



苹果开发
与应用系列

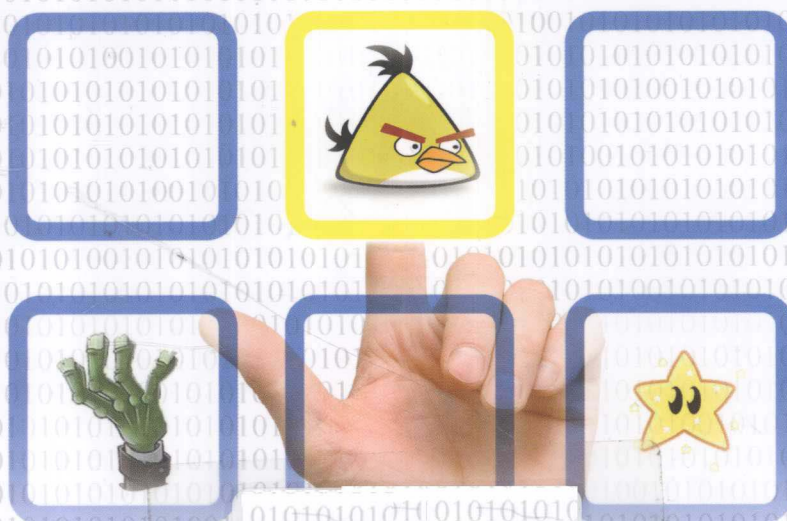


PEARSON

(美) Michael Daley 著

张铮 胥大坤 贾旭 译

iPhone 游戏 开发实践指南



LEARNING IOS GAME PROGRAMMING
A HANDS-ON GUIDE TO BUILDING
YOUR FIRST IPHONE GAME



机械工业出版社
China Machine Press





苹果开发
与应用系列

(美) Michael Daley 著

张铮 胥大坤 贾旭 译

iPhone游戏 开发实践指南



LEARNING IOS GAME PROGRAMMING
A HANDS-ON GUIDE TO BUILDING
YOUR FIRST IPHONE GAME



机械工业出版社
China Machine Press

本书通过一个实例介绍了在 iPhone 上创建游戏的完整过程, 全面涵盖从最初的游戏设计到最终将游戏上传至 App Store 的整个过程。全书共分 16 章, 每一章都详细介绍了该游戏的一个特定组件及支持该组件的技术, 包括最初的游戏设计思路、之前的准备工作、游戏循环、图像渲染、精灵表、动画、点阵字、贴图地图、粒子发射器、声音、用户输入、游戏界面、游戏对象和实体、碰撞检测, 以及如何对游戏进行测试等。

本书适合 iPhone 游戏开发人员阅读。

Simplified Chinese edition copyright © 2011 by Pearson Education Asia Limited and China Machine Press.

Original English language title: Learning iOS Game Programming : A Hands-On Guide to Building Your First iPhone Game (ISBN 978-0-321-69942-8) by Michael Daley, Copyright ©2011 .

All rights reserved.

Published by arrangement with the original publisher, Pearson Education, Inc., publishing as Addison-Wesley.

本书封面贴有 Pearson Education (培生教育出版集团) 激光防伪标签, 无标签者不得销售。

封底无防伪标均为盗版

版权所有, 侵权必究

本书法律顾问 北京市展达律师事务所

本书版权登记号: 图字: 01-2010-6649

图书在版编目 (CIP) 数据

iPhone 游戏开发实践指南 / (美) 戴利 (Daley, M) 著; 张铮等译. —北京: 机械工业出版社, 2011.8

(苹果开发与应用系列)

书名原文: Learning iOS Game Programming : A Hands-On Guide to Building Your First iPhone Game

ISBN 978-7-111-35176-4

I . i... II . ①戴... ②张... III . 移动电话机—游戏—应用程序—程序设计 IV . TN929.53—39

中国版本图书馆 CIP 数据核字 (2011) 第 125537 号

机械工业出版社 (北京市西城区百万庄大街 22 号 邮政编码 100037)

责任编辑: 陈佳媛

北京市荣盛彩色印刷有限公司印刷

2011 年 8 月第 1 版第 1 次印刷

186mm×240mm·20.75 印张

标准书号: ISBN 978-7-111-35176-4

定价: 59.00 元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

客服热线: (010) 88378991; 88361066

购书热线: (010) 68326294; 88379649; 68995259

投稿热线: (010) 88379604

读者信箱: hzjsj@hzbook.com

对本书的赞誉

“一本带领读者走入游戏编程的优秀书籍，它详细解释了针对 iPad、iPhone 以及 iPod touch 的游戏设计和实现的每个方面。对于那些对编写游戏感兴趣并准备付诸实现的读者，本书是一个不错的选择。”

——Tom Bradley, TBXML 设计者，软件架构师

“一位优秀的开发人员和一款超赞的游戏——读者能在这本书中找到这些内容，以学习为 iPhone 创建酷炫游戏。没准你就是下一个 AppStore 畅销者！”

——Sebastien Cardoso

“有了这本书，读者可以立即开始编写自己的游戏。书中的代码附有详细的解释，将为读者节省许多在文档中和网上论坛中查找资料的时间。”

