移动与嵌入式开发技术

Learn cocos2d Game Development with iOS 5

iOS 5 cocos2d 游戏开发实战(第2版)

[美] Steffen Itterheim 著
[德] Andreas Löw 著
同济大学苹[illegible]部 译

Apress®

清华大学出版社

移动与嵌入式开发技术

iOS 5 cocos2d 游戏开发实战

(第 2 版)

[美] Steffen Itterheim
[德] Andreas Löw
著

同济大学苹果俱乐部 译

清华大学出版社

北 京

Steffen Itterheim，Andreas Löw

Learn cocos2d Game Development with iOS 5

EISBN：978-1-4302-3813-3

北京市版权局著作权合同登记号　图字：01-2012-6656

图书在版编目(CIP)数据

iOS 5 cocos2d 游戏开发实战(第 2 版)/(美)伊特海姆(Itterheim, S.)，(德)勒夫(Low, A.) 著；同济大学苹果俱乐部 译. —北京：清华大学出版社，2012.10

(移动与嵌入式开发技术)

书名原文：Learn cocos2d Game Development with iOS 5

ISBN 978-7-302-30303-9

Ⅰ. ①i…　Ⅱ. ①伊…　②勒…　③同…　Ⅲ. ①移动终端—游戏程序—程序设计　Ⅳ. ①TN929.53　②TP311.5

中国版本图书馆 CIP 数据核字(2012)第 237829 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：宋　林

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 刷 者：清华大学印刷厂

装 订 者：北京市密云县京文制本装订厂

经　　销：全国新华书店

开　　本：185mm×260mm　　**印　　张**：28.5　　**字　　数**：694 千字

版　　次：2011 年 12 月第 1 版　2012 年 10 月第 2 版　　**印　　次**：2012 年 10 月第1 次印刷

印　　数：1～4000

定　　价：59.80 元

产品编号：048812-01

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。他在 Doom 和 Duke Nukem 3D 社区表现活跃，并因此获得了他的第一份自由职业，成为 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。

Andreas Löw 在 10 岁的时候有了一台 Commodore C16，从那时起他就对计算机产生了狂热的兴趣。他自学了编写游戏的技术，并在 1994 年发布了自己的第一款游戏 Gamma Zone，这是一款针对 Commodore Amiga 平台的游戏，用纯汇编语言编写完成。在获得电子工程学的学位后，他进入 Harman International 公司，负责为汽车行业开发具有语音识别功能的导航和娱乐系统。他开发了自己的编程语言和开发工具，现在世界上采用语音识别技术的每辆汽车都在使用他的编程语言和开发工具。

iPhone 出现后，他有了回归本行的打算，开始开发一款叫做 TurtleTrigger 的游戏。他意识到 cocos2d 社区存在对好的开发工具的强烈需求。于是，利用自己在游戏和工具开发方面的知识，他开发出了 TexturePacker 和 PhysicsEditor，它们迅速成为 cocos2d 用户进行开发时必不可少的工具。

技术编辑简介

Boon Chew 是 Nanaimo Studio 的执行董事。Nanaimo Studio 位于西雅图和中国上海，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，喜欢与出色的人们一起工作。您可以通过 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。如果没有想到你们可能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！

我要感谢 Jack Nutting，他最先使我有了编写这本书的想法。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，与他交谈让我感到很愉快。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、Chris Nelson(他们是本书的开发编辑)和 Boon Chew(本书的技术编辑)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!Chris 为本书的第 2 版提供了重要的帮助，他指出了许多细微但却很关键的地方，使我能够进行修正。

感谢编辑 Kim Wimpsett。如果没有你的帮助，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。

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，并且是我的榜样。

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！

译者序

时隔一年，《iOS 5 cocos2d 游戏开发实战(第 2 版)》与大家见面了，而我也长大了一岁，告别了学术气氛浓郁的同济大学苹果俱乐部，来到了美国继续学习。到了美国，最令人称奇的是，近八成人手上拥有至少一台苹果设备，近五成人手上拥有 iPhone 手机。我想这无疑是对 Apple 开发人员的一种鼓舞。想象一下，当全球的苹果用户打开自己的 iPhone 或 iPad，开始为你开发的 Apple 应用痴狂时的感受吧！相信本书有助于你实现梦想。

一年中，随着 iOS 5 的推出，cocos2d 游戏引擎也有了翻天覆地的变化。很多旧方法被摒弃，新的技术得以崭露头角。同为 Apple 开发爱好者的我们，不要停止学习的步伐，只有跟上时代发展的潮流，我们才能开发出更具竞争力的应用或游戏。

此次再版，在保留第 1 版原有内容的基础上，添加了有关 cocos2d 与 UIKit 视图结合使用以及 Kobold2D 的内容，并对原有内容进行了技术细节上的修改。相信学习过本书第 1 版的人一定觉得受益匪浅，但是你仍然可以从本书中学到很多之前没有见过的技术和实现方法。没有看过第 1 版的人，也没有关系，因为本书将会手把手教会你从零开始，循序渐进地进入 cocos2d 世界。

本书的翻译、整理和审校工作由赵晓欣、张雪薇、王得希和我共同完成。特别感谢张雯莹老师为我们的翻译工作提供了大力支持和帮助。由于时间仓促，加之水平有限，书中难免会有错漏之处，希望读者能够提供反馈，我们将不胜感激。

