

覆盖设计、原型、开发、上架完整知识
iOS 设计/开发疑难问题图文解析
从 0 到 1 做出属于自己的 iOS 应用

Broadview[®]
www.broadview.com.cn



自己动手做 iOS App

从设计开发到上架 App Store

张子怡 著

—— Sketch 44 + iOS 10 + Swift 3.1 ——



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



自己动手做 iOS App

从设计开发到上架 App Store

张子怡 著

电子工业出版社

Publishing House of Electronics Industry

北京•BEIJING

内 容 简 介

本书为想要接触 iOS 应用设计、开发的读者提供了由浅入深的详细指导。从 iOS 应用制作的步骤是什么,应该使用什么软件,如何发布应用到 App Store,到 iOS 的设计理念是什么,如何正确书写 Swift 语言,再到后端和客户端是如何交互运作的等,本书配合图示,精辟、直观地阐明了 iOS 应用制作中的种种疑问。

如果你是一位第一次接触 iOS 应用制作的新手,那么读完本书你将会充满信心地着手把自己的想法带到现实。即使是有一定经验的 iOS 设计师也可以学到如何编写代码,程序员则可以学习到设计知识,并都能够获得独立完成个人作品的 ability。

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

自己动手做 iOS App:从设计开发到上架 App Store / 张子怡著. —北京:电子工业出版社,2017.8
ISBN 978-7-121-32019-4

I. ①自… 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 140710 号

责任编辑:黄爱萍

印 刷:北京千鹤印刷有限公司

装 订:北京千鹤印刷有限公司

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:720×1000 1/16 印张:11.5 字数:140 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 zltts@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。

献给我的父母和妻子同同
爱是一切作品的根源

「前言」

关于这本书

这几年 Apple 产品的迭代速度变快，连带着对应用制作需要学习的知识更新也加快了。连有经验的制作者都感叹今天学习的知识很快就用不上了，对于新人来说，恐怕更不知道如何下手接触应用制作。但是即使知识更迭得再快，只要沉淀下基本的构建思想和制作过程，便能以不变应万变，一步步深入地学习应用制作技能。

我喜欢把做软件应用比作设计制造一辆汽车，有各种专业书籍对怎么设计车的外型、用什么材料、发动机应该怎么组装、以及车内该如何布置等详细讲解，但是对于一个新人来说，这些书籍摆在他面前，他也无法知道从何开始，即使把发动机部分学得滚瓜烂熟，去实际生产一辆车恐怕也是障碍重重。本书将需要用到什么工具、如何设计、如何开发、如何发布到市场等知识贯穿起来，手把手地带领新人从零开始，我想从这个角度切入 iOS 应用制作，来得更实际一些。

2016 年年底我设计、开发并上架了一款名为 Oslo 的 iOS 应用，可以在 <https://itunes.apple.com/us/app/oslo-mobile-unsplash/id1184226442?mt=8>，或者在 App Store 中搜索 **Oslo Mobile Unsplash** 下载。这款应用的 UI 设计是 Sketch，应用图标设计是 Affinity Designer，开发环境是 Xcode 8.2.1，开发语言是 Swift 3.0.2。我相信边学习、边实践是最快、最有效的途径，因此这本书会以这款应用为案例，带你一步一步制作一款最终上架到 App Store 的应用。学习过程中不但涉及设计或者开发方面的概念讲解，同时还结合了实际的制作，让你对 iOS 的应用制作有更深刻的了解。

当你跟随这本书全部做下来后，你将会充满信心地做出属于自己的 iOS 应用，当初那种对设计缺乏自信，对编程敬而远之的心情将不复存在。所以，上路吧！

谁需要读这本书

如果你是一名没有接触过 iOS 应用制作的新人，你是否有过这样的问题：

- 我该从哪里入手学习 iOS 应用制作，哪些资源是针对初学者的，又如何入门？
- 要使用哪些工具，如何快速上手？

... ..

如果你是一名设计师，你是否有过这样的问题：

- 我应该用什么规格的画布来做 UI，应该导出什么样尺寸的图标应用到开发中？
- 怎样才能快速填充好各种头像？
- 怎样做快速原型（Fast Prototype）？
- 看到代码就害怕，即使学习了也无法应用到实际中，该怎么办？

... ..

