第 1 章　Android 的基础知识

Android 的英文原意是机器人，Android 图标也是一个机器人（如图 1-1 所示），来源于法国作家利尔·亚当在 1886 年发表的科幻小说《未来夏娃》，将外表像人的机器命名为 Android。学习 Android 先从了解 Android 的历史开始，Android 的诞生与 Andy Rubin 有关，Andy Rubin 被称为 Android 之父。

图 1-1　Android 图标

1.1　Android 与 Andy Rubin

1989 年，Andy Rubin 被一名苹果公司工程师引荐到当时处在第一个全盛时期的苹果公司，参与名为 Magic Cap 的智能手机操作系统开发工作。

1999 年，Andy Rubin 创立了 Danger 公司，开发了一个名为 Hiptop 的类似智能手机雏形的设备，提出了“智能手机”的概念——“支持互联网”和“其上运行着能够实现不同功能的各种应用”。

2002 年，Andy Rubin 在斯坦福大学做了一次讲座，听众中包括 Google 公司的两位创始人 Larry Page 和 Sergey Brin。互联网智能手机的理念深深打动了 Larry Page，尤其是他注意到 Danger 产品上默认的搜索引擎为 Google。

2003 年，Andy Rubin 等人创立了 Android 公司，并组建 Android 开发团队，注册了 android.com 域名，立志设计一个基于开源思想的移动平台。当时的手机操作系统都是手机厂商单独开发的，操作系统也是各手机厂商的核心技术，具有很强的封闭性。

2005 年，Andy Rubin 靠自己的积蓄和朋友的支持，艰难地完成了 Android 系统。但在寻找投资方时并不顺利，突然 Andy Rubin 想到了 Google 公司的 Larry Page，于是给他发了一封电子邮件。仅仅几周后，Google 公司收购了成立仅 22 个月的 Android 公司。Andy Rubin 成为 Google 公司工程部副总裁，继续负责 Android 项目。

2014 年，Andy Rubin 离开 Google 公司。2015 年，Andy Rubin 创立 Essential 公司，开发 Android 智能手机。两年后发布首款手机 Essential PH-1，但是销量惨淡。2020 年 2 月 12 日，Essential 公司正式宣布停止运营。

1.2　Android 的开发环境

Android 刚发布时，Google 公司只提供了 SDK，并没有提供官方的开发环境。开发者只能通过第三方工具进行开发，如 Eclipse 和 IntelliJ IDEA。直到 2013 年，Google 公司发

电子工业出版社版权所有 盗版必究

布了基于 IntelliJ IDEA 的 Android 集成开发工具——Android Studio，支持 Windows（32 位和 64 位）、Mac 和 Linux 操作系统。

1.2.1 Android Studio 的下载

截至 2020 年 3 月 8 日，Android Studio 的最新版本是 3.5.3，下载网站地址是 https://developer.android.google.cn/studio/。打开网站，单击“DOWNLOAD OPTIONS”链接，跳转至 Android Studio 下载列表（如图 1-2 所示）。

Platform	Android Studio package	Size
Windows (64-bit)	android-studio-ide-192.6241897-windows.exe Recommended	749 MB
	android-studio-ide-192.6241897-windows.zip No .exe installer	752 MB
Windows (32-bit)	android-studio-ide-192.6241897-windows32.zip No .exe installer	751 MB
Mac (64-bit)	android-studio-ide-192.6241897-mac.dmg	762 MB
Linux (64-bit)	android-studio-ide-192.6241897-linux.tar.gz	766 MB
Chrome OS	android-studio-ide-192.6241897-cros.deb	647 MB

图 1-2 Android Studio 下载列表

1.2.2 Android Studio 的安装

1. Windows 版的安装

在 Windows 操作系统中，双击下载的 exe 格式安装文件，根据安装向导的提示安装 Android Studio 和所需的 SDK 工具（如图 1-3 所示）。安装进度完成后，单击“Finish”按钮（如图 1-4 所示）。

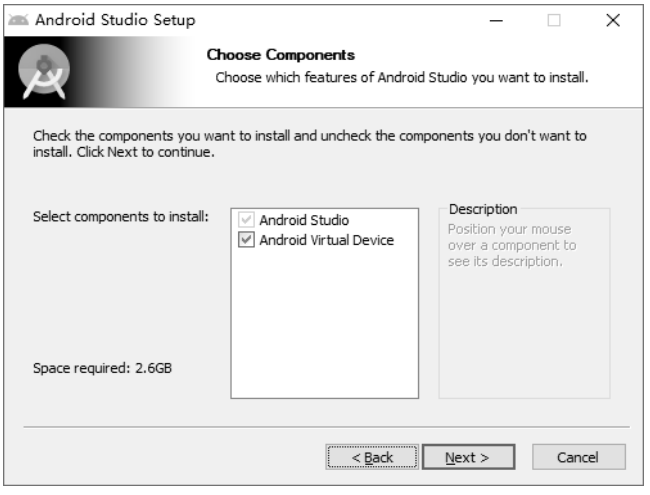

图 1-3 Android Studio 安装向导

电子工业出版社版权所有 盗版必究

图 1-4　Android Studio 完成安装

2．Mac 版的安装

在 Mac 操作系统中，双击下载的 dmg 格式安装文件，在安装界面中将 Android Studio.app 拖曳到 Applications 文件夹中（如图 1-5 所示），即可安装完成。

图 1-5　拖曳安装

1.2.3　Android SDK 的安装

由于目前国内无法直接访问 Android 官方网站的部分地址，在 Windows 操作系统中首次运行时会打开“Android Studio First Run”对话框（如图 1-6 所示）。

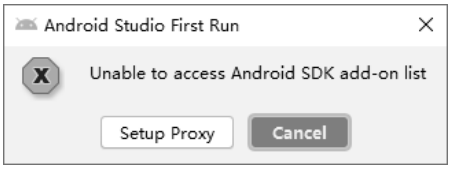

图 1-6　“Android Studio First Run”对话框

电子工业出版社版权所有

单击“Setup Proxy”按钮，设置代理地址为“mirrors.neusoft.edu.cn”（如图 1-7 所示），单击“OK”按钮完成设置。

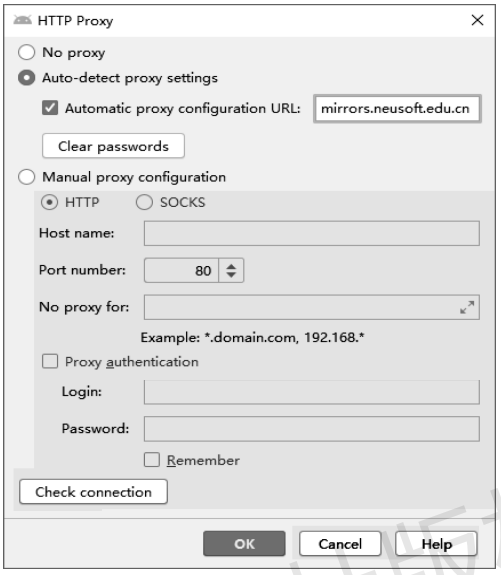

图 1-7 “HTTP Proxy”对话框

提示：HTTP Proxy

当无法直接访问 Android 官方地址进行更新时，需要设置代理地址，否则无法进行自动更新和下载。可以使用东北大学的代理地址：mirrors.neusoft.edu.cn。

在“Android Studio Setup Wizard”对话框中，选择下载的 SDK 程序和保存路径（如图 1-8 所示），单击“Next”按钮进行安装，完成安装后单击“Finish”按钮。

