CS5530
Mobile/Wireless Systems
Android UI

Yanyan Zhuang

Department of Computer Science

http://www.cs.uccs.edu/~yzhuang

cat announce.txt_

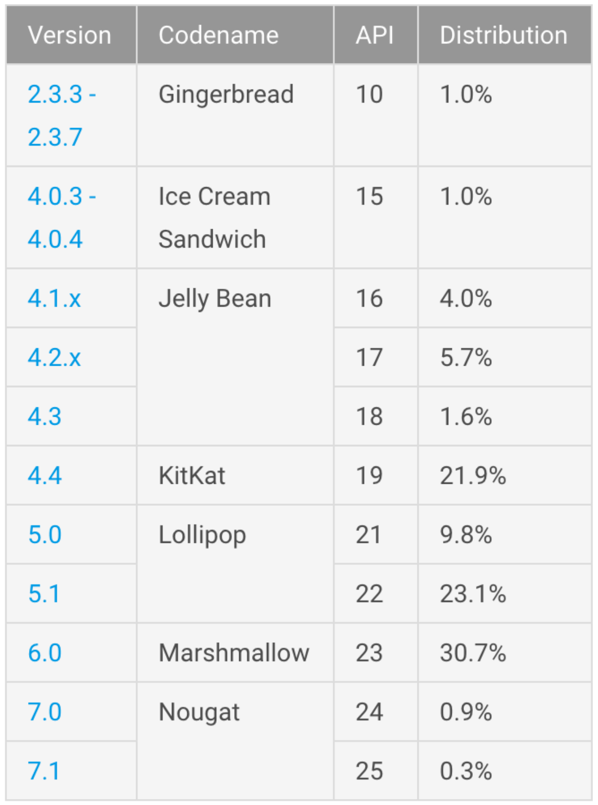
- Assignment 2 will be posted soon
  - Due after midterm
- I will be away next Monday
  - Dr. Chow’s guest lecture
- Midterm date
  - March 20

Android…

- Android
  - A mobile operating system developed by Google
  - Based on Linux kernel and designed primarily for smartphones and tablets
- IDE
  - Android studio https://developer.android.com/studio/index.html
- Android API
  - Java as the programming language

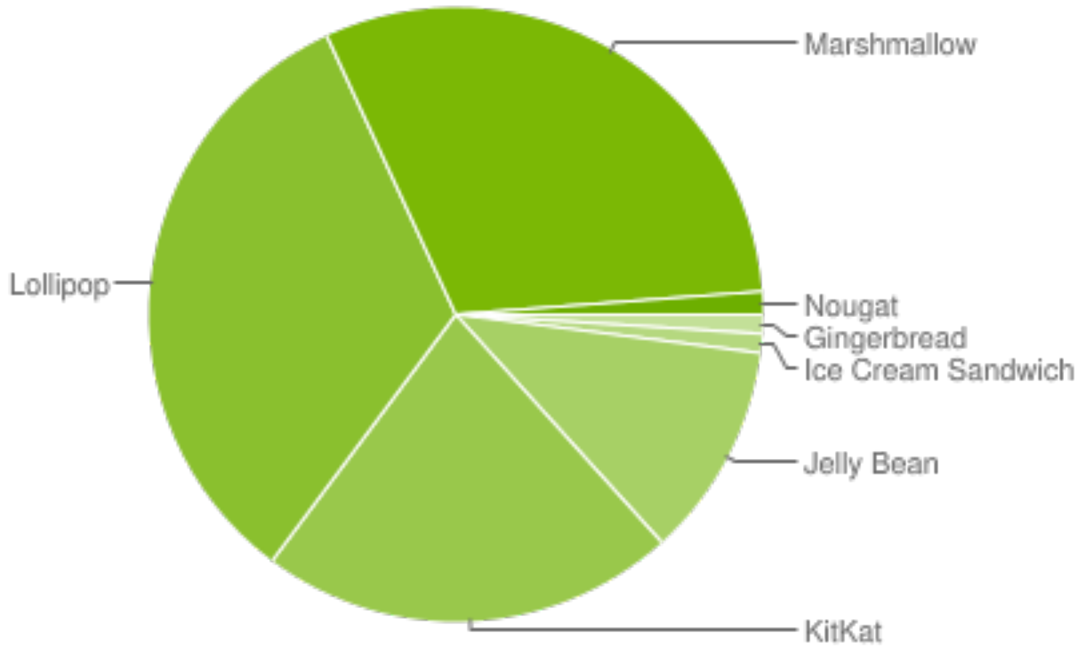
Android…

- A fast evolving OS: Dashboards
  - https://developer.android.com/about/dashboards/index.html



