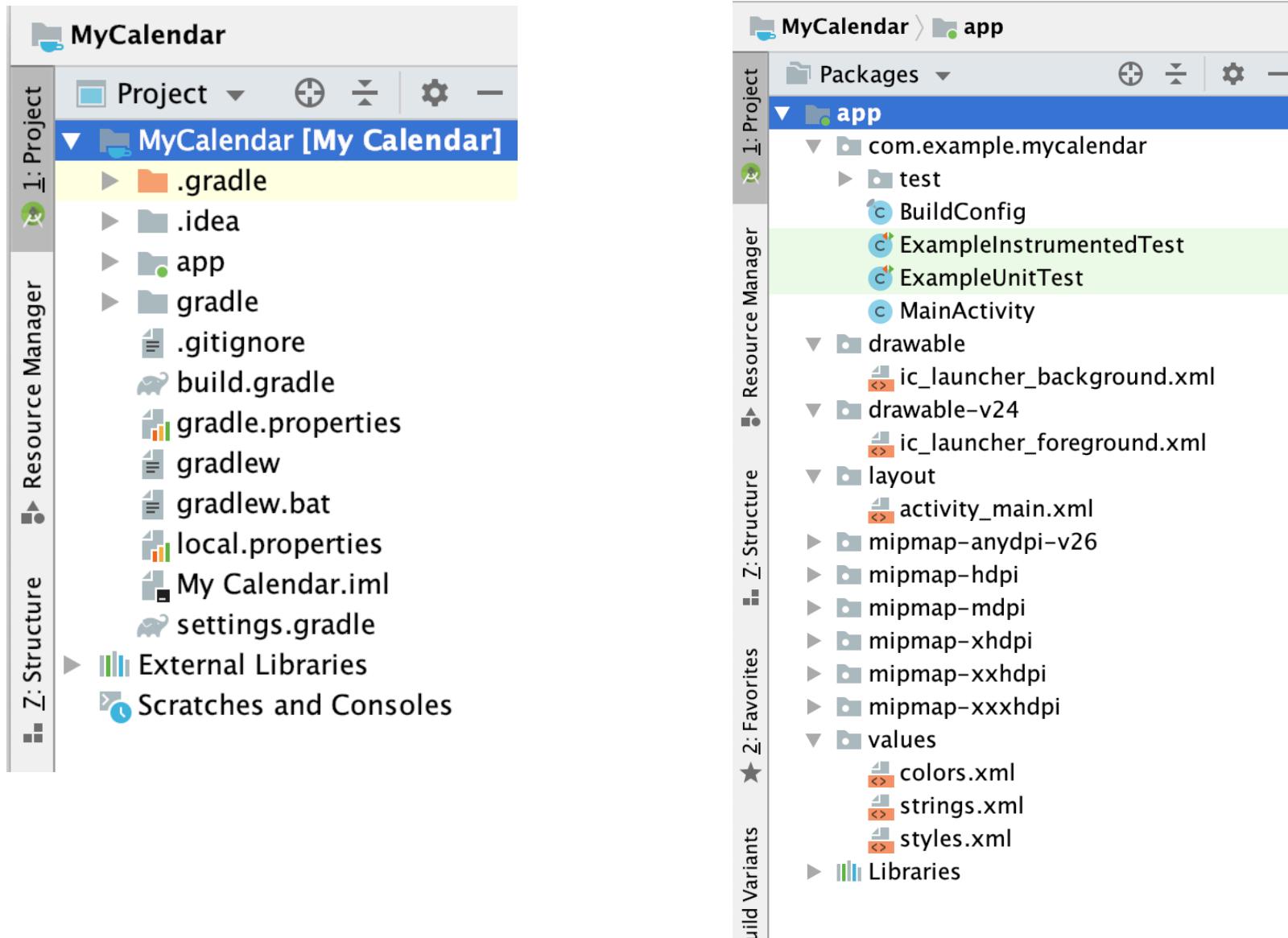
- Click [Run] button to install and run your app on your AVD



Project Window (Android View)



