

Pemrograman Mobile

3 SKS | Semester 7 | S1 Sistem Informasi

Nizar Rabbi Radliya
nizar@email.unikom.ac.id

Lesson 1

Introduction to Android

Create Your First Android App



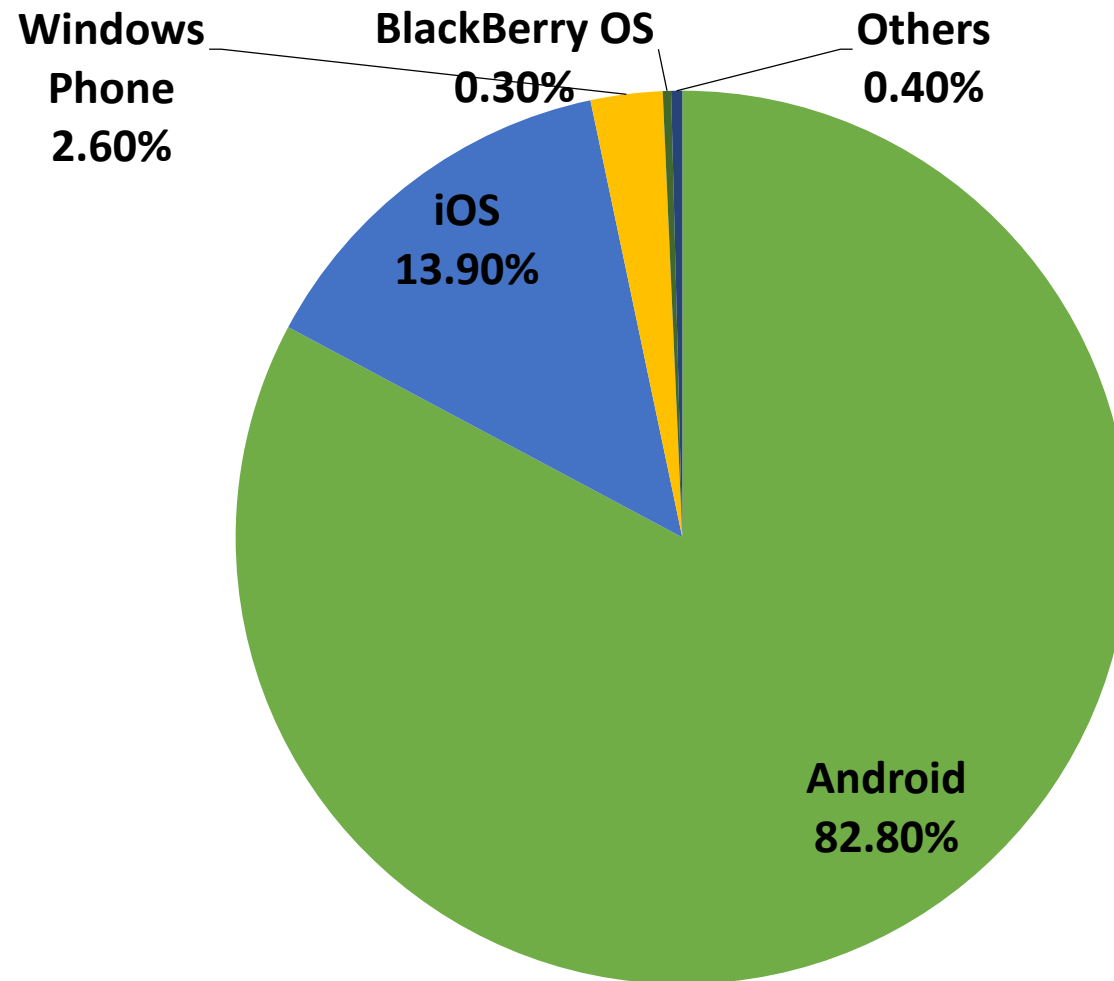


Mobile Device



Mobile Platform/OS

Mobile OS Market Share, 2015 Q2



Source: IDC, Aug 2015

Dasar Pemilihan Android

- ✓ Pangsa pasar lebih unggul
- ✓ Adanya *Google Play*
- ✓ Lengkap (*Complete Platform*)
- ✓ Terbuka (*Open Source Platform*)
- ✓ Gratis (*Free Platform*)
- ✓ Tersedianya dokumentasi dan komunitas (forum)
- ✓ Masih minimnya implementasi SI berbasis Android



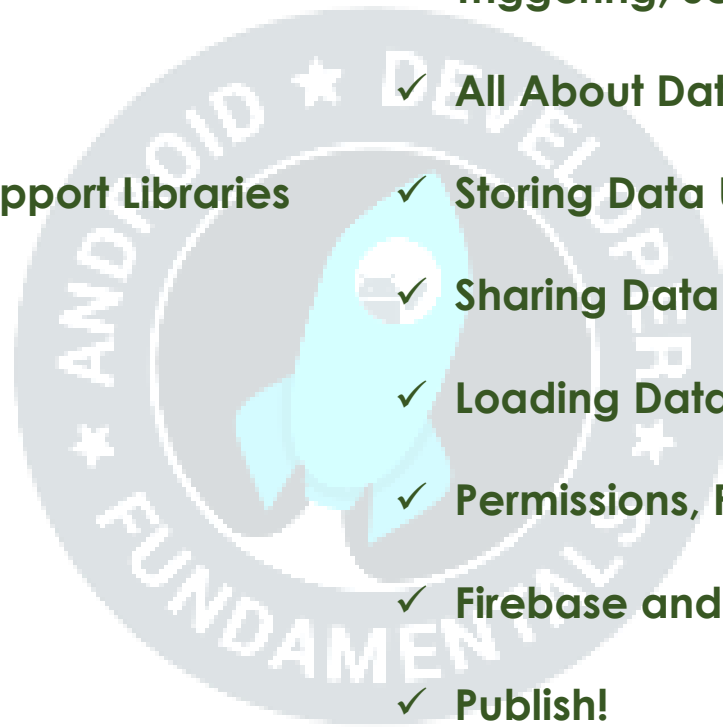
Mata kuliah “Pemrograman Mobile” (Prodi SI)

- ✓ Pembangunan aplikasi pada perangkat mobile
- ✓ Menggunakan platform android
- ✓ Dimulai dari persiapan perlengkapan pembangunan
- ✓ Sampai pada tahap file distribusi dan pendistribusian aplikasi



Silabus “Pemrograman Mobile” (Prodi SI)

- ✓ Build Your First App
- ✓ Activities
- ✓ Testing, Debugging, and Using Support Libraries
- ✓ User Interaction
- ✓ Delightful User Experience
- ✓ Testing your UI
- ✓ Background Tasks
- ✓ Triggering, Scheduling, and Optimizing Background Tasks
- ✓ All About Data
- ✓ Storing Data Using SQLite
- ✓ Sharing Data with Content Providers
- ✓ Loading Data Using Loaders
- ✓ Permissions, Performance and Security
- ✓ Firebase and AdMob
- ✓ Publish!



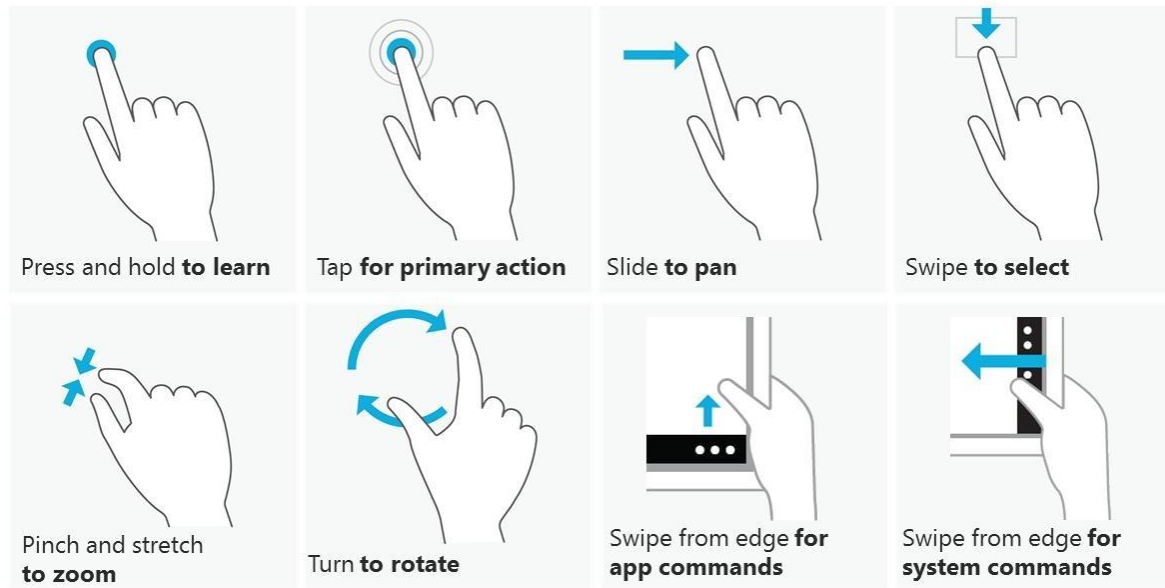
What is Android?

