

## <<Cocos2D应用开发实践指南>>

### 图书基本信息

书名：<<Cocos2D应用开发实践指南>>

13位ISBN编号：9787111425076

10位ISBN编号：7111425073

出版时间：2013-6-19

出版时间：机械工业出版社

作者：Rod Strougo, Ray Wenderlich

译者：李正阳

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

## &lt;&lt;Cocos2D应用开发实践指南&gt;&gt;

## 前言

前言你想成为一个游戏开发者吗？

为iPhone或iPad开发游戏可以获得极大乐趣。

这是少数几件能让你找回童年感觉的事情之一。

似乎每个人都有关于游戏的创意，还有比iPhone和iPad更好的开发平台吗？

游戏开发涵盖广泛的计算机科学技能—图形、音频、网络，有时让人应接不暇而失去信心，这是阻碍大部分人真正着手开发游戏的原因。

当首次开始时，能自如运用Objective - C语言可能已经是一个艰巨的任务，尤其是看到OpenGL ES、OpenAL和其他低层游戏应用程序接口时，会感到更加困难。

为iPhone和iPad开发游戏不需如此困难，的确如此。

对于简化2D游戏开发任务，没有比Cocos2D更有远见的做法了。

我们不再需要为制作iPhone游戏而不得以使用低层OpenGL编程API，同时你不需要是数学或物理专家。

当前有一个更快捷、更方便的途径—使用流行且免费的开源游戏开发框架Cocos2D。

Cocos2D非常有趣且易于使用，使用它可以跳过低层细节，专注于使游戏与众不同。

本书教授如何使用Cocos2D制作自己的游戏，现在就引导你一步步领略制作一个真正的苹果应用软件商店（App Store）游戏的过程！

本书中编写的游戏名为“Space Viking”，讲述了一个厉害的维京海盗被传送到陌生星球的故事。

在游戏制作过程中，可以获得使用Cocos2D框架所有重要元素的直接经验，并了解到每个元素如何相互协作而成为一个完整的游戏。

下载游戏你可以从苹果应用软件商店下载Space Viking游戏。

这个游戏是免费的，所以继续并且下载游戏，开始运行游戏，看看你是否足够厉害，可以将游戏打通。

这部史诗般教程将向你展示如何使用Cocos2D制作一款真正的游戏。

书中有许多代码可供实践，与此同时我们会解释每个步骤。

当你读完本书并完成相应的编码后，你将完成一个完整的游戏。

最棒的是，你将获得制作自己的游戏项目所需的知识和信心。

书中每一章会详细介绍游戏中的一个特定组件，以及支撑它所需的技术，不管是瓷片地图编辑器（tile map editor），还是我们用Cocos2D、Box2D及Chipmunk创建的一些特效。

在完成功能和技术介绍后，该章还会详细介绍如何在Space Viking游戏中实现该组件。

这种理论与实践相结合的方式填补了其他游戏开发书籍的空白。

Cocos2D是什么Cocos2D是一个开源的、基于Objective - C开发语言的、专注于iOS和Mac OS X平台的2D游戏开发框架，可以为iPhone、iPad、iPod Touch、Mac等硬件设备开发游戏。

Cocos2D既可以作为库文件导入Xcode创建的项目中，也可以在使用Cocos2D模板创建新游戏时自动添加进去。

Cocos2D框架使用OpenGL ES类库进行图形渲染，让用户在目标设备上获得图形处理器（GPU）的极高速度和性能。

Cocos2D框架还有许多其他特性和能力，通过本书你会逐渐了解到。

Cocos2D起初是一个用于2D游戏编写的Python框架。

2008年年底，它被移植到iPhone平台，并使用Objective - C重写。

现在已被广泛移植到Ruby、Java（Android）和Mono（C#/.NET）上。

注意Cocos2D有一个充满活力的、积极的支持者与贡献者社区。

Cocos2D论坛非常活跃，是学习和解决问题的一个非常优秀的资源，同时该论坛积极跟踪着Cocos2D的最新发展。

为什么应该使用Cocos2DCocos2D框架可使开发者避免纠缠于低层API，集中精力在游戏项目核心上。App Store这个市场流动性很强，经常发生变化。

## &lt;&lt;Cocos2D应用开发实践指南&gt;&gt;

快速创建游戏原型并将其开发出来是在App Store获得商业成功的关键，而Cocos2D是避免陷入学习OpenGL ES或OpenAL的泥潭，快速开发游戏项目的最佳工具。

