



OSUI 开发捷径

利用Interface Builder高效、优雅地开发UI

郭晓亮◎著



中国工信出版集团



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

郭晓亮◎著

利用Interface Builder高效、优雅地开发UI

iOS UI 开发捷径



电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

这是一本详细讲解如何使用 Interface Builder 的书籍,主要讲解如何利用 Interface Builder 快速、高效地开发 UI。本书从 Interface Builder 菜单、UIKit 控件这些基础知识到如何使用 Interface Builder 鲜为人知的高级功能、如何利用 Interface Builder 进行代码的优化等都进行了详细讲解,内容覆盖了利用 Interface Builder 开发 UI 所涉及的各个知识点。除此之外,本书还着重介绍了经常与 Interface Builder 配合使用的 Auto Layout,以及很多与 UI 学习甚至与 iOS 开发学习相关的知识。

本书理论与实践相结合,理论部分全面而且深入,让读者知其所以然,实践部分不仅讲到了在开发中经常遇到的情况,而且也关注了一些非常重要的细节。本书适合想快速、高效地开发 UI,想学习 Interface Builder 的各个层次的开发人员阅读,对于 iOS 的 UI 学习来说,本书也具有很强的指导意义。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,侵权必究。

图书在版编目(CIP)数据

iOS UI 开发捷径:利用 Interface Builder 高效、优雅地开发 UI / 郭晓亮著. —北京:电子工业出版社, 2017.8

ISBN 978-7-121-31721-7

I. ①i…II. ①郭…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 121092 号

策划编辑:符隆美

责任编辑:张 玲

印 刷:三河市鑫金马印装有限公司

装 订:三河市鑫金马印装有限公司

出版发行:电子工业出版社

北京市海淀区万寿路 173 信箱

邮编:100036

开 本:787×980 1/16 印张:18

字数:390 千字

版 次:2017 年 8 月第 1 版

印 次:2017 年 8 月第 1 次印刷

定 价:69.00 元

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话:(010) 88254888, 88258888。

质量投诉请发邮件至 zlts@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

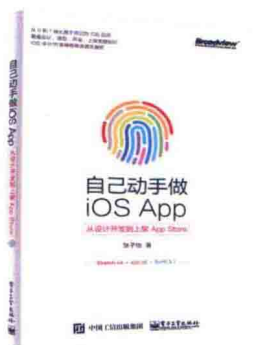
本书咨询联系方式:(010) 51260888-819, faq@phei.com.cn。

作者简介

郭晓亮

一直奋斗在iOS开发一线的大龄“码农”，从iOS 4开始就与Xcode“纠缠”，像大多数人一样，默默地享受着技术带来的快乐和成就感；像大多数人一样，没有干过“惊天动地”的项目，但岁月磨灭不了一颗热爱技术的心。

好书分享



前言

iOS 系统在移动互联网如火如荼、方兴未艾的发展中有着举足轻重的地位，iOS 开发也成为近几年比较热门的行业。UI 开发是 iOS 开发中重要的一部分，不仅负责信息的展示，还可以与用户交互，直接体现出应用程序的“颜值”，影响着用户体验，所以说 UI 开发相当重要。对于已开发的应用程序，用户并不关心开发人员用了哪些技术和解决方案，只关心页面做得是否好看，交互操作是否顺手，滑动时会不会卡，这些都与 UI 开发息息相关。在实际的开发工作中，UI 开发往往占据了大部分的时间和精力。而我们所追求的是快速开发 UI，把主要精力放在业务逻辑上，这也是苹果公司所希望的，因此，苹果公司将 Interface Builder 作为 UI 开发的“大杀器”，并在逐步地改良。这本书讲的就是如何利用 Interface Builder 高效、优雅地开发 UI。

写作的动机

从技术角度来讲，UI 开发主要有两种方式：

1. 利用代码开发。
2. 利用可视化开发工具——Interface Builder 开发。

在实际的工作中，大部分情况是将两种方式结合使用。但是在国内，Interface Builder 的应用相对于国外来说并不是特别广泛，使用的场景相对简单，使用的技术也比较浅，并没有展示出 Interface Builder 强大的功能和应有的价值。Interface Builder 在国内是一项被低估的技术。在 2015 年苹果全球开发者大会（WWDC）的“*What's New in Storyboard*”会场中，苹果公司的演讲嘉宾现场统计了使用 Interface Builder 开发者的人数，从他的言语中可以推测，当时国外开发者中使用 Interface Builder 的比例已经相当高了。从侧面可以看出，在 iOS 开发上，我们和国外的开发者还有一定的差距。

