

- 世界知名iOS开发专家详细讲解创建优秀iOS移动应用程序的技术细节，系统阐释如何高效开发iOS程序
- 用直观的代码演示当前流行的编程技巧，为iOS开发者提供行之有效的解决方案

iOS核心 开发手册

(原书第5版)

[美] Erica Sadun Rich Wardwell 著 爱飞翔 译



The
Core iOS
Developer's Cookbook
Fifth Edition



机械工业出版社
China Machine Press

iOS核心 开发手册

The Core iOS
Developer's Cookbook
Fifth Edition

(原书第5版)

[美] Erica Sadun Rich Wardwell 著 爱飞翔 译



机械工业出版社
China Machine Press

图书在版编目 (CIP) 数据

iOS 核心开发手册 (原书第 5 版)/(美) 萨顿 (Sadun, E.), (美) 沃德韦尔 (Wardwell, R.) 著; 爱飞翔译. —北京: 机械工业出版社, 2015.3
(iOS/ 苹果技术丛书)

书名原文: The Core iOS Developer's Cookbook, Fifth Edition

ISBN 978-7-111-49185-9

I. i… II. ①萨… ②沃… ③爱… III. 移动终端—应用程序—程序设计—手册 IV. TN929.53-62

中国版本图书馆 CIP 数据核字 (2015) 第 015570 号

本书版权登记号: 图字: 01-2014-4435

Authorized translation from the English language edition, entitled *The Core iOS Developer's Cookbook, Fifth Edition* 9780321948106 by Erica Sadun, Rich Wardwell, published by Pearson Education, Inc., Copyright © 2014.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc.

Chinese simplified language edition published by Pearson Education Asia Ltd., and China Machine Press Copyright © 2015.

本书中文简体字版由 Pearson Education (培生教育出版集团) 授权机械工业出版社在中华人民共和国境内 (不包括中国台湾地区和香港、澳门特别行政区) 独家出版发行。未经出版者书面许可, 不得以任何方式抄袭、复制或节录本书中的任何部分。

本书封底贴有 Pearson Education (培生教育出版集团) 激光防伪标签, 无标签者不得销售。

iOS 核心开发手册 (原书第 5 版)

出版发行: 机械工业出版社 (北京市西城区百万庄大街 22 号) 邮政编码: 100037

责任编辑: 关 敏

责任校对: 董纪丽

印 刷: 北京市荣盛彩色印刷有限公司

版 次: 2015 年 3 月第 1 版第 1 次印刷

开 本: 186mm×240mm 1/16

印 张: 35.25

书 号: ISBN 978-7-111-49185-9

定 价: 119.00 元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

客服热线: (010) 88378991 88361066

投稿热线: (010) 88379604

购书热线: (010) 68326294 88379649 68995259

读者信箱: hzsj@hzbook.com

版权所有·侵权必究

封底无防伪标均为盗版

本书法律顾问: 北京大成律师事务所 韩光/邹晓东

本书是一本很受欢迎的技术书，从第 1 版到第 5 版，虽然书名和编排形式略有调整，但是 Cookbook 这个特点（本书英文版书名为《The Core iOS Developer's Cookbook》）却一直在延续。

这种食谱式的秘诀手册之所以受欢迎，原因之一就在于开发者非常容易通过范例来学习一门语言或一套开发环境的用法。有了生动易懂的范例，我们很快就能明白如何制作出简单实用的程序。再经过反复练习，即可逐渐掌握开发流程。

另外，秘诀手册与普通的范例教程还有个非常重要的区别，那就是范例的选择。一本优秀的秘诀手册不仅要涵盖中级和高级范例，而且还要选择有代表性的范例，使读者在学会该范例之后，能够根据实际需要对其加以修改，以便将它快速整合到自己的程序中。

从上述两方面来看，本书都极好地体现了 Cookbook 的特点。作者根据长年积累的开发经验，将一百多条解决方案划分为 iOS 开发中的十几个专题，基本上满足了日常编程的需要。有些方案讲得尤为细致，使读者可以看到如何按照需求来深度定制各种组件，比如第 6 章的文本编辑器和第 7 章的自制容器等。除了冠以解决方案之名的范例项目外，还有一些以程序清单形式出现的代码片段也比较实用，每段代码都相当于一份小巧的模板，演示了某些对象或技术的配置方式。

综观这十几个专题，很多都与图形界面及手势有关，这对于提升程序流畅度和用户体验大有好处。作者还提供了范例项目的源代码。对于较为复杂的项目来说，我们可以先运行范例程序，看到实际效果，然后再与书中各步骤详细比对，以了解整个流程；也可以自己先尝试一种实现方式，然后与作者给出的方案或其他教程里给出的方案相比较，看看有何异同，并讨论各自的优缺点。

虽说本书所举的范例都较为浅显直观，但读者还是需要略微懂一些 Objective-C 语言和 iOS 开发的基础知识。如果你是新手，不妨先参考一下作者在前言里所推荐的几本入门书籍。此外，如果对后面的 Core Data、网络编程、设备适配及辅助功能等内容感兴趣，可以参考与这些主题有关的专著，以便深入研究。

在翻译过程中，得到机械工业出版社华章公司诸位编辑与工作人员的帮助，在此深表谢意，并感谢 goldlion 及 ChenGe 两位友人对术语翻译所提的建议。

由于译者水平有限，错误与疏漏之处在所难免，恳请各位读者批评指正。可发邮件至 eastarstormlee@gmail.com 与我联系，也可访问网页 <https://github.com/jeffreybaoshenlee/zh-translation-errata-cidc5/issues> 留言。

爱飞翔

欢迎阅读本书新版!

