

iOS PROGRAMMING 3RD EDITION

iOS编程 (第3版)

THE BIG NERD RANCH GUIDE

[美] Joe Conway & Aaron Hillegass 著

夏伟频 译 李骏 审校



华中科技大学出版社

<http://www.hustp.com>

iOS 编程(第 3 版)

[美] Joe Conway 著
Aaron Hillegass

夏伟频 译

李 骏 审校

华中科技大学出版社

中国·武汉

内 容 简 介

本书第3版更新了Xcode 4.3和iOS 5的内容,涵盖了开发iOS应用的方方面面。从Objective-C基础知识到新增加的语言特性;从AppKit库到常见的Cocoa设计模式;从Xcode技巧到Instruments等。如果读者刚接触iOS编程,阅读本书前两章可以快速入门。如果读者已经有iOS编程经验,阅读本书可以拾遗补缺。本书的另一个特色是教读者以正确的方法解决问题。Objective-C的习惯约定有哪些(例如命名约定、内存管理约定)、创建子类时如何处理初始化方法、Cocoa的常见设计模式有哪些、如何选择数据保存方法等。这些问题,如果读者自己摸索,难免费时费力。本书已经将这些知识整理、归纳并清楚地呈现在读者面前。

Original English language edition copyright © 2012 by Joe Conway and Aaron Hillegass The Chinese Translation edition Copyright © 2013 by HUAZHONG UNIVERSITY OF SCIENCE AND TECHNOLOGY PRESS in arrangement with BIG NERD RANCH, INC.

湖北省版权局著作权合同登记 图字:17-2013-001号

图书在版编目(CIP)数据

iOS编程(第3版)/(美)Joe Conway, Aaron Hillegass著;夏伟频译;李骏审校. —武汉:华中科技大学出版社,2013.3

ISBN 978-7-5609-8605-0

I. i… II. ①J… ②A… ③夏… ④李… III. 移动终端-应用程序-程序设计 IV. TN929.53

中国版本图书馆CIP数据核字(2012)第308864号

iOS编程(第3版)

[美]Joe Conway, Aaron Hillegass 著
夏伟频译 李骏审校

策划编辑:徐定翔

责任编辑:陈元玉

出版发行:华中科技大学出版社(中国·武汉)

武昌喻家山 邮编:430074 电话:(027)87557437

录 排:华中科技大学惠友文印中心

印 刷:湖北新华印务有限公司

开 本:787mm×960mm 1/16

印 张:38

字 数:976千字

版 次:2013年3月第3版第1次印刷

定 价:99.00元

责任校对:朱霞

责任监印:周治超



华中出版

本书若有印装质量问题,请向出版社营销中心调换
全国免费服务热线:400-6679-118 竭诚为您服务
版权所有 侵权必究

翻译说明

本书的术语、词汇翻译优先使用 Apple 公司的正式译法，我们参考了 Apple 公司中国官方网站、iOS 设备中文版使用手册、中文开发文档、中文软件系统界面的译法。有些翻译无固定译法可循，特作如下说明。

- 在 Objective-C 中，“发送消息”和“调用方法”的概念虽略有差别，但是可以认为实际效果相同。例如，“向某个对象发送 `release` 消息”和“调用某个对象的 `release` 方法”意思相同。
- 原书不区分“实例”（instance）和“对象”（object），中文版根据原文翻译。两者意思相同。
- 本书将 iOS 4.0 加入的语言特性 Blocks 翻译为“Block 对象”，因为 Blocks 是简称，其正式的名称为 Block Object。
- Override 有多种译法，本书译为“覆盖”。

我们列出了详细的中英文术语对照表，并尽可能提供译法的出处。欢迎访问：
<http://www.iosprogrammingbook.com>。

致谢

Acknowledgments

虽然封面只列出了作者的名字，但是本书的面世离不开幕后很多人的帮助。作者借此表达真诚的谢意。

- iOS 新手培训课程的其他教员不断地提供建议和修改意见。他们是 Scott Ritchie、Brian Hardy、Mikey Ward、Christian Keur、Alex Silverman、Owen Matthews、Brian Turner、Juan Pablo Claude 和 Bolot Kerimbaev。
 - 本书编辑 Susan Loper 将原本晦涩难懂的文字润色为通顺流畅的好文。
 - 几位技术审校者帮助我们找到并修正了多处瑕疵。他们是 Bill Monk 和 Jawwad Ahmad。
 - Ellie Volckhausen 设计了本书封面（照片拍摄的是自行车架的一个底部支撑脚）。
 - 来自 IntelligentEnglish.com 的 Chris Loper 设计并制作了本书的 EPUB 版本和 Kindle 版本。
 - 来自 Pearson Technology Group 的团队耐心地指导我们走完整个出版流程。
- 最后还要感谢我们的学生，他们激发我们编写了本书。

