



游戏开发人员书库

释放如潮水般的创造性

游戏设计

——原理与实践

Game Design: Theory and Practice



[美] Richard Rouse III 著
尤晓东 等译



电子工业出版社
Publishing House of Electronics Industry
<http://www.phei.com.cn>



游戏开发人员书库

游戏设计——原理与实践

计算机游戏一个重要但往往被忽视的因素是游戏的可玩性，即使得游戏能吸引人并具有娱乐性。本书就这一难以捉摸的主题进行了探讨。作者详细探讨了一些关键的游戏设计主题，包括游戏结构设计、故事情节设计、玩家动机设计、输入/输出设计、人工智能设计以及测试设计等。本书涵盖了交互式游戏的各个方面，包括计算机、控制台和街机，并涉及了许多游戏的类别，如策略游戏、冒险游戏、模拟游戏、动作游戏、角色扮演游戏、体育游戏和战争游戏等。

本书特色：

- ※ 给出了整个游戏的开发过程
- ※ 通过与一些专家的会谈，了解顶级游戏设计人员的技术
- ※ 通过对几个游戏的详细分析，理解使游戏变得成功的因素
- ※ 找到更有效地使游戏想法成型的方法，包括焦点的使用、文档的设计、故事的提炼、脚本的和技术规范的编写等。附录中一个完整的设计文档示例了优秀游戏开发文档的原理

释放如潮水般的创造性

游戏开发人员书库

- 3ds max游戏角色设计与建模
- **游戏设计——原理与实践**

ISBN 7-5053-9212-3



9 787505 392120 >



责任编辑：奕 昊
特约编辑：赵宏英
封面设计：毛惠庚

本书贴有激光防伪标志，凡没有防伪标志者，属盗版图书

ISBN 7-5053-9212-3/TP · 5325 定价：48.00 元

www.pdf365.com 免费pdf 电子书分享

游戏开发人员书库

游戏设计——原理与实践

Game Design: Theory and Practice

[美] Richard Rouse III 著

尤晓东 等译



B1282252

电子工业出版社

Publishing House of Electronics Industry

北京·Beijing

内 容 简 介

“迸发如潮水般的创作激情”，这就是本书要告诉读者、游戏设计者、游戏开发商和游戏爱好者的东西。刚看到这本游戏书时并未特别兴奋，甚至因为它一开始便声明本书并不是介绍游戏编程的而有些许担心，它的内容好吗？实用吗？还没有看完一半，这种顾虑便完全打消了，甚至是被书籍内容深深地吸引住了。本书由多位游戏开发名家共同撰写而成，其中有几位作者的作品都是游戏界的经典之作。书中以一个个经典的游戏开发实例开始，介绍其开发的背景、工具、挫折、开发过程、市场营销方案，让我们深入了解了游戏开发过程背后许许多多的情况，接着便是游戏相关开发者的访谈，诙谐活泼的语言揭示了游戏开发者的理念、风格、结构设计、故事情节设计、脚本设计、动画片头设计、人工智能（AI）设计以及文档资料设计等等，字里行间满溢着游戏开发设计工作中众多的经验和技巧，开发者的欢乐、遗憾、成功和喜悦生动地流淌在纸面之上。

对于游戏的开发者、游戏玩家，这样的游戏设计书籍是非常难得的佳品！

Authorized translation from the English language edition published by Wordware Publishing, Inc. Copyright © 2001. 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 the Publisher.

Simplified Chinese language edition published by Publishing House of Electronics Industry, Copyright © 2003.

本书中文简体版专有翻译出版权由 Wordware Publishing, Inc. 授予电子工业出版社。其原文版权及中文翻译出版权受法律保护。未经许可，不得以任何形式或手段复制或抄袭本书内容。

版权贸易合同登记号：图字：01-2001-4596

图书在版编目（CIP）数据

游戏设计：原理与实践 /（美）劳斯三世（Rouse III, R.）著；尤晓东等译。—北京：电子工业出版社，2003.10
（游戏开发人员书库）

书名原文：Game Design: Theory and Practice

ISBN 7-5053-9212-3

I. 游... II. ①劳... ②尤... III. 游戏-应用程序-程序设计 IV. G899

中国版本图书馆CIP数据核字（2003）第088901号

责任编辑：婁 昊 特约编辑：赵宏英

印刷者：北京兴华印刷厂

出版发行：电子工业出版社 <http://www.phei.com.cn>

北京市海淀区万寿路173信箱 邮编：100036

经 销：各地新华书店

开 本：787 × 1092 1/16 印张：28.5 字数：712千字

版 次：2003年10月第1版 2003年10月第1次印刷

定 价：48.00元

凡购买电子工业出版社的图书，如有缺损问题，请向购买书店调换；若书店售缺，请与本社发行部联系。

联系电话：（010）68279077。质量投诉请发邮件至 zltz@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

译者序

相信大多数电脑用户都玩过电脑游戏,有些只是简单地在学习和工作之余玩一下系统自带的几个简单游戏,有些则是资深的大型游戏玩家。

但是,作为游戏玩家,你知道这些游戏是怎么设计出来的吗?在设计游戏时主要考虑哪些因素?有哪些困难点?如何吸引玩家?由资深电脑游戏设计师和程序员兼作家Richard Rouse III精心编著的本书,系统地回答了上述问题。

在本书中,作者详细讲述了游戏设计内容的关键,包括游戏平衡、游戏的故事情节、非线性故障、游戏者的促动因素、输入/输出、人工智能、设计水平和游戏测试等等。此书全面论述了游戏间的相互作用,包括计算机、操纵台、弓形字幕,以及各种各样的游戏种类,包括战略、探险、模拟动作、游戏角色、运动和战争游戏等。

本书引导读者系统地学习游戏设计的理论和制作出最好游戏的知识,读者还可以在与电脑游戏行业最著名的六位设计师的谈话中学习关于游戏设计的知识。

为了向国内读者介绍国际上最新的游戏设计理论与最好的实践,电子工业出版社引进出版了本书,相信会受到读者的欢迎。

参加本书翻译、审校工作的人员有曹晟、陈红梅、高崇、高青、顾莉、顾铁成、管永川、黄晓霜、姜昊、李朝晖、李欣、龙劲松、商广红、沈金河、林满山、宋雪滢、王培培、王松、吴巧泉、杨春华、尹刚、尤晓东、张素霞、周靖、周雨红、朱崇高、朱晓明等。

由于水平和时间所限,相信其中仍有不少错误,敬请读者指正。

About the Author

作者简介

Richard Rouse III 是 Surreal Software 公司 (www.surreal.com) 的一名电脑游戏设计师、程序员和作家。Rouse 在 7 年的职业生涯中一直在从事专业的游戏设计, 在 PC、Macintosh、Sega Dreamcast、Sony PlayStation 和 PlayStation 2 等平台的游戏开发过程的设计工作中充分发挥了其游戏创作的天赋。他曾参与了 "Centipede 3D"、"Odyssey: The Legend of Nemesis and Damage Incorporated" 这些游戏的制作工作。在 Surreal 公司, 他将主要的精力都投入了一度非常保密的 PlayStation 2 的动作/冒险游戏项目中, 而同时还参与了 PlayStation 2 平台上 "Drakan" 的开发工作。Rouse 还在一些已出版的畅销书中负责编写了游戏设计方面的内容, 其中包括 "Game Developer"、"SIGGRAPH Computer Graphics"、"Gamasutra" 和 "Inside Mac Games"。

Your Feedback

联系方式

欢迎你提出关于本书的更正、意见或者简单的评价。可以将这些反馈信息通过电子邮件发送给本书的作者: rr3@paranoidproductions.com。在本书的网站上也可以找到本书的勘误信息、更新和其他读者感兴趣的东西, 网址是 www.paranoidproductions.com。

About the Artist

本书的美术师

Steve Ogden 曾是一个美术师、插图画家和卡通漫画家, 他从事这方面的创作近 20 余年, 而且不可思议的是, 他的灵感好像从来就没有中断过。在数字化领域的项目中, 他曾参与过 Bally's Game Magic 的娱乐场游戏项目的开发, 而且还参与过 Centipede 3D 的项目, 最近他刚刚完成了 Cyan 公司的 realIMYST 的项目, 并在项目中担任美术设计督导和产品负责人 (当在几个小时之内完成本书中的插图后, 他可能就去睡觉了)。如果 Cyan 公司的人能够找到他并让他安心地在办公桌前工作, 他也可能在忙碌于 Cyan 的下一个游戏。如果想了解他的更多作品, 请浏览 www.lunaentertainment.com, 其中的种类包括 2D 和 3D 游戏。可以通过 E-mail 写信给他: ogden@lunaentertainment.com。他现在可能在远处的海滩上睡觉。

Introduction

前言

我常回想起我最早游戏的经历。记得，我在家乡的一个墨西哥小餐馆里偶然发现了 *Space Invaders*，那时候我好像只有 6 岁，对我来说，*Space Invaders* 可以说是我见过的最棒的东西了，至少可以和 *LegoLand* 相媲美。我以前曾经听说过“街机”游戏，但那是我第一次玩。我知道 *Space Invaders* 比电视游戏更好玩，因为我可以使用游戏杆控制屏幕底部的小飞船，我自己可以向外星人射击，而不再是看着别人玩。我被那个游戏深深地迷住了。而且可笑的是，在那个时候，我却不知道自己需要在游戏中坚持数十分钟才能获得胜利。游戏是以“attract”模式（招引购买者的演示方式——编辑注）运行的，就和其他的“街机”游戏一样，当时幼稚的我还以为自己使用游戏杆正在控制游戏，而实际上我什么都没控制。但是那次经历却使我非常兴奋。

本书是关于如何开发原创电脑游戏的，其中的内容会对玩家所产生的影响，就如同“*Space Invaders*”对我的影响一样。本书从游戏设计人员的角度讲述了游戏开发的过程。虽然有许多书籍都是关于电脑游戏编程的，但当我成为一个满怀热情的电脑设计人员的时候，我可以回忆起当初无法找到一本关于游戏设计的书有多么让人沮丧。从某种意义上讲，我是为我自己编写这本书——为 10 年前的自己编写本书。从我谦虚的观点来看，游戏设计者在电脑游戏的制作过程中充当最有意义的“角色”。游戏的设计可以表达出游戏可玩性的形式和模型，这也是我们这种艺术媒体区别于其他媒体的一个重要因素。



什么是“游戏可玩性”

你可能会问：“游戏可玩性是什么呢？”许多人都认为自己知道什么是游戏可玩性，而实际上，这个概念有许多不同的、合理的定义。我愿意为这个概念下如下的定义，它可以囊括本书中可以找到的所有术语的用法。游戏可玩性是电脑游戏的组成部分之一，该部分是无法在其他的艺术形式中找到的——交互性。游戏的可玩性是游戏所包含的交互性的程度及特点，也就是说，它所指的就是玩家与游戏世界及游戏世界对玩家的选择做出反应的方式。在 *Centipede* 这类动作类游戏中，游戏可玩性就是在屏幕的底部移动飞船，并且无情地向敌人射击；在 *SimCity* 中，游戏可玩性就是建造一个城市，然后观察居住在城市中的居民；在 *Doom* 中，游戏可玩性就是在 3D 的环境中高速移动，向敌人射击，并沿途收集开门所需的钥匙；在 *San Francisco Rush* 中，游戏可玩性就是沿着奇怪的道路驾驶一辆汽车，并与其他赛车手争抢位置；在 *StarCraft* 中，游戏可玩性就是在地图上操纵那些军事单位，找到资源然后进行开发，壮大实力，最后与一个

强大的敌人进行战斗；在 Civilization 中，游戏可玩性就是探索世界，从头开始建立一个社会，开发新的科学技术，并且与世界上其他地区的居民交流。



关于游戏设计

那么什么是游戏设计呢？实际上，前面谈到游戏可玩性的时候已经定义过了这个概念，游戏设计的概念是很容易解释的：游戏设计就是定义游戏可玩性的内容。游戏设计决定了玩家可以在游戏世界中做出什么选择，并且这些选择会在游戏的其余部分造成哪些不同的结果。游戏设计决定了游戏中胜利或失败的标准、用户可以控制游戏的方式、决定了游戏可以向玩家所传达的信息，并且可以决定游戏的难度。简单地说，游戏设计决定了游戏可玩性实现过程中的每个细节。



谁是游戏设计人员

此时，大家应该很清楚游戏设计人员的任务了：他要通过制作游戏的设计方案来决定游戏可玩性的特点。“游戏设计人员”和“游戏设计”的术语长久以来就在很多环境中使用，因此这些术语的含义已经变得模糊不清，而且难于限制其使用范围。有些人似乎会认为游戏设计是游戏开发的同义词。这些人将从事电脑游戏开发的所有人员，无论是美工、程序员还是制作者，统统都称为游戏设计人员。我更倾向于使用一个明确的定义，也就是我前面曾概括的定义：游戏设计人员就是设计游戏的人，他们确定游戏可玩性的模型和特点。

游戏设计人员参与的任务是很值得注意的。游戏设计人员可能会提出某些概念方案，或者制作一些在游戏中使用的美术作品，但是他却不必实际完成这些工作。游戏设计人员可以去写游戏中所有人物的对话脚本，但他不必直接实现脚本。游戏设计人员可以参与到编程工作中，甚至可以担任编程负责人的职务，但他不一定参与编程至少不负责全部程序。游戏设计人员可以设计部分或者全部的游戏世界，制作游戏的关卡（如果这些项目中需要制作关卡的话），但是他不必实际实现关卡。从管理和生产的角度来讲，游戏设计人员是项目管理者，严密关注制作队伍中的成员，看他们是否可以高效地完成其工作，但他不一定实际做每项工作。游戏设计人员所需要做的所有事情就是，确定游戏可玩性的形式。实际上，许多游戏设计人员在项目中都要完成许多人物，但是他们所要关心的焦点应该是——游戏的整体和可玩性。



本书的主要内容

本书包含了关于游戏设计的大量信息，所涉及的范围十分广泛。当然，没有哪一本书可以成为某种艺术形式的权威著作。因此本书不是完整的电脑游戏编程方面的书籍。有很多书籍介绍游戏编程，正如我在本书的后面部分所介绍的那样，掌握编程方法对游戏设计很有帮助。然

而，这种技能却并不是游戏设计的必要组成部分；许多优秀的游戏设计人员根本不知道如何编程。

本书中的各个章节可以分为三个大类。第一类是 12 个核心章节，其中讨论了电脑游戏开发的各个方面，内容涉及建立游戏的焦点、编写游戏设计的文档，确定故事介绍的游戏模式，以及对即将完成的产品进行测试。这些章节介绍了游戏设计所依据的理论，以及设计人员为了制作最好的游戏所要掌握的内容。这些章节中还包括游戏过程中可能出现的问题的讨论，使用了一些来自我经验中的例子，并运用到实际的游戏设计过程中。

本书中有 5 章分析性的内容，分别介绍了 5 种风格不同的、但都非常优秀的 5 个游戏。其中游戏设计人员必须掌握的最重要的技能就是，分析他所喜欢的游戏，以便能够更好地理解那些游戏。通过理解这些游戏，设计人员可以将那些优点运用到自己的项目中。但这并不是暗示说优秀的游戏设计人员仅仅是去抄袭其他游戏设计人员的作品。了解其他游戏取得成功的原因，可以帮助这些设计人员在游戏设计的总体上有更全面的理解。每个设计人员都应该拿到那些他们认为最好的游戏，然后思考是哪些因素使该游戏获得了成功。我在本书中所介绍过的游戏——Centipede、Tetris、Loom、Myth: The Fallen Lords 和 The Sims 等都是些非常独特的游戏。虽然你要制作的项目可能与这些游戏并不相似，但是对这些游戏加以分析可以学到许多东西。据称，射击游戏的第一发起人是在看过了冒险类游戏之后，才成功地设计出了他们自己的游戏风格。当然，角色扮演类游戏的设计人员也从“街机”游戏设计人员那里学到了许多东西。如果要在自己制作的项目中改进风格，并且制作出真正原创的游戏，其最佳方式就是从其他风格的游戏借鉴一些技术。

本书中还加入了一组访谈录，采访对象是在这个行业短暂的历史中最为受人尊重的 6 位游戏设计大师，他们都曾设计了最为出色的游戏。这些访谈内容丰富，比那些可以在 Internet 上或杂志上找到的宣传资料式的采访内容要丰富得多。在每次采访中，所讨论的主题都是这位设计师职业生涯中最好的作品，并且谈论了为什么他们相信这些游戏会取得成功。这些设计师们还谈到了在开发游戏中自己所使用的技术。在从事游戏开发的工作中，我发现，与其他的电脑游戏设计人员进行交谈非常有帮助，从中可以获取很多对游戏进行完善的知识。从这些章节中可以得到很多颇具价值的信息，无论你在游戏设计方面有多少经验，其中的信息都会有所帮助。

在本书的末尾有一个术语表。作为游戏设计的术语表，它并不很完整，但其中却包含了我在本书中使用的许多更为深奥的词汇，比如说我自己比较喜欢使用的一个词“替身”。每个游戏设计人员都会使用一些行话来指代自己所制作游戏的各个方面，并且各个设计人员之间使用的行话也不尽相同。例如，行话可以更透彻地解释游戏可玩性和游戏的控制方式。此外，如果读者发现本书中的内容包含了很多的知识，就会发现这个术语表对于了解实时游戏的定义以及两种不同的 RTS 之间的区别会有很大帮助。而关于游戏设计的讨论也会出现降格到区分语意的问题，因为当设计人员提到游戏“引擎”的时候，他们可能指的并不是同一个概念。我希望使用词汇表可以帮助读者在阅读本书的时候避免出现这类问题。



本书的读者对象

对于希望了解电脑游戏开发过程的读者，本书是为你写的。显然，读者的范围并不是严格

地从游戏设计的角度来划分的。前面我曾经说过，有许多关于游戏编程方面的书籍，也有介绍如何使用 Photoshop 和 3DS Max 编写游戏的书籍。本书不会介绍这些内容，而会集中讨论关于游戏设计的话题，重点讨论如何使你所制作的游戏具有更好的可玩性。虽然编程和美工对游戏的成功来说至关重要，但是无论有多少漂亮的图形或者先进的代码，都无法弥补毫无生气的设计方案。

我在编写本书时包含了各种类型和规模的游戏项目。无论你所制作的游戏是为了商业产品，还是希望有一天可以作为共享软件来发行，或者只是制作一个自己和朋友可以玩的游戏，本书对上述过程中的游戏设计都是很有帮助的。此外，对于读者到底是在一个大型开发团队中制作游戏，或者是与几个同伴协作开发游戏，甚至完全是自己制作游戏却是无关紧要的。在本书中，我经常提到几类项目的参与人员。当我提到“编程人员”时，可能指的是一个具有 10 位经验丰富的编程人员的工作组，具有大量的资金，并可以使用最先进的实时 3D 技术；也可能只是在指你，你要自己编写游戏中每个部分的代码。当我提到“游戏测试人员”时，我可能指的是由 15 个人组成的经验丰富且非常专业的测试小组，他们能够对你制作的游戏进行全面测试；也可能指的是你的朋友 Bob 和 Judith，他们就像你一样喜欢游戏，也非常愿意玩游戏。优秀的游戏当然并不都是由大型团队开发制作出来的。即使是在今天，当数百万美元的开发预算已经是司空见惯的事情时，最好的游戏也有可能是某个人自己想出来的，而且他也不需要置身于庞大的开发队伍之中，更不需要观看或了解游戏制作的全程。

在本书的许多地方都会提到，设计人员在游戏项目中的领导作用。当然，并不是每个设计人员都在其项目中从事领导的工作。即便是游戏设计的负责人，也会经常发现自己在游戏设计中并没有最终的决定权。在这个问题上，本书考虑的是一种理想模型。无论你在项目的工作中具有多大权力，最重要的问题就是，如果你具有决策权，就要确定好自己在项目中必须完成的目标。然后再与团队中的其他人共同朝着这个目标努力。如果颇具说服他人的能力，并且认为自己是正确的，那么便有了一个很好的机会来说服团队成员，以你认为正确的方式来工作。项目的领导者并不一定是那些资历最深或者头衔最高的人；项目真正的领导者应是那些在工作中“崭露头角”的人，是那些关注项目并投身其中的人，是那些愿意为了开发出最好的游戏而倾注大量时间和精力的人。



理论与实践

每一种媒体都有其独特的语言和表达方式，相应媒体的用户要负责去发现这种方式。我确信，电脑游戏的表达方式和其他媒体的表达方式都一样有力。与印刷品、音乐、视觉艺术或者戏剧相比，电脑游戏是相对新颖的艺术形式，这对找到电脑游戏最强有力的表现方式造成了障碍。本书就是要帮助读者在他们自己的项目中找到合式的表达方式。在这个过程中可能会遇到比较理论性的问题——为什么玩家喜欢游戏，而且也会遇到非常实际的问题——如何有效地与测试人员共事。如果想要制作一款非常棒的游戏，那么游戏设计人员就必须了解游戏整个设计过程中可能涉及的理论 and 实际方面的问题。

Contents

目 录

第 1 章 玩家的需要	1
1.1 人们为什么玩游戏.....	2
1.1.1 玩家需要挑战.....	2
1.1.2 玩家需要交流.....	2
1.1.3 玩家需要独处的经历.....	4
1.1.4 玩家需要炫耀的权利.....	4
1.1.5 玩家需要情感体验.....	4
1.1.6 玩家需要幻想.....	5
1.2 玩家的期望.....	6
1.2.1 玩家期望一致的世界.....	6
1.2.2 玩家期望了解游戏环境的界限.....	6
1.2.3 玩家期望合理的游戏解决方案.....	7
1.2.4 玩家期望指导.....	8
1.2.5 玩家期望逐步完成游戏.....	9
1.2.6 玩家期望能沉浸到游戏中.....	9
1.2.7 玩家也期望“失败”.....	10
1.2.8 玩家期望公平的机会.....	11
1.2.9 玩家不期望重复自己.....	11
1.2.10 玩家期望没有令人绝望的障碍.....	12
1.2.11 玩家期望玩，而不是看.....	13
1.2.12 玩家不知道他们要什么，但是他们知道什么是他们要的.....	13
1.3 一个无截止符的清单.....	14
第 2 章 Sid Meier 访谈录	15
第 3 章 游戏构思：游戏、技术和故事	34
3.1 游戏设计的起点.....	35

3.1.1	从游戏入手	35
3.1.2	从技术入手	36
3.1.3	从故事入手	38
3.2	制约因素	40
3.2.1	Odyssey: The Legend of Nemesis	40
3.2.2	Damage Incorporated	41
3.2.3	Centipede 3D	43
3.3	制约因素的利用	44
3.3.1	技术环境	44
3.3.2	蘑菇的启示	45
3.3.3	时间限制	46
3.4	不选择也是一种选择	46
第 4 章	游戏分析: Centipede	48
4.1	经典街机游戏的特点	50
4.2	输入	52
4.3	互联性	53
4.4	不断增强的紧张局势	55
4.5	人不同, 游戏不同	57
第 5 章	焦点	58
5.1	建立焦点	59
5.1.1	一个例子: Snow Carnage Derby	60
5.1.2	焦点的作用	62
5.2	坚持焦点	64
5.2.1	生动化焦点	65
5.2.2	改变焦点	66
5.3	补充焦点	68
5.4	使用焦点	70
第 6 章	Ed Logg 访谈录	72
第 7 章	游戏可玩性的元素	98
7.1	独特的解决方法	99
7.1.1	预想相对复杂的系统	99
7.1.2	显露	100
7.2	非线性	101
7.2.1	非线性的种类	101

7.2.2 执行.....	102
7.2.3 非线性的目的.....	104
7.3 虚拟现实.....	104
7.4 指导玩家.....	106
7.4.1 奖励.....	107
7.5 输入/输出.....	109
7.5.1 控制和输入.....	109
7.5.2 输出和游戏世界的反馈.....	112
7.6 基本的元素.....	115
第 8 章 游戏分析: Tetris.....	116
8.1 拼图游戏还是动作游戏.....	117
8.2 作为经典街机游戏的 Tetris.....	118
8.3 技术.....	120
8.4 人工智能.....	121
8.5 紧张程度升级.....	122
8.6 简单与对称.....	123
8.7 晚出十年, 谁会发行它.....	124
第 9 章 人工智能.....	125
9.1 游戏中 AI 的目标.....	127
9.1.1 向玩家提出挑战.....	127
9.1.2 不要制作迟钝的东西.....	129
9.1.3 不可预测.....	130
9.1.4 有助于讲述故事.....	132
9.1.5 创造一个逼真的世界.....	134
9.2 不平衡的游戏场地.....	134
9.2.1 要多真实才算真实.....	135
9.3 AI 主体及其环境.....	136
9.4 要多好才算好.....	137
9.5 脚本化.....	139
9.6 人工加入的低能.....	139
第 10 章 Steve Meretzky 访谈录.....	140
第 11 章 故事介绍.....	167
11.1 设计人员的故事与玩家的故事.....	168
11.2 故事介绍的位置.....	170

113529/10

11.2.1	游戏以外	171
11.2.2	在游戏中	174
11.2.3	外部材料	176
11.3	沮丧的线性作家	176
11.4	游戏故事	178
11.4.1	非线性	179
11.4.2	与游戏相结合	179
11.5	梦想	180
第 12 章	游戏分析: Loom	182
12.1	注重游戏机制	183
12.2	用户界面	185
12.3	草稿系统	186
12.4	难度	187
12.5	故事	188
12.6	Loom 是一个冒险类游戏	190
第 13 章	实现设计方案	192
13.1	灵活的过程	194
13.1.1	太多太早	194
13.1.2	保持简单	196
13.2	制作游戏	196
13.2.1	核心技术	196
13.2.2	渐进的步骤	197
13.2.3	完整的功能范围	198
13.2.4	经历多次修改	199
13.3	编程	200
13.4	什么时候会有趣	201
第 14 章	Chris Crawford 访谈录	203
第 15 章	游戏开发文档	226
15.1	为游戏编写文档	227
15.1.1	概念 (Concept) 文档或 Pitch 文档或建议	227
15.1.2	设计文档	227
15.1.3	流程图	228
15.1.4	背景故事	229
15.1.5	原稿	230

15.1.6	背景艺术	232
15.1.7	故事剪辑	232
15.1.8	技术设计文档	232
15.1.9	计划和商业/市场文档	233
15.1.10	非标准文档	233
15.2	提供文档的好处	233
第 16 章	游戏分析: Myth: The Fallen Lords	235
16.1	技术的使用	236
16.2	游戏重点	239
16.3	剧情描述	240
16.4	专业级游戏	241
16.5	多人游戏	242
16.6	整体表现	243
第 17 章	设计文档	245
17.1	写作风格	246
17.2	单元划分	248
17.2.1	目录表	248
17.2.2	概述/简介或操作概要	249
17.2.3	游戏机制	249
17.2.4	人工智能	254
17.2.5	游戏元素: 角色、物品和对象/机制	256
17.2.6	故事简介	258
17.2.7	游戏进程	258
17.2.8	系统菜单	259
17.3	个人观点	259
17.4	糟糕的设计文档	260
17.4.1	薄本或省略符号文档	260
17.4.2	背景故事书	261
17.4.3	杀伤力过度文档	261
17.4.4	空中楼阁式的文档	262
17.4.5	石化文档	263
17.5	重量问题	263
17.5.1	让别人读它	263
17.6	文档只是个开始	264

第 18 章 Jordan Mechner 访谈录	265
第 19 章 构思设计工具	291
19.1 所希望得到的功能性.....	292
19.1.1 关卡形象化.....	293
19.1.2 大画面.....	295
19.1.3 跳转至游戏.....	296
19.1.4 编辑游戏世界.....	297
19.2 脚本语言和对象行为.....	299
19.3 我们和他们.....	301
19.3.1 最好的愿望.....	302
19.4 万能的游戏编辑器.....	303
第 20 章 游戏分析: The Sims	305
20.1 放弃创作权.....	306
20.2 熟悉的主题.....	307
20.3 安全的试验.....	308
20.4 深度和焦点.....	309
20.5 界面.....	310
20.6 受控行为和自主行为.....	311
20.7 要吸取的教训.....	312
第 21 章 关卡设计	314
21.1 各种游戏中的关卡介绍.....	315
21.1.1 关卡划分.....	316
21.1.2 关卡顺序.....	317
21.2 关卡的组成成分.....	319
21.2.1 动作类.....	320
21.2.2 探险.....	320
21.2.3 解谜.....	321
21.2.4 剧情叙述.....	322
21.2.5 美工.....	323
21.2.6 平衡所有的元素.....	324
21.3 关卡流程.....	324
21.4 优秀关卡的元章.....	326
21.4.1 不要卡住玩家.....	326
21.4.2 设定子目标.....	327
21.4.3 路标.....	328

21.4.4	主线	328
21.4.5	减少回头路	328
21.4.6	首次通关的条件	328
21.4.7	清晰标注导向区域	329
21.4.8	提供多种选择	329
21.4.9	个人规则	329
21.5	设计步骤	330
第1步	预备工作	330
第2步	描绘草图	331
第3步	基本建筑	331
第4步	细化建筑结构,直到关卡令人感到有趣为止	332
第5步	基本游戏可玩性	332
第6步	优化游戏可玩性,直到游戏比较有趣为止	333
第7步	细化美工	334
第8步	游戏测试	334
21.5.1	开发步骤的各种变通方法	335
21.6	关卡设计人员配备方法	335
21.7	协作开发	336
第22章	Will Wright 访谈录	337
第23章	游戏测试	369
23.1	找到合适的测试人员	370
23.1.1	应该找哪种测试人员	371
23.1.2	不应该找哪种测试人员	373
23.2	何时进行测试	374
23.3	如何进行测试	375
23.4	有指导的测试和无指导的测试	377
23.5	谐调	377
23.5.1	游戏难度过高	378
23.6	艺术观点	380
总结		382
附录	设计文件示例:“Atomic Sam”	385
1	概述	389
2	游戏机制	390
2.1	概述	390

2.2	摄影机	391
2.3	插入式图形用户界面	391
2.4	重新开始和保存游戏	392
2.5	控制方法简介	392
2.6	运动方向	393
2.7	飞行运动	393
2.8	地形	395
2.9	捡起物品	396
2.10	扔出投射物	396
2.11	电能比拉鱼	398
2.12	动作	398
2.13	互动战斗环境	399
2.14	观察	400
2.15	朋友	400
2.16	对话	401
2.17	片头动画	402
2.18	游戏故事	402
2.19	关卡	403
3	人工智能	404
3.1	敌人 AI	404
3.2	玩家探测能力	405
3.3	运动	405
3.4	飞行	405
3.5	目标寻找能力	405
3.6	打击承受能力	405
3.7	攻击能力	406
3.8	躲闪能力	406
3.9	特殊动作	406
3.10	奚落	407
3.11	圈套设置	407
3.12	非战斗主体	407
3.13	朋友	408
4	游戏要素	409
4.1	物品	409
4.2	角色	411
5	故事简介	418
6	游戏过程	419
6.1	故事背景	419

6.2	序幕	420
6.3	Gargantuopolis	420
6.4	Electric Priestess 的泡泡屋	421
6.5	Benthos	421
6.6	Harmony	422
6.7	New Boston	422
6.8	Electric Priestess 的泡泡屋	423
6.9	Ikairus	423
7	参考书目	424
术语表		425
精选参考书目		437

Chapter 1

第1章 玩家的需要



“当思索 Pokemon 为什么如此流行时，我发现最主要的原因在于，它的缔造者和设计师 Tajiri 先生并不是出于商业目的来做这件事情的。换句话说，他的目的并不是想创造出一个非常流行的游戏；他仅仅想做一个自己喜欢玩的东西，除了他对整个创造过程的喜爱，其中并没有包含任何的商业意义。然而出于某种原因，这个他为自己创造的东西却为他的同胞们所喜爱，并且其他国家的人们也可以一同分享……这就是关键所在：不是要制作可出售的商品，一件流行产品，而是因为喜爱，来创作一个所有人都会深爱的东西。这才是我们做游戏时所应有的最内在的感觉。”

——Shigeru Miyamoto 在谈到 Pokemon 创作过程时的话

游戏设计师花费了大量的时间，想要了解游戏玩家在电脑游戏中到底想寻找什么？他们能把什么放到电脑游戏当中，并且是以前没有做过、而又能够让玩家兴奋的呢？游戏设计师经常忽略玩家的所思所愿，相反，他们只加入了那些以前尝试过的游戏创意，改头换面地重复去年在游戏玩家中流行的游戏。当然，玩家去年喜欢的东西，今年仍然会喜欢。但很难还存有原发的激情。玩家一般不想买其他游戏的复制品，不希望“新游戏”使用的是旧思路，而并没有任何创意。然而成功的游戏对我们会有所帮助，当然不是为了克隆，而是为了分析。作为游戏设计者，我们可研究以前出现的游戏（那些我们在过去曾经热衷的游戏），并从中找出具有指示性的信息，这些信息解释了是什么原因驱使众多玩家们急于去尝试那些游戏，并解释了这些游戏长久地吸引住许多玩家的原因。



1.1 人们为什么玩游戏

我们应该思考的第一个问题是，为什么人们要玩游戏？为什么愿意打开电脑玩 Doom 游戏，而不是去参观博物馆或者去看场电影？相对于人类对其他娱乐的追求，电脑游戏的独特之处在哪里？游戏提供了哪些其他活动无法提供的东西？依照通常的理解，游戏吸引人的是它使我们可以更强调自己的独特之处，使我们的艺术形式区别于其他形式，而这正是其他媒体所无法满足的。为了能够获得成功，我们的游戏需要利用这些独特之处，使之升华，并利用它们创造出最好的游戏。



1.1.1 玩家需要挑战

如果提供了更强的挑战性，许多人都热衷于玩游戏。这极大地刺激了单人家庭游戏的成长，交流或炫耀力量好像不是问题。随着时间的流逝，游戏能够为玩家提供乐趣，尽管他们每次玩的游戏不同，但他们可以集中精神，而这与印刷品、电影或者其他艺术形式是不同的。就像玩弄 Rubik's Cube 或者是玩“拿掉铁环”这样的智力一样，游戏促使玩家主动思考解决问题的不同方案，设法理解游戏的机制。

当一个人面对挑战并且战胜挑战的时候，便已学到了很多。而挑战是存在于数学课本当中还是在电脑游戏中并不重要。所以，挑战游戏可以学习经验。即使学习局限于游戏本身，玩家也可以在游戏中学习，例如如何通过第 8 关……在最好的游戏中，玩家可以从游戏中吸取经验教训，并应用于生活的其他方面，而他们自己却没有意识到这一点。这也许意味着他们能够把问题解决方法应用到工作中，使用他们已经提高了的空间技巧更好地布置家具，或者也许通过游戏中角色的扮演来学习更好地表达感情。许多玩家热衷或渴望游戏提供的挑战，并且通过挑战来充实自己。



1.1.2 玩家需要交流

我有个朋友坚持说“游戏是反社会的”。这当然是很荒唐的，因为如果要发挥某种作用，几乎所有的非电脑游戏都需要一个社会团体。几万年前，出于朋友家人可以参加的挑战活动的渴求，游戏作为一个社会活动出现。电脑游戏设计者应该记住，游戏的根源及其吸引力的重要部分是它的社会性。

对大多数人来说，他们玩游戏的根本原因是与朋友或家人进行交流。在这里，我指的不是电脑游戏，而是棋盘类游戏和纸牌类游戏，比如说象棋、Monopoly（一种富翁棋牌游戏）、桥牌、Scrabble Diplomacy 或是 The Settlers of Catan。人们喜欢玩这些游戏，是因为他们喜欢和朋友们一起玩，这样的集体活动比看电影或看电视更受欢迎。事实上，虽然不少人喜欢玩单人纸牌游戏，但是更多人喜欢的是多人游戏而不是单人游戏。这是因为人们喜欢群体游戏时的感受。

但是如何在电脑游戏上实现交流呢？考虑到所有已经存在的游戏，其中大多数是单人的游戏。当然也有很多多人游戏，从 Doom 的“死亡模式”和与之类似的游戏，到经典的不择手段的 M.U.L.E. 游戏，再到在 MUD（挖泥巴）中的永恒世界，或与它们具有相似商业价值的网上游戏。





几乎所有“死亡类型”的多人游戏都是由单人游戏改成的。但是也有例外，例如 *Quake III* 和 *Unreal Tournament* 除了多人游戏 (Multi-Player, MP) 以外，这些游戏也提供了单人游戏 (Single-Player, SP)。单人和多人游戏有着几乎相同的规则和机制。但即便是玩“由单人转成多人” (single-player-turned-multi-player) 的游戏，玩家还是希望能够在玩游戏的过程中与其他人进行交流，任何与朋友们在一间屋子内一起通过局域网玩过这类游戏的人都可以证实这一点。局域网游戏的环境中充斥着玩家彼此呼来喊去的对话，夸耀他们最近的“战果”，或是告知他们已经被杀。而对于像 *Quake* 之类可以通过 Internet 来玩的游戏，交流的机会就没有那么多了。玩家可能相距甚远，只能用计算机交流。由于这些游戏强度高、动作快，如果玩家想“活”得更久一些，就根本没有时间将消息发送给对手。但这类游戏仍然提供了聊天功能，在游戏中处于比较安全的角落时、死后或是在游戏中间，他们都可以给彼此发消息。在游戏进行到比较激烈的时候，飞快打出来的消息都很短，可能只有几个字母。玩家在高强度的游戏中仍试着彼此发消息，是玩家想要表现其“高超技艺”。



图 1.1 Unreal Tournament 游戏提供了多人游戏的环境

另一种多人游戏叫做 *Persistent Universe* (不灭的宇宙) 或是“大容量多人游戏”。这些游戏更倾向于角色扮演类游戏的风格。玩家在虚拟世界里游荡，与代表其他人的角色进行交往。这些游戏在 Internet 这样的超级网络上运行，而不是通过局域网。所以，玩家只能通过他们输入到电脑里面的东西来实现交流。这些游戏比死亡模式的游戏在节奏上慢得多，因而玩家在游戏当中有更多的机会与人交谈。MUD 是这类游戏中第一个流行起来的，它从 20 世纪 80 年代末期在大学生中开始流行。那时大学生是长时间泡在 Internet 上的主流人群。这些纯文本的游戏要求玩家完成很多有趣的设置。然而玩家们需要花费大量的时间互相聊天、交流和扮演角色，解谜和完成任务已经不再占据首要的位置了。许多人玩网络游戏的时候就像和朋友交流一样，尽管





这些朋友可能是根本没有见过面的网友。的确，在永恒的世界中，MUD 把众多不喜欢玩单人游戏的朋友集中到了一起。这些人玩游戏的目的就是和别人聊天。游戏是他们进行交流的一种方式。

目前出现的多人游戏越来越多，许多游戏开发者都迅速指出他们的游戏在 AI 方面具有竞争力的优势。如果以人为对手的话，游戏就更加不可预知，并且比大多数游戏中的 AI 更具挑战性。这也就是人们为什么喜欢多人游戏的原因。但这些多人游戏最大的优点在于，它们将电脑游戏转换为实际可用的社会经验，这也是玩家玩游戏时最具推动力的因素。



1.1.3 玩家需要独处的经历

也许我第一次说玩家想要交流的时候读者已经很困惑了，但是我还提出玩家想要独处的经历。当然，这两种情况不会同时发生。有些玩家正在寻找交流的机会，而另外一些人却试图寻找可以独自享乐的机会。有时朋友不在身边或是厌倦了朋友，或只是不想与他人说话。其差别就像去电影院看电影或独自在家看影碟，“反社会性的单人游戏”具有足够多的朋友。

但是游戏是区别于看书或看影碟这样的社会经历的，游戏本身与玩家之间存在着交流。游戏有像人一样的反应，或至少能够模仿人类的行为。但是玩家可以随意开始或停止游戏。所以电脑游戏可以“伪造”人类行为中有趣的部分，而不包含任何潜在的令人烦恼的行为。这样，玩家和电脑游戏之间的行为是动态的、交互的、非反社会的。



1.1.4 玩家需要炫耀的权利

玩家玩游戏也是为了赢得尊重，特别是在多人游戏中。在 *Doom* 中跟朋友吹牛，能让他们尊重你。比如说“Bob 不擅长代数，但是他肯定能在死亡游戏里消灭我”。甚至在单人游戏里面，玩家跟朋友谈论怎样结束一个游戏或是多么擅长另一个游戏。玩家吹嘘他们如何只在几个小时之内通过整个游戏的所有难关。如果看了“街机”游戏的旧版和新版，最高分的奖台和能够用其他人的名字“溜进”游戏，诱惑玩家反复玩游戏。如果很少有机会在日常生活中吹嘘自己，在学校里也没有引以为豪的良好成绩，他们可能会走上街头，在 *Centipede* 游戏中向朋友表现出自己的机智。即使未明显地显露，当他们在游戏中取胜的时候，玩家也会产生很强的自我满足感。当玩家挑战游戏并取得胜利的时候，他们意识到自己能够做得很好，或许比大多数人都好，这让他们自我感觉良好。



1.1.5 玩家需要情感体验

像其他形式的娱乐一样，玩家在玩电脑游戏的时候也在寻求付出情感。这很简单，就好像 *Doom* 那样快速反应游戏带来的刺激和紧张。或者可以想得更复杂一点，在 *Steve Meretzky's Planetfall* 中，当玩家的机器人伙伴为了玩家而牺牲了自己时，玩家就会产生失败感。遗憾的是，玩家的情感仅局限于面对冲突时的兴奋、紧张，老是完不成任务的失望以及获得成功的喜悦和成就感。但是很多人喜欢悲剧性游戏，或希望游戏能有电影式的悲伤结局，甚至仅仅为了听听忧伤的音乐。玩家想要的是游戏所带来的感觉，这感觉不必是积极的或幸福的。人们也许能从





游戏中得到渲泄，这使得游戏变得更有价值。许多经典游戏，比如 Centipede 或 Space Invaders 都是赢不了的。不管玩家水平如何高超，最终都是游戏击败玩家。某种程度上说，这种游戏是“面对失败的教材”，因为每次玩家的结局都是“悲剧”——然而玩家却不会放弃。游戏总是战胜玩家，玩家可能会产生无望的感觉，这也是不应忽视的感觉。玩家所得到的感觉就是艺术的目标。

感情范围不是电脑游戏所能挖掘的。这个例子来自 Planetfall，是我在前面引用的很少的例子之一。在这个例子中，玩家很少能够遇到游戏中的人物，除非最后被杀。许多开发人员意识到开发这样的游戏太伤感了。但是在 Planetfall 中，悲剧故事交织在游戏当中，设计者 Steve Meretzky 挖尽了所有的伤感……留在玩家记忆中的尽是悲情。游戏设计者聪明地把精力集中在扩展游戏感情经历上，集中在除兴奋和成就感以外很少被发现的感情领域。



1.1.6 玩家需要幻想

“讲故事”这种艺术形式的流传主要原因是幻想。无论看小说、电影或连环画，许多人把讲故事看做是脱离世俗生活的一种方式，以逃入完全不同的世界。这个世界充满了英雄式的人物，他们进入了世外桃源或遇到了其他迷途的人。当然不是所有的故事都讲述了让人喜爱的主角，但肯定的是故事的大部分被定义为“逃离现实者”。许多批评家嘲笑这样的逃离者。确实许多非常好的书、电影和喜剧有更现实的背景和主题并产生了很大的影响。然而事实仍然是这样，许多人想要进入到一个比现实世界更精彩的虚幻世界中。

电脑游戏比逃离主义更有魔力让人沉浸其中。在游戏中，玩家有机会真正成为更辉煌的某个人，控制虚拟的冒险家、勇敢的剑客或是歌剧中的英雄。观众只在小说或电影里面才见过的精彩、完美的人物。一个设计良好的游戏中，玩家能够真正有机会过上幻想中的生活。更美妙的是，这些虚拟生活与现实生活不可同日而语。在大多数游戏中，玩家不用担心吃喝睡觉或是洗澡之类的琐事。所以游戏创造了没有枯燥细节的“纯洁生活”。最重要的跟其他艺术形式相比，由于游戏的交互性的本质，幻想的水平被大大地提高了。

电脑游戏产生的幻想使玩家产生了在安全社会环境下不能接受的行为。许多流行的游戏允许玩家假扮成罪犯或是刺客。Diver 就是一个很好的例子。尽管游戏解释说玩家玩的其实是卧底。在 Diver 中，玩家假装他是必须在汽车底盘中躲避警察的罪犯。超越警车会异常害怕，尤其在被他人拉了一下子的时候。尽管多数玩家从来没有考虑过在真实生活中匍匐在汽车底盘下，但是被禁止的行为总是很有诱惑。电脑游戏为玩家提供了这种可能性，让他们发现隐藏在日常生活中人的“两面性”。

玩家也许会幻想历史事件。如果玩家见过拿破仑，滑铁卢战役也就大为不同了。如果玩家是 20 世纪的 railroad 男爵，也许就能够创造一个强大的金融帝国。整个历史情节的游戏，从战争游戏到经济活动模仿，允许玩家探索历史事件。玩家可以看到因选择的不同，而出现的不同结果，其结果可能与真实历史大相径庭。然而许多玩家花费很多时间停留在“过去”，探索如果做出不同的选择，事件该如何发展。游戏给玩家了一个“改变历史”的机会。

甚至当没有令人兴奋的原因时，如果一个人的实际生活平淡无味，花点时间扮演游戏中的





人物也许就是很有意思的事情。好的电脑游戏能够提供给玩家别人得不到的机会，通过别人的眼睛来看世界。很多玩家能够证明角色扮演游戏和幻想游戏是很有趣的。



1.2 玩家的期望

一旦玩家因为某种原因决定了玩某个游戏，他就会对这个游戏本身有一些期望值。除了游戏很精彩、看上去相当漂亮之外，玩家还有一定的期望，如果这些期望没有满足，玩家很快就会失望，并寻找其他的游戏来玩。游戏设计者的任务就是使游戏可以满足这些期望，以便使玩家选择其游戏。那么，什么是玩家想要的游戏呢？



1.2.1 玩家期望一致的世界

当玩家玩游戏的时候，他们就会马上了解在游戏环境中可以进行哪些操作，及其操作的结果。当玩家期待一个动作产生特定的结果，但是游戏因为不可知的原因产生了不同的后果，那么几乎没有什么事情比这更让人沮丧了。更糟糕的是，随之而来的结果是不可预测的，玩家无法产生任何期望。尝试了一个动作之后，并没有发生所期望的事情，这只能让玩家困惑或失望。很快他就会去找一个更有可靠的游戏来玩。因为不可预知的游戏让人很沮丧，所以必须保证动作和结果的一致性。

游戏动作结果的可预知性是非常重要的。玩家绝不希望一个动作有时可行、有时不可行，对于不同的结果都应有合乎情理的原因。比如，在 Tekken 游戏中，如果玩家一脚没有踢中对手，那是因为跳起来后距离对手太远了，或是玩家可以理解的其他原因。让玩家了解动作失败的原因是很值得注意的。游戏的内部逻辑（在这个例子中是碰撞系统）会知道玩家为什么没有踢中对手。但是，如果玩家不能找到动作失败的原因，那就和没有原因一样糟糕。此外，如果只有专家级的玩家才能够理解动作失败的原因，当因为不能理解的原因而被击败的时候，许多新手就会感到沮丧。如果在情况非常相似的情况下，玩家可以一脚踢中对手，而下一次却失败了，玩家也会感到气馁。

弹球游戏是另外一个很有意思的例子。当然，因为这个游戏是以真实世界的物理机制为基础，所以它是完全的结果可预知的游戏。专业级玩家知道这一点，会有效地利用它，但是新手可能会有问题。当球在两个挡板之间直行，或是滚向槽边时，无经验的新手并不会发现自己“做错了”。这些新手会以为弹球游戏是靠运气来玩的，不想再玩下去了。事实上，在弹球游戏当中，不同水平的玩家会有完全不同的成功率，这说明了玩弹球游戏不只是靠运气。但是只有那些之前失败过很多次，然而还一直坚持不懈的人才能了解到这一点。我并不是暗示建议放弃弹球游戏，或是说应该进行简化，但这个游戏有一个缺点，对于看不到动作和结果之间的联系的新手来说，这个游戏似乎缺少吸引力。



1.2.2 玩家期望了解游戏环境的界限

玩游戏的时候，玩家肯定需要了解哪些动作是可行的，哪些是不可行的。他不需要立刻明



白在特定的环境下需要做出什么样的动作，但是一定想先了解哪些动作是可行的，哪些动作超出了游戏空间范围之外。



图 1.2 在 Doom II 中，玩家不希望与他正在攻击的怪物进行对话

例如，在 Doom 游戏中，玩家凭直觉断定他不会跟正在战斗的怪兽说话，甚至不会在建议投降(最有逻辑的动作)的时候和怪兽谈话。玩家觉得这样的交流是超出游戏范围的。假设 Doom 中有一个怪兽，只有在玩家对它很友好的情况下才能击败它，要通过富于机智的谈话来取得胜利。这时玩家就会感到沮丧，因为他们开始意识到，在游戏中玩到了这一关，要想取得胜利，惟一需要做的就是消灭一切可以移动的东西，同时避免被人击中。交谈完全不在可想象的动作范围内。

当然，健谈的怪兽在 Doom 中是很极端的例子，有不可预测的界限，但很多游戏破坏了这个原则。这些游戏让玩家使用某种特定的游戏机制完成动作和升级，之后再插入新的难题，这些难题只能用全新的游戏机制来解决。问题是，当玩家已经学会了游戏规则后，游戏却突然要求玩家去寻找或尝试新的规则。一旦玩家掌握了游戏使用的规则，他们就不愿再接受随机出现的新规则，也不愿意接受完全不熟悉的机制。

1.2.3 玩家期望合理的游戏解决方案

一旦玩家花了很多时间来玩游戏，他便会熟悉和游戏环境的规则。在解决了大量的难题之后，他就会知道哪种方案是可行的。在后面的游戏中，当不断地解决了各种难题后，玩家就会得到他所认为完美解决方案。如果他努力尝试的方法并不可行，并且没有什么合理的原因可以解释，他就会感到沮丧，并有上当受骗的感觉。





对于试图在某种程度上模仿现实世界的游戏来说，这种设计上的困难尤为突出。在现实世界中，想完成既定的目标可以使用许多方法。所以在电脑游戏中也应该如此。当然，设计者应该提供至少一种解决难题的方法，并使其成为相当合理的方案。但同时也可能存在其他同样合理的方案，除非设计者可以保证这些方案可以行得通，否则当玩家尝试“备用方法”却发现它们并不管用的时候，玩家就会感到很气愤。游戏设计师的任务是预测玩家在游戏世界中会做些什么，并保证玩家尝试这些动作的时候会发合情合理的事情。

1.2.4 玩家期望指导

好的游戏应告诉玩家他们可以做什么。玩家希望创造自己的成功故事，希望找到游戏中取胜的方法和自己独特的东西。但与此同时，玩家要有完成游戏的设想。如果没有指导，好像和实际生活吻合了，这是玩家已在进行却并不希望的生活。许多玩家玩游戏的目的是远离现实生活，他们渴望幻想和解脱，因此不愿意在电脑上重复现实生活。

玩家想要知道自己的目标，并且希望得到如何实现目标的提示。当只有目标而不知道如何实现的时候，玩家就会到处乱撞，尝试自己能想像到的所有操作，当所有的尝试都失败的时候，他们就会感到非常沮丧。当然，不知道目标是什么，玩家只能无目的地到处乱转，欣赏图片或感叹游戏世界的广阔。而在游戏世界里无所事事，游戏本身也就失去了意义。如果玩家不知道目标是什么，目标也就如同不存在一样。



图 1.3 SimCity 3000 是在一个系列的城市模仿中的第三个“软件玩具”，玩家在玩游戏的时候并没有确定的目标

SimCity 是无目标游戏的经典例子。实际上，游戏的创造者 Will Wright 称它为“软件玩具”，而不是游戏。SimCity 像玩具一样，玩家可以对它做任何想做的事情，不必清楚地知道成功或失



败。某些时候，SimCity 像一套乐高拼装玩具，为了获得创造带来的激情，玩家能够建造任何想要的东西。SimCity 是城市模拟游戏，规则是只要愿意，玩家就可以建造城市。但是由于游戏模仿了现实（建设和管理城市）并且玩家知道什么是现实中的成功（一个有着大型露天体育场、富丽堂皇的图书馆和幸福的居民的城市），玩家会本能地利用自己的规则取得游戏中的成功。玩家努力构想完美的城市、保持居民的幸福和经济的繁荣。巧妙的是，玩家依靠现实中的经验指引。如果 SimCity 的模拟系统是玩家毫不熟悉的，它也就没有那么流行了。尽管游戏没有一个明确的目标，但游戏的本质和现实中的背景鼓励玩家去实现他们的目标。玩具从最初变成了游戏，这也驱使玩家继续玩下去。

1.2.5 玩家期望逐步完成游戏

假设玩家了解在游戏中所要实现的目标，他们当然想知道完成任务的正确途径。最好的方式是，在实现目标的路径上提供大量的像主要目标那样的次要目标。于是，像完成主要目标一样，但玩家只能获得比较小的奖励。当然，玩家可以沿此路径一步步地走下去，从一个次要目标到达下一个次要的目标，这是获得正确方法的必要手段。如果这种方法没有提供正确的反馈，并且达到目标的道路漫长而艰辛，玩家可能会以为他们的方法不正确。当游戏没有积极地提示他坚持这个方法的时候，玩家就有可能尝试其他的方法。当他不能解决遇到的障碍的时候，他会因沮丧而放弃，并告诉他的朋友玩此游戏是多么痛苦。

1.2.6 玩家期望能沉浸到游戏中

一个音乐喜剧的导演，当看到演员碰到舞台两侧的幕布时会很生气。她认为当观众看到幕布晃动的时候，观众的注意力就会从演员身上转移开。这种怀疑是站不住脚的。观众想到他们看到的只是一出戏而不是所有，比如幕布的晃动。夸张一点说，如果观众看到了幕布的晃动，百老汇就会崩溃。

但是上面的导演说中了一点，这一点也直接应用到了电脑游戏中。一旦玩家进入游戏，处于某一通关过程中，玩家了解游戏的控制规则，因此变得兴奋，扮演起了幻想中的角色，玩家不会轻易走出这种经历——当然，游戏不要崩溃（否则将会是最不愉快的经历）。除此之外，玩家不会考虑太多的游戏界面 GUI。如果 GUI（图形用户界面）设计得不够明晰，不适合游戏世界的艺术，就会显得很和谐，并破坏玩家的专注性。如果一个应该走在路上的人物开始走向天空，而没有任何原因可以解释，那么玩家意识到这是一个“缺陷”（bug），他就会开始产生怀疑。如果玩家遇到了一个难题，随后找到一个完美的解决方法，而这种方法并不奏效，玩家会再次提醒自己只是在玩电脑游戏。所有这些缺点都会转移玩家对游戏的专注。每次玩家都会突然从游戏世界的幻想中觉醒，很难再次集中到游戏之中。要记得，许多玩家玩游戏的目的是想沉浸于幻想当中。当游戏的特有风格总是提醒玩家“这只是个游戏”的时候，玩家就很难沉溺于幻想。





图 1.4 尽管 Mario 很有名，但他并没有非常独特的个性。这是他在 Super Mario 64 中出现的场景

另一个玩家专注的重要方面是他在游戏中控制的人物。大多数游戏在某种程度上来说都是角色扮演类型的。如果玩家控制的人物——游戏世界中的主体玩家不喜欢，或玩家看不到他自己，玩家就不会那么专注。例如，在第三人称动作/冒险游戏 Super Mario 64 中，玩家控制的角色 Mario 就没有独特的个性。Mario 水管工人的打扮看上去相当特别。但是他从不多说，就像化身一样，玩家在上面可以创造自己的个性。另一方面，一些冒险游戏主角的行为像被宠坏的孩子，玩家不得不看着他的人物一再做出讨厌而又白痴般的事情。每当人物说了玩家不想说的话时，玩家便不得不提醒自己——这是在玩游戏，游戏中的任务不由他控制。玩家为了真正沉浸其中，必须把自己看做游戏中的替代品。

1.2.7 玩家也期望“失败”

如果第一次玩游戏的时候就可以“打通关”，便会乐趣顿失。如果游戏如此没有挑战性，以至于玩家第一次尝试的时候就通过了——它就不是什么好游戏。如果玩家需要玩这样简单的游戏，他还不如去看一场电影。记住，玩家之所以玩游戏，是因为他们需要挑战。挑战必然意味着玩家不能轻易获胜，在获得最终成功之前，必须做出很多尝试才能克服困难。太容易取得的胜利是廉价的胜利，这与在拳击比赛中战胜身材比自己矮小一半的人没什么两样。

玩家想要输不是游戏本身的特性，而是源于玩家自己的感觉，理解这一点很重要。玩家失败的时候，他会看到本应该做的事情，并立刻寻找自己为什么失败。如果玩家认为游戏通过“小把戏”或“暗算”战胜了他，他也会对游戏失去兴趣。玩家不会为失败而责怪自己，而会责怪没有挑战性的游戏。

让玩家在游戏开始的时候尝到一点甜头，这也是一个好主意。这会让玩家沉浸在游戏之中，觉得游戏并不是很难。玩家甚至能在游戏中产生优越感，随后，困难必须逐渐升级，让玩家失败。这时，玩家已经沉浸于游戏之中，有时间去探索游戏，并想要一直玩下去，直到克服击倒他的困难。如果玩家在游戏中过早地被打败，他会认为游戏太难了，或者不知道如果继续玩下



去会得到什么奖励。允许玩家在最初的时候赢，玩家就会知道能够成功，进而努力地克服困难。

1.2.8 玩家期望公平的机会

玩家不想使“尝试/错误”成为惟一的克服困难的方法，错误会引起游戏当中人物的死亡或游戏的结束。玩家可以通过“尝试/错误”找到正确的方法去跨越障碍，但是在第一次尝试的时候，玩家应该可以找到获得成功的方法。再把这个方法沿用到整个游戏之中，如果玩家很小心并且很熟练，那么即便是以前没有玩过游戏，玩家也可以在不丢掉性命的情况下通过游戏。也许没有玩家能够在第一次玩的时候如此娴熟，当我们讨论的时候，在完成游戏之前，玩家需要失败多次才能完成游戏。然而一次不死就获得成功，这对玩家来说理论上是可以实现的。玩家很快意识到解决问题的惟一方法的就是尝试各种办法，直到其中一种奏效为止。如果玩家可以逃脱次次“暗算”，但他们仍感到游戏设计者目光短浅，并不存在真正的好方法来避免死亡，他们同样会变得很沮丧，开始诅咒游戏，并且停止在这个游戏上花费时间。

1.2.9 玩家不期望重复自己

一旦玩家完成了游戏中的目标，通常不希望多次重复。如果设计者设计了特别具有挑战性的难题，就算玩家解决了它，并感觉很难完成，也不应该在游戏中重复上述过程。比如，在3D动作/冒险类游戏中，除非解决难题有很多乐趣，或者每次解决难题之后都可以得到不同的奖励，否则同样痛苦的难题不应该再以类似或稍加改动的形式出现。如果玩家在游戏中一直在解决这类问题，并且没有更多的乐趣，他就会因为缺乏新挑战而失去兴趣，并指责游戏设计人员。

当然，许多游戏的原则就是让玩家可以重复自己，或者至少以稍有不同的方式重复自己的动作。尽管出现在不同游戏中的挑战是游戏的独特性所在，但运动游戏 NFL Blitz 和竞赛类游戏 San Francisco Rush 都一再重复同样的背景。经典街机游戏 Centipede 和 Defender 也存在有相当多的重复。Tetris 也许是重复游戏的冠军，然而玩家从未厌倦过这种游戏。玩家在游戏当中不想重复自己，在这类不想重复的游戏中，探索是娱乐的关键部分。在探索过程中，在很多情况下玩家所面临的挑战是没有变化的。探索过了一次游戏世界之后，随后的探索就没有那么多乐趣了。每次玩家玩 Tomb Raider、Doom 或是 Fallout 的时候，面临的挑战都大致相同。所以玩家不介意前面游戏的重复，然而在游戏的后面要强迫重复自己，他们会因此降低兴趣。

游戏存活的原因是玩家不期望重复自己。一旦能够完成非常勇敢的任务，应该能够退到玩过的地方，当死掉的时候，玩家就能够保存那个位置。如果游戏给玩家出了一个大难题，经过多次尝试，玩家最后解决了，玩家应该有机会去保存游戏的进程，允许玩家保存游戏可帮助玩家避免重复。

许多游戏自动保存玩家经历的部分，这个过程被称做保存断点 (Checkpoint)。这是个高级方法，是因为玩家成功完成了一个艰难的任务之后，就可以接近一个更新、更刺激的阶段，他会立刻想要去探险，经常在兴奋点上忘了保存游戏。于是当他在新的阶段被击败的时候，游戏就会让他回到他上次保存过的地方，那是玩家早前克服困难的地方。现在玩家要再次面对经历过的挑战。然而，如果游戏设计者意识到了这个困难是很难克服的，他就会帮玩家自动保存玩家的位置，这样当玩家在新地区死掉的时候，他也能够在新地区重新开始游戏。但是，自动保





存在玩家要求“替代保存”的时候不能使用。但是替代和保存可以联合起来使用。这样，玩家习惯的保存工作便显得很合理。经常忘记保存游戏的玩家允许用其喜欢的方式通过游戏而不必点击“保存”按钮。事实上，自动保存给玩家提供了一个更好的经验，每次玩家接触保存游戏的屏幕或菜单的时候，他就会提醒自己，他正在玩游戏。如果玩家没有保存就通过了游戏，他的经验就会更加明显。

1.2.10 玩家期望没有令人绝望的障碍

在玩一个游戏时，绝不允许出现玩家根本无法胜利的情况，哪怕是再小的可能都不可以。很多旧的冒险游戏却以打破这个基本定律为乐。在那些游戏里经常出现的情况是：如果玩家在特定时间没有做特定事情，或者没有在先前的游戏进程中于某处拿到一个小道具时，玩家将永远不可能完成游戏。而问题就在于，玩家往往要经过很长时间毫无收获的游戏之后才会意识到这一点。这个玩家的游戏实际上已经（失败地）终结了，可是玩家却还在“玩”。没有什么能比玩一个永远不能胜利的游戏更令人沮丧的了。

举个例子来说，时髦的3D探险游戏，不论是 Unreal 还是 Super Mario 64，都需要考虑到玩家可能会在3D世界中遇到令人绝望的障碍。在这种类型的游戏中，经常可能有陷阱或深坑，使玩家掉进去而不会死亡。而这时非常关键的就是，要提供脱离这些深坑的途径，比如梯子或可以让玩家回到正常游戏的平台。用来逃脱出陷阱的方法可以很难，这没有问题，但是必须可以逃脱。如果玩家掉进的是一个无法逃脱的陷阱，那又该怎么处理呢？如果确实玩家无法逃脱，那么玩家的角色应该被陷阱中的某些东西杀死，可以是掉落的同时就发生（比如陷阱的底部是通电的）或者在之后比较短的时间内发生（比如说陷阱会在玩家角色落入10秒钟后被岩浆淹没）。总之绝对不允许出现有玩家（的角色）被阻止在一个无法继续游戏之处而又不会死亡的情况。

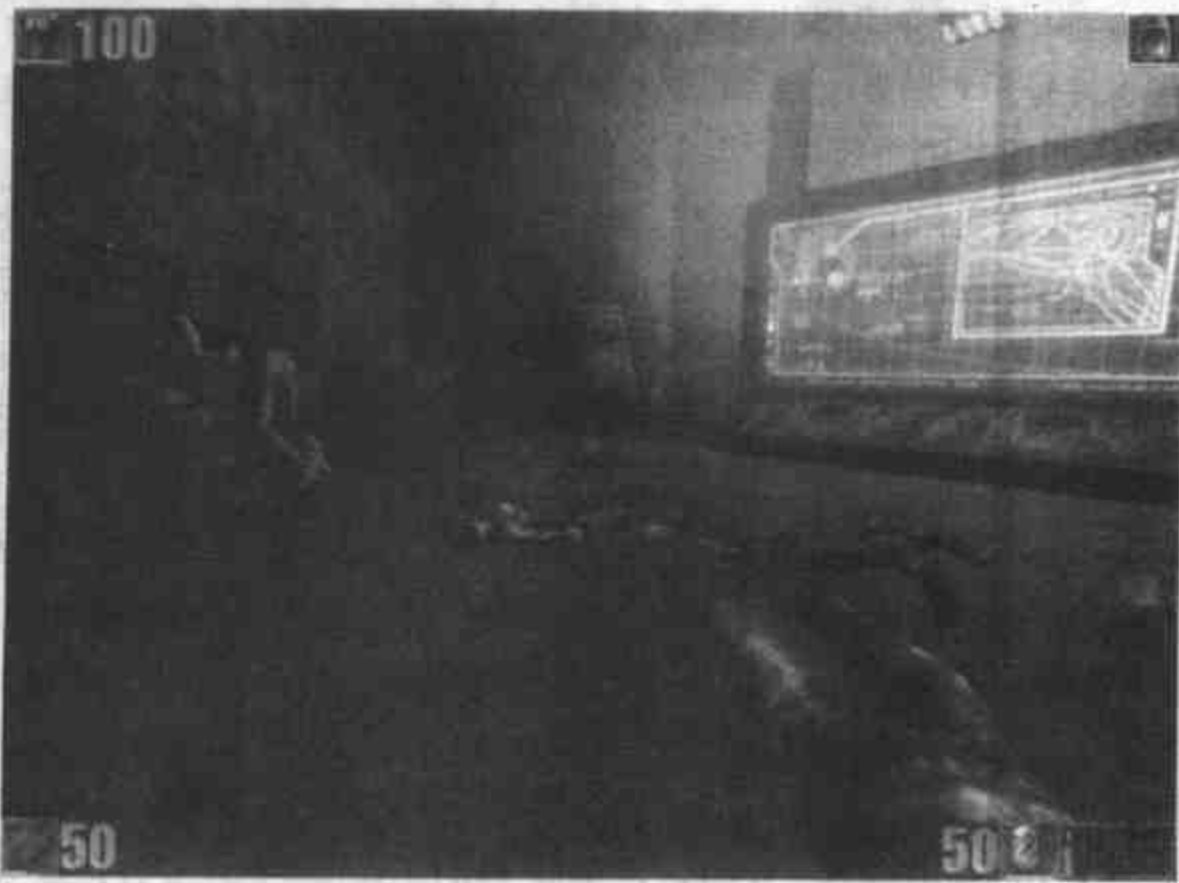


图 1.5 对于类似 Unreal 这样的 3D 动作/冒险类游戏，在关卡地，图中要保证不能让玩家永远陷入无法出来的建筑物里



对于 Civilization (文明社会) 的一个主要的批评就是, 实质上已经结束的游戏却可能继续进行很久, 除此一点之外, Civilization 也的确是很优秀的游戏。当只剩下两个国家时, 并且其中一个已经毫无希望地远远落后的时候, 此时游戏已经失去了其真正趣味, 而强者却仍然在继续搜寻和屠杀弱者。实际上, 落后的那个国家在技术上来说并不是毫无希望的。那个落后的玩家仍然可以后来居上, 并且赢得游戏, 这并非没有可能。这个玩家遇到的这个困难的程度与落入绝对没有出口的陷阱的玩家是不一样的, 但这个玩家可能落后得太多, 以至于基本上没有翻身的机会; 他要赢得游戏所需要拥有的运气以及期待对方会犯错误都是摇摆不定的。对于这种情况, 一个可行的解决办法就是让 (落后方的) AI 能够发现敌方已取得压倒性的优势并且投降, 就像一个已经落后到毫无希望的玩家会做的一样: 退出, 再重新启动游戏。

1.2.11 玩家期望玩, 而不是看

有一段时间, 业界对于交互式电影 (Interactive Movies) 的前景相当看好。在这段时期内, 计算机游戏的片头动画 (Cut-Scene) 越来越长。一些小有名气的电影演员也开始在片头动画中扮演主角。游戏变的越来越缺乏交互性, 事实上, 也越来越不像游戏。而预算却在不断上涨。然后令人惊讶的是, 玩家一点也不买这种游戏的账——他们不喜欢此类游戏。于是制作公司倒闭, 业界的每个人都在抓耳挠腮的想到底哪里不对。玩家当然是知道原因的, 游戏设计者们也很快知道是哪里出了错。问题在于以前玩家想玩, 而不是看。他们现在也还在这样想。

我并不完全反对使用片头动画, 对于交待游戏的故事, 或者传递给玩家在下一段中必要的一些信息时, 这些动画是非常有用的工具。也就是说, 我认为片头动画应该缩短并且最少化到一个绝对短的程度——这个程度已经是叙述游戏和构建后续游戏进程所必须的了。超过一分钟的片头动画, 尤其那些并非提供后续情节基本信息的片头动画是要避免的。片头动画的方式倒无所谓, 可以是黑幕上滚动显示的文字, 或者是真人出演的电影画面, 赛璐珞动画 (Cell Animation), 或者是直接由游戏引擎生成的画面, 总之只要使游戏中断的时间之和不要超过一分钟就行。如果这其中包含了玩家的参与, 比如玩家部署下一关的军队位置, 那么这就不是一个严格意义上片头动画, 它可以根据需要而延长。并且, 如果片头动画包含了游戏过程中关键性的信息时, 设计者应该让玩家能够无限次的重放这段画面。

片头动画的品质也并非至关重要。有些游戏的片头动画的演技可以用“历史上最烂”来形容, 这些常是助理出品人和首席测试师出演的。也有些游戏有着好莱坞电影品质和内容的片头动画, 甚至更好。但是最后, 只要游戏本身有着吸引人的地方, 玩家就会希望尽快开始游戏而跳过片头动画。

总而言之, 人们玩游戏是因为他们希望得到一些不同于电影、图书、录像节目或喜剧的感受。我不会把“图书馆关门了”或“电视信号不好”等原因也归结到人们玩游戏。玩家需要游戏, 所以游戏设计者就应该给他们一个游戏。

1.2.12 玩家不知道他们要什么, 但是他们知道什么是他们要的

可能有人会认为这是一个背叛团队的论点, 但实际上并非如此。在开发游戏的过程中, 游戏测试者的工作是其中很重要的部分。游戏测试者的责任, 我认为不仅仅是在游戏中寻找缺陷





bug，而是分析游戏的可玩性，并给出建设性反馈信息。一旦游戏开发到可以判断游戏大部分的可玩性的时候，游戏设计者就应该让很多的人来试玩这个游戏。

另一方面，在创作一个游戏之前，要一个由玩家组成的团队来集思广益是没有什么用的。玩家当然适合于判断一个他们正在玩的游戏是否好玩。但他们可能并不能以最有效的方式阐述清楚一个游戏中哪些方面让人喜欢，哪些方面让人讨厌，然而他们肯定是知道什么时候他们玩得开心，是否实现了其幻想，是否遇到了一个真正的挑战，或者游戏是否让他们感到兴奋。然而仅仅因为他们玩过很多的各种成品游戏，并不意味着他们能够合格的来评判那些原始的游戏创意。同样地，他们提出的关于游戏的创意也不一定是好创意。只有少数几个人可以讨论电脑游戏的创意，并且判断最终制作出的游戏是否好玩。有这种能力的人是最适合当游戏开发者。不是所有的玩家都有这种能力，所以当被问到喜欢玩哪种游戏时，玩家可能真的不知道他们要什么样的游戏。但是，如果他们看到一个好游戏时，他们就知道这就是他们想要的了。



1.3 一个无截止符的清单

当然，关于玩家需求的探讨可以写满一整本书，并且能不停止地继续下去。我鼓励你们，无论是有抱负的游戏设计者，还是那些已经发布了多个成熟游戏的设计者，都来做出你们自己的关于玩家需求的清单。想想在游戏中是什么让你灰心丧气，什么给了你最大的满足感。然后找出为什么你们会对一个游戏的机制做出了这样的反应。该游戏哪里做的对、哪里做的不对？以便得出你自己规则的清单，并且应用到游戏设计中去。没有游戏测试者的反馈，一般是很难判断出来游戏是否有意思或者是否有吸引力。但是有这么一套可以系统地应用到设计中的规则，你就可以知道其他人是否会喜欢你的游戏了。



Chapter 2

第2章 Sid Meier 访谈录



Sid Meier 是最著名的计算机游戏设计者，尤其在西方电脑游戏界广为人知。在他近 20 年的游戏设计生涯里，涉足了游戏设计的方方面面。他不仅参与创建了 Microprose 公司，并首先致力于飞机模拟器的开发，而且在完成了他的经典之作 F-15 Strike Eagle 和 F-19 Stealth Fighter 之后，他的事业达到了顶峰。随后，他转移到目前他较为知名的那一类游戏，开发了诸如 Pirates!、Railroad Tycoon、Covert Action 和 Civilization 等经典游戏，最后这个游戏被认为是此种类型中最受欢迎者。最近，在他的新公司里，Meier 创建

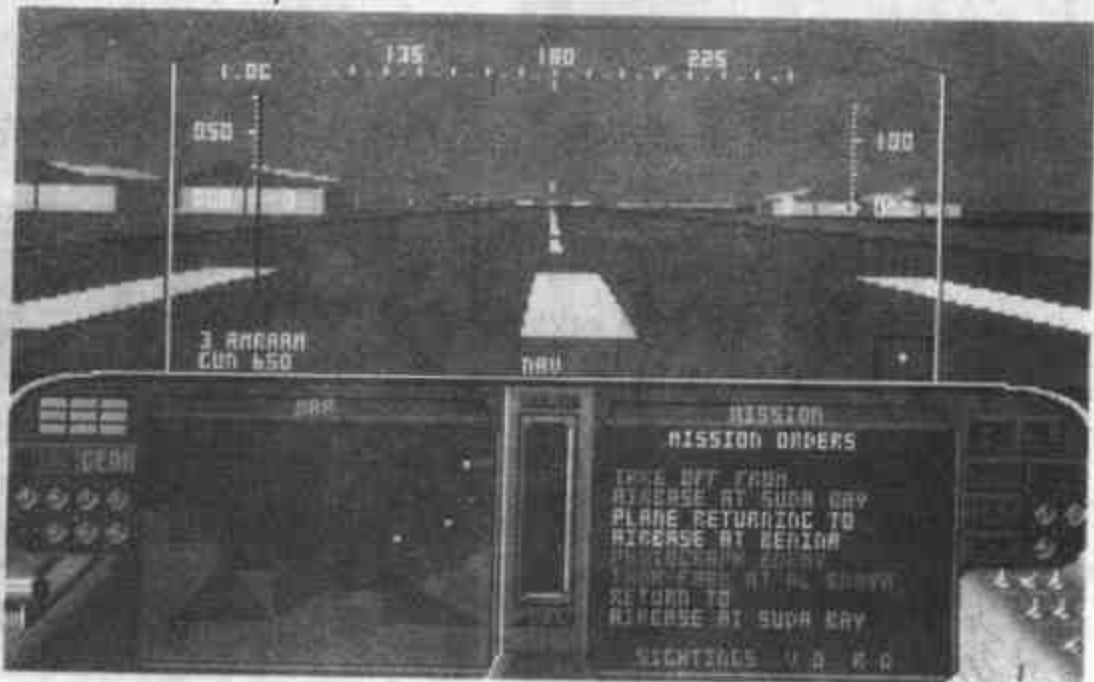


图 2.1 F-19 Stealth Fighter



了独特的实时类型名为 Gettysburg! 的战争游戏。他所设计的游戏经常能打动回头客，因为游戏总是能保持高水平，而且他从不重复自己，他总是在自己的项目中采用新颖别致的东西。如果这个世界上有一个人，总是能使游戏变成一种引人入胜的体验，那他就是 Sid Meier。



问：你最初的游戏作品是飞行模拟器，后来你转移到战略游戏，这个游戏使得你很出名。

是什么原因使你从一个类型转到另一个类型的？

答：这并不是一个经过深思熟虑过的计划。我经常去尝试开发一些我感兴趣的题目。我想到过很多种游戏主题，比如当我是孩子时感兴趣的，或是发现一本关于内战的书籍，或是飞机……总之，任何我所能想到的。我想科技也是促使我转变的原因，在有些时候，科技会对某种游戏的实现起到重要的帮助作用。或是某种战略游戏的市场已经成熟，或是你一直想开发某个游戏，但认为时机不当。我的转变——特别是从飞行模拟器到战略游戏的转变，主要有两个原因。第一个原因是，我刚刚完成了 F-19 Stealth Fighter 这个游戏，它包含了我对飞行模拟的所有想法，在此之后，我所能做到的只能是增加些好的图形，更多的声音或更多的场景，或其他诸如此类的东西——我觉得对飞行模拟已经没有新的想法了。我能想到的所有关于飞行模拟器的好想法都已在那个游戏中了。与此同时，我已经在玩 SimCity 和 Empire 这两个游戏，对战略游戏有一定的认识，我觉得它们值得花时间和精力去探讨。这两个因素使我开始开发 Railroad Tycoon，然后是 Civilization，即一系列的战略游戏。

我发现先考虑类型、然后再考虑主题是非常危险的。比如说我想做一个实时战略游戏，好，那主题呢？我可能会这样说：“我想制作一款关于铁路的游戏。那么怎样制作这款游戏才能让它很有趣呢？是实时类型的，还是基于回合制，或者是第一人称类游戏……”首先要提出游戏的主题，然后找到合理的方式和适当的风格来制作游戏，或者换一种方式来说：“好的，我想制作第一人称的射击游戏，还需要做什么呢？”“如果要从风格的角度上来讲，基本上就可以说：“我会找一种现成的模式”。那么我认为，如果以这种态度来做事的话，就不会做出真正好的游戏来。他们首先会想：“这的确是个不错的主题，如果再加以修改的话，就肯变成一个实时战略游戏。”



问：当你为一个新游戏提出自己的想法时，你是否是首先开始设置游戏，而不是首先确定游戏的风格。

答：我认为 Pirates! 是一个比较好的例子。这个游戏的想法是制作一款海盗游戏，并且实际上也是按照这种想法来做的。“关于海盗，我并没有什么比较好的创意。海盗电影、在海上航行、舞剑作战，停留在不同的村庄里，还有各种各样的人，真的不太适合形成某种风格”。所以我们挑选并选择了不同的



图 2.2 Pirates!





事情，如实时方式的一系列航行和基于菜单的冒险系统，进入一个村庄，然后进行一系列的战斗。对于游戏的不同部分，我们选择了自己认为比较合适的各种风格，而不是说：“我们要制作一个实时的游戏，或者是基于回合制的，或者是第一人称的，或者是其他类型的”，然后再把海盗的想法加到游戏中去。

① 问：我认为Pirates!游戏中设计的那些小游戏很有意思，但是你再也没有使用过无联系的子游戏。难道你不喜欢在其中结合小游戏的方式吗？

答：我认为在Pirates!游戏中使用子游戏的效果很好，但是这样做并不适合于所有的情况。在这些年里，我所了解的游戏设计中一个重要原则就是：一个伟大的游戏要超过两个子游戏。而且除非你非常小心，否则加入过多的子游戏就会失去玩家的关注。换句话说，如果使用比较好的小游戏，玩家就会专注于其中。当这段小游戏结束时，他们可能就会失去故事的整体线索，或者小游戏过于引人入胜，则会扰乱整个故事的流畅性。坦率地说，Pirates!中的小游戏非常简单，不会使玩家迷失自己的位置或自己的目标，也不会忘记自己该做什么。但在多年之后，我编写了一个名为Covert Action的游戏，其中有许多联系紧密的小游戏。玩家需要进入一幢建筑物，走过每个房间，投掷手榴弹并向敌人射击，开辟一条安全通道，并和所有同伴在10分钟之内通过这幢建筑物，并可以找到更多的线索，当从建筑物中出来的时候，你可能会对自己发问：“我的任务是什么来着，我在这里要做什么？”这就是一个很好的例子：我把小游戏错误地加到了整个故事之中。

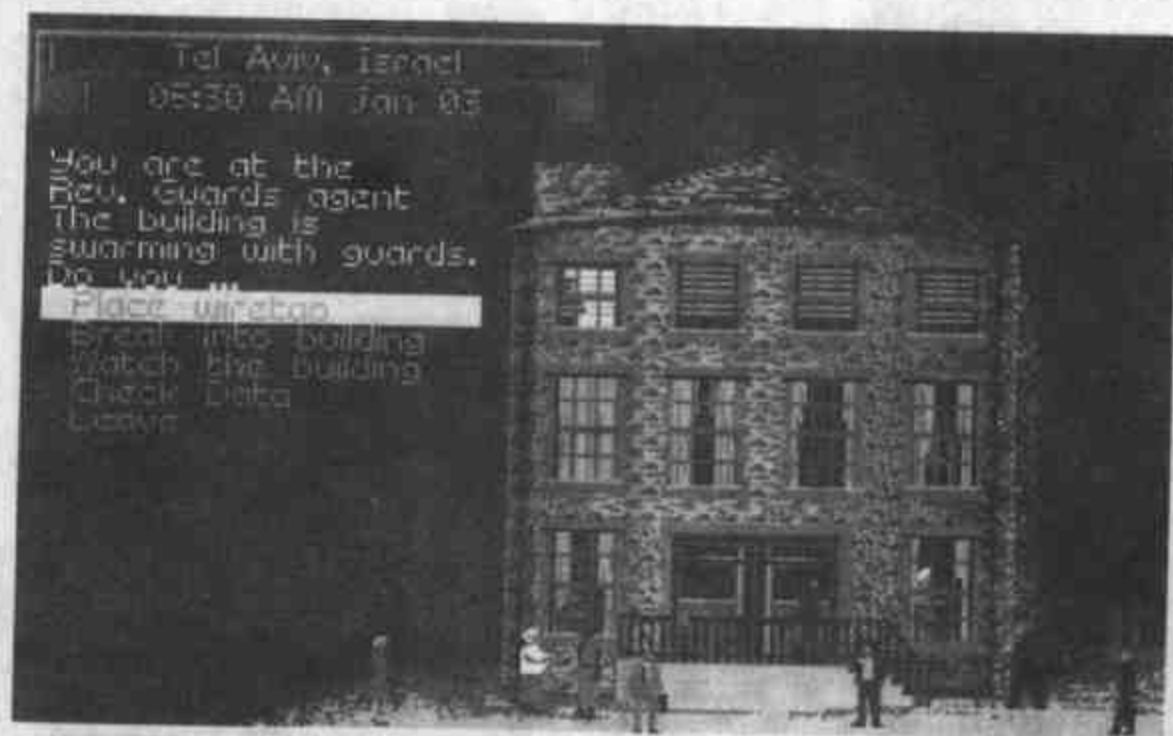


图 2.3 Covert Action

① 问：据我所知，在你自己设计的游戏中，Covert Action是你最喜欢的一个。

答：我很喜欢这个游戏，但其中有一个很特殊的问题，一系列的小游戏过于繁琐，使玩家的思维偏离了整个故事情节。游戏中讲述了一个正在酝酿中的计划，玩家需要走过一座座城市，走过一个又一个地方，尝试寻找一些线索，逐步了解计划的内容，并找到其中涉及到的人物。我认为这是一种很单纯的想法，但却很难实现。如果我能够重新设计这个游戏的话，就会对其





做出一些改动。其中有一个解码序列、流程整理，还有一些巧妙的难题。我认为整个情节中有许多简单的概念，但整个情节又不像单独的部分那样好。如果是现在设计这个游戏的话，我想我会做出很多改进。



问：Covert Action看起来与Pirates!有相似之处。你一开始曾说过“我想制作一款隐秘的侦探游戏……”

答：是的，这是游戏比较吸引人的方面。但遗憾的是，根据技术可以达到的水平，我可以更加深入各个部分的细节，这样就使我无法把精力集中在对整个游戏的理解上。



问：在Pirates!和Covert Action游戏里，玩家能够在游戏当中看到他们自己的特征，并且玩家在游戏中确实扮演一个角色。相反地，在Railroad Tycoon、Civilization或Gettysburg!的游戏中，玩家不需要扮演角色。对于游戏设计中的转变，我感到很好奇，玩家过去常常是一个具体的角色，而现在更多的是像上帝一样的人物。

答：像上帝了！？很好啊，我认为这是一个真正的规模问题，而不是一个具体游戏设计的选择需要方案。看到自己是很有趣的，在Civilization这个游戏中，我们可以看到自己的宫殿，并确实可以看到自己做的事情。另一方面，海盗看起来会很酷，而铁路业巨头看起来就不那么酷了。既然如此，我为什么还将他放到屏幕中呢？我没有想过太多，但是我想这可能更多的是规模问题。如果你穿越成百上千年的时间，成为像上帝一样的人物，做很多不同的事，但还不是很酷的角色，那么看起来就不像比较酷的角色那么有意思了。



问：是什么原因让你首先开始制作Railroad Tycoon这个游戏呢？

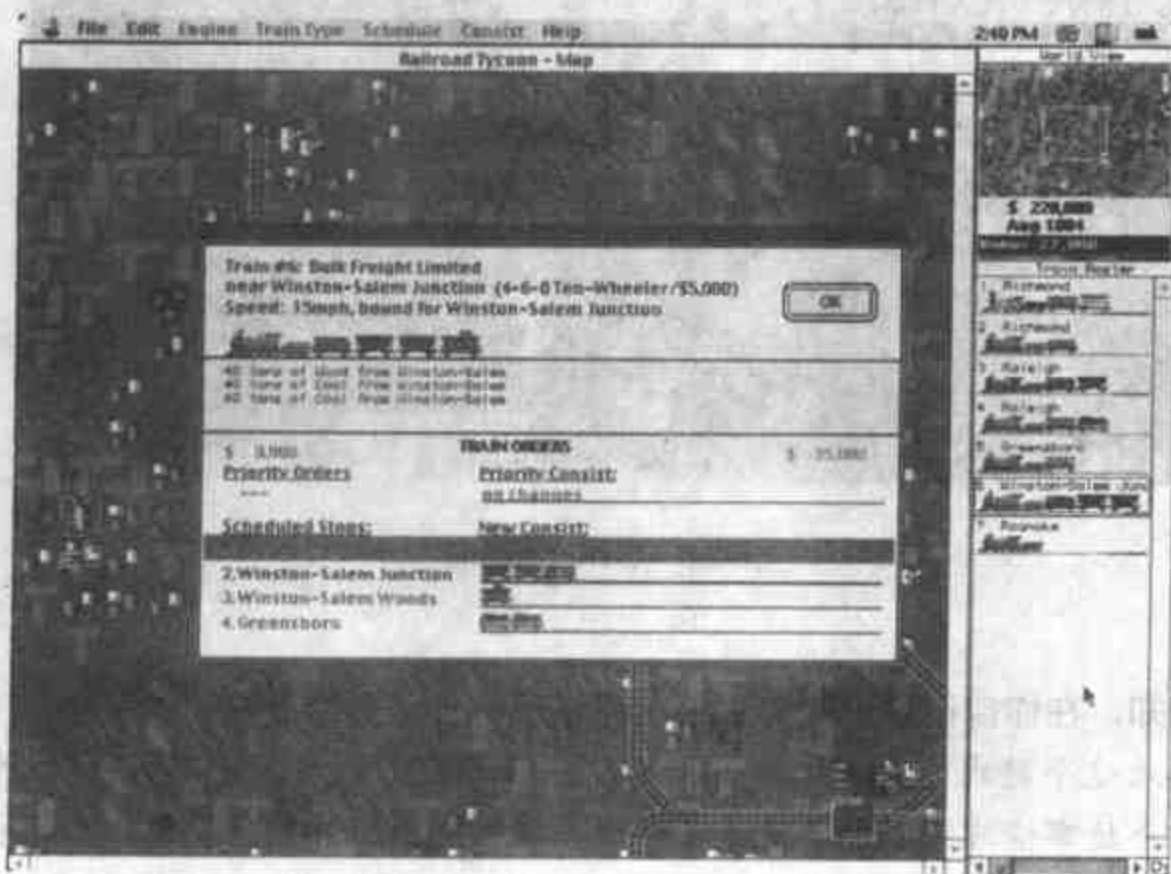


图 2.4 Railroad Tycoon



答：一开始，这个游戏是作为铁路模型游戏来开发的，没有经济方面的考虑，这个项目甚至是在低水平的情况下启动的。我们实际上要在原始的模型中转换按钮，并操作各种信号。我从 1830 这个游戏中获得了许多灵感，它是 Avalon Hill 出品的棋盘类游戏，由 Bruce Shelley 设计，我曾与他合作设计过 Railroad Tycoon。它产生了很多经济影响，尤其是对游戏股票市场的影响。正如我们补充的那样，在游戏中我们可以大显身手，从轻融开关到运行股票市场。而且我们会一直使用更新的引擎技术，并升级到柴油机。所以当你不得不做所有低水平的有关铁路的事情时，就有太多的微观管理工作需要做。所以我们把它提升一级，在转换开关和信号的时候，相关的工作人员就会出现，并自动完成相关的操作。但是如果你想操纵火车，而且有一份快件或特殊的货物或者其他的什么，你可以去那儿运这些东西，如果你想确保火车准时穿越一个桥，或桥没有了，你就不得不停下火车。但这是一个铁路运输模型游戏，随着时间的推移，我们加入了更多的策略元素。

从很多方面来讲，结合多个可以紧密联系的不同的系统方面，这的确激发了 Civilization 的灵感。经济系统、运营系统或是股票市场，所有的系统都有其重要作用，在这些系统开始进行交互的时候，也就是整个系统开始运作的时候。与 Pirates! 和 Covert Action 游戏相反，Civilization 中有可以独占计算机的子游戏。在玩家和对手比剑的时候，其他情节均会停止。这时，游戏中同时运行着一些子游戏，并与其他人交互，这种做法在 Railroad Tycoon 和后来的 Civilization 游戏中效果很好，在 Civilization 游戏中，我们考虑了军事的、政治的、经济等所有能同时发生的因素。



问：所以说在某种程度上，你仍在使用子游戏，它们一直都会在游戏中出现。

答：在 Pirates! 中使用子游戏不是偶然的。无论何时要做出决定，都要同时考虑所有的方面。那正是 Civilization 中有趣的部分。我们已经获得了这些相当简单的独立系统，军事系统、经济系统和生产系统，这些系统本身都是很容易理解的。但一旦将这些系统结合起来，情况就变得复杂了：“我得到一个机会，在那里创建一些东西，我的军队还需要一部战车，但是人们正需要一个宫殿……”这些因素一直存在于游戏中，我认为正是这些因素使游戏变得更加丰富有趣。



问：在 Railroad Tycoon 游戏当中，你设计了非常有趣的经济环境模式，同时玩家又有兴致去建造铁路，更多的是孩子都有这种想法。你认为这一点有助子游戏的成功吗？

答：的确如此。它是我所设计的第一个这种类型的游戏——从游戏的开始到游戏的结束随时都可能戏剧性的变化。在游戏的开始你什么都没有，或只有两个车站，有一段路轨，到游戏结束的时候，你会看到错综复杂的蜘蛛网般的铁路线，然后说“我做到了”。我把这个想法再一次用到了 Civilization 游戏中。游戏开始时，只有一个定居者和一小块属于他的土地，在游戏结束的时候，这个定居者创造了整个文明社会发展的故事，你可能回头看看然后说，“那就是我，我做到了”。游戏的状态从开始到结束发生了如此戏剧性的变化，这是一种在其他地方曾经有过的感觉。在 Pirates! 和其他一些游戏中，玩家可以挣分或完成某些事情，但从未有过使世界发生天翻地覆的变化这种感觉。我认为这要归功于 SimCity，因为这个游戏成功地创造了这种感觉。



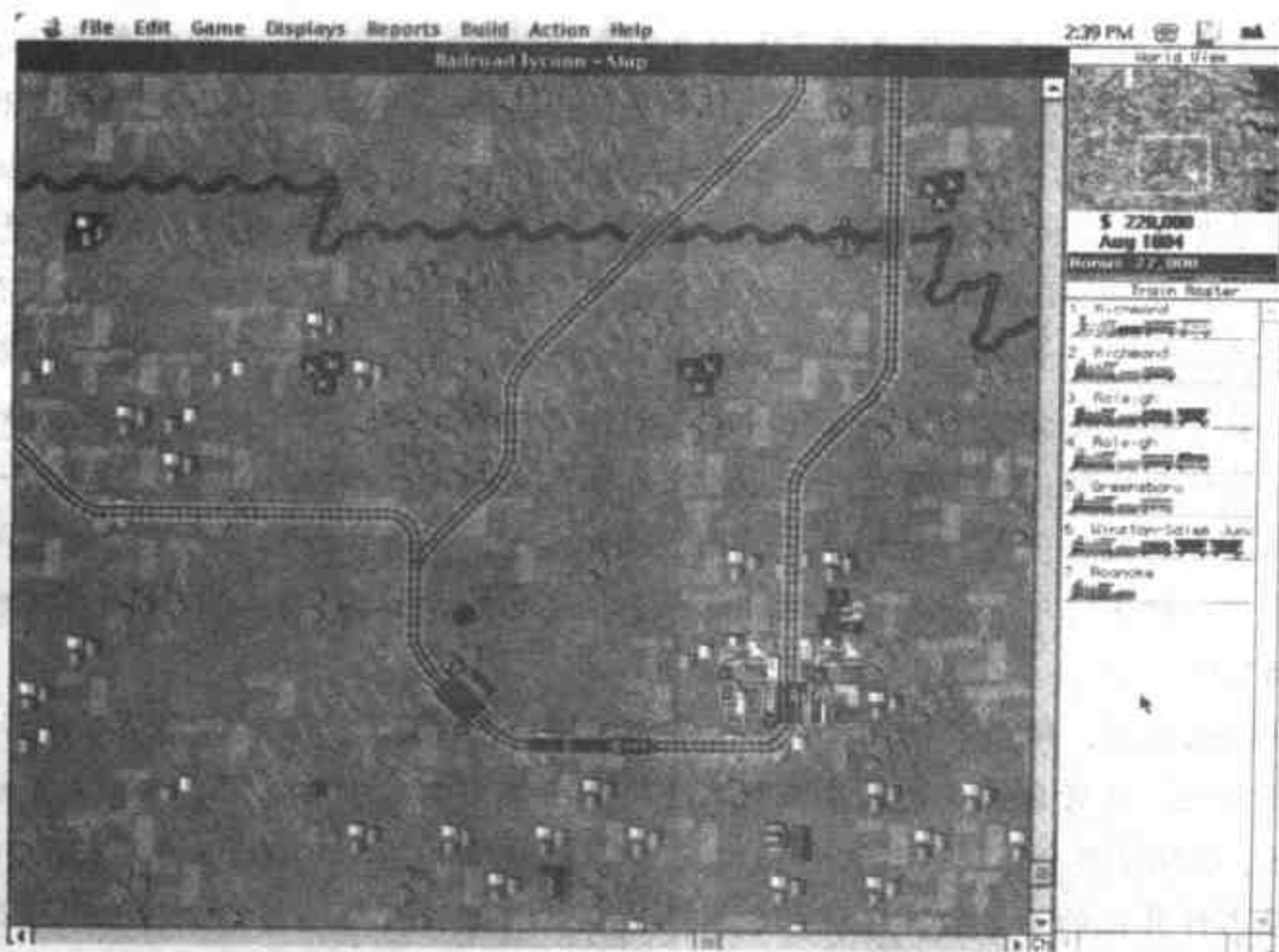


图 2.5 Railroad Tycoon



问：当你在制作电脑游戏时，是否受到了Avalon Hill公司棋盘类游戏Civilization的启发？

答：我们的确是玩过这个游戏，对它也很熟悉，但说实话，比起 Empire 或 SimCity，这个游戏的确少了点灵感。我想这主要是由于棋盘类游戏的限制。其中有一些单纯的想法，但在 Civilization 游戏中也有一些很好的概念，其中的探索过程和多个系统的同时运行，都是很难在棋盘类游戏中实现的。在游戏中有一些单纯的想法，所以我很喜欢这个名字[笑]。但是对于那些比较实际的概念，却主要是来自于其他来源，而不是 Civilization。



问：你的很多游戏看起来都从棋盘类游戏中得到了启发。但正如你刚刚说的，Civilization 不能成为棋盘类游戏。你是如何把在棋盘类游戏中所喜欢的部分转移到计算机游戏中，而不是直接转换过去？

答：在计算机出现之前，我玩过很多棋盘类游戏，并深入研究过 Avalon Hill 公司出品的游戏。我想他们提供了很多关于游戏的最初想法。对于游戏的重要部分、吸引人的部分和与主题无关的部分，这些游戏提供了一个很好的模型。但在机制和界面等方面，棋盘类游戏和电脑游戏存在着很大的区别。在棋盘类游戏中有很多有趣的研究资料。通常都是关于“我们需要一些技术”或“我们需要考虑哪一部分”之类的内容。就基本的游戏组成部分来说，其中会有某种重叠。但在使用方法等方面，棋盘类游戏和计算机游戏有相当大的区别。我想说的是，棋盘类游戏提供了对现有的和有趣主题的回溯。但在我看来，一旦把它应用于实际游戏中，计算机游戏和棋盘类游戏间有很大的不同。



问: Civilization最引人注意的是其让人着迷的品质。我想知道这只是碰上了运气,还是你从一开始就有了整体计划。

答: 我没有想像到这一点。我们想让所有的游戏都充满乐趣,并且希望它们可以在某种程度上使人痴迷。但是 Civilization 有一种神奇的吸引力,这种吸引力并不是出自设计,我们也并没有预料到这一点。我认为,任何以自然的方式结合在一起的事物都具备这种品质。我认为这个游戏的成功是其中各个部分协调一致的结果,我们选择了正确的比例、比较合适的复杂程度,所要做的事情在数量上也恰到好处。在设计这个游戏的时候,我们做出了明智的选择。所有这些选择组合在一起就形成了这种吸引力,并且我认为它是一个很不错的主题。游戏中包含了许多因素,并且所有这些因素组合在一起就产生了这种令人着迷的品质。虽然在设计的时候并没有故意地加入这些品质,但我们却了解这种品质的存在。大约在这个游戏开发到一半进程的时候我们才发现,原来其中充满了许多乐趣。对我们来说,这是个非常令人愉快的发现。

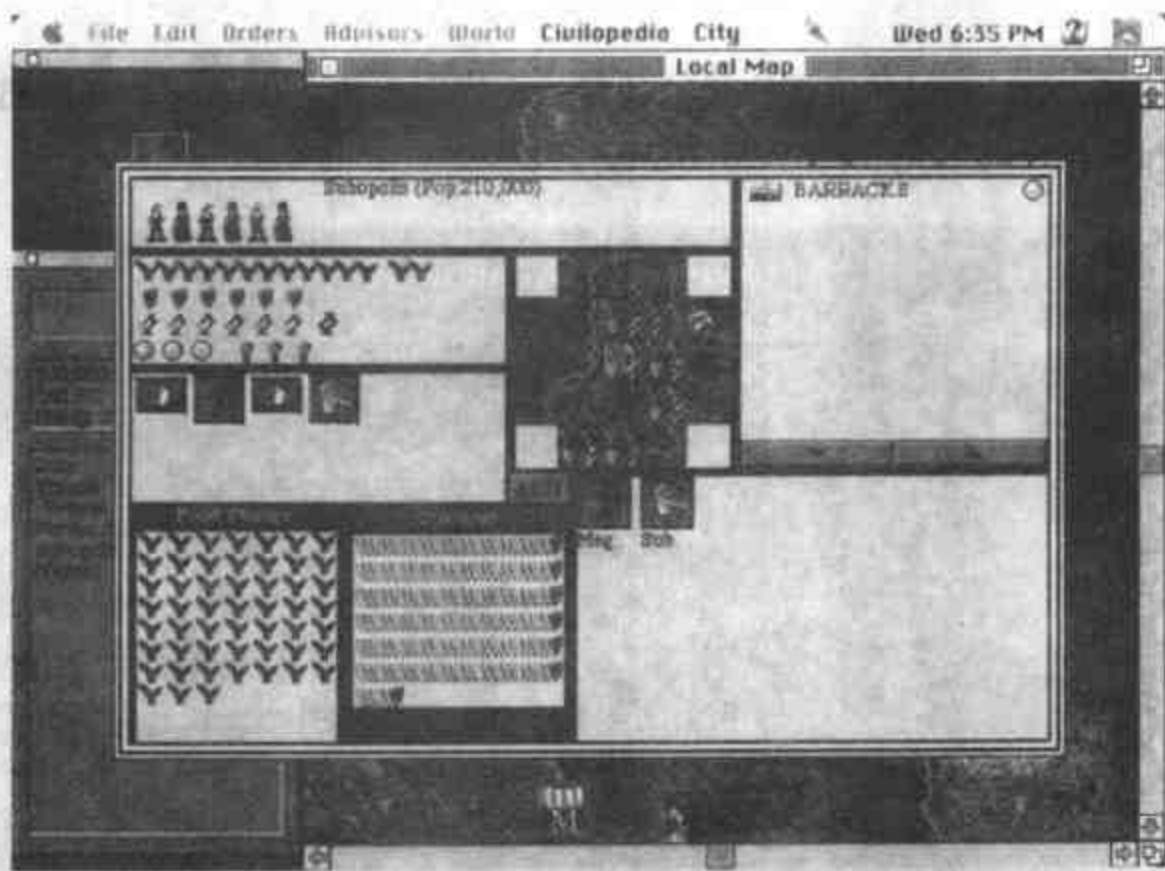


图 2.6 Civilization

问: 你能否为其他游戏设计者提供一些建议,怎样做才能在游戏中加入这种吸引人的品质呢?

答: 我们是事后才知道,或者我们认为自己知道游戏如此令人痴迷的原因,也可能有我们自己的道理。其中一个因素就是“吸引人的选择”。对于我们来说,这就表示你可以得到多种选择机会,其中的选择并不复杂,基本上都可以从某个选择列表中随机选择。对于过于复杂的选择,你有可能会说:“哦,我已经有了这三个选择。是的,我可以花费 5 分钟时间来分析情况,但我的确需要继续游戏的设计,所以我要选择 B,因为这个选择看起来很不错”。而在另一个极端,又有一些过于简单的选择:“很明显我必须选择 A,因为它显然比其他选择要好”。在 Civilization 这个游戏中,我们试图提供一些判断起来很容易的选择,但是在某种特定的情况





下，你可以选择 A。而在某种稍有变化的情况中，B 是一种比较好的选择，在另一种情况中，C 也是个不错的选择。所以你就会说：“下一步我可以采用三种技术”。并且你会对自己说：“好的，现在我要与那些到处侵略的罗马人作战。所以我需要使用某种技术，使我可以制造更好的军事单元。但自制地图好像也挺有意思。下一次我要采用这种技术进行开发”。如果你做出的某种决定使得玩家总是这样说：“下次我要试试那种技术，因为它看起来也很有趣”，这样就形成了一种观念——玩家总有比较多的选择可以挑选，但也仅仅是浅尝辄止而已。

我认为，这种令人着迷的品质还源自于我们未让多件事情同时进行。一方面，在游戏中已经使用了下一种技术。你的科学家正在研究那种技术。这个城市也正在制造第一辆坦克，这正是你想要的。这里有一个军事单位正在相邻的大陆上巡逻，并且很快他就会找到感兴趣的东西。在游戏中，你已经得到了自己想要的许多东西，但决不会得到所有想要的东西。游戏中也不会有重置状态。在游戏中总有两三件事情同时发生，并且你也希望这些事情可以完成。所以游戏过程一直都在进行。我认为这样就产生了“你从未停止过玩游戏”的感觉。



问：我了解到 Gettysburg! 并不是你设计的第一个实时游戏，但看起来好像部分受到了经典的实时游戏 Command & Conquer 和 WarCraft 的启发。

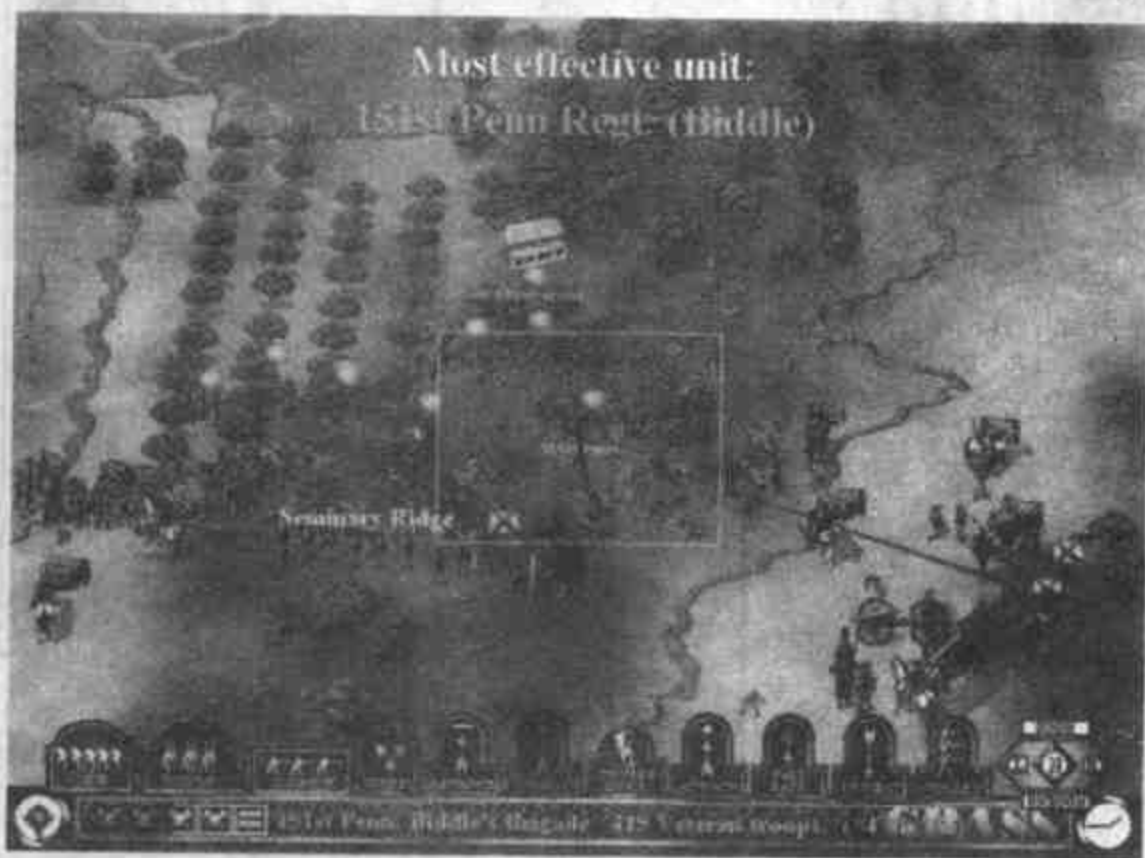


图 2.7 Gettysburg!

答：我认为从技术的角度来讲，完全可以让许多人物实时地在屏幕上进行他们的工作。我们所说的“实时”是一个奇怪的术语，虽然我们一直制作的都是实时游戏，但我们并不认为游戏是实时的，因为它们仅仅是看起来很自然而已。但我猜想当基于回合制的游戏有其自己的风格时，我们就不得不对其加以区分了。我认为 Gettysburg! 是自己很久以来一直梦想设计的游戏，但直到最近，在技术上才可以达到所需要的水平。在目前可以达到的技术水平上，我们可以让多个士兵在近似真实的战场上进军，为步枪装弹、射击，或组成各种阵形，并且可以做出我们在 Civil War 战争中能想像到的任何事情。在技术上，这些事情已经都可以实现了。



问：技术使WarCraft和其他的实时游戏表现得很好，但是又添加了更深层次的模拟，其中还加入了侧面包抄和其他传统战争游戏的特点。你的目的是制作一款更为复杂的游戏，然后加入快速的实时模式？

答：我的想法是制作一款类似于 Gettysburg 的战争游戏，并且最好是具有“实时”的风格。当我在玩其他一些棋盘类游戏的时候，总感觉到缺少了什么。当我在玩棋盘类游戏或 Gettysburg 的时候，并没有找到阅读历史和战争故事的感觉。对于时间的压力、混乱的感觉，还有不同阵型的含义等因素，在我们自己做出选择之前，对它们丝毫没有任何概念。然后你突然意识到：“小伙子，想简单地操纵几下就赢得战斗不是那么容易的，要是有人成功过就好了”。或者说“现在我知道他们为什么排列成这些阵形，我以前一直以为这样做很愚蠢”。当战斗在进行的时候，许多事情都显得越来越明显。其中加入的想法包括侧面包抄、鼓舞士气等在 Civil War 中所使用的策略。加入了这些因素后，可以找到 Civil War 中战斗的感觉，而不是向玩家灌输生硬的战争游戏概念。通过这些影响战斗或战略的关键因素，你可以很自然地学习如何成为一名指挥官。虽然不必遵守一系列的规则，但要意识到，“喔，如果我能给这些人一些支持和鼓励，他们就能成为更好的战士。如果我从侧面进军，攻击就可以成功”。你可以经历一个学习的过程，而不是让他人来教你如何成为一名好将军。你可以利用这种方式不断地学习。这就是我们的目的所在。

问：我很奇怪，你为什么让用户以拖放方式部署军队，这与其他实时游戏使用的方式不同。虽然这并不是标准系统，但使用这个系统是不是因为你认为这个系统更好？

答：我不能肯定现在自己是否还会使用这种方式。我认为拖放方式很方便，可以将要控制的目标用光标选中，并拖动它们沿着我们所选定的路径前进。其实我也很愿意使用标准界面，所以如果是现在的话，我更有可能使用标准实时界面。使用拖放方式已经是很久以前的事情了。WarCraft 没有使用这种方式，但 StartCraft 采用了这种方式，Age of Empires 就在这个时候发布了，那时接口标准并没有统一。为此，我们为玩家提供了一种选择机会，使他们可以选择右击或左击操作方式。但是如果是现在来做同样的事情，我就会使用标准的实时方法，并生成拖放的选项。

问：与Railroad Tycoon或Civilization相比，Gettysburg!的各个场景之间是独立的，玩家经过努力后，战斗就结束了，得到一个简报，并重新部署军队。你为什么选择这样的游戏过程？

答：我这样做是因为 Gettysburg 中笨拙的战斗需要许多战斗单位！【笑】我更喜欢与真正的战斗相近的完整战斗。基本上来说，如果要使游戏充满乐趣，我发现玩家希望至少可以调动 10 至 15 个独立的战斗单位。遗憾的是，整个战斗中有 70 到 80 个连队，所以玩家对它们完全失去了控制。我们试过改造规模，并且说：“虽然军队看起来好像是一些小组，但实际上是有编制的，你可以控制整个战斗。”但以这种方式编制军队的规模并不正确，因此我们有了下述说法：“是的，大多数刺激的战斗使用的都是这种规模的编制。”这就表示这种编制是战斗的一部分。我们必须接受这种方法、使用这种方法，并且将这种方法发挥到最佳效果，并且我认为





场景系统就是这种做法的一种尝试。



图 2.8 Gettysburg!

我认为，在现实的世界中，我可能会选择 Battle of Hunter's Run（亨特战役）或者其他仅使用三个连队的战斗，这样的话，在一个场景中就可以控制所有的部队。但是没人愿意买 The Battle of Hunter's Run，他们都想买 Gettysburg!。所以大规模的战役是历史中不太适宜的部分。我认为它运行得相当好。但是当人们这样说时我也会理解：“我想要整个战役。”我也在尽力做到这一点，并向他们表明在这个系统中实际上不需要这样做。这是历史和现实没有为游戏创造理想情形的一种情况。但我们的感觉是，并没有提供整个战役，而提供 80 个战斗单元，玩家很难控制，也没有将整个战斗都放在一个场景中，使玩家玩游戏的时候感到费解，所以既让游戏好玩，又使其具有历史真实性是最好的“折中办法”。我认为游戏中有一些比较吸引人的场景。Civil War 更倾向于核心内容，虽然它很吸引人，但不能称其为 Civilization。



问：你一直致力于创作恐龙为主体的游戏，你在整个方面的目标是什么？


答：制作这类游戏的目标与我制作的其他游戏的目标相同——找到游戏主体中吸引人、独特且有趣的部分，并将其变成电脑游戏。很久以来我就认为恐龙是个很有趣的主题，而且我认为应在电脑上实现这个主题。我尝试过将科学上和历史上证明是真实且准确的内容放在游戏中，但并没有什么指导意义，最好让玩家用自己的知识来玩游戏。许多人都了解关于恐龙或历史的某些知识，如果他们可以将这些知识用到游戏中，就会增加许多乐趣，玩家自己也会感觉很好。玩家并不需要查看手册就能懂得怎样玩游戏，因为他们自己了解恐龙的知识。他们意识到拥有火枪和车轮这样的东西很有趣。

同样，人们知道 T.rex 是最坏的恐龙。所以，我们会在游戏中使用某种方法让人们了解关于主题的基本内容。我们尝试在游戏中加入一定的现实性和精确性。然后在这基础之上使游戏充满乐趣。电影使用相同的方式提供所有的娱乐性、一系列的行动以及故事的所有重要部分，然





后迅速略过枯燥乏味的情节。我认为游戏也有同样的责任，它可以把我们带入关键点，然后转向下一件有趣的事情。我们正在尝试在恐龙游戏中使用相同的方式，将其带入现实生活，展现其有趣和独特的方面，去除乏味的部分。目前我们正处于设计阶段，对于其中的细节并没有什么可说的；但愿过几个月我们能够更详细地介绍该游戏的内容。

 问：据报道，从80年代开始，你已经有很多的制作游戏的经验了。我想知道你认为整个产业在何时发生了转变？


答：我认为在游戏的质量上有总体的、全面的提高。那时出现了一些很不错的游戏。我很喜欢 StarCraft、Age of Empires、Diablo，我认为 The Sims 这个游戏很有意思，并且 RollerCoaster Tycoon 也很不错。所以我认为那些游戏比以往的游戏更受欢迎。这些游戏比我们 5 年前或 10 年前制作的游戏更为精彩。玩家可以看到，在外观、图形、视频和其他所有方面，这些游戏都有很大的改进。而游戏的核心部分——游戏的设计过程进展却相当慢。就游戏的设计来说，我认为诸如 Pirates!、SimCity 和 Civilization 之类的游戏的确很出色。即使对于现在的设计水平，这些游戏的设计也是相当出色的。这些游戏并没有随着时间的推移而黯然失色。所以就游戏的设计来说，法则上所说的质量每年都会提高一倍，处理器的速度每年都会提高一倍，对于这种说法我并不认同。偶尔也会出现某种新的流行想法，使我们意识到某些新近可能发生的事情。我认为这种情况会延续下去，而我认为有意思的是，一台三年前的电脑已经完全过时，但三年前的游戏却仍可能很好玩。

 问：只要你能让它运行……

答：是的，只要能运行三年前的电脑运行这个游戏就行。我认为它们在发展速度上有所不同。技术的发展速度较快，而游戏的发展速度却相对较慢。

 问：你认为游戏的发展速度比产业的发展慢了吗？

答：我并没有发现有什么重大的变化。其中有一个现象就是：我们只记得过去的好游戏。过去的游戏好像都很优秀，现在的游戏有些很出色，但更多的游戏是粗制滥造出来的。以前也有一些比较差劲的游戏，但随着时间的推移已经被人们遗忘了。目前的许多游戏设计工作做的也很出色。在游戏设计方面有许多还未开发的领域，我们还有机会可以进行创新。随着网络技术的发展，又出现了许多可以进行创新的新领域。所以在现在的优秀作品和多年前出现的优秀作品之中，我并没有看出什么实质性的区别。

 问：你以前在Microprose和Firaxis这种小开发工作室中工作过，后来又在规模壮大之后的Microprose中工作过。你觉得哪种环境更利于你创造出好游戏？

答：我个人觉得在小一点的环境中更舒服。与其他可以造就好游戏的规律相比，这纯属是个人感觉。我认为从趋势上来讲，组织和团队都是趋向于越来越大——这也是事物的发展规律。如果有什么因素使我感觉到自己优点老化，那可能就是因为当我在比较大的组织环境中工作时，





感觉并没有其他的开发者那样好。我认为在那种模式中成长起来的开发者对大项目会感到更加适应。我在洛杉矶参加 E3 展览会，Hall of Fame 大奖的得主是 Hironobu Sakaguchi，他设计了 Final Fantasy，这是一个剧情非常复杂的游戏。对我来说，处理如此复杂的剧情简直是难以想像的。但有些设计者却非常擅长于此。这是设计者的个人偏好，我想这可能是因为我开始设计游戏的时候，还没有如此庞大的团队，并且我喜欢团队，而有些人会喜欢大项目。

问：从你进行游戏开发的时候开始，开发团队的规模从一两个人已经发展到了现在的20人以上。你认为这会使游戏缺少个性化吗？

答：我想是的，但当今的游戏仍有个人影响力。而且我认为那些有个人影响力的游戏都很优秀。有很多并无什么乐趣可言的游戏都是“由委员会来设计，由一大批人来编程”。这些游戏看起来不错，但都带有其他优秀游戏的痕迹，而缺少个性。对我来说，RollerCoaster Tycoon 是这类有个人影响力游戏的典范。人们都认为这个游戏很出色。没有人会说：“这个游戏真愚蠢”。有很多想法是很巧妙的，但如果你将这些想法提交给委员会，他们就会说：“人们会不会认为这个主意很愚蠢？”即使 Final Fantasy 这种由庞大的开发团队开发出来的游戏，也极具个人想像力。如果能在一个规模较大的团队中开发这类游戏，将是一个好主意。但我认为很难让大团队完全理解个人的想像力，而且也很难不受到委员会的影响。

问：在项目中，你既是首席编程人员，又是首席设计人员。同时充当这两个角色，你觉得自己工作得很快乐吗？

答：如果不这样，我根本无法想像自己的工作会是什么样。对我来说，最好就是能想出一个主意，然后立刻敲入计算机，而不是向其他人解释然后再看结果。而且这也是我的个人风格，也是最具效率的工作方式。

问：在其他公司的大多数项目中，都会有一个人担任首席设计师，一个人担任首席编程人员，他们都非常忙。如果要同时充当两个角色，可能很难承受。

答：我认为他们需要花费一半的时间相互交流，而不需要这样做。取消了所有这些交流，可以提高效率。当然，两种方式都可行，但我个人偏好是设计者兼任程序员的方式。

问：现在你在一个较大的团队中工作，你如何将你的游戏设计创意传授给团队中的成员。让他们也能对这个项目感到兴奋？

答：我们使用的主要工具是原型法 (Prototype)。在我们的开发过程中，一个人同时充当程序员和设计者的优点就在于：在一周到两周的时间内，我们就可以拼凑一个游戏的原型。这样就可以向人们介绍相关的游戏概念、它的工作方式以及公用参数。此外，如果大多数人对我们所介绍的历史性和科学性主题有所了解，他们就会了解关于这些主题的内容。然后他们就会开始讨论，说这就是我们要玩的游戏，其中还有三四个吸引人的趣事。将这些情节放在一起就可以让人们了解游戏的发展方向。此时可能并不需要人们了解整个故事情节，但是就精彩的艺术





设计、声效和美妙的想法来说，却要让他们了解自己的感受，并提出一些我们没有想到的问题。不要让人以为事情已经做完了，因为我们需要让人们感觉到他们可以做出自己的贡献，而这些已经超出了我们的想像。

问：如果有人提供了一些比较好的想法，并加入到游戏中，即使没有想到过这些内容，你是否也会很乐意将这些想法加入到游戏中。

答：我很乐于“窃取”这些想法，并在几年之后声称这些都是我自己的想法。【笑】

问：使用你的“原型系统”，你是否曾经遇到过没能达到预期效果的游戏？

答：是的，在我的硬盘上保存的许多游戏都属于这种类型。许多游戏都是在不同的方向上进行开发的。例如 Civilization 原本就与 SimCity 非常类似，将某个区域划分为农场，在这里建造一座城市，并看着它发展起来。这个游戏最初是要让玩家作为旁观者看着城市的发展，而几个月之后它又变成了基于回合制的游戏。我提到过 Railroad Tycoon 开始是一个铁路模型游戏。有许多时候，原型要经过很多次修改才能开始实施。这是原型的整体概念：它可以告诉我们某种想法是否可行，告诉我们其中存在的问题。它会让我们首先集中在最为突出的问题上，对其进行修正。

问：你设计的游戏很容易掌握，也很容易玩。但同时这些游戏又非常深奥和吸引人。你是怎样同时满足这两点的？

答：易玩的部分很容易理解。我认为界面规范和人们所熟悉的主题可以帮助玩家了解游戏，因为玩家对自己所要做的事情已经有所了解。在游戏开始的时候就要给玩家正确的反馈信息，让他们知道自己做事的方式是正确的。在 Civilization 中，人们以非常快的速度建造宫殿，人口

达到一个里程碑式的数字时，第一个城市就建成了。在游戏的早期阶段，我们就让玩家知道他们做事的方式是正确的，对于他们所做的每一件事，电脑都予以承认和肯定，并且认为玩家做得很出色。这样可以使玩家可以全身心地投入到游戏当中。

至于游戏的深度，那是因为我们玩过游戏。原型的另外一个优点就是，如果要花费两年的时



图 2.9 Civilization





间来编写一个游戏，就要花1年零11个月的时间来玩游戏。很快你就会对游戏的开始部分感到厌倦。从某种意义上讲，在游戏中加入深度可以让自己保持编写游戏的兴趣。如果在一个场景中有20或40小时的游戏剧情，那是因为我们在这类场景中玩过了20或40小时，并且发现，在大约20小时之后，游戏就变得有点乏味了，我们必须提出一种新的概念，使这个游戏可以更加有趣，也更为复杂。所以游戏中有许多深奥之处，完全是因为我们已经玩了很长时间。



问：你认为使用原型也有助于平衡吗？

答：确实，原型有助于平衡。如果你玩游戏的时间足够长，它还有助于编写AI（人工智能）程序，这样的话，你就可以真正理解哪些是好战略，哪些是糟糕战略。玩游戏的经验还有助于编写AI，对增加游戏的深度也有好处。但其中的危险就是，可能会忽略初级玩家。这就是之所以我们要在游戏开发结束时进行测试的原因。并且我们会说：“这就是我们对游戏提出的看法，尝试玩游戏。”而且我们也总是会发现这些初级玩家还是不能玩。游戏中有太多精彩的内容。所以我们会说：“好的，你在哪里遇到了困难？”我们无法发现游戏在那些方面的问题，但游戏中既需要深度又需要易玩性，两者都非常重要。



问：你的游戏都是根据历史或真实生活中的事件制作的，而许多游戏都是一些幻想或者科学小说。这是因为你想创造一个玩家已经有所熟悉的游戏世界吗？

答：我认为这很重要。当可以在游戏中应用自己的知识时，这一点显得尤其重要。这样会使玩家对自己更有自信，并且也有一种积极的作用。就历史和现实情况来说，我觉得自己有很多工作要做。在我成长的年代没有“中世纪”，没有科幻影片，也没有“星球大战”。现在的小孩子觉得这些事情就像历史一样真实。对于许多小孩子来说，太空船、魔法师和术士就像飞机、潜水艇之类的东西一样真实。这些东西都是经过改进的，但在我看来它们就像飞机和潜水艇，Civil War 和 Roman Empire 同样吸引人，并且我会尝试将这些东西制作在游戏中。



问：我很奇怪你是如何在历史事实和游戏之间取得平衡的。Gettysburg! 看起来就是这样的——一个游戏：你将游戏分为若干个场景，既保持了历史的准确性，又使游戏非常好玩。

答：在那个游戏中我们必须服从历史事实。从常规上来讲，如果趣味性和历史之间发生了冲突，我们就要服从趣味性。玩家可以从历史的角度来判断游戏的结果。幸运的是，与历史的准确性相比，我们的游戏几乎总是向着有利的一面发展。在 Gettysburg! 中，我们遇到了无法编造故事情节的情况，而且还要尽可能多地编造这种情况。在其他许多情况中，我们会在游戏中提出自己认为可行的想法，然后找到历史先例或者历史性的解释来验证。我们不会尝试或者盲目地保持历史的准确性。基本的原则是，我们要有一定的灵活性，创造自己的历史，而不仅仅是遵循历史的轨迹。即使某些情节是比较具有现实性的，也可以使这些情节在完全不同的方向上发展，很可能是根本没有发生过的。当然，Civilization 游戏中比较有趣的情节就是 Zulus 或 Mongols 可以称霸世界。任何人都可以“称霸世界”，最后赢得胜利的并不一定是某个国家的人，玩家也不会做历史的奴隶。





图2.10 Gettysburg!

问：你在开发飞行模拟器的时候，在图形方面所采用的技术并不是最领先的。你认为这个选择是明智的吗？

答：正如我所说的，在制作原型的过程中，即便是在最后一刻，情况都有可能发生改变。许多领先的技术都需要从游戏开发之初就开始研究，而且在技术上的投入也很大。假设在我们的工作中情况经常发生变化，实际上就不太可能将原型法和先进的技术融合到一起。除非我们知道自己的目的，或者知道自己要达到的技术水准，否则就无法启动研究项目。在一开始制作游戏的时候，我们并不了解这些情况，而且也不想在游戏设计方面让自己受到束缚。在这方面我曾考虑过很多次。如果使用第三人称 3D 视角可以看到一定量的场景，然后发现如果能看到更多内容的话就会使游戏更有趣，那么就需要对于位置或其他情况更多的信息，这时就会有些为难。

在图形的功能性和外观之间总是存在矛盾的。有些游戏虽然看起来很漂亮，但是却没有提供玩家所需的信息，或者游戏的条理并不是很清晰。这种追求外观而牺牲功能的做法并不可取。我们会尽可能地让游戏看起来很漂亮。但在所有优秀的游戏中，漂亮的图像都是在想像中出现的，而不是出现在屏幕上。如果在某个小镇中有人说“我们喜欢国王日”，如果你真正地投入到游戏中，那么在脑海里就会产生一个很令人兴奋的画面，而无法将其呈现在屏幕上。反之亦然，如果你没有投入到游戏中，就会将屏幕上出现的内容分离开来。我们的目标就是，让玩家融入到游戏当中，并且根据我们在屏幕上提供的暗示，在脑海中产生那些精彩的画面。

问：你是第一个将自己的名字印刷在游戏包装盒子的游戏设计者，我很感兴趣其由来。

答：这要追溯到 Pirates!。那是我第一个在上面加上名字的游戏。那时我在 Microprose 工作，我的搭档是 Bill Stealey，他负责商务/市场方面的事物，我负责开发/创作工作。在 Pirates! 之前



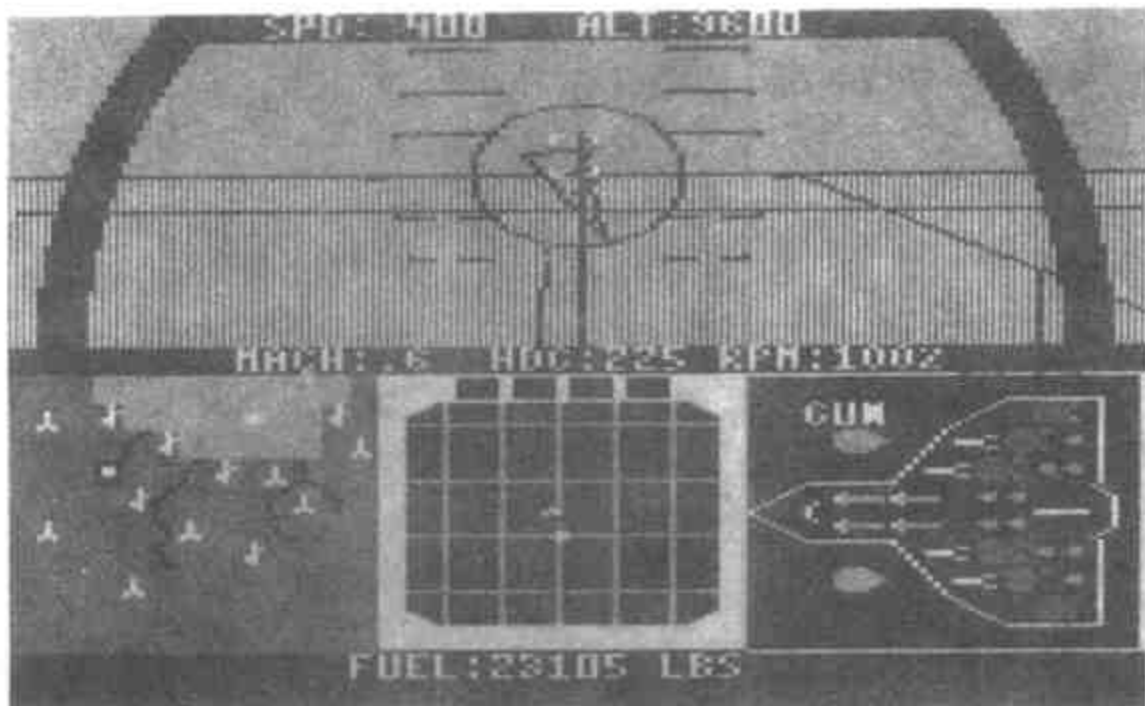


图 2.11 F-15 Strike Eagle

的游戏就是飞行模拟器游戏，我对 Bill 说：“我要在这个游戏中加入海盗。”他说：“海盗？等等，海盗是没有飞机的，你不能这样做。”“我认为这肯定是个很不错的游戏。”他回答说：“那好吧，谁会买海盗游戏呢？如果把你的名字放在游戏的包装盒上，他们会知道这个游戏会类似于 F-15 之类的游戏，也许他们会试一下。”发布这个海盗游戏有很实际的问题，没有人会

对它感兴趣，谁会对海盗游戏感兴趣？玩家需要的是飞行模拟器。所以可以说：“好吧，你需要飞行模拟器，但你可能会有兴趣尝试一下这个海盗游戏，因为这个游戏的作者就是编写飞行模拟器的那个人。”我想这可能是一种原始形式的商标。这是因为我们在游戏的类型上做出如此之大的转变，并且保持游戏之间的联系。



问：所以这并不是你对名声的渴望？

答：【笑】不，不是。即使是现在，名声也和电脑游戏无关。我认为这样很好。它仍是非个人化趋向的事业。我认为人们会记得优秀的游戏，而且会在一定程度上记住其中的人物。但 Robin Williams（一位非常出色的电脑艺术设计大师——编辑注）却没有崇拜者，你知道，电影明星们都是有一群崇拜者。我认为这样很好。而一旦我们了解到由于自己的身份而可以避免所有的事情，那么情况就不太好了。



问：但就是这种信心使你制作出了 Pirates!，不是吗？

答：【笑】是的，这是个好游戏。如果这个游戏不好的话，这种策略就不灵了。



问：你的许多游戏都产生了不同的反响，但在这些游戏中，你从未担任过首席设计者，这是为什么呢？

答：我认为他们已经做得很出色了。我很少重新去制作某个主题的游戏，主要是因为我的创意还是源源不断的，所以我宁愿制作恐龙游戏，而不愿意去制作旧主题的游戏。我并没有过多的精力去关心玩家对游戏的反映。有些游戏反映很好，但也有一些并不是很好。与其让某个主题让人逐渐淡忘，还不如让它有个结果的好。在现实生活中，我想参与到所有的事情中去，但实际上我并不可能那样做。所以我更有兴趣参与到新游戏中去，而不是等待结果。如果玩家



需要下一个版本的 Railroad Tycoon 或者 Civilization 的话，那当然是让人兴奋的。能做到这一点我也很高兴。在 Civilization III 中，因为这个游戏要在 Firaxis 公司完成，所以我可以直接加入自己认为比较好的想法。如果我有时间的话，我可能会制作新版本的 Railroad Tycoon II 和 Pirates!。但我并没有时间来�做这些事。



问：我听说很多人在谈论游戏中的故事陈述。他们所说的故事陈述通常指的是使用片头动画或者分支对话树之类的东西。你的游戏从未过多地考虑过故事陈述问题。



图 2.12 Civilization

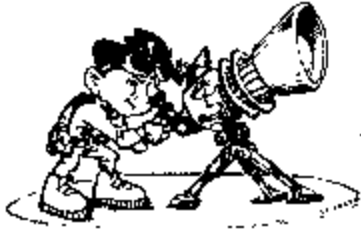
答：Civilization 对我来说是个史诗性的故事。我认为自己感兴趣的游3戏都是关于玩家的，而不是关于设计者的。有些玩家常满足于自己做次要的决定，而让设计者做主要的决定。但是我觉得如果反过来的话，游戏就会变得更有意思。从某种意义上来讲，我认为游戏都是在讲述故事。在我看来，创造故事更多的是玩家，而不是设计者。在 Civilization 中，我觉得在每个游戏中都有奇妙的故事，它们已经超越了传统意义上的故事。在我们关于游戏设计的规则中，游戏可以分为三类：分别是设计者可以获得所有乐趣的游戏、电脑可以获得所有乐趣的游戏，还有玩家可以获得所有乐趣的游戏。我们认为自己制作的游戏应该让玩家获得所有的乐趣。而且我认为，游戏可以趋向于让设计者获得所有的乐趣或者至少大部分的乐趣，让玩家做出一些次要的决定，但也只是一些顺从设计者做出的决定。这种做法并不是很糟，但我们的原则是，尽可能地让玩家可以做出更多的决定。



问：虽然在 Gettysburg! 游戏中有多人选项，但从总体上来说，这个游戏一直是单人模式的。你对多人游戏有何看法？

答：我一直认为应该多人游戏中加入更多的内容，因为游戏也可以是多个玩家的。但我自己仍比较喜欢单人游戏，所以我并不肯定何时会达到这个目的，或者使用何种方式。从我开始





制作电脑游戏以来，联机多人游戏可能是惟一在技术上的革命性发展。其他的所有方面都非常具有革命性——更好的图形、更快的速度、更多的内存等等。但多人联机游戏在我们用于制作游戏的工具中具有革命性的变革。我对在这个方向上发展非常感兴趣，但并不确定现在的游戏将来是什么样子的。



问：在以前的“Next Generation”杂志的采访中你曾经说道：“游戏会在整个世界流行。这要花费一些时间，但电脑游戏比其他的娱乐方式更加具有吸引力。”棋盘类游戏固然流行了很长时间，但并没有在全世界流行。我不知道你认为电脑游戏中有什么可以让你觉得它如此具有吸引力。

答：是的，我坚持这种观点。我认为是互动性使电脑游戏如此特别。它们可以像玩家一样与人单独地交流，这与电影、电视或者音乐这样的娱乐形式完全不同。当然，看电视的时候拿着遥控器，从一个频道转换到另一个频道也是一种互动……【笑】。但是电脑游戏的互动性却截然不同，这种差别使电脑游戏非常吸引人。现在我们仍在学习如何使用这个工具，在许多方面都不如电视和电影这类的娱乐方式运用的好。但我认为要去了解这个工具的优点，了解它比其他的娱乐方式所有的优势。

我认为棋盘类游戏也有互动性，但在游戏中需要对手。电脑具备棋盘类游戏所没有的许多功能。对我来说，互联网的出现和具有交互功能的游戏是非常美妙的组合。我认为在使用这些技术的同时，游戏也可以变得越来越吸引人。这时出现的问题就是，人们希望游戏变得好玩吗？这完全是 Deer Hunter 现象，还有 Slingo 和类似的游戏，我正尝试着将这些想法加入到自己的游戏模式中，但目前还没有做到。我从中得到的经验是，游戏中有更为广泛的玩法，比我熟悉的还要多。我们学习过程的一部分就是将这些玩法加入到设计游戏和制作游戏的方式中。但我仍认为自己的游戏可以流行于整个世界。

Sid Meier 作品一览

Hellcat Ace, 1982
NATO Commander, 1983
Spitfire Ace, 1984
Solo Flight, 1984
F-15 Strike Eagle, 1985
Decision in the Desert, 1985
Conflict in Vietnam, 1985
Crusade in Europe, 1985
Silent Service, 1986
Gunship, 1986
Pirates!, 1987
F-19 Stealth Fighter, 1988
Railroad Tycoon, 1990
Covert Action, 1990
Civilization, 1991



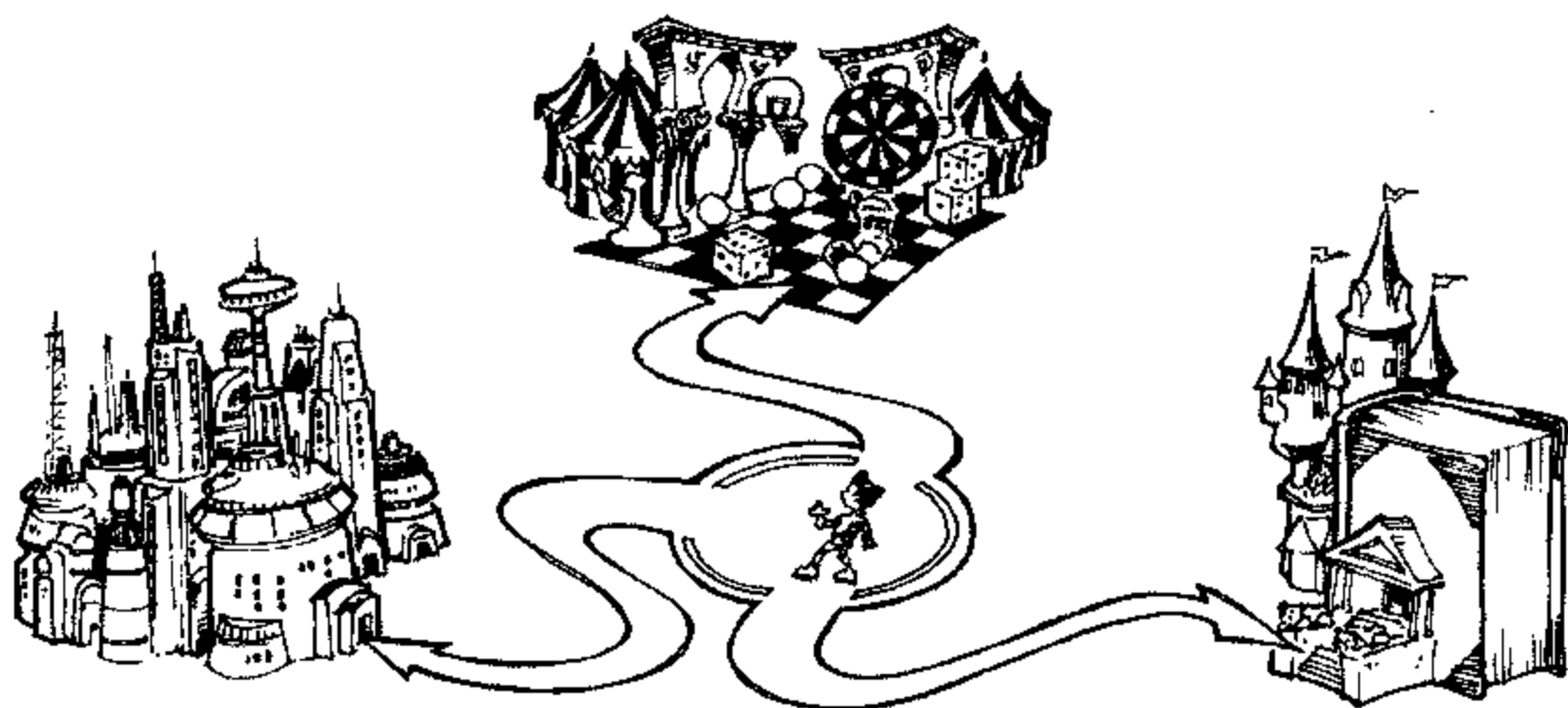


Colonization, 1994 (顾问)
Civilization II, 1996 (顾问)
Gettysburg!, 1997
Alpha Centauri, 1999 (顾问)



Chapter 3

第3章 游戏构思：游戏、技术和故事



“我在某所名牌大学演讲时，经常被问及一个愚蠢透顶的问题，你猜他们的问题是什么？他们一个劲儿地问我‘你从哪儿得到的启发？’我的回答是‘Schenectady’，大约40年以来，我一直这样敷衍着他们。每当这时，他们总是目不转睛地望着我，于是我说：‘没错，Schenectady，纽约州的那座城市。要知道，那儿提供一种启发服务，每星期我寄给他们25美金，然后他们给我寄来6包新近挑选的构思。’”

——Harlan Ellison

Harlan Ellison 也许是嘲弄那种试图阐明构思来源的想法。毫无疑问，如果一位小说家构思不畅，他就可能怀疑自己是否选择了合适的职业。同样，对一名优秀的游戏设计者来说，在任何时候，他都会拿出不下五种精彩的创意，并积极地将这些创意融入电脑游戏。在游戏领域，创意并不是什么稀有之物。那些抱负远大的游戏设计者们常常认为，他们可以向开发公司兜售自己的创意。在他们的眼里，游戏开发商们似乎总是在坐等热门创意的出现，之后，便花费数百万美元将这一创意转化为现实。然而，游戏开发面临的难题并不在于酝酿出色的构思，而在于贯彻良好的创意，并利用它创造出一款精彩绝伦的游戏。这就是本书接下来努力探讨的课题。

在游戏设计领域，受一系列因素的影响，提出合理创意的过程往往会变得更为复杂，而小说家们则不必为这一问题劳神费心。在某种程度上，这是因为电脑游戏的构思可以来自这一艺



术形式的三个泾渭分明、互不相干的方面：游戏、技术和故事。这些不同的创意来源以某些奇特的方式相互联系，其中的创意来源制约着设计者在其他方面可能取得的成果。因此，游戏设计者在着手构思并打算设计某一款游戏时——即根据游戏、技术或故事进行构思时——必须考虑到，最初的构思将影响到游戏成品的所有方面。



3.1 游戏设计的起点

举一个简单的例子。游戏设计者打算制作一款取材于某个希腊神话故事的游戏。这就要求设计者围绕故事情节展开工作，而这将直接制约设计者计划采用的游戏类型。显然，它不可能成为一款 Civilization 风格的战略游戏，因为这类游戏与那些关于宙斯、赫尔克里斯、阿瑞斯诸神的古典神话故事毫不相干。此外，也应排除实时战略游戏，因为这类游戏并不适于讲述那些仅仅涉及到几位主人公的故事。高品位的飞行模拟游戏可能也不理想。不过，这位设计者仍然可以采用动作、角色扮演或冒险类游戏类型。同样，技术上也会受到限制。为了描述这段希腊神话故事，他必须采用某种方法将大量的先行情节传达给游戏玩家。要做到这一点，就必须具备适当的技术。另一方面，如果设计者在这一阶段选择了某项技术，并将其应用于这款游戏，这就会进一步影响到可能采用的游戏类型。例如，选用 45 度视角的 2D 引擎最适合于角色扮演或冒险类游戏，而不是动作游戏。如果打算采用 3D 技术，为了恰到好处地再现这一希腊神话故事，它就必须能同时支持室内和室外环境，这样，许多 3D 游戏引擎将不在考虑之列。

每当设计者针对一款尚未投入制作的游戏做出某项决定时，他必须了解到这一决定会制约游戏未来风格的方式。如果设计者试图将一种并不合适的引擎套用于某个游戏类型，制作出来的游戏就会黯然失色。例如，在制作“上帝也疯狂”这款模拟型游戏时，采用第一人称室内游戏 Quake 风格的 3D 引擎就是一种极端错误的做法，就像通过飞行模拟游戏讲述希腊神话故事一样，这样的游戏根本无法取得成功。此外，这里还涉及一大难题，即一些所谓的“High-Concept”，这通常是一些营销专家们的伎俩，目的在于使某一产品能满足不同的市场需求。如果游戏的各个方面无法有效配合，讨论这种概念所涉及的问题便毫无意义：面对这款游戏，玩家们将提不起丝毫兴致。



3.1.1 从游戏入手

在游戏开发过程中，从已有游戏着手是最为常见的起始方法之一，对那些由设计者和管理者主导的项目来说，情况尤其如此。构思游戏风格通常是最易于把握的关键性环节，特别是当这种风格类似于某款已经面世的游戏之时。“这是一款赛车游戏！”“这是一款飞行模拟游戏！”“这是一款类似于 Super Mario 64 的 3D 动作兼冒险游戏！”游戏开发商通常会对以上游戏类型中的某一款游戏产生兴趣，并希望在这款游戏中发挥自己的创造力。如果设计者对某款游戏的总体构思感到满意，他就应该根据游戏可玩性制定出这款游戏应该达到的水准。它是一种什么类型的赛车游戏？应该掌握哪些适合于玩家们的赛车特点？当设计者针对游戏类型做出更加具体的构思后，就应该着手考虑这一构思对游戏所需技术以及可能采用的故事类型造成的影响。





此外，还应该根据你希望为玩家设计的游戏类型对设计工作需要采用的具体技术进行分析。这款游戏需要采用 3D 引擎还是 2D 引擎？采用 2D 引擎是否足以应付，甚至更加合适？应该让玩家看到什么样的游戏环境？是静态还是动态环境？在屏幕中，游戏动作的表现是急促还是缓慢？是否需要大量的游戏实体同时来回移动？游戏环境的规模如何？必须对上述所有问题及其他诸多问题进行分析，以便了解游戏引擎必须达到什么样的水准才能恰到好处地将游戏构思付诸实际。当然，应用于游戏的技术必须能在目标系统上实际运行，无论它是个人计算机、游戏控制器，还是定制的街机。此外，还必须询问一下游戏编程小组是否正在努力创造必要的技术条件。技术可行性最终可能会制约游戏风格的选择范围。更为糟糕的问题是：引擎小组的现有技术是否能发挥有效的作用？他们是否应该放弃现有技术并重头做起？他们是否拥有足够的资金和时间另起炉灶？如果你觉得必须改动游戏风格，使其适应现有引擎技术，就根本不必以游戏作为构思来源，而是利用技术展开创作，这就是下一节将要探讨的内容。如果在投入设计时只能使用某种引擎，最好不要将这项技术强加于无法与之调和的游戏风格之上，而是应该努力思索适合于这种引擎的游戏风格。

同样，游戏类型也制约着故事类型。角色扮演游戏讲述的故事远比动作/冒险类游戏复杂、深刻，而动作/冒险类游戏讲述的故事则比街机射击游戏更富于真实感。特定类型的故事与某些类型的游戏根本无法契合，就像上述在飞行模拟游戏中套用希腊神话故事一样。同样，爱情故事可能并不适合策略游戏，而有关人际交往的故事则无法与动作紧凑的第一人称的射击游戏完美结合。由于我们的选择是首先提出游戏风格，因此必须优先选择最适合于这种游戏风格的故事类型，然后再努力讲述好这一故事。有时候，设计者不仅拥有讲述故事的欲望，而且还希望对某个游戏类型进行探索，可以尝试在同一款游戏中揉合这两种因素，即使这两者无法相互协调。如果叙事类型与游戏风格毫不相关，就不要围绕游戏勉强拼凑生硬的故事情节，无论是在故事的复杂性还是题材方面。应该暂时搁下这一故事，待某款游戏的风格更加有利于故事的叙述时再行考虑。技术受小组在规定的时间内完成任务能力方面的限制，而故事则仅仅取决于我们的叙事能力。应该选择并应用那种与游戏风格最相适宜的故事情节。

3.1.2 从技术入手

有时候，在投入某个游戏项目时，针对游戏的大部分技术已经开发完毕，这也是一种比较常见的现象。如果这并非开发小组在一家新公司内首次共同参与的项目，他们就可能已经具备一定的技术基础，而这一技术正是项目得以展开的基础。即使项目经理打算采用一种“新型”引擎，这种“新型”“引擎”通常也只是旧式引擎的升级版本。因此，与这种引擎最相契合的游戏风格不会受到很大的影响。即使打算为项目重新编写引擎，领头的程序员和下属们可能已经处于最佳状态，随时准备设计出符合要求的引擎，无论它是室内还是室外型，是即时还是预渲染的，是 3D 还是 2D 的，采用复杂的物理系统组织动作还是应用更加简单的方法。程序员们可能会致力于开发一种特殊的光线或透视效果，并最终设计出一种在这些方面表现不凡的引擎。面对这一新的技术，设计者的任务是制作出能彻底吸收这一复杂技术的游戏。

有些项目经理会果断地采用其他方面如游戏开发商或纯技术性公司特许的引擎，有些人则更富于远见，他们会先考虑一下游戏的类型，然后再选择一种恰到好处的引擎。不过，更为常



见的情况是，在选择特许引擎时，他们选中的似乎总是那些最能体现“物美价廉”这一特色的引擎。确定引擎之后，设计小组的任务便是，利用它使游戏和故事完美地结合在一起。

如果希望从某种现有类型的游戏入手，游戏设计者就应选择相应的引擎类型。同理，如果从某种技术入手，就应首先考虑适合于这种技术的游戏风格。倘若设计者采用的是3D引擎，就应该使游戏环境产生3D效果，并利用这一环境创造出充满情趣的3D游戏风格。如果仅仅使用2D引擎，就应排除第一人称类的射击游戏。如果引擎具备复杂的物理系统，在设计过程中就应该利用这种机制创作谜题并设计玩家动作。当然，设计者不必采用程序员认为必须创造的每一项技术。因为重要的是，使游戏风格与引擎相互协调，而不是相互冲突。通常，如果项目经理者打算采用特许引擎，这种技术的创造者必定考虑到相应的游戏类型。设计者应该认真考虑可以偏离这一初始技术的限度。因为，显然更为简便的作法是，让引擎承担指定的任务，而不是使它背离程序员们的初衷。比如广泛的应用Quake引擎，它开发的目的是处理室内环境的游戏，即那种采用第一人称视角、动作紧凑、充斥着射击场面的游戏。尽管有些设计小组在得到这种引擎的特许权后，试图改变它的某些特点。但迄今为止，在艺术上最为成功的引进者Valve公司，在他们推出的Half-Life中大量保留了这种引擎最为擅长的游戏风格，即标准的Quake风格。当然，为了设计出理想中的游戏类型，Valve公司也在这种引擎技术中添加了不少属于自己的特色。不过，他们并未尝试一些不明智的作法，如将游戏主要环境设置在室外，或仅仅采用混战模式。有了现成的技术，游戏设计者就可以利用它从事游戏设计。对他们来说，最为明智的做法是，了解这项技术的不足之处，并在游戏中将它们转化为有利因素。

此外，技术还会制约游戏故事的类型。在某些游戏中，玩家必须输入信息与角色们进行沟通，因此没有语言分析程序，就很难讲述这一类型的故事。同样，没有能相当出色地处理室外环境的引擎，就很难制作出一款含有登山运动的游戏；没有可靠的人工智能思想，就很难设计出一款出色的人际交往游戏；没有能储存和重放大量声音文件的压缩技术，就难以处理大量的对话，从而难以生动地刻画角色，因为他们的语调对故事的表述非常重要；没有能同时显示大量移动单元的技术，就不可能让玩家们在故事中置身于辉宏壮观的战争场面。在只能采用某种引擎的情况下，游戏设计者必须考虑如何通过它将故事情节传达给玩家们。将某个故事与不相称的游戏风格撮合在一起，会严重损害游戏的质量。同样，如果试图通过不合适的引擎讲述故事，也会造成这种后果。再次以上面提及的Half-Life为例，如果Valve公司的设计小组试图将游戏场景安排在“死亡之谷”内，并让玩家同时与20只硕大的怪虫作战，Quake引擎就会失去应有的作用，游戏的运行效果也会让人无法接受。在死亡之谷的关卡中，Valve公司本来应该讲述他们理想中的故事，但这样做就会严重影响游戏的运行速度，面对如此骇人的场面，玩家们将感到索然无味。为了提高游戏的质量，必须使故事和技术相互协调，保持一致。





图 3.1 Half-Life 的设计者们明智地采用了室内环境的第一人称射击游戏风格，这种风格最初由 Quake 引擎确立。在 Half-Life 的创作过程中，设计者们得到许可后采用了这一引擎。图为 Quake II 中的场景



3.1.3 从故事入手

最后，在构思游戏时，设计者有可能从以下几个方面着手：

- 打算采用的故事背景
- 打算讲述的故事
- 打算深入刻画的角色

比起技术或者游戏风格，这种起始方法并不那么常见。事实上，由于许多游戏根本不需要以故事作为铺垫，因此从故事着手进行设计这一概念可能会显得很别扭。不过，这种情况也并非前所未闻：游戏设计者可以先根据自己的意图构思出故事情节，然后再探讨什么样的技术和游戏类型最适合于表述这一故事。在构思这种游戏的过程中，任何一名优秀的游戏设计者都会考虑到以下几个问题：

- 如何在游戏中展开故事？
- 如何使玩家与故事产生互动关系？
- 如何根据玩家在游戏环境中的行为以不同的方式展开故事？

由此可见，设计者考虑的不仅仅是故事情节，还包括游戏的风格。不过，可以将故事作为游戏设计的起点，于是其他方面都将受制于这一核心因素。

当然，故事的类型对项目必须采用的游戏类型有着不容忽视的影响。如果设计者打算围绕一群朋友展开故事情节，并让这些人置身于杀气腾腾的魔幻世界，与难以计数的怪物展开殊死搏斗，就应采用协同作战的第一人称射击游戏风格。对任何类型的游戏来说，如果需要玩家与



各种类型的角色进行交谈，并让他们对这些角色进行“探索”，那么可以运用更多角色扮演方面的技巧对游戏进行处理。如果讲述的是有关滑铁卢战役的故事，最好采用战略游戏风格，同时对这种风格进行调整，以便根据自己的初衷尽可能完美地表现滑铁卢战役的方方面面。设计者是否希望玩家在游戏中以将军的身份审视全局？如果是，就应该采用能对战术和后勤进行跟踪的游戏形式。倘若设计者打算从士兵的角度讲述故事，就应该让玩家逐一跟踪和操纵自己的部队。如果同非玩家角色进行交谈在故事的表达中占据重要地位，设计者就应该利用语句输入、分支对话选项或其他行之有效的方法设计出能够实现这种对话的游戏结构。设计者必须选择一种恰到好处的游戏风格，以便让玩家体验到所述故事中的一些至关重要的因素。

当然，技术也必须与故事密切配合，因为设计者选择最适于某一故事的游戏风格后，必须具备能支持这种游戏风格的技术。如果对话在表达故事的过程中起着非同小可的作用，编程小组就应该有能力开发出一种对话系统。如果对环境的探索和发现在故事的叙述中占据不小的篇幅，3D引擎就有可能最适合这种游戏。因为这样玩家就可以利用游戏摄影机随意观察周围的环境。设计者可能会发现，特意编写的事件对表达故事的诸多方面非常重要；玩家必须能观察到在不同的场景和特定的时间内发生的特殊事件。在这种情况下，程序员应该赋予关卡设计者构筑这些场景的能力。技术是同玩家进行沟通的媒体，因此故事将直接受制于技术的叙述能力。



图 3.2 Maniac Mansion 是 LucasArts 公司首次应用 SCUMM 系统设计的一款以故事为核心因素的冒险游戏

Infocom 和 LucasArts 公司制作的一些冒险游戏充分体现了这种以故事为中心展开游戏设计的方法。两家公司的所有冒险游戏都采用了标准化的游戏结构和技术。设计者们在创作过程中使用的是公司特有的冒险游戏制作技术，即 Infocom 公司的“文字冒险创作”工具或 LucasArts 公司的 SCUMM 系统。当游戏设计者参与某个项目之时，他就会以编故事为起点展开创作过程。毫无疑问，故事必须与冒险游戏的形式完美结合，而且在实现过程中只能应用现有的工具装备。Infocom 和 LucasArts 公司的制作工具都具有广泛的通用性，因此设计者可以利用它们创作出各





种类型的游戏。虽然这些工具的核心结构必须包含 Infocom 公司的以打字为基础的文字冒险风格，或者 LucasArts 公司以鼠标点击为特点的图像冒险游戏风格，但它们都在故事的叙述方面具有很强的应变能力。在创作过程中，游戏设计者的主要动机是讲述故事，因此游戏结构的设计和技术的开发将居于次要地位。就像一位电影导演在拍摄过程中不仅受到摄影机的限制，而且还必须考虑到银幕的尺寸和每秒 24 帧的放映速度一样，在 Infocom 和 LucasArts 公司内，冒险游戏的设计者们也会受到现有的冒险游戏创作系统在结构上的制约。无论对电影导演还是冒险游戏的设计者们来说，早在投入项目之前，媒体结构就已经牢固确立。因此，他们的首要任务是讲述故事。



3.2 制约因素

有经验的游戏设计者都会了解技术、游戏和故事对游戏创作所产生的制约作用。当这些设计者们参与集体讨论会时，他们能深切地体会到围绕讨论中的游戏做出的某种选择是如何进一步制约游戏的创作过程的。每次就游戏做出一项决定时，许多可能性将不复存在。如果已经就游戏的特性进行了充分的论证并做出了决定，就有可能只剩下一种游戏类型能完全达到设计者期望的效果。做出重大决策的阶段已经结束，摆在面前的是数以千计的有关实施工作的“次要问题”。

在我设计完成的三款游戏 *Odyssey: The Legend of Nemesis*、*Damage Incorporated* 和 *Centipede 3D* 中，我采用了不同的起始方法投入开发工作。凑巧的是，其中一款从故事入手，另一款从技术着手，第三款则以游戏为创作起点。在每一款游戏的开发过程中，我总是竭力回想它的来龙去脉以及当初希望的效果。游戏设计的初期工作和目标制约着游戏的所有方面，就是在这种情况下，我设计出了能让人感到满意的游戏作品，并达到了当初制定的所有目标。



3.2.1 *Odyssey: The Legend of Nemesis*

Odyssey: The Legend of Nemesis (奥德塞：复仇女神的传说) 是一款以故事为创作起点的游戏。事实上，我是在 2D 技术和角色扮演游戏结构取得重大进展之时接手这一项目的。故事已经拟就，但还谈不上完整，因此我对它并无多大兴致。由于这是我首次参与设计一款能真正正式发布的游戏，因此我毫不犹豫地投入了创作，根据自己的意图对故事进行了改编。多年来，我一直积极参与游戏开发工作，试图讲述一些具有互动性的非线性故事，因此我即刻埋头编写这一类型的故事，这类故事让玩家进行一些道德选择，而不是仅仅让他们决定“杀还是不杀”。我希望在自己创作的游戏里，玩家们做出的选择能真正改变故事的结局，并让人感到回味无穷。因此，我将故事作为自己惟一关心的对象，开始大胆地探索。

幸运的是，现有的相关技术和游戏结构基本符合我叙述的这篇故事。当出现冲突时，我就会对游戏结构做出必要的调整。当非玩家角色的智能必须以某种方式运行才能有利于故事的叙述时，我就会努力采用该种方式。如果有必要采用强迫交谈模式，如要求一名非玩家角色走近玩家并主动与之交谈时，我就会对游戏结构做出适当的调整。关卡设计的目的也仅仅在于支持



故事的叙述。由于在设计关卡时并未考虑激烈的战争场面，因此，在这款游戏中，战斗场面并不那么动人心魄。我对故事本身的痴迷胜过于对战争的兴趣。与那些奇特、凶猛的怪物频繁搏斗的情景经常出现在角色扮演游戏之中，因此这已经成为一种固定的模式。不过，我总是优先考虑故事情节的发展，而将战斗场面置于非常次要的地位。结果，这款游戏最终与角色扮演游戏相去甚远，它更接近于冒险游戏。但是，这正好符合我的意图，因为只有这样才能恰到好处地支持故事的叙述。

回顾 *Odyssey* 这款游戏，我们可以发现其中存在着不少缺陷。但我觉得，出现这些问题并不是因为它是一款以故事为开发起点的游戏。也许，这种游戏开发方式并不常见，但它仍然是一种切实可行的起始方法。假如当时我在游戏设计方面具有更加丰富的经验，我就会努力使游戏的其他因素和故事一样引人入胜。同时，这一史诗性故事的恢宏气势也将秋毫无损。

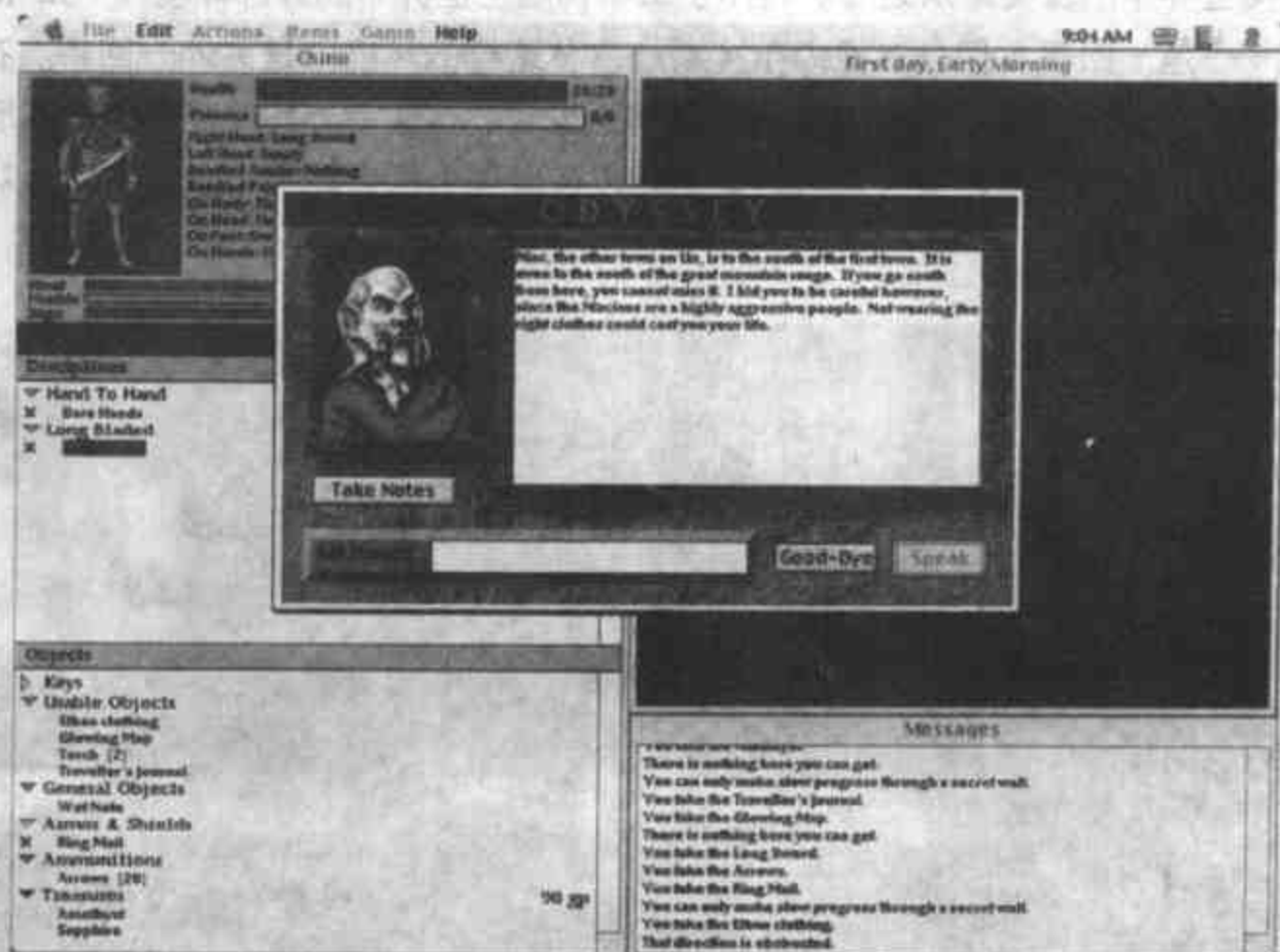


图 3.3 围绕故事展开奥德塞游戏的关卡设计



3.2.2 Damage Incorporated

在 *Damage Incorporated* 的开发过程中，作为开发商的 MacSoft 设法获取了一种尖端技术（在当时来说）的特许权，他们试图将这种技术应用于一款游戏之中。这是 Bungie 软件公司为制作 *Marathon* 和 *Marathon 2* 而开发的一种技术。这两款游戏给我留下很深的印象，尤其是 *Marathon 2*，它至今仍然是有史以来最为出色的第一人称射击游戏之一，连 *Doom* 也无法与之抗衡。虽然 *Marathon 2* 在营造火爆的战斗场面和恐怖的气氛方面不及 *Doom*，但它那动人心魄、错综复杂的故事情节，以及出色的关卡设计和优良（尽管有些简单）的物理模型足以弥补这一缺憾。由于我对 *Marathon* 系列游戏非常着迷，因此决定在自己设计的游戏中采用 *Marathon* 创造的技术和风





格。我打算以这项特许技术为基础投入游戏创作，并尽可能最大限度地发挥它的效力。

我从技术着手，摸索出了适合于 Marathon 技术的游戏风格和故事情节。当然，我们为游戏和引擎添加了一些新的功能。其中在游戏结构上添加的主要功能，就是使玩家能在游戏环境中向队友们发号施令，这便赋予游戏一种即时战略的色彩。我们还对引擎作了大量的改进，使之能创造出门的开合、物体的运动和其他一些特技效果。只有这样，游戏环境才能更加类似于真实世界。在游戏故事方面，我仍然相当投入，只是在程度上不及 Odyssey 而已。事实上，同非玩家角色交谈这一特点并不符合 Marathon 游戏结构，因此，我便将这些角色穿插在玩家的队友之间，当玩家在游戏环境中调遣他们时，他们就会喋喋不休地相互交谈起来。

这款游戏的缺点之一是，在项目之初，我对 Marathon 引擎的限制因素不太了解。这种引擎最适合于创造室内环境，因此，一旦涉及到室外场景，制作出来的画面就显得不够真实，特别是在描绘现实世界中的真实地点之时。利用引擎模仿陌生世界中的外部环境——如 Marathon 2 进行的尝试——是一回事，但创造像内布拉斯加森林这样的环境则是另外一回事。在 Damage Incorporated 中，大约有半数关卡采用了室外环境。结果，这些室外场景表现得均不尽如人意。假如我当时对这项技术了解得更加深入，就会在设计过程中更多地采用室内环境，从而更好地发挥这种引擎的作用。



图 3.4 取材自 Marathon 特许技术的 Damage Incorporated

凑巧的是，当我利用 Marathon 2 引擎创作 Damage Incorporated 时，MacSoft 公司的一个设计小组也采用这个引擎开始创作一款名叫 Prime Target 的游戏。这个小组的设计人员并不像我一样热衷于 Marathon 2 引擎，他们希望创作出一款更加富于 Doom 风格的射击游戏，使战斗场面更加紧凑、快捷和火爆。他们没有从技术着手，也没有采用这种技术所擅长的游戏类型，而是



以自己喜爱的游戏类型为创作起点，并对引擎进行了修改，使之更加适合这种游戏类型。为了满足自己的需要，Prime Target 设计小组花费了大量时间来修改这种引擎；相对来说，我们花费的时间则少得多。由于这一原因，Prime Target 的风格与 Marathon 2 或 Damage Incorporated 相去甚远。它并未超越后者，也不比它们逊色，只是风格不同而已。这种差别可以追溯到游戏构思的来源，以及他们对特许引擎的处理方式。

3.2.3 Centipede 3D

在启动 Centipede 3D 项目之时，作为开发商的 Hasbro Interactive 公司与这款游戏的开发商 Leaping Lizard 软件公司展开交涉，要求采用他们的 Raider 技术开发 Centipede 的更新版本。不久前，Hasbro 公司在改进 Frogger 的过程中尝到了甜头，因此他们试图在已经获得版权的 Centipede 上大做文章。Hasbro 公司的制作者们在一本杂志上看到一篇有关 Raider 的预告文章后，认为这种游戏风格正好适合于 Centipede 项目。Hasbro 公司非常清楚自己应该给 Centipede 3D 套用什么样的游戏风格：游戏结构应该类似于经典的 Centipede。不同的是，它采用了 3D 环境。Leaping Lizard 公司的设计小组答应了他们的要求。在当时，采用直观、典雅的街机风格的游戏并不多见，而使其适应 3D 环境无疑是一种前所未有的挑战。

在制作 Centipede 3D 时，设计者们以游戏风格作为这款游戏的开发起点。在这一项目的整个开发过程中，大家首先考虑的是重现 Centipede 原作的风采。当 Hasbro 开始物色一家能处理这款游戏的公司时，他们知道应该找到一种能应付更多室外环境和较大场面的引擎，因为这些都是改进后的 Centipede 必须具备的场景类型。他们知道，为了达到理想的游戏效果，不能选择 Quake 风格的技术。Leaping Lizard 公司的 Raider 引擎非常适合这款游戏，但它也并非完美无缺。必须下功夫对这种引擎进行修改，以便取得经典“街机游戏”的快速反应能力。对 Centipede 3D 来说，Raider 采用的 physics system 基本上起不到什么作用，因此，他们对 physics system 做了大量的删减。这样，经过一番调整，这一技术与他们理想中的游戏风格完美地结合在一起了。

在我参与设计的所有游戏中，Centipede 3D 的故事最为简单。在某种程度上，这是因为经典街机游戏的特点之一是，它们都缺乏讲述真实故事的基础。对于某些游戏如 Centipede、Pac-Man 和 Space Invaders 来说，构筑环境就已经足够，游戏需要的只是一个能实现自身的基本前提而已。此外，参与 Centipede 3D 项目的所有人员都将游戏风格作为优先考虑对象，而故事则仅仅是一种次要因素。正如我们所预见的，这款游戏具有操作方便、易于上瘾的特点，吸引玩家们的是 Centipede 3D，而不是它所包含的故事。经典的街机游戏风格根本不需要以故事作为铺垫。贫乏的故事情节起到的主要作用是，提供游戏布局，影响游戏外观，解释玩家腾挪闪转及射杀那些蜈蚣和蘑菇的理由，还可提供过关后游戏环境改换面貌的原因所在。旧版 Centipede 利用花园内的场景和昆虫来解释游戏的发展，而新版的 Centipede 3D 则运用故事情节来支撑游戏的进行。总之，Centipede 3D 关注的焦点是游戏的可玩性。



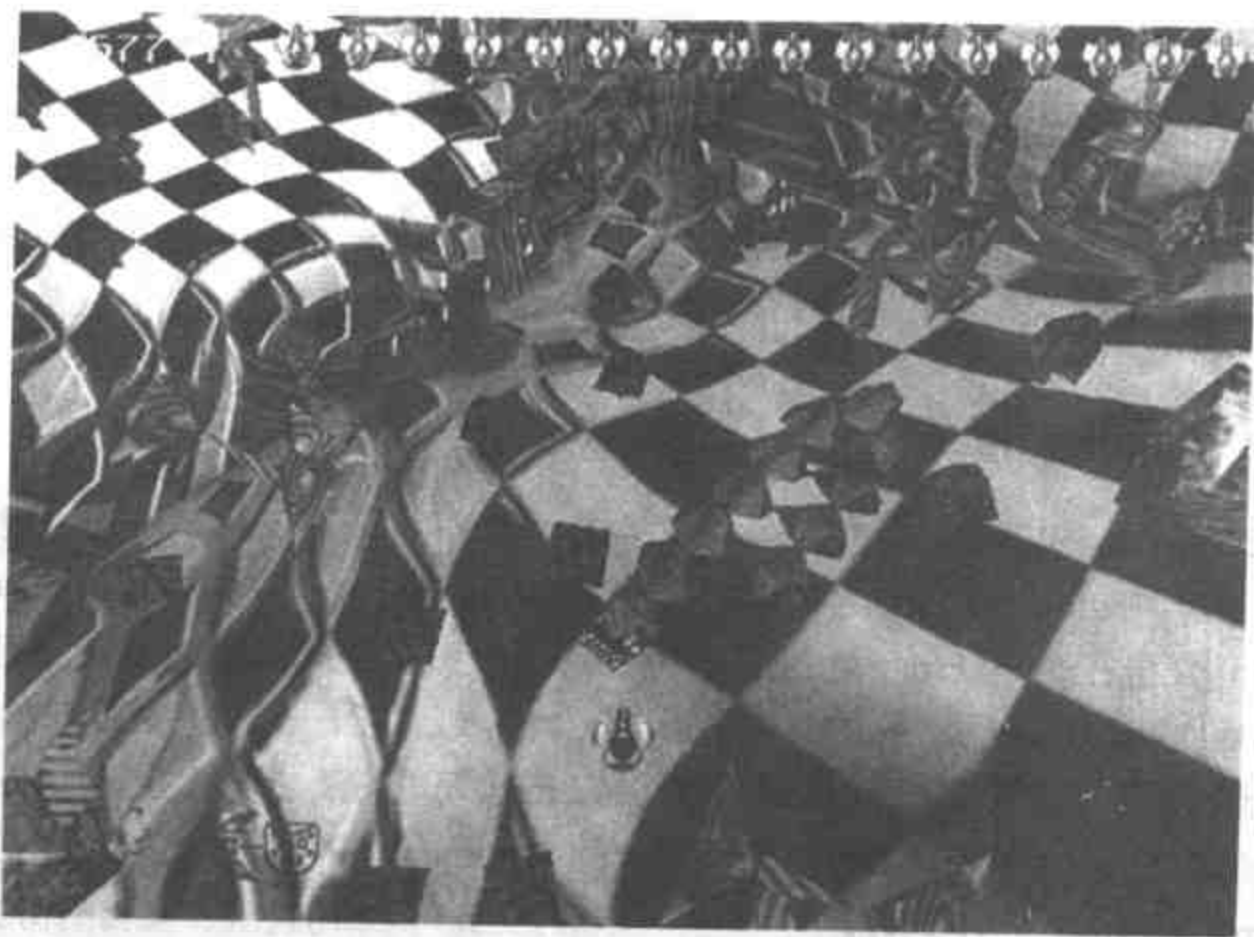


图 3.5 采用 3D 技术的新版 Centipede 以旧版 Centipede 中的经典“昆虫射击”游戏风格为基调



3.3 制约因素の利用

在许多情况下，开发一款游戏时必须首先了解你所面临的制约因素，并使这些制约因素朝着有利的方向转变。在这一章中，我已经论述了游戏构思的来源，即游戏、故事和技术以及设计者理解它们的方法。通过这种理解，设计者必须认识到，一种游戏因素制约其他因素的方式，如某种游戏类型为什么需要与一定类型的故事和技术相结合，技术是如何与特定的游戏和故事类型保持协调的。设计者的任务是将所有因素结合在一起，去粗取精，创造出一款令人陶醉的游戏来。

事实上，设计者很少能根据自己的意图构思游戏，然后再找到实现这种构思的正确方法。几乎在任何情况下，设计者都要受到客观条件的制约。这些制约的表现形式可能包括，现有的技术条件、同事们的具体情况、游戏开发经费以及制作期限等。制作人的主要职责是保证在预算之内、按时完成游戏的制作，而设计者则必须考虑到自己所面临的各种制约因素。只有这样，他们才有可能设计出一款上乘之作。



3.3.1 技术环境

通常，在较大的公司内，设计者只能利用现有的技术从事创作。这种技术有可能是上一部作品留下的引擎，或者编程小组仅仅具备处理 2D 图像的经验，结果，他们在一定的时间内只能开发出 2D 技术。即使设计者有幸为项目物色到一种特许技术，他们仍然要受到引擎的质量以及经费的限制。



如果开发者喜欢单独行动，在某个项目中身兼数职，既是设计者又是程序员，旁人也许会认为他能毫无顾虑地从事设计工作。然而，情况决非如此，因为他很快就会受到自身编程能力及实际可承受工作量等因素的制约。一名程序员无法创造出能与 Quake III/IV 或更高版本相抗衡的全方位 3D 技术，这是根本不可能的事情。兼任程序员和设计者独自承担一个项目固然有不少可取之处，但毫无疑问，这样做会限制个人能力的发挥。

即使程序员有能力为自己的游戏设计出完美无缺的引擎，但是，如果速度跟不上时该怎么办？如果游戏需要大量口齿清晰的角色，并采用 3D 即时室外环境，那么以当今的技术，帧频根本达不到要求。只要给游戏中的所有怪物附上尖端级的人工智能，游戏就会以每秒一帧的速度运行。实质上，这与播放幻灯片毫无二致。如果设计者仍然不肯放弃，他就只能等待几年才能办到。当发行商得知在改进技术之前无法重新启动游戏项目之时，他们往往会当即停止支付高额的酬金。

3.3.2 蘑菇的启示

在设计 Centipede 3D 的过程中，我们经常为帧频问题所困扰。记得对于这款游戏，我们首先考虑的是实现一种保留原创经典街机游戏的风格。可是，Centipede 需要在屏幕上同时显示大量的蘑菇，以及同样数量不菲的其他蜘蛛和节肢类动物，这些漫天飞舞的昆虫使玩家们的小小“射击”船随时面临毁灭性的打击。此外，这款游戏还需要采用俯视效果，因为这样可以扩大游戏环境的可视范围，从而便于玩家们观察那些可怕动物的一举一动。结果，在屏幕上同时出现数百个 24 边形的蘑菇、二十只 40 边形的蜈蚣体段和不计其数的小动物。除此之外，Hasbro 公司还希望将 Centipede 3D 设计成一款畅销游戏。于是，这一产品的最低系统要求被预定为奔腾 133MHz，而且无需硬件图形加速设备的支持。最后，在运行 Centipede 时，帧频必须达到一定的水准才能跟上紧凑的游戏动作，否则，就根本无法提起玩家们的兴致。

在参与项目的过程中，我们经常遇到逐渐升高的边数问题，美工们总是试图从使用频繁的蘑菇模型上削减几个边。有一次，一名美工提议说，如果将蘑菇降格为两个叠加的金字塔，也许就能以最低标准，即通过 6 个或 8 个边来表现这种植物。事实上有人指出，如果对所有游戏模型均采用这种极简表现手法，就能通过有效限制边数的做法创造出一种独特的游戏环境：画面上充斥的物体看起来就好像是立体派画家们的作品。毫无疑问，对于一款游戏来说，这将是一种独到的表现手法，而且，由于 Centipede 3D 中的游戏环境本来就带有某种超现实主义的色彩，因此它显得非常协调。“利用这些不利因素吧！”在讨论过程中我冒出这样一句话来，就好像一位倍感厌烦的教授做出最后的宣告，“Eureka！”（找到啦！）所有在场人员都认为我的表态非常可笑，可是，后来当我回想这一事件时，我还是坚信这一想法确实有助于游戏开发工作。遗憾的是，开发工作已经进行得相当深入，我们无法以极简表现手法彻底取代原有的艺术风格，更不用说采用这种极简游戏手法蒙骗开发商可能会招惹的麻烦了。

不过，总的来说，我仍然认为，只要有可能，游戏开发者应该利用自己所面临的制约因素。如果拥有现成的引擎，而且这也是项目的惟一选择，为什么还要竭力设计一款与这种技术背道而驰的游戏呢？也许你的设计方案新颖独特、考虑周到，但如果现有的技术无法很好地实现这一方案，你仍然将以失败而告终。因此，最好暂时搁下那些与技术相抵触的想法（以后随时可





以重新考虑这一想法), 提出一种更加适合于现有工具的设计方案。一旦发现分配给自己的引擎存在着一些局限性, 最好还是避开这些局限性, 而不是尽力消除它们。这并不是说设计者应该尽可能设计那种极其简陋的游戏, 或者不应该尝试那些复杂的和实验性的设计方案。如果一位精明的戏剧导演发现一名指定给他的演员非常乐于同他合作, 他就会挑选最好的剧本使这名演员特有的表演技术大放光彩。同样, 游戏设计者应该考虑到技术的适用性, 并以这一适用性为基础, 确定游戏的设计方案和故事情节。

3.3.3 时间限制

预算和日程也是设计者需要面对的制约因素, 虽然在本章我并未对它们进行深入探讨, 但它们的重要性仍然不容忽视。尽管应付这两种因素主要是项目制作人的考虑对象和职责范围, 但游戏设计者同样有必要了解它们对项目的制约方式, 就像了解技术、游戏或故事的制约作用一样。决定采用某种技术时, 设计者必须回答以下问题: 能否在给项目预定的时间内完成这一技术? 是否能在关卡应用和调试时及时完成这一技术? 提出的设计方案是否要求制作大量、复杂的关卡以及极度脚本化的动作, 以至于仅靠一名设计者无法在 18 个月内完成这些工作? 就像日程表会限制花费在项目上的时间一样, 预算将影响到这一期间参与项目的人数。“它是可行的”, 如果将预算资金增加一倍, 设计工作可能会轻而易举地在一年半的时间内完成。可是, 如果将预算资金减少一半, 设计者就必须调整原有设计方案, 使其更加切实可行。此外, 如果开发工作拖延 6 个月后依旧遥遥无期, 发行商就会停止投资, 这时, 无论你的设计方案在理论上如何出色, 也于事无补了。没有人能接触到你的游戏, 因为你对推广这款游戏的后勤工作缺乏周详的考虑。如果你没能安排充足的时间对游戏进行调试和改进, 开发商可能会在设计工作完成之前要求你提交这款游戏。一款上乘之作就会这样中途夭折, 而原因只是没有足够的时间彻底完成它而已。

在避免时间和预算限制方面, 那些单独行动的游戏开发者处于更加有利的地位。如果一个人独自承担游戏的美工、编程和设计等工作, 并利用自己的空闲时间从事游戏的开发, 而且不必依靠游戏开发所获得的收入维持生计, 那么, 只要他愿意, 他就会有更多的自由随意发挥自己的艺术想像力。当然, 他仍然要受到自己的才能、艺术创作的质量以及编程能力的制约, 不过, 至少这些都是仅有的制约因素。在艺术创作方面, 值得一提的是, 没有人会对资助者感恩戴德。

3.4 不选择也是一种选择

总之, 在构思游戏时, 游戏制作人、程序员、美工和设计者往往会忽视其中的制约因素。无论构思源自游戏概念、技术设想还是对故事的思索, 一旦游戏构思酝酿成熟, 它就会开始制约游戏可能达到的效果, 直到大功告成。游戏设计者必须认识到, 并不是每一种技术都能与每一种游戏设计方案保持协调, 每一种设计方案也并不一定适合于每一种故事类型, 而且每一种故事类型也并不一定能适应每一种技术。





游戏开发者往往试图将一连串令人折服的设想注入同一款游戏。程序员中的负责人也许会热衷于开发一种切线逆向运动学系统。自从首次领略过帝国时代这款游戏后，领头的游戏设计者也许一直希望尝试一下即时战略游戏类型。游戏故事的编写者可能会认为在电脑游戏中充斥着过多的暴力，因而打算创作一篇爱情故事。如果制作人不够精明，当小组人员对自己开发的游戏沾沾自喜时，他也许会感到分外激动；此外，他还会以为，结合 IK、RTS 和爱情故事这三种因素，就一定能设计出一款别具一格的游戏来。

毫无疑问，只要具备一点点判断力，任何人都会知道这样的游戏将注定遭受失败。假如，在集体讨论会上，设计小组能够断定自己希望重点把握的游戏因素，他们就会尽力保持游戏的完整性。假设他们选择了 IK，并一致认为它将有助于一款最富于完整性的游戏的诞生。然后，经过深思熟虑，设计者认识到也许 Street Fighter II 风格的格斗游戏最能发挥 IK 系统的长处。故事编写者就会叙述这样一则故事：一位勇士与万神殿中的希腊诸神逐一决斗，希望终有一天与自己的心上人 Hera 会面。经过一番努力，这款游戏可能会拥有很强的可玩性，因为所有因素都和谐地结合在了一起。总之，你一定不会希望自己设计的游戏仅仅包含优秀的技术、动人的故事或充满智慧的游戏设计方案。如果这些因素无法相互依托，故事就会显得缺乏新意，设计方案就会空洞肤浅，技术也显得不够完善，而游戏自然不会取得成功。



Chapter 4

第4章 游戏分析：Centipede



人们可能认为那些经典的街机游戏是一种电脑游戏，就如同认为荒诞剧是电影的一种形式，或者牵强地认为侦探小说是一种文学形式一样。随着可供使用技术的不断进步，商业游戏公司很快就对这些传统的街机形式失去了兴趣。但是，许多独立的游戏开发者却仍沉迷于经典街机游戏之中，或者是出于自身的兴趣，或者准备以免费软件或共享软件的方式进行发布。在这些出于之中的劳动成果中，有许多是现有街机游戏的翻版，但也有很多是对一些新游戏非常有趣的尝试。在游戏的形式方面仍有至关紧迫的事情要做，一个人并不需要非常复杂的 3D 引擎就能制作非常有趣的传统街机游戏，这就会使得“形式”成为人们乐于从事的一件事情。

需要提醒的是，我所提到的经典街机游戏，并不是说所有的传统游戏都是经典的。其中的许多都相当糟糕。一个老的街机游戏，如果在发布几十年后还能通过各种媒介被人记起或者谈论，人们往往认为这是个好游戏，进而使人产生一个错误的“黄金时代”的印象。那些糟糕的街机游戏已经随历史而消亡。“经典街机游戏”指的是形式上的经典，而不是游戏本身，就好像人们讨论“古典音乐”时那样。当然，“街机游戏”一词范围也过宽，因为从这个词来看，



好像应该涵盖了在大街上所能见到的所有游戏，包括竞赛游戏、射击游戏以及格斗游戏，但上面提及的都不是我所关心的形式。

在 70 年代末期到 80 年代初期，传统的街机形式达到了商业上和创新上的全盛时期，当时大街的两边各种机器展示着这种形式。从整体上来看，我们能总结出它们之间一系列共同的特点。出于街区的商业考虑，形式上的某些方面已经得以实现。基本的思路是使玩家更容易地理解游戏，这样，在玩过第一次游戏之后，他能对游戏的玩法及获胜的方式有较好的了解。第二，玩家的游戏（即使是专业玩家的游戏也是如此）不能持续时间过长，因为玩家只投入了 25 美分的硬币，如果半小时才赚 25 美分的话，经营者就无利可图。要让玩家不断地玩游戏，并不断地投入硬币。因此，从某些方面讲，街机游戏需要比在家里玩的游戏更为讲究。玩家购买家庭游戏时，通常费用至少是玩一次街机游戏的 100 倍——销售即告结束。如果玩家并不是完全厌恶这个游戏的话，就不太可能将游戏退回。游戏中的分数和高分榜等东西主要服务于这一本质，即让人对街机游戏更加入迷，鼓励玩家继续花钱。

另外，当时技术条件的局限会限制游戏的表现，从而影响了游戏在可玩性方面所能达到的目标。如果当时的游戏设计者能够拥有足够的内存及处理能力，将比屏幕大许多倍的游戏世界在游戏中以全屏滚动的方式展现，他们也许会这样做的。如果他们能在游戏中重复播放某种全动态视频图像，也许设计者会在游戏中加入更多的故事线索。但实际情况依然如此：一种独特的游戏类型产生后，如果当时的商业及技术限制决定了它的形式，那么也只能这样了。这就跟早期的电影一样，不得受无声及放映时间较短的条件限制，尽管游戏设计者能够设计出一些

非常优秀的游戏，但他们会受限于所能够创造出来的东西。通常情况下，在各种限制条件下工作，会促使艺术家们将创造性集中于一种方式，比起随心所欲地工作，这种方式可以产生更为理想的工作效果。

许多经典街机游戏的主要组成部分，就在于它们在游戏风格上存在巨大的差异。Centipede, Missile Command, Pac-Man 以及 Frogger 等游戏相互之间都有着根本的区别。另有，许多经典街机游戏在主题方面也各有不同：Centipede, Space Invaders, Galaga 及 Tempest 均围绕着一主题发展，即向不断蜂拥而至的进攻敌人射击。但是，这些游戏所采用游戏方式的区别要远远大于现代游戏，现代游戏更加满足于无休止地克隆已被实践验证成功的游戏类型。尽管传统街机游戏在游戏风格上各不相同，但我们仍将这些游戏视为精品，并将其作为计

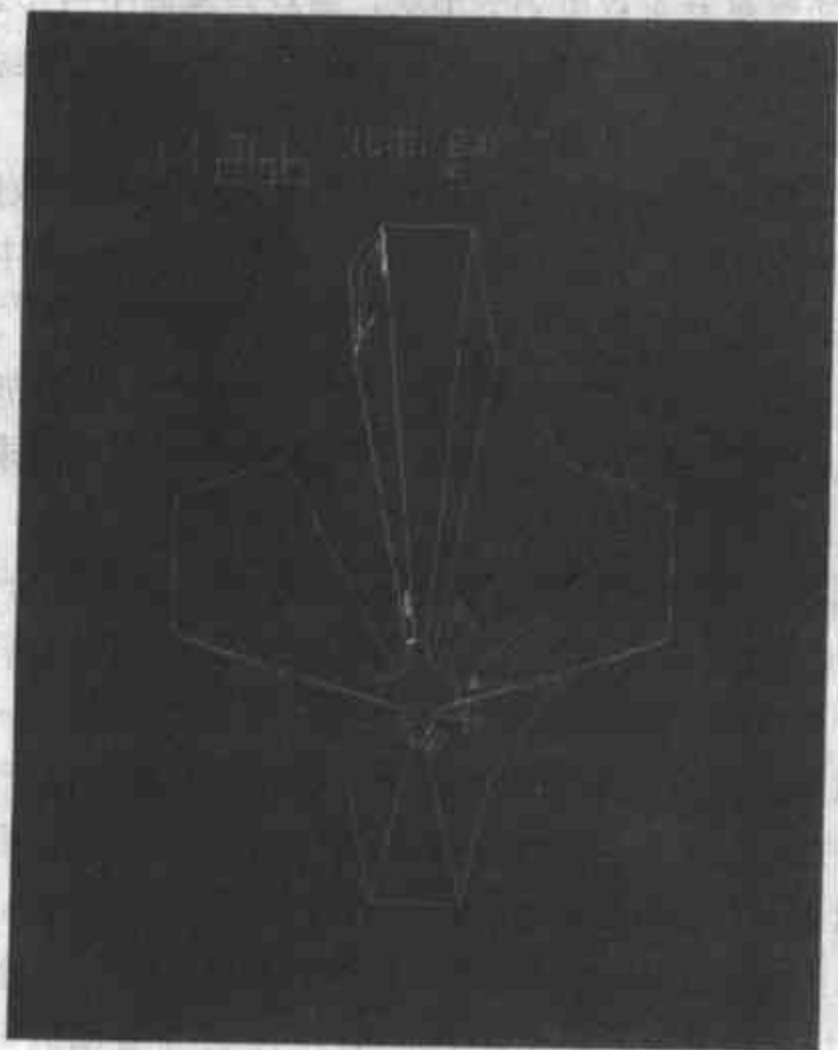


图 4.1 Tempest 是许多经典街机游戏之一，游戏的主要方式是向不断靠近的敌人射击。人们之所以对 Tempest 印象很深，是因为其中加入了许多独特的手法





计算机游戏简短历史上一个艺术发展阶段。通过分析这些游戏的共同点，现代的游戏设计者可以学习到，如何创造自己的游戏，才能让玩家体会到趋之若鹜的感觉。



4.1 经典街机游戏的特点

- **单屏游戏**：在经典的街机游戏中，大部分的游戏都在单屏中进行，玩家操纵他的替身在屏幕的范围内活动，有时只在屏幕的一个区域活动。之所以这样做，部分原因是由于技术条件的局限，但在游戏设计上，这样做也有其非常重要的艺术结果：玩家任何时候都可以观察整个游戏环境，能够依据其所了解到的实际情况做出决策。很显然，给玩家提供这样的信息将大大影响游戏可玩性。许多经典的街机游戏中还包括多个屏幕的游戏内容，通过切换场景或修改现有的场景，玩家可以创造出不同“级别”的新游戏。之类的例子包括 *Joust*，*Pac-Man* 以及 *Mario Bros*。在整个游戏中，虽然这些游戏可能不只包括单屏的游戏内容，但在任何时候，玩家的游戏环境只局限在一个屏幕。
- **无限游戏**：玩家可以无休止地玩下去。游戏没有终结，因此也不会有赢家。这样做部分原因是使玩家可以自我挑战，看看用一枚硬币能玩多长时间。玩家永远不会说：“*Asteroids* 游戏我已经打通关了”，这样玩家就能一直玩下去，不断投入硬币。同时，无法打通关的游戏，对玩家来说都意味着失败。每一次游戏都以玩家的死亡而结束，因此可以称得上是一种“悲剧”。无法战胜的游戏必须使游戏对玩家来讲越来越困难，因此在游戏设计中就要不停地、无限制地增加难度。随着家庭游戏市场的出现，游戏开发商不再希望玩家永远只玩一种游戏。相反，他们希望玩家玩通一个游戏后再购买另一个游戏。这就是现在很少再会看到无限类型游戏的原因。
- **多重生命**：通常情况下，在游戏进行的过程中，经典的街机游戏允许玩家拥有有限数量的生命，或有限次数的“尝试”。这种做法可能是起源于弹球游戏，在游戏进行过程中，多重生命的游戏会给新手了解游戏机制的机会。如果给他们足够的时间来找出游戏窍门，他们则更有可能再玩一次。对高手来说，拥有生命使游戏能够带给他们另外一种奖励的刺激：额外的生命。多重生命也可以使游戏更加有意思，死掉一次并不代表游戏的结束，这样可以鼓励玩家冒在其他情况下不敢冒的风险。
- **计分/高分**：几乎所有的经典街机游戏都包含计分的特性。通过计分，玩家可以累积分数来完成游戏中的不同目标。比如说，在 *Centipede* 游戏中，玩家每摧毁一个蘑菇便可得到一分，十分可以得到一段蜈蚣，一百分可以得到一个蜈蚣头，一千分可以得到蝎子。街机游戏另外的一个组成部分来源于弹球游戏，虽然不可能打通关，但分数能够决定玩家玩游戏的水平。引入高分榜的作法，目的是让玩家在分数的后面输入自己名字的首字母，然后通过计分榜中排名，玩家可以看出自己究竟是什么水平。只要游戏机一直接入电源，它就会记住上述分数。有些游戏，比如 *Centipede*，在拔掉电源后，仍能记忆高分榜或其中的部分内容。高分榜使经典的街机游戏充分利用了人们玩游戏的一个主要动机——“吹牛的权利”。玩家能把高分榜中她的名字指给她的朋友们看，以证明自己的勇气。朋友之间也可以相互对垒（几乎所有的游戏都有两人模式，玩家可以各处游戏），看看





谁得的分数能高过其他人。

- 简单易学，游戏方式简单：经典的街机游戏对玩家来说非常容易学习，但完全掌握却并不太可能（或者至少是非常困难）。某人来到 Centipede 的游戏机边，塞入硬币，到第三条命的时候，他已经对游戏的进行方式以及如何才能玩得更好有了一定的了解。丢命的原因玩家是一清二楚的。这些游戏通常没有涉及到需要组合按钮“特殊的动作”，玩家必须要经过尝试及失败才能学会使用。也很少有游戏使用复杂的，比如“健康”、“防护”及“能力提升”等概念。商业上的考虑是使这些游戏简单易学的另一个因素。在这些游戏引入之初，并没有一个确立的玩家市场，街边的游戏厅也很少。这些游戏从比萨饼屋及酒吧中逐渐发展起来，在这些地方任何人都可以到游戏机前一试身手。如果游戏过于复杂，或者是让人摸不着头脑的话，人们可能就会被吓走。当然，简单并不都意味着这些游戏“缺乏创意”或“糟糕”；它也可能意味着“一流的”或“精致的”。
- 无故事情节：经典的街机游戏几乎毫无例外地远离试图“向人讲述某种故事”这样一种想法，现代的许多游戏依然如此。这些游戏往往安排一个背景，玩家能够很容易地辨认出来。虽然某些背景涉足于战争、幻想、体育等领域，但许多背景还是围绕科幻这一主题。这其中的许多游戏，比如 Pac-Man 和 Q*Bert 就创造了自己非常独特的背景，与游戏中所表现出的毫无拘束的创造性保持了一致。那些经典街机游戏的设计者们觉得并不需要充实他们的游戏环境，编造一种说法来解释玩家为什么要向给出的目标射击，或者一口口吃掉某种小点，而游戏的乐趣并没有受到影响。

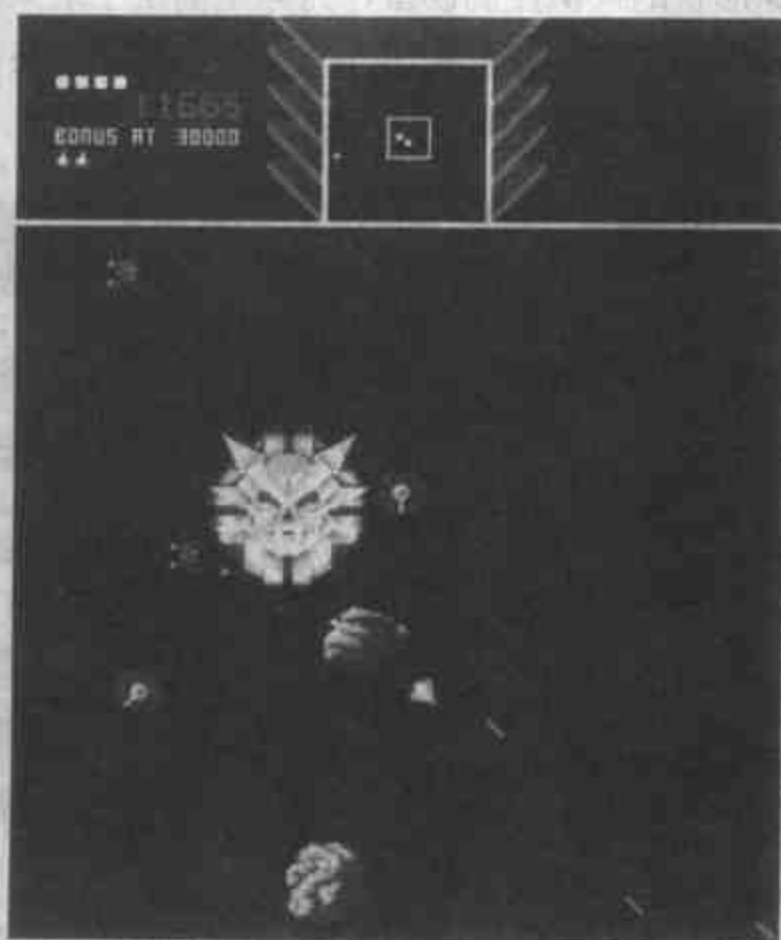


图 4.2 Sinistar 游戏中的战斗并非局限于单屏范围内，但仍被视为经典街机游戏的典范

当然，一些游戏打破了上述游戏在形式上的规则，但仍被人们视为经典的街机游戏。例如，Sinistar 及 Defender 两个游戏都采用了滚动式的游戏环境，供玩家在其中移动，但在任何一个时





间点上, 玩家却不能同时看到游戏世界中的所有场景。Battlezone 这款游戏, 粗看起来确实实和其他早期的经典街机游戏有些另类, 但如果参考上述特点, 大家就会发现这款游戏也是以无限游戏、多重生命以及计分、简单易学、几乎没有故事情节为其主要特点。大体上讲, 上述三个游戏采用的机制都符合经典街机游戏的形式。这样, 我们就可以将这些游戏与 Space Invaders 及 Asteroids 等游戏划分为同一类别, 它们都符合我们上述所开列的所有条件。

作为能够阐释这种形式的一个游戏, Centipede 符合上述经典街机游戏形式各个方面的条件。虽然从今天的标准来看, 这款游戏不是十分复杂, 但它的奇迹却在于, 所有互不相同的游戏元素结合在一起, 创造出了极富挑战性的游戏。所有的一切都井然有序, 高度统一, 和谐平衡。分析 Centipede 这款游戏的主要目的, 是试图弄明白如何才能设计出一款完美的游戏。



4.2 输入

为街机设计游戏最大的一个优势在于, 设计者能够完全控制玩家使用的设备。对于 PC 机, 设计者只能指望玩家有一个键盘和鼠标, 而对街机来讲, 玩家使用控制台, 设计者则要立足于某种控制台所能提供的标准控制器。街机游戏设计者尽管会有种种预算方面的限制, 但还是能够为游戏挑选最佳的控制方式, 向玩家提供最佳的控制系统。然后, 设计者可以围绕这些控制设计游戏, 精确协调游戏规则与输入方式, 使之完美配合。Centipede 非常专业地做到了这一点。它为玩家提供了一个轨迹球形状的模拟控制设备, 完美地满足了玩家的射击飞船在屏幕底部移动的需要。只要当时场景需要, 玩家可快可慢地移动。对于 Centipede 的游戏迷而言, 他们最先记住这款游戏的其中一个方面就是其优异的控制系统。

