

Android 应用软件设计

E1: Hello World

学号: SA18225023

姓名: 常承启

2018 年 9 月 29 日星期六

一、概述

作业的总体目标是实现一款在 Android 平台上运行的订餐 APP, 第一次作业内容主要是开发环境的搭建, 创建应用并实现一个展示图片的功能并打包应用签名。

1、搭建 android Studio 开发环境

安装 Android Studio(以下简称 AS)、JDK、android SDK, 主要完成开发环境的搭建, 本人使用的 Mac, 故安装的时候没有进行环境变量的配置。

2、新建一个应用, 按照要求设置包名以及入口程序, 并实现一个图片展示功能

该任务的主要目的是熟悉 AS 的使用, 学会建立新的工程并设置相应的包名和入口程序。

Android 应用入口 activity 的设置方法有两种: 一种是在新建工程的时候便设置好, 另一种是在 manifest 里的 activity 标签来设置, 本文将阐述两种设置方法。

图片展示则通过 ImageView 标签来实现

3、打包应用签名

这里使用创建 key, 并签名生成 release 版本的 app, 并在平板电脑上测试成功

二、假设

作业的总体目标是实现一款在 Android 平台上运行的订餐 APP, 第一次作业内容主要是开发环境的搭建, 创建应用并实现一个展示图片的功能并打包应用签名。

三、具体实现

1、GitHub

2、APK:

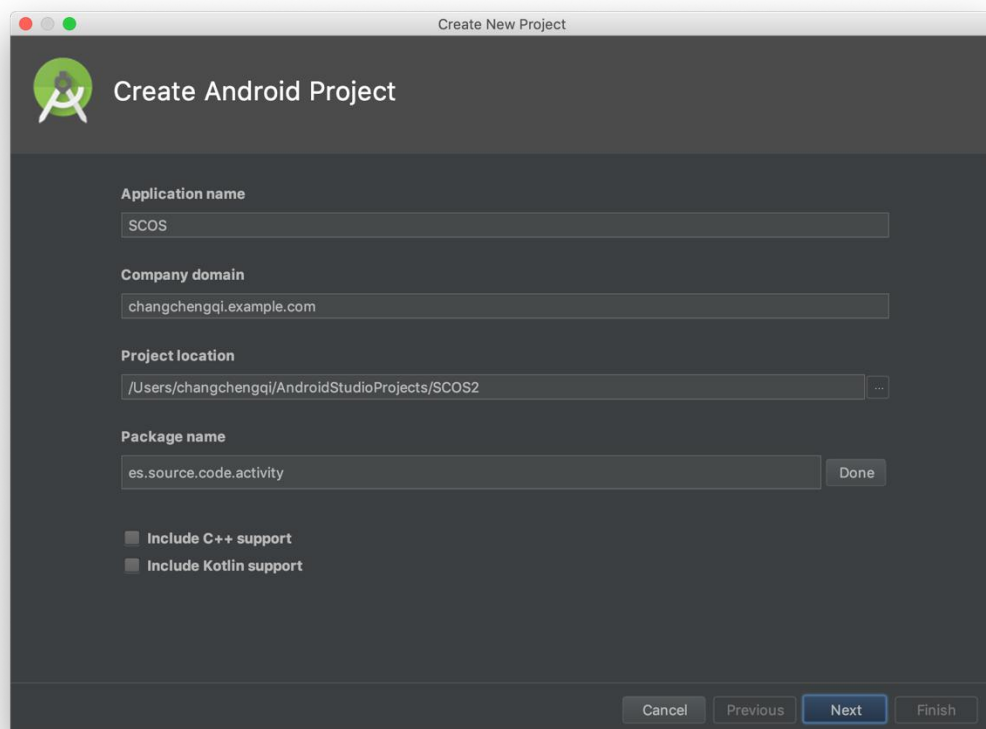
参见 GitHub

3、搭建 android studio 开发环境

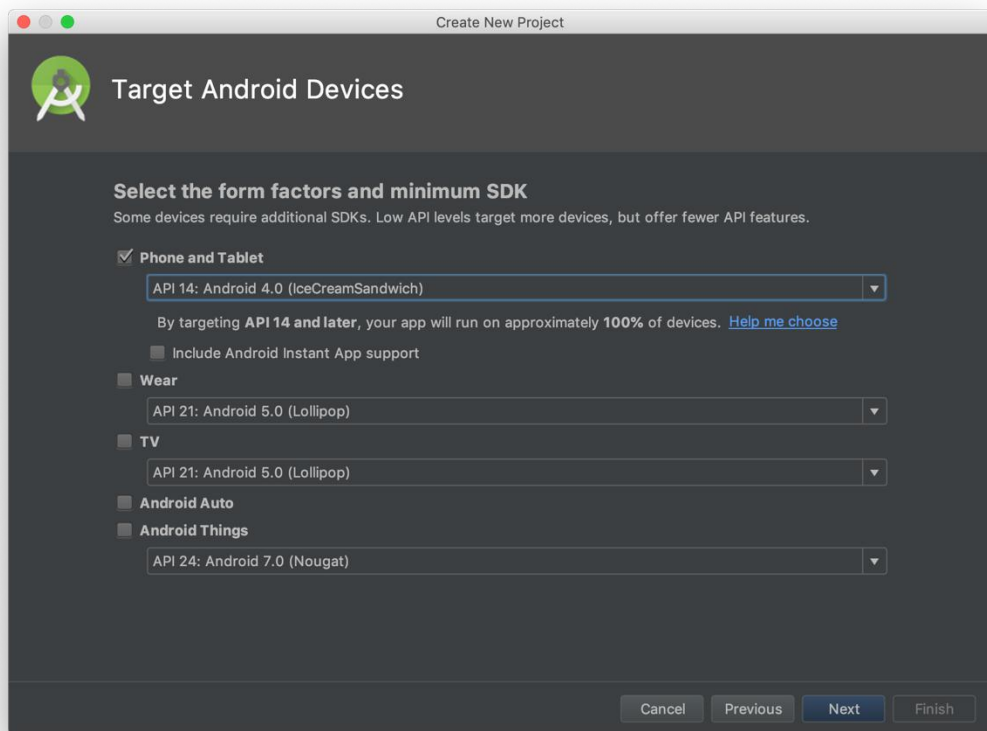
从 android 官网上下载 android studio, 在 oracle 官网上下载 JDK, 并打开 AS 下载所需管理工具, 下载所需 SDK。[1]

