

Broadview
www.broadview.com.cn

来自**Sony、Blizzard、Disney、LucasArts、Activision**
3DO等公司的主要游戏开发人员的最抢手的权威技巧和技术。

顶级 游戏设计

→ 构造游戏世界

Ultimate Game Design → Building Game Worlds



[美] Tom Meigs 著
陈贵敏 杜敬利 韩琪 译

Mc
Graw
Hill



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

http://www.phei.com.cn

来自Sony、Blizzard、Disney、LucasArts、Activision、3DO等公司的主要游戏开发人员的最抢手的权威技巧和技术。

顶级游戏设计

→ 构造游戏世界

Ultimate Game Design → Building Game Worlds

[美] Tom Meigs 著
陈贵敏 杜敬利 韩琪 译

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书不同于一般游戏设计编程类的“技术型”书籍，而是一本涉及游戏设计技术、开发和产业方方面面问题的著作。

本书共分 11 章，前 7 章详细讲解游戏设计的基本知识，如游戏开发中的预视化技术、关卡设计、游戏分类、各种视觉效果、音效、脚本事件以及游戏测试。第 8、第 9 两章阐述如多人在线游戏和移动电话游戏 / 无线游戏这些特殊类型游戏设计需要考虑的问题。第 10、第 11 两章讲述游戏开发团队的分工，以及如何在游戏产业中找到合适工作的问题。

另外，本书每一章的后面还都附有一个作者对人物的访谈录，这些被访对象都是具有广泛游戏设计经验的专家和游戏产业中有影响力的人士，所以他们能给读者提供有关游戏设计和产业的独到和有价值的见解。

通过本书学习，读者不仅能独立完成游戏设计工作，而且还会对游戏设计开发和游戏产业有系统和深入的理解。

本书可供游戏设计工程师、开发商及游戏爱好者学习与使用。

Tom Meigs

Ultimate Game Design: Building Game Worlds

ISBN: 0-07-222899-7

Copyright © 2003 by the McGraw-Hill Companies, Inc.

Original Language published by The McGraw-Hill Companies, Inc. All Rights reserved. No Part of this publication may be reproduced or distributed in any means, or stored in a database or retrieval system, without the prior written permission of the publisher.

Simplified Chinese translation edition jointly by McGraw-Hill Education (Asia) Co. and Publishing House of Electronics Industry.

本书中文简体字翻译版由电子工业出版社和美国麦格劳-希尔（亚洲）出版公司合作出版。未经出版者预先书面许可，不得以任何方式复制或抄袭本书的任何部分。

本书封面贴有 McGraw-Hill 公司防伪标签，无标签者不得销售。

版权贸易合同登记号 图字：01-2003-8814

图书在版编目（CIP）数据

顶级游戏设计：构造游戏世界 / （美）梅格斯（Meigs, T.）著；陈贵敏，杜敬利，韩琪译。—北京：电子工业出版社，2004.5

书名原文：Ultimate Game Design: Building Game Worlds

ISBN 7-5053-9801-6

I. 顶… II. ①梅… ②陈… ③杜… ④韩… III. 游戏—软件设计 IV. TP311.5

中国版本图书馆 CIP 数据核字（2004）第 026543 号

责任编辑：顾慧芳

印 刷：北京市增富印刷有限公司

出版发行：电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

经 销：各地新华书店

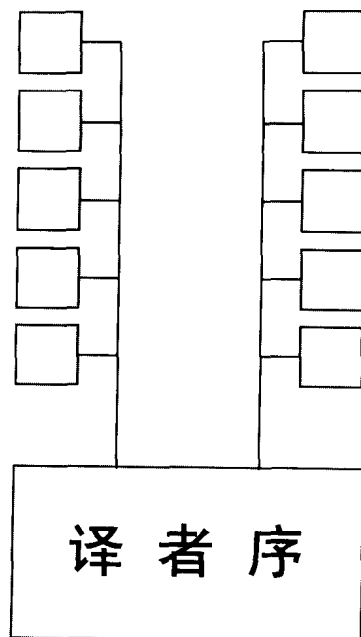
开 本：787×980 1/16 印张：23.5 字数：422 千字

印 次：2004 年 5 月第 1 次印刷

印 数：8000 册 定价：35.00 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 zlt@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。





可以说这是一本有关游戏设计“软件工程”的著作。

从内容上来看，本书大致可分为三个部分：

1. 游戏开发中的预视化技术、关卡设计、游戏分类、各种视觉效果、音效、脚本事件以及游戏测试；
2. 大型多人在线游戏和移动电话游戏 / 无线游戏的设计；
3. 游戏开发团队的分工，以及如何在游戏中找到合适的工作。

作者 Tom Meigs 从 1993 年一直到现在，先后在 THQ、Disney、Sony、TBD 等多家公司工作，担任过的职务包括开发团队成员、设计师、制作人、质量保证员、音效师等，参与开发的游戏更是数不胜数，可以说，作者对游戏产业的方方面面都非常熟悉，而本书正是作者对自己多年的工作经验的总结。读者朋友可以通过访问作者的网站 www.tommeigs.com 获取更多信息。

作为译者，我想我的任何评价都会带上比其他人更多的个人色彩，还是看看亚马逊 (www.amazon.com) 上的读者评论吧：

“这本书非常有用，它提供了许多在其他地方找不到的信息。我强烈推荐这本书。作为一名游戏设计者，它在很多方面帮助了我。”

“我向我见过的每一个对怎样制作游戏感兴趣或者想要亲自创作游戏的人推荐这本书。至于我，我想成为一名游戏美工。书中每一部分都有我在其他地方找不到的有用内容。如果你想要踏踏实实制作游戏的话，这本书可以帮助你渡过难关。单单‘参考信息’这一部分（注：指附录 A）就是一座金山。”

“我想进入游戏产业，我买了很多参考书，但是这一本是最喜欢的一本。我一拿到它，就在书中找到了许多一直困惑我的问题的答案。我喜欢体育竞技游戏，所以所有有关构造快速游戏的材料对我都很有用。我想任何想要进入游戏产业的人都应该好好读一下这本书。”

……

对于这样一本“经验型”的书籍（相对于编程类的“技术型”书籍而言），翻译过程非常艰苦和枯燥，大部分译文都是在反复斟酌之后才最终确定下来的。非常感谢电子工业出版社的朱沐红女士，是她促成了本次翻译合作，并给予我充足的翻译时间。

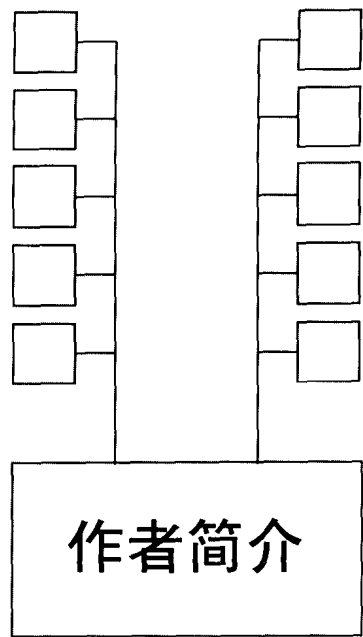
在翻译过程中，刘小院、郭永献和梁丽萍等阅读了部分译稿，并提出了很多宝贵意见，在此表示感谢。

译文中一定还有很多不妥之处，敬请读者朋友批评指正。我的电子邮箱是：
efoxxx@eyou.com。

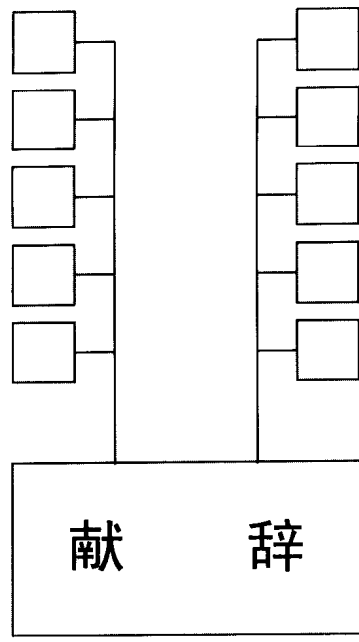
希望本书能帮助那些想要进入游戏产业的朋友找到合适的位置，帮助那些已经在游戏产业的朋友做得更好！同样，也希望我国的游戏产业更上一层楼。

译者于西安电子科技大学

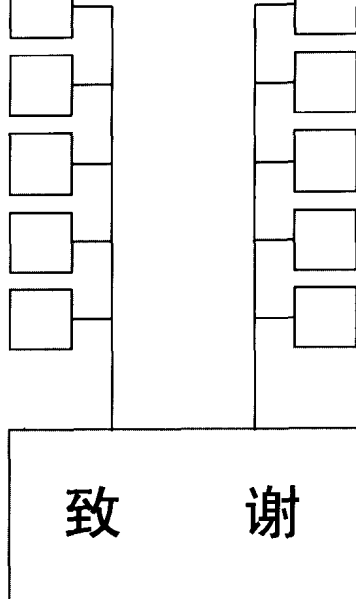
2004.1



Tom Meigs 是一位有着十多年经验的电子游戏制作人和设计师。他曾经在多种获奖游戏作品的开发平台上工作过，这些平台包括：Nintendo Game Boy, Sega Game Gear, Sega Genesis, Super Nintendo, Sega Saturn, Sony PlayStation/PlayStation2, PC, Macintosh, 移动电话, theme park kiosks, location-based entertainment, 网络, 以及 short-lived Bandai Pippin。他参与过的游戏作品有: *Madden Football'95*, *Sports Illustrated Golf*, *Sea Quest DSV*, *Akira*, *Jungle Strike*, *Bassmaster's Classic*, *The Mask*, *Angel Devoid 2*, *Youngblood: Search and Destroy*, 以及迪斯尼的多部作品。Tom Meigs 在加利福尼亚州立大学长滩 (Long Beach) 分校获得理学硕士学位。



谨以此书献给 Vivian E, Meigs, MCW 和 Larry Siegel, 是他们每个人的鼎力支持, 才使得这本书最终得以出版。

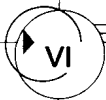


许多人以不同的方式帮助我完成了本书。首先，我要感谢 David Fugate, Wendy Rinaldi, Dr. Jon Orwant, Athena Honore, Monika Faltiss, Bill Mcmanus, 以及工作在 McGraw-Hill/Osborne 的每一位工作人员，是他们支持了我走过艰辛的每一步。

其次，我要特别感谢每一章后边的访谈录的参与者们。他们每一个人均具有广泛的游戏设计的专家经验和影响力，他们在百忙之中参与到本书的访谈中来，是值得我们称赞的。感谢 Andrew Holdun, John Kreng, Rick Sanchez, Nathan Hunt, Aaron Odland, Andrew Forslund, Melinda White, Mike Weiner, Dave Warhol, 以及 Bill Roper, 他们凭借自己丰富的经验为本书提供了很多独特的、有价值的见解。这是我见过的最慷慨的帮助，也是我见过的最有见识的访谈。

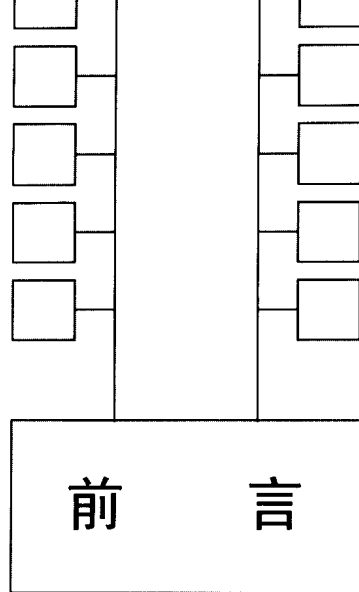
感谢 Andrew Forslund 在脚本部分所做的大量、及时的工作，感谢 Andy Wang/Netamin 允许我使用从《顶级棒球在线》—— Ultimate Baseball Online 中截取的很多图片。

我个人还要感谢 Art, June 和 Jeanette Meigs; Larry, Sandy 和 Scott Kessenick; Kevin Wright, Gene Hoglan/SYL, Tommy Lasorda Baseball 工作团队, Metroid 工作团队, Richard Holmes 教授和 Simon Schama 教授, Gordon Sumner, 迪斯尼, Eitetsu Hayashi, MLB great Darrell Evans, Tomahawk, Melvins, Michael 和 Julie Allen, Dave Moses, Adrian Belew, Roscoe's, Mykonos, Harbour House, 在 Waterside Productions 工作的每一个人, Gnomon 视觉特效学校, Alex Alvarez, Tao Tong, Long Beach Ice Dogs 机构, Patrik Augusta, Master Kreng, 几位博士: Paul Tang, Charles Hughes, George Spangler, Shane, Andre, Doug Deaver, 和 Bill Hyde; Harold Budd, Trey Gunn, Marjorie Stettbacher, Brendan Perry/Quiwy, Jim Wright, Andy Summers, Elvis 和 Baloo。



顶级游戏设计——构造游戏世界

还有整个 Black Pearl Software/THQ 开发团队, Electric Dreams 开发团队, Realtime Associates 团队, 迪斯尼游戏团队, UBO/Netamin 开发团队, Blizzard Entertainment 的每一个人, Orange County, CA 国际游戏开发协会 (CA International Game Developers Association) UC Irvine 分部, 和 Art Institute-LA, 感谢他们给予我的支持、灵感以及对各种各样知识的共享。



这本书是专门为每一位热衷于学习实用游戏设计 (applied game design) 的读者写的。在某种程度上讲,这本书更适合于游戏开发新手,但是,书中也花了大量的笔墨讲述有关游戏设计过程本身的内容,对于那些对游戏制作过程有不同熟悉程度的开发人员来说,这些内容都是很有用的。

整本书是按照游戏开发过程的顺序组织的,从设计过程开始,直到游戏开发的最后阶段或者叫做“黄金”开发期——整个游戏制作完成并交付到游戏玩家手中。

按顺序阅读各章内容,你将从头至尾经历一次基本的游戏开发流程,并了解许多基本的游戏设计领域以及所面临挑战。当然,你可以直接跳到你特别感兴趣的章节阅读,或者从相关附录中的帮助信息入手——这也许是进一步研究书中主要内容的有效切入点。

我写这本书的目的,是试图就游戏设计细节的解决方法给开发新手和经验丰富的行家们提供一些相似的共同点。书中有关游戏设计理论方面的内容也写了很多,但相比之下,还远远少于所谓的“实用游戏设计”方面的内容。千万别误会:我还是相信游戏设计理论更加重要的。我想,我们可以很有把握地认为:理论有助于指导并启发实践。但我只是想尝试着为所有对游戏设计感兴趣的朋友写一本更像是讨论实用游戏设计的书,而且我有一些很实际的理由这样做。

当你开始逐渐了解当前制作的大多数游戏的开发环境时,你就会清楚地认识到考虑实用游戏设计是非常必要的。归根结底,我们还是要按照游戏的内容强制性地将其分为很多类,不过,即使是经验丰富的开发人员,要做到这一点有时候也都非常困难。这与

商业的生存观恰恰相反——游戏制作商在游戏设计质量方面通常依赖于可复现的结果。

开发一个具有绝妙可玩性的游戏常常会遇到很多障碍，其中的原因有很多，我们接下来就会讨论这个问题。为了开发成功，游戏开发者们需要在比以前更短的时间内更加稳妥地增强游戏的趣味性并添加令人着迷的情节。然而，游戏原型开发技术以及围绕着游戏可玩性的多方位体验，都给游戏开发者设置了重重挑战。在不增加太多开发成本的前提下，游戏原型开发和游戏动态体验仍然是一件棘手的事情。

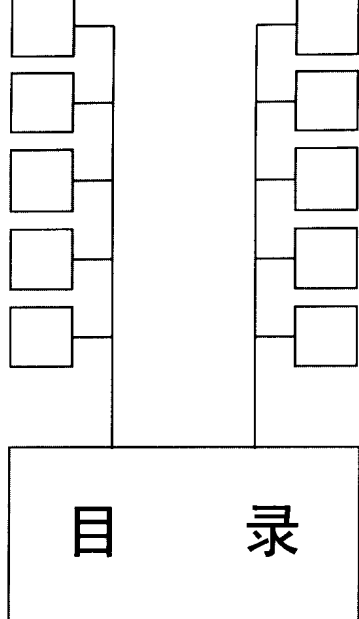
请牢记，本书试图解决的一个最重要的问题是：大体上需要些什么才能让实用游戏设计更适合于游戏开发人员？我试着给出了几种答案。

