

TURING 图灵程序设计丛书



Real-World Flash Game Development
How to Follow Best Practices and Keep Your Sanity

实战Flash游戏开发

[美] Christopher Griffith 著
李鑫 译



人民邮电出版社
POSTS & TELECOM PRESS

图灵程序设计丛书

Real-World Flash Game Development
How to Follow Best Practices and Keep Your Sanity

实战Flash游戏开发

[美] Christopher Griffith 著
李鑫 译

人民邮电出版社
北京

图书在版编目 (C I P) 数据

实战Flash游戏开发 / (美) 格里菲斯
(Griffith, C.) 著 ; 李鑫译. — 北京 : 人民邮电出版
社, 2011. 6

(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
ISBN 978-7-115-25281-4

I. ①实… II. ①格… ②李… III. ①动画制作软件,
Flash②游戏—应用程序—程序设计 IV. ①TP391.41
②G899

中国版本图书馆CIP数据核字(2011)第078339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 CS4 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。

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

◆ 著 [美] Christopher Griffith

译 李 鑫

责任编辑 朱 巍

执行编辑 俞 花

◆ 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>

中国铁道出版社印刷厂印刷

◆ 开本: 800×1000 1/16

印张: 19

字数: 449千字

印数: 1—3 000册

2011年6月第1版

2011年6月北京第1次印刷

著作权合同登记号 图字: 01-2010-6902号

ISBN 978-7-115-25281-4

定价: 55.00 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, 1st Edition by Christopher Griffith, ISBN: 978-0-240-81178-9.

Copyright © 2010 by Elsevier. All rights reserved.

Authorized Simplified Chinese translation edition published by the Proprietor.

ISBN: 978-9-812-72712-1.

Copyright © 2011 by Elsevier (Singapore) Pte Ltd. All rights reserved.

Elsevier (Singapore) Pte Ltd.

3 Killiney Road

#08-01 Winsland House I

Singapore 239519

Tel: (65)6349-0200

Fax: (65)6733-1817

First Published 2011

2011 年初版

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 and Taiwan. 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和台湾地区) 出版与销售。未经许可之出口, 视为违反著作权法, 将受法律之制裁。

致 谢

假如没有妻子 Delayna 和女儿 Miriam 贯穿始终的理解支持与关爱，我根本无法完成本书。世上最好的家庭莫过于此。另外，我还得感谢一直陪伴在我身边的其他家庭成员：

我的父母

Meg 与 Leigha

Andrew、Caleb 及 Carson

Delilah、Kurt、Vanessa、Isaac 及 Virginia

Daniel 与 Kelli

Dottie 与 Charlie

如果没有同事 Jason Fincanon 的支持，我想我永远也想不到自己也能写出一本书来，他可是个天才。

我还要感谢下列人士所提供的专业支持（排名不分先后）。

Josh Dura。

Blockdot 公司的全体同事，特别要感谢的是：Paul Medcalf, Matt Schmulen, Bo Harris, Andrew Richards, Jim Montgomery, Curry McKnight, Stephen Hess, Mike Christian, Jon Stefaniak, Jack Dearnbarger, Dan Ferguson, Jason McMinn, Mike Bielinski, Guy Stables, Matt Bugbee, Andrew Langley, Andy Brooks, Jack Doyle（及其出色的 Tweening Platform），Alessandro Crugnola, Grant Skinner, Allison Emerson 与 Marisa Murphy。

Focal/Elsevier 团队：Laura Lewin, Chris Simpson, Dawnmarie Simpson, Anais Wheeler。

最后，我还要感谢 Adobe 的优秀团队，他们不断推动 Flash 平台向前发展，希望他们继续努力！

引言

游戏开发是一种奇怪的混合物，它融合了很多技术与表现风格。游戏可以说是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，但很多最佳游戏都有一个共同点——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。不同行业的人对游戏的兴趣也可能迥然不同。很少会有一款游戏一直吸引全球各地所有人。

