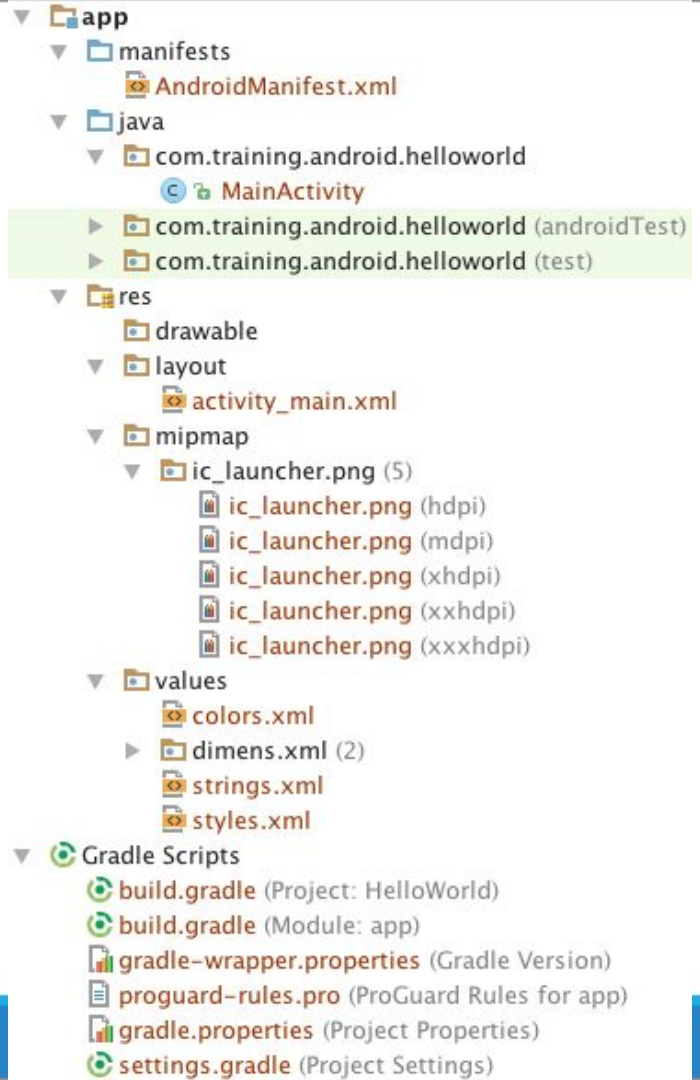
Android Project Structure

Application Fundamentals

Goals

- To identify the files and resources that makes up an Android application.
- To know what the Gradle Build file is.
- To understand how components are being glued together by the manifest.

Hello World


app
manifests
AndroidManifest.xml
java
com.training.android.helloworld
MainActivity
com.training.android.helloworld (androidTest)
com.training.android.helloworld (test)
res
drawable
layout
activity_main.xml
mipmap
ic_launcher.png (5)
ic_launcher.png (hdpi)
ic_launcher.png (mdpi)
ic_launcher.png (xhdpi)
ic_launcher.png (xxhdpi)
ic_launcher.png (xxxhdpi)
values
colors.xml
dimens.xml (2)
strings.xml
styles.xml
Gradle Scripts
build.gradle (Project: HelloWorld)
build.gradle (Module: app)
gradle-wrapper.properties (Gradle Version)
proguard-rules.pro (ProGuard Rules for app)
gradle.properties (Project Properties)
settings.gradle (Project Settings)

app/src/main/java

- This contains the Java source files associated with your project. For example, the Activity class called MainActivity.java is stored in this directory under the package name you specified in the Android project wizard. This MainActivity class provides all the application code associated with the Hello World app.

app/src/main/res/

- “res” stands for resources.
- All the resources needed by the app is placed there.
- The following are the vital resources in Android:
  - Drawables and mipmaps
  - Layouts
  - Strings
  - Dimensions
  - Assets
  - etc...

app/src/main/res/layout/

- This is the place for XML layout files. Layout files are XML files which define how various Android objects (such as textboxes, buttons, etc.) are organized on the screen.

app/src/main/res/drawable-<density>/

- Here we store the various graphic files. We can see various types of drawable folders. This is because there are many Android devices with different screen densities. By default, there are several versions of this folder such as: *drawable-mdpi*, *drawable-hdpi*, and so forth. This is required in order to adapt to different screen resolutions.

app/src/main/res/mipmap-<density>/

- The mipmap folders are for placing your app/launcher icons (which are shown on the homescreen) in only. Any other drawable assets you use should be placed in the relevant drawable folders as before.

app/src/main/res/menu/

- Directory for files that define your app's menu items.

app/src/main/res/values

- Directory for other XML files that contain a collection of resources, such as string and color definitions.

app/src/main/AndroidManifest.xml

- The manifest file describes the fundamental characteristics of the app and defines each of its components.

app/build.gradle

- Android Studio uses Gradle to compile and build your app. There is a `build.gradle` file for each module of your project, as well as a `build.gradle` file for the entire project. Usually, you're only interested in the `build.gradle` file for the module, in this case the `app` or application module. This is where your app's build dependencies are set, including the `defaultConfig` settings:

app/build.gradle (cont.)

- compiledSdkVersion is the platform version against which you will compile your app. By default, this is set to the latest version of Android available in your SDK. (It should be Android 4.1 or greater; if you don't have such a version available, you must install one using the SDK Manager.) You can still build your app to support older versions, but setting this to the latest version allows you to enable new features and optimize your app for a great user experience on the latest devices.

app/build.gradle (cont.)

- `applicationId` is the fully qualified package name for your application that you specified during the New Project workflow.

app/build.gradle (cont.)

- `minSdkVersion` is the Minimum SDK version you specified during the New Project workflow. This is the earliest version of the Android SDK that your app supports.

app/build.gradle (cont.)

- targetSdkVersion indicates the highest version of Android with which you have tested your application. As new versions of Android become available, you should test your app on the new version and update this value to match the latest API level and thereby take advantage of new platform features. For more information, read Supporting Different Platform Versions.

See Building Your Project with Gradle for more information about Gradle.

References

http://developer.android.com/training/basics/firstapp/creating-project.html

End of Slide