# NFC-connected Phone as a User Interface? There's an App For That! – Hands On

## Class 3: Building an Android Application from Scratch

September 25, 2019

Charles J. Lord, PE
President, Consultant, Trainer
Blue Ridge Advanced Design and Automation

Presented by:

DesignNews

Blue Ridge Advanced Design and Automation
Asheville, North Carolina

CEC CONTINUING EDUCATION CENTER

Digi-Key ELECTRONICS

# This Week’s Agenda

9/23 Introduction to the Project and Development Environment

9/24 An NFC Primer and Introducing the NXP NTAG

9/25 Building an Android Application from Scratch

9/26 Adding NFC Capability and Communications to Our App

9/27 Putting it All Together

# This Week’s Agenda

9/23 Introduction to the Project and Development Environment

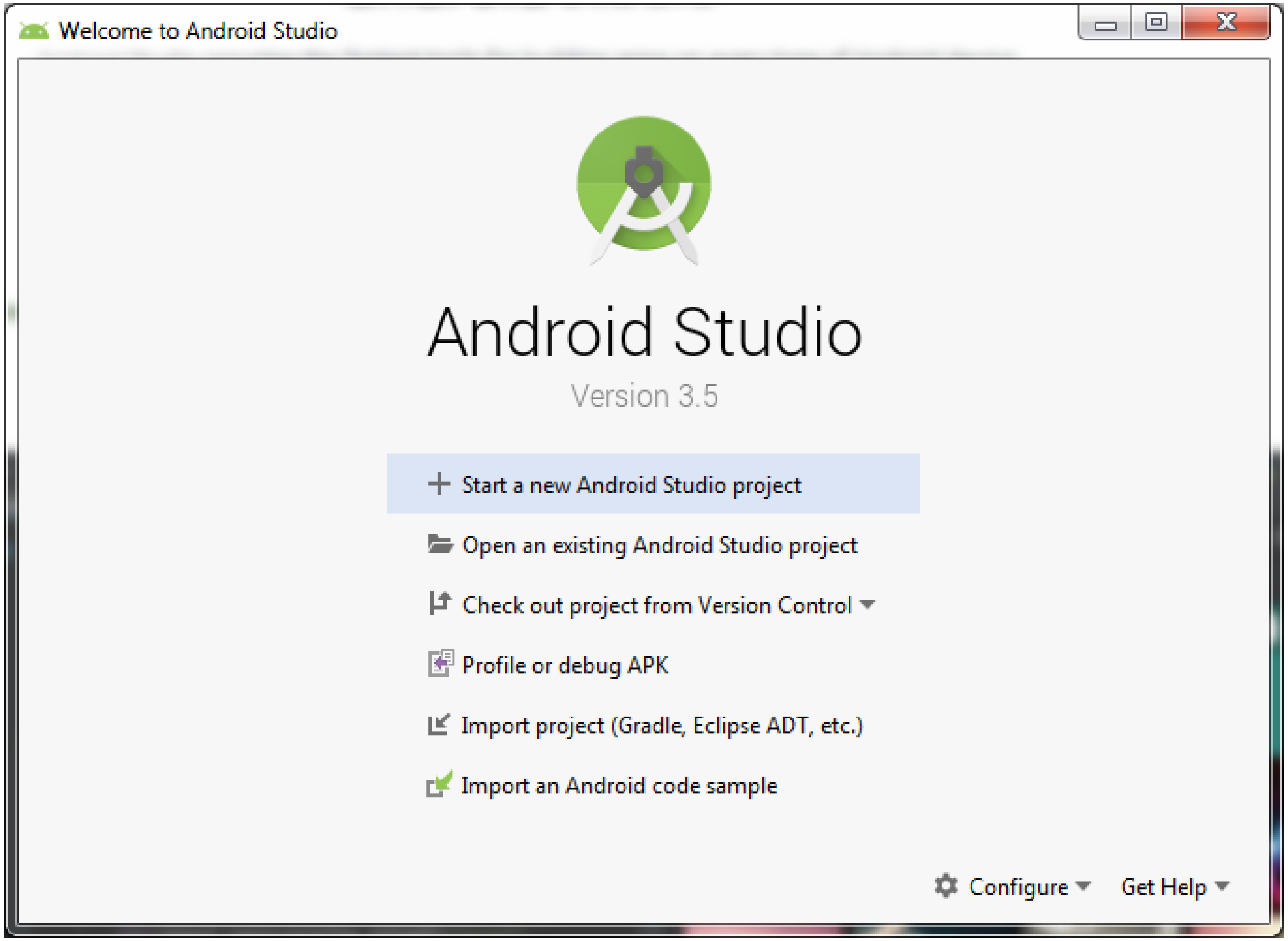
9/24 An NFC Primer and Introducing the NXP NTAG

9/25 Building an Android Application from Scratch

9/26 Adding NFC Capability and Communications to Our App

9/27 Putting it All Together

# From Monday…



Welcome to Android Studio

Android Studio

Version 3.5

+ Start a new Android Studio project
Open an existing Android Studio project
Check out project from Version Control
Profile or debug APK
Import project (Gradle, Eclipse ADT, etc.)
Import an Android code sample
Configure
Get Help

# Some beginning concepts

Question 1: Will you be running through the exercises on Android Development?

# Apps provide multiple entry points

- Android apps are built as a combination of components that can be invoked individually. For example, an activity is a type of app component that provides a user interface.
- The "main" activity is what starts when the user taps your app icon, but you can take the user straight into a different activity from other places, such as from a notification or even from a different app.
- Other components such as broadcast receivers and services also allow your app to perform background tasks without a user interface.

# Apps adapt to different devices

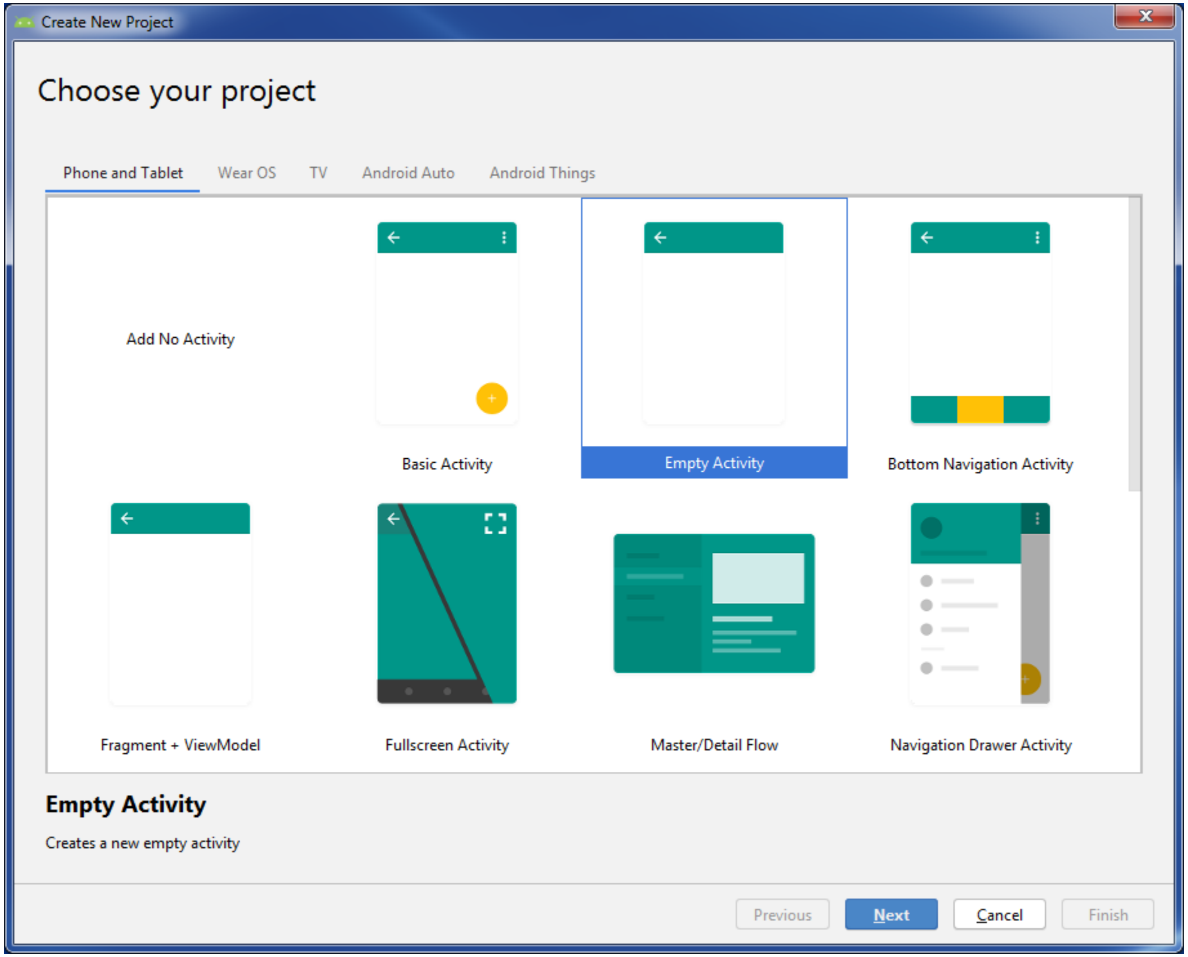
- Android allows you to provide different resources for different devices. For example, you can create different layouts for different screen sizes. Then the system determines which layout to use based on the current device's screen size.
- If any of your app's features need specific hardware, such as a NFC, you can query whether the device has that hardware at runtime and then disable the corresponding features if not. You can also set some features as required so Google Play won't allow installation on devices without them.

# Studio Supports Two Languages

- Java
- Kotlin (OOPL derived from Java)