Version	Codename	API	Distribution
2.3.3 - 2.3.7	Gingerbread	10	1.0%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	1.0%
4.1.x	Jelly Bean	16	4.0%
4.2.x		17	5.7%
4.3		18	1.6%
4.4	KitKat	19	21.9%
5.0	Lollipop	21	9.8%
5.1		22	23.1%
6.0	Marshmallow	23	30.7%
7.0	Nougat	24	0.9%
7.1		25	0.3%

ata collected during a 7-day period ending on February 6, 2017.



Android…

- Specify Minimum and Target API Levels
  - AndroidManifest.xml

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android" ... >
  <uses-sdk android:minSdkVersion="4" android:targetSdkVersion="15" />
  ...
</manifest>
```

- Check System Version at Runtime

```
if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.HONEYCOMB) {
    ……
}
```

Running Android Code

- Run code on simulator
- Run code on a real device
  - No license needed
  - On Android 4.2 and newer, Developer options is hidden by default
  - Need to enable **developer option and USB debugging** (Galaxy example): this is all you need to do
    - Go to Settings > More > About Device, scroll down to Build Number
    - Tap it repeatedly (7 times)
    - See the Developer options menu under Settings > check USB debugging

Android Debug Bridge (ADB)

- Android Debug Bridge (adb)
  - Command-line tool to you communicate with a device
  - Installing/debugging apps, and a Unix shell
- A client-server program with three components
  - A **client** runs on development machine
    - Invoke a client by issuing `adb`
  - A **daemon** (adbd) runs commands on a device
    - Runs as a background process on device
  - A **server** manages communication between client and daemon
