



译者：杨栋

邮箱：yangdongmy@gmail.com

第一章

介绍

有没有曾经梦想过自己可以编写一个电脑游戏，然后出售赚钱？现在通过为苹果的 iPhone, iPod Touch 和 iPad 编写程序，然后在 iTunes 程序商店（App Store）出售，你可以很容易地实现这个梦想。当然，编写游戏不是件容易的事情，你还要学习许多关于游戏开发和游戏编程的知识才行。不过因为你现在正在阅读本书，所以我相信你已经决定要参与到游戏开发中来。而且你已经选择了一个最有趣的游戏引擎：cocos2d for iPhone。

使用 cocos2d 的开发者们有着很多不同的背景。有些像我一样已经从事好多年甚至十几年的专业游戏开发工作。另一些则刚刚开始学习 iOS 编程，或者刚刚进入令人兴奋的游戏开发领域。不管你来自哪一个领域，你都可以通过本书学到一些东西。

所有 cocos2d 开发者的一个共同特点是：每个人都喜欢游戏，也喜欢创造和编写游戏。本书向此精神表示敬意。我也介绍了一些可以让开发过程变得容易的工具。最重要的是，你将从本书学习到如何制作游戏和将这些知识应用到真实的游戏开发中去。

我很反感那些花费所有的页面教人如何使用特定的 API 来制作另一款“小行星克隆”（Asteroids Clone）游戏的书。我想最重要的应该是游戏设计概念和工具。这些东西不会随着 API 和你的个人编程喜好的改变而改变。在过去的 20 多年时间里，我积累了几百本编程和游戏开发的书籍。现在我最看重的书是那些超越特定的技术，告诉我那些技术的设计原理。这本书不仅提供可以运行的代码，而且告诉你为什么这些代码可以运行和需要做出什么样的取舍。

我希望你可以学到如何编写有影响力的，在应用商店里出名的游戏。通过本书，我会带你理解这些游戏的理念和技术概念，也会告诉你 cocos2d 是怎样和 Objective-C 一起让你的游戏产生活力。本书的源代码有很多的注释，这些注释将会帮助你理解代码的确切含义。

当我学习一样新的事物时，通过学习别人带有注释的代码是对我最有效的学习方法。我想这个方法也同样适合你。因为你可以使用本书的代码作为你游戏的基础代码，我很期待可以在不久的将来玩到你的游戏！不要忘记告诉我你写的游戏！你可以发邮件给我：steffen@learn-cocos2d.com 或者访问我为学习 cocos2d 专门设立的网站：www.learn-cocos2d.com。

为什么使用 cocos2d for iPhone?

当游戏开发者寻找游戏引擎时，他们首先会衡量一下各个引擎。我觉得 cocos2d 是很多开发者非常好的选择，原因如下：

免费使用

首先，cocos2d 是免费的。你可以用它为 iPhone，iPod Touch 和 iPad 编写免费或收费的应用程序。你不需要付任何版税。

因为 cocos2d 基本上是 Ricardo Quesada 一个人创造的产品，你可以考虑向他捐款，或者购买他的一些收费源代码项目，以支持他进一步开发 cocos2d。你可以通过 cocos2d 的官方网站商店捐款或者购买 Richardo 的产品：

<http://www.cocos2d-iphone.org/store>。

开源

cocos2d 是开源的。你可以学习游戏引擎的源代码，或者在必要的时候修改代码。你可以在此下载 cocos2d: <http://www.cocos2d-iphone.org/download>。

完全使用 Objective-C 编写

cocos2d 是完全使用 Objective-C 来编写的。Objective-C 是苹果用来编写 iPhone 应用程序的原生编程语言。iPhone SDK 用的就是 Objective-C，这样你就可以更容易的理解苹果的文档和使用 iPhone SDK 的功能。

其他的一些诸如 Facebook Connect 和 OpenFeint 同样是用 Objective-C 编写的，因此这些功能也可以很容易整合到 cocos2d 里面。