- Most examples are in Java

# Choose Empty Activity



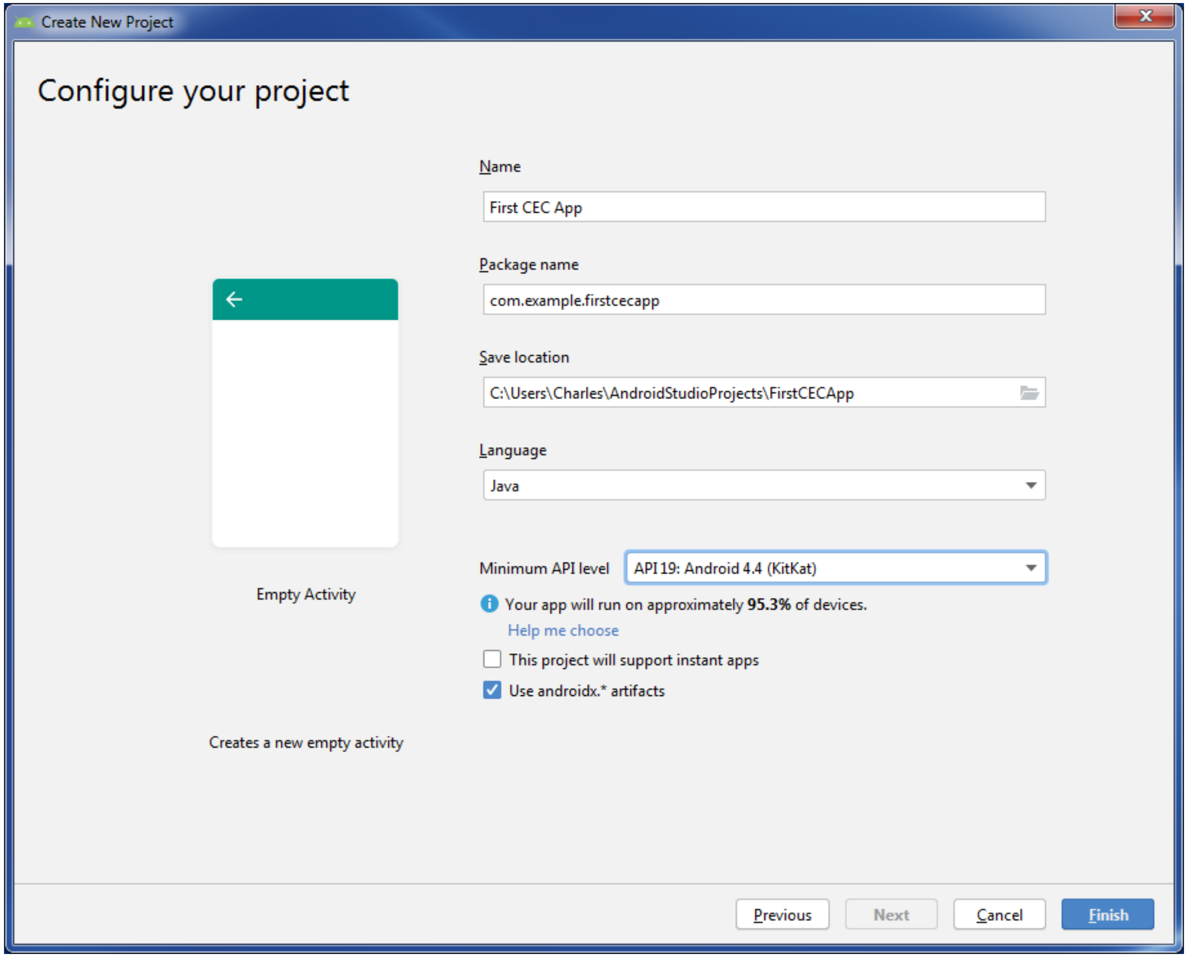
Create New Project

Choose your project

**Empty Activity**

Creates a new empty activity

# Name our Project



Configure your project

Creates a new empty activity

Name: First CEC App

Package name: com.example.firstcecapp

Save location: C:\Users\Charles\AndroidStudioProjects\FirstCECApp

Language: Java

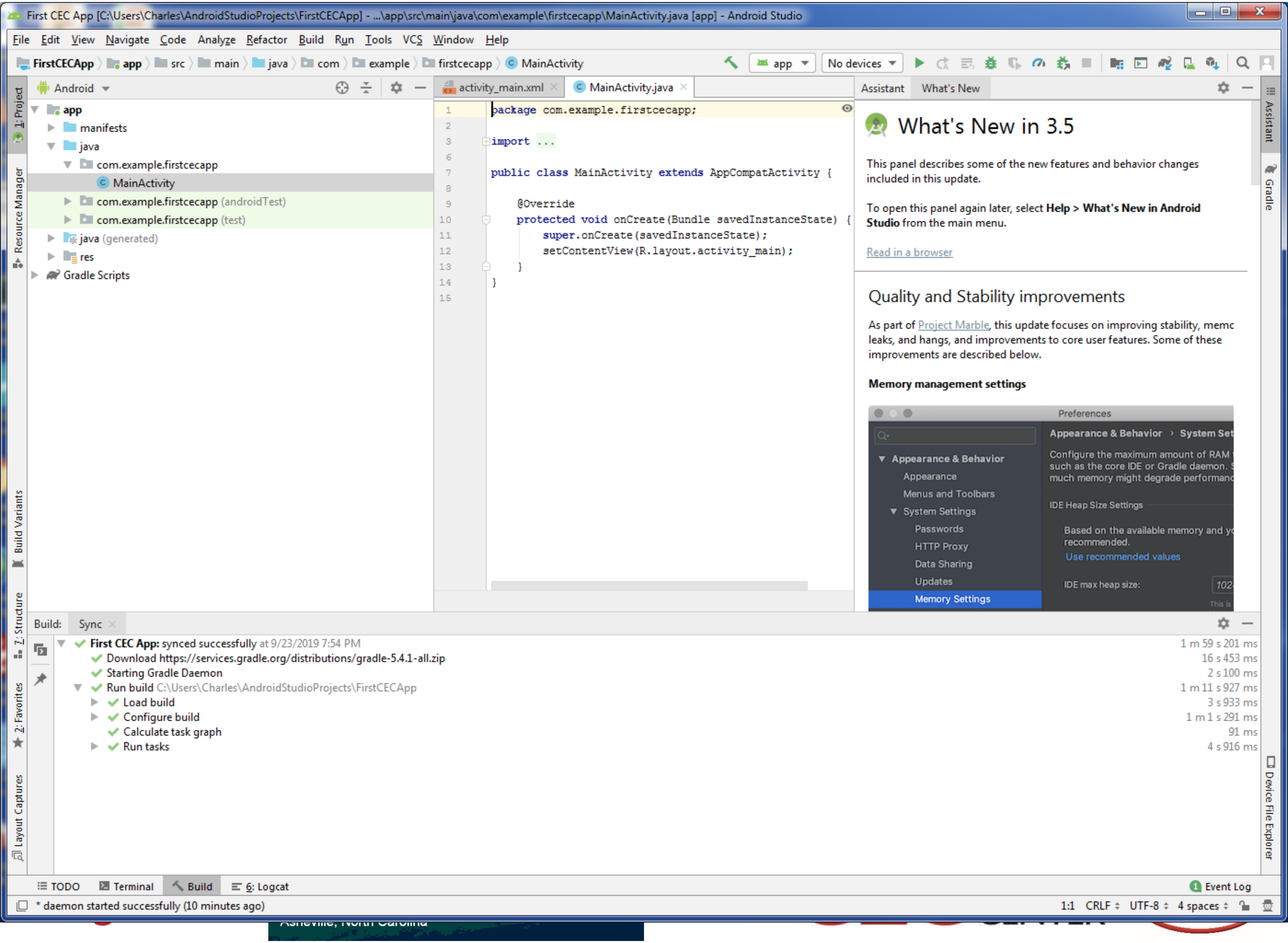
Minimum API level: API 19: Android 4.4 (KitKat)

Your app will run on approximately **95.3%** of devices.

Help me choose

☐ This project will support instant apps

☑ Use androidx.* artifacts


First CEC App [C:\Users\Charles\AndroidStudioProjects\FirstCECApp] - ...\app\src\main\java\com\example\firstcecapp\MainActivity.java [app] - Android Studio
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
FirstCECApp 〉 app 〉 src 〉 main 〉 java 〉 com 〉 example 〉 firstcecapp 〉 MainActivity
app
No devices
Android
app
manifests
java
com.example.firstcecapp
MainActivity
com.example.firstcecapp (androidTest)
com.example.firstcecapp (test)
java (generated)
res
Gradle Scripts
activity_main.xml
MainActivity.java
Assistant
What's New
1: Project
Resource Manager
Build Variants
Z: Structure
2: Favorites
Layout Captures
Gradle
Device File Explorer
Build:
Sync
TODO
Terminal
Build
6: Logcat
Event Log
1:1 CRLF UTF-8 4 spaces

```java
package com.example.firstcecapp;

import ...

public class MainActivity extends AppCompatActivity {

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

## What's New in 3.5

This panel describes some of the new features and behavior changes included in this update.

To open this panel again later, select **Help > What's New in Android Studio** from the main menu.

Read in a browser

### Quality and Stability improvements

As part of Project Marble, this update focuses on improving stability, memc leaks, and hangs, and improvements to core user features. Some of these improvements are described below.

**Memory management settings**

Configure the maximum amount of RAM such as the core IDE or Gradle daemon. much memory might degrade performanc

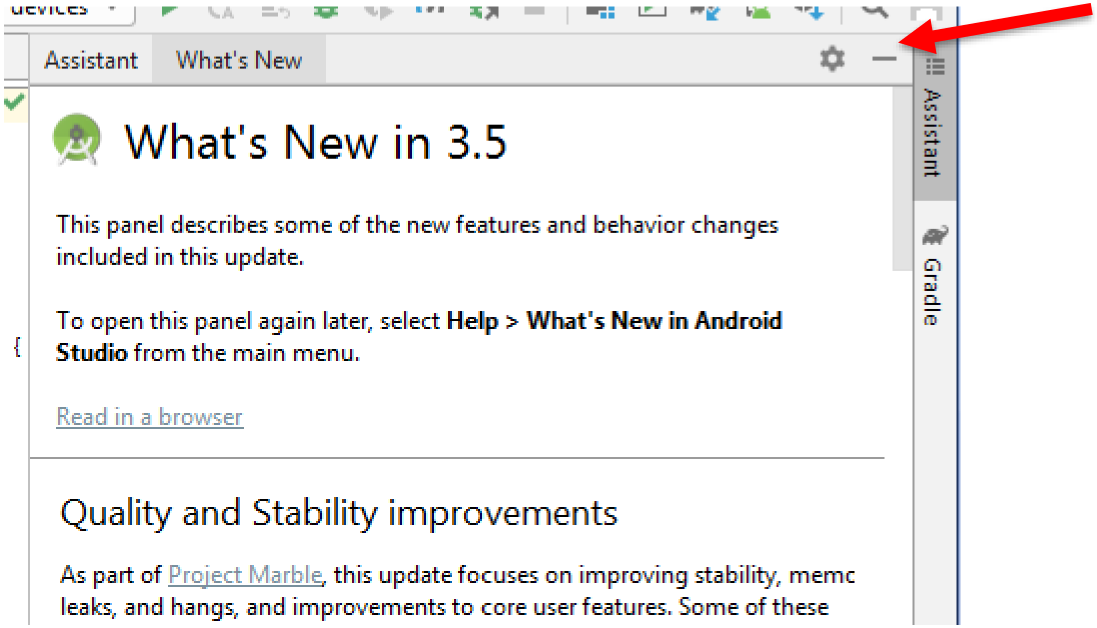
Based on the available memory and y recommended.

