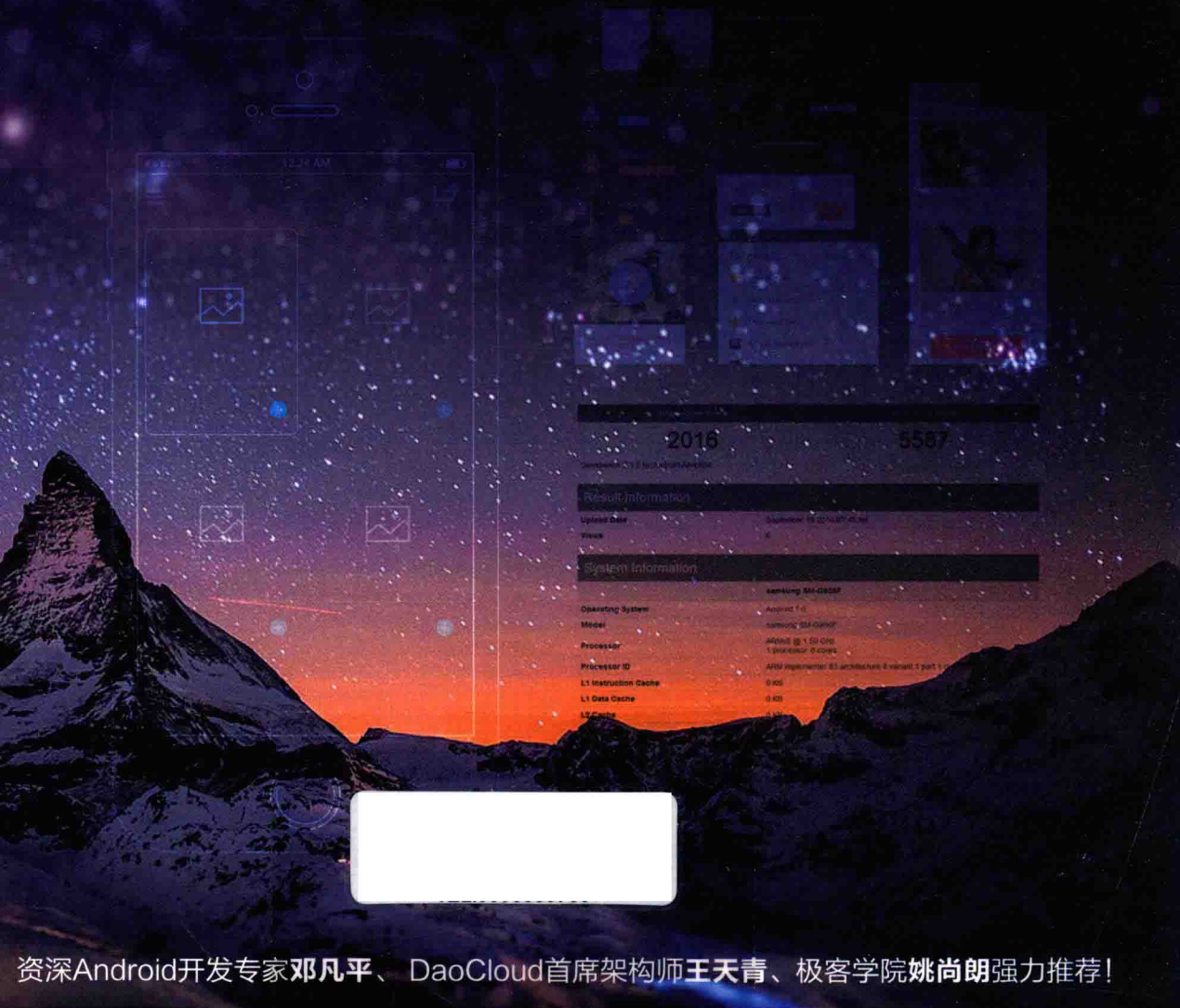


CSDN博客专家、极客学院Wiki专栏作家最新力作
基于最新**安卓7.0**和**Android Studio 2.2**



资深Android开发专家**邓凡平**、DaoCloud首席架构师**王天青**、极客学院**姚尚朗**强力推荐！

程序员
典藏

Android

开发入门百战经典

张亚运◎著



清华大学出版社



Android

开发入门百战经典

张亚运◎著



清华大学出版社
北京

内容简介

本书内容系统全面,采用层层递进的方式进行讲解,让读者理解起来更为容易。全书分为10章,主要包括Android Studio的常用操作和技巧、Android的属性和布局、Android的基础控件、Android的系统组件、Android几种常用的数据存储方式、Android动画、Android网络、Android手机的基本功能及多媒体操作等。

另外,本书还创新地引入了扫描二维码查看动态图的功能,让纸质图书也能和读者交互起来,提升阅读的乐趣。本书适用于广大初、中级Android开发者。对于初级开发者,本书对常用核心的基础知识通过实例的形式进行了系统的讲解,保证初学者学习后可迅速上手进行Android应用开发;对于中级开发者,本书有助于查缺补漏、夯实基础。另外,本书还可以作为高等学校电子信息类专业和计算机类专业本科生的教材以及Android应用开发技术人员的参考书。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

Android开发入门百战经典 / 张亚运著. —北京:清华大学出版社, 2017

(程序员典藏)

ISBN 978-7-302-47345-9

I ①A… II. ①张… III. ①移动终端—应用程序—程序设计 IV. ①TN929.53

中国版本图书馆CIP数据核字(2017)第109645号

责任编辑:张敏

封面设计:杨玉兰

责任校对:徐俊伟

责任印制:沈露

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址: 北京清华大学学研大厦A座 邮 编: 100084

社总机: 010-62770175 邮 购: 010-62786544

投稿与读者服务: 010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn

印装者: 三河市中晟雅豪印务有限公司

经 销: 全国新华书店

开 本: 185mm×260mm 印 张: 22.75 字 数: 554千字

版 次: 2017年8月第1版 印 次: 2017年8月第1次印刷

印 数: 1~3000

定 价: 69.80元

产品编号: 072883-01

目

录

第1章 认识 Android 001

- 1.1 Android 系统 001
 - 1.1.1 Android 的系统架构 001
 - 1.1.2 Android 的历史 002
 - 1.1.3 Android 系统的优势 002
- 1.2 Android Studio 安装 004
 - 1.2.1 Android Studio 安装 004
 - 1.2.2 SDK 更新 005
- 1.3 第一个 Android 项目 005
 - 1.3.1 创建一个新项目 005
 - 1.3.2 创建 Android 模拟器 007

