

TURING

图灵程序设计丛书

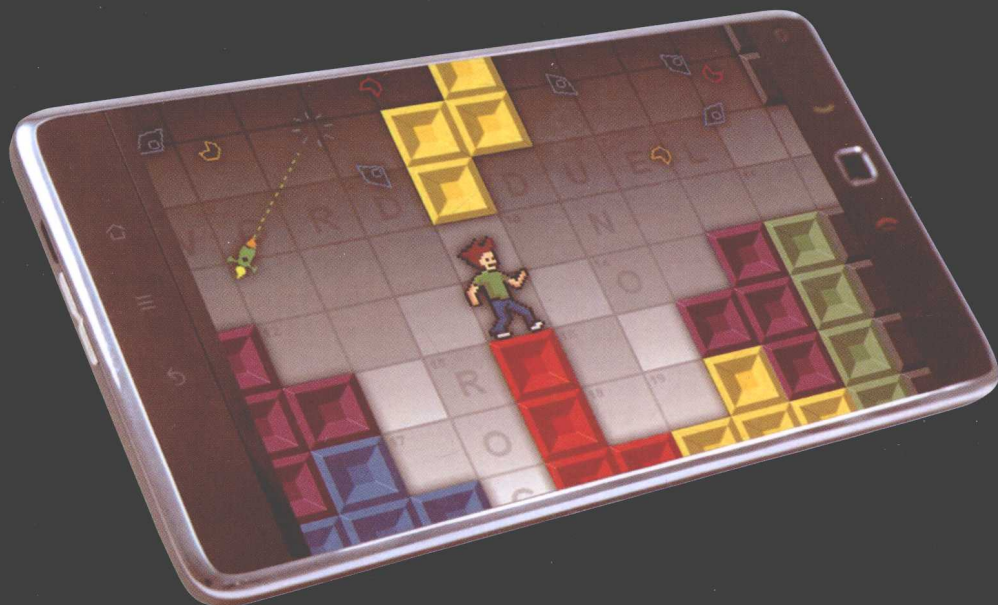


Real-World Flash Game Development

How to Follow Best Practices and Keep Your Sanity **Second Edition**

实战Flash游戏开发 (第2版)

[美] Christopher Griffith 著
李鑫 杨海玲 译



人民邮电出版社
POSTS & TELECOM PRESS

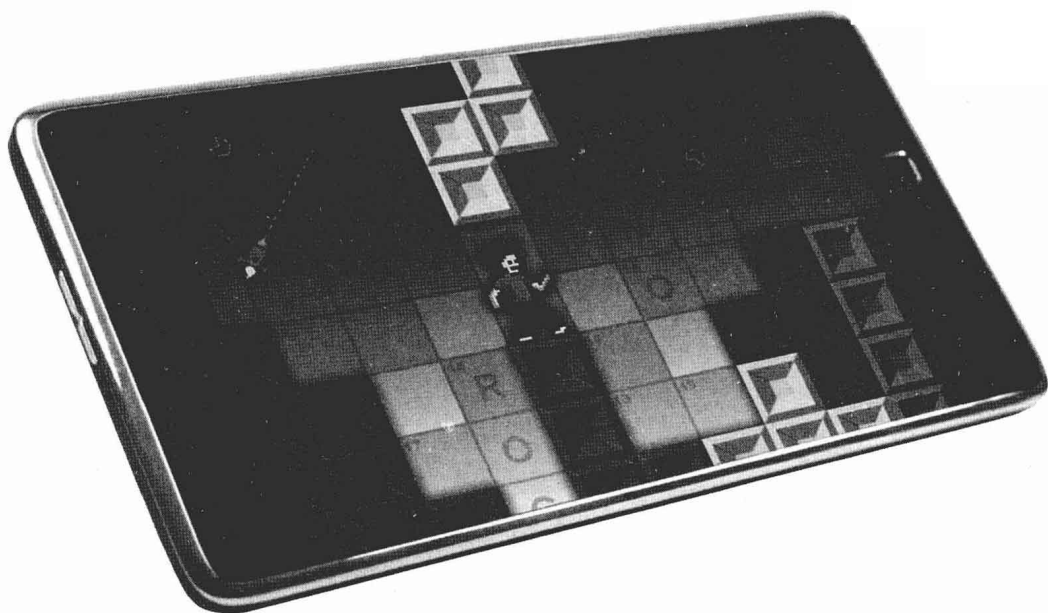
TURING 图灵程序设计丛书

Real-World Flash Game Development

How to Follow Best Practices and Keep Your Sanity **Second Edition**

实战Flash游戏开发 (第2版)

[美] Christopher Griffith 著
李鑫 杨海玲 译



人民邮电出版社

北京

图书在版编目 (CIP) 数据

实战Flash游戏开发：第2版 / (美) 格里菲斯
(Griffith, C.) 著；李鑫，杨海玲译. — 北京：人民
邮电出版社，2012.5

(图灵程序设计丛书)

书名原文: Real-World Flash Game Development:
How to Follow Best Practices and Keep Your Sanity,
Second Edition