- Mobile operating system based on Linux kernel
- User Interface for touch screens
- Used on over 80% of all smartphones
- Powers devices such as watches, TVs, and cars
- Over 2 Million Android apps in Google Play store
- Highly customizable for devices / by vendors
- Open source



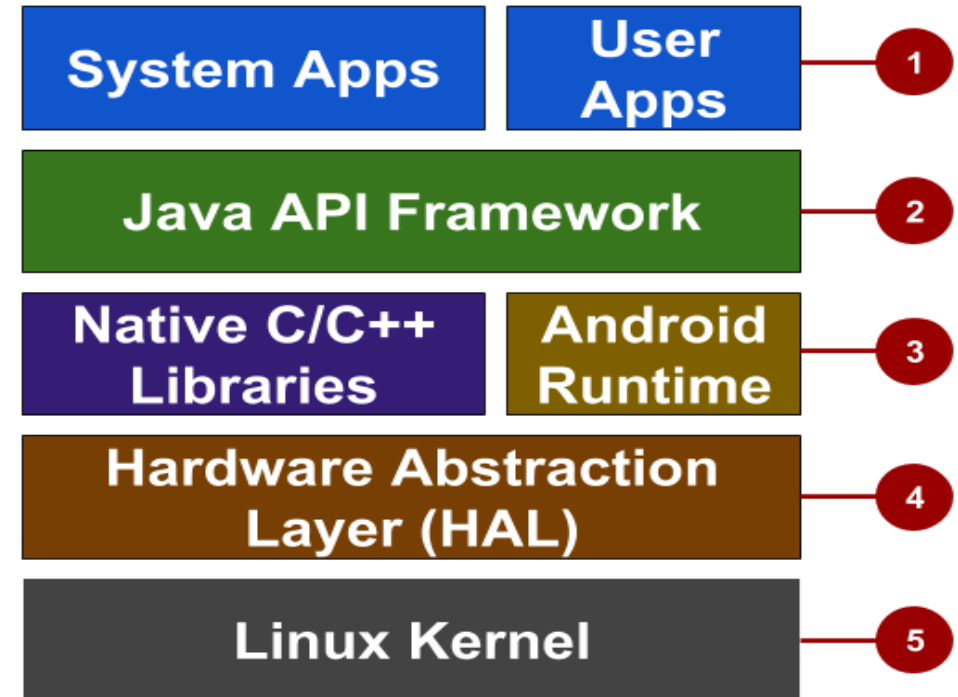
Android user interaction

- Touch gestures: swiping, tapping, pinching
- Virtual keyboard for characters, numbers, and emoji
- Support for Bluetooth, USB controllers and peripherals

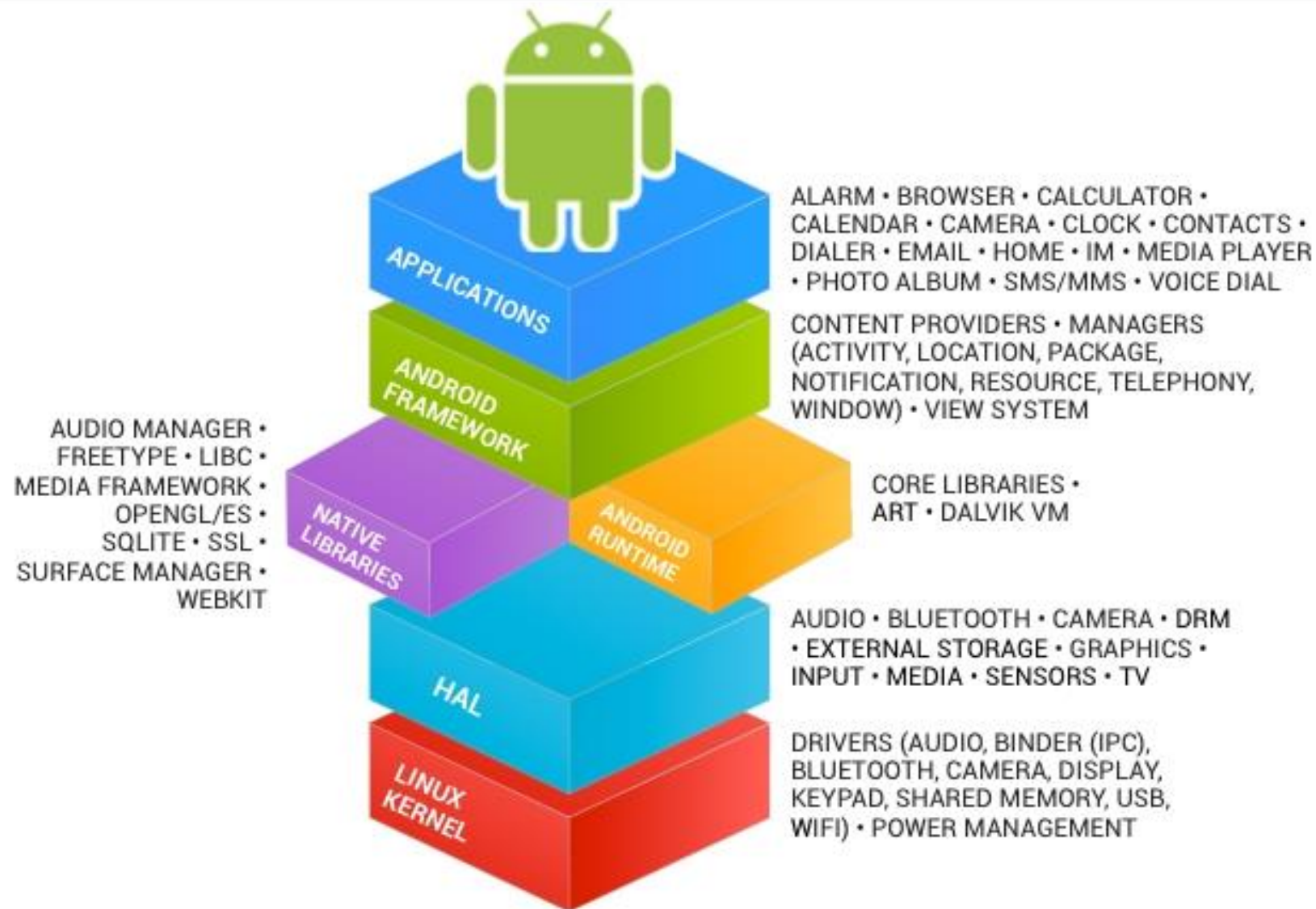


Arsitektur Android

1. System and user apps
2. Android OS API in Java framework
3. Expose native APIs; run apps
4. Expose device hardware capabilities
5. Linux Kernel