据说互联网上的玩家最喜欢的游戏是那种被称为“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拟真感。这并非是说有些休闲游戏就不具备深度与拟真感，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 20 世纪 70 年代末到 80 年代中期所开发的游戏，在那个年代里，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不只是如何开发游戏，还有如何用 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，这些都会使得你开发的游戏能被尽量多的玩家接触到。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。

如何充分利用本书

本书是为下列两种读者而准备的：对 Flash（CS4 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 CS4 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产生的麻烦。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战场景中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。

本书最后还要介绍一些课题，比如该如何在较大游戏中与团队其他开发者共享资源。另外还将介绍一些符合开发流程的 Flash 优化方法，这将为你正确处理某些问题提供一些灵感。

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，你还可以借用它们来开发自己的作品。网站上所有范例源代码、一些代码片段、脚本以及能够有效地加速游戏开发的工具你都可以随便拿去用。另外，你可以在这个网站上询问一些问题，而且还可能不时地会看到别人对该问题的看法及解释。上去看看吧！

目 录

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

2.5.5 Flash 与传统游戏开发	19
2.6 最佳创作工具	19
第 3 章 一份计划抵得上 1000 片	
阿司匹林	20
3.1 用一两句话来描述游戏	20
3.2 为包含所有游戏界面的游戏流程 列出提纲或绘制线框图	20
3.3 概述游戏所使用的核心机制	21
3.4 列出所需的资源清单	23
3.5 列出游戏所需的技术清单	24
3.6 用 UML 建模工具设计类图 (可选)	26
3.7 规划步骤的快速回顾	27
第 4 章 //太棒了!	28
4.1 适当的提醒	28
部分 1 类	28
4.2 包	29
4.3 类就像文件一样	29
4.4 构造函数	29
4.5 常量、变量与方法	29
4.6 getter 与 setter 方法	31
4.7 类标识符	32
4.8 继承与多态	33
4.9 接口	34
4.10 将 Flash 中的资源与类链接起来	36
4.11 类与基类	37
4.12 使用没有类文件的导出元件	38
4.13 getDefinitionByName 与 Casting	39
部分 2 事件	40
4.14 dispatchEvent 方法	40
4.15 addEventListener 方法、 removeEventListener 方法以及事件 流程中的各阶段	41
4.16 事件的传播与取消	43
4.17 自定义事件	43
部分 3 错误	45
4.18 try、catch、finally	45
4.19 自定义抛出错误	46
部分 4 数据结构与列表	47
4.20 Object	47
4.21 数组	48
4.22 Vector	50
4.23 Dictionary	51
4.24 ByteArray	51
4.25 到底该使用哪种列表	51
4.26 自定义数据结构	52
部分 5 让所有人都能读懂你的注释	52
4.27 底线	53
部分 6 Flash 为什么会这么做	53
4.28 事件流程	53
4.29 帧脚本	54
4.30 使用多个 SWF 文件	56
4.31 内存垃圾回收	57
4.32 本章小结	59
第 5 章 管理资源与使用图像	60
5.1 小议组织库元件	61
5.2 使用图像	61
5.3 常用的光栅图像格式	62
5.3.1 压缩	65
5.3.2 平滑	66
5.3.3 解块	67
5.3.4 外部的图像编辑工具	67
5.4 要点	67
第 6 章 让它动起来: ActionScript 动画	68
6.1 几个术语	68
6.1.1 缓动	69
6.1.2 序列	69
6.2 是否补间? 这是个问题	69
6.3 一个由脚本驱动的单射类游戏	70
6.3.1 Projectile 类	70
6.3.2 SimpleShooter 类	70
6.4 记忆力游戏: 补间动画	72
6.4.1 MemoryCard 类	73
6.4.2 Memory 类	73
6.5 本章小结	76