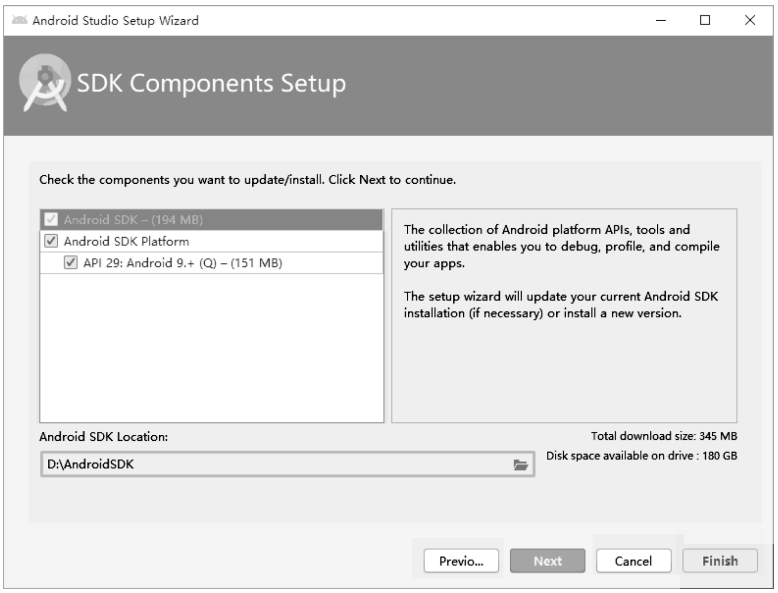

图 1-8 Android SDK 下载

电子工业出版社版权所有 盗版必究

提示：SDK 占用磁盘空间

Android SDK 占用磁盘空间非常大，至少要准备 10GB 磁盘空间。如果系统盘空间较小，建议安装在其他盘中。

1.2.4　Android Studio 界面

Android Studio 界面主要由菜单栏、工具栏、导航条、左侧工具条、工具窗口、编辑器、右侧工具条、运行工具窗口、状态栏等组成。Mac 版的界面（如图 1-9 所示）与 Windows 版的界面稍有不同。

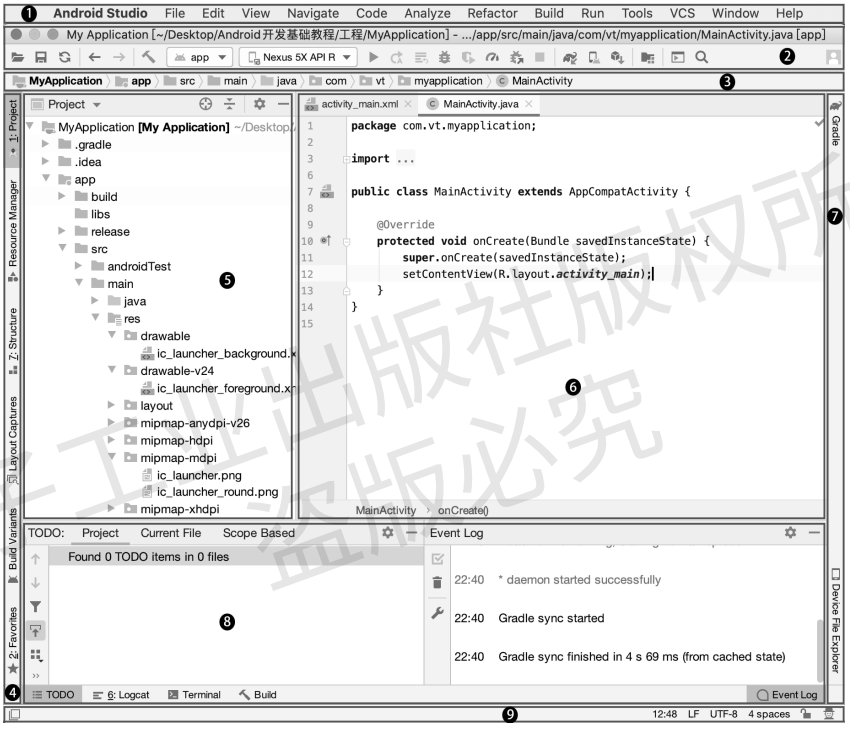

图 1-9　Mac 版的 Android Studio 界面

1．菜单栏

菜单栏包含文件（File）、编辑（Edit）、视图（View）、导航（Navigate）、代码（Code）、分析（Analyze）、重构（Refactor）、构建（Build）、运行（Run）、工具（Tools）、版本控制系统（VCS）、窗口（Window）、帮助（Help）等功能菜单（如图 1-10 所示）。

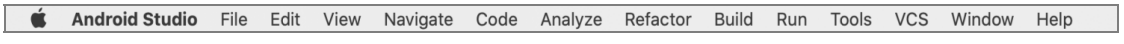

图 1-10　菜单栏

2．工具栏

工具栏包含从菜单栏中提取出来的一些常用功能（如图 1-11 所示），能够进行快速操作，提高效率。

电子工业出版社版权所有 盗版必究

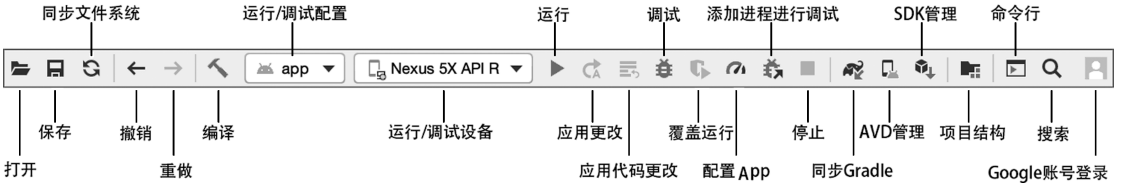

图 1-11　工具栏

3．导航条

导航条用来辅助查看打开的项目和文件（如图 1-12 所示），单击文件夹可以快速选择子文件夹或文件。

图 1-12　导航条

4．左侧工具条

左侧工具条用来放置窗口的切换按钮（如图 1-13 所示），包含“Project”“Resource Manager”“Structure”“Layout Captures”“Build Variants”和“Favorites”窗口。

5．工具窗口

工具窗口最多可以同时显示两个窗口（如图 1-14 所示），单击左侧工具条中的按钮可以进行切换。上面可以显示“Project”或“Resource Manager”窗口，下面可以显示其余的任意一个窗口。

