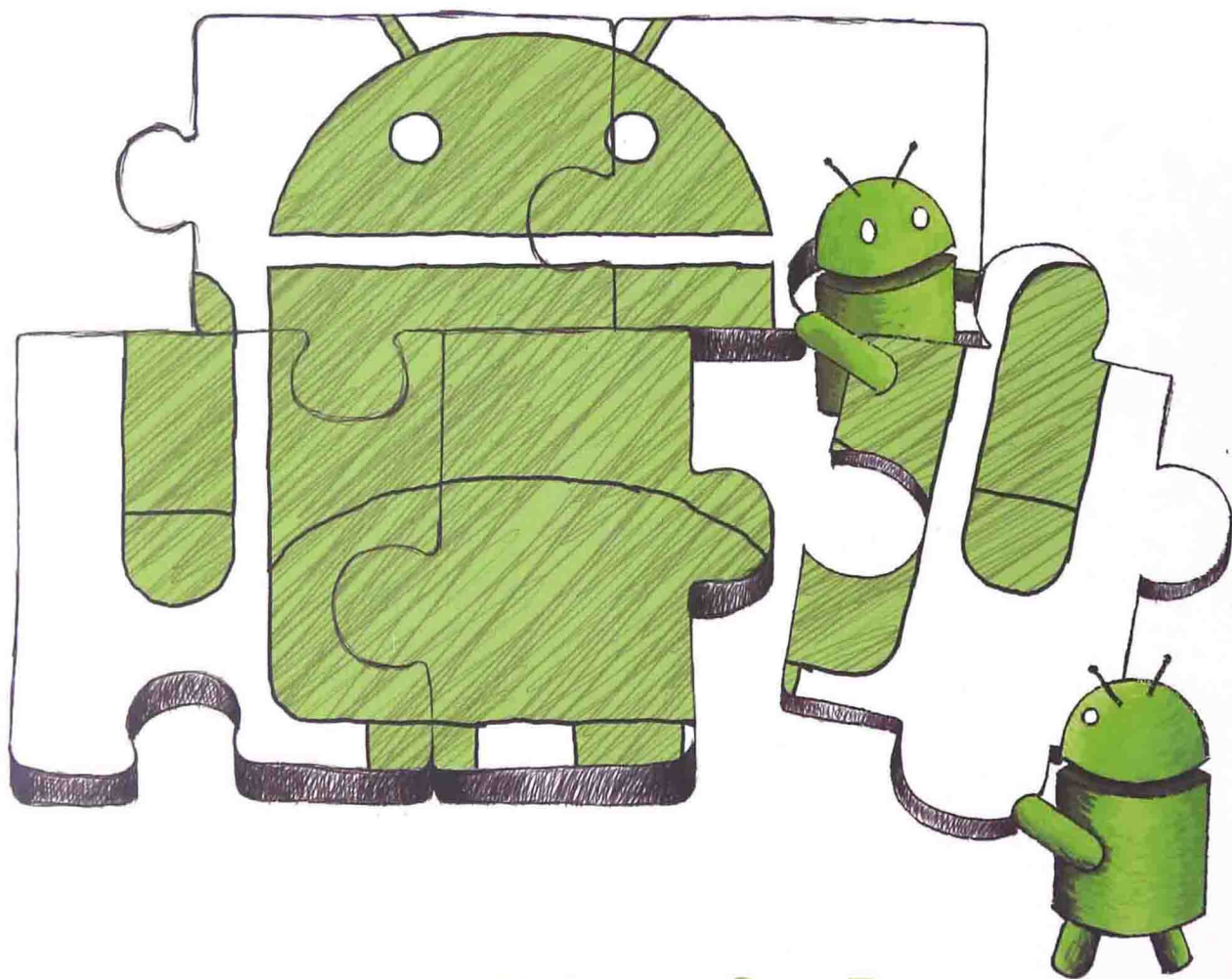


覆盖游戏、多媒体、地图、可穿戴设备、平板应用等领域一百四十余个精彩实例

Broadview®
www.broadview.com.cn



Android

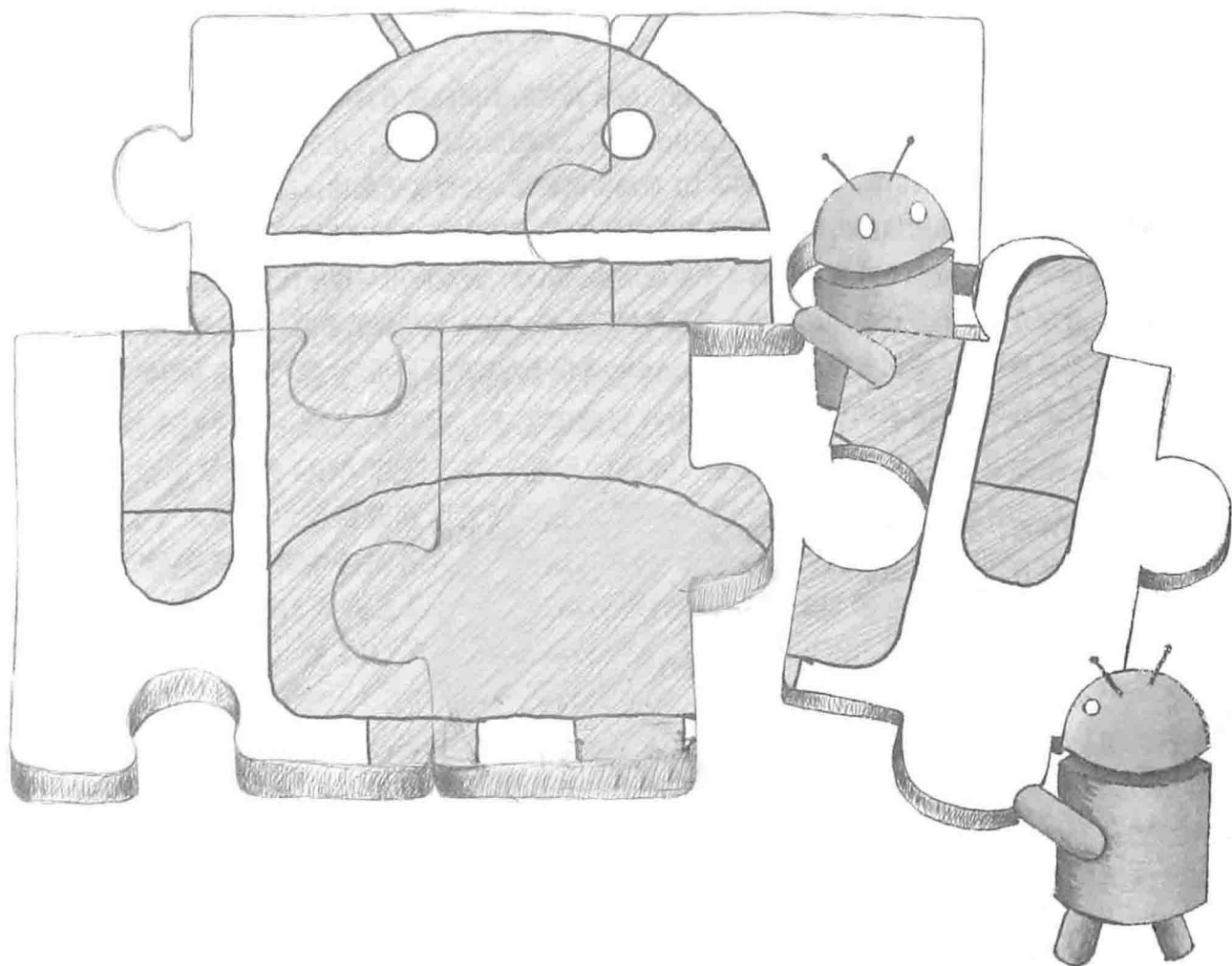
开发实例大全 第2版

王东华 等编著



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

Andro



Android

开发实例大全 (第2版)

王东华 等编著

電子工業出版社
Publishing House of Electronics Industry
北京 · BEIJING

内 容 简 介

本书以 Android 应用程序的开发为主题, 结合真实的案例向读者详细介绍了 Android 基本组件的使用及应用程序开发的整个流程。本书的讲述由浅入深, 实例全面且典型, 几乎囊括了所有和 Android 应用相关的项目。全书分为 18 章, 分别讲解了 UI 布局实例集锦、控件实例集锦、自动化服务实例集锦、数据存储实例集锦、电话和短信实例集锦、图形图像实例集锦、和网络有关的实例集锦、多媒体实例集锦、Google 地图实例集锦、Google API 实例集锦、平板应用实例集锦、游戏应用与程序优化, 最后通过 4 个大型综合实例, 分别介绍了开发手机地图系统、音乐播放器系统、NBA 激情投篮游戏、综合邮件系统的基本过程。

本书基础翔实, 实例丰富、真实, 适合 Android 程序开发的不同用户, 既可以作为初学者的参考资料, 也可以作为向此领域发展的程序员的参考书。

未经许可, 不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有, 侵权必究。

图书在版编目 (CIP) 数据

Android 开发实例大全 / 王东华等编著. —2 版. —北京: 电子工业出版社, 2014.9
(Android 移动开发技术丛书)
