CS260 Intro to Java & Android
04.Android Intro

Winter 2017

Android - Getting Started

- Android SDK contains:
  - API Libraries
  - Developer Tools
  - Documentation
  - Sample Code

- Present development tools:
  - IntelliJ IDEA
  - Android Studio

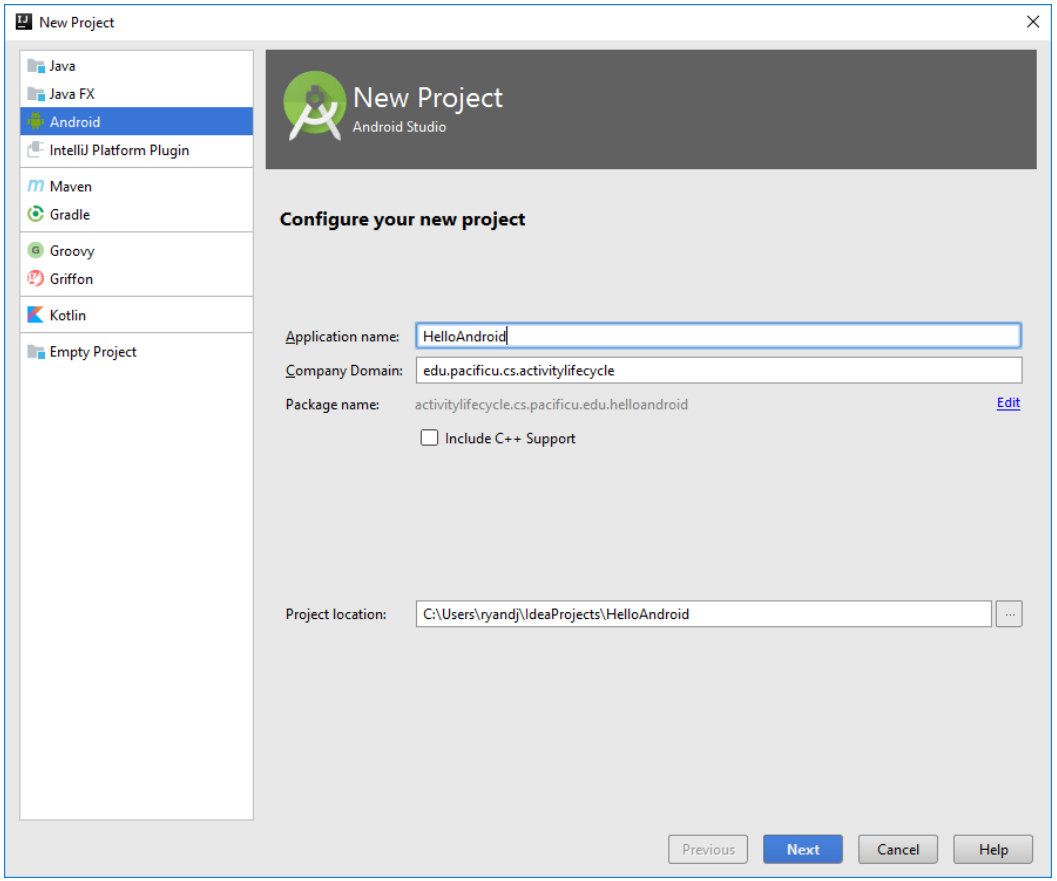
Android Portability

- Android applications run within the Dalvik virtual machine
- ART is a new Android runtime being introduced in 4.4

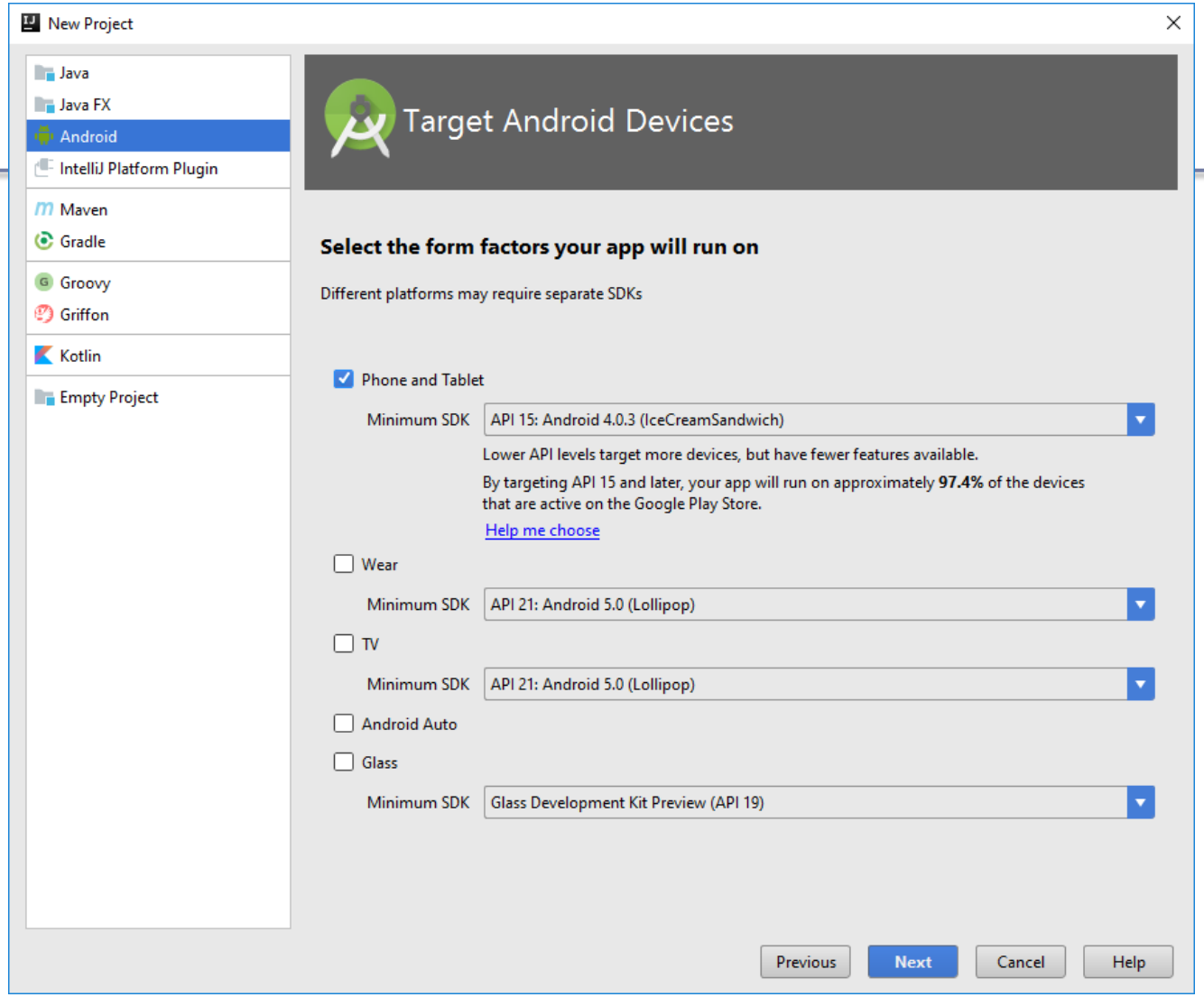
- Development Platforms:
  - Windows (XP, Windows, 7, 8)
  - Linux
  - Mac OS 10.4.8 or later (Intel chips only)