随着 4.0 英寸、4.7 英寸、5.5 英寸 iPhone 手机的出现，适配成了 UI 开发中很烦琐的一件事情，对于要适配横屏、竖屏或 iPhone 与 iPad 通用的应用程序尤为如此。利用 Interface Builder + Auto Layout + Trait Variations 可以十分简单、高效地解决这个问题。苹果公司每个新版本的 Xcode 都会对 Interface Builder 进行相应的功能升级，而且从苹果公司每年在 WWDC 上推出的新技术来看，很多是与 Interface Builder 相关的。这也说明，使用 Interface Builder 是 UI 开发的大势所趋。

在程序员看来，没有什么自动化是一个脚本解决不了的，如果有，那就是再写一个脚本。但通常来说，程序员都喜欢追求效率，而 Interface Builder 对开发效率的提升无疑是巨大的。

笔者希望通过这本书能够让国内的开发者重新认识 Interface Builder，对 Interface Builder 产生浓厚的兴趣，最后达到精通 Interface Builder 的目的。同时希望开发者能将其更多地应用到实际开发中，提高自己的工作效率，从而更加从容地面对互联网行业紧张、快速的节奏。笔者不才，如果本书对你来说没能达到期望的效果，那么只希望它能对你学习 Interface Builder 起到一个抛砖引玉的作用。

本书的特点

本书的特点如下。

1. 细致。本书每个知识点的讲解都力争做到细致，让 Interface Builder 零基础的 iOS 开发者可以在入门时更加从容。如果你是有一定 Interface Builder 基础的开发者，那么可以略过一些自认为琐碎的细节，以免浪费宝贵的时间，但是如果你有兴趣了解这些细节的话，也一定会有所收获，它会让你更加系统、更加深刻地掌握 Interface Builder。
2. 全面。本书的讲解也力争做到全面，让读者可以掌握 Interface Builder 开发中的各种技巧，了解各个细节，并能够系统地掌握 Interface Builder 相关的原理。由于 Interface Builder 相关知识碎片化严重，体系化不明显，如果只学习在开发过程中的应用技能，不深入理解一些原理，那么在一段时间不用之后很容易遗忘，只有掌握了相关的原理，理论配上实践，才能全面地掌握它。
3. 深入。本书在介绍 Interface Builder 的基础知识和基本使用之后，加入了大量的进阶内容，包括有些章节中的“深入学习”和“Interface Builder 进阶”。这些内容绝对能在 Interface Builder 开发中起到至关重要的作用，相信会改变你对 Interface Builder 原有的认识，虽然这里介绍的很多 Interface Builder 的应用技巧并没有在开发中得到普遍使用，但是它们的确很方便、实用。如果能掌握这些深入的知识和技巧，那么在利用 Interface Builder 开发时一定会变得更高效、更优雅。

4. 拓展。与此同时，本书加入了一些 iOS 开发中的其他知识点，一部分穿插在整本书中，另一部分则以“拓展学习”章节的形式出现，用于配合 Interface Builder 的讲解。Interface Builder 开发与 iOS 其他方面的开发并不是孤立的，实际上学习 Interface Builder 时，需要具备很多 iOS 其他方面的知识，这些知识可以让读者学习 Interface Builder 时更轻松，更便于理解，同时也能学习得更深入。拓展部分的内容包括 iOS 开发中的一些基本原理和应用细节，例如 App 启动流程分析、UIApplication 自定义、iOS 开发中的内存管理、SDK 开发应该掌握的一些技巧等，还包括其他方面的一些内容，例如 CocoaPods 和“黑魔法”——如何实现在低版本中使用高版本中才存在的类等，这些内容对大家学习 Interface Builder 之外的 iOS 开发知识非常有帮助。
5. 实用。本书还会介绍一些细节，这些细节虽然很小，但是在开发中却很重要，也很实用。其中很多都是笔者自己踩过的“坑”，本书介绍的这些细节可以帮助读者避开这些“坑”，也可以从笔者个人的经验中学到一些东西。本书的最后还会介绍一些 Interface Builder 开发中的小技巧 and 笔者在使用 Interface Builder 时遇到的一些 bug，相信这些实用的内容会给读者带来帮助。

