Android Week Seven

Yehyun (Joseph) Ryu
UMN App Developers

Two Weeks Left!

- Today
  - Android File Structure
  - How to Use Android Studio
  - XML Syntax
- December 2nd
  - RecyclerView
  - ViewPager and TabLayout
  - Fragments

Android File Structure

Android Manifest

Strings

Layouts

XML

Colors

Images

User Interaction

Activity

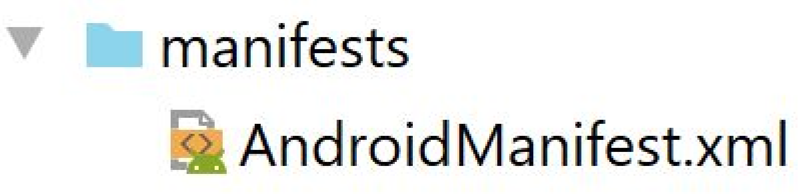
Java

Networking

Data Storage

Gradle

Android Manifest



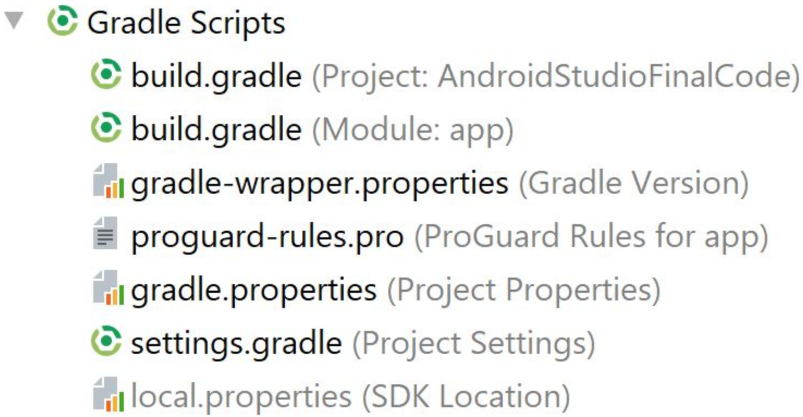
manifests

AndroidManifest.xml

- Every app has an AndroidManifest.xml
- Provides essential information about your app to the Android System
    - Package Name
    - Components
    - Permissions
    - Android API Level
- Documentation
    - https://developer.android.com/guide/topics/manifest/manifest-intro.html

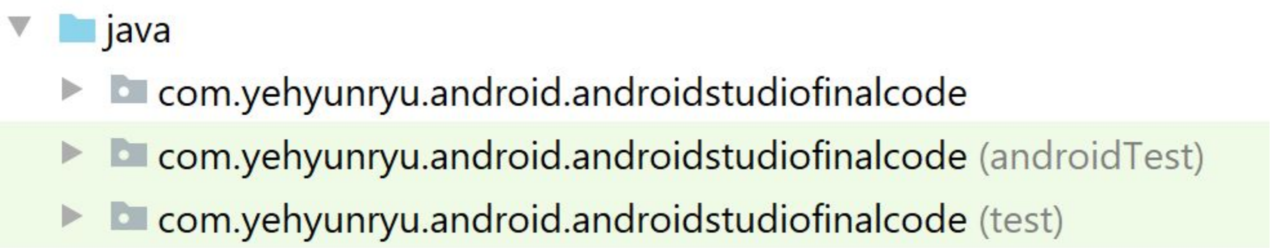
Gradle

- Short Answer:
  - Gradle is a build system.
- Long Answer:
  - 1. Takes source files and 2. Applies appropriate tool and 3. Groups them together into a single compressed file: APK.
  - To automate these tasks, we have to write scripts.
  - Scripts can be written in Java and you can distribute them!
  - Libraries
- Stack Overflow Answer:
  - https://stackoverflow.com/questions/16754643/what-is-gradle-in-android-studio



Gradle Scripts
- build.gradle (Project: AndroidStudioFinalCode)
- build.gradle (Module: app)
- gradle-wrapper.properties (Gradle Version)
- proguard-rules.pro (ProGuard Rules for app)
- gradle.properties (Project Properties)
- settings.gradle (Project Settings)
- local.properties (SDK Location)