自苹果公司发行 iOS 移动操作系统以来, iOS 7 的变革是最为重大的。这本教程将会指导各位开发者针对这个新发布的优秀操作系统来制作应用程序。2013 年的全球软件开发年会 (Worldwide Developers Conference, WWDC) 公布了一些新的特性和视觉范式, 而本书这次修订已经将它们全都涵盖在内了, 笔者将会向你演示如何将其融入自己的应用程序里。

发行团队将这次修订过的 Cookbook 材料分成两本书来印刷, 以控制每本书的篇幅。本书的英文书名叫作《*The Core iOS Developer's Cookbook*》, 讲述了日常开发所需的关键知识, 介绍了使用标准 API 与界面元件来创建 iOS 应用程序时所需用到的类。同时, 本书以“解决方案”的形式讲解创建移动应用程序时所需的图像、触摸及视图等技术。

另外一本书的英文名叫作《*Learning iOS Development: A Hands-on Guide to the Fundamentals of iOS Programming*》, 其中包含了一些入门知识, 相当于老版本 Cookbook 的前面几章。该书适合想要从头学习 iOS 7 基础知识的开发者阅读。《*Learning iOS Development*》讲述了 Objective-C 编程语言、Xcode 开发环境以及与调试和部署有关的内容, 你可以通过它学会如何使用苹果公司的开发工具套件。

学习本书所需的材料和知识

想开发 iOS 应用程序, 肯定得有一台测试应用程序用的 iOS 设备, 而且最好是一台新款的 iPhone 或 iPad。下面列出阅读本书所需的材料和基础知识:

- 苹果公司的 iOS SDK——你可以从苹果公司的 iOS Dev Center (<http://developer.apple.com/ios>) 下载最新版的 iOS SDK。如果打算在 App Store 上发售应用程序, 那么还必须成为付费的 iOS 开发者。个人开发者每年需要付款 99 美元, 企业开发者每年需要付款 299 美元。注册成为付费开发者之后, 就会收到一份证书, 开发者可以用这份证书来签署应用程序, 并将其下载到 iPhone、iPod touch 或 iPad 中进行测试与调试, 此外, 付费开发者还可以提前获得预览版的 iOS 系统。未付费的开发者可以在 Mac 系统的模

拟器上测试软件，但是不能将其部署到设备中，也不能将其提交到 App Store。

- **运行 Mac OS X Mountain Lion (v 10.8) 系统的新款 Mac**，如果装的是 Mac OS X Mavericks (v 10.9) 系统就更好——你需要为软件开发留出足够的磁盘空间，而且应该尽可能把 Mac 的 RAM 装配得大一些。
- **iOS 设备**——尽管 iOS SDK 里的模拟器也能用来测试应用程序，但是开发者仍然需要一台 iOS 硬件设备，以便针对 iOS 平台做开发。你可以把 iOS 设备与电脑相连，并把自己构建的软件装上去。在开发真实的 App Store 程序时，应该准备数款硬件和固件各不相同的设备，以便在目标用户可能会用到的各种平台上进行测试。
- **因特网连接**——连上网之后，就可以测试应用程序在使用 WiFi 和使用移动数据网络时的效果了。
- **熟悉 Objective-C**——要想编写 iPhone 程序，需要了解 Objective-C 2.0。这是一门基于 ANSI C 的语言，并且带有面向对象扩展，也就是说，你需要了解一点 C 语言的知识。如果原来用 Java 或 C++ 写过程序，而且又熟悉 C 语言，那么应该能够迅速适应 Objective-C。

学习 Mac/iOS 开发的路线图

一本书不可能把各类读者所需的全部知识都囊括其中。假如两位作者把你所需的全部内容都写到这本书里，那它会重到根本拿不起来。实际上，想要开发 Mac 及 iOS 平台上的程序，需要学习很多内容。如果你刚处于起步阶段，而且没有写过程序，那么首先应该学习一门大学水平的 C 语言课程。大部分编程语言都以 C 语言为根基，对于想要成为开发者的人来说，自然也要从 C 语言学起。