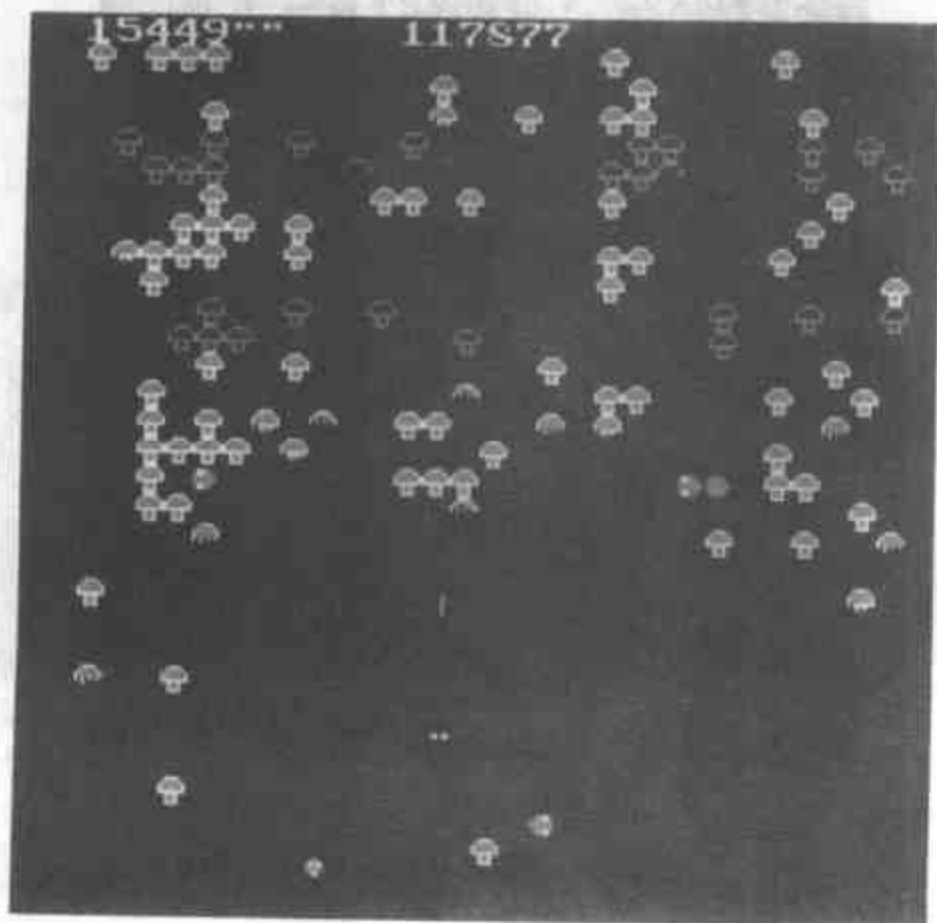


图 4.3 Centipede 游戏中玩家的射击飞船比 Space Invaders 中的更加灵活。除了左右移动外还可上下移动。本图为 Centipede 游戏画面



射击飞船可以对玩家操纵的轨迹球做出非常灵敏的反应，玩家能够容易、直观地理解操纵轨迹球与飞船移动之间的关系。很显然，Centipede 受到了其他诸如 Space Invader 等游戏的启发。在 Space Invader 游戏中，其显著的一个特点就是，玩家的替身被固定在屏幕的下半部分，只允许左右移动，然后发射。Centipede 使这种风格又往前迈了一大步：玩家还是被限制在屏幕下半部分区域，但射击飞船却可以在一个六行宽度的垂直区域内移动，这样玩家就可以躲过下排来的敌人。同时，飞船还是只能向前射击，无法消灭飞船下方的敌人。除轨迹球外，剩下的唯一的控制就是一个按钮，用于从飞船上发射类似激光的武器。游戏允许玩家拥有无限制的快速火力，但在任何一个时间点上，只允许发射一束火力；也就是说，玩家不能仅仅一直按着发射键，而要三思而行。如果玩家把射击飞船移到蘑菇下面，则可以按下开火键，快速向蘑菇射击四次，足能摧毁蘑菇。但在屏幕的上方，玩家却无法在此区域操纵飞船，摧毁一个蘑菇要花长得多的时间，因为玩家在下次发射之前必须等待第一束火力击中蘑菇。在玩家的火力射向远处的目标时，无法因为要消灭向下俯冲轰炸的敌人而再次射击。对于自己的射击，玩家要仔细计划，这种设计要素为游戏的机制增加了一定的深度。



4.3 互联性

Centipede 这款游戏最大的优点之一在于，所有不同的游戏要素完美地结合在了一起。各种不同的昆虫敌人都试图要将玩家置于死地。蜈蚣从屏幕上方蜿蜒而下，水平游向底部玩家的区域。蜈蚣的外表或者是一个单独的、十二节长的大蜈蚣，或者是一些带有若干单独头颅的较短的蜈蚣。一轮攻势开始之初，屏幕上总是会有十二节的蜈蚣，然后是蜘蛛，按斜线方向跳跃式地穿过屏幕底部区域，不断出入于玩家的活动范围。接下来是跳蚤，朝玩家径直落下。这些昆虫的运动方式并无太过复杂之处。实际上，跳蚤和蜈蚣一旦在游戏区域中出现，会以一种完全可以预见的方式向玩家靠近。蜘蛛曲折前进，有着较大的随机性，但是即使这样它也不会真正去追赶玩家。因此，玩家一旦玩过几次之后，应该能够完全预期得到敌人会如何向自己发起进攻。单独与任何一种动物战斗对玩家来说几乎没有什么挑战性，可如果这些昆虫共同行动，联合在一起，就会给玩家提供一个极具挑战性的场景。缺少了上述的任何一个对手，游戏的挑战性即使不会完全丧失，也将会被大大削弱。

在整个的游戏世界中，每种昆虫都与蘑菇有着各自独特的联系。蘑菇存在的主要原因在于，它可以加速蜈蚣向屏幕下方移动。蜈蚣每次撞到蘑菇后，便下移到下面的一排，就如同已经移到了游戏区域的边缘。这样，一旦屏幕上布满了蘑菇，蜈蚣便可以非常快地到达游戏区域的下面，给玩家造成巨大的危险。因此，玩家必须尽其所能消灭游戏区域内的所有蘑菇，即使在某些情况下，蘑菇本身并不会构成直接的威胁。使游戏更加复杂化的是，玩家每打下一段蜈蚣，在蜈蚣被打死的地方会留下一个蘑菇。因此，干掉一个十二节长的蜈蚣会留下大把大把的蘑菇，玩家还要继续与之斗争。



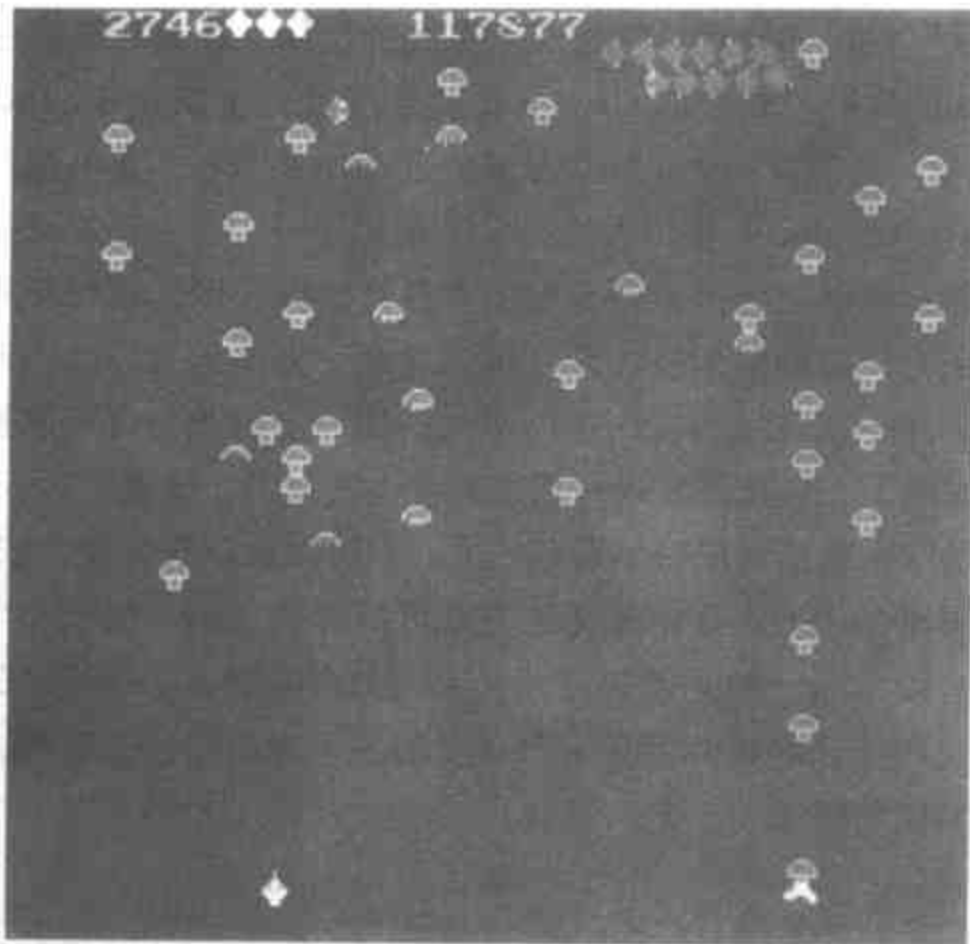


图 4.4 在 Centipede 游戏中，跳蚤朝屏幕下方坠落，将蘑菇抛在后面，而蜘蛛则吃掉阻挡其移动的所有蘑菇

跳蚤朝游戏区域坠落的时候，会在身后留下一串蘑菇，玩家惟一的阻止手段只能是将其消灭。如果屏幕下方的蘑菇少于某个数目，跳蚤才会进到游戏区域。在这种方式下，倘若玩家消灭了靠近的所有蘑菇，便会有更多的跳蚤出来，留下更多的蘑菇。跳蚤这种动物给玩家造成了最大的威胁，它还有一个副作用：吃掉蘑菇。这就使玩家陷入了进退两难的境地：开枪打死还是躲避开蜘蛛，好让它带出更多的蘑菇？最后是蝎子，水平越过屏幕上半部分区域，虽不会与玩家碰撞和试图消灭玩家，却能向身体上方的蘑菇施放毒气。中毒后的蘑菇与蜈蚣相撞，会以一种不同的方式影响蜈蚣。它不是下移一排，而是直接垂直降至屏幕底部。因此，蜈蚣与毒蘑菇撞击后，其带来的威胁较之以前更为严重。蝎子经过后，玩家必须花大力气击落屏幕上方的所有毒蘑菇，或者作好准备，消灭垂直落向自己的蜈蚣。

这样我们就可以看出，游戏中的每一种动物都与蘑菇有着特殊的、独特的关系。正是这些关系的相互影响，才给玩家带来了挑战。跳蚤丢下的蘑菇越多，蝎子就要向越多的蘑菇施放毒气。蜘蛛会在屏幕底部带出蘑菇，然后逃离玩家，可它也会吃掉许多蘑菇，这样跳蚤又开始出现。玩家如果在蜈蚣距离屏幕顶端过近的时候将其消灭，便会在上方留下大把大把的蘑菇，因为距离较远，非常难以消灭，还会导致以后蜈蚣以更快的速度到达屏幕下方。但是，如果玩家一直等着蜈蚣到达屏幕下方时再还击，自己就可能会被杀死。蘑菇的作用就好像是一道难题，使 Centipede 这款游戏成了街机玩家与实时游戏难题的一种混合体。当然，某些玩家会逐渐形成自己的一套策略，能够阻止跳蚤的出现，这样会使蜈蚣以较慢的速度朝屏幕下方下落，玩家存活的时间就会长很多。玩家的敌手与蘑菇之间的相互作用，敌手之间的相互作用，给玩家创造出了一种独特的挑战氛围。



4.4 不断增强的紧张局势

Centipede 这款游戏的一个成功之处在于,在整个游戏过程中,不断加强游戏的紧张性。实际上,游戏创造了许多高潮和低谷。消灭一节蜈蚣后,紧张局势升级为高潮,轻松片刻,游戏切换到下一波进攻。游戏通过跳蚤这样一个小手段在几秒钟内快速升级紧张局势(跳蚤是游戏中唯一需要击中两次的敌人)。被一次击中后,跳蚤的移动速度急剧增加,玩家必须立即再次射击,以免跳蚤撞上自己。在跳蚤短暂加速的那一刻,玩家紧张感随之加剧。单就蜈蚣而言,每次被击中后便会分身,游戏的紧张性也会加剧。如果玩家击中一条十一节长蜈蚣的中间一节,它会分成两个五节长蜈蚣,朝相反的方向前进。毫无疑问,从总数上讲,玩家是消灭了屏幕上的一节蜈蚣,但现在却马上又有了两个对手要操心。因此,有经验的玩家最终会向蜈蚣的头部或尾部射击,以避免蜈蚣一分为二。

游戏中大部分不断升级的紧张局势来自于蜈蚣向屏幕下方的移动,还有玩家想在蜈蚣到达之前将其消灭,而时常付出的疯狂努力。一旦蜈蚣头部接触到屏幕的底端,便会激活一个特殊的蜈蚣头部发生器,向玩家的区域吐出更多的蜈蚣头。玩家如果未能在此之前将蜈蚣消灭掉,这些多出来的蜈蚣头会使局势进一步紧张。这多出来的蜈蚣会不断涌来,直到玩家设法消灭掉屏幕上剩余的一节节蜈蚣为止。蜈蚣头出现的速度随时间的递增不断加快,这样玩家颇费时力将其消灭之后,更多的蜈蚣会更加迅速地出现,玩家会更加忙乱。

一旦玩家消灭了最后一节蜈蚣,游戏就会转入下一波进攻,蜈蚣又会在屏幕上方出现。这给玩家提供了非常重要的、暂时的缓冲时间,来歇口气。打败蜈蚣后,特别是在新的蜈蚣头刚刚产生出现时,玩家都会感觉到强烈的振奋。另外,刚刚产生的蜈蚣好像更容易消灭,因为它产生的地方离玩家区域较远。

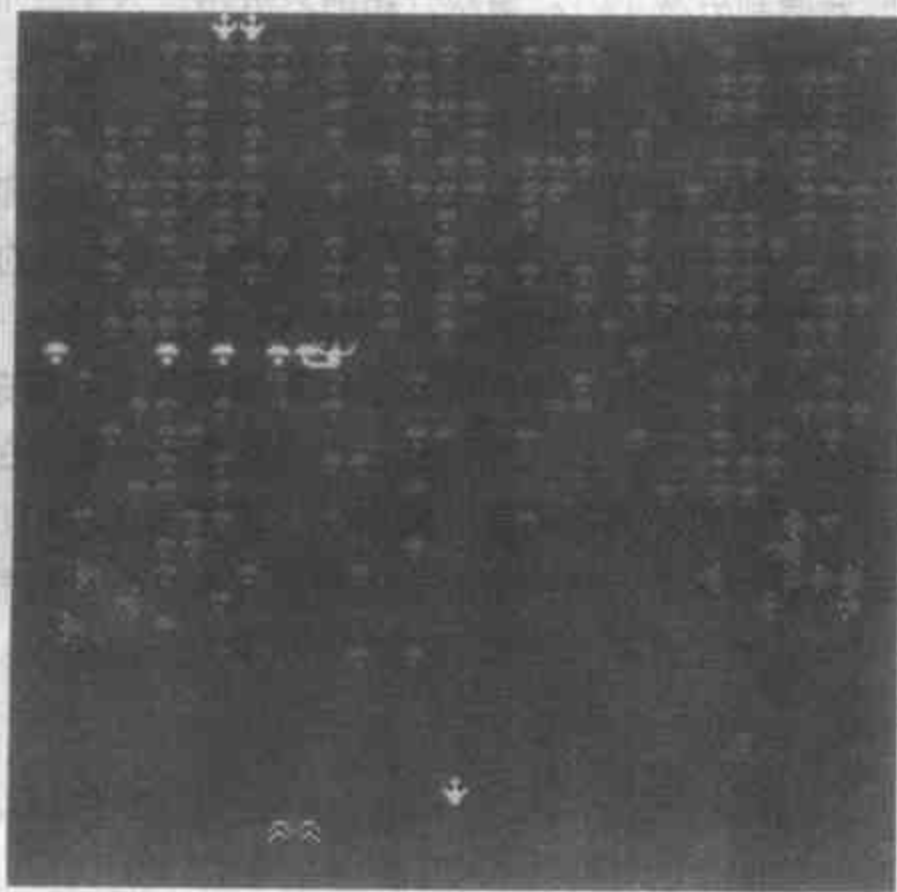


图 4.5 Centipede 在游戏过程中,蘑菇会越来越紧密地在游戏区域排列起来





在玩家的整个游戏过程中，无可避免地会有越来越多的蘑菇，更加密集地占满整个屏幕。当有众多的蘑菇向屏幕下方移动时，如果玩家能把下面的一半消灭干净，即会感觉相当幸运了。因为被下面的蘑菇挡住，因此靠近顶端的一半根本没有机会消灭。同样，若蝎子在屏幕顶端部分丢下一些毒蘑菇的话，玩家无论如何也不会有机会将其消灭，因此，在每一次攻势中，蜈蚣都会向屏幕底部区域俯冲轰炸。随着游戏的继续进行，游戏区域的上方变成了由蘑菇组成的铜墙铁壁。蘑菇越密集，向下移动的速度就越快。当蜈蚣能以极快的速度迅速到达屏幕底部时，玩家游戏的节奏会越来越快，赶在蜈蚣到达之前消灭掉也会使玩家越来越惊恐。蘑菇密度不断增加会加剧游戏的紧张感，这并不仅仅体现在每一轮攻势中类似蜈蚣头发生器所产生的紧张感，而是体现在一波又一波的进攻当中，因为除非玩家将蘑菇消灭，否则它们永远不会离去。

随着玩家分数的增加，Centipede 也会相应增加设计这些怪兽的难度。玩家的分数永远不可能减少，只会增加，在整个过程中，游戏的紧张程度也会不断加强。最明显的莫过于蜘蛛，玩家的分数增加到 5000 时（如果玩家选择“困难”级别，这个分数为 1000），其速度大约会加快一倍。玩家的分数达到一个非常高的数字时，蜘蛛便只在屏幕底部一个越来越小的区域活动，直至最后只从玩家六排大小活动区域中移动出一排的距离。蜘蛛这样被限制起来，会更容易地命中玩家，玩家想要射中它却更加困难。回想一下前面讲过的这种情况：跳蚤从屏幕顶部的下落取决于屏幕下半部分蘑菇的数量。玩家开始游戏时，如果在此区域蘑菇的数量不足五只，跳蚤就会下移，在下移过程中还会丢下更多的跳蚤。可是随着玩家分数的增加，就需要更多数目的蘑菇来阻止跳蚤的出现。玩家必须在上述区域留下越来越多数目的蘑菇，以阻止跳蚤出现，防止屏幕上方堆集大量蘑菇的情况发生。

每轮攻势开始之初，游戏总会在屏幕顶端生成出十二节蜈蚣和蜈蚣头。这就意味着，如果屏幕顶端出现一个十二节长的蜈蚣，那只能是唯一的一只。如出现的是七节蜈蚣，那么就会再出现六只蜈蚣头，总数达到十二这个神奇数字。出现的蜈蚣越多，玩家想要全部击中的挑战性就越大，蜈蚣也就越可能会溜到屏幕下方。游戏开始时只放出一只十二节长的蜈蚣。在接下来的一轮攻势中，会出现一个十一节长的，还有一只蜈蚣头，但行动缓慢。随后的一轮攻势，出现的是一个快速移动、与头部结合在一起的十一节蜈蚣。然后出现的是行动缓慢的十节蜈蚣和两个蜈蚣头。每轮攻势玩家需要留意的单个蜈蚣越来越多，紧张局势也随之不断升级。游戏造出十二只蜈蚣头后，游戏将重新开始，不过游戏变得愈加困难，制造出的蜈蚣速度更快。

玩家的死亡也提供了一个短暂喘息的机会。玩家的飞船被击落后，一轮攻势重新开始，蜈蚣重又返回屏幕上方。但是，在此之前，会在屏幕上重新放置所有的蘑菇。这意味着原先被部分消灭的蘑菇又回到了原来未被消灭时的状态，而且所有被蝎子污染过的蘑菇也回到了未被污染时的状态。在游戏的许多攻势中，蘑菇密度不断增加，想要击落它们几乎不可能，而且由于毒蘑菇的存在，每一轮攻势中，飞驰而过的蜈蚣会向飞船狂轰乱炸。这样，玩家在飞船被击落后，精神上得到了几乎完全的放松，所有的毒蘑菇也会从屏幕上端消失。这样可以使玩家更加放松，至少暂时是这样。



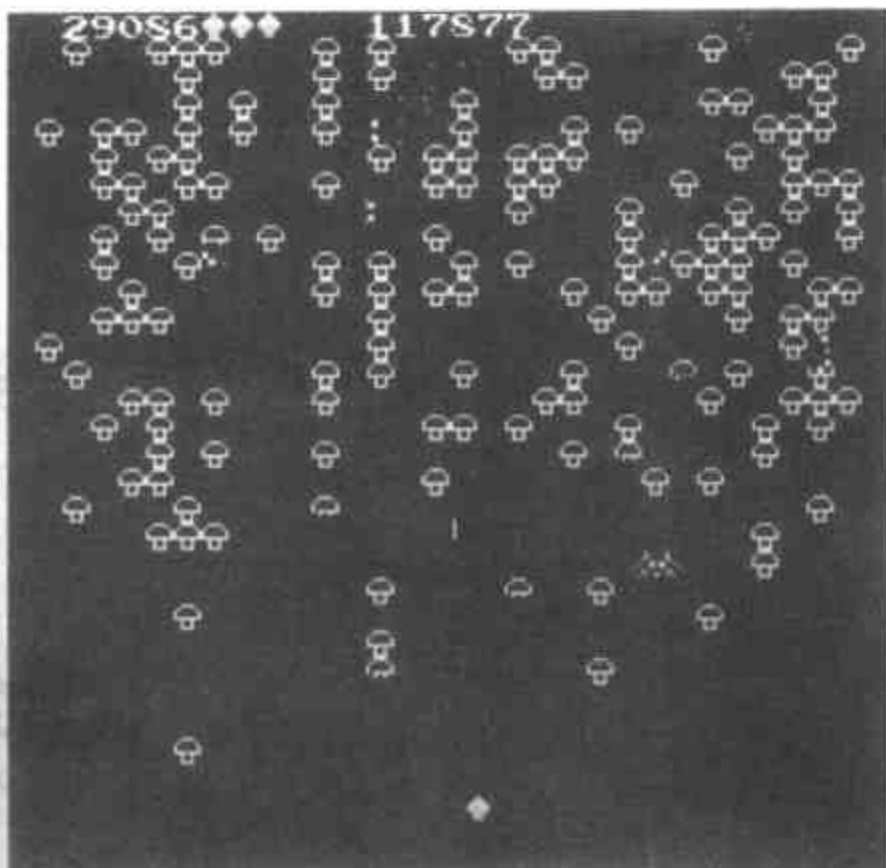


图 4.6 Centipede 这一款疯狂的游戏，虽然为玩家提供了休息的间歇，但玩家在大部分时间里都保持高度紧张

Centipede 这款游戏的非凡之处在于，它为玩家创造并保持一种紧张的局面，同时还提供了行动中“喘息的机会”。现代游戏的设计者们，他们总是关注于不断增加难度，他们应该能够通过分析这款游戏，学习如何使玩家自始至终处于备战状态，而不是不公平地制服玩家。



4.5 人不同，游戏不同

许多人也许会在这款游戏出现二十年后嘲笑它。毫无疑问，比起当代的许多游戏作品，它的确没有令人惊奇的成就，那些没有仔细研究过它的人，很可能会拒绝承认它是一款严肃的游戏，而把它作为一种轻松的休闲娱乐。Centipede 所做的一切是如此熟练灵巧，游戏的机制如此精确，做到了完美的平衡；游戏如此地引人注目，的确是电脑游戏设计上的杰作。人们需要记住的一点是，这款游戏产生于单人单游戏的时代，当时游戏开发小组基本上只有一名成员，在本游戏中这个人就是 Ed Logg。一个人独自控制整个游戏开发项目，那些极富天分的设计者就能体会出游戏中的细微差别，其形成的最终产品极有可能画面清晰，手法技巧娴熟。当然，不仅庞大的开发小组会设计出非常糟糕的游戏，个人也会；但是，我们势必会产生疑问：一个孤独的游戏开发者，是否真的不可能开发出完美的游戏？



Chapter 5

第5章 焦点



“感受水的流动，融入其中，你就能知道它要流向何方。”

——Warrel Dane

20 人的小组用两年时间开发一个游戏远比艺术品的创作更像是一场战斗。许多人认为适当的冲突可以创造出伟大的艺术品，尤其是像今天商业计算机游戏那样合作完成的软件，更是如此。优秀的游戏可能是来自于为了更好的完成游戏的某一方面而产生的争论。如果仅仅是因为争论而导致游戏的四分五裂，那就不可能产生好游戏。在制作游戏的过程中肯定会经历很多的“战斗”，你必须忍耐那些小冲突，挺过每次失败。由于你和其他开发成员或者和自己之间总是存在着潜在的思想冲突，在游戏开发的过程中，开发者很容易背离自己的初衷。有没有可能你只是迷失在自己的想像世界里？为什么这个游戏会很好玩，很吸引人，很特别？是不是你根本就是喜欢电脑游戏？

有时候在游戏开发到一半时，因为出现了技术上的难题，或者是开发组成员之间产生了分歧，开发者会很容易走上岔路；还有可能仅仅是因为开发商告诉你 A、B、C 等特性需要修改。这时候你就会开始怀疑这个游戏是不是曾经很酷，它还能不能完成？这个时候就会显示出优秀游戏开发者和普通游戏开发者之间的区别。优秀的游戏开发者会牢记自己的目标，坚持方向，越过这些困难。



我们在这一章节中所学习到的技巧，不一定是每个游戏设计者都会用到的，但是每个设计者都会从中获益。许多设计者只会使用而不能掌握这种技术。还有一些设计者会用不同的方法去保证游戏的娱乐性，而不违背他们开发的初衷。你不能希望每个设计者都关心：“你现在项目的重点是什么？”而且希望他们回答的和本章探索的方法一样。如果你在项目的开始就能严格地贯彻自己设计的初衷，我想，你的游戏质量将会给人一个满意的结果。



5.1 建立焦点

一个游戏最重要的就是要体现设计者的想法。在本章中，我鼓励设计者用一小段话记下自己所要设计游戏的重点，这样可以时刻理解和牢记游戏设计者的想法。这是设计者要注意的最重要的一点。在一定程度上，一个游戏的焦点类似于一个公司的“任务陈述”，就像一个有意图的任务陈述能够指导公司决定那样。

作为一个游戏设计者，要知道你自己开始设计时的焦点是什么，并从这个焦点开始。在这个项目的初期，在你开始搜集相关资料和拟定主要的设计思路之前，应该问问自己有关这个游戏的几个问题：

- 这个游戏最无法抗拒的是什么？
- 这个游戏要去完成什么？
- 这个游戏能唤起玩家哪种情绪？
- 玩家能从这个游戏中得到什么？
- 这个游戏是不是很特别，与其他游戏有何不同？
- 玩家在游戏世界中该控制哪种角色？

基于这些问题，你就会知道计划中游戏的核心是什么。如果你很难回答这些问题，现在就考虑这些直到能有个准确的答案。现在——在那些人开始处理这个项目之前，在“烧钱指数”被消耗和游戏的预算被抬高之前，在营销部门开始影响游戏的内容和方向之前——是集中处理这些问题的时候了。只有坚固你早期建立的游戏意图，才能有机会看着它实现。

如果你对这些问题没有太大困难，你可能要记述整页或者更多的游戏不同点。但是一页也太多了。我们必须使这些与游戏焦点相关的句子尽量简洁，最多只要一个小段落。最好是在你的同事目光变呆之前能够读完。应该记下任何你注意到的问题，并用一个简短的语句概括，一个中等大小的段落。坚持你最无法抗拒的主意。你不用列出游戏的每个单一特性，或者是任何与其他游戏的不同点。保存你对游戏的想象力是最重要的，如果你丢掉了想象力，那么游戏的弱化将是不可避免的。

如果本来没有游戏背景，就不用把它包含在游戏的现行焦点中。如果你的游戏是虚拟、科幻或者是20年代的罪行小说背景，如果游戏要表达在紧张环境下的人物关系，或者是包围战的精妙，这些都没有问题。如果你要完成的游戏不需要游戏背景，那么不要考虑它。当然，最初的设计动机也无望与背景交互。如果你的设计是基于某个想法，比如说美国某个小镇中装束特别的超级英雄，那么游戏情节就会根据这些特征在特定的场景下展开，这些都会成为你所关注的焦点。游戏中大部分重要的概念要逐步完善，这也是游戏设计中独有的焦点。所有的这些都





要保留在重点之中，没有这些元素，你的游戏也不会长存。

焦点能抓住游戏内在的东西，激起你的创造源泉，并且使你感觉非常愉快。如果在这个最初的阶段还不能使你害怕，那么当交货期一再延期，预算失去控制，项目还差 3 个关卡，而且得力助手跳槽到别的公司时，你就难以再积聚热情去完成它。Chris Crawford 在他的《计算机游戏设计技术》中简略地提到游戏的焦点，如同他呼唤的游戏设计的目标一样：“这是表达自己的一个机会，选择一个你相信的目标，一个能表达你的审美感和世界观的目标……无论你的目标是什么，只要它符合你自己的兴趣、信念和热情。”如果你不相信自己的游戏，你将永远做不出完美的游戏。

即使你有工作合同的约束，或者与一个专横的开发商合作，或者在一个自负的程序主管手下做事，你也要能为项目做出自己的目标。如果在设计游戏的工作中需要你从事自己不感兴趣的事，你要考虑一些方法使自己能愉快地接受并完成它。如果给你足够的时间，你也无法有所成绩，只是有一种被迫的感觉，没有比这更糟的情形了。你要用热情和斗志去完成你的游戏，直到它最后运行。

许多章节已经记下这种方式，暗示你负责的项目，至少来自游戏设计的观点。当然，实际情况也许不是这样。或许你是项目组的一个成员，或许你才工作不久，总之你是资历最浅的一个。这不能做为你无法决定游戏的重点是什么和能否保证游戏的预知方向的托辞。但愿设计主管已经决定项目的目标，并把这些目标包括在介绍游戏设计的文件中。如果你找不到这些资料，应该找主管好好谈谈，询问计划是不是真的在做，但不必用对质的方式；如果你有关于这个计划走向的好的主意，这将有助于游戏朝更好的方向发展。

如果游戏设计主管的心中没有明确焦点，或许有其他的组员知道，可能是编程主管或者是美工主管。然而，尽管你尽最大的努力去研究这个计划，但这个计划看起来依旧没有方向，那么你可以通过自己的努力找到问题所在。通过与别人讨论，找到它的方向。与其他的设计者、美工、程序员和开发商沟通，尽力告诉他们有关这个游戏的一切，使每个人都赞同你的观点。会议是完成这些事的最好方式；当团队中存在一些由不同观点和性格产生的矛盾时，通过会议都能被消除。也没有必要为了保证游戏的方向性而担任领导职务。作为游戏开发的全权设计者，你要使游戏有个清晰的发展方向并且让游戏出现美好的未来。



5.1.1 一个例子：Snow Carnage Derby

让我们假设有一个包括雪上汽车和格斗的游戏场景。它会使你在驾驶雪上汽车时而感到刺激吗？它是否正在冒险穿过加拿大西北方的领上，尝试写实地模拟一个很棒的雪上汽车作长途旅行？不？或许让你前进的雪上汽车看起来更像一个有趣的能驱动的交通工具，而你在一个安全的游戏环境中享受操作的乐趣；在那里，你可以在雪地上跳跃和旋转而完全不会使自己受伤。在这种情况下，不是驾驶汽车的问题，而是驾驶汽车的乐趣；在这个环境中，允许你有很多很酷的机动动作。既然雪上汽车部分在你的想法中非常重要，那么必须将它归纳到游戏设计的重点之中。你的说明可以用这样的句子来表达：“游戏者的体验就是对实际上不存在的驾驶雪上汽车的一种身临其境的感受，游戏者能在置身雪景的移动和跳跃中感受到乐趣和挑战；游戏的平衡点不在于现实而在于乐趣”。





现在，什么是抓住你注意力的战斗元素呢？你将看到鲜血洒在雪地上，雪上汽车互相撞击，车手为了活命而悬挂在汽车上，并且不顾一切地抓住车把避免被汽车甩下的景象。这些雪上汽车为何战斗？这并不重要，只要像战争那样刺激就好。这场战争为何发生无关紧要，雪上汽车互相冲撞的场面会使你激动万分。或许在审查严格的游戏零售商面前你会有些麻烦，但这一切只是你的想象而已。因此应包含一个有关战斗性质的句子：“游戏产生了一种发自内心的颤动，考虑到死亡或者使敌人的汽车驾驶员丧失战斗力，游戏还想用暴力最大限度地制造出戏剧性的效果。”

在你的想象中，关于雪上汽车战斗游戏还有其他的重要部分吗？你想逼真地模拟燃烧和被毁坏汽车的情景吗？是否固定你的汽车作为游戏的特定部分？不真实；它似乎可以加到游戏中去，无需加入你的想象。游戏将是3D模式还是2D模式？好吧，实际上，游戏可以在任一模式下运行。在现今的商业化市场或许更需要游戏的3D化，但这对你的想象并不重要。在你的焦点中，不包括游戏方面，但需要获得更多的赞助并完成出版更多的作品。稍候你才能考虑它的商业运行。你有关心自己想象的权利，因此你不需要考虑是2D还是3D。实际上，或许你做每件事都有它的重点。但记住，焦点一定不要太长。

现在，可以把两个句子合成一个段落并且给游戏命名了。尽管这看起来还早，但这个时候给游戏命名是非常好的主意。你需要让你的工作组、营销部门以及商业人员尽快喜爱这个游戏，而且有一个他们能够提到的游戏名，这是非常重要的。他们会认真地讨论：“这个游戏是Richard的构想吗？”给游戏命名比实现你自己的想法更现实，这看上去也没什么不合理的。

用你最大的努力去讨论那些你喜欢的名字，直到游戏被最终生产出来。通常，游戏早期的名字都能保留到最后。不要选择一个白痴般的游戏名字也就显得尤其重要。举个例子，让我们称你的游戏为Egyptian Rumba。你的工作组一直沿用这个游戏名Egyptian Rumba，他们开始将你的好游戏与这个愚蠢的标题相关联，工作在美工组的人开始赋予一个着埃及图案的人物。故事组的成员消耗大量的时间去思考这个游戏为何叫Egyptian Rumba，并以这个名字为主线，展开一段特别巧妙的故事。如果稍候你想更改名字，你的同事将会难过甚至可能会愤怒，对他们编的故事也不会再有任何感觉。甚至“suit”也开始喜爱Egyptian Rumba这个主题。他们认为他们将能够占领地质冒险市场和古巴的跳舞市场。而且很快，如果你还记得，你将会说这是该改变游戏名称的时候了，但是每个人都会说，“为什么？我们喜欢Egyptian Rumba！它是最好的名字！”你只好附和。当大众看到你的游戏摆在货架上会想：“架上放的是什麼？名字听起来真愚蠢，”而且会很快地传递比较合理的游戏名称。

所以，最后你选择了Snow Carnage Derby。可能后来你想出了一个更贴切的名字，但这个也可以凑合着用。现在，把你的说明一点一点的集中成一个段落，尽量写得整齐简洁一些。参照游戏目前的状态，就好像它已经存在了。用“Snow Carnage Derby是一个令人振奋的……”代替“Snow Carnage Derby将是一个令人振奋的……”这样，人们在读到你的说明时会对你的游戏留下更具体的印象。在某些方面表现出来的不仅仅是一个说明，要是在脑海中已经是一个游戏该有多好。另外，需要避免的是用一般的话语进行描述，这实际上不能给读者提供任何有效信息。例如：“Snow Carnage Derby是一个高质量的、有趣的……”当然，它或许是有意思。但有人会去制做一个枯燥的游戏吗？或者低质量的？要去掉说明中那些不能表达游戏重要信息的部分。





把焦点的各部分都集中起来，你会得出以下结论：

Snow Carnage Derby 令人振奋，它是一部雪上汽车快速行进的破坏性游戏。游戏者的体验就是对实际上不存在的驾驶雪上汽车的一种身临其境的感受，游戏者能在置身雪景的移动和跳跃中感受到乐趣和挑战；游戏的平衡点不在于现实而在于乐趣。游戏产生了一种发自内心的心颤：考虑到死亡或者使敌人的汽车驾驶员丧失战斗力，游戏还想用暴力最大限度地制造出戏剧性的效果。Snow Carnage Derby 演绎了一种令人心颤的快速运动，在这个运动中，游戏者要在避免毁灭的同时尽量减少竞争对手。



5.1.2 焦点的作用

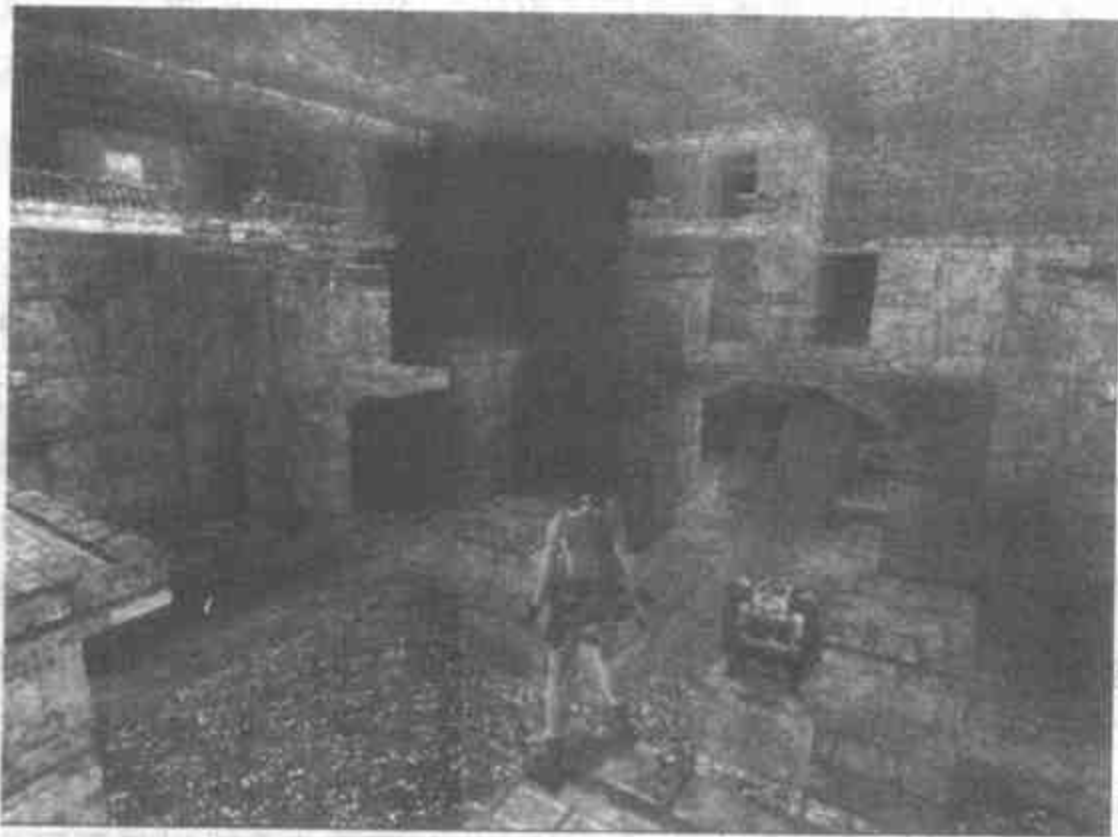


图 5.1 你的游戏可以与其他的游戏类似，比如 Tomb Raider，但是你必须要在焦点中表达这个游戏本身的方向而避免与其他的游戏做比较

尽量不要参考其他游戏的焦点，要让自己的游戏焦点能够描述出自己游戏的本质和精髓。如果你的说明是“Voltarr 就像 Tomb Raider，但被安排在一个叫 Dongo 的古怪星球上，那里有激烈的激光枪战”，那么人们很难理解你的说明并很难弄清楚所描述的那个古墓丽影是什么意思。看一看古墓丽影再决定它的重点可能是什么。然后根据重点，去掉对游戏无关紧要的部分，加上能使游戏更实质化的新想法。机遇就是你的那些关于“古墓丽影”的能给其他人不同理解的令人感叹的想法。当你的同伴读到“就像‘古墓丽影’”时，很可能她所想到的游戏场景与你的想象不同。当然，我们是假设他已经玩过“古墓丽影”了。既然设计出来的焦点是要引导你和小组的同伴们，那当每个人读到它的时候需要得出同样的意思。即使这个焦点一开始是给你自己用的，在分析“古墓丽影”的过程中将决定你想摹写的，并将帮助你更好的理解你自己的游戏。你需要一个适当的能够独立运行的最新式焦点，而不用去考虑正在使用这一焦点的人是否懂得其他特定的游戏。所有与焦点相关的重要信息必须被包含在焦点之内，并且不用其他的参考书目。常常出现这样的情况：当设计者们着手创造“像 Game X 但……”的游戏时，他们



总是不会注意到自己所模仿的游戏从一开始就带有很大的局限性。于是，为了掩饰自己的游戏和“Game X”差不多这一事实，继续给自己的游戏打上拼凑的印记会让游戏变得十分愚蠢。把你的焦点中的参考资料拿到其他游戏中有助于展示你正着手设计的真实特性。

为你的设计项目建立一个说明对限制游戏的范围没有什么用，而且也不会有这种可能。你的游戏仍然可以是一个大规模的、复杂的具有英雄气概的游戏。事实上，如果恰当的话，这种复杂和深度或许可以在你的说明中提到；为了简洁地表达计划的着重点，须用一些词句对这个游戏描述一下。你甚至可以在游戏中总结说明：相同的游戏有着不同的玩法。比方说你的游戏模拟的是海盗的生活。你肯定是想模拟一种在海上航行的探险模式，或是在战争中占领另一艘船的作战模式，或是在刀光剑影中打败敌方将领的搏击模式，甚至是一种买卖战利品的交易模式（实际上，Sid Meier 已经编过这样的游戏，叫做 Pirates!）。但是拥有这种多样化的游戏结构，诸如“通过大量不同的战争方式重新创造一种海盗生活不同侧面的游戏”，并不意味着你设计的焦点就能吸引玩家与海盗拼死搏杀。玩游戏的人能够体验到做歹徒的刺激，包括经济上和身体上的冒险。如果你的游戏有多种独立的玩法，你的焦点就应该能应用于计划中的每一个不同的子游戏。

如果你正独自设计或者和几个人一起合作，那么没有必要描绘出你的焦点。毕竟你仅需要解释给每一个需要知道的人听，又何必犯傻把它写下来呢？这里我要说明一下，描述焦点的关键是抓牢你要完成的游戏本质。脑海里的想法和写在纸上的句子会是两种截然不同的结果。那些写在纸上你能看见的句子真正确定了你脑子里的想法，代表了整个计划中最重要的东西。起码，当你向别人描述你的计划时，在纸上不会出现，“是，是，这是一部分，这里是另一部分，我说的实际上是这个意思。”如果没有写在纸上，它就不是游戏的焦点了。读到纸上焦点的人们，即使得不到你的解释，也能够理解你的想法。我发现，写下来的焦点能真正使游戏所要达到的目的变得更清晰更明确。

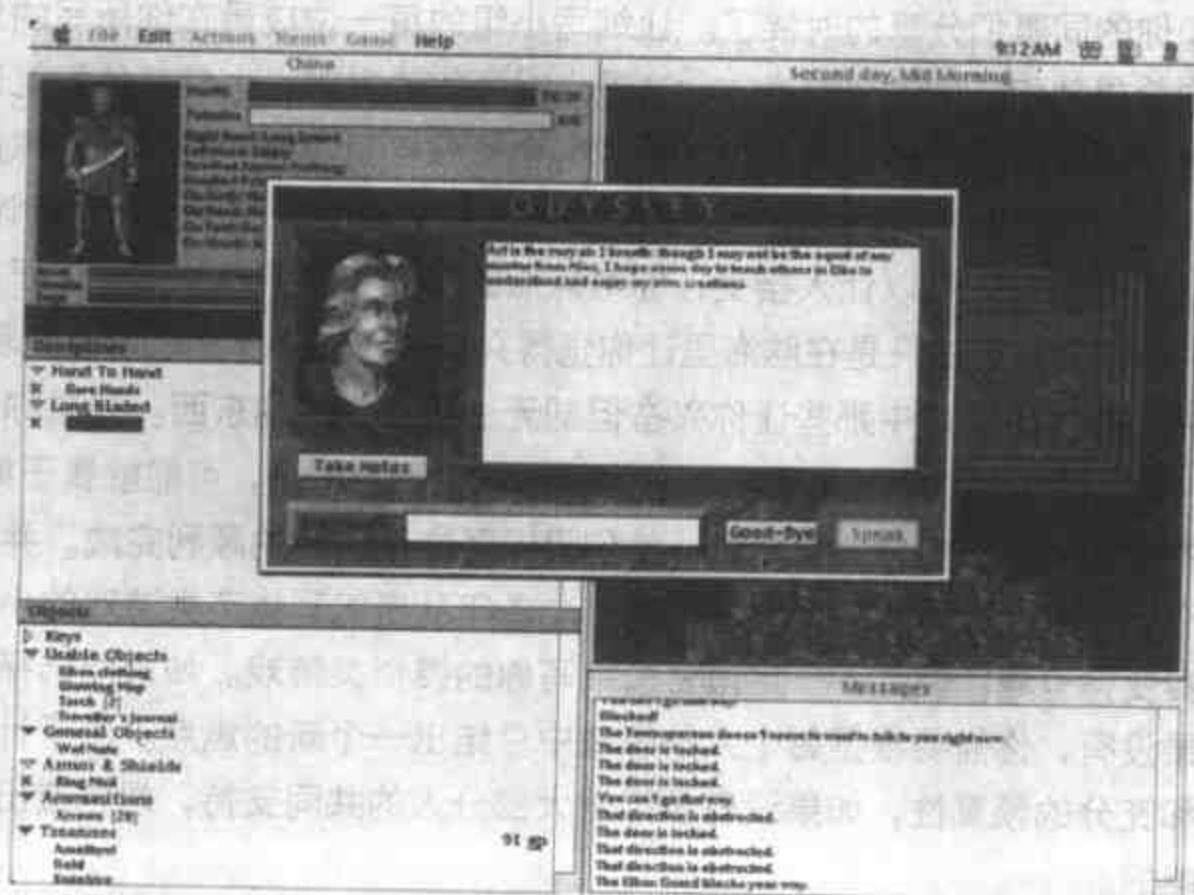


图 5.2 尽管当时我不知道游戏的发展方向，但是 Odyssey 是以一个特殊的故事为主线的





当我编写第一部游戏 *Odyssey* 时，我没有一个做焦点的完整计划，也没有认真的坐下来想过这个问题。我似乎还记得围绕这个故事反复思考的最初目的。故事讲的是一个有强大魔法的疯狂魔法师，他酷爱科学幻想，常拿一些倒霉的类人动物进行试验。他宁愿做社会试验而不愿做生物试验，因为他喜欢看人们在某些特定的环境下会怎样对付别人。实际上，他是在研究所有有意识的生物邪恶的一面。所以“奥德赛”的焦点是要揭示在特定的环境下一些人对另一些人采取的邪恶方式；同时还想给玩游戏的人创造一种氛围，让他们亲眼目睹并有机会去真正改变他们的生活。我的脑海里也有一些非线性和多样化的解决方法，所以我打算不只是设计出惟一的正确方法，还要让玩游戏的人能够找出不同的战略方法来解决遇到的问题。于是我就有了自己的焦点。在还没有真正想出焦点的时候，我已经决定怎样去处理我的游戏了，而且在继续创作这个游戏的过程中我能始终坚持这一点。既然基本上是我独立完成这个计划，我就不用来告诉其他人这个焦点了。虽然我清楚自己想怎样完成这个游戏，但却无法向别人解释清楚我的目的和想法。现在回过头来，我可以总结如下：

在奥德赛里，玩游戏的人可以在剧情丰富的故事里探索到人类邪恶的天性，可以看到人们在某些特定的环境里会用极不道德的方式对付别人。这是一个简单的 RPG/冒险游戏。虽然刀光剑影和魔法巫术充满整个游戏，但掩盖不了游戏的主线。游戏的主线允许多样化的解决方式和随时可能出现的非线性的情节，这样玩家遇到游戏中的 NPC 时才能体会真正的改变。



5.2 坚持焦点

一旦你把焦点写下来，就会对它感到满意，当你读完时会说：“是的，当然，这就是我要的。”这也是和你的同事们分享的时候了。让编写小组的每一位成员在你所写的焦点上签字是非常重要的。你希望他们认同这个焦点——这是小组合作的方向，也希望他们能认同自己，并将看到一部惊人制作会因此而最后诞生。如果你的合作伙伴里没有人觉得你的焦点很吸引人，或者尽管你为此尽了最大的努力也没有人觉得有意思，你只可能得出以下两个结论之一。第一，可能是你的想法不那么好，难以让人接受，你对焦点的陈述和对游戏的描述过于简单而又缺乏新意和诱惑力。或许你的想法只是在脑海里让你觉得兴奋，写出来却不能恰当地表现它的观点。这时你应该回去尽力找出游戏中那些让你兴奋但却无法揭示焦点的东西。如果你坚持认为你的游戏是出色的，而且你的焦点也恰恰能揭示它为什么这么出色的话，可能就属于第二种结论了：这个聚集起来的编写小组仅仅是一个错误，他们阻碍了这个游戏的顺利完成。并不是每一个编写小组都能够有助于每一种游戏的完成。比如一个多年从事编写体育类游戏的小组，他们知道怎样使一场比赛变得有趣，但并不一定适合去编写你的模拟类游戏。如果能选择新的合作小组当然最好，如果没有，你需要根据每个人的想法中总结出一个新的焦点。由于计算机游戏具有新式的合作性和充分的预算性，如果没有小组里大部分人的共同支持，想创造出一个好的游戏实际上是不可能的。

在设计一个游戏时，如果绝大部分是靠自己的努力完成，而其他人的贡献明显比你少，你





根本就没有必要推销你的观点。当然，由一个兼设计者/程序员/美工的人设计出来的游戏可能是电脑游戏中最具焦点的游戏。既然一个人创造了游戏内容的绝大部分，他就能运用独立的操控对每一个细小的差异进行控制。独立完成的游戏的发展通常不再是那些能养家糊口的方式，但我知道有一些人这么做。当然，事实上一个大部分都是由个人创造出来的游戏不会成为流行的焦点。如果个人自身没有头脑或者没有中心思想，对游戏有用的机会也一样不会被利用起来。即使她是个明智的、有头脑的人，如果在完成计划的过程中抓不住游戏的焦点，游戏也可能以没有焦点而结束，成为一个缺乏合作性、没有计划、拖沓的游戏。

对于一个正和开发小组一起开发编制游戏的设计者来说，确定其他的合作者，包括美工、程序员和制造商们，能明白这个游戏焦点的特性是非常必要的。没有一个坚定的焦点指导行动，程序员和美工们可能搞不清楚游戏要怎样完成，可能会把你的游戏做成他们想象中的另一种样子。他们自己并没有什么过失，但结果却把你的游戏做得面目全非。所以你不得不说：“不，这样不合适，重做一遍。”如果制作小组有一个焦点可以去把握，而且能够记得去把握，那他们几乎不会做出与你的游戏不相符的工作来。明确的焦点让你不至于花大量精力去盯着美工和程序员的工作，当然这样也可以省掉很多麻烦，不必过于频繁的对他们指手画脚。

5.2.1 生动化焦点

一旦小组对这个计划表现出热心，能抓住焦点，知道游戏应该做成什么样子，就可以通过一份完整的计划书把你的想法完全表达出来。为了使人们能更快地了解你的想法，你甚至可以在计划书的一开头就表明焦点，作为对这个游戏特性的总结。（对计划书的特性和创造，在第17章有更详尽的介绍）计划书应该使你的游戏围绕和扩充你的焦点，详细说明你的焦点是怎样通过游戏来实现的，其中游戏又是怎样准确发挥其功能的。你还要草拟出整个游戏的大致过程，比如游戏场景是什么样的，玩游戏的人会遇到什么样的物体。当然，在做计划书的时候还应该多包括一些对可能出现的问题的正确解决方案。控制系统应该用方法A还是方法B？哪一种环境更能吸引玩游戏的人呢？玩家会遇到何种的敌人？设计出一个合理的焦点可以让你当做参考来解决在设计过程中遇到的很多问题。完成了以上的工作，应该继续参考焦点来检查要增加的其他东西是否和焦点相匹配。你必须认真考虑增加的游戏场景是不是会把整个游戏带到另一个方向。区分那些与你的焦点相违背的附加物非常重要，然后修改或删掉那些错误的附加物。

要避免那些特色，或是叫做“酷”的东西，它们并不支持游戏的焦点而且会使其变得臃肿，还会分散游戏者的注意力。你的焦点就是很好的工具，依据它，可以制止这种过度的膨胀，砍掉游戏中琐碎的片断，只保留主要场景。你和你的合作伙伴们的许多想法都是很好的构思，但如果这些构思阻碍你们顺利地继续工作，就不值得探讨和进行下去了。但也不要把这些想法完全抛弃，把它记在笔记本上，以供下一次设计游戏时参考使用。如果是好的想法，或许在其他游戏中可以被很好的使用。如果它们确实非常好，甚至可以再为此设计出另一个游戏。但为了正在继续的工作，你必须参考焦点来决定这些额外的、“酷”的特色，就整体而言，对你的游戏是有好处还是有坏处。

当计划书和生产前其他要素都准备好之后，你的游戏制作工作就可以全面开始了。这时，工作小组里的程序员、美工和其他员工开始为你在计划书里的规划努力工作。在计划的进行当





中会出现很多问题，你的计划书里不可能包含能使整个游戏成功运行的所有要素。你要面临的问题可能是怎样表现这个特色、除了新想法之外怎样改进这个游戏等等。面对这些问题，应该再一次参考你的焦点来指导整个计划的方向。你始终抓牢焦点了吗？它扰乱了整个游戏的紧张运行了吗？这种扰乱有没有引起更多的分心？对焦点的陈述能始终贯穿整个计划，使你的游戏创造保持正确的方向，以做出更好的作品。游戏者知道，有焦点的游戏和没有焦点的游戏有什么样的区别，即使他们不会用这么多语言来表达他们的感受。他们会乐于玩有焦点的游戏，并会很快丢弃那些没有焦点的游戏。



5.2.2 改变焦点

有时候，因为种种原因，游戏的目标也会改变，即使你的计划书已经做好或者游戏已全面投入开发。你可能发现原来的焦点有不足之处或有很大的错误。在创造游戏的过程中，你认识到你的游戏可以提供一个更令人注目的体验，而这一体验超出了原焦点所能提供的范围。根据计划的进展，你可能会改变焦点。当你还处在做计划书的阶段，这种特殊的决定不算痛苦。实际上，如果不站在常人的观点上，当你在做计划书的时候，应该期望你的焦点能多修改几次。想要找出原来构思的漏洞和不足之处，尽力写下所有与游戏有关的重要信息其实也算不了什么。

抛开计划书不讲，在开始着手制作游戏的时候就可能会发现你并不希望看到的弱点和漏洞。因为之前你只是在头脑中构思游戏场景而不是真正在电脑面前来操作。在这种情况下，改动焦点并不会对你的计划产生毁灭性的破坏，也不用去重做太多的工作。在游戏研制过程中修改问题、修改焦点总比剩下的计划陷入困境而做出低劣的游戏要好过。

改变焦点的时候，应该像刚开始建立焦点时一样认真。确保改过的焦点能完全表达出你对整个计划的想法。如果焦点发生了本质性的变化，你需要告诉你的小组并获得他们的认同。记住，你需要你的小组在背后支持整个计划以确保其成功；如果改变的计划让他们失去了继续工作下去的兴趣，那你就得再考虑一下是否坚持这种改变了。

不管因为什么原因对焦点进行了改变，考察游戏的哪些部分应该保留和看看这些部分与你的新焦点有多大分歧是非常重要的。检查一下你的计划书并把你的新目标融入进去。考虑一下哪些游戏技术可以放进去，看看它们是否对新焦点有利；检查所有可能存在的标准（希望在这点上不会出现太多问题），看看它们是否适合新焦点。不管是不是在计划书中出现，代号、标准设计或者是美术等，任何不适合的都需要重做，这样才能让新焦点得到更好的支持。

如果太多的有用资源（关卡、对话或者美术）需要重做，或者是离交货时间太近，或者是没有足够的资金，你应该重新考虑有没有必要改变你的焦点。原来的焦点仍然可以做出有趣的游戏，还是它从本质上就有很大的缺陷？如果你不是坚决要改变焦点，或许那些有用资源还可以再用？你做出的最坏决定可能就是创造游戏中任何需要支持新焦点的新资源，而那些旧的有用资源仍然支持你先前所信奉的旧焦点。这对游戏者来说是显而易见的，你的两种焦点最终会导致游戏的分裂，不是使它具有焦点，而是完全不具备任何焦点。尽最大的努力给你的设计制定一个具备焦点的计划，这样才不会让自己变得没有计划性和预见性。意识到你的设计并不如预想的那么好，但缺少时间和资金去恰当地改善它，这才是最困难的境地。在这种困境中，找出最好的解决办法可能是最具挑战性和最让人沮丧的事情。





做 Centipede 3D 时，在刚开始做计划不久我们就停下来改变焦点，结果大量的工作需要重做，但最后还是使我们的游戏显著地增强了。于是，这个由 Atari 原创的 Centipede 自从被 Ed logg 修改成 Centipede 3D 以后也就成了一个典型的、具有更多爱情的特殊游戏。在做修改或写续集的时候，一定要看看被改变对象的原始游戏版本，而且自己要非常明白它的焦点所在，这对启发你进行修改很有必要。当然，我无法描述 Logg 在 1979 年到 1980 年时是怎样创造原著“Centipede”的，但我能描绘出它的焦点所在：

Centipede 是一个包含着各种各样对手的快速射击游戏，玩家必须杀死他们以求自保。敌人在完全可预见和可先预定的模式里活动，这些仿生者的活动和游戏场景里的其他物体的结合给玩家创造了一次具有挑战性的体验。玩家试图尽力改变游戏场景让对手的活动变得更加可预见，而且玩家可以马上看到整个游戏场景。游戏可以继续直到玩家死亡的次数达到某个具体的数字，玩家在特定游戏中的战绩如何则用累计点数来表现。在 Centipede 中没有胜利和结束。

这个焦点作为游戏焦点来说可能太长、太过于详细，但对我来说，要读懂 Ed logg、弄清他在做“Centipede”时的主要想法是非常困难的，所以我总结了我在这个游戏中能找到的最重要的部分。或许，他所用的焦点和上面所描述的根本没有一点关系。

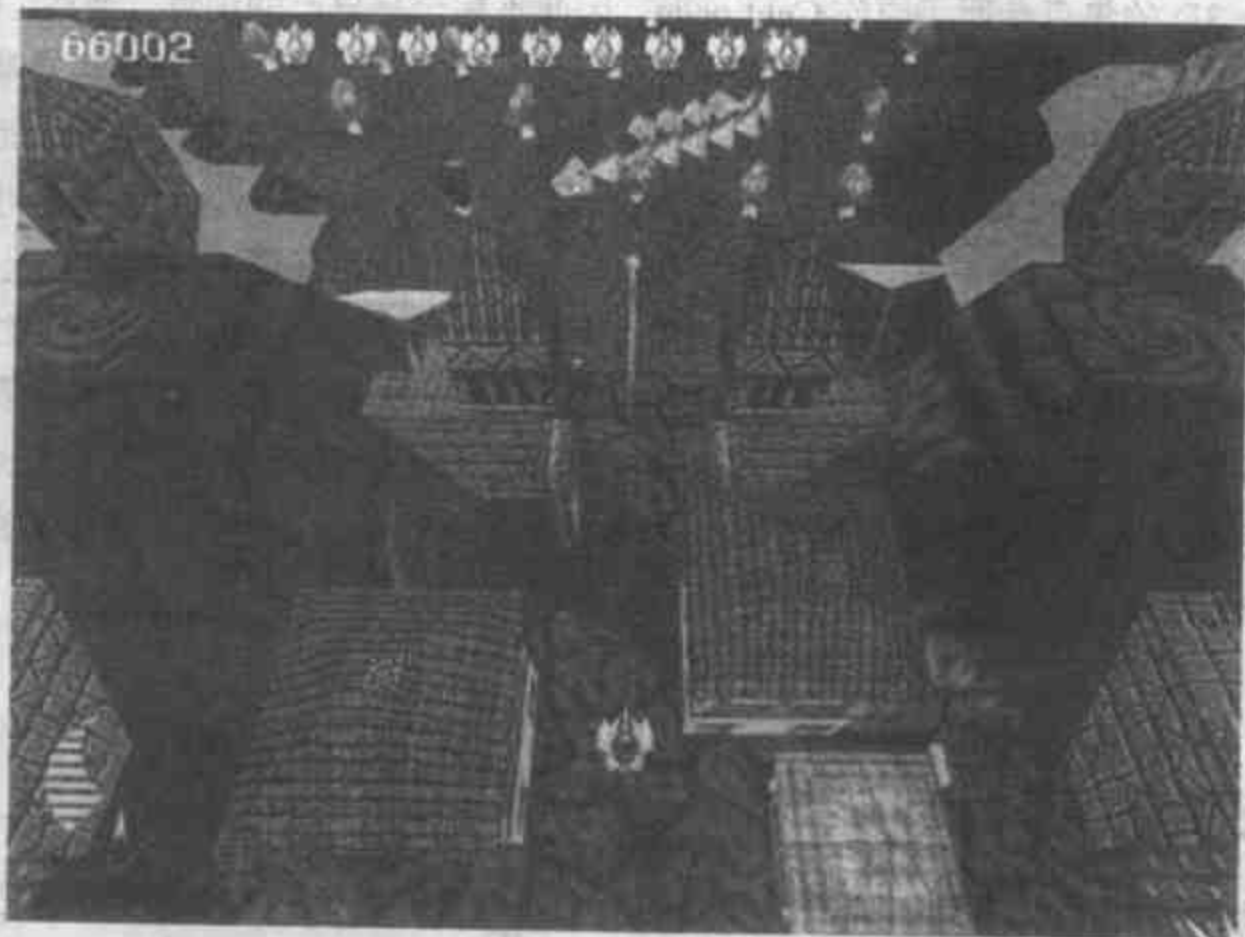


图 5.3 Centipede 游戏中 3D 版本的焦点所要创作的游戏就是，要在三维的、基于关卡的环境中获得原来街机游戏的可玩性

在 Centipede 3D 最初的开发过程中，人们的想法只是采用 Centipede 中的基本角色——游戏者的射击船、蜈蚣、蜘蛛、跳蚤和蘑菇——让他们在三维世界里相互作用，并没有把太多的注意力放到原作中游戏的技术上，也没有过多关注 AI 是怎样和其他角色合作发挥作用的。从原作中提出来的这些要素被更多的用于美学而不是其他的方面。当我们最初的游戏原型变得不再那么好玩的时候，我们决定在新 3D 版本中尽量改善原来的游戏场景，对 1981 年版“Centipede”





的一切进行任何可能的模仿和更新。在寻求新焦点的过程中，我们发现，学习比较这部经典游戏越深，新游戏中的乐趣就越多。虽然当时没有写下我们的焦点，但你可以从下面的字里行间看出来：

Centipede 3D 是对街机游戏 Centipede 的改写，它尝试着让原来的游戏变得更好而且把它移植到 3D 环境中来。原来的“Centipede”的特色是在波浪中进行快速射击战斗，取胜的关键是游戏者对船的熟练操作，而且对手完全是在可预见性的模式里活动。不像在 Centipede 中，整个游戏只用一个关卡，Centipede 3D 会带领游戏者在不同的关卡中间行进。新的游戏还利用了一种具有先进控制的游戏模式，比如可重演的关卡、奖金目标、障碍航海等。然而 Centipede 3D 中的一部分活动和战斗几乎是对原来游戏的重复和回忆，在任何地方都使用了同一个 AI，还保留着原始场景的感觉。

在新焦点里面，我们要开发的有效资源被改变，大量的关卡被丢弃，而其他方面要进行很大的改动；一小部分法规要修改，但幸运的是没有必要在美工上进行改变。总的来说，重新调整焦点对我们的工作有一些损失。然而，这种损失是值得的，因为最后 Centipede 3D 成了一个未定的、有焦点风格的游戏，由此它也成了有乐趣的游戏。

就像在本章前面主张的一样，我要说明 Centipede 3D 的焦点并不是独立的焦点，这个很重要。Centipede 3D 的焦点参照了原作 Centipede，因此不完全是自己的观点。当然，Centipede 3D 是一个改写，所以它可以参考原作来开展自己的计划，同样说是做一个续集也是事实。因为不论是改写还是续写，你正进行的工作在焦点上和你参考的原作有着直接的关系，而这个游戏是否成功，主要看它对原作续的好不好。正因为这样，在整个游戏的开发过程中，小组的工作人员们必须时刻提醒自己：自己的工作是否与原作紧紧相连，自己所要完成的游戏是否符合逻辑，是否能成为相称的后继者。既然这个问题倍受关注，那它自然属于焦点。全体小组在进行续写或改写的时候应该让原作贯穿始终，并且要做到很合理的去理解它。然而这里要说明的是，Centipede 3D 的焦点包含了一段对原作 Centipede 的最初呼吁的简明描述，如果焦点必须要参考其他游戏，那最重要的一点就是要确定每一个参与此项设计的人都明白：这个焦点即使是别的游戏的，也一样适合此游戏。



5.3 补充焦点

通过增加补充焦点，有助于使焦点技法再上一个新的台阶。这样你就可以在坚持游戏设计初衷的前提下使你的游戏变得丰富，有血有肉。补充焦点不同于主要的焦点目标，它对主要焦点起补充、辅助和支持的作用，更有助于实现你的焦点目标。它不能单独存在，它主要是对焦点目标的支持，使游戏既不偏离设计初衷，又能实现一些其他的目标。

我将用 Snow Carnage Derby 这个游戏来说明补充焦点的作用。正如你所记得的，在游戏中你得让玩家在一种战斗场景下操纵雪上汽车。这样你就知道 Snow Carnage Derby 游戏的焦点所在了。你想想这个游戏还可以实现什么其他的目标？为了确保这个游戏尽可能好的展现它的游戏效果，开发小组还应该关注些什么？





接下来应该思考的问题就是：你认为什么样的玩家会喜欢玩这个游戏。你希望是斗志坚决的玩家，还是那些并不经常玩计算机游戏的人？这将会直接对游戏的许多方面产生影响，不仅包括游戏控制系统（斗志坚决的玩家能够忍受迟钝的和复杂的操作系统，而那些偶尔玩游戏者则需要易操作的傻瓜型系统），还包括仿真需要达到的程度（那些斗志坚决的玩家往往要求更仿真、更复杂的游戏）。

假如，你的目标玩家是那些偶尔玩一下游戏的人群。你可以通过一些补充焦点使游戏偏向他们的需求：“Snow Carnage Derby 是适合大多数游戏者的易玩游戏。”这样就能有效解释此游戏的大众性和易上手性。也许最大的问题就是操作，你希望游戏者能很容易进入和玩 Snow Carnage Derby 游戏，不需要记一大堆的键才能操纵雪地汽车。那你的焦点就可以是：“该游戏提供最简单的操纵系统，只需记几个简单的键就可以自如地玩游戏了。即使初级玩家也不需要阅读操作手册或使用训练版，游戏级别中包括了学习级。”说明你实际上不想去研究它的操纵系统。这些要保存在设计文档里。现在，你仅仅是在为实现游戏的目标而努力，具体的实现细节还不在考虑中。你也许想要谈谈游戏的难易问题，它分为哪些级别。如果你瞄准的是那些偶尔玩一下游戏的人群，那游戏的难易程度就比那些为游戏高手们设计的游戏要低。也许你想说这个游戏有不同的难度级别：“Snow Carnage Derby 游戏的难度相对来说整体偏低，玩家可以自由选择不同的难度级别，即使是水平不高的玩家也可以选择低难度级别打穿它，放手去玩吧。”

在补充焦点中讨论游戏会使用何种发动机和图形是很有用的。我们可以在开始设计前讨论是用 2D 呢，还是用 3D；但是要确定，这不是达到游戏最终效果的中心问题。不会是我们最初关注的焦点，但是可以作为补充焦点，帮助我们更好地开发游戏。现在，应该想想怎样努力使游戏达到最佳视觉效果。告诉设计小组要达到什么目标，同时也要让开发小组明白为了达到这一效果应该怎样去开发。这可以从对此游戏的一些总的要求开始：“Snow Carnage Derby 追求视觉上的豪华，对比鲜明的背景，雪上汽车和乘坐者要采用明亮的色调，要和地面上的冰雪形成对比。”你可能决定采用 3D 设计，因为这样能够达到更好的动画效果，能帮助我们获得操纵雪上汽车的兴奋体验，而且市场需要的也是 3D 游戏。采用 3D 引擎，游戏者就有了第三只眼，能够更好地让玩家在跟踪那些竞争者的同时控制好自己的雪上汽车和驾驶者。你的焦点陈述可以包括：“游戏使用 3D 设计，允许多辆雪上汽车和它们的驾驶者在屏幕上同时出现。玩家拥有第三只眼，可以在实现对自己的雪上汽车和驾驶者的最佳控制的同时留意其他雪上汽车驾驶者。”关于玩家驾驶雪上汽车的势力范围的说明也是必要的。这样就容易知道哪些地方可以去及如何简单驾驶？哪些地方是玩家可以迎接挑战的区域？说到这里，你也许会记起我们曾提到过的补充焦点。我们这个游戏是为偶尔玩游戏的玩家设计的，应该相对容易玩一些。因此，很难理解的过程和作战区域不考虑：“游戏的设计让玩家能够毫不费力地知道哪些是可以行驶的区域，哪些不是。”

当然，Snow Carnage Derby 游戏还有一些其他的补充焦点，包括游戏的一点一滴，从游戏的机构到游戏包含的故事主线，到平均玩一次要花多长时间。不过要记住，努力避免放进太多细节。这是对设计文档的要求。因为这里仅仅要求的是项目方向，而不是具体的实现。但是作为我们提供的例子，我们现在有足够的补充焦点，你也可以给我们提供类似的焦点和补充焦点：

“Snow Carnage Derby 令人振奋，它是一部雪上汽车快速行进的破坏性游戏。游戏





者的体验就是对实际上不存在的驾驶雪上汽车的一种身临其境的感受，游戏者能在置身雪景的移动和跳跃中感受到乐趣和挑战；游戏的平衡点不在于现实而在于乐趣。游戏产生了一种发自内心的心颤：考虑到死亡或者使敌人的机车驾驶员丧失战斗力，游戏还想用暴力最大限度地制造出戏剧性的效果。Snow Carnage Derby 演绎了一种令人心颤的快速运动，在这个运动中，游戏者要在避免毁灭的同时尽量减少竞争对手。”

观众

对大多数游戏新手来说，Snow Carnage Derby 是一个相对易上手的游戏。游戏尽可能的提供简单的操纵系统，玩家只需记几个简单的操作键就可以自如地玩这个游戏了。初级玩家也不需要阅读操纵手册或使用训练版，游戏级别中包括了学习级。Snow Carnage Derby 游戏的难度相对来说整体偏低，玩家可以自由选择不同的难度级别，即使是水平不高的玩家也可以选择低难度级别打穿它，你可以放手去玩。

视觉效果

Snow Carnage Derby 追求视觉上的豪华，对比鲜明的背景，雪上汽车和乘坐者要采用明亮的色调，要和地面上的冰雪形成对比。游戏使用 3D 设计，允许多辆雪上汽车和它们的驾驶者在屏幕上同时出现。玩家拥有第三只眼，可以在实现对自己的雪上汽车和驾驶者的最佳控制的同时留意其他雪上汽车驾驶者。游戏的设计让玩家能够毫不费力地知道哪些是可以行驶的区域，哪些不是。

要注意那些补充焦点是怎样被偏离设计初衷的标题抵消掉的。这样，焦点的读者就会很容易知道最初的及最重要的焦点是什么，补充焦点是怎样进入游戏具体部分的细节中的。

在实现补充焦点时，不仅要注意保持跟其他补充焦点的一致，还要时刻记住保持与设计初衷的一致。例如，视觉方面的补充焦点旨在实现游戏的易进入和上手，它能使玩者更容易从视觉上理解这个游戏世界。如果在确定这些焦点的时候你自己的想法是互相矛盾的，那就根本不可能写出有意义的游戏总体设计方案。随着游戏开发文档的越来越多，一致性和连贯性的保持也越来越难。但是，既然相应的焦点是简短、明了和容易理解的，那么保持游戏焦点的互相支持应该不成问题。



5.4 使用焦点

不管你是和哪个营销部门合作推销你的游戏，游戏的焦点说明都需要能够很方便地取得。通常，营销部门想要在游戏真正成型之前了解它的实质。此外，许多（不是全部）营销人员并不一定会很喜欢你的游戏，他们希望你能用一些句子简洁地告诉他们为什么这个游戏会这么吸引人。如果获得一个不错的销售数字是议程中的一项（让我们假设这并不是你开发这个游戏的初衷，因为还有更能挣钱的职业），那么让游戏的营销人员在销售它时感到疯狂跟让游戏开发者在开发它时感到痴迷是一样的重要。营销人员会去销售他们信任的产品，会去销售他们认为概念很棒的产品，而你的焦点说明可以更快地让他们明白你的点子有多棒。当然，营销人员也



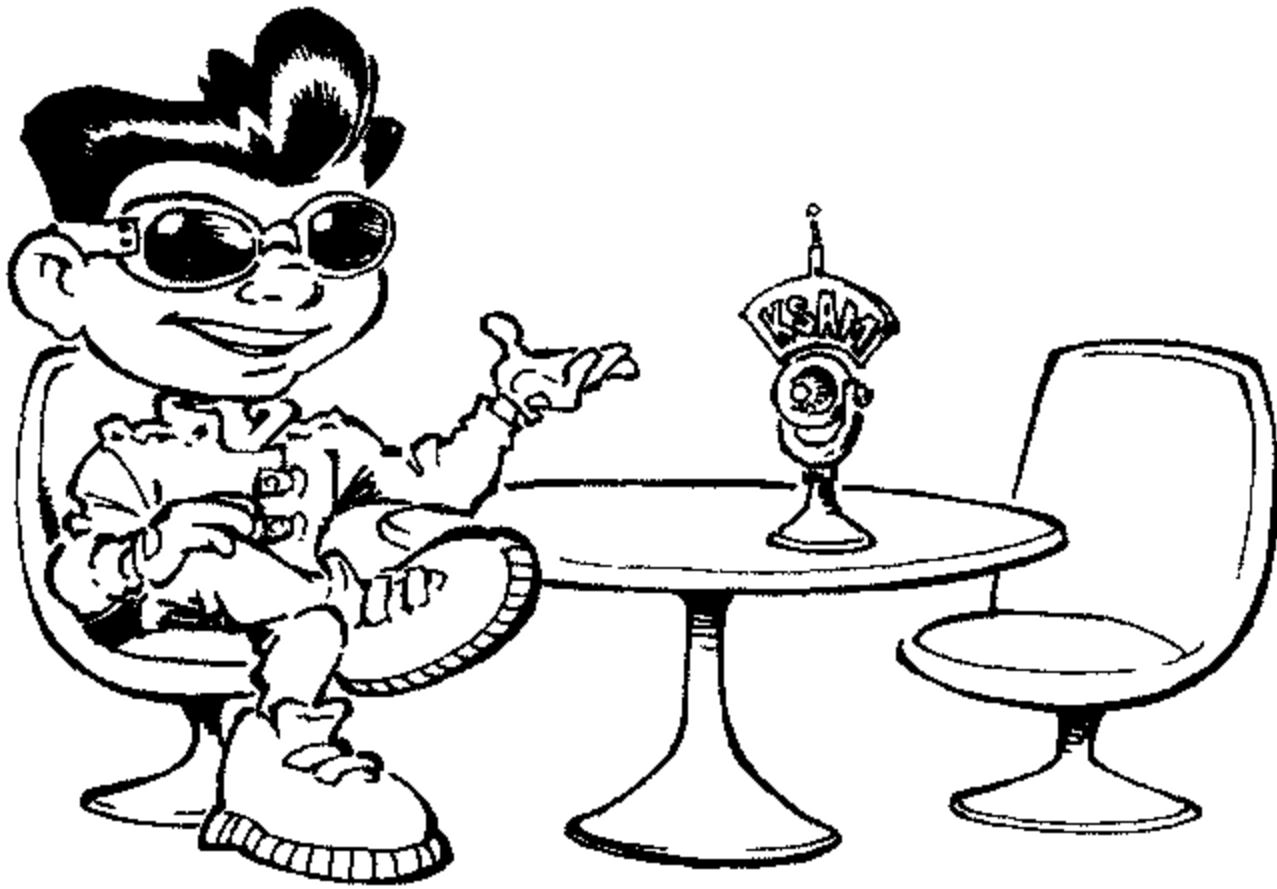
欢迎一些比较性的说明，比如：“Tetris 与 Quake 一样好玩。”所以，你应该尽可能地提供一些与你的游戏同类型的游戏，以及营销人员所熟知产品的比较说明。但是要注意避免相对其他游戏不必要的焦点，以保持该游戏的独特性。如果市场营销人员认为“Tetris 与 Quake 一样好玩”是个很不错的主意，他们会想要了解更多关于你的游戏的信息，而你的焦点恰好满足了他们的需求。

为了保证游戏开发的顺利进行而确定焦点是你首先写下焦点的主要原因。许多游戏设计者在开发游戏时并没有焦点，这些在后来都显现出隐患来了。当然，也有可能什么概念都没有却能开发出很棒的游戏来，赢得大奖也是可能的。然而，当你的生计、名誉和游戏的最终品质处于危险状态时，你更想要的是一台任意的数字发生器来决定游戏是成功了还是失败了。事先确定焦点将帮助你创造一致而连贯、易上手且吸引人的好游戏。



Chapter 6

第6章 Ed Logg 访谈录



Asteroids、Centipede 和 Gauntlet。如果有哪个游戏设计者的游戏可以令人难以忘记，那就是这些游戏了。如果能够投身于鲜为人知的经典游戏 Super Breakout、Millipede、Gauntlet II 和 Xybots 中，你就可以获得无可比拟的事业。在街机游戏风行的时候，Ed Logg 在 Atari 公司设计并开发了这些游戏。而现在，投币式游戏市场已经日渐衰落，游戏产业的注意力都集中到了家庭游戏市场。目前，Logg 仍在继续从事游戏开发工作，将类似于 San Francisco Rush 之类的 Atari 街机游戏改编到控制台上，包括 Nintendo 64 和 Sega Dreamcast。经典的街机游戏看起来很简单，但就是这种简易性，使游戏的设计者可以让游戏变得更为完美。Logg 所设计的投币游戏仍是到目前为止最好的电脑游戏，通过研究这些游戏，设计者们可以从中学习到很多东西。



问：70 年代末、80 年代初时，在 Atari 游戏机上设计游戏的感觉如何？

答：那时候我们既年轻又有精力。那时候的工作氛围就像是现在 Internet 刚刚兴起时一样。在 Coin Operated Games Division 中，我们是相对较小的团队。这样，每个人都可以相互熟悉。人们可以非常随意地交流思想。因为我们所研究的是一个新的领域，在这个领域我们可以做任



何事情，并且都是很“新潮”的。即使是 Rich Moore 设计的 Lunar Lander 在多年前就已经完成，对我们的玩家来说仍是很新潮的。



问：你在游戏中的大多数想法都从何而来？

答：这些想法来自很多方面。例如在 Atari 公司工作的工程师 Owen Rubin，他告诉我 Nolan Bushnell 曾建议他对 Breakout 进行扩展。我采纳了他的意见，并添加了许多自己的想法，然后就制作出了 Super Breakout，也就是我第一个成功的游戏。设计 Asteroids 的想法来自 Lyle Rains，那时候他负责管理工作。他从以前的一个投币游戏中获得了灵感。制作 Xybots 是应 Doug Snyder 的要求，他是 Atari 公司的一位硬件工程师。我们想制作一款类似于 Castle Wolfenstein 的多人游戏，但苦于没有“位图”硬件。于是我就根据 8×8 邮票开发出了一种算法，他负责来制作硬件。Centipede 来自一群人的集思广益。Atari 公司每年都会提出一些新的想法，其中的“虫子射击游戏”就成为 Centipede 游戏的起点。

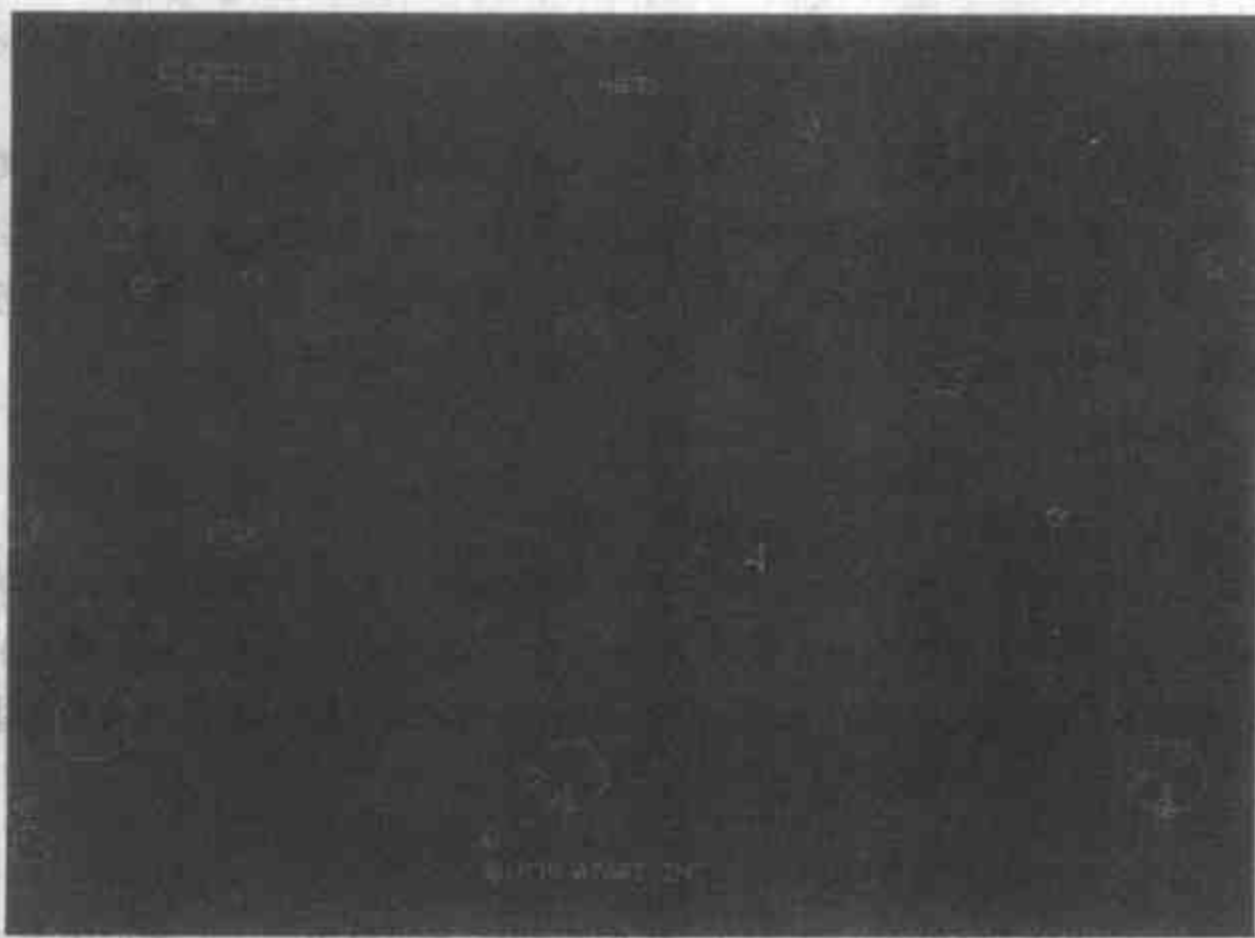


图 6.1 Asteroids

管理人员对这些想法加以评价，玩游戏，然后给出反馈信息。有时一致表决的结果是否定的，那么这个游戏计划就会取消。其中的大多数想法会付诸实施，直到可以进行“实地测试”。这就意味着要由玩家来判断游戏实际的受欢迎程度。然而在这些评价中，有时也会有一些好的建议。其中最重要的一个建议是由 Dan Van Elderen 提出的，他负责管理工作。他问我为什么不能在 Centipede 中向蘑菇射击。是的，游戏中的蘑菇原本是静止不动的。正是他的建议使游戏中的想法得以改进，从而使这个游戏更为有趣。





问：你觉得在 Atari 公司进行游戏开发让人兴奋吗？

答：实际上我在其他的环境下从事游戏开发已有多年的，在上高中的时候，60年代在 Berkeley 的时候，还有我在 Control Data Corp 公司的第一份工作的过程中，我一直在从事游戏开发。我在 Stanford 和 CDC 的计算机上曾经移植过 Star Trek 和原来的 Dungeon 游戏。

为了制作和玩游戏，在加入 Atari 公司的一两年之前，我就组装了一台电脑。我曾去过 Pizza Time Theater 并玩过 Pong 和 Breakout，所以我很了解投币游戏的行情。我曾玩过很多游戏，在 1977 年的圣诞节聚会上，深受 Atari VCS (2600) 的启发。所以在工作中的变化对我来说也显得很自然。那时候，我觉得他们付薪水让我制作并玩游戏真的是太棒了。



问：Dirt Bike 是你在 Atari 制作的第一款游戏，但我了解到这个游戏并没有投入市场。这个游戏属于哪种类型？

答：这个游戏是由 Dennis Koble 发起制作的，并受到了很多玩家的欢迎。除了要驾驶一辆肮脏的自行车，而且要使用一些把手来控制方向，而不是使用方向盘控制以外，这个游戏与 Sprint 非常类似。

我们对这个游戏进行了实地测试，所赢得的利润足以使其维持下去，但并不足以将其批量投放市场。然而，在制作 Dirt Bike 的同时，我还制作了 Super Breakout。在 Atari 公司，以前还从来没有同时制作两款游戏。Super Breakout 游戏获取了很多利润，这也可能是导致不再发行 Dirt Bike 的原因。一想到 Super Breakout 的成功，我就感到非常的振奋。



问：Super Breakout 这个游戏的灵感从何而来？

答：原来的想法包括在 Breakout 上做出六种更新版本。我的设想是发行三种游戏，每个游戏中有两个版本。而实际上却只有一个完整的版本——Progressive Breakout。最后我们在一个游戏中放入了三个版本：Progressive、Double 和 Cavity Breakout。其他没有加入的版本的确不是很有趣。



问：你在制作 Super Breakout 游戏的过程中有创作上的自由吗，或者说因为它是前一个成功作品的升级版而受到了约束？

答：对我来说，Super Breakout 并不是升级版。原来的游戏并不是以软件的形式制作的。我们要从头开始编写代码，虽然使用了相同的控制方式，但游戏的情节也与原来的完全不同。



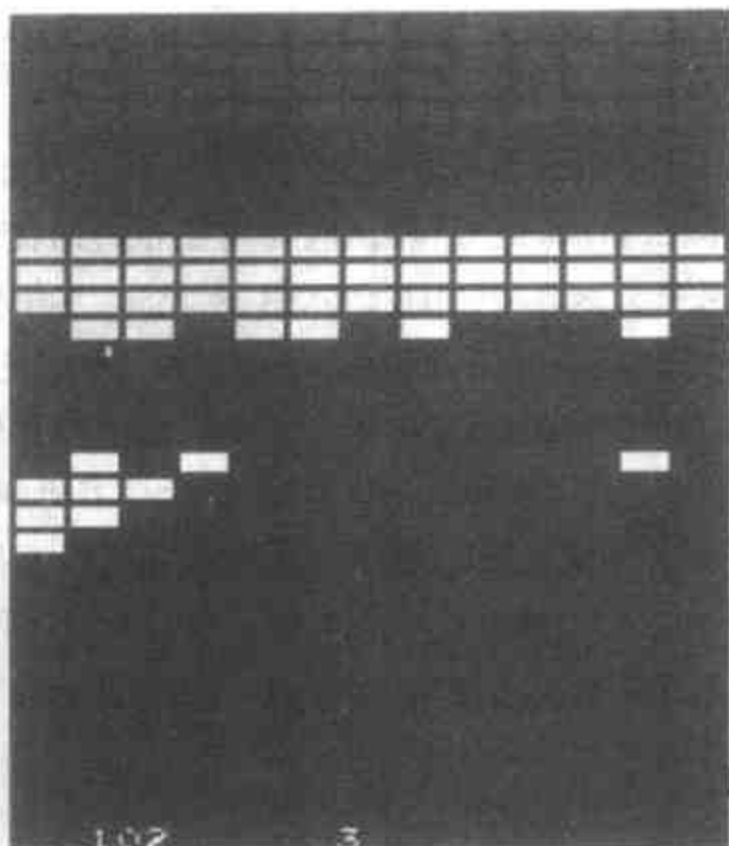


图 6.2 Super Breakout

在制作这个游戏的过程中，我有很大的自由度，因为这个游戏并没有得到正式的批准。这也不是我最后一次做这样的事。在那时，很短的时间内就可以制作出一款游戏，所以在别人知道你做的事情之前，就可以做出有趣的东西来。

可能我应该解释一下那时候我们开发游戏的方式。我们有一台 Digital 主机，其中有基于 6502 芯片游戏的汇编程序。我们让几个助手将纸面上手写的记录输入到程序中，然后给出计算机的打印结果和纸带。是的，你没有听错。然后我们就会通过开发系统将纸带送入 RAM，以替代 PCB 上的游戏 ROM。在调试程序时我们使用的是原始的工具和硬件分析器，然后将修改结果写在打印纸上。因为这个过程在调试阶段和开发下个版本之间留下了一定的时间，我就可以用这段时间来开发另一个游戏。我会转换图形 PROM（是的，我们手工制作图形），然后装载新的纸带。



问：使用这样原始的方法来开发游戏，这听起来真是很令人震惊。那么在各个版本之间这样长的一段时间里，你是怎样做到对游戏进行调整的呢？

答：实际上，我非常善于使用新设备在 RAM 上打补丁，所以很容易发现游戏中细小的变化。我们还有一个 HP 分析器，可以在许多条件下捕获这些变化，这样就可以找到许多 bug，而有些 bug 甚至使用现在的开发工具也无法找到。实际上，在等待下一次修改的过程中可以编写一些新代码，所以可以比想像中节省更多的时间。



问：那么你肯定是认为现代的开发工具使游戏的开发工作变得更容易了？

答：其中存在几个问题。首先，当我们了解到目标硬件的信息时，就可以更容易地知道什么地方出现了错误。现在，目标硬件的信息对我们是屏蔽的，但可以使用多个层面上的软件来进行调试，或者完成那些我们认为比较困难的任务。所以从这种意义上来讲，现在的工作变得更为复杂了。而且目前这些软件和硬件层经常不提供文档、文档中出现的错误及会对我们造成





的障碍。其次，因为要在会引起各种问题的许多字节之间进行互操作，硬件也变得非常复杂。第三，处理器也变得非常复杂，引起了各种调试上的困难，尤其在处理缓存的时候更是如此。第四，目前在开发游戏的过程中会有许多程序员参与进来，所以很容易与合作者引起矛盾。

令人惊讶的是，虽然计算机的计算能力和 RAM 都有了很大的提高，但开发环境并没有在过去的几年中得到改进。举例来说，我在 25MHz、6MB 的 Mac II ci 机器上编译文件的速度，都不亚于使用 550MHz、带有 512MB RAM 并运行 NT 的 PC 机。即使是相同的项目，在我 150MHz 的 Indy 上的生成速度都比使用 550MHz 的 PC 机还要快。我很有自信地认为每个工具开发者都应该使用最慢的系统来开发软件！否则的话，我们就注定无法在升级之后来提高速度。

现代化的工具比以前的使用方法更好，很难想像使用以前的工具也能工作得如此顺利，但是不要忽略在学习新的软件工具、处理器和操作系统中所花费的时间。另外，由于我不了解某些硬件和软件的信息，所以花费了大量的时间在新系统中查找 bug。但除非有极特殊的情况，否则我是不会选择重新使用旧工具的。



问：我从未见过你的下一个游戏 Video Pinball，这个游戏是怎么玩的？

答：通过在镜子下使用一个带有监视器的半涂银面镜和镜子下方的游戏区域，这个游戏可以模拟弹球游戏。监视器可以显示抛掷器和球，这样就可以在游戏区域产生白球的效果。游戏区域实际上是由程序控制的 LED，用于模拟发光物体。另外，控制面板上装有门框，玩家可以“轻推”小门给球一些推动力。这个游戏的概念不是我想出来的，我认为它是 Dave Stubben 的想法。



问：你怎样才能让玩家去玩 Video Pinball，而不是去玩真正的弹球？

答：我并不认为 Video Pinball 会取得成功，并且我也问过同样的问题。对于电脑游戏可以达到的许多方面，弹球游戏都无法达到。而最终这个游戏却取得了很大的成功，赚取了很多利润。我必须承认我的第一印象是错的，这种情况是很少发生的。



问：对于你认为没有什么乐趣的项目，工作的时候会不会觉得很困难？最后制作出来的游戏好玩吗？

答：游戏是比较好玩，但无法与实际的弹球游戏相比较。我很奇怪它为什么会有这么好的销售量。是的，对于自己认为不太好的想法，工作起来确实很困难。但我既年轻又有活力……我还能说些什么呢？



问：Asteroids 游戏的设计思路从何而来？

答：Lyle Rains 曾向我提出一种游戏的概念：让玩家可以射击小行星，因为以前有一个投币游戏，其中有一个玩家不断射击但无法摧毁的小行星，而使玩家无法达到预定的目标。我告诉 Lyle 我们需要一个飞碟使玩家可以击中小行星，而不是让他们浪费时间。我还建议将岩石分为多个小块，使玩家可以有自己的策略，而不只是首先去射击大块的岩石。



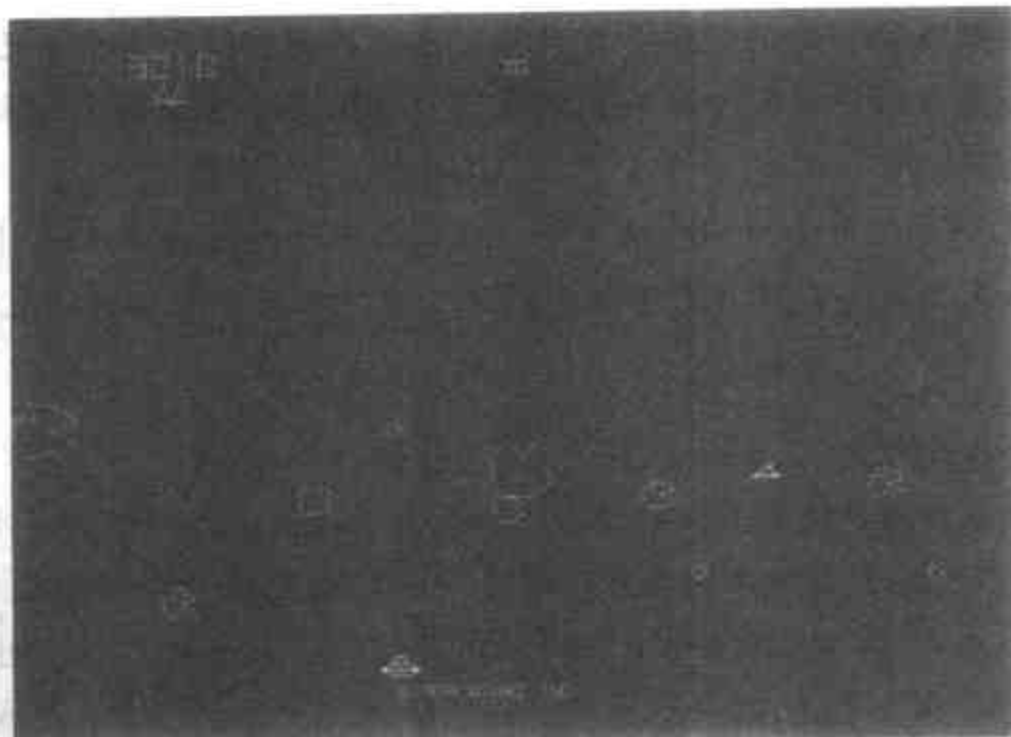


图 6.3 Asteroids

Lyle 向我提出了这些想法。人们经常把成功归功于我们中的某个人。我可能还没有想出自己的一些想法，如果其他人制作了这款游戏的话，那么这个游戏很有可能就完全不同。所以说实话，我们在这个想法上都有功劳。如果设想出这个概念而没有矢量硬件，Asteroids 也不会取得成功。所以它的成功涉及到了许多人和事。我很高兴能够在那时参与到这个游戏的制作中。

在开发的过程中，与原来的想法相比，这个游戏并没有什么改变。我制作了两个飞碟，一个很笨拙，另一个很灵活。在项目结束时我做了一个重要的改动，这个改动具有深远的含义。原来的情况是，只要玩家一进入屏幕，飞碟就可以射击。玩家对这种做法产生了抱怨，我也认为这是不公平的。飞碟在屏幕边缘以外是无法看见的，如果它逼近了玩家的飞船，玩家就无法进行防护。所以在它第一次射击之前我加入了一个延迟。当然，这样就会产生“潜伏”策略。当进行测试的时候，我尝试在某一点上潜伏，并发现这样并不起作用；从这里你可以看出游戏设计者自己玩自己的游戏时有多熟练。



问：对于 Asteroids 的成功，你感到惊讶吗？

答：对于它的成功我并不感到惊讶。在我的印象当中，它听起来就像一个有趣的的游戏。即使在几个星期之后，人们也会来拜访我，问他们什么时候可以玩这个游戏。这就是游戏吸引人的标志！

在我们第一次对这个游戏进行实地测试的时候，我看到一个玩家启动了一个游戏，然后在 20 秒钟之内死掉了三次。他又在继续开始新一轮游戏。这个现象告诉我，玩家觉得自己死掉是由于自己的错误，并且他认为自己可以做得更好。这就是游戏设计者要达到的主要目标，并且我很清楚 Asteroids 做到了这一点。



问：你曾经提过在脑海中玩过这个游戏，你认为这是否是一种预知游戏是否好玩的有效的
方法？

答：对我来说，这种方法非常有效。我还曾想象过一些坏主意：我问自己：“怎样可以出





错？”或者“玩家会被我所提出的想法弄晕吗？”我发现有些设计者对自己的想法非常满意，而不愿意接受可能不太行得通的概念。我无法说出自己脑海中曾经出现过多少好主意，但最后证明是比较差劲的想法。

我曾经遇到过几个像我一样的设计者，我们都曾经放弃过自己设计的游戏。我认为这是个很好的特点。如果游戏不能像想象的那样好玩，为什么还要浪费时间来制作它呢？



问：你现在是否在制作升级版——Asteroids Deluxe？

答：我没有制作升级版，这个升级版是由 Dave Shepperd 来制作的。那时我已经提升为监管的职务。当时我还在负责四人游戏 Football 的项目，所以很忙。如果制作升级版是明智选择的话，我并没有任何问题。我有了一些新的想法，所以我想制作 Millipede。Gauntlet II 是一个逻辑选择，因为与我合作的程序员 Bob Flanagan 和我们都了解代码，并且知道这是我们能够设计出的最好的游戏构想。



问：在 Asteroids 之后，你就没有制作过基于矢量的游戏。你不喜欢和硬件打交道吗？

答：实际上，因为矢量硬件可以呈现出 1024×768 这样尺寸的图片，所以我很喜欢使用。而那时游戏行业已经转向使用彩色监视器。Dave Theurer 是将 Tempest 作为彩色矢量游戏来制作的，但彩色监视器上的彩色光栅不能采用高分辨率。除此以外，还不能在基于矢量的游戏中使用彩色来填充屏幕，所以随着彩色游戏的发展，这种介质的采用在逐渐缩减。



问：Asteroids 不是第一个使用高分榜的 Atari 游戏吗？

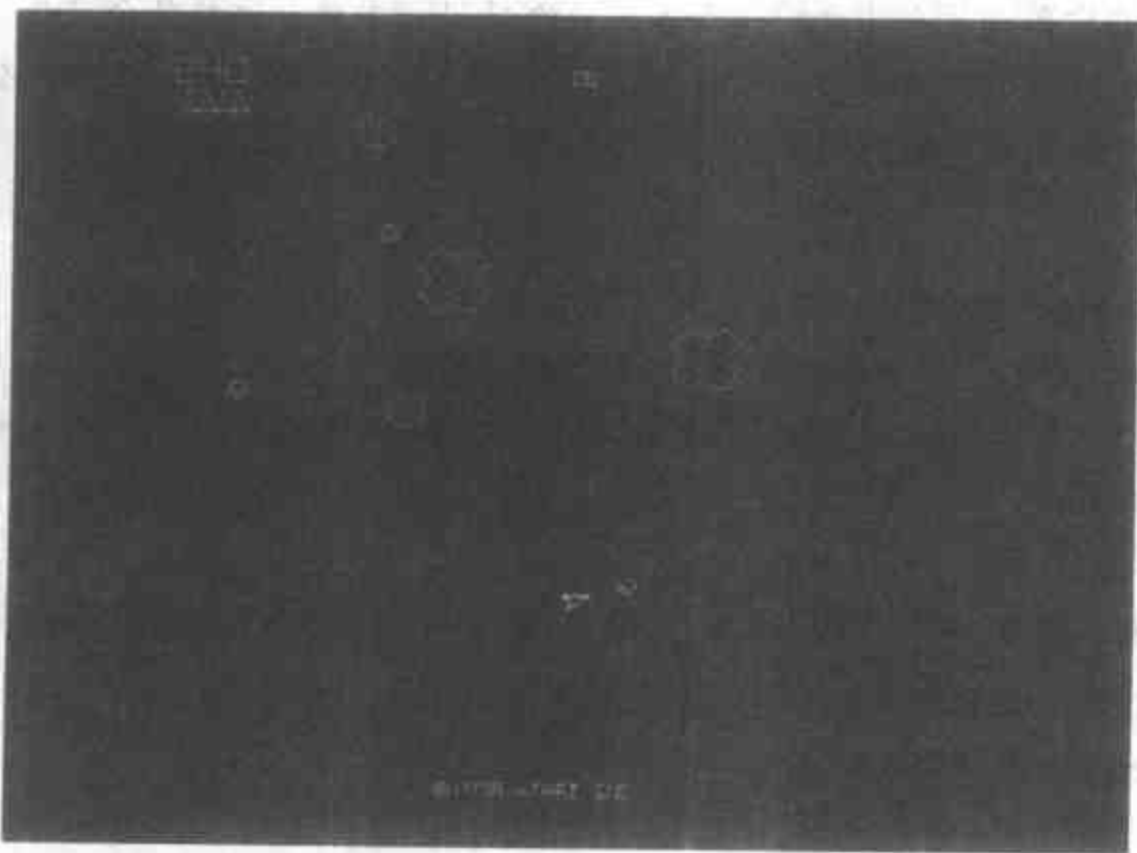


图 6.4 Asteroids

答：实际上，Asteroids 并不是第一个有高分榜的游戏；以前就有其他的游戏曾经使用过高



分榜。我认为保留分数让其他人可以看到是比较好的做法，所以我就把这种做法添加到 Asteroids 中。我认为它取代了 graffiti 的作用。当然，现在它成为了一种标准，通常的做法就是用以电池供电的 RAM 或 EEROM 来永久地保存这些分数。



问：大约在那个时候你在 Atari 2600 上制作了 Othello。我了解到，你在 Stanford 的时候曾经研究过 AI。Othello 这个项目是出自你对 AI 的兴趣吗？

答：不，实际上 Asteroids 更多地受到了我在 Stanford 经历的影响。在 Stanford AI 实验室的时候，我曾经在 PDP 机器上玩过 Space War。在 Student Forum 咖啡店里，我还玩过这个游戏的投币游戏机版本。在我的印象中，这是第一个电玩。Pong 当然是第一个商业化的电玩。无论如何，Asteroids 中太空船的设计都是原来 Space War 飞船的副本。

我曾经把 Othello 当做棋盘类游戏来玩，并对各种可能的策略很感兴趣。所以我在家里制作这个游戏，并产生了一种想法，那就是可以在不使用任何 AI 的模式匹配下来玩游戏。换句话说，电脑不能预知玩家对其动作的任何回应，此时使用的是标准 AI 方式。所以我制作的 Othello 游戏没有 AI。这样做对初级玩家和普通玩家比较适合。而从任何意义上来讲，这个游戏都不是高级游戏。除此以外，2600 型号的机器上只有 128 字节的 RAM，所以就没有足够的空间来预测。

实际上，通过为我提供可以在西洋跳棋盘上画棋子的内核部分，Carol Shaw 完成了游戏中最艰难的部分。在 2600 型号的机器上很难完成什么复杂的任务，它专门用于制作 Pong 风格的游戏。你需要使用所有的活动视频计数器来控制绘制屏幕。这就需要由 Vblank 进行思考或者做其他工作。由于 RAM 的空间有限，所以复杂的信息无法保存到 RAM 中。Othello 游戏共有 2048 字节，其中大多数是内核部分。所以我经常尝试去掉一些字节，以便添加一些新的内容。



问：Centipede 是你的下一个游戏吗？

答：不是，我曾说过，在那时我是监管人员。并且我还是四人游戏 Football 的项目经理，主要负责在原有 Football 游戏的基础上进行升级。

在 Centipede 游戏中，我想出了蜈蚣段的想法，并且设计出了蜈蚣腿移动的方式。我认为在原来的“虫子射击游戏”中并没有提到过这些概念。实际上，没有人曾经站出来“虫子射击游戏”是他们的主意。这可能是因为在完成之后的作品与原来的概念完全不同的缘故。我指派了一个新的程序员——Donna Bailey 来做 Centipede 的编程工作。在 Football 那个项目进展到一半的时候，我离开了这个职位，并升职为监管人员（我不喜欢这个工作，因为我无法

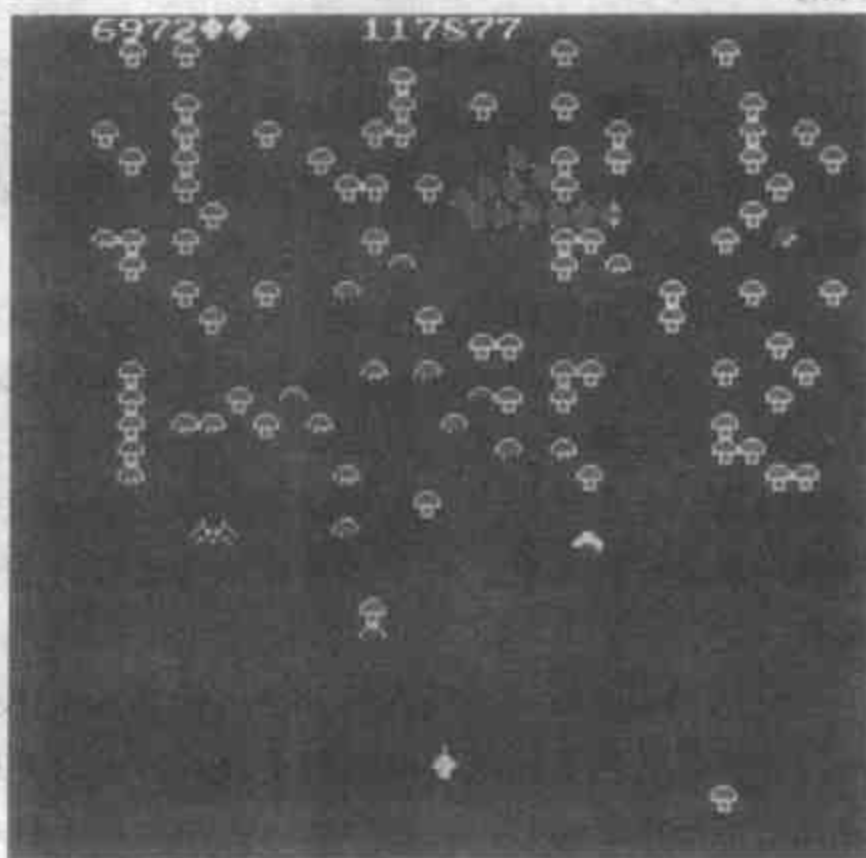


图 6.5 Centipede





再制作游戏了)，开始制作 Centipede。



问：所以 Bailey 在游戏开发过程中的作用相当重要？

答：我想大约有一半的编程工作都是她完成的。因为她还要把一半的精力放在她的第一个项目上，所以游戏设计的工作就落在了我身上。



问：看起来 Centipede 比大多数的街机游戏更加吸引女性玩家，你认为 Bailey 与这有关吗？

答：我真希望自己能知道这个问题的答案。有人曾经说过，在我所设计的游戏中，没有哪个游戏可以像 Centipede 那样受到女性玩家的欢迎。

人们提出了许多观点。其中一种说法是这个游戏是由一位女性制作的，另一种说法是杀虫很适合于女性的思维。我觉得这个游戏之所以很吸引女性玩家，是因为它并不像战斗游戏、RPG 游戏和运动游戏那样有性别上的偏见。其他一些这样的例子，如 Pac-Man 和 Tetris 也是很出名的。

我知道 Centipede 有吸引众多玩家的基本标准。它有全新的、吸引人的外观（让玩家愿意去尝试玩这个游戏）、有明确的目标（射击物体）、明确的规则、简易的控制、一种完成任务的意识（在蜈蚣打败你之前就要杀掉它）、变化多端的策略（捕获蜈蚣并杀掉蜘蛛，或者采用渐进战术，或者引诱蜈蚣，或者就是直截了当地玩）、有足够的随机性，可以在每次玩游戏的时候都有不同的感觉、使玩家可以继续玩下去的目标（每 12 000 点就加一条命）、积累经验就可以玩得更好的明确含义，还有所有的失败都是玩家的错误这种感觉。



问：所以你说 Centipede 是很多人想法的结合。在 Atari 公司中搜集想法的过程是如何进行的？

答：集思广益的想法来自公司中的每个人。在每次正式会议之前的前几周，他们都会在公司外聚会，通常这些想法就是主题。大多数的提议都是某种简单的构想或者表现形式，以给读者更多的信息。偶尔也会有人提出完整的游戏说明，其中包括硬件、控制方式、表现形式和游戏设置。

在举行集体研讨会议期间，当每种想法提出之后，大家就会提出一些建议，并对其改进。另外，市场部门会提供销售计划和行业的现状。我们会分为各个小组来讨论特定类型的游戏，或者研究游戏本身。最后，我们会再次聚集起来，提出自己的想法，并投票选出比较受欢迎的想法。我承认大多数的想法都是程序员和设计者想出来的，但公司里的美工和其他人也提供不少意见。我发现，对于这些想法要进行许多的加工，这样才能使原来大家提出的主意改头换面。

Atari Game 公司（现在的 Midway Games West 公司）每年还都举办这样的活动。但诚实地说，许多最新推出的投币游戏都是旧版游戏的再版。例如，Rush 或 Cruisin' 的新版本都是如此。这都是市场所驱动的：这些游戏过去在市场上销路很好，公司不愿意冒险尝试新项目。



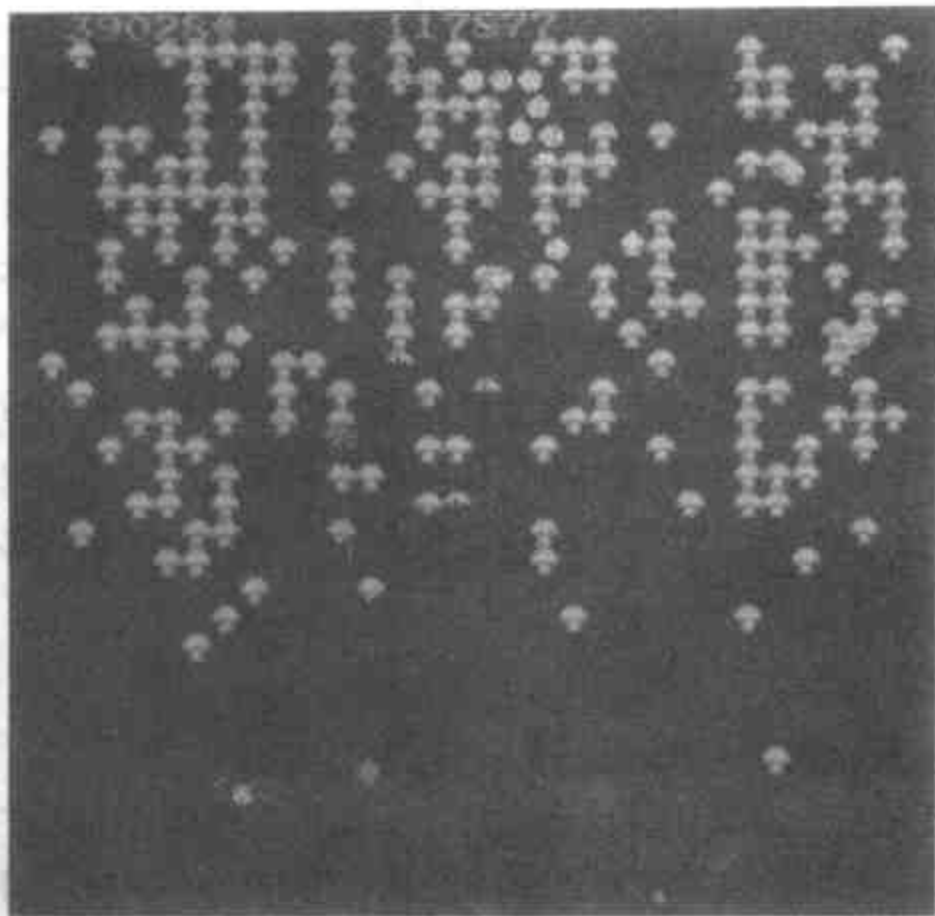


图 6.6 Centipede

问：那么 Centipede 是如何改变游戏开发路线的呢？

答：我说过，Dan Van Elderen 曾经询问，玩家为什么不能向蘑菇射击。我及早地意识到自己需要采取一些措施来生出新的蘑菇。这样，玩家在射中了蜈蚣段之后就会留下一个蘑菇。我还制作出了一些昆虫，在屏幕上落下时，它们可以留下一些轨迹，给这种模式添加了一些随机性。换句话说，我不想让玩家只创造出单一的蘑菇模式。蜘蛛也可以成为我的“Asteroids 飞碟”，它可以让玩家一直在移动；蜘蛛还必须吃掉蘑菇，使蘑菇不会侵入玩家所在的区域。在蜈蚣模式中还加入了蝎子，以添加一些随机性，并且在蜈蚣段落到屏幕底端的时候，还会产生一种惊慌的感觉。

问：在你制作游戏的过程中，是否在尝试让不同的玩家使用不同的战术赢得胜利呢？

答：我正在努力使玩家可以设计出尽可能多的战术。所以从某种意义上来讲是这样的，我鼓励玩家试验并尝试各种不同的战术。我也在尝试不让这些战术永远都奏效，也不会让游戏变得过于简单。但我想给玩家留下这样的印象：如果他的表现好一点，他就可以取得胜利。

问：为什么在 Centipede 游戏中选择使用轨迹球？

答：我从一开始就使用轨迹球。我曾经在 Football 中使用过轨迹球，我需要让这些不那么笨重，并且可以自由地来回移动。这就是在 Centipede 中使用轨迹球的原因。轨迹球就像计算机上的鼠标一样，使用它可以控制任意方向和速度。没有哪种控制器可以做得这么好。为了提高玩家的可控制性，轨迹球是首选控制器。





问：在我看来，Centipede 是迄今为止最为谐调的游戏。要达到这样一种谐调，是否进行了很多次的试验呢？

答：因为并没有尝试或者放弃什么，所以这里我不会使用试验这个术语。游戏中有一只蝗虫，我们想让它跳向玩家，但是蜘蛛可以强制玩家移开，所以蝗虫从未得逞过。当然，如果看一下自测试图像的话，你就可以看到蝗虫的图像。

研制过程中当然也有很多调整。游戏的进展时间和速度会随着项目的进展而改进。这种谐调来自游戏自身的规律，以及所要把握的尺度：知道什么时候让游戏继续，什么时候应该修改某些东西。有些人可以领悟到其中的巧妙之处，而其他的人却不能。我只能用“游戏意识”这个词来定义。



问：在设计 Millipede 的时候，你在设计上有自由度吗？

答：根据我过去的记录，在设计游戏的过程中，我比其他任何人都拥有更多的自由度。大多数人所不能理解的东西，也就是我曾设计过但没有投入发行的那 50% 的游戏。没有人曾听说过我的失败。有些游戏的设计计划是我自己取消的。而在 Atari 公司，我还没有听说过任何人这样做过。当然，有些我尝试取消的游戏，最终却没有取消。最近你就可以在市场上看到这些游戏，这些游戏可以收回开发成本。但是在投币游戏市场中，想不赚取大的利润都是很难的。

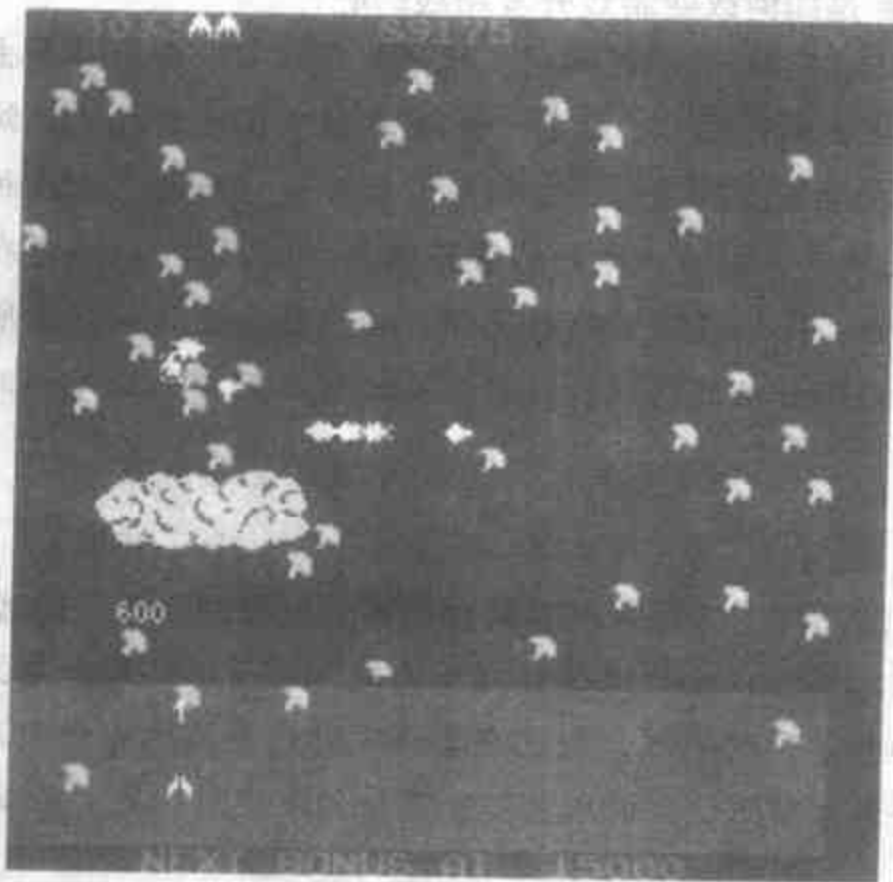


图 6.7 Millipede



问：在 Millipede 游戏中，玩家可以更快地进入到游戏中，例如得到 45 000 点的分数。这样是否会缩短专业级玩家取得胜利的时间？

答：这是增加分数的一种方式。游戏可以让比较熟练的玩家以比较高的分数开始游戏，而游戏的难度可能超过其能实现的水平。这样就可以缩短游戏的时间，并且可以达到更高的分数。



从某种意义上讲，我这样做是考虑到市场的原因。这种做法并不是首先在 Millipede 上使用的。Tempest 早在 1981 年就采用过这种做法。



问：我非常喜欢 Millipede 中不断“生长”的蘑菇，这是使用“生命”算法做到的吗？

答：是的，它是基于生命的游戏，两三个相邻的蘑菇可以生成一个新蘑菇，而超过或少于这个数量，蘑菇就会死掉。这也是一段很有趣的历史。Mark Cerny 曾经问过我为什么不能在蘑菇中使用生命算法。我告诉他我很忙，但如果他能在游戏中添加算法的话，可以去做。当然，Mark 是个很聪明的人，他看过了我的代码之后，很快就添加了这个算法。他还添加了 attract 模式，用于演示所有的生物体。



问：从 Asteroids 到 Millipede，几乎你制作的所有游戏都移植到了许多系统中：比如 2600，Apple II 等。你是怎样看待这些改版的呢？

答：这对公司的发展有好处，所以具有商业价值。当然，看到自己制作的游戏在很多机器上出现，我自己也感到很荣耀。但对于有些移植版本，我也有一定的担心。我知道某些系统存在限制，所以我想让公司发行最佳的版本。在许多情况下我都想让游戏具有所有的特性，但遗憾的是，很少能做到这一点。

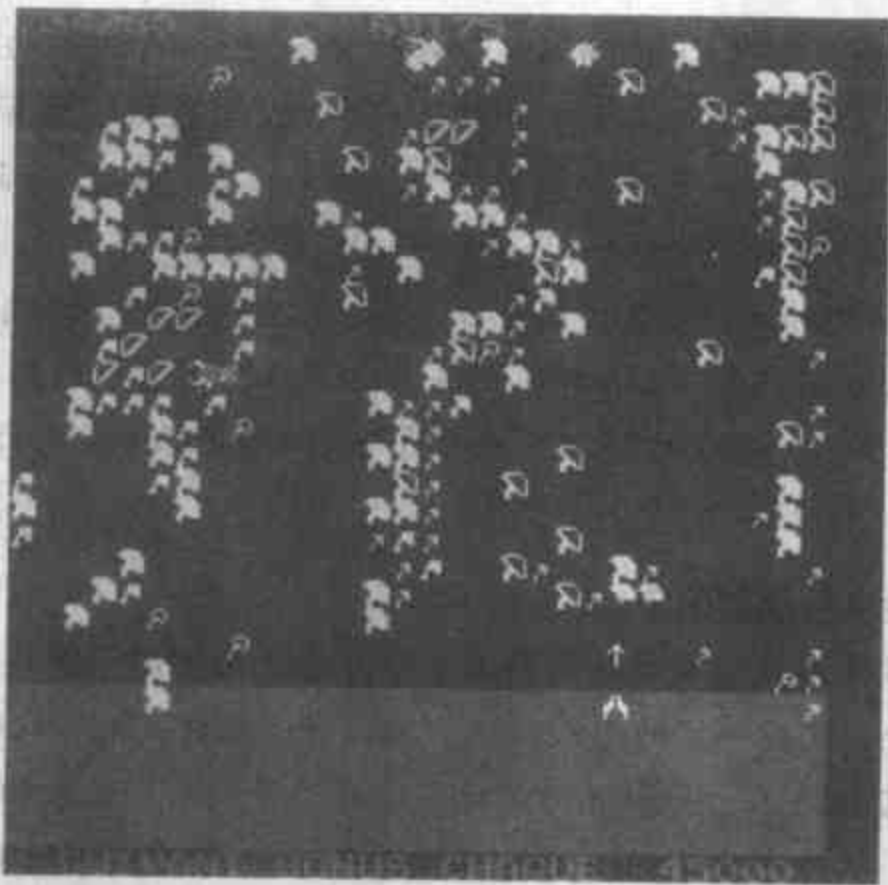


图 6.8 Millipede

有些版本对游戏进行了改进，而这些改进在投币游戏市场上是不可能实现的。例如，在 Gauntlet 这个游戏中，他们加入了一个 quest 模式，生命值是有限的。这在投币游戏机上是无法实现的，因为在正常的基础上，对象可以得到更多的钱。至于另一个例子，可以看看 Atari 2600 盒式游戏机上 Pong 改进版本的数量。为消费者考虑而对游戏加以改进，的确是明智之举。





① 问：在制作了 Millipede 之后，Maze Invaders 是你要制作的下一个游戏吗？我知道这个游戏一直都没有发行。

答：它是一个很好玩的谜题类游戏。没有成功发行这个游戏，我并不感到难过；在实地测试的时候，这个游戏并没有获得足够的利润。而我的儿子很喜欢这个游戏，在我的车库里还放着其中一个原型。另一个是 Texas 的一个操作人员购买的。他很喜欢这个游戏，所以就说服 Atari 卖给了他。

我记得自己曾说过，在我制作的游戏里，几乎有一半没有投入发行。有一些工程师的发行率比较高，尤其是 Dave Theurer。但也有一些人制作的游戏从未发行过。

② 问：Maze Invaders 这个名字暗示可能有什么东西受到了 Pac-Man 的启发，是这样吗？

答：是，在某种程度上是这样。它是一个类似于迷宫式的游戏，但这个迷宫是动态变化着的。其主要特性非常像 Pac-Man；他也很聪明。其中有些部分我觉得很迷惑，比如说迷宫什么时候会突然拦住我。我无法解释这个奇怪的问题，这也可能是这个游戏失败的原因。

③ 问：我了解到，在 1983 年你还曾经制作过 Road Runner 光盘游戏。这个游戏是基于 Warner Bros. 卡通人物的吗？

答：是的，这个游戏是根据 Chuck Jones 创作的 Road Runner 制作的。玩过 Road Runner 部分的玩家可以令 Wile E. Coyote 落入陷阱。我让时代华纳公司发给我所有的 Road Runner 的卡通片。我仔细看过了每一集，然后选出了最好的片断加到光盘中。因此，当玩家成功地破坏掉 Wile E. 之后，游戏就会切换到卡通片中类似的场景中——Wile E. 被干掉了。

我一直很喜欢 Road Runner，而且我认为自己可以把它变成一个游戏。当我开始制作这个游戏的时候，在我的想象之中就有一些很特别的东西。游戏肯定可以达到要求，但却没有我所设想的那样好玩。我很喜欢所有的旧卡通片，而且也想与 Chuck Jones 见面，但是……

④ 问：所以这个游戏项目被取消了？

答：光盘游戏在投币游戏市场中的失败是由于可靠性的问题。游戏实际上可以获得所承诺过的利润值，但不能是以光盘游戏的形式出现。所以当让他们让我把游戏移植到其“System I”硬件平台上时，我婉言拒绝了，并和他们说我已经有了另一个要追求的目标。我很高兴他们让我继续充实这个想法，因为它就是 Gauntlet。Road Runner 最后移植到了 System I 上，并最终得以发布。

⑤ 问：Gauntlet 是否非常接近于你最初的设想，或者在开发的过程中做过许多修改？

答：最近我回顾了最初的游戏设计文档，并惊奇地发现，最终产品中的图像和游戏设置与文档中的内容是如此吻合。当然，在开发过程中发生变化的是硬件方面。我设计出了一种算法，它可以处理 1000 个对象，而不会给处理器增加负担，或者减慢帧频。我曾经问过电气工程师





Pat McCarthy, 是否可以扩充现有的硬件, 然后他就找出了一种方式, 使我可以显示所需的所有对象。最终, 在 Gauntlet 的开发过程中, 总共提出了 5 项专利技术。

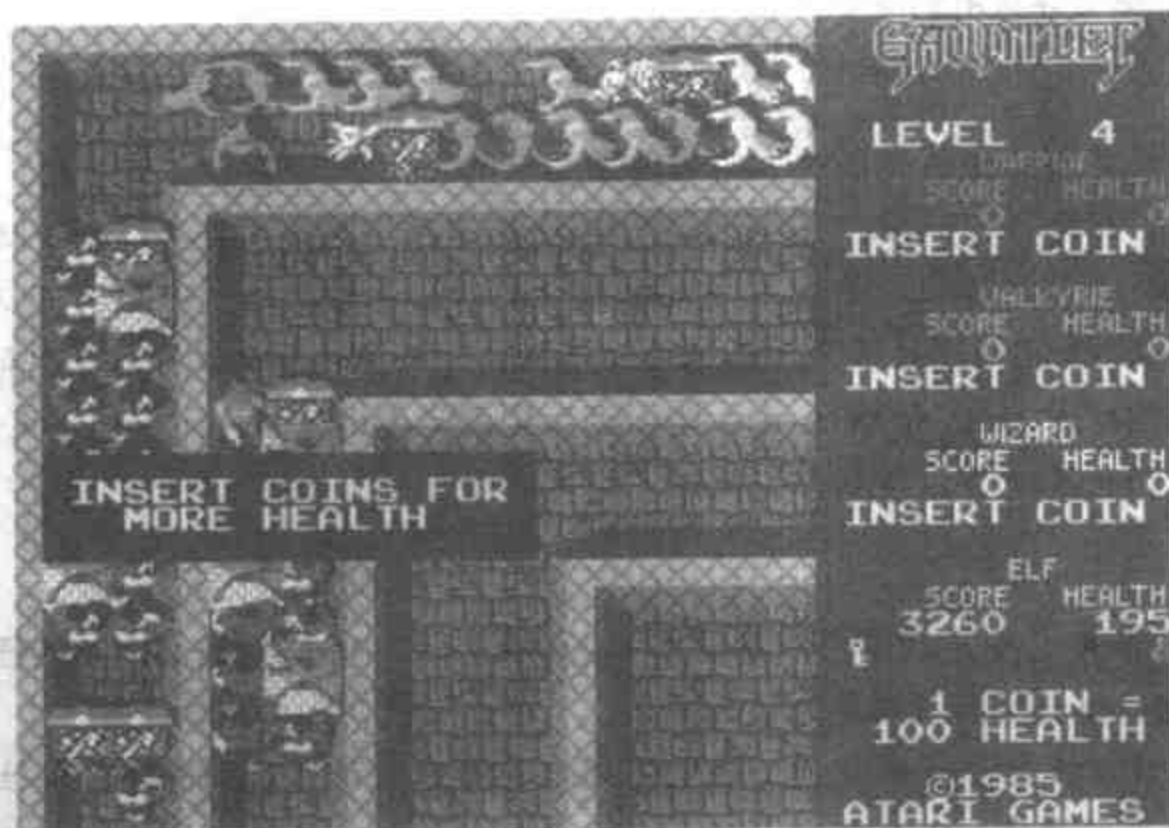


图 6.9 Gauntlet

因为印刷电路板 (PCB) 长度和日文工具包 PCB 长度的限制, 我们决定在 Gauntlet 中使用四层 PCB。Atari 从未设计过这样的布局, 也没有用过我们所需要的这样小的轨迹。但最终我们在 Atari 准备好了所有以后要使用的 PCB。所以, Gauntlet 不仅在这个行业中取得了成功, 而且在 Atari 公司中的管理和制作游戏方面, 也使我们迈出了坚实的一步。

问: 在我的印象中, 在 1985 年, Gauntlet 好像是第一个可以四人同时玩的动作游戏。

答: 这是第一个四人游戏, 玩家可以在任何时候终止或离开, 并且玩家的动作可以控制屏幕的滚动。而这并不是第一个多人游戏。在 Tank 8 这个游戏中, 可以有 8 个玩家同时在屏幕上作战。但所有的玩家都必须同时进入游戏。Gauntlet 从一开始就采用了四人的做法。我怀疑这是因为我只能将四个玩家放在屏幕中的原因。

我认为 Gauntlet 是第一个让玩家在任何想玩的时候都可以参加进来的游戏。我不想让玩家等待, 就像 Tank 8 一样, 每个人都要同时投币。惟一的解决办法就是让玩家来去自由。生命值总是在游戏的一开始就设置好了。这个想法来自于 Dungeons & Dragons, 那时非常流行。所以从逻辑上来讲, 用钱就可以买到更多的生命值。因为每个投币游戏的设计者都希望玩家能够尽可能地投入更多的钱, 所以我想不出什么理由不让玩家投入硬币之后增加他们的生命值。事后发现, 这是一个很妙的主意, 因为从心理上来讲, 丢掉 2000 点的生命值比再投入一枚硬币要痛苦得多。除此之外, 玩家在他们控制的任務死掉之前, 也不需要再从口袋里掏出硬币投入游戏机。

问: 让游戏说“红色勇士需要食物, 糟糕”这样的主意是从何而来的?

答: 我不记得了。我怀疑这个主意不是我想出来的。它可能是我的合作编程人员 Bob Flanagan





或者 Atari 公司的其他人想出来的。在很多情况下，我们都会有大量的词句让“Dungeon Master”说出来嘲弄玩家。有几个词句可能已经深深映入每个人的脑海中。我最喜欢的一句是：“看起来 Wizard（我）要吃掉所有的食物”。



问：许多人认为 Valkyrie 是四个人中力量最强大的。

答：实际上，玩家可以一直使用 Hulk 或 Wizard 来玩。这是由日本的一个玩家在玩单人游戏的时候发现的。如果玩家没有在最后一关丢掉过多的生命值，这个问题就可以通过减少下一关的食物量来解决。Valkyrie 被设计为最稳重的角色，但射击能量值、射击速度和射程比其他的属性更为重要。这就是 Hulk 和 Wizard 看起来力量最强大的原因。对许多玩家来说，Elf 是比较有趣的，因为使用他总是可以得到比其他玩家更多的食物或宝物。



问：在 Gauntlet II 中，玩家可以都使用 Valkyries、Elves 或者他们想要的任意组合。这表示人物的等级比在第一版游戏中更为平等吗？

答：不，对于让人物平等的问题，我想不起来曾经做过什么。加入这种特性是因为有些玩家想要玩一个特别的角色，而我不想让他们等待自己想要的人物无人扮演时才能去玩。因此，实际上我又排除了另一个障碍，使玩家可以很快进入到游戏当中。

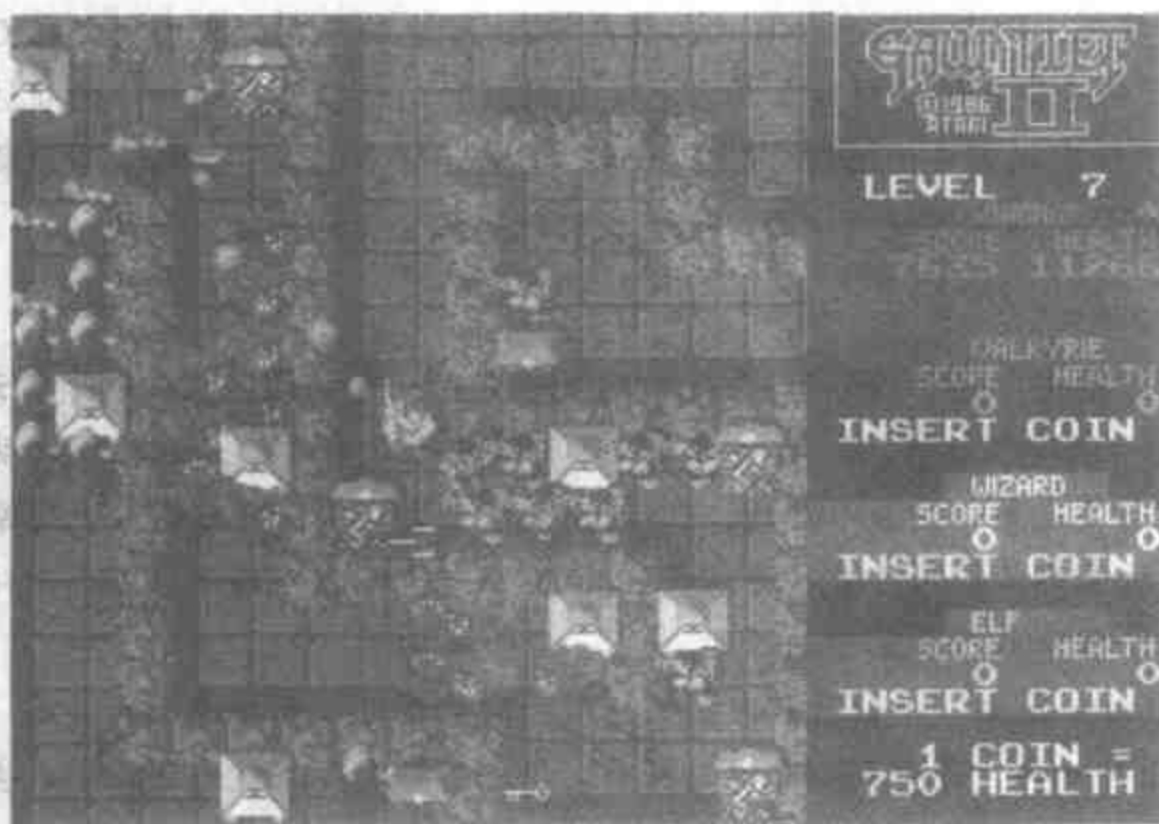


图 6.10 Gauntlet II



问：Xybots 是继 Gauntlet II 之后你的下一个项目吗？

答：实际上，当我发现原来设计的游戏没有预想的那样好玩、并且很快取消掉之后，Bob Flanagan 和我就开始制作另一个游戏。

就如我以前所提到过的那样，Xybots 是为了制作 Castle Wolfenstein 而制作的。我想将这个





游戏做为双人分屏 Gauntlet III 来制作。而在向市场推广的过程中，他们却说市场上需要与 Gauntlet 有所不同的游戏。所以我就对人物和敌人进行了修改，使他们更像 Major Havoc。我很遗憾修改了主题，并且试图保留原来的游戏理念。

① 问：制作游戏的 3D 外观是否是在技术上的挑战？

答：我开发出一种非常有趣的算法，可以使用 8×8 像素大小的图形进行 3D 旋转。我不知道如何解释，在不让原来的框架可视化显示的情况下，怎样才能让这一技术发挥作用。我可以让玩家旋转，但不是以 90 度角为增量；但如果只允许 90 度旋转的话，就可以更容易地设置游戏。

① 问：我记得这个游戏的控制非常有趣和特别。

答：游戏的控制器非常特别，它提供了 8 向操纵杆，顶端还有一个旋钮，可以转左或转右来指示方向。这种控制增加了游戏的难度，在投币游戏市场中简直是自我毁灭的做法。对于 3D 游戏，尽管玩家有地图可以用，但却不能很容易地想象到自己的位置。另外，玩家也有可能背后受到袭击，这也增加了受挫的因素。

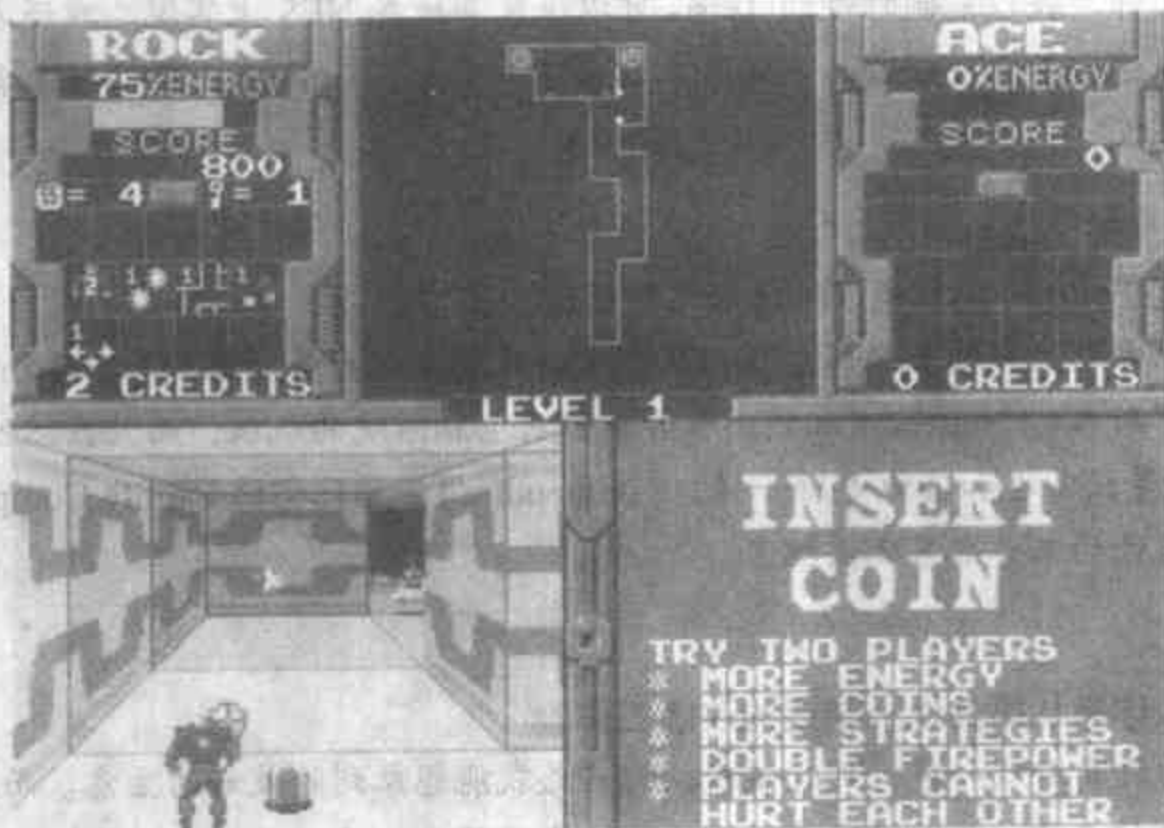


图 6.11 Xybots

① 问：你是怎样参与到 Atari 公司 Tetris 游戏的制作中的？

答：我玩过 Tetris 的一个版本，而且很快就被吸引住了。我让我们的法律顾问 Dennis Wood 去拿到版权。因为我曾经在 Nintendo Family Computer 参与反向工程工作，这家公司迅速成为美国的 Nintendo Entertainment System 公司，所以我决定在 FC 和 NES 上制作这个游戏，然后通过 Tengen——也就是 Atari 的消费发布商来销售。Dennis Wood 得到了这个游戏的版权，并且我们首次在 June Consumer Electronics Show 上展示了 Tetris。为了对游戏进行改进，我重新制作了界





面，并在随后的一月份举办的 CES 上发行了这个游戏。

应该指出，在制作这个游戏的时候，我还在参与另一个游戏的制作工作，所以无法在 Tetris 这个项目上投入所有的时间。也正是因为这个原因，我将 Tetris 这个项目转交给 Greg Rivera 和 Norm Avellar，并在投币式游戏中开辟市场。在重新参与到我的项目中之前，我已经将原来的代码在投币式游戏机上运行过了。这就是我的名字出现在投币版本制作人员名单上的原因。



问：你喜欢 Tetris 游戏的哪些方面？

答：我只知道这个游戏实在是太吸引人了，所以我们需要玩。回想起来，我可以解释为什么这个游戏如此出色，但我并不能确定它可以证明什么事情。除此以外，真正的问题就是：“为什么我没有想出这个主意？”



问：让 Tengen 销售的 Tetris 是你惟一的 NES 项目吗？

答：在与 Atari 公司的诉讼案之前，我一直将 Centipede 和 Millipede 在 FC 上运行，但最终的裁决结果是，在 Atari 销售前，所有游戏的版权都归属于时代华纳的 Tramiel。所以我不得不放弃我所做的工作。我以前制作过的 Tetris 可以很容易地在 NES 上实现。我还添加了两个玩家的模拟特性，使这个游戏比所有其他的版本都出色。后来，Tengen 出售的版本卖到了 150 美元以上。



问：为什么 Tengen 出售的 Tetris 版本最终停止了发行呢？

答：你可以看到故事的多个版本，但我怀疑匈牙利人拥有基础版权。当 Nintendo 提出新版权的时候，俄罗斯人从 Nintendo 收钱。尽管我们有电脑系统的版权，而且虽然他们在日本出售的 Basic 和键盘等设备也和其他电脑一样，但 Nintendo 仍声称他们的 Family Computer 不是电脑。我很遗憾看到自己的工作成果都丢掉了。



问：为什么你想从事其他人游戏的改版工作？

答：与我的许多游戏一样，那时候我认为这是我能想象到的最好的主意。而在这种情况下，因为我很喜欢这种工作，所以很容易做出决定。有哪种更好的方式可以玩自己如此喜欢的游戏，而且还可以保证我能根据自己的喜好来制作游戏呢？



问：下一步你计划制作什么游戏呢？

答：在“Tetris 事件”中，我最终取消了我正在制作的游戏计划。我认为 Steel Talons 是我的下一个项目。我想制作 3D 的 Red Baron 飞行/射击类游戏，但战飞机的游戏在青少年中并不太受欢迎，他们是投币游戏的主要玩家。市场上需要喷气式战斗机，但我认为这个主意不怎么样，因为谁会喜欢看到远处的黑点相互射击？我想要的是让玩家可以从近距离地看到和自己作战的敌人。直升机是个比较明智的选择。



① 问: Steel Talons 是真正的直升机模拟游戏吗?

答: Steel Talons 有直升机上的所有控制: 方向舵、控制高度的集中控制系统, 还有转向用的手杆。当然, 如果没有任何协助, 控制直升机的飞行是很困难的, 所以我有电脑助手, 就像真正的军用直升机那样。我添加了自动集中控制系统, 这样的话, 玩家就可以维持水平飞行, 并且可以很顺利地降落。如果飞机前面有斜坡, 飞机还可以增加高度。“真实”模式取消了这些帮助, 并且增加了玩家的加速度作为补偿。这是一种比较特别的特性, Atari 为此申请了专利。

这个游戏还有另一个非常有趣的特性, 而这从未在以前的游戏中使用过。我们在座位下安装了一个弹球击键声检测器, 用于表示自由的游戏。在首次实地测试中, 这个击键声检测器的电压比实际需要的要高, 当它发射的时候, 第一次使用它的玩家几乎会从座位上跳起来。在整个街上都可以听到这个声音。

在第一次实地测试中, 我们还发现了一个以前从未发现的新问题。我走出去检查收集箱, 并且尝试拿掉硬币箱。如果你曾见过 Steel Talons, 会看到硬币箱放在一个角落, 需要操作者完全伸展开手臂才能抬起来。这并不是抬起重物最容易的位置。再转回话题, 我尝试把盒子抬起来, 但却无法移动它。我以为盒子被卡住了。但我很快就发现盒子里满满的, 重得几乎无法移动。所以在手册里就有了这样一条奇怪的指示, 那就是让操作者每隔几天就清空硬币箱。

② 问: 在 Steel Talons 中, 你没有与 Battlezone 的创作者 Ed Rotberg 一起工作吗?

答: 是的, 我与他一起工作过。在 Atari, 他度过了 Battlezone、Asteroids 等游戏创作的黄金时期。他离开了 Atari, 开创了一家名为 Sente 的公司, 过了几年又返回了 Atari。他在制作 Tube Chase 类的游戏时所使用的硬件就是与 Steel Talons 中相同的 3D 硬件。这种硬件是 Hard Drivin, PCB 简化成本的版本。所以 Ed 很自然地在这个项目上和我一起工作。这个游戏中另一个有趣的特性是迷雾。当我告诉原来的 Hard Drivin' 开发组我可以在喜爱的环境中添加迷雾时, 他们都不相信我。我对实现了这种效果感到很得意, 当看到这种效果的确可以出现时, 他们都感到很惊奇。

③ 问: Space Lords 项目是如何开始的呢?

答: 我想将 Gauntlet 中所采用的多人游戏的做法延续下来, 并在 Xybots 和 Steel Talons 中继续采用。所以我选择了 3D 空间环境, 并将四个小格相互连接在一起。每个小格都有两个类似于 Cyberball 的监视器。我尝试使用 Atari 的“生长动作对象”硬件来降低成本, 这比 Steel Talons 中使用的 3D 硬件要便宜得多。它不能画出 3D 多边形, 但可以展开或收缩平面材质。

④ 问: 我了解到 Space Lords 在销售状况上不太好。

答: Space Lords 采用了一些奇怪的盈利模式。在某些街区, 两个双机箱游戏机每周都可以赚到 1000 美元。而在一些较小的街区, 单个机箱游戏机就只能赚到 75 美元。根本原因是由于它的造价, 而且这种游戏机只能在少数几个地方销售, 所以我们在销售的时候感到很艰难。用每个玩家一个监视器的概念来制作投币式游戏是非常困难的。即使我以一半的价格作为枪手加入第二个玩家, 许多人仍认为这样没有当飞行员好玩。





问：就在飞行类游戏开始衰落的时候，Space Lords 出现了。

答：因为飞行类游戏经常是“工具箱”，比大型专用的直立式游戏机要便宜得多，所以使 Space Lords 在销售的时候遇到很多困难。Street Fighter II 赚取了很多利润，而且在很长一段时间内的盈利情况都很好。



问：实际上，自从 90 年代初开始，街机游戏就一直局限于少数几种风格中。对于目前推出的这些游戏，你持什么看法？

答：你说得对，投币游戏市场好像都是驾驶类、飞行类和射击类游戏，偶尔还会有运动类的，如高尔夫球。其中存在几种原因。驾驶类游戏是大家都喜欢的游戏，并且通常在很长一段时间内都可以盈利，所以它是最受人欢迎的游戏主体。除此以外，大多数的家庭中没有方向盘和油门踏板，无法使玩家有那种置身于汽车之中的感觉，所以在家里无法得到这种体验。战斗类游戏现在很难在街机游戏中打开销路，这是因为在大多数高级的控制台上玩家也可以得到相同的体验。而在以前，这些游戏玩起来很便宜，而且也赚了不少钱。射击类游戏的市场一直比较稳定，因为玩具枪并不是控制台和 PC 的标准控制器，所以玩家可以得到这种体验的惟一方式就是去玩街机游戏。

总结起来，目前大多数街机游戏都很普通，也仅限于有限的几类。我不认为街机游戏彻底萎缩，但已经不想往日那样受人欢迎了。



问：Space Lords 是你制作的最后一个投币游戏吗？

答：在离开 Atari 以前，我一直在制作射击类游戏。那个游戏虽然失败了，但是它的枪却用在后来的 Area 51 中。我加入了 Electronic Arts 公司，并组织起他们自己的投币游戏开发团队。我的目的是开始制作消费类游戏。EA 公司中有我在 Atari 的一些老朋友，于是我决定加入他们的行列。我曾制作过一个谜题类游戏，但后来被我自己取消掉了，而当他们决定退出投币游戏市场的时候，我正在制作最后一个射击类游戏。然后我坚定地加入了消费类游戏的制作队伍中。



问：你是如何开始进行 N64 编程的？

答：我正在寻找一个可以制作的项目，所以我联系了许多公司，看他们是否可以提供。我原计划与来自 EA 的另一个程序员一起工作，但是他决定和一些朋友开办一家新公司。Atari 需要在 N64 上制作投币式的 Wayne Gretzky 3D Hockey 游戏，我也正希望能够在那种平台上制作游戏。其中不仅因为游戏有可能会比 PSX 的外观更好，还因为我们是第一个制作曲棍球游戏的。因此我加入了 Atari 的一个团队，并开始制作 Wayne Gretzky 3D Hockey。结果表明这项工作比我想象的要复杂，不仅因为 N64 开发系统的状况，还因为在我们发行游戏之前，投币游戏无法完成。



问：正如你所说的，玩 San Francisco Rush 这类街机游戏的乐趣就在于，可以坐在椅子上操纵变速挡、方向盘，感受反弹力等。你是怎样在并没有这些功能的 N64 上实现这些功能的呢？

答：你说得很对。家里没有街机游戏的环境，但我们可以家庭游戏中实现那些永远无法在街机游戏中实现的东西。我们可以为玩家提供更多的选择，使他们有更多的途径可以学习，并且可以发现更多的乐趣。

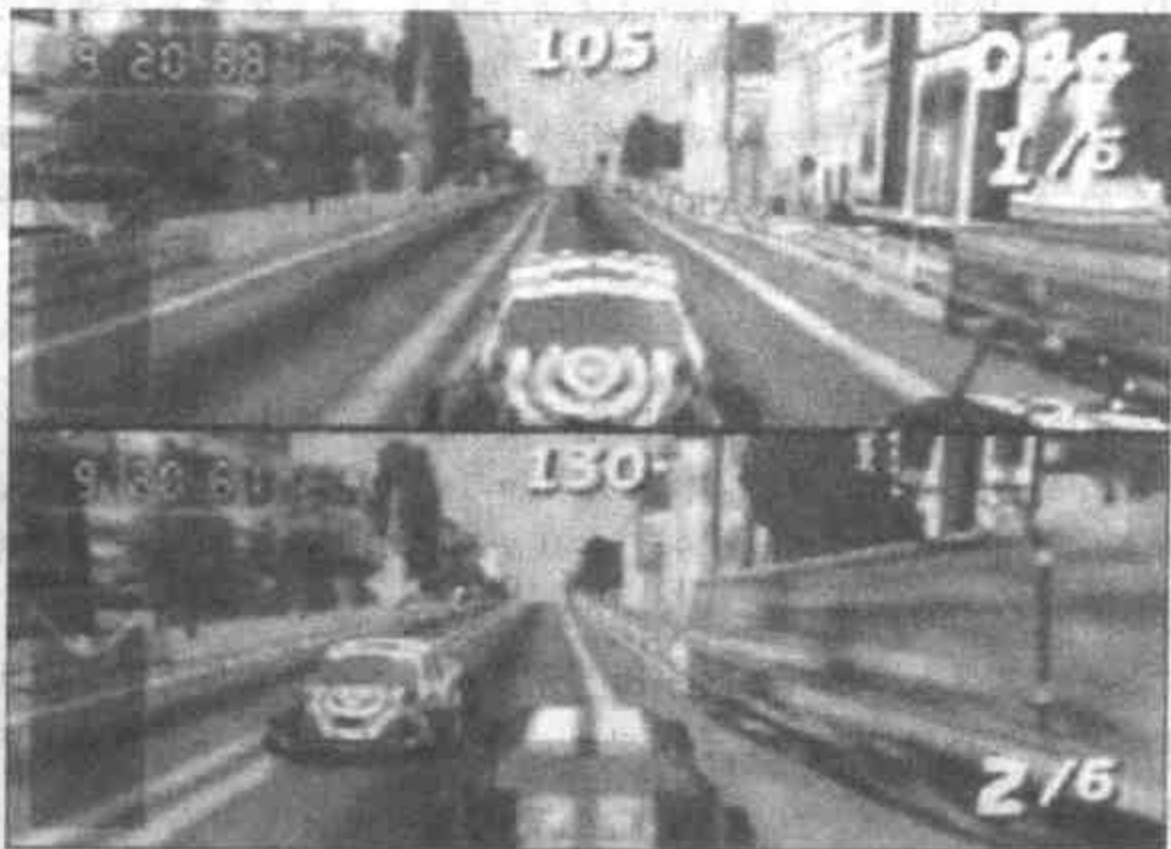


图 6.12 San Francisco Rush: Nintendo 64 上的疯狂赛车

我试图保持游戏的基本部分不变，但总会向游戏中加入新的内容。这是当我加入 Atari 时马上搞清楚的第一件事。Atari 需要我只做移植工作。过去这样做总是很奏效。但我并不认为这样做会有什么起色，所以我告诉他们我要在游戏中添加一些新的内容。例如，在 Gretzky 中我们添加了一个标准规格的溜冰场、新的 AI、立即回放、更多的玩家、完整的季节等。总之，家庭游戏有相当多的工作要做。我还认为我们可以为家庭游戏市场制作各种不同的游戏，而这些游戏是以前从未在街机游戏中制作过的。所以对我来说，这将带来新的机会。

街机游戏必须使用很明了的规则，以便于玩家学习，并为玩家提供 90 秒钟的娱乐。家庭游戏则不为这些规则所限制。对于家庭游戏，需要提供更多的生命值。这就意味着玩家需要更长的时间来“结束”游戏。即使玩家结束游戏之后，也必须能够提供一种理由，以对玩家为什么会重新再玩一次游戏做出解释。



问：在制作家庭版本的游戏时，你喜欢技术上的挑战吗？

答：实际上，我很倾向于“旧风格”，也就是尝试加入各种技术。我还喜欢加入一些技巧，并尽可能地提高帧频。将 SF Rush 所有的内容放在 8 MB 的空间里真是很有趣，其中包括 3 MB 的音频和所有的图像。





问：你抛弃了原来的设计方案？

答：是的，我的确抛弃了旧的设计方案。2D 环境对玩家来说比较容易了解。我还喜欢在游戏中加入固定的规则和足够的随机性，这样，玩家就可以设计出不同的玩游戏的风格和自己的策略。

我并不知道是否能将使用“旧设计方案”制作的游戏卖出去。玩家现在已经有了不同的期待。他们想要 3D 的设计环境，或者在 Internet 上进行游戏，或者需要高分辨率的材质和预告电影，或者需要非常高级的人物……除此以外，我现在所做的任何事情都会引出这样的评论：“这只是对 xxx 游戏加以修改，并带有 zzz 游戏的味道”。据我所知，许多旧的设计方案都是基于以前的游戏思路而来的。要知道，虽然 Asteroids 中有许多是原创的设计，但它还带有一点 Space War 的痕迹。



问：你已经在 Atari 工作了 20 多年，所以你一定非常喜欢这个公司。

答：是的，Atari 公司一直对我很好。我对公司和与我共事的人都有一种忠诚感。除此以外，我还喜欢自己的工作，所以我没有理由离开。我认为自己的忠诚要归功于工作的延续性。工作这样持久是因为我喜欢自己的工作。

制作游戏需要一些其他人所不具备的东西。许多人无法一直在压力下长时间地工作，他们认为自己的“身体”会在 18 个月的重体力劳动之后被累垮。而其他一些程序员和美工却认为这项工作非常有趣。

我必须承认我经常想去做其他的什么事情。但我只是没有找到比我现在的工作对我更有吸引力的事情。这种情况有可能会改变，也有可能到退休之前，我都会一直从事游戏制作工作。



问：在过去的几年里，Asteroids、Centipede 和 Gauntlet 都曾经重新制作过。你是怎样看待这些改版的呢？

答：许多改版就像大多数游戏的想法一样注定要失败。Gauntlet 的改版是个正面的例子，玩家对它的反响很好。Arkanoid 是 Breakout 的改版，也受到了玩家的欢迎。所以改版也可以受到欢迎，但却与原版有所不同。

真正的失败源于与原来游戏的比较。例如，制作 3D 版本的 Centipede 使游戏变难，因为 3D 信息使玩家更难于处理。要知道，设计者有 20 年的时间来玩这些旧游戏，并且加入一种新的想法就可以制作出一款优秀的游戏来。而他们没有这样做，看起来是表示不一定会做到。并不是不可能，而是不一定。



问：你想重新制作自己的哪一款游戏？

答：如果我能够回答这个问题，并且如果我认为它是最好的想法，那么我就会去做。除此以外，如果我告诉你，那么就会有人去做，是不是？换句话说，我不知道如何改进某些旧的经典游戏，并向其中添加新的内容，以在目前的市场上受到欢迎。

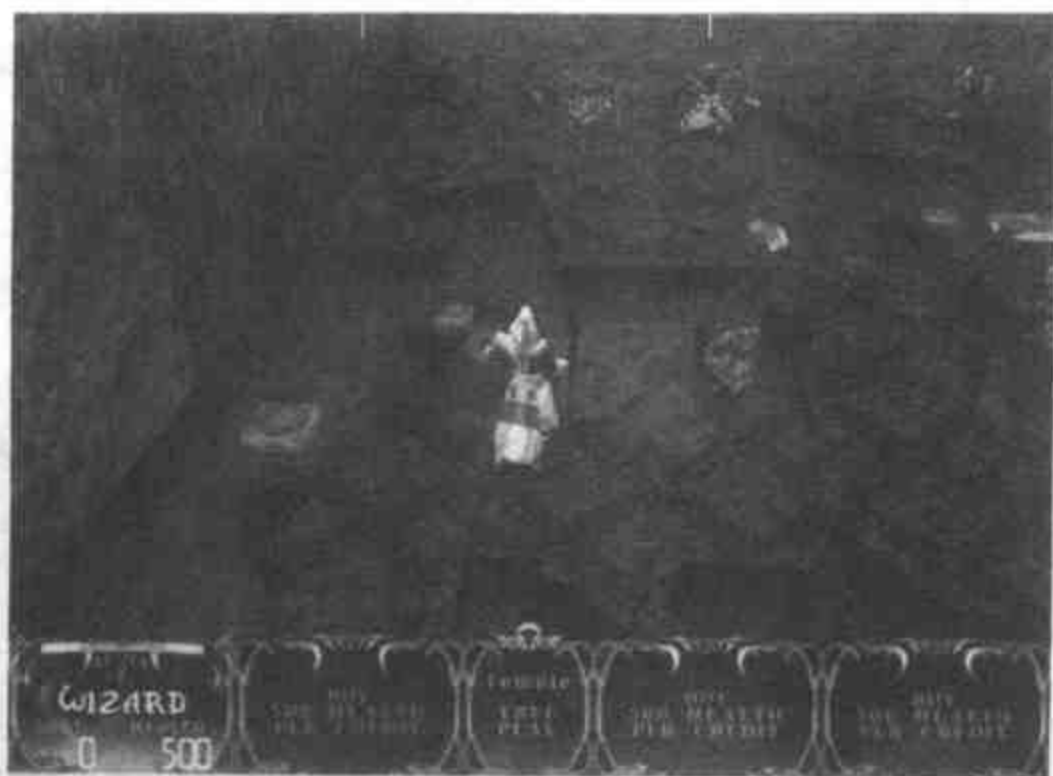


图 6.13 Gauntlet Legends



问：游戏开发行业在这些年里是如何发生变化的？

答：游戏行业肯定是在改变，但仍是电玩的行业。当我开始制作游戏的时候，电玩还没有形成 70 亿美元的产业规模。大公司的收入较多，因此必定要对资金支出进行控制。因此公司的管理层中就会有很多不同的意见。管理层的干涉是因为他们要对开支进行控制，而我认为实际的原因是因为游戏本身也在发展。我的意思是，在早些年里，我们可以在三个月内设计并制作出一款游戏。而发展到了今天，一个人在合理的时间内根本无法制作出一款游戏来。一个项目需要多个程序员、多个美工、一个声效专家、还需要有人在 12 到 24 个月内对项目进行管理。台式机游戏市场也在发生变化。前些年，你根本不需要花费 10 亿美元来启动新台式机游戏的项目，但是现在就要花这么多钱。因此，随着游戏开发周期的增加，制作游戏的成本也在提高。高成本带来了高回报，因此就要对资金的支出实施更多的控制（干涉）。



问：在你原来的设计中，你既是设计者又是首席程序员。你很喜欢同时担任两份工作吗？

答：如果你能处理好它们之间关系的话，同时担任游戏的设计者和程序员的确是个好主意。很少有人可以同时做好这两件事，因此现在我并不提倡采用这种做法。例如，对于现在复杂的多人物游戏和多关游戏来说，作为设计者，我并不能像制作其他类型游戏时做得那么好。所以我宁愿将这个工作交给别人来做。

程序员需要实时设计方案，如果设计者的思想并没有很好地传达给程序员的话，所制作出的游戏就会与设计者的设想有所区别。我认为程序员常会自己创造或者破坏游戏的“感觉”。

你似乎忽略了一个问题。在许多项目中，我还担任项目负责人的职务。这是一个我很擅长的职务，但并没有得到认可。我的项目总是可以按时完工，如果存在问题的话，管理层就会事先通知我。Atari 公司以外没有人知道。遗憾的是，我不喜欢这个职务，所以我总是在项目的管理工作上花费尽可能少的时间。





问：在以前的游戏中，你甚至做过美工的工作，是吗？

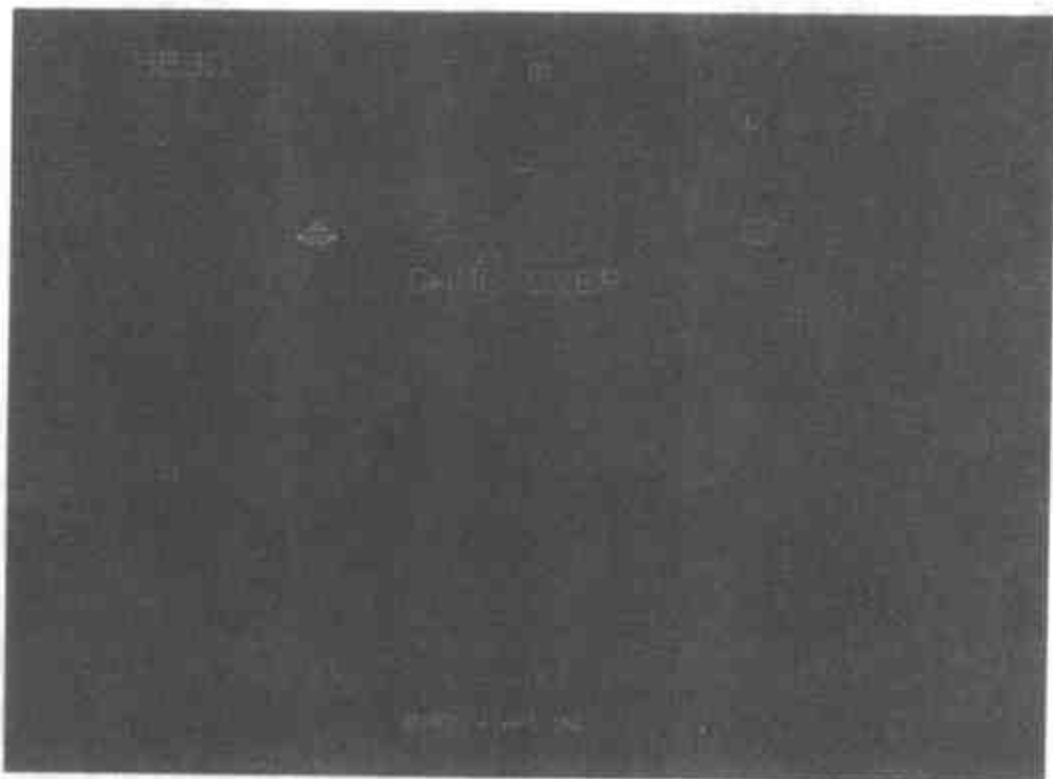


图 6.14 Asteroids

答：以前这是个不错的想法。没有理由训练一个美工，让他在图纸上画岩石，把画好的图纸交给我，然后我再把它放到我的游戏中。当游戏中对图形或声效的要求不太严格的时候，就没有必要找某个专业人士来做这样的工作。而现在情况完全不同了。目前，这样的专家是绝对必要的。



问：你感觉自己制作的某些游戏被人低估了吗？

答：作为游戏设计者，我并没有感觉我的游戏被人低估。如果游戏设计得很出色的话，那么就会很好玩，而且销售情况也会很不错。而作为程序员，我确实有这种感觉，有一些游戏的想法、算法和编程速度没有得到重视。许多编程技巧都是我的个人喜好，所以我并没有寻求外界的认可。



问：在过去，技术给开发者的限制性较大，你认为技术上的限制明显改善了吗？

答：是的，我不得不承认。过去，许多次地，我要花费时间思考在给定的硬件上做某些事情，并在游戏上实现。Xybots 当然是这类游戏之一。Gauntlet 建立在新硬件上，可玩性更好。



问：在进行原创的游戏设计中，你如何起步？

答：首先我会提出游戏的计划，然后研究一下游戏的所有方面。从市场的角度：看游戏是否会有销路、发行时间是否适宜以及许可需求和竞争等因素。从玩家的角度：游戏的好玩之处在哪里、什么是游戏吸引人的独特之处。从开发的角度：在人物和装备上要怎样制作游戏，如何好玩。想法本身来自各种可能的来源。我曾提到，其中某些想法来自以前的游戏、大家集思广益想出来的主意、技术挑战，还有其他人的建议。



问：那么，一旦你有了自己的主意，你是立即开始编码呢，还是安排一些时间提前想出计划呢？

答：由于现在的游戏在预算和开发队伍方面都有很大的规模，所以在进行长期的开发工作之前，编写游戏设计文档和技术设计文档是很有必要的。然而，在起草设计文档的同时，我就已经在某些关键方面开始工作了。我认为在可能会影响或打断游戏设计概念的方面开展工作是极其重要的。游戏前的电影、故事线索、前后端屏幕的制作都可能等到游戏成型之后再制作。有时这个原型阶段是很快就可以完成的，但一般都要经过几个月的时间。



问：一旦你证实在原型中游戏的概念是可行的，会怎样进行其余的开发过程呢？

答：对我来说，游戏的开发可以分为四个阶段。在项目一开始，就要有一些有创意的做法，让人感觉这的确是个很棒的游戏。在很短的时间内，项目就会有大的跨越。项目的中间部分很平常，验证游戏设计的概念，但仍有很多工作要做，游戏也不会因为你所有的努力而有多大的变化。第三个阶段通常充满了紧张和压力。这是当你想结束项目，在发行游戏前的那个阶段。第四个阶段是在发行了游戏之后感到满意结果的时候。

目前的项目越来越大，我经常感到对于自己的努力，所得到的回报越来越少，所以我很高兴能完成游戏。在我看来，几乎所有我在游戏中的计划都实现了，所以我很高兴地告诉自己项目完成了。尽管在最后一刻还是会发现令人讨厌的 bug……



问：在原型完成之后，你就对开发过程不太感兴趣了？

答：是的，可以说在核心游戏概念证明可行之后，游戏的大部分开发工作就已经完成了。然而，在开发工作完成之前，在漫长的开发过程中仍有一些问题是值得探究的。在过去 30 年里制作了这么多游戏之后，继续工作的时候，可以说用户界面已经无法再让我感到兴奋。

不，我会设计出原型，然后让其他人去完成具体的工作。但我对游戏的设置仍具有一定的想象力，而不会信任其他人或开发团队能继续完成我所设想的游戏。因此最终我感觉游戏并不像我所预期的那样，也不再是我的了。我觉得如果我能亲自来制作游戏直至最终完成，一定会比其他人做得要好。我猜想如果我没有看到自己成功地制作游戏的话，自己的感觉就不会如此了。



问：你认为 AI 在游戏中起着什么样的作用？

答：在旧游戏中并没有 AI。通常，敌人都会遵循固定的规则，并会有一定的随机性。而现在，情况却完全不同了。AI 对现代游戏非常重要。有些人建议，在适当的时候，每个项目都要有专人来研究游戏的 AI。对于某些游戏来说，我同意这样做。



问：为什么你认为游戏需要更完善的 AI？

答：我认为主题和大多数新游戏的设置中都需要更多的 AI。虚拟类游戏、射击类游戏等都尝试提供一种真实智能生命的感觉。如果游戏不能模拟真实生命，那么有些规则就不太适用了。





① 问：你认为让游戏中的 AI 具有“真实性”有多重要？也就是说，仅使用玩家处于 AI 主体位置上所具有的信息来提供 AI 吗？

答：这并不是必要的，但可能会导致更为可信的敌人 AI，所以在某些情况下我建议这样做。例如在 Steel Talons 中，在看到你之前，敌人的枪手不会转过身来或者向你开枪。如果在路上有座小山，或者就在他们视野的边缘紧趴在地上，他们就看不到你。这种情况下就很有必要加入 AI。

② 问：后来，游戏与故事的结合越来越受到重视。可能是由于其特点使然，许多街机投币游戏几乎都没有故事情节。你是怎样看待在游戏中讲述故事的做法？

答：我从未重视过故事情节。我觉得让玩家抓住主题是绝对必要的：设置、周围的环境还有目标。有时，故事情节可以使游戏的目标更容易让人理解。有些游戏制作出来就像一场电影，所以故事情节可以起到很好的作用：玩家感觉自己就是游戏中的主要人物。在投币游戏中，除非使用 attract 模式，否则故事情节就毫无意义。当玩家想玩游戏或者投入更多的硬币时，我们不想让玩家浪费时间来看什么东西。

③ 问：你以前提到过，你非常想为家庭市场制作游戏，这是为什么呢？

答：我想制作家庭游戏而不是投币游戏，这是因为我看到在家庭游戏市场中制作新游戏的机会。

④ 问：你觉得投币式街机游戏还有什么前途吗？

答：就像你看到的那样，我觉得街上的投币游戏会朝向更便宜的模拟驾驶（身体移动或者被环境包围）的方向发展。这些游戏是家庭游戏所无法提供的，而且也比在迪斯尼乐园玩耍便宜得多。我认为投币街机游戏的市场仍然存在，它的发展方向是需要更加便宜。所以我觉得投币盒或者不需要长时间开发的带有简单游戏的廉价 PCB 更有发展前途。

我认为消费市场已经控制了投币游戏行业。我无法提供具体数字，但只需看看热门游戏的销售量和他们的价格就很清楚了。这些情况可以反映投币游戏行业的变化。我很高兴自己成为这个行业中的一分子。我



图 6.15 San Francisco Rush 2049 的街机版本





感觉自己在恰当的时机得到了好的机遇。

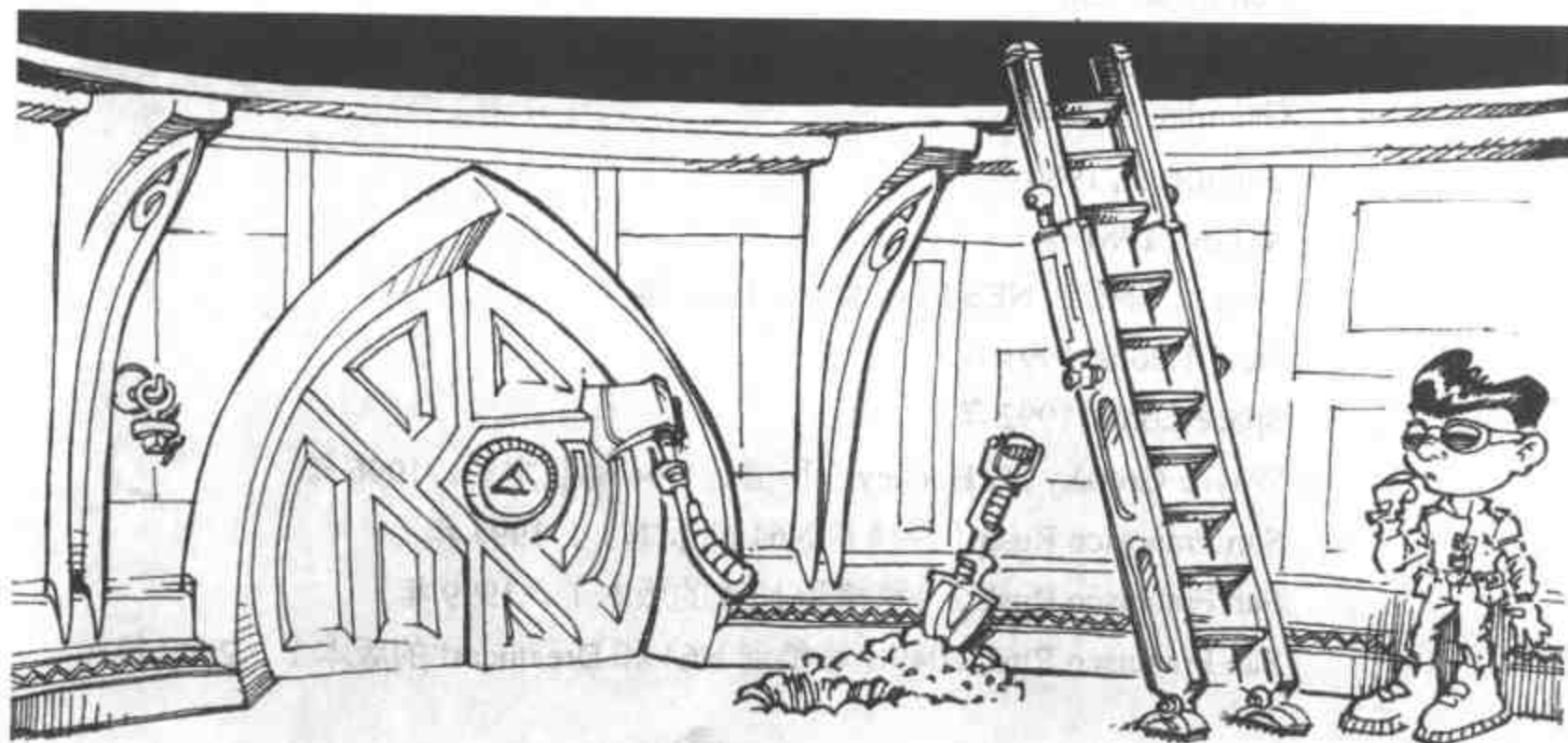
Ed Logg 作品一览

- Super Breakout, 1977 年
- Video Pinball, 1979 年
- Asteroids, 1979 年
- Othello (用于 Atari 2600), 1979 年
- Football (四人版本), 1979 年
- Centipede, 1981 年
- Millipede, 1982 年
- Gauntlet, 1985 年
- Gauntlet II, 1986 年
- Xybots, 1987 年
- Tetris (移植到 NES 的版本), 1988 年
- Steel Talons, 1991 年
- Space Lords, 1992 年
- Wayne Gretzky 3D Hockey (移植到 N64 的版本), 1996 年
- San Francisco Rush (移植到 N64 的版本), 1997 年
- San Francisco Rush 2 (移植到 N64 的版本), 1999 年
- San Francisco Rush 2049 (移植到 N64 和 Dreamcast 的版本), 2000 年



Chapter 7

第7章 游戏可玩性的元素



“我们制作了一个自己都不知道如何获胜的游戏。虽然我设计了游戏的整个系统，但是我仍不知道什么才是最佳的战略或战术。正是这一点造就了优秀的战略游戏。”

——这是 Julian Gollop 在论及他的游戏时所说的一番话
X-Com: UFO Defense

要制作出一款真正优秀的游戏，到底需要哪些设计元素呢？当然，这个问题没有确定的答案。虽然如此，作为游戏设计者，你应该可以凭着直觉得到这个问题的答案。理解游戏设计，如同理解其他艺术形式一样，这是一种非常内在的理解，一种“发自内心”的反应，一种也许会有“感觉”。这种感觉可能无法用语言来表达，但是你需要了解游戏的优缺点，并且知道如何使前者可以取代后者。游戏设计者和玩家的经验在理解游戏乐趣的过程中占有重要的地位。

根据我多年玩游戏和创作游戏的经验，关于如何制作出优秀的游戏，我已经有了自己的答案，并且在本章中我将来探讨其中的一些特性。其中的某些主题相互之间看起来好像毫无关系，但在我看来，在制作优秀的游戏时，它们却具有非常重要的作用。当然，我无法列出我所知道的所有知识，因为正如我提到过的，比起我希望写在书中的知识，我的理解不过是一些类似“第六感”的东西。但在本章中所介绍的概念应该会对你有所帮助。



7.1 独特的解决方法

作为游戏设计者，我最兴奋的时刻之一就是，当我听到有人正在谈论如何玩我的游戏，并且他们说出了我从未考虑过的成功战术的时候。这也许是一个特殊谜题的解决方案，一种降低游戏挑战力度的途径，或者一种穿越危险峡谷的方法。我认为我开发的游戏好像处于创造状态中，游戏玩家能利用他们自己的创造力来获得成功。当玩家的创造力带来了他们那些我从未料想到的解决方法时，说明我的游戏正在完成它的工作。

7.1.1 预想相对复杂的系统

优秀的游戏设计人员会设法猜测玩家下一步的行动，并且使他们的游戏能很好地对那些行为做出反应。例如，拿一个RPG角色扮演类的游戏作为例子，游戏中描述了这样一个谜题：将一些重物放在一系列压力板上（我曾经将这样一个谜题放在我自己的游戏中，但我想恳求其他游戏设计者能够想出较之更有创意的设计方案，因为压力板是现有的谜题装置中使用过多的装置之一。但是我离题了）。假设在压力板的谜题中，设计者在一些房间里留下了一堆引人注目的石头。谜题最明显的解决方法就是，把这些石头放在压力板上来完成预想的结果。但是如果玩家尝试将他不同的武器放在压力板上怎么办？这是一个极佳的有效解决方法，只要玩家有相应重量的武器，它同样也能起作用。如果玩家有召唤小怪兽的魔法，使得玩家能够召唤出许多种不同的小怪兽怎么办？如果玩家在压力板上召唤那些怪兽，他们也许一样会获得成功。

现在设计者在完全考虑过这个谜题后，可以让程序员在游戏中加入代码，使得游戏系统对于无论板上放置的是岩石、武器或是怪兽都能正确做出反应。这就是游戏设计的预测训练，设计者考虑玩家下一步的行动并且让游戏很好地与这些行动协调。我承认这种策略确实比只允许一种解决方法要好。然而，如果玩家想出把其他一些重物放在压力板上怎么办？如果玩家在压力板上使用Berkshire暴雪魔法，让雪落在压力板上怎么办？在板上堆积足够的雪也有相当重量。可是，如果游戏只与岩石、武器或是怪兽们相关联，游戏系统将不能做出正确反应。玩家想出了一个完美且合理的方法，但游戏系统却不识别。

但是，设计者可以让程序员做出这样一个系统来代替当中的联系：游戏中每一个对象都有与之相对应的重量，这包括了岩石、武器、怪兽、天气因素、洪水以及一切能在游戏世界中发现的东西。然后，如果程序员让这些压力板仅获得放在它上面对象的重量，而不管它们的类型，那么这个综合的方案将对所有对象起作用。如果每个对象都被设置一个合理的重量，那么无论玩家在板上放置什么对象都不会有影响，因为它们将自动起作用。

后一种方法少一些游戏设计的预测系统，但在实现上更为整体全面。它更多地依赖于创造一个游戏运行所需要的可靠稳定的系统。然后对于一个诸如上面描述的压力板的谜题，设计者和程序员能想出一系列成功的条件。规则变成了“当板上放置了正确的重量时，谜题解决”，而不是“如果玩家使用岩石、武器或是怪兽来启动压力板，谜题解决”。当然，这个谜题例子有些简单，但是同样的技术能够被应用到更加复杂有趣、容许多种成功游戏风格的系统中。





7.1.2 显露

众多强大且合乎逻辑的系统开发能引导玩家在游戏环境下产生他独特的解决方案。你可以用一个游戏设计圈内颇为流行的时髦词来描述这些解决方案，它们是从游戏的系统设计中“Emergent(显露)”出来的。建立一个与玩家容易理解和所利用的逻辑规则相一致的游戏世界，它允许玩家对于游戏中的问题提出他们自己的方案。当玩家尝试用某种不明智且不明显的方法来解决谜题或战斗时，发现这个方法确实有效对他们来说就是最好的奖赏。相互之间同时正常工作的系统越复杂，各种环境下的解决方案就会越丰富多彩。拿拥有众多并行系统的游戏——Civilization 来说，这些系统共同创造了最有趣的的游戏。



图 7.1 Civilization 系列的游戏是交互稳定的并行系统显露复杂游戏操作的最佳例子之一

另一个此类“显露”战略的例子在早期的 Centipede 中也能找到。任何曾经玩过这个游戏的玩家都知道要想游戏玩得时间长些，堆蘑菇是一个最大的障碍，而且许多玩家都会了解尽可能保持游戏区域干净的重要性。当在游戏中遭受一次次失败后，他们开始注意到一些模式。首先，他们认识到，虽然被毁掉的蜈蚣残部也会扔下问题蘑菇，但是跳蚤扔下了其中的绝大部分。其次，他们知道了跳蚤不会在游戏的初始阶段就出现。第三，跳蚤的出现总是与半屏底部蘑菇的缺少有联系，这一点被玩家观察到。这样，一个连游戏设计者 Ed Logg 都没有预料到的方法，著名的“blob”战略就产生了。要使用“blob”战略，玩家必须清除掉初始时面板上的全部蘑菇，然后只允许屏幕右下方的蘑菇存活。如果在小心地摧毁了蜈蚣后，仍然只有这个区域有蘑菇，那么跳蚤就不会出现，这使得游戏的难度大大降低。这个解决方案是玩家在 Centipede 游戏中为了追逐高分而显露出来，当它被发现时，毫无疑问玩家会对此非常自豪。此外，甚至游戏的创作者 Logg 也不知道会有这样的发现，那就是优秀的游戏设计。





7.2 非线性

非线性是游戏工业中又一个流行且很恰当的词汇。非线性是一个有趣的游戏所能包含的一切，但在工作中这一点却为许多游戏设计者所遗忘。非线性赋予了互动的意义，如果没有非线性，游戏开发人员不如去做电影。在你的游戏中非线性的部分越多，游戏就越优秀。

通常来说，当有人说某样东西是线性时他们是指它沿着一条线而行。线是二维或三维空间中一系列连接在一起的点，可以用一个特殊的方程找到这条线上的任何点，例如，在二维的情况下， $y = mx + b$ 。在外行人看来，这意味着线必须是直的。如果有人考虑线上的任意两点，称它们为 A 和 B，从 A 到 B 只有一种连线的方法，没有其他的选择；必须通过 A 与 B 之间所有的点。在数学世界之外，我们也能将读书看成是一个线性的例子。如果一个人正在阅读一本 323 页的书，并且他不会略过某些页或某些章节，那么读这本书只有一种方法：从第一页读起，读完直到 323 页的所有页。

然而，游戏是非线性的工作。在下棋过程中，有多种方法可以捕获对手的国王，从游戏开始到游戏结束可以有多种玩法。确实，下棋有许多种不同的方法，然而正是这种多样性使游戏变得有趣。这些选择使棋类游戏具有非线性。假设棋盘是一维而不是二维的，每个玩家的棋子只能朝一个方向移动，并且每个玩家只有一个棋子。这种棋就是线性的，因为没有可供玩家考虑的选择，每一个游戏的结果都可以预见。当然，它也会失去很多的乐趣。



7.2.1 非线性的种类

这样，当我们说想让我们的游戏具有非线性时，是指我们想让游戏从开始到结束能提供给玩家更多的选择，提供给玩家从点 A 到点 B 众多不同道路的选择权。我们可以通过不同的途径实现这一点：在游戏的故事情节中、在玩家完成游戏挑战的过程中、在玩家排除这些挑战的顺序中，以及玩家选择遭遇到怎样的挑战的过程中。所有这些成分构成了一个游戏的非线性成分，而且游戏设计者创造的非线性成分越多，每一个玩家的体验就越不相同。此外，每一个不同非线性部分之间的互动使得它们的整体比它们简单的组合要棒得多。

- 故事介绍：在第 11 章“故事介绍”中，我将更细致的讨论非线性的故事介绍。当然，一个非线性的故事主线同一个非线性的游戏是紧密联系在一起的，如果游戏本身只提供玩家非常少的选择，那么没有人会为设法制作非线性的故事情节而操心。故事叙述也许是非线性过程中最容易被忽视的游戏部分，因为许多开发人员在容许非线性游戏的同时却把他们的游戏限制在一个完全线性的情节中。
- 多样的解决方案：在上面我已经讨论过了，一个设计优秀的游戏怎样使玩家在面对游戏挑战时想出他自己的解决方法。每一个玩家在解决一种情形时所考虑的方法都不会一样，假设这些解决方法都是合理的，那么任何挑战都必须提供给玩家多样的途径来克服它。游戏中单独的挑战都拥有多样的解决方法，在非线性中占很大一个部分；它使得玩家从点 A（遇到挑战时）到点 B（解决这些挑战）有多种的途径。
- 顺序：除了用独特的方法解决挑战外，玩家还希望能够选择他们所遇到挑战的顺序。许





多冒险类游戏都有“过度线性”这个错误，它们只允许玩家在一个给定的时刻遇到仅仅一个谜题。要想出现第二个谜题，玩家必须完成第一个。这是一个线性的思考方式，由于游戏线性的本质，当玩家在一个特别的谜题处受到阻碍时将不能做任何事情，除非谜题被解开。给予玩家解开不同谜题的选择权将容许他们把一个麻烦的谜题放在一边，然后去解答另一个谜题。在完成了第二个谜题之后，重新恢复活力和精力的玩家可以返回到第一个谜题上，这样也许会有一个更好的机会来解开它。

- 选择：另一种使游戏非线性的方法就是，允许玩家选择他们想要克服何种挑战。比如说，在游戏中点 A 和点 B 之间有三种无序独立的挑战，X、Y 和 Z，那么玩家就可以按照他所想的顺序来完成这些挑战。如果玩家一旦克服挑战 X，他不必回头去解决挑战 Y 或 Z，就可以到达游戏中的 B 点，也许永远不回到 Y 或 Z 怎样？如果玩家不选择 X 而是选择排除 Y 或 Z 也是一样可以的。任何一种选择都容许玩家继续游戏。这样的好处在于，如果玩家发现挑战 A 不能克服，他可以选择尝试 Y 或 Z。这极大地降低了玩家永久受阻的几率。Z 不需要 Y 比 X 容易的情况；不同之处仅仅在于，它还是依靠玩家自身的力量但增加了玩家过关的机会。其他玩家也许会认为 X 比 Y 或 Z 要简单，但给予玩家选择挑战的权利使每个玩家对于通关都能实施其独自的技巧。当然，在完成了挑战 X 后，为了游戏的乐趣或是为了增加到下一关的机会，玩家也可以选择返回完成挑战 Y 和 Z。也许完成了挑战 Y 和 Z 可以给玩家所扮演的角色增加更多的经验值或财富。这种非线性同样也可以用来在游戏中增加一些完全可供选择的支线任务。虽然它们也许可以降低游戏难度或是仅仅是增加一些娱乐，但玩家在通关过程中无须严格地完成这些任务。无论情况如何，这些可选择的挑战增加了非线性的程度，促使玩家的游戏经历更加个人化。

7.2.2 执行

我的第一个游戏——Odyssey: The Legend of Nemesis，毫无疑问是我曾经做过的游戏设计中最具有非线性的，它包括了以上所描述的所有的非线性特点。Odyssey 是一个角色扮演类游戏（RPG），故事发生在一个有待玩家开发的七个主要岛屿组成的群岛上。虽然玩家在去游戏中其他岛屿之前必须至少完成第一个岛上的一个任务，但有两个任务可供玩家选择并且对于玩家来说每个任务都有多种解决方案可供选择。甚至，如果有聪明的玩家知道怎样找到一个特别的城里人（townsperson），他也可以略过这些任务。玩家可以从那里自由移动到其后的五个岛屿上，挑选他想去开拓和仅仅想经过的岛屿。当然，玩家必须到达第七个岛屿，并且要想结束游戏玩家需成功地探索每一个岛屿，杀掉阻碍他的怪兽。自然，如果玩家能获取完成任务的奖赏，杀死那些怪物就会变得简单一点。但是如果玩家这样选择，就会略过游戏的整个中间部分。当然，极少有玩家会这样做，大部分玩家都会去探索他所遇到的不同的任务和情景。这些任务中几乎每一个都有多种解决方法，而且玩家的每一个行为都会对完成这个岛屿上的情节有直接的影响。最后，游戏本身还有多种结局，随着玩家所完成任务的不同也会有不同的结局：生存、死亡，或是一种公正与和谐。虽然游戏有固定的故事情节，但我还是相信任何尝试过此游戏的玩家所采用的方式都不会完全相同。



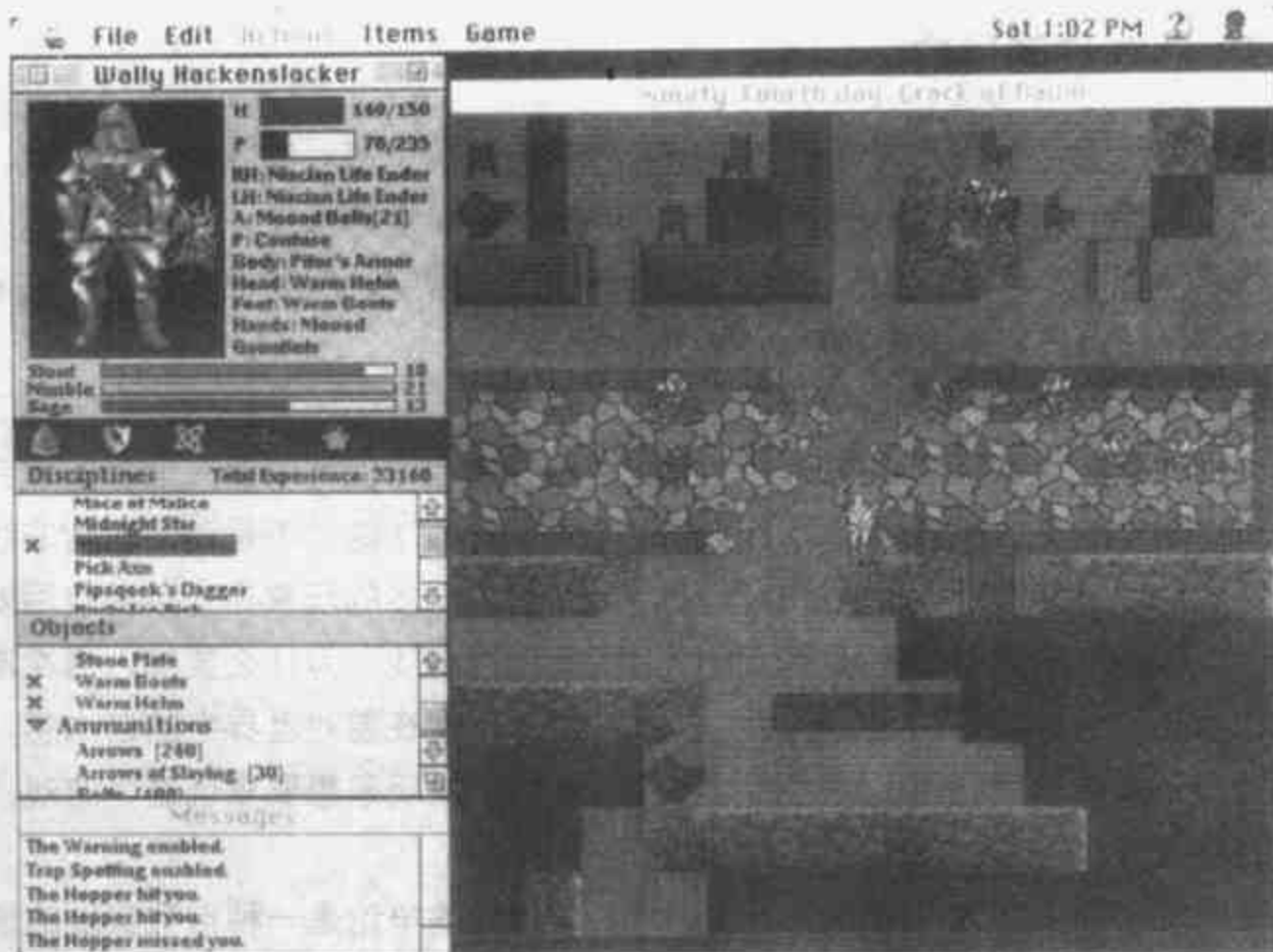


图 7.2 Odyssey 是一个极度非线性的游戏，它允许玩家按他们所选择的顺序来完成任任务，还可选择要完成的任务。游戏中任何一个谜题都有多种解法

在设计游戏的过程中，非线性是一个强有力的工具，设计者可以使用以上所描述的这些非线性类型，在玩家看来这些非线性类型也许是显而易见的。但让人惊讶的是，许多的游戏没有提供给玩家任何实质的非线性，相反坚持玩家按一条从点 A 到点 B 的单一路线来玩游戏。这样做的一个原因就是，创造所有这些非线性的因素可能要耗费掉大量时间。拿点 A 和点 B 的例子来考虑，虽然有以上所提到的挑战 X、Y 和 Z，但是玩家要继续游戏只需要克服其中一种，比如说挑战 X。也许玩家直到玩到游戏结尾也不会与挑战 Y 和 Z 进行互动。作为一个非线性的游戏，这就是玩家的特权。但当一个成本会计看到这个游戏，并且想计算出何处才能调整这款游戏的预算时，问题出现了。好吧，明显地，如果 Y 和 Z 都没有什么严格的用处，为什么还要保留着它们呢？如果玩家有可能永远不会看到 Y 和 Z，为什么还要在 Y 和 Z 的编程、美工及设计上花费大笔的钱呢？不幸的是，会计通常对游戏中的这些好观点了解很少，所以当你说，“非线性能让这款游戏变得优秀！”时，他们多半会因为“深奥”而对你不加理会。

非线性通常也很难从一个设计中实现，当然更要比简单的线性难得多。这也许就是设计者在即使有第一种优势的情况下仍要避开它的另一个原因。当玩家都遇到同一个挑战时，需要提供给玩家足够多的不同选择，这其中许多的设计障碍是难以排除的。在 X、Y、Z 挑战的例子中，如果 Z 明显的比 X 和 Y 要简单得多，就不会有人去为 X 和 Y 费心。在某种程度上说，一个设计差劲的游戏与一个完全没有其他选择的游戏没有什么差别。你的游戏提供的非线性对于玩家来说必须是有意义和有用的，否则就是废品。那些过多关心自身设计技巧的设计者也许会在他们的设计中缺少非线性，因为他们想让玩家经历包括在游戏中的每一个元素。“为什么要在游戏中不是每个人都能看到的部分花上大量的时间？”这些以自我为本位的设计者这样说，他们开始听起来像那些会计了。





7.2.3 非线性的目的

从一定意义上来说，包含非线性的游戏是让玩家按他们玩游戏的方式编写故事，而时刻记住这一点是很重要的。如果从游戏开始至结尾都强迫玩家持续停留在一条特定的线上，他们就会感觉像是被束缚住了。即使这条线上的挑战也许构思非常好，但如果玩家除了按顺序来接受这些挑战而没有任何选择，那么慢慢地游戏的乐趣就减少了。

玩家反复玩某款游戏的一个重要原因就是非线性。一个所有挑战都被克服了的游戏再玩起来就不会那么有趣。当玩家再次玩同一个非线性的游戏，他可以避开上次游戏完成的挑战而去选择游戏中其他的支线。然而，认识到游戏可以重玩的功能并不是游戏包含非线性的主要动机。我曾经听到过一些游戏设计人员的抱怨，因为有如此多的玩家在开始游戏后就没有完成过游戏，所以再次游戏的能力无关紧要。既然他们完成过游戏，为什么要加入再次游戏的能力呢？这些设计者还没有认识到正确的观点，非线性赋予玩家在游戏世界中的自由感，让每个玩家都有自身独一无二的游戏经历，讲述他自己的故事。如果玩家想要重玩这个游戏，那很好，但是非线性的首要目的是在一定程度是让玩家成为作者。

此外，关于玩家极少完成游戏因此游戏不需要非线性的争论是一种自我实现的预言。玩家不能结束游戏通常是因为他们在某个特别的连接点被困住了。这也许是关底太难，谜题太让人迷惑，或者仅仅是找不到一个给定领域的出口。如果游戏更非线性，那么，玩家在游戏中任一个点被困住的机会就少得多，因为多种多样可行的方法增加了玩家用他们各自不同的才能通过其中一点的可能性。

在一个论题为“主要的标准游戏理论”的游戏开发人员会议上，Noah Falstein 建议当非线性允许玩家按他们所想的顺序来完成一系列必需的挑战时，完成其中的一个应该降低其他挑战完成的难度。如果是收集谜题的情况，可以通过在玩家完成一个谜题后提供给他们其他谜题的暗示来做到这一点。如果是要遭遇某种战斗的情况，可以通过提供给玩家额外的武器，以便使他们在其他的战斗中存活下来这种方式来做到这一点。不管什么样的情况，使用这种技术增加了玩家完成现有挑战继续游戏的机会。

要谨慎的一点，所有设计者应该明白，非线性不是要让玩家毫无目的的在游戏世界中游荡。如果在游戏的一个非线性的点上玩家不知道该完成什么，或者不知道应怎样着手做，这个非线性也许就做得有点过火了。许多游戏设计者在鼓吹他们正在开发的游戏时经常用这样的声明，

