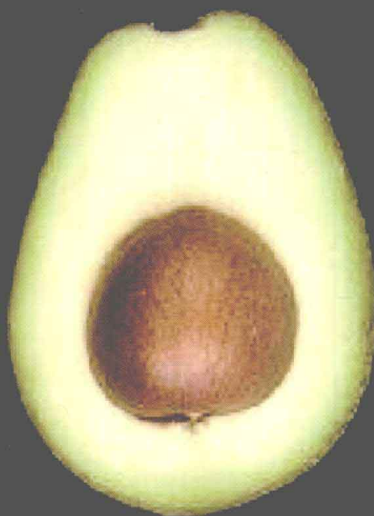


移动与嵌入式开发技术

Learn iPhone and iPad cocos2d Game Development



# iPhone & iPad cocos2d 游戏开发实战

(美) Steffen Itterheim 著  
同济大学苹果俱乐部 译

Apress®

清华大学出版社

移动与嵌入式开发技术

# iPhone & iPad cocos2d 游戏开发实战

(美) Steffen Itterheim      著  
同济大学苹果俱乐部      译

清华大学出版社

北 京

Steffen Iltterheim

Learn iPhone and iPad cocos2d Game Development

EISBN: 978-1-4302-3303-9

Original English language edition published by Apress, 2855 Telegraph Avenue, #600, Berkeley, CA 94705 USA. Copyright © 2010 by Apress L.P. Simplified Chinese-Language edition copyright © 2011 by Tsinghua University Press. All rights reserved.

本书中文简体字版由 Apress 出版公司授权清华大学出版社出版。未经出版者书面许可，不得以任何方式复制或抄袭本书内容。

北京市版权局著作权合同登记号 图字：01-2011-2256

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

iPhone & iPad cocos2d 游戏开发实战/(美)伊特海姆(Iltterheim, S.) 著; 同济大学苹果俱乐部 译.

—北京: 清华大学出版社, 2012.1

(移动与嵌入式开发技术)

书名原文: Learn iPhone and iPad cocos2d Game Development

ISBN 978-7-302-27441-4

I. i… II. ①伊… ②同… III. ①移动电话机—游戏程序—程序设计 ②便携式计算机—游戏程序—程序设计 IV. ①TN929.53 ②TP368.32

中国版本图书馆 CIP 数据核字(2011)第 236314 号

责任编辑: 王 军 李维杰

装帧设计: 牛艳敏

责任校对: 成凤进

责任印制: 李红英

出版发行: 清华大学出版社

地 址: 北京清华大学学研大厦 A 座

<http://www.tup.com.cn>

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: 北京嘉实印刷有限公司

经 销: 全国新华书店

开 本: 185×260 印 张: 21.5 字 数: 523 千字

版 次: 2012 年 1 月第 1 版 印 次: 2012 年 1 月第 1 次印刷

印 数: 1~3000

定 价: 48.00 元

---

产品编号: 040477-01

# 译者序

近几年来，苹果公司成了全球科技与时尚完美结合的标志，iPhone、iPad 等产品更是深得大众好评。于是，越来越多的开发者希望在这一平台上一显身手。本书介绍的 cocos2d 是 iOS 平台最强大的开源游戏引擎之一，无论是多年从事苹果开发的资深人士，还是刚刚接触苹果开发的初学者，都可以从本书中学到切实有用的知识和技巧。

在学习 cocos2d 开发的过程中，我有幸读到了这本书。作者风趣的语言和清晰的讲解使我学到了很多。于是，我们同济大学苹果俱乐部的几位同学——一群热爱苹果技术的大学生，萌生了翻译本书的想法，希望更多的人可以从这本书中受益。在对本书的学习过程中，我们将同你一起剖析和解读书中涉及的知识要点。

或许你是初次接触 cocos2d。cocos2d 整合了 Box2D 和 Chipmunk 两大物理引擎。在本书中，作者将手把手、循序渐进地教你运用 cocos2d 技术实现滚屏、摇杆、粒子效果，甚至是整个游戏世界。同时，作者还介绍了一些非常实用的辅助开发工具，这些看似简单的小工具在实际的开发过程中却能发挥重要的作用。