Java

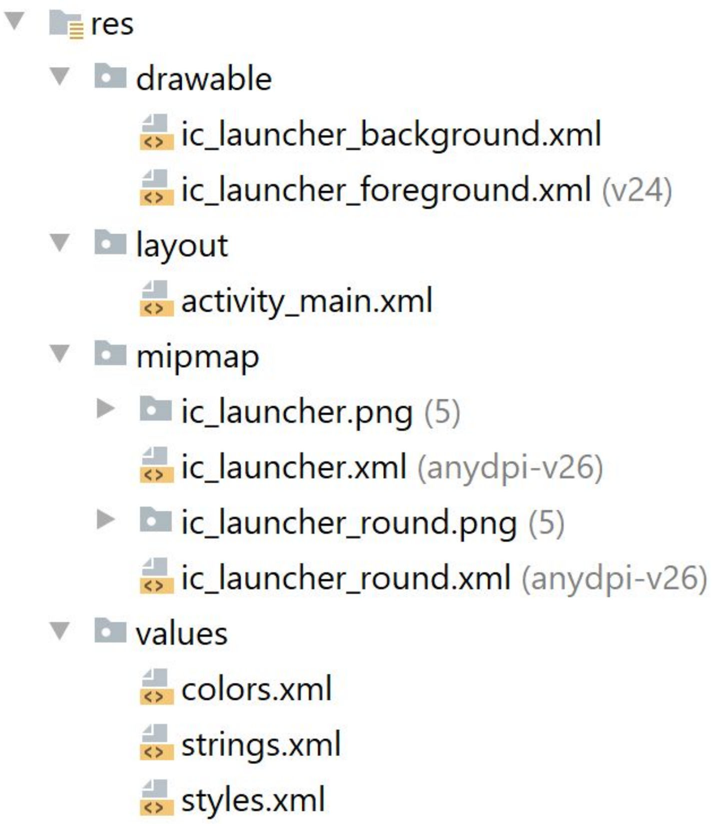


```
▼ java
    ▶ com.yehyunryu.android.androidstudiofinalcode
    ▶ com.yehyunryu.android.androidstudiofinalcode (androidTest)
    ▶ com.yehyunryu.android.androidstudiofinalcode (test)
```

- Handles what is going on behind the scenes.
- User Interaction (swipe, touch)
- Activities and Views
- Other Components
- Networking (connecting to a server)

XML

- Markup Language
  - Describes presentation
- Handles Design Components
  - Layout
  - Images
  - Colors
- What the User Sees
  - Strings



```
▼ res
  ▼ drawable
      ic_launcher_background.xml
      ic_launcher_foreground.xml (v24)
  ▼ layout
      activity_main.xml
  ▼ mipmap
    ▶ ic_launcher.png (5)
      ic_launcher.xml (anydpi-v26)
    ▶ ic_launcher_round.png (5)
      ic_launcher_round.xml (anydpi-v26)
  ▼ values
      colors.xml
      strings.xml
      styles.xml
```

Get Comfortable with Android Studio!

Android Components

Activity

Broadcast Receiver

Content Provider

Service

Activity

- Full-screen windows
- Something user can interact with
- Documentation
    - https://developer.android.com/reference/android/app/Activity.html

Content Provider

- Manager for data
- Handles sharing and storing data
- Communicates with other apps and gives them access to the app’s data if allowed
- Documentation
  - https://developer.android.com/reference/android/content/ContentProvider.html

Service

- Long running operations that run in the background
- No user interface
- ex) Periodic call to a server to retrieve data
- Documentation
  - https://developer.android.com/guide/components/services.html

Broadcast Receiver

- Android apps send or receive broadcast messages
- Broadcast messages are sent when a particular event of interest occurred
- Broadcast Receivers listen for these events and perform a certain action
- Documentation
    - https://developer.android.com/guide/components/broadcasts.html

XML Syntax (General)

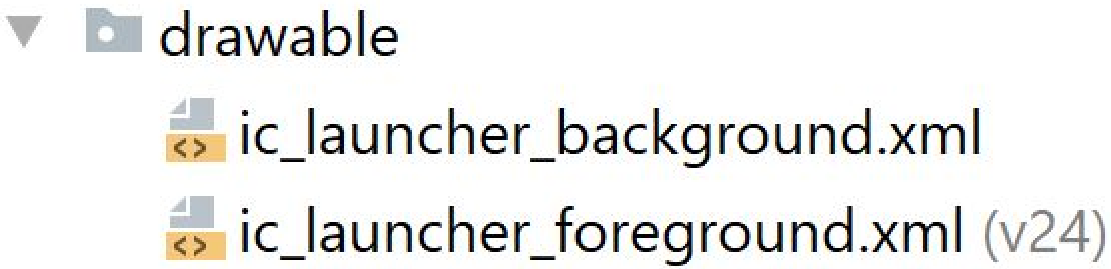
```
<TypeName
 attributes=”values”>
  . . .
</TypeName>
```