Android HelloAndroid Application

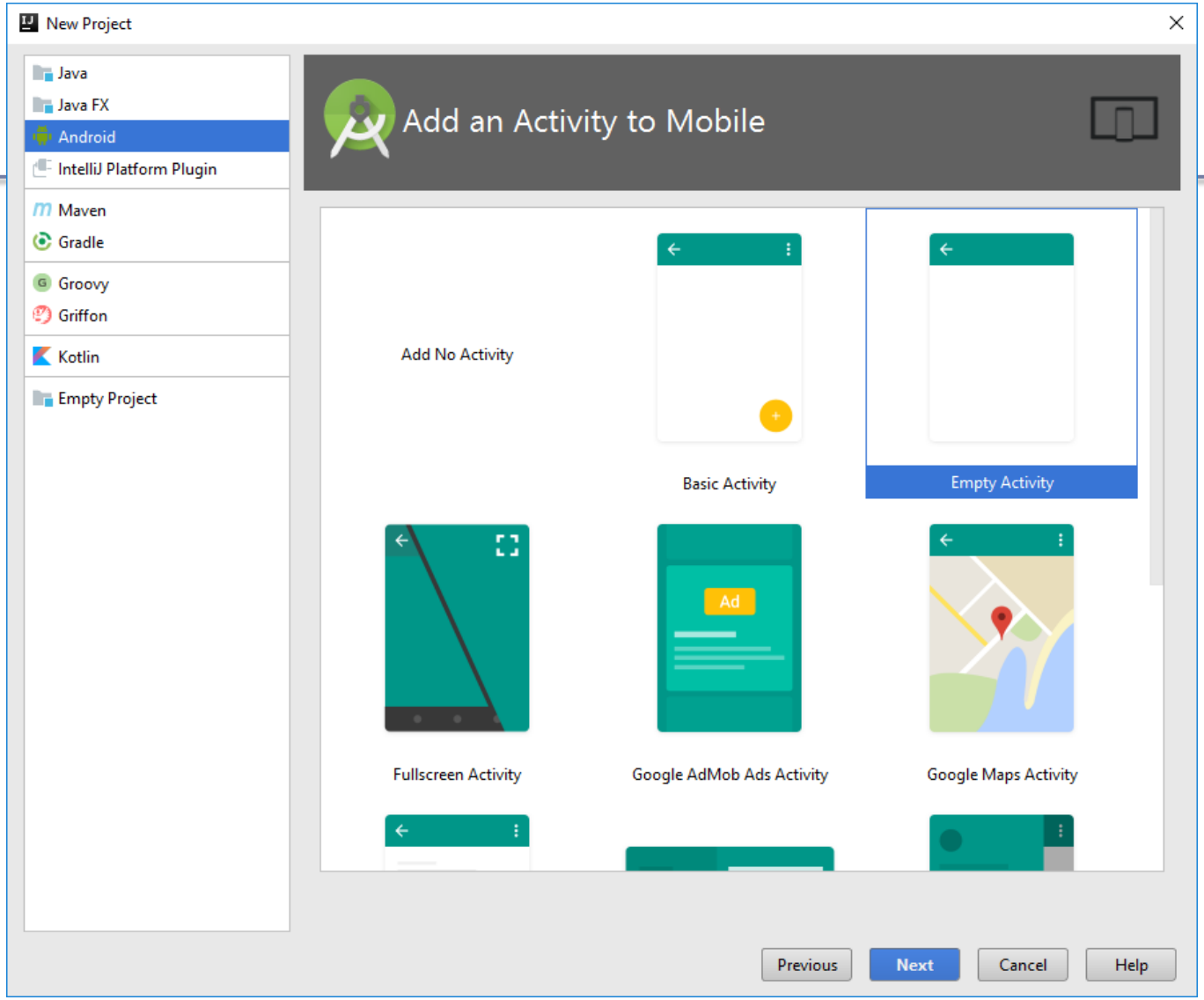
- Start Android Studio
- We will create our warm fuzzy HelloAndroid App