我认为按照本书所尝试的方法深入研究实用游戏设计，会给所有游戏开发新手一个学习关于设计挑战的第一手资料的机会，同时，会引导有经验的开发人员去思考有助于减少其中一些挑战的方法。我把这种研究看成是有助于使你能制作出更加有趣的游戏的一次对话。

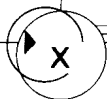
当然，紧接着下来我们就会详细介绍这些总是阻碍我们的可恶的开发细节：开发工具，开发工具与游戏引擎之间稳定的接口，能够胜任原型开发和试验的能力，带着完好的概念开始你的开发周期（为了改善和增强游戏的趣味性，可以对这些概念随意地进行改变和调整），等等。

那些游戏开发人员或者是中间件（middleware）供应商，成功地为游戏内容的构造提供了最强有力且灵活多变的开发方法支持，并使得开发人员有可能搭建出最出色的游戏，很有可能会发现自己在游戏销售排行榜中竟然名列前茅。这已经不再是一个不为人知的秘密了，现如今排在游戏销售排行榜中前几名的游戏都是以像 RenderWare（Criterion Software 公司开发的 3D 套装工具软件）这样的技术为基础来开发的，这些技术使得游戏制作商可以留出更多的时间充实那些令人兴奋的细节内容，而不必担心要去跨越巨大的技术障碍。我的观点可以总结如下：如果说内容是国王，那么时间就是宝座。

这就是我写此书的初衷。到了由你提出一直困扰你的问题，并在应用游戏设计领域内找到答案的时候了。此刻，你也该甩掉那每隔二十分钟就要学习全新的游戏设计工具的包袱了。我知道，如果你把这里所收集的资料作为设计的出发点，你很快就会发现很多的方法，可以快速增进或者加强你对很多有助于塑造游戏设计的影响因素的理解。



第 1 章 预视化技术	1
1.1 预视化过程的介绍	3
1.2 预视化的逐步实现	3
1.3 功能	7
1.4 关卡场面纸上设计	13
1.5 预视化案例研究评论	18
MEGA 提示	24
第 2 章 关卡的规划及制作	25
2.1 规划你的关卡工作	28
2.2 游戏原型设计	39
2.3 关卡占位设计的后处理考虑	41
2.4 关卡实施的案例研究评论	49
MEGA 提示	57
第 3 章 灯光、材质、粒子、特效以及音效	59
3.1 灯光	61
3.2 动态灯光	70
3.3 材质	71
3.4 粒子	74



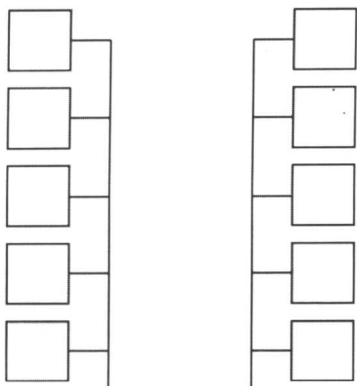
顶级游戏设计——构造游戏世界

3.5	效果	79
3.6	音效处理	81
3.7	设计工具集的变化	84
3.8	基础元素的案例研究	85
	MEGA 提示	92
第 4 章	角色、道具、物品以及镜头的细节	95
4.1	布置角色	97
4.2	道具的类型以及用法	102
4.3	物品和威力升级的类型及其布置	104
4.4	镜头的考虑	105
4.5	有关演员加载和镜头调整的评论之案例研究	109
	MEGA 提示	117
第 5 章	分类游戏设计	119
5.1	运动类游戏	121
5.2	格斗类游戏	124
5.3	解谜类游戏	127
5.4	即时战略游戏	129
5.5	角色扮演类游戏	133
5.6	第一人称和第三人称动作游戏	135
5.7	模拟类游戏	136
5.8	构造电影动画	138
5.9	开发背景故事	139
5.10	设计游戏对话	140
5.11	设计人员的工作工具总结	142
5.12	设计转型的案例研究评论	142
	MEGA 提示	148
第 6 章	编写动作事件脚本	151
6.1	脚本技术的选择	153
6.2	脚本应用示例	156

6.3 应用触发器	160
6.4 构造行为	166
6.5 引擎实现与 Unreal 引擎	167
6.6 有关脚本 / 编辑系统的因素	169
6.7 棒球游戏脚本评论的案例研究	170
MEGA 提示	180
第 7 章 质量保证以及实战测试反馈	183
7.1 质量保证	184
7.2 实战测试反馈	191
7.3 质量保证影响的案例研究	197
MEGA 提示	204
第 8 章 大型多人在线游戏的设计考虑	207
8.1 MMOG 的制作挑战	208
8.2 MMOG 游戏的构造要素与解决方案	214
8.3 MMOG 游戏设计因素	217
8.4 MMOG 游戏与设计取向	220
8.5 体育爱好者或者体育团体与 MMOG 游戏	224
8.6 挑战《顶级棒球在线》的案例研究	227
MEGA 提示	234
第 9 章 移动电话和无线游戏	235
9.1 日益迫近的繁荣	237
9.2 移动电话上的设计问题	242
9.3 无线游戏初期开发因素的案例研究	250
MEGA 提示	255
第 10 章 开始游戏开发	257
10.1 为什么要掌握多种技能	259
10.2 增长的区域与新的机遇	264
10.3 广告游戏	267

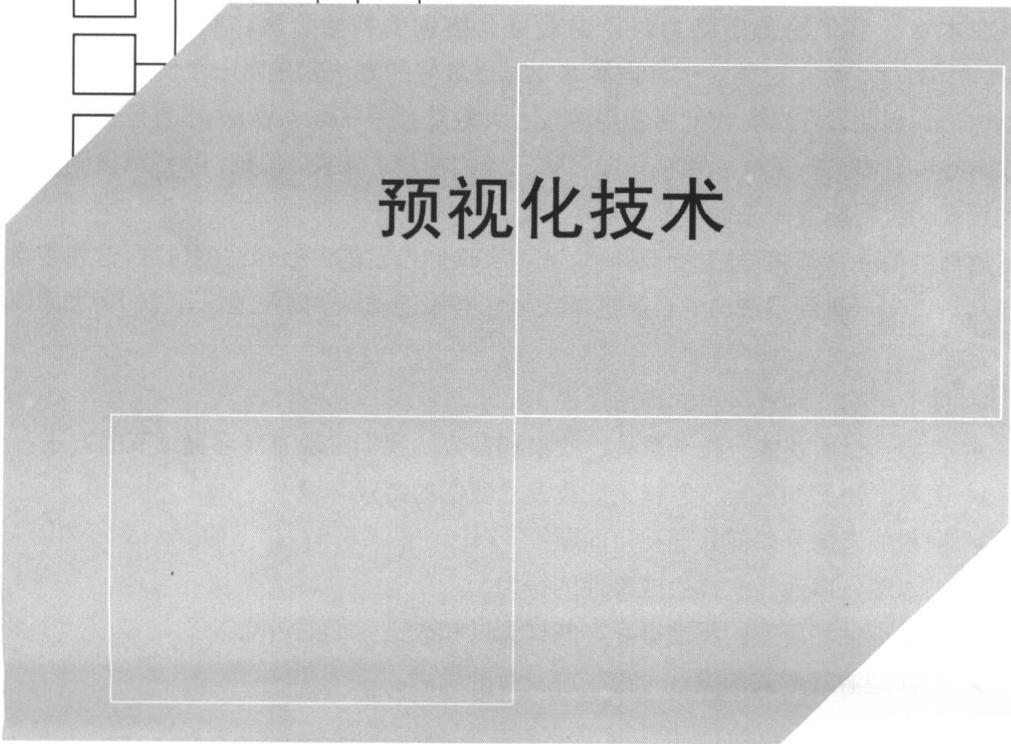
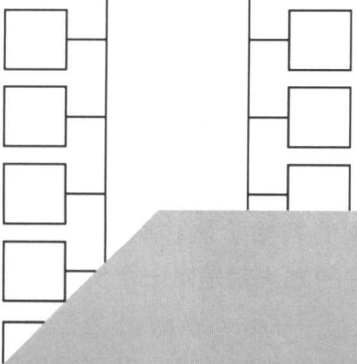
顶级游戏设计——构造游戏世界

10.4	具有现实对照物的网上游戏娱乐	269
10.5	游戏开发前线的趣闻轶事	271
10.6	对设计人员起决定作用的想法的案例研究	275
	MEGA 提示	288
第 11 章	在游戏开发产业中择业	289
11.1	程序设计	290
11.2	美工	293
11.3	游戏设计	294
11.4	游戏制作	296
11.5	音效	297
11.6	质量保证	298
11.7	创办独立的游戏工作室	299
11.8	游戏工作室的分工	301
11.9	关于测试途径案例研究评论	307
	MEGA 提示	309
附录 A	参考信息	311
附录 B	讨论过的工具	321
附录 C	职业指导工作单	331
附录 D	设计人员的快速主题总结	337
附录 E	顶级棒球在线图库	349

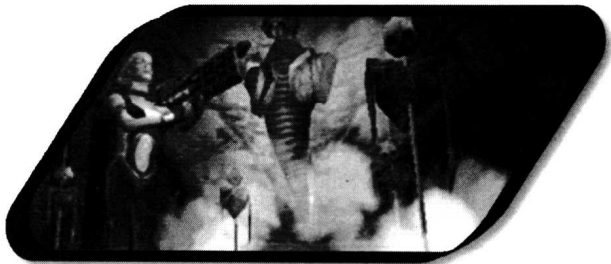


CHAPTER

1



预视化技术



游戏设计过程是从各种游戏性的想法和概念的相互综合和谐调开始的，毋庸置疑，游戏设计的前期工作主要是概念上的设计。游戏设计过程中，游戏概念是必须实实在在地规划出来，并加以具体化和定义的。当把这些概念放在一起去经历一次游戏开发过程时，它们就像是在大海上共乘一叶扁舟似的，要求它们必须紧密合作，同舟共济。只有我们规划好的游戏概念之间能够通力合作，才能取得任何振奋人心的进展。那些看上去“只是在随波逐流”的游戏是因为它们没有达成这种协调一致性。有的时候，会出现我们在前期制定好的概念而没有在实际游戏设计过程中使用的情况。请一定记住，专业的游戏设计过程通常是和大量的规则约束、开发目标以及各种深思熟虑同时进行的。

在本章，我们将集中精力讨论游戏概念协调的初始阶段，有时称其为预可视化（*previsualizat ion*）过程。也许理解这个过程最好的方法就是以例子为背景对它加以讨论，因此，在这里我们将使用游戏《大教堂》——*cathedral* 作为范例。虽然该动作游戏中可以进行第一人称视角或者第三人称视角的切换，但是，我们对该例子所讨论的预可视化过程能够用于很多的游戏类型。

后面的每一章节将详细地介绍构造或者实现你自己的游戏想法的过程，这将包含很多与从具体化一直到塑造出你自己的游戏概念有关的话题。简要地说，我们将考虑以下几个方面：

- 关卡结构 游戏环境的构造过程；
- 光照、材质映像、粒子系统、效果和音效 我们实现游戏环境细节的方法；
- 道具、物品和行为 怎样策划或设立我们的游戏环境；
- 视角的考虑 如何处理视角问题；
- 脚本动作事件 怎样生成事件的行为；
- 质量评估和玩家的反馈循环 怎样测试和改进我们的游戏；
- 对新型游戏形式的设计考虑 游戏新的玩点。



1.1 预可视化过程的介绍

预可视化过程能使你在到达目的地之前就知道你要去哪里,这样你就可以提前做好准备。是去热带丛林吗?那就带上喷雾式的杀虫剂。是到一个异国他乡的岛屿上吗?是的话就带上防晒油和捕鱼鱼叉等工具。与之相似,如果你在进行一个游戏开发,那么游戏的规划和对可能出现问题的地方的尽量预测对你的游戏成功开发来说是至关重要的。因此,在大多数游戏的开发过程的前几天,都要进行一系列快速的预可视化审查,其目的是为游戏确定一种可视化风格——即使是你正在设计另一个非常棒的关于外形奇异的像章鱼幼仔的弹珠猜谜游戏,也需要进行预可视化过程。一个外形奇异的像章鱼幼仔的弹珠是什么样的?有人能拿给我看一下吗?有没有参观它的博物馆?博物馆开放吗?有些开发小组不太重视预可视化过程,而有些却非常严肃认真地对待这个问题。无论你是否开始开发下一个狂战士、射击手,还是开发一个环境动作类游戏,你都需要一个直观的“路标”。

预可视化的真正目的,是为了帮助你,使你在大脑中形成的游戏景象与约定的范围一致,并构造一个视觉上或风格上的参照点;也可以说它是视觉定位点。当然,也可以偏离该参照或定位点。在游戏的整个开发周期中,视觉风格都会不断地得到发展,它可能会缓慢发展,也可能会剧烈变化。预可视化处理为项目中每一位开发者简单地构造了一个十分有用的出发点。

以后,当我们考察预可视化过程本身的时候,我们需要检查以下内容:

- 利用概念和参考图样。
- 实现基本的关卡构架和环境设计(关卡是游戏经历中一个独立的部分,它有自己的开端和结局。大多数游戏都分成很多关卡,只有通过了所有关卡才算完成了整个游戏)。
- 在纸上进行概念方面的工作并绘制地形参考图。
- 对设计中有用的资料进行简单实用的分类。



1.2 预可视化的逐步实现

由于大部分的游戏开发周期很少允许开发者进行长时间的准备工作,所以经常要求

开发者快速地提交具有创新性和技术性的设计文档来，这是很富有挑战性的事情。任何能够在严格的时间限制内完成的预视化工作，都能够在整个项目细节理清后正常地交付完成。在整个游戏的坎坷的开发过程中，预视化只占用很短的一段时间。

对于很多游戏开发者来说，专门拿出时间积极进行预视化处理可有些奢侈，但是，那些能挤出一点儿时间进行一些预视化处理的开发者通常在整个开发过程中会节省大量的时间。如果一个开发小组不得已在制作中期来确定大量的可视化风格方面的细节问题，而这些问题在预视化阶段就应该已经解决好了，那么开发进度就会被拖延，而且会浪费很多有价值的资源。

接下来，我们将浏览一个简单的预视化全过程，最后以《邪恶的教堂》——*cursed cathedral*的构造作为我们的范例。

■ 利用环境参考和草图

每一个游戏都有环境设置选项——也就是构造一个游戏进行的物理场所。我们不仅是一个没落的村庄构造了一个地形平面图（那只是规划的一个小分支），我们是在“构造”游戏的比赛场地！不管你是否正在试图模拟西方娱乐场的氛围，还是一个绚丽多彩甜美的卡通世界，或者是位于遥远星体上的一个烧毁了的矿藏隧道系统遗迹（哦！——或许三个都有），环境都会在很大程度上帮助你限定并支配游戏的气氛，这些气氛构成了联系玩家和游戏的情感纽带的一部分。游戏技巧和游戏策略使得玩家们沉浸其中、保持主动和兴奋，这就是情绪驱使的或情感上的经历进行下去的原因。如果你不能够抓住玩家的情绪，玩家的游戏体验将平淡而没有生气。我们都喜欢玩的大部分游戏看上去都能够将玩家的情绪和游戏技巧天衣无缝地结合起来，你无法说出它们中的一个在哪里停止而另一个又从哪里开始。如果你没有在监视器或电视机前高声尖叫的话，那么任何一个编制这个游戏的人都可能再也不会去开发另一个游戏了。

简单地说，环境应该支持和弥补游戏的可玩性——而不应该减弱它。环境本身，利用其特有的视觉风格，能够实实在在地转移或改变人的情绪。温暖、幸福也许能够描述一个精心制作的甜蜜世界所产生的氛围；黑暗、焦虑和沉闷也许能够描述你自己专有的天启（Apocalypse，宗教用词）世界里所产生的心情。因此，当我们试着确定环境设置视觉风格时，往往要借鉴充足的参考资料，比如说有助于改变视觉和风格导向的照片、图画、插图、图表和图像等等。这一点看上去似乎是显而易见的，然而通常在有关开发的争执过程中被开发者遗忘了。参考资料可以使小组有具体的东西进行讨论。在概念阶

段,建议将要开发的游戏看起来应该像《银翼杀手》——*Blade Runner*,这是很有用的,而且这仅仅是建议了一种游戏风格,它并没有真正帮助我们确定那些零乱的细节问题。难道还要把投币式的公用电话这样的游戏细节也要和“银翼杀手”中的设计成一模一样吗?想一想你到底想要知道哪些东西,尤其是那些和下一步方案有关的东西。