或许对于简单的 cocos2d 开发，你已经驾轻就熟并希望了解更多的周边技术以及市场情况。相信作者在书中介绍的 Game Center 编程，以及作者结合自身实践经历讲述的开发之外的经验，比如如何宣传你的软件、如何与出版商合作等，都将对你有极大的帮助。至少这些让我受益匪浅。

本书的翻译及校对工作由沈赟程、毕嘉奇、陈逸飞、何若运和我共同完成。在翻译过程中，我们也遇到了一些困难，但是好在有冯家琪、王得希、佟强、许家伟等人的大力协助，才使得本书的中文版如期与大家见面。这里我更要感谢我们的老师郭明森和张雯莹，谢谢对我们翻译工作的支持。另外特别感谢李阳编辑和李维杰编辑，他们对译文的严格审核和仔细修改使本书的行文质量大大提高。

霍 巨  
2011年9月8日

# 作者简介

Steffen Itterheim 从 20 世纪 90 年代开始就一直热衷于游戏开发。在 Doom 和 Duke Nukem 3D 社区的工作是他的第一份自由职业，当时他是 3D Realms 的一名 beta 测试人员。作为职业游戏开发者，Steffen 拥有 10 多年的丰富经验，其中大部分时间担任 Electronic Arts Phenomic 的游戏和工具程序员。2009 年 Steffen 第一次接触 cocos2d，那时他与其他人共同创办了一家 iOS 游戏公司——Fun Armada。他乐于将自己的宝贵经验传授给其他游戏开发者，以帮助他们更上一层楼。有机会你可能会在白天看到他在住所附近茂密的葡萄园周围散步，也可能在晚上看到他在 Nevada 沙漠收集瓶盖。

# 技术编辑简介

**Boon Chew** 是 Nanaimo Studio 的执行董事。Nanaimo Studio 位于西雅图和上海，是一个专注于互联网和移动游戏的工作室。Boon 拥有丰富的游戏开发和交互性媒体经验，曾就职于 Vivendi Universal、Amazon、Microsoft 以及其他游戏工作室和广告代理商。他热衷于创造，喜欢与伟人一起工作。您可以通过 [boon@nanaimostudio.com](mailto:boon@nanaimostudio.com) 与 Boon 取得联系。

# 致 谢

本书的这一部分使我有一点担心，我不想忘记每一位对本书的编写有指导和帮助的人，但是我知道不可能提到你们中的每一位。如果这里没有提到你，并不代表我不感谢你！给我一支笔，我将马上在书上写下你的名字，并且为我没有提及你抱以诚挚的歉意！

首先我要感谢的是你们，亲爱的读者。没有你们，这本书将没有任何意义。没有你们喜爱本书并且希望从中获得知识，我可能从一开始就不会想到要写这本书。在写这本书的过程中，我博客的读者和其他一些人以见面或邮件的方式向我提出了一些很有价值的建议和要求。感谢你们！感谢所有人！

接下来我将按照我遇见他们的顺序介绍那些对本书有帮助的人。

首先我要感谢 Jack Nutting，他最先使我有编写这本书的想法。我很感激他没有掩饰写书所需要的工作量，所以我编写本书并不是毫无准备。

Clay Andres，我不得不感谢这么善良的一个人。他对于书中章节所提出的建议是不可估价的，并且也十分恰当。他还帮助我整理写本书的思路，并且他也是一个非常健谈的人。Clay，我希望暴风雨没有冲走你的房子。

很感谢 Kelly Moritz，她是本书的流程编辑。虽然她很忙，但是她总能抽出时间耐心地解答我的问题，满足我的要求。当事情变得糟糕而混乱时，她总是能让每件事情回到正轨。

我收到了 Brian MacDonald(本书的开发编辑)和 Boon Chew(本书的技术审校者)很多很多的回复和建议。他们使我更有信心。Brian 帮助我理解了写作本书的纷繁难懂之处，而 Boon 则指出了很多技术点不明确的地方和需要详细解释的地方。非常感谢两位！

感谢编辑 Sharon Terdeman 和 Damon Larson。如果没有你们的帮助，本书的文字——用我们程序员的说法——将充满各种语法错误和编译警告。当 Sharon 说我的章节很容易编辑时，我实际上并不相信，虽然我知道她是专家。感谢 Damon 从读者的角度为我指出了那么多不易理解的段落，以便我修改。

我还想感谢 Bernie Watkins，他负责管理 Alpha Book 的反馈以及我的合同。同样感谢 Chris Guillebeau，她是一位出色的富有灵感的博客作者和角色模特儿。