Arsitektur Android



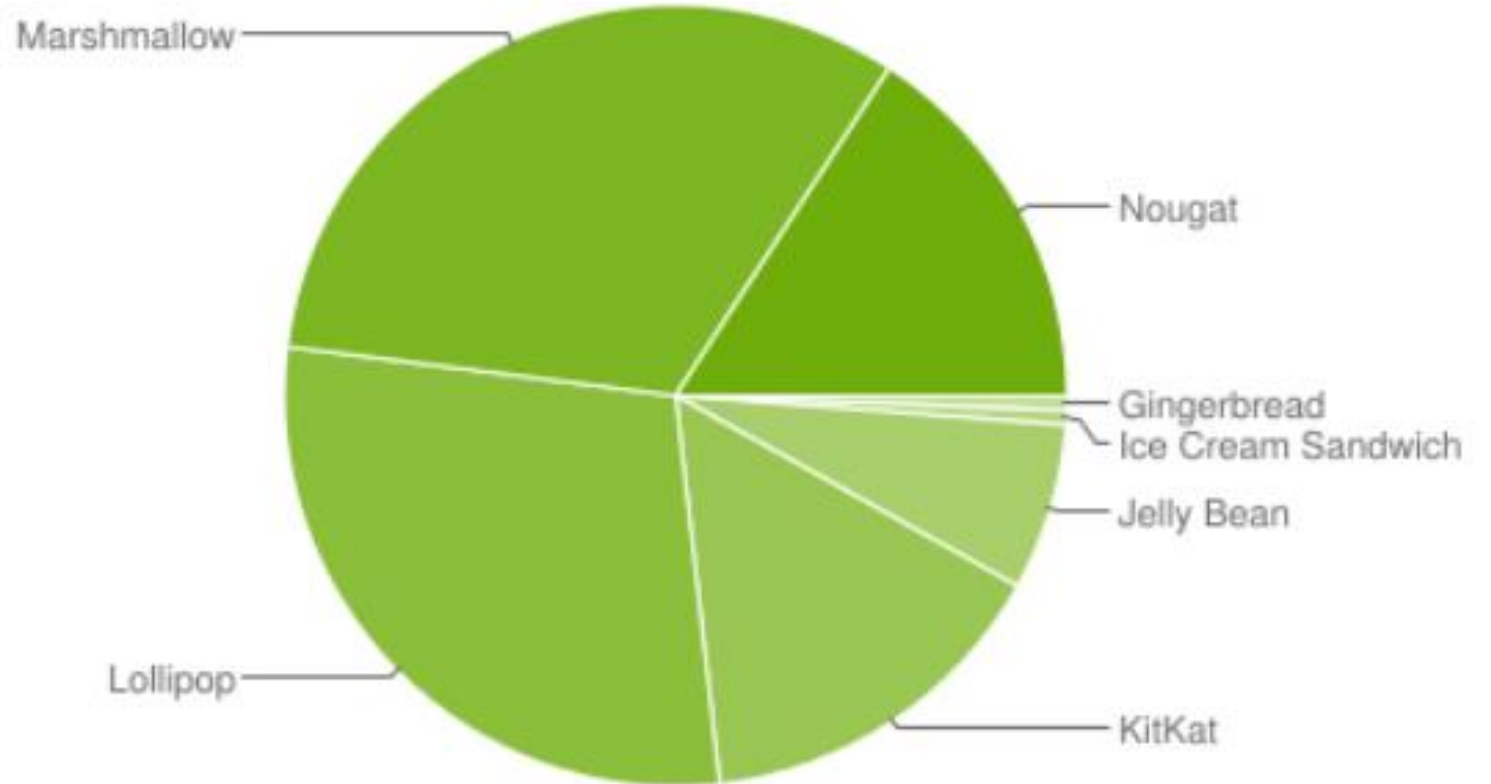
Platform Versions

Versi	Nama Kode
Android 1.0	
Android 1.1	
Android 1.5	Cupcake
Android 1.6	Donut
Android 2.0	Eclair
Android 2.0.1	Eclair
Android 2.1	Eclair
Android 2.2–2.2.3	Froyo
Android 2.3–2.3.2	Gingerbread
Android 2.3.3–2.3.7	Gingerbread
Android 3.0	Honeycomb
Android 3.1	Honeycomb
Android 3.2	Honeycomb
Android 4.0–4.0.2	Ice Cream Sandwich
Android 4.0.3–4.0.4	Ice Cream Sandwich
Android 4.1	Jelly Bean
Android 4.2	Jelly Bean
Android 4.3	Jelly Bean
Android 4.4	KitKat
Android 5.0	Lollipop
Android 5.1	Lollipop
Android 6.0	Marshmallow
Android 7.0	Nougat
Android 8.0	Oreo



Platform Versions

Version	Codename	API	Distribution
2.3.3 - 2.3.7	Gingerbread	10	0.6%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	0.6%
4.1.x	Jelly Bean	16	2.4%
4.2.x		17	3.5%
4.3		18	1.0%
4.4	KitKat	19	15.1%
5.0	Lollipop	21	7.1%
5.1		22	21.7%
6.0	Marshmallow	23	32.2%
7.0	Nougat	24	14.2%
7.1		25	1.6%



Data collected during a 7-day period ending on September 11, 2017.

Any versions with less than 0.1% distribution are not shown.

What is an Android app?

- One or more interactive screens
- Written using Java Programming Language and XML
- Uses the Android Software Development Kit (SDK)
- Uses Android libraries and Android Application Framework
- Executed by Android Runtime Virtual machine (ART)



Challenges of Android development

- Multiple screen sizes and resolutions
- Performance: make your apps responsive and smooth
- Security: keep source code and user data safe
- Compatibility: run well on older platform versions
- Marketing: understand the market and your users



App building blocks

- Resources: layouts, images, strings, colors as XML and media files
- Components: activities, services, ..., and helper classes as Java code
- Manifest: information about app for the runtime
- Build configuration: APK versions in Gradle config files



Component types

- **Activity** is a single screen with a user interface
- **Service** performs long-running tasks in background
- **Content provider** manages shared set of data
- **Broadcast receiver** responds to system-wide announcements



Think of Android as a hotel

- Your app is the guest
- The Android System is the hotel manager
- Services are available when you request them (intents)
 - In the foreground (activities) such as registration
 - In the background (services) such as laundry
- Calls you when a package has arrived (broadcast receiver)
- Access the city's tour companies (content provider)