4、按照要求创建一个新应用并设置包名和入口程序

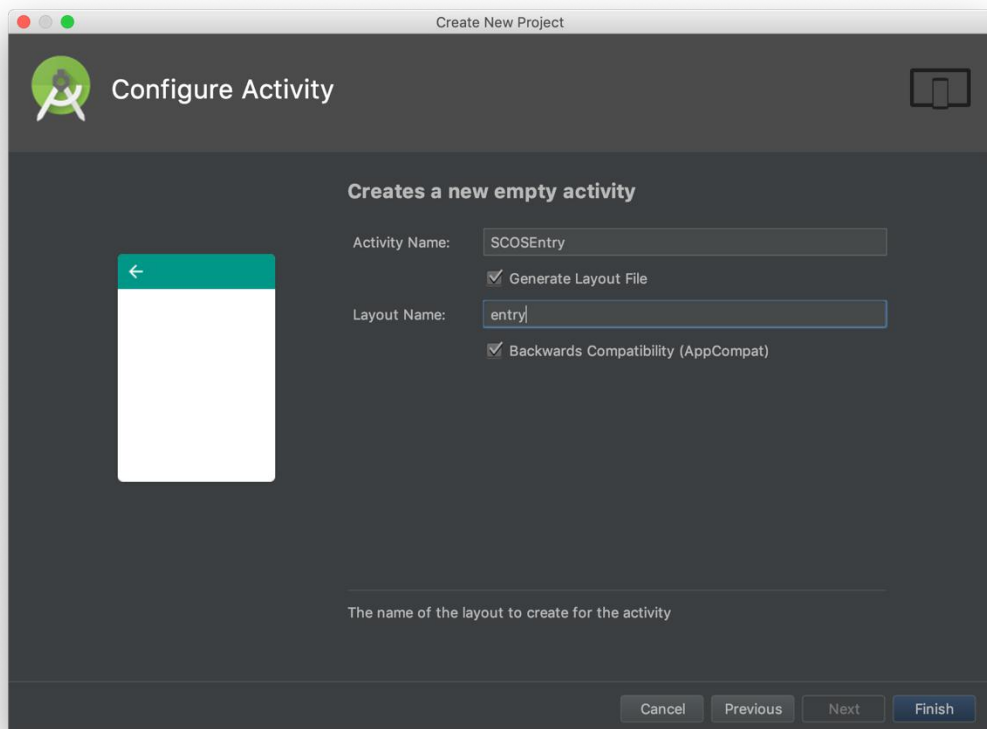
打开 AS, 出现欢迎界面, 选择新建工程, 设置应用名称为 SCOS, 公司域, 包名, 这里为了方便起见, 这里直接在新建工程的时候设置包名, 不然之后设定包名会比较麻烦, 可参考网站[3]来知晓如何在创建应用后修改包名。



之后选择最低的兼容版本, 这里选择 Android 4.0, 可以确保适配市面上几乎所有的设备。下面的 Wear、TV 等因为还未在国内普及, 故不做选择[4],



之后选择一个空的 activity，同样为了方便，可以直接设置 Activity Name 为 SCOSEntry，即直接把 SCOSEntry 设为了入口 activity，并将 layout 设为 entry。

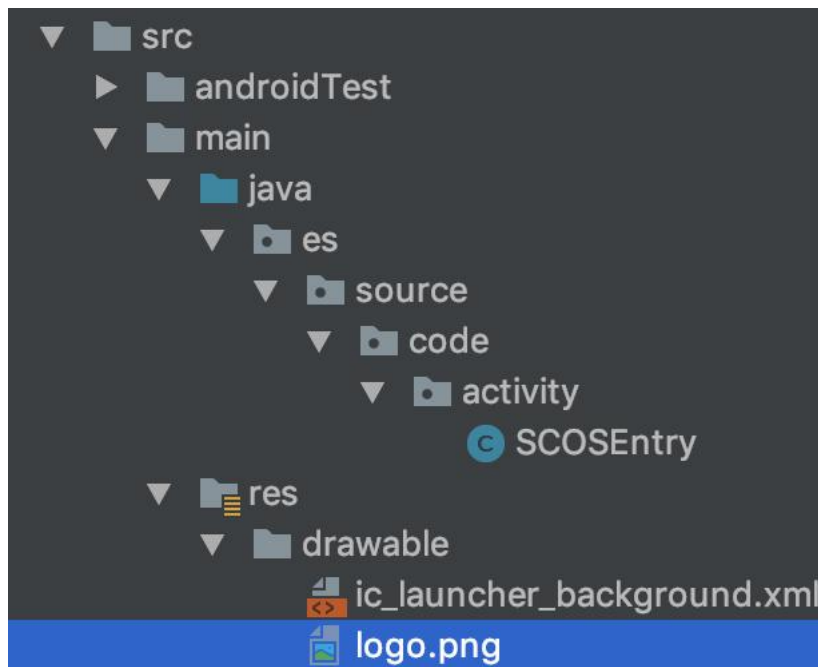


另外在下图中我们也给出了在 AndroidManifest.xml 文件中设置入口程序的方法，将 intent-filter 标签（包含 action 和 category 两句）放入需要设置为入口的 activity 对应的标签中即可。action 节点中的”android.intent.action.MAIN”表明它所在的 Activity 是整个应用程序的入口点。而 category 中的”android.intent.category.LAUNCHER”意思是把这个 Activity 归属到加载器类，即把这个 Activity 标注为会自动加载和启动的 Activity,这样程序启动时候就先加载这个 Activity 了。[5]

```
<activity android:name=".SCOSEEntry">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
```

