



iOS

项目开发教程

郭朗 卓国锋 / 主编

孟瑞 刘盼盼 / 编著

谢安俊 / 主审



清华大学出版社



iOS 项目开发教程

郭朗 卓国锋 / 主编
孟瑞 刘盼盼 / 编著

清华大学出版社
北 京

内 容 简 介

本书基于 Apple 最新发布的 iOS 7 编写。书中循序渐进地介绍了 iOS 项目开发的一般步骤和基础知识,主要包括设计及美化用户界面,用 Interface Builder 构建视图,切换和弹出框,用导航控制器导航页面,用表视图结构化数据,读写和显示数据以及地图和定位功能等。

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 iOS 应用程序所需的基本知识,适合有一定编程经验的读者阅读。本书可作为高等学校教材,也可供从事 iOS 开发的人员参考。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

iOS 项目开发教程/郭朗,卓国锋主编. —北京:清华大学出版社,2018

ISBN 978-7-302-50901-1

I. ①i… II. ①郭… ②卓… III. ①移动终端—应用程序—程序设计—教材 IV. ①TN929.53

中国版本图书馆 CIP 数据核字(2018)第 185376 号

责任编辑:焦虹 李晔

封面设计:常雪影

责任校对:白蕾

责任印制:董瑾

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 邮 编:100084

社总机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课件下载: <http://www.tup.com.cn>, 010-62795954

印装者:三河市少明印务有限公司

经 销:全国新华书店

开 本:185mm×260mm 印 张:15.75

字 数:366 千字

版 次:2018 年 9 月第 1 版

印 次:2018 年 9 月第 1 次印刷

定 价:39.80 元

产品编号:062178-01

iOS 平台改变了公众对移动设备的看法,iPhone 凭借着功能齐备的应用程序、界面架构以及其他平台无法媲美的触控,提供了方便的手机功能,证明了小屏幕也能成为高效的工作区。

在 Apple 手机的开发人员看来,用户体验至关重要。他们设计了 iOS,其外观和行为不再像笨拙的桌面应用程序的移植版本,用户能够使用手指(而不是光笔或键盘)控制手机,从而让应用程序使用起来更加自然而有趣。

通过 AppStore,Apple 公司向开发人员提供了一种最佳的数字发布系统。开发人员可以将应用程序提交到 AppStore,且只需要支付少量的年度开发人员会费。人们开发了针对各种领域的游戏和应用程序,其范围涵盖了从学前教育到退休生活的所有阶段。由于 iPhone、iPod Touch 和 iPad 的用户群非常庞大,因此不管什么内容都能找到合适的用户。

Apple 公司每年都会发布新的 iOS 设备,其速度更快、分辨率更高、功能更强大。每次硬件更新都带来了新的开发机会,提供了将艺术融合到软件中的新途径。

1. 谁能成为 iOS 开发人员

只要有学习兴趣,有时间探索和使用 Apple 开发工具,并拥有一台运行 Lion 的 Inter Macintosh 计算机,便可开始 iOS 开发了。虽然不可能在一夜之间就开发出 iOS 应用程序,但只要多加练习,完全可以在几天内编写出第一款应用程序。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。

2. 本书适合的读者

本书是为熟悉 iOS 开发语言 Objective-C 的读者编写的,读者不需要具有 Cocoa 和 Apple 开发的经验;当然,读者如果有一定的开

发经验,将会更容易掌握这些工具和技术。尽管如此,本书对读者还是有一定要求的,就是读者必须愿意花费时间进行学习与练习。如果读者只是阅读每章的内容,而不完成其中的项目,就会错过一些重要概念。另外,读者还需认真阅读 Apple 开发文档,并研究本书介绍的主题,这样才能掌握全部内容。

