

松崗

iOS

创意程序设计家

林柏全 编著

李强 改编



全面的开发知识：

从创意、设计、应用开发等全方位介绍了在 iOS 平台开发 App 的技术和技巧。

主流的热点技术：

加速度计、电子罗盘、照相机、多点触控、智能感应器、陀螺仪、界面控制器、通讯录、多媒体、定位与地图等。

人民邮电出版社
POSTS & TELECOM PRESS

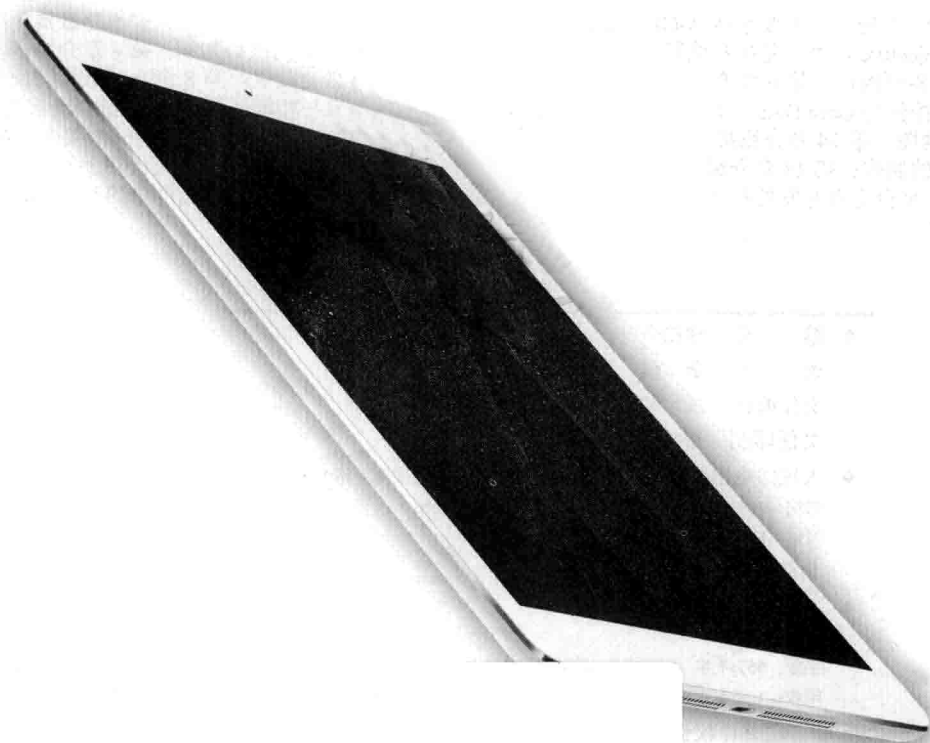
松崗

iOS

创意程序设计家

林柏全 编著

李强 改编



人民邮电出版社

北京

图书在版编目 (C I P) 数据

iOS创意程序设计家 / 林柏全编著. — 北京: 人民邮电出版社, 2014.6
ISBN 978-7-115-34702-2

I. ①i… II. ①林… 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069951号

版 权 声 明

本书繁体字版名为《iOS 创意程序设计家 第三版》, 由松岗资产管理股份有限公司出版, 版权属松岗资产管理股份有限公司所有。本中文简体字版由松岗资产管理股份有限公司授权人民邮电出版社独家出版发行。未经本书原版出版者和本书出版者书面许可, 任何单位和个人均不得以任何形式或任何手段复制或传播本书的部分或全部。

内 容 提 要

本书共分 20 章, 从创意、设计、应用开发等方面介绍了在 iOS 平台上开发 App 的全部技术和技巧, 主要包括的内容如下。

第 1 章 iOS 的硬件及应用, 主要包括 A-GPS、加速度计、电子罗盘、照相机、多点触控、智能感应器、陀螺仪等; 第 2 章讲解 Objective-C 语言基础和编程技术; 第 3 章讲解开发工具 XCode 的核心知识; 第 4 章介绍基本类与机制; 第 5 章介绍界面控件; 第 6 章介绍界面控制器; 第 7 章介绍操作信息的截取; 第 8 章介绍数据访问与打印; 第 9 章介绍表格控件与 Core Data; 第 10 章介绍通讯录; 第 11 章介绍相机与相簿; 第 12 章介绍影音多媒体; 第 13 章介绍定位与地图; 第 14 章介绍推送服务; 第 15 章介绍使用外界服务; 第 16 章介绍其他的框架库; 第 17 章介绍通用应用程序的制作; 第 18 章介绍漫步在云; 第 19 章介绍完善你的程序; 第 20 章介绍程序上架流程。

本书适合在 iOS 平台上做开发的程序员和爱好者阅读, 也适合作为大中专院校相关专业师生的参考书, 以及培训学校的教材。

- ◆ 编 著 林柏全
- 改 编 李 强
- 责任编辑 张 涛
- 责任印制 彭志环 焦志炜

- ◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路 11 号
邮编 100164 电子邮件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京艺辉印刷有限公司印刷

- ◆ 开本: 800×1000 1/16
印张: 27.75
字数: 557 千字 2014 年 6 月第 1 版
印数: 1~2 500 册 2014 年 6 月北京第 1 次印刷