ISBN 978-7-121-24177-2

I. ①A… II. ①王… 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 (2014) 第 197382 号

策划编辑: 张月萍

责任编辑: 董 英

印 刷: 北京中新伟业印刷有限公司

装 订: 河北省三河市路通装订厂

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱

邮编: 100036

开 本: 787×1092 1/16

印张: 43.75

字数: 1148 千字

版 次: 2014 年 9 月第 1 版

印 次: 2014 年 9 月第 1 次印刷

印 数: 3000 册 定价: 89.00 元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888。

质量投诉请发邮件至 zlts@phei.com.cn, 盗版侵权举报请发邮件到 dbqq@phei.com.cn。

服务热线: (010) 88258888。

前言

随着 3G 的到来，无线带宽越来越宽，使得更多内容丰富的应用程序装入手机成为可能，如视频通话、视频点播、移动互联网冲浪和内容分享等。为了承载这些数据应用及快速部署，手机功能将会越来越智能，越来越开放。为了实现这些需求，必须有一个好的开发平台来支持，在此由 Google 公司发起的 OHA 联盟走在了业界的前列，2007 年 11 月推出了开放的 Android 平台，任何公司及个人都可以免费获取源代码及开发 SDK。由于其开放性和优异性，Android 平台得到了业界的广泛支持，其中包括各大手机厂商和著名的移动运营商等。继 2008 年 9 月第一款基于 Android 平台的手机 G1 发布之后，三星、摩托罗拉、索爱、LG 等主流手机制造商都推出了自己的 Android 平台手机。在 2011 年底，Android 超越了塞班和 iOS，雄踞智能手机市场占有率榜首的位置。