第2章 Android Studio 使用技巧 ... 010

- 2.1 Android Studio 基本配置 010
 - 2.1.1 改变主题 010
 - 2.1.2 改变字体大小和样式 011
 - 2.1.3 改变 Logcat 窗口字体、主题 013
 - 2.1.4 显示行号 015
 - 2.1.5 自动导包 016
- 2.2 Android Studio 常用快捷键 016
 - 2.2.1 Ctrl 组合快捷键 016
 - 2.2.2 Ctrl+Alt 组合快捷键 020
 - 2.2.3 Ctrl+Shift 组合快捷键 022
 - 2.2.4 其他组合快捷键 024
- 2.3 Android Studio 调试 026
 - 2.3.1 Logcat 调试 026

- 2.3.2 断点调试 027

- 2.3.3 高级调试 029

第3章 Android 属性和布局 032

- 3.1 Android 项目文件结构 032
 - 3.1.1 布局属性 032
 - 3.1.2 配置属性 034
 - 3.1.3 其他文件 036
- 3.2 Android 布局属性值 037
 - 3.2.1 Android padding 属性用法 038
 - 3.2.2 Android margin 属性用法 038
- 3.3 Android 布局之线性布局——LinearLayout 041
 - 3.3.1 LinearLayout 基础用法 041
 - 3.3.2 LinearLayout 嵌套 043
- 3.4 Android 线性布局的重要属性 045
 - 3.4.1 gravity 属性 045
 - 3.4.2 layout_weight 属性 048
 - 3.4.3 weightSum 属性 052
- 3.5 Android 布局之相对布局——RelativeLayout 053
- 3.6 Android 布局之帧布局——FrameLayout 056
- 3.7 Android 布局优化 059
 - 3.7.1 过度绘制 059
 - 3.7.2 布局优化之 include 标签 061

第4章 Android 基础控件操作

实战 064

- 4.1 炫酷之星——TextView 控件 064
 - 4.1.1 常用属性介绍 064
 - 4.1.2 TextView 实战演练 064
- 4.2 用户之窗——EditText 控件 069
 - 4.2.1 常用属性介绍 069
 - 4.2.2 EditText 实战演练 070
 - 4.2.3 EditText 实战进阶 074
- 4.3 交互之王——Button 控件 078
 - 4.3.1 Button 单击事件响应 078
 - 4.3.2 clickable 属性设置无效
分析 084
 - 4.3.3 Button 实战进阶 086
- 4.4 执行中的指示器——ProgressBar 088
 - 4.4.1 ProgressBar 样例 088
 - 4.4.2 ProgressBar 基础用法 089
 - 4.4.3 ProgressBar 模拟下载 093
- 4.5 对话框之父——Dialog 095
 - 4.5.1 AlertDialog 096
 - 4.5.2 单选和多选对话框 099
 - 4.5.3 ProgressDialog 进度
对话框 101
 - 4.5.4 定制对话框 104

第5章 Android 控件进阶操作

实战 108

- 5.1 控之经典——ListView 108
 - 5.1.1 ArrayAdapter 适配器 109
 - 5.1.2 SimpleAdapter 适配器 110
 - 5.1.3 BaseAdapter 适配器 112
- 5.2 控之经典——ListView 进阶 116
- 5.3 控之经典——GridView 121
- 5.4 控之经典——GridView 进阶 126
 - 5.4.1 GridView 动态图删除
子项 126
 - 5.4.2 GridView 动态图增加
子项 130

5.5 新控件——RecyclerView 控件 132

- 5.5.1 RecyclerView 线性布局 134
- 5.5.2 RecyclerView 网格布局 137
- 5.5.3 RecyclerView 瀑布流布局 137

5.6 多页面切换器——ViewPager

- 控件 139
- 5.6.1 ViewPager 的基本用法 140
- 5.6.2 ViewPager 导航条 143

第6章 Android 系统组件操作

实战 148

- 6.1 Activity 生命周期 148
- 6.2 指向器——Intent 152
- 6.3 指向器——Intent 隐式启动方式 156
- 6.4 Mini 型 Activity——Fragment 160
 - 6.4.1 静态方式 160
 - 6.4.2 动态方式 162
- 6.5 Mini 型 Activity——Fragment
生命周期 166
- 6.6 FragmentPagerAdapter&FragmentState
PagerAdapter 172
 - 6.6.1 FragmentPagerAdapter
实现页面切换 173
 - 6.6.2 FragmentStatePagerAdapter
实现页面切换 176
- 6.7 Android 广播接收器之
BroadcastReceiver 179
 - 6.7.1 静态注册
BroadcastReceiver 179
 - 6.7.2 动态注册
BroadcastReceiver 180
 - 6.7.3 广播接收器 BroadcastReceiver
实用实例 182
- 6.8 Android 自定义广播 Broadcast 186
 - 6.8.1 普通广播发送和接收
实例 187
 - 6.8.2 有序广播发送和接收
实例 188

