

PEARSON

- 示例丰富实用、代码可重用
- iOS 高级开发必备秘笈
- 全面兼容 iOS 8

iOS 组件与框架

——iOS SDK 高级特性剖析

iOS Components and Frameworks

Understanding the Advanced Features
of the iOS SDK

[美] Kyle Richter Joe Keeley 著
袁国忠 译

人民邮电出版社
POSTS & TELECOM PRESS



PEARSON

iOS 组件与框架

—iOS SDK 高级特性剖析

iOS Components and Frameworks

Understanding the Advanced Features
of the iOS SDK

[美] Kyle Richter Joe Keeley 著
袁国忠 译



人民邮电出版社
北京

图书在版编目 (C I P) 数据

iOS组件与框架 : iOS SDK高级特性剖析 / (美) 里克特 (Richter, K.), (美) 基利 (Keeley, J.) 著 ; 袁国忠译. -- 北京 : 人民邮电出版社, 2014. 9
ISBN 978-7-115-36553-8

I. ①i… II. ①里… ②基… ③袁… 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167886号

版 权 声 明

Kyle Richter, Joe Keeley: iOS Components and Frameworks: Understanding the Advanced Features of the iOS SDK

Copyright © 2014 Pearson Education, Inc.

ISBN: 0321856716

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise without the prior consent of Addison Wesley.

版权所有。未经出版者书面许可, 对本书任何部分不得以任何方式或任何手段复制和传播。

本书中文简体字版由人民邮电出版社经 Pearson Education, Inc. 授权出版。版权所有, 侵权必究。

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

-
- ◆ 著 [美] Kyle Richter Joe Keeley
译 袁国忠
责任编辑 傅道坤
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: 29.75
字数: 622 千字
印数: 1 - 2 500 册
- 2014 年 9 月第 1 版
2014 年 9 月北京第 1 次印刷
- 著作权合同登记号 图字: 01-2014-4925 号

定价: 89.00 元

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

反盗版热线: (010) 81055315

内容提要

本书讲解了如何使用功能强大的 iOS 组件和框架来开发杰出的应用，使之具备优异的性能、可靠性、交互性，并对媒体提供支持。本书清晰地介绍了一些开发应用所用到的复杂中高级技术，并通过大量实例项目来演示将这些技术集成到 iOS 应用中的方法。

本书分为 25 章，其内容涵盖了使用 UIKit Dynamics 提供基于物理现象的动画效果；充分利用 Core Location、MapKit 和地理围栏；利用排行榜和成就等 Game Center 功能；让用户能够在应用中访问地址簿和媒体库；使用轻量级 JSON 在服务器、应用和网站之间传输数据；使用 UIDocument 和键/值存储同步通过 iCloud 同步应用；使用钥匙串保护数据；使用通知通告用户与应用相关的重要事件；使用 Core Data 在本地存储和检索持久化数据；使用 Objective-C 高级特性编写更简洁、更易于管理的应用；使用 GCD 提供的并发功能提高应用的响应速度；使用 TextKit 进行发杂的文本处理和显示；使用 Xcode 5 和 Instruments 卓有成效地调试应用；使用 PassKit 创建凭证等内容。

本书为想要开发高性能 iOS 应用的人员提供了实用的技巧、可重用的代码以及专家级的建议。本书适合具有一定 iOS 开发经验，要想进一步提升其开发技能的从业人员阅读。

序

2008年，iOS SDK的前身iPhone SDK Beta版面世，当时我主要致力于编写Mac桌面应用，没有过多地考虑移动应用开发，但从那时起我一直在使用iOS SDK。

既然选择了做早期使用者，就得自力更生。那时Apple提供的文档很少，而要使用SDK，就必须签署保密协议（NDA，它俨然成了解码魔戒），这使得早期使用者更只能靠自己：通过Google搜索不到相关内容；无法前往StackOverflow寻求帮助；根本就没有介绍SDK的图书。

Apple推出iPhone 6年（确实只有6年）后的今天，情况有了很大改观：iPhone SDK变成了iOS SDK；探讨iOS开发的图书、博客、播客和会议随处可见。然而，从2009起，WWDC入场券始终一票难求，开发人员要想了解即将推出的新功能更难了——无论对新手还是老手来说都如此。对iOS开发人员来说，需要学的东西实在太多。

