



CS260 Intro to Java & Android

04.Android Intro

Winter 2018

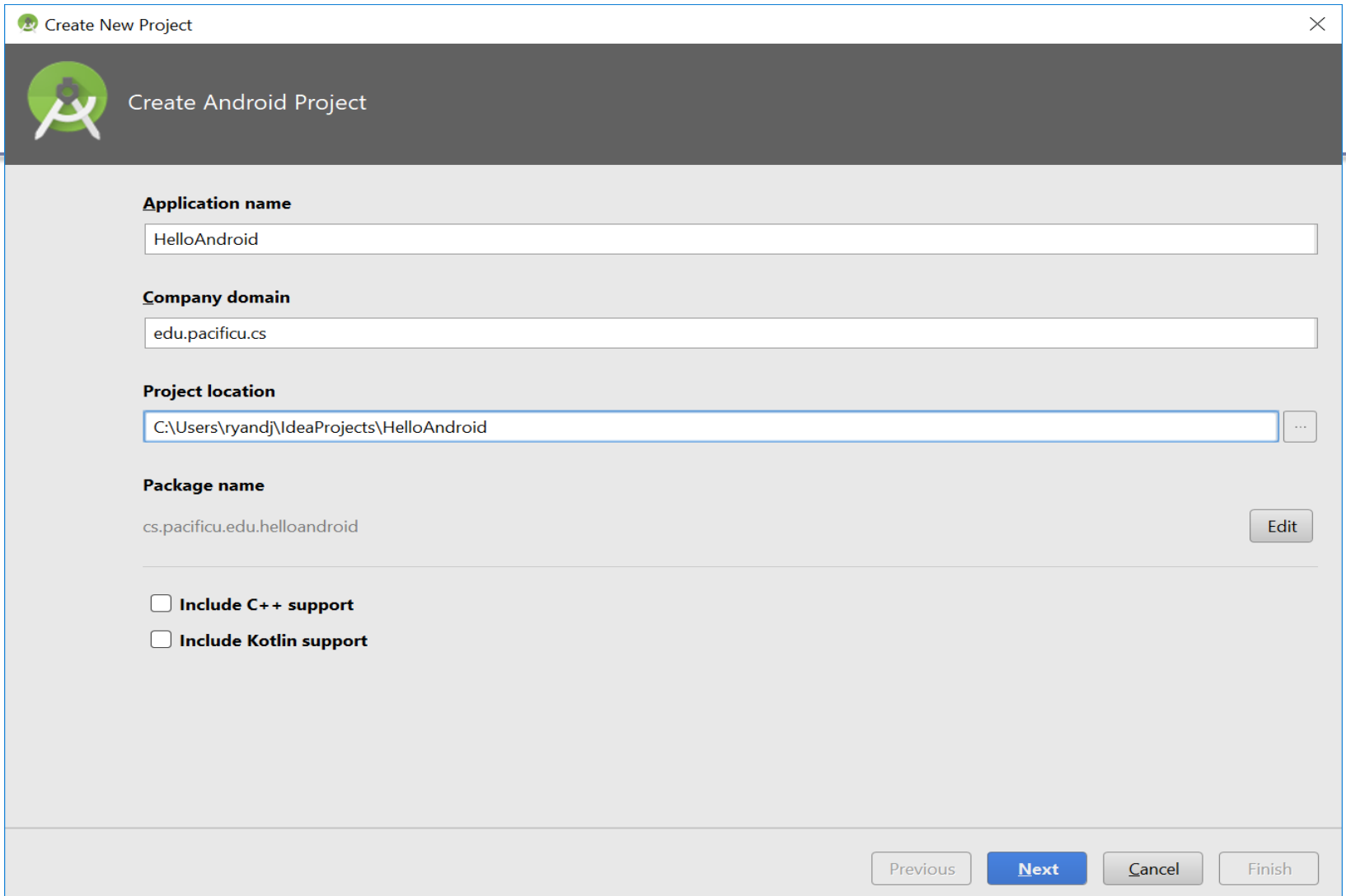
Android - Getting Started

- 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



Create New Project

Create Android Project

Application name
HelloAndroid

Company domain
edu.pacificu.cs

Project location
C:\Users\ryandj\IdeaProjects\HelloAndroid

Package name
cs.pacificu.edu.helloandroid Edit


Include C++ support

Include Kotlin support

Previous **Next** Cancel Finish

New Android Project

Create New Project ✕

 Target Android Devices

Select the form factors and minimum SDK
Some devices require additional SDKs. Low API levels target more devices, but offer fewer API features.

Phone and Tablet
API 15: Android 4.0.3 (IceCreamSandwich) ▼
By targeting **API 15 and later**, your app will run on approximately **100%** of devices. [Help me choose](#)
 Include Android Instant App support