Cocos2D也同样自带大量工具类，例如TextureCache（纹理缓存），它能自动缓存图形，支持游戏快速平滑播放。

TextureCache在后台运行，是许多你甚至不用知道如何使用的Cocos2D功能中的一个，这些功能对你来说是显而易见的。

其他有用的工具包括字体渲染、精灵表（sprite sheet）、强大的声音系统等。

Cocos2D是一个极佳的原型开发工具，能帮助你在一小时内（无论如何不会超过你阅读第2章所花费的时间）快速开发出属于你的游戏。

你阅读本书是因为你想要为iPhone、iPad设备开发游戏，而毫无疑问，使用Cocos2D是达到此目的的最快方式。

Cocos2D关键特性你仍然不确定Cocos2D是否合适？

好吧，看看这些可以使你在制作下一个游戏时轻松许多的Cocos2D的惊人特性。

动作动作（Action）是Cocos2D中最强大的功能之一。

动作可以让你轻松地移动、缩放、控制精灵和其他对象。

举个例子，在5秒内平滑地移动太空货运飞船，在屏幕上向右前进400像素的距离，需要的所有代码如下：就是这样，只有两行代码！

图P. 1阐释了作用于太空货运飞船的moveAction。

图P. 1阐释了作用于太空货运飞船精灵的moveAction的效果在Cocos2D中有许多内置动作：旋转、缩放、跳跃、闪烁、淡入淡出、着色、动画序列等。

也可以把一些动作串联在一起，并且用很少的代码调用自定义回调得到很棒的效果。

内置字体支持Cocos2D使得文本的处理变得很方便，这对于游戏的菜单系统、分数显示、调试，以及更多功能十分重要。

Cocos2D包含嵌入TrueType字体支持，也包含一个快速位图字体渲染系统，所以只需要几行代码就可以在屏幕上显示文本。

庞大的特效库Cocos2D包含一个强大的粒子系统，可以很方便地为游戏添加诸如烟雾、火焰、雨滴和雪花等很酷的特效。

Cocos2D也内置了诸如翻转、渐现渐隐等游戏中屏幕间转换的特效。

理想的瓷片地图游戏支持Cocos2D包含对瓷片地图（tile - mapped）游戏的内置支持，当拥有一个由微小的可复用图片组成的巨大游戏世界时，这是非常有用的。

Cocos2D也使得来回移动摄像机（camera）以滚动背景或层变得很方便。

最后，Cocos2D也支持视差滚动，从而赋予游戏3D深度错觉和透视效果。

音频 / 声音支持Cocos2D中的声音引擎允许你方便地使用OpenAL的强大威力，而不用关心低层API。

使用Cocos2D声音引擎，只需要一行代码即可播放背景音乐或音效。

两个强大的物理引擎Cocos2D捆绑了两个强大的物理引擎，Box2D和Chipmunk，它们对游戏来说都是很棒的。

使用这两个物理引擎，你可以为游戏营造前所未有的真实感，并创建出全新的游戏类型—而不需要是一个数学大师。

重要概念在开始之前，重要的是确保你熟悉关于Cocos2D和游戏编程通用的一些重要概念。

精灵可以看到术语“精灵”（sprite）经常使用在游戏开发中。

精灵是一个可以独立于其他图像在屏幕上移动的图像，可以是玩家角色、敌人，或者是用于背景中的一张大图。

实际上，精灵主要由PNG或PVRTC格式的图像文件构成。

一旦被加载到内存中，精灵通过iPhone GPU转换成纹理，渲染到屏幕上。

单例单例（singleton）是一个特殊类型的Objective - C类，它只有一个实例。

一个例子是，iPhone应用委托类（Application Delegate class），或Cocos2D中的Director类。

在代码中调用一个单例实例时，无论是哪个类调用它，总是会获取到这个类的唯一一个实例对象。

## <<Cocos2D应用开发实践指南>>

OpenGL ES OpenGL ES是一个移动版本（ES表示嵌入式系统，Embedded System）的开放图形语言（Open Graphics Language，OpenGL）。