    - Runs as a background process on development machine

Android Debug Bridge (ADB)

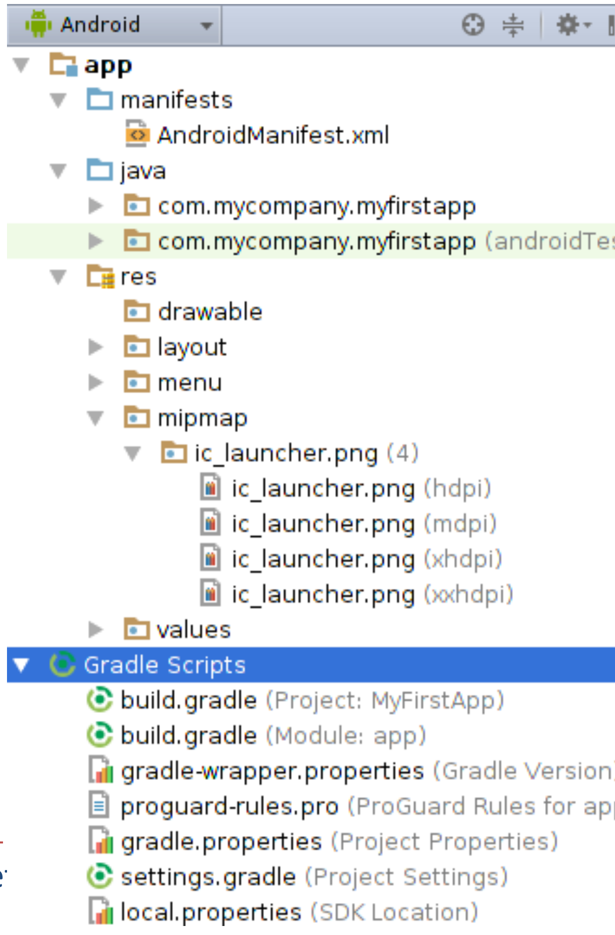
- To install adb (Mac OS example)
  - Install homebrew
    - ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"
  - Install adb
    - brew install android-platform-tools
  - Start adb
    - $ adb devices

      List of devices attached

      07f105740c8cad3f device
    - $ adb shell

Android App Structure

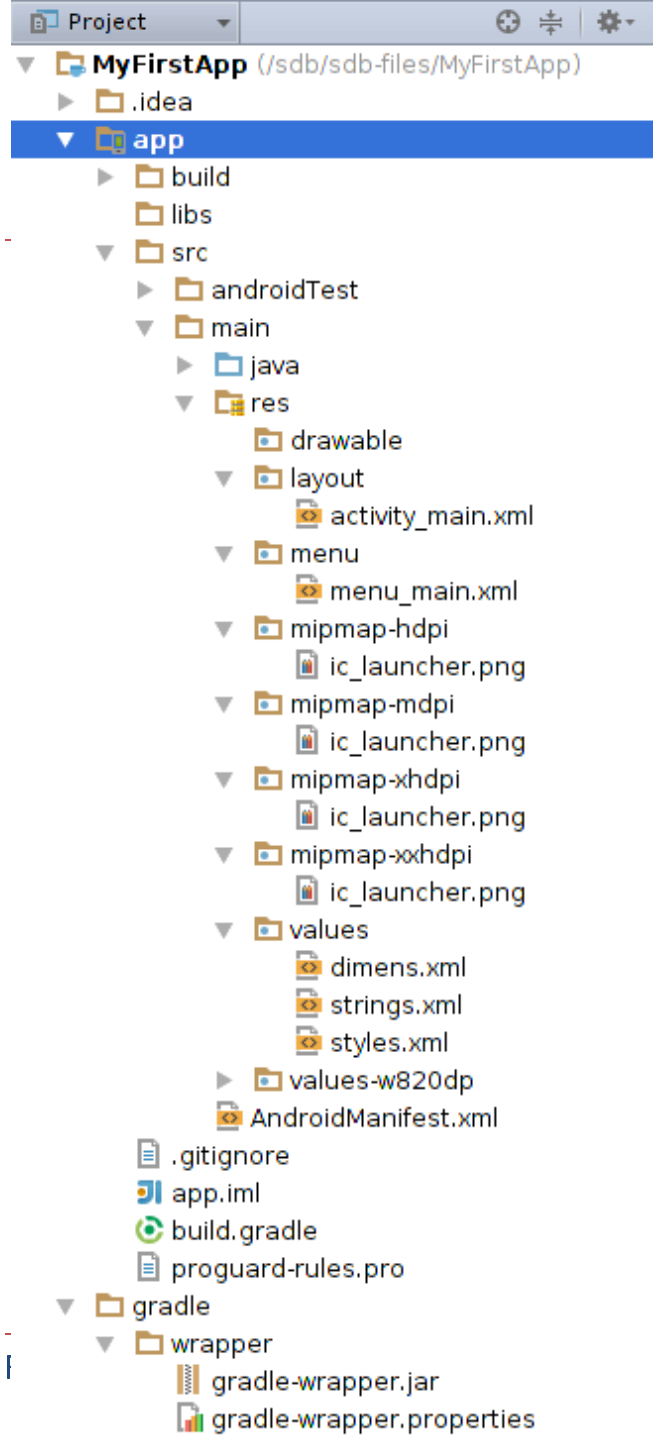
- Project files
  - By default, Android Studio displays files in **Android** view
  - manifests
    - AndroidManifest.xml file
  - java
    - Java source code, separated by package names
  - res
    - All non-code resources
      - XML layouts, UI strings, images



```
Android
▼ app
  ▼ manifests
      AndroidManifest.xml
  ▼ java
    ▶ com.mycompany.myfirstapp
    ▶ com.mycompany.myfirstapp (androidTe
  ▼ res
      drawable
    ▶ layout
    ▶ menu
    ▼ mipmap
      ▼ ic_launcher.png (4)
          ic_launcher.png (hdpi)
          ic_launcher.png (mdpi)
          ic_launcher.png (xhdpi)
          ic_launcher.png (xxhdpi)
    ▶ values
▼ Gradle Scripts
    build.gradle (Project: MyFirstApp)
    build.gradle (Module: app)
    gradle-wrapper.properties (Gradle Version
    proguard-rules.pro (ProGuard Rules for ap
    gradle.properties (Project Properties)
    settings.gradle (Project Settings)
    local.properties (SDK Location)
```

Android App Structure

- Project files
  - Project view
    - Actual file structure of the project
      - Including all files hidden from Android view
  - Looks fairly complex now