学会 C 语言及编译器（基础的 C 语言课程会讲到它）的用法之后，剩下的事情就简单多了。此时可以直接跳到 Objective-C 语言，学习如何用它来编程，同时还可以学习 Cocoa 框架。图 1 是一张流程图，里面列出了培生教育出版集团所出版的一些关键书籍^①，它们可以帮助你成为一名熟练的 iOS 开发者。

了解 C 语言之后，就可以通过多种方式来学习 Objective-C 编程了。如果想深入了解 Objective-C，那么可以阅读苹果公司自编的文档，也可以翻看下列 Objective-C 教材：

- 《*Objective-C Programming: The Big Nerd Ranch Guide*》，第 2 版，Aaron Hillegass 与 Mikey Ward 著（Big Nerd Ranch，2013 年）
- 《*Learning Objective-C 2.0: A Hands-on Guide to Objective-C for Mac and iOS Developers*》，第 2 版，Robert Clair 著（Addison-Wesley，2012 年）^②

① 《The C Programming Language (Second Edition)》一书由 Brian W. Kernighan 及 Dennis M. Ritchie 著，中文版叫作《C 程序设计语言》，由机械工业出版社于 2004 年出版，影印版由机械工业出版社于 2006 年出版。《Xcode 5 Start To Finish》一书由 Fritz Anderson 所著，由 Addison-Wesley Professional 出版社于 2014 年出版。——译者注

② 该书中文版名为《Objective-C 2.0 Mac 和 iOS 开发实践指南》，李强等译，机械工业出版社 2012 年出版。——译者注

■ 《Programming in Objective-C 2.0》，第6版，Stephen Kochan 著（Addison-Wesley，2012年）^①

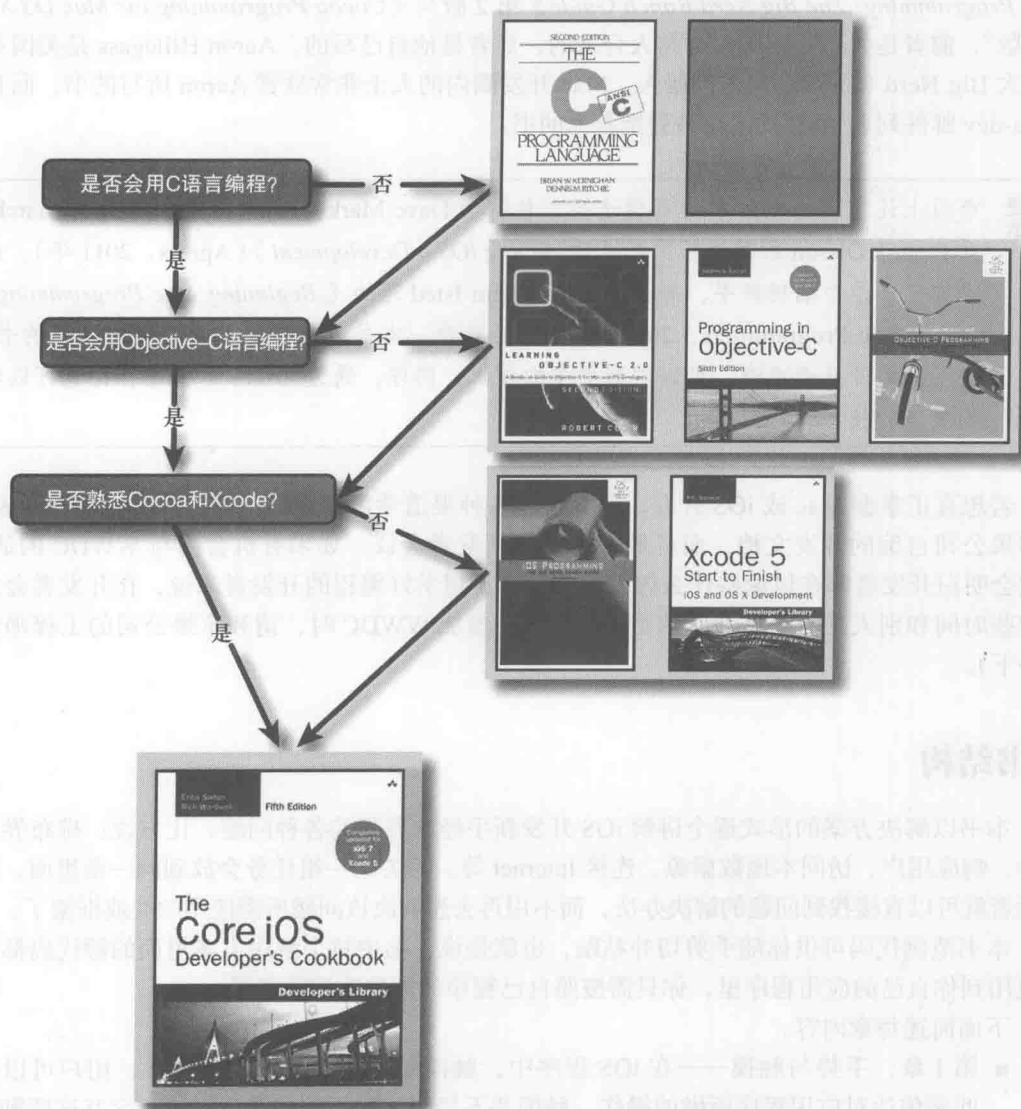


图1 成为iOS开发者的路线图

学会编程语言之后，接下来应该学习 Cocoa（适用于 Mac 开发）、Cocoa Touch（适用于 iOS 开发）以及开发工具，这个开发工具指的就是 Xcode。这一阶段也有几种不同的学习途