身为iOS开发人员，及时地掌握所有组件和框架是我面临的巨大挑战之一。iOS人机界面设计指南（Human Interface Guideline, HIG）原本能够提供这方面的帮助，但其深度和广度都有限。当前，通过Google搜索或浏览StackOverflow，确实能够找到一些答案，但这些答案通常只告诉您如何做，很少说出个所以然，更别说您亟需知道的细节了。

在本书中，Kyle和Joe填补了这种空白，他们提供了丰富的细节，可帮助您全面认识重要的iOS SDK组件。

我有幸认识Kyle和Joe多年。他们是我见过的最聪明的开发人员，多年来开发了不少出类拔萃的应用，还一如既往地跟iOS开发社区分享其掌握的知识：在会议上发表演讲、编写iOS开发著作。如果您遇到iOS开发方面的问题，Kyle和Joe很可能提供答案。

然而，Kyle和Joe之如此杰出，凭借的不仅仅是百科全书式的iOS知识，还有乐意与任何人分享知识的品质。在Kyle和Joe的字典里，没有竞争对手，只有朋友。

本书充分展露了Kyle和Joe渊博的iOS SDK知识，这也是我喜欢本书的原因之一。本书详细介绍了涉及的每个组件，其详尽程度之高绝非网上资源可比。

我也喜欢本书的组织结构。本书并不适合从头到尾地阅读；您只需在遇到搞不定的问题（如实现集合视图或在后台线程中执行任务）时求助于它即可。您在需要时拿起它，找到解决方案并编写实现代码后可将其束之高阁，直到再次需要它。对任何iOS开发人员来说，无论其水平高低，本书都是必备的参考资料。您可能认为自己掌握了Core Location和MapKit，但我敢肯定您将通

过本书学到以前根本不懂的知识。

Kyle 和 Joe 不自负、不自夸，从不表现得比任何开发人员都优秀。他们身上浸透着 Mac 和 iOS 开发人员社区的精神，这种精神让 Mac 和 iOS 开发人员社区成了计算行业最友好、最有帮助的社区之一。本书再次展示了 Kyle 和 Joe 渴望与人分享知识的品质。

我总是将本书与 Marks、LaMarche、Sadun 的开天辟地之作一起，放在触手可及的地方。2008 年我开始学习 iOS 应用开发时，如果能够参考本书，那该多好！拥有本书对您来说真是幸事。

——Kirby Turner

White Peak Software 首席程序猿；著有 *Learning iPad Programming, A Hands on Guide to Building Apps for the iPad, Second Edition*；Cocoa 开发人员社区的组织者，热衷于参加各种会议
于 2013 年 8 月 28 日

关于作者

Kyle Richter 是备受赞誉的 iOS 和 Mac 开发公司 Dragon Forged Software 的创始人，还是 iOS 应用开发咨询公司 Empirical Development 的联合创始人。Kyle 于 20 世纪 90 年代初涉足编程，一直致力于 Mac 平台的开发。他著有多部 iOS 开发方面的著作，还在众多深受欢迎的开发人员博客和网站上撰写文章。他管理着一个由 20 多名专职 iOS 开发人员组成的团队，还负责三家程序开发公司的日常运营工作。Kyle 在世界各地就程序开发和创业发表演讲，当前与博德牧羊犬 Landis 居住在基韦斯特。

Joe Keeley 是 Dragon Forged Software 公司的 CTO，还是 Empirical Development 的项目负责人，当前正致力于开发 Resolve 和 Slender，还领导大家为客户完成了众多项目。从接触到 Apple II 起，Joe 就爱上了编码；在整个职业生涯中，他使用过各种技术，参与开发了大量系统项目。在美国各地召开的 iOS 和 Mac 会议上，他就多个不同的技术主题发表过演讲。Joe 现与妻子和两个女儿居住在科罗拉多州丹佛市，并渴望重拾竞技击剑这项业余爱好。

献辞

谨将本书献给我的合著者，是他驱使着我对本书稿一再修改完善。

——Kyle Richter

谨将本书献给我的妻子 Irene 和两个女儿 Audrey、Scarlet。你们无限的活力和爱是我每天前进的动力。

——Joe Keeley

致谢

如果没有众多幕后人员的巨大努力，本书就不可能付梓；虽然封面上只有两个人的名字，但本书是数十个人劳动的结晶。这里首先要感谢 Trina MacDonald，如果没有她督促我们按时交稿，本书就不会有完工的一天。Pearson 的编辑提供了极大帮助，他们指出了每一页中可能出现的错别字和技术问题。本书是各位编辑专心致志的结晶，他们是 Dave Wood、Olivia Basegio、Collin Ruffenach、Sheri Cain、Tom Cirtin、Elaine Wiley 和 Cheri Clark。