毕竟 Android 平台被推出的时间才短短 7 年，了解 Android 平台软件开发技术的程序员还不多，如何迅速地推广和普及 Android 平台软件开发技术，让越来越多的人参与到 Android 应用的开发中，是整个产业链都在关注的一个话题。为了帮助开发者更快地进入 Android 开发行列，笔者特意精心编写了本书。本书以 Android 应用程序的开发为主题，并结合真实的案例向读者详细介绍了 Android 基本组件的使用及应用程序开发的整个流程。本书的讲述由浅入深，实例全面且典型，几乎囊括了所有和 Android 应用相关的项目。

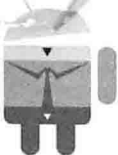
从技术角度而言，Android 是一种融入了全部 Web 应用的平台。随着版本的更新，从最初的触屏到现在的多点触控，从普通的联系人到现在的数据同步，从简单的 Google Map 到现在的导航系统，从基本的网页浏览到现在的 HTML 5，这都说明 Android 已经逐渐稳定，而且功能越来越强大。此外，Android 平台不仅支持 Java、C、C++ 等主流的编程语言，还支持 Ruby、Python 等脚本语言，Google 甚至专为 Android 应用开发推出了 Simple 语言，这使得 Android 有着非常广泛的开发群体。



本书的内容

在本书的内容中，以成稿时最新的 Android 4.4 平台为基础，按不同的门类开发了 166 个实例，其中包含 4 个大型、完整的案例，主要内容如下。

章	主要内容
第1章	UI布局实例
第2章	控件实例
第3章	自动化服务实例



章	主要内容
第4章	数据存储实例
第5章	电话和短信实例
第6章	图形图像实例
第7章	和网络有关的实例
第8章	多媒体实例
第9章	Google地图实例
第10章	Google API实例
第11章	平板应用
第12章	游戏应用
第13章	Android程序优化
第14章	可穿戴设备开发精粹
第15章	手机地图系统
第16章	音乐播放器系统
第17章	NBA激情投篮
第18章	综合邮件系统

参与本书编写的人员有：王东华、王振丽、熊斌、朱桂英、周秀、邓才兵、罗红仙、王石磊、孙宇、程娟、王文忠、王梦、陈强、于洋、管西京。本团队由于水平所限，书中难免有不足之处。如有纰漏和不尽如人意之处，诚请读者提出意见或建议，以便修订并使之更臻完善。另外，为了更好地为读者服务，我们专门提供了技术支持网站 www.topchuban.com，欢迎读者光临论坛，无论是书中的疑问，还是学习过程中的疑惑，本团队将尽力为大家解答。请访问 www.broadview.com.cn/17317 下载本书的相关资源。

编著者
2014 年 7 月

目 录

第 1 章 UI 布局实例集锦..... 1

实例 001: 使用线性布局 (LinearLayout)	
来布局屏幕.....	1
实例 002: 使用相对布局 (RelativeLayout)	
来布局屏幕.....	3
实例 003: 使用表格布局 (TableLayout)	
来布局屏幕.....	5
实例 004: 使用绝对布局 (AbsoluteLayout)	
来布局屏幕.....	6
实例 005: 使用标签布局 (TabLayout) 来	
布局屏幕.....	7
实例 006: 使用层布局 (FrameLayout)	9
实例 007: 创建一个桌面组件 Widget	10
实例 008: 在屏幕中实现一个按钮效果.....	11
实例 009: 在屏幕中显示文字.....	12
实例 010: 在屏幕中显示编辑框.....	15
实例 011: 在屏幕中显示复选框.....	17
实例 012: 在屏幕中显示单选框.....	19
实例 013: 在屏幕中显示下拉列表框.....	20
实例 014: 在屏幕中实现自动输入文本效果.....	23
实例 015: 使用日期选择器控件 DatePicker.....	26
实例 016: 自动选择一个输入时间.....	27
实例 017: 实现屏幕自动滚动.....	29
实例 018: 实现一个进度条效果.....	29
实例 019: 开发一个评分程序.....	31
实例 020: 在屏幕中显示一幅指定的图片.....	33
实例 021: 设置一幅图片作为按钮来使用.....	34
实例 022: 实现 QQ 空间中的显示照片效果.....	36
实例 023: 实现书架展示排列布局效果.....	38
实例 024: 实现树形菜单布局效果.....	40

第 2 章 控件实例集锦.....42

实例 025: 显示在文本框中输入的信息	42
实例 026: 使用 RadioGroup 控件实现选项	
选择功能	44
实例 027: 使用 CheckBox 控件实现同意条	
款效果.....	45
实例 028: 使用 Spinner 控件实现选项选择	48
实例 029: 使用 Gallery 控件实现个人相簿	
功能.....	51
实例 030: 开发一个文件搜索程序.....	54
实例 031: 单击按钮后实现按钮置换.....	56
实例 032: 模拟实现一个时钟界面效果.....	58
实例 033: 联合使用 DatePicker 和 TimePicker	
来选择日期和时间.....	60
实例 034: 动态分隔手机屏幕	62
实例 035: 使用 ListActivity 实现动态菜单	
列表.....	65
实例 036: 加载手机磁盘中的文件.....	67
实例 037: 实现抽屉样式效果	69
实例 038: 在屏幕中动态显示多种对话框.....	74
实例 039: 手机秘书——重要的事情提醒	79
实例 040: 动态添加或删除菜单	85
实例 041: 自定义 Android 控件实现单选按钮.....	89
实例 042: 使用自定义控件在屏幕中绘制一	
条虚线.....	93
实例 043: 在屏幕中实现手风琴效果.....	94
实例 044: 在屏幕中实现有秒针的时钟	96
实例 045: 修饰屏幕中的 Button 按钮.....	98
实例 046: 在屏幕中实现定时器效果.....	99
实例 047: 设置屏幕中控件的样式.....	101



