

# 第 1 章

## Android 编程入门

本章主要内容:

---

- Android 的定义
- Android 版本和功能集
- Android 构架
- 不同类型的 Android 设备
- Android Market 应用商店
- 如何获得开发 Android 应用的工具和 SDK
- 如何开发你的第一个 Android 应用

源代码下载: 本章没有源代码下载。

欢迎来到 Android 世界。本章将介绍 Android 是什么, 以及为什么它会如此受到开发者和设备厂商的关注。随后会介绍如何获得和安装所有必要的工具, 使你可在 Android Studio 2 的 Android 模拟器中测试应用, 并且开发你的第一个 Android 应用。学习完本章以后, 你将掌握必要的基础知识, 并且能够学习更加复杂的技术和技巧, 开发强大的 Android 应用。

### 1.1 Android 的定义

Android 是基于 Linux 改编的一个移动操作系统。刚开始是由一家名为 Android 的公司开发的。2005 年, 作为进入移动行业的战略之一, Google 收购了 Android 公司并且接手了

所有开发工作(包括整个开发团队)。

Google 希望 Android OS 是开放并且免费的，所以大部分 Android 源代码是使用开源 Apache 许可证发布的。这就意味着任何想使用 Android 的人都可以下载完整的 Android 源代码。与此同时，供应商(尤其是硬件生产商)可以在 Android 中加入自己的专有扩展，定制化 Android，使之与其他产品有所区别。这种开发模式使得 Android 对于供应商具有极大的吸引力，尤其是那些受到苹果 iPhone 影响的公司，苹果 iPhone 手机取得了巨大成功，并且改变了整个智能手机行业。当苹果手机发布时，许多智能手机生产商不得不争先恐后地寻找新方法重振他们的产品。这些生产商认为 Android 是一个解决方案。他们可以继续设计自己的硬件并且使用 Android 操作系统。有一些公司，包括摩托罗拉和索尼爱立信，利用 Android 开源政策优势来开发自己的移动操作系统。

采用 Android 的最大优势是它为应用开发提供了一套统一的途径。开发者只需要为通用的 Android 系统开发应用，他们的应用就可以在众多不同的设备上运行，只要这些设备使用的是 Android 操作系统。在智能手机世界，应用是成功链条中最重要的一环。

### 1.1.1 Android 版本

自从 Android 第一版发布，它已经经历了好几轮的更新。表 1-1 列出了各种 Android 版本和代号。

表 1-1 Android 版本简史

Android 版本	发布日期	代码名称
1.1	2009 年 2 月 9 日	
1.5	2009 年 4 月 30 日	Cupcake
1.6	2009 年 9 月 15 日	Donut
2.0/2.1	2009 年 10 月 26 日	Éclair
2.2	2010 年 5 月 20 日	Froyo
2.3	2010 年 12 月 6 日	Gingerbread
3.0/3.1/3.2	2011 年 2 月 22 日	Honeycomb
4.0	2011 年 10 月 18 日	Ice Cream Sandwich
4.1	2012 年 7 月 9 日	Jelly Bean
4.4	2013 年 10 月 31 日	KitKat
5.0	2014 年 11 月 12 日	Lollipop
6.0	2015 年 10 月 5 日	Marshmallow
7.0	TBD	Nougat

2016 年，Google 发布了 Android 7.0；以下是 Android 7.0 的主要更新：

- 分屏多窗口模式
- 重新设计了通知栏

- 优化了 Doze 功能
- 不再使用 JRE(Java Runtime Environment, Java 运行环境), 改用 OpenJDK(Java 平台的开源化实现)

当你了解 Android 版本信息时, 需要特别注意 Android 的每一个版本都有自己的功能和 API。因此, 如果为最新版本的 Android 编写应用, 该应用使用了旧版 Android 中没有的 API, 那么只有运行新版本的 Android 设备才能使用该应用。

### 1.1.2 Android 功能

因为 Android 是开源的并且生产商可以自由地定制, 所以不存在固定的硬件或软件配置。然而, 标准 Android 操作系统提供了许多功能, 包括:

- **存储**——SQLite, 一个轻量级关系数据库, 用来执行数据存储。第 7 章会详细讨论数据存储。
- **连接**——GSM/EDGE、IDEN、CDMA、EV-DO、UMTS、Bluetooth(蓝牙, 包括 A2DP 和 AVRCP)、Wi-Fi、LTE 以及 WiMAX。第 11 章会详细讨论网络连接。
- **消息传送**——SMS 和 MMS。第 9 章会详细讨论消息传送。
- **多媒体**——支持 H.263、H.264(使用 3GP 或 MP4 容器)、MPEG-4 SP、AMR、AMR-WB(使用 3GP 容器)、AAC、HE-AAC (使用 MP4 或 3GP 容器)、MP3、MIDI、Ogg Vorbis、WAV、JPEG、PNG、GIF 以及 BMP。
- **硬件支持**——加速度传感器、摄像头、数字指南针、距离传感器和 GPS。
- **多点触控**——多点触屏。
- **多任务**——多任务应用。
- **热点**——通过有线或者无线热点共享互联网连接。

Android 的网页浏览器基于开发开源项目 WebKit 和 Chrome 的 V8 JavaScript 引擎。

### 1.1.3 Android 构架

要了解 Android 是如何工作的, 请参考图 1-1。图中展示了组成 Android 操作系统(OS)的各个层。

Android 操作系统被大致分为主要四层, 共五大部分:

- **Linux 内核(Linux Kernel)**——这个是 Android 的内核。这一层包含了 Android 设备所有硬件组件的底层设备驱动。
- **代码库(Library)**——这里包含了 Android 操作系统主要功能的代码。例如, SQLite 库提供了数据库支持, 应用可以使用它存储数据。WebKit 库提供了网页浏览的功能。
- **Android 运行库(Android Runtime)**——Android 运行库和代码库属于同一层。为开发者提供核心库, 使开发者可使用 Java 编程语言编写 Android 应用。Android 运行库同时包含 Dalvik 虚拟机, 它使每一个 Android 应用运行在自己的进程上, 并使用自己的 Dalvik 虚拟机实例(Android 应用被编译成 Dalvik 可执行文件)。Dalvik 是特

别为 Android 设计的虚拟机，为有限的内存和 CPU 性能的电池供电移动设备做了特殊优化。

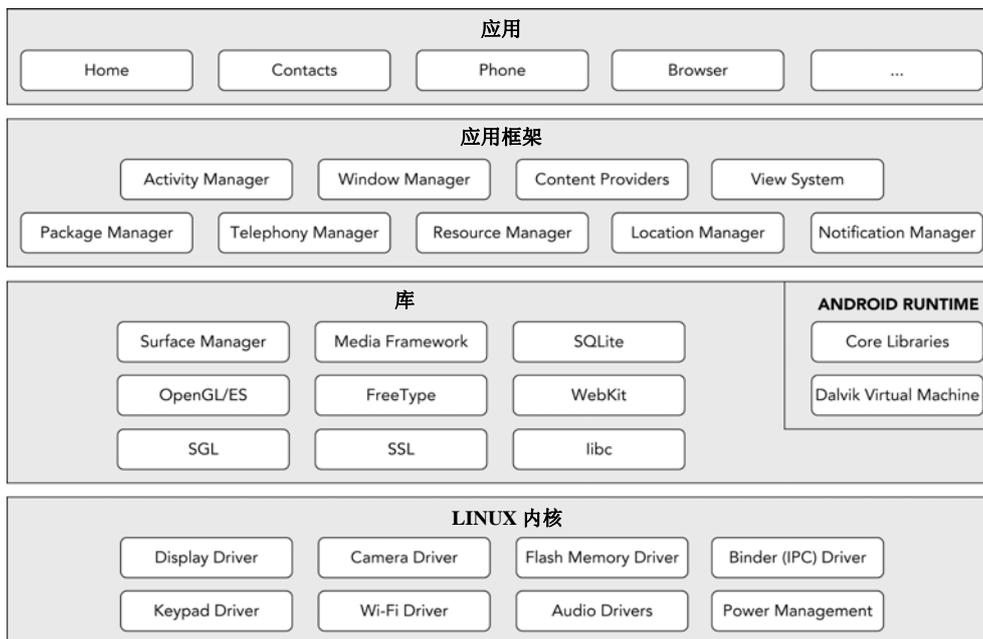


图 1-1

- **应用框架 (Application Framework)**——应用构架层为应用开发者开放了各种 Android 操作系统的功能，使开发者可以在自己的应用中使用。
- **应用层 (Application)**——最高层包含了 Android 设备上的原生应用(例如电话、通讯录、浏览器等)，以及用户从 Android Market 下载和安装的应用。所有你开发的应用也位于这一层。

