第1章

开始 iOS 开发之旅

在学习 iOS 移动开发之前,首先应该将开发环境配置完成,并且对需要使用的开发工具有一 定程度的了解。

本章首先向读者介绍 iOS 11 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 11 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) iOS 应用程序的生命周期。

1.1 iOS 移动开发现状

iOS 是 Apple 公司(简称 Apple)推出的一款操作系统,是用于 Apple 移动设备的移动操作系统,和 Apple 的 macOS 操作系统一样,属于类 UNIX 的商业操作系统。

在 2007 年 1 月 9 日的 Macworld 大会上, Apple 公司公布了这个系统。由于最初是设计给 iPhone 使用的,因此该系统原名为 iPhone OS,即"iPhone 运行 OS X",该系统使用和 macOS 一样的 UNIX 内核。

iPhone OS 系统后来被陆续应用到 iPod touch、iPad 等产品上,这样 iPad、iPhone、iPod touch

三个系列的产品都在使用 iPhone OS 系统,因此在 2010 年的 WWDC 大会上,Apple 宣布其 iPhone OS 改名为 iOS,并获得了思科 iOS 的名称授权。

经历了十余年时间的发展, iOS 系统经历过多个版本, 支持多代的 iPhone、iPad 和 iPod 设备, 目前已经成长为市场上影响力最大、功能最丰富、生态最完整的移动操作系统之一。

1.1.1 iOS 的历史

2007 年,当乔布斯(史蒂夫•乔布斯, Apple 公司联合创始人,别名"乔帮主")带来 iPhone 手机的同时, iOS 移动系统也跟大家见面了。

如今,该系统已经升级到了 iOS 11,它原来的掌舵人也从 Scott Forstall 变为了乔纳森。在图 1-1 中列出了从 iOS 1.0 至 iOS 11.0 共 11 个版本的 Home Screen 的界面变化。



图 1-1 从 iOS 1.0 至 iOS 11.0 共 11 个版本的 Home Screen (摘自 coolketang.com)

第一代 iPhone OS 面世时,存在着各种各样的问题,诸如无法发送 MMS、无法使用复制粘贴 功能、全封闭的系统,并且只有黑色的背景,无法自定义壁纸等。但是这些问题并不能阻止它成为 一个伟大的操作系统。iOS 1.0 拥有大量的创新功能,它展现了革命性的人机交互方式, iPhone 之 前的智能手机都是使用键盘或手写的方式进行人机交互,而 iPhone 则实现了一种全手控操作模式。 所以谷歌地图在第一代 iPhone 上的用户体验比当时在其他平台上的版本好很多。

一年后,伴随着 iPhone 3G 的出现, iPhone OS 才算是得到了真正意义上的完整。2008 年 7 月, Apple 为 iOS 发布了 App Store,并为第三方应用提供了一种可管理的标准模式,用来开发、浏览、 下载和安装应用。如今的 App Store 绝对是 iOS 发展史上最重要的杀手应用,它不仅帮助 Apple 建 立了一个庞大的应用生态,而且让众多的开发者因此找到了自己的商业模式和商业机会,而 Apple 也因此积累了庞大数量的精品应用。

在 iPhone 3GS 发布的同时, Apple 也带来了 iOS 3.0 系统。iPhone 3GS 型号中的 S 也代表速度 (Speed) 的意思,由于硬件上的提升,因此 3GS 在速度上有了不小的升级。从此, iPhone 手机每

两年进入一次大升级,中间一年推出的设备型号的尾部常被标以 S,如 iPhone 4S、iPhone 5S 等,表示在设备性能上进行了一些升级。

此时,自 iOS 3.2 开始被应用于 iPad, Apple 针对大屏幕进行了 UI 优化设计,并且专门针对大 屏幕做了功能优化设计。

在 iPhone 4 时代, iPhone OS 被正式更名为 iOS, iOS 4 开始支持多任务技术,此时的多任务 方式不会受到后台应用占用内存的影响,也能保证不错的续航能力。多任务的菜单是通过双击 Home 键的方式调出的。

Siri 是 iOS 5 最大的亮点,实现了语音控制输入的功能。用户可以通过 Siri 技术,使用语音提问的方式进行人机交互。Apple 在 iOS 5 中取消了对数据线的依赖,可以使用 Wi-Fi 进行设备的激活,这意味着 iPad、iPhone 已经彻底变成了个人移动终端,减轻了对桌面环境的依赖。

Siri 语音助手在 iOS 6 中得到了加强, iOS 6 用户已经可以使用语音打开应用程序, 甚至可以 发送状态到 Facebook 和 Twitter。iOS 6 也被称为一个告别谷歌的系统版本, 最大的一个变化是不 再使用从第一代开始一直内置的谷歌地图, 而是采用了 Apple 自己开发的地图服务。

iOS 系统最突出的一次演变是在 iOS 7 时发生的。iOS 掌门人换成乔纳森后, iOS 7 也进行了 大调整,大家熟悉的拟物化图标全都被拍扁了,多任务界面也发生了巨大的变化,变得更加直观, 用卡片式替代了原有的应用图标,如图 1-2 所示。



图 1-2 左图为拟物风格,右图为扁平化风格

这个变化是从第一代 iPhone 问世以来最明显的不同。

iOS 8 正式版本的系统于 2014 年 9 月 17 日向用户推送,它支持 iPhone 4S、iPad 2、iPod touch (第 5 代)及更新的设备。在这个版本中,Apple Pay 终于和大家见面了,只是当时的中国内地用 户还暂时无法使用该功能。在 iOS 8 中,Apple 也对一直被诟病的输入法进行了升级,终于开放了 对第三方输入法的支持,也正是从该版本开始, iOS 越狱版的用户变得越来越少。

iOS 9 系统比 iOS 8 更加稳定,功能也更加全面,而且还更加开放。iOS 9 加入了更多的新功能,包括更加智能的 Siri 和省电模式。Siri 变得更加智能,甚至在你开口之前,新的主动建议功能就可以帮你把事情办妥; Slide Over、Split View 和画中画功能可以让用户能够以全新的方式在 iPad

上进行多任务处理;还有全新的 QuickType 功能,让快捷键触手可及,从而使格式设定和文本选择 更为轻松。

iPhone 7 刚发布不久,苹果就于北京时间 2016 年 9 月 14 日凌晨 1 点开始全面推送 iOS 10 正 式版。iOS 10 涵盖众多新特性。锁屏界面可以显示丰富的通知功能,左滑右滑可以进入不同的功 能页面,并且可以使用 3D Touch 直接管理通知并进行各项操作,无须解锁就可以进入系统。Siri 变得更加开放,允许开发者的 App 调用 Siri API,你可以使用 Siri 发送微信、调用 Uber 和滴滴打 车软件等;加入了智能脸部和场景的识别,并且增加 LivePhoto 编辑功能。HomeKit 可以管理所有 连接 iOS 的智能硬件,提供场景化功能。而针对中国用户,苹果对电话功能进行了十分体贴的优 化,增加了骚扰电话识别功能。

iOS 11 为全球最先进的手机操作系统设定了全新的标准:利用 Core ML 搭配机器学习的强大 力量使你的 App 变得更智能化; ARKit 帮助你实现美妙得仿佛身临其境的增强现实体验;更有 iPad 的拖放功能、全新的"文件" App、新版相机 API、不同以往的 SiriKit 框架、Apple Music 整合等 最新的多任务性能,助你实现更加统一和美妙的用户体验。

iOS 11 将这个先进的移动操作系统又一次提升至新标准。它不仅翻开了 iPhone 的新一页,更 是开启了 iPad 的新篇章,还将游戏和 App 中的增强现实体验带到这两种设备上,让更多令人眼界 大开的可能成为现实。

iOS 11 让 iPhone 和 iPad 变得更为强大、更个性化,同时也变得更为智能。

一直以来, iPad 都是高效工作方式、畅快娱乐体验和强大学习工具的代表。而 iOS 11 还将其带到了一个新境界。众多新特性和新功能将帮你更高效、快捷地处理事务,为你带来更强大、更个性化的 iPad 使用体验,如图 1-3 所示。而且,很多原来你想用它来做的事,现在它都能做到了。



图 1-3 iPad 下的分屏技术大幅提高工作效率(图片来自 apple.com)

1.1.2 iOS 11 的十一大变化

iOS 在过去 10 年的时间里,对智能手机的发展做出了巨大的贡献,其实现了对使用习惯、用 户体验和人机交互等多方面的多种创新和革命,功能也变得更加完善。 从整个 iOS 11 的发布来看,苹果在这款系统上关心的并非是用户界面的全新设计,而是更注重人工智能、增强现实等系统层级的应用,如图 1-4 所示。Siri 获得了机器学习功能,摄像头也增加了 AR 增强现实功能,同时也融入了人脸识别的算法。



图 1-4 AR 增强现实(图片来自 apple.com)

苹果 iOS 11 的发布或将引领整个手机行业正式进入人工智能和 AR 的时代。在 iOS 的第 11 个版本中, Apple 共有以下 11 项重大革新。

- AR 增强现实:利用手机的摄像头来反映现实,用户可以在屏幕中自由增加一些物件,例如 在一张空白的桌子上放上一杯虚拟的咖啡和台灯。ARKit 是目前最大的 AR 应用资源,通过 AR,用户可以在家里的桌子或地板上获得很有意思的游戏体验。同时,ARKit 提供了位置跟 踪和情景理解功能,使你能创造出让人身临其境般的增强现实产品。
- 机器学习: CoreML 是一个新的基础机器学习框架,能用于众多 Apple 的产品,包括 Siri、相机和 QuickType。CoreML 带来了极速的性能和机器学习模型的轻松整合,使你仅用几行代码就能为产品构建智能化的新功能。
- HomeKit: 当前 HomeKit 已经可以用于控制音箱,并支持同时控制多台音箱,从而形成立体 声的效果。
- Map: 苹果内置的地图不止可以查看地理位置,同时也支持商城或机场中不同楼层的显示, 以及优化停车路线。
- Apple Music: 通过 Apple Music 可以看到你的朋友在听什么歌。
- Camera: 针对中国用户进行优化,苹果将二维码扫描功能加入了相机中。
- Control Center: 控制中心大变样,还能自定义。在控制面板可以增加更多控制按钮,重新优化功能界面,每个按钮都可以呼出一个特定的模块。笔者非常喜欢控制面板提供的录屏功能,终于不需要借助第三方工具即可方便地录制应用视频了。
- App Store: 加快了新 App 的审核(最快数小时审核完毕),对开发者比较有意义的改变是,终于可以回复用户的评论信息了。除了控制中心外,界面大变身的还有 App Store。为了打击刷榜行为, App Store 经过了重新设计,新的界面增加了 Today 选项,它会每天向用户推荐一款应用和一款游戏以及一个应用清单,以供用户选择。
- Siri: Siri有了新的声音,通过深度学习,Siri的声音变得更加自然。同时,还添加了即时翻译功能,支持将英语翻译为中文、法语、德语和意大利语等。

