

一线程序员撰写，凝聚自己多年开发经验结晶，系统且深入阐释Objective-C和Sprite Kit
游戏开发涉及的工具、方法和实践



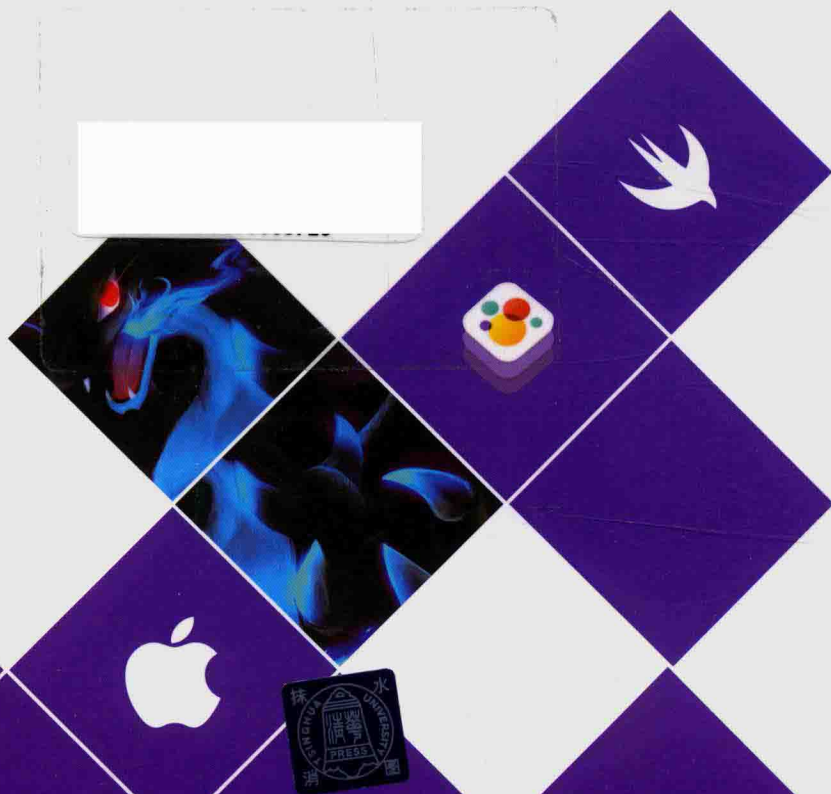
由浅入深剖析Objective-C和Sprite Kit游戏开发过程中遇到的各个层面的问题，涉及
Xcode、Objective-C、Foundation框架、Sprite Kit以及游戏开发相关技术等

Objective-C & Sprite Kit
Game Development Introductory Tutorial

Objective-C和Sprite Kit

游戏开发从入门到精通

曹化宇◎著
Cao Hua Yu



清华大学出版社



Objective-C & Sprite Kit
Game Development Introductory Tutorial

Objective-C 和 Sprite Kit

游戏开发从入门到精通

曹化宇◎著
Cao Hua Yu

清华大学出版社
北京

内 容 简 介

本书是一线程序员凝聚自己多年开发经验的结晶之作，深入浅出地讲解Objective-C语言、Foundation框架、Sprite Kit框架，以及其他开发iOS和Mac游戏所需要的基础知识，帮助读者零基础实践iOS和Mac游戏开发。

本书内容可以分为四个部分，第一部分（第1~5章）介绍如何使用Xcode和Objective-C语言编写程序；第二部分（第6~12章）讨论Foundation框架中常用的开发资源，这些资源不但可以为游戏开发服务，同样也可以应用于各种类型的iOS或OS X应用开发；第三部分（第13~18章）讨论Sprite Kit在游戏开发中的应用；第四部分（第19~23章）讨论游戏开发相关的技术，如网络状态检测、Game Center排行榜、应用内购买等。

本书内容安排合理，架构清晰，注重理论与实践相结合，适合作为零基础学习iOS和Mac游戏开发的初学者的教程，也适合作为有一定编程基础的程序员的参考手册。

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

Objective-C和Sprite Kit游戏开发从入门到精通 / 曹化字著. — 北京：清华大学出版社，2017
ISBN 978-7-302-45370-3

