

SAMS
**Teach
Yourself**

- 全球销量逾百万册的系列图书
- 连续十余年打造的经典品牌
- 直观、循序渐进的学习教程
- 掌握关键知识的最佳起点
- 秉承Read Less, Do More (精读多练) 的教学理念
- 以示例引导读者完成最常见的任务

每章内容针对初学者精心设计, **1**小时轻松阅读学习,
24小时彻底掌握关键知识

每章**案例与练习题**助你轻松完成常见任务,
通过**实践**提高应用技能, 巩固所学知识

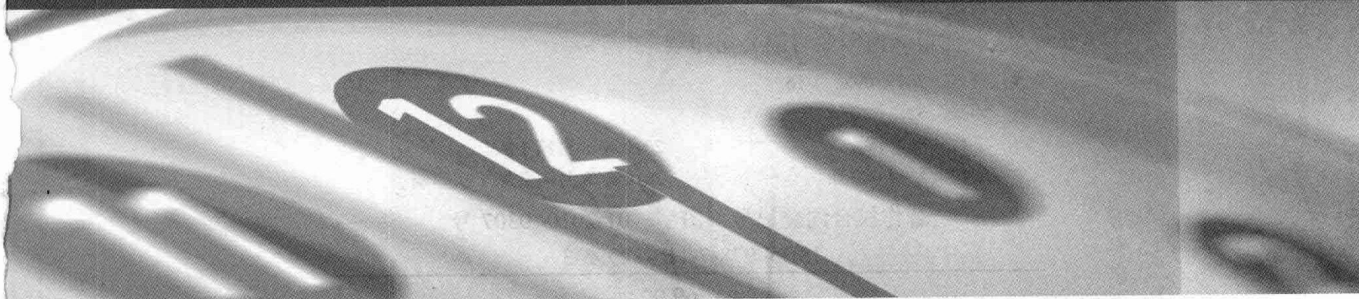
iPhone 开发 入门经典

[美] John Ray 著
Sean Johnson
袁国忠 译

iPhone 开发 入门经典

[美] John Ray 著
Sean Johnson

袁国忠 译



人民邮电出版社

北京

图书在版编目 (C I P) 数据

iPhone开发入门经典 / (美) 雷 (Ray, J.), (美) 詹森 (Johnson, S.) 著; 袁国忠译. — 北京: 人民邮电出版社, 2010.8
ISBN 978-7-115-22995-3

I. ①i… II. ①雷… ②詹… ③袁… III. ①移动通信—携带电话机—应用程序—程序设计 IV. ①TN929.53

中国版本图书馆CIP数据核字(2010)第079648号

版权声明

John Ray, Sean Johnson: Sams Teach Yourself iPhone Application Development in 24 Hours
ISBN: 9780672330841

Copyright © 2010 by Sams Publishing.

Authorized translation from the English language edition published by Sams.

All rights reserved.

本书中文简体字版由美国 Sams 出版公司授权人民邮电出版社出版。未经出版者书面许可, 对本书任何部分不得以任何方式复制或抄袭。

版权所有, 侵权必究。

iPhone 开发入门经典

-
- ◆ 著 [美] John Ray Sean Johnson
译 袁国忠
责任编辑 刘映欣
 - ◆ 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 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
三河市潮河印业有限公司印刷
 - ◆ 开本: 787×1092 1/16
印张: 32
字数: 802 千字 2010 年 8 月第 1 版
印数: 1-4 000 册 2010 年 8 月河北第 1 次印刷

著作权合同登记号 图字: 01-2010-0307 号

ISBN 978-7-115-22995-3

定价: 69.00 元

读者服务热线: (010)67132705 印装质量热线: (010)67129223

反盗版热线: (010)67171154

内容提要

本书基于 Apple 最新发布的 iPhone OS 3.0 编写,循序渐进地介绍了从事 iPhone 开发所需的基本知识,包括使用 Xcode、Interface Builder、Objective-C 和 Cocoa Touch 等开发工具,设计及美化用户界面,读写和显示数据,播放和录制多媒体,调试、优化和发布应用程序等主题。

本书通过简洁的语言和详细的步骤,帮助读者迅速掌握开发 iPhone 应用程序所需的基本知识,适合没有任何编程经验的新手阅读,也可供有志于从事 iPhone 开发的人员参考。