前言

Introduction

要成为一名优秀的 iOS 程序员，需要攻克以下三项难题。

- 必须学会 Objective-C 语言。Objective-C 是 C 语言的扩展，小巧简单。读完本书的前 4 章，读者就能掌握 Objective-C 的基础知识。
- 必须掌握 Cocoa 的常用技术。其中包括内存管理、委托机制（delegation）、固化机制（archiving），以及如何正确使用视图控制器（view controller）。理解这些技术需要花些时间。本书前半部分会介绍这些内容。
- 必须掌握框架（framework）。读者的最终目标是了解 iOS 的所有框架，学会如何使用框架中的每一个类和方法。但这几乎是不可能完成的任务：iOS 有 3 000 多个方法，200 多个类。而且随着 iOS 的版本升级，Apple 还会不断地加入新的类和新方法。本书将会介绍 iOS SDK 中的各个组成部分，但是不会太过深入。作者的目標是带领读者入门，使读者能够自行阅读并理解 Apple 的参考文档。

Big Nerd Ranch 公司采用本书作为“iOS 新手培训课程”的教材。这些内容经过了长期的检验，并且帮助很多人成为 iOS 程序员。真心希望本书也能给你带来帮助。

本书适合哪些读者

本书假设读者已经跃跃欲试，准备开发 iOS 应用，所以不会花费笔墨去证明 iPhone、iPad 和 iPod touch 是很棒的产品。本书假设读者了解 C 语言并对“面向对象编程”有一定的了解。如果不是，建议读者先阅读 Big Nerd Ranch 出版的《Objective-C 编程》。

第 3 版有哪些更新

第 3 版使用的是 Xcode 4.3，并且所有的应用都运行于安装有 iOS 5 的设备或模拟器。

从 iOS 5 开始，ARC（automatic reference counting）成为 iOS 系统的默认内存管理机制。作者重写了介绍内存管理一章以涵盖 ARC。此外，本书的所有例子都会使用 ARC。

第 3 版加入了新的章节，分别介绍 iOS 5 新加入的特性 gesture recognizer、storyboard、**NSRegularExpression** 和 iCloud。此外，还新增了两章专门介绍 Model-View-Controller-Store 设计模式。Model-View-Controller-Store 是 Big Nerd Ranch 公司的程序员在开发 iOS 应用时经

常使用的设计模式，作者相信该设计模式适用于绝大多数的应用。

除了以上这些明显的变化，作者还根据读者和学生提出的问题，对本书做了大量修订。可以说与前一版本相比，新版页更有改进。

教学理念

本书将向读者传授 iOS 编程的基本概念。在阅读的同时，读者还要输入大量的代码，并创建一组应用。这样，完成本书的学习后，读者得到的将不仅是知识，还有经验。相信你我多少都有过被灌输知识的痛苦经历，所以本书转而使用“边做边学”的教学方法——概念与代码并重。

多年的 iOS 编程教学工作让作者了解到：

- 某些概念是学习 iOS 编程必须知道的。本书会集中介绍这部分内容。
- 学习能立即派上用处的概念，效果最佳。
- 知识与经验并重时，学习效果最佳。
- 实际操作很重要。本书会要求读者先输入代码，再理解含义。读者可能会觉得这种不明就里的模仿意义不大，但是“找出错误并修正代码”是学习编程的好方法。这种最基础的调试过程不仅不是累赘，反而能帮助读者彻底理解代码。这也是作者鼓励读者自己输入代码的原因。虽然可以直接下载例子代码，但是拷贝粘贴不是编程。本书对读者及读者的编程技能有更高的期望。

这种模仿对读者意味着什么？意味着读者要信任本书作者，而且要有耐心。本书会尽可能地将问题讲透，但读者有时只能相信作者的意见（如果读者对此有异议，请往下看——作者列出了若干解决方案，也许能有帮助）。遇到暂时不能理解的概念时不要气馁，因为本书不会将涉及某个概念的所有知识一次全部介绍完，这是有意为之。如果某个概念没有解释清楚，那么很可能在需要时再提供更详细的介绍。有些初看无法理解的概念，可能会在读者第一次（或第十二次）实际应用时突然变得清晰易懂。

