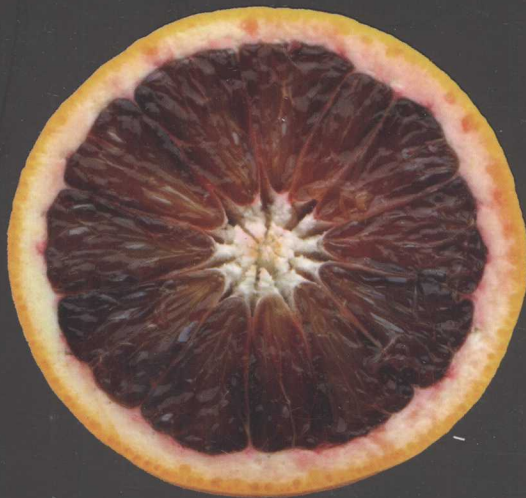


Apress®

Amazon
计算机
畅销图书

- 创造销售奇迹的系列经典著作最新版
- 深度剖析 iOS 6 高级特性与开发案例
- 不只是基础，更在于提高，进阶必备



More iOS 6 Development
Further Explorations of the iOS SDK

iOS 6

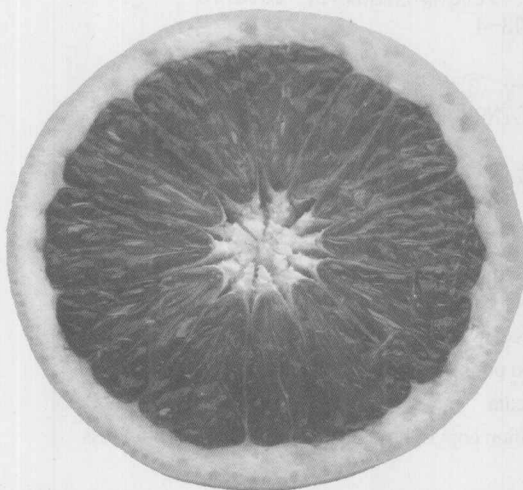
开发进阶与实战

[美] Dave Mark Alex Horovitz 著
Kevin Kim Jeff LaMarche
麦秆创智 译

人民邮电出版社
POSTS & TELECOM PRESS

013071472

TN929.53
536



More iOS 6 Development
Further Explorations of the iOS SDK

iOS 6

开发进阶与实战

[美] Dave Mark Alex Horovitz 著
Kevin Kim Jeff LaMarche

麦秆创智 译



北航

C1680454

人民邮电出版社
北京

TN929.53

536

图书在版编目 (CIP) 数据

iOS 6开发进阶与实战 / (美) 马克 (Mark, D.) 等著
; 麦秆创智译. — 北京: 人民邮电出版社, 2013.10
ISBN 978-7-115-32603-4

I. ①i… II. ①马… ②麦… III. ①移动电话机—应用程序—程序设计 IV. ①TN929.53

中国版本图书馆CIP数据核字(2013)第165977号

版 权 声 明

More iOS 6 Development: Further Explorations of the iOS SDK

By Dave Mark, Alex Horovitz, Kevin Kim, Jeff LaMarche, ISBN: 978-1-4302-3807-2

Original English language edition published by Apress Media.

Copyright © 2012 by Apress Media

Simplified Chinese-language edition copyright © 2013 by Post & Telecom Press

All rights reserved.