著作权合同登记号 图字: 01-2012-4608 号

定价: 69.00 元

读者服务热线: (010)81055410 印装质量热线: (010)81055316

反盗版热线: (010)81055315

广告经营许可证: 京崇工商广字第 0021 号



前 言

从第一代 iPhone 开始，苹果已经成了媒体版面上报道的常客。这家公司在智能手机所创造出来的奇迹，以及顾客的忠诚度，短期间恐怕很难有公司可以望其项背。这个现象的背后所代表的真正意义是：创意的时代已经取代制造生产为主的时代。而软件产业则将会成为这个创意时代的最佳摇篮！使用者要的不再只是单纯的由硬件外观所带来的视觉感受，取而代之的是结合了软件所带来的，包括触觉、听觉等在内的炫彩的科技体验。

从 App Store 的总下载量突破几百亿大关，以及这期间所创造出来的 iPhone 新贵的现象来看，再也没有人可以小看这个市场了。与其说第一次体验到 iPhone 的使用会有一种令人难以言喻的新鲜感，那么，第一次可以让自己的创意程序运行在 iPhone 上，所带来的成就感才是前所未有的！市面上虽然已经有不少关于 iPhone 开发的书籍，然而，这些书籍除了清一色全都是外文书籍外，对于大多数的读者来说，可能很难从中体会出创意的乐趣。本书以创意、设计、实用为主轴，希望能让有创意的你成为这场盛宴当中的焦点。

从 iPhone 一代到 iPhone 3GS，甚至于现在的 iPhone5，Apple 总是能够带给消费者许多惊喜。除了硬件的进步外，也看到 Apple 不断地更新其操作系统 iOS。当然，这些版本的演进也会伴随着为数不少的新的 API 加入，对于开发者而言，这更意味着可以做开发应用的领域越来越广。在本书内容里，希望用浅显易懂的内容与范例，为读者奉献一本全方位的开发书籍。

在 2008 年我开始投入 App 软件开发之际，遭遇了不少挫折，其中最大的挫折感便是相关文件的缺乏。后来，将自己的学习过程记录下来，这才成为了本书。本书内容全面，包含了大部分开发所需的 API，并增加了许多 iOS 新版本的功能。苹果公司除了不断进行 iOS 的完善工作外，也持续开放更多的 API 让开发者使用。让开发者高兴的是，新一代的 XCode 已经解决了以往开发上的一些缺陷，其中最大的改变便是整合了原有的 InterfaceBuilder。这使得本书的内容扩充到 20 章。对于开发者而言，这样的改变使得开发效率变得更高了！在迈向 iOS 应用程序开发的路上，您除了需要 XCode 这个好用的工具之外，更需要的是像本书一样的全方位介绍 App 应用开发的参考书籍，源程序下载地址：www.toppr.net。编辑联系邮箱：zhangtao@ptpress.com.cn。

编 者

目 录

第 1 章 从硬件的认识开始	1
1.1 iOS 的硬件及其应用	3
1.1.1 上网能力	3
1.1.2 A-GPS	3
1.1.3 影音播放	5
1.1.4 加速度计	5
1.1.5 电子罗盘	6
1.1.6 照相机	6
1.1.7 多点触控	6
1.1.8 智能感应器	7
1.1.9 陀螺仪	7
1.2 必须了解的事	7
1.2.1 挑选适合开发用的计算机	7
1.2.2 账号申请	8
1.2.3 Apple 提供的管理工具	9
1.2.4 开发的注意事项	14
1.3 学习阶段	15
1.3.1 熟悉 Mac OS	15
1.3.2 语言基础的建立	16
1.3.3 开发工具的使用	16
1.3.4 iOS Provisioning Portal	17
1.3.5 各组件与 Framework 的使用	17
1.3.6 iTunes Connect 的使用	17
1.4 热身操	17
第 2 章 Objective-C 语言的基础	18
2.1 Cocoa 初体验	18
2.1.1 Carbon	19
2.1.2 快速一瞥	19
2.2 从 C 到 Objective-C	20
2.3 有趣的 Objective-C	22
2.3.1 选择器与消息	23
2.3.2 输出到控制台	24
2.3.3 类	25
2.3.4 #import 与@class	26
2.3.5 变量的访问权限	27
2.3.6 构造函数与析构函数	29
2.3.7 点运算符	29
2.3.8 协议 Protocol	31
2.3.9 分类 Category	32
2.3.10 Block 对象	33
2.3.11 异常捕捉	34
2.3.12 异常抛出	34
2.3.13 NSAssert	34
2.4 数据类型	35
2.4.1 原始数据类型	35
2.4.2 扩展的数据类型	36
2.4.3 动态识别	37
2.4.4 常用类	38
2.5 内存管理	43
2.5.1 内存管理模式	43
2.5.2 自动释放内存	44
2.5.3 自动参考计数 ARC	45
2.6 命名风格	48