第 3 章 自动化服务实例集锦..... 102

- 实例 048: 自动获取手机屏幕的分辨率..... 102
- 实例 049: 自动获取手机电池的剩余电量..... 105
- 实例 050: 来电时自动发送信息提醒..... 107
- 实例 051: 获取手机存储卡的容量..... 111
- 实例 052: 操作存储卡和内存卡中的数据..... 114
- 实例 053: 设置闹钟..... 120
- 实例 054: 实现黑名单来电自动静音处理..... 126
- 实例 055: 设置一个自动开机界面..... 128
- 实例 056: 更换手机屏幕背景图片系统..... 130
- 实例 057: 设置自动更换手机桌面背景..... 133
- 实例 058: 自动控制后台系统服务..... 143
- 实例 059: 在屏幕中拖动一个按钮..... 145
- 实例 060: 启动和关闭 AlarmManager 服务.... 149
- 实例 061: 自动拦截黑名单..... 151

第 4 章 数据存储实例集锦..... 155

- 实例 062: 在屏幕中显示 SharedPreferences
中存储的信息..... 155
- 实例 063: 演示数据添加、删除等操作..... 156
- 实例 064: 编写一个手机日记本程序..... 161
- 实例 065: 使用 ContentProvider 开发一个手
机日记本..... 167
- 实例 066: 使用文件保存数据..... 177
- 实例 067: 将网上的图片保存到 SD 卡并在手
机中显示出来..... 179
- 实例 068: 保存联系人信息..... 182
- 实例 069: 使用文件方式保存信息..... 184
- 实例 070: 读取上次开机时保存的信息..... 186

第 5 章 电话和短信实例集锦..... 188

- 实例 071: 编写一个拨号程序..... 188
- 实例 072: 发送一条短信..... 190
- 实例 073: 单击按钮实现拨号功能..... 192
- 实例 074: 一个通用发短信程序..... 195
- 实例 075: 收到短信后用 E-mail 邮件进行
通知..... 199
- 实例 076: 实现触屏拨号功能..... 202
- 实例 077: 一个短信群发程序..... 204
- 实例 078: 来电后自动发送邮件通知..... 206
- 实例 079: 监听手机拨接状态..... 208

实例 080: 检查短信是否发送成功..... 211

第 6 章 图形图像实例集锦..... 215

- 实例 081: 在屏幕中绘制一个矩形..... 215
- 实例 082: 在屏幕中绘制一个画布..... 218
- 实例 083: 实现水纹的透明效果..... 220
- 实例 084: 渲染几何图形..... 223
- 实例 085: 用图片为背景实现动画效果..... 226
- 实例 086: 在手机屏幕中实现 Frame 动画
效果..... 228
- 实例 087: 对图片进行缩放处理..... 230
- 实例 088: 旋转屏幕中的图片..... 233
- 实例 089: 实现旗帜飞扬的效果..... 236
- 实例 090: 模拟构建一个 3D 场景效果..... 240
- 实例 091: 模拟实现粒子系统效果..... 243
- 实例 092: 在 Canvas 中写文字..... 246
- 实例 093: 移动屏幕中的图片..... 253
- 实例 094: 在屏幕中绘制一个三维圆柱体..... 256
- 实例 095: 获取并显示图片的宽和高..... 264
- 实例 096: 编写一个手机屏保程序..... 266
- 实例 097: 以触摸方式移动图片..... 273
- 实例 098: 显示存储卡中的图片信息..... 276
- 实例 099: 在手机屏幕中显示系统内的图片
信息..... 278

第 7 章 和网络有关的实例集锦..... 281

- 实例 100: 传递 HTTP 参数..... 281
- 实例 101: 在屏幕中使用 HTML 程序..... 284
- 实例 102: 在手机中浏览网页..... 285
- 实例 103: 使用内置浏览器打开网页..... 286
- 实例 104: 显示 QQ 空间中的照片..... 288
- 实例 105: 通过 RSS 系统在线浏览新闻..... 290
- 实例 106: 下载网络图片作为手机背景..... 300
- 实例 107: 将文件上传至服务器..... 302
- 实例 108: 开发一个移动博客发布者..... 305
- 实例 109: 远程下载并安装一个软件..... 309
- 实例 110: 控制蓝牙系统..... 313
- 实例 111: 解析并生成 XML 文件..... 321
- 实例 112: 获取某个指定网页的 HTML 源码..... 322
- 实例 113: 使用 PULL 解析 XML 数据..... 324
- 实例 114: 采用 POST 方式向服务器传递
数据..... 327