Wear
API 23: Android 6.0 (Marshmallow) ▼

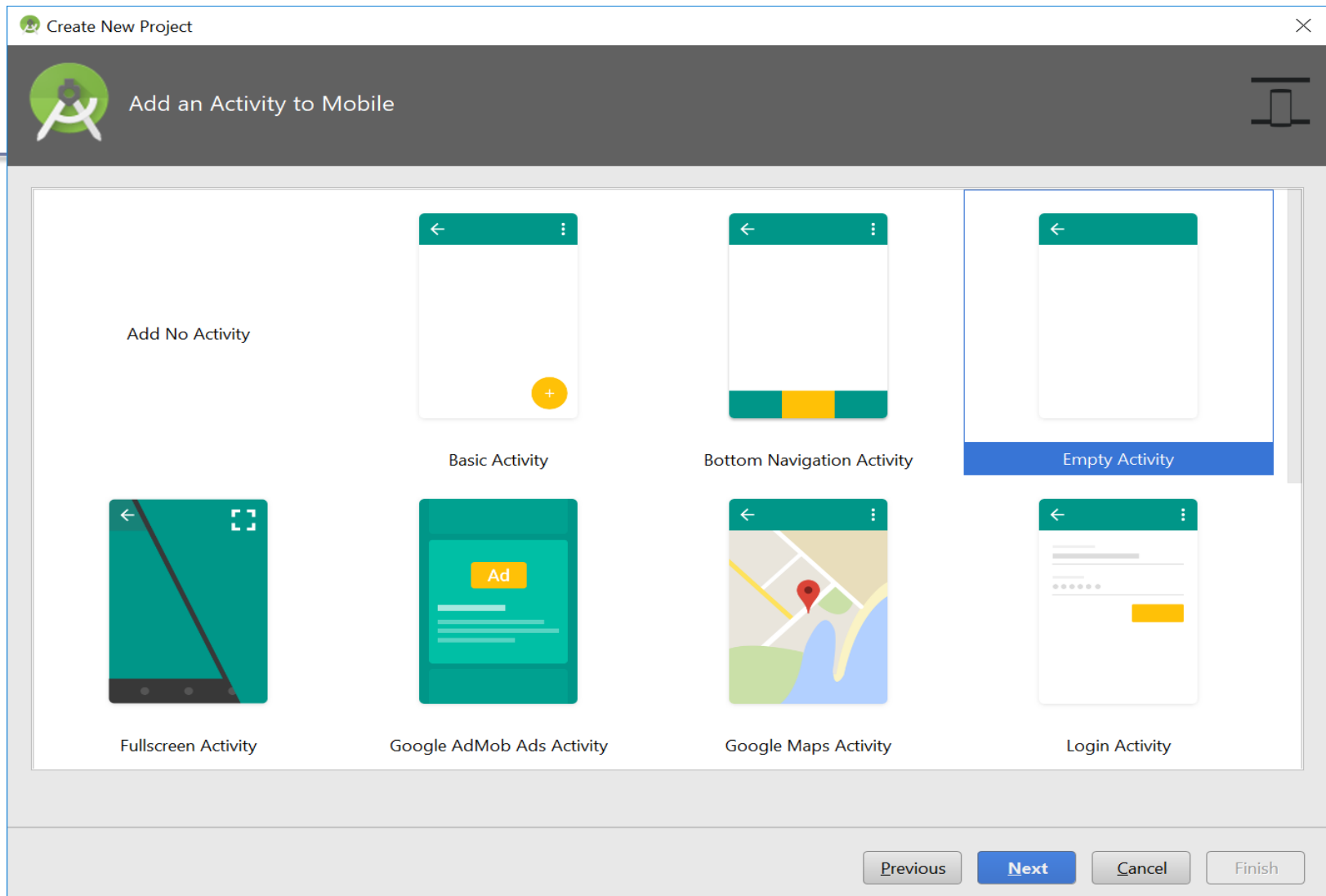
TV
API 21: Android 5.0 (Lollipop) ▼

Android Auto

Android Things
API 24: Android 7.0 (Nougat) ▼

Previous **Next** Cancel Finish

Click "Next" takes us to



Click "Finish" takes us to

Create New Project

Configure Activity

Creates a new empty activity

←

Activity Name

MainActivity

Generate Layout File

Layout Name

activity_main