Project View and Package View



AndroidManifest.xml

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

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

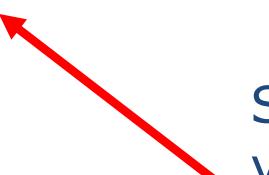
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

(Java)

MainActivity.java

```
package com.example.mycalendar;  
  
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);  
    }  
}
```



Set the layout of the view as described in the **activity_main.xml** layout

(Kotlin)

MainActivity.kt

```
package com.example.kotlin

import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle

import kotlinx.android.synthetic.main.activity_main.*

class MainActivity : AppCompatActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
    }
}
```

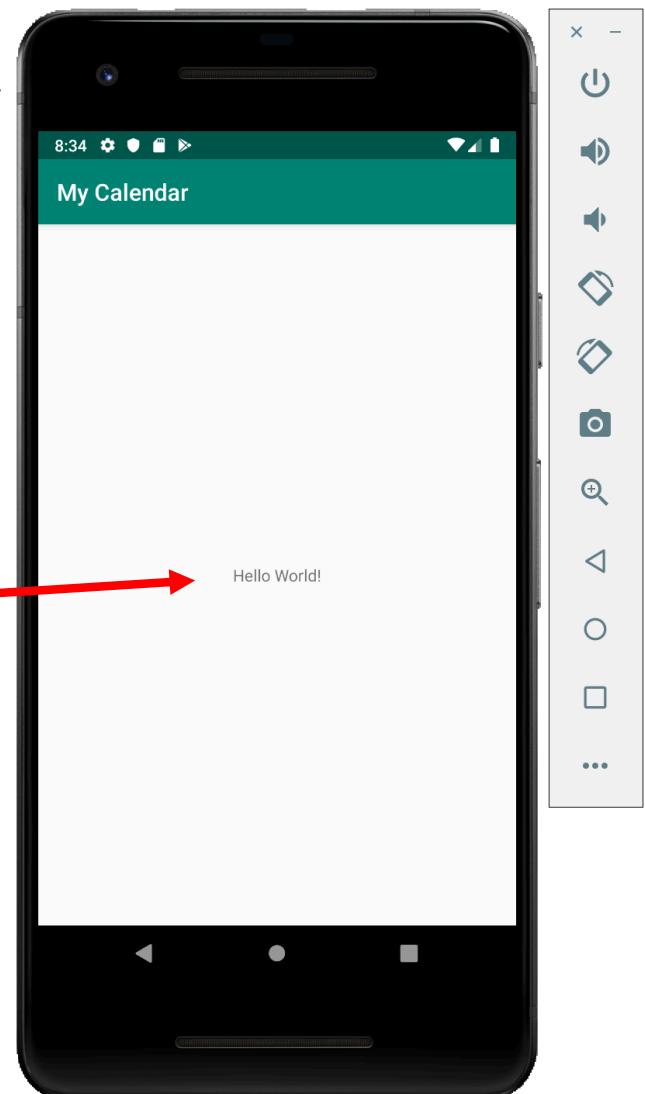
Set the layout of the view as described in the `activity_main.xml` layout