I. ①O… II. ①曹… III. ①游戏程序—程序设计 IV. ①TP317.6

中国版本图书馆 CIP 数据核字(2016)第 260827 号

责任编辑：杨如林 秦 健

封面设计：李召霞

责任校对：胡伟民

责任印制：王静怡

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：北京鑫海金澳胶印有限公司

经 销：全国新华书店

开 本：186mm×240mm

印 张：19.25

字 数：472 千字

版 次：2017 年 1 月第 1 版

印 次：2017 年 1 月第 1 次印刷

印 数：1 ~ 3500

定 价：59.00 元

产品编号：071828-01

为什么要写这本书

这是一个真实的故事。

儿时与兄弟们玩街机和FC游戏的场景仍历历在目，不知不觉中就玩到了20世纪90年代。在学习和使用计算机过程中，发现除了能敲出自己的名字，还可以玩一些那么有吸引力的游戏——海阔天空、千军万马，数不清的战斗！数不清的不眠之夜！玩果然是天性！

玩着玩着就想到，如果自己开发游戏是一件多么酷的事情！于是开始了软件开发之旅，不知道有多少学习软件开发的朋友是受到了游戏的影响！相信不止我一个。好吧，事情没那么简单，学着学着就开始工作了，此时才发现，怎么在做ERP系统呢？……

打拼了几年，终于有时间回到当初的梦想。但此时，游戏已不是计算机的专利了，智能手机的兴起，让“游戏机”装入口袋。很快，自己也小试了一下，使用Java ME开发了一个俄罗斯方块游戏，并在自己的Nokia E71上玩了很久，只是在家人的Nokia 5300中，显示效果完全不是一回事儿，Java ME版本问题，没办法解决！这段经历的总结是，在移动游戏开发方面，选择Nokia算是选错队友了！

继续选择新队友，首先，了解了一些平台，特别是移动平台，当然还有游戏开发框架，有专用的，也有跨平台的；最终，还是选择了Apple朋友圈，原因是多方面的，或者干脆说，喜欢做就做了，需要理由吗？！

软件开发，首先需要掌握一种编程语言，于是，Objective-C就来了，那时还没Swift什么事呢！换个角度讲，到现在为止，Swift的每一次改变都会让一些代码罢工，成熟的工具不应该是这样的。也许，Swift 3.0以后会好起来，我们拭目以待吧。

再说游戏开发，一个好用的框架是必不可少的，是的，Sprite Kit就是这样一个好用的游戏开发框架。使用Sprite Kit开发游戏，我们不需要使用复杂的CG技术就可以创建出各种游戏角色（比如，只需要一张图片），也不需要复杂的计算就可以让角色动起来（比如，只需要一个简单的动作类型）。简单！易用！就是它了！当然，还有一个重要的因素，Sprite Kit是Apple公司自己的游戏开发框架，所以，兼容性和技术支持等方面会更好，而且，同时支持iOS和OS X系统。

学习、实践、完成项目，在这个过程中，我都进行了大量的记录，有Objective-C编程语言方面的，有Sprite Kit应用方面的，当然也包括自己对技术应用和游戏开发的思考、理

解。有一天，我把这些资料整理了一下，惊奇地发现已经有了这么多内容，都可以成书了！

好的，被您猜中了，这就是本书形成的过程，也是我的游戏开发历程。

那么，为什么考虑出版本书呢？原因很简单，就是想为学习iOS或OS X游戏开发的朋友提供一份简单、实用的学习和参考资料，分享游戏开发的经历与乐趣。

本书特色

1. 简单、实用

简单，并不意味着本书的内容很少，正相反，一方面，本书对游戏开发的相关技术和方法进行提炼，包括了在iOS和OS X系统中进行游戏开发的必要知识；另一方面，本书也突出了知识的实用性，对内容深度和广度都进行有效的把握。通过这些内容的学习，相信读者能够打开游戏开发的大门。

2. 真正从零开始

学习本书，并不需要读者有特殊的编程经历，只要循序渐进，逐渐掌握书的内容，完全可以开发出iOS或OS X游戏。

3. 轻松学习

无论是玩游戏，还是开发游戏，我都会提倡趣味性与娱乐性，学习虽然是一件很艰苦的事情，但我们完全有理由去享受这一过程，特别对于游戏开发的学习过程。本书中，语言和示例都会以轻松的、直观的、有趣的方式来展现，希望大家也能够感觉到，我们就是在开心、轻松的聊天过程中来学习游戏开发的。