作者简介

John Ray 当前是俄亥俄州立大学研究基金会的高级商业分析师兼开发团队经理。他在 Macmillan、Sams 和 Que 公司出版了大量图书，其中包括 *Special Edition Using TCP/IP*、*Teach Yourself Dreamweaver MX in 21 Days*、*Mac OS X Unleashed* 和 *Macromedia All-In-One*。作为一名从 1984 年起就开始使用 Macintosh 的用户，他努力确保在每个项目中都平等对待 Macintosh。即使在编写技术性图书（如 *Special Edition Using TCP/IP*）时，也在其中介绍了大量有关 Macintosh 及其应用程序的信息。他的写作手法简洁明快，即使对初中级读者来说也简单易懂，因此深受好评。

Sean Johnson 很久以前就是一名 Mac 开发人员，拥有超过 15 年的产品开发经验，受聘的企业包括小型独立软件开发商、新兴企业以及诸如 IBM 等大型软件开发商。他在 Mac Developer Network 上开辟了一个有关产品设计的专栏，还为 IBM developerWorks 和各种杂志撰写文章。Sean 经营着产品开发咨询公司 Snooty Monkey, LLC，该公司致力于为创建出类拔萃的 Web、Mac 和 iPhone 应用程序提供技术咨询。他与妻子和两个孩子居住在风光秀丽的北卡罗来纳州教堂山市。

献 辞

献给孜孜以求的读者。

——John Ray

献给我的妻子兼最好的朋友 Michelle；献给我的好孩子 Samantha 与 Scott，你们也许再也不会听到类似“别闹，我正在写书呢”的话了。

——Sean Johnson

致 谢

万分感谢 Sams Publishing 出版社的编辑小组成员 Laura Norman、Patrick Kanouse、Keith Cline、Matthew David，感谢你们确保本书顺利付梓且语言流畅易懂。感谢朋友和家人在我写作本书期间给予的帮助，还要感谢我的草坪耐心地等待我去修剪。

最重要的是，感谢读者有兴趣学习 iPhone 开发，我迫不及待地想在 App Store 看到你开发的应用程序。

——John Ray

如果没有我美丽、勤劳的妻子 Michelle，本书就不可能面世。当我坐在电脑前夜以继日地写作时，她做了大量需要完成的工作，深表感谢！

合著者 John Ray 经验丰富、能力超群，能够与他合作我深感荣幸。还要感谢 Pearson 公司杰出的编辑、技术和制作专业人员，本书无疑是团队协作的产物，置身如此杰出的团队真是三生有幸。

还要感谢宠物鼠 Carolina 和 Savannah 在我写作期间始终在办公室陪伴左右，还没有哪位作者的同伴像它们那样对写作过程如此充满敬意。

感谢 Steve Jobs、Jonathan Ives 以及 Apple 公司成百上千位参与开发 iPhone 的工程师和设计师。无论是使用 iPhone、开发 iPhone 应用程序还是从事有关 iPhone 的写作，都充满乐趣。它给我及众多其他的人带来了灵感，让我想起我为何酷爱开发软件。它太好了，远远超过了我的预期。

最后，要感谢“美国最宜居小城”——北卡罗来纳州的教堂山市，感谢它给我提供了一个居住并获得灵感的胜地。

前言

两年前 Apple 公司还未涉足手持计算领域，而现在却在該领域占据了统治地位。iPhone 和 iPod Touch 平台改变了公众对移动计算设备的看法。凭借着功能齐备的应用程序和界面架构，iPhone 证明了小屏幕也能成为高效的工作区，并成为用户和开发人员首选的智能手机。

让 iPhone 如此成功的部分原因是神奇的界面和高效的软件分发方法。设计 iPhone 操作系统旨在让用户能够使用手指（而不是光笔或键盘）来控制手机，应用程序使用起来自然而有趣，其外观和行为不再像笨拙的桌面应用程序的移植版本。现在就看 Windows Mobile 的了！

通过 App Store，Apple 向开发人员提供了一种最佳的数字分发系统。任何年龄和派别的程序员都可将其应用程序提交到 App Store，且需要支付的年度开发人员会费不多。在 2008 年，简单的音效应用程序每天的利润就达数千美元，而诸如 Koi Pond 等游戏销售了数十万份。鉴于 iPhone 用户群庞大，不管什么内容都能找到合适的用户。

在 2009 年中期，Apple 通过发布 iPhone OS 3.0 给 iPhone 平台添加了众多新功能，还发布了 iPhone 3GS 平台。这种改进给开发人员提供了创建独特而充满创意的游戏、实用程序和应用程序所需的工具。