它是你在iPhone或iPad上将0和1发送给GPU的最常用语言。

OpenGL ES是在iPhone或iPad上渲染图形的最快方式，由于它的起源，它是一个低层的应用程序编程接口。

如果你是一个游戏开发新手，会觉得OpenGL ES很难学习，但是幸运的是使用Cocos2D不需要了解OpenGL ES。

在iPhone和iPad上，支持两个版本的OpenGL ES，即OpenGL ES1.1和2.0。

Cocos2D中计划支持OpenGL ES2.0，但当前只支持1.1版本。

语言和屏幕分辨率Cocos2D由Objective-C语言编写而成，与编写Cocoa Touch类库和主要的Apple iOS API的语言相同。

在Objective-C中，了解一些基本的内存管理技术是很重要的，因为它是让你在iOS平台上成为一个高效游戏开发者的基础。

Cocos2D支持从最早期的iPhone到iPad再到iPhone4上的视网膜屏（retina）显示，支持所有设备的原生分辨率。

2D与3D要想跑得先学会走。

游戏开发也是如此，在潜心于3D游戏的更深层概念之前，要学习如何制作2D游戏。

在Cocos2D中有一些3D特效和转换，例如一个3D声波特效和一个轨道摄像机移动，然而，大部分功能是面向2D游戏和图形的。

Cocos2D是为2D游戏设计的（因此名字中有2D字眼），所以本书中的指南和例子也是2D的。

如果想要制作3D游戏，你要考虑不同的框架，例如Unity、Unreal Engine（虚幻引擎），或直接使用OpenGL。

本书背后的游戏：Space Viking本书将带你体验为iPhone和iPad创建一个全功能的基于Cocos2D的游戏的全过程。

在本书中创建的游戏名为Space Viking。

如果想要现在尝试一下游戏，可以在App Store下载一个游戏的免费版本，并且在iPhone、iPod touch或iPad上安装它。

当然，如果你更有耐心，可以在学完本书后，构建属于自己的游戏，并且在设备上加载它。

没有比有能力测试自己构建的游戏更美妙的学习经历了。

你不仅可以学到如何构建游戏，还可以回过头去，调整一下代码以改变相关的事情，去看看对游戏体验有哪些具体影响。

好东西只给善于等待的人。

本书将教你如何使用Cocos2D的全部特性和能力，但更为重要的是，如何把它们应用于真正的游戏中。

在学完本书后，你将拥有在App Store发布游戏的必要知识和经验。

从构建Space Viking游戏学得的观念适用于从动作到解谜等不同种类的游戏。

Space Viking的故事每个游戏都从创意开始，用角色和故事情节将其转化成游戏。

这里是Space Viking的故事。

在未来，地球上的人类后裔被迫殖民到太阳系之外的行星。

为了创造气候宜人的环境，庞大的星际机器在北欧和格陵兰岛采掘巨大的冰块，横跨星系运送到这些星球。

科学家们不清楚，在这些冰块的一块中包裹着北欧海盗Ole，他在亿万年前，在击败入侵蛮族回家的路上跌入了冰冷的河水中，被装在一个冰冷的坟墓中过了许多世纪，由于外星太阳的温暖，Ole在几千年后苏醒—距离家乡几光年之外，如图P.2所示。