内容的组织

本书一共分为 7 章。

第 1 章主要介绍 Interface Builder、xib、Bundle、sb、nib、storyboardc 等内容，理解这些内容是系统学习 Interface Builder 的基础。这些内容决定了理解和掌握 Interface Builder 的深度，所以非常重要。除此之外，第 1 章还介绍了 Interface Builder 的优缺点及其学习的特点，了解优缺点便于大家更好地使用它，同时也有利于项目中 UI 开发的技术选型，便于更快速地掌握它。本章最后介绍了 iOS 开发中的依赖管理工具——CocoaPods。

第 2 章首先讲解如何有效地解决 Interface Builder 文件的冲突。文件的易冲突是很多开发者拒绝使用 Interface Builder 的一个重要理由，所以本章解决了这些开发者的担忧。然后介绍如何创建并使用各种 Interface Builder 文件，由于在不同情况下创建和使用方式不同，所以有些细节需要特别注意。本章还讲解了 Interface Builder 文件与源文件的“关联”、与源文件建立 @IBOutlet 和 @IBAction “连线”等一些基本而又重要的操作，这些操作在平时开发中经常被使用。另外还会介绍 Bundle 的创建，这对开发 SDK 很有帮助。除此之外，还介绍了 App 启动过程的分析和自定义 UIApplication，它可以帮助大家理解 App 启动后是如何与代码关联的。

第3章对 xib 进行了全面的讲解,包括 Utilities 中的每个检查器菜单、各个菜单中的重要标签、控件栏中的重要控件、Show the Object library 菜单(主要讲解了在 xib 中使用手势),以及在 xib 中使用 tableView、tableViewCell、Asset Catalog 等内容。这些都是 Interface Builder 开发的基础,是必须掌握的内容。

第4章将介绍 Auto Layout 在 Interface Builder 中的使用。Interface Builder 的最大优势在于可以快速地开发 UI,减少大量的“胶水代码”,其次就是利用 Auto Layout 可以高效、优雅地做适配,所以本章将详细地介绍 Auto Layout 在 Interface Builder 中的使用,讲解在不写代码的情况下,如何利用 Interface Builder + Auto Layout 满足产品经理刁钻的 UI 需求。本章首先会详细介绍 Auto Layout 相关的基础知识,包括 Auto Layout 的理解、如何正确设置约束等,然后介绍 Interface Builder 中 Auto Layout 的菜单,及其重要的标签,最后讲解 Interface Builder 中使用 Auto Layout 进行布局的实践。除此之外,还会介绍如何实现在低版本中使用高版本中才存在的类等拓展内容。

第5章全面讲解 storyboard,介绍 xib 中没有的一些菜单和功能,其中重点讲解 segue,包括常规的 segue、自定义的 segue、Embed Segue、Unwind Segue 等,还有一部分重要的内容是在 storyboard 中使用 tableView(包括静态 tableView)和 tableViewCell,这些内容都可以在 xib 的基础上再次提高开发效率,是 storyboard 的优势所在。

第6章是 Interface Builder 的进阶章节,主要包括用 Trait Variations 做适配、IB 文件的加载过程、App 的本地化、Storyboard Reference 和 RBStoryboardLink 的使用、IB 中 Object 与 External Object 的使用、IB 中关键字的总结等内容,这部分都是在平时开发中用得比较少,但很实用的内容,有一些内容比较难理解,需要上面几章的基础知识做铺垫。

第7章介绍在 Interface Builder 开发中的一些技巧,帮助大家提高开发效率,并讲解笔者在 Interface Builder 开发中遇到的一些 bug。

适合的读者

本书适合以下几类读者。

1. iOS 入门开发者。

使用 Interface Builder 是 iOS 入门最简单的方式,它对快速学习并掌握 UIKit 有事半功倍的作用,同时 Interface Builder 也是 UI 开发的趋势所在。本书并不会介绍 Objective-C、Swift 等开发语言,也不会介绍 Foundation、UIKit 等基础类库及 iOS 基础知识,它不是那种“从入门到精通”的工具书,但是可以配合这本书,帮助开发者入门 iOS。