当然我的朋友和家人也都参与了这本书的编写，他们给了我一些反馈意见，并忍受我写作时的过分投入。在此谢谢你们！

# 前 言

2009年5月,我第一次接触了 Mac OS 平台,并且学习了 Xcode、Objective-C 和 cocos2d。即便是对于经验丰富的程序员来说,这也是一个不小的挑战。就在那段时间,我意识到 cocos2d 真的很棒。但是,相比我当时学习的其他技术,cocos2d 的教程、文档和说明文章实在是太匮乏了。

转眼就到了 2010 年 5 月。在这一年里,我完成了 4 个 cocos2d 项目。我对 Objective-C 和 cocos2d 的使用都更加娴熟了。但是,我发现其他很多的程序员还在为一些基本的问题感到困惑,甚至产生误解,这些情景让我想到了一年前痛苦的自己。有关 cocos2d 的文档依然处于严重缺乏的状态。

今天,有不少使用 cocos2d 的开发者在博客上发布 cocos2d 教程、分享他们的使用心得,并因此引起了广泛的关注。大家都在积极地撰写着 cocos2d 的文档,只可惜你一言我一语,太过分散,读者很难对 cocos2d 有一个系统的理解。所以,这时候就需要有一个网站来整合这些散落在网络上的宝贵资料。

为此,我创建了一个网站([www.learn-cocos2d.com](http://www.learn-cocos2d.com))来分享我对 cocos2d 和游戏开发的理解。这个网站上有一些教程,列出了一些常见问题的解答,并提供链接以便对 cocos2d 感兴趣的读者能够找到所有与 cocos2d 有关的重要资料。相应的,我也会出售一些与 cocos2d 相关的产品,希望有一天它能助我达到经济独立的终极目标(已经有不少光辉的典范证明了这种商业模式的可行性)。而且我很喜欢帮助别人,在我工作过的所有公司里我都经常帮助同事们解决问题。一想到这个网站可以令所有人受益,我就觉得无比兴奋。

从网站发布的第一天起,它就获得了很大的成功(我做梦也没想到)。同时,在网站发布后的 24 小时内,Jack Nutting 就问我是不是考虑写一本有关 cocos2d 的书。于是,在经历了一系列小故事之后,就有了现在你手上的这本书。

我把我所知道的一切都放在了我的网站上,也写进了这本书里,但这些内容最多也就占全书的四分之一。我希望这本书可以以前所未有的详细叙述向大家介绍 cocos2d 的工作原理和使用方法,如果真是这样,那么我这 4 个月夜以继日的辛勤劳动就真的值了!

在这本书的撰写过程中,我学到了很多,其中包括一些我根本没想过要去了解的东西(虽然这听起来有点奇怪)。而现在,我最大的期待就是你能够从本书中学习到所有你想要知道的 cocos2d 和游戏开发知识!



# 目 录

第 1 章 简介	1	2.1.5 下载并安装 cocos2d	13
1.1 选择 iPhone 版 cocos2d 的理由	2	2.2 HelloWorld 应用程序	16
1.1.1 免费	2	2.2.1 HelloWorld 文件在项目中的位置	16
1.1.2 开源	2	2.2.2 资源	17
1.1.3 Objective-C	2	2.2.3 其他资源	17
1.1.4 2D 游戏引擎	3	2.2.4 类	18
1.1.5 物理引擎	3	2.3 cocos2d 中的内存管理问题	22
1.1.6 技术难度较低	3	2.4 改变世界	24
1.1.7 依然需要编程	3	2.5 你还应该知道的	26
1.1.8 超棒的 cocos2d 社区	4	2.5.1 iOS 设备	26
1.2 注意事项	4	2.5.2 关于内存的使用	27
1.2.1 Section 3.3.1	4	2.5.3 模拟器	29
1.2.2 平台移植	5	2.5.4 关于日志	30
1.3 本书读者对象	5	2.6 本章小结	30
1.4 阅读前提	5	第 3 章 基础知识	31
1.4.1 编程经验	5	3.1 cocos2d 中的单件类	31
1.4.2 Objective-C	5	3.2 Director 类	33
1.5 本书内容	6	3.3 场景图	33
1.5.1 iOS 游戏开发新手将学会什么	7	3.4 CCNode 类层次体系	34
1.5.2 iPhone 应用程序开发者将学会什么	7	3.5 CCNode 类	34
1.5.3 cocos2d 开发者将学会什么	7	3.5.1 节点的处理方式	34
1.6 章节介绍	8	3.5.2 动作的处理方式	35
1.7 问题和反馈	9	3.5.3 消息调度	36
第 2 章 入门	11	3.6 场景和层	39
2.1 准备工作	11	3.6.1 CCScene	40
2.1.1 系统要求	11	3.6.2 场景和内存	40
2.1.2 注册成为 iOS 开发者	12	3.6.3 推进和弹出场景	41
2.1.3 证书和授权文件	12	3.6.4 CCTransitionScene	42
2.1.4 下载并安装 iPhone SDK	12	3.6.5 CCLayer	43
		3.7 CCSprite 类	46
		3.7.1 定位点揭秘	46