4. 理论与实践相结合

对于每个示例，都有很强的针对性，一切为了游戏开发；针对各种技术，我们不但介绍了基本的应用和实现，同时，讨论了各种实际应用的场景，力求让读者能够真正理解并融合这些知识，并能够应用到实际的游戏开发中。

5. 完整性

与很多介绍某种开发技术的图书不同，本书并不是单一技术的手册，而是包括了iOS或OS X游戏开发的完整内容，从基本的代码、Sprite Kit开发资源的应用，再到Game Center排行榜、应用内购买等，这些内容完整地展示了游戏从开发到上架所需要的知识。

读者对象

在这里，我们看看本书的内容都适合哪些朋友阅读和使用。

☐ 游戏开发爱好者

☐ iOS或OS X应用开发者

如何阅读本书

本书主要包括四个部分的内容：

第一部分介绍了如何使用Xcode和Objective-C语言编写程序。

第二部分讨论了Foundation框架中常用的开发资源，这些资源不但可以为游戏开发服务，同样也可以应用于各种类型的iOS或OS X应用开发。

第三部分讨论了Sprite Kit在游戏开发中的应用。

第四部分讨论了游戏开发相关的技术，如网络状态检测、Game Center排行榜、应用内购买等。

如果是Apple环境开发新手，建议从第1章开始逐步学习；如果是一名经验丰富的开发者，可以根据实际需要选择其中的内容阅读；实际工作中，可以将本书作为参考手册随时查阅。

阅读本书的建议

- 对于开发初学者，建议逐章阅读，并能够亲手实践书中的示例代码。
- 对于已经掌握Objective-C语言的读者，可以根据需要阅读相关的内容。
- 对于Foundation、Sprite Kit或常用功能实现部分，大家可以随时参考，并根据实际需要修改使用。

进一步学习建议

阅读并掌握本书的内容以后，相信读者已经掌握了Objective-C，并对iOS或OS X应用有了一定的了解，特别是，应该可以开发出属于自己的游戏了。

不过，我们也知道，开发一个大型游戏的知识绝不止书中这些内容，所以，完成本书内容的学习以后，大家可以更进一步，以下是关于深入学习的一些建议：

- 无论是Foundation还是Sprite Kit或Store Kit框架，都值得我们深入学习和理解，并能够在应用中合理地使用。关于这些内容，建议大家能够抽时间看一看官方文档。对于某些问题的众多解决方案和技巧，能够在学习和应用过程中主动思考，并亲手验证它们。
- 如果需要跨平台的游戏开发，建议大家了解至少一种框架，并能深入地学习和应用。

□ Swift是一种现代的编程语言，如果大家需要长期进行Apple环境的开发工作，建议学习并能够熟练使用它。

勘误和支持

由于作者水平有限、编写时间仓促，书中难免会出现一些疏漏或不足之处，而读者的批评和指正，正是我们共同进步的强大力量，欢迎您通过清华大学出版社网站（www.tup.com.cn）与我们联系，同时，也欢迎大家直接与作者交流，作者的邮箱是chydev@163.com。

致谢

感谢出版社的编辑老师耐心的交流与指导，使得本书能够顺利与读者见面。

感谢我的家人，他们承担了大量的家务，容忍了我在家里的“懒惰”，为我创造了一个温暖的家、一个安心的工作环境，特别是我的孩子们，他们总是说“爸爸在工作，我不打扰他”，而这些正是我快乐生活和努力工作的力量源泉。