#### 1.1.4 市场中的 Android 设备

在市场上能找到各种形状和尺寸的 Android 设备，包括但不限于下面几种类型：

- 智能手机
- 平板电脑
- 电子阅读器
- 网路电视
- 自动驾驶汽车
- 智能手表

很可能你至少拥有一个以上的上述设备。图 1-2 所示为 Samsung(三星)Galaxy Edge 7。

另外一种比较流行的设备是平板电脑。平板电脑主要有 7 英寸和 10 英寸两种尺寸(对角线测量)。

除了智能手机和平板电脑以外，Android 也用在专用设备上，比如电子阅读器。图 1-4

所示为使用 Android 操作系统的由 Barnes and Noble's 出品的 NOOK Color 电子阅读器。

除了我前面提到的流行的移动设备，Android 也出现在了我们的手腕上。智能手表以及可穿戴设备已经成为一个主要的 Android 分支。图 1-3 所示是摩托罗拉的 Moto 360 智能手表，运行的是 Android Wear(专门为可穿戴设备设计的一个 Android 操作系统)。



图 1-2



图 1-3

当笔者正在写作本书时，三星 Galaxy Nexus(见图 1-4)是唯一运行纯净版本的 Android 的设备。许多生产商在 Android 操作系统上添加了自己的修改，以便用在他们特定的设备上。摩托罗拉的设备运行 Motoblur，HTC 的设备运行 HTC Sense 等。然而，Nexus 设备只会运行没有修改过的纯净版本的 Android。



图 1-4

### 1.1.5 Android Market

如前所述，应用是决定一个智能手机平台是否成功的关键要素之一。iPhone 的成功就再一次印证了应用在决定一个新平台是否成功中扮演着多么重要的角色。与此同时，如何让普通用户更加方便快捷地下载和安装应用就变得极其重要。

用户可以非常方便地使用预先安装在 Android 设备中的 Google Play 来获取第三方提供的应用。在 Google Play 商店中，用户可以下载各种付费和免费的应用。由于一些法律问题，付费应用只在某些国家开放下载和安装。

**注意：**第 13 章将详细介绍 Google Play 商店以及如何在 Google Play 商店中销售自己开发的应用。

## 1.2 准备开发所需的工具

现在你已经知道 Android 是什么并且知道它能做些什么，我相信你已经迫不及待地想要开发一些 Android 应用了。但是在开始开发第一个应用之前，还需要准备好一些必要的开发工具。

可以在装有 Mac、Windows 或者 Linux 系统的电脑上开发 Android 应用。Android 开发中所有的必备工具都是可以免费获取的。本书中几乎所有的程序示例都是在 Android Studio 中开发的。笔者使用的是 Windows 10 操作系统。如果你使用的是 Mac 或者 Linux 系统，本书中出现的截图会和你在电脑中看到的大致相同。即使稍有区别，也不会影响你的操作过程。

现在就让我们开始这段奇妙之旅吧！

### Java JDK 8

使用 Android Studio 2 需要在电脑上安装 Java SE Development Kit 8 (JDK)。如果你的电脑上还没有安装 JDK 8，需要在进入下一步之前先到以下地址下载并安装 JDK8：

[www.oracle.com/technetwork/java/javase/downloads/jdk8-download-2133151.html](http://www.oracle.com/technetwork/java/javase/downloads/jdk8-download-2133151.html)

### 1.2.1 Android Studio

首先需要下载的最重要的开发工具是 Android Studio 2。当下载并安装了 Android Studio 2 以后，就可以使用 SDK 管理器下载和安装多个不同版本的 Android SDK。同时安装多个版本的 Android SDK 可以使你能够为不同设备开发应用。例如，你可以开发一个针对 Android Nougat 的应用，但是现在只有不到 1% 的 Android 手机使用这个系统。由于安装了多个版本的 Android SDK，因此也可以同时发布针对 Android Marshmallow 和 Android Lollipop 用户的应用。可以使用 Android 设备管理器新建设备模拟器。

可从以下地址下载 Android Studio 2(见图 1-5):  
<http://developer.android.com/sdk/index.html>。

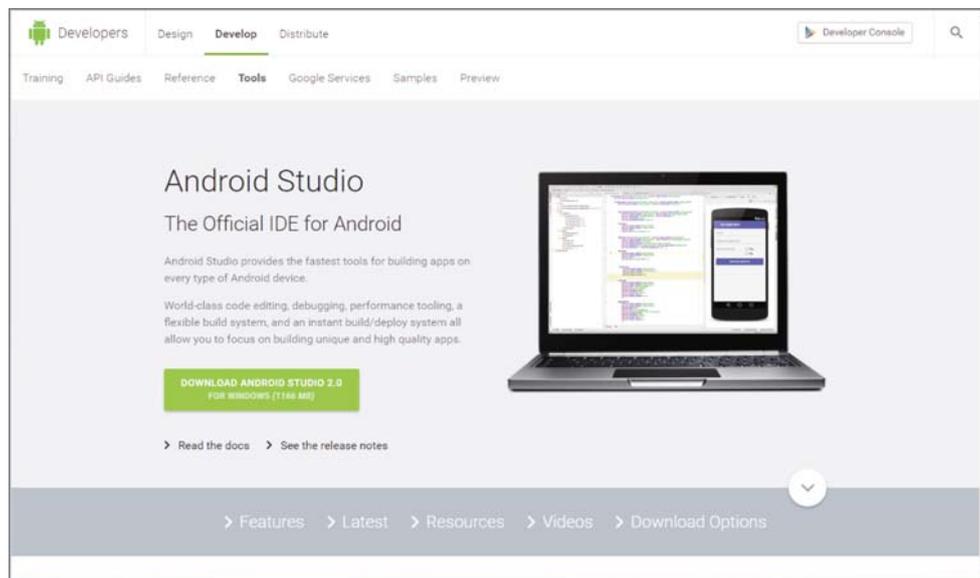


图 1-5

Android Studio 2 打包在一个可执行文件中。运行该执行文件安装和配置 Android Studio 2。当下载并运行安装程序以后，可通过以下步骤完成安装：