```
Project
▼ MyFirstApp (/sdb/sdb-files/MyFirstApp)
  ▶ .idea
  ▼ app
    ▶ build
      libs
    ▼ src
      ▶ androidTest
      ▼ main
        ▶ java
        ▼ res
            drawable
          ▼ layout
              activity_main.xml
          ▼ menu
              menu_main.xml
          ▼ mipmap-hdpi
              ic_launcher.png
          ▼ mipmap-mdpi
              ic_launcher.png
          ▼ mipmap-xhdpi
              ic_launcher.png
          ▼ mipmap-xxhdpi
              ic_launcher.png
          ▼ values
              dimens.xml
              strings.xml
              styles.xml
          ▶ values-w820dp
          AndroidManifest.xml
      .gitignore
      app.iml
      build.gradle
      proguard-rules.pro
  ▼ gradle
    ▼ wrapper
        gradle-wrapper.jar
        gradle-wrapper.properties
```

Create an Android Project

- Start a new Android Studio project, or File → New Project
  - Application Name: "MyFirstApp"
  - Company Domain: "example.com"
- Target Android Devices: keep the default values
  - We will get back to this later
- Add an Activity to Mobile: select Empty Activity
- Customize the Activity: keep default values → Finish
  - Takes a long time to Finish…

Create an Android Project

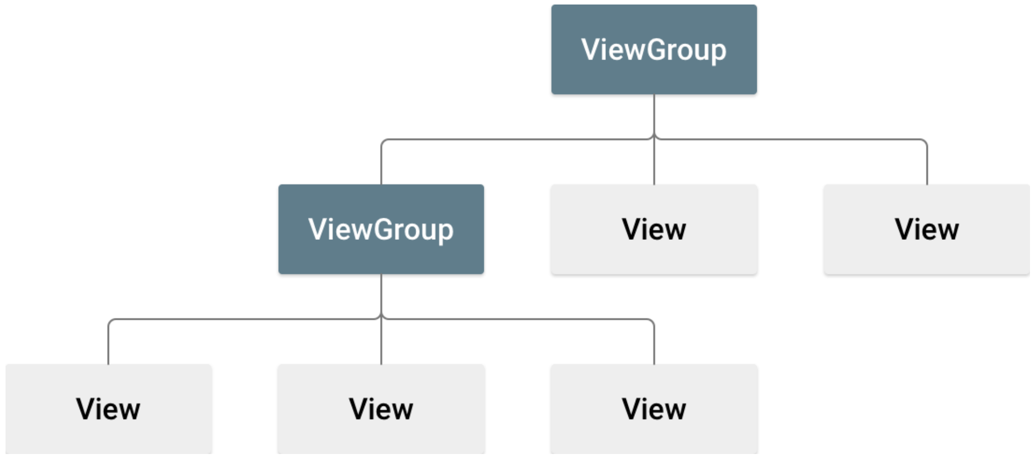
- In Android view
  - app > java > com.example.myfirstapp > MainActivity.java