ISBN 978-7-115-27645-2

I. ①实… II. ①格… ②李… ③杨… III. ①动画制
作软件, Flash IV. ①TP391.41

中国版本图书馆CIP数据核字(2012)第062879号

内 容 提 要

本书细致深入地阐述了 Flash 游戏开发的完整过程，尤其针对 Flash CS5 与 ActionScript 3，展开介绍了如何充分调动各种元素，简化流程，高效创建有用代码，进而开发制作出高水准的 Flash 游戏。新版增加了几章 Flash 移动开发的内容，介绍如何将 Flash 游戏部署到 iPhone 和 Android 设备上。

本书适用于 Flash 游戏设计开发人员以及任何想做网络游戏的程序员。

图灵程序设计丛书

实战Flash游戏开发 (第2版)

-
- ◆ 著 [美] Christopher Griffith
 - 译 李 鑫 杨海玲
 - 责任编辑 王军花
 - 执行编辑 丁晓昀
 - ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街14号
 - 邮编 100061 电子邮件 315@ptpress.com.cn
 - 网址 <http://www.ptpress.com.cn>
 - 北京鑫正大印刷有限公司印刷
 - ◆ 开本: 800×1000 1/16
 - 印张: 27.5
 - 字数: 650千字 2012年5月第2版
 - 印数: 5001-9 000册 2012年5月北京第1次印刷

著作权合同登记号 图字: 01-2011-3963号

ISBN 978-7-115-27645-2

定价: 79.00元

读者服务热线: (010)51095186转604 印装质量热线: (010)67129223

反盗版热线: (010)67171154

版 权 声 明

Real-World Flash Game Development: How to Follow Best Practices and Keep Your Sanity, Second Edition by Christopher Griffith.

ISBN: 978-0-2408-1768-2.

Copyright © 2012 by Elsevier. All rights reserved.

Authorized Simplified Chinese translation edition published by the Proprietor.

Copyright © 2012 by Elsevier (Singapore) Pte Ltd.

All rights reserved.

Printed in China by POSTS & TELECOM PRESS under special arrangement with Elsevier (Singapore) Pte Ltd. This edition is authorized for sale in China only, excluding Hong Kong SAR, Macao SAR and Taiwan Province. Unauthorized export of this edition is a violation of the Copyright Act. Violation of this Law is subject to Civil and Criminal Penalties.

本书简体中文版由 Elsevier (Singapore) Pte Ltd. 授权人民邮电出版社在中华人民共和国境内（不包括香港特别行政区、澳门特别行政区和台湾地区）出版与发行。未经许可之出口，视为违反著作权法，将受法律之制裁。

本书贴有 Elsevier 防伪标签，无标签者不得销售。

引 言

自从写作本书以来，感觉好像已去经年。它在第一年就卖了 4500 本，远远超出了我最理想的预期目标。非常感谢那些买书的人以及花时间向他人推荐本书的人。不过，科技的发展永不停步，优秀作者也不能驻足于过去的成就。本书是第 1 版的修订版，上一版的内容多半都囊括在内（尽管以后你只能在网上看到某些内容），另外还添加了一些我认为是崭新且激动人心的高级内容，比如移动开发。

游戏开发非常奇怪，它融合着很多技巧与表现风格。游戏可以被称为一种最复杂的娱乐形式。高质量的代码、迷人的艺术设计以及良好的用户界面固然重要，但很多游戏佳作都有一个共同特点——好玩。这种趣味因素显得尤为意味深长，因为这完全是由玩家的主观性来判定的。不同行业的不同人对游戏的兴趣也可能迥然不同。很少会有一款游戏能够一直吸引全球各地的所有人。

据说，因特网上的玩家最喜欢的游戏是那种“休闲”类型的游戏。休闲游戏应该具有广泛的受众，它应该更关注如何使游戏简单易玩，而不是具有一定的深度或真实感。这并非是说有些休闲游戏就不具备深度与真实感，但是请想想看，一类玩家是在游戏主机上玩复杂战略模拟游戏的，而另一类则只在工余午休时间才有可能玩上 10 分钟，这两类玩家还是有很大区别的。从经典的街机游戏（比如《吃豆人》）到逻辑益智游戏（比如《俄罗斯方块》），休闲游戏的类型变幻无穷。实际上，我刚提到的这两款游戏都有一个共同点：它们都是在 20 世纪 70 年代末到 20 世纪 80 年代中期开发的游戏，那时，开发者们所关心的不是游戏的图像与音效是否具有如影片般恢弘的效果，而是游戏是否好玩。

Flash 游戏

既然你选择了本书，那么我想你可能关心的还不只是如何开发游戏，你可能更关心如何用 Flash 来开发游戏。Flash 是一个非常出色的游戏开发平台，它特别适于开发网络休闲游戏。选用 Flash 开发游戏是很明智的，因为其插件文件尺寸很小，性能也很优秀，全球 98% 的网民都安装了这种插件，这些都会使你开发的游戏能尽量被玩家接触到。人们曾经认为 Flash 游戏是有缺陷的、粗制滥造的低端产品。但这种情况正迅速改观，因为 Flash 游戏现在变得越来越精致，也更接近于“传统”的电脑与视频游戏了。