第 7 章 如雷贯耳：如何使用音频	77	10.2 简要介绍一下弧度与 pi 值	130
7.1 常用音频格式	77	10.3 Flash 中的 3D	133
7.2 常用导出设置	78	13.1.1 位置	134
7.3 使用外部文件	80	13.1.2 旋转	134
7.4 声音编辑工具	80	13.1.3 透视投影	134
7.5 编写音效脚本	81	10.4 简单的 TunnelShooter 范例	135
7.5.1 理解与声音有关的类	81	10.4.1 基本机制	136
7.5.2 SoundEngine 类	82	10.4.2 所需类文件	136
7.5.3 使用 SoundEngine 类	91	10.4.3 Tunnel 类	136
7.5.4 SoundMixer 类	92	部分 2 物理学	145
第 8 章 重塑“视频游戏”	93	10.5 标量	145
8.1 视频编解码器	93	10.6 矢量	146
8.2 外部视频应用：过场与菜单	93	10.7 Vector3D 类	146
8.3 CutsceneManager 类	96	10.8 位移	146
8.4 时间轴上的视频	100	10.9 速度	146
8.4.1 文件尺寸	100	10.10 加速度	147
8.4.2 库：易于使用与杂乱无章	100	10.11 摩擦力	147
8.4.3 性能表现	100	10.12 惯性	147
8.4.4 自由的动态模糊	101	10.13 物理仿真与假象	147
8.5 建立内部视频	101	10.14 真实性与期望值	147
8.6 本章小结	104	10.15 范例：顶视角的驱动引擎	148
第 9 章 XML 与动态内容	105	10.15.1 Vehicle 类	148
9.1 引入数据：理解 URLLoader 类	105	10.15.2 Time 类	149
9.2 XML	105	10.15.3 Game 类	150
9.3 E4X	106	10.16 范例：顶视图且带有漂移效果的 赛车游戏	153
9.4 纵横字谜	106	10.17 本章小结	155
9.4.1 CrosswordTile 类	109	第 11 章 别碰我	156
9.4.2 CrosswordClue 类	113	11.1 能力与需求	156
9.4.3 CrosswordPuzzle 类	114	11.2 hitTestObject——最基本的 检测方法	156
9.5 内容并非只有现成的：纵横字谜 编辑器	122	11.3 hitTestPoint——更先进的 检测方法	157
9.6 数据传送	123	11.4 半径/距离检测法——极适合圆形的 检测方法	161
9.7 示例：XML 与 Flash Vars 的应用对比	124	11.5 矩形边框检测法	162
9.8 本章小结	125	11.5.1 新的 Enemy 类	163
第 10 章 就是一个词儿：数学	126	11.5.2 SimpleShooterCollisions 类所新增的内容	164
部分 1 Math 类	126		
10.1 几何学及三角学	126		

11.5.3 该方法的一些不足之处	165	15.2 数据流程	202
11.6 这些办法都不灵? 那就用鸡尾酒 疗法	167	15.3 游戏流程与功能	202
第 12 章 建筑师——由来已久的梦想	168	15.3.1 游戏背景设定	203
12.1 OOP 概念	168	15.3.2 关卡设计及墙壁的作用	203
12.1.1 封装	168	15.3.3 出口 (portal)	203
12.1.2 继承	169	15.3.4 玩家角色	203
12.1.3 多态性	169	15.3.5 物品	203
12.1.4 接口	169	15.3.6 敌人	204
12.2 游戏开发中的实用 OOP 技巧	170	15.4 关卡文件格式以及资源结构	205
12.3 单例模式: 一种良好的文档模式	171	15.4.1 关卡的 XML 文件	205
12.4 本章小结	173	15.4.2 资源 SWF 文件	207
第 13 章 驿馆闲谈	174	15.4.3 游戏所用类的概述	208
13.1 简单地进行封装——类与容器	174	15.5 引擎类	208
13.2 将有关数值存储到变量与常量中	175	15.5.1 ISprite 接口	209
13.3 不要信任舞台	176	15.5.2 IPlayer 接口	209
13.4 不要使用不理解或不适用的架构 或设计模式	176	15.5.3 IEnemy 接口	210
13.5 要把握好向别人展示自己作品的 最佳时机	177	15.5.4 IItem 接口	210
13.6 本章小结	177	15.5.5 IPortal 接口	211
第 14 章 MixUp——一个简单的游戏 引擎	178	15.5.6 IWall 接口	211
14.1 主文档	180	15.5.7 PlatformerEvent 类	212
14.2 MixUP 类	180	15.5.8 PortalDestinations 类与 PortalRequirement 类	212
14.3 Title 类	182	15.5.9 PlatformerConfig 类	213
14.4 RulesPanel 类	183	15.5.10 GridReference 类	213
14.5 Game 类	184	15.5.11 CollisionGrid 类	215
14.6 接口	186	15.5.12 PlatformerEngine 类	216
14.7 GameBoard 类	188	15.6 游戏实现类	230
14.8 SourceImageEmbedded 类	194	15.7 资源类	232
14.9 GameHistory 类与 Results 类	196	15.7.1 Player 类	233
14.10 SourceImageCamera 类	198	15.7.2 Enemy 类	234
14.11 本章小结	200	15.7.3 Item 类	235
第 15 章 技术大融合: 制作平台游戏	201	15.7.4 Portal 类与 Wall 类	236
15.1 平台类型游戏	201	15.7.5 游戏中的资源	237
		15.8 本章小结	239
		第 16 章 独乐不如众乐: 开发多人游戏	240
		16.1 RTMFP	240
		16.2 Stratus	240
		16.3 MixUp 多人游戏	241
		16.3.1 游戏所用的类	241