- iMessages: 全部信息可以同步到 iCloud 中,并提供安全加密技术,保证用户使用安全。
- Apple Pay: Apple Pay 不仅可以在网页上支付、向商家支付,同时还增加支持用户之间的交易,也就是类似微信的转账功能。

1.1.3 iOS 应用开发的八个特点

iOS 作为移动操作系统,在开发运行在其平台上的应用和游戏时,需要注意和传统软件的开发 方式的一些不同之处。

- 屏幕的不同:采用 iOS 系统的 iPhone 屏幕较小,只是把需要显示给用户的内容合理地组织在 一块小小的屏幕上,所以需要设计者进行精心的设计和排版。
- 交互方式的不同: iOS 系统采用手指触摸的方式进行人机交互,所以要尽可能使按钮等交互 控件的尺寸保持在 44 点以上,以避免误操作。
- 内存的不同:运行 iOS 系统的移动设备,通常内存在 512MB~3GB 之间。用户需要在应用 中合理地使用多媒体素材,保证应用不会因太耗内存而被系统自动关掉。
- 电量的不同: 作为运行在移动设备上的应用,需要尽可能降低电量的消耗。比如及时关闭地 理定位服务,减少不必要的网络请求;尽量避免以轮询的方式工作,不然会使 CPU 无法进入 睡眠状态,从而引起电量的长时间消耗。
- 安全方面的限制:一个 App 作为一个程序束(bundle)存在, App 只可以访问其资源束之内的文件夹或其他资源文件。
- 可访问的设备众多:在 iOS 中运行的应用可以访问移动设备自带的加速计、陀螺仪、地理定 位设备、蓝牙、相机等。
- 少用菜单: iOS 系统中的应用很少使用菜单进行页面之间的跳转,而是通常采用导航控制器 或标签控制器进行页面之间的导航。
- 程序退出方式的不同: iOS 系统中的应用没有最小化和关闭按钮。用户通过按下设备底部的 Home 键退出正在运行的应用,应用退出后仍然会在内存中保留一段时间。

1.2 iOS 11 和 Xcode 9 新增的开发技能

Apple 公司的技术团队是非常棒的,他们把基于 macOS 的移动计算与多点触摸屏幕、位置感 知等技术结合在 iOS 中。

Apple 公司于 2008 年 3 月首次推出 iPhone SDK beta 版本,仅仅用了 10 年, iOS 的版本号就 已然风风火火地升级到了 iOS 11。遥想当年, iOS 1.0 版本还没有问世,就被乔布斯认为这个不符 合用户需求,内部 pass 掉了。后来的 iPhone SDK 上市不到一周,下载量就超过了 10 万。

Apple 保持着每年更新一个技术版本功能, iOS 迭代的速度之快让同行望尘莫及。iOS 5 增加 了 ARC 和 iCloud, iOS 6 增加了内购, iOS 7 增加了对 Bit 64 支持、TextKit 和全新的扁平化设计, iOS 8 增加了 Size Class 和 Autolayout, iOS 9 增加了 3D Touch 和 iPad 分屏等丰富有趣的新功能, iOS10 增加了 SiriKit 框架。那么 iOS 11 带来了哪些新的 iOS 开发技能呢?下面我们一一 道来。

1.2.1 ARKit 增强现实

iOS 11 中增加了 ARKit 的框架,利用 AR 增强现实可无缝融合虚拟对象与真实世界,提供浸入感十足的愉悦体验,使开发者可以在全球最大的增强现实平台一一 iOS 11 之上为数亿用户创造 无可比拟的逼真增强现实体验。

通过将数字对象和信息与你周围的环境相融合,ARKit 为 App 解开了屏幕之缚,带领着它们跨越屏幕的界限,让它们以全新的方式与现实世界交流互动。

ARKit 可以被应用在以下场景:

- 语音和视频通话。
- 发送消息。
- 发送或接收付款。
- 搜索照片。
- 约车。
- 管理健身。
- 设置 CarPlay (仅支持汽车企业)。
- 酒店预定 (需要 Apple 的额外支持)。

1.2.2 CoreML 机器学习

CoreML 允许你将众多机器学习模型集成到 App 中。它不但有 30 多种层来支持广泛的深度学 习,而且还支持诸如决策树、SVM 和广义线性模型等标准模型。由于它构建在 Metal 和 Accelerate 等低阶技术之上,因此 Core ML 能无缝地利用 CPU 和 GPU 来提供最大化的性能和效率。

CoreML 可以被应用在以下两个场景。

- 视觉:在 iOS 11 中,开发者可以轻松地在 App 中构建计算机视觉机器学习功能,支持的功能 包括脸部跟踪和检测、地标、文本检测、矩形检测、条形码检测、对象跟踪和图像配准。
- 自然语言处理:利用机器学习,Foundation 的自然语言处理 API 拥有语言识别、标记化、词 形还原、词性和命名实体识别等功能,能深入地理解文本的内容。

1.2.3 Face ID

全新的解锁、验证和付款的安全方式,让用户仅扫一眼,就能够快速访问你的 App。借助于原深感摄像头,面容 ID 不仅成为可能,设置起来也非常简单。原深感摄像头投影并分析超过 30000 个肉眼不可见的光点,以加密并保护设备上 安全隔区的面谱。Face ID 的设置如图 1-5 所示。



图 1-5 Face ID 的设置

1.2.4 拖放操作

iOS 11 增加了全操作系统范围的拖放功能,为用户提供了一种快速、简便的方法,将文本、 图像和文件从一个应用程序移动到另一个程序。只需点击并按住以拾取内容并将其拖动到其他应用 程序。多点触摸作为一种全新的选择多个项目的方法,让你可以通过点击快速拾取多个项目,如图 1-6 所示。使用 spring-loading 弹性加载,你可以通过在 Dock 或主屏幕上的应用程序图标上拖动内 容来复制到另一个应用程序。



图 1-6 借助强大的 Multi-Touch 拖放操作使用户在 iOS 中的不同 App 之间 能够快速简单地移动文字、图像及文件

1.2.5 Core NFC

在 iOS11 中,苹果开放了 NFC (Near Field Communication,近场通信),作为推广 Apple Pay 的一种策略。应用程序可以通过 NFC 读取标签,从而给用户提供更多有关其物理环境和真实世界 的对象内容信息。例如,你的应用程序可以给用户提供商店里的商品信息,或者是他们在博物馆参 观时陈列品的信息。

虽然从 iPhone 6 开始就已经有 NFC 硬件了,但是开发接口一直没有开放,苹果官方的 Apple Pay 应用则使用了 NFC 功能。虽然从文档看来目前只开放了 NFC 读数据的接口,而没有开放 NFC 写数据的接口,但是这也算是一个好的开始了。Apple Wallet 之前的应用都是登机牌之类的应用,通过光学扫描 QR 码实现,不过相对来讲,NFC 则更可靠,也更安全。

2018 年 3 月 30 日凌晨,苹果公司正式推送了 iOS 11.3 系统,此次更新最大的升级点是加入"快捷交通卡"功能,也就是 iPhone 和 Apple Watch 借助 NFC 技术变为交通卡。用户期待多年的手持 iPhone 乘公交、地铁的功能终于得以实现,同时还加入了对 Apple Watch 的支持,算是一个惊喜。

1.2.6 安全区布局指南

iOS 11 为 UIViewController 视图控制器和 UIView 视图增加了两个新的属性: safeAreaInsets 和 safeAreaLayoutGuide,通过这两个属性,开发者可以获得安全区域的范围,通过图 1-7 可以很清楚地 看到安全区域的范围,开发者要做的是让那些不能被遮挡的内容和控件在安全区域范围内显示。



图 1-7 针对 iPhone X 的安全适配区域(摘自 apple.com)

需要注意的是, safeAreaInsets 属性适用于手动计算, 而 safeAreaLayoutGuide 属性则适用于自动布局。

1.2.7 Asset UIColor

Xcode 9之后,开发者不仅可以在 Asset Catalogs 中添加图片,还可 以在其中添加 Color,这样我们就可以省略声明一大堆颜色,并且使用 Asset UIColor 可以很方便地定义和更换应用程序的主题颜色。

此外,Asset Catalogs 还支持 Preserves Vector Data,当开发者在代码 中加载矢量图片的时候,如果需要让图片显示的尺寸比本身尺寸大,系统 在运行时就会自动把它放大。也就是说,系统在渲染图片时,不会有任何 质量损失,如图 1-8 所示。



图 1-8 系统渲染的图片

1.2.8 Wireless Development

在 Xcode 9 和 iOS 11 之前,开发者在调试项目的时候, iOS 设备都要用数据线和 Mac 相连,这样 Xcode 软件才能识别设备。升级到 Xcode 9.0 之后,Mac 电脑可以通过 WIFI 连接处于同一 WIFI 环境下的 iOS 或 tvOS 设备进行无线调试。

这是由于 ARKit 的到来,为了方便开发者测试增强现实应用,因此苹果开发了一种无须通过数据线即可连接真机进行应用的测试。

目前,无线调试功能除了向测试设备输出数据比有线传输稍慢一些之外,在便利性方面大大 超过有线调试,未来将是应用测试的首选方式。 Wireless Development 对版本的要求是: Xcode 9.0 以上版本、macOS 10.12.4 以上版本、iOS 11.0 以上版本、tvOS 11.0 以上版本。

1.2.9 GitHub

在 Xcode 9 之前,虽然 Xcode 也带有代码版本 控制的功能,但是由于功能相对比较简单,因此容 易出现提交错误的情况。

在新的 Xcode 9 中,苹果引入了 GitHub,新的 源代码管理导航器、分支机构和标签,使 Xcode 和 GitHub 进行深度整合,对开发者来说就更加方便了, 如图 1-9 所示。

DemoApp Branches Y master Tags	New Branch New Tag Checkout	/S
v Indes v v1.0	Merge from Branch Merge into Branch	
	Create "DemoApp" Remote on GitHub Add Existing Remote	
	View on GitHub	
	Delete	
	Source Control Navigator Help	