——Pablo Gomez Basanta, Shifting Mind 的创始人

“作者在学习和教育方面的激情、解决问题的能力以及将一个完整的游戏作为例子程序使得这本书成为我所阅读过的好书之一。”

——Eugene Snyetilov

“如果你对于使用 OpenGL 和 OpenAL 在 iOS 平台上进行 2D 游戏编程很感兴趣，那么本书将在避免技术细节障碍的同时，带领你经历创建一个完整并有趣的游戏的过程。”

——Scott D. Yelich

“作者使朦胧的 iPhone 应用开发变得清晰。具体的例子、透彻的解释以及省时的小技巧使得这本书成为了每个即将开发 iPhone 游戏的开发人员的必备参考书。”

——Brandon Middleton, Tic Tac Toe Ten 的创建者

“这是一本 iOS 游戏开发的完全手册；这本书带领读者走过从基础和术语到实践使用技术，以开发完整游戏的全过程。在你认识到这一点以前，你将发现在开发自己的游戏时，从这本书中学到的扎实的原理和技术会提供很大的动力。在开发自己的游戏时，没有比这本书更好的参考资料了。”

——Rod Strougo, Prop Group 的创始人

译者序

苹果公司的 iPod、iPhone、iPad 以及 Mac 现在已经成为市场焦点，越来越多的人购买了这些产品。2008 年苹果公司发布了针对 iOS 的 SDK。而 Appstore 已经有超过 25 万个应用程序，每天都有更多的程序开发人员加入到这个流行趋势中来，为 iPhone 等产品开发更多的应用程序。与此同时，针对 iOS 应用开发的书籍也如雨后春笋般冒了出来。

目前在市场上可以看到很多关于 iOS 应用开发的书籍，包括商务、教育和娱乐等各方面的开发知识的讲解，其中大部分图书都来自于国外优秀的程序开发人员，但是在众多书籍中，由资深程序开发者 Michael Daley 撰写的本书显得尤为突出。

正像作者本人提过的一样，他在编写本书之前曾查阅过大量书籍，试图找到一本适合学习 iOS 游戏编程的书籍，但他发现虽然那些书都提供了很多详细的信息和丰富的小例子，却并不能帮助读者开发一个完整的足以发布的游戏。正因如此，Michael 开始了自己的写书历程。

本书以作者已经在 App Store 上发布的游戏 Sir Lamork's Quest: The Spell of Release 为主线，涵盖了从游戏的设计、图像的渲染、贴图地图、粒子系统、音乐、用户输入、游戏界面和碰撞检测等各个方面的知识，带领读者从零开始，一步一步学习 iOS 游戏编程。从小组件的实现到将整个游戏整合起来，读者不仅能从细部上学到编写 iOS 游戏所需的全部内容，还能从宏观上掌握整个游戏开发的全过程，是菜鸟级开发人员和资深开发人员的必备资料。

正因如此，我们组织翻译了本书。目的就是让更多的开发人员学习并投入到 iOS 游戏编程这个大领域中来，并获得乐趣。如作者在书中所说，即使我们不是游戏编程的大牛，也可以坐在家编写游戏。

在本书的翻译过程中，王杉、胥大坤、贾旭和我一起承担了主要的翻译工作，感谢郝杰和张皓琳等参与本书的翻译，感谢王命达和李广鹏对本书部分章节的校验，感谢杜强对本书的翻译工作提出的宝贵的建设性意见。由于时间仓促，本书在翻译过程可能存在问题，欢迎读者批评指正，可通过 aaron_boy_2008@sohu.com 与译者联系。

张铮

2011 年 6 月

前言

编写游戏不是一项轻松的任务，即使你是个经验丰富的程序员，游戏的设计模式、术语和思考过程看上去也会有点奇怪和不合常规。由于我的大部分工作时间都投入到了创建商业应用程序上，因此编写游戏只能算是一项业余爱好，这些年来我的孩子们享受着玩我编写的游戏的乐趣。随着 iPod touch 和 iPhone 的发布，我觉得是时候将我的一款游戏公布于世了。

我的首要任务是找一本开发 iPhone 游戏方面的优秀书籍。经过大量搜寻，我认为我想找的书并不存在。由于从我写的大量在线教程上获得了很多反馈，我决定自己写一本书。对于我来说，这是一个绝佳的机会，去编写一本我一直想要的游戏编程方面的书。

这些年来，我阅读了许多游戏开发方面的书籍，但还是觉得不够。尽管它们提供了许多创建游戏需要的各个部分的知识，也包含了许多小例子，但是从未包括编写出一个足够发布的完整的程序的全过程。我一直认为，一本好书应该告诉读者创建游戏需要什么，同时也应该展示在一个完整的游戏项目中应该如何实现每个组件。

所以，本书不只是介绍在 iPhone 上编写游戏需要的组件和技术，也会介绍创建一个完整的游戏 Sir Lamorak's Quest: The Spell of Release 的全过程。从 App Store 上可以免费下载此游戏，你可以通过它来学习如何编写游戏。

下载游戏

可以通过如下网址从 App Store 中下载游戏 Sir Lamorak's Quest : <http://itunes.apple.com/us/app/sir-lamoraks-quest-the-spell/id368507448?mt=8>。此游戏是免费的，下载后就可以开始玩了，试着帮助 Sir Lamorak 逃出城堡！