第 8 章 多媒体实例集锦.....	330	实例 146: 勇敢派递员游戏.....	465
实例 115: 播放一个指定的 MP3 文件.....	330	第 12 章 游戏应用.....	467
实例 116: 调节手机音量的大小.....	333	实例 147: 一个简单的贪吃蛇游戏.....	467
实例 117: 实现相机预览和拍照功能.....	336	实例 148: 开发一个魔塔游戏.....	472
实例 118: 在手机中播放影片.....	342	实例 149: 疯狂足球游戏.....	491
实例 119: 手机录音系统.....	345	实例 150: 21 点游戏.....	512
实例 120: 设置手机中的铃声.....	350	实例 151: 俄罗斯方块游戏.....	515
实例 121: 在线播放网络中的 MP3.....	353	第 13 章 Android 程序优化.....	519
实例 122: 在线下载音乐作为手机铃声.....	359	实例 152: 测试计算机的性能.....	519
实例 123: 播放 GIF 格式的动画.....	365	实例 153: 测试内存性能.....	521
实例 124: 在手机中播放 MP4 视频.....	370	实例 154: 优化 Android 模拟器.....	524
实例 125: 在线观看 3GP 视频.....	373	实例 155: 使用 merge 标签优化 UI 界面.....	527
第 9 章 Google 地图实例集锦.....	381	实例 156: 优化 Android Layout.....	529
实例 126: 定位当前位置的坐标.....	381	实例 157: 优化 Bitmap 图片.....	531
实例 127: 在手机中使用 Google 地图.....	384	第 14 章 可穿戴设备开发精粹.....	533
实例 128: 在地图中定位输入的坐标.....	389	实例 158: 使用 SensorSimulator 传感器.....	533
实例 129: 在地图中快速查询某个位置.....	392	实例 159: 使用磁场传感器.....	537
实例 130: 随着手机的移动自动更新位置.....	394	实例 160: 使用加速度传感器.....	538
实例 131: 在手机地图中描绘线路并计算 线路距离.....	398	实例 161: 使用姿态传感器.....	540
实例 132: 在 Google 地图中显示指定的位置.....	404	实例 162: 使用温度传感器.....	543
实例 133: 在手机地图中查询一个地址.....	406	第 15 章 综合实例——手机地图系统.....	546
实例 134: 在手机中实现地址查询.....	407	实例 163: 使用 Google 地图开发一个综合地 图系统.....	546
实例 135: 开发一个路径导航系统.....	411	15.1 项目分析.....	546
第 10 章 Google API 实例集锦.....	418	15.1.1 规划 UI 界面.....	546
实例 136: 模拟验证官方账号.....	418	15.1.2 数据存储设计.....	547
实例 137: 在手机中开发一个 Google 搜索 程序.....	424	15.2 具体实现.....	548
实例 138: 在手机中编写一个翻译软件.....	429	15.2.1 新建工程.....	548
实例 139: 生成一个二维码.....	431	15.2.2 主界面.....	548
实例 140: 使用 Google Chart API 生成二 维码.....	434	15.2.3 新建界面.....	550
第 11 章 平板应用.....	439	15.2.4 设置界面.....	552
实例 141: 开发一个音乐播放器.....	439	15.2.5 帮助界面.....	557
实例 142: 开发一个在线天气播报系统.....	445	15.2.6 地图界面.....	559
实例 143: 开发一个网页浏览程序.....	454	15.2.7 数据存取.....	569
实例 144: 使用文件来保存音乐状态.....	459	15.2.8 实现 Service 服务.....	574
实例 145: 在屏幕中显示不同的进度条效果.....	462	15.3 发布自己的作品来赢利.....	576
		15.3.1 申请成为会员.....	576
		15.3.2 生成签名文件.....	578



15.3.3 使用签名文件	583
15.3.4 发布	585

第 16 章 综合实例——音乐播放器系统..... 586

实例 164: 开发一个音乐播放器	586
-------------------------	-----

16.1 项目介绍	586
16.1.1 项目背景介绍	586
16.1.2 项目目的	587
16.1.3 对读者的意义	587
16.2 需求分析	587
16.2.1 构成模块	587
16.2.2 功能结构图	590
16.2.3 系统功能说明	591
16.2.4 总体目标	591
16.3 设计数据库	592
16.3.1 字段设计	592
16.3.2 数据库连接	592
16.3.3 创建数据库	593
16.3.4 操作数据库	594
16.3.5 数据显示	595
16.4 具体编码	595
16.4.1 服务信息主界面	595
16.4.2 播放器主界面	597
16.4.3 播放列表功能	606
16.4.4 菜单功能模块	609
16.4.5 播放设置界面	611
16.4.6 设置显示歌词	614
16.4.7 文件浏览器模块	616

第 17 章 体育类游戏——NBA 激情投篮..... 620

实例 165: NBA 激情投篮	620
------------------------	-----

17.1 项目介绍	620
17.1.1 意义和技术分析	620
17.1.2 策划游戏	621

17.1.3 准备工作	621
17.2 项目架构	622
17.2.1 总体架构	622
17.2.2 规划类	622
17.3 具体编码	624
17.3.1 编写主类 LanqiuActivity ...	624
17.3.2 编写常量类	627
17.3.3 设计游戏试玩界面	632
17.3.4 绘制游戏场景	637

第 18 章 综合邮件系统..... 645

实例 166: 开发一个综合邮件系统	645
--------------------------	-----

18.1 项目介绍	645
18.1.1 项目背景介绍	645
18.1.2 项目目的	645
18.2 系统需求分析	646
18.2.1 构成模块	646
18.2.2 系统流程	649
18.2.3 功能结构图	649
18.2.4 系统功能说明	650
18.2.5 系统需求	650
18.3 数据存储设计	650
18.3.1 用户信息类	651
18.3.2 SharedPreferences.....	654
18.4 具体编码	656
18.4.1 欢迎界面	657
18.4.2 系统主界面	659
18.4.3 邮箱类型设置	666
18.4.4 邮箱收取设置	669
18.4.5 邮箱发送设置	673
18.4.6 邮箱用户检查	678
18.4.7 设置用户别名	682
18.4.8 用户邮件编辑	685
18.5 总结	691