```
First CEC App: synced successfully at 9/23/2019 7:54 PM                         1 m 59 s 201 ms
  ✓ Download https://services.gradle.org/distributions/gradle-5.4.1-all.zip     16 s 453 ms
  ✓ Starting Gradle Daemon                                                       2 s 100 ms
  ✓ Run build C:\Users\Charles\AndroidStudioProjects\FirstCECApp               1 m 11 s 927 ms
      ✓ Load build                                                               3 s 933 ms
      ✓ Configure build                                                        1 m 1 s 291 ms
      ✓ Calculate task graph                                                         91 ms
      ✓ Run tasks                                                                4 s 916 ms
```

* daemon started successfully (10 minutes ago)

Asheville, North Carolina

# Collapse!



## What's New in 3.5

This panel describes some of the new features and behavior changes included in this update.

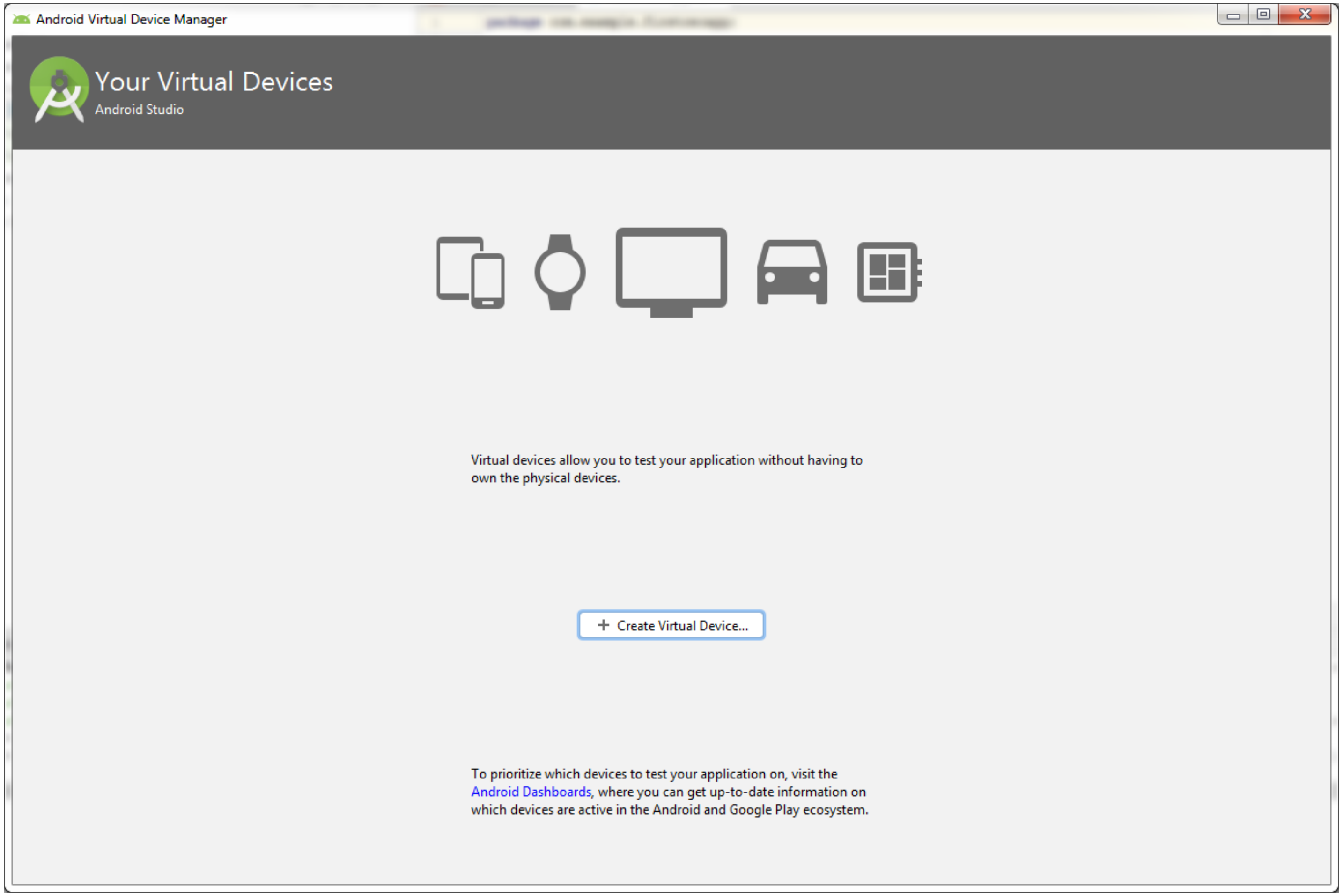
To open this panel again later, select **Help > What's New in Android Studio** from the main menu.

Read in a browser

## Quality and Stability improvements

As part of Project Marble, this update focuses on improving stability, memc leaks, and hangs, and improvements to core user features. Some of these

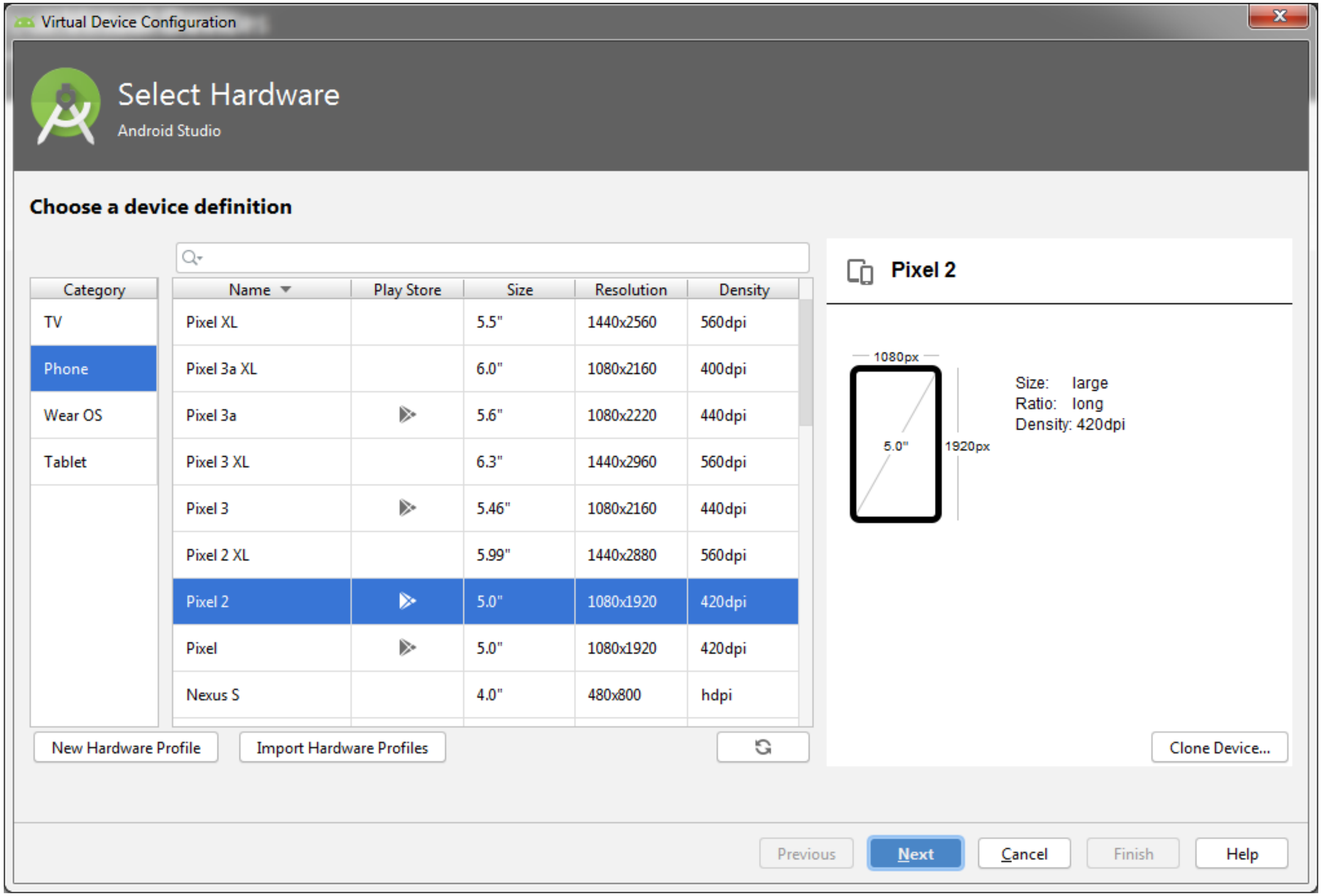
# Let’s build a virtual device



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.

# My Samsung Galaxy S5 isn’t here



Choose a device definition

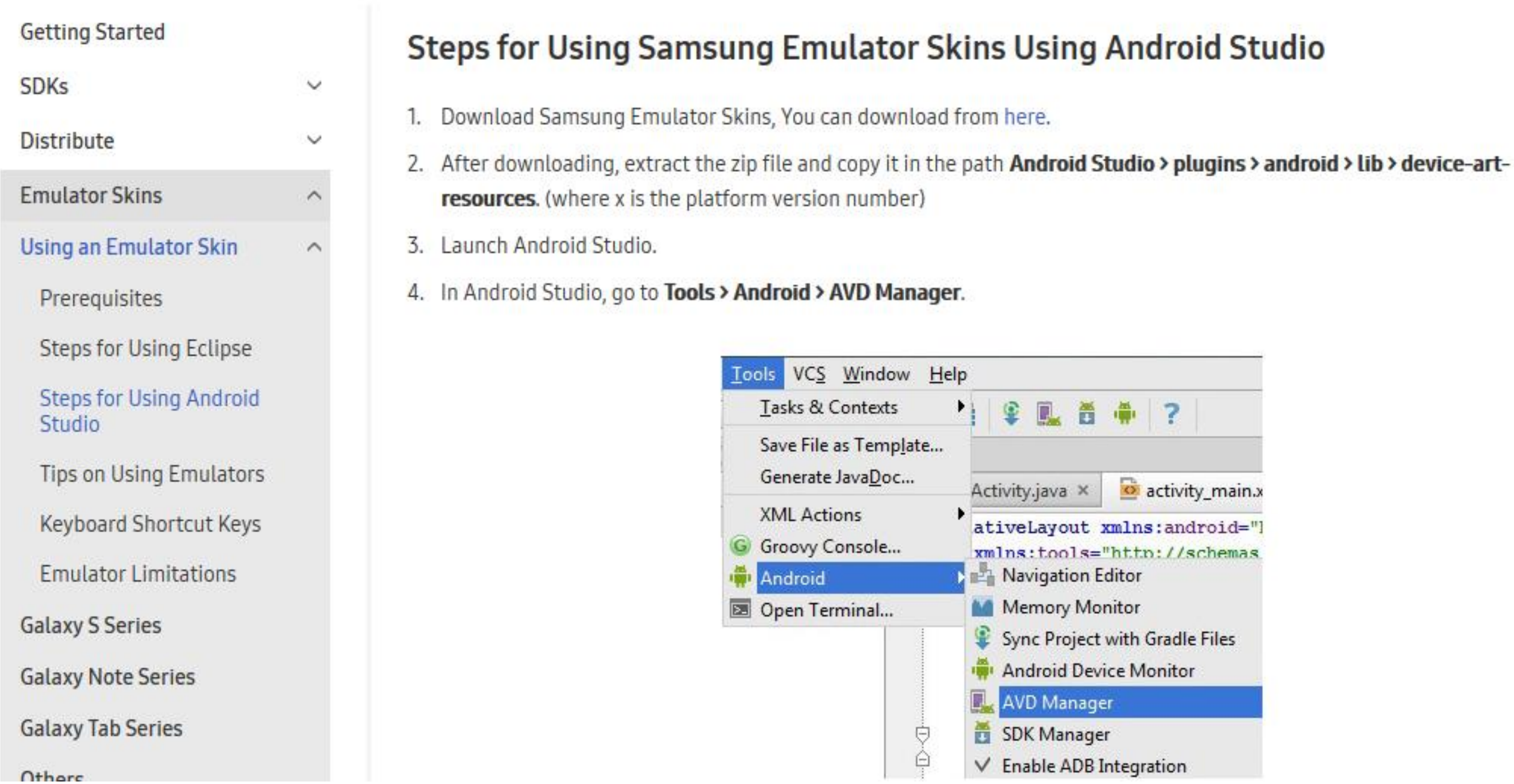
| Category |
|---|
| TV |
| Phone |
| Wear OS |
| Tablet |

| Name | Play Store | Size | Resolution | Density |
|---|---|---|---|---|
| Pixel XL | | 5.5" | 1440x2560 | 560dpi |
| Pixel 3a XL | | 6.0" | 1080x2160 | 400dpi |
| Pixel 3a | ▶ | 5.6" | 1080x2220 | 440dpi |
| Pixel 3 XL | | 6.3" | 1440x2960 | 560dpi |
| Pixel 3 | ▶ | 5.46" | 1080x2160 | 440dpi |
| Pixel 2 XL | | 5.99" | 1440x2880 | 560dpi |
| Pixel 2 | ▶ | 5.0" | 1080x1920 | 420dpi |
| Pixel | ▶ | 5.0" | 1080x1920 | 420dpi |
| Nexus S | | 4.0" | 480x800 | hdpi |

Pixel 2

Size: large
Ratio: long
Density: 420dpi

# We go to developer.samsung.com

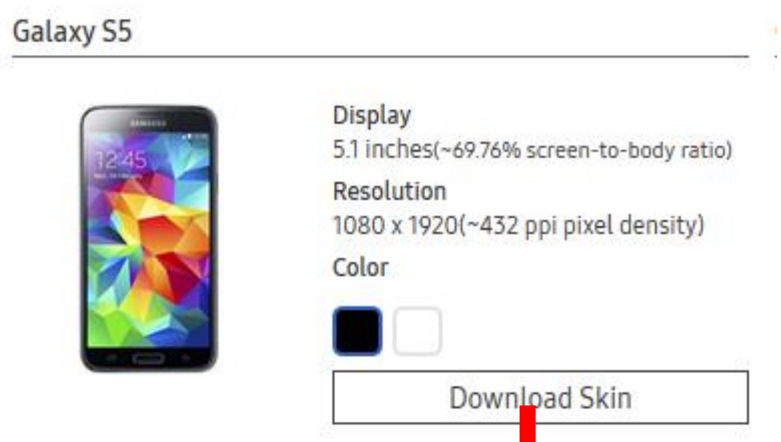


- Getting Started
- SDKs
- Distribute
- Emulator Skins
- Using an Emulator Skin
  - Prerequisites
  - Steps for Using Eclipse
  - Steps for Using Android Studio
  - Tips on Using Emulators
  - Keyboard Shortcut Keys
  - Emulator Limitations
- Galaxy S Series
- Galaxy Note Series
- Galaxy Tab Series
- Others

## Steps for Using Samsung Emulator Skins Using Android Studio

1. Download Samsung Emulator Skins, You can download from here.
2. After downloading, extract the zip file and copy it in the path **Android Studio › plugins › android › lib › device-art-resources**. (where x is the platform version number)