“在我们的游戏世界中，玩家可以做任何他们想做的事情，没有任何限制。游戏是完全非线性的。”这样的游戏同样也会让人完全讨厌。当然，到那时这些装载了绝大多数非线性的“完全非线性游戏”被剥离出来，玩家被留下来按轨道来解开谜题。在“按轨道”的游戏和完全自由中间有一个概念，玩家既有自由的感觉又有被引导的感觉。



7.3 虚拟现实

在电脑游戏中，虚拟现实是这么多年来推动游戏发展的一个动因。那些支持者说，我们做的游戏越真实，游戏者就会越投入。但会总是这样吗？在一个诸如 Tetris 或 Centipede 的游戏中





加入更高层次的现实性会怎么样？它肯定不如现在这样能让人投入。拿 *Civilization* 这样的游戏来说，它已经虚拟了一些真实性。如果再增加一点真实性能够使它更加有趣吗？实际上，恰恰相反：在游戏中加入一种更真实的经济模式或是战斗系统会降低游戏的力度，而且很有可能使游戏变得让人讨厌而不是有趣。

当游戏要模拟的现实超出玩家的实际需要时，虚拟现实的麻烦就来了。Alfred Hitchcock 将电影描述成“剔去污点的生活”。其实，游戏也可以被看成是略去了沉闷乏味部分的虚拟生活或者生活的某个方面。如果一个设计者执意要达到一个更高层次的现实度，而决定在游戏中加入太多无关紧要并且无趣的细节，游戏玩起来就会单调乏味。我最喜欢的一个例子就是角色扮演类游戏中食物的使用。许多 80 年代的角色扮演类游戏总是设法使自己的游戏比其他角色扮演类游戏更真实，这样每一款新游戏的推出都在加高这种赌注。有一种设计者使用的方法就是增加了食物，玩家要记住不时地给他的角色喂食物，不然他们就会饿死。这就是不需要加入的无趣之处，尤其当玩家在一个危险的世界中冒险时，规律性的饮食不会是玩家脑中第一个跳出的念头。

然而，以现实生活作为游戏的基础也有它的好处。首先最重要的一点，它提供给玩家一个他们所熟知的世界，他们知道什么样的行为是合理的，什么样的行为会出问题。不论是在 *Civilization*、*SimCity* 或是 *Deadline* 中，一定程度的真实环境可以帮助玩家直接“进入”到游戏世界中。他们知道或自认为知道这个世界怎样工作，以及他们做什么才能成功。玩家开始玩游戏后不久就会知道他们要完成些什么。另一方面，一个像 *Centipede* 或 *Tetris* 这样更加抽象的游戏，要通过阅读指引或通过在游戏世界中的经历来教给玩家他们应当完成的抽象的目标。



图 7.3 早期像 *Marathon* 这样的第一人称的射击类游戏不允许玩家跳跃或者爬行。但是第一人称射击类游戏的现实性出现不久就引起玩家要求加入这特点





依赖现实世界的潜在不良方面在于，因为游戏模仿了一个玩家所熟悉的现实，玩家将会期望游戏世界中的元素都按一种确定的方式工作，并且他们很快就会注意到那些不能做到这些的元素。比如，许多早期的第一人称的射击类游戏，好比 *Doom* 和 *Marathon*，以前并不允许玩家角色跳跃。这些第一人称视角射击类游戏的世界比玩家熟悉的其他电脑游戏世界更“真实”，它们是如此的真实以至于玩家的期望被提高，他们很快便抱怨他们甚至连一个只有齐腰高的障碍都跳不过去。因此下一代的第一人称射击类游戏加入了跳跃的能力，然后是爬行的能力，再然后是可以向上和向下看，就这样随着每一个现实性因素的增加，游戏越来越复杂。现在，因为拥有实时 3D 引擎的世界看起来比以往更加真实，所以玩家经常问这样的问题，“为什么我不能平躺在地面上？现实生活中我就可以这么做，在游戏中为什么不可以？”可以这样说，特别对于那些初玩游戏的玩家，这些第一人称视角的射击类游戏正是因为他们这种对虚拟现实的需要而变得过于复杂。一定程度的虚拟现实能以 *Centipede* 或 *Tetris* 游戏世界完全做不到的方式来提高玩家的期望。玩家从来不用怀疑他们在这些世界里的适应能力，因为首先这些边界全部都是随意的。

那么这就是在游戏中应不应该虚拟现实的确切答案吗？当然不是，在所有的游戏设计中没有像这样简单的答案。作为一个游戏设计者，你必须打破现实和抽象的平衡，从游戏可玩性的观点出发，按照故事情节以及设置要求和机器的合理操作来衡量你的游戏需要什么。需要谨记的，同时也是许多设计者经常忘掉的，就是太多的现实并非一定是好东西。



7.4 指导玩家

虚拟现实也许能让进入游戏世界的玩家拥有优势，通过他们自身的生活经验，玩家将会了解游戏世界。然而，即使是最现实的游戏，玩家也需要时间来学会怎样玩游戏。这种学习的经历通常就是玩家在游戏所有经历中最重要的一段时间。一个玩家在你的游戏上最初花费的这几分钟经常会做出决定，是否继续玩下去。每当一个玩家将游戏告诉其朋友时，新玩家会记住最初的那几分钟，然后说，“哦，玩起来有些怪”，或是更好一点，“它棒极了。我直接进入游戏而且发现还不错。”

在过去，许多电脑游戏都是依靠手册来指导玩家玩游戏。如果不先读一大块手册，玩家几乎没有完成游戏的机会。因为认识到玩家在买来一款新游戏后最后想做的事情才是坐下来看游戏手册，所以今天许多游戏都想脱离这种对读者阅读能力的依靠。玩家显然都有一种直接拿起操纵杆开始玩游戏的强烈欲望。现在有如此多的游戏准许玩家这么做，允许玩家“直接进入”游戏的重要性增加了。如果你的游戏一开始很难上手，玩家也许就会放下它去玩其他的游戏。

这并不意味着你的游戏必须降低智能或是简化，仅仅意味着你必须通过游戏而不是手册来介绍你的游戏世界的复杂性。比如，在游戏开始可以让玩家做一些简单的动作。假如说你在创作一个类似 *Tomb Raider* 的第三人称动作/冒险类游戏。一开始教会玩家怎样正确移动角色最重要。然后，在玩家有可能熟悉了这种水平运动的操作后，你可以介绍一些场景给玩家，让他们跳过一个溪谷或是爬上一个悬崖。在这些都足够了以后，你可以介绍一些简单的战斗任务，让玩家了解怎样使用所扮演角色的武器。

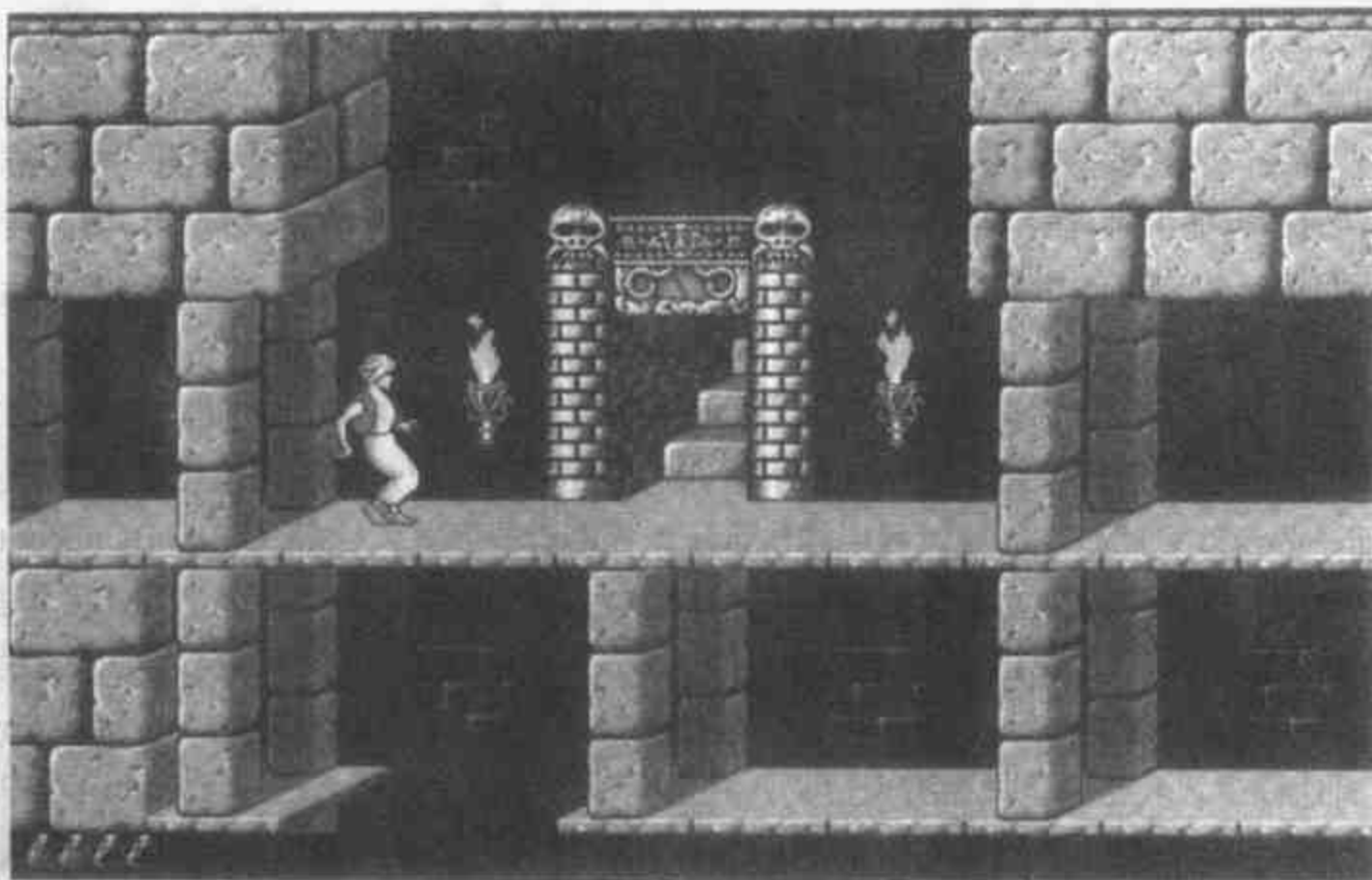


图 7.4 Prince of Persia 仔细地教会玩家如何预料像崩塌的地面和锋利的长钉这样的陷阱

在这些操作的介绍中，帮助玩家学习的环境必须是安全的。如果玩家总担心每一步是否会死掉，并且游戏总是不能原谅玩家的错误，那么玩家心灰意冷的机会就会迅速增加。Half-Life 这一点就做得很好，游戏提供给玩家一个安全而又有趣的环境让玩家熟悉操作，而不会很快威胁到玩家的安全。Prince of Persia 是另一个在介绍游戏方面做得特别好的游戏，它通过一些例子而不是惩罚来指导玩家。例如，当玩家开始遇到崩溃的地面时，掉下去是没有生命危险的。同样的，长钉也是通过这种方式介绍的，玩家很容易注意到并且能够存活下来。后面再次遭遇长钉将不会被原谅，但到那时玩家已经了解到它们对于游戏世界中角色的威胁，并且如果他够聪明就能活下来。

7.4.1 奖励

在这段学习期间，即使是最简单的成功也要奖励玩家，这一点是重要的。这时玩家感觉到她是在游戏正确的轨道上，并且激励他继续玩下去。玩家并不希望游戏太简单且没有挑战性，但因为他们刚开始游戏时犯的错而惩罚他们并不是产生挑战的正确方法。关键是先给玩家早期的成功，把他吸引到游戏中来，让他认为他对这个游戏了如指掌，认为他自身很强。“哈哈，这款游戏很简单，我能够控制它。”他也许这样说。然后当游戏突然变得具有挑战性的时候，玩家已经被吸引到游戏中，会认为挑战很合理，并相信他一定能克服。毕竟，游戏很简单，对吧？

近来，许多复杂的游戏通过一种真正游戏世界之外的教学关卡将玩家引导进入游戏世界。玩家可以通过主菜单上的选项进入这个教学的世界开始一种“真实”的游戏。这些教学关卡是一个好想法，比起用手册指导玩家确实是一种提高。这些教学关卡做了一件电脑最擅长的事，提供了一种互动的学习经验。这些教学关卡的一个问题就是因为它们玩起来没有什么乐趣，所以许多玩家都会略过它们直接进入实际的游戏。许多玩家普遍感觉教学关卡不是“真实”游





戏的一部分，而他们想尽快开始玩“真实”的游戏。如果设计者加入教学关卡是想让他的游戏在开始时就有难度，避免在游戏过程中指导玩家该如何玩，那么那些跳过教学关卡的玩家就会受阻。教学关卡对于那些想要先接受指导的玩家来说是不错的，但在游戏开始一定要简单。同样，*Half-Life* 也提供了指导玩家的教学关卡，但是它同实际游戏本身的开局相协同，玩起来很简单，而且有一个友好的学习曲线。当然，要使教学关卡尽可能的有趣，并能够激励玩家去玩它还有很长的路。

这些教学关卡经常包括一些指示来解释要获得某种效果玩家需按什么键。经常有声音附随着屏幕上的文档告诉玩家“按下空格键是发射你主要的武器”或是“按住蓝色的 X 键是超级跳跃”。而一些游戏竟然是在游戏过程中才告诉玩家控制键是哪些，比如 *Crash Bandicoot*。这些对于玩家该做什么来获取成功的详细解释使一个复杂的游戏变得容易上手。然而除了那些以外，像 *Spyro The Dragon* 和 *The Legend of Zelda: Ocarina of Time* 这样的游戏甚至让实际游戏中的人物告诉玩家游戏角色的控制。“Spyro，按住蓝色的按钮滑行”，年迈友好的龙在游戏 *Spyro The Dragon* 时这样说。我认为这样做有点过火，它完全破坏了玩家对不信任的悬念。游戏中的角色不知道玩家的任何信息，当然关于一个 PlayStation 的操纵者更是一无所知。但是，我认为当玩家正在游戏时，通过更多无关的图形界面显示和非游戏角色的语音提示，对于提醒玩家游戏的操作还是有帮助的。许多现代的游戏包含有如此复杂的操作以至于它们可能疏远了那些非专业级的玩家，而提醒那些新手要实现一个确定的移动需要做些什么，这是一个好主意。



图 7.5 诸如 *The Legend of Zelda: Ocarina of Time* 这款游戏中这样的控制标题能很好地指导玩家如何控制游戏

回顾过去，我可以这么说，我的所有游戏都太难，当然玩家也很难上手。在通过前面简单的关卡引导玩家进入游戏世界方面，*Damage Incorporated* 可能是做得最好的。相反，我的回合制角色扮演类游戏 *Odyssey* 就做得不好。游戏从一个岛上的船只失事开始，没有任何武器和任何种类的财产。我想让玩家很快受到惊吓，然后需要在附近的洞穴中找一个安全的藏身之处。



在玩家到达海岸后，我让一些怪兽开始就在玩家移动方向上阻挡几个回合，这样来达到以上的效果。玩家靠自己是没有办法打败这些怪兽的，他需要躲进附近的洞穴才能存活下来。起初，我让这个洞穴隐藏在森林中，使玩家很难找到，这样使得游戏对玩家的错误更加不可原谅。幸运的是，我的游戏测试员让我相信游戏的引导太难了，然后我将洞穴移到了空旷处以便玩家可以轻易看到。但是问题仍然存在，在玩家还没有熟悉这些操作之前，他就被陌生的怪兽攻击，而不知道该做些什么。我经常想知道有多少玩家被这种过度的挑战吓跑，结果永远不再玩剩下的游戏。



7.5 输入/输出

游戏的输入输出系统是游戏中两个主要的因素，这些因素既可以决定游戏的学习曲线的难易程度，也可以决定玩家是否觉得游戏玩起来很容易。使用你设计的输出输入系统，玩家应该能轻松地控制和理解游戏。设计这些系统是游戏设计中最难的部分之一，因为如果它们设计优秀，玩家甚至不会知道它们的存在。但是如果它们设计得很差劲，玩家容易受到挫败，同时也会抱怨游戏操作妨碍了他们在游戏中真正想做的动作。设计输入输出系统是“看不见的艺术”，因为创造它们的目的是为了让他们在玩家看来一目了然。有时这也许会导致设计者不能完全考虑到怎样使他们游戏中的输入输出系统工作得最出色，如果想要你的游戏玩起来有乐趣就要消除每一个错误。



7.5.1 控制和输入

作为一个玩家，知道其在游戏世界中扮演的角色应做什么，但实际操作却不允许他实际做出来，那么没有什么比这个更让人失望了。优秀的游戏操作不会想到控制本身；把谜题留在游戏世界中而不是留在控制计划中。游戏控制是真实世界与游戏世界的界面。为了让玩家在游戏中的经历显得真实，玩家必须按他操作真实世界的方式凭直觉操作游戏世界。每次玩家不得不想“现在我该按什么键”，这种投入状态就被破坏了。

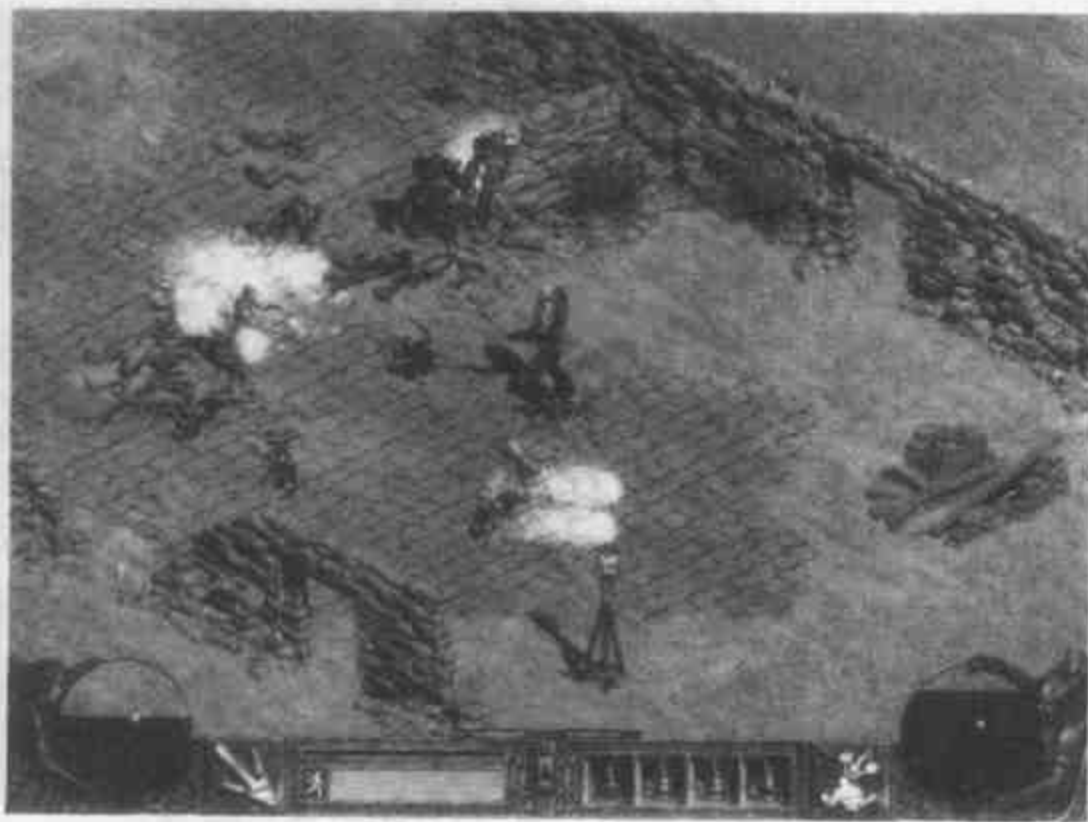


图 7.6 Diablo 系列极其简单的操作使它成为最容易学会的游戏之一

虽然许多电脑游戏的控制看起来越来越复杂，尤其是那些 3D 的动作类游戏，但是

那些 3D 的动作类游戏，但是





保持游戏控制的简洁还有许多要说。实际上许多像 Diablo、Command&Conquer 和 The Sims 这样的游戏，它们的成功都归功于一个因素，那就是玩家可以用一只手玩游戏，只用鼠标就可控制所有东西。如果使用得当，鼠标会是一个极有力的输入设备。其有力之处在于，它是绝大多数非玩家的电脑使用者都已经熟悉的设备。这使得只用鼠标操作的游戏很容易上手，因为它们减少了玩家学习游戏控制的时间。

设计一个基于鼠标的优秀界面很大一部分是制作一个系统，它看起来不应像 Windows 的文件管理系统那样毫无创意而且商业化，但又要保持使用上的简易性。制作具有吸引力的系统通常是一个构思良好的艺术，但使它具有吸引力又要不失去其直观性和功能有相当的难度。无论何时，当一个艺术家建议做一个这样的按钮时，设计者必须考虑这个新设计是否会远离玩家了解这个按钮功能的能力。通常，你可以从其他界面借用一些简洁易懂的图标，或从其他游戏中或从现实世界的设备中，比如 VCR 或 CD 播放器。比如，每个人都知道录音设备上的“快进”按钮，如果在你的游戏中使用恰当的话，玩家立刻就会知道这个按钮是做什么的。在游戏中制作易于掌握的按钮，并且让按钮看起来也很吸引人，其重要性与游戏的创造性和游戏测试工作同样重要。如果那些测试设计它们游戏的人说游戏的按钮不明显而且让人困惑，那么它们也许确实如此，你需要重新设计它们。

游戏设计中一个普遍的错误是想包容的东西太多。应用到游戏可玩性的各个方面，特别是在控制上，“越少就是越多”这样的陈词滥调有时还是对的。每次在游戏中加入一些按钮或是按键，你必须问问自己你刚刚增加的游戏控制的复杂度是否与它实现的功能相当。尤其当设计一个 PC 游戏时这种诱惑更大，因为键盘提供的按键比游戏需要使用的更多。不幸的是，一些游戏为了想要使用所有的按键，给每个按键都捆绑上了某种独特的功能。复杂的键盘操作虽然能得到内行玩家的赞赏，但也疏远了那些新手，可能会导致喜欢它的玩家人数剧减。由于街机（console，控制台）可提供按键的数量限制，控制面板允许设计者设置有更多的限制。不像许多其他的设计者，特别是那些从 PC 机转换到街机的设计者，我总感觉这种限制不错。控制面板强迫设计者定义他的控制，剔除所有无关的操作，把游戏世界中玩家所有能做的动作合成一些集中的操作。这使得学会玩游戏变得更容易。实际上，许多最流行的街机游戏甚至仅使用部分按钮。因为在他们的配置中有着巨大的键盘，PC 游戏的设计者不用被强迫将他们游戏的控制按同样的方式进行，而且我认为他们的游戏正是因为这一点受到损害。正如我上面提到的，一些最流行的 PC 游戏设法把他们所有的控制都放进了鼠标里。

游戏控制复杂度的增加大部分归功于实时 3D 游戏统治地位的提高。这些游戏因为想包容玩家游戏世界代理的一些能力，比如向前向后移动、向上向下、左边右边、转左转右、向上投掷向下投掷，在游戏中仅仅移动而不做其他的事情就已经使用了大量的控制。在许多方面，要想直观且简单地控制 3D 空间中完全自由的角色，完美的方法还有待探索。这就是迄今为止发行的 3D 游戏中，允许玩家完全自由控制角色的成功游戏非常少的原因。实际上，最成功的 3D 游戏，例如 Super Mario 64、Quake 或是 Tomb Raider 也限制了一些平面上的运动。

使游戏控制对各种玩家都有直观性的一项技术就是，包容完成某种效果的多种途径。例如，如果一个玩家看到实时战略游戏 StarCraft 使用的界面，可通过点击选择某对象，控制它，再点击对象操作的按钮，然后点击地图上他想要该单位实施动作的地点。玩家也可以点击这个对象来选择它，然后迅速在游戏世界中右击，这个单位会做出一个最合理的动作前往你点击的地点，





无论意味着移动到那里还是攻击那里的单位。此外，StarCraft 也允许玩家通过一个热键来完成这些动作而不用点击相应按钮。对于保持界面足以简单让新手掌握游戏有令人高兴的副作用，因为它全部都是点击，而内行的玩家可以花时间记住那些热键来提高他的游戏技巧。在许多街机动作类游戏中，控制板上不同的按钮实现的可能是相同的动作。尤其是在 PlayStation 游戏上有一种普遍的选择，就是允许玩家通过左边的控制板或是左边的操纵杆来控制玩家的行动。例如 Crash Bandicoot 允许玩家使用控制板或是操纵杆来移动，而且还允许玩家按下一个触发按钮或是控制器顶部的一个按钮来获得滑行的能力。提供给玩家完成游戏世界中单一行动的多种方法，有助于确保一个特定的玩家会喜欢这些方法中的一种。



图 7.7 StarCraft 提供给玩家一个第一流的界面，它允许玩家使用不同的技巧发出指令

在游戏设计中还有许多创意的空间，但是控制不是发挥创意最好的位置。游戏应该在游戏可玩性、故事情节和其他内容方面有新创意，但没有必要发挥在控制上。一些最成功的游戏提供玩家在其他游戏中所熟悉的操作规划，然后应用到新的具有强制性的内容中。有时这些早已建立的控制规划也很虚弱，但通常它还不至于虚弱到要重新建立一个全新的控制系统。作为设计者，你必须权衡通过一个超级的控制规划能获得什么和因为玩家的困惑会失去什么。例如，Sid Meier 的实时战略游戏 Gettysburg! 使用的是“点击然后插旗”的系统作为命令部队移动的默认方法，而不是使用当时其他游戏用的“点击然后再点击”的系统。他的系统更有创意，而且实际上它比起之前建立的系统能更好地控制游戏。然而，它还没有那么好，因为它超出了初次玩这个游戏的玩家所经历的困惑，这一点在第 2 章对他的采访中他也承认了。街机游戏特别善于提供统一的控制规划，有些游戏迷能够从这种类型的其他游戏中挑出任何一款立刻开始游戏，即使这种游戏他从未见过。

在游戏的开发过程中，当你一遍又一遍玩这个游戏的时候，会很容易就熟悉这种差劲的操作。虽然操作也许被设计得很差或是不直观，但是作为游戏设计人员，你应该反复的使用它们





使它们变成第二种。然而一旦有人第一次玩这款游戏，玩家会很快对里面的操作失去信心而且很有可能停止玩它。一个正确的游戏测试阶段包括许多第一次玩这款游戏的玩家，而且他们对操作最初的反应用于了解你的操作直观度实际有多少是很重要的。不要认为“玩家会习惯的”或是“傻瓜！这些控制这么明显，为什么会看不到呢？”，而应当考虑“我的控制差在哪里？我该怎样来调理它？”

设计玩家可以直观发现的控制是相当有挑战性的，特别在 PC 市场，不同的游戏有多种不同的控制方式。例如，很难断定什么才是第一人称射击类游戏的“标准”操作，以上三个成功的第一人称射击类游戏各自都有一套独特的控制方案。过去 20 年中几乎每一个发行的 PC 动作类游戏都允许玩家自己重新设置控制方案，对于那些需要玩家操作一大堆按钮的 PC 游戏来说这是完全必要的。但是，因为想立刻开始游戏的欲望或是缺乏对计算机知识的了解，许多玩家将不会找到或是使用这些控制配置方案。许多的玩家将还是使用那些默认的按键设置，这就是为什么设计者的工作要确保这些默认的设置尽可能适于玩家游戏的原因。永远不要因为仅仅是主管程序员喜欢或是开发小组已经熟悉，而在游戏里使用一种奇怪的或让人迷惑的默认控制方案。始终确信默认的游戏控制应尽可能的直观。

特别在动作类游戏中，当你的控制方案设计得完美时，把玩家从游戏世界隔开的那堵墙就消失了，玩家会开始感觉他就是游戏世界中的角色。这是一个能让人投入游戏的最终征兆，没有强有力的控制是不可能达到这种效果的。在那些能让人沉浸在其中的游戏里，控制对玩家而言必须是完全不可视的。这对设计者也是一个难题。为什么这么辛苦做出的东西如果实现得完美还要完全的不可视呢？为了使游戏让人更满意，设计者必须认识到正是控制的透明度让玩家喜欢游戏提供的剩余部分。

7.5.2 输出和游戏世界的反馈

玩家直观地控制游戏的能力也许就是一个成功游戏的关键，游戏世界中的信息对玩家也一样很重要。电脑游戏包含了无数的复杂系统，一般地它们都在进行计算而不为玩家所知。实际上，这就是电脑游戏的长处。把这些数量巨大的数据浓缩成最具有代表性的形式，然后把这些信息传送给玩家是设计一个优秀输出系统的关键。

拿一个策略游戏来考虑，玩家有许多单位分散在一张大地图上。地图如此之大以至于只有很小的一部分能显示在屏幕上。如果玩家在屏幕外的一组单位受到攻击但是玩家又不知道，那么他一定会恼火。拿角色扮演类的游戏来考虑，玩家团队中的每个成员都需要定期地喂食物，但是游戏不提供可以表明角色饥饿程度的信息。那么，如果其中的一位成员突然因为饥饿死亡，玩家同样也会很失望。为什么玩家必须知道这些重要的游戏信息呢？在一个动作类的游戏中，如果玩家不得不射击敌人身体的某个特殊部位来杀死他，比如说眼睛，当他成功做到的时候玩家就需要有一个明确的信息反馈给他。一旦受到强力攻击，敌人也许是在痛苦中翻滚或是痛苦的大叫。如果玩家没有收到这样的反馈信息，那么他怎么知道还在游戏的正常轨迹中呢？当然，所有的电脑游戏都隐藏了一些信息，游戏不可能将游戏世界中所有的信息都传送给玩家。但是它们必须传送一些玩家角色应该知道的信息，而且要传输得有效率。

几乎所有的游戏都在输出系统的中心位置把游戏世界展现给玩家。通过这种展现，玩家可





以看见他正在控制的对象及其位置和状态。你的游戏要通过这种方式传送尽可能多的信息给玩家。考虑一个第三人称的3D动作类游戏。玩家当然可以看到游戏世界的环境和位置，但是关于玩家角色的情况呢？也许他的健康下降了，玩家角色就不能正常行走而是跛脚或是蹒跚而行。同样的，玩家护甲的力量也可以通过角色身上所穿护甲的颜色变化来表现，当护甲受到伤害并且快要毁坏时，它就会显得越来越难看。玩家正在使用的武器可以通过看到角色身上所佩戴的武器来表现。如果玩家在角色身上释放一个保护魔法，也许角色身上就会发射出一种光来简单的告诉玩家角色的状态。设计者也许会把这种信息包含在某种仪表盘中，通过游戏的主要显示区域将信息传送给玩家，使玩家理解起来更简单透明。

许多游戏将游戏显示区域所不能表示出来的东西包含在某种图形用户界面中，它一般都在游戏显示区域的边上或是像仪表盘一样覆盖在顶部。这种图形用户界面可以很简单，比如在 *Centipede* 中就只显示游戏得分和剩下的命数，在 *Prince of Persia* 中屏幕底部会显示很小的生命点数，或者像任意一款 Infocom 游戏一样显示得分和移动。游戏越复杂，图形用户界面通常也会越复杂，比如在 Maxis 的 Sim 系列游戏中使用的按钮条，或是老版本的 *System Shock* 中显示的大量状态，或是许多像 *Bard's Tale* 这样的角色扮演类游戏在游戏中提供大量团队人员的数据。许多早期游戏中的图形用户界面只是为了把大块的屏幕分割成小块。这么做不是某种设计的决定，而是因为游戏引擎还不足以快到能将游戏全屏显示。随着引擎技术的发展，游戏正试图让游戏显示区域占据绝大部分屏幕，而尽可能的缩小图形用户界面。



图 7.8 *Oddworld: Abe's Oddysee* 完全摒弃了内置在游戏中的图形用户界面，而是给了玩家一个自由的游戏显示区域

只有极少数游戏没有任何的图形用户界面。特别是 *Oddworld: Abe's Oddysee*。游戏的主管 Lorne Lanning 强烈地感受到任何图形用户界面都会疏远玩家与游戏世界的距离。因此，Abe 的健康状况是通过他行动的方式传达给玩家的。因为游戏让玩家总是拥有无数条命，所以没有必要显示剩余的生命，而许多街机游戏的图形用户界面现在也只包括这仅仅一种元素。例如，Crash





Bandicoot 默认为一个完全自由的游戏显示区域，如果玩家点击一个按钮显示图形用户界面，那么在上面只会显示出还有多少条命。当然，随着技术允许游戏显示占据整个屏幕，取消屏幕上仪表盘的倾向越来越大。最小化图形用户界面在吸引玩家上的优势是显而易见的，如果游戏世界能够有效地传送所有玩家需要的信息，有时就根本没有使用图形用户界面的理由。

设计图形用户界面最重要的一点就是，尽可能让它形象化。特别是在快速的动作类游戏中，设计的图形用户界面要尽可能迅速的将玩家现在的健康状况、可用弹药或是附近的怪物等信息传送给玩家（附近的怪物可以通过某种雷达来传送）。如果有什么不同的话，图形用户界面在控制电脑的主要模式上占有统治地位，开始是通过 Macintosh 达到的，然后就是 Windows，这表明大多数人都是直观地思考而不是通过数字或是词句。因此，游戏中设计优秀的图形控制盘比一大堆数字或是单词更容易让玩家看到并且理解。这说明健康条比健康数字或百分比更优越。美术设计们更喜欢使用图形控制盘，因为它比一堆枯燥无味的数字更吸引人。

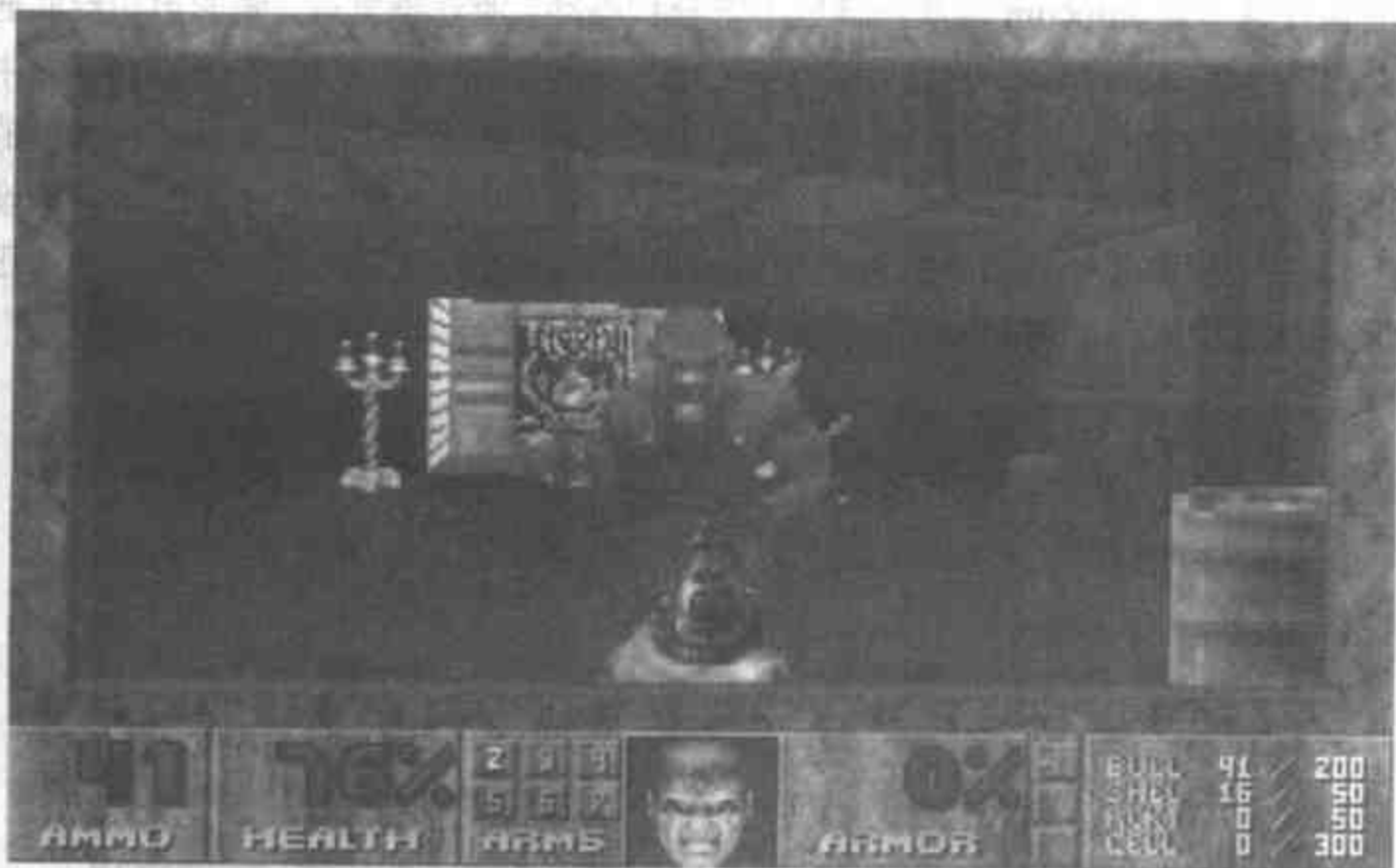


图 7.9 Doom 中屏幕底部的头像是一个设计得很好的界面元素，因为它可以将玩家现在的健康状况直观地传达给玩家

一个设计特别好的游戏元素就是在 Doom 和 Quake 使用的“头像”。这个显示在屏幕底部中间的面容完全直观的表现出玩家大概的健康状况。这张脸开始是健康而且在咆哮着，准备迎接挑战。随着玩家游戏的进行，当他损失了一些血时，头像看起来就有瘀伤或在流血。最后当玩家没有血时，它看起来几乎像死了。游戏中任何时候玩家匆匆向下看一眼他的头像就可以立刻知道他还剩多少血，如果换用数字来表示，玩家如果只扫视一眼很难知道他的健康状况。时间上的差别也许只有几毫秒，但在一个快速的动作类游戏中，那可能就是生与死的差别。

当然，如果数据直观的表达太愚蠢以至于玩家不能轻易的理解，那么这种表达同样也会有负面效果。比如，在 WarCraft 中，一个单位能实现的不同动作的按钮全部都是用图标来表现的，而我也鼓励这么做。然而，其中一些按钮最初不易弄明白。幸运的是，如果玩家把鼠标的指针在这些按钮上停顿一下，游戏就会在屏幕的底部显示消息，告诉玩家如果点击这个按钮会做什么。



么。如果这些按钮上的图标能够再明显一点就更好了。不可否认的是，要想通过 32×32 大小的图标来表示一个诸如“守卫”这样现实世界中的动作有相当的难度。游戏的图形用户界面需要在直观表现的优势同文档的清楚透明之间做出权衡，也许需要的是二者的结合。

声音输出也是在游戏中经常使用的输出设备。并不是所有关于游戏世界的信息都需要通过 stimuli 传送给玩家。比如，在 The Sims 这个游戏中，玩家可以根据成员声音的语调来判断他的角色是否喜欢交谈。在 Command & Conquer 中，玩家通过一个对象提供的声音提示“I'll get right on it!”知道这个单位已经接受了命令。同样的，当屏幕外的对象受到攻击时，游戏也会通过说“Unit attacked”或“Unit lost”来告知玩家。声音提示可以给屏幕上的信息提供极好的辅助，是有效传送重要信息的惟一方式。

游戏优秀的输出系统既要强大又要直观。它允许玩家直接进入游戏并且知道游戏世界中正在发生什么，但它也可以有效的提供给那些游戏老手所需要的所有信息。随着时间的推移，游戏传送给玩家的数据会变得更加明晰。玩家不必考虑了解这个世界，他们只要看一眼屏幕就“知道”需要做什么，并且通过直觉和相应的控制迅速做出反应。正如我前面陈述的观点，开发输入输出系统的过程中不需要太多的创造性。从其他游戏中得来的这些优秀范例通常是因为一个原因而有优势：它们有用。“好的艺术家取形，伟大的艺术家取神”这句话在游戏的 I/O 设计中尤为正确。



7.6 基本的元素

在本章节里我讨论了一些优秀游戏的元素：独特的解决方案、非线性、虚拟现实、指导玩家，以及输入输出系统。我觉得，当你开始开发一个游戏时，其中的每一个部分都值得认真思考。当然，这远不是一个完整的清单，当你成为一个游戏设计者时，你将会积累起你自己的元素清单，它包括了你认为对优秀的游戏有帮助的元素。没有人能够说出优秀的游戏设计所包含的所有元素。每个游戏设计者都必须自己来判断。这种个人的偏好使每个游戏都带有其作者独特的标记，并且使这些最好的游戏具有个性。



Chapter 8

第 8 章 游戏分析：Tetris

Alexey Pajitnov 设计



很少有游戏能像 Tetris 这样深受游戏开发者们的好评。通常，当一个游戏变得像 Tetris 一样流行时，那么几乎所有的系统中都会出现相应的版本，当销售量达到数百万时，游戏专家们就会开始抱怨这个游戏有多么差劲。Myst 就是个很好的例子。当它发布的时候，被游戏评论界一致誉为是个趣味盎然的冒险游戏，并且界面设计得非常漂亮。它的开发者们虽然没有这么沾沾自喜，但也认为这个游戏相当好。在随后的数年中，当它以几百万份的销量位居销售榜榜首时，原来赞赏这个游戏的部分人开始憎恨它，但仍未阻止其长盛不衰的势头。而在游戏的一片贬低声中，叫得最响的就是这个游戏的开发者们。是这个游戏现在变差了吗？当然不是。那么是因为游戏专家、游戏评论者和游戏开发者一样，厌恶它的良好销售情况吗？看来是这样。

但 Tetris 的情况却并非如此。Tetris 曾风行全世界，但我们还很难听到关于这个游戏的负面



评价。是什么使 Tetris 幸免于批评呢? 看起来是这个游戏设计版本的朴实无华, 简洁流畅, 使得最苛刻的游戏开发者也不得不承认它的伟大。和 Myst 的情况正相反, 当它刚发布的时候, 有关它的评论主要局限于其俄罗斯的老家, 而且看起来波澜不惊, 至少是不怎么令人激动, 都是些关于玩法之类的东西。这个游戏是这么的简单平朴, 它使用的技术毫无花哨之处, 看来评论界好像也没有办法把它写得激动人心——至少开始是如此。现在, 这个游戏已经毫无争议地成为经典之作, 任何一个游戏评论家都会很高兴告诉你, 他曾沉迷于这个游戏, 乐而忘返。

Tetris 的玩法非常简单易学。游戏界面是一个很高的长方形 2D 盒子, 积木出现在盒子的上方, 这些积木都由四个小方块组成, 并按照各种方式排列, 其中每一个方块都至少和另一个方块相连。然后, 这些积木慢慢地落向盒子的底部, 玩家可以把这些方块左右移动或旋转。一旦这个积木碰到阻碍, 比如盒底或者另一个积木, 它就会停止运动, 而玩家也就失去了对这个积木的控制, 这时会有一个新的积木出现在屏幕的顶端, 玩家又可以对之加以操纵了。当屏幕底端的方块拼满一横行时, 这一层的方块就会消失, 而这一行以上的所有方块都会下降一层。而当那些没有拼完整的横行填满整个长方形盒子, 新的积木无法再出现时, 游戏就结束了。



8.1 拼图游戏还是动作游戏

Tetris 经常被称为拼图游戏, 这有一定的道理。很显然, 它包含了一些拼图游戏的要素, 比如说, 玩家需要找出如何能把各块积木最好地拼在一起的方法。从这个意义上来说, Tetris 相当于直角的智力拼图游戏, 或其他任何一种“在这有限空间内组织这些几何形体”的拼图。其实, 它更像五角拼图, 据说 Tetris 的设计者 Alexey Pajitnov 就是从五角拼图游戏那里获得灵感的。玩

五角拼图要把 12 块不同的碎片拼放在一个矩形的盒子里, 其中每个碎片都是由五个小方块组成的。这两者初看起来很相似, 但 Tetris 已经把这个游戏变得截然不同了, 它更具挑战性也更有吸引力。Pajitnov 没有像其他的开发者对 Jigsaw Puzzle 或 Sliding Number 类型的拼图所做的那样, 简单地把五角拼图游戏直接搬到计算机上。虽然那样做会因为不能去触摸那些小块而损失一些乐趣, 但



图 8.1 Tetris 小心地在动作元素和拼图元素间取得平衡, 以创造独有的游戏体验。这是本章的所有插图 The Next Tetris 中的经典模式





它仍不失为一个很好玩的程式。但 Pajitnov 把它改造成了一个只有在电脑上才能玩的游戏，由此创造了一个独一无二的新游戏，比原来的五角拼图更具娱乐性。

很多情况下，当游戏界的精英们把一个游戏说成是拼图游戏时，都带有轻蔑的意味。对于他们来说，拼图游戏只是把一些静态的拼图交给游戏玩家，这些拼图永远不会改变玩家的行动，也永远不会对玩家的行动做出反应。而他们认为一个游戏应当对玩家的行动做出相应的反应，给玩家提供一个与之对抗的对手。所以，评论家们得出结论说，这些所谓的拼图游戏根本就不是游戏，它们只不过是一些拼图罢了。更重要的是，这些拼图游戏都只有一种获胜方法，这就进一步限制了玩家的互动经验。这样的例子有很多，比如说 Zork、Myst，甚至是 Grim Fandango。这些游戏确实给玩家提供了一个可以去探索的世界和富有挑战性的拼图，但它们无法创造一次玩家独有的经验。

Tetris 却从来没有因为这些问题受到过批评。为了创造真正引人入胜的游戏，它巧妙地平衡了拼图游戏与动作游戏的机制。因而每个人每一次玩这个游戏，都会有一次不同的体验。Tetris 所用到的一种动作游戏机制是越来越逼近的威胁感，留给玩家的时间总是非常有限。在 Centipede 中威胁是 Anthropod 从屏幕的上方蜿蜒而下，而在 Tetris 中则是积木从上面落下来。如果玩家不在积木落到屏幕底下前移动或翻转它，不能给它找一个最有利的位罝，这块积木就可能会堵在某处，而它下面的几行也就没有办法拼接完整，玩家也就离游戏结束又靠近了一行或几行。在游戏进程中，积木落下来的速度越来越快，这样对玩家的挑战性也就越来越大，游戏的难度也逐步增加。

Tetris 与动作游戏的另一相似之处使得它在拼图游戏中更显得与众不同，那就是 Tetris 可以创造不同的游戏场景：玩家所玩的每一次都是独一无二的。这种游戏机制建立了无穷无尽的不同游戏，玩家的每一步都影响了他后面的游戏。一块积木被摆放的位置直接影响到了下一块积木该如何被摆放。在一个四个方块长的缺口里，玩家应该把一个倒置的“L”形积木插进去吗？还是他应该一直坚持，并绝望地等待所需的四个方块的长条出现？“L”形积木不能完全填上这个缺口，可是天知道什么时候才会来“I”形块？还有一些情况玩家可以有好几种不同位置用来放一块积木，但他必须做出长远打算，推想出如果他这么放现在这块积木，他给以后的几块留下了什么样的空位。玩家可以事先构想出几块积木的摆法，但他不可能对他所将面对的挑战做全面的准备，因为每一次游戏都是完全不同的。



8.2 作为经典街机游戏的 Tetris

实际上，不少人宣称 Tetris 是我所说的“经典街机游戏”形式的一个例子。尽管事实上它并不是为街机设计的（虽然它的广泛流行导致它最终出现在那里），而且几年后才在街机中出现该游戏。看一看第 4 章在 Centipede 分析中所列出的经典街机游戏清单，我们就会发现 Tetris 是多么的符合这种形式的基本原则。

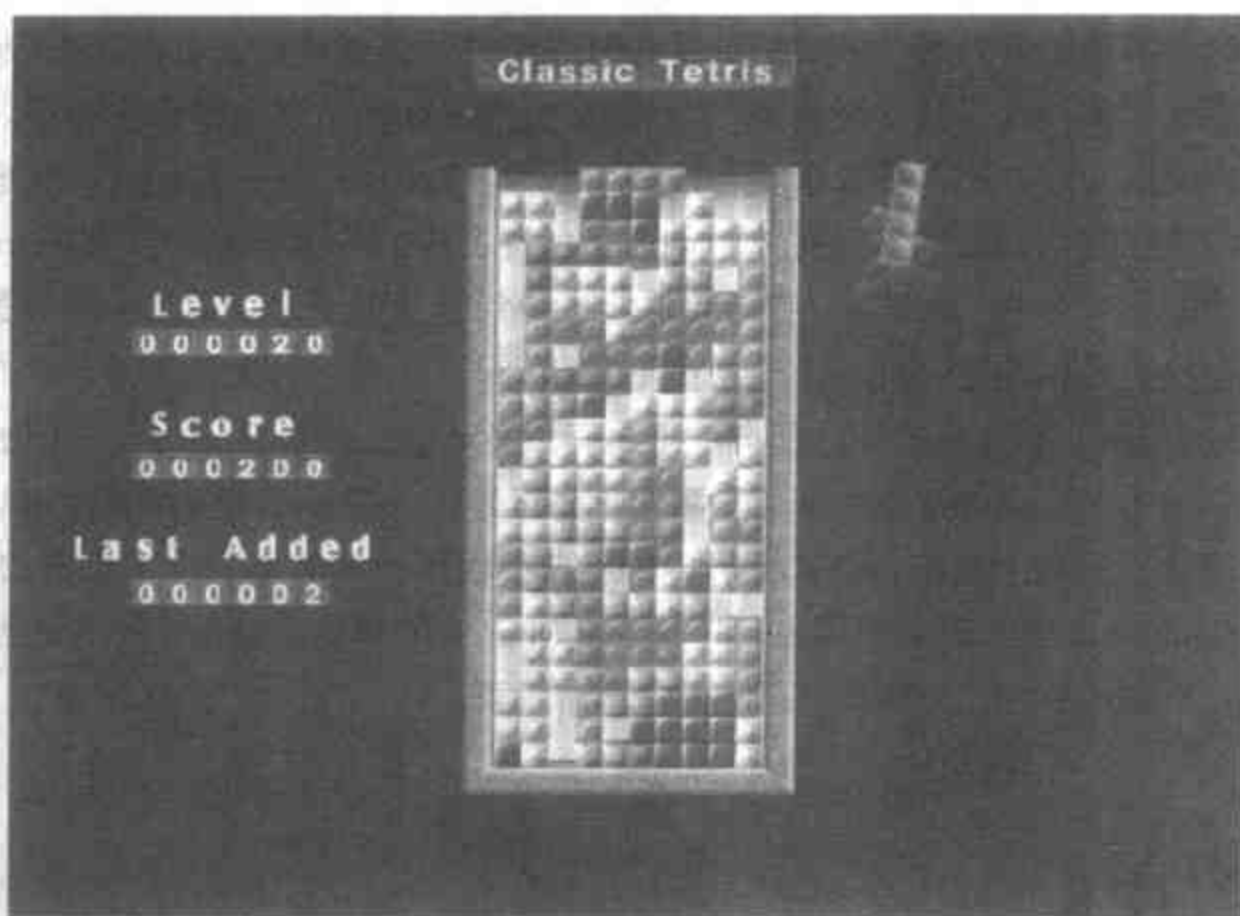


图 8.2 尽管是在经典街机游戏已经过时后才开发出来, Tetris 的游戏操控却体现了这种游戏类型的设计原则

- 单屏游戏: 当然, Tetris 全部反映在一个屏幕上。玩家可以在同一时间看到整个游戏环境, 在此基础上决定怎样处理手上的这块积木。这个游戏当中没有任何探险成分在内, 它不会让玩家真正大吃一惊(除了不知下一块积木是什么), 也就是说玩家看得到让游戏顺利进行下去的所有信息, 所以他输了的话只能怪他自己。
- 无限游戏: 只要积木还没有堆到盒顶, Tetris 就允许玩家一直玩下去, 不管他做出了多少错误决定。每次游戏都以失败而告终, 没有人真正能说他“打败”了这个游戏。玩家们总是能找到提高打 Tetris 技艺的方法, 这就是 Tetris 和传统拼图游戏的根本不同。一旦玩家拼好了一幅拼图, 只要他还记得他是怎么拼的, 那么这个拼图就对他再无挑战性了。所以人们通常不会多次玩同一个拼图, 但一个设计精妙的游戏却总能让人玩上一遍又一遍。Tetris 正是这样一个游戏。
- 多条生命: 和大多数的街机游戏不同, 最早的 Tetris 只给玩家一条命。积木一堆到盒子顶部, 玩家就输了。但是游戏的设计允许玩家出现错误, 同时还不会立即败下阵来。当积木叠在长方盒子的底部时, 玩家可以看到自己所犯的 error, 在游戏结束前, 他还是有时间想一想如何把后面的积木拼放得更合理些。这样, Tetris 虽然没有给玩家多条生命, 但却在他重新开始前, 给了他一个改善游戏技巧、获得一些小成就的机会。
- 积分/最高分: 从 Asteroids 或 Galaga 等游戏所用的系统中, Tetris 直接引入了一个计分模式, 它给玩家计分并把分数记录在高分排行榜上。实际上, 因为你不可能击败这个游戏, 所以, 取得一个更高分数的可能性就成为玩家反复玩这个游戏的真正动力所在。
- 简单易学, 容易操控: Tetris 的真正杰出之处在于它的游戏机制简单平朴, 浅显易懂。玩好这个游戏只需要 3 个按键, 所有操作都在屏幕上一览无余。这实质上意味着, 不论是否熟悉计算机游戏, 任何人都可以走过来并立即开始玩它。然而, 玩家也永远不可能完





全掌握这个游戏，因为这个游戏的难度与时递增并且它在理论上可以玩无限长的时间。

- 没有故事：即使和大多数的经典街机游戏相比，Tetris 的故事情节也少得很。所以，它经常被用来说明游戏并不需要故事情节以吸引玩家。Tetris 只有一种场景，就是它的俄罗斯起源地，它在这个游戏的不同变体中创造了各种各样的美学效果。当 Spectrum Holobyte 发行游戏的第一个 PC 版本时，在游戏的背景中包含了俄罗斯生活的不同场景，音乐也以传统的斯拉夫风格为主。但人们一经发现 Tetris 是一个多么伟大的游戏，它的那些后续应用，就像 Nintendo Gameboy 的那样，不再有俄罗斯的主题甚至不再有任何故事场景了。缺少故事情节对这个游戏毫发无损。事实上，正是因为 Tetris 完全没有故事场景，使之与那些力图把自己建立在某种奇幻世界中的经典街机游戏截然不同，不管是 Galaga 的外层空间，Centipede 中花园里的昆虫，还是 Pac-Man 中的那些追着一个小黄人到处乱跑的可怕鬼怪。Tetris 没有这些借口，这使得它卓尔不群。



8.3 技术

Tetris 同经典街机游戏的另一个相似之处是，它们都不依靠技术来打动玩家。对于那些 CAG 们来说，80 年代早期所能做出的街机的画面，比起玩家在电影电视等其他媒介所能找到的画面要黯然失色，所以游戏必须从其他方面来吸引玩家。结果就是，如果一个游戏想生存下去就一定要真正引人入胜。尽管到 80 年代后期 Tetris 在西方发布的时候已经可以做出更加精美的画面了，但这个游戏依然坚持简单的 2D 技术的应用，而不需要更绚丽的画面。不管它的技术应用多么简单，Tetris 的游戏本身太强大了，它仍然具有广泛的娱乐性。

Tetris 的技术应用是这么的简单，以至于很多天才的游戏编程者开始克隆它。实际上有无数游戏公司试图改进它的画面效果，其中包括把它演变为 3D 游戏。最早的尝试可能是由 Pajitnov 设计的 Welltris，它是 Tetris 系列游戏中的一个。在 Welltris 中，一个 3D 的方井取代了 Tetris 中的盒子。Tetris 风格的积木（虽然不一定都由四个方块组成）沿着井的边缘向下落，你必须把它们在井底码齐。这个游戏变得相当的复杂，却没有因此增加多少趣味性和挑战性。结果，玩家失去了对它的兴趣，又回到传统的简洁中去了。很多 Tetris 的升级版本企图对原版加以“改进”，或者通过制造更好的视听效果，或者使用各种各样的特殊积木。这些企图中没有一个是特别成功的，玩家总是想玩回最初的游戏。

给 Tetris 增加技术含量的尝试不只在商业上没有取得成功，在艺术上也失败了。这些升级产品中所使用的先进技术其实有损于最初的游戏设计，使这个游戏失去了它的简单朴实，更使游戏过程丧失了它原有的优雅和美妙。当然了，我们从这个故事中得到的教益就是：先进的科技不见得对所有的游戏都是有益的，一旦花哨的引擎效果开始妨碍到游戏最初的娱乐性时，游戏设计者就要当心了。

虽然 Tetris 可能并不需要多高级的计算机技术来实现其功能，但也应当指出，如果没有计算机就不可能有 Tetris。Tetris 并不是简单地把一个纸笔游戏或硬纸板游戏直接搬到屏幕上，它只能存在于一个由计算机精心控制并管理的环境中。像我们以前提到的那样，Pajitnov 据说是从非计算机拼图游戏五角拼图中得到灵感的。在把它改成电脑游戏的过程中，Pajitnov 赋予了它一种



新的形式, 这种形式只能在计算机上生存。屏幕上方一块块积木的缓缓降落、它们同屏幕底下那些积木结合的方式以及玩家所遇到的积木的随机选择, 所有这些操作都只能由计算机程序才能完成, 同时它还要允许玩家从中获得乐趣。这些都是计算机所擅长的任务, 而 Pajitnov 能想到把这些加进这个游戏中, 真是天才。



8.4 人工智能



图 8.3 Tetris 应用了非常有限的人工智能, 即随机选出将要落下来的积木。这种 AI 虽然简单, 却能给玩家带来真正的挑战

这个游戏包含的所有 AI 内容就是随机数字生成器, 由它挑出进入游戏区域的下一块积木。然而, 游戏机制使得这个随机数字足以完全地改变每个游戏, 每一块积木的落下都给玩家带来了新的挑战。这种随机性保证玩家永远不会知道下一块积木是什么, 所以他只能按照不管下一块要来的是哪种积木都最有利的方式处理积木 (Tetris 的许多版本中都设有“下一块”影像, 告诉玩家下一块将要出现的积木是什么, 这确实使这个游戏变得容易了一些。但即使如此, 玩家仍然不知下一块的下一块将是什么, 因此他们在决定把目前这块放在哪里时, 同样只能靠一种指导下的猜测)。如果游戏是对抗性的, 也就是说一个玩家必须面对一个对手的挑战, 而且是单人游戏即对手是计算机, 那么从 Tetris 由一个随机数据生成器制造了所有挑战的这个事实, 我们可以得出重要的结论。玩家所面对的 AI 不需要太高级, 只要能满足游戏机制的需要就足够了。AI 要让玩家置身于一种富有挑战的环境中, 至于它如何实现这一点则无关紧要。它可能像 Civilization 这类高难度的战败游戏中的 AI 一样复杂, 也可能像 Tetris 中我们所发现的那个随机积木选择器那样简单。重要的是 AI 机制要同游戏机制相称才能给玩家提供足够的挑战。





下一块出现在屏幕顶端的积木的随机本质可能会给读者一种暗示：玩 Tetris 的成功可能全靠运气。如果玩家得到的积木是随机选择的，那么不同玩家的分数如何能互相比呢？要点在于，从长期来讲，得到各种积木的机率是相同的。这就好比在硬纸板的拼图游戏中，整个游戏中各种形状的碎片总是相同的一样，在 Tetris 中玩家所遇到的那些随机的积木最终来说并不是随机的。因为总共只有七种类型的积木，而每一种都由四个方块构成，而且玩家（即使是最初级的）有一个很大的空间来处理它们，所以这种随机性在让游戏不可预知的同时又使不同玩家的游戏具有了可比性。在游戏的整个过程中，玩家会得到几百块积木。玩家得到她正需要的积木的次数，大体相当于她没得到想要的积木的次数。有可能玩家因为没有在合适的时间得到合适的积木，而玩家的盒子里又已经堆满了方块，结果游戏就此终结了。但如果出现了这种她再也没地方放一块积木的情况，这个玩家一定是已经犯下了很多错误，所以才使自己的处境如此危险。总而言之，Tetris 中的随机积木选择器给所有的玩家提供了公平且一致的挑战。



8.5 紧张程度升级

Tetris 在玩家的整个游戏过程中毫不留情地逐级增加紧张程度。和 Centipede 那样的游戏不同，玩家在一轮结束时得不到休息，也不能在他丢了一条命后“重头再来”。在 Tetris 中，当盒子里堆满了积木，玩家又不能拼好一行时，他的游戏就结束了。这就要求他一直都要十分小心，并不停地思考如何处理正在落下的积木。即使是 Doom 那样的快节奏游戏都让玩家有大量的行动间歇。在这个第一人称射击游戏中有安全的角落可以藏身，还有那些房间，一旦里面的危险都清除了，玩家就可以在那里无限期地等下去，而不会受到任何威胁。Tetris 则没有一刻放松，玩家总要不不停地面对新的挑战。

玩家在 Tetris 中所很到的仅有的喘息就是当她从一个很局促的局面“打回去”时。就是说玩家把一些积木放错了位置，因此把几层没拼好的方块堵在了下面。这样在积木到达底部停下来前，玩家所能支配的时间和空间都变少了，结果游戏因而变得很艰难。作为玩家所犯错误的结果，游戏的紧张程度增强了。现在玩家可以通过谨慎地处理后面的几块积木，消掉一些没摆好的方块，并最终拼完下面的那几层。这样游戏的紧张程度降低，玩家又回到了以前的局面，有更多的空间和时间来处理落下的积木。玩家如负重释的同时也会得到一种成就感。她可以暂时放松一下，因为她有了块“白板”，可以从新再来。当然，这一切只持续到她犯下另一个错误，然后游戏的紧张程度再一次增加。

游戏进程中还会通过加快积木落下来的速度加快来进一步增加紧张度。当玩家的积分超过某一特定值时，积木开始以更快的速率移动，这就把玩家的神经进一步绷紧了。因为积木块落下来得更快，玩家只能用更短的时间推想积木应放置的最佳位置，同时用来摆放积木到那个位置的时间也更少了。当游戏达到最高速度时，大多数玩家都无法把一块积木放到理想位置上，而当一块积木放错位置后，游戏只会变得更难。就在速度加快前，玩家可能刚刚觉得自己掌握了 Tetris，可以把这个游戏无休止地玩下去。但随着积木下落速度的增加，玩家以前找到的节奏被打破了。现在，玩家要做以前所做的一切，只得用更快的速度。

在 Tetris 中玩家一旦开始犯错误，这些错误就会相互作用，使游戏变得越来越难玩。如果玩





家在屏幕底部留下一些没拼好的行，那他可用来操纵积木的空间就会越来越少。当玩家不小心让一块积木掉到了错误的位置，它下面的那些行就没办法被拼满了，这又令以后的积木更难以放在这块积木周围。当一个玩家努力支撑，等候一个“I”形的积木来填上一个长条形缺口时，他只能不断地把其他的积木放到那些很可能并不是最好的位置上，一直到他要的“I”形块终于在某个无法预知的时间降临。**Tetris** 就是通过这些方法来惩罚玩家所犯的**错误**。**Tetris** 不像其他计算机游戏那样给玩家机会让她追上来，它所做的只是惩罚，而且还让先前所犯的**错误**更难以挽回。当玩家用一个“I”形积木一下子同时消去四行时，她会得到奖励分数，而这使得情况更复杂了。通过这个策略，游戏引诱玩家去冒可能会使游戏结束的风险。



8.6 简单与对称

我们谈到过，**Tetris** 是一个非常简单的游戏。它的成功很大一部分要归功于它在那么富于挑战的同时又那么简单明了，容易掌握。要玩这个游戏，玩家不必去学习任何特殊的招数。这个游戏只用到了很少的几个按键，同时这些按键的操纵结果直接显现在屏幕上。研究一下 **Tetris** 中用到的积木也非常有趣。它们都是由四个小方块组成的，而且，游戏中所用到的七种不同类型的积木事实上代表了四个小方块相互联接的所有组合方式。由于玩家可以朝每个想要的方向旋转，所以这些方块实际上只有这七种不同的可能组合方式。

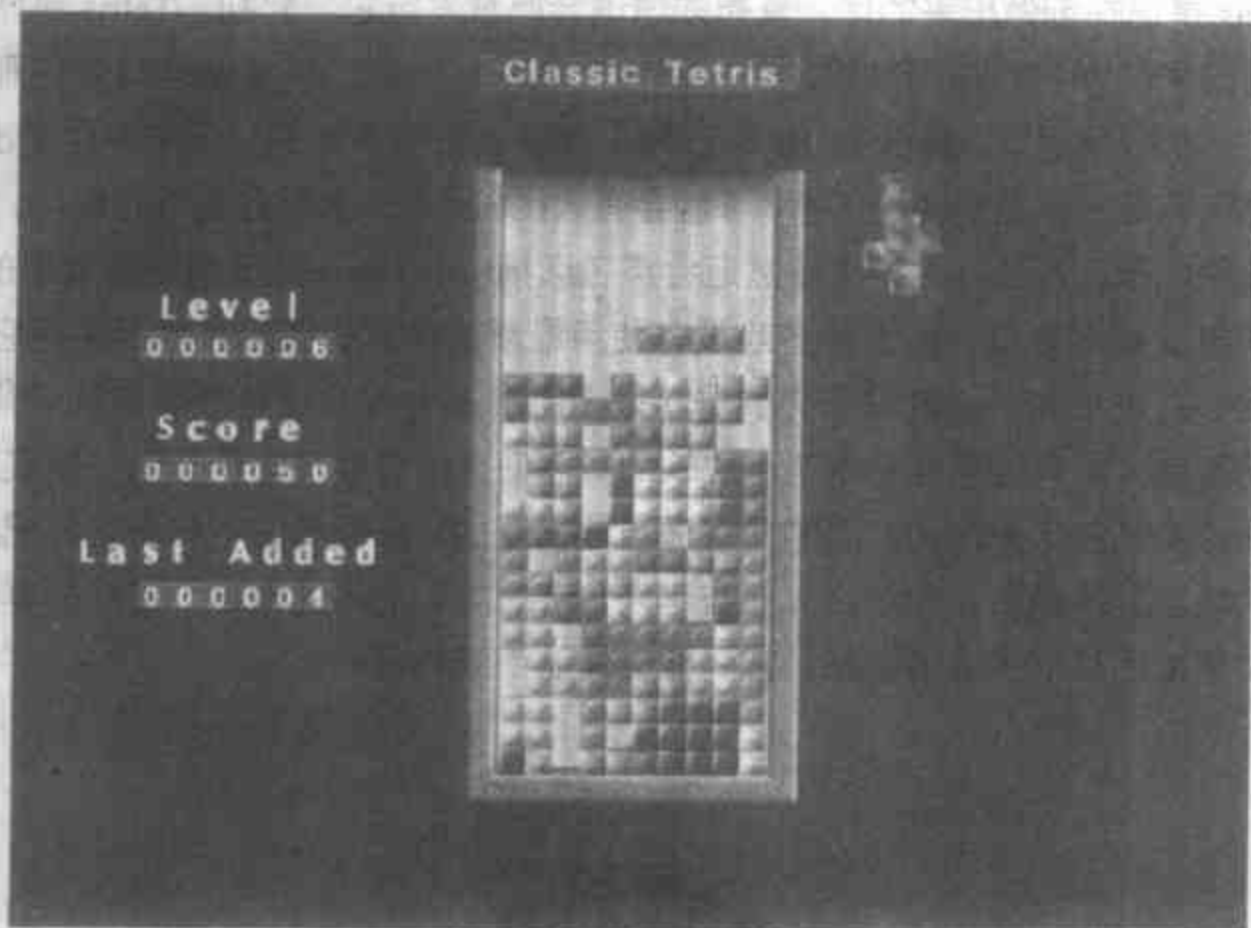


图 8.4 **Tetris** 中所有的积木都是由四个小方块组成的，每个小方块都至少和另一个相连，这使游戏具有一种内在的稳定性与平衡感

听说 Pajitnov 在创作 **Tetris** 的时候，起初曾经考虑过用五个小方块组合成十二种不同的积木。实际上，他从中获得灵感的五角拼图游戏用的就是十二种不同的五个方块积木。但 Pajitnov 很快就意识到，在 **Tetris** 这样高度紧张的环境下，玩家只有有限的时间来找到适合摆放每一块积木的





最好位置，十二种不同积木未免太多了。当然一个用五个方块积木的游戏会有它自己的挑战性，或许更慢的降落速度和更大的游戏区域可能会补偿大积木块带来的复杂性。但它还会是 Tetris 吗？不会！它还会像 Tetris 那么趣味盎然引人入胜吗？可能不会。游戏复杂到一定程度就会掩盖它的核心本质，把玩家们搞糊涂而不是挑战他们。用五个方块而不是四个方块会毁掉 Pajitnov 一直追寻的简洁性，所以最后他减少了一块积木所包含的小方块个数。

其实用五个小方块组成积木，其中每个小方块都同另一个相连，一共有十三种不同的组合。所以看来原来的五角拼图游戏并没有全部用上所有形状的积木，它只用了十二种。我从来没有试玩过五角拼图，因此我不知道那种拼图游戏玩起来是否有趣。但在一定程度上，正是积木类型的完整性使 Tetris 显得如此的优雅。游戏中使用了四个小方块互相连接的所有可能的组合方式。从 Tetris 中抽去任何一种积木都会破坏它的平衡。玩这个游戏时，玩家常会遇到一些情景，呼唤着某种积木的到来。屏幕底部积木的某种排列往往会留有一些空间，而只有某种特定的 Tetris 积木块才能完美地把它们填满。Tetris 的平衡感部分来自于如下事实：Pajitnov 明智地采用了所有可能的积木类型并与所有类型的缺口相配。Tetris 这种积木块上的天然完整性和对称性正是它平衡感的核心。



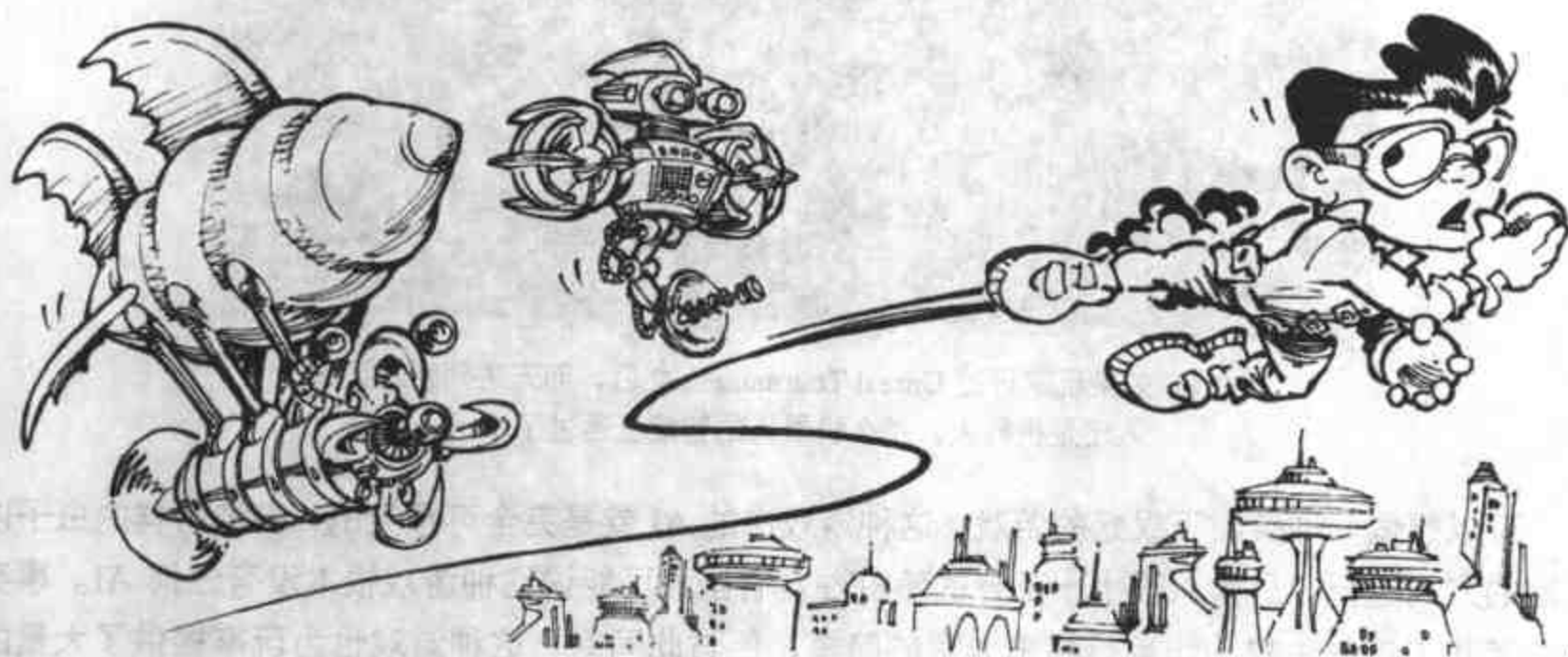
8.7 晚出十年，谁会发行它

假如 Tetris 是现在创做出来的话，真想知道谁会愿意发行它。起初，Tetris 是作为“来自俄罗斯的游戏”出售的，并与相同性质的艺术和音乐联系在一起，简直就是向我们那些当时在苏联的新朋友们摆的一种姿态。倘若 Tetris 是被哪个孩子在依阿华州的一个车库里梦出来的，那么它最终能否发行真是一件值得怀疑的事（可能会有人比较乐观，猜想他会将它编码，并在 Internet 上作为共享软件发布，最后这个游戏依然还会声名远扬）。Tetris 是低科技游戏里的终极产品，而很多游戏的出版商则会仅仅因为该游戏没有使用最新的计算机图形技术而拒绝它。不管怎么说，让他们到哪儿去给盒子封套找漂亮的屏幕截图呢？这个游戏没有故事甚至连背景都没有，这对那些市场推销人员来说同样也是不可或缺的。他们在广告里写些什么呢？事实上，Tetris 这样一个天才的游戏是注定无法用有限的文字适当地描述的，更不要说一句简短易记的俏皮话了。即使看到它的静态屏幕截图也无法让人兴奋起来，它根本无法传达出这个游戏的卓越艺术水平。游戏是一个文字描述难以表达的主题，它只能去体验而无法言传。



Chapter 9

第9章 人工智能



人工智能 (AI) 在不同的环境中有不同的含义。学术领域中人工智能有时是这样定义的：即一种能有效通过 Turing 测验的系统。在 Turing 测验中，人们在指定的电脑终端中键入各式各样的问题，答案就会出现在电脑屏幕上。即便这些答案实际上是由电脑提供的，但如果用户相信这些回答是由人提供的，那么人们也会因电脑通过测试而将其说成具有人工智能。

人们可以在电脑游戏中进行相同的测试。当玩家在玩 Unreal Tournament 时，分不清游戏对手是否是真人，他们就会说这种游戏已通过了一部分 Turing 测试，因此具有某种人工智能。然而，在实际情况下，即使这种游戏没有通过测试，人们可能也会说这种游戏具有人工智能，只不过不是那种很好的人工智能。当游戏研发者谈论人工智能时，他不是指电脑具有那种欺骗玩家以为对手是真人的能力。相反，游戏研发者指的是那种无论用什么样的代码来控制对手，游戏比赛都看起来像具有人工智能。电脑怎样对玩家这些活动做出反应都取决于游戏本身的 AI。这种反应可能是完全随意或完全符合逻辑的；不论在哪种情况下，那些控制这些反应的代码就被当做该种游戏的人工智能。



图 9.1 如果玩家玩过 Unreal Tournament 之后，而无法判断对手是人还是机器人，那么机器人的智能就通过了 Turing 测试

可以想像一种类似于蜈蚣的游戏。这种游戏中的 AI 就是完全可预见的，各种各样的虫子以既定的方式运行，只掺入很小一部分的随机性。有些人可能说这种游戏根本没有任何 AI。事实上，游戏中这些生物动作运行起来非常的简单，但与此同时，这种游戏也为玩家提供了大量的挑战。这种与蜈蚣类似的游戏生成 AI 的困难部分完全依赖于这些动物动作的设计，这些动物运动方式的逐步发现将会给玩家带来很有趣的挑战。相对于运行来说，AI 会有更多的设计。Tetris 可能是一个更极端的例子。这种游戏惟一的可以被称为 AI 的就是那个随机数字生成器，这个数字生成器决定下一轮会有哪个部分掉进游戏区域。然而 Tetris 被设计成那样以至于这也是游戏中惟一需要的 AI。

读者可能不明白为什么我在一本游戏设计的书中谈论游戏智能的问题。当然 AI 是一种编程工作，既然本书并非关于编程的，那么本章中涉及 AI 的讨论就似乎是题外话了。但决定 AI 做什么和设计哪些动作就是两项显而易见的任务了。首先主要是创造，其次才包括一整套繁重的工作。游戏设计者应确保游戏 AI 动作尽可能与构思相同，并且操纵起来尽最大可能给玩家提供挑战并使玩家在游戏中积累经验。设计一款好的游戏，其中一部分就是为游戏设计好的 AI。而对于那些正要离开 AI 创造升到程序设计员的设计者，更希望他们是好的 AI 设计者。如果不是的话，那他设计的游戏估计就不会很好玩。

如果电脑游戏像一个即席表演的电影那样，在那里玩家成为主要人物或人物组的指挥，游戏中其他人物由人工智能操纵。作为游戏设计者，你希望指挥那些 AI 控制的人物为玩家提供尽可能的刺激经验，那些 AI 部分的元素不只是玩家可能与之斗争的敌手，也可能是与玩家有关的任何角色。聚满人群的城市如何作为？他们怎样对待玩家的行动？设计 AI 就是设计游戏的很大的一部分。



9.1 游戏中 AI 的目标

在不同的游戏中玩家对 AI 期待的内容会不同。像 Centipede 之类的街机游戏或 Tetris 之类的迷宫游戏，玩家不希望有太多的 AI。正像我曾说过的，这种游戏用各种各样的简单思维或完全愚蠢的对手，就足以给玩家提供许多的挑战。而在 Close Combat 这种战争游戏中，玩家对敌方军队的智能则有更多的期待。在角色扮演游戏中，玩家希望进入一个各种角色多少能像在现实生活中那样行动的仿真世界里。而在 Sims 这种游戏里，AI 的多少就是游戏本身，如果 AI 太差的话游戏也就根本不值得玩。

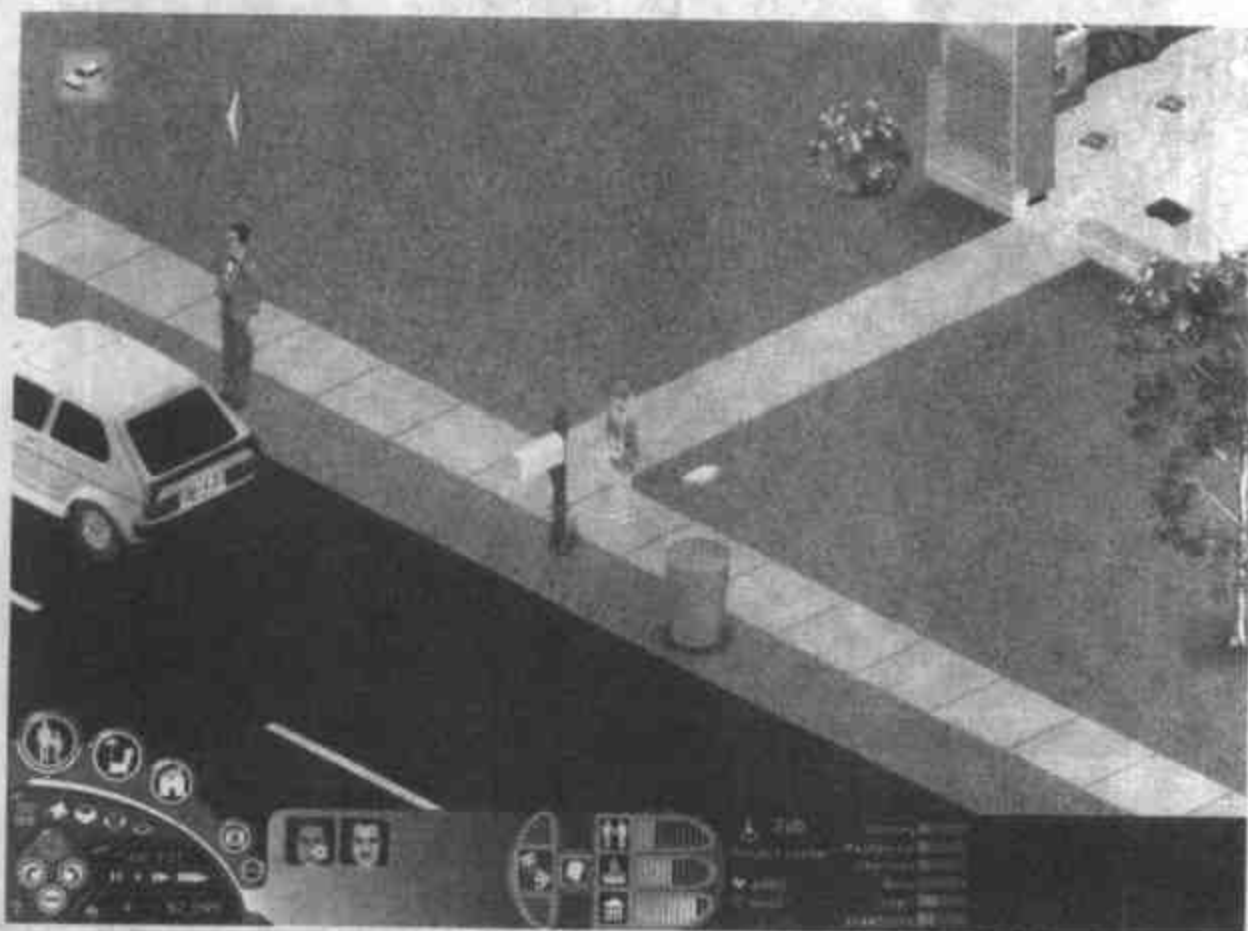


图 9.2 The Sims 的成功完全依赖于人工智能的优点

因此，在不同的游戏中玩家对 AI 部分所需要的自动程度会有不同的期望。但我们仍可以列出任何电脑游戏 AI 的常规目标，这些目标只有在给定游戏设计目标发生变化这种重大场合下才会改变。

9.1.1 向玩家提出挑战

为玩家提供一种合理的挑战一定是任何电脑游戏 AI 的首要目标。若不设置某种挑战，游戏就不会有挑战性也很容易被击破。更糟糕的是，若游戏没有挑战性，就根本不能成为游戏，而变成一种互动式的电影。

在像 Robotron 2084 这样的经典街机游戏或像 Doom 这样的第一人称射击游戏中，这种挑战大部分来自敌人数量上和能力上的压倒性优势。例如，像我曾用过的 Centipede 这个例子，那些虫子通过与玩家接触就能杀死玩家，而玩家只有射中他们才能将其杀死，这就使玩家陷入某种不利。这在许多虫子同时攻击玩家的情况下会更糟糕。结果是，这些虫子的 AI 可以相当简单并





且可以预见，但还是给玩家提供了挑战。

这种不平衡在 Doom 中也是实际存在着的，玩家可能弹尽粮绝，敌人则永远不会。在黑暗里玩家可能寸步难行，而敌人则会像在有光的情况下一样容易地发现玩家；像飞行生物那样的敌人还可以去玩家不能去的地方。这些生物数量上的优势补偿了它们智力上的贫乏。Doom 里的 AI 看起来要比 Centipede 的昆虫的 AI 更复杂，因为 Doom 世界似乎要比 Centipede 世界更真实，像我将要谈到的那样。这里的 AI 将一直以能力和数量超过玩家而为玩家提供挑战。

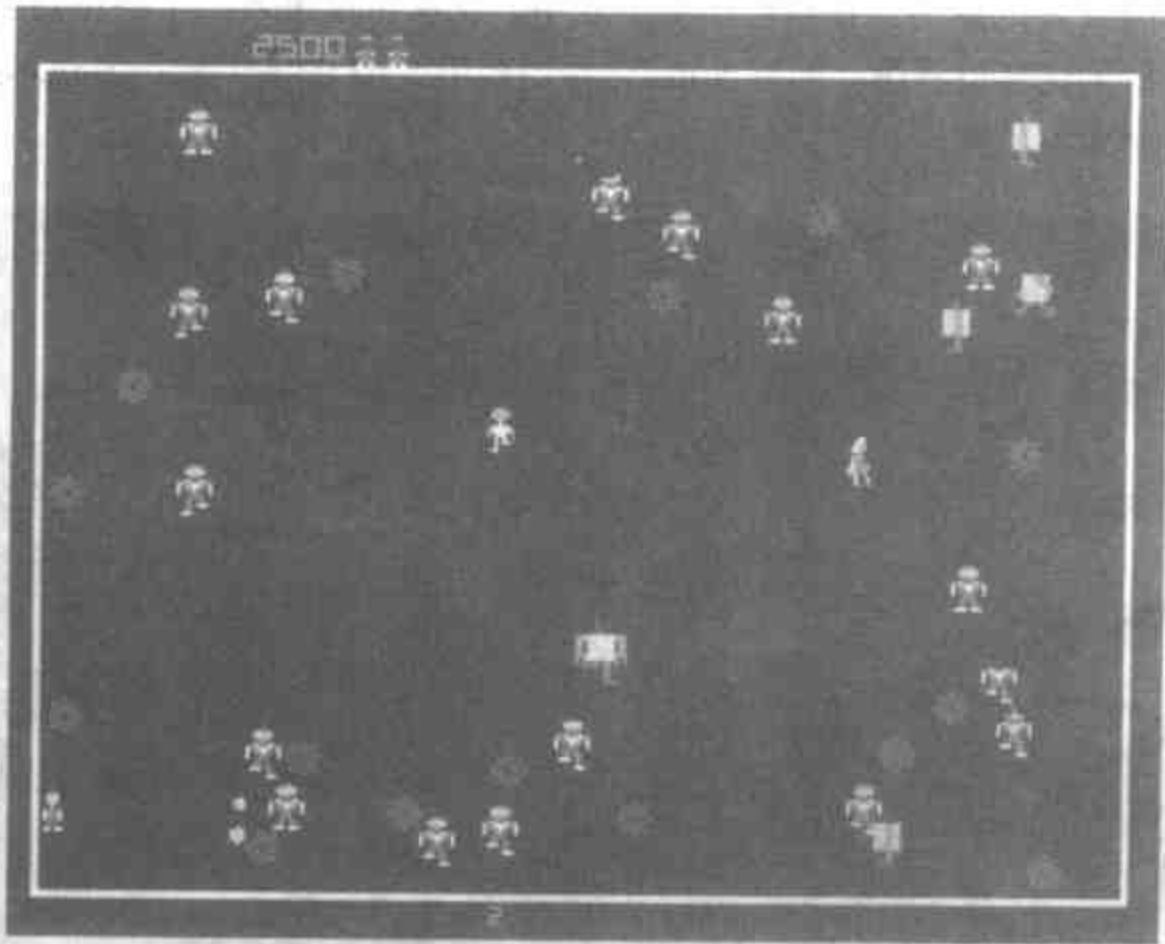


图 9.3 在诸如 Robotron 2084 这样经典的街机游戏中，其中的挑战在于，玩家要对付非常多的敌人

为像 StarCraft 那样的联机战略游戏设计一款挑战性的 AI 是更困难的工作。这种游戏需要玩家与对手力量相当。在一般的联机战略游戏中，玩家和对手每一方都可以指挥数量庞大的军队并在需要的时候可以建造更多。每一方通常都需要开采某种资源并用它来建造建筑物和队伍。基本上，实时游戏中的 AI 必须要做玩家做过的任何事，并且在做这些时也像是自动操作一样。AI 通常有一种优势，就是能看到玩家不能看到的各个级别，可能拥有数量更大的开始单元，以及更为容易获得更多的这些开始单元的途径，或者更大的可供提取的资源库。然而，因为实时游戏需要 AI 在个人单元级别之外计划各个单元的运动，就像玩家使用它们那样使这些单元看起来有组织地工作，为这种游戏设计挑战就是相当困难的了。

为实时游戏设计挑战性 AI 的困难只有在像 Alpha Centauri 这种基于回合制的战略游戏中才会被夸大。这里的 AI 被认为要像玩家那样操作。当然，基于回合制策略游戏是最为深思熟虑的游戏中的一部分，以至于夸大了设计强制性敌方 AI 的问题。更进一步地说，电脑根本没有从它极快的运行能力中获得任何好处。既然游戏是基于回合制的，玩家不得不尽可能按照他喜欢的方式运行。基于回合制战略的 AI 经常通过各种复杂的欺诈行为为玩家创造挑战，尽管我可能是最后一个去指责这样制作任何特殊游戏的人。





图 9.4 在 Alpha Centauri 这种基于回合制的战略游戏中开发具有挑战性的 AI 是非常困难的，因为玩家希望与自己实力相当的敌人作战

除了游戏类型外，AI 必须给玩家提供一种有趣的挑战。如果没有好的 AI，游戏就可能像你同你小孩子下棋一样，具有某种无目的性。不同之处在于，当你同小孩子下棋时，你希望教他这种游戏的细微差别，以便有一天他会成一个优秀的棋手。你也可以与他进行交流做一些其他值得做的游戏。可悲的是，电脑游戏 AI 根本没希望改进，并且若要提供玩伴，空间就非常不够了。游戏 AI 确定其生存的方式就是，必须为玩家提供挑战。



9.1.2 不要制作迟钝的东西

电脑游戏的 AI 一定不要显得愚蠢。当 AI 做得很笨拙时，玩家一定会加以嘲笑的。若 AI 主体对像消防栓或树这样的小障碍都过不去，或者当某一主体冲向悬崖时只能等待厄运的到来，这种情况只能造成玩家对游戏 AI 的不信任。对于玩家来说，在每种情况下 AI 应该做什么简单且显而易见。但是，对玩家来说显而易见的事情若由主体来完成或依赖理解，就是相当复杂的了。然而为了让游戏不至于成为笑柄，游戏 AI 就必须对玩家认为显而易见的事情有明确地了解。



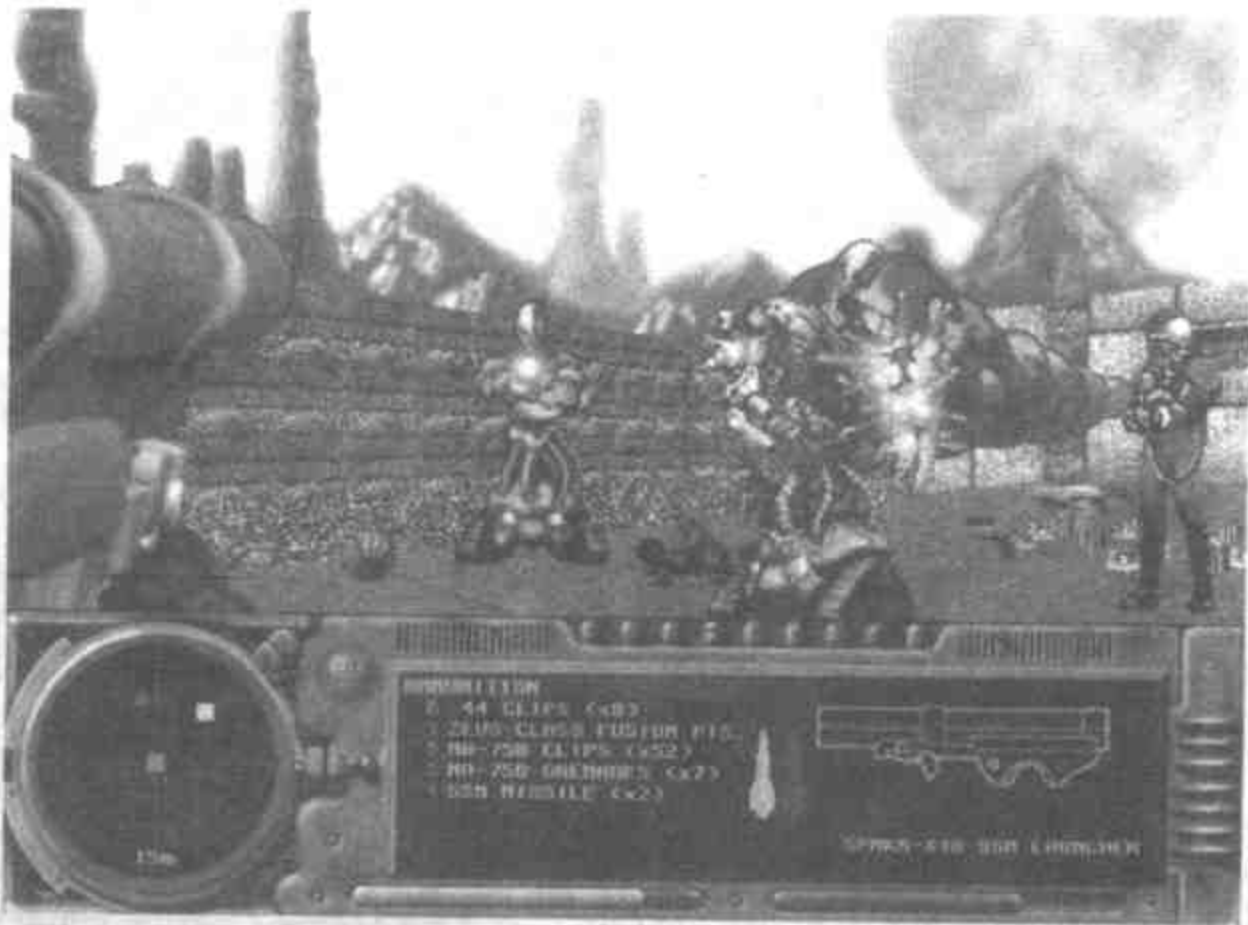


图 9.5 在 Marathon 2 这样的与外星人的作战中，玩家不会希望敌人有太高的智能

AI 能排除的麻木物体的数量与 AI 所要表现的智能类别有直接的联系。例如，在我的 *Damage Incorporated* 单人射手游戏中，玩家被假定为几乎独自同所有对手作战。而在 *Marathon 2* 中，玩家则是同混合一些机器人的许多地球外物种作战。*Marathon 2* 中的敌人不会显得很愚蠢，因为他们是非人类生物。相反的，在 *Damage Incorporated* 中，敌人也是人类，所以他们必须看起来更聪明些。对另外一个例子 *Damage Incorporated* 来说，根据游戏故事和游戏级别的外在表现，这些动作应像从真实生活环境里蒸发出来的一样。另一方面，*Centipede 3D* 在一个与真实世界只有类似切线联系的虚幻世界里产生了。因此，当在 *Damage Incorporated* 里的士兵需要像真人那样跟踪玩家时，当在 *Centipede 3D* 里的蜘蛛不能为玩家设计一条直线却还要在蘑菇丛中绕来绕去时，这种情况可能就少了一些不合理之处。AI 笨拙性的可接受程度取决于电脑游戏想要表现的环境。



9.1.3 不可预测

人类是不可预测的，这就是使他们成为游戏中好对手的部分原因。这也是人们喜欢玩多人游戏的主要原因之一。一个熟练的人可能以电脑决不会采用的方式作战，而这些方式就是人类对手的不可预测能力。电脑游戏中的 AI 对手也应该这样。当游戏渐渐达到某一点，这点就是玩家能确定的知道敌军在给定的时间内将要做什么，游戏的趣味性就迅速减弱了。玩家希望 AI 给他们惊奇，以不可预测的方式击败自己。当然多人游戏仍具有包括社交部分的优势，那是他们成功的主要因素，游戏中的 AI 将决不能像另一个人那样成为玩家的朋友。但如果不能提供多人游戏的社交部分，至少应努力使 AI 主体提供与人类对手所能提供的同样的挑战和不可预测性。

在所有艺术中，观察者希望看到他未曾预料过的东西，一些能挑战他期望的东西。当在最初的 10 分钟内，如果知道电影、书或戏剧的精确结局时，体验那些作品的很大一部分热情就会



丧失。同样的事情也会发生在电脑游戏中。当然，即便玩家知道预定的故事情节，或者知道下一级别会发生在什么样的环境下，或者知道那个大机器人老板像什么，游戏也能给玩家带来惊奇。但如果 AI 也能使这些变得不可预见，游戏就获得了游戏其他部分不能提供的再玩性。玩家将一直玩这种游戏直到它具有挑战性，不再有新鲜感觉时为止。AI 要始终给他们以惊奇，给他们以挑战，使他们兴趣提高。

成功的不可预见性在游戏中可以采取许多不同的方式。它可以像 Tetris 中那种决定下一次落下哪部分的任意键那样简单。当然这是非常简单的事情，乐观地讲我们也希望许多游戏能够提供更深入的不可预见性。但同时人们必须意识到，对于 Tetris，它是相当数量的不可预见性。如果玩家们知道下一次来的是哪一部分，游戏就会丧失许多挑战性。实际上，游戏的特征变得明显的简单起来（因为下一次会落下什么在屏幕的边上都有所体现）。纯粹的随意性真的是保持玩家对 AI 兴趣的好途径，它使玩家迷惑，猜测它要做什么，而实际上它只是随意的。Tetris 中的随意性提供的不可预测性需要玩家数小时地持续挑战。



图 9.6 Tetris 中所需要的惟一的 AI 就是随机数生成器，此图为 Tetris 的经典模式

有时电脑游戏 AI 的目标可能变得很混乱，当需要宗教的圣洁之杯时，设计者或 AI 程序员就会以一个愚蠢的对手来结束游戏的设计。当然，主体总是需要在当时给定的情况下做出理性的决定，他甚至可以做一些最可能赢得现在的战斗的决定。但如果这些逻辑决定对玩家是显而易见的，那与 AI 作战还有什么乐趣可言？你发现那里总是有一个在纺纱的海怪，将一个棍子举过头顶，游来游去总在试图捕捉你，下次你再去这间屋子时同样的情况肯定会减少挑战性。如果有时海怪被玩家的突然出现吓着了怎么办？这之后海怪是逃出大厅还是蜷缩在某个角落里？如果有时海怪决定将棍子掷向玩家而不是作为一种斗殴武器会怎么样？那都足够让玩家裹足不前的了。你要记住每个人是不同的，许多人都会因各种原因而具有不合理性。这些不合理使生活多姿多彩。如果玩家在电脑游戏中同人类或类人怪物/外星人作战，一丁点儿的不合理都会使





对手看起来更真实，更可信，玩起来也更有趣。

模糊逻辑是 AI 设计者和程序员用来试图保持 AI 主体的不可预测性和生动有趣的方式之一。主要的，模糊逻辑采用了一种逻辑系统并在里面添加了一些随意性。在模糊逻辑中，AI 在给定的形势下，有几个动作过程值得选择，而不只是一个。假设玩家在一个适当的距离带着适当的武器，而 AI 主体则在适当的安全级别内被相当多的武器武装。在这种情况下主体就有三种可能，它以不同的数字来表示每种选择的价值优劣。假设运行起来后攻击玩家时意义重大，因此会跃升 5 分。做一个威胁性的布阵方式吓吓玩家也会有点意义，因此会升 2 分。在玩家身旁绕来绕去迷惑玩家也是可看到的，那就会升 3 分。使用这些重量标准，玩家可以在 1 到 10 中任意挑选一个数（这是重量级的最高值）。如果小于或等于 5，主体就会运行程序进行攻击。如果是 6 或 7，主体就会只吓吓玩家。如果是 8 到 10，主体就会尽最大可能迷惑玩家。这种重量值给了 AI 一个在给定情况下做出决定的机会。如果 AI 有足够的不同方案可供处理，玩家永远不可能准确的知道 AI 将要做什么，使 AI 变得不可预测。在最后的分析中，AI 的决定建立在随机的基础上，使得主体看起来像是在执行一个复杂的判断，而实际上却不是这样。玩家永远不会知道那个值得怀疑的 AI 只是随意挑选的动作。相反，如果主体的动作看起来并不笨，玩家就可能试着猜测为什么 AI 选择做那些事情，并可能认为主体相当狡猾，而实际上它只是随机而动。

当然，游戏 AI 主体的不可预测性不能与我在这里曾提过的其他 AI 目标相矛盾。如果主体忙于不能预测性以至于不能组成一个坚固的攻击玩家的计划，他就没有太多的威胁性，因而也不具有挑战性。理性地讲，不可预见性增大了 AI 提供的挑战性，而不会有任何损伤。如果 AI 在正要走向胜利时随意选择做一些愚蠢的事，玩家无能为力只能困惑，为什么 AI 会做这么愚蠢的事情？当对付游戏中的生物时，一个眼睛盯着大一点的图像观察 AI 正要试着攻克什么是很重要的。

9.1.4 有助于讲述故事

游戏 AI 可以帮助展开游戏故事情节。例如，在 RPG 中，玩家可以游览一个城市，该城市居住着许多对外来人的到来心怀恐惧的居民。如果玩家只是观察这些人群，能看到他们游荡在城市之中，就像人们在真实的城市中那样走进商店、饭馆、工厂。这些为城市设立了一种场景，对于玩家来说使它看起来是真实存在的。但无论什么时候玩家一旦接近这些人群，他们就会转身走开，逃到安全的角落避免同玩家的接触。为什么会这样呢？对这座城市和那里居住的这些人群应该说些什么？为什么他们这样恐惧？玩家希望知道原因，于是就会开始探寻这个游戏故事。英语教师在这方面很著名，他们告诉学生最好用显示而不是告诉的方法。这在像电脑游戏这种可视媒体中尤其真实可靠。相对于只是以切片场景的方式看那些城市的居民对陌生人非常害怕，一种经过适当设计的 AI 能准确的显示给玩家这些有趣的信息。

即使与玩家作战的敌人，经过适当调整也会对故事讲述的进程有帮助。假设玩家同敌军的一个援兵作战，在这种作战情况下，可能就不会太符合逻辑。玩家可能派出一些狙击兵去对付敌方火力部队，那些部队在双方交火区域的边境服役。如果敌方部队的 AI 经过认真地设计，那种缓慢的进军方式就会激怒玩家。一旦被激怒，部队将佯装攻击然后从玩家军队中撤回，使其陷于不利。一部分游戏故事就通过 AI 在这儿讲了出来。





图 9.7 在 *Damage Incorporated* 中，玩家的队友所表现出的 AI 在讲述游戏故事方面起着重要的作用

在 *Damage Incorporated* 游戏中，玩家是一个掌管 4 人火力小组的美国海军陆战队的中士。玩家同他的士兵一起冲进许多由各种重型设备武装的对手的防御工事里。那些人都有彼此不同的强势和弱点。有些战士很顽强，作战勇敢。而那些新兵再开枪时较其他人更为小心，结果是打击玩家和其他队员的可能性减少了。这些个人特征都是通过队员使用的 AI 方式表现出来。在每一任务之前，玩家选出 13 个士兵组成自己的队伍，每一个都带有玩家能阅读的有关文件。这个文件提供了每一队员的生理资料，这样可以直观地透视它们的特性。更进一步地说，当实际执行任务时，队员也始终在不停地讲话，他或者是回答玩家的命令或者只对给定情况进行说明。这些可以进一步看出他们的特点和他们在战场上的表现。如果玩家读了这些资料，并且仔细地注意了这些广场士兵的个人特点，他就能注意提醒某些队员并不具有彻底的生理平衡。对一些队员来说，让他们执行过多的任务，对他们来说就是一种折磨。他们有可能临阵脱逃。另一些队员，若被赋予他们并不同意的特殊任务，可能会背叛玩家和他们的队员。AI 当然要控制这些情况，这就有助于讲述故事中的这些角色。

有一个 AI 设计者经常回避，但对故事的讲述却非常有用的情况。相对于将故事和游戏融合为一体，设计者更多的是在游戏中穿插故事。更进一步，设计者经常讲述一个固定的故事：在故事里，除了玩家在游戏世界中的行动和玩家怎样对付那些特殊角色之外，给定的角色怎样回应玩家都是完全预定好的。设计者常常努力使战斗和行动结果尽可能生动又不可预见，他们几乎总是希望故事在玩家每次经历时都保持完全一样。为什么玩家在遭遇不同的 NPC 时不能有情绪反映？可能如果玩家每次都认为这理所当然，每次都不问敏感问题，NPC 也会对玩家友好起来。可能玩家只能在那些第一次成为朋友后的角色里哄骗出重要信息。可能玩家只有名誉才能感动他们，玩家的一举一动都会在别的方面直接影响着 NPC 如何对待玩家。如果玩家在游戏之前已做了不太明智的举动，可能玩家不得不在继续进行游戏之前在一名角色的监视下获得解救。





当然，在相当广阔的领域里，不同的元素使用游戏 AI 可创造有趣的人与人之间的关系。可悲的是，这些游戏竟是到目前为止商业游戏中未被开发的部分。相对于讲故事，我更想指出一点，尽管未能完全阶段性地生成，玩家玩游戏的方式也会有细微的差别。使用 AI 从一个游戏到另一个游戏去增加情趣或改变故事使得讲故事更困难一些，但由此给游戏增加的非线性和再玩能力是巨大的。

9.1.5 创造一个逼真的世界

在许多游戏中，AI 不只是给玩家提供威胁与挑战。游戏玩家有可能根本不会直接接触 AI 主体。AI 可以代替控制游戏所生成的现实世界。游戏世界可以无限地由客体的内容、客体的外形与声音来描述，但玩家习惯于真实世界，这些真实世界也包含着为他们自己着想且以有趣的形式运行的有机体。因此，用枯燥的客体创造一个枯燥的游戏世界对玩家来说不能算一个真实的存在。人们不需要狂热追求用周围复杂的 AI 主体来填充游戏世界有一点就足够了。这是否意味着几只小鸟飞遍浩瀚长空，昆虫走遍大地，或者人类从事他们的日常公务；将周边的生活加入进来，对于玩家来说可以使游戏世界看起来更真实。真实程度越强，玩家置身其中的可能性就越大。

将周边生活填入游戏与用 AI 讲述游戏故事有着密切的联系。生成游戏中的居民对建立游戏的设置或讲述故事关键部分的设置有重要作用，但游戏中的周边生活却远远超出建立这些设置；它会使玩家在游戏中减少孤独感。多少次玩游戏时都感觉像在荒芜的无主之地上行走，好像一个利器曾成功的消灭了所有生命的讯号？玩家喜欢看到周边生活中有生命存在，他宁愿看到生物而不愿将它们杀死；在这个世界中增加死亡可能是没有价值的。

9.2 不平衡的游戏场地

程序员们在一起谈论电脑游戏中的 AI 时，经常会把谈论的焦点集中在怎样使 AI 主体与玩家保持平衡上。这当然就是我在几年前参加游戏开发大会中的 AI 圆桌会议上的案例。那些专家希望他们的 AI 系统能知道玩家知道的，能看到玩家看到的等。他们认为这样就会在 AI 与玩家之间产生冲突，使游戏更真实、更有趣。

当然，数年来，对于玩家来说，游戏赋予了 AI 更多的不公平的优势，使 AI 比玩家有着更多的制高点。他们与玩家的比例是 100 比 1。AI 主体在游戏的每一部分都有熟练的心理知识，这些知识使它们能在任何给定的时间里精确地知道玩家身处何方，这当然是不公平的优势。现在，有些游戏 AI 还有欺诈行为。AI 程序员当然会说这对玩家不公平。他们宣称 AI 应与玩家在同一起跑线上，并单独通过自己的智慧战胜玩家。

但将 AI 与玩家在竞技场上置于同一级别真的很好吗？首先，似乎极可能导致 AI 对玩家无法再提供挑战性。那时可能就会出现这种情况，一个特别差劲的玩家也可能比一个最复杂的游戏 AI 聪明，如果那个游戏 AI 没有制造多少麻烦的话。将玩家与 AI 放在同等条件下会给那些程序员带来更大的挑战，他们会把无数的时间花在那些也许只有一点点可能战胜玩家的 AI 开发部





分，却不能在其他部分的开发上投入太多时间。结果就是，他们结束了 AI 设计却没有提出任何引人入胜的电脑游戏。更糟糕的是，AI 往往会忙于运行一些闹哄哄的愚蠢步骤。

在游戏领域中，促使程序员将玩家与主体放在同一标准的主要原因是程序员的自我意识。那些程序员满足于自己的工作，并宣称能使 AI 不用高级的数字、更强的力量或任何的欺骗就能挑战玩家。他们希望自己的 AI 能与人一样聪明。他们经常把许多时间花费在追求所谓平等的复杂运算规则上。结果是，有些东西必需捆绑在一起才能实现。这种游戏 AI 的目的是支持游戏并增加玩家的经验，而不是作为人工智能技术的试验平台。

另外，对于玩家来说，除了 AI 军队数量上绝对超过玩家，有着更好的武器，甚至具有科学地图的优势这些情况之外，玩家击败 AI 对手还有一些浪漫的事。就像好莱坞动作片中的英雄克服无数的障碍那样，玩家也希望为胜利而战胜不可克服的困难。增加平衡表中 AI 的优势，只会使玩家在得到最后的成功时感觉到成果更香甜，当然除非这种设计使游戏更困难。



9.2.1 要多真实才算真实

另一个潜在的 AI 设计陷阱是设计一种 AI，而该 AI 尽管实际操作时像个真人，但结果却降低了游戏的整体价值。在他们讲述的故事中，游戏经常发明于特别严格的真实世界，只是因为有趣，不是因为真实，也不是 AI 必须支持。

想像一下詹姆斯·邦德的电影。就像许多大众游戏，他们注意设计动作和惊险的场景，对于角色发展或故事是否有意义却极少关注。在几乎每个电影中，邦德都在某种情况下被抓住，并被绑在一个特别丑陋的死刑架上。这种设备并没有对邦德立刻执行死刑，相反却采取了某种缓慢的方式，比如在绑着邦德的桌子上用激光稳稳地在桌面中央烧出一个孔。为什么那些坏蛋不马上杀了邦德？或简单地将激光对准他？为什么坏蛋几乎总是在死刑还没有执行完毕时就离开？为什么坏蛋在执行死刑时总要对邦德讲述整个的犯罪计划？这些没有一样是明智之举，但看起来却很有趣，也适用于电影的完整情节。邦德电影的主要目的就是娱乐观众。就很少会考虑真实性了。

这也同样适用于游戏。如果敌人足够聪明，他当然应该意识到自己没有机会战胜玩家，那他就应该把自己锁在安全的坑道里，不为任何人打开那扇门。实际上，这样做是在自我保护，这无疑也是最明智之举。那他对游戏的意义何在？玩家就会被困住，他到不了敌人的区域，游戏也就无法进行。另一个例子可能是笨拙的主体在负伤累累的情况下从玩家身旁逃开。许多游戏都受这种方式的影响。但如果主体比玩家还快，若负伤到一定程度时，是否就可能先跑进安全地带。AI 主体就会逃离战斗，而玩家别无选择只能尾追其后。如果 AI 速度更快走得也更好，玩家要抓住他就会很困难。那结论就会是，本应是一个有趣的游戏却变成无聊的追逐，因为 AI 主体已严重受伤并无康复的希望。玩家会同那些蛇一般的狡猾老板打仗，这些老板全副武装，远比玩家装备精良，只有在张嘴时才能被射死。因此严格的逻辑选择的结果是，无论玩家是否有机会射击，他们始终闭着嘴。这种决定很简单，但玩家永远无法取胜，那还有意思吗？

还有一点是，AI 决不能盖过电脑游戏，它决不能改变开发组计划的真实目标：制作有趣和可玩的游戏。如果 AI 真的非常复杂，那结果必然是游戏不能玩或极其困难。玩家不会说 AI 多么聪明。玩家可能注意到那些先进的运算规则增加了给定标题的可视性。他可能只谈这些并欣





赏这些游戏的运算价值，即便游戏本身不怎么样；但一个无程序基础的玩家对那种复杂的 AI 却会瞧不上眼，如果游戏的特性早已决定了它并不好玩。



9.3 AI 主体及其环境

电脑游戏不能在真空中设计或开发。游戏 AI 若要运行良好，需要与游戏及游戏环境紧密结合，游戏环境就是游戏发生的地方。而一个简单的事实是，没有任何 AI 主体能够聪明到在任何情况下都能执行。在宽阔的地方，有些 AI 可能运行的很好；但若将它们放入狭窄的领域中，可能就会遇到程序员未曾预料到的问题。如果程序员设计出的 AI 是在有限的空间里运行，那他在开放环境下可能就运行不好。最好是 AI 在特定种类的游戏环境中有作战机会。如果级别和 AI 不在某一特定时期开发的话，那么玩家所对应的对手就根本没机会表现得更聪明。

这就产生了一个问题，即怎样最好地制作一款游戏。在进入最后平衡前，级别设计经常是游戏开发中的最后一个任务，然后是评测，最后是销售。同样，AI 通常在游戏各部分都各就各位时才会工作，玩家运动的大部分机器都能运行，许多其他重要的程序任务也完成得差不多。现在，如果同一个人既设计级别又制造敌人 AI，将两者的开发融为一体就可能很简单；但这在现代游戏中，即便不是闻所未闻，也是相当稀少的。结果是，你拥有两组队员，程序员和级别设计师并行工作。不幸的是，这两组人员经常是各自为政，不会充分考虑到对方的情况。级别设计师还没有 AI，他也不会让他的级别支持 AI。等式的另一端则正好相反；程序员没有级别，因此使它的 AI 在那些级别上良好运行就比较困难。这种情况就是 Catch-22。一旦级别制定出来了，AI 最终要添加其上；如果游戏有一定的趣味性，其中一组就要理智地重做。最糟糕的是两者都没时间重做，那结果就是游戏自相矛盾。

当然，级别设计者会强调 AI 应按设计出的级别制作。同样，程序员抱怨级别应由开发出的 AI 决定。因为这两样工作我都做过，我可以以一个特殊的立场来裁判这场纠纷。我的意见是，哪一方都不完全正确，每一方都需要做出一点取舍。我乐意试着首先做些简单的好玩的 AI，他不必在每种情况下都自由而完满的运转。如果他在某些情况下运行的相当好，级别设计师就可以制作级别以促进这种已知 AI 的运行。在级别设计师指引的方向下，程序员可以进一步工作，排除游戏世界中一目了然的弊病。程序员若看到了级别中的问题，例如 AI 不可能处理的情况他应与设计师交流。同样的，设计师也可以给程序员有关 AI 技巧的建议。这样，游戏世界里的鸿沟有可能消除。游戏设计的这种结构可能更适合大组方式。如果程序员在运算规则中增加功能，使得主体能分辨出这些位置并采取相应的行动，那么结果必然是 AI 的功能也增强了。

若那些具有冷酷外表的几何学不能让 AI 起作用，级别设计师就不得不丢掉它们。如果 AI 不能运行，游戏就毫无乐趣可言，而级别提供给玩家的首要目的就是，给玩家带来紧张而有趣的经历。在我的 *Damage Incorporated* 游戏中，玩家不仅要对自己负责，还要在 3D 环境下指挥 4 个组员。当制作这款游戏时，我遇到的最大挑战之一就是，让组员 AI 以对玩家看起来比较智能的方式运行。幸运的是，在所有真实的级别设计开始之前，我已有这种 AI 的基本工作模式。我认为，这款 AI 比那些还不能在区域里跳或游的组员 AI 超前。这就意味着级别必须有根据地进行设计，否则组员就不会同玩家闯关到最后尽头。组员也会过于受束缚，活动空间有限，四处





乱跑或阻碍玩家的进程。级别要在较宽阔的区域里制作，以便 AI 主体有机会良好地执行。

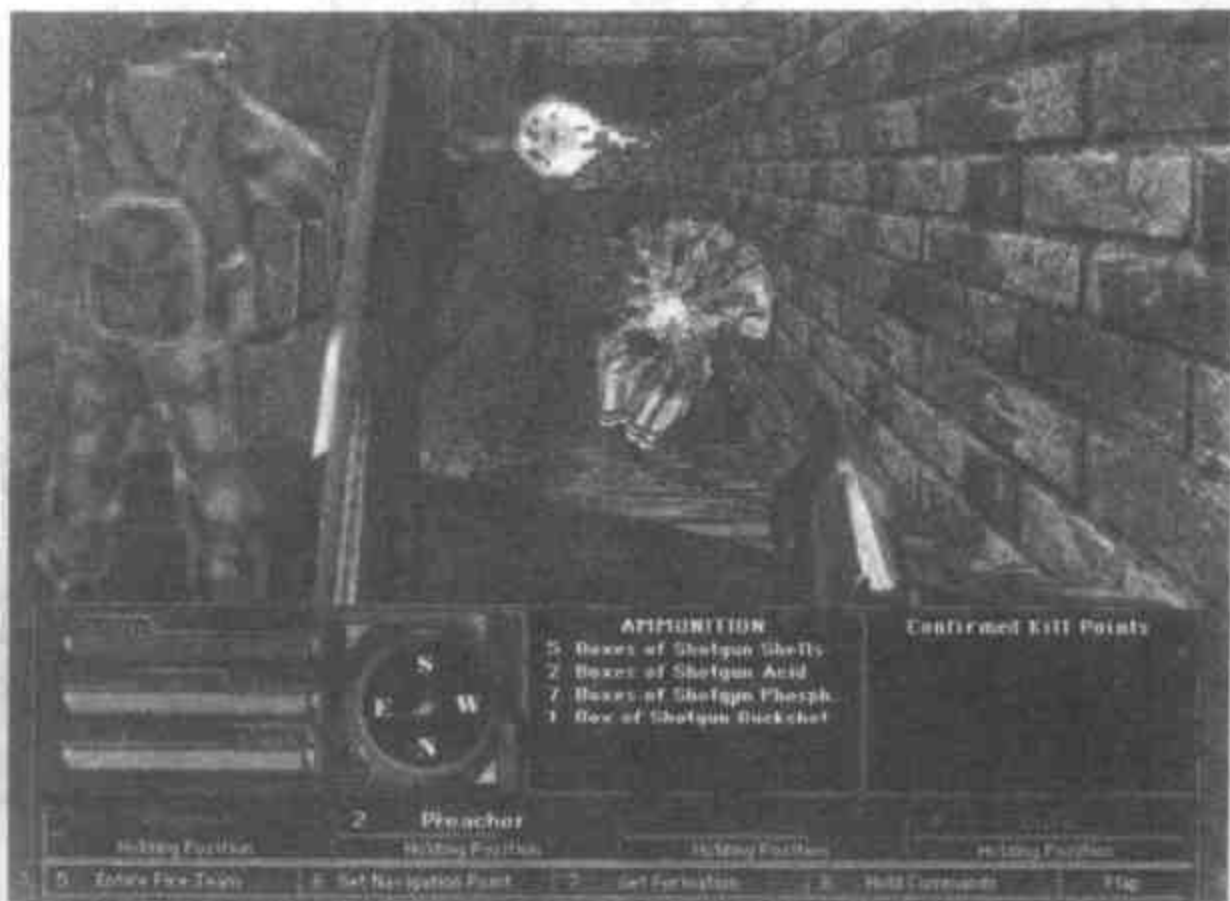


图 9.8 使 Damage Incorporated 中的 AI 主体能够正常运行，需要对关卡做出许多修改

即使具有复杂游戏 AI 的先决知识，一旦进入评测阶段，仍会有无数的问题。组员经常会在根本不曾料想的小地方被困住。最终的解决之道是，25%的代码保留下来，75%的代码重做，使得部分级别减少主体阻碍自己的机会。会有无数我本来希望有一定用武之地的级别部分要被迫删掉，只因为 AI 在那里不能运行。看到被删掉的部分，我很悲哀，但也不会有玩家眼睁睁看着他的组员在裂缝里裹足不前那么悲哀。若要使最终的游戏玩起来有趣味，AI 和级别应该合作良好。



9.4 要多好才算好

Damage Incorporated 提示了有关在不同游戏中所需 AI 复杂性的另一个有趣点。正是这一点，使得 Damage Incorporated 这么具有挑战性。这一点就是玩家依靠 AI 为自己执行特定行动的事实。如果玩家命令组员移到特定的地点，他希望潜艇也到那并进行作战防御。如果 AI 做不到，那玩家可能只有死路一条，并可能埋怨 AI 害了他。更遭的是，如果玩家命令 AI 在某一特定地点重置，而大队伍又很难到达那里，玩家就会很泄气，特别是当到达那个地点的相应道路对玩家来说相当明显的时候。但如果敌军主体也没法发现到那儿的路径时，那笨的就不只是玩家一个人。如果对手偶然间困在某个角落，玩家可能就会把 AI 主体的愚蠢当做一种功绩并带着大批军火前去扫荡。然而，如果组员也深陷窘境时，就不会跟随玩家一起成功闯关。玩家的部队不到达 0 级，玩家就不会有最终胜利，AI 的错误会使玩家不幸夭折。没有什么能比错误的 AI 更使玩家泄气的了。



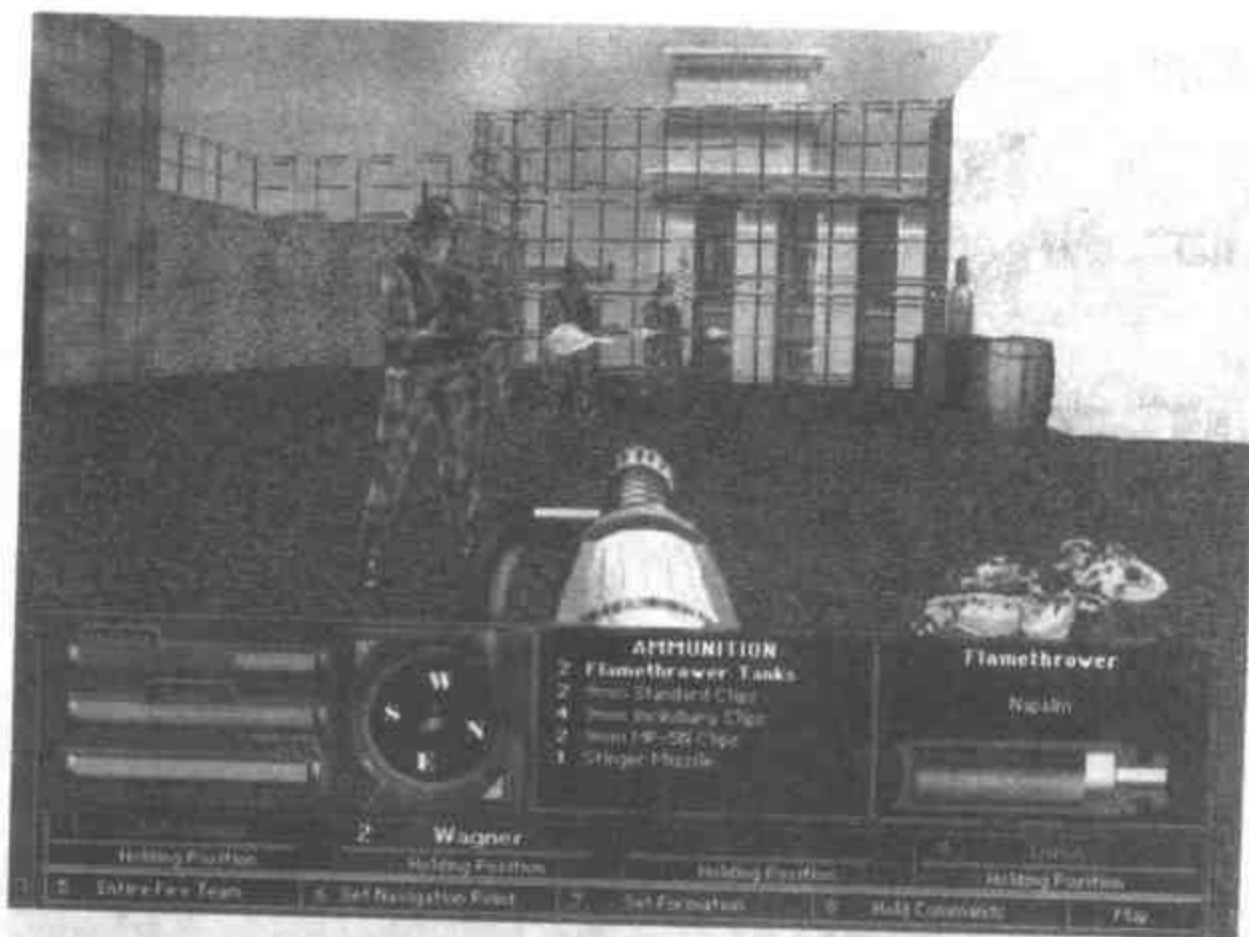


图 9.9 在 *Damage Incorporated* 这种有队友参加的游戏中，如果 AI 主体没能与玩家有同样的表现，那么就会对玩家玩游戏的能力造成障碍

大家可以从 *Damage Incorporated* 运行中总结教训。一是绝不在复杂的 3D 环境下同组员制作游戏。另一个结论就是，游戏所需 AI 复杂性的数量依赖于 AI 对玩家的失败的影响。如果 AI 紧紧拴住玩家之时游戏却结束了，那就坏透了。如果由于 AI 的错误使得惟一的结果是使游戏变得容易了些，只要它有机会存在，玩家就会与失败相伴始终。因此设计师制作 AI 系统和检查程序时，必须始终注意系统的正确运行有多么重要。只有这种情况是可以接受的，即只有开发组能知道 AI 的不足，而玩家完全看不到它的缺点。

使游戏的每一个系统都尽可能聪明当然很好，但生产周期的实际表明，用于开发游戏每个给定部分的时间有限。程序员很少能在做完工作后仍有时间做一些他喜欢的装饰工作。花很多时间在过量的 AI 系统上就会占用其他必做工作的时间。读者可能注意到，我在上面列举游戏 AI 所需要的要素时，并没有列举相应的学院质量的人工智能。游戏只需做到能挑战玩家且操作起来不会显得太笨就可以了。

在那个万众瞩目的 *Buried Paul* 游戏开发者大会上，Brian Moriarty 讨论了游戏中 *Constellation* 的概念。这个理论与游戏 AI 有特殊的关系。概括起来其理论就是，在提供了一些相似的随意性数据后，人们可以使其具有一定意义，使其遵守秩序，在可能有的地方寻找含义。事实上，什么也没有。那么对于 AI 来说，Moriarty 认为 AI 操纵起来像是随意的动作会使玩家认为 AI 有某种完美和智能的计划。玩家可能按照以下的思路思考：为什么坦克能突然间冲过小山？似乎没有什么原因。可能他们知道一些我不知道的东西。可能它们在我看不到的地方重新组织了军队？玩家不是游戏开发人员本身，他们可能会相信游戏 AI 主体做出了智能的选择。当然这样很好，如果玩家看到 AI 主体无目的地撞进了一堵墙，他们可能认为有东西出错了。玩家不希望在游戏发现糟糕的东西，他们相信他们看到的那些角色的智能，记住这些是很重要的。由于投入了一些随意性的动作，你的 AI 主体看起来要比实际上聪明得多。



9.5 脚本化

当然游戏 AI 不需要及时考虑到每一个游戏中执行的行为。在某些游戏中，预定轨道上自动 AI 和手动行为的结合会为玩家带来最大的刺激体验。通常，手动工作在那种有预定区域而玩家又不可能重复玩那些级别的游戏中最适用。在那种游戏中，玩家可能从某一地点进入给定的区域，因此设计者可以假想任何给玩家带来最有趣挑战的攻击计划。

单人射击游戏是游戏与手动行为协调极好的例子之一。Half-Life 可能是用 AI 手工制作玩家乐于与之作战的对手的一个经典例子。这个游戏因为 AI 的力量在电脑报中被广泛的赞扬；然而，事实上许多人指出，AI 主体若进入特殊的情境时，手动路径能够战胜智能。

特意为级别设立手动行为已被级别设计师充分考虑过了。级别设计师早已需要考虑对手应该被放在什么地方以最大化地发挥游戏的效力。对于手动行为，设计师需要反复考虑，以设计出能让 AI 隐藏的最危险区域，这一区域可以在体质下降时用做撤退之路，怎样设置才能选择最好的机会击败玩家。当然，AI 主体不能总是走一条道。AI 必须仍旧被用于使主体决定在何种情况下哪些位置需要攻克。更进一步的，当手动计划没有开发出来或不准备开发手动计划时，这种 AI 也必须能够实现。有些人可能认为 AI 主体采用手动的预定的行为不会产生出我曾讨论过的不可预测性。有些人可能奇怪手动行为怎么会有预见性。基于这些原因，手动行为应该给 AI 主体一些建议，这些建议有，哪些地方应为零分？或者哪些地方准予通过，但不一定说明主体必须经过那里。主体必须能反击玩家而不至于显得太笨。



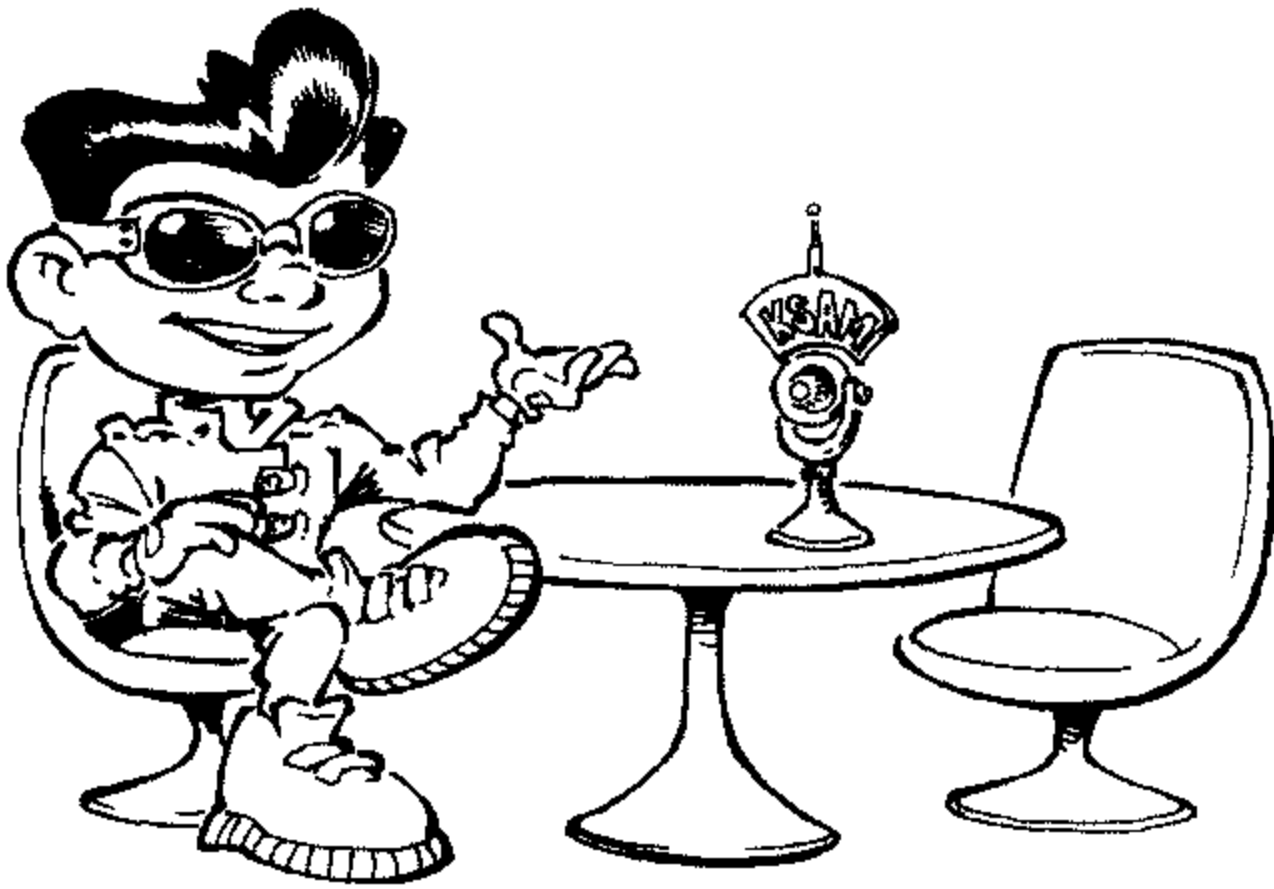
9.6 人工加入的低能

游戏中经常会提到这样一种事实，就是许多游戏中 AI 的质量与其说是人工智能毋宁说成人工低能。几乎所有游戏中的 AI 都不是游戏本身自带的这样的现象当然令人印象深刻。然而当时想一下试图阻碍玩家的条件是什么，任何好的游戏都真的会很好地设计电脑对手。尽管多人游戏为有趣的游戏设计和制造提供了许多途径，但相当多的游戏人还是喜欢单人游戏。单人游戏会提供一些独特的经验，游戏 AI 尽可能使游戏的经历变得生动有趣，但却有些困难。去仔细地考虑人工智能是设计者的责任，设计者应确保对 AI 必须做的事情有着清晰的了解，以便完成挑战和取悦玩家。



Chapter 10

第 10 章 Steve Meretzky 访谈录



在 80 年代早期，Infocom 的游戏就非常与众不同，以至于公司里的工作人员并不把它们叫做“游戏”，而称为“互动作品”。与其他游戏公司发行的街机游戏克隆产品和演化产品相比，Infocom 公司的作品显得非常独特。因此，它的互动作品吸引了一大批精于此道的游戏爱好者。这些游戏内容非常新颖，带有浓厚的文学色彩，并且制作也相当专业化，在当时来看，还没有其他游戏制作能与之媲美。他们强调以文为本的原则，使游戏有了文学的成分，从而得到了一定程度的尊重和认可，对游戏 Deadline 的评论足以收录到纽约时代回顾书刊中，而两位游戏设计者 Steve Meretzky 和 Dave Lebling 也因此以互动作品的作者身份加入美国科幻作家联盟。在此之前，纽约时代回顾书刊从未将电脑游戏列入其中，而科幻作家联盟也因此不得不修改评选制度，以阻止其他的互动型作品作者加入其中。Steve Meretzky 是 Infocom 最有天赋的天才之一，他不但参与设计过 Infocom 的畅销游戏 The Hitchhiker's Guide to the Galaxy，还参与设计过另一款备受推崇的游戏 A Mind Forever Voyaging。在 Infocom 被转让后，Meretzky 成立了自己的公司 Boffo Games。该公司推出了广大游戏迷们喜爱的 The Space Bar，使用一些传说故事作为游戏题材，以继续保留在冒险游戏中的文学传统。现在，Meretzky 正与 Internet 游戏公司 WorldWinner.com 合作。但是近年来冒险游戏已渐渐失去发行商和游戏迷的青睐，人们



对此无力挽回，只是感到疑惑：那些曾经为 Infocom 带来巨大成功的游戏迷们现在为何不再青睐于它呢？



问：电脑游戏最初吸引你的是什么？

答：在 70 年代末 80 年代初的时候，我实际上还是很反感电脑游戏的，事实上我讨厌一切和电脑有关的东西。我认为它们是低俗的、反社会的，而且似乎只要谈及和电脑有关的事情，英语就会被滥用得一塌糊涂。自从计算机科学实验室开发出游戏 Zork，在我的宿舍很多人爱玩这个原始主机型游戏，而我对此非常反感。我只是偶尔在实验室玩玩 Maze Wars，有时也会对 Space Invaders 嗤之以鼻，但仅仅如此而已。

这种情况一直到 1981 年。那时我的室友 Mike Dornbrook 是 Infocom 的第一个也是唯一的一个游戏测试员。他开始在我们的餐桌上用 Apple II 测试 Zork I。当他不在的时候，我偶尔也去玩一玩，并且很快就迷上了这个游戏。不久，Zork II 又出现在我们餐厅的“测试桌”上，我找出了所有我能发现的 bug，虽然麦克找 bug 是有酬劳的，而我则是义务的。



问：是什么原因让你去 Infocom 工作的呢？

答：我在麻省理工学的是建筑工程管理专业，在 1979 年 6 月毕业后的头几年，我一直从事这方面的工作。那工作真是太糟糕了：单调的工作、乏味的同事、管理落后的公司。所以在 1981 年秋季，当我的室友 Mike 去芝加哥读商学院后，Marc Blank（Infocom 发展部的副经理）需要一名新的测试员来测试他即将推出的神秘游戏，这个游戏后来正式取名为 Deadline。由于我在义务为 Zork I 和 Zork II 测试时已经向他们证明了我是一名合格的测试员，Marc 决定正式雇佣我代替 Mike 成为新的专职测试员。当时，Infocom 还没有设立办公室，只有一两个全职的工作人员，所以我继续在家里用 Apple II 来测试游戏。

1982 年 1 月，Infocom 搬到坐落在剑桥边上环境舒适的办公室，于是我开始工作，但并不承担任何管理职责，我的工作就是测试 Deadline，然后是测试 Zork III 以及 Dave Labling 继 Zork 之后的新作——Starcross。6 月，我成为一名半职雇员，仅仅是一个承包人。

甚至在这个时候，我还没有设想自己会成为游戏作家——我仅仅是觉得在改变中做了一些有趣的事情而感到很高兴，或者是说在寻找下半生想做些什么。同时我也写了一些关于 MIT 的东西，向各种杂志社寄一些科幻小说，但从未被发表过一个字。



问：你是怎么从测试人员向作者转变的呢？你不需要首先证明自己一下自己吗？

答：1982 年夏末，Marc Blank 问我是否对写游戏感兴趣，我立刻答应了，仔细想想，测试游戏非常有趣，写游戏可能就更有意思了。源于以下原因，我没有必要证明自己什么。其一，在 MIT 的时候我就已经认识 Marc 很多年了，我们一起从事校园电影的筹划，他知道我工作非常努力而且很积极。其二，Infocom 公司非常小，且很不正规，实际上没有任何行政机构的批准。其三，在测试游戏的过程中，我已经表明我对这个游戏的理解和对设计过程的疑惑。





问：Planetfall对你有什么启发呢？

答：作为读者和作者，我最感兴趣的是科幻小说，所以我认为这个游戏将会是 SF。角色互动是 Infocom 开发系统中最薄弱的环节，那么像荒废的星球这样的游戏场景就算是好想法了。除此之外，我没什么好讲的了。



问：你设计Floyd这个角色出于什么目的？

答：塑造一个生动的非玩家角色是 Planetfall 早期的设计重心。Infocom 游戏为达到这一目的，通常会在每个游戏中设计五六个这样的角色，比如在 Zork II 中就有巫师、妖怪、龙、公主和侏儒等角色。由于这样的角色过多，反而使每个角色都显得很平庸。我认为只需要一个角色（不包括开场时 Blather 和外星大使的短暂出现），可以把它做得更有趣、更生动。

我不记得怎样得到 Floyd 这个想法的，尽管“聪明的机器人”是很早的决定。可能是受 Star Wars 三部曲的影响，接着是 Empire 和 Jedi。而 Robert Heinlein 的著作《Red Planet》中一样聪明的外星人——Willis，对我也有一定的影响。



问：你是一直计划将游戏设计成只有Floyd被杀死才能赢得这场游戏吗？

答：不，这个决定是在设计游戏的过程中才明确下来的。这样 Floyd 呈现的比预先构想的更具幽默感，更加富有感情：发现老朋友 Lazarus 尸体的时候，他会在你肩膀上蹭头，表示受到感情伤害，等等。很明显，人们越来越喜欢他了，这在某些方面也正说明我能创造出这样感人的一幕。

而且这个决定所带来的另一个影响也值得一提——那时，Electronic Arts 公司刚刚成立。他们做了一系列的广告，建立了稳定的游戏设计群。他们其中的一个设计者如是说：“我想创造一个电脑游戏，让看见它的人都流泪。”我对刚刚出现的竞争对手稍有所了解，而且希望能够赶超他们。



问：《Hitchhiker's Guide to the Galaxy》是对畅销广播剧和图书的改编，你又准备怎么把一个线性的故事游戏编成互动的形式呢？

答：它的确是改写最理想的境界，简短的故事线索，包含伟大的角色、宏大的场景和高超的技术等，虽然故事的线索不是惟一重要的。但它是一种挑战，而且是一种好的挑战，没有坏处。



问：你是怎样和Douglas Adams一起共事的？

答：顺便说明一点，Douglas 早就是个 Infocom 迷，并且玩过好多我们的游戏，他知道什么是冒险游戏，了解我们系统的潜力和缺陷。另一方面，他以前从未写过非线性的游戏，把握它总是需要一个困难的过程。而且刚开始时我有点害怕和他一起工作，没办法完全坚持自己的原则。所以你们可以看到游戏的开始有点太过线性，包括 Arthur 房子的毁灭和 Vogon 船在海上的





场景。后来，Douglas 逐渐适应了互动的的设计，而我也没那么羞怯了，我们的游戏才编成现在这种非线性的设计。和 Douglas 合作非常棒，他很聪明，很有创造力和幽默感。Douglas 不是每时每刻都在搞笑，但他在写作里却能用大量的词句创造讽刺和辛辣。他经常创造一些我从来想象不到的滑稽方式：让游戏对你说谎，有“no tea”这样史无前例的句子，把一些犯了语法错误的词掉进蛀虫洞里并由此引发一场星际大战等。



问：你们俩之间是怎样平等分工的呢？

答：刚开始，我们一起设计游戏，Douglas 写那些重要的文本段落，而我再对它们进行加工润色，并使用 Infocom 开发系统完成这种高水平的编程。

1984 年，在我们开始工作一个星期后，Douglas 来到剑桥。在那个非局域网 e-mail 还很少使用的年代里，我和 Douglas 每天互发 e-mail，每个星期我们还会打电话交换意见。Douglas 还是世界上最拖拉的人，这一点最让人忍受不了，他总是把时间表上的计划往后拖了又拖。同时，他着手的第四本关于 Hitchhiker 的书《So Long and Thanks for All the Fish》在拖了一年之久后仍只字未动。

公司怕他在伦敦分心，于是把他派到了英格兰西郊一个乡村小旅馆。这家漂亮旅馆原来有个高贵的身份：Huntsam Court，我去那里看他，并花了一个星期来完成游戏的设计，然后我再返回美国。为了及时赶上为时不长的暑期测试，我仅仅用三个星期完成了游戏。9 月份，Douglas 赶回来完成游戏关键文本部分的最后创作，我们的游戏在 10 月底得以发行。很快，游戏登上了最畅销名单的榜首并一直持续了好几个月。



问：我们看到的 Hitchhiker 是一种非常复杂的游戏，这是你的创作意图吗？

答：Douglas 和我们都认为探险游戏有些过于简单，新的 Zork 就比最近新出的很多游戏要复杂得多，而它们最令人着迷的地方正是对玩家脑力的每天 24 小时千方百计的考验。所以我们认真吸取了这些经验，并在其他方面做了更深入的工作。当然，Infocom 的测试员工们也一直极力主张游戏简单化。

另一方面，对游戏意见的大相径庭也成了一个经典，Infocom 甚至开始打出 T 恤的招牌广告，“我解决了鱼的困惑。”很可能一些人被游戏的难度吓倒，而另一些人则被其深深地吸引。我的感觉是，人们发现很难从游戏中得到提示，但觉得太简单的人却会因为无法使它变得复杂一点而苦恼。

对游戏测试的时间过短可能对游戏的难度做出了贡献，由于 Douglas 的拖拉，我们不得不做一份更积极的时间表。通常，过多的测试会导致游戏向简单化发展，因为就算只有一个测试者，他也会倾向于把游戏简单化。



问：A Mind Forever Voyaging 完全不像你其他的名作那么富有幽默性，但你仍然认为它是最得意的作品。你的目的就是把这个游戏做得很严肃吗？

答：是的，部分原因是我刚刚完成一个纯喜剧游戏的反作用（HHGTTG），部分原因是因





为我感觉互动故事是一种非常吸引人的手段，它真正地接受了某个人的生活，是一个表达某种政治或社会启示的理想方式。正如游戏中那样，我尝试着改变这个世界。我的目的不仅仅在于创造一部有启示意义的严肃游戏，更在于消除困惑并增强其可信性。



问：玩家在AMFV中存在的虚拟环境很有趣，有别于其他的Infocom游戏。你觉得有必要用这个标题“打破模式”吗？

答：我不太确定是什么启发我想出让一部有意识的电脑作为AMFV的主要角色，但我记得当时我正和一位室友坐在餐桌前吃饭。海上航行和界面的不同看起来似乎只是对最初想法的自然延伸而已。这种“打破模式”的方式并不像我想的那样，正如我先提到过的，我希望能在游戏的内容上做到“打破模式”。



问：你在做Infocom项目的时候遇到过很多麻烦吗？你在做先前的游戏时自由发挥了自己的想法吗？这个游戏中的恐怖会有些与众不同吗？

答：根本就没什么阻力，事实上我的游戏能达到今天这个水平，判断力和市场的成功功不可没。但那个时候Infocom的观点却给我们主要的玩家造成了混淆——比如Zork和Enchanter——和许多试验型的游戏一起，旨在积极推动封面的发展和扩大互动故事的玩家。后者的典型例子就是像Seastalker和Wishbringer这样“初级水平”的游戏，它们极力尝试着把互动作用推荐给年轻的玩家。这个游戏中的谜题太少也是一个值得关心的问题，事实上我们在最后一部分而不是结尾部分安排了Ryder的出现来增加谜题。

AMFV也将封面推向了技术的方向，成为第一个“Interactive Fiction Plus”型的游戏，它要求的是128K的内存而不是64K，这比现在Infocom的其他游戏要大两倍。



问：正如你所说的，这个游戏蕴含着强大的道德暗示。以前为什么没有想过做一个严肃的游戏？

答：我想是因为我真的是很喜欢做AMFV，我觉得电脑游戏和书本、电影、戏剧一样可以带有很多艺术成分。这些年我从AMFV的玩家那里也得到很多反馈。一些人跟我提到过他们因为玩这个游戏而进了电脑游戏行业。

不幸的是，即使AMFV对玩家们有很大的影响，也只有大约三万人去玩或买这个游戏，远不及其他游戏。毕竟以前我在Infocom所推崇的那种创造性的自由还没有得到实现。当游戏的预算升高到七位数，开发商们就会对那些尚未正式发布或处于试验阶段的东西毫无兴趣了。



问：其他媒介似乎允许严肃的作品与轻松有趣的作品并存，你为什么认为这不适用于电脑游戏呢？

答：问题在于电脑游戏产业不像其他的媒介那么容易赚钱。听说一部电影只要说了或做了某些事就不至于亏本，包括它的海外销售、音影制品等。而主要的电脑游戏市场都在亏本。所以我想在电影行业里制作那些试验性的或者稍微有些不同的东西会很困难，而在电脑游戏行业





就更困难了。我这个行业的负责人最怕做一些毁灭希望的事情。

另一个不同之处是影片可以做得廉价而又能附庸风雅，电脑游戏却不可能走类似的路线。有迹象表明，网络可能有这样的发展趋势。让人振奋的是电脑游戏开发者协会已经组织了相关的游戏节。但是电脑游戏行业能真正达到的还是远不及 Blair Witch Project 或者 Crying Game。因此，我认为不能有试验型的游戏，而且游戏开发商们又比其他行业的生产商要保守得多，他们的底线就是制造“安全游戏”。这些游戏不能太严肃而且含义要比较轻松。



问：你是不是认为网络传播可能会利于创造更严肃的作品？

答：如果传播渠道都合并起来，我想这种现象可能会发生，况且网络似乎是产生这种现象最好的赌注。实际上它不仅仅是销售，也有服务的一面。所有的杂志都绝不会忽略出版游戏产品。我会记得刊登在某个专栏里的一句小小的广告词。重温游戏的不是杂志而是只有三寸大的地方，仅仅因为这个专栏的作家曾提过很喜欢玩这个游戏。生产价值很低的游戏只是在网络上作为共享软件，这样听起来也很不错，我很喜欢这类游戏，所以会抢先把它下载下来。但电脑游戏杂志确实很少告诉人们游戏的种类。

当然，除了像这样传播信息之外，还有其他的方法向人们介绍游戏。你得有一套很好的传播系统，试想如果没人看到你的游戏，它又有什么用呢？如果真的能做到这样的合并，我们将能够做一个更廉价的游戏，可以把一些东西做在车库里，那样将更有意思，还可以用某些方法推动封面的发展，这比那些高生产成本的游戏要划算得多。而且你还可能创造出一种像 SimCity 那样的新型游戏。



问：你是怎样想出 Leather Goddesses of Phobos 来的？

答：这是一个很有趣的故事。Infocom 刚搬进剑桥办公室的时候还很不起眼，几个月后公司决定举办一场小型的啤酒宴会，招待公司的员工及顾问、董事会、零售商、广告公司和批发商之类的合作伙伴。这次聚会不算很正式，参加的人也不多，但看到 Infocom 的首次社会活动举办得如此成功，公司董事长 Joel Berez 表现得非常兴奋。聚会是在 Infocom 办公室中最大的一间举行的，这里既是会议室也是微机室，我们的 Apple II、Atari 800、TRS-80 都放在那里。房间里有一整面墙被当成了一块大黑板，上面永远都画着一个表格，记载着游戏发售的各种数字。就像：

	TRS-80 Model I	TRS-80 Model III	Apple II	Atari 800	IBM PC	DEC PDP-11
Zork I	42	42	44	45	42	45
Zork II	17	17	17	19	17	17
Deadline	31	31	29	30	33	33
Zork III	10	12	12	10	13	13
Starcross	28	29	30	28	28	31

也就是说，每当房间里收集到游戏的新版本，就会换成新的版本数字记录。当机器上出现新的版本数字，它马上就会被写到黑板上去。如果说 Apple II 里的 Zork III 用完了，我们就会去





订购最新的经过检验的版本，黑板上的数字也相应地会被更新。

言归正传，这之后不久我安静地走到黑板前面，在 *Leather Goddesses of Phobos* 下面画了一道线。纯属巧合地我选择了这个名字，可能有些不好意思但还不至于难堪。正当这时，Joel 看出来并赶在其他人到来之前把它擦掉了。然而这么多年来，这个名字没有变过，当人们想把一个不存在的游戏名字和一个句子联系起来的时候，自然就会想到 *Leather Goddesses of Phobos*。

1985 年，我一直在考虑围绕这个名字真正做出一个游戏，毕竟人们喜欢这个名字好多年了。我来开发这个游戏听起来确实有点讽刺的味道，这个想法马上得到了 Marc 和其他游戏作者的认可，连我以前的室友 Mike Dornbrook 也支持我，他是商业学校毕业后回到 Infocom 来开拓市场的。

我们花了更长的时间来说服上层管理部门，尤其是我们那个缺乏幽默感的 CEO（首席执行官）Al Vezza。他只对公司的商业生产感兴趣，喜欢做一些无聊的游戏。一年后，当 LGOP 快要完成，Infocom 的商业生产也快要破产了的时候，Infocom 得到了 Activision 的帮助。实践部部长 Jim Levy 对游戏和游戏开发略知一二，Al Vezza 带着他在办公室周围参观。当 LGOP 展现在他们眼前的时候，Al 有些紧张地说道：“当然，这不一定是最后的名字。” Jim 大笑：“什么？我还不愿意它叫别的名字呢！”这自然让每个人都感觉好受一点。不幸的是，事隔一年 Jim 就被 Activision 董事会的董事长解雇了。



问：你是怎么想到做 Zork Zero 的呢？

答：它只是我关于游戏前传的一些构思，但是我的同事们对这个构思都很感兴趣，故而称它为 Zork Zero。整个续集牢牢抓住电脑游戏产业，并表现出极大的趣味性。我写了 *Sorcerer*，第二个游戏 *Enchanter* 三部曲也被私下认为是 Zork V。它和 Zork 涉及的领域相同，我编写了 Zork 的历史、时间、地点、任务等因素，认真整理了 Zork 游戏和第一个 *Enchanter* 游戏，尽量尝试把这些复杂的地理环境和历史背景联系在一起，并写出了这部游戏的第一部分简概。刚开始，这些和原著作者的初衷有些冲突，但很快显现出有多么的重要——因为后来它变得十分流行。

当他们刚开始讨论 Zork Zero 的时候，我就对这个游戏的环境相当了解了。应该可以说我比原著的作者们还要更了解 Zork 的相关细节。这几年来，我一直想做一个像 Zork Zero 这样的游戏，如果不是因为当时我在做 *Planetfall* 续集、*Stationfall*，Zork 可能已经是我的作品了！而且那个时候 Brain Moriarty 决定在 Zork 领域进行研究，并已经完成了名叫 *Beyond Zork* 的 RPG-探险体系，这两个 Zork 游戏太过相似，所以只能取其一。

完成 *Stationfall* 之后，我又必须在 Zork Zero 和一个多年来的想法之间做出选择：一个发生在泰坦尼克号首航过程中的冒险故事。结果公司最后却认为“人们对泰坦的故事可不感兴趣”。事后在向许多开发商提出这个想法时我也无数次地听到此类说法。因此当电影 *Cameron* 上市并成为空前流行的影片时，我有一种哭笑不得的感觉。

公司一旦决定开发 Zork Zero，我做的第一件事就是坚持和元老们召开了一次集体研讨会。Marc Blank（后来离开 Infocom 去了 west coast）、Dave Lebling（现仍是 Infocom 的游戏作者）和 Tim Anderson（现仍是 Infocom 的“高级工程师”，专门从事游戏的程序设计）当时都在场，还包括已离开很久的第四位元老 Bruce Daniels。召开这个会议惟一的结果就是开始游戏前传的





工作，结束“West of a white house”。会议还为新游戏设计了一个基本框架：Dimwit 城堡的环境、Flathead 王朝灭亡的原因，以及收集属于 12 个 Flatheads 的史前古器物。

问：Zork Zero 是一个很奇怪的混合游戏：除了掺杂一些有图形的片断，剩下的绝大部分是文本。为什么会想出这种设计呢？

答：当时，Infocom 面临一些压力并极力寻找出路。这几年来我们的产品销量一直在下降。Infocom 实践管理部门的想法就是“无论我们做出什么样的 Infocom 游戏，都有成千上万的游戏迷去购买。所以我们的策略应该是生产尽量多的游戏。”1987 年我们制作了 8 部作品，而此前我们的年产量从未超出 5 个，而且销售也很糟糕。因此，1988 年的生产思路就变成“文字游戏已是穷途末路，我们的游戏应该有图形”。

Infocom 总是排斥图形化的冒险游戏，在 80 年代早期和中期这样做是对的。但是正如我们在图形化冒险游戏中看到的一样，早期的微指令可以很好地显示街机游戏风格的画面，而且画质非常好。后来苹果机出现了，它能显示比以前更好的黑白画面；接下来是 Amiga，它显示的彩色画面更漂亮。IBM 微机的图形显示卡也越来越好。为了盈利，制作图形化游戏而停止文字游戏的开发更有理由，但是 Infocom 认识到这一点稍微迟了一些。

1987 年末 1988 年初，为了更好地掌握和运用图形化设计，Infocom 的开发系统进行了全面的革新。同时，游戏的作者们也在冥思苦想怎么在游戏中运用图形化设计。一些人决定用图形来表明特殊的场景，比如 Dave Lebling 就在他的 IF 版本的 Shogun 中用图片来做特殊说明。

Zork Zero 的目的是想做成一个典型的解谜冒险游戏，那我首先要做的就是谜题环境中使用图形。于是我做了 5 个图形场景：谜、河内塔、跳桩游戏、卵石计数游戏以及一种叫 double fanucci 的卡片游戏。为了不让游戏的剩余部分落入老式文字冒险游戏的俗套，我又设计了三种不同的装饰界面：室外、建筑物内和地牢里。我给每个房间设计一个图标，且可以在屏幕上的图形地图上使用这些图标，这是一个很好的记忆存贮设备。最后，我使用了在图书馆看到的一本名为《Encyclopedia Frobozzica》的书中的图形说明，并以此图形说明为基础作为要编译的 Zork 游戏版本的总提纲，那时我正在制作 Sorcerer。

但这种图形化游戏还没有前一年的文字游戏卖的好，因此 Activision 决定在 1989 年中期撤销了 Infocom。

问：这些游戏根本没有提高销售量吗？

答：我想说的是 1987 年，所有的游戏销售量都在两万份左右，而 1988 年末 1989 年初的 4 部图形化游戏也达到了相同的销量。

问：那你认为是什么原因呢？当时 LucasArts 和 Sierra 在图形化冒险游戏上是非常成功的。

答：的确，当时 Sierra 的游戏卖出了几十万份。但是 Lucas 并非如此，他们觉得推出的游戏在技术上与 Sierra 几乎完全相同，在写法和内容上甚至更胜一筹，但仅仅卖出了 Sierra 的五分之一到三分之一。其实，这一点我也想不通。可能是 Sierra 制作的游戏很好地定位给了中等层次的玩家，而这个层次拥有最大范围的玩家。Lucas 的游戏类似 Infocom 的，复杂的情节导致它们只





能吸引少量玩家。



问：你为什么认为图形化游戏还不算真正的开始？

答：不，我更多的是在想图形化游戏为了追求形式上的图形冒险，不仅使图形化游戏而且也使动画图形游戏变得更加复杂，就像那个时期的 Sierra 和 Lucas 的游戏。Infocom 的游戏也只能算是带图形说明的文字冒险游戏。尽管使用了图形设计，我认为也不能体会这个时期 Infocom 所创作出来的动物图形。



问：如果 Infocom 对游戏的设计更完整，你是否认为它能够更成功的使用图形游戏？

答：如果 Infocom 在 1987 年宣布“我们将像 Lucas 那样模仿 Sierra 的冒险游戏平台”，很难说将会是什么样的结果。而且我总认为谁先抓住市场谁就能占有市场，这也是 Sierra 能凌驾于 Lucas 之上的原因。当然，有很多公司加入文字冒险游戏的行列，他们为此付出了很多努力，也确实做出了一些优秀的文字冒险游戏。比如，80 年代中期的 Synapse 软件公司就用他们的 BTZ 平台做出了很多不错的游戏，但他们最终没有发售。争当市场的领头羊确实很不容易，就算游戏同样优秀，也无法改变被抢先占领市场所造成的损失。另一方面，Infocom 这个名字的含义正好和冒险游戏相同，如果说有人能比 Sierra 抢先占领市场的话，那很可能是 Infocom，不过这很明显是从理论上讲的。



问：Infocom 公司制作的游戏使用的都是相同的故事介绍系统，各个游戏使用的几乎都是相同的游戏机制。你认为这样是否有效地共享了技术和设计方案？

答：当时这种工作方式非常适用。我们可以迅速地构思出游戏的整个线索，并在新电脑发行的几周之内制作出可以运行的游戏。这具有很高的商业价值，那时候的市场比较松散混乱，市场上总是会出现各种不同的平台，还有各种新的、不兼容的平台。例如，在 Macintosh 机器上曾一度有过 25 个游戏，其中的 15 个都是 Infocom 公司制作的。这使那些使用 Mac 机器的玩家非常恼火，因为我们没有使用 Mac 的 GUI。

此外，无论是包装还是游戏本身，我们制作的游戏看起来都有相似的外观。这就使游戏具有书本的感觉：Infocom 公司制作的游戏看起来都很相似，就像所有的书看起来都很相似一样。

但即使是现在，相同的引擎也可以在多个游戏中使用，尤其是在包括了扩展包的情况下更是如此。即使最终的产品看起来很相似，Infocom 的库中实际上也使用了 ZIL 引擎的多个不同版本。当“Interactive Fiction Plus”出现时，引擎进行了一次相当大的改进，以 AMFV 开始；然后在 1987 年左右制作出一款全新的、功能更为强大的解析器时，引擎再一次进行了重大的改进。当然，在 1988 年出现了图形之后，引擎又进行了一次全面的改进。



问：Infocom 公司在解析器上投入了很多精力，所制作的解析器被视为这个行业中制作得最好的解析器。它所达到的完美程度有没有达到你认为无法更加完美的水平？

答：当然，就在大约 1987 年发布新解析器的时候，虽然这个解析器可以得到改进，但我们





认为它所达到的水平完全可以满足游戏环境的需要。毕竟，我们无法理解所有的自然语言，而只能理解那些用得比较频繁的祈使句。我认为有可能继续改进的惟一领域就是与 NPC 对话的方面。但是制作 NPC 的主要问题看起来更加深奥，实际上并不是由于解析器的限制，而是因为如果我们要保持人物角色看起来不呆板，即便是保持一小会儿的话，也要在这个人物身上投入大量的精力。

无论是从玩家还是从游戏设计人员的角度来讲，我个人都非常喜欢基于文本的接口。但我并不在乎是阅读还是输入信息，而有些人并不喜欢其中的某种方式，或者两者都不喜欢，这样就会限制游戏玩家的范围，那些不需要阅读也不需要输入信息的游戏就会吸引很多的玩家。我发现基于解析器的输入接口是功能最强大的，而且也非常灵活，玩家可以尝试所有能想象到的事情，游戏的设计人员可以设计出各种各样的谜题，而这些是在“选中-点击”的图形接口中无法做到的。所以许多图形化的冒险游戏都只需要以各种可能的组合点击视野中的每个物体，而不是在谜题中思考。



问：有人批评解析器的接口总是显得很令人迷惑，而不是显得直观，对于这些批评你想说些什么？虽然玩家可能知道自己要做什么，但他们也有可能是在找关于某种动作的词汇时遇到了麻烦，对这个问题你是怎么看的呢？

答：我认为这都要归咎于并不完善的解析器。我记得自己曾玩过一个 Sierra 公司制作的游戏，屏幕上有一只我认为是马的动物，我在这匹马身上尝试了所有可以对马做的事情，而结果表明这是一只独角兽。在那些日子里，游戏的分辨率很低，因此我并没有注意到那个用于表示角的像素。所以在我说“上马”的时候，游戏中就给出“这里没有马”的信息，这时游戏本应提示我它有可能是一只独角兽。但游戏中没有出现这样的信息，却只显示出“你不能这样做”或者其他不具有帮助性的信息。因此对我来说，错误不在于游戏有解析器接口；错误在于游戏项目编写时并没有做好，或者没有进行全面的测试。

即使是玩最完美的 Infocom 出品的游戏，根据玩家个人情况的不同，他们也需要 1 分钟到半个小时的时间来学习语法。但对大多数人来说，一旦他们度过了最初的困惑阶段，解析器比较完善的游戏就不会令人迷惑。甚至在后来 Infocom 公司制作的游戏里，我们引入了一些新的事物，旨在使游戏最初体验的难度降低一些：注意玩家在那种模式中时所做的事情，建议他们转入正确的方向。如果玩家输入了错误类型的句子，比如问一个问题或者使用非祈使性的语气，游戏就可以捕捉到这些信息。如果玩家连续做了二三次同样的操作，游戏中就会说：“与游戏交谈的正确方式是”，然后举出一些例子。

我认为关于解析器接口真正关键的问题与输入信息无关，在输入的信息中可以使用自然语言。



问：你曾觉得受到 Infocom 开发系统的限制吗？

答：Infocom 公司的开发系统功能非常强大，而且非常灵活，可以相当容易地满足各种游戏的需要。惟一例外之处就是，在进行任何修改的时候都需要改动“解释器”。每个出售的游戏都包括游戏组件，这些组件是独立于机器的，解释器是各个机器专用的程序，它可以让游戏的





组件在专门的微电脑上运行。因为现在的解释器已经超过了 20 余种（用于 Apple II、Mac、DEC Rainbow、NEC PC-800 等等），对解释器进行修改时不只是修改一个程序，而是要修改 20 个以上的程序。所以这种情况很少发生，或者只有在非常重要的时候才会去做，比如修改 Deadline 中的状态线以显示时间，而不是显示分数和动作。

人们希望在尽可能多的机器上制作游戏，这样就产生了更为严格的限制，因此我们经常受到那些机器在能力方面的限制。以前，这种有限制的机器是 TRS-80 Model 1，这种机器的磁盘驱动器的容量将第一个游戏的可执行长度限制在 78K。当更旧的机器从待支持列表中消失时，这些限制也缓缓提高，但即使在我编写 HHGTTG 的时候，游戏仍限制在 110K 以内。一般来讲，通过测试，这种限制可以达到中等水平，然后对于在游戏中每次加入的内容、修正 bug 或者由测试人员处理合理的输入信息，都需要对某些文本进行更为痛苦的搜索，任何文本都要剪切或者精简。有时需要编写一些精炼中肯的文字。但这个过程经常变得很可怕，使我们感觉自己在屠杀自己的孩子。这有点儿夸张。



问：Infocom公司的开发过程是如何进行的？你在选择制作游戏的时候会相当自由吗？

答：以前，公司的工作相当不正规，所有的决定都是在非常不正规的一致表决中决定的。后来，尤其是在公司被 Activision 收购之后，所有的决定都由高级管理部门做出。一般来讲，游戏的选择取决于各个作者。像 Dave Lebling 和我这样有经验的作者就要比那些生手更有回旋余地。当然，其中也有市场方面的考虑，比如完成三部曲的强烈愿望，或者参与到像 HHGTTG 这样经过许可的游戏制作工作中的机会。

在这七年里有一件约定俗成的事情——我在 Infocom 时是在“制作人的午餐”上，或者简单地说是“Imp Lunches”。在每周的午餐会上，游戏作家们聚在一起讨论正在开发过程中的游戏，分享各自的想法，评价其他人的作品等等。这是每周中最有趣味的时刻。

在游戏开发的前几个月是不会有有什么疏漏的，制作人都是在独自工作，不参加 Imp Lunches，没有即兴发挥的思想，也不需要帮助/建议。但一旦游戏进入了测试阶段，首先要经过其他制作人的审阅，然后是内部测试人员，最后还要找到外部的“beta 阶段测试人员”，游戏需要经过连续数月的审查。在这个阶段中，会出现成千上万的 bug 和建议。



问：Infocom公司的游戏在设计上的流畅性和灵活性如何？

答：这个问题会根据不同的制作人而有所不同。我自己的风格就是在开始设计之前，首先在纸面上进行设计，大多是制作游戏的环境和“背景世界”的文档，比如 Sorcerer 中的时间线索，或者 Planetfall 中沙漠化行星上语言的规则。但是对于大多数游戏，我会立刻开始编码，创做出我在脑海中想象出来的人物和谜题。

与那个时候我曾用过的基于图形的系统相比，Infocom 的开发系统相当棒，因为游戏设计人员自己就可以在几天之内制作出游戏的整体部分，然后可以测试它的运行状况，看看它的运行效果。如果因为游戏无法运行而需要返工，也不会对时间和资源造成很大的浪费。这样可以重新做很多工作，并且可以重新编写游戏的大部分内容，这种事情在现在是无法想象的，因为这样的一个决定会扔掉已经完成的价值数十万美元的图形制作工作。





问：在 Infocom 公司制作游戏的过程中，是不是有很多的测试工作？

答：要进行很多的测试。因为在 Infocom 制作游戏的过程中，开发系统是非常稳定的，所以测试工作就可以集中在游戏特有的 bug 和游戏的内容上。在 alpha 阶段之前，最好是有两个星期的测试阶段；在这个阶段中，其他的游戏设计人员会来玩这个游戏，然后就要让我们自己的测试人员进行二至三个月的 alpha 测试，随后是让一些外部的志愿者进行为期一个月的 beta 测试。如果时间允许的话，还要有一个月的“gamma”测试，这个测试与 beta 测试类似，不同的是除非发现非常重大的问题，否则不会对游戏进行修改。

对游戏特有的 bug 和游戏内容的测试几乎是同时进行的，在早期的测试中要倾向于测试游戏的内容，在测试的后期倾向于游戏中的 bug，这个时候不会对游戏的内容做出重大修改。

在游戏的开发过程中，早期的测试阶段可能是最有趣、也是最令人激动的时候。一方面，在长达数月的单独工作之后，根本就不知道游戏是否比自己在直觉上的感觉要好，突然有一群人来玩这个游戏，通常也会很喜欢它，然后他们就会给出很多的反馈信息。这正是一种冲击。此外，我们还提供了自动记录脚本的功能，网络可以自动记录每个玩家的会话，这样我就可以了解每个人在游戏中的各个地方进行过哪些尝试，所以我经常可以发现出错的情况，但测试人员并不需要意识到这些错误。如果发现他们尝试过的某些行动是解决谜题的合理尝试，我会用一个巧妙的反应来奖励这种尝试，或者给出一个提示，而不会只是提供这样一个默认的消息——“不能把桌布放在那里”。

在测试阶段，游戏会变得越来越棒。进入测试阶段之后，游戏就像是一副骨架，拿我们的一个测试人员的话来说，测试阶段就是要“把肉加到骨头上”。这个阶段中有很多有趣的事，对古怪的输入信息的反应、难度的微妙程度、去掉不合理的谜题——这些都是 Infocom 公司那些优秀的测试人员的工作成果。



问：Infocom 公司制作的游戏在包装上很独特。为什么公司在这方面比其他的游戏开发商做得都好？

答：当 Infocom 公司成立的时候，电脑游戏包装的标准有点类似于 Ziploc 袋。它仅仅是一个明亮的塑料袋，顶部有一个 Ziploc 袋，还有一个洞，可以挂在商店的墙钉上；袋子里可以放一张软盘，而且经常也会有一个比较便宜的影印手册。实际上，Zork 的早期 Radio Shack 版本就放在这样的一个包装盒里。

Zork I 原来的开发商是 California 的一家名为 Personal Software 的公司。实际上，Personal Software 公司中制作 Zork 时的产品经理是 Mitch Kapor，他后来创办了 Lotus 公司。就在 Personal Software 开始发行 Zork 之后，他们编写出了一个名为 Visicalc 的程序，这是在电脑上使用的第一个成功的商业软件。他们将公司的名称从 Personal Software 改为 Visicorp，他们不想继续在游戏上浪费时间，所以就把 Zork 还给了 Infocom。

Infocom 没有去找新的开发商，公司决定自己承担开发工作，并且雇佣一个代理商来设计包装。结果 Zork I 和 Zork II 都使用了“硬质泡沫塑料衬垫包装”，这也是第一次将这种包装用于电脑游戏。这种包装将透明的模型塑料贴在一个纸板的后面，通过透明的塑料可以看到盒子上的内容；在这种情况下，就无法看到它后面的带有 Zork 手册的磁盘。





当需要为 Infocom 公司制作的第三个游戏(Deadline)设计包装时, Marc Blank 带着一些 1930 年由 Dennis Wheatley 编著的绝版书去找代理商, 书中使用了 Murder Off Miami 和 Who Killed Robert Prentiss? 这样的名字, 包含有很多报告和线索, 就像是调查案件时警方的侦探: 与目击者的交谈、打印信件、手写笔记、火车票、报纸剪辑、用过的火柴杆, 还有很多其他的東西。玩家就是侦探, 在筛选了证据之后, 玩家判断谁是凶手, 他是怎样做案的, 然后打开书中密封的部分看自己的判断是否正确。

Marc 在制作 Deadline 的时候受到这些书的很大影响——实际上, 原来的标题是 Who Killed Marshall Robner? ——他想要代理商在制作 Deadline 的包装时受到这些主题启发。Marc 想要让玩家从打开包装的时候起就感觉自己已经是案件中的侦探了。此外, 因为游戏的容量有严格的限制, 所以在包装上加入实验室的报告和嫌疑人交谈内容可以腾出空间, 以便在游戏中加入更多的互动性内容。Deadline 的包装使人可以很容易地想起 Dennis Wheatley 的书, 上面有犯罪场景的照片、谈话笔录、指纹、在尸体旁找到的实验室化验分析, 甚至还有一包药丸, 上面标有“在尸体旁发现的药丸”。那些东西实际上是白色的 SweetARTS。

Deadline 的包装获得了很大成功, 我们又将价格提高了 10 美元, 即将 MSRP 从 40 美元提高到 50 美元。我们认为这个游戏做得很好玩、很超值, 也提高了竞争能力, 使其他的竞争者很难争夺我们的市场空间; 而且更重要的是, 这也是打击盗版游戏的一种方式。复制软件包和软盘的难度大, 而且也不划算。此外, 因为游戏的包装很简洁, 而且与玩游戏的经验紧密相连, 所以许多人都感觉除非购买了完整的游戏软件, 否则游戏就是不完整的。

随后的游戏是 Zork III 和 Starcross。为了适应这种搭配方式, Zork III 放在了硬质泡沫塑料衬垫包装盒中; 而 Starcross 放在一个很大的塑料飞盘里, 附带了一张飞船附近的行星地图, 看起来很醒目时髦。近来, 在 eBay 的网站上, 这种原版包装的 Starcross 版本售价为 3000 美元。

在我所制作过的游戏中, 我最喜欢的包装是 LGOP, 它具有“scratch 'n' sniff”卡片和 3D 连环画。这些连环画是我这个连环画作家和另一个人协作的结果, 他很擅长于将 2D 图画转换为 3D 效果。对于“scratch 'n' sniff”卡片, 我从制作气味的公司中得到了很多实例。每种气味都涂在一张卡片上, 并在卡片上写上气味的名字。然后我将 Infocom 公司的雇员请来, 把他们的眼睛蒙上, 让他们刮开每个卡片来辨别。这样, 我就可以选出在包装盒上使用的 7 种最容易辨别的气味。看着各种不同的气味使人做出的反应很有意思, 有一个人在闻了樟脑球卡片之后, 脸上露出无奈的笑容: “我祖母的阁楼”。

作为制作人员, 我们在包装的元素上具有很大的自由度, 只要不超出预算就可以。市场部门也经常会展想出好主意, 他们会建议我将 Zork Zero 中的一本书改为一个日历, 并在 Lurking Horror 的包装盒上加上爬行的橡皮虫这样的东西。但是大多数好主意都是游戏的作者们想出来的。

最好的包装都是在游戏制作的一开始设计出来的, 而不是在游戏进入了 alpha 阶段之后才想出来的。有许多其他的公司都在包装上加入了反盗版版权保护, 但是这些措施都很明显, 而且也很让人扫兴, 比如“请输入手册上第 91 页的第 7 个词”。而 Infocom 公司所制作的包装绝不会让人感觉到这种反盗版的行动, 因为包装盒上的元素与游戏情节紧密相连。当然, 在完全是文本的游戏环境中, 包装盒上的元素也向玩游戏的经历中加入了图形的部分。





问：在Infocom公司与其他公司制作的游戏之间有很明显的区别，尤其在质量的一致性方面更是如此。为什么会这样呢？质量问题是如何保证的呢？

答：我认为其中部分原因是因为这是 Infocom 公司的一贯原则，即使是在 Infocom 公司成立之前，当公司正在制作 Zork 的时候就要承接一个有趣的游戏——Adventure，而且做得更好。所以大家都有很强烈的愿望要把游戏制作得最好。此外，还有一部分原因是因为 Infocom 公司的员工都是一些非常聪明和有天赋的人。另外还有运气。以前我们曾经获得过成功，所以当制作每个新游戏的时候，我们都可以投入大量的时间和资金，游戏的销售量与投入的资金是成正比的，而其他许多的公司都无法预计销售量，因此无法承担与我们的投资规模相同的资金投入。

我们一直在改进开发环境、分析器等，这也是使游戏能够具有高质量的另一个重要原因。有才干的测试人员，在测试、修正 bug 和常规的改进方面所投入的时间也是一个很重要的因素。



问：Infocom公司所制作游戏的稳定质量是不是使公司相当容易地度过了80年代中期的经济危机？

答：80年代中期的经济危机是从电玩开始的，然后就波及到了 PC 市场。许多公司都同时涉足于电玩和微电脑 SKU 的领域，而 Infocom 却完全置身于 PC 市场。此外，我们的游戏都是尽可能地不去模仿电玩。80年代中期的大萧条之所以没有对 Infocom 公司的游戏销售产生过多影响，其中另一个原因就是我们所制作的游戏可以在许多机器上玩，我们可以迅速地在新发布的机器上制作游戏。例如，Mac 机早在 1985 年就面世了，我们的游戏在早期的 Mac 机上取得了巨大成功。当然，由于我们制作的游戏质量较高，而任何萧条期中最先消失的总是那些劣质的游戏。



问：对我来说，看起来只有 Infocom 公司在 80 年代早期制作的游戏还没有过时。你认为这是什么原因呢？

答：在 80 年代早期制作的游戏，图形看起来是很糟糕的，但文本仍旧是文本。所以文字冒险类游戏总是不显得过时。而且我们也已经讨论过，我们制作的游戏质量较高，所以可以使游戏免于很快地过时。此外，一旦解决了技术上的陈旧性，对于一个很有创意的作品来说，十至二十年的时间并不长。看看与游戏同期出现的书籍、电影和电视节目就可以知道了。只有一些明显具有时事性的作品看起来好像过时了，而 Infocom 公司制作的游戏当然不是时事性的，不过 AMFV 是个例外。人们在这些作品出现很长时间之后仍会去欣赏它们：I Love Lucy 是 40 年前的作品，Gone With the Wind 是 60 年前的作品，Charlie Chaplin 和 Buster Keaton 的电影是 80 年前的作品，Alice in Wonderland 是 150 年前的作品，而 Shakespeare 的戏剧已经有 400 年的历史了。



问：Infocom的游戏制作人员认为文字冒险可以一直用下去吗？

答：我们当然认为它们会有所改进，无论是在可以预见和不可预见的方面，都是如此。而每个人的想法都是不同的，可以说在 1985 年时人们对于这个问题的思考是这样的：图形可以加





以改进，以添加到冒险类游戏中，游戏的重点将会更加侧重于故事中的谜题，游戏和游戏世界将会变得更大，其中将会有更多的现实性，在冒险类游戏中将会出现令人相信的人物，还有许多其他媒体中成功的介绍故事的人，比如小说作家和电影导演，这样就更加趋向于将冒险类游戏做为将来进行故事介绍的媒体。回忆起来，在这些部分中只有第一个部分得以实现了。

尽管游戏中会出现可以预见的改动，但是我想，每个人都会认为冒险类游戏从某种形式上来说具有不确定性。我并不认为每个人都会觉得在本世纪结束的时候，所有形式的冒险类游戏都会做为一种商业化游戏种类而消失。



问：有趣的是，书籍看起来可以与电视和电影同时存在。为什么你认为文字冒险类游戏做不到这一点呢？

答：文字冒险类游戏仍旧有相当广阔的市场空间。现在仍有很多人在写这类游戏，也有许多人在玩，这不仅仅是一个符合经济规律的市场。编写这些游戏的人并不是为了钱才来写游戏的，他们这样做只是出于兴趣，玩这些游戏的人也大多不用付钱。我认为在编写文字冒险故事和写书之间有一定的相似性，那就是这个过程都趋向于个人的创作，假设你使用的是一个现有的文字冒险写作系统。一个没有受过专业训练的人可以在开始写作之后的几个月内就写出一个文字冒险游戏，只要有一个打字机、字处理程序或者一摞纸和一只钢笔就可以开始写作了。



问：可能这只是个规模的问题，就如同你以前所说的那样。喜欢玩电脑游戏的人数比喜欢其他传统、非交互式媒体的人要少得多。

答：虽然不知道具体的人数，但我认为情况的确如此。我想一本最热销的书也不会超过100万的发行量。我记得在我们制作一款游戏的时候，Hitchhiker的书总共卖出了700万册，那么他的每一本书都有可能会超过100万吗？看电视的人数当然也比这个数字要多出好几倍。



问：Spellcasting系列游戏的界面非常有趣。人们可以像Infocom的文字冒险游戏一样玩这些游戏，然后添加一些功能，使玩家可以通过使用鼠标点击一系列动词、名词的方式来玩游戏。制作这种新接口的想法是什么？

答：这个界面来自Legend的民间故事，在Infocom公司完成了Arthur之后，Bob Bates开始为The Abyss制作这个接口，这是根据当时还来发行的一个电影而来的。当Infocom公司被Activision关闭的时候，这个游戏项目被取消了，并且这时Legend决定开发他们自己的冒险游戏，所以他们就继续开发那个接口。

制作这个接口的原动力并不是因为这是一种特殊的感觉，不是因为这是玩冒险游戏时的一个好的/有用的/友好的/聪明的接口，而是因为人们感觉到冒险游戏在衰落，玩家需要在玩游戏的时候一直都能看到图片，他们讨厌敲字。我从来都不喜欢这种接口。图像部分相当棒，通过双击图片中的门就可以来回移动，或者双击就可以拾起物体。但是出于很多原因，我对菜单并不感兴趣。第一点，使用菜单比用手敲击键盘输入内容更加麻烦和耗时。第二点，菜单泄露了很多信息，因为其中会显示所有可能的动词和可见的物体。第三点，在实现游戏的时候要去做许多





额外的工作，而得到的回报却很少。第四点，这些菜单排除掉了任何不会引用的不可见对象。



图 10.1 Spellcasting 101: Sorcerers Get All the Girls

问：此外，Spellcasting这个游戏因为提供了全屏图形而优于Zork Zero。你是否因此改变了自己编写设计游戏的方式呢？

答：不至于这样。我认为自己可以对在 Infocom 制作的所有图形较少的游戏进行处理，让一个美工在每个房间都制作一些图形，然后翻新到 Legend 的图形引擎上。菜单驱动接口比图形更容易出现问题。相反，我为 Legend 制作的所有游戏都有一个热键，使用它可以关闭图形，把游戏当做旧式的文字冒险游戏来玩。所以图形只是一个附加的部分，并不是必要的组成部分。

问：在游戏可玩性的整体体验方面，在向文字冒险类游戏中添加了图形之后，你觉得有什么得失呢？

答：有人认为我们根据文字描述在想像中制作出来的图片，根本没有那些在高分辨率、带有上百万种色彩的显示器上显示出来的图像更为生动，而人们对于这个问题正在进行无休止的、热情洋溢和非常严谨认真的讨论。我自己的感觉就是，可能有人会在想像中创作出最好的图像，而有些人的想像力却很差劲。尽管如此，改进使冒险类游戏向着更庸俗、更加缺乏品味的方向发展。

其次，加入图形就会去



图 10.2 Spellcasting 201: The Sorcerer's Appliance





除掉一些谜题。例如，谜题要依赖于对某些事物的描述，让玩家通过检查和试验来找到答案。举一个 Zork I 中的例子：没有充气的橡皮阀不能称为橡皮阀，要称为“一堆塑料”。玩家必须对橡皮阀进行检查，找到阀门，然后想办法使用气泵充气，这时候玩家才会发现这是一个橡皮阀。而在图形化的游戏中，玩家可以立即看到这是一个没有充气的橡皮阀。

第三点，也是最重要的，制作图形要比制作文本投入更多的成本。就像 Brian Moriarty 所说的那样，“在图形化的冒险游戏中，游戏必须显示所有的信息——但是要显示所有的信息就会有很大的开销！”因此，图形游戏所能显示的信息较少，但更重要的是，图形游戏所提供的谜题解决方案、通过游戏的其他路线、在解决谜题时对合理但并不正确的尝试所做出的响应，还有对各种动作有趣的响应都减少了。换句话说，图形冒险类游戏比 Infocom 制作的文字冒险类游戏缺乏很多实质性内容。这样，你就会得到很多令人不快的反响，比如“你不能那样做”或者是你在游戏中的角色/化身会向你耸耸肩膀。



问：Superhero League of Hoboken 是如何制作出来的？你只是想暂时使用那种风格吗？

答：很多年来我就想制作一款 RPG 游戏；在 90 年代初的时候，RPG 是比较畅销的冒险游戏。直到 Diablo 的发行才结束了 RPG 游戏的全盛时期。但是我认为“Tolkien-esque”奇怪的设置和 RPG 游戏中人物的服饰注定要失败，我突然想起来超级英雄是另一种非常棒的风格，很适合于 RPG 游戏，而且可以使用超级能力代替魔法咒语。

我原本计划将其制作为一款纯粹的 RPG 游戏，但 Legend 所制作过的游戏都是纯粹的冒险游戏，他们感到很紧张，所以惟一可以说服他们的方式就是制作一款 RPG/冒险混合型的游戏。



图 10.3 Superhero League of Hoboken



问：这是我所知道并不算可怕的惟一一款超级英雄游戏。你认为现在市面上的超级英雄游戏如此之少是什么原因？

答：我认为市面上缺少超级英雄游戏主要是因为法律/授权的问题。大多数公司可能认为只有著名的超级英雄中的其中一个才值得作为游戏的题材，但很难拿到这样的版权。即使获得了许可证，获得授权所需的开销也很大，这就意味着在开发预算中的资金会大大减少，这也就是为什么经过授权的游戏寥寥无几的原因，而不仅仅是超级英雄游戏。我可以使 Superhero League 中原来的内容获得授权，因为它是一个讽刺作品。但我并不认为自己可以说服 Legend 以相同的风格和引擎制作一款“纯粹的”超级英雄游戏。



问: Superhero League是你制作的惟一一款RPG游戏。是什么使你想在RPG游戏的方向上继续尝试游戏设计的呢?

答: 我仍旧很喜欢玩RPG游戏, 如果有人制作了我喜欢玩的游戏, 那么我就总是想尝试自己去制作这些游戏。我总是喜欢做一些以前从未做过的事情, 而无论游戏采用这里所提到的新风格, 或者使用像AMFV这样严肃的主题, 或者从类似于Hitchhiker's这样的作品中进行改编, 或者采用像Zork Zero这样更大的规模。当然, 这只是我自己的偏好。开发商们却总是有其他的想法。



问: 看起来, 该游戏可以自动为玩家做许多事情, 而其他的RPG游戏需要玩家自己去做这些事情。你的设计目的之一是否就是使RPG元素便于玩家的控制呢?

答: 因为这是一个冒险/RPG混合类型的游戏, 所以我们猜想肯定会有很多RPG游戏的玩家不太熟悉冒险类游戏, 而许多冒险类游戏的玩家不太熟悉RPG游戏。由于RPG游戏交互过程非常详细的特点, 所以我就尽自己所能地让谜题显得比较简单, 使界面保持尽可能地简单和友好。



问: Superhero League of Hoboken这个游戏非常流行, 所以我很奇怪从那之后你为什么没有制作另一款RPG游戏?

答: 是这样, 那款游戏的销售情况并不像看起来的那样好。我认为它的销售量不会超过20000到25000套。这当然很令人失望, 因为我在这个游戏上投入了大量的时间, 比我在Legend公司制作的任何一款游戏投入的时间都长。游戏得到了好评, 但销售情况却很令人失望。当时负责发布游戏的是Accolade, 但那时Legend的财政状况并不是很好, 所以在市场方面的工作的确没有做好。Accolade公司的角色是开发商, 而不是发行商, 他们的工作是处理所有的广告和PR, 所以没有能力兼顾多方面的问题。而且作为开发商, Accolade在实力上也不如像EA这样的大公司。

我认为对Superhero League的销售情况造成影响的一个原因就是, 从最开始的发行日期起这个游戏就耽搁了大约一年之久。其中部分原因是由于Legend公司制作的前一个游戏的延期, 部分原因要归咎于游戏中的某些内容无法在Legend的开发系统中完成, 这就需要一些额外的支持。他们雇了一个程序员来做这些工作, 但是他在工作上却有些消极, 因此程序工作不得不由内部人员来返工。这就造成了游戏的延误, 因此游戏最终只能在1995年中期发行, 而不是1994年中期。这是一款普通的VGA游戏, 而与此同时, 所有的游戏都已经升级为Super VGA。因此, 游戏发行的时候它就已经非常过时了。实际上, 我记得与这个游戏同时发行的还有Colonization。而且我记得自己在玩Colonization的时候对游戏糟糕的外观印象很深。我敢肯定, 人们在第一次玩Hoboken的时候也会有同样的感觉。



问: 那么你想再制作一款RPG游戏吗?

答: 我在GameFX制作的很多项目都是角色扮演类游戏, 但不可否认的是, 这些游戏最终都没有完成。我对制作Hoboken很有兴趣, 而且我也喜欢玩角色扮演类游戏, 所以自然也愿意





继续制作另一款 RPG 游戏。



问: Hodj 'n' Podj是你所制作的最与众不同的游戏。你是不是想让这个游戏吸引一些新的玩家呢?

答:我并不是想吸引一些新玩家。与我其他的设计方案一样,游戏的玩家基本上都是我自已。我总是希望其他人也能像我一样喜欢我所设计的游戏,这样的话,游戏就可以取得成功。

设计 Hodj 'n' Podj 是我五年前的想法,而最终这个想法变成了一个实际的项目。我在设计这款游戏的时候曾经想过要把那些有趣、简单的游戏都组合在一起,因为对于那些可以推动开发过程中所做出决定的专业级玩家来说,这样简单的游戏是不会让他们满足的。这当然是在经典的游戏可以通过 CD-ROM “游戏包” 随处获得之前的情况,而现在要通过 Internet 才能获得。

那时,我觉得这样的一些游戏需要将某种结构关系联系在一起,以制作出容易让人接受的经济型游戏包,这样就可以把价格过高的棋盘类游戏与神话故事主题的游戏放在一起。当然,与此同时,许多公司发行的游戏包并没有相互关联的主题或者结构,而销售情况也很不错。尽管如此,我还是很愿意将 Hodj 'n' Podj 这种小游戏作为较大游戏的一部分来制作。

就在游戏顺利进入开发阶段之后,我才开始考虑游戏是否可以吸引一些不同的玩家。随后,“临时玩家(casual gamers)”这个词就进入了游戏行业的术语中。作为外部测试人员,雇员的朋友和家人开始玩游戏的早期版本,我们很惊奇地发现,即使是那些不喜欢玩电脑游戏的人也愿意玩这个游戏。我们感到很



图 10.4 Superhero League of Hoboken



图 10.5 Hodj 'n' Podj



高兴，并且发现游戏也吸引了许多女性玩家。而且最终我们发觉，游戏在一个很小的环境中很受欢迎，而那时我们还没有认识到这个小环境的存在——那就是“家庭游戏”：也就是父母和孩子一起玩游戏。这要感谢游戏中不同的难度设置机制，父母可以和孩子在相对较难的关卡一起玩游戏，而不必“降低”到孩子的水平。这是惟一一款我制作过的可以用来自己取乐的游戏，现在我还时常和孩子们一起玩。



问：The Space Bar这个项目是如何开始的，在这个项目中你的设计目标是什么？

答：这是另外一个设计概念，我已经酝酿很久了。我认为这个想法的起源要追溯到1986年或1987年，那时因为我们公司的客户新闻简报称为New York Times，所以New York Times（《纽约时报》）威胁说要控告Infocom公司。虽然我们的律师已经完全消除了这个威胁，但是当Activision开始协商要收购Infocom公司的时候，他们坚持要我们驱散所有这样的“阴云”，因此我们不得不修改新闻简报的名称。客户中引发了命名上的争议，公司内部也出现了很多意见，最终新闻简报命名为The Status Line。但与此同时，我还建议开办The Space Bar，然后把正在编著的小说加入到新闻简报中；这个小说是由酒吧里的一些人编著的，并且小说中描述的人物都是酒吧里的“常客”。我并不知道这是为什么，但是从某种程度上来讲，新闻简报中的想法慢慢变成了游戏的概念。

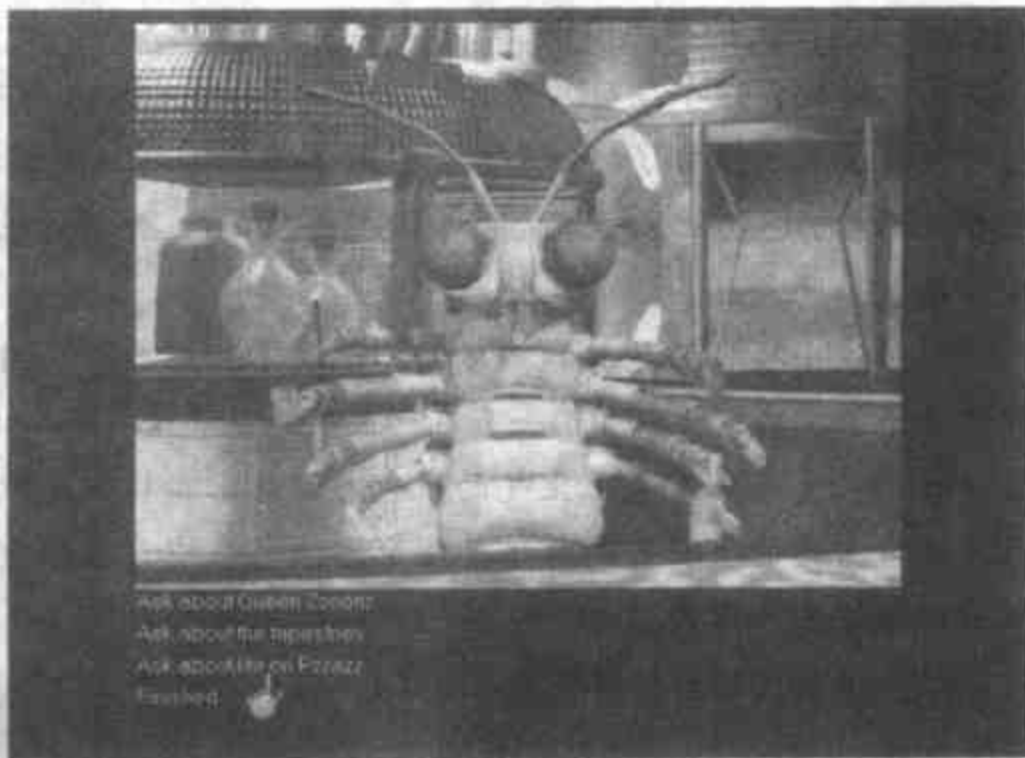


图 10.6 The Space Bar

这个主要设计目的是制作一款由许多小冒险游戏组成的冒险游戏：一部小说与简短的故事集相比，就如同常规的冒险游戏与The Space Bar的关系一样。除了想要尝试一些新鲜事物的渴望以外，我还感觉到（这也反映了我在游戏中的需要）人们越来越缺少时间，制作冒险类游戏需要腾出大量的时间。但如果你对自己说，我只是现在玩这一“章”，而其他的部分以后再做，这样就可以更容易地说服自己开始设计游戏。设计游戏时的第二个目标就是制作一个像电影Star Wars（《星球大战》）中那样引人注目的太空船发射降落场，制作一个黑暗的Bogart-esque气氛；这样做的确很好玩，并且可以证明自己能够制作一款类似于Infocom文字冒险游戏的图形冒险游戏，游戏中还可以有很多“吸引人的内容”。就像Hodj 'n' Podj一样，我觉得一系列独立的游戏过于松散，需要一个连贯的线索，因此原故事中包含了Alias Node对变形者（shape-shifter）的搜索——Ni'Dopal。Empathy Telepathy就是一种便利的方案，应可以将“简短的故事”与原故事衔接起来。





在项目一开始的时候, Rocket Science 对于使用其他的媒体来“协助推动”项目特别感兴趣:如电影、动作人物、棋盘类游戏和书籍等等。我建议将指定由 SF 作家编著短篇故事选集的工作作为 The Space Bar 的配套项目, 每个作者选择一种我们为 The Space Bar 创造的角色/种族, 然后编写一个关于这个种族或角色的原创故事。这并不是常规的编著游戏中小说的项目, 但的确很有趣。但是, 尽管他们最初很有热情, 而且我们也一直在游说, 但 Rocket Science 并没有做出任何举动。

问: 如果我错了的话就请你更正, 但看起来 The Space Bar 是你预算最大的项目。你是不是很想获得这样大的产品价值?

答: 是的, 这个项目的预算已经超过了 Hodj 'n' Podj 项目预算的两倍, Hodj 'n' Podj 已经是当时预算最高的项目了。但是与当时的其他图形冒险类游戏相比, 它的预算还是相对比较低的。Boffo 公司的运作入不敷出, 受到 Rocket Science 公司的很多指责, 我们的转包商 Dub Media 的反应也是如此。

虽然这个项目的预算非常庞大, 但也不算是浪费, 因为我们的项目经常是在没有足够资金的情况下来完成, 所以我们一直在偷工减料。举这样一个例子: Alias 的 PDA 原本要制作成一副生动的面孔, 而不是毫无生气的声音。因此, 在我们想做的事情与有可能做的事情中, 财政状况是项目考虑的重要方面。这是图形冒险游戏中存在的重要问题, 也是这种媒体此时无法渡过财政难关的主要原因。

这个项目虽然在预算方面显得非常紧张, 但仍不失为一个好游戏。与 Ron Cobb 这样的概念型艺术家一起工作, 是我的职业生涯中非常难忘的事情。The Space Bar 的开发组也是我所指导过的最大的开发组, 当然, 与这个开发组共同工作占去了预算中的大部分; 让这么多人一同工作也是相当让人激动的, 因为几乎所有人都可以在他的专长/职责以外有所作为。尽管我们是在偷工减料, 但却达到了每一个设计目标, 这一点是很令人满意的。

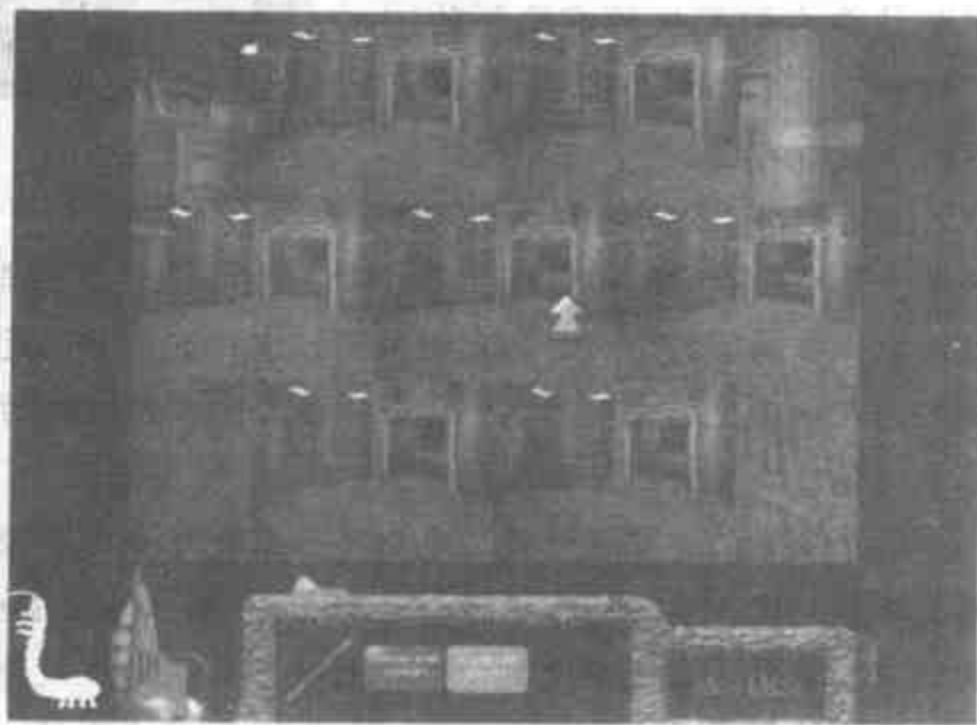


图 10.7 The Space Bar

问: 你是怎么去了 WorldWinner.com 的呢?