第 1 章 UI 布局实例集锦

外观向来是工业产品的设计核心，是激发用户购买欲望的主要因素之一。本章讲的 UI 布局指的是手机界面布局，一款手机的屏幕界面效果是吸引用户购买的重要元素之一，因为消费者更倾向于选择界面美观的产品。在设计优美的界面之前，一定要先对屏幕进行布局。本章将通过具体实例的实现过程介绍在 Android 系统中规划 UI 界面的方法。



实例 001：使用线性布局（LinearLayout）来布局屏幕

源码路径：daima\001

知识点介绍

在 Android 布局中，需要了解视图容器组件——ViewGroup 的概念，使用视图容器组件 ViewGroup 的语法格式如下：

```
Android.view.Viewgroup
```

ViewGroup 的功能是包含并管理下级系列的 Views 和其他 ViewGroup，是一个布局的基类。类 ViewGroup 好像一个 View 容器，负责对添加进来的 View 进行布局处理。一个 ViewGroup 可以添加到另一个 ViewGroup 中去。这是因为 ViewGroup 也继承于 View.Viewgroup 类，是其他容器类的基类。它们之间的关系如图 1-1 所示。

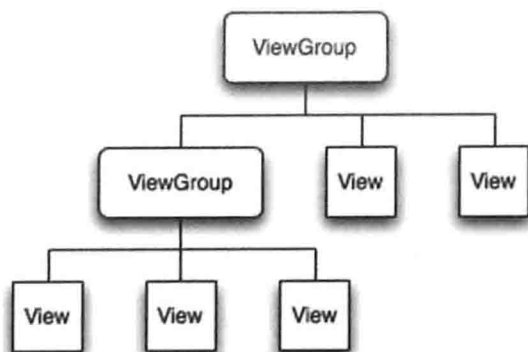
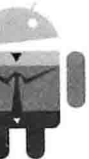


图 1-1 各个类的继承关系

我们知道，一个 Android 程序是由一个或多个 Activity 组成的，每个 Activity 是一个 UI 容器，Activity 本身不在用户界面中显示出来。在 Android 中，类 View 起了非常重要的作用，View 是一个最基本的 UI 类，几乎所有的 UI 组件都是继承于 View 而实现的。



使用 View 的语法格式如下所示。

```
android.view.View
```

线性布局即 `LinearLayout` 布局，是 Android 屏幕中常用的布局方式之一，功能是垂直或水平地显示 `ViewGroup` 的子视图（View）元素。

具体实现

- step 1** 使用 Eclipse 创建一个名为“001”的 Android 工程。
- step 2** 编写布局文件“res/layout/main.xml”，代码如下所示。

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="horizontal">
    <Button android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="第一个按钮"
        android:layout_weight="1"
    />
    <Button android:id="@+id/button2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="第二个按钮"
        android:layout_weight="1"
    />
    <Button android:id="@+id/button3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="第三个按钮"
        android:layout_weight="1"
    />
    <Button android:id="@+id/button4"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="第四个按钮"
        android:layout_weight="1"
    />
    <Button android:id="@+id/button5"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="第五个按钮"
        android:layout_weight="1"
    />
</LinearLayout>
```

在上述代码中，在根 `LinearLayout` 视图组（`ViewGroup`）中包含了 5 个按钮（`Button`），它的子元素是以线性方式水平布局的。上述代码的运行效果如图 1-2 所示。



图 1-2 LinearLayout 布局效果



实例 002：使用相对布局（RelativeLayout）来布局屏幕

源码路径：daima\002

知识点介绍

相对布局是设计用户界面的有力工具，因为它消除了嵌套视图组。相对布局是指一个 ViewGroup 以相对位置显示它的子视图（View）元素，一个视图可以指定相对于它的兄弟视图的位置（如在给定视图的左边或者下面）或相对于 RelativeLayout 的特定区域的位置（如底部对齐或中间偏左）。

RelativeLayout 允许子元素指定它们相对于其他元素或父元素的位置（通过 ID 指定）。所以可以用右对齐、上下对齐或置于屏幕中央的形式来排列两个元素。如果第一个元素在屏幕的中央，那么相对于这个元素的其他元素将以屏幕中央的相对位置来排列。如果使用 XML 来指定这个布局，在定义它之前，被关联的元素必须定义。RelativeLayout 结构如图 1-3 所示。

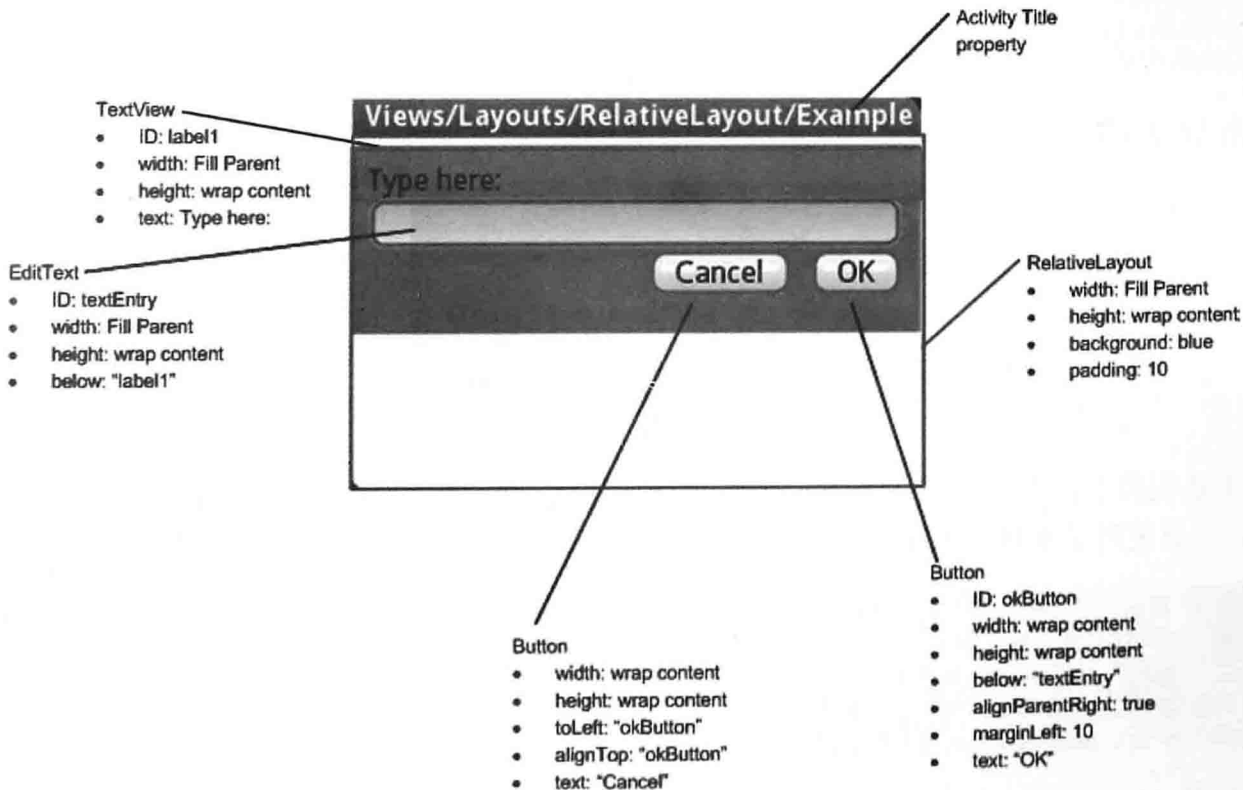


图 1-3 RelativeLayout 结构

具体实现

step 1 使用 Eclipse 创建一个名为“002”的 Android 工程。



step 2 编写布局文件“res/layout/main.xml”，代码如下所示。