3.7.2 纹理大小	47	5.5.2 CCParallaxNode	97
3.8 CCLabel 类	47	5.5.3 CCRibbon	99
3.9 菜单	48	5.5.4 CCMotionStreak	100
3.10 动作	50	5.6 本章小结	101
3.10.1 重复动作	50	<b>第 5 章 深入了解精灵</b>	<b>103</b>
3.10.2 流畅动作	50	6.1 CCSpriteBatchNode	103
3.10.3 动作序列	51	6.1.1 何时使用	
3.10.4 瞬时动作	52	CCSpriteBatchNode	104
3.11 cocos2d 测试案例	53	6.1.2 示例项目	105
3.12 本章小结	53	6.2 精灵动画初体验	110
<b>第 4 章 你的第一个游戏</b>	<b>55</b>	6.3 用于创建动画的辅助类别	112
4.1 按部就班地创建项目	55	6.4 使用纹理图册	114
4.2 添加 Player Sprite	59	6.4.1 何为纹理图册	114
4.3 加速计输入	62	6.4.2 Zwoptex 工具介绍	114
4.4 首次测试运行	63	6.4.3 使用 Zwoptex 桌面版创建	
4.5 玩家速度	63	纹理图册	115
4.6 添加障碍物	66	6.4.4 在 cocos2d 中使用纹理	
4.7 碰撞检测	72	图册	117
4.8 得分标签	73	6.4.5 改进 CCAAnimation 辅助	
4.9 CCBitmapFontAtlas 与 Hiero		类别	119
简介	74	6.4.6 将所有图像都放入一个	
4.10 添加音频	76	纹理图册中	120
4.11 移植到 iPad	76	6.5 自己动手	121
4.12 本章小结	77	6.6 本章小结	122
<b>第 5 章 游戏组件</b>	<b>79</b>	<b>第 7 章 滚屏射击游戏(上)</b>	<b>123</b>
5.1 使用多个场景	79	7.1 高级视差滚屏	123
5.1.1 添加多个场景	79	7.1.1 将背景创建为底纹	123
5.1.2 正在加载下一段, 请做好		7.1.2 在代码中重建背景	125
准备	81	7.1.3 移动 ParallaxBackground	126
5.2 使用多个层	83	7.1.4 视差滚动的速度因素	127
5.2.1 实现关卡的最佳方法	89	7.1.5 实现背景的无限滚动	129
5.2.2 CCColorLayer	90	7.1.6 消除闪烁	131
5.3 从 CCSprite 类继承游戏		7.1.7 重复贴图	132
对象	90	7.2 虚拟手柄	133
5.4 使用 CCSprite 复合游戏		7.2.1 引入 SneakyInput	133
对象	91	7.2.2 集成 SneakyInput	134
5.5 奇妙的 CCNode 派生类	96	7.2.3 触摸按钮产生射击	135
5.5.1 CCProgressTimer	96	7.2.4 为按钮添加皮肤	137