Backwards Compatibility (AppCompat)

Previous Next Cancel Finish

Development Environment

HelloAndroid - [C:\Users\ryandj\IdeaProjects\HelloAndroid] - [app] - ...\app\src\main\java\cs\pacificu\edu\helloandroid\MainActivity.jav

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

HelloAndroid app src main java cs pacificu edu helloandroid MainActivity

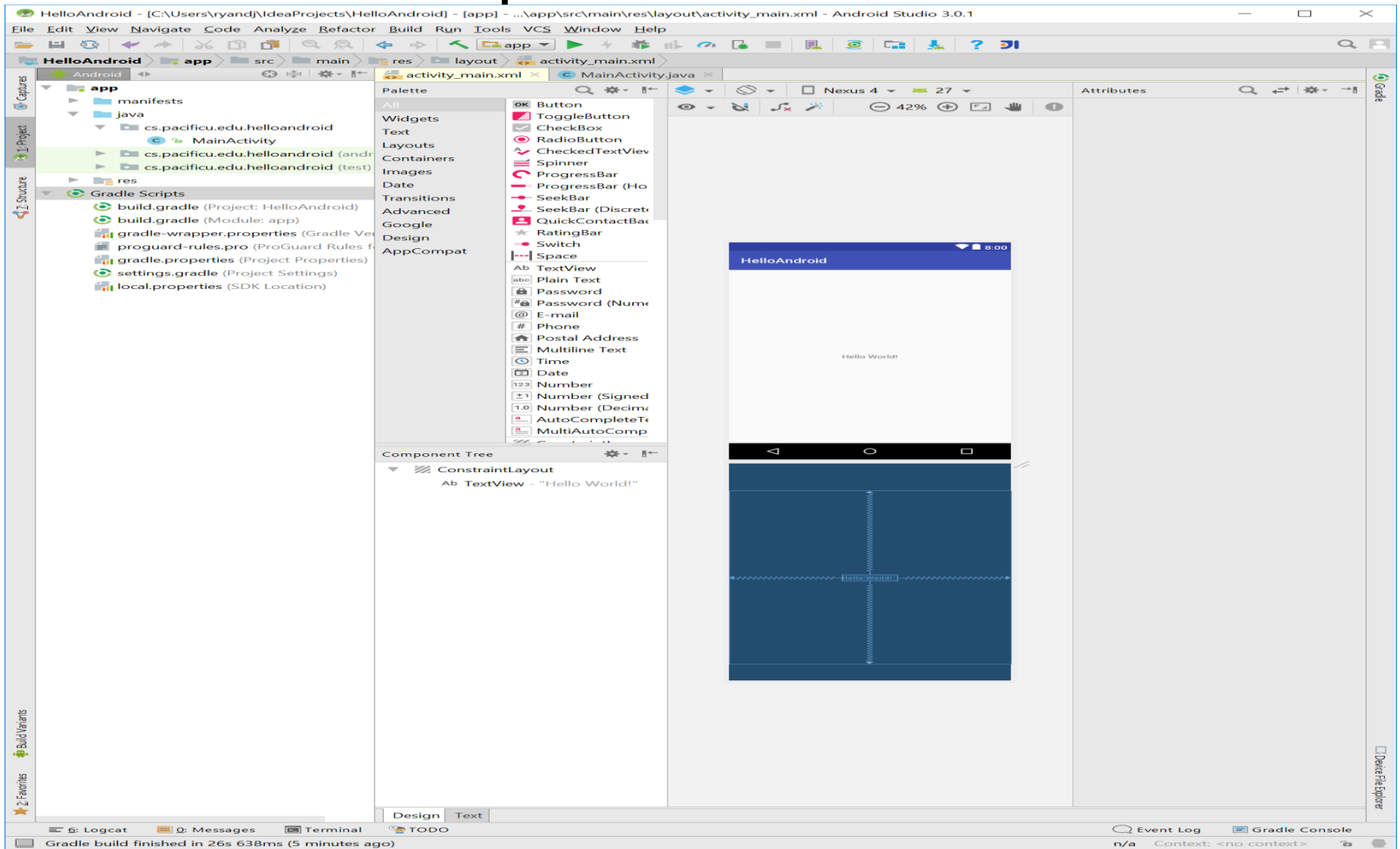
Android activity_main.xml x MainActivity.java x

