

Swift+Objective-C双语言版，提供源代码下载

赠送iOS UI开发视频教程和源代码，共36堂课，播放时长超过13小时

双语言版
移动开发
丛书



iOS 10+Xcode 8+Objective-C+Swift 3

双语言版

iOS 开发实战

从入门到上架App Store 第2版

· 张益琿 编著 ·



清华大学出版社



移动开发
丛书

iOS 开发实战

从入门到上架App Store 第2版

· 张益琿 编著 ·



清华大学出版社
北京

内 容 简 介

本书站在开发者的视角,以 iOS 10+Xcode 7 和 Objective-C+Swift 3 双语言代码示例,完整地介绍了 iOS 应用程序从开发到上架的全过程,包含 iOS 应用开发的各种热点技术和编者多年的开发经验分享。全书共 11 章,包括:开发环境搭建、界面开发、传感器技术、布局与动画技术、网络和数据技术、打包与上传发布流程等,基本上每一章都配备了实战范例,使读者可以在编写代码中学习编程,在应用开发中提高开发技能。

本书技术先进,注重实战,示例丰富,适合 iOS 开发新手使用,也很适合作为 iOS 培训机构的教材或大专院校软件开发课程的教学参考书。

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 开发实战. 从入门到上架 App Store/张益珉编著. —2 版. —北京:清华大学出版社,2017
(移动开发丛书)
ISBN 978-7-302-48356-4

I. ①i… II. ①张… 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(2017)第 218417 号

责任编辑:王金柱

封面设计:王翔

责任校对:闫秀华

责任印制:刘海龙

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

印装者:三河市铭诚印务有限公司

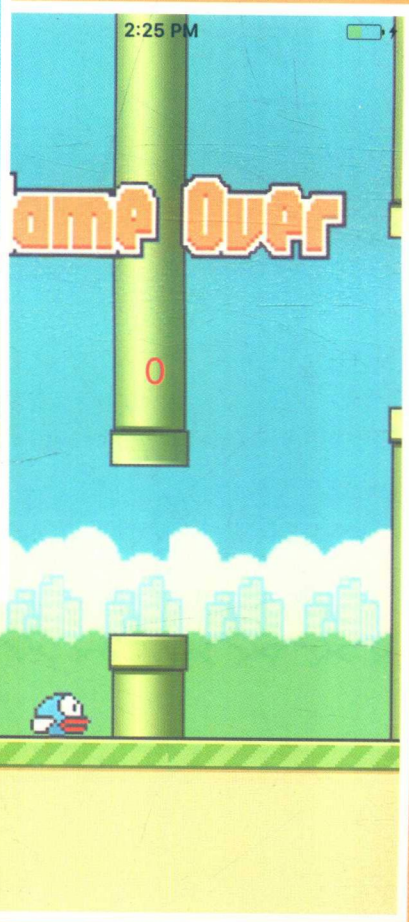
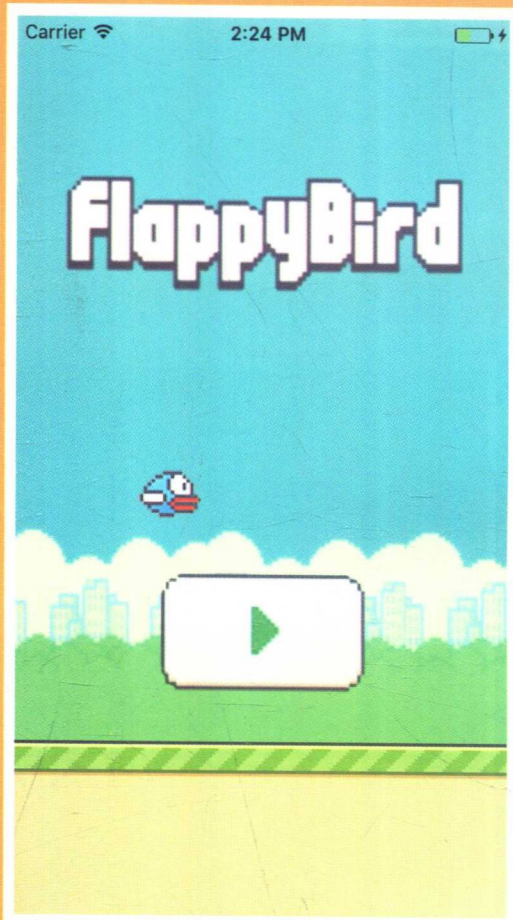
经 销:全国新华书店