2. 不熟悉 Interface Builder，在开发中也很少使用 Interface Builder 的 iOS 开发者。

本书介绍的基础知识可以让读者应对大部分与 Interface Builder 打交道的情况。可能有些开发者在开发中并不使用 Interface Builder，但是如果开发团队中有人在使用 Interface Builder 的话，有这方面的知识会让团队合作更顺畅，也更容易熟悉工程中其他人写的代码。除此之外，现在在网上下载一个 Demo，几乎都是基于 Interface Builder 的，苹果公司官方的 Demo 更是这样，熟悉 Interface Builder 是适应开发大环境的需要。

3. 想在 Interface Builder 上进阶的开发者。

很多人认为，Interface Builder 很简单，很容易上手，根本没有可以深入学习的内容。其实，这是一种不正确的想法。Interface Builder 的确很简单，很容易上手，但同时 Interface Builder 有很多功能并不被大多数开发者所知所用，例如，如何用 Interface Builder 简单地进行本地化？如何用 Interface Builder 优雅地实现 VC 间的回调传值？如何在 Interface Builder 菜单栏中显示一个类自定义的属性？甚至还有一些灵活的用法可以改变工程中类之间的依赖结构，一些高级的用法还可以优化工程架构，使架构更简单、耦合性更低。Interface Builder 绝不仅仅停留在 UI 上，这里有很多内容值得大家去探索。

本书的约定

本书用 UI 特指 User Interface——用户界面，包括与用户的交互行为，在某些语境下也指代 UI 设计师；用 PM 指代产品经理；用 UE 指代交互设计师。本书中提到的 App 特指在 iOS 系统上运行的一个应用程序——Application；提到的苹果特指苹果公司。

在接下来的学习中，会用 IB 指代 Interface Builder；用 sb 指代 storyboard。storyboard 是一种文件的格式，可以理解为.storyboard 格式的文件。所以，本书提到的 sb 就是指.storyboard 格式的文件。同样，xib 就是指.xib 格式的文件。本书提到的 IB 文件就是指.xib 文件和.storyboard 文件。在本书中，用 VC 或 viewController 代表一个 UIViewController 子类的实例，在某些语境下也用页面代表 VC，用 view 代表一个 UIView 类或其子类的实例。在 Interface Builder 中，通常用 View Controller、Button 和 Label 等指代控件区域中相应的模板，用 viewController、button 和 label 指代用这些模板生成的实例对象。本书中的 OC 指代 Objective-C 开发语言；crash 指代程序运行出错。

本书使用的 IDE 为 Xcode 8.0，使用的编程语言为 Swift 3.0。在某些情况下，如果 IB 的使用在 Swift 和 Objective-C 上有不同，那么会将两者不同的用法都介绍一下。本书提到的源文件指 Swift 语言中的.swift 文件，或者 Objective-C 语言中的.h、.m 文件，例如 Person 类的源文件，就是指 Person.swift 文件，或者 Person.h、Person.m 文件。

读者服务

轻松注册成为博文视点社区用户（www.broadview.com.cn），扫码直达本书页面。

- **提交勘误：**您对书中内容的修改意见可在 [提交勘误](#) 处提交，若被采纳，将获赠博文视点社区积分（在您购买电子书时，积分可用来抵扣相应金额）。
- **交流互动：**在页面下方 [读者评论](#) 处留下您的疑问或观点，与我们和其他读者一同学习交流。

页面入口：<http://www.broadview.com.cn/31721>



目录

第 1 章	Interface Builder 概要	1
1.1	Interface Builder 简介	1
1.1.1	Interface Builder 是什么	1
1.1.2	xib 是什么	2
1.1.3	拓展学习: Bundle 概念的理解	4
1.1.4	理解 Bundle 对于 IB 的指导意义	5
1.1.5	storyboard 是什么	5
1.1.6	nib 是什么	6
1.1.7	storyboardc 是什么	8
1.2	Interface Builder 的优点	9
1.2.1	开发和维护效率高	9
1.2.2	减少大量的 UI 代码和“胶水代码”	10
1.2.3	适配变得十分简单	10
1.2.4	IB 也可以做一些非 UI 的事情	11
1.2.5	利用 IB 学习控件可以达到事半功倍的效果	11
1.3	Interface Builder 的缺点	11
1.3.1	IB 的执行效率没有纯代码高	12
1.3.2	使用 IB 开发的过程中容易出现一些小问题	12
1.3.3	有一定的学习成本	13
1.3.4	文件易冲突	13
1.3.5	没有代码表达清晰	13
1.3.6	不利于代码的封装和工程架构的组织	13