最基本的一点是,提供充足的视觉参考对画家、设计者和那些正在确定参考点的开发队伍通常都是很有用的。游戏开发的很多方面都是从那些大量实用的参考点开始。这些参考点并不全是可视化的:一些只是概念上的;另外一些则是很具体的游戏开发手段,它们在其他的游戏开发中曾经发挥过巨大的作用。

作为一个能够用做参考材料的图画示例,如图 1-1 所示,这是 Giovanni Piranesi (1720-1778) 的作品 *The Mole of Hadrian* (摘自“*Antichita Romane*”)。但愿你能立即看出像这样的一幅图画会怎样影响周围的建筑物、氛围、材质以及光照的。这是一幅不错的图画,但是你并没有必要去试着仿造它的外观和感觉,尽管你可能有试一下的欲望。更好的方法是,你可以把这一幅图作为部分或者整个游戏可视化风格开发中的一个重要的参考点和出发点——一个通向具有通用出发点的可视化思想的跳板。这是利用环境参考和草图来支持你那通常所用的“匆忙型”预可视化阶段的至关重要的环节。

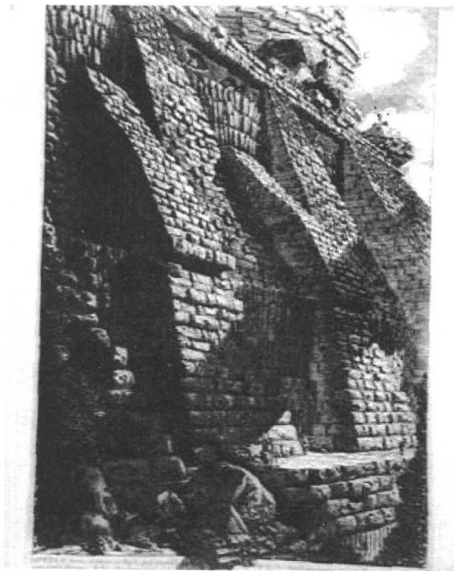


图 1-1 Piranesi 的 *The Mole of Hadrian*

■ 游戏关卡的结构

在游戏世界中，游戏的整体结构上只能提供有限的自由度，这是一种“受控制”的自由度，因为当你在构造游戏的环境时，你必须经常要权衡各种选择，并作出一个折中。比如，为了使游戏看起来更精彩，你可能需要添加某些物理特性和复杂的東西，但是这些特性会降低游戏的性能，或者减慢游戏的运行速度，甚至于慢得玩起来没有任何乐趣，所以只能放弃。这一点根本就不需要做，因为当夜幕降临，所有的灯都关了，Junior把自己藏进跑车的睡椅上，这就是所有的乐趣。时断时续的帧速率没有任何意思。请记住，游戏关卡是指整个游戏过程中较为独立的一部分，并且每一个特定的关卡都有自己独特的过关考验。

帧速率是指屏幕上一帧接着一帧地重画的速度，有点儿像翻动书页的动画。你打算以多快的速度翻动页面？那就是你的帧速率。“时断时续”的帧速率就是指帧速率太低了，以至于在玩游戏的过程中不能提供足够的运行性能，无法令玩家满意。你希望提供尽可能多的视觉冲击，同时又要保持基本的性能，这就是问题的症结和关键点。因此我们需要权衡利弊。对于现代的三维游戏，最简单的情况就是，在场景中通过画多边形来表示物品，当你往场景上放的物品越多，你就会损失越多的执行效率。

而且，那些在传统的系统结构上很管用的东西不一定非要强加于游戏环境的开发中。一般来说，内部的关卡——比如一座城堡内的结构——可以根据实际建筑物加以改造“变形”成游戏中的形状。你不但想让系统的构架适合于游戏中关卡的布局，而且希望在其中畅游或随处走动的时候充满乐趣（并且速度很快）。因此，从一个现有的参考点开始对你来说通常是一个不错的选择，比如从城堡的平面图开始，接着对它进行扭曲、修改，使之适合游戏中所需要的形状。或者，如果愿意的话，你可以从一些全新的东西出发。不管哪一种方法，你都需要实实在在的参考材料。

在数字世界（计算机）里工作的其中一个好处就是你可以把许多个不同城堡中零零碎碎的部分结构拼装在一起构成最终的城堡内部结构。如果你想在游戏中构造一个和真实事物一模一样的城堡，虽然会很中看，但是，它将会像用沙子做成的城堡一样，海浪就会使它土崩瓦解。不管你正在构造什么类型的游戏，从一个参考点出发，然后对它的游戏可玩性进行修改、细化，这样做总是很有用的。实际上，游戏的关卡布局应该由游戏可玩性本身决定。请记住，环境能够增加可玩性，但是可玩性才是游戏设计的决定因素。

■ 基本环境设计

近几年来制作的多数游戏都是以基于网络的多人游戏为特点的。在这种基于网络的多人游戏环境中,考虑几个布局上的简单想法是很有意义的。毕竟,如果你打算让几个玩家同时在同一关卡或者比赛场上漫游,你就需要仔细地考虑他们在你的地图上怎样通行。

入口和出口

通常来说,狭窄的路段和死胡同在多人游戏环境中效果并不好。然而这可以认为是一条“软”规则,因为经常有一些打破这条“软”规则的例外情况。比如说,有的人就指出一种情况:狭窄的路段可以很好地击退敌人的进攻。(“硬”规则是指不存在例外情况的规则,就像“要避免使你的玩家失去信心”这样的规则)对于环境设计来说,“软”一些的规则通常为给实际的测试反馈(*testing feedback*)留下修改的空间,测试反馈指的是玩家对游戏中哪些设计不错而哪些设计不行的评论。我们将在第7章中学习关于这个过程的所有内容。现在需要你和你的小组来指出为什么有些设计不错而有些却设计不行。测试反馈是衡量你是否成功地游戏构造了一个很好的比赛场所惟一的、也是最重要的标准。

如果你在死胡同处构造了一个储备站或能量增加站(能增加各种能量的地方,比如生命值或弹药),玩家就不得不冒很大的危险去找到它,因为通向死胡同的走道可能交通十分拥挤——很多敌人会一起围攻玩家。这就表明设计多个入口和出口会更好一些。如果在某一地方只有一个进口和一个出口(见图1-2),那么玩家首先会采取什么样的逃跑措施是显而易见的。通常情况下设计多个入口和多个出口比较好,这样的话玩家就很容易地从某种遭遇中逃脱,而且他们的出现地点或者逃离地点也不那么容易被预测到(见图1-3)。



1.3 功能

在预可视化过程中就考虑功能总是很有帮助的,在这里所说的功能是某些与“游戏目的”类似的东西。一个精心制作的游戏环境同时具有好几个目的(例如,以一种令人兴奋的方式支持某些动作和角色技能,能够在给定的硬件上以最优速度运行,环境布局的显示上具有逻辑性,并且便于操纵)。最低限度,你应该希望你的游戏环境能够以正确

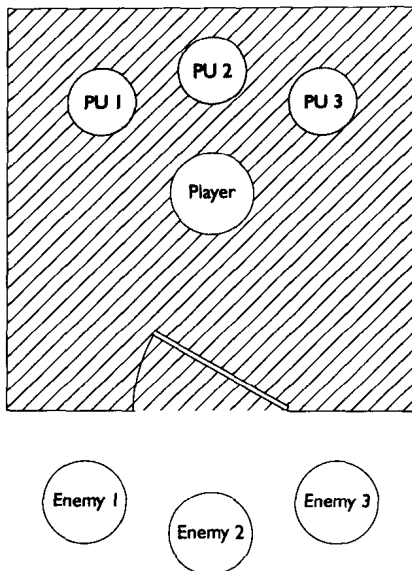


图 1-2 玩家在狭窄的路段处被擒

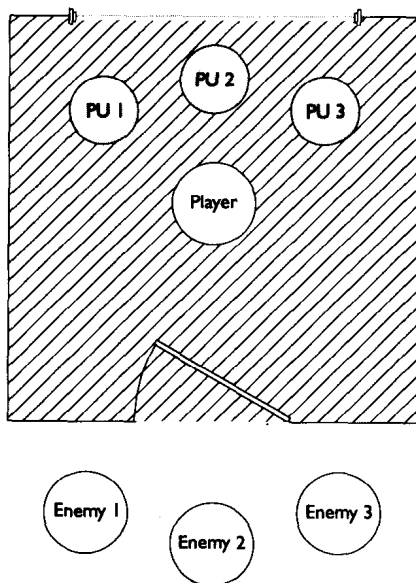


图 1-3 玩家可以向北逃跑

的物理方式支持某类动态的玩法，这些玩法正是你尽力构造或者是移植到你的游戏环境中去的。这是一个跨越游戏类型的规则。它和我们正在讨论的是哪类游戏毫无关系——从有很强视觉效果的第一人称视角的射击游戏（FPS）到多玩家的竞技类游戏——在所有的情况下你都需要预先构思好游戏的环境以及它的功能（游戏目的），并指导着你的思维向提高游戏可玩性的角度去思考。最终，你正在构造的环境是要用做你的游戏比赛场景。

构造用来“主办”游戏比赛的环境和构造只具有视觉冲击效果的环境是截然不同的两件事情。你需要两者兼而有之——一个既能很好地进行游戏比赛又具有视觉冲击效果的环境。这里一个重要的概念是：对于你的关卡（第一人称视角游戏或者第三人称视角游戏）、竞技场（生死比赛或者玩家对玩家单挑比赛）、地图（实时战略游戏或者角色扮演类游戏），以及室外运动场（具有动作游戏、争夺游戏或者射击风格的游戏），你都需要一个正确设置的环境来实现你的主要目标：精彩的游戏。如果你很少甚至根本没有考虑怎样最好地构造一个环境来增加游戏的可玩性（不管什么原因造成的，是开发的时间计划太紧张也好，是资源短缺也好），其结果通常都会让人备受挫折，而且无法实现精彩的游戏。

因此，当你在考虑功能和布局时，请考虑一下你正在努力构造的游戏机制和游戏目标。你如何开始设置游戏进行的位置和布局将取决于你试图要构造的游戏机制的类型。当然了，这也会随着你所开发的游戏类型的不同而有所变化。

首先提问自己一个问题：我们游戏中令人心跳的东西是什么？“令人心跳的东西”是指存在于你的游戏底层的最主要或最基本的游戏机制。它是你的游戏的驱动力，它也是玩家们迷恋你所设计的游戏的原因。应该时刻牢记你的游戏中那些令人心跳的东西。毫无疑问，在它一路坎坷的游戏开发过程中，这个心跳也会经历很多次的困难和挫折，但应该牢记你的游戏中这些令人心跳的东西，它可以帮助你开发过程中挡在开发小组面前的很多问题作出决定。

如果你忘掉了游戏中那令人心跳的东西，那么你开发的游戏几乎会在一夜之间变成一只粗暴的五头怪兽，到时候你将不得不对你游戏中的那些充满祸害的脑袋进行刀砍斧劈，整治一番。这可是一种非常难受的境地。只要清楚地记住一些简单的原则，在开发的曲折道路上就可以对很多问题作出决定。通常，在现实世界的资源约束条件下学习怎样才能做到这一点具有一定的挑战性。作为一个开发小组，你需要从4002个信息源中找到你的游戏开发的指导思想，这些信息源包括那些为你的游戏开发出资的人和那些正

在与你交涉的、拥有自己的性格和爱好的人。

游戏中的令人心跳的东西到底是什么呢？把这个问题的答案从你头脑中抹掉，也把和你在同一小组的成员的答案抹掉。拿出切实的行动来。

尽管很多游戏开发者在底层的细节问题上意见不一，但是很多游戏中令人心跳的东西表达出来竟然出奇得简单，而且这一点对各种类型的游戏都适用：

- ▶ 消灭其他玩家或事物，或者被对方消灭（例子包括《雷神 III 竞技场》——*Quake III Arena*，《小行星》——*Asteroids*和 *Twisted Metal Black*）；
- ▶ 通过某种方式增加自身技能、能力、能量、影响力或者公认度等等（比如 *Everquest*和 *Diablo II*）；
- ▶ 由玩家控制某一过程的模拟（《过山车大亨》——*Rollercoaster Tycoon*和《模拟市民》——*The Sims*）；
- ▶ 三维冒险类 X、Y、Z 游戏（《冥界狂想曲》——*Grim Fandango*和《迷雾之岛》——*Myst*）。

缩小、简化到这种程度还会有用吗？答案是肯定的。你的心跳可能会与这些陈述中的几条有关联，但是要注意了，大多数成功的游戏都没有涉及太广泛的“心跳陈述”，你应该把你的游戏中心跳的陈述减少到一条或者最多留两条。不要想把什么都包含进去。像这样简单的一些陈述，在一个技术进步永无止境的时代里还会有一线生机，并有胜算的可能吗？我能说的就是，有时有些事物在归纳简化之后会有更加巨大的生命力。

你应该怎样利用心跳陈述的呢？当定义好一条心跳陈述后，你和你的开发小组应该对游戏中潜在的每一个特性、功能、模式、所有物或有利于还是不利于你对这条心跳陈述的实现能力进行评估。开发游戏是一种探险活动，只有你尝试之后才会知道结果，但是通过检查一个特性或功能的思想是否背离你的游戏“心跳”就可以限制你向一些“次要的思想”偏离。如果它不能够把你的心跳陈述更进一步地呈现给玩家，那你就别再为它费心了，如果它能够做到这一点，那么就进一步权衡它的真实效果并考虑其可行性。

要知道，构造下一代游戏需要团队合作。你将作为团队中的一员为构造你们的游戏的“心跳”而奋斗。你不能够在仆人们把下午的茶准备好之后，在高高宝座上孤立地向下发布你的“心跳”声明。换句话说，游戏设计过程的一部分本身就是学习如何作为一个团队一起工作，以便实现一次“相互认可”或在“游戏心跳”上达成一致。每一位团队成员本身特有的在游戏方面的经验、倾向和热情，都会支持或排斥游戏的指导思想。在通常情况下，游戏制作者和设计人会经过很多次的反复和修改，引导着这个过程向构

造一个清晰的“游戏心跳”的方向进展——整个团队共同分享和共同奋斗的“游戏心跳”的目标。

至于功能，首先提问你自己如下的一些问题：

- 你是在开发这样的游戏吗——游戏中竞赛选手能够增强和定制他们的赛车连续比赛或者在竞赛跑道上的性能？
- 你是在开发集体攻防的战斗游戏吗？
- 你是在开发一个模拟成长过程的游戏吗？

了解游戏的“心跳”是最基本的要求，而且它会帮助你回答这些问题。首先构造游戏的“心跳”，并从那里开始继续往下做。游戏的“心跳”将会帮助你获悉那些细节，并帮助你确定游戏的功能，它还会告诉你怎样以正确的方式为你的游戏构造并使之发展成正确类型的环境。

■ 空间流程

空间流程很重要，因为它使你的游戏功能设想具体化。现在的很多游戏就是依靠从一个空间到另一个空间的内部作为游戏进行的环境。比如第一人称视角和第三人称视角射击游戏按惯例都设置在大建筑物内部的小空间里。对于高级的概念关卡，空间之间必须以某种式样相互联系在一起，比如通过过道直接地或暗含地连接在一起。一条“暗示的”路径（比如一条十分陈旧布满灰尘的信道就是一个明显的例子，关卡区域内从前到后或者是从上到下的逻辑连接也是另一个不那么显而易见的例子）对于玩家在你设置的关卡内游历时是很有用的。不同的空间类型之间和不同空间规模（比如空间的大小和你所扮的角色或车辆的大小相比较）之间通过暗示的通道相互连接。举例来说，某一关卡中你也许在“监狱房间”区域中进行探险，而和它相连的是一个地下的指挥中心，而该指挥中心又和一个通向机场的隧道相连。

空间内部随着游戏的功能、气氛、规模、目的以及许多其他特定条件的不同而不同。那么，是如何把各种空间有机地拼接在一起构成游戏中的关卡的呢？空间流程预可视化通常需要考察两个关键的地方：空间的逻辑性和对称性。

在预可视化阶段，当你考虑空间布局的逻辑性时，你需要提出一些基本的问题：各个空间之间有逻辑上的联系吗？当游戏从一个地方过渡到另一个地方，基本的空间流程是怎样工作的？你打算在这些游戏空间中包含哪些玩法和哪些主要的动作要素？游戏的可玩性原则应该经常影响并左右着环境设计的选择。如果你打算使用远程的发射武器，

你可能不希望空间距离既短又狭窄。空间的布置甚至可以提示玩家作出某些反应，比如一个房间的布局逻辑会暗示玩家这是一个进行搜查的好地方，它可能是一个潜在的高冲突地区，或者是一个隐藏得很好的再补给的站点。