注：即使你喜欢使用别的编程语言，我还是建议学习 Objective-C。虽然我已经使用 C++ 和 C# 很长时间，但是第一次看到 Objective-C 的语法时还是觉得很奇怪。一开始其实我不大情愿学习这种据说已经是古老和过时的编程语言的。我花费了不少时间来习惯它的语法，这要求我放弃很多既有的习惯和期望。

但是不要被 Objective-C 语法的表象分散你的注意力。你确实需要花些时间来习惯它的语法，但是很快你就会发现它的好处。即使只是为了看懂苹果的那一大堆文档，你也应该学习 Objective-C。我保证你不会后悔花时间学习 Objective-C！

它是 2D 的

当然，cocos2d 里面的“2D”是有意义的。相比于其他的 iOS 游戏引擎，cocos2d 专注于帮你制作 2D 游戏。你也可以自己编写 3D 渲染代码或者使用第三方的解决方案，在 cocos2d 里面加载和显示 3D 模型。但是我觉得 iOS 设备更加适合 2D 游戏。一般来说，2D 游戏更容易开发，也更容易让人理解。而且很多时候，2D 游戏对硬件的要求更小，这样就允许你创建更生动和拥有更多细节的游戏图形。

自带物理引擎

cocos2d 整合了两套物理引擎。一套叫 Chipmunk，另一套叫 Box2d。它们的功能基本相同，最大的区别是编写它们所使用的编程语言：Chipmunk 是用 C 写的，Box2d 是用 C++ 写的。当然，如果你真要找到它们的不同之处，你会找到一些。

不过这要求你理解物理引擎是如何工作的，从而做出选择。一般来说，你应该选择比较容易理解的，并且提供更好文档的物理引擎。对大多数开发者来说，Box2d 是比较好的选择，它面向对象的特性也更容易和 Objective-C 整合在一起。

降低了技术复杂度

游戏开发者喜欢 cocos2d 的一个原因是它隐藏了底端的 OpenGL ES 代码。大多数 cocos2d 游戏中的图形由简单的 Sprite 类利用图片文件生成。换句话说，一个 Sprite 就是一个贴图。你可以通过改变 CCSprite 类的 Objective-C 属性来改变这个贴图的大小，旋转和颜色。你不需要关心这些效果是怎样用 OpenGL ES 代码来实现的。

同时，cocos2d 允许你自由的在任何时间为任意游戏对象添加你自己的 OpenGL ES 代码。你还可以添加一些 Cocoa Touch 的用户界面元素到你的程序中。

cocos2d 不仅让你不用处理 OpenGL ES 带来的复杂性，而且也提供了常用功能高层次的抽象。这些功能可能需要你花费很多时间学习 iPhone SDK。不过 cocos2d 不会阻止你直接使用底层的功能。

需要编程知识

虽然 cocos2d 让游戏开发变得简单一些，但是说到底你还是需要很好的编程能力和知识。其他的一些 iOS 游戏引擎，比如 Unity, iTorque 和 Shiva 专注于提供工具和流程以降低对编程知识的要求。但是使用那些工具的同时，你也牺牲了灵活性，当然还有钞票。对于 cocos2d，虽然你需要花点时间学习，但是你是最接近于真正的游戏编程的，同时你不需要和引擎的核心打交道。

cocos2d 有很好的社区支持

你总能在 cocos2d 的社区中找到愿意提供帮助的人，快速得到答案。开发者们很愿意分享他们的知识和经验。

几乎每天都会有新的教程和源代码示例分享出来，而且大多数是免费的。你也可以在网上很多地方找到学习资源。

一旦你的游戏制作完成，并在应用商店上架。你甚至可以在 cocos2d 的官网上进行推广。至少你可以得到开发者们的关注和反馈。

注：我建议你跟随cocos2d在Twitter上的链接：<http://twitter.com/cocos2d>，你可以在第一时间得到来自cocos2d社区的消息。