本书中文简体字版由 **Apress Media** 公司授权人民邮电出版社独家出版。未经出版者书面许可, 不得以任何方式复制或抄袭本书内容。

版权所有, 侵权必究。

-
- ◆ 著 [美] Dave Mark Alex Horovitz
Kevin Kim Jeff LaMarche
- 译 麦秆创智
- 责任编辑 刘 洋
- 责任印制 彭志环 杨林杰
- ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子邮件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京天宇星印刷厂印刷
- ◆ 开本: 800×1000 1/16
印张: 27.25
字数: 618 千字 2013 年 10 月第 1 版
印数: 1-3 500 册 2013 年 10 月北京第 1 次印刷

著作权合同登记号 图字: 01-2013-3881 号

定价: 79.00 元

读者服务热线: (010)67132692 印装质量热线: (010)67129223

反盗版热线: (010)67171154

广告经营许可证: 京崇工商广字第 0021 号

内容提要

本书从Core Data、iCloud、Game Kit、Map Kit、消息、媒体以及安全性等诸多方面，对iOS平台开发中的进阶内容进行了详细的讲解。本书共16章，其中第2章至第7章详细而深入地讨论了有关Core Data的内容；第8至第13章分别讲解了云服务技术iCloud、Game Kit中的蓝牙连接、LBS服务Map Kit、媒体库与多媒体播放、消息传送与社交媒体，以及应用程序安全性；第14、15章介绍了并发响应、多线程与高级调试技术；最后，第16章为读者提供了一些有用的外部资源。

本书所涉及的内容是开发具有一定深度和价值的iOS应用程序所必须了解和掌握的，所以，本书不仅适合对iOS开发有一定基础的读者，更是即将开发产品级应用的开发人员所必须阅读的书籍。另外，本书也可供领域内产品经理、项目经理等相关职位人士参考。

致 谢

尽管封面上印着我们的名字，但编写这样一本书确实超过了我们几个作者的能力，它是许多人共同努力的结晶。

首先，我要感谢Dave Mark和Jeff LaMarche，他们为本书提供了初稿，从而我可以在此基础上进行深入构建和扩展。

我要感谢Apress的编辑，他们给予了我工期上的最大宽限。Brigid Duffy为我完成本书给予了指导；Tom Welsh提供了文字审阅工作，以保证我的语言清晰流畅；Mary Behr让整本书的版面变得更漂亮。我还要感谢Brandon Levesque为本书问世所打的广告。

感谢技术评审Nick Waynick和Mark Dalrymple，他们保证了我编写的每一段代码都能成功运行；如果还有问题，那就只能怪罪我自己了。

感谢我的朋友和AppOrchard的同事，感谢他们能够在本书即将结束的最后几个月忍受我怪异的脾气，并帮助我完成整个项目。

还要特别感谢我的妻子Annie，当我开小差看棒球或者玩吉他的时候，她能够让我重新返回工作状态。感谢我的猫咪，PK、Manny和Leela，我中途休息时可以给它们喂吃的；其中要特别感谢Manny，它为书中的许多示例提供了素材。

最后，要感谢你这位读者购买了本书。我们一直认为编程是一门科学技术，但实际上它是一种神奇的艺术。如果这本书能够帮助你更好地理解iOS程序开发，那么我的心意就达到了。

献词

谨以本书献给安妮（Annie），以及所有爱她和支持她的人。

译者序

本书的翻译由3名译者共同完成，我们进入信息化和互联网领域多年，创建了从事企业信息化建设和互联网产品研发的北京麦秆创智信息技术有限公司，并一直努力想要和更多人分享我们的思考和认识，同时也将更多国外的优秀书籍引入国内，为国内更多的读者提供帮助。在翻译本书的过程中，我们一直思考着几个问题，那就是移动开发如何在最近这几年的时间产生如此深刻的变革，手机、平板电脑何以成为人们生活、工作和学习中最重要的电子设备，移动互联网又如何影响着整个世界。

许多人睡前最后一秒做的事情是将手机放在枕边，醒来的第一件事情是重新在枕边摸索出手机——没了手机，仿佛跟世界失去了一切联系，活在原始社会。我们无法判断这是人类社会的一种进步还是衰退：进步在于现代人可以利用这些移动终端，借助互联网，高效、准确地获取几乎一切所需；衰退在于，支撑起每人每日生活的居然是一堆电子设备。

当然，讨论进步抑或衰退并不是你我的事情，也不是你我能够改变的。我们需要看到的，或者需要做的——无论你是一名开发人员，还是产品、项目经理，甚至UI/UE设计师、测试人员——在于感受并实践技术和产品所带给我们和整个社会的强大推动力。