16.3.2	SourceImageWebCamera 类	242	18.1.2	为所要接受的输入设置 最低延时	264
16.3.3	ConnectionPanel 类	245	18.1.3	一经发现恶意使用就将 游戏关闭	264
16.3.4	Title 类	246	18.2	数据保护	264
16.3.5	Results 类	247	18.2.1	内存攻击	264
16.3.6	MixUp 类	248	18.2.2	数据的散列加密	265
16.4	本章小结	250	18.2.3	拆解数据	267
第 17 章	找到就干掉：猎虫行动	251	18.2.4	插入误导性数据	267
17.1	Bug	251	18.2.5	保护要发送与接收的数据	268
17.1.1	trace	251	18.2.6	保护传输数据安全 的散列加密法	268
17.1.2	FlashTracer	253	18.2.7	矩阵加密法	268
17.1.3	调试器	254	18.3	SWF 文件保护	269
17.2	性能与优化	255	18.4	本章小结	269
17.2.1	FrameRateProfiler 类	256	附录 A	摄像头与麦克风	270
17.2.2	MemoryProfiler 类	258	附录 B	本地化	276
17.2.3	Sample 包	259	附录 C	JSFL——情人们的 JavaScript	283
17.3	本章小结	262	后记		289
第 18 章	当心恶贼	263			
18.1	恶意使用	263			
18.1.1	关闭不再需要的事件侦听器	263			

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是很重要的,这样我们就能理解在本书其余部分中所使用的术语了。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,那就请重新翻开本章。要是你对所有这些长单词与抽象概念有些不知所措,也请不要担心!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(有些自身无法被简单归类的游戏甚至还以此为荣),但多数游戏却可被归为下列类别之一。

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,并且一般会有一个或多个主角。从玩家感觉上来说,玩这些游戏和看电影最为相像(有些游戏据说已经快要拍成电影了),它们大量依靠剧情对白、搜寻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 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基本功,并且通常对系统配置的要求也较低,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)可能会涉及许多游戏玩法以及子类别,但就一般而言,动作类游戏考验的是一个玩家的机敏程度、反应时间以及在面临压力时头脑是否灵活。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动作类游戏。Flash 平台可以很好地支持动作类游戏的一些子类别,特别是那种年代较早的动作类游戏,比如像《太空入侵者》或《超级马里奥兄弟》