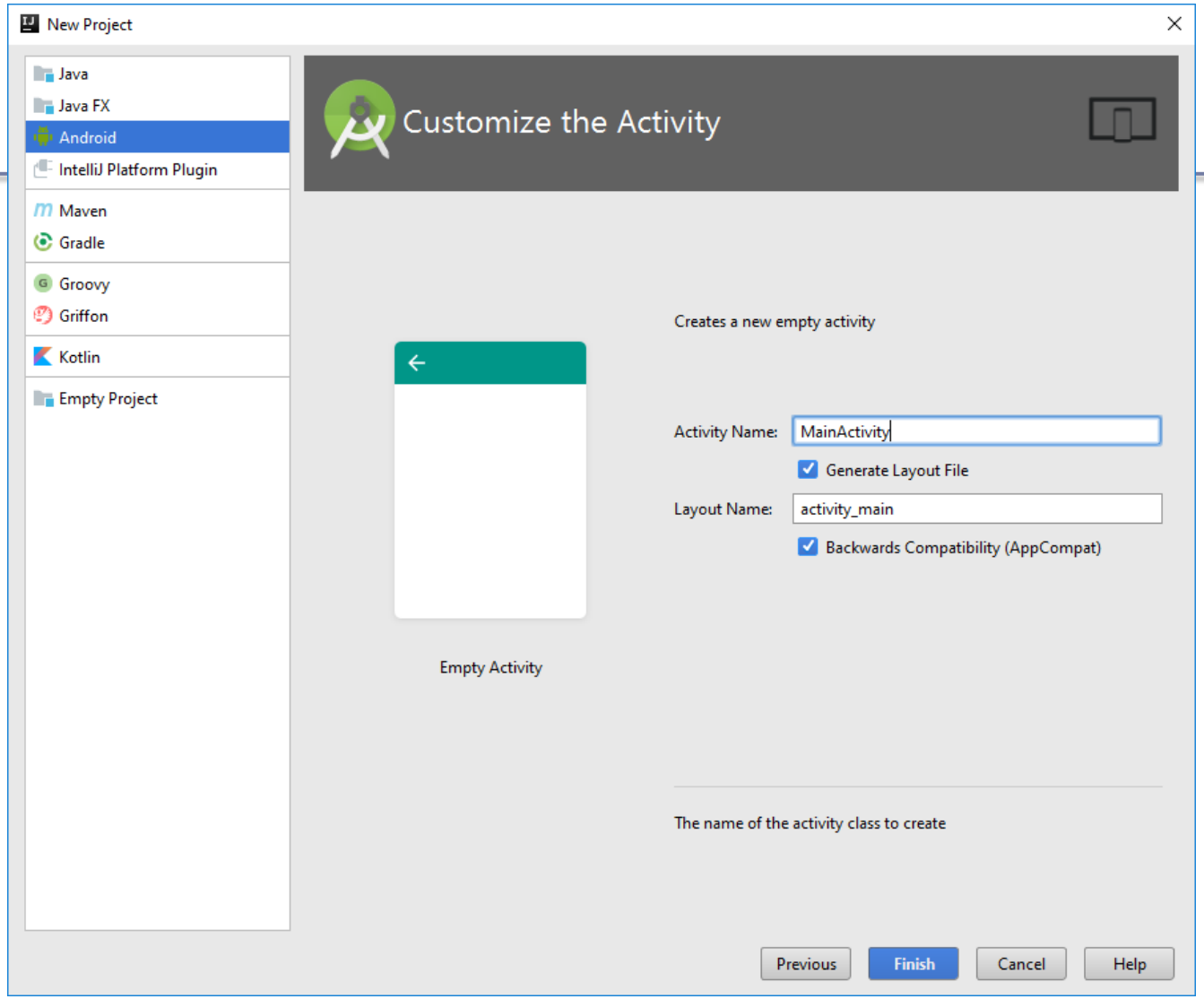
New Android Project



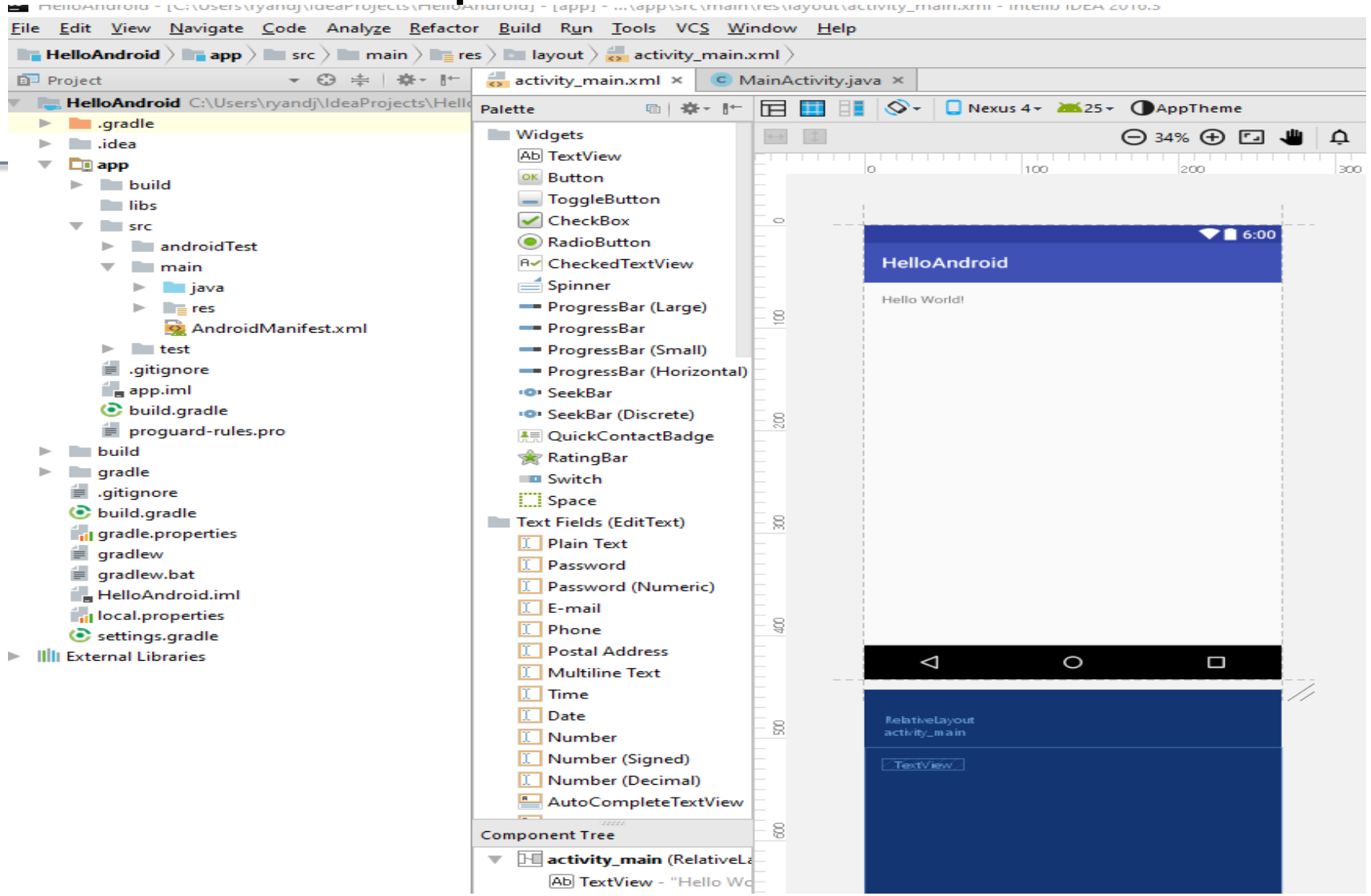
Click “Next” takes us to



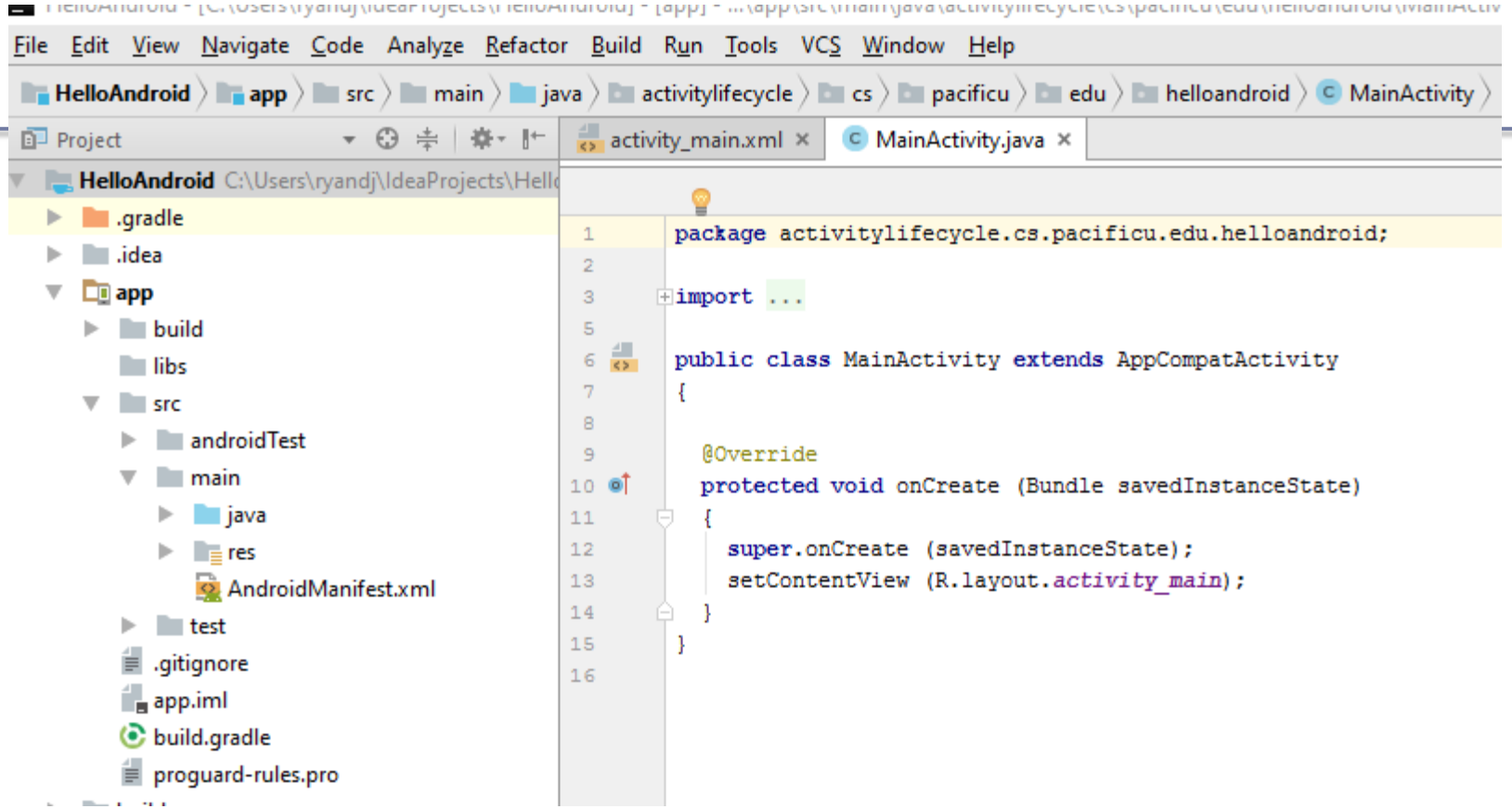
Click “Finish” takes us to



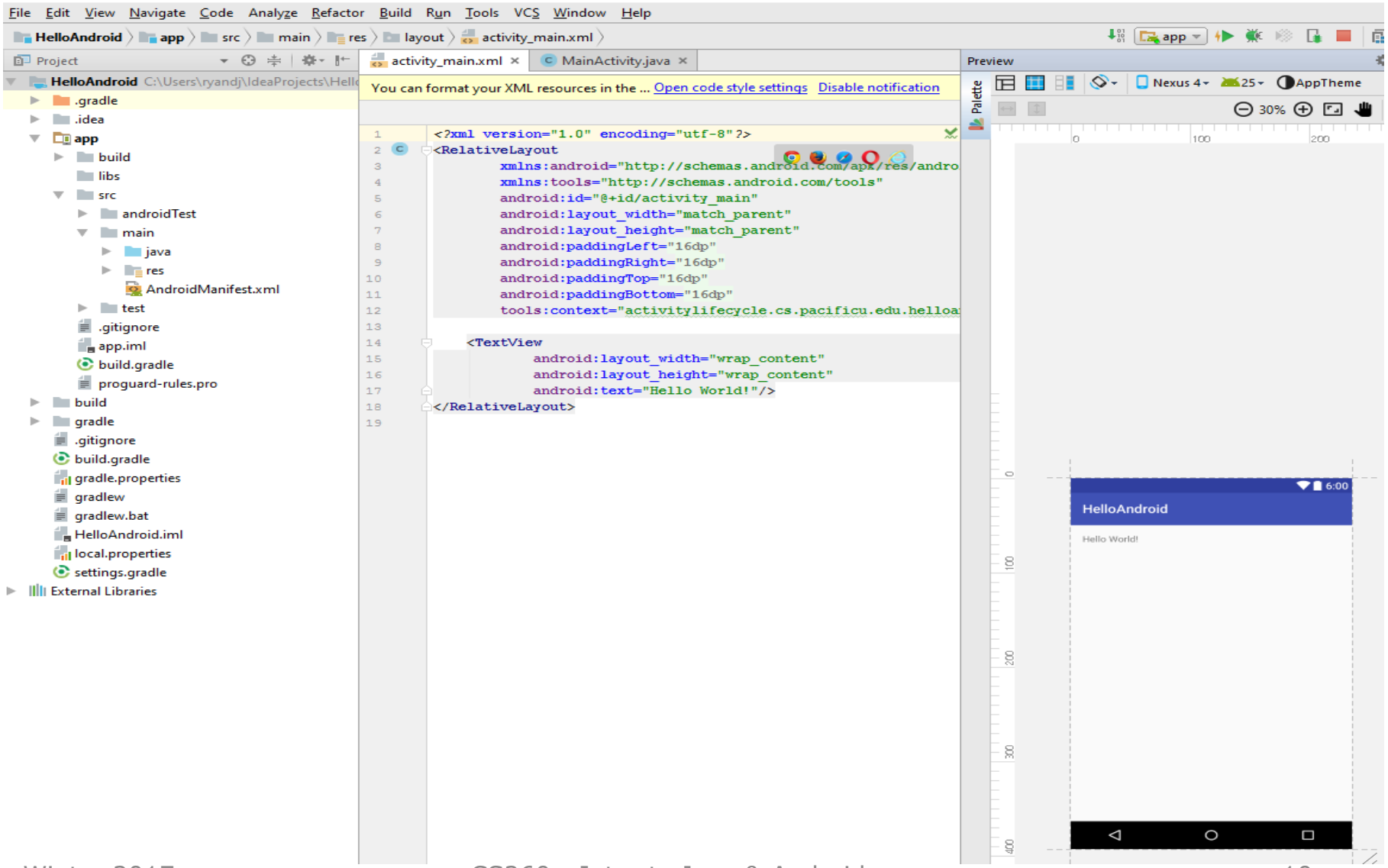
Development Environment



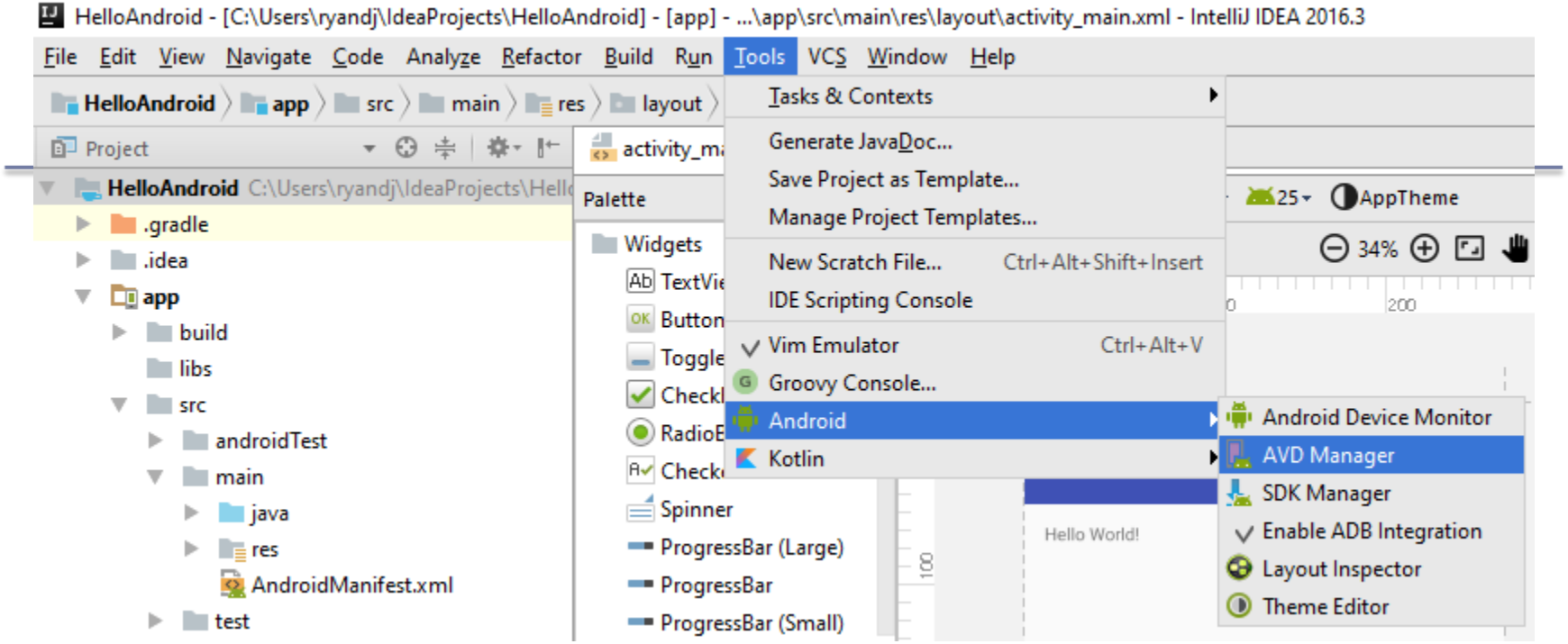
Development Environment

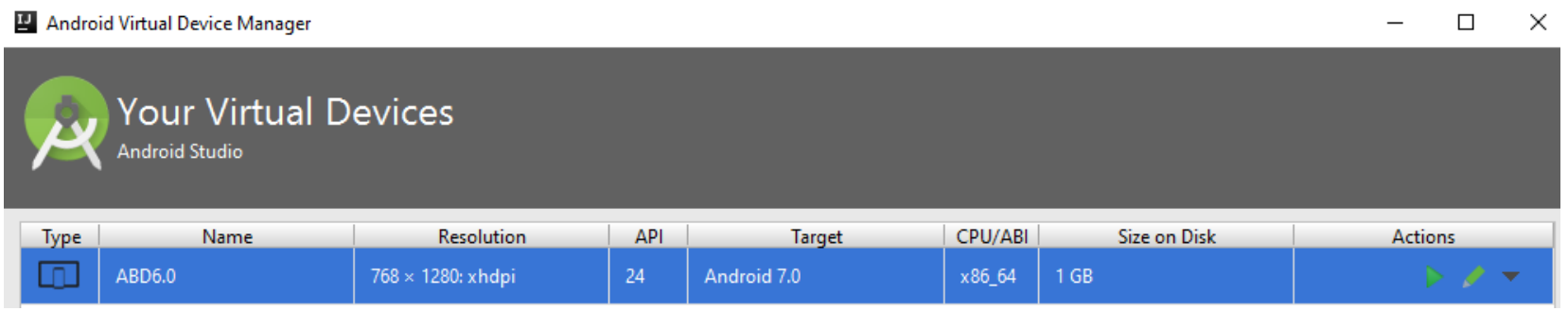


xml


File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