图 1-9 Xcode 9 的 Source Control Navigator 界面

1.3 成为一名 iOS 开发者

iOS 开发人员和其他传统开发者相比较,并非只需坐下来即可进行编码操作。 首先,iOS 开发者需要拥有一些其他开发者无法具备的硬件和软件条件,下面进行详细的说明。

1.3.1 硬件上的需求

开发 iOS 应用,首先你需要使用 Apple 的电脑。虽然也可以使用黑苹果或者虚拟机在 PC 上进行 iOS 开发,但是作为一名专业的 iOS 开发者,你应该有一台真正的 Mac 电脑。

当你打算购买 iOS 开发必需的设备时,可以依据是否移动办公和资金限制两个因素进行设备的选择:

- 第一个因素是你是否需要移动办公,如果只是在固定的位置办公,你可以选择 iMac,它拥有 21 寸和 27 寸两种屏幕,每个尺寸的屏幕又有低配和高配的区别。 如果需要移动办公,比如经常进行代码 Review、功能演示、进度通告、客户沟通等,建议你 选择 MacBook Pro 笔记本,该设备分为 13 寸和 15 寸两种型号,每种型号又有低配和高配的 区别。使用 MacBook Pro 可以提高开发效率,并且在进行幻灯片演示和讲解时非常方便。在 此并不推荐使用 MacBook Air。
- 对于资金有限的朋友来说,可以选择 MacBook 笔记本。投入最低的选择是 Mac Mini 加普通的显示器,这样只需要 4000 元左右就可以搭建好 iOS 开发环境了。

硬件采购的原则是,在资金允许的情况下,尽可能购买配置较高的设备。由于移动开发周期 短的特点,开发过程往往处于赶工的状态。当使用一台配置较高的设备进行开发工作时,你会发现 这的确会节省大量的开发时间。iOS 开发者最大的痛苦莫过于,当你急需处理用户或测试人员反馈 的 Bug 时,电脑的编译速度却像蜗牛一样不给力。

1.3.2 加入 Apple 开发者计划

现在你已经拥有了一台可以进行 iOS 开发的设备,在编写代码之前还需要加入 Apple 的开发 者计划(Developer Program)。加入开发者计划之后,你就可以下载 Xcode 开发软件,编写 iOS 程序,并且在 iPhone 或 iPad 模拟器中运行你的程序,而这一切都是免费的。

当然免费是有一定限制的,你无法使用 beta 版的 iOS SDK,无法进行真机测试,也无法向 App Store 发布你的产品。

iOS SDK 大部分功能是可以在模拟器中测试的,无须真机测试,但是像应用内购买、社会化 分享、3D Touch、超过两点的多点触摸等功能,以及相机、传感器等设备的使用,就需要在真机 环境进行测试了。

创建 Apple ID

在苹果开发者网站可以创建一个 Apple ID,作为享有苹果所有服务的凭证。首先访问 Apple 的开发中心 https://developer.apple.com,点击右上角的"Account"链接,进入开发者账户管理页面,如图 1-10 所示。如果你已经通过 iTunes、iCloud 或其他的 Apple 产品创建了 Apple ID,可以直接 使用该账号进行登录。



图 1-10 Apple 开发者首页

如果你还没有 Apple ID,可以点击登录页面的"Create Apple ID"按钮,创建一个 Apple ID, 如图 1-11 所示。这样就进入了注册页面,如图 1-12 所示。

 Sign in with your Apple ID - × Apple Inc. (US) https://idmsa.apple.com/IDMSWebAuth/lo. 	 = ₽ ☆ ₽	Create Your Apple ID × Create Your Apple ID × Create Your Apple Inc. (US) https://appleid.apple.com/account?localang=	- ŵ
🗯 Developer		Create Your Apple ID	
tapinfinity@gmail.com		One Apple ID is all you need to access all Apple services. Already have an Apple ID? Find it here >	
		nam e@example.com	
Create Apple ID Sign In		password	
and the second second second second second		confirm password	
		first name last name	
		birthday	
rems of use T Privacy Policy system status			A.

图 1-11 登录页面

图 1-12 Apple ID 注册页面

在创建 Apple ID 的页面中,注册流程将引领你创建一个新的 Apple ID。由于 Apple ID 是使用 邮箱作为账号的,因此当完成注册流程后,Apple 会往你的邮箱中发送一封邮件,点击邮件中的链 接,即可激活你的 Apple ID。

加入付费的开发者计划

当你拥有一个 Apple ID 之后,便可以申请加入付费的开发者计划了。这里有 3 个选择,如表 1-1 所示,根据你的情况选择适合的项目。

项目	说明
Individual	个人开发者账户,每年 99 美金。可以在 App Store 发布产品或在 100 个 iOS 设备上进行测试。 该账户在 App Store 中显示为个人姓名
Company	公司开发者账户,每年 99 美金。可以在 App Store 发布产品或在 100 个 iOS 设备上进行测试, 并且允许多个开发者协作开发。该账户在 App Store 中显示的销售者为公司名称。申请该类型的 账户时,需要填写公司的邓白氏编码(D-U-N-S)
Enterprise	企业开发者账户,每年 299 美金。该账号不能在 App Store 中发布产品,但是可以在企业内部进行无设备数量限制的分发

表 1-1 Apple 开发者计划的类型和说明

申请加入付费开发者计划,需要等待一定时间通过 Apple 的审核。同时需要注意的是,在填写银行账户信息时,还需要填写信用卡账号,以便 Apple 从中扣除 99 美金或 299 美金的年费。

申请加入付费开发者计划成功后,需要在 iTunes Connect 的"协议、税务和银行业务"页面填写储蓄卡账号,以接收 Apple 发送的在 App Store 中的收益,如图 1-13 所示。

Banking Information	
You will not be able to modify your Bank Account Number. If the Bank Account Number listed below is incorrect, you will have to Add a New Bank Account.	
Bank Country: China	
CNAPS Code: 1041000044	
Bank Name: Bank of China Limited	
Bank Account Number: ****1692	
Account Holder Name: Note: You must enter the Account Holder Name EXACTLY as it appears on your bank account or your payment may be rejected.	
Bank Account Currency: USD - US Dollar 🔯 🕐	

图 1-13 开发者账号的银行信息设置页面

1.3.3 下载安装 Xcode 和模拟器

终于到了安装 Xcode 开发软件的环节了。

Xcode 的安装有很多方法,之前大家都是使用迅雷或者在各大网盘进行下载,毕竟当时从 Apple 官网下载 Xcode 比较慢,而且经常出现即将下载成功时,突然莫名其妙停止了。但是自从 2015 年 Xcode Ghost 事件发生后, iOS 开发者再也不敢从其他渠道下载 Xcode 了,而是返回 Apple 官网进 行下载。好在目前 Apple 已经对服务器进行了优化, Xcode 的下载不再费时费神。

步骤 ① 我们选择从 Apple 官网进行 Xcode 的下载和安装。Xcode 的下载和安装十分简单和方便, 只需要在 macOS 中打开 App Store 软件,然后在右上角的搜索框内输入 Xcode 并按 Enter 键,就可以 找到 Xcode 开发软件。

(步骤 2) 点击"获取"按钮即可自动下载和安装,如图 1-14 所示。



图 1-14 下载并安装 Xcode

#爾**③** Xcode 下载完成后,打开 Xcode 软件,你会发现它已经集成了最新版的 iOS 模拟器。但 是在 iOS 开发工作中,我们经常需要测试老版本的 iOS 系统,所以你需要手动安装其他版本的 iOS 模 拟器。依次点击 Xoce→Preference→Components 命令,进入组件面板,然后在该面板中选择安装所需 版本的 iOS 模拟器,如图 1-15 所示。

eral Accounts Behaviors Navigation Fo	A onts & Colors Text Editing	Key Bindings So	urce Control Components	Locations Server & Bots
	Sin	nulators		
Simulator				Size
(a) iOS 11.1 Simulator				2.1 GB
iOS 11.0 Simulator				2.09 GB
iOS 10.3.1 Simulator				1.98 GB
iOS 10.2 Simulator				1.93 GB
(In the second s				1.9 GB
iOS 10.0 Simulator				1.89 GB
(4) iOS 9.3 Simulator				1.53 GB
iOS 9.2 Simulator				
iOS 9.1 Simulator				1.49 GB
iOS 9.0 Simulator				1.46 GB
iOS 8.4 Simulator				1.37 GB
iOS 8.3 Simulator				1.36 GB
iOS 8.2 Simulator				

图 1-15 安装其他版本的模拟器

1.3.4 创建开发和发布配置文件 Provisioning Profile

如果你加入了付费的开发者计划,并且下载了 Xcode 开发软件和 iOS 模拟器,仍然无法在真机上测试编写的程序,那是因为你还没有创建并安装开发配置文件。

由于 Apple 对 iOS 应用分发的限制,禁止开发人员将开发的 iOS 应用随意进行传播,因此需要将你的账户和应用相关信息添加到开发配置文件中。该配置文件包含 iOS 开发者、应用程序 ID 以及运行该应用程序设备的唯一标识符,从而使你开发的应用程序能够安装在测试设备上供测试人员使用。如果需要在 App Store 上发布你的应用,那么还需要创建并安装发布配置文件。

创建开发证书

步骤① 你需要在 Apple 开发者平台创建配置文件,首先使用开发者账号登录 https://developer.apple.com网站,然后选择左侧的"Certificates, IDs & Profiles"菜单,进入证书、应用 程序 ID 和配置文件的创建页面,如图 1-16 所示。

É Developer	Account	Li Fazhan ↓ Li Fazhan
Program Resources		
≔ Overview	Li Fazhan	
① Membership	Apple Developer Program	
 Certificates, IDs & Profiles 		
(1) iTunes Connect		

图 1-16 点击 Certificates, IDs & Profiles

步骤 ② 在 "Certificates, IDs & Profiles"页面中,我们开始创建开发配置文件,首先添加一个开发证书,如图 1-17 所示,点击右上角的+号图标。

iOS, tvOS, watchOS 🔹	iOS Certificates			
Ø Certificates	2 Certificates Total			
a All	Name	Туре	Expires	
Pending	Li Fazhan	iOS Development	Mar 06, 2017	
Development	Li Fazhan	iQS Distribution	Dec 21, 2016	

图 1-17 点击右上角的+号图标

步骤 ③ 接着在证书类型选择页面中选择 "iOS App Development"选项,创建一个开发类型的 证书,然后点击 "Continue" 按钮,进入证书签名 (Certificate Signing Request) 页面,如图 1-18 所示。

Pending Development Production	What type of certificate do you need?
Identifiers App IDs	Development
 Pass Type IDs Website Push IDs ICloud Containers 	iOS App Development Sign development versions of your iOS app.