不可辩驳，iOS是一个强大的平台，App Store拥有丰富的应用，正是这些应用改变了人们的生活，甚至改变了我们自己。我不禁想到了Symbian（塞班）平台，这个“古老”的智能手机操作系统，就在几年前，它还是用户的骄傲，而现在，已经渐渐被人们所淡忘。如果拿它跟iOS相比较（OK，我知道这么做有点滑稽）就会发现，像iOS、Android这样的平台为什么能够在顷刻之间横扫市场，成为绝对的赢家。

我认为，最主要，也是最重要的一个原因是平台支撑，你也可以比较时髦地把它叫做平台生态系统。何谓生态系统？就是不是只有你一个人存在，必须借助、依赖甚至依附于周围的平台，同时发挥自己的作用。例如，一个简单的应用，在加入了云存储功能之后，可以增强数据的安全保障、实现多设备无缝共享；在接入了位置服务之后，可以提供基于用户所在地点的特定信息、特定功能和特定响应；在借助于社交媒体之后，可以将单点应用拓展成整个网络，还可以顺便带来推广的便利。而这些，就是如今高质量移动应用所必须具备的功能，也是它们在一大堆同质应用中脱颖而出，在应用商店中傲视群雄的关键。当然，也是iOS、Android这两个具有强大生命力的平台战胜Symbian等传统平台的根本。

应用程序本身可以很小，功能也可以很简单，但通过庞大的云服务、LBS、社交媒体等平台，它可以远远超出自身所能及。而整个iOS平台，通过一些开发人员友好的SDK框架，就将简单的

应用与强大的支撑平台连接到了一起，进而，这个“简单的应用”就成了人们生活的必需品，睡前、醒来都不离手。作为产品经理或者开发人员，设计和实现这样的应用是梦寐以求的。而本书，正是为实现这一目的而生。

本书从Core Data、iCloud、Game Kit、Map Kit、MessageUI、Social框架、媒体和安全性等诸多方面，对支撑整个iOS平台的“生态系统”进行了全面而深入的讲解，掌握它们，是每一个想要或者正在实现产品级应用的开发人员的必修课。正是这些，构成了一个成功的“简单应用”，我们也希望你能够借助它们，将梦寐以求的产品变成用户整天不离手的依赖。

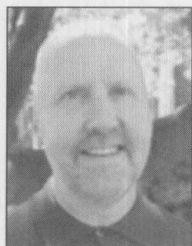
由于译者水平有限，且时间仓促，本书译文难免有错误和疏漏，或者某些翻译不当之处，还望各位读者批评指正。编辑邮箱：liuyang@ptpress.com.cn。

另外，我们还要特别感谢马煜坤、贾学来、童琪颖、周志浩、李科达、张皓博、王璁、罗一峰、王思敏等人在本书的翻译过程中提供的支持和帮助，他们也为本书做出了贡献。

译者

2013年7月于北京

作者简介



戴夫·马克 (Dave Mark) 是一位资深的Mac开发人员, 编写了多本与Mac和iOS开发相关的技术书籍, 包括*Beginning iPhone 4 Development* (Apress, 2011)、*More iPhone 3 Development* (Apress, 2010)、*Learn C on the Mac* (Apress, 2008)、*Ultimate Mac Programming* (Wiley, 1995) 以及*The Macintosh Programming Primer*系列 (Addison-Wesley, 1992)。戴夫是iOS和Android开发工作室MartialCraft的创始人之一, 他与他的妻子和3个孩子一起住在弗吉尼亚 (Virginia)。