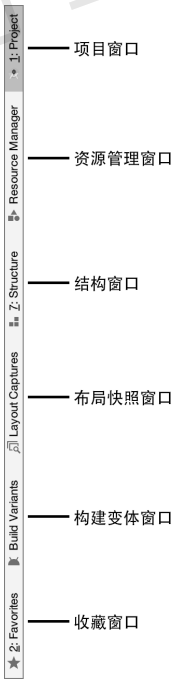

图 1-13　左侧工具条

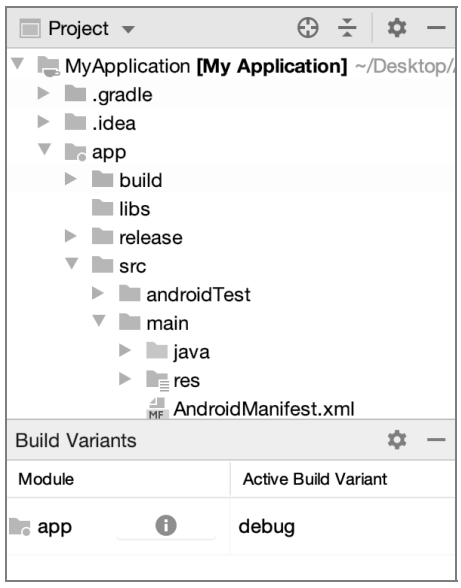

图 1-14　工具窗口

电子工业出版社版权所有 盗版必究

6．编辑器

编辑器由文件标签栏、左边栏、编辑区和代码定位栏组成（如图 1-15 所示），是编辑配置信息、编写代码和调试断点设置的区域。

7．右侧工具条

右侧工具条包含“Gradle”窗口和设备文件浏览器（外接 USB 设备或虚拟设备），单击后在编辑区右侧显示（如图 1-16 所示）。

图 1-15　编辑器

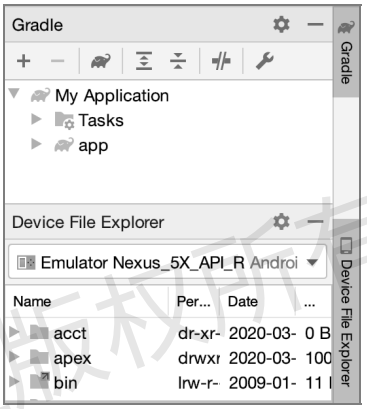

图 1-16　右侧工具条

8．运行工具窗口

运行工具窗口主要显示 Android Studio 的运行过程（如图 1-17 所示），左侧显示“Run”“TODO”“Profiler”“Logcat”“Terminal”或“Build”窗口，右侧显示“Event Log”窗口。

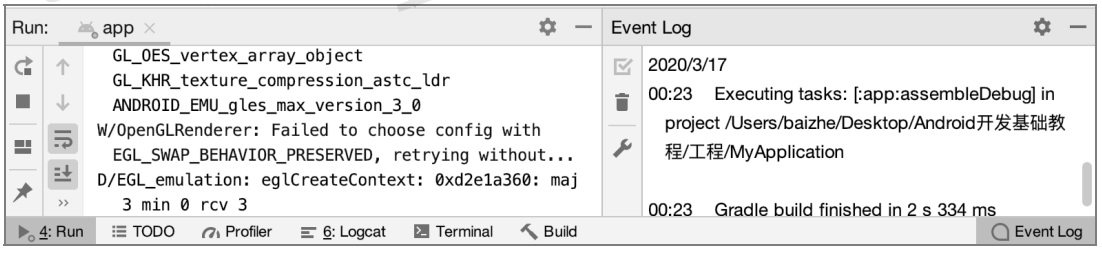

图 1-17　运行工具窗口

9．状态栏

状态栏通常在界面的底部，主要显示 Android Studio 当前的状态和执行的任务（如图 1-18 所示）。

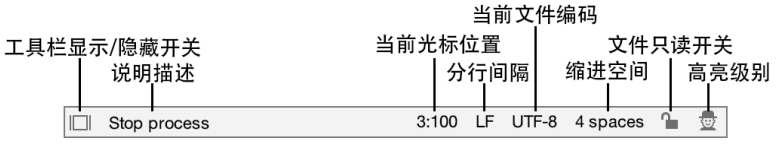

图 1-18　状态栏

电子工业出版社版权所有 盗版必究

1.2.5 Gradle 更新

Gradle 是 Android Studio 默认的 App 构建工具，根据构建规则和配置文件自动构建 App，自动构建包括编译、打包等流程。安装 Android Studio 后，还要注意 Gradle 的更新，Gradle 自动更新时并不通过 Android 官网下载更新文件，而是通过 Gradle 网址（http://services.gradle.org/distributions/）。在国内下载速度较慢甚至无法下载，此时可通过手动下载的方式进行更新。

1．Windows 版的手动更新

将下载的新版本 Gradle 压缩包解压到 gradle 文件夹中（如图 1-19 所示）。

图 1-19　更新版本所在的文件夹

在 Android Studio 中，选择【File】→【Settings】命令（如图 1-20 所示）。在打开的对话框中，选择左侧的“Gradle”选项，然后选择右侧的“Use local gradle distribution”单选按钮，单击“Gradle home”后面的路径选择按钮，选择刚才解压的 gradle 文件夹，单击“OK”按钮（如图 1-21 所示）。

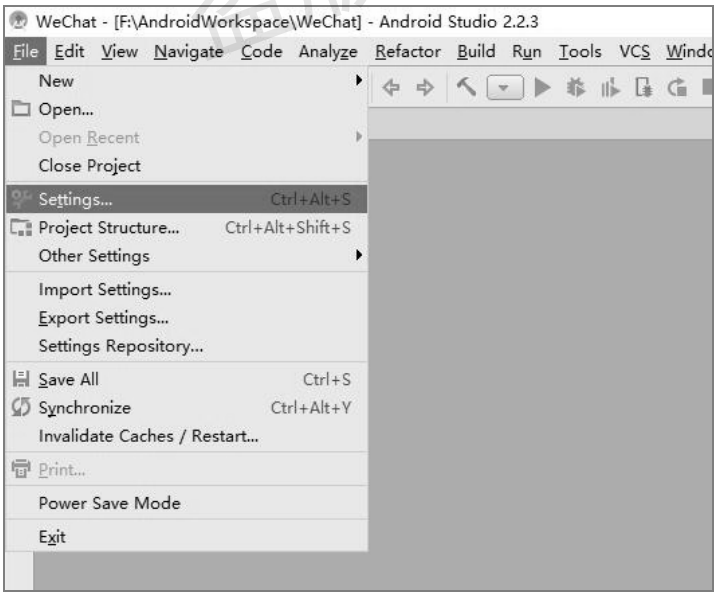

图 1-20　选择【File】→【Settings】命令

电子工业出版社版权所有 盗版必究

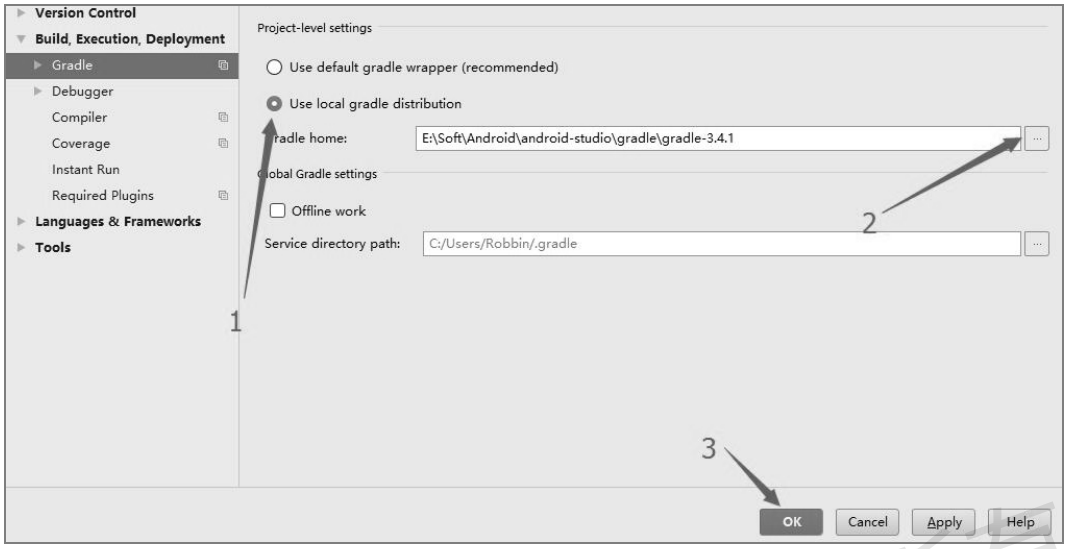

图 1-21　设置本地 Gradle 路径

2．Mac 版的手动更新

在启动台中打开终端，输入“open.gradle”命令后回车，打开更新文件所在的文件夹。将手动下载的“gradle-x.x-all.zip”文件放置在相应的文件夹中（如图 1-22 所示）。重启 Android Studio 后会自动进行更新（如图 1-23 所示）。