图 1-18 选择 "iOS App Development"选项

步骤 ④ 要创建证书签名 CSR 文件,首先依次进入并打开"应用程序"→"实用工具"→"钥 匙串访问"工具,接着在顶部的菜单栏中依次点击"钥匙串访问"→"证书助理"→"从证书颁布机 构请求证书"命令。然后在弹出的证书助理设置窗口中,在"用户电子邮件地址"输入框内输入你的 电子邮箱,然后在"请求是"设置区域中选择"存储到磁盘"选项,最后点击"继续"按钮,即可完 成 CSR 文件的创建。 步骤 5 CSR 文件创建完成后,返回浏览器,在 CSR 选择页面中找到并打开该文件,最后点击 Continue 按钮,完成开发证书的创建。

创建 App ID

● 每个 iOS 应用都有一个唯一的 App ID,作为其在 App Store 中的唯一标识符。点击左侧的 App IDs 菜单,进入 App IDs 列表页面,然后点击右上角的+号图标,注册一个新的 App ID,如图 1-19 所示。

ID Identifiers	
App IDs	The App ID string contains two parts separated by a period (.) — an App ID Prefix that is
Pass Type IDs	defined as your Team ID by default and an App ID Suffix that is defined as a Bundle ID search
Website Push IDs	string. Each part of an App ID has different and important uses for your app. Learn More
iCloud Containers	
App Groups	App ID Description
Merchant IDs	
Devices	Name: App ID for Demo
= All	You cannot use special characters such as @, &, *, ', "
Apple TV	
Apple Watch	App ID Profix
= iPad	App 10 Frenx
I iPhone	Value: N43JNYX84B (Team ID)
■ iPod Touch	
Provisioning Profiles	App ID Suffix
Development	Explicit App ID
Distribution	If you plan to incorporate app services such as Game Center, In-App Purchase, Data
	Protection, and iCloud, or want a provisioning profile unique to a single app, you must register an explicit App ID for your app.
	To create an explicit App ID, enter a unique string in the Bundle ID field. This string
	snould match the Bundle ID of your app.
	Bundle ID: com.coolketang.appName
	We recommend using a reverse-domain name style string (i.e.,
	com.domainname.appname). It cannot contain an asterisk (*).

图 1-19 创建 App ID

(步骤2)在页面中依次输入以下3项内容。

- App ID Description: App ID 的描述信息,比如 App ID for Demo。
- App ID Prefix: 设置一个团队 ID, 用于创建一系列应用。
- Bundle ID: 应用的唯一标识符,使用反写域名的方式,例如 com.coolketang.appName。

步骤 ③ 输入完成后,点击页面底部的 Continue 按钮,完成 App ID 的创建。需要注意的是, App ID 一经创建无法删除。

添加测试设备

步骤① 添加测试设备的 UDID 号,获得 iPhone 或 iPad 设备的 UDID 号的步骤为:将设备连接 至计算机,然后打开 Xcode 软件,依次点击 Window→Devices 菜单命令,打开 Devices 窗口,在该窗 口中获得设备的相关 UDID 号。

步骤 2) 返回开发者平台网页,点击左侧的 Devices→All 命令,进入测试设备列表页面,接着点击右上角的+号图标,注册一款新的测试设备,如图 1-20 所示。

步骤 ③ 在注册设备页面中, 依次填写设备的名称和 UDID 信息。设备名称可以由开发者自行定 义, UDID 可以将在 Xcode 的 Devices 窗口中复制的内容粘贴在此输入框内。最后点击页面底部的 Continue 按钮, 完成设备的添加。

Devices	
= Ali	Register Device
Apple TV	Name your device and enter its Unique Device Identifier (UDID).
Apple Watch	Name: Tester
iPad	
iPhone	UDID: ebe5d26fb2fd670eaa36bc57f2e4c0604bd6f6d2
iPod Touch	

图 1-20 注册一款新的测试设备

生成开发配置文件

步骤 ① 在开发者管理平台,点击左侧的 Provisioning Profiles→All 菜单命令,进入证书列表页 面,然后点击页面右上角的+号图标,进入创建配置文件页面,如图 1-21 所示。在该页面中,选择 iOS App Development 选项,创建一份开发配置文件。

Apple TV Apple Watch	Select Type Configure Generate Download
≡ iPad ≡ iPhone ≡ iPod Touch	What type of provisioning profile do you need?
Provisioning Profiles	
 All Development 	Development
Distribution	iOS App Development Create a provisioning profile to instanti development apps on test devices.
	 tvOS App Development Create a provisioning profile to install development apps on tvOS test devices.

图 1-21 选择 iOS App Development 选项

步骤 ② 点击页面底部的 Continue 按钮,进入下一步设置页面,依次设置证书的 App ID、选择 测试设备 UDID,以及选择之前创建的开发证书等刚刚完成的操作。最后点击"下载"按钮,下载生 成的开发配置文件。

步骤 ③ 打开 Xcode 软件,双击下载的开发配置文件即可完成所有开发配置文件的安装。生成 发布配置文件的流程和开发配置文件相同,而且不必再重复前 3 步操作,只需在第 4 步创建一个 Distribution Provisioning Profile 即可。

1.4 创建一个基本的 iOS 项目

我们已经完成了 Apple ID 的注册、加入 Apple 开发者计划、Xcode 软件和模拟器的下载与安装以及开发配置文件的创建等操作,现在可以创建并运行一个 iOS 项目了。

1.4.1 项目的创建

步骤 ① 打开下载并安装好的 Xcode 9.2 软件,首先进入的是 Xcode 欢迎页面,如图 1-22 所示。 在 Xcode 欢迎窗口的左侧可以创建一个 playground 来学习 Swift 语言,也可以创建一个空白项目, 或者从 SCM 源码管理系统中 check out 一个项目。窗口的右侧是曾经打开过的项目列表,从中可以选择并快速打开一个历史项目。



图 1-22 Xcode 9.2 的欢迎页面

步骤 ② 点击选择左侧的 Create a new Xcode project 项目,打开 Xcode 项目模板选择窗口(也可以依次点击 File→New→Project 命令),如图 1-23 所示,在模板选择窗口中,选择 Single View App 选项,创建一个单视图项目。

000 🕨 🔳						
	Choose a template for y	our new project:				D Ø
	IOS watchOS tvOS	S macOS Cross-pl	latform	(Filter	
	1	R	AR			No Selection
	Single View App	Game	Augmented Reality App	Document Based App	Master-Detail App	
1. 洗择项目模板		*		\bigcirc		
	Page-Based App	Tabbed App	Sticker Pack App	iMessage App		0
	Framework & Libra	ry				
	ē		(M)	2	2. 点击 Nex	t 按钮
	Cocoa Touch Framework	Cocoa Touch Static Library	Metal Library			
	Cancel			Pre	vjous Next) Filter

图 1-23 Xcode 项目模板选择窗口

步骤 ③ 点击 Next 按钮,进入项目信息设置页面,如图 1-24 所示。在该页面中,依次输入产品 名称、组织名称、组织标志符,然后选择项目的语言为 Swift,选择设备的类型为 Universal,即当前 的产品将可以在 iPhone、iPod 和 iPad 上运行。



图 1-24 项目信息设置页面

步骤 ④ 点击 Next 按钮,进入下一步页面,在该页面中选择项目的保存位置,最后点击 Create 按钮,完成项目的创建。Xcode 将自动打开创建的项目,效果如图 1-25 所示。



图 1-25 Xcode 自动打开创建好的项目

项目创建后,Xcode 已经自动生成了 App 的唯一标签符、版本号、编译号等信息。

- Display Name: App 图标下方的名称。
- Bundle Identifier: App 的唯一标识符。
- Version: App 当前的版本号。
- Build: App 当前的编译版本号, 你可以向 App Store 提交多个相同 Version 号而不同 Build 号的 App。主要用于审核失败时, 重新提交相同 Version 的 App。

1.4.2 设置项目的信息

项目创建完成后,我们需要对其进行一些设置操作,首先进入 Deployment Info 设置区,进行一些部署设置,如图 1-26 所示。

V 🗟 DemoApp	🔲 😂 General	Capabilities	Resource Tags	Info	Build Settings	Build Phases	Build Rul
DemoApp AppDelegate.swift ViewController.swift Main.storyboard Assets.xcassets	Deployment Info Deplo	o yment Target S Devices	0.0 Universal		0		
LaunchScreen.storyboard Info.plist ▼ Products jAt DemoApp.app	N	fain Interface M e Orientation	Main Portrait Upside Down Landscape Left Landscape Bight				
+ (S Filter	Ste	atus Bar Style	Default Hide status bar Requires full screen		٢		

图 1-26 Deployment Info 设置区

- Deployment Target: 设置 App 可以在 iOS 9.0 以上系统中运行。
- Devices: 设置 App 可以在所有 iOS 设备上运行。
- Main Interface: 设置 App 默认启动的故事板文件为 Main.storyboard。
- Device Orientation: 设置 App 在设备中运行的朝向为竖向(共有竖向和横向两种)。
- Status Bar Style: 设置顶部状态栏的样式,默认为暗色背景。

App 在设备上运行时,在图标的下方会显示 App 的名称。App 的名称可以在 Build Setting 面 板中设置,如图 1-27 所示。在搜索框内输入 product name 搜索相应的参数。该参数的默认值为 DemoApp,在参数名称右侧的值字段上双击,打开编辑窗口,并输入新的产品名称,输入完成后 按键盘上的 Enter 键完成命名操作。

V 🚡 DemoApp	D \$A	General	Capabilities	Resource Tags	Info	Build Settings	Build Phases	Build Rule
DemoApp AppDelegate.swift ViewController.swift	Basic	All Combir	1. 输入	.参数名称	Q~	product name		۵
Main.storyboard	▼ Packag	ging Setting			Pr Demo	App		-11
LaunchScreen.storyboard		Product Name	e		第一桶金	<2	更改产品名	称
Products								

图 1-27 在 Build Setting 面板设置产品名称

1.4.3 设置应用的 Icon 图标

为了支持更多的 iOS 设备、更高分辨率的屏幕, iOS 开发者往往需要为自己的应用准备各种尺 寸的主程序图标文件,而且这个文件列表随着苹果移动设备类型的增加而越来越多。