    - Main activity (entry point for your app)
    - When build and run an app, system launches an instance of this Activity and loads its layout
  - app > res > layout > activity_main.xml
    - Defines the layout for the activity's UI
  - app > manifests > AndroidManifest.xml
    - Describes the characteristics of the app and defines each of its components
  - Gradle Scripts > build.gradle
    - 2 files with this name: one for the project and one for the "app" module
    - Mostly work with module's build.gradle file to configure how the Gradle tools compile and build your app

Running the App

- On a real device
  - Windows may need USB driver for the device
    - https://developer.android.com/studio/run/oem-usb.html
  - Enable USB debugging (earlier)
- On a simulator
  - Create an Android Virtual Device (AVD) definition
    - Tools > Android > AVD Manager
    - Create Virtual Device > Select Hardware
    - System Image > Download (one of the recommended system images)
      - Takes a long time again

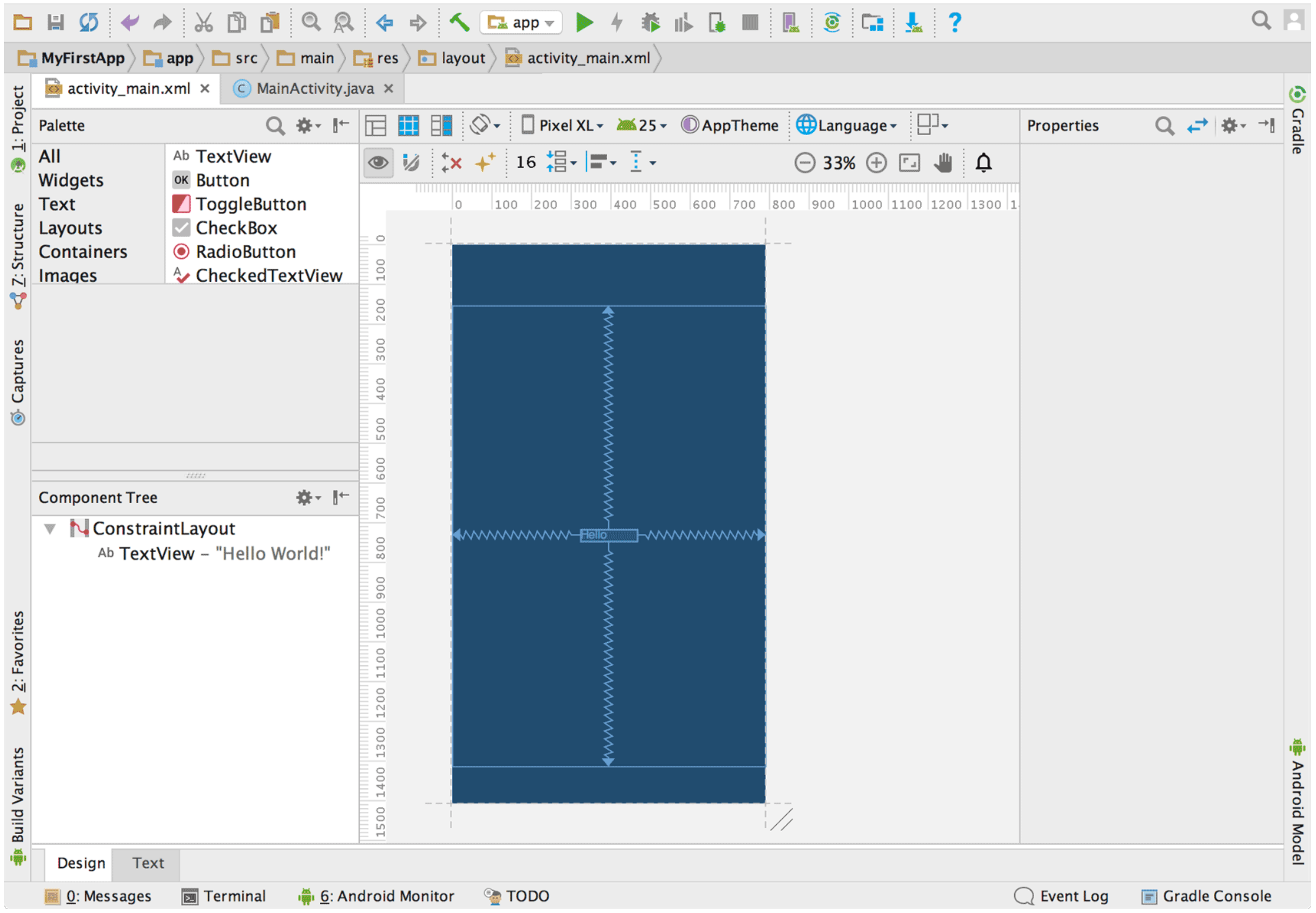
Building Simple User Interface

- UI is built w/ a hierarchy of layouts (ViewGroup objects) & widgets (View objects)
  - Layouts are invisible containers that control how its child views are positioned
  - Widgets are UI components like buttons and text boxes



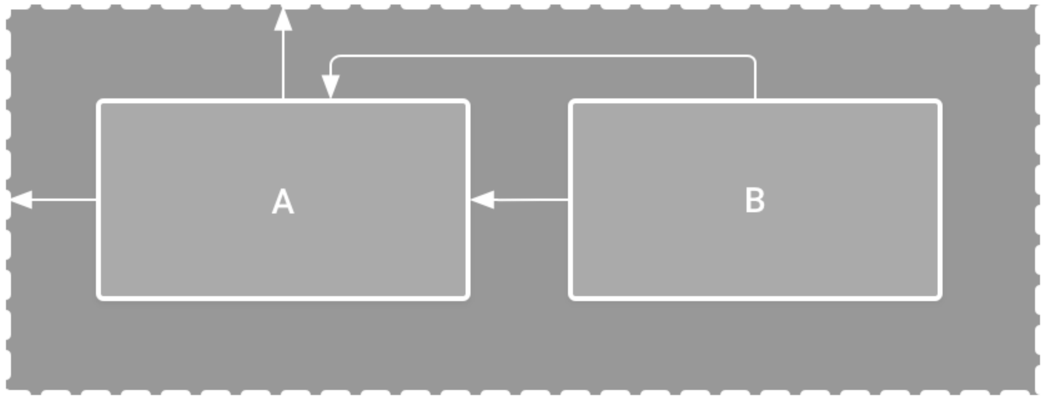