编著者

目录

Contents

第 1 章 iOS 应用开发概述	1
1.1 iOS 应用开发的历史与现状	1
1.2 iOS 应用的基本架构	2
1.2.1 可触摸层	2
1.2.2 媒体层	3
1.2.3 核心服务层	4
1.2.4 核心操作系统层	5
1.3 iOS 应用开发的特点	5
1.4 iOS 开发工具简介	6
1.5 创建并运行第一个 iOS 应用	9
1.6 基础知识与技能回顾	11
练习	11
第 2 章 为开发做好准备	12
2.1 客户端的准备	12
2.2 服务端的准备	12
2.2.1 Web Services 的搭建	12
2.2.2 DB Server 的搭建	14
2.3 几个必备的基础技能	25
2.4 基础知识与技能回顾	26
练习	26
第 3 章 MyDemo 项目介绍	27
3.1 项目背景	27
3.2 项目需求分析	27
3.3 项目用例分析	27
3.4 项目数据库分析	28

3.5 基础知识与技能回顾	29
练习	29
第4章 用户注册	30
4.1 用户注册总体设计	30
4.1.1 流程图	30
4.1.2 时序图	30
4.2 数据库的准备	31
4.3 服务端接口的准备	32
4.4 用户注册的实现	32
4.4.1 客户端代码开发	32
4.4.2 客户端与服务端交互	64
4.5 用户注册的调试	71
4.6 让用户免去注册的烦恼	73
4.7 基础知识与技能回顾	73
练习	73
第5章 用户登录	74
5.1 用户登录总体设计	74
5.1.1 流程图	74
5.1.2 时序图	75
5.2 服务端接口的准备	75
5.3 用户登录的实现	75
5.3.1 客户端代码开发	76
5.3.2 客户端与服务端交互	82
5.4 用户登录的调试	84
5.5 使用第三方账号登录	86
5.5.1 什么是第三方账号	86
5.5.2 第三方账号登录方式	86
5.5.3 使用第三方账号登录	86
5.6 基础知识与技能回顾	95
练习	95
第6章 向用户展示内容	96
6.1 数据库的准备	96
6.2 服务端接口的准备	97
6.3 实现内容展示静态页面	98

6.3.1	图文列表展示	98
6.3.2	详情内容展示	115
6.3.3	客户端与服务端交互	121
6.4	图片的处理与效果实现	131
6.4.1	图片添加手势	131
6.4.2	分页与翻页	133
6.5	基础知识与技能回顾	138
练习		138
第7章	支持用户基于 LBS 的应用	139
7.1	用户定位	139
7.1.1	LBS 与常见第三方地图	139
7.1.2	在地图上找到自己	140
7.2	摇一摇	144
7.2.1	客户端代码开发	145
7.2.2	客户端与服务端交互	149
7.3	基础知识与技能回顾	151
练习		152
第8章	让用户搜索	153
8.1	服务端接口的准备	153
8.2	常用搜索方式与应用开发	154
8.2.1	客户端代码开发	154
8.2.2	客户端与服务端交互	162
8.3	基础知识与技能回顾	163
练习		163
第9章	与用户互动	164
9.1	数据库的准备	164
9.2	服务端接口的准备	165
9.3	让用户参与评价	166
9.3.1	客户端代码开发	167
9.3.2	客户端与服务端交互	178
9.4	让用户分享	187
9.4.1	什么是分享	187
9.4.2	让用户将内容分享到社交平台	187
9.5	给用户推送消息	191

9.5.1 推送原理	192
9.5.2 第三方推送介绍	192
9.5.3 集成第三方推送	193
9.6 基础知识与技能回顾	202
练习	203
第 10 章 添加商户信息	204
10.1 服务端接口的准备	204
10.2 添加商户信息的实现	205
10.2.1 客户端代码开发	205
10.2.2 客户端与服务端交互	211
10.3 基础知识与技能回顾	213
练习	213
第 11 章 让用户的使用体验更佳	214
11.1 用户网络环境	214
11.2 用户手机环境	216
11.3 基础知识与技能回顾	218
练习	218
第 12 章 发布和管理 iOS 应用	219
12.1 发布 iOS 应用	219
12.1.1 申请发布证书	219
12.1.2 发布应用到 App Store	224
12.2 版本管理	229
12.3 让用户升级	232
12.4 基础知识与技能回顾	235
练习	235
第 13 章 HTML 5	236
13.1 什么是 HTML 5	236
13.2 用 HTML 5 实现内容展示	236
13.3 基础知识与技能回顾	242
练习	242
参考文献	243

iOS 应用开发概述

本章首先介绍 iOS 应用开发历史与现状,然后对 iOS 框架进行整体介绍,最后介绍苹果开发工具包 iOS SDK 及开发环境的搭建。

1.1 iOS 应用开发的历史与现状

苹果 iOS 是由苹果公司开发的手持设备操作系统。苹果公司最早于 2007 年 1 月 9 日的 Macworld 大会上公布这个系统,最初是设计给 iPhone 使用的,后来陆续套用到 iPod touch、iPad 以及 AppleTV 等苹果产品上。iOS 与苹果的 Mac OS X 操作系统一样,也是以 Darwin 为基础的,因此同样属于类 UNIX 的商业操作系统。