2.6.1 类的命名	48	4.1.4 UIWindow	96
2.6.2 变量的命名	48	4.2 代理机制与事件处理	97
2.6.3 Selector 的命名	49	4.2.1 应用程序生命周期相关事件	97
2.7 框架库	50	4.2.2 状态栏的事件	100
2.8 热身操	51	4.3 消息通知机制	101
第 3 章 开发工具	52	4.3.1 通过观察者接收消息	101
3.1 XCode	52	4.3.2 运用消息通知侦测系统属性变化	102
3.1.1 认识 XCode	53	4.4 Target-Action 机制	103
3.1.2 窗口的切换	56	4.5 热身操	104
3.1.3 项目导航区域	58	第 5 章 界面控件	105
3.1.4 编辑区域	58	5.1 一切界面控件的开始 UIView	105
3.1.5 调试区域	59	5.1.1 坐标系统	107
3.1.6 检视窗口	60	5.1.2 子界面的处理	108
3.1.7 控件库窗口	64	5.1.3 重绘与多点触控事件处理	109
3.1.8 项目模板	65	5.1.4 具有圆角的 UIView	111
3.2 建立第一个应用程序	69	5.1.5 动画效果	112
3.2.1 HelloWorld 1.0—使用 XIB	69	5.1.6 应用范例：任意变	117
3.2.2 HelloWorld 2.0—使用故事板	75	5.2 选择与响应控件	120
3.2.3 修改图标以及显示名称	78	5.2.1 对话框控件 UIAlertView	120
3.2.4 在真实手机上执行	79	5.2.2 操作列表控件 UIActionSheet	123
3.3 建立静态库	81	5.2.3 日期选择控件 UIDatePicker	123
3.4 真假难分—iOS 模拟器	84	5.2.4 滚轮控件 UIPickerView	125
3.5 管理工具—Organizer	86	5.3 状态显示控件	127
3.6 应用程序加载工具		5.3.1 活动状态指示控件	
—Application Loader	88	UIActivityIndicatorView	127
3.7 热身操	89	5.3.2 进度状态指示控件	
3.7.1 沙盒—应用程序的限制	89	UIProgressView	128
3.7.2 MVC 模式	89	5.3.3 标记的使用	129
3.7.3 打开潘多拉的盒子	90	5.4 文字输入与显示控件	129
第 4 章 基本类与机制	91	5.4.1 标签控件 UILabel	129
4.1 基本类	91	5.4.2 单行文本框控件 UITextField	130
4.1.1 UIDevice	91	5.4.3 多行文本框控件 UITextView	136
4.1.2 UIApplication	93	5.4.4 再谈键盘	139
4.1.3 UIScreen	95	5.5 其他控件	141
		5.5.1 一般按钮 UIButton	141

5.5.2	开关控件 UISwitch	142
5.5.3	滑轨控件 UISlider	142
5.5.4	分页控件 UIPageControl	143
5.5.5	分类控件 UISegmentedControl	144
5.5.6	计数器控件 UIStepper	145
5.6	手势识别	146
5.6.1	认识手势识别器	146
5.6.2	如何在可视化控件 加入手势识别	146
5.7	认识 Responder	149
5.8	UIAppearance 的使用	150
5.9	热身操	151
第 6 章	界面控制器	152
6.1	界面的管理	152
6.1.1	界面方向的管理	152
6.1.2	内存使用的管理	153
6.1.3	界面的加载流程	153
6.1.4	常见接口模式	154
6.2	导航栏控制器 UINavigationController	155
6.2.1	界面堆栈的概念	156
6.2.2	使用故事板来处理界面堆栈	157
6.2.3	单选按钮	158
6.2.4	如何建立导航栏应用程序	160
6.3	标签页控制器 UITabBar Controller	162
6.4	事件检测	164
6.4.1	晃动检测	165
6.4.2	应用范例: 魔术秀	165
6.4.3	加速度计	168
6.4.4	应用范例: 水平仪	169
6.5	Undo 与 Redo 机制	172
	应用范例: 破裂的手机	173
6.6	热身操	176

第 7 章	操作信息的截取	177
7.1	认识 Core Motion	177
7.1.1	数据获取的两种方式 Push 与 Pull	177
7.1.2	操作管理器 CMMotionManager	177
7.2	获取的数据	178
7.2.1	加速度计数据 CMAccelerometerData	179
7.2.2	陀螺仪数据 CMGyroData	179
7.2.3	设备操作数据 CMDeviceMotion	179
7.2.4	应用范例: 旋转的方块	181
7.3	热身操	184
第 8 章	数据访问与打印	186
8.1	文件路径	187
8.1.1	各文件夹路径的获得	187
8.1.2	应用程序的路径	188
8.2	文件的访问	188
8.2.1	PList 文件的访问	188
8.2.2	XML 文件的访问	189
8.2.3	iOS 的文件总管	199
8.3	数据库	201
8.3.1	SQLite 的使用与语法	202
8.3.2	在 iOS 中使用 SQLite	204
8.3.3	使用 FMDB 来读写 SQLite	206
8.4	打印	208
8.4.1	AirPrint	208
8.4.2	文件预览	210
8.5	热身操	212
第 9 章	表格控件与 Core Data	213
9.1	滚动的界面	213
9.1.1	UIScrollView 的概念	213
9.1.2	应用范例: 可缩放的图形	215
9.2	表格控件	218
9.2.1	表格控件的组成	218