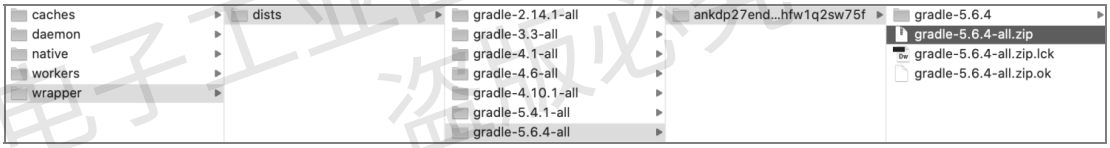

图 1-22　更新文件所在文件夹

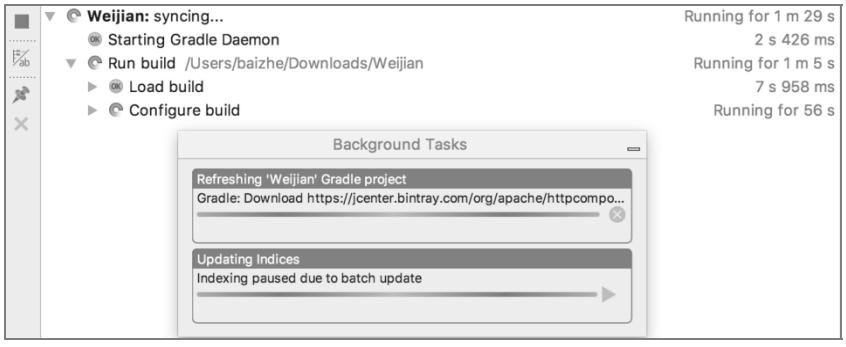

图 1-23　重启 Android Studio 后自动更新 Gradle

1.2.6　重构 Java 工程

Android Studio 升级 Gradle 后，打开原有工程会自动弹出“Plugin Update Recommended”提示框（如图 1-24 所示）。单击后打开“Android Gradle Plugin Update Recommended”对话

电子工业出版社版权所有 盗版必究

框，单击“Update”按钮（如图 1-25 所示），自动重构 Java 工程，详细进度显示在“Build”窗口中。

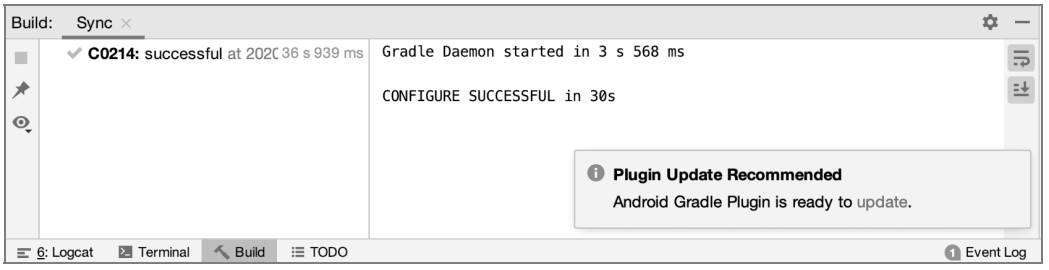

图 1-24　更新 Gradle 的提示

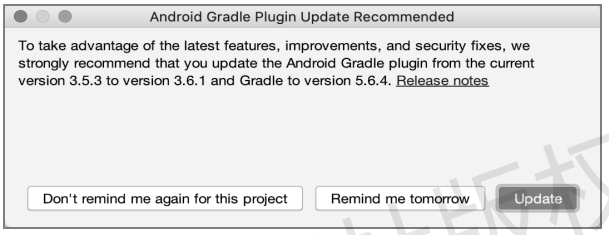

图 1-25　更新 Gradle 的对话框

1.3　创建 Android 工程

1.3.1　Android 工程的新建命令

选择【File】→【New】→【New Project】命令（如图 1-26 所示），打开“Create New Project”对话框，进行工程创建的向导。

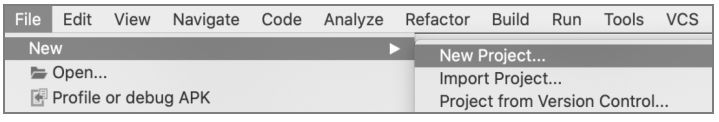

图 1-26　【New Project】命令

1.3.2　Android 工程的创建向导

工程创建向导的第 1 页（如图 1-27 所示）用于选择工程的类型，建议选择“Empty Activity”进行手机项目的 App 开发。若进行单项练习，可以针对练习项目选择相应的类型。

提示：Activity

Activity 是一个应用组件，用户可与其进行交互，用于绘制用户界面的窗口。窗口通常会充满屏幕，也可小于屏幕并浮动在其他窗口之上。后续将详细介绍 Activity。

单击“Next”按钮，进入工程创建向导的第 2 页（如图 1-28 所示），包含“Name”“Package name”“Save location”“Language”“Minimum SDK”等配置选项。