① 该书中文版名为《Objective-C 2.0 程序设计（原书第2版）》，张波，黄湘琴等译，机械工业出版社2009年出版。——译者注

径。你依然可以在苹果 Developer 查阅苹果公司自编的 Cocoa、Cocoa Touch 及 Xcode 文档 (网址是: developer.apple.com), 也可以通过阅读书籍来学习它们。笔者推荐两本经典的教材: 《iOS Programming: The Big Nerd Ranch Guide》第 2 版与《Cocoa Programming for Mac OS X》第 4 版^①, 前者是 Aaron Hillegass 与人合著的, 后者是他自己写的, Aaron Hillegass 是美国亚特兰大 Big Nerd Ranch 公司的创始人。Mac 开发圈内的人士非常欣赏 Aaron 所写的书, 而且 cocoa-dev 邮件列表^②里面的人也最爱推荐他的书。



市面上还有非常多的书籍可供选择, 包括由 Dave Mark、Jack Nutting、Jeff LaMarche 及 Fredrik Olsson 所著的畅销书《Beginning iOS 6 Development》(Apress, 2011 年)。如果你完全是个编程新手, 那么可以看看 Tim Isted 写的《Beginning Mac Programming》(Pragmatic Programmers, 2011 年)。不要只看一本书, 也不要只看一家出版社的书。和不同的开发者交流, 可以学到更多的知识, 同理, 通过阅读多本书籍, 你也可以学到更多的技巧和窍门。

若想真正掌握 Mac 或 iOS 开发, 还需通过各种渠道学习: 看书、看博客、看邮件列表、看苹果公司自编的开发文档, 而且最好能参加开发者会议。如果有机会参与 WWDC 的话, 你就会明白开发者都在讨论些什么内容。对于真正想学好编程的开发者来说, 在开发者会议上花些时间和别人交流是件非常值得的事情 (在参加 WWDC 时, 请和苹果公司的工程师交流一下)。

本书结构

本书以解决方案的形式逐个讲解 iOS 开发新手经常遇到的各种问题, 比方说: 排布界面元件、响应用户、访问本地数据源、连接 Internet 等。相关的一组任务会放到同一章里面, 这样读者就可以直接找到问题的解决办法, 而不用再去想解决该问题所需使用的类或框架了。

本书范例代码可供你随手剪切并粘贴, 也就是说, 书中每个解决方案里面的源代码都可以复用到你自己的应用程序里, 你只需按照自己程序的需求来调整即可。

下面简述每章内容。

- 第 1 章, 手势与触摸——在 iOS 程序中, 触摸是一种非常重要的手段, 用户可以由此来传达对应用程序所做的操作。触摸并不局限于按下按钮及通过键盘交互这两种行为。本章将介绍直接操纵界面、多点触摸以及其他一些内容。你将会学到如何创建这样一种视图: 用户可以在屏幕上试验各种手势, 并看到不同手势之间的区别。另外, 本章还会告诉你如何创建自定义的手势识别器。