3. Launch Android Studio.
4. In Android Studio, go to **Tools › Android › AVD Manager**.

# Take note of the parameters



Galaxy S5

Display

5.1 inches(~69.76% screen-to-body ratio)

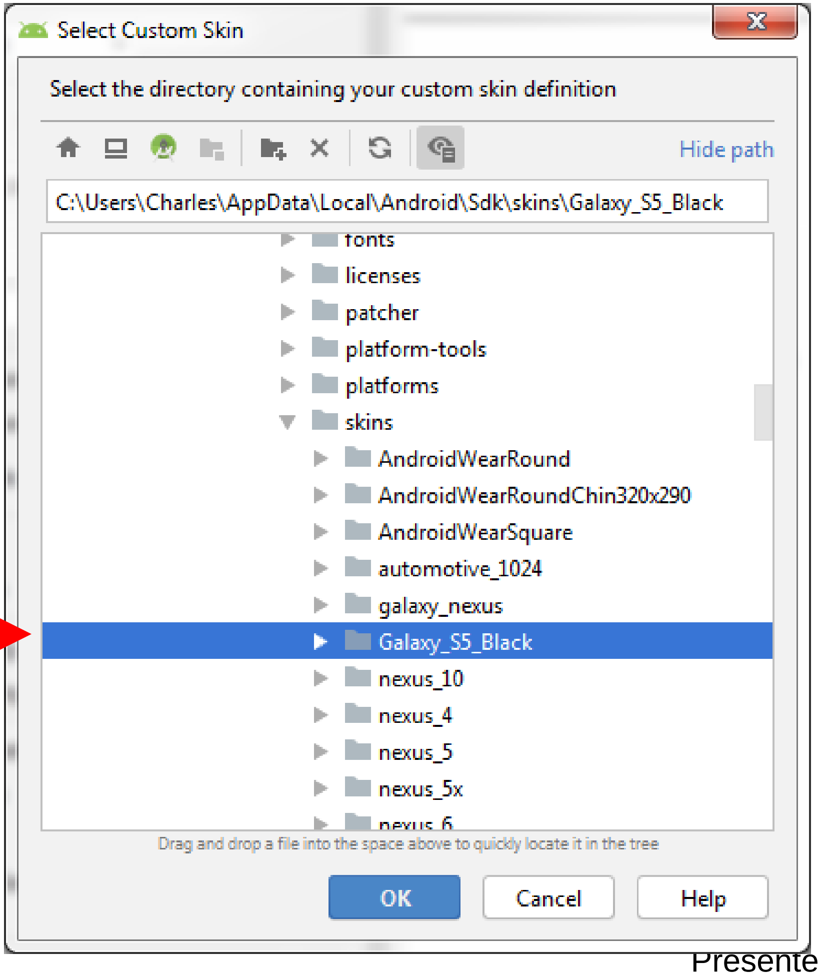
Resolution

1080 x 1920(~432 ppi pixel density)

Color

■ □

Unzip



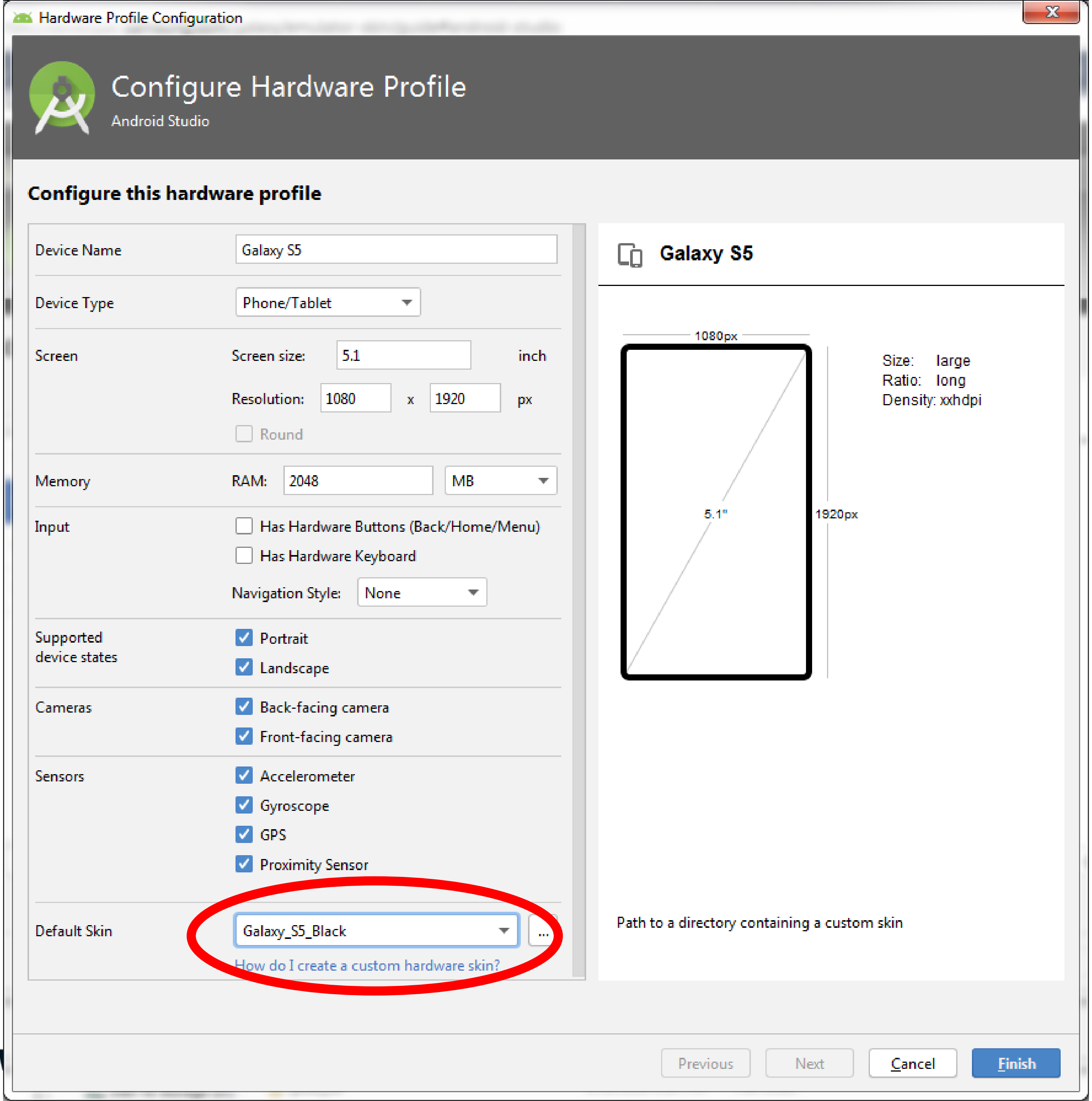
Select the directory containing your custom skin definition

C:\Users\Charles\AppData\Local\Android\Sdk\skins\Galaxy_S5_Black

- fonts
- licenses
- patcher
- platform-tools
- platforms
- skins
  - AndroidWearRound
  - AndroidWearRoundChin320x290
  - AndroidWearSquare
  - automotive_1024
  - galaxy_nexus
  - Galaxy_S5_Black
  - nexus_10
  - nexus_4
  - nexus_5
  - nexus_5x
  - nexus_6

Drag and drop a file into the space above to quickly locate it in the tree

Presented by:



Hardware Profile Configuration

Configure Hardware Profile
Android Studio

**Configure this hardware profile**

| | | | | |
|---|---|---|---|---|
| Device Name | Galaxy S5 | | | |
| Device Type | Phone/Tablet | | | |
| Screen | Screen size: | 5.1 | | inch |
| | Resolution: | 1080 | x 1920 | px |
| | □ Round | | | |
| Memory | RAM: | 2048 | MB | |
| Input | □ Has Hardware Buttons (Back/Home/Menu) | | | |
| | □ Has Hardware Keyboard | | | |
| | Navigation Style: | None | | |
| Supported device states | ☑ Portrait | | | |
| | ☑ Landscape | | | |
| Cameras | ☑ Back-facing camera | | | |
| | ☑ Front-facing camera | | | |
| Sensors | ☑ Accelerometer | | | |
| | ☑ Gyroscope | | | |
| | ☑ GPS | | | |
| | ☑ Proximity Sensor | | | |
| Default Skin | Galaxy_S5_Black | ... | | |

How do I create a custom hardware skin?

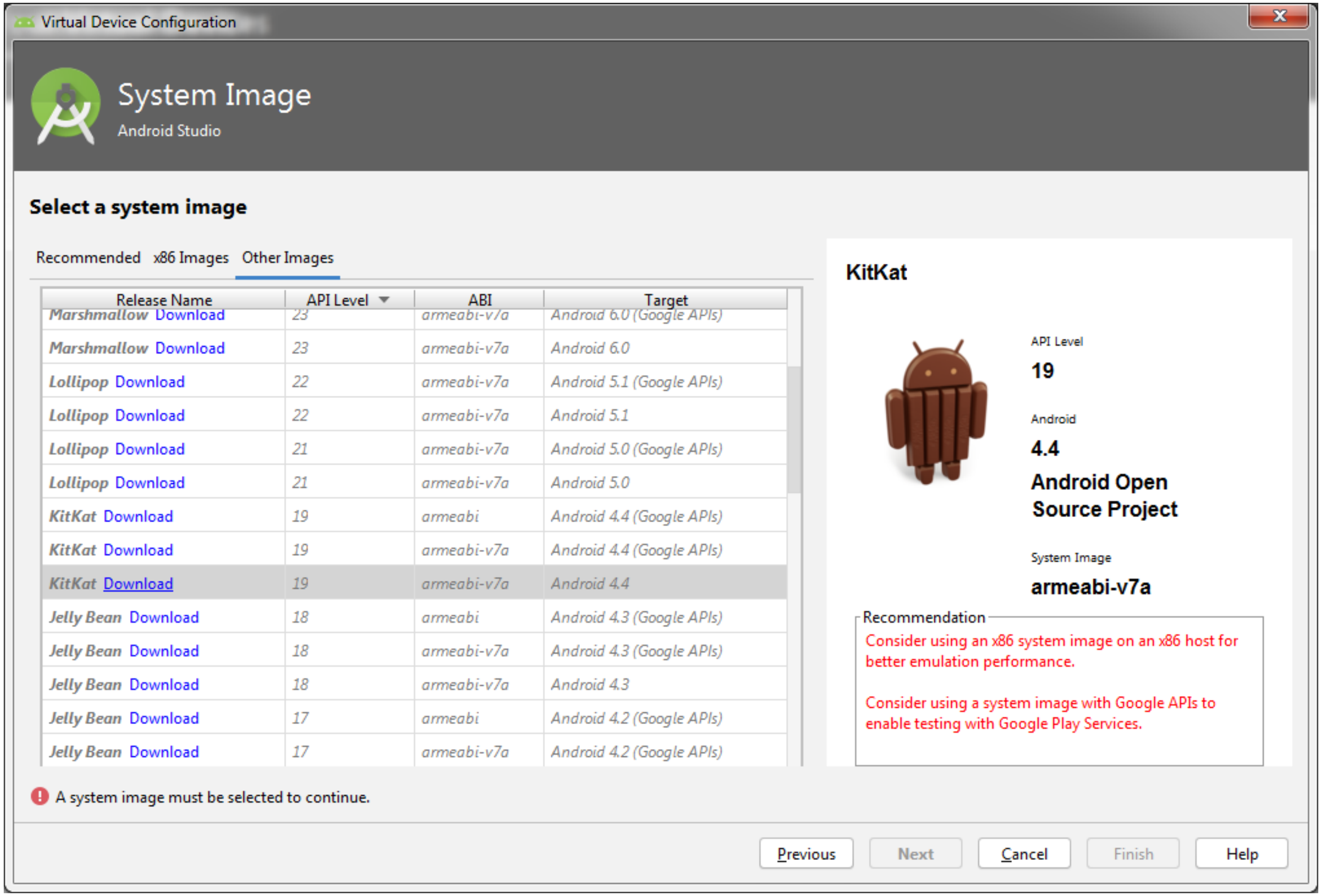
**Galaxy S5**

Size: large
Ratio: long
Density: xxhdpi

Path to a directory containing a custom skin

Presented by:

# Our target OS version no longer recommended



**Select a system image**

| Release Name | API Level | ABI | Target |
|---|---|---|---|
| Marshmallow Download | 23 | armeabi-v7a | Android 6.0 (Google APIs) |
| Marshmallow Download | 23 | armeabi-v7a | Android 6.0 |
| Lollipop Download | 22 | armeabi-v7a | Android 5.1 (Google APIs) |
| Lollipop Download | 22 | armeabi-v7a | Android 5.1 |
| Lollipop Download | 21 | armeabi-v7a | Android 5.0 (Google APIs) |
| Lollipop Download | 21 | armeabi-v7a | Android 5.0 |
| KitKat Download | 19 | armeabi | Android 4.4 (Google APIs) |
| KitKat Download | 19 | armeabi-v7a | Android 4.4 (Google APIs) |
| KitKat Download | 19 | armeabi-v7a | Android 4.4 |
| Jelly Bean Download | 18 | armeabi | Android 4.3 (Google APIs) |
| Jelly Bean Download | 18 | armeabi-v7a | Android 4.3 (Google APIs) |
| Jelly Bean Download | 18 | armeabi-v7a | Android 4.3 |
| Jelly Bean Download | 17 | armeabi | Android 4.2 (Google APIs) |