这里还要感谢 Langille Design (<http://jordanlangille.com>) 的 Jordan Langille，感谢他设计了第 3 章和第 4 章的示例游戏“打仙人掌”，他的努力让 Game Center 示例项目更加引人入胜。

为本书付出大量时间的不仅有我们两位作者，还有我们的家人和同事。这里要感谢与我们日常生活相关的每个人，感谢他们从我们的肩上接过重担，并理解著书立说需要付出大量劳动。

最后，要感谢 iOS 开发社区。为获得反馈或问题答案，我们常常求助于开发人员论坛、博客和同侪。如果没有 iOS 开发社区各位成员的艰苦努力，本书也不可能付梓。

前言

欢迎读者阅读本书。

iOS 入门级图书汗牛充栋，探讨特定主题（如 Core Data 和安全）的专著也不少，但衔接初级和高级的著作却少得可怜。

本书介绍一些中高级开发主题，这些主题不值得出专著，这并不是因为它们平淡无奇或乏善可陈，而只是不够庞大。这包括其他图书通常不介绍的主题——从使用 JSON 到访问照片库，但专业 iOS 开发人员每天都要用到这些框架。

另外，本书还简要地介绍了一些高级主题，让开发人员能够快速上手。500 页的 Core Data 专著令人望而生畏，但本书第 13 章简要地介绍了这个主题，让您能够马上着手使用它。本书简要介绍的其他高级主题还包括调试与 Instruments、UIKit、语言特性以及 iCloud。

对于诸如 Game Center 排行榜和成就、AirPrint、音乐库、地址簿和 Passbook 等主题，本书也做了全面介绍。无论您是刚开发第一个 iOS 项目还是一位经验丰富的开发人员，都能在本书中找到感兴趣的内容。

无论您发现了错误或 bug，还是有什么建议，都请通过 icf@dragonforged.com 与我们联系。我们期待您的改进建议，并将为提高本书的质量不断努力。

必须具备的知识

我们尽了最大努力来确保本书的示例和解释简单易懂，但这毕竟是部中高级著作，要读懂它，必须对 iOS 开发、Objective-C 和 C 语言有基本了解，并熟悉 Xcode、开发人员门户（Developer Portal）、iTunes Connect 和 Instruments 等工具。要学习 Objective-C 和 iOS 基本技能，请参阅 Stephen G. Kochan 的著作 *Programming in Objective-C* 以及 Maurice Sharp、Rod Strougo 和 Erica Sadun 合著的 *Learning iOS Development*。

需要的软件和硬件

虽然使用 iOS 模拟器也能开发 iOS 应用，但建议至少有一台用于测试的 iOS 设备。

- **Apple iOS 开发人员账户：**Xcode 和 iOS SDK 等 iOS 开发工具的最新版本都可从 Apple 开发人员门户 (<http://developer.apple.com/ios>) 下载。要将应用发布到 App Store 或安装到个人设备上进行测试，还需要有付费的开发人员账户，这种账户的年费为 99 美元。
- **Macintosh 计算机：**要使用 Xcode 进行 iOS 开发，需要一台运行最新版 OS X 的 Mac 计算机。
- **Internet 连接：**很多 iOS 开发功能都要求用于开发的 Mac 计算机和目标 iOS 设备有 Internet 连接。

组织结构

几乎每章都自成一体，只有介绍 Game Center 和 Core Data 的几章例外。读者可从头到尾地阅读本书，也可根据兴趣和需求阅读相关的主题；本书是为方便查阅而编写的，可助您完成众多常见的 iOS 开发任务。