本书介绍了创建二维游戏的必要组件，涵盖了诸如 OpenGL ES 和 OpenAL 的技术，以及必要的键游戏引擎组件，如精灵表、动画、触摸输入和声音等。

本书的每一章都详细介绍该游戏中的一个特定组件，以及支持该组件的技术，例如贴图地图编辑器，或者尝试用 OpenGL ES 创建的一些特效。介绍完功能和技术后，紧接着会介绍在游戏 Sir Lamorak's Quest 中实现这些组件的细节。这种理论和实现相结合的方式，有助于填补其他开发书籍留下的空白。

关于游戏 Sir Lamorak's Quest

1982 年我得到了 48KB 的 Sinclair Spectrum 游戏电脑作为圣诞礼物，由此开始了玩游戏的经历。从那一刻开始我沉迷于计算机，并从此与之结缘。

一想到要为此书开发的游戏，我就想到了 20 世纪 80 年代玩的那些游戏。这些游戏当初给我

留下深刻的印象，尽管它们在视觉上并不是很炫目，但是玩起来还是很有意思的。

我在游戏设计上花了很多时间，不仅包括游戏中人物的设计，还包括如何在 iPhone 上实现游戏设计。游戏的一个关键方面在于它应该玩起来自然随意——即其观念应该简单并易于学习，而且玩家可以在不丢失进度的前提下随意开始和停止游戏。

同时我希望游戏的控件能很轻易地被玩家识别出来，因此我决定实现一个屏幕上的手柄来控制主要角色。然而让玩家能移动手柄位置也很重要，那样的话惯用左手和右手的玩家都能觉得舒适。

对于游戏情景本身，我决定采取我在 20 世纪 80 年代玩的那些游戏中的设计观念，即玩家被困在一个废弃的城堡中，他需要找到一个魔咒帮助他逃离城堡。

本书的组织结构

本中总共分 16 章，每章具体讲解创建游戏 Sir Lamorak's Quest 的一个特定领域，各章简介如下：

第 1 章“游戏设计”——本章描述设计游戏 Sir Lamorak's Quest 时的设计因素。本章会深入讲解开始创建游戏时的想法。虽然本章并不涵盖每个类别的游戏需要的所有可能的游戏设计决定，但确实会讲解最重要的设计决定。

第 2 章“术语、技术和工具”——即使资深程序员也会被这 3 个单词搞混。本章介绍创建游戏 Sir Lamorak's Quest 和其他游戏时使用的基本技术、术语和工具。本章将帮助读者了解贯穿本书的术语和技术。

第 3 章“开始探索旅程”——从本章开始我们将正式关注代码并往 iPhone 屏幕上渲染点东西。本章涉及用 Xcode 中的 OpenGL ES 样例项目创建第一个项目的过程，详细讲解该样例并为下一章设置好场景。

第 4 章“游戏循环”——任何游戏的核心都是游戏循环。循环负责确保所有的游戏元素（诸如 AI 和渲染）在正确的时间按正确的步骤执行。这听起来很简单，但是由于有很多种涉及游戏循环的方法，所以也不是那么简单。本章将介绍这些方法，并详细介绍游戏 Sir Lamorak's Quest 使用的方法。

第 5 章“图像渲染”——将图像绘制到屏幕上是任何游戏的基本要求。本章简单介绍 OpenGL ES，运行一些已经创建好的类来简化图像的创建和渲染过程。

第 6 章“精灵表”——精灵表是包含很多小图像的大图像。精灵表用于减少内存中存储的独立图像的数目，并减少 OpenGL ES 需要绑定以改善性能的纹理数目。创建动画精灵时也会用到精灵表。本章介绍创建精灵表（其中包括游戏中要使用的图像）的过程，无论其维度是固定或可变的。

第 7 章“动画”——用精灵表存储了动画中需要的不同帧后，本章描述如何将单独的图像组合成动画，如玩家角色跑动的动画。

第 8 章“点阵字”——与游戏玩家进行交互比较常见的方法是使用文本信息。能将操作指南和信息（例如玩家的分数或如何使用游戏的指导）渲染到屏幕上十分重要。本章描述如何用开源工具将任何字型转换成点阵字。创建好点阵字后，本章会介绍如何创建精灵表，用来存储渲染字型的字符时用到的图像。同时本章详细介绍游戏 Sir Lamorak's Quest 中用到的 Bitmap 字型类，它提供了一个简单的 API 用于渲染文本信息到屏幕上。

第9章“贴图地图”——使用贴图地图可以用一系列小图像创建出大型游戏世界。开发人员经常在内存有限时使用这种方法来创建大型游戏世界（想想任天堂的超级马里奥兄弟吧），在早期的家庭游戏系统中尤为如此。直到现在，贴图地图技术还是很受欢迎的，本章将描述使用开源贴图编辑工具来创建贴图地图，同时也会介绍将这些地图渲染到屏幕上的类。

第10章“粒子发射器”——许多游戏都有令人难以忘怀的特效，如火、爆炸、烟雾和火花等。这些特效通常是用粒子系统创建出来的。粒子系统负责创建和控制很多粒子；每个粒子都有自己的诸如大小、形状、方向、颜色和生命周期等属性。在其生命周期中，粒子的位置、速度、颜色和大小都会基于其设置发生变化。本章将详细介绍如何创建粒子系统，并用此来产生许多基本特效。