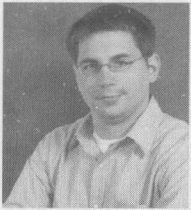


杰夫·拉马赫 (Jeff LaMarche) 是一位拥有20多年编程经验的Mac和iOS开发人员。杰夫写了若干本关于iOS和Mac开发的书籍, 包括*Beginning iPhone 4 Development* (Apress, 2011)、*More iOS 3 Development* (Apress, 2010) 和*Learn Cocoa on the Mac* (Apress, 2010)。杰夫是iOS和Android开发工作室MartialCraft的总负责人。他为MacTech杂志写过不少关于Cocoa和Objective-C的文章, 还有一些发布在苹果的开发者网站上。杰夫同时也在自己的博客上发表了关于iOS开发的文章: www.iphonedevdevelopment.blogspot.com。



凯文·金 (Kevin Kim) 是AppOrchard LLC的联合创始人和开发人员, 由Tipping Point Partners投资, 专注于可持续的iOS开发。从卡内基·梅隆大学毕业后, 凯文以一名开发人员的身份在Pittsburgh Supercomputing Center接触到了NeXTStep计算机 (iPhone的前身), 并从此被它深深地迷住了。他的事业跨度很广, 涉及金融、政府、生物和计算机技术, 苹果公司也是他的客户之一——他曾担任过纽约地铁的苹果企业服务团队的经理。凯文同时也是*Pro iOS 5 Tools* (Apress, 2011) 一书的联合作者。他目前和妻子以及一群被收养的猫咪一起居住在纽约的字母城 (Alphabet City)。

技术评审人简介



尼克·威尼科 (Nick Waynik) 已经在IT领域工作了超过13年, 工作内容几乎覆盖所有方面——从网络管理员到网站开发人员。自从苹果第一次发布SDK起, 尼克就开始编写iOS应用。从那时起, 他开始运营自己的事业, 专注于iOS开发。在闲暇时, 他喜欢和妻子Allison、儿子Preston在一起, 有时还会去打高尔夫。他的博客是nickwaynik.com, Twitter账号为@n_dubbs。

目 录

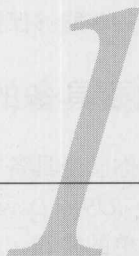
第 1 章 踏上新征程	1
1.1 本书讲什么	1
1.2 你需要具备的基础	2
1.3 启程前的准备	2
1.4 本书主要内容	4
第 2 章 Core Data 剖析	6
2.1 Core Data 简史	6
2.2 创建 Core Data 应用程序	7
2.3 Core Data 概念和术语	11
2.4 数据模型	11
2.5 Managed Object	21
2.6 键—值 (Key-Value) 编程	21
2.7 Managed Object 上下文	22
2.8 小结	33
第 3 章 完美的开端: 添加、显示和 删除数据	34
3.1 建立 Xcode 项目	35
3.1.1 添加场景	39
3.1.2 场景和导引线	40
3.1.3 Storyboard 文档大纲	40
3.2 应用程序架构	41
3.3 设计视图控制器界面	42
3.4 创建 HeroListController	45
3.4.1 创建 Connection 和 Outlets	47
3.4.2 导航栏按钮	49
3.4.3 标签栏 User Defaults	51
3.5 设计数据模型	53
3.5.1 添加一个实体	53
3.5.2 编辑新实体	53
3.5.3 为 Hero 实体添加属性	55
3.6 声明 Fetched Results Controller	59
3.7 实现 Fetched Results Controller	60
3.8 Fetched Results Controller 的 Delegate 方法	61
3.9 一切就绪	63
3.9.1 错误处理	63
3.9.2 实现 Edit 和 Add	63
3.9.3 为表格视图数据源和 Delegate 编写代码	64
3.9.4 表格视图排序	65
3.9.5 运行时载入 Fetch Request	66
3.10 跑起来吧	66
3.11 完成, 但还要继续	67
第 4 章 隐藏在详情视图中的“怪兽”	68
4.1 视图实现方案选择	68
4.2 创建详情视图控制器	70
4.2.1 接通导引线	72
4.2.2 HeroDetailController	73
4.3 详情视图中的挑战	74
4.4 使用 Property 列表控制表格结构	75
4.5 使用 Property 列表搭建表格结构模型	76
4.6 通过 Property 列表定义表格视图	77
4.7 解析 Property 列表	80
4.8 显示详情	82
4.9 编辑详情	84
4.10 详情视图的编辑模式	84
4.11 创建自定义 UITableViewCell 子类	86
4.12 保存修改	89
4.13 专门的输入视图	90