对各章的内容概述如下。

- **第 1 章，“UIKit Dynamics”：**iOS 7 引入了 UIKit Dynamics，可用于给 UIView 添加类似于现实世界的动画和行为。在本章中，您将学习如何给标准对象添加力学动画、物理属性和行为。本章按从易到难的顺序演示了 7 种行为——首先是重力，最后是物体属性。
- **第 2 章，“Core Location、MapKit 和地理围栏”：**在 iOS 6 中，Apple 引入了自己提供的地图和地图数据。本章介绍如何使用 Core Location 来确定设备的位置、如何在应用中显示地图以及如何使用标注、覆盖层和说明来定制地图；还介绍了如何配置区域监视（地理围栏），在设备进入或离开指定区域时通知应用。
- **第 3 章，“排行榜”：**Game Center 排行榜为给 iOS 游戏或应用添加社交功能提供了简易途径。本章简要地介绍一款功能齐备的 iPad 游戏——“打仙人掌”，循序渐进地引导读者添加排行榜功能。您将学习实现 Game Center 排行榜所需的全部步骤，并对如何使用自定义界面显示排行榜有大致认识。
- **第 4 章，“成就”：**本章改进第 3 章介绍的游戏“打仙人掌”，您将学习如何在功能齐备的 iPad 游戏中实现 Game Center 成就。本章首先介绍如何使用 iTunes Connect 显示成就进度，提供了马上着手实现成就所需的全部知识。
- **第 5 章，“地址簿”：**对众多现代项目来说，集成用户的通讯录是至关重要的一步。Address Book 是最古老的 iOS 框架之一，本章将介绍如何使用这个框架。您将学习如何使用联系人选择器、如何访问地址簿原始数据以及如何修改和保存这些数据。
- **第 6 章，“使用音乐库”：**本章介绍如何在应用中访问用户收藏的音乐，包含如何使用

音乐的元数据以及如何选择并播放音乐。

- **第 7 章，“使用和分析 JSON”**：JSON（JavaScript Object Notation）是一种在不同计算平台和架构之间传输数据的轻量级方式，已成为 iOS 客户端应用同服务器交换复杂数据集的首选方式。本章描述如何根据既有 iOS 对象创建 JSON 以及如何将 JSON 转换为 iOS 对象。
- **第 8 章，“iCloud”**：本章介绍如何使用 iCloud 在设备之间同步键/值存储和文档，包括如何配置应用以支持 iCloud、如何实现键/值存储和基于文档的存储，以及如何检测并解决冲突。
- **第 9 章，“通知”**：iOS 支持两种通知。一是本地通知，它能够在设备上独立地运行，不需要联网；一是远程通知，这需要一台服务器，它通过网络将推送通知经 Apple 推送通知服务发送给设备。本章阐述这两种通知之间的差别，并演示如何在应用中实现它们。
- **第 10 章，“使用 Game Kit 蓝牙联网技术”**：本章详尽地介绍如何创建基于蓝牙的实时聊天客户端，让您能够使用蓝牙连接到朋友的设备，并相互发送文本消息。您将学习如何使用 Game Kit 蓝牙功能，查找要连接的对等设备并发送数据。
- **第 11 章，“AirPrint”**：AirPrint 是常被低估的 iOS 功能之一，它让用户能够将文档和多媒体打印到任何与 AirPrint 兼容的无线打印机。本章介绍如何在应用中快速而轻松地支持 AirPrint，让用户能够打印视图、图像、PDF 乃至渲染的 HTML。
- **第 12 章，“Core Data 简介”**：Core Data 是个庞大而令人生畏的主题，本章尝试从门外汉的角度出发，指出在什么情况下 Core Data 可能是不错的选择，在什么情况下使用它犹如牛刀杀鸡，还以简单易懂的方式阐述一些基本的 Core Data 概念。
- **第 13 章，“使用 Core Data”**：本章演示如何配置应用以使用 Core Data、如何建立 Core Data 数据模型以及如何在应用中实现众多常用的 Core Data 工具。如果您只想马上着手使用 Core Data，又不想去啃 500 页的专著，那么本章非常适合您阅读。
- **第 14 章，“语言特性”**：自 iOS 面世后，Objective-C 一直在不断演化。本章介绍一些语言和编译器层面的变化，指出开发人员如何以及为何要使用这些新特性。这包括用于数字、数组和字典的字面量语法；块、ARC、属性声明；一些古老而优秀的特性，如句点表示法、快速枚举和方法替换。
- **第 15 章，“使用 Social Framework 集成 Twitter 和 Facebook”**：集成社交功能是未来的发展趋势，几乎所有应用都应这样做。本章引领您使用 Social Framework 在应用中添加 Facebook 和 Twitter 功能。您将学习如何使用内置的书写器创建 Twitter 和 Facebook 消息；您还将学习如何从这两种服务获取并分析摘要信息；最后，您将学习如何使用这个框架通过自定义界面发送消息。阅读本章后，您将牢固地掌握 Social Framework，能够使用 Twitter 和 Facebook 给应用添加社交功能。