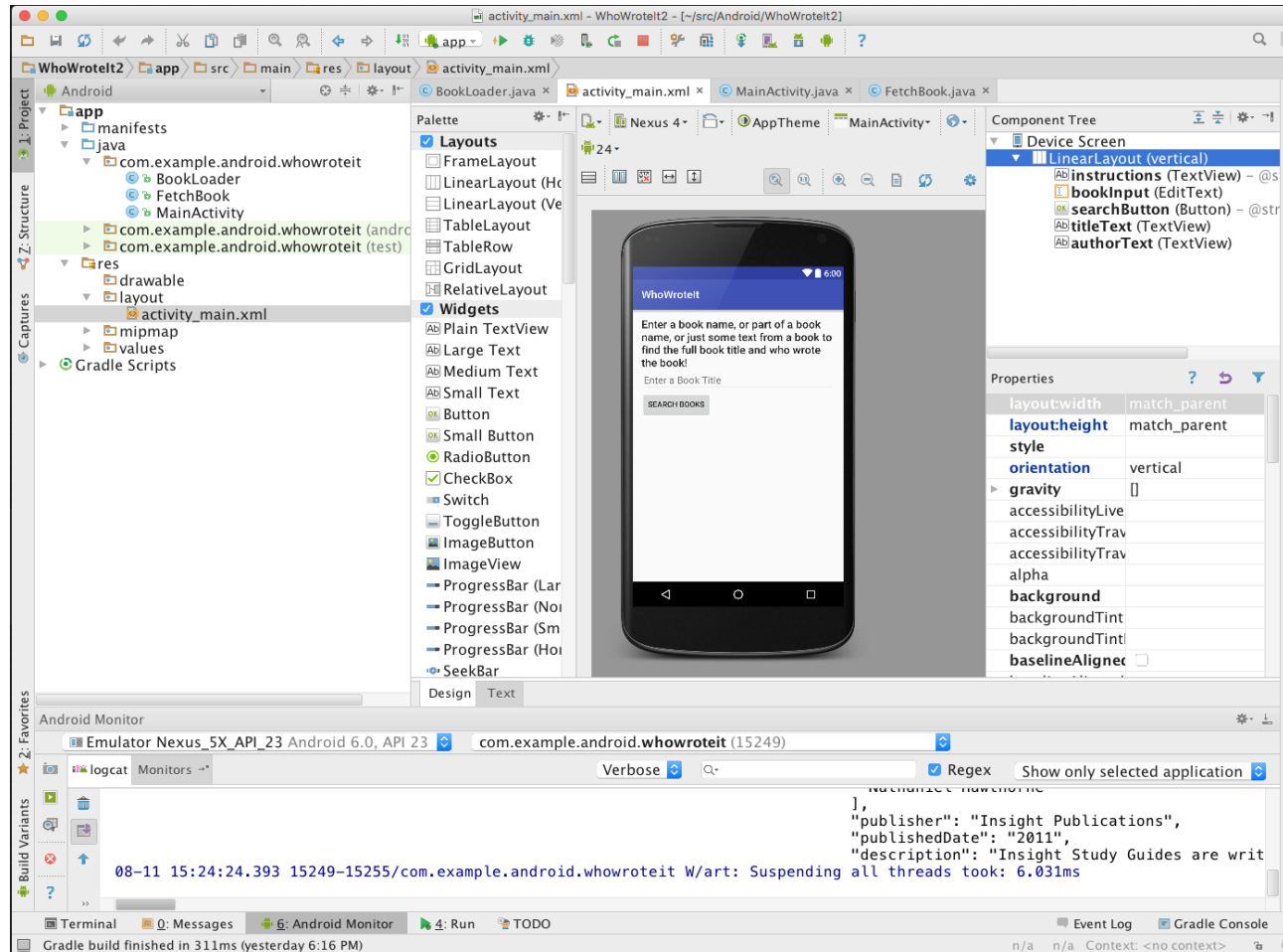


Installation Overview

- Java JDK
- Android SDK
- Download and install Android Studio from
<http://developer.android.com/sdk/index.html>
- IDE Eclipse
- Android ADT

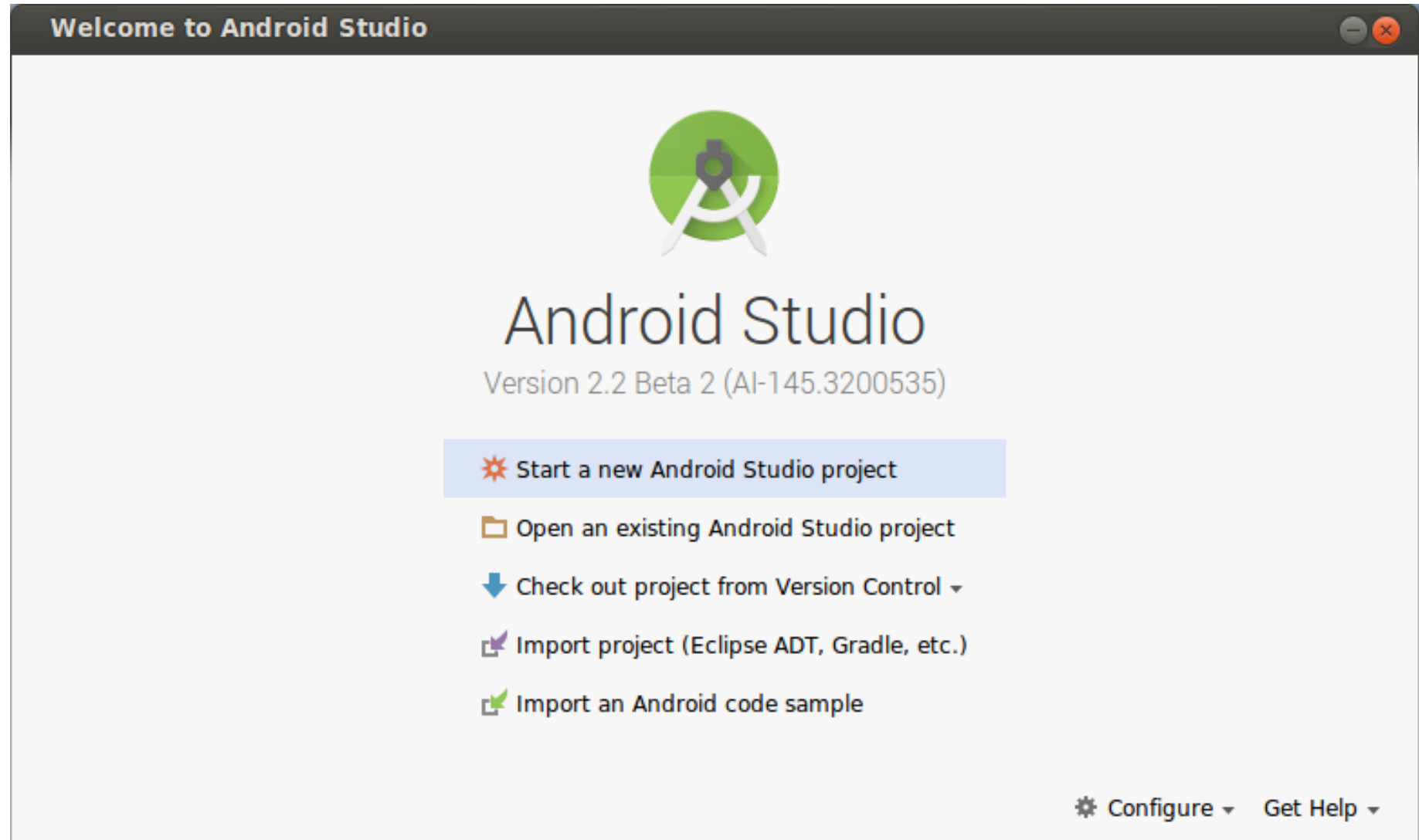


What is Android Studio?

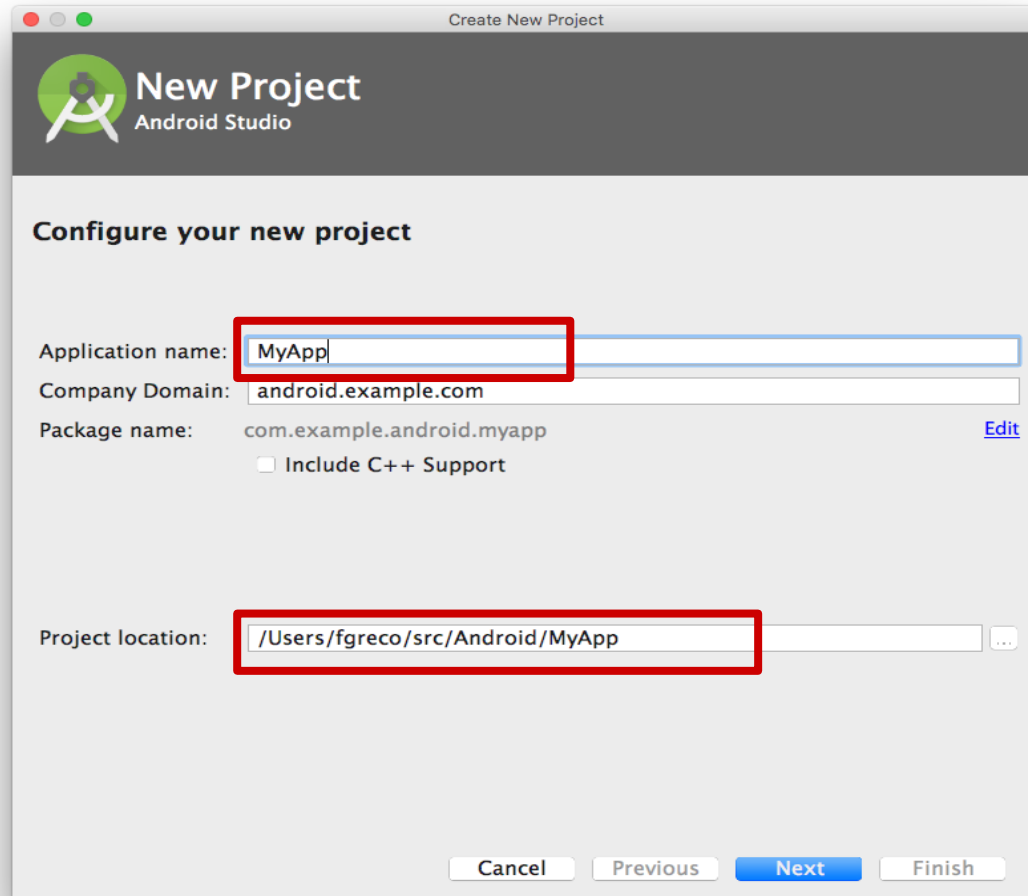


- Android IDE
- Project structure
- Templates
- Layout Editor
- Testing tools
- Gradle-based build
- Log Console
- Debugger
- Monitors
- Emulators

Creating Your First Android App



Name your app



Create New Project

New Project
Android Studio

Configure your new project

Application name:

Company Domain:

Package name: [Edit](#)