谨以此书献给我的家人，以及热爱游戏、热爱开发的朋友们！

曹化宇

2016年7月

目 录

第1章 准备工作	1	2.10.4 参数指针	31
1.1 本书内容	1	2.11 static关键字	32
1.2 使用Xcode	2	2.12 块(block)	33
1.3 第一个Objective-C程序	4	2.13 枚举	35
1.4 获取帮助	7	2.14 结构	37
第2章 Objective-C开发基础	8	2.15 随机数	38
2.1 代码文件	9	2.16 预处理	38
2.2 注释	12	2.16.1 #define和#undef指令	39
2.3 变量	12	2.16.2 条件编译指令	39
2.4 常量	13	2.16.3 #import指令	40
2.5 整数	15	第3章 代码流程控制	41
2.5.1 取值范围	15	3.1 比较运算	41
2.5.2 算术运算	16	3.2 条件语句	42
2.5.3 NSLog()函数与格式化输出	17	3.2.1 if语句	42
2.5.4 组合运算符	17	3.2.2 ?:运算符	43
2.5.5 增量与减量运算	18	3.3 选择语句	44
2.5.6 二进制与位运算	18	3.4 循环语句	46
2.6 浮点数	23	3.4.1 for语句结构	46
2.7 布尔类型	24	3.4.2 while语句结构	47
2.7.1 BOOL类型	24	3.4.3 do-while语句结构	47
2.7.2 布尔运算	25	3.4.4 break语句	48
2.8 字符	26	3.4.5 continue语句	48
2.9 指针	28	3.5 goto语句与标签	49
2.10 自定义函数	29	3.6 异常处理	49
2.10.1 返回值类型	30	第4章 面向对象编程	51
2.10.2 函数名	30	4.1 面向对象编程基础	51
2.10.3 参数	30		

4.2 类与对象.....	53	5.5 对象深复制（实现NSCopying 协议）.....	79
4.2.1 接口部分.....	53	第6章 数组、集合与字典.....	82
4.2.2 实现部分.....	54	6.1 C风格数组.....	82
4.2.3 创建对象（实例化）.....	55	6.2 不可变数组（NSArray类型）.....	84
4.2.4 类的成员.....	56	6.2.1 创建NSArray对象.....	84
4.3 方法（任务）.....	58	6.2.2 数字对象（NSNumber类）.....	84
4.3.1 创建方法.....	58	6.2.3 使用NSValue类.....	85
4.3.2 description方法与NSLog()函数.....	61	6.2.4 数组成员操作.....	86
4.4 属性.....	61	6.2.5 保存与载入.....	87
4.4.1 使用@property和@synthesize指令.....	62	6.3 可变数组（NSMutableArray类型）.....	88
4.4.2 使用setter和getter方法.....	63	6.3.1 创建NSMutableArray对象.....	89
4.5 初始化方法.....	64	6.3.2 添加成员.....	89
4.6 继承.....	65	6.3.3 删除成员.....	90
4.6.1 成员的访问.....	65	6.3.4 替换成员.....	90
4.6.2 重写属性和方法.....	67	6.4 集合（Set）.....	91
4.6.3 继承关系中的初始化.....	67	6.4.1 不可变集合（NSSet类）.....	91
4.7 分类.....	69	6.4.2 可变集合（NSMutableSet类）.....	92
4.7.1 命名分类.....	70	6.5 字典（NSDictionary）.....	93
4.7.2 匿名分类.....	71	6.5.1 创建字典对象.....	93
4.8 对象复制与传递.....	71	6.5.2 常用成员.....	93
4.8.1 对象的复制.....	71	6.5.3 NSMutableDictionary类.....	94
4.8.2 对象作为参数.....	72	第7章 字符串.....	96
4.9 动态处理类和对象.....	72	7.1 C风格字符串.....	96
4.9.1 对象类型判断.....	72	7.2 不可变字符串（NSString类）.....	97
4.9.2 方法存在判断.....	73	7.2.1 创建NSString对象.....	97
4.9.3 动态调用方法.....	74	7.2.2 返回字符数量.....	98
第5章 协议.....	75	7.2.3 截取子字符串和字符.....	98
5.1 创建协议.....	75	7.2.4 大小写转换.....	99
5.2 实现协议.....	76	7.2.5 转换为C风格字符串.....	99
5.3 可选成员.....	77		
5.4 实现多个协议.....	78		