Project Structure

- app
 - manifests
 - java
 - cs.pacificu.edu.helloandroid
 - MainActivity
 - cs.pacificu.edu.helloandroid (andr)
 - cs.pacificu.edu.helloandroid (test)
 - res
 - Gradle Scripts
 - build.gradle (Project: HelloAndroid)
 - build.gradle (Module: app)
 - gradle-wrapper.properties (Gradle Ver)

```
1 package cs.pacificu.edu.helloandroid;
2
3 import ...
4
5
6 public class MainActivity extends AppCompatActivity
7 {
8
9     @Override
10    protected void onCreate (Bundle savedInstanceState)
11    {
12        super.onCreate (savedInstanceState);
13        setContentView (R.layout.activity_main);
14    }
15 }
```

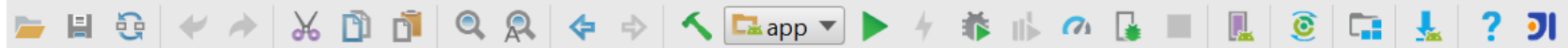

Development Environment



xml

HelloAndroid - [C:\Users\ryandj\IdeaProjects\HelloAndroid] - [app] - ...app\src\main\res\layout\activity_main.xml - Android Studio 3.0.1

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help



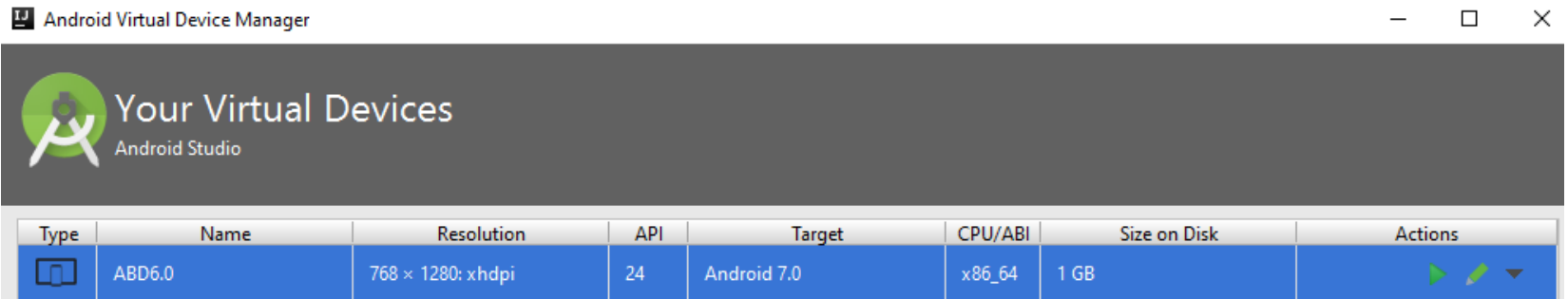
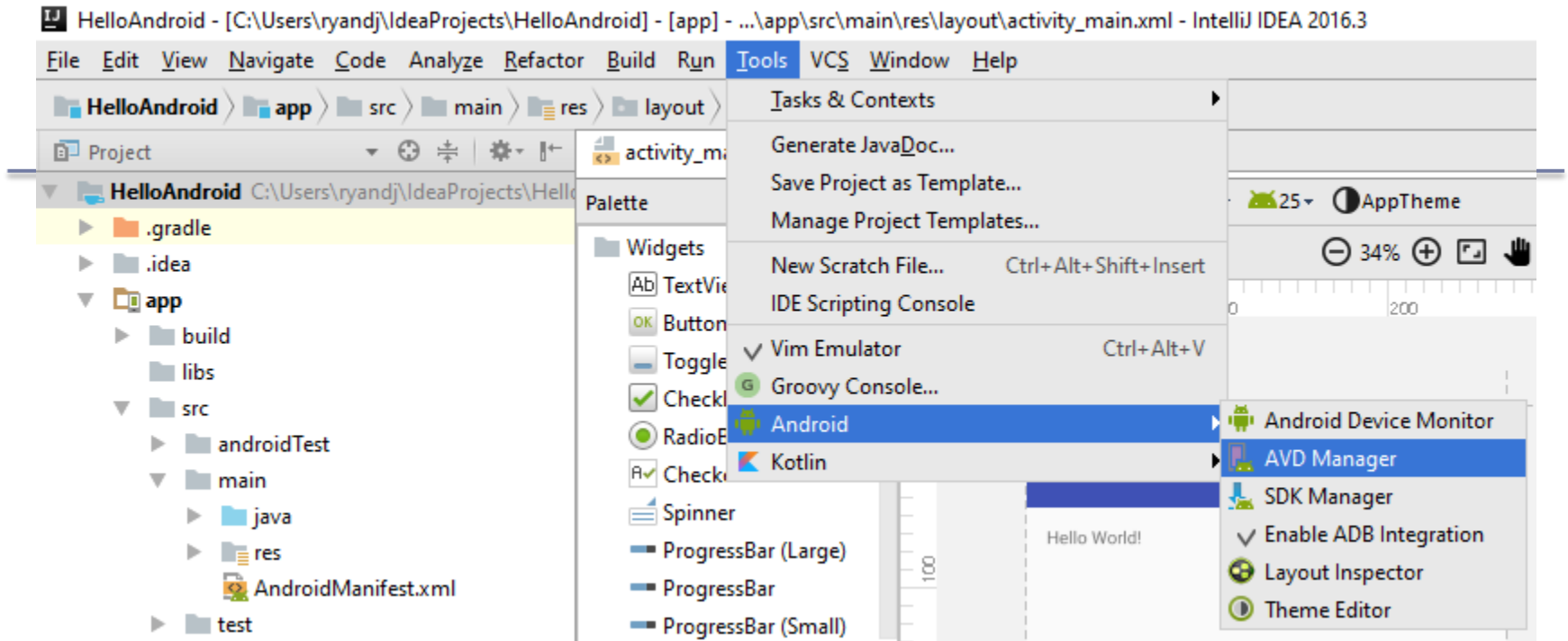
HelloAndroid > app > src > main > res > layout > activity_main.xml