物理对称性对于像夺旗(CTF)和其他的组队进行比赛的游戏来说可能是最重要的。当你正在入侵另一支队伍的地盘、夺取他们的一些东西并带回来时，你的队伍所走的路程必须和另一支队伍所走的路程一样远，不然的话就会存在一个明显的不平衡。然而，即使是双人合作或者是单人游戏的地图，那些具有某些物理对称性的地图通常都要比那些根本就没有对称特征的地图使玩家更少会感到迷惑，而且更易于游历、更使人着迷。那些拥有没完没了的、相互连接在一起的一序列空间的地图或关卡(地图中一连串的空间连在一起，一个挨着一个没有尽头)，会让玩家很快就失去了“位置”的感觉。这就使得玩家很难说出自己在关卡内相对于其他任何东西的位置。另一方面，那些在布局上具有单方向或者隐含有对称性的关卡会给玩家提供很有价值的视觉参考点(比如，在中心拳击场处的高塔)。对称性的关卡要比那些缺少或根本没有对称性的关卡“感觉”起来更好，并且更容易在其中走动(甚至到处乱跑)。

实际上，当你使用俯视图或拓扑地图作为你的关卡布局的基础开始时，你就能理解这种对称性。这就是需要你在纸上设计出来的内容的一部分。当你在最喜欢的关卡中跑来跑去的时候，停下来花点时间考虑一下对称性。不知道你是否注意到这一点没有？如果你是玩家，你没有意识到这一点那是再好不过了，在这种情况下对称性就会发挥它的魔力。然而，作为一个设计者，你需要寻找像对称性这样的模式。你需要寻找那些驱动游戏进行下去的模式。

■ 从内部到外部

从内部到外部的转移是一种很重要的功能表达。很多流行的游戏都使用那些能够在内部环境和外部环境之间平滑过渡的环境特征。内部到外部过渡的地点应该事先规划好，而且有必要进行逻辑性检查，以确保整个关卡布局上从更大范围的图来看它们是有意义的。在空间到空间之间迅速过渡到毫不相关的外部场所的内部场所，通常会使玩家相信他们在某类环境条件下待的时间不是很长。如果玩家从内部相邻的两个出口退出时，分别到达两个毫无关联的外部场景(例如，在牢狱的出口处，从其中一个门出来一下子就到了一片丛林外边，而通过另一个门却立刻到达一片荒漠)，这看起来似乎很酷，但却并不能给玩家制造出任何悬念。我们一直在努力地给玩家提供一种“连续”的感觉

或者信心,让他们体会到我们把他们放在了什么地方。荒漠中的狱营就是一个极好的例子。如果能让丛林和荒漠同时出现,你就需要为玩家考虑好他们之间如何平滑过渡过去。

■ 增加氛围

构造氛围有助于完成并具体实现你的游戏功能概念。在预可视化处理过程中你应该尝试着自问自答一些有关“游戏是怎样把某种氛围渲染给玩家”的问题。早期的一些概念草图就应该包含有那些有助于定义氛围的注释和框架。音效和视觉是怎样融合在一起来传递某种气氛的呢?如果你正在构造一个卡通世界中的虚幻战士,你怎样传递那种气氛?特定的场所决定了特定的氛围。

你希望气氛能真正反映出你要传达的那种体验。体育类游戏或第一人称视角射击游戏都提供有快速而疯狂的动作场面,但气氛是通过使玩家沉浸在视觉和听觉的暗示中而进一步得到加强的。相对轻微的感觉触动常常能够传递大量的气氛,比如大型运动场中播放的像真的在运动场中播音员的声音,就好像正在踢球的时候你听到的一样,这确实会在很大程度上调动玩家的情绪。音效在游戏中绝不可忽略,尽管在以前曾经被忽略了。通过提供或不提供音效上的暗示,可以在很大程度上增强或减弱我们对游戏气氛的理解。音效是构造和支持气氛的基础,是游戏世界设计中的一个基本因素。



1.4 关卡场面纸上设计

每一幅地图、每一个关卡和每一片竞技场——简单地说,就是每一个游戏环境——都需要从纸上开始进行设计。这样做的原因有很多。首先,它可以更加方便地编辑、更新和验证自己的想法。其次,它能够节省大量花费在无用资源上的时间和资金。你可以在关卡内某些位置、空间流程和逻辑性上十分随意地发挥。我发现那些大块儿的“干擦除”(dry-erase)白色写字板很好用。尤其是在小组自由讨论的时候,它们简直太棒了,你可以迅速地汇集大量的有价值的见解和想法。纸上的关卡场面设计在讨论时也十分有用,它能够让你快速地对关卡的流程思想、边界线、道具的放置意见和关卡的连续场景作出改变。

■ 快速绘制地形图

诚然,我们生活的世界上存在着许多令人称奇的编辑器和3-D软件包——而每一

个都有一些不同于其他的和令人难以接受的内容细节——但是很多概念的测试，在纸上比在其他任何媒介上都要更快、更容易。你可以更容易地识别出逻辑上的缺陷，设计出转移地点，检验你的想法，并且能够更加切实地把握住关卡的总体视图。而且，这对于开始运用组件化思想也是很重要的。组件化思想就是要规划出怎样从一些简单的、连接在一起的、并且可重用的组件构造自己的关卡。这样构造关卡就好像从建筑模型玩具包中取出几块并按照某种有意义的方式把它们拼接在一起一样。

好了，你已经准备好了图纸，接下来干什么呢？通常来说，你需要一幅不错的地形图或者是总体风格的地图。图 1-4 是地形图样本。我喜欢把它画在白色写字板或纸上，整理好注释，然后用微软的 Visio 应用软件制作一个简易而整洁的版本。

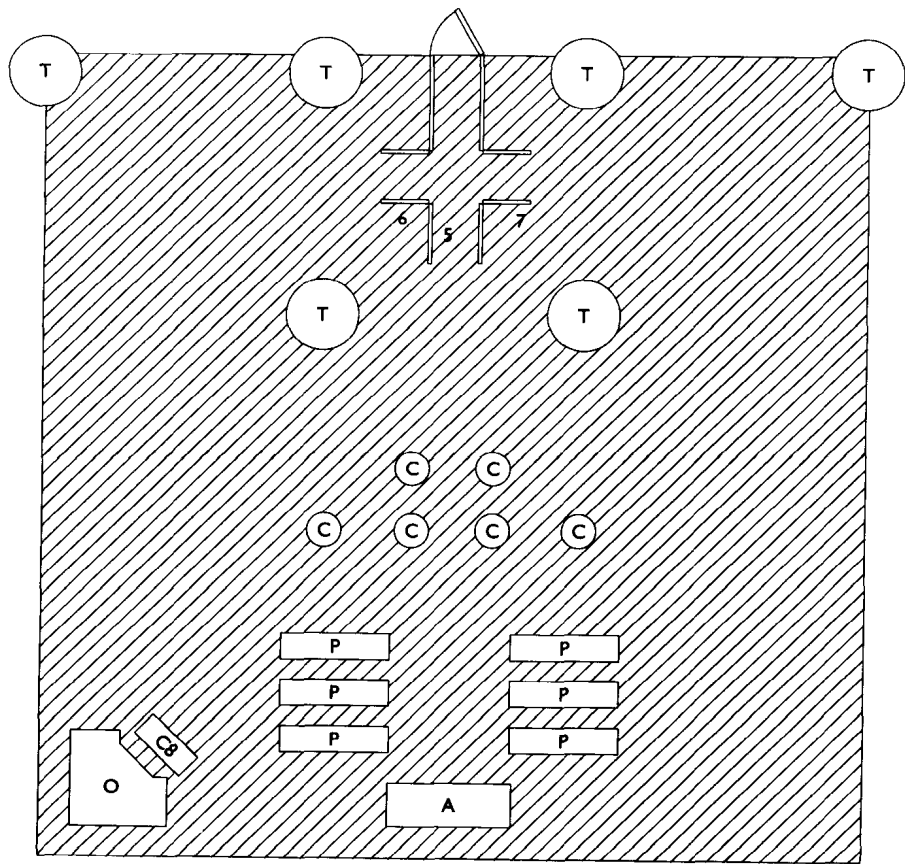


图 1-4 地形图样本

图 1-4 示意了你所设计的关卡中的教堂区域部分可能的样子。图中的圆圈代表着教堂的角楼或是塔楼，较小的圆圈只是简单的几根圆柱子。你可以用一个小的图例来说明地形图中的每一个元素分别对应什么，下面就是图 1-4 中的一个示例：

- T 角楼或塔楼
- C 圆柱子
- P 教堂内的长椅
- O 管风琴
- OB 管风琴椅
- A 圣坛

现在我们有了一个基本的位置布局（当然了，并不是按一定的比例画的）和图例体系。请记住像这样简单的地形图仅仅是一个参考点，是开发关卡的一个开端，后边还有漫长的道路等着你呢。现在让我们一起介绍性地看一下如何利用材质、道具、特效和脚本事件开始细化我们的关卡。

材质

材质将映射或附着在你的几何建筑物之上。材质是表层材料的信息——颜色、样式、对比度和色度，而且它还能表征物体的物理特性，比如崎岖不平、生锈或像石头式的等等。我们可以把材质想像成油漆桶的包装纸，它覆盖在圆柱形漆桶的外观，并给我们提供可见的材料信息。

我们需要对示例中的教堂列出一个初步的材质清单。大致上看一下我们可能会在这个地方用到的材质：

- 彩色的玻璃窗户
- 石质地面 1 和石质地面 2
- 铺地毯的地板
- 角楼楼梯
- 墙 1 和墙 2
- 角楼墙 1 和角楼墙 2
- 塔楼墙齐整的边缘
- 门 1 和门 2
- 圣坛覆盖物

- 风琴覆盖物
- 风琴椅
- 木质长椅
- 装饰用的圆柱

这些作为第一步的清单已经足够了。按照这些材质是怎样支持和定义你正在努力构造和复制的视觉外观和感观来考虑它们，这一点是很重要的。要确保收集到了尽可能多的相关参考材料。像“它崎岖不平”或者“它生了锈”这样的描述根本没有多大的用处。是什么样子的崎岖不平？生的是哪一种锈？一条视觉参考胜于一千句话。时刻准备好向你们的材质美术家展示你那些逼真的或示例性的样本。这就是为什么预视化阶段和这种视觉数据的组织能够帮助开发小组节省时间的原因。

道具

道具是随便堆放或者安置在三维环境中的物体的三维模型。在一个环境中你在周围所能找到的任何一个真实物体都是我们所构造的环境中潜在的道具。桌子、椅子、电话亭、垃圾桶和街灯都是道具的例子。我们需要对这些道具进行建模，并把它们放到我们的环境中来。再次说明一下，在预视化处理中的什么地方设置可视化风格是非常重要的。我们不只是简单地制作任何一个街灯，而是我们正好需要一个街灯来补充和支持我们正在实现的或努力实现的可视化风格。《史酷比》——*Scooby Doo* 中的街灯和《银翼杀手》——*Blade Runner* 中的街灯截然不同，也很可能和你所见过的街灯完全不同。

那么，在我们的大教堂示例中应该用哪些道具呢？其中的一些在我们的地图中已经指出：

- O 需要一架管风琴
- A 需要一个圣坛
- P 需要几个木质长椅

我们还需要一些随意放置在环境中的各处的道具，以增加一些真实感。大多数教堂的内部装饰都不简陋，因此我们第一阶段的道具清单中还应该包括：

- 风琴凳
- 一些大烛台
- 墙上的挂毯
- 被冰冻的野兽雕像 1 和 2（两个版本）

在这里，那些被冰冻的野兽雕像是特别的道具，它们的作用就是给你讲述一个发生在教堂里的背景故事。它是被一个诅咒给冻住的，这些可怜的、倒霉的野兽成了冰冻的雕像而永存于世——除非你以后能够破了那条诅咒。听起来耳熟，是吧？

特效

接下来，需要考虑一下在我们的教堂示例中所需要的一些特效了。是的，即使是这些特效也需要和我们在预可视化阶段规划好的可视化风格相匹配。像《辛普森一家》——*The Simpsons* 中“河流”的特效看起来就和《关键报告》——*Minority Report* 中的“河流”有所不同。这是怎样处理的呢？一般来说，像《虚幻2》——*Unreal2 III* 或者《毁灭战士 III》——*Doom III* 这样的游戏引擎（在诸如输入/输出系统、渲染和动画系统、音效系统和游戏循环/游戏逻辑系统等核心代码模块的基础上构造）将这些特效（像冰从冰球鞋上飞溅出来的效果）作为基于平面贴图的效果或内置的粒子或动态系统的一部分来处理。

尽管我们还处在预可视化处理的第一个阶段，我们已经知道我们需要一些从视觉上来细化我们的环境的特果，列举如下：

- 寒冷的水汽（从被冰冻的野兽中冒出来）
- 神秘的光辉（环绕着我们圣坛）
- 蜡烛的火光（烛台上的）

特效增加了一定的视觉冲击和刺激，并且帮助告诉我们一些关于被冰冻的野兽的最新“对抗”（*confrontations*）。这些种类的视觉细节可以用多种方法进行程序化的处理，但是为了达到预可视化的目的，我们需要考虑哪些特果是我们必须构造的，以及我们所使用的游戏引擎会怎样处理它们。

脚本事件

我们已经取得了一些实质性的预可视化进展！我们已经完成了地形图、前期的关于材质和道具的思想以及特效的分析——而且，我们把它们放在一起进行交叉检查，以使它们符合我们视觉的定位点。这就是为什么要在整个产品周期中的早期就完成那些灵光闪现的概念图的原因。

现在我们必须考虑脚本事件了。脚本事件（*Scripted events*）是在游戏环境中我们想要包含的各种动作和行为，它是为了构造游戏的可玩性，组成故事要点，从非玩家角色（NPCs——像城堡里会说话的怪兽）引出各种行为，以及增强游戏情节。

对于前边的大教堂示例，我们可能会在玩家触动地板上的动作触发装置同时，让敌人从屋顶上冲过来；我们可能想通过打到或者射击圣坛图标时，由教堂的奴仆发起一次小规模进攻；我们还想通过按下风琴的键来揭示出某个地方藏有宝藏。这些都是基于脚本的动作。我们将在后面的章节（比如第6章）里进一步详细讨论脚本。这里需要强调的一点是，我们需要在预视化阶段就要考虑脚本事件，这样就能确定为了支持这些脚本想法，还需要哪些附加的艺术资源。脚本事件创造了额外的角色模型、道具或特效吗？一般来说答案是肯定的。这些脚本事件需要多少额外的资源呢？这些资源又是怎样和已构造起来的视觉参考点联系起来呢？

还是跟前面所说的一样，你在前期所做的任何预视化工作都能够为你节约时间，并且有助于避免整个开发过程中开发者们中间出现的混乱状况。



1.5 预视化案例研究评论

1996年，我所在的开发小组被请求为索尼游戏工作站（*Sony PlayStation*）开发一个动作/RTS（即时战略）游戏，时间期限大约是15个月。这个授权游戏发生在一个精心制作的未来世界里，必须马上进行产品开发。这类授权游戏，如“*Spongebob Squarepants*”和《蝎子王II》——*The Scorpion King II*，都必须尽快开发上市，因为通常它们都是倚靠相配套的电视剧和电影使它们为人所知、流行起来并具有购买吸引力：“你看到了吧！玩玩试试！”（或者是类似的想法）

在我们的那个案例中，由于那“富有挑战的”开发计划，根本就没有时间去完成一个完整的预视化过程。我们的小组以尽可能快的速度确定了设计和制作中需要考虑的事项，接着就开始开发游戏了。随着游戏引擎的细节和技术要素的“逐步成长”，游戏的设计同时也在“成长”。这些都没有什么特别的。我们有机会利用已开发好的角色并去构造新的角色。我们对环境设置和执行负有全部的责任。整个游戏世界和所有的角色都需要有一致的外观，“外观”规划失败时会很快表现出来。在我们的案例中，外观规划动摇了。在今天的各种控制台和PC机的视觉表现能力条件下，缺少一致的外观是完全不能够接受的。

由于连一个简单扼要的预视化阶段都没有完成，尽管我们尽了最大的努力，在开发得热火朝天的时候我们还是不得不中途停下来去考虑很多悬而未决的细节问题。这确实制造了一个拖延进度的因素，它使我们浪费了很多宝贵的资源。当进入了正式批准阶段

层次后，有时这种障碍几乎是不可避免的。尽管如此，作为游戏开发者，我们需要不停地去寻找一些方法来使这种开发影响最小化。使这种障碍因素最小化的一种方法就是尽自己最大努力在每一个开发计划中都要设有预视化过程。