每个人的学习方法不同。读者可能会喜欢本书这种“按需分步介绍概念”的方法，也可能不会不喜欢。如果是后者，这里提供若干建议。

- 先不要着急，等待在后续的章节中将问题讲透。
- 查阅索引，先阅读相关概念的详细介绍。
- 查阅 Apple 提供的在线文档。这些文档是非常重要的开发工具，需要多多练习使用。读者应该尽早地，也应尽可能地使用在线文档。
- 如果在学习 Objective-C 或面向对象的编程概念时遇到困难（或是预感会有困难），那么建议读者先阅读 Big Nerd Ranch 的《Objective-C 编程》（华中科技大学出版社出版）。

如何使用本书

本书内容基于 Big Nerd Ranch 的培训课程，所以有其特定的阅读方法。

读者可以先设定一个合理的目标，例如“每天阅读一章”。然后在阅读时为自己找一个安静的场所，至少一个小时内不会被打断。关掉 Email、Twitter 客户端和聊天工具——读书无法多任务并行，必须集中精力。

在阅读的同时，读者还需要编写代码。根据读者的喜好，可以先通读整章。但是真正的学习从编写代码开始。要真正理解某个概念，需要编写程序并进行实际操作，尤其是调试 (debug) 程序的过程，会特别有帮助。

书中的部分项目需要使用额外的文件，例如第 1 章的 Quiz 应用需要一个图标文件。本书已经为读者准备好了这些文件，通过以下网址可以下载本书的项目代码和资源文件：
<http://www.bignerdranch.com/solutions/iOSProgramming.zip>。

学习分两类。学习历史时，要做的只是在已经理解的知识架构上添加更多细节。我们将这类学习称为简单学习 (easy learning)。学习历史的确需要花费很长的时间，但是难度不大。学习 iOS 编程则是困难学习 (hard learning)，会经常卡壳，尤其是在刚开始的时候。作者编写本书的目的是帮助读者越过难度陡增的学习曲线。下面提供两则建议，希望帮助读者轻松越过障碍。

- 找位懂 iOS 开发，并且愿意回答读者提问的程序员。第一次尝试将应用安装至 iOS 设备时，如果缺少有经验的程序员的协助，就可能会遇到困难。
- 保证足够的睡眠。缺少睡眠将无法记住所学的知识。

本书是如何组织的

本书各章都会先介绍一个或多个 iOS 开发概念，然后给出具体的示例代码。每章末尾有练习，为读者提供编写代码的机会。建议读者至少完成部分的练习，以巩固所学，同时也提升自信。此外，多数章节最后会有一到两个“深入学习”部分，对之前介绍的内容做一些补充。

第 1 章介绍如何创建并安装一个很小的应用，以此带领读者入门 iOS 开发。读者将开始学习如何使用 Xcode 和 iOS 模拟器，以及创建项目和文件所需的全部步骤。第 1 章还介绍了 Model-View-Controller 及其在 iOS 开发中的作用。

第 2、3 章简单介绍 Objective-C 和内存管理。这两章没有创建 iOS 应用，而是构建并调试了一个名为 RandomPossessions 的工具，以帮助读者理解这些概念。

第 4、5 章介绍 Core Location 框架和 MapKit 框架，并创建一个名为 Whereami 的地图应用。读者将获得大量与委托机制有关的经验（委托机制是一种重要的设计模式），并且有机会使用协议 (protocol)、框架 (framework)、对象图、调试器和 Apple 开发文档。

第 6、7 章通过 Hypnosister 应用和 HypnoTime 应用集中介绍 iOS 用户界面。读者将有机会练习如何使用视图和视图控制对象，实现滚动和缩放，并通过 UITabBar 对象切换屏幕。

第 8 章介绍如何创建一个规模稍小的应用 **HeavyRotation**，同时会介绍通告机制（notifications），以及如何实现自动转屏（autorotation）。此外，还会介绍如何通过自动缩放（autosizing）让 **HeavyRotation** 能够自动适应 iPad 的屏幕尺寸。