(1) 接受如图 1-6 所示的使用条款。

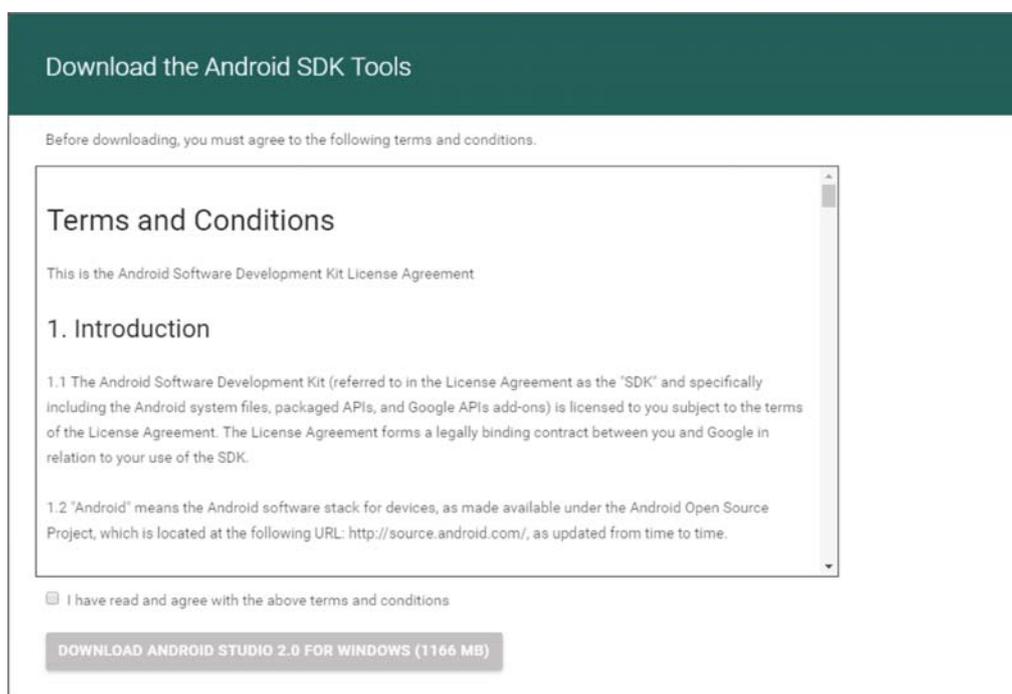


图 1-6

(2) 如果在你的电脑中已经安装了旧版本的 Android Studio，安装程序会自动提示你卸载旧版本。虽然旧版本的 Android Studio 被卸载了，但是所有配置都会保留。因此当你安装完毕后，仍可以在 Android Studio 2 中继续使用之前的配置。如图 1-7 所示，安装程序自动提示你卸载旧版本的 Android Studio。

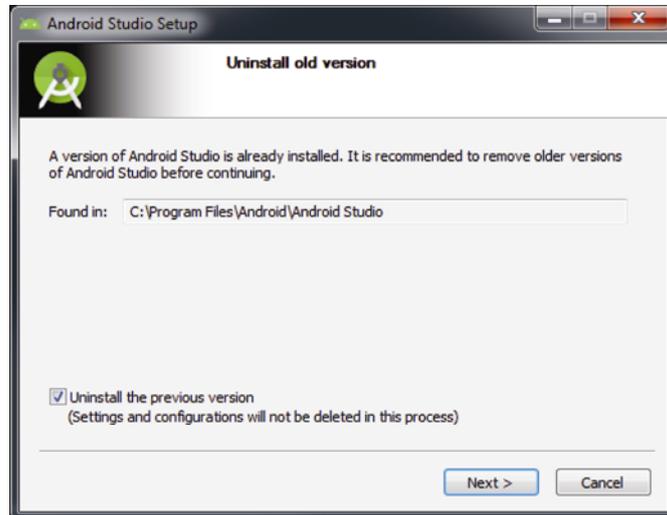


图 1-7

(3) 在 Welcome to Android Studio Setup 界面中单击 Next 按钮(如图 1-8 所示)。



图 1-8

(4) 如图 1-9 所示，选择想要安装的 Android Studio 组件。Android Studio 是默认选中的(并且不能取消)，显然我们的目的就是要安装 Android Studio。Android SDK 和 Android Virtual Device 也是默认选中的。我们就使用默认设置，单击 Next 按钮继续安装。

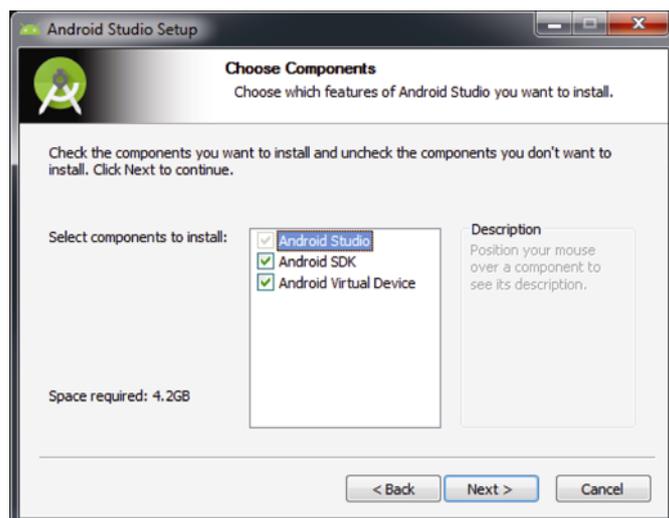


图 1-9

(5) 如图 1-10 所示，我们会看到一个 License Agreement 界面。单击 I Agree 按钮并继续安装。



图 1-10

(6) 在安装配置界面，最好使用安装程序指定的默认位置并单击 Next 按钮。然后我们会看到 Choose Start Menu Folder 的界面(如图 1-11 所示)。单击 Install 按钮开始安装 Android Studio 2。

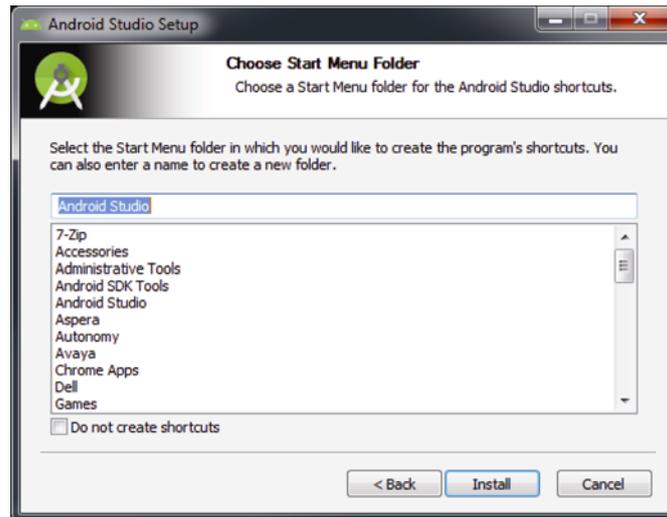


图 1-11

(7) 由于不同的电脑配置，安装 Android Studio 2 会需要一些时间。你会看到一个安装进度条，它能够帮助你跟踪安装状态。Android Studio 2 会同时安装一个默认的 SDK，在笔者的例子中是 Android Marshmallow。在后面的安装过程中，你有机会选择安装其他 SDK。Android SDK 允许你为特定 Android 版本编写应用。也就是说，基于 Marshmallow SDK 编写的应用可以在运行 Marshmallow 的 Android 设备上使用。如果应用中没有使用 SDK 特定的功能，那么它可以在其他 Android 版本上使用。

(8) 安装完毕后，将会看到一个 Completing Android Studio Setup 界面(如图 1-12 所示)。保持 Start Android Studio 复选框为选中状态并单击 Finish 按钮。



图 1-12

(9) Android Studio 2 会询问你是否使用之前旧版本中的配置或者使用新的配置。如果在第一步的安装过程中卸载了旧版本，Android Studio 会询问你是否在新版本中使用旧版本

的配置信息(如图 1-13 所示)。

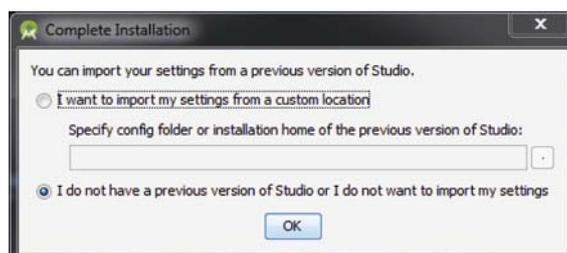


图 1-13

现在我们已经成功地安装了 Android Studio 2，你需要通过以下步骤做一些相应的配置：

(1) 在 Welcome 界面中单击 Continue 按钮。如图 1-14 所示，在 Install Type 界面中选择 Standard，并且单击 Next 按钮。

