

HTML5 2D 游戏编程核心技术

[美] 戴维·吉尔里 著 郝刚 黄丕超 赵刚 译 黄敏 审校
(David Geary)

Core HTML5 2D Programming

- 初学者的入门指南，从零开始完整讲解了一个复杂的横向卷轴平台视频游戏的开发过程。
- 本书涵盖丰富的代码和演示实例，通过高可靠性、可复用的代码帮助专业开发者掌握必备知识。



机械工业出版社
China Machine Press

HTML5 2D 游戏编程核心技术

[美] 戴维·吉尔里 著 郝刚 黄丕超 赵刚 译 黄敏 审校
(David Geary)

Core HTML5 2D Game Programming



机械工业出版社
China Machine Press

图书在版编目 (CIP) 数据

HTML5 2D 游戏编程核心技术 / (美) 戴维·吉尔里 (David Geary) 著; 郝刚, 黄丕超, 赵刚译. —北京: 机械工业出版社, 2016.10

(游戏开发与设计技术丛书)

书名原文: Core HTML5 2D Game Programming

ISBN 978-7-111-55129-4

I. H… II. ①戴… ②郝… ③黄… ④赵… III. 超文本标记语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2016) 第 252208 号

本书版权登记号: 图字: 01-2014-7248

Authorized translation from the English language edition, entitled Core HTML5 2D Game Programming, 978-0-13-356424-2 by David Geary, published by Pearson Education, Inc., Copyright © 2015.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc.

Chinese simplified language edition published by Pearson Education Asia Ltd., and China Machine Press Copyright © 2016.

本书中文简体字版由 Pearson Education (培生教育出版集团) 授权机械工业出版社在中华人民共和国境内 (不包括香港、澳门特别行政区及台湾地区) 独家出版发行。未经出版者书面许可, 不得以任何方式抄袭、复制或节录本书中的任何部分。

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

HTML5 2D 游戏编程核心技术

出版发行: 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

责任校对: 殷虹

印刷: 北京文昌阁彩色印刷有限责任公司

版次: 2016 年 11 月第 1 版第 1 次印刷

开本: 186mm×240mm 1/16

印张: 30

书号: ISBN 978-7-111-55129-4

定价: 99.00 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) 88379426 88361066

投稿热线: (010) 88379604

购书热线: (010) 68326294 88379649 68995259

读者信箱: hzit@hzbook.com

版权所有·侵权必究

封底无防伪标均为盗版

本书法律顾问: 北京大成律师事务所 韩光 / 邹晓东

The Translator's Words 译者序

时下，HTML5 已经不再是 Web 技术爱好者口中的下一代 Web 语言，在历经 8 年的艰辛努力后，万维网联盟（W3C）终于在 2014 年发布了正式版的 HTML5 标准规范。HTML5 由此结束了摇摆不定的状态，正式成为行业标准，成为互联网巨头们争相热捧的 Web 技术和框架。乔布斯的预言正在成为现实。

早在 HTML5 标准规范制定初期，HTML5 游戏就已经出现，不过那时候的游戏更多的是以 HTML5 教学用例或个人学习成果的形式出现的。随着浏览器厂商不断完善对 HTML5 的支持，HTML5 游戏也得到了进一步的发展，并在 2012 年形成一波 HTML5 游戏浪潮，此后便转入沉寂。2014 年 7 月，《围住神经猫》这款 HTML5 小游戏在微信朋友圈火热转发，让业界看到了 HTML5 移动游戏的优势和潜力；此后的 HTML5 版《愚公移山》则让移动游戏开发者看到了让 HTML5 游戏实现商业化的希望。

得益于移动智能设备硬件性能的不断提升及移动端浏览器的强大支持，HTML5 游戏正在摆脱小游戏、轻游戏的困局，具备了开发中度、重度游戏的基础。越来越多的大型游戏开发商正在加入 HTML5 游戏开发的队伍，全新 Unity5、虚幻引擎、3D 引擎 Sting Ray 等游戏引擎开始支持 HTML5 游戏开发，HTML5 游戏的前景得到了行业的广泛认可。