1. 历史

2007 年 10 月 17 日,苹果公司发布了第一个本地化 iPhone 应用程序开发包(SDK)。2008 年 3 月 6 日,苹果公司发布了第一个测试版开发包,并且将 iPhone runs OSX 改名为 iPhoneOS。2010 年 2 月 27 日,苹果公司发布了 iPad,iPad 同样搭载了 iPhoneOS。同年,苹果公司重新设计了 iPhoneOS 的系统结构和自带程序。2010 年 6 月,苹果公司将 iPhoneOS 改名为 iOS,同时还获得了思科 iOS 的名称授权。2012 年 6 月,苹果公司在 WWDC 2012 上宣布了 iOS6,提供了超过 200 项新功能。2013 年 6 月 10 日,苹果公司在 WWDC 2013 上发布了 iOS7,几乎重绘了所有的系统 App,去掉了所有的仿实物化,整体设计风格转为扁平化设计。2014 年 6 月 3 日,苹果公司在 WWDC 2014 上发布了 iOS8,并提供了开发者预览版更新。

2. 现状

iPhone 在全球创造的庞大应用市场,使应用开发公司开始争抢 iOS 软件开发人才。另外,由于 iOS 系统开发技术位于全球手机系统的前端,其他系统平台应用开发公司和系统研发公司同时也在高薪招聘。72%的招聘公司正在招聘 iOS 平台开发人才,其中 38%的招聘公司表示,iOS 平台开发经验要比任何其他平台开发经验更受招聘公司的青睐。由于国内 iOS 软件开发起步相对较晚,人才培养机制更是远远跟不上市场的发展速度,有限的 iOS 开发人才成了国内企业必争的资源,甚至有的企业不得不考虑通过收购

来填补人才空缺。一名 iOS 开发新手要比普通软件开发新手高出约 20%~30% 的薪资，符合条件或有项目经验的开发工程师更是有价无市。

1.2 iOS 应用的基本架构

iOS 的基本架构分为四个层次：核心操作系统层(Core OS layer)、核心服务层(Core Services layer)、媒体层(Media layer)和可触摸层(Cocoa Touch layer)，如图 1.1 所示。

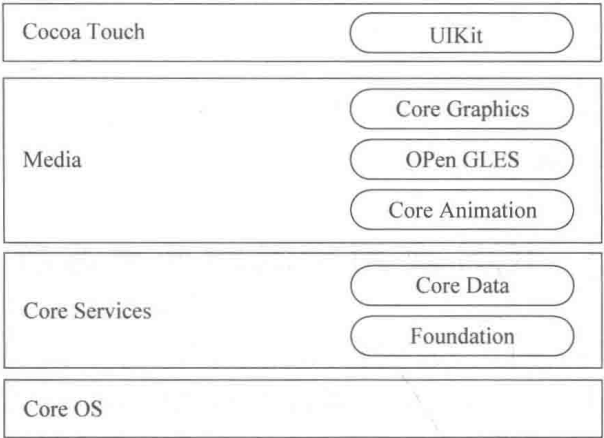


图 1.1 iOS 基本架构

1.2.1 可触摸层

可触摸层包含创建 iOS 应用程序所需的关键框架。上至实现应用程序可视界面，下至与高级系统服务交互，都需要该层技术提供底层基础。在开发应用程序的时候，尽可能不要使用更底层的框架，而是使用该层的框架。该框架中比较重要的框架如下。