你开始扮演维京海盗Ole，并且与外星人战斗在陌生的世界中，希望找到一条路让Ole回到他的故乡和时代。

使用屏幕左侧的拇指游戏操纵杆，控制Ole左右移动。



## &lt;&lt;Cocos2D应用开发实践指南&gt;&gt;

右侧是跳跃和攻击按键。

开始Ole只使用双拳。

在后来的关卡（level）Ole找到了他信赖的木槌，在物理关卡使用加速计控制Ole。

Space Viking是一个动作冒险类游戏，动作是其重点。

我们的目标是从无到有创建一个真正的游戏，所以可以学习到不只是Cocos2D的知识，还有如何在一个全功能的游戏中使用它。

游戏创意来自概念艺术家、图形艺术家Eric Stevens和其他一些资深游戏玩家，在开发早期我们共同讨论了这个游戏创意。

Space Viking由许多关卡构成，每个关卡展示Cocos2D或游戏类型的一部分。

例如，第一关是一个横向滚动的生死战（beat抗m up），第四关是一个小车竞速关卡，这个关卡展示了来自Box2D和Chipmunk引擎的游戏物理效果。

我们希望你可以在完成本书后，复用Space Viking游戏的一部分，制作自己的游戏！

没错，你可以自由地复用本书中的代码来构建自己的游戏。

本书的组织结构本书的目标是随着构建Space Viking游戏讲解关于使用Cocos2D开发游戏知识（了解更多关于探索和维京海盗Ole的故事）。

我们将从一个简单的关卡和一些基本的游戏机制开始，通过努力提升以创建使用物理系统和粒子系统的关卡，最终到本书末尾完成游戏。

首先学习Cocos2D的基础知识，用Ole的基本的奔跑和跳跃移动创建一个小的关卡。

第二部分展示如何为Space Viking添加动画序列（animation）、动作、特效和文字。

第三部分游戏更进一步，添加更多关卡和场景、声音和滚动的游戏体验。

在第四部分中，用Box2D和Chipmunk物理引擎把真实感带到游戏中。

最终在第五部分中，学习如何添加粒子系统、添加高分榜、连接到社交网络，以及调试优化Space Viking以为未来构建游戏找出最佳实践。

本书中有17章和一个附录，每章处理创建Space Viking的一个特定部分。

第一部分学习如何安装Cocos2D并且开始使用它创建Space Viking。

学习如何为Ole和他的敌人添加动画序列，并让他们运动。

第1章本章介绍如何在Xcode中安装Cocos2D框架和模板，以及一些方便游戏开发的配套工具。

这些工具都可以免费使用，并且有助于创建Cocos2D使用的元素。

第2章这里创建了Space Viking的基础，这个游戏贯穿全书。

起初只是一个基本的Cocos2D模板，并且添加英雄（维京海盗Ole）到屏幕上。

在本章的第二部分，添加处理触摸输入的方法，包括来回移动Ole，并使他跳跃。

第3章在本章中，通过给Ole在屏幕上移动时添加动画序列，学习如何让游戏看起来更具有真实性。

第4章在本章中，学习如何实现简单的碰撞检测，以及添加第一个敌人到Space Viking游戏中，这样Ole可以开始战斗以离开这个星球！

第二部分学习如何为Ole的战斗创建更复杂的敌人，在此过程中学习Cocos2D动作和特效。

使用Cocos2D文本功能，完成一个现场的、屏幕上的调试系统。

第5章动作是Cocos2D中的一个关键概念—是来回移动对象、使它们变大或者消失等的简单方式。

在本章中，通过添加道具和武器到关卡实践它们（动作），并学习其他一些重要的Cocos2D能力，例如特效和调度器（scheduler）。

第6章大部分游戏中都有文字，Space Viking也不例外。

在本章中，学习使用Cocos2D中不同的可用方法，将文字添加到游戏中。

第三部分学习如何通过添加菜单、使用声音和滚动，来扩展已经构建完整的Space Viking游戏关卡。

第7章几乎所有的游戏都不止一个“屏幕”（screen，或称“场景”（scene），在Cocos2D中称为场景），通常至少是主菜单、主游戏场景、关卡完成和致谢场景。

在本章中，通过在Space Viking中实现它们来学习如何创建多场景。

第8章在游戏中添加音效和音乐，可以大大提升游戏体验。

Cocos2D使用CocosDenshion声音引擎使这变得十分简单，所以本章中，尝试一下它！

## &lt;&lt;Cocos2D应用开发实践指南&gt;&gt;

第9章大部分游戏都有一个单一屏幕不能容纳的世界，所以这个世界需要随着玩家的移动而滚动。正确处理这个问题是一个技巧，所以本章展示如何转换一个“生死战”为横向滚动（sidescroller），使用Cocos2D瓷片地图以提高效率。

第四部分Box2D和Chipmunk物理引擎包含在Cocos2D中，可用来为游戏添加惊人的特效，例如重力、逼真的碰撞，甚至人偶动画（ragdoll）效果。

在这个部分将为Space Viking添加一些基于物理（physics - based）的从简单到高级的关卡。

第10章正如Cocos2D可以使你在不了解OpenGL低层细节的情况下，很容易地为iPhone制作游戏一样，利用Box2D你可以很容易地为游戏对象添加物理效果，而不必是一个数学专家。

在本章中，通过创建一个对象根据重力移动的有趣的解谜游戏，来学习如何开始使用Box2D。

第11章在本章中，通过启动横向滚动小车竞速游戏，展示了一些Box2D可以处理的纯粹的物体。

在此过程中，你将学习如何模拟任意形状的图形、添加物理形体的关节移动约束，以及更多内容！

第12章在本章中，通过添加要躲避的尖刺和史诗般的关底战，为小车竞速关卡添加更多令人惊叹的效果。

将学习关节、如何检测碰撞，以及如何为敌人添加逻辑。

第13章Cocos2D自带的第二个物理引擎名为Chipmunk，与Box2D相似。

本章展示了如何使用Chipmunk引擎，将其与Box2D进行比较，以及通过制作一个银河战士风格的逃逸关卡，帮读者获得直接的实践经验。

第五部分学习如何快速创建粒子系统并将其添加到游戏中，如何为在线高分榜和成就系统整合苹果游戏中心（Apple的Game Center），以及一些性能主题和技巧以使游戏运行得更快。