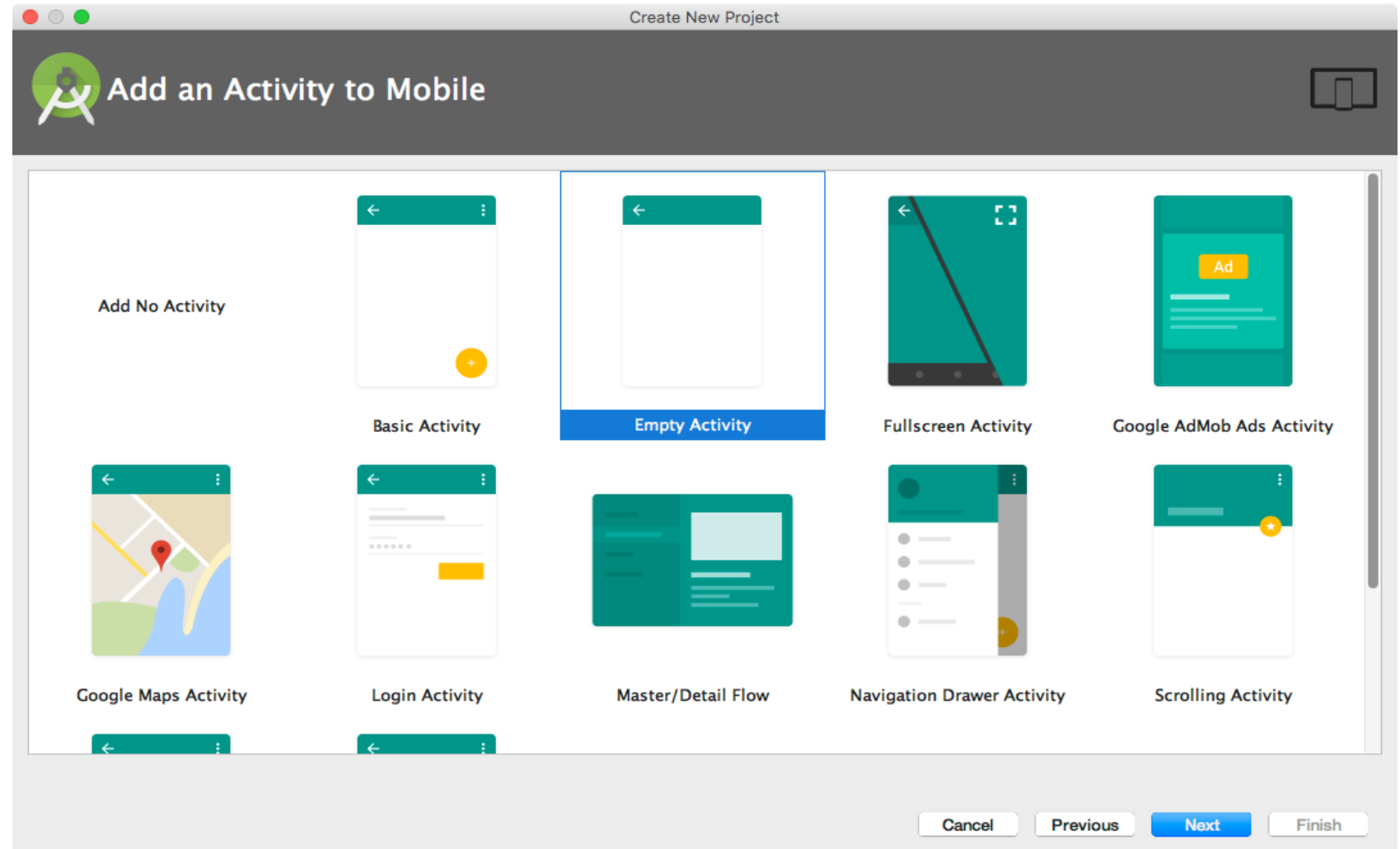
☐ Include C++ Support

Project location:

Pick activity template

Choose templates for common activities, such as maps or navigation drawers.

Pick Empty Activity or Basic Activity for simple and custom activities.

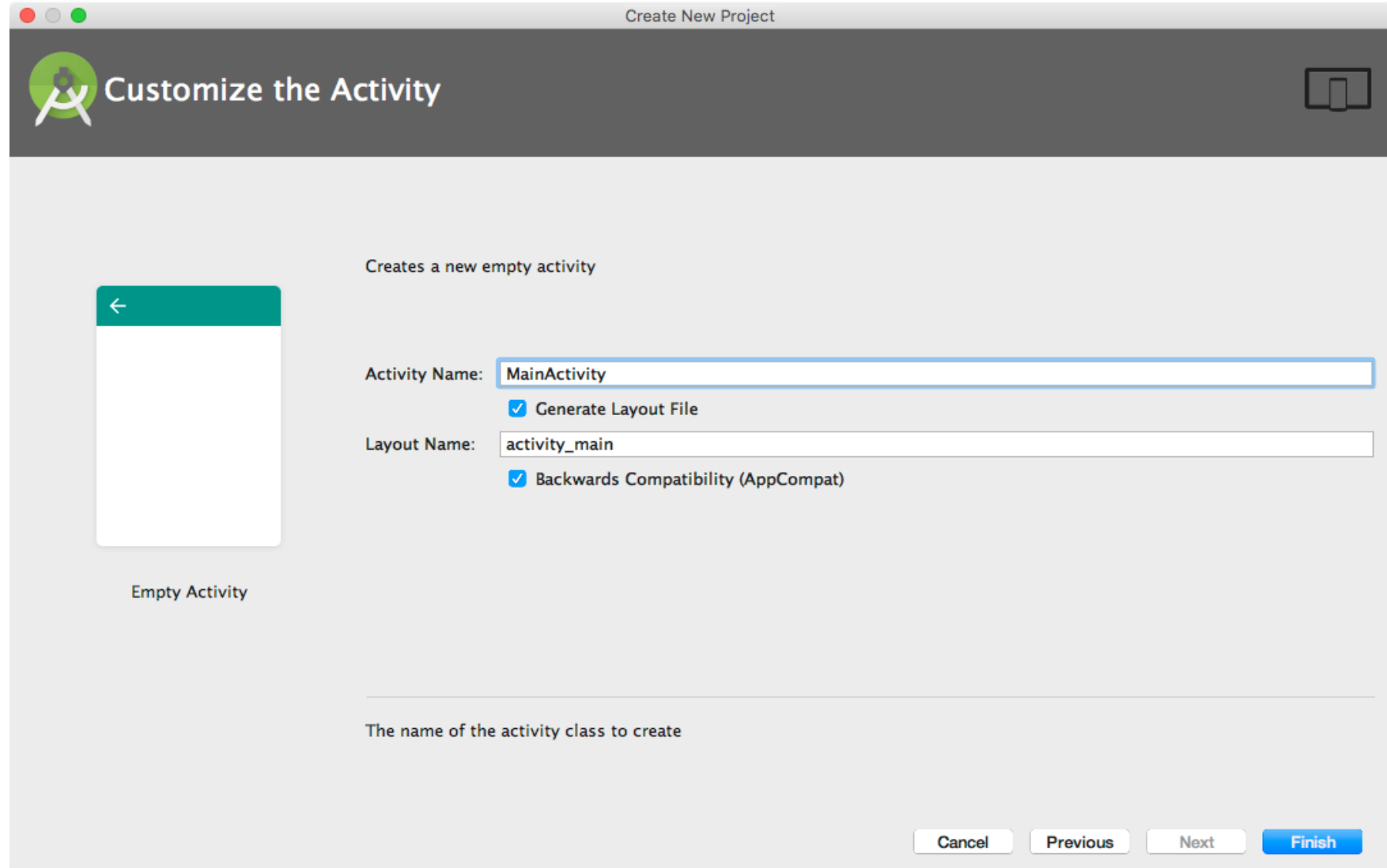


Name your activity

Good practice to name main activity MainActivity and activity_main layout

Use AppCompatActivity

Generating layout file is convenient



Create New Project

Customize the Activity

Creates a new empty activity

Activity Name: MainActivity

☒ Generate Layout File

Layout Name: activity_main

☒ Backwards Compatibility (AppCompat)