1.4	Interface Builder 学习的特点	14
1.4.1	简单、容易入门	14
1.4.2	容易犯错误	14
1.4.3	有很多“坑”，需要积累属于自己的经验	14
1.5	Interface Builder 的发展	15
1.6	拓展学习：CocoaPods	15
1.6.1	CocoaPods 介绍	15
1.6.2	Podfile 文件与 CocoaPods 的三种依赖方式	17
1.6.3	podspec 文件	18
第 2 章	使用 Interface Builder	21
2.1	解决冲突	21
2.1.1	解决普通 IB 文件冲突	21
2.1.2	解决 Xcode 8 引起的 IB 文件冲突	22
2.2	创建 xib 文件	24
2.3	“关联” xib 文件与源文件	24
2.3.1	“关联” xib 文件与 UIView 子类的源文件	25
2.3.2	@IBOutlet 与 @IBAction	26
2.3.3	拓展学习：设计模式之 MVC	29
2.3.4	理解 File's Owner	32
2.3.5	“关联” xib 文件与 UIViewController 子类的源文件	33
2.3.6	总结 xib “关联”源文件	34
2.4	使用 xib	36
2.4.1	使用与 UIView 子类源文件关联的 xib	37
2.4.2	使用与 UIViewController 子类源文件关联的 xib	44
2.4.3	xib 嵌套使用的问题	46
2.4.4	深入学习：加载非 main bundle 中的资源	48
2.5	拓展学习：创建 Bundle	51
2.5.1	使用简单的方法创建普通的 Bundle	51
2.5.2	创建基于 macOS Bundle 的 Target 来获得 Bundle	51
2.6	创建 storyboard	53
2.7	使用 storyboard	54

2.8	对比 xib 和 storyboard 的使用情况	55
2.9	拓展学习: App 启动过程分析与 UIApplication 自定义举例	56
第3章	全面学习 xib	63
3.1	操作页面	63
3.2	介绍 Utilities 中的检查器	64
3.2.1	Show the File inspector (文件检查器)	65
3.2.2	Show Quick Help inspector (快速帮助检查器)	67
3.2.3	Show the Identity inspector (ID 检查器)	67
3.2.4	Show the Attributes inspector (属性检查器)	69
3.2.5	Show the Size inspector (尺寸检查器)	85
3.2.6	Show the Connections inspector (连接检查器)	87
3.2.7	“连线”方式总结	88
3.3	Show the Object library 菜单	91
3.4	在 xib 中使用 UITableView	94
3.4.1	将基于 sb 的工程变成基于 xib 的工程	94
3.4.2	利用 xib 简单展示 UITableView	95
3.4.3	利用 xib 自定义 UITableViewCell	96
3.4.4	利用 xib 自定义 UITableViewCell 的使用	99
3.5	Asset Catalog	100
3.5.1	使用 Asset Catalog	101
3.5.2	Asset Catalog 的优缺点	104
第4章	在 Interface Builder 中使用 Auto Layout	105
4.1	IB 与 Auto Layout	105
4.1.1	学习 Auto Layout 的特点	105
4.1.2	在 IB 中使用 Auto Layout 的优缺点	106
4.2	深刻理解 Auto Layout	106
4.2.1	如何理解“Auto Layout 有别于 frame 系统”	107
4.2.2	如何理解“添加约束”	110
4.2.3	如何理解“间接表示”	111