使用何种 Flash 版本

在此，我觉得应该花一点儿时间来谈谈 Flash 的版本问题。本书第 1 版所使用的是 Flash CS4。

那时它已经发布了差不多有整整一年了，所以比较适于用作开发工具。而到了 2010 年春，Adobe 发布了 Flash CS5，故此本书将主要用它来进行开发。除了最后那两个手机游戏之外，所有范例都可以利用 CS5，而并不需要其他新出的工具（即使那些工具在技术上能做到，我也还是没有采用，稍后再予以详述）。就在写这本书时，我也在用 CS5.5 的 beta 版。由于它是在本书上架销售之时正式发布的，所以我觉得没能及时地介绍一些 CS5.5 中的特殊功能，大意了。^①CS5.5 的一些新流程或选项可以让人更轻松的处理一些具体任务，我会在书中多加介绍这些内容。此外，为了利于 Android 和 iOS 开发，CS5.5 还清除了一些存在于 CS5 中的较为零散的流程和选项，因此书中也将给出 CS5.5 的屏幕截图，因为它将是未来的标准。第 15 章和第 16 章的范例虽说可以用 CS5 来实现（不过还需从 Adobe 网站上下载一些额外的工具），但就性能、选项以及工具的易用性而言，使用 CS5.5 无疑更为理想。

如何充分利用本书

本书是为下列两种读者而准备的：对 Flash（CS5、CS5.5 或者更早版本）这种动画及网站创建工具的掌握程度在中等以上的读者，正打算换用 Flash 开发游戏的其他平台开发者。本书并不打算从零开始讲解 Flash 基本用法，因为其他一些专业作者及讲师已就此写过很多书了。我更希望当读完本书时，你能够完全自信地用 Flash 开发游戏。

本书首先介绍了很多游戏开发基本术语及概念，这些都是你应该理解的内容，另外还介绍了如何在一页纸上规划出游戏的整个流程。接下来，本书将介绍如何在 Flash 中管理音频及可视化资源，如何处理游戏逻辑（我们将分析一个完整的游戏脚本中的每个核心功能），以及究竟如何构建游戏才能免去将来可能出现的麻烦。我还将介绍一些关于代码及库组织方式的最有效的方法。

Flash 中的问题往往可以用很多方法来解决，游戏也不例外。有时一些外部的力量（比如说客户要求或者截止日期等）会左右我们所采用的方法。本书第三部分将介绍在一些实际情况如何应用前半部分所学内容，有了这些实战指导，当开发时间只剩一半时，你就不必为了赶进度而牺牲那些良好的开发原则了。

最后，我们将在这一版中介绍在移动环境中的 Flash 应用，以及如何针对这种媒介进行优化。这些范例不仅探讨了针对 iPhone 的对象打包程序，而且还介绍了如何将游戏配置到 AIR for Android 上。

网站上的资源

本书还有一个配套网站 www.flashgamebook.com，上面提供了你将要用到的本书全部的源文件^②，你还可以借用它们来开发自己的作品。你可以在上面下载到本书所有范例的源代码以及从上一版的纸书中“退休”的几章内容。^③该网站还可以让读者提问，而且还会公布某些已确定了更新和勘误。一定要来转转哦。

① 在翻译本书时，Adobe 已推出了 CS5.5 正式版。——译者注

② 本书所有源代码也可以在图灵网站（ituring.com.cn）本书主页免费下载。——编者注

③ 本书已将原书网站上的资源翻译整理，分别对应第 17、18、19 章以及附录 D。——编者注

目 录

第 1 章 计算机科学并不适合所有人	1	1.5.4 包	9
1.1 一些基础知识	1	1.5.5 创作时事件、编译时事件及 运行时事件	10
1.2 常见游戏类型	1	1.6 醒一醒	10
1.2.1 冒险类游戏	1	第 2 章 最佳编程工具	11
1.2.2 动作类游戏	1	2.1 倒叙几句	11
1.2.3 益智解谜类游戏	2	2.2 Flash 的一些优点	12
1.2.4 词汇类游戏	2	2.2.1 玩家占有率	12
1.2.5 策略与模拟经营类游戏	3	2.2.2 灵活性	12
1.2.6 角色扮演游戏	4	2.2.3 上市速度	12
1.2.7 驾驶类游戏	4	2.2.4 美观性	13
1.2.8 桌面式和卡牌式游戏	4	2.3 白璧微瑕	13
1.3 常用开发术语	5	2.3.1 缺点：代码编辑器	13
1.3.1 伪码	5	2.3.2 解决方案：使用附加工具	14
1.3.2 算法	5	2.3.3 缺点：性能及内存管理	14
1.3.3 过程式编程	5	2.3.4 解决方案：使用第三方解 决方案或自己创建	15
1.3.4 面向对象编程	5	2.3.5 缺点：内容调试	16
1.3.5 设计模式	6	2.3.6 解决方案：使用跟踪输出 与自定义工具	16
1.3.6 类	6	2.3.7 缺点：缺乏内建游戏库和 工具	17
1.3.7 Public、Protected、Private 和 Internal	7	2.3.8 解决方案：自己开发工具 或寻找开源工具	17
1.4 游戏特有的开发术语	7	2.4 不要和 Flash 对抗	17
1.4.1 人工智能	7	2.5 Flash 所能创建的内容	18
1.4.2 游戏循环（或主循环）	7	2.5.1 动画与游戏	18
1.4.3 游戏视角	8	2.5.2 应用程序与游戏	18
1.4.4 卷动背景	8	2.5.3 网站与游戏	18
1.4.5 区块式游戏	8		
1.5 Flash 程序开发术语	9		
1.5.1 舞台	9		
1.5.2 显示对象	9		
1.5.3 事件和侦听器	9		