7.2.6 字符串比较与匹配	100	9.6.3 月份与名称	123
7.2.7 转换为数值	101	9.6.4 日期与名称	124
7.2.8 保存与读取	101	9.6.5 属相	125
7.3 可变字符串 (NSMutableString 类)	102	9.6.6 测试	125
7.3.1 创建NSMutableString对象	102	第10章 文件与目录	127
7.3.2 NSMutableString常用成员	102	10.1 获取系统信息	127
7.4 使用NSURL类	103	10.1.1 获取文稿目录 (Documents)	128
第8章 本地化字符串	105	10.1.2 获取临时目录与GUID	128
8.1 判断系统语言类型	105	10.1.3 更多系统信息	129
8.2 NSLocalizedString	106	10.2 NSData与NSMutableData类	130
8.3 NSLocalizedStringFromTable	107	10.3 使用NSFileManager类	130
8.4 综合应用	108	10.3.1 文件或目录是否存在	131
第9章 日期与时间	109	10.3.2 复制文件和目录	131
9.1 NSDate类	109	10.3.3 删除文件和目录	132
9.1.1 获取时间信息	110	10.3.4 移动、重命名文件和目录	133
9.1.2 时间的计算和比较	110	10.3.5 文件与目录的属性	133
9.2 NSDateComponents类	111	10.4 文件操作	134
9.3 NSCalendar类	112	10.4.1 读取文件内容	134
9.4 时区与区域设置	113	10.4.2 写入文件	134
9.4.1 NSTimeZone类	113	10.4.3 比较文件内容	135
9.4.2 NSDateFormatter类	114	10.4.4 检测文件读写权限	135
9.4.3 NSLocale类	115	10.5 目录操作	135
9.5 封装CDate类	115	第11章 归档	137
9.5.1 初始化方法	117	11.1 归档与解档	137
9.5.2 时间、区域和时区	118	11.1.1 实现NSCoding协议	137
9.5.3 日期与时间数据	119	11.1.2 使用NSKeyedArchiver类和 NSKeyedUnarchiver类	139
9.5.4 测试	120	11.2 利用归档复制对象	140
9.6 获取中国农历信息	120	第12章 通知中心与对话框	142
9.6.1 初始化方法	121	12.1 通知中心	142
9.6.2 年份与名称	122		

12.2	OS X对话框	146
12.3	iOS对话框	147
12.3.1	回顾UIAlertView类	150
12.3.2	使用UIAlertController类	152
12.3.3	iPad中的UIAlertController	154
第13章 Sprite Kit基础 156		
13.1	第一个Sprite Kit项目	156
13.1.1	启动界面	159
13.1.2	视图	161
13.1.3	创建场景	162
13.2	iOS设备与系统信息	163
13.2.1	设备类型与iOS版本	163
13.2.2	屏幕尺寸	164
13.2.3	封装——创建CApp.h和CApp.m文件	165
13.3	颜色	167
13.4	节点树	168
13.5	Core Graphics	171
13.5.1	CGPoint结构	171
13.5.2	CGSize结构	172
13.5.3	CGRect结构	172
13.5.4	CGRectContainsRect()函数	172
13.5.5	CGRectContainsPoint()函数	172
13.5.6	CGRectGetMidX()和CGRectGetMidY()函数	173
13.6	坐标系	173
13.6.1	位置 (position) 与锚点 (anchorPoint)	173
13.6.2	坐标转换	174
13.7	场景切换 (SKTransition)	174
13.8	游戏循环	176
13.9	场景 (SKScene) 中显示对话框	178

13.10	Mac中的Sprite Kit项目	180
13.10.1	项目初始化	180
13.10.2	响应鼠标与键盘	181
13.10.3	屏幕与尺寸	182
第14章 精灵、纹理与角色控制 185		
14.1	SKSpriteNode类	185
14.1.1	创建精灵节点	185
14.1.2	NSBundle类	187
14.1.3	组合节点	188
14.2	SKTexture类	189
14.2.1	截取纹理内容	189
14.2.2	精灵动画——翻滚的小行星	191
14.2.3	更多的小行星	191
14.3	移动与碰撞	192
14.3.1	场景初始化	193
14.3.2	碰撞检测与游戏状态	197
14.4	触摸控制	200
14.4.1	触摸响应方法	200
14.4.2	控制太空船	201
14.4.3	单击	202
14.4.4	手势	202
14.5	Mac中的太空船	207
14.5.1	处理鼠标操作	210
14.5.2	处理键盘控制	212
14.5.3	在update:方法中处理键盘控制	213
第15章 动作与声音播放 216		
15.1	基本动作类型	217
15.1.1	移动	217
15.1.2	尺寸与缩放	218
15.1.3	显示和隐藏	219
15.1.4	旋转	220