答: 在 GameFX/THQ 取消项目的一年之后, 我在等机会离开这家公司, 这时我与一个招聘人员一同工作, 她建议我去 WorldWinner。Hodj 'n' Podj 项目中的各个游戏都值得回味, 这无疑也是吸引我的主要原因之一。此外, 在多人在线的环境中工作也具有很大的诱惑力, 因为我以前从未在这种环境中工作过。





问：你认为与现在能够利用的途径相比，Internet提供了一种新的可能性，使你可以制作更多的游戏吗？

答：是的，就提供更加个性化或者更具试验性的游戏来说，我当然是这样认为的。抛开这些不谈，就目前来说，带宽的确存在一些缺点。对于我现在正在制作的游戏，虽然的确都很有趣，也的确很吸引人，但如果回到100K都可以称为大数字的时期也是很烦的事情，而且在某些情况下，这个数字也的确显得很大。当开始使用CD-ROM的时候，我就已经消除了那种感觉，游戏的容量在大多数情况下已经变得完全无关紧要，而现在又突然重新回到了那个时候。但从整体上来讲，其中当然既有优点也有缺点，整体上的优点是充满希望的，而缺点就是某些游戏由于带宽的问题导致我们无法制作——但人们可以通过普通的、传统的渠道来制作这些游戏。



问：你觉得在写游戏和玩游戏之间，哪一种工作更有乐趣呢？

答：当然是玩游戏。写游戏的工作有时候会很好玩，有时候是个苦差事，有时也会非常痛苦，就好像你的公司要破产一样。但玩游戏的过程总是很有趣的。当然，制作游戏过程中最有趣的部分比玩游戏过程中最有趣的部分乐趣更多。



问：在游戏中要写作这么多的内容真是可怕，你认为在编写游戏的过程中，哪些东西是首先要记住的呢？

答：所有类型的写作都是不同的，有很多出色的小说家根本不会编写电影剧本，反之亦然。而编写游戏与这两种情况都不一样。当然，其中也有例外情况。如果能成为一个游戏玩家，则会对编写游戏有很大帮助。如果一个小说家从来看过电影的话，就不能指望他成为一个电影剧作家！有许多游戏著作都很糟糕，就是因为这些作品就好像是为其他媒体编写的那样。当然，某些作品是因为其作者并不具有这方面的天赋而做得不好。和游戏设计一样，程序员和制作人都错误地感觉自己也能从事写作工作。

游戏的写作之所以会如此与众不同，其中一个原因就是游戏很少会有不连贯的部分，一个词或一个句子就可以对玩家的各种动作做出反应。在保持这种细节的趣味性和保持简洁性之间会有不同的权衡方案。此外，出于游戏可玩性和提供谜题的目的，游戏的写作必须进行谨慎的构思——去掉足够数量的线索，不要太多，不要误导——而作品的质量经常要退而居其次。游戏非线性的特点是游戏著作成为优秀作品的另一个障碍。如果你不知道Line A和Line B哪个将会首先出现，那么通常就要有信息的副本，显得不够衔接或者过分啰嗦。最后，就是重复的问题。在冒险游戏中，我们可以看到相同的部分会一次又一次地出现，这种内容上的相似性使得玩家会轻视此类游戏作品。



问：在你的游戏中，设计过程的灵活程度如何？加入图形是否对你在游戏中做出的改动产生了限制，程度如何？

答：非常灵活，但你是对的，一旦经过了最开始的设计阶段之后，图形游戏对游戏的修改产生了很大限制。在我制作的所有游戏中，AMFV可能是在制作过程中修改次数最多的游戏。





这个游戏非常雄心勃勃，很少有面向故事的成分，几乎是一个“未来模拟器”，玩家可以在现在设置参数，然后旅行到N年后的将来，可以看到这些决定在将来会产生什么结果。

我还认为，在实现过程中，当游戏的情节逐渐展开时，开发工作进行得最好，而不是在首先制订/策划游戏到一个相当详细的程度，然后再开始进入实现过程。这也是文字冒险类游戏优于图形冒险类游戏的一点。在 LGOP 或者 Hoboken 这样的游戏中，当游戏仍处于画板上的阶段时，我可以找到或者改进游戏的表这方式/风格，这样可以制作出更好、风格更加统一的作品。



问：冒险类游戏最大的问题可能就是难度问题。例如，如果游戏太难的话，就有可能吓跑玩家。但如果游戏过于简单，那些专业级玩家就不会继续支持你的游戏。对于这个问题，你想出解决方法了吗？

答：游戏的难度是一个很常见的问题。我们制作的游戏一直都很简单，但似乎并没有吸引到新的玩家，而又必定会使专业级玩家失望。可以考虑使用提示手册，或者在游戏中提供提示信息，这样既可以保持游戏的难度，又不会让那些新手、不够熟练而且缺少疯狂热情的玩家丧失信心。这是一个相当不错的解决方法，因为如果游戏难度过高，提示信息可以提供帮助，这样，游戏对于较弱的玩家来说就是一种比较不错的体验；但是如果游戏过于简单，就会使比较熟练的玩家大失所望。另一个解决方法就是设置多个难度级别，在比较容易的级别上加入多一些的线索提示，但很明显，这就需要在设计、编程和谐调方面做更多的工作。



问：关于冒险类游戏，玩家最经常抱怨的就是这些游戏没有重新再玩一次的价值。作为一个游戏设计人员，你要在游戏中加入什么内容才能使游戏值得玩家重新玩过呢，或者说你是不是并不认为这是大问题？

答：是的，当与 Civilization 之类的战略游戏和 WarCraft 之类的实时游戏相比，而不是与其他冒险类游戏和 RPG 游戏相比时，这就变成了一个大问题。在某种程度上，冒险类游戏也有一定的可重玩性。例如，Suspended 就是非常值得重新玩过的 Infocom 出品的游戏，因为即使是在最低的级别上要玩过游戏也是很难的。而对于 Zork I 这个游戏，我记得 New Jersey 的一对夫妇曾经写信告诉我以最少的移动赢得游戏的新方法。而使用一些其他的谜题方案，并使用“一些生动的内容”对游戏中古怪的输入做出反应也是延长游戏时间的方式。但对于大多数游戏来说，让玩家花费 30 到 40 小时的时间来玩游戏，或者让玩家花 40 到 50 美元来购买游戏就足够了。



问：你曾想过要忘记谜题，然后将游戏重心大部分转移到故事上吗？在 Hodj 'n' Podj 中，你似乎已经想出了“所有的谜题”。

答：我的愿望就是在游戏中加入更多的故事情节，减少谜题，我认为这也适合于大多数的冒险类游戏作家；但是如果要达到 30 至 40 小时的游戏时间目标，谜题还是必不可少的。在我制作的所有游戏中，AMFV 是故事情节最多的，而 Zork Zero 是谜题最多的。我当然并不认为 Hodj 'n' Podj 已经列出了所有的谜题，因为棋盘类游戏有非常不错的开端和结局，在这些书档之间加入一些游戏情节：王子救公主、王子对抗他的兄弟等等。



问：你是否曾经向游戏中加入谜题，而目的只是延长游戏持续的时间？

答：我的确只是因为要延长游戏时间而向其中加入谜题。我曾说过，AMFV 的整个第三部分就是这种情况，部分原因是担心游戏难度过高而且缺少谜题，如果游戏中不加入一些谜题的话，玩家就会感到厌烦而不会继续玩游戏。我认为 Planetfall 和 Stationfall 就是属于这样的游戏，当游戏进入测试阶段时，大家对游戏的印象是游戏过于简单，而且结束的过快。游戏中需要加入一些内容，而不要让玩家在 10 个小时之内就玩完游戏，也不要让他们觉得自己花的钱不值。



问：你是否担心过会有人喜欢冒险类游戏中的故事情节，但又被非常难的谜题吓倒的情况？

答：这是个很难解决的问题，因为冒险类游戏中吸引玩家而且令人着迷的正是那些难题，这些难题可以让玩家熬夜去解决，即便不是坐着玩游戏的时候也会去思考那些问题。然后，当玩家不在玩游戏、但正在思考问题的时候，突然就会想出一个主意：“天哪，我怎么没有想到呢，我还从来没有试过！”这时，玩家就会迫不及待的冲回家，调出保存的游戏，然后去拿到金橘，把它带回来重新试过。它可能会发挥作用，这当然是最棒的感觉，也有可能根本没用，这种感觉很糟糕，虽然它可能没什么用处，但至少可以向玩家指出新的方向或者新的提示。在没有谜题或者谜题相当容易的游戏中，玩家就不会得到这种感觉。但另一方面，尤其是过了一段时间之后，好像就会有越来越多的人来玩冒险类游戏，而这些人也的确不喜欢太难的谜题。这些玩家的需要是很难满足的。设计人员试图满足那些喜欢游戏没有难度的玩家，但一段时间之后，玩家却不愿意再玩这样的游戏，因为游戏给人的体验已经不再具有吸引力。



问：在你制作的所有游戏中，你是否也做程序员的工作呢？

答：在制作 Hoboken 这个项目的过程中，我既要负责设计又要负责编程，从那以后就只负责设计工作。如果有可能的话，我还是希望不参与编程工作；那样做真是太痛苦了。当然，这样做在效率方面有很大的好处，而且可以在程序员和设计人员之间实现 100% 的全面交流。即使我喜欢编程工作，现在的游戏对一个程序员来说也太过复杂，所以我根本无法自己完成所有的设计和编程工作。



问：在冒险类游戏、尤其是文字冒险类游戏中，限制玩家的行为是游戏的主要部分。如果玩家经常看到“你不能那样做”这种字样，他们就会非常沮丧。你在解决这个问题的时候有多大难度？

答：这个问题的其中一个解决方案就是限制游戏的地理范围。原来的设置选择可以有所帮助。这就是很多游戏都有明确定义的边界范围的原因，比如笼子、城堡、岛屿、飞艇等等。如果不让玩家感觉到边界的存在，他们的感觉就会更好一些，这样的设计比告诉玩家“那个方向没有什么有趣的东西”或者“如果要通过沙漠的话，你可能会渴死”之类的话要好。

另一种解决方法就是卷起袖子，根据玩家最初的猜想加入尽可能非默认的反应，考虑测试人员的建议，或者通过阅读测试人员测试游戏的手稿而获得更多的好主意。加入这样的反应只受时间的限制，而且更多的情况下也受磁盘空间的限制。这也是加入提示的好方法；玩家可以





尝试一些并非“正确答案”的事情，但这些事情都是“合理的”。我会使游戏所做出的反应成为失败的解释，但可能是要尝试动作的线索。

例如：

>把三明治给那个老人

他看起来很累，以至于无法马上吃这块三明治。

还有一种解决方法是让默认的反应尽可能地灵活有趣。例如，在 Hitchhiker's 中，FILL 这个动词的默认反应就是“Phil 是谁？”。Phil 是 Zaphod 在聚会时的别名。举另一个例子，在 Zork I 中，对于许多“不可能的”行为的默认反应都要从一个表中选择，这样就可以做出不同的反应。所以玩家不会得到以下信息：

>所有的都拿走吧。

面包片：拿走。

刀：牢牢地插在面板上。

面板：你不能拿走它。

水槽：你不能拿走它。

火炉：你不能拿走它。

烤箱：你不能拿走它。

玩家会得到这样的信息：

>所有的都拿走吧。

面包片：拿走。

刀：牢牢地插在面板上。

面板：好一个主意。

水槽：再想想。

火炉：或许不太血腥。

烤箱：再想想。



问：在制作一款新游戏的时候，你是否有一个特殊的起点呢？

答：根据游戏的不同会有所不同。AMFV 的制作是从游戏的主题/信息开始的。Sorcerer 从复杂的时间旅行开始，解释自己的迷惑，并且从那里展开故事情节。关于 Planetfall 和 The Space Bar 最初的想法，我以前曾经解释过。一般来说，在转向下一个工作之前，我不会去做所有的事情。我不会编写整个游戏的情节，然后开始制作游戏格局，再构思那些谜题。相反，我会构思出故事的线索，然后设计游戏格局的核心部分，加入人物和谜题，提炼故事线索，在最终的格局中添加新的剧目，与此同时加入两个谜题，再加入两个人物，向游戏中的实验室部分添加一些房间，再添加一个谜题，以便充实最终的游戏，想象出 Esmerelda 为什么会首先从家中逃走等等。



问：你认为冒险类游戏目前如此不景气是什么原因？

答：简单地说，冒险类游戏的收支模式无法盈利，几乎没有哪个开发商愿意接手，因为几乎所有在开发问题上的决定都要从纯粹商业化的角度做出，而不是根据游戏的创造性。冒险类游戏是制作成本最高的游戏，排行榜前N名冒险类游戏的销售额要比其他所有类型前N名游戏的销售额少得多。

当然，可以说冒险类游戏并没有彻底失败，而是发展成为动作/冒险类游戏，例如 Tomb Raider 和 Mario、Crash 这样的平台游戏。从个人角度来讲，我并不认为冒险类游戏一定要在某种程度上依赖于手眼协调。

我认为重要的技术革新可以重新让这种游戏得到玩家的欢迎，但我并不知道是否要加入声音识别接口、通过 Turing 测试的 NPC 和 3D 环绕 VR 环境等。



问：当每个人都认为某个投资充足的游戏做得很不错，但是最终的销售量却不好时，这种事总是会让人感到特别痛苦。尤其是 Grim Fandango，也属于这种情况。

答：是的，Grim Fandango 也是如此。虽然并不知道确切的数字，但我认为它的销售量不会超过 10 万。而且那个游戏还是人们当年在冒险类游戏中的一致选择。那是个很棒的游戏。从谜题的角度来看，我并不认为这个游戏有多好，但是从艺术的角度来看，它可能是我见过的最好的冒险类游戏。



问：令人感到奇怪的是，冒险类游戏曾经是销售情况很不错的游戏，而现在的销售情况却非常差。可能我掌握的数字不是很准……

答：不，实际情况就是这样。当 King's Quest 这些游戏在 80 年代末 90 年代初推出的时候，这些游戏在当时很畅销。而且 Infocom 的冒险类游戏从大约 1983 年到 1985 年也是很畅销的。我们曾经一度有 5 个游戏进入了销售量排行榜的前十名。



问：那么冒险类游戏的玩家反应如何？

答：现在的某些游戏风格那时没有，而那时候存在的某些游戏风格却产生了变化。因此，人们玩过了一些游戏之后就开始喜欢交互式的体验，但如果他们可以把游戏拿到手的话，他们会去玩那些现在比较流行的游戏。在 1985 年的时候根本没有第一人称射击类游戏，也没有即时战略游戏。

现在仍会有一些玩家喜欢玩冒险类游戏，只是其中的一些人不会再以相同的程度支持这些游戏了。比起 15 年前用 20 万美元制作的冒险类游戏，现在花费 200 万美元制作的游戏需要 10 倍的玩家人数来支持。现在的市场规模可能已经扩大了一倍，但是绝不会达到原来 10 倍的规模。因此游戏的利润已经降到了临界值以下，无法获得经济效益。





问：是什么使你对游戏着迷了这么长时间？你曾考虑过写一部小说或者在其他非交互性的媒体中从事写作吗？

答：我经常考虑要写一部小说或者电影剧本，尤其是在我的游戏写作生涯比较低迷的时候：项目被取消、公司破产、游戏的销售状况差。但游戏写作总是可以赚到钱，所以我将其他的写作工作作为第二职业；由于要教育子女和其他外部原因，所以我不会有很多时间来从事免费的写作。制作游戏过程中的任何挫折和不快都完全来自于工作方面；我一向认为制作游戏过程中有创造性的过程都是一种突破。这仍是一个不断发展的行业，所以，发明一种新的“艺术形式”是相当令人激动的。因为这个行业中的薪水相对较低，所以与你一起工作的每个人都很主动，他们热爱自己从事的工作；就赚钱生活来讲，这是一种很不错的方式。比如说，有多少父亲可以送给他们的孩子印有已经取消了的 WarCraft 冒险类游戏的 T 恤衫呢？

Steve Meretzky 作品一览

Planetfall, 1983

Sorcerer, 1984

The Hitchhiker's Guide to the Galaxy, 1984

A Mind Forever Voyaging, 1985

Leather Goddesses of Phobos, 1986

Stationfall, 1987

Zork Zero: The Revenge of Megaboz, 1988

Spellcasting 101: Sorcerers Get All the Girls, 1990

Spellcasting 201: The Sorcerer's Appliance, 1991

Spellcasting 301: Spring Break, 1992

Leather Goddesses of Phobos II: Gas Pump Girls Meet the

Pulsating Inconvenience from Planet X, 1992

Superhero League of Hoboken, 1994

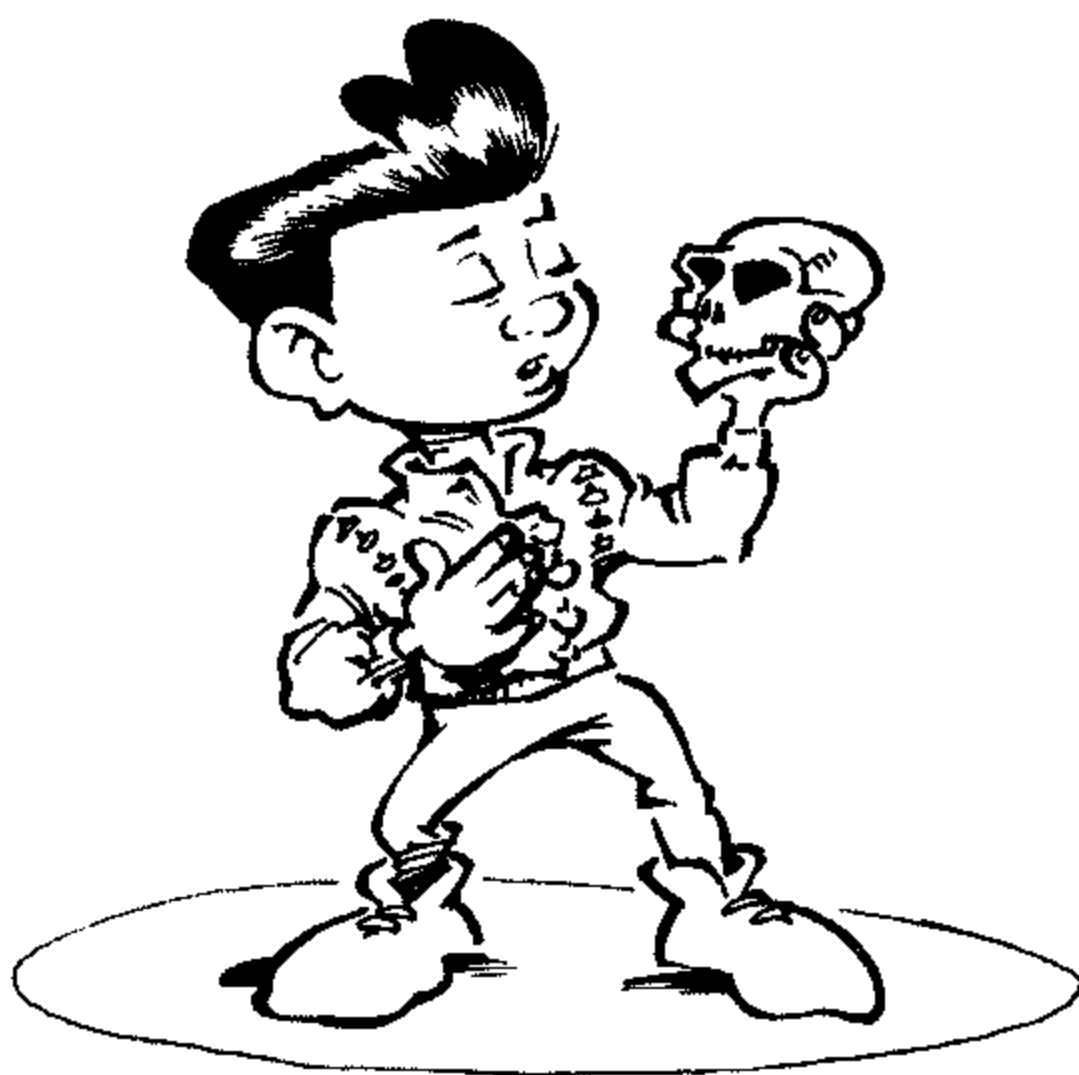
Hodj 'n' Podj, 1995

The Space Bar, 1997



Chapter 11

第 11 章 故事介绍



“对设计者来说，他们陷入到自己设计的故事中是危险的。他们忘了故事介绍是一个线性的叙述类的事情。你的故事越生动，互动性越高，则游戏中的人物级别就越高。有点像‘噢，结果已经定了，那关键点是什么？’”

——Eugene Jarvis

严格地说，电脑游戏不需要讲故事。几年来，已经有大量的精彩游戏几乎不提供故事介绍。例如 Tetris 根本就没有故事。再如 Centipede 和 San Francisco Rush 中，仅能在游戏初始时了解故事简介；但如 Marathon、Command & Conquer 和 Thief 这类游戏却讲述了一个故事，并可作为游戏剧本的关键部分。设置的故事情节如此丰富多彩，比没有故事情节的游戏更让玩家沉迷于游戏世界。还有一些游戏，如 A Mind Forever Voyaging、Myst 和 Ultima 系列游戏中，故事和



游戏是完全一致的，否则这些游戏几乎是无法想像的。所以游戏的确不需要故事情节，但当使用恰当时，故事情节却能使游戏变得更有吸引力。



11.1 故事情节对于 Myst 游戏是非常重要的，如果没有它，就很难想像这个游戏的存在

实际上，将我拉入游戏开发中的首要原因就是交互式故事梦想。想像在小说中，用故事情节来迷住读者，让她关心故事中的人物，改变她对世界的理解，在一些特殊情况下改变她的生活方式。现在，如果让读者自己成为故事中的主要人物，而不是仅观看别人的活动，并且玩家还能通过选择来决定故事发展的形式、方向和结局，那将是一个多么迷人的游戏！玩家在不同的情形中通过交互进行不同的选择，就能够看到不同的故事情节。因为是玩家的选择决定了故事情节，所以交互故事比传统故事更能诱发玩家的兴趣。故事情节具有多种可能性。当然，要达到这个目标仍然需要走一段长长的路，目前还没有一个游戏能接近这个理想。但是它确实能吸引玩家，所以设计者还需要继续实验，期待着真正的交互故事诞生的那一天。



11.1 设计人员的故事与玩家的故事

当我们谈论游戏故事时，我们的意思是什么？许多游戏开发者把游戏故事看成是一连串已有的戏剧性事件，非常像小说或电影中的一个故事。在游戏世界里，不管玩家的动作怎样，这些事件都是固定不变的，在游戏部分之间，将故事代表性传递给玩家。例如，在 *Command & Conquer* 中，告诉玩家的是一个具有不同使命的 GDI 和 Nod 双方力量间的冲突故事。故事部分地决定了使命发生的地点和玩家必须做的事，但典型特点是一旦玩家完成一关，故事只有一个发展方向。故事仅有的可能结局是成功或失败，当玩家完成所有预先设置好的目标之后，就是成功的结局；如何任何一处没有完成，就是失败的结局。一些游戏在其故事线中允许有一些简单的分支，但每一个分支仍然是由游戏设计者预先设计好的，通常这些分支在范围上是相当受



限制的。

但还有一种完全不同类型的游戏故事。如果我上面描述的类型称为设计者故事，我们可以将下面要说的称为玩家故事。再看 *Command & Conquer* 的例子，每次玩家玩游戏，他会玩出与以往不同的新故事。确实，每个关卡都编辑了关于玩家如何输赢的一个小故事。例如，玩家以 GDI 一方开始他的游戏，建立大量的速射轻机枪步兵团、手榴弹步兵团和 Humm-Vees。但在敌人的火焰喷射器步兵团对玩家来说太强大的时候，一次早期的 Nod 进攻几乎能消灭这些军队力量。然而，玩家能在附近探到一些资源并建立一个先进的武器加工厂和一些军营。然后，玩家集中建立火箭步兵团并制造巨大的坦克。当 Nod 火焰喷射器步兵团再来进攻时，玩家就可以很容易地用坦克将敌人击退。大量步兵团开始撤退，玩家的坦克和火箭步兵团追击到敌方的基地。在那里，GDI 步兵团可以从远距离炮轰 Nod 设施，用巨大的坦克击退反击。从而，玩家赢了这一关，这就是玩家故事。

现在，当许多游戏设计者谈到游戏中的故事介绍时，他们几乎不可能谈论以上所述的玩家故事。然而，人们发现玩家故事在游戏中是最重要的故事，因为它是玩家参与最多、玩家的决定最有影响的一种故事。在多数情况下，一旦玩家击败过运用狡猾战略的关卡，他将对有命令、全动态影像 (FMV) 的设计者故事更缺乏兴趣，这种故事在各关卡间发生，并解释将要玩到的下一关卡。当然，设计者故事有某种优势。它能包括有趣的角色和境遇，并能使用传统的讲故事策略，例如构造复杂故事，制造紧张气氛，预示某种可能等等。遗憾的是，这些策略的使用常常牺牲了故事的互动性。另一方面，依赖于玩家如何玩游戏，上面提到的 *Command & Conquer* 玩家故事可能没有太多戏剧性和叙述性所特有的紧张，因此，作为一个讲故事的经验，可能是有点偏离。

互动式故事介绍的理论是把设计者故事和玩家故事合成一体，所以，在保留故事戏剧性的同时，玩家能真正影响故事。互动式理论讲故事的经验有两个好例子。Chris Crawford 喜欢使用第一个例子：父母给孩子讲故事。父母的头脑中已有故事中将会有的角色、惊奇之处、故事大概将如何展开和如何结局等构想。但当孩子问些关于故事的问题时，父母将因此而改变传说。父母可以用一本指导书，但在必要时会放弃它。例如，故事开始：“当一个公主在漆黑的森林里迷路时，她看到许多不同的东西，包括大的蝶螈、一个黑洞和一个旧室，她感到非常害怕。”当父母讲故事时，孩子可能会问：“蝶螈是什么颜色的？”“蝶螈是一种奇异的深黄色，这种颜色公主只在皇家调料的芥菜中见到过。”“那洞呢？”“从洞里飘出一种可怕的气味，让人想起硫磺燃烧的气味。”“可能在里面有个老巫师，在制药。她走进去了吗？”“她走进洞里，为了避免在黑夜里摔跤，她小心地迈着每一步。当她进入洞中很深的地方时，她开始看到一束光线，一个声音喊到。‘是谁进入了我的洞穴？’当她走得更近时，公主看到一个衣裳褴褛的老巫师”……也许实际上在父母最初想讲这个故事的时候，故事中没有这个巫师，但当孩子问问题的时候，父母没有像电脑游戏中设计的那样，简单地回答说：“你不能走那条路”或“关于它没什么特别的地方”，而是根据孩子的提问改编了故事，加上一些细节并介绍一些必要的新角色和境遇。全部故事和它的主要角色可能没改变太多，但孩子扮演了一个决定故事实际发展情况的真正角色。

发现另一个真正的互动式讲故事的例子是在许多角色扮演游戏中，例如 *Dungeons & Dragons*。在 D&D 的游戏中，地牢主人 (DM) 领导这个游戏，通过游戏世界指导其他玩家，当





故事发生时向他们讲这个故事。地牢主人事先策划玩家将要探索的场所、在哪个场所里玩家将会遇到的一些角色的想法和可能知道将要发生的一些主要冲突。然而，玩家控制他们调查关卡的哪个部分并处理他们所遇到的不同的 NPC。例如，DM 可能没有不同的 NPC 当其靠近时将会说什么的脚本。相反，他知道这些 NPC 的个性和可能的反应。当玩家问 NPC 一个问题时，DM 会有合理的反应。一个聪明的 DM 决不会说，“NPC 不明白你的问题。”就像家长给孩子讲故事一样，DM 将可能使玩家始终在他想讲的全部故事之中；同时，在故事如何展开，甚至可能如何结局方面给玩家大量的自由。

当然，根据如上所述的故事创作一个互动式故事的电脑脚本，其问题是要求人去讲故事，因为现代电脑将不可能像人一样动态地跟踪故事的发展。所以，一个游戏设计者当前能做的是试着去重新创作一个互动式故事。但是，在产生故事线的动态场所里，要预测到玩家可能会问的所有问题、他想去的所有地方和他想说的所有对白。当然，这是一个巨大的任务，无论设计者有多少预测，她将不可能想周全每一个玩家想尝试的事情。至少，设计者必须试着允许不同的玩游戏的风格和调查游戏环境的不同关卡，而不是把玩家归入玩游戏和探索其故事的惟一路子。如果设计者确实对互动式故事感兴趣的话，设计者故事的适应性足够使其变成玩家故事就成为了她的责任。



11.2 故事介绍的位置

游戏可以用许多种方式讲述故事。通常，游戏使用许多不同的讲故事设置去传达他们的故事，不同的游戏比其他的游戏依赖更多的设置。你希望讲故事的类型、你想用到的技术和你的游戏可玩性将决定哪种游戏设置对你的游戏最佳。

最简单的区别方法是看故事发生的前后关系：

- 游戏以外：当游戏运行但实际上玩家没玩游戏的时候，这是在电脑上运行任何故事都要做的事情。这包括片头动画，在此时，玩家没有控制其角色的能力，例如，在 *Command & Conquer* 或者 *Super Mario 64* 里非互动部分的关卡间发生的片头动画或使命的摘要。
- 在游戏中：逻辑上，这是上面的相反方面，是当玩家实际上正在玩游戏时发生的故事。这包括游戏世界的背景、玩家对手的行为、玩家可能有的任何动态对话和玩家可以做的任何互动预使命。
- 外部材料：包括电脑以外所做的全部故事介绍，例如在指南或随游戏附赠的任何物品中写的引言，例如一张地图或一些收藏的珍宝。

一个特定的游戏可以用以上提到的故事介绍类型中的一种或三种。*Half-Life* 是包括戏内讲故事的游戏的一个例子；玩家从游戏开始到结束从不失去对其角色的控制权。*The Infocom games* 是用戏内和客观材料讲故事的游戏中的一个好例子。除了玩家在游戏本身中的对话和游戏世界的描述，*The Infocom games* 总是与附加的文档和小玩意儿（有助于提高玩家对游戏世界的理解）一起出现，除有时要求完成游戏之谜以外。*Command & Conquer* 通过它的设置和使命设计使用戏内讲故事，多数故事线是通过戏外、非互动的片头动画传达的。*Tekken* 是一个无实体的、几乎全部通过戏外片头动画讲故事的游戏例子。玩家在击败一个具有特殊个性的角色后播放动画





的游戏。尽管他们的战斗方式常与他们的国籍有关，但不同的舞台设置与故事线无任何关系，角色自己也没在场景中展示个性。实际上，为什么 Tekken 的设计者感觉是被迫包括故事线的原因还是不清楚的。也许他们想给玩家一些东西，以奖励他们在游戏中获胜；片头动画是他们能想像的仅有的最合适的奖励。

11.2.1 游戏以外

在游戏中，戏外讲故事也许是现在最流行的形式，它以各种各样的形式流行起来。戏外讲故事的流行归功于它与其他媒体讲故事的相似性。例如，片头动画非常像一个电影，并应用了已建立的影片技术，而关卡的文本摘要很像在小说里读到的。这两种媒体类型比电脑游戏流行了更多年，并且都有使它们能非常有效讲故事的已有的语法。在某种程度上，通过这些方法比通过游戏脚本讲述故事更容易。但作为一个设计者，你必须问你自己，非互动的片头动画是希望游戏干什么？如果你的游戏可玩性好，玩家将想返回去玩而不是坐着等待长长的片头动画。玩家为了互动才玩游戏。如果他们想要更多被动经历的话，他们将会去电影院或从图书馆里借一本书。非互动的讲故事在游戏中可以有它的位置，但设计者需要了解的是，它必须补充和不降低精彩的游戏经历。

与我已论述的一样，在游戏外讲述故事有许多不同的方法。主要的方法概述如下：

- 片头动画：一般地，这是当片头动画用影片技术向玩家叙述时所谈到的。这可以发生在 2D 或 3D 中，常包括切换、摇镜头、180 度规则和能够在电影或电视中看到的熟悉的其他设置。
- 文本：许多游戏用文本去描述一个故事或告诉玩家将到来的使命目标。文本可以全屏显示，必要时轻击显示另一屏，或以能让玩家有足够时间阅读的慢速滚屏显示。
- 图像：有时用一些简单图像向玩家传达故事线索的某些部分。这不受标准片头动画精确的限制，因为它们不包括镜头切换或其他影片技术，尽管一个简单的镜头全景可能用于扫过一张不适合用于屏幕的图像。这个图像可能是一个地区的地图、一个正建立的即将到来的挑战的图像，或玩家已经完成部分的扼要重述。图像常与文本混合排列，有时用漫画书的技术但通常没有圈起人物的讲话。
- 音频：有时提供给玩家一些语言对话或其他音频的指示。当视频对话不存在，或者当对话播放时玩家正看其他信息（例如地图，人物档案或其他文档）时，通常使用音频。

在制作片头动画时，最重要的目标之一是，在片头动画和游戏之间建立一个始终如一的视觉外观。如果可能的话，片头动画与游戏静止部分使用的引擎要一样。在 90 年代中期，游戏转换成以 CD-ROM 作为选择的发布媒介，游戏首次能包括实际的视频录像重放装置，即使这些电影常常不能全屏显示。从而形成恐怖的视频游戏，例如 The 7th Guest 这类游戏用迷你游戏呈现长视频剪辑，导致产品不像游戏更像电影。在这些游戏中，玩家的大部分时间实际上不是花在游戏上，而是花在观看非互动的片头动画上，这些影片片断通常很业余，其水平低于最低劣的电视节目。这说明了为什么玩家很快不喜欢这个流派的原因。另一类游戏，例如前述的 Command & Conquer 和 Dark Forces，在制成实际游戏的关卡间使用视频部分。实际上包括可行的强制游戏，包括视频却能以游戏名义站出来，可见，这些游戏是足够幸运的。然而，这些游





戏的视频部分是用真的演员创作的,看起来不像游戏世界里发生的。其他游戏,例如 MechWarrior 2 和我自己的 Centipede 3D, 用极难的多边形、3D 环境去处理这些片头动画, 创作一个看起来不像用于游戏操控的实时 3D 引擎产生的环境。结果对玩家来说是一个杂乱的视觉体验, 这种体验破坏了玩家可能建立的任何半信半疑的感觉。事实上, 片头动画的使用对玩家来说已经是非常刺激的体验; 这一分钟玩家在过程中扮演一个积极的角色, 下一分钟他不得不被动停下来去观看。所用的片头动画看起来不像只有恶化事件的游戏世界。

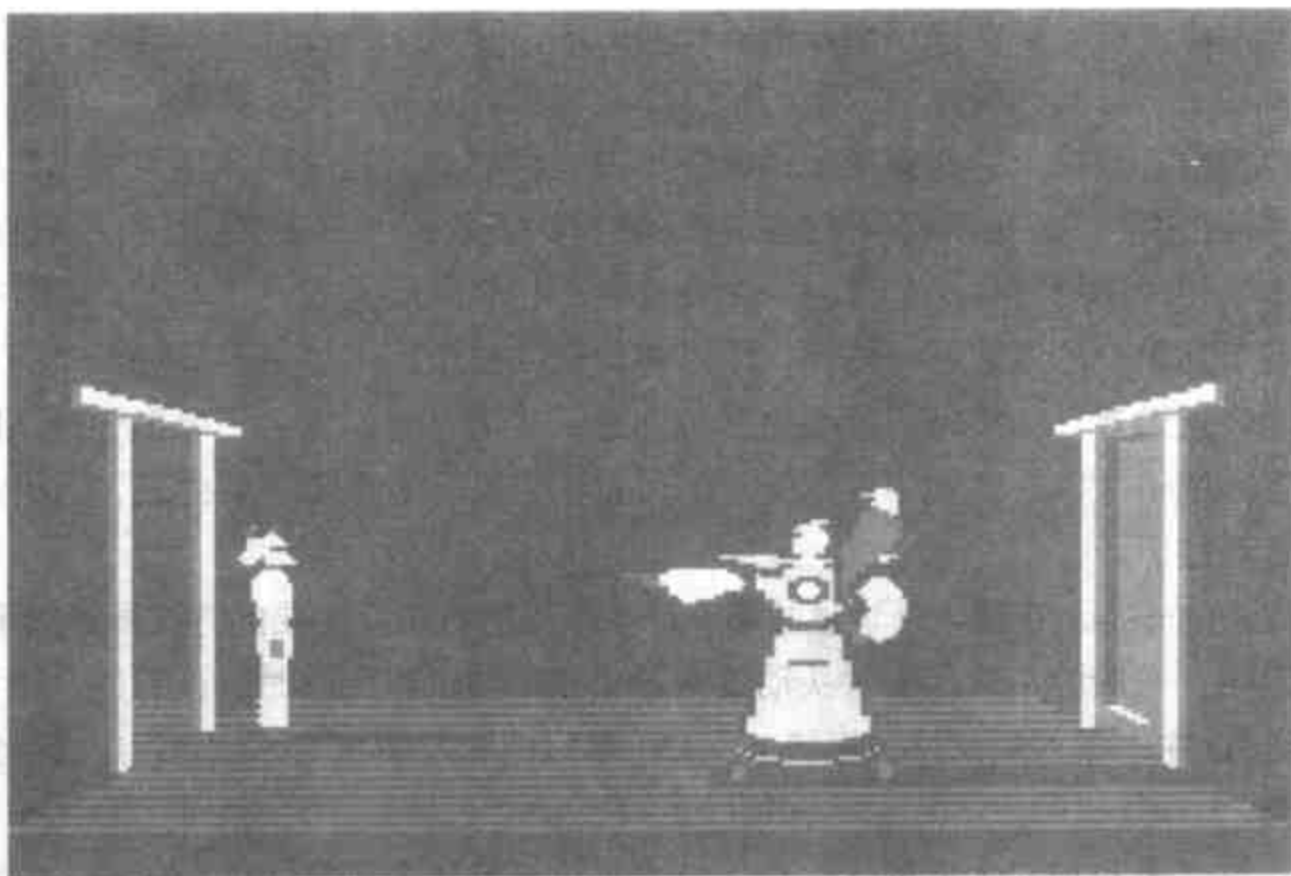


图 11.2 在 Karateka 里的片头动画是用游戏引擎控制的为玩家创造一个无伤痕的视觉体验

许多游戏已经成功使用片头动画。这种片头动画使用与游戏内的视觉一样的图形, 从 2D 游戏 (如 Pac-Man 和 Karateka) 一直到现代 RT3D (Legend of Zelda: Ocarina of Time and Drakan: Order of the Flame)。在这些游戏中, 虽然玩家可能失去游戏主要的控制权, 但至少玩家有一个完美的视觉体验。艺术家可能抱怨片头动画看起来不好, 毕竟, 他们仅能处理用于实时播放的多边形数量。但是, 游戏从头到尾的连贯性可以弥补在视觉质量方面可能丢失的东西。

在许多电脑游戏中, 片头动画的另一个奇特方面是其非互动的本质, 这表示设计者不了解计算机是一个可具有互动功能的装置。考虑两方面情况, 一方面是在电影院或赛场的观众, 或者观看电视转播的全国性的观众。这些产品的观众不能用任何方式影响进程: 演出开始然后结束。另一方面, 一个人能以他想要的任意速度来读书、看录像或听故事。书中的页可以重读, 录像带可以倒退或快进, 且一个孩子可以让他的父母进一步解释或重读他不能理解的部分。关键的不同点是, 前者的非互动媒体的观众是一大组人, 而后者是个体。

电脑游戏的观众是一组人还是个体呢? 显然, 对多个玩家游戏, 观众可以多于一个, 但多个玩家游戏几乎从不被任一种片头动画所困扰。不对, 需要片头动画的讲故事游戏几乎都设计成单一玩家。那为什么当一个游戏的主要使命以文本形式滚屏出现时, 使用者不能转回它? 甚至, 为什么它一直在滚动? 计算机是一个优秀的工具, 让使用者控制其运动, 并且由于玩家通





常自己玩游戏，所以如果当她使用滚动条或键盘上的箭头来控制速度并以自己的速度读文本，谁会介意呢？对于片头动画也有相似之处：为什么使用者几乎从不能转回再次观看片头动画呢？倘若她忘记她想听的那部分故事或想再玩一次将会怎样？看起来电脑游戏的戏外部分几乎比其他单一经历的媒体对使用者更为不友好。看起来像是游戏设计者认为他们是电影导演，因此尽管他们所用媒体的本质完全不同，他们还想创作类似电影院放映的电影。

有些游戏足够漂亮，允许玩家控制片头动画的重放。The Last Express 给设计思想带来特别的源泉，用它的独一无二的“蛋”保存游戏，这个特征允许玩家返回到游戏的任意一处去重玩。这种游戏实时发生或接近实时，还在这方面引以为荣，因此，玩家能手动调节时钟返回他感兴趣的特殊时段，且游戏将带他回到那一点，这一特征对于理解复杂的游戏故事至关重要。为了确保玩家了解他们在关卡中必须做的事情，我自己的游戏 Damage Incorporated 使用了互动任务简报。玩家能使用箭头在文本或图形文件间控制其前进或后退。显示任务简报期间，还有补充屏幕播放材料的对话。玩家可使用显示在屏幕底部的大型录音机来让对话暂停、返回和快进。用这种方式，玩家就能以他们想要的任意速度读文本和重听他们没听明白的对话。

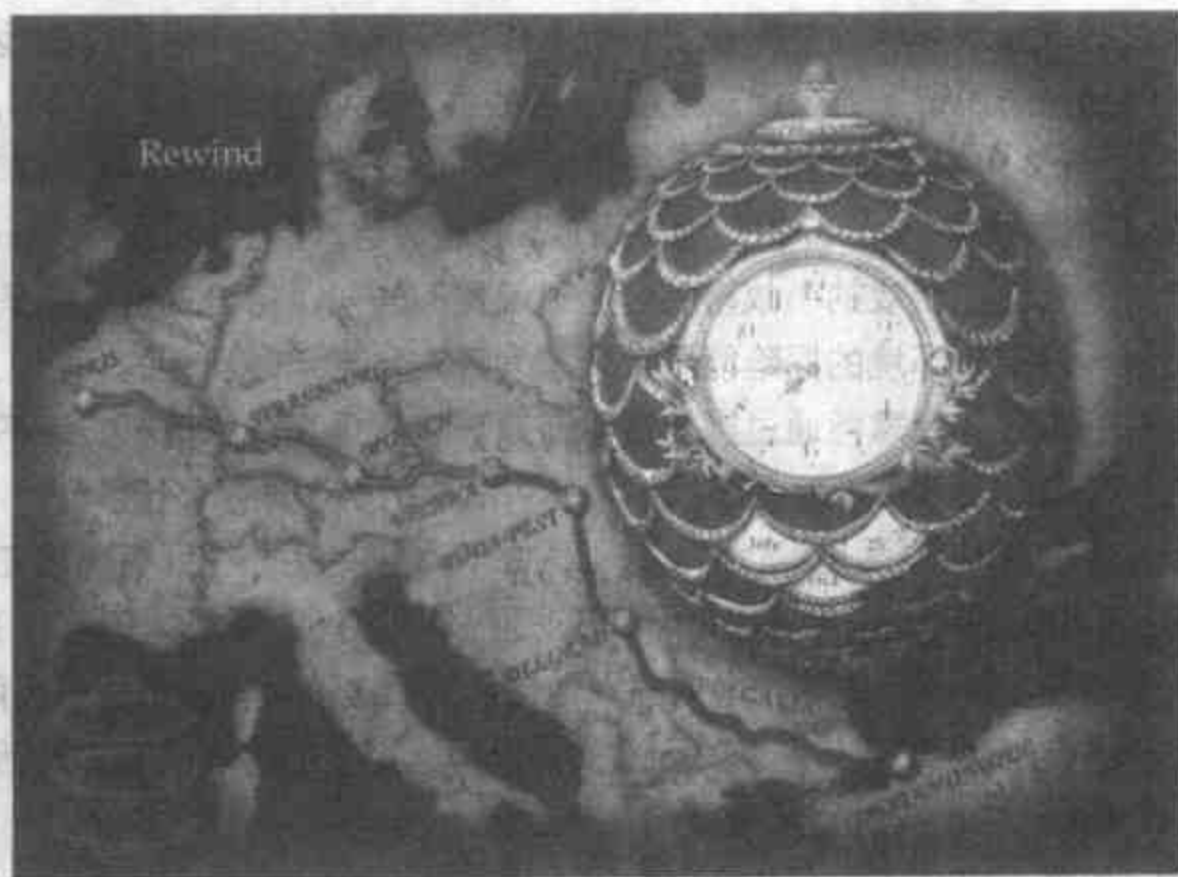


图 11.3 The Last Express 更明智的保存游戏系统允许玩家为了重看片头动画或再玩游戏的某部分而回调游戏时间

不幸的是，许多游戏的片头动画中仅有的互动是玩家跳过全部片头动画的功能。当显示任一游戏测试部分时，由于许多玩家想跳过游戏的非互动部分，所以这个功能是必要的。没有游戏尝试强迫玩家观看片头动画，这是完全不必要的限制。如上所述，比完全跳过更好的设计是允许玩家在看片头动画时前进或后退，以他们自己喜欢的速度观看和重看。

如果一个人停顿片刻去考虑在游戏中讲故事所用的戏外设置的本质，他将受一个陌生概念的影响，使非互动的经历破坏了互动效果。例如：当你去看电影时，电影院工作人员曾经停播电影、打开灯、指导观众去看他们发的书吗？有时在屏幕上显示些文字，但从不在某种程度上要求观众每次读许多文字。相反，电影为观众呈现的是连贯的媒体经历。另一方面，游戏在表





面上不由自主地与媒体混合，强迫想玩游戏的使用者先读一小段书，看一个电影，然后开始玩。的确，讲故事、介绍情节和游戏本身的角色有一种更好的方式，它比戏外讲故事高级，至少为玩家维持一个可改变的经历。

11.2.2 在游戏中

在游戏中讲故事有许多的高级技术。*Half-Life* 以其故事实力在游戏出版界得到普遍赞扬。然而，如果一个人看这个游戏故事，实际上它并不引人注目，也许甚至是不新奇的。许多游戏，甚至许多第一人称射击游戏，已经包含引人入胜的故事。然而，*Half-Life* 做得好的地方是完全在游戏内讲故事。玩家从不会失去对他角色的控制，即使在他被锁进观察室里、被困在一辆电车上，或被扔入垃圾捣碎机。通过合并关卡设置、爱讲闲话的科学家、PA 系统的宣布和 NPC 脚本化行为来共同传达故事。在游戏结尾，玩家留下的印象是这个故事是因为引人入胜的讲故事方法而优秀的。

通过游戏操控讲述故事时可以用不同的技术：

- 文本：通过围绕游戏环境的文本设置，玩家能了解许多内容。标识目的地的方向、一个特定地区的早先居民留下的笔记、在墙上的涂画，或乱放的供玩家读的书。
- 关卡设置：不管他们是否尝试去不去讲故事，几乎所有的游戏都用这项技术。看看 *Centipede* 的花园设置、*Doom* 的似地狱设置，或者甚至木制台板游戏 *Monopoly* 的艺术装饰不动产设置。这些游戏的小故事完全通过设置而告知，但设置也可能是讲更复杂游戏故事的关键。对游戏环境的探险能导致玩家发现在特定地区居住的一类人，或者是过去的时代居住在那里。玩家能通过操控这个游戏环境看见真实的情况，而不是在片头动画里读到这个地区的衰败。设置是演故事代替讲故事的一个理想例子。
- 对话：在游戏中 NPC 对话是设计者能增强讲故事效果的另一个强有力的工具。在游戏中通过玩家和 NPC 的会谈说出这段对话；在那里，玩家能选择他的角色对 NPC 的对话的反应，或者选择多种反应中的一个或输入他自己的反应。对话在游戏中也能发生非互动、NPC 在游戏中或友好或不友好、对玩家交谈以传达更多的游戏故事。对话也能来自于计算机终端、PA 系统，或大型录音机到几个装置。
- NPC 行为：当然，NPC 不应只与玩家交谈；还应该能根据故事线采取相应的行动。例如，玩家在游戏中与两个不同种族联盟战斗，依据两个种族痛恨彼此的故事线索。如果玩家同时与两者斗争，他将能欺骗他们相互斗争。在一个和平的村庄里，如果玩家带着武器接近 NPC，也许 NPC 将逃离玩家。在一个较敌对的城市中，如果玩家撤退失败，NPC 可能带着他们自己的武器和恐怖事件攻击玩家。NPC 也可用于向玩家传递关于游戏环境本质信息的脚本化行为。例如，一个城市的人民生活在类似纳粹警察势力的恐怖之中。当玩家进入时，他可能看到一位遭到许多警察粗暴及不公平毒打的市民。

Marathon 游戏使用文本熟练地传达其故事线索，从不将玩家带到戏外。游戏以计算机终端分散遍及玩家操控的关卡为特征。玩家能走入这些终端中的一个，键入热键去击活它。随后，游戏环境的玩家风景将被终端的特写镜头代替。然后，玩家能使用箭头键在不同的文本屏或前后的翻动，这些文本能显示更多的细节并告诉玩家在当前关卡她的目的。这些终端最大的问





题是当玩家正在阅读时，虽然玩家再不能看到游戏环境，游戏环境仍然非常活跃，玩家可能遭到外敌的攻击或淹没在上涨的水里。有时这会使终端的阅读非常紧张，使玩家的游戏环境处于危急状态。当然，玩家能控制文本经过屏幕向前和向后翻动，以任意速度重读文本。

我自己的游戏 *Damage Incorporated* 使用 NPC 行为和对话的合并给玩家一些角色的感觉，这些感觉是来源于陪伴他完成游戏任务的队友。玩家能在 13 个不同的舰队中精选出他想要的 4 个，陪伴他完成一个特定的任务。每个舰队有截然不同的个性人物，通过他在任务本身过程中所说的话来体现这一点。这个对话可以包括对玩家指示的反应、关于任务本身本质的评论，或对玩家特殊有效杀敌方式的反应。此外，不同的队友将对不同的任务起不同的作用。有些舰队的内心声音很少，如果获得太多的任务，他们将变成“震惊的炮弹”随意地运行在关卡中，并始终嘀咕乱语。其他舰队将有一些道德目标的任务。结果，这些流氓队友在某确定环境下将背叛玩家和他的其他队友，他们继续乱闹，呼喊他们对即将到来的任务的不满。这样，对话和 NPC 的合并产生一组带有真正个性的队友，几乎这些个性全都在玩游戏本身的过程中体现出来。

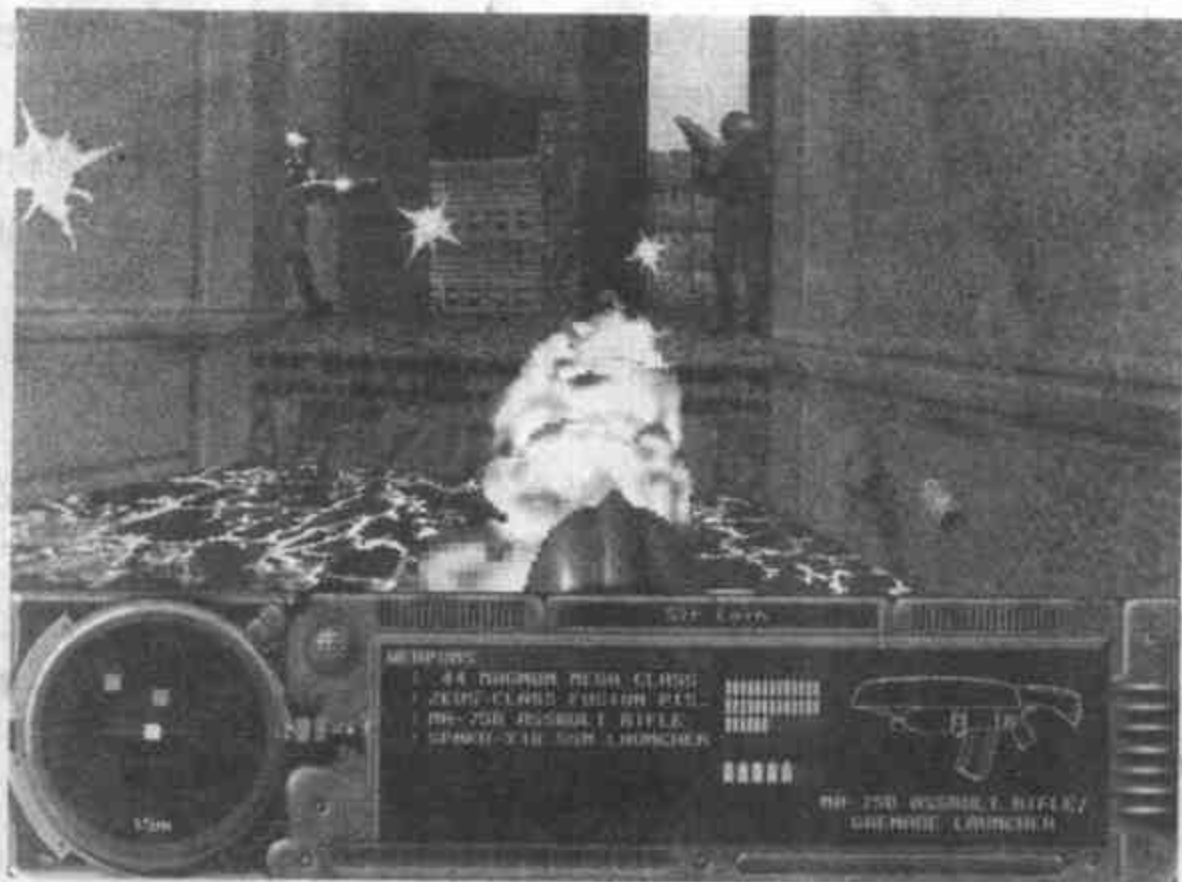


图 11.4 *Marathon* 游戏允许玩家在遍及关卡分散的计算机终端记日志。在那里他们能阅读更多的关于游戏的复杂故事

对于在游戏内讲故事，令人关心的问题之一是玩家可能错过故事的一些部分。倘若玩家没有看到故事，将会怎样？因为玩家在游戏内讲故事中从不失去控制，这使玩家避免与角色说话、证明脚本化 NPC 行为或阅读标志成为可能。在面对非互动片头动画或滚动文本时，锁定玩家，以保证看到设计者想让玩家看见的东西，这是事实。但是，如我以前所述，一个人需要记住游戏是互动形式，如果玩家没有经历故事的每一个元素，这是互动的本质。如果玩家对故事的所有部分感兴趣，搜寻出这些部分，这是玩家的职责。如果玩家比较喜欢操作那些重视游戏操控性的游戏，那是她的选择使然。确实，在再次玩游戏时，发现故事的不同关卡是重玩游戏的一个重要动机。

几乎每个人都有英语老师，老师在讲写作课时，一是强调表现而不仅是讲述，不能平述人





口众多，不能希望读者从作者对一个地区的描写，推论出这个地区有大量的人口。对于游戏，在游戏内讲故事是表演的等价物，而片头动画和其他方式便属于“讲述”。游戏内讲故事，玩家会经历故事本身而不是间接地被告知。除了继续使玩家沉浸在游戏环境中以外，游戏内讲故事向玩家展示故事，而不是只向他讲述。

11.2.3 外部材料

许多游戏使用外部材料讲故事。在 20 世纪 80 年代，当硬盘容量受限制时，设计者不能把想要包括的故事的全部内容都存储在小容量软盘中，这个事实是明显的。一些设计者使用手册来介绍游戏的背景故事，叙述的内容将把玩家带到开始玩游戏的地方。有些游戏，例如经典的 *Wasteland* 甚至使用“段落”指示，玩一会游戏，然后，当玩家得到一个故事的接合点，将得到指示“现在是段落 47”。有时手册以副本形式使用，因为没有手册，玩家将不能玩游戏。

街机游戏也使用外部材料。通常，游戏角色的名字写在图板上，而不是游戏中。一些图板甚至包括少量的句子，进一步解释游戏的设置和玩家的任务。艺术工作作为街机游戏的特征，使用图板为单调的故事线索增加了一点深度。

现在，不赞成在手册和其他材料里讲故事。由于技术限制而需要通过外部材料讲故事，我们确信这种情况已不再存在。因此，通常手册里的故事并不是游戏设计者或者他们磋商后写的。所以几乎不能把这些故事看做游戏本身的部分，但市场部门试着去创作一个游戏环境，通过包装盒宣传游戏的背景。我自己的游戏从不使用手册去传达故事，因为我认为它降低了在电脑或者控制台上连续玩游戏的体验。

也就是说，一些游戏已经非常有效地使用了外部材料。特别是，*Infocom* 游戏总是在游戏包装包里包括增加玩家游戏可玩性经历的材料。游戏常涉及这些材料，大意是：“你发现的弹药库与游戏包里出现的相同”。这些材料通常是与游戏作者一起协商准备的，因此成为游戏本身有效的部分。关于 *Infocom* 怎样使用其游戏包材料去增加故事深度和创作动机等更多信息，请见第 10 章 *Infocom* 的作者 Steve Meretzky 的介绍。

11.3 沮丧的线性作家

在许多电脑游戏中，主要故事的问题之一是，写故事的人希望在一种更线性的媒介中编写故事。有时，在游戏项目失败的编剧或小说家之中，他们必须做游戏工作并且游戏工作纯粹是为了钱而做。这时他们常常感觉失望，并同时发现自己在以艺术形式进行赌博。通常，他们采用线性方法写作却讨厌互动写作的训练，这些作家试着将好多年已习惯的、全部的线性写作手法应用到游戏中。在这里，他们会败得很惨。

有时，游戏开发者自己秘密地或不太秘密地希望他们在另一种媒介中工作，使他们的故事写作有机会被选中。毕竟，因为在游戏存在时，电影已经是一个更受重视、更流行和更易赚钱的媒介，其角色、导演有时甚至作者都成为巨大的、个人崇拜的对象。游戏设计者被这种魅力所吸引并变成电影从业者的嫉妒者。这些设计者开始强调他们游戏的影片本性，有时甚至根本



不承认它们是游戏，称它们为“互动电影”。这些游戏电影的片头动画变得越来越长，预先确定的故事线完全支配了游戏。

在某种程度上，在媒介的初期，游戏开发者把故事放入其游戏中的错误是可宽恕的。例如，当介绍已有的制片技术时，许多第一批电影以舞台剧为原本。摄相机放在固定的三角架上，演员把摄像屏幕看做他们的舞台，好像他们面对观众在演戏。因为电影语言还没发明，所以没有剪辑、摇镜头，或者任何一种照相机的移动。然而，随着时间的推移，电影摄制者发现他们的电影能比舞台表演更直接，更具备利用新媒介的实力。在某些方面，游戏仍旧遭受了同样的问题，就是将已有的媒介（尤其是电影）带入游戏，并没有考虑怎样用一种适合互动的语言把故事讲述得更好。

由这些灰心的线性写作的作者创作的项目既是游戏又是电影，通常最后的结果，他们做得都不好。使用互动方式的讲故事技术比使用传统线性技术更难，但是最后，从游戏中获利证明了它是值得的。在线性写作时会有不好的征兆出现，当问题出现时就能意识到，这对防止其破坏你的游戏是至关重要的。

第一个问题强迫玩家仅有一条预定的路径下经历故事。线性作者常常感觉将剧情展开只有一条路，如果玩家尝试追踪任何其他的事件，他或他的角色将被杀。线性作者不允许玩家在故事空间里发现不同的路，仅有一条路有利于更强有力的叙述。线性作者没有意识到的是，游戏就是让玩家通过游戏环境找到他的路，尽管那条路可能毫无兴趣。因为玩家感觉到了自己的主人身份，所以它补偿了戏剧中可能失去的东西——这是玩家故事而不是设计者故事。

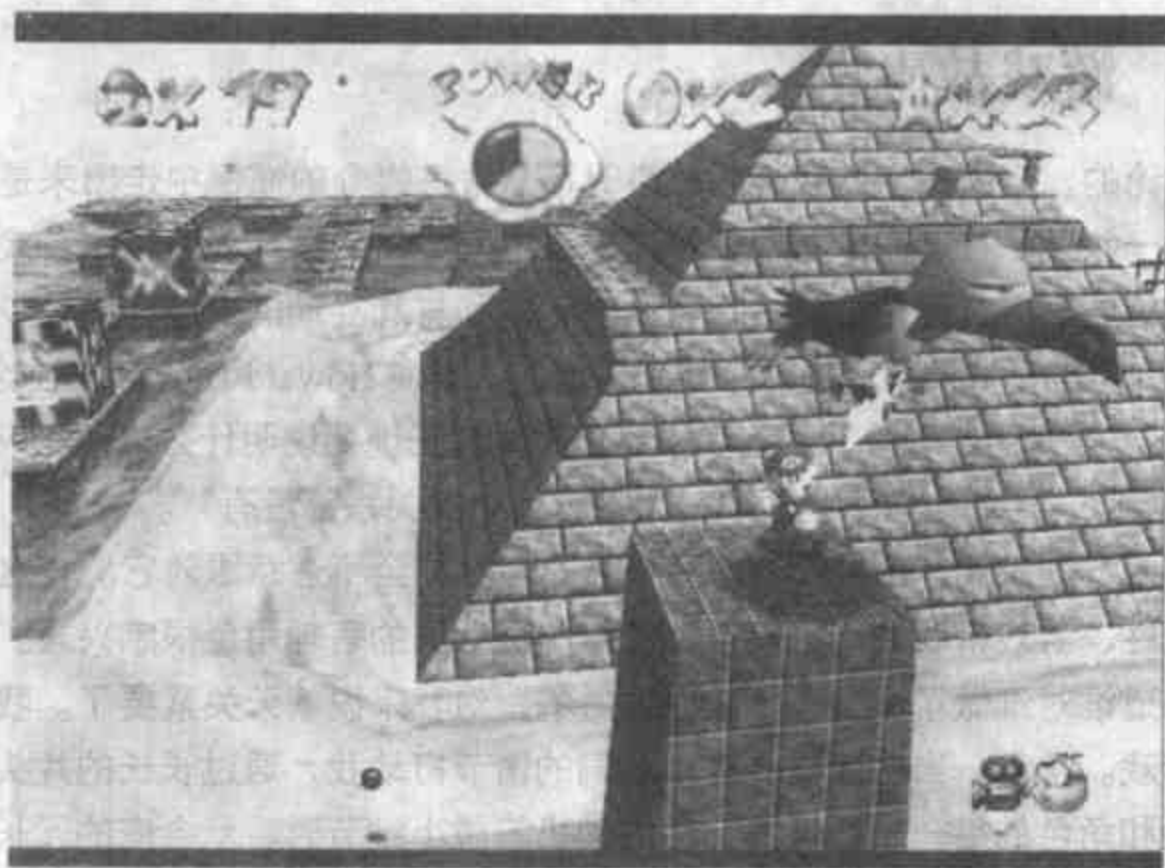


图 11.5 除了可能是现有的最著名的电脑游戏之外，Mario 具有相对不明确的个性

线性作者也常常试着去强迫玩家角色有一个较强的个性。在游戏设计中有一个普遍的误解就是，游戏想有一个强个性的角色去控制，特别是在冒险类游戏和战斗类游戏中。但如果看看在这些流派中最流行游戏的记载，将很快注意到玩家角色的个性常常保持在最小值。如 Super





Mario 64。虽然 Mario 有一个相当与众不同的面貌，但是他真正的个性是什么？实际上，根本没有，对于不确定的他，玩家将将自己的个性加在他的身上。在 Tomb Raider 里的 Lara Croft 如何呢？也有一个非常与众不同的外表，但却非常不明确的个性。如果看看在 Doom 中的空间舰队或者在 Half-Life 中的 Gordon Freeman，就会发现他们没有个性。

原因很简单：当玩家玩游戏时，他们常常想按自己想法来玩。如果他们控制的角色固有非常强的个性，并对玩家有预先设定的许多限制，让他感觉好像他不能真正控制游戏。特别地，冒险类游戏强烈刻画玩家角色的性格，一直说一行行愤怒的对话，这种性格是无效的。我记得在一个冒险类游戏中，玩家必须控制一个受溺爱的小子，这小子经常向他遇到的人说些讨厌的话，做白痴般的事。谁愿意控制这样的角色？角色的对话实际上写得非常好并且很有趣，但被迫使用讨厌的角色作为他游戏环境的代替品，这对游戏的玩家却不有趣。游戏的作者远离主要角色有趣的特性，没有意识到在玩家游戏经历中的不利影响，这是很明显的。

我不是建议在你的游戏中不能有讨厌的角色，确实，没有个性角色，你的游戏将失去故事的大部分。不是试着用大量个性浸透到主要角色，而是用玩家在游戏中遇到的难忘和有趣的 NPC。如果玩家发现这些讨厌角色完全可接受，意味着玩家感觉他们有足够的个性。但是玩家角色应该是无定形的和未展开的，玩家以他的方式怎么看角色都适合。不要害怕，在角色上花 40 小时或更多小时之后，玩家将提出他自己的动机并操纵他的游戏环境代理人。在他的头脑中，他制作的角色将成为他喜欢的一个人，他将和他一起继续玩。



11.4 游戏故事

正如我所讨论的，当为一个游戏编写故事时，从线性媒介的惯例中走出来是重要的，例如强迫玩家跟随惟一叙述，在玩家的游戏环境代理人中有太多的角色。当创作游戏故事时，除了陷阱之外，游戏编剧家应避免过多关注玩家所处环境和与其互动的角色，而较少关注整体的故事情节。毕竟，许多电影导演想了解这个技术。例如，导演 Howard Hawks 在谈论他的电影 *The Big Sleep* 时说：“制作这部影片时，我了解到你不必真正地要说明什么事件。只要你有 一个好的图片做你的好场景——如果它不是故事的大部分就一点关系都没有”。

我已经玩了无数的游戏，但全部情节我都忘了，我完全不注意跟随它。在这些游戏中，我常常对游戏可玩性、游戏带我进入的情景和我在那里遇到的有趣角色很感兴趣。因为角色和情景有趣，那么知道谁对谁做了什么和什么时候做的，对我来说就无关紧要了。我所知道的是我正在有趣地玩游戏。常常，当游戏试着去用他们的情节打动我，通过长长的片头动画演义关于游戏环境的陈述和角色的每一个最近的行动的动机等微小细节时，就会变的冗长乏味。记住，玩家想玩游戏。如果故事增加很多经历，那是好的；但如果故事以这种方式开始，那是坏的。清楚地说明太多的故事内容是新手共有的缺点。读者、浏览者和玩家给他们比作者更多的称赞。故事的作者能让玩家详细了解角色的动机是有意义的，在作者的笔记本里，记述了所有不同的曲折和情节的细微差别，但在游戏中需要包括故事的结尾吗？不，最重要的是送给玩家的故事是能与全部故事相容的。如果不直接讲清楚最后的情节，玩家是不会介意的。

在第 9 章“人工智能”中，我谈论 Brian Moriarty 的“星群”概念和怎样创作更有趣的 AI。





故事情节，不管是否需要，游戏评论家都会感到失望。没有故事的游戏被看成是过时的和落伍的。如果你的故事是非线性的或者允许玩家制作他自己的故事，那么这些营销者甚至游戏评论家确实不会考虑。毕竟，公司和市场将喜欢有强个性的角色，因为它将获得更多的机会，能扮演动作人物并在星期六早晨的卡通节目中演出。从不介意角色的强个性可能疏远玩家。

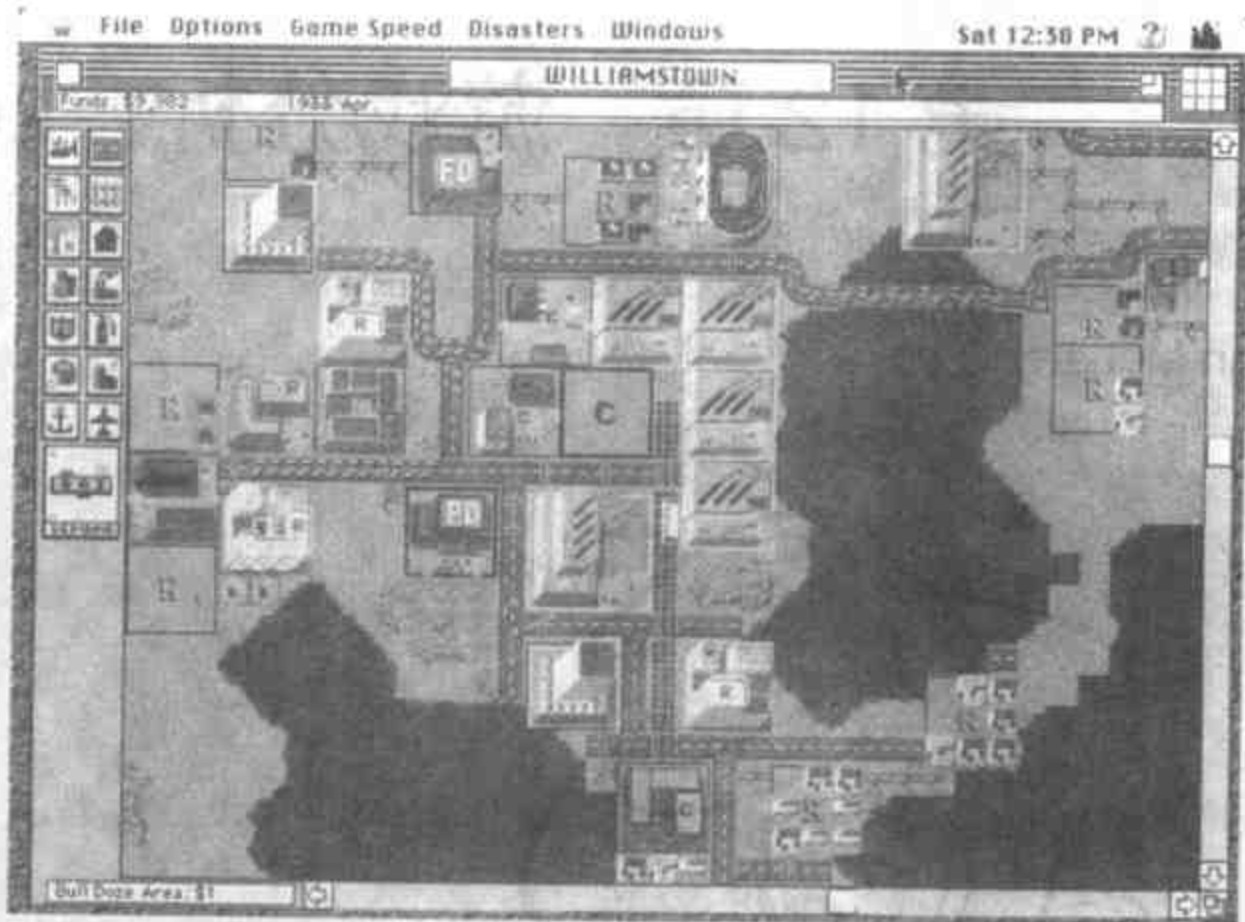


图 11.7 像 SimCity，允许玩家真正地讲述他们自己的故事，设计者很少有指导

作为一个游戏设计者，你的欲望一定不限于制造娱乐新闻或者做游戏评论家。许多大型游戏完全依赖于传统的讲故事方式。我们立即会想到 Civilization 和 SimCity，毫无疑问它们是那种允许玩家讲他们自己故事的大型游戏；在这些游戏中，设计者只提供一个初始点并让玩家由此展开整个故事。游戏确实不需要一个规定好的故事。然而，在一个真正的互动游戏中，玩家根据自己的选择就能改变故事，而保持情感的共鸣和讲述小说故事的能力是具有强迫性的想法。这种强迫性使我们很难想像会出现任何真正有抱负的游戏设计者，他们不会希望这种想法变成现实。





种故事。如果你试着去与故事中的游戏操纵或者技术斗争，那是不合适的；在一个劣质的执行游戏中，你将被落下。有许多故事要讲，有许多讲故事的方式。作为设计者，你的工作是发现一个故事和讲述那个故事的方法，这个方法与你将使用的游戏设计和技术一起工作。

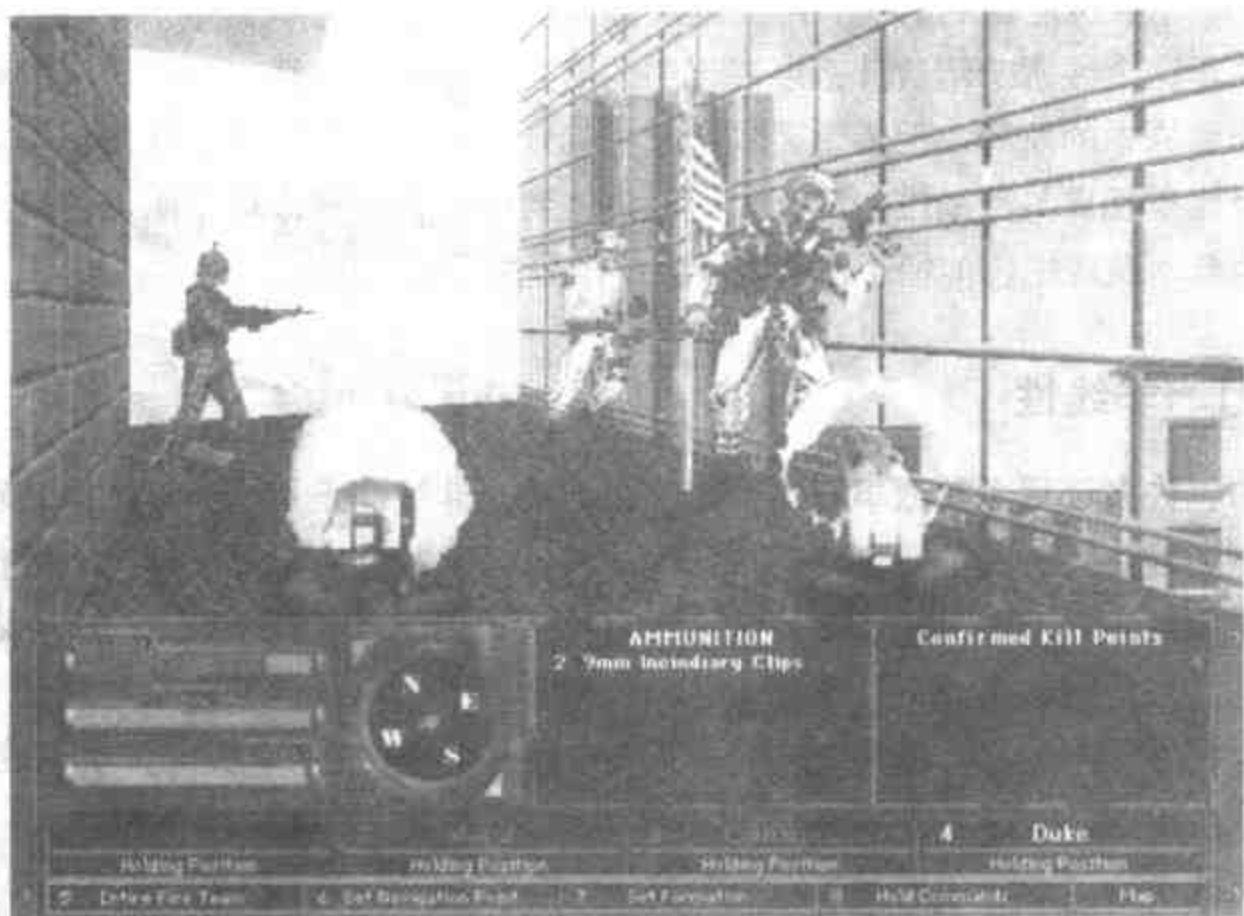


图 11.6 *Damage Incorporated* 的故事使其成为适合游戏操纵和技术的故事

对我来说，把故事自然地看成是游戏的结果。我很少孤立地想一个故事，并试着将它放入一些游戏中。相反，我看到了将伴随我的工作约束，开始想最有趣的部分可能为那个空间。我不把这些约束看成讲故事的限制，而把它看成指导方向或者甚至是灵感的来源。例如，在 *Damage Incorporated* 中，游戏有一个故事以前的很长时期，在头脑中有技术和游戏设计的构思。游戏设计集中在以玩家为中心在一个 FPS 环境中控制队员，陪伴玩家的不同队员跳出思想，他们每个人都具有独特的个性。哪种人会在 20 世纪 90 年代的舰队军团里？他们在一个搏斗情境中如何反应？当他们看到他们的指挥官被杀，他们的反应将是什么？有些问题以控制游戏故事的发展结束。这些问题是因游戏设计的局限性产生的。



11.5 梦想

有人会说游戏的目的是，允许不同的玩家自己决定各种各样成功的类型、奖赏玩家试验和探险、授权玩家自己的选择。这些因素允许玩家在玩你的游戏时制作他们自己独特的故事。如果你想通过你的游戏讲一个更预先决定的故事，也使玩家感觉是她自己的独特故事，可做任何事情，这是重要的。玩家在她的游戏活动中应该感觉有控制权。

营销者和游戏评论家喜欢在游戏中讲故事，因为在游戏中讲故事比游戏设计更易理解和讨论。一个故事的背景或文本很容易复制，比游戏操控更易描述。现在，如果你的游戏没有太多





故事情节，不管是否需要，游戏评论家都会感到失望。没有故事的游戏被看成是过时的和落伍的。如果你的故事是非线性的或者允许玩家制作他自己的故事，那么这些营销者甚至游戏评论家确实不会考虑。毕竟，公司和市场将喜欢有强个性的角色，因为它将获得更多的机会，能扮演动作人物并在星期六早晨的卡通节目中演出。从不介意角色的强个性可能疏远玩家。

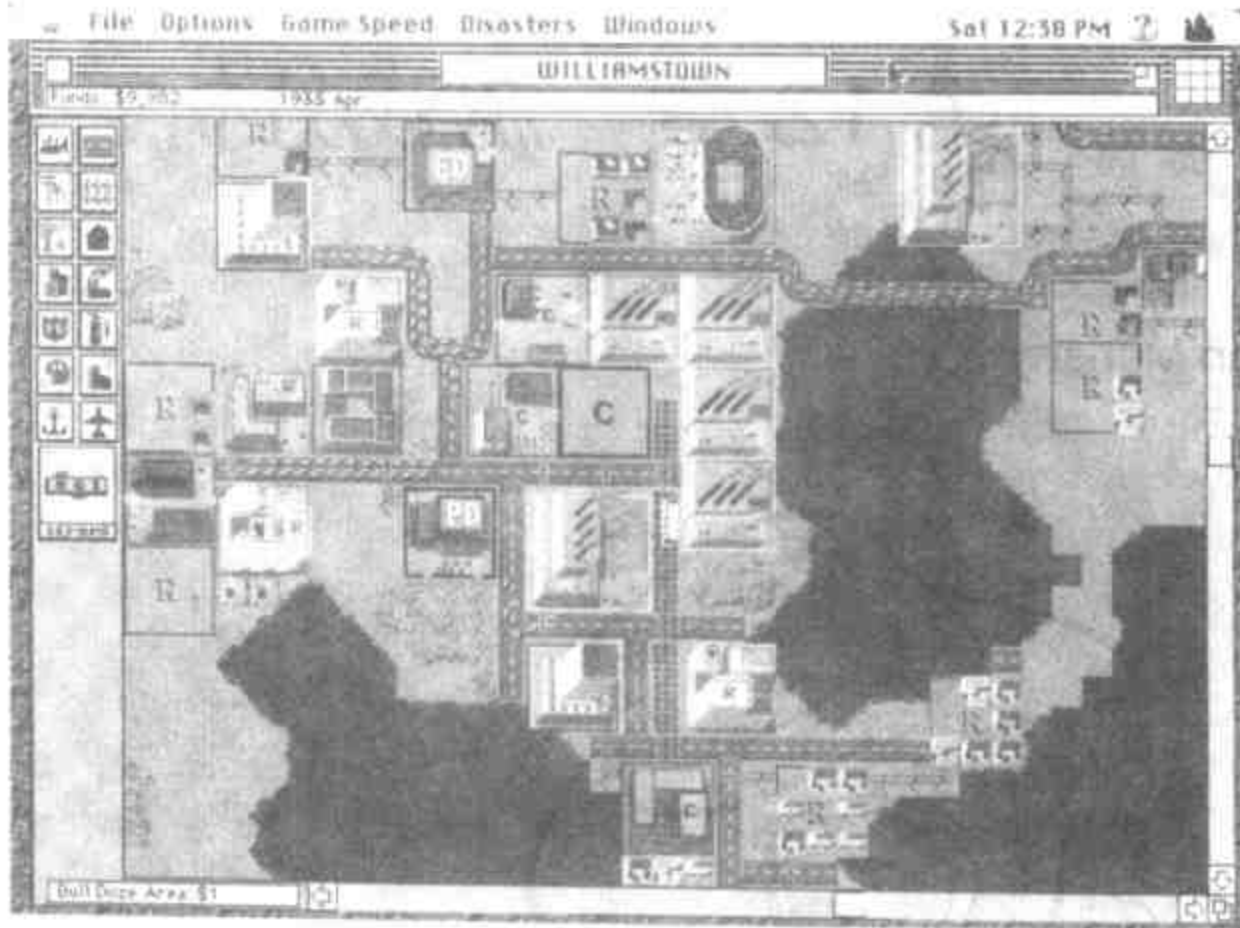


图 11.7 像 SimCity，允许玩家真正地讲述他们自己的故事，设计者很少有指导

作为一个游戏设计者，你的欲望一定不限于制造娱乐新闻或者做游戏评论家。许多大型游戏完全依赖于传统的讲故事方式。我们立即会想到 Civilization 和 SimCity，毫无疑问它们是那种允许玩家讲他们自己故事的大型游戏；在这些游戏中，设计者只提供一个初始点并让玩家由此展开整个故事。游戏确实不需要一个规定好的故事。然而，在一个真正的互动游戏中，玩家根据自己的选择就能改变故事，而保持情感的共鸣和讲述小说故事的能力是具有强迫性的想法。这种强迫性使我们很难想像会出现任何真正有抱负的游戏设计者，他们不会希望这种想法变成现实。



Chapter 12

第 12 章 游戏分析：Loom

由 Brian Moriarty 设计



在 1990 年发行的时候，Loom 游戏立刻就成为一种风格独特的冒险类游戏。与之前由 LucasArts 发行的图形冒险类游戏相比，尽管它们在玩法上有许多相似之处，但是从叙述故事的角度出发，Loom 则致力于归纳出游戏的核心结构，去掉了所有枝节的东西。在游戏指南中可以发现，游戏的作者清醒地意识到，他们正在创造一些与众不同的东西，正如下面“关于 Loom”部分的摘要中指出：

Loom 与传统的“冒险类游戏”相比在许多方面都有所不同，它的目的在于，让我们投身于逐步发展并充满想象力的幻想之中。它既不是角色扮演类游戏（尽管包含角色扮演的成分），也不是与玩家斗智的人物集合。这个游戏的神秘之处就在于可以激发想象力，使你融入到故事当中，而不是使你感到灰心，或者增加完成该游戏所花费的时间。



在后来的“我们的游戏设计原则”部分，我们发现了与 Loom 游戏独特性有关的其他参考：

我们相信，你是基于娱乐的目的而购买游戏的，并不想在（在玩的过程中）每次犯错误的时候，都像遭到当头棒击一样（令你难过）。因此，当你在一个从未到过的地方进行试探时，我们不会让游戏嘎然而止。与俗套的冒险类电脑游戏不同，你不会突然发现自己偏离了方向，也不会因为拾到了什么利器而死掉。

我们认为，你会比较喜欢通过探索和发现未开启游戏的神秘之门，而不是要死掉一千次之后才达到目的。我们还认为，你花时间玩游戏，并不想在得到某一确定目标的电脑指令之前，只是试探着键入一些具有象征意义的东西。

读过上述内容就可以了解到，Loom 也许只是该游戏作者 Brian Moriarty 看到其他冒险类游戏在令玩家愉悦方面的不利之后，自己所做出的一种反应。我们并不清楚游戏指南部分是否是 Moriarty 亲自编写的，但是看起来，这些指南至少确切表达了他在这一问题上的感受。Loom 决定保留冒险类游戏中那些确实有益的叙述故事的成分，同时去除了所有与玩家的愉悦相冲突的东西。Loom 备受欢迎，它热切地希望玩家完成一个有趣的故事。

在 Brian Moriarty 加盟 LucasArts 并着手制作 Loom 游戏之前，他已经在 Infocom 公司工作多年，这家公司以其文字冒险类游戏的优质和深度而著称。在 Infocom 公司，他制作了两款文字冒险类游戏：Wishbringer 和 Trinity，还有一个纯文字冒险/角色扮演相混合的游戏：Beyond Zork。Wishbringer 最初是为初级玩家设计的，比较容易玩，而 Trinity 和 Beyond Zork 则都是大型的、相当难于完成的游戏。Loom 看起来是游戏设计方向上的一个转折点，它不是仅仅为了挑战玩家而设计的，其中包含的挑战都是那些对游戏至关重要的。而且，Loom 也是 Moriarty 设计的第一个不含有文字解析器的游戏，如果我们相信游戏指南确实表达了他切身的感受，就会知道，Moriarty 多想把这种输入方法从游戏中去掉。Loom 游戏的简单性是对从前那些冒险类游戏不必要的复杂性做出的反应，这些游戏指的是通常意义上的，也包括 Moriarty 自己设计的游戏。在 Loom 中，故事是最重要的，所有不利于情节发展的环节都应该去掉。



12.1 注重游戏机制

有一类游戏将重心完全放在它所要完成的目标上，Loom 就是这类游戏的典范。Moriarty 在经过长期且艰难的思索过程后，想出了一个对于叙述故事必要的、包含内容最少的游戏机制，而并不是试图将所有可能包含的游戏机制均包含其中。因此，他删除了所有与该故事无关紧要的东西，这极大地简化了这个游戏，同时也使 Loom 成为一个出色的、相当容易操纵的游戏。





图 12.1 Loom 的游戏机制注重叙述游戏的故事情节

Loom在开发过程中使用了SCUMM故事系统，LucasArts制作的所有冒险类游戏中都使用了这个故事系统。由于Ron Gilbert和Aric Wilmunder的缘故，在应用于第一个游戏之后，SCUMM就代表Maniac Mansion的脚本化实用工具了。如果看一看LucasArts制作的其他冒险类游戏，我们会注意到，几乎每一个的游戏机制和用户界面都比Loom强许多。在Maniac Mansion（1987）和The Secret of Monkey Island（1990，与Loom同年）这两个游戏中，玩家不仅可以在各种物体上点击各种动词，而且还允许玩家操作存货清单。这两个游戏都是用SCUMM系统制作的，如果Moriarty愿意的话，他也可以把存货清单和动词系统同样加到游戏中。实际上，在Loom游戏发行之之前，几乎所有的冒险类游戏都把存货清单和动词系统作为常规的组成部分（自从Loom发行之以来，很多冒险类游戏已经去掉了存货清单和动词系统，最值得一提的就是，Myst和许多与其类似的游戏）。所以，当Moriarty在游戏中略去这些结构时，是对SCUMM系统和传统的彻底摒弃，这种举动是一种重大的突破。如果作者重新构思这个故事，并考虑采用存货清单和动词系统，可能会极大地增加游戏的深度，但是没有这些，游戏仍然运行得非常完美。

许多冒险类游戏都使用了带有分支的对话树。在这种系统中，当玩家角色与另一角色说话时，系统就列出这个角色可以讲的各种句子，玩家可以从中选择，从而在对话过程中获得某种程度的互动性。The Secret of Monkey Island游戏的作者Ron Gilbert正是使用了这样的系统，特别是按照传统的打斗次序，使游戏可以达到一种平衡。但是，正如动词及存货清单一样，Loom游戏中也没有分支的对话树。当玩家和某人说话时，他只是看着一个展开的非互动性的片头动画，却无法控制它。在某种程度上，这会削弱玩家与游戏的互动程度。但是在最后的分析中，分支的对话树系统总是包含有限数目的分支，因此，这样的系统只是让玩家逐一地点击每个选项而已（在几个以分支对话来增加游戏深度的例子中，The Secret of Monkey Island是其中之一）。对





于Loom游戏，Moriarty使用了片头动画的对话形式，因为它能最有效地表达这个故事。而且Moriarty注重故事叙述，他不会任由冒险类游戏的惯例阻碍这个目标的实现。



12.2 用户界面

Loom 游戏的界面只需玩家使用鼠标和一个按钮即可，这是其简单性的集中体现。当然，这样的界面使游戏对任何熟悉“指向-点击”系统的人来说，学和玩的过程都很容易。这与其他的一些冒险类游戏，特别是那些在 20 世纪 80 年代盛行的纯文字冒险类游戏形成了鲜明的对比，其中也包括 Moriarty 曾经制作的游戏。它们几乎都含有文字解析器，从比较理想的情况来说，这样可以使玩家使用自然语言做任何她想让角色去做的事情。像“去拿书”、“向西北方向”、“用红色的钥匙开门”和“看图画”这样普通的命令都出自文字冒险类游戏。这类游戏的局限性是，许多文字解析器没有整套的英语词汇，也不能正确地解析复杂的句子。事实上，发行 Moriarty 制作的 Wishbringer、Trinity 和 Beyond Zork 的 Infocom 公司，拥有目前为止可用的最好的文字解析器。然而，这种解析器的使用正面临着挑战。当玩家确切地知道他在游戏中需要做事情但却无法找到正确的词汇来表达时，他们就会感到灰心丧气。更不用说玩家需要把每个单词都拼写正确，系统才能正常运行，而很少有人擅长这项工作。充其量不过是经过一段时间之后，我们习惯了一个文字解析器的特性，但对一个初学者来说，受支配感同样让人倍受打击。

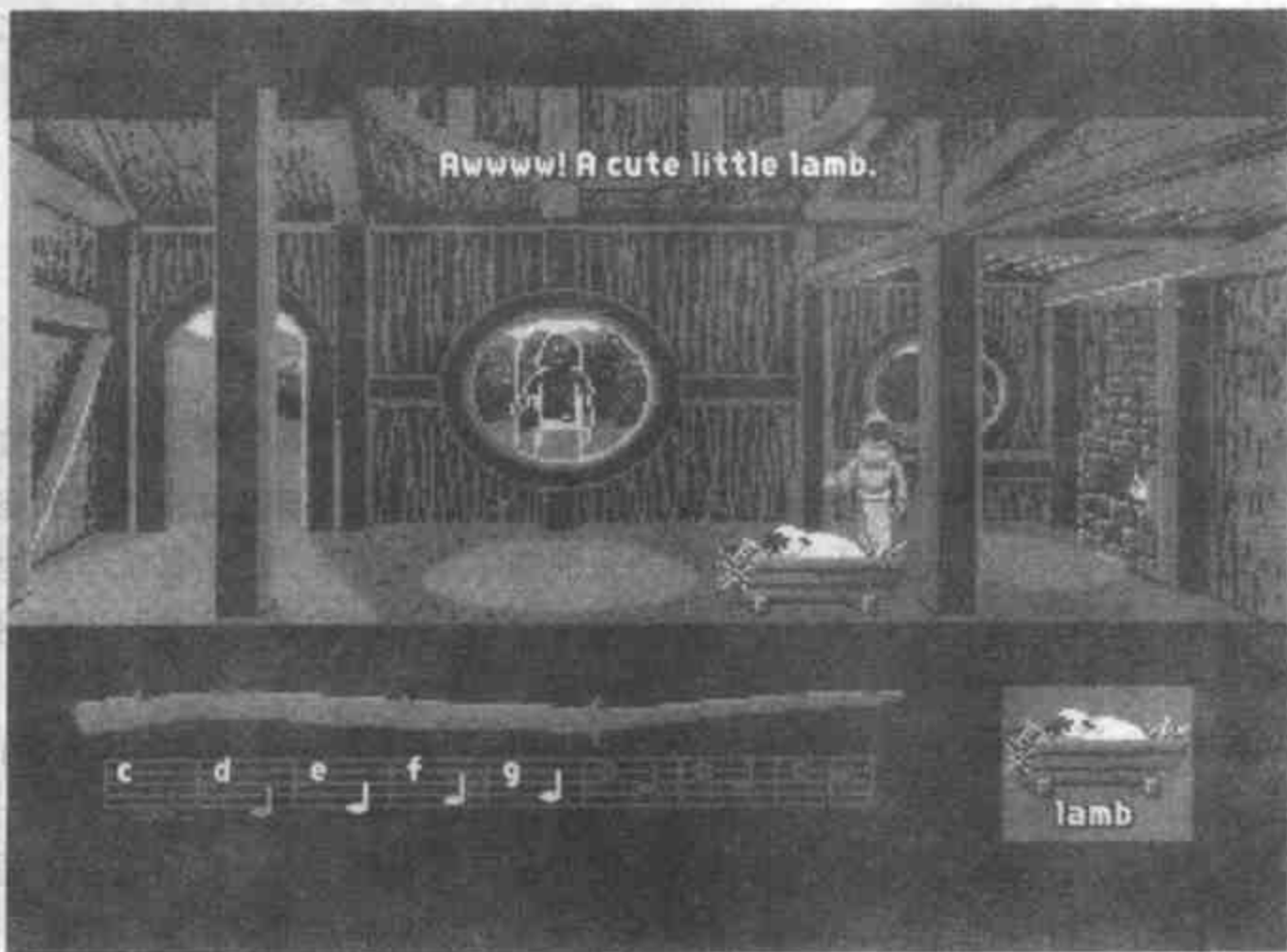


图 12.2 Loom 游戏尽可能地使其界面保持简单，玩家只需使用鼠标就可与游戏环境互动

在前面提到的游戏指南的摘录中，确实指出旧的文字解析器常被人嘲笑。看起来，Moriarty 更喜欢一个直观的、易于学习的界面。当然，任何界面的基本要求之一都是易于学习。玩家面





临的挑战应该在于游戏本身，而不在于必须学会操纵才能影响游戏环境的控制。Maniac Mansion 使用了一个完整的指向-点击界面，Loom 从这个游戏机制中借鉴了很多东西，这一点至少可以在环球漫游方面有所体现。玩家只要点击他想去的位置，就可以将他的角色 Bobbin Threadbare 移动到世界的任何角落。从 Diablo 到 Grim Fandango，再到 Command & Conquer，对于那些在上述游戏中见过无数指向-点击移动系统的现代玩家来说，这一点是显而易见的。这个系统的妙处还在于其明显性，一旦玩家看到系统在运行，就想像不出还有什么别的办法可以用鼠标指引某个角色的移动。

然而，Maniac Mansion 及其他图形冒险类游戏仍然为玩家提供了用于点击的动词。这些动词基本上是由文字解析器遗留下来的，玩家可以先点击一个物体，然后点击一个动词，以此来操纵该物体。还有一些图形冒险类游戏已经用图标取代了这些动词，这些图标在功能上与文本是一致的。当然，在许多情况下，一个特定的物体仅受一个动词/图标的影响，因此，图标的广泛功能便显的不那么重要了。Loom 游戏完全取消了动词，允许玩家双击一个给定的物体，至于他要如何处理这个物体，要让游戏自己去解决。如果玩家双击一个人，Bobbin Threadbare 就会和他或她说话；如果双击一个带文字的物体，Bobbin 会把上面的文字读出来；如果双击一只绵羊，Bobbin 会去碰碰它。这个游戏是与玩家一起在运行，它允许玩家只做对自己有用的事情，不会与其背道而驰。双击显然是对单击的扩展。单击可以把 Bobbin 移动至某物体；双击则让他使用这个物体。显然，这种输入系统与指向-点击系统在 Macintosh 和 Windows 平台上的使用方法是一致的，因此玩家在开始玩游戏之前就可能了解了它的用法，这样则具有更大的优势。我们应当学习的经验就在于，从已经建立的标准规范中学习借鉴输入方式的思想，要比拼凑新的方法好得多；因为玩家在学习一种新的输入方式的过程中会经历挫折，这种情况下，他用此方法可能获得的任何微小的收获几乎都会被抹煞。



12.3 草稿系统

虽然这个游戏没有用到存货清单、动词和分支对话，但它却在其中增加了一种独特且设计完善的机制，玩家只需利用状态条即可操作这个游戏。该系统与其他游戏中的施咒系统截然不同，它允许玩家使用相同的咒语对游戏中的各种物体施加魔法。这个系统与其他游戏的施咒系统完全不同，尤其是那些在 1990 年出现的游戏。此外，游戏的界面完全可以使用指向与点击的方式操作，非常容易学习。

在这个系统中，玩家可以在不同的情况下听到不同的声调，然后根据声调本身来进行重复，这种方式会让我们联想到西门说过的一个游戏。如果玩家双击某个转轮，就会听到一系列四声的声调，这些声调也会反映在玩家的状态条上，出现在电脑屏幕下方。在状态条下面的对应位置还会出现一系列的音符：例如 c、d、e、f 等等，一直可以高到八度音。当玩家第一次听到那些声调时，这些音符就会变亮，以向玩家表明这些音符的区别。玩家必须记住这些声调（通常是写下来），然后就可以重复它们，从而用特殊的“草稿”或施咒词语在不同的物体上施加法术。玩家只要点击状态条不同的位置就可以重复音乐记录了，可见，状态条是一个相当直观的界面。



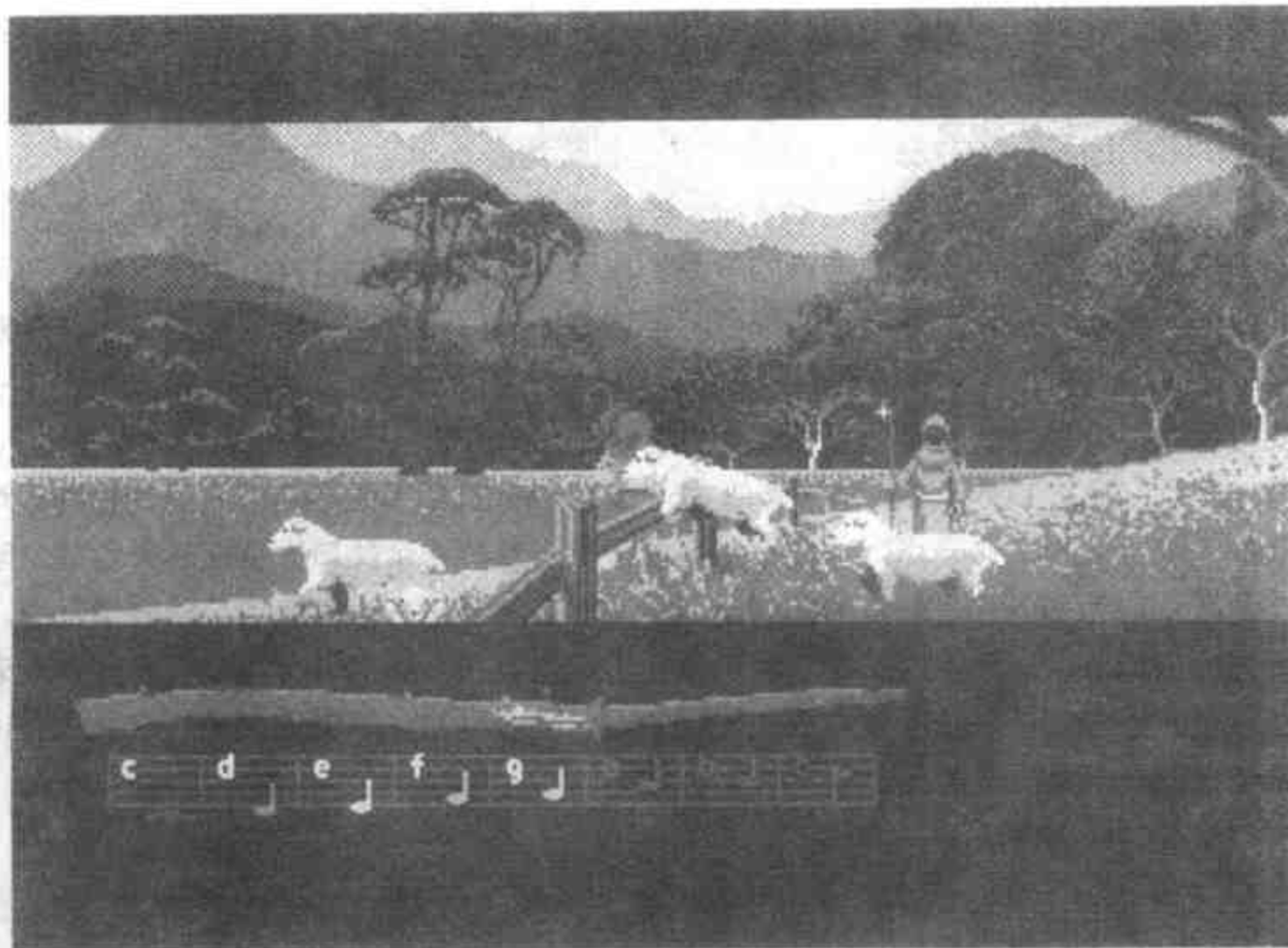


图 12.3 如果在专业级玩家设置下玩 Loom 游戏，那么状态条上的音符就会消失，这使得游戏的难度更高

如果是专业级玩家设置，那么想记下草稿就更难了。玩家只能听到那些声调，而音符不会再闪现在电脑屏幕上，音符也不会再在状态条上闪烁。这就迫使玩家不得不“用耳朵玩游戏”以获得成功。同时，与其他完全不需要声音的游戏相比，草稿中音乐的特点和整个游戏都是一种很大的突破。Loom 游戏不仅使用音乐作为背景音，实际上音乐已是游戏中完整的组成部分。

如果声调的顺序颠倒，也会导致相反的结果。最初的声调是玩家通过双击游戏中的物体而获得的，这些物体与玩家记下的草稿相一致：双击一个刀片，表示使其变得锋利（sharpen）；双击从瓶子中流出的水滴，表示把瓶子倒空（emptying）；双击一个盛满冒泡染料的罐子，表示用它来染色（dye）等等。随着状态条旋转的草稿是对游戏中的物体施加作用的基本方法。有时玩家记下的草稿不太明确，他需要一些创造性的思维来判断在什么地方应该用到哪个草稿。应用在某处的草稿，以后可能会得到相关的但却完全不同的应用。例如，最初用于孵蛋的一个草稿，后来却可以很方便地用于开门；用来治愈一个人的草稿也可以用来缝合宇宙中的裂缝。所有的联系都很微妙，但却合乎逻辑。如何运用这些草稿构成了游戏难题的主要来源，也使这些难题不至于错综复杂地缠绕在一起。一旦解决一个难题，玩家再不会感到它反复无常且变幻莫测。Loom 游戏就是这种冒险类游戏之一。



12.4 难度

我们从游戏指南中可再次推断：Loom 游戏最初就设计为一个易玩的游戏。它确实希望玩家





成功，并希望玩家看到这个美好故事的结局。传统的冒险类游戏则乐于让玩家伤脑筋，玩家不得不一遍遍地玩游戏，才能在经历许多痛苦之后得到微薄的回报。

Loom 游戏防止玩家死掉或受阻，这是与其他冒险类游戏明显不同的地方。在很多冒险类游戏当中，即使玩家在刚玩的时候就忘记了做一些最基本的事情，也允许玩家升级。接着玩家就会到达一个位置，在那里得不到她需要的东西，也没有办法回头去寻找。由于玩家是不可能在游戏中升级的，因此，实际上她已经死掉了。但这是一种假死，此时玩家仍然可以与游戏环境互动，但却没有机会真正在游戏中获胜，这种状态比真正死亡更糟糕。Loom 游戏制定了一个标准：不要对玩家不公平。以后的许多冒险类游戏都纷纷模仿这个标准。

也有一些玩家报怨 Loom 游戏过于简单。那些曾经在以往的冒险类游戏中遭受种种艰难的热衷于游戏的人们也发现了这一点。他们已经习惯于在游戏的各个关口面临随时死亡的危险，并且可以花费数小时为那些几乎无法理解的谜团想破脑袋。至少在没有游戏攻略的情况下，许多玩家都无法完成游戏；对于这种结果，他们早就习以为常了。然而，仅为吸引那些游戏老手而制作的游戏，对于刚刚接触冒险类游戏的新手来说，则显得非常困难。如果玩家从未有过这些错综复杂的大脑受虐经历，游戏会令他感到十分灰心，以至于再也不想尝试玩另一个了。



12.5 故事

鉴于游戏机制把重心放在叙述故事上，如果 Moriarty 想讲述的故事不具备一流的质量，那么整个游戏就没有任何价值。幸运的是，这个故事的质量的确是一流的。被大家选出来的 Loom 孩子 Bobbin Threadbare 的任务是重回现实，关于他的故事是美好且富有寓意的。在 Bobbin 十七岁生日那天，他被唤到长辈们面前，惊奇地看到大人们都变成了天鹅。织布工 Dame Hetchel 曾经是 Bobbin 的母亲，她把不幸告诉了他。这个年轻的织布工必须找到 Loom 慢慢被破坏的原因，并且在结果变糟之前拯救它。这样，Bobbin 的冒险之旅开始了，他来到 Loom 的各类行会，将游戏逐渐地拉向一个特别的高潮，还以一个苦乐掺杂的结局作为故事的结局。在这个过程中，包含了许多商标和妙语连珠的 LucasArts 展现出的幽默（这种幽默风格在 The Secret of Monkey Island 中有最集中的体现），但他的幽默并未在故事中占支配地位。有些玩家也许会认为这个故事是面向小孩子们的，Loom 是个儿童游戏，就如同 The Hobbit 是一本儿童书，The Dark Crystal 是一部儿童电影，Bone 是一本儿童连环漫画书一样。它们都包含诡秘和智慧的成分，我们只要像孩子般天真诚实，就可以享受其中的乐趣，而不必非得是个孩子。



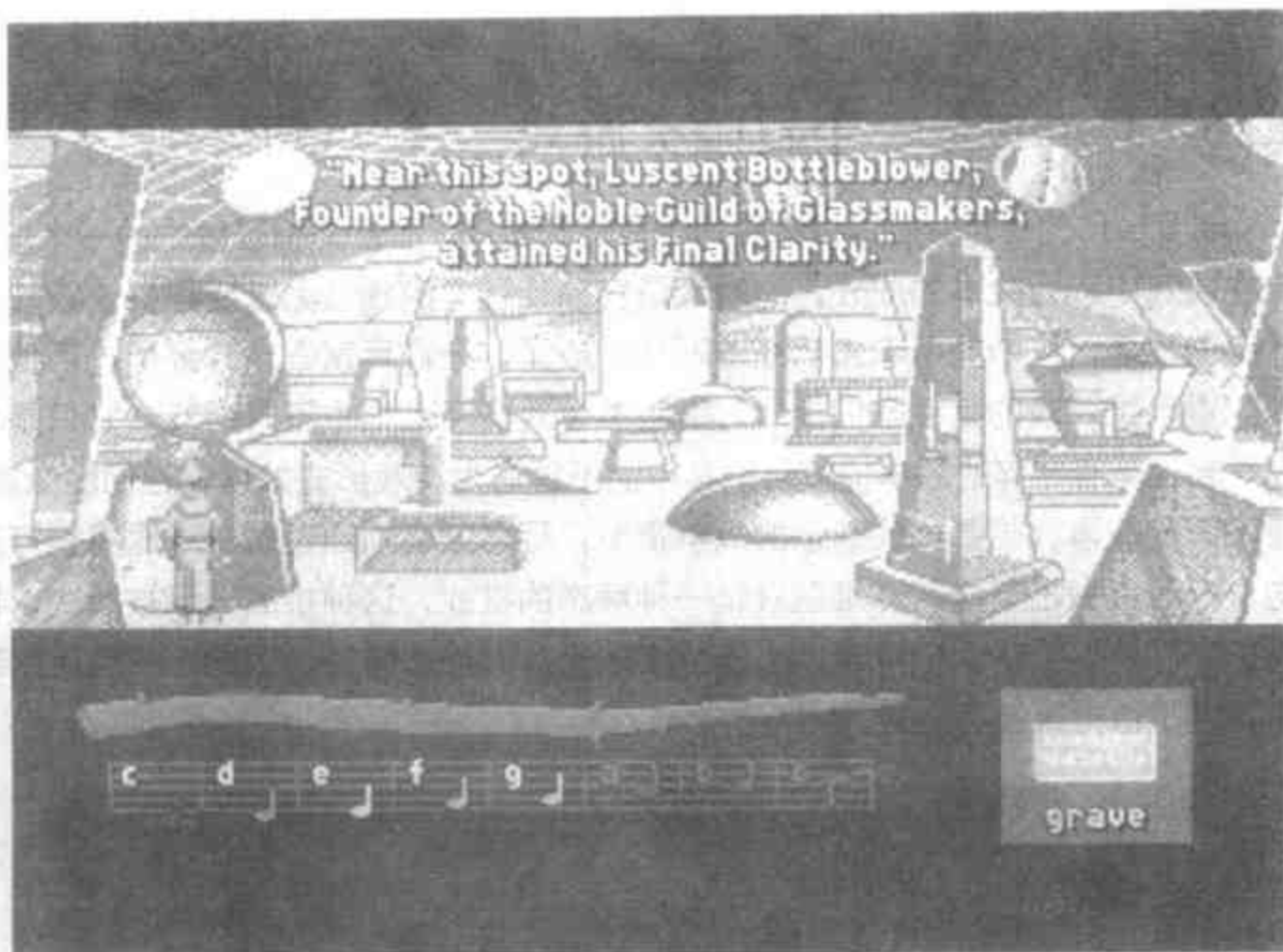


图 12.4 Loom 游戏之所以成功,很大程度上依赖于奇异怪诞的游戏故事

这个故事利用航海及旋转草稿等简单动作,完美地体现了 Loom 游戏的可玩性。同时,这个故事并没有像其他游戏故事那样,为了游戏的可玩性而使故事显得不自然。故事中的文字没有过分深入细节,并保持在最小限度,允许玩家用自己的想象填补其中的漏洞。通过玩家的探索和试验,通过与 Bobbin Threadbare 的性情相匹配的状态条,游戏叙述的故事是完全可见的。因为 Bobbin 在游戏初始时就获得了五线谱,所以他自然不是个游戏专家,这样玩家多次尝试使用草稿并遭受失败后,会发现它们与 Bobbin 的情况完全吻合。这一点与其他冒险类游戏不同,在那些游戏中,尽管玩家控制着一个聪明且经验丰富的角色,但玩家必须解决一些愚蠢的难题,例如在显然已知的情况下指出角色口令,从而将航海日志记入电脑系统。

第三人称冒险类游戏是指玩家可以在游戏中看到她的角色,而不是只看到这个角色能看到的東西。这类游戏的问题在于,所指的角色个性强烈,以至玩家很难适当深入于游戏当中。如果角色与玩家目前的状态相背离,说一些玩家不想说的话或有其他愚蠢的举动时,玩家就会很灰心。围绕这个问题, Loom 游戏是这样做的:它让 Bobbin Threadbare 穿着一件斗篷,玩家只能看见他的眼睛。这样做,使主角变成一位无名氏,实际上,玩家可以想象斗篷里面的就是她自己。Bobbin 在游戏中摘掉头巾的那一刻,游戏迅速切换到一个完全不同的场景,就像玩笑一样,使得主角仍保持无名氏的身份。Bobbin 讲话声音平板,又没有具体身份,他从不讲可能惹恼玩家的话。许多游戏发行者和开发商谈到,要制作健全的角色,这些角色以后可用于动作人物和电影实况。但是,能让玩家感觉愉悦的游戏中经常会有无名氏角色,这样,玩家可以在头脑中刻画出自己认可的勇士形象。





12.6 Loom 是一个冒险类游戏

尽管 Loom 有自己的优点，但它仍然是一个冒险类游戏，而且是相当线性的一个。冒险类游戏最关注传统游戏中叙述故事的成分，同时又最容易让玩家的创造力遭受挫折。冒险类游戏讲述的是设计者的故事，这个故事事先已经清晰地形成了，允许玩家去亲身经历，但他无法改变故事的结局。冒险类游戏的批评家们指出，冒险类游戏实际上根本不是游戏，它只是将一系列的难题串在一起，伴有零碎的故事情节夹杂其中。不管这类游戏形式如何，它们就像锁着的门，门中间夹着不同的故事，想要体验其余的故事情节，玩家就必须通过解决难题将锁打开。他们说，游戏应该对玩家做出反应，而难题只是一种静态的挑战，这种挑战一旦被解决，再试一次就没什么意思了。这些批评家们暗示，由于故事本身具有静态特性，因此，一旦玩家亲身经历了游戏故事，就不值得再经历一次。

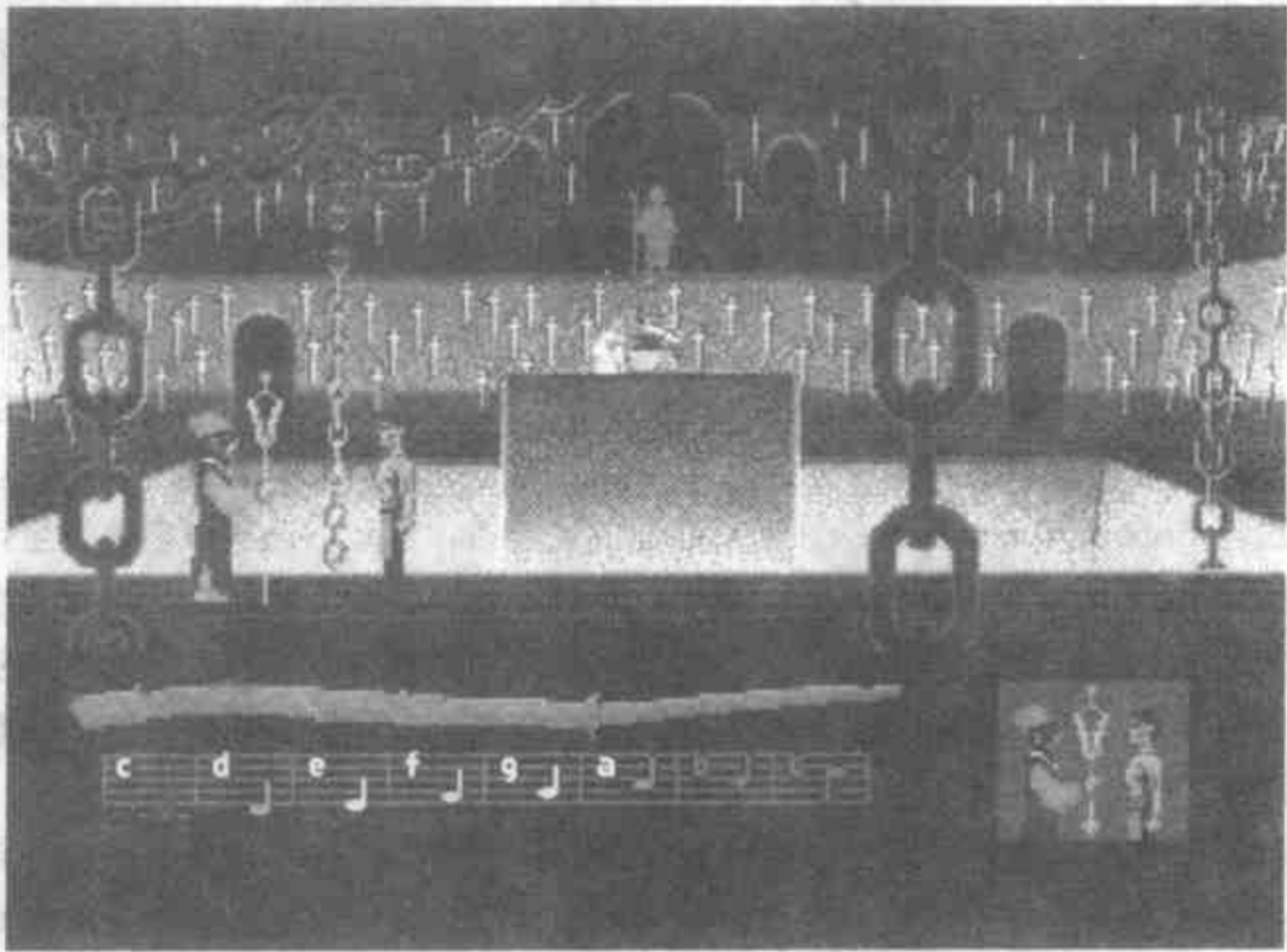


图 12.5 Loom 游戏以解决简单但却讲究的难题为中心，一旦难题被解决，就没有重玩的价值了

尽管 Loom 游戏在设计方面有自己的妙处和优点，但它仍然存在一些冒险类游戏共有的问题。在片头动画的对话过程中，游戏本身是完全线性的，玩家无论如何都不能控制游戏：如果次数较少，还可以接受，但 Loom 游戏中的一些片头动画持续的时间相当长。如果玩家试图点击屏幕上的某样东西，只想看看哪些物体可操作，那么，游戏有时会降级。如果一个物体可操作，这是玩家完成整个游戏的好机会。当然，这样做有好处也有坏处：好处在于限制玩家只做有助于游戏的事情，而不是扯些不相干的东西和进行无意义的转向之后指引他走错路；坏处在于限制了玩家与游戏环境的互动性。有时，从游戏的地形中很难断定 Bobbin 是否可以在某处航行。

然而，这些抱怨确实微不足道。难道 Loom 游戏是一个事先确定的故事，真的那么糟糕吗？



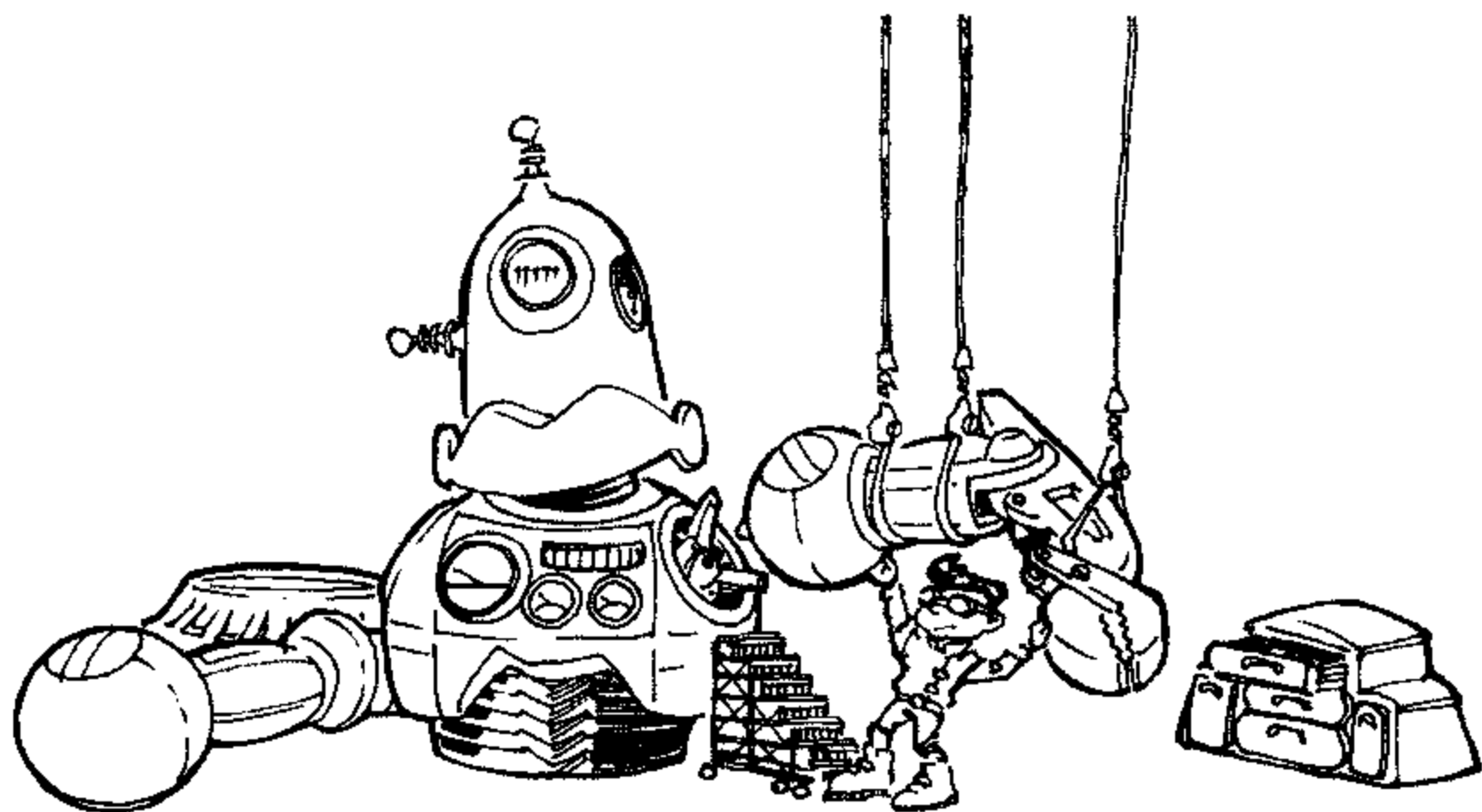


就像重读一本书或重看一部电影一样，这个游戏也值得再玩一次。当然，如果书和电影足够好的话，很多人愿意重复地读和看。Loom 游戏也许并不像 Civilization 那样具备互动性，但每个游戏都需要那种互动性吗？Civilization 游戏讲述了一个关于帝国壮大和技术进步的有趣故事，但是对我来说，从没有哪个游戏像 Loom 一样，讲述了一个可以激发玩家兴趣并令人感动的故事。这个故事其实可与 Civilization 游戏相媲美。批评家们也许会问，为什么不把 Loom 的故事讲述得像一本书或让它栩栩如生呢？当然，故事可以这样写，但如果是这样，当玩家探索游戏并与游戏环境形成互动时，他也能如此这般地深入其中吗？通过像 Loom 这样的冒险类游戏，玩家获得某种深厚的感情，并卷入一些在别的宣传媒介中不可能发生的事件中。也许 Loom 并不是一个完全另类的游戏，但这一点都未影响它的可玩性。



Chapter 13

第 13 章 实现设计方案



“那些想成功的人必须抛开杂念，满怀激情，探寻真正的联系和根本的主题。”

——Neil Peart

好莱坞有一种系统，这种系统非常著名，它有完整定义的目标，但最大的问题是“资金从哪来？”而不是“我们怎样制作电影？”好莱坞的制片商和天才们知道如何把故事变成电影剧本，如何对剧本进行多次修订，然后将可以把剧本制作成电影的人员聚集起来，既可以准时交工又不会超出预算。好莱坞总是或多或少地对最终制作的电影是否卖座有所把握，至少他们知道如何制作电影。当电影已经进入实拍阶段后，很少会重新修改剧本、裁减人员，或者无计划地向剧组中添加人手。通常情况下，电影在计划发行之前很长时间就已经完成拍摄工作了。有时候电影可能在定剧本阶段就搁浅了，或者完成之后，也不会如预期那样发行。但总的来说，好莱坞在制作电影方面有一套很高效的系统。

另一方面，电脑游戏开发者们却没有这样的系统。游戏设计的开始很混乱，无法预知的开发过程中充满了各种问题，即便是有经验的制作者、设计者或者程序员也都无法预见。通常，电脑游戏的开发可以持续到最后一刻，在将母盘送交制造商以前仍可能改动。对 PC 游戏来说，



在发行正式版之后，很快就会出现游戏补丁，因为游戏从来就没有真正完成过。为什么电脑游戏的开发如此难于预知，而电影的制作却是可以预知的呢？好莱坞制作电影的历史要比电脑游戏行业制作游戏的历史长得多，因此好莱坞的经验更多。但除此以外，好莱坞制作的更多的都是有计划的影片。不同的电影有不同的剧情和人物，在制片技术上也会有所不同，并且许多电影制作的工作量都是可以估计到的。

另外，每次制作的原创游戏都是全新的产品。部分问题在于转换技术目标，在每个项目中，程序员必须学习新的平台、操作系统和 3D 加速卡，而且很多游戏都需要先进的图形引擎。但纯粹从设计的角度来看，与其他同时期的游戏相比，原创游戏比同期创作的电影更具独特性。例如 Civilization、The Sims 或者 Doom。这些游戏中的游戏设置与以前的游戏有很大区别。许多游戏都比我刚刚列出的游戏缺乏考验和创新；在按时完成项目和不超出预算方面，遵守一些规则就会有更高的成功率。这样的例子包括 Infocom 的冒险游戏、Sierra 的冒险主题、运动类游戏每年的升级版，还有新版本的街机驾驶游戏。然而，在故事情节和界面上可能会加入新的内容，使游戏的可玩性与去年相同。当游戏尝试实现一种新游戏形式时，即使它只是在现有的主题基础之上加以修改，在开发中的可预知性也是可把握的。



图 13.1 Doom 所提供的游戏操控与以前的游戏区别很大，这个游戏的开发是一种大胆的尝试

只有真正优秀的设计者才能预知游戏中哪些内容会比较好玩，有经验的设计者会在设计出想要的游戏之前，使用大量原型、进行很多次试验，并反复尝试。这些天才的设计者们没有水晶球；他们只是在预测哪些内容会使游戏比较好玩的方面有很高的机率。他们的确不会比任何人知道的更多。

在开发原创游戏时，游戏开发过程中最可靠的办法就是，让游戏的一小部分先运转起来，然后再去开发游戏的其余部分。这一部分可以称为原型、演示版、对概念的证明、一个关卡，或者只是游戏的当前生成版本，而不仅仅是演示游戏技术的演示版。相反，它是在演示游戏的





可玩性，包括在游戏焦点中描述过的所有特性（如第 5 章中的介绍）。对于这个演示版，开发组中的每个成员都可以了解、可以玩，并且可以说：“是的，这个游戏很好玩，我想要玩这个游戏。”通过将注意力集中在游戏的一小部分上，以判断游戏是否可以正常运行或者是不是很好玩，开发者可以了解到最终制作出的游戏是否好玩。如果游戏的效果并不如所预想的那样，原型就可以警告开发者是不是需要将游戏重新指向一个更为有希望的方向，或者更糟——完全放弃这个项目计划。



13.1 灵活的过程

在我制作的游戏里，我倾向于尽可能灵活地控制开发过程，并且我总是不会对所有的事情都做出完整的计划。这种做法可能与大多数开发工作室的做法相反，但我发现这是最有效率的方法，可以制作出最好的游戏。因为游戏设计本身的不可预知性，所以采用灵活的进程可以给我留有更多的空间和时间对游戏的可玩性进行检验。我不需要编写复杂的文档，而是在向游戏中加入细节和深入的情节以前，首先使游戏的某些部分充满乐趣。如果不加以限制，过早地向游戏中添加太多的内容会造成很大的浪费。明确的细节可以采用详尽设计的文档、游戏对话的剧本、玩家可能走过的各个区域的详细地图或者以游戏的关卡等形式给出。在有很大的把握可以证明或者游戏的原型可以证明游戏会很好玩之前，制作游戏的这些元素是没有意义的。



13.1.1 太多太早

在制作剧本、文档或者关卡时，如果没有建好原型，则可能的问题是，这些东西会假设游戏能正常运行；而在实际玩游戏的过程中，假设可能会是错误的。如果设计者根据原则所制作的游戏设计文档最终是错误的，那么整个游戏设计方案可能就需要重新制订，或者完全放弃。如果人们花费大量时间来制作这些存在错误的文档，虽然心里明白，但仍会很不情愿丢弃这些方案。如果设计者过于沉浸于这些概念，即使后来证明这些概念并不可行，他也会坚持这些想法。毕竟，在进行了大量规划设计工作之后制作出来的设计文档，怎么能就这样丢掉呢？能不能再修改一下文档，然后就可以用了呢？如果你没有足够的勇气丢弃所有不合适的内容，最终制作出来的游戏就会需要补丁程序来修正，而不是一开始就有一个比较明确的方向。

当我开始制作自己的第一个发行游戏 *Odyssey: The Legend of Nemesis* 的时候，无可否认的是，我对自己要制作的游戏毫无概念。我使用了以前的开发者使用的游戏引擎和游戏机制的某些部分。那时，项目投入的资金少得可怜，因此，开发商只需要游戏进度的少量文档。我草拟了一份 6 页长的文档，其中简要描述了玩家会遇到的所有冒险活动。首先要提的是，这些文档并不是十分详细，关于游戏中每个主要的岛屿只有一页文档。这就使我可以控制的余地加大了。其次，在我完成了前两个岛屿的时候，我了解到了游戏进展的方向，决定丢掉最后三个岛屿，并重新进行设计。因为一开始我只编写了游戏情节的简要情况，所以实际上并没有丢弃太多的工作。



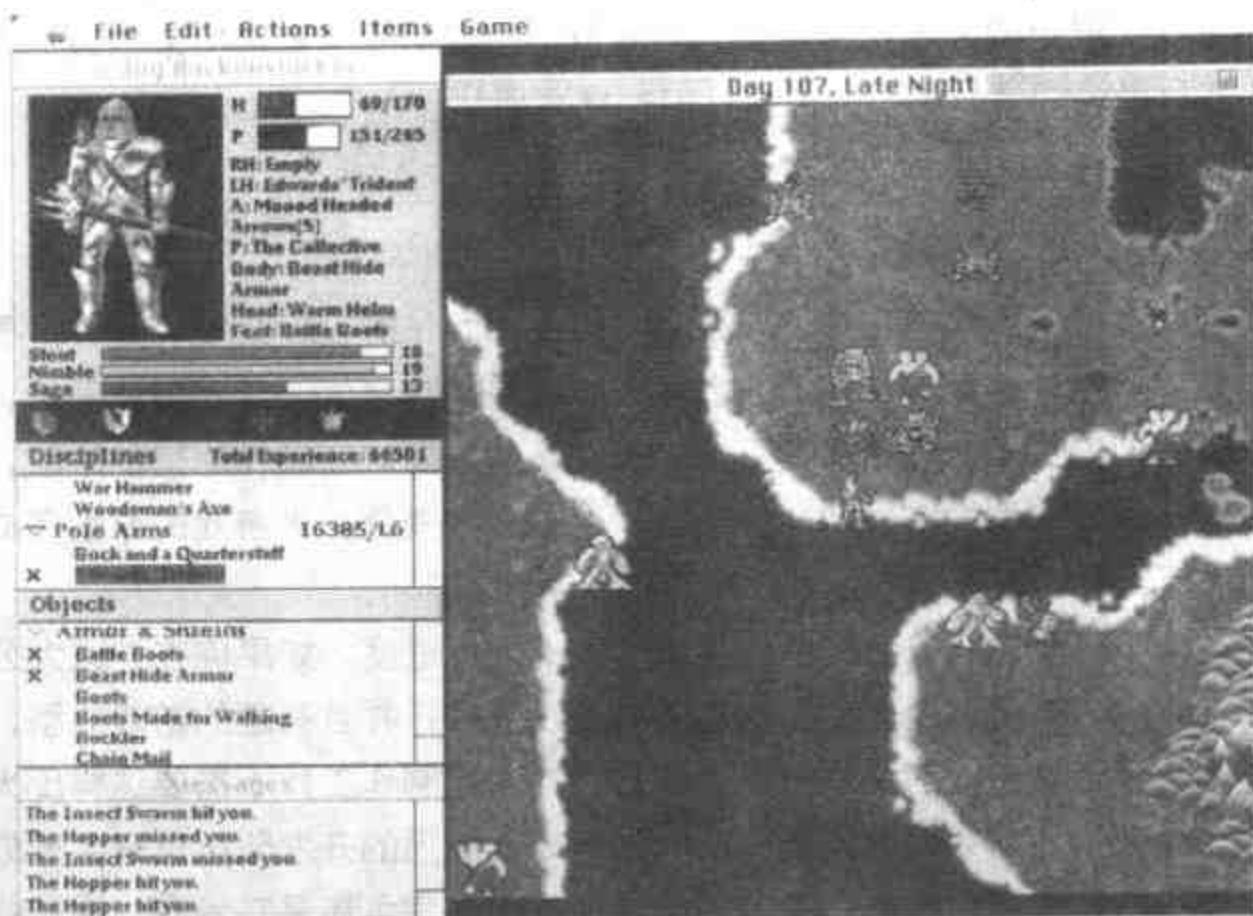


图 13.2 保持开发文档的清晰，并使用替换内容使 Odyssey 的开发过程非常灵活

在 Odyssey 这个游戏的创作中，另一个有趣的方面是，我在开发游戏的时候完全使用了替换内容的方法。使用游戏的引擎，我从另一个项目中继承了相当多的技巧，并尽可能多地使用。因为项目的资金投入有限，所以在游戏开发的大部分过程中，我并没有雇用美工，与其说做出这个决定是因为有先见之明，倒不如说是一种必要。然而，在我有足够的资金雇用美工来完成项目的时候，所有的游戏设计都已经完成，并且是完全可行的，因此美工在游戏中所制作的东西都会用得上。用替换内容的方法可以将对游戏开发进度的影响降低到最小。我首先将精力集中到让游戏可以正常运行上，然后再把注意力放在界面上。因为如果修改设计方案的话，我并不会受到已经制作好的界面的制约，并且可以在任何我认为合适的方向上修改设计方案。

在 Centipede 3D 中，重要的工作在游戏可玩之前就已经完成了，因此，几乎所有的工作都可以放弃。从游戏设置的角度来看，游戏原本的思路与如何实现 Centipede 之间毫无关系；其特点是更加多变，缺少方向性。使用这种原创游戏的概念，这个游戏实际上只生成了 6 个关卡，而其他的关卡都是在纸面上设计出来的。出于各种原因，游戏看起来并不怎么好玩，于是我们开始研究如何处理这个问题。最终，我们加入的敌人 AI 更像是原创游戏的敌人，并且对游戏设置进行相应的调整。当我们进行尝试的时候，并不确定这是否可以行得通，而后来证明游戏的风格是非常不错的。由于与新风格的游戏不能相互搭配，所以其他在纸面上设计的关卡计划都放弃了。在实际生成的 6 个关卡中，为了支持新的游戏风格，其中的三个也不得不放弃，而为了能够使游戏更好玩，其他的关卡也进行了重大的改动。

总结来说，如果在制作大量的关卡之前，我们把精力放在游戏的设置上，那么就可以避免不必要的工作。当游戏的可玩性证明可行之后，我们就可以制订文档，描述游戏其余部分的实现方法。对于大部分的开发工作，我们都可以在开发过程中遵守这些文档的要求，只需要少量必要的修改。当然，如果我们不能或不愿意丢弃已经完成的工作，那么对项目来说简直就是一种灾难。如果我们坚持使用所有的关卡设计，而不对其做出修改，那么游戏就会表现出这些关





卡比那些正确游戏设置中的关卡要差得多。如果我们愚蠢地完全坚持原来的设计方案，整个游戏就会失败，并且最终的产品也不会像预计的那样有趣。



13.1.2 保持简单

在开发的初期，把精力集中在焦点问题上，比注意冗长的设计文档要更为重要。焦点问题比较简单，如果游戏设计改变方向的话，就可以很容易地完全修改。而与此同时，对于你努力在游戏情节上要达到的目标，焦点问题可以给出明确的方向。在制作原型的阶段，当转变游戏的目标以符合游戏所要得到的结果时，焦点会发生很多次变化。当游戏的原型完成之后，将会有一个固定的焦点，可以在其余的游戏开发过程中遵循。

遗憾的是，你不可能总有机会灵活地控制游戏的设计进度。如果你在一家大公司里工作，那么从项目的一开始，就会有庞大的工作组，即使没有可行并且有趣的游戏原型，那些人也要忙碌于制作界面、建造关卡或者编写代码。最开始的游戏制作工作不需要这样庞大的团队，而实际上，当你想让他们了解如何制作游戏的时候，如此庞大的开发队伍也只会形成一种障碍。你还要听候提供项目开发资金者的命令，而无论他是你的雇主还是开发商。在游戏的原型开发完成之前，那些付钱的人就想看到完整的设计文档或剧本。当游戏所产生的结果和预想的有所不同时，你可能要被迫放弃那些文档。很明显，过早地编写这些文档会非常浪费，而你还要一直对提供项目资金的人心存感激。如果有可能的话，在制作好完全可行的原型之前，最好能自己出资金对项目提供支持。如果你是在一家大公司工作的话，可以在管理人员的管辖之外制作游戏，或者在找到开发商之前就做好游戏的原型。除此以外，提供游戏的演示版本可以更容易地将游戏卖给开发商或者许可委员会。没有什么可以比吸引人的原型更能向这些提供资金的人证明你的游戏是值得投资的。



13.2 制作游戏

制作游戏的最佳方式是采用渐进式的方法。不要在游戏所有的部分上都只完成少量工作，应该先完成一个系统，然后再转向另一个系统。首先完成最基本和最重要的系统，然后在这个系统的基础之上生成其他的系统。这样就可以先实现某个系统，对其进行测试，看它的运行是否正常，在所有的测试全部通过后再转向下一个系统。以这种方式，如果必须修改底层的系统并使其能够正常运行，后面的系统就可以进行相应的改动。当多个程序员同时编写在一起运行的系统时，很有可能会导致一场灾难。如果其中一个系统需要修改，那么其他的系统也需要彻底的返工。在根据这个系统生成其他系统之前，最好打下一个坚实的基础。程序员经常只会注意自己的代码，而不会考虑这些代码如何与项目的其他部分进行配合。开发组必须能够始终全面地考虑问题，这样才能制作出好玩的游戏。



13.2.1 核心技术

当然，所有的电脑游戏都要依赖于那些与游戏毫无关系的底层技术，这种技术通常指的是





游戏的引擎。在游戏的所有开发工作开始之前，要保证这种底层技术可以在某种程度上正常发挥作用。然而，在生成原型之前，却不需要使引擎非常完善或者具有完整的特性。实际上，在使用先进的引擎技术来制作项目之前，如果等待引擎技术全部完成的话，就没有足够的时间对游戏本身进行改进了。使用不熟悉的技术开发项目的危险就在于这种技术所能实现的功能。如果在设计游戏的时候认为屏幕上可以同时显示 10 个敌人，而结果表明这种引擎只能处理 3 个，那么就需要对设计方案做出很多修改，这样才能适应这种限制。有一点可以肯定的是，设计得最好的游戏在发行的时候都不会使用最先进的技术。

如果还没有选定使用哪种技术，据我所知，有很多游戏设计者都会使用前一个项目中所使用的技术来制作原型。虽然技术可能会影响游戏本身，但实际上不会影响游戏的设计。虽然技术有时也会让人无法估计，但比起游戏的设计方案来它更具有确定性，所以在制作游戏的原型时没有必要担心技术的问题。因为你所制作的前几个阶段的项目无论如何都是会丢弃掉的，所以使用一种最终也会放弃的技术来进行制作也不会造成浪费。



13.2.2 渐进的步骤

一旦技术达到能够开始开发游戏的阶段，就可以先确定游戏设计中需要完成的最基本的任务，然后再根据这些任务逐步进行制作。例如，假设你正在制作一个动作类游戏，在游戏中，玩家可以控制类似于人的人物在采集猕猴桃的时候使用苍蝇拍与一个主体作战。处理玩家的导航系统从逻辑上来讲是第一个需要解决的问题。首先要让人物可以前后移动和转向，这是游戏中基本的导航。要对这种动作进行改进，直到感觉良好，发现自己喜欢使用这种简单的、只有导航功能的方式来玩游戏为止。现在你可以在此基础之上添加更多的动作选项，如平移、卧倒和跳跃。在添加每种新的动作类型时，注意一定不要破坏以前的动作，只有这样它们才能在一起协调运行。只有在这些动作都能够正常运行之后，玩家才具有使用苍蝇拍的能力。在能够使用苍蝇拍之后，至少也是以某种有限的方式，才可以将对手加入到游戏中。AI 的功能可以分为各个生成块来完成，就如同玩家的动作一样。首先，要在游戏中加入 AI 主体，这样玩家才能使用苍蝇拍来打它们。下一步，在添加功能使它们可以进行“检查”和“纸上谈兵的”攻击之前，要让 AI 主体可以在游戏中移动。最后，可以向游戏中添加猕猴桃，让玩家可以摘下它们，然后用苍蝇拍发射出去。在这种一步一步进行开发的过程中，最重要的是，在每个步骤中游戏都可以玩，而且还会很有趣。当向游戏中添加了会破坏前一部分的内容或者使游戏的趣味性降低时，就必须立即解决这个问题。在游戏投入制作之前，就已经是修改设计方案的时候了。

在这个项目的开发过程中，我认为一直保留一个游戏的可玩版本非常重要。通常，开发组会忙于编写游戏的各个部分，而不会提供一个让人可以坐下来玩的版本。当游戏长时间处于无法运行的状态时，你就会很容易丢失游戏所要达到的目标。当然，游戏可以使用很多方式进行分解，各个部分不会如设想的那样运行，并且在很多地方使用替换方法。但只要有一个可以运行的游戏，开发组的成员就可以玩这个游戏，并且他们可以了解到自己的工作，以及自己的工作会对游戏产生什么影响。如果在添加或修改了什么内容之后使游戏不如原来好玩了，要能立即发现这个问题并解决它。





13.2.3 完整的功能范围

一旦让游戏机制中的各个组成部分运行起来，而且你也对它感到满意之后，下一步就是让游戏的整个部分可以像最终希望的那样运行。在很多游戏类型中，这表示游戏的一个特殊阶段。你可以认为自己已经具有了游戏功能的所有组成部分，但是一旦让整个游戏运行起来，你很快就会发现忘记加入的内容或者没有达到预计要求的地方。在转向其他阶段的制作之前，要将精力集中在这个阶段，使其尽可能地接近游戏的最终状态。如果你的观察能力比较强，那么在这个阶段中，你就会吸取到关于如何才能让关卡的设计适合于这个游戏的很多教训。一旦度过了这个阶段，你就会觉得现在制作好的游戏并不是能力范围内所能做到的最好的；你会学习到很多东西，以后制作的关卡就会比刚开始时制作的关卡要好。虽然现在还不需要丢弃原型，但要记住，在完成游戏之前，你可能要对游戏进行重大的修改。

这样的一个例子来自于我的 *Damage Incorporated* 游戏的开发过程中。在完全了解游戏的机制或者使用那些工具来制作关卡之前，在第一个关卡中，我完成了单人游戏的制作。但结果是游戏并不好玩，所以这个阶段的工作很快就被丢弃了。我所制作的第二个关卡虽然不是游戏中最好的部分，但足以制作成成品游戏。游戏还包括了死亡模式的网络作战，其中使用了完全不同的关卡。因为时间的限制，我在谐调网络作战方面花费的时间较少。尤其是我为网络游戏制作的第一个关卡“*My Mind is Numb, My Throat is Dry*”，最终的结果并不是很好玩。其中又有很多比较吸引人的地方，但却没有很好地配合在一起，关卡中的许多部分都不太合理，是一些杂乱的死亡陷阱。我的一个游戏测试人员甚至建议说最好丢掉这个关卡，并从头开始制作一个新的。遗憾的是，我没有时间来重新制作一个新关卡，这个糟糕的关卡最终随着游戏一起发行了。而幸运的是，游戏中还有其他 7 个比较有趣的网络关卡。尽管如此，如果我完全放弃在网络关卡上的第一次尝试并且更换一个新的关卡，情况可能会更好。

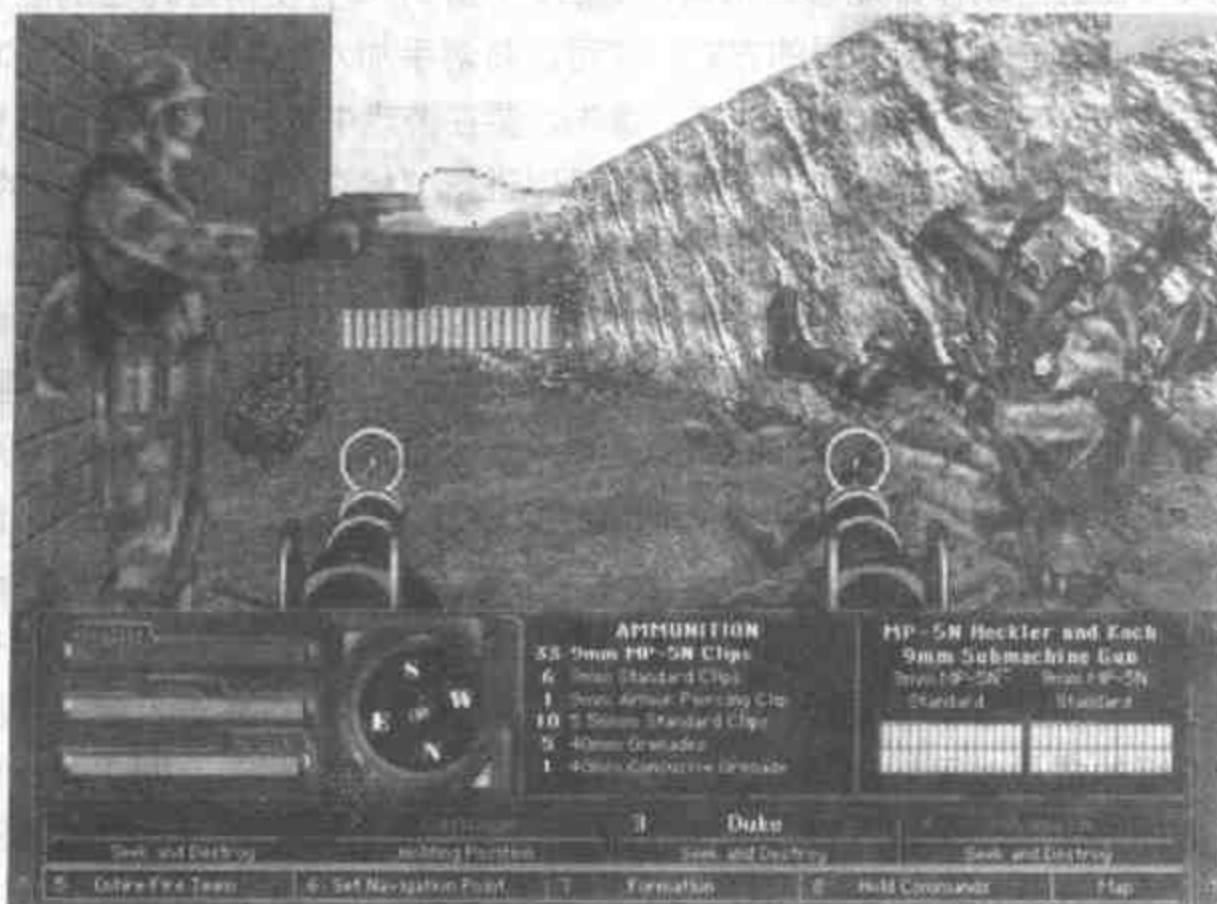


图 13.3 该图是在 *Damage Incorporated* 中制作的第一个网络关卡，也是游戏中最糟糕的一个关卡。最好是丢弃这个关卡并重新制作一个新的



当制作游戏的第一个完整可运行版本时，需要知道的是，游戏玩起来时到底有多难？游戏的难度通常可以进行调节，并在以后的开发过程中、游戏测试和谐调过程中逐渐调整。当需要使游戏原型可以运行时，应该在难度方面加以权衡，以保证初级玩家也可以玩这个游戏。介绍一些朋友或者同事，让它们也来玩游戏。注意它们进入游戏时的难易程度。如果你发现游戏比预计的要难，那么就应对游戏的设计方案进行修改，以降低游戏的难度。

13.2.4 经历多次修改

灵活的游戏设计过程中，一个比较重要的部分就是，肯于抛弃自己的工作，或者是开发团队中其他人的工作。其中包括美工、代码、关卡，甚至是总体设计本身。当修改游戏的设计方案时，所有的游戏内容都有可能需要修改。游戏的某个部分本身可能并没有什么缺陷，但是如果它与游戏在整体上配合得不好，就要去掉这个部分，并从头开始制作。许多开发者不愿意这样做，所以这样的部分出现在了他们的游戏中。他们的游戏受到了最初的设计文档的束缚，结果游戏运行的实际情况与理论并不相同；或者在最终产品成型以前，游戏变成了一个大杂烩。一旦设计者认为游戏的开发方向需要修改，游戏的所有部分都必须重新进行评估，以判断这些部分是否可以适应新的需要。如果不能的话，就必须返工。

我曾提到过，我的项目 *Centipede 3D* 在开发过程中做了很大的改动，结果使我们不得不丢弃大量的工作。幸运的是，开发队伍中没有人对这种做法感到不满，所以我们都认为这是这个项目最大的收益。在我以往参与过的项目中，我曾经非常地固执，当同事和朋友提出需要返工的请求时，我总是不予理睬。我很不情愿丢弃已经做得很不错的作品，即使这些部分并不适合于这个游戏时也是这样。有时，解决游戏中存在问题的第一步就是，要承认游戏中的确存在问题。

当然，你必须注意不要在其他的方向上走得太远，以至于防得抛弃那些不需要丢弃的内容。当你参与某个项目的时候，对于自己制作的内容会非常熟悉，这种熟悉会导致忽视某些问题。例如，制作出某个关卡的很长时间之后，设计者很可能非常不愿意日复一日地重新查看相同的内容。设计者可能会觉得需要重新制作那个关卡，并不是因为真的需要这样做，而是因为需要某些新的内容。这是一种浪费，因为玩家是第一次玩这个游戏，所以关卡的内容是很新的，也是非常刺激的。对游戏的修改不能是为修改而修改，否则就会花费更多的调试时间、延迟项目的交付，并且开发队伍也会非常不解：为什么已经完成的工作要丢弃，而且还要重新制作？

第一印象非常重要，在游戏设计方面更是如此。回忆一下，当你在玩某个关卡时，或者尝试进行某种移动时，自己的第一感觉如何。是太难还是太简单？很直观还是比较复杂？长时间地从事于某个项目的最大问题就是：设计者会在设计中习惯性地犯错误。游戏的控制可能不太直观，或者某个敌人以十分独断或者不公平的方式攻击了玩家。当他们重复地玩游戏时，设计者就会学会克服并避免在游戏设计中的这些问题，使他们得到了一些错误的印象，认为游戏并没有什么地方出了错。游戏测试是用于在游戏的设计中查找弱点的一种重要手段，开发队伍已经逐渐采用这种做法，有关游戏测试的内容我会在第23章“游戏测试”中介绍。然而，在进入游戏测试阶段之前，要记得自己对游戏某方面的第一印象。问问自己是否发现并解决了问题，或者是否问题依然存在，是否会对第一次玩游戏的人造成迷惑。最好在发现这些问题的时候就





尽早去解决，因为如果留下这些问题的话，就有可能遗忘。



13.3 编程

本章是从对设计者和程序员的可利方面来考虑的，因为在我所有的项目中，我都是充当了这些角色。尤其是在能够体验游戏方面，这样可以具有某些独特的优点。设计者/程序员能够了解游戏的主题，并迅速实现自己想要达到的目标。自己不能编程的设计者不得不首先与程序员交流游戏的思想，并且希望他可以理解设计方案。这种交流经常会停滞不前，设计者也不能确切地表述他的目的：设计者的设想实现起来可能并不太理想。因此，游戏可能会存在不足，或者设计者必须找到程序员，并向他解释某个特性具体要达到什么效果。因为游戏设计是个反复和试验性的过程，所以在设计者和程序员之间必须经常有信息反馈。很明显，如果设计者和程序员是同一个人的话，这个过程就会大大简化。

我经常发现，作为一个设计者，如果自己亲自去编写程序的话，就可以更容易地尝试某些想法。实际上，我认为自己想出的许多想法都很糟糕，不足以让别人来进行制作，因为我对这些想法缺乏信心，而交给别人去做只会浪费他们的时间。但最终结果证明这些想法在游戏中的表现很好，如果我无法亲自从事编写代码的工作，那么有些想法就无法尝试。

设计者/程序员还能够更好地理解项目中所涉及到的技术，并且可以判断哪些技术可以很容易地实现，而哪些技术却不可以。不会编程的设计者可能会设计出在引擎中非常难于实现的游戏方案。虽然游戏的类型不同，但是如果在技术上是相同的话，使用某种游戏技术就可以运行得更好。如果设计者/程序员注意到了这个问题，就可以大大简化游戏的开发过程。假设某个设计者需要一把剑，要让这把剑使用一种特殊的方式与向它施加魔法的玩家交流。设计者可能需要这把剑能够在玩家的手里折弯。接受这项工作的程序员就会诅咒那个设计者，因为出于游戏中所使用引擎的限制，这个功能实际上是不可能实现的。虽然设计者也知道制作一个奇异的物体系统的做法也很不错，但是他并没有意识到这样的系统实现起来会更容易。结果是程序员害怕拒绝设计者的要求，他会花费大量的时间来实现这个功能；如果设计者能够更好地理解这些技术的话，那么选择一种更为容易实现的方案就可以令他满意。了解游戏编程的基本原理和游戏中所使用引擎的结构，可以帮助设计者了解想法的可行性。即使你不是项目中的主要编程人员，也可以更好地理解使用这些技术能够容易地实现哪些功能，并且知道哪些特性会长期占用大量的开发资源而不会对游戏的开发过程产生帮助。

当设计者和程序员对于游戏的概念有不同的理解时，这时又会出现另一个问题。我听一个设计者称之为“搁置否决权”。设计者可能会找到程序员，向他解释游戏特定部分的运行方式；如果程序员不同意这样做，他就可以不去实现设计者的要求。他甚至可以托词说设计者的要求难度太大，或者说实际上不可能实现。自己不能编程的设计者不得不接受反复无常的程序员们的奇思异想，这一直都是很令人沮丧的事情。

我的观点是，如果你想成为设计者的话，就值得去学习如何编程。实际上，这就是之所以我最初从事编程工作的原因。本书并不会介绍如何编程，有很多书籍可以找到那些在工作中需要用到的知识。有效的编程实际上就是训练的问题。你甚至不需要找一个相当好的程序员来帮





助你实现设计方案。实际上，几乎所有我认识的设计者/程序员都说自己并不是非常好的程序员，但他们的能力却足以完成想在游戏中实现的功能。正如我所说过的，如果懂得编程，将会对游戏中实现某些功能的难易程度有很好地了解。而且，如果需要以某种特殊的方式来实现游戏的设计方案，那么惟一的方式可能就是自己去编写程序。

如果在项目中并不是自己编程的话，那么找一个非常了解游戏概念的程序负责人是很重要的，这个人的意见你是可以信任的。实际上，有人会建议你只让那些开发组中的成员了解如何可以让游戏变得有趣。最终，只有程序员们做出的少量决定可以对游戏产生深远的影响，而这些都是设计者未曾设想过的细节。这些小细节会对最终制作出来的游戏产生很大的影响，决定着玩家在玩游戏时的“感觉”。那些不主动或者对事情漠不关心的程序负责人会支持在理论上比较好的主意，但结果并不是那么有趣。许多项目最终制作出的产品都不令人满意，因为程序员只是从一些规范的角度来实现各种特性，而从来不考虑游戏的整体效果。

本书中包括了 6 位游戏设计者的访谈录，他们都是游戏行业历史上最为出色的游戏设计师。值得注意的是，这些设计师们在职业生涯中都曾经做过程序员的工作，并且在他们所制作的最流行的游戏中参与过编程工作。实际上，早在电脑游戏行业的发展之初，开发过程的规模很小，通常都是一个人完成所有的工作，所以不需要把角色划分为设计者和程序员。尽管如此，访谈话题中的三个设计师仍是开发项目中的编程负责人。这并不是说不做程序员就无法成为优秀的设计师，但是我认为可以自己编程的设计师比那些不能编程的设计师更具优势，这个优势使他们可以设计出更好的游戏。



13.4 什么时候会有趣

让游戏可以运行是游戏设计过程中最重要的部分，而这也是最难于解释或者向别人传达的部分。而且对于理解游戏以哪种方式会更加充满乐趣的方面，没有哪本书可以解释清楚。实际上，游戏的设计在实现阶段经过了许多改动，以至于我并不认为在这个阶段中并不积极的设计者是真正地完成了游戏的设计。如果这个所谓的设计者只是敲出了 200 页的设计文档，然后交给编程负责人去实现，而自己去博拉博拉岛游玩，那么编程负责人就要负责做出一些重要的判断，而这些判断决定着游戏是好玩还是无趣、是刺激还是单调、是令人愉快还是让人乏味。当在实现过程中设计者不辞而别之后，编程负责人实际上就变成了游戏的设计者。

实现游戏设计方案的很大方面是个人“能力”的反映，毫无疑问，为别人设计游戏是有很大的难度的。这就是为什么需要游戏面向“大众市场”的原因，而这些游戏都是由那些专业级玩家设计的。从事设计工作的专业级玩家希望自己制作的是 *Grim Fandango*，但实际上却在制作 *Advanced Squirrel Hunting*。即使他能够克服对于项目本身的轻视，他也可能对于玩家在 *Advanced Squirrel Hunting* 游戏中感兴趣的方面毫无概念。游戏的特性是根据市场的需要添加在游戏中的，而完全不顾开发人员的反对。这些特性通常都是游戏中最糟糕的，不一定是因为其想法糟糕，而是因为开发人员不能理解为什么要向游戏中添加这些特性，或者他们并不知道如何提高玩游戏的性趣。最终就很难设计出好游戏，甚至自己都不愿意玩。如果你不想玩的话，其他人同样也不会想玩，而这些人正是游戏所要面向的玩家。



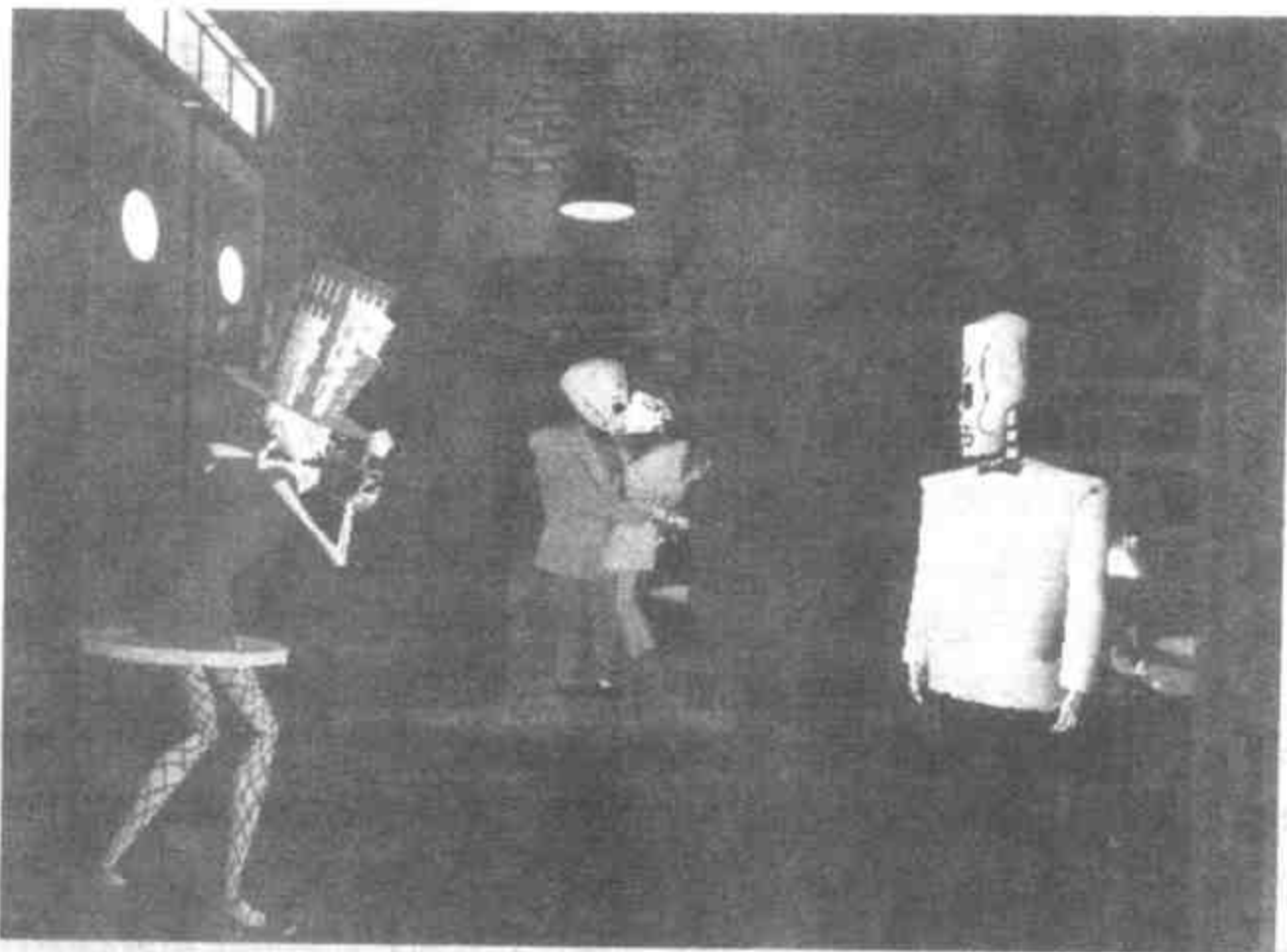


图 13.4 当游戏的开发者对自己所制作的游戏感兴趣并且愿意玩的时候，他们才能做好工作。Grim Fandango 就是这样一个极好的例子

设计游戏的第一步是让游戏可以运行起来，并且可以让人来玩这个游戏。一旦制作好了可以玩的原型，保证这个原型很吸引人，并且在某种程度上非常好玩，那么就应该回过头来，使自己知道如何使之充满乐趣，而且要知道如何将其扩展到游戏的其余部分。使用这个原型作为模型，你现在就可以开始制作游戏的其余内容，复制游戏中基本的特点，而保持新的有趣的内容。既然已经知道自己的游戏设计方案已经很完善，最后就可以构思出完整的设计文档，并可以解释在游戏的其余部分如何设计游戏和做出改进。这样也可以向其余的开发人员提供非常有价值的信息，并对游戏加以充实。从某种程度上来说，游戏的原型一旦完成，真正有创造性和挑战性的部分就完成了，并且游戏开发的其余部分也就只是简单的重复性工作而已。



Chapter 14

第 14 章 Chris Crawford 访谈录



现在，Chris Crawford 以他对电脑游戏之中对话设计方面的杰出贡献而闻名，其中包括他发起组织的电脑游戏开发者论坛、《Journal of Computer Game Design》的出版，《The Art of Computer Game Design》一书的编著。值得一提的是，《The Art of Computer Game Design》一书虽然是在 1983 年编写的，但是它却包含这一领域中最出色的东西，即这本书的精髓所在。包括像 Eastern Front (1981)、Balance of Power 这样无可争议的经典之作，以及 Crawford 自己喜欢的游戏 Trust & Betrayal: The Legacy of Siboot 在内，Crawford 的杰出之处同样是不可忽视的。在 90 年代的大部分时间里，Crawford 全身心地投入到他所热爱的工作中，即对交互式故事介绍系统的研究中，该系统叫做 Erasmatron，这种工具可以将拟定好的交互式故事转变为真实生动的经历。



问：最初是什么吸引了你用计算机制作游戏？

答：那要追溯到 1966 年，在我中学二年级的时候，一个叫大卫的朋友向我介绍一个叫做 Avalon Hill 的游戏。我们玩过后觉得很有趣。虽然在读书的时候我没有太多的空闲时间，但我还是忍不住这样做。到了我读研究生的时候，我遇到了一个在计算机中心工作的人，他正要在



电脑上玩 Blitzkrieg，这也是一款 Avalon Hill 游戏。我问他是不是疯了。我说：“别幻想了，那是不可能的！”但是那件事却从此埋下了种子。我仔细地想过这件事，一年之后，我决定试一试。从此我就开始研究，结果并不像我所担心的那么难，我用 FORTRAN 语言在 IBM 1130 上编了一个小程序。实际上，这个程序就是一个电脑游戏，即小型的装甲模拟战。那个游戏最初的面世是在 1976 年，在一个小型的电脑游戏洽谈会上，我将它介绍并推广出去。那一次，大家都玩了这个游戏，而且得到了好评。后来，我又买来 KIM-1，又在那个系统环境下重做了游戏。几年来，那次的游戏都是其他游戏不可比拟的，因为玩家可以进行真正意义上的隐蔽移动。大约在 1977 年或 1978 年，我又在游戏中设计了一些小站和分界线，使得每位玩家都有自己的领地。两个玩家打游戏，彼此注意对方的位置，其乐融融。



问：最初是什么让你认为这是不可能的？

答：应该是那时候在运用人工智能方面的困难。我想：“那几乎是不可能的。”鬼使神差，我忽然觉得那也许是可行的，而实际做起来也并不像担心的那么难。那个时候我总认为在电脑上制定装甲战略计划似乎有些可笑，现在你要知道，那可是在 25 年前，即使有 AI 的话，我也会认为那是实现不了的。但是，正如所发生的，我在不到一年的时间里解决了问题！



问：是什么让你认为在电脑上做游戏是值得做的事情？

答：让我这样想的是一件事，那就是捉迷藏游戏。对此我很好奇，无论你看怎样，在游戏里你总是能知道其他人在做什么。为此我也确实感到有些困惑，因为那根本是不切实际的。说起来似乎有些不可思议，我想到如果借助这种方式，游戏会更有意思。而实际上，当我们做这类游戏的时候，这些游戏充满了无限神奇的力量，要比传统游戏有趣得多。当我认识到这一点，就明白了——这就是我一定要研究的东西。但是，这种游戏的技巧还从来能适应那时游戏的某些简单方面。正如人们所说的情景：从对手身后偷偷溜走，笑话他那 20 公斤重的排气管，多有趣呀！这些实在是很好的素材，实在让人兴奋！



问：那么，从那段早期工作之后，你是如何到 Atari 工作的？

答：实际上，最开始我有一个 Commodore Pet，用 BASIC 和汇编语言子程序编了个程序。我在一些战争游戏洽谈会上展示了我的战略装甲游戏，并给每个人都留下了很深的印象。因此我把 Tanktics 作为商品销售，售价为 15 美元。在那之后，我又用 Commodore Pet 做了一个游戏，叫做 Legionnaire。为此，我在 Atari 公司得到了一份游戏设计的工作。实际上，我是那家公司仅有的几位有游戏设计经验的候选人之一。一切真是太微妙了！而电脑游戏本身便是如此。



问：那时 Atari 的氛围如何？

答：令人兴奋！对现代的人们来说，很难理解 20 年前的一切是多么的不同。至今我还记得和 Dennis Koble 的那次会面。那是在一天早上，要上班的时候，我们在停车场相遇了。路上，我们聊了很多。还记得我说：“相信吗？有一天，游戏设计将会成为很有发展的职业！”他说：





“是呀，或许有一天我们会像摇滚明星一样有名气！”我们都为这个可笑的想法大笑起来。当时，世界上的游戏设计师大部分都在 Atari，这些人彼此都认识，至少在 Atari 的人都互相熟悉。在那儿，因为许多人都不把自己看作是设计师，所以气氛非常和谐。

举个例子：我记得有一次，在一个会议上部门经理说：“请大家注意，我们准备为大家印制一些新名片，现在要确定使用什么头衔！”于是，引起了一些职员们的争论，名头是叫做“游戏设计师”呢还是叫做“程序设计师”呢？我还记得有人说：“啊！如果我们的头衔叫做‘游戏设计师’的话，有可能以后找不到另外的工作了！”而我认为，我们的职业最终会称为“游戏编程师”。但是，在那时，所谓的游戏设计和现在的意思大不相同，那时的概念有些模糊。因为我记得有一次有人问我：“你做什么工作？”我这样告诉他们：“我给 Atari 设计游戏。”他们又说：“哇！真是太奇怪了，你是如何工作的？”可以看出，在当时，那是个令人费解的职业。



问：在游戏设计方面，你能任由自己发挥吗？

答：那要取决于你正在做的事情。如果你在设计 VCS [Atari 2600] 游戏的时候，和你的主管讨论游戏的设计，他会给你很大的自由空间。他会告诉你：“总之，我们需要大量的游戏，而且确实需要创造性。因此，你只要跟着感觉走，看一看怎么做效果好，看一看你能否捕捉到有趣的素材。”一般情况下，除非你正在做的游戏是从别人那里窃取来的，否则你的主管都会给你很大的自主性。所以说，在这方面我们有很广泛的自由。而且一旦完成设计，其他设计人员都会仔细地看一遍。他们不是从销售的角度，而是从游戏设计的角度做一番评论。

大家都想用计算机 [Atari 800] 进行设计，因为它要比 VCS 好得多。因此，在 1979 年我打算为此而设计的时候，曾有这样的规定：首先要在 VCS 上完成一个游戏设计。只有这样，才能进行计算机设计。为此，我还发了一些牢骚，因为我根本不赞同这个主意。可是我还是学习了有关 VCS 的知识，之后还编了个游戏。然而，又有这样一条规定：游戏的设计要在 2K ROM 的条件下运行。那时，4K ROM 刚刚投放到市场，而且价格不菲。他们的意思似乎也是：“不要使用 4K 的 ROM！我们要知道你能在 2K ROM 的条件下设计成功！要让我们给你施展空间的话，你就必须证明你自己是一个有价值的设计者！”因此我只好做了个只使用 2K 的游戏。

我设计了一个叫做 Wizard 的游戏。这个游戏在 2K 游戏中算是很优秀了。可是虽然我完成了设计，但不巧的是，还是赶上了 Atari 正要推出 4K 游戏之际。大家都意识到 4K 游戏并不是像原来估计的那样稍稍胜过 2K 游戏，而是远远胜过 2K 游戏。因此大家都认为，准备销售的游戏都要和 4K 游戏进行比较。而作为 2K 游戏，我的设计根本无法与其相比。所以，其他的设计者最后只好说：“对于 2K 游戏来说，这个设计很不错，但是将来的盈利不会大。”他们希望把它重新设计为 4K 游戏。我本应该重做这个游戏并发布，但是我想：“我已经在 VCS 上设计出了游戏，现在我倒乐意进行计算机游戏设计，那就别在这个问题上绕圈子了！”所以我争辩说：“这本来就是作为 2K 游戏而设计的，我不会在上面加什么东西！如果你们要 4K 游戏的话，自己做吧，那是惟一的方法！”我嘟囔着，悄悄从人群中走掉。也正是这样，我才能进行计算机 [Atari 800] 的设计。因此，那个游戏从未发表过，而我没有任何遗憾。





问：你在 Atari 最成功的产品是 Eastern Front (1981)。但是我也理解，你在让人们相信战争游戏一定会取得成功方面遇到的困难。当时你确信人们会喜欢战争游戏吗？

答：不是的，实际上我并不在意这些。那时我想，这就是我要设计的游戏，所以我在空闲时间进行设计，那些都是晚上或是周末的时间。与此同时我还做了大量的其他设计。1980 年的 10 月或 11 月，我升了职，不再从事游戏设计。我成为第一个硬件宣传员。我为 Atari 工作，就像 Guy Kawasaki 为 Macintosh 机所做的宣传一样，实际上，我也做得很成功。我让人们使用 Atari，因为它比苹果机要好用得多，它需要的就是好的技术销售员。为此，我在全国各地举办研讨会，发宣传单等等。同期，在程序设计员社团中引起了很大的反响。那是在我开始工作一年后的一件很轰动的事情。为此，我也受到了极大的赞誉。

总之，我在 1980 年 10 月或 11 月左右开始此项工作，为此我拿出一些演示软件来演示 Atari 的各种特性。我想，最后自己要花一些时间来学习这些大家虽然知道、但是还没有人会用好的特性。所以，我静下心来，不惜时间地来研究。终于，在不到两个星期的时间里，我成功了！做出我自己也很满意的大滚动平面。对于那时的条件来说，这个结果可称得上是有创造性的了。我记得当时我兴奋极了，把它拿到 S.S.I，那里是当时最棒的采用苹果机开发研制战争游戏的公司。还记得我把它拿给一位职员时，他的态度很不热情，说道：“这是永远也不会在战争游戏中起作用的！”而我认为那是对 Atari 的一种偏见，他们觉得 Atari 并不是真正的计算机。当时我有些不知所措，觉得他们有些心胸狭窄，于是决定自己动手。那个时候，我就用一流的方法编了这个游戏，这个游戏在现在还经常会用到。那就是用小巧的、能够表现出神奇图像的技术、并且是动态的方法来完成。那次的成果便是 Eastern Front。1981 年的 6 月份运行了这个游戏，可是该游戏的可玩性却有些糟糕。于是我又用大约两个月的时间研究该游戏的可玩性。最终在 8 月份通过 APX [Atari Program Exchange] 完成并发布了该游戏。事实证明那次是一次巨大的成功，人们把它看作是 Atari 游戏的第二部经典之作，第一部当然是 Star Raiders 了。



问：那么，实际上你是首先致力于绘图方面的研究，然后基于此进行游戏的创作，是吧？

答：那是每个设计者必须要做的努力。刚开始进行设计的时候要研究一些小的技术性的窍门，到最后如设计师一样技术成熟时，就可以不考虑这么多了。



问：你最后发布了 Eastern Front (1981) 的源码，是什么驱使你这样做的呢？

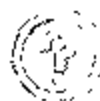
答：那确实是很不合乎惯例的举动。那时我想，电脑领域是一个高速变化的领域，而且在该领域我做得很好。到其他人开始使用我的成果时，我会有新的技术发现。我会继续探索其他的东西。我没有想独占的感觉：“那是我的劳动成果，不许其他人知道。”我的想法还是：我们都大大受益于工业的进步。我不是个无政府主义的学者，我认为人们有权利赢得自己所需要的东西。我只是觉得这样做还要有很大的限制，只有在工作量很大的条件下才能这样做。而我所做的只是小事，因此，我把它公之于众。

有趣的是，还有许多我已经公开但却没有人真正使用的技术。例如滚动技术，虽然有一些人使用，但都还是一知半解。另外，从未有人学习 Ed Logg 展示给我的那种奇妙的绘图方法，





当初我就是凭着对它的一点了解进行工作的，而且做了大量的改进和扩展。而这种渐隐技术是让人兴奋的，你会看到各种各样的变化，那么的漂亮。奇妙的编码！如果在位图上使用这种技术，你会惊讶的发现很奇妙的东西。我使用过几次那种技术，但似乎没有其他人用过。但是我想许多人一定看过 Eastern Front 的源码，也就会意识到其实游戏并不是那么难设计的。



问：你所从事的宣传工作是否占据了你开发游戏的时间呢？

答：实际上我只做了一年的软件宣传工作。之后，Alan Kay 让我参加他的研究组。事实上，我是他第一个邀请的人。三个月的时间里，Atari 研究组由 Alan Kay、他的行政助理 Wanda Royce、我和我的助手 Larry Summers 组成。那时，他们惟一能安排我们的地方只有管理人员宿舍的一个空房间。Larry 和我在那儿设计程序。同时有一个叫做 Ray Kassar 的人也住在那里，他是 Atari 的主任，但是却是个呆板、迂腐的家伙。他实在是讨厌我们住在那儿，我的意思是我们的工作也真的影响了他的休息。所以我们很快搬了家。



问：请介绍一下游戏操控情况，好吗？

答：基本思想就是：玩家处于一个 8 人构成的队伍中，目标是要受大家的欢迎。事情发生在学校舞会之前；要让大家选他（她）为舞会之王或王后的话，就要从事一些社交活动，要拜访大家。这个游戏有一个真正良好的界面。就像参加游戏节目的选手一样，游戏里，8 个人坐成两行四列，玩家坐在左上角的位置。首先，用游戏棒从其他七位选手中选择一位，然后按动按钮，这时那个人的位置会响起电话声。他会拿起电话。这时再用游戏棒选择另一个人。一旦选择完之后，你就要放下游戏棒，控制你的表情，时而一边大笑一边不断点头，时而皱着眉头不断摇头。这些表情可以表现出你是否喜欢这个人。你还会对一些人说：“我喜欢他们。”他们也会说：“我也喜欢他们。”这时，他们还会告诉你其他人说的一些话，比如这个人喜欢他，那个人喜欢他……重要的是玩家要从这些话中来决定加入哪一个社会小圈子，从而来迎合大家，受到所有人的喜爱。从某种程度上来讲，这里面也涉及到一些伎俩，例如，你要告诉每个人：“我很喜欢你！”还会说：“如果你讨厌他，我也讨厌他！”但是这样做的同时还可能冒犯别人，因此，玩家要十分小心。这就是一个有趣的小游戏的例子。



问：在电子游戏衰落时期，Atari 的思想模式是怎样的？

答：有些悲惨。事实证明，电子游戏的失败是大势所趋，也就是说，其实真正发生的是 Atari 的衰败。实际上 Atari 做了很多错误的决策，是造成它衰败的原因。Atari 的衰败使整个行业失去了信心，因此，许多并没有做错决策且健康发展的公司也都逐渐衰落。可以说这是一个行业的失败，因为它的带头公司极大的失去了信用。愚蠢的是，大家只是觉得失败临头，都在想：“大家都完了，那就完了吧。”而没有解决根本问题。

事情有了转机，但不幸的是，恢复的形式却让大家惊讶。最初的失败让大家不再信任电子游戏，但实际上电脑游戏没有受到什么影响。可糟糕的是，在那时，大多数电脑游戏来源于电子游戏，因此，许多抄袭电子游戏的电脑游戏公司也倒闭了。那段时间真的是很糟糕，我没有





工作，什么都没有。对我来说，两个新的事物摆在我面前，那就是 Balance of Power 的编著以及 Macintosh。当时我和 Amiga 的人就是否会为他们做软件宣传进行了严肃的讨论。实际上归结为两者的选择，是 Mac 还是 Amiga？我考虑了很多，因为我知道一旦选择就不能后悔。最终，我选择了 Mac，结果证明，这个选择是正确的。

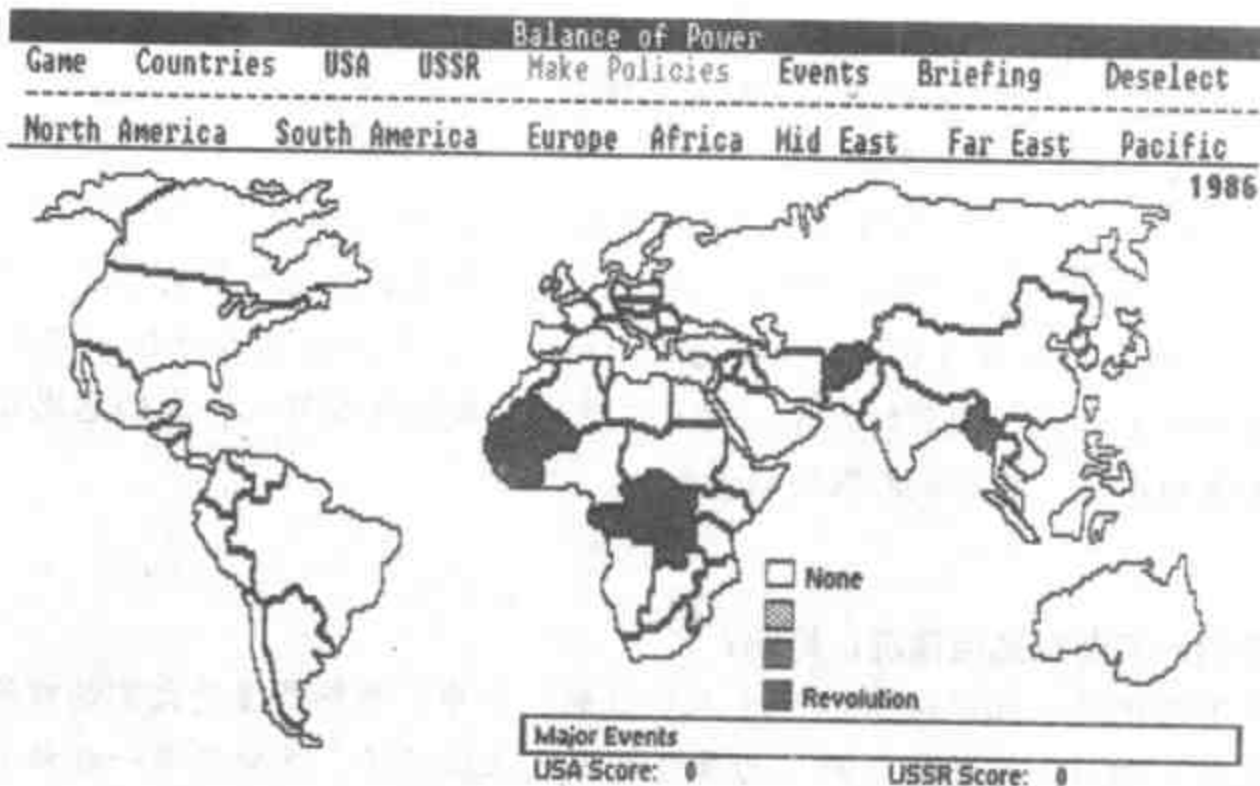


图 14.1 Balance of Power

之后，我开始着手于 Balance of Power。那时，我最大的愿望就是能够顺利地重振这一行业。你知道吗，真的有一个机会，那时在 1985 年到 1987 年那段时间，也就是对于电脑游戏行业来说的重要时期。而实际上 1990 年这一行业便开始转运。如果这样说的话，我认为那应该是 Chris Roberts 的 Wing Commander 改变了这一行的状况。我们真的有一个机会，看来我们就要胜利了。



问：你怎么认为 Wing Commander 改变了这一行业的命运？

答：在 1985 年，这一行业的大问题是谁来销售。似乎没有人确切地知道，但却有一些可以考虑的东西。Balance of Power 的巨大成功让人们觉得：或许技术精湛的游戏，但不是电子游戏，会有一个好的未来。这一思想让人很兴奋。过去很成功的其他游戏组成了 Infocom 游戏，这些都有益于改变原来的不良状况。



问：是因为这些完全不同于电子游戏吗？

答：是的。你把两者放在一起，就会发现它们都有同一个目的。因此，为了达到目的就要进行大量努力。这一行业的状况仍然不是很好，因为设计电子游戏是很容易的，而且也可以把它们卖给那些受电子游戏衰败影响不大的人们。我们确实为此而犹豫，怎么办？是继续电子游戏的发行，还是进行更广泛意义上的游戏研究？Wing Commander 使得选择明朗化。这里有两个原因。一个原因是：Wing Commander 所做的决定行业命运的主要行为是它带来了市场份额。也就是说，是因为 Wing Commander 昂贵的价格。有趣的是，Chris Robert 否认它的高成本，是因



为一些机动的内部账目的原因。我们回到当时，大约1990年的时候，一个游戏的投资一般是10万美元到20万美元，还有一些更便宜，但是一个30万美元的游戏就是相当贵的游戏了。而Wing Commander的价格大概是100万美元。就那个时代而言，可以说是荒谬的。实际上，一个内部知情的人告诉我说，Wing Commander本身从未能偿付它的投资款，而是由后来设计的软件和附加软件收回了投资。但是他们实际做的就是花费那么多的钱，以至于只有让产品成为顶尖产品才可以有利可图。而他们确实达到了目的。问题是，他们已经提高了整个行业的门槛，我们不得不生产100万美元的游戏。不幸的是，这些产品只有成为一号产品才会有利可图。但是却只能有一个一号产品。因此，这就使得这一行业更加保守。我们花费如此之大，决不能让产品成为第20号产品。没有排在前十位的任何产品都会有损失。因此，这一行业很快成为追求顶尖产品的行业。其实，80年代后期已经有这样的萌芽，但是Wing Commander改变了它。因此一旦成为尖端行业，那么，整个市场战略、经济波动状况等都会发生很大变化。另一个原因是：Wing Commander似乎要把动作游戏作为未来研究方向重新开发。基本上说来，那也是这个行业的强点所在。



问：在那次行业衰败之后，你写了《The Art of Computer Game Design》，是吗？

答：是的，实际上我一参加Atari研究组就开始写了。有趣的是，在那个组里，我的一个目标就是真正改造整个游戏设计领域。因此，从本质出发，我想在Atari的员工里建立一个电脑游戏开发者的讨论会。我准备在每个星期五下午开一次研讨会，结果又不是很顺利。我给所有Atari的设计者发出邀请；其中有一个“自助洗衣店”软件开发小组的负责人，他是一个固执愚蠢的家伙，因为没有事先得到他的同意，所以惹怒了他。他认为我没有按照级别关系办事，因此，禁止他的员工参加我的会议。那也是Atari失败的原因之一。所以，研讨会没有办成。因此，我决定，我要把这些想法写下来，于是开始写这本书。1982年，我完成了该书，但是CEO——Ray Kassar也是一个固执的家伙，他一定要在此书出版之前亲自审阅原稿。我只好把书稿交给他，结果原稿在他那里延误了一年的时间。



问：你现在还经常回顾那本书的内容吗？

答：当然，我养成了习惯。如果那时我知道多年以后的今天，人们还在读那本书，并还会从中受益的话，我会很吃惊的，而且不敢相信这是真的。现在，我还会收到提及那本书的e-mail。毫无疑问，人们还在受益于那本书。而且那本书也说到整个游戏行业 and 游戏界的一些不良方面，以及这一领域中新思想的匮乏，都是些令人遗憾的事。



问：实际上，到现在为止还没有另一本关于这方面的书，是吗？

答：是的，其他同类书籍写的也只是关于程序方面的知识。很遗憾，没有人来写超越那本书的东西。





问：自从出版了那本书，你一直都很关注游戏设计方面的动态。为此我很好奇你为什么会有这样？

答：有两个独立的原因。首先，通过写作使我的想法更加尖锐，这方面，我已做了大量的工作。其次，也为了与其他人交流思想。当然，有些工作有重叠性。大部分时间我自己随便写些东西。我保存了很多很多关于特殊设计的小文章，文章里主要都是我的关于设计方面的个人观点。然而，有时我更愿意只为大众写文章，通过网页或其他途径给大家看，这样做的目的和写书时完全不同。当然，我经常是带着两种目的来进行写作的。

问：那么，你决定组织电脑游戏开发者研讨会的想法，是源于你游戏设计的写作吗？

答：在我出版《Journal of Computer Game Design》这本书时，已经有了这个想法。结果书的发行很成功，很快就有了很多订户。外面有好多人在等着订阅，见到那种场面，我意识到可以组织一个研讨会。因此，我决定先在家里组织一个微型研讨会。于是，定货会上我宣布：“我要开一个研讨会，准备参加的人请与我联系。”最后有26个人参加了会议，一整天的时间里我们都坐在楼上的大房间里，谈论着游戏设计。那实在是一次令人兴奋的经历！大家都认为这个聚会真是太棒了，应该再举办一次。他们都对我说：“Chris，再组织一次吧！”我同意了。我想了一会儿，然后决定招收新成员来帮我扩大会议的规模。我觉得让他们能真正加入的惟一方式，就是让他们有主人翁的意识。如果我把他们当做助手的话，就不会有太好的效果。因此，我决定建立一个有董事会的团体。我请了五个人作为董事。虽然我用自己的资金来运作这个组织，但是为了培养他们的主人翁意识，我还是放弃了完全持股权。这样，大家都有一样的份额。这个团体的建立是一个很大的成功，而且这个队伍每年都在不断地壮大。

问：那时你就预知：研讨会一定会发展壮大，是吗？

答：不是的。从某种程度上来讲，这似乎违背了我最初的想法。在董事会内部，我们曾经讨论过：这个组织应该办成像E3一样的节目，还是应该办成像AAAI一样的学术会议？我的想法是：这个组织的核心应该是让游戏开发者们交流思想。我们也可以办节目，但这只能是附属活动，只适合在角落里进行。这个组织是要让人们交流思想，为此，我提出了“圆桌会议”的想法。不幸的是，现在它真的成了一个节目，而会议则成了次要的活动。

问：从Atari出来，你成了一名独立的游戏开发者，我想知道，你为什么不再继续到另一家大公司工作呢？

答：最开始，的确有些压力。但是我一从Atari出来，就开始写《Balance of Power》这本书，这项工作本身也具有独立性。后来，也许是习惯了，就一直不再觉得有压力了。

问：你更喜欢独立的工作，是吗？

答：是的，我是一个非常喜欢单独工作的人。我很在意自己工作的效率。当你和其他人一





起工作的时候，你会花很多时间与他们沟通，而实际上没有做任何事情。我有很多问题要解决，但是要进一步探究问题的时候，就必须独自进行。

① 问：最初是什么驱使你写《Balance of Power》的？

答：由于我有部分的越南血缘，所以我对战争游戏有很大的兴趣。许多人反对战争，而我却想更多的了解战争，目的是最终为此做一些事情。我觉得在树丛中做防御工作是很特别的，那种防御是暂时性的，效果也十分有限。我不断地问诸如战争是如何开始的这样的问题。从70年代早期到80年代早期，我一直学习战争知识，尽自己所能学习军事历史。到了1984年，我终于觉得我可以用这些知识设计一个游戏。我还能用 Bob Dylan 的歌“Blowing in the Wind”来增强游戏的感染力。有这样一句话：“永远禁用炸弹前，还会有多少颗炸弹爆炸？”这句话使我在很多困难的情况下，能够坚持设计战争游戏。而正是许多艰苦卓绝的努力，才使得问题得以解决，设计工作才得以进行。

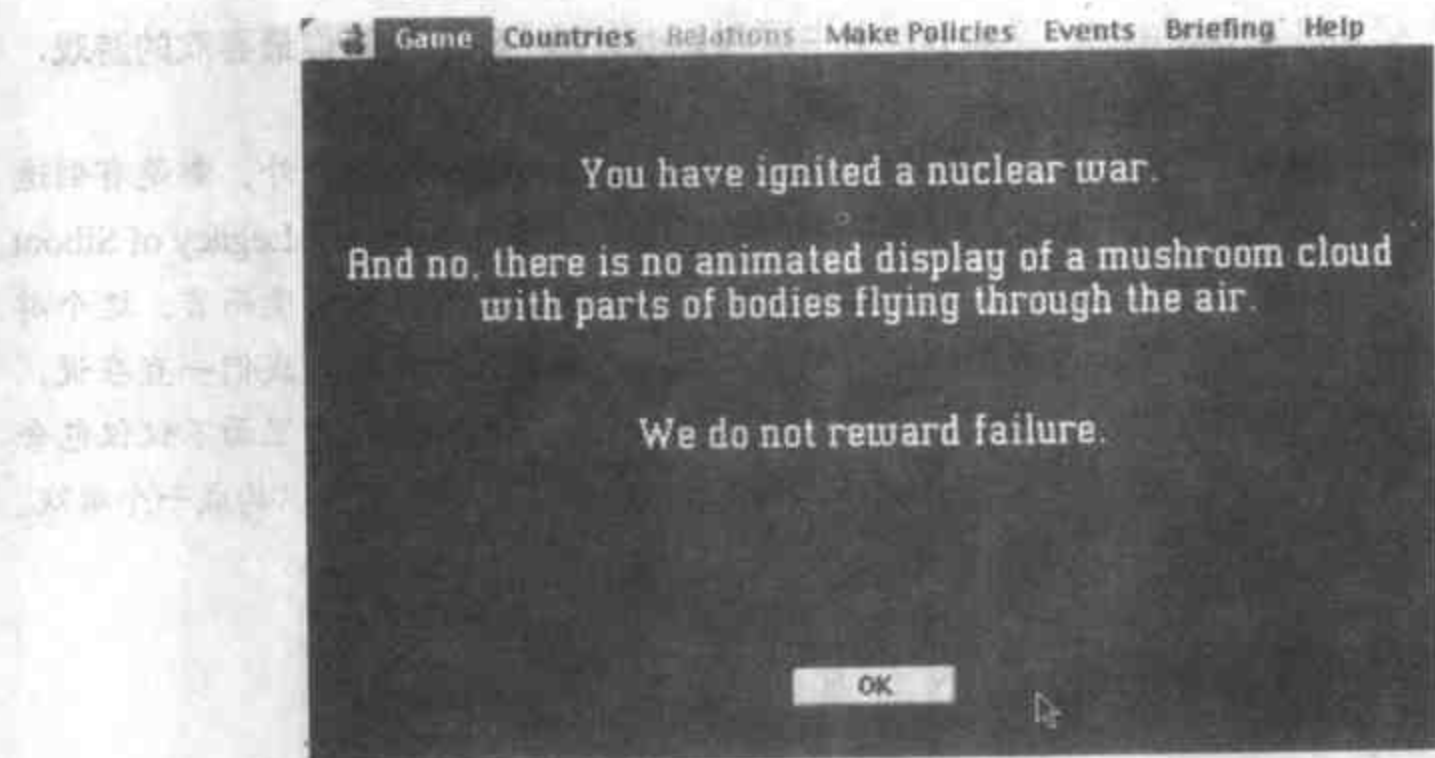


图 14.2 Balance of Power II: The 1990 Edition

① 问：你觉得这种构思值得吗？

答：是的。我真的想设计一个非战争游戏，因为已经有很多的战争游戏了。

① 问：写《Balance of Power》一书的时候，是你在战争游戏研究过程的顶峰时期，是吗？

答：是的，那时是战争游戏的顶峰时期。不知道你记不记得，只要战争游戏一出现，屏幕上就会出现字幕：“败者为寇。”那句话让很多人感到惊讶。

① 问：你是否有时会担心有些游戏不是很成功？

答：我并不奢望一个游戏能获得多大的成功，但是我觉得努力去做是很重要的。这个过程





与创作 Eastern Front 的过程非常相似，当初我只是自己随便写写，你瞧，结果受到大家的欢迎！多好呀！《Balance of Power》同样如此。但是同时，我也写了其他一些没成功的游戏。



问：在《Balance of Power》一书中，你是如何权衡现实主义与游戏之间关系的？

答：人们总是把现实主义和游戏放在一起谈论，似乎觉得两者是互相对立的，一定要从中做出一个选择。而我却没有把这个问题仅仅看作是一个困难的选择。比方说，一位画家在画素描的时候，为了表现对象的特征，会特意把一些表面特征突出出来。可是人们不会把这看作是现实与表现的冲突，而把它当作是艺术。同样的道理，我觉得有必要把设计游戏、编辑书籍和艺术联系在一起考虑。当然，在战争游戏里，我确实把世界抽象为一个危险的地方，把许多复杂的東西简化，再把这些融入游戏中，这样，游戏就称得上是一个纯粹的地理政治学方面的竞争了。这就是能够清晰地表现双方利害冲突的游戏。



问：我读过《Trust & Betrayal》这本书，书中提到 Legacy of Siboot 是你最喜欢的游戏，为什么呢？

答：在我所写的游戏中，除了在开发商催促下写的《Balance of Power》以外，都是有创造性的。也就是说，除了《Balance of Power》，作品中都蕴涵了新的设计思想。而 Legacy of Siboot 这个游戏在设计深度上和其他作品是不同的。也就是说，就超越传统游戏的程度而言，这个游戏是最优秀、最先进的。在 Siboot 里我提出了人们至今还没有接触到的思想。我们一直在说，在某种程度上 Gossip 是很优秀的，而 Siboot 还是远远超过了它。另外，Siboot 里面不仅仅包含了一个设计思想，至少有三个大的设计思想蕴涵在里面，而且每个思想本身就可以构成一个游戏。

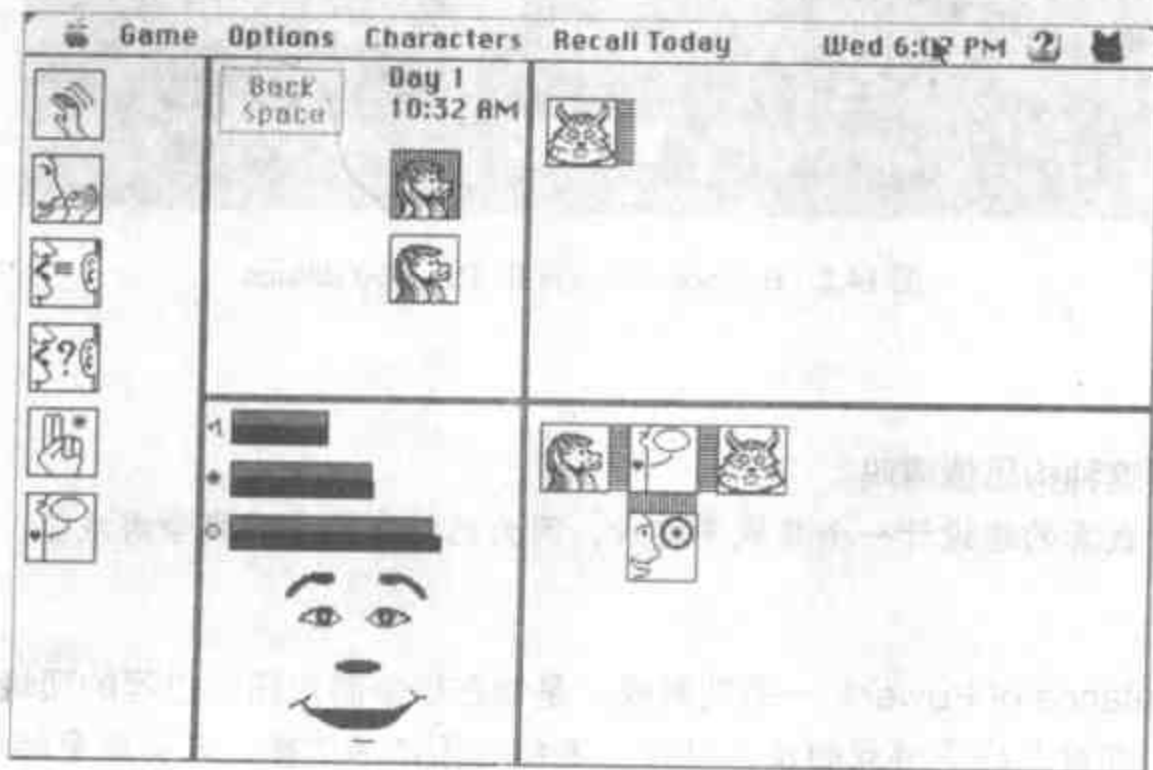


图 14.3 Trust & Betrayal: The Legacy of Siboot

另外，该游戏里面还有一些小的思想，下面就是一个例子。现在有一个用户界面概念叫做“提示工具”。如果你把光标放在界面上不动，几秒钟后，就会跳出一些描述性文字。我最早使用了这种方法，并提出一些类似的想法——玩家可以通过按按钮来进一步查看功能说明。四





年后，电脑界把提示工具作为用户界面项来使用。在 Siboot 里，这并不是什么重要的设计思想，但是，在那时却没有人使用它。

问：那么设计思想中的三个大的创新是什么？

答：第一，是语言的使用。人们互相谈话，和书本上的语法完全不同，因为我认为那不是纯粹的基于语言学的交流。第二，使用逆向语法分析。在 1979 年，语法分析的核心概念已经由 Texas Instruments 申请为专利，那时我还不知道这件事。但是，我所运用的完全不同于此，因此，根本涉及不到侵权的问题。TI 的方法是菜单驱动的，而我的方法则和它在功能上类似，但具体内容不同。这是三个创新中主要的一个，但遗憾的是，至今还没有人继续研究它。第三个创新是不及物战争的使用，从那时起有一些游戏已经使用了这种方法。这种不及物战争的概念有些纸上谈兵的思想，它有巨大的发展潜力，人们可以基于这种方法设计游戏。对这种思想的探索有着广阔的空间，而且做起来也比较容易，但遗憾的是，没有人以这种非传统的方式来思考。

问：这种非传统方式是否有些不利于 Siboot 的普及流行？

答：那不一定。实际上，只有 Mac 销售 Siboot，而从未卖过 PC 译本。当时我们大约卖了 4000 本，对于那时的水平来说，有些不尽如人意。但如果是 PC 译本的话，销售量会是原来的十倍。所以，我们一直说，如果有 PC 的话，销售量可以达到 2 万册到 4 万册，但是开发商却不同意这样做。

问：那么，对于 Gossip 来说，在游戏里加入人物是你的目的吗？

答：是的。在 Gossip 里，我进一步采用了“人物而非事物”的概念。另一个创新是采用了一些不相关的小故事，这些小故事不时地在整个游戏中出现。

问：在开发商的要求下你才坚持了 Balance of Power II 的创作，是吗？

答：是的。我写完 Siboot，他们出版了该书，但是很明显，这本书不会给他们赚多少钱。于是开发商们很失望，一直要求出一本续集。这次，他们给我施加了压力，而我也觉得似乎欠了他们什么，因此，我写了《Balance of Power》的续集。

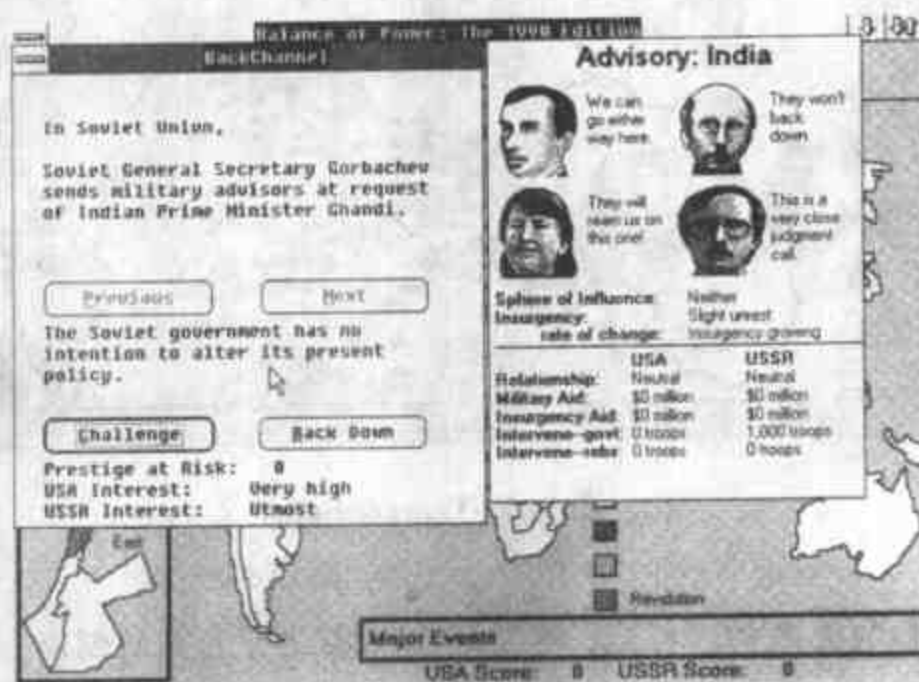


图 14.4 1990 年版的 Balance of Power II





问：那么，你没想改进一下《Balance of Power II》的原稿吗？

答：是的，我没有。实际上，我觉得尽管《Balance of Power》的续集比较有趣，但是，它只是 B.o.P.I 的改编。就当时电脑设计行业的水平来说，这本书是新作品，而且可以称为“第二版本”。但就我自身的水平来说，那本书的工作只不过是整理工作，再加上一些不起多大作用的零零碎碎的东西。



问：对于 Guns & Butter 的思路你是如何得来的？

答：不约而同地，世界三大优秀电脑游戏设计师几乎同时有了相同的想法。我们都说：“我打算设计一个关于征服世界的游戏——Empire 游戏。”这三位设计家分别是我、Sid Meier 和 Dan Bunten。有趣的是，我们三个人努力的方向却完全不同。大家都知道 Sid 的方法，而且大获成功。我们三个人曾经一起讨论过游戏的设计，那时 Sid 有一个明确的观点，设计中一定要带有幽默性。他还说：“只要游戏有趣，我决不在乎是否符合实际情况。”

我的想法是让游戏有教育意义，而 Dan 的想法着重于游戏的社会性。Dan 做的一个很好的小游戏叫做 Global Conquest，你可以和其他人一起玩这个游戏。我认为这个游戏优秀得可以称得上是一个未被挖掘的宝藏。然而，他和 Electronic Arts 之间似乎有没完没了的问题，至今我还是不明白为什么他那么忠于 Electronic Arts，因为他们不断要求他在游戏里，特别是在 M.U.L.E 游戏里加入枪战镜头。而我认为，就他使用的平台以及那时的设计技术来说，这个游戏是他所

做的最优秀的游戏。它充满了幽默性，但不幸的是，这个游戏却从未能接入现代机器。而其中的原因就是 Electronic Arts 坚持：玩家要互相射击。但是 Dan 拒绝了，他说：“那不可能！”Global Conquest 也遭到了同样的下场。

我做的征服世界游戏是 Guns & Butter，它更多的是关于宏观经济学的东西。实际上，开发期间，我把它叫做 Macro-Economic Conquest。我觉得它成功地用游戏的形式告诉人们历史

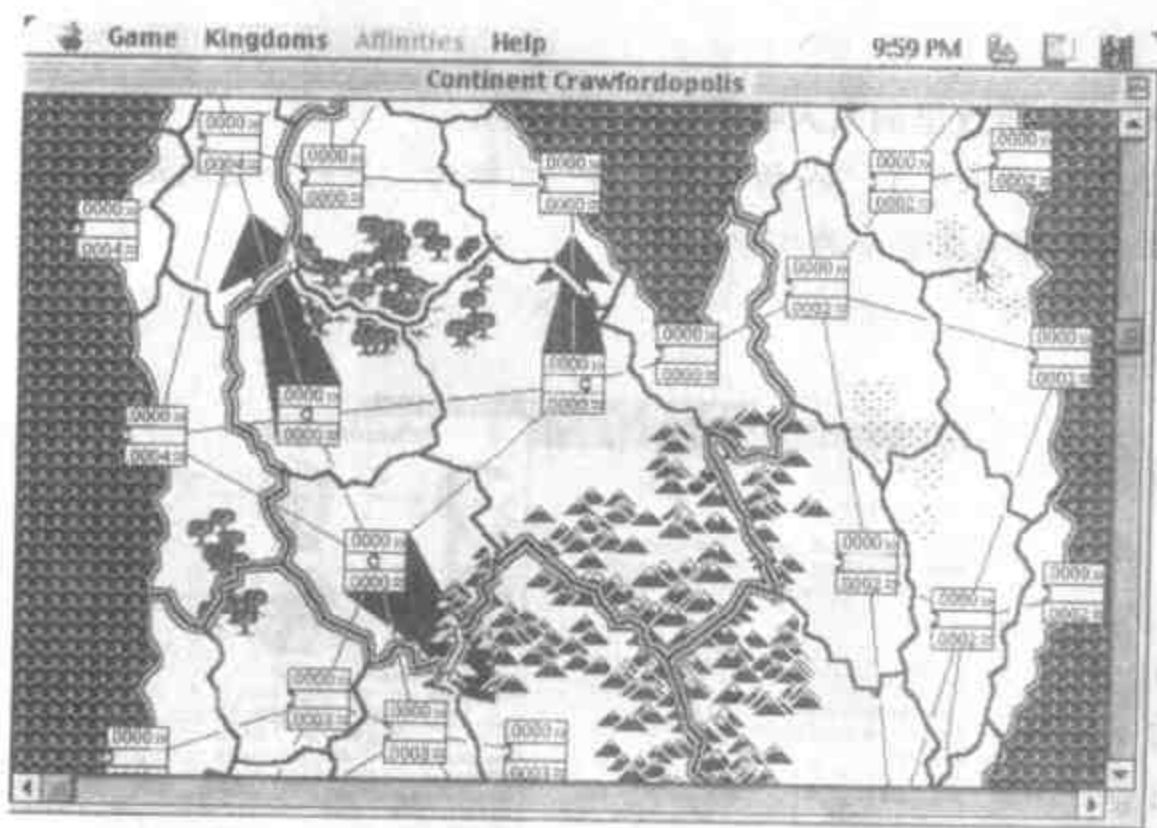


图 14.5 Guns & Butter

是如何发展的。当然，这个游戏确实没有太多的创新，只有一些小的设计思想。比如说，Guns & Butter 有产生一些陆地的小算法，还有能给地区省份命名的小算法，为此我感到很骄傲。但是却没有创新的东西，仅此而已。



问: Guns & Butter 包含了一些关于平衡综合系统的有意思的想法, 但你却认为这没有什么大的意义, 是吗?