你也可以跟随我的Twitter链接：<http://twitter.com/gaminghorror>。

打开Twitter页面后，在搜索框中输入“cocos2d”，然后点击“Save this search”链接，你就可以在以后一键查看与cocos2d相关的新闻了。你会经常看到在别的地方看不到的cocos2d新闻。你也可以在这里认识和你一样的cocos2d

开发者。

两条重要的关于cocos2d的花边新闻

我觉得cocos2d开发者需要知道以下两件事情：

Section 3.3.1

这个听起来有点像《星际迷航》中的秘密政府组织。其实Section 3.3.1是苹果开发者许可协议中的一节。当iPhone SDK 4发布时，这一节代表了苹果开发者协议内容的重要变化。此条政策限制iOS开发者只能使用Objective-C, C, C++和JavaScript来开发iOS应用程序。这条限制让iOS开发者议论纷纷，也引起了一部份人的担忧。

因为cocos2d是完全用Objective-C开发的，而且像物理引擎这样的外部库也是用C和C++开发的，并且不使用任何的私有API。所以cocos2d开发者不需要担心开发出来的应用会被苹果拒绝。

苹果的这条政策主要是用来防止使用Adobe Flash中间件开发出来的应用在iOS市场上占据一席之地。

移植到其他平台上

你可能注意到cocos2d的接口在其他平台上也存在，包括Windows和Android。虽然它们都用一样的名字和开发理念，但是它们都是用不同的语言由不同的作者所开发，和cocos2d for iPhone没有直接关系。比如，Android cocos2d的接口是用Java写的 - Java是开发Android应用的原生编程语言。

如果你想把你的游戏移植到其他平台，你首先要认识到不同平台上的cocos2d游戏引擎很不一样。比如移植你的游戏到Android上就不是件容易的事情。首先，语言不一样，你需要将所有Objective-C代码用Java重写。完成重写后，你还需要做出相应的修改以适应目标平台上不同的cocos2d API，甚至有些功能还不被支持。最后，任何移植都会有它们自己的bug，每个平台都有它们自己的技术限制和挑战。

总体上来说，移植iOS游戏到别的同样叫cocos2d游戏引擎的平台需要花费与开发全新软件一样的投入。不存在简单的移植。不同平台上的cocos2d引擎只是共享了相同的名字和理念而已。

这本书是为你写的

我可以想像到你拿起这本书是因为被它的标题所吸引。我想你可能是想给iPhone和iPad开发游戏，而你选择的游戏引擎就是cocos2 for iPhone。或者你并不关心使用什么样的游戏引擎，你只是想给iOS设备开发2D游戏。也有可能你正在寻找关于cocos2d的比较有深度的学习材料，因为你已经用它一段时间了。不管你选择本书的原因是什么，我敢确定你可以在这里学到很多东西。

先决条件

与所有的编程书籍一样，你需要符合一些先决条件，有一些几乎是必须的。

编程经验

本书唯一一个必须先决条件是你需要有一些编程经验。你应该理解一般的编程概念，比如循环，函数，类等等。如果你以前写过程序（最好用的是面向对象的语言），应该没有多大问题。

Objective-C

如果你有编程经验，但是可能从来没有用Objective-C写过任何程序。虽然你不需要知道Objective-C才能使用本书，但是知道一些基本的语法绝对可以帮到你。如果你熟悉C++，C#或者Java这样的面向对象语言，你可能可以在阅读本书的过程中慢慢熟悉Objective-C的语法。不过坦白地说，即使我已经有15年使用C++，C#和许多种脚本语言的编程经验，我发现自己并不能做到这一点。因为Objective-C总是有一些很小的，但是烦人的事情让你搞不明白，而且还会转移你的注意力。如果你碰到这样的事情，最好有可以随时查找的Objective-C资源。

在此我推荐一本很有用的Objective-C书籍：Mark Dalrymple和Scott Knaster写的《Learn Objective-C on the Mac》（由Apress出版）。此书可以帮助你学习Objective-C和Xcode。