1. Address Book UI 框架

Address Book UI 框架(AddressBookUI.framework)是一套 Objective-C 的编程接口，能够显示创建或者编辑联系人的标准系统界面。该框架简化了应用程序显示联系人信息的工作，另外它也可以确保应用程序使用的界面和其他应用程序相同，进而保证跨平台的一致性。

2. Game Kit 框架

iOS 3.0 引入了 Game Kit 框架(GameKit.framework)。该框架支持点对点连接及游戏内语音功能，可以通过该框架为应用程序增加点对点网络功能。此框架通过一组建构于 Bonjour 之上的简单而强大的类提供网络功能，这些类将许多网络细节抽象出来，从而让没有网络编程经验的开发者可以更加容易地将网络功能整合到应用程序中。

3. Map Kit 框架

iOS 3.0 导入了 Map Kit 框架(MapKit.framework),该框架提供一个可被嵌入到应用程序的地图界面,该界面包含一个可以滚动的地图视图。开发者可以在视图中添加定制信息,并可将其嵌入到应用程序视图,通过编程的方式设置地图的各种属性(包括当前地图显示的区域以及用户的方位),也可以使用定制标注或标准标注(例如使用测针标记)突出显示地图中的某些区域或额外的信息。

4. UIKit 框架

UIKit 框架(Uikit.framework)的 Objective-C 编程接口为实现 iOS 应用程序的图形及事件驱动提供关键基础。

1.2.2 媒体层

媒体层包含图形技术、音频技术和视频技术,这些技术相互结合就可为移动设备带来最好的多媒体体验。更重要的是,它们让创建外观音效俱佳的应用程序变得更加容易。开发者可以使用 iOS 的高级框架更快速地创建高级的图形和动画,也可以通过底层框架访问必要的工具,从而以特定方式完成某种任务。该框架中比较重要的框架如下。

1. AV Foundation 框架

iOS 2.2 引入了 AV Foundation 框架(AVFoundation.framework),该框架包含的 Objective-C 类可用于播放音频内容。通过使用该框架,开发者可以播放声音文件或播放内存中的音频数据,也可以同时播放多个声音,并对各个声音的播放特效进行控制。

2. Core Audio 框架

Core Audio 框架提供 C 语言接口,可用于操作立体声音频。通过 iOS 系统 Core Audio 框架,开发者可以在应用程序中生成、录制、混合或播放音频,也可通过该框架访问设备的震动功能(对于支持震动功能的设备)。

3. Core Graphics 框架

Core Graphics 框架(CoreGraphics.framework)包含 Quartz 2D 绘图 API 接口。Quartz 是 Mac OS X 系统使用的向量绘图引擎,它支持基于路径绘图,抗锯齿渲染,渐变,图片,颜色,坐标空间转换,PDF 文件的创建、显示和解析。虽然 API 基于 C 语言,但是它使用基于对象的抽象以表示基本绘图对象,这样可以让开发者更方便地保存并复用图像内容。

4. Core Text 框架

iOS 3.2 引入了 Core Text 框架(CoreText.framework),该框架包含一组简单高效

的 C 接口,可用于对文本进行布局以及对字体进行处理。Core Text 框架提供一个完整的文本布局引擎,开发者可以通过它管理文本在屏幕上的摆放。所管理的文本也可以使用不同的字体和渲染属性。

1.2.3 核心服务层

核心服务层为所有的应用程序提供基础系统服务。可能应用程序并不直接使用这些服务,但它们是系统许多部分赖以建构的基础。该框架中比较重要的框架如下。

1. Address Book 框架

Address Book 框架(AddressBook.framework)支持编程访问存储于用户设备中的联系人信息。如果应用程序要使用联系人信息,则可通过该框架访问并修改用户联系人数据库的记录。例如,通过使用该框架,聊天程序可以获取一个联系人列表,利用此列表初始化聊天会话,并在联系人视图显示列表中的联系人。

2. Core Data 框架

iOS 3.0 引入 Core Data 框架(CoreData.framework)。Core Data 框架是一种管理“模型-视图-控制器”应用程序数据模型的技术,适用于数据模型已经高度结构化的应用程序。利用此框架,开发者再也不需要通过编程定义数据结构,而是通过 Xcode 提供的图形工具构造一份代表数据模型的图表。在程序运行的时候,Core Data 框架就会创建并管理数据模型的实例,同时还可对外提供数据模型访问接口。