答: 是的, 主要因为我当时没能处理好。我常想: 如果再给我半年的时间, 这个游戏会运作得很好, 但是当时我却做得太匆忙了。

问: Balance of the Planet 似乎是一部极富教育性的游戏, 那是你的初衷吗?

答: 绝对是的, 因为我没有要做一个幽默游戏的打算。我只是想: 已经有那么多诸如枪战内容的游戏, 我没有必要再做了, 应该从另一个方面编个游戏, 于是我选择了人道主义这方面。我要进行纯粹的模拟, 进行具有现实主义和教育意义的模拟。那时大家都知道, 1990 年的地球日快到了, 所以应该在地球日到来之际发布这个游戏。我觉得那会是我的一大贡献。因此, Balance of Power 是关于越南战争的游戏, 而 Balance of the Planet 是关于地球日的游戏。



图 14.6 Balance of the Planet

问: 在 Balance of the Planet 发行后不长时间, Will Wright 又发行了 SimEarth。但是比较两者, 可以发现, 他的作品只能称得上是一种玩具, 而你的作品是一种有目标性的游戏。

答: 其实, SimEarth 并不是 Will 作品中比较好的一个。虽然他写得已经很不错了, 但我觉得这个作品的目的不是很清楚。它只告诉人们: “这里有一个星球, 有一些地理方面的处理方式, 还有这些生命体等等。”而给人的感觉是没有设计焦点。要知道, 那种设计方法绝不会收到好的效果, 而且对他自己也没有多大帮助。当然, 因为这个作品的画面比较好看, 还有很多小的特色, 所以它要比 Balance of the Planet 成功, 但是它却不像 Balance of the Planet 那样有教育意义。





问：我认为 SimEarth 里包含了很多虽然有趣但却让人难以理解的方法。

答：虽然他已经简化了很多东西，但是由于仅仅是为了人们易于接受它，而还是没有目的性，所以这些方法还是没起多大作用。即使表达这些方法如何运作的模型易于理解，但是对我来说还是没有什么用处。



问：我听说有人批评 Balance of the Planet 缺乏乐趣，你认为趣味性是游戏设计的必要条件吗？

答：问题问得好。的确，很多人都把幽默性当做是游戏设计的必要条件，而正因为这样，游戏设计行业已经走进了它的误区。现在，电子游戏和电脑游戏已经差不多是同一个概念。由于枪战镜头的画面、许多图形、图片以及对幽默性的强调，使得它们已经没有什么不同，都是可视化游戏。但是我不明白为什么电脑游戏一定要限制在这个模式中，就好像有人说：“我看了 Das Boot 这部电影，但它的内容一点都不幽默，因此这不是

部好片子！”毕竟这部电影不是幽默片。再比如说，我们都知道 Saving Private Ryan 不是一部幽默题材的电影，但它却是一部不可多得的好片子。同样，电影 Schindler's List 也是如此。当然，也有许多幽默电影，如 Star Wars。可见，Hollywood 从不像电脑游戏设计行业那样限制它的作品。整个 80 年代和 90 年代早期，为了让游戏产业成为一个更具广泛意义的而不是幼稚逗乐的行业，我不断地为此努力，但是最终还是失败了。到现在，这一行业还没有成为一个让人满意的行业，而且也许永远不会。人们都在钻牛角尖，一味地追求逗乐，多么可笑！看到人们做诸如的滑稽游戏，就好像看到一些吹泡泡糖的孩子走进了漫画书店。虽然慢慢地也有一些人意识到这一点，但我还是不敢确定什么时候人们才能走出那个死胡同。

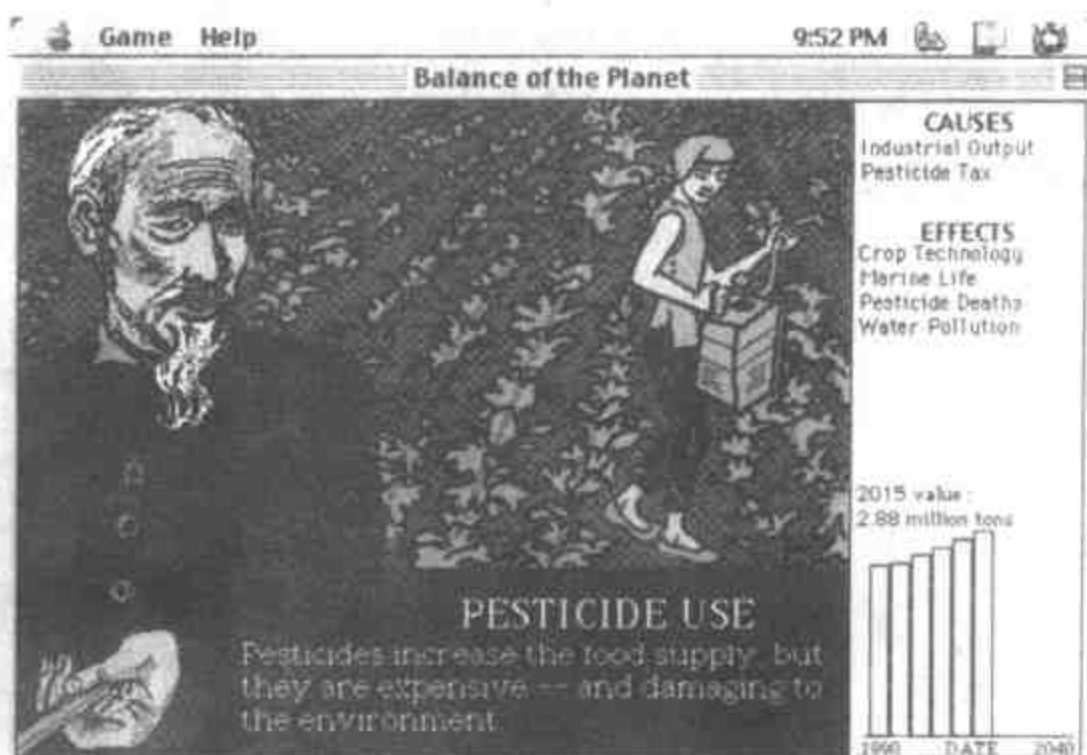


图 14.7 Balance of the Planet



问：你认为游戏行业的发展已经有些停滞不前了，是吗？

答：是的，只是人们还没有解决问题的意愿，甚至还没意识到那个问题。



问：我想那一定是你离开游戏设计行业而转入研究 Erasmatron 的原因吧？

答：实际上有两个原因。自从我上了“Dragon Speech”一课，我就离开了原来的工作。那





次课……我给你讲讲那次课最后的情形吧！课上，我一直用“Dragon”（龙）来比喻艺术。就在要上完课的时候，我不再和下面的学生说话，而是模仿直接和一条龙对话。我说：“我全心全意地寻找龙的时候，突然，它出现了，我能清楚地看到它。”于是我开始和龙之间激烈的对话。记不清当时都说了些什么，只记得我说：“你力大无比，你美丽漂亮，可是你又那么可怕！啊！你吓死我了！”然后我又尖叫道：“你别伤害我，我觉得你的利爪正在撕裂我！”那时我已经不像是在讲课，我十分激动地喊出那句话：“别伤害我！”而且这把大家都吓坏了。那些学生习惯了传统的上课方式，不习惯课上有这么大的情绪波动。之后，我说：“我还是经验不足，以至于不能很好地给你们上课，要面对你们的话，我还要继续积累经验。”说完，我拔出剑大喊到：“为了真理，为了完美，为了艺术，冲啊！”于是，我穿过讲课大厅，从走廊跑了出去，再也没有回来。



问：这是发生在计算机游戏开发商的会上吗？

答：是的。许多人这样想：“Chris 已经发表了最后的演说，就不会再回来了。”但是事实上，我第二年就回来了，并且我有各种打算去继续发挥我的专长。我本想改变我的方向而去做些别的，但是，员工需要我，所以我还要继续留在这儿。不幸的是，涉及了董事会成员的整个难对付的事件使这一切成为了泡影。这是因为不仅各位董事会成员的行为，而且整个团体的态度都是那么伤人。他们说：“嗨，这就是硅谷，你不是正在为达到你的目的而努力吗！如果他们是在打棒球的话，如何得到大生意？”我的反应是：“我才不想加入你们那个肮脏的团体呢。”那是一段如此痛苦的经历，使我不得不离开硅谷而搬到 Oregon。更滑稽可笑的是，现在每当我经过那儿时，我都会贪婪地看着硅谷。而且这件事的确一直在烦恼我。



问：这就是让你开始 Erasmatron 工作的原因吗？

答：我曾经在那个方面有过发展。但是使我离开它的原因有两个：第一，这个行业的发展方向是我不喜欢的；第二，这个行业在道德和社会方向上的发展也是我不喜欢的。



问：那么 Erasmatron 的计划是如何产生的？

答：我要研究一个交互式故事介绍系统。那时候我说：“现在我要回去，回去做我的 King Arthur 游戏。”因为我曾经在 Atari 做过这个让我骄傲的 King Arthur 游戏，并有许多好的主意，但是我觉得这么说并不是很公平，所以我总觉得我对此有一些亏欠。于是，我重新开始全力以赴去努力尝试一种全新方法，这使我与故事介绍更为接近。然而，每一件事都是需要编码处理的，这是非常困难的。我们到所有的大公司去游说，想让他们对我们的事感兴趣，可是没有人对它有兴趣。

大概就在那个时候，我偶然遇到了一个叫 Edith Bjornson 的女士，她掌管着 Markle 基金。她建议我在其他不同的方面采用这个技术，作为一种可行的技术让那些非技术人员能够创造属于他们自己的故事世界。我非常喜欢这个想法，因此 Markle 提供给我基金资助，并且要把这个项目的战略在标语中表达成“释放如潮水般的创造性”。因而，我为此开发了三个软件。





Erasmatron, 这是个引擎的编辑软件; 引擎, 实际上它是用来进行启动的; 最后是前端接口, 这是用来把它传递给用户的。这是一个巨大的工程, 我整整做了两年。但是不幸的是, 问题比我想像中的还要复杂。好在两年后工作渐渐进行得很顺利, 事实上, 它在技术上已经是很充分的, 但是我认为它在商业上并不是很充分。



问: 你的确切含义是什么?

答: 创造一个充满娱乐性的最终结果是需要花费非常多的努力的。Laura Mixon 花了近 18 个月的时间来做 Shattertown Sky。但是 Shattertown 并没有收到预期的效果。它的娱乐性并不是很强, 甚至并没有完成, 因为有一些地方会停下来。然而, 她比大家所想像的工作时间更长, 更努力。在她工作的部分并没有出现失误, 而我这一部分工作的失误在于我过低的估计了工作量, 我认为一年的时间就足够了。因此, 首先, 她在至少 6 个月的时间内没有得到完全操作软件; 其次, 她所使用的工具功能不全, 以至于她做了许多烦琐的工作。而结论是 Erasmatron 需要加大投入和马力, 还有, 要完善一些引擎的修饰, 但是这些的确是相对次要的。从那开始, 我在 Erasmatron 2 上所做的大部分工作都使故事世界创造的整个过程变得更容易。



问: 你有没有得出结论: 做一个故事世界是一项很艰巨的任务? 你有没有找到使其变得更容易些的方法?

答: 其实用 Erasmatron 2 来创作故事世界要比用 Erasmatron 1 容易很多, 这对于我来说是毫无疑问的。Erasmatron 2 整理了创造一个故事世界的过程, 缩短了大约一半的时间。你知道, 做 Erasmatron 1 时, 我们仿佛是在黑暗中射击, 对于创造的过程会是什么样也说不太清楚。人们说 Erasmatron 1 是个糟糕的设计, 对此我也有同感。事实上它要比最初的设计计划好得多。虽然我已经做了许多的改进, 但这些仍然不够。我认为运用 Erasmatron 2 人们可以用时间上充分的投入来创造一个出色的故事世界, 而这个时间我想大概要 6 个月或是一年, 但我还没有具体的证实。这就是阻碍我整个工程的原因: 我需要证据。



问: 那是不是你希望通过 Le Morte D'Arthur 方案提供的东西?

答: 我不知道。我曾经遇见过许多的作家, 他们都阻碍这个项目, 我不明白为什么。我认为其中的一个因素就是感觉上的士气消沉。我曾经在这个项目上花了 9 年的时间, 然而最终还是失败了, 因为除了 Markle 基金以外, 没有人对这个感兴趣。而且还经常会有一些大的泡沫。但是, 现在有三个独立的公司表示对此感兴趣, 因此, 这说明我们没有到绝望的地步。可是, 虽然有些计划还一直在进行, 并且仍然有希望进行, 但是, 却没有任何进展, 因此, 我已经变得有些失去信心了。



问: 那么, 理想的 Erasmatron 故事介绍系统应该是什么样的?

答: 我会从战术上和战略上两个方面来阐述。从战术上来讲, 就是玩家具体的每时每刻的经历; 从战略上来讲, 是玩家宏观的整个经历。战术上, 观众会在屏幕上看到一个静态的表示





刚刚发生事情的画面。屏幕会显示刚才做事的人和他旁边人的面孔，还会有一些解释性的语句。另外，我还要用一种类似漫画技术的方式。也就是说，漫画技术也会在画面背景之间很好地表现行动。这样也就需要两种画面背景。我打算使用漫画中蕴涵的那种艺术风格。在 McCloud 的《Understanding Comics》一书中，他就提到了那种代表抽象数量的三角的概念。



问：一边是笑脸，另一边是真实图片，是这样吗？

答：是这样的。我的想法是：我们应该沿着那个三角形移动，远离放置真实图片的角落，越接近抽象和画相越好。因此我觉得，我们正在谈的是更为抽象的放置类型。然后，是用完整句子表达的选择菜单，允许玩家说的话和做的事情就在里面。从战略上来讲，最大的不同是所有的故事世界本身都有其曲折的特性，而“Barroom Brawl”却不是这样，因为它本身是单一场景的。即使“Corporate Meeting”的内容有一点曲折，但它也是单一场景的，并且我们已经明白如何处理这些问题。本来我认为使用一些地图点就可以了，但是现在我和 Laura 都想到了一个方法。我不想把它作为一个新发现来具体描述，它是我们慢慢酝酿了几年的概念。我们正不断给它注入新鲜的东西，相信它会进展得很好。

这个想法就是：故事里存在一些东西，它们是超过玩家所控制的核心计划的。然而，玩家控制着大量的交互作用，这种作用不会影响但却决定故事的最终结果。比如说谋杀之谜的故事：Shattertown。故事里时间流逝，从各种可能来说，结果是要么家族的人打算宣战，要么 Sky 揭露事实的真相，要么谋杀者抓住 Sky。故事的结尾是编好的，但是，玩家控制的交互作用会影响结果。也就是说，结果如何取决于玩家所做的事情。Le Morte D'Arthur 也是同样的道理，它已经清楚地讲到：无论发生什么事，故事结尾时 Mordred 都会叛变，这时其他人就要选择加入哪一方，一些人会加入 Mordred，另一些人会加入你这一方。因此会有一场大战，势力大的一边才会赢。而让人们加入 Mordred 还是加入你这一方就取决于玩家所做的事情。

对于 Le Morte D'Arthur，我已经有了另外一个想法。我正努力向其中融入一些现有的 Le Morte D'Arthur 的元素。在这个游戏里，你不再是 Arthur，而是一个来自于未来的移民，叫做 Merlin，任务是建设现代化的亚瑟世界，阻止黑暗世界的到来。你努力改造这个社会，让它更高效地运行，教会人们注意公共设施的建设，意识到教育的重要等等。在这一过程中，所有的贵族都憎恨你，他们不断地反对你，用尽各种阴谋陷害你，甚至要杀了你。结果 Mordred 叛变了，这个结果虽然不是那么曲折，但让人感觉更有意义。



问：这么说，玩家不仅仅是以目前版本的内容引导的，而且还是以一种方向引导的。

答：我们并没有让玩家做出什么社会革新，而是通过给玩家各种机会来让他发起这种革新。例如：我们会告诉玩家：“好吧，你认为该提醒人们注意公共设施卫生了吗？或者，你觉得该教他们如何使用马蹬了吗？”而每个问题都需要时间。而且就在你帮助这个社会解决问题的时候，还会有其他情节发生。故事里还会有很多交互情节发生，而我们要进一步研究的是“半情节”或“虚拟情节”或一种“框架情节”的概念，它们能够按照所期望的故事情节继续进行。故事里玩家仍然会经历一些事情，但是这不会像传统故事那样影响整个故事，也不会决定故事的发展。





问：这么说，不象读故事那样，玩家会更多的参与到故事中去，但是这种参与并不是完全自由的。

答：是的。也就是说，你要受一种戏剧化的约束，但不是人为的约束。这就是戏剧，它需要在某些规则下进行。运用这些规则不应引起玩家的不满，否则玩家会在和 Arthur 说话的时候挖鼻孔，而这并不是合理的戏剧化。有些人辩解说，如果你不给玩家可以充分发挥的自由，那么就不会有什么效果。但是我却完全不同意这种说法。我认为，只要给玩家所有合理的或者大部分合理的自由选择，还是会收到好的效果的。

问：为什么你没有把你的 Erasmtron 系统称为某某游戏呢？

答：原因有两个。第一层原因是市场。现在一提起计算机游戏，人们就会想到 Quake、Command & Conquer 以及其他一些类似的游戏。而且人们对那个名词的联想都是关于充满枪战的镜头以及智谋管理，也就是说，这些联想已经深深印在大众的脑海里了。这样，如果我也把 Erasmtron 叫做游戏的话，那些联想就会误导人们的想法。而且，如果你查字典的话，会发现“游戏”这一词比大多数词的解释要多。我曾拿它和“做”、“吃”、“有”这样的词做比较，结果发现“游戏”一词的解释确实比较多。因为它的语义广泛，所以很多场合都可以使用，而且都是完全不同的意义。但是，很快又发生一些变化，虽然你可以把“游戏”一词用于整个产品和活动领域，但是一旦人们联想到计算机，就会说“电脑游戏！”这样，所有其他语义都为此而不被使用了。或许，我应该把 Erasmtron 做在网页上，叫它网页游戏；或者如果在 Mac 上做，我就叫它 iGame，但我就是不敢把它叫做电脑游戏或电子游戏。

问：你为什么认为面部表情对故事讲述如此重要呢？

答：面部表情是人与人之间进行交流的一个基本形式。滑稽的是，我认为是交流的东西其他的人却认为是图形。在用户与计算机之间进行的主要是一种交流。我十分渴望优化这种交流。也就是说，我要把计算机所显示的内容设计成最能为人脑所接受的形式。并且我们所擅长的是对面部的识别和对语言的理解。因此，这也是计算机应该侧重的两个方面。而电脑游戏两者都不具备，这也很使我胆寒。就现在而言，面部表情和语言的

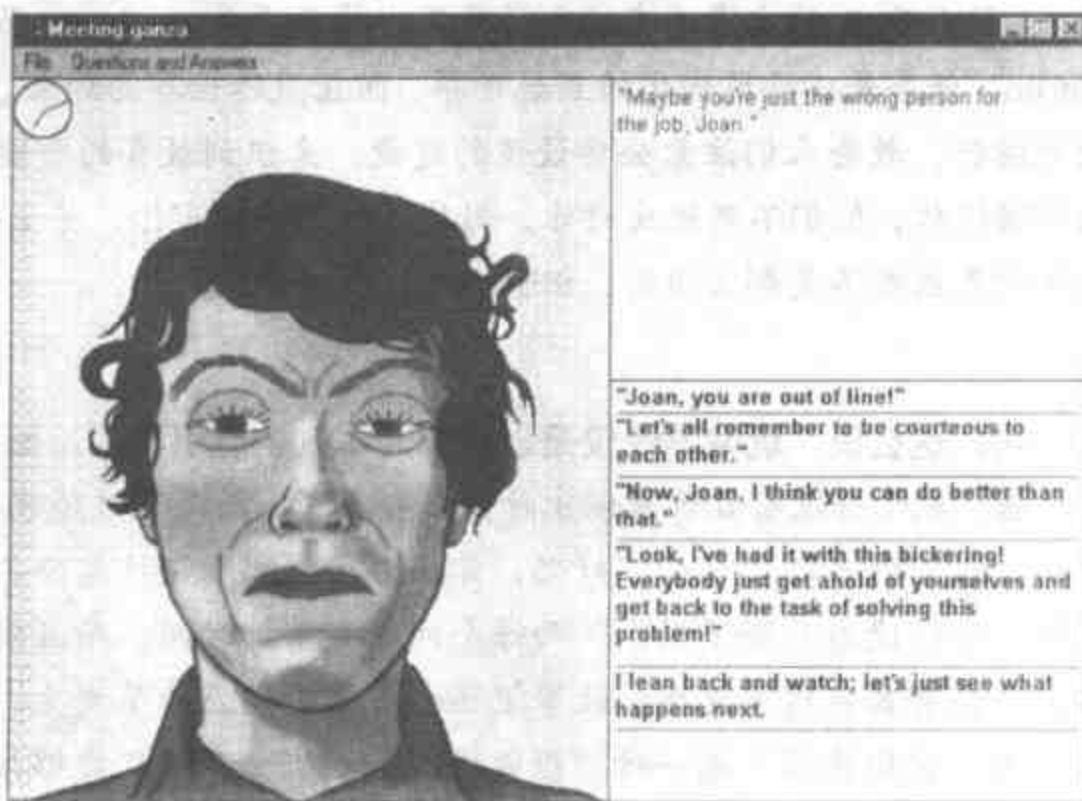


图 14.8 Erasmtron 中的 Corporate Meeting





理解是开发的重要方面。如今，尽管人物的表情还没有表现得很好，你还是可以得到很好的 3D 面部模型。这是因为设计这些面部模型的人都不是艺术家，他们都是计算机方面的工程师，所以他们也只能设计出从解剖学上看是正确的头部模型了。世界上每一个卡通人物的创作者都知道你所画的每一张脸都不会是真实脸部的再现。正因如此，我们不得使用卡通脸，而不用真实的面孔。



问：当我玩 Erasmaganza 时，有时会让我在三个动作选项中选择一个，而我根本就不喜欢这三个中的任何一个。从这方面看，它有点像是带有分枝对话树的古老的冒险类游戏。你不认为这是一个问题吗？

答：真正的问题不是“喂，你只得到了三个选项。”真正的问题是，不允许你生动地表达出合理的事，这也是故事世界设计中的一个缺陷。两个演示版的故事世界都有这个问题，因为他们都是很小的故事世界。如果你想摆脱这个问题，就必须设计一个更大的故事世界。“Brawl”中大约有 50 或 60 个动词，“Meeting”中大约有 100 个。我习惯地认为 500 个动词对娱乐价值来说只是皮毛。我现在更喜欢 1000 个动词。而“Meeting”没有给你很多选项是因为它太小了。

至于用户是否会对所给予的有限的选项表示满意，我根本就不认为会有什么。当然，在这种安排下，你不会有对细微之处做选择的权利。但是你会所有的引人注目的合理的选项。除此之外，如果我们给你一些可以选用细微之处的系统，以至于你会说“我将用有点讽刺的话来形容它。”那么以下的结构对这种情形来说会是可怕的。这会使游戏变得十分单调、乏味。因此我认为，可以有效地解决这个问题的惟一方式就是：把它限制在一个菜单结构中。事实上，有些游戏把执行细微之处作为他们交互作用的主要特征。在这些游戏中你和某个人相互作用，并且你会有这样一些可选择的滑块：一个是暴力，一个是幽默，另一个是魔力，这是你得到的全部。你对某个人用暴力、用幽默、用魔力的形式做出反应。我曾用了很长时间试探着把这些加入到 Erasmatron 中。但是问题是：第一，大众化；第二，界面得保持清洁和可用。现在，用简洁的菜单，只需要你去看和按一下就可以了，我认为这对大众媒体来说是很重要的。另外，状态滑块是专为游戏迷而设计的。



问：Siboot 用图标构造句子的方法和颠倒的分列线的使用是很有趣的。你为什么没有选择对 Erasmatron 用这样的方法？

答：因为 Siboot 中的大量句子都是自我完成的。在 Siboot 中，玩家只需点击一个图标；通常，句子的其他部分会自动填充，因为它们是可得到的惟一选项。顺便说一下，如今做这个的方法是用弹出式菜单。我可以在 Erasmatron 中利用这个方法。例如，假如你有这样的一个传统菜单，菜单上说：“你得用六支枪来换我的马。”这里的六支枪和马可以用弹出式菜单的形式来代替，提供更多的交易选项。这会需要更大的 Erasmatron 系统，但这不会很严重。我还没有这样做的惟一原因是，我不想再增加复杂性。我相信系统已经有了它所需要的足够的复杂性，然后再扩展。虽然增加游戏的复杂性是很容易的，但是我想使它简化。



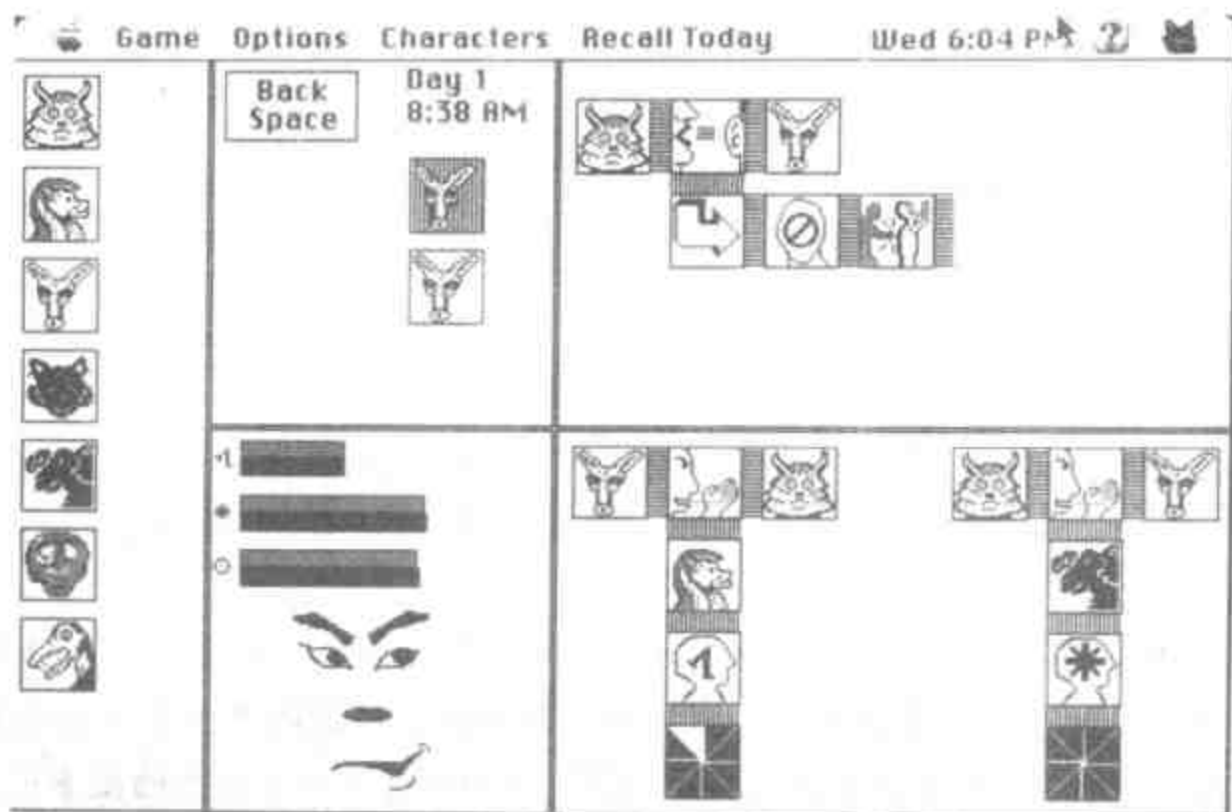


图 14.9 Trust & Betrayal: The Legacy of Siboot



问：好像有很多人成功地应用 The Sims 作为进行交互式故事讲述的工具。你用过它吗？

答：The Sims 不是进行交互式故事讲述的尝试。我和 Will Wright 就 The Sims 曾经通过几次 e-mail，他承认 The Sims 不是一个交互式的故事讲述平台，但是他指出有很多人在那样用它。The Sims 就像它所说的一样，是一个模拟而不是一个戏剧。没有一个戏剧模拟的是真实世界。莎士比亚的戏剧中，Henry V 在 Agincourt 对士兵的讲话中，没有说：“请稍等，我得去洗手间。”而在 The Sims 里，他说了。有一次，我玩 The Sims，有一个小女孩因有幽灵来吓唬她，她便无法入睡。顺便说一下，幽灵的引入很好。幽灵使她整晚无法入睡，她整晚到处走动直到睡着，因为 Sim 熬夜熬得太久就不会打瞌睡。她突然倒在她父母卧室的地板上睡着了。早上到了，妈妈醒来，伸伸懒腰，下了床，去洗手间，迈过了她女儿不动的身体！这是日常生活中自然过程的很好模拟，但它却是日常生活中情感过程的很糟糕的模拟。

Will 建立了一个极好的身体模拟。但是没有人物内容。这是直接与我所讲的“人物而不是事件”相背的，因为它所注重的是生活事物的方面，呆板的细节。去洗手间是那场戏中的一个重要组成部分，然而情感过程却不存在。我不想批评一个巨作：Will 制定一个很清楚的目标并且要实现它，这真是太好了。但他的目标不是我要实现的目标，你瞧，他也没有实现我的目标。我不会批评 The Sims，因为作为一个设计它是很好的。它有一个清晰的目标，并且很好地实现了目标。它只是一个不同的产品，而不是一个交互式的故事讲述。



问：那么是什么使你从事交互式故事讲述的呢？

答：其中有很多相关的原因。此外，我想它比设计游戏更为有趣。现在，我对电脑游戏中的设计问题感到烦恼，因为它们并不是复杂的问题。它们倾向于小模型，十分容易计算。我还对这些游戏中的低级智力要求感到胆寒。计算机中的敌人是个十足的白痴，那是仍使我感兴趣





的惟一因素。因此，我可能会去做一个有真正的人工智能的游戏，在这个游戏中，计算机中的敌人可以真正的比你聪明，我指的不是像下国际象棋那样，我所指的是要像战争游戏一样复杂。但是战争游戏自身太显而易见。我感觉我已经掌握了那种形式，那我为什么还要沉湎于其中呢？我有太多其他更重要的任务，例如，交互式的故事讲述就是一例。这是一个挑战！一个我真正可以沉湎于此的任务。不幸的是，看来我似乎是摸了老虎的屁股。



问：你担心过你会对 Erasmatron 感到不满意了吗？

答：我认为这是我一生的工作，这是我所有经历所达到的顶点。如果我继续做这个工作的话，毫无疑问，我还会改进这项技术。现在，我主要的疑虑是它在商业上的可行性。也就是说，我可以肯定20年后，人们将会意识到交互式故事介绍是一个极具商品价值的事情，并且是我们应该做的事情。我相信，我们能开发出一种产品，它们不再是资源管理而是戏剧性事件，因此，人们能够从中找到比在电脑游戏中更多的兴趣。但是，遗憾的是，人们却没有看到这一点。当然，游戏这个行业不会这样，而且以后也不会这样。游戏界的人们将会意识到：The Sims 代表的那个方向才是正确的。因为他们能继续使人的脸部变得更有立体感，并且使他们跳舞跳得更好等等。但是关于这种让人激动的决定，他们并没有开始着手。

也许，如果他们继续沿着这条路线走下去，给我和其他的一些关心交互式故事介绍的人留下那些实际的问题是一件好事。因为，这正表明对于戏剧性内容的渴望。例如，Sony 称在 PlayStation2 中的芯片为“情感引擎”。那是吹牛，纯粹的吹牛。那只是一个图表的处理程序，并没有与情感相关的模型。但是这些表明，他们或许的确想要实现一些情感的内容，只不过他们不想对这种水平的产品进行投入。还有 Internet 和好莱坞这样两个同时存在的因素，在两者之间，迫切要求通过电脑游戏来建立一种绝对的等同关系。因此，在 Internet 和好莱坞之间，我认为我们确实要有交互式故事介绍。在这个方向上有很多的迹象。6年前，当我手拿帽子去好莱坞想要引起他们的兴趣的时候，我很惊讶，因为他们都刚刚从拥有自己的游戏部分的兴奋中恢复过来了。所以，现在他们正在开始着手做一种基于网页的东西，这是一种外观完全不同的事物，他们或许对这个能感兴趣。



问：我想知道，你对批评家所说的交互式介绍故事有些违背了故事介绍的基本结构的观点持什么样的态度，而且显然你没有把这看作是什么问题。

答：我根本没有把它看作是什么问题。而且实际上，我为这种争论的浅薄而惊讶不已。举个简单的例子来反驳这种观点：一位爷爷给他的孙女讲故事，他说：“从前，有一个小女孩，她有一匹马……”这时，小孙女说：“是一匹白色的马吗？”这位爷爷没有说：“闭嘴！小孩子别插话，别破坏我编的故事！”而是说：“哦，是的，这匹马是白色的，像雪一样白。”就这样，小孙女和他互相问着答着，继续他们的故事。爷爷鼓励小女孩的参与，并把此融入到故事中去，从而使故事变得更精彩了。自从有了人类，这种讲故事的方式就一直存在着，那么长的时间证明，让人们介入到故事中不会破坏故事的内容。





问：在关于游戏方面的作品中，你总是注重内容和技术两方面，而在 Erasmatron 这本书中你却只是侧重于技术方面，而让其他人做内容方面的事。那么，你为什么要做这种转变？

答：因为很多人都能够提供艺术化的内容，但是，我却是惟一能提供工具的人，因此，从道义出发，我应该发挥那种只有我才有的天赋，去做其他人不能胜任的事情。然而，还有其他一些我要做的事情，而且还有多好的计划搁置在一边，因此我不得不好好分配我的时间。

问：结果，你没有很多时间着手研究 Le Morte D'Arthur，是吧？

答：是的，我只能下意识的想一想它，让它在我的脑子里多呆一段时间吧！不知道什么时候能发表，或许永远也不会。

问：那么继 Erasmatron 之后你会从事什么技术的研究呢？

答：我觉得，这种基本技术马上会向商业化方向发展。虽然我们应该做充分的准备使技术具备更加尖端等特点，但我们还是要立刻开始这种商业化的进程。因此我的下一个任务就是使技术商业化，但我还没有具体的打算。

问：你曾经对再做一个更传统的游戏感兴趣吗？

答：就这一点而言，我愿意请教设计各种游戏的设计师。也就是说，我不会介意大家一起研究设计基础问题，或者帮助别人研究这方面的问题。但是我不愿意在这个时候负责为游戏业生产商品，而会很高兴地帮助别人设计。可是这样的话，设计产品的过程就会变成有政治色彩而且让人不愉快的过程。这样，在产品创造方面就会花费越来越少的的时间，大部分时间会用在琢磨产品的政治性上，而这是我不感兴趣的。

Chris Crawford 作品一览

- Tanktics, 1978
- Legionnaire, 1979
- Energy Czar, 1981
- SCRAM, 1981
- Tanktics (修订版), 1981
- Eastern Front (1941), 1981
- Legionnaire (修订版), 1982
- Gossip, 1983
- Excalibur, 1983
- Balance of Power, 1985
- Patton vs. Rommel, 1986
- Trust & Betrayal: The Legacy of Siboot, 1987
- Balance of Power II, 1988





The Global Dilemma: Guns & Butter, 1990
Balance of the Planet, 1990
Patton Strikes Back, 1991



Chapter 15

第 15 章 游戏开发文档



许多游戏设计者都宣称自己擅长编写文档，并且说他们才是适合主管要求的惟一人选。这些固执的设计者们坚信游戏设计无法写到纸面上。其实他们只对了一部分；编写高质量的开发文档是很困难的。你所遇到的许多开发文档，看起来只是为了开发游戏的目的而写的，或许是用来敷衍那些需要看纸上文字的开发商的。然而，在制作电脑游戏的过程中，文档是必不可少的；并且设计者的工作就是要写出这些文档，并有效地使用。

提供游戏开发文档的必要性是日益壮大的游戏开发队伍所带来的副作用。在游戏开发早期，开发组的成员仅是一位多才多艺的人，游戏文档的作用并不重要。如果一个人能够设计出游戏的方案并加以实现，那么他是否编写开发文档就显得无关紧要。

当开发组成员由一人发展到 5 人，5 人发展到 10 人，十人发展到 20 人，20 人发展到 30 人以至更多的时候，保持设计的核心内容就显得越来越重要。随着开发组成员在某些领域的日益专业化，为了确保他们能够明白给定的系统是实现其功能的方式，以及确保他们的工作如何更好的符合设计，他们可以求助于参考文档，而这样的文档是十分必要的。因此使用了大量的参考资料，如设计文档、艺术文档、技术设计文档和其他大量的参考书目，它们来指导游戏内容的创作。开发文档是使设计不偏离主题的一个重要手段，它能够防止由于设计人员不切实际的



想法而使设计偏离主题。当游戏内容太多或设计不能及时完成时，把思路和故事情节写下来对于快速回想是一种有益的方法。

好的文档不仅有助于游戏开发，而且对游戏设计自身的提高也是有益的。Chris Crawford 已写了许多有关游戏设计的内容，其数量远多于他人。这点可以通过访问他的网站（www.erasmatazz.com）来得到证实。Crawford 用文档来记录他的想法，并描述出一个设计是怎样围绕其想法而演变出来的。就我个人来说，我用速记法来记下我对给定目标的想法。后来，我发现我能由这些标记明白我是怎样想出我所做的设计，并且能回想起那些当时想到而又忘了很久的好思路。

当然，在其他方面追求过多，或在文档上花太多时间，而文档实际上对游戏开发并无作用，这些都是可能的。拥有大量重复文档没有好处，尤其是设计人员感觉自己好像是在文档的海洋里随波逐流一样，没有任何实际意义。不借助任何文档来开发游戏也是可能的，但是，如果一个人希望在那种开发商业效果好的专业电脑游戏的开发室工作，那么习惯借助文档也是完全必要的。



15.1 为游戏编写文档

作为一个游戏设计者，你主要关心的是通常所说的设计文档，这一部分将在第 17 章中探讨。然而，在制作电脑游戏的过程中，还要借助许多别的文档。即使不使用它们，了解每个文档包括什么以及它们之间的相互关系是十分重要的。所以在深入研究设计文档的本质之前，有必要通览一下不同类型的文档。在不同公司、不同环境工作的人总能识别不同名称的文档。因此你应注意到我这里使用的命名方式是很特别的，但是整个游戏开发业使用的文档类型却是很普遍的。



15.1.1 概念（Concept）文档或 Pitch 文档或建议

一个确定的游戏一般都有正式文档。通常，编写正式文档是为了开发商批准（如果开发者本身不能出版其作品的话）或高层管理人员（在公司内部）。总之，这个文档是拿给批准委员会看的，目的是确保他们在这个创意上拨大笔资金以资助游戏开发，批准委员会是能够提供资金和批文的决策者，或随便怎么称呼。Concept 文档通常比较简洁，一般由委员会编写，其成员主要包括游戏的监制人、主设计者、主程序员，现有的市场开发商和提供各种各样舞台布景、创意画面和背影模型的主美工师。Concept 文档对游戏设计的所有方面进行了详述，包括市场定位、预算和开发期限、技术应用、艺术风格、游戏开发的辅助成员和游戏的一些概括描述。这些文档在游戏的实际开发过程中并不经常用到，尽管它们在其他文档开发的基石，如设计文档或艺术文档。由于 Concept 文档对于游戏的实际开发没有太大作用，这里就不做深入的阐述了。



15.1.2 设计文档

在软件开发业的其他领域，与设计文档等同的内容通常称之为功能说明书。实际上，一些





游戏开发人员就称设计文档为功能说明书。而我则喜欢称它为设计文档，因为它广泛地使用术语并能更好地展现了文档的内容。设计文档的目的是充分描写和详述游戏的操控方法。对于大型协作项目来说，设计文档作为重要参考工具，用来说明游戏各不同部分需要怎样运行。夸张一点来说，在整个游戏开发过程中，团队成员都要求助于设计文档。监制人员通常把设计文档作为一个跳板，依据它来安排计划。当一个游戏随后被转移到由不同开发组完成的另一个平台的时候，一个文字优美且完整的文档是至关重要的。当这些新手开始着手一个新的设计时，文档作为一个理想的参考工具，让他们明白游戏是怎样运行的。

广泛应用的功能说明书可以说是极其详尽和完整的。然而，由于游戏开发具有条理性、生动性、重复性的特点，所以设计文档不必是十分完整的，这已在第13章“游戏运行”中阐述了。对于一个大型协作项目的设计者来说，设计文档把你需要了解的内容做了一个基本的说明。设计文档的实质是游戏机制的逐一说明：在游戏环境里玩家能做什么，他们怎样做和如何产生激发兴趣的游戏体验。设计文档包括游戏故事的主要内容和玩家在游戏中所遇到的不同关卡或环境的逐一说明。同时也列举了游戏环境中对玩家产生影响的不同角色、装备和事物。一个设计者对设计文档精华的理解程度应当与新闻撰稿人对新闻故事的理解相似：即玩家做什么（玩家采取什么行动）、在哪儿做（游戏的背景）、什么时候做（在不同的时间和不同的命令下玩家采取不同的行动）、为什么做（玩家的动机）、怎样做（操控游戏的命令）。

设计文档也能用它未涉及的内容来界定。本章列举的其他文档所包含的大部分内容在设计文档中都不能找到，包括原稿、技术设计文档和背景艺术中的大量信息。特别提到的是，设计文档不应从技术角度花费时间来描述游戏的技术方面。平台、系统要求、代码结构、人工智能算法和类似的东西都是涵盖在技术设计文档中的典型内容，因此要避免出现在设计文档中。设计文档应描述游戏将怎样运行，而不是说明功能将如何实现。

同样，在设计文档中讨论游戏的销售、探讨和其他游戏比较如何进行市场定位以及销售计划等都是不合适的。此外应该略去计划、预算和其他目标管理信息。这些信息应该在这样一些文档上记录，如 pitch 文档或项目计划书，但应严格排除在设计文档之中。我想对于任何一位从事设计文档的人来说，如此排除是显而易见的。但是我就曾见过许多设计文档用其一半的纸张来讨论游戏如何销售。设计文档需要描述游戏怎样运行，以至于开发组成员能够确切地知道自己需要做什么。涉及游戏开发商业方面的更多内容，只能从更为合适的信息中获得。

设计文档及其开发将在第17章“设计文档”中更详尽地阐述。

15.1.3 流程图

流程图通常是作为设计文档的一部分而包含其中或自成一个独立的文档。以我的经验来讲，流程图在游戏设计的过程中实际上不是很有用，尽管它们有利于同团队的其他人员或开发商交流游戏应如何提高。在游戏开发中，流程图有两个基本用途。第一个是追踪玩家游戏外菜单选择路径，如玩家使用它们开始一个新的游戏或继续一个已存盘游戏。流程图也能用于描绘玩家在游戏游历范围，尤其适用于关卡游戏。流程图或者是手工绘制，或者是使用各种各样流程图制作工具绘制，如 Visio 软件。主要地，我发现流程图给开发商留下很深印象，而开发组却很少借助于流程图。





15.1.4 背景故事

因为游戏是在叙述故事，所以故事的大部分内容应包涵在设计文档中。当然，概括游戏故事的主要内容是很关键的，同时游戏流程的详尽描绘也需要包括故事部分，但不能全部包涵在设计文档中。如果被开发的游戏叙述的是涉及大量角色和地点的复杂故事，或故事发生在特定历史下的太空，那么前面讲述的情况也是对的。背景故事是为信息提供文档的最好场所。通常，游戏故事的作者对游戏范围外的太空及其人物有丰富的想象力，如游戏角色的来源和他们的动机，以及玩家玩时所处的那个游戏环境是怎样形成的。玩家所体验的仅仅是闻名遐迩的冰山的顶峰，故事作者头脑中形成的对游戏环境的刻画要比实际传送给玩家的详尽得多。有关太空其他方面的内容仅仅被提示。即使玩家不能够完全了解这一切，作者也能通过对背景故事制定一个全面的计划而更好地保持叙述的连贯性和合理性。

背景故事对游戏蕴含的复杂背景提供文档方面有很大的好处。正确的区分设计文档和故事背景，可以避免给设计文档附加一些与游戏设计主旨不相关的信息。使用大量的背景故事从而减化设计文档是一个简单的方法，这可以达到预计的深度和完整性，但是却掩盖了这样一个事实，细节部分未能充分表明游戏机制和一些更为重要的信息。然而，背景故事仍是重要的，因此背景故事文档的提供是有价值的。一旦背景故事被创作出来，当一位美工师希望对其塑造的角色有更多的了解时，他能求助于背景故事，找出角色的原型；他能使角色更加适合背景，而使其艺术才华得到更好的发挥。当一位配音演员想知道她该怎样演好这个角色时，如果她看过背景故事，那么她会像美工师那样依据故事信息来扮演好该角色。正确的使用背景故事，会使游戏更加连贯。

当开发组成员为了派生出来的项目而试图了解哪种新型故事能被精巧地制作成游戏的时候，游戏背景故事就显得十分有用，而由游戏创作出来的背景故事可能是连续的或脱节的。因为游戏背景故事要比原创设计包涵更多的内容，所以要给大量尚未开发的游戏配备新的人员。如果合理地按背景故事创作，那么新的游戏将与原有游戏保持更好的连贯性。当团队制作新的游戏时，故事能被扩充并不断更新，以至于未来的项目也能持续不断。

背景故事是完全公开的，故事的作者应该使版本尽可能符合他计划要涵盖的信息。通常，背景故事包涵大量不同篇幅的历史叙述。叙述可以描写游戏环境的来历，详述玩家玩游戏时导致其进入所处环境的主要事件。同样，文档也包括玩家在游戏中所遇到的各主要角色的叙述。重点描述的是角色的原型及他如何进入任何指定的地点，还有他的行动动机。了解角色背景，使作者在撰写游戏原稿的时候能更好的创造出不同角色之间激发兴趣并令人信服的对白。背景故事中还包涵不是很重要但比较合适的内容，即玩家在游戏中找到的各主要备品的来历。当非玩家角色向玩家谈论一把神奇宝剑的时候，可能会暗示它背后有一个动人的故事。一束奇异的光也可能有它自己多彩的故事。然而，作者应该是细心的，要在脑海中持有这样一种认知，不要觉得自己有义务对游戏中每一个关键角色和事物的世系加以解释。仅需要包涵的是对游戏的制作非常重要的信息。

背景故事的写作风格应当更具有散文的特点，而不应像设计文档那样重点突出。应用背景故事的团队成员很多都想立刻坐下来读几页，十分欣赏所读的故事内容。用角色、装备、主要事件来编写的文档对读者是有益的，因此使用恰当数量的有代表性的标题是一个好的想法。你





也可以期望文档中用各种各样的图表来补充文字内容，如时间表、事件流程图或角色关系树。在帮助读者了解格外复杂的游戏环境时，这些图表被证明是有用的。

另一方面，即使有一个复杂的游戏环境，你也可以不需要背景故事。如果游戏原稿作者头脑中能够反映出角色行动的路线及其动机，并且另一组成员连续工作的可能性很低，那么创作复杂的背景故事就不能充分利用每个人的时间。背景故事依赖于团队的创作风格，尤其是主设计者和撰稿人，他们可以是同一个人，也可以不是。当然，如果那样的话，许多有名的作者已经写出了要比你的游戏更为复杂的故事，他们自己仅留下少量的草稿。许多复杂的电影仅有一个原稿来讲述电影中的故事，演员们仅仅依赖于他们所要说的台词来解释角色的动机。原稿作者编写故事情节可能是出于个人目的，而不是与更多的人来分享。背景故事是一个工具，它有助于创作游戏故事；但是对每一个原稿作者或游戏设计者来讲，可能会觉得没有必要使用背景故事。



15.1.5 原稿

如果游戏叙述的是一个故事，在某种程度上玩家被迫听旁白、角色的对白，或者看下一个任务的信息。对白及其相应的情景描写（舞台指导说明）应包括在游戏的原稿中。撰写游戏原稿需要许多人：设计者、美工师、游戏的监制人，或者是只承担撰写原稿任务的人，还有特别雇佣的具有对话写作技巧的人。

依据伴随对白发生的游戏事件的类型，原稿可以呈现出不同的形式。例如，如果游戏具有电影式的片头动画，那么原稿可能十分贴近于电影原稿，带有玩家亲眼目睹的情节描写和任一特定瞬间摄影机所拍下的简单镜头。或者，这种片头动画的原稿更像是过分强调对白的戏剧。为了游戏的可交流性，原稿侧重强调对白，因为玩家一直在操控着游戏，从而操控着游戏发展的方向。但是侧重对白的原稿还可以包括伴随角色动画制作的描写或舞台指导说明，以帮助美工师创作合适的艺术画面来陪衬对白。

例如，下面是一个冒险类游戏中片头动画的原稿摘录：

当玩家在使 **TREE OF PLENTY** 复活后向 **ROGET** 和 **BARTLET** 靠近时，对于玩家的到来，**ROGET** 表现出明显的激动。对玩家的成就立刻流露出由衷的赞赏：

ROGET：那正是我们一直企盼的解决方法，你挽救了我们的圣灵树，没有什么能够足以表达我们对你的谢意，请接受这份满载我们感激的信物……

ROGET 把 **BAG OF FLIMFLAMS** 抛到玩家的脚下，**BARTLET** 走上前来：

BARTLET：[道歉地说] 我们知道礼物很微薄，但……

ROGET：[打断 **BARTLET**] 它是我们所仅有的！

BARTLET：[畏缩地说] 请不要厌恶我们的贫穷……

游戏非线性的特点要求原稿能区别于戏剧、电影或电视剧本来编排和提供。当玩家玩角色扮演类游戏或冒险类游戏时，如果玩家与非玩家角色有间接对话，那么原稿将需要采取一种独特的形式，这样有助于相互交流的非线性特征。当玩家听到不同片段对话的时候，原稿可以使用少量的虚拟代码，如用 **IF-THEN-ELSE** 或 **SWITCH** 语句来表达。





回到我们冒险类游戏的例子上来，对于更多的非线性对话，这里有一个合理的版面编排。冒险类游戏使用陈旧的关键词交谈体系，在玩家输入一个命令的地方，与之交谈的角色对于交谈的话题可能了解也可能不了解：

如果玩家问到“FLIMFLAM”：

ROGET: A Flim Flam 是一滴露水，清晨从空中降下，隐藏于树木的嫩叶之后。当涂在人的颈部时，对人有特殊的疗效。

如果玩家问到“TREE”或“PLENTY”：

ROGET: 在很久以前，树木是我们人类生命的源泉。没有树木提供阴凉，我们人类在烈日下将会精疲力竭。没有树叶，我们就没有食物。没有树木的挺拔，那么人类也会失去斗志。

如果玩家问到别的问题，如 DEFAULT：

ROGET: 我不知道你在说什么，陌生人。我们不是智力发达的人类；我们不像你这样的冒险家一般聪明。

在游戏中，对白的内容在一些表达方式上是随机变化；这些句子表达同一内容，但却用不同方式。在对白过程中，使用大量简单口语形式与玩家交流，游戏将随机地从不同对话内容中选择回答内容。

再以冒险类游戏来说吧，我们给出了对白例子，玩家在游戏实际操作中可能听过：

当玩家遇到 ROGET，他会说：

“请原谅，拜托了。”

或

“噢，亲爱的，我好像挡了你的路。”

或

“我母亲说我生来就给她添麻烦。”

这里没有行业标准语句来规定相互交谈原稿的模式。正是那些设计师、监制人和原稿作者们提出了一种模式，为其游戏中要使用的对话提供了最好的文档。

在原稿中，人们从任务简介或他们读过的一本书中可以找到有关角色的文本。游戏中包涵很多文本，从标记和墙上的海报到发给玩家的命令都包含在游戏原稿中。

当游戏试图与越来越多的故事相融合时，游戏中包含大量对白，对白内容做文档也就变得很有必要。当为游戏编写原稿时，要记住的最重要的事情是，人们玩你的游戏不是因为对白，而是因为游戏的可玩性。如果他们想看电影，那他们可以去电影院。那样的话，他们就不会玩你的游戏。在玩游戏时，他们也许喜欢听一些机灵的对话，但他们通常不喜欢针对于无穷无尽的故事背景而进行无休止的叙述。如果游戏非常好玩，那么玩家就会打算快速地了解这部分。如果玩家发现他们对游戏中的对白而不是游戏本身着迷，那么你需要反省一下，为什么对制作游戏感到如此困惑。





15.1.6 背景艺术

背景艺术主要由两部分构成，一部分是概念描述，另一部分是有助于美工师从事创作各种游戏视觉效果的资料。当游戏出现新的要素时，图像里会伴有文本，有时是手写形式的概念叙述，有时是美工师应遵循的文本叙述。背景艺术通常不是由设计师编写的，而是由开发组的美工师来完成的。当然，背景艺术中所包含的信息要与故事情节和其他文档中所描述的角色相对应、相一致。这些文档包括设计文档、原稿和背景故事。因此，当构造背景艺术时，美工师要与设计师、作者、监制人等合作，以确保他们的工作与游戏的整体效果一致。

在背景艺术的基础上，游戏的视觉效果和环境效果才能完全地体现出来。在游戏中，通过背景草图可以把游戏的艺术风格表现出来。这种表达效果远胜于语言表达的方式。在非设计文档中保留游戏艺术风格的描述是很重要的，这可使每个文档作为工具起到应有的作用。当然，项目设计师应当仔细阅读背景艺术描述，这样做的主要原因是可以了解到游戏的其他部分。背景艺术可以包含技术指导，美工师应以它作为标准，来创作与游戏相符的场景。技术指导应与技术设计文档中的内容一样详细。技术指导包括要遵守的多方面限制条件或在不同动画中的画面时间和画面数量。

15.1.7 故事剪辑

故事剪辑是电影电视的创作手法。在将故事拍成电影之前，用它来描述或模拟画面。故事剪辑可以是艺术背景的一部分，也可以作为独立的文档而单列出来。故事剪辑在编辑独立的片头动画方面是很方便的。片头动画的实质是电影，因此十分适合剪辑。开发组成员可以根据片头动画来提供反馈信息和改正见解，以免别人拍电影或使用这些情节时产生麻烦。如果还没有准备好上机演习，那么故事剪辑可以模拟出游戏世界是如何展示在玩家面前的。故事剪辑是很有用处的，它即可以确保全体开发组成员在游戏开发的早期明白游戏的设计方向，又可以让支持项目开发的赞助商们有信心。

15.1.8 技术设计文档

技术设计文档是设计文档的姐妹篇。设计文档阐述游戏怎样运行，而技术设计文档讨论怎样应用这些功能。技术设计文档有时称技术说明，它通常由游戏的主设计师来编写，编程组将其作为一个参考因素。在技术设计文档中，要对代码结构进行编辑和分析。编程人员可以求助它，来明白他们应怎样应用一个特殊的程序。文档中可以包含全部代码结构、代码的主要类型、使用结构的描述、AI 怎样发挥作用的描述，以及大量应用信息。在技术设计文档中，对虚拟代码不作要求，但要准确。尽管技术设计文档是个很好的创意，但在以前没有技术设计文档的情况下，游戏也能够成功地开发出来。实际上，我设计过的所有游戏，没有一个有技术设计文档，我所认识的从事游戏开发的人也如此。

正如我已经说的那样，技术设计文档主要由编程组使用。因此，一位经验丰富的编程师查看一些与他设计相关的技术设计文档将会有助于他的工作。因为技术设计文档里包含有关 AI、算法运行以及游戏操控要点的信息。正如通览背景故事对于设计师很重要一样，通览技术设计





文档能够使设计师确保程序设计组朝正确方向前进，即使设计师不能充分理解它。

15.1.9 计划和商业/市场文档

我在游戏开发文档中提到过这些，目的是为了强调计划预算和市场信息不属于设计文档。在我所读过的设计文档中，很多文档都包含如何销售游戏的部分。实际上，这些所谓的文档只不过是经过修饰的市场计划。在设计文档中，这些商业导向的信息既不准确也不属于我们这里提到的文档，它应属于概念文档。技术文档是设计游戏功能，并不是讲述如何打广告，如何销售。我们最好将这些市场计划和商业材料提出来，归为特殊的文档。这类文档最适合给对这些信息感兴趣的人看。

当你从事耗资庞大的项目，并且希望在资金上得到支持时，准备一份考虑周全的市场方案、市场预算、详细计划和一些相关文档是很重要的。所有这些文档对那些为你工作的开发商、市场代理人、广告师都很有用。在编写和保留文档方面，主设计师应尽其所能给予帮助，尽管这些是销售人员和管理人员的分内之事而非创作人员的职责。在很多时候，这却成了游戏设计者的责任。从良心上讲，确保为该项目提供资金的赞助商们了解他们所赞助的游戏，也应是设计师的职责。这样，当游戏设计完成又不值赞助商们在广告上所花的费用时，他们也不会太沮丧。而当他们对你的游戏感到满意时，他们更不会提出更改意见或取消投资。如果商人对你完成的作品真的感兴趣，他们会对扩大游戏的销售表现出极大热情，这意味着将有更多的人喜欢玩你的游戏。

15.1.10 非标准文档

不同的公司对指导游戏开发的文档有不同的标准。与我所提到的文档相比，不同公司对文档会使用不同的名称，我认为我所提到的文档种类基本上概括了那些公司编制的绝大多数文档。一些开发组将设计文档分成两个独立的部分，一部分只涉及游戏攻略策略，另一部分是关于故事情节及过关的叙述。一些开发组可以只编写一个设计文档，无需背景故事。一些程序设计组可能发现他们不需要设计文档。一些艺术总监可以使整个游戏开发过程中没有正式的背景故事。甚至，那些从事上百万美元开发项目的开发组在整个项目中根本没有提供任何文档。然而这种情况是我闻所未闻的。因为开发商要求提供文档，以便他们能对其所资助的游戏有更多的了解。进一步来讲，开发商希望有切实的证据，这些证据能证明开发组的确知道他们在做什么。通常，开发商所要求的文档数量与你及你们开发组的知名度成反比。你的开发组创立的越晚，越没有知名度，那么赞助商们越希望得到更多的保证，以便确保你没有浪费他们的钱财。

15.2 提供文档的好处

除了使赞助商们高兴之外，无论你是一个人在僻静的角落里开发游戏，还是与几十人的开发组共同工作，优秀的文档真能帮助你，使你的游戏更加完美。作为一位游戏设计者，你要对上面提到的所有文档的创作感兴趣。作为一名首席或高级设计师，创作及保留设计文档、背景





故事和原稿都是你的职责。所有文档可以由个人编写，也可以是集体智慧的结晶。例如，如果有现成的作家，他更适合创作对白，那就不必一个人编写对白草稿。作为一名首席设计师，你还必须关注故事、草稿、游戏玩法之间是否协调一致。确保不同文档之间彼此连续，并且与设计思路和游戏主旨相吻合，这些问题设计师都应认真对待。



Chapter 16

第 16 章 游戏分析: Myth: The Fallen Lords

由 Jason Jones 设计



Jason Jones, 他既是游戏设计人员, 也是游戏编程人员, 他设计的游戏总是使用别人没有用过的技术, 并领导着游戏技术的新潮流。他设计的第一个游戏名为 **Minotaur**, 是一个纯粹的网络游戏, 在那个时候 (1992 年), 纯网络游戏还没有开始流行。玩家在游戏对手是网络另一端未曾谋面的真人, 因此该游戏给玩家带来了前所未有的刺激。在他的游戏 **Pathways into Darkness** 中使用了简单的 3D 技术, 并把该技术与动作/冒险类游戏相结合, 从而创造了一个剧情驱动的游戏世界, 让人在其中流连忘返。此后, 他在游戏 **Marathon** 和 **Marathon 2** 中对 3D 技术进行了改进, 并把改良的 3D 技术应用到动作游戏中, 其设计思想比同期的其他第一人称视角



射击游戏更进步。最近, Myth 游戏系列给游戏方式指明了新的方向, 给玩家带来了策略游戏史诗般的战役方式, 这一点是其他游戏都未曾做到的。最重要的是, 其他游戏没有做到以技术去改进游戏方式。在 Jones 的游戏中, 技术和游戏设计是互为促进的关系, 这给玩家带来了独特的体验。

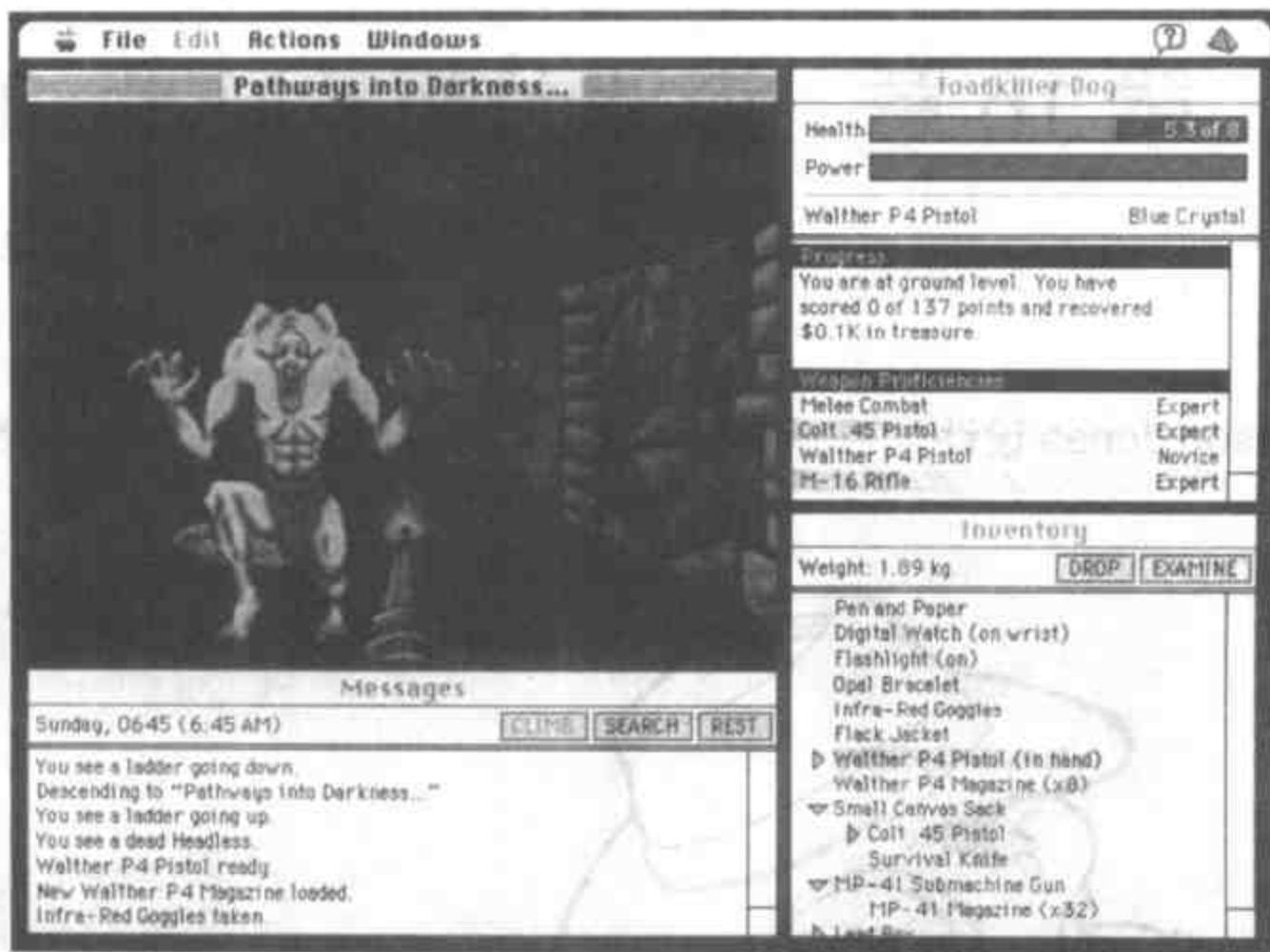


图 16.1 在 Jones 的第二个游戏 Pathways into Darkness 中, 技术帮助创造新的游戏



16.1 技术的使用

讲述如何在一个成熟的游戏类型中加入新的元素, 从而使该游戏类型展示出与众不同的新面貌, Myth 是一个最佳范例。毫无疑问, 在这个范例中, 其原始的游戏类型是类似于 Warcraft 和 Command & Conquer 的即时战略游戏, 远在 Myth 的开发之前, 即时战略游戏就风靡了好几个年头。由于这种类型的游戏相当流行, 并且从技术的角度看上去比较易于开发, 因此似乎一夜之间所有的游戏出版公司都推出了自己的即时战略游戏。不过, 充斥于市场的是一些对 Warcraft 或 Command & Conquer “克隆”的游戏。这些游戏大部分仅仅是对 Warcraft 和 Command & Conquer 进行功能上的仿制, 最多只有一些小小的改进, 例如改进了部队移动的路径点控制和增加了部队制造队列等等。这些改动并不能称之为革新, 正因为如此, 这些游戏没有让大众产生购买欲望。最终, 这些游戏在市场上消失得无影无踪。

从某种意义上来说, Myth 也是一个即时策略游戏, 但是 Jones 聪明地认识到不能仅仅去“克隆”那些成功的即时策略游戏。相反, 他对这种类型的游戏进行了与其他人不同的思考: 如何在更为底层的技术上对这种类型的游戏进行改造和改进? 如果把这种游戏采用的 2D 图像技术





替换成真正的 3D 图像引擎会怎么样? 除了他的第一个游戏 *Minotaur* 之外, Jones 所有的游戏都是 3D 的, 因此他决定继续在新的项目中使用 3D 技术。然而, 即时策略游戏中加入 3D 技术并不是仅仅为了满足视觉需要。Id 公司的 *Wolfenstein 3D* 游戏, 几年前的前身是一个较简单的动作游戏, 但是通过使用 3D 技术, 大大地改变了游戏设计的性质; 与 *Wolfenstein 3D* 一样, *Myth* 采用了即时战略的游戏方式, 并对其进行修改, 以适应新的 3D 技术。其结果就是创造了一个新的游戏, 而不是简单的“克隆”。

该游戏的设计方针并不只是使用了 3D 技术而已。其开发出的 3D 引擎是当时唯一支持户外场景建模的引擎, 从而可以支持即时战略游戏方式。Jones 没有把他以前的 *Marathon 2* 游戏中的 3D 引擎用到即时战略游戏中, 而是着手建立一个新的游戏引擎。*Marathon 2* 使用的是 *Doom* 风格的 *BSP* 引擎, 只适合简单的室内环境应用, 并不能满足即时战略游戏的要求, 即时战略游戏要求的是一个开放的户外环境。因此, Jones 开发了一个全新的引擎, 以满足 3D 即时战略游戏项目的开发。

毫无疑问, 使用了 3D 技术之后, 由 *WarCraft* 和 *Command & Conquer* 建立的即时战略游戏的基本设计模式就必须改变了。在 *Myth* 中, 地势的高低会很大程度地影响到了各作战单位的作战性能。把弓箭手放置到山顶将会使该弓箭手获得最大攻击能力。如果把弓箭手安排到峡谷中, 那么他们的命运就是被屠杀。同时, *Myth* 中也使用了一些简单有效的物理系统, 用来强调 3D 地形的特性。当玩家派遣一个侏儒投弹手向山顶上的敌人投掷燃烧瓶时, 瓶子可能在爆炸前顺着山坡滚下来。如果炸弹命中目标, 玩家会对爆炸的视觉效果感到惊讶, 爆炸后会留下弹坑, 爆炸对地形的改变会一直持续到游戏结束。如果目标被消灭, 玩家会看到敌人的残肢碎片, 并可能顺着山坡向侏儒投弹手滚去。



图 16.2 使用 3D 地形引擎, *Myth* 在即时战略游戏类型中加入了新的游戏操控元素





使用 3D 引擎对即时战略游戏带来的另一个显著的改进就是，玩家可以看到以前俯视或者斜 45 度方式 2D 游戏所不能提供的更为详细的战场细节。在 2D 即时战略游戏中，往往会发生某个障碍物挡住游戏单位的情形，但是现在玩家可以通过旋转镜头来看到被挡住的部分；也可以用该办法来选择某个战役的最佳观看角度。不仅如此，玩家还可以通过放大和缩小镜头来观看战斗场面。“放大”带来的一点好处在于，可以近距离地观看一场战争，这是以往的 2D 即时战略游戏不能提供的功能。玩家视角和以前也不同了，玩家可以用比以前的战略游戏低得多的角度去观看战争。毫无疑问，加入镜头调整的原因部分是出于美工上的考虑，部分是出于游戏可玩性上的考虑。无论动机如何，Myth 提供的近距离视角功能给玩家带来了更为刺激的游戏体验，与以往的即时战略游戏提供的远距离视角相比，Myth 游戏中一个个游戏单位变得更为重要并更加真实。通过这种方式把 Marathon 之类第一人称视角射击游戏的亲历感与即时战略游戏的技巧性嫁接在一起，最终带来的是新的游戏体验。

在 Myth 中使用的 3D 引擎并不是很成熟，尤其是用三年后的标准去评价这个引擎。例如，在地图上的角色不是完全 3D 多边形的，而是简单的动画精灵。毫无疑问，这是由于屏幕上一次需要显示众多游戏角色造成的原因。如果一个即时战略游戏任何时候在屏幕上最多只能显示三个单位，这个游戏还会有什么好玩的呢？甚至到今天，如果一次要渲染大量的 3D 世界，在屏幕上也会让我们的 PC 变得很慢。

在 Myth 中，把技术发挥得淋漓尽致，每一个技术都在游戏中发挥了其最好的效果，这是 Jones 设计并编写的游戏项目中的显著特点。这位多能设计师懂得何种技术可以在游戏中最好地表现，并且了解如何让游戏方式引人注目，从而使游戏开发最经济。一旦发现某个技术可以增加游戏可玩性，设计师和程序员会立即采用它，并让该技术在游戏中发挥最大的效果。这一点和其他的许多游戏项目不同，在那些项目中，程序员经常抱怨程序的功能没有被完全用上，因为设计人员根本没有完全弄清楚程序员的意图。

当然，把 2D 改为 3D 在游戏可玩性上并不是没有缺陷。例如，在 Myth 中虽然可以放大和缩小某个区域，但是只有部分人会这样做，还有一部分人根本不愿意在战斗中对场景进行缩放。这里的原因一部分是因为以往的即时战略游戏带来的习惯；以前的即时战略游戏采用 2D 引擎，因此具有非常远的视角，玩家通过这个视角可以去观察、追踪和移动大量的游戏单位。Myth 在正式版发布后不久后就推出了一个补丁程序，在补丁中玩家可以用更远的视角去拉远镜头，从而可以看见更多的地图区域，不过带来的副作用就是降低了游戏的帧速率，这是因为当前视野中包含了更多的地貌和多边形。其实，游戏引擎可以让玩家用更宽的视野去观看地图，但是在画面中增加的多边形会带来限制，这样会把游戏的整体速度降低到不能忍受的地步。因此 3D 引擎的局限限制了设计人员能使用的游戏操作方式。由 3D 技术带来的另一个缺点是，让玩家失去方向感。虽然玩家可以旋转镜头来从任何角度观看所需的画面，但是镜头的旋转不利于玩家定位方向，让玩家分不清到底哪个作战单位分布在地图的哪个位置。对一个新手、“过路”玩家或者方向感不强的玩家来说，不容易掌握镜头的移动。





16.2 游戏重点

Myth 游戏也是游戏重点设计合理的典型范例。我们在前面讲述过, Myth 是在另外两个成功的即时战略游戏 Command & Conquer 和 WarCraft 发行几年后才出现的。在这两个游戏中, 玩家需要修建获取地图中自然资源的建筑物, 然后才能建造其他的单位。然后, 玩家才能用不同的方式去引导自己的单位与对手作战。这种方式的即时战略游戏是一种混合游戏操作, 一方面需要进行资源管理和建筑, 另一方面才是战争。后续的一些即时战略游戏, 无论是成功还是失败的作品, 都是原样照抄这个通用的游戏方式, 把玩家的工作分成生产作战单位、资源探索和从策略上配置各个单位等几个部分。



图 16.3 Myth 摒弃了其他即时战略游戏中的资源收集和管理功能, 把重点放在了作战技巧上

但是在 Myth 中, 并不需要采矿, 也不需要为修建什么建筑去收集资源。相反, 玩家的所有注意力全部放在作战技巧和作战体验上。在游戏关卡的开始, 就为玩家配备了一定数量的作战兵援, 并且在大部分关卡中, 玩家所拥有的也仅仅是一开始就配备的兵援。在某些关卡中, 玩家在关卡中会得到一些附加的兵援, 但这种关卡毕竟是少数。Myth 只保留了其他即时战略游戏中的战斗元素, 而摒弃了其他方面, 使得游戏的关注点更加集中。

在整个游戏设计中, 以战术为重点的方式有几种好处。首先, 由于不需要考虑建立资源收集系统, Jones 可以集中精力处理战斗模块, 并尽可能地把它设计得完善。这导致该游戏比同期的其他即时战略游戏更为成熟, 在战争细节上更为完美。在 Myth 中, 兵援的朝向、编队方式和部署比其他的策略游戏更为详细。由于不需要考虑玩家是如何使用资源的, 设计人员可以在物理系统模拟和其他可以增强战斗体验的技术上花更多的时间。例如, 可以考虑一些细节, 让弓





箭手在树丛中射箭，需要考虑树是否会挡住箭，考虑天气对箭的飞行轨迹有什么影响，地形的高低会如何影响到箭飞行的距离等等。

不给玩家提供制造额外兵援的机会，会让玩家在开始一个关卡之前考虑使用什么类型的作战单位。在 WarCraft 中，玩家如果在关卡开始时犯了一些错误，也可能最终夺取胜利，因为他可以在采集到足够的资源和建设足够的作战兵援后再去夺取胜利。在 Myth 中，这种错误往往是致命的，越来越难的关卡对玩家犯的错误的饶恕的机会越少。玩家攻击计划失败后惟一的机会只能是重新装入这个关卡。因此这个游戏的操作方法就不同于 WarCraft。在 Myth 中，玩家需要深思熟虑后才能开展某项行动，鲁莽行事是行不通的。玩家能够获得的兵援有限，因此玩家必须考虑每个队伍的安全。在 WarCraft 中，制作一个作战兵援是很容易的，所以玩家只把这些兵援当作炮灰看待。与之相反，Myth 中的一个特定兵援可能对完成关卡任务起到至关重要的作用，一旦这个兵援被杀就回天乏术了。



16.3 剧情描述

Myth 不仅含有经典的游戏设计方式，而且其中的剧情叙述也占了很大比例，游戏中使用了一些相当不错的综合剧情讲述模式。第一种模式的剧情介绍是截屏方式的，这在整个游戏中只是零星地出现，用于描述游戏中重要的地点和特定关卡的设置。这些截屏方式的介绍类似于“电视插播广告”形式，对剧情的推动作用也不是很大。第二种模式是每个关卡之前的任务简介。任务简介对于“光明”和“黑暗”双方的战争细节介绍得比较详细（“光明”和“黑暗”是游戏中对立的两个派系）。任务简介向玩家讲述将要进行的关卡的意义是什么，让玩家感到即将进行的关卡需要完成的任务是什么，而不是仅仅感到需要完成游戏设计人员强加给他们的任务。

第三种方式就更有趣了，其中使用了一种游戏内嵌的剧情讲述模式。当然，含有这种内嵌方式剧情讲述模式的关卡只在符合游戏剧情主线的地方出现，无论这些地方是潮湿、荒凉的山区，还是火热的熔岩区域。关卡中包含的战斗和任务与关卡开始的任务简介是一致的。但是玩家也可以看见和听见不同角色在关卡中的反应。例如，小镇居民可能会给玩家指出裁缝家在哪里。玩家的部队可能会对玩家做出一些建议，例如：“我们最好回到桥上去！”诸如此类的对话是在游戏关卡中的特定场所被自动触发的，玩家并不会因为这样而失去对部队的控制权。在一个任务中，当玩家的军队中遇到大批的盲从者（Myrmidon）之后，玩家在游戏中守卫的天神就会走出来宣布：“让我来处理吧！”。然后他会开始与敌人的首脑巫师进行交谈，接下来游戏的剧情就会在游戏时刻展开。

很多游戏都只把剧情讲述当作已知关卡的附属品处理，与之相比，在 Myth 中，游戏主线、关卡和游戏操纵得到了有机的结合，互为依赖，互相补充。玩家喜欢这个游戏是因为他们喜欢这种操作方式，而不是这个游戏中有许多冗长而缺乏交互性的截屏式剧情介绍。玩家还是希望在游戏中有剧情介绍，因为这样他们可以了解这个游戏的目的和意义。介绍一个完整剧情最佳的方式是把剧情介绍和游戏操作结合起来，在整个游戏过程中这个地方介绍一些，其他地方再介绍一些。在这个方面上，Myth 做得相当专业。当然，实际上 Myth 的故事主线相当优秀，剧情脚本写得很好，其中专业的语音效果也起到了相当大的帮助。如果故事情节很乏味、对话很





矫揉造作或者语音效果很差,那么即使采取用游戏操作来讲述剧情的方式也不会让游戏变得更好。

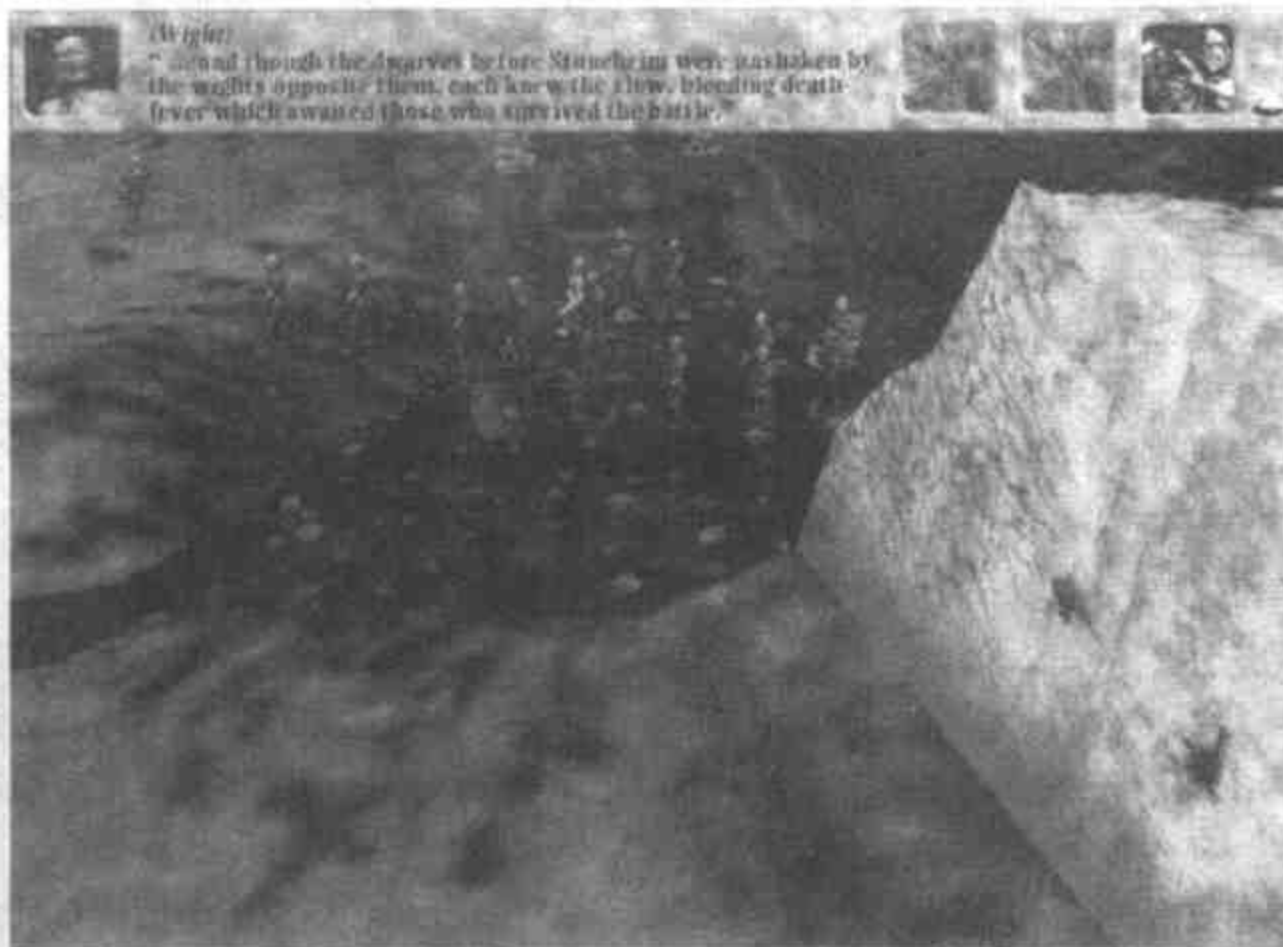


图 16.4 Myth 把任务简介、关卡设计和游戏操控有机地结合起来,讲述了一个引人入胜的故事



16.4 专业级游戏

Myth 是由专业级玩家设计并面向专业级玩家的一个游戏,并且坚持不在游戏的难度上进行折衷。许多公司的游戏在市场定位上是面向“主流”玩家和“过客”玩家的,但是 Myth 的设计理念却不是如此,如果有人对其他即时战略游戏不是很熟悉或者动作不灵敏,那么他很快就会被 Myth 的难度吓退。这当然不是一件坏事,而且我们很高兴看到某个游戏能够坚持自己的艺术理念,并相信该艺术理念对玩家有一定号召力。实际上,既然整个游戏的设计人员是一些专业级别的游戏玩家,他们当然最清楚什么是玩家喜欢的。通常,当一群专业级的玩家要制作一个让想像中的“过客”玩家也能喜欢的游戏时,他们会制作一个自己喜欢的游戏,这样那些“过路”玩家也不会太在意游戏的难度。如果要强求一个艺术家去制作一个与自己创作风格不一致的作品,结果就是谁都看不懂他的作品。

但是 Myth 没有这样的问题:游戏制作人员的设计并不是面对“过客”玩家的。理由之一是游戏控制相当复杂并富有挑战性。以 3D 旋转镜头为例,与同期的即时战略游戏相比,这些游戏的镜头只能在水平方向上移动,Myth 的镜头可以平移、放大、缩小、围绕某个点转动或者旋转。就是对相当有经验的游戏玩家来说,要适应这套系统都有一定难度。然而,一旦玩家掌握了移动镜头的方法,会发现这种镜头移动方式的设计是多么的专业,可以提供各个自由度上的移动,





他会觉得游戏中使用这种技术非常合理。游戏中对不同的动作也提供了特殊的按键，例如编队、特殊动作和切换攻击方式。同样，玩家一旦掌握这些命令，就可以随心所欲地控制部队的行进和攻击，但是要掌握这些命令确实需要花些时间。实际上，这些组合键使得在游戏中仅靠一个鼠标是不行的，在这个方面几乎所有的其他即时战略游戏都是如此。其中“gesture-clicking”也是一个有趣的功能。不过该功能太强大、太复杂了，以至于如果没人教或者不花大量时间练习的话，根本不可能学会。不过，对于专业级玩家来说，如果花了大量时间去学习那些复杂的控制，最后的结果就是能在游戏中获得很大的乐趣。

Myth 是一个很难的游戏。即使对于一个相当熟悉即时战略的玩家来说，刚上手玩 Myth 也是有一定难度的。一般来说，一个游戏的开头几关都是比较简单的，这是为了给玩家一个熟悉控制的机会，给他们提供一些作战的经验和机会。但是 Myth 却反其道而行。一开始玩家就会碰到很难完成的任务，只要犯一个错误就可能难以完成关卡任务。老练的玩家会发现，如果丧失某个特定的单位，就会难以在游戏中继续维持下去，他们会觉得没有必要苦苦支撑下去，不如从头开始这个关卡。最令人伤心的是，游戏难度非常高，虽然游戏开头的关卡相对简单，但是游戏的难度从此以指数的级别增长。然而，真正的玩家会觉得这才是真正的挑战。这不是那些不公平、人为的或者无法预料的挑战，至少不总是这样。大部分情况下，玩家可以首次玩某个关卡就能通关；不过要一次通关也会相当难。

大部分游戏出版单位都会要求游戏足够简单，从而非专业级玩家不会被游戏的复杂控制或关卡难度而吓跑，但是 Myth 不是这样。但是，如果这个游戏相当简单，游戏还会如此引人注目吗？可能不会。尽管可能会争取到小部分“过客”玩家，但是会失去众多的专业级玩家。



16.5 多人游戏

和前期作品 Marathon 一样，Bungie 把 Myth 定位成既能够提供单人游戏又可以支持多人作战。Bungie 在这两方面都做得很好。许多游戏在这两方面都是顾此失彼。例如，Quake 和 Quake II，这两个游戏中的网络联机性能受到了一致好评，但是大家对它们的单机游戏评价较低。很多其他游戏的多人游戏支持都是事后才想到的，希望能做到锦上添花。Centipede 3D 就是这样的一个例子，出于市场考虑，在项目的后期才添加了多人游戏功能，而程序员几乎没有花时间去完善多人游戏功能。

Bungie 公司制作游戏的宗旨是：在单机版和多人版上都要胜过他人。当他们建立了游戏的核心引擎技术之后，下一步的工作就是满足网络功能。一旦游戏具有了网络功能，整个开发小组开始玩这个网络版，直到他们认为游戏很有趣为止。此时，单机版本的制作工作还没有开始，开发小组的全部注意力都放在加强网络多人游戏体验上。直到网络功能的核心设计完毕之后，他们才开始设计单机版的关卡。然而，并不是说单机版本是匆匆忙忙赶出来的。这不过意味着整个开发小组知道在单个关卡创建之前哪些工作会让游戏更有趣，从而不至于对单机关卡进行返工，并且在最终版本保证游戏的可玩性。

由于开发小组在多人联机游戏中花费了足够多的时间，因此游戏的网络功能经得起时间的考验。如果开发小组开发的游戏玩起来没什么意思，那么连他们自己都会对这个游戏感到厌倦。





Myth 的多人游戏功能给玩家提供了多种不同方式的玩法，各种方式的玩法都不相同。其中最有趣的是，游戏前的作战部队出售系统，让玩家可以考虑如何建立自己的“杀手”小组，这有点类似于 Magic: The Gathering，玩家需要花时间去挑选自己最好的牌。其中一种方式为“团队游戏”（Team Play），在这种方式下，多个玩家分别控制一个联军中的各个作战部队，从而对抗敌方团队，这比单人游戏中的策略要复杂得多。由于 Bungie 公司的开发人员在游戏的多人游戏功能上花了很多时间，因此游戏才会令人注目。



16.6 整体表现

Myth 在稳定的核心设计的基础上也没有忘记对游戏进行润色。其他即时战略游戏的任务一般都是相互独立、自包含的游戏空间，但是 Myth 游戏中由于使用了部队升级系统，使得所有的任务形成了一个整体。如果某个部队能够从一个关卡的战役中生还，那么玩家可以把他们带入到下一关，并且他们的技巧比新的部队要明显高一些。这让玩家重视每一个这样的部队，如果需要侦察危险的地形，他们会派出新兵去冒险，而舍不得牺牲老兵。在 Myth 中还有一个细致的设计，部队会在地面上留下脚印，玩家可以通过雪地中的脚印去追踪敌人的部队。游戏中提供的任务是各种各样的，比一般的即时战略游戏的任务类型多，玩家必须在不同的关卡按照不同的任务采取不同的作战策略。

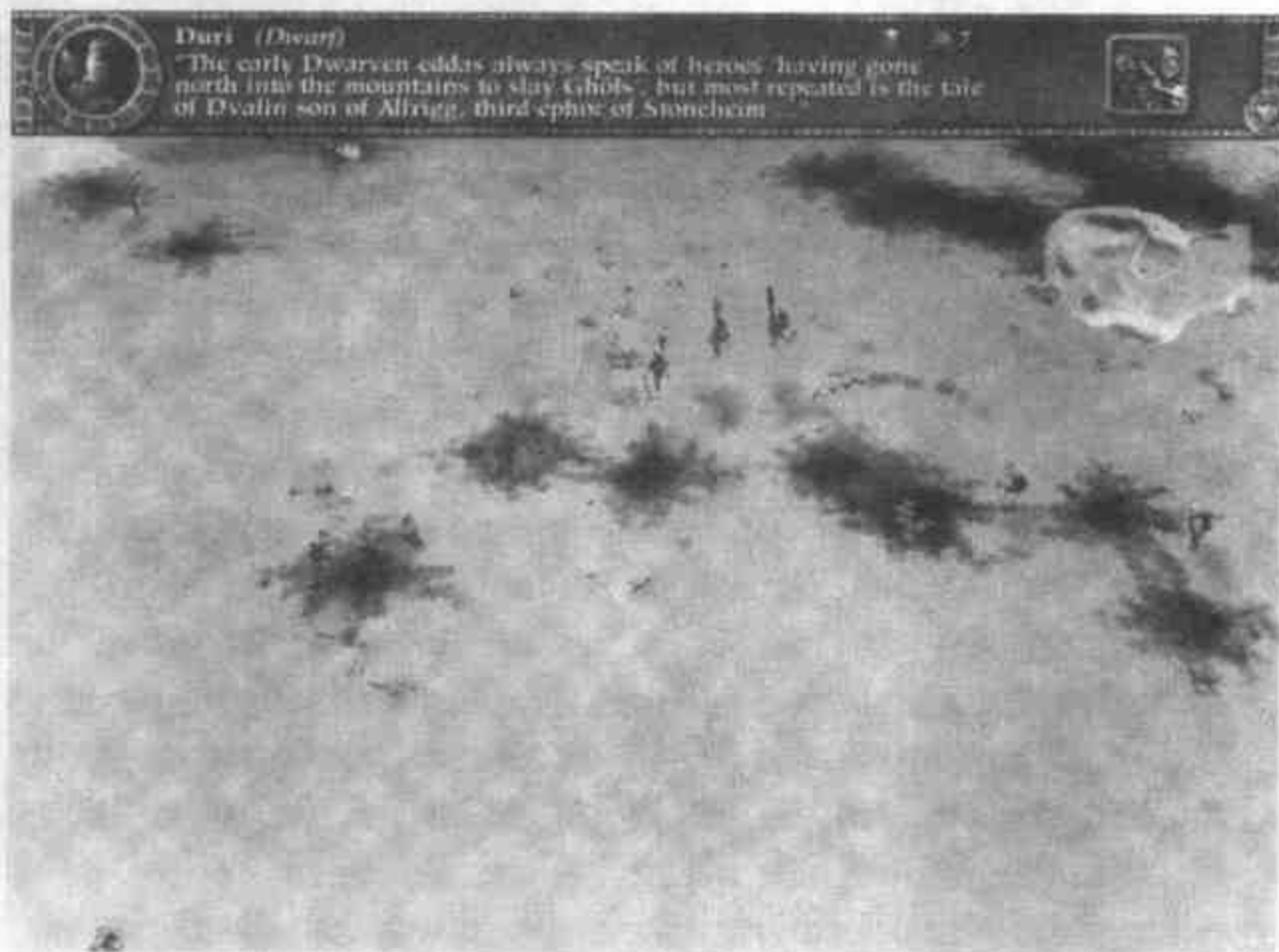


图 16.5 Myth 的开发人员相当重视细节描绘，这给玩家提供了深刻的游戏体验

当然，Myth 并不是十全十美的，即使对一个能够接受复杂的游戏控制键、并且不畏惧复杂关卡的玩家来说也是如此。有时候点击顶端的地图会让镜头转到预料之外的方向，从而让玩家迷失游戏中的方位感。顶端的地图是半透明并且是覆盖在场景地图上的，玩家很容易无意点中



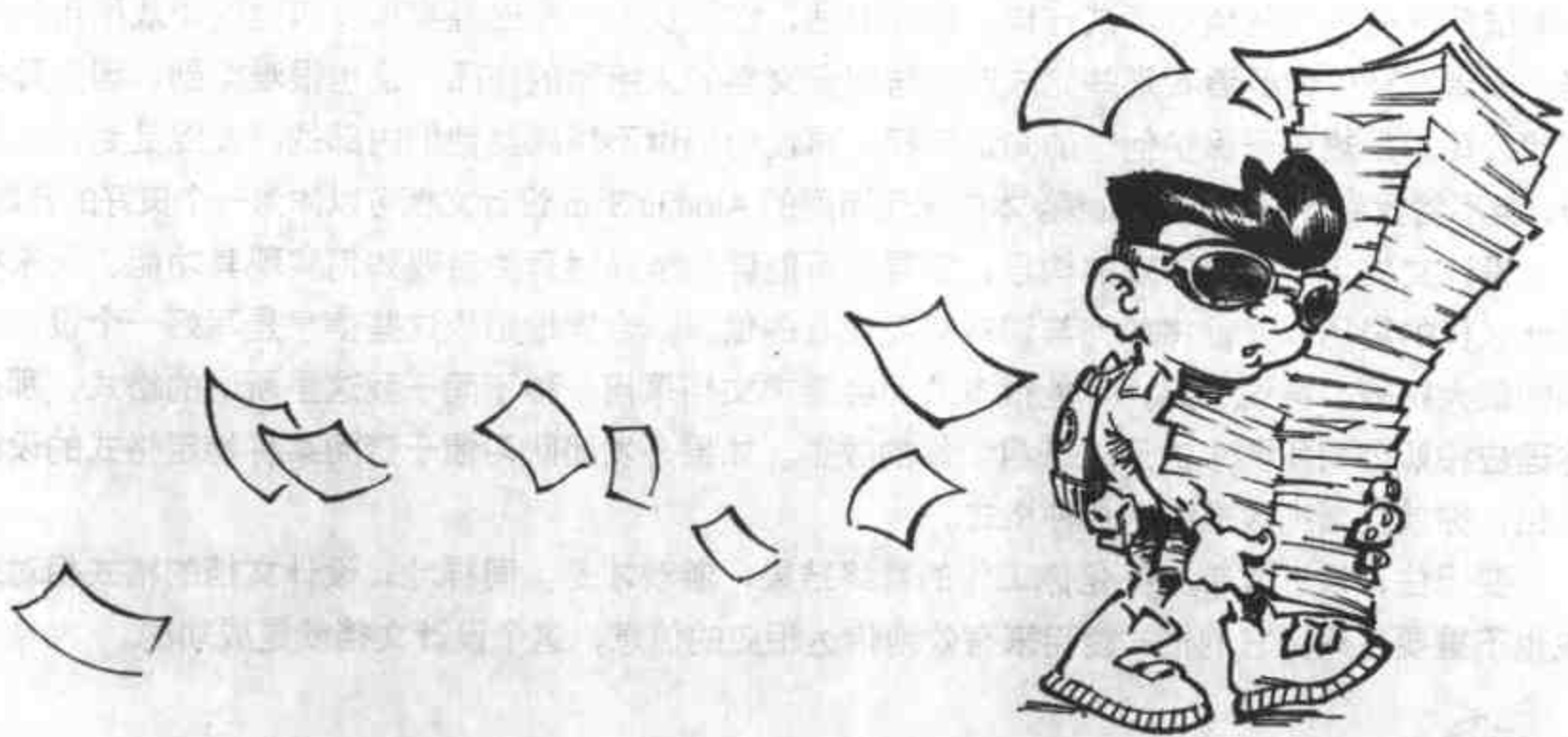


它。而且，玩家不会理会关于技术限制的解释，他们还是希望能够在—个屏幕上看到更多的战场区域。所以，虽然整体设计是非常优秀的，但是在游戏中还是存在一些微小的瑕疵。Myth是技术与游戏可玩性完美结合的一个代表。这不是短期的“婚姻”，新娘和新郎相敬如宾，互助共进，从而创造一个美好的明天。



Chapter 17

第 17 章 设计文档



“直到发行 *Ultima IV: Quest of the Avatar* 之后，*Ultimas* 才确实有了引人入胜、意图明确的故事情节，它是这一系列中第一个有社会内涵的游戏。我不只想建造一个宏伟、壮观、有趣味的环境，我还想给每个游戏增加更深远的意义，虽然在所扮角色的行动中，这种深层含义不一定表现得很明显，但它最终将给这个游戏更长久的含义。所以，在 *Ultima IV* 中，你得证明自己是个好人，是个可以作为大不列颠典范的人。这个游戏的行为有点像‘大哥’，你要表现出英雄气概才能赢。我认为这个设计相当酷，因为游戏者总是扮成一个英雄，并且要痛打镇上所有的家伙，只有这样他才能打败那个被设定为大坏蛋的角色，虽然总的来说这个家伙没干什么坏事。”

——Richard Garriott

多年以前，当我还是一个野心勃勃的专业设计者时，我一直希望有人告诉我设计文档的正式格式应该是什么样的。我知道好莱坞的电影剧本都有着精确的格式，所以我想像设计文档一定也会有比较严格的东西。它里面应该包含哪些信息？它应当如何展开？它该用什么格式？直到最近，在从事这个职业多年之后，我才得知了这个大秘密，而我很高兴能在这本书中把它传授给你。是的，在这里，我在游戏行业中的多年经验将提供有价值的信息。

根本就没有格式！每个编写游戏文档的人只按照他们自己的格式来写！你听到过这么不可



思议的事情吗？不论何时，我问别人我的某个文档该用哪种格式，他们总是千篇一律地回答说“嗯，你知道，标准格式。”没有人真的清楚这个神乎其神的“标准格式”是什么样的，但每个人都提到它。最终，只要能有效地传送游戏的本质并且包含充足的细节，不管你交给别人什么样的东西，它都会被看成是“标准格式”。确实，设计文档内必然包含某种类型和数量的信息，但如何记录这些数据却没有一个标准化的形式。

当然，在一些公司内部，特别是一些大公司，可能会有一些统一的格式，公司内部的设计师必须在文档中使用这些格式。如果你的设计文档和行业内的其他文档差别太大，可能会显得太特殊而被否定。最好去参看一下你能找到的所有正式设计文档，就好像要去参加重要的标准化考试前先去进行些模拟考试一样。乐观的话，你能获得一些已经实际发布过的游戏所用的文档。或者，至少你要看看那些完成游戏与配套文档的人所写的作品。这也很难得到，因为那些游戏公司总是热衷于保护他们的知识产权，再说他们也不想暴露他们内部的开发混乱到什么地步，但不管找到什么你都得看看。本书末尾所附的 Atomic Sam 设计文档可以作为一个良好的开端。

设计文档用于传送游戏的构思，它要尽可能详细地描述有关游戏如何实现其功能、玩家将有什么样的经历以及他将如何与游戏环境交互的信息。合理地组织这些信息是写好一个设计文档的最大挑战。再说一遍，可能很多公司会要求文档采用一种不同于我这里所讲的格式，那么你理应按照公司所要求的形式来组织你的数据。如果开发团队习惯于查阅某种特定格式的设计文档，你就应当把数据纳入那种格式。

要记住，设计文档并不是你工作的最终结果，游戏才是。同样地，设计文档的格式相对来说也不重要。只要它的格式能用来有效地传达相应的信息，这个设计文档就是成功的。



17.1 写作风格

在探讨设计文档应当分为哪几个部分及涉及哪些领域之前，最好先探讨一下该用哪种风格来写作。因为设计文档是作为参考工具使用的，所以，要尽量让它易于检索和查阅。要做到这一点首要的就是做个好目录，关于目录我们马上就会谈到。在写作文档的正文时，应当把它们分成具有层级的大标题、标题、小标题的形式。这会使读者更易于浏览文档并定位所需的信息。把你的信息列入清单，在所有可能的地方标上序号或圆点，这将进一步允许读者轻易了解游戏的哪一部分要有哪些不同特点。实际上，标圆点风格的写作更难一些，因为它要求你不停地转换缩进和粗体字标题，而不能只在一个叙事段落中写下所有的想法。你会发现最简单的方法是先写出你的文档，然后再编排文档格式。使用这种先写内容再编辑的方法，可以同时把它转换成合适的格式。虽然用圆点风格来写作会花去你更多的工作时间，但最后写好的文档却对你的团队更有用处。更重要的是，经理们和行政人员们会更欣赏它，因为它使文档更加易于浏览。

有些设计者使用特殊写作工具来编辑文档。这些工具可能更适于写那些带有很多标题、副题、项目清单的文本。这些品目繁多的工具允许自动格式化和正文缩进，能够节省大量安排拖放标志和制表位的时间。虽说如此，我从没用过这样的工具，也没有见过哪个和我一同工作过的人用过。这些工具的主要问题是，文档一旦写好，需要用电子方式把它们传递给所有要读它



的人。而所有这些都拥有这种独特的格式化工具的机率是微乎其微的。相反，每个人都有常规的文字处理器。从你所在开发小组的其他成员到管理人员直到出版公司的行政人员，每个人都要读你的文档。你不可能期望这些人都安装了你选来编辑设计文档的工具。如果你用的工具可以像 Rich Text Format (.rtf) 一样把文件转换成为标准文字处理器格式，那么这个问题可以解决，但要确保它导出的文件和你编辑的一模一样。还有，我通常使用标准文字处理器来完成工作，没觉得需要一种更强大的工具。

虽然尽可能扩充你的文档使其完整详尽是一个极大的诱惑，但应当想尽一切方法避免重复的信息。在你提及游戏的某一元素，而它直接依赖于你在十页前讲到的另一系统时，避免重复的信息显得尤为困难。不要再描写一遍那个系统，而应让读者参阅它的原始定义。这非常重要，因为这样的话，当你随着项目的进程而更新文档时，只需要改一处而不是多处。在很多情况下，如果你在不止一个地方详细论述同一个游戏机制，那么当要改变它时，经常会只更新其中的一处。这会使其他地方的描述过时，使你的文档内部自相矛盾。对读者来说，再也没有比在一个设计文档中发现互相冲突的信息更糟糕的了。说明书内容的前后不一致还给出品商发出警报，他会因为你无法把事情理顺而开始怀疑你开发游戏的能力。

很多人喜欢在计算机上阅读设计文档，以便能够轻易地按单词查阅文档，而不必在桌子上翻阅大堆纸张。对于这些人，最好在合适的地方加上超链接。绝大多数流行的文字处理器都能够很方便地创建从文档的某部分引到另一部分的链接，使读者能够快速跳跃到相关的章节。当你试着避免文档中并不绝对必要的重复时，这会相当有帮助。所以，当读者需要回想某个系统功能时，不要去重复它，可以通过一个超链接让他跳到相关的位置上去。

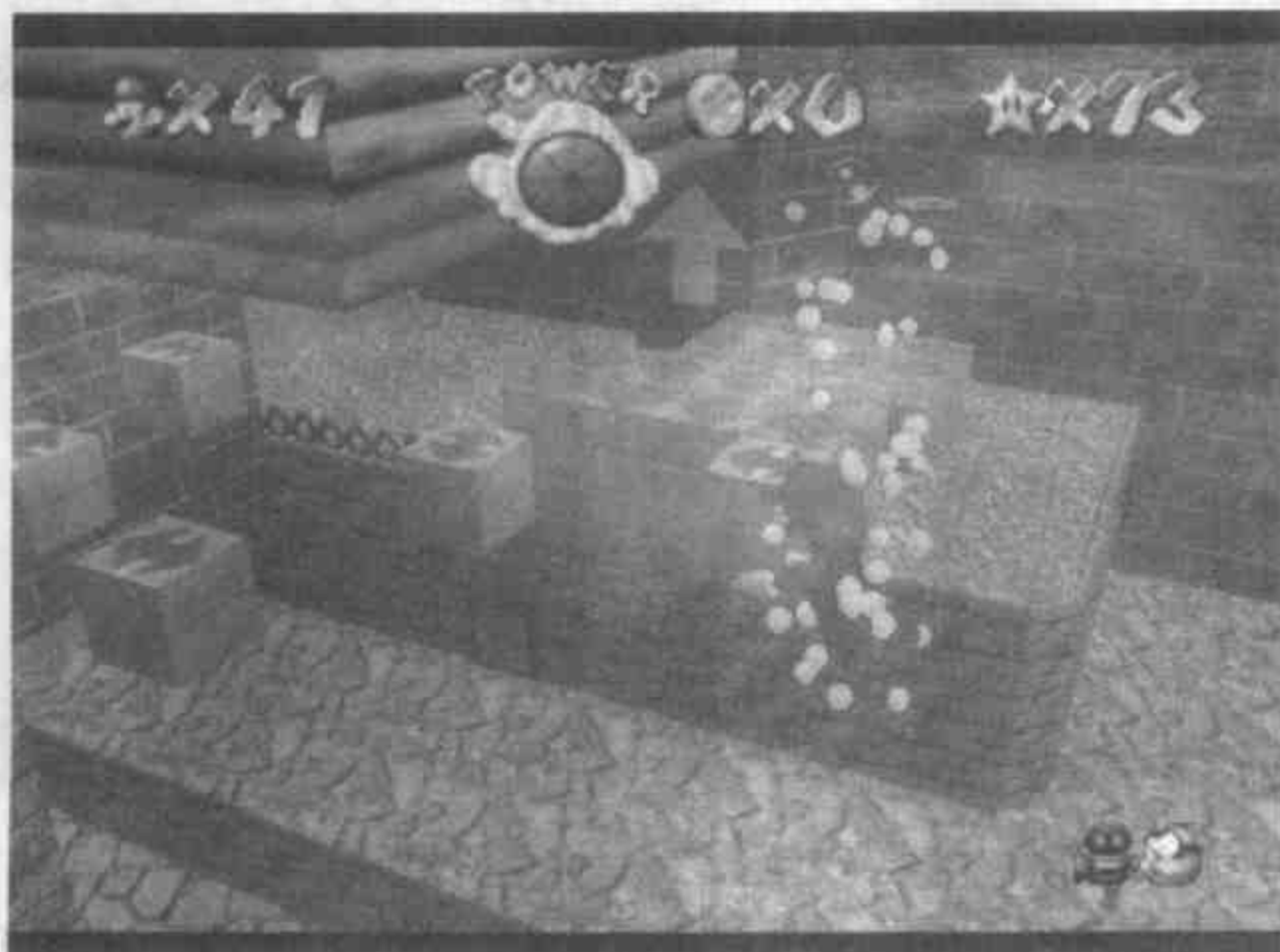


图 17.1 在设计文档中，虽然同其他现存游戏，例如经常提到的 Super Mario 64 的对比可能是合适的，但设计者要注意详细说明他想通过对比说明什么

在写文档时，人们常常想把它写得尽善尽美，但应牢记在心的是，设计文档是为创造一





个富有娱乐性的游戏而做的，它本身并不具备娱乐性。你应当用简洁明确的方式来传达必要的信息。不要花大量时间来考虑文档是否能让人读下去。没有人在读大部头的设计文档时是来寻找刺激的，他们找的是有价值的信息。我总是努力把概述和故事简介写得最有声有色，因为人们往往会坐下来津津有味地通读这些部分。至于文档的其余部分，只要你能够把必要的信息都包括在内就算成功了。花大量的时间使用华丽的文字来润色文档，丝毫不能改变你的游戏。同样地，尽管应当尽可能做到文理通顺，但也不用花太多功夫去反复检查文档，改正语法错误。文档的读者或是你团队的成员，只要能得到所需的信息，他们就会满足的。他们才不会关心你的动名词用得正确与否呢。

在写作设计文档时，你会不由自主地想把设计中的某些元素与其他游戏进行比较，当然，很可能是那些读者玩过的游戏。虽然在第5章里我劝阻你们不要过于关注这种对比，但实际上，在设计文档中进行对比还是有用的，只是要注意：即使它“和 Super Mario64 中所用的游戏机制一模一样”，你还是必须详细地描述你的系统。通过与流行的游戏进行比较，可以给读者一个开始了解你所描述的复杂的游戏系统的出发点。如果他们能回忆起那个游戏，他们立刻就会对你要说什么有了一个大致的印象。当然，仍然需要对你设计中的那个部分进行详尽的叙述，以避免混淆。因此，千万不要依赖对比，不要把它当做一支拐杖，靠它去逃避记录游戏时遇到的麻烦。不管怎么说，用对比作为开始，读者能够更好地理解以后将要详细描绘的系统。



17.2 单元划分

我写游戏设计文档时，一般会把它们划分成下面几个主要章节。在每一部分里，还可以做进一步地细分，但不是每个游戏都必须包括下面所有这些章节。

- 目录表
- 概述/简介
- 游戏机制
- 人工智能
- 游戏元素
- 故事简介
- 游戏进程
- 系统菜单



17.2.1 目录表

把目录表列为文档的一个重要组成部分，读者可能会觉得很好笑。一个长达 50 页并包含很多章节的文档当然要有个目录表。然而，我这里要强调的是目录表的性质。因为给一个设计文档那么长的文本加上索引是非常花费时间的任务，你很可能没有时间完成它。在没有索引表的情形下，目录表就成为人们用以查阅设计文档的最佳手段。当你所在开发团队的成员要在你的文档中查找某条信息时，会先通过目录表来查找那条信息所在的位置。所以，目录表越详尽，





越能迅速地找到所要的信息。

在设计文档中不要使用小说风格的那种简单目录表——也就是说，不要仅仅列出各主要单元，让读者自己到每个单元里去自行查找。目录表必须包含章、节、小节、甚至还有小节的进一步分节。我们已经讨论过为什么要在文档中使用粗体字标题以让它更易查阅。另外，所有的商用文字处理器都允许你把这些标题转成目录里的条目。当在文档中移动这些标题时，相应的条目也会自动更新。大多数文字处理器甚至允许人们在电脑上读文档时，只要点击目录表上的条目，就可以直接读到文档相应部分的内容。为设计文档生成一个详尽的目录表是非常重要的。



17.2.2 概述/简介或操作概要

在文档的开头做一个一页纸的游戏设计概述是个不错的主意。对于那些正在开发这个项目的人来说，游戏设计概述没什么太大用处，如你所知，他们是文档的目标读者。然而，对于那些新加入这个项目的团队成员，概述可以成为理解这个游戏的良好出发点。事实上，对于任何一个第一次读文档的人，不管他们是出品人、行政人员，还是市场营销人员，通过一页纸的概述来获得对游戏的总体印象都是很有帮助的。即使读了概述后没有时间再读文档的其他部分，这页概述也能让他了解游戏的性质。

概述应该严格地限制在一页纸内，太长就再不是有效的简介了。一页纸无法容下的信息可能是游戏设计的核心部分。如果超过了这个长度，在概述中找出最不重要的信息并删掉它。重复这个步骤，直到概述精简到一页纸以内。可以把这个概述想像成你的简历：超过一页就会失去读者。第一段要写得引人入胜，总括整个游戏，以后的段落则要依照开始时设定的轮廓来写。

在动笔写作之前，要确定游戏的焦点，像我在第 5 章“焦点”里所探讨的那样。这个焦点是简介的极好的开端。还记得吗，焦点就是把游戏最吸引人的地方浓缩到一段文字中。以焦点作为概述的开头，然后在下面的段落里再详细描述游戏中每个引人入胜的部分。

概述的主体文字中应当有一段是用来简述游戏故事情节的，如果有故事情节的话。在这一段中，要重点描写玩家在游戏中所将经历的冒险，而不要过多着地描写游戏环境的故事背景或历史背景。通过对故事的简介描述游戏，还要提及玩家将畅游的不同类型的环境、他们将遇到的角色。随时铭记这只是一个概述，所以不要深入刻画。只需触及故事的要点，然后就可以转向下一段了。

简介的另一主体段落要用来讨论游戏操控的不同方面，通过主要方面来突出重点。哪些特点是这个游戏的中心环节，最有助于吸引玩家一小时一小时地玩它？当然，不要在所有游戏都有的那些特点（“X 项目能够在任何时候储存玩家的游戏进度！”）上浪费笔墨，而应着重于该游戏与众不同的地方，那些能让游戏独一无二并引人入胜的特点。

然后，应当对整个概述做个总结，要特别强调为什么这个游戏能够吸引玩家，这个游戏有哪些东西是其他游戏所没有的。读者读完这一页纸时应当心情振奋，对这个项目充满热情。就把这页概述想像成誓师宣言，鼓舞士气吧，应当让人们不必读完整个文档就对这个项目热血沸腾。



17.2.3 游戏机制

游戏机制部分是文档最重要的部分。它也可被称为“游戏操控”部分，因为它描述在游戏





中玩家可以做什么，以及游戏如何运行。通过描述玩家能进行哪些活动，游戏机制部分定义了游戏自身。因而，游戏机制部分是设计文档中最难写的部分。描述游戏操控是一个极其困难的命题，所以很多差劲的游戏设计文档干脆略过这一部分，而宁可聚焦于故事、视效或菜单系统，因为所有这些都是相对容易写的主题。常言道：“描绘音乐就像在建筑物周围的舞蹈一样困难。”描绘游戏操控也一样困难重重，然而为了使设计文档能有助于团队创建游戏，这是必须完成的工作。

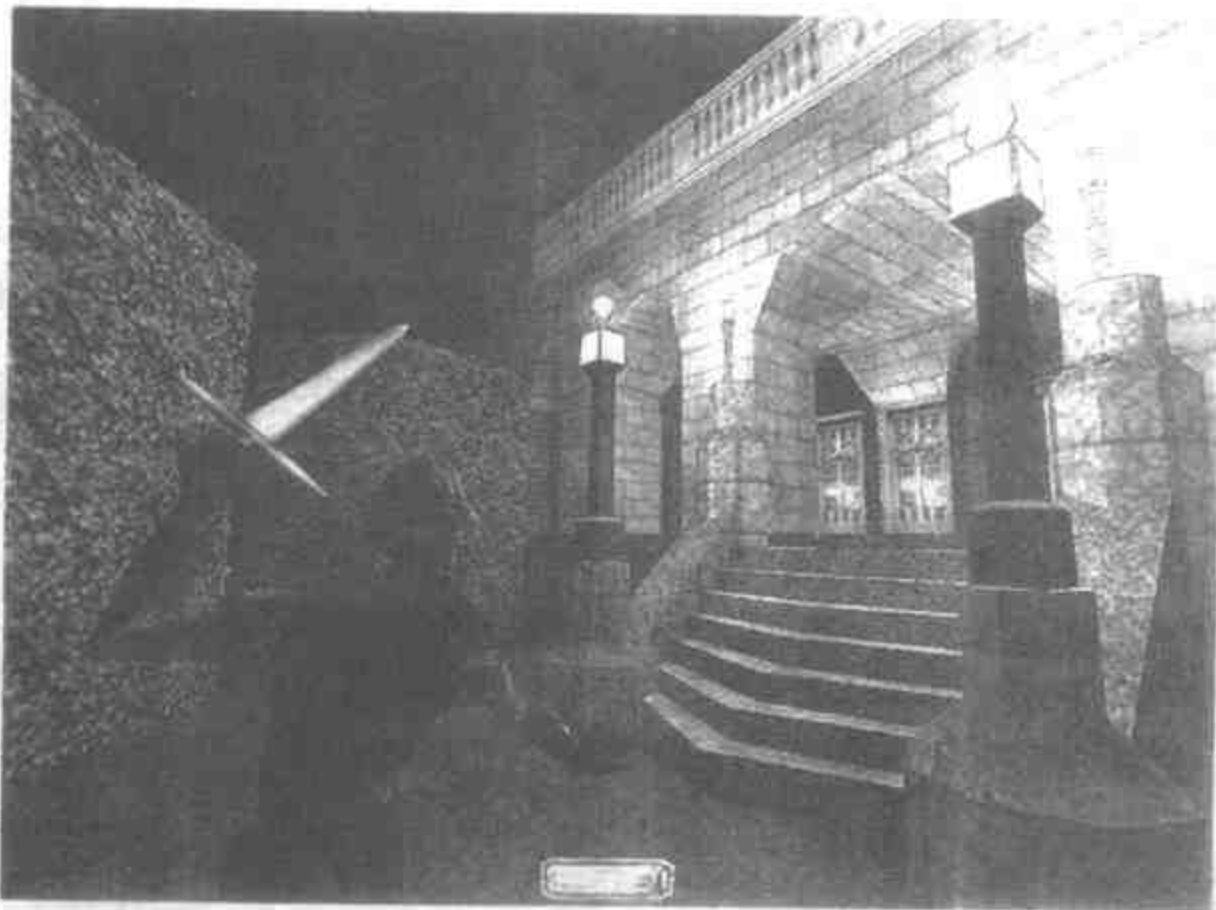


图 17.2 系列作品，例如 Thief 和 Thief II，在其设计文档中经常能够使用相同或极其相似的游戏机制部分。图为 Thief II

除了必须提及玩家角色外，游戏机制部分要避免详细描述游戏玩境中的物品或角色。这些描述可以留给文档后面的相关章节。例如，你可能想描写玩家能拿到的不同武器的各种效果，以及玩家如何能够控制这些武器，但你应把这些内容放到后面游戏环境中不同武器的实际清单中去。特定武器可作为例子用来说明在游戏机制部分所描写的功能。你可以这样理解这个问题：在游戏机制部分描写的东西可以用在不同游戏上。例如，Thief 系列游戏设计文档中的游戏机制部分差不多都是一样的。从 Thief 到 Thief II 只有武器、物件、难度级别和敌人发生了变化。游戏的核心还是一样的，而记载在游戏机制部分的正在于核心。

按照先后次序，以玩家首次玩这个游戏的经历来介绍游戏的各种能力不失为一个好办法。例如，从简入深来写。玩家最基本的动作会是什么？假如你在开发这样一个游戏，由玩家控制一个游戏环境中的替代品（可能是另一个人、一个宇宙飞船、一个飞机、一个机器人或其他任意一个想像的东西）。你大概可以从这个角色如何前进或后退、左转或右转等基本动作开始。在介绍了这些最简单的动作后，就可以适当地讲一些更复杂的了，比如跳跃、蹲伏、滚动等。如果你的游戏是 RTS 或 Diablo 风格的 RPG 游戏，玩家使用点击法来移动，你就需要精确地描述它如何运动。玩家的寻路术需要有多好？当替代品无法到达玩家点击的地方时，游戏会如何发展？是用不同按键选择角色来移动它，还是使用单键控制？



当描述角色的动作时，需要列出玩家完成这些动作所需的物理指令。例如，“前进，玩家需要按住前进键不放。如果玩家只是轻敲前进键，他的角色将只移动一点。”最好在你的文档中按玩家的操控来称呼不同的键或按钮，而不要提到它们特定的名称；用“前进键”而不是“向上的箭头”或“蓝色 X 按钮”。这样能使控制方法与系统平台无关，当以后要改变这些键时，不必把设计文档中大量的“向上的箭头”都换掉。实现操控系统的程序员不见得会关心分配给某一指令的按键的字面名称，但她需要知道用户将用到多少不同的命令，以及这些命令在游戏环境内和什么样的动作联系在一起。

一旦描述好玩家如何指挥游戏，下一个必然的步骤就是说明游戏的运动模式。它是遵循现实世界中的物理模式还是某种简化的模式？它是逐渐加速还是在瞬间达到最终速度？它上坡时比在平面上走得慢吗？它的反应是像 *Quake* 那样灵敏还是像 *Tomb Raider* 那样缓慢而精确呢？当它撞上了别的物体会怎样——滑落、转向还是仅仅停下来？这类细节都要深入思考并详细说明。

也许，移动游戏部件或玩家替代品并不是游戏的关键操作。那么设想一下玩家开始这个游戏时会先做什么，然后把它写出来。比如你在描述 *Railroad Tycoon*，要谈玩家铺设轨道的方法和管理它们的规则。如果你在给旅鼠游戏写设计文档，应当说明玩家如何把一个一般旅鼠转变成一个特殊旅鼠，比如瓦匠或挖掘者。如果你在描述 *SimCity*，应当解释玩家如何划定区域。

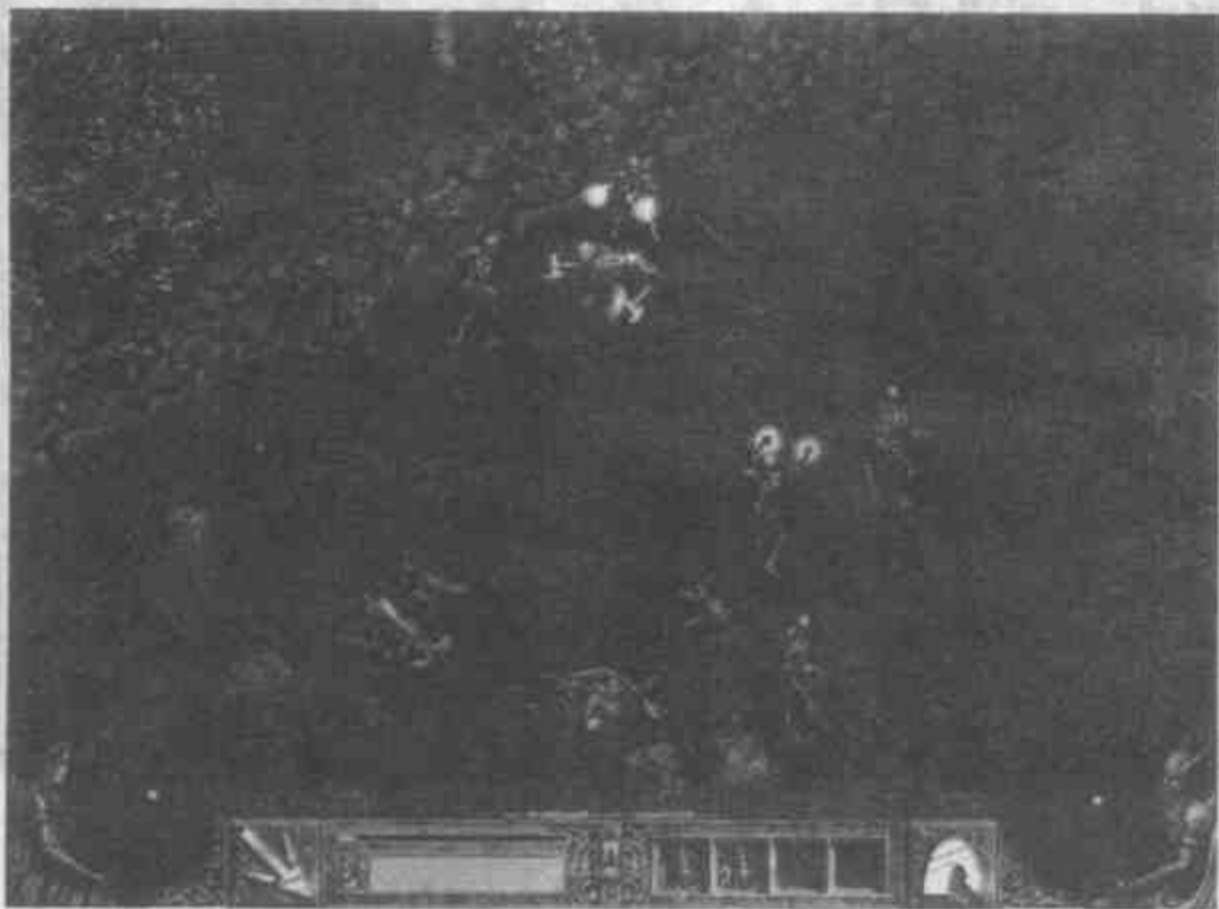


图 17.3 *Diablo II* 这样的 RPG 游戏通常以玩家创建她的角色开始这自然要在设计文档中进行详尽的说明

如果你的游戏始于玩家创建她的角色，像 *Diablo* 风格的角色扮演游戏所做的那样，你需要描述这个过程，概括玩家所选择的每一个属性值的意义。“力量”和“灵敏度”代表什么？这样在游戏机制部分的后面，当你要描述某一属性值影响下的一个动作时，能提醒读者回头去找这个属性值的原始定义。

在从最基本的地方开始文档后，可以深入叙述玩家能做的更复杂的操作，这时要注意文档





的逻辑顺序，每一种后续的动作都要建立在它前一动作的基础上。要让各个游戏机制环环相扣，让读者能一目了然游戏的间架结构。当然，还要记得不要引用没有定义或没有详细讲述过的机制。

当然，文档涵盖哪些主题要视所创建游戏的类型而定。如果你的游戏涉及战役，就要详尽系统阐述它，解释玩家如何使用不同的武器，以及这些武器在游戏环境中会有什么样的作用。如果玩家替代者能够拿起或操纵物品，就要详尽解释怎么拿起它们、怎么能得到它们、物品清单如何管理以及类似问题。

游戏机制部分也是说明玩家在游戏环境中会遇上什么谜题的合适场所。实际上，如果是智力游戏的话，这一部分需要占很大的比重。需要描述谜题的性能、玩家如何操纵它们，需指出谜题是如何建立的，但不必列出具体谜题。就像对待具体武器一样，谜题的列表也应当放到文档后面的内容中。举个例子，假如你要说明 *Prince of Persia* 原始版本中的谜题，应当解释谜题可能牵涉到压力板、隐蔽的活动天花板、陷落地板块、可以通过压力板起降的门、从地面或墙壁上伸出的长钉、仙丹、某些类型的魔法效果，以及其他所有游戏环境所允许的成分。不必细致地列出这些成分的具体内容，它们应当放到关卡中。把它们留给文档后面的关卡部分好了，或者干脆让关卡设计者自己去推想。在这里，应当列出的是组成这些物品的调色板，以及谜题创建的方式。

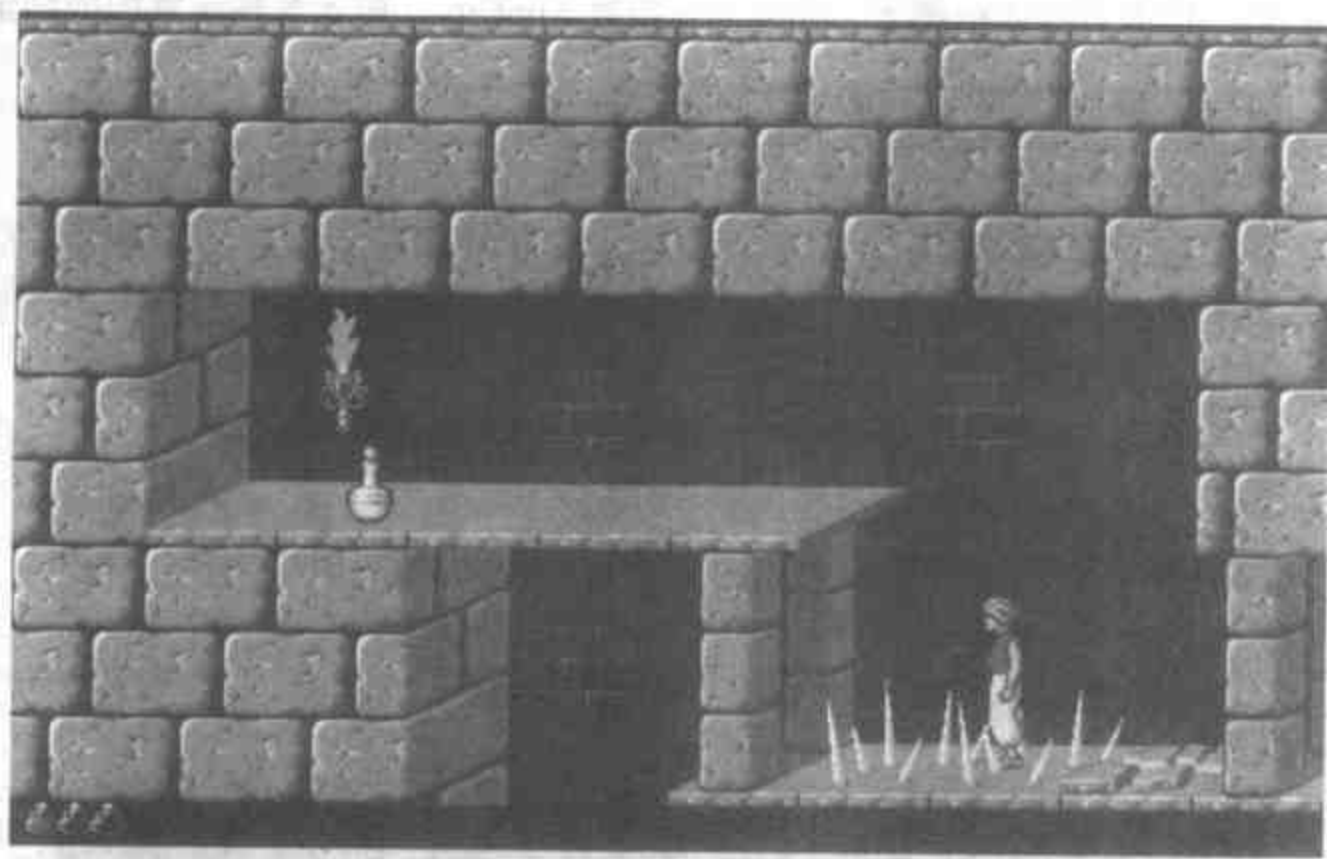


图 17.4 游戏机制部分很适合描述像 *Prince of Persia* 这类游戏中见到的不同的谜题成分

如果游戏中玩家为了完成不同的任务要在各种模式间转换，那么每种这样的模式都应当进行详细论述。例如，玩家在 *Drakan* 中扮演 Rynn 时，使用前进键或后退键在游戏环境中行进，用鼠标转换角色。然而，当玩家按下物品清单键时，游戏转入物品清单模式。在这个模式里，玩家不再控制 Rynn 的运动，而是利用鼠标的标准拖放功能处理 Rynn 的物品。在 *Drakan* 的设计文档中，设计者需要清楚地说明玩家的控制如何从一种模式转换成为另一种模式，以及在每种模式中游戏环境如何控制。

设计文档中的某些部分应根据游戏所使用的技术而取舍，是 2D 还是 3D，室内还是室外，





实时的还是预渲染的。虽然人们尽力把游戏技术方面从设计文档中分离出来放到技术设计文档中，但我们所创建的毕竟是计算机游戏，因此它与所使用的技术是紧密相连的。写一个设计文档而不去理会用什么样的技术实现它通常是不可能的，至少也是不切实际的。不一定要知道引擎每秒处理多少个多边形，或它支持曲面的非均匀样条技术与否。但是需对设计者能用到的工具有个基本的了解。设计 3D 环境下还是 2D 环境下的动作控制或战役系统完全是两回事。应当根据游戏所使用的技术扬长避短。

例如，游戏机制部分需要说明玩家玩游戏时所见到的景象。这包括玩家如何看这个环境、使用什么样的镜头角度、玩家如何影响镜头的位置。要写这些，需要知道摄像机能做什么，而这完全依赖于游戏的引擎。引擎可能只支持第一人称视角，侧面视角，或有其他形式的限制。然而，玩家如何看游戏环境是游戏设计的中心环节，必须在游戏机制部分进行讨论。

游戏内图形用户界面 (GUI) 对游戏至关重要，因此，应当在游戏机制部分做细致地描述。需写出所有用于游戏环境描述的数据，比如，在一个动作游戏里，游戏操控过程中所用到的玩家健康或其他属性值。GUI 部分要包含游戏中所有 GUI 内容，比如玩家代表与人交谈时或玩家管理物品时所看见的东西。图形界面的描述在 Alpha Centauri 或 The Sims 这类游戏中尤为重要，因为这类游戏包含很多不同的 GUI 并且玩家不停地使用 GUI 来玩游戏。对这些 GUI 的描述可以统一在游戏机制部分的一段文字中，也可以分散在系统描述的章节中，在每个相关之处进行详细地描绘。记住，要努力使设计文档更为读者着想。如果是一个艺术指导来查找将要创建的各个 GUI，而它们却分散在整个游戏机制部分，他可能会漏掉一些。另一方面，一个程序员可能更想在对某个系统的描述中找到相应的 GUI。必须确定哪一方对你的文档和项目更感兴趣。在游戏机制部分，只需要描写用在游戏中与游戏操控有关的 GUI。玩家开始新游戏或装载旧游戏时用的前端 GUI 都不能放入游戏操控部分。所以，前端 GUI 应该放到系统菜单部分，我在本章后面再讨论它。

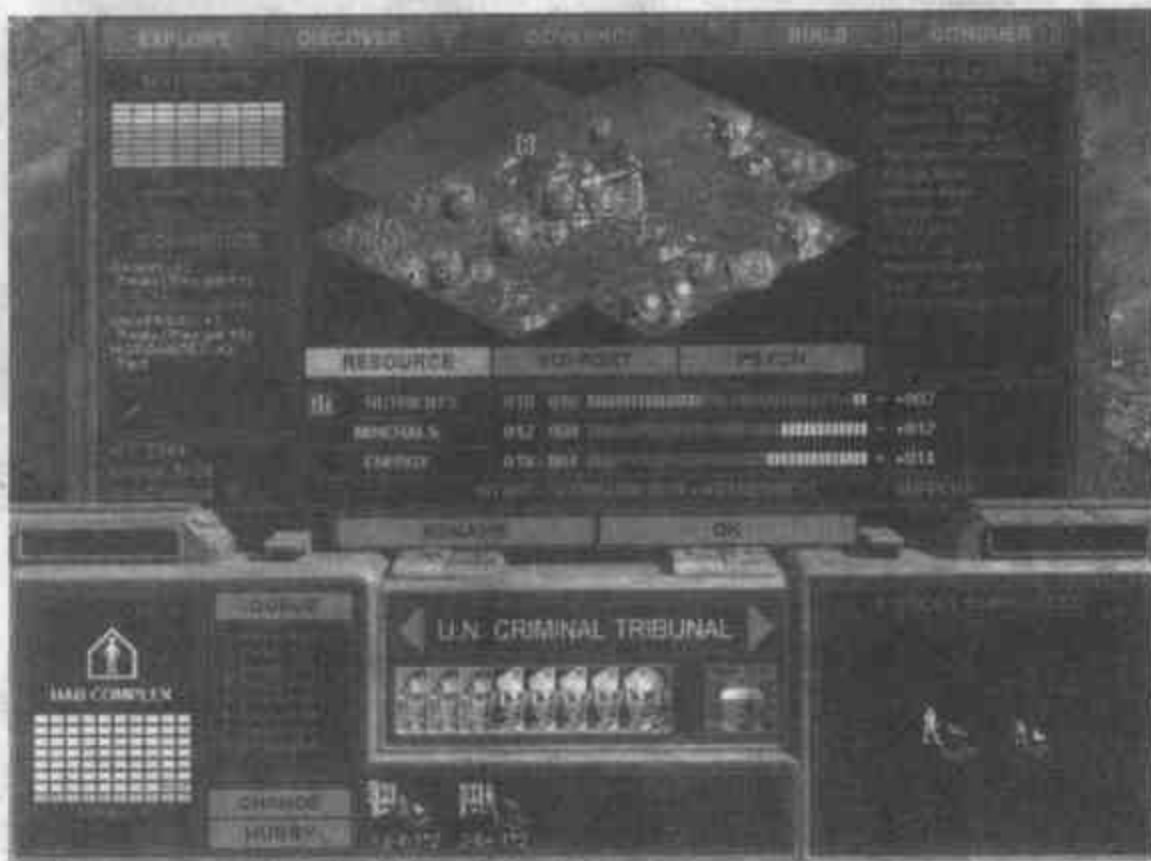


图 17.5 在 Alpha Centauri 这样的游戏中，图形界面的各种元素极为重要，需要在设计文档中进行详细地描述





在写作游戏机制部分时，很容易做一些假设，但一个好的设计者应当避免把任何事情想象成理所当然的事。比方说一个设计者正在写一个 Quake 模式的第一人称射击游戏。他可能会设想当玩家跑过一个物体时，她的角色会自动拾起它。设计者可能玩过太多的第一人称射击游戏，因此对他来说让系统这么工作是显而易见的。但如果他没有把它写在文档中，程序小组则可能会根据自己喜欢的游戏，采用另一种工作方式。不要想当然地认为对你来说很明显的游戏成分对读文档的人也同样显而易见。要把所有的东西都清楚明确地写下来，以防止混淆。

可以把游戏机制部分写成是整个手册中最详细的地方。你要极为深入细致地描写玩家如何在游戏世界中完成不同的动作——玩家会使用哪些命令以及这些命令将产生什么样的结果。如果你像新闻记者写新故事一样来写你的设计文档，那么在游戏机制部分该关注的是“什么”和“怎样”——玩家在游戏中会做什么事情和怎样做。之后，在文档中，还要弄清“何地”、“何时”以及“为什么”。

17.2.4 人工智能

如果说游戏机制部分描写玩家如何与游戏环境互动的話，那么人工智能部分则记录这个环境对玩家的行动如何反应。玩家在游戏环境中所面对的对手将如何行动？在某种情况下它们将做什么？这一部分也要描写当玩家什么也不做时，游戏环境如何运转。例如，它可能要讨论诸如居民如何进行日常活动这类周边行为。

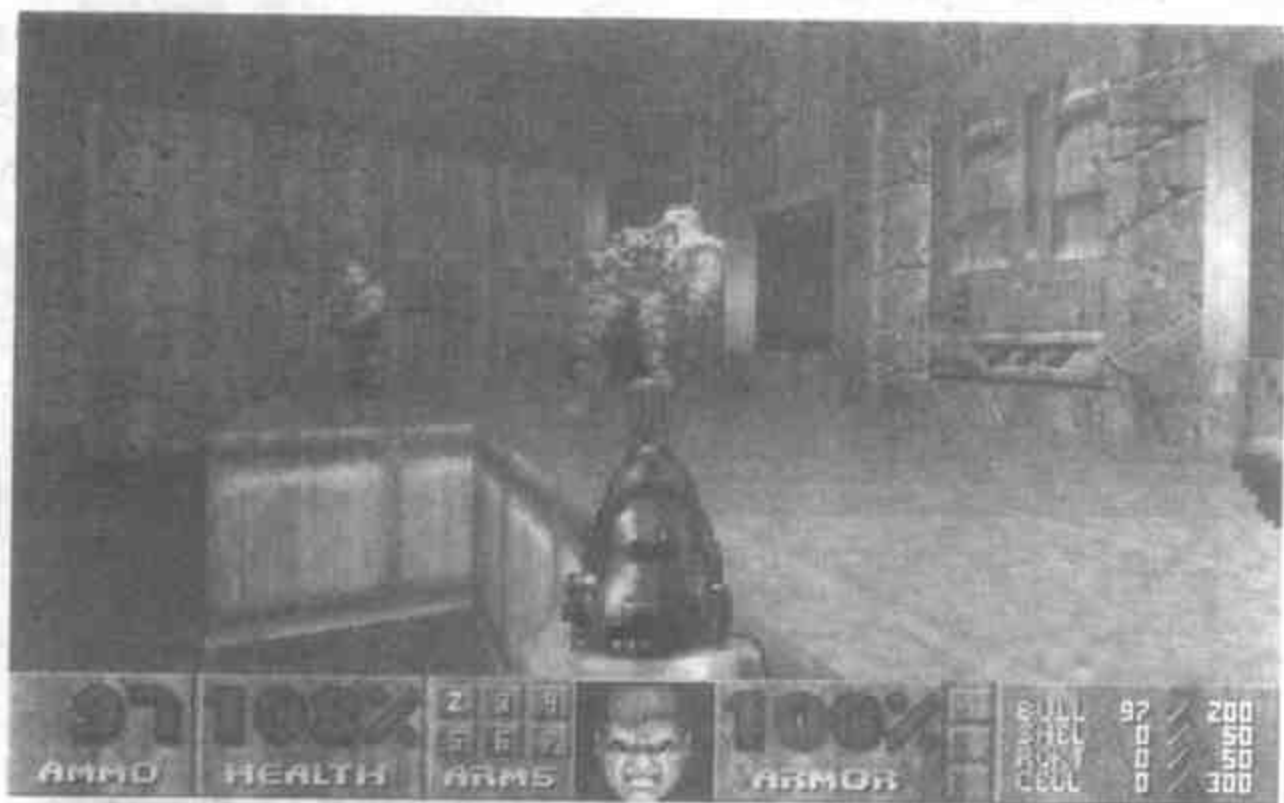


图 17.6 在 Doom II 这类游戏中，玩家机制和 AI 主体的行为互不关联，可以在设计文档的不同部分进行描述

一些设计文档的作者倾向于把人工智能部分包含在游戏机制单元中，但只要可能，我宁可把它们分开。是否把人工智能部分放在游戏机制单元中应取决于游戏的性质。对于像旅鼠那样、玩家的控制与 AI 紧密地交织在一起的游戏来说，作者在设计文档的同一单元内讨论它们相当明智。但对 Doom 这类游戏来说，玩家对其游戏环境替代品太空舰队的操纵与敌人的行为相对分离，因此应当把它们划分成两个部分。这种划分使程序员能更方便地查看文档，因为玩家行动



的开发过程和创建与之敌对的生物通常是相互分离的。

在 AI 单元中，应该尽可能详尽地说明期望你的游戏如何对待玩家。如果正在开发一个玩家在游戏环境中行走时会遇见各种其他角色的游戏，则一定要描写那些角色如何行动。玩家不和他们谈话他们就不理睬她吗？还是他们会攻击玩家？他们是有一定的智能可在某个区域搜索道路，还是沿着预先规定的路线走？有些 NPC 可能会向玩家发起一场战斗，那么何时且为什么他们决定这么做？是因为他们看到了角色？听到她？还是他们被关卡设计者设定的触发器激活？或者视不同的情况分别是这三者？这些角色有多聪明？他们能躲在角落里，在安全的地方伏击玩家吗？他们受伤后会逃走吗？在 AI 单元要回答很多问题，让 AI 的程序设计者明确知道他们需要执行什么。回答的问题越多，程序员们就越有可能创造出符合你想象的游戏。

给战略游戏设计 AI 文档意义更加重大。假设你在开发一个像 WarCraft 那样的实时游戏或是像 Civilization 那样的回合制的战略游戏。敌方会采用哪些战略来压倒玩家的部队呢？这些队伍如何协作呢？如果可能的话，计算机玩家何时决定创建更多的部队，又会创建多少呢？AI 能否识别并抵抗玩家不同类型的进攻，例如说翼侧包围部署呢？敌人的 AI 比得上玩家的吗，或者是否因为电脑有更多强大的装备而与玩家打个平手？如果有必要的话，可以提供某个游戏的概要介绍，以及敌方 AI 在游戏的不同关头如何行动。



图 17.7 在 WarCraft 这类战略游戏的设计文档中，描述 AI 将要用到的协作战术是非常重要的

人工智能部分的文档编号最好取得团队里程序员们的帮助。找出他们做过的 AI 类型，看看它们能否应用到你的项目中。找出哪些难以完成而哪些比较容易。通常，设计者（尤其是那些非程序员）很难理解，让 AI 主体受伤后逃走是一项微不足道的工作，而让它找路上楼梯或跳过障碍却非常困难。游戏中 AI 的能力不应建立在空中楼阁的想象上，而应去追求真正的、可以实现的目标。记住，当程序员读到一个充满凭空想象且难以置信的 AI 描述的设计文档时，他很可





能被此激怒，从此很难再认真地对待文档。所以在写游戏 AI 部分的文档时让程序员同你一同工作，会使文档的这一部分变得更强，同时也能保证 AI 程序员真正理解程序主体应该是什么样的。

在编写人工智能单元文档时，要尽力遵循与游戏机制部分相同的法则。不要提及游戏中具体的 NPC，而要描写不同主体将会展现的一般行为方式。应该把具体的 NPC 和他们的行为方式放到文档后面的游戏元素单元。另外，不要去假想任何事情。游戏中主体的行为方式写得越细越好，即使它看上去对你很明显。



17.2.5 游戏元素：角色、物品和对象/机制

如果把小组里的关卡设计者看成是画家的话，那么游戏元素就是他们调色板里的颜料。这些元素在游戏中组合起来，给玩家带来一关又一关引人入胜的经历。设计者们要创造它们，并以独特有趣的方式把它们捏合起来，创建各种各样的关卡让玩家在游戏中流连忘返。当然，并不是每个游戏都有关卡的，但几乎每个游戏都有游戏元素。不管这些游戏元素是玩家在 Robotron 2084 中遭遇到的敌人、他们在 SimCity 中所能建设的各种建筑，还是 Tetris 中的不同积木，游戏元素都应在游戏元素部分详细地列出。

在介绍游戏中的游戏机制和人工智能方面已经花费了大量的笔墨，现在可以转向具体内容了。记得你在游戏机制部分和 AI 部分写得相当的具有通用性，以至于它们可以为不同的游戏所使用。这些部分甚至可以不修改就用到续集中去，如果有续集的话。但敌手、NPC、物品及玩家在游戏环境中碰到的各种机制则可能是游戏所独有的。这些内容通常同故事情节关系密切，这些故事情节还要在以后的故事简介单元和游戏进程单元中深入探讨。在故事简介单元之前还是在其之后来列出角色、物品和对象，实际上很难定夺。这要由你来决定，如何做对你的文档和游戏更有好处。

我通常把游戏元素分为三类：角色、物品、对象/机制。你可能希望在你的设计文档中为每类元素单独写一个章节，或者在一个统一的游戏元素部分分列三个小标题。

- **角色**：角色包括玩家要打的所有敌人、他会碰上并可能交谈的所有人物，还有游戏中所有不同的 AI 主体。可以说，角色包括游戏中所有活动着的、非玩家操控的元素。
- **物品**：物品包括任何玩家能够拾起、使用或用某种方式操纵的东西。玩家可能用到的武器当然应当列在其中，同样还有任何可以放到玩家物品清单里的东西，比如盔甲、钥匙或丹药等。
- **对象/机制**：第三类包括出现在游戏中的各种实体，它们不是 AI 驱动的，玩家不能拾起它们但能以某种方式操纵它们。这包括门、开关、谜题或其他能在游戏过程中操纵的东西。

另外，根据游戏的类型，不一定每一类元素都需要。像 Half-Life 那样的射击类游戏，可能三种元素都需要：角色有玩家要打的外星人，他发现的武器要列到物品中，还有玩家在游戏环境中遇到的各种机制要放到第三类中，比如可重定向的激光束。而像 StarCraft 那样的实时游戏可能只要一个清单（事实上是角色和物品的统一体）列出玩家或敌方所能控制的不同部队，再加上一个对象/机制列表列出玩家能作用的任何对象，例如门道或传送点。然而，如果你设计的 RTS 中部队能够拾起物体，那么最后还是要设立第三个类别。像 Diablo 那样的 RPG 可能要加上





第四组或第五组游戏元素分别来列玩家的技能或符咒，因为它们不属于前面说到的那三类游戏元素。尽量把游戏元素，不管是什么，符合逻辑地分组。根据游戏的特点，仅有一组或多达十组都没有问题，都可以用来创建吸引人的游戏。

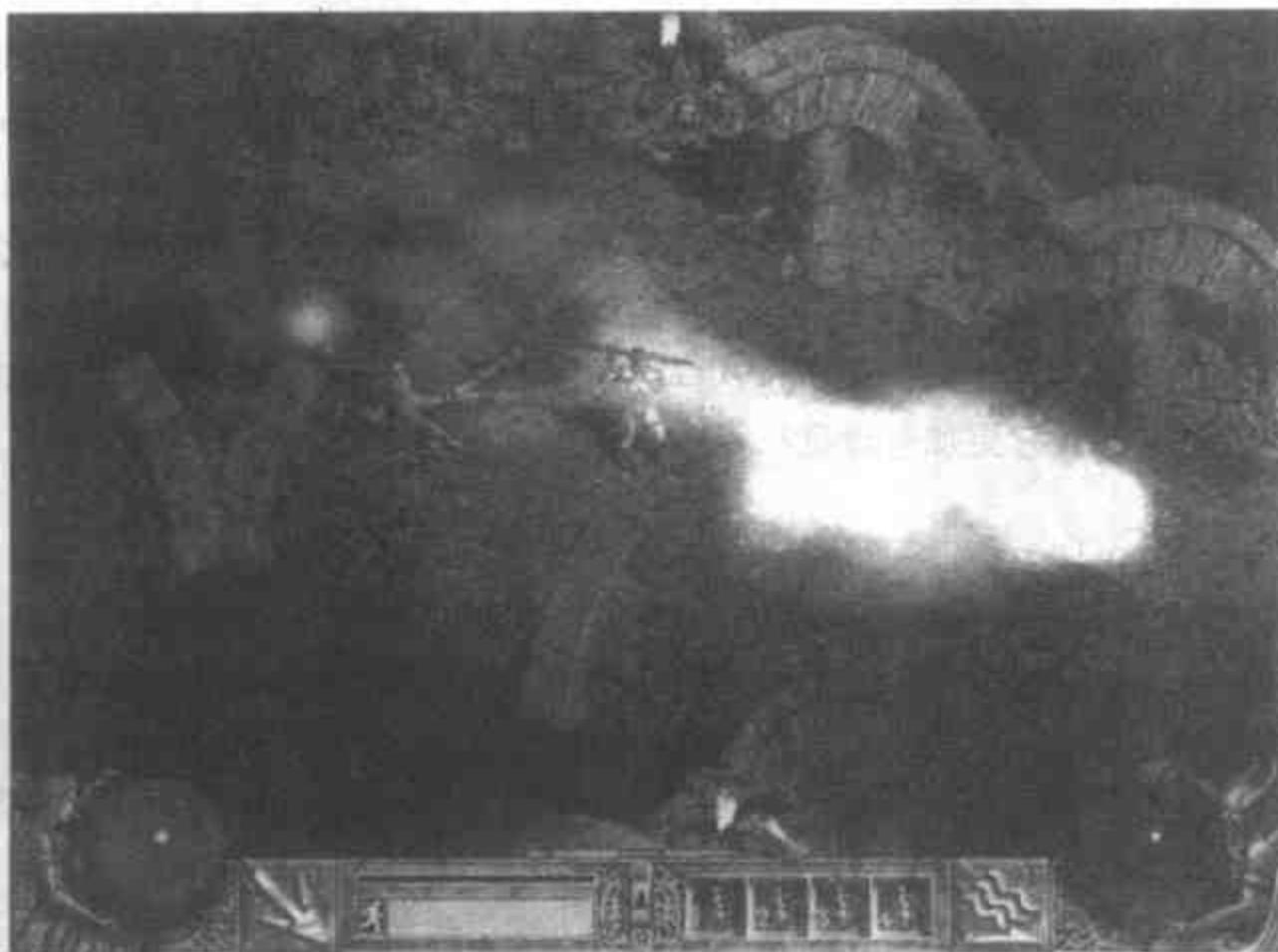


图 17.8 Diablo 的设计文档可能包含专门的部分来描写玩家的符咒和技能

在每一个类别里，都要尽力按照逻辑顺序来排列物品或把它们划分为不同的小组。比如，你在设计一个 RPG，你应该把所有的药品放在一起，所有短兵相接的武器放在一起，而所有远程武器放在另一组。RTS 可以把部队分成进攻型、防御型或建设型，或者分成静止的和机动的。再说一次，先看看你所做游戏的类型，然后尽可能用最适合你数据的方式分组，使它能让读者易于查找和理解。游戏元素部分既要给艺术小组也要给程序组提供信息。艺术小组需要确定所描述的所有游戏元素的艺术构想，而程序小组则希望把游戏元素与游戏机制及 AI 部分结合起来，以期全面了解游戏该怎么做（当然，理想的情况是，如果游戏机制和 AI 部分写得很严谨的话，程序小组根本不需要看游戏元素单元）。在给游戏的角色、物品及机制分类时，要同时考虑到画家和程序员们，以及游戏可能需要的分类。

在列举和描述这些游戏元素时，要避免给它们分配实际的统计值。在得到一个可运行的游戏以检测 AI 行为或武器并适当地平衡它们以前，无法预测有关物品和敌人等细节。在作品开发前提出这些统计值，没有机会去平衡与修正，不仅浪费你的时间，同时也浪费阅读者的时间。

应当尽量通过回答问题并与其他元素相联系的方法来叙述游戏元素。它们互相比难度如何？某一个 AI 主体有哪些特点？这种元素用到何种 AI 功能，要取得什么效果？玩家应该看到什么样的总体效果及各种具体效果？它与其他物品相比有多大？要提供足够的信息，以便程序员知道怎样为实体编码，同时要有详细的叙述以便画家能够画出概念草图。尽可能提供有用的细节，但也要适可而止。当你只是为了记载而记载时，读者们，无论是画家、程序员还是其他的设计者都会看出来的，那时你的文档就不再实用了。不要写太多繁杂琐碎的东西浪费时间。





17.2.6 故事简介

虽然不一定非得有，我想一个简短的故事简介对设计文档大有好处，假设你的游戏有故事情节的话。好的简介通俗易懂地叙述游戏中所发生的一切。就像设计文档的简介一样，故事简介也能让小组成员迅速了解故事梗概。要实现这个目的，需把简介限制在很短的篇幅内同时还要涵盖故事的所有要点。两页纸应该够了，但这也要视故事的复杂程度而定；射击游戏可能只要一页纸，而角色扮演游戏则要多几页。

当然，不用叙述游戏所有的分枝或玩家与其他角色的每一次交谈。尽可能让故事简介生动有趣，让人们愿意读它。游戏机制部分由于它的条条框框和太多的细节而难读，而读故事简介则应是令人高兴的事。事实上，如果不是这样，就要去找出原因。是因为故事不够吸引人吗？是否该精简并改进它，让它更加有趣呢？

17.2.7 游戏进程

因为游戏的本性，游戏进程部分很可能是设计文档中最长的部分。在这儿，游戏设计者把游戏分解为玩家经历的各种事件，并叙述它们如何发展变化。这一单元指导艺术小组和关卡设计组创建游戏中各种类型的环境。关卡设计者们依照本章来设计各个关卡所包含的细节，把游戏的各个方面组合在一起。

对于很多种游戏，包括 RPG、RTS 游戏、第一人称射击游戏、动作/冒险游戏，还有任务制飞行模拟游戏，游戏进程最好按关卡来细分。在每一关中，要细节描述玩家将面临的挑战、他们所发生的故事，还有其艺术界面。设想并描绘出每一关的主要挑战：在 A 地点迎战大队敌人，在 B 地点与某一角色相遇并交谈，在 C 地点解出一个游戏谜题。当然，不必把每一关再细分成一次次单独的冲突，并把每分钟发生的小事都列出来。至于角色的统计值，只有在实际做这一关，有办法尝试并检验这些冲突后才能得到。可能的话，应当说明这一关如何表现游戏的故事。何种对象和物品应在何地出现才能使故事顺利进行？还要讨论一下游戏“调色板”中的哪些元素能用在这一关里。玩家沿着这条路会遇到哪种敌人？

最重要的是，尽量写出这一关如何影响玩家，不只是这一关如何困难这一类的话，而是说一玩家会有什么样的游戏经历。玩家在每一关会感到持续不断的斗争和挑战，还是这一关节奏很慢，更着重于探索？故事在这一关达到高潮，强度激增，还是这一关只是聚集于给出游戏背景故事，节奏缓慢？在写游戏进程时，要始终想像玩家在每一关该有什么样的感觉，并把这种感觉在文字中表现出来。

当然，并不是每个游戏都分关卡，所以你的游戏进程可能没法如此自然地划分。但大多数游戏都有阶段的划分。尽量决定游戏怎样划分阶段，并把游戏进程照此划分。例如，原始的街机游戏 Centipede 分成一系列的轮次。在该游戏中，一旦玩家杀掉了蜈蚣的所有节，他就进入了下一轮，这种轮次是循环的，每一个新的轮次都会扔给玩家一条更难、可能更长、行动更快速。另外，每次进入下一轮时，敌人出现的条件也发生变化。比如，跳蚤从来不会在十二节蜈蚣那些轮次出来。如果有人要写 Centipede 的游戏进程（那需要很长），他会把它按轮次来划分，清晰地描绘出上一轮到下一轮游戏是如何变化的。





图 17.9 像 SimCity 系列这样的无定式战略游戏不需要游戏进程单元，因为游戏中发生的一切都完全取决于玩家的抉择和游戏机制

有些游戏可能根本就不需要游戏进程这一单元。比如，为 Civilization 一类的战略游戏或 SimCity 一类软件玩具所写的设计文档，可能在游戏机制、AI 或游戏元素部分就描写了所有相关的游戏控制方式。不管怎么说，这些游戏很少形成关卡，所以也就不需要编号游戏进程文档。然而，如果我们所谈的游戏包含了某些场景，它们在某个特定配置一达到预定水平就开始出现（像 SimCity 那样），那么游戏进程部分是描写这些场景以及它们如何挑战玩家的理想地方。



17.2.8 系统菜单

系统菜单单元是详细说明主菜单和玩家在游戏之外不同情况下面对的其他各种选项屏幕的地方。这些菜单实际上对游戏没多大影响，因此，应当分出来单写。在这里，要描写玩家如何储存和装载游戏；这些菜单给玩家提供了什么样的界面：玩家要用鼠标点击呢，还是要用 Enter 键或箭头键，亦或是二者兼用？此处，文档应写得要尽量完整，保证系统菜单足以指引玩家享受游戏本身的乐趣。开发商喜欢看到详细描述这些菜单的流程，所以包含系统菜单部分很重要，虽然，就我个人来说，一个完整的设计文档并不一定要有它。恐怕把系统菜单另写一个单独的文档更合理些，因为它们和游戏意图没什么关系。



17.3 个人观点

在前面的段落里，我提出了我喜欢使用的游戏设计文档格式。让我重复一遍，这绝不是游戏行业的标准格式。很多好的设计文档所使用的格式都与我所述的文档不同，不管是在结构上





还是在细节上。但如果按照我介绍的结构提交一个设计文档，绝不会被取笑或被当成傻瓜。如我前面所说的，最重要的是把你对游戏的设想传达给阅读文档的人。只要能让你的数据最大限度地清楚、实用，你可以用任何形式来给出你的设计信息。

不同的项目，设计文档的格式不同，部分原因在于游戏还没有形成（我也不认为它们将会成为）一个标准化的艺术形式，像戏剧、电影或交响乐那样。当然，游戏当中也有一些特定的种类，某些种类的设计文档格式，比如第一人称射击游戏，可以被标准化。但即便如此，当射击游戏的形式变化时，例如采用了新的游戏风格与机制，文档的结构也要相应改变以便更有效地传达它们。不能想像一个 Half-Life 类的第一人称射击游戏的文档能和 Alpha Centauri 那样的战略游戏的文档使用一样的形式。这两种游戏的成就和玩家的经历根本不同，因此它们的设计文档必定不同。



17.4 糟糕的设计文档

我以前说到过，尽量去找一些专家的游戏设计文档来让你对这个行业的常规做法有所了解。这可能很有用。然而一定要小心，很可能你得到的文档没什么优点。很多发布过的游戏的设计文档或有经验的专家所写的文档都很可怕。为了用反面典型让你学会该避免做什么，我将拿出几个不同类型的整脚的设计文档，说说它们为什么如此失败。



17.4.1 薄本或省略符号文档

这些小薄本决不会超过 30 页，令人吃惊的是，作为经验丰富的游戏设计者，他们完全没有写入有用的信息。他们使用“游戏将很有趣”、“反应会很激烈”这类毫无意义的描写。在这些文档中，进行了很多与其他游戏的对比：“这里像 Super Mario 64”或“这个游戏有着与 Quake 近似的控制设计”。像我以前说过的，这些类比几乎没什么用处，因为 Wafer-Thin 文档的作者一直没能继续说清楚 Super Mario 64 或者 Quake 的控制设计是什么，所以这个游戏所用的机制也就没有人知道了。

这些文档经常花费大量的时间，有可能是一半的篇幅，来讨论游戏的背景故事。而这种背景故事往往很单调很乏味，和游戏的发展关系也不太大。薄本文档同样花了大量时间谈论菜单如何工作，不是游戏内部菜单，而是玩家选择玩哪种类型游戏、设定选项等等的系统菜单。很多模型建立起来，各选项被详细列出，而各种选择会如何影响游戏却很少具体描写，因为对游戏本身就没有进行很好的定义。最好在游戏已经开始开发了再去构思系统菜单，因为那时设计者才会清楚哪个选项更重要，玩家要有哪些不同的游戏操控选项。系统菜单既不是游戏设计中最难的部分，也不是最重要的，要首先敲定的是系统。

薄本文档通常是由那些自以为是的游戏设计者的管理人们做出的。这些文档又被称为“省略号文档”的原因是，它们经常乱用省略号。例如，玩家在游戏中的环境可能被如此描述：“丛林环境是一个又热又可怕的地方，在那里，Garguflax 猴子荡来荡去折磨玩家……”这就是整个文档中所有对环境的描写，以省略符号结束，就好像说“把游戏设计插在这儿”一样。





不清楚这些文档的作者们是否计划以后回来把省略的地方填上，还是根本认为真正解说游戏如何工作是浪费他们的宝贵时间。他们只是假设大概会有什么人把它们都填好。

省略符号文档的另一个例子是，“玩家可以选择多种很酷的武器。例如，Gargantuan Kaboom 比起玩家的其他武器有双倍的杀伤力，还有一种特殊效果。Barboon Harpoon 允许使用者在远处利用瞄准镜头杀死敌人。其他武器也很有趣很酷……”这里省略符号的作者没去对游戏中的武器进行任何细节描写，只列了两种武器，而把余下的交给读者去想象。当然了，读者已经十分有效地被告知其他的武器“很有趣很酷”了。省略符号的作者误以为那就是开发游戏所必须进行的所有描述了。

薄本或省略符号文档的惟一优点就是，允许其他人利用它的设计，别人能轻而易举地接过项目，并把它变成自己的。我说这是个好的方面，因为通常这些经理人员写在薄本文档中的想法总是离奇无比，无法在游戏中实现。还特别要小心，当这个经理 6 个月后抱怨说：“那不是我写的东西”的时候，问题就来了。



17.4.2 背景故事书

与省略号文档的作者们不同，写“背景故事书”的设计者在文档写作上面花费很多时间。这些书（很难把它们仅仅叫做文档）通常长达上百页——300、400 甚至 500 页。这些文档里面包含有大量的信息。

这些文档所犯的第一个错误往往是目录表很差而且还缺少索引表。在一个设计文档中，组织有序的信息和好的目录表可以取代索引表，但这两者都没有就是大错了。当文档又像《战争与和平》那么长时，问题就更严重了。设计文档存在的首要原因是，让团队成员能迅速地查找所开发游戏某一方面的信息。如果一个程序员想知道某个敌人的人工智能如何工作，需要方便快捷地查到这个信息。如果程序员找不到这一信息，就可能去编造一些。同样，当一个画家想了解游戏某一区域的构造时，他也想尽快查到描写那个区域的地方。设计文档不是小说。没人从头到尾地读它。总体来说，设计文档属于参考资料，如果团队成员不能轻易获取信息，他们很可能会放弃它。

然而，如果一个人真的开始查看这些背景故事书，并吃惊地发现，实际上书里根本没有游戏操控方面的信息，书中全都是背景故事。在这 500 页中，背景故事远远多于游戏将要用到的。游戏中角色的历史、他们的朋友、他们的父母兄弟等一切细枝末节都有描述。它可能相当有趣（尽管总是组织得乱七八糟），但读者最后还是不知道游戏是如何工作的。许多游戏中故事叙述是游戏的中心之一，背景故事对创建游戏很有用处。在这种情况下，在设计文档中对游戏故事进行某种程度的讨论还是很对的。但从始至终，设计文档包含的都应是游戏的设计，这和游戏故事有很大的区别。虽然这种背景故事数量惊人，并很有可能打动那些爱冒险的投资者，但使用这种大厚书作为惟一工作指南的程序员最终很可能要自己来设计游戏。



17.4.3 杀伤力过度文档

有些设计者认为，他们能够在设计文档中描绘游戏的所有细节。确实，很多设计文档缺乏必要的有用细节，像我们上面提到的省略号文档，但与此同时，走向另一个极端也是对设计者





的时间和需要从过多细节中寻找有用信息而时间的浪费。更有甚者，这种文档还往往会设计者误以为他已经创建了一个详尽的文档，而实际上因为他对某些题目的细节写得太多，结果反倒略过了一些其他需要讨论的方面。

例如，假设游戏中有一些角色要在游戏环境中进行某些活动。比如说游戏中有一些居民，他们要走动、坐下和起立、互相交谈，还有睡觉。文档应当在 AI 部分描写这些行为。一个真正详尽的文档可能会把这些分解成单独的动画：从坐到站，从站到坐，随意站着，随意坐着，走动，用手势交谈，等等。恐怕这并不必要，因为优秀的动画师可以比设计者做得更好。有些设计者会狂热到真正画出或列出每个动画草图的程度。这太荒诞了。在设计文档阶段根本无从得知一个动画需要多少幅草图，这要在游戏开发过程中决定和调整，更不要说列出动画草图会让动画师因为被过渡约束而失去积极性。更重要的是，设计文档应当限于描写游戏操控的描述，不应进入艺术圣经或其他艺术文档。

另一个例子是我所说的“平衡数据”。也就是游戏中武器、物品和角色的实际统计值。设计文档应列出武器和角色会有哪些属性。比如，一个武器可能有射程、精密度、射击次数以及开火速率。另外，设计文档可能还应描写某种特定的武器：“双筒猎枪射程短，准确性低，但能造成大面积杀伤。”然而，在设计文档中实际列出一个武器的属性值没多大用处。比如说“猎枪精确度：2”没有任何作用，因为数字“2”没有上下文，所以也就没有意义。这些值最好等到游戏可以运行时再来决定，那时玩家可以使用这些武器，设计者可以尝试设定不同的值来达到最好的效果。在这些数据没法检验时就写出大量数据，是对时间的巨大浪费。

像动画细节和精确的平衡数据一样，源代码也不要放在文档中。在设计文档中就写出算法的设计者走得太远了。如果设计者同时也是程序员，这样做没什么关系，但设计文档中不该有编码，甚至是伪代码。写进编码只会让文档臃肿，并忽略文档中该有的信息。就算在杀伤力过度文档中有些有用的信息，它们也会被淹没在各种无用数据的洪流中，团队成员在它们面前会望而却步。杀伤力过度文档的作者以为自己能够预先计划好所有一切，而且自己远比团队的其他任何成员聪明。虽然对细节的过分重视对那些不知自己要做什么的人来说很有用，但如果设计文档走得太远只会激怒其使用者。

17.4.4 空中楼阁式的文档

这些设计文档通常有着真正杰出游戏的宏图大志。但可悲的是，其作者对操作电脑的能力以及一个 20 人小组在一年半时间里能做什么缺乏了解。结果，这些野心太大的文档成为毫无现实基础和无法实现的奇思怪想，只能以失败而告终。

空中楼阁文档包括诸如“一个曼哈顿的模型将成为玩家的主要游戏环境，实时的 AI 主体代表这个城市的 700 万居民”这类念头。空中楼阁文档的作者不想这些细节，如现存计算机系统不能模拟 700 万人，不管在什么样的时间框架下（更不要提实时了）。空中楼阁式的文档的另一个想法可能是“包含一个自然语言分析器，允许用户输入完整、复杂的英语句子，而角色们则会用他们自己富有活力的对话来回应。”这个该死的设计者不想听什么研究机构已经在自然语言程序上工作了数十年但还是无法应付短小的简单句子这种话。空中楼阁式的文档经常和省略号文档结合在一起，形成完全无用的设计文档，这类文档只能勾勒出完全没有可行性的项目，而不去费心探讨相关细节。





17.4.5 石化文档

上述任何一种有缺陷的设计文档都可能同时成为“石化文档”。事实上，就算没有上面那些问题的设计文档，即使一度是个良好的参考工具，如果设计者不勤快地更新它，在项目进程中同样也可能成为石化文档。我从没听说过，哪个游戏项目在开发过程中，它的原始设计没有经受过很大变动的；如果一个设计改变了而它的设计文档却没变，那个文档就开始成为石化文档了。

设想一下开发团队的某个程序员在石化文档中查东西。如果这个人找到的信息已经过时了，他们可能会用上旧的、早就改掉了的功能。另一方面，某个知道这些变化的设计者或出品者会注意到程序员正在创建一个已经不再恰当的系统，并因此责备那个程序员。这会使双方都感到灰心，更不用说浪费程序员的时间了。更重要的是，将来程序员想要知道某些设计时，他不再信任设计文档，而会直接去找设计者或出品者以查找某个系统应该是什么样的。当然，当设计者必须停下所做的工作来给程序员解释系统时，设计文档也就失去了意义。以后，新的系统可能会写进文档，但程序员却不会再去用这个石化文档了。当设计改变而设计者没有更新文档时，整个文档变得毫无用处。没人相信它，没人会去读它了。



17.5 重量问题

设计文档经常被戏称为不是用来读的，而是用来称的。如果说很多设计文档都很重，而团队成员很少有人想去读它，没什么稀奇的。令人震惊的是，这种情况经常是真的。我曾经听到某个主要游戏开发公司的前出品人谈到她关于设计文档和项目审批程序的亲身经历。她说决策者们通常带一个天平到他们的“上层”会议。如果有两个相似的项目都值得投资的话，他们会把这两个项目的设计文档放到天平上。哪个更重哪个就会被采用，而另一个则被否定。我非常痛心地向你告诉你，如果你在商业游戏公司工作，渴望开发商的投资，得让你的文档重一些。你需要给人深刻印象以便被采纳。很多人永远不会读它，但每个人都会拿起它掂一下重量。

当然，很多这些极厚的文档含有大量项目开发所需的不可忽略的宝贵信息。这些文档也可能是我在前面论述的某个失败的文档类型的例子，比如背景故事书或杀伤力过度文档。你很难做到让文档尽可能只提供实用有效的信息，同时使它足够重以打动别人。人们可能会在里面加上大量的流程图或概念草图，或者使用大号的字体，同时还不要太显眼。实际上，一个了不起的游戏（虽然同时也是简单的）可以有个仅仅十页长的完美的设计文档。不知道有多少伟大而简单的游戏，因为设计文档的重量不引人注目，而被开发商们扔到了一边。

17.5.1 让别人读它

设计文档写好后，面临的最大挑战就是让开发团队里的人来读它。通常，很多程序员、艺术家甚至其他设计者都不愿花时间去仔细读设计文档。其他人可能因为以前总是被埋在糟糕的





设计文档中，想当然地认为你的文档也一样差。让你的文档随时更新，只写进有用的信息，把内容控制在实际可行的游戏成分内，可能会有所帮助。如果团队成员看到某一部分文档发现它写得非常好，他们可能在开发系统或制作画面时会回来参考它。像所有文档一样，如果希望它们有人读，就要赢得读者的信任。

另一个让别人读设计文档的关键是，让那些想读它的人能够轻易找到它。把它放入团队使用的同一个资源管理系统中。你要使团队成员能像获得游戏最新进展一样容易地得到新近的文档版本。因为要不断地修改和更新文档，以保证它跟上项目的发展（防止它变为石化文档），资源管理将是追踪先前版本的一个有价值的工具。

当检验过文档的最新版本后，给你的团队发一封 e-mail，告诉他们你的文档已经投入使用了并做了什么修改。如果某些改变和他们的工作相关，他们能够从网上得到最新版本并查阅相关的更新信息。如果没人知道你更新了文档，或人们依然读旧版本，这种更新也就毫无用处。如果可能的话，给你的文档加上版本序号是个好办法，比如 1.3 或 2.7 等。加上这个版本序号和日期，人们就能知道他们读的是哪个版本的文档，他们是否需要一个新的版本。



17.6 文档只是个开始

一些设计者以为，一个详尽的设计文档本身就足以建造一个游戏了。还有些情况，公司从设计者那里买来设计文档，而那些设计者转而去写其他的设计文档，让另一个团队去实际执行他们的设计。一个设计文档只是粗线条的轮廓，仅仅提出游戏的设想而没有实际参与到游戏创建中，只要游戏的母盘还没做好，就不能说游戏被设计好了。一个以工作为重的设计者希望自己自始至终都在那儿，随时准备更改设计以使它更引人入胜，并在设计更改时更新文档（其他的也要不断修改）。一个投入的游戏设计者要在那儿平衡武器、AI、控制，当然还有关卡，以保证游戏没有偏离最初版本确定的目标。

如果设计者编写了设计文档，然后把它交给其他人去实际创建这个游戏，那些进行实际开发的人会按照他们自己的兴趣和艺术构来修改这个设计。设计文档将会成为他们自己创造工作的跳板，成功将不是原设计者的。设计文档是游戏开发中必不可少的部分，但它不是所要求的全部。设计者要参与项目的整个过程，才能作为完整游戏的开发者。换句话说，写作设计文档是计算机游戏设计的比较轻松的部分。按照文档实际来创建一个引人入胜的游戏，后续工作要难得多。



Chapter 18

第 18 章 Jordan Mechner 访谈录



对于 Jordan Mechner 在电脑游戏方面的贡献，人们能感受到的惟一缺憾是，他未能创造出更多的游戏。由他设计和领导创作的每一款游戏，如 Karateka、Prince of Persia、The Last Express 等都具有一种独特、典雅和精益求精的风格，这种风格在电脑游戏领域并不多见。不过，Mechner 曾几度退出游戏行业，从事同样令他痴迷的电影制片工作。事实上，正是他在电影方面的知识造就了他的游戏风格。但这种风格并非体现在史诗般的过场动画以及缺乏互动性的游戏机制上，虽然游戏开发者们在试图融合电影和游戏两种艺术形式时经常会出现这种情况。与此相反，Mechner 则以独特、新颖的方式融合了电影和游戏的制作手法，栩栩如生地讲述着一个接一个的游戏故事，同时仍然不失游戏佳作所具备的诸多品质。这次采访围绕 The Last Express 的发行而展开，旨在为《Inside Mac Game》杂志提供素材。本书在采纳这篇访谈录时，Mechner 曾慨然对文章内容进行了充实和扩展，使其包含了他在电脑游戏开发领域中整整 15 年的工作历程。



问：当初你是怎样迷上电脑游戏的呢？

答：嗯，那是在 1979 年，当时我还是一名高中二年级学生。我有机会接触到的第一台计算



机是校方为我们提供的 PDP-11 型机器。不过，大家并没有多少时间摆弄它，而且，管理这台计算机的教师不允许学生们阅读说明书，因为他担心，如果这样，我们将有能力深入系统，改变优先级以及诸如此类的东西。因此，我们只得凭猜测了解这台计算机的操作方法。后来，一位朋友向我展示了他刚刚买到的苹果 II 型计算机，这就好比一场美梦成为现实，从此以后，只要愿意，无论何时我们都可以在他家里共享这台机器了。这位朋友对我们彻底开放，我们可以打开机箱了解它的构造，并可以阅读所有随机附带的说明书。当然，具有讽刺意味的是，当时我无法找到任何关于汇编语言的说明文件。因此，我只好通过计算机操作系统的汇编码破译各种指令的含义。多年来，我一直在学习这方面的知识，并陆续出版了一些书籍。这完全归功于那台宝贝计算机。



问：你是否一直打算利用那台计算机制作游戏？

答：是的，我想，在当时游戏是我所知道的惟一的软件类型，也是我惟一喜爱的软件类型。当时，我确实尚未认识到文字处理或电子表格软件的用途。我尽情地玩着那些我所能找到的任何游戏，闲暇时也曾试着自己编写游戏。就是在这一时期，我第一次感受到了学有所用的滋味。



问：那么，这便是 Karateka 的由来了，是吗？

答：那是几年以后的事情。我制作的第一款游戏是 Asteroids，这确实是一个雄心勃勃的计划。我试图按照 Apple Invaders 的路数来创作 Asteroids，Apple Invaders 是当时最为流行的投币式游戏。我暗想，如果 Apple Invaders 因精确模仿投币游戏而一举成功的话，那么我也能靠 Asteroids 达到同样的效果。可是，时机似乎不那么凑巧。事实上，当时我已经完成了用汇编语言编写的、高分辨率的 Asteroids 版本，而且还与开发商签订了一项协议。但就在此时，Atari 公司领悟到，这些电脑游戏侵犯了它的街机游戏专利权，并将剥夺其巨额利润，于是，他们的律师向所有人士发出了警告，就这样，Asteroids 未能得到出版。



问：然后，你就开始制作 Karateka，是吗？

答：不，接下来我制作了一款与 Asteroids 极为相似的游戏。不过，这一次我采用的是色泽鲜艳、来回跳动的彩球，而不是石头，这些彩球在屏幕边框上弹跳，而不是绕行，于是我将它命名为 Deathbounce。我将它寄给 Broderbund 公司（那是 1982 年，我刚刚步入大学校门之时）。随后我便接到 Doug Carlston 的回复电话，Doug Carlston 当时负责处理投递作品，并承担公司的经营事务。收到电脑游戏业内人士打来的电话后，我感到万分激动。他说：“看来程序编得相当不错，流畅的动画等等给我们留下深刻的印象。不过还是觉得有点过时。请看一看我们新近推出的 Choplifter 吧。”好心的 Doug 给我寄来 Choplifter 的复制品以及一个游戏操纵杆。这是当时最为畅销的一款游戏，设计者是 Dan Gorlin。正是这款游戏使我真正领悟到，我大可不必摹仿他人的街机游戏，而完全可以按照自己的意图进行设计！

我将各方面的设想同时集中在一起，创做出 Karateka 这款游戏。我通过 Choplifter 了解到平稳滚动画面的方法以及新颖独特的设计方案。同时，我还获得大量接触电影艺术的机会；耶鲁





大学拥有大约12个电影学会，在这四年里，我尽力观摩有史以来拍摄的每一部影片，其中《七武士》是我最喜爱的作品。当时，我母亲沉迷于空手道，暑假期间，我在当地的一家柔道训练馆上过几节课。最后，我选修了电影研究课程（虽然时刻都有放弃的危险），并逐渐产生一些气势磅礴的幻想，认为电脑游戏是一种正处于萌芽状态的新兴艺术，就像1920年的卡通片或1900年的电影一样。于是，我将各方面的灵感全都发挥在 Karateka 这款游戏上。与众不同的是，我使用 Super 8 型摄影机将空手道老师的招数拍摄下来，并在剪辑机上一格一格地跟踪他的动作。这就是 rotoscoping 技术，早在1930年，迪斯尼公司就在《白雪公主》一片中应用过这种方法。这样制作出来的动画比手工制作的效果好得多，而且也是市面上其他游戏所无法企及的。Karateka 的制作花费了我两年的课余时间，在第二学年结束时，我便将它寄给了 Broderbund 公司。他们欣然出版了这部作品。

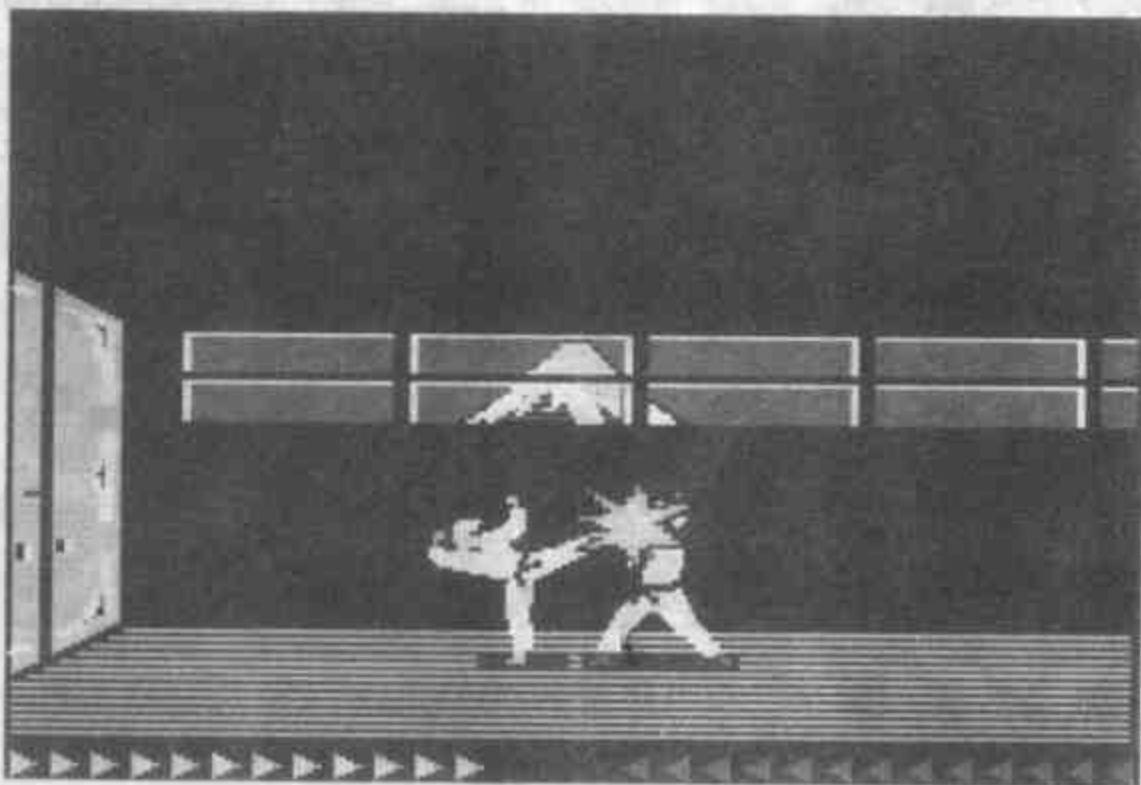


图 18.1 Karateka



问：那么，你的目标之一是将电影技巧融入动作游戏，创造出一种独特的复合体，是吗？

答：非常正确。1915年，D·W·格里菲斯曾采用逐渐加快的交叉剪辑手法营造悬念气氛，我觉得有必要在电脑游戏中尝试这一技巧。在处理场景过渡时，我沿袭了《七武士》中的横向画面消除法。此外，我还在游戏开头插入了滚动文字序幕。也有一些败笔，如在游戏结束时提示“剧终”而不是“游戏结束”等。我还绞尽脑汁地想出一些在 Apple II 型计算机上有效处理高分辨率图形的技巧。



问：事实上，Karateka 这款游戏相当短小，这是为了加强游戏效果而有意做出的决策吗？

答：其实，在当时，对我来说这款游戏并不短小。不错，当我将它投递给 Broderbund 公司时，它只有一个关卡：玩家们最好一进入宫殿便展开搏斗。他们首次向我提出的建议之一是，将游戏分为三个关卡：宫殿外面、宫殿里面以及宫殿下面。但我考虑的并不是游戏能持续多久，我只是想让它酷一点儿。





问：游戏结尾有一个令人费解的骗局，即如果玩家采用“攻击”姿势接近公主，公主就会用脚踢他。为什么要这样处理？

答：这似乎是一个挺逗人的小骗局。在这款游戏中，玩家只有一次生命：无论玩到多远，一旦倒毙就会“剧终”，即必须从头开始。因此我觉得，对大多数玩家而言，当他们最终抵达游戏结尾时，都会直接投入她的怀抱。但也不能说这完全是一个骗局，因为这里有一个小小的暗示：她向你伸出双臂，而当你跑向她时，她就会放下自己的胳膊。可以说，这就是发生不测的一种征兆。

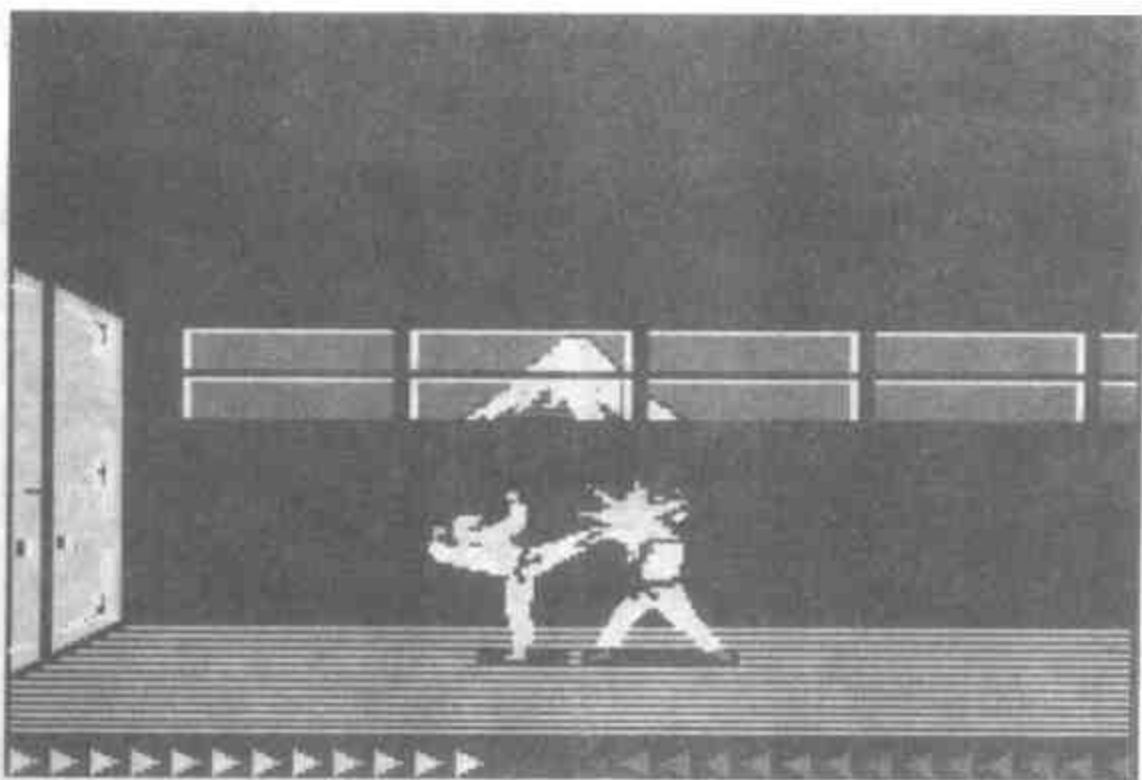


图 18.2 Karateka



问：不过我相信，任何人在第一次玩这款游戏时都不会顺利地闯过这一关。

答：是的，回想起来，可真令人感到难堪。不知现在我们是否还能成功地运用这一手法。在 Karateka 中，另外一件处理得当的事情是，如果播放磁盘背面，也就是说将磁盘上下面颠倒过来，游戏画面也会上下颠倒。我想至少会有一些人给 Broderbund 公司的技术支持部门打电话：“屏幕颠倒了，我想要么是显示器，要么是电脑出了毛病。”这时，技术支持人员就可一本正经地说：“噢，可能你把磁盘放倒了吧。”然后顾客就会满意地挂上电话，并认为所有计算机软件都具有这种特性。我觉得，对开发商来说，仅仅为了这么一出恶作剧而将每件商品的成本提高 25 美分是一种极有胆识的行为。



问：这么说，Prince of Persia 来源于你在设计 Karateka 时的经历，是吗？

答：是！不过，就我的个人经历来看，Karateka 和 Prince of Persia 之间存在着很大的差别。毕业后，我闲了一年工夫，也不能确定自己是否打算设计另一款电脑游戏。最直接的灵感来自 Ed Hobbs 设计的一款游戏，即 The Castles of Doctor Creep。不过，这款游戏不太流行，可能是因为它只能在 Commodore 64 机器上运行的缘故吧。在大学宿舍里，我和同伴们曾尽情地玩过这





款游戏。它具有鲁本·戈德堡 (Rube Goldberg) 式的匠心独运的谜题, 如敲击某个开关后, 一扇门打开, 另一扇门关闭等等。因此, 概括来说, 有关 Prince of Persia 的想法是, 在设计过程中将 The Castles of Doctor Creep 的精巧以及 Karateka 中流畅的动画结合起来。这样, 当你在奔跑和跳跃时, 就不再只是一个小精灵在空中飞来荡去, 你所扮演的角色就像确实具有了重量和质量一样, 当你摔在尖硬物上时, 它就会表现出受伤的样子。

另一个灵感来自《Raiders of the Lost Ark》的开头 8 分钟。我很想制作一款具有这种动作风格的游戏。此外, 还有天方夜谭中的环境。我试图寻觅一种尚未被电脑游戏滥用的环境, 于是, Broderbund 公司的两位动画设计师 Gene Portwood 和 Lauren Elliot 向我推荐了这本书。回来后我将天方夜谭重新翻阅了一遍, 我感到豁然开朗: 它包含了隐藏在人类集体无意识中的所有故事元素——妖怪、辛巴德之旅、阿拉丁的洞穴等。这本书非常适合于改编成电脑游戏。



问: 你说过在制作 Prince of Persia 之前你曾经空闲过一段时间, 是什么最终促使你准备重新开始制作另一款游戏的呢?

答: 那一年, 我正在创作我的第一部电影剧本。题材由 Larry Turman 挑选, 他是一个很有风度的人, 曾制作过大约 50 部影片, 其中包括电影《毕业生》。我们花了一年工夫与导演和制片厂联系, 眼看就要投拍, 但最终却未能达成协议。后来我觉得, 对一名初出茅庐的编剧来说, 这样的开头还不算糟糕。不过, 在这之前我已经尝到电脑游戏的甜头, 于是我暗自想道: “我的上帝, 为了等候消息, 我在洛杉矶呆了整整六个月, 而到头来电影竟然没能投拍。” 相比之下, 我认为, 如果完成了 Prince of Persia 的制作, 它肯定能得到出版。因此, 我得出结论: 最好还是坚持下去。此时, 在剧本的情况逐渐明朗之际, Prince of Persia 的设计工作已经进行了约六个月; 为了专心从事剧本的写作, 我曾将这项工作搁置了将近一年时间。耽误这么久之后再重新回到编程工作, 难免让人失去信心; 我担心自己已经忘掉那些源代码。不过, 我还是重新挑起这项任务, 并最终完成了它。

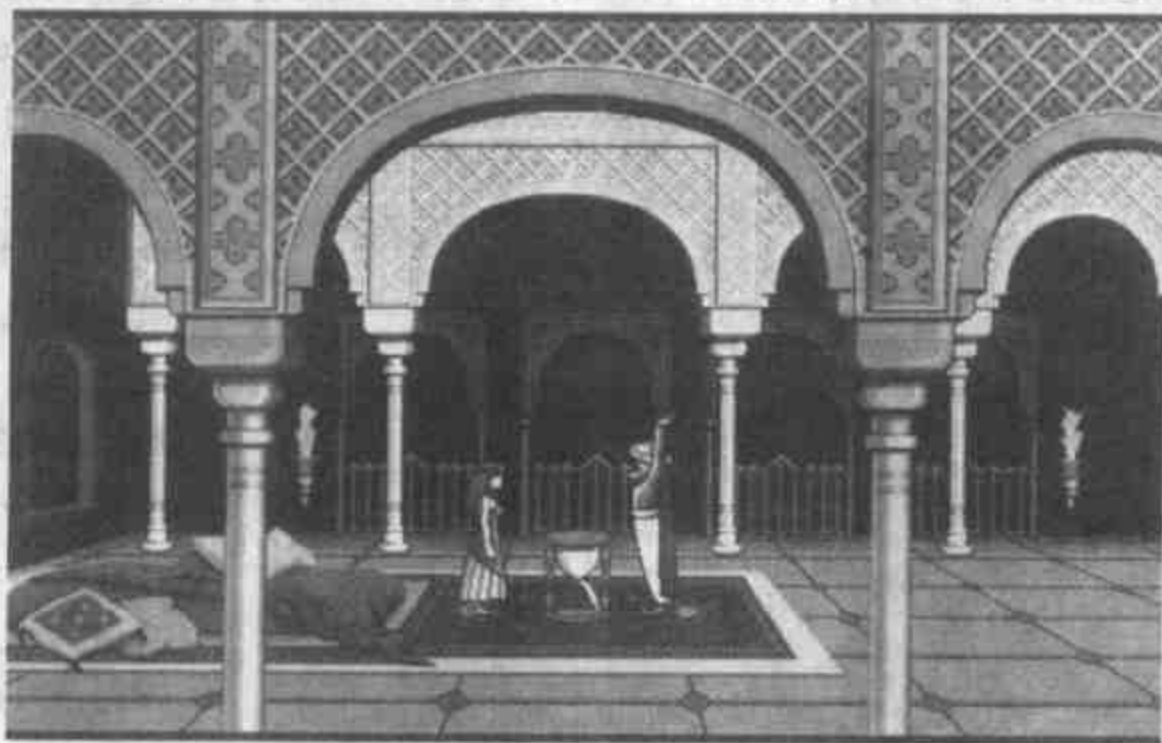


图 18.3 Prince of Persia





问: Prince of Persia 的特点之一是, 它采用了有限的游戏元素, 并将它们分布在所有关卡之中, 但却从不让人感到单调或重复。不知你是怎样做这一点的?

答: 这确实是设计工作中的一大难题。采用组合法可将有限的游戏元素以不同的方式重新组合。在电影制作中我们经常采用同样的手法: 插入一段对话和一件重要的物品, 15 或 20 分钟后, 就能以令人意想不到的方式使其产生某种效果。比如, 在 Prince of Persia 中, 那些松动的楼板就是其中一例。第一次遇到这样一块楼板时, 它是一个圈套: 你必须跳过它才不至于掉下去。不久, 松动的楼板重新出现, 但这一次不是圈套, 而是逃路: 你必须跳起来敲击天花板, 以便发现一块松动的楼板, 然后从下面击落这块楼板。接着, 还可用楼板撞击卫兵头部, 将他杀死、用它顶开压力板, 或者——这又是一个圈套——无意中打坏了压力板, 从而无法再将其打开。

由于计算机内存有限, 在 Prince of Persia 中这种组合法显得非常必要。流畅的角色动画、频繁出现的插入画面以及数不清的动作造型都会消耗大量的内存, 而当时计算机的内存容量只能达到 64KB。如果性能不成问题, 就可以无休止地提高游戏的效果, 如添加全新的游戏环境、特效或敌人。可是, 如果内存或磁盘空间消耗殆尽, 就只能发挥创造性的思维方式了。同样, 对玩家们来说, 他们也不得不多多开动脑筋。采用玩家们认为自己已经熟悉的元素时, 应该要求他们以不同的方式对这一元素进行思考, 要做到这一点, 确实得下一定的功夫。



问: Prince of Persia 的确是一款易于操纵的游戏, 与当代动作游戏相比, 这一点尤显突出。这是你的设计目标吗?