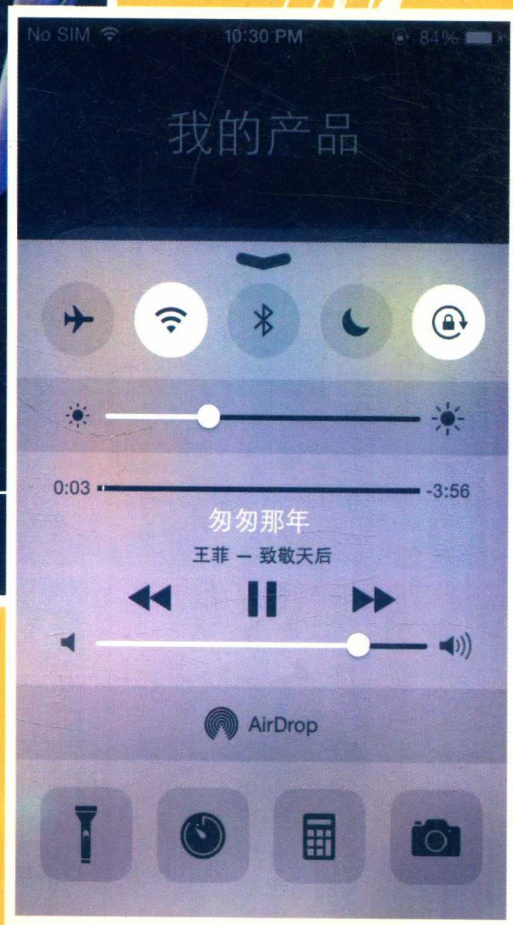
开 本:190mm×260mm 印 张:35.5 彩 插:2 字 数:909 千字