- **第 16 章，“执行后台任务”**：iOS 4 引入了一种新功能，让应用即便不在前台也能执行任务，随后又添加了其他后台功能。本章阐述如何在应用从前台进入后台后执行任务以及如何执行 iOS 允许的后台活动。
- **第 17 章，“使用 GCD 改善性能”**：如果在主线程中执行资源密集型操作，可能导致应用反应迟缓。本章阐述 GCD 提供的多种技术，使用这些技术可并行地执行繁重的任务，而不影响主线程的性能。
- **第 18 章，“使用钥匙串保护数据”**：保护数据是应用开发中重要的一步，但常常被忽视。最近几年常常爆出新闻，说著名大型公司以明文方式存储用户的信用卡信息。本章简要地介绍如何使用钥匙串来保护用户数据，并将安全视为整个开发过程中不可或缺的一部分。阅读本章后，您将能够使用钥匙串保护用户设备上的少量数据，让他们把心放在肚子里。
- **第 19 章，“使用图像和滤镜”**：本章首先介绍一些基本的图像处理技巧，然后探讨一些使用 Core Image 对图像应用滤镜的高级技巧。本章的示例应用让您能够探索 Core Image 提供的各种滤镜，还能以交互方式实时串接多个滤镜。
- **第 20 章，“集合视图”**：集合视图是 iOS 6 新增的一个功能强大的 API，它提供的工具让开发人员能够灵活地排列基于单元格的可滚动内容。除新的内容布局选项外，集合视图还提供了激动人心的动画功能，使得可以以动画方式在集合视图中增删内容以及在不同集合视图布局之间切换。本章的示例应用演示了如何创建基本的集合视图、自定义的流式集合视图以及自定义的非线性集合视图布局。
- **第 21 章，“TextKit 简介”**：iOS 7 引入了 TextKit，它比 Core Text 易用，同时极大地扩展了 Core Text 的功能。TextKit 让开发人员能够在应用中提供交互式富文本，虽然这是个极其庞大的主题，但本章为您实现多种常见任务（从文本绕图到设置自定义字体属性）打下了坚实的基础。阅读本章后，您将对 TextKit 有深入认识，为进一步探索 TextKit 打下坚实的基础。
- **第 22 章，“手势识别器”**：本章阐述如何在应用中使用手势识别器。手势识别器为您识别并响应手势提供了简单而清晰的途径，让您无需直接处理和解读触摸数据。另外，您还可使用手势识别器定义和识别自定义手势。
- **第 23 章，“访问照片库”**：大家上传到 Flickr 等网站的照片数量表明，iPhone 实际上已成为深受欢迎的相机。本章阐述如何在应用中访问用户的照片库以及如何处理和视频。本章的示例应用演示了如何重建 iOS 6 应用“照片”。
- **第 24 章，“Passbook 和 PassKit”**：在 iOS 6 中，Apple 新增了 Passbook，这个独立的应用可用于存储机票、优惠券、会员卡、音乐会门票等凭证。本章阐述如何设计凭证，如何创建和分发凭证以及如何在应用中与凭证交互。

- **第 25 章，“调试和 Instruments”**：在应用开发中，最重要的一个方面是有能力调试和剖析应用。这个主题很少有人介绍，哪怕是只言片语。本章简要地介绍如何在 Xcode 中进行调试以及如何使用 Instruments 分析性能。首先简要地介绍了计算机 bug 的历史以及常用的调试技巧；然后，简要地介绍了断点和调试器命令；最后，介绍了如何使用 Time Profiler 剖析应用以及如何使用 Leaks 分析内存使用情况。阅读本章后，您将对如何诊断和调试 iOS 应用有清晰认识，无论是在模拟器还是设备上诊断和调试。

示例代码说明

本书的每章都自成一体，因此除第 25 章、第 12 章和第 14 章外，每章都有一个示例项目。第 13 章和第 14 章虽然使用的是同一个基本项目，但以不同的方式对这个项目进行了扩展。在每章开头，都简要地介绍了示例项目，并详细阐述了其中与当前主题无关但比较复杂的部分。