6.9	Android Service——startService 和 bindService	191	8.4	Android 属性动画—— ObjectAnimator	252
6.9.1	startService 启动服务	192	8.4.1	属性动画与传统动画 的区别	252
6.9.2	bindService 启动服务	195	8.4.2	旋转动画	254
第 7 章 Android 存储操作实战 200			8.4.3	尺寸动画	254
7.1	轻型存储器—— SharedPreferences	200	8.4.4	渐变动画	254
7.1.1	SharedPreferences 基本用法	200	8.4.5	XML 方式实现属性动画 ..	257
7.1.2	SharedPreferences 实现自动登录功能	204	8.5	Android 属性动画—— ValueAnimator	259
7.2	Android 数据库 SQLite	209	8.6	Android 属性动画集	262
7.2.1	SQLiteOpenHelper 类	210	8.6.1	简单的组合方式	262
7.2.2	SQLiteDatabase 类	212	8.6.2	PropertyValuesHolder 方式	264
7.3	数据中心——ContentProvider	218	8.6.3	AnimatorSet 方式	265
第 8 章 Android 动画操作实战 223			8.7	Android 属性动画实现浮动菜单 ..	266
8.1	Android 传统动画—— Tween (补间动画)	223	第 9 章 Android 网络操作实战 271		
8.1.1	AlphaAnimation—— 渐变动画	224	9.1	Android 网络核心控件 WebView -	271
8.1.2	RotateAnimation—— 旋转动画	229	9.1.1	简单的 WebView	271
8.1.3	ScaleAnimation—— 尺寸动画	232	9.1.2	丰富 WebView 功能	273
8.1.4	TranslateAnimation—— 位移动画	235	9.2	WebView 滚动事件	276
8.2	Android 传统动画进阶	238	9.2.1	WebView 滚动监听 的实现	276
8.2.1	动画插值器 Interpolator	238	9.2.2	WebView 一键回到 顶部功能实现	278
8.2.2	动画监听器 AnimationListener	241	9.2.3	WebView 退出记忆 功能实现	280
8.2.3	动画集 AnimationSet	243	9.2.4	WebView 联合滚动实现 ..	281
8.2.4	LayoutAnimationController 组件动画	246	9.3	网络连接类—— URLConnection	283
8.3	Android 传统动画——Frame Animation (帧动画)	248	9.3.1	URLConnection 打印网页	284
			9.3.2	URLConnection 下载图片	287
			9.3.3	URLConnection 保存图片	290
			9.4	Android Handler 消息处理机制 ..	294

9.4.1 消息类 Message	295	第 10 章 Android 手机基本功能	
9.4.2 消息处理类 Handler	295	及多媒体操作实战	315
9.4.3 Handler 实现倒计时功能 ..	298	10.1 Android 拨打电话功能实例	315
9.4.4 Handler 延迟操作	301	10.2 Android 发送短信功能实例	319
9.4.5 Handler postDelay 实现循环调用	303	10.2.1 直接发送短信	320
9.4.6 Looper 用法	304	10.2.2 跳转到短信发送界面	322
9.5 Android 异步操作类 AsyncTask ..	307	10.3 Android 播放音乐功能实例	323
9.5.1 AsyncTask 基本用法	308	10.4 Android 播放视频功能实例	329
9.5.2 AsyncTask 实用实例	310	10.5 Android 录制音频功能实例	335
		10.6 Android 拍照功能实例	341
		10.6.1 Intent 方式	342
		10.6.2 借助 Camera 类	344

第 1 章 认识 Android

1.1 Android 系统

Android 是谷歌推出的基于 Linux 的手机平台，作为开源的移动操作系统，不存在任何阻碍移动产业创新和发展的专利权障碍，因此 Android 一经面世就获得了空前的发展，在移动操作系统市场份额一度超过 80%，处于绝对的垄断地位。

现如今 Android 已经不局限于手机系统，越来越多的车载、穿戴、电视设备也集成了 Android 系统。我们相信随着物联网的不断深入和发展，Android 系统将会以更多样的形式融入到我们的生活、学习和工作之中。因此，学习 Android 不会过时，正当其时！

1.1.1 Android 的系统架构

Android 的系统架构，如图 1.1 所示。

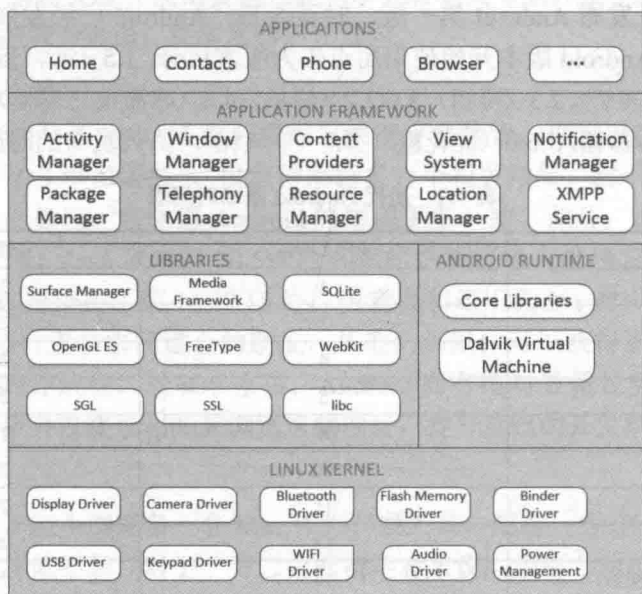


图 1.1 Android 的系统架构

和其他操作系统一样，Android 的系统架构也采用了分层的结构。从架构图来看，Android 分为四个层次，下面分别总结这几个层。

1. 应用程序 (Applications)

Android 应用程序的源程序除了包含 Java 代码之外，还包含各种资源文件（放置于 res 目录中）、将源程序进行编译可以得到一个 APK 安装包，这个安装包可以安装到 Android 手机上，将对应一个 Android 应用程序。Android 软件开发人员可以使用应用程序框架层提供的 API 快速开发 Android 应用，这也是 Android 的巨大潜力所在。

2. 应用程序框架 (Application Framework)

Android 的应用程序框架层有供 Android 开发者所调用的丰富 API，该层实际上就是一个应用程序的框架。框架中不仅包含各种 API，同时还内置了方便开发者开发的各种控件，例如 Views (视图组件)，其中又包含了 List (列表)、Grid (栅格)、Button (按钮)、TextView (文本框) 等，甚至还内置了一个浏览器。有了这些基本控件，开发者可以更快地构建应用程序，提高开发效率。

3. 各种库 (Libraries) 和 Android 运行环境 (Android Runtime)