第11章“声音”——即使在现在的现代游戏中，将声音反馈给玩家依然是十分重要的。本章将介绍 iPhone 的媒体播放器功能和 OpenAL，以及介绍如何用这些技术在游戏中播放酷酷的音迹和 3D 音效。

第12章“用户输入”——本章将介绍如何用 iPhone 的独特触控技术和加速器功能来控制游戏。本章详细描述如何同时获取和处理多点触控，以及如何在游戏中获取和使用加速器的数据。

第13章“游戏界面”——本章着眼于游戏 Sir Lamorak's Quest 中游戏界面的实现，包括如何处理旋转事件以确保正确显示用户界面，以及如何混合使用 OpenGL ES 和 UIKit 界面控件等。

第14章“游戏对象和实体”——当玩家在游戏 Sir Lamorak's Quest 的城堡中走动时，我们希望他能找到和捡起对象并能和坏人进行战斗。本章描述如何实现游戏 Sir Lamorak's Quest 中的对象和实体。

第15章“碰撞检测”——如果玩家和坏人能穿过墙壁和门，游戏就看起来很糟糕，所以有必要检测玩家和对象间，或者玩家与游戏中的对象和实体间的碰撞。本章描述不同类型的碰撞检测，并介绍如何在游戏 Sir Lamorak's Quest 中实现它们。

第16章“组装起来”——到此时为止，本书已经介绍了很多基础内容。然而，读者还可以为自己的游戏锦上添花。本章介绍在手机来电导致玩家退出游戏时如何保存游戏状态。同时介绍使用仪表进行的单元检测以及如何对游戏进行 β 测试等内容。

本书的目标读者

本书是为那些有程序开发经验但没有开发过计算机游戏的开发人员准备的。尽管本书假设读者已经有利用 Objective-C 编程的经验，但每章都会提供 Objective-C 和其他技术的相关知识，以供读者了解其概念和实现。

读者阅读完本书后，能掌握如下知识：对于游戏 Sir Lamorak's Quest 的游戏引擎的深度认识，以及创建 2D 游戏引擎需要的关键能力和需要考虑的事项。本书将使你具备使用本书中的游戏引擎进行自己的游戏开发的能力，或者简单利用创建游戏的知识来使用其他可用的 iPhone 游戏引擎（例如 Cocos2D）的能力。

如果你已经开发过 iPhone 应用，但想将注意力从实用应用转到游戏上，那么本书就是为你准备的。本书建立在读者已有的开发知识之上，通过描述需要的术语、技术和工具以及提供实际的实现例子，带领读者进入游戏开发的领域。

不适合本书的读者

本书讲的是二维平面游戏，如果你的兴趣不在这里，比如说你感兴趣的是用 OpenGL ES 来应用三维图形，那么本书并不适合你。

在阅读本书之前，希望你已经对 Objective-C、C、Xcode 和 Interface Builder 有了一定的了解。尽管本书中描述的实现已经尽量简化，并且限制在 C 语言下进行开发，对于其他语言的坚实基础也是很有必要的。

如下书籍会提供一些基础：

- *Cocoa Programming for Mac OS X, Third Edition* by Aaron Hillegass (Addison-Wesley, 2008).
- *Learning Objective-C 2.0*, by Robert Clair (Addison-Wesley, 2011).
- *Programming in Objective-C 2.0*, by Stephen G. Kochan (Addison-Wesley, 2009).
- *Cocoa Design Patterns*, by Erik M. Buck and Donald A. Yacktman (Addison-Wesley, 2009).
- *The iPhone Developer's Cookbook, Second Edition*, by Erica Sadun (Addison-Wesley, 2010).
- *Core Animation: Simplified Animation Techniques for Mac and iPhone Development*, by Marcus Zarra and Matt Long (Addison-Wesley, 2010).
- *iPhone Programming: The Big Nerd Ranch Guide*, by Aaron Hillegass and Joe Conway (Big Nerd Ranch, Inc., 2010).

这些图书和其他一些网上的资源可以帮助你了解更多如何为 Mac 和 iPhone 编程的信息，并加深对 Objective-C 和 Cocoa 框架的理解。

下载源代码

除了本书之外你还可以了解很多信息。游戏 Sir Lamorak's Quest 的完整的带有注释的源代码可以从 InformIT.com 下载。[⊖]

本书通篇穿插了大量的代码，并为读者准备了很多练习，但是读者仍然很有必要阅读苹果为开发者准备的工具，诸如 Xcode 和 iPhone SDK 等。可以从 Apple iPhone Dev Center[⊖] 下载这些资源。

作者简介

白天，Micheal Daley 在为世界上最大的商业软件公司工作，该公司的主要顾客为大型企业。晚上，Micheal 学习如何为 iPhone 手机构建游戏。Micheal 最开始用 BASIC 在 Sinclair Spectrum 48KB 上编写历险游戏，后来进一步发展到了 Commodore 64 和 Amiga A500 上写游戏。他从未停止对游戏编程的热爱，自 iPhone 问世后，Micheal 开始学习 Objective-C 和如何为 iPhone 创建游戏。