4.13.1 DatePicker SuperDBEditCell 子类	90	6.8 创建 Color 表格视图单元格	127
4.13.2 使用 DatePicker SuperDBEditCell 子类	92	6.8.1 自定义颜色编辑器	127
4.13.3 实现一个选择器	94	6.8.2 自定义 Color 表格视图 单元格	129
4.14 “怪兽”的终结	97	6.9 清除选择器	132
第 5 章 为变化做准备：迁移和 版本控制	98	6.10 还有一件事	135
5.1 关于数据模型	99	6.11 前路更多彩	138
5.1.1 数据模型会被编译	99	第 7 章 关系、属性获取和表达式	139
5.1.2 数据模型可以拥有多个版本	100	7.1 扩展你的应用程序：“超能力”和 “英雄榜”	139
5.1.3 创建新版数据模型	101	7.2 关系	141
5.1.4 当前数据模型版本	102	7.2.1 对一关系	142
5.1.5 数据模型版本标识符	103	7.2.2 对多关系	142
5.2 迁移	103	7.2.3 逆关系	144
5.2.1 轻量 VS 标准	103	7.2.4 Fetched Properties	145
5.2.2 标准迁移	104	7.2.5 在数据模型编辑器中创建关系 和 Fetched Properties	145
5.3 为轻量迁移设置应用	104	7.2.6 删除规则	146
5.4 让我们迁移吧	105	7.3 表达式和融合	146
第 6 章 自定义管理对象	106	7.4 添加“超能力”实体	147
6.1 更新数据模型	108	7.5 创建 Power 关系	148
6.1.1 添加 Age 属性	109	7.6 创建逆关系	149
6.1.2 添加“最喜爱颜色”属性	109	7.7 创建 olderHeros Fetched Property	149
6.1.3 为 name 属性指定最小长度	110	7.8 创建 youngerHeros Fetched Property	151
6.2 创建 Hero 类	111	7.9 创建 sameSexHeroes Fetched Property	151
6.2.1 修改 Hero 的 Header	113	7.10 创建 oppositeSexHeroes Fetched Property	151
6.2.2 设置默认值	113	7.11 向 Hero 类添加关系和 Fetched Properties	152
6.3 校验	114	7.12 修改详情视图	152
6.3.1 单属性校验	115	7.12.1 重新思考配置	154
6.3.2 nil vs. NULL	116	7.12.2 封装和信息隐藏	157
6.3.3 多属性校验	117	7.12.3 数据驱动配置	159
6.4 虚拟 Accessor	118	7.12.4 添加“超能力”	160
6.5 添加校验反馈	118	7.13 重构详情视图控制器	165
6.6 更新详情视图	120	7.13.1 重命名 Configuration 类	165
6.7 重构 SuperDBEditCell	122	7.13.2 重构详情控制器	167
6.7.1 Xcode 重构选项	122	7.13.3 重构 Hero 实例变量	167
6.7.2 搬移代码	124		
6.7.3 可编辑 Property	126		