第14章使用Cocos2D的粒子系统可以为游戏添加一些令人惊叹的特效—极其简单！

在本章中，将学习如何使用粒子系统为Space Viking添加一些特效，例如飞船尾焰。

第15章利用苹果游戏中心，可以十分方便地为游戏添加成就系统和高分榜，对玩家来说会有更多的乐趣，也可能会帮助提高销量。

本章介绍如何在Space Viking中一步步设置。

第16章在本章中，将学习如何解决优化中要面对的一些常见挑战和问题，以及最大程度优化和完善Cocos2D游戏。

你可以获得最常用的调试性能问题的经验和解决方案。

第17章最后一章回顾了所学的知识，并且介绍下一步的方向：3D，在其他平台（例如Android）使用Cocos2D以及更多高级的游戏开发主题。

附录附录提供了一个Cocos2D中将要使用的主要类的概述。

此时你已经完成了本书的阅读，将拥有从头制作一个很棒的游戏的实际经验！

可以采用所学到的概念（甚至是一些代码），把自己的游戏变成现实。

本书的读者本书的读者包括那些迟迟不愿进行游戏开发的开发人员，因为他们认为学习会很漫长、很复杂。

许许多多的开发者有编写游戏的意愿，但是不清楚从哪里入手游戏开发或Cocos2D框架。

本书是一本指南，引导你从非常初级的Cocos2D使用到运用Box2D和Chipmunk物理引擎中的高级物理概念。

本书是针对有兴趣在包括iPhone、iPad和iPod touch等iOS设备上创建游戏的开发者。

本书假定读者对Objective - C、Cocoa Touch和Xcode工具有基本的了解。

不需要了解任何低级应用程序接口（Core Audio、OpenGL ES等），因为这些都是Cocos2D内部使用的。

本书的目标读者如果你已经在为其他平台的iPhone开发应用程序，但是想要从功能应用转移到游戏，那么本书很适合你。

本书基于读者已掌握的开发知识，通过介绍术语、技术和必要的工具，以及一个真实的游戏实现例子，引领读者步入游戏开发的世界。

非本书的目标读者如果读者已经理解创建一个游戏必要的工作流，或者有一个坚定的游戏创意必须要为3D图形使用OpenGL ES，那么本书不适合这样的读者。

## &lt;&lt;Cocos2D应用开发实践指南&gt;&gt;

我们期望，在阅读本书之前，读者已经熟悉了Objective - C、C、Xcode和Interface Builder。

虽然本书中所描述的实现已经尽可能简单，对C语言的使用也是最低限度的，但这些语言的坚实基础是必备的。

以下书籍可以帮助提供通读本书的必要背景知识：这些书和其他能在网络上找到的资源，将帮助读者学习更多在Mac和iPhone上编程的方法，让读者更深入地了解Objective - C语言和Cocoa框架的知识。源码、视频指南和论坛获取信息不局限于本书。

还有完整的（也包括Space Viking的全部源代码）源码和视频指南可帮助读者掌握每章介绍的概念。贯穿全书有大量的代码和习题，所以假定读者已经获取了苹果开发工具，例如Xcode和iPhone SDK。所有这些工具都可以从苹果iPhone开发中心下载。

在阅读本书时，如果想与同行交流，可以访问本书的论坛。

致谢如果没有以下这些人的辛勤工作、热心投入以及全力支持，本书不会完成：首先，感谢我们的编辑Chuck Toporek及其助手Olivia Basegio。

Chuck在完成本书的过程中给予了我们太多的耐心帮助和鼓励（尽管我们两个都是第一次写书），并协调管理了把一个简单的Word文档变成读者今天手中所捧的这本书的全部工作。

Olivia在整本书编写过程中，对成员间的工作协调以及技术评审有着巨大的贡献。

再次感谢你们使本书得以问世！

另一位我要感谢的人是Addison - Wesley公司的Chuti Prasertsith，他为本书精心设计了封面。

衷心感谢Cocos2D的主开发者兼协调者Ricardo Quesada（也被称为Riq），以及其他的Cocos2D贡献者，如Steve Oldmeadow及其他同仁。

如若没有Riq团队的苦心经营和他们的奉献精神，便不会有Cocos2D今天所拥有的完善框架和社区，更不会有这本书的诞生。

同时，我们相信Cocos2D已经给很多人的生活带来了颠覆性的改变，使他们梦想拥有专属于自己游戏的愿望得以实现。

Riq把Cocos2D作为他的全职工作，如果你想感谢他为本书所做出的辛勤付出，可以登录[www.cocos2d-iphone.org/store](http://www.cocos2d-iphone.org/store)为他捧场。