2.5.4 Flash 与传统游戏开发	19	4.5.1 Object	50
2.6 最佳创作工具	19	4.5.2 数组	51
第 3 章 一份计划抵得上 1000 片阿司匹林	20	4.5.3 Vector	53
3.1 步骤 1	20	4.5.4 Dictionary	53
3.2 步骤 2	20	4.5.5 ByteArray	54
3.3 步骤 3	21	4.5.6 到底该使用哪种列表	54
3.4 步骤 4	23	4.5.7 自定义数据结构	55
3.5 步骤 5	24	4.6 部分 5: 让所有人都能读懂你的注释	55
3.6 步骤 6 (可选)	26	4.7 部分 6: Flash 为什么会这么做	56
第 4 章 //注释太棒了!	29	4.7.1 事件流程	56
4.1 适当的提醒	29	4.7.2 帧脚本	57
4.2 部分 1: 类	29	4.7.3 使用多个 SWF 文件	59
4.2.1 包	30	4.7.4 内存垃圾回收	60
4.2.2 类就像文件一样	30	4.8 本章小结	62
4.2.3 构造函数	30	第 5 章 最低要求与架构师所采用的方法	63
4.2.4 常量、变量与方法	31	5.1 简单地进行封装——类与容器	63
4.2.5 getter 与 setter 方法	32	5.2 将有关数值存储到变量与常量中	64
4.2.6 类标识符	33	5.3 不要信任舞台	65
4.2.7 继承与多态	34	5.4 不要使用不理解或不适用的架构或设计模式	65
4.2.8 接口	36	5.5 要把握好向别人展示自己作品的最佳时机	66
4.2.9 将 Flash 中的资源与类链接起来	38	5.6 转换到架构	66
4.2.10 类与基类	39	5.7 OOP 概念	66
4.2.11 使用没有类文件的导出元件	41	5.7.1 封装	67
4.2.12 getDefinitionByName 与 Casting	41	5.7.2 继承	67
4.3 部分 2: 事件	42	5.7.3 多态性	68
4.3.1 dispatchEvent 方法	42	5.7.4 接口	68
4.3.2 addEventListener 方法、removeEventListener 方法以及事件流程中的各阶段	43	5.8 游戏开发中的实用 OOP 技巧	69
4.3.3 事件的传播与取消	46	5.9 单例模式: 一种良好的文档模式	70
4.3.4 自定义事件	46	5.10 本章小结	72
4.4 部分 3: 错误	47	第 6 章 管理资源与使用图像	73
4.4.1 try、catch、finally	48	6.1 一种更好的文件格式	73
4.4.2 抛出自定义错误	49	6.2 小议组织库元件	74
4.5 部分 4: 数据结构与列表	50	6.3 使用图像	75
		6.4 常用的光栅图像格式	76

6.4.1 压缩	79	9.6 本章小结	129
6.4.2 平滑	80	第 10 章 XML 与动态内容	130
6.4.3 解块	80	10.1 引入数据: 理解 URLLoader 类	130
6.4.4 外部的图像编辑工具	80	10.2 XML	130
6.5 本章小结	82	10.3 E4X	131
第 7 章 让它动起来: ActionScript		10.4 纵横字谜	132
动画	83	10.4.1 CrosswordTile 类	135
7.1 几个术语	83	10.4.2 CrosswordClue 类	138
7.1.1 缓动	83	10.4.3 CrosswordPuzzle 类	140
7.1.2 序列	84	10.5 内容并非只有现成的: 纵横字谜	
7.2 是否补间? 这是个问题	84	编辑器	150
7.3 由脚本驱动的简单射击游戏	85	10.6 数据传送	151
7.3.1 Projectile 类	85	10.7 示例: XML 与 Flash Vars 的应用	
7.3.2 SimpleShooter 类	85	对比	151
7.4 记忆力游戏: 补间动画	88	10.8 本章小结	153
7.4.1 MemoryCard 类	88	第 11 章 就是一个词儿: 数学	154
7.4.2 Memory 类	89	11.1 Math 类	154
7.5 本章小结	93	11.2 部分 1: 几何学及三角学	154
第 8 章 如雷贯耳: 如何使用音频	94	11.3 简要介绍一下弧度与 Pi 值	158
8.1 常用音频格式	94	11.4 Flash 中的 3D	161
8.2 常用导出设置	95	11.4.1 位置	162
8.3 使用外部文件	97	11.4.2 旋转	162
8.4 声音编辑工具	98	11.5 透视投影	163
8.5 编写音效脚本	98	11.6 Simple TunnelShooter 范例	164
8.5.1 理解与声音有关的类	99	11.6.1 基本机制	164
8.5.2 SoundEngine 类	99	11.6.2 所需类文件	164
8.5.3 使用 SoundEngine 类	111	11.6.3 Tunnel 类	165
8.5.4 SoundMixer 类	114	11.7 部分 2: 物理学	179
第 9 章 重塑“视频游戏”	116	11.7.1 标量	179
9.1 视频编解码器	116	11.7.2 矢量	179
9.2 外部视频应用: 过场与菜单	116	11.7.3 Vector3D 类	179
9.3 CutsceneManager 类	119	11.7.4 位移	180
9.4 时间轴上的视频	125	11.7.5 速度	180
9.4.1 文件尺寸	125	11.7.6 加速度	180
9.4.2 库: 易于使用与杂乱无章	126	11.7.7 摩擦力	180
9.4.3 性能表现	126	11.7.8 惯性	180
9.4.4 自由的动态模糊	126	11.7.9 物理仿真与假象	181
9.5 建立内部视频	126	11.7.10 真实性与期望值	181