之前我们为应用设置图标通常需要制作各种分辨率的图标,并且按照规则进行命名。而现在 给应用程序设置图标变得非常简单,首先点击项目信息设置页面的 General 标签,进入通用信息设 置页面,然后在 App Icons and Launch Images 设置区点击 App Icons Source 右侧的右向箭头,进入 Icon 设置面板,如图 1-28 所示。



图 1-28 进入 Icon 设置面板的操作步骤

在 AppIcon 设置页面中,按用途和 iOS 的版本对图标进行了分组,每组包含不定比例的尺寸,从 1 至 3 倍的缩放比例不等。例如左上角的 29pt 的 3 个图标,其中 1x~3x 分别用在 iPhone 3GS 及 iPad 2 以下的非 Retina 设备、iPhone 4 至 iPhone 5 和 iPad 3 以上的 Retina 设备、以及 iPhone 6 以上的设备。

你需要把 29px×29px、58px×58px 和 87px×87px 三种规格的 PNG 图标拖放到对应的位置。 PNG 图标素材不需要包含透明像素,并且不需要进行特别的命名,只需要把对应尺寸的图片拖放 到相应的位置即可,最终效果如图 1-29 所示。



图 1-29 设置应用的 Icon 图标

1.4.4 设置应用的 Launch 图片

当用户使用手指点击某个应用图标时, iOS 就会将该应用加载到内存中。有时加载的时间比较 长,此时系统就会根据当前设备的分辨率加载开发者设置好的 Launch 图片,以避免用户失去等待 的耐心。请注意, iOS 无法容忍加载时间过长的应用,超过一定加载时间的 App 会被 iOS 系统直 接关闭掉。

现在来给新建的项目添加 Launch 图片。

步骤① 在 Xcode 界面左侧的项目导航区选择 Assets.xcassets,进入图像资源管理页面。Asset Catelog 是自 Xcode 5 起引入的一个新的图片管理方式。Assets.xcassets 可以自动管理内部的图片,如 picName@1x、picName@2x、picName@3x,读取图片的时候,只需要使用 picName 名字即可, iOS 会根据当前设备的类型加载相应尺寸的图片。

步骤 ② 点击底部的加号图标,弹出资源导入菜单,并依次选择 App Icons & Launch Images → New iOS Launch Image 菜单命令,添加名为 Launch Image 的 Asset Category,如图 1-30 所示。由于之前已经设置了应用的图标,因此此处存在一个名为 AppIcon 的资源组。

步骤 ③ 在 Launch 图片设置页面中,按设备类型和设备的朝向对 Launch 图片进行了分组,每组 包含相同的设备朝向和不同的设备类型。针对不同的设备和不同的朝向,将 Launch 图片拖放至相应 的位置。最终的结果如图 1-31 所示。图标对应素材的尺寸如表 1-2 所示。







图 1-31 设置 Launch 图片

表 1-2 竖向设	备 Launch	图片尺寸列表
-----------	----------	--------

设备类型	尺寸
iPhone Portrait Retina HD 5.5	1242×2208
iPhone Portrait Retina HD 4.7	750×1334
iPhone Portrait 2x	640×960
iPhone Portrait Retina 4	640×1136
iPad Portrait 1x	768×1024
iPad Portrait 2x	1536×2048

步骤 ④ 添加完 Launch 图片之后,还需要进行一些设置操作,以取消默认的 Launch Screen File, 并设置 Launch 图片的源,如图 1-32 所示。

The DemoApp	Dp \$	General	Capabilities	Resource Tags	Info	Build Settings	Build Phases	Build Rul
DemoApp AppDelegate.swift ViewController.swift Main.storyboard Assets.xcassets	▼ App Icons a	and Launch I Lau	mages App Icons Source Inch Images Source	Apploon	0	1.	设置启动图	片源
Info.plist	▼ Embedded	Binaries	unch Screen File			2.	清空此坝的	川谷

图 1-32 取消默认的 Launch Screen File

1.4.5 编译并运行项目

至此,已经完成了第一个 iOS 项目创建和配置的所有操作,只是还没有进行任何编码操作。 现在我们来编写一行代码,修改当前视图控制器根视图的背景颜色。

步骤① 在 Xcode 界面左侧的项目导航区选择编辑 ViewController.swift 文件, 然后在 viewDidLoad 方法中输入一行代码, 将根视图的背景颜色修改为红色:

```
1 self.view.backgroundColor = UIColor.red
```

效果如图 1-33 所示。



图 1-33 修改当前视图控制器根视图的背景颜色

步骤 2)点击 Xcode 窗口顶部的 Set the active scheme, 弹出模拟器列表, 在列表中选择需要进行 测试的模拟器, 如图 1-34 所示。



图 1-34 选择需要进行测试的模拟器

步骤 ③ 点击左上角的 Build and run the current scheme 按钮,打开选择的模拟器,并在模拟器中运行第一个项目。

模拟器打开后,在加载应用的过程中会显示刚刚设置的 Launch 图片,如图 1-35 所示,接着很快就会进入应用程序界面,如图 1-36 所示。

接着点击模拟器底部的 Home 键,或者依次点击 Hardward → Home 菜单命令,也可以使用 Command + Shift +H 快捷键,在模拟器中返回 Home Screen,你会在应用列表中发现名为第一桶金 的应用,其图标为 1.3.3 节中设置的图标,如图 1-37 所示。



1.5 iOS 应用的生命周期

对于一个 iOS 应用来说, 生命周期是一个至关重要的概念。iOS 应用的生命周期是指从应用程序的启动到应用程序结束整个阶段的全过程。

整个 iOS 应用的生命周期包含各种状态,有时系统会从应用的一种状态切换至另一种状态来响应系统发生的事件。例如,当应用程序崩溃、电话打入或其他中断应用的事件发生时,当前运行的应用程序会通过切换状态来响应突发事件。

作为 iOS 开发者,需要熟知 iOS 应用的生命周期,在不同的生命周期做出相应的操作,比如 在用户按下 Home 键退出程序时,就需要进行一些数据的备份操作。

1.5.1 iOS 应用程序的状态

iOS应用程序的生命周期共包含Not running未运行、Inactive未激活、Active已激活、Background 后台和 Suspended 挂起 5 种状态,这 5 种状态下的互相转换如图 1-38 所示。



图 1-38 iOS 应用程序生命周期中的 5 种状态

iOS应用程序5种状态的描述如表1-3所示。

表 1-3	iOS 应用程序生命周期的各个状态
-------	-------------------

状态类型	状态说明
未运行(Not running)	应用程序尚未被启动
未激活(Inactive)	应用程序正在前台运行,但是无法接收事件。通常当应用程序从一种状态进入另一 种状态时,中间会在此状态短暂停留,比如用户锁屏或者响应来电时。在没有事件 处理的情况下,应用程序通常处于此状态
己激活(Active)	应用程序正在前台运行,并且可以接收各种事件。应用程序在前台运行时,绝大部 分处于此状态
后台(Background)	应用程序处于后台,并且还在执行某些代码。应用程序在进入挂起状态时,通常会 在此状态停留一会儿。应用程序可以通过一些特殊的请求从而延长处于后台状态的 时间
挂起(Suspended)	应用程序处于后台,并且不能执行任何代码。当挂起时,应用程序仍然处于内存中, 由系统决定何时从内存中清除处于该状态的程序,以提供更多的内存给处于前台的 其他应用程序。当系统发生低内存告警时,会将处于挂起状态的程序从内存中清除

1.5.2 演示 iOS 应用程序的各种状态

每一个 iOS 应用程序都包含一个 UIApplication 对象, iOS 系统通过 UIApplication 对象监控应 用程序生命周期的全过程。

iOS 应用程序都要为UIApplication 对象指定一个代理对象,并由该代理对象处理UIApplication 对象监测到的应用程序生命周期事件。

UIApplication 代理对象的生命周期函数说明如下。

- application:didFinishLaunchingWithOptions 当应用程序载入后执行该方法。
- applicationWillResignActive 当程序将要进入非活动状态时,调用此方法,在此期间,程序 不接收消息或事件。
- applicationDidEnterBackground 当程序被推送到后台的时候,调用此方法。如果要设置当程序进入后台仍然继续某些动作,在这个方法里面添加代码即可。
- applicationWillEnterForeground 当程序将要从后台重新回到前台时调用此方法。
- applicationDidBecomeActive 当程序进入活动状态的时候,执行该方法。
- applicationWillTerminate 当程序将要退出时,调用该方法。通常是用来保存数据和进行一些 退出前的清理工作。

接着在 AppDelegate.swift 文件的各个生命周期方法中添加 print 日志输出方法。当应用程序进入某个状态时,在控制台输出相关的日志,如图 1-39 所示。



图 1-39

接着点击 Xcode 界面右上角的"编译并运行"按钮,打开模拟器运行该项目。此时在控制台 输出相应的日志信息,如图 1-40 所示。

		11 4	\rightarrow	≙ @ ⋞	🥦 第一桶金	
				<	didFinishLaunchingWithOptions applicationDidBecomeActive	\sim
+ 🕞 Filter	Auto 🗘 🕴	0 0 0	Filter		All Output \$	Ŵ DD

图 1-40 应用程序处于活动状态

然后依次点击 Hardward → Home 菜单命令,使模拟器返回 Home Screen,此时日志面板输出的信息如图 1-41 所示。



图 1-41 应用程序被推送到后台

最后在 Home Screen 点击应用程序的图标,再次进入应用程序界面,此时日志面板输出的信息如图 1-42 所示。



图 1-42 应用程序重新处于活动状态

1.6 小 结

本章主要讲解了 iOS 系统的演变历程和 iOS 开发圈子的现状。同时为初次学习 iOS 开发的同 学分析了 iOS 开发的特点,以及如何加入开发者计划,从而获得发布应用程序的资格。此外,还 讨论了如何下载和安装 Xcode 软件,以及创建开发配置文件。 同时,我们通过创建一个简单的项目向读者展示了 iOS 项目从创建至运行的整个流程。虽然 这是一个简单的项目,但是和复杂的项目没有什么区别,它们具有相同的创建、配置、编码、测试 流程。

最后,我们还谈到了应用程序的生命周期。因为熟悉应用程序的生命周期在 iOS 开发中是相当重要的,开发者经常需要在应用程序不同的生命状态下进行不同的业务操作。

通过本章的学习,相信你对 iOS 开发已经有了一个全局和概括的认识,在接下来的章节中, 你将陆续学习到 iOS 开发领域具体的知识点。

第2章

Xcode 9 IDE 和 iOS 模拟器

Xcode 运行于苹果公司的 macOS 操作系统,是苹果公司向开发人员提供的非开源的集成开发 环境,用于开发 macOS、iOS、watchOS 和 tvOS 的应用程序。