4.3	约束	112
4.3.1	Auto Layout 的数学公式	112
4.3.2	约束属性	113
4.3.3	约束关系	114
4.3.4	multiplier 与 constant	115
4.3.5	约束的创建与添加	115
4.3.6	约束的优先级	119
4.4	Auto Layout 相关菜单介绍	120
4.4.1	Update frames 介绍	120
4.4.2	Embed In Stack 介绍	123
4.4.3	Align 介绍	124
4.4.4	Add New Constraints 介绍	125
4.4.5	Resolve Auto Layout Issues 介绍	127
4.5	Show the Size inspector 再介绍	128
4.5.1	Constraints 菜单	128
4.5.2	Intrinsic Size——固有尺寸	130
4.5.3	Content Compression Resistance——压缩阻力	131
4.5.4	Content Hugging——内容吸附	133
4.5.5	使用 NSLayoutConstraint 与 @IBOutlet “连线”	134
4.6	设置约束的方法介绍	135
4.6.1	在 IB 中设置约束	135
4.6.2	用苹果原生的 API 设置约束	136
4.6.3	用 VFL (Visual Format Language) 设置约束	138
4.6.4	通过第三方库设置约束	139
4.7	在 Interface Builder 中使用 Auto Layout 进行布局的实践	140
4.7.1	用 Auto Layout 布局拨打电话页面	142
4.7.2	stackView 的使用	148
4.7.3	拓展学习: FDStackView	151
4.7.4	拓展学习: “黑魔法”——实现在低版本中使用高版本中才存在的类	152
4.7.5	Auto Layout 的异类——UIScrollView	159

第5章 storyboard 全面学习	164
5.1 Inspector 其他菜单和标签的使用	165
5.1.1 使用 View Controller	165
5.1.2 使用 Page View Controller	167
5.1.3 使用 Table View Cell	169
5.2 使用 Extra View	171
5.3 使用 storyboard 中的 segue	173
5.3.1 使用 Relationship Segue	173
5.3.2 常见的 segue 选项	175
5.3.3 如何利用 segue 传参	178
5.3.4 更灵活的 segue	179
5.3.5 自定义的 segue	180
5.3.6 拓展学习: iOS 中内存管理简介	181
5.3.7 深入学习: Embed Segue	184
5.3.8 Embed Segue 与 UINavigationController 的配合使用	188
5.3.9 深入学习: Unwind Segue	193
5.4 在 sb 中使用 UITableView	198
5.4.1 在 sb 中使用自定义 tableViewCell	199
5.4.2 在 sb 中给 UITableView 添加 tableViewHeader 和 tableViewFooter	203
5.4.3 segue 在 UITableView 中的使用	204
5.4.4 在 sb 中使用静态 cell	205
5.5 在 sb 中使用 UICollectionView	208
5.6 Use as Launch Screen	212
5.6.1 Launch Screen 的使用	212
5.6.2 使用 Launch Screen 的好处	213
第6章 Interface Builder 进阶	216
6.1 Use Trait Variations	216
6.2 User Define Runtime Attribute	221
6.3 IB 文件的加载过程	223
6.4 本地化	227
6.4.1 本地化的介绍	227

6.4.2 文本的本地化	229
6.4.3 Info.plist 的本地化	233
6.4.4 图片资源的本地化	234
6.4.5 App 内设置语言的本地化	235
6.5 Storyboard Reference 的使用	239
6.6 使用 RBStoryboardLink	241
6.7 用 Object 重构“神 VC”	243
6.7.1 使用 Object	244
6.7.2 用 Object 重构“神 VC”的思路	247
6.7.3 如何用好 Object	249
6.8 用 External Object 重构“神 VC”	250
6.8.1 xib 中 External Object 的使用	250
6.8.2 用 External Object 重构“神 VC”的思路	252
6.8.3 Object 和 External Object 总结	253
6.9 IB 中的关键字总结	255
6.9.1 @IBAction	255
6.9.2 @IBOutlet	256
6.9.3 @IBDesignable	257
6.9.4 @IBInspectable	262
6.9.5 User Defined Runtime Attributes、@IBDesignable、@IBInspectable 总结	264
第 7 章 在 Interface Builder 开发中的技巧和 bug	267
7.1 调整 View 的尺寸, 使它与显示内容的尺寸相适应	267
7.2 查看各个 View 之间的距离	268
7.3 在 IB 中添加参考线	268
7.4 快速调整底层被挡住 View 的位置	269
7.5 快速查看 View 的 UI 层次关系	270
7.6 “连线”小技巧	271
7.7 使用“吸管”快速设置颜色	272
7.8 IB 中的复制与粘贴	273
7.9 利用 Media Library 快速设置图片	273
7.10 IB 开发中遇到的一些小 bug	274