```
<TextView
    android:id="@+id/label"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="请输入信息:" />
<EditText
    android:id="@+id/entry"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:background="@android:drawable/editbox_background"
    android:layout_below="@id/label" />
<Button
    android:id="@+id/ok"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@id/entry"
    android:layout_alignParentRight="true"
    android:layout_marginLeft="10dip"
    android:text="确定" />
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_toLeftOf="@id/ok"
    android:layout_alignTop="@id/ok"
    android:text="取消" />
```

执行后的效果如图 1-4 所示。

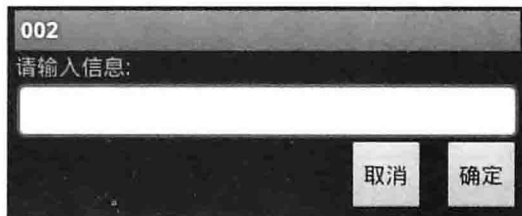


 图 1-4 执行效果

在上述实例中，在 RelativeLayout 视图组中包含了一个 TextView、一个 EditText 和两个 Button，并且使用了下面的属性代码来设置布局的显示。

```
android:layout_below="@id/label"/>
android:layout_below="@id/entry"
android:layout_alignParentRight="true"
android:layout_toLeftOf="@id/ok"
android:layout_alignTop="@id/ok"
```

在使用相对布局方式时，正是使用这些类似的属性来定位视图到你想要的位置的，它们的值是你参照的视图的 ID。这些属性的意思很简单，就是英文单词的直译，这里就不多做介绍了。



实例 003：使用表格布局（TableLayout）来布局屏幕

源码路径：daima\003

知识点介绍

表格布局（TableLayout）其实是一个 ViewGroup 以表格显示它的子视图（View）元素，即行和列标识一个视图的位置。其实 Android 的表格布局与 HTML 中的表格布局非常类似，TableRow 就像 HTML 表格的<tr>标记。表格布局通常用于把子元素放入行与列中，不显示行、列或单元格边界线，但是单元格不能横跨行，像 HTML 中一样。表格布局效果如图 1-5 所示。



图 1-5 表格布局效果

在使用表格布局时需要了解如下 3 点。

- android:shrinkColumns: 对应的方法是 setShrinkAllColumns(boolean)，作用是设置表格的列是否收缩（列编号从 0 开始，下同），如果有多列则用逗号隔开（下同），如 android:shrinkColumns="0,1,2"，表示表格的第 1、2、3 列的内容是收缩的以适合屏幕，不会挤出屏幕。
- android:collapseColumns: 对应的方法是 setColumnCollapsed(int,boolean)，作用是设置表格的列是否隐藏。
- android:stretchColumns: 对应的方法是 setStretchAllColumns(boolean)，作用是设置表格的列是否拉伸。

具体实现

step 1 使用 Eclipse 创建一个名为“003”的 Android 工程。

step 2 编写布局文件“res/layout/main.xml”，代码如下所示。