版 次:2016 年 8 月第 1 版 2017 年 10 月第 2 版 印 次:2017 年 10 月第 1 次印刷

印 数:1~3000

定 价:108.00 元

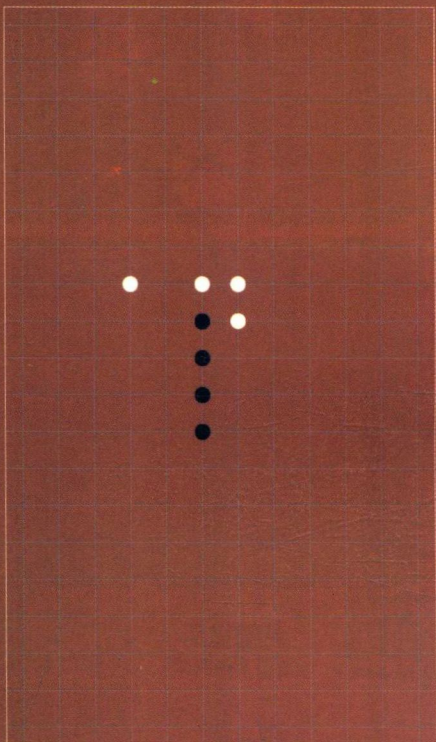




无 SIM 卡

下午 4:08
请您下子

97%

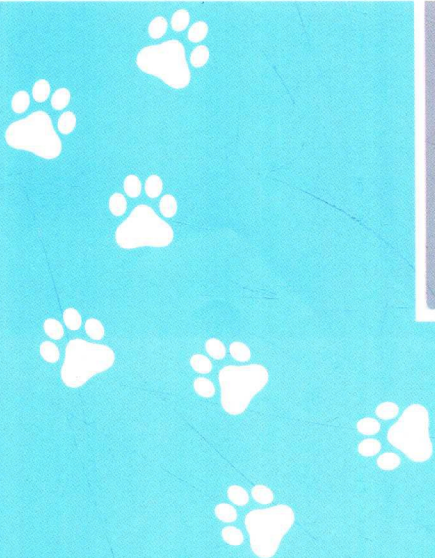
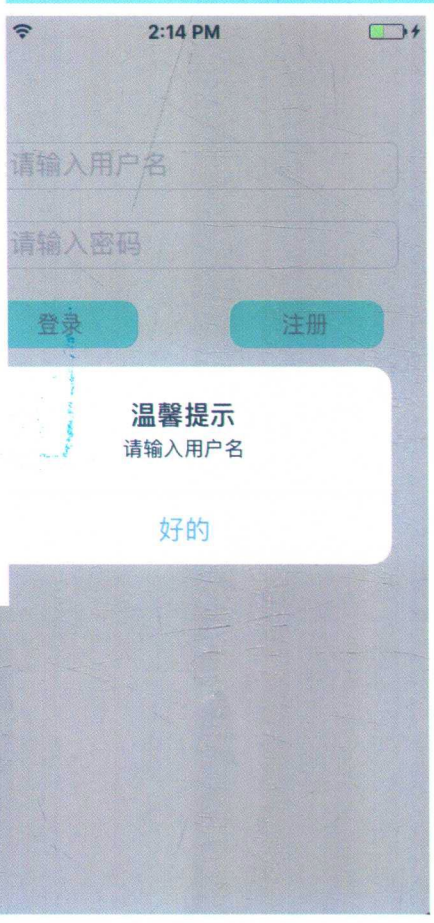
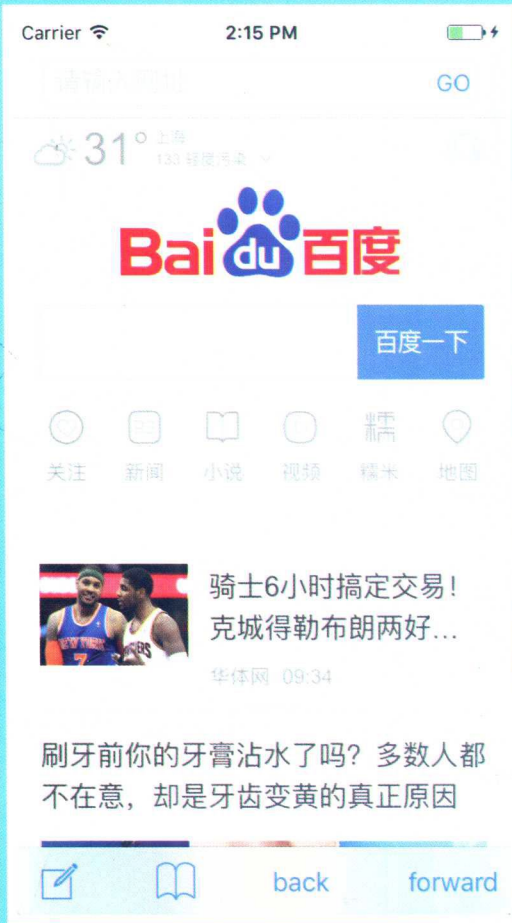


2:26 PM

创建游戏

温馨提示
请确保您的蓝牙可用

好的



前 言

编写本书的目的

当你拿到这本书时,相信你最关心的问题便是学习完这本书后是否可以完全掌握 iOS 软件开发技术。十分抱歉,对于这个问题,笔者无法给你准确的答复。编程技术日新月异,移动开发领域的新技术也层出不穷,我们都需要站在前人的肩膀上前行。但是如果你问本书是否可以帮助你入门 iOS 软件开发,笔者倒可以拍着胸脯说没问题。

开发一款完整的 iOS 软件是一个复杂的过程,开发者除了需要有编程语言的基础外,还需要对程序设计有宏观的把控。本书在编写时,定位的目标就是要帮助并无太多基础的读者快速上手 iOS 应用开发。从开发环境准备到程序界面开发,从网络与数据逻辑开发到动画与传感器技术,本书都有专门的章节进行介绍。如果说学习理论是一个枯燥的过程,那么在编程中的动手实践就是对理论学习成果的最好检验。本书中每一个模块基本都配备了实战项目,你可以通过实战项目的练习,在学习中体会独立开发软件的成就感。

本书主要内容

本书在结构上分为 11 个章节。下面介绍各个章节的主要内容及其联系。

第 1 章是为学习应用开发做准备,其中将介绍开发环境的搭建与开发工具的使用,这一章虽然为准备章节,但对初学者来说却至关重要。

第 2 章将介绍 iOS 开发中的一些基础 UI 控件,移动端应用一个很重要的特点就是要有绚丽的界面,应用程序的界面决定了用户使用这款应用程序的体验与心情,这一章向读者独立地介绍每个基础控件的用法,并通过一些综合实战来给读者提供综合使用这些控件的机会。

第 3 章在第 2 章的基础上,将向读者介绍 iOS 开发中经常使用的更多高级控件的用法,同样也会为读者提供实战机会。

第 4 章主要讲解了 iOS 应用开发中的网络编程技术,由于网络编程的演示需要有网络数据支持,很多有关网络教学的文档书籍都只讲授理论,却没有办法使读者切身地进行测试与练习。在编写本章时,特意注意了这个问题,本章除了讲授网络编程在 iOS 应用开发中的相关知识外,还将教读者如何使用网上免费的 API 服务真正做出一款网络应用。