本章主要介绍如何使用 Xcode 9 和 iOS 模拟器。在 iOS 开发工作中,90%以上的工作都需要利用这两个工具来完成,它们为开发者提供了设计、编码、调试、发布等功能。

2.1 Xcode 9 的使用

Xcode IDE 是进行 iOS 开发的核心工具。Xcode 与 Cocoa 和 Cocoa Touch 框架紧密集成,为构建精彩的 mac OS、iOS、watchOS 和 tvOS 应用程序带来难以置信的高效环境。Xcode 提供了设计、开发、调试、测试、发布等功能,绝大部分开发工作都可以在 Xcode 中完成。

Apple 公司在 2017 年发布并推出了 Xcode 9.0 版本,当你使用 Xcode 打开一个 iOS 项目时, 它的程序界面如图 2-1 所示。

Xcode 界面共划分为5个区域,每个区域都有各自的主要职能。

- (1) 工具栏 Toolbar: 两侧各提供了一些常用的快捷功能,中间显示项目的各种状态信息。
- (2) 导航器 Navigator: 包含默认的项目导航器以及符号、搜索导航器等 9 个导航器。
- (3) 编辑器 Editor: 拥有设计故事板界面、编写代码文件等功能。
- (4) 工具区 Utility: 包含对象检查器、属性检查器等面板,常用来配置故事板中的元素。
- (5) 调试区 Debug Area: 用来调试应用程序,并显示应用程序输出的日志信息。



图 2-1 Xcode 9.2 用户界面

2.1.1 工具栏的使用

在 Xcode 界面顶部的工具栏中, 左侧包含 Build and then run the current scheme (编译并运行) ▶ 和 Stop the running scheme or application (停止运行) ■ 两个按钮, 你会经常使用这两个按钮 来运行和中止应用程序。

接着是 Set the active scheme 选项 <u>PermoProject</u>, 点击此处时, 会显示测试设备列表, 你可以在列表中选择进行测试的真实设备或者各种类型的模拟器。

工具栏中间的状态条用来动态显示项目的各种状态信息,比如 Xcode 在编译应用程序时,在 此处会显示应用程序当前编译的进度。

工具栏右侧有两组按钮,每一组都包含功能相似的3个按钮。

- Standard editor 按钮 📰 默认处于激活状态,表示 Xcode 会显示拥有一个编辑器的标准视图 模式。
- Assistant editor 按钮 将会以左右相邻的方式同时显示两个编辑器,可以在两个编辑器中显示同一个文件的代码,以对比文件的不同代码区域;或者在一个编辑器中显示故事板文件,而在另一个编辑器中显示故事板元素对应的代码。
- Version editor 按钮 🔄 可以用左右两个编辑器分别显示一份文件的两个不同的历史版本, 方 便你查看和管理文件在诸如 Git 版本管理系统中的历史状态。

工具栏右侧第二组的3个按钮主要用来切换一些功能面板的可见性。

- Hide or show navigator 按钮 🔲 用来切换左侧的导航器面板的可见性。
- Hide or show debug area 按钮 🔲 用来切换底部调试控制台的可见性。
- Hide or show the utilities 按钮 □ 用来切换右侧的工具面板的可见性。

2.1.2 导航器的功能介绍

Xcode 包含 9 个导航器,其中 Project navigator 项目导航器处于默认的显示状态,可以通过点击导航窗口顶部的 9 个图标在导航器之间进行切换,或者使用键盘上的"Command+(1~9)"进入导航器的切换。

这9个导航器功能各有不同,如表2-1所示。

导航器类型	功能说明
项目导航器 🗀	以树状结构的方式展示和管理项目中的代码文件、故事板、多媒体资源以及项目配置文件
源码导航器 🗵	Xcode 9 带来了源码管理导航器, Xcode 9 高度融合了 GitHub, 从现在开始, 在 Xcode 9 中使用 GitHub 进行源码版本的管理将变得十分简单
符号导航器 記	符号导航器显示了项目中的类列表,点击类名左侧的右向箭头可以查看类下的属性和方法 列表。在方法上点击可以进行快速定位
搜索导航器 Q	在搜索导航面板中,允许通过输入关键字来查找项目中的所有文件,找出拥有该关键字的 文档
状况导航器 🥂	状况导航器显示了项目中所有文档中的代码问题。黄色标志表示各种不影响程序运行的小问题,红色标志表示致命错误,需要修复才能使项目运行
测试导航器 🛇	测试面板显示了项目中的所有测试用例
调试导航器 🎫	调试导航器用来显示应用程序在运行时,设备各硬件的运行状态,包含CPU、内存消耗、 硬盘读取、网络访问等数据
断点导航器 □	断点导航器列出了项目中的所有断点
报告导航器 🗐	报告导航器记录了当前项目所有的历史动作

表 2-1 Xcode 9 个导航器功能列表

2.1.3 项目导航器中的内容

在 Xcode 的 9 个导航器中, Project navigator 项目导航器是默 认打开的,也是最常用的,如图 2-2 所示。

它以树状结构的方式展示和管理项目中的所有文件。当使用 Single View Application 模板创建一个新的项目时, Xcode 会在项目 中自动创建一些文件。点击某个文件时, Xcode 会在中间的编辑区 打开该文件。这些文件的用途如表 2-2 所示。

🔻 📓 DemoApp
▼ 🛄 DemoApp
AppDelegate.swift
🛃 ViewController.swift
💽 Main.storyboard
Assets.xcassets
LaunchScreen.storyboard
info.plist
V Products
À DemoApp.app

图 2-2 项目导航器

名称	功能说明
DemoProject 文件夹	以项目名称命名的文件夹,包含项目所有的主体内容,之后往项目中添加的各种文件都将放置在该文件夹下。随着项目内容的增多,你可以在该文件夹下创建子文件 夹或者组,从而对项目文件进行管理

表 2-2 项目导航器主要文件及文件夹功能说明

(续表)

名称	功能说明
AppDelegate.swift	作为应用程序的代理文件,你可以在该文件中管理应用程序生命周期中的各个状态,也可以在此文件中创建一些全局变量
ViewController.swift	视图控制器类,应用程序中显示的内容、各种业务逻辑都是在不同的视图控制器类文件中实现的
Main.storyboard	Apple 倾向于推荐开发者使用 Storyboard 进行应用程序界面的搭建,所以该文件作为应用程序的入口,你可以在 Storyboard 中自由发挥你的界面设计创意
Assets.xcassts	除了用来管理应用程序的图标 App Icon 和启动图片外,还可以管理整个项目中所有的图片资源
LaunchScreen.storyboard	当 iOS 系统加载应用程序时,会首先加载该 Storyboard 文件,作为应用程序的启动页面。你可以在该 Storyboard 文件中对启动页面进行多分辨率的适配工作
Info.plist	项目配置文件包含产品名称、版本号、设备朝向等各种项目属性信息
Products 文件夹	该文件夹包含编译之后的应用程序文件包

2.1.4 创建一个组和 Swift 文件

在 Xcode 中,我们通常使用 Group (组)而不是文件夹来分类和管理项目中的众多文件。组和文件夹是有区别的,项目中的文件夹与 Finder 中的文件夹是一一对应的。在 DemoProject 文件夹上右击,然后选择快捷菜单中的 Show in Finder 命令,就可以在 Finder 中定位 DemoProject 文件夹的位置,如图 2-3 所示。

••• • • • A-	iPh 🔍 🔍	DemoProject		
)? 🆑 ? 🔟 🚿	
DemoProject DemoProject AppDelegate.swift Oper ViewController.swift Aim.storyboard Show Assets.xcasets LaunchScreen DemoProject.app New New	In Finder with Exte hal Editor As File Inspector File 译 Show in Finder e Group Group from Selection	● DemoProject ● DemoProject stodeproj	AppDelegate.swift Assets.xcassets Base.lproj Info.plist ViewController.swift	Ruk
1. 在文件夹上右击 Sort Sort	by Name by Type	11		
+ Find Find	n Selected Groups	5 项, 59.93 GB 可用		

图 2-3 查看 DemoProject 文件夹

组则是逻辑上的文件夹,它在 Finder 也就是电脑磁盘上并不存在。组被用来管理项目中的所 有文件,对文件进行有意义地分类,使项目中的文件在 iOS 开发工作中更利于检索和维护。

创建一个组

要创建一个新组,需在 DemoProject 文件夹上右击,然后选择 New Group with Folder 命令,如 图 2-4 所示。

组创建完成后,组名左侧的图标 和文件夹的图标 稍微有些区别,组名图标的左下角多了 一个三角形的标志。

	Show File Inspector	oject					< 🔺 >
DemoProject DemoProject	New File Add Files to "DemoProject"	Capab	lities Resource Tags	Info	Build Settings	Build Phases	Build Rul
 AppDelegate.sw ViewController.s 	Delete						
Main.storyboard	New Group	Display Name	DemoProject				
Assets.xcasset	New Group with Folder	ndle Identifier	com.coolketang.DemoProject	t			
Info plist	New Group from Selection	Version	1.0				
Products	Sort by Name Sort by Type	Build	1				
	Find in Selected Groups						
	Source Control		🛿 Automatically manage sign	iing			
+ Trilter	Project Navigator Help		Xcode will create and update certificates.	profiles, app 1	Ds, and		

图 2-4 选择 New Group with Folder 命令

组并不是 DemoProject 下的子文件夹,你可以在 Customer 组中右击,然后在快捷菜单中选择 Show in Finder 命令,打开组在 Finder 中的位置,如图 2-5 所示。

□ □ □ □ □ △ ■ DemoProject		□ Der	moProject	? 🔟 🔍 搜索
ViewControll Main.storybc	☆A 此始時 Show in Finder Open win External Editor Open As Show File Inspector	2.1.6Demooject_Debug 2.2.1DemoSimulator	DemoProject DemoProject.xcodeproj	AppDelegate.swift Assets.xcassets Base.lproj CustomerVintroller.swift Info.plist ViewController.swift
LaunchSc Info.plist Products	选择 Show in Finder			
. 在组名上右击	Lew Group	6项, 53	3.69 GB 可用	1001
+ 🗑 Filter	New Group from Selection	Xcode will create certificates.	e and update profiles, app IDs, and	_