3. Core Foundation 框架

Core Foundation 框架(CoreFoundation.framework)是一组 C 语言接口,为 iOS 应用程序提供基本数据管理和服务功能。

4. Core Location 框架

Core Location 框架(CoreLocation.framework)可用于定位某个设备的当前经纬度。它可以利用设备具备的硬件,通过附近的 GPS、蜂窝基站或者 WiFi 信号等信息计算用户方位。Maps 应用程序就是利用此功能在地图上显示用户当前位置。开发者可将此技术结合到应用程序,以此向用户提供方位信息。例如,应用程序可根据用户当前位置搜索附近饭店、商店或其他设施。

5. Core Media 框架

iOS 4.0 引入了 Core Media 框架(CoreMedia.framework)。此框架提供 AV Foundation 框架使用的底层媒体类型。只有少数需要对音频或视频创建及展示进行精确控制的应用才会涉及该框架,其他大部分应用程序都用不上它。

1.2.4 核心操作系统层

核心操作系统层的底层功能是很多其他技术的构建基础。通常情况下,这些功能不会直接应用于应用程序,而是应用于其他框架,但是在直接处理安全事务或与某个外设通信的时候,必须应用到该层的框架。该框架中比较重要的框架如下。

1. Accelerate 框架

iOS 4.0 引入了 Accelerate 框架(Accelerate.framework)。该框架的接口可用于执行数学以及 DSP 运算。和开发者个人编写的库相比,该框架的优点在于它根据现存的各种 iOS 设备的硬件配置进行过优化,因此开发者只需一次编码就可确保它在所有设备都能高效运行。

2. External Accessory 框架

iOS 3.0 引入了 External Accessory 框架(ExternalAccessory.framework),通过它来支持 iOS 设备与绑定光盘通信。光盘可以通过一个 30 针的基座接口和设备相连,也可通过蓝牙连接。通过 External Accessory 框架,开发者可以获得每个外设的信息并初始化一个通信会话。通信会话初始化完成之后,可以使用设备支持的命令直接对其进行操作。

3. Security 框架

iOS 系统不但提供内建的安全功能,还提供 Security 框架(Security.framework)用于保证应用程序所管理的数据的安全。该框架提供的接口可用于管理证书、公钥、私钥以及信任策略。它支持生成加密的安全伪随机数。同时,它也支持对证书和 Keychain 密钥进行保存,是用户敏感数据的安全仓库。

CommonCrypto 接口还支持对称加密、HMAC 以及 Digests。实际上,Digests 的功能和 OpenSSL 库常用的功能兼容,只是 iOS 无法使用 OpenSSL 库。

4. System 框架

系统层包括内核环境、驱动及操作系统底层 UNIX 接口。内核以 Mach 为基础,它负责操作系统的各个方面,包括管理系统的虚拟内存、线程、文件系统、网络以及进程间通信。这一层包含的驱动是系统硬件和系统框架的接口。出于安全方面的考虑,内核和驱动只允许少数系统框架和应用程序访问。

1.3 iOS 应用开发的特点

iOS 平台上的程序拥有一些共性,这些特性会影响自身的使用体验。与这些特性相适应的程序会更加成功,与设备一起为用户提供更好的使用体验。

1. 最先响应屏幕

iOS 对屏幕反应的优先级是最高的,它的响应顺序依次为 Touch→Media→Service→Core 架构。换句话说,用户触摸屏幕之后,系统会最优先处理屏幕显示,然后才是媒体(Media)、服务(Service)以及 Core 架构。因此 iOS 应用很少出现卡顿或者延迟现象。

2. 基于 GPU 加速

iOS 系统对图形的各种特效处理基本上都是基于 GPU 硬件进行加速的,可以不用完全借助 CPU 或者程序本身,而是通过 GPU 进行渲染以达到更流畅的操控表现,使得 iOS 应用在操控过程中有着非常好的流畅性。

3. 机制效率高

iOS 采用 Objective-C 作为开发语言,编译器为 LLVM。LLVM 编译后的代码又被苹果专为 iOS 架构进行优化,运行过程中不需要虚拟机插手,执行效率非常高,并且系统不需要占用大量内存来换取执行速度,从而使得程序占用系统资源非常少,显著提高了程序运行的流畅度。