第 5 章主要讲解 iOS 应用程序开发中的音频与视频技术,这类技术在开发音频软件和视频软件中意义重大。

第 6 章将作为动画专题,向读者介绍 iOS 应用开发中的动画技术,章节设计由简到难,并且都配有代码演示。

第7章将作为传感器专题，向读者介绍 iOS 开发中可以调用的设备传感器的相关知识。

第8章是界面布局专题，编写中参阅了很多 iOS 应用开发教材，其中都没有过多提到界面布局的相关知识，笔者认为这是一个十分大的弊端，界面布局技术是衡量一个开发者是否合格的重要指标，笔者相信读者学习 iOS 开发技术绝对不只是想简简单单地做出一个 DEMO 自己玩，做出“产品”才是读者的真正目标，而一款成熟的产品一定是具有兼容性的，并且一定是优雅的。因此，本书特别将 iOS 界面布局技术作为一个单独的章节来向读者介绍。

第9章是数据持久化专题，本章将介绍有关 iOS 应用开发中的文件操作、数据库操作的相关知识。

通过前9章的学习，你已经具备独立开发一款 iOS 应用的基础能力，但是仅仅做出产品还不够，如何让自己的产品在市场发布，使用户可以下载使用也是开发者不得不去了解、学习的内容，第10章将完整地向读者介绍提交自己的应用到 AppStore 的整个过程。

第11章是扩展章节，此章也是读者开发能力提升的一个章节，本章将介绍一些独立于前面章节，但在实际开发中也举足轻重的编程技术。

本书特点

本书的在第一版的基础上，将所有代码示例除了使用 Objective-C 语言实现外，还使用 Swift 语言实现了一遍。双语版本可以让你无论学习的是哪种语言基础，在学习本书时都不会太费力。

下载资源

本书 iOS 开发源代码及 Swift 教学视频下载地址：<http://pan.baidu.com/s/1qYhYCIK>（注意区分数字和英文字母大小写）。如果下载有问题，请发送电子邮件至 booksaga@126.com，邮件标题为“求 iOS 开发实战：从入门到上架 App Store（第2版）下载资源”。

如果你想获取更多关于 iOS 开发的视频教学资源，可以访问笔者的网络学院：<http://edu.csdn.net/lecturer/1283>。

本书 QQ 群（iOS 开发技术交流群）：203317592

致谢

最后，对于第一版中出现的差错，衷心的向读者表示歉意。本版中对读者提出的问题都一一进行了修正，在终稿前，笔者日思夜看，希望本版可以完美地出现在读者面前，然而，世上完美之事少之又少，一己之力实在不敢保证此书中再无疏漏。此书一经出版已成定局，然而人是灵活的，如果你在阅读本书时发现任何问题或有任何疑问，都可以直接联系笔者，QQ：316045346。