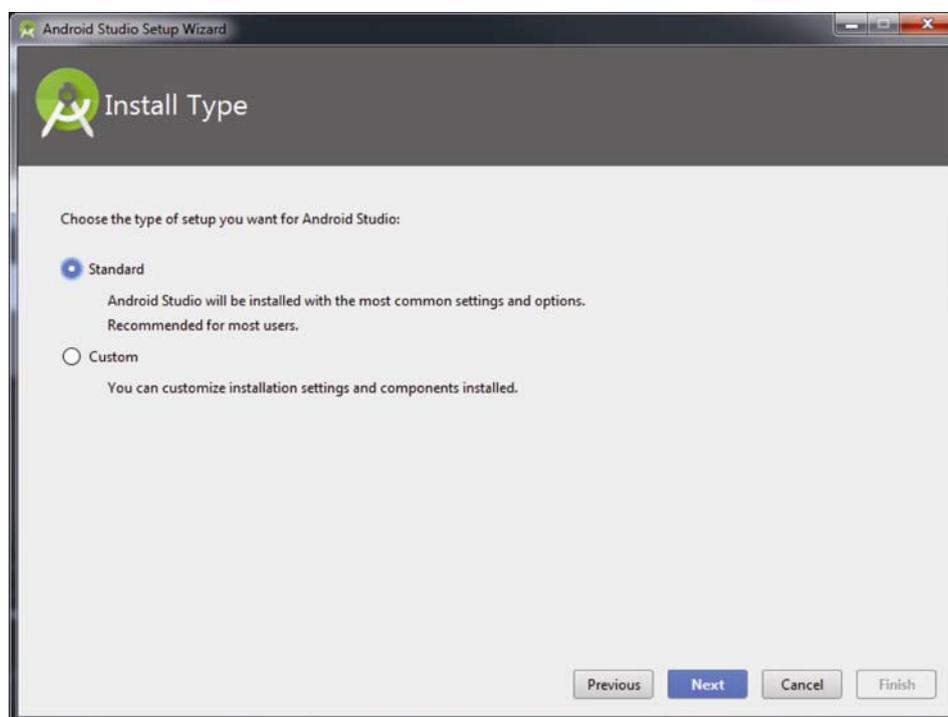


图 1-14

(2) 在 Verify Settings 界面中单击 Finish 按钮，完成 Android Studio 2 的安装。当整个安装过程结束后，你会看到 Welcome to Android Studio 界面(如图 1-15 所示)。

现在已经成功地安装了 Android Studio。下面是时候下载最新最出色的 Android SDK 了。

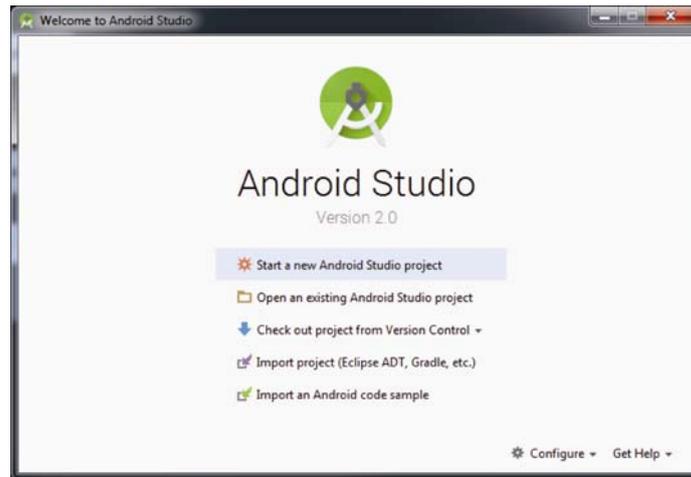


图 1-15

## 1.2.2 Android SDK

Android SDK 是需要下载的软件中最重要的部分。它包括我们在开发应用过程中所需要的类和工具。SDK 是根据与之对应的 Android OS 命名的。Android Studio 2 会默认安装 Marshmallow SDK, 这就意味着你开发的应用可以顺利地运行在运行 Android Marshmallow 的设备上使用。

然而, 如果想安装不同版本的 Android SDK, 也可以从 Android Studio 欢迎界面(如图 1-15 所示)启动 SDK Manager 来安装。在 Android Studio 欢迎界面的右下角, 单击 Configure 下拉菜单。在 Configure 选择菜单中选择 SDK Manager。

SDK 配置界面(如图 1-16 所示)显示 Marshmallow SDK 已经安装。Android N 可以安装(在撰写本书时, Android Nougat 还处于最终的 Beta 版本, 所以现在它可能不叫这个名字)。

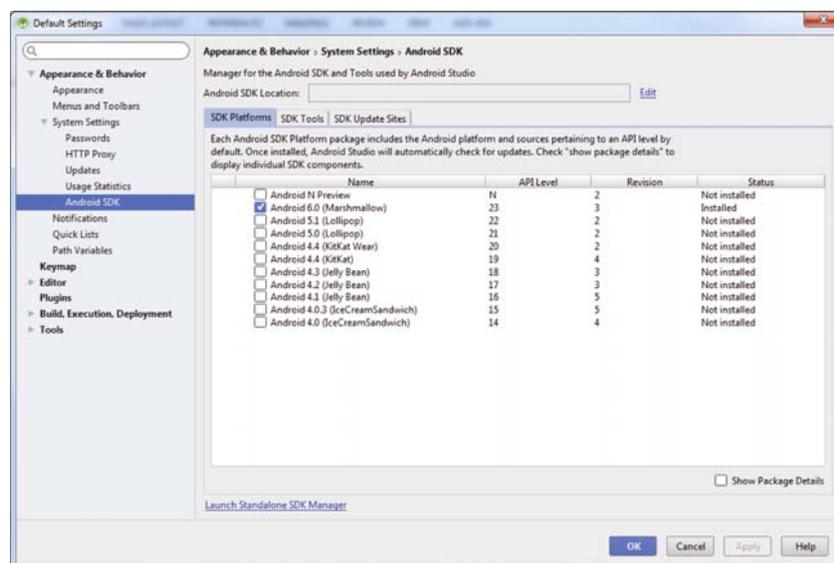


图 1-16

选择 Android Nougat，单击 Apply 按钮并接着单击 OK 按钮。在 SDK 完成安装以前，你需要接受相应的同意许可，如图 1-17 所示。

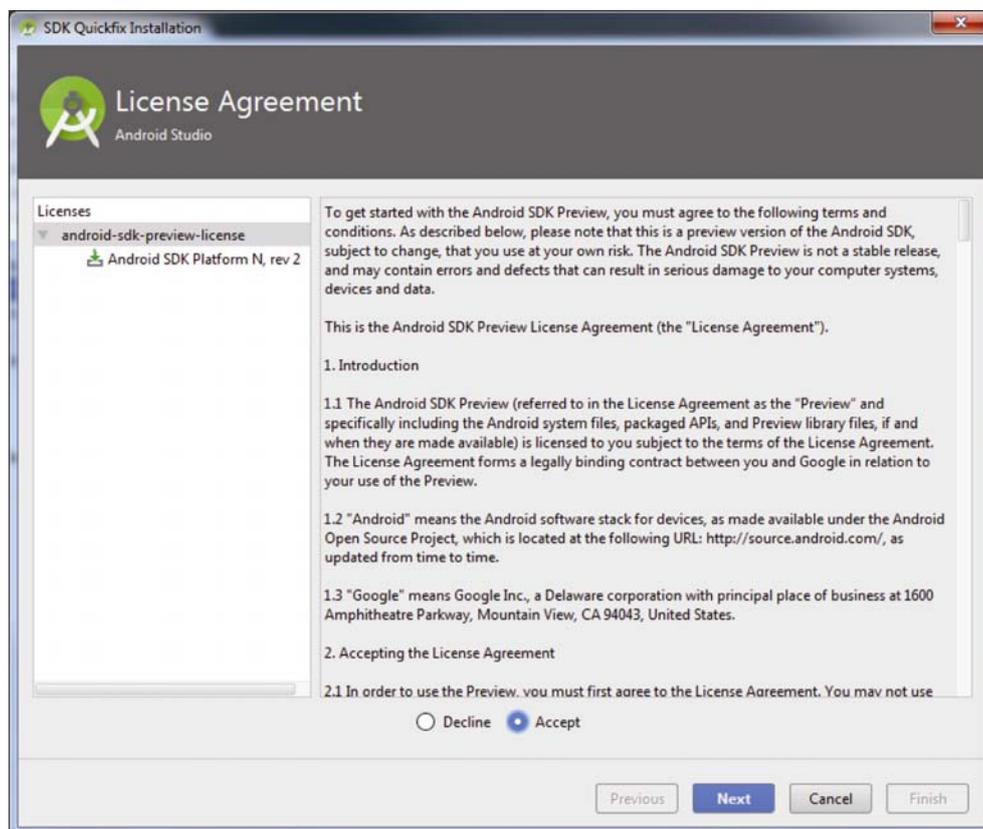


图 1-17

现在已经完成了 Android Studio 的配置过程。下一部分将详细讨论如何设置 Android 虚拟机。成功地设置了虚拟机，就可以在虚拟机上测试你的 Android 应用。