答: 是的。在 Karateka 和 Prince of Persia 中, 这一点是重点考虑的对象, 为此我常常花费很长时间努力寻找对某些动作进行整合的方法。它应该与游戏控制器配合还是与按钮配合? 就我个人而言, 我对那些需要使用一个或两个以上按钮的游戏持有强烈的偏见。事实上, 这就是我不习惯当代动作游戏的原因之一。当我摸清自己是否需要使用 A、B、X、O 键, 或控制板底部某个只有在紧急情况下才使用的小按钮时, 就早已失去了这些角色正在由我操纵的感觉。

理想的情况是, 应该使玩家们能极其熟练地操作游戏控制器和各种按钮, 这样, 他们就会逐渐感到这种动作仿佛就是他们自身的扩张。显然, 切实可行的方法是, 在添加一个以前从未使用过的新动作时, 应该以某种方式让玩家们预先“知道”哪一个按钮或什么样的操作组合能实现这一动作。对 Prince of Persia 中的某些动作, 我

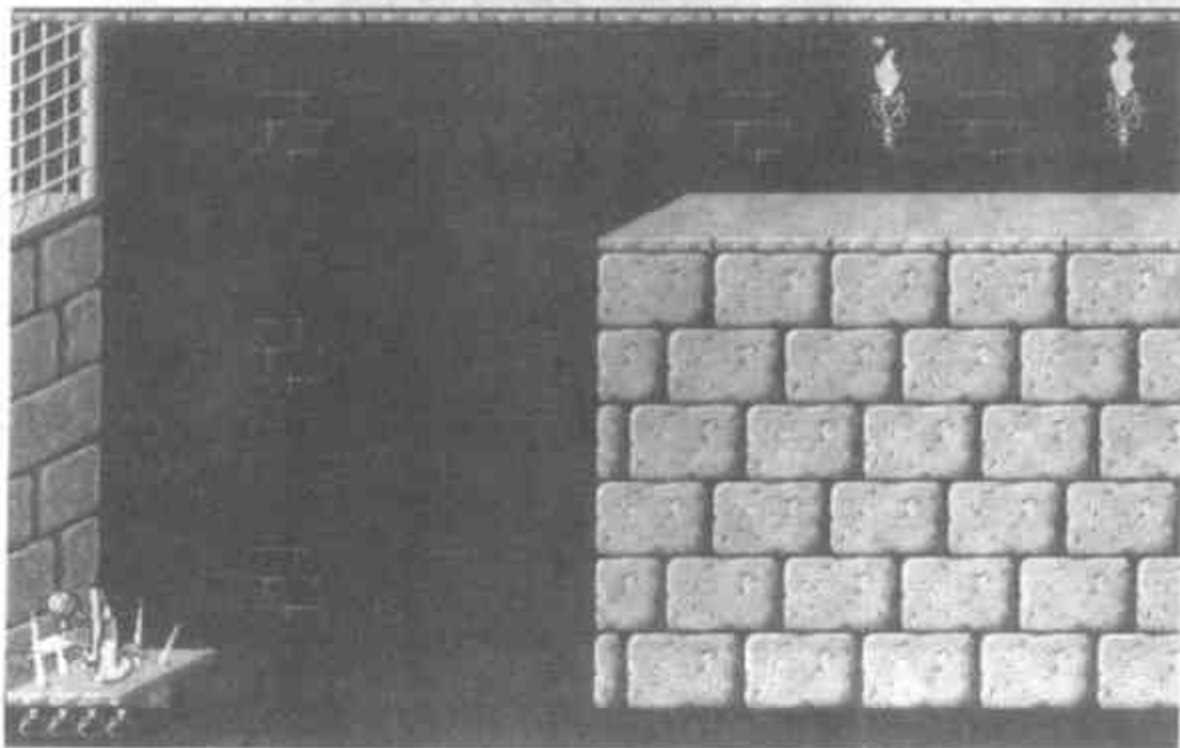


图 18.4 Prince of Persia



的看法是：“尽管动作不同凡响，但由于没有相应的操作按钮，还是放弃为好。动作妙极了，但从整体上看，它对游戏并无益处。”加强限制能使操作方法保持简单、协调。

问：就游戏设计来说，Prince of Persia 似乎是 Karateka 设计工作的逻辑延续。类似地，Prince of Persia 2 是 Prince of Persia 的延续。可是，The Last Express 似乎脱离了这一轨迹，转到了一个全新的方向。是什么诱使你创做出像 The Last Express 这样一款风格迥异的游戏的呢？

答：照我看来，The Last Express 并未转向。我依然在努力应付同样的问题：如何讲述故事，如何营造戏剧感以及玩家的参与感。自从推出 Prince of Persia 后，逐渐形成了一系列公认的动作游戏形式。在某种程度上，我对制作冒险类游戏产生兴趣的原因是，它似乎是一个非常广阔的领域，在那儿，还没有多少设计者能找到一种制作冒险类游戏的有效范例。

问：这么说，你的灵感并非来自其他冒险类游戏，是吗？

答：是的，而且事实上正好相反。纵观那些始于 20 世纪 80 年代的 Scott Adams 式的文字类冒险游戏，我们会惊奇地发现，从玩家们获得的体验（包括沉迷感以及从角色和故事中获得的生命感）上来看，冒险类游戏并未取得多大进展。因此，我试图振兴或改造这一濒临灭绝的游戏类型；我认为，正是这一挑战激发了我的兴趣。

问：你为什么要将这款游戏的环境设置在 1914 年的东方特快列车上呢？

答：在电脑游戏的设计过程中，我们总是试图找到一种能给人带来紧张感和冒险感的环境。而且，这种环境必须限制在有限的空间内，以便围绕环境设计出一款令人满意的游戏。例如，有些环境，如城市等极难处理。于是，我突然想到一列火车应该是绝好的游戏环境。这样就可以在有限的空间内安排有限的角色，而且也可避免那种身处鬼屋时的静态感，因为事实上火车本身就在不断运动。游戏一开始，玩家就置身于一节封闭、运动的车厢，它不仅正在驶向自己的终点，即从巴黎开往君士坦丁堡，而且时间也在不停地流动，即从 7 月 24 日到 7 月 27 日，从和平转向战争。时钟的滴答声表现出一种不可抗拒的趋势，并推动着情

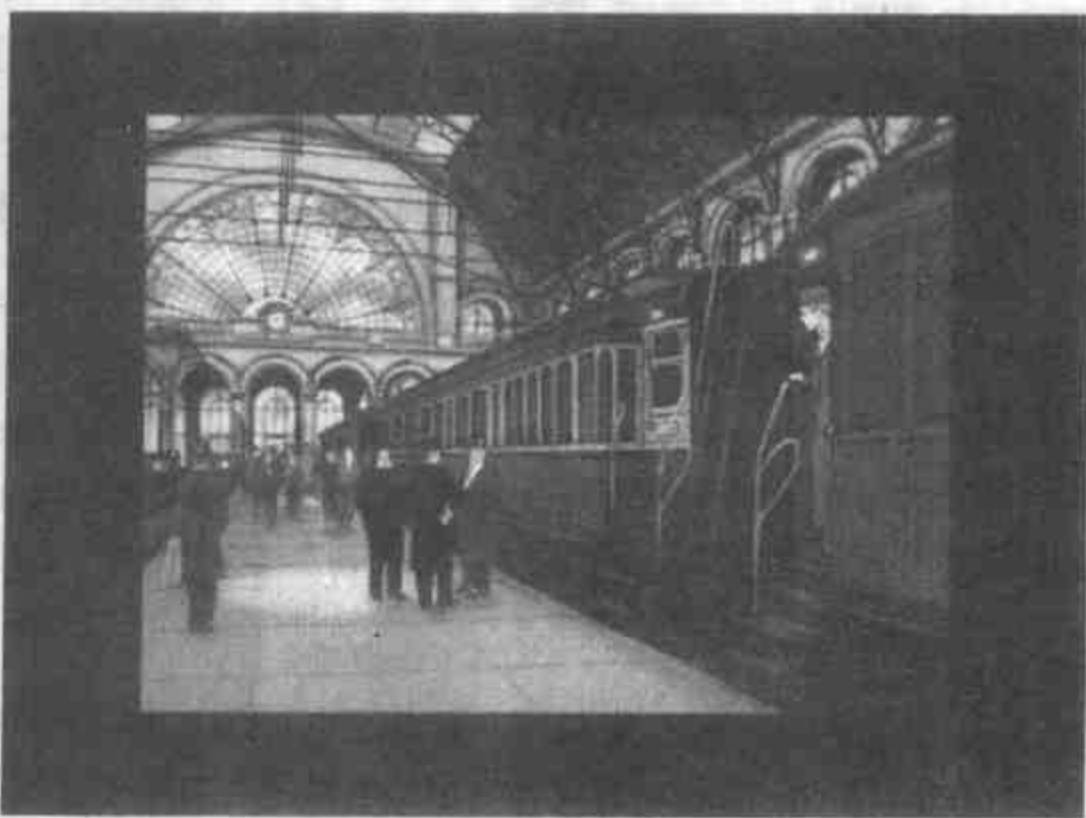


图 18.5 The Last Express





节的发展；对电脑游戏来说，我认为这种手法恰到好处。

毫无疑问，对于这则有关第一次世界大战爆发的故事来说，东方特快是一个理想的选择。在1914年，东方特快尚属“新生事物”，就像当今的新欧洲经济共同体一样，它是一项创新成果，是欧洲统一的象征。当时，乘坐这列快车可以在数日内畅通无阻地横跨欧洲大陆，而以前则需花费数星期时间。在列车上，可以看见来自不同国家、不同社会阶层以及不同职业的典型人物——限制在狭小环境中的欧洲社会的缩影。这些人们一道旅行，一起做生意；可是，战争爆发后，他们猛然发现自己已被民族阵线分割开来。这场战争不仅毁掉了原有的社会结构，而且也毁掉了承载东方特快的铁轨。我觉得，在对这场战争的描述中，东方特快是一个极富戏剧性和令人辛酸的象征。此外，游戏环境也极好地衬托了故事情节。



问：谈到 The Last Express 的创作起点，不知你是说：“我打算制作一款冒险类游戏，采用这种形式，我该讲述什么样的故事呢？”还是说：“这就是我希望讲述的故事，应该采用哪种游戏类型才能使故事更加生动呢？”

答：当然是后者。Tomi Pierce（The Last Express 的合作者）和我打算讲述 1914 年战争爆发前夕发生在东方特快上的一个故事。从何处着手呢？其实，我并不在乎它是否会成为 Prince of Persia 的变种，或者它是否具有市场价值。直到我们编好故事、确定角色数量、对他们的动机和性格加以渲染并认识到对话和采用不同语言的重要性后，才知道自己设计的是一款冒险类游戏。我有意回避冒险游戏的色彩，因为我个人对大多数冒险类游戏并不欣赏。我希望玩家们能在列车上四处穿行，并能看到周围那些熙熙攘攘、充满生机的人群，从而产生一种贴近现实的感觉，而不是像在一般的冒险游戏中所表现的那样，步入一个空空如也的地方，无法施展自己的身手。



问：这就是你在 The Last Express 中添加“实时”因素——虽然它在冒险类游戏中并不常见——的原因吗？

答：当然，这并非技术意义上的实时，它与放电影并无二致。时钟不停地滴答作响，我们只是在时间流逝的速度上做了些手脚而已。在需要渲染戏剧效果时，我们将时间放慢，在铺叙事件时则加快时间步伐。当玩家不在列车上时，就略去这段时间……这样，当他重新出现时，时间便已经过去一个钟头。



问：不过，它的实时感仍然比一般的冒险类游戏更强一些，人们可能会感到不大习惯。

答：甚至还包括动作游戏。我对许多所谓的动作游戏感到不解的是，当你放下操纵杆，对着荧光屏靠在椅背上时，看到的是一片空白。一旦消灭了房内敌人，你就能这样坐上几个小时。



问：你在前面谈及电影制作，据我所知，你在 1993 年曾独立制作过一部纪录片，片名是 Waiting for Dark。你在电影制作方面的经验是否有助于 The Last Express 的制作？

答：受益匪浅。不过，我认为这也可能是一种不利因素。电影艺术包罗万象，在过去一百年的电影发展史中，电影制作者们积累了丰富的技巧、惯例和风格，让人叹为观止。其中一些





已经在电脑游戏中得到应用而且效果确实不错，另一些有待开发，其他一些则不但起不到什么作用，反而还会在应用过程中削弱游戏效果。典型的例子是所谓的“互动电影”，即通过选择目录将一系列片头动画串在一起：如果这么做，就会出现片头动画 A，游戏继续；如果那么做，则出现片头动画 B，游戏失败。在 The Last Express 中，我希望玩家们能在列车上自由活动，四周洋溢着人间的气息，其他角色都在忙着各自的事情。如果有人过道处和你擦身而过，你可以转过身来，看着他向过道另一边走去，然后跟上他，了解他想去的地方。如果对此并无兴趣，也可以继续走自己的路。我认为这是一种在最具线性特征的环境中，即特快列车上的非线性经历。



问：你设计的每一款游戏或多或少都包含着一些片头动画，在 Karateka、Prince of Persia 和 Last Express 中，这些片头动画与游戏结合成一个有机的整体，因此在视觉上无法将其与游戏区分开来。这是你有意安排的吗？

答：是的。这三款游戏都包含一些审美价值，比如，当你停下来望着屏幕时，就能享受到一种流畅的视觉经历，像看电影一样。另一方面，如果进入游戏状态，能得心应手地进行操纵。这一手法无论对玩家还是对站在玩家背后的旁观者都具有感染力。片头动画和游戏画面应尽可能在视觉上保持一致。为了制造悬念，Karateka 采用了实时交叉剪辑手法：当你跑向卫兵时，画面切换到向你跑来的卫兵，然后再回到你自己，最后返回到卫兵进入画面时的场景。虽然这是一个极为简单的例子，但其效果却相当突出。

请看 Last Express 中的一个类似例子：当你处在第一人称视角时，看见 August Schmidt 沿着过道向自己走来时，画面就会切换到 Cath 和玩家角色对这一动作的反应。然后，你就会听见 August 的声音，画面随之切回到 August，整个过程几乎无法让人意识到游戏已经切换到片头动画，并开始第三人称对话。当 August 沿着过道远去时，这一场面也随之结束，玩家也返回到

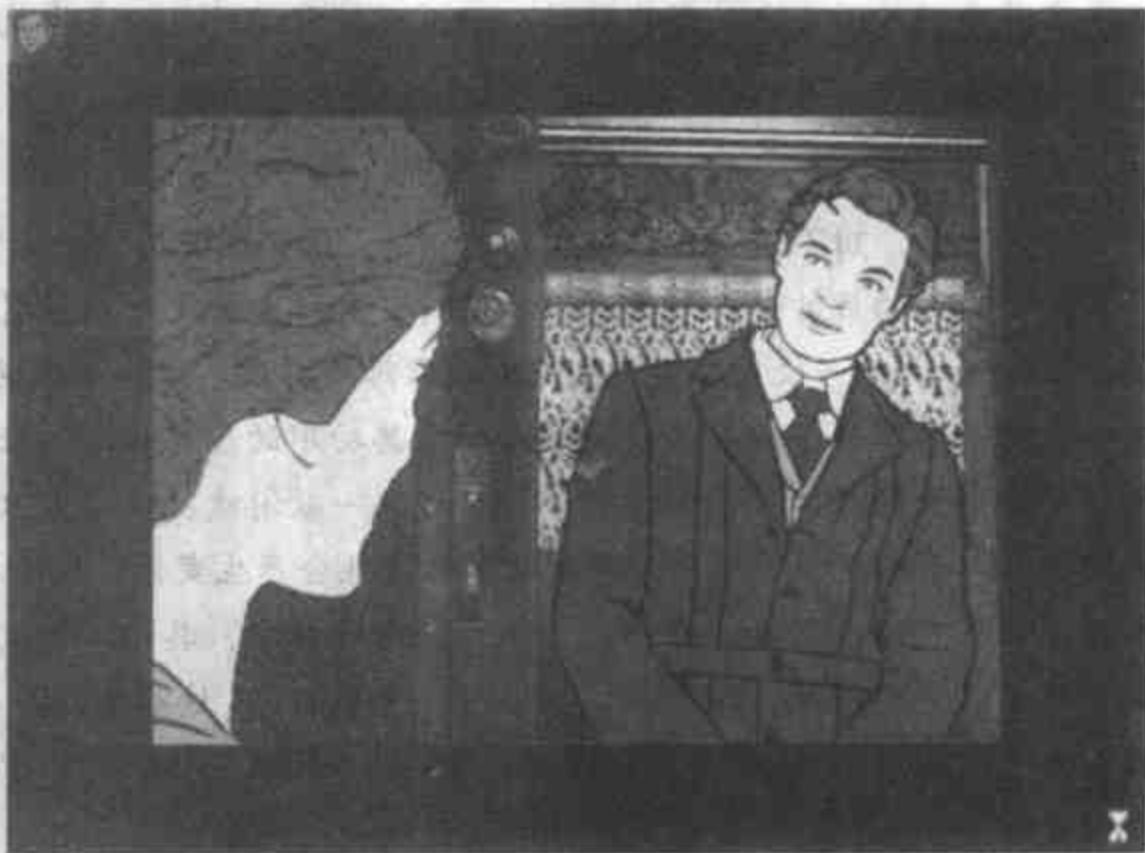


图 18.6 The Last Express

第一人称视角，并重新开始操纵游戏。我们凭直觉就能理解这一连串画面的含义，因为这种场面在电影中极为常见。一个经典范例是 Alfred Hitchcock's Rear Window (阿尔弗雷德·希区柯克的后窗)：整部影片构建在客观镜头、视角镜头和反应镜头(穿插镜头)的基础之上，其中大约半数镜头通过 James Stewart 的主观视角展现在观众面前。这也是 Last Express 在运用蒙太奇手





法时采用的基本结构单位。



问：另一方面，在 Prince of Persia 2 中，片头动画其实只是一些预先绘制的图像，它们看上去与现场游戏差别很大。记得在当时我似乎并不十分喜欢这些……

答：我同意你的看法。那些片头动画给人一种脱节的印象，并让人感到好像是在看小人书一样。不过，在当时这是我们刻意追求的效果。



问：眼下，在电脑游戏领域似乎存在一种回避全动感片头动画的趋势……

答：确实如此，因为全动感片头动画的成本有时与整个游戏不相上下，而且对于它们是否能真正改善游戏效果还存在着争议。另一个问题是，在大多数情况下，与优秀的电视和电影相比，片头动画的质量相当低下。



问：于是你就有意识地融合电影和游戏的制作方法，试图创造出一款全新的电脑游戏，是吗？

答：我的愿望是，Last Express 能提供其他冒险类游戏或任何类型的游戏从未提供过的东西。你会发现自已处在一个充满人群的游戏环境中，那些生动、丰满的角色不仅在外表上各不相同，而且还具有鲜明的性格特点、在故事中的意图以及各自的行动计划。这样就营造出一个视觉效果丰富并充满人间气息的游戏环境，通过比较常用的鼠标点击机制，玩家们就可与这样一种环境产生互动了。



问：那么，对于玩家与游戏的互动方法，你是如何设计的呢？

答：我们的目的是尽量做到简单易行。鼠标点击之所以能引发我的兴趣是因为，我始终将 Last Express 看作一款主要面向主流成年群落的游戏。那些不常玩电脑游戏而且对操纵杆并不拿手的人们是不会静下心来了解大量按钮及其功用的。在我们的社会中，鼠标点击是成年人熟悉的操作方法。因此，关键在于能设计出这样一款游戏：玩家们只需握住鼠标，并使之在屏幕上移动即可。当鼠标指针移动到不同的区域时就会产生变化，提示相应的行动方式：可以向左转，也可以同另一角色搭话。在尝试这一方法的过程中，具体工作机制也在不断改进。开发期间，我们解决了一些问题，如：对于“‘向上’和‘向前’这些动作是否需要采用不同形状的指针？”我们决定不采用；对于“‘仰视’和‘起立’是否应该有所区别？”我们决定一律采用上箭头。在很大程度上，鼠标点击是这种独特设计的一个方面。



问：那么，你为 Last Express 拍摄了多少影片素材呢？在这款游戏中，影片素材似乎格外丰富。

答：由于规模庞大，整个项目的后勤工作难度很大。事实上，拍摄这些影片仅花费了三个星期。时间并不太长，因为拍摄一部普通的故事片至少得花费四个星期，平均每天拍摄三页剧本。可是，在这三个星期内，我们每天能完成 15 页剧本。我们采用了一些能加快进度的技巧：





所有物品均为蓝色；事先将声音录好，在拍摄时则不必录音；放慢拍摄速度，有些场景以每秒7帧拍摄，另一些则以每秒5帧拍摄。这是因为，我们的目的是选择关键性的画面，然后再将它们制作成动画，这一点在游戏中不难看出。在这些技巧的配合下，我们拍摄了大量的素材影片。

不过，至于拍摄情况……举个例子吧，第一阶段是拍摄列车过道。我们铺设了一道55英尺长的小径，在涂成蓝色的地板上添加了一些黄色线条，在小径后面竖起一面可回收使用的墙壁。在三天内，我们让所有30个角色沿着列车过道来回走动。当角色靠近摄影机时，关键性的时刻是目光接触的那一

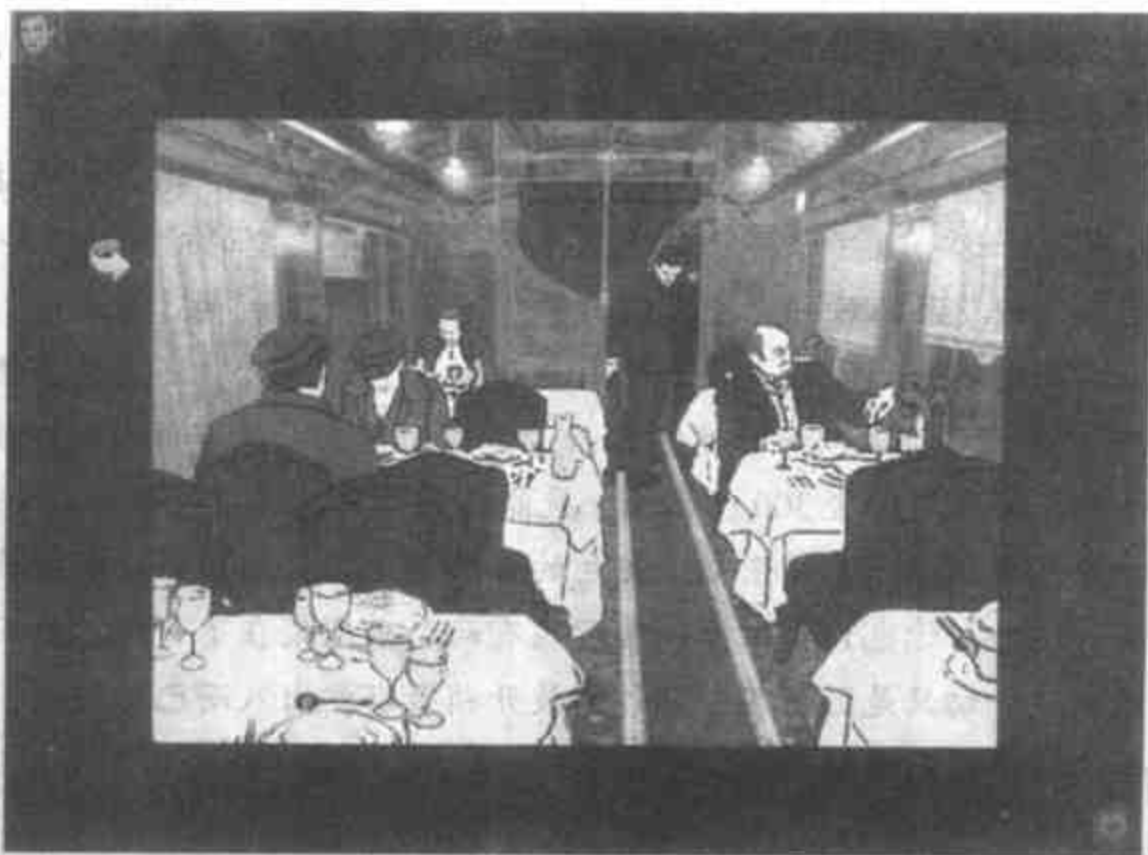


图 18.7 The Last Express

瞬间——无论是友好的还是敌视的目光。这种微妙的目光是使玩家们融入游戏角色如 Cath 等的因素之一，它能使你感到自己不再是列车上的一个幽灵，因为即使在过道上与你擦肩而过时，人们也会对你有所反应。在头三天里，我们仅仅拍摄了在过道上的行走动作，工作极为细致。在这一期间，我们锁定了摄影机，使其保持静止，因为一旦移动，拍摄到的镜头就无法衔接。

在过道上逗留三天后，我们转到餐厅。同样，我们必须以一种极不寻常的方式投入拍摄。在拍电影时，通常是一次拍摄一个场景，摄影机将以不同的设置拍摄每一个场景，而我们则是一次采用一种设置进行拍摄。每采用一种设置，我们会拍摄整个故事中处于这一角度的所有场景和动作。在每一个方位，我们都会将摄影机固定起来，即采用“坐在桌旁看着前面”的视角。我们会摆上另外一些桌子，然后以这一视角拍摄每一段动作——August Schmidt 走进来，坐下，订餐，侍者端上食物，于是开始就餐，放下餐巾，站起身来，走出去。然后，我们又将摄影机置于餐厅内的不同角度，这样，我们就能看到同一个演员重复同样的动作。为了尽可能提高拍摄效率，我们费了不少心思：安排哪些演员参加哪些天的拍摄，什么时候让他们离开，将摄影机多移动一次，以便让某些演员早点回去，还是保持原来的位置，付给演员们一天的工资。采用哪一种方法更为经济？这19天无疑是一段极为复杂、艰难的拍摄过程，我们从许多角度拍摄了大量的动作，因为在这款游戏中，我们无从知晓玩家们将采用什么样的视角。



问：拍摄完毕后，整理素材肯定是一项极其艰巨的任务。

答：我们是在一台 Avid 编辑机上进行这项工作的，如果没有这台机器，我不知道事情会是什么样子。我们将所有素材输入这台编辑机大容量硬盘上，这是一种以 Macintosh 机为基础的非





线性编辑系统。我们将这台 Avid 机的性能发挥到了极限。有一次，电影编辑不得不打电话寻求技术支持，原因是系统运行速度大幅下降。当他告诉他们自己制作的特技数量后，他们大吃一惊，不敢相信这台机器仍然能够运行。仅仅为一个场景处理的叠化图像比他们所能想到的任何普通故事片中的还要多。我们挑选一些静止画面，然后让它们相互叠化，这样，游戏中的每一帧画面都是一个特技。

根据正式的统计数字，我们在这款游戏中加入了 40 000 帧动画图像。不过，比起一部动画故事片，这一数字小得让人感到迷惑。在一个典型的对话场景中，静止画面之间的叠化次数平均为每秒或每两秒一次，而常规电影则以每秒 24 帧的速度放映。因此，为了达到这一水准，根据实际拍摄的动作量，必须用 40 000 乘以 24。此外，还有许多画面可以重复使用。比如，我们以 150 帧画面表现一个角色沿着过道走向镜头，然后再以同样数量的画面表现他离开镜头时的情景。仅仅利用这 300 帧画面，就能让列车员这一角色在游戏中走上 10 小时。步入餐厅时，你会看见六张桌子，每张桌旁发生的事情都是相互独立的。如果将游戏从头至尾玩上五遍，在玩第六遍时，你也许会注意到餐厅内有两个角色聚在了一起，然而，在这之前，他们始终是分开的。这一切只是因为某个动作的展开稍微有些出入而已。因此，在很大程度上，这些画面的组合次数是毫无限制的。



问：那么，你为什么要在 Last Express 中采用每一秒或两秒钟使图像叠化一次的特技呢？为什么不在整个游戏中采用更为常规的全动态风格呢？

答：在我们看来，全动态基本上是一种昂贵的特技。它的视觉效果非常出色，如在过道和打斗中所表现的那样。可是，如果我们决定在整个游戏中采用这一手法，我想可能最终会造成一种外表华丽但缺乏深度的风格。我们不仅受到内存保存画面的能力以及 CD 碟片数量的限制，而且最终还要受到处理器运算能力的制约。当你步入餐厅时，里面布满人群，这时，一系列相互独立的动画场景同时显示在屏幕上，CD 碟片也播放出多声道音频流。要达到这一效果，就只能让每个角色每隔几秒钟活动一次。

此外，全动态技术在美学上也存在着一个缺陷。假如技术障碍已经得到克服，我们可以为一个正在用餐的角色制作 30 秒钟的循环播放动画。但迟早你会意识到这个角色正在不断重复同样的动作，因此你会问：“为什么每当他喝一口酒，吃一口



图 18.8 The Last Express



牛排后，牛排总是一点不少呢？”这对游戏毫无益处，因为这些微不足道的全动态细节会分散玩家的注意力。在这一点上，我们认为，重要的是使玩家们相信这一角色确实在餐厅里呆了一小时十五分钟，而且在这段时间内玩家可以随时和他交谈。事实上，采用静止画面相互叠化的手法也能出色地表现“用餐”时的情景，而且在某些方面给人的印象比全动态图像更加深刻，因为在影片循环播放时它可以避免上述干扰。因此，一旦玩家接受这一手法，就能畅通无阻地讲述这一结构庞杂的故事：在三天三夜的时间里，30个角色干着各式各样的事情。假如我们拘泥于全动态技术，故事的生动性就会大打折扣。



问：我在制作人员名单中注意到，几乎所有角色都由两名演员扮演，一名承担有形表演，即玩家在屏幕上看到的形象，另一名担任配音。你为什么要采用不同的演员来表现这款游戏的视觉和听觉因素？

答：演员的安排是一项极其艰巨的任务，因为在所有角色中只有两名美国人，其余的是法国人、俄国人、奥地利人、塞尔维亚人、阿拉伯人……东方特快是一个确实确实的多语种世界。我们决定不仅让角色能说带有外国口音的英语，如当他们与作为主角的美国人交谈之时，而且也能说他们的本国语言，当然，每到此时就会出现字幕。当两名法国列车员在下班后相互交谈时，他们自然会说法语。因此，模仿德国或法国口音的美国演员根本无法让人接受。我们需要为每一种语言安排持本族语言的人。非常幸运的是，我们找到了一个无论在形象上还是在语音上都非常优秀的演员班子。可是，如果要求完美的形象、口音和国籍全都集中在一个演员身上，那就难说了——尤其是在旧金山，以我们有限的资金！值得一提的是，由于没有采用全动态配音手法，在演员的安排上反而更具灵活性。

Tatiana 就是这样一个例子。为了找到符合这位16岁俄国公主形象和口音的演员，我们联系了三家演员代理机构，并对来自洛杉矶、旧金山的数百名演员进行了考核。承担配音工作的女演员是一位住在洛杉矶的俄罗斯人，在影片中亮相的则是一位来自旧金山的美国人。如果想找到一住两金其美的演员，那就只好出国了，即使不一定是去俄罗斯！

顺便提一下，我们先录好音，然后才开始制作相应的动画图像，这样，配音演员就能自由地发挥自己的表演风格，就像为广播剧和迪斯尼卡通片配音一样。与后期录音相比，这一方法能使语音表演更趋自然。我认为，如果迫使演员为已经拍好的动作配音，就会损害表演的发挥。



问：在设计这款游戏的过程中，真实性似乎是你的首要目标，无论是从母语配音还是逼真模仿列车车厢来看。你为什么如此看重这款游戏的真实性？

答：这关系到对玩家负责的问题。我认为，无论对于历史环境还是虚构环境，游戏制作者们对细节和一致性的注重程度都会引起玩家们的关注。玩家们也许并不会留意那两个列车员，也不会了解到其中一个即将退休，另一个是位已婚青年，或者他们的政治观点位处两极等等，可是，即便如此，每当你在过道上碰到他们，无意中听到他们的只言片语，你就会下意识地感到，这是两个真人在进行交谈。如果我们并未下此功夫，那么，每当玩家们路过此处，他们就会听到一些做作的语言，并暗想：“得了，这声音就像在特地为我做戏一样。”即使在游戏中你看到的只是冰山的顶部，所有角色的生活经历及其本来面目都隐藏在表象之下，你也会感受





到所有这一切的实际存在以及它们的分量，虽然事实上除了冰山顶部你什么也看不见。



问：总的来说，你认为电脑游戏应该尽力追求真实吗？

答：其实，真实的含义非常丰富。我并不是说在表现真实环境方面电脑游戏应该更加富于真实色彩，即使像 Super Mario Bros 这样完全采用虚构环境的作品也有其自身的一致性。如果在某个场景中一个角色能从悬崖上跳下后飘然落地，在另一个场景中就不能让他摔得粉身碎骨。只要创作者肯下功夫思考：“在这个世界上重力有何规律？人在什么情况下才会受伤？”只要游戏具备自身的规则，玩家们就会欣然接受。在 Last Express 中，我们选择了一段真实的历史，并特别注意忠实地描述那一时期的世界局势；此外，在限定故事环境时也非常重视这一历史的真实。



问：在 Last Express 中，你采用了一种极为独特的技巧：尽管演员们都被拍入影片，但最终他们看起来就像绘制精美的卡通人物。你为什么决定采用这样的制作方法？

答：首先，在美学观点上，我对卡通形象非常欣赏。我觉得在 3D 背景的衬托下卡通人物的形象极富魅力。像 Snow White and the Seven Dwarfs 这类影片采用平面图像是有其技术原因的——它们都是画在胶片上的——不过这些影片还是出色地表现出角色的形象。我认为，对于那些像孩子一样希望步入卡通世界并扮演其中一个角色的人们来说，这种形象具有丰富的含义，

我认为，与真人拍摄相比，在电脑游戏中采用卡通角色还具备另外一个优势。只有当游戏玩家们置身于幻想世界，摒弃怀疑，并相信自己的行为确实能对那些虚构角色造成影响时，他们才能感受到乐趣。倘若看到的是摄制



图 18.9 The Last Express

好的真人形象，他们就会想到，这是在摄影棚内利用服装、灯光和摄影机拍摄的人物，即他在屏幕上的言行举止都是事先安排好的。他们知道这只是电影片断而已。可是，由于卡通片本来就不具备真实性，因此，如果玩家们相信卡通角色可以走路和交谈，那么，他们的举动为什么不能影响这些角色的行为呢？例如，当他们看见玩家走来时就会跑掉。



问：这样做有助于消除怀疑，是吗？

答：或者，至少不会像电影动作那样加深怀疑。我认为这就是插入影片往往不能成功地应用于电脑游戏的原因之一。它根本就不适合这种应用方式。

当然，卡通风格之所以能在 Last Express 这款历史题材的游戏中起到有效作用，还有另外一个原因。我们搜集到的涉及 1914 年以来文化氛围的图像大都是一些当时的绘画作品，比如由 Alphonse Mucha 和 Toulouse-Lautrec 等艺术家创作的报纸插图、杂志广告、海报图案等。这些作品都具有一种新艺术主义风格，而这种风格正是现代漫画书的前身。因此，我认为，如果我们看见身着 1914 年服装的人们以卡通形象出现，就会产生某种程度的亲切感。可是，如果看见这些人们以 3D 多角模型亮相，就无法产生这种共鸣。



问：你是否认为环境更富于现代气息的游戏也能利用卡通角色获得这种视觉效果？

答：是的，我很赞成这种表现形式，它在许多场合都能发挥作用。我并不认为它仅仅适用于历史环境。不过，对 Last Express 来说，它之所以如此完美，这只是原因之一而已。



问：既然这些角色最终看起来都像卡通人物，为什么不在一开始就将他们画出来，而是先将演员拍下来，然后再把他们处理得像一幅幅绘画？

答：原因之一是，为了获得高质量的动画和胶片 cell 类型的表现手法——正如在迪斯尼影片中看到的那样，就必须像迪斯尼公司一样花费巨额资金。尽管按照电脑游戏的标准，这款游戏耗资不小，但比起动画故事片制作费用来，就显得小巫见大巫了。为了确保一致性，必须使同一个角色始终保持原来的模样，无论他们位于近处还是远处，生气还是高兴，无论不同角度的形象如何难以绘制。要想通过 4 万张动画图像达到这一效果，以现有的资金是根本办不到的。

我们利用自动 roscope 技术把图像处理成黑白效果，并将它转化为类似于钢笔线条画的形式，然后由美工对图像进行润色，在不到两分钟的时间里完成着色工序。整个过程非常快捷，就像流水作业一般。此外，由于数字化和 roscoping 的过程均为自动控制，因此，即使两名美工处理同一个角色，也会得出极其相似的结果。例如，无论某一序列图像由谁来着色，Anna 的形象始终如一。

我们不想让图像看上去就像处理过的电影画面，也不希望它与卡通图片一模一样。假如你看见一个角色沿着过道向你走来，而你无法确定自己看到的是一幅绘画还是处理过的电影图像，那么，在很大程度上，我们已经达到了自己的目的。我认为我们已经做到了这一点。偶尔有人会问：“这些都是手工绘制的吧？”如果他们无法分辨出这是电影图像，那就大功告成了。



问：我认为，在这款游戏中，最富于创新精神的设计特征之一是你们采用的游戏保存系统。其实，玩家们根本不必保存游戏，因为 Last Express 会自动记下他们的一举一动，这样，如果他们希望尝试不同的玩法，就可以“倒退”至游戏的任一场景。你为什么要采用这一系统的呢？

答：很高兴你能提出这样一个问题。我为这种游戏保存系统而感到非常骄傲。有趣的是，一些人，包括某些评论家对这一功能一无所知。直到现在，我们偶尔还会在一些评论文章上看





到这样的言论：“竟然无法保存游戏，这太扫兴了。”当然，我们的目标是，推广那种应用于点击系统的设计原则，即尽量保持简单、明确和直观的操作特点。如果必须考虑到自己是在一台计算机上操作，必须保存文件，给文件命名，并对保存后的游戏文件进行比较，我想这就会破坏游戏经历的连贯性。我们的想法是，玩家们只需坐下来玩自己的游戏，如果打算停下来，就直接退出游戏，然后要么去吃晚饭，要么用计算机干别的事情等等。当玩家回到游戏时，就能自动跳到上次退出的地方。如果操作失误，还可以像倒录像带一样，回到你认为做得不对的地方，从那儿开始游戏。我认为这一方法非常奏效。我想，采用六个颜色不同的鸡蛋也许是受到了 Monopoly 的启发。在 Monopoly 中，你可以选择自己想要的物品：帽子、小车……我们的思路是，如果你家里有六个人，那么每个人就能拥有一个鸡蛋，想玩游戏时，便可切换到自己的鸡蛋，从上次退出的地方开始游戏。有人会抱怨，不多不少只能保存六个游戏文件，或者只能使用颜色而不能使用文件名，这些人的思路显然停留在传统的游戏文件保存系统上，因此，他们无法理解这一要点。鸡蛋所代表的文件并不是保存下来的游戏，它实质上是一段录像，这段录像不仅包含最近保存的地方，还包括没能停下来保存的所有地方。通常，我们可以回放距离目标处三到五分钟的实时录像。



问：在 Last Express 中，音乐似乎也得到了有效的运用，它随着游戏情节的发展而不断变化。与此相反，在大多数冒险类游戏中，音乐只是一种单调的背景音效。你是怎样处理这款游戏的音乐特点的呢？

答：我们认识到，音乐对于游戏的结构极其重要，因此，必须找到合适的作曲家。我们找来了 Elia Cmiral，一位才华横溢的捷克籍电影作曲家。随便提一下，他不是一位游戏玩家，以前也从未给电脑游戏谱过曲，我想直到现在，他从未玩过一次电脑游戏。我们将游戏处理成一段故事，分成若干场景；当他认识到，在同一个故事中可以出现相互矛盾的场景时——比如在一种结局中，Cath 被刺身亡，而在另一结局中，他逃过这一劫难，继续在故事中活动——他便能轻松地谱写不同的变调（Elia 后来曾为 Ronin、Stigmata 和其他一些影片谱过曲，并取得成功，成为一名好莱坞作曲家）。

事实上，尽管常见的情况是，作曲家总是希望增加音乐的分量，他们会压低音响效果，以增强乐感，但 Elia 对音乐的作用极为克制。对于某些场景，我认为应该配上富于戏剧效果的音乐，或至少有所渲染，但他却说：“不行，这样太俗气，不如略去音乐。”因此，他其实是在减少音乐分量，以便将其应用于能真正发挥作用的地方。在 Last Express 中，我们没有使用装饰性音乐；不会有那种重复的背景音乐伴随玩家们的行动。在这款游戏中，真正的音乐是列车的噪音。当周围环境发生微妙的变化时，你会感受到一种非常和谐的气氛：一扇门被打开，列车噪音加大，或者听见关门声、或者听到远处隆隆的雷声、或者听到列车到站时放慢速度的声音。所有这一切都安排在前景音乐的声道上，因此，当音乐确实出现时，就会显得极为突出。在一些戏剧性的场景和插入动画中，我们以电影手法配上了音乐，试图细致地刻画不同的角色和不同的场景。由于女主角 Anna 是一位小提琴演奏家，因此 Elia 在音乐中加入了一个器乐主题。在游戏的第二天，一连几个小时 Anna 都在自己的车厢里练琴。穿过列车，就能听见她正在演奏巴赫的组曲，时而调一调音，然后继续演奏，如此反复不断。小提琴乐曲的主题也是 Anna 的性格





基调，而且，随着故事的发展，这种性格基调会在不同的场合以不同的形式反复出现。



问：这款游戏如果关掉声音的话，的确是让人玩不下去。

答：毫无疑问，如果失去声音，游戏将大为逊色。在 Last Express 中，声音不仅仅限于对话。如果没有环境音响的变化、没有音乐、没有音响效果的暗示、没有靠近去才听得清的谈话声以及无论靠多近也听不懂的外语对话……所有这一切都是体验 Last Express 时不可或缺的因素。这一点很有意思，因为人们注重的往往是图形效果。不过，我们在声道制作上运用了一项更富于技术创新的手法。大多数人对此毫无觉察：事实上，我们采用的是六声道技术，即从 CD 碟片上同时播放出六声道信号，并动态合成这些信号。例如，你可以听见列车周围发出的各种声音，如开门声、两个人的交谈声、远处隆隆的雷声以及从最近一次插入动画延续下来的音乐声等，所有这些音响效果都在同一时刻播放出来。这确实创造出一种极为丰富的声音背景。



问：此外，与其他冒险类游戏不同的是，Last Express 为玩家提供了一种颇具非线性色彩的经历，这样玩家似乎就能以不同的方式抵达故事的结尾。你是否认为在冒险类游戏中非线性占据着重要地位？

答：是的，至关重要，否则就不能称其为游戏。我打算沿袭的游戏模式有两种：在第一种模式中，玩家必须采取某种行动才能抵达下一个片头动画，否则故事便无法继续下去；另一种模式为分支结构，即“自选冒险”风格，在这种模式中，故事可以有十种结局，如果尝试所有方式，就能体验每一种结局。在 Last Express 中，我认为最成功的谜题序列之一是开头那一段：玩家遇到 Tyler 的尸体后，必须设法将其清除。有几种同样可行的解决方案，但每一种都有其自身的缺点，都会在情节的发展中激起波澜。例如，如果将尸体藏在床上，当列车员过来收拾床铺时，事情就会败露，于是，玩家就必须设法应付这一局面。为了避免这一问题，玩家可以

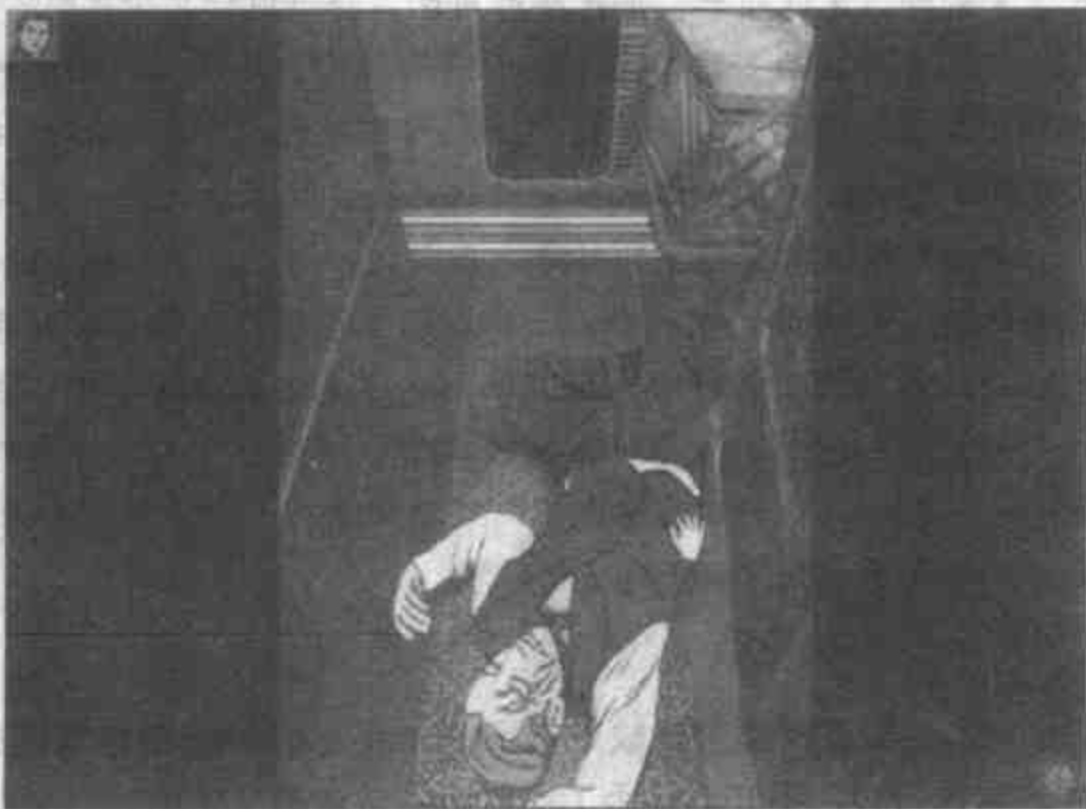


图 18.10 The Last Express

将尸体扔出窗外，可是如果这样做的话，警察就会发现尸体。他们会在下一站登上列车，依次进入每一节车厢检查护照，这时，玩家就必须设法躲开警察。无论采用哪一种方案，你的行为都会对周围的乘客造成影响。另一个例子是，如果将尸体扔出窗外，也许你会偷听到那个名叫 Francois 的小男孩对他母亲说：“嗨，我看见一个人被扔到窗外去了。”然后，母亲对儿子说“闭嘴，你这淘气鬼，不要胡说！”





问：我可一点儿也没注意到。

答：在这款游戏中，这样的细节比比皆是。



问：因此，这就是你不喜欢其他冒险类游戏的原因了，即它们拘泥于“primrose path”的风格，是吗？

答：某些冒险类游戏也会有一些巧妙之处，但从整体感受来看，它们很难保持这一水准。在 Last Express 中也存在着一些不尽如人意的地方，即达不到这种尸体处理谜题所选定的标准。拆除炸弹就是我感到不太满意的地方。你必须咬紧牙关，步步紧跟，别无其他选择。这一谜题处理得并不高明。不过，话又说回来，首先应该考虑的是故事的整体性以及玩家对整体经验的满意程度。



问：我曾听到不少冒险类游戏的设计者说，为了提高故事叙述的生动性，必须限制玩家们的选择，迫使他们选择一条特定的路径。你是否赞成这一观点？

答：的确如此，只是在限制玩家行动的方法上各有不同而已。在 Last Express 中，一个极为明显的限制是，玩家们无法走下列车。一旦下车，游戏便会终止。取得胜利的惟一途径是呆在列车上，直到抵达君士坦丁堡。因此，在这一意义上，它仍然属于线性故事：玩家处于列车内，无法离开。不过，即便如此，在列车内，也应该尽量给予玩家活动的自由。某些门必须始终保持关闭，因为它们会对故事造成过多的影响，而且它们也无法让我们得到一个理想的结局。如果你拿着枪，穿过列车，杀死所有角色，会出现什么情况呢？我们决定不让玩家达到这一目的。因此，这里存在着一种确定无疑的平衡规律。给玩家提供的选择越是古怪、疯狂，就越会限制故事的复杂性和感染力。不过，如果希望精雕细刻，保持叙述的中心地位，而且叙述范围又非常广阔，那么，为了确保玩家继续玩下去，就必须关掉一些车门。

每一款游戏都会以不同的方式应付这一挑战。在 Last Express 中，列车主题提供了保持故事发展所必需的比喻。我认为，一旦人们觉得自己正在乘坐列车，时间正在流逝，每到某个站口之前必须采取相应的行动，而且他们必须抵达君士坦丁堡，否则就无法触及故事的结尾；一旦他们接受这些情况，故事就大功告成了。关键在于为每一个特定故事的有效叙述机制找到和谐点。适用某个故事的叙述机制不一定适用于另一个故事。如果打算制作一款非列车环境的游戏，我就不会考虑应用 Last Express 引擎。



问：由于冒险类游戏缺乏市场，Last Express 的销售情况似乎不太理想。但冒险类游戏也曾非常流行过。不知你是否了解冒险类游戏的全体玩家们受到了什么样的影响？

答：问得很好。必须承认，当我意识到冒险类游戏的市场不复存在时，我惊诧不已，因为对于游戏类型我确实从未做过深入的思考。即使在制作 Last Express 时，不管对不对，我也没有考虑到这一点，但事实上 Prince of Persia 属于动作游戏而 Last Express 属于冒险类游戏。出于多方面的原因，作为一名游戏玩家，我自己也并不热衷于冒险类游戏。在这类游戏中，图形效果往往不尽如人意，故事情节随意、做作，以我心目中的电影或小说故事来衡量，其角色和情节往往毫无根据。





因此，在制作 Last Express 时，我试图加入一些我认为在大多数冒险类游戏中无法找到的特点。因此，作为一名玩家，我觉得自己必须承担拒绝支持冒险类游戏市场的部分罪责。我低估了游戏市场按不同游戏类型划分层次的程度。显而易见，人们将自己看成动作游戏玩家、战略游戏玩家、角色扮演游戏玩家或者冒险类游戏玩家。我从不以这种方式购买游戏，但我猜想，在 90 年代开头那几年里，就连一些电脑游戏出版物也开始将游戏分门别类。然后是开发商和销售商店，我想当时我并未认清这一趋势。



问：这么说，你并不了解冒险类游戏市场不景气的原因了，是吗？

答：是的，作为一名玩家，我只注意自己的感受。在 Scott Adams 流行的日子里，我喜欢玩冒险类游戏，后来又对这种游戏有些厌倦。我认为冒险类游戏的制作者们不应该总是问：“市场情况如何？”我想该问的应该是：“为什么人们不再喜欢玩这类游戏。”也许问题在于游戏本身。



问：你最初的两部作品 Karateka 和 Prince of Persia 都是自己单独完成的，即你承担了设计、写作、编程乃至美工等所有工作。你认为和大型设计组一起制作 Last Express 与单独工作有何不同？

答：比单独工作更加令人激动，更加有益，因为这样你就有机会和一大群有才能的人合作，这些人不仅富于献身精神，而且在各自的专业领域出类拔萃。在我的职业生涯中，这是最令人感到兴奋的经历之一。当然，美中不足的是，大家总是在为下一个月的工资而发愁。在以往那些日子里，非常令人舒心的一点是，开发一款游戏的成本微不足道。2000 美元一台的计算机和五张软盘基本上就是一款游戏的所有成本。但对于大型项目来说，就必须面对预算和期限的压力。

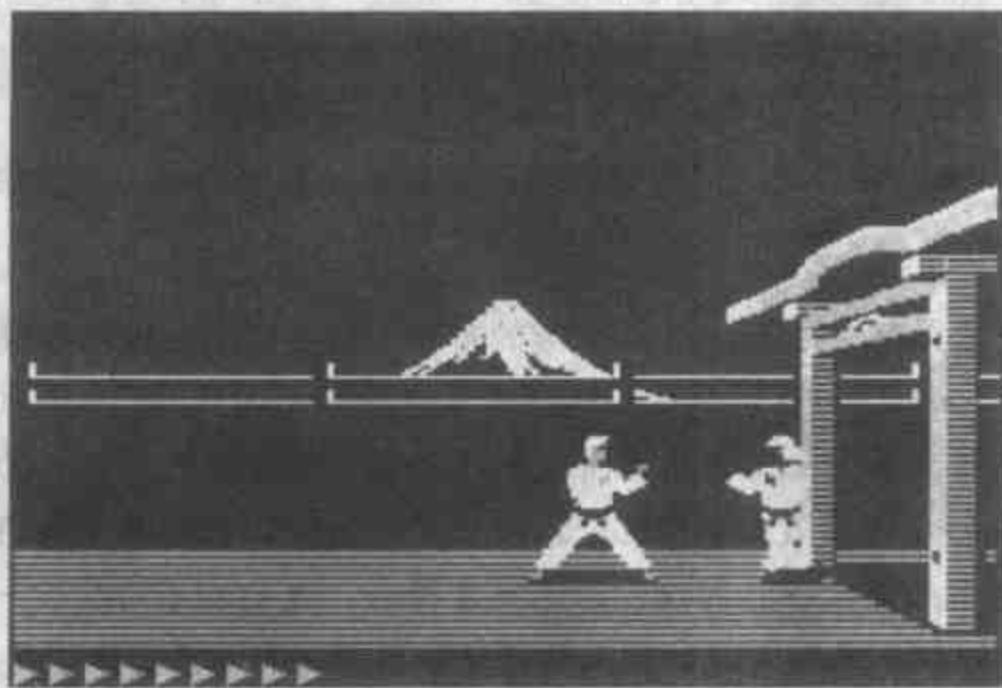


图 18.11 · Karateka



问：至少对于商业性游戏来说，电脑游戏似乎是仅有的从个人创作为主转向集体创作的艺术形式之一。你认为这最终会对游戏造成什么影响呢？

答：这一点很有意思。我目前从事的工作，即编写电影剧本，更容易让人联想到程序编辑，





而不是我曾经做过的其他事情。与程序编辑一样，编写剧本基本上也是将自己关在一间除了一台计算机外别无他物的房间内。你必须从头做起，一旦写出的剧本投入拍摄，那么，与当代电脑游戏制作一样，就必须汇集高度专业化且技术熟练的人员，建立一个完整的创作队伍，然后，数百或更多人将凭借各自的专业知识展开集体创作。无论影片耗资多么巨大，制作过程多么艰辛，它的成败最终取决于剧本的优劣，而剧本通常在数年以前就已完成。同样，当代电脑游戏也是一种集体创作，不过，它的预算资金极其有限，采用每天结算的方式；通常，生产商会要求创作者按期交货。此外，电脑游戏的成败取决于设想，即最初构想的优劣，而这种构想通常在投入制作之前就已形成。一旦投入制作，就根本没有时间改动原先的游戏构想。



问：但这就会限制某些游戏设计者取得成功的可能性，是吗？在1982年，如果设计者需要对整个项目进行重大调整，以便找到理想的设计方案，他们会比当今的设计者做得更加出色。如果是现在，他根本无法进行这样的工作。

答：因为在当时设计者们根本不可能制作耗资数百万美元的作品。我想任何人都有能力构思和编写出像 Tetris 这样的游戏。如果他们需要花费一年时间来确定一套完美的设计规则，以使游戏具有无穷的吸引力的话，这将是一件轻而易举的事情，费用也不会太大。可是，如果项目以3D引擎为基础，而且一共有40名美工在你的指挥之下，你就不应该在半道上冒出这样一句话来：“唉，现在我知道这款游戏真正缺乏什么了。要是在一年以前想到这一点就好了。”

游戏行业正处于一个相当艰难的时期。很难说从事游戏开发能创造出丰厚的经济效益。不过，我认为，对开发商来说，做到这一点倒有一定的可能，但即便如此，能取得成功的也寥寥无几。这是一个令人手足无措的时代，因为成功的游戏实在太少，而这些游戏的创作规模比以往任何时期都要大。如果你能出版像 Myst 和 Tomb Raider 这类销量达两三百万套的游戏，那就好说了：即使其他十款游戏一败涂地，你也能生存下去。但如果你是一个小小的开发商，只有一款游戏投入制作，就像 Smoking Car 的情况一样，你就必须做到一举成功。每年只有一小部分游戏能达到50万套以上的销售量，只有达到这一数量，才能证明投入的资金是值得的。



问：为了制作一款具有 Last Express 水准的游戏，确实有必要投入数额巨大的资金吗？

答：我想在制作 Last Express 时，为了达到实际取得的画面效果，投入的资金远远超出了预定的数值。我们节省了大量资金，说服雇员在低于一般工资水准的情况下为我们效劳，或推迟发放工资的时间；在影片的拍摄上我们也并未耗费多少资金，我们雇用的都是一些非工会成员的演职人员，也来聘请知名人士。因此，为了节省资金，我们几乎想尽了一切办法。可是，受项目的性质、游戏的规模、参与人数以及制作时间等因素的影响，最终还是耗费了大量资金。



问：如果你不介意的话，请问这款游戏耗费了多少资金？

答：大约500万美元。





问：开发工作耗时四年，这是你最初的打算吗？

答：比预定的时间多花了两年。

问：为什么拖延了这么长时间呢？

答：工具开发是原因之一。为了开发出专用的 rotoscoping 技术，我们不得不进行大量的试验，还有各类服装、化装以及使其达到理想效果的处理方法。此外，还有 3D 造型；模型结构庞大，包括列车的外表和内部，需要处理的图像不计其数。3D 造型和处理、动画以及工具开发大大超出了原定的工作限度。事实上，影片拍摄本身并未超出原定的时间和预算，这一部分工作比较容易。

问：现在看来，你是否觉得应该设法在较短的时间内以较少的资金完成这一项目？或者你是否认为完全有必要花费这么多时间？

答：对我个人来说，我因制作 Last Express 而在财务上遭受了不小的损失。因此，在某种意义上，这也许并不是一次明智的举动。我们的投资者也曾希望这款游戏能卖出 50 万套，但最终未能如愿，我对此感到非常内疚。这有点像购买到一张极其昂贵的彩票。

另一方面，我为这款游戏而感到骄傲，这是一件值得高兴的事情，而且我不认为当初能以更少的资金完成这款游戏。我对完成后的游戏感到非常满意。当然，理想的情况应该是设计出一款更小的游戏。如果一开始我们就对情况了如指掌并声称：“很好，这要花费四年时间和 500 万美元，”那么，世上就不会有任何开发商愿意进行尝试。连我自己也不会！不论好坏，在软件行业中，人们大都沉湎于某种程度的自欺心理，认为自己早上一定可以从床上爬起来。即使某一款游戏需要两年时间才能开发完毕，但一开始开发商和营销部门便相互通气，声称可以在一年内完成，圣诞节前即可交货。对任何项目来说，在技术上越是自负，就越难以认识到自己的处境。

与此相反，在电影行业，人们能较好地把握拍摄预算和日程事宜，拍摄时间视需要而定。但另一方面，他们往往不会尝试全新的事物。当他们做出这种努力时，如首次采用一种新的技术，去战争肆虐的国度拍摄外景或在海上拍摄等，即使拍摄时间像制作电脑游戏一样漫长，他们的预算和日程也不会受到什么影响。Last Express 的整个制作过程都与 rotoscoping 这种新技术的开发密切相关，因此，在某种程度上，当我们一开始便声称“好的，我们会在若干个月内以若干资金开发出这款游戏”时，我们基本上是在一种盲信的指导下展开工作，想当然地认为自己能解决任何实际问题，而且将无往而不胜——最终事情确实如此。当我们第一次尝试某项工作时，很难制定出精确的时间和成本计划。在 Last Express 中，我们同时尝试了大约十种以前从未做过的事情。也许这并不是明智的做法。

总之，对开发者来说，不切实际的计划无法取得希望之果，也不会有助于我们的工作。我对这一项目感到遗憾的事情之一是，我们每天都要面临财务上的巨大压力，我的一半时间在为游戏而担忧，另一半时间则在为筹措资金而焦虑。承担这一雄心勃勃的项目后，我们就陷于这样一种窘境之中。





问：Last Express 是你第一次不承担编程工作的独立项目。你是否对这项工作念念不忘？

答：编程工作的一个美妙之处是，当你真正时来运转之时，可以将自己关在房间里，体味日积月累的满足感——一切都是你和机器的功劳。我对这项工作最感怀念的时刻，通常是当我花费两天时间参与背靠背会议之后。为什么要举行这样的会议？为什么我必须参加这样的会议？我们有四位程序员在为 Last Express 这一项目效劳，尽管我常常羡慕他们的幸运，但我也正忙于自己的工作，如游戏设计、剧本编写，与美工和动画师配合、给演员安排角色、在录音和拍摄时对演员进行指导、同作曲家及音响设计师和编辑合作等，这些都是我真正乐于从事的工作。有时候，我也会给程序员提供协助，但由于我原先掌握的编码技术是 6502 汇编语言（适用于 Apple II 机型），他们认定我确实帮不了他们。



问：无论在场景设置方面还是在游戏设计方面，Last Express 都是一款极为独特的游戏。相比之下，最近问世的其他游戏似乎大都以幻想或科幻故事为背景，而且它们都建立在去年的热门游戏之上。你对游戏行业的这种“跟潮”趋势有何看法？

答：除了少数不同凡响的游戏之外，我认为你对大多数游戏的看法完全正确。我不能确定这种“跟潮”问题是否主要存在于游戏背景方面。我觉得它更多地存在于游戏类型方面。比如我们可以参考 Doom 这款游戏，对它的纹理进行修改，绘制出一辆 1914 年的特快列车，但我觉得游戏行业真正需要的并不是这种创作方式。对我来说，更为重要的是对游戏设计本身的探索，如采用什么样的游戏结构，玩家的实际行为是什么，努力创造出一种新的、有效的游戏形式。当开发商的股票价格不会因这种探索的成功或失败而发生起伏时，这一工作要容易得多。如果探索的内容是一种公认形式的延续或变异，它无疑将得到更加热烈的支持。越是难以描述或解释的新形式，就越难以找到愿意为其承担投资风险的个人或者公司。不过，我并不知道应该如何处理这一情况。这是一种周而复始的现象，它的结果几乎是不可避免的；当我们去电脑商店寻觅下一款热门游戏时，说实在的，我们希望找到什么样的游戏呢？我们更加关注的是那些在推销上大做文章的游戏，那些我们在杂志广告上经常看到的游戏。于是，那些未经过大肆吹捧的游戏将难以占据较大的市场份额。因此，在某种意义上，作为消费者，我如愿以偿。但作为游戏设计者，我确实感到失落。



图 18.12 The Last Express

我有一位以拍电影为职业的朋友，他过去总是说：“唉，伙计，我真羡慕你们这些制作电脑游戏的人。你们有机会从事真正富于独创性的工作，而在好莱坞这儿，他们想做的只是改编



去年的续集。”有趣的是，不知怎么回事，如今游戏行业也出现这些电影制作者们抱怨多年的问题。也许情况还要糟糕。随着制作成本和市场的膨胀，以及颁奖仪式的日益奢华，我们也陷入了拘泥于原有风格的困境，取得突破性进展变得更为艰难。



问：因此，你更多的还是感到失意，是吗？

答：我想是这样的。我认为每一种艺术形式都会经历相应的发展阶段。在电脑游戏领域，我们度过了早期的狂热阶段，如今我们正处于发展时期的困难阶段。这并不是说创新活动已经停止。即使在具有百年历史的电影制作领域，无论是由于社会变迁还是技术变革，每两年就会出现一部几年前无力制作的影片，这就是一种意义重大的进步。我们必须排除数百部千篇一律、平庸无聊的影片，才能发现这些极少数的艺术精品。我认为，在电脑游戏行业，我们正处于同样的情况。每年，在数百款新出笼的游戏中，总有那么两款以新颖、独到的方式给这一行业带来活力。我们应该尽力保持这种努力，告别那曾一度辉煌的岁月，因为它们已经成为过去！



问：你对 Prince of Persia 3D 项目的投入程度如何？

答：我的工作仅限于在开始时给他们下这行动命令，并在工作期间偶尔向他们提出建议或提供创作参考意见。这是 Broderbund 公司的项目，由制作人 Andrew Pedersen 发起。整个项目由他一手经营。他组建了设计小组，并为它操劳两年之久。因此我对这款游戏并无功劳可言。



问：将一款 2D 游戏改造为 3D 游戏是一项极其艰难的工作，因为必须让玩家们具有全方位的活动能力。

答：这的确是问题所在。将 Prince of Persia 转化为全方位 3D 游戏时，我们面临的一个问题是如何使操作保持简便。另一个问题是如何让玩家了解自己所处的环境，因为他只能看见眼前的情况。举个粗略的例子：你跑向峡谷边缘。如果具备侧视功能，你就能观察一下周围的环境，摸清应该采用三级跳还是四级跳，以及是否能跃过去。如果峡谷太宽，你就会知道根本不必尝试。可是，在一款全方位 3D 游戏中，除非进行尝试，否则你就无法确切了解峡谷的宽度。即便如此，当你跌下去时，你还会疑惑不解地问：“是没有靠近峡谷边缘，还是起跳方向不大正确？”于是，游戏的类型因此而彻底改变。那种身临其境的感觉得到加强，当然，环境的丰富性也有所提高。不过，我认为，这也是损害游戏的纯粹战略因素。



问：这么说你认为让每一款游戏都采用 3D 图形并不一定行得通，是吗？

答：是的，你必须将实时 3D 图形技术与某种特定的界面区别开来。我认为，实时 3D 图形引擎还有很大的改进空间。显然，第一人称射击游戏 Doom 是 3D 游戏的原型，它所引导的潮流持续了两年时间。随后，Tomb Raider 和 Super Mario 相继问世。Prince 3D 也属于这一类型。因此，我认为重要的是采用一些新的方法，以电影形式来表现游戏动作。这样，既保持原作的趣味性，也不失新型 3D 游戏的所有视觉效果。我认为有待探索的领域非常之多。在 Prince 3D 中有几处颇为精彩的地方，我希望能对这些地方做出重大改进，为 3D 动作游戏创造新的成就。





问：我从一些报刊上了解到你对 Tomb Raider 非常欣赏。这似乎促使你将 Prince of Persia 转化为 3D 环境，创造出一些新颖、刺激的游戏效果。

答：我认为关键在于新颖这个词。的确，作为一名玩家，我对 Tomb Raider 极为热衷，这是因为我以前从未见过这样的游戏。不过，我觉得，那是已经完成的东西，现在我们能更加清楚地看到这类游戏的优点和不足。假如现在重新制作 Tomb Raider 这款游戏，就应该找到一种能超越 1996 版的设计方法。我们不能反复沿袭原有的东西。

问：那么，在 Prince of Persia 3D 中，你是否针对 3D 空间导航提出过一些有效的解决方案呢？

答：我想，Prince of Persia 3D 在武器和动作数量方面略嫌复杂。这并不是由我独立设计的游戏，他们的目标是吸引一个特殊的玩家群体。我认为这一群体的核心是那些比原作玩家更加狂热的人们，这一点他们也有所了解。

问：你是否觉察到在某个项目的开发过程中你的游戏设计方案出现大幅变化？

答：在设计 Karateka 和 Prince of Persia 的过程中，我曾经体验过这种幸运，即让游戏随着时间的推移逐渐完善，因为这两款游戏均由我一个人关在房间内用一台计算机设计完成，不必为预算和公司效益担忧。原

想 Prince of Persia 可以在一年内完成，但结果却花了三年时间，不过情况还不错——达到了预想的效果。

Last Express 的不同之处在于，它是一个规模浩大的项目。由于采用的是由数百人建造的设备和网络计算机，时间显得极为宝贵，因此，中途改变设计方案不在考虑之列。项目开始时，我花费了大量时间尽可能制定出详细的设计方案。我只能

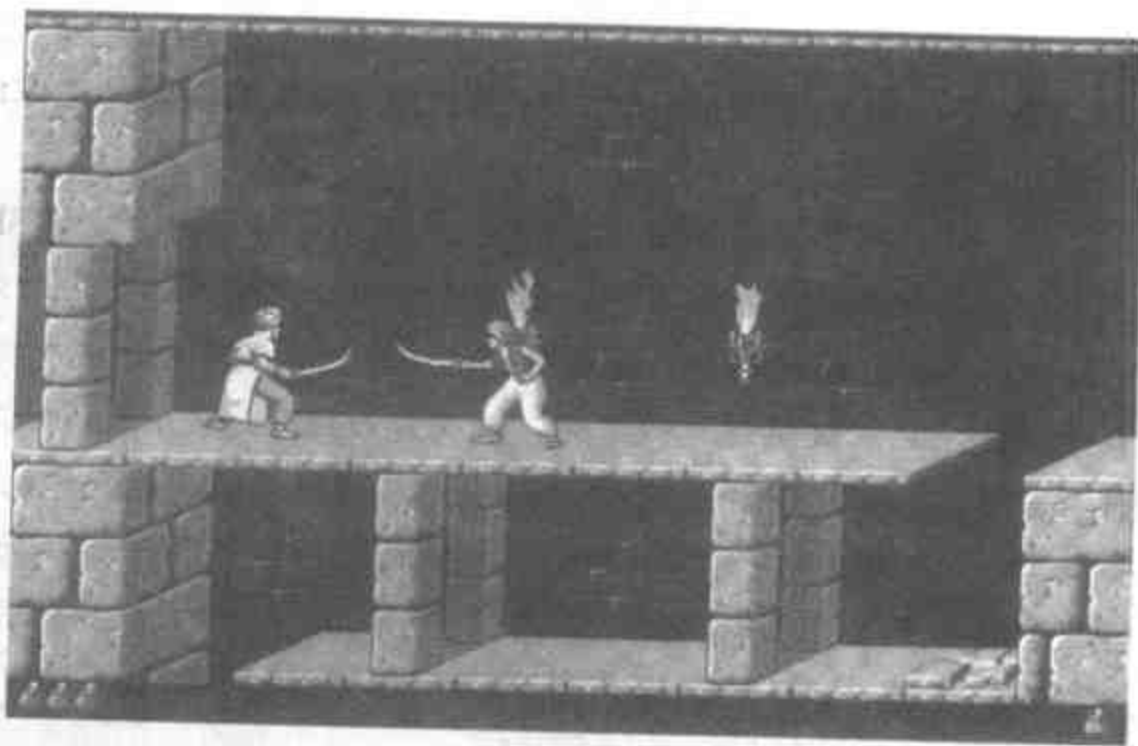


图 18.13 Prince of Persia

暗自祈求最初的设计方案切实可行，没有什么严重的缺陷，而且自始至终也不会出现这些缺陷。

问：不过，你的早期游戏在开发过程中都出现过显著的变化，是吗？

答：是的。举个例子：原先我并不打算在 Prince of Persia 中采用格斗内容。在设计过程中，我想出一个绝妙的主意，即回应许多电脑游戏中千篇一律的暴力场面。我决定让玩家在布满荆棘和陷阱的险恶环境中充当一名手无寸铁的无辜者。玩家将面对许多暴力行为，他的任务是避



开这些暴力攻击，因为他无法真正与之抗衡。这一方法也可用来解决另一个问题——我认为已经没有足够的内存让另一个角色同时在荧光屏上活动。幸运的是，一些忠实的朋友坚持不懈地敦促我添加格斗内容。如果朋友们对你说你的游戏枯燥无味，你最好听一听他们的意见。

那个叫 Shadow Man 的角色是一个意外的收获，因为当时我认为：“内存不足，没有办法再添加另一个角色了。”除非这个角色与公主长得一模一样，而且采用同样的动画图像。我不记得这是谁出的主意：把角色稍稍改动一下，然后再将改动后的结果与角色本身并列在一起，这样，就能获得一个黑色的鬼影，它具有微微泛白的轮廓。于是，我尝试了这一方法，当我看见 Shadow Man 在屏幕上来回跑动时，我说：“好极了，又多出一个角色。”由此又设想出一种贯穿整个故事情节的装置，即那面镜子。穿过镜子，就会产生一个邪恶的自我，他会时刻跟随着你，妨碍你的行动，如当你打算开门时，他会紧闭大门，或者往你的头上扔东西。此外，还必须注意，当你最终和 Shadow Man 交手时，你不能杀死他，因为他就是你自己，如果杀掉他，你自己也会死去。因此你必须找到一种解决方案。你可以称之为荣格（Jungian）分析法，或别的什么名称，但这一切都归功于内存不足这一事实。



问：后来，为了安排其他角色，你一定设法获取了更多内存。

答：我们将已经完成的部分重新设计一遍，使其更加小巧，运行更加迅速，尽管这期间我们本应投入编程工作，就像在一台计算机如 Apple II 机上给 Prince of Persia 编程一样。最后，这一部分确实得到改进，于是，我们获得足够的空间来应付卫兵们有限的角色阴影。如果细心观察，就会发现许多卫兵们无法完成的动作。他们无法跑动、跳跃和追赶玩家，他们只能格斗。



问：你的游戏都具有很强的视觉感染力。不知你是如何在视觉感染力和游戏要求之间做出权衡的？

答：我认为，除了我们已经讨论过的一些因素，如操纵的简便性和界面的一致性，还应考虑视觉效果，在这一点上我们往往很容易做出让步。你会盘算：“嗯，可以在这儿插入一个菜单栏，在屏幕右上角可以加入一项数字显示，表示玩家喝饮料的次数。”等等。简单的解决方法总会产生一些副作用，影响游戏外观。因此，我始终恪守一个基本规则，即屏幕整体布局必须做到美观、醒目和简洁。这样，那些不打算玩游戏的人走进房间并看到其他人正在玩时，就会被美观的画面结构所打动，他会驻足观看一分钟，心想：“画面不错，就像在看电影一样。”这确实会促使设计者们努力为目录等问题寻找最佳解决方案。我们不能一开始便接受那种显而易见的解决方案，我们必须在自己限定的范围内努力寻求问题的解决。



问：那么你为什么放弃游戏事业，转而专门从事电影编剧工作呢？

答：我总是在电脑游戏和电影拍摄之间来回穿梭。我想，我们非常有必要为自己在不同领域的创造能力补充能量。如果在完成 Karateka 后的两年里，我并未做过电影方面的探索，没有编写过剧本，Prince of Persia 就不可能具有如此丰富的感染力。对 Last Express 来说，情况也是如此。我在古巴拍完纪录短片 Waiting for Dark 后便立即投入这个项目。因此，我也拿不准，不





能说就此了结。也许有一天我会着手设计另一款游戏，不过，目前对我来说，编写剧本和拍摄一部出色的电影远比制作电脑游戏更加令人神往。我认为，对于那些诱人的项目，应该说服自己放弃而不是投入。

此外，值得一提的是，电脑技术日新月异。如今的电脑游戏与十年以前的大相径庭，谁能说出在下一个十年里我们会取得什么样的成就呢？



问：因此，并不是说你偏好那种更富于线性特征的艺术形式。更为确切地说，你是在交替从事这两种工作。

答：虽然艺术形式不同，但两者涉及的诸多挑战却有着惊人的相似之处。对于电脑游戏，尽管它以非线性方式展开故事，但我们仍然可以从一个特定世界成一些特定角色领略那种令人陶醉、充满神秘色彩的境界，而这正是电脑游戏的动人之处。因此，人们会说：“我喜欢玩这款游戏，我想成为 Mario。”而面对另一款在技术上可能同样出色的游戏，则会说，“我对这一环境中的这个角色毫无兴致。”对电影来说，情况也是如此。在观众和叙述者之间存在着一种神秘的默契：即使根据预告片，观众也能就是否打算接受这一特定的故事做出决定。

当我们埋头工作、煞费苦心地将一系列要素以某种确定的形式组织起来时，就会发现这两种艺术形式并非截然不同。无论从事游戏创作还是电影创作，我们面临的挑战极为相似，例如：采用既定类型并在此基础上努力创新，取得出人意外、充满悬念的效果，以及合理应用现有要素等。在设计 Karateka 和 Prince of Persia 的过程中，最让人头痛的事情是，在日复一日的工作中，面对那些花费一个星期编写出来的程序，说道：“那又怎么样？结果固然不错，但我现在必须抛弃它，必须寻找新的思路。”对编剧工作来说，情况同样如此。在漫长的项目过程中，为了确定这些要素并使其正常运作，应该乐于反复放弃自己的成果。

Jordan Mechner 作品一览

Karateka, 1984年

Prince of Persia, 1989年

Prince of Persia 2, 1993年

The Last Express, 1997年

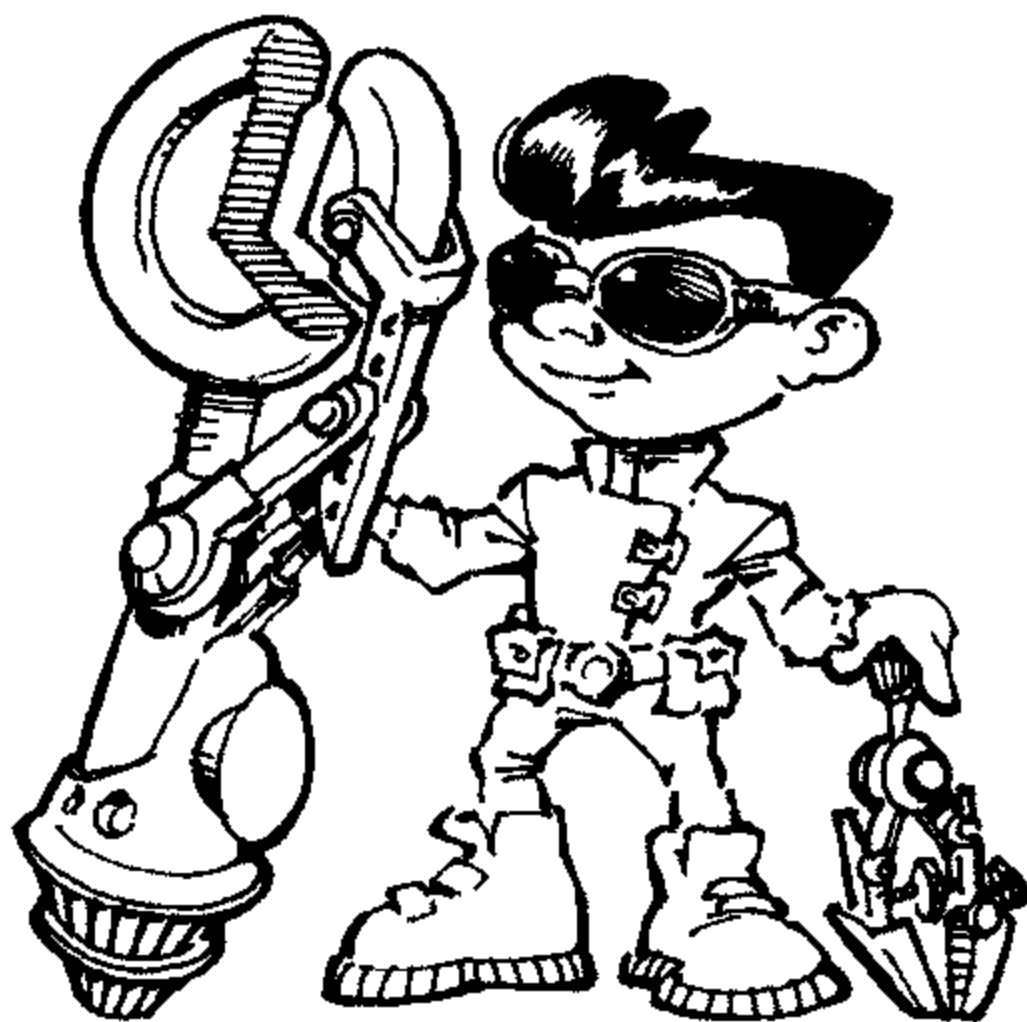
Prince of Persia 3D, 1999年 (顾问)

这篇访谈录最初以不同形式登载于《Inside Mac Games》杂志，该杂志的网址为 www.imgmagazine.com。本文经许可后采用。



Chapter 19

第 19 章 构思设计工具



“人是使用工具的一种动物……没有工具，人类将一无所有；使用工具，人类就拥有了一切。”

——托马斯·卡莱尔

开发一个好的游戏所不可或缺的一部分就是，为游戏创造出富有说服力的内容。为了创造出品质优良的游戏内容，设计小组需要装备一些设计精良、功能强大的游戏制作工具。因此，我们可以得出这样的结论：设计一个好的游戏就意味着构思、设计出好的游戏制作工具。

不同于编程人员编译游戏代码时所使用的开发环境，以及美工设计师设计游戏美工时所使用的图形组件，最常使用的游戏开发工具是关卡编辑器。这种工具跟我以前所提到的那些工具的不同之处在于，它是专门为某一个具体项目而制作的，或者至少可以说，是为开发小组用来强化游戏编程的引擎而专门制作的。开发小组的主要职责是使这个关卡编辑器的功能尽可能强大，有助于关卡设计人员的工作，以便他们创造出最佳的游戏环境。

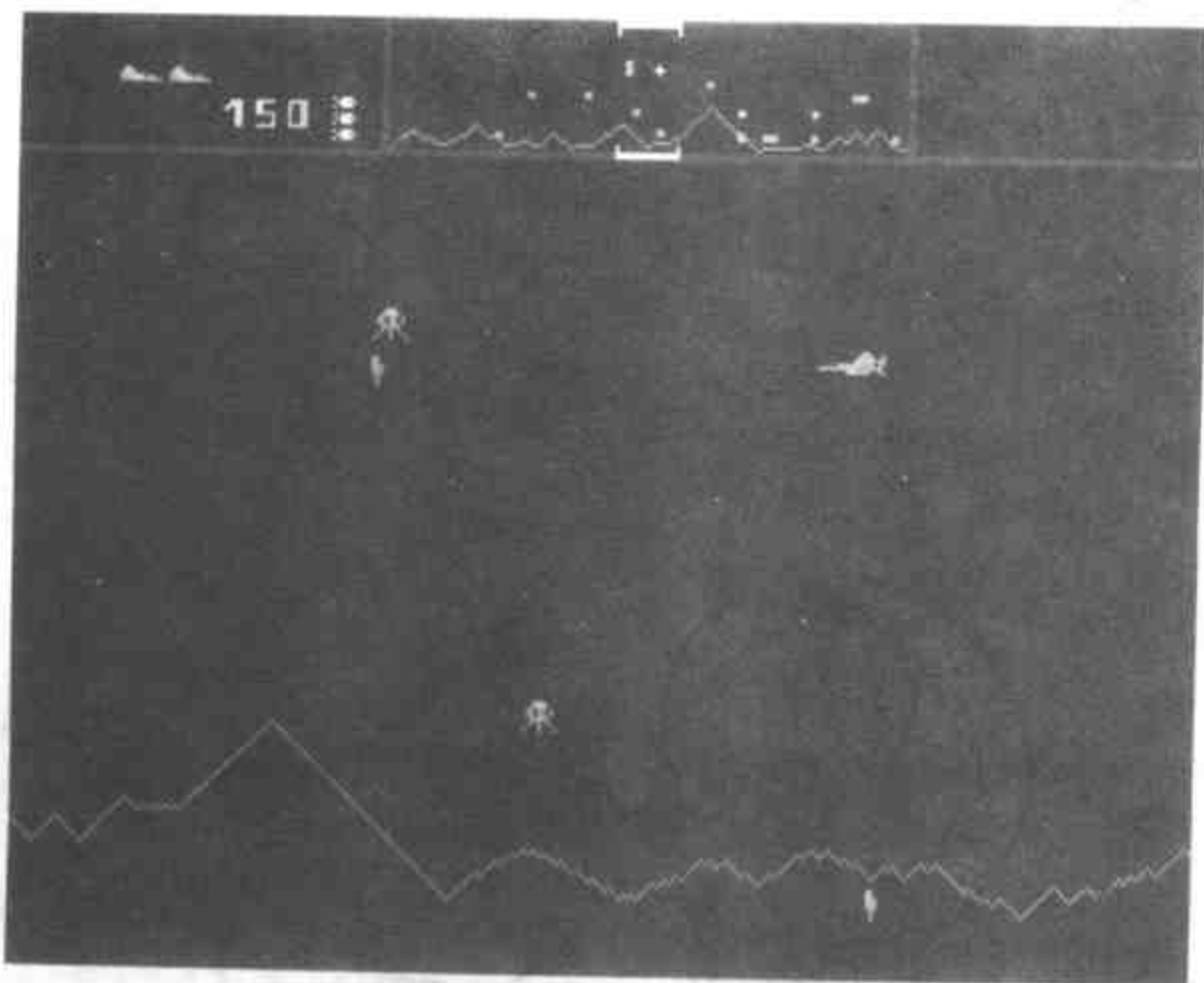


图 19.1 一些早期游戏，比如 Defender，其关卡非常简单，并不需要制作出复杂的关卡编辑器

当然，并不是每个游戏都有关卡。我们现在回想一下，20 世纪 80 年代的许多经典街机游戏，比如 Missile Command 或是 Space Invaders，都没有这样的关卡。那些有关卡的游戏，如 Defender 或 Tempest，创造出其中的游戏环境当然不需要复杂的关卡编辑器。Civilization 和 SimCity 这类游戏会自动生成关卡的主要成分，其余的部分由玩家和计算机 AI 自己完成。体育类游戏的关卡非常容易，大多只是要求建造一些视觉上令人悦目的体育场来环绕整个游戏画面。我会在第 21 章“关卡设计”中更为详细地讨论游戏中关卡的本质。有众多的现代游戏使用了非常复杂的关卡，这些关卡对游戏运行的方式有非常巨大的影响。这类游戏要求游戏开发小组制作出专门的关卡编辑器，供关卡设计人员搭建游戏环境。

令人非常吃惊的是，许多开发小组并没有投入足够的时间来制作最佳的开发工具。通常，整个小组对行业内所使用的其他工具的标准一无所知。经常是，他们对关卡编辑器如何运行并没有花费足够的时间进行事先计划和透彻的设计。上述因素的结果是，要做到关卡编辑器能够勉强使用就要花费好几个月的时间。编程人员常常在业已排得满满的工作日程之外，陷入不断发挥、改进关卡编辑器功能的工作，而被迫采用“地狱式的代码编程”方法编写代码，以便按时完成任务。一些关键的省时功能往往在整个项目过半时才被加入，这时，游戏设计人员已然绝望地落下了一大段的工作。



19.1 所希望得到的功能性

那么，关卡编辑器应具有什么样的功能性呢？许多人也许会建议说，任何关卡编辑器的一



个重要部分应该是，所有重要功能都有与之相对应的热键。另有一些人极力建议说，应该有足够的配置设置，需要某些功能时，不同的设计人员可以打开或关闭自己选择的功能。不用说关卡编辑器还要工作稳定，设计人员能够工作许多小时而不会锁死，而且这些建议所讲的都是些非常明显的问题，这些问题只是一个实用的编辑器所应该达到的底限。那么一个出类拔萃的编辑器应该具有什么样的功能，才能使设计人员更好地工作呢？

19.1.1 关卡形象化

游戏环境设计工具的一个重要目标必须是，设计人员能够亲眼看到他所设计的环境，同时又能够对游戏环境进行修改。这在文字处理及桌面出版领域通常被称为“所见即所得”，但并不是所有的关卡编辑器都能很好地做到这一点。我把这种所见即所得的视图称为“玩家的视图”，因为它代表了玩家在游戏时所看到的场景。设计人员所精心设计的环境，应该能够通过游戏所使用的同一个渲染引擎，在玩家的视图窗口中得以展现，或以 2D 的方式，或以 3D 的方式；或者采用子图形或模型的方式；或者是软件驱动，或者是硬件加速。这似乎应该是任何一个关卡编辑器最为重要的功能。试想，设计人员如果想要创造一个漂亮的游戏环境，他怎么能够先调节编辑器中各种环境的设置，然后再运行另外一个程序才能看到自己设计的东西到底是什么样子？

设计人员能够非常容易地在玩家视图中移动摄像机，在设计关卡时可以迅速地调整到所需的位置。这种移动的最佳方式可以通过一种简单的“飞行”模式来实现，玩家可以简单地使用移动控制键和转向控制键来控制摄像机的位置。在这种模式下，摄像机移动时不能与游戏中的几何模型或其他物体冲突。也许有人还想为玩家视图提供一种特殊的模式，在这种模式下，设计人员可以在游戏环境中自由调整摄像机的位置，就像玩家在成品游戏中能做到的一样；但是，如果这样的话，编辑器应该一直允许设计人员无限制地移动。为了更加精确地编辑一个关卡，设计人员必须能够仔细观察他想要观察的东西，而不用担心前面会有棵树挡住视线。

设计人员在编辑器中所看到的，和实际游戏中所表现出来的，二者之间存在的任何差异都会使设计的关卡看起来糟糕无比。假设编辑器中的视图只有使用 3D 硬件加速渲染才能得以展现，而游戏本身除了硬件外还要运行在软件模式下。这会给设计人员带来极大的挫折，因为他也无法说出在软件中的关卡会是什么样子。经过加速后的关卡看起来肯定很棒，但是，如果没有三重线性过滤，各关卡图像纹理的失真将会非常严重。当然，在编辑器中采用硬件加速，这一点无可厚非，因为它比在软件视图方式下工作更为流畅，设计人员也可以更快、更好地工作。但对一些既需要在有 3D 加速卡情况下运行，又需要在没有 3D 加速卡情况下运行的游戏来说，编辑器应该能很容易地切换加速视图与非加速视图；这样，设计人员就能更快、更容易地保证游戏的外观，不管游戏采用的是哪种渲染模式。

当然，游戏中将会出现的环境并不总是编辑这个环境的最佳视图方式。因此，除了玩家视图之外，关卡编辑器还要有一个“编辑视图”。编辑视图通常都采用自上而下俯视的模式，但也会包括一个可旋转的线框视图或者多个视图。最后一个选项对编辑 3D 游戏环境尤其有用。例如，非常流行的、用来编辑 Quake 这款游戏引擎所使用的编辑工具 Worldcraft，同样被用来设计 Half-Life 中的关卡。这种工具为玩家提供了“三重视角”的设置，设计者可以在三个不同的窗





口中，同时自上而下俯视观察（沿 Y 轴），从一侧观察（沿 X 轴），或从另外一侧观察（沿 Z 轴）。三个侧面视图与 3D “玩家视图”可以同时出现。编辑复杂的、有重迭的 3D 建筑物，比如大家在 *Quake* 这款游戏各关卡中所见到的那些建筑物，多视图方式尤其重要。玩家视图与编辑视图相比，前者主要用来向设计人员展示游戏中的关卡最终看起来会是什么样子，而后者的主要目的是使设计人员能够更加容易地修改、构造在玩家视图窗口中所能看见的一切。当然，编辑视图与玩家视图应当能够同时在屏幕上出现，在一个窗口所进行的改动应当立即在所有的视图中反映出来。

在某些情况下，可能并不需要单独的编辑器和玩家视图模式，比如一些 2D 游戏。在我的第一个游戏 *Odyssey: The Legend of Nemesis* 中，玩家视图足能满足编辑关卡的需要。我在设计该游戏的各个关卡时，从来没有希望得到另外一种视图方式。同样，在 *StarCraft* 这款游戏中，游戏所代表的环境十分清楚，设计人员可以直接对其进行修改。因此，*StarCraft* 的战役编辑器只提供了玩家视图窗口供设计人员编辑使用。但是，对 *StarCraft* 的编辑人员来说，如果能有一个单独的编辑视图将会对他们大有裨益。由于游戏采用的是等角视图，有时这种视图看起来会很混乱，设计人员在编辑关卡和放置、操纵各个单元及其他游戏元素时，一个真正的自上而下的俯视视图会大有用处。*StarCraft* 还真有一个正在设计之中关卡俯视角度的“小地图”，可实际上设计人员并不能在这个视图进行修改，而且这个小地图也不够大，修改起来也不会容易。

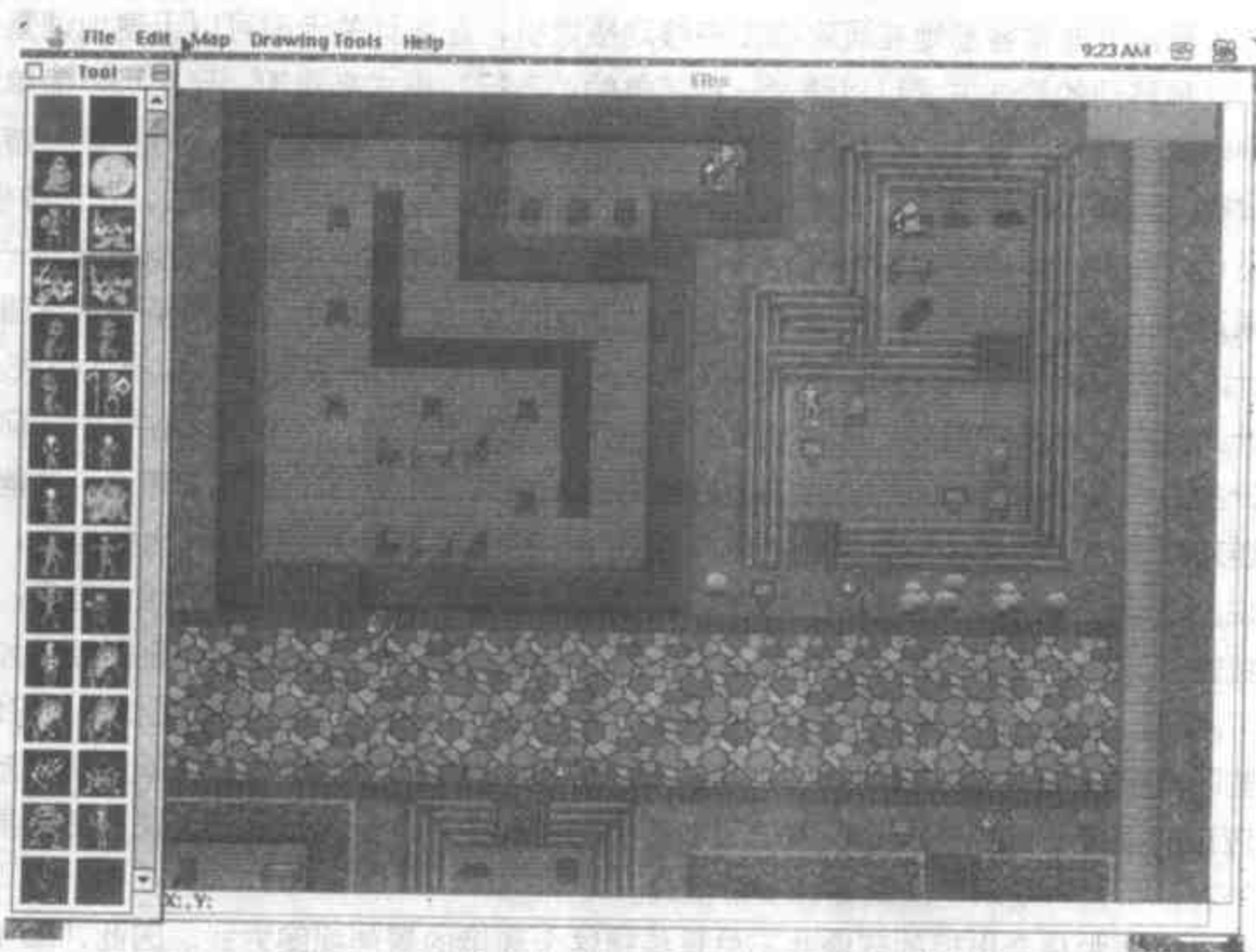


图 19.2 *Odyssey* 游戏所使用的 Zoner 关卡编辑器提供了一个非常完美的视图，适于编辑 2D 游戏环境



19.1.2 大画面

我曾经提出过，游戏关卡编辑器很重要的一点是，设计人员看到的关卡应该与最终游戏中的关卡完全一样，但玩家视图窗口并不总是需要呈现和玩家最终所能看到的完全一致的画面。如果关卡编辑器能同时显示其他各种信息给设计人员，将有助于游戏的开发，这一点非常实用。例如，假设正在开发的游戏中有按事先设定好的路径移动的各种各样大怪物。能够精确地看到这些路径的移动方向，这一点非常关键，它有助于理解关卡的运作方式；同时，清楚地看到这些路径在游戏环境中的移动方向，对于确保路径的正确设置至关重要。

在许多关卡编辑器中，这类功能性信息只在编辑视图中传递，而在玩家视图中却无法做到；但是，更合理的方法是，两处地方都能得到信息的交流。当然，玩家视图窗口中不能都是这类关卡功能性信息，但打开或关闭不同信息交流的功能会相当有用，有助于设置关卡的表现。对 3D 游戏来说，尤其如此。我们还是回到路径的这个例子中，2D 俯视视图或侧视编辑视图中的路径在 3D 视图中最终会是怎样？对于这个问题，设计人员为什么非要在脑海中进行精确的推断呢？这些应该由编辑器描绘出来，而再也不用凭臆想工作了。

在开发 Centipede 3D 游戏时，当时编程人员正在加入一段代码，来阻止玩家沿太为陡峭的斜坡向上爬升。为了调试这段禁止爬坡的新代码，程序员增加了关卡编辑器的功能性，使编辑器可以将组成画面的三角形线条打开或关闭。这些线条依据某个边缘能否为玩家越过而显示不同的颜色。对于太过陡峭的坡度，玩家无法在上面停留，这里的三角形被标上一个红色的 X。编程人员加上这种功能性主要是为了帮助调试禁止爬坡的代码，而从来没有意识到，这给关卡设计人员带来了多大的方便。现在，设计人员可以精确地看到关卡中的哪些地方玩家可以去，哪些地方去不了。一种更好的侧视效果是三角形边界着色渲染，它能产生出一个画面的线框视图，这种功能性在以往的编辑器中还尚未见到过。因为设计人员能够精确地看到哪些三角形组成了哪些斜坡，然后可以相应地修改关卡，这极大地简化了几何上的编辑。新增添的线框视图和禁止爬坡的标记产生的直接结果就是，最终游戏中的几何图形更好、更精确。这种功能性的美妙之处在于，人们可以在编辑器中开关这项功能。因此，如果设计人员想要浏览一下关卡的外观，可以把它关闭；如果他想要观察一下游戏是如何运行的，可以把它打开。

对于路径，如果设计人员也能控制一些对象着色的开关，比如触发器或其他通常不可见的对象，这也会非常实用。同样，这对于显示游戏环境中对象的界定信息大有帮助（这些信息通常跟对象的子图形或模型的视觉组成并不完全匹配），因此，玩家可以很容易地察觉到这些界定信息是如何影响自己和 NPC 在游戏中导航的能力。区分开玩家所能去的地方和不能去的地方同样也非常有用。而且，这些功能性数据的任何部分都应该能够通过热键、菜单或者按钮来打开或关闭，这样，设计人员在解决问题时就多了一种选择，可以查看自己想要的那部分数据。这些数据理所当然地应当在玩家视图中得以渲染，设计人员就可以精确地看到，这些触发器、路径、禁止爬坡或者其他对象是如何被放置在游戏环境中的，而不用在俯视视图中妄做猜想。通过采用真实的视觉视角来观察游戏环境，以显示游戏的运动状态数据，设计人员能像设计游戏的各种属性一样，更好地致力于游戏的感官质量。





19.1.3 跳转至游戏

在一些游戏中，玩家要操纵一个角色通过整个游戏环境，这时，最为重要的是，设计人员能够容易地了解到，玩家通过这些关卡时“感觉如何”。为此，除了提供代表玩家在游戏中所看到的游戏环境的玩家视图外，设计人员还应该能够在这一视图中进行移动操作，就像在实际游戏中所能做到的那样，这一点非常有用。有了这些额外的功能，设计人员就可以测试玩家能否做出一个跳跃动作、沿“S”形曲线通过时感觉如何，以及玩家扮演的角色能否平稳地沿台阶往上移动。在这种“游戏”模式之外，关卡编辑器还应该保留一种无限制的“飞行”模式，这种模式我在前边曾提到过。

Bungie 为 Marathon 游戏引擎所设计的 Vulcan 编辑器，非常适合设计人员的需要。在建造关卡时，设计人员通过它可以测试这些关卡的“感觉”。Marathon 所采用的技术跟 Doom 这款游戏的技术很相似，但比 Doom 的要稍好一点儿；这种技术经授权后被许多游戏所使用，其中包括我自己的游戏 Damage Incorporated。Vulcan 编辑器随后进行了修改，改名为 Forge，跟这一系列的最后一款游戏 Marathon Infinity 一同发布。Vulcan/Forge 编辑器允许使用一种“视觉模式”，其作用就像是一个玩家视图窗口。在视觉模式中，设计人员能够在游戏环境中导航，就如同玩家在最终游戏中导航那样。这种模式的缺点是，在这种模式下，除了材质结构及画面的明暗分布外，设计人员不能编辑游戏环境。毫无疑问，这是由于编辑器产生之时处理器运算速度及当时能够负担得起的显示器较小屏幕的限制。但是，Vulcan 编辑器中的视觉模式非常实用，从编辑模式转换到游戏模式的速度非常快，设计人员能够很轻松地进行修改、体验游戏的感觉，然后转回原来的模式，再进行必要的修改。

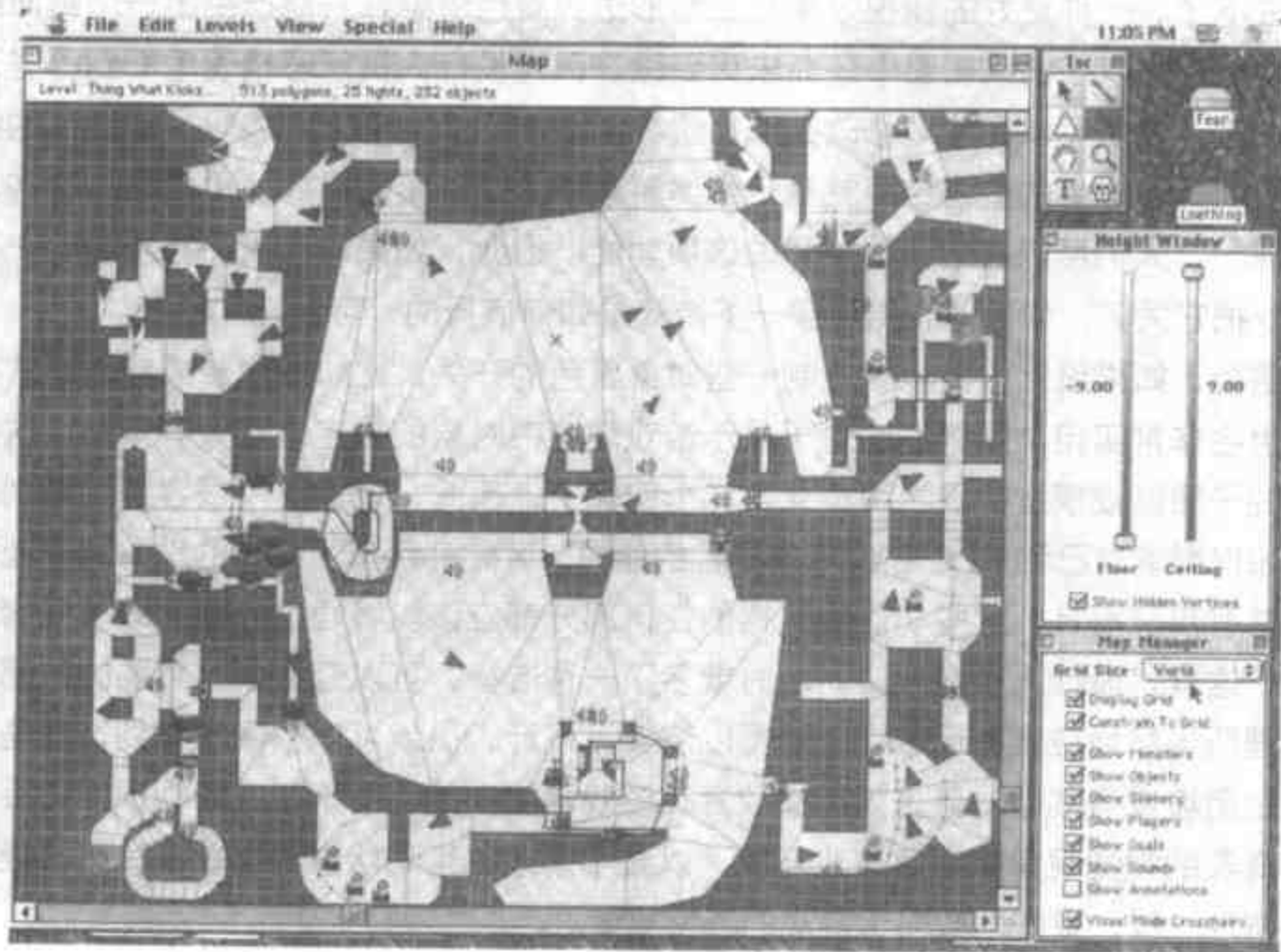


图 19.3 Marathon 游戏引擎使用的、Bungie 的 Forge 关卡编辑器包括有一个“视觉模式”，设计人员可以像玩家在实际游戏中那样，在整个关卡中调整



当然，有人会得出这样的结论：从逻辑上讲，下一步应该是，设计人员能够在游戏模式中进行实际游戏。这样，设计人员就能看到游戏中不同的机制是如何发挥作用的，以及各个对手会给玩家带来什么样的挑战。但这会给编程人员带来许多代码编写方面的困难。如果能让游戏中的对象跟实际游戏中的一样运行，许多对象就要从玩家开始一个关卡时的起始位置开始运行。比如，富有攻击性的巨人会朝玩家跑过去，然后开始攻击。这些移动对象在关卡编辑器中是否也发生实际的运动呢？如果设计人员将游戏保存成另外一种状态又会怎样呢？很显然，这并不是一个好主意，因为各种实体精心放置的位置会变化。设计人员需要的是，能够快速测试关卡中的某个位置，测试完毕后迅速回到“游戏前”的状态。解决这一问题最好的办法是，设计人员能够快速进入一种“测试模式”，然后能够同样快速地退出，迅速回到关卡编辑器的功能状态。这种转换越快越好，因为转换速度越快、越容易，设计人员越有可能喜欢转来转去，来回测试关卡的可玩性。假设设计人员要等上一分钟或更长的时间来进行测试，在自己的时间用尽之前，他将无法尝试为游戏设计许多不同的变化。由于以上原因，合乎道理的做法是，应该有一位专门的编程人员，其主要工作是，尽可能使这种转换更加流畅，速度更加快捷。

任何一个经验丰富的设计人员都会告诉你，决定游戏成败的很大一部分因素取决于游戏测试工作进行得怎样，以及游戏的谐调工作做得如何。起初非常出色的游戏设计，也可能因为编写的游戏没有经过彻底的测试而终告完全失败。我这里不仅仅指通过试运行游戏找出程序中出现的错误，而且还要测试游戏的可玩性、测试游戏时的感觉、测试玩家对它的着迷程度。游戏测试是一个反复进行的过程，它包括尝试某种类型的游戏，再进行修改，然后再尝试，这个过程一直循环下去，直到游戏让人觉得好玩为止。如果关卡编辑器无助于游戏关卡的修改，设计人员能够容易地验证所做的修改，那么重复游戏的测试将会非常困难。设计人员越是能够容易地跳转到游戏里，他们就越有可能一遍又一遍地重复整个游戏测试循环过程，直到游戏变得尽可能地完美。如果关卡编辑器无法方便地进行这种测试，设计人员有可能变得毫无信心，或者没有足够的时间来衡量整个游戏。



19.1.4 编辑游戏世界

开发游戏最好的工具是由一些非定制的、公开的软件和具有专利权的编辑器所共同组成的，这两种软件微妙地结合在一起。一个高效的开发小组知道，他们在多大程度上可以使用、依赖这些软件，这样，在有合适的商业程序包能够满足他们的需要时，他们就不必浪费编程人员的时间，让他们开发一些过分复杂的工具；同时，他们也不会禁止设计人员在关卡编辑器中提炼、加工游戏的内容，从而无端地限制设计人员所做出的努力。虽然没有人禁止游戏开发小组使用任何关卡编辑器进行工作，但同样不明智的做法是，强迫开发小组使用专利开发工具完成所有游戏内容的制作工作。

关卡编辑器很重要的一点是，它允许设计人员实际修改关卡中所有非常关键的部分。对我来讲，这应该是编辑器应当具备的非常明显的首要条件，但同时，我也曾听到过许多开发小组里所发生的一些事情，我感觉有必要在这里提一下。这些小组使用诸如 3D Studio Max 以及“实体编辑器”之类的工具。通常，这些开发小组认为，他们使用一些现成的工具，比如 Max，就能侥幸地创造出游戏环境中的所有几何结构，然后再设计一个关卡编辑器，用来从 Max 中导入





网状结构，定位各个物体、NPC 以及其他游戏环境实体。采用这种方法并不会创造出好的关卡。设计人员往地图中摆放创造出来的人物时，他需要能够同时改变其中的几何结构，以适应物体的摆放。如果设计人员对所做的修改进行检查之前，先得退出编辑器，再启动 3D 模型构造软件（这类软件的运行速度都不会太快），接着修改程序中的几何结构，然后再将关卡导入自己开发出来的编辑器，他肯定会垂头丧气，因为要对几何结构进行太多的调节。其直接结果将是，即使成品游戏还能勉强运行的话，其中的几何结构看上去也会跟设计的不大相同。不允许设计人员在编辑器中修改关卡里的关键结构，实际上等于束缚了设计人员的手脚。只有我亲身经历的这种方式，才能真正使设计人员更好地工作。

我刚刚开始接触 *Centipede 3D* 的设计工作时，所使用的关卡编辑器实际上是一个游戏实体操纵器，而不是严格意义上的关卡编辑器。关卡中的几何结构来自于一个灰度图像的方块高度图（*Centipede 3D* 使用的都是 32 像素的正方形），其中的每个像素代表场景中的一个高度值。这些高度图可以由 Photoshop 或其他位图编辑软件进行制作，用来创造关卡几何结构的初稿，是一种挺不错的方法。但非常不幸的是，在项目初期所使用的编辑器版本中，高度图只能在位图绘画软件中进行修改，而无法在编辑器中修改。这样做非常丢人，因为以 2D 的方式观看一个 3D 关卡，无法更好地了解到游戏的关卡最终看起来会是什么样子。因此，项目中早期的一些关卡非常简单，并且有些单调。这并不是说设计人员没有花大力气使关卡有趣些，而仅仅是因为当时的工具只能提供这些非常可怜的功能。

但是，项目进行到一半时，编辑器增加了新功能，允许设计人员在编辑器中直接编辑高度图。高度图同样可以用 Photoshop 来创建，然后再引入到游戏中，这仍然是创建关卡初步结构的最好方法。初步结构创建完成后，就可以很容易地在编辑器中对其操纵，设计人员可以一边修改高度图，一边观察 3D 关卡。这样做的结果是，设计人员可以不断调整几何结构的设置，直到完美无缺。关卡的质量得到了巨大的改观。通常情况下，时间不允许我们回过头去，重做前面设计好的关卡。因为我们是按照关卡在游戏中出现的顺序来制作的，任何一位玩 *Centipede 3D* 这款游戏的人都应该能够指出来，在关卡的某个位置，应该提供给设计人员一个新的、改进后的设计工具。这并不是说，使用前期的编辑器，设计人员就无法设计关卡，我所说的是，那样的话编辑关卡会非常非常的困难，设计出来的关卡也不会和设计人员预期的一样精彩。

如何用纯 3D 特性的软件创造出精美的关卡几何结构，这一点还有许多内容要讲；虽然许多关卡编辑器拥有复杂的几何编辑功能，但如果它能够从外界导入已创建好的结构，将会大有益处。创建高品质游戏美工资源，其最主要的一点是，能够从商业软件包中导入，不管是 2D 子图形还是 3D 模型。我不知道，是否曾有人被迫只借助企业内部开发工具来为游戏进行 2D 子图形美工设计。然而，似乎许多不幸的美工设计人员，他们只被允许使用拥有自主产权的模型构造工具，来构造人物或其他对象的模型。我已经和大家讨论过了，允许关卡设计人员在编辑器中控制关卡的组织构造，这一点非常非常重要。但是，强迫游戏设计人员和美工们在编辑器中构造每一个游戏环境中的要素，这当然也是一个巨大的错误。美工设计人员应当能够使用自己喜欢的模型构建程序来创建游戏中的对象，比如，树木、武器或者垃圾桶。简单地说，游戏编程小组根本无法编写出一个美工编辑软件包，能够跟 Photoshop、3D Studio Max、Maya、Softimage 或其他众多非定制的流行产品一样，做到功能强大、运算迅速、运行稳定。如果缺少了这些软件包所提供的某些功能，美工设计者将无法创造出最佳的品质。而且，大部分的美工设计人员





已熟悉一种或几种上述提到的程序包，这样，当他们开始接手项目后，能非常容易地做到满足时间的要求。

与此同时，开发小组必须能够使用自主开发的工具来处理这些美工设计。使用企业内部的编辑器，创建动画、骨架节点、碰撞数据或其他信息，对于保证游戏中美工作用的正常发挥是最基本的。如果开发小组试图避开设计某种美工编辑软件这样一个环节，那么，为了使这些美工设计和动画能在游戏中正常运行，美工、设计人员或其他任何从事美工配置工作的人，将会感到垂头丧气。如果想让自己开发游戏的经历走向成功的话，一个重要的组成部分是拥有一款属于企业自主产权的美工操作工具，它能够执行游戏引擎所要求的功能。



19.2 脚本语言和对象行为

游戏开发使用某种系统，设计人员可以不求助于编程人员，按照自己的需要，在系统中精确地创建和平衡敌人、武器以及其他游戏中的行为，这似乎已经成了一种标准。现在的许多游戏中都使用了脚本语言，这些脚本语言相对来讲可能简单一些，但却能在不用重新编译游戏引擎的情况下创建出复杂的实体。脚本语言为游戏开发提供了诸多益处。也许最重要的一点是，这些脚本语言能够鼓励设计人员创造出游戏中更多的独特行为，不管这些语言是否可以被游戏中的实体重新利用，比如 NPC 或是不同关卡中独特的行为及事件，比如玩家在 *Half-Life* 中所看到的某个特定对话进行时的 NPC。

设计得当的脚本系统最大的好处在于，它能够完全移植到其他系统中。这就是说，当游戏被移植时，比如从 PC 机移植到 MAC 机时，只要脚本解释器和相关的功能被正确移植，在 PC 机中编写及调试后的敌人的行为，在 MAC 机中会同样起作用。在这种情形下，使用一种功能强大的脚本语言比使用 C 语言进行编程也要更加稳定一些。使用脚本语言编程，游戏彻底崩溃的可能性将减少，如果脚本出现了非法错误，游戏会弹出一个相应的说明性信息，而不是将游戏锁死。通常，脚本语言并不像 C 编程语言那么复杂，因此，设计人员就多了一些编程技能，可以进行游戏环境中独特行为的创造；这样，就能将编程人员解放出来，从事更为复杂的工作，要知道，编程人员是越来越难找到了。在大部分系统下，脚本可以根据需要载入，意思是说，只有当游戏某部分需要时，脚本才驻留在内存中，这样也可以节省大量的代码开销。游戏使用脚本语言的另一个好处是，它能够让玩家用户进行复杂的修改。设计得当、功能适中的脚本系统，能够让一些兴趣盎然的玩家编写出自己的“修改版本”，然后分发给朋友们。

脚本语言也有自己的缺点。首先是编写脚本系统代码所需要的时间。如果真的要使这种语言功能强大，就像我们在前面描述的那样，它工作起来要非常稳定，还要为用户提供强大的功能，那么代码编写的工作量肯定不会太小。调试有问题的脚本也非常麻烦，因为游戏开发人员根本没有时间编写出符号调试器，一个跟 Visual Studio 或者 CodeWarrior 附带的同样棒的调试器。大部分情况下，脚本是在运行时编译的，结果是，其运行速度比 C++ 代码要低得多。同样，不管开发人员对脚本进行何种的优化，其编译能力还是无法与 Watcom、Microsoft 或 Metrowerks 等公司制作的 C++ 编译器匹敌。最后，虽然脚本语言最大的优势原本应该是，非专业编程人员也能使用，但实际结果往往是，如果脚本语言真的功能强大，可以用来为 NPC 创建 AI，那么，





脚本语言也会非常复杂，只有专业编程人员才能有效使用。如果编程人员的时间被脚本的编写占用，为什么非要让他住手，要他停止用 C 语言编写代码呢？

当然，脚本主要的一个优势是，它能够容易地平衡游戏。设计人员可以调整脚本中的数值，然后运行游戏，而不用调整代码中的某个数值，然后再等待重新编译后运行。但是，假定某人不想使用脚本系统而又想获得以上好处时会怎么样呢？再假定，如果设计人员在关卡编辑器中就可以调整行为参数呢？这些正是 Surreal Software 的 Riot 引擎所采取的方法。在 Surreal 的关卡编辑器中，准许设计人员调整某个 AI、某种武器或其他游戏环境实体的设置或“行为变量”。这些行为本身是用 C++ 语言编写的，能够决定游戏环境中对象行为方式的所有重要设置，比如，对象移动的速度、对象的探测半径、对象被消灭后变成其他什么对象，以及诸如此类的设置，编程人员都留有“钩子”。这样，设计人员能够随心所欲地对游戏进行设置，同时仍能利用强大的 C++ 编译器和调试器的速度，实际上它也提供了脚本语言下平衡游戏的长处。这种功能使关卡编辑器不仅仅只是一种修改游戏关卡的工具，同时在很大程度上也把它变成了一个游戏编辑器，设计人员可以借助它快速地改变大部分游戏中的内容。

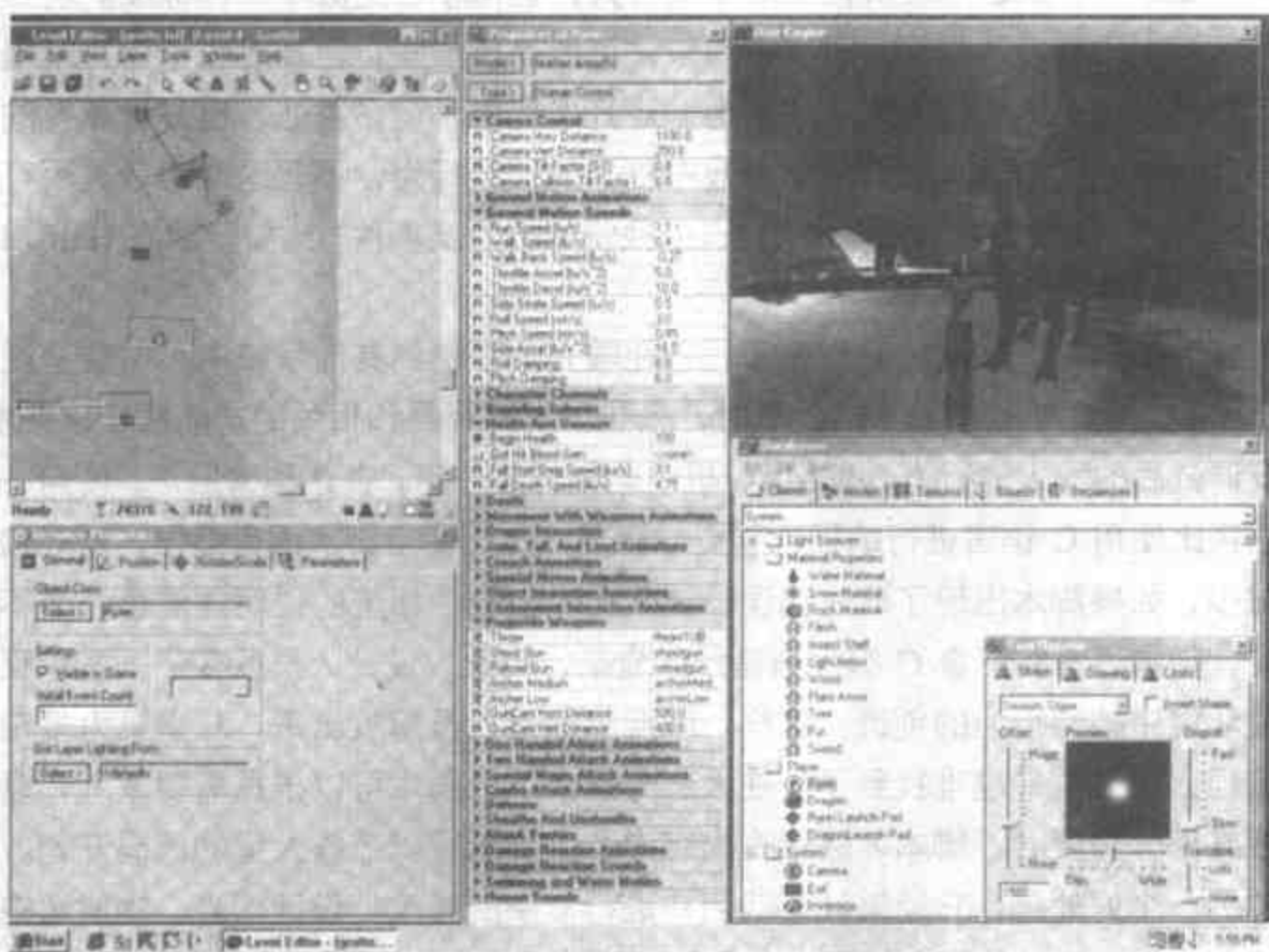


图 19.4 Surreal Software 公司的 Riot 引擎关卡编辑器允许设计人员调整游戏环境中不同实体的各种设置

游戏脚本语言可以很好发挥作用的另外一个方面是“脚本事件”。游戏中每个关卡都可以拥有自己独特的脚本，这些脚本设置、引发同一关卡中各种独特的行为。游戏开发人员近来对于那些复杂、独特的行为越来越关注，特别是在 Valve 使用脚本事件创造出了 Half-Life 中如此强烈的效果之后。当然，“脚本事件”与设置这些“脚本事件”所使用的“脚本语言”之间有一个主要的区别。Half-Life 中的脚本事件很棒，但很显然，建立这些脚本事件使用的方法却并不容易。创造一套既可靠又简单的脚本系统，才能确保设计人员愿意使用它，这是一种最佳的





方式。为了替代以文本方式为基础的脚本语言（这种脚本语言需要单独编译），关卡编辑器应该能够让编程人员非常容易地创建复杂的游戏事件。此类功能，StarCraft 的战役编辑器可算得上是一个非常好的正面例子。其“触发”编辑器，允许设计人员使用一种非常熟悉的点选式交互界面，来创建复杂的脚本化事件。弹出式菜单列出了可供选择的所有命令，第二级弹出菜单向设计人员显示了可赋予这些命令的各种参数。所有命令都是用平白的英语书写的，第一次接触这套系统的人也可以非常容易地看明白。这样，通过战役编辑器，StarCraft 的关卡中就会有許多独特的事件，而不用花费额外的开销来购置一套成熟的脚本语言。



19.3 我们和他们

非常不幸的是，某个项目工具的开发经常演变为编程人员 and 设计人员之间的一场争斗。游戏编程人员经常会因为各种原因而不愿进行工具的开发。首先，许多想从事游戏具体编程的编程人员，他们经常这样做，因为他们不想编写数据库，不想编写数据表，或者是 3D 模型组件。他们想要设计游戏，而工具的设计太像“真正意义上的编程”了。还有这样一种理解，即编写游戏代码比编写工具代码要重要得多。如果这款游戏获得成功的话，会有成百万的人来玩，而某个项目的工具也只不过有十几、二十几个人用。如果他的朋友问他在游戏公司是做什么的，大部分编程人员都不愿说“我是设计工具的”。那没有什么炫耀之处。

更为复杂的是，有人有这样一种想法，即编程人员的时间要比设计人员的时间宝贵。因此，如果因为编程人员无法抽出时间更好地改进关卡编辑器，设计人员就要花费五倍的时间来制作，那么，这不是很好吗？关卡还是设计出来了，不是吗？

就像我在前面所讲述的那样，游戏开发人员不应该这样问自己：“这些工具是否考虑到了游戏内容的创建？”相反，他应该问：“这些工具是否考虑到了如何更好地创建游戏的内容？”如果设计人员需要不停地和这些关卡创建工具较劲，他将无法将时间投入到对游戏进行精心加工的工作中去。事实上，看到编程人员的懒惰，设计人员也许会愤怒异常，而无法全心全意地致力于关卡的开发。一个好的设计人员会被好的工具所激发，尽最大的努力更好地完成工作，因为他能看到最直接的结果。我在讲述 Centipede 3D 中关卡设计工具和关卡设计品质时所使用的例子，对游戏开发人员来说，也是非常生动的一课。设计出高品质的关卡编辑器，关卡的质量也会在很大程度上得到提升。

设计工具的编程人员设计出一款真正令人满意的工具软件，进而促进设计人员的工作，他应该为此感到骄傲。编程人员负责开发出构思精巧、代码编写准确的关卡编辑器，极大地促进了创建许多漂亮关卡的工作，他应该感觉到，他在创建这些关卡的工作中扮演了一个至关重要的角色。因为如果缺少了关卡编辑器中的某些功能，设计人员就无从创造出那些游戏中的景观或者结构。设计人员应该重视编程人员的工作，要记住是因为编程人员的工作，才使这些漂亮关卡的创建成为可能，同时要相应地承认编程人员所付出的努力。





图 19.5 Blizzard 公司 StarCraft 游戏中所使用的战役编辑器能够自动创建不同场景结构之间的过渡，节省了设计人员大量的工作

我曾在 Riot 引擎关卡编辑器中加入了材质构造功能。Riot 引擎采用了平铺纹理的方式来构造游戏中的场景；当纹理变换时，比如，由草地纹理转换为岩石纹理时，中间有过渡纹理。我所增加的功能可以使编辑器在两种不同类型的纹理之间，自动置入合适的过渡。非常有趣的是，六年前公开发行的我的第一款游戏 *Odyssey: The Legend of Nemesis*，其所采用的关卡编辑器就包含这种功能。确实确实，这种自动过渡的功能在许多 2D 地形关卡编辑器中也能见到，例如 Blizzard 公司的 StarCraft 战役编辑器。在增加此功能前，Surreal 的关卡设计人员要手工挑选出所需的过渡纹理。当然，自动过渡功能对关卡的创建并不一定是绝对必需的。Drakan 这款游戏中的所有关卡并没有使用自动过渡这一类的工具，同样，这些关卡也非常漂亮，合适的地方过渡也很恰当。主要的区别在于，这些过渡是设计人员花费了大量的时间一点一点创建起来的。在我增加了自动过渡功能后，设计人员非常兴奋，因为大量繁杂的工作现在已经没有了。有人甚至说：“理查德下个月可以休息了，我们照给他开工资。”他非常欣赏我所增加的功能，同时又体贴周到地将这种谢意传达给了我。受到这样的称赞，我会不断地为编辑器增加更多、更好的功能。

19.3.1 最好的愿望

可是，你要小心啦。有时编程人员授命为编辑器增加新功能，最终结果却是，没人真正需要他们所增加的那些功能。如果编程人员大部分情况下并不参与游戏关卡的设计工作，他也就不会花太多的时间致力于关卡编辑器的设计，想要让他弄清楚编辑器缺少什么样的功能，也就太免为其难了。确实，编程人员看来很酷的一个功能，设计人员却无人愿意使用。编程人员费



尽心思为编辑器设计一项功能，而设计人员却不愿意使用，编程人员的不满之情便会油然而生。然后，当设计人员找到编程人员，要求为编辑器增加某个所需的实用功能时，编程人员可能会置之不理：“他从来也不使用‘顶点扭曲’工具，那可是花了我很大功夫的呀，那么，我为什么要为他设计这个模型自动对齐工具？忘了它吧。”

在这个行业工作过的人都知道，在许多方面，设计人员和编程人员的思维方式是不一样的。因此，设计人员和编程人员之间，要就编辑器需要什么样的功能及如何最好地编写、设计这些功能，进行不断的交流。开发内部工具套件，编程人员拥有巨大的优势，即他们与设计人员有密切的接触。他不用猜测设计人员需要什么样的程序，相反，他可以找到他们进行询问。同样，设计人员也有非常大的优势，他们可以给关卡编辑器开发人员提出建议，提示软件该如何运行。双方进行良好的信息沟通，工具的功能必定会不断提高。

为了促进工具的开发，可以使用另外一种可能的技巧，这就是，将任务指派给一位编程人员，主要负责关卡编辑器的维护及改进。这位编程人员会对工具的开发工作越来越熟悉，如果他能设计出一款好的软件，他将会为此感到骄傲。如果有一位编程人员负责大部分的编辑器开发工作，设计人员就会知道，他们的改进建议可以向谁诉说。反过来，编程人员也能充分意识到，设计人员喜欢什么、不喜欢什么。当然，被指派任务的这位编程人员如果一门心思地想进行照明效果或 AI 的设计工作，其结果必定是，设计出来的工具非常糟糕。这种策略想要成功的话，最重要的一点是，要找到一位真正想从事工具开发工作的人。

另外一种实用的策略是，找一位编程人员，实际使用工具创建一个完整但简单一些的关卡。用这种方法，编程人员可以很容易地找出工具中需要改进的地方，最终弄明白设计人员近段时间的抱怨之所在。编程人员如果不坐下来实际使用自己设计出来的软件，往往可能会得出这样的结论：设计人员过分强调了编辑器中的问题（行业中的说法是“whining”）。通过实际使用自己开发的工具，编程人员可以很容易地找出编辑器中的不足之处，经过几个小时的代码编写工作，稍加修改即可改正。设计人员经常不理解各种编程工作的复杂性，要求在编辑器中增加无法实现的功能，而对于一些非常容易改正的问题，他们却认为无法修正。也许所有这一切最好的解决办法是，找一位身兼两职的工作人员，花大气力进行编辑器的开发工作。这位身兼两职者被推动着去改进每天要用得着的工具，他有可能会尽最大的努力开发出一款最好的工具。十年前，我敢肯定，这种情况并不少见。但是，在大规模项目开发的今天，这种情况已经非常罕见了。为工具编程和设计关卡，现在已经成了两项单独的工作，需要花费两个人的全部精力，身兼两职的时代已经成了昨日黄花。



19.4 万能的游戏编辑器

关卡编辑器实际上应该没有任何错误。如果你够幸运的话，或许能在商店里买到这种没有任何错误的软件。一款内部使用的游戏开发工具，即使已经做得很好，其中也会有大把大把的错误。重要的一点是，这些错误只出现在能够预见得到的情况下。这些错误以固定的模式出现，设计人员可以学会如何预测到它的出现，并训练自己如何避免错误的出现。一旦设计人员熟练掌握工具的使用，他就能知道，自己不能干什么，怎样才能方便地绕过这些错误点。拥有专利





权的关卡编辑器属于软件开发中的另外一种情况。有一个笑话这么说：“医生，那样做很痛！”“那你就别那样了！”真是一语道出了实情。

当然，如果开发游戏项目所使用的工具真的不错的话，不仅市场销售状况能如期跟进，而且还会使人产生这样一种美好的想法：“嗨，我们能跟游戏一起发布我们的工具了！”确实，将关卡编辑器与游戏一起发售，用户可以创建出附加关卡，在游戏发布很久之后还能保持用户对游戏的兴趣。铁杆游戏迷喜欢创建他们自己的“模式”，在朋友或公众之间传播。对于要发布的工具来说，软件中的错误要相对少一点儿，或者至少要比内部使用时稳定一些。向游戏迷发布关卡编辑器的可能性，应该能够作为鼓励编程小组开发出最佳工具的一种激励动机。当然，让游戏迷群体创建附加关卡，对于这一点，一些开发商还没有弄清其中的逻辑，并拒绝发布这些工具。他们经常给出的论据是，如果用户自己创建了那么多的关卡，那么又有谁会来购买游戏的续集？当然，id 软件公司，是这家公司普及了向公众发布关卡编辑器的做法；从财政方面讲，他们好像一直做得挺好。这说明，保护主义者在关卡编辑器这一点上的想法有点愚蠢。

所有这一切都要归结于游戏行业应该得到承认的一个公理：游戏的好坏要取决于设计游戏时使用的工具的好坏。一个构思精巧的关卡编辑工具能够决定游戏的成功与平庸。人们可以这样想象：理想的关卡编辑器就像某一个地方，在那里，设计人员能够完全控制游戏的整个游戏环境：其建筑（玩家所能到达的地方）、其美学表现（照明、纹理以及声音）、其游戏性（NPC、项目以及其他实体的放置、运动以及行为）。当然，不标准的引擎、有严重缺陷的游戏设计方案或者毫无士气的开发小组，对于这些问题，即使是世界上最好的关卡编辑器也于事无补。不过，这些是我们下一章将要谈论的话题。



Chapter 20

第 20 章 游戏分析：The Sims

由 Will Wright 设计



根据概念本身来看，The Sims 并不是很多人想玩的一种游戏。实际上，这个项目以前的开发人员合作得不是很好，游戏的设计者 Will Wright 在开发这个项目的过程中遇到了很多困难。那为什么这个游戏还能这么好玩呢？“在虚拟社区中控制家庭中的一组人物。”听到了关于游戏的这些描述，感觉不像是真的，平淡无趣的生活也可以如此充满乐趣。实际上，游戏所模拟的就是家庭生活——而不是出去听音乐会或者让这些模拟人物出去溜冰。但是听到有人谈论 The Sims 就立即来了兴趣。“我让自己虚拟的人物和这个女的调情，他的丈夫就打了我控制的人物！”那么是什么使这个游戏如此出色，如此充满乐趣呢？

总结起来，玩家在玩 The Sims 的时候，首先要创造他想控制的人物，并为他指定一些不同的品质：整洁（Neat）、外向（Outgoing）、活跃（Active）、顽皮（Playful）和正派（Nice）。



玩家可以把这些人物安排在自己家的房子里，这个房子是预先设置好的，或者是自己建造的。从这里开始，玩家就要保证房子里有模拟人物需要的所有物品：床、卫生间、厨房、电话、娱乐设施等等。Needs 指示器可以用于表达模拟人物要得到幸福感的各种需要，包括 Hunger、Energy、Comfort、Fun 和 Social。玩家还必须为模拟人物找到赚钱的方式，以便可以负担得起玩家所购买的东西；这个目的可以通过在报纸上找工作来达到。另外，游戏中还设置了完善的社交部分，模拟人物们可以相互邀请、交谈、一起娱乐、调情并且成为朋友。游戏提供了很多活动的领域，玩家可以在游戏中探索；有人会感到惊讶，所有的活动范围在功能上都是如此完备。



20.1 放弃创作权

关于 Doug Church 在游戏开发者论坛上所说的电脑游戏中称为“放弃创作权”问题，The Sims 是一个非常好的例子。也就是说，游戏设计者并没有事先想出游戏的故事；现在的电脑游戏中有 95% 都是冒险类游戏、角色扮演类游戏和动作类游戏，游戏中故事的创作权已经转交给了玩家。无论游戏会如何无聊、乏味或者平庸，玩家都可以任意控制游戏发展的方向。实际上，玩家首先不会想到这种经历可以成为一个故事，就好像他不会想到自己的生活会成为故事一样。而这的确是个故事。在 The Sims 中，故事的延续要依赖于各方因素的努力：控制故事发展的玩家和提供框架、工具以及玩家可用空间的游戏设计者。因为玩家密切地融入到了故事的创作中，那么游戏就变成了他的，因此玩家也就更加投入到了游戏中。和许多其他的游戏由游戏设计者控制游戏的情况不同，现在控制游戏的是玩家。能够控制游戏的感觉是非常有满足感的。



图 20.1 The Sims 提供了一种框架，玩家可以创作自己的故事



很多人都认为 The Sims 是一个软件玩具，而不是技术角度上的游戏，但经常有人称之为游戏，并且与其他的游戏主题一同讨论。实际上，The Sims 的确是一个玩具，因为虽然它也可以逐渐发展或预示着某个目标，但它并没有为玩家提供一个明确的目标。对于 The Sims 来说，玩家无法定义“成功”或者“失败”。当玩家的模拟人物在厨房火灾中丧生时，他可能就会认为自己失败了。或者当模拟人物成功地建造了附近地区中最大、最豪华的房子，并且达到了事业的顶峰时，玩家就有可能认为自己胜利了。然而，这些胜利/失败的条件都是玩家自己加在游戏中的，而不是游戏所需要的。这样做会将故事的著作权交给玩家，而游戏也不是以往的那种面向目标的游戏了。例如，玩家每次玩 San Francisco Rush 这样的赛车类游戏时，游戏的结果是预先设计好的：一旦玩家或者他的某个竞争对手通过了终点线，游戏就结束了。所以 Rush 这个游戏中“故事”的结局是预先设计好了的。玩家可以把握赛车在比赛中的表现，以及为了取胜所使用的技巧，但故事的结束方式已经是确定了的，无法改变。诸如 Civilization 这样的游戏，即使给玩家较大的自由度，但也仍假设当 2000 年到来的时候游戏就结束了；那个时候玩家赢得了空间竞赛，或者达到了军事上的优势。通过设置取胜的条件，游戏设计者决定了游戏结束的方式。因为 The Sims 等软件玩具并没有规定游戏必须结束，当游戏需要结束时，要由玩家来做出决定。有些玩家，尤其是一些专业级玩家，认为这种没有胜利或失败的情况对游戏不利，但是对许多玩家来说，这样可以使游戏经历更加吸引人。



20.2 熟悉的主题

当然，The Sims 并不是最早出现的软件玩具，也不是 Will Wright 的第一个作品。Will Wright 第一个获得成功的软件玩具是 SimCity。该游戏模拟了一个非常完善的系统，使玩家可以真正地控制城市的命运。虽然 SimCity 是非常棒、非常好玩的主题，但 The Sims 更具吸引力。其中更多的原因是因为 The Sims 的玩家是在控制人，而不是在控制城市。换句话说，它与 Chris Crawford 的那种游戏应该“关注人而不是物”的观点相一致。通常来说，大多数玩家会发现控制人比控制物要有趣，而且玩家会更容易与模拟人物之间产生感情，而与虚拟的城市之间却不那么容易亲近。在玩过了 The Sims 以后，如果他们热情的求爱遭到拒绝，或者他们的房子毁之一炬，玩家就会感到很伤心。虽然 The Sims 中的模拟人物不如真人那样聪明或者有趣，但是这些人物会让玩家相信他们在虚拟世界中的存在，并且可以填补他们在虚拟环境中的不足。

此外，在开始玩游戏之前，几乎所有玩过 The Sims 的玩家都会对要模拟的主题有一定的了解。他们会觉得自己在这种“乡村生活”的主题上可以算得上是行家，并且认为自己可以把游戏玩得更好。例如，玩家根据直觉可以知道自己应该配备一个带有淋浴喷头的浴室、一个卫生间，还有一个水池。如果游戏中要模仿另一个星球上外星人日常的生活方式，那么玩家就会觉得无从下手，在想玩好游戏之前，必须思考一下那种生活方式。因为玩家已经知道了 The Sims 中的许多内容，他们能在游戏中更加投入。从玩家开始玩游戏的那一刻起，他就会有很好的感觉，因为他可以把现实世界中的知识应用到这些虚拟的生活中。当 Will Wright 制作 SimEarth 时，SimEarth 游戏中包含的系统对玩家来说非常陌生，这就是为什么许多人觉得这个游戏非常难玩的原因。对于 SimCity 来说，玩家对于所要完成的事情有更多的了解；虽然他们可能并不是城市





规划和动力学的专家，但至少可以知道城市的基本布局，并且熟悉交通、污染和犯罪等问题。在 *The Sims* 中，比起城市规划来，大多数玩家会对其中的主题更加了解。因此，游戏玩起来也更加有意思。没有哪些东西可以如此地吸引玩家。

当然，虚拟一个许多玩家都熟悉的主题也是一种挑战；如果设计者出现失误的话，玩家立即就会发现。在模拟外星人的生活时，谁能精确地说出世界和生物到底是如何起源的？这样就使得设计者在如何构造世界方面具有更多的艺术发挥余地。而在 *The Sims* 这样的虚拟现实的游戏里，如果设计者关于如何调动一个模拟人物去做某种动作的问题采用了错误的选择，那么玩家就会看出这个错误，并且这种疑问会立即传播开来。制作玩家熟悉的主题可能会很吸引他们，但是如果出现失误的话，同样也会失去玩家的青睐。



20.3 安全的试验

在第一次玩游戏的时候，玩家可能不会认为 *The Sims* 所虚拟的东西有多么有趣。实际上，如果这些郊区居民会用电脑来玩游戏，并且有可以任意支配的收入来购买电脑，那么 *The Sims* 的游戏世界与现实世界还有什么区别呢？看起来，许多游戏中逃避现实和理想主义的特点在 *The Sims* 中都找不到。此外，*The Sims* 并没有去掉生活中枯燥乏味的部分，玩家所虚拟的人物必须忙碌于很多现代生活中比较平庸的方面，如洗澡、工作、付账单，还有倒垃圾。这很有趣吗？很奇怪，这的确很有趣，因为这些生活中枯燥的部分增加了生活的真实感，使玩家的成功或失败都变得更有意义。

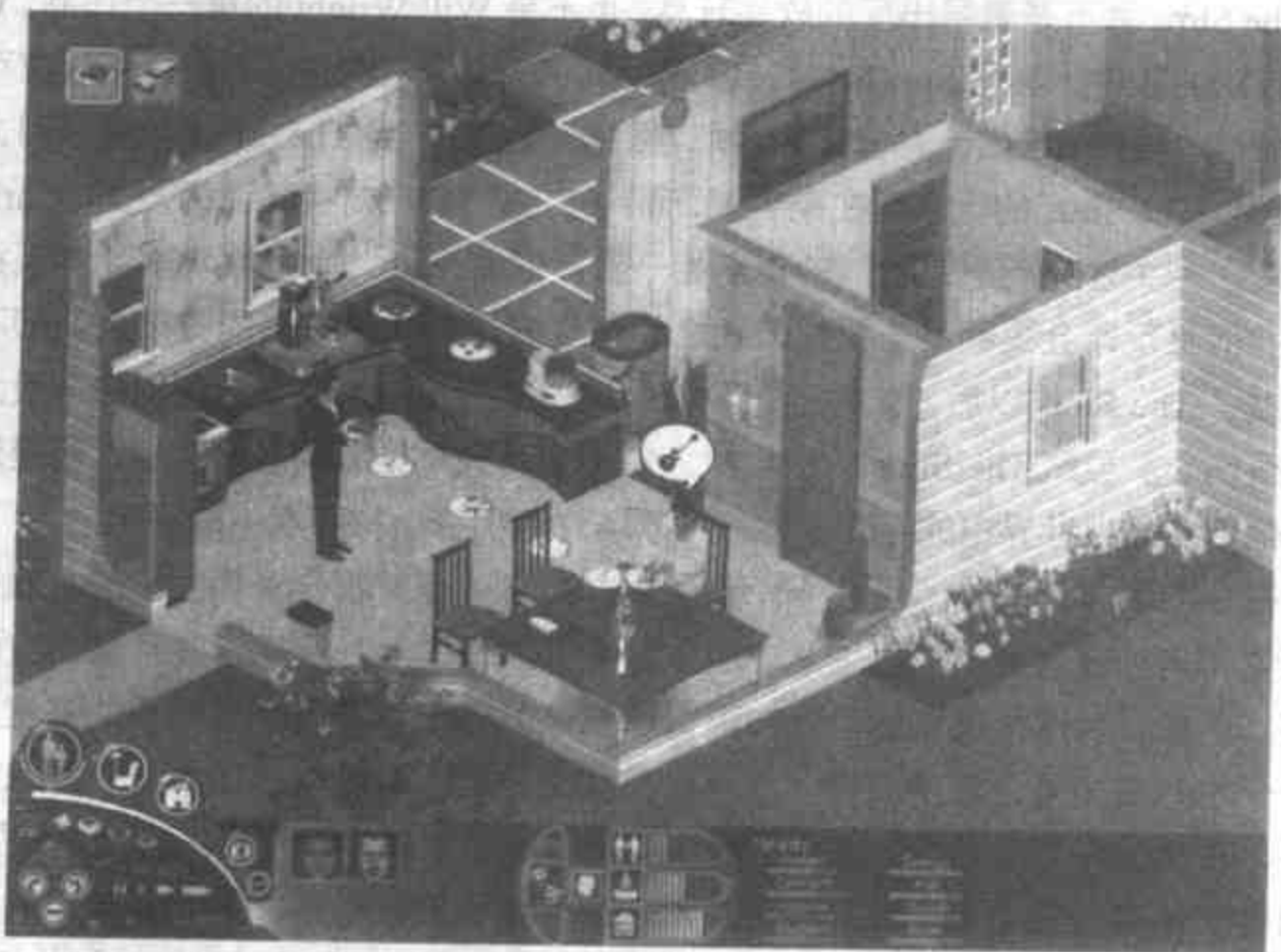


图 20.2 虽然 *The Sims* 中的主题很通俗，但是游戏依然很让人着迷，因为游戏提供了一个玩家可以试验的安全的世界



The Sims 提供给玩家的是安全试验的测试环境。谨慎可以防止玩家在现实世界中成为一个罪犯或者职业运动员，游戏也可以允许玩家让他的模拟人物在那些方向上发展，但不会对玩家造成任何威胁。在 The Sims 中，玩家可以建造豪华的房子、让他们的模拟人物在许多无关紧要的东西上大把花钱、参加野外热水沐浴聚会，或者搞其他联谊活动，他们只想知道如果自己以另一种方式生活的话，那么生活将会是什么样子。如果这些试验性的生活方式并没有如玩家预期的那样好的话，惟一的损失就是会对模拟人物造成影响，而不会造成现实世界中的破产或者遭到现实社会排斥那样严重的后果。实际上，如果玩家在灾难性的事件或灾难性的决定之后并不存盘，那么所造成的损失就完全不会存在。玩家在 The Sims 中所控制的生活可能与自己的生活非常相近，但游戏所提供的这种尝试新鲜事物而不必担心会有严重后果的能力，使这种体验十分有趣和刺激。



20.4 深度和焦点

The Sims 之所以会获得成功，其中主要的原因是因为玩家在控制模拟人物的时候有许多选择。放弃创作权的做法很好，但是如果设计者没能为玩家提供足够有意义的选择，玩家就会发现自己所创造出来的故事范围十分狭窄。实际上，在制作软件玩具的过程中，设计者的责任就是向这个玩具中加入广泛的可能发生的事情，使得游戏对玩家的吸引力不会很快消失。在 The Sims 中，Wright 确实做到了这一点，使玩家感觉到在游戏世界中有很多可以做可以看的事物，而这些事物都是玩家从未想象过的。

玩家可以集中精力建造她的房子，可以从预先建造好的房子开始玩，也可以从头开始。一系列完备的房屋建造工具和地形整理工具使玩家可以建造出各种各样的房子，即便是有成百上千的人在玩游戏，可能也不会有两幢房子一模一样。一旦建好了房子或者购买了房子，玩家就可以集中精力去布置所有的器具，这会对房子的居住者产生各种不同的作用。当然，玩家还要制造居住者，选择他的个性、体型、年龄、种族，甚至还有发型，并且还包括选择孩子还是成人、男性还是女性。一旦模拟人物住进了房子，玩家就可以决定他们要吃什么、学习一些什么知识、从事何种职业、如何娱乐、与谁去交流。无论房子是否是自己建造的，所得到的财产、职业、人物的创造或者对生活方式的控制，这些组成部分中任意一项都是其他大多数游戏所无法提供的。当所有这些不同的系统组合在一起的时候，玩家可以选择的范围呈几何级数递增，使得游戏的深度完全无法预测。

当然，模拟人物在游戏中无法做到的事情也相当多。除了去工作以外，模拟人物不能离开他的家，并且在他们上班的时候，玩家不能跟着他们。可以去其他地方自然不错，但是要知道，如果要虚拟世界的其他地方的话，那么这个游戏将会变得非常复杂。如果这样做的话，则需要做大量的工作；还好，当初开发游戏的时候没有加入这个概念，否则至今为止这个游戏可能还没有完成。游戏把注意力集中在家庭生活中，在不让游戏规模加大的前提下，可以让游戏的情节正常进行。简单地说，拓宽游戏的范围就会降低游戏的深度。如果游戏的设计者花费所有的时间在游戏中加入很多不合理的内容，而没有将精力集中在游戏中的某个方面，那么游戏中的很多特点就会显得非常肤浅。





The Sims 中还比较娴熟地采用了“只需要再进一步”的游戏风格。这类游戏的典范就是 Civilization，在这个游戏中，玩家总是要找到下一种要开发的技术、下一个要建造的单位，或者是找到下一块领地。在 The Sims 中也是一样，玩家要让她的模拟人物与陌生人会面、在他们的事业上有所进展、对房子进行改进，并且考虑某一天让他们养一个小孩，所有的事情都同时发生。因为这些都是平常生活的需要，所以并没有什么适当的时候可以停止游戏；总是有某些事情要去做。这个游戏相当让人着迷，玩家会花费很长时间来玩这个游戏。



20.5 界面

游戏的界面最好不要影响玩家的体验。界面的任务就是告诉玩家游戏世界的状况，并接收玩家的指令，以便对游戏世界进行控制。使用输入/输出界面的做法本身并没有什么乐趣可言；这是玩家与游戏世界的交互过程，应该是一种必要的体验。但是界面决定玩家与游戏世界交互的方式。如果界面没有达到要求，情况好的话，玩家只是会感到沮丧；情况糟的话，玩家就无法完成自己想做的事。

The Sims 的用户界面非常漂亮，可以作为制作游戏界面的典范。它可以向玩家提供大量关于游戏世界的信息，同时又能使玩家容易并直观地控制模拟人物的举动。与许多现代的动作类游戏不同，该游戏的指南主要向玩家提供如何玩游戏的信息，而不是提供如何控制界面的信息。游戏的界面既简单又直观，玩家可以很容易地掌握，无疑这是严格的测试的结果。其中的关键就是帮助贯穿于整个界面中，玩家点击任何文本项就可以了解到其具有重要性和意义的原因。



图 20.3 The Sims 提供了相当直观界面，玩家可以采用多种方式完成相同的动作。



The Sims 中输入/输出模式的成功大部分要归功于,它与玩家玩游戏之前所了解的系统的相似性。例如,决定游戏发展速度的按钮看起来好像磁带式录音机上的按钮,而录音机上的按钮是几乎所有玩家都很熟悉的。许多界面都留有 Microsoft Windows 的痕迹,指向和点击都是玩家在 OS 中经常采用的操作。各个项目的控制也带有 Windows 的痕迹;玩家可以使用拖放的方式来放置物体,或者只需要点击再点击即可。对话框右上角有一个标准的 Windows 中的“X”,表示这些窗口可以关闭,常规的 OK/Cancel 按钮可以组合起来使用。虽然有些功能与 Windows 在许多方面都是相对应的,但是要注意,界面的外观与 Windows 看起来并不相像。所有的按钮都是以一种很友好的艺术风格来描绘的,与 Windows 冰冷的、功能性的外观截然不同。如果游戏使用 Windows 提供的对话框风格,那么玩家很快就会以为自己在使用文件搜索程序或者其他的 Windows 界面,而不是他乐于回忆的经历,当然也就不是“有趣的”活动。然而,通过在 Windows 的行为之上设置一种新的可视风格,玩家就会觉得界面非常直观、非常熟悉,而不会让他想起文件管理器。

另一个例子就是在整个游戏中使用的“头像”菜单。当玩家想让模拟人物在某个物体上进行操作时,只需要点击该物体即可。从点击处会出现当前模拟人物的头像,头像周围会出现一系列他可以做出的动作。玩家只需移动鼠标到他想做的动作上,然后点击一下就可以了。当移动鼠标指针的时候,模拟人物的头像会随着鼠标而移动,看它移动的方向。这个菜单功能等同于 Windows 中的弹出菜单,但是有一些明显的优点。首先是它看起来不像弹出菜单,因此玩家不会将其与枯燥的 Windows 功能联系在一起。其次,菜单仅列出了那时在当前物体上可以进行操作的选项。正常的弹出式菜单会列出所有可能的项目,而对当前不可用的选项采用灰色注释。再次,将模拟人物的头像放在正中间,菜单将玩家带向了采取动作的核心:它要指示模拟人物做一个特定的动作。这样就使得玩家向模拟人物发出的指令显得更为亲切,比使用呆板、乏味和标准的弹出式菜单好得多。



20.6 受控行为和自主行为

在游戏中,玩家能够指示他的模拟人物做出某些特定的动作:取出盘子、给朋友打电话、洗澡等等。然而,模拟人物还可以有自主行为,而不需要玩家的指示。模拟人物有足够的思维逻辑可以完成当前最需要做的事情,这些事情可以是吃饭、洗澡、玩弹球或者是阅读当天的报纸。当玩家对房子进行整修或者购买其他物品的时候,模拟人物就会奔向新的主题,赞赏或者发出抱怨,他们的反应要看对特定物品的喜好程度。这样就可以告诉玩家模拟人物是否会喜欢新的东西,或者是否他宁愿不要这些东西。因为整个房子的设置是影响模拟人物心情的重要部分,所以这就为玩家提供了关于如何最优化设置房子的关键信息。

模拟人物自主的行为可以让玩家设置好房屋,然后坐下来看模拟人物如何在里面生活。这就使游戏更像 SimCity;在那个游戏中,玩家只能安排城市的框架——街道、地区、标志性建筑物——然后看城市里的居民如何生活。The Sims 的玩家可以建造一幢漂亮的房子,他觉得住在这样的房子里面会很舒服,然后坐下来看模拟人物采用默认的生活方式住在里面。这样就使游戏增添了另一个乐趣。



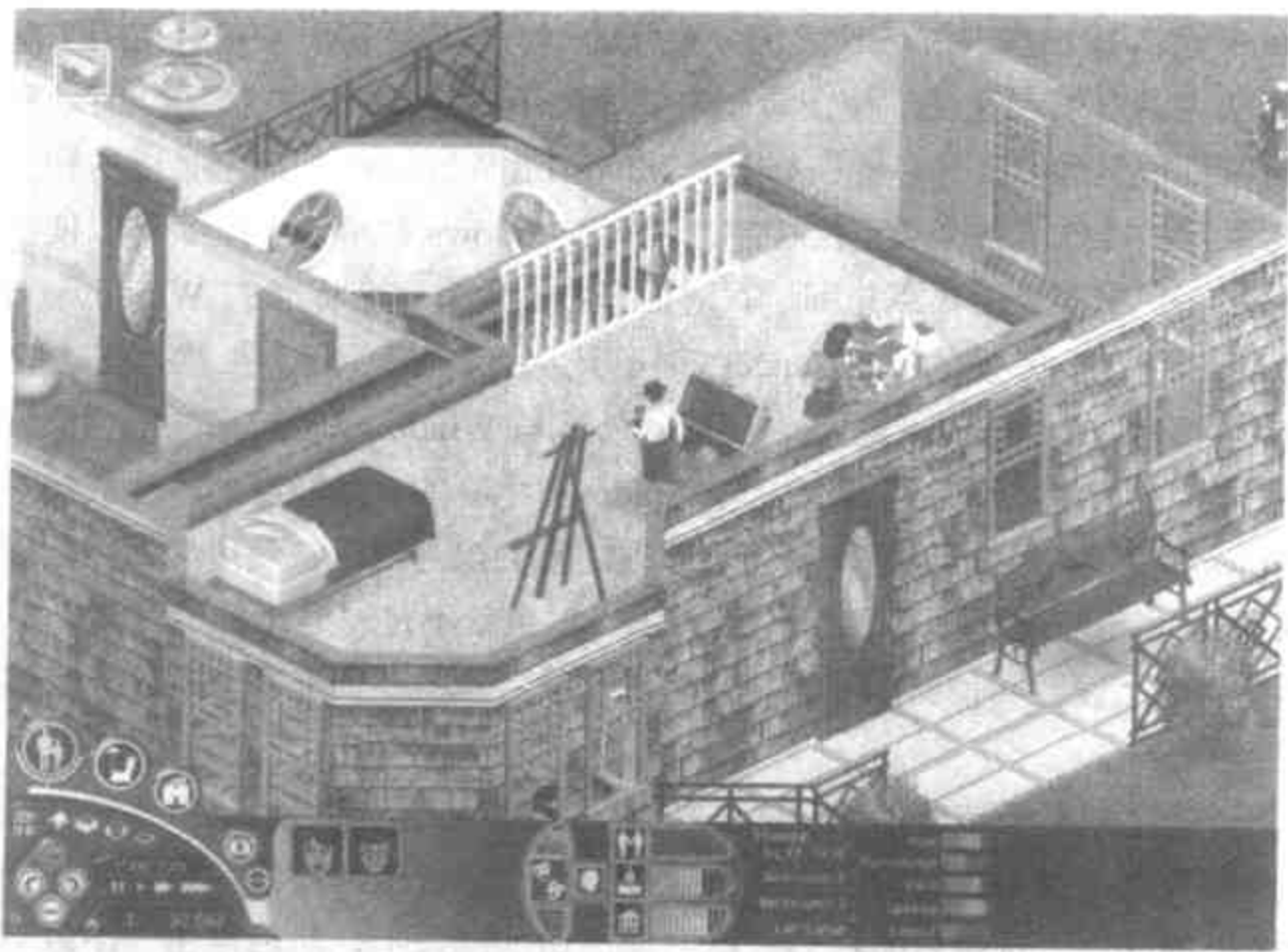


图 20.4 The Sims 游戏本身具备一定的智能，可以让玩家不必担心模拟人物生活的细节

然而，模拟人物通常不会有玩家的预见性，因此如果玩家来指示模拟人物的每个举动，游戏就可以玩得更好、更有效率，也更加有趣。例如，模拟人物不会自己去提高职业技能，如通过学习画画来提高创造性。所以如果玩家想让模拟人物将自己的潜力发挥得最好的话，那么最好由玩家来代替模拟人物来对下一步的举动做出决定。然而，自主的行为可以让玩家避免事必躬亲地做出每个无关紧要的决定。实际上，能够告诉模拟人物每一步所要做的事情是游戏中的关键部分，但如果玩家要同时控制多个模拟人物，让每个人物在特定的时刻计划做某些事情的确是很艰巨的任务。当玩家不想费心去管理的时候，模拟人物自身的行为有助于减轻玩家的负担。



20.7 要吸取的教训

The Sims 可能是近些年来最新颖的商业游戏设计产品。该游戏并没有参照其他任何发布过的游戏，而完全是出自 Will Wright 的灵感。玩过这个游戏，你就会感叹其创造性和新颖性。The Sims 中值得一提的方面很多，整本书都可以用来介绍对设计方案的分析。游戏就像是一个计算机化的游戏室，让我们可以亲临真人的场景，以便更好地理解他们。游戏中对游戏室的描述很有启发性：

Will Lloyd Wright Doll House

这个奇妙的游戏设计适合于每个人，孩子和成人都可以去控制一个小型家庭。这些不可思议的复制品组成了理想的家具和装饰品。如果人们花很长时间沉溺于这个游戏中，千万不要感到奇怪。



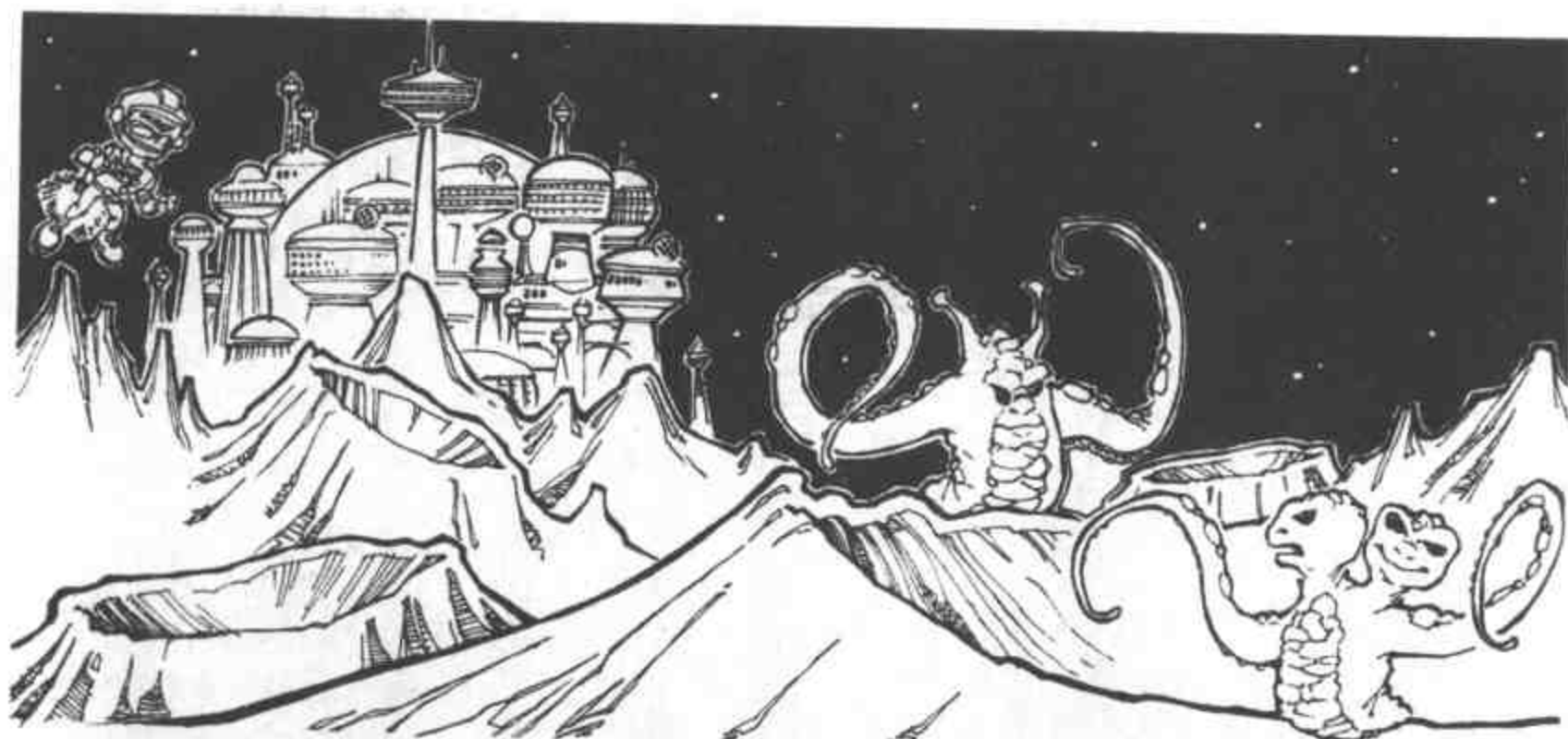
The Sims 中最有趣、也是最吸引人的就是，它实际上是在教我们如何面对自己的生活。我们与自己所拥有的一切之间到底是什么关系？我们生活的空间是如何影响我们的生活的？在人际关系中的嫉妒心是如何产生的？

当然，没有人会说 The Sims 是人类活动完全精确的模拟，但它是不是需要精确到可以让我们以一种新的有趣的方式思考一下我们的生活呢？当我们让模拟人物到处活动，然后观看他们如何相互影响时，可能不会认为这种模拟反映出了他们的行为。但是在这种不赞成的观点中，我们会考虑自己到底想让他们做些什么事情，而且也反映出了我们在现实生活中所存在的人际关系。这看起来是电脑游戏——不会让我们逃离现实或者代替现实，但是却开辟了一种新的思想，可以以一种不同的角度来看这个世界，并且为我们的生活提供了很多宝贵的信息。



Chapter 21

第21章 关卡设计



“作为游戏玩家的时候，我们总是沉迷于如何玩游戏，但是在成为游戏设计人员后，我们对游戏的看法就有了变化。当然，这些变化根据不同的游戏会有所不同，但都是为了一个目的，就是对不同的玩家提供不同的玩法。如果游戏给玩家提供了多种解决问题的方法，玩家就能按自己的方式更投入地玩这个游戏。比如在 Underworld 游戏中，如果要开一个门，玩家可以用钥匙，也可以砸烂门锁，或者干脆对门锁使用魔咒。如果玩家可以选择自己的方式或途径去绕开一个障碍物，当他们达到目的地的时候，一定很有成就感。游戏元素中的灵活性可以让玩家在游戏世界中按自己的方式前进。”

——Doug Church 在谈论 Ultima Underworld 时所说的一番话

随着电脑游戏的发展，游戏的内容和规模都有所增长，过去由一个人承担的任务，现在需要多人合作才能完成。形成这种劳动分工的原因一方面是因为开发商的需要，开发者能够及时地完成复杂而又庞大的游戏项目，另一方面是因为市场的需要。随着游戏制作人员的劳动分工，产生了一类独特的角色：关卡设计者。在游戏设计中，一旦建立好了游戏的核心和框架结构，下面的工作就是关卡设计者的任务了，他们将要创造游戏的世界；正是他们创造的游戏世界，



让玩家在其中乐此不疲。

在一个游戏开发项目中，所需关卡设计者的数量大致和游戏中关卡的复杂程度成正比。对于一个 3D 游戏，如果其中具有非常复杂的细节结构，以至于必须全部由专门的关卡设计者来进行建模，那么在这种项目的人力资源估算中，哪怕让一个关卡设计人员仅仅负责其中一到两个关卡的设计，都是合理的。有些时候，游戏的设计人员也可能承担关卡设计的任务，但有时她也许只对项目中的关卡设计小组进行监督。对于一个 2D 游戏来说，如果不是太复杂，游戏的设计负责人自己来设计所有的游戏关卡也不是不可能的事。

关卡设计是对游戏中各个组成部分进行组装的工作。建造关卡有点类似于拼七巧板：要建立关卡，关卡设计者必须使用游戏引擎、美工和游戏控制内核。关卡设计经常是暴露游戏设计问题的地方。如果游戏引擎不正常，设计出的关卡在执行时就会不稳定，或者游戏帧速率达不到设计效果。例如图片设计尺寸出错，或者发生一些渲染错误，那么关卡设计人员在游戏环境中放入这些图片的时候就会有困难。如果游戏不能支持各个关卡的切换，从而无法满足游戏的完整性，或者更糟糕的是，游戏根本不好玩，那么在关卡设计过程中就会暴露出这些问题。关卡设计者要负责提醒整个开发小组去注意暴露出的问题，并关注问题的解决。这样，经常导致关卡设计人员成为开发小组中最不招人喜欢的成员，因为他们总是在纠缠着其他开发人员去解决问题。不过，如果他们不去这样做，而是忽视遇到的问题，最终只会让游戏变得更糟。游戏的关卡设计人员需要具备高度的责任心才行。

关卡设计人员的工作并不轻松，他们需要从各个角度关注游戏的内容。即使游戏的各个部分运行稳定，但由于关卡设计者本身的工作质量不高，那么这个游戏也有可能是失败的作品。倘若设计出的关卡并不能体现出游戏引擎、美工和操纵性的最好一面，那么即便游戏的各个组成部分再好，恐怕也是一种浪费。设计得不好的关卡将各个成分组织起来，开发出的游戏就发挥不出它的潜力。



21.1 各种游戏中的关卡介绍

不同的游戏对“关卡”的定义是不同的。一般来说，在横轴游戏、第一人称视角游戏、冒险类游戏、飞行模拟和角色扮演类游戏中会存在着关卡的概念。在这些游戏中，存在着一些截然不同的区域，我们称之为“关卡”。这些区域可以按照不同的地理区域划分（火山或冰雪世界），也可以按照一次能载入内存的容量划分，或者按照游戏操纵的“适合”程度来划分，也就是说，玩家连续玩多久，才不至于感到疲劳的程度，这样在下一关开始之前，玩家可以得到适当的休息。在一些经典的街机游戏中，如 Centipede 或 Space Invaders，只有一个关卡；还有其他类型的游戏如 Pac-Man 或 Joust，通过在游戏画面中提供一些简单的变化来延长游戏时间，这样，在 Pac-Man 中不同的迷宫就形成了它的关卡。战略游戏如 StarCraft，其中的关卡或剧情是按照地图划分的，在这些地图中玩家必须完成某个目标，例如要求选择人类一族的玩家在指定的时间内抵抗神族的入侵。在赛车游戏中，一个关卡可能是游戏中的一个赛道。在体育竞技游戏中，例如棒球游戏，关卡可能是游戏中出现的不同的体育场。此时不同的关卡之间就仅仅是美工上的区别，无论比赛发生在 Wrigley Field 体育场还是 Yankee Stadium 体育场，游戏的玩法





都区别不大。

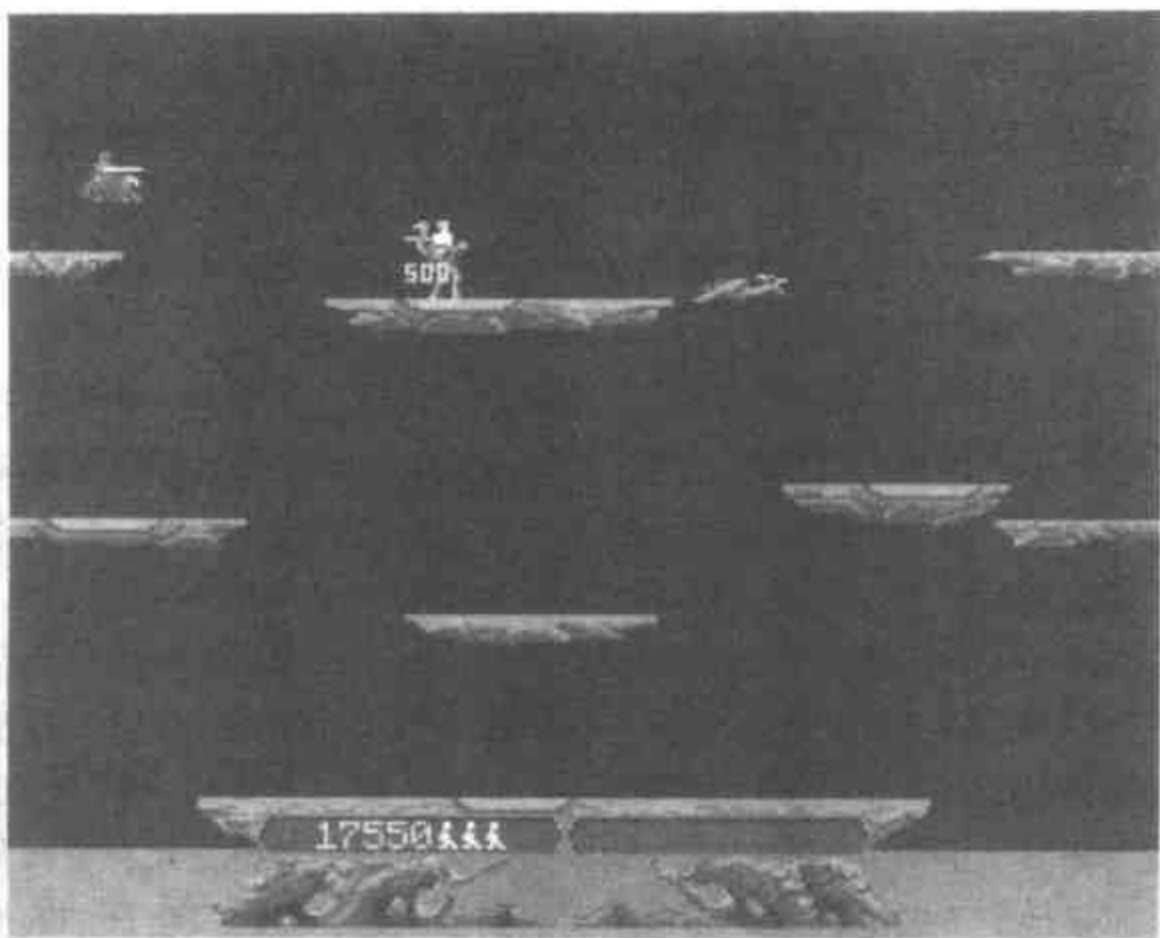


图 21.1 Joust 在游戏中稍微做一些改动就形成了新的关卡

有一些游戏，例如文明（Civilization）和模拟城市（SimCity），确实也具有关卡，但是它们和我们上面提及的游戏相比有一个关键的区别，就是对于玩家来说，整个游戏发生在一个单独的关卡之上。其中的基础关卡常常是随机生成的，在这个基础上，基本上是由玩家负责建立起所玩的关卡。这就是我们为什么把这种主题的游戏称做“建筑工”游戏的原因。对于这种主题的游戏，关卡的创作权基本上被转移到玩家身上。

本章主要讲述使用预先设计关卡的游戏，这些关卡会影响到游戏的可玩性。虽然体育竞技类游戏和“建筑工”游戏可能含有关卡，但这些关卡的建立分别由美工和玩家负责，因此总的来说和游戏设计人员关系不大。然而，对于类似 Doom、古墓丽影（Tomb Raider）、超级玛丽（Super Mario 64）、Maniac Mansion、Pac-Man、StarCraft 或 Fallout 这样的游戏来说，关卡的设计和游戏的操纵是紧密相关的，因此需要设计人员更投入到关卡设计工作中。



21.1.1 关卡划分

游戏的关卡划分和该游戏的流程紧密相关。玩家经常是一次玩游戏的一个关卡。假如一个小女孩正在玩游戏的时候，父母宣布开饭了，这个孩子有可能恳求父母说：“玩过这关就来。”在控制台游戏中，玩家常常只能在关卡之间存储游戏进度，以此作为一个游戏单元结束的重要标志。一个关卡类似于某个戏剧的一幕、一本书的一个章节、交响乐的一个乐章。它让听众或观众在宏观上了解整个作品的一个独立的单元，了解刚才结束的是哪个部分，以后还有多少单元没有欣赏。设计良好的关卡，从开始到结束，在难度和紧张程度上是逐渐增加的，并且在最后可能会有一个小小的难题。有可能要求玩家去击败一个大怪兽，或者完成一项特殊的任务。





玩家在关卡结束后，就会知道已经完成了游戏中的部分重要任务，并且为自己感到自豪。

技术的约束通常决定某个关卡应该在什么地方结束。由于每次只能载入有限的纹理、声音和关卡数据到内存之中，当这些资源使用完毕后，就需要使游戏停止一段时间以载入新的游戏关卡数据。新的技术使得关卡之间能够更好的平滑过渡。即使在技术受限的 PlayStation 平台上，Insomniac 公司的开发人员也能做到在 *Spyro the Dragon* 游戏中避免出现载入画面，在后台进行必要的数据交换的同时，用 *Spyro* 在天空飞翔的动画替代了载入画面，当进入新的关卡时，*Spyro* 就飞回到地面上。与显示令人不愉快的“Loading”画面相比，欣赏飞舞的 *Spyro* 会让玩家感到更舒适。不过，在 *Spyro the Dragon* 游戏关卡中，还是把在未载入屏幕区域的数据分成几个不同的块，这意味着在这些关卡中游戏的操纵还是受到内存空间的限制。当然，一个优秀的设计人员，在可以利用的内存空间中，能够挑战内存的限制并且合理地利用它们去建立有趣和富有挑战性的关卡。设计人员必须挑战硬件设备的极限，并合理地使用它们。

另一个有趣的例子是 *Half-life* 的关卡分级。在这个游戏中，Valve 开发小组希望为玩家创造一个更为无缝连接的游戏，但却受到 *Quake* 引擎使用上的一些限制。*Quake* 本身就有 30 多个关卡，每个关卡在载入时都有一个显著的停顿时间。在 *Quake* 中，各个关卡之间是相对独立的：不会有一个怪兽从一个关卡追逐玩家到另一个关卡，玩家不能从现有的关卡回到上一个关卡。Valve 小组的程序员对关卡做了一些改进，他们的思路是这样的：如果关卡足够小，那么就可以在 5 秒钟内载入一个关卡。另外，他们对游戏进行了修改，在游戏中允许怪兽越过地图的边界追逐玩家。为了达到这个目的，Valve 的关卡设计人员把关卡设计得非常小，这些关卡小于一个标准的 *Quake* 关卡；虽然关卡很小，但是关卡数量相当多。在两个关卡区域的边缘，建筑物都是一模一样的，因此当玩家从一个关卡跑到下一个关卡时，除了看见短暂的载入信息外，不会意识到自己已经到了下一个关卡。这使玩家体验到了关卡的平滑过渡。很明显，设计小组还是觉得有必要在游戏中进行剧情分段，因此在游戏的关键点屏幕上会出现短暂的标题文字。由于程序员和设计小组能够制作几乎无缝连接的关卡载入系统，因此设计小组可以在游戏中最合适的地方单独制作出一个剧情叙述单元，而不是使用传统剧情叙述方式。当然，类似于 *Half-life* 的游戏，最理想的状况就是完全消除载入时间。总有一天会存在这种技术：当玩家即将需要进入下一个关卡时，系统就把一个新的关卡完全放入内存缓存起来。到目前为止，为了达到创造关卡无缝连接游戏的目的，设计者们必须努力做到让关卡的载入尽可能地更快、更隐蔽。

21.1.2 关卡顺序

在整个游戏的流程中，关卡的发生顺序也是很重要的。例如，充满大量枪战的关卡应该和剧情关卡或以解谜为主的关卡交替出现。如果一个游戏把剧情关卡都放在前面，在游戏的后面全部都是些动作为主的场面，这样的游戏就有点不平衡。设计人员至少应该知道关卡的顺序是如何影响游戏流程的，并且应该了解不同的关卡安排方式是如何影响流程的。例如，假设某个游戏有 30 个关卡，6 个关底怪兽，合理的安排是把这些敌人放到第 5、10、15、20、25 和第 30 关的结尾处。当然，并不一定要精确地按照上面的关数去安排关底怪兽的位置，在不至于引起大问题的情况下，把一些关底怪兽放到稍微靠前或靠后的关卡都是可行的。如果把这些怪兽分别放在游戏最后 6 个关卡中，那么这个游戏就明显不平衡了。这意味着玩家可能在玩过前面





24 个没有大怪兽的关卡后，在最后的几关中，每一关都要突然面对一个杀气腾腾的老怪，这会令人感到很怪异。

对于不同的游戏，关卡的划分和关卡顺序的排列是不同的。以 Unreal 游戏来说，它类似于以前的 Doom 和 Quake，对设计人员的惟一要求就是让关卡更酷，而不用太多考虑剧情问题（因为这种游戏基本上就没有什么剧情），并且不用考虑剧情中事件之间的关系。不过往往需要注意游戏中在什么地方安排某种敌人第一次出场，这样，在较前的关卡中，可以使用的生物的选择范围更小。同样的，靠前的关卡需要设计得稍微简单一些，后续的关卡逐渐变难。但是，大部分情况下，关卡设计人员只需要把自己的关卡设计得越来越酷，几乎不需要和其他设计人员发生什么联系。当然，他们可能会看看别人的设计做得怎么样，以此激励自己把关卡做得更好。不过并不要求这些关卡与其前后关卡有一致的主题，对整个游戏不会产生什么不良影响，因为这种游戏本来就没什么故事情节。

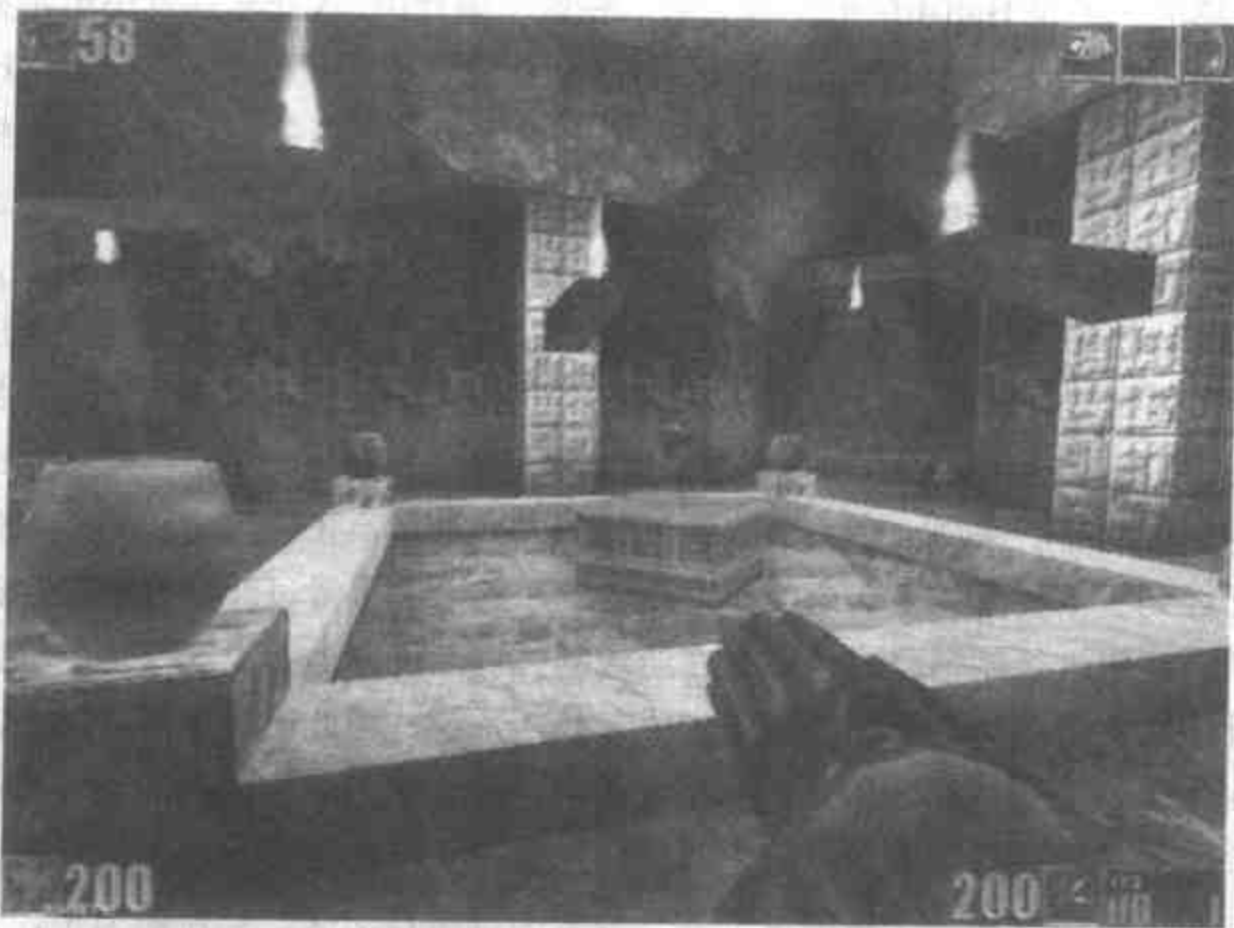


图 21.2 Unreal 游戏的关卡设计人员的目标就是设计一些看起来很酷的关卡，不用考虑关卡之间的联系

然而，在另一些游戏中，如 Indiana Jones 和 Infernal Machine，剧情的成分就很重。为了满足剧情的发展，游戏关卡需要为剧情服务。对于以剧情为中心的游戏，游戏的设计小组和剧情策划小组必须预先安排剧情的事件应该发生在哪个关卡上。并考虑一些这样的问题：这些关卡应该发生在什么场合？在什么场合玩家需要和什么样的敌人战斗？在这种游戏中，关卡的顺序就不能像 Doom 中那样被轻易更改，每个关卡的游戏操纵必须在一开始就策划好。不过，不必在策划初期就把某个关卡设计得详细到分钟的程度，具体的细节最好留给关卡设计人员去做，他们可以在关卡中合适的地方放置个人设计的遭遇战、对象或者一些小小的谜题。需要一个小小的设计文档，对设计人员描述为了满足游戏剧情的需要，本关卡需要完成什么内容，这样将会让关卡设计人员清楚地了解在这一关内需要包含什么内容；在此基础上，她就可以完成其余的详细设计。



21.2 关卡的组成成分

当游戏中需要的关卡被确定下来，同时设计人员对这些关卡在游戏剧情的支持方式上形成了一个大概的想法，下面的任务就是实际动手去创建这些关卡了。无论关卡处于游戏的哪个具体位置，从全局角度来考虑，各个关卡的目标都是为了给玩家提供预定的游戏体验。设计人员需要考虑以下问题：这个游戏企图达到什么目的？这个游戏的各个不同方面有多重要？为了支持游戏的操纵特性，这个关卡需要做些什么？另外，在设计一个关卡时，要依赖其他已经事先完成的关卡，设计人员需要考虑如何让这个关卡玩起来和其他关卡显得不同：如果上一个关卡是一个以动作为主的关卡，本关是不是一个以“思考”为主的关卡？这一关里面的探险和探索成分是不是比让玩家练功的成分要多？



图 21.3 使用技术更为成熟的 Quake III Arena 引擎的关卡，在制作中耗费的时间要比简单 2D 游戏的关卡多。因此对 Q3A 关卡进行修改所花的时间要明显多一些

在关卡设计开始前，设计小组应该把游戏中不同的游戏组成成分进行汇总和分解，从而让每个设计人员完全了解游戏的功能。在开始设计之前，每个关卡设计人员必须了解玩家将怎样玩他负责的关卡。在某些游戏中，大规模修改关卡的界面是很容易的，比如在基于地图的游戏 StarCraft 中。如果某个关卡有问题，这个关卡很容易返工重做。但是对于使用 Quake III 引擎的游戏，一旦建立了某个关卡后再大规模地修改这个关卡，将会是很耗时的工作。如果关卡出了问题，让制作者再投入一个月的时间去重新对关卡进行建模，这会是一件很令人头疼的事情。因此事先了解游戏和关卡的操控方法是相当重要的。我们对游戏关卡的玩法分成下面几种类别：动作、探险、解谜、剧情叙述和美工；这种分法可能过于简单，但是确实行之有效。





21.2.1 动作类

大部分游戏含有相当多的动作成分，对于一些游戏，动作成分实际上是游戏关卡存在的惟一理由。当然，还有一些完全摒弃了动作成分的游戏，例如冒险类游戏和解谜类游戏。但是几乎所有其他的游戏都包含一些动作成分，无论是疯狂的射击游戏 *Doom*，在蘑菇上面蹦来蹦去的 *Super Mario 64*，痛宰异型的 *Fallout*，还是超速掠过竞争车辆的 *San Francisco Rush*。

无论正在开发的游戏中的动作成分是什么，关卡设计人员都要去了解当前关卡包含多少动作成分，了解关卡中玩家碰到动作成分的间隔和频度。设计人员还需要清楚地知道，在设计的关卡中动作成分的百分比是多少？玩家将要进行多少场战斗？战斗是否相当激烈？在主要的战斗中是否存在间歇？玩家在整个关卡中是否由于怕死而一直都处于兴奋恐惧状态？当然，游戏中动作成分的量要取决于游戏的风格；尽管如此，开发人员还是需要对游戏中玩家可能遭遇战斗场面的数量有清晰的了解。

对于包含大量打斗成分的游戏，在关卡设计中设计人员要知道如何结束这些战斗。设计人员需要了解：游戏中敌方 AI 的功能和什么样的地图会导致最有趣的战斗？什么样的地形会让玩家有很多地方去蹲下或者卧倒，从而避开敌人的火力？关卡如何设计才能让玩家发挥更多的主观能动性，从而用他们的策略击败对手？只有在清楚本游戏的游戏动作种类并且了解如何让这些动作发挥得更好之后，才会让动作游戏更具有可玩性。

21.2.2 探险

在没有战斗的场合，玩家会做什么？探险。探险是很多动作/冒险类游戏的主要部分，例如 *Tomb Raider* 或 *Super Mario Bros*。设计合理的探险场景会让玩家觉得非常开心，这时候，探险并不仅仅是连接两个动作场景之间的桥梁。在一个地图上辛辛苦苦开发几个月的开发小组往往很难看到这一点。由于他们过于熟悉这些场景建筑，从而对在场景中探索的乐趣熟视无睹了。别忘了玩家在第一次游历一个地图的时候，那种在虚拟世界中探索的感觉是非常刺激的。因此有时候把自己的关卡演示给没有见过这个关卡的访客或测试人员是非常重要的，同时可以得到他们关于游戏中好玩和不好玩之处的反馈。

设计人员需要了解玩家将会如何在关卡中行进，这样就会知道如何把最好的关卡表现出来。例如，玩家在下一个拐角会看到什么壮丽的场景？玩家发现一个新的区域后，他的兴奋程度或恐惧程度是怎么样的？在创造令玩家兴奋的建筑之前，需要建立令人兴奋的探索方式，并把这种方式作为游戏的一个部分来进行考虑。这也取决于游戏的流程，并取决于玩家进入一个令人兴奋的新场景需要满足的条件。进入这种场景有两种方式：一种是一下子就把玩家丢到看起来很漂亮的建筑中间；另外一种则相反，先让玩家在地图的其他区域摸索，最终引导玩家到达这个漂亮的建筑，这样，这个建筑就像对玩家探索的奖励，这种方法显然比前一种方法更让玩家兴奋。

有一些观点认为，游戏中探险场景的制作决定了关卡的流程。让我们考虑以下问题：玩家在到达游戏的主路径之前需要在路径的分支中探索吗？玩家在游戏中是否只有一条路径可以选择？是不是玩家完成游戏关卡的路径看起来非常明显？是否玩家需要在游戏场景中寻找一番才





能找到这条道路？动作型为主的游戏一般喜欢直来直去，给玩家提供的路径就是一条直通下一个战场的道路。一般来说，那些鼓励玩家在地图上闲逛的游戏提供的路径就稍微隐蔽一点。

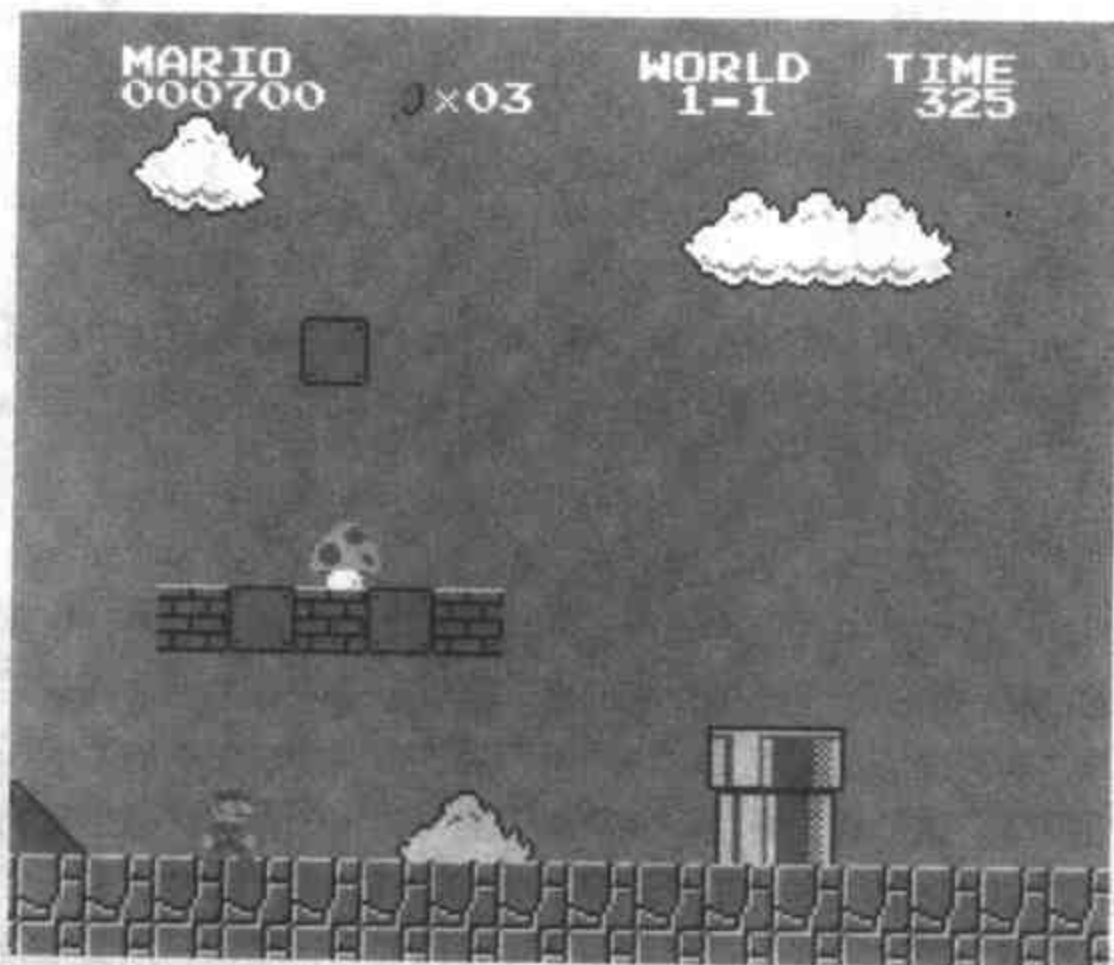


图 21.4 在早期任天堂游戏机上的 Super Mario Bros 游戏，Miyamoto 的游戏把探险当做关键的游戏操作因素

我曾经听过一些人批评 Shigeru Miyamoto 公司的游戏，说它们都是些探险类游戏，因此没有非常好的游戏。但是稍后出品的 Mario 的重点就是探险，而且非常成功。这些批评者犯的错误就在于，他们认为探险不是游戏中好玩的部分，这将受到数百万 Mario 爱好者的反对。玩家感到接受探险中遇到的挑战是值得的，并可以从中获得乐趣，这要归功于 Miyamoto 出品的游戏在这一点上的努力。

21.2.3 解谜

有时候，在某个关卡中，玩家需要完成的任务仅仅是寻找通往另一个区域或关卡的入口。需要玩家在满足特定条件后，才能打开游戏中的一个门或清除路上的一个大障碍物。也许这一类游戏中最糟糕的就是那些所谓“拨开关”的谜题，我们可以在很多第一人称视角射击游戏中见到这种例子。在这些游戏中，也没有什么特殊的理由，设计者让玩家在巨大的场景里面跑来跑去——仅仅是为了找到一个开关并把门打开。打开一个开关，就打开了通往另一场景的门，然后玩家又需要新的场景里面寻找开关……如此循环。这个开关可能是一把钥匙或者其他什么东西，从而打开关闭的门或者其他阻碍玩家前进的设备。这是动作/冒险类游戏中最简单的解谜形式。这种解谜方式一般是让玩家在地图中探索，然后提供一些谜题让玩家解开，而解谜的方法一般都是毫无意义的。以开关为例，一旦玩家找到了这个开关，剩下的事情不过是摁下这个开关而已。





门和开关的问题总是一对矛盾，其实在游戏设计中我们可以用更为巧妙的方法来实现解决矛盾的方法，这需要玩家实实在在地进行一番思索后才能解开这对矛盾。假设设计出下面的谜题：游戏中有一个激光发生器，玩家必须把激光经过多次反射后投射到某个角落，才能在游戏中继续前进。为了正确地反射光线，玩家需要移动一些反射镜。因此玩家必须具备一些基本的物理知识，这样才能控制光线的反射角度。这时，游戏的重点就从“仅仅需要找到谜题”而进一步转变到了“需要玩家去思考解开谜题的方法”。这样玩家的游戏经验也得到了增长，而不仅仅是设置一些无聊的谜题去拖延游戏结束的时间。设计人员应该了解在自己负责的关卡中解谜的重要程度是多少，尤其要结合整个游戏的整体情况考虑。如果游戏中前面的关卡都是以动作为主的关卡，然后突然蹦出几个莫名其妙的谜题，这样会让玩家感到不适应。



21.2.4 剧情叙述

故事背景环境在剧情叙述中占很大比重，并且关卡是建立游戏故事背景环境的重要元素。因此，关卡是讲述游戏剧情的一个不可分割的部分。很多游戏在关卡之间或关卡内部有一些描述剧情的过场动画，如果过场动画不足以讲述所有的剧情，那么游戏的关卡和剧情之间的良好协调就显得更为重要了。这取决于游戏中剧情叙述的方式；有可能玩家会碰到一些游戏中的角色并与之交谈，就像 *Half-life* 中或者在几乎所有角色扮演游戏中那样。因此在关卡中设置那些游戏中的角色就变得很重要了。我们从一些游戏中可以明显地看出，从关卡设计的开始就考虑到了剧情支持的需要。例如，在 *Myth: The Fallen Lords* 游戏中，在特定关卡里面，玩家的目标和剧情的发展是密切相关的。在历史战争游戏如 *Gettysburg!* 中，玩家进行的战役必须和历史紧密相关，否则这个游戏就不能称做历史模拟游戏了。

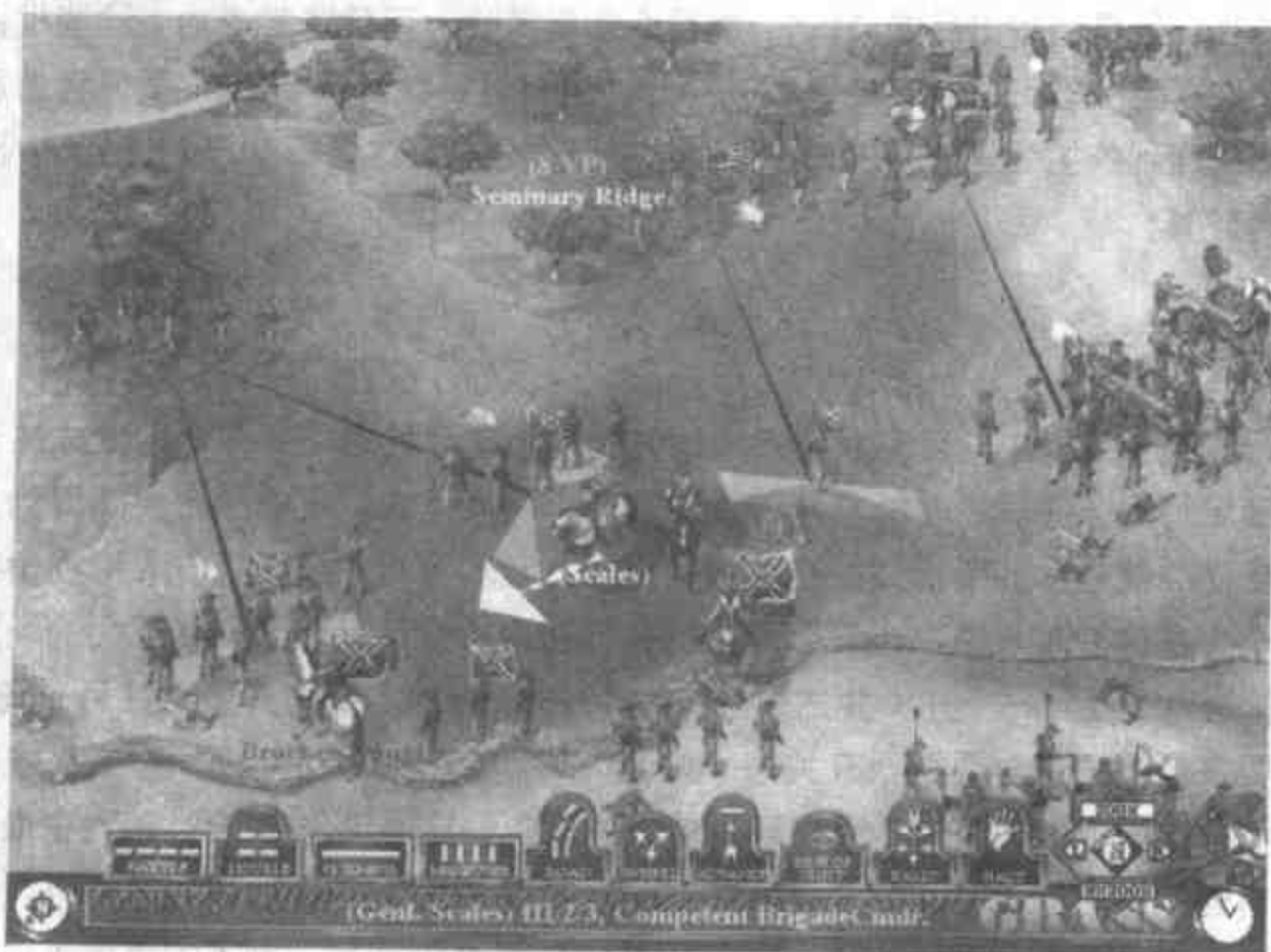


图 21.5 在历史游戏 *Gettysburg!* 中，游戏的风格和特定历史故事紧密相关



为了让剧情叙述更为连贯，设计人员在建立关卡之前需要了解这个关卡的剧情目标，这一点是至关重要的。不过，也要给剧情留一定余地，从而允许设计人员在自己的关卡中发挥创造能力去建立最好的关卡。这也和游戏的可玩性有关，和游戏的策略、动作、解谜及探险成分的平衡有关，因为在关卡真正被制作出来之前，要完全平衡这些元素几乎是不可能的，因此关卡设计者不要被严格的剧情捆住了双手。实际上，在关卡中，剧情常常由于游戏可玩性的缘故而进行修改，但是了解某个特定的关卡大致需要讲述什么内容还是有必要的，这样可以使这个关卡中的剧情和全局性的剧情讲述更为匹配。