Android

- app
 - manifests
 - java
 - cs.pacificu.edu.helloandroid
 - MainActivity
 - cs.pacificu.edu.helloandroid (android)
 - cs.pacificu.edu.helloandroid (test)
 - res
 - Gradle Scripts
 - build.gradle (Project: HelloAndroid)
 - build.gradle (Module: app)
 - gradle-wrapper.properties (Gradle Wrapper)
 - proguard-rules.pro (ProGuard Rules File)
 - gradle.properties (Project Properties)
 - settings.gradle (Project Settings)
 - local.properties (SDK Location)

```
1 |<?xml version="1.0" encoding="utf-8" ?>
2 |<android.support.constraint.ConstraintLayout
3 |    xmlns:android="http://schemas.android.com/apk/res/android"
4 |    xmlns:app="http://schemas.android.com/apk/res-auto"
5 |    xmlns:tools="http://schemas.android.com/tools"
6 |    android:layout_width="match_parent"
7 |    android:layout_height="match_parent"
8 |    tools:context="cs.pacificu.edu.helloandroid.MainActivity">
9 |
10 |    <TextView
11 |        android:layout_width="wrap_content"
12 |        android:layout_height="wrap_content"
13 |        android:text="Hello World!"
14 |        app:layout_constraintBottom_toBottomOf="parent"
15 |        app:layout_constraintLeft_toLeftOf="parent"
16 |        app:layout_constraintRight_toRightOf="parent"
17 |        app:layout_constraintTop_toTopOf="parent"/>
18 |
19 |</android.support.constraint.ConstraintLayout>
```

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

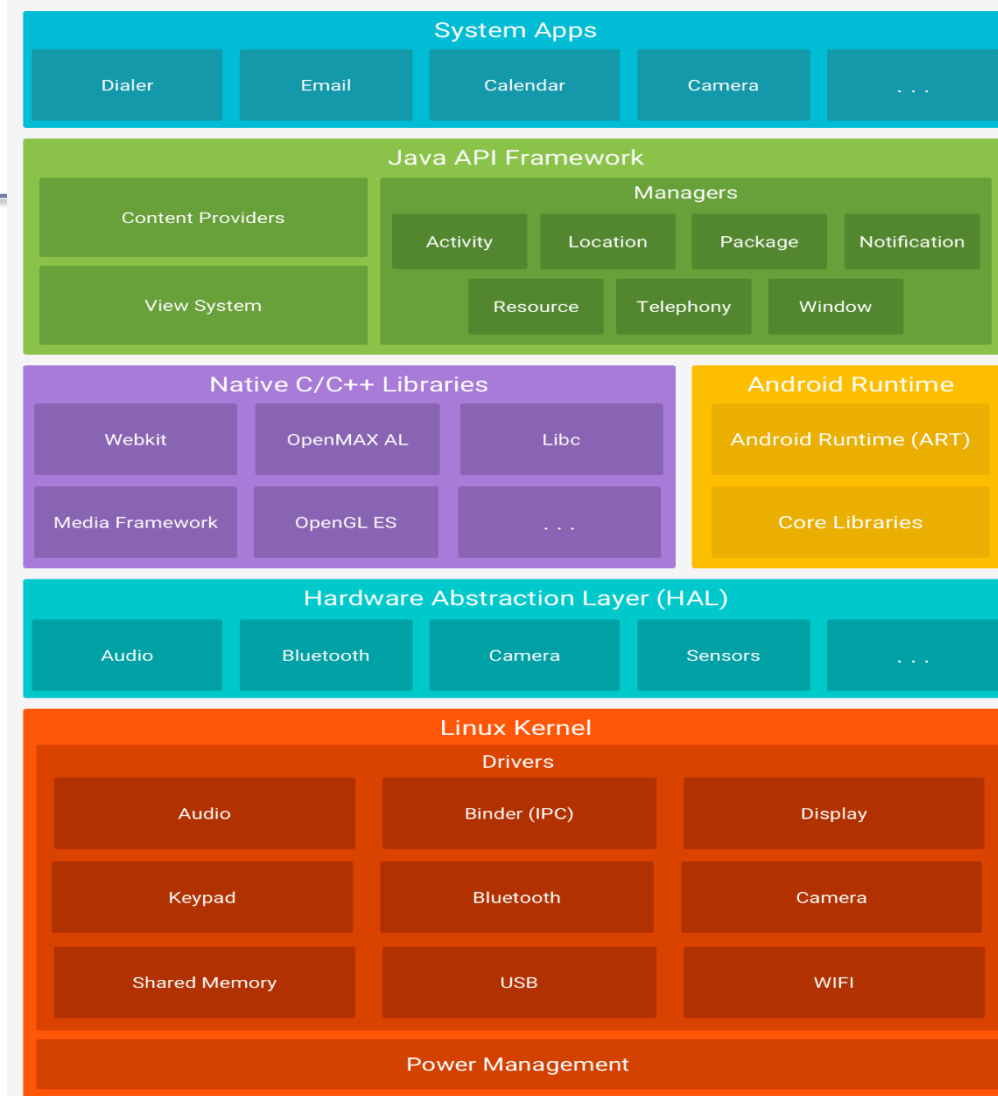
What is Android?