9.2.2	数据源与事件代理	220	11.2.5	动态摄影	280
9.2.3	如何表示表格单元的位置	222	11.3	热身操	281
9.2.4	表格单元的样式	222	第 12 章 影音多媒体	282	
9.2.5	电子书的好搭档 UIPageViewController	224	12.1	音频	282
9.2.6	应用范例：小说的制作	227	12.1.1	使用系统音效	284
9.2.7	数据索引	237	12.1.2	使用 AVAudioPlayer 来 播放音频	285
9.2.8	数据搜索	237	12.1.3	应用范例：我的小小八音盒	286
9.3	Core Data	241	12.1.4	使用音频文件服务 解析 ID3 标签	289
9.3.1	Entity-Relationship 模型	241	12.1.5	播放 iPod 的音乐	291
9.3.2	为项目加入 Core Data 的支持	242	12.1.6	使用 AVAudioRecorder 来录音	296
9.3.3	使用多个表格	249	12.1.7	应用范例：神奇的麦克风	298
9.3.4	条件语句查询	252	12.2	视频	301
9.4	热身操	254	12.2.1	使用 MPMoviePlayerView Controller 播放	301
第 10 章 通讯录	255		12.2.2	使用 MPMoviePlayer Controller 播放	305
10.1	联系人数据的访问	256	12.2.3	播放 Youtube 影片	312
10.1.1	再谈 ARC	256	12.3	热身操	315
10.1.2	数据字段的获取	257	第 13 章 定位与地图	316	
10.1.3	数据字段的设置	261	13.1	Core Location	316
10.1.4	联系人新增与删除	262	13.1.1	CLLocationManager	316
10.2	信息显示	263	13.1.2	地理位置信息 CLLocation	318
	选择联系人的处理	263	13.1.3	地址与位置坐标之间的转换	319
10.3	热身操	265	13.1.4	方向信息 CLHeading	321
第 11 章 相机与相簿	266		13.2	地图	322
11.1	从图片谈起	266	13.2.1	地图控件	323
11.1.1	存储到 Documents 目录	266	13.2.2	模拟用户位置	324
11.1.2	上传到服务器	267	13.2.3	标记与注记	325
11.1.3	存储到相簿	268	13.3	热身操	327
11.1.4	显示相簿图片	269	第 14 章 推送服务	328	
11.2	照相机的使用	274	14.1	远程消息推送	328
11.2.1	拍摄静态相片	274	14.1.1	特色与限制	329
11.2.2	自定义相机界面	275			
11.2.3	应用范例：鬼影追追追	275			
11.2.4	相片特效处理	279			

14.1.2 产生 Provider 使用的证书	329
14.1.3 消息接收端的实现	331
14.1.4 消息发送端的实现	333
14.2 本地端消息推送	337
14.2.1 消息对象	337
14.2.2 消息的接收	339
14.3 热身操	339
第 15 章 使用外界服务	340
15.1 Bonjour	340
15.1.1 什么是 Bonjour	340
15.1.2 Bonjour 服务的表示方式	341
15.1.3 建立 Bonjour 服务	342
15.1.4 服务搜索与浏览	343
15.2 Game Kit 框架库	345
15.2.1 P2P 联机能力	346
15.2.2 配对界面	347
15.2.3 联机时的事件处理	347
15.2.4 数据传送	349
15.2.5 In-Game Voice	350
15.2.6 应用范例：自由讲	351
15.3 JSON	355
15.3.1 认识 JSON	356
15.3.2 NSJSONSerialization 的使用	356
15.3.3 SBJson 的使用	358
15.3.4 应用范例：Youtube 影片搜索器	359
15.4 与 Facebook 连接全世界	363
15.5 与 Twitter 整合	367
15.5.1 TWTweetCompose ViewController	367
15.5.2 TWRequest	369
15.6 热身操	370
第 16 章 其他的框架库	371
16.1 通过 Core Telephony 获取电信 运营商信息	371
16.1.1 获取电信运营商信息	371
16.1.2 获取通话状态	372
16.2 通过 Event Kit 访问日历	372
16.2.1 获取日历资料	373
16.2.2 显示日历界面	375
16.3 通过 iAd 投放广告	378
16.3.1 广告用语的认识	379
16.3.2 iAd 的设置	380
16.3.3 使用 iAd	380
16.3.4 广告载入事件	383
16.4 文件预览 Quick Look	384
16.4.1 界面控制器 QLPreview Controller	384
16.4.2 数据源 QLPreviewController DataSource	385
16.5 空中播放 AirPlay	387
16.5.1 体验 AirPlay	387
16.5.2 启用 AirPlay	388
16.6 热身操	389
第 17 章 通用应用程序的制作	390
17.1 如何产生通用应用程序	390
17.2 通用应用程序制作秘诀	391
17.2.1 图文件的载入	391
17.2.2 判断所使用的硬件	391
17.2.3 避免在程序代码内直接 指定对象大小	392
17.2.4 关于 UISplitViewController	392
17.3 热身操	392
第 18 章 漫步在云	393
18.1 iCloud 的存储种类	393
18.2 存储上的考虑	394
18.3 启用 iCloud	395
18.4 使用文件存储方式	396
18.4.1 获取 Ubiquity Container 路径	396

18.4.2 获取并建立 Documents 目录	398	19.3.2 单元测试	416
18.4.3 UIDocument	398	19.4 热身操	420
18.5 使用 Key-Value 存储方式	401	第 20 章 程序上架流程	421
18.6 热身操	402	20.1 上架前需知	421
第 19 章 完善您的程序	403	20.1.1 审核标准	422
19.1 设置	403	20.1.2 上架流程	423
19.1.1 应用程序的配置文件 Info.plist	403	20.2 编译发布用的应用程序	423
19.1.2 用户设置的访问	408	20.3 iTunes Connect	424
19.2 多国语言	413	20.3.1 应用程序的管理	425
19.2.1 让应用程序支持多国语言	414	20.3.2 应用程序的上传	430
19.2.2 配置文件的多语言	415	20.3.3 应用程序的状态	431
19.3 虫虫大作战	415	20.3.4 合约管理	432
19.3.1 内存泄漏	415	20.3.5 兑换码	432
		20.4 结语	433