HelloAndroid app src main res layout activity_main.xml
Project
HelloAndroid C:\Users\ryandj\IdeaProjects\Hell
.gradle
.idea
app
build
libs
src
androidTest
main
java
res
AndroidManifest.xml
test
.gitignore
app.iml
build.gradle
proguard-rules.pro
build
gradle
.gitignore
build.gradle
gradle.properties
gradlew
gradlew.bat
HelloAndroid.iml
local.properties
settings.gradle
External Libraries
activity_main.xml
MainActivity.java
You can format your XML resources in the ... Open code style settings Disable notification
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/andro
xmlns:tools="http://schemas.android.com/tools"
android:id="@+id/activity_main"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:paddingLeft="16dp"
android:paddingRight="16dp"
android:paddingTop="16dp"
android:paddingBottom="16dp"
tools:context="activitylifecycle.cs.pacificu.edu.helloa
<TextView
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Hello World!"/>
</RelativeLayout>
Preview
Palette
Nexus 4
25
AppTheme
30%
6:00
HelloAndroid
Hello World!

Creating Virtual Devices





Important Android Dates

- Google acquires Android, August 2005
- Open Handset Alliance (OHA) announced, November 2007. OHA developed Android and is "…committed to commercially deploy handsets and services using the Android Platform." [10]
- First Android Phone, G1, October 2008
- Android SDK 1.0, October 2008