图 2-5 DemoProject 文件夹下并没有名为 Customer 的子文件夹

从图 2-5 中可以得知, DemoProject 文件夹下并没有名为 Customer 的子文件夹。

往组中添加类文件

接着往 Customer 组中添加一个 Swift 类文件。你可以在 Customer 组名上右击,然后在弹出的 快捷菜单中选择 New File...命令。

也可以点击项目导航器底部左下角的+号图标,弹出文件导入命令列表,然后选择列表中的 File...命令,如图 2-6 所示。

	DemoProject		,A, D.≎	General	Capabi	ilities	Resource Tags	Info	Build Settings	Build Phases	Build Rule
	Customer		Identity								
	 AppDelegate.swift ViewController.swift 			Dis	play Name	DemoPro	ject				
1.	点击左下角的+号			Bundle	e Identifier	com.coo	ketang.DemoProje	ct			
	LaunchScreen.storyboard				Version	1.0					
	File	- 2	2. 选择	≩ File	命令	1					
	New Group New Group with Folder		Signing								
	Add Files to "DemoProject"	6				Autom Xcode certific	atically manage sig will create and updat ates.	i ning :e profiles, app	IDs, and		

图 2-6 选择 File...命令

当执行 File...命令后,Xcode 自动弹出新文件模板选择窗口。在模板窗口的左侧是 Apple 四大系统的列表。

由于 Xcode 可以为 iOS、watchOS、tvOS 和 macOS 开发应用程序,因此在模板窗口中,在左 侧选择应用程序运行的操作系统环境,这里保持默认的 iOS 系统。

然后选择 iOS 系统下的 Source 选项,表示创建运行在 iOS 系统下的代码文件。

接着在右侧的文件模板中选择 Cocoa Touch Class 模板,并点击下方的 Next 按钮,进入下一步 设置页面,如图 2-7 所示。



图 2-7 选择 Cocoa Touch Class 模板

在 Swift 类文件选项设置窗口中,首先在 Class 输入框内输入待创建的 Swift 类的名称 CustomerViewController,然后在 Subclass of 输入框内输入 UIViewController 作为 Swift 类的父类,同时保持默认的 Language 选项为 Swift,最后点击 Next 按钮,完成 Swift 类的设置,如图 2-8 所示。



图 2-8 输入待创建的 Swift 类的名称 CustomerViewController

最后,在下一个弹出的窗口中保持默认的 Swift 类的存放位置,点击 Create 按钮,完成 Swift 类的创建。你会发现在 Customer 组中多了一个名为 Customer ViewController.swift 的文件,该文件 将自动进入编辑模式,如图 2-9 所示。

如果此时在 CustomerViewController.swift 文件上右击,然后选择快捷菜单中的 Show in Finder 命令,你会发现该文件仍处于 DemoProject 文件夹下。

	1 // 2 // CustomerVi 3 // DemoProjec 4 // 5 // Created by 6 // Copyright 7 // 8 9 import UIKit 10 class Customer 12 override f- 5 super.	 ↑人收置 Macintosh HD 予約所有文件 例 AirDrop 企 印程序 ○ iCloud Drive □ Desktop ① 文稿 ① 下载 	DemoProject DemoPrcodeproj	AppDelegate.swift Assets.xcassets Bace.lproj CustomerViewController.swift Info.pust
page DemoProject.app	15 16 // Do		选择了 1 项(共 6 项),59.0	9 GB 可用

图 2-9 CustomerViewController.swift 文件

2.1.5 在项目中搜索文件

随着项目的深入,工程中的各种组、类文件、资源文件的数量越来越庞大。这时在项目中快速定位目标文件变得越来越困难。Xcode充分认识到这一点,并提供给开发者4个方便、快捷的功能,以辅助开发者快速查找、定位目标文件或代码段。

通过名称过滤文件

在项目导航器的底部有一个 Show files with matching name 文件名的过滤框,你可以在此处输入待查找的文件的一部分名称,快速过滤项目中名称包含指定输入内容的文件。

如图 2-10 所示,在输入框内输入 view,然后过滤出名称中包含 view 的文件,并且 view 以高 亮的方式显示在文件名称中。

V 📓 DemoProject	□ A D. ≎	General Capabilities	Resource Tags	Info	Build Settings	Build Phases	Build Rule
V DemoProject							
Customer Customer <u>View</u> Controller.swift	• Identity						- 1
ViewController.swift		Display Name	DemoProject				
		Bundle Identifier	com.coolketang.Dem	oProject			
		Version	1.0				
		Build	1				

图 2-10 使用文件名过滤框快速查找文件

文件名过滤框的右侧有3个按钮,它们的作用依次是:

- 取消过滤按钮 🛽 用来取消对项目文件按照名称的过滤操作。
- Show only recent files 按钮 ① 用来显示或隐藏最近打开或使用过的文件。当点击此按钮时, 只显示最近使用过的文件,再次点击则取消对历史文件的过滤。
- Show only recent files with source-control status 按钮 区 用来显示或隐藏拥有版本控件状态的 文件。当点击此按钮时,将显示被编辑后尚未被提交至版本控制中心的文件,再次点击此按 钮,则取消对版本状态的过滤。

使用跳转栏

在编辑器的上方是跳转栏 Jump bar,使用跳转栏可以快速定位处于任意层级的属性、方法或 类文件。它提供了一种分段的层次关系的文件路径,如图 2-11 所示。

```
🔡 🤇 🔰 DemoProject > 🛅 DemoProject > 🛅 Customer > 🔄 Customer ViewController.swift > 💽 CustomerViewController
```

图 2-11

当你在某个层级的分段处点击时,可以显示该层级的项目列表,如图 2-12 所示。



图 2-12 显示该层级的项目列表

在项目列表中点击某个文件即可快速打开并跳转到该文件。这里还有一个小技巧,当你在按 住 Option 键的同时,点击项目列表中的某个文件可以在辅助编辑器中打开该文件,这样就可以方 便对比两个代码文件,或者进行 Storyboard 元素和代码的链接,如图 2-13 所示。



图 2-13 快速打开并跳转到该文件

在跳转栏最左侧是一个名为 Related items 的按钮 🔀 ,点击它可以显示最近打开的文件列表,如果你正在使用 Objective-C 开发项目,它也可以显示当前正在编辑的实现文件(.m)对应的接口 文件(.h),如图 2-14 所示。

在文件列表中点击某个文件即可快速打开该文件。

	🔡 < 🗦 📓 DemoProj	ect 🔪 🛄 D	ect 👌 📰 Cuer 👌 📓 CustomerViewController.swift 🤇 💽 CustomerViewController	< 🕰
🔻 📓 DemoProject	Recent Files	•	💽 Main.storyboard (Base)	
V DemoProject	Counterparts (2)	•	WiewController.swift	
Customer	0		CustomerViewController.swift (Interface)	
CustomerViewController.swift	Superclasses (3)	•	CustomerViewController.swift	
AppDelegate.swift	Siblings (18)	•	DemoProject	
ViewController.swift	Extensions		Assets.xcassets eserved.	
Main.storyboard	Categories		Clear Menu	
S Assets.xcassets	Protocols		Number of Recent Files	
LaunchScreen.storyboard	User Interfaces			
📄 Info.plist	Includes		ntroller: UIViewController (
Products	Included By			

图 2-14 Related items

在 Related items 按钮的右侧是两个方向箭头,点击这两个箭头可以在最近使用的历史文件中进行快速跳转。如果你使用苹果笔记本进行开发,也可以在触摸板上按下两个手指,然后通过左右 滑动在历史文件中进行快速切换。

使用 Open Quickly 功能

你可以通过 File → Open Quickly 命令使用该功能, 推荐使用 Command + Shift + O 快捷键快速 调出该功能,这是笔者最喜欢的一个功能。和 Eclipse 软件的快速查找功能相似,你可以在搜索框 内输入文件的部分名称,快速查找并打开你想要编辑的文件,如图 2-15 所示。

或者在输入框内输入方法的名称,搜索包含该方法的类文件,然后在列表中双击,快速跳转 到目标代码段,如图 2-16 所示。



图 2-15 快速查找并打开想要编辑的文件



图 2-16 输入方法的名称

使用搜索导航器

在搜索导航器中,允许你通过输入关键字来查找项目中的所有文件,找出拥有该关键字的文 档或者代码段。通过点击导航器顶部的 Q 按钮进入搜索导航器面板,然后在搜索框内输入进行 搜索的关键词,接着按下键盘上的 Enter 键即可。

如图 2-17 所示,在搜索框内输入 viewdidload 关键词,然后按 Enter 键,搜索出在两个文件中的 4 个结果,在其中一个搜索结果中点击,即可快速跳转到指定的代码段。



图 2-17 在搜索框内输入 viewdidload 进行搜索

2.1.6 代码的查找和替换

在使用 Objective-C 语言进行开发的项目中,如果需要更改一个方法的名称,可以使用 Edit \rightarrow Refactor \rightarrow Rename 菜单命令对该方法及调用该方法的地方进行方法名称的批量修改。而对于 Swift 语言来说,在 Xode 8 之前是无法使用 Rename 功能的。Xcode 9 重新带来的 Refactor 功能位置则移 到 Edit \rightarrow Refactor \rightarrow Rename。此外,我们还可以使用 Xcode 提供的 Find and Replace...功能来实现 文字替换。

要调用查找和替换功能,可以依次点击 Find → Find and Replace...菜单命令,或者使用 Command + Option + F 快捷键在编辑器的上方打开查找和替换面板。当你在上方的文本框内输入需 要查找的内容时,Xcode 将在输入文字的同时自动查找并高亮显示搜索结果。然后在下方的文本框 内输入替换为的文字,接着点击右侧的 All 按钮,将文件中的所有 addImage 文字全部替换为 addCompanyLogo,如图 2-18 所示。



图 2-18 代码的查找和替换

2.1.7 代码的 Debug 调试

在 iOS 开发工作中,你会经常使用到 Xcode 软件的 Debug area 进行程序的调试,查找一些变量的状态或者诊断出现的 Bug。

步骤① 在使用 Debug 功能之前,首先在 ViewController.swift 文件中添加一个名为 appendThreeViews 的方法,用来往当前视图控制器的根视图中添加3个 UIView 子视图,代码如下:

```
2
    func appendThreeViews()
3
   {
 4
         let firstView = UIView(frame: CGRect(x: 40, y: 80, width: 100, height: 100))
 5
         firstView.backgroundColor = UIColor.green
         self.view.addSubview(firstView)
 6
 7
 8
         let secondView = UIView(frame: CGRect(x: 40, y: 80, width: 120, height: 120))
 9
         secondView.backgroundColor = UIColor.blue
10
         self.view.addSubview(secondView)
11
12
         let thirdView = UIView(frame: CGRect(x: 40, y: 80, width: 140, height: 140))
13
         thirdView.backgroundColor = UIColor.purple
```