11.8 范例：顶视角的驱动引擎.....	181	14.2 数据流程.....	234
11.8.1 Vehicle 类.....	181	14.3 游戏流程与功能.....	234
11.8.2 Time 类.....	184	14.3.1 游戏背景设定.....	235
11.8.3 Game 类.....	185	14.3.2 关卡设计及墙壁的作用.....	235
11.9 范例：顶视角且带有漂移效果的 赛车游戏.....	189	14.3.3 出口.....	235
11.10 本章小结.....	192	14.3.4 玩家角色.....	235
第 12 章 别碰我：碰撞检测技术	193	14.3.5 物品.....	236
12.1 能力与需求.....	193	14.3.6 敌人.....	236
12.2 hitTestObject——最基本的检测 方法.....	193	14.4 关卡文件格式以及资源结构.....	237
12.3 hitTestPoint——更先进的检测 方法.....	194	14.4.1 关卡的 XML 文件.....	237
12.4 半径/距离检测法——极适合圆形 的检测方法.....	199	14.4.2 资源 SWF 文件.....	240
12.5 矩形边框检测法.....	201	14.4.3 游戏所用类的概述.....	240
12.5.1 新的 Enemy 类.....	201	14.5 引擎类.....	241
12.5.2 SimpleShooterCollisions 类 所新增的内容.....	202	14.5.1 ISprite 接口.....	241
12.5.3 该方法的一些不足之处.....	204	14.5.2 IPlayer 接口.....	242
12.6 像素级精确的碰撞侦测与物理 仿真.....	205	14.5.3 IEnemy 接口.....	243
12.7 这些办法都不灵？那就用鸡尾酒 疗法.....	206	14.5.4 IItem 接口.....	243
第 13 章 MixUp——简单的游戏引擎	207	14.5.5 IPortal 接口.....	244
13.1 主文档.....	209	14.6 IWall 接口.....	244
13.2 MixUP 类.....	209	14.6.1 PlatformerEvent 类.....	245
13.3 Title 类.....	212	14.6.2 PortalDestinations 类与 PortalRequirement 类.....	245
13.4 RulesPanel 类.....	212	14.6.3 PlatformerConfig 类.....	246
13.5 Game 类.....	213	14.6.4 GridReference 类.....	247
13.6 接口.....	216	14.7 CollisionGrid 类.....	249
13.7 GameBoard 类.....	218	14.8 游戏实现类.....	267
13.8 SourceImageEmbedded 类.....	225	14.9 资源类.....	269
13.9 GameHistory 类与 Results 类.....	227	14.9.1 Player 类.....	269
13.10 SourceImageCamera 类.....	229	14.9.2 Enemy 类.....	271
13.11 本章小结.....	232	14.9.3 Item 类.....	272
第 14 章 技术大融合：制作平台游戏	233	14.9.4 Portal 类与 Wall 类.....	273
14.1 平台类型游戏.....	233	14.9.5 游戏中的资源.....	275
		14.10 本章小结.....	277
		第 15 章 Marble Runner：我们的 第一款手机游戏	278
		15.1 部分 1：开发 iOS 游戏的最佳 实践.....	278
		15.1.1 滤镜（包括由 PixelBender 所创建的滤镜）.....	279