如果你是一名工程师，你是否有过这样的问题：

- 怎样通过 Storyboard 快速、简单地实现多屏幕适配？
- 对色彩和绘图完全不在行，这样如何做设计？
- 我想了解一些最新的语法和开发环境的特性。

... ..

所以无论是入门者或专业人士，都可以从本书吸取到适合自己的新的知识，或者对平常模棱两可的问题得到答案。当然如果你只是初次接触 iOS 应用制作，那么这本书会由浅入深，一步一步带你拓展和提升，最终获得不输给专业人士的能力。同时，本书适合有热情制作 iOS 应用的所有人。

使用到的工具

本书使用到的硬件有 MacBook、iMac、Mac mini 三种，使用系统为 macOS。



Xcode 8.3.3

Xcode 是 Apple 系列产品开发的主要工具，同时包含了 Swift。在 **Mac App Store** 中搜索 **Xcode** 下载，或者在 <https://developer.apple.com/download/> 下载测试版。书中使用的版本是 Xcode 8.3.3。注意，Xcode 8.0 才包含 Swift 3，这也是书中主要使用的程序语言。



Sketch 44.1

Sketch 帮助快速实现应用原型，也能满足界面要求较高的制作。在 <http://sketchapp.com/> 下载。书中使用的版本是 Sketch 44.1。



Swift 3.1

Swift 为书中使用的编程语言。同时也是 Apple 系列产品开发的主要编程语言。Swift 还在不断演化过程中，可以在 <https://swift.org/> 了解学习。正式版 Xcode 包含了最新的稳定版 Swift，如果想体验测试版的 Swift，可以下载测试版 Xcode。书中所使用的版本是 3.1。

本书主要使用到的工具就是以上这些，在学习过程中需要使用到其他软件时会单独说明。

设计资源和源代码

在章节讲解的过程中，会指出资源的下载地址。这些资源能够自由用于个人或商业用途，用于公开演说或者教育用途时，希望能够注明来源。

源代码可以在 <https://github.com/hipposan/Oslo> 获取，但不能用于出售或其他交易。

随着 Sketch、Xcode 和 Swift 的不断更新换代，对于可能出现的制作方式及语法的更新，我也会尽自己所能更新本书，同时更新下载地址中的资源。

勘误和反馈

对于书中出现的任何错误，或者在使用过程中有不明白的地方，可以发送邮件到 zzy0600@gmail.com，我会认真查看每一封信件，希望能和大家多多交流。

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/32019>

「目 录」

设计	1
Sketch 介绍	2
- 模板 Template -	6
- Artboard -	8
- 导航栏 Navigation Bar -	10
- Craft -	12
- 字体 font -	14
- 对齐 alignment -	16
- 用 Profile 和 Me 界面来练习 -	18
- 同一界面的多种不同状态 -	22
- 导出 export -	31
- 在“真机”上看到自己的设计 Mirror -	33
有用的 Sketch 知识和技巧	34
- 图形的“组合加减” -	34
- 文字操作 -	35
- Alpha Mask -	36
- 插件 plugin -	37
用 inVision 来制作原型和管理设计 (Bonus)	38
原型	42
Xcode 介绍	43
第一次 Build	44

Storyboard	49
- 组件 Component -	49
- Table View Controller -	51
- Navigation Bar -	53
- Table View Cell -	55
- Preview -	57
- Visual Effect View 和 Web View -	60
- Collection View -	62
- Container View -	65
- Stack View -	66
Auto Layout	69
- 对齐 & 间距 Alignment& Spacing -	69
- Table View 的 Auto Layout -	72
- 多个元素的居中 -	74
- 用 Photo 界面练习 -	77
- 连接 Storyboard -	79
编程	83
Swift 介绍	84
用代码控制界面	85
- 关联 Storyboard 和代码文件 -	85
- 连接组件到代码中 -	87
- Protocol -	89
- 自适应高度 -	91
- Collection View -	93
- 定义组件事件 -	95
- 触发 Segue -	97
- Delegate -	98
- xib-	101
- App Security -	104
- UIActivity -	107