15.1.5 等待	221
15.2 动作的组合	221
15.2.1 动作组	221
15.2.2 动作序列	222
15.2.3 动作重复	222
15.2.4 动画动作	223
15.3 声音播放	223
15.3.1 使用动作播放声音	223
15.3.2 使用AVAudioPlayer播放声音	224
15.4 动作的使用	224
15.4.1 执行动作	225
15.4.2 通过键(Key)执行动作	225
15.4.3 取消所有动作	226
15.4.4 判断节点是否有动作执行	226
15.4.5 修改动作速度	226
第16章 更多节点类型	227
16.1 SKShapeNode	228
16.1.1 基本图形节点	229
16.1.2 根据路径创建图形节点	230
16.2 SKVideoNode	231
16.3 SKCropNode	232
16.4 SKEffectNode	233
16.5 SKEmitterNode与粒子效果	234
16.5.1 在Xcode中创建粒子	234
16.5.2 下雪场景	235
16.5.3 模拟爆炸	236
16.5.4 SKEmitterNode 类	238
16.6 节点的组合	240
第17章 Sprite Kit游戏常用算法	242
17.1 两点距离	242
17.2 碰撞测试	243

17.3 角度	245
17.4 视线	246
17.5 躲避障碍物	249

第18章 综合测试——Mac版 坦克大战 251

18.1 场景初始化	252
18.1.1 创建坦克	254
18.1.2 创建敌人	254
18.1.3 创建岩石	255
18.2 坦克的操作	255
18.2.1 坦克的动作	256
18.2.2 鼠标和键盘控制	257
18.2.3 自动执行	258
18.3 敌人的AI	259
18.4 处理已发射炮弹	261
18.5 自己动手	262

第19章 加速计与陀螺仪 263

19.1 自动响应	264
19.2 调用数据	267
19.3 游戏控制方法小结	269
19.3.1 触摸屏	269
19.3.2 键盘与鼠标	270

第20章 网络状态 271

20.1 检测网络状态	271
20.2 监视网络状态	272

第21章 Game Center排行榜 274

21.1 配置排行榜	274
21.2 在应用中处理排行榜	275
21.2.1 Game Center登录状态	275

21.2.2 提交成绩.....	277
------------------	-----

第22章 应用内购买..... 280

22.1 准备测试用户和设备.....	280
22.2 创建App内购买.....	281
22.3 在项目中使用App内购买.....	282
22.3.1 检测访问限制与网络.....	282
22.3.2 执行购买操作.....	285
22.3.3 恢复已购项目.....	287
22.3.4 在场景(SKScene)中执行购买操作.....	287
22.3.5 在场景(SKScene)中执行恢复操作.....	290

第23章 发布到App Store..... 292

23.1 注册开发者与设备.....	292
23.2 发布准备.....	293
23.2.1 应用图标.....	293
23.2.2 iOS应用截图.....	294
23.2.3 OSX应用截图.....	295
23.2.4 上传应用.....	295
23.2.5 项目审核.....	295

第 1 章

准备工作

兵马未动，粮草先行！

当然，在这里只需要一杯茶或咖啡就可以了。对了，如果你手边正好有一台Mac，现在也可以打开了，因为在内容简介后，我们很快就可以开始写代码了。

1.1 本书内容

Objective-C是一种基于C语言的面向对象的开发语言，目前主要应用于OS X和iOS系统的应用开发。虽然Apple公司已经发布了另外一种编程语言Swift，但它完全取代Objective-C恐怕还需要一些时间。

本书就是讨论如何使用Objective-C编程语言进行iOS和OS X应用，特别是游戏类应用开发的相关内容，主要由以下几个部分组成：

- ❑ Objective-C编程语言，从第2章到第5章主要包括基本的代码编写、数据处理、流程控制，以及面向对象、协议等内容。
- ❑ Foundation框架中的常用开发资源，从第6章到第12章主要讨论集合、日期与时间的处理、文件与目录的应用、归档、消息对话框等内容。
- ❑ Sprite Kit是一套功能强大的开发库，它针对多媒体开发进行了大量的封装，包括二维图形、游戏场景、模拟物理环境、动画、声音播放等，而且，Sprite Kit同时支持OS X或iOS系统中的游戏和其他多媒体应用开发工作，为开发者提供了极大的便利。从第13章到第18章将讨论Sprite Kit开发游戏的相关内容。
- ❑ 第19章到第23章介绍游戏应用开发相关的技术和注意事项，比如，如何使用加速计与陀螺仪、如何使用Game Center排行榜、如何实现应用内购买，以及应用发布需要注意的问题等。