对应一般的嵌入式操作系统，本层相当于中间件层次。Android 中的本层分成两个部分，一个是各种库 (Libraries)，另一个是 Android 运行环境。本层的大多数代码是由 C 和 C++ 语言实现。Android 运行环境指的是 Android 虚拟机技术 Dalvik。

4. 操作系统层 (OS)

Android 基于 Linux，使用的是 Linux 2.6 操作系统，以它作为底层。Android 对操作系统的使用包括了核心和驱动程序两个部分，其中驱动程序有显示驱动、蓝牙驱动、相机驱动、网络驱动和各种传感器设备驱动等。

1.1.2 Android 的历史

自 2008 年 9 月发布 Android 第一版，时至今日，Android 已经发展到了 7.0 时代。从 2009 年 5 月开始，Android 版本开始使用甜点作为版本代号：1.5 (纸杯蛋糕)、1.6 (甜甜圈)、2.0 (泡芙)、2.2 (冻酸奶)、2.3 (姜饼)、3.0/3.2 (蜂巢)、4.0 (冰激凌三明治)、4.1/4.2 (果冻豆)、4.4 (奇巧)、5.0 (棒棒糖)、6.0 (棉花糖)、7.0 (牛轧糖)，历代发布的时间请参考表 1.1。

表 1.1 历代 Android 版本时间表

版 本 号	API Level	发 布 时 间
1.1	1	2008 年 9 月
1.5 (纸杯蛋糕)	3	2009 年 4 月
1.6 (甜甜圈)	4	2009 年 9 月
2.0 (泡芙)	5	2009 年 10 月
2.2 (冻酸奶)	8	2010 年 5 月
2.3 (姜饼)	10	2010 年 12 月
3.0 (蜂巢)	11	2011 年 2 月
3.2 (蜂巢)	13	2011 年 7 月
4.0 (冰激凌三明治)	14	2011 年 10 月
4.1 (果冻豆)	16	2012 年 6 月
4.2 (果冻豆)	17	2012 年 10 月
4.4 (奇巧)	19	2013 年 9 月
5.0 (棒棒糖)	21	2014 年 11 月
6.0 (棉花糖)	23	2015 年 5 月
7.0 (牛轧糖)	24	2016 年 8 月

1.1.3 Android 系统的优势

对于想要从事移动开发的读者来讲，开始都会万分纠结的问题就是到底该学习 iOS 还

是 Android 系统, 对于一些初学者来说, 这一定是一个单选题。本书主要对 Android 知识进行讲解, 当然要自卖自夸一下。比较上述两系统而言, 学习 Android 系统可以有如下优势:

1. 更容易上手

对于初学者来说, 最缺乏的就是基础知识, 最渴望的就是快速上手, 最苦恼的莫过于一头雾水。Android 系统使用 Java 语言进行开发, 对计算机语言稍有基础的同学而言, Java 语言都不会陌生, Java 语言也是常年霸占计算机语言流行榜 No.1 的位置, 国内 Java 语言学习风气浓厚, 随便百度一下即可获得海量 Java 语言学习资源。同时 Java 语言也是以其简单、易用而闻名, 所以对于初学者来说, 这第一个骨头并不难啃。而 iOS 系统采用 Object-C 进行开发 (2014 年推出了 Swift 作为新的开发语言), 相对 Java 语言来说, 其学习难度要大不少, 此外志同道合者较少, 因此, 学习资料也就相对缺乏, 一起讨论交流的朋友也比较少。因此, 就上手难易程度来说, Android 系统确实优于 iOS 系统。

2. 更宽广的就业方向

即使不了解开发的人都会知道, iOS 是闭源的系统, 开发者除了能开发 iOS 应用什么也做不了, 而 Android 是开放的系统, 源代码公开, 从上层的应用开发、到 Framework 层再到底层驱动都可以进行研究和学习, 任何一个环节、任何一个模块都可以作为今后从业的方向。此外, 学习好 Android 的开发语言 Java, 就算以后不从事移动端开发, 还可以转向 Web 开发等, 而学习 iOS 开发语言就只能从事 iOS 相关专业开发了。因此学习 Android 开发将拥有更宽广的就业渠道, 更丰富的研究方向。

3. 更多的学习资源

在百度搜索中输入“Android 学习资料”关键字并搜索, 你可以获得 8 640 000 个相关结果, 而输入“iOS 学习资料”, 仅获得两百多万个相关结果。对于初学者最好的老师——搜索引擎来说, 显然, 它知道 Android 的知识更多一些。此外, 根据 2016 年 TIOBE 世界编程语言排行榜, Java 语言以 20.5% 占有率的绝对优势占据榜首, 而 iOS 的开发语言 Swift 和 Object-C 则排在了第 14 和 15 位, 两者之和还不到 3% 的占有率。因此, 学习 Android 系统你将拥有更多志同道合的朋友, 从他们那里你可以获得更多帮助和指导。最后, 由于 Android 的开放性, 较 iOS 来讲, Android 拥有绝对数量优势的优秀开源项目, 有一定基础的开发者可以登录 github 浏览这些项目, 提升自己的开发能力。

4. 学习成本

学习 iOS 系统, 至少得配备一台 Mac 作为开发工具, 配备一台 iPhone 作为调试工具, 这两种开发工具都价值不菲, 对于一穷二白的初学者来说, 经济上的拮据是不可避免的问题。对于缺乏定力和恒心而半途而废的初学者来说, 损失就更大了。而学习 Android 系统只需一台具有 Windows 操作系统的电脑就好了 (基本每个人都有), 对于调试工具可以选择模拟器, 也可以花几百元买一台入门级 Android 手机, 所以前期投入很少, 不存在任何风险。因此, 想学习 Android 开发马上就可以开始, 不需要太大的经济投入, 没有经济压力和风险。

综上, 对于踌躇不前、犹豫不决的初学者来说, 何不先选择 Android 系统学习一下呢? 因为它简单、易上手且无须任何前期投入。我也相信, 鉴于 Android 系统的开放性、流行性, 只要尝试, 你肯定会爱上它, 因为作者本人就是这么掉进“陷阱”里来的。