5、添加图片展示

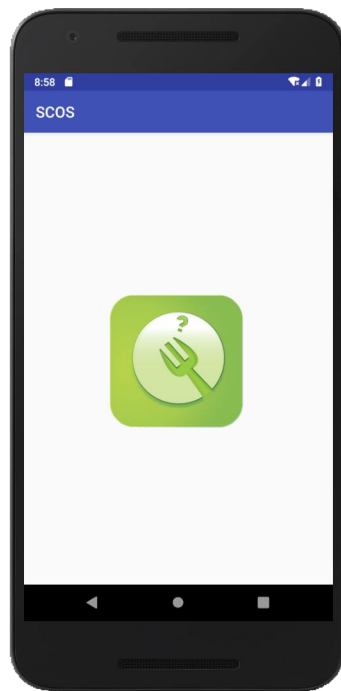
首先我们在 src/res/drawable/文件夹中放入所需展示的图片，这里展示的图片名称为 logo.png。



之后我们在 entry.xml 文件中设置使用 RelativeLayout 相对布局，将原来的 TextView 的文本展示标签替换为用于展示图像的 ImageView 标签,按照惯例在其中设置 id 以方便在 java 中引用，设定 layout_width 和 layout_height 为”wrap_content”即自适应内容大小。设置 centerInParent 属性实现水平垂直居中，并设置 src 属性来完成图片资源的设定。最后设置 padding 和 margin 来把图片缩放到合适的大小。

```
<ImageView
    android:id="@+id/logo"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_centerInParent="true"
    android:src="@drawable/logo"
    android:padding="80dp"
    android:layout_margin="20dp"/>
```

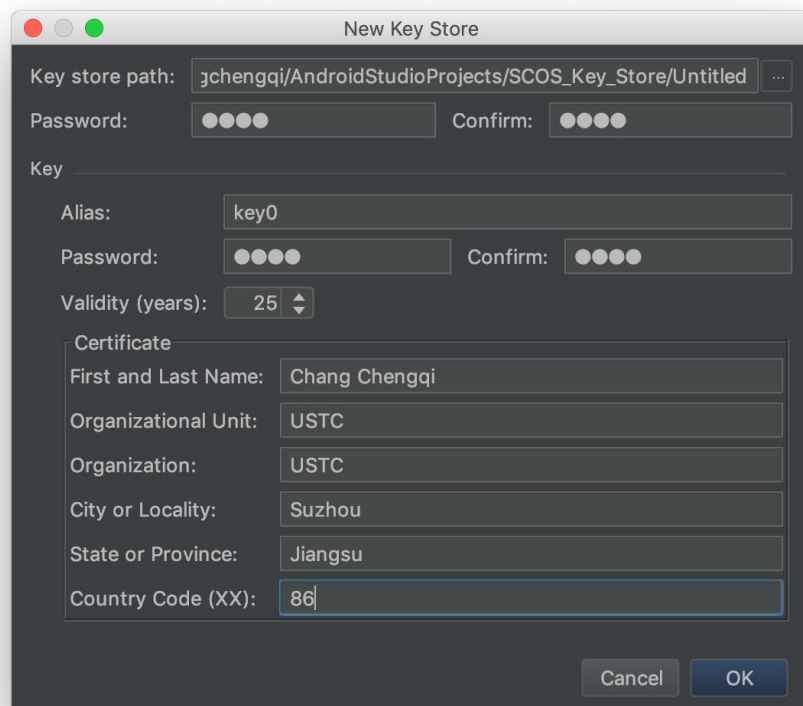
最后展示图片如下图所示



6、数字签名并打包

我们使用 Android Studio 运行或测试我们的 app 它使用一个默认的 debug.keystore 进行签名。生成 APK 文件可以通过使用 AS 来生成，也可以通过 Gradle 来生成。此处使用 AS 创建一个新的 keystore，并发布我们的 APK。