7.2.5 控制动作 .....	139	10.2 使用 Zwoptex 处理图片 .....	191
7.2.6 数字控制 .....	142	10.3 Tiled 地图编辑器 .....	193
7.2.7 另一个选择: GPJoystick .....	143	10.3.1 创建新的瓦片地图 .....	193
7.3 本章小结 .....	143	10.3.2 设计瓦片地图 .....	194
<b>第 8 章 滚屏射击游戏(下) .....</b>	<b>145</b>	10.4 在 cocos2d 中使用直角 瓦片地图 .....	196
8.1 添加 BulletCache 类 .....	145	10.4.1 定位被触摸的瓦片 .....	199
8.2 关于敌人 .....	149	10.4.2 提高性能和可读性 .....	201
8.3 Entity 类的继承体系 .....	151	10.4.3 使用对象层 .....	202
8.3.1 EnemyEntity 类 .....	151	10.4.4 绘制对象层矩形 .....	203
8.3.2 EnemyCache 类 .....	155	10.4.5 滚动瓦片地图 .....	206
8.3.3 组件类 .....	159	10.5 本章小结 .....	207
8.4 射击开火 .....	161	<b>第 11 章 斜角瓦片地图 .....</b>	<b>209</b>
8.5 大怪物的生命条 .....	163	11.1 设计斜角瓦片地图图形 .....	209
8.6 本章小结 .....	165	11.2 使用 Tiled 编辑斜角瓦片 地图 .....	211
<b>第 9 章 粒子效果 .....</b>	<b>167</b>	11.2.1 新建一个斜角瓦片 地图 .....	211
9.1 粒子效果实例 .....	167	11.2.2 创建新的斜角瓦片集 .....	212
9.2 用复杂方法创建粒子效果 .....	170	11.2.3 设计斜角瓦片地图的 基本规则 .....	213
9.2.1 变化度属性 .....	173	11.3 将斜角瓦片地图应用到 游戏编程中 .....	214
9.2.2 粒子数量 .....	173	11.3.1 在 cocos2d 中加载斜角 瓦片地图 .....	214
9.2.3 发射器持续时间 .....	174	11.3.2 在 cocos2d 中设置斜角 瓦片地图 .....	214
9.2.4 发射器模式 .....	174	11.3.3 定位一个斜角瓦片 .....	216
9.2.5 粒子位置 .....	176	11.3.4 滚动斜角瓦片地图 .....	218
9.2.6 粒子大小 .....	177	11.3.5 斜角瓦片地图的边界 问题 .....	219
9.2.7 粒子方向 .....	177	11.3.6 增加一个可移动的玩家 角色 .....	221
9.2.8 粒子生存期 .....	177	11.4 在游戏中加入更多内容 .....	228
9.2.9 粒子颜色 .....	178	11.5 本章小结 .....	228
9.2.10 粒子混合模式 .....	178	<b>第 12 章 物理引擎 .....</b>	<b>231</b>
9.2.11 粒子贴图 .....	179	12.1 物理引擎的基本概念 .....	231
9.3 Particle Designer 介绍 .....	180	12.2 物理引擎的局限性 .....	232
9.3.1 使用 Particle Designer 生成 的粒子效果 .....	182		
9.3.2 分享粒子效果 .....	183		
9.4 在射击游戏中添加粒子 效果 .....	185		
9.5 本章小结 .....	186		
<b>第 10 章 瓦片地图 .....</b>	<b>189</b>		
10.1 瓦片地图简介 .....	189		