那样的游戏。

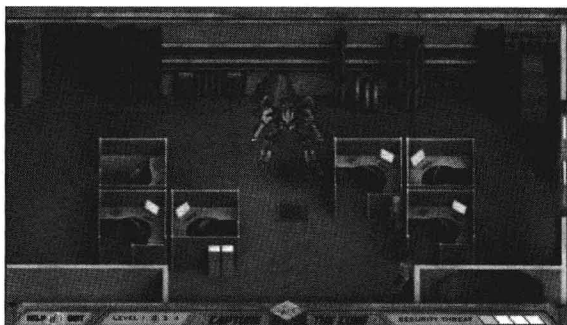


图 1-1 冒险游戏《自动售货机变形金刚》(Mountain Dew Capture the Cube)

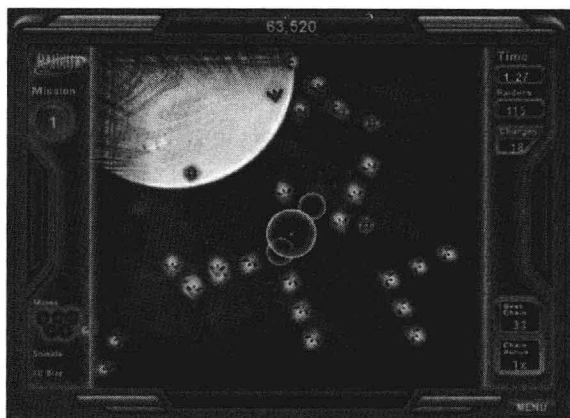


图 1-2 动作类游戏《辐射光环》(Raidiux)。Blockdot 公司版权所有

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所需要的美术工作一般较少, 这意味着独立开发者经常靠自己就可以将其制作完成。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, 所以我将它们划为独立的体系。寻找单词、纵横拼字谜、拼字游戏以及回文构词^①, 都属于此类游戏的范畴(见

^① Anagram, 用一组随机选出的字母构成新单词。——译者注

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。



图 1-3 JinkyPOP。Blockdot 公司版权所有

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、资源管理以及决策制定,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程度来界定。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(比如像用 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特例。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》。Blockdot 公司版权所有

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描述得更多。传统上, RPG 游戏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进展, 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 游戏是大型多人在线 RPG 游戏 (也叫作 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。由于网站社会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更多的风险, 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 中搭建全三维 (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 (见图 1-5)。^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。

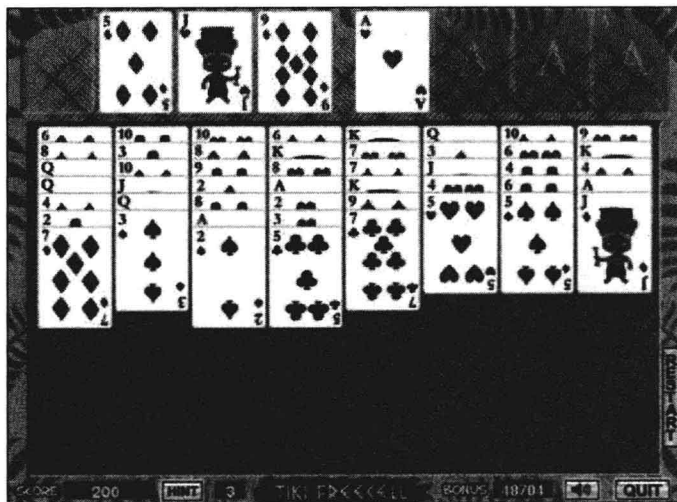


图 1-5 《蒂基空当接龙》(Tiki Freecell)。Blockdot 公司版权所有

^① 在美国, 但凡言及扑克, 即指德州扑克, 盖其为大众所熟知, 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) 都被称作过程式语言。过程式编程可被抽象理解为编写一系列任务（或者叫作子程序）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，有时我们也把它叫作主循环语句。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，但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，发送与接收从其他对象传递来的消息，以及处理自身内部逻辑。例如，在 OOP 中，一个人可以是一个对象，而他的一个朋友则是另一个对象。他们共享一些组件，他们都是人，但也都有各自唯一的特征。而且他们彼此间会使用一种通用语言来传递消息进行交流。ActionScript 的有些方面是以 OOP 的方式运作的，我们稍后会在书中予以详述。