14 self.view.addSubview(thirdView)

15 }

在之后的章节会为读者讲解 UIView 视图的使用,这里你只需要了解这些代码的作用,依次创 建了三个位于相同位置(40,80)的视图。它们拥有不同的背景颜色,从第一个视图至第三个视图依 次是绿色、蓝色和紫色。这三个视图也拥有不同的尺寸,从第一个视图至第三个视图尺寸依次变大, 所以第三个紫色的视图会遮挡前两个视图。

(步骤 2) 在 viewDidLoad 方法中调用 appendThreeViews 方法:

```
16 override func viewDidLoad() {
```

- 17 super.viewDidLoad()
- 18 // Do any additional setup after loading the view, typically from a nib.
- 19 appendThreeViews()

20 }

步骤 ③ 在第 23 行代码的行号上点击,添加一个断点标志,当程序运行到此行代码时,会自动 停留在这个位置,如图 2-19 所示。



图 2-19 添加一个断点标志

步骤 ④ 点击 Xcode 界面左上角的"编译并运行"按钮 ▶ ,打开模拟器调试项目,项目运行 后,将自动停留在第 28 行代码中,并在底部的调试区显示调试面板。调试面板的上方是一排调试按 钮,左侧是变量信息列表,右侧是日志区。如果需要查看 firstView 对象的属性,可以在日志区 (lldb) 右侧输入 po 命令 (print object 的缩写) 和变量的名称,然后按键盘上的 Enter 键,查看对象的描述信 息,如图 2-20 所示。



图 2-20 查看对象的描述信息

你可以使用调试区顶部的一排按钮进行程序的调试,调试按钮的功能如表 2-3 所示。

调试按钮	功能说明				
调试区 🔽	隐藏调试区面板				
断点 🍉	激活所有断点,或使所有断点暂时失效				
继续 □▷	跳过所有断点,继续程序的执行				
跳转 🛆	跳转至下一行代码				
跳入 🛓	跳入所在行的方法内的第一行代码				
跳出 1	从一个方法中跳出				
视图层级 🕕	在三维视图下显示窗口中视图的层级关系				
内存图形 💝	展示内存分类的顺序和多少,如图 2-21 所示				
地理位置 🗸	在模拟器中模拟地理位置的定位				

表 2-3 调试按钮功能说明

步骤 ⑤ 分别使用 △、 之、 1 3 个按钮进行程序的调试,最后点击 I> 按钮,结束程序的 调试。在这些按钮中,你可以点击 Debug memory graph 按钮 ≫ 查看当前页面内存的分配情况,如 图 2-21 所示,选择左侧的 ViewController 项目,在右侧显示该项目的内存分配状况。



图 2-21 项目的内存分配状况

步骤 6 另外,一个实用的工具是 Debug View Hierarchy 🕕 按钮。做 iOS 移动开发,界面管理 是很重要的内容,如果想要查看伙伴制作的页面的元素结构,或者查看自定义控件的构成,就可以使 用该工具。

如图 2-22 所示,点击 Orient to 3D 按钮,转换为三维视图模式,然后拖动左侧的 Adjusting the spacing between view 滑竿调整各视图在三维坐标系统 Z 轴的距离。



图 2-22 Debug View Hierarchy

通过拖动右侧的 Adjust the range of visible view 滑竿可以从前往后或者从后往前逐个隐藏窗口中的视图,以方便进行界面上的调试。

2.2 iOS 模拟器的使用

当你在使用 Xcode 软件开发 iOS 平台的应用程序时,可以使用 Apple 提供的 iOS 模拟器进行 应用程序的测试。Apple 提供了各个版本的 iPhone 和 iPad 模拟器供开发者测试应用程序,这样你 就无须花费大量的金钱去购买测试设备。

尽管 iOS 模拟器无法模拟所有的真机功能,但是它可以提供真实设备绝大部分的功能,所以 iOS 模拟器在应用程序开发工作中的地位非常重要。

2.2.1 使用多种 iOS 模拟器

要使用 iOS 模拟器,首先打开本书附带的示例项目,在示例项目中添加一个 UIImageView 图 像视图。点击 Xcode 界面左上角的"编译并运行"按钮 ▶ ,打开模拟器之后,模拟器窗口中将 显示一幅图像,如图 2-23 所示。

由于当前使用 Xcode 9 运行项目,因此系统启动了 iPhone SE 模拟器。如果需要使用其他的模 拟器,可以点击 Set the active scheme 选项 A DemoProject J Hone SE 弹出测试设备列表,如图 2-24 所示。测试设备列表拥有 4 个分组,每个分组的含义如下:





 在位于顶部的 Device 分组中列出了所有连接 Mac 计算机的 iOS 真机设备,如果你需要在真 机中测试程序,可以选择该组中的真实设备。

图 2-24 测试设备列表

- 如果你需要将应用程序编译并发布至 App Store 或面向 企业内部的分发,需要选择 Build Only Device 分组中的 选项。
- 在 iOS Simulators 分组中列出了所有可以使用的各种版本的 iOS 模拟器。
- 在位于底部的分组中,允许安装 Xcode 目前不具备的模 拟器。

在 iOS Simulators 分组中,选择 iPad Pro (9.7-inch) 模拟器,然后点击"编译并运行"按钮 ▶,将切换至 iPad Pro 9.7-inch 模拟器,并在该模拟器中重新安装并打开当前的应用程序,如图 2-25 所示。iPad Retina 版本以上的设备由于分辨率较高,往往在 Mac 电脑中占据很大一块面积,如果影响到了工作,你可以使用 Window → Physical Size 命令将其缩小至原来



的一半大小。或者将鼠标放置在模拟器的四角,然后按下鼠标 图 2-25 iPad Pro (9.7-inch)模拟器 并拖动以自由调整模拟器的大小。Xcode 9 的模拟器在默认情况下是带有设备边框的,如果需要模 拟器像 Xcode 8 那样只显示应用程序的屏幕,可以取消勾选 Window → Show Device Bezels 选项。

另外,也可以使用 Hardware → Device 命令切换至其他模拟器。

在模拟器中,你可以使用鼠标替换手指进行操作,如同在真实设备上操作一样使用点击、滑动等手势。如果使用多点手指,可以按下键盘上的 Option 键,此时将出现两个灰色透明圆点,这样就可以使用双指捏合等操作了。如果需要改变两个圆点的位置,保持 Option 键的按下状态,再次按下 Shift 键,即可同时移动两个圆点的位置。

2.2.2 iOS 模拟器方向的切换

如果你的应用程序支持多朝向,可以对模拟器执行旋转操作,以测试应用程序界面在 4 个朝 向的变化。依次点击 Hardware → Rotate Left 或 Hardware → Rotate Right 菜单命令,对模拟器进行 向左或向右旋转,也可以使用 "Command + 左向箭头"或 "Command + 右向箭头"快捷键对模拟 器进行旋转,模拟器旋转 90 度的效果如图 2-26 所示。



图 2-26 模拟器旋转 90 度

2.2.3 iOS 模拟器语言的切换

iOS 模拟器默认为英文环境,如图 2-27 所示。如果需要更改模拟器的语言,以进行应用程序本地化的测试,可以在模拟器中依次点击 Settings → General → Language & Region → iPhone Language 命令,进入语言列表页面,然后在列表中选择"简体中文"选项,接着点击右上角的 Done 按钮,将 iPhone 模拟器调整为简体中文,如图 2-28 所示,调整后的效果如图 2-29 所示。



图 2-27 默认为英文环境

图 2-28 调整语言

图 2-29 调整为简体中文的效果

2.2.4 截取模拟器屏幕

当你向 App Store 提交应用程序打包后的产品时,需要在 iTunes Connect 中上传应用程序在各种设备中的截图。由于 Apple 对应用程序截图的尺寸进行了限制,因此最好直接截取 iOS 模拟器中的界面。

截取模拟器的界面非常简单, 依次点击 File → New Screen Shot 菜单命令, 或者使用键盘上的 Command + S 快捷键, 模拟器的截图将默认保存在桌面位置, 如图 2-30 所示。



图 2-30 截取模拟器的界面

2.2.5 应用程序的退出和删除

在模拟器打开应用程序之前,Xcode 已经将该应用程序安装在 iOS 模拟器中,就像在真实设 备上安装应用程序一样。如果需要在模拟器中退出当前的应用程序,可以点击模拟器底部的 Home 键,或者使用 Hardware → Home 命令,也可以使用 Command + Shift + H 快捷键。

在进行应用程序测试工作时,我们经常需要将应用程序从内存中清除。如果要从内存中移除 该应用程序,可以双击 Home 键,或者在按下 Command + Shift 键的同时,快速按两下 H 键,即可 进入后台程序列表界面,如图 2-31 所示。然后在应用程序的缩略图上按下手指,并向上方拖动, 即可将应用程序从内存中移除。

将应用程序从内存中清除后,该应用程序仍然保留在模拟器中,直至你删除它。要删除模拟 器中的某个应用程序,可以使用鼠标长按该应用程序的图标,以进入删除模式,就像在真机上操作 一样,如图 2-32 所示。然后点击图标左上角的删除按钮,即可删除该程序。

如果需要删除在模拟器中由开发者安装的所有应用程序,并将模拟器恢复至默认状态,可以 使用 Hardware → Erase All Content and Settings...命令,然后在弹出的警告窗口中点击 Erase 按钮。 重置模拟器后,除模拟器自带的应用程序外,其他由开发者编译安装的应用程序都将被清除,如图 2-33 所示。



图 2-31 进入后台程序 列表界面



图 2-32 点击图标左上角的删除按钮, 图 2-33 清除所有自行安装的 即可删除该程序



应用程序

2.3 小 结

本章讲解了 Xcode 9 及其自带的 iOS 模拟器的使用。读者可通过本章学到 Xcode 9 各个功能区 域的使用,以及查找、替换、调试等各种实用的功能。

通过学习 iPhone 及 iPad 模拟器的使用,可以方便、快捷地对应用程序进行测试操作,而无须 连接真实的设备。

磨刀不误砍柴工,熟练使用这两个开发和调试工具,在如今强调敏捷开发的时代显得尤为重 要。如果你能够记下 Xcode 的各种常用快捷键,那就再好不过了。