21.2.5 美工

很多关卡设计人员的工作是为了满足关卡的视觉需要和听觉需要，这是设计的幕后驱动因素。当然，如果一个关卡的外观对关卡的整体效果至关重要的话，我们自然不会对此提出批评。但是，如果设计人员只关心关卡的视觉表现，那美工因素就会成为一个问题了，这种情况经常会对关卡可玩性造成不利影响。假设在关卡中设计人员耗费了大量时间去建立一个非常壮观的大教堂，在设计人员的脑子中只考虑如何让这个教堂看上去更好、更壮观。但是如果玩家很难找到这个教堂怎么办？游戏中的AI找不到通往教堂的路径怎么办？也许整个结构超出了游戏引擎能够支持的范围，从而导致游戏运行速度变慢，这时候又怎么办？如果这个教堂看上去非常壮观，并且设计人员耗费了大量的工时去建立它，有谁忍心把它从游戏中去掉呢？也许可以把其壮观的截图贴在游戏的包装盒背面；但令人遗憾的是，这个关卡一点都不好玩。

我们前面讲述过，关卡设计人员的一项重要任务就是平衡关卡的视觉表现与关卡中其他必需元素之间的比例。经过细心调整各元素的比例，一定可以达到一个合适的状态，使得关卡看上去也不错，玩起来也流畅，渲染起来速度也很快，并且满足游戏剧情的需要。关卡设计者往往要耗费大量的时间去学习特定引擎和关卡编辑器的诀窍。如何用最少的多边形使游戏看上去还是很不错？这里提出的解决方法是宁“假”勿“真”。我们知道，为建造虚拟世界而创建的关卡，其目的都是为了造“假”，因此关卡设计人员不用担心是否某个效果是通过“伪造”而得到的。只要玩家分不出这个东西是假的或者看不透设计人员使的“障眼法”就可以了。所有特殊效果的宗旨都是为了建立一些和需要模拟的目标看上去既像又不像的东西。关卡设计人员的工作也是使玩家认为某件物品看上去既像又不像；这给关卡赋予了一层华丽而又奇特的外衣，离开了这件“外衣”，这个关卡会变得索然无味。

关卡的可视部分会深深影响到前面提及的关卡的其他方面。例如，为了让关卡的可玩性更强，在纹理的安排上应该做到让玩家可以看到哪些地方通不过，哪些地方能通过。比如游戏中有一个斜坡，如果需要玩家去猜测是不是太陡峭而爬不上去，这样的设计显然不太好，而应该用不同的纹理贴图去提示玩家哪些斜坡是可以通过的，哪些是不能通过的。灯光可以用来隐藏秘密场所，关卡中的主要谜题可以设计成让玩家考虑把灯打开的办法。如果关卡的目的是为了奖励玩家在游戏里的勤奋探索，可以把那些特殊区域的外观设计得令人难忘，这样能够维持玩家在这个关卡中的兴趣。

关卡中的美工会耗费大量的时间。耗费的时间量大致与引擎和所用关卡设计器的复杂程度成正比，同时也和关卡中期望的视觉效果成正比。实际上，可以先把游戏的玩法和剧情设定好，





然后再花几个星期去逐渐完成视觉效果。灯光可以在最后才调整，初期使用的纹理最后可以用其他纹理替代，多边形可以在最后调整得更精细，从而达到预期的设计目的。设计人员必须始终了解，关卡中效果的变化会影响游戏的可玩性。

21.2.6 平衡所有的元素

由于一个出色的关卡必须平衡动作、探险、解谜、故事叙述和美工，因此关卡设计人员的工作就是平衡这些因素。即使某个关卡从某个方面看起来很不错，但是否符合剧情？游戏剧情有没有要求这个关卡不能有太多的打斗场面？打斗场面在游戏中的重要性如何？本关卡能够完全没有打斗场面吗？是否关卡中的解谜元素过多而影响玩家在探索中获得的乐趣？关卡中动作、探险、解谜、故事叙述和美工这些元素成分是相互制衡、互相依赖的，因此设计人员必须时刻关注并调整这些元素的平衡。要设计出良好的关卡，就需要时刻关注各元素的平衡。



21.3 关卡流程

不同类型的游戏，所完成关卡的流程是明显不同的。我们以动作/探险类游戏 *Super Mario 64*、*Tomb Raider* 和 *Doom* 为例。虽然这三个游戏的操控方法明显不同，但是三个游戏中关卡的功能实际上是相当类似的。在这些游戏中，玩家通关的方式都是从一个不同的起点到另一个不同的终点。玩家大部分时间都是在关卡包含的场景中探索，因此，一旦玩家玩通一个关卡后，往往玩第二次时就没什么乐趣了。不仅如此，玩家在关卡中什么地方该碰到什么敌人都是已经设定好的。每次玩家玩同样的关卡时，他的感觉和上一次玩这个关卡没什么两样。这种关卡的流程多少有点线性化，从 A 点到达 B 点往往只有一条路径可选择。

RPG 提供的游戏方式大致和上面讨论的动作/探险类游戏相似，不过可能加上了一些非线性的方式。设计人员通常倾向于让玩家按照指定的方式到达指定的地点。RPG 的非线性方式比动作/冒险类游戏要多一点，通常允许玩家选择不同事件发生的顺序。一般来说，“hub”风格的游戏允许玩家选择不同的路径出外进行冒险，最终回到游戏的中心点，例如回到某个城镇。如果愿意，玩家也可以呆在城里一直锻炼技能。虽然如此，总的来说，RPG 游戏的关卡流程和动作/冒险主题游戏的关卡流程还是非常相似的。

然而，战略游戏中的情节很少是一成不变的，例如在 *WarCraft* 或 *Civilization* 中，一般很少有固定的关卡流程。*WarCraft* 和 *Civilization* 与 *Super Mario 64* 和 *Doom* 在关卡内的流程方式是不同的，但是从总体上看，这些游戏使用关卡的方式是一致的。探险不是这些战略游戏的重点，战斗可能发生在地图的任何角落。并且，在不同的场所，如果采用正确的策略，能发挥出更大的效能。战斗发生的场所也可能从一个地点转移到另一个地点，玩家和他的对手都有可能没有走完地图的所有区域。在这种地图上，游戏的玩法比在动作/探险类游戏的地图上更缺少预定性。游戏的流程更为模糊。



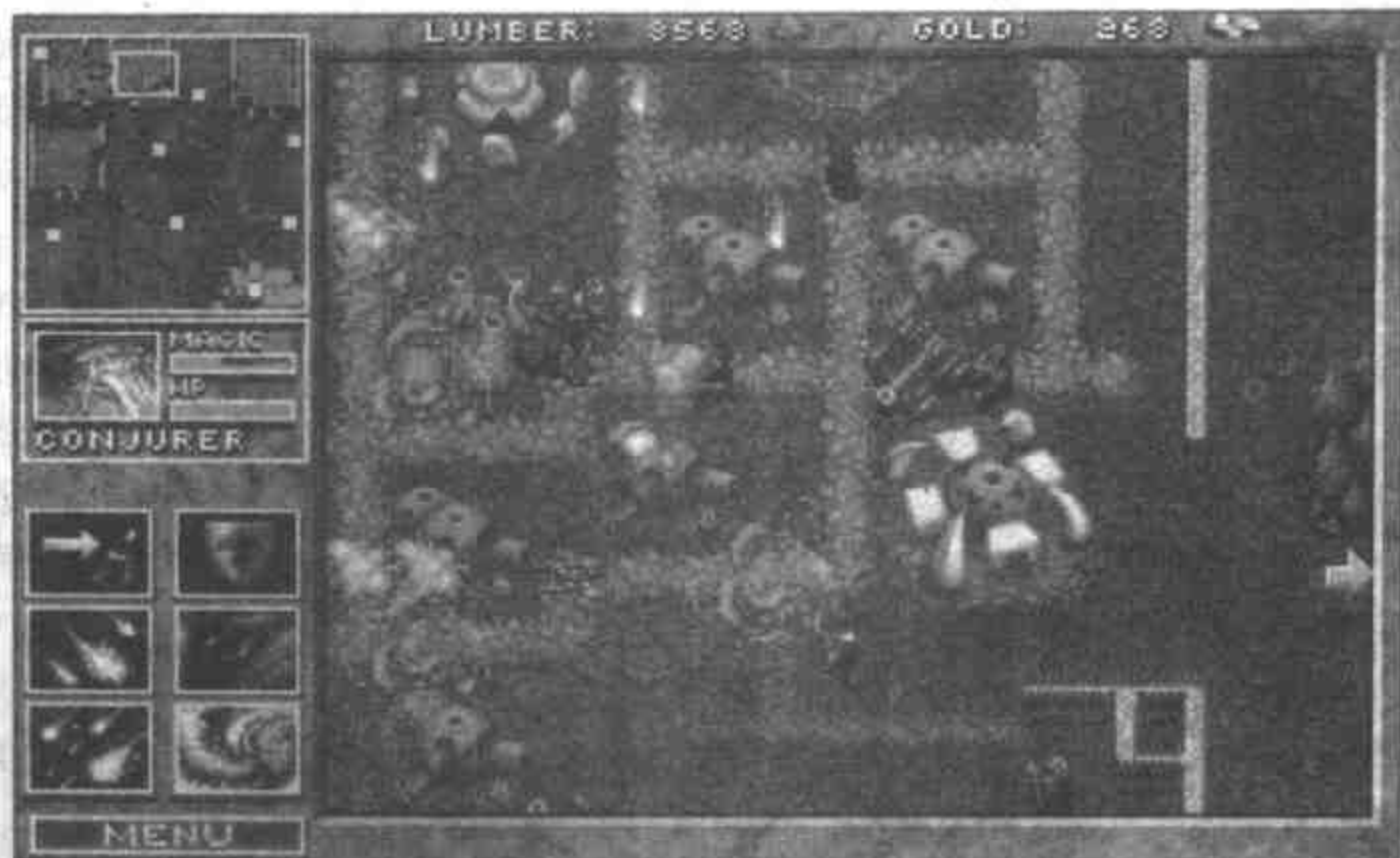


图 21.6 实时战略游戏 Warcraft 关卡流程的自由度比动作/探险类游戏的多：在地图的任何地方都可能发生战斗

当然，Civilization 游戏与 Warcraft 游戏至少有一个明显区别。在 Civilization 中，任何玩家都一直在一个关卡中进行游戏。也就是说，玩家在开始 Civilization 的一场游戏之后，直到知道胜利或者失败为止，他都一直在这个关卡里面。但是，在 Warcraft 中，玩家需要通过一系列剧情关卡。从玩家的角度来看，Civilization 代表着一种更为连续的游戏过程，这样更能令玩家上瘾。Warcraft 之类的游戏给玩家提供了容易停止游戏的地方，也就是各个关卡的末尾，而 Civilization 游戏就没有这种分隔点。这两种游戏所包含的关卡可能都具有无法预料的流程，不同的玩家在同一关的玩法可能有显著的不同，但是由于 Civilization 的玩家把所有的时间都消磨在同一个地图上，因此对游戏的整体感觉是根本不同的。当然，事实上 Civilization 是回合制游戏，Warcraft 是实时的，实时战略游戏同样也会改变游戏流程，但这只不过是对游戏方式的改变，而不是改变了关卡设计和使用的方式。

让我们回头再研究动作/探险类游戏。如果读者曾经玩过 Quake 的联机多人死亡竞赛模式关卡的话，就会了解，在这种情况下，关卡的流程实际上和战略游戏的非常相像。也就是说，在这种方式下，对关卡探索的重要性降低了，并且战斗随时可能发生在地图的任何场所。实际上，多人联机游戏的许多玩家都喜欢找一张自己喜欢的地图并且一直玩它，最少在一段时间内。如果玩家想在这张地图上获得胜利，必须事先把整个地图游历一遍；不过，当然是和有经验的玩家竞赛的时候。如果玩家想在以后的战斗中取得胜利，必须游历并且记熟地图；对地图进行探索并不意味着游戏的结束，这和单人游戏不同，单人游戏的探险是玩游戏的快乐之一。

体育竞技类游戏在游戏的玩法上是典型的非线性的，除了赛车游戏之外。例如棒球游戏的关卡流程和死亡竞赛或者战略游戏的关卡流程相当类似，但是与动作/探险类游戏的关卡地图不同。整个关卡中都是动作成分，玩家在关卡地图中来回运动，因此在地图中的位置是无法事先预料的。这时对关卡的探索就相对不重要了，因为这种关卡的形状都比较简单，一般我们在屏幕上看到的是全部或者大部分的体育场。





在赛车游戏中，玩家从明显的开始位置移动到明显的终点为止。这种移动类似于基于探索的动作游戏，例如 *Doom*，其中关键的区别在于赛车游戏的起点和终点一般是相同的（封闭赛道），并且关卡中常常需要玩家在赛道上重复跑好几圈才结束。这种流程和动作/冒险主题游戏的流程一样，是线性的。现代赛车游戏，例如 *San Francisco Rush* 或者 *Cruisin' World*，加入了一些动作/探险游戏中的探险性元素到游戏中，使关卡提供了各种视觉效果，并且有些视觉效果惊人，例如当玩家第一次拐过一个拐角，可能会看到壮丽的场景，并感到震撼。老式的赛车游戏如古老的 *Pole Position*，只是让玩家在和对手的挑战中获得乐趣，而不是让玩家驾车通过全新的令人惊讶的场景来获得乐趣。许多现代赛车游戏也提供了多条路径，玩家可以通过走捷径来改变游戏结果。虽然游戏的流程还是单一的，有一个共同的目的地，但是有一条支路可以让玩家选择总比强求玩家只能走某条路线要好。

在讨论完游戏的类型和它们分别对应的关卡流程后，可以把游戏大致分为两大类：线性关卡为主的类别（动作/冒险、角色扮演和赛车）和非线性、预料事件较少的关卡为主的类别（战略、体育竞技和多人联机死亡竞赛游戏）。当然，并不是说这两种类别就不能重叠了。例如某些 *StarCraft* 关卡尽量鼓励玩家按照指定路线前进，尤其在建筑内行进的关卡。同样的，一些 *Super Mario 64* 的关卡地图提供多条分支路线供玩家选择。如果设计人员努力发挥创造性，这两种类别的关卡就不会如此界限分明，会带来更加有趣、更多选择的游戏可玩性。



21.4 优秀关卡的元素

在设计一个关卡的时候，似乎有无穷无尽的细节需要设计人员去注意。设计人员既要关心动作、探险、解谜、剧情讲述和视听效果，又要和美工、程序设计员一起去达到设计意图。对于 3D 关卡，还必须确定整个关卡是否能被优化，从而能在预定系统中运行良好。而且还不得不去对付难以驾驭的设计工具，这些麻烦的设计工具经常妨碍设计者的理想设计。

在设计关卡的时候，设计人员确实有一系列的规则需要去遵守，即使她没有把这些规则写下来。每个设计人员在头脑中都有自己的设计准则，准则规定了“可以这样”和“不可以那样”，但是这些准则是随着项目的变化而变化的。有些游戏设计小组在设计之前会制定出一些“设计规则”，设计人员将要遵守这些规则。但是也有一些适用于各种项目的通用规则。现在笔者列出自己的一些设计规则，我曾经用这些规则设计过比较刺激的关卡。



21.4.1 不要卡住玩家

这一点应该是明显的。玩家在你设计的关卡中不应该被无助地卡住。关卡中不应该有让玩家掉进去就爬不出来的陷阱，不应该有什么东西在被错误的移动后就永远地阻挡玩家前进的步伐，并且不要设计这样一扇门，当玩家通过其他路径到达门口却发现打不开它。虽然这个目的看上去很明显，但也需要耗费关卡设计人员大量的时间去满足要求。假设有一个谜题需要玩家用一些炸药在墙上炸一个洞才能继续前进。那么，如果玩家炸错了目标又用完了炸药怎么办？如果不提供更多的炸药，玩家就被永远卡住了。同样地，假设玩家需要和某个特定的 NPC 交谈



后才能拿到某个特定的物品。如果玩家还没交谈就把 NPC 杀了会怎么样？要么游戏立即结束，要么提供一个可替代的方案让玩家继续游戏。应该按这个思路去设计关卡：无论玩家做了什么，他都能继续完成关卡任务；这需要仔细的考虑和计划。作为关卡设计人员，必须经常考虑：如果玩家按其他方法去玩游戏会怎么样？

21.4.2 设定子目标

当玩家玩一个关卡时，应该提供一些易于理解的子目标给玩家。让玩家认识到完成一个个不同的子任务对最终目标是有帮助的，而不是在整个关卡中只提供一个大的目标或者出口让玩家去完成或达到。这里有一个很简单的例子，就是在 *Doom* 中不同的钥匙。玩家知道，一旦拿到蓝色钥匙，就更接近关卡的终点。在大多数街机赛车游戏中，如 *San Francisco Rush*，游戏在一个赛道中提供多个断点，而不是只让玩家从起点一直跑到底，达到一个断点就给玩家一些奖励时间，这相当于鼓励玩家：你刚才做得很好。在 RPG 中，玩家的任务致力于消灭肆虐大地的邪恶力量，但是在此途中，他可能会碰到一些子任务，如村民们需要他帮忙做什么事情。这些子任务引导玩家通往更大的目标，同时给予玩家积极的反馈，告诉玩家他真的在游戏中玩得很棒。如果玩家不能了解他做到了什么，那么这种子任务就是无意义的。因此，达到某个目标后，适当地给玩家一些奖励也是同样重要的。这些奖励也许是满足视听感觉的钟声和喝彩声，也许是一件新的武器、奖励分数或赛车的奖励时间。如果游戏设计人员没有提供子目标或者这些子目标不够清晰，最终会导致玩家不清楚自己到底完成了什么事情，玩家可能拒绝去做要求做的事情或者不了解他离最后的成功到底还有多远。



图 21-7 赛车游戏 *San Francisco Rush* 系列，通过设置断点的方式给玩家提供子目标，并给通过断点的玩家奖励更多的时间。这个截图来自于 *San Francisco Rush: The Rock Alcatraz Edition*





21.4.3 路标

越复杂的关卡越容易使玩家在其中迷失方向。除非让玩家困惑是关卡的目的（一般很少有这种设计目的），一般来说，最好在关卡中设置一些好记的路标来方便玩家的游历。路标可以是游戏中任何独特的物体，当玩家第二次见到就能认出来，可以是一个特殊的装饰华丽的房间、一个大雕塑或者是一个热气腾腾的火山熔浆池。这样，在探险的时候，当玩家回到这个路标前面，就会知道回到了以前来过的地方，因此就容易了解关卡的布局。路标当然不必是一个上面写着“断点 A”的大红招牌，而应是更符合游戏环境和关卡环境的物品。

21.4.4 主线

虽然笔者相当支持非线性游戏，但是也很喜欢那些具有良好主线的关卡。主线可以给玩家以方向感，玩家可以沿着主线去完成关卡任务。这里的方向可能是实实在在的方向，例如“朝东”或者“朝着彩虹”，也可能是更为隐蔽的目标，例如寻找一个生物并打败它，或者取得某样重要物件。给玩家提供一个主要目标去完成，这对关卡的可玩性是至关重要的。应该给玩家提供一个目标，并且就像前面讨论过的，让玩家从一些子目标达到主要目标。应该让玩家了解他当前的目标和子目标是什么，并且明白怎样做才能推动关卡进度的发展。单独可选的附加目标可以稍微隐蔽一些，但是没有比不知道自己该干什么更让玩家感到沮丧的。设计一个制作清晰的主线，可以防止玩家迷失方向。

21.4.5 减少回头路

假设一个游戏以探险成分为主，其中玩家不得不为了继续游戏而走回头路，并通过已经走过的路径回到主线上来，这样的设计就不是很好。当然，这并不是说关卡的设计不能包含让玩家探索的分支路径，而是说每个分支都可以回到主干上去，而不需要玩家沿着走过的路走回头路。如果游戏中角色扮演和冒险的成分比较多，而且在游戏中需要的拟真度比较高，那么设计一些回头路还是可以接受的。当然，在即时战略游戏或体育竞技游戏中，玩家需要在同样的场所上来回移动，这是因为棒球游戏或者 WarCraft 的探险成分没有 Super Mario 64 高。在探险游戏中，Super Mario 64 在这方面就做得很好，完全没有回头路可走。

21.4.6 首次通关的条件

如果大多数玩家第一次玩某个关卡就可以顺利通关，说明这个关卡的设计可能有点过于简单。虽然如此，也可能是某个玩家在第一次过这个关卡时使用了各种手段去通关。然而，并不是说玩家仅仅是凭运气才能顺利过关。相反地，设计人员应该提供足够的资料给玩家，如果玩家观察能力足够仔细，反应足够快，就能够有机会绕开道路上的障碍物。当玩家在这个关卡遭受失败之后，要让玩家感到只要她更仔细，或者比上次考虑更周全，就能有更多的机会去避免失败。如果玩家发现通过这个关卡的惟一办法只是靠瞎撞运气的话，只会让玩家感到沮丧。当然，也可能这个关卡难度过高。关卡中的线索可能过于隐蔽，要打败的怪兽可能真的很强，需要进行的选择可能挑战性真的很高，但是如果玩家正确地处理每一道难题，也有可能一次就通





过这个关卡。

21.4.7 清晰标注导向区域

在关卡中，应该让玩家清楚哪些地方可以通过。会在上面滑倒的斜坡的坡度应该明显比那些可以在上面行走的斜坡要大。应该用纹理贴图告诉玩家什么地方可以通过，什么地方过不去。如果一个区域看上去无法通过，但是玩家发现只有通过这里才能到达某个特殊场所，这会让玩家感到失望。我们举例说明：比如存在一个有十个门的房间，玩家试了其中三个门，结果发现它们都是锁着的。这时玩家可能会推断出这些门只是为了展示用的，便不再去试开其他的门。当试过前三个门发现都是锁住的后，又没有其他明显的信息提示玩家其他的门哪个可能可以打开，如果此时离开房间的惟一出路就是通过其中恰好没被锁住的门，那么我们就认为这种设计是非常差的。找出离开这个房间的惟一办法是一场非常乏味的考验。游戏的乐趣在于试着进入某个区域或者在这些区域移动时得到的刺激，如果需要玩家找出设计人员强制设定的哪些路可以走，哪些路不可以走，就不怎么有趣了。

21.4.8 提供多种选择

给玩家提供选择的好处是显而易见的，但是有些关卡设计人员在建造关卡的时候常常忘记这一点。好的关卡会让玩家选择完成目标的方法，就像好的游戏玩法提供给玩家大量的选择以决定如何玩这个游戏一样。选择并不意味着一定要有多条路径可以通关，虽然这也是一个好主意。在第三人称射击游戏中，提供选择意味着给玩家不同的方法去消灭某个房间里面的敌人——提供大量的场所让玩家躲藏、不同的地方可以射击敌人等等。这样的设计可以让玩家使用各种策略去消灭超级怪兽。这种选择也可能是对玩家的一些奖励，并鼓励玩家去得到这些奖励，例如在熔浆池中间放置一个火箭筒——玩家可以自己决定去不去冒险得到它。在战略游戏中，可以提供一些有趣的选项，如提供某个场地让玩家在此处开展战役，或者提供某个场地让玩家重整军队或者采集资源。在冒险类游戏中，最差的设计莫过于不给玩家提供足够的选择；多种选择意味着提供解决谜题的多种方法，不同的角色可以进行交谈，不同的方法可以通关。玩家最讨厌的就是只能用一种方法才不会被游戏卡住，尤其是这种惟一的方法是他们不喜欢的方法。

21.4.9 个人规则

当然，上面提到的规则是远远不够的。当从事关卡设计工作时，关卡设计人员有必要时刻记住自己的规则列表。当设计出一个被同行和玩家都称道的关卡后，要尝试着去总结分析自己到底在这个关卡的哪些地方做得较好。然后把这些优点提炼出来形成自己的个人规则，并应用到以后设计的关卡中。不一定要求把这些规则写在纸上，记在脑子里就足够了。读者可以以上面列出的规则为基础，也可以总结出一套完全不同的规则。毕竟各个设计人员有着自己的设计风格。





21.5 设计步骤

不同的设计人员在建立关卡时遵循不同的步骤。适合某人的步骤并不一定适合其他人。也就是说，下面将要讨论的步骤只是比较适合笔者。在开发的时候，我并不一定百分之百地按照这些步骤进行，但是总的来说，与没有合理规划和设计就匆匆忙忙去制作关卡相比，采用这些步骤会让工作变得更有条理、更有效率。



第 1 步 预备工作

在开始设计游戏关卡之前，要考虑如下的问题：是否这个游戏的玩法已经基本确定下来了？有没有可能在以后对这个游戏进行大规模修改，从而导致设计出的关卡没什么好玩的？或者考虑更糟糕的情况，是否这个关卡会被砍掉？例如，假如设计人员正在设计一个第三人称视角动作/冒险类游戏，如 Tomb Raider。在开始设计游戏中的某个关卡之前，需要考虑游戏中的主人公会有一些什么样的动作。是不是以后会给角色加入很多动作？游戏中的英雄是不是有能力做一些前空翻动作，从而显著提高她能跳过的距离？如果游戏本身的制作还需要很久时间才能完成而设计人员已经开始设计关卡，那么游戏中主角的动作通常会被改变。如果设计小组知道游戏主角的动作在以后的设计中会做根本的改变，那么要求设计人员过早的开始关卡设计就是非常不明智的。

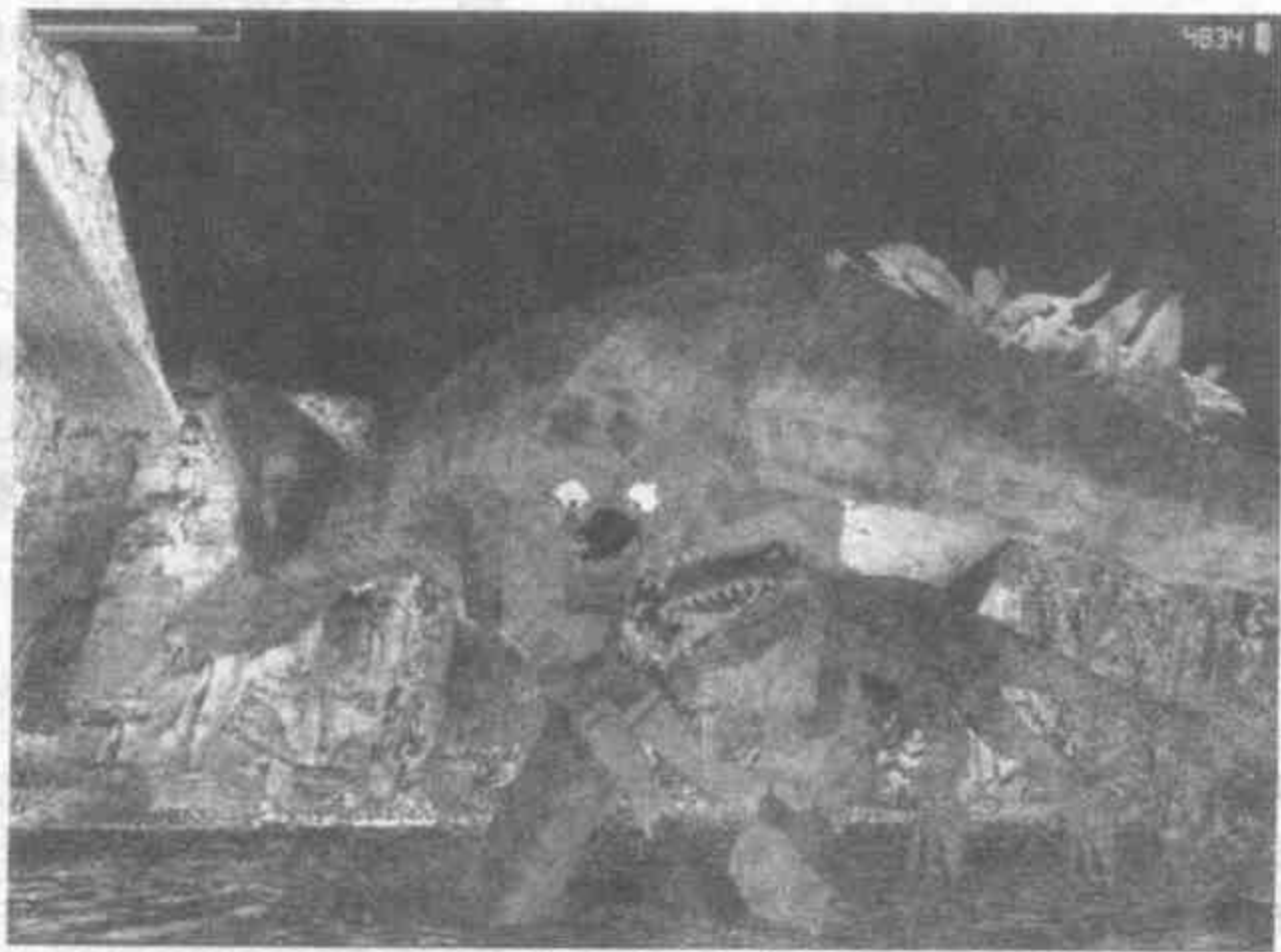


图 21.8 在开始设计类似于 Tomb Raider 系列的游戏关卡之前，需要明确地定义好主人公使用的一些动作

在本人曾经参与的一个项目中，我们在主角的跳跃动作还没加入到游戏之前就开始了关卡制作工作。后来我们受到了惩罚，一旦主角被加入了跳跃动作，我们只能对关卡进行返工，回





过头去在修改关卡，从而加入一些能够利用主角跳跃的场景。很不幸，在游戏中加入跳跃动作后不久，我们认识到跳跃并不怎么有趣，结果我们又要重新把上次加入的有关跳跃的场景逐个从关卡中删除。这还不如一开始就搞清楚游戏中的跳跃到底是怎么回事。这个悲剧的教训就是，在游戏机制还没有最终敲定和实施之前，我们就开始了关卡制作。就像在第 13 章“Getting the Gameplay Working”（实现设计方案）中讨论过的一样，在实施游戏操控性的同时，先制作出一个关卡就可以了，这样可以在这个关卡中测试游戏中加入各种动作后的表现。如果此时多个关卡一起开工，实际上是在浪费时间，并且最后会对项目产生危害。此外，一旦游戏操控性被最终确定后，最好废弃掉测试用的关卡，因为测试用的关卡质量往往离设计人员的最佳设计要求差得很远。

第 2 步 描绘草图

在开始关卡制作之前，从游戏操控和剧情角度去了解这个关卡需要做什么是很重要的。需要考虑玩家在这个关卡中将要面对什么样的挑战？什么样的环境最适合这种挑战？本关卡中游戏的操作令人兴奋或令人头疼的程度是多少？在哪里需要给玩家一些奖赏？这个关卡需要展示的剧情成分有多少？不仅在计划阶段，而且在任何时候都必须清楚游戏的重点是什么，了解设计中关卡应该如何支持游戏的重点。

有一个很好的办法：一旦设计人员对某个关卡要完成的目标有一些了解后，立刻在草稿上勾勒出这个关卡的草图。这样可以避免以后让自己的设计陷入“绝境”。假设某个设计人员正在设计 3D 第一人称视角射击游戏的一个军事场地。在这个场地里面，需要包含具有一个大发电机的房间。在开始建模后，他首先制作了所有的大厅，然后在制作发电机房之前，就开始制作其他看上去很酷的房间。最后才发现，没有留出放置发电机的足够空间。发电机房设计得太小，玩家很难找到这个地方。很不幸，给发电机腾出地方的惟一办法就是裁掉很多已经制作好的大厅。这种情况下，很多设计人员就把发电机房放到其他不太理想、不怎么符合逻辑的场所，这样他们可以不用重做已经花时间建造好的结构。当然，绘制关卡的草图并不能完全避免这些问题，但是如果勾勒得正确的话，设计人员会意识到他设计的发电机房太小了，这时候要修改就简单多了，只需要用橡皮就可以了。毕竟修改草图要比修改整个建好的结构要简单得多。当需要把自己的设计给项目负责人审阅的时候，草图也是很有用的，项目负责人也希望浏览一下每个人的设计，从而保证小组各个成员的设计没有出轨，并保证游戏的整体性。

第 3 步 基本建筑

如果设计人员对自己的草图感到满意了，实际的关卡制作工作就可以开始了。制作阶段的范围和时间长短取决于要创建的关卡的复杂程度。例如，使用 2D 平面贴图引擎的关卡就要比使用 3D 引擎的关卡在制作速度上快得多。同样地，所用 3D 引擎的复杂程度也会明显影响建造关卡所耗费的时间。用 Doom 引擎制作出一张出色的地图，需要设计人员苦干 1 天到 2 天。用更成熟的 Quake III 引擎制作同样质量的关卡会令人苦干数周。

这时，别忘了现在所干的工作仅仅是在制作关卡的基本布局，还没给关卡“打扮”呢，灯光和纹理还都没有加上，因为现在还不是给建立的几何体“美容”的时候。在这个步骤中，需





要关注的是玩家是否可以在关卡中顺利导航，并检查是不是所有的场所玩家都可以到达。以此让设计人员了解这个关卡的布局是否正确。



图 21.9 随着游戏引擎越来越成熟，很大程度地增加了建立关卡需要的时间。例如，使用 Quake III Arena 引擎制作的关卡往往需要数周才能完成。

第 4 步 细化建筑结构，直到关卡令人感到有趣为止

现在需要重复第 3 步，直到这个关卡看上去开始有点不错，并且在里面游历开始令人感到有趣。例如，假设设计人员正在开发一个第一人称视角射击游戏，现在就应该使用游戏角色体验一下在关卡中的感觉，看看在拐角间跳跃闪躲是否有趣，跳跃动作的难度是否正常，设计出的场所是否符合自己的要求。从整体上看这个关卡是否满足自己的期望。如果设计人员真的像玩家一样在这个关卡中游历了，并花了足够的时间，而不是仅仅在关卡编辑器中看看这个场景，那么设计人员就可以判断并决定本关卡是否已经满足要求了。如果关卡不满足要求，那么重复修改到满意为止。

第 5 步 基本游戏可玩性

现在，关卡在玩家导航方面已经比较完善了，应该着手实施本关卡将要用到的游戏可玩性了。当然，在前面的几个步骤中我们始终注意了游戏的可玩性，现在是真正检查游戏可玩性是否符合要求的时候。有经验的设计人员能够把关卡的思路和草图最终转化成关卡中有趣的部分。其他人员则从草案开始，再建立一些结构，当加入游戏操作时可能会发现必须对建立好的结构进行修改。有经验的设计人员可以预计到抽象的概念是不是有趣。然而，在成为专家之前，开发过程中难免会犯很多的错误。



图 21.10 在 Duke Nukem 3D 之类的游戏中，建立游戏可玩性的工作包括放置怪兽、武器和配置谜题

关卡的可玩性由关卡中允许玩家执行的动作组成。在第一人称射击类游戏中，如 Duke Nukem 3D，这意味着放入一些玩家将要与之战斗的怪兽和玩家可以捡起物品。在角色扮演和冒险类游戏中，指的是玩家需要解决的谜题、玩家可能与之交谈的游戏角色和这些 NPC 可能给玩家出的问题。在实时战略游戏中，设计人员需要设置玩家及其对手的初始单位的地点和数量，同时设置关卡中要出现的援军。运动和赛车主题的游戏在这个步骤就相对简单一些，因为这种游戏中各关的玩法基本相同，所以不需要对某个体育场或者赛道进行太多的设置。

第 6 步 优化游戏可玩性，直到游戏比较有趣为止

当然，游戏的可玩性决定了游戏的好坏，因此设计人员必须重复第 5 步，直到游戏比较好玩为止。有时候，为了优化游戏可玩性，设计人员会追溯到第 3 个步骤上去做一些修改。有可能出现这样的问题：设计时觉得应该比较好玩的地方，但是由于 AI 的能力，使得关卡不满足要求。或者设计了某种动物，需要让它从悬崖的裂缝中扑出到玩家身上，但是实际上没有设计足够的空间把它藏起来。在开始实施游戏可玩性后，如果发现了上述类似的问题，为了修改这些问题，有可能需要修改关卡中的布局。

对于一些设计人员，修改现有的关卡去适合游戏可玩性可能是一个非常痛苦的过程。例如，某个设计人员建造了一些她认为从美工角度看上去非常不错的建筑。如果这个建筑并不符合游戏所需，设计人员可能会勉强进行返工并对这个建筑的几何结构进行调整，或者干脆降低游戏的可玩性——这显然是错误的做法。无论多难舍，为了让游戏最具可玩性，设计人员必须对自己的这部分工作忍痛割爱。这就是为什么我们建议只需要建立基本建筑框架，而不需要过分细化它的原因；如果需要对关卡进行重大修改，也不至于浪费太多工作成果。





在这个步骤中，制作中的关卡才被真正整合起来，现在可以开始去感觉一下这个关卡是否是个成功的关卡。现在就可以开始在这个关卡中真正的玩起来了。如果到目前为止连设计人员自己都觉得这个关卡还是不怎么好玩，就需要对自己的关卡进行审核，并找出这个关卡不好玩的原因。最糟糕的情况是自己觉得这个关卡一点都不好玩，结果就是需要开始重做这个关卡。然而，在理想状况下，这个关卡会让设计人员觉得自己付出的努力总算有了回报。

第7步 细化美工

现在，这个关卡基本上完成上述步骤了，可以动手对关卡进行美容了。我们讲过，第3步和第4步只是建立了基本的结构，仅仅可以让设计人员在关卡中导航并对关卡有一定认识。现在需要给关卡的结构贴上纹理贴图，并加上灯光效果，然后增加一些装饰物，给关卡赋予视觉效果。许多关卡设计人员在关卡的美工上耗费大量时间；当然，如果有时间，把关卡装饰得越漂亮越好。但是，就像以前强调过的那样，一定要先保证这个关卡已经运行良好并且满足了游戏可玩性目标，然后才能去做这些修饰性的工作。否则，如果把关卡某个区域制作得非常漂亮但是最终又不得不废弃掉，那就是在浪费时间了。在对关卡的美工进行精雕细琢时，千万要注意不要破坏已经设置好的游戏可玩性。

第8步 游戏测试

现在，关卡的各个部分都组装完毕，可以展示给别人看了。让他们测试这个关卡，并且得到他们的反馈。游戏测试是游戏设计的重要步骤，关卡设计也不例外。测试的对象可以是小组成员，但是应该包括一些基本没接触过这个项目的人。对游戏不熟悉的人可以发现很多问题，并且会告诉设计人员自认为有趣的地方是否真的有趣。

关卡测试只需要把关卡交给某个人，请求他试玩这个关卡，并且要求他告诉设计人员他是怎么想的。另外一个适合关卡测试的方法是，和测试者呆在一起并观察他是怎么玩这个游戏的。并观察是否他被卡在你没有预料的地方，是否他寻找路径有困难，是否游戏的玩法让他感到有足够的挑战性。观看其他人玩自己设计的关卡是相当有益的，并且可以及时得到关卡流程是否运行正常的信息。

最糟糕的情况下，关卡测试会反映出设计的关卡没有预期的那么好玩，那么下面主要的工作就是重新修改关卡使其可玩性增强。作为一位设计人员，当别人告诉你这个关卡导航非常困难或者令人迷惑或根本不好玩时，一定不要有抵触情绪。当然，可以再听取一下其他测试人员在这方面的意见，当大家都对游戏某个方面有同样的抱怨时，这意味着他们说确实是对的，关卡中确实有一些地方需要重新编写。许多设计人员，当他们在某个关卡投入了太多的时间和精力后，就很难接受别人的批评了。不可否认，听取别人的意见并废弃掉花费数月心血制造的成果，确实令人心疼，但是这正是游戏测试的目的所在。设计人员需要接受测试人员的评价，认识到关卡中存在的问题，然后重新开始制作关卡。真正优秀的关卡和不太好的关卡的区别往往就是，前者接受了彻底的测试。





21.5.1 开发步骤的各种变通方法

前面描述的关卡设计步骤并不是制作关卡的惟一途径。与早些时候提到过的关卡设计方式一样，这种“要做什么”和“不要做什么”的规则是因人而异的，每个设计人员都应总结出最适合自己的和小组的开发方法和步骤。很多优秀的设计人员使用的设计步骤与我使用的步骤区别并不是太大，但各人都有不同之处，都有一些更适合自己的方法。

有一种比较有效的变通方法就是把第 3 步到第 6 步结合起来。可以从某个特殊的房间或区域开始，先建立局部的结构，而不是一开始就从整个关卡开始制作。在做这个关卡的其余部分之前，先加入游戏操控到已经做好的局部关卡中。如果觉得这个局部总的来说比较满意，再去做剩下的部分。当一个场景制作完毕，就往里面加入游戏操控元素，让这个片断可以玩起来。按照这种方法，即使以后扩大了某个场景的范围并且还要保持游戏操控良好运行，浪费也不会太多，因为本来这个场景的周边区域就没有建立起来。我们以前提醒过，要千万小心，不要把自己的设计做得没有回旋余地。没有人愿意在耗费大量时间制作一个场景并设计好了该场景的游戏操控后，又不得不去掉这个部分，只因为这个场景附近的某个场景需要扩大从而导致空间不够用了。如果需要在建立整个关卡之前为某个场景加入游戏操作元素，那么应该把这个场景或者建筑当做一个完全离散的独立单元考虑，例如某个特殊的建筑或者结构。做到了这一点，才能在做下一个场景之前在这个完整的独立区域中加入游戏操作元素。

另外一个可行的办法是，把流程中的测试步骤提前，有可能在第 6 步之后就可以开始测试了。当一个关卡具备可玩性后，就可以找一些比较信任的人来试玩这个关卡。这个阶段美工可能还没有最后细化，不过可以对试验者说明这一点，这样在关卡制作的早期就能得到一些玩家反馈，使设计人员可以在耗时的美工优化工作之前就对关卡做一些重大改进。然而，过早的测试也有一些弊病，因为测试人员可能不知道视觉效果还没有完成。这样的结果会导致试玩者只顾批评游戏的效果而不能对游戏的可玩性做出反馈。因此一定要和测试人员进行沟通，告诉他们这个阶段你需要什么样的反馈，希望测试人员不要太在意现在是否缺乏光影效果等。早期阶段的测试并不能替代关卡的最终测试，不过可以避免一些意料之外的错误，并且可以使最终测试进行得更为顺畅。



21.6 关卡设计人员配备方法

在整个章节中，我们都假定某个关卡设计人员在制作关卡时，是由他来负责关卡的各个方面的。许多设计工作室在关卡设计时确实还在采用一个设计人员负责一个关卡的方法。必须承认，这样做有很多好处，可以让一个关卡更为整体化。设计人员对关卡中的游戏可玩性、美工和编程保持一致的认识，并且让关卡更为协调。例如在设置关卡的灯光效果时，设计人员知道他的设计意图是让某个场所显得比较阴暗，并配上一些无方向的闪烁的灯光，这样会让游戏具有最佳可玩性。一个关卡自始至终都由一个人制作会保证关卡视觉风格的一致，从而保证可玩性最好。

但是一个设计人员负责一个关卡的方法不是惟一可选的方法，许多设计者采用了更为团队





的关卡设计方法。如果小组中某个成员特别擅长建模，但是不擅长控制游戏 AI，这时可以让另外一个成员帮忙设计这个组员所负责关卡的游戏操作性。也可能某个设计人员特别擅长灯光效果设置，同时其他人的强项是脚本控制。或者希望音响设计人员专门负责音效，因为他在这方面很熟练。当然，和分解任何同一个目标给多个人相同，如果把一个关卡的任务分给多个人，必须保证他们对关卡需要完成的目标有一致的看法。例如，结构设计人员可能设计一个峡谷，他认为是比较适合埋伏的地方，但是轮到负责游戏操控的设计人员工作时，他也许不会注意到这个峡谷的特殊之处，并且可能会把游戏中的 NPC 对手安排在不是太合适的地点。就像在其他的开发、美工、设计小组中一样，从事一个关卡制作的几个成员之间需要保持交流。

本文开头讲述过，由于游戏变得越来越复杂，需要把以前由一个人完成的工作改变成多人之间的协作。在游戏日渐复杂的同时，设计人员的专业化分工也越来越明显，多人从事同一个关卡的设计已经是大势所趋。在这种项目中，保持游戏的整体性就成为了一个很大的挑战，项目负责人和关卡设计负责人的角色越来越重要。然而，在人们专长于关卡设计的某个特定方面的同时，他们在专长领域内的经验越来越丰富。除此之外，如果某个组员负责游戏中所有关卡的 AI 设置和游戏操控，这会让整个游戏的游戏操控从总体上获得更好的一致性，这可能比一个设计人员负责一个关卡的游戏操控要好。如果管理得当，这些高度分工的专业化关卡设计人员会让游戏的关卡表现得更好。



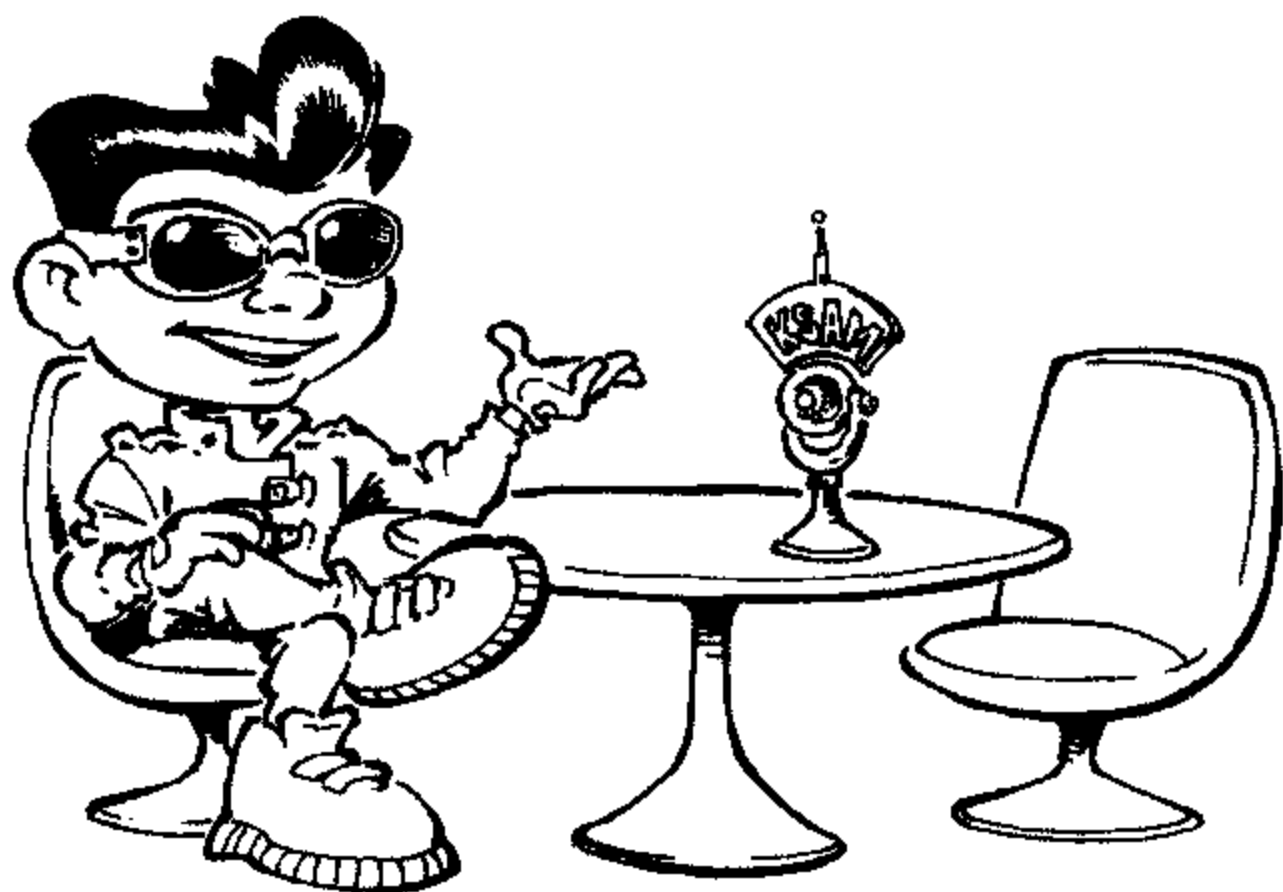
21.7 协作开发

在游戏日益复杂的同时，游戏设计需要的关卡设计人员数量也随之增加。过去，一个设计人员可以控制游戏设计的各个方面，现在，设计负责人必须信任关卡设计人员，相信他们对游戏的设计会做出贡献。虽然设计负责人可以用最大的努力亲自做设计，但最终她会把大部分游戏制作工作委派给小组中的成员。这是一把双刃剑，好处在于众人拾柴火焰高，集体的智慧会让游戏更稳定、更优秀；弊端在于众说纷纭、各执己见，多人参与一个项目会让原本清晰的艺术观点变得模糊。这就是现代商业游戏开发方法中存在的危险。



Chapter 22

第 22 章 Will Wright 访谈录



Will Wright 的游戏 SimCity 对游戏产业方面的影响难以估量。在 1989 年开始发布这个游戏的时候，它就显示出与其他互动电脑游戏的截然不同——其他的游戏要花好多年的时间才能找到开发商。现在，这个游戏的影响在每年出版的“建造”模拟游戏中随处可见。Sid Meier 曾经公开承认他的游戏 Civilization 最初的创作灵感之一就是来自 SimCity。Wright 的最新游戏 The Sims 已经发布，他完全放弃了以前的成就，重新开始奋斗，使游戏取得了成功。目前发布的大多数游戏仅仅是游戏发展改进过程中初级阶段的作品，而 Wright 的 The Sims 却代表着近年大家所看到的最有创意设计的游戏，一场真正意义上革命的产品。同 Wright 谈话本身就是一种体验，他会让你即刻明白为什么他能开发设计出如此卓越并富有创意的游戏。



问：你是怎样开始对游戏设计产生兴趣的？

答：大约是在 1980 年，我买了台苹果 II 型电脑，很快我就沉湎于此。像孩子一样，我花费大量时间建造模型，并且还购买了一些较早版本的游戏，如带有线框图形的 Flight Simulator 的最早版本等。用自己的游戏设计语言程序给游戏程序加上补丁，这样做会让你感到其乐无穷。而这个能让人在电脑里构筑一个属于自己的微小世界的想法引起我的极大兴趣。我把电脑视为



一种制作模型的工具。所强调的一点是，由于我沉湎于游戏，所以才决定试试亲手设计游戏。大概也就是在那个时候，Commodore 64 开始投放市场。我买了一台 Commodore 64 机器，觉得在新机器上开始设计游戏比较好，因为在我决定设计游戏的前几年里，有些人曾学过使用 Apple II 型电脑，而当时人们使用 Commodore 64 电脑只不过用于娱乐而已。Commodore 一问世，我就购买了一台，并立即进行钻研，以最快的速度学会设计游戏。这就是我第一次设计游戏时的情况。



问：你怎么想到要设计 Raid Over Bungeling Bay 游戏？

答：大家都知道，那时的游戏都是街机游戏。而我一直热衷于直升飞机，所以我想设计直升飞机游戏。当时我正在钻研 Commodore 电脑。对我来说，电脑操作技术要比设计游戏重要得多。我发现，Commodore 机器的功能很强，使用它可以重新定义字符集，使其看起来像个图形，然后可以在屏幕上平稳地滚动。这样就可以提供滚动较大位图图像的效果，而实际上我们所做的只是在屏幕上移动 ASCII 字符。当我发现这个特点时，觉得这种表现的确很不错，因为在 Apple 机上无法如此流畅地在屏幕上移动图像。所以我在设计游戏时利用了这个特点。



问：我了解到，该游戏在日本比在美国流行得更广泛，你怎样看这个问题？

答：我认为恰恰在那个时候盗版达到顶峰。我们在美国出售了 30 000 套这种游戏。像那样的游戏卖到这个数量是很正常的。可那时我所接触到的有 Commodore 电脑的人都玩过这种游戏。然而，同样的游戏在日本的 Nintendo 就卖出去 750 000 套。在日本用的是一种盒式存储器系统，没有盗版。



问：回顾游戏设计，你还依然那么自信吗？

答：是的。我以美好的记忆回顾此事。那是一种学习的经历。你必须要用 10% 的时间把游戏推向市场，那段时间确实很难度过。如果你不制订出那最后 10% 的时间计划，这个游戏就等于判了死刑。我从中学到了好多经验。想一想，那时候的编制程序不比现在的编制程序复杂。每个游戏都是由一个人去编写。游戏的计算机语言大约有 8000 行。你能完全控制存储器，也能完全控制计算机。它是一种很好的学习媒体。遗憾的是，当今学习编程的程序员还是在从完全不同的角度掌握这种媒体。



问：你的意思是他们正在使用高标准的编程语言吗？

答：噢，是的。我想，这并不糟糕。不过你仍然能看到像我这样的电脑高手。当我完成那个游戏的时候，计算机上仅有 8 个字节的存储空间。我感觉不好，不能使用那最后 8 个字节了。因为内存是至关重要的，所以当你用完内存的时候，就得利用一些窍门。有一些很好的应用窍门。





问：听说我读的 Bungeling Bay 关卡编辑工具是你设计 SimCity 的灵感？

答：实际描述一些带有城镇和小路之类的群岛是游戏的一个特性。通过编辑程序来勾画背景有如此广阔的领域，是我自己在设计游戏个性上的发展。就像画图程序一样，我能使电脑屏幕上下左右滚动。我发现，我对画图程序比对游戏本身还喜欢，以至于设计完那个游戏之后，我还热衷于画图程序。最终发展成我的 SimCity。



问：你愿意说说对 SimCity 产生作用的其他游戏实例吗？

答：如果有的话，那产生最大作用的应该被公认为系统动态学之父的 Jay Forester 的作品。Jay Forester 是最早用计算机设计模拟游戏的人之一。当开始构思 SimCity 的时候，我到图书馆去查阅资料。20 世纪 50 年代，Jay Forester 写了大量著作，有关于早期计算机的，还有关于早期模型的。他还是尝试制作模拟城市的第一人。他制作的游戏如同 20 个可变量：有人口，有出生数，还有出生率等等，都是非常简单的模型。

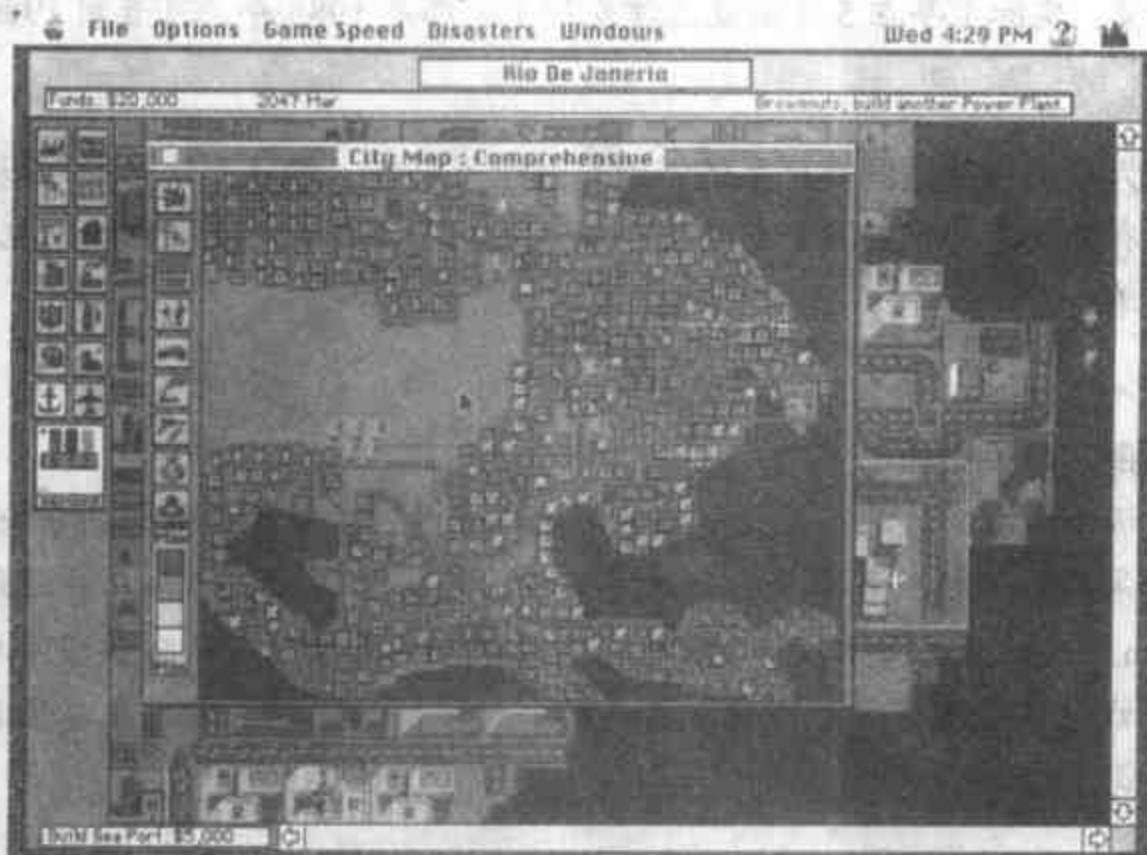


图 22.1 SimCity

系统动态学是观察某个系统的一种方法。基本上，系统动态学分为股分动态学和流率动态学。股分是指数量，如人口；流率是指比率，如死亡率、出生率、移民等等。用这两种特性似乎可以规范所有的事物。接下来就是如何启动系统动态学，如何走近模型。当我开始制作 SimCity 时，我发现了其中的要素，并开始自学制作模型技术。我也发现了关于细胞自动机较新的要素，SimCity 就是这两种方法的混合产物。因为他的方法一点不占用空间，而细胞自动机真正给你许多用于传输、网络流程、接近等等有趣的工具。因此污染从这里开始，蔓延到其他地方，直到慢慢地减弱。实际上，可以通过这些空间结构模拟传输波浪。在某些背景下 SimCity 很像一个大的三维细胞自动机，每层都具有像罪恶、污染、土地价格等背景特性。而层与层之间在三维上又相互作用，所以罪恶层和污染层可以冲击土地价格层。





问：你认为像这样一个学者气的技术为什么能让大家喜欢？

答：就这点来说，我原来不打算制作那些为大家娱乐而游戏的东西。就像刚才我自己做游戏时那样开心。同时我还在读有关都市动态学理论方面的东西。当我读到有关这方面的主题时，在我的电脑上有了那个小豚鼠城，这样使得这个主题更加有趣。因此我能读一种理论，然后力求以图像的形式把它表现出来，给它定型，为它编程，制成模型，看结果会是怎样。



问：从哪一点上你开始认为它可能成为人们喜爱的游戏？

答：大约 6 个月以后，我开始给它加一些图形。开始时，图形是完全抽象的。那时我开始思考，要知道，这可能是个非常有趣的游戏。实际上，我已经用 Broderbund 软件制作了我的第一个游戏。当时我把游戏演示给大家看，他们认为游戏非常酷。于是他们接受这个游戏，和我签订了合同。为使其达到尽善尽美，我又工作了大约一年的时间。他们一直认为游戏没有最后完成。他们总是问：“游戏什么时候完成？什么时候会有一个最后胜利或败北的结局？”当时发行时间很不正常，实际上早在 5 年前就应该发行了。那是 1985 年，可直到 1989 年才发行。



问：他们认为只适应其他同类产品的游戏是远远不够的，你认为这么说对吗？

答：他们没有见到怎样把产品销售出去。我只是把产品留在那里，他们也把产品留在那里，事情就是这样简单。



问：你当时是不是失去信心了？

答：我总是认为我所制作的不过是一件较酷的小东西，我真的没有想到它会成为市场主流。不过，我认为如果能把它投放到市场上，所费的精力还是值得的。后来，我遇到了我的最后合作伙伴 Jeff Braun。我把游戏演示给他看，他认为游戏确实非常酷，他确实被陶醉了。他认为，像那样的游戏也许会有较大的市场。就是那个时候，我们俩决定建立自己的公司，也就是当时的 Maxis 公司。



问：当时没有发布，是不是又过了几年才发布？

答：是的，搁了几年才发布。我们决定创建 Maxis 公司的时候，Macintosh 刚刚问世，Amiga 正在投入生产。我们决定为这些游戏重新编写程序。于是，我们聘用了几个编程人员，并且我还重新编写了模拟程序。以前都是流水作业。我们的其他编程人员在前端处理机上帮助处理 Mac 和 Amiga 上的图形。实际上，这些是发行的第一个版本。我们走了回头路，第一个版本大约发行一个月后，又发行了 Commodore 版本。



问：原始版本的 SimCity 没有鼠标点击界面吗？

答：不是的，事实上是有的。当我制作 Commodore 上的产品时，The Lisa 就已经问世。我用图标实施一个基于鼠标的系统。界面在 Commodore 上，仍旧有图标，是触摸式画图程序，看





上去和 MacPaint 一样。实际上，它确有相似的图形界面处理机，但是分辨率较低。

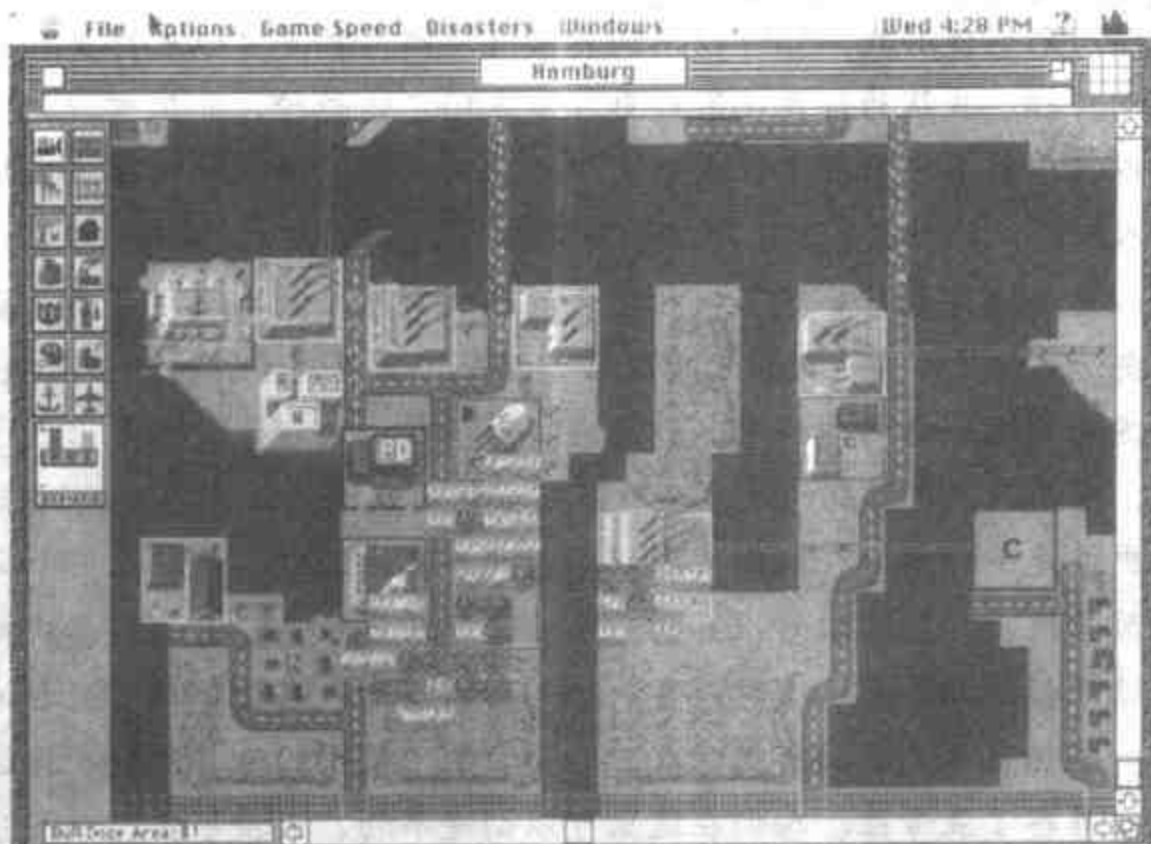


图 22.2 SimCity

问：与你的最初版本相比，设计上的变化是不是很大？

答：制作更精美了，加了更多的层，布局图上的分辨率更高了，但模型和相同附件仍保持原来的基本结构。比如，只有道路的、没有道路的以及高速公路的。原布局图的比例是 80:90，替代它的是 128:128。当然，图像的分辨率较低，大约是 4 像素代表一小片，替代了 16 像素的。模型的核心和模型的转向实际上变化不大。对于 SimCity 2000 或 3000，实际上变化也不大。

问：Maxis 公司最终是自己发行产品投放市场的吗？

答：这实际很有趣。自从我们在 Mac 和 Amiga 上把它重新制作以后，我们知道我们能够承担生产包装箱和其他一切费用了。不过，我们必须要有个经销商。实际上，我们回到了 Broderbund 公司，把我们的产品演示给他们看。当看到 Mac 和 Amiga 版本时，他们被打动了。后来，又过了几年的时间，市场上开始热卖有趣游戏。那个时候，他们提出要做我们的经销商，后来我们与 Broderbund 公司建立了隶属出版关系。当时，我们正在遭受财政危机。我们要支付包装箱及一切费用，而他们实际上没有什么风险。Broderbund 公司的员工非常友好，一点都不嫉妒。他们为 Maxis 公司的掘起帮了大忙。创建 Broderbund 公司的 Carlstone 是我生意上的楷模。他们确实是非常友好容易相处的人。

问：你是不是以“软件玩具”起家的？

答：我想是的。因为我在游戏开发者论坛上讲过这个，我决定把它作为我的发言题目。那就是“软件玩具：创造力和心领神会的交叉产品”。





问：你是怎样区分软件玩具和游戏的？

答：玩具可以用来创建游戏。你可以用玩具来玩游戏，并且可以玩得更加自由些。然而，玩具没有直接活动目标。我认为玩具比游戏更具有不固定的特性。我们可以用球来做游戏，比如篮球，我们可以把球前后投来投去，或者我们可以用这个球做试验，拍球出场是另一回事，所以我把玩具列入较广泛的种类。玩具也可以彼此组合。我可以把 Barbie 捆绑到我的 R.C. 汽车上，并驾驶它到处走。这样也就创建了一种组合玩具的新活动。游戏要是离开设置规则就是孤立的，我们人类要是离开了宇宙也就没有意义了。换一个角度思考，这是最近版本的一些理念，那就是我倾向考虑用更高一筹的方法来制作游戏。大多数游戏被认做是电影或电影形式。电影有开头，有结尾，有高潮，有特别的故事情节，更多的游戏就是以这种形式制作出来的。

我们的游戏更像是一种嗜好。人们可以以不同的方式走近它。就好像一个火车模型，有的人喜欢把它开进风景区，那里有惟妙惟肖的悬崖峭壁以及丘陵，还有的人喜欢把它开进中世纪的小山村，开进铁路枢纽站等等。有时，喜欢玩某种游戏的人在这个社区里会彼此对垒。在这个社区里也会有一些热衷于某方面游戏的人，他们会利用他们的专长教别人玩这种游戏。在这个社区里他们已经具备了亚专业。人们在这里可以创造事物、交换事物，以及分享观念。我注意考虑游戏社区比考虑电影模型多一点。更多的是分享经验，是一种文化潮流。就像有人问你：“喂，你昨天晚上看那个电影了吗？你觉得怎么样？”



问：不过，如同 SimCity 软件游戏一样，在任何时候只有一个人才能真正玩游戏。

答：我现在提及的社区比以往提及的要多，这就是在线社区。我们可以上网，可以和他人谈论生意策略，或者可以建造我的城市、我的家庭和我的故事，还可以赋予 The Sims 护肤能力。如果他们对此非常精通，他们可以在这个社区有一个身份：“噢，他创造了最佳护肤能力。”于是，随着游戏的发展，网上拥有了这么个整体社区，人们可以利用它创造事物、分享事物。



问：和 SimCity 原版比较，哪一种版本更合理？

答：回顾 SimCity 问世的时候，确实有几家像 CompuServe 的公司和后来的 AOL 在网上发布过零散的信息。大多数都是些聊天讨论之类的东西，根本没有人们讨论的焦点问题。它不涉及真正的网上社区。甚至在我们拥有我们自己的第一个网站之前，人们已经在 AOL 上建造他们的城市，并且进行交易了。数以百计城市之间的交易占了大部分。CompuServe 是大量城市集合出现的第一个地方。时隔不久游戏就问世了。



问：关于 SimCity 这个游戏，我所听到的最大抱怨就是，产品既不像游戏又没有任何目标，根本抓不住玩家的注意力，而且其他游戏开发者也是这样说。

答：我认为我的游戏吸引着不同兴趣的玩家。实际上，一些人玩游戏的目标是直接的。它促使人们决定目标。当你开始玩 SimCity 时，最有趣的事情，就是你必须决定“我想做什么？我是否要建一座最大的城市？在那座城市里是否居住着最幸福的居民、拥有最多的公园以及最低的犯罪率？”每次你都会在脑子里把它们理想化，“理想的都市对我意味着什么？”它需要更





主动的玩家。因为我们对你的目标不加以限制，那么在某种意义上说什么才能使你有更多再次玩游戏的可能性。我们只能说“让你的城市在十年内达到一万人口，否则你就算失败了。”让你总是以这种态度玩游戏。到达那里得实施一种战略，人们会描绘出战略，事情不过如此。留下更多不固定的结局，让人们能以不同的方法去玩它。这就是更像玩具的地方。

总之，模拟游戏给人们更广泛的游戏空间去开发。在 SimCity 游戏中，大概还没有不同的人创建的两个城市是一模一样的。然而，如果你注意到像 Zelda 那样的游戏，我确信会有数以万计保存的 Zelda 游戏是完全相同的。从数量上来看，可视其为系统的相位空间，或者描述成当前系统状态发生变化的多少。观察这个问题的另一个方法看是否允许玩家开发创造。你的游戏什么地方比我们的游戏更有独到之处。从某种意义上说，还暗示着创造水平的高低。在某种情况下，也可以理解为以不同的方法解决一个特定的问题。所以，如果我们从一个同样确切的交通繁杂的城市开始，我们会有大量成功解决问题的办法。在许多游戏中，都有一个锁着的门，除非你找到开门的钥匙，不然你是打不开门的。

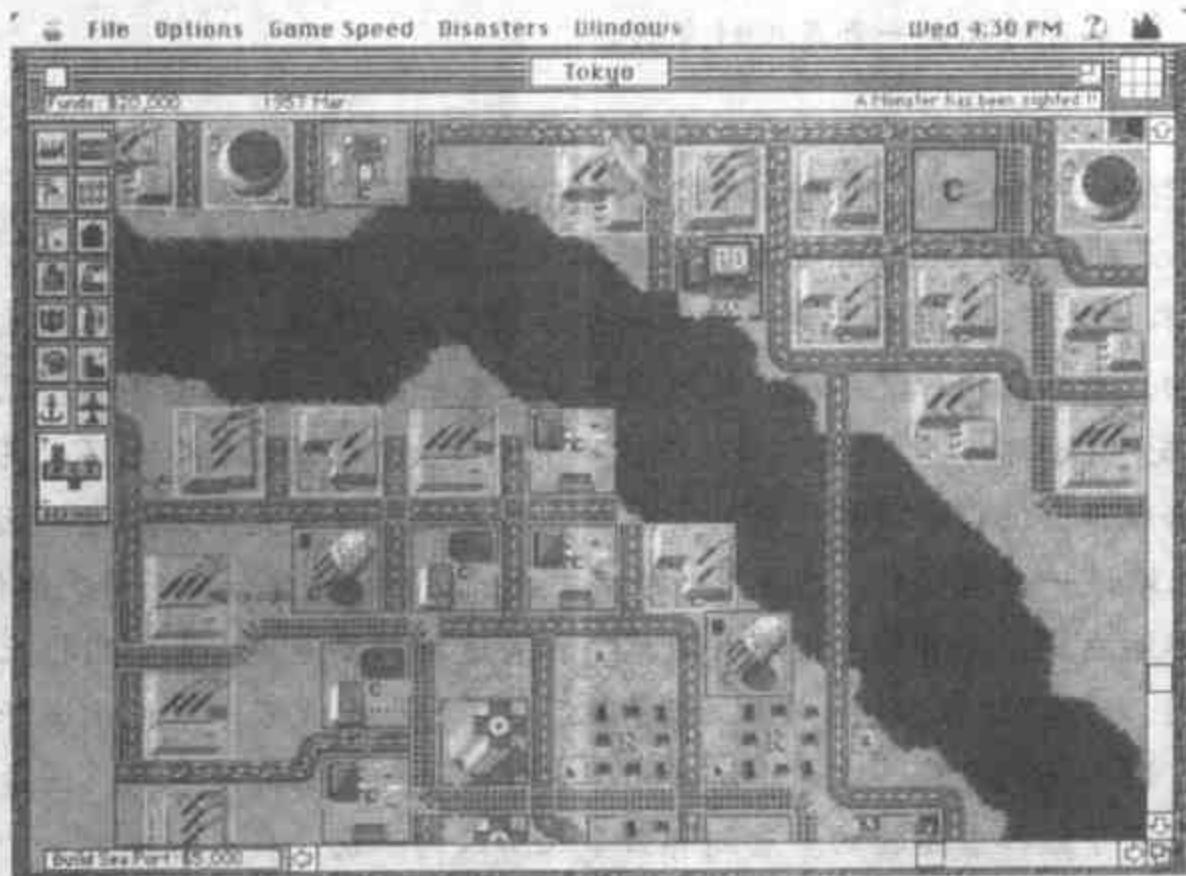


图 22.3 SimCity



问：所以你为玩家提供更多的可变化性，对吗？

答：不仅存在有多种变化，而且还因为每一个玩家只能使用惟一的入口，所以他们变得更富有创造性。玩家在游戏中实现的创造力越多，就越迷恋游戏。特别地，你在 The Sims 游戏中见到的就是这样。如果他们花费心血建立起一个家庭，又在一起生活了几个月的话，确实会对这些人物着迷。因为他们投入了那么多时光来创建它们。那些景中的人物，不仅反映的是他们自己，而且还反映他们对流行游戏的理解。SimCity 也一样。你可以在任何时候看一下某人在 SimCity 中创建的城市，城市的设计反映他们对模型的理解。出自对它们的理解，才可能有建造城市道路网络最好的方法。





问：一旦他们对游戏能更好地理解……

答：事情确实是在不断变化。你可以回到一座古老的城市，并且说：“噢，对了，我想象中的道路果然不错，我没学会以前，道路不是这样的。”所以从某种意义上说，反映的是你的游戏智力模型。



问：如果你是第二次玩 Zelda 游戏的话……

答：你的智力模型演变不会太大。你一玩完 SimCity 这个游戏，就会惊奇地发现，你的模型基础结构并不都是完全不同的。



问：我对 SimCity 游戏中的灾难特征有些好奇。玩家花费大量的时间建立营造物，然后又用潮水或火来毁灭它们，这是不是有些不正常？

答：是的，我本身对此也一直是挺好奇的。



问：你一定是预先考虑的，因为你在游戏一开始就已经把它设置进去了。

答：实际不是的。在 Commodore 的最初版本中就没有，是后来才加上去的。当我首次演示 Commodore 版本的时候，能有的只是一台推土机，用它来删除错误。如果你偶然建了一条道路或者一个营造物，而后发现位置不对，你可以用推土机铲掉它。我总是发现，玩家在玩游戏的前 5 分钟就会发现推土机，他们会意外地毁掉一个营造物。然后他们会开怀大笑。接着他们会用推土机去攻击城市。他们会毁掉所有的营造物，然后他们将狂笑不止。这一点使我困惑。因为这就好比某人路过一个蚂蚁窝，然后用一根棍去捅，观察会发生什么一样。他们要用 10 分钟退出该系统，然后发现，坚硬部分没有被毁掉，但营造物却要重建。人们用推土机毁掉城市花费了大量的时间，后来他们发现，“哇，能量耗尽了。哇，起火了。”他们会开始重建，这确实使他们上瘾。因为他们发现，在这个游戏中，毁掉营造物是如此的容易。回过头来看，当时的所有游戏都是关于毁掉营造物的。看到那么多人所做的一切，我最后决定，“我真的该让他们走出这种游戏，我要给游戏补加一些灾难。”就是这些给了我补加灾难菜单的想法。



问：你所加的灾难是随机发生的吗？

答：是的，我加进灾难菜单之后，似乎很明显灾难应该随时发生，但是起初并没那样。



问：SimEarth 好像是 SimCity 逻辑上的延伸。你又是如何构思这个游戏的呢？

答：迫使我介入的那些题目会更让我感兴趣。我对某些理论很感兴趣，有著名的 Gaia 假设，还有很多时候看的非直观论的环境文章。我想，制作一个全球生态系统模型会是很有趣的。我从 SimEarth 学到很多。实际上，我为 SimEarth 的模拟感到自豪，而在游戏设计上却感到非常失望。



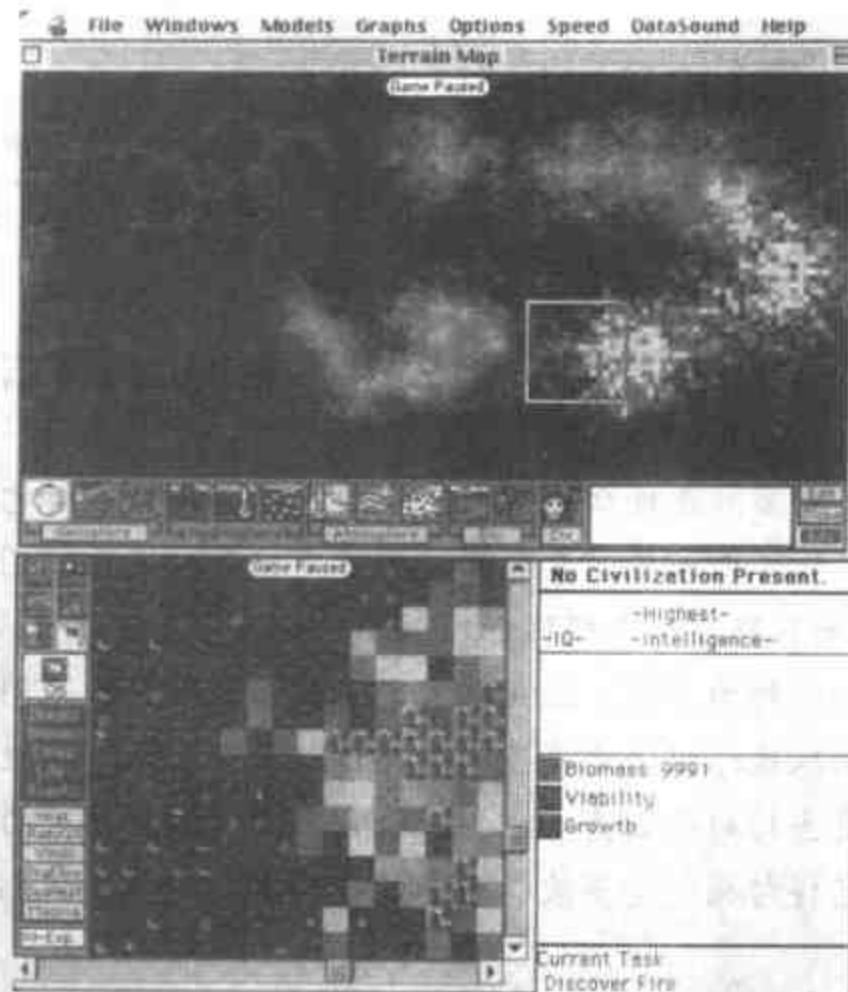


图 22.4 SimEarth



问：你如何解释？

答：它不是非常讨人喜爱的游戏。实际上，它是非常好的模型。我对流行的气候模型做了大量的研究，未见哪个人把整个石圈、水圈和大气层像那样放在一起制作出一个完整的模型。我们在那种模型方面的真正影响越来越大，真正是大大胜过了其他同类模型，甚至连像 NCAR (National Center for Atmospheric Research) 那样精心制作的模型都不能像我们的模型那样引起关注。但是随着游戏的进展不断深入，我们开始感觉到，要是你粗略地看一下我们所有的 Sim 游戏，就会把它们分为两种类型，一种是经济类，另一种是生物类。总之，经济类的制作较好些。



问：你把它们归为哪个类别？

答：虽然 SimCity、SimTower、SimCity 2000、The Sims 和 SimFarm 两种类型都兼有一点，但 SimAnt、SimEarth 和 SimLife 大概要算是生物类的。



问：为什么你认为经济类的游戏更成功呢？

答：我认为，对于系统控制，需要做大量的工作。生物类的过于软，需要做些较细的工作，还要重新演示，重新编写。我并不十分清楚你对它做过哪些，因为它会围绕你不断深化。为此，在经济类的游戏上，你具有较好的信誉。当发生某些错误时，你会说，“噢，那是因为我忘记了，我该买一个。”对于经济类模型，我认为人们会通过失败找出原因，把声誉更容易地归结到失败上。加上你有钱和你要赚钱的意识，你在这方面的消费是很自然的。每当你进入复杂的事物中，如变化的事物群、食物网，如此之类的事物，人们没有这方面的本能。





问：没有比玩游戏时不知道为什么失败更糟糕的……

答：的确如此。所以在 SimEarth 游戏中，人们玩着玩着会突然出现星球被冻结的现象，他们找不到冻结的原因。我作为模型设计师，也不能告诉他们为什么会是这样。



问：我喜欢 SimEarth 游戏的原因之一就是它可以演奏音调，这种音调会把游戏中星体的有关信息传递给你。

答：我一直想在这上面多做点什么，但做得还是不够全面。在听觉数据方面本来就有一些有趣的工作。替代视觉，你可以采集复杂数据使图像有伴音，因为我们在识别领域对某种声带研究得还不够。Santa Fe 研究院在这个领域已经做了一些工作。值得一提的是，他们采集地震仪上面的数据，把地震等构画转为声波，运用相同的声波形式勾画出不同的频率。他们还用此方法做了地下核试验。如果从地球仪上看声波，两个试验完全相同，要说出核试验与地震试验有什么区别确实很难。但是当你勾画出声音，核试验会有明确的细弱无力的声音，你立即会识别出来。有趣的是，不管怎样勾画视觉声波，都不能在两者之间找出识别的方法。而一旦勾画出声音，区别就会明显。

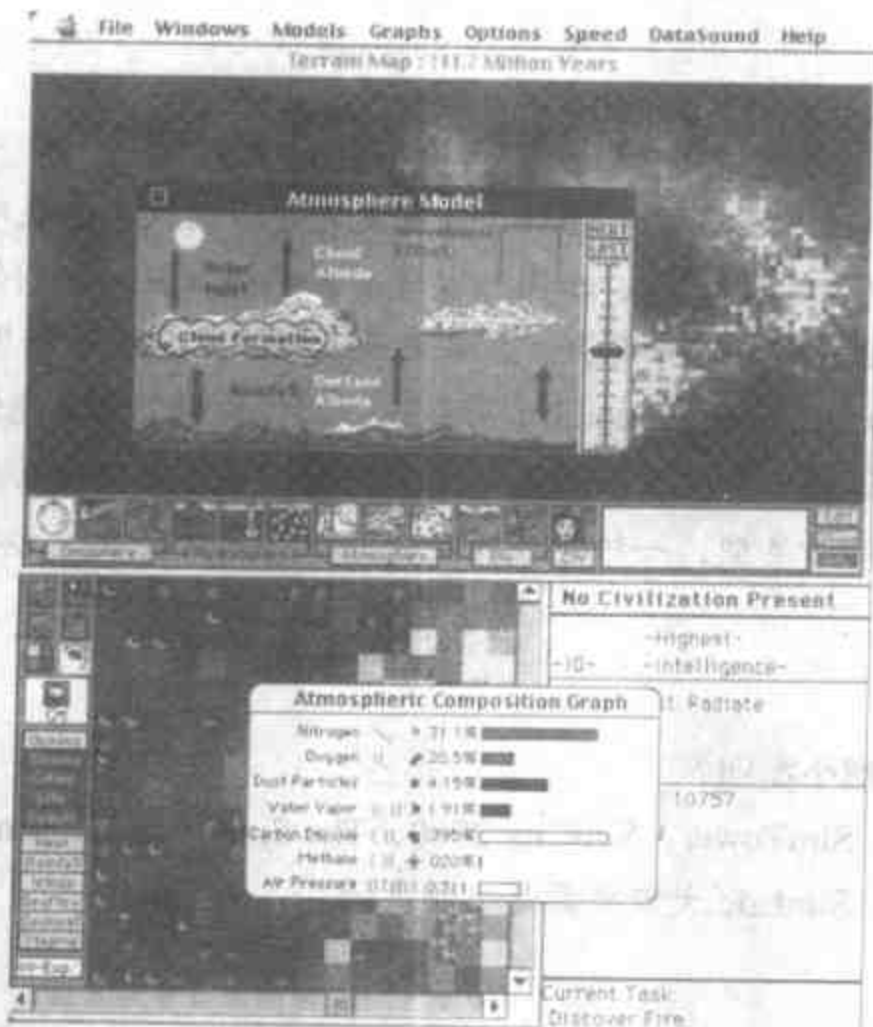


图 22.5 SimEarth



问：你认为通过声音传递玩家能更好地了解星体的有关情况吗？

答：是的，那只是一种有点愚蠢的尝试。就某方面来说，我喜欢坐下来把它做好。我认为做得非常漂亮的事是，不断地把大气层的色调勾画出来，从北极到南极。如果你以低音量离开



背景时，它会非常有用，因为你能说出的变化比你能看到的反映在视觉图形上的变化要快。就像入口的警钟，我认为做的非常好。因为你实际上是在潜意识地做这件事。短时间以后，你开始习惯这种低音调，然后，突然音调变化，到达你头脑最重要的位置。当你做其他事的时候，它也在进行，所以你不必坐下来整个时间玩游戏。我总以为这样很酷。

问：与众多的其他游戏题目相比较，SimEarth 是非常让人感兴趣的，为什么你选择它呢？

答：在游戏中，我不想做得太过于人性化。游戏的规则之一是行星上的人类只是偶尔进化为赋予智力的人类。它可能不过是三脊椎动物或者是其他什么动物。所以我尽量避免以人类为中心的游戏。而实际上游戏的焦点就在行星上。很久以前，当我工作时，我力求把自己放进意识设置中。我的意思是，当你从事某事时，你应该入迷，进入角色。直到今天我仍是大陆上漂流的什么东西，就像人们听到毒品一样令人讨厌。对于这个题目，我很难用语言表达。SimAnt 实际上也是一样。还有，我认为蚂蚁是我们周围最酷的东西，我认为通过这个游戏我还没有清楚地把我的意思传达出来。

问：SimAnt 似乎比 SimEarth 和 SimCity 古怪得多。

答：对待蚂蚁，太严肃的确有点难。SimAnt 还真让我吃惊。这是我制作游戏以来的第一次，我完全被这不寻常的数量统计所吸引，它超出了我的想象。SimAnt 游戏轰动一时，它的对象是 10~13 岁的孩子。父母亲们会购买它，孩子们会玩它，并且还会喜欢上它。直到今天仍有许多人告诉我：“我喜欢 SimAnt，它是最喜爱的游戏。”我对此非常满意。正像我所希望的那样，欣赏这个游戏的更多老年人把令人惊异有趣的蚂蚁作为散布信息的楷模。从某种意义上说，我力求用古怪的方法，演示蚂蚁作为信息处理系统是何等内在有趣。但实际上，我对刚刚喜欢上蚂蚁游戏的 12 岁孩子结束了呼吁。

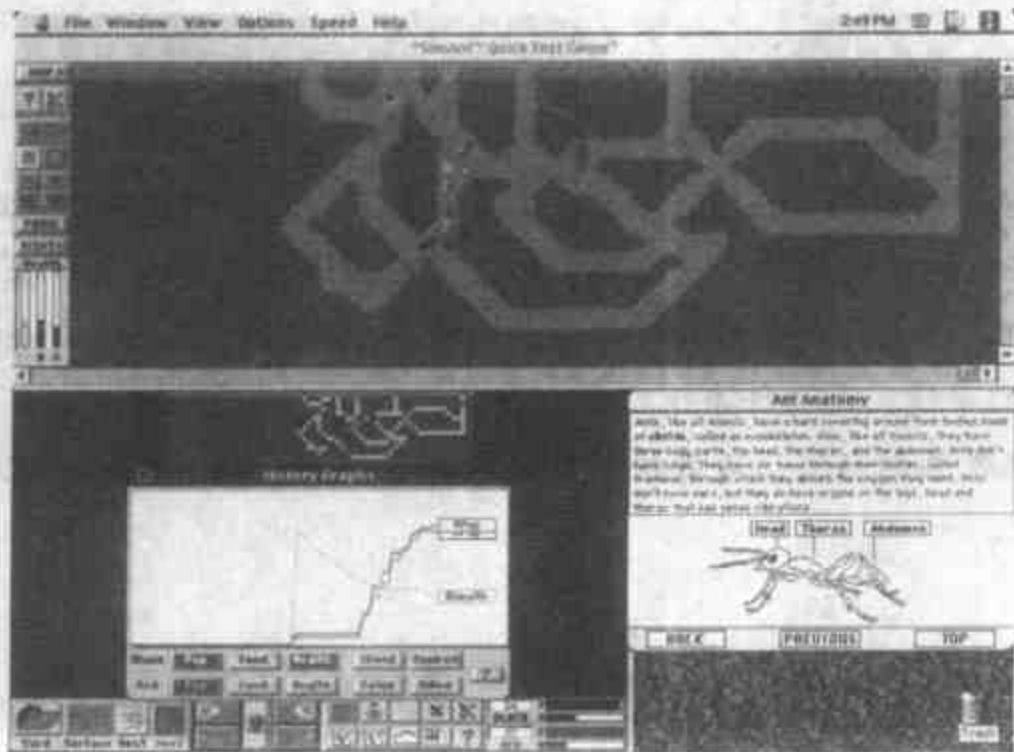


图 22.6 SimAnt





问：蚂蚁模拟器作为游戏来说前提似乎是非常奇特的，你为什么选择制作它呢？

答：我涉足其中是因为我喜爱蚂蚁。SimAnt 对我来说似乎是显而易见的。我一直疑惑的是为什么没有人制作一个计算机化的蚂蚁农场。我已经期望多年，可还是没有人做。时间似乎是最好的证明。我的大多数游戏是受我所读的书的影响。SimEarth 是受 James Lovelock 和 Gaia 假设的启发产生灵感的。SimAnt 明显受 Edward Wilson 作品的启发产生灵感的。Edward Wilson 是喜欢蚁类研究的那种人。他写过许多书。SimAnt 发布那年，他写了一本叫“The Ant”的书，获 Pulitzer 奖。这本书让人感到作者智谋惊人。在制作 SimAnt 营造物模型时，我们大量采用了他书中的内容。实际上，没有他的作品，我们不可能设计出我们的蚂蚁模型，就像没有 James Lovelock 我们不可能设计出 SimEarth 一样。



问：制作 SimAnt 这样独特奇妙的游戏，你遇到过阻力吗？

答：没有，一点没有。我认为在制作 SimEarth 时遇到过阻力，因为大家都期待 SimCity 2 问世，而实际上我还真不想制作 SimCity2。我想做些不同的东西。



问：与 SimCity 和 SimEarth 相比，SimAnt 更像游戏。

答：我认为 SimAnt 要比 SimEarth 的反映强烈得多。当 SimEarth 问世的时候，我感觉末日来临，就像坐在 747 飞机座舱里，飞机直往下俯冲，就像大多数玩家感觉的那样。所以我想让 SimAnt 逆道而行：没有恐惧，只有轻松快乐，能够真正清楚什么地方出现了什么错误。既然如此，我就不会说它有什么成功的了。我想做的一件事情就是要用 SimAnt 表现出一种思想，让人们轻松地进入游戏，可人们会感觉它越来越严肃。这就是我们把有关蚂蚁这部小型百科全书放到在线数据库的原因所在。这些思想足以使人们对它感兴趣。正是通过这个游戏，人们将实际开始阅读这部小型百科全书，其中许多内容涉及游戏操控。因此，你能够在学到游戏新战略的同时获得有关蚂蚁方面较酷的知识。



问：这个游戏让我想到一个非常不可思议的战争游戏。

答：它像是 RTS 那种游戏。在 SimAnt 里，我们做了一些古怪的东西。对于 SimAnt，在某种意识上说是很有经验的。那里边有一些稀奇古怪的东西，如同有一个神秘的按钮。界面上有一个带大问号的按钮，它就是那个神秘按钮。每次按下这个按钮，就会有奇异的事情发生。一共会发生 30 件不同的事情，每一件事情都是离奇的。比如，你的蚂蚁都死了。或者你的蚂蚁数量成倍增加，或者狂风暴雨来临，或者换了个画面。当你点击那个按钮，完全非线性的以及随机性的事情就会发生。



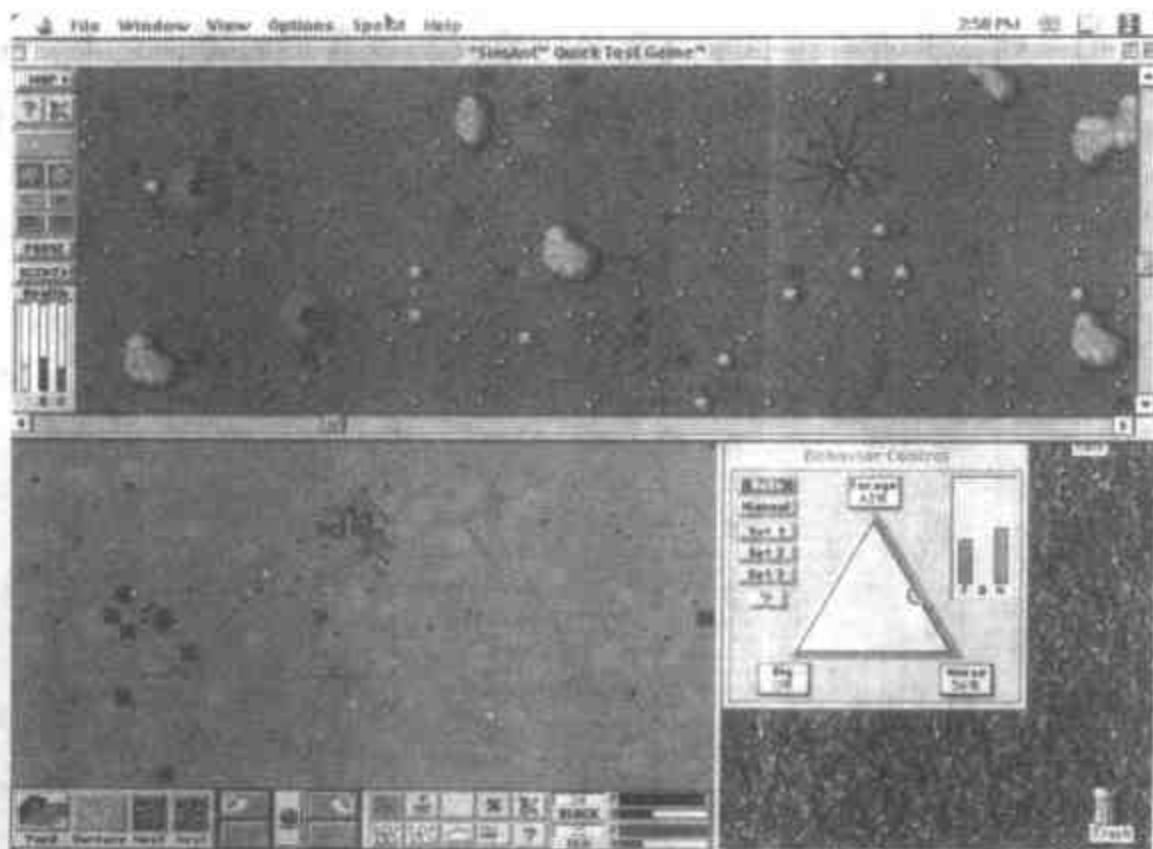


图 22.7 SimAnt

问：像 SimCity 游戏中的那种灾难导致了一个极其……

答：它几乎是一个超级灾难。突发事件会删除你的游戏，或者成倍地增加敌方的数量。像 SimCity 中的灾难一样，大多数人要做的就是玩、玩、玩，一玩就是数小时。一旦他们想停止，退出之前他们会将城市烧掉，因为城里有魔鬼。而在 SimAnt 里，人们会玩一会儿游戏然后退出，在退出之前，他们要点击一下神秘按钮，看看今天会发生什么。

问：你接下来的设计是 SimCity 2000，它是怎样问世的？

答：事实上，在我制作它之前，我已经花费了大约 6 个月的时间准备给我的 The Sims 游戏第一次赋予人形。我制作了一个小原型，并且加了密码。那时，Fred Haslam 在制作 SimCity 2000。他和我最终完成 SimCity 2000，还和我一起制作了 SimEarth。SimCity 2000 运转的不如人们想象的那样快，人们不喜欢它的图形和其他所有要素，所以我做的也就有点拖拉。这个时候，我们的公司确实依靠 SimCity 2000，它当时是个畅销品。所以我和 Fred 基本上放下所有其他工作而专注于 SimCity 2000。事实上，我获取了为 The Sims 编写的密码框架，把它用到 SimCity 2000 上以此结束。今天，回过头看一下为 SimCity 2000 编写的源码，绘画程序会显示出 DrawHouse 和 DrawYard，因为它是 The Sims 的密码框架。那时我投入其中，Fred 和我，基本上是从起点开始的。Fred 和我在一起工作非常愉快。因为游戏较复杂，我们几乎是用记录时间的方式来制作它。我们大约在一起呆了 12 个月。

问：对已经运转良好的 SimCity 原版的改进和完善是当时的想法吗？

答：粗略地讲是这样的。也就是在那个时候，数以百计的爱好者来信说：“噢，你应该再出个新的 SimCity，应该加上这个，加上那个等等。”我读了所有的来信。有几点建议是具有普





遍意义的，于是我们采纳了具有普遍性和突出性的建议加进我们的设计之中，如：海拔高度、山脉、水利系统、更多的道路类型以及事物的类型等。此外，所有这些，我希望我能够在 SimCity 中做到，现在的计算机运转速度快，图形较好，我想我们能做到。



图 22.8 SimCity 2000



问：与 SimAnt 相比，SimCity 2000 似乎少了许多稀奇古怪的东西。是不是因为你的参与？

答：我认为以它自己的方式够稀奇古怪了。它具有 SimCity 中所期望的稀奇古怪，又加上了许多在原版 SimCity 中没有的东西。我们在 SimCity 2000 中藏匿了许多东西，人们有可能很长时间发现不了它，这对延长游戏的寿命大有帮助。游戏中有 Loch Ness Monster。人们在玩游戏的过程中，每 2 个月或者 3 个月才会出现那么一次怪物，而且只出现 4 秒钟。关于这个有许多传闻。安装游戏 2 个月后，人们开始讲述他们在水中见到了这个怪物。因为它出现的机会实在是太少了，所以很多人听了都不相信。安装游戏之后差不多一年里，有的人确实想尽办法把那个怪物镜头从屏幕上抢拍下来。再后来，你有了上尉英雄。只有在某些离奇情况下，你才可以得到这位超级英雄。他可以飞来飞去为你解除灾难。我在游戏中藏匿了许多如毒品类的东西。原版 SimCity 确实没有这样深度的关卡。



问：因为你刚刚在做续集，你感到有压力吗？

答：没有。那时我更注意项目管理方式。因为我们基本上是在制作续集，所以工作起来比较简单。设计会是什么样的我脑袋里有非常清晰的构思。进一步设计会是很优良的，而且演示也是不错的。它是一个密封非常好的密码部件。原版 SimCity 2000 在 Mac 上运行占用 1.3 兆。对于那样的游戏，在那么小的存储器中运转确实是非常严格的。





问: SimCopter 是你的下一个设计吗?

答: 那种想法来得相当晚。实际上, 自从我着手制作 The Sims 背景的时候, 就在制作 SimCopter。因此, 就在那时我有了一个程序设计器, 用它来设计 The Sims。事实上, 在 SimCopter

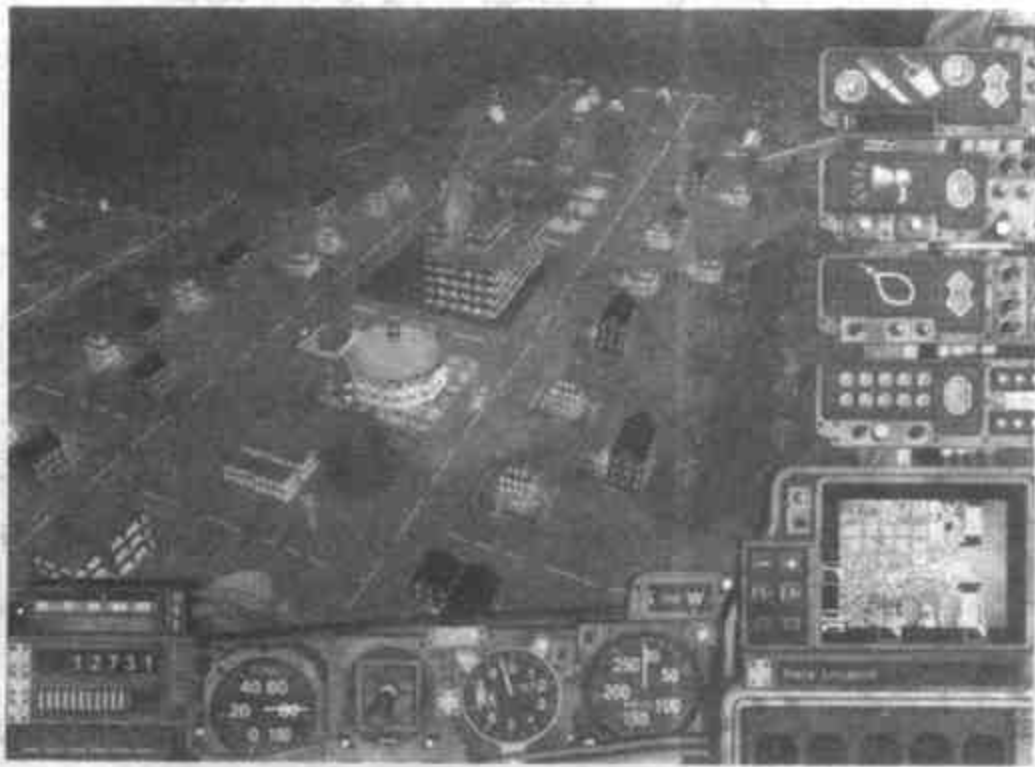


图 22.9 SimCopter

以在认识曲线上, 我们比其他人晚几年的时间。SimCopter 存在的最大问题, 我想不是游戏设计问题, 而是图形问题。它发布的时候确实用的是非标准语言。

中, 人们到处走路的举止, 正是使用最初期的 Edith 形式, Edith 形式是我们开发 The Sims 时使用的编程语言。Maxis 公司的许多人决定, 让我尝试在 SimCity 里制作 3D 游戏的一些东西。因此那就是 SimCopter 的原始前提。他们问我: “你能设计一个在 SimCity 中用 3D 制作的游戏吗? 不管它是什么, 可以到处行驶到处飞的游戏。”所以 SimCopter 是我发起的设计。它是我制作的第一个 3D 游戏。事实上, 也是我们团队多数人制作的第一个 3D 游戏。所



问: 你喜欢它的转变方式吗? 既然你对制作 The Sims 那么有兴趣, 你是不是非常关心它?

答: 关心。我把精力都放在制作 SimCopter 上。我们团队人员不足, 基本上是 4 个人制作它。在那个时候, 制作 3D 游戏人手根本不够。所以我感觉在产品设计制作上我们的智慧真的是被逼出来的, 加上我们制定了必须执行的严格计划表。因多种理由, 我们不能错过圣诞节, 这意味着我们确实不能把目标定得太高。我们还有 6~8 个月时间来完成它, 图形上我认为非常非常好。游戏操控和调音我非常满意。它已经可以运转几个新加的任务。花上数小时在 SimCity 上建立一个城市的确是一件不错的事情。然后完全是在 3D 中突然发生的事情, 人群到处乱跑, 汽车到处乱开。那才是真正可怕的特性。游戏制作好了。



问: 现在, 你没有参与 SimCity 3000 的制作。建立另一个城市模拟器的整个构想还存在吗?

答: 是的, 你说到了点子上。在 Maxis 公司流传一个笑话, 那就是, 不管什么时候 SimCity 团队征求我的建议, 我都会同意的。他们最终还是认输了。你知道, 他们运行 SimCity 3000 的那天是我一生中最幸福的日子。他们已经证实了, Maxis 公司有我们的团队, 这个团队知道没有我的参与时该如何制作 SimCity。以前, SimCity 2000 问世的时候, 就我一人, 没有人轮班制作。我只得自己制作, 或者不去制作。而现在, 我们有专门制作 SimCity 的专家工作间, 以及真正伟大、有天赋的团队工作人员。就我个人观点来说, 特权是一个有利助手。





问：你对未参与其中满意吗？

答：那是一个打折扣的报道。制作一个续集对我来说已经是够累的了。一旦投入 SimCity 2000，我过得非常愉快。有许多我喜爱的游戏我还没有开始制作，相反的，我对一些我的老游戏却重新进行了制作。或许我喜爱设计的游戏是学习研究新的主题，我会全身心地投入其中。我花大量的时间阅读有关都市动态学和城市规划的书籍。我仍然喜爱这个主题，但在这个领域的开发我是无能为力了。有许多其他的主题，我现在正在学习，并将投入其中。



问：我有一个关于 SimCity 3000 的问题。我最初见到的 SimCity 3000 原型是完全 3D 的。可安装运行时，却完全回到了传统的等距离视点。为什么变化会这么大呢？

答：这里面原因很多。要想做出决定可不那么容易。回想起来，我相信那是正确的决定。有一部分决定是关于用户界面的。玩 SimCity 游戏的人是个很广泛的群体，大部分人只是偶尔玩玩。他们移动、控制 3D 摄像机确实要经过一段时间。你把编制一个系统的构思放在首位，然后给他们一个 3D 摄像机，操作如以前一样非常简单，就是像 Lego（乐高拼装玩具）一样的东西，将它转换到 AutoCAD 上。“我看见的是什么？噢，知道了，我见到的是 2 英寸以外的建筑。”它成为一种经历。这只是其中一部分。另一部分是技术。对你所营造的，如果没有与之相配的真正严格的限制，我们就不会有合适的帧频，不会有等距离视点的关卡。现在我们正在向可行性目标进军。但是你涉及的是实际的存储器限制、质地限制和多边形限制。那时我们工作在 SimCity 3000 上，没有按要求的那样把足够的人手放在 3D 硬件上。所以我们不得不有一个可接受的软件解决方案。有许多理由，不过我说过只有两个是主要的，那就是性能和用户界面。



图 22.10 SimCity 3000



问：所以，你一完成 SimAnt 就着手 The Sims 制作。

答：是的，那是很长时间以前的事了。我同时着手设计几个项目，可中途都取消了。





问：那样做有意思吗？

答：有意思。我设计了项目 Z。同时我还设计项目 X、Y 和 Z。X 项目是我们称呼为 The Sims 的游戏，是制作时间最长的项目。Y 项目是 SimCopter。至于项目 Z，我想设计一个 Hindenburg 模拟。我确实去研究了它，并真的喜欢它。那的确是个稀奇古怪的构思。如果你可以设想它的话，那么它就是 Myst 和飞行模拟器的结合体。虚拟的 Hindenburg 被非常细心地绘制，那么美丽，那么细致地被刻画着，不管哪个犄角旮旯，它都可以穿行，它还可以进行探险。它功能配备齐全，无论扭动哪个阀门，都会产生逼真的效果；无论轻击哪个开关，都会发生与轻击真开关一样的事情。猛然间你会发现，你自己就在 Hindenburg 上面，越过太平洋，直奔 Lakehurst。你是这条魔船上的惟一旅行者。历史会重演，如果做的不正确，当你到这 Lakehurst 时就会船毁人亡。所以它是一个神秘游戏。我们正在制作 10 种或 20 种最佳推测，推测 Hindenburg 为什么会爆炸，现实中这种情况很少。你每玩一次新游戏，它都会随机选择其中一种爆炸原因，所以你每次玩游戏都不会是同一个爆炸原因。你必须防止爆炸，做完全不同的设置。其实，你也可以爬上驾驶室操纵驾驶。然后你就可以飞往不同的区域。你必须从零开始学习怎样驾驶齐伯林飞艇，这对一个人来说相当不易。



问：与你的其他游戏相比，它的确十分不同。

答：是的。大家知道，我的大部分游戏设计都被砍掉了。为什么要放弃制作它们呢？理由似乎真的很简单。在 Hindenburg 尾部有个曲十字装饰。许多人看到这个曲十字装饰，就把 Hindenburg 和德国纳粹党联系在一起。于是我们把它取了下来。遗憾的是，负责设计和制造 Hindenburg 的人是个纳粹党的极端反对者，他实际上为制造曲十字装饰与某位助手签了协议。所以纳粹党为最终建造付了代价。不管怎么说，这是我设计游戏失败的一个例子。



问：综观 The Sims 的整个发展过程，它同样保持有趣的风格吗？

答：它明显地经历了一个调整的变化，从建筑风格到与人相关的变化，但这不是主要的。事实上，我覆盖过一盘录像带，正是在我们完成 The Sims 之前，是哪一盘我忘记了。那是一盘 1993 年关于我们最初焦点组的录像带。在那盘焦点组录像带中，会议主席讲述了我编制 The Sims 的观点，明显地与我们的最终产品相近。



问：焦点组喜欢那个想法吗？

答：不喜欢。实际上，那大概是我遇到的最消极的焦点组了。The Sims 实际上是相当引人注目的，但焦点组的人普遍讨厌它。



问：那就是最初你不让员工参与设计的理由吗？

答：是的，不过那只是其中一部分理由。他们确实也帮不了什么忙。把焦点组放在首位不是我的主意。我们的市场销售人员说：“嘿，让我们组成一个焦点组，来确认我们的事情。”





当然，当焦点组的每个人都说：“我没有购买它的理由”时，对我推销该游戏存在着更多的困难。



问：你最后如何抓住机遇制作它的？

答：至少我使人们确信应该给我配备一个编程人员来制作背景。那是一个叫 Jamie Doornbos 的小伙子，他是最后的领头编程员。一个来自斯坦福的年轻小伙子，成绩优异的理科学生。就是他和我们一起开发背景方面行为模型的。我们尝试如何能够模拟一个不固定系统，在那里行为人可以伸展。行为还具有游戏要求我们制作的智力关卡。行为基本上能够应付所有的家庭生活，我们会合理地模拟它。作为一个小型研究项目，我和 Jamie 大概在行为模型上花费了一年半的时间。我们把它制作出来，看上去确实很好。那时，我们开始组建团队。我必须为我所得到的人而战斗、反冲、奋斗。



问：SimCity 制作成功以后，令人吃惊的是没有人相信你。

答：事实上确实是很滑稽可笑的，因为就在最近我正准备开始制作一些开发周期较长的游戏。就拿我刚做的一个来说，我开始把我的构思告诉人们，大家都说：“太棒了，太棒了，把它做出来，你是个编程员。”从某种意义上说真是令人失望。如果大家都说：“没什么意思，做出来有什么用”那样的话，你会反驳他们；如果大家不是众口一词地说“好极了！”我会放弃争斗。



问：你制作 The Sims 的最初灵感从何而来？

答：我想，我制作 The Sims 的最初灵感来自自由 Berkeley 建筑学教授 Christopher Alexander 写的《模式语言》一书。这本书非常有趣，当时在建筑界引起争论。这本书几乎就是风水的西方版本。书中有 256 条设计规则，每条规则都着眼于人类行为的某个方面，你可引用设计规则为自己所用。前面讲的规则是关于城市应该建在乡村边上。过了 10 个或者 15 个规则，它开始讲述城市设计、街道设计以及市内循环系统的设计。接着是较高级的规则，是关于街区应如何设计、学校应该设在哪里、娱乐中心应该设在哪里等等。再后来的规划

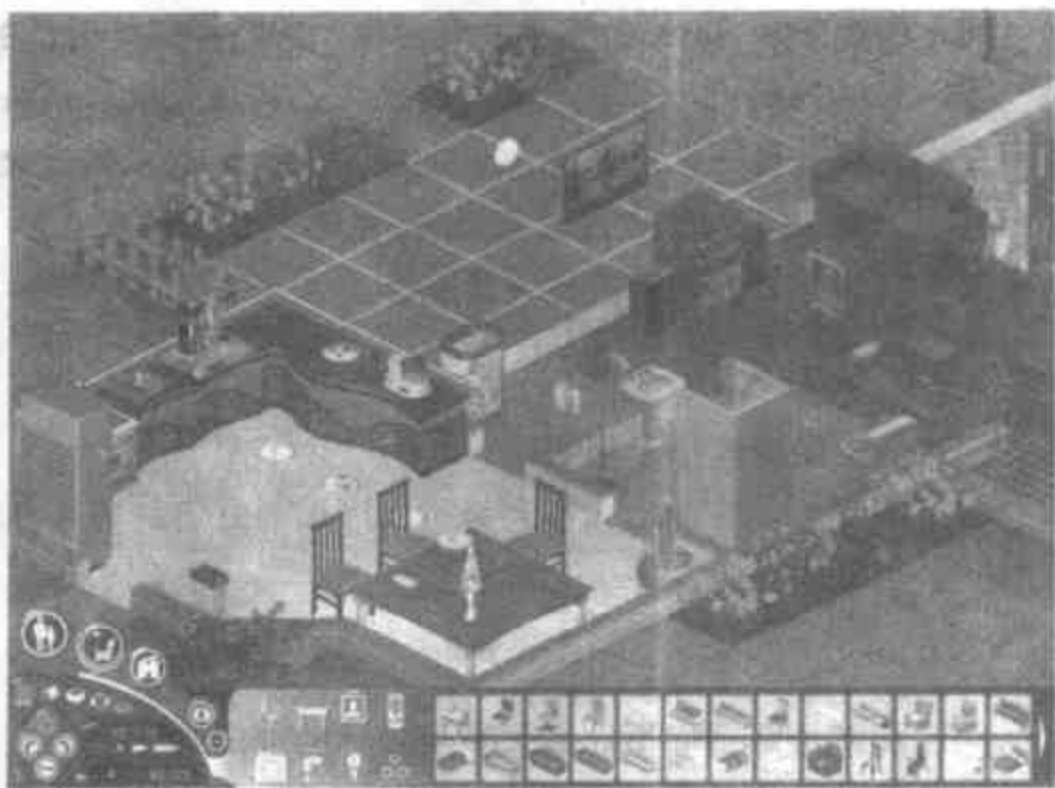


图 22.11 The Sims





离我们就更近了，应该在院子里怎样建自己的房子，以及在房子内如何设置自己和共用的区域等等。最后是最高级规划，那就是如何在窗台上摆放花盆，如何摆放公园长椅等等。虽然所有规则按不同级别分类，但它们都是以人类行为的方方面面为基础的。它们试着推断着。而实际上，我们喜欢有私人空间，我们在家内的多数活动是私人活动，此外才是社会活动。所以房屋的设计应该反映出这一点。在室内应该有非常明显的私人区域和更明显的共用区域。所以，这些就是他对人类行为方面的认识，从中推断出设计规则。然后，基本上他能给出设计基本原理的建议。大多数人对他所给出的独特的基本原理都会有不同看法，不过我总是认为他的这种尝试是非常了不起的。



问：所以你想效仿他的规则用于你的模拟制作上。

答：我效仿的不仅是他的规则。我效仿的是力求得到人类的行为和设计之间的结合。如果你看一下当今的建筑学杂志就会知道，大部分内容都是关于今年的建筑质地，什么颜色了、构造了、装饰风格了等等。这些对人类行为关系都不大。建筑学是研究怎样为人类提供行为、工作和活动空间而设计建筑物的学科。他写的较早的一本书叫《Notes on the Synthesis of Form》，使家的观点更明显一些。他实际上设计了许多关于第三世界的东西。他研究第三世界的部落和文化，研究过着完全原始生活的人并观察他们的活动，哪些活动他们是一起做的，哪些活动是群体合作的。根据这些，他能够按他们的文化推断出设计规则。他们的住房应该安排在什么地方，他们的城镇和他们的村庄应该如何安排等等。我认为这是重新认识建筑学的问题，回到了建筑学实用前提和要求上来，而与美学和建筑学属于当代艺术的观点背道而驰。如果你多看看这些现代建筑艺术的书，是不会住进书中描述的那样的房子里去的。那些房子看起来真的好酷，也真的好漂亮，特别当里面没有任何东西时，它们是那样的完美。我想象不出住在那里面会是什么样的感觉。离现实简直太远了。



问：所以还得建造有自己风格的房子？

答：通过设计要做出更多的现实行为和互动。从某种意义上来说，我要保持我自己的风格。但是还没有等量焦点。



问：当我玩游戏时，我更多的是全神贯注于人与人之间的那种互动影响之中。

答：是的，我想那是焦点真的改变了。我们不会意识到游戏的社会属性是什么样子。原始的概念是你要保持这个家庭幸福美满。你要有这样的思想准备：你家里会来与你发展长期关系的客人，这是一个较新的观点。



问：于是有机地变得不适合于游戏的其他方面？

答：为使我们的行为模型取得成功，运行得比我们想象得更出色，我们必须去做许多工作。或者说，至少要让大家理解我们的行为模型。就是说，我们正在愚弄他们，而且比我们想的还要甚。





问：你是在说人们比实际印象更理解行为模型？

答：事实上，这也成为设计的重要焦点。另一本书成为后来设计中非常有影响的书，就是 Scott McCloud 写的《理解连环画页》。他指出一些非常适合游戏设计的要点。我们使用最多的理念之一是：能动性就是合作，在这里指的是游戏设计者和玩家之间的合作。你呈现的抽象关卡给玩家一个非常有意义的线索。通过这个线索，他们会明白要在电脑上做的模型与他们头脑中想象的模型相差多少。事实上，当某人玩 The Sims 游戏时，他们在表现那种经历，虽然他们还没有感觉到，可他们在头脑中而不是在电脑上正做着许多模型。电脑中会弹出一些急速而不清的对话。实际上，大多数人坐在那里只是粗略地理解了电脑中所说的话。他们会说，“哦，我明白了，他心烦意乱那是因为她没有清理垃圾。”他们在头脑中模拟模型的另一面，比电脑能做到的更详细的关卡。人们对事情发生的先后次序只能看着，却无能为力，只能在它上面写一些叙述。

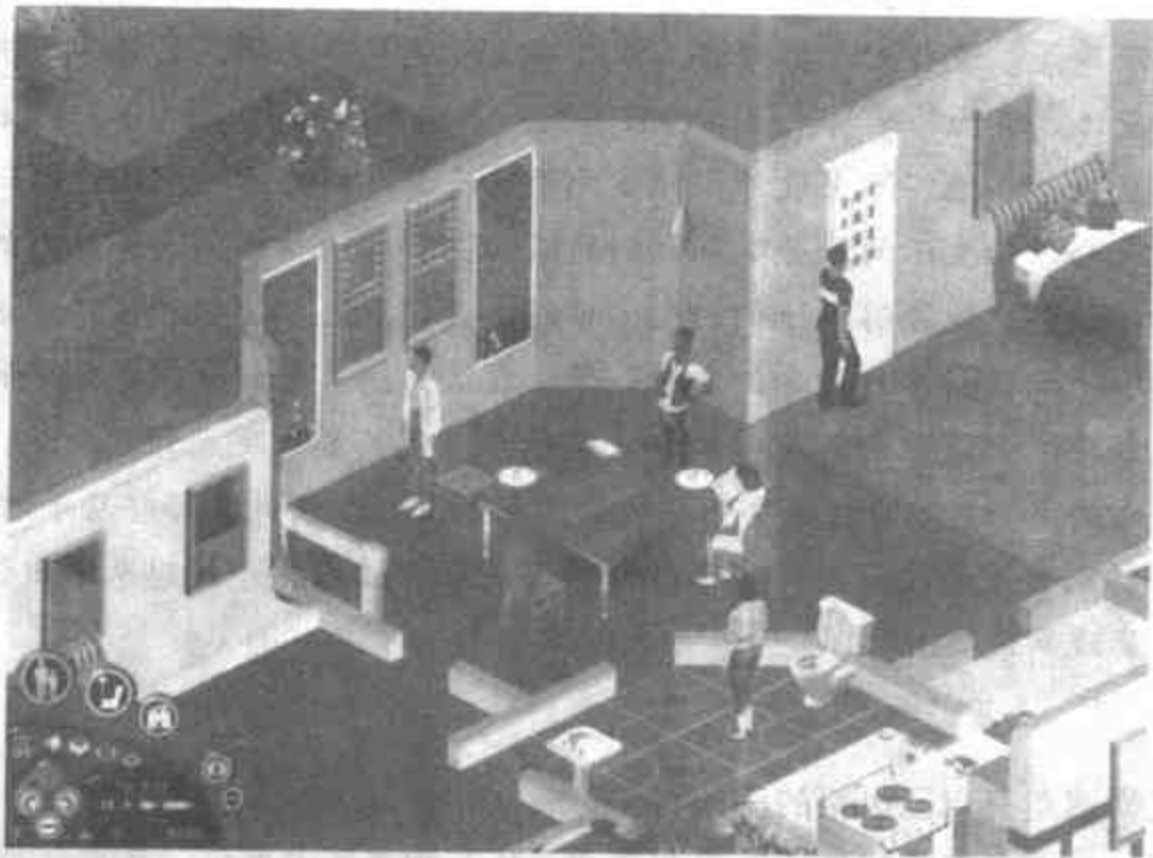


图 22.12 The Sims

我们注意到了那个背景，决定使用它。所以当我们设计那些对话、画面上人物的语言以及它们的姿态时，我们试图留给它们以展开表演，这样会使玩家入戏，尽量让玩家在屏幕上面看见有适当创造性的表演。然后，我们观察人们在早期测试版上玩游戏，他们创造的一些故事情节是那样引人入胜，滑稽可笑。这给我们一个启示，把剪贴簿特性放进游戏里。这样，他们可以实际录制他们自己特有的正在进行的叙述，然后分享它。



问：你认为 The Sims 会是非常成功的吗？

答：我一直认为 The Sims 比 SimCity 有更多潜在的东西。我对 SimCity 从来没抱过什么信心。我也不知道为什么我对 The Sims 抱有信心。只是因为它太接近人类自然家庭吗？我一直怀疑人们会喜欢与他们能接近的人在一起玩游戏。多数游戏不让你接近他人，或者是在那种脚本





化的、线性版式里接近。那不是一种不固定的版式。

① 问：通常，这在 Zelda 一类游戏里出现较多，在那种游戏中你可以与人物说话，但是他们说的却总是相同的事情。

答：确切地说，模型会即刻在你的头脑中中断，你会说：“噢，它是一个机器人，会一遍又一遍地重复着同一件事情。”如果我们能够保持它不固定，并试图不与人们太接近，给玩家留有余地，那么玩家会通情达理地相信这些小动物具有欲望、亲属关系和一切事情。

② 问：在所有的赞扬中，我见过许多对游戏的抱怨。如没有周末了，从不能在外面玩你的 The Sims 等等。对你的游戏你常常听到这样的抱怨吗？

答：听过许多。对 The Sims 的抱怨比其他我制作的游戏要多，或许更多的人认为他们自己是从事蚂蚁和行星热力学学科方面的专家。很难看到 SimEarth，也很难听说：“对呀，我真的认为海洋流动不会与大气层一样有那么快的热能传输速度。”但任何人都可以看到 The Sims，并会说：“噢，我想我不会为了那个而指责她。”在那个领域我们是专家，那是很自然的事情。另一件事是，人们用感觉来评判，不会知道成功和实际工作的比例是多少。因为在他们抱怨关于周末的问题之前，有上百件事情要做。因为周末对我们来说是件大事，它意味着我们许多员工曾经为它留过汗，那可是实实在在的劳动。

③ 问：你如何决定留下哪些功能？删去哪些功能？完成这个游戏需要多长时间？

答：无疑需要很多时间。不论什么时候碰到这样的情况，我们都会努力留下这样一个不固定的游戏，以使我们在下载方面能够扩展。关于下载，我不能全面说明利用它我们能扩展多少游戏。对大家想要的周末，其实也不难，只要他们不想思考我们制作的全部细节直到得出解决办法。当我坐下来向他们解释为什么没有周末时，大多数人会突然意识到，说：“噢，你是正确的，我想我不要周末了。”

④ 问：你如何决定要对模拟加以限制？

答：许多是资源发生问题。我们把那种限制放进夜总会、我们的工作中和其他所有能加进的地方。又过了一年，我们把它加进游戏开发中。就在那个时候，我们度过了最好的时光。另一件事是，我们在类似的程序表上做了我们所能做的一切，但所做的其实更糟。我想我非常希望把房子制作好，而不去做其他事情。实际上，我知道将要发生什么事情，也知道项目的去向。

⑤ 问：所以，你对游戏设计者的忠告就是集中焦点于设计上？

答：你也必须真正懂得游戏中娱乐核心的去向是什么。如果添加这些要素，你可以把更多的子弹点放在箱子的后面，而这样做不会使游戏变得更加有趣，完全是浪费精力。我喜爱一句关于园艺的古老的日本格言，“直到花园里的园艺工作都做完了，你的花园才算完成。”





问：所以你想把格言应用到游戏设计上？

答：就是这样。如果看一下我们在这个游戏中取消要素的数量，或许会让你吃惊。大家知道，我们必备的东西有 8 个。开始是 12 个，后来是 10 个，最终剩 8 个。我们更关心的是简化游戏而不是扩展游戏。我们的界面要通过 11 次重复才能彻底完成界面的重新设计。每次设计都是以不是这里丢下一个按钮，就是那里丢下一个按钮而告终，或者以发现兼有功能的方法而结束。我真的想过，如果 The Sims 能容易让人接近，那将会吸引非常广泛的观众，但它必须是通过界面让人容易接近。界面不能是你的标准战略游戏界面，那样的话我们大多数基础客户都会不感兴趣的。所以我们故意做成那样的界面。大多数人感觉它的某些部分并不是多么精致。我的意思是，界面当中有的地方还很粗糙，而我们真正为之留过汗的东西，却是很次要很次要的细节，但是却以巨大的不同而告终。许多补加的东西是次要的东西，如馅饼菜单等。你既可以点击、拖动和释放一个对象，也可以点击、释放、移动和再点击。我们基本上反映的是多数人习惯使用的 Windows 功能。

随着 3D 磁头的抬起和响应，顺着移动鼠标的方向看着。界面上每一个文本单位都隐藏着帮助。大多数人并不知道这些，你可以向下滚动界面上的字。实际上，当你滚动界面上的字时，那些字会呈现醒目状态。如果点击那些字，它就会显示出非常详细的解释。我们做了许多隐藏帮助。像那样的东西是补加上去的。实际上没有一件事情使其工作。在去年的开发中，我们大概对 100 个游戏测试者进行了测试。其他设计者或者我坐下来看着他们玩游戏约一小时，我们把他们出的错误和误解都记录下来。我们在界面上做了大量游戏测试。如果发生 5 个人犯相同概念错误的情况，或者他们用那种方法试求拖动一个对象，那么我们将勾画出解决的办法而不用中断其他人玩游戏。



问：你总是在游戏中使用传统界面，可是每一个界面又与前面的界面区别不大。那是为什么呢？

答：制作一个有来龙去脉的界面确实不易。真的需要好好看看游戏需要什么，游戏中的事件应该怎样互相作用。只有这样才能决定你的界面怎样做。你也可以看一看你所生存的环境，也就是说别人在运用什么，别人在玩什么游戏？我们在 The Sims 中所做的一些东西与 SimCity 3000 的保持着一致性。如右上卷钮，在那里你可以右击和拖动以及边上卷等。我们试求在那里反映 SimCity。总的来说，你会学明白。我想我制作的每一个界面都比上一个制作得要好。游戏吸引着越来越多的临时观众，而这些临时观众对界面提出了更多的要求。如果你打算抓住这些观众，就要把游戏设计得更容易些。人们玩游戏的电脑一般是硬核的，它们会容纳任何东西。现在玩游戏的人是越来越不确定，如果他们发现界面在两分钟内受挫，他们会放弃。

一般来说，包括我自己在内的 PC 设计者都在追求控制台显示器。这是很久以前人们在 Nintendo 和 Sega 上学的那种控制台，这些控制台要应付那些广泛的临时观众，而这些观众大多数是年轻人。在我们有电脑控制台显示器以前很久，它们就是易受影响、简单且容易理解的界面。





问：对于 The Sims，你有一个 3D 特性的混合世界，在 SimCity 3000 中也是那样吗？

答：是的，因为编辑和建造了那个房子及其他一切，所以我想，要是我们有全角 3D 摄像机以及所有那些我认为像现在这样容易制作的办法就好了。我们也还会有一些图形装载问题。如果是关于几何学的，我们将不能获得与之相关的详细资料。

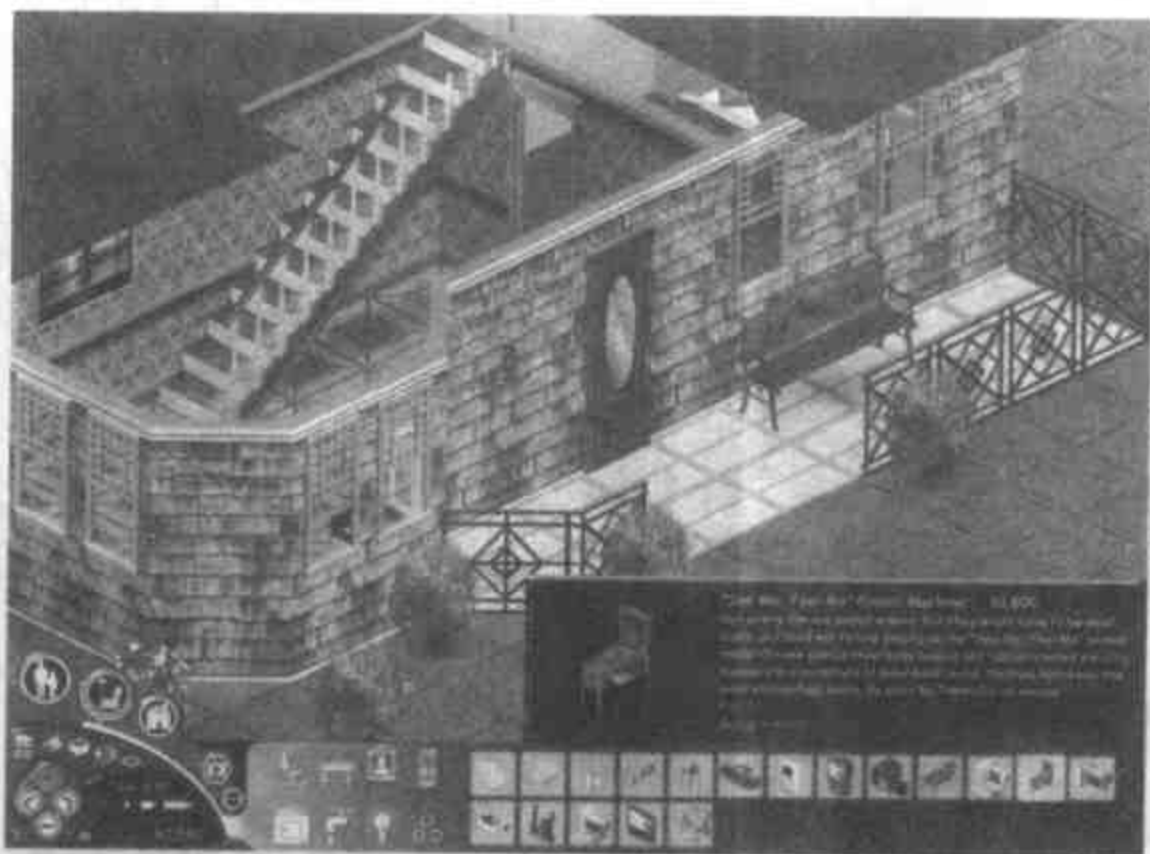


图 22.13 The Sims

问：既然其他许多游戏都是 3D 的，那么你制作 3D 游戏有压力吗？

答：大约 3 年以前，几乎所有游戏都趋向 3D。好像如果你做的不是 3D 游戏，你就要死掉一样。不过现在那种歇斯底里的事情已经过去了，人们开始注意那些销量高的游戏，并且他们意识到：嘿，仍然是你的 Age of Empires、SimCity，所有那些销售好的游戏都不是 3D 的。事实上，看一下那些销售量高的游戏，很少部分是 3D 的。所以现在人们接受非 3D 的思想是既定的。没有必要是 3D 的意识。

问：我非常欣赏 The Sims 中人物的说话方式。是磁盘空间的限制，还是你有意做出快速而不清的说话，为的是留给玩家一些空间去理解？

答：即使我们有 5 盘已经录好的 CD 录音也不够，还是要将它重复的。我最大的顾虑是怕它不能重复，听不到一遍又一遍同样的句子。而事实上我录制了数百条声音句子，每一条都有不同的感情色调。我们决定声音完全是感情的内容：你可以根据声音的语调和节奏等判断出那个人是否爱调侃、容易沮丧、爱打赌或易感疲倦。因为你得不到语义，因为你听不清话语，所以你能坐下来想象流体般的句子。但根据说话的感情色调和前后意思，你会很容易理解句子。你会知道：“哇，她在方便。”

所以，实际上我真的对产生的那种方式感到满意。你一遍又一遍地听他们说话，但很难听





出确切重复的内容。因为，事实上你最终听到的不过是大量波浪声在重复。而实际我们设计的语言是很难查明的。描述怎样制作，确实是一个漫长的过程。起初，我们计划使用真实的语言，一种人们听不懂的真实但朦朦胧胧的语言。我们同 Navajo 和 Estonian 一同做了大量试验，结果他们依然能辨认出来。即使你不懂这种语言，你依然可以辨出，“噢。这个是我刚听过的。”说话时，大部分是用一种语调发出的硬辅音，进行当中既有节奏又有速度。要想把它们都描绘出来那可需要时间了。

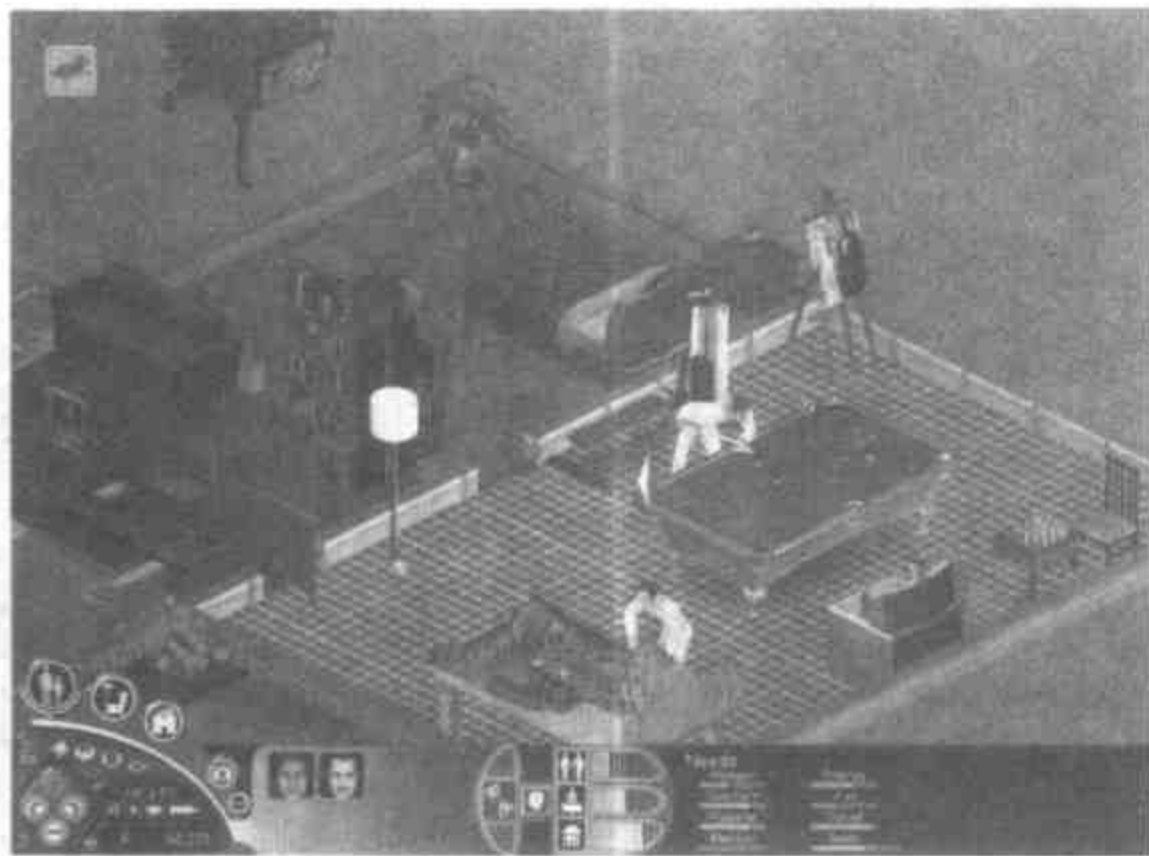


图 22.14 The Sims



问：The Sims 游戏包含了颇有争议的同性恋问题。为什么你允许那样呢？

答：我们知道，这个游戏是现实家庭的模型。我让玩家玩游戏时做的最后一件事是说：“噢，我们不打算结识你的家人。”所以我们想给人们一个合理的、无限制的方式去建造家庭、去想象家庭、去组成家庭。我们现在正在小心翼翼地穿过伦理道德的雷区。在游戏中，有许多事情我们都故意省去。在游戏的各个关卡，我们确实想放进许多事情；同性恋是有争议的（在有些国家甚至是违法的），玩家们应能正确地处理。



问：你都故意省去了哪类事情？

答：有几件事情有点争议，我们稍微做了些修正。其中之一是关于家庭暴力。当游戏中的人物心烦意乱时，他们有可能会互相出手伤人。我不知道你是否注意到，游戏中有两种掌击类型。一种是向后举起手，然后重击，像是要击碎其颌部。另一种掌击类型像是英国军队的那种无礼的掌击。无论什么时候，都有同一性别的掌击方式，他们会使用真正有力的掌击。比如一个男人掌击另一个男人，一个女人掌击另一个女人。但是无论何时，当一个男人掌击一个女人，或者一个女人掌击一个男人，使用的都是温和的掌击类型。因为以前，游戏上出现过一个丈夫





用力掌击他的妻子的画面，这使许多人很恼火，显示出家庭暴力的观点。以上仅仅是我们涉足其边缘的一个例子，我们要小心翼翼，才不失我们的本色。



问：保持那种不带有太疯狂的情感色彩内容。

答：对，不让人们想起残忍的家庭虐待。事实上挺有意思，因为我们也制作了一个相互攻击的情境。如果他们真的彼此不喜欢，他们会互相打斗。但是因为我们做的是像卡通片里那样的打斗，有一大朵云彩，你能看见有胳膊和腿伸出来，对此无人怀疑。即使是一个男人和一个女人，因为是卡通化，也不会出现像掌击那样的结局。有些地方我们一点也不想介入。比如：恋童癖。一般他们不会互相残杀。虽然游戏里的人物会残杀他们，各种灾难会残杀他们，但 The Sims 本身不会直接相互残杀。所以，有些我们曾经确定的东西也会被删除，有些东西我们要安排，还有些东西我们会特别小心地对待。



问：对于包括像同性恋这样的问题，那些一直关心暴力的参议员，对 The Sims 有异议吗？

答：实际上有，但让我吃惊的是丝毫没有具体化。一点点都没有。对此真的没有一点反映，所以让我非常吃惊。最初我想，如果出现异议的话，应该来自基督徒、保守派或其他什么团体。或许他们不玩这些游戏，或许他们不关心这些游戏，我也不知道究竟什么原因。我们绝对没有一点问题。在海报上有几个大概是 14 岁孩子发的牢骚，根据他们说的话，你能猜出他们的年龄。



问：在游戏设计方面，你似乎做出了许多道德方面的决定。例如，玩家似乎改进了你的职业，使你获取到更多的素材。这种观点好像是唯物主义的。

答：是的，那确实是我的目的。正像多数人见到游戏时所能理解得那样。他们玩了一会儿，就会认识到那种游戏是非常唯物主义的。只有长时间玩游戏的人才会开始意识到下边的情况。每一个对象都会有一些制作方面的失败情况或必要的条件。如果你继续购买素材，最终它会变坏，或者死掉或者需要被清理掉或者其他什么的。就好像你正在自己的房里设置那些潜在的定时炸弹。就某种观点来说，你结束了花费那么多时间处理的这些事情。那些你购买的物件为你节省了时间，不再浪费你的时间。从你进入游戏到你开始意识到这些，要经过相当长的时间。用那种方法进行设计是非常明确的。所以从某种意义上来说，第一次玩游戏的人会说：“天哪，我不能相信这个游戏是这么唯物。”但是那些发烧友却会说：“天哪，下次我再也不会买这废物来玩了。”



问：我猜想，玩家能够试图把精力集中在社会方面，而不是物质获得方面，够开放的了。

答：从某种意义上来说，社会方面有同样的动力。你在社会上交结了一些朋友，可是友谊会随时间的推移而渐渐减弱。你的朋友，一旦某一天不得志了，会打电话说：“嘿，最好邀请我过去，有一段时间没有见到你了。”一旦你交了 20 个这样的朋友，你就会开始注意到，每天都有人吵着要过来，他们以不同的方法耗费你的时间。





问：你能谈谈关于 Edith 脚本语言的一些情况吗？

答：好吧。我和 Jamie 在那上面花费的时间最长。它是一种程序脚本语言，是直观的。那实际上是开发了我们的编辑程序和调试器，将它们与游戏集成为一体。你玩游戏时它就在运行。你可以一边玩游戏，一边通过对象编程、调试和设置分步程序。



问：所以你能用它把新对象加到游戏里去。

答：事实上，游戏中几乎所有行为都是这些对象，包括人们在社会中的互相影响，都是以这种语言编写的。这种语言都在 C 级程序上。这种 C 级代码程序类似于路径选择原语、变量的置与取操作。这种语言本身是非常规范的，大概有 30 到 40 种原语都出自于此。虽然主要的事情是贯穿于对象之中所有独立于机器的、象征性的代码。也就是说，你可以在游戏中加入新的对象，并且人们会即刻知道什么时候使用它、什么时候可以使用它、如何使用它。动画、声音效果、代码以及一切都包含在你下载的对象之中。



问：因此你创造了更容易添加新对象的语言。

答：是的。它是语言原始形式的说明。我们想有一种能够在目标关卡上编写所有可扩展行为的语言。人们在房子内的行为方式完全是他们房间内要素的功能。我们总是能够添加新的事物，甚至特洛伊马，都可以放在房间里。



问：诸如豚鼠对象之类。

答：是的，豚鼠对象就是一个例子。在设计中，我们认为豚鼠应该患病，我们已经计划给它制作疾病，但我们却运转超时。后来我们意识到，“嘿，我们能做一个下载。”当然了，没有人打算下载疾病。我们把它隐藏在豚鼠内。有趣的是，一些游戏的早期检查官却说：“已经检查了所有的要素，没有疾病。我不明白为什么。”当然，他们或许是当我们把游戏给他们时才抱怨的那种人。我们发行这种语言的理由是，最终让使用者开始制作这类事情。



问：你使制作变得非常简单，即使不是硬核程序员，也可以使用它。

答：你要知道如何编程，但你不一定非是硬核程序员。我的意思是，它比 Visual Basic 语言要简单得多。



问：用一个像那样的工具，决不会“做好”游戏，这种观点不烦扰你吗？

答：烦扰的。可我还是认为，即使你回到那种消遣模型上，也没有人会去制作消遣模型。消遣模型仅仅是不断发展的东西，但其吸引人的功能增长得非常快，这真是一种罪过。它们的工具功能越强，消遣本身也就越强，还影响着更多的人。





问：我读过你的格言，你说过：“The Sims 长期的吸引力是因为它已成为一个讲故事的平台。”现在，当大多数游戏开发商谈论起游戏中的故事时，用的都是 Zelda 观念。他们认为，像 The Sims 这样的东西，没有任何故事可讲。

答：Zelda 和 The Sims 之间存在着非常明显的区别。在 The Sims 里你是在创造故事，而在 Zelda 里你是让大家看故事。就

某种意义上来说，故事只是玩家涉及到的事物的一个方面。事实上，所有这些都不同的水平。

那些偶尔玩游戏的人，玩几个小时，娱乐一下，就放下了。而长时间玩的人，把精力投入到真正

地设计一个酷的房子，或许还会将它们放到网址上，让其他人看。另一些人，投入游戏更深，

他们不仅建立有趣的家庭和酷的房子，而且还用它讲故事，放到互联网上让其他人共享。更有甚者，那些发烧友开始编辑游戏中人物的皮肤或者壁纸一类的

东西，并与其他人分享。不久，他们会创建自己的对象和顾客对象，并将其放到网上与人共享。

所以，玩家有着涉及事物的不同水平。水平越高，玩的人越少。从某种意义上来说，他们是在为低水平的人提供游戏。我们为上万个人在我们的网站上安了家，而那些上万个家已经被数十万次访问过。

我们为上万个家已经提供了数十万次访问过。

所以，玩家有着涉及事物的不同水平。水平越高，玩的人越少。从某种意义上来说，他们是在为低水平的人提供游戏。我们为上万个人在我们的网站上安了家，而那些上万个家已经提供了数十万次访问过。



图 22.15 The Sims

问：所以，它像是个金字塔型设计。

答：确实是这样。好像有 30 人在制作游戏对象的漂亮皮肤。但是大概有三万人在下载它们，使用它们。因此，对于那些真正的天才发烧友来说，如果给他们提供工具，他们做出的内容 99% 会让你满意。这样做对双方都有利。把工具提供给那些想建造对象的观众，也就是给观众一个酷的游戏要素，那样会更深地抓住观众。并且，这还将增加那些偶尔玩游戏的人最终成为发烧友的可能性。

问：会有一天，行星上的每个人都要玩 The Sims。

答：对。这种游戏就像神蛇计划一样。神蛇到达某个地方，开始吃大脑，然后把其他人都变成神蛇。开始，只有 5 只神蛇与世界对抗，而接下来情况看起来不太好，一旦某人得到起关键性作用的神蛇，他们会相当快地把其他人都转变成神蛇……





问：在 The Sims 中，你只被列为一个游戏设计者，而在过去，你既是程序员又是设计者。你在项目中做过编程工作吗？

答：我用 Edith 代码做过大量编程工作。在 The Sims 中，我未使用 C 码进行编程。The Sims 大概是我第一个没有参加编程的项目。我用 Edith 代码做了许多社会交互作用和要素的编程工作。但那个时候，我研究的大部分难题是进去、扭动以及转动等规则系统，这些规则系统都是我想要用的。我们确实用了一个好团队来制作 The Sims，一个真正伟大的工程师团队。因此，我认为我没有必要去涉足编码。



问：你不怀念什么吗？

答：噢，我要怀念的。对 Edith 和 hacking 要素，我很欣赏。在设计方面有更多的事情要做，所以我没有时间花在编程上。并不是说编辑程序是浪费时间，并且我也决不是一个伟大的程序员。我一直很固执，我能用计算机代码做出更酷的要素就是因为我国执。我的意思是，我了解伟大的程序员，但我不是。



问：你向工程师团队传达你的设计意见时遇到过麻烦吗？

答：遇到过一些。但不是那种缺乏远见之明和理解力的麻烦。而仅仅是因为那是一件复杂的事情。事实上，我不知道我们这么长时间在建造什么，大部分时间都是在实验。用编程人员的话说，我可能总是坐下来，解释我在找的动力，并且自信能找到它们。



问：你经历了完全由自己动手制作 SimCity 到领导一个大团队来制作 The Sims 的转变。这个团队有多少人呢？

答：人数取决于你如何计划。大家知道，有时会有 60 人在工作，可我考虑团队人数应再增加 30 人。



问：从在一个小组里工作到领导一个大团队是一个非常大的转变。对一个大团队来说，管理是相当重要的。

答：是的。对于团队人员的素质，有大量的事情要做。《电子艺术》也以完全不同的定位加入进来。他们进入以前，我们只有四、五个人编制 The Sims。当时确实是一个非常优秀的小组，做出了很大成就，可我不能得到更多的财力支持。当《电子艺术》加入进来时，就会提出：“你们需要什么？”这样的问题。那时，我们正好开始组建团队。《电子艺术》也有一个非常强的开发商要开发的概念。它们好像有 10 个级别的开发商，给予开发者非常重的负担。所有这些事情之一是，如何把合适的人员放在合适的位置，以便有效地管理一个非常大的团队。如果你没有把合适的人员放在合适的位置，那可是大错特错。在那个关键时刻，雇用实践成为重要的事情。如何面谈，才能确保招聘到合适的人员，如何快速发现你所招聘的人员是否适应工作。所以，把合适的任务和合适的人员放在一起是一种工作模式。但如果招聘到的人员不适合工作，





你就应该把他辞掉。

我们基本上还是招聘到了合适的人员。同时，在 Maxis 公司的工作中，《电子艺术》带来一个小伙子，他负责技术室，对我们老式的管理模式进行了很大改进。他叫 Luc Barthelet。Luc 和我从那天起开始在一起工作，我们相处得非常融洽。从各方面来说，Luc 都不是一个典型的管理者。我的意思是说他对技术非常精通。当时制作的 SimCity 3000，在交通模型上正好出现了问题，他加盟进来，编写了交通代码。



问：果然如此吗？

答：是的。他编写了 C 级代码。不寻常的是，有人既能管理技术室，又能编写模型中最复杂的代码。Luc 就是这样一位小伙子。他确实有管理艺术，Luc 的伟大之处在于他确切地知道哪一天需要在哪个关卡上集中精力。也就是那个时候，正是严酷的时候，我们看到了一篇关于 SimCity 3000 的特写报道。这篇报道对产品的成功将有很大的冲击。也就是那一天，他搬出他的编辑器，开始编写交通代码。在大多数情况下，他会想：“德国的分销商对这个产品感觉怎样？”并且他打电话给德国分销商进行询问。你确实不得不选择竞争。如果你选择了正确的竞争，也就赢得了他们的百分之五。所以不管怎样，有了这些商业见识，即《电子艺术》带来了观众。了解了这些，我被深深地打动。



问：当设计和开发 Sim 系列游戏时，有什么必须遵守的指导原则吗？

答：有。虽然我们偶尔打破那个规则，但基本上总是将游戏的大部分设计成非暴力的。因为我们已经在考虑一个我们的显著特征。我们的许多职员确实想为 Maxis 公司好好工作，因为 Maxis 公司以他们的非暴力游戏而知名。我不想宣称，好像我正在发表道德陈述，因为我喜爱 Doom 和 Quake 及属于我自己的东西。我最喜欢的游戏是战争游戏，我沉迷于战争游戏。我认为有许多人在制作那些我们不必要制作的游戏，不过他们也做得非常好。所以我宁可制作没有人制作的游戏。但是来自社会大众的观点，我们确实因为倾向于制作更多非暴力的、更多与教育与社会有关联的游戏而赢得了声望。



问：制作 Maxis 风格的模拟游戏，你是否有一种被束缚住的感觉？你想制作 Raid Over Bungeling Bay II 游戏吗？

答：从某种意义上讲，SimCopter 几乎就是 Raid Over Bungeling Bay II。在 SimCopter 中，隐藏着许多复活节炸弹。事实上，你可以制作一个 Apache 去破坏城市。如果你有了 Apache，你就可以发现核电厂，就可以毁掉整个城市。甚至在 The Sims 中，有几次，我试图避开各处政治上的处罚。所以在 The Sims 中我们做过的许多事情不是绝对符合一般政治性准则的，但是大家知道，谁都没有意识到那是更古怪方面的。我们努力不让 Maxis 的事情束缚住我们，家庭暴力或许就是一个很好的例子。你会看见，许多的游戏里有比暴力更甚的内容，还有比一个男人掌击一个女人更甚的内容。但是我们对人们在玩游戏时怎样理解这些内容，怎样认识家庭是很敏感的。



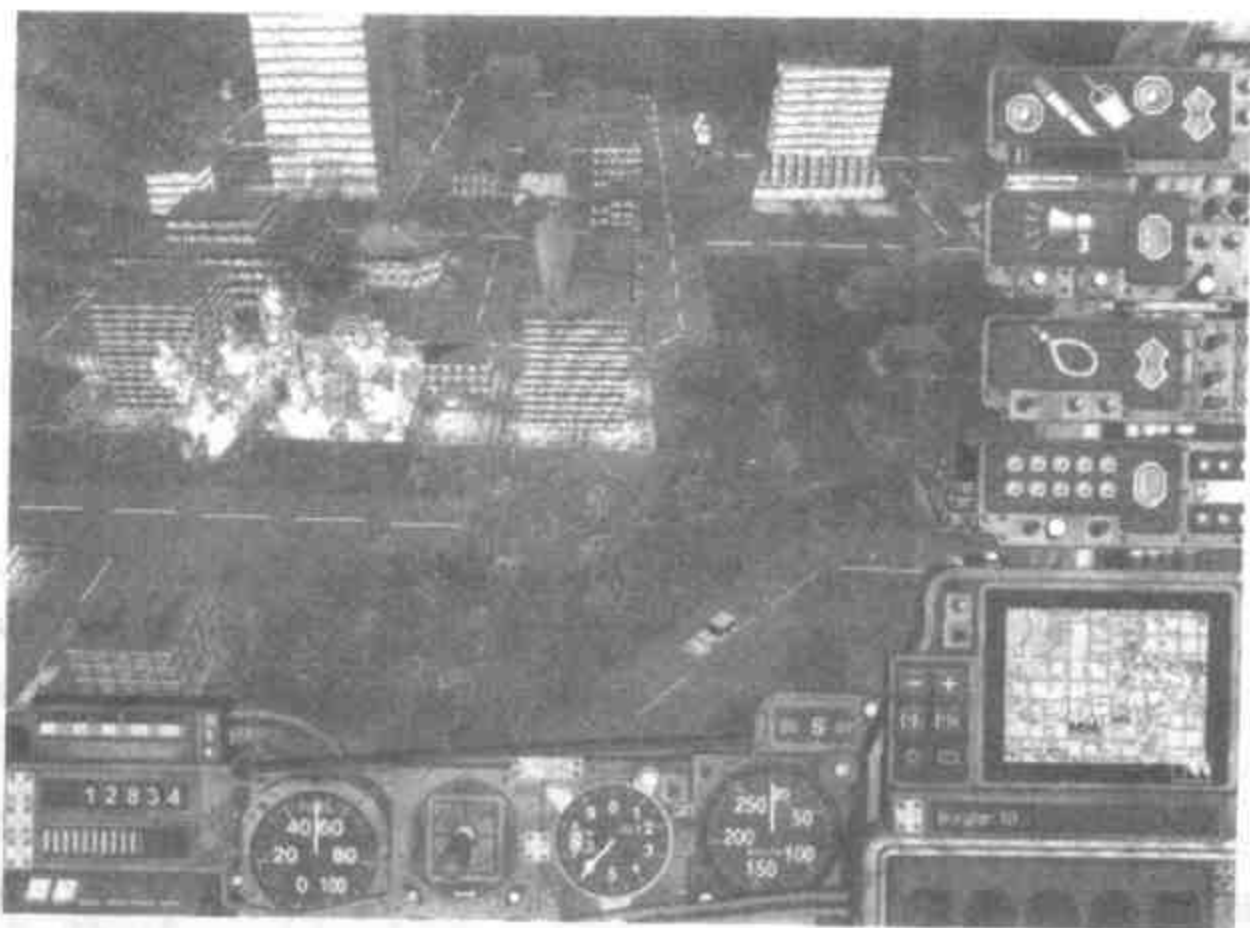


图 22.16 SimCopter



问：你的游戏总是看似富有较强的教育成分在里面。我想知道你是如何平衡教育与制作娱乐游戏的？

答：直到我感觉游戏有趣，才关心教育的。如果人们不去玩它的话，程序具有的任何教育价值也不过是一种浪费。我从 SimEarth 上学到了非常多的东西。SimEarth 是我制作的游戏中最潜在教育最多的一个，但它并不是只为玩玩用的。购买它的人数令人吃惊，销售数量也令人吃惊。我想购买它的大部分人玩上两个小时，然后就会把它放到一边。因此，我确实以为兴趣应该是第一位的；教育方面，不是你要首先涉足的东西，要从基础设计中得到。在 The Sims 中，要从行为中学习推断设计。那确实是很深的课程，不仅仅是我正在讲述的事情。那更像一种观察事物的方法。如果整个设计是真实的，它可能在某些深层级别上是一个具有教育意义的游戏，你可能玩上数小时游戏并没意识到是在受教育。SimCity 教育的主要特点之一是，它不是在述说，而是在那里存在，虽然是无序状态。事实上，精心安排的设计可能总是出错，系统比你想象的要复杂得多。铺建一条道路的目的是解决交通问题，而不是让它产生交通问题。这一类型的课程在其他媒体上是很难解释通的。但是当你已经经历了像制作 SimCity 的过程，你将会得到更深的知识。与其说是解说倒不如说是经验。



问：你曾经有过放弃现实而去制作有趣的的游戏吗？

答：噢，不但有过，而且是每时每刻。在游戏里有些经常发生的事情，比如我们决定与期望相匹配但又不真实。事实上，核电厂没有爆炸。它们是不爆炸的。但是每个人见到它时，都会说：“噢，一个核电厂，我能让它爆炸吗？”这仅仅是他们想这样。所以我们做的许多事情，只是因为人们希望它们发生，而目的是为了取乐，尽管它不是真实的。



问：关于你的随意性种类的游戏，你要花费许多时间进行游戏测试吗？

答：是的，测试花费了我们许多非常宝贵的时间。你要花费那个时间，否则你将花费数月的时间制造错误的事物，解决错误的问题。我们仅仅在我们工作的 The Sims 多媒体播放机的一个小部件上进行了我们称之为“kleenex”的测试。我们有这个旋转缠绕的数据显示。几天前，程序员把它做完，所以今天我们安排了5个人来这里。我们称他们为 kleenex 游戏测试者，这是因为我们雇用他们一次，然后就不会再雇用他们，我们希望，人们以前没有见过这个小游戏，对它完全没有偏好。我们甚至不告诉他们是什么东西，我们只说：“看一下那个，玩玩它。”让他们给我们描述他们见到了什么，那个代表什么。今天我们得到了那5个人非常一致的回复，我们理解了，我们向他们传递的三个可变信息，他们都明白了。对另外三个可变信息他们还没有思路。因此，对最后的测试者，我们去掉最后三个可变信息，测试是理想的。现在，项目的每个阶段我们都这样做。并不是当我们把全部事情都做完时才做测试，小部件我们也这样做测试，甚至艺术原形我们也这样做。我接受这个课程，它是由已故的 Dani Berry 教给我的。她负责制作 M.U.L.E. 和所有其他事情。数年前她告诫我，游戏测试是最容易被轻视的事，任何游戏设计者都应该进行游戏测试，并且确实是你必须做的。我开始采纳她的忠告，她是正确的。这么做的结果对游戏的成功是不可估量的。

问：关于 SimCity 和 The Sims 这两个游戏，你曾经在说服别人时有过烦恼，而实际上这两个游戏将会受到欢迎。你认为有许多游戏存在着同样的问题吗？当时绝看不到今天的黎明吧？你向那些制作古怪游戏的人建议过应该做些什么吗？

答：噢，我确信那已经是事过境迁了。考虑这个问题让人沮丧，多少优秀的杰作已经面世。作为我，不过是一个非常非常固执的人。我想，如果你确实是固执的人，如果你确实想要什么东西，你就能使它发生。那可能要花费几年的时间。实际上，SimCity 好像是用了5年时间才发布第一版的，The Sims 好像是用了7年。撇开这个不谈，以我的跟踪记录为基础，我不知道我是否是提供忠告的人。无论什么时候，The Sims 出现了不正常

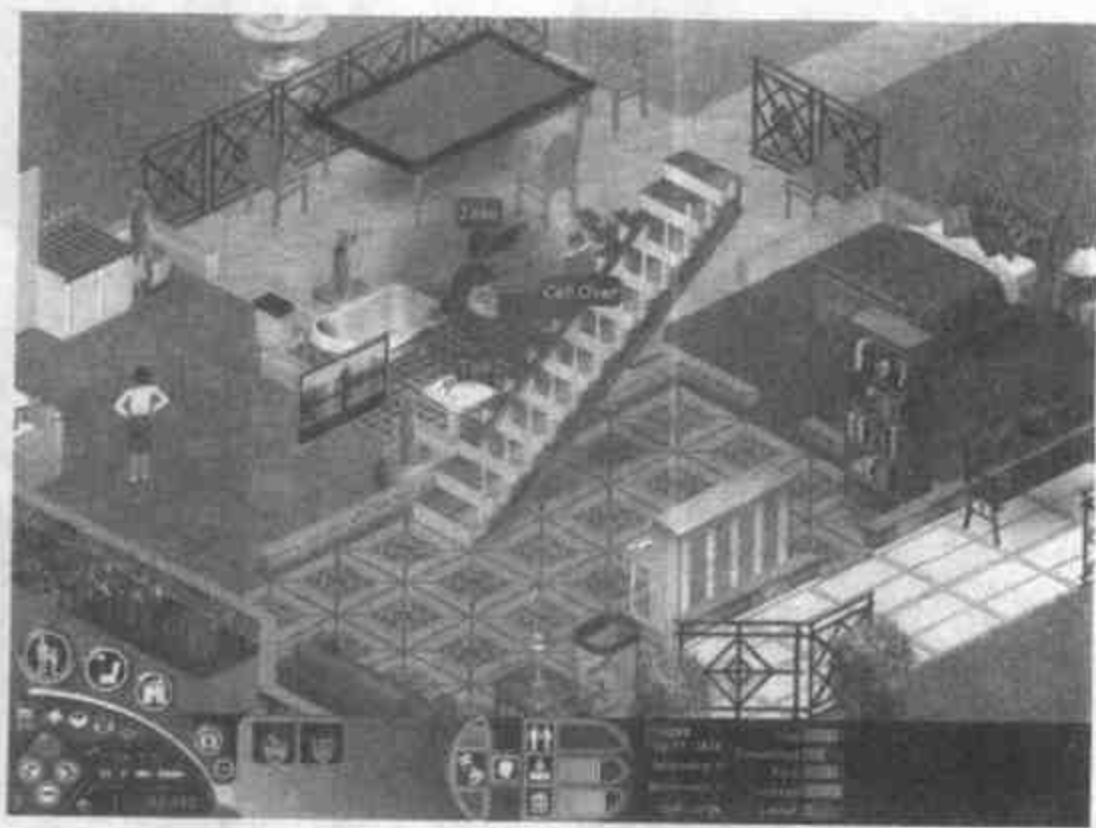


图 22.17 The Sims

的一些事情，我都相信，所有突发事件的当事人仍会说：“嘿，那个游戏的确不一般，销得太好了！”或许那样会放行一些其他公司的项目，其他公司有问题需要得到证实。但是切合实际





地想，他们会更愿意说：“噢，我要一个像 The Sims 那样的游戏。”不幸的是，那或许正是他从中吸取的教训。

Will Wright 作品一览

Raid Over Bungeling Bay, 1984年

SimCity, 1989年

SimEarth, 1990年

SimAnt, 1992年

SimCity 2000, 1994年

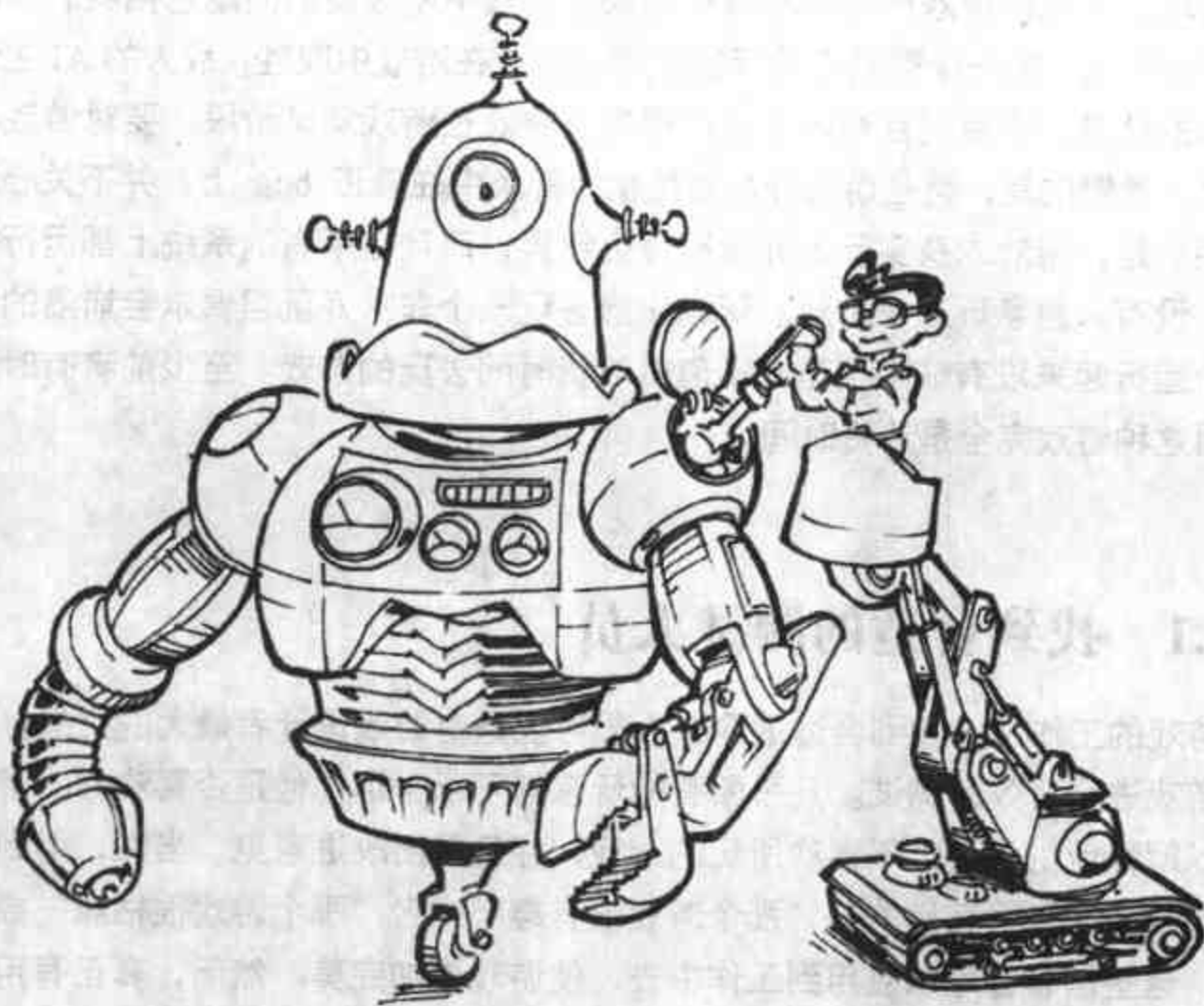
SimCopter, 1997年

The Sims, 2000年



Chapter 23

第23章 游戏测试



“我猜测，共同的特点是热情。每个人都说‘为什么游戏不能做得更好呢——为什么找不到更多的好游戏？’我认为，问题的答案是这个行业无法做到的，那就是游戏的制作者没有去玩自己制作的游戏。我们制作游戏，并且让开发人员每时每刻都在工作，但没有给他们时间来玩游戏。这就是问题的所在——坐下来，并且用大量的时间来玩这些游戏。”

——Peter Molyneux

在游戏的开发过程中，游戏测试是最令人愉快的部分。这时你可以拿到开发人员耗费数月时间开发出的项目，此前只有开发人员玩过游戏，并且你要把它拿出去展示给人们。如果情况进展得比较顺利，你可以看到游戏使人们娱乐，人们想用更多的时间来玩这个游戏，这不仅肯定了你的工作，而且还有助于你制作出更好的游戏。游戏测试并不仅仅是将游戏提供给生产商



或者上传到 Internet 的手段。实际上，游戏测试是一个很关键的阶段；在这个阶段中，游戏可以在水平上有质的飞跃，并且可以真正实现原来允诺在游戏中加入的内容。没有哪个出自开发人员之手的游戏是绝对完美的。从理论上来说，只有在测试阶段，游戏才能改进得更加完美。

当谈到游戏测试的时候，我必须澄清自己所要表达的意思。这个过程与调试过程不同。调试是面向编程的任务，在这个阶段中要找出游戏有内在缺陷的方面，并且加以修正。这些问题涉及游戏机制中的不足，以及图形混乱和实际可能会导致游戏崩溃的问题。这些 bug 必须消除掉，但这是编程人员要解决的问题。

游戏测试相当于在设计阶段的 bug 修正。当游戏测试人员拿到游戏的时候，他们会尝试判断游戏是否好玩，并找出游戏设计本身存在的缺陷。其中所涉及的问题包括实时游戏中的某个物品的功能过于强大，第一个得到它的玩家就可以完全在游戏中取胜；敌人的 AI 主体攻击玩家的不符合逻辑的特性；不直观且难以使用的控制系统。在游戏测试阶段，要对游戏机制本身进行测试和改进。遗憾的是，有些游戏开发者把精力都集中在修正 bug 上，并不关心游戏是否真的好玩。结果就是，游戏本身实际上并没有什么错误，而且在所有的系统上都运行得很稳定。但糟糕的是，没有人愿意玩这个游戏。玩家宁愿去玩一个非常好玩但偶尔会崩溃的游戏，而不愿意去玩一个运行起来没有缺陷但是却不值得浪费时间去玩的游戏。至少前者有时还是很好玩的，而玩后面这种游戏完全是浪费时间。



23.1 找到合适的测试人员

在测试游戏的工作中，找到合适的测试人员可能是对游戏设计者最大的挑战。并不是每个人都可以对游戏进行有效的测试。几乎所有的玩家都可以告诉你他是否喜欢你的游戏，但是只有少数人可以解释他们为什么不喜歡所玩的游戏，并且提出改进意见。当然，得到某人关于游戏一般印象的反馈也是很有用的：“那个游戏很有趣”或者“那个游戏很枯燥”或者“那个游戏太难了”，这些信息都可以应用到工作中去，使游戏更加完美。然而，真正有用的建议会采用一种更有建设性的形式：“当我在第三关中与第 12 个小丑打斗的时候，我觉得他简直太难对付了。我根本不知道如何才能干掉他，也不知道我进行的攻击到底对他有没有效果。我想自己可以向它扔一块石头，但我不知道怎样去做。”在这个例子中，游戏测试者就所存在的问题为设计者提供了非常详尽的信息，并且对游戏并不好玩的原因给出了详细的解释。可以做出这种分析的游戏测试者一向非常少，如果游戏的开发队伍中有一位有天赋的游戏测试者，那真是非常宝贵的财富。

与游戏测试者顺利合作的关键就是了解他们所提意见的认真程度，以及他们可能会有的偏见。不同的测试者会有不同的动机，这样就会在某种程度上影响他们提出的观点。这就是随便在街上找到一个人来测试游戏可能并不起什么作用的原因，因为你没有与他一起工作的经验，也就不知道自己是否可以相信他的意见。如果你有过与某个测试者共同工作的经验，那么就可以了解到这个人是否有某些缺点。例如，有些测试者可以称之为“牢骚虫”，它会抱怨所有的东西，甚至不存在缺陷的内容他也会抱怨。而有些测试者可能会比较拘谨，他们只会说：“你可能需要看一下 Elephant Rider 单元的能量”，他们的意思就是：“很显然，应该从游戏中去掉



Elephant Rider。”要尽可能地了解一同工作的每个测试者的个性；利用他们为你提供的反馈信息非常重要。

23.1.1 应该找哪种测试人员

一个项目中可能会有各种不同类型的测试人员，最好能够在每种类型的人员中都选出一些参与到项目中来。没有哪一类测试人员可以提供项目所需要的所有信息，因此项目中需要各种类型的测试人员。实际上，项目中最好能有多个测试人员的参与，因为广泛的意见可以消除个人意见的影响，并且能够真正了解到游戏是否好玩。有人说应该将开发队伍的规模限制在较少的人数之内，尤其是设计人员和程序员，而有更多测试人员的参与会使工作更为轻松愉快。

第一类游戏测试人员是开发队伍的成员。在整个项目的开发过程中，让游戏的开发人员来玩游戏是很重要的。其中包括多个目的。首先，这样可以使他们保持对项目的热情。他们能够看到自己制作的美工、声效、代码或者在游戏中使用的关卡。其次，当他们看到自己工作的成果可以运行时，就会更好地了解如何对其进行改进。再次，他们可以就游戏如何运行和如何进行改进提出反馈意见。尤其是在项目快要结束的时候，当所有的美工、大部分代码和关卡都已经完成之后，对于游戏中需要做出最终改进的部分，开发组中的成员可以提供非常重要的反馈信息。当然，开发组中的成员是直接参与项目的人，因此他们的意见可能会缺乏客观性。此外，因为他们已经玩了很长时间的 game，所以他们很难以全新的角度看待这个游戏；他们的观点会有所偏差。还因为他们参与了项目的开发工作，所以出于个人的原因，他们可能会倾向于喜欢或者讨厌自己的工作成果。同样，他们也有可能喜欢或者讨厌开发队伍中其他人的想法，这并不是因为这些想法本身的特点，而是因为他们对那个人个人观点的好恶。尽管有这些弊端的存在，从开发队伍中获得游戏测试的信息反馈还是非常重要的。

第二类游戏测试人员是传统意义上的游戏测试人员。这些人在游戏进入“alpha”阶段的时候就开始对游戏进行测试，这个阶段中的游戏已经是成品，并且这种测试工作一直要持续到项目交付时为止。通常，这些测试人员会将一半的时间用来跟踪代码中的 bug，并且还会提供一些重要的反馈信息：比如如何玩游戏、游戏的难度过于简单还是过于困难、游戏的控制是直观还是迟钝等等情况。在资金充足的项目中，这些测试人员一般都是受雇的员工，他们要花费一整个工作周来测试游戏，并且要提交 bug 报告。这些测试人员都很喜欢玩电脑游戏，而且也玩过很多游戏，无论是在这种兼职工作中还是业余时间里，他们都会玩游戏。因此，他们关于如何改进游戏的观点更加趋向于专业级玩家的角度。此外，因为这些测试人员会参与到项目中很长时间，所以他们会习惯于游戏中某种固有的问题，而且可能不再抱怨那些缺点。

第三类测试人员比较注重第一印象。Will Wright 在他第 22 章的访谈录中将这些人称为“克里奈克斯型（一种面巾纸的商标名）测试人员”，因为在 Maxis 公司，这种测试人员只用一次，并且以后就不再雇用。Wright 主要是让他们来测试 The Sims 的 GUI。这些人既不是开发组的成员，也不是全职测试游戏的人员。实际上，这些测试人员只是短时间内参与到项目中，从事玩游戏的工作，然后提供对于玩游戏最直接的感觉。这段时间可能是几个小时，也可能是几天。这些第一印象测试人员的工作也很有帮助，因为他们以第一次玩游戏的玩家身份来看待这个游戏。关于游戏中不太直观的控制方式、不清晰的信息表示或者游戏中难度设置不合理的部分，





他们都可以提供重要的反馈信息。选用第一印象测试者时需要注意的一点是，你必须不断地选择新的测试人员，因为一个人对一个游戏只能有一次第一印象；他们受到在游戏中了解到的某些情况的“污染”之后，就不会再有第一印象。尤其是到了项目完成的时候，当开发人员已经对游戏非常熟悉，并且传统的游戏测试人员已经玩过了数千小时的游戏，判断游戏学习起来不会很难，第一印象测试人员的工作是必不可少的。

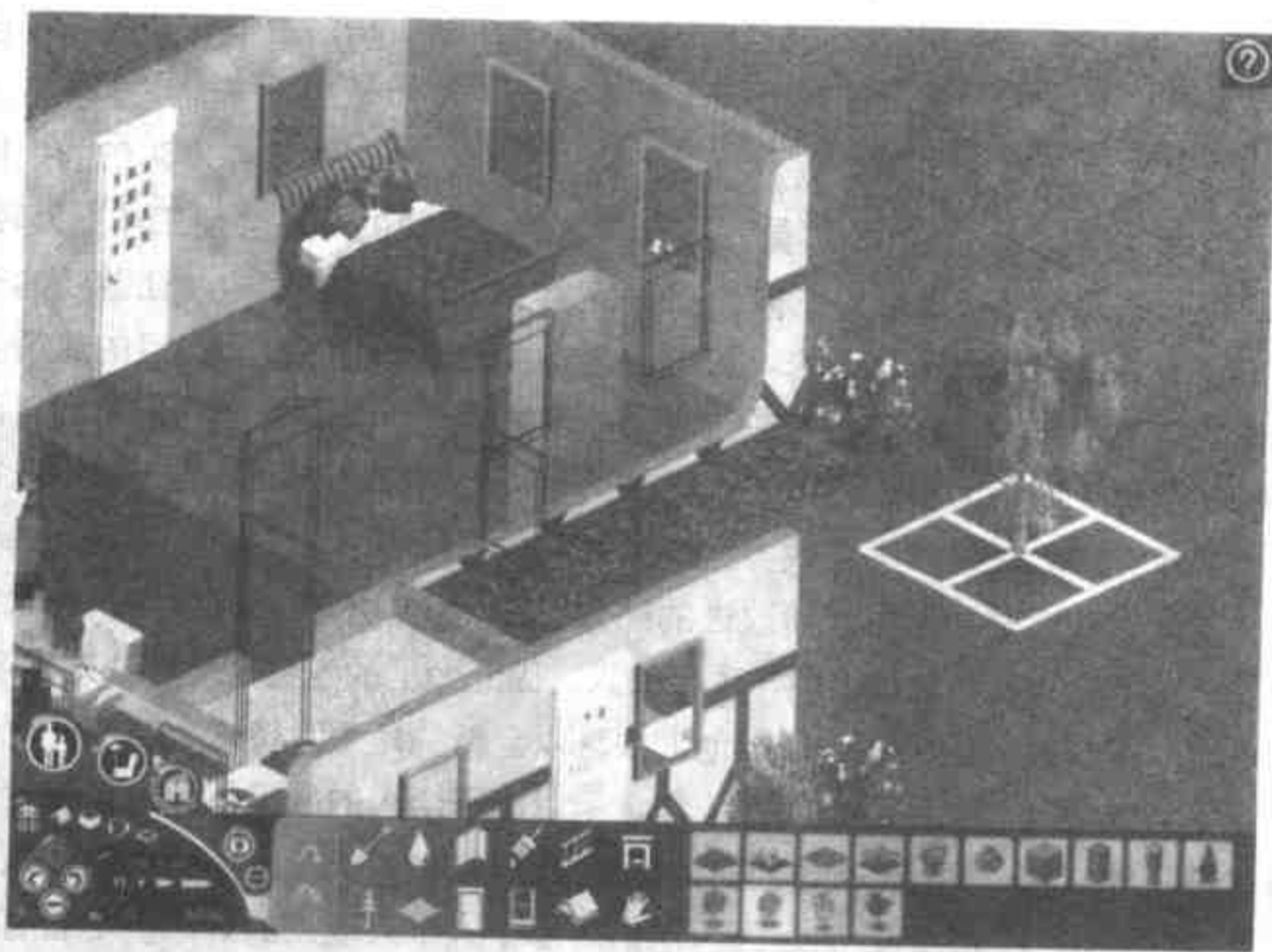


图 23.1 许多第一印象测试人员都可以对 The Sims 中的界面加以改进，并使其更加完美

第四类游戏测试人员是游戏设计者或者开发人员，他们并不实际从事项目中的工作。这些人是你了解并信任的，并且你会尊重他们的意见。他们可能不会像全职的游戏测试人员那样对项目进行测试，但是他们提供的反馈信息会非常有用。不在项目中参与工作的游戏设计者也可以玩你的游戏，并且会采用比其他测试人员更为独到的方式提出自己对游戏优缺点的见解。这些测试人员在某种程度上理解游戏设计的思想，可以分析出项目为何存在缺点，并且提供改进方案。许多有经验的游戏设计人员都会在开发过程的前期使用这些测试人员，在那个阶段，他们想了解新的设计方案是否会吸引人。这些游戏设计人员在作为测试人员之后，可以更好地忽视游戏早期阶段中明显的缺点，如那些 bug 或者不完整的特性，并且更有远见地看出将来的游戏会按照设想成为一个出色的游戏。在第 10 章中，Steve Meretzky 提到了“Imp Lunches”的作用。在吃午餐的时候，Infocom 公司的制作人会聚集在一起讨论他们各自不同的游戏设计思想。当 Infocom 公司又一个新的主题游戏面世时，其他的制作人就会首先来对游戏进行测试。当然，一般情况下这些游戏设计者都非常忙碌，不会有太多的时间来玩游戏并且提供反馈信息。无论这些设计者们提出什么反馈信息，这些信息都是非常有帮助的，既可以帮你处理预期没有想到的问题，又可以肯定你的设计方向是正确的——如果实际上的确是正确的话。

第五类我认为非常有帮助的测试人员并不是游戏玩家。以前我所提到过的所有类型的测试人员都是游戏爱好者，他们都有很大的耐心，可以忍受游戏中糟糕的东西，比如控制过于复杂，



或者游戏太难玩。让那些并不是很喜欢玩游戏的人来测试游戏，可以得到异想不到的反馈信息，他们会指出专业级玩家可能会忽略或者放过的重要问题。这些测试人员可以是任何人：修理咖啡机的修理工、邻居、开发组成员的父母，或者就是在街上走着的人。只要他们可以诚实地提出他们对游戏的意见，那么任何人的观点都是有价值的。在判断游戏界面是否令人感到混乱或者游戏是否太不可原谅的问题上，将第三类、第一印象测试人员和非玩家测试人员所提出的意见结合起来会非常有帮助。在如何对游戏进行改进方面，这些测试人员很少能提供建设性的反馈信息，但是他们会以独到的方式指出游戏中存在的重要问题。

23.1.2 不应该找哪种测试人员

有很多人或者团体不能担任游戏测试人员。有些人的观点被他们的个人动机所左右，或者有人不愿意提供真实客观的意见。虽然你可能会很不情愿听取这些人的反馈信息，但是能够了解他们言词之后的动机却非常重要，这样的话，你就可以正确地采纳他们的建议。

在这些不适合作为游戏测试人员的人中，首先要提到的是你的老板。在游戏设计者与测试人员的关系中，其中最关键的部分就是设计人员可以从测试人员那里得到反馈信息，然后应用到设计人员认为合适的地方去，而不是听从测试人员的指示。当遇到某个问题时，测试人员通常并不能非常好地理解游戏，也不能提出最好的解决方案，如果你的老板是发现问题的人，他很有可能会强加一个解决方案给你，而这种方案可能并不是最好的。有些老板很明智，他们懂得这个道理，他们了解作为游戏设计人员的你，知道应该以怎样的方式更好地解决问题。尽管如此，那些在支票上签字的人所提出的建议，与较低管理职位上的人提出的建议仍旧是不同的。

第二类不适合作为游戏测试人员的人是市场部门的人员。当看到你的游戏时，市场部门的人会有过多互相矛盾的想法，他们不可能告诉你真正的想法。相反，他们会告诉你什么才是“目标市场群体”的需要。当我反复在本书中提到这个问题时，很难想象到除我自己之外的读者会不会喜欢这本书，而市场部门的人正是这样想的。不要听信他们的预言，如果本来游戏的设置是正确的，而他们却说是错误的，这样就会搞乱你的游戏。

第三类不能作为游戏测试人员的人是那些与你的个人关系过近的人、非常要好的朋友、你的家人，或者是其他什么很重要的人。当这些人玩你的游戏时，虽然他们声称自己是很客观的，但是真正的意图是要增进与你的个人关系，因此他们不太愿意苛刻地批评你的游戏。有些朋友可能会理解与你增进个人友谊的最佳方式就是告诉你事实，但是许多人都会对他们的意见加以掩饰，试图让你更喜欢他们。实际上，有许多的作家都让他们的配偶作为第一个读者，并且得到了许多有价值的评价，如果你可以与这些亲近的人发展出这样一种诚实的关系，那真是件非常美妙的事情。但是现实情况是，现在的许多关系并不是那么诚实。

第四类不适合作为测试人员的人是傻瓜。傻瓜们总是会说出愚蠢的话，提出一些愚蠢的意见，因此对你不会有多少帮助。最好能发现这种人，然后离他们远一些，如果必须与他们共同工作的话，就要学会忽略他们说过的任何话。当然，我的说法显得有点夸张；测试人员中的大多数人当然不会是傻瓜。但是你偶尔会遇到这样的测试人员，对于他的意见最好是完全忽略掉。

第五类不适合做测试人员的人是那些认为自己可以为你设计游戏的人。这些测试人员可能会提出某些有用的建议，但大多数情况下，他们希望你改变游戏的某个方面，这并不是因为





这些方面出现了错误，而是因为他们要以完全不同的方式来实现它。真正优秀的测试人员会意识到你制作的项目运用了艺术的力量，游戏可以反映出你的个人偏好。这些优秀的测试人员会提出改进游戏的方法，而不是提出改变游戏的方案。

第六种需要注意的人是那些执着的游戏爱好者，尤其是那些盲信你的游戏风格，或者执迷于前一个游戏版本的人。这些测试人员将你的游戏与其他风格游戏的不同视为严重的设计错误，因此他们会否定你在新游戏中加入的富有创造性的内容。自己的游戏有忠实的拥护者固然重要，但是如果遵从他们的所有意见，有可能会使你制作的游戏与以前的游戏没有什么本质上的区别。



23.2 何时进行测试

那么，应该何时开始对游戏进行测试呢？正如我在本章的前面所提到过的那样，从游戏可以运行时开始，一直到最终发行的过程中，游戏测试可以说是游戏开发过程中的一个重要环节。也就是说，特定类型的测试要在指定的时间进行，如果在其他时间段上进行测试的话，那么测试就可能没有什么用处，甚至是没有意义的。了解何时需要某种类型的测试人员，这样不会浪费他们的时间。

当然，开发队伍应该尽可能地在游戏开发的所有阶段都玩游戏。正如我说过的，这样做很重要，可以让开发人员保持对开发项目的兴趣，并且可以让他们完成最出色的工作。假设项目并没有分解开，开发人员能够很清楚地了解自己如何对项目做出贡献，并且了解项目的最终结果，他就会更好地了解各种信息，而且有动力尽可能更好地完成他的工作。

早期的游戏测试最好由那些在游戏开发过程中经验比较多的人来完成，这些人都是你比较熟悉的，并且你会非常尊重他们的意见。早期的游戏测试需要测试人员忽略许多问题：游戏会经常崩溃、所有的美工都是要替换的部分、游戏的情节都是不完整的、只有一个关卡可以玩等等。许多人在得到这样一个游戏的时候，都无法忽视这些极端的缺陷。例如传统的测试人员，即使你告诉他们要忽略游戏中需要修正的大部分情节，他们也还是无法越过这些问题。另一方面，如果有个朋友来玩这个游戏，而他自己也是游戏设计人员，他就可以越过当前存在的缺陷，判断游戏是否实现了方案中设计的内容。这些设计者曾经制作过的项目也经历过与你的项目相同的状态，他们理解为什么项目中并不是所有的方面都可以正常运行。这些有经验的专家比其他任何人都会更好地发现并解释你的游戏设计方案中所存在的问题。

如果能找到一些你可以信任的人，并且能够在游戏开发的各个阶段让他们来玩你的游戏，那么这种做法就会比较有意义。这些人可以是游戏设计人员，或者是了解游戏开发过程的朋友，他们可以提供有用的反馈信息。在开发项目的过程中，你可能需要经常向这些你信任的人展示自己的游戏，这样他们就可以看到游戏的进展情况，并且向你提出自己的意见，告诉你他们是否喜欢这个游戏，还有他们是否认为游戏在最佳的方向上发展。因为这些测试人员将会在整个项目开发的过程中与你共事，所以他们可以更好地了解游戏，也可以了解游戏发展到目前状态的原因。

当实现 GUI 和控制部分时，让一些第一印象测试人员来试验这些新的控制会比较好。设置一个简单的测试关卡、区域或者玩家可以使用游戏控制和 GUI 的模拟部分，然后看这些测试人





员的进展情况。这样做比较有效，因为大多数界面和控制设计的最重要方面就是系统要尽可能地直观，并且判断这一点的最佳方式就是让第一印象玩家来测试。判断 I/O 系统是否直观并不会占用很长时间，因为如果玩家不能立即得出结论的话，你就知道游戏仍需要改进。

当游戏接近完成的时候，也就是当游戏的大多数特性都已经完成，而且游戏的大部分都可以玩的时候，最好让传统的测试人员进行测试工作。这个阶段一般称为“alpha”测试，而这个定义在各个公司之间会有所不同。当他们首先开始测试的时候，传统的测试人员将会发现代码中好像有数不尽的 bug，他们会尝试所有开发人员不曾设想过的手段来进行测试，但是你应该告诉他们略过这些 bug，并向你提供关于游戏设置的反馈信息。当然，在早期阶段获得反馈信息要比在“beta”阶段好得多，如果项目的时间计划比较紧迫，那么就可以在改进游戏的工作上花费较少的时间，而把更多的精力放在如何完成项目上。从某种意义上讲，要阻止对游戏的设置进行重要的修改，以免在某些重要的方面对游戏造成影响。因此，当有大量时间对所有的 bug 进行跟踪时，需要进行大规模的修改。

对于那些时间进度要求比较严格，而且“必须在圣诞节前交货”的项目来说，管理人员可能会考虑是否需要及时地使用测试人员来加速游戏的开发速度，有时甚至在游戏尚未进入 alpha 测试阶段的很长时间就开始测试。他们错误地认为，以这种方式，一旦游戏进入 beta 测试阶段，游戏中就已经消除了大多数 bug，并且可以立即交付。当然，他们没有了解到的是，在游戏完成所有特性之前，从编写代码的角度来讲，游戏可能还要经过重大的修改。当在比较重要的方面对代码进行修改之后，原有的 bug 完全消除了，但是又出现了新的 bug。如果测试人员指出了原来代码中的 bug，并且程序员必须花费时间去进行修改，那么这样做无非是在浪费时间，因为在大部分代码重新编写之后，那些 bug 都已经完全消除了，并且游戏中将会留下重新编写代码之后所产生的新 bug。

从某种程度上讲，这种情况在游戏的设置上也会发生。当游戏的大部分没有做好的时候，测试人员所报告的“第 10、12 和第 17 关没有和玩家战斗的敌人，所以并不是很好玩”的情况就没什么用处。让设计人员去检查这些无意义的 bug 所浪费的时间将会比节省下来的时间多得多。只有在游戏的确进入了适合于进行测试的状态之后，让传统的测试人员进行测试工作才有意义。因此，过早地让测试人员开始测试只会拖延游戏的进度。



23.3 如何进行测试

如何让测试人员进行测试的问题，与让谁来完成测试工作和何时开始测试的问题同样重要。在关于如何与测试人员打交道的问题上，游戏设计人员经常会犯一些重大的错误，以至于降低了测试人员的工作效率。这些问题都是很容易避免的，只要设计人员采用适当的方式与测试人员交流，并且知道该说什么不该说什么即可。

与游戏测试人员交流的重要部分实际上是在大多数时间里看他们玩游戏，而不是告诉他们如何玩游戏。要让他们以自己的方式玩游戏，并且看他们如何进展。想要纠正测试人员玩法的诱惑很大，而且也很难抵制。在传统的测试人员开始玩游戏之前，设计人员已经玩过了很长时间的的游戏，因此他会玩家在某种情况下所做出的反应非常熟悉，而且对于要如何玩游戏也很





了解。当第一次在测试人员的身后看他玩游戏时，设计人员有可能会说：“下一步到那里，”或者“你得用机枪按钮才行，”或者“为什么不试试关闭电源呢？”看着别人笨手笨脚地玩自己所熟悉的游戏，设计人员很快就会变成一个指导教师。

但是，游戏测试的重要问题就在于，要看玩家在没有游戏设计人员指导每个动作的情况下如何来玩游戏。当然，设计人员不能放进随游戏发行的盒子里，也不能从 Internet 下载。在学习如何控制游戏的过程中会遇到一些困难，而游戏测试的最好方好就是让测试人员自己去摸索。如果玩家遇到了障碍，或者好像总是不能掌握游戏的控制方式，设计人员就需要自问：是什么原因造成了这些问题。是游戏太难还是太混乱？怎样才能使游戏更简单，以便玩家可以更有机会去了解其中的问题，并且学会如何玩游戏呢？从游戏测试的过程中，设计人员可以得到许多教训，但是如果设计人员在测试人员玩游戏的每一步中都对其进行更正，那么他们就不会从中吸取任何教训。

当看着测试人员玩游戏的时候，设计人员应该观察他们玩游戏的方式。玩家可能不会在特定的情况中运用设计人员已经想好的方法或者方案。然后设计人员就会问，游戏会支持测试人员要做的事情吗，如果不支持的话，能不能让它支持，或者是不是应该支持呢？在测试阶段中，设计人员可以向游戏中添加许多内容，使游戏可以接受多种玩法。从这种意义上来讲，玩游戏的人已经限制在开发组，并且限制在初级测试人员的水平上。既然有各种各样的人玩过这个游戏，设计人员就可以观察到很多玩法，而且比他能想象到的要多。在测试阶段中，设计人员可以让游戏接受这些玩法，使玩家可以真正地以自己的方式来玩游戏。

当然，设计人员无法亲临游戏的所有测试过程，也无法了解游戏是否可以进行完全的测试，并在合理的时间内发行。通常，设计人员需要测试人员的报告来获得他们玩游戏的体验。虽然这些信息不如亲自观看测试人员玩游戏所能获得的信息那样有用，但是这些信息也会有很大帮助。当获得这些反馈信息时，倾听测试人员的意见是很关键的。这个过程看起来很简单，但是许多设计人员都会忽略他们在游戏中获得的反馈信息。大多数游戏测试工作，尤其是那些传统测试人员所进行的测试工作，在开发的过程中，都要在项目中大量的工作已经完成之后进行。此时，设计人员非常相信自己的游戏会像预期那样运行。因此，设计人员很难接受测试人员的反对意见，而这些测试人员可能会指出设计人员在数月的开发过程中一直忽略的重要问题。

设计人员的第一种托词一般就是声称测试人员不了解自己的想法。他们还会想出各种说法，说测试人员太蠢，不适合于游戏的测试工作，或者说测试人员并不是游戏所要面向的玩家，或者说测试人员只是为了抱怨而抱怨。测试人员在一般情况下提出的对游戏进行修改的建议最好不要去管，如果在十个测试人员中只有一个人提出了游戏需要修改的建议，那么这就有可能是那个人的个人偏见了。但是当设计人员从很多测试人员那里听到的都是相同的抱怨，他就应意识到游戏中可能的确出现了问题，需要他去注意。设计人员一定不能忽视测试人员的抱怨，并且诚实地对待每一个抱怨的问题，以判断这个问题是否有价值去解决。令人惊讶的是，很多设计人员对于测试人员的部分或全部建议都不加理睬。而在游戏最终发行的时候，这些游戏设计人员就会后悔自己的固执，让玩家和出版社的人提出了当初测试人员所抱怨的问题。当然，一旦游戏发行之后，解决这些问题就为时过晚了。



23.4 有指导的测试和无指导的测试

游戏中的测试工作可以分为两类：有指导的测试和无指导的测试。有指导的测试通常在项目开发的早期进行，那时的游戏还没有完全成型。在那个阶段中，设计人员清楚地了解游戏有哪些部分是不完整的，但是需要从他认为运行得相当好的部分中获得反馈信息。然后设计人员可能会指示测试人员去测试某个关卡或者游戏的某个部分。有指导性的测试也可以在项目开发的后期进行，那时整个游戏已经成型，但是某个部分可能刚刚修改过或者返工过。此时，设计人员可能正需要那一部分的反馈信息，以判断所做的修改是否解决了现有的问题，或者是否会在某个重要方面对游戏造成影响。

允许并鼓励测试人员进行无指导性的测试也是很重要的。将游戏交给他们，告诉他们开始玩，观察他们会做些什么，并且倾听他们的反馈信息。许多设计人员都会错误地仅使用有指导性的测试方法，并让测试人员仅在他们工作的系统中玩游戏。当测试人员对于游戏的某些部分提出意见时，设计人员就会抱怨他现在对那些内容并不感兴趣，或者游戏中存在问题的部分已经“完成”了。有指导性的测试固然有其作用，但是如果所有的设计人员都这样做的话，就有可能错过游戏中重大的问题，而他自己可能并不会意识到出现了问题。无指导性的测试可以向设计人员提供关于游戏全面的反馈信息，这些信息对于解决所有的问题至关重要。

当然，即使当你指示测试人员只对游戏的某一部分进行测试时，他们也会指出所遇到的其他问题。这时就需要一个极其服从命令的测试人员来测试设计人员要求测试的系统。在你当前没有制作的游戏部分获得反馈信息可能比较令人沮丧，但是从长远的角度来看，这样做非常有帮助。当测试人员向你提出关于如何改进游戏的无关建议时，即使你不想立即解决这些问题，也要记下这些问题，以后要重新审视。没有什么会比在游戏发行之后才意识到问题更加令人沮丧的了，所以最好在测试人员报告问题的时候就意识到这一点，这样就有足够的时间来解决问题。



23.5 谐调

谐调游戏惟一的机会可能就是游戏的大部分工作完成的时候。提前对游戏进行谐调，也就是在所有的游戏控制和关卡都制作完成之前，这种谐调工作只能认为是初步的谐调工作。直到游戏的内容完成之前，你无法真正了解整个游戏如何才能正常运行，难度如何才能随着游戏的过程而升级。你可以将自己的游戏视为组合成一个大系统的不同系统的集合。对于基于关卡的游戏来说，每个关卡都可以当做是一个系统。而在每个关卡中，玩家所遇到的每一次战斗或者谜题都可以当做是一个系统。为了能让游戏谐调运行，所有的这些系统都必须就位，因为在修改了一个系统之后，只有相应修改其他系统才能达到我们所寻求的整体平衡。

游戏在整体上完成并且达到真正谐调的时候，也就是游戏进入全面测试阶段的时候。这是一种最好的安排，因为游戏的谐调工作和测试工作是紧密关联的。谐调工作通常包括修改游戏中的某个设置，然后继续玩游戏，以判断那些修改是否达到了你所要求的效果。在遇到每一个谐调的问题时，你和游戏测试人员都应该尝试去玩游戏。然后测试人员可以就你对游戏所做出





谐调努力的效果提出意见，与你自己对游戏状况的分析结合起来，可以做出更多的修改，并且可以再次重复整个过程。很少有人在没有其他测试人员意见的前提下就能自己成功地完成游戏的谐调工作。想自己对游戏进行谐调的设计者可能会成功地为他们自己完成这个工作，但是通常就会导致游戏难度过大。

对游戏进行谐调的最佳方式就是将不同的系统分为若干组，这样调整起来就很容易。例如，假设你正在制作一款战斗动作类游戏。如果玩家在游戏中使用了棒球棍，这根棒球棍上有很多不同的属性，比如它的破坏力、攻击速度、在损坏之前可以使用的次数、它的价格、需要几只手来握住它等等。同样，敌人、玩家和其他的系统属性也可以分为不同的类别，然后根据不同的用途和需要进行调整。要不断地调整和修改这些属性，以便达到你所要求的效果。

当进行谐调工作时，必须密切注意不同的值之间是如何相互作用的。你可能需要对某种武器进行修改，使一种战斗形势比较刺激，但是在游戏中的另一个位置却是无法战胜的。游戏越复杂，那么在系统中可能会忽视的各种修改所造成的影响就越多。当进行谐调工作时，必须认为游戏中做出修改的每个部分都会相互影响，并且必须保证这些修改不会影响游戏的整体效果。保证自己没有破坏游戏整体效果的惟一方式就是，对游戏进行全面测试。因此，在临近交付日期时做出重要的修改是很伤脑筋的事情。一旦自己所做出的修改对系统造成了影响，而在交付给制造商之前又没有人发现的话，情况将会怎样？