本书的内容先简单说到这里，大家一定等不及要开始写代码了，好的，我们马上来了解一下开发工具。

1.2 使用Xcode

又是句老话：工欲善其事，必先利其器。

虽然很老套，不过很实际，对吧！那么，在开发iOS和OS X应用时，我们使用什么工具呢？相信大部分人会首先想到Xcode，这是一个由Apple公司提供的集成开发环境（IDE），我们可以通过它创建各种类型的应用，并可进行编码、测试，以及应用的上传等工作。重要的是，它是免费的。

现在是使用Mac的时候了，如果你的计算机中还没有安装Xcode，可以在App Store中找到它的正式版本。说实话，Xcode安装包确实有点大，下载和安装会需要一点时间。

不过没关系，大家可以在安装Xcode的时候，先接着往下看，先熟悉一下要做什么也不是一件坏事。

实际上，在软件开发时，能够在纸上写出来、画出来，先真正弄清楚要干什么，再通过计算机实现，这确实是一个不错的工作方式。

（不知道过了多长时间……）

好的，Xcode一不小心就已经安装好了。现在，我们可以在Dock中单击Launchpad图标（类似火箭的那个），然后，找到Xcode图标，单击它就可以进入Xcode了。

对了，Xcode图标长得和图1-1所示的差不多。

友情提示，打开Xcode后，在Dock中的Xcode图标上，按住Control键，然后再单击鼠标，在菜单中选择“选项”→“在Dock中保留”，就可以在Dock上显示Xcode图标，再打开时就方便了。此外，如果使用的是双键鼠标，可以直接使用右键菜单操作（要不怎么说来自苹果宇宙的技术就是有一些不一样呢）。

当我们打开Xcode后会是什么呢？一般会出现一个欢迎界面（真是太客气了），如图1-2所示。

这个界面中，可以在右侧列表中选择正在进行的项目或者通过右下角的“Open another project...”打开其他已存在的项目，也可以创建一个新的项目，比如，我们选择“Create a new Xcode project”可以创建新的项目。

在讨论Objective-C编程语言和Foundation常用资源时，我们会使用简单的项目类型，也就是OS X命令行工具项目（Command Line Tool），如图1-3所示。