- 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

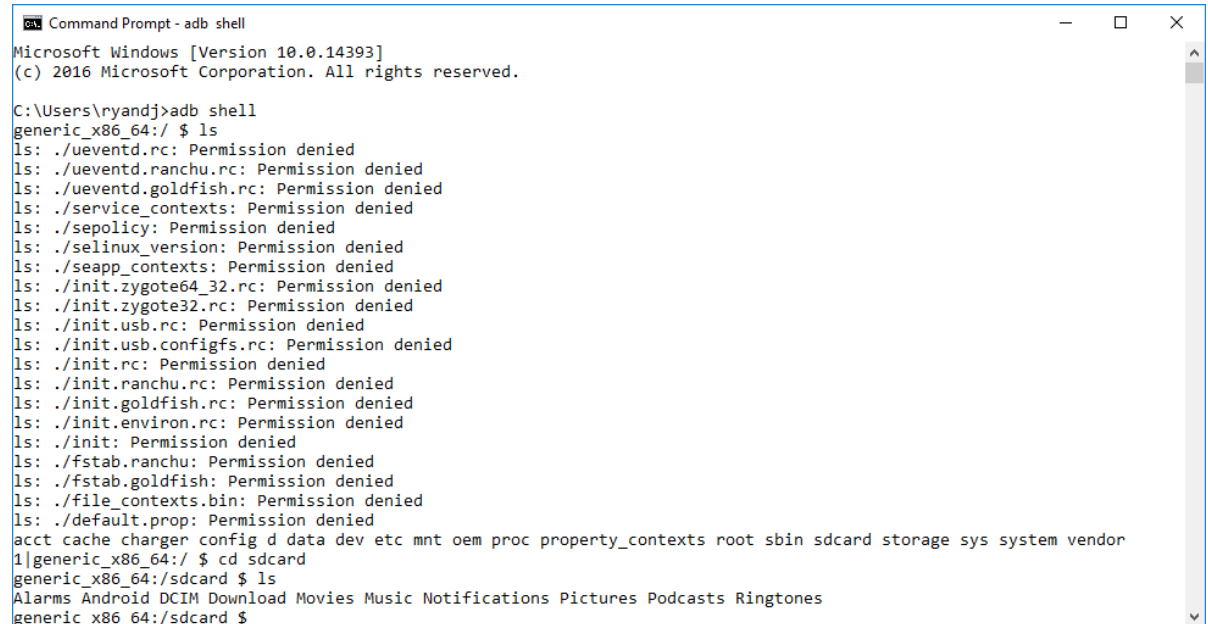


Linux Kernel

- 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



```
Command Prompt - adb shell
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\ryandj>adb shell
generic_x86_64:/ $ ls
ls: ./ueventd.rc: Permission denied
ls: ./ueventd.ranchu.rc: Permission denied
ls: ./ueventd.goldfish.rc: Permission denied
ls: ./service_contexts: Permission denied
ls: ./sepolicy: Permission denied
ls: ./selinux_version: Permission denied
ls: ./seapp_contexts: Permission denied
ls: ./init.zygote64_32.rc: Permission denied
ls: ./init.zygote32.rc: Permission denied
ls: ./init.usb.rc: Permission denied
ls: ./init.usb.configfs.rc: Permission denied
ls: ./init.rc: Permission denied
ls: ./init.ranchu.rc: Permission denied
ls: ./init.goldfish.rc: Permission denied
ls: ./init.envron.rc: Permission denied
ls: ./init: Permission denied
ls: ./fstab.ranchu: Permission denied
ls: ./fstab.goldfish: Permission denied
ls: ./file_contexts.bin: Permission denied
ls: ./default.prop: Permission denied