15.1.2	矢量图形（及形状补间）	279	15.9.12	Results 类	321
15.1.3	文本处理	280	15.10	游戏设计注意事项	323
15.1.4	补间动画	280	15.10.1	关卡设计	323
15.1.5	绘图 API、遮罩以及混合模式	281	15.10.2	风险与回报	323
15.1.6	运行时加载的 SWF 文件	281	15.11	有待改进之处	324
15.2	利用 GPU	281	15.11.1	计分	324
15.3	代码也很关键	282	15.11.2	可拾取物品	324
15.3.1	声明类型	283	15.11.3	可卷动的关卡	324
15.3.2	使用静态属性与函数	283	第 16 章	桌上冰球：采用多点触摸的平板电脑多人游戏	325
15.3.3	确定最终答案	283	16.1	主题三重奏	325
15.3.4	放之四海而皆准的垃圾回收	284	16.2	移动设备的多点触摸输入方式	326
15.3.5	避免使用极其庞大的架构和库	284	16.3	有限状态机	326
15.3.6	减少显示列表层级	285	16.4	利用 Box2D 来进行物理仿真	327
15.3.7	无需用事件来处理函数所擅长之处	285	16.5	双人桌上冰球游戏	328
15.4	权衡：继承还是接口	286	16.5.1	XFL 文件	328
15.5	程序实例	286	16.5.2	游戏所用类	329
15.5.1	XFL 文档	287	16.5.3	Main.as	329
15.5.2	Space.as	287	16.5.4	Title.as	331
15.5.3	Rock.as	288	16.5.5	Rules.as	331
15.5.4	文档类 SpaceRocks.as	290	16.5.6	GameTouchController.as	333
15.5.5	FrameRateProfiler 类	292	16.5.7	Game.as	336
15.6	部分 2：Marble Runner	293	16.5.8	AirHockeyEngine.as	345
15.7	Accelerometer 类	293	16.5.9	准备创建	353
15.8	加速计数据是如何计算出来的	294	16.6	本章小结	353
15.9	Marble Runner 游戏	295	第 17 章	找到就干掉：猎虫行动	354
15.9.1	XFL 文件	296	17.1	bug	354
15.9.2	游戏所用到的类	296	17.1.1	trace	354
15.9.3	LabyrinthEngine	297	17.1.2	Flash Tracer	356
15.9.4	LabyrinthLevel	303	17.1.3	调试器	357
15.9.5	IBall、IHazard、IWall	306	17.2	性能与优化	358
15.9.6	基本文档类	307	17.2.1	FrameRateProfiler 类	359
15.9.7	标题界面类	309	17.2.2	MemoryProfiler 类	362
15.9.8	得分排行榜界面类	310	17.2.3	sampler 包	366
15.9.9	GameClip 类	314	17.3	本章小结	369
15.9.10	Marble 类	317	第 18 章	当心恶贼	370
15.9.11	HazardPit 和 StandardWall 类	320	18.1	恶意使用	370
			18.1.1	关闭不再需要的事件侦听器	370

18.1.2 为所要接受的输入设置最低 延时	371	19.4.4 供给配置文件	387
18.1.3 一经发现恶意使用就将游戏 关闭	371	19.4.5 DemoApp	388
18.2 数据保护	371	19.4.6 XFL 文件	388
18.2.1 内存攻击	371	19.4.7 文档类	389
18.2.2 保护要发送与接收的数据	375	19.5 仅是开端	394
18.3 SWF 文件保护	376	19.6 改变设置	395
18.4 本章小结	377	19.6.1 文档分辨率	395
第 19 章 移动开发介绍	378	19.6.2 发布设置	396
19.1 移动设备：新的热点	378	19.6.3 部署	398
19.2 新增 API	379	19.7 本章小结	398
19.3 开发流程：先 A 后 I	379	后记 Flash 在游戏业中的前景	399
19.3.1 编译	379	附录 A 摄像头与麦克风	400
19.3.2 程序设置与设计	381	附录 B 本地化	408
19.4 第一个 iPhone 程序	382	附录 C JSFL——情人们的 JavaScript	415
19.4.1 开发者证书	382	附录 D 使用 AMFPHP 开发游戏	422
19.4.2 UDID	385		
19.4.3 AppID	386		

第 1 章

计算机科学并不适合所有人

1.1 一些基础知识

在深入介绍Flash之前，了解一点游戏开发的基础知识十分重要，这样我们就能理解书中所用的术语了。如果以后你忘了某个术语的含义，或者不明白它在特定情况下的用法，那就请重新翻阅本章。要是你对所有这些长单词与抽象概念有些不知所措，也不要担心！我们得承认，游戏开发本来就是一项复杂工程（尤其对于高效且熟练的开发而言）。要知道，每个编写游戏的家伙都经历过同样的焦虑和困惑。如同生活中的任何事情一样，只有不断练习并加以实战才能变得精通。因此，拿起一杯你喜爱的能提神的饮料，让我们开始游戏编程之旅吧！

1.2 常见游戏类型

尽管有许多不同类型的游戏（况且有些游戏自身无法被简单归类，它们甚至还以此为荣），但多数游戏却可归于下列类别之一。

1.2.1 冒险类游戏

冒险类游戏（见图1-1）通常是由背景故事为导向而展开的，并且一般会有一到多个主角。从玩家感觉上来说，玩这类游戏很像是看电影（有些游戏据说快要拍成电影了），它们大量依靠剧情对白、搜索探险以及解决逻辑谜题来推动玩家的游戏进程。在20世纪80年代末与20世纪90年代初，冒险类游戏非常流行，其中LucasArts公司和Sierra（雪乐山）公司都出品过一些堪称绝佳典范的冒险类游戏。由于其开发流程非常依赖美术功底，并且对系统配置的要求通常也较低，所以用Flash制作的这类游戏又开始流行起来。