第 1 章 从硬件的认识开始

iOS（包含 iPod Touch、iPhone、iPad）创造了许多历史，包括：击败了曾经是手机行业老大的 Nokia 与 Motorola 两家公司，掀起触屏操控的风潮，建立起 App Store 成功的典范，以及超过一百亿次的惊人下载次数。这其中的关键之处并非在于硬件方面的创新，而在于 iOS 整合了软件、硬件以及服务的完整配套措施，以及令人惊艳的用户体验。

这股潮流目前也正改变着软件从业人员的命运，能够圆很多人创业的梦想。关于在 App Store 上面创业的故事似乎永远也说不完，本书这里就列举三则故事。第一则故事是我认为最经典的，同时也是纽约时报在《The iPhone Gold Rush》这篇文章中所报导的，即有关 iShoot 的作者 Ethan Nicholas 的传奇故事。Nicholas 是 Sun Microsystems（现已被数据库巨擎 Oracle 公司收购）的一位工程师，其房贷、小孩的医疗费用问题，再加上 SUN 冻结了当年度的员工分红，似乎所有的压力都朝他排山倒海而来，这迫使 Nicholas 不得不另寻财源。忽然间他想起了曾看过一篇文章，有个幸运的家伙 Steve Demeter 因为写了“Trism”这款 iPhone 游戏而赚进了 25 万美元。不过，此时的 Nicholas 心中只想要渡过眼前的难关就好了。Nicholas 虽然有多年的开发经验，不过却从来没有碰过 Objective-C。于是，他开始上网寻找资料来学习，由于他对于射击游戏很感兴趣，因此决定从这方面着手。于是他便到图库网站上买了些便宜的图片与音效，再加上自己绘制一些所需的图形，就开始了他的淘金梦。整整六周的时间，他几乎每天早、中、晚都在工作：早上在 SUN 开发 Java 的程序，下午开始他的 iPhone 应用开发，傍晚开始他的家庭作业——照顾他的两个小孩，有时候还得一手敲着键盘，一手摇着最小的孩子 Gavin 的摇篮让他入睡。不过，他的辛苦总算有了回报，Nicholas 提交给 Apple 审核的游戏 iShoot 在 10 月 19 日正式上架到 App Store 上。这套售价 4.99 美元的软件，第一天便让他赚进 1000 美元，让他笑得合不拢嘴。第二天更攀升到 2000 美元，在那之后则一直维持每天 50 美元的进账。于是 Nicholas 开始思考他的策略，并于 2009 年一月推出一个功能较少的免费版本，希望借此刺激付费版本的销售量，结果他成功了！很多人后来都升级为付费版本，且升级的价格也只要 2.99 美元。最高峰时曾经一天卖出一万七千套，也就是一天之内他赚进了三万五千美元之多。现在的 Nicholas 呢？早就把他原来的老板炒掉了，并且带着家人到华盛顿渡假，还自己开了家叫“Naughty Bits Software”的公司呢！

接下来这一则故事是个无心插柳柳成荫的故事：一位年轻的斯坦福教授之前从来也没有想到要通过这种方式来赚钱。不过，他有个特殊的癖好，就是喜欢找几台笔记本电脑一起播放音乐，然后玩玩电子交响乐。有个朋友便提议他可以把这个点子放到 iPhone 应用里面，没想到他真跑去开了家叫“Smule”的公司，其中最著名的就是 Ocarina 这个陶笛软件，这个软件使他不到一个月内竟然赚进 100 万美元！

最后一则故事则是关于 Apple 新的家族成员 iPad 开发的故事。苹果应用开发者尤瑞-塞鲁基奥夫 (Yuri Selukioff) 早在 iPhone 的平台上就开发了一款可用来阅读 PDF、Word、Text 文件的软件, 叫做 “Good Reader”, 并于后来转移到 iPad 平台上销售。而由于在开发前, 他们尚无法取得 iPad 的硬件, 因此在那之前只能在仿真器上进行测试。在 iPad 上市的当天, 他立刻取得 iPad 并进行测试, 又很快地发布修正版本, 而这个软件竟让他在十天内赚到近七万美元。这个故事说明一个重点, 那就是 iPhone 与 iPad 的兼容性十分高, 大部分的应用程序不需要修改就可以执行了。

这些创业家的故事颠覆了大家以往对软件的概念, 以为只要收钱的软件就不会成功。事实证明, 便宜与创意才是这种模式成功的主因, 这也刚好符合目前世界的潮流。即使您写的是免费的软件, Apple 也提供了一个完善的机制让您取得应有的报酬。所以不管是付费还是免费软件, 只要能好好发挥您的想象力, iOS 可能就是让您发光发热的好机会!