本书旨在向新一代开发人员介绍 iPhone 开发，以循序渐进的方式提高他们的技能。读者只需通过 24 个课时的学习，就能掌握所有基本知识——从安装开发工具、向 Apple 注册到将应用程序提交到 App Store。

谁能成为 iPhone 开发人员

只要有学习兴趣、有时间探索并实践 Apple 开发工具并拥有一台 Intel Macintosh 计算机，便可开始 iPhone 开发了。

学习开发 iPhone 应用程序并不能一蹴而就，但只要花时间多加练习，只需几天便能开发应用程序。在 Apple 开发工具上花费的时间越多，创建出激动人心的应用程序的可能性就越大。

在 iPhone 应用程序开发中，要以创建自己想用的软件为宗旨，而不是创建你认为其他人想用的软件。如果只想着一夜暴富，你很可能会失望。虽然其空间很大，但 App Store 中争夺销售排

行榜的竞争非常激烈。然而，如果将重点放在创建有用而独特的应用程序上，你的劳动成果得到用户赏识的可能性将大得多。

本书适合的读者

本书是为从未进行 iPhone 开发但使用过 Macintosh 平台的读者编写的，读者不需要有 Objective-C、Cocoa 和 Apple 开发工具方面的任何经验。当然，如果你有一定的开发经验，将更容易掌握这些工具和技术，但作者假定读者以前没有从事过编码工作。

虽然如此，但对读者还是有一定的要求。具体地说，读者必须愿意花时间学习。如果读者只是阅读每章的内容，而不完成其中的项目，很可能会错过一些重要概念。另外，读者还需花时间阅读 Apple 开发文档并研究本书介绍的主题。有关 iPhone 开发的信息浩如烟海，而本书的篇幅有限。本书为你进行自我探索打下了坚实的基础。

本书的内容

本书是基于 iPhone OS 3.0 编写的，其中的大部分内容都适用于所有 iPhone OS 版本，但也介绍了 iPhone OS 3.0 新增的一些功能，如辅助功能、A/V 库、地图集成等。

不足的是，本书并非完整的 iPhone 应用程序编程接口（API）参考手册，因为本书的篇幅无法满足有些主题的需求；所幸的是，你将在学习第 1 章内容时下载的免费工具中包含 Apple 开发文档。很多章都包含名为“进一步探索”的一节，这些内容为你学习其他感兴趣的主体提供了指南。这里需要重申的是，探索精神是成功的 iPhone 开发人员必须具备的重要品质。

涉及编码的每章都有配套的项目文件，其中包含编译并测试示例所需的一切；但更佳的做法是根据书中的介绍自己创建应用程序。请务必从本书的配套网站 <http://iphonein24hours.com> 下载项目文件。

本书使用的约定

本书使用了一些约定，让读者能够充分利用和参考书中的信息。

- 需要你在代码中输入的内容用粗体表示。
- 菜单选项用→分开。例如，指示读者打开菜单 File 并选择菜单项 New Project 时，表示为“选择菜单 File→New Project”。
- 书中列出的有些语句过长，一行容纳不下。在这种情况下，将在合理的地方换行；但有时将使用续行字符指出下一行是当前语句的一部分。

目 录

第 1 章 为开发准备好系统和 iPhone.....1	
1.1 欢迎进入 iPhone 平台.....1	
1.1.1 显示屏和图形.....1	
1.1.2 应用程序资源约束.....2	
1.1.3 连接性.....3	
1.1.4 输入和反馈.....3	
1.2 成为 iPhone 开发人员.....3	
1.2.1 加入 Apple 开发人员计划.....4	
1.2.2 安装 iPhone 开发工具.....6	
1.3 创建开发供应配置文件.....7	
1.3.1 什么是开发供应配置文件.....7	
1.3.2 生成并安装开发供应配置文件.....8	
1.3.3 使用 iPhone 应用程序测试 配置文件.....14	
1.4 开发技术概述.....15	
1.4.1 Apple 开发套件.....16	
1.4.2 Objective-C.....16	
1.4.3 CoCoa Touch.....16	
1.4.4 模型-视图-控制器.....16	
1.5 小结.....17	
1.6 问与答.....17	
1.7 作业.....17	
1.7.1 测验.....17	
1.7.2 答案.....17	
1.7.3 练习.....18	
第 2 章 Xcode 和 iPhone 模拟器简介.....19	
2.1 使用 Xcode.....19	
2.1.1 创建和管理项目.....20	
2.1.2 删除文件和资源.....23	
2.1.3 编辑和导航代码.....24	
2.1.4 生成应用程序.....28	
2.1.5 项目属性.....30	
2.2 使用 iPhone 模拟器.....31	
2.2.1 在模拟器中启动应用程序.....32	
2.2.2 模拟多点触摸事件.....33	
2.2.3 旋转 iPhone.....33	
2.2.4 模拟其他情况.....34	
2.3 小结.....35	
2.4 问与答.....35	
2.5 作业.....35	
2.5.1 测验.....35	
2.5.2 答案.....35	
2.5.3 练习.....36	
2.6 进一步探索.....36	
第 3 章 探索 Apple 平台语言 Objective-C.....37	
3.1 面向对象编程和 Objective-C.....37	
3.1.1 什么是面向对象编程.....38	
3.1.2 面向对象编程术语.....39	
3.1.3 什么是 Objective-C.....40	
3.2 探索 Objective-C 文件结构.....40	
3.2.1 头文件/接口文件.....41	
3.2.2 实现文件.....44	
3.2.3 免费获取结构.....45	