1.2.2 动作类游戏

此类游戏（见图1-2）可能会涉及许多玩法以及子类别，但就一般而言，动作类游戏考验的是一个玩家的机敏度、反应时间以及在面临压力时是否能作出正确的判断。第一人称射击游戏、横向与纵向卷轴游戏及格斗游戏都可归入这一类。Flash平台可以很好地支持动作类游戏中的一些子类别，特别是那种早期的动作类游戏，比如像《太空入侵者》或《超级马里奥兄弟》那样的游戏。

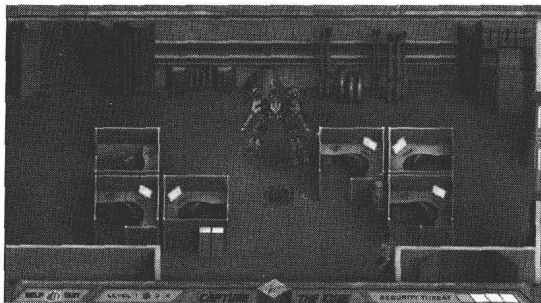


图1-1 冒险游戏《自动售货机变形金刚》（*Mountain Dew——Capture the Cube*）

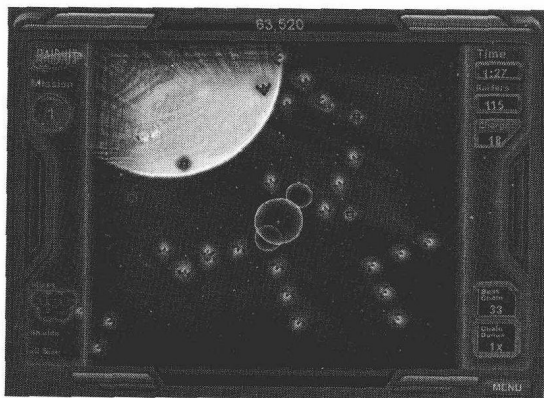


图1-2 动作类游戏《辐射光环》（*Raidiux*）。版权归Blockdot公司（www.blockdot.com）所有

1.2.3 益智解谜类游戏

想想《俄罗斯方块》、《宝石迷阵》（*Bejeweled*）还有《数独酷》（*Sudoku*）吧！像这样的游戏还可以列出很多。凡是涉及逻辑推理、解谜、图案匹配，或者包含以上所有这些构成元素的作品都可归入此类游戏。这类Flash游戏很多，原因主要有以下两方面。首先，创作一款简单的益智解谜类游戏不需要太多的设计工作，这意味着独立开发者经常靠自己就能制作完成。另外，网上主要的休闲游戏玩家年龄都偏大，他们一般比较喜欢节奏慢些的益智解谜类游戏。

1.2.4 词汇类游戏

这类游戏可被认为是益智解谜类游戏的一个子类，但是创建方法却大为不同，所以我将它们划归到独立的体系。寻找单词、纵横拼字谜、拼字游戏以及回文构词^①，都属于此类游戏（见图1-3）。基于和其他益智解谜类游戏相同的原因，Flash平台是此类游戏的最常见载体。

① anagram，用一组随机选出的字母构成新单词。——译者注

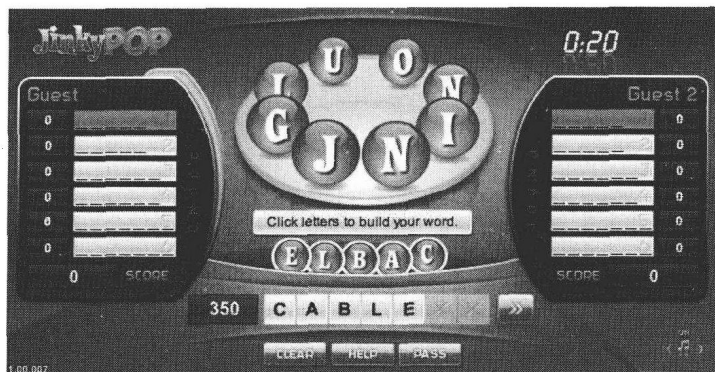


图1-3 JinkyPOP。版权归Blockdot公司（www.blockdot.com）所有

1.2.5 策略与模拟经营类游戏

我承认，把这两类游戏归并为一类确实有偷懒之嫌。但这两者有一些共同点。策略类游戏一般需要精心筹划、资源管理以及制定决策，比如说规划一个城市或者组建一支庞大的军队。策略类游戏与模拟经营类游戏的区别经常可用玩家需要维持的游戏元素细节度来界定。有些游戏会非常复杂，以至于要想进行宏观管理，玩家可能就要用到游戏提供的全部选项。而更多休闲型策略类游戏（比如用Flash创建的多数策略类游戏）则采用一些措施来简化游戏玩法，比如通过削减可用选项以及只关注一些主要任务。塔防类游戏可算是休闲型策略类游戏的一个常见子类。在这种游戏中，玩家要策略地摆放各种不同的武器来阻止敌人通过防线。（见图1-4。）