| Jelly Bean Download | 17 | armeabi-v7a | Android 4.2 (Google APIs) |

**KitKat**

API Level
**19**

Android
**4.4**
**Android Open Source Project**

System Image
**armeabi-v7a**

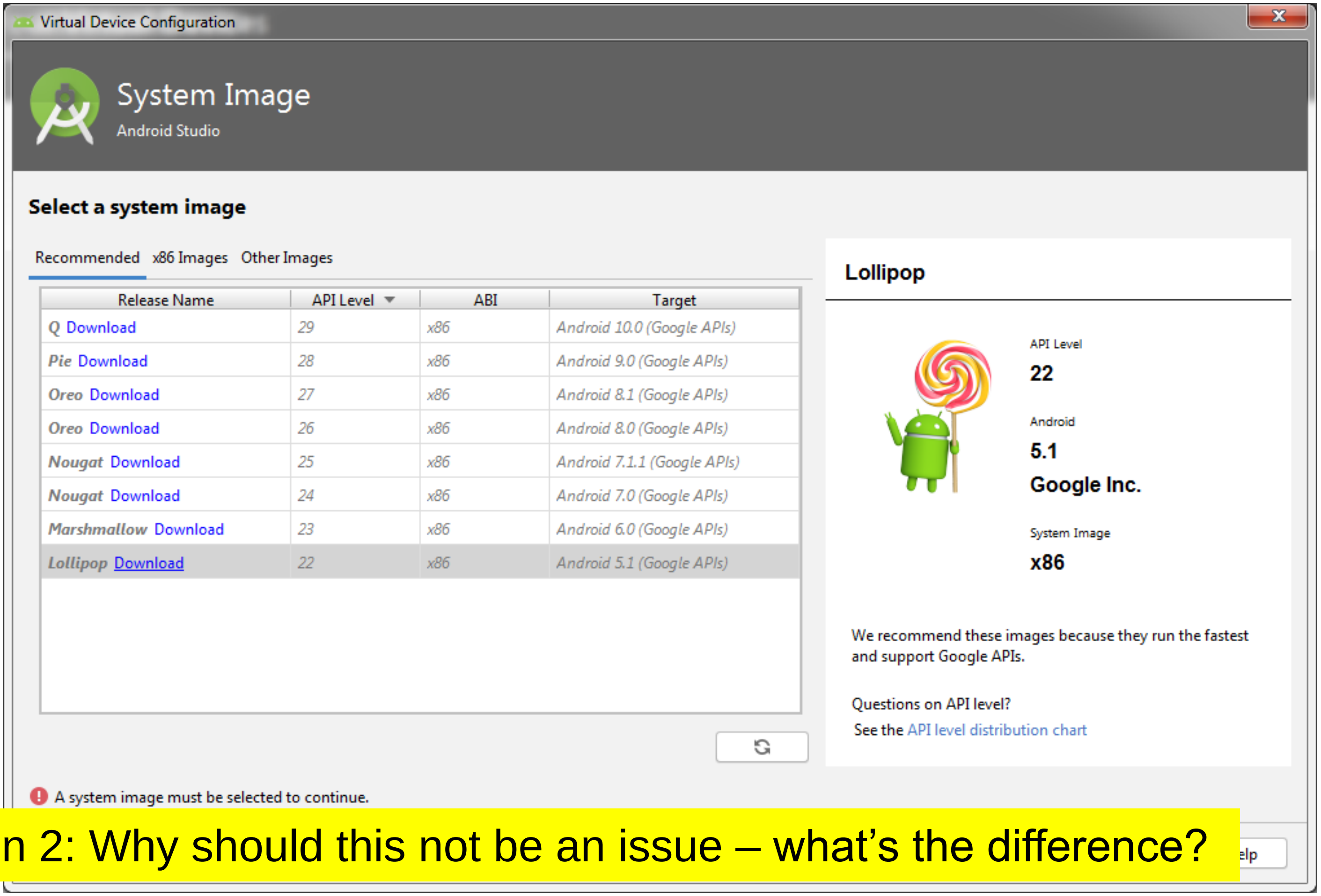
Recommendation

Consider using an x86 system image on an x86 host for better emulation performance.

Consider using a system image with Google APIs to enable testing with Google Play Services.

A system image must be selected to continue.

# We will use Lollypop – still 85% coverage



Virtual Device Configuration

System Image

Android Studio

**Select a system image**

Recommended | x86 Images | Other Images

| 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 Download | 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.

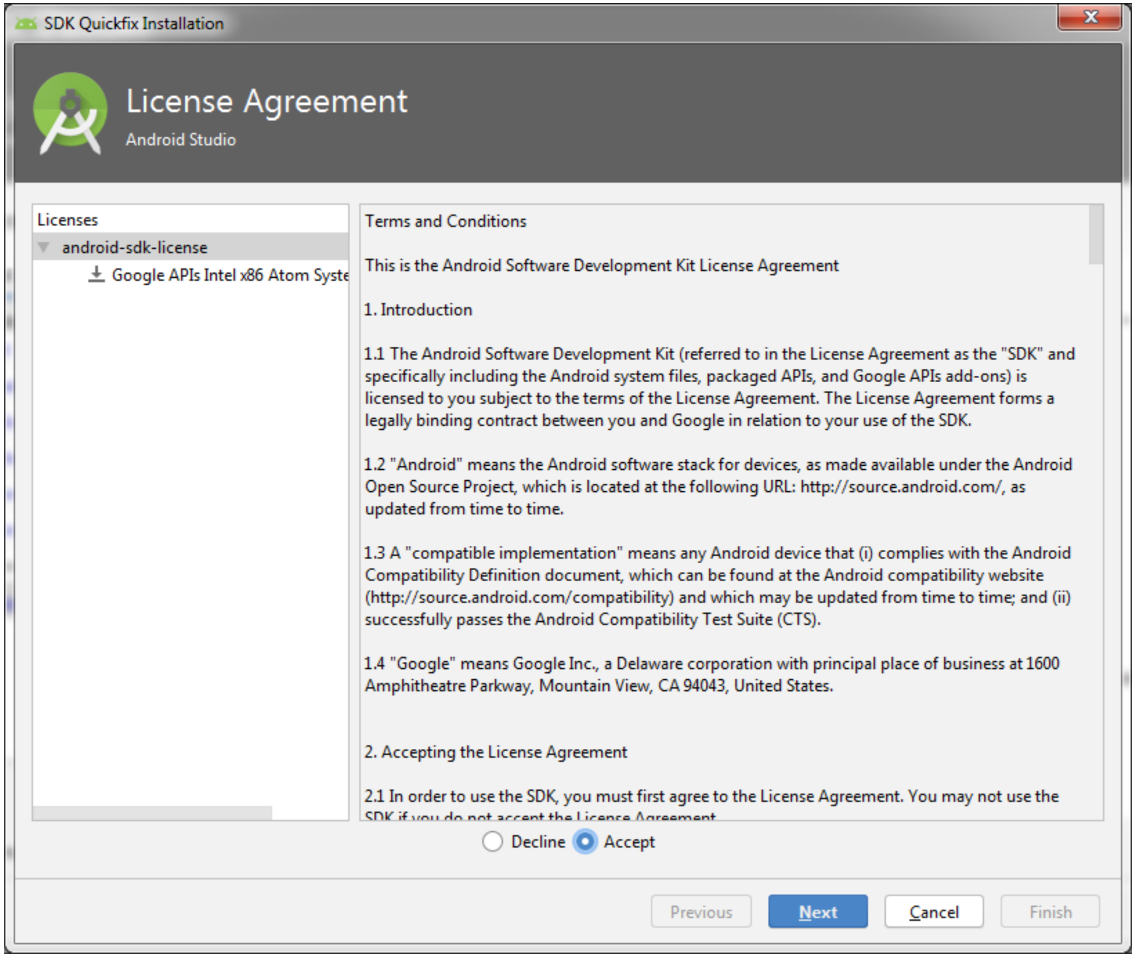
Questions on API level?
See the API level distribution chart

A system image must be selected to continue.

Question 2: Why should this not be an issue – what's the difference?

Presented by:

# Agree to etc etc for OS



License Agreement

Android Studio

Licenses

- android-sdk-license
  - Google APIs Intel x86 Atom Syste

Terms and Conditions

This is the Android Software Development Kit License Agreement

1. Introduction

1.1 The Android Software Development Kit (referred to in the License Agreement as the "SDK" and specifically including the Android system files, packaged APIs, and Google APIs add-ons) is licensed to you subject to the terms of the License Agreement. The License Agreement forms a legally binding contract between you and Google in relation to your use of the SDK.

1.2 "Android" means the Android software stack for devices, as made available under the Android Open Source Project, which is located at the following URL: http://source.android.com/, as updated from time to time.

1.3 A "compatible implementation" means any Android device that (i) complies with the Android Compatibility Definition document, which can be found at the Android compatibility website (http://source.android.com/compatibility) and which may be updated from time to time; and (ii) successfully passes the Android Compatibility Test Suite (CTS).

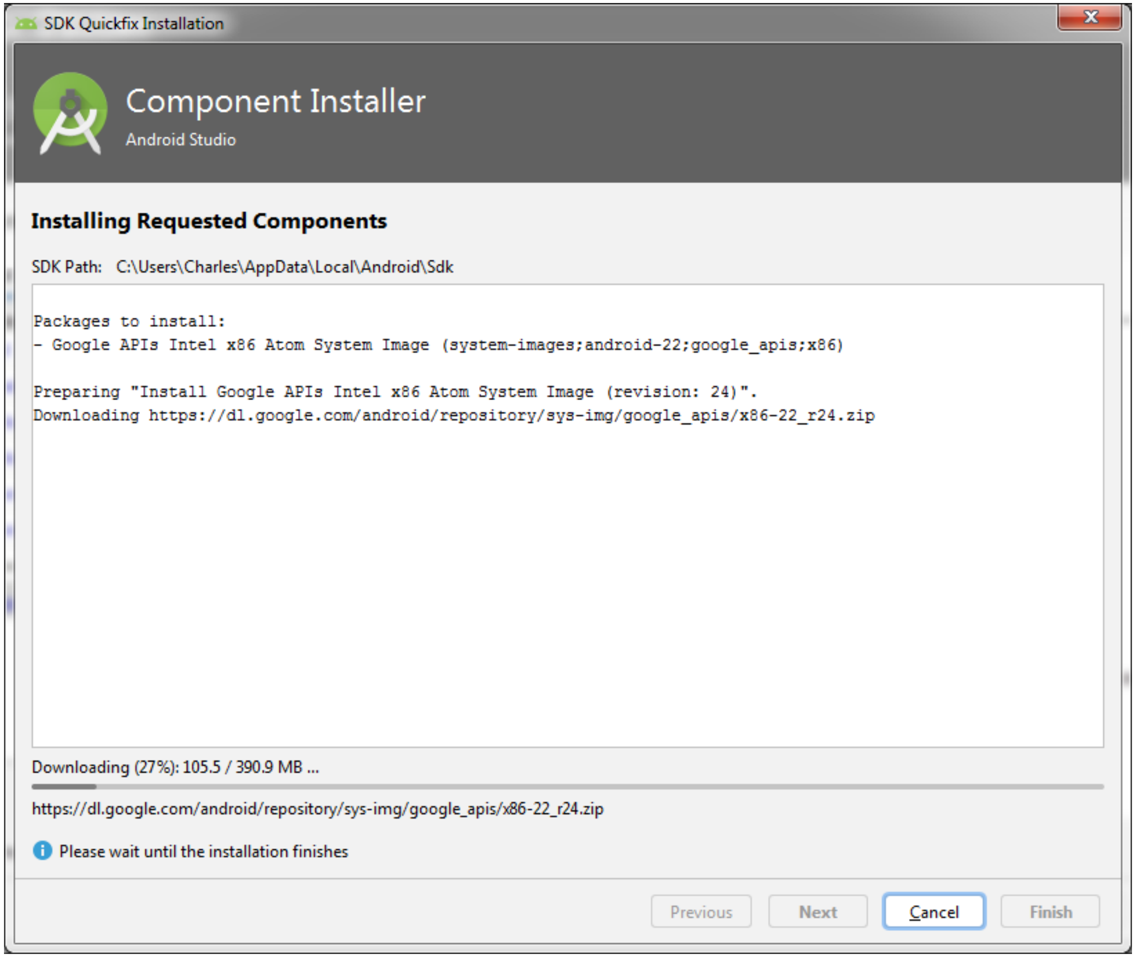
1.4 "Google" means Google Inc., a Delaware corporation with principal place of business at 1600 Amphitheatre Parkway, Mountain View, CA 94043, United States.

2. Accepting the License Agreement

2.1 In order to use the SDK, you must first agree to the License Agreement. You may not use the SDK if you do not accept the License Agreement.

○ Decline ● Accept

# This takes a few minutes



**Installing Requested Components**

SDK Path: C:\Users\Charles\AppData\Local\Android\Sdk

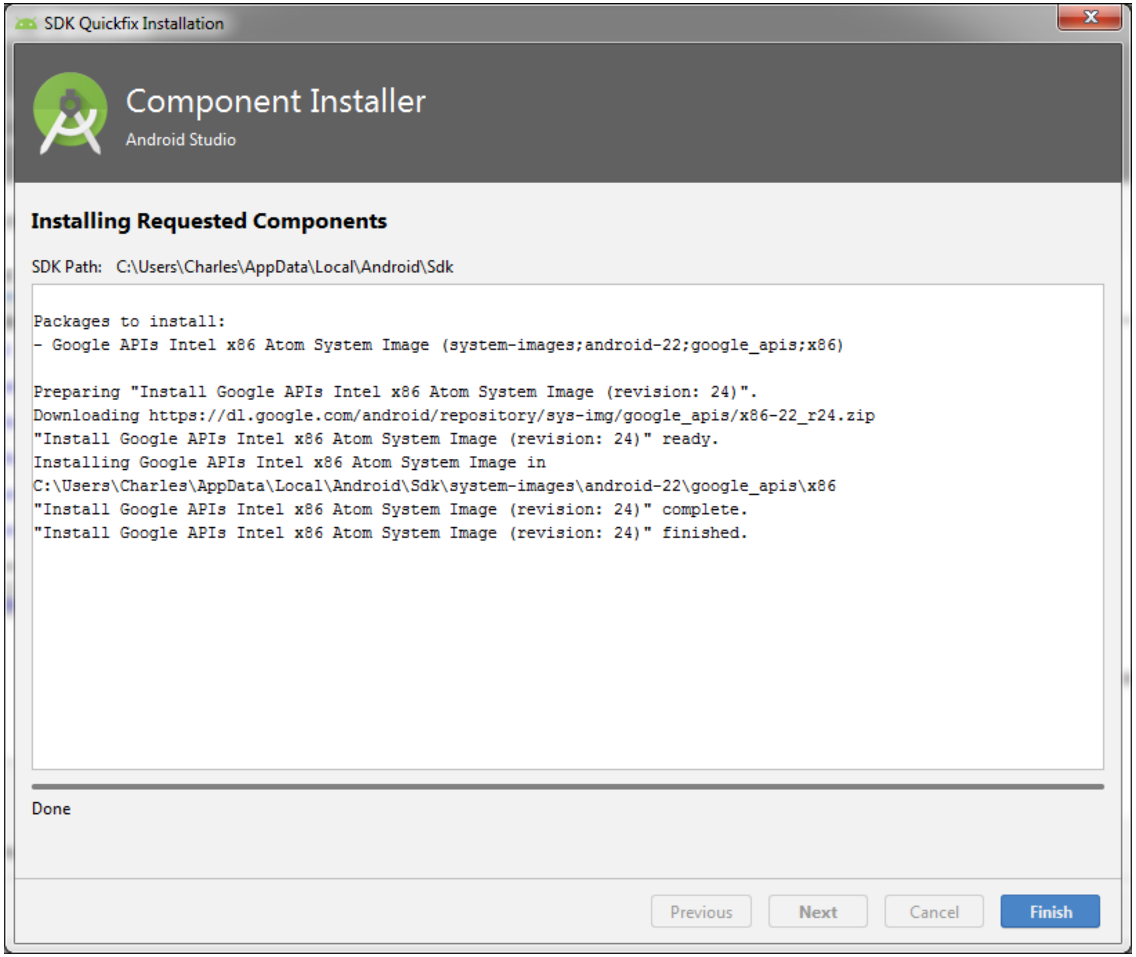
```
Packages to install:
- Google APIs Intel x86 Atom System Image (system-images;android-22;google_apis;x86)