acct cache charger config d data dev etc mnt oem proc property_contexts root sbin sdcard storage sys system vendor
1|generic_x86_64:/ $ cd sdcard
generic_x86_64:/sdcard $ ls
Alarms Android DCIM Download Movies Music Notifications Pictures Podcasts Ringtones
generic_x86_64:/sdcard $
```

- Now you can examine the Linux file system of the phone which aids 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 Android Runtime (ART)
 - ART uses the Linux kernel for functionality such as threading and low-level memory management

Features of ART?

- Executes DEX files (special bytecode format)
- Ahead-of-time (AOT) and just-in-time (JIT) compilation
- Optimized garbage collection

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?

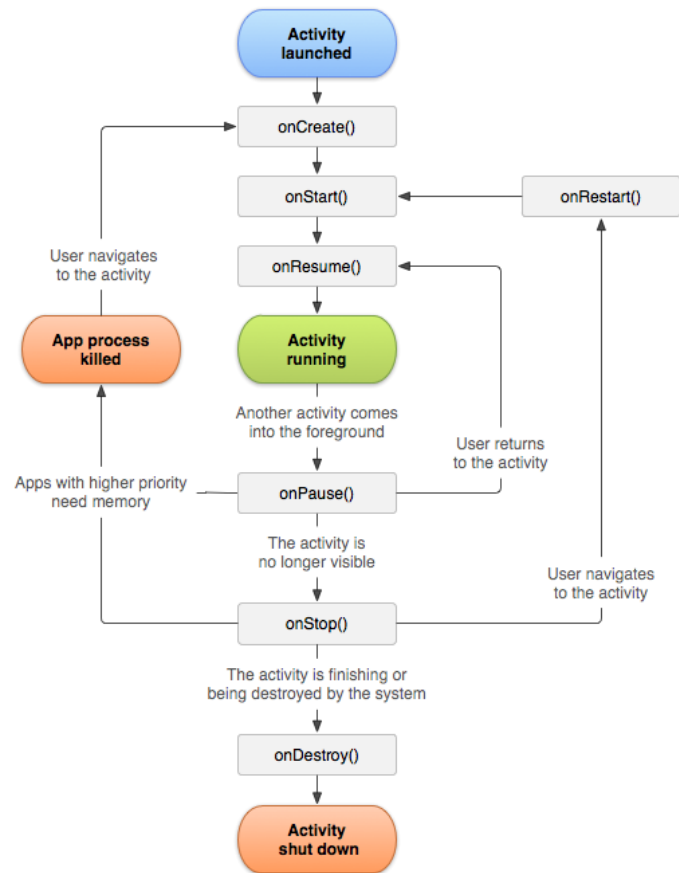
- An Activity represents a single screen with a UI
- Ex: Email Application consists of activities for
 - Showing list of emails
 - Composing an email
 - Reading an email
- Each activity is independent
- Other applications can use a particular activity if the email application gives permission to do so

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?