Riq也出售他的游戏Sapus Tongue的源代码，以及一个名为Level - SVG的优秀物理编辑器。

感谢Erin Catto（Box2D首席开发者）和Scott Lembcke（Chipmunk首席开发者）在非凡的物理引擎库上的工作。

如Riq对Cocos2D所做出的贡献一样，Erin和Scott的团队使无数程序员得以更加轻松快捷地写好一款很炫酷的基于物理的游戏。

Erin和Scott极其投入地专注于对物理库和社区的支持与维护，并抽出时间来评审本书中物理引擎相关的章节。

十分感谢Cocos2D声音引擎CocosDenshion的首席开发者Steve Oldmeadow。

Steve评审了音频章节。

Steve的付出使很多游戏开发者可以快速轻松地他们的游戏添加音乐和音效。

Eric Stevens是美国杰出艺术家、兼职游戏插画家。

他和我们紧密合作，制作了游戏中的所有音效（包括Ole苏醒）。

音乐家Mike Weiser为Space Viking游戏制作了摇滚声轨和音效。

我们认为音乐使Space Viking游戏与一般游戏有巨大的不同，真正给了我们期许的音乐气氛。

特别感谢Andrew Peplinski的维京海盗呼啸声和Rulon Brown带来的在游戏开始时听到的和声。

万分感谢我们的技术审校：Farim Farook、Marc Hebert、Mark Hurley、Mike Leonardi，以及Nick Waynik。

这些工作伙伴及时地纠正在整个过程中我们正要犯下的愚钝错误，并为我们提供了很多的意见，这使我们更好地完成了全书的每一章，非常感谢你们，我的伙伴们！

我们每一个人心中都还有很多感谢的话想说。

Rod Strougo的致谢十分感谢我的妻子和家人在我全身心投入此书的写作中时，对我能够如此耐心。

曾经有无数个夜晚，我埋头在办公室里，写作、编辑以及写代码。

## <<Cocos2D应用开发实践指南>>

假如没有Agata的支持和理解，这本书将不可能存在。

我的大儿子Alexander在这本书完成的过程中，也扮演着十分重要的角色，他协助完成了Space Viking的测试工作，我的孩子Anton在我完成最后一章的时候出生了。

Agata，感谢你所给予我的所有鼓励、爱和支持。

我还要感谢Ray的参与，Ray完成了Box2D、Chipmunk和Game Center章节的编写。

Ray非常出色地完成了对Box2D和Chipmunk的深度覆盖，并为Space Viking增添了很多有趣的关卡。

Ray Wenderlich的致谢首先，十分感谢我的妻子，也是我最好的朋友，Vicki Wenderlich，感谢她在我写作本书的整个过程中对我一如既往的支持，给我鼓励并提供宝贵意见。

没有她的帮助，我今天无法制作iOS应用，它们不可能像现在这样出色！

此外，也要感谢我的家人。

你们信任我，伴我走过作为一名iOS独立开发者的起起伏伏，支持我一路走来。

非常感谢你们！

最后，感谢我的iOS教程博客的所有读者和支持者。

没有你们的关注、鼓励和支持，我将失去继续完成教程的动力，也许将会和出版此书的机会失之交臂。

万分感谢你们，让这本书的出版成为现实，同时我也希望你们都能喜欢这本书。



## <<Cocos2D应用开发实践指南>>

### 内容概要

资深iOS开发工程师撰写，Box2D创始人Erin Catto等多位专家联袂推荐  
系统讲解Cocos2D中的各种功能组件及技术实现细节，详细介绍Box2D、Chipmunk两大物理引擎的使用方法和粒子系统的原理，以及iOS游戏的性能测试和优化  
以一个真实有趣的游戏案例Space Viking为主线，全面展示了Cocos2D游戏开发的流程、方法和技巧，实战性强

## <<Cocos2D应用开发实践指南>>

### 作者简介

作者：（美国）Rod Strougo （美国）Ray Wenderlich 译者：李正阳Rod Strougo，资深iOS开发工程师和游戏开发工程师，精通Cocos2D和Box2D等移动游戏开发引擎，经验丰富。

Rod接触物理引擎和游戏开发的时间非常早，可以追溯到早期的Apple II，当时他使用Basic编写游戏。

Rod是Prop Group工作室的创始人和首席开发者，近年来热衷于帮助其他人开始游戏制作之路。

Ray Wenderlich，资深iOS开发工程师和游戏开发工程师，精通Cocos2D和Box2D等移动游戏开发引擎，经验丰富。

Ray是Razeware LLC创始人，著有一系列iOS应用和苹果应用开发相关的教程，涵盖Objective - C、iPhone/iOS、Cocos2D、Corona、Unity和OpenGL等众多主题。