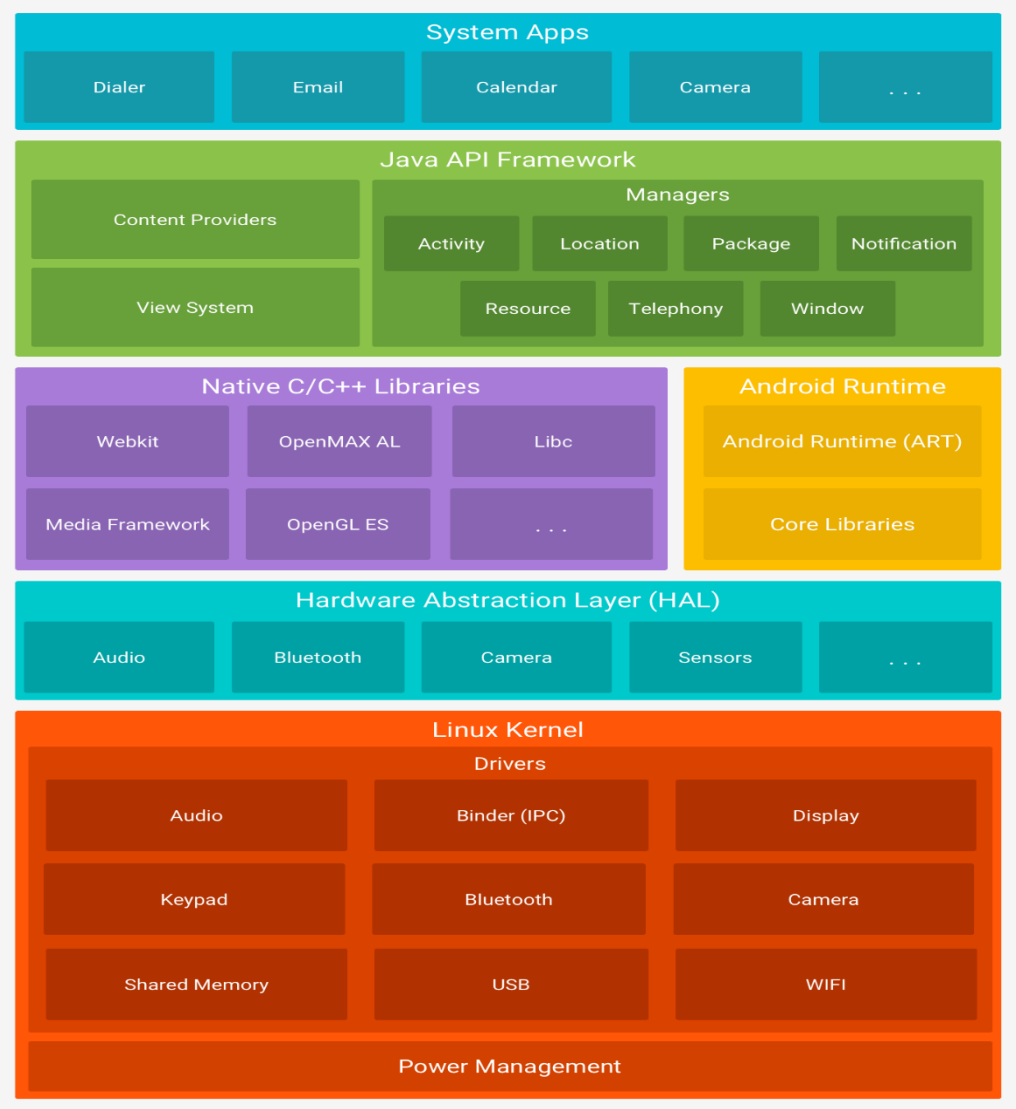
What is Android?

- Android is a software stack (set of programs working together) for mobile devices that includes:
  - an operating system
  - middleware
  - applications

Application Fundamentals

- Written in Java (could be C/C++ native)
- SDK tools compile code into APK (Android application package … e.g. helloandroid.apk)
- One .apk file contains all app contents

Android Architecture

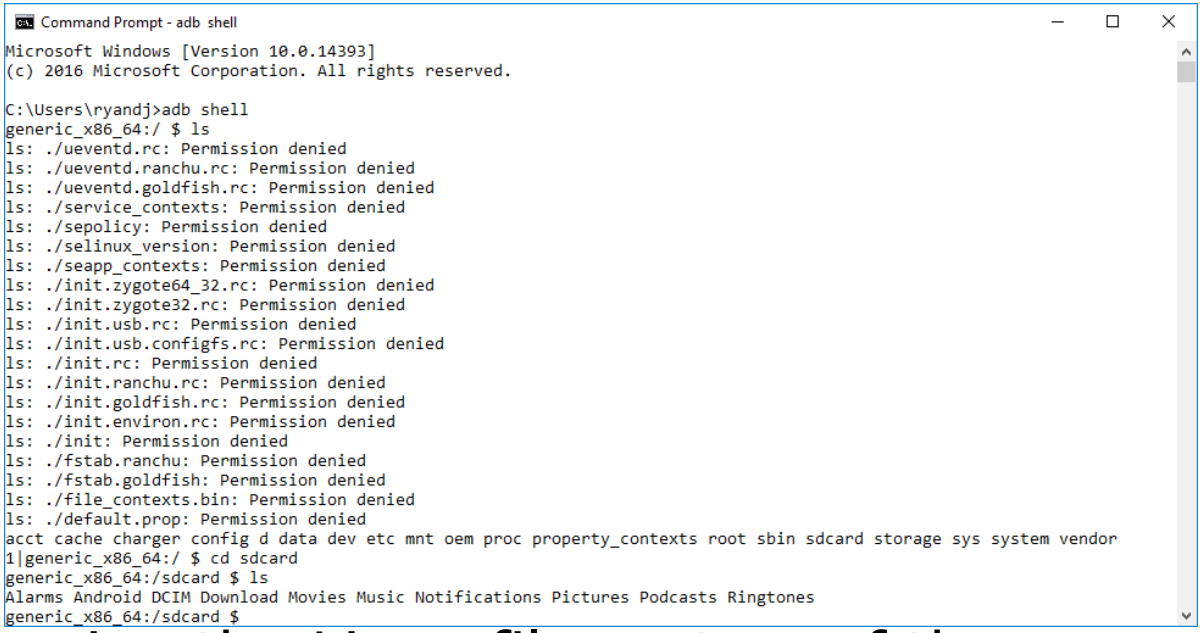


Linux Kernel

- Android relies on Linux version 2.6 (3.x from Android 4.0 Ice Cream Sandwich) for:
  - memory management
  - process management
  - security
  - networking
- You will not make Linux system calls
- Some utilities interact with Linux
  - e.g. adb shell

adb shell

- With an emulator running, open a Windows command shell
- Type adb shell
- Type ls