最后，本书得以顺利完成，全仰赖一个人的辛勤努力，他就是清华大学出版社的王金柱编辑。感谢他在笔者写作过程中的指导与鼓励。

编者
2017年7月

目 录

第 1 章 开发准备	1	2.4 按钮控件——UIButton	32
1.1 iOS 10 新特性简述	2	2.4.1 创建一个按钮改变屏幕颜色	32
1.1.1 新增触觉反馈编程接口	2	2.4.2 更加多彩的 UIButton 控件	35
1.1.2 SiriKit 框架的开放	4	2.5 文本输入框控件——UITextField	37
1.1.3 引入 Messages App	5	2.5.1 在屏幕上创建一个输入框	37
1.1.4 通知框架的整合与扩展	6	2.5.2 UITextField 的常用属性介绍	39
1.2 熟悉 iOS 开发环境	6	2.5.3 UITextField 的代理方法	40
1.2.1 安装 Xcode 开发工具	6	2.5.4 实现一个监听输入信息的用户名 输入框	41
1.2.2 了解 Xcode 开发工具主界面	8	2.6 开关控件——UISwitch	43
1.2.3 Xcode 开发工具的使用技巧及 常用快捷键	9	2.6.1 创建一个开关控件	43
1.3 创建第一个 iOS 项目	10	2.6.2 为 UISwitch 控件添加触发方法	44
1.4 使用 Git 进行项目版本管理	13	2.7 分页控制器——UIPageControl	45
1.4.1 Git 与 Github 简介	13	2.8 分段控制器——UISegmentedControl	46
1.4.2 注册 GitHub 会员	13	2.8.1 UISegmentedControl 基本属性的 应用	46
1.4.3 使用 Xcode 创建 Git 仓库	15	2.8.2 对 UISegmentedControl 中的按钮 进行增、删、改操作	48
1.4.4 用 Xcode 建立本地 Git 仓库与 GitHub 代码托管平台的关联	16	2.8.3 UISegmentedControl 中按钮宽度的 自适应	48
第 2 章 基础 UI 控件	18	2.9 滑块控件——UISlider	49
2.1 iOS 系统 UI 框架的介绍	19	2.9.1 UISlider 的创建与常规设置	49
2.1.1 MVC 设计模式	19	2.9.2 对 UISlider 添加图片修饰	51
2.1.2 代理设计模式	20	2.10 活动指示器控件—— UIActivityIndicatorView	51
2.2 视图控制器——UIViewController	20	2.11 进度条控件——UIProgressView	53
2.2.1 UIViewController 的生命周期	20	2.12 步进控制器——UIStepper	53
2.2.2 UIViewController 的视图层级 结构	27	2.12.1 步进控制器的基本属性使用	54
2.3 文本控件——UILabel	27	2.12.2 自定义 UIStepper 按钮图片	55
2.3.1 使用 UILabel 在屏幕上创建 一个标签控件	27	2.13 选择器控件——UIPickerView	56
2.3.2 自定义标签控件的相关属性	28	2.13.1 创建一个 UIPickerView 控件	56
2.3.3 多行显示的 UILabel 与换行模式	30		

2.13.2	UIPickerView 选中数据时的 回调代理	58	3.4.2	通过网络请求加载 UIWebView	102
2.14	通过 CALayer 对视图进行修饰	59	3.4.3	通过 HTML 字符串加载 UIWebView	103
2.14.1	创建圆角的控件	59	3.4.4	通过 NSData 数据加载 UIWebView	104
2.14.2	创建带边框的控件	60	3.4.5	UIWebView 中常用方法解析	104
2.14.3	为控件添加阴影效果	60	3.4.6	UIWebView 的代理方法	106
2.15	警告控制器——UIAlertController	61	3.5	表格视图——UITableView	107
2.15.1	UIAlertController 的警告框	61	3.5.1	UITableView 的创建与复用 机制	107
2.15.2	UIAlertController 之活动列表	64	3.5.2	创建一个表格视图 UITableView	108
2.16	扩展篇	65	3.5.3	关于表格数据的载体 UITableViewCell	111
2.16.1	搜索栏控件——UISearchBar	65	3.5.4	设置 UITableView 的行高和 头尾视图	113
2.16.2	日期时间选择器—— UIDatePicker	69	3.5.5	UITableView 的用户交互行为	115
2.16.3	警告视图——UIAlertView	71	3.5.6	为 UITableView 添加索引栏	118
2.16.4	活动列表——UIActionSheet	72	3.6	复杂布局视图——UICollectionView	119
2.17	实战：登录注册界面的搭建	73	3.6.1	UICollectionView 控件的优势与 布局方式	119
第 3 章	高级 UI 控件	80	3.6.2	使用 UICollectionView 进行 九宫格式的布局	119
3.1	导航控制器——UINavigationController	81	3.6.3	创建更加灵活的流式布局	122
3.1.1	导航控制器的工作原理	81	3.6.4	自定义 UICollectionViewFlowLayout 进行参差瀑布流布局	123
3.1.2	使用导航控制器进行多界面 搭建	82	3.6.5	使用 UICollectionView 进行圆环 布局	128
3.1.3	导航栏 UINavigationController	85	3.7	实战：开发一款手机网页浏览器	133
3.1.4	导航按钮 UIBarButtonItem	87	3.7.1	网页浏览器工程的搭建	133
3.1.5	导航控制器的工具栏	90	3.7.2	核心网页视图的设计	135
3.1.6	iOS 8 之后导航控制器的一些 有趣功能	91	3.7.3	历史记录界面的设计	147
3.2	标签控制器——UITabBarController	92	3.7.4	收藏界面的设计	151
3.2.1	标签控制器的工作原理	92	3.7.5	启动页面、图标及应用名称的 相关优化	154
3.2.2	标签控制器的基础用法解析	92			
3.2.3	关于 UITabBarItem 的使用	94			
3.3	滚动视图——UIScrollView	96			
3.3.1	使用 UIScrollView 展示视图 内容	96			
3.3.2	UIScrollView 的代理方法	98			
3.4	网络视图——UIWebView	101			
3.4.1	App 网络传输安全策略	101			