### 1.2.3 创建 Android 虚拟机

在本节中，将创建一个 Android 虚拟机(Android Virtual Device, AVD)。这样你就可以在虚拟机上测试 Android 应用。AVD 是一个用来模拟真实设备的模拟器。每一个 AVD 由一个硬件配置文件、系统镜像、虚拟存储空间比如 SD 卡所组成。需要注意的是模拟器不是完美的。像游戏(非常依赖 GPU)或者一些使用 GPS 传感器或者加速计的应用，它们无法保证在模拟器和真机上有同样的运行速度和一致性。但是，如果要测试一些普通的功能，模拟器还是一个很好的选择。

可以创建任意数量的 AVD，使用不同的配置测试 Android 应用。这样的测试是非常重要的，它能够确定应用在不同性能的设备上能否正常运行。

按照下面的步骤创建 AVD。这个示例演示了如何创建一个 AVD(即 Android 模拟器)，模拟一个具有 Nexus 5x 的硬件配置，并且运行 Android N 的 Android 设备。

(1) 启动 Android Studio 显示 Welcome 界面(如图 1-15 所示)。单击 Start a New Android Studio Project 按钮。将会看到 Create New Project Wizard 界面(如图 1-18 所示)。

(2) 新建一个 HelloWorld 项目(在本章节最后部分会使用到)。在 Application Name 文本框中输入 Chapter1HelloWorld。

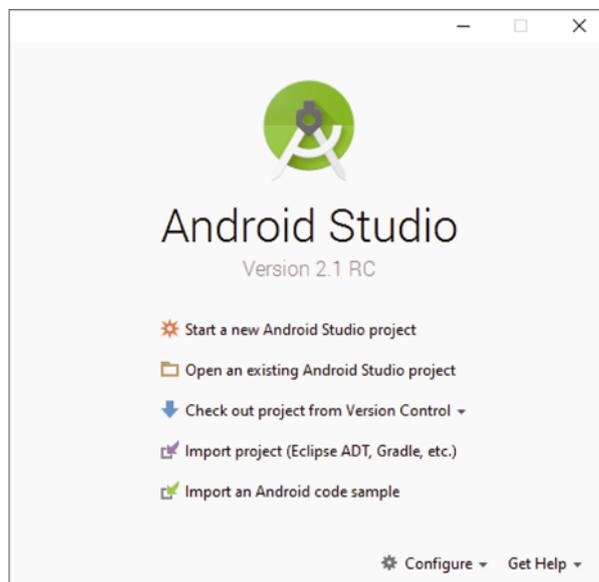


图 1-18

(3) 在 New Project 界面中，可以保留其他文本框中的默认值(在后续章节中会详细解释)。单击 Next 按钮。

**注意：**为了快速地创建一个 Hello World 项目并且创建一个 AVD，在项目创建的过程中，你将接受许多默认值，不需要任何解释。现在是没有问题的，因为所有的设置在随后的章节中会得到非常详细的解释。

(4) 你将会看到 Targeted Android Devices 界面。默认情况下，Create New Project Wizard 将会根据 Google Play 统计数据为你选择使用率最高的 Android SDK 版本。当写作本书时，Google Play 显示 74.3%的活跃设备使用 Android Jelly Bean。现在是接受默认值，如图 1-19 所示，单击 Next 按钮。

(5) 在 Add an Activity to Mobile 界面，接受默认选项——Empty Activity(如图 1-20 所示)，单击 Next 按钮。