图1-4 《少女、修士与食人魔》（The Maiden, Monk, and Ogre）。版权归Blockdot公司（www.blockdot.com）所有

1.2.6 角色扮演游戏

角色扮演游戏 (RPG) 类似于冒险类游戏, 但通常对主角在游戏故事进程中不断成长的历程要描述得更多。传统上, RPG都发生在幻想设定的故事背景中, 并且它注重于玩家统计数据的发展, 比如力量、智力或敏捷这样可增长的角色特性参数。近来最流行的RPG是大型多人在线RPG (也叫做MMORPG), 玩家可以在这种游戏中通过相互间的对抗或合作来使角色成长。由于网站社交化与网页载体的需求, 一些由Flash构建的MMORPG游戏也开始崭露头角。但由于这类游戏通常耗资巨大并且开发周期很长, 游戏厂商在开发时尚且要冒很多的风险, 对于独立开发者来说则更不可行。

1.2.7 驾驶类游戏

顾名思义，这类游戏都会让玩家操控某种交通工具，这些交通工具可能会在陆地上行驶，可能是在水域中航行，或者也可能在天空与太空中飞行。为了获取真实效果，这些游戏通常会以第一人称或第三人称视角来进行。由于系统配置需求以及在Flash中搭建全3D环境的复杂性，多数此类游戏一般都采用二维视角。

1.2.8 桌面式和卡牌式游戏

这种类型的游戏（见图1-5）通常是现实世界此类游戏的数字式呈现，比如国际象棋、国际跳棋、二十一点和德州扑克^①。因为系统配置需求较低，所以Flash平台极为适合创建大多数桌面式和卡牌式游戏，关于这点，网上大量存在的赌场式游戏站点可为之佐证。

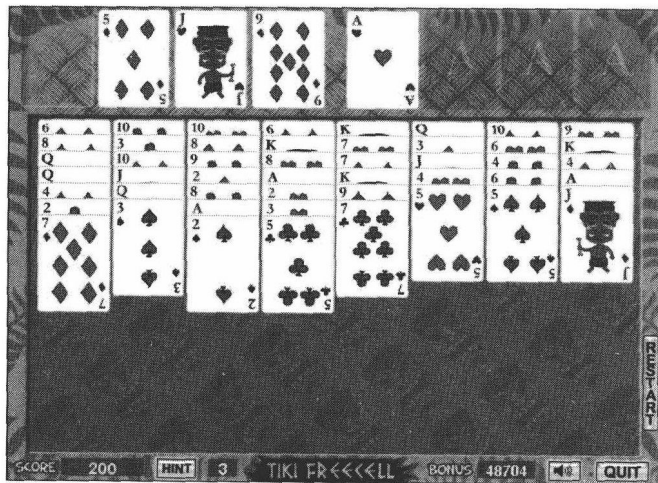


图1-5 《蒂基空当接龙》(Tiki Freecell)。版权归Blockdot公司(www.blockdot.com)所有

① 在美国，但凡言及扑克，即指德州扑克，因其为大众所熟知，故取此意。——译者注

1.3 常用开发术语

计算机科学这个领域很深奥，绝对不适合那些只想做游戏的人。但了解一些基本的核心概念与编程知识却有助于我们以后对游戏的逐层分析。确实，它不仅枯燥，而且偶尔会显得很乏味，但我可以向你保证，理解了这些内容之后，好玩的东西多着呢！

1.3.1 伪码

伪码只不过是使用标准语言对一系列编程步骤所作的解释，有点像是一种逻辑总结。你以后就会看到，在本书的一些例子中，我会在编写实际的ActionScript程序之前先用伪码来分解游戏逻辑。因为伪码容易紧扣编程语法并忽略逻辑缺陷，所以在将问题付诸实际代码之前，我们先用英语来对它们的逻辑进行拆分，这样做几乎总能使问题的处理变得更为简单。我所用的函数和属性的名称大多来源于我写的伪码。

1.3.2 算法

算法就是定义了问题解决方式的一系列指令与判断。它们不是代码或特定语言，因此用直白的英语来表述会较易理解。例如在一个程序里，我们要按照单词的长度来对它们进行排序，算法可能会跟程序的处理过程一样简单。下面就是用伪码写出的算法。

```
for all in wordlist
    sort by length

sort by length (word A, word B)
    if A.length > B.length
        return B
    else
        return A
```

1.3.3 过程式编程

许多早期的编程语言如（Basic或Pascal）都被称为过程式语言。过程式编程可被抽象理解为编写一系列任务（或者叫做子程序）的过程。这些子程序的执行次序是随意的，但所有命令都是由一个主逻辑控制语句来驱动，有时我们也把它叫做主循环语句。本书的范例将会融合过程式编程技术与下面我们将介绍的面向对象编程技术。

1.3.4 面向对象编程

过程式编程关注的是一组待解决的任务，然而面向对象编程（OOP）则专注于“对象”间的交互。OOP是一个非常复杂的课题，你可能很难全面理解它，而现在你只需知道的是：每个对象都是一个独立实体，它能够定义属性，发送并接收从其他对象传递来的消息，以及处理自身内部