- @IBInspectable -	107
网络	110
- Client & Server -	110
- 通信 -	111
- HTTP Request Methods -	112
- API -	112
- JSON -	115
- 储存 API 信息 -	116
- 建立网络层 -	117
- MVC -	121
- Grand Central Dispatch & OperationQueue -	126
- 缓存 -	127
- 下拉刷新 & 划动加载 -	129
- 用 Segue 传输数据 -	132
- 更新 xib 信息 -	137
- OAuth 2 与登录 -	141
- UserDefaults -	146
- POST -	147
- 用 delegate 来传输数据 -	149
其他	157
- 动画 -	157
- 本地化语言 -	161
- 提交 TestFlight 测试 -	165
- 提交到 App Store 审核 -	170

设计

用 Sketch 制作精准像素 (pixel perfect)，可直接用于开发的设计。



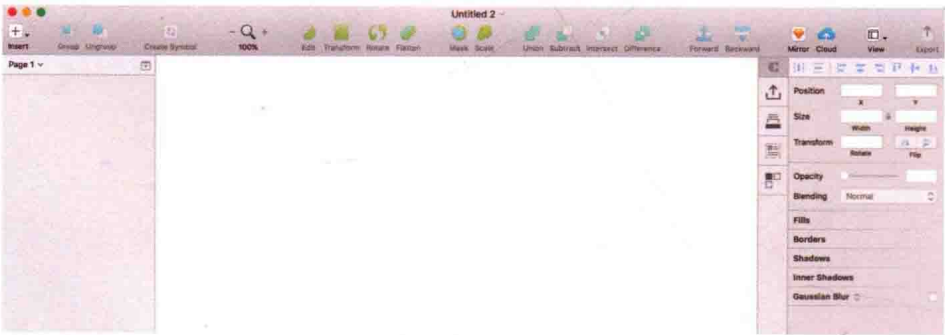
从现在开始，我们将会从 0 到 1，设计并开发一款名为 Oslo 的 Unsplash¹ 第三方客户端。在这个过程中边做边学习设计理念、工具使用、程序编写等知识。如果你做好了准备，那我们就开始吧！

Sketch 介绍

早些年大家做设计时基本都用 Photoshop。但 Photoshop 毕竟是一款作图软件，在做 Web/App UI 设计，尤其是原型设计时，通常不需要用到精细的图片打磨或者多样的笔刷，反而是合适的画布选择和方便的资源（assets）导出等是首选需要。因此对于新手来说，打开 Photoshop 后要先去了解大量的非常用功能，让人十分头疼。

由于这个原因，Sketch 出现了。Sketch 在作图方面与 Phtoshop 的定位不同，Sketch 定位于 Web/App UI 设计，因此对于这部分设计人员来说更易上手，使用起来也更加直接，并且完全是矢量图，于是很多设计师马上从 Photoshop 转移到了 Sketch 阵地。