第 9 章介绍本书最庞大的应用 **Homepwner**（顺便提一句，**Homepwner** 并没有拼错。读者可以通过 www.urbandictionary.com 找到 **pwn** 的定义）。**Homepwner** 的功能是保留一份财产清单，以供发生灾难后核对。本书将花 9 章的篇幅来完成 **Homepwner**。

第 9、10 和 15 章介绍如何使用并显示表格。读者将学到 **UITableView**、**UITableViewController** 和数据源机制；学会如何在表格视图中显示数据、如何让用户编辑表格及如何改善相关的界面。

第 11 章以第 7 章的导航机制为基础，介绍 **UINavigationController** 的用法，并为 **Homepwner** 增加一个垂直（drill-down）界面和一个导航条。

第 12 章介绍如何通过相机拍照，以及如何为 **Homepwner** 实现显示图片和保存图片的功能。读者将有机会使用 **NSDictionary** 和 **UIImagePickerController**。

第 13 章介绍模态视图控制对象及 iPad 特有的 **UIPopoverController**。此外，还会将 **Homepwner** 改为通用应用（universal application），能在 iPhone 和 iPad 上全屏运行。

第 14 章介绍多种保存/读取数据的机制。值得注意的是，**Homepwner** 应用会通过 **NSCoding** 协议固化（archive）数据。本章还介绍了切换应用状态（例如激活状态、后台运行状态和悬停状态）的各个过程。

第 16 章介绍 Core Data。**Homepwner** 应用将改用 **NSManagedObjectContext** 对象来保存/读取数据。

第 17 章介绍与国际化、本地化有关的概念和技术。通过 **Homepwner** 的本地化过程，读者将学会如何使用 **NSLocale**、字符串对照表（strings table）和 **NSBundle**。完成 **Homepwner** 应用。

第 18 章将介绍如何通过 **NSUserDefaults** 来永久保存用户预置（user preferences）。

第 19、20 章将创建一个名为 **TouchTracker** 的绘图应用，以介绍触摸事件。读者将学到如何为应用增加多点触摸功能，以及如何使用 **UIGestureRecognizer** 来响应特定的手势。此外，读者还将了解第一响应对象（first responder）和响应对象链（responder chain），并针对 **NSDictionary** 做更多练习。

第 21 章将向读者介绍如何使用 **Instruments**，并借此优化应用的性能。此外，第 21 章还会介绍 **Xcode** 的方案（scheme）和静态分析器（static analyzer）。

第 22、23 章将介绍层（layer）和 **Core Animation** 框架，并为 **HypnoTime** 应用实现动画效果。读者将学到隐式动画（implicit animation）和动画对象，例如 **CABasicAnimation** 和 **CAKeyframeAnimation**。

第 24 章将介绍 **Storyboard**，一项能够帮助读者构建应用的 iOS 新特性。读者将学会如何

用 **UIStoryboard** 来构建应用，以及用 Storyboard 创建程序的优缺点。

第 25 章将通过创建一个名为 Nerdfeed 应用，介绍如何实现 Web 服务通信。Nerdfeed 会通过 **NSURLConnection** 和 **NSXMLParser**，从指定的服务器获取并解析 RSS 文件。此外，Nerdfeed 还会用 **UIWebView** 对象来显示相应的 Web 页面。

第 26 章将介绍 **UISplitViewController**，并为 Nerdfeed 添加一个 **UISplitViewController** 对象，以适应 iPad 更大的屏幕尺寸。

第 27 章将介绍 Block 对象 (blocks)，Block 对象正逐渐成为 iOS SDK 的一项重要特性。第 27 章还会创建一个简单的应用，为第 28 章在 Nerdfeed 中使用 Block 对象做好准备。

第 28、29 章将修改 Nerdfeed 的架构，以使用 Model-View-Controller-Store 设计模式。读者将学到如何处理请求逻辑 (request logic)，以及如何设计应用，方便存取外部数据。

第 30 章将介绍如何为应用添加 iCloud 功能，用来同步和备份用户多个设备上的数据。

代码风格

本书包含大量代码，作者希望这些代码及其背后的设计思路能具备参考价值。作者在编写这些代码时，已经尽可能地符合 Cocoa 编程习惯。尽管如此，其中的某些部分还是会和 Apple 的示例代码，或者其他书籍中的代码有一定的差别。虽然读者目前可能还无法理解这些差别的具体含义，但是在正文开始前，还是有必要将这些问题先列出来。