霍 巨

2012 年 8 月 29 日

写于 Washington DC

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)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。我知道这个网站可以令所有人受益。

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1/4。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，尤其是在本书第二版更新内容的过程中，我学到了很多东西。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！

写作cocos2d图书的过程让我意识到cocos2d还有改进的空间。我强烈感觉到需要一个更好的、让游戏开发初学者更容易上手的cocos2d，因而创建了Kobold2D。本书第16章和www.kobold2d.com上都介绍了Kobold2D。读者可以放心的是，在本书中学到的几乎全部内容都适用于Kobold2D。

目　录

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——iOS 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 平台开发，或者刚刚进入游戏开发这一充满激情的领域。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 cocos2d 开发人员走到一起：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(Asteroid)，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!你可以在 Cocos2D Central(www.cocos2d-central.com)上分享你的游戏或者咨询一些问题，也可以通过我的邮箱(steffen@learn-cocos2d.com)联系我。我还创建了一个专门用来学习 cocos2d 的网站，网址为 http://www.learn-cocos2d.com)。现在我正在开发 Kobold2D，目的是对 cocos2d 加以改进，你可以在下面这个网址了解我的工作进展：www.kobold2d.com。

1.1 本书第 2 版的新增内容

首先，很高兴 Andreas Löw 也参与创作了本书的第 2 版。Andreas 是 TexturePacker 和 PhysicsEditor 工具的开发者，他尽心尽力地用新代码和更好的图像改进了本书多个章节中的项目。

最重要的是，第 2 版的目的是让本书跟上最新的开发成果，所以书中使用了 cocos2d 1.0.1 这一最终版本以及 Xcode 4 和 iOS 5。正文、代码和插图都已经用 cocos2d、Xcode 和 iOS SDK 的新版本做了更新。

第 2 版中还采纳了读者的反馈。我们对第 3 章做了大幅修订，增加了对 cocos2d 关键功能的介绍，而且增加了许多插图来说明关键的概念和类。本书中的配图数量在整体上增加了不少。

在过去一年中，出现了很多新的 cocos2d 游戏开发工具。为了反映这一点，本书使用 TexturePacker 取代 Zwoptex 作为主要的纹理图册创建工具。Löw 全职开发他的工具，所以推出更新和新功能的速度很快，而且他还提供了出色的支持，这让用户受益匪浅。第 2 版中还使用 PhysicsEditor 取代了 VertexHelper，因为它提供了更好的工作流以及更强大、更方便的功能。最后，第 2 版介绍了 Glyph Designer，这实际上是一个 Hiero Bitmap Font 工具，但是具有 Mac OS X 风格的用户界面，而且没有 Hiero 那么多 bug。

从第 6 章第一次引入，在第 7 章到第 9 章中一直使用的滚屏射击游戏的图形也做了彻底更新。它们比上一版中由我自己做的图形好看多了。类似地，第 13 章中弹球游戏的代码和图形也都做了更新。因此，这几章是第 2 版中变动最大的地方。

第 2 版中还增加了全新的两章：

- 第 15 章探讨了把 cocos2d 和标准的 UIKit 视图结合使用的可能性，而这一主题还没有被开发人员正确地认识，因为利用程度不够。在这一章中，你将学到如何在 cocos2d 应用程序中添加 UIKit 视图，以及如何在现有的 UIKit 应用程序中添加 cocos2d。学完这一章后，无论你以后打算使用 UIKit 还是 cocos2d，跨越两者之间的鸿沟都不是问题。
- 第 16 章介绍了 Kobold2D(www.kobold2d.com)，这是我为了让 cocos2d 成为一个更好的平台而做的努力。Kobold2D 的目标是简化频繁执行的任务，同时将预先配置好的库加入到发布版中，例如 wax(Lua 脚本)、ObjectAL(OpenAL 音频)和 cocos3d(3D 渲染)。它还提供了大量项目模板，其中许多都基于本书中讨论的项目。

1.2 选择 iOS 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。综合很多因素之后，我认为 cocos2d 对许多开发者来说会是一个非常棒的选择。

1.2.1 免费

首先，cocos2d 是免费的。不需要花钱就可以用它来进行开发。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。甚至还可以用它开发 Mac OS X 应用。说真的，这是完全没有附加限制的。

1.2.2 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做些改动。这使得 cocos2d 既可以扩展，又十分灵活。

可以从 www.cocos2d-iphone.org/download 下载源代码。

1.2.3 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 iOS 应用程序的原生编程语言(native programming language)。由于 iOS 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 iOS 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，你可能更喜欢其他编程语言，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!

1.2.4 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。

cocos2d 也可以用于加载并显示 3D 对象。事实上，现在它已经有了一个完整的增件，

叫做 cocos3d，这是一个开源项目，用于为 cocos2d 添加 3D 渲染支持。

但是我想说的是，iOS 设备是一个非常理想的 2D 游戏平台。时至今日，在 iTunes App Store 中发布的新游戏绝大多数仍然是全 2D 的。2D 游戏通常比较容易开发，算法也比较容易理解和实现。很多情况下，它们对硬件的要求比较低，因此你可以创建色彩更鲜明、更细致的图形。

1.2.5 物理引擎

目前有两种集成在 cocos2d 中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 语言编写而成的，而 Box2D 是用 C++语言编写而成的，但它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。

1.2.6 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。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，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 iOS 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或使用 iOS SDK 的一些功能。

1.2.7 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。