7.13.4 再抽象一点	167	9.6.1 创建 Session	210
7.13.5 新的 HeroDetailController	169	9.6.2 查找并连接到其他 Session	211
7.14 “超能力”视图控制器	171	9.6.3 监听其他的 Session	211
7.15 Fetch Properties	174	9.6.4 向连接点发送数据	212
7.16 和 Core 说再见	177	9.6.5 信息打包发送	213
第 8 章 在云端	178	9.6.6 从连接点接收数据	213
8.1 使用 iCloud 进行数据存储	178	9.6.7 关闭连接	214
8.2 iCloud 基础	178	9.7 Peer Picker	214
8.3 iCloud Backup	179	9.7.1 创建 Peer Picker	214
8.4 在应用程序中启用 iCloud	180	9.7.2 处理连接	215
8.5 键—值数据存储	180	9.7.3 创建 Session	215
8.6 文档存储	181	9.8 创建项目	216
8.6.1 UIDocument	181	9.8.1 关闭待机计时器	216
8.6.2 支持 iCloud 的 UIDocument	185	9.8.2 导入 Game Kit 框架	216
8.6.3 NSMetadataQuery	185	9.8.3 设计界面	217
8.7 支持 iCloud 的 Core Data	187	9.8.4 定义应用程序常量	217
8.8 升级 SuperDB	189	9.8.5 设计棋盘	219
8.8.1 Entitlements	189	9.8.6 创建数据包对象	221
8.8.2 创建启用 iCloud 的 Provisioning Profile	190	9.8.7 搭建视图控制器头	223
8.8.3 修改 Persistent Store	198	9.9 内测	234
8.8.4 修改 Managed Object 上下文	200	9.10 游戏开始	235
8.8.5 收到 DataChanged 通知时 更新 UI	201	第 10 章 Map Kit	236
8.8.6 测试数据存储	202	10.1 本章的应用程序	236
8.8.7 “脚踏实地”	202	10.2 概览和术语	237
第 9 章 使用 Game Kit 进行蓝牙 点对点通信	203	10.3 地图视图	238
9.1 Game Center	203	10.3.1 地图类型	238
9.2 点对点连接	204	10.3.2 用户所在位置	239
9.3 In Game Voice	205	10.3.3 坐标区域	239
9.4 本章的应用程序	205	10.3.4 设定显示区域	241
9.5 网络通信模型	208	10.3.5 地图视图 Deleate	241
9.5.1 客户端—服务器模型	208	10.4 标注	243
9.5.2 点对点模型	209	10.4.1 标注对象	243
9.5.3 客户端—服务器与 点对点混合	209	10.4.2 标注视图	244
9.6 Game Kit 中的 Session	210	10.4.3 添加和移除标注	244
		10.4.4 选中标注	244
		10.4.5 提供带有标注视图的 地图视图	245
		10.5 地理编码和反向地理编码	246
		10.6 搭建 MapMe 应用程序	247

10.6.1	链接 Map Kit 和 Core Location 框架	247	12.1.5	音乐播放控制器	279
10.6.2	构建界面	248	12.2	简单的音乐播放器	283
10.6.3	实现视图控制器接口	249	12.2.1	构建 SimplePlayer 应用程序	283
10.6.4	实现标注对象类	249	12.2.2	构造用户界面	284
10.6.5	实现 MapMe 的 ViewController	251	12.2.3	声明 Outlet 和 Action	286
10.7	升级啦, 年轻人	256	12.3	MPMoviePlayerController	291
第 11 章	消息: 邮件、短信和社交媒体	257	12.4	MPMediaPlayer	292
11.1	本章的应用程序	257	12.5	AVFoundation	302
11.2	MessageUI 框架	259	12.6	AVMediaPlayer	303
11.2.1	创建邮件编写视图控制器	260	12.7	小心! 前方有洪水	327
11.2.2	填写标题行	260	第 13 章	加把锁: iOS 安全	328
11.2.3	填写收件人	260	13.1	安全性考量	328
11.2.4	设置邮件正文	261	13.2	安全性技术	329
11.2.5	添加附件	261	13.2.1	加密	329
11.2.6	显示邮件编写视图	261	13.2.2	散列	329
11.2.7	邮件编写视图控制器的 Delegate 方法	261	13.2.3	证书和签名	330
11.2.8	消息编写视图控制器	262	13.2.4	身份	330
11.3	Social 框架	263	13.3	iOS 安全性	330
11.3.1	SLComposeViewController	263	13.3.1	随机 Keychain	331
11.3.2	SLRequest	264	13.3.2	证书、密钥和受信服务	335
11.4	活动视图控制器	265	13.4	Keychain Viewer	336
11.5	构建 MessageImage 应用程序	266	13.4.1	创建 Certificate Authority	336
11.5.1	构建用户界面	266	13.4.2	创建 Keychain 应用	340
11.5.2	拍照	266	13.5	安全不打烊	361
11.5.3	调用照相机	267	第 14 章	保持界面响应	362
11.5.4	选择消息发送器	267	14.1	探索并发问题	363
11.6	寄往: 下一章	268	14.2	创建 Stalled 应用程序	364
第 12 章	媒体库访问和播放	269	14.2.1	设计界面	364
12.1	MediaPlayer 框架	269	14.2.2	实现 Stalled 视图控制器	364
12.1.1	媒体项	270	14.3	Timer	367
12.1.2	媒体项集合	274	14.3.1	创建 Timer	367
12.1.3	媒体查询和媒体属性 Predicate	275	14.3.2	停止 Timer	368
12.1.4	媒体选择控制器	277	14.3.3	Timer 的局限性	368
			14.4	使用 Timer 修改 Stalled 应用程序	368
			14.4.1	创建 Batch 对象	368
			14.4.2	修改 nib	370
			14.4.3	修改视图控制器头文件	370