12.3	Box2D 与 Chipmunk .....	232	14.1.2	建立排行榜和成就 .....	282
12.4	Box2D .....	233	14.1.3	创建 cocos2d Xcode 项目 .....	283
12.4.1	Box2D 眼中的世界 .....	234	14.1.4	配置 Xcode 项目 .....	284
12.4.2	把移动范围限制在 屏幕内 .....	235	14.1.5	小结 .....	286
12.4.3	转换点 .....	237	14.2	Game Kit 编程 .....	287
12.4.4	在 Box2D 世界中添加 盒子 .....	238	14.2.1	GameKitHelper 代理 .....	287
12.4.5	连接精灵和刚体 .....	239	14.2.2	检查 Game Center 是否 可用 .....	289
12.4.6	碰撞检测 .....	240	14.2.3	验证本地玩家身份 .....	289
12.4.7	连接刚体 .....	243	14.2.4	block 对象 .....	292
12.5	Chipmunk .....	244	14.2.5	接收本地玩家的好友 列表 .....	294
12.5.1	面向对象的 Chipmunk .....	244	14.2.6	排行榜 .....	296
12.5.2	构建 Chipmunk 物理 空间 .....	244	14.2.7	成就 .....	300
12.5.3	将盒子添加到物理 空间中 .....	246	14.2.8	联机 .....	304
12.5.4	添加小盒子 .....	247	14.3	本章小结 .....	308
12.5.5	更新盒子的精灵 .....	249	第 15 章	番外篇 .....	311
12.5.6	Chipmunk 碰撞实践 .....	250	15.1	一些有用的技术 .....	311
12.5.7	Chipmunk 中的关节 .....	251	15.1.1	社交网络 .....	312
12.6	本章小结 .....	253	15.1.2	套接字服务器技术 .....	313
第 13 章	弹球游戏 .....	255	15.1.3	广告和分析 .....	313
13.1	图形: 凸多边形和逆时针 方式 .....	255	15.1.4	推送通知供应商 .....	314
13.2	应用 VertexHelper .....	256	15.2	从源码项目中受益 .....	314
13.3	创建弹球桌 .....	258	15.2.1	Sapus Tongue .....	315
13.4	加入动态元素 .....	264	15.2.2	LevelSVG .....	315
13.4.1	BodyNode 类 .....	264	15.2.3	iPhone RPG Game Kit .....	315
13.4.2	球 .....	267	15.2.4	Line-Drawing Game Starterkit .....	316
13.4.3	反弹器 .....	271	15.3	参考 .....	316
13.4.4	发射器 .....	272	15.4	与出版商合作 .....	317
13.4.5	挡板 .....	276	15.5	寻找自由职业者 .....	318
13.5	本章小结 .....	280	15.6	寻找免费的艺术品和 音频 .....	319
第 14 章	Game Center .....	281	15.7	寻找交易工具 .....	319
14.1	激活 Game Center .....	281	15.8	市场 .....	320
14.1.1	在 iTunes Connect 中 创建应用程序 .....	282	15.8.1	营销游戏和自己 .....	320
			15.8.2	公关和新闻发布 .....	321

15.9 寻求帮助.....	322	15.9.3 教程和常见问题解答 .....	324
15.9.1 cocos2d 首页 .....	323	15.10 写在最后 .....	324
15.9.2 栈交换网络 .....	323	15.11 本章小结 .....	325

# 第 1 章

## 简 介

可曾想象，有朝一日你会自己写一个计算机游戏然后靠它赚钱？有了 Apple 的 iTunes App Store 及其配套移动设备，如 iPhone、iPod Touch 和 iPad，要实现这个梦想不再是一件难事。当然，这并不表示开发一个游戏有多简单，你仍然需要学习很多游戏开发和编程的知识。不过，既然你选择了阅读本书，我就有理由相信你已经下定决心踏上这条游戏开发之路了。恭喜你选择了一个可能是全世界最有趣的游戏开发引擎——iPhone 版 cocos2d。

使用 cocos2d 的开发者可能有很多不同的专业背景，来自不同的领域。有些人(比如我)可能是已经从事游戏开发好几年甚至几十年的专业人士，还有一些人可能是刚开始接触 iOS 平台游戏开发，甚至有些人是刚转行加入这一充满激情的游戏开发领域。不管你属于哪一类人，我保证，看完这本书一定能够有所收获。

是这样一种信念让我们走到一起：我们都热爱游戏，也热爱设计和编写游戏。本书十分推崇这种信念，并会向读者介绍一些能够帮助简化游戏开发过程的工具。最重要的是，这本书会教大家写一些很有借鉴意义的小游戏，从中你可以学会如何把一些理论知识运用到现实的游戏开发中。

有些书会整页整页地教读者怎样用一些特定的游戏编程 API 来写一个无聊的战机类太空游戏(Asteroids)，我读到这种书的时候总是觉得特别没劲。我觉得一本好书应该向大家介绍游戏编程的理念和开发工具，因为这些东西是永恒的，不会随着 API 或编程喜好的变化而变化。我读编程书籍和游戏开发书籍已经有 20 多年了，我认为最有价值的书是那些高于技术本身的，是能够让我明白为什么这个地方会这样设计、这样编程、这样做有什么好处的书。所以本书不仅会关注游戏代码的含义，更会关注它的工作原理以及在哪些处理上需要根据情况权衡利弊。

我希望你能学着写出一些有价值的、能在 App Store 上热卖并且受玩家欢迎的游戏。我会介绍这本书里的示例游戏背后深藏的思想和技术理念，当然，我也会告诉你在游戏编程中如何使用 cocos2d 和 Objective-C。本书源代码中有大量注释，它们可以帮助你正确地理解代码的含义。