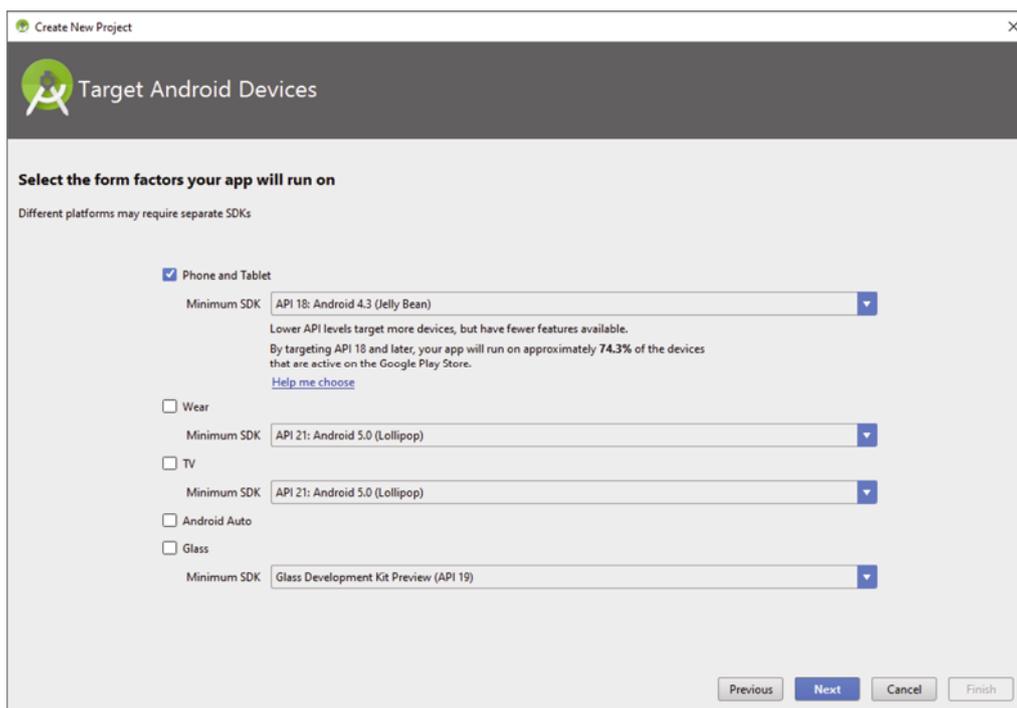


图 1-19

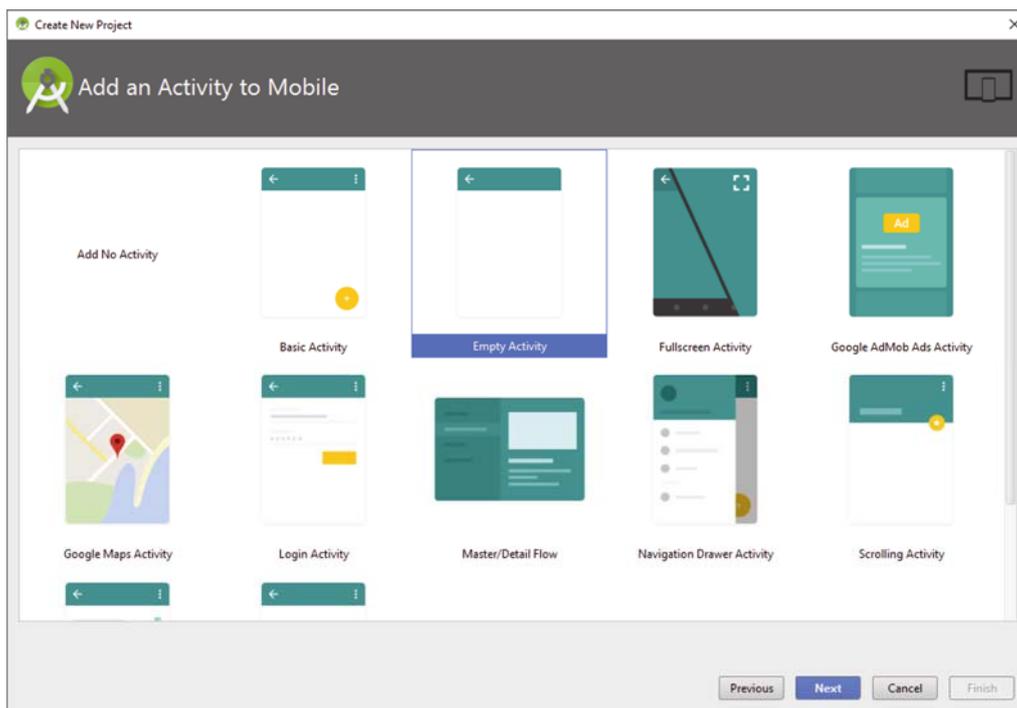


图 1-20

(6) 在 Customize the Activity 界面接受默认值，如图 1-21 所示，单击 Finish 按钮。如

图 1-22 显示打开的 Android Studio IDE 界面。

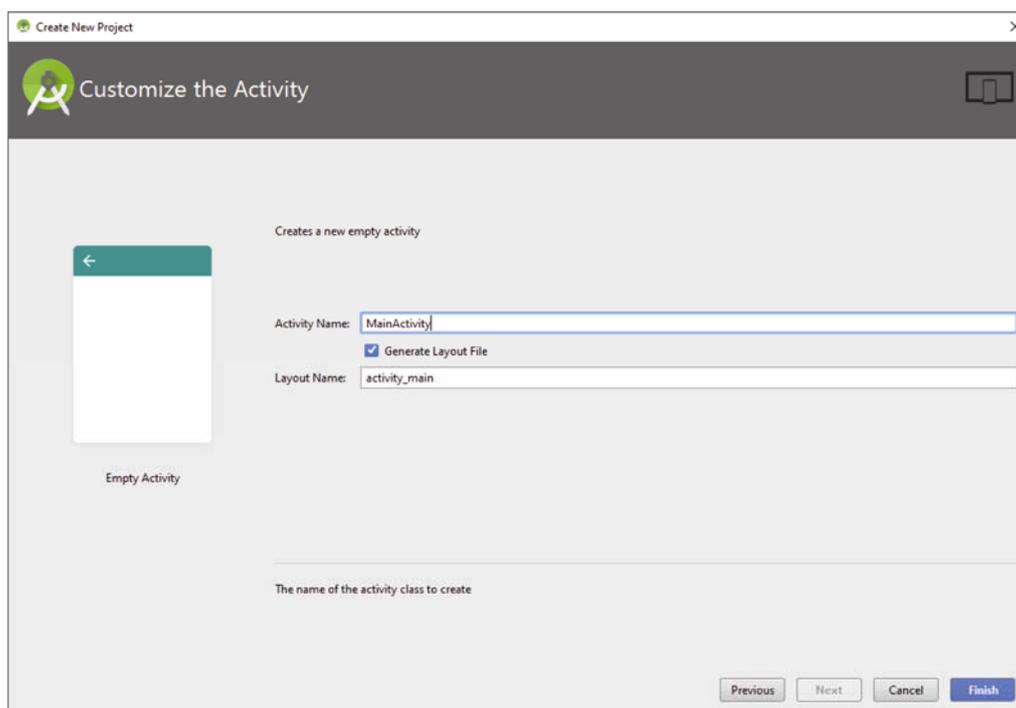


图 1-21

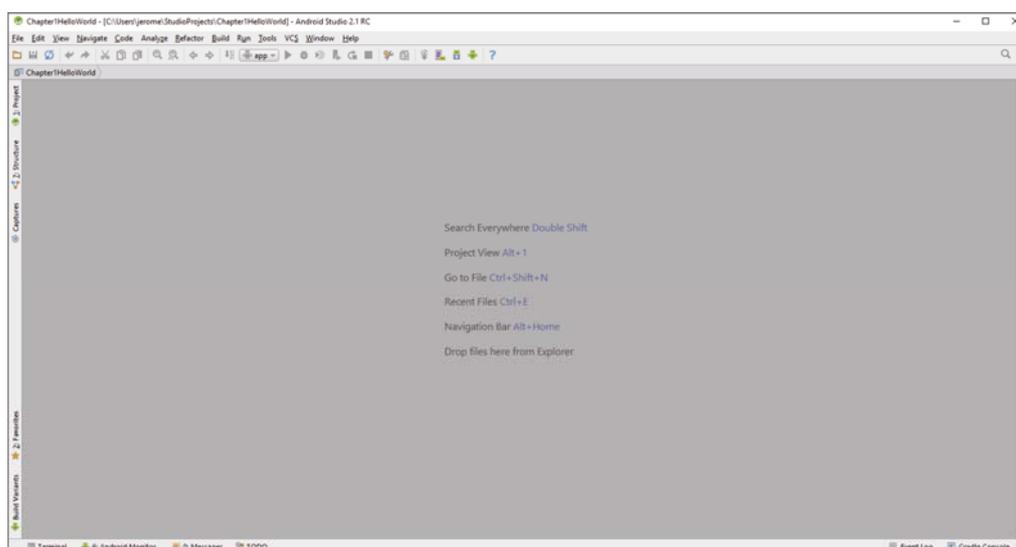


图 1-22

(7) 选择 Tools | Android | AVD Manager 或者单击工具栏中的 AVD Manager 按钮启动 AVD Manager。图 1-23 显示了 Android Virtual Device Manager Wizard，你可以在桌面上使用它来创建 AVD，用来在 Android 中模拟应用。

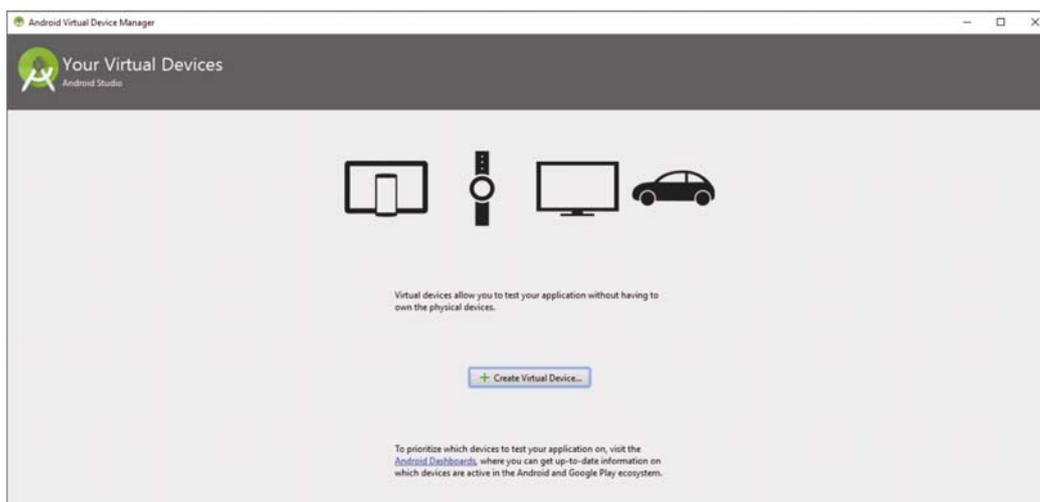


图 1-23

(8) 单击+ Create Virtual Device 按钮新建一个 AVD。我们会看到打开了 Virtual Device Configuration 界面，如图 1-24 所示。