电子工业出版社版权所有 盗版必究

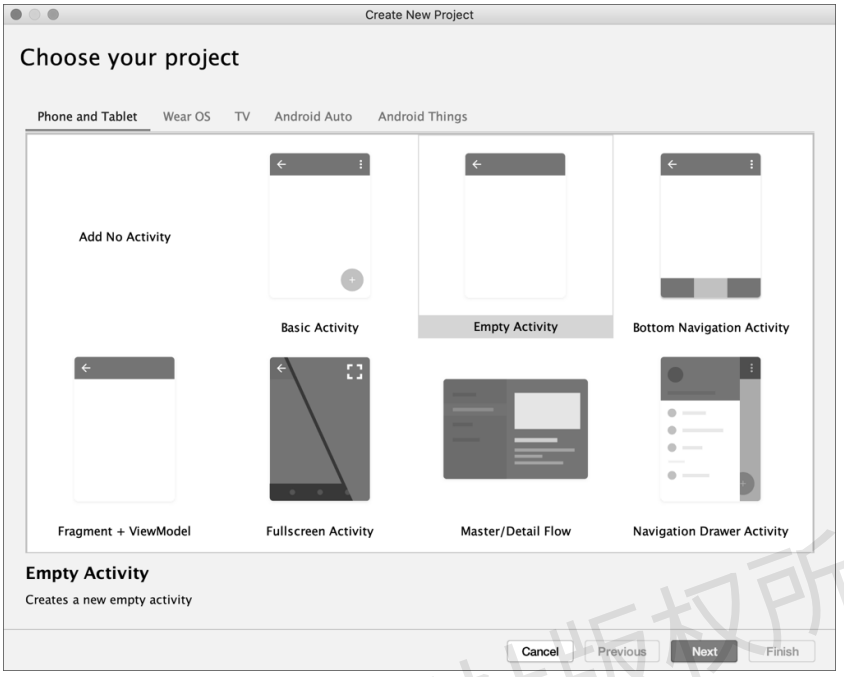

图 1-27　选择工程类型

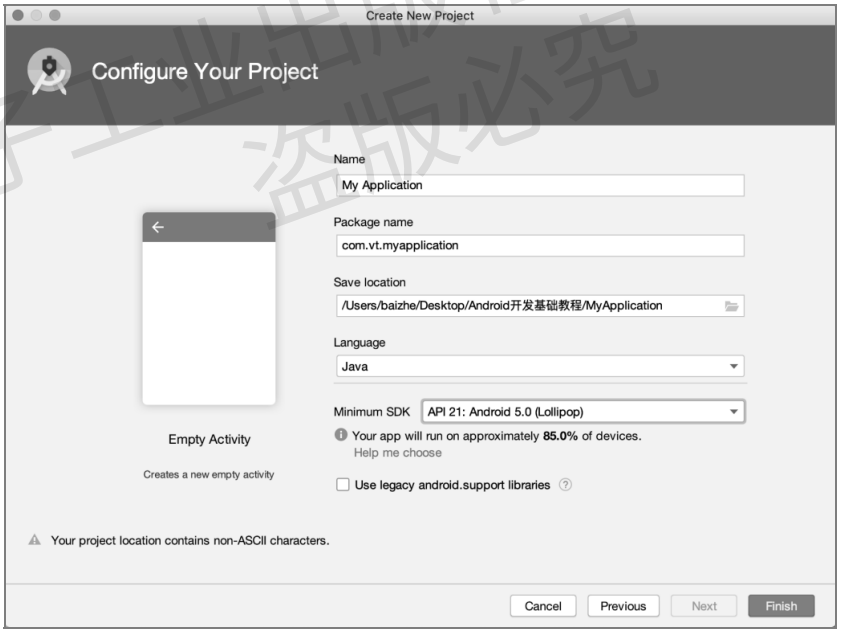

图 1-28　配置工程

- Name（应用名称）：安装到 Android 设备上后显示的名称。
- Package name（包名）：一般以企业域名倒置作为前缀，然后加上 App 的英文名。
- Save location（存储位置）：保存工程文件的路径。
- Language（语言）：可以选择 Java 或 Kotlin。

电子工业出版社版权所有
盗版必究

- Minimum SDK（最小 SDK 版本）：可以安装该 App 的安卓设备最低版本。
- Use legacy android.support libraries（使用传统 Android 支持库）：不勾选时使用 AndroidX 库，API level 29 及其以后的版本无法使用传统 Android 支持库。

单击“Help me choose”选项，可查看官方统计的各版本使用比例（如图 1-29 所示）。

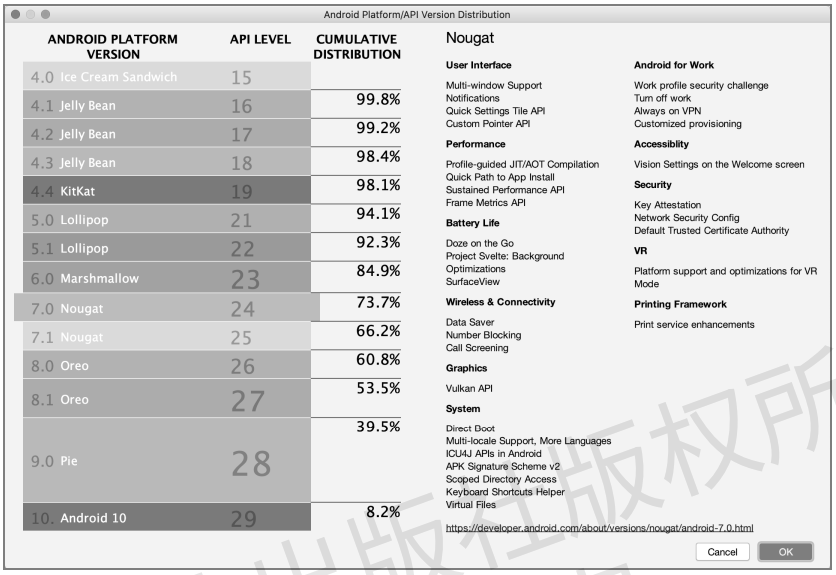

图 1-29　各版本使用比例

单击“OK”按钮，再单击“Finish”按钮完成工程的创建向导，显示 Android Studio 界面（如图 1-30 所示）。

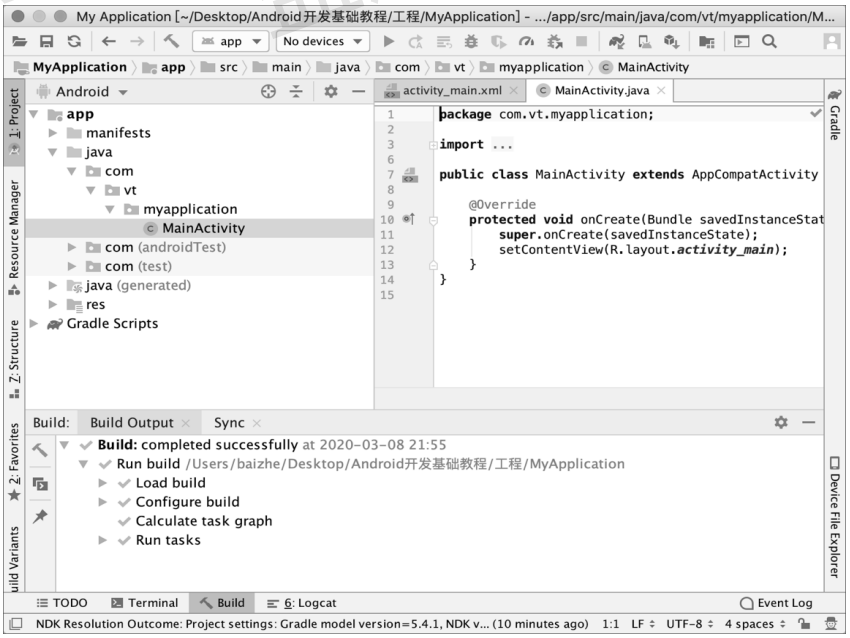

图 1-30　创建工程完成后的 Android Studio 界面

电子工业出版社版权所有 盗版必究

1.3.3　虚拟设备运行工程

1．创建虚拟设备

在工具栏中，单击“AVD Manager”按钮（如图 1-31 所示），打开“Android Virtual Device Manager”对话框。

图 1-31　“AVD Manager”按钮

在“Android Virtual Device Manager”对话框中，单击“Create Virtual Device”按钮（如图 1-32 所示），打开“Virtual Device Configuration”对话框。

图 1-32　“Android Virtual Device Manager”对话框

在“Virtual Device Configuration”对话框中，显示虚拟设备的参数列表（如图 1-33 所示）。根据需求选择相应的虚拟设备，然后单击“Next”按钮。

在选择 Android 系统的版本时，如果没有下载相应的系统镜像，需要选择“DownLoad”选项下载系统镜像（如图 1-34 所示）。

选择“DownLoad”选项后，会下载 SDK 文件并自动安装。安装完成后，单击“Finish”按钮，确认虚拟设备的配置信息（如图 1-35 所示），然后单击“Next”按钮完成虚拟设备的创建。