第 4 章 网络编程	157	第 5 章 音频、视频开发	216
4.1 使用 NSURLConnection 请求网络数据	158	5.1 iOS 音频开发基础——AVAudioPlayer 类的使用	217
4.1.1 申请一个免费的 API 服务	158	5.1.1 使用 AVAudioPlayer 进行 MP3 音频文件的播放	217
4.1.2 使用 NSURLConnection 进行 API 服务数据的获取	161	5.1.2 进行音频播放相关属性的控制	218
4.1.3 使用 NSURLConnection 进行异步网络请求	162	5.1.3 后台播放音频及用户交互的优化	225
4.1.4 使用 NSURLConnection 类通过代理回调的方式异步进行网络请求	164	5.2 iOS 视频开发基础	229
4.2 设计封装一个更加易用的网络请求类	165	5.2.1 使用 MPMoviePlayerController 向应用中嵌入视频模块	230
4.2.1 设计自定义的网络请求连接类	166	5.2.2 MPMoviePlayerController 常用属性与方法解析	231
4.2.2 设计自定义的网络请求管理类	167	5.3 视频播放器视图控制器——MPMoviePlayerViewController	236
4.3 JSON 类型数据的解析与数据模型的设计	171	5.4 AVPlayerViewController 视频播放框架与画中画开发技术	238
4.3.1 JSON 数据简介	171	5.4.1 使用 AVPlayerViewController 进行视频播放	238
4.3.2 在 iOS 中解析 JSON 数据	173	5.4.2 iPad 的画中画播放技术	241
4.3.3 数据模型 Model 类的设计	175	5.5 实战：“歌手王菲”音频播放器的开发	244
4.4 使用 CocoaPods 进行第三方库的管理	180	5.5.1 工程搭建与 LRC 歌词文件简介	244
4.4.1 在 MAC 上安装 CocoaPods	180	5.5.2 LRC 歌词解析引擎的设计	245
4.4.2 用 CocoaPods 搭建一个使用第三方网络请求框架 AFNetworking 的工程	182	5.5.3 核心播放器引擎的设计	253
4.5 使用 AFNetworking 进行网络请求	184	5.5.4 歌曲列表与歌词显示视图界面的设计	265
4.5.1 详解 HTTP/HTTPS 协议	184	5.5.5 播放器主页面的实现	274
4.5.2 使用 AFNetworking 进行网络请求	185	5.5.6 后台播放音频用户交互的处理	285
4.6 实战：开发“笑一笑”应用程序	188	第 6 章 动画开发	288
4.6.1 工程项目框架的搭建	188	6.1 使用 UIImageView 播放图片组帧动画	289
4.6.2 “笑一笑”界面数据载体 cell 的设计	190	6.2 UIView 层动画的应用	290
4.6.3 “笑一笑”界面的搭建	193	6.2.1 执行 UIView 层过渡动画的 3 个类方法	290
4.6.4 实现下拉刷新与加载更多功能	200		
4.6.5 “趣图吧”界面数据载体 cell 的设计	203		
4.6.6 “趣图吧”界面的设计	207		