① 该书中文版名为《苹果开发之 Cocoa 编程 (原书第 4 版)》, 黄敏, 郝刚等译, 机械工业出版社 2012 年出版。——译者注

② 可访问 <https://lists.apple.com/archives/cocoa-dev/> 查阅其内容。——译者注

- 第2章，构建并使用控件——本章将深入讲解如何操控应用程序。你将会详细了解控件的运作机制，还能学会以多种方式来构建并自定义控件。这一章包含很多解决方案，有的比较简单，有的比较复杂，你可以把它们复用到自己的程序里。
- 第3章，提醒用户——iOS 提供了多种在屏幕上向用户显示信息的方式，比如弹出式对话框、进度条、本机通知（local notification）、popover 和 audio ping 等。本章将会讲解如何在应用程序中实现这些信息通知手段，以帮助读者用更多的方式向用户显示信息。本章将介绍这些类的基本使用方法，另外还会提供一些解决方案，使你可以通过基于块的 API（blocks-based API）来轻松地处理与警示信息有关的交互操作。
- 第4章，编排视图及其动画效果——UIView 类及其子类可用来填充 iOS 设备的屏幕。本章将会从头开始讲解视图。与视图有关的解决方案会分别演示如何获取视图对象、如何制作视图的动画效果以及如何操纵视图对象。你将会学到怎样构建、检视及分解视图层级，并了解多个视图是如何组织起来的。通过学习本章，你会发现在图形界面中创建并摆放视图的时候，视图位置的排布是非常重要的，另外，你还会学到如何制作视图在屏幕上移动和切换时所具备的动画效果。
- 第5章，视图的约束系统——Auto Layout 机制彻底改变了 iOS 程序里视图的排布方式。苹果公司的这种布局特性使开发者可以轻松地设计出更为协调一致的界面。此特性对于同一系列不同屏幕大小、不同界面、不同屏幕方向、不同语言的设备来说尤为重要。本章将会介绍如何用代码来做视图约束方面的开发。你会学到怎样在屏幕上的物件之间创建关系以及怎样指定布局规则，使 iOS 能够自动排布应用程序中的视图。看完本章后，你就能设定一套健全的屏幕布局规则了。
- 第6章，文本输入——本章的解决方案都与文本有关，这些解决方案能够解决许多问题。你会学到如何控制键盘、如何使屏幕上面的控件支持文本输入、如何扫描文本、如何格式化文本，等等。这一章会把与 iOS 程序文本处理有关的各项技术都涵盖在内，包括文本框、文本视图以及 iOS 内置的拼写检查器。
- 第7章，使用视图控制器——本章将会讲解各种视图控制器类的用法，这些类使得用户可以在更大的范围中与应用程序交互，而开发者也可以借此来排布视图。你将通过本章的各解决方案学到页面视图控制器、分栏视图控制器、导航控制器等视图控制器的用法。
- 第8章，常用的控制器——iOS SDK 里面有很多系统自带的控制器，开发者可以用它们来完成日常的开发任务。本章将介绍最为常用的控制器。你会学到如何从照片库中选取照片、如何拍照、如何录制并编辑视频。另外，你还会学到如何在程序中编写电子邮件及文本消息，以及如何在 Twitter 及 Facebook 等社交媒体上张贴信息。
- 第9章，创建并管理表格视图——表格（table）是一种可以滚动的交互类，它在屏幕较小的设备上面效果很好，在屏幕较大的平板电脑上面效果也很不错。由于表格可以把内容以一种简单而自然的方式组织起来，所以很多 iOS 应用程序都是以表格为中

心的。本章将介绍表格的用法，解释表格的工作原理，讲解可供开发者使用的各种表格，并且告诉你如何在程序中利用表格的各种特性。

- 第 10 章，集合视图——集合（collection）视图的许多概念都与表格相同，但是功能更加强大，而且更加灵活。本章将会讲述使用集合视图所需的各种基础知识，包括如何创建可以横向滚动的列表、如何创建网格布局、如何创建圆形等特殊方式的布局，等等。你将学到怎样通过布局规格（layout specification）把视觉效果集成到集合视图里面，以及怎样使集合视图中的内容在滚动之后自动调整位置，另外，你还会学到如何利用内置的动画支持来创建出最有效的互动效果。
- 第 11 章，分享文档与数据——在 iOS 系统中，应用程序可以分享信息和数据，另外，开发者也可以使用系统所提供的许多特性，把控制权从一个程序转移到另一个程序。你可以用本章所介绍的几种方式在应用程序之间分享文档及数据。你将学到如何把这些特性添加到自己的应用程序之中，以及如何灵巧地使用分享功能，令自己的应用程序可以和 iOS 生态系统里的其他程序协同运作。
- 第 12 章，浅谈 Core Data——Core Data 提供了一套受托管的数据存储机制，使开发者可以在应用程序中查询并更新存储区里的数据。它提供了一套基于 Cocoa Touch 的对象接口，使得 iOS 开发者能够像使用 SQL 查询语句那样，通过 Objective-C 代码来管理关系型数据库中的数据。本章介绍 Core Data。通过其中的各解决方案，你可以初步了解这项技术，同时还能以本章内容为出发点，继续深入学习 Core Data。你将学到如何设计受托管的数据库存储机制、如何添加和删除数据、如何用代码查询数据，以及如何把这些操作同 UIKit 中的表格视图及集合视图相集成。
- 第 13 章，网络编程基础——在连接到 Internet 的设备上面，特别适合用 iOS 程序来订阅基于网络的服务。苹果公司为 iOS 平台提供了各种坚实的网络计算服务及支持技术。本章将介绍网络编程中的常用技术，同时也提供一些解决方案，用来简化日常的网络开发任务。本章介绍 iOS 7 新引入的 HTTP 系统，并且提供实现数据下载功能（包括后台下载）所用的一些范例代码。你还会学到如何判断网络是否可用，以及如何使用 Web 服务，这其中包含了一些范例代码，它们告诉你如何通过 XML 解析及 JSON 序列化来访问一些在线服务。
- 第 14 章，针对特定设备的开发——每台 iOS 设备都有许多属性，有些属性是该设备所独有的，而有些则是许多设备所共有的；有些属性是持续变化的，而有些则保持不变。这些属性包括设备当前的物理方向、型号名称、电池状态以及是否可以访问机体内的硬件等。本章将会讲解如何查看设备的硬件规格，以及如何查看设备中可供使用的感应器。这一章所提供的解决方案可用来查询当前设备的各项信息。
- 第 15 章，辅助功能——本章简单地介绍 VoiceOver 这项辅助功能，开发者可以通过该功能尽量扩大应用程序的受众。你将学到如何为应用程序添加与辅助功能有关的标签及提示，以及如何在模拟器和 iOS 设备中测试这些特性。
- 附录 A，Objective-C 字面量——本附录介绍了 Objective-C 语言里与数字、数组及字典有关的一些新特性。