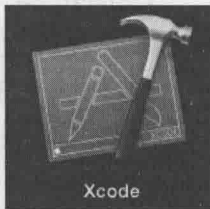


图1-1 Xcode图标

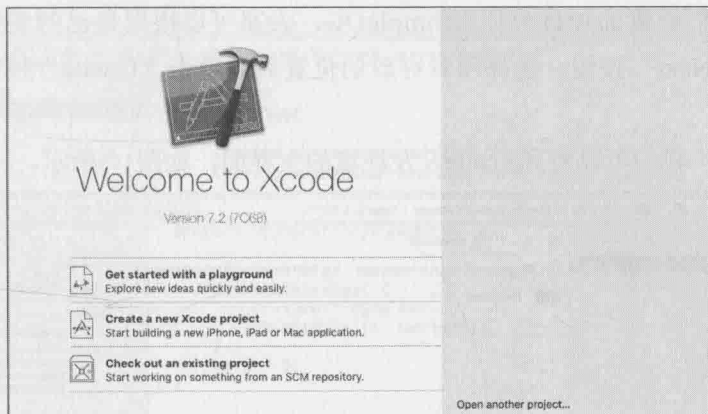


图1-2 Xcode欢迎界面

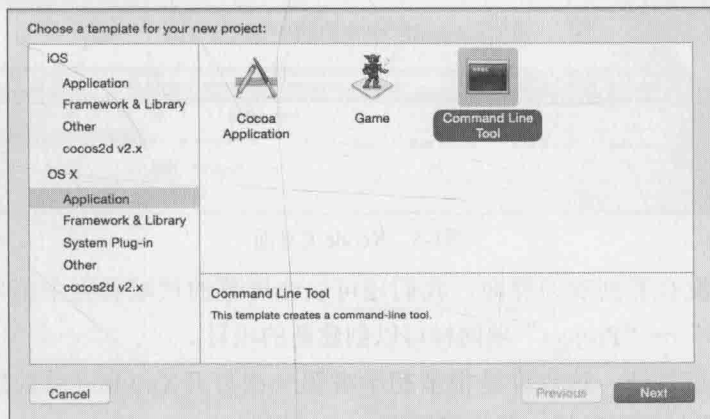


图1-3 选择项目类型

单击“Next”按钮，在出现的界面中输入项目名称等基本信息，如图1-4所示。

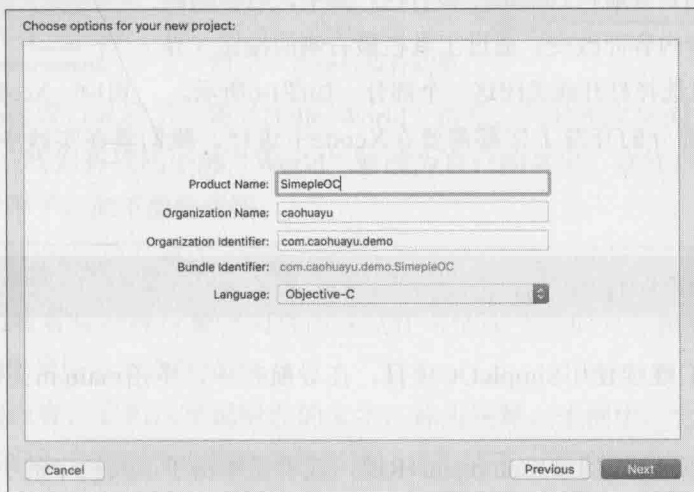


图1-4 设置项目基本信息

本例中，我们设置的项目名称为SimpleOC，大家可以根据自己的爱好来设置项目名称，然后单击“Next”按钮，选择项目存放的位置后，单击“Create”按钮，即可完成项目的创建。

项目创建后，我们可以看到Xcode开发环境的主界面，如图1-5所示。

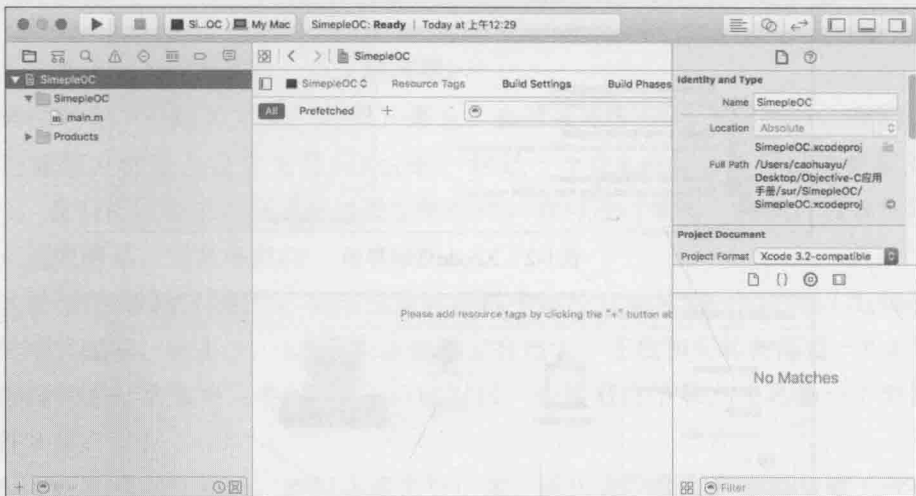


图1-5 Xcode主界面

此外，如果没有看到欢迎界面，我们还可以在屏幕的顶端看到主菜单，通过其中的“File”→“New”→“Project”项同样可以创建新的项目。

开发环境有点复杂，这也许是很多初学者第一次打开Xcode以后的感觉，不过没关系，工具嘛，总是越用越熟练，我们慢慢来。

图1-5中，可以看到Xcode主界面的4个主要组成部分，最上方的是工具栏，下面三个部分，从左到右分别是导航栏、编辑栏和右栏。其中，右栏的内容会根据当前工作内容而改变。通过工具栏最右侧的按钮（界面右上角），可以选择打开或关闭这三个部分，如图1-6所示。



图1-6 Xcode操作区打开或关闭

接下来，大部分的开发工作都需要在Xcode中进行，我们就在实践中慢慢熟悉它的使用方式吧。

1.3 第一个Objective-C程序

接下来，我们继续使用SimpleOC项目，在导航栏中，单击main.m文件，如图1-7所示（已关闭右栏）。

如果不小心按了键盘上的Command+R键，或者是单击了工具栏中的小三角图标，你会发现，在界面的下方会显示一条信息，如图1-8所示。