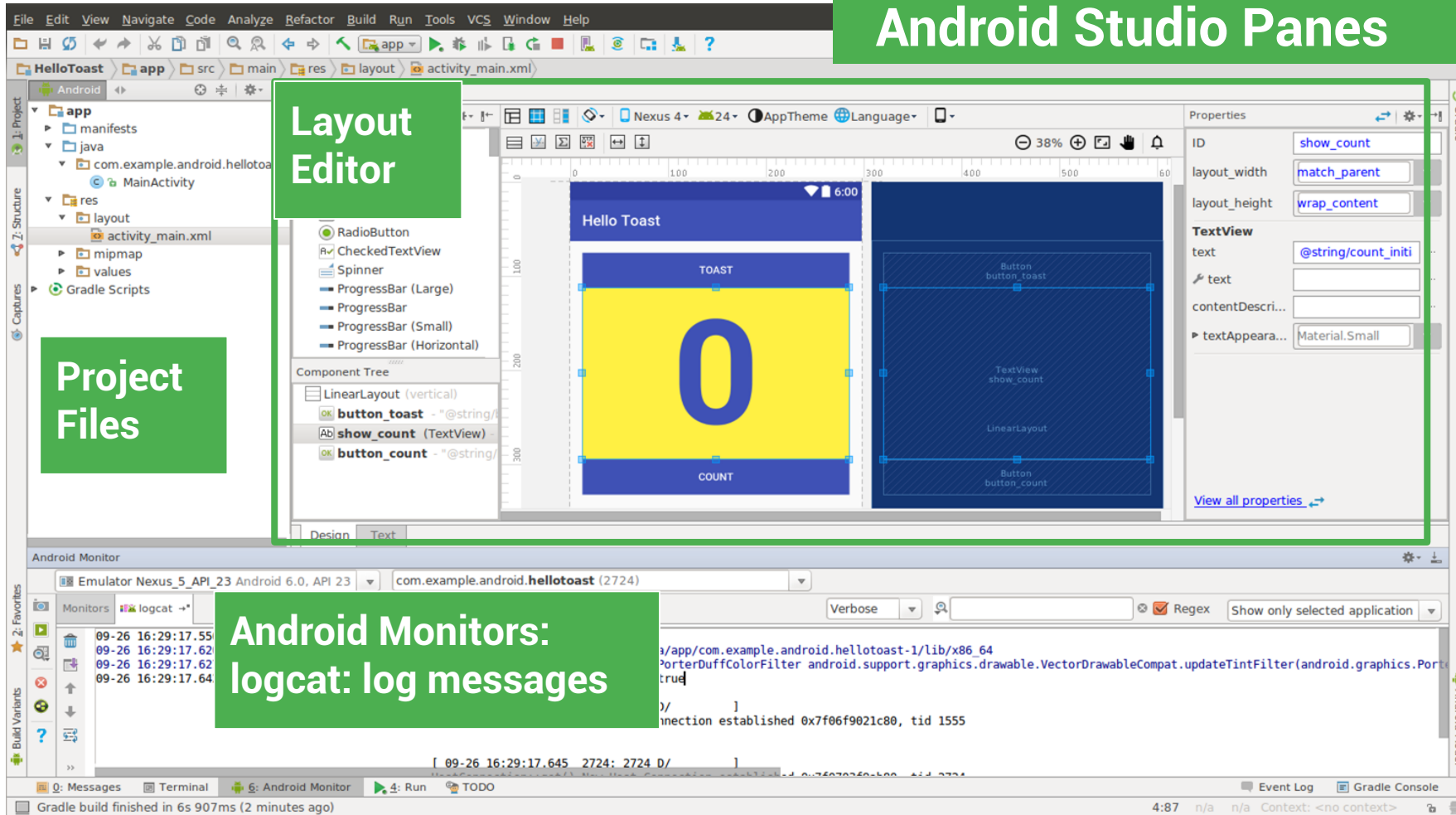
Empty Activity

The name of the activity class to create

Cancel Previous Next Finish

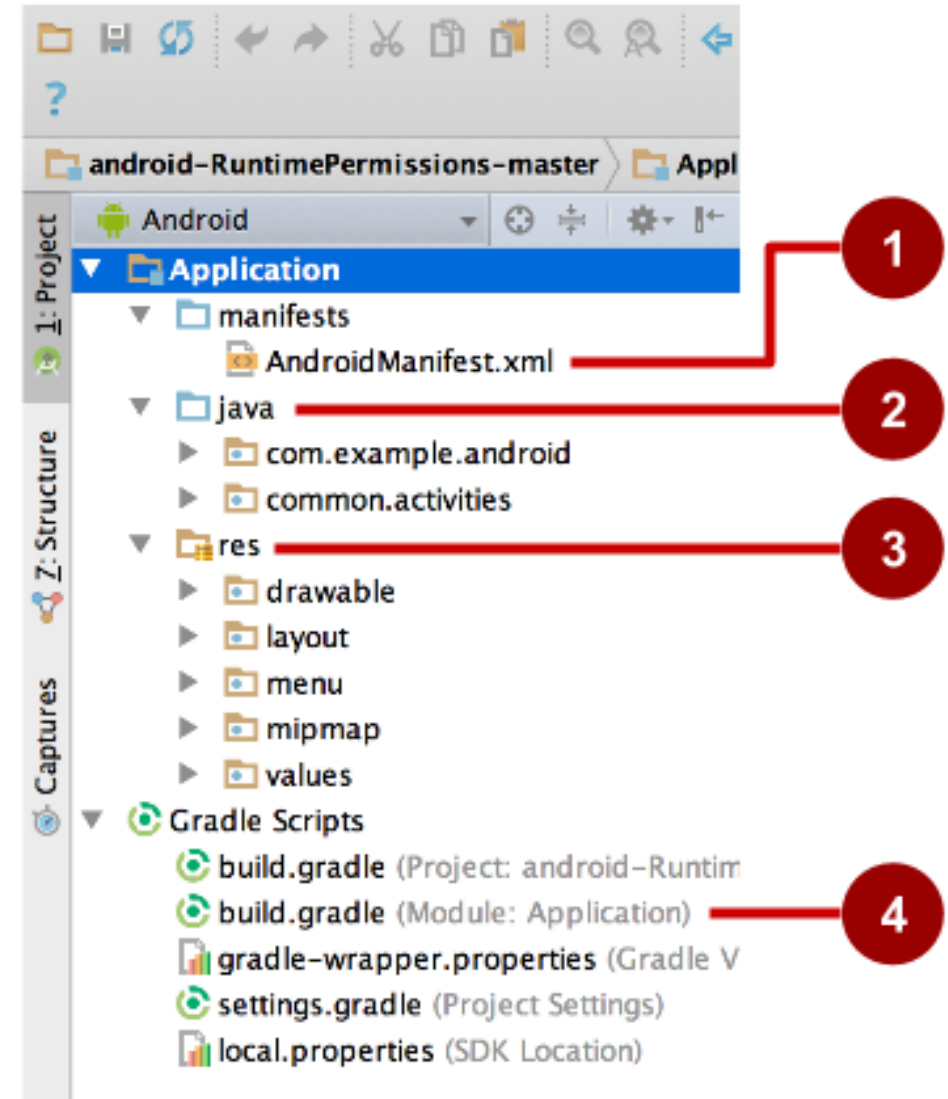
Android Studio

Android Studio Panes



Project folders

1. manifests—Android Manifest file - description of app read by the Android runtime
2. java—Java source code packages
3. res—Resources (XML) - layout, strings, images, dimensions, colors...
4. build.gradle—Gradle build files