许多年来 Micheal 为自己的孩子编写了很多游戏，iPhone 的发布让他想为更多的人编写游戏，而不仅仅是为自己的孩子。对于学习并应用新的技术，Micheal 很有热情。同时他也是个 Apple 粉丝，在最新的 Apple 设备上投入了大量时间和金钱。

⊖ Apple 的 iPhone DevCenter 网址：developer.apple.com/iphone。

⊖ 也可访问 www.hzbook.com 下载。

目 录

对本书的赞誉
译者序
前言

第1章 游戏设计	1
1.1 游戏的开始	2
1.2 主导思想	3
1.2.1 适合iPhone的游戏	3
1.2.2 故事情节	4
1.2.3 名字的内涵	4
1.2.4 游戏的目标	5
1.3 游戏的元素	5
1.3.1 时间	6
1.3.2 生命	6
1.3.3 健康值	6
1.3.4 物品	7
1.3.5 羊皮碎片	7
1.3.6 门	8
1.3.7 武器	8
1.3.8 实体	8
1.3.9 玩家	9
1.4 小结	9
第2章 术语、技术和工具	10
2.1 术语	10
2.1.1 精灵	11
2.1.2 精灵表	12
2.1.3 动画	14
2.1.4 点阵字	15
2.1.5 贴图地图	16
2.1.6 粒子系统	16
2.2 碰撞检测	18
2.2.1 人工智能	18

2.2.2 游戏循环	19
2.3 技术	21
2.3.1 Objective-C	21
2.3.2 Cocoa Touch	21
2.3.3 OpenGL ES	22
2.3.4 OpenAL	24
2.4 工具	25
2.5 小结	30
第3章 开始探索旅程	32
3.1 在Xcode中创建项目	32
3.2 程序的运行	34
3.3 准备就绪	34
3.4 程序代理	35
3.4.1 查看头文件	36
3.4.2 查看实现文件	37
3.5 EAGLView	40
3.5.1 EAGLView.h	40
3.5.2 EAGLView.m	41
3.6 ES1Renderer	48
3.6.1 审查ES1Renderer.h	48
3.6.2 查看ES1Renderer.m	49
3.6.3 创建帧缓冲区和渲染缓冲区	50
3.6.4 render方法	52
3.6.5 定义颜色值	54
3.6.6 定位	55
3.7 OpenGL的工作原理	56
3.7.1 对模型应用变换	56
3.7.2 在屏幕上渲染	57
3.8 小结	59
第4章 游戏循环	61
4.1 时间控制就是一切	61
4.2 冲突检测	62

4.3 游戏循环	62	6.2 使用Zwoptex	116
4.3.1 基于帧的循环体	63	6.3 SpriteSheet类	117
4.3.2 基于时间的固定间隔循环体	64	6.3.1 初始化	117
4.4 开始	65	6.3.2 提取精灵	120
4.4.1 EAGLView类的修改	65	6.4 PackedSpriteSheet类	121
4.4.2 EAGLView.m文件的修改	66	6.4.1 初始化	121
4.4.3 ES1Renderer类	68	6.4.2 解析控制文件	122
4.4.4 配置视图端口	71	6.4.3 提取精灵	123
4.5 游戏场景和游戏控制器	72	6.5 小结	123
4.5.1 创建游戏控制器	72	6.6 练习	124
4.5.2 GameController类	72	第7章 动画	125
4.5.3 创建单态类	74	7.1 动画篇的项目	125
4.5.4 GameController.m文件内部	74	7.2 动画简介	125
4.5.5 AbstractScene类	76	7.2.1 帧	126
4.5.6 GameScene类	77	7.2.2 状态	126
4.6 小结	79	7.2.3 类型	126
4.7 练习	79	7.2.4 方向	126
第5章 图像渲染	81	7.2.5 跳动帧	126
5.1 渲染入门	81	7.3 Animation类	127
5.2 四边形的渲染	82	7.3.1 初始化	127
5.3 纹理映射	84	7.3.2 添加帧	128
5.4 交错顶点数组	86	7.3.3 更新动画	128
5.5 结构体	88	7.3.4 渲染动画	130
5.6 图像渲染类	89	7.3.5 完成	131
5.6.1 Texture2D类	89	7.4 小结	133
5.6.2 TextureManager类	96	7.5 练习	133
5.6.3 ImageRenderManager类	98	第8章 点阵字	134
5.7 Image类	104	8.1 点阵字项目	134
5.7.1 初始化	105	8.2 点阵字简介	134
5.7.2 获取子图像	107	8.3 创建点阵字精灵表	135
5.7.3 复制图像	108	8.4 BitmapFont类	137
5.7.4 渲染图像	108	8.5 与C语言的关系	138
5.7.5 getter和setter	111	8.5.1 初始化方法	138
5.8 小结	112	8.5.2 解析控制文件	139
5.9 练习	112	8.6 渲染文本	142
第6章 精灵表	113	8.6.1 渲染合理的文本	143
6.1 精灵表简介	113	8.6.2 文本宽度和高度	145
6.1.1 简单精灵表	113	8.6.3 重新分配	146
6.1.2 复杂精灵表	114	8.7 小结	146