苹果公司发布在网上的《Introduction to the Objective-C Programming Language》也是很好的参考。你可以在这里找到：

<http://developer.apple.com/mac/library/DOCUMENTATION/Cocoa/Conceptual/ObjectiveC/Introduction/introObjectiveC.html>

Objective-C的方括号可能看起来有点吓人。而且你可能也听说了一些关于内存管理的恐怖故事，而且iPhone还没有垃圾收集机制。

不过不要太担心。首先，Objective-C只是一套不同的衣服而已。虽然表面看上去不一样，但是底下的编程概念是和别的语言一模一样，像循环，类，继承和函数调用。术语可能不一样，比如：Objective-C开发者说“发送消息”，其实就是“调用方法”的意思。对于内存管理，cocos2d已经让它变得尽量简单，而我会帮助你理解如何使用简单和基本的规则来做内存管理。

你会学到什么

我会和你分享很多游戏开发经验，为你展示交互性游戏是怎样制作的。虽然在过去的20多年时间里我看了很多“参考手册”式的游戏开发书籍，但我相信学习编程不应该是记住API方法。那样的书和官方的API文档没有多大区别。20多年前我开始学习编程的时候，是通过查看一大堆编译器参考手册来学习的。那时候，编译器手册还是打印出来的，没有网上的版本，因为因特网还在它的婴儿期。所以所有的信息就堆在我的桌子上，足有15英寸厚，学习起来非常费劲。

到现在，我还是记不起绝大多数的API和方法，虽然很多是我以前熟知的。我还是要经常查看手册。经过20多年的编程，我知道最重要的一件事是：思想。好的编程思想和最佳实践会伴随你很久，而且可以用于任何编程语言中。而学习编程思想最好的方法就是理解设计，组织和编写代码时所做出选择的原因和逻辑。

新iOS游戏开发者将会学到什么

不要担心，我会让你慢慢接触到cocos2d最重要的几个方面。我会专注于那些你应该在编程时能够记起来的类，方法和概念，因为这些是cocos2d编程的基础。你将会学习一些与cocos2d配合使用的工具，比如Zwoptex和Particle Designer。没有这些工具，你只能成为“半个”cocos2d开发者。这些工具可以帮助你制作日益复杂的游戏。由于篇幅的限制，本书中的游戏例子是不完整的，也没有优化过。我也不可能讨论每一行代码，但是代码提供了注释以帮助你理解。

我把这些骨架游戏项目留给你去改进和提高。对于可能会出现的结果我感到很兴奋。在本书中我提供了很多种不同的骨架游戏项目，我想这样肯定好过用整本书来告诉你怎样制作一款经典的“小行星”游戏。

本书选择的游戏项目是基于它们在应用商店里的受欢迎程度和游戏开发者的相关性来决定的。游戏开发者经常会问我关于如何解决这些游戏里发现的一些特定问题。例如，目前画线游戏类型在cocos2d开发者当中很流行，但是画线游戏需要你解决一些很复杂的问题。

我看了很多其他开发者的cocos2d代码，也参加了很多关于代码设计，组织和风格的讨论。我的代码依赖于复合（Composition），而不是继承的架构。我将会解释为什么要复合而不是继承。另外一个经常被问到的与代码设计相关的问题是：不同的对象应该如何沟通。在这里，不同的代码设计和组织方式有各自的优点和缺点。我会将这些概念传递给你，以帮助你编写更稳定的，高效运行的代码。

iPhone程序开发者会学到什么

如果你已经用iPhone SDK开发过应用程序，你将会对如何使用其它工具（不是Interface Builder）制作游戏感兴趣。虽然这些工具没有苹果的耀眼，但是它们一样是非常有用的。