我们竭尽全力让示例代码简单易懂，这常常导致代码并非最优或并非解决问题的最佳方式。在这种情况下，我们将指出那些对实际应用来说不合适的做法。这些示例项目并非最终的应用，设计它们旨在方便讨论相关的功能。示例项目没有考虑具体情况，这是有意为之的，旨在避免不相关的代码干扰读者，让读者能够将注意力集中在相关的代码上。我们竭尽所能地删除不必要的代码，将代码压缩到尽可能最少。

示例项目没有使用自动引用计数（ARC），很多读者发现这一点后大为惊讶，这也是有意为之的，因为放下内存管理的包袱比背上它容易。为满足读者的不同需求，提供了两个版本的示例代码：ARC 版和非 ARC 版。在示例代码中，类名都使用了前缀 ICF，而大多数示例项目都是根据章名命名的。

在第 3 章和第 4 章，束 ID 与真实应用相关联，并使用了笔者的 Apple 账户，这旨在确保这些示例能够正常运行；另外，与这个示例项目交互时，填写多名用户的数据可能会有所帮助。在第 8 章、第 9 章和第 25 章，详细介绍了必须对应用所做的配置，要让这些项目能够正常运行，您必须使用自己的开发账户配置新 App ID。

下载示例代码

要下载本书源代码的最新版本，可前往 <https://github.com/dfsw/icf>。这些代码是开源的，任何人都可下载。下载包中有两个文件夹，分别包含 ARC 代码和非 ARC 代码。每章都有一个独立的文件夹，其中有且只有一个项目。欢迎读者就这些源代码提供反馈和建议，以便我们不断地改进和优化。

安装 Git 及使用 GitHub

Git 是一种版本控制系统，近年来日益受到大家欢迎。要克隆（clone）并使用 GitHub 上的代码，需要安装 Git；当前，可从 <http://code.google.com/p/git-osx-installer> 下载 Git 安装程序。另外，还有多个 Git GUI 前端，它们都是 GitHub 开发的，适合不喜欢命令行界面的开发人员使用。如果您不想安装 Git，也可从 GitHub 下载 Zip 格式的源代码文件。

您可前往 <https://github.com/signup/free> 免费注册 GitHub 账户。安装 Git 后，从终端命令行执行命令 `$git clone git@github.com:dfsw/icf.git`，将源代码副本下载到当前工作目录。欢迎您提交代码，为改进示例项目做贡献。

与作者联系

如果您在阅读本书时遇到问题或者有什么建议，请给我们发电子邮件（邮件地址为 icf@dragonforged.com），也可通过 Twitter（[@kylerichter](#) 和 [@jwkeeley](#)）与我们联系。

目录

| | | | |
|---|----|-------------------------------|----|
| 第 1 章 UIKit Dynamics | 1 | 2.3 显示地图 | 22 |
| 1.1 示例应用 | 1 | 2.3.1 理解坐标系 | 22 |
| 1.2 UIKit Dynamics 简介 | 2 | 2.3.2 配置和定制 MKMapView | 22 |
| 1.3 实现 UIKit Dynamics | 2 | 2.3.3 响应用户交互 | 24 |
| 1.3.1 重力 | 3 | 2.4 地图注释和覆盖层 | 25 |
| 1.3.2 碰撞 | 4 | 2.4.1 添加注释 | 25 |
| 1.3.3 连接 | 6 | 2.4.2 显示标准和自定义的注释 视图 | 27 |
| 1.3.4 弹簧效果 | 7 | 2.4.3 可拖曳的注释视图 | 30 |
| 1.3.5 吸附 | 8 | 2.4.4 使用地图覆盖层 | 31 |
| 1.3.6 推力 | 8 | 2.5 地理编码和反向地理编码 | 32 |
| 1.3.7 物体属性 | 10 | 2.5.1 对地址进行地理编码 | 32 |
| 1.4 深入了解 UIDynamicAnimator 和 UIDynamicAnimatorDelegate | 11 | 2.5.2 对位置进行反向地理编码 | 36 |
| 1.5 小结 | 12 | 2.6 地理围栏 | 38 |
| 1.6 练习 | 12 | 2.6.1 检查区域监视功能 | 38 |
| 第 2 章 Core Location、MapKit 和 地理围栏 | 13 | 2.6.2 定义边界 | 39 |
| 2.1 示例应用 | 13 | 2.6.3 监视变化 | 40 |
| 2.2 获取用户的位置 | 13 | 2.7 获取线路 | 41 |
| 2.2.1 需求和许可 | 14 | 2.8 小结 | 46 |
| 2.2.2 检查定位服务是否已开启 | 17 | 2.9 练习 | 46 |
| 2.2.3 开始位置请求 | 17 | 第 3 章 排行榜 | 47 |
| 2.2.4 分析和理解位置数据 | 19 | 3.1 “打仙人掌”游戏 | 47 |
| 2.2.5 重大变化通知 | 20 | 3.1.1 生成仙人掌 | 49 |
| 2.2.6 使用 GPX 文件进行位置 测试 | 20 | 3.1.2 处理玩家与仙人掌的交互 | 51 |
| | | 3.1.3 显示生命数和得分 | 53 |
| | | 3.1.4 暂停和继续 | 54 |