8.8 练习	147	10.4 粒子发射器类	185
第9章 贴图地图	148	10.4.1 TBXMLParticleAdditions类	185
9.1 贴图地图入门	148	10.4.2 ParticleEmitter类	187
9.2 贴图地图简介	148	10.5 开始游戏	198
9.3 贴图地图编辑器	150	10.6 小结	199
9.3.1 贴图调色板	151	第11章 声音	200
9.3.2 图层	151	11.1 声音项目	200
9.4 创建贴图地图	152	11.2 iPhone的声音简介	200
9.4.1 新建一个贴图集合	152	11.2.1 音频会话	200
9.4.2 创建地图图层	153	11.2.2 播放音乐	202
9.4.3 创建对象图层	154	11.2.3 播放音效	202
9.4.4 绘制地图	154	11.2.4 创建音效	204
9.4.5 放置对象	154	11.2.5 立体声与单声道	205
9.5 认识Tiled配置文件	155	11.3 声音管理器类	205
9.5.1 map元素	155	11.4 音效管理	219
9.5.2 tileset元素	155	11.4.1 载入音效	220
9.5.3 layer元素	156	11.4.2 播放音效	223
9.5.4 objectgroup元素	157	11.4.3 停止播放音效	225
9.6 地图类	157	11.4.4 设置音效和监听器位置	226
9.6.1 Layer类	158	11.5 处理声音播放干扰	227
9.6.2 TileSet类	162	11.6 小结	228
9.6.3 TiledMap类	163	第12章 用户输入	230
9.6.4 初始化	165	12.1 用户输入项目	230
9.6.5 解析地图文件	166	12.2 用户输入简介	231
9.6.6 创建图层图像	174	12.3 处理触控事件	232
9.6.7 图层渲染	175	12.3.1 touchesBegan阶段	234
9.6.8 获取贴图信息	177	12.3.2 touchesMoved阶段	235
9.7 小结	178	12.3.3 touchesEnded阶段	236
9.8 练习	178	12.4 处理敲击	237
第10章 粒子发射器	179	12.5 加速器事件	238
10.1 粒子发射项目	179	12.6 小结	240
10.1.1 粒子系统简介	180	第13章 游戏界面	241
10.1.2 粒子系统参数	181	13.1 游戏界面项目	241
10.2 粒子的生命周期	182	13.2 OpenGL ES界面	242
10.2.1 粒子的诞生	182	13.2.1 界面渲染	242
10.2.2 粒子生存	183	13.2.2 按钮边界定义	245
10.2.3 粒子消亡	183	13.2.3 点击处理	245
10.2.4 粒子重生	184	13.2.4 可视化边界	246
10.3 粒子发射器的配置	184	13.2.5 转换处理	248

13.2.6 OpenGL ES定向	249	15.6 实体与地图间的碰撞检测	293
13.3 UIKit界面	251	15.7 实体与实体间的碰撞检测	294
13.3.1 创建界面	252	15.8 小结	296
13.3.2 界面连接	254	第16章 组装起来	297
13.3.3 UIKit定向	257	16.1 摄像机	297
13.3.4 显示或隐藏UIKit界面	259	16.2 保存游戏状态和设置	299
13.4 小结	261	16.2.1 保存游戏状态	299
第14章 游戏对象和实体	262	16.2.2 载入游戏状态	300
14.1 游戏对象和实体项目	262	16.2.3 保存游戏设置	302
14.2 游戏对象	263	16.2.4 载入游戏设置	303
14.2.1 AbstractObject类	263	16.3 保存高分	304
14.2.2 EnergyObject类	265	16.3.1 添加分数	306
14.3 游戏实体	272	16.3.2 保存高分	307
14.3.1 AbstractEntity类	273	16.3.3 载入高分	307
14.3.2 人工智能	275	16.4 性能和测试	308
14.3.3 Player实体类	276	16.4.1 使用仪表	309
14.4 保存游戏对象或实体	284	16.4.2 Leaks仪表	310
14.5 小结	286	16.4.3 使用OpenGL ES仪表	312
第15章 碰撞检测	287	16.4.4 用Thumb编译	314
15.1 碰撞检测简介	287	16.5 β 测试	314
15.2 基于帧与基于时间	289	16.5.1 多种设备类型	315
15.3 轴对齐边界框	289	16.5.2 反馈	316
15.4 检测碰撞	290	16.6 小结	316
15.5 碰撞地图	291		

第1章 游戏设计

我热爱游戏，话虽简单，但它包含了我对游戏的无限深情。自从1982年得到了第一台电脑，我就一直在玩游戏。这台 Sinclair Spectrum 48KB（见图1-1）是我向父母软磨硬泡了好几个月后才得到的。1982年圣诞节的早上，躺在一堆撕开的包装纸间的，就是我的第一台电脑。它全身漆黑闪亮，有着灰色橡胶按键。对我来说，它是一个美好而神奇的事物。