- 存取方法 (accessor method) 有一种称为 dot-notation 的调用语法。本书会解释 dot-notation，但不会使用它。对初学者而言，dot-notation 容易产生混淆。
- 创建 **UIViewController** 子类时，本书会将指定初始化方法 (designated initializer) 统一改为 **init**。这样，在创建实例时就不用知道 **UIViewController** 子类是否使用了 XIB 文件，更不用知道相关的文件名。
- 本书始终采用编写代码的方式来创建视图控制对象。虽然可以通过 Xcode 直接在 XIB 文件中创建 **UIViewController** 实例，但这样做会导致项目难以理解，并增加调试的难度。
- 本书的绝大部分项目都创建自最简单的 Xcode 模板：empty application (空白应用)。这里没有采用其他模板，是因为这些模板所生成的代码并不符合本书的代码风格。

作者相信，上述规则能让本书的代码更容易读懂，也更容易维护。当读者（在以上规则的限制下）完成本书的学习后，可以尝试打破这些规则，以检验这些规则是否真的有用。

版式说明

为了方便读者阅读，本书会对某些特定的内容使用专门的字体。其中，类名、方法名和函数名会以粗体、等宽的字体显示。类名的首字母会用大写，方法名的首字母会用小写。正文中

出现的类名、方法名和函数名将采用加粗的 Bitstream Vera Sar 字体表示，例如，“在 **RexViewController** 类的 **loadView** 方法里，使用 **NSLog** 函数输出结果。”

正文中出现的变量、常量和类型将采用 Bitstream Vera Sar 字体表示，但是不加粗。例如，“将变量 **fido** 定义为 **float** 类型，并赋予初始值 **M_PI**。”

应用程序和系统菜单选项将采用 Helvetica 字体表示，例如，“打开 **Xcode**，从 **File** 菜单选择 **New Project...**。”

书中的所有示例代码都将采用 Bitstream Vera Sar 字体表示。需要读者输入的代码会加粗显示。例如下面这段代码，除了第一行和最后两行外（这三行是已经存在的代码，列出来是为了让读者知道应该在哪里加入新的代码），其余部分都需要读者自行输入。

```
@interface QuizAppDelegate : NSObject <UIApplicationDelegate> {
    int currentQuestionIndex;

    // model 对象
    NSMutableArray *questions;
    NSMutableArray *answers;

    // view 对象
    IBOutlet UILabel *questionField;
    IBOutlet UILabel *answerField;
    UIWindow *window;
}
```

开发所需的硬件与软件

开发 iOS 应用要使用（也只能使用）基于 Intel 芯片的 Mac 计算机。读者需要下载由 Apple 提供的 iOS SDK，其中包括 Xcode（Apple 的集成开发环境）、iOS 模拟器，以及其他开发工具。

读者还需要注册加入 Apple 的 iOS 开发者计划，费用为每年 99 美元。原因有以下三个。

- 注册后可以免费下载最新的开发工具。
- 任何应用必需先经过数字签名，然后才能在设备上运行。因为只有注册会员才能签名应用，所以如果读者需要在设备上测试应用，就需要注册。
- 只有会员才能将应用提交至 App Store。

如果读者打算花时间读完本书的全部内容，那么注册加入 iOS 开发者计划是值得的。注册网站：<http://developer.apple.com/programs/ios/>。

进行 iOS 开发需要准备哪些设备？本书前半部分所涉及的多数应用都是针对 iPhone 的，但是也能在 iPad 上运行。在 iPad 上运行 iPhone 应用时，系统只会以 iPhone 的屏幕尺寸显示。这样虽然无法充分利用 iPad 的大尺寸屏幕，但是在没有 iPhone 的情况下，也是可以的。本书的前几章会集中介绍 iOS SDK 的基础部分，这些内容和设备无关。稍后的章节会介绍若干只针对 iPad 的内容，以及如何让应用能够在所有的 iOS 设备上全屏运行。