3.3 Objective-C 编程基础.....45	第 5 章 探索 Interface Builder.....73
3.3.1 声明变量.....46	5.1 了解 Interface Builder.....73
3.3.2 分配、初始化和释放对象.....47	5.1.1 Interface Builder 采用的方法...74
3.3.3 使用方法及发送消息.....48	5.1.2 剖析 Interface Builder XIB 文件.....74
3.3.4 表达式和决策.....49	5.2 创建用户界面.....77
3.4 内存管理.....52	5.2.1 对象库.....77
3.4.1 释放对象.....53	5.2.2 布局工具.....78
3.4.2 使用 autorelease 方法.....53	5.3 定制界面外观.....81
3.4.3 保留对象.....53	5.3.1 使用属性检查器.....81
3.4.4 在 dealloc 中释放实例变量.....54	5.3.2 设置辅助功能属性.....82
3.4.5 释放规则.....54	5.3.3 测试界面.....83
3.5 小结.....54	5.4 连接到代码.....84
3.6 问与答.....55	5.4.1 在 Xcode 中启动 Interface Builder.....84
3.7 作业.....55	5.4.2 实现概述.....85
3.7.1 测验.....55	5.4.3 输出口和操作.....86
3.7.2 答案.....55	5.4.4 对象身份.....89
3.7.3 练习.....56	5.5 小结.....89
3.8 进一步探索.....56	5.6 问与答.....90
第 4 章 Cocoa Touch 内幕.....57	5.7 作业.....90
4.1 Cocoa Touch 是什么.....57	5.7.1 测验.....90
4.1.1 冷静面对大量的功能.....58	5.7.2 答案.....90
4.1.2 年轻但成熟.....58	5.7.3 练习.....91
4.2 探索 iPhone 技术层.....59	5.8 进一步探索.....91
4.2.1 Cocoa Touch 层.....59	第 6 章 模型-视图-控制器应用程序设计...92
4.2.2 多媒体层.....60	6.1 了解设计模式模型-视图- 控制器.....92
4.2.3 核心服务层.....60	6.1.1 制作“意大利面条式”代码...93
4.2.4 核心 OS 层.....61	6.1.2 使用 MVC 将应用程序设计 结构化.....93
4.3 跟踪 iPhone 应用程序的生命 周期.....62	6.2 Xcode 和 Interface Builder 如何 实现 MVC.....94
4.4 Cocoa 基础.....63	6.2.1 视图.....94
4.4.1 核心应用程序类.....63	6.2.2 视图控制器.....94
4.4.2 数据类型类.....64	6.2.3 数据模型.....95
4.4.3 界面类.....66	6.3 使用模板 View-Based Application.....96
4.5 使用 Xcode 探索 iPhone 框架.....68	6.3.1 实现概述.....96
4.5.1 Xcode 文档.....68	6.3.2 创建项目.....97
4.5.2 快速帮助.....70	6.3.3 在视图控制器中创建输出口和 操作.....100
4.6 小结.....71	6.3.4 创建视图.....101
4.7 问与答.....71	
4.8 作业.....72	
4.8.1 测验.....72	
4.8.2 答案.....72	
4.8.3 练习.....72	