1.2 Android Studio 安装

俗话说：“工欲善其事，必先利其器。”要想获得快速的开发效率和学习速度，选择一样合适的开发工具是首先要做的事情。很长一段时间，开发者都习惯了使用 Eclipse 并结合 ADT 插件来开发 Android 应用，但这一习惯将随着 Android Studio 的不断强大而必须改变了。自从 2013 年 5 月 16 日，在 I/O 大会上推出的 Android Studio 雏形，到现在更新到了最新的 2.2.2.0 版本，Android Studio 越来越稳定，功能也越来越强大，是时候该享受全新的开发工具了。

1.2.1 Android Studio 安装

安装 Android Studio，首先需要下载安装包，这里推荐一个下载地址（Android Studio 中文社区）<http://www.android-studio.org/>。可以看出，当前最新的 Android Studio 已经更新到了 2.2.2.0 版本，单击如图 1.2 所示的“下载”按钮，下载安装包到本地。

双击下载完成的安装包安装 Android Studio，如图 1.3 所示，单击 Next 按钮跳转到下一界面，如图 1.4 所示。



图 1.2 Android Studio 下载



图 1.3 Android Studio 安装一

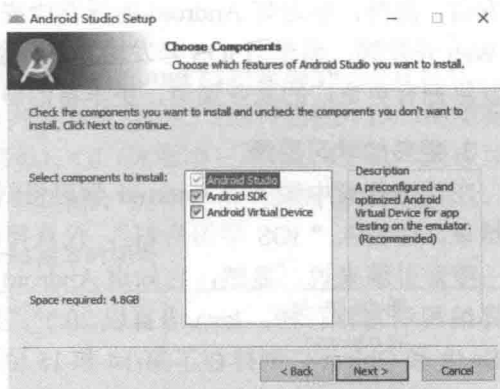


图 1.4 Android Studio 安装二

将默认一起安装 Android SDK 和 Android Virtual Device。不断单击 Next 按钮到安装成功页，如图 1.5 所示。



图 1.5 Android Studio 安装成功页

1.2.2 SDK 更新

单击 Finish 按钮进入 Android Studio, 刚进去可能需要更新 SDK, 如图 1.6 所示。

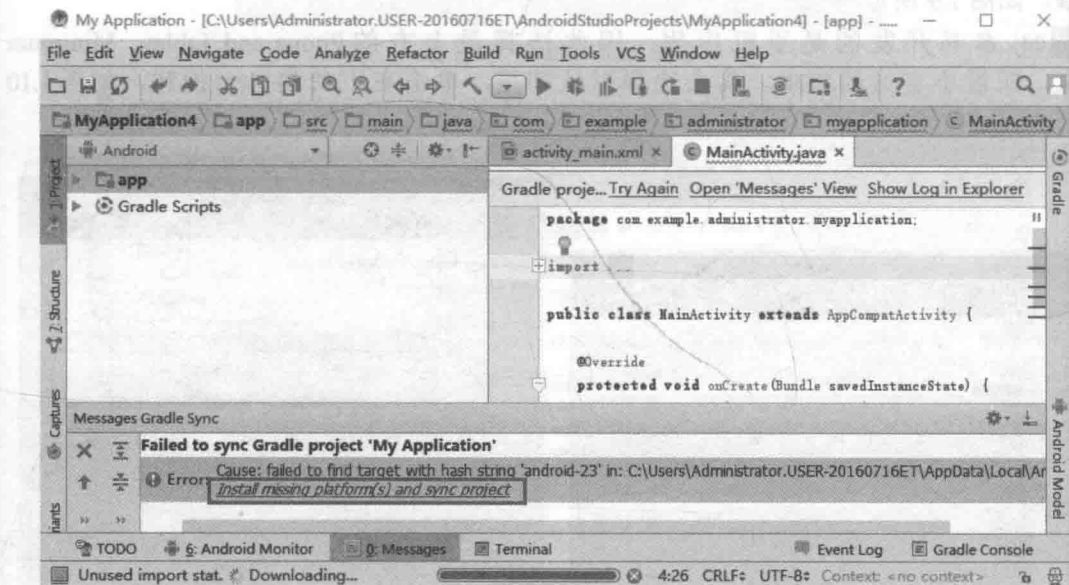


图 1.6 Android Studio 页

单击 Install missing platform(s) and sync project 按钮更新 SDK, 如图 1.7 所示。

单击 Finish 按钮完成 SDK 更新, 届时 Android Studio 安装完毕。

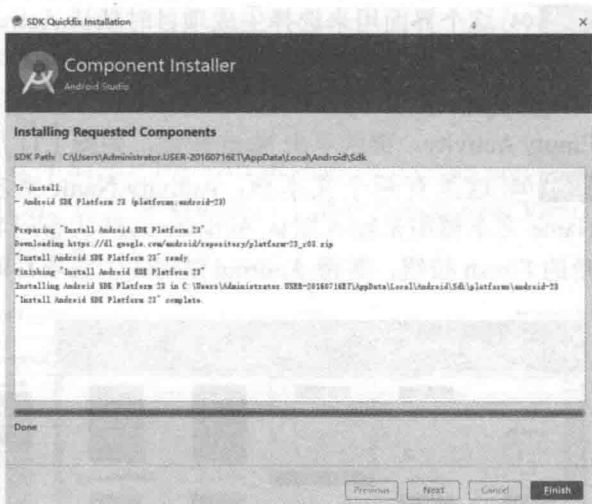


图 1.7 Android Studio 更新 SDK

1.3 第一个 Android 项目

万事开头难, 凡事都有套路, 勇敢迈出第一步就成功一大半了。本节将带领初学者迈出属于自己的一小步。本书的开发工具采用 Android Studio, 因此, 首先通过图文讲解如何使用 Android Studio 新建我们的第一个 Android 项目。