安装好 Sketch 后，请打开 Sketch，选择 New Document，展现的界面如下图所示。



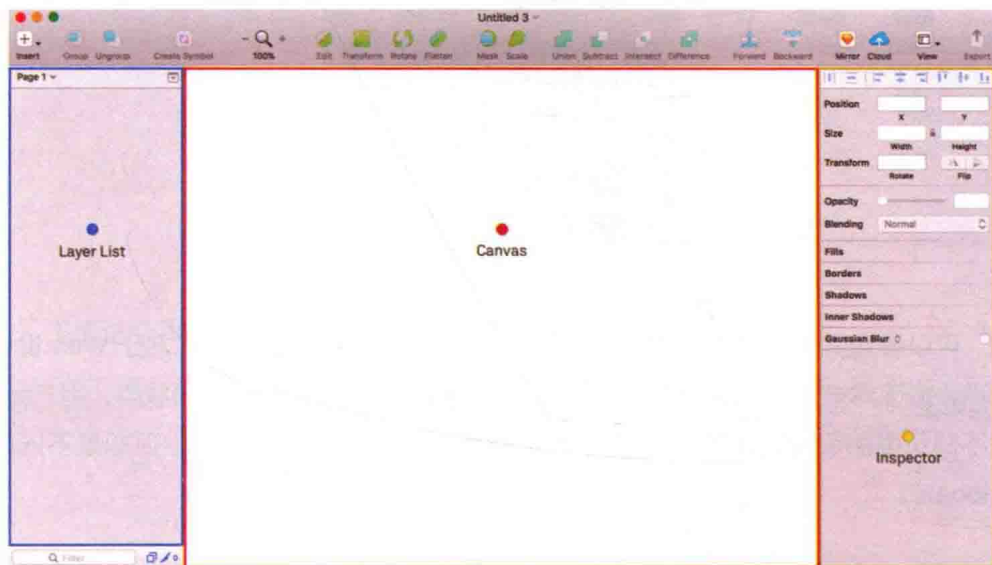
首先是顶部的 Toolbar²，如下图所示。

1 Unsplash 是一个高清摄影的分享平台，里面的图片可以在个人或商业作品中自由使用。
2 由于使用的工具为英文版，在网上搜索的资料也对应为英文，因此之后的专业名词大多以英文出现。



工具栏主要是一些常用的快捷操作，可以通过单击右键来进行自定义。在实际操作过程中，用顶部菜单、右键菜单或者快捷键也能达到相应目的。

接下来的三个区域是在实际项目中最为常用的。由左开始，分别是 Layer List, Canvas 和 Inspector，如下图所示。

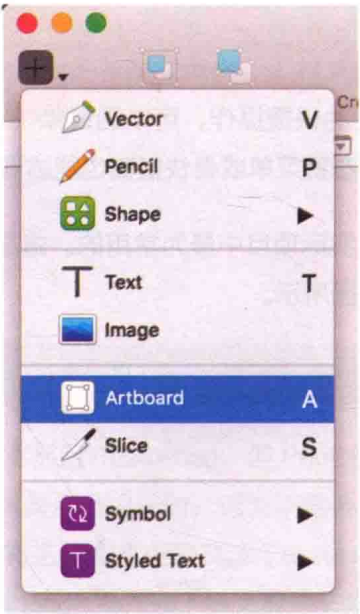


Layer List 是图层的结构，类似于文件目录。这里需要注意的是 Page 和 Artboard。

按下图顺序单击可以新建一个 Page。

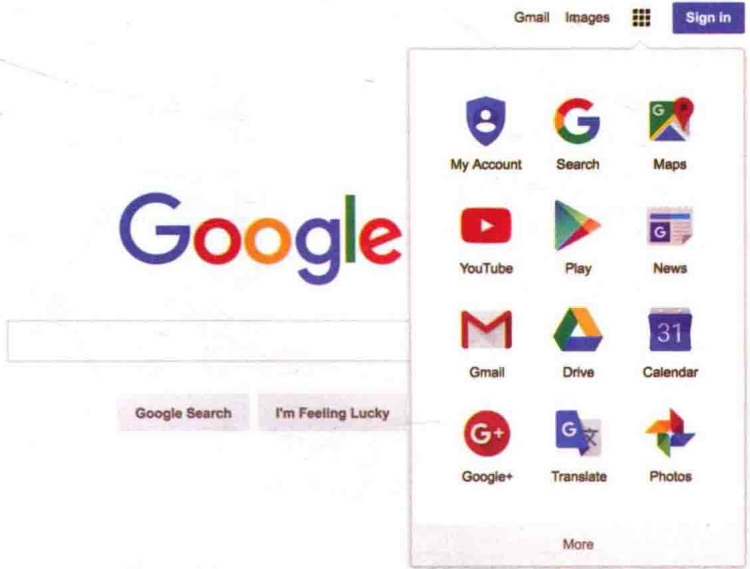


单击下图中的 Artboard，然后在中间 Canvas 区域随意拖曳会新建一个 Artboard。

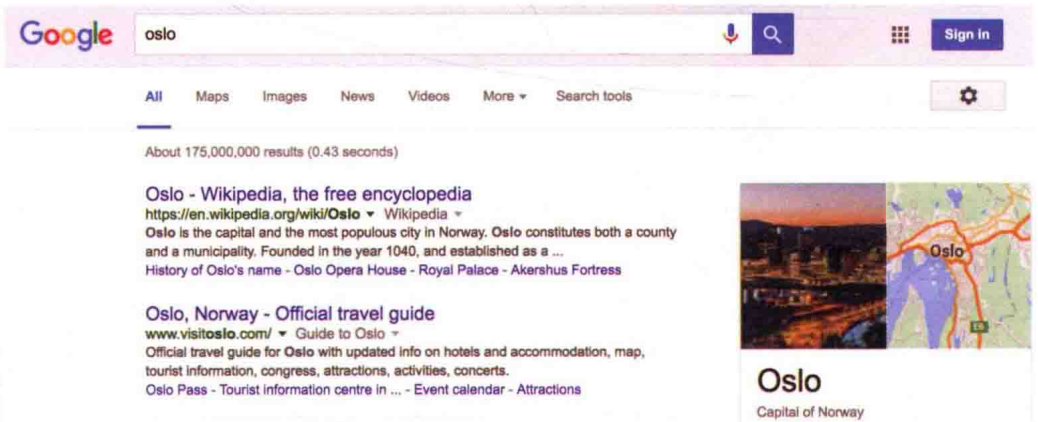


可以看到 Artboard 是从属于 Page 的，它们的关系是怎样的呢？对于 Web 设计来说，由于同一页面不同状态较多，用 Artboard 做同一页面间的不同状态，用 Page 做不同页面的设计，比如下图的几个页面应该都属于同一个 Page，但却是不同的 Artboard。

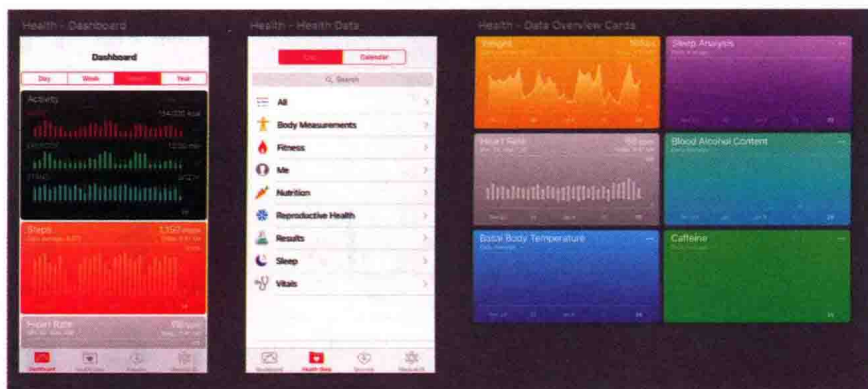