接下来, 我们介绍一下硬件部分。毕竟, 软件是必须依赖硬件而存在的, 很多人拿到 iPhone 或 iPad 后就急着想先写个 “HelloWorld” 来玩玩, 到后来反而对于硬件一知半解, 不免可惜。因此, 在揭开序幕的这一章中, 我们将为您介绍 iOS 设备的硬件特性以及它们的一些相关应用。表 1.1 所示是当前已经发布的 iOS 硬件的对照表 (请注意, iOS 4.3 以后的版本将无法安装在 iPhone 3G 上)。

表 1.1 iOS 硬件规格对照表

规格	iPad	iPad2	iPhone 3Gs	iPhone 4	iPhone4s	iPhone5s
屏幕尺寸	9.7 英寸		3.5 英寸			4 英寸
分辨率	1024×768 (132 像素/英寸)		480×320 (163 像素/英寸)	960×640 (326 像素/英寸)	960×640 (326 像素/英寸)	1136×640 像素
WIFI	802.11/b/g/n		802.11/b/g	802.11/b/g/n		
3G 传输	7.2Mbit/s		3.6Mbit/s	7.2Mbit/s	14.4Mbit/s	支持 4G
GPS	A-GPS + 电子罗盘					
照相机	无	前: 30 万/后: 70 万	3MP	前: 30 万/后: 500 万	前: 30 万/后: 800 万	800 万
CPU	A4 (单核) 800/1G MHz	A5 (双核) 800/1G MHz	600/800 MHz	A4 (单核) 800/1G MHz	A5 (双核) 1G MHz	苹果 A7(双核) 1G MHz
录像功能	无	1280×720 30fps	640×480 30fps	1280×720 30fps	1080p 30fps	1080P
内存	256MB	512MB	256MB	512MB	512MB	1GBRAM
影片播放	M4V、MP4、MOV、Motion JPEG 等					
音乐播放	AAC、Protected AAC、MP3 (8-320kbs)、MP3 VBR、AIFF、WAV 等					
感应器	加速度计、靠近感应、环境亮度感应、陀螺仪 (除 iPad 一代与 iPhone3Gs 没有)					
Siri	无				有	

1.1 iOS 的硬件及其应用

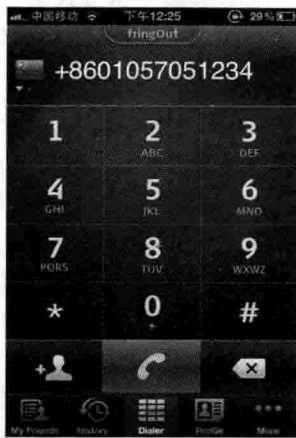


对 iOS 各个硬件间的差异有了一些初步的认识之后,接下来我们就可以进一步了解各个硬件组件以及其应用。由于 iOS 家族包含了 iPod Touch、iPad、iPhone 等硬件产品,因此本节的内容里面并不会特意去区分 iOS 的设备。

1.1.1 上网能力

自从第 2 代的 iPhone 上市后,苹果公司一直强调的就是 3G 这个字眼,以区别于第 1 代的 2G。iPhone 3G 支持的网络规格十分广泛,包括属于 2.5G 的 GPRS (general packet radio service, 通用封包无线服务技术),有时也被称为 2.75G 的 EDGE (enhanced data rates for GSM evolution, GPRS 的延伸标准),称为 3.5G 网络的 HSPDA (high speed downlink packet access),还有无线网络 WIFI (包含 802.11b/g/n) 的标准等。3G (3rd Generation) 泛指第 3 代移动通信技术,有别于第 1 代的模拟手机、第 2 代的数字手机(如 GSM),它可将无线通信与因特网等多媒体通信结合在一起,在室内、室外和行车环境分别提供至少 2Mbit/s、386kbit/s 和 144kbit/s 的传输速率。

由于 3G 的普及,以往在计算机上才可以使用的 MSN、Skype 等实时通信软件也都转移到智能型手机上了,通过像 Skype out 这样的服务就可以让您以便宜的费率来拨打国际电话,如图 1.1 所示。^①



▲ 图 1.1 Fring 上的 fringOut

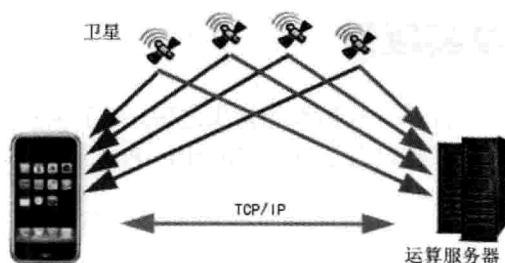
1.1.2 A-GPS

拜节能减碳的风潮所赐,现在全世界都兴起了单车热。玩单车的玩家除了具有基本配备外,GPS 似乎也变成了高级版的必备品。GPS 可以说是 iPhone 的卖点之一,就连 iPad 也都配备了定位的功能。

传统的 GPS 要接收到信号,必须在信号条件极佳的开放天空中接收到四颗以上的卫星才能解析出目前的所在地,如图 1.2 所示。除此之外,还得不间断地接收 30 秒以上的时间。虽然定位十分精准,但是在高楼林立的都市内,传统的 GPS 就不太合适了。

与 GPS 完全依赖卫星来定位不同,A-GPS 则是靠移动电话无线基站的辅助,通过基站上的定位解析运算服务器来进行定位运算,再回传结果给移动电话,因此,A-GPS 的定位并不是十分精准,但是在定位速度与运算速度折衷的情况下,这应该算是个不错的方案。