读者是否已经准备好了呢？下面开始正文。

目录

Table of Contents

前言	xiii
本书适合哪些读者	xiii
第 3 版有哪些更新	xiii
教学理念	xiv
如何使用本书	xv
本书是如何组织的	xv
代码风格	xvii
版式说明	xvii
开发所需的硬件与软件	xviii
第 1 章 第一个简单的 iOS 应用	1
1.1 创建 Xcode 项目	2
1.2 创建界面	5
1.3 模型-视图-控制器	9
1.4 声明	11
1.5 创建关联	13
1.6 实现方法	17
1.7 针对模拟器构建并运行应用	20
1.8 安装应用	22
1.9 应用图标	23
1.10 启动图片	25
第 2 章 Objective-C	29
2.1 对象	29
2.2 使用实例	30
2.3 编写命令行工具 RandomPossessions	33
2.4 创建 Objective-C 类的子类	39
2.5 异常与未知选择器	56
2.6 快速枚举	58
2.7 练习	58
2.8 初级练习：查找问题	59
2.9 中级练习：另一个初始化方法	59

2.10 高级练习：另一个类	59
2.11 关于深入学习部分	59
2.12 深入学习：如何为类命名	60
第 3 章 通过 ARC 管理内存	61
3.1 堆	61
3.2 栈	62
3.3 指针变量与对象所有权	63
3.4 内存管理	64
3.5 强引用与弱引用	67
3.6 属性	72
3.7 copy 特性	77
3.8 用点号调用存取方法	78
3.9 深入学习：Autorelease 池与 ARC 历史	78
第 4 章 委托与 Core Location	81
4.1 项目、目标与框架	82
4.2 Core Location	83
4.3 委托	87
4.4 使用调试器	91
4.5 初级练习：距离过滤	97
4.6 中级练习：获取方位信息	98
4.7 深入学习：构建阶段、编译器错误和连接器错误	98
第 5 章 MapKit 与文本输入	103
5.1 对象图	103
5.2 MapKit 框架	104
5.3 界面属性	105
5.4 设置 MapView 对象的委托对象	108
5.5 初级练习：地图类型	120
5.6 中级练习：修改地图类型	120
5.7 高级练习：扩充标注	120
第 6 章 UIView 子类与 UIScrollView	121
6.1 视图与视图层次结构	122
6.2 创建自定义视图	124
6.3 UIView 的 drawRect: 方法	128
6.4 Core Graphics	131
6.5 UIKit 中的绘图扩展	132
6.6 重画视图	134

6.7 运动事件	136
6.8 使用 UIScrollView	138
6.9 隐藏状态条	142
6.10 初级练习：颜色	143
6.11 中级练习：图形	144
6.12 高级练习：另一个视图与曲线	144
第 7 章 视图控制对象	145
7.1 UIViewController	145
7.2 UITabBarController	155
7.3 视图控制对象的生命周期	159
7.4 UIViewController 子类与模板	168
7.5 初级练习：增加一个标签项	168
7.6 中级练习：控制逻辑	168
7.7 深入学习：main 函数与 UIApplication	169
7.8 深入学习：Retina 显示屏	169
第 8 章 通告机制与自动转屏	173
8.1 通告中心	173
8.2 UIDevice 通告	175
8.3 自动转屏	176
8.4 强制使用横排模式	184
8.5 初级练习：设备近身通告	185
8.6 中级练习：编写代码设置自动缩放掩码	185
8.7 高级练习：覆盖自动转屏	185
8.8 深入学习：覆盖自动转屏	185
第 9 章 UITableView 与 UITableViewController	187
9.1 编写 Homepwner 应用	187
9.2 UITableViewController	189
9.3 UITableView 数据源	192
9.4 UITableViewCell 对象	198
9.5 代码片段库	203
9.6 初级练习：表格段	205
9.7 中级练习：固定行	205
9.8 高级练习：修改 UITableView 对象的外观	206
第 10 章 编辑 UITableView	207
10.1 编辑模式	207
10.2 增加行	213

10.3	删除行	215
10.4	移动行	216
10.5	初级练习：更改“删除”按钮的标题	218
10.6	中级练习：禁止移动某个表格行	218
10.7	高级练习：彻底禁止移动某个表格行	218
第 11 章	UINavigationController	219
11.1	UINavigationController 对象	219
11.2	额外的视图控制对象	223
11.3	UINavigationController 的导航功能	228
11.4	UINavigationController	233
11.5	初级练习：显示数字键盘	238
11.6	中级练习：关闭数字键盘	238
11.7	高级练习：压入更多视图控制对象	238
第 12 章	相机	239
12.1	通过 UIImageView 对象显示图片	239
12.2	初级练习：编辑图片	257
12.3	中级练习：删除图片	257
12.4	高级练习：Camera Overlay	257
12.5	深入学习：摄像	257
第 13 章	UIPopoverController 与模态视图控制对象	261
13.1	将 Homepwner 改为通用应用	261
13.2	UIPopoverController	263
13.3	更多的模态视图控制对象	266
13.4	初级练习：将 Whereami 改为通用应用	274
13.5	中级练习：使用不同的动画效果	275
13.6	高级练习：UIPopoverController 对象的外观	275
13.7	深入学习：视图控制对象之间的关系	275
第 14 章	保存、读取与应用状态	279
14.1	固化	279
14.2	应用沙盒	281
14.3	NSKeyedArchiver 与 NSKeyedUnarchiver	283
14.4	应用状态与状态切换	286
14.5	通过 NSData 将数据写入文件	289
14.6	再谈内存过低警告	291
14.7	模型-视图-控制器-存储设计模式	292
14.8	初级练习：PNG	293