1.3.1 创建一个新项目

安装成功 Android Studio 会默认生成一个 Android 项目, 这里新建一个属于我们自己的项目。

Step 01 选择 File → New → New Project, 如图 1.8 所示。

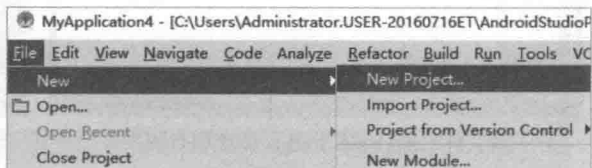


图 1.8 Android Studio 新建项目

Step 02 选择 New Project 后会弹出新建项目页，在 Application name 中输入项目名称，在 Package name 中输入包名（需要单击右边的 Edit 按钮），单击右下角的 Next 按钮进入下一步骤，如图 1.9 所示。

Step 03 本书开发的是手机应用，因此选择最上方的 Phone and Tablet、Minimum SDK，即最小支持的 SDK，其余选择默认即可，单击右下角的 Next 按钮，如图 1.10 所示。

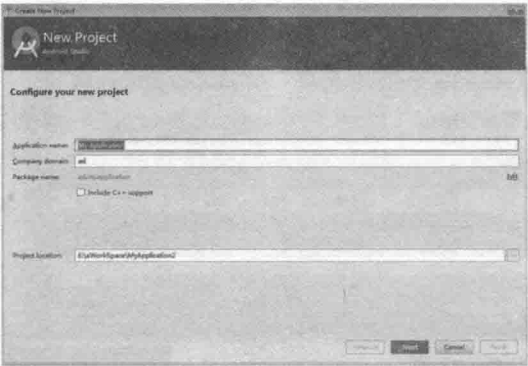


图 1.9 Android Studio 新建项目页一

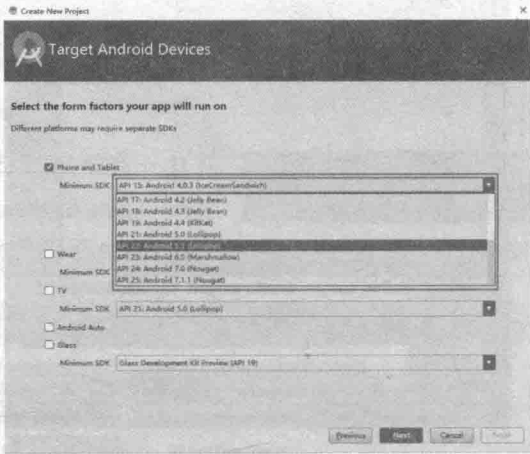


图 1.10 Android Studio 新建项目页二

Step 04 这个界面用来选择生成项目时默认 Activity 的样式，Android Studio 提供了丰富的 Activity 模板供我们选择，有 Basic Activity（基本 Activity）、Empty Activity（空 Activity）、Google Maps Activity（谷歌地图 Activity）、Login Activity（登录 Activity）等，这里选择 Empty Activity，继续单击 Next 按钮，如图 1.11 所示。

Step 05 这里有两个文本框，Activity Name 文本框用来输入默认的 Activity 名，Layout Name 文本框用来输入默认 Activity 的默认布局名称，这里都选择默认的即可，单击右下角的 Finish 按钮，等待 Android Studio 生成项目即可，如图 1.12 所示。