^① Skype 目前已经不支持 Fring 了。



▲ 图 1.2 A-GPS 示意图

下面说明一下 iOS 定位的方式。其实 iOS 定位的方式并不是只有一种，而是有 3 到 4 种方式。您可以在程序代码中要求定位的精度，若 iOS 无法正确定位，则会使用次一级的定位方式，直到无法定位为止。

首先，iOS 会使用卫星来进行定位，这种方式其实就是传统的 GPS 定位，必须站在空旷的地点才能接收到。如果 Google Map 显示为一个实心的蓝色小圆点，就表示是传统的定位方式；反之如果是空心的一大圈，那就表示是 A-GPS 定位方式。如果卫星定位不到，则会使用 SkyHook Wireless 公司定位技术，它的原理是通过用户使用的 WiFi 或 WiMax 路由器的 Mac 地址，然后在 SkyHook 的数据库内搜寻相对应的数据，从而知道用户的位置。这种方式基本上算是很精准的了，毕竟 iOS 设备和这些路由器的位置不会相差太大。不过由于这种方式依赖的是 SkyHook 庞大的数据库还有 WiFi 路由器的位置，这就产生了一个问题，如果这个路由器是移动的，那么数据库内的数据就没有太大意义了。

其次则是通过移动电话基站来进行三角定位，其原理是搜寻附近的 4 到 5 个移动电话基站，并以它们彼此间的位置来推算出用户现在的位置，这样定位的精确度虽不令人满意，但是勉强可以接受。最后一种方式则是通过搜寻 Internet 服务供应商 (ISP) 在附近机房的位置（这也是 SkyHook 提供的服务）来进行定位，当然，这样的精确度就更差了。所以当用户所在的地方不一样，定位出来的精确度也会有很大差别。

另外要特别说明的是，A-GPS 会联机到服务器以获取卫星数据，虽然其联机仅在启动导航软件之初，并在取得卫星数据后即刻断线，但这还是会有数据传输产生。关于这一点，读者可以在“设置”→“通用”→“用量”的界面中观察到，其数据传输产生的费用则根据各电信业供应商收费不同而有所不同。

在应用部分，目前使用 GPS 的应用程序大致上可以分为导航类软件以及社交软件两大类。例如，Facebook 的打卡服务就可以让您跟朋友分享目前所在位置以及在做什么事，如图 1.3 所示。



▲ 图 1.3 导航系统 GPS 应用

1.1.3 影音播放

影音功能大概是 iOS 除了上网功能外的另一个卖点吧！iOS 支持的格式相当广泛，音频部分包含 AAC、Protected AAC、MP3、MP3 VBR、Audible (formats 2,3,4)、Apple Lossless、AIFF 与 WAV 等，视频部分则支持 MPV、M4V、MP4、MOV、3GP 格式。新版的 iOS5 甚至还提供了 AirPlay，可以让您将 iOS 上的影音传输到另一部 iOS 设备上（包含 Apple TV）。

有关于影音的应用程序其实还有很多，例如 TVUPlayer 就可以让您在 iPhone 上看电视，如图 1.4 所示。

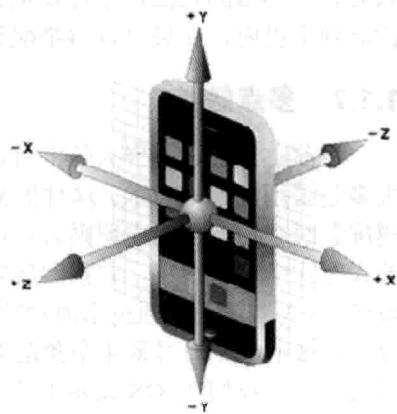


▲ 图 1.4 TVUPlayer

1.1.4 加速度计

加速度计 (accelerometer)，顾名思义就是用来检测物体的加速度的。用过传统的多功能手机 (feature phone) 的读者知道，这些手机的画面几乎都是很死板的，无论怎么转，它的画面也只能朝单一方向来看，然而 iPhone 开始改变了这样刻板的印象——手机的画面会跟着您一起转。这是个很实用的功能。举例来说，浏览网页的时候，网页的画面通常是以矩形居多。以一段英文的文章段落来说，由于书写顺序是由左而右的，因此横向的文字通常会比纵向的文字来得多。如果要使得阅读起来更为舒适，就必须减少眼睛的移动，这时候只要把手机横着看就可以达到这样的目的了。特别要提出的一点是，在 Apple 的中文网站上，加速度计被称为方向感应器，两者其实是同样的东西，在本书中将统一称为加速度计。iOS 的加速度计是由 x 、 y 、 z 3 个轴构成的，如图 1.5 所示。这 3 个轴构成的立体空间可以检测到您在 iOS 上面的各种动作。当 iOS 与这 3 个轴间呈现某个倾斜角度的时候，我们就可以得知分布在不同轴向的重力加速度大小。例如，当我们将 iOS 垂直握住时（也就是与 y 轴平行），在 y 轴就可以感测到 1 个单位 (1G) 的重力加速度。

单独使用加速度计的软件并不多见，大多数是以它作为辅助。尤其是一些用来看小说或漫画的软件都会提供这样的



▲ 图 1.5 加速度计的 3 个轴



转向功能。值得一提的是，由于很多人习惯躺在床上看电子书，这种自动转向的功能就很容易造成用户的困扰，因此，iOS 4.0 以后就增加了可以关闭这个功能的选项。

1.1.5 电子罗盘

电子罗盘是 iPhone 3GS 现身后的卖点之一，购买 iPhone 3GS 的用户会发现，手机多了一个指南针的软件。这个电子罗盘可以用来指示正确的方向，就像是 iPhone 3GS 内建的 Google map 一样，只要轻按一下定位的按钮就可以发现画面上多了方向的指示。