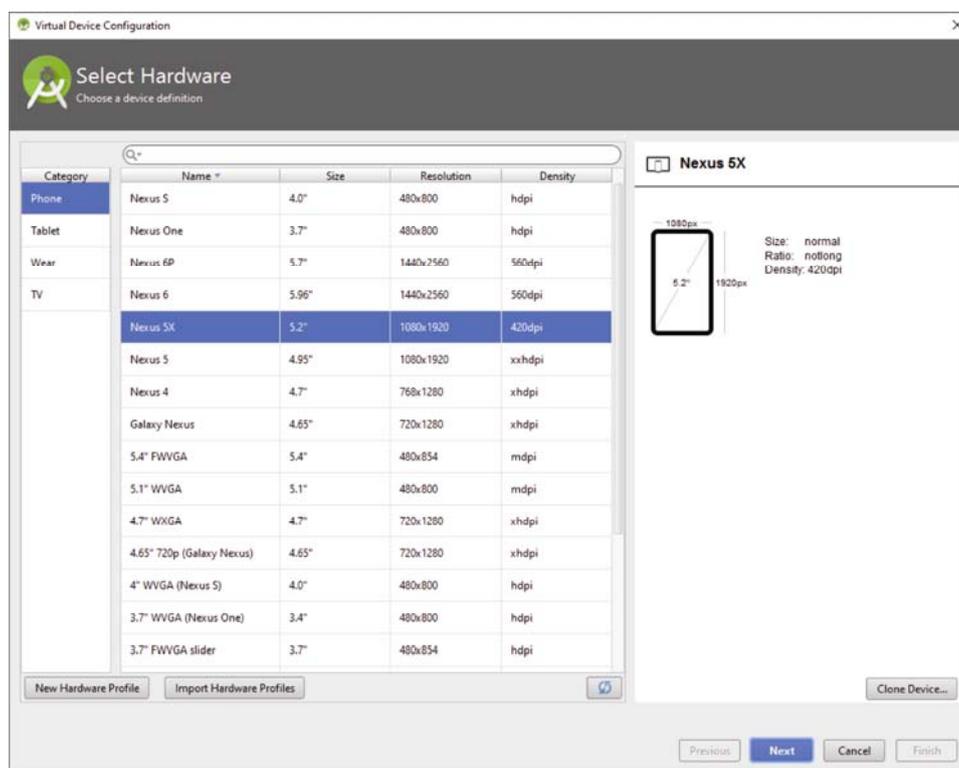


图 1-24

(9) 选择 Nexus 5x 硬件配置文件并且单击 Next 按钮。虽然没有一个模拟器可以提供与其真实硬件一样的性能，但 Nexus 5x 在大多数 x86 平台上应该还是可以正常运行的，并且

它仍然能够提供中端到高端 Android 设备的硬件配置。

(10) 对于系统镜像，选择并安装最新的选项，当写作本书时，最新的选项是 Android Nougat。单击 x86 Images 选项卡(如图 1-25 所示)，在镜像列表中选择 N，并且单击 Next 按钮。