ANDREW HOLDUN 访谈录



Andrew Holdun 是一位极其多才多艺而且很有天赋的艺术家，他曾就职于多家著名的游戏开发商，比如 LucasArts、迪斯尼和 THQ 等，并为它们创造过最受玩家欢迎的游戏的记录。我们可以在运行于个人电脑上的游戏“《绝地武士》——Jedi Knight”以及运行于任天堂 64 的游戏“《幽灵帝国》——Shadows of the Empire”中看到他的杰作。他在卜纳特大学获得了他的建筑学的学士学位。

TM: 准备进军游戏产业是不是就像在准备一场斗牛赛？

AH: 我读了很多欧内斯特·海明威的作品……也（为斗牛赛）喝过很多的龙舌兰酒。在你知道答案之前，假设我带着一件斗篷和一顶帽子，和这些长毛的愚蠢的球状物在一起，我在成千上万只鼻子里冒着热气的低等哺乳动物群中间疲于奔命。

TM: 好精彩！预视化过程通常是怎样完成的？它重要吗？

AH: 我想预视化在计算机图像处理领域的每一方面都变得越来越重要了——从电影、电视到网络和游戏开发。实际上，我想它是从电影和电视的商业制片中的故事情节串联图板逐渐派生出来的。

虽然情节串联图板在那些领域已经应用了很长时间了，而它渗透到游戏开发的过程仍然用了不少时间。随着游戏变得更具电影味和越来越复杂，就产生了一个需求：利用类似情节串联图板一样的工具来表达正在发生的事情——对剧情的创作者和制作人来说都需要这种工具。

当我在 LucasArts 公司的时候，艺术工作的主要任务就是对游戏中将要上演的内容进行概念化处理……角色的外观……环境的外貌。我们的大部分时间花费在了游戏脚本和反复试验的功能的实现上，并没有真正意义上的预视化，甚至连情节串联图板也没有。公平地说，我相信当时在电影工作室也没有做很多甚至根本就没有做预视化工作。

后来有了数字化工具，人们开始把情节串联图板分块，并将它们制作成电影、电视节目和游戏中的动作的数字化视觉显示。随着像 Industrial Light and Magic (ILM) 这样

的电影工作室开始更进一步地使用预视化技术，它们就从情节串联图板分块的方法（动画）逐渐转向了电影/电视节目/电视的粗略的三维视觉显示。可见其优越性是很大的。

在往工程项目中投入大量的资源以前，一个人可以切实地节省大量的时间和金钱，还可以试验各种游戏情节，并在投入资源以前进行察看。随着产品开发费用的不断增加，这几乎成了生产制造中的一个标准，而不像在几年前那样几乎没有人会这样做。

这也都渗透到游戏开发中。随着游戏开发成本、复杂程度和剧情质量的不断提高，开发人员对预视化技术的接受程度也在增加。实际上，当开发周期很短时这几乎是必须的。通常，完成预视化过程的方法是：从剧本出发到情节串连图板，然后把这些情节串联图板转变为三维的改造物。角色可以用线条粗略地勾画一个轮廓出来，环境也可以进行同样的处理。重要的一点就是能够产生一种可理解的感觉，让别人知道这个工程项目将是什么样的。

如果你能够把一个粗略的预视化作出令人振奋的效果的话，那么剩下的事（添加材质，提高模型、环境、灯光和音效的质量）只不过是小菜一碟。

TM: 为环境部分构造可视化参考的最好方法是什么？

AH: 无论是从网络下载的照片还是其他像《国家地理》（National Geographic）这样的巨大的资源库得来的照片，我都非常喜欢。而且，到图书馆和书店搜寻一些自己感兴趣的材料，比如像《完全技术：未来考古学指南》（曼弗雷德·汉姆等著）这样的书，也非常不错。

另一个重要的资源就是一台数码摄像机，你应该随时随地携带着它。你绝不会知道你什么时候会看到一座谷仓或一个锈迹斑斑的工业油罐而让你的眼睛为之一亮。你还应该随身携带草稿本和笔……这样你就能以不同角度记下关于某一主题的所见和所玩的东西。

TM: 构造一个成功的三维环境需要哪些类型的美术技能？

AH: 我是从一个建筑师起家的，因此我有着良好的训练基础，使我能够以一种更加完整的方式观察我周围的环境。我认为任何训练，像画画、雕塑和摄影都是进行三维环境创作的基本技能。

还有，不仅仅通过观赏各种类型的电影进行直观的自我练习，特别是科幻和幻想影片对视觉的冲击，还有就是认真地参观各种博物馆和舞台展览，在将来工作中，这些对于你构造一些有意无意可能会用到的三维可视素材极有帮助。

TM: 传统的建筑和“数字化”建筑是怎样联系到一起的？

AH: 它们有着很大的相似性。但传统的建筑受到现实中购买者和重力的约束, 数字建筑物不受这些约束。我们必须给游戏提供一定程度的真实性, 否则整个游戏就会土崩瓦解。

比如说你在开发一个赛车游戏, 这时重力就是一项必须考虑的内容, ……但是你可以通过数字化进行大量扩展并创造一些在现实世界中并不存在的环境。

TM: 你是怎样把你建筑业的经验应用到三维游戏创作中去的?

AH: 我利用这些经验来制作城市规划图。我想起了《幽灵帝国》中的 *Mos Eisley* 城市。而且, 使用计算机辅助设计 (CAD) 中的一些约定, 在建筑中也是非常平常的事儿……不同的物体要设置到不同的层上。

我还在三维游戏中使用了关于空间的形状 (一条隧道和一间天花板很高的房间) 是怎样在不知不觉中影响一个人的感觉的建筑学知识。

TM: 你是怎样为游戏中的各个关卡或各种场景构造建筑物的?

AH: 首先, 我尽力去领会游戏设计者的设计意图什么……要什么样的感觉? 然后的工作就更像是电影剧组的制作了。先以剧本和概念草图为基础, 大体上描绘出整个环境, 然后, 提供一些简单的灯光和材质处理, 并尽快在这个场景进行测试看它是不是有效。

这些场景是不是太大、太小、太单调沉闷或者令人眼花缭乱? 然后就仅仅是一个反馈、改进、再反馈、再改进的问题了。最后, 直到产生我们希望的那样一个满意的环境。

TM: “真实的”建筑平面图是否能够转化为令人兴奋的游戏地图吗? 你会考虑把它们作为你的设计的参考点吗?

AH: 完全可以, 但是我认为, 当一个人从一座现实中的房屋或城镇开始设计时, 最常见的情况就是他想扭曲它、将它夸大以产生某种效果。除非是你的游戏需要尽量接近现实的效果, 比如棒球运动场和赛车游戏。

TM: 说到基本的环境设计, 对于房间的入口和出口、房间流 (从房间到房间) 和从内部到外部空间的转变等这些问题, 你有何高见? 这些问题在大多数游戏中都会多次出现。

AH: 我认为空间流程必须或多或少地依赖于现实生活。在建筑学的研究中, 流程在所有的空间设计中都是要考虑的一个重要方面, 它和游戏世界中一样有着同等重要的地位。

也就是说, 有时设计者需要对现实进行扭曲, 以让玩家获得更丰富的游戏体验……

我正在考虑《半条命》，就拿它来说吧，其中的空间几乎是真实的物理空间的歪曲模型。

TM: 在开始构造图形之前，你会在纸上进行构造或规划吗？

AH: 是的，一点儿也不错。我先勾画出一幅平面图并确定一些高度信息，接着画出空间的粗略的轴视图（三维视图）。我来回不停地做……平面图、高度信息、三维视图，接着又回到平面图，直到将某些东西推敲好为止。

TM: 学习三维环境制作的过程是怎样的？应该怎样开始？

AH: 这个过程几乎就是一个反复试验的过程。我会到很多不寻常的地方和环境去旅行，不停地画图，拍尽可能多的照片，然后着手研究手边儿这个特定项目。接着需要对环境进行建模并亲自体验一下，看效果怎么样。不断地进行尝试，希望每次试验它都会变得容易一点，因为每次我都会学习到一些关于哪些东西使环境变得更好的知识。

TM: 是否应该加强游戏图形模块化程度？

AH: 这依赖于事物的本质。基于模块化，很多射击手都可以在游戏中凑合使用，但是为他准备更多（种类）的猎物更为重要。

但是，如果是游戏的需要，比如说大厦里的房间，把它们都模块化的话会导致不利的一面，就是游戏的空间独特性只能靠这个游戏的体验和迫使玩家在每个房间都要为不同的难题作出解答了。因此我认为这确实和你所开发的游戏种类息息相关。

TM: 模块化对游戏的开发速度意味着什么呢？开发者用这种方法构造游戏环境会不会更快？

AH: 我认为这确实是一个值得考虑的好方法。制作 *Mos Eisley* 城市的时候，我们就用了模块化思想，因为存在着—对相互矛盾的需求：存储能力——要求占用空间小，视觉复杂度——要求很好的视觉效果。

通过利用缩放和材质可以使同样的物体具有不同的外观，这样我们就可以在仅有几个模型的资源的基础上做修改，从而构造出各式各样的复杂外观。因此，模块化这条路是一定要走的。并且以这种方式构造关卡会比较快，因为你是用一整套的零部件来完成你的设计。这完全变成了拖拖放放……像乐高拼装玩具一样……从简单到复杂。

TM: 对于游戏环境，你把什么东西作为你的可视化灵感的来源？

AH: 我会在互联网上搜索一些关于一个总的主题的资料，而且我也会逛一些书店——特别是那些拥有大量的艺术类和建筑类书籍、杂志的书店。你绝不知道什么时候会有一些东西刹那间闯入你的大脑。

TM: 对于那些有兴趣进一步学习有关三维环境建模的人来说，你有什么小窍门要

告诉他们吗?

AH: 我想首先是找一些很酷的地方去观光旅行,并为这些到过的地方绘画和摄影,这是很有帮助的。接下来是,只是简单地敲敲电脑,做一些有趣的事……这大概就是我要对他们说的吧。

TM: 在三维创作中你最喜欢的是什么?是构造游戏(还是其他)环境吗?

AH: 嗯,我想有这种至上的观点(最喜欢什么)不会有任何的坏处。我开始构造真实的或虚幻的角色和环境,并在其他人与它们交互的时候仔细观察,再进行改进,这样它们就会变得比最初的版本要好一些。

我喜欢以那些最有活力的方式上进行创作,而且我喜欢在计算机上进行创作,因为这样我就能够以一种更加深刻的方式来感受我周围的世界。它以一种非常精彩的方式改变了我观察世界和与周围世界交互的方式。

TM: 随着像《无尽的任务》(Everquest)这样大规模的多人在线游戏(MMOG)风格的游戏的出现,开发人员怎样才能不断地交流大规模游戏环境的经验?其中的一些难点是什么?

AH: 嗯,我想在某种程度上应该会有一个像我们以前所说的模块化的组件。我也看到更多正在开发的游戏算法,你花费在环境上的时间越多它就变得越复杂。

关卡细节的规模也在变大。地形的生成将变得更为重要,因为它要真实地显示一些生物的形状,如植物和树木等。随着计算机的功能越来越强,这些都将成为不是问题的问题。

TM: 根据你以各种手段做开发的经验,你想给那些有兴趣从事三维环境制作的人提供三个什么样的忠告?

AH: 1. 疯狂地画画。

2. 把现实世界中你所感兴趣的环境和模型拍摄下来。

3. 不要放弃学习和研究美术、建筑以及雕塑。

TM: 感谢你花时间和我交谈!你还有其他技巧、建议、警告或者意见向我们说吗?

AH: 我认为旅行、画画和阅读应该坚持不懈。在所使用的技术的约束下,出于你能够构造的和你想要构造的之间的平衡考虑,你的设计可能会被付之一炬。离开你的计算机,去感受这个世界,这样你在三维制作中所做的就是你对这个世界的表达;不只是模仿某一个游戏的样子……要加一些有趣的东西……发挥你的想像并考虑有趣的事情。

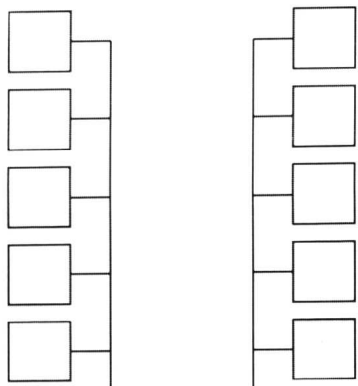
TM: 严肃地说,斗牛和游戏制作如何联系在一起?

AH: 在你的职业生涯中你将见到很多的“牛粪”……最重要的是不要让你的眼睛离开牛，但同时你还要留神不要踩到“牛粪”。

TM: 这让我想起了电影《大儿子小爸爸》(*Back to School*)中的罗德尼·丹杰菲尔德(Rodney Dangerfield)——“先为自己着想。不要管其他人。”试着在为自己着想同时也为其他人着想，为你的团队、你的思想和你的想像力着想。坚持走自己的路。Andrew，十分感谢你为我们做了精辟的分析。

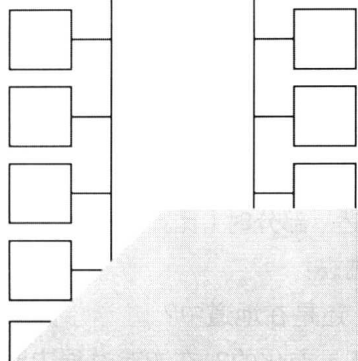
MEGA 提示

1. 如果资源条件允许，通常都要完成预视化序列并演练每一项设计。
2. 收集并组织丰富的参考资料，以支持预视化过程。
3. 通过预先构造所有主要的可视化参考点、准备一本风格宝典(一份儿正式文档，有时候是由授权人提供，有时候得自己创作)、并深入了解如何使用画面来支持游戏可玩性，为后边的开发工作节省时间。这个过程可能会演变，但是确定好参考点是有意义且有价值的。

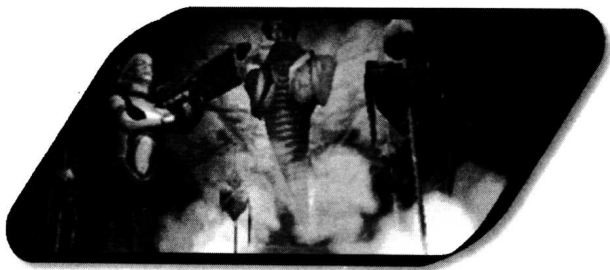


CHAPTER

2



关卡的规划及制作



当你完成了预可视化这一阶段后，就应该把所有这些信息收集起来整理到一个整洁的、随时可供查阅的设计文档中。假如你对如何书写完整的设计文档感兴趣，乐意花点儿功夫学习学习的话，你可以在 www.gamedev.net 网站上找到大量的设计文档模板，而且模板数量还在不断地增加。在设计文档中具体要包括哪些类型的信息，还要由你所开发的游戏的类型以及开发平台的详细情况决定。开发平台的细节情况决定了你的设计形式。开发平台的硬件将会决定设计中的大多数方面的细节问题。一个简单的射击游戏，是针对掌上电脑(PDA)或移动电话(cell phone)的平台设计还是针对 PlayStation2 平台上的第三人称视角的动作游戏开发，两者的定位有着很大的不同。

当我开始写设计文档中的“关卡或者任务描述”部分时(比如说是关于游戏内容的规格书)，我可能会试着把如下这些细节信息写进去：

- **场所** 游戏是在什么地方进行？在墓地？还是在地道中？
- **一天中的时间** 游戏中的时间是固定的还是变化的？在游戏过程中时间的变化是否会改变视觉效果？
- **天气状况** 游戏是在雨中进行？在暴雨中？下着冰雹？还是刮着大风？随着时间的改变，天气状况会发生变化吗？
- **故事情节的组织** 关卡或使命中为玩家设置什么样的主要故事情节？
- **故事的连贯性** 关卡或使命中哪些主要的故事情节需要突出？
- **使命或关卡概要** 为使命或关卡提供一份儿完整的总体“预演”描述。
- **地形图** 为你的关卡或使命提供一幅鸟瞰图。
- **特定的目的或目标** 游戏设置中要求玩家完成什么任务？
- **子目的或子目标** 如果有的话，玩家还需要经历哪些其他的小考验？
- **地形特征** 描述出来并加以详细说明。地形特征之间是如何相互连接在一起的？