## &lt;&lt;Cocos2D应用开发实践指南&gt;&gt;

## 书籍目录

本书赞誉译者序前言第一部分 Cocos2D初接触第1章 你好, Cocos2D11.1 下载和安装Cocos2D21.1.1 下载Cocos2D21.1.2 安装Cocos2D模板31.2 创建你的第一个Cocos2D应用HelloWorld31.2.1 查看Cocos2D模板41.2.2 构建Cocos2D HelloWorld项目41.2.3 使HelloWorld更上一层楼61.2.4 添加移动71.3 为更好奇的读者:理解Cocos2D HelloWorld81.3.1 场景和节点81.3.2 开始101.3.3 进一步探索Cocos2D源代码141.4 安装CCHelloWorld到iPhone或iPad上151.4.1 让Xcode为你做所有事情161.4.2 为iPhone或iPad构建应用171.5 小结171.6 习题17第2章 你好, Space Viking182.1 创建Space Viking项目182.2 创建Space Viking类对象192.3 创建背景层202.4 GamePlayer层:把维京海盗Ole加入游戏中232.5 GameScene类:把层与场景连接起来252.6 Cocos2D Director262.7 添加运动272.7.1 导入JoyStick类282.7.2 添加JoyStick和按键282.7.3 将JoyStick运动应用于维京海盗Ole312.8 纹理图集352.8.1 纹理和纹理图集的技术细节362.8.2 创建场景1纹理图集382.8.3 添加场景1纹理图集到Space Viking412.9 为更好奇的读者:测试CCSpriteBatchNode412.10 在iPhone3G和老旧设备上调校性能432.11 小结432.12 习题43第3章 Cocos2D动画和动作概述453.1 Cocos2D中的动画453.2 Space Viking设计基础493.3 Cocos2D动作和动画基础523.4 使用属性列表文件存储动画数据533.5 组织、常量和通用协议553.5.1 创建常量文件563.5.2 通用协议文件573.6 GameObject和GameCharacter类593.6.1 创建GameObject593.6.2 创建GameCharacter类643.7 小结663.8 习题66第4章 简单碰撞检测和第一个敌人674.1 创建RadarDish和Viking类674.1.1 创建RadarDish类674.1.2 创建Viking类724.2 最终步骤864.3 小结924.4 习题92第二部分 更多敌人和更多乐趣第5章 更多动作、特效和Cocos2D调度器935.1 道具945.1.1 Mallet道具945.1.2 Health道具965.1.3 空间货运飞船985.2 敌人机器人1005.3 添加PhaserBullet115.4 GameplayLayer和Viking更新1145.5 为更好奇的读者:Cocos2D中的特效1185.5.1 Space Viking中有趣的特效1185.5.2 运行EffectsTest1205.5.3 将精灵和对象恢复正常1205.6 小结1215.7 习题121第6章 文字、字体和书面语1226.1 CCLabelTTF类1226.2 理解锚点和对齐1236.3 CCLabelBMFont1256.4 使用Glyph Designer字体设计器1256.5 使用Hiero字体构建工具1266.6 为更好奇的读者:实时调试1296.6.1 更新EnemyRobot1296.6.2 更新GameplayLayer1326.6.3 文字调试器的其他应用1336.7 小结1336.8 习题133第三部分 从关卡到游戏第7章 主菜单、关卡完成和致谢场景1357.1 Cocos2D中的场景1357.2 GameManager概述1367.3 Cocos2D中的菜单1447.4 场景组织与图像1447.5 创建主菜单1467.5.1 创建MainMenuScene1467.5.2 MainMenuLayer类1477.6 其他菜单和GameplayLayer1537.6.1 导入Intro、LevelComplete、Credits和Options场景及层1537.6.2 GameplayLayer1537.6.3 修改SpaceVikingApp-Delegate类1557.7 为更好奇的读者: IntroLayer类和LevelComplete类1567.8 小结1577.9 习题157第8章 添加音效1588.1 CocosDenshion概述1588.2 导入并设置音频文件1588.2.1 为Space Viking添加音频1598.2.2 音频常量1598.3 同步异步加载音频的比较1618.3.1 同步加载音频1618.3.2 异步加载音频1638.3.3 添加音频到GameManager1638.4 添加SoundEngine到GameObject1748.5 为EnemyRobot添加声音1778.6 为Ole添加音效1808.7 为菜单场景添加音乐1858.7.1 为Gameplay添加音乐1858.7.2 为MainMenu添加音乐1858.8 为更好奇的读者:如果你需要更多音频控制1858.9 小结1868.10 习题186第9章 当世界变得更大:添加滚动1879.1 为更广阔的世界添加逻辑1889.2 创建更广阔的世界1909.2.1 创建第2个游戏场景1919.2.2 创建滚动层1979.3 与视差层一起滚动2039.4 无限制滚动2069.4.1 创建滚动层2069.4.2 创建平台场景2149.5 瓷片地图2169.5.1 安装Tiled工具2179.5.2 创建瓷片地图2189.5.3 Cocos2D压缩TileMap类2209.5.4 添加TileMap到ParallaxNode2229.6 小结2249.7 习题225第四部分 物理引擎第10章 基础游戏物理学:用Box2D增加现实性22710.1 开始22710.1.1 关于死亡的疯狂梦想22910.1.2 创建新场景22910.1.3 添加Box2D文件到你的项目23110.1.4 Box2D单位23510.2 你好, Box2D23610.2.1 创建Box2D对象23710.2.2 Box2D调试绘制24010.2.3 整合所有元素24110.2.4 创建地面24310.3 基础Box2D交互和装饰24510.3.1 拖动对象24710.3.2 质量、密度、摩擦力和弹性25110.3.3 用精灵装饰Box2D形体25410.4 制作一个Box2D解谜游戏26110.5 小结27110.6 习题271第11章 中级游戏物理学:建模、竞速和跨越27211.1 开始27211.1.1 添加资源文件27211.1.2 创建基本的Box2D场景27311.2 用Box2D创建一个小车28311.2.1 用Box2D创建一个定制图形28411.2.2 使用Vertex Helper工具28511.2.3 用Box2D旋转关节添加轮子28811.3 让小车移动和跳跃29111.3.1 利用加速计使小车移动29211.3.2 让它可滚动29411.3.3 力和冲量30211.3.4 固定倾卸台30211.3.5 让小车跳跃30311.3.6 更灵敏的转向30611.4 小结30711.5 习题307第12章 高级游戏物理学:更真实的物