电子工业出版社版权所有 盗版必究

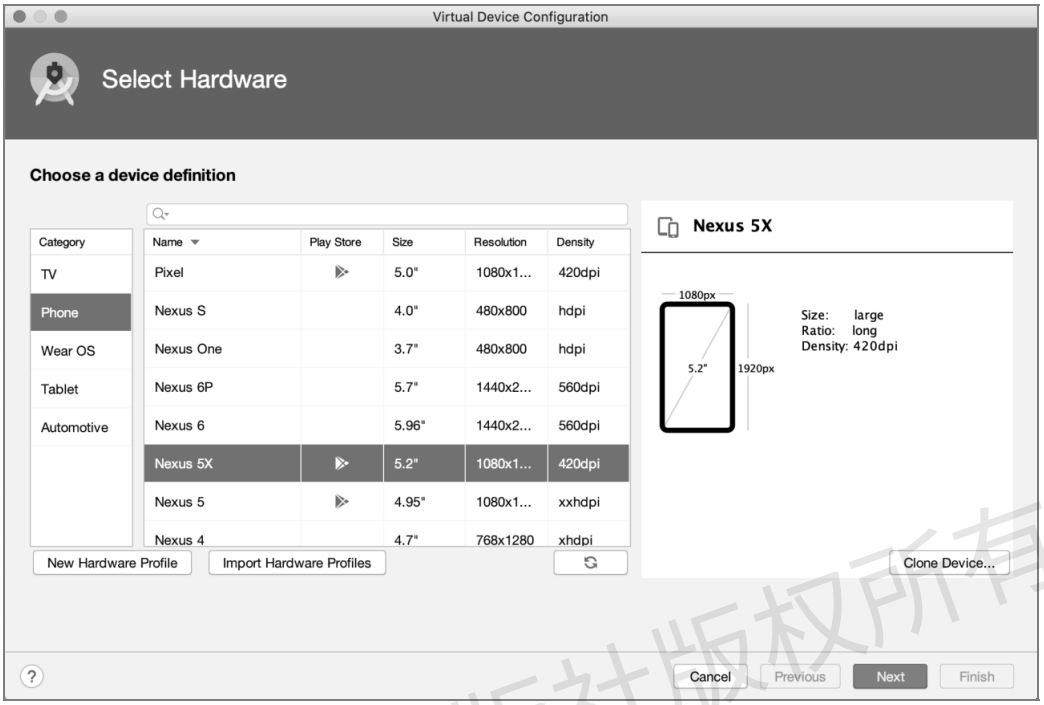

图 1-33　选择虚拟设备

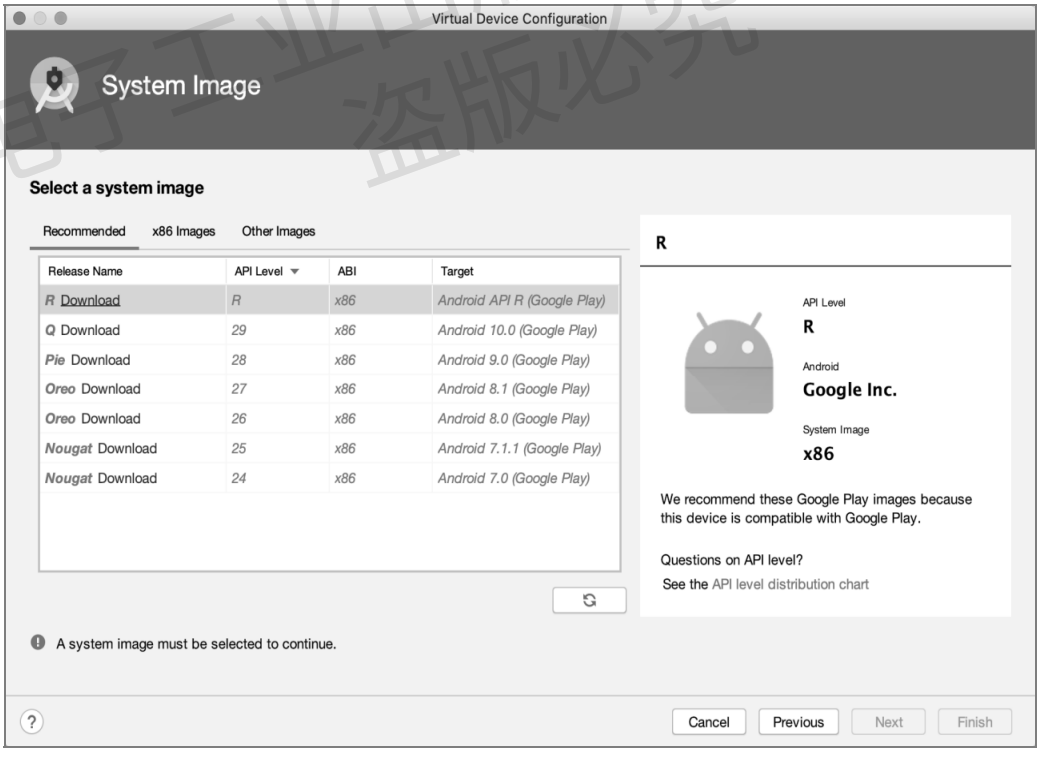

图 1-34　选择系统版本

电子工业出版社版权所有 盗版必究

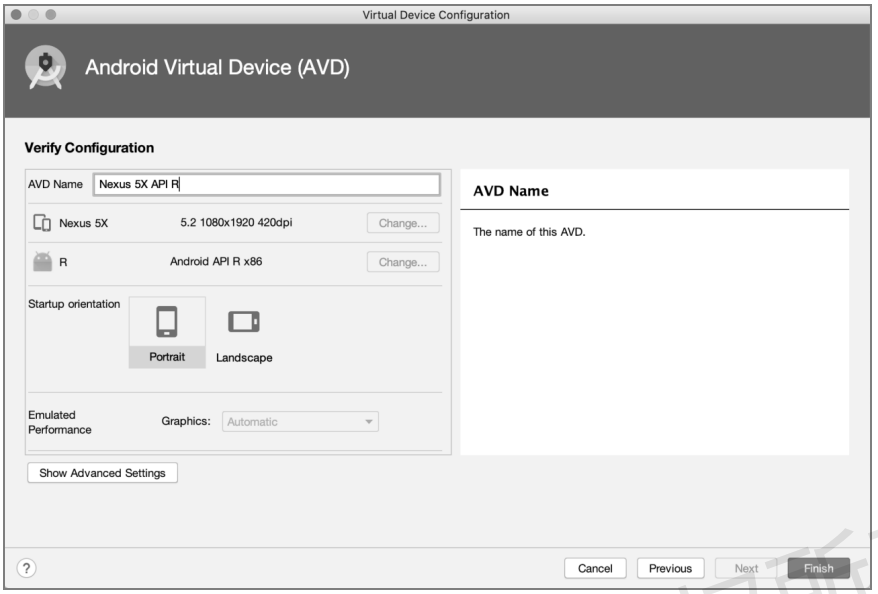

图 1-35　确认虚拟设备的配置信息

返回“Android Virtual Device Manager”对话框，选择相应的虚拟设备，单击“Launch this AVD in the emulator”图标（如图 1-36 所示），启动虚拟设备。