14.4.4 修改视图控制器的 实现文件·····	371	15.2.3 Debug Area ·····	405
14.5 操作队列和并发·····	373	15.2.4 尝试 Debug Control·····	406
14.5.1 线程·····	374	15.2.5 Breakpoint Navigator 和 Symbolic 断点·····	409
14.5.2 操作·····	379	15.2.6 条件断点·····	409
14.5.3 操作队列·····	381	15.2.7 断点行为·····	411
14.6 使用操作队列优化 Stalled 应用程序·····	382	15.2.8 静态分析·····	413
14.6.1 创建 SquareRootApplication ···	383	15.2.9 有关调试的补充·····	413
14.6.2 自定义 ProgressCell ·····	386	15.3 使用 Instruments 进行性能分析·····	414
14.6.3 调整用户界面·····	386	15.4 胜利的终点线·····	416
14.6.4 修改 ViewController.h·····	387	第 16 章 路在脚下·····	417
14.6.5 修改 ViewController.m·····	387	16.1 逃离困境·····	417
14.7 排队站好·····	392	16.2 苹果提供的文档·····	418
第 15 章 单元测试、调试和相关工具·····	393	16.3 邮件列表·····	418
15.1 单元测试·····	393	16.4 论坛·····	418
15.2 调试·····	401	16.5 网站·····	418
15.2.1 断点·····	402	16.6 博客·····	419
15.2.2 Debug Navigator·····	404	16.7 如果一切都失败了·····	419
		16.8 再见·····	419



你仍然在从事iPhone应用程序开发，对吧？真是太好了！iOS和App Store已经赢得了极大的成功，它改变了移动应用的发布方式，甚至完全颠覆了移动设备的功能。自从2008年3月第一版iOS SDK发布以来，苹果就始终致力于为其移动设备添加新的功能，同时对已有功能进行提升和改进。回看最初的苹果平台，至今仍令人激动不已，不过事实上，随着苹果不断地扩大功能范围，以及引入像我们这样的第三方开发人员，其各方面表现都令人为之赞叹。

从本书上一版*More iPhone 3 Development* (Apress, 2010) 出版以来，苹果又发布了好几个框架、工具和服务。包括但不限于：

(1) **Core框架**：Core Motion、Core Telephony、Core Media、Core View、Core MIDI、Core Image和Core Bluetooth。

(2) **Utility框架**：Event Kit、Quick Look Framework、Assets Library、Image I/O、Printing、AirPlay、Accounts以及Social Framework、Pass Kit。

(3) **服务及其框架**：iAds、Game Center、iCloud、Newsstand。

(4) **以开发人员为中心的增强特性**：Blocks、Grand Central Dispatch (GCD)、Weak Linking Support、Automatic Reference Counting (ARC)、Storyboard、Collection View、UI State Preservation、Auto Layout、UIAutomation。