## &lt;&lt;Cocos2D应用开发实践指南&gt;&gt;

体30812.1 关节和人偶：带Ole回到场景30812.1.1 限制旋转关节30912.1.2 使用柱状关节31012.1.3 如何在恰当地点创建多个形体和关节31112.1.4 添加Ole：实现31312.2 添加障碍物和桥梁31812.2.1 添加桥梁31812.2.2 添加尖刺32112.2.3 添加改良的主循环32512.3 关底战斗32612.3.1 危险的挖掘机33412.3.2 最后一击：添加通关后电影序幕34012.4 小结34412.5 习题344第13章 Chipmunk物理引擎（不需要Alvin）34513.1 Chipmunk引擎是什么34513.2 Chipmunk引擎入门34713.2.1 添加Chipmunk引擎到你的项目中35113.2.2 创建基本的Chipmunk引擎场景35413.3 添加精灵并使它们移动36113.3.1 通过直接设置速度跳跃36613.3.2 通过设置表面速度移动地面36713.3.3 地面碰撞检测36713.3.4 Chipmunk引擎裁判器和法线36813.3.5 碰撞检测实现36813.3.6 移动和跳跃实现37213.4 Chipmunk引擎和约束37613.4.1 旋转平台37813.4.2 枢轴、弹簧和普通平台38013.5 伟大的逃脱38513.5.1 跟随Ole38513.5.2 布局平台38713.5.3 Ole动画序列38713.5.4 音乐和音效39113.5.5 添加背景39213.5.6 添加赢/输条件限制39313.6 小结39413.7 习题394第五部分 粒子系统、游戏中心和性能第14章 粒子系统：创建火焰、雪花、冰和更多39514.1 内置粒子系统39614.1.1 运行内置粒子系统39614.1.2 在沙漠中降雪39714.2 粒子设计器入门39814.3 为Space Viking创建并添加粒子系统40114.4 小结40614.5 习题406第15章 游戏中心的成就系统和高分榜40715.1 什么是游戏中心40715.2 为你的应用启用游戏中心40915.2.1 获取iOS开发者账号40915.2.2 为应用创建一个应用程序ID40915.2.3 在iTunes里注册你的应用41215.2.4 启用游戏中心支持41515.3 游戏中心身份验证41615.3.1 确定游戏中心可用41615.3.2 尝试给玩家进行验证41615.3.3 如果验证状态发生变化则告知41715.3.4 实现41815.4 配置成就系统42315.4.1 添加成就系统到iTunesConnect42315.4.2 成就系统如何工作42515.5 实现成就系统42615.5.1 创建游戏状态类42615.5.2 为加载和保存数据创建帮助函数42915.5.3 修改GCHelper以发送成就数据43115.5.4 在Space Viking中使用GameState和GCHelper43615.5.5 在应用中显示成就44015.6 设置并实现高分榜44215.6.1 在iTunes Connect里设置高分榜44215.6.2 高分榜的工作原理44315.6.3 实现高分榜44415.6.4 在游戏中显示高分榜44515.7 小结44815.8 习题448第16章 性能优化44916.1 CCSprite与CCSpriteBatchNode比较44916.1.1 测试性能差别45316.1.2 对纹理和纹理图集的建议45316.2 复用CCSprite45516.3 Cocos2D分析器45716.4 用Instruments发现性能瓶颈45916.4.1 时间分析器46016.4.2 OpenGL Driver Instrument46116.5 小结46416.6 习题464第17章 结束语46517.1 展望未来46617.2 Android及其他平台46617.3 最后的思考467附录 Cocos2D的主要类468



## <<Cocos2D应用开发实践指南>>

### 编辑推荐

《Cocos2D应用开发实践指南:利用Cocos2D、Box2D和Chipmunk开发iOS游戏》编辑推荐：深iOS开发工程师撰写，Box2D创始人Erin Catto等多位专家联袂推荐！书中系统讲解Cocos2D中的各种功能组件及技术实现细节，详细介绍Box2D、Chipmunk两大物理引擎的使用方法和粒子系统的原理，以及iOS游戏的性能测试和优化，以一个真实有趣的游戏案例Space Viking为主线，全面展示了Cocos2D游戏开发的流程、方法和技巧，实战性强。

<<Cocos2D应用开发实践指南>>

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>