作为一个标准，HTML5 不仅仅是一种标记语言，也不仅仅局限于游戏开发，而是一种 Web 应用开发的框架和平台，包括了无需插件的视频、音频、本地存储、Canvas 元素及绚丽的渲染效果等。而其中，最让前端开发者激动的当属 Canvas 元素，它赋予了前端开发者充分发挥自身丰富想象力的可能，结合 JavaScript 语言、CSS（层叠样式表），前端开发者能渲染出各种美轮美奂的视觉效果，带给用户最佳的使用体验。

译者也正是因为在实际工作中需要在浏览器端实现精美的数据可视化效果而开始学习、使用 HTML5 技术的。像大多数的软件开发人员一样，译者也常感到在学习一项新技术时，抛开技术本身的难度不说，如何在实际工作中应用这项新技术才是决定把知识转化为能力的

关键。而对于 HTML5 来说,从严格意义上讲,它只是一个停留在 W3C 众多草案中的标准,实际应用效果如何除了看浏览器的支持程度外,更重要的还是如何在开发中利用 JavaScript 语言和 CSS 来充分发挥 HTML5 的各项优势。因此,掌握 HTML5 技术的困难之处不在于对 HTML5 标准规范的理解,或是对 HTML5 单个特性的掌握,而在于如何把众多分散在 HTML5 技术、JavaScript 语言、CSS 中的知识点有机地结合起来,在实际开发中应用这些核心技术、满足开发需求,实现最终目的。

好在作者及时写就了本书。

本书是作者在《HTML5 Canvas 核心技术:图形、动画与游戏开发》(已由机械工业出版社出版,书号为 978-7-111-41634-0)之后的又一力作。在前作中,作者全面讲解了 Canvas 元素的 API 接口,以及如何利用 Canvas 元素进行图形绘制、动画制作、物理效果模拟、碰撞检测、游戏开发等内容,并使用大量的实例,演示了具体的实现方法。本书虽然也涉及相似内容的讲解,但绝不是前作的重复,而是对这些技术的综合、系统应用。

全书以完善作者在写作《HTML5 Canvas 核心技术:图形、动画与游戏开发》过程中编写的一款名为“Snail Bait”的二维横向卷轴游戏作为主线,系统地讲解了如何从游戏构思、收集原始素材、搭建游戏框架、逐步添加游戏功能到最终实现一个功能完善的游戏开发过程,全面展示了二维游戏开发的方方面面。

在开发过程中,作者没有使用任何当下流行的 HTML5 图形框架或游戏框架,而是从零开始,一步步编程实现各种二维视频游戏的常见功能。因此,读者可以从本书中学习如何通过编程实现平滑动画、sprite 行为、粒子系统、碰撞检测、开发者后门、游戏难易度设计等所有内容。这能够让读者排除游戏框架带来的干扰,深入、细致地了解游戏实现的内部逻辑、机制及技术细节,对读者理解游戏框架、引擎的工作原理,更好地使用游戏框架、引擎进行游戏开发也是大有裨益的。

而作为作者用于系统讲解相关技术的载体,Snail Bait 游戏虽然是只有一个小关卡的小游戏,但却具备了二维视频游戏的各种要素:同时操作多个具有不同行为的 sprite 对象;在不同的 sprite 对象之间发生碰撞时,产生各不相同的效果;在播放背景音乐的同时,根据游戏状态同时播放多种音效;游戏还具备了方便开发者调节游戏难易度、拖曳游戏主角的开发者后门;一个可以减慢或加快游戏整体速度的时间系统。对于学习 HTML5 游戏开发来说,Snail Bait 游戏是一个经典的案例:其技术覆盖面广泛,可以让读完整了解一个游戏开发的各个方面;游戏自身的可玩性强、难易度适中,既不会让读者在学习过程中感到枯燥无味,又不会让读者感觉到晦涩难懂。