这里根据具体信息填写即可。



New Key Store

Key store path: jchengqi/AndroidStudioProjects/SCOS_Key_Store/Untitled ...

Password: (masked) Confirm: (masked)

Key

Alias: key0

Password: (masked) Confirm: (masked)

Validity (years): 25

Certificate

First and Last Name: Chang Chengqi

Organizational Unit: USTC

Organization: USTC

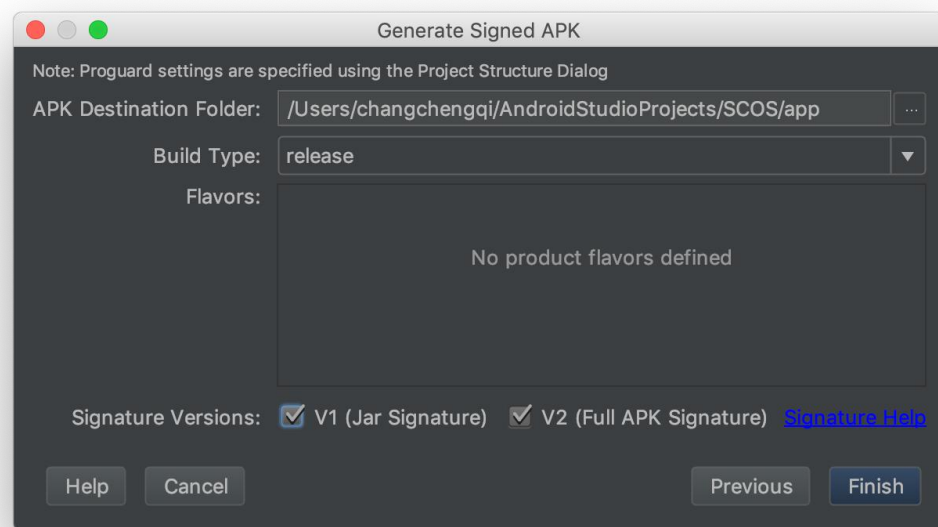
City or Locality: Suzhou

State or Province: Jiangsu

Country Code (XX): 86

Cancel OK

之后需要选择签名的版本，通常 Signature Version 中 Jar Signature 和 Full APK Signatur 都默认勾选



四、结论

本次作业主要是熟悉 android 应用开发环境的搭建，学会在 android studio 下创建工程，熟悉工程文件夹目录、创建应用并利用 ImageView 展示一张 Logo 图片。

由于安卓设备的碎片化现象十分明显，为了适应不同分辨率的机器，图片资源应当放在不同的像素密度文件夹中，以放置对应分辨率缩放图片，就像放置图标的文件夹在应用创建好时就已经像下图一样创建了多个文件夹，用于放置不同分辨率的图标。



另外应当注意的是设置签名 apk 的时候，V1 (Jar Signature) 和 V2 (Full APK Signature) 通常都应当勾选上，不然可能会出现安装应用时权限校验失败的问题。

五、参考文献

- [1] <https://blog.csdn.net/yuanpeng1014/article/details/78188839>
- [2] https://blog.csdn.net/sinat_30949835/article/details/62222351
- [3] <https://blog.csdn.net/myosotis5/article/details/79208707>
- [4] 《第一行代码》
- [5] https://blog.csdn.net/wh_19910525/article/details/7941617
- [6] <https://blog.csdn.net/nimasike/article/details/51457229>