Preparing "Install Google APIs Intel x86 Atom System Image (revision: 24)".
Downloading https://dl.google.com/android/repository/sys-img/google_apis/x86-22_r24.zip
```

Downloading (27%): 105.5 / 390.9 MB ...

https://dl.google.com/android/repository/sys-img/google_apis/x86-22_r24.zip

Please wait until the installation finishes

# Done



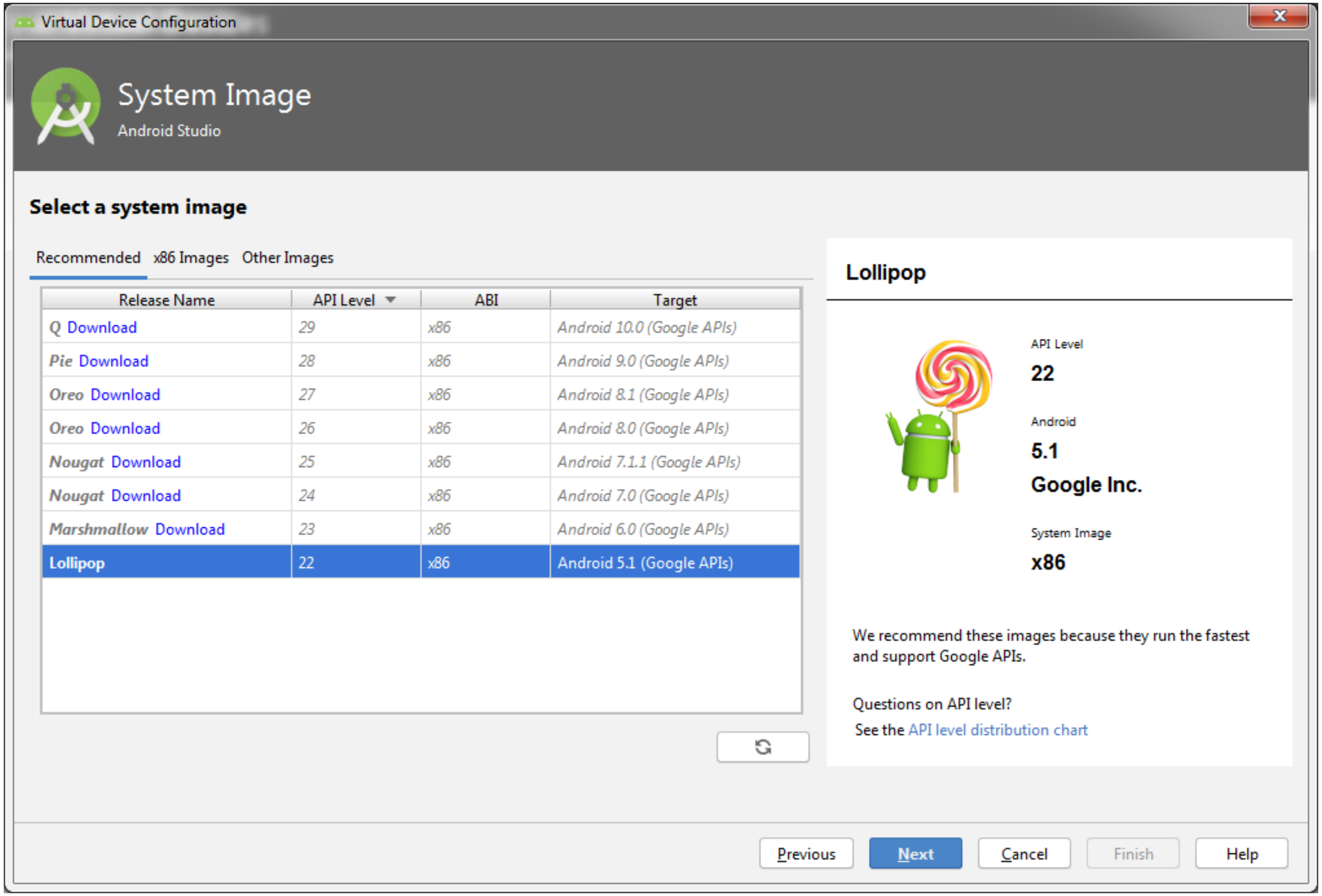
**Installing Requested Components**

SDK Path: C:\Users\Charles\AppData\Local\Android\Sdk