作者没有刻意对全书内容的先后顺序进行编排,一切都是顺着游戏从构思到最终完善

的渐进过程自然而然形成的。比如在实现游戏主角的跳跃行为时，作者使用了两章内容来讲解，先实现一个基本的跳跃行为，然后逐步加入重力、时间等因素，最终实现一个真实的跳跃行为。在代码的组织上，作者也颇费了一番工夫，按照每章内容讲解完成时编写的代码来组织版本，从而使读者在学习时能够快速定位到本章内容的实现上，不至于淹没在代码中。没有使用“需求分析、系统设计、编程实现”这种固定的软件开发流程，作者使用的开发方法契合了个人和小团队的实际开发模式需要，更有助于读者在一边学习技术的同时，一边动手开发属于自己的 HTML5 游戏。

此外，作者在讲述主要内容时，也简明扼要地介绍了使用 node.js 部署游戏、移动设备中的 HTML5 游戏开发技术等内容。在代码编写方面，作者遵从低耦合的原则，使大部分代码具备很强的通用性，能够直接应用到读者自己的 HTML5 游戏中。

在本书的翻译过程中，我们得到了机械工业出版社各位编辑及工作人员的热情帮助，在此表示由衷的感谢。

译者

2016 年 9 月

前言 Preface

本书是为希望使用 HTML5 进行 2D 游戏开发，且具备一定 JavaScript 编程经验的开发人员而准备的。在本书中，笔者从零开始完整讲解了一个复杂的横向卷轴平台视频游戏的开发过程，并将该视频游戏命名为“Snail Bait”。在开发过程中，笔者没有使用任何第三方图形框架或游戏框架，因此读者将从本书中学会以编程方式实现平滑动画、sprite 爆炸、开发人员后门程序、游戏难易度设计等所有内容。如果读者正在使用某个游戏框架开发游戏，那么本书对于帮助读者理解游戏框架内部的工作原理也同样大有裨益。

出于教学目的，Snail Bait 游戏仅实现了一个关卡，然而它却具备了街机风格游戏的全部特点。Snail Bait 游戏在滚动的背景上同时操作多个动画对象，即 sprite 对象，在播放游戏背景音乐的同时添加多种音频效果。sprite 对象具备多种行为，包括但不限于跑动、跳跃、飞行、闪光、弹跳、来回移动、爆炸、碰撞、平台着陆、从游戏画面底部跌落等。

另外，Snail Bait 游戏还实现了多个特色功能。例如，一个可以减慢或加快游戏整体速度的时间系统；游戏加载过程动画；当游戏主角失去生命时，震动游戏画面的特殊效果；模拟烟雾和燃烧的粒子系统等。当游戏窗口失去焦点时，Snail Bait 游戏会暂停运行，而当游戏窗口重新获得焦点时，Snail Bait 游戏会以一个倒计时动画作为开始，以便玩家有时间来恢复对游戏的控制。

虽然没有使用游戏框架或图形框架，但是 Snail Bait 游戏使用 Node.js 和 socket.io 技术实现了将游戏中的一些指标数据发送到服务器端、高分存储和检索，并在游戏窗口顶部显示高分榜等功能。当游戏运行缓慢时，Snail Bait 游戏还将显示警告信息。而当按下“Ctrl+D”组合键启动游戏时，Snail Bait 游戏将启动开发人员后门程序，从而赋予操作人员特殊权限，例如修改时间轴的流动，或者显示 sprite 对象碰撞检测矩形以及其他功能等。

当 Snail Bait 游戏检测到自己运行在移动设备上时，将通过调用触摸事件处理句柄、调整游戏窗口大小来重新配置自身，以便适应移动设备的屏幕大小及控制方式。

本书将展示如何一步一步编程实现 Snail Bait 游戏的所有功能，这样读者就可以在自己的游戏中实现类似的功能。

本书简史

早在 2010 年，笔者从一款流行的 Android 系统下的开源游戏 Replica Island 中下载了图片和声音，并使用它们在 Android 系统中实现了 Snail Bait 游戏的初始版本。