- 6.2.2 创建 UIView 层的阻尼动画……293
 - 6.2.3 动画参数配置与组合动画……293
 - 6.2.4 UIView 层过渡动画支持的属性……297
 - 6.3 使用 commit 方式进行 UIView 层动画的创建……298
 - 6.3.1 使用 commit 方式进行 UIView 层过渡动画的创建……298
 - 6.3.2 两种 UIView 层动画创建方式的优劣……300
 - 6.4 UIView 的转场动画……300
 - 6.4.1 重绘 UIView 视图时使用的转场动画……300
 - 6.4.2 切换 UIView 视图时使用的转场动画……302
 - 6.5 核心动画编程技术——CoreAnimation ……303
 - 6.5.1 锚点对视图控件几何位置的影响……303
 - 6.5.2 色彩梯度层——CAGradientLayer ……304
 - 6.5.3 视图拷贝层——CAReplicatorLayer ……305
 - 6.5.4 图形渲染层——CAShapeLayer ……307
 - 6.5.5 文本绘制层——CATextLayer ……309
 - 6.5.6 CAAnimation 动画体系介绍……310
 - 6.5.7 使用 CABasicAnimation 创建基础动画……312
 - 6.5.8 使用 CAKeyframeAnimation 类创建关键帧动画……316
 - 6.5.9 CALayer 层的转场动画——CATransition ……317
 - 6.5.10 CALayer 层的组合动画——CAAnimationGroup ……320
 - 6.5.11 CATransform3D 变换的应用……321
 - 6.6 炫酷的粒子效果……324
 - 6.6.1 粒子发射器——CAEmitterLayer ……324
 - 6.6.2 粒子单元——CAEmitterCell ……327
 - 6.6.3 创建粒子火焰动画……330
 - 6.7 播放 GIF 动态图……332
 - 6.7.1 使用 UIWebView 进行 GIF 动态图播放……332
 - 6.7.2 使用 UIImageView 帧动画进行 GIF 动态图播放……334
 - 6.8 实战：小游戏 Flappy Bird 的设计与开发……336
 - 6.8.1 小鸟对象的设计……336
 - 6.8.2 游戏开始界面的设计……341
 - 6.8.3 游戏结束界面的设计……344
 - 6.8.4 Flappy Bird 游戏主框架的搭建……347
- ## 第 7 章 传感器开发……360
- 7.1 为应用程序添加手机密码及指纹识别的安全验证……361
 - 7.1.1 使用手机密码为应用程序添加安全验证……361
 - 7.1.2 使用用户指纹为应用程序添加安全验证……363
 - 7.2 使用加速度传感器、螺旋仪传感器与磁力传感器获取设备空间状态……364
 - 7.2.1 使用 UIAccelerometer 获取设备空间状态……364
 - 7.2.2 使用 CoreMotion 框架获取设备空间状态信息……365
 - 7.3 距离传感器的应用……371
 - 7.4 iOS 蓝牙开发技术……373
 - 7.4.1 中心设备管理类——CBCentralManager ……374
 - 7.4.2 外围设备管理类——CBPeripheralManager ……382
 - 7.5 GPS 应用与地图编程技术……388
 - 7.5.1 进行设备地理位置定位……388
 - 7.5.2 原生地图开发技术……392
 - 7.5.3 在地图中添加大头针及标注……395
 - 7.5.4 在地图视图中添加覆盖物……398

7.5.5 在地图中进行线路导航与附近 兴趣点检索	402	9.2 使用归档技术进行数据模型持久化	489
7.6 实战: 简易蓝牙对战五子棋	410	9.2.1 进行单一系统数据类型的归档与 解归档操作	489
7.6.1 游戏核心通信类的设计	410	9.2.2 对多个对象进行数据归档	491
7.6.2 棋盘瓦片的设计	427	9.2.3 进行自定义数据模型的归档	492
7.6.3 核心游戏视图与游戏核心逻辑的 设计	430	9.3 小型数据库 SQLite 在 iOS 开发中的 应用	495
7.6.4 核心游戏视图控制器的设计	448	9.3.1 SQLite 数据库常用语法介绍	495
第 8 章 界面布局	454	9.3.2 使用 iOS 原生框架 sqlite3 对 SQLite 数据库进行操作	497
8.1 iOS 中传统的 UIViewAutoresizing 布局 模式	455	9.4 核心数据管理框架 CoreData 的使用	501
8.1.1 通过代码设置视图控件的 UIViewAutoresizing 模式	455	9.4.1 使用 CoreData 设计数据模型	502
8.1.2 在 xib 文件中可视化地配置 控件的 autoresizing 属性	457	9.4.2 CoreData 编程框架中 3 个 重要的类	504
8.2 autolayout 自动布局框架	459	9.4.3 CoreData 编程框架的数据操作	510
8.2.1 初识 autolayout	459	9.4.4 使用 CoreData 进行数据与页面的 绑定	516
8.2.2 autolayout 的属性意义与一个 简单的自动布局示例	461	9.5 网络缓存策略	524
8.2.3 使用 Objective-C 风格的方法 进行代码 autolayout 布局	464	9.5.1 为网络请求设置缓存策略	524
8.2.4 使用格式化的字符串进行 autolayout 布局对象的创建	467	9.5.2 应用缓存管理类 NSURLCache 简介	525
8.2.5 与约束相关的几个方法	470	第 10 章 提交应用程序到 AppStore	527
8.2.6 使用 autolayout 设计一个高度自适应 的聊天输入框和动画优化	470	10.1 使用 Xcode 开发工具进行程序调试	528
8.2.7 使用第三方库 Masonry 进行 autolayout 的约束布局	474	10.1.1 使用自定义断点进行代码 调试	528
第 9 章 数据持久化	482	10.1.2 添加全局异常断点	529
9.1 使用 plist 文件进行轻量级数据持久化 管理	483	10.1.3 使用 LLDB 调试器进行程序 调试	530
9.1.1 在工程中读取 plist 文件数据	483	10.2 Apple 开发者账号的申请	531
9.1.2 在程序沙盒 Documents 目录中 创建和使用 plist 文件	485	10.2.1 几种类型的开发者账号	531
9.1.3 使用 NSUserDefaults 类进行 数据持久化	486	10.2.2 申请开发者账号的过程	531
		10.3 进行应用程序的打包	534
		10.3.1 在 iTunes Connect 中进行应用的 创建与配置	534
		10.3.2 使用 Xcode 打包与提交 iTunes	541