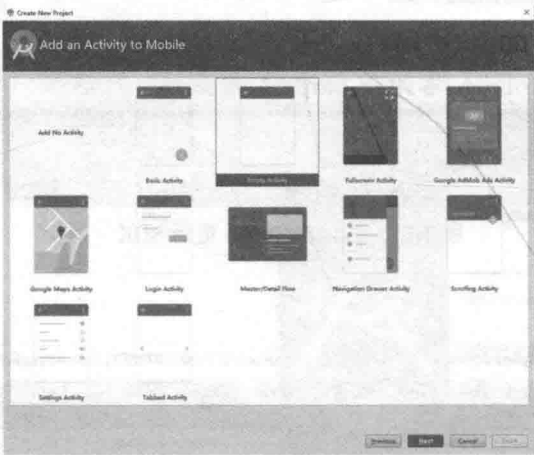


图 1.11 Android Studio 新建项目页三

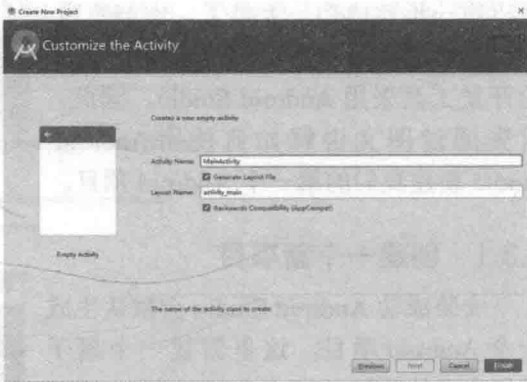


图 1.12 Android Studio 新建项目页四

Step 06 等待 Gradle 编译完成，如图 1.13 所示。生成的 Android Studio 页面如图 1.14 所示。

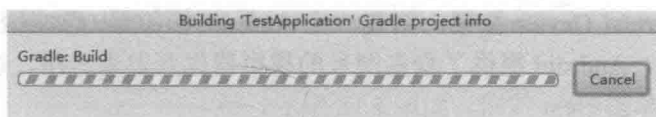


图 1.13 Gradle Build 编译

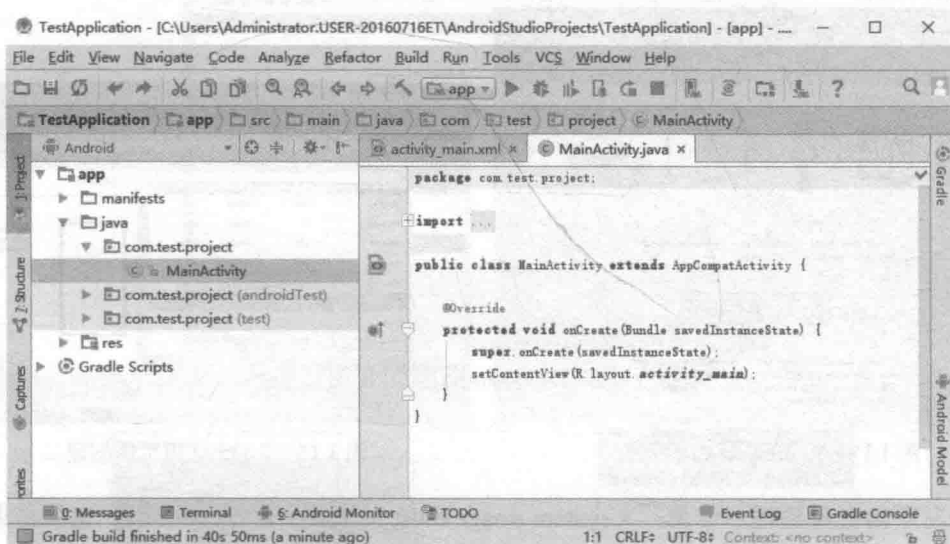


图 1.14 Android Studio 页面

1.3.2 创建 Android 模拟器

单击工具栏中的 Run 按钮，如图 1.15 所示。

弹出 Android 模拟器选择框，此时看到提示：No USB devices or running emulators detected，也就是没有检测到 Android 模拟器，因此需要单击左下角的 Create New Virtual Device 按钮创建一个新的模拟器，如图 1.16 所示。

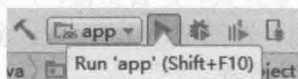


图 1.15 Android Run 按钮

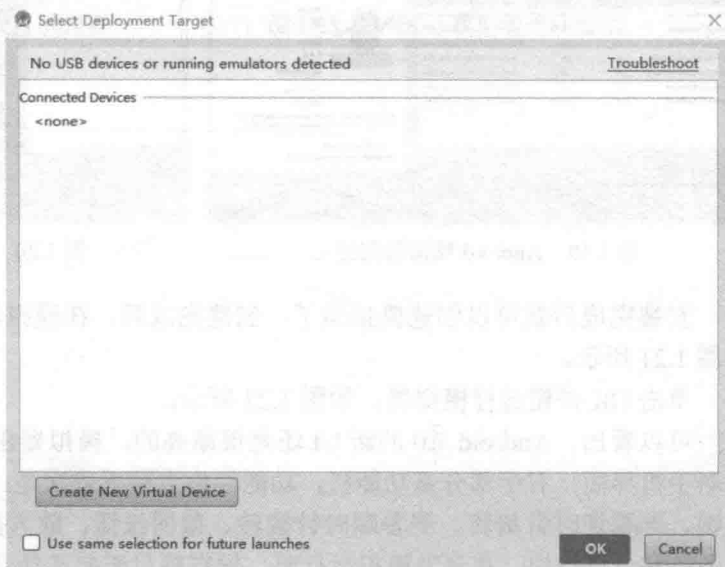


图 1.16 Android 模拟器选择框

单击 Create Virtual Device 按钮，创建一个 Android 模拟器，如图 1.17 所示，这里需要选择手机模拟器，Android 提供了众多型号的模拟器供开发者选用，选择一款你喜欢的手机作为模拟器，单击 Next 按钮，如图 1.18 所示。

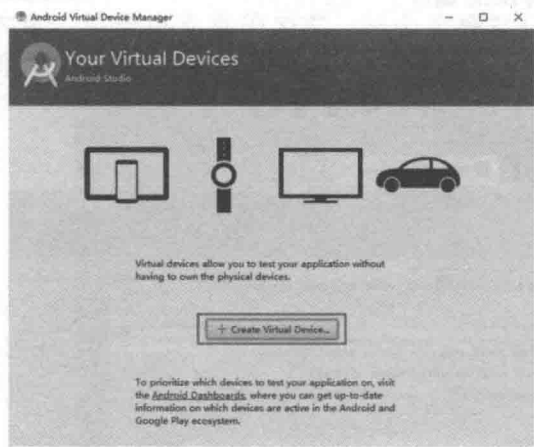


图 1.17 Android 模拟器创建一

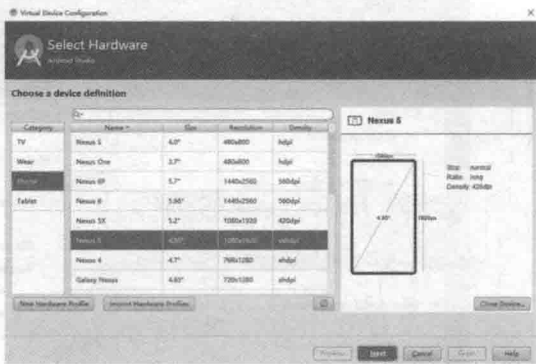


图 1.18 Android 模拟器创建二

在左下角会看到提示信息：A system image must be selected to continue，也就是说必须先安装一个系统镜像，单击 Download 按钮后如图 1.19 所示。单击 Next 按钮安装系统镜像，如图 1.20 所示。

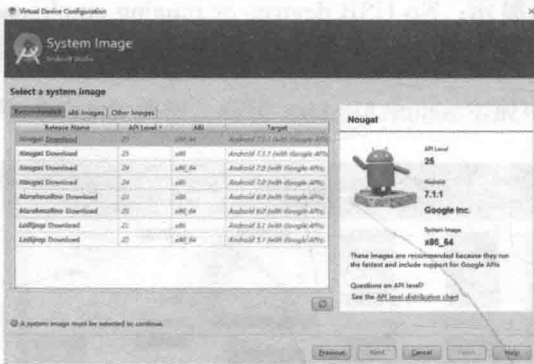


图 1.19 Android 模拟器创建三

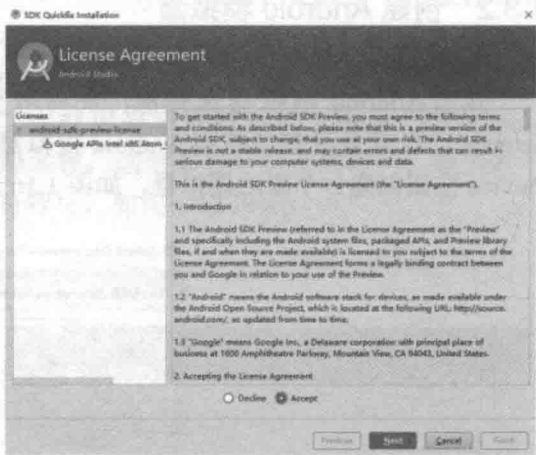


图 1.20 Android 模拟器创建四

安装完成后就可以创建模拟器了，创建完成后，在模拟器列表就出现了新的模拟器，如图 1.21 所示。

单击 OK 按钮运行模拟器，如图 1.22 所示。

可以看出，Android 7.0 的新 UI 还是很漂亮的，模拟器由两部分组成：左半部分是模拟器手机界面；右半部分是功能栏，功能栏由上到下依次是：屏幕开关、声音上键、声音下键、屏幕逆时针旋转、屏幕顺时针旋转、截图按钮、放大按钮、返回键、Home 键、多任务键和设置按钮。在开发模拟运行时，根据项目需要选择合适的操作。