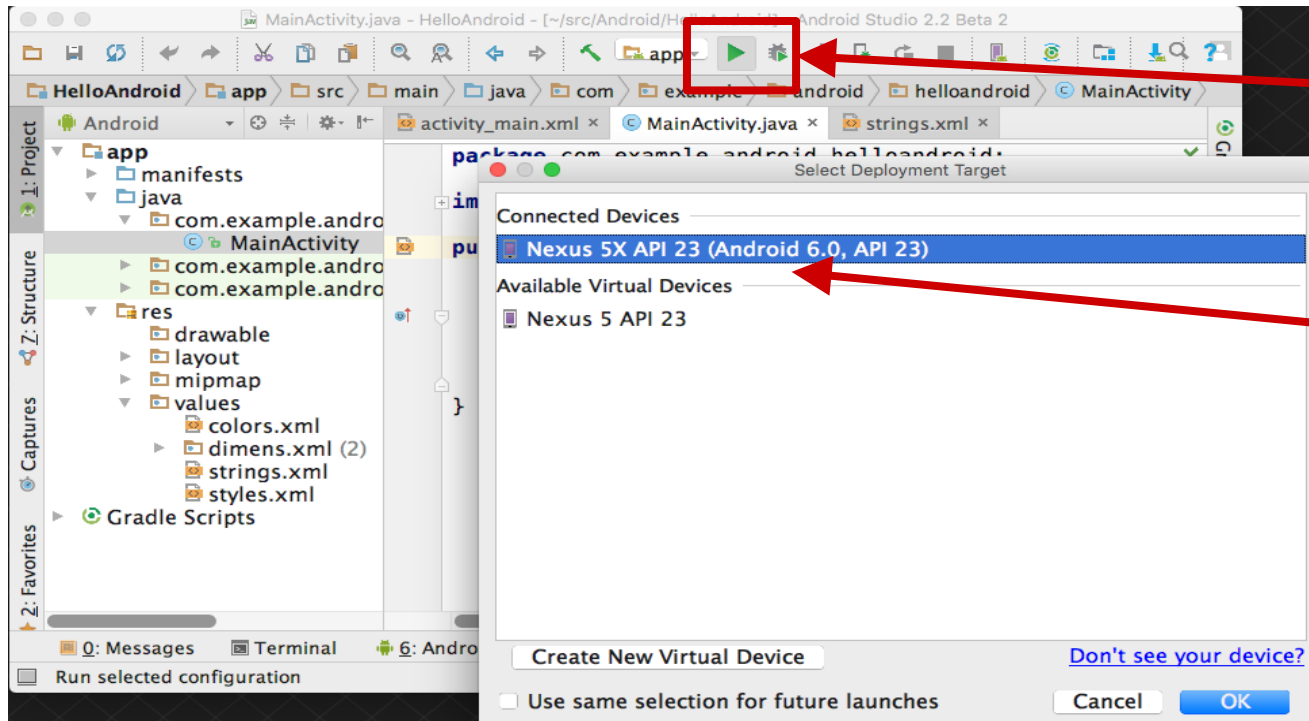


Gradle build system

- Modern build subsystem in Android Studio
- Three build.gradle:
 - project
 - module
 - settings
- Typically not necessary to know low-level Gradle details
- Learn more about gradle at <https://gradle.org/>



Run your app



1. Run

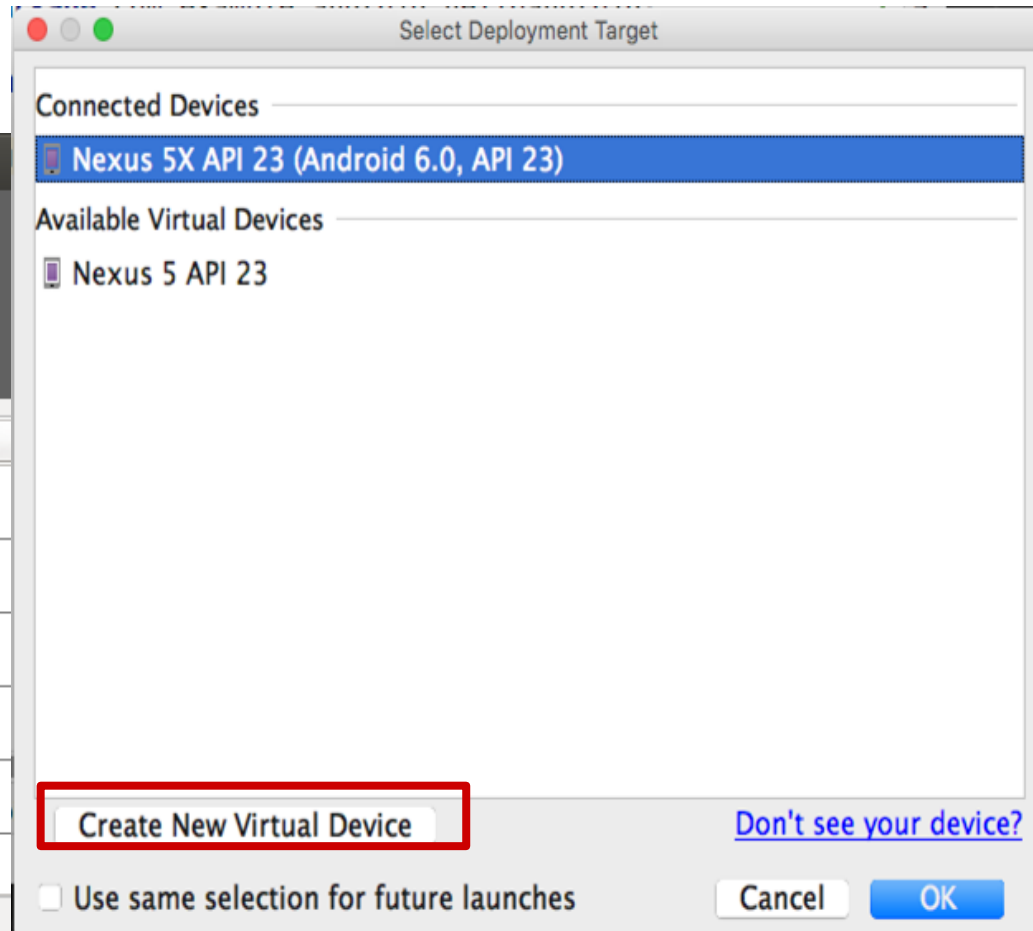
2. Select virtual
or physical
device

3. OK

Create a virtual device

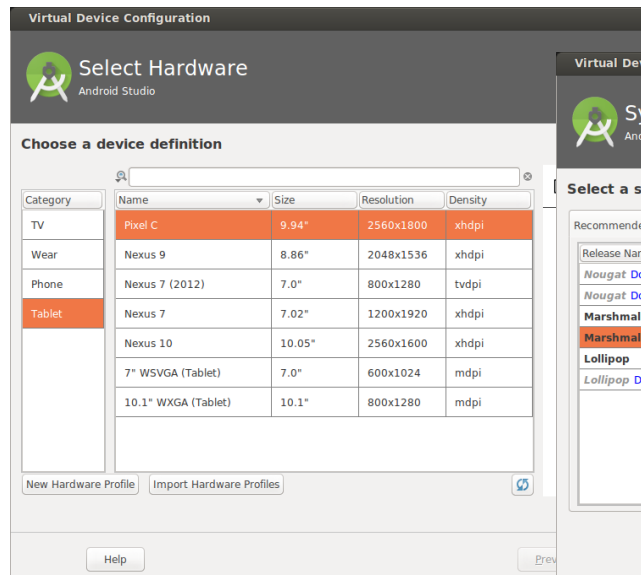
Use emulators to test app on different versions of Android and form factors.

Tools > Android > AVD Manager or:

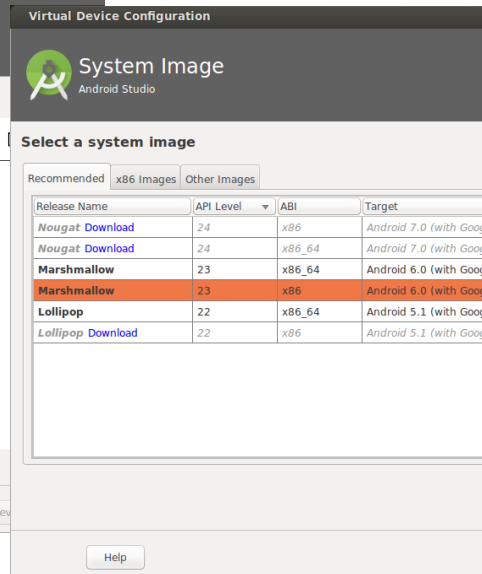


Configure virtual device

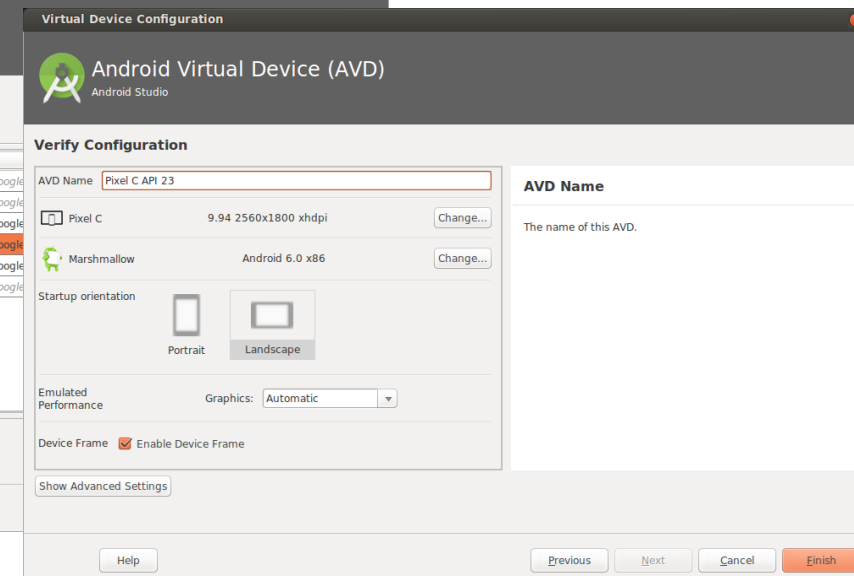
1. Choose hardware



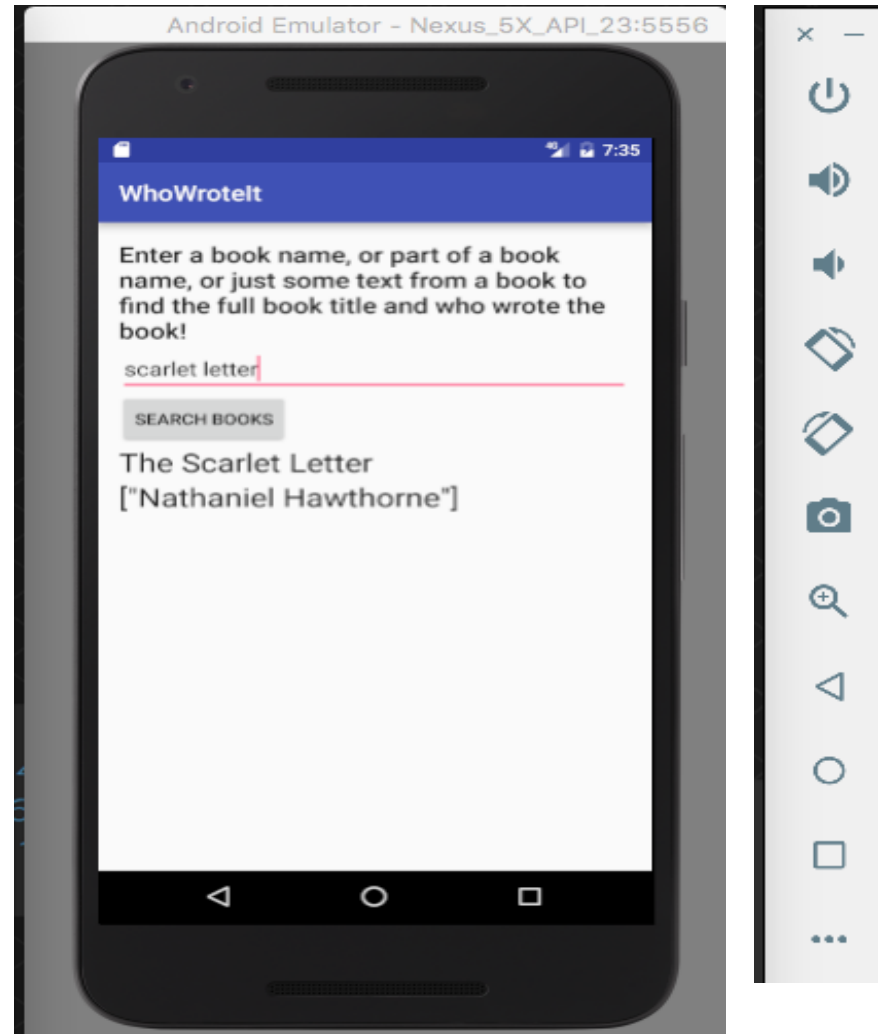
2. Select Android Version



3. Finalize



Run on a virtual device



Run on a physical device

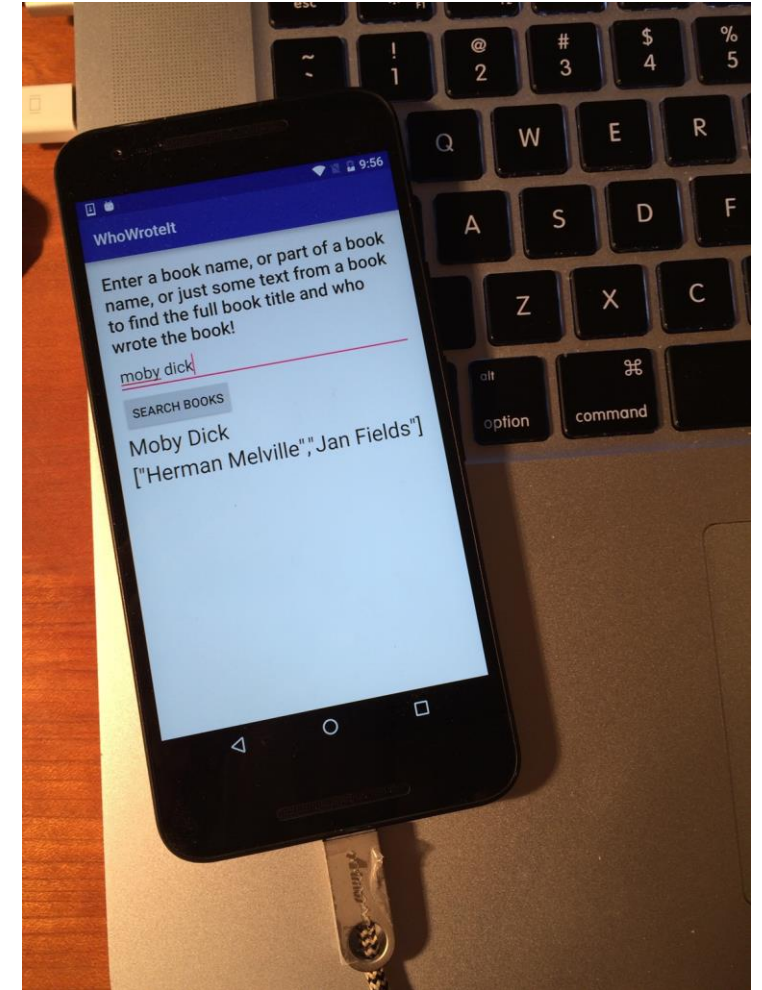
1. Turn on Developer Options:
 - a. **Settings > About phone**
 - b. Tap **Build number** seven times
2. Turn on USB Debugging
 - a. **Settings > Developer Options > USB Debugging**
3. Connect phone to computer with cable

Windows/Linux additional setup:

- [Using Hardware Devices](#)

Windows drivers:

- [OEM USB Drivers](#)



Learn more

- **Android History** <https://www.android.com/history/#/marshmallow>
- **Introduction to Android** <https://developer.android.com/guide/index.html>
- **Platform Architecture** <https://developer.android.com/guide/platform/index.html>
- **UI Overview** <https://developer.android.com/guide/topics/ui/overview.html>
- **Platform Versions** <https://developer.android.com/about/dashboards/index.html>
- **Supporting Different Platform Versions** <https://developer.android.com/training/basics/supporting-devices/platforms.html>
- **Android Studio User's Guide** <https://developer.android.com/studio/intro/index.html>



Learn more

- **Official Android documentation** <http://developer.android.com/index.html>
- **Image Asset Studio** <http://developer.android.com/tools/help/image-asset-studio.html>
- **Android Monitor page** <http://developer.android.com/tools/help/android-monitor.html>
- **Official Android blog** <http://officialandroid.blogspot.com/>
- **Android Developers blog** <http://android-developers.blogspot.com>
- **Google I/O Codelabs** <http://codelabs.developers.google.com>
- **Stack Overflow** <http://stackoverflow.com/>
- **Android vocabulary** <http://developers.google.com/android/for-all/vocab-words>
- **Google Developer Training website** <http://developers.google.com/training>



What's Next?

Views, Layouts, and Resources