当然，在我以上介绍过的谐调方法中，游戏的设计人员必须可以访问那些影响游戏不同实体行为的数据，并且可以修改这些数据。这就意味着，编写代码要采用可以很容易修改这些信息的方式。这最后的一点看起来很简单，但是我见过许多游戏引擎在修改诸如武器参数的信息时并不容易，甚至根本就不可能修改。从游戏开发一开始的时候，程序员就必须牢记在项目结束的时候设计人员将会如何对游戏进行谐调。如果他们反而遗忘了代码中的某些“魔数”，游戏就会被“限制”在某个特定的状态，使得游戏的谐调工作无法进行。虽然谐调工作只能在游戏的大部分工作完成之后进行，但开发人员也要从项目一开始就为谐调工作做好准备，否则就无法做到有效的谐调。如果设计人员有机会较好地谐调游戏，这种谐调信息必须摆脱配置文件、关卡编辑工具或者其他设计人员可以使用的格式的限制。

23.5.1 游戏难度过高

当对游戏进行谐调时，要一直记住一条经验：你的游戏太难了。无论你正在制作的游戏属于什么类型，或者开发组的成员有多么聪明，在游戏接近完成并进入测试阶段的时候，游戏都会显得难度过大。这通常是因为，在这个时候，一直只有开发人员在玩游戏。开发人员已经在项目上工作了9至18个月，在这段时间里，他们已经积攒了一些玩游戏的技巧，并且对这个游戏非常在行，其水平可能比将玩游戏的玩家的水平高出90%。为了使游戏可以吸引他们自己，开发人员在游戏中加入了具有挑战性的内容来玩，这对于90%的玩家都太难了。

测试人员所提出的第一个意见经常就是：“这个游戏太难了。”正如我以上说过的那样，你的第一反应就是不去理会这个抱怨，并认为这是因为他们的技巧不够娴熟，或者在这个游戏上没有经验。“他们会表现得更好”，你会这样说。但遗憾的是，这的确是事实。如果游戏要经历三个月的测试，那么测试人员也会像开发人员那样玩得熟练。那时候他们同样也不会认为





游戏的难度过高。完全有可能，由开发人员制作出来的游戏，包括测试人员在内都不会了解游戏到底有多难。

作为设计人员，你必须非常认真地了解游戏的实际难度，并且在和游戏进行谐调期间，让游戏的难度可以使初玩游戏的人有机会取得胜利。还要记得测试人员的第一印象，并且问自己是否解决了他们发现的问题。如果有必要的话，应该另找一些第一印象测试人员，让他们来判断游戏是否还是太难。

遗憾的是，仅通过谐调过程，有时可能无法使游戏的难度降低。在你所制作的游戏方案中，可能在比较简单的关卡中难度就过高。如果你真的希望自己的游戏可以让初玩者很容易地投入，那么从游戏设计的一开始就应注意这个问题。我所制作的项目 *Centipede 3D* 就是一个好例子，游戏的难度可以完全超出开发人员的想象。我曾尝试过降低游戏的难度，但是游戏的内部设计就是要达到原创街机游戏的难度。因此，*Centipede 3D* 竭尽全力缩短玩家玩游戏的过程，并且具有很快的速度。遗憾的是，家庭游戏的玩家希望他们的游戏可以持续长一点的时间，至少要比花 25 美分在街机上玩的时间长。就发行版本的游戏难度来说，很难想象在进入谐调阶段之前，它的难度要容易 10 倍。

当设计师 Jason Jones 对 *Marathon* 游戏进行谐调时，使用了一种有趣的技术，可以保证游戏不会太难。如果他和开发组的另一个成员仅使用游戏中的“拳头”武器就可以以最难的设置打通关卡的话，他就得以得出游戏对于其他玩家也具有合理挑战性的结论。当然，其他玩家可以获得威力更加强大的武器，使用起来也比拳头要容易，他们也不必使用最难的设置来玩游戏。Jones 为难自己是为了判断游戏对于普通的玩家来说到底有多难。类似于这样的技术使用起来很方便。如果设计人员可以在不使用武器的情况下赢得胜利，那么其他玩家也可以有相当大的机会使用武器来取得胜利。

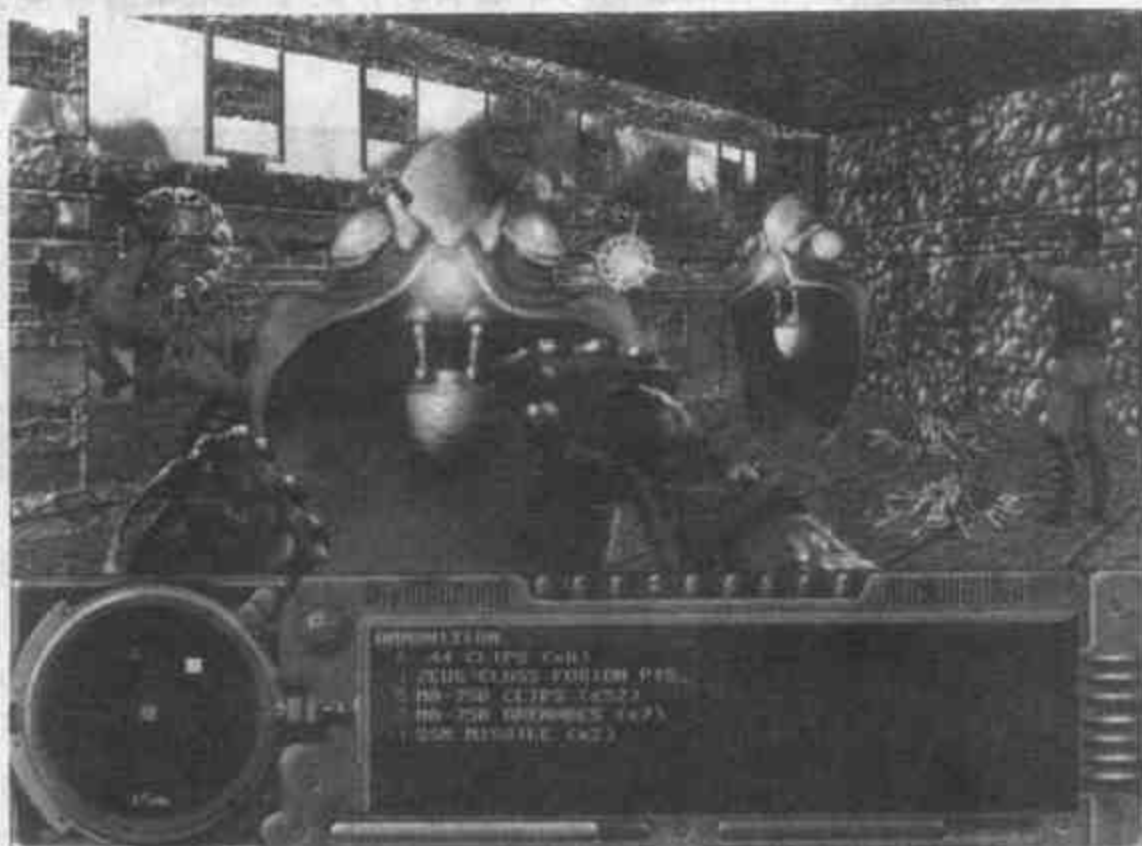


图 23.2 在测试 *Marathon* 游戏难度的时候，开发人员必须使用最难的设置，并仅使用最弱的武器——拳头 此图为 *Marathon 2* 中的场景

最后要说的是，比起其他的工作，对游戏的谐调工作更是一种“本能的感觉”。虽然你可





能总是会假设游戏的难度过高，但在对游戏进行谐调的过程中，却没有其他的规则可以遵循。你必须能够全面地观察游戏，以理解比你玩游戏的经验少得多的玩家如何来玩这个游戏，并且还要了解哪些地方可以对他们提出挑战，而又不会不公平或者过于残酷。了解如何在游戏中进行谐调是一种随着经验增长的技巧，这种经验既包括玩其他游戏的经验，又包括玩自己游戏的经验。为了能够在谐调工作中得心应手，必须尽可能地积累这两方面的经验。



23.6 艺术观点

我曾在本书中多次提到过 focus group 会起到很多消极作用。理解游戏测试与 focus group 之间的区别至关重要。focus group 通常是指一些“站在街边”的人，他们会用一两个小时的时间来表达自己的观点，而这些观点要涉及一系列的游戏。他们不能去玩游戏，因为游戏还没有开发出来。他们根据描述了解到游戏的概念，并被人询问是否会愿意买这样的游戏。而另一方面，游戏测试人员是开发组成员认识的或者至少有机会认识的人。了解一个人在何种程度上采纳他们的意见是很重要的。此外，游戏测试人员会去玩这些要测试的游戏，而 focus group 的成员通常不会。正是因为这些关键的区别，focus group 倾向于反对创作原创游戏和有创造性的游戏，而鼓励开发稳妥的、没有什么创新的游戏。因此可以想象到，focus group 只喜欢 Pac-Man、Tetris 或者 Civilization 这样的游戏。从第 22 章与 Will Wright 的谈话中我们可以了解到，支持 The Sims 的 focus group 如此之少，以至游戏计划几乎都要被取消了。应该说，focus group 是由市场部门的人员组成的，而游戏测试的任务是由开发人员来完成的。第一组的主要兴趣是最简单的方式为公司赚钱，而开发人员的兴趣在于能制作出吸引人并且刺激的游戏。当然，两者的动机可以相似，但是当游戏主要考虑前者而没有考虑后者时，这个游戏最终可能会两个目的都无法达到。

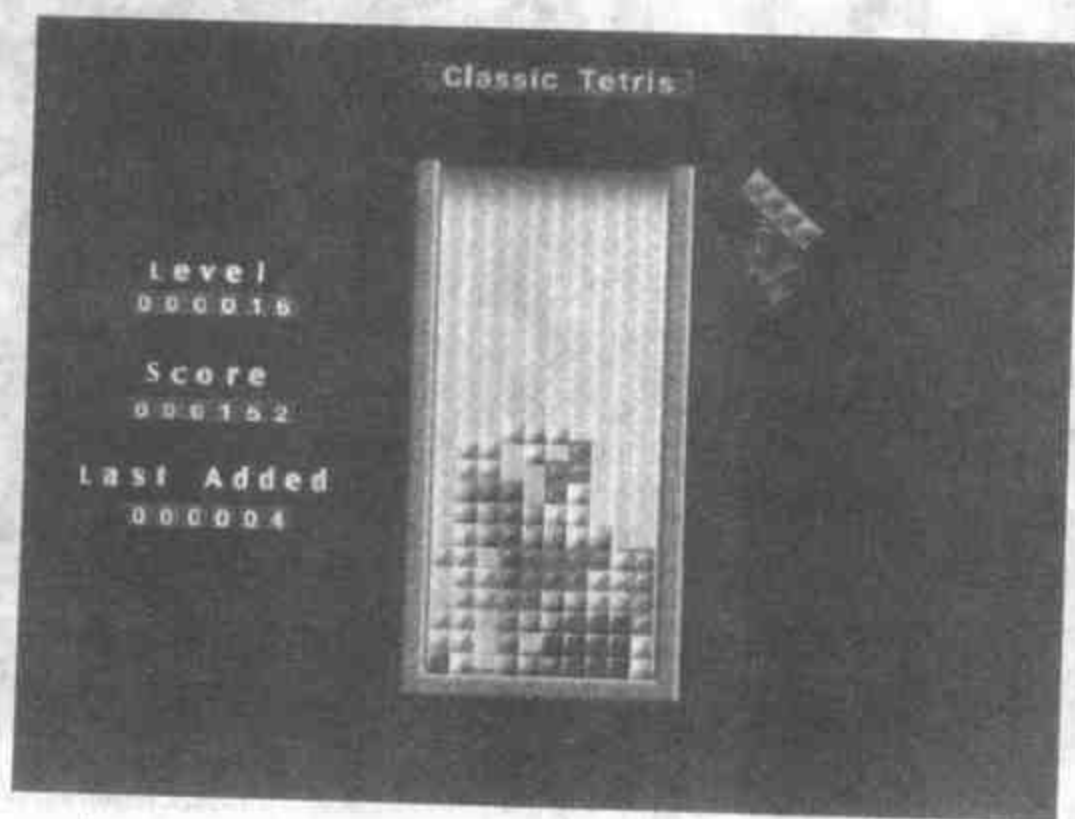


图 23.3 当 Tetris 发行时，它是一个非常独特的游戏。可能是早期支持这个游戏的 focus group 都已经消失了。此图为 The Next Tetris 游戏中的经典模式



当进行测试的时候，一定要知道自己无法取悦每一个人。在一个足够庞大的测试队伍中，一定会有人不喜欢游戏的某些部分，甚至也会有人不喜欢整个游戏。如果你想让测试队伍中的每个人都高兴的话，结果有可能会使游戏对其他人来说缺乏乐趣。你制作的某个游戏可能有很多人非常喜欢，而有一些人认为枯燥，如果你想取悦所有的人，最终结果可能会使每个人都认为很好，但是可能没有人对这个游戏还怀有真正的热情。假设提出某种选择，我总是倾向于让一些人可以得到他们所钟爱的体验，而不会选择让每个人得到他们只有一点兴趣的游戏。

测试也不意味着应该由委员会进行游戏设计。你不需要采纳开发人员提出的每个意见并加以实施。有些想法可能是相当合理的，但你可能认为这些想法并不适合于加入到游戏中。这也是一种相当合理的反应。最终，每个游戏测试人员都有可能告诉你游戏的某个部分需要修改；但是，作为一个艺术家，如果从直觉中判断出并不需要修改游戏的那个部分，那么就把它放在那里好了。你必须是在最终在游戏中做出决定的人。至于委员会，无论其中包括执行官、测试人员还是包括开发队伍中的成员，不会有一个具有统一的观点和确定的目的。



总 结

正如我在前言中说过的，本书并不是电脑游戏设计的权威指导。没有哪一本书可以成为权威。除了共享六位大师级游戏设计者的思想，本书还致力于向读者传授我所了解的关于游戏设计的知识。当然，如果读者不能正确理解书中所传授的内容，那么本书中的所有信息都毫无价值。与其他任何艺术形式一样，电脑游戏需要其制作人员全身心地投入到游戏的创作中，这样的游戏才值得一玩。我觉得电脑游戏的巨大力量可以影响它的玩家，所以游戏设计人员有很大的责任合理地利用这种力量。



艺术

游戏开发行业一直对于“电脑游戏是否是一种艺术形式”的问题纠缠不清。另外一些讨论则围绕在电脑游戏是否曾经是“正统的”艺术上。这样的讨论是毫无结果的。我们不能只是吹响号角并且鼓吹自己的成就而让公众认为我们从事的是正统的事业。有些人并没有把电影、爵士乐或者喜剧书当做是“正统的”艺术，这些艺术都有具体的形式，而且由于它们存在的历史较长，使得电脑游戏与它们相比真是相形见绌。有人肯定会问：“无论电脑游戏是不是一种艺术，你们能够做出什么不同的东西出来吗？”无疑，如果要想让公众相信我们从事的是正统的事业，那么最好的方式就是让电脑游戏和其他的媒介一样引人注目。

电脑游戏当然是一种艺术。难道这还不够明显吗？如果有人引用了我所喜欢的关于艺术的定义，那么情况就更是如此，这个定义来自 Scott McCloud 的“Understanding Comics”一书：“艺术，在我看来就是所有没能超越我们人类两大本能的任何活动：生存与繁殖。”许多一直在叫喊“游戏是门艺术”的游戏开发人员都有一种不安全的复杂想法，并感觉需要向家人、朋友、公众甚至他们自己证明制作游戏的工作是很正确的。这种不安全感几乎不会使艺术家完全发挥他的能力，因为他一直在设法证明他自己。这样也无法制作出好的作品；而更有可能制作出一堆自己认为满意的垃圾。当有人问 Alfred Hitchcock 是否同意有些评论家说他的电影是艺术这种说法时，他回答道：“他们这么说我非常高兴，但是这并不等同于拿出剧本的一页然后说，‘我正在完成一部艺术作品。’这是很荒唐可笑的——不能这样做。”只有当开发人员的真正动机是想为玩家制作出最好的游戏时，才有可能制作出合格的游戏来。



媒体

一直以来，我们这些游戏开发群体很羡慕其他媒体。这可能是由于游戏设计人员希望得到



其他媒体在社会上得到的尊重，也就是我所说的“正统”。其他人可能会偷偷地、下意识地甚至完全公开地表示他们希望自己的工作不是在制作游戏。某个游戏设计人员可能会说：“我想让自己的游戏达到与电影 *The Godfather* 类似的效果！”或者说“我想让人们喜欢玩我的游戏，就像是喜欢听 *Jimi Hendrix Experience* 的 *Electric Ladyland* 一样！”但是这种方法并不正确。我们的这种媒体所具有的优势在于它与其他的媒体不同，它在人们中间所能激发的情感是其他的艺术形式所无法做到的。如果我们无休止地模仿其他媒体，我们就会永远从事二流的、模仿性的工作。可以肯定的是，*Jimi Hendrix* 并没有在制作 *Electric Ladyland* 的时候去模仿他所看过的电影。同样，*Francis Ford Coppola* 知道必须对 *Mario Puzo* 的“*The Godfather*”一书做出重大修改，才能根据它拍出好电影。*Coppola* 依靠对电影技术的掌握，使他拍出的电影可以比电影情节所依据的书要精彩得多。虽然电影和书的故事、人物及对话几乎都是相同的，但 *Coppola* 以电影手段所讲述的故事几乎在所有的方面都比 *Puzo* 用文学手段表达得更好。虽然游戏对玩家的影响与书对读者的影响、电影对观众的影响、歌曲对听歌的人所产生的影响不同，但至少也不会有什么更坏的影响，仅仅是不同而已。电脑游戏有自己的长处，如果我们要制作出最好的作品，就必须了解这些优点。无疑，我们的媒体会对选择从事这个工作的人提出挑战，而这种挑战在其他的艺术形式中却无法找到；如果我们希望自己不仅仅是在吹牛的话，就有责任去面对这些挑战。

Marshall McLuhan 在他的“*Understanding Media*”一书中说过这样一句名言：“……媒体就是消息。这仅仅是说任何媒体的个人和社会性的结果——也就是说，我们自己的任何进步——都是因为我们的每一次进步或者任何新技术所引发的事情中所产生的新的衡量标准。”*McLuhan* 提到，当人们关注于电视节目的内容或者戏剧和音乐时，媒体的真正信息并不是来自其内容，而是来自媒体本身。现在，我当然不能声称自己是 *McLuhan* 式的学者，但我不得不设想一下我们这种电脑游戏的媒体具有什么特点；当 *McLuhan* 写这些话的时候，这种媒体还没有出现。电脑游戏这种内在的交互式的特点形成了一种庞大的媒体，鼓励玩家主动参与到其中来，而其他的媒体却不能。我不得不得出这样的结论，我们的媒体所提供的最基本的信息就是参与和给予。

在多人游戏中，游戏设计人员能够制作与其他人可以方便地进行交互的游戏，也可以在单人游戏中让一个人只与电脑进行交互。在后面这种情况中，如果说真正的交互是在人与电脑之间进行的，那么这种说法多少有些不太正确，因为电脑只不过是一种用于交互的媒体；交互实际上是在玩家与游戏制作者之间发生的。当我花费数周的时间在我父母房子后面漆黑的电脑室里玩 *The Bard's Tale* 和 *The Bard's Tale II* 的时候，我从来都不认为自己是孤独的。从某种角度来讲，我是与游戏的制作人 *Michael Cranford* 在一起，在他所创造的世界中游玩，在游戏中进行探索。这种媒体的力量如此强大，所以我很快就开始这方面的工作，制作自己的游戏，这样我就可以将我自己的一部分经历放在游戏中，让玩家去体验。



动机

我曾在本书中详细地介绍了玩家玩游戏的原因，但是最重要的问题是，作为一个游戏设计人员，你应该问的是自己制作这些游戏的原因。电影导演 *Krzysztof Kieslowski* 曾说过，如果有哪个艺术家不能理解他自己、他的生活，还有他所做任何事情的原因，那么这个艺术家就无法





理解他的作品。当你开始从事游戏设计行业时，质问自己从事这项工作的动机对于有效地使用这种媒体是非常重要的。

设计人员问自己的第一个问题就是自己从事电脑游戏设计的原因。这是偶然事件吗？是在商业界的朋友偶然知道一个职位有空缺？他是不是无目的地在这个领域中寻找，只为了找到一个关于游戏开发的广告，然后说：“嘿，这个工作好玩吗？”他是不是把游戏开发看做是很好玩的事情，觉得自己比那些送报纸为生的朋友更熟悉内情？他是不是想从事其他领域的工作。比如电影或者电视；当他发现自己从事的职业并不如想象的那么好，而从事游戏行业就是为了能付得起所有的账单，使生活变得更好？或者是不是因为游戏制作是一种可以让他提高技能，并且可以赚很多钱的职业？

作为读者，你可能会想，以上这些原因都不是从事游戏制作工作的最好动机。一些人从事游戏制作工作有很纯的动机，他们在这种工作中有一定的追求，因为他们非常愿意做这样的事情。当然，设计人员可能会带有错误的动机进入游戏开发行业，只是为了找到自己为制作游戏所激发出来的热情。无论他因为什么原因而开始从事游戏开发的工作，最重要的是他现在正在开发游戏，他想制作出最好的游戏。

我仍旧感到很惊讶，并且由于许多人出于错误的原因从事游戏制作工作而感到失望：因为这是很好玩的工作，因为薪水高，因为他们没有更好的事情可做。游戏开发可能比较好玩、比较有个性，而且可能比其他的职业薪水更高，但其中也存在只有具有真正目标的设计人员才会得到的另一种好处：制作引人注意的交互式的体验。当其他的动机成为设计人员的主导思想时，他的工作就会毫无希望地妥协，使他无法发挥自己所有的潜力。

设计人员最有可能制作出相当优秀的游戏的情况就是他有自己的梦想。这个梦想就是要发展游戏的艺术，使其超越平庸和轻浮，而不会随着时间的流逝而不断地沉沦。这个梦想要创造一个精彩的游戏世界，使玩家在玩游戏的时候摆脱平常生活的方式。这个梦想要创造一种具有吸引力的作品，这种吸引力是其他媒体所不具有的。这个梦想要丰富玩家的生活，并向更好的方向发展。你有这样的梦想吗？



Appendix

附录 设计文件示例：“Atomic Sam”



以下是一款操作简便的动作游戏“Atomic Sam”的设计文件。这款游戏本身并无新意，从设计角度看，它的魅力部分来自它的简单性。它是我以前承担过的一个项目的组成部分，在该项目中，这款游戏始终未能设计完成。尽管如此，读者仍可相信这一文件的“可靠性”，因为在我已经完成的一些项目中，设计文件采用的风格和形式与它毫无二致。

由于“Atomic Sam”结构简单，因此它的设计文件并不太大。在其他一些项目中，我编写的设计文件往往是它的五倍，然而，在游戏行业中，即便这些文件如此小也并非无可比拟。当时，我有意使这一文件的部分内容保持简短，因为我并未打算编写一份完整的设计文件，我的意图是让读者对“Atomic Sam”有一个大致的了解。此外，我还特意精简了某些内容，例如，敌方机器人列表远远小于游戏中的实际敌人列表。同样，完整的设计文件应该对 Sam 扔出的投射物、操纵的计策和机关以及在游戏环境中遇到的各种角色进行更为全面的描述。因此，这款游戏还可进一步扩大，即在本文所描述的五个区域的基础上再增加一些区域。

事实上，整个文件还可以采用更多细节。从这一文件的编写方式可以看出，它的作者将参与整个开发过程，以便指导设计工作，使其保持正确的方向。正如我在本书其他章节所述，作



为一名游戏设计者，我希望参与的仅仅是那些我能彻底了解的项目。假如作者并不打算积极参与这一文件所涉及的项目，那就应该在文件中增添更多的细节，以便明白无误地说明项目应该保持的方向。

例如，关卡设计部分可能会得到非常具体的描述。不过，倘若关卡设计小组能透彻地理解游戏，并能承担滑稽关卡的设计任务，那么，设计文件所包含的内容就足以成为关卡设计的开端。借助这一文件，关卡设计者在制作关卡的方法上能获得极大的自由。如果关卡设计者有能力应付这一挑战，这种方法将起到有效作用。当然，如果打算独立设计许多关卡，就不必预先进行详细、周密的计划。许多成功的游戏均采用这种方法制作完成，包括我承担过的一系列项目。例如，在制作 Centipede 3D 的关卡之前，我仅仅对 AI、蘑菇类型以及 power-ups 做出一个大致的构想，结果表明，这是一种行之有效的方法。

当然，正如我在本书其他章节所述，在编写设计文件之前，设计者应该对游戏的主旨有一个明确的认识。例如，在着手为“Atomic Sam”编写设计文件之时，我首先拟定这款游戏的主旨说明：

“Atomic Sam”是一款非暴力、快节奏的动作游戏，游戏的主要任务是以新颖、独特的方式，使用各种投射物和环境计策，打败不同类型的罪恶机器人。故事的主人公是一位首次离开父母的小男孩，在一些良师益友的帮助下，带着种种好奇，开始了解自身所处的世界。“Atomic Sam”的故事发生在一种独特的“回溯未来世界”里，它以怪诞、荒谬的手法为故事和动作的展开营造出独一无二的背景氛围。

在主旨说明的引导下，游戏设计就能自然而然地过渡到下述设计方案。如前所述，设计文件并无一定之规。设计者的职责是以明确的表达方式尽可能详细地拟出设计方案，以便明确无误地将其传达给设计小组的所有成员。

“Atomic Sam”

设计文件

第 2.0 版

本文档和“Atomic Sam”的版权和注册商标由 Richard Rouse III 所有。

“Atomic Sam”中的角色由 Richard Rouse III 和 Steve Ogden 设计

文件目录

1 概述	389
2 游戏机制	390
2.1 概述	390
2.2 摄影机	391
2.3 插入式图形用户界面	391
2.4 重新开始和保存游戏	392
2.5 控制方法简介	392





2.6	运动方向	393
2.6.1	调整方向	393
2.6.2	改变运动速度	393
2.7	飞行运动	393
2.7.1	上下运动	393
2.7.2	停止运动	394
2.7.3	飞行速度	394
2.7.4	飞行方向	394
2.7.5	爆发速度	394
2.7.6	飞行时间限制	394
2.7.7	着陆	394
2.7.8	跌落	395
2.7.9	高度限制	395
2.7.10	火箭背囊升级	395
2.8	地形	395
2.9	捡起物品	396
2.10	扔出投射物	396
2.10.1	物品目录	396
2.10.2	收集投射物	397
2.10.3	准备投射	397
2.10.4	扔出投射物	397
2.10.5	投射速度与距离	397
2.10.6	打击能力	398
2.11	电能比拉鱼	398
2.12	动作	398
2.12.1	开关和按钮	398
2.12.2	推动与操纵	398
2.12.3	手动拾物	398
2.12.4	交谈	399
2.12.5	阅读	399
2.13	互动战斗环境	399
2.14	观察	400
2.15	朋友	400
2.16	对话	401
2.17	片头动画	402
2.18	游戏故事	402
2.18.1	环境	402
2.18.2	朋友	402





2.18.3	无线电	402
2.18.4	告示牌	403
2.19	关卡	403
2.19.1	关键路径	403
2.19.2	训练关卡	403
2.19.3	Electric Priestess	403
2.19.4	环境顺序	404
3	人工智能	404
3.1	敌人 AI	404
3.2	玩家探测能力	405
3.3	运动	405
3.4	飞行	405
3.5	目标寻找能力	405
3.6	打击承受能力	405
3.7	攻击能力	406
3.8	躲闪能力	406
3.9	特殊动作	406
3.9.1	绑架	406
3.9.2	内置维修臂	406
3.9.3	协作	406
3.10	奚落	407
3.11	圈套设置	407
3.12	非战斗主体	407
3.12.1	逃避	408
3.12.2	与 Sam 交谈并提供帮助	408
3.13	朋友	408
3.13.1	无敌	408
3.13.2	跟随 Sam	408
3.13.3	殿后	409
3.13.4	提供建议	409
3.13.5	故事叙述	409
4	游戏要素	409
4.1	物品	409
4.1.1	Sam 的投射物	409
4.1.2	火箭升级	410
4.1.3	其他装置	410





4.2 角色	411
4.2.1 Atomic Sam	411
4.2.2 朋友	412
4.2.3 其他角色	412
4.2.4 敌人	413
5 故事简介	418
6 游戏过程	419
6.1 故事背景	419
6.2 序幕	420
6.3 Gargantuopolis	420
6.4 Electric Priestess 的泡泡屋	421
6.5 Benthos	421
6.6 Harmony	422
6.7 New Boston	422
6.8 Electric Priestess 的泡泡屋	423
6.9 Ikairus	423
7 参考书目	424



1 概述

“Atomic Sam”是一款故事性很强的动作游戏。在这款游戏中，玩家操纵的是 Sam，一位失去父母的小男孩，Sam 必须在充满敌意的环境中奋力搏斗，打败那些试图阻止他找到父母下落的机器人。这款游戏需要敏捷的反应和灵活的计划，其怪诞的未来环境不仅能激发孩子们的兴趣，而且也将取悦所有爱好快节奏动作游戏的玩家们。本游戏适用于任何新型控制系统。

在“Atomic Sam”中，玩家的主要任务是操纵小 Sam 穿过各种游戏环境，并打败他所遇到的机器人。尽管本游戏以这类战斗场面为主体，但它始终保持非暴力色彩，即 Sam 并不会摧毁那些阻止他的机器人，他只是将他们制服而已。每次被打败时，Sam 总是受到惊吓或者落入圈套，但从不会真正死去。由于“Atomic Sam”具有这种怪诞和乐观的特性，因此它不能突出任何形式的血腥内容，暴力也应保持最低水准。

这款游戏将设置一些特定的场景，在这类场景中，玩家可以使用环境物品战胜那些追赶自己的机器人，从而充分发挥自己的创造性思维。鲁本·戈德堡式的机关随处可见，因此，Sam 可以采用一些千奇百怪的方式制服眼前的许多机器敌人。玩家必须领会到，在不同的场合中应该采取什么样的措施，这一点与快速反应和熟练操作同样重要。

“Atomic Sam”是一款易于掌握、操作方法简单且直观的游戏。游戏开头出现的插入式指导内容，可使毫无经验的新手们轻松地掌握游戏玩法。游戏中间包含三个部分，在每一部分中，





Sam 都会得到一些特殊朋友的陪伴，这些人将协助他打败眼前的敌人。与此同时，他们还会给 Sam 讲述有关这一未来世界的逸闻趣事。

“Atomic Sam” 的场景设置在未来世界，不过，它并非我们现在想象的那个未来世界，而是 20 世纪上半叶人们对未来做出的预言，在那个世界里，所有关于技术改变人类生活方式的乐观预言均已成为现实。原子能创造了一个幸福美好、无忧无虑的世界，机器人将听命于人们的每一个使唤，人类已经步入极乐世界。不过，值得注意的是，人类在 20 世纪后半叶取得的一些关键性成就在这个世界里毫无踪影。例如，喷气式飞机并未得到广泛应用，结果，市民们只得乘坐螺旋推进式飞机和齐柏林飞艇在规模庞大的都市之间往来穿梭。同样，人们对激光唱盘、微波技术、个人计算机或电子游戏等一无所知。

游戏故事以 Sam 放学回家后发现父母神秘失踪为起点。Sam 戴上父母送给他的火箭背囊，前往他们的办公室，希望能在那儿找到他们。途中，Sam 频频遭受一些机器人的威胁与袭击。抵达办公室后，Sam 并未发现父母的踪影，不过，他偶然遇到那位神秘的 Electric Priestess，她吩咐 Sam 分别去海底城市 Benthos、机器人城市 Harmony 和月球城市 New Boston 寻找自己的父母。一路上，Sam 不断搜集信息，并最终得知父母遭到 Max Zeffir 的绑架，因为他们知道一些令 Max Zeffir 担心的内情。Zeffir 是世界上最富有的人士之一，同时也是 Sam 父母的老板。于是，Sam 登上 Ikairus 号飞船——一艘由螺旋桨推进的巨型核动力飞行器，与 Zeffir 展开较量。最后，Sam 打败对手，与父母团聚。

很显然，由于故事的荒诞性和主人公充满朝气的形象，这款游戏的吸引对象主要应该是未成年的孩子们。当然，令家长们感到欣慰的是，这款游戏要求玩家逮住敌人而不是杀死他们，而且，当玩家在某个场合失利时，Sam 也总是以某种非致命方式丧失反击能力。不过，那种激烈、疯狂的游戏场面，独特、多变的环境以及造型新颖的敌人也会引发年轻人的兴趣。此外，由于“Atomic Sam”的 retro-futuristic 色彩及其对情节的强调，使它同样能吸引一些年龄更大的玩家，他们可能会深切地感受到 50 年以前人们所预言的未来与现实世界之间存在着的极大差别。



2 游戏机制



2.1 概述

“Atomic Sam” 是一款摄影机处于浮动状态的第三人称 3D 动作游戏，它沿袭的是《Super Mario 64》或《Spyro the Dragon》的传统。不过，“Atomic Sam” 也有其独到之处。比如，游戏侧重的并不是探险，而是使玩家在战斗中通过各道关卡，并设法避开那些试图挡住 Sam 去路的机器人和其他敌手。因此，它的游戏机制应该允许玩家直观、全面地控制游戏角色，欣赏并置身于那种妙趣横生的游戏世界。





附图 1

2.2 摄影机

在本游戏中，玩家将操纵 Atomic Sam 这一角色。无论何时，Sam 都出现在屏幕中心，摄影机在角色后上方“浮动”，采用的是“后窥”视角。摄影机与 Sam 保持一定的距离，便于玩家观察 Sam 和当前环境。由于摄影机相当“精明”，因而它不会穿透景物，玩家始终能看清 Sam 的活动。如有必要，在情况紧急时，摄影机可以靠近 Sam。如果 Sam 在屏幕上形象过大，妨碍玩家充分观察周围环境，他就会呈半透明状，以便玩家看清眼前的景物。这种半透明状态只对玩家显示，它对游戏环境或敌人看见 Sam 时的反应并无影响。

摄影机将尽量保持在 Sam 的背后，以便为玩家提供平稳的视觉经历。如果 Sam 匆忙转身，摄影机将慢慢调整方向，而不是陡然改变方位。如果玩家改变 Sam 的方向后很快又恢复到原来的方位，摄影机根本不会转向。因此，当玩家稍微调整 Sam 的方位时，摄影机不会剧烈晃动。

2.3 插入式图形用户界面

游戏环境和玩家角色 Atomic Sam 构成的画面占据了游戏屏幕的大部分面积，其中 Atomic Sam 大致位于屏幕中心。画面上方叠入其他一些图形，以便玩家了解 Sam 在游戏环境中的状态和行动。

- **当前投射物+数量：**画面左下角显示的图标表示 Sam 当前准备妥当的投射物，与之相邻的一系列“小人”或“标记”则表示在物品目录内这一投射物的数量。在后文中的投射物说明以及游戏要素部分可详细了解游戏中使用的各种投射物。





- **选择当前投射物：**按住“下一个投射物”按钮，玩家就能在屏幕上方看见一串水平显示的投射物，它们均出自 Sam 的物品清单。这时，玩家可以滚动这一列表，选择自己希望 Sam 采用的物品。这些武器均以图标形式出现。一旦玩家松开“下一个投射物”按钮，这一显示就会消失。
- **飞行时间：**Sam 的火箭背囊只提供有限的飞行时间。飞行状态由一道水平条纹表示，条纹紧靠屏幕右下角的火箭背囊图标。当 Sam 的火箭背囊装满燃料时，条纹就会完整地显示，但随着飞行时间的增加，它将逐渐缩短。有关火箭背囊及其功能的详细说明，请参看后面的“飞行运动”部分。
- **当前对话：**在游戏过程中，Sam 将与各种人物进行交谈；和 Sam 一起冒险的朋友、Electric Priestess 以及 Sam 遇到的其他角色都会同 Sam 对话，其中 Priestess 将通过无线电与 Sam 联络。因此，必须事先录制好所有对话内容，然后再回放给玩家。在屏幕左上角会出现角色的 2D 卡通图形，旁边则显示字幕。如果玩家在游戏过程中关掉声音或者不打算听到对话，字幕会显得非常重要。这种 GUI 功能将持续相当一段时间才会消失，这样，玩家便能从容不迫地读完字幕。不过，如果游戏处于非互动性的片头动画状态，对话字幕就会出现在屏幕底部偏上部位，就像带字幕的电影一样。

2.4 重新开始和保存游戏

在“Atomic Sam”中，玩家毫无“气数”可言。当 Sam 被某个机器人或其他的敌手打败（相对来说始终以非暴力方式）后，玩家可返回到刚才未能通过的小关卡，反复运行这一段游戏，直到顺利过关。在各关卡中分布着一些小关卡，游戏能自动、精确地记录玩家抵达某个小关卡的时间。应该认真地安排好小关卡的位置，做到既能提高游戏难度，又不至于让玩家感到灰心丧气。

在游戏过程中，玩家可以随时保存游戏。不过，在调出保存过的游戏时，玩家只能在保存游戏的关卡重新开始游戏，而不是精确地返回到 Sam 在这一关卡所处的具体位置（或小关卡）。这样做可以鼓励玩家在停止游戏之前坚持打过这一关。

2.5 控制方法简介

玩家可运用不同的控制方法操纵 Atomic Sam，使其在游戏环境中自由活动。接下来，我将对这些控制方法进行详细论述。不过，首先有必要对不同的命令进行一番简介，读者从中可以大致了解 Sam 的各项能力。这些控制方法的设计均考虑到各种新型的游戏控制器，因此，它们能轻而易举地适应本游戏所面向的任何系统。

- **上、下、左、右（模拟控制器 Analog Controller）：**玩家可运用这一控制方法操纵 Sam 在游戏环境中的水平地面活动。由于这一方法采用了模拟技术，因此，如果轻轻按住控制按钮，Sam 就会缓缓移动；但如果用力按下表示某个方向的按钮，Sam 就会朝那一方向迅速移动。
- **上飞、下飞（左、右扳机按钮 Left and Right Back Triggers）：**玩家可利用这些控制方法使 Sam 在游戏环境中做上下垂直运动。





- 扔出（右控制板“向下”按钮 Right-Pad Down Button）：这一按钮可使 Sam 扔出当前投射物。
- 下一个投射物（右控制板“向右”按钮 Right-Pad Right Button）：玩家可使用这一按钮滚动 Sam 的投射物目录。
- 动作（右控制板“向上”按钮 Right-Pad Up Button）：使用这一按钮，玩家可在游戏环境中执行各式各样的动作，如扳动开关、与角色交谈或捡起大的物品等。
- 观察（右控制板“向左”按钮 Right-Pad Left Button）：玩家可使用这一按钮激活摄影机的观察功能。

2.6 运动方向

当 Sam 位于地面或浮在空中时，玩家可使其在游戏环境中做前、后、左、右运动。利用游戏控制板上的模拟控制器，玩家便能操纵 Sam 朝这些方向运动。控制方法始终与摄影机的视角相关联。因此，按住控制器上的向前（forward）或向上（up）按钮，Sam 就会远离摄影机，反之，按住向后（backward）按钮，Sam 则会走近摄影机。同样，按住向左（left）或向右（right）按钮，Sam 也会参照摄影机的位置在游戏环境中朝相应的方向运动。

2.6.1 调整方向

当 Sam 试图朝调整方向移动时，起先他仍然会保持当前方向，然后才会转身朝新的方向走动。例如，当 Sam 背向摄影机运动时，玩家按住“向左”按钮，这时，Sam 就会朝这一方向侧移或侧飞。玩家必须在一小段时间内（大约一秒钟）保持这一方向，才能使 Sam 的整个身体面对新的方向。同样，从当前方向改为向后运动也是如此：起先 Sam 开始向后运动，一秒钟后，他才会转过 180 度，并继续朝这一方向运动。这样，当 Sam 稍微调整运动方向时，他其实并未真正转向。

2.6.2 改变运动速度

如果使用控制系统中的模拟控制器操纵不同方向的运动，能调整 Sam 在某一方向上的运动速度。如果玩家使劲按住模拟控制器上表示某一方向的按钮，Sam 就会朝这一方向快速运动，反之，如果只是轻轻按住，Sam 运动的速度将大大降低。这样，玩家就能精确地控制 Sam 在游戏环境中的方位。

2.7 飞行运动

当 Sam 在游戏环境中活动时，他背在身上的火箭背囊起着关键性的作用。玩家可用上飞（Fly Up）和下飞（Fly Down）按钮对火箭背囊进行控制，这样，在游戏环境中，Sam 就能沿垂直方向运动。如果两个按钮均未按住，那么，当 Sam 升入空中时，他就会在给定的高度盘旋。

2.7.1 上下运动

Sam 不会以恒定的速度上下运动。如果玩家使劲按住向上（Up）按钮，起先，Sam 会快速运动，而且，按住这一按钮的时间越长，速度也就越快。当这一速度最终达到极限时（向上运





动大约一秒钟后)，Sam 的运动速度就不会再继续提高。同样，向下运动基本上也以这种方式操作。

2.7.2 停止运动

当玩家停止向上、向下或某一方向的飞行时，Sam 不会立即中止运动，而是“悠然”而止。当 Sam 停止运动时，动画将显示他正在迅速调整自己的重心，以改变火箭背囊面对的方向。这就意味着，玩家必须不断地练习飞行技术，只有这样，Sam 才能准确地在预定的地方停止运动。

2.7.3 飞行速度

Sam 的火箭背囊并不是一种极快的装置，它的最高速度约为 Sam 在地面慢跑时的 1.5 倍。每当玩家操纵 Sam 回到地面时，Sam 就会恢复行走和慢跑动画状态，这时，Sam 将放慢运动速度，因为他已经置身于地面。

2.7.4 飞行方向

当然，Sam 也可在向前、向后、向左或者向右运动的同时沿垂直方向运动。玩家只需同时按住模拟控制器上的方向按钮和向上、向下飞行按钮即可。这时，Sam 将适度调整自己的方向，使其与整体运动保持一致。

2.7.5 爆发速度

向上按钮和向下按钮都能使 Sam 达到最高运动速度，但如果将其中任何一个方向按钮快速敲击两次，就能获得一种“爆发”速度，即在短时间内，Sam 的运动速度将达到上述最高速度的 1.5 倍左右。不过，如此高的运动速度也会让火箭背囊消耗更多的能量。这一控制方法有助于快速躲避敌人的攻击。

2.7.6 飞行时间限制

火箭背囊的飞行时间并非毫无限制，不过，幸运的是，它能在非使用状态下自行补充能量。无论向上、向下还是自由盘旋，只要 Sam 脱离地面，火箭背囊就会消耗能量。火箭背囊中剩下的能量数值由位于屏幕右下角的游戏环境视角图上端的一道小条纹表示，这样，玩家就能随时了解 Sam 剩下的飞行时间。火箭背囊的能耗状况取决于它的使用方式。以下是在不同使用方式下能耗比率的大致情况：

使用方式	能耗率
上飞	4
下飞	2
自由盘旋	1
向上爆发	6
向下爆发	6
着陆	-3

2.7.7 着陆

由于火箭背囊的能量有限，玩家必须定时让 Sam 着陆，以便补充能量。玩家只需操纵 Sam



靠近地面或任何他可以站立的平面，即可使其着陆。由于 Sam 的飞行距离有限，因此玩家必须根据情况计划好 Sam 的每一次行动，以便他能顺利地从一个地方抵达另一地方。这种情况为谜题的设置提供了机会：由于 Sam 的飞行能力有限，在每一关卡中，玩家必须摸清使 Sam 抵达某一地点的具体方法。在飞往远处的着陆点时，笔直的路线往往并不可取。

2.7.8 跌落

当 Sam 在半空中耗尽火箭背囊中的所有能量时，他并不会跌落而死。Sam 的装备包括一种特制的减震靴，由于靴底格外厚实，因此 Sam 能从任何高度安全跌落。当火箭背囊中的能量完全耗尽时，Sam 就会以极快的速度从高空垂直跌落；在这一过程中，玩家将体验到一种极度失控的感受。

2.7.9 高度限制

此外，火箭背囊只能达到有限的高度。如果玩家试图让 Sam 超过应有的飞行高度，火箭背囊就会劈啪作响，表示 Sam 已无法飞得更高。因此，尽管每一关卡的天空都处于开放状态，但事实上玩家根本无法飞出任何一个关卡。

2.7.10 火箭背囊升级

在整个游戏过程中，Sam 将不时得到升级火箭背囊的机会。有时，Sam 会找到一些能装入背囊的附加部件，有时也可能遇到几位能替他修理、改进背囊的角色。这些改造将以不同形式提高 Sam 的飞行能力。

- **增加飞行时间：**Sam 能飞行更长时间而无需着陆。这就意味着，为了抵达某地，Sam 也许将不得不获取相应的升级服务。
- **提高爆发速度：**利用背囊的“爆发”功能，Sam 能获得更快的飞行速度。
- **提高总体速度：**由于背囊的最高速度和加速度得到提高，Sam 将获取更高的垂直运动速度。
- **改善可操纵性：**背囊的“立即停止”能力将有所提高。这样，一旦玩家松开操纵杆，Sam 就能立即停止运动，而不会像原先那样悠然而止。

2.8 地形

一般来说，Sam 能在任何平面上行走或降落，无论它是人行道、地面或高空中的降落平台。Sam 将无法在曲面或斜度过大的地表降落。如果 Sam 试图登上或在曲面及斜坡上降落，他就会从上往下滑落，直到降至水平地表。

此外，Sam 还无法在某些物质上降落或行走。这些物质包括水、焦油覆盖区或带电的地板。如果玩家让 Sam 直接站立在这类物质上面，Sam 的动画就会显示这类物质造成的危险。例如，当 Sam 走上带电地板时，会表现出受到电击的样子。如果玩家不及时改变控制方向将 Sam 调离这种地板，他很快就会丧失行动能力。同样，如果玩家在火箭背囊的能量尚未耗尽之时试图让 Sam 降落在这类物质的表面，Sam 也会遭受电击，这时，Sam 的动画形象将提前显示地板的危险，以便玩家能及时将 Sam 调离这一区域。

如果玩家降落在这类物质表面时能量已经耗尽，Sam 就会跌倒并失去活动能力，这时，玩





家将无法帮助他脱离危险。当然，每当 Sam 丧失活动能力时，玩家必须从上次自动保存的小关卡重新开始游戏。为了在游戏中取得胜利，玩家应该避免让 Sam 步入这类物质表面；此外，当 Sam 降落其上时，应保证火箭背囊内的能量尚未消耗殆尽。

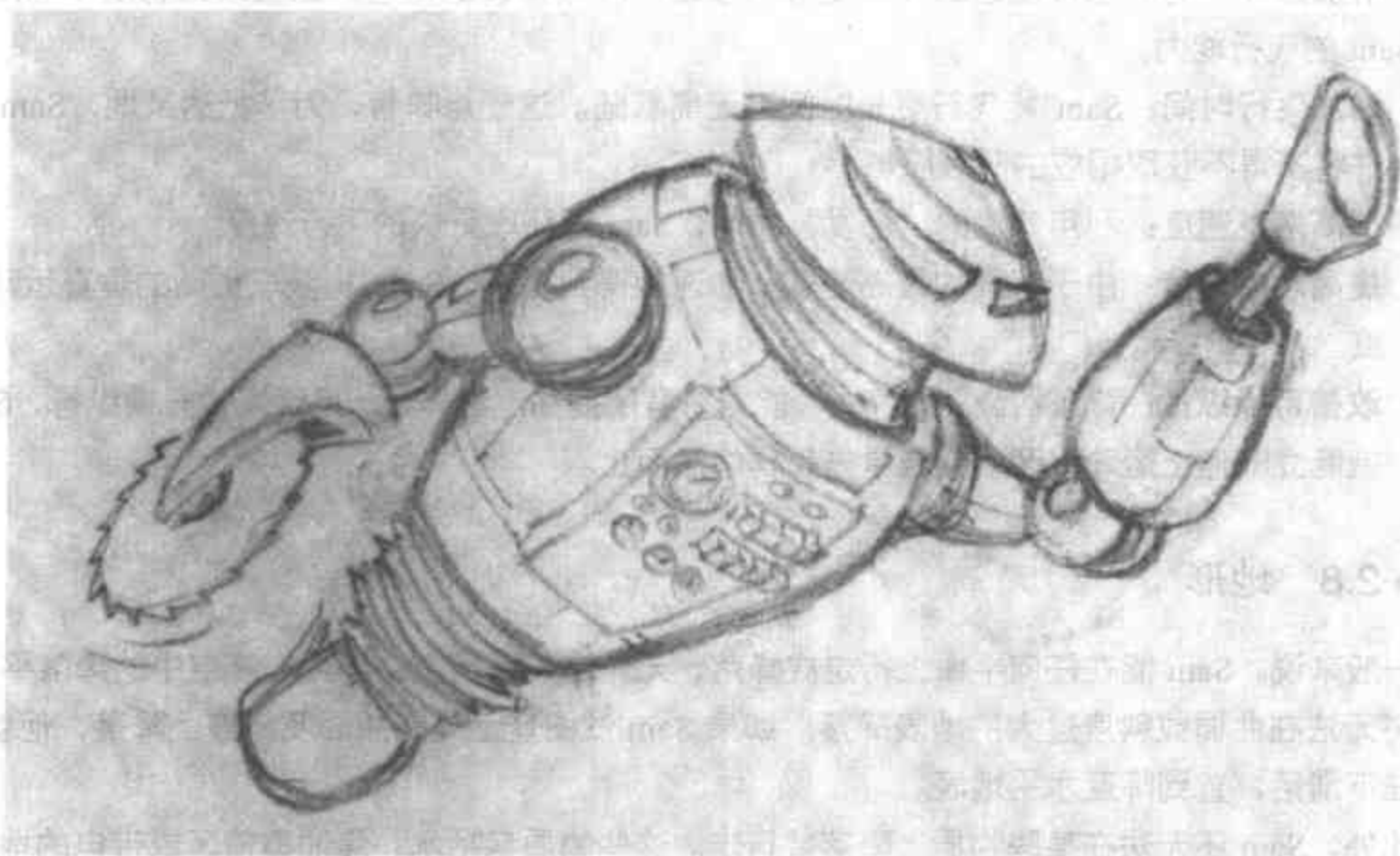
2.9 捡起物品

每当 Sam 飞近某个物品时，他能捡起这一物品。这时，如果物品目录有足够的空间，他会自动捡起物品。Sam 能捡起的物品包括投射物、火箭升级部件和电能比拉鱼。Sam 的动画形象及其伴随的声音将表明他已经捡起某个物品。

此外，Sam 也能捡起某些尺寸更大的物品，但他无法将其纳入物品目录。有时，Sam 必须移开这些物品才能解开某个谜题；有时则能将其投向敌人，使敌人失去反击能力。当 Sam 靠近敌人时，玩家可按住动作键（Action）让 Sam 捡起这类物品，然后释放动作键将其砸向敌人。

2.10 扔出投射物

在整个游戏过程中，Sam 可以找到并扔出不同类型的物品，这些物品在应付眼前的敌对机器人时起着至关重要的作用。尽管 Sam 无法找到或使用枪支，但他可以获取不同类型的物品，然后将它们投向敌人，使其失去反抗能力。



附图 2

2.10.1 物品目录

Sam 拥有一个简洁的物品目录，该目录可容纳 50 枚不同类型的投射物。Sam 捡到的投射物将自动储存在这一目录之中。目录使用简便，但玩家无法通过减少某种投射物为另一种投射物





腾出空间。Sam 只能以投射的方式清除目录中的物品。

2.10.2 收集投射物

游戏开始时，Sam 拥有少量投射物；随着游戏的推进，他将陆续找到更多投射物。通常，当 Sam 发现一种投射物时，数量往往相当可观，比如 10 个水瓶或 20 个粘性球等。操纵 Sam 靠近这些投射物，他就会自动捡起它们。如果 Sam 扔出的投射物并未打中敌人，那么，战斗结束之后，他还可以走过去回收那枚已经落地的投射物。这样，那些投射不太准确的玩家便可重新捡起扔出的物品，并在需要之时再行投射。

2.10.3 准备投射

当某个投射物处于准备状态时，玩家会看到 Sam 手持这一投射物的形象。同时，屏幕右下角将出现一个图标和计数器，表示这种投射物的当前数量。如果某个投射物处于准备状态，那么，一旦玩家按住投射按钮（Throw），Sam 就会将其射出。

玩家可用“下一个投射物”（Next Projectile）按钮选择“合意”的投射物。如果玩家快速点击这一按钮，Sam 就会改用物品目录中的下一个投射物（如果存在的话）。倘若玩家按住“下一个投射物”按钮，屏幕上方就会出现物品目录中现有各类投射物的水平列表，被选中的武器位于列表中央。玩家可采用方向控制器中的向左和向右按钮依次选中前一个和下一个投射物，这时，投射物列表将分别向左或向右滑动。由于这一列表“首尾相接”，玩家只需反复点击其中一个按钮便能浏览到所有投射物。当玩家松开“下一个投射物”按钮时，位于屏幕中央的投射物将成为 Sam 重新选中的武器。

选择妥当后，玩家就会看到 Sam 手持当前投射物的形象。之后，如果玩家并未扔出或重新选择当前投射物，五秒钟后，Sam 便会隐藏手中的投射物。因此，表面上看，Sam 不会在任何场合都摆出一付投射姿态。不过，即使 Sam 并未摆出投射姿态，一旦点击投射按钮，Sam 将立即扔出投射物，动作和手持投射物时一样敏捷。

2.10.4 扔出投射物

玩家可使用“投射”（Throw）按钮扔出 Sam 的当前投射物。投射物将沿着玩家面对的大致方向前进，Sam 不必“精确瞄准”便能击中目标，因为游戏会自动瞄准 Sam 面前最近的敌人。当前目标会附上十字形瞄准线，以便玩家始终能识别 Sam 准备打击的目标。此外，有必要对自动瞄准状态进行调整，这样既能避免投射物击中玩家不打算攻击的目标，也可防止游戏过于简单化。

2.10.5 投射速度与距离

一旦松开“投射”按钮，Sam 便会扔出投射物。玩家只需点击一下“投射”按钮，即可完成简单的投射动作。不过，如果玩家持续按住这一按钮，Sam 扔出的投射物就能飞得更快、更远。这表现在，当玩家按住“投射”按钮时，Sam 的手臂开始旋转，就像垒球运动中投球手准备投球时一样，只是 Sam 的手臂将持续进行旋转运动。最后，当 Sam 手中的投射物以最高速度飞出时，他的手臂会因猛烈转动而一片模糊，就像卡通影片一样。尽管自动瞄准功能可将投射物对准敌人，但如果玩家在投射时用力不够，它就达不到目标，因而也就无法击中目标。对玩家来说，这款游戏的挑战之一是，以足够的力量甩出投射物，使其能达到并击中预定目标。





2.10.6 打击能力

在这款游戏中，每一种投射物都能击溃特定类型的敌人。例如，粘性球可使行走在地面或墙壁上的敌人牢牢地粘附其上，无法动弹。不过，在对付飞行中的敌人时，粘性球将无济于事。另一种投射物，即水瓶，最适于对付那些不防水的机器人，使它们的电路发生短路。装甲厚实的机器人或敌人不会受到水瓶的损害。玩家必须小心地捡起能在某一固定环境中使用的投射物。玩家可以在游戏要素部分找到对投射物的打击能力更为细致的描述。



2.11 电能比拉鱼

除了 Sam 在游戏环境中发现的投射物和得到改进的火箭背囊，玩家还会发现一种特殊的物品，这一物品能以消极方式抵御玩家受到的攻击。电能比拉鱼是一种金属结构的鱼形绿色装置，当 Sam 发现并捡起它们后，它们将围绕 Sam 浮动或“游泳”，就像沿着轨道飞行的卫星一般。当敌方扔出投射物时，比拉鱼可进入该投射物的飞行轨迹并将该投射物“一口吞掉”，从而起到截击敌方投射物的作用。如果敌人试图展开近距离搏斗，而 Sam 拥有一只电能比拉鱼，这时，比拉鱼就会咬伤来犯之敌，使其丧失反击能力。当比拉鱼成功地保护 Sam 免受敌人攻击后，它会在爆炸中解体。Sam 一次能收集四只电动比拉鱼，当 Sam 处于极其危险的境地时，电能比拉鱼将起到关键性的作用，帮助 Sam 渡过难关。



2.12 动作

玩家可利用特殊的动作按钮操纵 Sam 完成各种游戏动作。这种动作按钮可提供各式各样的动作，而且，游戏将自行确定在某一场合下 Sam 应该采用的正确行动。

2.12.1 开关和按钮

如果 Sam 靠近某个按钮或开关时玩家敲击一下动作键 (Action)，Sam 就会按住按钮或扳动开关手柄。开关可以完成一些简单动作，如门的开关或平台的升降等，也能执行一些更为复杂的动作，如起重机的启动或蒸汽孔的打开等。

2.12.2 推动与操纵

Sam 可以推动某些物品；按住动作键，Sam 就能执行这一动作。这些物品包括板条箱、圆桶以及各式各样的球形物，需要推动这些物品的原因多种多样，如阻拦或疏通道路等。

2.12.3 手动拾物

Sam 可使用动作键捡起某些形状较大的物品。这些物品不同于 Sam 能自动捡起的投射物，因为他不会将它们添入物品目录，而且，当 Sam 手持这类物品时，他便无法扔出任何投射物，除非将其放下。当 Sam 靠近这类物品时，玩家可按住动作键，操纵 Sam 捡起它们。一旦捡起这类物品，Sam 便能随身携带它们，直到玩家松开动作键，才能将其扔下。当 Sam 手持物品，尤其是沉重的物品时，他的运动速度可能会明显减慢。玩家可命令 Sam 带上这些物品，并应用这些物品打败敌手。例如，Sam 能捡起一块硕大的铁砧，携带它飞上天空，然后巧妙地将其砸向某个难以对付的机器人。





2.12.4 交谈

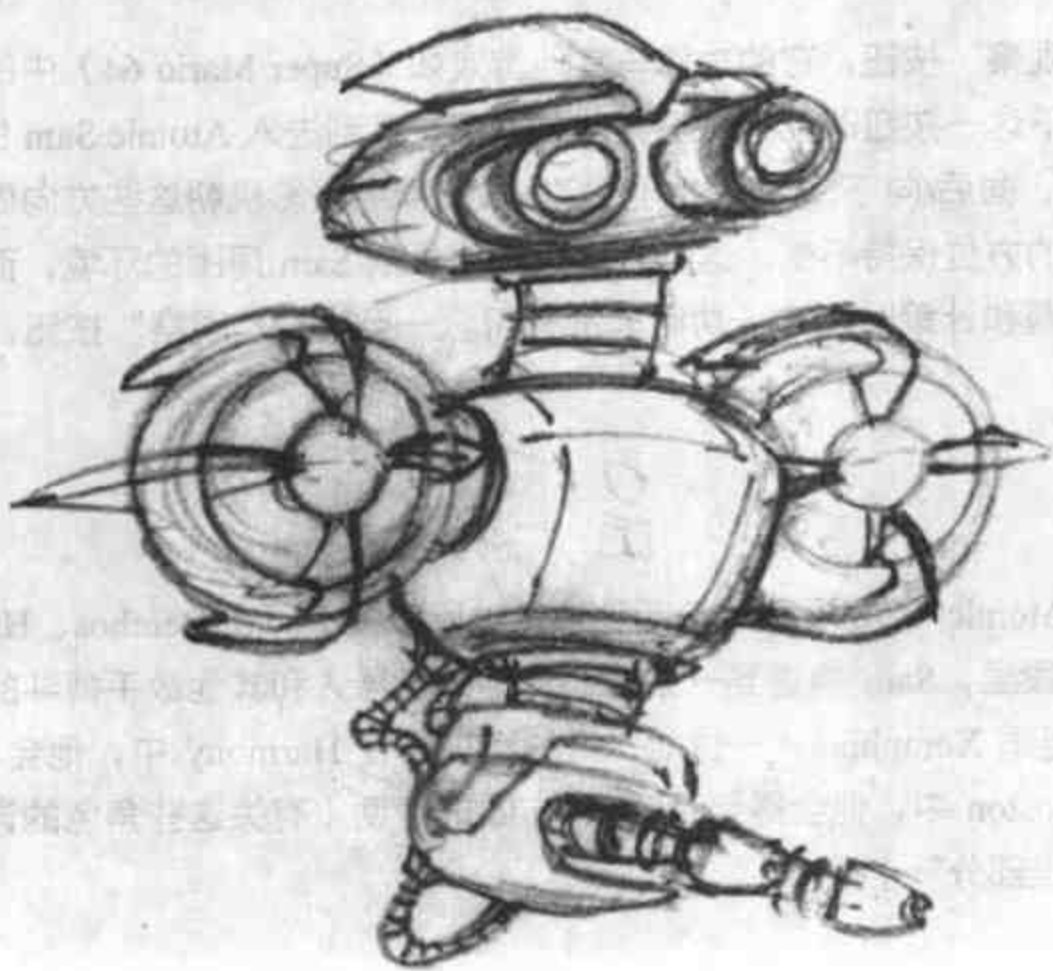
在“Atomic Sam”中，某些非敌方角色将乐于和主人公交谈，不过，这种交谈通常只能持续一到两句。如果玩家希望 Sam 与某个角色交谈，就必须等 Sam 靠近这一角色后按下动作键。这类角色不仅能补充“Atomic Sam”的一些前续故事，同时也能使关卡富于人间气息和趣味性。这类角色包括一种“信息机器人”，它是 Sam 那一时代的发明成果，其任务是给人类提供有用的建议。除了获取信息之外，Sam 还应主动和某些角色交谈，因为这些角色能提供火箭升级服务。

2.12.5 阅读

玩家可以看到各种告示牌或张贴在墙上的海报。点击动作键即可快速放大和阅读这些告示。这些告示包括地图或“旅游信息”，其中地图有助于玩家通过每一关卡，而“旅游信息”则记载了 Sam 所处地点的历史记录。

2.13 互动战斗环境

除了用投射物打击敌人之外，Sam 还能利用一些关卡要素来对付和打败他们。玩家可使用动作键引发各种事件，从而促使 Sam 面对的各类敌人丧失反击能力。在“Atomic Sam”的关卡中，这类计策随处可见，有些甚至达到鲁本·戈德堡式漫画的复杂水平。在整个游戏过程中，领悟和正确使用各式各样的计策是打败各类机器敌手的重要因素。事实上，离开这些阴谋诡计，玩家便无法击败某些敌人。在某种程度上，这些计策就是一系列“战斗谜题”，玩家必须在实时环境中解开这些谜题，以便找到克敌制胜的最佳方法。



附图 3





这些计策均由关卡设计者构思和确定，以便能恰到好处地将它们应用于各个关卡中。有些关键性的计策可能会以不同的形式在某个关卡中反复使用。有些计策只能使用一次，而另一些则可反复使用。对于能多次使用的计策，玩家能通过反复尝试找到正确的使用方法。在设计这些计策和环境时，关卡设计者应该采用适当的结构方式，以便玩家能比较顺利地摸清它们的作用以及正确使用它们的方法。

以下是几个潜在的计策例子：

- **蒸汽孔：**打开位于蒸汽孔旁侧的开关，蒸汽就会喷涌而出，击昏或熔化处于气流之中的敌人或物品。如果玩家恰好等到敌人误入蒸汽喷流之时扳动开关，敌人就会在蒸汽的灼烧下失去反击能力。
- **风扇：**打开一台大型风扇旁侧的开关，风扇便能做短时转动。例如，当风扇对面的蒸汽孔进入工作状态时，风扇及时运转所产生的强大风力可将怪物吹进蒸汽孔。
- **油桶和杠杆：**也许 Sam 会偶然发现一块木板横亘在一口铁箱之上，组成简单的杠杆结构。这时，可将一只巨大的空油桶置于杠杆下端。如果玩家等到 Sam 靠近杠杆上端时敲击动作键，Sam 就能按下杠杆，从而使油桶弹入空中，在这一过程中，油桶也许会击中一两个敌人。

如果这些计策使用不当，就会适得其反并最终伤及 Sam。例如，如果 Sam 在扳动蒸汽孔开关时正好处于气流之中，他的火箭背囊就会在高温中熔化消解，并将他猛然击倒。毫无疑问，在很大程度上，有效使用这些计策取决于敌人是否处于适当的位置。对玩家来说，将机器人和其他敌手诱入这些圈套是一项有趣而艰巨的任务。

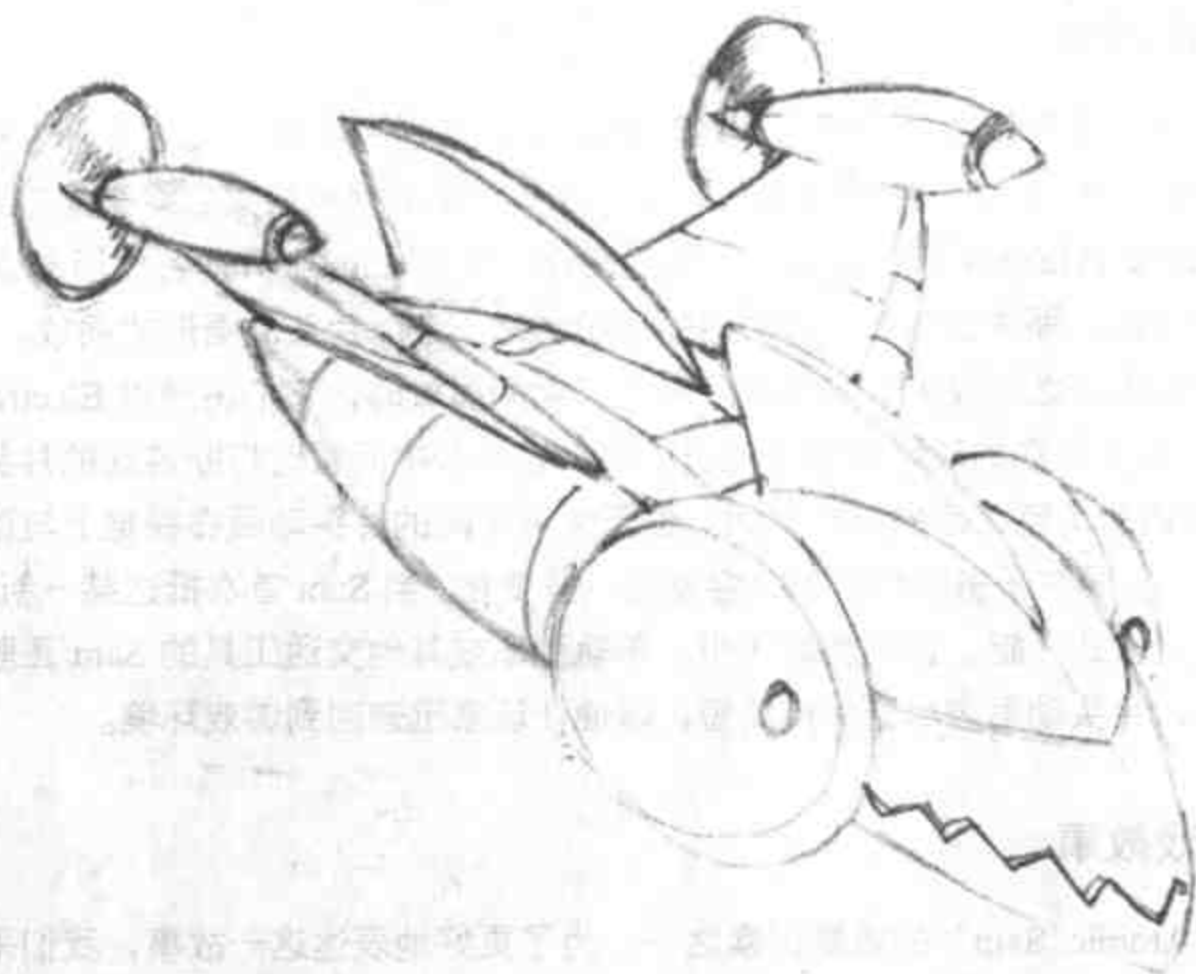
2.14 观察

玩家还可使用“观察”按钮，它的功能与其他游戏如《Super Mario 64》中的“观察”按钮毫无二致。当玩家按下这一按钮时，摄影机会向前推进，直到进入 Atomic Sam 的头颅。这时，玩家可使用向前/向上、向后/向下、向左和向右控制按钮操纵摄影机朝这些方向做俯仰和回旋运动；与此同时，Sam 的方位保持不变。这样，玩家就能看清 Sam 周围的环境，而不必使其逼近这些景物。在研究谜题和计策时，这一功能非常有用。一旦松开“观察”按钮，摄影机就会恢复到正常游戏模式。

2.15 朋友

在这款游戏中，Atomic Sam 不必单枪匹马地闯过所有关卡。在 Benthos、Harmony 和 New Boston 这三个中间阶段里，Sam 将遇到一些能协助他与机器人和其他敌手搏斗的游戏角色。在 Benthos 中，Sam 将邂逅 Xeraphina，一位能飞行的女孩；在 Harmony 中，他会与机器人 Scrap 结伴而行；在 New Boston 中，他会得到月球人 Dulo 的帮助（有关这些角色的详细情况，见本设计文件的“游戏过程部分”）。





附图 4

这些朋友不会像 Sam 那样善于打败敌人，但他们能帮助 Sam 识别敌人，警告他袭击就要来临，暗示谜题答案，指出 Sam 能捡起物品以及可以躲藏的地方，或者提示继续前进的最佳方向。在通过每一道关卡的旅途中，朋友们会与 Sam 频繁交谈，提供前续故事、有用信息以及风趣的对话等。在通常情况下，这些朋友从不会真正死去或被俘，他们总是能击退那些针对自己的来犯之敌。关于朋友 AI 的详细情况，见本设计文件的“人工智能”部分。

2.16 对话

对玩家来说，Atomic Sam 之所以能成为一个富于魅力和令人难忘的角色，在很大程度上归功于他在整个游戏中的对话语句。这些对话不仅存在于片头动画，而且还会出现在实际游戏过程中。当 Sam 战胜各类敌手时，他会说出一些一般性的语句，玩家无法控制这些语句，因为它们的存在只是为了丰富游戏经历而已。这些语句不仅符合 Sam 的年龄，而且也符合这款游戏的回返未来的乐观基调。这些口头禅包括：“你无法阻止将来！”、“Atomic 就是答案！”、“该死的机器！”以及“你是落后技术！”等。当 Sam 的投射物已经用完，或者火箭背囊中的能量即将耗尽时，他就会提供一些有用的信息。此外，Sam 还会在特殊的游戏事件中说出一一些特殊的对话语句，例如当他第一次在月球表面行走或第一次遭遇特定的大怪兽之时。如果 Sam 在实际游戏中保持这种对话状态，玩家就会逐渐迷上这一角色，进而对他在游戏环境中的境遇关心倍至。





2.17 片头动画

为了更好地将故事情节传达给玩家，这款游戏将采用一些简短的片头动画。这些片头动画由本游戏的 3D 引擎支持，因此，互动游戏和非互动片头动画的视觉效果显得毫无二致。片头动画包括 Sam 和 Electric Priestess 等各类角色之间的交谈、陪伴 Sam 旅行的各类朋友或 Sam 在不同地域发现的其他角色。那些仅包含三言两语的极短对话不会以片头动画形式播放。

片头动画可以在关卡之间或关卡内播放。在关卡之间播放时，它们将通过 Electric Priestess 提供的信息对 Sam 即将面临的环境和挑战进行说明。在关卡中间暂时打断游戏的片头动画包括 Sam 与那些偶然相遇的角色之间的简短对话。位于关卡中间的片头动画在视觉上与游戏环境完全一致，它们的区别主要在于摄影机的视角会发生一些变化。当 Sam 首次抵达某一新的地点时，玩家将会看到，乘坐软式飞艇、自动旋翼飞机、单轨列车或其他交通工具的 Sam 正驶入这一游戏地点。总的来说，片头动画应尽量保持简短，以便让玩家迅速回到游戏环境。

2.18 游戏故事

游戏故事是“Atomic Sam”的重要因素之一；为了更好地表达这一故事，我们采用了各式各样的表现手法。当然，这包括前面提及的片头动画。这些片头动画将传达所有重要信息，它们是玩家在游戏中取得成功的必要因素。不过，由于这些片头动画属非互动性部分，因此，它们的长度将受到严格限制，以便玩家能迅速回到游戏环境。为了丰富游戏故事，在实际游戏中，一些故事外情节将通过 Sam 采用的各种计策得到揭示。

2.18.1 环境

毫无疑问，环境（关卡）本身就具备一种关键性的叙事功能，即烘托背景气氛。必须特别注意的是，应确保“Atomic Sam”中每道关卡与游戏环境相互一致，并与所有故事要素毫无冲突。

2.18.2 朋友

Sam 遇到的朋友将陪伴他在各类环境中旅行，他们跟随 Sam 飞行时会向他透露一些信息。这些角色时而会介绍某一特定环境的历史情况，时而会提供一些涉及将来世界的有趣信息。毕竟，Sam 还是一个孩子，对生活充满好奇。当然，只有在非战斗状态时这些朋友才会与 Sam 交谈，因为这时玩家将致力于探险，而不是打败那些气势汹汹的机器人。如前文所述，朋友们的所有谈话将通过内置式 GUI 显示在荧光屏上。

2.18.3 无线电

首次见面后，Electric Priestess 会送给 Sam 一台袖珍无线电通信机，Sam 可将其置于耳旁。在探索关卡的过程中，玩家可通过这台无线电收听各种信息。与那些朋友一样，Priestess 也会向 Sam 讲解他路过的某一地区的风土人情以及当前敌人的特点。如前文所述，在无线电上收听到的所有谈话将通过内置式 GUI 显示在荧光屏上。





2.18.4 告示牌

正如在“动作”部分所述，Sam 还会看到一些可阅读的静态信息显示。在“Atomic Sam”的环境中，这些告示牌只是表现故事的另一种方式而已。

2.19 关卡

“Atomic Sam”与其他由控制器操纵的第三人称动作兼冒险类游戏的不同之处在于，它强调的并不是探险活动，而是 Sam 在各关卡中的战斗经历，而且，在这些关卡中，Sam 将避开那些阻拦自己的机器人和其他敌手。固然，在设计这些关卡时，应使其在视觉上充满趣味性和感染力，但要想保持玩家对某一关卡的兴致，重点在于让他们面对更多新奇的挑战，而不仅仅是发现下一道关卡。总之，每道关卡应保持疯狂、激烈的游戏场面；为了使 Sam 最终找到自己的父母，玩家的瞬间决定和操作技巧将起到关键性的作用。一般来说，Sam 将以两种方式和机器人展开搏斗。第一种方式是同时与多个机器人较量，不过这些机器人的能力较差；第二种方式是和远远强于自己的单个敌人或“关底老怪（boss）”交手。通常，和关底老怪交手时，必须找到一种克敌制胜的特定方法，而且，与同时对付多个敌人相比，玩家应该更加善于思考。本设计文件的前半部分已经就玩家操纵 Sam 的方法以及同游戏环境的互动方式做过具体讨论，结论是，在这款游戏中，并非每时每刻都应充满那种疯狂的打斗场面。在与机器人搏斗的过程中，也会出现一些平静、“安全”的时刻，这时，玩家可稍事休息，恢复原有的姿态。那些能自动保存游戏的小关卡（见后文）就设置在这些比较安宁的场景中。这样，玩家就可以在比较安全的区域内重新开始游戏。此外，某些“安全”区域可能会要求玩家解决一些简单的谜题，然后才能继续玩下去。

2.19.1 关键路径

“Atomic Sam”的每道关卡都设置了一条固定不变的“关键路径”，玩家最好通过这条特定的路线结束当前关卡，同时进入下一道关卡。尽管在这条路线旁侧可能会出现一些异想不到的或者秘密的地区，但在整个关卡中，关键路径将保持主导地位。在游戏的每一区域（包括 Gargantuopolis、Benthos、Harmony、New Boston 以及 Ikairus）中，玩家必须以特定的顺序完成这一区域内的所有关卡，因为这样做将有助于故事情节的完美表达，有助于适当营造紧张的气氛，有助于在关卡进程中逐渐提高游戏难度。

2.19.2 训练关卡

游戏一开始，倘若玩家愿意，他将有机会接受一次特别“训练”。起先，当 Sam 回到家里，发现父母双双失踪时，他会毫不犹豫地穿上火箭背囊，踏上寻亲之旅。火箭背囊附有一台颇为有用的说明机器人 Instructobot，这台一品脱大小的机器人能以机器腔调向 Sam 传授火箭背囊的使用方法。也就是说，这台说明机器人将鼓励玩家试用火箭背囊，使其掌握这一工具的操作方法。在 Sam 家的安全区域内，玩家可尝试各种操作方法，然后再冒险进入危机四伏的外部环境。

2.19.3 Electric Priestess

在本游戏中，最“平静”的区域是 Electric Priestess 的泡泡（玻璃）屋。这是一个远离战事





的迷你型关卡，在这一关卡中，泡泡屋成为衔接 Benthos、Harmony 和 New Boston 三个区域的“枢纽”。在 Electric Priestess 的家里，玩家可与她交谈，并选择一个打算前往的游戏区域，同时进入下一道关卡；在这一过程中，Sam 将不会受到任何威胁。有关 Electric Priestess 和各区域的详细介绍，见本设计文件中的“游戏进展”部分。

2.19.4 环境顺序

玩家可选择体验各主要游戏区域或“世界”的顺序。在游戏开头部分，结束 Gargantuopolis 各关后，Electric Priestess 将建议 Sam 前往以下区域：Benthos、Harmony 或 New Boston。每个区域的难度大致相当，不过，由于给每个区域安排的挑战各不相同，因此对各区域难度的感知可能会因人而异。在这种情况下，玩家可以选择自己认为最容易的区域（在某一区域当中，如果玩家觉得这一区域难度太大或者只是对它感到厌倦，可以随时回到 Electric Priestess 的泡泡屋，在那儿，玩家能重新选择其他区域）。有关游戏流程的详细介绍，见本设计文件的“游戏过程”部分。



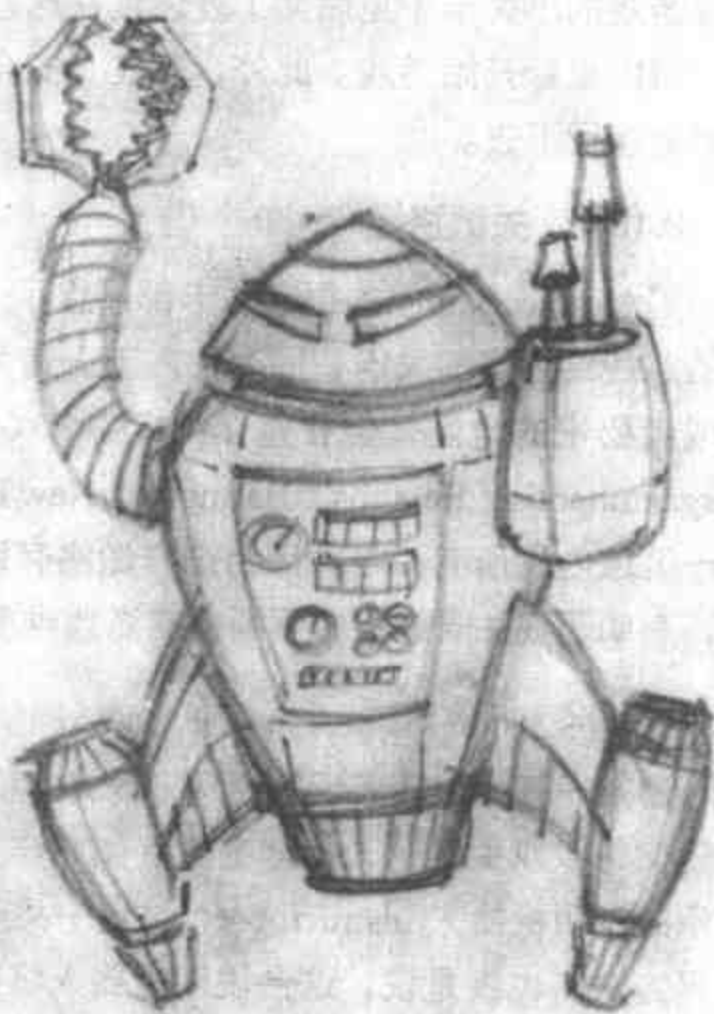
3 人工智能

由于“Atomic Sam”是一款以妙趣横生的战斗场面为基调的游戏，因此，在这款游戏中，AI 的首要任务是支持这些战斗场面，为玩家提供动人心魄的挑战。AI 的另一个关键性作用是，赋予 Atomic Sam 邂逅的朋友们一定程度的生命感，使其不至于显得过于机械。



3.1 敌人 AI

Sam 面对的许多敌人都是机器人。因此，一方面，这些敌人的 AI 可能会相当迟钝，另一方面，他们仍将保持一定的可信度。事实上，由于某些对手头脑简单，Sam 可利用在各关卡中发现的互动环境给他们设置陷阱。不过，并非所有机器人都那么愚蠢。随着游戏的继续，关卡难度逐渐升级，机器人也越来越聪明，从而更加难以对付。再往前推进，玩家将“冷面雇佣兵”这样的人类敌手开战。这些人类敌手在进行战斗决策时应表现得与真人一样聪明。



附图 5



3.2 玩家探测能力

不同的 AI 主体探测和跟踪玩家的能力各不相同，而这又将影响玩家面临的挑战难度。有些机器人只能以极其狭小的视野观看眼前的景物，而另外一些机器人则能以 360 度的视野进行全方位观察。此外，敌人们的探测距离也会因人而异：有些敌人只能在靠近 Sam 时才能看到他，而另外一些敌人则能在 Sam 看到他们之前便认出他的身影。某些机器人具有“超视觉”功能，他们能透视墙壁，无论 Sam 躲在何处，他们总是能发现他的身影。

有些机器人的记忆非常短促。如果 Sam 设法紧跟后，完全避开他们的视野，他们可能会彻底忘记 Sam 的存在，恢复到空闲状态。某些机器人一旦锁定 Sam 的方位，就会紧跟不舍。玩家必须摸清敌人探测 Sam 的能力，努力使自己处于有利地位。

3.3 运动

所有敌人将以真实可信的方式进行运动，他们均采用一种简单的物理系统，以便使 Sam 所处的环境更为逼真。不过，从 Sam 在游戏中的表现可以看出，这是一款由控制台操纵的动作游戏，因此不必过于依赖完全“真实”的运动系统。事实上，“Atomic Sam”中的回返未来基调以及那些奇异、荒谬的飞行器充分表明，这个世界并未严格遵循物理规律。

3.4 飞行

与 Sam 交手的许多敌人都能在空中飞行，因此，必须使这些怪物的飞行动作具备一定的可信度，这一点非常重要。飞行动作的式样将随敌人使用的飞行设备的不同而发生显著的变化。乘坐小型飞艇的敌人只能慢速转向，他们无法快速向上或向下运动。带有翅膀和推进器的怪物可轻松转向，不过，在完成这一动作时他们的身体会发生倾斜。在本游戏中，Sam 是唯一拥有火箭背囊的角色，这一设备将大大提高他的机动性，使其在面对众多敌手时占据极大的优势。此外，虽然这些怪物采用的飞行模式不必做到惟妙惟肖，但必须真实可信，只有这样，玩家才能感到 Sam 面对的敌人确实正在飞行。

3.5 目标寻找能力

探测到 Sam 的存在只是机器人面临的第一道难关。一旦发觉 Sam，那些头脑较为简单的机器人往往因为过于蠢笨而无法真正接近他。目标寻找能力的起伏非常显著，从最愚蠢到最聪明的机器人，不一而足。最愚蠢的机器人将采用一种“直线”技术，他们无法绕过那些挡住去路的障碍物。稍微聪明一点的机器人可以绕过去，但他们往往会挂在物品尖角上。最聪明的机器人和人类总是能设法接近玩家，他们可以开门，如有必要，也可以推开眼前的障碍物。为了取得胜利，玩家应该设法找到机器人在目标寻找能力方面的缺陷。

3.6 打击承受能力

只需一枚投射物便能将 Sam 面对的许多机器人和其他敌手击垮。较大一些的机器人可承受





多次打击，然后才会失去反击能力。例如，防护能力极强的电动机器人能经受水瓶的三次打击，然后才会出现短路。当然，不同的投射物在对付不同的敌人时能产生不同的效果，有些机器人或敌人可能对某种进攻毫不在乎。有关投射物的详细情况，见“游戏要素”标题下的“投射物”部分。

3.7 攻击能力

在“Atomic Sam”中，AI 主体可选用不同进攻类型使小 Sam 失去反击能力。在特定的情况下，许多敌人能选择多种进攻方式；例如，NPC 可短兵相接，也可使用投射物进行远距离进攻。NPC 能挑选最为有效的进攻方式，如果几种进攻方式的打击力不相上下，他们就会随机选择，或轮番使用这些进攻方式。

3.8 躲闪能力

由于 Sam 扔出的投射物以慢速飞行，因此较为聪明的机器人能避开来袭之物。当然，这些 AI 主体的躲闪能力并不会高超到玩家始终无法打中他们的程度，这种能力的存在只是为了给玩家提供一种有趣的挑战而已。

3.9 特殊动作

为了保持挑战的新鲜感和趣味性，还应设置各种特殊动作，但只有那些较为先进的机器人和人类敌手才能执行这些动作。这些动作将在游戏的后半部分出现，届时，为了赢得胜利，玩家不得不设法适应它们。

3.9.1 绑架

玩家与敌人之间的战斗通常在居民区内展开，为了给游戏增添色彩，应安排一些非敌方角色在这些地区四处行走。有些较为聪明的 AI 主体知道可以抓住部分 NPC，将他们扣为人质。这时，Sam 就应避免投射物打中这些人质。如果玩家操纵 Sam 飞近这些人质并按下“动作”键，他就能抓住这些人质，携带他们飞到安全区域。

3.9.2 内置维修臂

某些机器人在投射物的打击下会出现故障，一些更为复杂的机器人将有能力进行自我修复。常见的修复方式是，从机器人的某个隔室内弹出一只专用“维修臂”，维修臂曲伸自如，能抵达机器人的任何部位，焊接损坏部件。这一动作富于卡通效果，真实感并不强烈，只能给人一种机器人正在修复自己的印象。有些机器人受伤后可能会逃至比较安全的地方，如某个角落或远离 Sam 的去处。另一些机器人则具备多任务功能，他们能一边操纵维修臂，一边继续与 Sam 搏斗。

3.9.3 协作

某些敌人，特别是“冷面雇佣兵”，将懂得如何共同协作。许多机器人只能单独行动（攻击 Sam），他们并不知道其他机器人也会同时向 Sam 发动进攻。不过，那些智力极高的雇佣兵

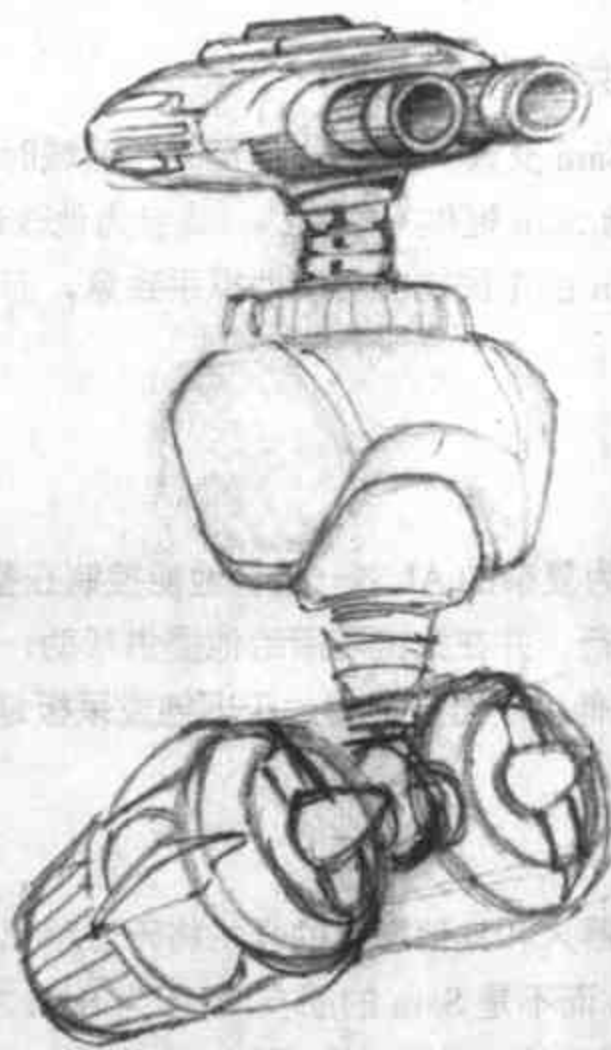




会认识到，共同协作能大大增强打击效果。例如，当某个雇佣兵正面进攻 Sam 时，其他人可绕到其侧面发动袭击。毫无疑问，由于敌人具备共同协作的能力，玩家将面临更大的挑战。

3.10 奚落

当 Sam 与这些敌人搏斗时，会听到他们贬损自己的话语，意思是，他们人多势众，绝无打败他们的可能：“认输吧，人类！”、“据统计，你不可能取得胜利”以及“钢铁比肉体坚固，住手吧！”并非所有机器人都能说英语，有些只能发出“嘀嘀”和“嘎嘎”声作为联络方式。另外一些机器人则更加残忍，他们会奚落 Sam，说他将永远无法重新见到自己的父母。



附图 6

3.11 圈套设置

在“Atomic Sam”中，游戏机制的很大一部分内容是，玩家利用自己所处的环境，启动各种圈套和机关，以便更为轻松地打败 Sam 面对的机器人。其实，AI 往往会帮助玩家有效利用这些圈套，因为那些机器人的智力并不高明。此外，设计者还可对这些敌手进行设置，使其倾向于进入玩家能设法制服他们的地方。例如，当玩家可以撬动杠杆并弹出置于其上的空油桶时，机器人往往会飞近空油桶的弹射轨迹。

3.12 非战斗主体

Sam 所到之处也是那个世界内的人们生活和工作的地方。因此，这些地方不仅被那些派去





追捕他的敌人占据，也应该居住着一些普通市民。这些市民并不十分聪明。在各关卡内插进这些角色不是为了创造一种“真实”的环境氛围，他们的存在主要是为了给游戏增添色彩，同时也能给 Sam 制造一些麻烦，即在投射时避免误伤这些市民。

3.12.1 逃避

通常，一旦出现危险，这些市民就会四下逃窜，寻找藏身之所，以避免 Sam 和机器人之间的战斗。当然，对未来世界中的人们来说，出现能飞的机器人或一名身着火箭背囊的小男孩并不能激发他们的好奇心，因为这些都是他们司空见惯的景物。只有当战斗打响后，这些市民才会意识到自身所处的危险境地。关卡设计者能对这些市民的行走路线以及他们试图躲避的安全场所进行设置。

3.12.2 与 Sam 交谈并提供帮助

当然，某些市民将乐于和 Sam 交谈。当 Sam 行至某一区域时，某些市民可能会向他介绍这一区域的情况。有些人甚至会给 Sam 提供一些物品，或者为他改进火箭背囊的性能。当 Sam 飞临某地时，那些有能力帮助 Sam 的市民往往会朝他挥手致意，而其他市民则只能给游戏环境增添色彩和变化。



3.13 朋友

Atomic Sam 必须具备的最为复杂的 AI 之一是，他能控制在整个游戏过程中遇到的朋友们。这些游戏主体必须能与 Sam 同行，并在关键场所给他提供帮助，他们从不会迷路或陷入困境。制作能帮助玩家的队友 AI 并让他们不至于显得过于迟钝或呆板是一件非常困难的事情，不过，这样做将大大改善游戏效果。

3.13.1 无敌

Sam 在各关卡中发现的机器人和其他恶意的怪物将无法杀死或抓住那些跟随他的朋友们。首先，敌方怪物倾向于攻击 Sam 而不是 Sam 的朋友们，因为 Sam 才是他们受命前来制服的对象。其次，朋友 AI 主体将有能力抵御任何来犯之敌。同样，如果 Sam 不小心将投射物砸向一位朋友，这位朋友能轻而易举地将其挡开，并提醒他“别乱扔那些东西！”如果朋友 AI 可以被打败，而且这一结局将对游戏造成某种影响，那么，游戏结构就会变得极其复杂和难以处理。不过，可以让他们暂时昏迷，然后在几秒钟内完全恢复到协助状态。

3.13.2 跟随 Sam

对这些朋友 AI 主体来说，最为重要的任务是跟随玩家穿过关卡。这就意味着，这些朋友必须能不露丝毫破绽地跟随玩家通过那些可能非常复杂的 3D 环境，因为正是这些 3D 环境构成了“Atomic Sam”的游戏世界。如果玩家偶尔转过身来，发现一位朋友被远处某个结构物卡住，那么，游戏将无法继续。

NPC 不必总是紧紧跟随 Sam。事实上，这些具备飞行能力的朋友们可自由出入画面，这样，即使他们并非一直出现在屏幕上，玩家也会觉得他们始终在附近活动。有时候朋友们跑在 Sam 前面，有时候则跟在后面，不过必须始终靠近 Sam。





3.13.3 殿后

Sam 的朋友们将在游戏中起到关键性的作用，他们会在 Sam 毫无觉察的情况下指出敌人的袭击方向：“注意，Sam，当心背后！”在某些情况下，当敌人尚未接近 Sam 时，这些 AI 主体可采用自己的进攻方式或扔出投射物，协助 Sam 打败敌人。不过，无论在什么情况下，这些主体的进攻能力远远逊色于 Sam。重要的是，玩家仍必须亲自与机器人战斗，而不能坐享其成，让朋友们替自己对付所有敌人。

3.13.4 提供建议

同样，在“Atomic Sam”中，当敌人来犯之时，这些朋友们也能为玩家提供相关建议：“看来这个家伙挺厉害！”或“我想水瓶对付不了那个家伙的！”在关卡内的某些情况下，朋友们可为 Sam 指出一些秘密区域或一组投射物，否则，Sam 也许无法觉察到它们。玩家可操纵 Sam 靠近某位朋友，然后按下“动作”键，这时，朋友必将做出答复。有时候，这些答复并无多大意义：“Sam，见到你很高兴”或“刚才你对付的那个机器人可挨得不轻呢！”有时候，玩家可操纵 Sam 与朋友们交谈，以便得到一些暗示：“走左边那条岔道吧，它更近一些”或“最好扔一些粘性物品来对付这些正在爬坡的机器人。有这种物品吗？”

3.13.5 故事叙述

除了给 Sam 提出一些小小的建议外，朋友们还担负着向玩家传达故事内容的重任。当 Sam 抵达关卡的某一部分时，其中一位朋友可能会开始讲述这一地区的历史或者他们本人的经历。这样，就能以非片头动画形式为游戏提供额外的故事内容，因为，当 Sam 倾听朋友讲述时，他仍然在游戏环境中活动。这些朋友们非常聪明，他们只会在“安全”环境下，即 Sam 未遭受敌人频繁进攻时打开话题。



4 游戏要素



4.1 物品

4.1.1 Sam 的投射物

当 Sam 在关卡中飞行时，他可以捡起各种不同类型的投射物，然后利用这些物品打败敌人。不同类型的投射物在不同情况下对付特定的敌人时，发挥的作用有大有小，因此，在任何时候，玩家必须坚持选用最具威力的投射物。各类投射物的性能列举如下：

- **粘性球：**由粘性物质制成的淡绿色球体，它可将那些在地面行走或在墙壁上爬行的怪物粘附在地表或墙壁上。怪物可能会暂时失去活动能力，也可能永远无法动弹，具体情况取决于怪物力量的大小。
- **水瓶：**水瓶能使暴露在外的电路发生短路，从而使机器人丧失活动能力。如果机器人具备防护罩，则必须打击多次才能使其短路。
- **磁性块：**一种磁性极强的重物。当机器人处于飞行状态时，磁性块能吸附在他们的金属





表面，将其拽到地面。

- **弹簧笼**：一种黑色立方体，表面伸出六根金属棒。撞击目标后，弹簧笼会自动伸展并围住目标，将其圈闭在一只结实的笼子里。对付体形微小的飞行敌人效果最佳，体形较大的敌人将有能力破笼而出。
- **电磁干扰器**：一种小球体，扔出后可飞行一段距离，然后自行启动，使一定半径内的电动设备失去活动能力。处于飞行状态中的机器人会垂直坠地，攀附在墙壁上的机器人会失足跌落，地面机器人则会嘎然而止。电磁干扰器对人类和原子能机器人毫无作用。如果玩家持有电能比拉鱼（详见“游戏机制”部分），使用电磁干扰器时必须多加小心，因为这种装置也会使 Sam 的比拉鱼停止工作并坠地。
- **泡泡棒**：类似于孩子们吹泡泡时使用的泡泡棒或泡泡圈。不过，这种泡泡棒产生的泡泡结实得多，它们能裹住目标，使其至少在数分钟内无法逃脱。在 Sam 的“可投射”物品中，泡泡棒是较为理想的武器之一；不过，它无法对付那些持有利器、尖物或推进器的敌人。
- **原子流星锤**：本游戏中最具威力的投射物之一，外形与普通流星锤相仿：一根金属线上接着两个黑色球体。不过，这种流星锤具有能量，当它绕住某个目标时，两个球体内的引擎开始工作，使金属线不断绕向敌人，将其紧紧缚住。原子流星锤无法对付任何有飞行能力的敌人，因为他们带有推进器或轮转叶片。

4.1.2 火箭升级

在整个游戏过程中，玩家可对 Sam 的火箭背囊做出各种改进，要么请一位 NPC 对背囊进行修补，做出改进，要么会让 Sam 找到一些附件，然后自行安装。这些升级将大大提高 Sam 的能力。

- **主爆发器**：主爆发器是对背囊的简单改进，当玩家使用背囊速度爆发功能时，主爆发器产生非常高的速度。
- **加速器**：加速器可提高火箭背囊的总体速度，相应地，Sam 的旅行速度将比升级前有所提高。
- **Gyromatic**：当 Sam 使用火箭背囊时，Gyromatic 将使飞行更加稳定，停止和起飞更为迅速，以消除那种慢悠悠的停止动作。Gyromatic 是一种简单的“敲入”式附件，Sam 可轻而易举地自行安装。
- **原子压缩器**：一种结构简单、带有刻度盘的盒形装置，可安装于背囊侧面，给 Sam 提供更长的飞行时间。这种装置采用独特的方法“压缩”背囊内持续产生的能量，从而随时给背囊储存更多的能量。

4.1.3 其他装置

在“Atomic Sam”中，Sam 还能捡到其他一些五花八门的装置。这些装置的用途各不相同，但它们均可提高 Sam 在各关卡旅行和生存的能力。

- **电能比拉鱼**：在整个关卡中，Sam 将发现大量的电动比拉鱼，这种小小的装置能在 Sam 周围的空中“游荡”，化解对 Sam 的进攻。电能比拉鱼的用途在“游戏机制”部分有详细介绍。
- **Spidersonic**：利用 Spidersonic 套件，Sam 可像蜘蛛一样贴附在任何垂直面上。借助





Spidersonic 套件，在飞往下一个地点之前，Sam 可以抓住某座楼房的侧面，同时停止飞行，让火箭背囊补充能量。

- **月球服：**Sam 将在 New Boston 找到这种使用方便的月球服。穿上它，Sam 即可在月球城外旅行，经受月球表面的恶劣环境。幸运的是，Sam 的火箭背囊和工具带均可套在月球服的外面，这样，Sam 就能继续飞行和投射；不过，这两种动作将在不同程度上受到月球引力的影响。

4.2 角色

在“Atomic Sam”中，Sam 会遇到形形色色的角色。这些角色包括 Sam 的朋友、支持者、敌人以及绑架其父母的人。

4.2.1 Atomic Sam

玩家操纵的 Atomic Sam 是一位十岁的小男孩，在整个游戏过程中，他身着火箭背囊，利用自己的聪明才智避开无数机器人和人类敌手的袭击，更不用说还要穿过一些危机四伏的地方了，所有这一切都是为了找到他的父母。Sam 身高三英尺，身穿一条棕色马裤和一件红色飞行服，其中后者镶有金色的饰物。此外，他还有一条棕色的皮带，皮带上系着用途各异的小袋。Sam 脚穿一双硕大、笨重的银色“月球鞋”。背上套着一台赖以飞行的原子能火箭背囊。这是一种小巧紧密的装置，比他的肩膀窄几英寸，位于皮带和脖子之间，两端一共留有几英寸的空隙。Sam 头发黑短，戴着一不副 1930 年式样的飞行护目镜。Sam 的能力在本设计文件各部分均有描述。Sam 的个性与大多数前途光明的十岁男孩相仿：乐观、聪明。不过，这是 Sam 有生以来第一次离开父母，因此，面对必须独自探索的世界，他多少有点不知所措。



附图 7





4.2.2 朋友

- **Xeraphina:** 在 Benthos, Sam 将遇到一位名叫 Xeraphina 的十二岁女孩。Xeraphina 是一对艺术家的女儿, 从小在 Benthos 长大, 从未去过地面, 那儿是她极其向往的地方。Xeraphina 能利用父母发明的一种独特的翅膀在城市上空滑翔, 并协助 Sam 与机器敌手搏斗。Xeraphina 身穿一套轻便的紧身制服, 披一条半透明的绿色披肩, 当她在空中飞行时, 披肩就会随风飘舞。她的翅膀由透明度更差的物质制成, 颜色为暗绿色, 翼展达八英尺。翅膀结构牢固, 安装在她的肩胛骨上, 当她飞行时, 翅膀就会轻轻扇动。她的脸上总是挂着亲切的微笑, 棕色的长发在脑后挽成圆髻, 插上一只小小的画笔, 以免散开。
- **Scrap:** 在 Harmony, Sam 将邂逅 Scrap, 一台几星期前制成的、簇新的机器人。Scrap 是一个和蔼可亲的家伙, 一有机会, 他便乐此不疲地用自己的尖嗓门讲起笑话来, 双关语是他的拿手好戏。在许多方面, Scrap 的举止就像一个模仿十岁小孩制成的机器人, 他害怕有一天会被派去从事某项工作, 尽管他对具体情况一无所知。Scrap 对自己的机器人身份颇感满意, 不过, 他希望自己永远不会“长大”, 并梦想着周游世界的生涯。Scrap 的身高与 Sam 不相上下, 体形像人, 不同之处在于, 他拥有四只胳膊和一个小得出奇的脑袋。利用自己弹簧单高跷似的双腿, Scrap 能跳过很远距离, 他会想方设法协助 Sam 打败他的机器人敌手。
- **Dulo:** Dulo 是 Sam 父母的助手。他的外貌大致像 Torso Moonie, 这一点在接下来的月球人介绍中有详尽的描述。与众不同的是, Dulo 经常佩戴着自己的心爱之物, 一双特制的紫色手镯, 这一点有助于将他与别的月球人区分开来, 因为其他月球人——至少在躯干型或双头型月球人内部——都长得非常相似。此外, Dulo 还能四处蹦跳, 帮助 Sam 打败机器人敌手; 利用自己长长的触须, 他能轻而易举地将机器人从空中拽下, 在月面上摔得粉身碎骨。

4.2.3 其他角色

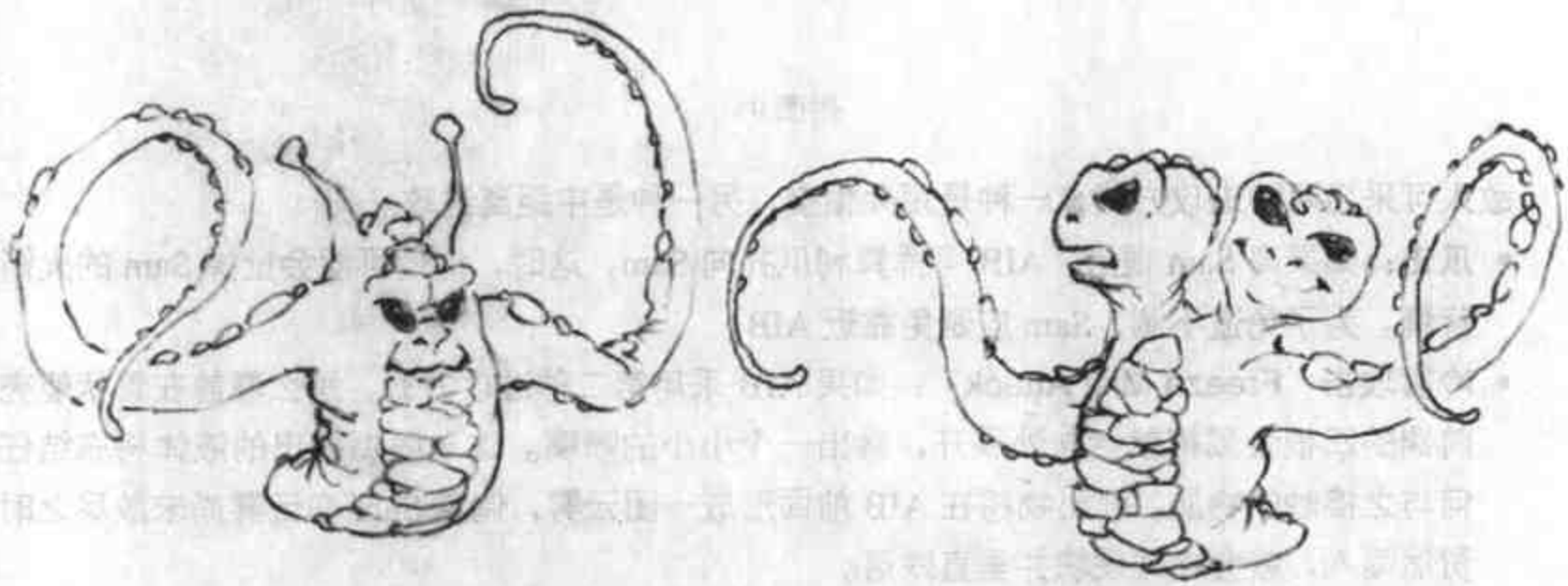
- **Electric Priestess:** Electric Priestess 就是那位帮助 Sam 了解父母下落的神秘女人, 她给 Sam 提供了不少有关那一世界的宝贵信息。直到故事结尾, 玩家才会明白这位 Priestess 原来就是 Max Zeffir 的妹妹, 同时也是他的首席研究员之一。一次, 由于 Zeffir 的疏忽大意, 他们乘坐的齐柏林飞艇发生意外, 在这次事故中, Priestess 失去了一条腿。Electric Priestess 仍然爱着自己的哥哥; 不过, 由于 Zeffir 变得日益贪婪, 疯狂地聚财敛富, Priestess 对他也不无反感。Priestess 身穿一件翠绿色长袍, 一顶大黑帽遮住了她的部分脸庞。她只剩下一条腿, 另一条由笨重的机械假肢代替。
- **Ike:** 在 Harmony 市, Sam 将遇到 Ike, 一台陈旧的机器人, 多年以前, 这台机器人曾是他父母珍爱的助手。同许多主人不同的是, 当 Ike 衰老后, Sam 的父母让他退休, 而不是将他关闭, 因此, 他得以回到 Harmony, 与其他机器人一起安度晚年。正如“游戏过程”部分所述, Ike 是一位记忆力逐渐衰退但仍然十分聪明的机器人。Ike 话语不多, 但如果记忆力得到激发, 他就会打破沉寂, 向 Sam 的父母表达无比的爱戴和尊敬之情。比起 Sam 遇到的其他机器人, Ike 的设计显得有些陈旧: 方正、笨拙的体形, 以及比许多新型机器人如 Scrap 大得多的结构。此外, 由于年事已高, 他的行动速度也相当迟缓。Ike 靠坦克





履带行走，依照设计，只有一只胳膊，这只长长的、由五个关节组成的手臂连接在胸部中央的躯干上。

- **Tool:** Tool 就是那位“机器人医生”，为了保住 Ike，Sam 必须在 Harmony 找到这位“机器人医生”。Tool 是一台体形巨大的机器人，从外表上看，他是一个极其凶狠、残暴的家伙。但事实上，他非常和善而且富于同情心，大有“软心巨人 (gentle giant)” 的风范。Tool 不能说话，只能通过胸前显示屏上的文字进行交谈。利用扎在腰部的抗重力设备，Tool 能脱离地面，做短暂的飞行。给机器人“动手术”时，他不会使用那双安装在上半身的巨臂和拳头。这时，位于胸部的一个小隔室会自动弹开，伸出一些小巧、细长的机械臂，开始精密操作。
- **月球人:** “月球人”（这是地球人对他们的称呼）的身高一般约为四英尺，利用身体下半部分（他们没有腿），他们可随意蹦跳。他们的前臂是一对修长、有力的触须，分别位于身体两侧，这对触须相当灵巧，可握笔写字。虽然月球人是一种无性生物，但他们也分为两种类型，每一种都具有不同的身体特征：一类月球人长着两个脑袋，与人类一样，它们都位于身体上部（地球人称之为“双头人”）；另一类月球人的眼睛和嘴巴都长在躯干上（地球人称之为“躯干人”）。此外，月球人身上还有一些白色的凸起，需要时，这些凸起可放出光芒，帮助他们穿过黑暗地区。这些月球人始终生活在月球背面，因此，离开这种照明功能，它们便无法行动。



附图 8

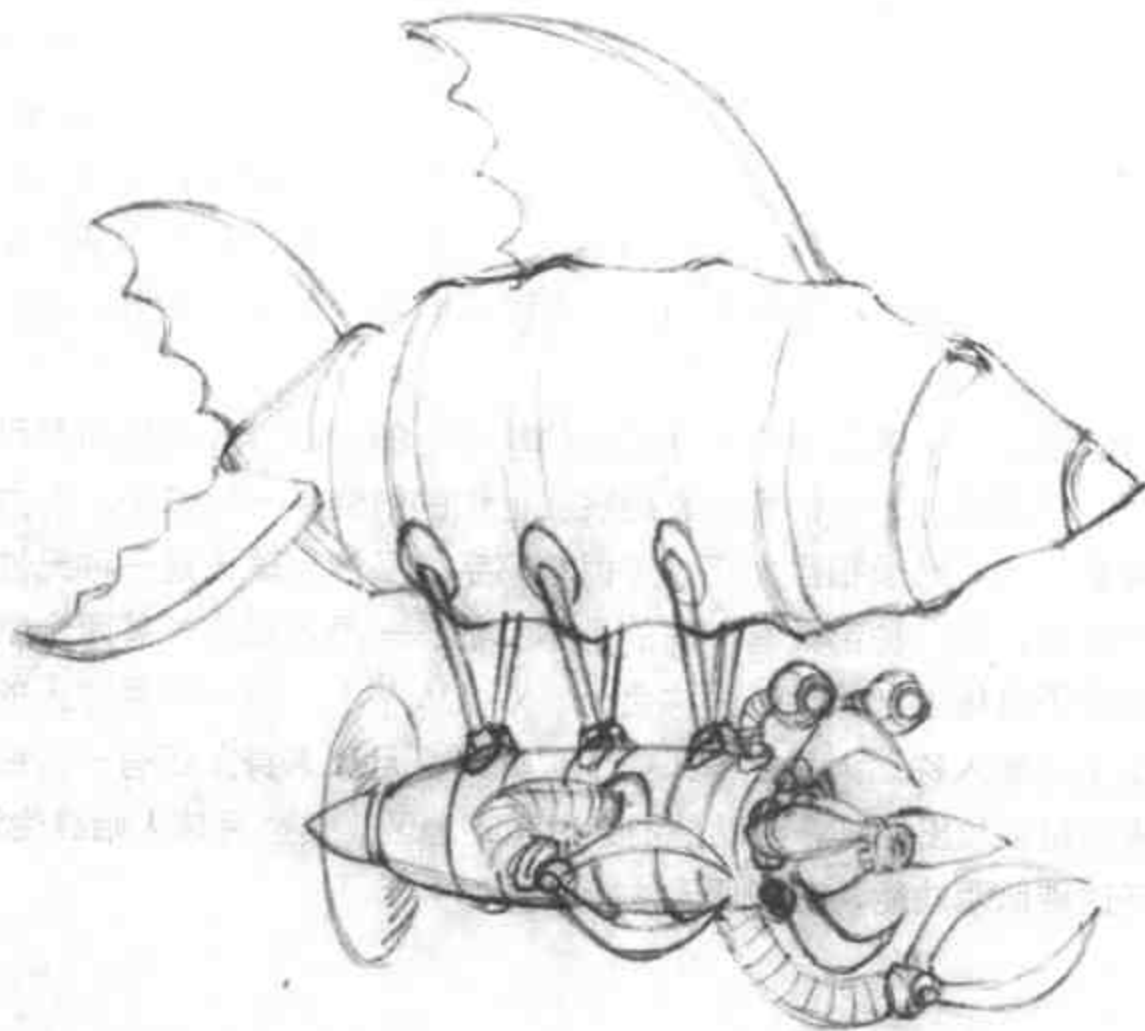
4.2.4 敌人

Arctic Immobilizer Blimp 在 Sam 必须面对或避开的敌人当中，Arctic Immobilizer Blimp (AIB) 是一种中下难度的机器人敌手。AIB 的外形像一个圆柱体，由银白色金属制成，他们悬挂在一艘微型齐柏林飞艇之下，能以较慢的速度在空中飞行，动力来自飞艇尾部的一台小型推进器。从 AIB 的管状躯壳两侧伸出两只金属爪，他们会在 Sam 面前气势汹汹地舞动这些利爪。管状躯壳的前端装有四根金属棒，封住前端开口。AIB 能上下运动（慢速），以便对齐 Sam 后展开进攻。不过，由于 AIB 行动迟缓，一旦 Sam 成功逃脱，他们将望尘莫及。此外，由于 AIB





可以飞行，Sam 将无法使用粘性球攻击他们。对付 AIB 时，最为有效的投射物是磁性块，这种磁性块可以迅速击落敌人。



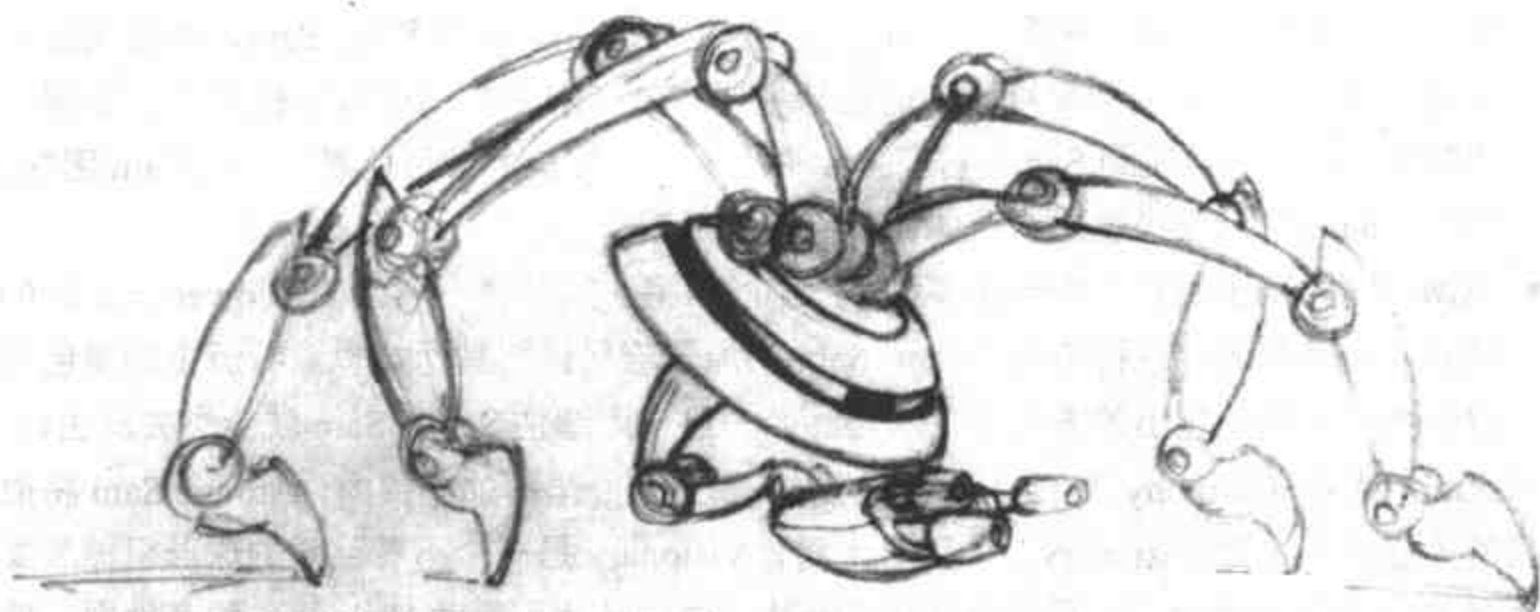
附图 9

敌人可采用两种进攻方式，一种是短兵相接，另一种是中距离进攻。

- **爪击**：如果离 Sam 很近，AIB 可挥舞利爪扑向 Sam，这时，他们可能会扯掉 Sam 的火箭背囊。为了免遭不测，Sam 应避免靠近 AIB。
- **冷雾攻击 (Freeze Mist Attack)**：如果 AIB 采用第二种进攻方式，那么覆盖在管状躯壳前端的四根金属棒就会向外展开，露出一个小小的喷嘴。这一喷嘴射出的液体将冻结任何与之接触的物质。喷出物将在 AIB 前面形成一团云雾，倘若 Sam 在云雾尚未散尽之时贸然闯入，就会冻成硬块并垂直跌落。

Arachnaught Arachnaught 是一种比较容易对付的机器敌人。Arachnaught 的外貌类似于四条腿的蜘蛛，每条腿由三节肢体组成，末端尖利。四条腿均连接在一个较小的躯干上，躯干内装有一台曲面视觉传感器，因此，这一怪物拥有良好的远视能力。Arachnaught 无法飞行，但能轻松自如地攀楼爬壁，就像在平地上行走一样。Arachnaught 步履如飞，能以极快的速度行走。由于它是一种在地面爬行的机器人，因此，Sam 无法用适合于打击飞行敌人的投射物来对付它们。不过，在粘性球的打击下，他们将束手无策。





附图 10

Arachnaught 可采用三种进攻方式，即短兵相接、投射和短距离“拖曳光束”。

- **爪击**：Arachnaught 能用自己的利腿展开攻击，这种装置能轻而易举地割下 Sam 的火箭背囊，从而使 Sam 丧失行动能力。
- **粘性蛛网球（Sticky Web Ball）**：Arachnaught 能发射大团粘性极强、运动缓慢的物质。如果 Sam 被这种球状物击中时位于地面，他就会粘附在地面上，无法动弹；倘若 Sam 位于空中，就会暂时失去投射能力，因为这时他必须努力挣脱这种粘性物质的束缚。此外，如果 Sam 带着蛛网球撞在任何物体上，他就会粘附在这一物体的表面，同时失去活动能力。
- **网绳**：Arachnaught 手中最为凶险的武器要属它的网绳装置。利用这种装置，Arachnaught 能向 Sam 发射一条长长的网绳，一旦击中目标，就能将 Sam 拽到自己跟前，然后伸出利爪，扯下 Sam 的火箭背囊，使其不便于行动。这时，Sam 必须朝与网绳正好相反的方向飞行，挣扎五秒钟后才能破网而出。

冷面雇佣兵（Merciless Mercenary） 尽管在本游戏中 Sam 的许多敌手都以各种类型的机器人结构出现，但在 Max Zeffir 的飞行堡垒 Ikairus 内，他也会遭遇一些人类敌人。这些被称为“冷面雇佣兵”（MM）的人类训练有素，Sam 将难以躲避或制服他们。这些 MM 身穿带有红色镶边的黑色军服，头戴一顶杀气腾腾的钢盔，可以利用系在腰间的一条抗重力带在空中飞行。重力带能帮助他们浮在空中，不过，为了产生推进力，MM 必须像游泳一样不停地划动。Sam 的许多投射物将无法对付这些 MM，惟一有效的武器是原子流星锤；原子流星锤可以缠住 MM 的双腿，使其无法继续“游动”。MM 是 Sam 所遇到的较富于机动性的敌手之一。由于 MM 具备跟踪 Sam 的能力，而且他们的运动速度几乎与 Sam 不相上下，因此，一旦遇到这类敌人，Sam 将很难逃脱。

MM 一共可采用三种进攻方式：一种是短兵相接，另外两种为远距离进攻。

- **Tri-Power 三叉戟**：MM 持有一种金色的金属武器，正式名称是 Tri-Power 三叉戟，具备两种功能。第一种是简单的短兵相接，用于 Sam 靠得太近之时。Tri-Power 三叉戟的尖端可轻而易举地将 Sam 的火箭背囊扯下。
- **Tri-Power 三叉戟的远距离攻击能力**：Tri-Power 三叉戟的第二种进攻方式为，向空中发





射一种外形硕大、运动缓慢、放出蓝色光芒的能量团。一旦靠近 Sam，能量团就会停止运动，炸裂成六个小型能量球。这些从主球体分裂出来的小球将以极快的速度漫无目的地向外迸射，一旦碰到 Sam，它们就会膨胀成一个完美的球形能量团，将 Sam 困在其中。这时，Sam 将无法挣脱这一球体，从而丧失活动能力。

- **电网：**雇佣兵还拥有一种电网装置，他们能用另一只胳膊（未持 Tri-Power 三叉戟的那只胳膊）将其扔出。这种电网类似于 Sam 的原子流星锤，其两端也系着沉重的黑色球体，以便拽住电网朝扔出的方向前进。当然，一旦电网缠住 Sam，Sam 就会失去反击能力。

Visionary 在 Harmony 市的尽头，即机器人制造中心坐落的小镇内，Atomic Sam 将遭遇一个凶猛的关底怪兽，即 Visionary。从外表上看，Visionary 是一个瞪着巨眼的球形机械装置，它的两边各有一个金属结构，上面装有直升机叶片。这些叶片可帮助 Visionary 脱离地面，使其具备很强的机动能力。Visionary 能上下晃动，其运动速度比 Sam 的火箭背囊快得多；不过，它的转向速度比 Sam 稍慢一些。

从眼球底部伸出三根钢制触须，每根触须末端均附有各不相同的机械装置，每种装置将对应 Visionary 的三种攻击方式之一。

- **电动刀：**其中一根触须装有三枚旋转刀片，它们像爪子一样对着同一方向。这些刀片不停地旋转，令人不寒而栗。不过，真正令人担心的是，这些刀片能发射一种电能冲击波，将 Sam 击昏。刀片飞速旋转时，其中心部位能释放冲击波。这种远距离光束式攻击方式将令玩家防不胜防；因此，一旦觉察到 Visionary 的刀片开始高速旋转，就必须加倍小心，操纵 Sam 逃离刀片所指的方向。
- **磁铁：**第二根触须的末端装有一枚巨大的 U 形磁铁。Visionary 能启动磁铁，吸引 Sam 火箭背囊上的金属，将 Sam 拉到自己跟前。Sam 必须想尽一切办法避免靠近这一敌人，否则，Visionary 就能利用带有刀片的触须将 Sam 的火箭背囊扯下。
- **烟雾：**第三根触须的末端装有一个巨大的漏斗。Visionary 可从这一漏斗喷出浓黑的毒气，一旦 Sam 不幸闯入，就会咳嗽不止（游戏也将因此而结束）。Visionary 喷出的烟雾会在空中停留一段时间，因此，玩家必须多加小心，以免飞进尚未散尽的烟雾之中。

就像对付这款游戏中的所有关底老怪一样，Sam 的大多数常规武器（投射物）将无法有效对付 Visionary。这些常规武器也许能在短时间内减缓机器人的活动速度，但无法彻底打败它。为了制服 Visionary，玩家必须设法利用关卡的内在机制。这样，无需多少技巧、重复和耐性，只要深入了解谜题，便能打败这一怪物；不过，一旦悟出谜题，就很难重复这一过程。由于玩家与 Visionary 展开搏斗是在 Harmony 的最后一关，因此这场战斗将发生在一座机器人制造厂内。Visionary 将从固定在游戏区顶层天花板上一个储存箱内出现，在那儿，Sam 将与这台机器人展开搏斗。这一地区分布着机器人工厂内必备的各种有用设备以及四台高能风扇。Sam 可靠近这些风扇，然后利用自己的“动作”能力打开它们。Sam 还可利用“动作”键改变风扇的旋转方向。启动储存箱下的所有风扇并使风向朝上后，Sam 能得到一股旋风，将 Visionary——由于它凭借直升机叶片始终位于高空——推回到箱子里面。这时，Sam 可扳动箱子近处的开关，关闭箱门，将机器人困在里面，从而取得战斗的胜利。

Max Zeffir Zeffir 是 Zeffir Zoom 公司的创始人和拥有者，是这个星球上最富有的人。Zeffir 靠一家生产拉链的公司即 Zeffir 拉链公司起家，随后他竭力转向几乎所有其他行业。他的公司包





括飞机制造公司 Zeffir Zeppelins、服装公司 Zeffir Zest 以及 Zeffir Zeitgeist 新闻网等。

此外，Max Zeffir 还是 Sam 父母的雇主，而且，情况表明，正是他绑架了 Sam 的父母，目的是为了使他们保持沉默。在游戏结尾，Sam 将最终与 Zeffir 展开决斗；在这场战斗中，Zeffir 将证实自己是一个十分强劲的对手。

当玩家最终与 Zeffir 相遇时，他身着一套 1920 年“铁路大亨”特有的黑色细条纹西服，头戴一顶超大号的大礼帽。Zeffir 蓄着一撮又黑又细的小胡子，脸上挂着不怀好意的奸笑。Zeffir 将在他的 Negativity Platform 上与 Sam 展开搏斗，这一平台之所以被称为 Negativity，是因为它能抗拒重力效应。反重力平台就好像在空中飘浮着的一个圆盘，它比 Sam 的火箭背囊更加富于机动性，围栏的高度与 Zeffir 的腰部平齐，以便平台飞动时用做扶手。

与玩家搏斗时，Zeffir 可采用两种攻击方式：

- **机器人：**Zeffir 可召集机器人与玩家即 Sam 展开搏斗。这些机器人包括 Sam 在整个游戏过程中接触到的所有类型的机器敌人；当 Sam 与 Zeffir 在一座宽阔的穹顶大厅内对峙时，这些机器人将从遍布于这一地方的各种隔室内走出。就像在游戏的其他部分一样，Sam 必须打败这些机器人。Zeffir 一次可召集三个机器人敌手。
- **音叉：**Zeffir 还可使用一种六英尺长的类似于音叉的装置。当他手持这种棒状物在 Negativity Platform 的栏杆上敲击时，音叉就会产生一种声音冲击波。一旦 Sam 被这种冲击波击中，他就会暂时昏厥，被迫停止飞行，并抬起双手捂住耳朵。这时，由于 Sam 失去活动或投射能力，因此，在机器人的攻击下，他将显得异常脆弱。

玩家可使用各种不同的计策打败 Zeffir，这些计策可采用不同的组合方式。

- **持久战：**如果 Zeffir 被大量有效投射物击中，他就会遭受失败的厄运。不过，只有部分投射物能有效对付 Zeffir，磁性块和弹簧笼都不怎么理想，另外一些也只能逐渐拖垮 Zeffir。打败 Zeffir 需耗费大量的投射物，同时，Zeffir 也将竭力召集更多机器人，并用自己的音叉打击 Sam。因此，这是一种最为艰难的取胜方式，但同时也是一种最为浅显的方式。
- **破坏反重力柱：**事实上，Zeffir 的反重力平台由地板上的四根反重力柱 (Negapost) 支撑，这些反重力柱分别位于大厅内的四个角落。当 Zeffir 在平台上活动时，这些反重力柱就会放出光芒。Sam 可破坏其中一根柱子，方法是向它连续投射三个水瓶。一旦这根柱子无法工作，飞船将暂时失控；不过，Zeffir 会很快使其恢复正常。Sam 必须破坏所有反重力柱，只有这样，反重力平台才会真正停止工作，并哗啦一声摔在大厅地板上，与此同时，Zeffir 将束手就擒。
- **抢夺听觉保护器：**玩家会注意到，Zeffir 戴着一对硕大的“听觉保护器”，这种装置的外形与耳机相仿，不过，它的作用是阻挡音叉发出的有害声音。细心的玩家会了解到，如果用力敲击反重力平台，听觉保护器就会从 Zeffir 身上震落。不过，仅仅用投射物打击 Zeffir 是无法使其震落的。玩家必须诱使 Zeffir 操纵反重力平台撞向体积更大、悬挂在大厅穹顶下的一根晃动的大梁。如果 Sam 在 Zeffir 飞向大梁的同时设法推动它，那么，撞击产生的震荡就会干净利落地将听觉保护器震落在地。这时，倘若玩家操纵 Sam 降落到听觉保护器的落地之处，就能使他戴上这一装置。于是，Sam 将不必担心 Zeffir 的打击。如上所述，采用持久战或者破坏反重力柱，玩家即可轻而易举地打败他。





5 故事简介

“Atomic Sam”讲述的是一位第一次离开父母的小男孩，这位小男孩将历尽艰辛，寻找双亲的下落。尽管“Atomic Sam”的核心内容与动作兼冒险类游戏相似，但其幽默、动人的故事使之有别于其他许多由控制器操纵的动作游戏。

“Atomic Sam”的时代背景是地球的未来，但与我们现在想象的未来世界并非完全一致。对这一未来世界的预测出现在20世纪上半叶的一些杂志如《Popular Science》和《The Electrical Experimenter》上，此外，当时的一些未来学家如Norman Bel Geddes和Buckminster Fuller对此也有过描述。某些我们现在看来司空见惯的创新成果，如喷气式飞机，在那一时代从未得到应用，人们乘坐的仍然是巨大的螺旋桨飞机和齐柏林飞艇。同样，对于个人计算机以及互联网，人们也是闻所未闻；不过，智能高超、乐于助人的机器人倒是随处可见。人类甚至已经在月球上建立殖民地，并发现生活在那儿的地外生命，即Moonie。“Atomic Sam”的故事就发生在这样一个怪诞、新奇的世界里。

一天，小Sam放学后回到家里，发现父母已神秘失踪。Sam的父母均为Zeffir Zoom公司（一家运输公司）的科学家，他们总是在Sam放学时呆在家里，这是他们从不违背的习惯。心烦意乱的Sam决定去寻找自己的父母。他穿上一件红夹克，披上原子能火箭背囊——父母送给他的生日礼物，并改名为Atomic Sam，试图以新的精神面貌唤起自己的勇气。

Sam穿过Gargantuopolis市前往父母的办公室，但沿途受到一些机器人的攻击，这些机器人试图阻止他的去路。Sam终于来到父母的办公室，但在那儿他同样未能找到他们，只看到一张神秘的字条。很快，一名友好的机器人走了进来，它将Sam护送到隔壁一座高耸入云的大楼里。Sam升至最高层，见到一位自称Electric Priestess的陌生女子，她告诉Sam，尽管她并不知道他父母的下落，但她会帮助他找到他们。她建议Sam前往三个地方，在那儿他可能会找到自己父母的下落。

Sam将抵达Benthos，一座海底城市。在那儿，他将邂逅Xeraphina，一位能飞的女孩，她会帮助Sam找到他父母的私人办公室。下一个地方是Harmony，这是一座机器人城市。在那儿，Sam将努力寻找Ike，多年以来，这名机器人曾经是他父母的忠实助手。途中，Sam会遇到Scrap，一名年轻、勇敢的机器人，这个机器人很古怪，他不想“长大”，也不愿意工作。最后，Sam来到月球移民城市New Boston，寻找父母的另外一位朋友，月球人Dulo。在每一个地方，Sam都会遭到一些机器人的无情打击，它们试图打败Sam，阻止他的调查。当Sam彻底搜索每一个区域后，他便会找到蜡筒的所有部件；将所有零件组装在一起后，蜡筒就能播放出有关Sam父母下落的内容。

这台wax cylinder内录有Sam父母的警告：雇主Max Zeffir正在开发一种单轨铁路系统，这期间，他们偶然发现这种系统存在着安全隐患。但遗憾的是，由于代价过于昂贵，Zeffir并不打算排除这一隐患。于是，他绑架了Sam的父母，以免他们揭露这一情况。这时，Electric Priestess将借给Sam一架直升机。乘坐这架直升机，他便能登上Zeffir的巨型飞船Ikairus。在那儿，Sam将与更多机器人展开战斗，然后才会直接面对并打败Max Zeffir。最后，Sam终于和父母团聚。





6 游戏过程



6.1 故事背景

“Atomic Sam”发生在未来世界，具体时间没有确定，也许是21世纪。它并非当今即2000年的人类所预想的那个未来世界，而是1920年、1930年和1940年的美国人的乐观预测。“Atomic Sam”的世界充斥着齐柏林飞艇和“巨翼”螺旋桨飞机，而不是那种穿梭于各大陆间的喷气式客机。在“Atomic Sam”中，核能并未像20世纪下半叶那样受到冷落，它满足了人们的热切期望，成为一种廉价、洁净的能源；由于精炼程度极高，它可以安全地应用于儿童玩具或齐柏林飞艇。

在这一未来世界里，人类已通过技术彻底消除了贫穷，摩天大楼达到前所未有的高度，任何人都有足够的空间安居乐业。私人飞机和旋翼飞机（飞机和直升机的混合体）已不再是什么稀罕之物，许多飞机就在高耸入云的摩天大楼上起降。在那儿，铁路运输发挥着极其重要的作用，与齐柏林飞艇相比，高速单轨铁路能以更快的速度往来于城市之间。

智能机器人随处可见。人们买到机器人后，要么让它们在工厂劳动，要么吩咐它们管理家务。不过，这些机器人并未使人们陷入失业的困境，它们给人类带来更多的业余时间，同时也增加了每个人的财富。与近20年来许多科幻小说描述的那种荒凉、混乱的未来世界截然不同，这是一个欣欣向荣的世界，在那儿，技术解放了人类，使他们过着幸福的生活。

不过，机器人技术的进步也带来了一些有趣的问题。在这一时期，机器人的智力水平基本上与最聪明的人类不相上下，发达的智力使它们具有了自己的感情和欲望。当然，机器人在身体上比人类更加强壮、坚韧。可是，机器人并未像许多科幻小说描述的那样起来征服人类（在本游戏中，Sam面对的所有恶意机器人都是在执行一个坏人的命令）。与此相反，这些机器人仍须遵守人类制定的法律，尽管其原因从未得到充分的解释。事实上，机器人没有任何权利，在很大程度上，人类只是将它们看做自己的财产而已，就像在美国历史的头100年里人们对待非洲奴隶一样。例如，当一台机器人出现故障时，更为划算的做法也许是彻底更换，而不是将其修好。一旦出现这种情况，机器人的主人将有权决定永久性地关闭这台机器人，几乎没有人会质疑这一决定是否正当。“Atomic Sam”确实影射过一个机器人“地下组织”，这个组织以尽可能人道的方式照顾着那些老龄机器人，这一情节出现在Harmony部分。

在这一时期，未来地球人已设法登上月球，并建立了一座月球移民城市New Boston。这座城市由许多穹顶建筑物组成，这些建筑物能产生可供呼吸的空气以及类似地球的引力。居民们可以在月面上行走，当然，他们必须穿上太空服。许多月球居民发现，这种短途旅行可使自己暂时从室内生活中解脱出来，从而体味到一种乐趣。

当人类最终登上月球时，他们意外地发现，那儿生活着一种外星人。许多世纪以来，这些不为人知的生物始终生活在月球的背面，只是在最近30年里，当人类在月球上建立殖民地时，他们才出现在人类面前。这些外形古怪的月球人分两种类型：“双头人”和“躯干人”。双头人拥有两个脑袋，长在身体顶部；躯干人则没有脑袋，只是在躯干正面长着一张嘴和两只眼睛。月球人不用呼吸，他们的身体密度远远超过人类，因此，他们既能在地球也能在月球大气中生





存。尽管月球人的技术落后于人类，但他们的智力同人类不相上下，与地球人接触后，他们很快便学会了英语。于是，在月球上，这些月球人与人类友好地生活在一起，并在各方面相互协作。

第一次见到月球人时，人类的反应是震惊和怀疑。不过，随着时间的推移，人们逐渐认识到月球人并无敌意，并开始与他们，特别是双头人和睦相处。由于双头人的外貌比躯干人稍微接近于人类，因此，人类似乎更加乐于接受他们。于是，只有双头人才能进入 New Boston，而躯干人则必须呆在月球的其他地方。人们发现，在科学项目方面，月球人是出色的合作者，他们能利用自己独特的思维方式帮助人类发展技术。不过，尽管两种月球人的智力水平不相上下，但只有双头人才能在学术领域和人类展开合作。

虽然我们会觉得上述许多技术成就要么毫无可能，要么不切实际，或者根本不受欢迎，但这是“Atomic Sam”中的世界，在这一世界里，环境的非逻辑正是它的魅力所在。另一方面，虽然上述未来世界包含着许多我们现在认为不可能取得的成就，但许多人们现在认为理所当然的成就在那儿毫无踪影。例如，在那一世界里，人们对个人计算机一无所知，因此，电脑游戏更是无从谈起。尽管早已发明了电视，但电视屏幕依然过于狭小，远远达不到电影的效果。一方面，在某些技术领域，“Atomic Sam”描绘的 21 世纪比 20 世纪的美国更为先进；另一方面，在其他一些关键性领域，它又的确落后于这一时代，呈现出一种独特的“原始将来”面貌。

6.2 序幕

玩家将操纵一位与游戏同名的角色，即 Atomic Sam。Sam 是一位身心正常但有些早熟的十岁男孩，一天，Sam 放学后回到家里，发现寓所遭人洗劫，父母神秘失踪。Sam 披上原子能火箭背囊——父母送给他的生日礼物，改名为 Atomic Sam，决定冒险穿过 Gargantuopolis，寻找自己的父母。

6.3 Gargantuopolis

简短的片头动画播放完毕后，玩家即可操纵呆在寓所中的 Sam。这时，火箭背囊附带的说明机器人 Instructobot 将给 Sam 提供说明，玩家可根据说明采取行动。通过这些说明，玩家可掌握有效控制 Sam 的方法。玩家也可跳过这一部分，离开寓所，深入城市，前往 Sam 父母的办公室。

Gargantuopolis 是未来世界中的一座超大城市，高耸入云的楼房给人一种莫名的恐怖感，就连 Sam 的火箭背囊也无法飞越这些楼房的顶部。穿过这座城市时，Sam 将遭到各式各样的机器人的袭击，它们试图阻止 Sam 找到父母的下落。在游戏的这一阶段，这些机器人的来历以及它们试图制服 Sam 的原因始终是一个谜团。

Sam 的父母是 Zeffir Zoom 公司的原子能科学家。这家公司试图利用原子能逐渐提高运输速度。进入 Zeffir Zoom 公司的综合性研究大楼并抵达 Sam 父母的办公室后，游戏转入片头动画，内容是 Sam 找到一张由父母匆忙写就的字条，在这张字条上，他们嘱咐道：“一定派人照看好 Sam！”此外，Sam 还将发现蜡筒的一个部件。蜡筒是一种录音设备，但由于这台 cylinder 残缺不全，因而 Sam 此时将无法播放录音。



6.4 Electric Priestess 的泡泡屋

由于无法找到有关父母失踪的任何线索，Sam 陷入狂乱之中。就在这时，一个友善的机器人突然出现在他的面前，并迅速将其引入附近的一座楼房。然后，Sam 乘电梯抵达顶楼，在那儿，他将遇到一位自称 Electric Priestess 的神秘女子。这个神秘莫测的 Electric Priestess 独自一人居住在一座摩天大楼顶层的球形“泡泡屋”内。泡泡屋的天花板为全玻璃结构，从那儿可以观赏到城市周围的壮丽景色。房内装有许多宽阔的钢门，这些钢门通向各种交通形式，Priestess 可任意支配它们。

Priestess 对 Sam 说，她知道他父母失踪的原因，并向他提供帮助。在故事的这一阶段，Electric Priestess 帮助 Sam 的原因仍然不得而知，不过，她似乎十分关心他的处境。了解到 Sam 对父母的担忧后，她主动相助，引导他找到那台蜡筒的其他部件。她给 Sam 提供交通工具，建议他分别前往三个地方；她相信，在这些地方，Sam 可以获取有关他父母和 cylinder 其他部件的更多信息。此外，她还送给 Sam 一台微型无线电通信机，将它套在耳上，Sam 就能和她保持联系了。

这时，玩家将恢复对 Sam 的控制，可以操纵 Sam 穿过三扇门中的任何一扇，找到交通工具，这些交通工具分别通往游戏的三个部分：Benthos、Harmony 和 New Boston。玩家可以任何顺序通过这三个区域，不过必须通过所有区域，然后才能前往最后一个区域，即 Ikairus。Sam 动身之前，Priestess 将乐于提供有关这些区域的背景信息。一旦玩家选定入口，就会出现一段简短的片头动画，表现 Sam 乘坐交通工具抵达目的地的过程。然后，在新的区域，玩家将恢复对 Sam 的控制。

6.5 Benthos

第一座城市 Benthos，位于海底。Electric Priestess 将吩咐 Sam 乘坐她的私人旋翼飞机（由机器人操纵）抵达通向 Benthos 的海底单轨铁路。Benthos 的居民主要由两类人组成：海底研究人员和视觉艺术家。后者移居 Benthos 是因为在这儿他们可以过上孤独、清静的生活。Benthos 是一座穹顶城市，它通过数英里以上、浮在海面上的管道吸取氧气。受穹顶高度的制约，Benthos 市内的建筑物均低于地面城市的超级摩天大楼。那些艺术家们创作的雕塑作品遍及整座城市；这些作品杂乱无章、抽象费解，但它们都具有流畅的线条，许多作品类似于毕加索的“太空时代”雕塑。

Priestess 告诉 Sam，他的父母在 Benthos 市拥有一所私人实验室，于是，Sam 将前往该市寻找自己的父母。就像在 Gargantuopolis 市一样，Sam 将遭到众多机器人敌手的攻击，这些机器人试图阻止 Sam 进入其父母的实验室。在 Benthos 市的战斗中，Sam 不必飞得像在 Gargantuopolis 市时那么高，因为穹顶限制了飞行的高度。不过，为了每战必胜，飞行仍然是快速行动和有效打击这些机器怪物的关键。抵达 Benthos 市不久，Sam 将遇到飞行女孩 Xeraphina，她会帮助 Sam 找到父母，并向他介绍 Benthos 市的情况。

最后，Sam 终于来到父母的实验室；这是一间窄小的办公室，里面摆满各种设备，墙上挂着 Sam 的几幅照片。一旦 Sam 抵达实验室，游戏便切换到片头动画，表现 Sam 找到 wax cylinder 另一部分的过程；这台重要设备是 Sam 的父母在失踪前制作的。拿到 wax cylinder 的另一部分





后, Sam 将再次乘坐单轨列车回到 Electric Priestess 的家里; 在那儿, 他可以进入下一区域。

6.6 Harmony

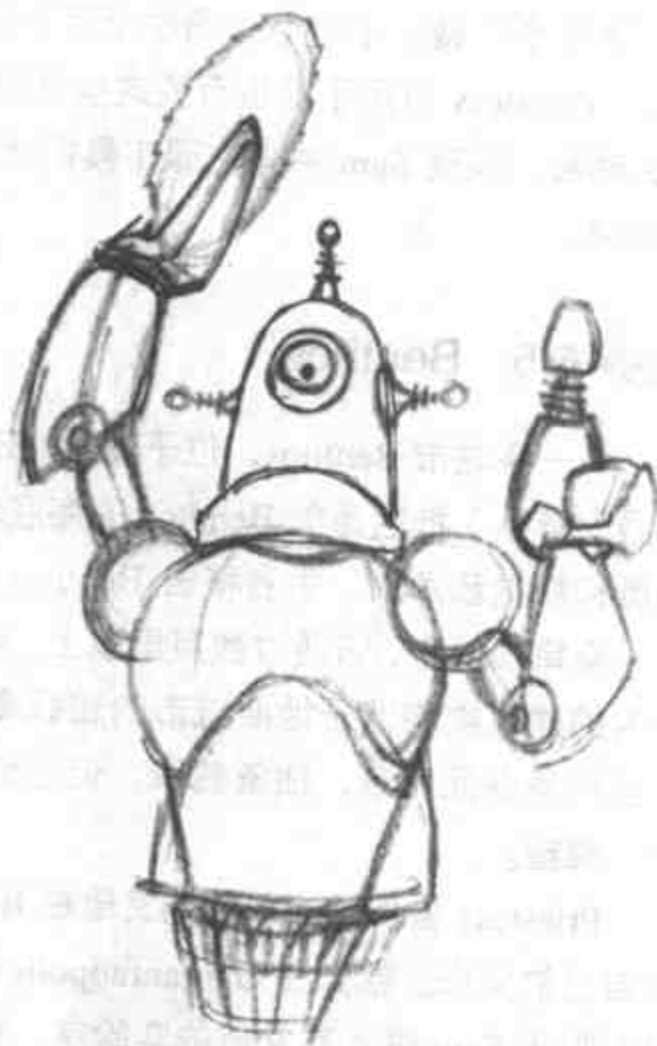
在 Priestess 的家里, 穿过一扇门, Sam 便可找到她的私人飞艇; 乘坐这艘飞艇, 玩家即可抵达 Harmony。Harmony 距 Gargantuopolis 非常遥远, 它是一个“精心设计”的特区, 由大面积的绿色公园、工业区和都市区组成。这个国家的大多数机器人均诞生于 Harmony 市, 在那儿, 机器人居民的数量远远超过人类。在 Harmony 市, Sam 必须学会区别友好的当地机器人和邪恶的机器敌人, 这些敌人将竭力阻止 Sam 寻找自己的父母。

在 Harmony, Sam 将遇到 Scrap, 一个极其友好的机器人; 他和 Sam 一见如故, 并将协助 Sam 对付那些阻拦他的机器人。此外, Sam 还希望能找到 Ike, 一台老龄机器人, 也是他父母过去的助手。Electric Priestess 告诉 Sam, Ike 住在 Harmony 市, 正与它的同类一起安度晚年。于是, Scrap 将 Sam 引到这个机器人长辈的身边。

可是, 找到 Ike 后, Sam 发现这台陈旧的机器人已失去记忆。根据国家规定, 它只有两种选择: 要么永远关闭, 要么更换一颗新的头颅。Ike 选择了后者, 因此, 很快他便会接受换头操作, 一种常见手术。不过, Scrap 担心 Ike 会失去对 Sam 父母的记忆, 因为, 在更换头颅的过程中, 记忆力往往会受到损害。Scrap 建议找一位“地下机器人医生”来实施手术, 这种机器人的职业是为老龄机器人提供秘密维修, 以保存他们的思维和记忆, 使其免遭废弃。

Sam 和 Scrap 必须穿过 Harmony 市的更多地方才能找到这种机器人医生, 然后将其带至 Ike 身边。最后, 他们终于找到一位同意为他们服务的机器人医生——一台体形巨大、名叫 Tool 的机器人。当然, 在穿过 Harmony 市时, 玩家仍将面对恶意机器人的频繁侵扰。

当 Tool 被带至 Ike 身边后, 游戏转入片头动画, 播放 Tool 为 Ike 恢复记忆的过程。Tool 出色地完成了任务, 这时, Ike 会想起 Sam 的父母曾将一台 wax cylinder 的部件交给了他, 于是, 他将部件递给 Sam。拿到 wax cylinder 的另一部分后, Sam 即可乘坐 Priestess 的飞艇回到她的泡泡屋。



附图 11

6.7 New Boston

最后, Sam 将抵达 New Boston, 一座月球移民城市。搭载 Electric Priestess 的私人火箭飞临这座城市后, Sam 将遇到一些态度友好的外星人, 他们被地球人称为“月球人”。

在一个研究项目中, Sam 的父母曾经和一位名叫 Dulo 的躯干型月球人进行过合作。Sam 必



须找到这位月球人，因为 Electric Priestess 猜想，那台 wax cylinder 的另一部分就落在他的手里。同 Benthos 一样，New Boston 也是一座穹顶城市，在那儿，居民们大致可以像在地球上一样生活。这座穹顶建筑的内部维持着和地球相同的重力，一种被称为大气产生器的装置可为居民们提供新鲜空气。为了协助研究项目，一些双头人也居住在 New Boston 市内。

当 Sam 打听 Dulo 的下落时，人们会告诉他，Dulo 是一个躯干人，无法进入月球城。于是，Sam 只得穿上太空服，离开 New Boston，在月球表面寻找他的去处。离开月球城后不久，Sam 便和 Dulo 相遇。原来，Sam 的父母注意到，在他们的研究领域，Dulo 具有一些特殊的才能，因此，他们自愿离开 New Boston，前往 Dulo 在月面上的住处。

Dulo 告诉 Sam，他确实拥有 wax cylinder 的一部分，但把它放在了家里，那儿离这座穹顶城市非常遥远。Sam 将陪伴 Dulo 取回 cylinder 的那一部件。当然，在 New Boston 以及月球表面，更多的机器人敌手将试图阻碍 Sam 达到这一目的。同 Xeraphina 和 Scrap 一样，Dulo 将与 Sam 协同作战，打败那些在月面上遇到的敌手，消除机器人的骚扰。抵达 Dulo 的住所后，Sam 将从 Dulo 手中得到那台 wax cylinder 的部件。然后，Sam 必须且战且退至 New Boston，并从那儿返回地球。



6.8 Electric Priestess 的泡泡屋

穿过这三个区域后，Sam 将收集到所有他认为必需的 cylinder 部件，并返回 Electric Priestess 的泡泡屋。在一段片头动画里，Electric Priestess 表示，她对 Sam 能找回所有 cylinder 部件感到极为钦佩。可是，当 Sam 试图组装 cylinder 时，发现还少一个部件。于是，Priestess 向他透露，最后一个部件就在她手里；拿到这一部件后，Sam 小心翼翼地将 cylinder 组装完毕。

幸运的是，Priestess 拥有一台能播放 cylinder 的机器。在 cylinder 上，Sam 的父母解释了他们的研究工作以及他们因这项工作而得罪 Zeffir Zoom 公司老板 Max Zeffir 的原因；显然，正是 Max Zeffir 绑架了 Sam 的父母。Sam 听到父母说，他们在 Zeffir Zoom 公司工作时发现 Zeffir 的一种新型单轨铁路系统存在着严重的安全隐患，如果重新设计这一系统，公司将蒙受巨大的损失。遗憾的是，他们继续说，虽然 Max Zeffir 本人已经发现这一问题，但他拒绝做出修正，而是要求他们保持沉默，以便这种单轨铁路系统能顺利竣工。

蜡筒中的内容播放完毕后，Electric Priestess 透露，她就是 Zeffir 的妹妹。她原先是 Zeffir Zeppelins 公司的一名首席科学家，多年前，在一次飞艇事故中，她失去自己的一条腿，她将那次事故归因于 Zeffir 降低成本的举措。她始终怀疑 Zeffir 与 Sam 父母的失踪不无关系，但她认为，她必须找到证据，然后才能向 Sam 表明自己的猜测。她告诉 Sam，事实上，她曾经是 Sam 父母的朋友。由于担心遭到 Max Zeffir 的抓捕，Sam 的父母破坏了证据，即那台蜡筒；然后，他们将拆散的部件分开存放，一个放在他们的家里，一个留在 Benthos 市的办公室内，剩下的则分别寄给了 Ike、Dulo 和 Electric Priestess 本人。于是，Priestess 断定，正是在 Zeffir 指使下，那些机器奴才们试图阻碍 Sam 找到有关他父母失踪的真相。



6.9 Ikairus

Electric Priestess 还告诉 Sam，他的父母很可能被关在 Zeffir 的原子能飞行堡垒 Ikairus 上。





这是一艘长期飘浮在空中的巨型飞船，在原子能的驱动下，可以无限期地飞行。这座飞行堡垒是 Zeffir 的骄傲和乐趣，同时也是他的住所。这艘飞船由大约八台螺旋桨引擎支持，在高空中，它就像一只巨大的飞鸟，大得足以让其他飞机在它上面起降。

Priestess 再次将自己的私人旋翼飞机借给 Sam，乘坐这架飞机，Sam 即可飞往 Ikairus。在这座飞行堡垒上，Sam 必须和更多的机器人以及那些极难对付的冷面雇佣兵交手。Ikairus 上的战斗发生在狭小、有限的空间即飞船的走廊内，在那儿，玩家必须相应地调整自己的战斗风格。最后，Sam 将与疯狂的 Zeffir 对峙。Zeffir 不仅关押着 Sam 的父母，而且还抓住了 Xeraphina、Scrap 以及 Dulo。当 Sam 和 Zeffir 搏斗时，Zeffir 吹嘘说，一旦打败 Sam，他将如何如何处置 Sam 的朋友。最后，Sam 设法打败了 Zeffir，并与父母团聚；Sam 的父母庆幸自己曾在 Sam 生日时送给他一台原子能火箭背囊。



7 参考书目

在构思“Atomic Sam”的背景和环境时，我的灵感主要来自以下书籍。这款游戏的制作者们将会发现，研究这些书籍对于全面了解“回返未来”环境相当有益。

“Yesterday's Tomorrows”：Corn、Joseph J.和 Brian Horrigan 合著，巴尔的摩，约翰斯·霍普金斯大学出版社，1984 年。

本书以开阔的历史视角探讨了上世纪人们对未来世界的诸多想象与憧憬，书中收集了大量珍贵的照片和资料。

“Tom Strong”：Moore、Alan 和 Chris Sprouse 合著，La Jolla, CA: America's Best Comics, 1999 年。

漫画书“Tom Strong”是 Moore 和 Sprouse 的杰作，它的故事发生在“清洁、和睦”的 21 世纪，讲述的是“科技英雄”Tom Strong 的冒险经历。

“Terminal City”：Motter、Dean 和 Michael Lark 合著，纽约，DC Comics, 1996 年。

与“Tom Strong”相比，Motter 和 Lark 在“Terminal City”中描述的未来世界略显凄凉和阴暗，不过，二者在技术发展上的迟缓却完全一致。“Tom Strong”和“Terminal City”均具备绝妙的视觉设计以及令人叹服的故事环境，完全适合于“Atomic Sam”之类的电脑游戏。



术语表

这部分内容包含了本书中术语的简要定义，尤其对于不熟悉电脑游戏方面术语的读者，可以提供一定的帮助。有些定义会转向讨论关于编程的概念，在这种情况下，我会提供足够的信息，使读者对术语的含义得到全面的了解。对于希望了解更完整定义的读者，建议大家去找一本从编程角度讲述电脑游戏开发的书籍，有许多这方面的书籍可供参考。

A*：这是电脑游戏中最流行的路径查找算法，虽然该算法并不出色，但它总可以一贯并迅速地找到最短最有效路径。**A***算法的基本原理：通过首先扩展离目标位置最近的有效节点，并不进行广度搜索的前提下，找到最短的可能路径。当然，几乎在任何介绍游戏编程的书籍中都会详细地介绍这个算法。请参见 **Pathfinding**（路径搜索）。

Agent（主体）：参见 **AI Agent**（AI 主体）。

AI：参见 **Artificial Intelligence**（人工智能）。

AI Agent（AI 主体）：AI 主体是指人工智能在游戏中控制的实体；并且是实体动作的主体。在电脑游戏中，AI 主体包括与玩家作战的怪兽，还有与玩家交谈的 **NPC**（**Non-Player Character**）。许多人将这些怪兽本身误认为是“AI”，但这正体现出他们缺乏对 AI 的正确理解。就如同不能说走在路上的人是“智能”，我们同样不能将游戏中的主体称为“AI”。参见第 9 章“人工智能”。

Algorithm（算法）：在游戏开发的领域中，这个概念通常是指设计用于解决特定问题的一小段代码，一般是数学上的问题。例如，我们可能会使用一个算法，用于判断 3D 环境中的人物是否可以互相看到。或者是用于查找从第一个人物到第二个人物可行路径的代码，也可以称其为算法。在 **SimCity** 这样的游戏中，可以根据玩家在构建城市过程中所设定的选项，使用算法在特定位置计算人口密度。

A-Life（A-生命）：参见 **Artificial Life**（人造生命）。

Alpha（alpha 测试）：通常用于描述未完全完成，但却已经可以玩的游戏。此时，设计过程和内容已经大体上完成，但仍需进行修正错误方面的改进，并对游戏的各部分进行调整。生产商经常使用它来定义开发过程中项目的状态，它的下一个状态通常就是 **Beta** 状态。其他的开发者可以对 **Alpha** 状态给出不同的定义，比如说用于表示处于可玩状态的所有游戏。

Arcade Game（街机游戏）：严格地说，电脑游戏是从街机游戏发展而来的。它也可以指相同游戏的家庭转换版本。更广泛地说，街机游戏描述的是以简短和激烈为特征的所有游戏。参见第 4 章中的内容，其中包含对街机游戏 **Centipede** 的分析，并对街机游戏作为一种派别存在的特点进行了探讨。

Art（美工）：在开发游戏的领域中，它经常用于描述游戏的图形内容。它也可以表示所有游戏开发者所热衷的东西，电脑游戏本身的创造性也可以称为一种美术。作者最青睐的对美术



的定义，来自于 Scott McCloud 所著《Understanding Comics》一书：“在我看来，艺术是没有发展为人类两个基本本能的任何活动：这两种本能就是生存与繁殖”。有些游戏开发者花费大量时间争论电脑游戏是否是一种艺术，但这些观点很缺乏建设性，或者没有什么价值。

Art Bible (美工圣经)：包含有游戏美工资源概念梗概的游戏开发过程中使用的文档，也有可能是一些描述性的文本。游戏的美工小组将美工圣经作为开发游戏图形内容的参考工具，通常是为了维护一致性。

Artificial Intelligence (人工智能)：游戏中的人工智能控制所有的实体和主体，它们对玩家做出响应，或者向玩家提供不可预知的挑战。单个玩家游戏中的人工智能，一般可以实现多玩家游戏中人的智能所提供的角色。第9章“人工智能”一章中提供了对这个概念的完整定义。

Artificial Life (人造生命)：通过指定 AI 主体的基本行为和需要（可以让它们出于“本性”地实现特定操作），具有人工智能的系统试图模仿生物生命。尽管人造生命在游戏 Creatures 中使用广泛，但它与大多数游戏中使用的 AI 类型相反。

Assets (资源)：游戏的内容，通常用于指美工、声效、音乐，并可以指游戏关卡。代码本身通常并不是资源。

Avatar (化身)：就如同游戏世界中的代替品一样，玩家的化身就是他在游戏世界中所扮演的角色。“The Avatar”还是 Ultima 系列游戏中玩家控制的角色名。

Beta (Beta 测试)：这是游戏在经过 Alpha 后达到的状态，并且也是游戏发行前或向公众发布前的最后一个步骤。在 Beta 版中，对游戏的修改严格地限制为只能纠正错误。某些开发者将第一个待发行的版本定义为 Beta 版。参见 Alpha (alpha 测试) 和 Release Candidate (待发行版本)。

Bible (圣经)：游戏产业中用这个词来指代在游戏开发过程中使用的各种参考资料。参见 Art Bible 和 Story Bible (美工圣经和故事圣经)。

Boss Monster (大怪兽)：它是指游戏中的敌人，本质上并不一定是“怪兽”，它的体积很大，或者比游戏中的其他对手更难对付。大怪兽经常在游戏的关底出现，并可以带起一阵高潮。

'Bot：“robot”的缩写，它是指游戏中人工智能的主体，设计为与人类的行为类似，一般都设计为在第一人称射击游戏的死亡竞赛中使用。Quake III Arena 和 Unreal Tournament 的特点都是在单个玩家游戏中，将 bot 作为玩家唯一的对手。

BSP: Binary Space Partition 的缩写。这是一种用于存储和渲染 3D 空间的方法，其中包括将世界划分为空间分区树的方法，这种方法在 id Software 公司的 Doom 和 Quake 游戏中得到广泛应用。

Builder Games (建造类游戏)：在游戏环境中，这个术语用于表示，在游戏中，玩家要负责建造持久稳定的建筑。从某种意义上讲，在建造类游戏中，玩家要负责关卡的设计。这类游戏的例子有 SimCity、Civilization、RollerCoaster Tycoon 和 The Sims。

Burn Rate (烧钱率)：它指一个开发人员工作一个月的时间公司所投入的资金量。其中一般包括雇员的薪水、房租、设备和其他持久不断的开销。有时，生产商会为开发者提供足够的资金，以达到烧钱率的要求，这样，开发者就不会短缺资金，并对生产商保持感激。

Candidate (候选者)：参见 Release Candidate (待发行版本)。

Capture the Flag (夺旗)：在游戏中包括两个小队，每个小队都有一面旗。这面旗放在特





定位置，并可能有人守卫，每个小队的玩家通过偷窃或暴力方式夺取另一个小队的旗。在电脑游戏中，这种游戏形式一般是在第一人称射击类多人合作游戏中出现，如 Quake 或 Unreal。

Choke-Point (阻碍点)：这是游戏中的一个点，玩家只有在通过某个特殊区域、完成特定的难题或者打败某个怪兽之后才能继续前进。通常，阻碍点前后的区域可以为玩家提供更大的自由度，而阻碍点为玩家提供了继续在游戏中前进之前必须完成的任务。

Classic Arcade Game (经典街机游戏)：经典街机游戏并不表示游戏一定要很古典，而表示在早期街机游戏流行时出现的，或者体现出那些游戏典型特点的所有游戏。经典街机游戏包括简单的单屏玩家，如 Space Invaders、Centipede、Robotron 2084 或 Pac-Man。经典街机游戏在第 4 章中有更为完整的定义。参见 Arcade Game (街机游戏)。

Code (代码)：当用于指游戏的时候，代码就是程序员输入到计算机中的文本行，计算机会将这些代码编译为游戏的功能。有才能的程序员经常被称为 code-jockey。

Color (风格)：抛开其明显的含义不谈，在游戏设计方面，它也可以指特定的内容和游戏设置。例如，在 Monopoly 游戏中加入了 Atlantic City 的街道名和萧条时期地产巨头主题，将其作为表现风格的手段。风格与游戏可玩性是独立开来的。

Concept Document (概念文档)：也就是 Pitch 文档。这种简短的文档包括了文本和概念的梗概，用于向生产商和其他提供资金的人传达项目的概念。概念文档为读者提供了游戏内容的概念，但并不包括实际开发游戏时的具体细节。如果项目受到采纳的话，概念文档通常可以演化为设计文档。

Concept Sketch (概念草图)：在实际创建图形或模型之前，游戏艺术资源的草图可以用于向其他人展示美工的外观。它也可以是游戏能够运行时来自游戏中某一场景的草图。

Creative Services (创作服务)：生产商提供的一种服务，通常包括包装盒上的美工设计，并负责游戏的其他广告和标志。

Critical Path (关键路径)：当通过游戏或其中某个关卡时，希望并鼓励用户选择的路径。它可以使人想起 The Wizard of Oz 中的黄砖路。

CRPG：角色扮演类游戏的计算机版本。参见 Role-Playing Game (角色扮演类游戏)。

CTF：虽然它可以指 Valve Software 的 Classic Team Fortress 游戏，但通常是指多玩家夺旗 (Capture the Flag) 游戏。参见 Capture the Flag (夺旗)。

Cut-Scene (片头动画)：游戏的非互动部分，一般用于向用户提供关于游戏故事线索的信息，有时会包含预渲染的或实时动作的动态画面，有时会使用游戏的实时图形引擎。片头动画经常在游戏的关卡之间出现，并且有时用来作为对玩家完成游戏某个具有挑战性部分的奖赏。

Death March (死亡竞赛)：当开发小组 (尤其是程序员) 长期从事于某个项目的时候，通常都会尝试安排在某种程度上无法完成的任务。人们总是认为死亡竞赛会很快结束，但其持续的时间通常比玩家想象的更长。

Death-Match (死亡竞赛)：在多人游戏中，玩家惟一的目标就是相互射杀。这种游戏通常是指诸如 Doom、Unreal 或者 Duke Nukem 3D 之类的第一人称射击类游戏。

Design Document (设计文档)：在开发游戏的过程中用于详细描述游戏设计每个重要方面的文本参考资料。有时称为 Functional Specification (功能说明)。在第 17 章“设计文档”中对设计文档有详细介绍。





DM: 根据上下文, 参见 Dungeon Master (地下城主) 或 Death-Match (死亡竞赛)。

Dungeon Master (地下城主): 在 Dungeons & Dragons 游戏中用于指代 Game Master (游戏管理者) 的术语。参见 Game Master (游戏管理者)。

Engine (引擎): 处理游戏中大多数基本功能的核心代码, 但并不包括控制指定游戏功能的代码。有时, 引擎分为 rendering engine (渲染引擎)、sound engine (音效引擎)、behavior engine (行为引擎) 等等。每个组成部分都可以当做游戏引擎整体中的一部分。引擎通常比特定的游戏更为常用, 可以在多个不同的项目中重用。然而, 有些开发人员使用引擎这个术语来指代游戏所有的源代码。例如, id Software 将他们的 Quake 引擎授权在许多游戏中使用, 其中包括 Half-Life、Soldier of Fortune 和 Heavy Metal:FAKK2。

Finite State Machine (有限状态机): 参见 State-Based AI (基于状态的 AI)。

First-Person Shooter (第一人称射击游戏): 这类游戏包括 Doom、Half-Life、Unreal、Marathon、Quake 和 Duke Nukem 3D。在第一人称射击游戏中, 玩家的视角是第一人称的, 并且他的目标是射击自己能够看到的所有东西, 而有些第一人称射击类游戏在这个方面做了一些改进。

Flight Simulator (飞行模拟器): 经常缩写为 flight sim, 这类游戏可以模拟真实飞机的飞行。仿真程度根据各个游戏会有所不同; 有些游戏非常逼真, 所以难度较大, 而有些游戏完全不会让玩家坠机。飞行模拟器游戏包括 Microsoft Flight Simulator、F-15 Strike Eagle、Flight Unlimited 和 Hellicats Over the Pacific。

FMV: 参见 Full-Motion Video (动态画面)。

Focus (焦点): 对于指导游戏开发最重要概念的一种简要的、三到四句话的描述。在第 5 章“焦点”中介绍了关于焦点的详细内容。

FPS: 根据上下文, 它可以指代第一人称射击类游戏, 或者指代当前正在渲染的系统中所使用游戏引擎的每秒帧数。参见 First-Person Shooter (第一人称射击游戏)。

FSM: 代表有限状态机。参见 State-Based AI (基于状态的 AI)。

Full-Motion Video (动态画面): 游戏中所有非实时图像以很快的速度连续显示, 以达到类似于电影的效果。动态画面中可以包括真人演员、电脑生成的环境, 或者是两者的结合。

Functional Specification (功能说明): 技术说明的同类文档, 因为这种说明文档从用户的角度描述了游戏运作的方式, 而不是从实现这些功能的程序员的角度来描述。在游戏开发过程中, 一般是指设计文档。参见 Design Document (设计文档)。

Fuzzy Logic (模糊逻辑): 这是一种 AI, 它在游戏进展所做出的判断中加入了某种程度的随机性。这就表示, 在输入相同的情况下, AI 主体可以根据随机性做出不同的判断。

Game (游戏): Oxford Universal Dictionary 中包含了大量“game”一词的定义。在本书中, 我们最感兴趣的定义是: “竞赛性质的转变、根据规则来玩, 并且要取决于高超的技巧、力量或者好运气。”如果换做另一种说法, 游戏所提供的是向玩家提出的娱乐挑战, 玩家能够理解这种挑战, 并且能够利用他们的智慧、灵巧的双手、运气或者这些因素的组合来赢得胜利。更进一步来说, 为了能够让这些挑战更有意义, 对于如何在游戏中取胜, 玩家必须能够有很多有趣的选择, 并且这些选择必须都是有意义的。并且, 为了能使这些挑战具有真正的意义, 游戏中必须制订取胜的标准。这样的话, 如 SimCity 之类的“软件玩具”就排除在游戏的范围之外。



当然，关于游戏的特点可以写出一整本书，但本书并不是这样的书。

Game Design (游戏设计)：游戏设计可以在游戏中确定框架和游戏设置的形式。游戏设计方案可以通过设计文档来实现，或者只存在于游戏制作人员的头脑中。参见 **Gameplay (游戏设置)**。

Game Designer (游戏设计者)：游戏设计者要在项目中负责确定游戏设计方案中的游戏设置形式。参见 **Gameplay** 和 **Game Design (游戏设置和游戏设计)**。

Game Engine (游戏引擎)：参见 **Engine (引擎)**。

Game Flow (游戏流程)：游戏流程组成游戏的一连串事件。游戏可以说成是在动作、探险、解决谜题和讲述故事这些组成部分之间的流程。在这些部分中，按照比例划分的时间量和游戏进行的速度组成了完整的流程。

Game Master (游戏管理者)：在角色扮演类游戏中，游戏管理者也是一个玩家，他负责掌管游戏世界中其他玩家的行为。游戏管理者还要构思玩家要经历的冒险活动，当玩家进行冒险时，他们要动态地编排这个故事。

Game Mechanic (游戏控制方式)：实现游戏的特定方式。例如，在 **Crash Bandicoot** 中进行跳跃——攻击的控制方式就是在跳跃起来之后按住“下”或“蹲”按钮。在 **WarCraft** 中将一个战斗单位发送到新位置的控制方式就是用鼠标左键点击该单位，并将鼠标移至地图上想到达的位置，然后使用鼠标右键点击该处。从整体上来说，游戏是由很多不同的控制方式组合在一起的。

Gameplay (游戏可玩性)：游戏可玩性是游戏区别于其他艺术媒介的重要组成部分。游戏可玩性定义了玩家与游戏世界交互的方式和游戏世界对玩家动作做出反应的方式。可以认为，游戏可玩性是游戏互动性的程度和特点。当然，不同的人对游戏可玩性有不同的定义，但就本书而言，游戏可玩性并不包括游戏的故事情节、图像、声效或者音乐。这就很容易理解为什么游戏可玩性是游戏区别于其他艺术介质的标志；这些组成部分都可以在文学作品、电影或者戏剧中找到。游戏可玩性也不包括增加游戏乐趣的代码，虽然游戏引擎也可以实现游戏可玩性，但是游戏可玩性中并不包含游戏引擎。然而，如果功能一致的话，游戏可玩性也可以使用完全不同的引擎来实现。

Game-World (游戏世界)：游戏世界就是游戏发展的空间。在诸如 **The Settlers of Catan** 这样的棋盘类游戏中，游戏世界就是进行游戏的棋盘。对于运动类游戏来说，游戏世界是真实的世界，但要限制在游戏场地的范围内。对于角色扮演类游戏来说，游戏世界控制在游戏管理者和玩家的想象之中。对于电脑游戏来说，在电脑的内存中有一个“虚拟”空间，玩家可以通过电脑屏幕看到它。玩家在游戏中的所作所为局限在游戏世界中，游戏本身或者其他玩家的反应也要限制在这个范围内。

GM：根据上下文，可以解释为 **Gold Master (母盘)** 或者 **Game Master (游戏管理者)**。

Going Gold (完工)：开发人员完成游戏的制作工作，并且可以制作母盘发送给制造商。参见 **Gold Master (母盘)**。

Gold Candidate (金质待发布版本)：参见 **Release Candidate (待发布版本)**。

Gold Master (母盘)：游戏的最终版本，通常要刻录在最优质的 CD 上（称金质 CD），制造商可以用它来制作实际发行的游戏副本。换句话说，母盘就是游戏的最终版本。





Graphical User Interface (图形用户界面): 玩家与电脑交互时使用的交流方法, 主要使用图形方式。例如, Macintosh 一直使用图形用户界面, 而没有使用 MS-DOS 或者 UNIX 中面向文本的界面。游戏使用 GUI 开始新游戏, 装载保存过的游戏, 并从主菜单中选择其他选项: 并且还可以用于向玩家传达从游戏世界的视角并不直观的信息。玩家控制人物的健康值、当前配备的武器、弹药量、生命数和分数等等。

GUI: 参见 Graphical User Interface (图形用户界面)。

Heads Up Display (仪表盘): 置于玩家游戏世界视角顶端的一种图形用户界面。其中可以包括玩家角色的健康值、区域的小地图或者某种雷达, 一般用于向玩家传达必须可以很方便获得的重要信息。仪表盘的名称来自喷气式战斗机飞行员使用的显示设备, 当他们驾驶飞机的时候, 这些仪表盘可以向飞行员提供重要的飞行信息。

High Concept (好主意): 尝试将各种类型的游戏可玩性或设置融入到一个游戏中的想法, 而不管这些不同的概念是否可以很好地相互配合。例如制作一款第一人称射击游戏, 而它同时又是一个赛车游戏, 或者在战争游戏中包括一个高尔夫球模拟器。通常也是“馊主意”的同义词。

HUD: 参见 Heads Up Display (仪表盘)。

IF: 参见 Interactive Fiction (互动小说)。

IK: 参见 Inverse Kinematics (反向动力学)。

Input/Output (输入/输出): 通常简称为 I/O, 它指的是计算机用于使玩家可以输入信息 (一般是使用键盘或鼠标) 以及将信息反馈给玩家的方式 (一般是使用监视器)。就电脑游戏来说, I/O 所指的是玩家操纵游戏控制以及游戏与玩家交流游戏世界当前状态的方式。

Interactive (交互式): 交互所指的就是两个系统 (人与人、人与计算机或者计算机与计算机) 在给定的过程中相互交流信息的过程。例如, 电视节目不是交互性的, 因为只有电视在输出数据, 并且完全忽略用户/观众的反应。而两个人之间的谈话是交互性的, 因为交流的双方都在倾听对方所说的话, 并且说出相关的内容或者做出反应。举另一个例子, 一场严格意义上的演讲并不具有交互性, 因为演讲者读的是准备好的演讲稿, 而不会接收来自听众的信息。而讨论组是具有交互性的, 因为讨论中的教授或者负责人将会回答学生的问题, 倾听他们的想法, 并且对其做出评价。游戏是具有交互性的, 因为玩家和电脑都要判断游戏中的情况。电脑游戏中长时间播放片头动画, 而玩家不进行任何控制的过程中, 游戏则不具有交互性。

Interactive Fiction (互动小说): 这个术语是由 Infocom 提出的, 它可以作为文字冒险的另一个名称。有些人使用互动小说来描述应用所有术语文字描绘场景并且包含文本解析器的游戏, 即使其中有图像也是如此。参见 Text Adventure (文字冒险)。

Interactive Movie (交互式电影): 这是那些参与游戏制作工作的人想出来的术语, 他们想让自己的工作听起来更好听, 类似于连环画行业有时将那些较长且更加复杂的作品称为“图像小说”。一般情况下, 交互式电影比普通游戏包含了更多更长的片头动画。遗憾的是, 所谓“交互式电影”的制作者们一般都会比交互式情况下加入更多的电影, 结果导致作品中几乎没有什么好电影, 并且由于缺乏互动性而无法称为好游戏。

Inverse Kinematics (反向动力学): 这是一种动画技术, 人物骨骼中的关节移动到指定位置时, 那个关节所依托的关节, 或者是受到该关节影响的关节就会自动移动到正确的位置。例



如，如果用动画显示一个人，那么这个人的手移动到门把手的位置，肘部和肩膀也会自动移动到合理的位置上。参见 **Skeletal Animation**（骨骼动画）。

I/O：参见 **Input / Output**（输入/输出）。

Isometric（等量）：等量的含义定义为“在测量上的等量”，特别是指画出的物体。如果从直接指向观察者的立方体的某个点上等量地画出一个立方体，那么立方体的边长度均相同，并且不会使用任何透视缩略效果。如 **Civilization II**、**SimCity 2000**、**SimCity 3000** 和 **StarCraft** 中都使用了等量绘画的方式。这样就使得游戏可以从 3D 高空视图来绘出，然后可以向所有的方向旋转，而不需要使用 3D 渲染引擎。游戏世界中的透视效果从技术上来说是错误的，但玩家不会注意到这一点。这种效果也可以称为游戏世界的“四分之三”视角。

LAN（局域网）：局域网的同义词。这些网络一般在指定的区域包含少量的电脑，这些电脑相互联网，但是不必连接到 **Internet** 或其他网络上。

LAN Party（网上聚会）：一些朋友聚会时，他们会把电脑带到一个集中的位置，在电脑上一起玩多人游戏。高速的网络可以使这些玩家玩更快且延迟更少的游戏，比在 **Internet** 上或者其他远程网络上玩的游戏快得多。

Linear（线性）：当通过线段只有一条路径可以从 A 点到达 B 点时，我们就说这种移动是线性的。线性表示在单维空间以外缺乏选择：前进或者后退。在游戏范畴中，线性游戏就是那些不会为玩家提供更多选择的的游戏。对于某些游戏来说，线性可能意味着完全没有选择，因为其中并没有提供后退的选项。

Lone Wolf（独行侠）：这个术语用于描述那些实际上在游戏开发的过程中自己完成所有工作的开发人员：设计、编程、美工、声效和写作。至少，这些独来独往的开发人员必须独立完成所有的设计和编程工作。这样的开发人员一般不会开发商业化发行的软件，但也有一些例外。例如，**Chris Sawyer** 独自完成了 **RollerCoaster Tycoon** 的设计和编程工作，承包人按照要求完成了美工。虽然没有自己完成美工的制作工作，但 **Sawyer** 仍可以称为独行的开发人员。

Massively Multi-Player（大容量多人游戏）：严格来说，多人游戏所指的是同时有非常多的人一起玩游戏，至少要 100 人以上。一般来讲，这样的游戏很持久，并可以在 **Internet** 上玩。**Ultima Online** 和 **Ever Quest** 就是这种大容量多人游戏的例子。参见 **Multi-Player**（多人）和 **Persistent**（持久）。

Media（媒介）：上街去买 **Marshall McLuhan** 的《**Understanding Media**》。好好理解一番，然后再回来。

Metagame（游戏理念）：根据 **Richard Garfield**（**Magic: The Gathering** 的制作人）的说法，游戏理念就是“游戏与生活的联系方式”。游戏理念表示玩家所喜欢的东西，从玩游戏的经历中得来，并且会影响到他以后玩游戏的方式。这特别适用于多人游戏。例如，将 **Quake III** 这样的游戏搬上 **Internet**。如果一个玩家在玩游戏的过程中不讲道德，并且使用一些不可取的招术，那么其他的玩家就有可能努力在以后的游戏中将他除掉。这就意味着由于他在以前游戏中的不良行为，可能会导致没有机会玩后面的游戏。这种各个游戏中玩家之间的关系并不是玩游戏本身的一部分，但是却是玩游戏过程中游戏理念的一部分。举另一个例子来说，在 **Magic: The Gathering** 中，玩家在玩游戏之前准备他的装束所花费的时间，虽然并不是游戏本身的一部分，但却是游戏理念的一部分。





Milestones (里程碑): 这是一个经常在生产商和开发人员之间使用的一个术语。里程碑是在指定时间完成项目中工作量的约定, 只有当那个里程碑到来时将完成的工作交给生产商, 生产商才能付钱给开发人员。

Mod: “改进 (modification)” 的缩写, 改进是用户制作的附加项, 或者是对现有游戏的修改。id Software 公司的开放结构策略中有很多的改进, 这样可以使玩家在 Doom 中制作自己的关卡。除了关卡以外, 改进中还包括新 AI、新武器、新美工或者这些内容的组合, 很有可能会产生一种基本上完全改变的游戏体验, 与原来的游戏完全不同。

MUD: (Multi User Dungeon 的缩写, 挖泥巴游戏)。MUD 类似于文字冒险游戏, 在核心的游戏机制中有很浓的 RPG 成分; 与其他游戏之间的重要区别是, 这种游戏要在稳定的、大容量的多人环境中才能玩。MUD 是 1980 年初由大学生们发起并组织的。当游戏的玩家达到足够高的经验级别或者头衔之后, 就会成为游戏内容的制作者, 让其他经验较少的玩家来玩。MUD 中多人游戏的主要特点是, MUD 中的玩家是社会的组成部分, 在进行 Dungeons & Dragons 风格的冒险之前, 他们倾向于与不相识的人交谈。在很多方面, Ultima Online 都表现为一个管理完善的图形化 MUD 游戏。另一个比较流行的游戏是 MOO, 它表示 MUD, Object Oriented (面向对象的挖泥巴游戏)。在游戏设计方面, MOO 和 MUD 是等同的, 只有编程方式和设置方式是不同的。

Multi-Player (多人): 玩家多于一人的游戏。目前, 这个词还可以指“联网的多人”, 每个玩家都有自己的电脑, 可以在网络上 (如 Internet) 与其他玩家争斗。

Non-Linear (非线性): 很明显, 这个词是线性的反义词。就游戏来说, 这表示玩家并没有局限在以特定的次序完成特定的目标, 或者以指定的方式完成所有的目标。相反, 玩家可以通过多条路径在游戏中走动, 并且可以使用多种方式取得成功。非线性留给玩家更多的选择, 使玩家可以使用自己的方式玩游戏。参见 Linear (线性) 和 On a Rail。

Non-Player Character (非玩家角色): 电脑游戏中所有不受玩家控制的人物。一般都是指游戏世界中并不与玩家敌对的人物, 如 RPG 中的村民。

NPC: 参见 Non-Player Character (非玩家角色)。

NURBS: 表示 non-uniform rational B-splines (非均匀有理 B 样条)。这是一种用于制作曲面的 3D 图形技术, 在 3D 图形编程的书籍中可以找到关于它的详细解释。

On a Rail or On Rails (On a Rail 或者 On Rails): 在一个游戏中, 玩家被迫要以一种非常明确、谨慎的控制方式在游戏中前进, 就如同被限制在铁轨上一样, 可以称这个游戏为 on a rail。称为“on a rail”或者“on rails”的游戏都是非常线性的游戏。有一种称为“轨道射击”的游戏, 它所达到的 on rails 的程度很高, 以至于玩家所驾驶的运输工具完全要按照事先预定好的路径飞行, 并且玩家只能向他们身边经过的目标射击。Rebel Assault 就是轨道射击游戏的一个例子。参见 Linear (线性)。

180 Degree Rule (180 度规则): 这是一种剪切剧情的电影技术, 摄影机必须一直放置在一条线段的某一边, 这条线段将镜头中两个受人注意的中心连接在一起。如果摄影机从不转到那些 180 度角以外的区域, 观众就不会对各个角色之间的镜头切换感到迷惑。

Parser (解析器): 在游戏中, 这个术语通常指文字冒险游戏中的输入方法。解析器可以接收玩家输入的自然语言词汇或句子, 并将其翻译成游戏逻辑可以理解的命令。解析器可以变



得非常完善，但仍会无法理解玩家作为命令使用的许多句子。自然语言的处理是 AI 研究的一个主要领域，远未达到完美的程度，所以解析器中无疑还存在许多问题。解析器更为先进的用法就是作为游戏脚本语言的解释程序。参见 **Text Adventure**（文字冒险）。

Pathfinding（寻路）：这是 AI 代码中的部分，这种技术可以让一个主体找出从游戏中一个位置到达另一个位置的方式。从理想的角度来说，寻路可以使 AI 主体避免受到障碍物或者其他主体的阻碍，但许多游戏中的寻路技术并不完善。在寻路中可以使用多种算法，例如 A* 算法，在生成路径的效率和质量方面可以出现不同的结果，而这个主题最好在关于编程的书籍中学习。参见 A*。

PC：可以指游戏的玩家角色；也可以指基于 Intel 的个人电脑，这种电脑是由 IBM 推广并且使用 MS-DOS 的机器。参见 **Player Character**（玩家角色）。

Persistent（持久）：持久的游戏是指持续运行并且维持游戏世界状态的游戏，而无论某个玩家是否会玩这个游戏。持久的游戏通常都是指大容量多人游戏，反之亦然。MUD 就是最早的持久游戏之一，而诸如 **Ultima Online** 和 **Ever Quest** 这类游戏使持久游戏备受主流玩家的欢迎。参见 MUD。

Pitch Document：参见 **Concept Document**（概念文档）。

Place-Holder（占位部分）：一般是指开发过程中所使用的美工和声效，但在游戏发行之前，开发人员会计划将其替换掉。

Platform（平台）：通常用于描述游戏可以运行的不同系统。过去和现在，流行的游戏平台包括 **Apple II**、**Atari 800**、**Commodore 64**、**IBM PC**、**Commodore Amiga**、**Macintosh**、**Atari 2600**、**Nintendo Entertainment System**、**Sega Genesis** 和 **Sony PlayStation**。

Player Character（玩家角色）：指玩家在游戏中控制的角色，如 **Super Mario 64** 中的 Mario，**Tomb Raider** 中的 Lara Croft，或者 **Doom** 中的太空陆战队员。这个术语源于诸如 **Dungeons & Dragons** 这样的 RPG。

Player Surrogate（玩家替身）：参见 **Surrogate**（替身）。

Playtesting（游戏测试）：这个术语用于指代测试游戏可玩性的过程，以判断游戏是否好玩。游戏测试不同于一般意义上的修正 bug 或者质量保证，因为游戏测试着重于游戏本身的表现，而不是一般的 bug 修正。参见第 23 章“游戏测试”。

Port/Porting（移植）：从一个游戏平台将游戏转移到另一个平台的过程，如从 **PC** 移植到 **Macintosh**，或者从 **Sony PlayStation** 移植到 **Nintendo 64**。一般来说，要移植的游戏都要首先在一个系统中完成，然后才能移植到其他系统中。

PR：参见 **Public Relations**（公共关系）。

Pre-Rendered（预渲染）：渲染成 2D 子图像的 3D 图像，或者是玩家玩游戏前的图像。**Myst** 中就使用了预渲染的 3D 图形，而 **Unreal** 中使用了即时 3D 图形。参见 **Real-Time 3D**（即时 3D）。

Proposal（提案）：参见 **Concept Document**（概念文档）。

PSX：Sony's PlayStation 控制台的缩写。实际上是根据系统早期使用的名称得来的——PlayStation X。尽管如此，这种缩写形式并不太常用。而 Sony 不喜欢有人称他们的新系统为 **PSX2**。





Public Relations (公共关系): 有一些市场部门的人员, 他们的主要任务是通过准备出版发行材料、游戏中的截图和其他信息在出版界中宣传公司即将完成的游戏。在授权使用本书中的截图时, 他们也提供了热情的帮助。

QA: 参见 Quality Assurance (质量保证)。

Quality Assurance (质量保证): 这是对游戏进行测试以确保游戏中没有 bug 且可以正常运行的过程。质量保证循环或周期就是当一个接近完成的项目在发行之前行进一步测试的时间。在大公司里, 质量保证部门或者成员就是那些进行测试的人员。

Rail, On a: 参见 On a Rail。

Real-Time (实时): 任何在玩家正在等待时进行的计算和处理都可以称之为实时的, 如处理图形和寻路。这与游戏情节实际发生之前进行的预先计算有所不同。实时游戏与基于回合制的游戏是有区别的。参见 Turn-Based (基于回合制)。

Real-Time Strategy (即时战略): 这是当前最流行的一种游戏, 包括 Command & Conquer、WarCraft、Total Annihilation 和 Myth:The Fallen Lords 等游戏。这个术语一般用于将实时游戏与基于回合制的战略游戏加以区分, 如 Civilization、X-Com:UFO Defense 和 Alpha Centauri。

Real-Time 3D (实时 3D): 这个术语描述的是当玩家观看时进行渲染的 3D 图像: 这样, 当玩家在游戏世界中移动时, 游戏中许多物体和配置的不同视角就可以动态生成。Unreal 使用的是实时 3D 图像, 而 Myst 使用的是预渲染的 3D 图像。参见 Pre-Rendered (预渲染)。

Release Candidate (待发布版): 这是游戏的一个生成版本, 开发人员认为这是可以发行的版本。release candidate 通常也要进行几天的测试, 最好是一至两周, 以判断游戏是否已经消除了所有的 bug, 并且可以交付给生产商。某个游戏经历 5 个以上的 release candidate 也是很正常的。

Role-Playing Game (角色扮演类游戏): 这是从诸如 Dungeons & Dragons 之类用笔和纸写成的角色扮演类游戏制作出来的。那些原来的非电脑游戏也是这种主题的游戏, 因为在那些游戏中, 玩家扮演他们自己创造的角色, 并且指引这些角色在幻想的世界中前进。RPG 中的许多乐趣就在于扮演这些角色的玩家具有不同的性格。具有讽刺性的是, 大多数电脑上的角色扮演类游戏所包含的传统 RPG 的方面都很少, 而是选择了将重点放在战斗技巧和奇怪的设置上。

RPG: 参见 Role-Playing Game (角色扮演类游戏)。

RT3D: 参见 Real-Time 3D (实时 3D)。

RTS: 参见 Real-Time Strategy (即时战略)。

Scripted (脚本化的): 在游戏中, Scripted 指预先安排好 AI 行为, 使 AI 主体在关卡中出现某种情况时看起来更聪明。每次玩家在玩某个关卡时, Scripted 事件都以相同的方式做出反应。Half-Life 使用了 Scripted 事件来生成让人印象非常深刻的效果, 让人以为它是非常聪明的 AI 系统。

Sim: simulator (模拟器) 或 simulation (模拟) 的缩写。参见 Simulation (模拟)。

Simulation (模拟): 在模拟游戏中, 游戏设计人员的主要目的就是准确实际地构造出一个现实的系统, 而并不仅仅是尽可能地使游戏好玩。这个系统可以是任何事物, 如飞机、赛车或者城市。

Simulator (模拟器): 参见 Simulation (模拟)。

Skeletal Animation (仿真骨骼运动): 用于替代 vertex deformation 的另一种 3D 动画技术。



使用仿真骨骼运动系统时，游戏可以记录动画人物骨骼的运动轨迹。动画可以控制这个骨骼，移动动画人物的骨骼，以相应匹配骨骼的运动。仿真骨骼运动系统的优点是，可以使动画部分比使用 vertex deformation 技术占用更少的空间，而且还可以获得更好的动画效果。此外，还可以按照顺序控制骨骼产生各种反向动力学效果。参见 Vertex Deformation 和 Inverse Kinematics（反向动力学）。

Skin（皮肤）：在游戏中，皮肤指的是诸如 Quake III Arena 或 Unreal Tournament 之类游戏中 3D 玩家角色使用的材质设置。玩家可以从游戏自带的材质库中选择皮肤，也可以制作自己的皮肤并加入到游戏中。在第三人称射击游戏的领域之外，皮肤在 The Sims 中得到很多应用。

SKU：stock keeping unit 或 shelf keeping unit（代表库存控制单元）。这是与每个条形码相关联的惟一号码，仓库可以使用 SKU 来记录库存清单。游戏的每个不同版本有时都要使用一个不同的 SKU 来标识。如果某个游戏要在不同的平台上发行，假设说是 Macintosh 和 PC，那么每个版本就会有一个不同的 SKU。同样，Thief 和 Thief Gold 虽然实际上是相同的游戏，但却是两个不同的 SKU。

Software Toy（软件玩具）：这是由 Maxis 公司的 Will Wright 创造出的一个术语，用于描述公司的第一个产品 SimCity。软件玩具非常类似于游戏，但它并没有定义游戏取胜的标准。玩家在玩游戏的时候不能希望自己“胜利”或者“失败”。而如果玩家可以定义自己的胜利条件，那么软件玩具就可以成为游戏。参见 Game（游戏）。

State-Based AI（基于状态的 AI）：这是一种 AI，它在每个主体中都使用了各种状态。状态包括空闲、行定、攻击等动作。然后，AI 会根据游戏环境中的情况将主体从一种状态转换为另一种状态。基于状态的 AI 也可以称为有限状态机或者 FSM。

State Machine（状态机）：参见 State-Based AI（基于状态的 AI）。

Story Bible（故事圣经）：该文档中提供了游戏世界中游戏元素的所有信息。故事圣经的内容可能会非常多，尤其是关联到确定历史的特性时，如 Star Trek 或 Ultima 世界。这些文档可以作为开发人员在制作游戏过程中的参考资料。在第 15 章“游戏开发文档”中介绍了关于故事圣经的详细内容。

Surrogate（替身）：这个术语用于描述玩家在游戏中控制的实体，也可以称为玩家角色或者玩家化身。参见 Avatar（化身）和 Player Character（玩家角色）。

TDD：参见 Technical Design Document（技术设计文档）。

Technical Design Document（技术设计文档）：该文档以设计文档中描述的方式来指定制作游戏的技术方案，并且以更多技术性、以代码为中心的术语解释了游戏可玩性的实现方式。在第 15 章“游戏开发文档”中介绍了关于技术设计文档的详细内容。

Technical Specification（技术说明）：技术设计文档的另一个名称。参见 Technical Design Document（技术设计文档）。

Text Adventure（文字冒险）：文字冒险类游戏中没有使用图形，仅使用文字向玩家描述游戏世界。玩家可以以命令的形式通过输入自然语言的句子与游戏世界进行交互，并使用这种方式告诉他们的角色下一步该做什么。Infocom 公司在 1980 年初使这种形式的游戏非常流行。参见 Interactive Fiction（互动小说）。

Three-Quarters View（四分之三视角）：一般是指具有等量视觉点的游戏。在所有使用这





种视角的渲染系统中都有一个从地面向上的仰视角，可以将摄像机放在以地面平面 45 度角的方向上。参见 Isometric（等量）。

Turn-Based（基于回合制）：在电脑进行下一步动作之前要等待玩家行动的所有游戏都称为基于回合制的游戏。例如 Civilization 就是一个基于回合制的战略游戏，而 WarCraft 是一个即时战略游戏。对于某些非电脑游戏的例子来说，象棋也是一种基于回合制的游戏，而英式足球是实时游戏。美国足球是实时游戏和基于回合制游戏的奇怪的复合体。

Turn-Based Strategy（基于回合制的战略）：参见 Turn-Based（基于回合制）。

Vertex Deformation：这是一种 3D 动画系统。在这种系统中，每个模型中的各个顶点要在动画的每一帧中一个个地移动到新位置。这是在编码方面最简单的 3D 动画方法，但是与仿真骨骼运动系统相比却有许多的缺点。有时还可以称为关键帧动画。参见 Skeletal Animation（仿真骨骼运动）。

Virtual Reality（虚拟现实）：从技术上来讲，虚拟现实或者 VR 是指高级的现实模拟系统，玩家要佩戴一副带有小型监视器或者显示设备的眼镜。这样，玩家就可以获得一种真正的 3D、立体声视觉的体验。此外，玩家在佩戴 VR 头盔时，还可以转动头部，让他们在虚拟世界中的视角相应改变，以对应于他所看到东西的新位置。VR 系统还包括带手套或者全身服装，这些装备可以检测用户的动作，并将这些动作转变为虚拟世界中的动作。虚拟现实是所有电脑游戏概念中最经常误用的术语。许多带有夸张思想的游戏开发人员认为他们所制作的 RT3D 第一人称游戏就是 VR 游戏，但因为游戏中并不需要带头盔，所以这些游戏并不是 VR 游戏。市场部门的人员最喜欢误用并且滥用这个术语。

VR：参见 Virtual Reality（虚拟现实）。

Wargame（战争游戏）：当战争游戏这个词在电脑游戏中引用时，通常是指那些面向战略的游戏，其中使用了基于笔和纸或者棋盘类战争游戏的玩法，如 Avalon Hill 公司制作的那些游戏。电脑上的战争游戏几乎都是模拟了历史上的战争，一般都使用了六边形的战场，而且使用了基于回合制的玩法。置于历史战争中但是没有战略成分的游戏不能称之为战争游戏。电脑上战争游戏的经典例子包括 Kampfgruppe 和 Eastern Front（1941），另外还有 Panzer General 和 Close Combat。



精选参考书目

在加强电脑游戏设计思想方面，以下参考书籍对我提供了很大帮助。我在这里列出这些书目，作为“建议阅读书籍”推荐给那些想继续学习游戏设计的人们，使他们的视野不至于受到本书的限制。

推荐书籍

Bogdanovich, Peter. "Who The Devil Made It", New York: Knopf 出版, 1997 年

这是与著名电影导演访谈录的精选集。本书中所使用的谈话方式就是 Bogdanovich 的谈话风格。

Campbell, Joseph. "The Hero with a Thousand Faces", New York: Bollingen Foundation Inc. 出版, 1949 年。重新印刷, Princeton: Princeton University Press, 1972 年

Campbell 编著的书是了解历史中神话特点、英雄故事的权威性书籍。

Crawford, Chris. "The Art of Computer Game Design", Berkeley, CA: Osborne/McGraw-Hill 出版, 1984 年

Crawford 的著作是第一本关于电脑游戏的著作，本书也受到了该书的启发。尽管书中所提到的电脑游戏行业所处的年代距今已经很久远，但目前仍具有很广泛的影响。虽然此书已经绝版，但是在 Internet 上的很多地方都可以找到它，其中包括 www.erasmatazz.com。

Hague, James. "Halcyon Days: Interviews with Classic Computer and Video Game Programmers", Issaquah, WA: Dadgum Games 出版, 1997 年

Hague 编著的书中提供了很多非常有价值的信息，其中提出，在游戏行业工作就如同是开始创业一样。所有的信息都是以一系列关于广泛话题的交谈形式提供的，本书中也讨论了其中很多人的作品：Eugene Jarvis、Dani Bunten Berry、Dan Gorlin、Brian Moriarty、Ed Rotberg 和 Chris Crawford 等等。Hague 的公司 Dadgum Game 在 www.dadgum.com 还提供了该书的 HTML 格式。

McCloud, Scott. "Reinventing Comics", New York: Paradox Press 出版, 2000 年

"Understanding Comics" Northampton, MA: Kitchen Sink Press Inc. 出版, 1993 年

虽然这些书从技术上来讲是关于喜剧的，其中也提供了关于媒体和艺术的大量见解。可以说 "Understanding Comics" 一书改变了我思考艺术的方式。

McLuhan, Marshall. "Understanding Media", New York: McGraw-Hill Book Co. 出版, 1964 年。重新印刷, Cambridge, MA: MIT Press, 1994 年

这是一本介绍所有类型媒体的权威书籍，是一部在 Internet 时代具有全新意义的著作。McLuhan 在写作风格上有些愚钝，但是他的见解是独一无二的。

Strunk, William and E.B. White. "The Elements of Style", New York: Macmillan Publishing 出版, 1959 年。第四次重新印刷, Ed., Boston: Allyn and Bacon, 2000 年



该书保留了权威性的观点和简洁的写作风格，编写设计文档、剧本或者编著关于游戏设计书籍的人最好阅读本书。

杂志

Computer Gaming World (Ziff Davis Media)

该杂志几乎与电脑游戏同期出现，**Computer Gaming World** 信息丰富，见解独到。

Game Developer (CMP Media, Inc.)

这是电脑游戏/台式机游戏混合内容的杂志，其中着重介绍了先进的游戏技术，有时还会提到设计游戏的理论。

Next Generation (Imagine Media, Inc.)

计算机和游戏机混合类杂志，重点关注前沿信息，包括技术、原理和人群等。

网站

www.mpath.com/dani

关于 **Dani Bunten Berry** 内容丰富的页面，他是著名的 **M.U.L.E.** 的极有天赋的设计师。其中包括 **Berry** 关于游戏设计的著作以及她对职业生涯的反思。

www.costik.com

Greg Costikyan 以他的铅笔和纸的游戏设计而闻名，他制作过许多电脑游戏，其中包括经典游戏 **Toon** 和 **Paranoia**。他的网站上有一系列他所撰写的文章，其中包括非常有趣的“**I Have No Words & I Must Design**”一文。

www.erasmatazz.com

这是 **Chris Crawford** 在网上的主页，内容主要是关于互动式故事讲述引擎 **Erasmatron**。其中有 **Crawford** 撰写的关于游戏设计的大量文章，还有他为 **Journal of Computer Game Design** 撰写的所有文章，并加入了“**The Art of Computer Game Design**”全文的链接。必读。

www.gamasutra.com

Gamasutra 是 **Game Developer** 杂志的姊妹网站。这个网站提供了一些最初的内容和杂志的某些重印内容。在它的各个页面中，可以找到大量的信息。

www.theinspiracy.com

这是 **Noah Falstein** 的游戏顾问公司 **The Inspiracy** 的主页。其中包括大量 **Falstein** 撰写的文章，还有他在游戏开发者论坛谈话的手稿。



[G e n e r a l I n f o r m a t i o n]

书名 = 游戏设计：原理与实践

作者 =

页数 = 438

SS号 = 11201072

出版日期 =