XML Syntax (General)

```
<TypeName
  attributes=”values”/>
```

Drawable (XML)

- “Something that can be drawn”
- Type of resources retrieved to draw things to the screen
  - Images
  - Shapes
  - Actions
- Documentation
  - https://developer.android.com/reference/android/graphics/drawable/Drawable.html



drawable
ic_launcher_background.xml
ic_launcher_foreground.xml (v24)

Other Resources (XML)

- layout
- mipmap
    - Launcher Icons
- values
    - colors
    - strings
        - texts
    - Styles
        - chunk of attributes grouped together
        - Multiple TextView that should look the same -> use a style

XML Syntax (Values)

```
<resources>

 <valueType name=”valueName”> value </valueType>

</resources>
```

To Access: @resourcesType/resource_name
ex) @drawable/android_image

XML Syntax (Layout)

Start Tag and Close Tag

```
<ViewType
    attributes>

    Children Views

</ViewType>
```

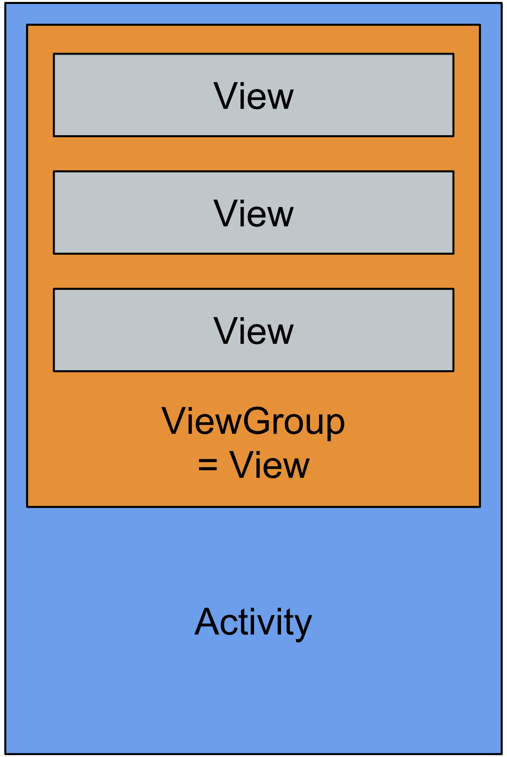
Self Closing

```
<ViewType
    attributes />
```

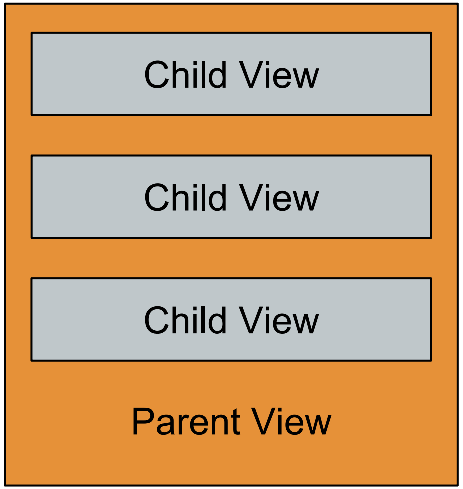
Activity, ViewGroup, Views

- Activity is where all user interaction will occur.
- Activity will always have a layout
- Views are the basic building blocks of a layout
- Views that contains other views are called **ViewGroups**

Activity, View, ViewGroup



Parent View, Child View



Child View

Child View

Child View

Parent View

Views We Learned

- TextView
- ImageView
- Button
- EditText
- LinearLayout
- RelativeLayout
- ConstraintLayout

ConstraintLayout

- ViewGroup that uses constraints (restrictions) to position its children views
- Constraints in position (top, right, bottom, left)
- Uses the app namespace
    - Need to add that in the constraint layout
- Uses app attributes to set constraints
- Might need to add this line in build.gradle (Module: app) in the dependencies block
    - compile 'com.android.support.constraint:constraint-layout:1.0.2'