```
Packages to install:
- Google APIs Intel x86 Atom System Image (system-images;android-22;google_apis;x86)

Preparing "Install Google APIs Intel x86 Atom System Image (revision: 24)".
Downloading https://dl.google.com/android/repository/sys-img/google_apis/x86-22_r24.zip
"Install Google APIs Intel x86 Atom System Image (revision: 24)" ready.
Installing Google APIs Intel x86 Atom System Image in
C:\Users\Charles\AppData\Local\Android\Sdk\system-images\android-22\google_apis\x86
"Install Google APIs Intel x86 Atom System Image (revision: 24)" complete.
"Install Google APIs Intel x86 Atom System Image (revision: 24)" finished.
```

Done

# Note it no longer says "download"



Virtual Device Configuration

System Image
Android Studio

**Select a system image**

Recommended | x86 Images | Other Images

| 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

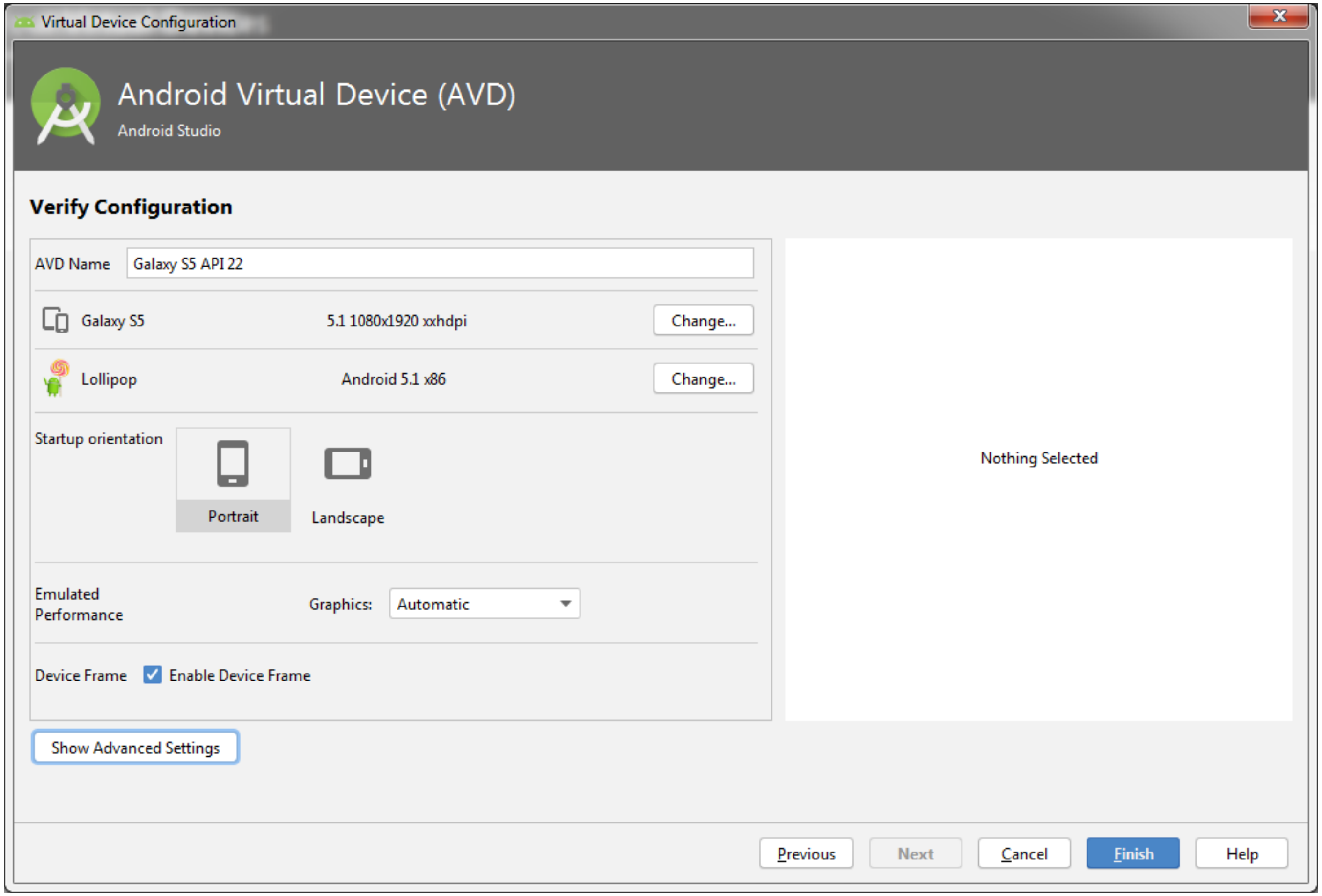
Android
5.1
Google Inc.

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

# Confirm



Virtual Device Configuration

Android Virtual Device (AVD)

Android Studio

**Verify Configuration**

AVD Name: Galaxy S5 API 22

Galaxy S5 — 5.1 1080x1920 xxhdpi

Lollipop — Android 5.1 x86

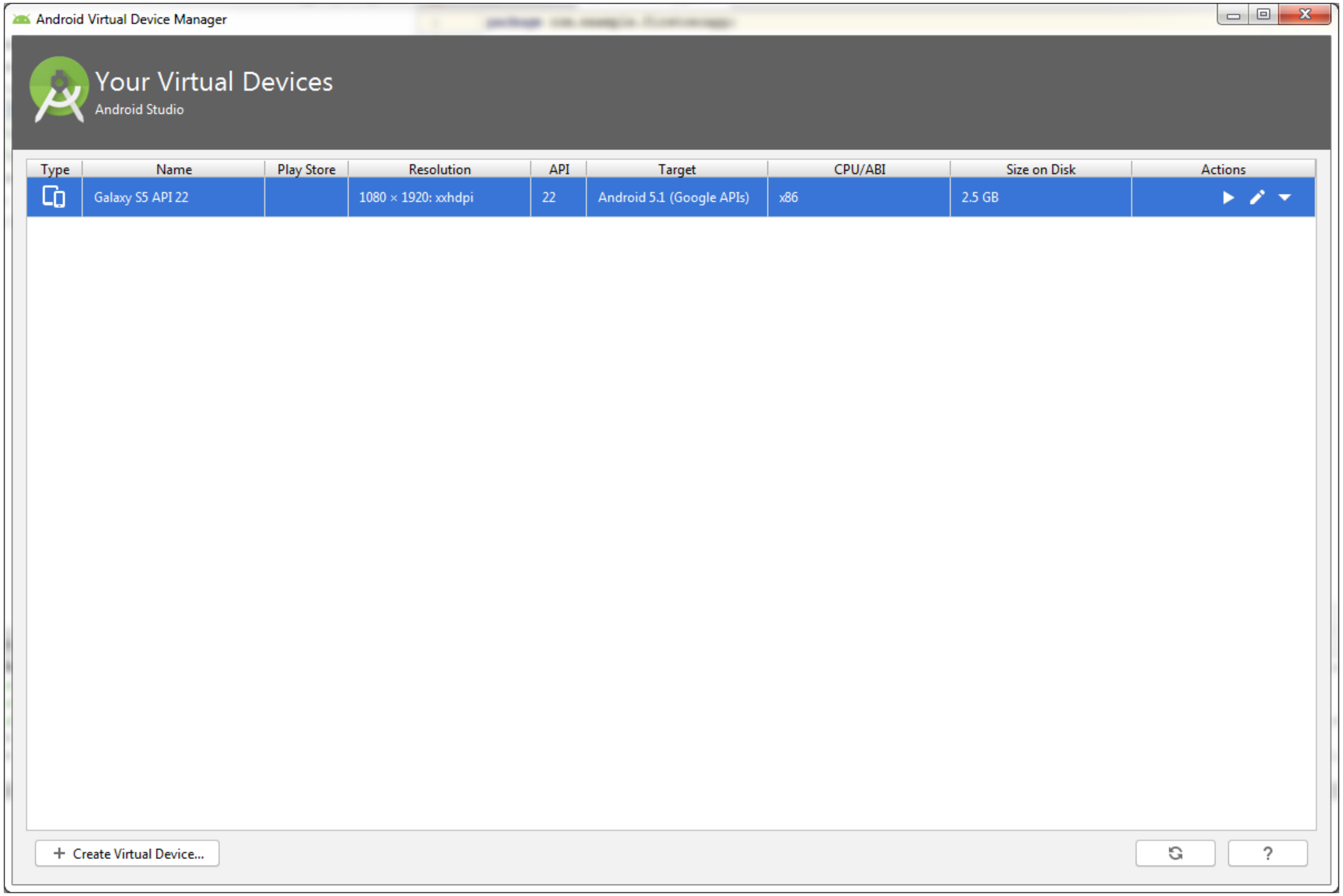
Startup orientation: Portrait

Emulated Performance — Graphics: Automatic

Device Frame: ☑ Enable Device Frame

Nothing Selected

# Now We have a virtual S5



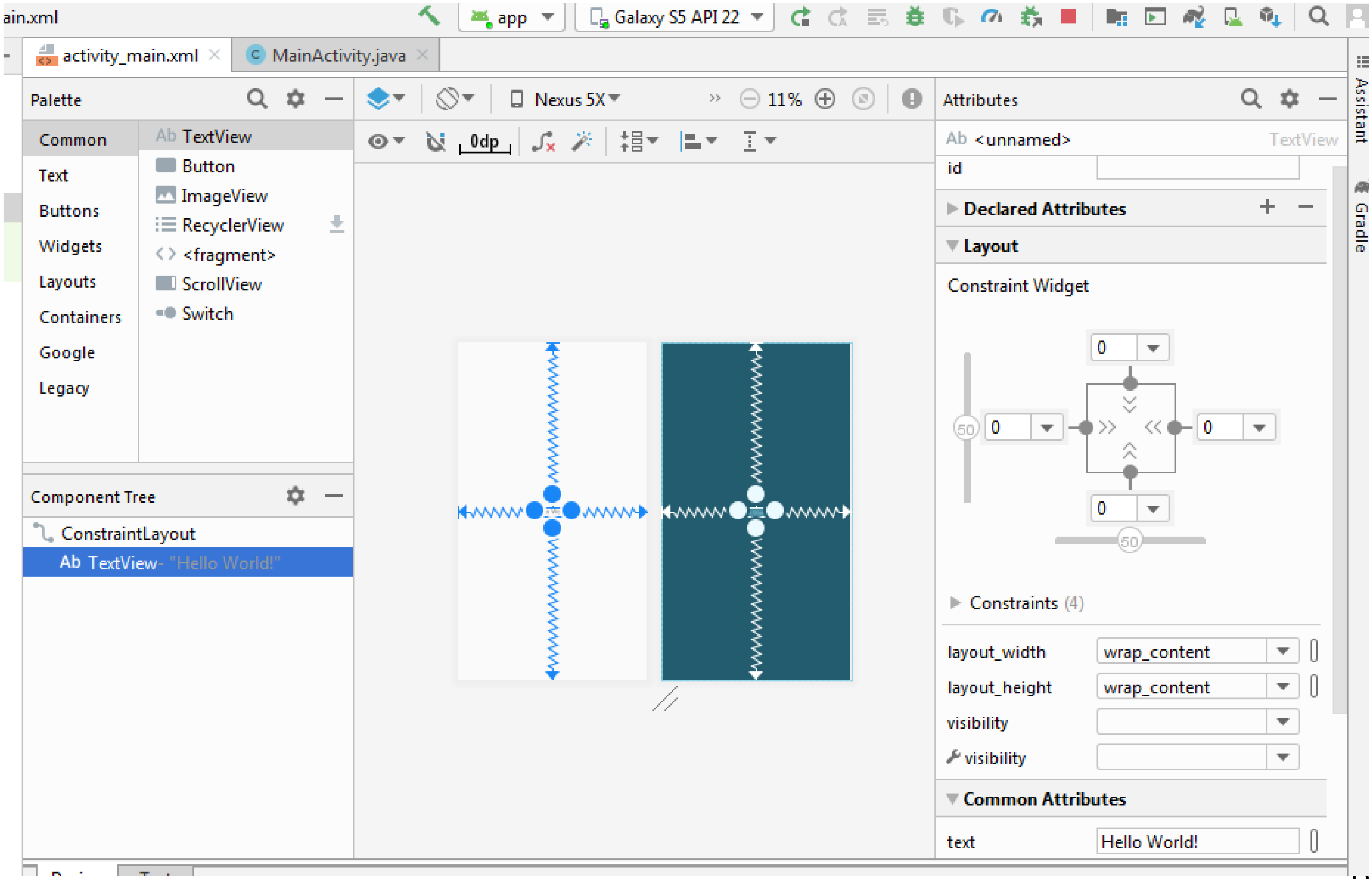
Android Virtual Device Manager

Your Virtual Devices

Android Studio

| Type | Name | Play Store | Resolution | API | Target | CPU/ABI | Size on Disk | Actions |
|---|---|---|---|---|---|---|---|---|
| | Galaxy S5 API 22 | | 1080 × 1920: xxhdpi | 22 | Android 5.1 (Google APIs) | x86 | 2.5 GB | ▶ ✎ ▾ |

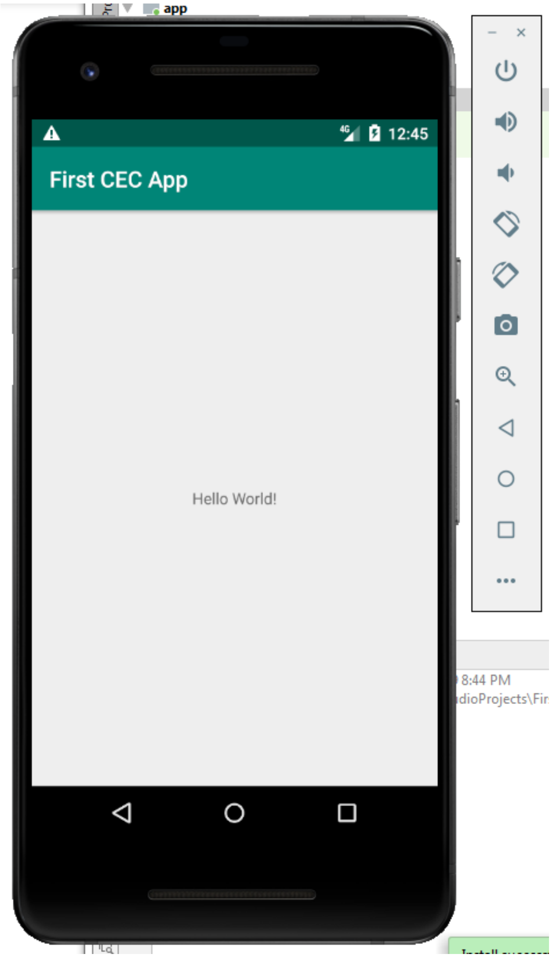
# We can select our AVD and run



| | |
|---|---|
| layout_width | wrap_content |
| layout_height | wrap_content |
| visibility | |
| visibility | |
| text | Hello World! |

Presented by:

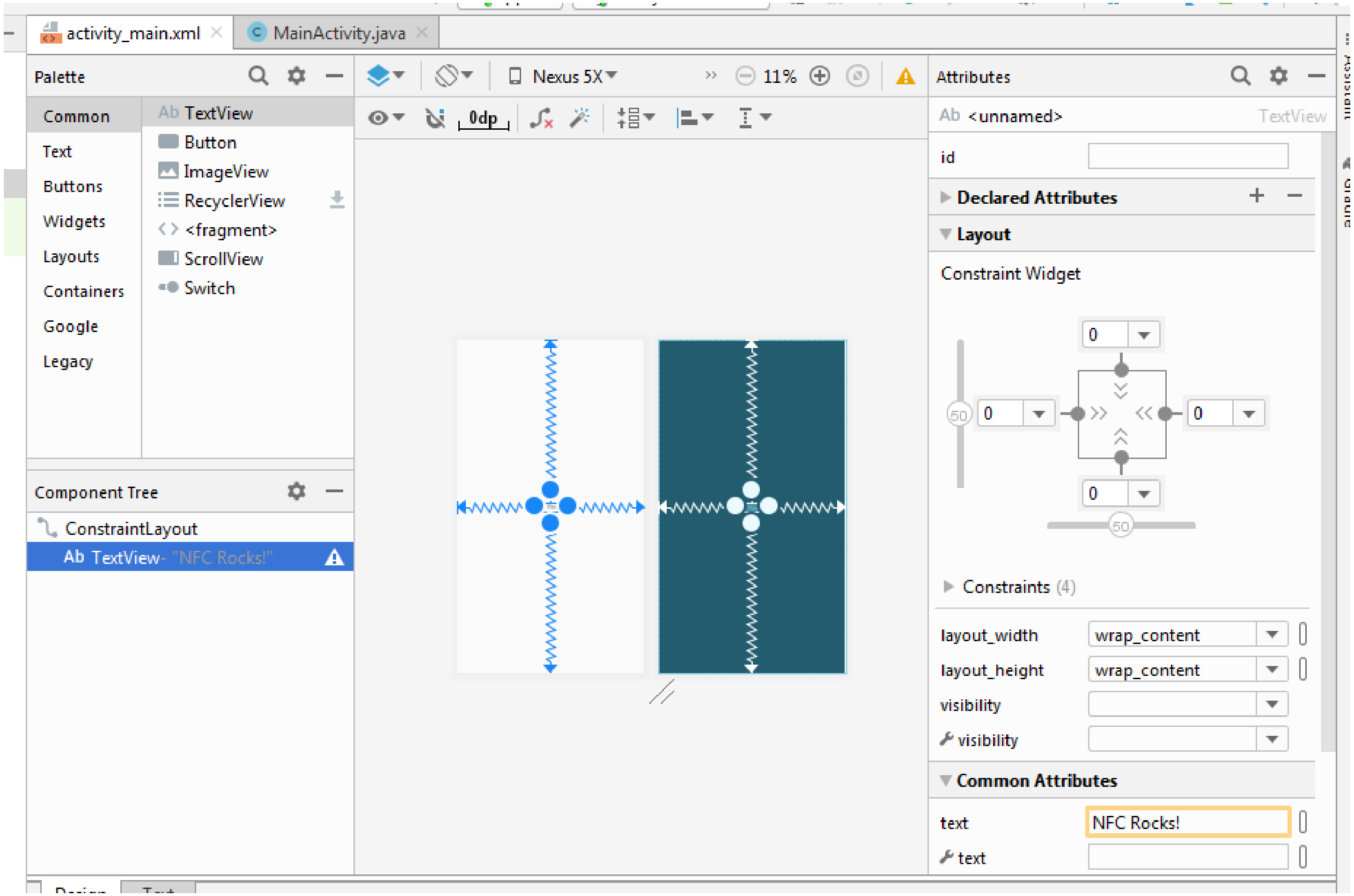
# Hello World in AVD



First CEC App

Hello World!

# Change Text – Warning!



id:

layout_width: wrap_content

layout_height: wrap_content

visibility:

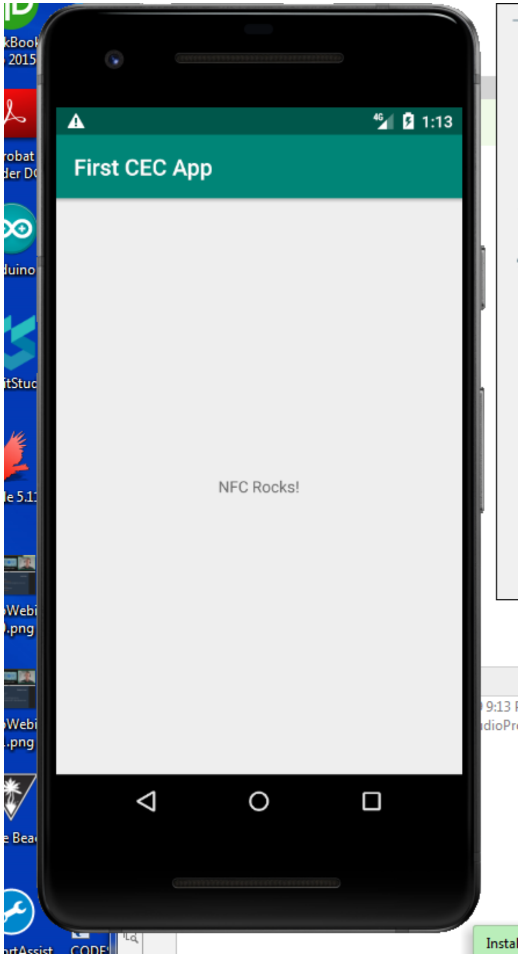
visibility:

text: NFC Rocks!

text:

Presented by:

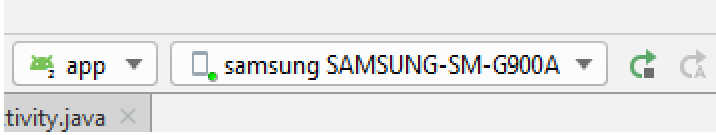
# But it works



Presented by:

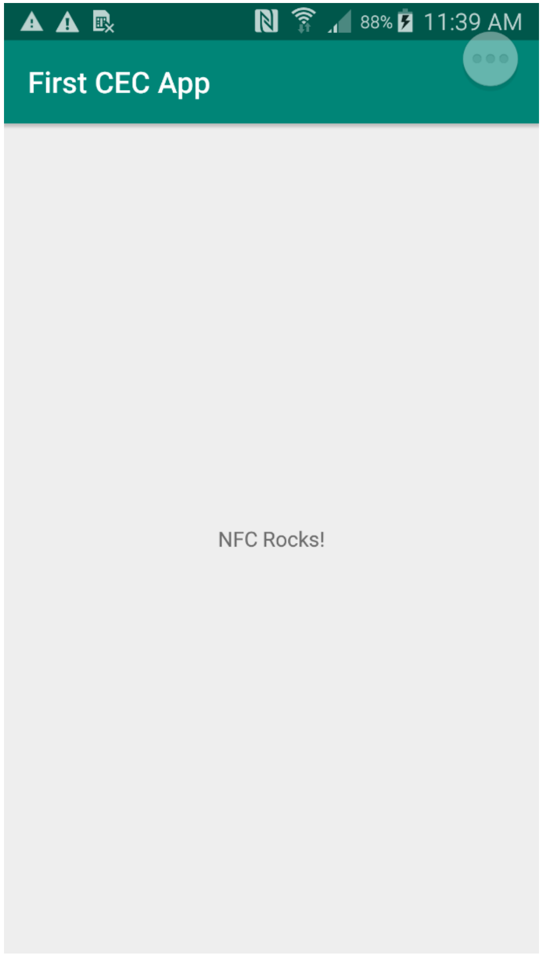
# Run on a real device

- Set up your device as follows:
- Connect your device to your development machine with a USB cable. If you developed on Windows, you might need to install the appropriate USB driver for your device.
- Perform the following steps to enable USB debugging in the Developer options window:
  - Open the Settings app.
  - If your device uses Android v8.0 or higher, select System. Otherwise, proceed to the next step.
  - Scroll to the bottom and select About phone.
  - Scroll to the bottom and tap Build number seven times.
  - Return to the previous screen, scroll to the bottom, and tap Developer options.
  - In the Developer options window, scroll down to find and enable USB debugging.
- Unplug then reconnect the USB – approve the computer's GUID
- Choose the device instead of the virtual device
- Run (Play protect will pop up)


app

samsung SAMSUNG-SM-G900A

# Screenshot from phone

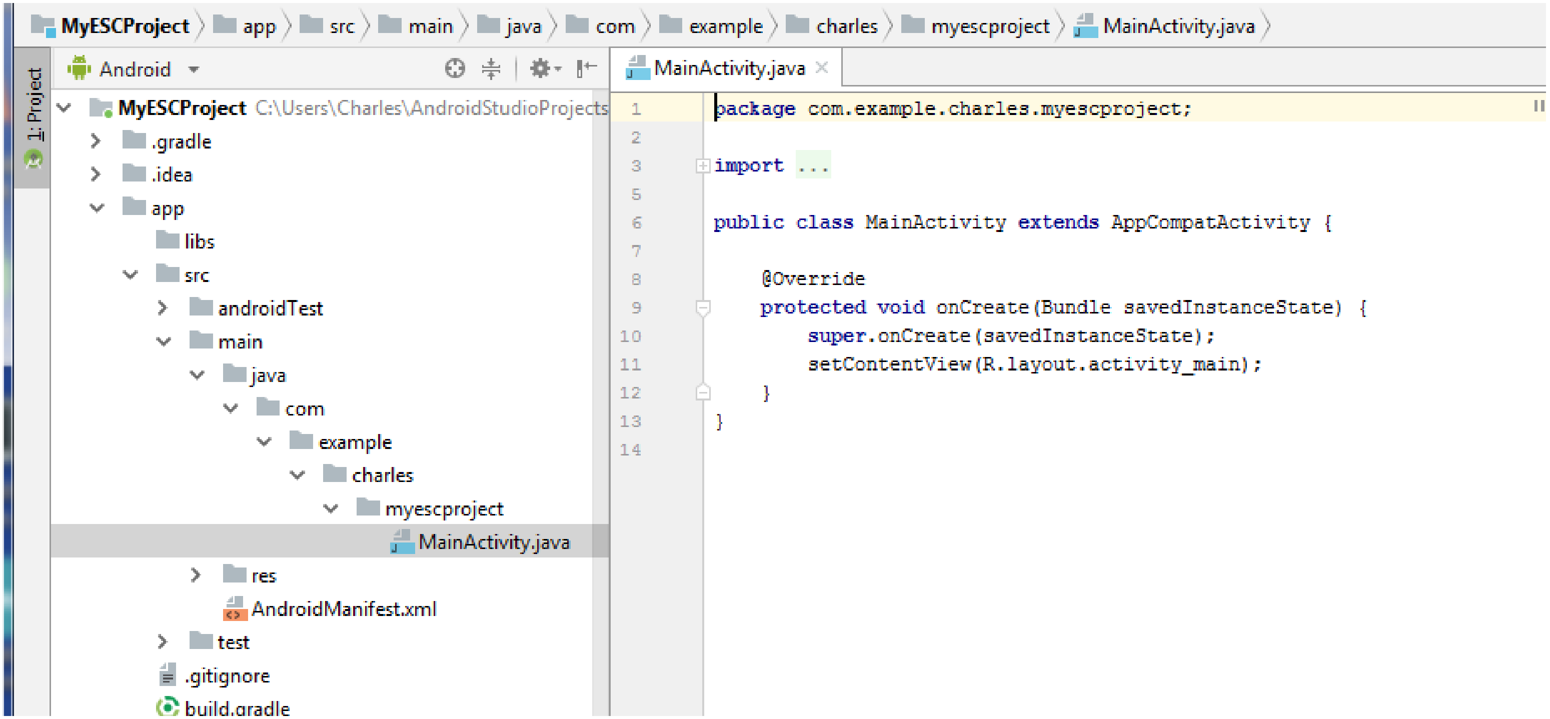


88% 11:39 AM

First CEC App

NFC Rocks!

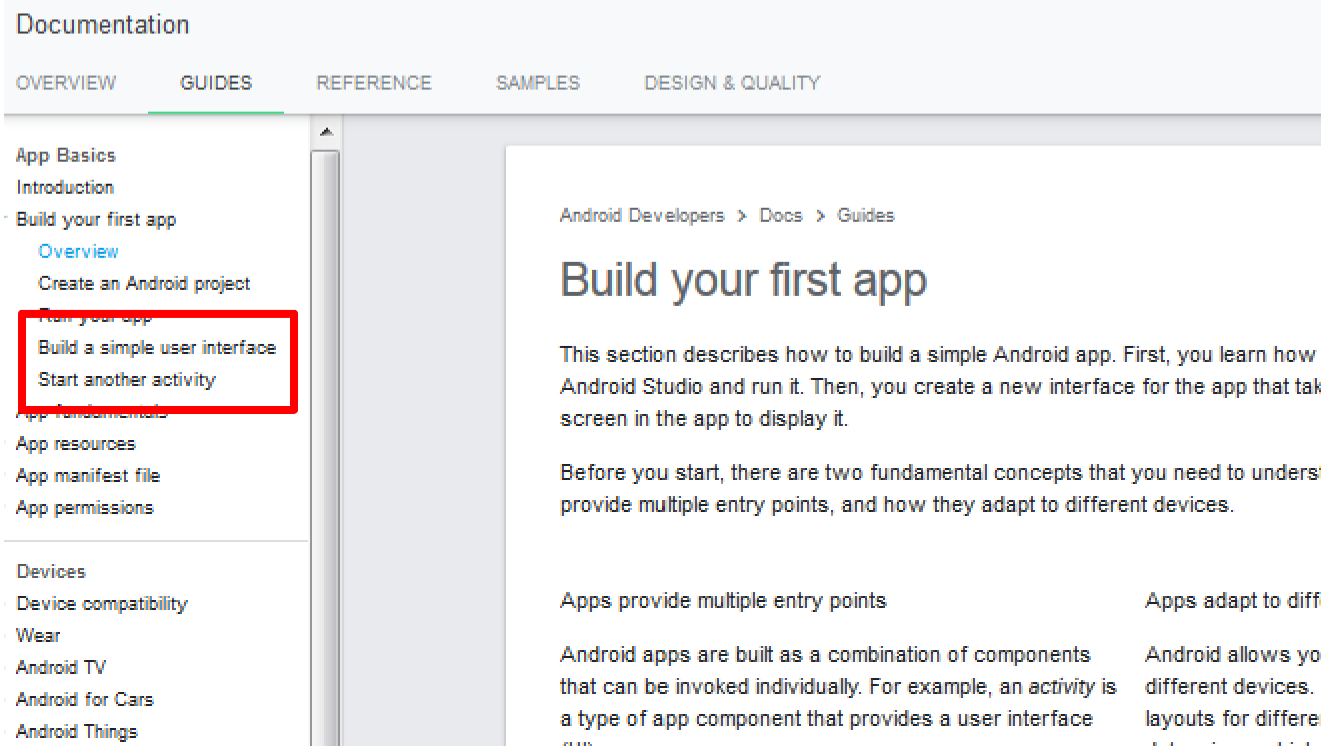
# MainActivity is the typical entry