对范例代码的说明

为了大家学起来方便，本书的范例代码只使用一个 `main.m` 文件。编写 iPhone 或 Cocoa 应用程序时，开发者一般都不会这么做，而且说实在的，也确实不应该这么做，但是，这种做法却非常适合展示一个大的概念。假如一份范例代码分成 5 个、7 个或 9 个文件，就不太容易讲述这个概念了。而将所有代码都写到一个文件里，则有助于专门把这个概念说清楚。

书中的范例代码不应当当成独立的应用程序来用。每份范例代码只对应于一个解决方案，而且只演示一个概念。每个 `main.m` 文件都是专门为了实现某个中心概念而编写的。读者在学会这些思路之后，可以按照平常开发时所用的文件组织结构及布局方式将其转换为普通的应用程序结构。本书所用的代码组织方式与日常开发中所应提倡的标准组织方式并不相符。笔者之所以采用这种方式，是为了提供精确的解决方案，而大家可以根据需求把它们集成到自己的工作项目中。

苹果公司的标准范例代码与本书不同，你必须查看很多源文件，才能在脑中构建出一套与待演示的概念有关的“思维模型”。那些范例代码都是完整的应用程序，里面通常会涉及一些与当前所要解决的问题没有关系的任务。我们必须花很大精力才能找到与当前问题有关的代码，这是得不偿失的。

本书还有些范例代码没有遵循“一个文件只说一件事”的规则：如果某个解决方案与类的实现有关，那么本书还会提供标准的类文件及头文件。有些解决方案并不是为了强调某项技术，而是为了提供某些类及 `category`（`category` 是一种针对现有类所做的扩展，它不产生新的类）的实现。对于这些解决方案来说，读者可以找到单独的 `.m` 与 `.h` 文件，而 `main.m` 文件里面则封装了一份框架代码，用来描述其余的事情。

本书大多数范例代码都只使用一个应用程序标识符，也就是 `com.sadun.helloworld`。只使用一个标识符的好处是：你的 iOS 设备里不会装很多范例程序。每安装一个范例程序，都会把前面那个替换掉，这样的话，设备的主屏幕就能干净一些。如果需要同时安装多个范例程序，那么只需给标识符加个后缀就可以了，例如 `com.sadun.helloworld.table-edits`。如果想令多个应用程序所显示出来的名称各不相同，那么可以编辑自定义的显示名称。你的 Team Provisioning Profile 能够匹配包括 `com.sadun.helloworld` 在内的每一个应用程序标识符。这样的话，无须修改标识符，就可以把编译后的代码安装到设备上面了，只是记得要在每个项目的 Build Settings 中更新 Code Signing Identity。

获取范例代码

你可以在开源项目托管网站 GitHub 中找到本书源代码：<http://github.com/erica/iOS-7-Cookbook>。每一章的源代码都放在一个文件夹内，其中包含书里的相关范例材料。解决方案的编号与其在书中的编号相同。比方说，第 5 章的第 6 个解决方案放在 C05 文件夹下面的 06 子文件夹中。

以 00 为编号的项目或是编号带有后缀（例如 05b、02c）的项目是为了便于搜索或创建

插图而使用的素材。比方说，第 9 章的 00 – Cell Types 项目是为了创建图 9-2 中的效果而编写的，那张图用来演示系统所提供的各种表格视图单元格样式。一般情况下，笔者会把这些多余的项目删掉。本书初稿的读者请求笔者把它们放在这个版本中。整个代码库里大约能找到六七个这样的范例项目。

为本书出力

范例代码绝不是一成不变的，它会随着苹果公司的 SDK 与 Cocoa Touch 库而不断进化。请各位读者一起参与这个过程。你可以提交 bug 修复和修改书中的错误，也可以扩充现有的代码。你可以对 GitHub 代码库做分支 (fork)，自己调整代码，实现一些功能，然后再分享回主代码库里。如果你有新的想法或思路，请告诉我们。我们很乐意将你的宝贵建议加到代码库中，并据此完善本书的下一个版本。

获取 git 工具

你可以使用 git 版本控制系统来下载本书源代码。Xcode 5 集成开发环境提供了非常健壮的 git 支持。Xcode 5 工具箱里面也包含了命令行式的 git 工具。此外，还有大量的第三方及商业版 git 工具可供选择。