第 11 章 更多功能与进阶技巧	545	11.2.2 通知中心 NSNotificationCenter 的 应用	549
11.1 Objective-C 中 block 语法的应用	546	11.3 多线程开发技术	550
11.1.1 声明与实现 block 语法块	546	11.3.1 使用 NSThread 进行线程 管理	550
11.1.2 block 代码块中访问对象的微妙 关系	547	11.3.2 使用 NSOperation 类与 NSOperationQueue 类进行多任务 管理	552
11.2 iOS 通知中心 NSNotificationCenter 的 应用	548	11.3.3 iOS 中 GCD 编程技术简介	556
11.2.1 通知类 NSNotification 简介	549		

第 1 章

开发准备

工欲善其事，必先利其器。在学习 iOS 移动开发之前，首先应该将开发环境配置完成并且对所需要使用的开发工具进行了解与熟悉。本章首先向读者介绍 iOS 10 系统相比之前系统的一些新特性，使读者对目前主流的 iOS 系统在宏观上有一定了解。后面将一步步演示开发环境的搭建，并向读者介绍开发工具 Xcode 的常用功能。

通过本章的学习，读者能够掌握：

1. 了解 iOS 10 的新特性和新功能。
2. 申请免费的 Apple ID 账号。
3. 使用 Xcode 开发工具创建 iOS 工程。
4. 使用 Xcode 开发工具编写与调试程序。
5. 熟悉 Xcode 工程结构。
6. 编写第一个程序 HelloWorld。
7. 使用 Git 工具进行版本管理。
8. 使用 GitHub 代码托管平台。

1.1 iOS 10 新特性简述

iOS 10 是 iOS 系统的一次重大升级，提供了更加人性化的触觉用户体验，也为开发者提供了接口，开发者可以使用接口使自己应用程序的用户体验更加优秀。iOS 10 系统首次向开发者开放了 SiriKit 框架，借助 Apple 的 Siri 服务，开发者可以开发出更加优秀的人机交互应用。除此之外，iOS 10 中的 Message App 和 CallKit 框架的开放、UserNotification 框架的引入等使得开发者可以更加灵活地开发新一代的 iOS 应用。熟悉 iOS 10 的这些新特性可以更好地帮助读者学习 iOS 应用程序的开发。

1.1.1 新增触觉反馈编程接口

在 iPhone 6s 发行之后，很多果粉都争先体验 3D Touch 给用户带来的额外维度上的交互，随着 iPhone 7 与 iPhone 7s 的问世，Apple 进一步深化了用户交互上的体验。iPhone 7 与 iPhone 7s 配合 iOS 10 以上的系统可以给用户带来视觉与触觉上的双重体验。在 iOS 10 以上的系统中新引入了 3 个类，分别为 UIImpactFeedbackGenerator 类、UINotificationFeedbackGenerator 类与 UISelectionFeedbackGenerator 类。当用户触发某些交互事件时，开发者可以通过这些类为用户提供额外的触觉体验，请看下面的示例代码。

Swift 语言版本：

```
//用户触发某些事件后执行
func feedbackStart(style:Int) {
    switch style {
    case 1:
        //进行强烈触觉反馈
        let imp = UIImpactFeedbackGenerator(style: .heavy)
        imp.impactOccurred()
    case 2:
        //进行正常触觉反馈
        let imp = UIImpactFeedbackGenerator(style: .light)
        imp.impactOccurred()
    case 3:
        //进行轻量触觉反馈
        let imp = UIImpactFeedbackGenerator(style: .medium)
        imp.impactOccurred()
    case 4:
        //适用于通知的反馈
        let imp = UINotificationFeedbackGenerator()
        //错误类型通知时的反馈
        imp.notificationOccurred(.error)
    case 5:
```