很显然，对于一本书而言，要容纳这么多的内容是不太现实的，所以，本书会尽可能地详细讲解那些对你而言非常重要的内容。

1.1 本书讲什么

这本书将指导你进一步沿着iOS开发之路向前迈进，创造出更好的应用程序。在*Beginning iOS 6 Development* (Apress, 2012) 一书中，我们的目标是让你完成基础内容的学习，帮助你构建第一个iOS应用程序。在本书中，我们假设你已经掌握了足够的基础知识，所以，除了向你展示如何使用几个新的iOS API之外，还将安排一些高级技术的讲解，从而铺平你开发大型、复杂iOS应用程序的道路。

在*Beginning iOS 6 Development*一书中，每一章的内容都相对独立，各自完成一个或多个项目。本书的后半部分将沿用这种模式，但在第2至第7章，将集中关注并逐步扩充有关Core Data的一个

应用，每章涵盖其中一个方面的内容，并对应用程序加以完善。同时，随着应用程序逐渐变大，还会着重讨论一些能够保持其可维护性的技术，以免程序变得笨拙而难以管理。

1.2 你需要具备的基础

本书假设你已经具备了一定的程序开发能力，并理解iOS SDK的基础内容——你可能已经学习了*Beginning iOS 6 Development*一书，或者其他讲解基础知识的资料。本书还假设你已经使用SDK进行了简单的实践，亲自编写过一两个小程序，并对Xcode有一定的切身体验。如果你已经有些生疏，那么可以快速回顾一下*Beginning iOS Development*一书中第2章的内容。

对iOS完全陌生？

如果你完全是一位iOS开发新手，那么在你阅读本书之前还可以看看其他的书籍。如果你没有编程基础，或者不懂C语言的语法，那么可以试试David Mark和James Bucanek编写的*Learn C on the Mac for OS X and iOS* (Apress, 2012)这本书，它全面地讲解了Macintosh平台下的C程序开发。(www.apress.com/9781430245339)

如果你对C语言有一定的了解，但对面向对象的开发思想比较陌生，那么可以学习*Learn Objective-C on Mac* (Apress, 2012)，这本书非常不错，深入浅出地讲解了Objective-C，其作者都是Mac程序开发方面的专家：Scott Knaster、Wagar Malik以及Mark Dalrymple。(www.apress.com/9781430218159)

接下来，进入Apple iPhone Development Center，并下载一份*The Objective-C 2.0 Programming Language*，它非常详细而全面地讲解了Objective-C这门语言，并且还可以在下面这个地址找到相关的语言参考：<http://developer.apple.com/library/ios/#documentation/Cocoa/Conceptual/ObjectiveC/Introduction/introObjectiveC.html>。

在你掌握了Objective-C之后，就可以开始学习iOS SDK的基础知识了。为此，你应该首先阅读本书的基础篇：David Mark、Jack Nutting、Jeff LaMarche和Fredrik Olsson所编写的*Beginning iOS 6 Development: Exploring the iOS SDK*一书 (Apress, 2011, www.apress.com/9781430245124)。

1.3 启程前的准备

在你可以编写iOS设备软件之前，还需要做一些准备工作。对于新手，你需要一个运行Lion (Mac OS X 10.7或更新版本)平台的Macintosh系统，2008年以后推出的任何一款Macintosh计算机——无论是笔记本还是台式机——都可以满足要求，不过有一点需要注意，机器使用的应该是英特尔芯片，否则将无法运行Lion。

另外显然，你得拥有一个iPhone (3GS或更高)、iPod Touch (第3代或更新)或者iPad (iPad 2或更新)。虽然许多代码都可以在iPhone/iPad模拟器中进行测试，但并不是所有程序都能在模拟器中运行，况且，在对外发布你的应用程序之前，一定要在实际设备上对其进行完整的测试。