使用 GitHub

GitHub (<http://github.com/>) 是最大的 git 托管网站，有超过 15 万个公开的代码库 (repository)^①。它可以免费托管公开项目，也可以付费托管私有项目。该网站提供了一套可以自定义的 Web 界面，其中包含 Wiki 托管、问题追踪等功能，是项目开发者之间的一个优秀的社交网络，开发者可以在这里寻找新代码，也可以协同开发既有的程序库。你可以在 GitHub 网站注册免费账号，注册好之后，就可以复制并修改本书的范例代码库了，另外，也可以创建自己的开源 iOS 项目，并与他人分享。

① GitHub 的代码库里面还可以存放除代码之外的其他内容。——译者注

Acknowledgments 致谢

Erica Sadun

若没有 Chuck Toporek, 此书不可能面世, 多年以来, 是他在为我的书做编辑, 并且不停地敦促我写稿。他在多家图书发行商那里工作过, 目前就职于 Omnigroup, 我非常想念他。没有他的帮助, 就不会有这本书。Chuck Toporek 可以玩转两套绝活: 一是能够激发作者完成他们自认为无法做到的事情, 二是能够挥舞着“务实的大棒”^①, 逼迫作者把书稿内容写得集中而又实用一些^②。这世上最难的事情, 莫过于有人不停地催着你必须赶在截止日期之前把书写完, 而且还必须写得很好。

同样要感谢 Trina MacDonald (本书的优秀编辑)、Chris Zahn (卓有才华的策划编辑) 以及 Olivia Basegio (兢兢业业的助理编辑, 负责处理幕后事务)。此外, 非常感谢整个 Addison-Wesley/Pearson 制作团队, 尤其要谢谢 Kristy Hart、Betsy Gratner、Kitty Wilson、Anne Goebel、Lisa Stumpf、Gloria Schurick 与 Chuti Prasertsith。同时, 感谢 Safari 团队的人员制作本书样稿, 并及时修复技术问题。

感谢多年以来的经纪人——Studio B 的 Stacey Czarnowki, 感谢最近退休的 Neil Salkind; 感谢技术评审 Collin Ruffenach、Mike Shields 及 Ashley Ward, 你们 3 位令本书内容更加贴近现实, 从而免于空谈; 感谢 TUAW、Ars Technica 及 Digital Media/Inside iPhone 博客的诸位同仁以及原来的各位同事。

在整个 iOS 开发者社区中, 有很多人曾经帮助了我, 在此深表感谢, 他们是: Jon Bauer、Tim Burks、Matt Martel、Tim Isted、Joachim Bean、Aaron Basil、Roberto Gamboni、John Muchow、Scott Mikolaitis、Alex Schaefer、Nick Penree、James Cuff、Jay Freeman、

① 原文是 wielding the large “reality trout” of whacking, 字面意思为: 挥舞着超级大的“现实鲑鱼”。这是一种比喻的说法, 大意是告诫某人应该怎样做。——译者注

② 在本书的编辑和制作环节中, 我们没有伤害到任何一条所谓的“鲑鱼”, 无论是真实的鲑鱼还是意念中的鲑鱼。但是我们却喝掉了无数罐饮料: Erica 喝了很多 Diet Coke (健怡可乐), Rich 喝了很多 Diet Mountain Dew (激浪轻怡), 这些听装饮料无私地“奉献”了它们自己, 为两位笔者写作本书效劳。

Mark Montecalvo、August Joki、Max Weisel、Optimo、Kevin Brosius、Planetbeing、Pytey、Michael Brennan、Daniel Gard、Michael Jones、Roxfan、MuscleNerd、np101137、UnterPerro、Jonathan Watmough、Youssef Francis、Bryan Henry、William DeMuro、Jeremy Sinclair、Arshad Tayyeb、Jonathan Thompson、Dustin Voss、Daniel Peebles、ChronicProductions、Greg Hartstein、Emanuele Vulcano、Sean Heber、Josh Bleacher Snyder、Eric Chamberlain、Steven Troughton-Smith、Dustin Howett、Dick Applebaum、Kevin Ballard、Hamish Allan、Oliver Drobnik、Rod Strougo、Kevin McAllister、Jay Abbott、Tim Grant Davies、Maurice Sharp、Chris Samuels、Chris Greening、Jonathan Willing、Landon Fuller、Jeremy Tregunna、Wil Macaulay、Stefan Hafener、Scott Yelich、chrallielinder、John Varghese、Andrea Fanfani、J. Roman、jtbandes、Artissimo、Aaron Alexander、Christopher Campbell Jensen、Nico Ameghino、Jon Moody、Julián Romero、Scott Lawrence、Evan K. Stone、Kenny Chan Ching-King、Matthias Ringwald、Jeff Tentschert、Marco Fanciulli、Neil Taylor、Sjoerd van Geffen、Absentia、Nownot、Emerson Malca、Matt Brown、Chris Foresman、Aron Trimble、Paul Griffin、Paul Robichaux、Nicolas Haunold、Anatol Ulrich (hypnocode GmbH)、Kristian Glass、Remy “psy” Demarest、Yanik Magnan、ashikase、Shane Zatezalo、Tito Ciuro、Mahipal Raythatha、Carbon Five 的 Jonah Williams、Joshua Weinberg、biappi、Eric Mock。同时感谢 irc.saurik.com 及 irc.freenode.net 的 iPhone 开发者频道 (iPhone developer channel) 里的每位网友。此外还有很多要感谢的人，由于数量众多，恕无法一一列名。有了他们所提供的技术、建议以及反馈，本书才能够得以完成。前述名单若有遗漏，还请帮助过我的诸君见谅。