6.3.5 实现视图控制器逻辑	105	8.2.5 添加图像视图	134
6.3.6 释放内存	106	8.2.6 添加滑块	137
6.3.7 生成应用程序	106	8.2.7 完成界面的创建	140
6.4 小结	107	8.2.8 实现视图控制器逻辑	141
6.5 问与答	107	8.2.9 释放对象	143
6.6 作业	107	8.2.10 生成应用程序	143
6.6.1 测验	107	8.3 小结	144
6.6.2 答案	107	8.4 问与答	144
6.6.3 练习	107	8.5 作业	144
6.7 进一步探索	108	8.5.1 测验	144
第7章 使用文本、键盘和按钮	109	8.5.2 答案	144
7.1 基本用户输入和输出	109	8.5.3 练习	144
7.1.1 按钮	109	8.6 进一步探索	145
7.1.2 文本框和文本视图	110	第9章 使用高级界面控件	146
7.1.3 标签	110	9.1 用户输入和输出	146
7.2 使用文本字段、文本视图和 按钮	110	9.1.1 开关	146
7.2.1 实现概述	111	9.1.2 分段控件	147
7.2.2 创建项目	111	9.1.3 Web 视图	148
7.2.3 创建输出口和操作	112	9.1.4 可滚动的视图	149
7.2.4 添加文本框	113	9.2 使用开关、分段控件和 Web 视图	149
7.2.5 添加文本视图	117	9.2.1 实现概述	149
7.2.6 创建风格独特的按钮	119	9.2.2 创建项目	150
7.2.7 隐藏键盘	125	9.2.3 创建输出口和操作	150
7.2.8 实现视图控制器逻辑	127	9.2.4 添加分段控件	151
7.2.9 释放对象	128	9.2.5 添加开关	155
7.2.10 生成应用程序	128	9.2.6 添加 Web 视图	156
7.3 小结	129	9.2.7 完成界面	158
7.4 问与答	129	9.2.8 实现视图控制器逻辑	159
7.5 作业	129	9.2.9 释放对象	162
7.5.1 测验	129	9.2.10 生成应用程序	162
7.5.2 答案	129	9.3 使用可滚动视图	163
7.5.3 练习	130	9.3.1 实现概述	163
7.6 进一步探索	130	9.3.2 创建项目	163
第8章 处理图像、动画和滑块	131	9.3.3 创建输出口	164
8.1 用户输入和输出	131	9.3.4 添加可滚动视图	164
8.1.1 滑块	131	9.3.5 实现滚动行为	166
8.1.2 图像视图	132	9.3.6 释放对象	167
8.2 创建并管理图像动画和滑块	132	9.3.7 生成应用程序	167
8.2.1 实现概述	132	9.4 小结	167
8.2.2 创建项目	133	9.5 问与答	167
8.2.3 添加动画资源	133	9.6 作业	167
8.2.4 准备输出口和操作	133	9.6.1 测验	167
		9.6.2 答案	168

9.6.3 练习	168	11.4.1 系统声音服务	207
9.7 进一步探索	168	11.4.2 播放声音和警告	208
第 10 章 使用选择器选择多种值	169	11.4.3 震动 iPhone	210
10.1 了解选择器	169	11.5 小结	210
10.1.1 日期选择器	170	11.6 问与答	210
10.1.2 选择器视图	170	11.7 作业	210
10.2 使用日期选择器	172	11.7.1 测验	210
10.2.1 实现概述	172	11.7.2 答案	211
10.2.2 创建项目	172	11.7.3 练习	211
10.2.3 添加日期选择器	173	11.8 进一步探索	211
10.2.4 完成界面	175		
10.2.5 实现视图控制器逻辑	176	第 12 章 使用工具栏创建简单的多视图应用程序	212
10.3 实现自定义选择器视图	179	12.1 探索单视图应用程序和多视图应用程序	212
10.3.1 实现概述	179	12.2 创建多视图应用程序	213
10.3.2 创建项目	180	12.2.1 实现概述	213
10.3.3 添加选择器视图	181	12.2.2 创建项目	214
10.3.4 完成界面	182	12.2.3 实例化视图控制器	215
10.3.5 向选择器提供数据	183	12.2.4 使用工具栏来切换视图	217
10.3.6 对用户在选择器中选择值做出反应	187	12.3 小结	223
10.3.7 微调选择器 UI	190	12.4 问与答	223
10.4 小结	193	12.5 作业	223
10.5 问与答	193	12.5.1 测验	223
10.6 作业	194	12.5.2 答案	223
10.6.1 测验	194	12.5.3 练习	224
10.6.2 答案	194	12.6 进一步探索	224
10.6.3 练习	194		
10.7 进一步探索	194	第 13 章 使用选项卡栏管理多视图界面	225
第 11 章 引起用户注意	196	13.1 选项卡栏和选项卡栏控制器	225
11.1 探索通知用户的方法	196	13.2 创建多视图选项卡栏应用程序	226
11.1.1 准备通知项目文件	196	13.2.1 实现概述	226
11.1.2 创建通知项目的界面	198	13.2.2 创建项目	226
11.1.3 连接到输出口和操作	198	13.2.3 添加选项卡栏控制器	228
11.2 生成警告	199	13.2.4 配置视图控制器类	230
11.2.1 显示简单警告	200	13.2.5 实现 AreaView 视图	230
11.2.2 创建多选项警告	201	13.2.6 实现 VolumeView 视图	236
11.3 使用操作表	203	13.2.7 实现 SummaryView 视图	239
11.3.1 显示操作表	204	13.3 小结	241
11.3.2 修改操作表的外观	206	13.4 问与答	242
11.3.3 对用户按下操作表按钮做出响应	206	13.5 作业	242
11.4 使用警告声音和震动	207		