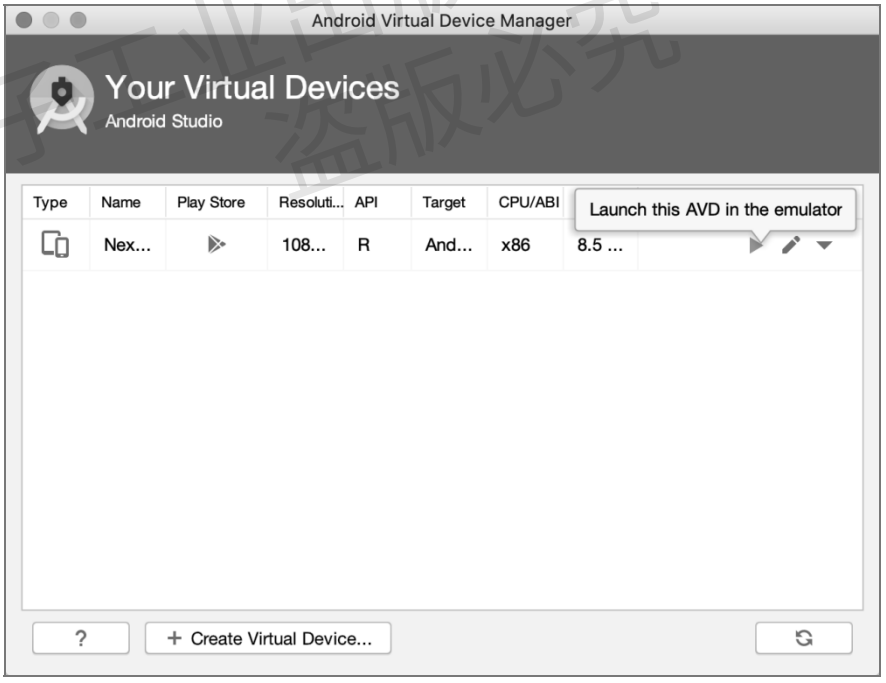

图 1-36　启动虚拟设备

Android 虚拟设备启动时间较长，若硬件配置较好，则可以大大缩短启动时间。启动完成后，显示虚拟设备界面（如图 1-37 所示）。

电子工业出版社版权所有 盗版必究

图 1-37　虚拟设备界面

2．运行工程

在工具栏中，选择创建的虚拟设备，然后单击“Run 'app'(^R)”按钮（如图 1-38 所示）。工程编译完成后，自动安装在虚拟设备中，然后自动启动运行（如图 1-39 所示）。

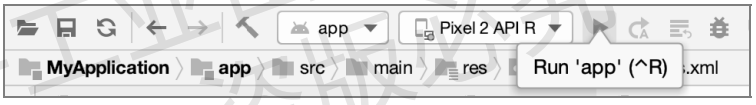

图 1-38　“Run 'app'(^R)”按钮

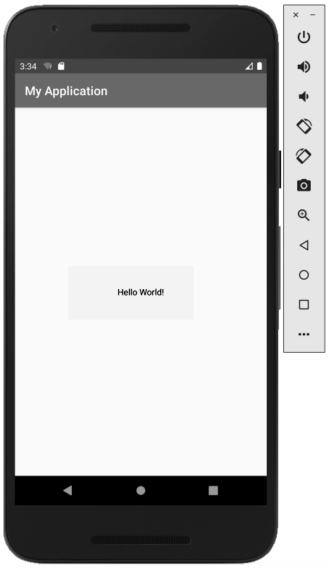

图 1-39　使用虚拟设备运行

电子工业出版社版权所有 盗版必究

1.3.4　物理设备运行工程

使用 USB 线将手机连接到计算机，在工具栏的运行/调试设备下拉菜单中会显示连接的手机作为首选项（如图 1-40 所示）。单击“Run 'app'(^R)”按钮，会编译工程并安装在连接的手机上，然后自动启动运行（如图 1-41 所示）。

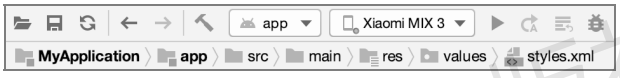

图 1-40　显示连接的手机

图 1-41　手机运行画面

1.3.5　生成签名的 APK 文件

APK 文件是 App 的安装文件，设置 APK 文件签名的目的是让 App 不被恶意生成的 APK 文件覆盖安装。在升级 App 时，只有同一签名的 APK 文件才能对 App 进行升级，从而避免恶意覆盖。

选择【Build】→【Generate Signed Bundle/APK】命令，打开“Generate Signed Bundle or APK”对话框，选择“APK”单选按钮，单击“Next”按钮（如图 1-42 所示）进入下一个页面。

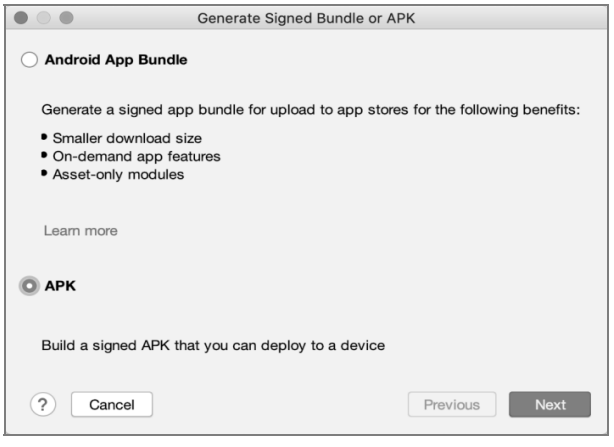

图 1-42　选择“APK”单选按钮

电子工业出版社版权所有 盗版必究

单击“Create new”按钮，打开“New Key Store”对话框，设置签名文件密码、签名密码和证书内容（如图 1-43 所示），然后单击“OK”按钮完成签名文件的创建。

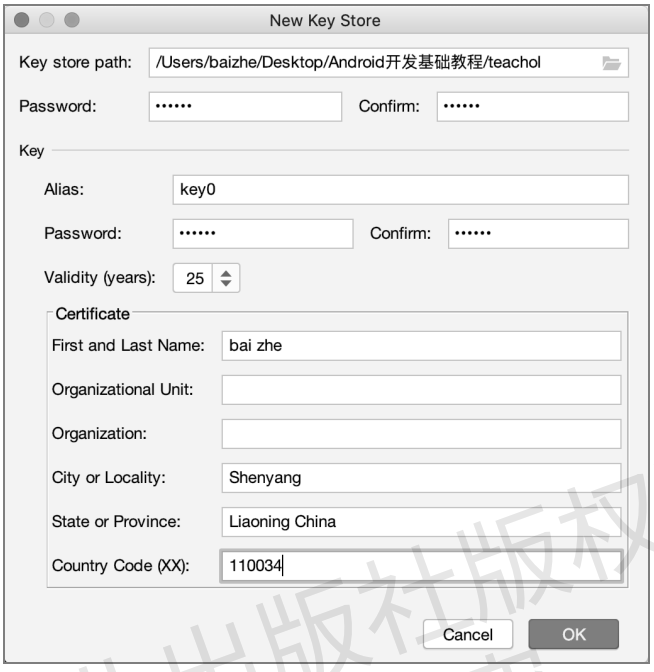

图 1-43　“New Key Store”对话框

创建签名文件后，在“Generate Signed Bundle or APK”对话框中会自动输入签名路径、签名路径密码、签名别名和签名密码（如图 1-44 所示），单击“Next”按钮，进入下一个页面。“Build Variants”选择“release”，勾选“Signature Versions”中的“V2（Full APK Signature）”复选框（如图 1-45 所示），单击“Finish”按钮生成签名的 APK 文件。