当时，笔者已经对 HTML5 Canvas 技术产生了浓厚的兴趣，并开始写作上一本图书《HTML5 Canvas 核心技术：图形、动画与游戏开发》。在写作期间，笔者持续完善了 Snail Bait 游戏，将它从基于 Java 语言及 Android 的系统环境，迁移到了基于 JavaScript 语言、HTML5 canvas 元素及浏览器的环境中。在 2012 年写作完成时，笔者已经开发出了一个虽然原始但功能基本完整的游戏版本。

2012 年年底，笔者基于 Snail Bait 游戏开始为 IBM developerWorks 网站撰写 10 篇关于游戏编程的系列技术文章（关于这些技术文章的链接，请参阅后面的“网上资源”部分）。在接下来的 10 个月中，基于写作需要，笔者继续完善、修改了 Snail Bait 游戏。

2013 年夏天，Snail Bait 游戏已经日趋成熟。笔者前往加利福尼亚的塞瓦斯托波尔，把 Snail Bait 游戏开发的全过程放在一起，拍摄了一部长达 15 小时的 O'Reilly 视频，取名为“HTML5 2D 游戏开发”。从某个方面而言，这部视频就是本书的电影版。虽然该视频直到 9 月才得以发布，但它依然成为当年的十大畅销 O'Reilly 视频之一（关于这段视频的链接，请参阅后面的“网上资源”部分）。

2013 年 7 月，从塞瓦斯托波尔回到家之后，笔者便全身心地投入到本书的写作中。笔者改写了 IBM developerWorks 网站上的 10 篇系列技术文章，将这些文章作为本书的开始章节，并在其后添加了 10 章的新内容。在写作的整个过程中，笔者不断迭代修改 Snail Bait 游戏的代码，使其更具有可读性。

2013 年 12 月，在本书第 1 ~ 19 章完成后，笔者决定增加最后一章。该章讲解了如何使用本书介绍的技术实现一个简单完整的视频游戏，这就是第 20 章中 Bodega's Revenge 游戏的由来。

如何阅读本书

本书的写作目的简单明确：通过讲解如何实现一个复杂的视频游戏，教会读者开发属于自己的游戏。

读者可以采用多种方式阅读本书。首先，笔者为了使本书更便于阅读，在书中添加了大

量的截图、程序清单和图表。

书中灵活地使用了“注意”“提示”“警告”和“最佳实践”模块。这些模块精简了本书主要内容，每个“注意”“提示”“警告”和“最佳实践”模块都有一个标题（除了只有一行的模块之外），这样读者一眼就可以判定这些辅助内容是否与自己的实际情况有关。总体而言，本书主要是在讲解开发方法，而这些模块则重点提示了开发的原因。如果读者时间有限，那么可以抛开这些模块，直接阅读本书的主要内容，从而快速掌握游戏底层的工作机制，同时确保不会遗漏任何重要内容。

本书第 1 ~ 19 章讲解了 Snail Bait 游戏的完整开发过程，从一个简单的显示图像的游戏版本开始，涵盖了 HTML5 视频游戏开发的所有特点。第 20 章总结全书，使用本书前 19 章讲述的大部分技术内容实现了第二个视频游戏。

如果读者计划精读本书，而不仅仅是把它当作一个参考，那么建议从第 1 章或第 20 章开始阅读，其中第 20 章除了对第 1 ~ 19 章所学内容进行了回顾与综述外，还增加了部分新知识，例如极坐标系统和旋转坐标系统的应用。

假如读者从第 20 章开始阅读，甚至只是略微地浏览一下第 20 章，读者都可以对前 19 章所讲述的内容有一个大致的了解，但是不要奢望在第一次阅读时就能理解透彻。