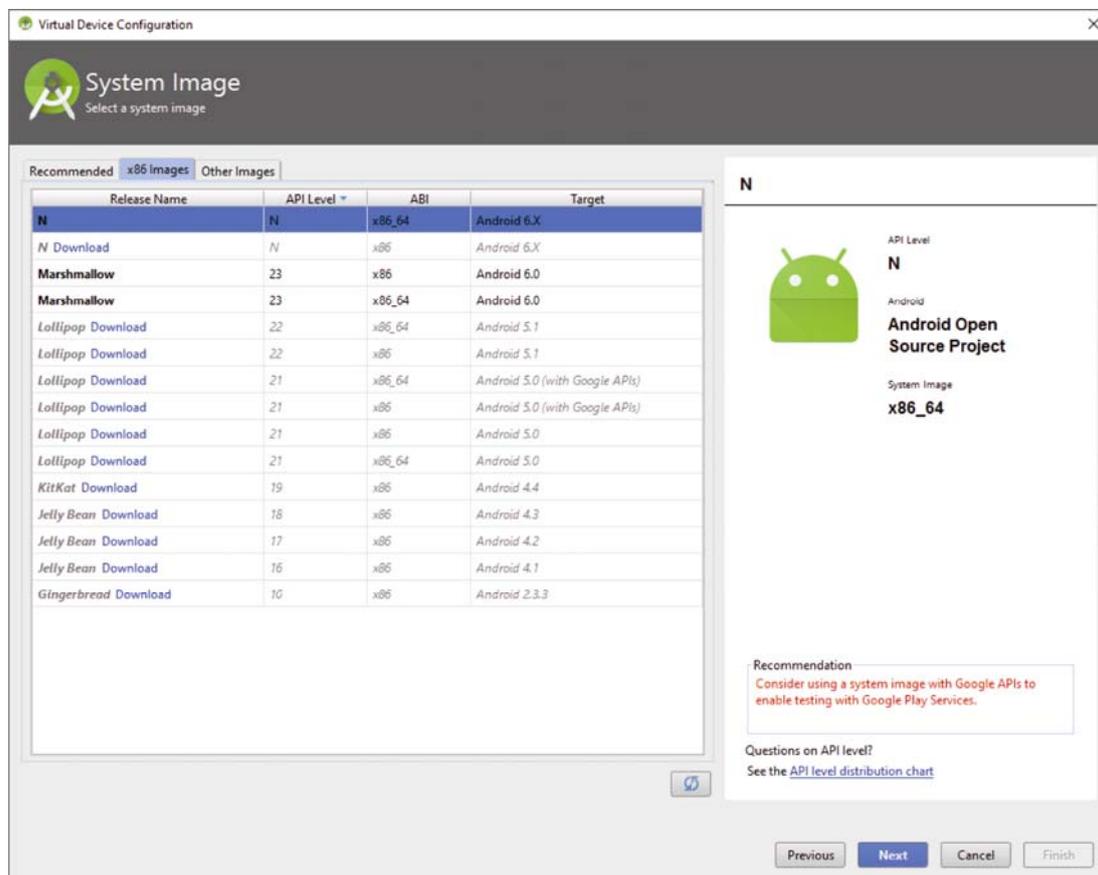


图 1-25

(11) 在 Android Virtual Device(ADV)对话框中，如图 1-26 所示接受默认值。单击 Finish 按钮开始创建 AVD。

**提示：**创建几个使用不同 API 版本和硬件配置的 AVD，在不同版本的 Android 操作系统上测试你的应用是一个不错的选择。

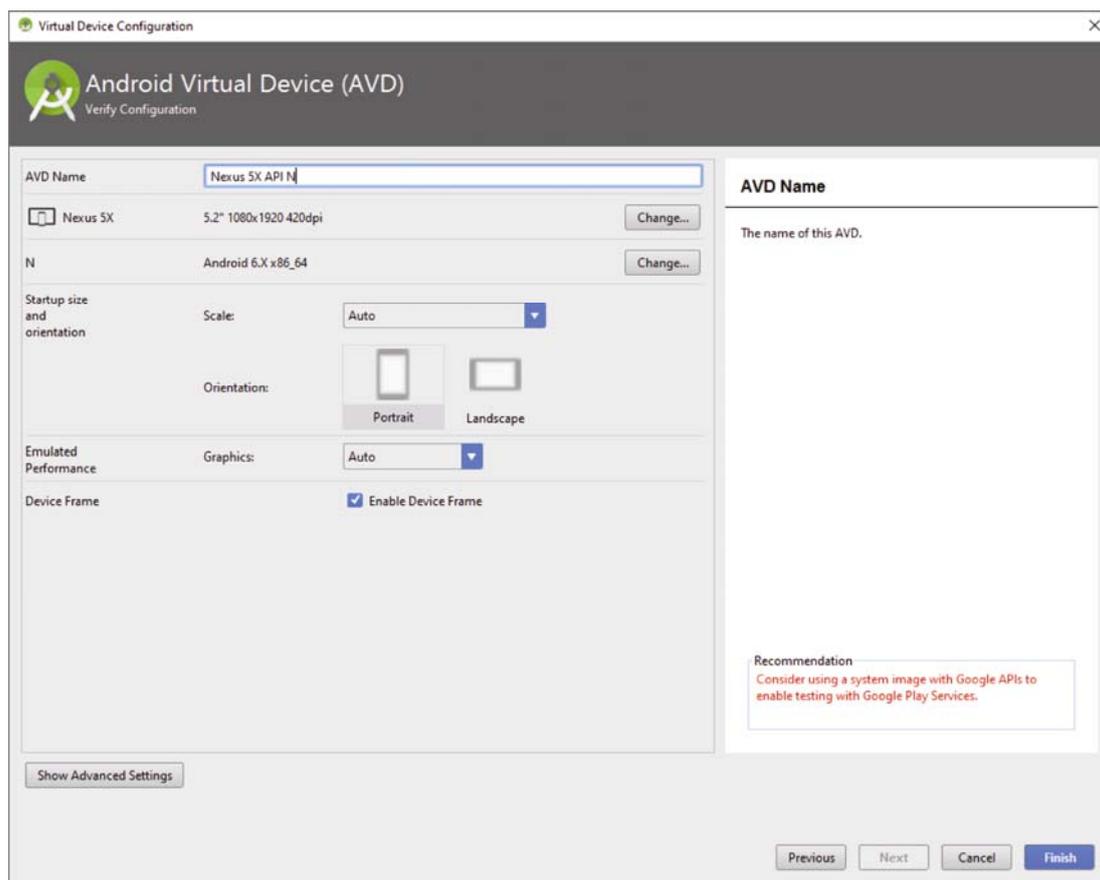


图 1-26

### 试一试 新建一个 JellyBean 模拟器

在本节的前面创建第一个 Android 项目时，创建进程确定 Jelly Bean 是在 Google Play 中最活跃的 Android 版本。在这个“试一试”练习中，你将手动创建一个 Android Jelly Bean 的 AVD。

(1) 选择 Tools | Android | AVD Manager 或者单击工具栏中的 AVD Manager 按钮启动 AVD Manager。

(2) 在 Android Virtual Device Manager Wizard 界面单击+ Create Virtual Device 按钮。

(3) 选择 Nexus 5x 硬件配置文件并单击 Next 按钮。

(4) 单击 x86 Images 选项卡，从镜像列表中选择 Jelly Bean，并且单击 Download 按钮。

(5) 接受用户许可并下载 Jelly Bean SDK。

(6) 当 SDK 下载完毕后，(在 x86 Images 选项卡中)再次单击 Jelly Bean 并单击 Next 按钮。

(7) 在 Android Virtual Device(AVD)对话框中，接受所有的默认值并单击 Finish 按钮。

创建 AVD 以后，是时候测试它了。没有比创建并运行 Hello World 应用更好的方法来完成这个任务了。

## 1.2.4 Android 开发者社区

现在已经是 Android 的第 7 个版本了，在全世界的范围内有一个很大的 Android 开发者社区。寻找问题的答案或是寻找喜欢分享应用想法和经验的开发者都是非常方便的事。

以下是几个开发者社区和网站，当你碰到和 Android 相关的问题时可以在这里寻找帮助：

- **Stack Overflow**([www.stackoverflow.com](http://www.stackoverflow.com)) ——Stack Overflow 是一个面向开发者的协作编辑问答网站。当你有和 Android 相关的问题时，也许有人已经在 Stack Overflow 上讨论过相同的问题了。或是有人已经提供了答案。更好的是，其他开发者可以为最好的答案投票，这样就可以知道哪个答案是最可靠的。
- **Google Android Training**(<http://developer.android.com/training/index.html>) —— Google 发布了一个 Android 培训网站，包含了大量有用的以主题分组的课程。当写作本书时，几乎所有课程都包含了代码片段。这些代码对刚开始学习基础 Android 的开发者非常有帮助。当学习完本书中的基础知识后，强烈建议你去看看那些课程。
- **Android Discuss**(<http://groups.google.com/group/android-discuss>) ——Android Discuss 是一个 Google 支持的使用 Google Groups 服务的讨论组。这里，你可以讨论和 Android 编程相关的各个方面。这个讨论组由 Google 的 Android 团队维护，所以这里是一个解决问题或者学习提示和技巧的好地方。

## 1.3 运行你的第一个 Android 应用

当所有的工具和 SDK 下载和安装好以后，是时候开始运行你的第一个 Android 应用了。和其他编程书籍一样，第一个示例便是无处不在的 Hello World 应用。通过这个示例可以更详细地了解组成 Android 项目的各种组件。这也将是你所编写的最简单的 Android 应用。

无论你是否相信，Hello World 应用其实已经完成了。默认情况下，在 Android Studio 中创建一个新的应用时，它会创建一个 Hello World 应用。让我们运行这个应用，看看它是如何工作的，在这个过程中，也会启动 Android 模拟器。

(1) 从 Android Studio 的菜单栏中选择 Run | Run App，将会看到 Select Deployment Target 对话框。如图 1-27 所示。

(2) 选择 Nexus 5X API N(也可以选择 Nexus 5X API 18，这是你在上一节的练习中创建的 Jelly Bean 模拟器)，并且单击 Next 按钮。

**注意：**如果此时还没有创建模拟器，可以创建一个。

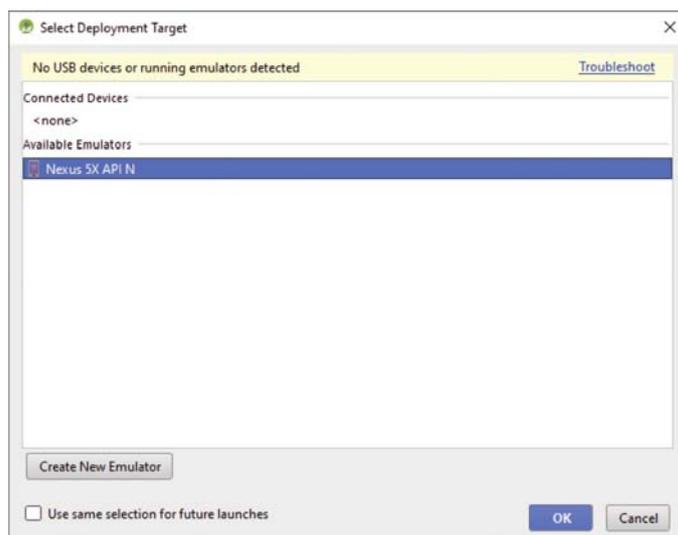


图 1-27

(3) 模拟器从开始到完全加载可能会持续 5 分钟或者更长时间(这取决于你的桌面电脑的硬件配置)。在这个过程中(第一次启动模拟器时),应用可能会超时。如果在 Android Studio 中跳出一个消息告诉你应用在等待 ADB(Android Debugging Bridge, Android 调试桥)启动时超时,或者类似的消息,只需要等待模拟器完全加载完成以后,再次从 Android Studio 菜单栏中选择 Run | Run app。

当模拟器完全加载并启动以后,Android Studio 将安装 Hello World 应用。应用会如图 1-28 显示出来。



图 1-28

这是一个非常快速的示例，教你如何创建和启动第一个 Android 应用。然而，从某种意义上讲，这个示例介绍了大部分主要技能，在本书中你将不断使用这些技能。

## 1.4 小结

本章概述了 Android 并重点介绍了它的一些功能。如果按照下载工具和 Android SDK 章节中的内容进行操作，你现在应该有了一个工作系统——可以开发比 Hello World 更加有意思的 Android 应用。在第 2 章中，你将在学习更加复杂的 Android 应用开发概念之前了解 Android Studio 的内部工作机制。

### 练习

- (1) 什么是 AVD?
- (2) 为什么在 Targeted Android Devices 对话框中 Jelly Bean 是被默认选中的?
- (3) SDK 指的是什么?
- (4) 可以用什么工具下载最新的 Android SDK?

你可以在附录中找到相应的答案。

### 本章中的知识要点

主题	主要概念
Android 操作系统	Android 是基于 Linux 操作系统的开源移动操作系统。任何人都可以在自己的设备上使用它
开发 Android 应用的编程语言	使用 Java 编程语言开发 Android 应用。完成的应用被编译成 Dalvik 可执行文件，在 Dalvik 虚拟机上运行
Google Play	Google Play 应用商店包含了所有由第三方开发者编写的 Android 应用
开发 Android 应用的工具	Android Studio、Android SDK 和虚拟设备