再稍等片刻，TestApplication 项目将运行起来，如图 1.23 所示，程序员们熟悉的

“Hello World!” 在模拟器中显示出来了。

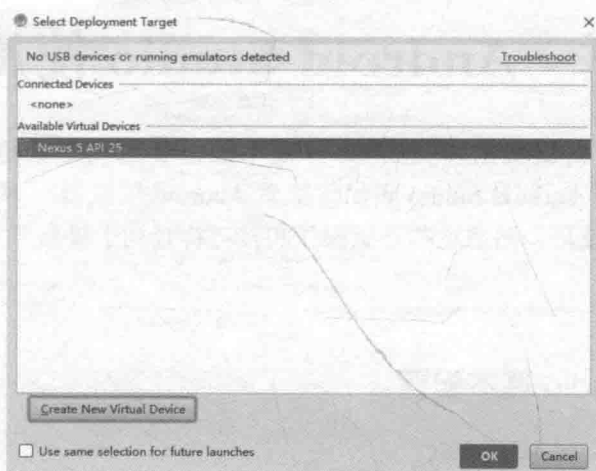


图 1.21 Android 模拟器选择



图 1.22 Android 模拟器



图 1.23 Android 模拟器项目运行

第2章 Android Studio 使用技巧

正所谓“磨刀不误砍柴工”，Android 开发中最重要的利器就是 Android Studio。上一章介绍了如何安装 Android Studio 和如何配置 Android 模拟器。本章主要讲解 Android Studio 的常用操作和技巧，熟悉这些常见操作和技巧将有利于提高开发效率，减少开发时低级错误的发生。

2.1 Android Studio 基本配置

2.1.1 改变主题

安装成功时，Android Studio 默认的主题名为 IntelliJ，其效果如图 2.1 所示。

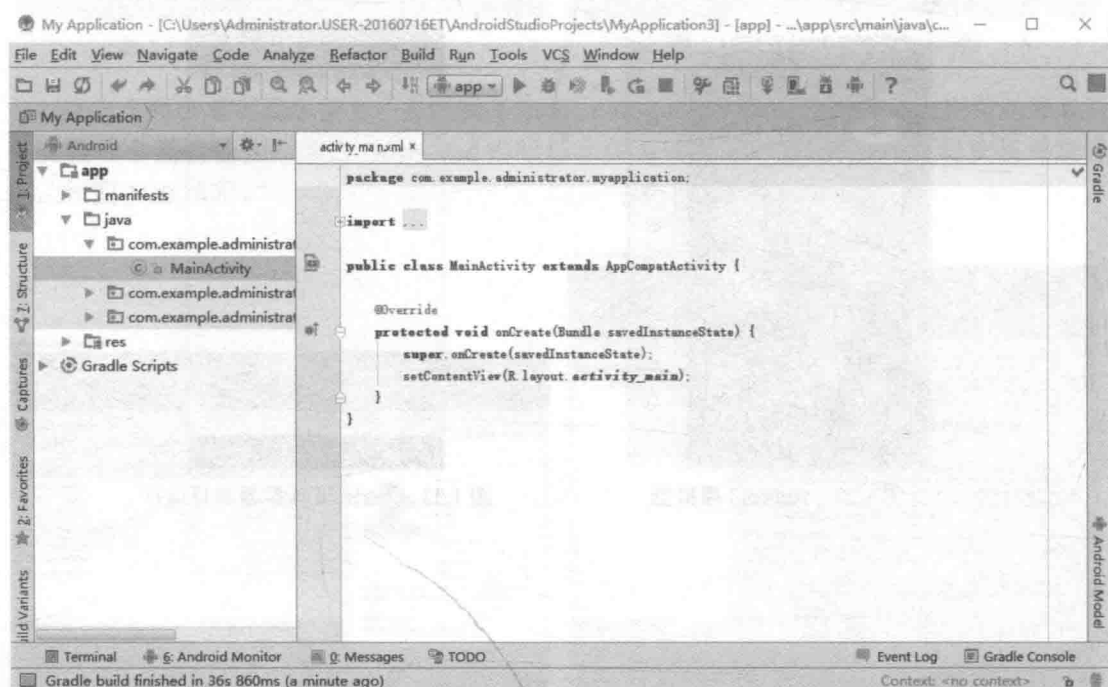


图 2.1 Android Studio 默认主题

这种主题比较亮，对于长时间盯着屏幕的程序员来说，最辛苦的莫过于眼睛了。考虑到这种情况，Android Studio 提供了“护眼模式”Darcule 主题，使用该主题，应按如下几个步骤操作：

- Step 01** 选择 File → Settings，如图 2.2 所示。
- Step 02** 此时将跳转到设置页面，选择 Appearance 标签中的 Theme 值，如图 2.3 所示。
- Step 03** 选择 Darcula 并单击 Apply 按钮即可使用这个主题，如图 2.4 所示。

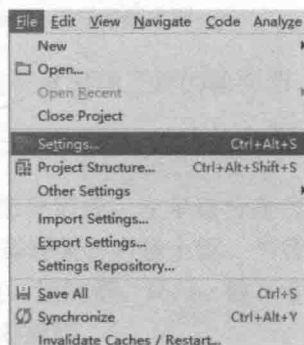


图 2.2 File 菜单栏