13.5.1 测验	242	15.6.3 练习	291
13.5.2 答案	242	15.7 进一步探索	291
13.5.3 练习	242		
13.6 进一步探索	243		
第 14 章 使用表视图显示和导航数据	244	第 16 章 读写数据	293
14.1 了解表视图	244	16.1 应用程序沙箱	293
14.1.1 表视图的类型	244	16.1.1 应用程序数据的存储位置	294
14.1.2 实现概述	245	16.1.2 文件路径	295
14.2 创建简单的表视图应用程序	246	16.2 应用程序 FlashCards	296
14.2.1 准备视图控制器	246	16.2.1 创建 FlashCard 类	296
14.2.2 给表视图提供数据	249	16.2.2 添加输出和操作	297
14.3 响应行触摸事件	254	16.2.3 设置 UI 并连接到输出和操作	298
14.4 了解基于导航的应用程序	255	16.2.4 添加另一个视图控制器	301
14.5 创建基于导航的多视图应用程序	256	16.2.5 实现应用程序的逻辑	303
14.5.1 创建项目	256	16.3 对象归档	309
14.5.2 给应用程序提供数据	257	16.3.1 归档 FlashCard 对象	310
14.5.3 创建详细视图	260	16.3.2 在 FlashCards 应用程序中归档	311
14.5.4 实现根表视图控制器	263	16.4 Core Data	313
14.5.5 处理导航事件	265	16.4.1 Core Data 的起源	313
14.5.6 微调 UI	266	16.4.2 Core Data 简介	314
14.6 小结	267	16.4.3 应用程序 FlashCardsCore Data	315
14.7 问与答	268	16.5 小结	326
14.8 作业	268	16.6 问与答	327
14.8.1 测验	268	16.7 作业	327
14.8.2 答案	268	16.7.1 测验	327
14.8.3 练习	268	16.7.2 答案	327
14.9 进一步探索	269	16.7.3 练习	327
		16.8 进一步探索	328
第 15 章 存储应用程序首选项	270	第 17 章 创建可旋转及调整大小的用户界面	329
15.1 设计方面的考虑	270	17.1 可旋转和调整大小的界面	329
15.2 案例研究	272	17.1.1 启用界面旋转	329
15.2.1 Apple 应用程序	272	17.1.2 设计可旋转和调整大小的界面	331
15.2.2 ESPN ScoreCenter	274	17.2 使用 Interface Builder 创建可旋转和调整大小的界面	332
15.3 读写用户默认设置	274	17.2.1 创建项目	332
15.3.1 隐式首选项	275	17.2.2 创建灵活的界面	332
15.3.2 使用 Settings 应用程序	279	17.3 旋转时调整控件的框架	335
15.3.3 在应用程序中设置首选项	286	17.3.1 创建项目	336
15.4 小结	290	17.3.2 创建界面	337
15.5 问与答	291	17.3.3 实现调整框架的逻辑	340
15.6 作业	291		
15.6.1 测验	291		
15.6.2 答案	291		