14.9 中级练习：为 Whereami 增加固化功能.....	293
14.10 深入学习：应用的状态切换.....	293
14.11 深入学习：文件系统的读取和写入.....	295
14.12 深入学习：应用程序包.....	297
第 15 章 创建 UITableViewCell 子类.....	299
15.1 创建 HomeownerItemCell.....	299
15.2 处理图片.....	305
15.3 由 UITableViewCell 对象转发动作消息.....	308
15.4 初级练习：设置颜色.....	316
15.5 中级练习：UITableViewCell 子类的基类.....	316
15.6 高级练习：缩放.....	316
第 16 章 Core Data.....	317
16.1 对象-关系映射.....	317
16.2 用 Core Data 重写 BNRIterStore 的数据保存功能.....	318
16.3 再谈 SQL.....	336
16.4 Faults.....	337
16.5 各种存取机制的优缺点.....	339
16.6 初级练习：AssetTypePicker 对象的 iPad 界面.....	339
16.7 中级练习：增加 BNRAssetType 对象.....	339
16.8 高级练习：显示某种类型的 BNRIter 实例.....	340
第 17 章 本地化.....	341
17.1 通过 NSLocale 实施国际化.....	342
17.2 资源的本地化.....	343
17.3 NSLocalizedString 与字符串对照表.....	347
17.4 初级练习：增加一套本地化资源.....	349
17.5 深入学习：NSBundle 在国际化过程中的作用.....	349
第 18 章 UserDefaults.....	351
18.1 更新 Whereami.....	351
18.2 使用 UserDefaults.....	353
18.3 中级练习：初始坐标.....	355
18.4 高级练习：简化坐标的存储.....	356
18.5 深入学习：“设置”.....	356
第 19 章 触摸事件与 UIResponder.....	357
19.1 触摸事件.....	357
19.2 创建 TouchTracker 应用.....	359
19.3 实现 TouchDrawView，完成绘图功能.....	360

19.4	处理触摸事件并创建线条对象	361
19.5	响应对象链	364
19.6	初级练习：保存与读取	365
19.7	中级练习：颜色	365
19.8	高级练习：圆圈	365
19.9	深入学习：UIControl	365
第 20 章	UIGestureRecognizer 与 UIMenuController	367
20.1	UIGestureRecognizer 子类	367
20.2	用 UITapGestureRecognizer 对象识别“按下”手势	368
20.3	UIMenuController	371
20.4	UILongPressGestureRecognizer	372
20.5	UIPanGestureRecognizer 以及同时识别多个手势	374
20.6	深入学习：UIMenuController 与 UIResponderStandardEditActions	376
20.7	深入学习：再谈 UIGestureRecognizer	377
20.8	初级练习：清空线条	378
20.9	中级练习：修正错误	379
20.10	高级练习：速度与宽度	379
20.11	高级练习：颜色	379
第 21 章	Instruments	380
21.1	静态分析器	380
21.2	Instruments	382
21.3	Xcode 方案	392
21.4	创建新方案	394
21.5	构建设置	396
第 22 章	Core Animation Layer	399
22.1	层与视图的关系	399
22.2	创建 CALayer 对象	400
22.3	层的显示内容	403
22.4	隐式可动画属性	405
22.5	初级练习：增加一个 CALayer 对象	407
22.6	中级练习：圆角	407
22.7	高级练习：阴影	407
22.8	深入学习：通过代码生成显示内容	407
22.9	深入学习：层、位图与上下文	408
第 23 章	通过 CAAAnimation 控制动画	411
23.1	动画对象	411