```
<TableRow>
<Button android:id="@+id/button1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="第一个按钮"
    android:layout_column="0"
    />

    <Button android:id="@+id/button2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="第二个按钮"
    android:layout_column="1"
    />
</TableRow>
<TableRow><!-- row2 -->
<Button android:id="@+id/button3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
```




```
        android:text="第三个按钮"
        android:layout_column="1"
    />
<Button android:id="@+id/button4"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="第四个按钮"
    android:layout_column="1"
    />
</TableRow>
<TableRow>
    <Button android:id="@+id/button5"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="第五个按钮"
        android:layout_column="2"
        />
</TableRow>
```

执行后的效果如图 1-6 所示。

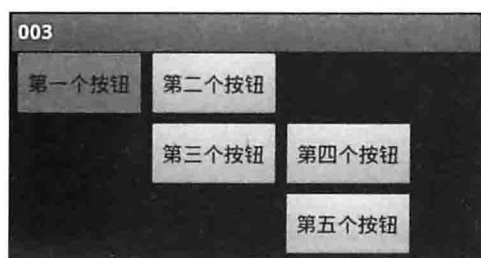


图 1-6 执行效果



实例 004：使用绝对布局（AbsoluteLayout）来布局屏幕

源码路径：daima\004

知识点介绍

绝对布局（AbsoluteLayout）是指一个 ViewGroup 以绝对方式显示它的子视图（View）元素，即以坐标的方式来定位在屏幕上位置。这种布局方式很好理解，在布局文件中编程设置 View 的坐标，从而绝对地定位。

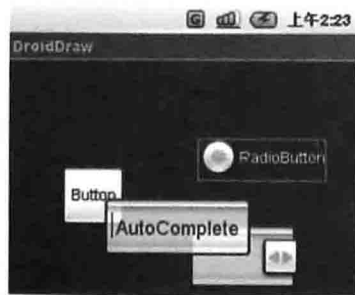


图 1-7 AbsoluteLayout 效果

AbsoluteLayout 可以让子元素指定准确的 x 和 y 坐标值，并显示在屏幕上。其中坐标(0, 0)为左上角，当向下或向右移动时，坐标值将变大。AbsoluteLayout 没有页边框，允许元素之间互相重叠（尽管不推荐）。我们通常不推荐使用 AbsoluteLayout，除非有正当理由使用它，因为它会使界面代码太过刚性，以至于在不同的设备上可能不能很好地工作。AbsoluteLayout 效果如图 1-7 所示。

具体实现

- step 1** 使用 Eclipse 创建一个名为“004”的 Android 工程。
- step 2** 编写布局文件“res/layout/main.xml”，代码如下所示。

```
<AbsoluteLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/AbsoluteLayout01"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >
    <TextView android:id="@+id/txtIntro"
        android:text="使用绝对布局"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_x="20dip"
        android:layout_y="20dip">
    </TextView>
</AbsoluteLayout>
```

执行后的效果如图 1-8 所示。



图 1-8 执行效果



实例 005：使用标签布局（TabLayout）来布局屏幕

源码路径：daima\005

知识点介绍

标签布局（TabLayout）是一个 ViewGroup 以标签的方式显示它的子视图（View）元素，就像在 Firefox 中的一个窗口中显示多个网页一样。为了创建一个标签 UI（tabbed UI），需要使用 TabHost 和 TabWidget。TabHost 必须是布局的根节点，它包含显示标签的 TabWidget 和显示标签内容的 FrameLayout。

具体实现

- step 1** 使用 Eclipse 创建一个名为“005”的 Android 工程。
- step 2** 编写布局文件“res/layout/main.xml”，代码如下所示。

```
<?xml version="1.0" encoding="utf-8"?>
```



```
<TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent" android:layout_height="fill_parent">
<TableRow>
    <Button android:layout_width="wrap_content" android:layout_height="wrap_content"
        android:text=" 7 " />
    <Button android:layout_width="wrap_content"
        android:layout_height="wrap_content" android:text=" 8 "
    />
    <Button android:layout_width="wrap_content"
        android:layout_height="wrap_content" android:text=" 9 "
    />
    <Button android:layout_width="wrap_content"
        android:layout_height="wrap_content" android:text=" / "
    />
</TableRow>
<TableRow>
    <Button android:layout_width="wrap_content"
        android:layout_height="wrap_content" android:text=" 4 "
    />
    <Button android:layout_width="wrap_content"
        android:layout_height="wrap_content" android:text=" 5 "
    />
    <Button android:layout_width="wrap_content"
        android:layout_height="wrap_content" android:text=" 6 "
    />
    <Button android:layout_width="wrap_content"
        android:layout_height="wrap_content" android:text=" * "
    />
</TableRow>
<TableRow>
    <Button android:layout_width="wrap_content"
        android:layout_height="wrap_content" android:text=" 1 "
    />
    <Button android:layout_width="wrap_content"
        android:layout_height="wrap_content" android:text=" 2 "
    />
    <Button android:layout_width="wrap_content"
        android:layout_height="wrap_content" android:text=" 3 "
    />
    <Button android:layout_width="wrap_content"
        android:layout_height="wrap_content" android:text=" . "
    />
</TableRow>
<TableRow>
    <Button android:layout_width="wrap_content"
        android:layout_height="wrap_content" android:text=" 0 "
    />
    <Button android:layout_width="wrap_content"
        android:layout_height="wrap_content" android:text=" = "
    />
    <Button android:layout_width="wrap_content"
        android:layout_height="wrap_content" android:text=" - "
```