笔者猜想大部分读者都希望把本书当作一个参考，因此除了在每章的开头部分加入该章主题外，还在每章的开始部分添加了一段关于该章内容的简要介绍。这样将给读者快速定位具体内容带来很大便利。同时，笔者还加入了许多关于如何实现具体功能的分步说明，这样读者就可以按照这些步骤实现自己的类似功能。

本书练习

被动地阅读一本书不会将任何人变成一个游戏开发程序员。读者必须通过自己动手编写一些代码来真正学会实现一款游戏的方法。为此，本书在每一章的结束部分都列出了一些练习，以帮助读者更好地学习。

为了完成练习，读者需要下载 Snail Bait 游戏源代码的最终版本，并且修改代码。在某些情况下，练习将指导读者修改某个具体版本的 Snail Bait 游戏源代码，这些版本的游戏源代码与具体的章节有关。关于 Snail Bait 游戏源代码版本划分的详细信息，请阅读下面的内容。

Snail Bait 游戏源代码及其版本划分

本书讲述的主要内容来源于两个视频游戏。读者可以通过“网上资源”部分的 URL 地址查看、下载这两个游戏及其源代码。

在阅读时，读者会发现参考 Snail Bait 游戏的源代码对于理解书中的内容非常有帮助，而通过参考与读者正在阅读的章节对应的源代码，这种帮助将更加明显。例如，在第 1 章里，我们实现了 Snail Bait 游戏的一个初始版本，简单绘制了游戏背景和游戏主角。这个版本的源代码与最终版本的源代码虽然有少量相似之处，但此时就查阅最终版本的源代码一点用都没有，大量的代码反而会让读者眼花缭乱。反倒是参考 corehtml5games.com/book/code/ch01 网址上这个对应第 1 章结束的版本更有助于读者理解第 1 章所讲述的内容。本书每章（除第 2 章外）对应的源代码版本的 URL 都按照 corehtml5games.com/book/code/ch?? 的格式设定，其中“??”代表了对应于第 01 ~ 20 章的两位数字。

如上所述，除非另有说明，否则本书每章最后部分的练习都对应于按照章节一步一步开发游戏，在章节结束时得到的 Snail Bait 游戏版本。

网上资源

本书配套网站：corehtml5games.com

Snail Bait 游戏的在线网址：corehtml5games.com/snailbait

Bodega's Revenge 游戏的在线网址：corehtml5games.com/bodegas-revenge

Snail Bait 游戏源代码下载地址：corehtml5games.com/book/downloads/snailbait

Bodega's Revenge 游戏源代码下载地址：corehtml5games.com/book/downloads/bodegas-revenge

O'Reilly 网站上作者的“HTML5 二维游戏开发”视频：shop.oreilly.com/product/0636920030737.do

IBM developerWorks 网站上作者的“HTML5 二维游戏开发”系列技术文章：www.ibm.com/developerworks/java/library/j-html5-game1/index.html

作者 2013 年在亚特兰大 HTML5 用户组会议上关于 HTML5 游戏编程的演讲视频：youtube.com/watch?v=S256vAqGY6c

作者出版的《HTML5 Canvas 核心技术：图形、动画与游戏开发》在亚马逊上的链接：<http://amzn.to/1jfuf0C>

预备知识

如果人们不能说某种语言或以这种语言书写，他们就不会考虑上这样一门创造性的书写课。同样，为了使用 HTML5 编程实现复杂游戏，读者就必须掌握 JavaScript 语言。JavaScript 语言是学习本书的前提基础。

本书程序清单所列的代码几乎都是用 JavaScript 语言编写的，但这并不代表只掌握 JavaScript 语言就够了，读者还需要掌握与之相关的 HTML 和 CSS，熟悉计算机图形学，并具备基本的数学知识。

关注自己的游戏

最后，让我们来讨论一下为什么我们能够聚在这里。笔者猜想读者之所以阅读本书是因为大家希望通过编程来实现一款属于自己的游戏。

本书按章节讨论了游戏编程的各个方面，例如实现 `sprite`、碰撞检测等。虽然它们属于 `Snail Bait` 游戏，但是读者也能轻松地将这些实现方法套用到自己的游戏当中。