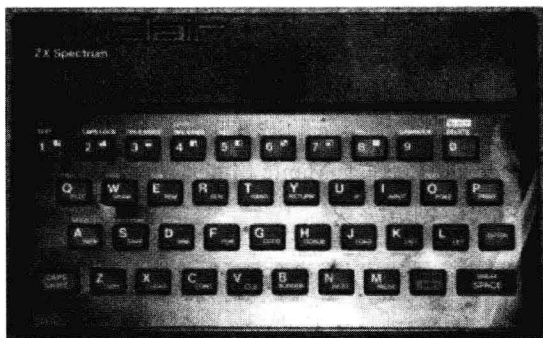


图1-1 我最早的电脑，1982年的 Sinclair ZX Spectrum

我不知道它如何工作——仅仅只是使用它而已。当我坐在电脑前，看着第一个游戏慢慢地载入并出现在便携式电视上，听着来自盒式磁带的古怪尖叫声。自此，我便与电脑结下了不解之缘。

得到 Spectrum 不久，我开始玩所有能找到的游戏。我想知道电脑以及游戏是怎样工作的。那时候，有很多杂志提供一些如何在 Sinclair 上用 BASIC 编程的信息。例如《Sinclair User》和《ZX Computing》里有程序的所有代码，这意味着在运行程序前，你必须非常仔细地一行一行地输入那些代码。我花了很多时间去尝试一些可以运行的东西，结果却很失望。

一些看起来像是徒劳无功的事情其实也是一种宝贵的学习和锻炼。通过输入所有代码，我了解了代码是如何工作的，以及一个游戏的必要组成部分。这也表明，任何一个人，并不仅限于一些大公司的职员，包括在卧室中进行学习并只有一个 Spectrum 的我在内，都可以编写游戏。

早些年，我经常反复思考如何在 iPhone 上创建游戏。我曾经尝试过许多想法，但都没有成功，感觉很失望。然而，有些想法的确成功了，这足以弥补我的失望。当你一直在做的东西确实按照你的设想生效时，那种获得成功的感觉很棒。直到现在，当看到编写的程序确实能工作时，我仍能感到十分兴奋。上班时编写软件，我也会有相同的感受，但说实话，这种感觉与我看到游戏中一些东西在屏幕上运行时的感觉是不同的。后者会令人觉得兴奋许多。

为 iPhone 编写程序时，我觉得又回到了 12 岁，只是不再需要看别人脸色行事也没有家庭作业了。很多人都意识到可以创作并分享自己的游戏。虽然游戏开发技术现在已改变了许多，分享游戏也比复制盒式磁带要简单许多，但提出想法并努力把自己的游戏变成现实这一过程都是大同小异的。

本章涵盖了设计的主要原则，以及我在设计并开发这本书所包含的游戏时所考虑到的游戏元素，包括：

- 高层次游戏设计。
- 故事情节。
- 游戏的元素。

建立一个稳固的框架是非常重要的，我们要在其之上开发游戏，而这一章提供了开始在 iPhone 上创建游戏时需要的信息。

1.1 游戏的开始

当时，我主要是受到了 Matthew Smith 的启发，他 17 岁创造了题为 Manic Miner[⊖]的游戏。Manic Miner（见图 1-2）是一款有 20 个关卡的游戏平台，它甚至有自己的游戏配乐，这在当时是首创。Matthew 在短短六个星期内创作了 Manic Miner，虽然他很有才华，但这仍旧证明了：任何人只要有想象力和技术，就可以创造游戏。

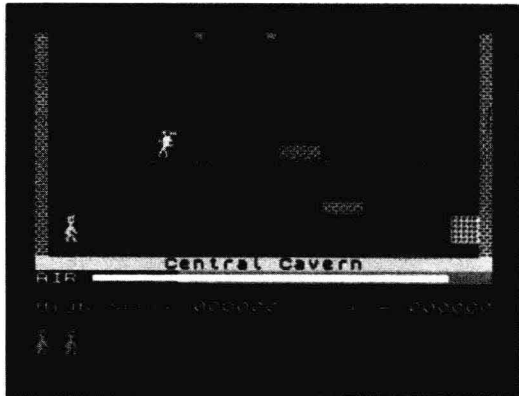


图1-2 ZX Spectrum上Manic Miner的第一关

从那以后，计算机不仅成为了我生命中巨大的一部分，而且也成了我的职业。虽然我的工作不是设计畅销游戏，但我也算是业内人士之一了。这令我很高兴，但我仍然希望能创建一个能让别人玩的游戏。这些年，我用各种语言（当然，Java 是最新的）创作了很多简单的游戏雏形，但直到现在为止，它们都没有成功。

早在 iPhone 上市的时候，我就成了 Apple 的忠实粉丝，并且，当我看见它的时候，我就喜欢上了它。苹果公司发布了 SDK 之后，我马上就下载了并拿来玩，就像其他成千上万的人一样，

[⊖] 了解该游戏的更多历史或是搜索游戏参见 en.wikipedia.org/wiki/Manic_Miner。

我关注着 App Store 上出现的越来越多的游戏。

这种感觉就像当年用 Spectrum 一样。就像游戏的发行商和经销商一样，人们坐在自己的卧室里，开发着 iPhone 游戏并使用着 Apple 和 App Store。不久之前，我在互联网上四处搜集能够帮助开发游戏的信息，而这最终导致我编写了此书。