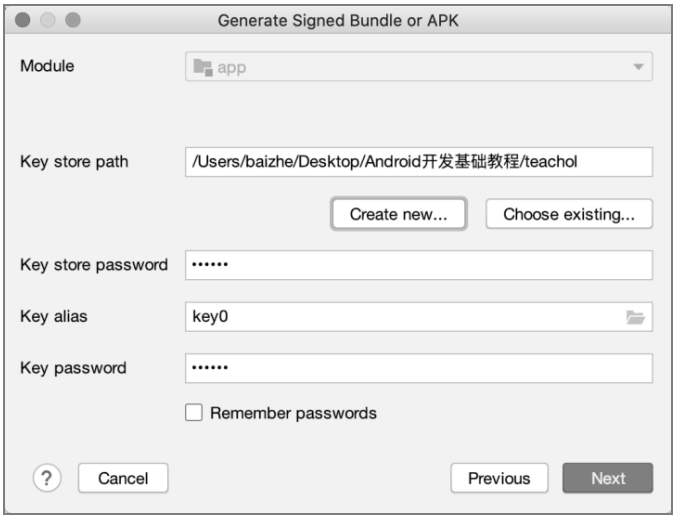

图 1-44　自动输入签名文件信息

电子工业出版社版权所有

图 1-45　选择完整版和完整签名

提示：防反编译

对 APK 文件进行反编译可以还原源代码，为了防止源代码被恶意使用，商业上使用的 App 还会进行防反编译处理。常用的防反编译方法包括混淆策略、整体 Dex 加固、拆分 Dex 加固、虚拟机加固等，推荐使用阿里聚安全、腾讯云应用乐固、爱加密、娜迦、梆梆等自动化加固处理以防反编译。

1.4　Android 的工程结构

1.4.1　Project 视图

Project 视图是以工程管理的形式显示文件的视图（如图 1-46 所示），主要包括 AndroidManifest.xml 文件（App 的配置信息）、build.gradle 文件（gradle 配置文件）、res 文件夹（资源文件）和 java 文件夹（源代码和测试代码）。

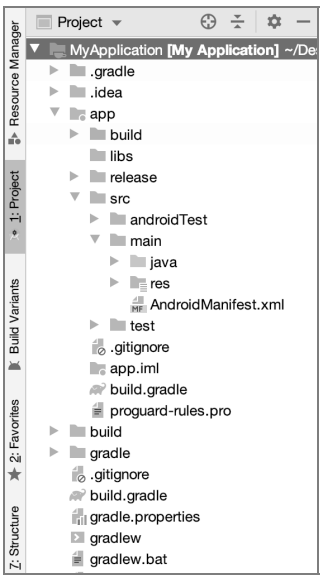

图 1-46　Project 视图

1.4.2　AndroidManifest.xml 文件

AndroidManifest.xml（如图 1-47 所示）是 Android 应用的配置文件，包含应用的基本信息，声明程序中的 Activity、ContentProvider、Service 和 Receiver，指定 permission 和 instrumentation。

1．<manifest>标签

<manifest>标签是 AndroidManifest.xml 的根节点，用于设置 xmlns:android 和 package 属性，还包括一个<application>标签。

电子工业出版社版权所有 盗版必究

- xmlns:android 属性：设置 Android API 的命名空间，用于识别控件的属性。
- package 属性：设置 App 的包名。

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.vt.myapplication">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="My Application"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

图 1-47　AndroidManifest.xml 文件

2．<application>标签

<application>标签用于声明应用程序，包含每个应用程序组件所声明的子元素，以及能够影响所有组件的属性。

- android:allowBackup 属性：设置 App 数据的备份和恢复功能，默认值为 true，可以通过 adb backup 和 adb restore 对 App 数据进行备份和恢复。
- android:icon 属性：设置一个 Drawable 资源作为 App 的普通图标（如图 1-48 所示）。
- android:label 属性：设置 App 的名称。
- android:roundIcon 属性：设置一个 Drawable 资源作为 App 的圆形图标（如图 1-49 所示）。

图 1-48　默认普通图标

图 1-49　默认圆形图标

- android:supportsRtl 属性：设置是否支持从右到左的布局，默认值为 true。当 build.gradle 文件中的 targetSdkVersion 值设置为 17 或更高时，可以使用 RTL（right-to-left）布局。
- android:theme 属性：设置界面皮肤主题。

3．<activity>标签

<activity>标签用于设置 Activity 的属性，Activity 必须被声明在<manifest>标签中，没有被声明的 Activity 不会被调用。android:name 属性设置声明的 Activity 的类名。

电子工业出版社版权所有

4．<intent-filter>标签

<intent-filter>标签用于指定 Activity、Service 或 BroadcastReceiver 能够响应的 Intent 对象类型，通过<action>、<category>和<data>子标签进行描述。

5．<action>标签

<action>标签用于给 Intent 过滤器添加子标签，<intent-filter>标签必须包含一个或多个<action>标签。如果不包含<action>标签，就不存在 Intent 过滤器。当 android:name 属性设置为 android.intent.action.MAIN 时，该 Activity 作为应用默认启动的 Activity。

6．<category>标签

<category>标签用于向 Intent 过滤器添加类别名称，android:name 属性的默认值为 CATEGORY_DEFAULT。

1.4.3　build.gradle 文件

本工程中包含 2 个 build.gradle 文件，分别是根目录下的顶级（top level）build.gradle 文件和 app 目录下的模块级（module level）build.gradle 文件。

顶级 build.gradle 文件位于项目根目录，用于配置适合项目中所有模块的构建设置。默认情况下，顶级 build.gradle 文件使用 buildscript{}代码块来定义项目中所有模块共用的 Gradle 存储区和依赖项。

模块级 build.gradle 文件位于 project/module 文件夹中，用于配置适合其所在模块的构建设置，通过配置这些构建设置可以提供自定义打包选项。

1.4.4　res 文件夹

Android 的资源文件存储在工程的 res 文件夹中（如图 1-50 所示），Android Studio 3 将资源分类存储在 4 类文件夹（前缀名为 drawable、layout、mipmap 和 values）中。drawable-v24 文件夹的后缀 v24 表示 API 的版本，mipmap-mdpi 文件夹的后缀 mdpi 表示 Android 设备的屏幕分辨率。

- drawable：用于存储图像和 XMLDrawable 文件，XMLDrawable 分为 AnimationDrawable、BitmapDrawable、ClipDrawable、ColorDrawable、GradientDrawable、NinePatchDrawable、RotateDrawable、StateListDrawable、ShapeDrawable 和 TransitionDrawable 等类型。
- layout：用于存储界面布局文件。
- mipmap：用于存储图标文件，默认包含自动生

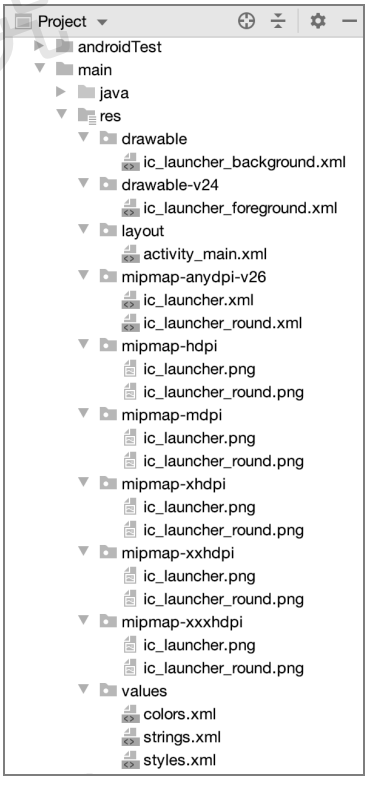

图 1-50　res 文件夹结构

电子工业出版社版权所有 盗版必究

成的 Launcher Icons（桌面图标），还可以存储 Action Bar and Tab Icons（桌面图标）和 Notification Icons（通知图标）。

- values：用于存储 colors.xml、strings.xml 和 styles.xml 等文件。

1.5 习　　题

1．下载最新版本的 Android Studio，安装后如果需要更新 Gradle，则将其更新到最新版本。

2．创建一个任意类型的手机 App 工程，分别使用虚拟设备和物理设备运行。

3．在 build.gradle 文件中修改支持的 API 最低版本，然后生成带签名的 APK 文件。

电子工业出版社版权所有
盗版必究