/res/layout/activity_main.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>
```



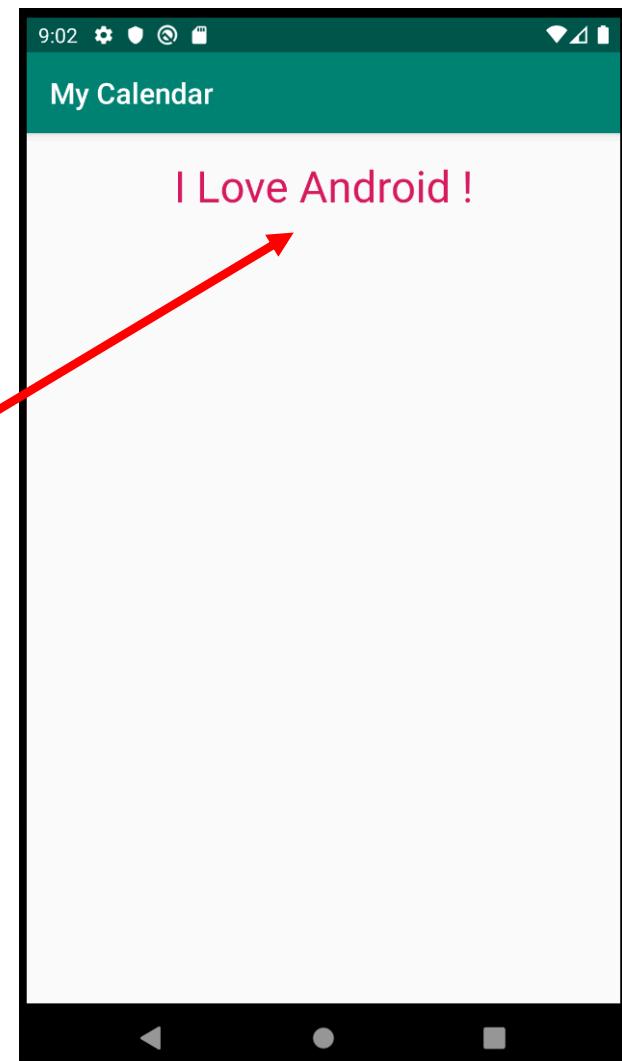
Run it!

I Love Android!

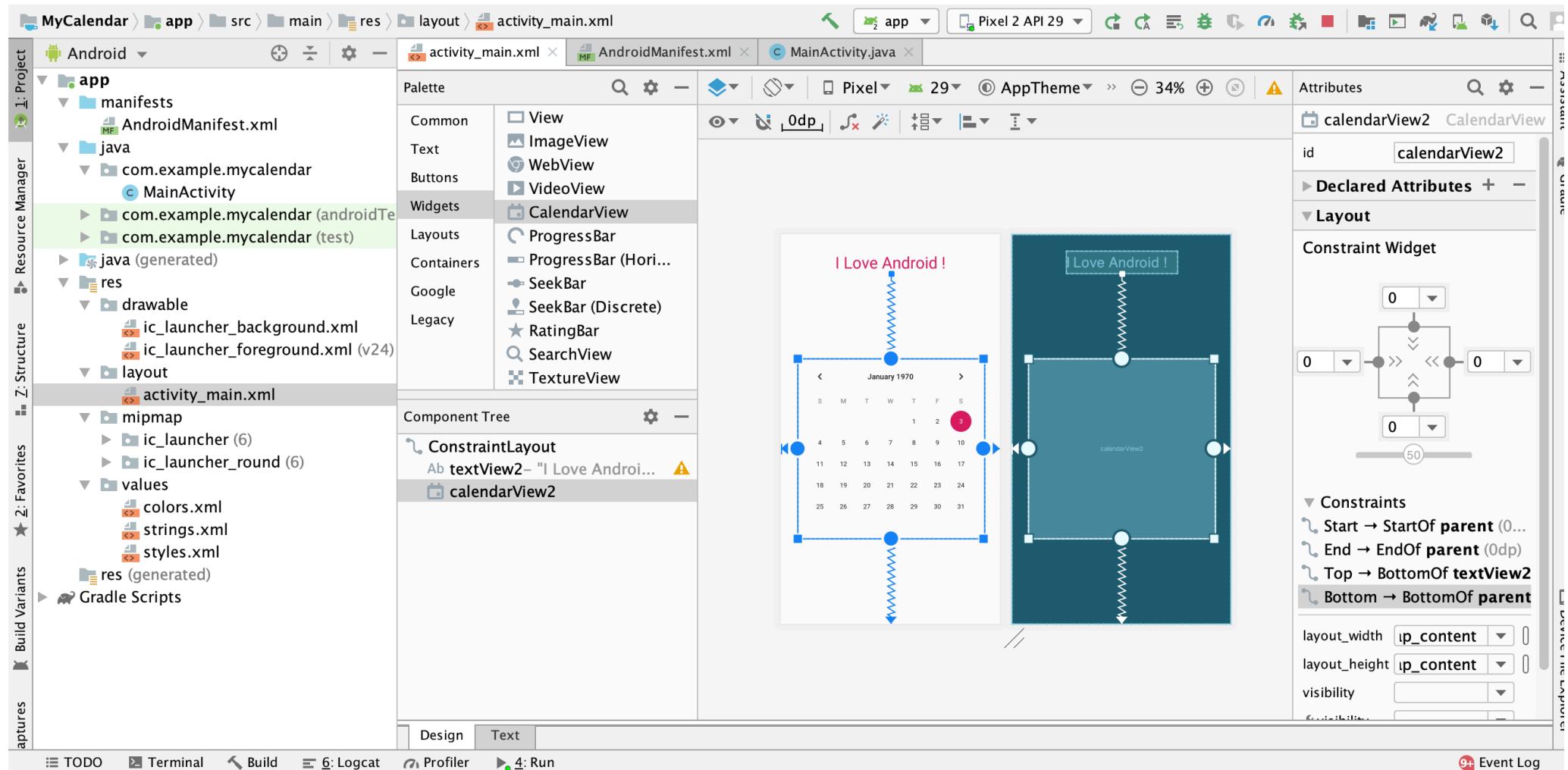
```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"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