本书章节顺序的编排也很重要，这些章节展示了如何从头开始一步一步编程实现一款游戏。在本书的开始部分，首先收集原始素材，搭建开发环境，然后从绘制游戏的基本图形开始编程开发。在随后的章节中，逐步向游戏中添加动画、`sprite` 对象、`sprite` 对象行为等功能，从而逐步完善 `Snail Bait` 游戏。如果读者正在从零开始开发一款游戏，那么可能会希望遵循同样的讲解方式，这样读者就可以一边学习实现功能的方法，一边在自己的游戏中实现这些功能。

在开始认真编写代码之前，大家应该多花一些时间来搭建开发环境，并尽可能地熟悉手中浏览器的开发者工具。建议大家在第 2 章结束前，尽可能缩短自己的开发周期，因为在准备阶段多花的时间将会大大提高后续的开发效率。

最后要感谢大家购买本书。笔者迫不及待地想要看到你自己创造的游戏！

David Geary

科罗拉多州，柯林斯堡

2014 年

Acknowledgements 致谢

我非常庆幸自己能够拥有一位非常优秀的编辑，他也是我 20 年写作生涯里的唯一一位编辑，一个总能从我的下一本书中读懂并接受我的观念的编辑，一个总是指导我将概念变成作品的编辑，Greg Doench。像往常一样，Greg Doench 在本书从计划写作到成书的全过程中给了我充分的帮助及指导。

Mary Lou Nohr 是一位非常棒的文字编辑，她参与编辑了我此前的所有著作。当她再次欣然同意担任本书的责任编辑时，我感到自己非常幸运。

这是我和 Alina Kirsanova 合作的第 2 本书，Alina Kirsanova 是一位非常出色的排版编辑，我的 PDF 文件经过她的处理后，总会变得非常漂亮。Julie Nahil 监督了本书的出版，使整个出版过程都能按照出版商的要求如期进行。

每当完成一本书的写作时，我都会选择那些个人认为能够帮助我改进作品的人来担任审稿人。这一次我选择了 4 位非常出色的审稿人：Jim O'Hara、Timothy Harrington、Simon Sarris 和 Willam Malone。Gintas Sanders 允许我在 Snail Bait 游戏中使用他的金币图片，并指出了 Snail Bait 游戏中的许多不足。

在为 O'Reilly 拍摄“HTML5 二维游戏开发”视频时，我有幸接触到了当时的现场观众，其中一位观众给我留下了深刻印象，他向我提出了很多问题和建议，此人就是本书的审稿人 Jim O'Hara。在此后的审稿过程中，Jim O'Hara 一如既往地认真负责，向我提出了许多好问题、好建议。

编辑 Greg Doench 介绍我认识了 Tim Harrington，他是德锐大学的一个高级学术应用分析师，并且具有游戏开发的实际经验。他的许多建议使我深受启发，并促使我重新思考如何更好地完成本书。

我一直在为本书寻找一位非常了解游戏编程的图形专家，最后我找到了 Simon Sarris。

让我喜出望外的是，他不仅是一位精通游戏编程的图形专家，还是一位优秀的作家，他提出的几种不同方法帮助我完善了本书。

最后，我很庆幸 William Malone 能够审阅本书。William 是一位专业的游戏开发人员，他开发了 Sesame Street 游戏（相关信息请查阅 <http://bit.ly/1nLSY3N>）。William 的工作对本书产生了深刻的影响，他指出了我忽略了的许多细节，特别是关于移动设备的问题。