- **道具** 有什么道具？它们是静态的还是可毁坏的？它们的行为如何？
- **危险物** 定义好游戏中所有的危险物。是掘开的坟墓？是倒下的树木？还是通电的栅栏？
- **谜题** 如果有的话，玩家需要付出哪些努力解谜题？
- **敌人** 这个关卡中要安排哪些敌人？它们的本领如何与环境交织在一起？
- **涉及的非玩家角色 (Nonplayer Character, NPC)** 关卡中会出现哪些非玩家角色？它们功能和目的是什么？
- **威力升级和收藏物** 有哪些威力升级和收藏物？它们功能和目的是什么？
- **效果的考虑** 要用到哪些视觉效果？疾病云雾 (Disease clouds) ？还是圣坛火光？
- **音效和音乐** 如何使用音效、周围环境以及音乐来烘托游戏的气氛？
- **脚本列表** 关卡或使命中需要什么脚本？
- **人工智能 (Artificial Intelligence, AI) 的考虑** 在单玩家模式中，玩家将会遭遇什么样的诡计多端的“进攻”？
- **网络或者多玩家方面** 为了支持网络或者多玩家模式，设计中必须考虑什么？
- **团队总体规划** 制定栅格尺寸以及工作程序 (workflow procedures) 。
- **游戏性的记录** 从团队发表独创性意见的讨论会中收集记录和思想。

上面这个列表看上去好像是一些需要确定下来的信息，的确如此。最终的成功取决于有一张可以帮助你设计进行到哪里的指示图。

一旦你已经在纸上完成了对关卡的思想测试和改进工作，并完成了你的文档设计工作，你已经为开始关卡的规划和布局过程做好了准备工作。

规划阶段能帮助你确定设计上和技术上的细节，而占位设计 (stubbing) 或者说布局阶段能帮助你设计出关卡或者使命的粗略的基本结构形式。这一章中，我们来检查将你的游戏想法从概念转化为最终付诸实施的工作流方面的一些事项。

在充分规划的情况下，游戏思想将会开始把你的地形图和参考草图转化成为简单的网格或者图形形式。这允许你在设计的早期就开始测试你的关卡的基本属性，比如说缩放比例、尺寸以及动作游戏中对跳跃的规定 (比如跳跃高度) 。

再次强调，跟第1章中讨论的一样，你要有一个出发点，并且要在纸上测试你的想法，这是很重要的。千万不要在没有任何准备的情况下就开始构造几何体。充分利用

你的参考草图或者参考点，发挥它们的所有价值。利用好你的设计文档——它在很大程度上帮助你当前进行到什么地方了。这看上去简单，但对许多游戏开发人员来说都是一个挑战。

制作游戏需要有始终向前的动力。如果你不知道进行到哪里了，那么你就可能不会取得任何的进展。对于大多数的关卡制作者来说，从草图开始制作远比仅凭乎想像和“自由构造”的制作关卡方法容易得多。下面，让我们看一看在开始制作关卡时应当牢记哪些东西。



2.1 规划你的关卡工作

现在，游戏业的投资风险非常的高，而且这个风险似乎每天都在升高。游戏环境必须快速地、高效地开发出来，并且要向玩家释放出最大限度的冲击。随着大型多玩家在网络游戏（massively multiplayer online games, MMOGs）的出现，为了进一步延伸、支持和增强玩家的游戏体验，开发商们必须不断地推出全新的“环境部分”或区域以及游戏升级，并提供给玩家。在第8章中你会学习到更多关于这些类型游戏的知识。

在过去的几年里，许多游戏都是使用基于区块贴图（tile-based）的系统构造的。区块贴图指得是类似“邮票”的基本二维方块，它们之间沿着其中一个边、两个边、三个边甚至全部的四个边相互连接或是联系在一起。它们像拼图玩具一样，一块一块地拼装起来。通过把大量这样的区块贴图按照预期的方式连接起来，你就可以制作出一个游戏关卡。最奇怪的是，虽然目前的大多数游戏都是在三维空间中制作的，一些一模一样的构造原理还同样适用。你或许会说，从设计的立足点看，保留下来的旧思想只有少数适于新技术。如果你仔细想想，你就会明白，三维环境通常都可以简化为由各种各样简单的部分组合起来的较为复杂的整体。

那些用于规划出你的关卡和环境作业的时间，通常会在它们的实施过程中的很多地方为你节省不少时间。盲目做事通常都会付出高代价却获得低效率。

游戏开发团队在挑选制作关卡的软件工具上可有多种决策选择。下面就是一些选择：

- ▶ 使用像玛雅（Maya, www.aliaswavefront.com）或者 3ds Max（www.discreet.com）这样的三维开发包中自带的脚本系统（分别是玛雅嵌入式语言——Maya Embedded Language[MEL]和 MAXScript），把定制的关卡工具功能构造到三维开

发包中去。你还可以把小范围或者大范围的关卡组件和编辑参数都构造到 3Ds Max 或者玛雅中去。

- 使用像 RenderWare (www.renderware.com) 和它的 Studio 工具集这样的游戏开发包解决方案。
- 使用那些带有授权引擎技术的工具, 比如 Unreal、Quake、LithTech (www.lithtech.com)、Serious Sam (www.croteam.com) 或者 NDL/NetImmerse (www.netimmerse.com) 这样的系统。
- 书写你自己的单独的编辑器, 用它来为你个人专用的引擎制作基本内容。

由于游戏开发者们背负着时间需求的包袱来努力开发新的、令人兴奋的游戏内容, 故许多开发者都在寻找基于预制构件、模块化和最大限度重用性的游戏构造系统。还有数个比例缩放和全局栅格的问题需要考虑, 还有一系列的易范的错误需要试着去避免。下面, 让我们详细看一下这些观念。

■ 预制图形和模块化

预制图形就是为以后某种方式的使用和重用而构造的简单图形。比如说螺旋式楼梯——在场景中你可以用它来从楼的一层过渡到另外一层, 就是一个预制图形的例子。它的用途不言而喻——它就是一个楼梯! 你也许会在一个完整关卡中的不同地方看到五到十次这样的楼梯。同一件预制图形一旦重新处理并对其大小缩放后再在其他地方使用, 玩家有可能会察觉不出来。

管道系统组件和门的造型是两个预制图形的例子, 它们可以在整个游戏环境中得到重用。目前, 为了开发的速度和效率, 我们不得不考虑使用组件来构造游戏环境。或许我们还可以在场景中通过翻转或者镜像的方法让门的造型变得不一样。这些预制好的单元有点儿和早些时候提到的二维区块贴图相类似。它们一块一块地“躺在”我们的玩具构造集中, 准备着构造我们的游戏环境。

为什么要考虑模块性? 因为我们需要“数字化的乐高拼装玩具”(可搭出房屋、车辆等)。我们必须快速、高效地完成游戏环境的构造, 同时还要使得游戏能够提供最大限度的视觉冲击和最大限度的可玩性支持。开发团队背负着很大的压力, 他们需要在短期内开发出令人震撼的游戏世界。尽管这看上去显而易见, 可也就是最近, 游戏开发者们才开始开发有着乐高玩具这样“掀扣组合”功能的高效工具。如今, 这些工具对于实现大规模的游戏环境来说极其重要。

尽管一些人会认为模块化设计是减少重复性工作的灵丹妙药，但是这并不完全正确。通过灵活地应用你的构造模块，你仍然可以表现出很好的创造性。那些你所钟爱的二维游戏都是使用模块化的区块贴图集构造的。三维空间赋予了游戏新的前景，但是也必须抱有某些同样的心态。

模块化场景设计是数个问题的很好的解决方案。如果我们不使用模块化的思想去思考，而且所有的环境特征还有所有的道具特征都完全单独定制的或是惟一的，我们将会：

- ▶ 饱受众多的机器执行性能问题的折磨。
- ▶ 不停地构造图形和材质、延长日程安排表，并且在开发过程中浪费大量有价值的资源。
- ▶ 不能高效地构造、改进以及对关键游戏可玩性的测试。
- ▶ 不能很快地完成相对较大的修改工作。
- ▶ 有可能会永远不能成功完成游戏的开发。

在规划阶段，我们需要按照模块化和重用的思想去思考。商业开发的时间表确实也不会给其他事情留出时间。我们如何才能花费最少的时间而获得尽可能多的收益呢？这是一个设计工程上的问题，一点儿没错。以往也一直如此！用于开发游戏的人力和资源都是有限的。规划如何以最少的资源换取最复杂的开发进展是开发团队经常要面临的挑战。

从第 1 章“大教堂”示例中所示范的地形图和基本物品分类，我们已经知道了一些定义关卡区域中这一部分关键的环境组件。这就是我们从概念到付诸实施的做法。

■ 比例和栅格尺寸的考虑

当使用模块化思想来考虑问题的时候，缩放比例通常都与游戏的种类息息相关。试问自己和你的团队如下这些问题：你想要完成什么样子的游戏？它又是哪种类型的呢？你会知道这些问题的答案的，因为你在此之前你已经完成了规划。然而，要确定你的模块缩放比例（相对于图形块儿，你的游戏区该是多大？），还要看你对这些问题的回答。是由地面上整个村庄的不同区域组成的地图吗？还是用许多较小的单独区块拼合起来的单个走廊和房屋内景？它们俩分别是大模块比例和小模块比例的范例。

如果你正在开发一个未来都市风景的太空飞船竞速游戏，你可能会拥有大量的预制件，比如形式各异的组合体中包含的全部建筑物块（译者注：指乐高玩具）。另一方面，如果你要让玩家通过隐秘的内部物理空间，比如废弃了的矿井隧道，你需要明细的组件

或者建筑件，使用它们显然可以很容易地组装起复杂的事物和细节。

从概念上讲，这种方法完全适用于第一人称视角和第三人称视角的动作游戏。它不但适用于有赛道或者比赛场布局的赛跑类游戏和体育竞技类游戏，而且对于那些取决于地图的详细设计和生物族群的角色扮演类游戏（RPG）和即时战略类游戏（RTS）也同样适用；它甚至也可以用于模拟类游戏，比如说《动物园大亨》——*Zoo Tycoon*和《过山车大亨》——*Rollercoaster Tycoon*。所有的这些游戏都可以以使用图形构造集制作出来的地图或者关卡为特征。这种方法还可以和飞行模拟器设计中的一些要素一起使用，比如说机场和飞机棚的布置。“飞行”的感觉是用游戏引擎做过一些特殊处理的，这种游戏引擎要有生成专门定制的卷动和镜头移动的功能。

栅格系统（Grid Systems）在游戏开发中占有很重要的地位。程序员们使用栅格来定位，并处理一些其他编程上的问题。另外，栅格也决定了全局比例。如果角色定为128个单位那么高，那32个单位高的栅栏就达不到预期的视觉效果——和Stonehenge（史前时期的巨大石柱群）被小矮人摧毁一样不相称。图2-1中显示的是全局或整体栅格比例设置对话框。对话框中的设置表示每一片儿图形都与栅格上每16个单位对齐。图2-2是编辑器中的栅格顶视图的一个例子。

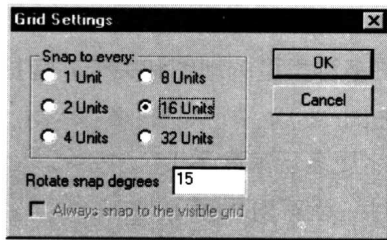


图 2-1 全局栅格设置

而且要记住，角色的动作以及角色之间的相互作用都与栅格密切相关。角色通常可以跳过一定数目单位的距离；它们可以够到一定数目单位以外的东西；它们可以爬上距离地面一定数目单位的梯子；它们一步可以走过一定数目单位的长度。栅格通常与游戏密切相关，所以在开发游戏环境时我们必须自始至终以我们所选择的栅格系统为参考。开发者们应当尽早统一并确定合适的栅格信息。否则，最后的结果可能是大量的关卡重做工作，这可是任何一个人都无法忍受的。记住下面这一点很重要：你的设计文档应当详细说明每一个动作 / 对象之间的相互作用。

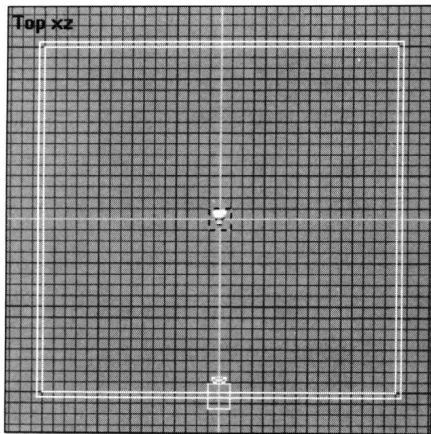


图 2-2 顶视图 (Top view) 栅格

你的动作 / 对象间相互作用是你制定栅格系统的指导方针的一部分。最重要的也许就是：用来组建我们的关卡场景的预制组件必须是根据大家（指的是开发团队）所共同参照的栅格系统构造的。

■ 避免常见的关卡错误

现今的游戏中出现了许多种类型的地图和关卡，包括：

- 单玩家（玩家对计算机）；
- 多玩家合作（两个或者更多玩家联手）；
- 死亡竞赛（Deathmatch，玩家对玩家）；
- 抢旗变奏曲（玩家对玩家或者团队对团队）。

这些关卡类型中的每一种，在设计上都有数个与其他类型不同的细微差别，而且这些差别中多数仍在研究之中。在开始一个关卡的构造工作之前，让我们先来快速浏览一些在构造关卡时经常会范的错误。下面这些仅仅是一个概括，而且它们也不一定在每个案例中都会出现。

- **草率设置的死胡同** 突然地或者不可思议地“截断”关卡流程的死胡同。
- **难以使用的房间流程** 房间到房间的过渡不起作用。千万不要把整个关卡设计成这样：一连串一间挨着一间的医务室，而中间是一条走廊。在狭窄的空间内无休止地进来、出去，没有一点儿趣味性。要记住，认真规划你的空间到空间

的过渡。要确保给玩家提供多样化的选择空间，帮助并指导玩家通过小空间到大空间的过渡、小空间到中等空间的过渡、中等空间到大空间的过渡，等等。在关卡中提供多种类型的空间过渡。

- ▶ **一次性应用所有的敌人或者威力升级** 因为你可以一次性地使用所有这些东西并不意味着你一定要这么做。一般来说，你想要调整应用敌人的速度，按照剧情的发展应用敌人。构造时要考虑到玩家胜利的希望。让你的玩家一下子看到前边“提供”的所有敌人并不是一件令人兴奋的事。当然，我们也不是说要在屏幕上一次性放上尽可能多的敌人。比如安置敌人以使玩家产生畏惧心理，以及设计一些有趣的游戏情节，这才更加接近我们的主题。
- ▶ **滥用开系统** 不要使用过多的开关。没有人愿意在每进入下一个走廊都要打开一次门。而且，不要设计太棘手的开关谜题来迷惑你的玩家。如果是我的话，我会立马停止玩你的游戏。
- ▶ **没有使人惊讶的事** 在哪儿我有机会发现些什么东西？不要让所有的东西都显得平淡无奇。我喜欢寻找那些隐秘的和未透露的秘密，我想要去探索这个你带我进入的游戏世界。给我一些幽默的内容，让我进入一个不为人知的关卡秘密吧！
- ▶ **没有得分点** 在我努力战胜敌人并通过陷阱之后，我得到了什么？我想要得到一个满意的奖励。给我小而形象的报酬。当我费了很大功夫进入了新的区域，我想要一个形象的得分点，这是最起码的要求。在关卡中放置一些形象的得分点。确定在哪里放置这些用来奖励玩家进步的得分点。
- ▶ **让玩家陷入困境** 如果我可以成功地通过你的关卡，你就不要“告诉我内幕”或者“帮助我”。拿我来测试你设置好的敌人出现的间隔、简单问题的解答以及敌人的威力。不要测试我的方向感，我就没有什么方向感。使用道具和材质作为标识（“呵呵，我记得这块儿蓝色的地板！”）来告诉我在关卡中的什么地方。在你的游戏界面中做一个简单的位置定位器。两个你都做了，我就能完成你的游戏。如果两样都没做的话，那你就别想了。
- ▶ **没有目标或者是太多的目标** 尽量不要让我感觉不知道在你的游戏世界里要干些什么。帮助我知道我的目标，而且不要把目标设计得太老套。另一方面，我一旦知道了游戏的目标后，就别再打搅我了。不要让一个非玩家角色或者解说员时不时地跳出来提醒我：你还没有找到你要找的东西。

- **指导我的每一步** 明智地使用非玩家角色。不要时不时地告诉我要做什么、什么时候去做、以及如何去做。尽量不要强迫我做一些事情，因为买你的游戏不是想体验被强迫做事情的感觉。指导我，但是不要强迫我。