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="16dp"
        android:text="I Love Android !"
        android:textColor="@color/colorAccent"
        android:textSize="30sp"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

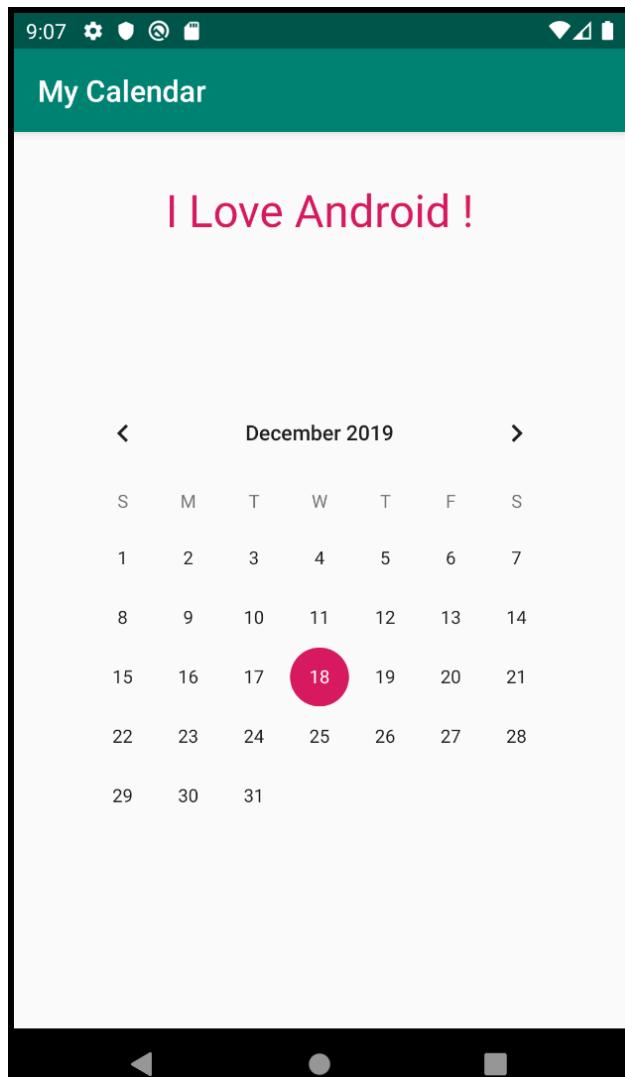
</androidx.constraintlayout.widget.ConstraintLayout>
```



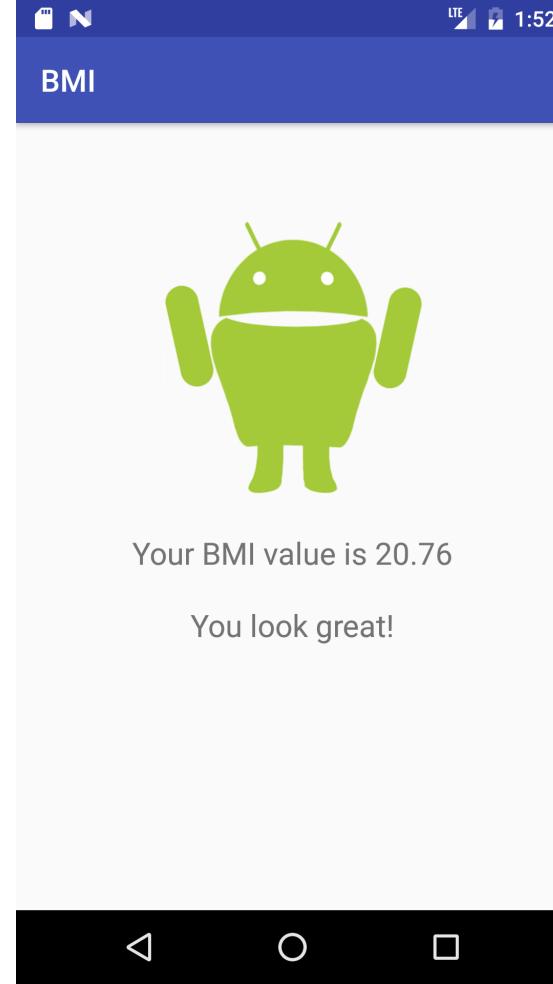
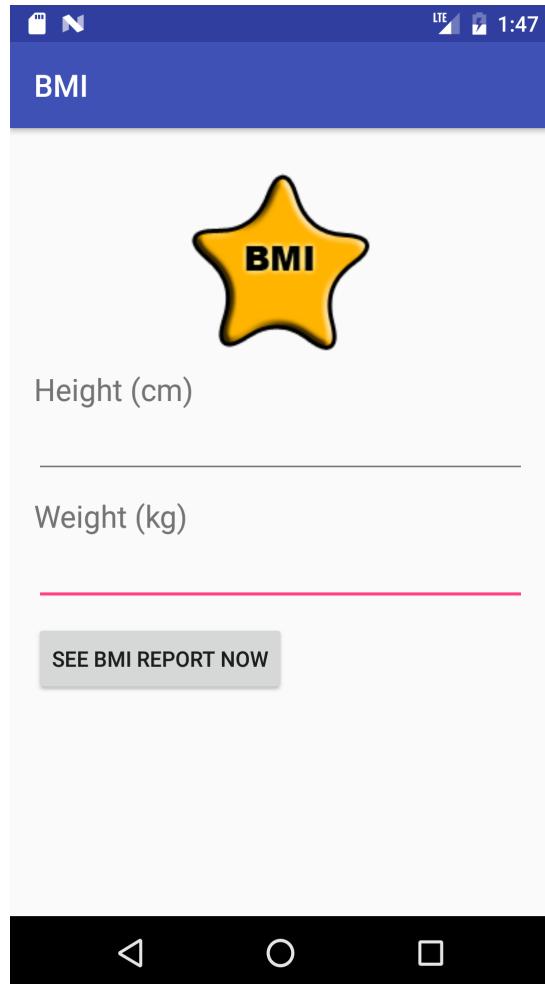
Add a CalendarView Object



Your Final My Calendar App



Next Week you will build a BMI App



Lab01 Homework

Select an Android app from the Google Play (<http://play.google.com>), which may be similar to an app that you want to develop in your course project. The purpose of this is to get your thinking about the course project early, to force you to dig into the Google Play and to look at other apps just to see what can be done in Android. Critically evaluate the selected apps. Submit **a 4-page evaluation report** with following items:

- The name of the app evaluated.
- A summary paragraph for the selected app, which identifies key features of the app.
- At least half page for identifying the target audience of the app, and evaluating the profit and/or humanitarian potential of the app (e.g. the expected market for the app, estimated number of paid downloads or estimated number of impressions/clicks per month if ad-based).
- A list of positive characteristics (e.g. high-quality graphics, fun, indispensable tool)
- A list of negative characteristics (e.g. force close, slow, confusing menu titles).

Lab01 Homework Submission

- Submit your evaluation in MS-Words format as attachment to course gmail account at ee5415@gmail.com on or before 11.00pm of **5th Feb 2020**. The subject of your email should satisfy the following format:
- EE5415 Lab 1 Homework – Student Name (Student Number)
- Example: EE5415 Lab 1 Homework – Chan Chi Man (51234567)