| | | | | | |
|-----------------|------------------|----|--------------------|------------|-----|
| 3.1.5 | 对游戏“打仙人掌”的最后思考 | 55 | 4.10 | 重置成就 | 92 |
| 3.2 | iTunes Connect | 56 | 4.11 | 更深入地了解成就 | 93 |
| 3.3 | 游戏中心管理器 | 58 | 4.12 | 小结 | 94 |
| 3.4 | 身份验证 | 60 | 4.13 | 练习 | 94 |
| 3.4.1 | 常见的身份验证错误 | 61 | 第 5 章 地址簿 | | 97 |
| 3.4.2 | iOS 6 身份验证 | 63 | 5.1 | 为何支持地址簿很重要 | 97 |
| 3.5 | 提交得分 | 65 | 5.2 | 地址簿编程的限制 | 97 |
| 3.5.1 | 在游戏“打仙人掌”中添加得分功能 | 67 | 5.3 | 示例应用简介 | 98 |
| 3.5.2 | 显示排行榜 | 69 | 5.4 | 使用地址簿 | 99 |
| 3.5.3 | 得分挑战 | 71 | 5.4.1 | 从地址簿读取数据 | 100 |
| 3.5.4 | 更深入地了解排行榜 | 72 | 5.4.2 | 从地址簿读取多值数据 | 101 |
| 3.6 | 小结 | 74 | 5.4.3 | 理解地址簿标签 | 102 |
| 3.7 | 练习 | 74 | 5.4.4 | 使用地址 | 103 |
| 第 4 章 成就 | | 75 | 5.5 | 地址簿图形用户界面 | 104 |
| 4.1 | iTunes Connect | 75 | 5.6 | 以编程方式创建联系人 | 109 |
| 4.2 | 显示成就进度 | 77 | 5.7 | 小结 | 111 |
| 4.3 | 游戏中心管理器和身份验证 | 78 | 5.8 | 练习 | 112 |
| 4.4 | 成就缓存 | 78 | 第 6 章 使用音乐库 | | 113 |
| 4.5 | 报告成就 | 80 | 6.1 | 示例应用简介 | 113 |
| 4.6 | 添加成就钩子 | 82 | 6.2 | 打造播放引擎 | 114 |
| 4.7 | 完成横幅 | 83 | 6.2.1 | 注册播放通知 | 115 |
| 4.8 | 成就挑战 | 83 | 6.2.2 | 播放控制 | 116 |
| 4.9 | 在游戏“打仙人掌”中添加成就功能 | 85 | 6.2.3 | 响应状态变化 | 118 |
| 4.9.1 | 一步获得的成就 | 86 | 6.2.4 | 时长和定时器 | 122 |
| 4.9.2 | 可分多步获得的成就 | 88 | 6.2.5 | 随机播放和重复播放 | 123 |
| 4.9.3 | 可分多次获得的成就 | 89 | 6.3 | 媒体选择器 | 123 |
| 4.9.4 | 捎带式成就及存储准确的成就进度 | 90 | 6.4 | 以编程方式选择媒体 | 125 |
| 4.9.5 | 基于时间的成就 | 91 | 6.4.1 | 随机选择歌曲 | 126 |
| | | | 6.4.2 | 使用谓词选择歌曲 | 127 |
| | | | 6.5 | 小结 | 129 |
| | | | 6.6 | 练习 | 129 |