译者序

前言

致谢

第 1 章 概述 1

1.1 Snail Bait 游戏 3

1.2 HTML5 游戏开发最佳实践 8

1.2.1 窗口失去焦点时暂停游戏 8

1.2.2 窗口重获焦点时实现倒计时
功能 9

1.2.3 使用 CSS 实现 UI 特效 10

1.2.4 对游戏运行缓慢的状态做出
检测和处理 11

1.2.5 添加社交功能 11

1.2.6 将所有游戏图像放在单独的
sprite 表单中 11

1.2.7 在服务器中存储高分榜和
实时游戏数据 13

1.3 特别功能 13

1.4 Snail Bait 游戏中的 HTML
和 CSS 15

1.5 开始 Snail Bait 游戏编程 21

1.6 本书中使用的 JavaScript 23

1.7 小结 25

1.8 练习 26

第 2 章 原始素材和开发环境 27

2.1 使用开发者工具 28

2.1.1 控制台 29

2.1.2 Chrome Canary 的帧速率
计数器 32

2.1.3 调试 34

2.1.4 时间线 35

2.1.5 Profiling 39

2.2 获取资源 40

2.2.1 图像 40

2.2.2 图像处理 41

2.2.3 音效和音乐 41

2.2.4 动画 42

2.3 使用 CSS 背景 43

2.4 生成小图标 45

2.5 缩短编码周期 46

2.6 小结 48

2.7 练习 48

第3章 图形和动画 49

- 3.1 使用 HTML5 canvas 元素绘制图形和图像 51
 - 3.1.1 绘制背景 52
 - 3.1.2 绘制跑步小人 53
 - 3.1.3 绘制平台 53
- 3.2 实现平滑的 HTML5 动画 55
 - 3.2.1 requestAnimationFrame() 方法 57
 - 3.2.2 requestAnimationFrame() 的 polyfill 实现 57
- 3.3 实现游戏主循环 60
- 3.4 计算帧速率 61
- 3.5 滚动游戏背景 62
 - 3.5.1 平移坐标系 62
 - 3.5.2 滚动 Snail Bait 游戏的背景 65
- 3.6 制作基于时间的运动 67
- 3.7 反转滚动方向 67
- 3.8 绘制动画帧 68
- 3.9 使用视差产生视深的假象 68
- 3.10 小结 71
- 3.11 练习 71

第4章 游戏的基本架构 72

- 4.1 在 JavaScript 对象中封装游戏函数 73
 - 4.1.1 SnailBait 的构造函数 74
 - 4.1.2 SnailBait 的原型 75
- 4.2 理解 JavaScript 语言中挑剔的 this 指针 77
- 4.3 处理键盘输入 79
- 4.4 玩家按 p 键时暂停或者恢复游戏 81

- 4.5 冻结游戏，确保它精确地在暂停的位置恢复 82
- 4.6 在窗口失去焦点时暂停游戏 83
- 4.7 使用动态的倒计时来恢复一个暂停的游戏 84
 - 4.7.1 向游戏玩家显示提示板 (简短信息) 85
 - 4.7.2 Snail Bait 的倒计时 87
- 4.8 小结 89
- 4.9 练习 90

第5章 游戏加载动画 91

- 5.1 定义 Snail Bait 游戏的窗口 93
- 5.2 使用 CSS 过渡让元素淡入、淡出 96
 - 5.2.1 让元素淡入 97
 - 5.2.2 让元素淡出 98
 - 5.2.3 snailbait-toast 元素的 CSS 99
 - 5.2.4 显示和隐藏提示板 100
- 5.3 让拥有与 CSS 过渡属性 Opacity 关联的元素淡入、淡出 103
- 5.4 实现加载动画 105
- 5.5 显示游戏画面 109
- 5.6 小结 112
- 5.7 练习 113

第6章 sprite 对象 114

- 6.1 sprite 对象 116
 - 6.1.1 sprite 对象的属性 118
 - 6.1.2 sprite 对象的构造函数 119
 - 6.1.3 sprite 对象的方法 119

6.2 将 sprite 对象整合到游戏 主循环中	121
6.3 实现 sprite 对象的 artist 对象	124
6.3.1 图形 artist 对象	124
6.3.2 图像 artist 对象	125
6.3.3 sprite 图像表单 artist 对象	125
6.3.4 定义 sprite 图像表单单元格	128
6.4 创建和初始化游戏的 sprite 对象	130
6.5 使用元数据定义 sprite 对象	133
6.6 滚动 sprite 对象	136
6.7 小结	138
6.8 练习	138