特别感谢我的家人及朋友，连月以来，本书发行了很多新的 beta 版本，我有时无故缺席聚会，也经常发泄情绪，而他们则一直耐心以对。衷心感谢他们一路伴我写完这本书。尤其要谢谢孩子们对我的理解，我有时没能当个好妈妈，而是在弓着背敲键盘，但他们却依然支持我。过去几个月来，孩子们帮了我的大忙，他们帮我测试程序并提出建议，这真的很棒！每天我都提醒自己：能和他们生活在一起，实在是种福气。

Rich Wardwell

临近交稿期限的时候，我有很多话要说，在此谨向 Erica 致以最深的敬意，感谢她邀请我参与本书新版的写作，我觉得非常荣幸。在她的督导之下，我学到了很多，至少可以说，我明白了她对书稿质量的严格要求。

感谢我们的编辑 Trina MacDonald，若没有她的坚持，我可能写完第 1 章之后就会抓狂无法坚持下去了。在写作本书的过程中，我时而沮丧、时而焦虑、时而逃避，但 Trina 一直在指导并鼓励我。也要感谢助理编辑 Olivia Basegio，她精心安排并管理了一个技术编辑团队。技术编辑们做了全方位的努力，大大提升了本书质量，如果没有他们，我自己是做不到这一点的，所以，我要对 Collin Ruffenach、Mike Shields 及 Ashley Ward 致以深深的谢

意。感谢本书制作团队及其成员 Betsy Gratner 和 Kitty Wilson，是他们提升了我的写作技能。Addison-Wesley/Pearson 团队中还有很多未曾说过话的朋友，也要感谢他们，感谢令本书得以面世的每个人。

特别感谢 Black Pixel 的 Bil Moorhead、George Dick 及 Daniel Pasco，在我专注写书而忽略了日常的工作时，你们给了我极大的理解。很荣幸能与 Black Pixel 的诸位同仁共事。

感谢我的父母 Rick 及 Janet，他们一直是我人生道路上最伟大的支持者，这次写书当然也是如此。感谢至亲 Steve 及 Cary 给我们提供了一个舒适的写作空间。

最后，感谢爱妻和两个孩子，他们是这个写作项目得以竣工的元勋。我没有陪着他们度过温馨甜蜜的家庭时光，而且也经常不和孩子们玩耍。有了他们的关爱，我才能把这本书写完。

目 录 Preface

译者序
前 言
致 谢

第 1 章 手势与触摸 1

1.1 触摸 1

1.1.1 触摸操作所处的阶段 2

1.1.2 UIResponder 类中的触摸 事件响应方法 3

1.1.3 对视图的触摸 4

1.1.4 多点触摸 4

1.1.5 手势识别器 5

1.2 解决方案：添加简单的直接操纵 界面 5

1.3 解决方案：添加拖动手势识别器 7

1.4 解决方案：同时使用多个手势 识别器 9

1.5 解决方案：限制移动 14

1.6 解决方案：测试触摸 15

1.7 解决方案：针对位图的触摸测试 17

1.8 解决方案：根据触摸情况在屏幕 上绘制内容 19

1.9 解决方案：令绘制效果变得平滑 21

1.10 解决方案：启用多点触摸 24

1.11 解决方案：检测圆圈手势 27

1.12 解决方案：创建自定义手势 识别器 32

1.13 解决方案：把滚动视图中的内容 拖曳到外面 34

1.14 解决方案：实时的触摸反馈 37

1.14.1 启用触摸反馈效果 38

1.14.2 拦截并转发触摸事件 38

1.14.3 实现 TOUCHkit 的 TOUCHkitView 类 40

1.15 解决方案：向视图添加菜单 42

1.16 小结 43

第 2 章 构建并使用控件 45

2.1 UIControl 类 45

2.1.1 目标 - 动作模式 46

2.1.2 控件的种类 46

2.1.3 控件事件 46

2.2 按钮 48

2.3 Interface Builder 中的按钮 50

2.4 解决方案：构建按钮 51

2.4.1 多行按钮文本 54

2.4.2 为按钮添加动画元件 54

2.4.3 为按钮添加额外状态 55

2.5 解决方案：使按钮以动画效果 来响应用户 55