- Now you can examine the Linux file system of the phone which aids in in debugging

Native Libraries

- The native libraries are written in C & C++
- The libraries are exposed through the Application framework

Application Framework

- Android developers have access to the same framework APIs use by the core applications

- Services and systems for applications include:
  - **Views** – including lists, grids, buttons, ...
  - **Content Providers** – methods for accessing data
  - **Resource Manager** – organizes non-code resources such as strings and layout files
  - **Notification Manager** – displays custom alerts
  - **Activity Manager** – manages lifecycle of applications

Android Runtime

Every Application:

- Runs in its own process space
- Has a separate instance of the Dalvik VM
  - The Dalvik VM uses the Linux kernel for functionality such as threading and low-level memory management
  - Dalvik VM != JVM
- All Android code is written in Java and run within the Dalvik VM

What is Dalvik?

- Dalvik is a VM optimized for low memory requirements
- Android code is compiled into bytecodes executed by the Dalvik VM
- bytecodes are machine-independent instructions

Android Applications

- Apps are written in Java
- Code is compiled into Android package (.apk file)
- All code (including data & resource files) in .apk is one application

Android Application Specifics

- Android is a multi-user Linux system where each application is a user
- Only one application is visible at a time
- Each process has its own VM running an application in isolation
- Two or more applications can share data
- Applications consist of one or more activities

What is an Activity?

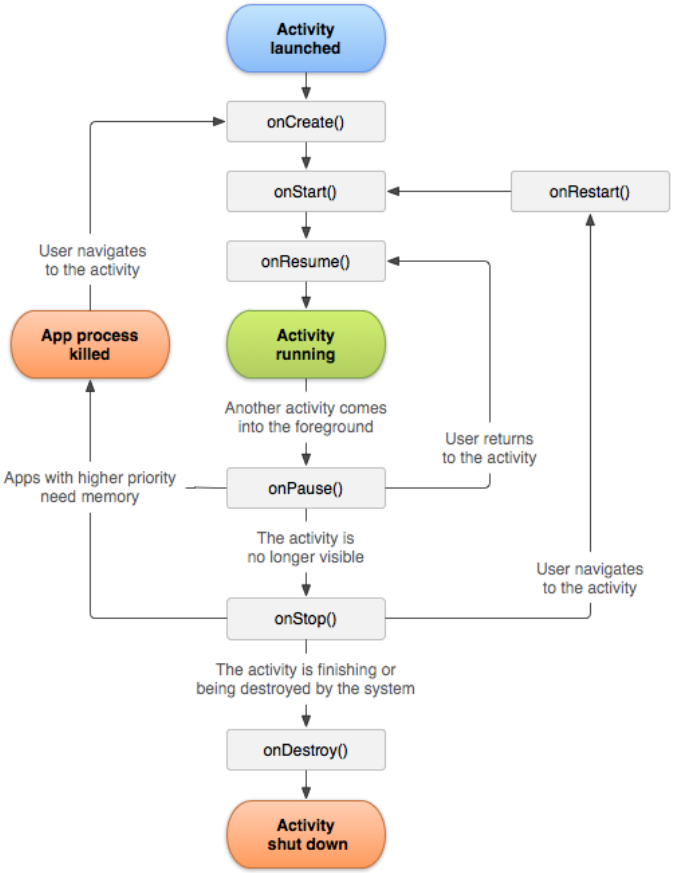
- An Activity represents a single screen with a UI
- Ex: Email Application consists of activities for
  - Showing list of emails
  - Composing an email
  - Reading an email
- Each activity is independent
- Other applications can use a particular activity if the email application gives permission to do so

Activity Lifecycle

Activity – a process that performs some specific action

- Every Android application is made up of one or more activities managed on an Activity Stack (AS) or the "back stack".
- A new activity is always placed on top of the AS and then becomes the running activity.
- The AS is LIFO; therefore, when the Back button is pressed the current activity is popped and destroyed

Activity Lifecycle Visual



Activity States

An activity has essentially four states:

- **running** – in the foreground of the screen
- **paused** – lost focus but still visible with all state maintained
  - How? A new activity that is transparent or not full sized is running on top of the stack
- **stopped** – a new activity completely obscures another activity
  - The stopped activity is no longer visible
  - State is maintained
- **destroyed** – the activity must be completely restarted and the state information must be

Activity Skeleton

```
 6  public class MainActivity extends Activity
 7  {
 8
 9    @Override
10    protected void onCreate (Bundle savedInstanceState)
11    {  // The activity is being created
12      super.onCreate (savedInstanceState);
13    }
14    @Override
15    protected void onStart ()
16    { // The activity is about to become visible
17      super.onStart ();
18    }
19    @Override
20    protected void onResume ()
21    { // The activity has become visible (is is now "resumed")
22      super.onResume ();
23    }
24    @Override
25    protected void onPause ()
26    { // Another activity is taking focus
27      super.onPause ();
28    }
29    @Override
30    protected void onStop ()
31    { // The activity is no longer visible (it is now "stopped")
32      super.onStop ();
33    }
34    @Override
35    protected void onDestroy ()
36    { // The activity is about to be destroyed
37      super.onDestroy ();
38    }
39    @Override
40    protected void onRestart ()
41    { // The user returns to the activity
42      super.onRestart ();
43    }
```

ActivityLifeCycleDemo Application

Copy the Android Project **ActivityLifeCycle** from CS260-01Public

1. Place the file in AndroidStudioProjects on your local machine
2. Let's take a look at the source code
3. Run the application

Q1: What is the difference between hitting the home button (HOME) and back button (ESC) ?

Q2: What is Log.v and how can it be used?