1.1.6 照相机

iPhone 的照相机算是它几个硬件里面比较逊色的了。iPhone 从 1 代到 2 代都维持 200 万像素的水平，没有闪光灯，更别说自动对焦等功能了。3 代的 iPhone 虽然小有长进，不过比起市面上大多数的智能型手机还是有些缺憾的，一直到了 iPhone 4 才有较令人满意的像素出现。

好在苹果公司在软件的部分弥补了不少缺憾，您可以很容易地通过它提供的 SDK 做出一个拍照用途或是相簿的程序，而这并不需要太多的程序代码就可以办到了。

照相机的应用相比之下并不是很多。或许是 iPhone 的照相功能太过于小众了，所以很多的应用程序都是用来取代内建的照相机，如全景照相机 Panorama 可以让您 180 度全景拍摄。另外，也有人将相机应用在二维条形码的读取方面，如图 1.6 所示。在我国台湾等地的很多店家都贴有二维条形码。当您经过的时候，只要用照相机一照，就可以把店家的各项信息读取到手机上。也有人将其应用在旅游网站上，并搭配地图的使用，只要看到不错的景点，就可以把它的二维条形码连同地理信息读取到手机内，如果可以再搭配导航系统的使用把用户引导到这些景点，那就更棒了！



▲ 图 1.6 二维条形码读取软件

1.1.7 多点触控

目前主流的触摸屏大体上可以分为电阻式与电容式两种。以往的手机只要配备有触摸屏，大多是采用电阻式技术。这种屏幕是由两层涂有透明导电物质的玻璃与塑料构成的，当用户触摸屏幕时会将两个导电层贴合使电压产生变化，进而取得当前触控位置。iOS 则采用更为先进的投射式电容触摸屏，它由四层复合玻璃组成，玻璃的内表面和夹层涂有一层 ITO（镀膜导电玻璃）。人体则是这项技术成功的关键之处，当人体接触了 iOS 后，iOS 会从人的手指吸走电流，而这个电流与屏幕 4 个角落的电极距离呈正比，这样一来就可以计算出触摸点的位置，也因为这样，我们在 iOS 设备上几乎看不到触控笔的使用。

乍听之下，这个技术似乎有些复杂，好在我们在处理多点触控的事件时没有这么复杂。因

为 iOS SDK 已经帮你把大部分工作都处理好了，您只要专心处理触控时所发生的事件就可以了。

谈到触控，不得不谈到手势（gesture）。由于 iPhone 本身内建的应用程序已经大量运用了手势，这些手势久而久之就变成用户的一种习惯。常见的手势如轻拍（tap）画面两下通常用于相片快速放大缩小，手指的往内或往外拨动则可以用来缩小或放大画面。读者在设计自己的应用程序时最好也可以遵循这样的习惯。

关于多点触控的应用还有很多，尤其是 iPad 的大尺寸屏幕也带来更多的应用。例如，以往只能单个人在手机上玩游戏的场景，换到 iPad 上之后就可以让两个人同时在同一个画面下玩。

1.1.8 智能感应器

这包括了两个感应器——环境光源感应器与距离感应器，这两项技术也算得上是个创举，但是知道它们的人可能不是很多。开启 iOS 上面的自动亮度的开关后，这组感应器就会被启动。环境光源感应器可以检测到附近光线的明暗程度，从而调整屏幕的亮度。而距离感应器则用在接听电话时，由于用户的耳朵贴近手机，屏幕并不需要显示（应该没有人可以这样听电话又可以看到屏幕吧），而会自动关闭屏幕；一拿开手机后屏幕又会自动出现，以节省电池并且可以防止您误碰到其他地方。目前，开发者在这两个感应器上可以发挥的地方不多，只可以取得其相关信息。

1.1.9 陀螺仪

陀螺仪（gyroscope）是从 iPhone 4 开始加入的硬件组件，当然，iPad 2 也加入了这个硬件组件。加速度计用来测量物体移动的加速度，而陀螺仪则用来测量物体旋转的角速度。它是基于物理学上的角动量守恒原理设计出来的，具有抗拒方向改变的趋向。不管如何倾斜或旋转手机，通过陀螺仪的帮助，iPhone 仍可以辨识出用户的方位。陀螺仪用户数字罗盘在方位上的辨认更为准确，且不受磁性物体干扰。可以在 App Store 下载 Gyroscope 这个应用程序，就能够了解陀螺仪的用处。

1.2 必须了解的事

在介绍了几个硬件组件及其应用后，相信读者早就跃跃欲试了吧。不过，在开始进行 iOS 应用程序的开发前，我们得先了解开发前的一些准备工作以及开发的方式。

1.2.1 挑选适合开发用的计算机

iOS 的开发平台毋庸置疑必须架构在 Mac 平台上，而且操作系统必须为 Snow Leopard 或 Lion（Leopard 或更早的版本已经不被支持了）。这是因为目前 Apple 的开发工具 XCode 只提供这两个版本，如果您尚未成为 Apple 的付费开发者的话，那么您只能下载 XCode 3.x 的版本，已付费的会员则可以免费获得 XCode 4.x 以上版本。