学习别人的源代码并且根据注释去关注一些重要设计对我来说是学习新知识的最好方法(我想它对你来说也会是一个很棒的方法)。你可以对这本书的随书源代码加以修改,进而做出自己的游戏。我非常期待在不久的将来能够玩到你的游戏!完成你的游戏千万别忘了告诉我!你可以通过我的邮箱([steffen@learn-cocos2d.com](mailto:steffen@learn-cocos2d.com))联系我,也可以登录我的网站(<http://www.learn-cocos2d.com>)给我留言。

## 1.1 选择 iPhone 版 cocos2d 的理由

游戏开发者在选择游戏引擎时首先会对他们要选择的产品做一些评估。综合以下因素,我认为 cocos2d 对许多开发者来说会是一个非常棒的选择。

### 1.1.1 免费

首先, cocos2d 是免费的。不需要花钱就可以用它来进行开发。其次,你可以随心所欲地开发 iPhone、iPod 和 iPad 应用,无论免费还是收费都可以。说真的,这是完全没有附加限制的。

cocos2d 几乎是由 Ricardo Quesada 一个人开发出来的。假如你想资助他对 cocos2d 进行一些后续开发,或者想购买他的一些收费源码项目,可以登录 <http://www.cocos2d-iphone.org/store> 访问 cocos2d Store。

### 1.1.2 开源

cocos2d 的第二个好处就是它是开源的,这就意味着可以自由地学习游戏引擎的源代码,或者在需要时对引擎作些改动。可以从 <http://www.cocos2d-iphone.org/download> 下载源代码。

### 1.1.3 Objective-C

另外, cocos2d 是用 Objective-C 编写而成的, Objective-C 是苹果公司用于开发 iPhone 应用程序的本地编程语言(native programming language)。由于 iPhone SDK 也是用 Objective-C 编写而成的,因此对于使用 cocos2d 的开发者来说,要理解苹果公司的官方文档和使用 iPhone SDK 提供的 API 并不困难。

其他很多有用的 API,如 Facebook Connect 和 OpenFeint,也是用 Objective-C 编写而成的,所以要集成它们也非常容易。

#### 注意:

相对于 Objective-C 而言,你可能更喜欢别的编程语言,但我还是建议你学习 Objective-C。我原本有很深的 C++和 C#背景,而且 Objective-C 语法乍一看还挺古怪,所以一开始我并不情愿去学这个据说是又陈旧又过时的编程语言。果不其然,有段时间我的日子过得相当挣扎,我必须摒弃已经养成的编程习惯和思维模式才能弄清楚怎样用

Objective-C 来写程序。

但是，千万不要因为困难就放弃学习 Objective-C。你确实需要花点时间去习惯它，但是这样的付出马上就会得到回报(只要教程和文档足够充分)。所以，再多努力都是值得的!

### 1.1.4 2D 游戏引擎

显然，cocos2d 中的“2d”已经表明了它是一个专注于开发 2D 游戏的引擎，这在当今众多的 iOS 游戏引擎中算是少见的。虽然它也可以用于加载并显示 3D 对象，但是你必须自己写出 3D 渲染代码，否则就只能采用别的方法来加载和显示 3D 模型了。我想说的是，iOS 设备是一个非常理想的 2D 游戏平台。它们通常比较容易开发，也比较容易理解。很多情况下，它们对硬件的要求比较低，因此你可以创建更色彩鲜明、更细致的图形。

### 1.1.5 物理引擎

目前有两种集成在 cocos2d 里的物理引擎可供选择：Chipmunk 和 Box2D。这两种物理引擎仅仅在编写它们的语言上有一些细微的差别：Chipmunk 是用 C 编写而成的，而 Box2D 是用 C++编写而成的，但他们的功能几乎完全一样。当然，如果仔细比较，还是会发现一些差别。不过，想要根据这些差别来作出选择，就必须对物理引擎的工作机制有非常深入的理解。通常，应该选择一个你觉得比较容易理解的且提供的文档比较好的物理引擎，所以大多数开发者都比较倾向于使用 Box2D。而且，因为 Box2D 也采用了面向对象的思想，所以与 Objective-C 一起使用会比较方便。