Building Simple User Interface

- Building UI
  - XML
  - Layout Editor



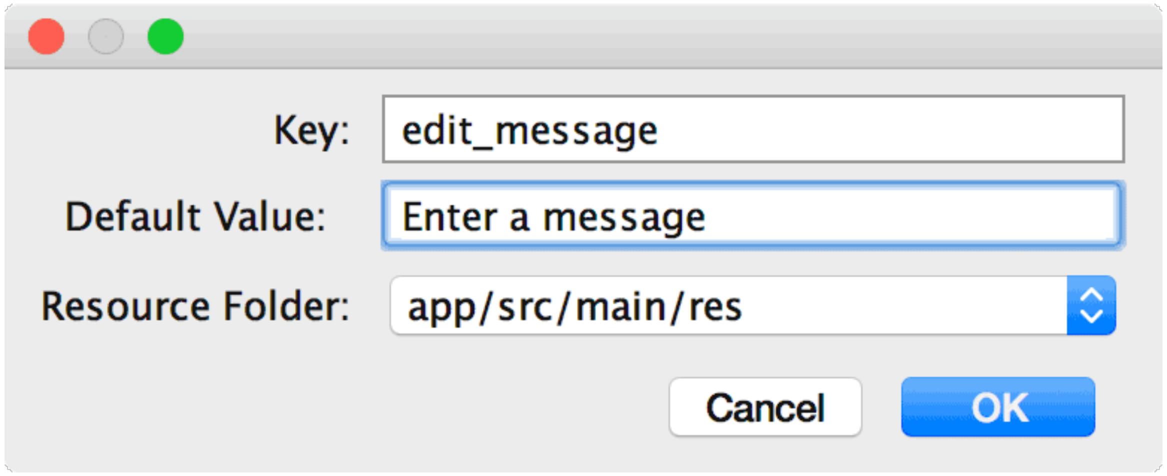
Building Simple User Interface

- Component Tree window
  - Shows the layout's hierarchy of views
- ConstraintLayout
  - A layout that defines the position for each view based on constraints to sibling views and the parent layout



Building Simple User Interface

- Change UI strings
  - res > values > strings.xml



Key: edit_message

Default Value: Enter a message

Resource Folder: app/src/main/res

Start Activity

- Add a method in MainActivity.java that's called by the button
  - Intent
    - An object that provides runtime binding between separate components, such as two activities
    - The Intent represents an app’s "intent to do something"
  - putExtra()
    - An Intent can carry data types as key-value pairs called extras
  - startActivity()

Add up Navigation

- Navigation return to the logical parent screen in app hierarchy
  - Declare which activity is the logical parent in AndroidManifest.xml

```
<activity android:name=".DisplayMessageActivity"
          android:parentActivityName=".MainActivity" >
    <!-- The meta-data tag is required if you support API level 15 and lower -->
    <meta-data
        android:name="android.support.PARENT_ACTIVITY"
        android:value=".MainActivity" />
</activity>
```