17.4 旋转时切换视图	341	19.3.2 设计 UI 并连接到输出口	382
17.4.1 创建项目	341	19.3.3 实现 UIAccelerometerDelegate 协议	383
17.4.2 创建界面	342	19.4 检测移动	385
17.4.3 实现视图切换逻辑	344	19.5 小结	386
17.5 小结	346	19.6 作业	386
17.6 问与答	346	19.6.1 测验	386
17.7 作业	346	19.6.2 答案	386
17.7.1 测验	346	19.6.3 练习	387
17.7.2 答案	346	19.7 进一步探索	387
17.7.3 练习	347	第 20 章 使用多媒体	388
17.8 进一步探索	347	20.1 探索多媒体框架	388
第 18 章 扩展触摸界面	348	20.1.1 Media Player 框架	389
18.1 多点触摸手势架构	349	20.1.2 AV Foundation 框架	389
18.2 检测轻按	350	20.2 创建用于练习使用多媒体的 应用程序	389
18.2.1 添加输出口	350	20.2.1 编辑项目文件	390
18.2.2 设计 UI 并连接到输出口	351	20.2.2 创建应用程序界面	391
18.2.3 编写代码	352	20.2.3 连接到输出口和操作	392
18.3 检测轻扫	360	20.3 使用电影播放器	393
18.3.1 更新游戏	360	20.3.1 添加 Media Player 框架	393
18.3.2 识别轻扫	361	20.3.2 添加多媒体文件	393
18.4 检测张合	363	20.3.3 实现电影播放	394
18.4.1 更新游戏	363	20.3.4 执行清理工作	395
18.4.2 识别张合手势	365	20.4 录制和播放音频	396
18.5 使用摇动手势	368	20.4.1 添加 AV Foundation 框架	397
18.5.1 更新游戏	368	20.4.2 实现录音功能	397
18.5.2 检测摇动	370	20.4.3 实现音频播放功能	399
18.6 小结	373	20.4.4 执行清理工作	399
18.7 问与答	373	20.5 访问并播放 iPod 库	400
18.8 作业	373	20.5.1 实现多媒体选择器	401
18.8.1 测验	373	20.5.2 实现音乐播放器	403
18.8.2 答案	373	20.6 小结	405
18.8.3 练习	374	20.7 问与答	405
18.9 进一步探索	374	20.8 作业	405
第 19 章 使用加速计输入检测移动	375	20.8.1 测验	405
19.1 加速计简介	376	20.8.2 答案	405
19.2 检测朝向	377	20.8.3 练习	406
19.2.1 添加输出口	378	20.9 进一步探索	406
19.2.2 设计 UI 并连接到输出口	378	第 21 章 与其他应用程序交互	407
19.2.3 实现 UIAccelerometerDelegate 协议	379	21.1 从 iPhone 图像库和相机获取 图像	407
19.3 检测倾斜	382	21.2 创建并发送电子邮件	414
19.3.1 添加输出口	382		

21.3 访问地址簿	419	第 23 章 应用程序调试和优化	457
21.4 小结	425	23.1 在 Xcode 中调试	457
21.5 问与答	425	23.1.1 使用 NSLog 进行调试	457
21.6 作业	425	23.1.2 创建用于调试的项目	459
21.6.1 测验	425	23.1.3 设置断点及单步执行代码	460
21.6.2 答案	425	23.2 使用 Instruments 监视	469
21.6.3 练习	426	23.2.1 跟踪应用程序	469
21.7 进一步探索	426	23.2.2 可用的工具	472
第 22 章 实现地图和定位服务	427	23.3 使用 Shark 分析性能	473
22.1 Core Location	428	23.3.1 连接到应用程序	473
22.1.1 位置管理器	428	23.3.2 理解性能分析结果	476
22.1.2 一个使用 Core Location 的 应用程序	432	23.4 小结	478
22.1.3 添加 Core Location 框架和 输出口	432	23.5 作业	478
22.1.4 设计 UI 并连接到输出口	433	23.5.1 测验	478
22.1.5 实现位置管理器代理	435	23.5.2 答案	479
22.2 Map Kit	438	23.5.3 练习	479
22.2.1 添加 Map Kit 框架和地图 视图控制器	439	23.6 进一步探索	479
22.2.2 经过一定的延迟后显示地图 视图	439	第 24 章 通过 App Store 分发应用 程序	480
22.2.3 导入 Map Kit 框架并添加一个 输出口	441	24.1 通过 App Store 发布应用程序 前的准备工作	480
22.2.4 设计 UI	442	24.1.1 创建图标	481
22.2.5 限制地图的范围	442	24.1.2 获取 iPhone 分发证书	483
22.2.6 添加地图标注	443	24.1.3 配置项目以便分发	485
22.3 指南针	447	24.2 提交应用程序以便批准	487
22.3.1 位置管理器和位置管理器 代理	447	24.2.1 上传应用程序	488
22.3.2 添加输出口和属性	448	24.2.2 在应用程序中建造商店	490
22.3.3 设计 UI 并添加箭头图像	449	24.3 推销应用程序	491
22.3.4 请求航向更新及存储当前 位置	450	24.3.1 使用 iTunes Connect 监视和 管理销售情况	491
22.3.5 计算前往查布尔希尔的 航向	451	24.3.2 利用网站和社交网络	492
22.4 小结	454	24.3.3 更新应用程序	494
22.5 问与答	454	24.3.4 调整价格	494
22.6 作业	455	24.4 其他分发方法	495
22.6.1 测验	455	24.4.1 ad hoc 部署	495
22.6.2 答案	455	24.4.2 企业部署	497
22.6.3 练习	455	24.5 小结	497
22.7 进一步探索	456	24.6 问与答	497
		24.7 作业	498
		24.7.1 测验	498
		24.7.2 答案	498
		24.7.3 练习	498