■ 关卡占位设计 (stubbing) 初排

在预制组件、模块化、有用资源分类（所需的环境特征、道具、材质、以及脚本事件的列表——如同我们前边所举的“大教堂”例子）、以及栅格等问题都解决了以后，我们就可以开始构造关卡了。记住，游戏开发者们使用许多不同的软件工具来完成关卡的构造和环境的构造。考虑到示范的目的，这一部分将向你展示一个最通用的做法。在这一阶段，我们开始把我们的地形图转化为早期的关卡图形或者关卡网格图。在上下文中，图形和网格图就表示环境特征的多边形线框模型，比如房间的定义，或者空间、拱形物、楼梯等等。图 2-3 显示了一个基本的线框视图。

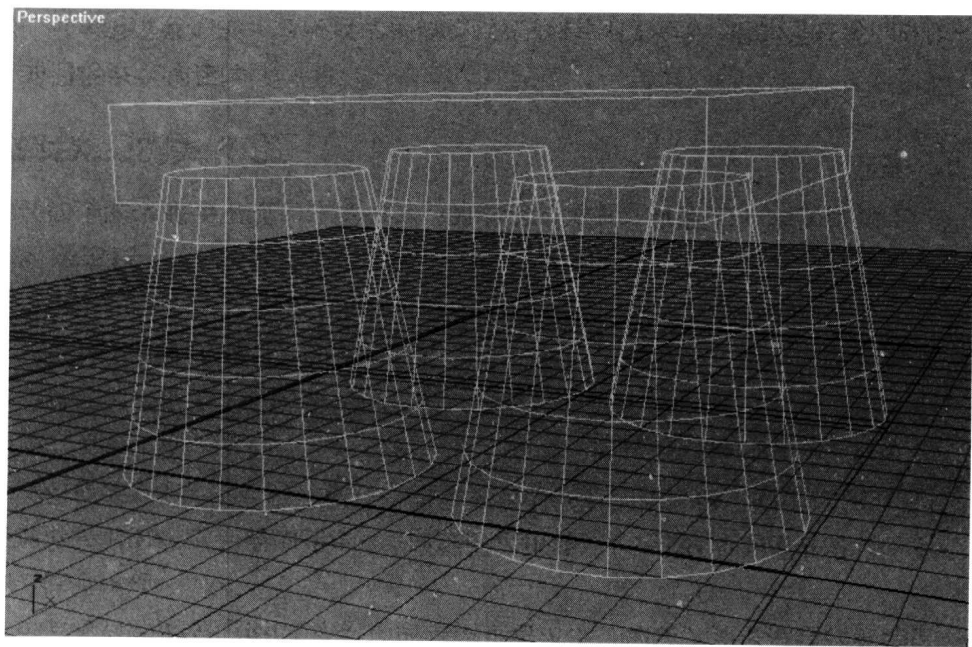


图 2-3 线框视图

有一个重要的实践问题需要在这里强调一下。和前边所讲的一样，当游戏开发者们

开始为游戏构造环境图形时，他们在软件工具的选择上有很大的空间。他们可以使用商业三维开发包，比如说玛雅或者 3DS Max 以及这些开发包内建的用于用户定制的脚本系统；他们也可以使用单独的关卡工具，这些工具是为了生成与他们自己的游戏引擎相兼容的资源而专门制作的。通常这两种工具他们都使用，不过有数种不同的组合方式。

图 2-4 显示了玛雅的工作场景，场景中的实例是使用从 Alias / Wavefront（译者注：已更名为 Alias）——玛雅开发商（www.aliaswavefrong.com）那里得到的一些游戏开发者培训资料制作出来的。

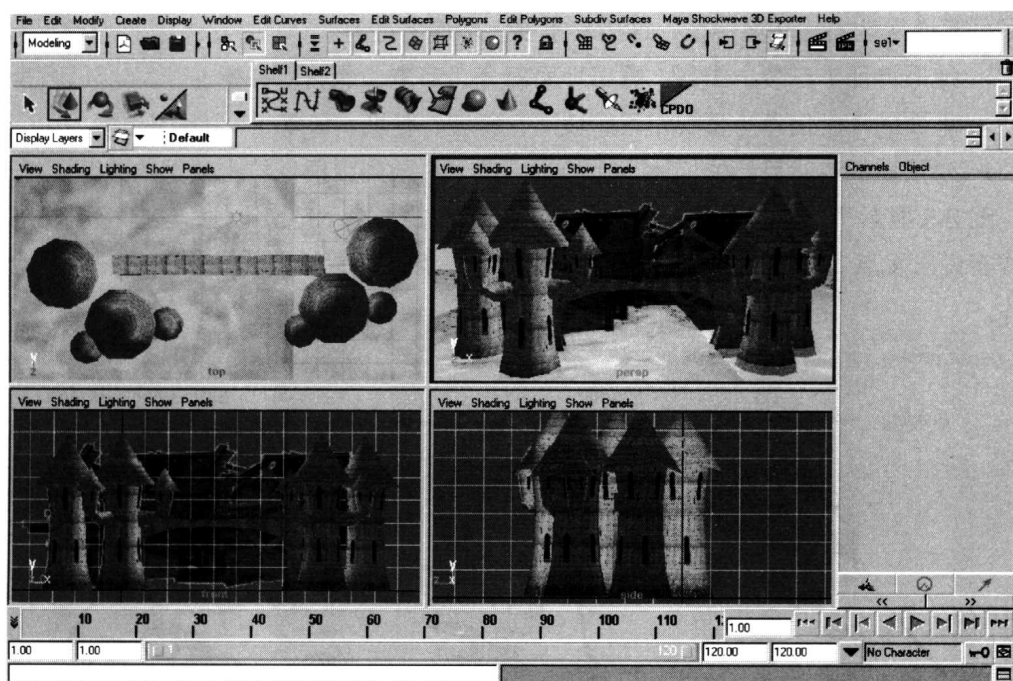


图 2-4 玛雅的工作场景

一些开发者选择使用玛雅来完成物品和关卡的制作，而其他开发者却选择 3DS Max。一些游戏开发工作室将这两个开发包合在一起使用。在最终的分析中，你选择使用最适合（高效）实现你所需效果的开发包。或许你的程序组更喜欢使用 MAXScript 来编写工具。这种情况下，你的团队可能就应该选择 3DS Max。现今的这两个开发包都非常地灵活、易于扩充、而且功能强大。

接下来，我们要考虑一下构造早期的关卡图形的过程，也称为“占位制作”关卡。这一过程通常都伴随着一个相当通用的五步程序。你可以修改这个步骤，以满足自己内容的需要。

■ 第一步：从你的地形草图开始工作

“占位制作”关卡的过程中第一步就是要确信你有一份儿关于你的地图的非常棒的地形草图，草图中描述了关卡的大体轮廓和一些物理特征。它可能包括总体的道具布置、初略的敌人编排、遇到的障碍和危险、以及威力升级等的标记。在这一阶段，模块性应该尽可能详尽地描述环境特征，这极其重要。环境特征有：

- 地形特征（丘陵、山谷、露出地面部分、河流、池塘，等等）；
- 地面的高度信息（楼梯布置、坡道、斜坡、陡坡，等等）；
- 对预期的游戏目标的描述（比如说，奋力拼杀到山顶，抢占敌军的战略要地）。

图 2-5 显示了你开始工作所需要的地形图草图的一个示范。记住，这些都仅仅是参考或者说出发点。我们要从宏观的关卡规划出发，努力朝着关卡的具体细节脚踏实地地工作。

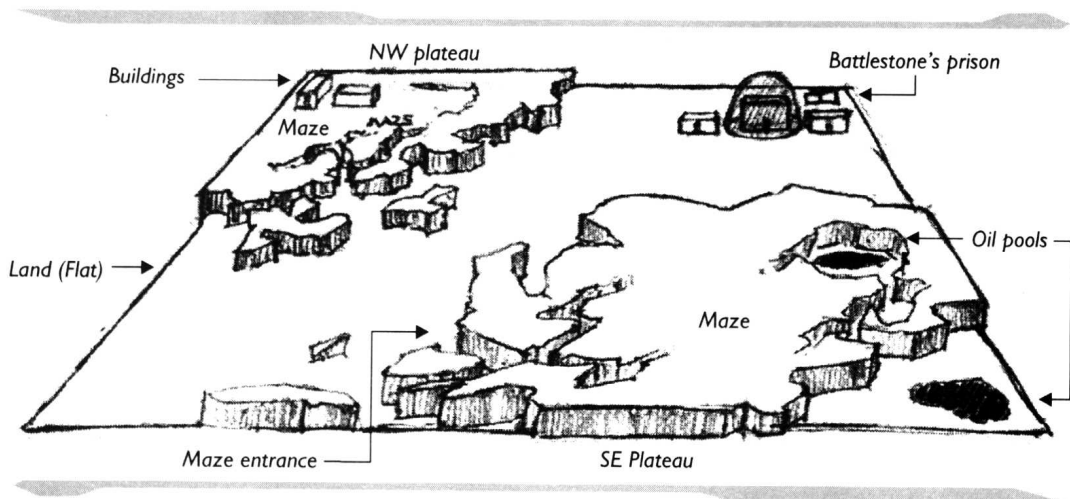


图 2-5 地形图示例

整理概念草图以及收集快速参考资料这些准备工作，你预先做得越多，构造环境图形的任务就会显得越有条理。这应该没有什么让人感到奇怪的，但是安排紧凑的时间表，会使得开发人员在游戏开发最紧张激烈的时候，中途放弃掉这些工作。在通常情

况下，开发人员根本就没有做大量、全面的预制作（preproduction）的时间，预制作与游戏开发是同时进行的。这就是为什么能够制作快速且易于扩展的概念构造草图，以及对规划物品使用的模块化构造常常显得这么重要的原因。

■ 第二步：生成等高线

你或许会选择把作为模板使用而绘制在纸上的地图扫描到图像文件中。把这幅图像加载到 3DS Max 或者玛雅的背景图像平面（几种造型的通用着手点，包括环境和角色的制作）。从那里，你就可以描绘出你的图像的等高线外形轮廓，从而生成关卡中的主要特征的样条路径或者外形轮廓。你只要有了这些等高线，就可以通过抬高和降低曲线来调整你的所有的高度信息，从而得到你想要的三维等高线。

图 2-6 中显示的是等高线的绘制过程，而且，图中这些等高线的最终位置包含有基本的“高度”信息。如果你从屏幕上的各个视图（顶视图、侧视图、正视图和透视图）来看的话，你可以看到它们现在都成了三维线条了。

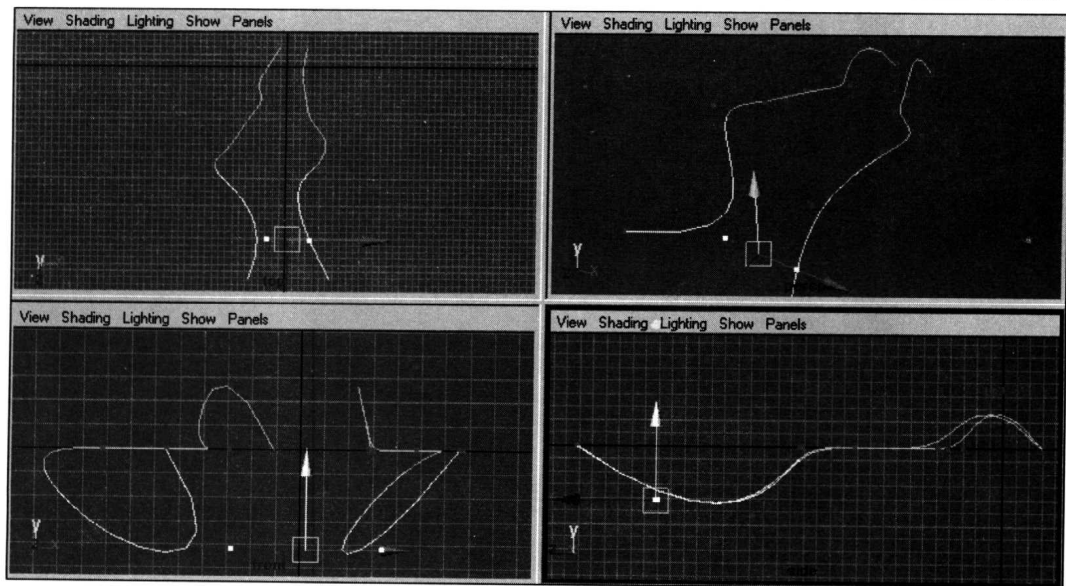


图 2-6 绘制等高线图

■ 第三步：用你的等高线构造网格

如果你是用图像平面背景中的图像文件来绘制你的等高线的，那现在把图像文件隐藏起来，并用你所构造并调整过的样条曲线构造一些基本的图形。我们只是想构造一个简单的网格图，让玩家在它上面试玩，以检验关卡的一些基本要素，比如尺寸和缩放比例问题等等。

图 2-7 显示了使用“放样”（loft）操作（一个简单的造型移动操作）将等值线变成多边形网格。你可以看到，现在我们有了一个简单线框模型，我们的环境开始初具规模。

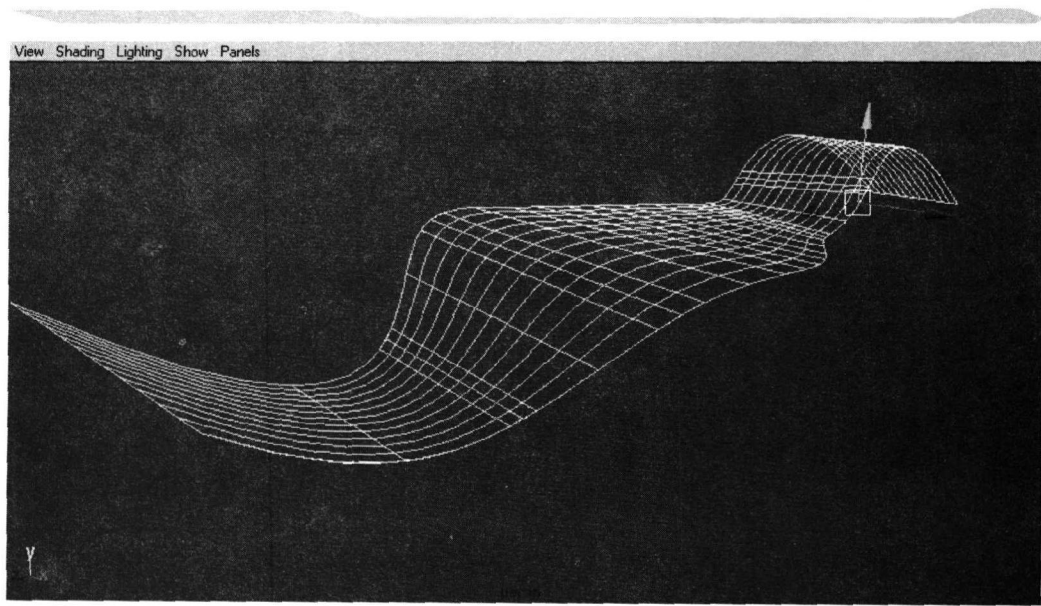


图 2-7 由等值线生成的网格图

■ 第四步：利用文件参考构造和拼装关卡

现在，你可以开始生成那些常常被称为“文件参考”的原型文件了。文件参考通过一个按钮与某些高度重用的图片相关联，允许你在地图或者关卡的任意位置“压印”或“实例化”出对象来。它就是模块化和预制规划真正体现的地方。

把文件参考按钮想作一个饼干成型的切割刀，你可以在整幅地图上压印图片，就像

在饼干生面团上压印出饼干样式一样。你一旦在饼干生面团上压印过了,你便已经完成、不能再修改了;而如果你在地图上压印一个模型出来,你还可以随意地移动它。噢,数字化的工具真是太棒了!