第 7 章 sprite 对象行为

7.1 行为基础	142
7.2 跑步小人行为	143
7.3 跑步小人的跑动行为	146
7.4 轻量级行为	148
7.5 游戏独立行为	150
7.6 组合行为	155
7.7 小结	159
7.8 练习	159

第 8 章 时间轴, 第 1 部分: 有限 行为及线性运动

8.1 实现一个初始的跳跃行为算法	161
8.2 将处理跳跃行为的职责转交给 跑步小人	162
8.3 实现跳跃行为	164
8.4 使用秒表记录动画时间	165

8.5 重新定义跳跃行为	168
8.6 实现线性运动	170
8.6.1 上升阶段	170
8.6.2 下降阶段	171
8.7 暂停行为	173
8.8 小结	175
8.9 练习	175

第 9 章 时间轴, 第 2 部分: 非线性 运动

9.1 理解时间及其衍生物	177
9.2 使用动画计时器和缓变功能 实现非线性跳跃	177
9.3 实现动画计时器	179
9.4 实现缓变功能	181
9.5 微调缓变功能	184
9.6 实现真实的跳动行为	185
9.7 使行为随机化	189
9.8 使用动画计时器和缓变功能 实现非线性颜色改变	190
9.9 小结	193
9.10 练习	193

第 10 章 时间轴, 第 3 部分: 时间 系统

10.1 Snail Bait 游戏的时间系统	195
10.2 创建和启动时间系统	197
10.3 将时间系统整合进 Snail Bait 游戏中	198
10.3.1 使用时间系统驱动游戏 动画	198

10.3.2	实现使用时间系统修改游戏 时间流的函数	199
10.3.3	在计算帧速率时分解 时间比率	199
10.3.4	使用时间系统暂停和 恢复游戏	200
10.4	重新定义秒表和动画计时器的 当前时间	202
10.5	实现时间系统	206
10.6	小结	208
10.7	练习	208
第 11 章	碰撞检测	209
11.1	碰撞检测过程	210
11.2	碰撞检测技术	211
11.3	Snail Bait 游戏中的碰撞检测	212
11.3.1	sprite 对象的碰撞矩形	212
11.3.2	跑步小人的碰撞行为	214
11.4	为碰撞检测选择碰撞候选 对象	215
11.5	在跑步小人与其他 sprite 对象 之间检测碰撞	216
11.6	处理碰撞	217
11.7	优化碰撞检测	220
11.7.1	改善包围盒	220
11.7.2	使用空间分割方法	221
11.8	监测碰撞检测的运行性能	222
11.9	编程实现碰撞检测的边界 案例	223
11.10	小结	226
11.11	练习	226

第 12 章	重力	228
12.1	为跑步小人实现坠落行为	228
12.2	实现重力	230
12.2.1	跑步小人的坠落行为	232
12.2.2	计算初始下降速度	236
12.2.3	在跑步小人坠落过程中 暂停游戏	237
12.3	最终版的碰撞检测	237
12.4	小结	238
12.5	练习	239
第 13 章	sprite 动画和特殊效果	240
13.1	实现 sprite 动画	241
13.2	创建特效	246
13.2.1	晃动游戏画面	246
13.2.2	场景切换	248
13.3	编排效果	253
13.3.1	炸死蜜蜂	254
13.3.2	引爆按钮	256
13.4	小结	258
13.5	练习	258
第 14 章	声音和音乐	259
14.1	创建声音和音乐文件	261
14.2	加载音乐和音效	262
14.3	设置声音和音乐控件	263
14.4	播放音乐	264
14.5	循环播放音乐	265
14.6	播放音效	267
14.6.1	创建 sprite 音频对象	270
14.6.2	定义音效对象	271