下面这个页面应该属于不同的 Page，不同的 Artboard。



对于 App 设计来说，由于交互复杂，所以大多数情况下都在一个 Page 内，而使用不同的 Artboard 来完成，如下图所示。



使用不同的 Page 完成各种不同的设计用途，比如 Page 1 是 App 设计，Page 2 是 App Store 截图，Page 3 是 App Icon 设计等。

Canvas 是主要的作图区域，也就是中间部分，所有的设计将在这里完成。右侧的 Inspector 用来调整元素属性。

Sketch 整体结构就是这样，很简单吧。

- 模板 Template -

在 Sketch 刚推出的时候，没有什么模板可以用。设计者需要去看 iOS Human Interface Guidelines，学习 iOS 上边距设置多少是合理的，字体应该怎么用，适合手指大小的单击区域是多大等知识，然后才能开始自己的设计。无可厚非的 iOS Human Interface Guidelines 融合了很多设计理念，是设计者们的思维结晶，但是学习到实际应用的时间还是有点长。

模板为设计者提供了更快捷的学习方法。比如 Great Simple Studio 推出的 iOS 10 GUI (<http://ios10.greatsimple.io/>) 。

这些模板都遵照 iOS Human Interface Guidelines 的标准规范，并且附带很多原生的 iOS 界面。我们可以从里面直接可视化地学到很多实用知识，以 Great Simple Studio 的 iOS 10 GUI 为例，下载 iOS10 GUI 后打开 Sketch 文件，具体如下。

比如 iOS 的状态栏 (Status bar) 设置宽度为 375，高度为 20。