你在编程时所要考虑的事情也会改变。通常在游戏编程中你不会发送和接收很多的事件，并且你会让很多对象决定如何响应事件。因为提高运行效率和降低用户输入延迟的原因，游戏引擎常常联系的很紧密。很多工作会在循环和更新方法中完成。这些方法要么是每一帧都调用，要么是在特定的时间点调用。iOS应用属于用户界面驱动类型的程序，对于iOS应用，它需要发送和接收很多的事

件。所以当用户界面驱动的程序花费大多数时间等待用户输入时，即使玩家没有做任何事情，游戏也会在后台不断推送数据和像素。所以有很多事情在后台完成，而游戏代码也更加注重运行效率。

cocos2d开发者将会学到什么

如果你已经对cocos2d很熟悉，你可能会怀疑是否能从本书学到任何东西。我认为你会学到一些东西。你可以跳过第一章，但是你绝对会被接下去的游戏示例源代码所吸引。你将会学到我是如何组织代码和这样组织的原因。你也会在阅读不同类型的游戏制作方法的过程中得到灵感。你也能从很多我提供的诀窍中受益。

这本书包含的内容

以下是对本书中各个章节的简单描述。

第二章 — 开始学习

我们将会学习如何设置cocos2d以用于开发，安装项目模板和制作首个“Hello World”项目。你将会学习cocos2d的基础知识，比如场景（scenes）和节点（nodes）。

第三章 — 基础知识

我将解释基本的cocos2d类。这些是你最常用到的类，比如精灵（Sprites），过渡（Transactions）和动作（Actions）。当然，我会解释怎样使用这些类。

第四章 — 你的第一个游戏

敌人从天而降，你要倾斜你的设备以躲开敌人。这将是我们的第一个用加速计控制（accelerometer controls）制作的简单游戏。

第五章 — 游戏擦擦擦

我们在这一章制作一个大点的游戏，这个游戏要求有更好的代码结构。你将会学到如何将场景和节点叠加起来，如何通过不同的方式让游戏中的对象交换信息。

第六章 — 深入了解精灵（Sprites）

你将学习什么是“纹理贴图集”（Texture Atlas）和为什么我们要在游戏中用到它。你也会学到如何使用Zwoptex这个工具创建纹理贴图集。

第七章 — 滚屏射击游戏

准备好纹理贴图集以后，你将学习如何实现一款由触屏输入控制的滚屏射击游戏。

第八章 一 幄擦滚氍麈击牕戏

如果没有敌人，我们就没有什么可以用来瞄准的。本章会教你如何生成，移动，击中和使敌方集群动起来。

第九章 一 粒子效果

本章你将会学习使用Particle Designer（粒子效果设计师）工具来制作粒子效果。这些粒子效果将会被加入横向滚屏游戏中。

第十章 一 瓦片地图（Tilemap）基础知识

本章首先学习瓦片地图的基础知识和如何利用 Tiled Map Editor（瓦片地图编辑器）制作瓦片地图。接着，利用你在上一章横向滚屏游戏里学到的知识，你也将在本章学习如何制作一款流行的纵向无限向上跳跃的游戏。

第十一章 一无透视三维地图

因为cocos2d支持使用TMX文件格式，我们将在这里学习如何使用Tiled Map Editor制作基于瓦片地图的游戏。

第十二章 一物理引擎

你将学习如何通过移动手指头控制物体的移动方向。

第十三章 一弹钢珠游戏

这里介绍了如何使用Chipmunk和Box2D，还有如何使用它们制作一些很有意思的东西。

第十四章 一游戏中心（Game Center）

本章将讨论如何在一款反重力-行星反弹-空间射击游戏中使用真正的物理效果。虽然游戏不是非常的真实，但是我们将使用真正的物理效果。

第十四章 一结语

本书结束的地方，但是你的学习还没有结束。

问题和反馈

我可以帮助你进入cocos2d和iOS游戏编程领域，同时在这个过程中用高级游戏编程理念来挑战你。

如果你有任何疑问，请发邮件给我：steffen@learn-cocos2d.com。

我将继续在我的网站上补进任何书中遗漏的部份。我非常欢迎你的反馈！