4. 硬件利用率高

iOS 平台非常封闭,所有的 App 运行对象都比较单一。由于每个应用程序都运行在 iPhone、iPad 等 iOS 产品中,所以有着很高的硬件利用效率。iOS 应用开发也因为 iOS 平台软硬件垂直整合而受益,使得开发者很容易针对 iOS 设备进行适配,从而极大地降低了开发成本和周期。

1.4 iOS 开发工具简介

打开 Mac 上的 App Store,搜索并下载 Xcode。

下载前需注册 iOS 开发者账号,登录后即可下载,大小约为 3.5GB,包括 Xcode、Interface Builder 和模拟器等工具,如图 1.2 所示。

1. Xcode

要开发 iOS 应用,需在 Mac OS X 上运行 Xcode 开发工具。Xcode 是 Apple 的开发工具套件,支持项目管理、编辑代码、构建可执行程序、代码级调试、代码的版本管理、性能调优等。这个套件的核心是 Xcode 应用本身,它提供了基本的源代码开发环境,如图 1.3 所示。

2. Interface Builder

利用 Interface Builder,可以拖曳需要的组件在程序窗口上进行装配。组件中包含标

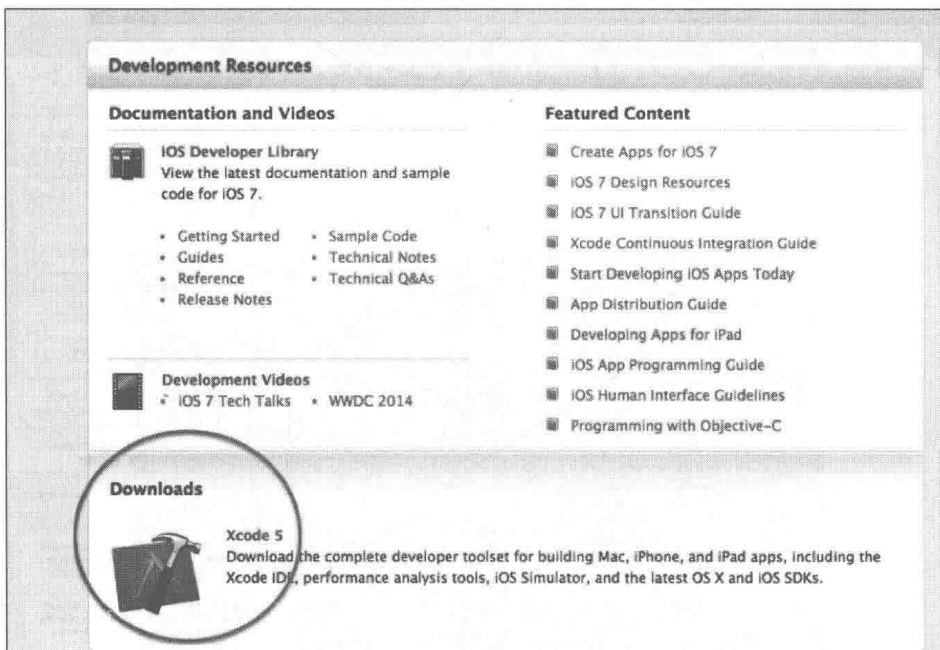


图 1.2 下载开发工具

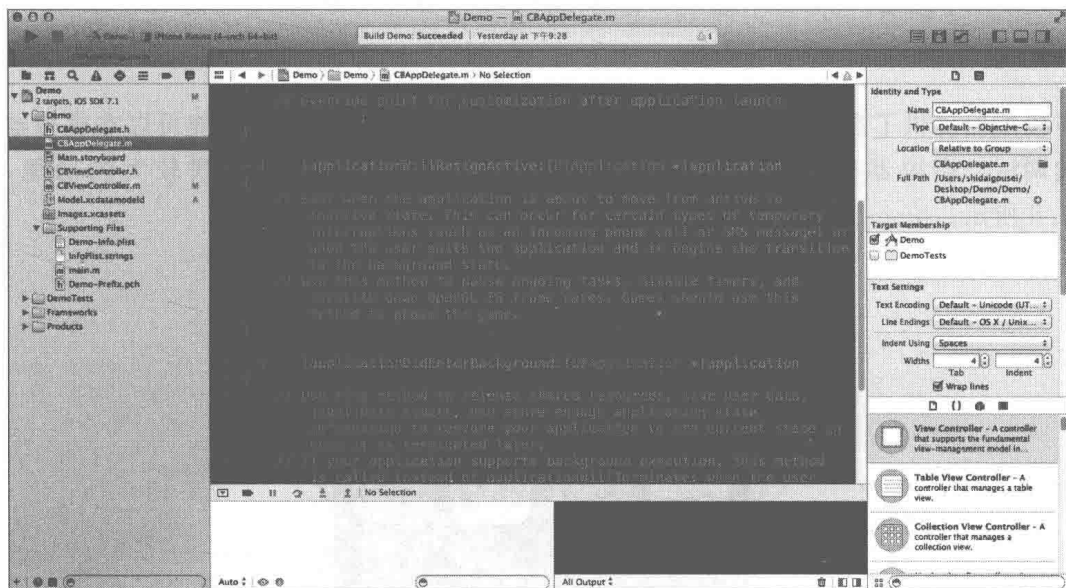


图 1.3 Xcode 界面

准的系统控件,如开关(switches)、文本框和按钮,还有定制的视图来表示程序提供的视图。在窗口表面上放置组件之后,拖曳它们可以确定位置。使用观察器(inspector)设置其属性,建立这些对象和代码之间的联系。最后将内容保存在一个 xib 文件中,这是一个自定义的资源文件格式,如图 1.4 所示。

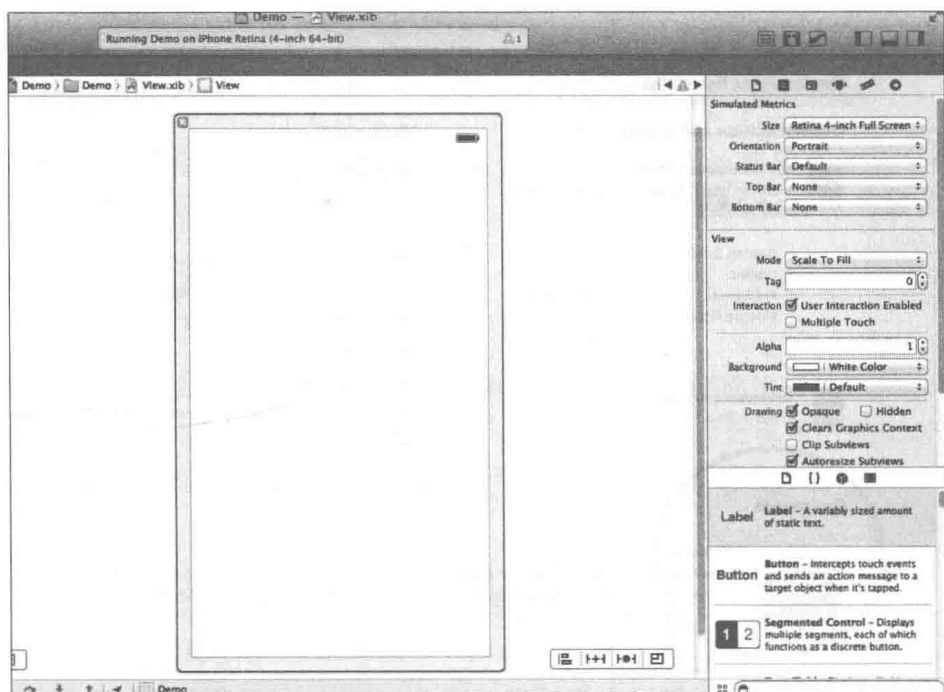


图 1.4 Interface Builder 界面

3. 模拟器

模拟器提供了在苹果电脑上开发 iOS 产品时的虚拟设备,部分功能可以在模拟器上直接调试。模拟器不支持 GPS 定位、摄像头、指南针等与硬件关联较大的功能,如图 1.5 所示。



图 1.5 模拟器