### 1.1.6 技术难度较低

游戏开发者最喜欢 cocos2d 的地方就在于它把底层的 OpenGL ES 代码封装得特别好。大多数图形都是用简单的精灵类(CCSprite)来显示的，而精灵对象又是根据图像文件创建的。也就是说，一个精灵对象就是一个具有缩放、翻转和着色能力的纹理，只要简单地对精灵对象相应的 Objective-C 属性稍作修改就可以完成这些效果。你并不需要关心 OpenGL ES 代码的具体实现，这就是 cocos2d 的美妙之处。

同时，你可以灵活地对任意游戏对象在任意时刻添加自己的 OpenGL ES 代码。而且，Cocoa Touch 中的用户界面元素在 cocos2d 中也是适用的。

cocos2d 引擎不仅封装了 OpenGL ES 的实现细节(这样就不用费尽心思去理解那些错综复杂的步骤了)，还对一些比较通用的操作进行了高度的抽象，其中包括一些实现起来需要大量 iPhone SDK 背景知识的操作。但是，cocos2d 并不会阻止你去接触底层的实现。

### 1.1.7 依然需要编程

总的来说，cocos2d 确实简化了 iOS 游戏的开发过程，但出色的编程技巧依然是需要掌握的。其他 iOS 游戏引擎，如 Unity、iTorque 和 Shiva，都着重于向用户提供工具箱和工作流来降低对编程能力的要求。使用这些引擎固然方便，但也失去了一些技术上的自由，



而且还得付费。至于 cocos2d，要使用它确实需要花些功夫，但是相比其他引擎，cocos2d 更能凸显出游戏编程的本质。使用 cocos2d 的开发者关注的是游戏编程的核心问题，同时由于 cocos2d 良好的封装性，他们又不必真正地去处理最底层的实现。

### 1.1.8 超棒的 cocos2d 社区

在 cocos2d 社区，总有人能很快地回答你的问题，并且开发者可以自由地在这里分享知识和信息。

在这里，几乎每天都有新的教程和源码示例发布，而且大多数都是免费的。你会发现网上实在有太多太多的资源可供学习、带来灵感。

一旦游戏制作完成并在 App Store 上架，就可以把它推荐到 cocos2d 网站上。这么做至少可以帮助你获得其他开发者的关注，幸运的话还会得到一些非常有价值的反馈信息。

#### 注意：

如果想要密切关注 cocos2d 社区的最新动态，建议你成为 Twitter 上 cocos2d 的粉丝 (<http://twitter.com/cocos2d>)。

如果想成为我的粉丝，可以访问 <http://twitter.com/gaminghorror>。

在 Twitter 的搜索框内输入“cocos2d”，然后单击 Save this search 链接。这样你平时只需一次单击就可以经常地查看 cocos2d 的新帖子了。通常来说，这么做能让你看到一些非常有用但却容易忽略的 cocos2d 的相关信息。而且，你一定会在这里结识不少使用 cocos2d 的开发者。

## 1.2 注意事项

我想到两个对 cocos2d 开发者来说非常重要的问题，所以先在这里提一下。

### 1.2.1 Section 3.3.1

虽然“Section 3.3.1”这个名称听起来有点像是电影《星际旅行》里的某个秘密政府组织，但它实际上是苹果公司的开发者许可协议中的一个章节。随着 iPhone SDK 4 的发布，它已经成为某新条款的代名词。该条款或多或少地规定了开发者只可以使用 Objective-C、C、C++ 或 JavaScript 来进行开发。由于对 iOS 开发所使用的编程语言加以限制，该条款一经宣布便在 iOS 开发者中间引发了大量的讨论和担忧。

由于 cocos2d 完全是用 Objective-C 编写而成的，而且其使用的外部函数库，如 Chipmunk 和 Box2D 这两个物理引擎分别是用 C 和 C++ 编写而成的，因此只要开发者是直接使用 iPhone SDK 提供的 API，并且没有使用任何私有 API，cocos2d 的使用者就不需要为这个新条款感到担心。苹果公司官方不会因为 Section 3.3.1 而否定用 cocos2d 开发出的游戏和应用。

至于苹果公司官方为什么要设立条款来限制编程语言并且杜绝“中间层”，大家普遍认为这主要是为了防止用 Adobe Flash 开发出来的应用和游戏在 iOS 市场占据太大份额。