```java
package com.example.charles.myescproject;

import ...

public class MainActivity extends AppCompatActivity {

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

```

# Homework

https://developer.android.com/training/basics/firstapp/



Documentation

OVERVIEW GUIDES REFERENCE SAMPLES DESIGN & QUALITY

App Basics
Introduction
Build your first app
Overview
Create an Android project
Run your app
Build a simple user interface
Start another activity
App fundamentals
App resources
App manifest file
App permissions

Devices
Device compatibility
Wear
Android TV
Android for Cars
Android Things

Android Developers > Docs > Guides

## Build your first app

This section describes how to build a simple Android app. First, you learn how
Android Studio and run it. Then, you create a new interface for the app that tak
screen in the app to display it.

Before you start, there are two fundamental concepts that you need to unders
provide multiple entry points, and how they adapt to different devices.

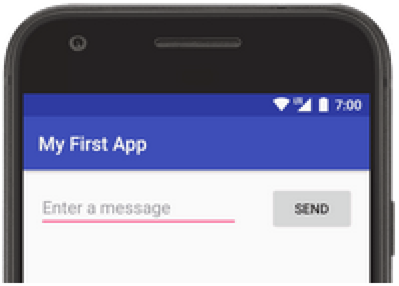
Apps provide multiple entry points

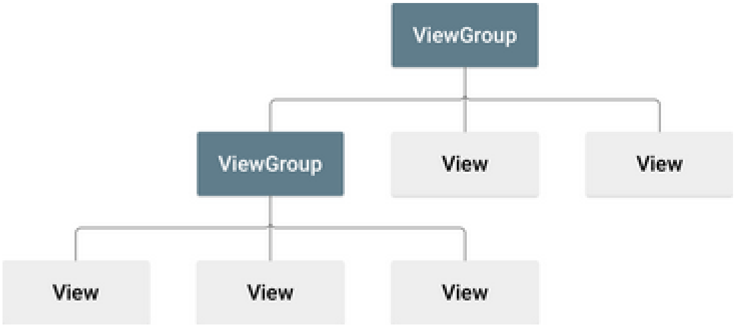
Android apps are built as a combination of components that can be invoked individually. For example, an *activity* is a type of app component that provides a user interface

Apps adapt to diff

Android allows yo
different devices.
layouts for differe

# Preview of Homework





ViewGroup

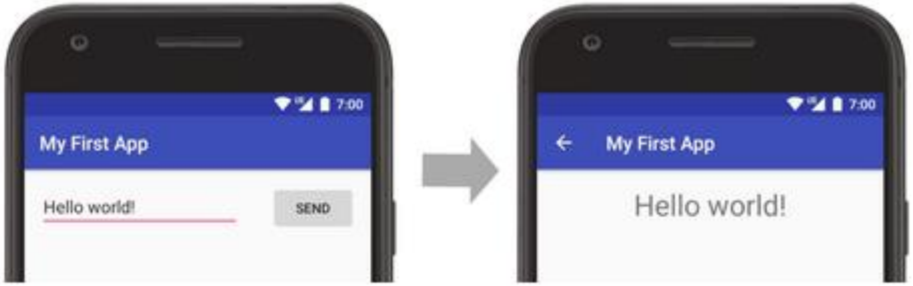
ViewGroup View View

View View View

- You will learn to build a hierarchy of viewgroups
- Using the layout editor, you will build the XML tables to define these views without writing a line of code
- You will also learn about string resources (remember our warning?)

Question 3: Why would we not want to use string literals?

# In Part 2 (“Start another activity):



- You will learn about ‘intents’ – ways of sharing data between activities
- You will build a new activity for the “SEND” button and a new view for the resultant screen.
- We will review this tomorrow and then add NFC!

# This Week's Agenda

9/23 Introduction to the Project and Development Environment

9/24 An NFC Primer and Introducing the NXP NTAG

9/25 Building an Android Application from Scratch

9/26 Adding NFC Capability and Communications to Our App

9/27 Putting it All Together

# Please stick around as I answer your questions!

- Please give me a moment to scroll back through the chat window to find your questions
- I will stay on chat as long as it takes to answer!
- I am available to answer simple questions or to consult (or offer in-house training for your company)
  c.j.lord@ieee.org
  http://www.blueridgetechnc.com
  http://www.linkedin.com/in/charleslord
  Twitter: @charleslord
  https://www.github.com/bradatraining