第 1 章

为开发准备好系统和 iPhone

本章将介绍：

- 是什么成就了 iPhone？
- 到哪里获取 iPhone 开发工具？
- 如何加入 iPhone 开发人员计划（Developer Program）？
- 创建（并使用）iPhone 供应配置文件（provisioning profile）；
- iPhone 开发技术概述。

iPhone 向开发人员展示了一个全新的世界——多点触摸界面、可始终在线、视频以及众多内置的传感器，这些传感器可用于创建从游戏到提高生产率的应用程序等各种软件。信不信由你，作为新开发人员，你有一个优势：你是白纸一张，不受以前知道的手持应用程序概念的羁绊。你的高见很可能变成 Apple App Store 上的优秀作品。

本章将让你为 iPhone 开发做好准备。你即将踏上成为 iPhone 开发人员的道路，但在开始编码之前还需要做些准备工作。

1.1 欢迎进入 iPhone 平台

你正阅读本书说明你有一部 iPhone，这意味着你已经知道拥有并使用这种系统意味着什么。独特而响应迅速的界面、活泼的图形、多点触摸以及成百上千的应用程序，这些都只是冰山一角。然而，作为开发人员，你需要习惯应对一个迫使你以不同方式思考的平台。

1.1.1 显示屏和图形

iPhone 屏幕大小为 320 像素×480 像素，这只能给你提供了有限的空间来展示应用程序内容和界面，如图 1.1 所示。iPhone 应用程序还摒弃了多窗口的概念，而只能使用一个窗口。你可修改该窗口的内容，但桌面和多窗口应用程序的概念一去不复返了。

图 1.1

iPhone 的屏幕大小
为 320 像素 ×
480 像素



屏幕方面的限制并非坏事。正如你将学到的，使用 iPhone 开发工具可创建层次与桌面软件一样多的应用程序，但其界面设计的结构化程度和效率更高。

你可在屏幕上显示复杂的 2D 和 3D 动画，这要归功于所有型号的 iPhone 都支持 OpenGL ES。OpenGL 是一个定义和操纵图像的行业标准，被广泛用于创建游戏。iPhone 3GS 采用了改进的 3D 芯片和更高级的 OpenGL 版本(2.0 版)，从而改善了这些功能，但所有型号的 iPhone 都有相当不错的图像处理功能。

1.1.2 应用程序资源约束

与台式机和笔记本的高清晰度显示器一样，我们也越来越习惯于处理器的速度比单击速度快。早期 iPhone 配置的是 412 MHz ARM，而 3GS 配置的是 600 MHz 版本，它们都通过降频 (underclocked) 来延长电池的续航时间。这与 MacBook Pro 不可同日而语，但应对日常应用程序和游戏还是绰绰有余。

为确保不管用户做什么 iPhone 都能迅速响应，Apple 做出了巨大努力。遗憾的是，这给 iPhone 带来了最严重的限制之一——不能同时运行多个第三方应用程序。这意味着你的程序必须提供用户所需的功能，以免他们被迫转向其他应用程序。这还意味着你的应用程序必须处于活动状态才能与用户交流。在 iPhone OS 3.0 中，Apple 提供了一种将通知“推给”用户的方式，但在你的应用程序已退出时不会做任何实际处理。

你不能忘记的另一个约束是可用内存。在最初的 iPhone 和 iPhone 3G 中，整个系统 (包括你的应用程序) 可用的内存只有 128MB，没有虚拟内存的概念，因此你必须小心管理应用程序创建的对象。在 iPhone 3GS 中，Apple 通情达理地提供了 256MB 内存，但别忘了，无法对早期的 iPhone 进行内存升级。

By the Way

注意：

全书将不断提醒你在用完内存后将其释放。虽然你看得都有些烦了，但应习惯这个重要的过程。