1.2 主导思想

每个游戏都源于一种观念，一个想法。当开始考虑这本书的游戏时，我想到了能提供素材的办法：我的孩子们。我独自坐了 30 分钟，才开始草草地写笔记，因为他们描述了一个非常不可思议的世界，这个世界中事物该如何互动和玩家该拥有什么样的武器等，以及他们认为最重要的：玩家如何死去。如果你有两个小男孩，这些都将成为血淋淋的细节。直到那时我才意识到，他们有多么重视他们的游戏。我立即解释说无法开发 Halo 10，他们便很快地减少了建议。

我必须承认，这个游戏大部分的创意来源于我在 20 世纪 80 年代玩的传统的 Spectrum 游戏。我非常喜欢它们，以至于在第一个 iPhone 游戏中，我无法抗拒地使用了其中一些经典的游戏玩法。

1.2.1 适合 iPhone 的游戏

当致力于思考一个游戏时，必须要着重考虑的因素是人们到底是如何使用 iPhone 的。我喜欢玩 iPhone 上的游戏，但是，我不像孩子们玩他们的 Xbox 一样，花很多时间在游戏上。大部分人拿起 iPhone 打开一个游戏，玩几分钟（通常是在开会时），然后就又把 iPhone 收起来了。在 App Store 上，这类主题的游戏也在不断增加。

这是一个重要的考量，它会在不同的方面影响游戏的设计，包括：

- 玩家能够轻易地退出游戏，并且能够在他们离开的地方（或者从他们退出时那个关卡的开始）继续游戏。
- 游戏的控制要直观、易于掌握和快捷。如果用户只想玩击打类或是冲刺类的游戏，是否拥有一个易于控制的机制就成了关键。还需要为 iPhone 进行专门的设计。利用加速器和多点触摸是需要慎重考虑的机制。
- 故事情节不应该太过复杂，以免长时间后玩家忘记了情节再玩时不知道怎么玩。
- 游戏需要有一个清晰而易于理解的目标。它可以是一个目标或者简单的基于分数或时间的评定。

这些是我在设计游戏时考虑的一些问题。我决定让游戏是随意的，因此玩家可以从上次来结束的游戏处继续玩。操作方面，玩游戏时可以使用屏幕上的手柄，即使玩家觉得熟悉，又易于操作。

注意

此处忽略了很多游戏控制的选项问题，本书将在第 12 章“用户输入”讨论这些选项和决策选择的过程。

这款游戏是休闲自然类的（这正是我不能开发 Halo 10 的原因），最后成了以闹鬼的古堡为基础的二维游戏。这座城堡住着幽灵和食尸鬼，几乎没人敢来这冒险。想到这里，我写了一个故事情节，帮助建立游戏视觉上的意象，并可以在编写游戏时提供一些可参考的东西。

提示

过去我发现面对一堆游戏代码时很容易偏离轨道，所以一些可供参考的东西将有助于步入正轨。

1.2.2 故事情节

玩家扮演一个年轻的骑士 Sir Lamorak，为了获得名利并引起村子里姑娘们的注意而四处探险。为了证明在 Ulpha 森林里、长期废弃的 Egremont 古堡里没有幽灵一类的东西，他开始了探险。

当 Sir Lamorak 到了长满青苔的旧城堡时，他开始感到不安。仿佛有无数的眼睛注视着他渐渐接近房子的入口。当他从信赖的马 Buttercup 上跳下来的时候，Buttercup 转身、嘶鸣并狂奔回城了，只剩下勇敢的 Sir Lamorak 独自一人。由于多年无人看管，生锈的合页在开门时咯吱作响，仅仅是风的作用，Sir Lamorak 被吹（或者是吸，说不定）进一个大型的木质城堡门内。

Sir Lamorak 盯着他面前的黑暗，慢慢向城堡的正门挪动。他什么都看不见，但是能听到风穿过城堡的呻吟声以及远处滴水的声音。突然，一阵怒号着的大风，横扫走廊，他身后的大门被用力关上，Sir Lamorak 害怕地向前跳了一下。他转身跑到门口，抓住手柄使劲拉，但门打不开了。他被困住了。

Sir Lamorak 站在黑暗中，回想着他当初决定拜访这个地方的原因时，他听到了一个邪恶的声音透过黑暗传来：

“只有能念出 Spell of Release 的人才能离开这里，否则，你将和我们一样永远被困在这里，哈哈哈哈哈！”

Sir Lamorak 恐惧得直发抖，他站在黑暗的古堡中，完全不知道“Spell of Release”是什么，该去哪里找它，以及这个邪恶的声音是谁的。然而，有一件事他可以肯定，那就是……他并不是一个人。

1.2.3 名字的内涵

我喜欢令人毛骨悚然的故事，这个故事情节为其余的设计设定了一个很好的基调。基于这一点，我想给游戏起个名字。以后就可以用这个名字来代替游戏本身，并且它有助于建立游戏的标识。

你或许并不觉得给自己的游戏命名可能会导致麻烦，但是当它要上传到 App Store 上时，确实需要注意一些重要的问题。在撰写本书时，App Store 上已经有超过 150 000 个应用。需要指出的是，有大量的应用程序和名字已经被占用了。在你花费时间也许还有金钱为游戏起名字的时候，要确保没有其他人在 App Store 上使用过这个名字。没有比在完成一切后才发现已经有别人