举个例子,你想要在地图上的一部分放置一组堡垒或者圆盘式卫星天线,然后在另一部分放上桥梁件和岩石墙壁。这是使用文件参考完成的。

假设你完成了一个拱道的部件,准备在整个关卡中重复使用。我们早已知道,巧妙的资源设计能提供最大程度的重用。我们一直在努力让每一件做好的资源发挥其最大的功效。你可能会生成一个拱道部件的标志符,但是拱道部件模型的最终版本或者“游戏中”(in-game)版本是由三维技师完成的。

为了便于说明问题,让我们假定你已经有了现成的拱道部件最终版本。当你已经构造了一个名称为“拱道”的文件参考按钮,将它与你的拱道部件图相关联,这样你就可以将它在整个地图上随意摆放并自由拖动,就像建筑集中的组件一样。这种方法在构造地图和关卡时非常有用。

文件参考使得你可以通过高效的模型重用来实现大量的细节和复杂的内容。这也会节省大量的内存,因为参考组件或者参考模型一旦被加载到计算机内存中,游戏引擎就可以把它放在所有需要的地方,并且基于一个实例计算机就知道如何绘制每一个实例对应的物品。继续吧,在每一个需要的地方都放上“拱道”部件。每一件都不仅仅是惟一的一件;每一件都有着最大程度的重用价值,为了可靠的运行效果和构造复杂的内容,我们需要重用。

■ 第五步: 开始检查布局细节

在整幅地图上放置像拱道部件或数列遮盖部件的实例化物品后,你的关卡画面就开始成型了。你可以根据团队统一的栅格系统来做一些早期的位置调整,并且可以开始在地图上操作移动一个角色或者临时角色(有时候就用与预定角色的栅格尺寸一样大小的立方体来表示)。在这里,你主要是检查尺寸、缩放比例、栅格问题,以及不合适的图形(部分网格有错误或者可以简化)。



2.2 游戏原型设计

游戏原型设计是目前游戏开发过程中所面临一个最大的挑战。如何从概念到实施

充分检验或者说“验证”你的游戏想法？现如今，在进行实际的游戏开发之前，游戏开发商们需要可靠的证据证明游戏想法是有趣的、有吸引力的、很好的可玩性，以及能为企业创造商业利润。

开发商通常都要求游戏开发人员提供一份儿精细而全面的概念验证报告（Proof of Concept, POC），来示范他们所要开发的游戏的可用版本。对于大多数单独的开发人员来说，准备这样一份儿概念验证报告送交开发商需要花费巨大的代价，而且这并不能保证可以得到开发协议。

游戏思想的测试到目前为止还没有简单实用的解决方案。举个例子，假定你的团队对构造三维立体（isometric）视图的竞赛游戏感兴趣。这个游戏的基本中心就是要将传统的车对车的速度竞赛放到立体视图的场景中。四分之三的画面或整个立体镜头视图让玩家立刻观看到赛道的全貌；或许你还要让玩家能够放置障碍来阻止对手，并收集威力升级、为赛道上的每一位在他们附近的对手制造大麻烦。你该如何测试这个游戏想法？如果你连一个完整的、可快速定制的游戏引擎都没有，或者即使你有，你还是要为完成像这样的游戏想法的概念验证报告而扔掉相当可观的资金和资源。

缺少成套实用的游戏引擎，就根本没有办法进行有效的游戏想法测试。从定义上讲，一些游戏想法要比其他的想法更容易进行原型设计。对环境游历和简单的解谜游戏来说，原型设计或许会相对简单一些。我们可以使用与 Macromedia 公司推出的 Director 类似的软件对像《迷雾之岛》——*Myst* 这样的游戏进行原型设计。然而，原型设计一个三维升级版的拱街动作游戏，比如说 *Smash-TV*，就很难，因为这需要相当多的测试编程和物品资源对它进行测试验证。

原型设计在其他面向娱乐的产业的应用也相当普遍。在市场上销售的每一种大型的玩具产品都是经过严格的原型设计、检验、改进、和再检验的。我们已经看到了，在游戏开发产业做这件事情并不那么容易。

有时候，开发人员用以前的游戏版本使用过的工具测试将要开发的游戏的新想法。可是，如果开发者考虑要做的游戏类型与以前的游戏完全不同的话，那些以前游戏已制作的工具和编写的代码也许就没什么大的帮助了。

游戏产业界内没有什么通用的“游戏构造”原型设计软件。测试游戏想法通常都需要大量的代码资源，为了从各个角度全方位测试游戏的想法，这些代码一般来说都必须针对想法专门编写。



2.3 关卡占位设计的后处理考虑

利用文件参考、模块化以及预制组件三者组合起来的威力，你现在已经占位制作了你的关卡的基本要求。很厉害，不是吗？但愿你已经明白了预视化和关卡规划阶段的重要性了。和往常一样，你正在努力将游戏概念和想法经由关卡图形制作而转移进入实际行动。要是有一个令人惊叹的环境用来游历的话那简直太棒了，但是我们还需要更多其他的好东西。我们想要在一个很棒的环境中完成很多很多有趣的比赛。关卡占位设计阶段的早期工作可能是令人兴奋的，但这仅仅是漫长的改进工作和交叉检查周期的开始。

从这里再往后，我们需要集中精力调整总体结构和地形特征，使它们以可能的最佳方式支持游戏的可玩性；而且我们需要检查整个新的关卡中角色流（character flow），（比如角色在场景中漫游）。台阶、物品和梯子放置的高度是否适合我们的角色去登、跳和爬？我们的环境中放置的危险物的比例、目的以及设计意图是否恰当？整个组合起来看，我们的关卡的各部分大小比例是否恰当而自然呢？正如你所看到的，这一块儿有许多特征和多个方面需要检查。

我们还得检查我们的概念。在处理这些细节当中，我们是否一直在考虑整体的游戏画面？在关卡开发中，早期的改进过程需要保持一定大小的前冲力，否则在细节问题的激烈争论中，我们的想法和目标会趋向于迷失方向甚至被冲淡。而且，在我们还没有走得很远的时候，我们想要能够在有需要的情况下作出一些大的调整。这就说明了获得玩家和测试人员的早期反馈意见的重要性。最后，当关卡目标开始变得越来越明确的时候，这些目标会不会支持和增强我们对游戏最初的设想和中心思想？接下来我们将会挨个讨论这些重要的问题。

■ 调整总体结构和地形使其适合于游戏

如果你已经正确地做完了大部分的栅格规划，你就要把将来调整或改变游戏中的总体结构特征或地形特征时那些不可避免的重叠工作量减小到最少。记住，这些特征应始终支持游戏的可玩性！这就是为什么对于你和你的团队来说，从纯粹的游戏可玩性的立场出发，设想并坚持记录将要发生什么事儿，是那么重要的了。如果你忽视或者逃避了这项“脑力演习”，而且还没有收集来自于它的一些细节，你的游戏可能会以有一个非常精巧的总体结构布置但却没有任何玩耍价值而告终。现在，如果你是在为客户构造结

构总体，那么，做一个漂亮的总体结构布置确实没有任何的错。但是，你要知道你是在制作游戏！而我们需要到的也不仅仅是一个总体的布置。

但愿你不要把出入口、窗户或者输送管道设计得太小，以至于主角都无法进入或退出。当你的游戏中需要一座能通过车的桥梁时，但愿你不要把它设计得太窄而无法并排开过两辆车。然而，尽管你已经针对你所设计的游戏规划了明确的图形和结构形式，并且为它构造了一个预制系统，你仍然会发现你的建筑集还是需要删删补补。把你的关卡建筑集看作是你的军火库，这一点很重要。试验一下。尝试用它制作出一些惊人的东西来。我希望，通过规划，你已经将草图转化成为深入制作关卡的精彩起点。

可是，事情并不总是如此。有的时候，尽管你们每个人都很优秀，团队也是团结一心，但是你规划好的内容在三维中却没能很好地实现。草图你画了，想法你有了，游戏可玩性的记录你也做了，但是最终作出来的结果却并不理想。没有关系，我们正在考虑基本的穿梭和游历，这才是这个阶段我们应该做的。记住，现代游戏设计有一大半的时间都是花在加工和改进上。

在模块化和文件参考的帮助下，你应该能够以很快的速度完成大量的调整工作。它们都是强有力的思想工具。如果你使用其他方法构造关卡图形，你可能会发现自己得手工编辑那些复杂的图形。也许你想要调整图形的大小或者裁剪图形，而后来你又需要准确无误地将它们拼合回原图——许多棘手的问题都可能出现在你面前。无论我怎么强调文件参考和模块化威力都不过分。

结构的模块化 / 预制系统的使用，使得游戏可玩性方面的总体结构编辑工作变得越来越简单。举例来说，假如我已经在整幅地图上粘贴了二十多座寺庙；现在如果我想对寺庙做一点修改，而且没有使用文件参考，那我就不得不在地图上挨个把这二十座寺庙找出来并手工修改过来。这简直就是一个杀手！可是，有了文件参考，我就可以仅仅修改一座寺庙——“寺庙”的文件参考的定义文件——地图上所有的这二十座寺庙就一下子全更新了。哇！简直太棒了。

另外一个例子：假使我想在我的地图或者关卡的所有地方粘贴某一种类型的地形图片。比如我有一张有岩石凸出的草地印花，岩石的高度已知；假设我在当前的关卡中共使用了四十三次这张印花。由于游戏可玩性的选择，现在我想要改变岩石露出的高度，该怎么办？也有这样的可能：其中有几个印花实例要让角色能够爬过去，而在这栅格上当前的岩石的高度太高使得角色根本过不去。要是能够仅修改这几个，而且还不用手工修改，那该多好哇？我可以制作一个独特的文件参考非常快速地完成这项任务！

■ 检查整个关卡的角色流

角色流指的是角色在环境的适应程度。角色是否可以登台阶、爬梯子、跳跃以及按动开关？角色的大小和技能与危险物或道具放在一起是否相称？

当你已经完成了基本的关卡布局，你就可以开始在关卡中移动一个大块头家伙到处走了。只是开个玩笑，请别当真。根据你的游戏规划，你可能是移动一个 128 个单位大小的东西（这个大小由你的栅格设置决定）。因为这个时候支持游戏引擎的动画系统可能还没有完工，也可能是角色的动画没有做完，或者是二者兼有之。就这样吧。

我们在整个关卡中到处走来走去，是在找什么东西吗？对，我们是在寻找关卡中存在的一些基本问题。比如说，有没有纯粹的图形错误，结构是否“整洁”（有没有多余的或者不需要的图形），有没有东西可以简化的，是否有东西可以做得更漂亮、更小巧或者更整洁。我们是在做早期的感观上的以及整体比例的检查工作。与我们的角色相比，寺庙和池塘的大小合适吗？如果我们做了完整可靠的栅格规划，那这些工作就应该可以结束了。尽管如此，说不准还会有一些需要改动的地方。

我们设计的内部空间和外部空间之间的切换是否能正常工作？我们有没有把关卡或者地图中用来结束或终止某一个物理特性的某件东西忘记了？熔岩会流向何处，它流到哪儿停下来？怎么停下来的？它绝不会像在我们的地形草图上一样在图纸的边缘处停止！在这儿我们可能需要在我们的建筑集添加一些过渡件。

毫无疑问，我们的角色或者我们的运输工具要在游戏环境中完成一些的事情，这一点没错吧？我们想要在早期就开始查看这些行为动作。角色可以使用大门的开关，而且可能会与环境进行一系列的交互。我们的栅格系统的细节中，有一部分应该用来定义这些活动发生在什么单位高度。这是为整个团队规定的设计信息，所有人都必须遵守执行。其他设计者或许正在制作其他关卡，但是，比如大门开关的高度，所有的关卡都必须统一。环境中饮用泉水的单位高度是多少？为了让角色能够抓得住，梯子应该放置在离地面什么单位高度的地方？

我们不得不在关卡开始成型的时候就检查整个关卡范围内的每一件东西。我们想要检查下坡路和上坡路、电梯的入口点、梯子、开关，以及阀门等。在这个时候，我们也许乘坐在用来代替小船或者圆筒的简单立方体上横渡池塘或者河流，因为到现在为止小船或者圆筒还没有“竣工”。这也说明对于游戏设计者来说，掌握基本的建模技巧为什么那么的重要。

另外，这个时候关卡仅仅完成了占位设计或者说尽可能完整地勾画出来了。然而，在这个过程中我们更乐意早些而不是晚些时候才去发现存在的问题。我们还需要做一些平衡工作（trade-offs）并权衡游戏价值。这是实现完善的游戏概念的一部分。

再一次强调文件参考的威力：我们可以用占位件（placeholder）文件代替对应的物品，以快速地模拟我们的游戏想法。当我们拿到了最终的作品，把它们替换掉就行了。

■ 跳跃高度、危险物以及缩放比例

我们都注意到了，许多游戏中都有不断的跳跃动作。我不仅仅是在说 *Frogger*。跳跃高度是另一个关于角色与环境之间各种交互的好范例。跳跃高度就是指栅格中的一个给定角色从起跳点可以跳起多少个单位的高度。这个单位数目可以在游戏编程中修改或者变化，从而使角色跳得更高或者低一些。游戏中的跳跃高度值很重要，因为它和所有的角色互动一样，当到了关卡占位设计过程快要结束的时候，它们就成了要测试的对象。

一个角色通常都有定义好的初始跳跃高度。玩家如果用喷气式长靴、“超级跳”（SuperJump）或者“反推式等离子移植体幸福鞋底”（reverse plasma graft happy soles）（你应该有这张图）来装备角色，那么角色的跳跃高度就会增加一倍、两倍甚至更多。

假设初始跳跃高度是 64 个单位，这就说明角色不能跳到任何超过 64 个单位高度的物体上。也许角色的步幅高度是 32 个单位，这意味着在制作动画时，此角色的步幅动画就得按照 32 个单位高度的步幅构造。你可以看到，要使得动作看起来更恰当，遵循你的世界（或者说总体）的栅格系统是多么的重要。也许角色“按动开关”的动画是对应于安装在 96 个单位高度的开关，它就应该在 96 个单位的高度处有一个生动的按动动作。如果游戏世界中那些开关都安装在 47 个单位高度位置，那么，角色用手指按动墙壁的动作就会显得笨拙不堪。

跳跃高度、步幅高度、按动高度、拖拉高度、推动高度、捕捉高度和梯子的高度——所有这些高度描述量都需要在你准备结束纯粹的关卡占位设计或者有突破性进展的时候进行核对。也许有一项任务调用了—一个没有确定高度信息的事件，那现在就该考虑这个高度信息了，并像处理占位件信息一样把它确定下来。

当占位设计阶段交付完成并且关卡也逐渐确定下来的时候，我们最好是开始着手撞击事件（collision events）的实现。当一个角色在平地上走着，比如说他撞到了台阶，这时就会引发撞击事件，比如我们想要使用这个事件来触发角色的跨步动画。对于同样

的事件，我们也可能想要触发使门被使劲关上的事件。通常情况下，根据你的游戏引擎和工具集的情况，设置或者描绘出撞击区域，为程序员生成数据。

像火坑、沸腾的油池、或者酸液容器这样的危险区域，它们的比例和流程都需要检查。这些环境特征从概念到实施的转变过程是否成功？这就是你要检查的。现在几乎没有一个类似这样的游戏，所以这个很难讲清楚。稍后在游戏测试中，为了实现完美的游戏布置效果，这些区域都需要作进一步的调整。在第7章中你将会学习到更多关于游戏测试的内容。

■ 始终留意游戏画面的整体效果

或许你正在给一个高尔夫球游戏构造或者占位设计高尔夫球场，或许你是在开发一个新式的车辆竞赛游戏，还有可能你忙碌于第二次世界大战期间或者是在古老的西方的第一人称射击类游戏。在所有的这些情况下，留意游戏画面的整体效果是很重要的。确实如此！

换句话说，交叉参考（cross-reference）你的概念图、游戏中心思想以及游戏可玩性记录，这是很重要的。当然，通常情况下你是从游戏设计文档出发开始制作游戏的，它是你的游戏设计的路标和“细节骨架”。设计文档详细说明了实施你的游戏的目的、功能和方法。（附录A提供了一些关于设计文档的参考信息。）

假定我们手上有一份儿制作精美的设计文档，而且刚刚制作完成我们的游戏第一关的要点，我们如何留意游戏的全局画面？这在很大程度上讲就是一个智力训练。起初通过从你的关卡的一小部分着手，把你的文档和参考资料当作跳板，试着在你的“心眼”（mind's eye）中描绘出你所希望的游戏在你面前玩耍的情形。如果仅从游戏的一部分入手看上去更容易的话，你就这么做，并努力让它在你的大脑中充实、丰满起来。获得更多的游戏可玩性记录，并和你的团队一块儿分享、改进它们。记住，这个过程就是一个精心加工和改进的过程！

在你脑海中的描绘出你和你的团队或者是你自己一直讨论的、梦想着的、着了迷的角色与环境相互作用的方式。试着去领会游戏可玩性的定义本身。是防御机制的吗？（例如，看守这堆战利品。）是谜题机制的吗？（解决一些逻辑或者物理的限制，从而完成任务。）还是进攻/防守机制或游戏操作？（进攻敌军1和敌军2的同时还要保护好你的大堆战利品。）只要付出足够多的努力，你一定会体验到顿悟“这就是它的工作方式！”这一时刻。那会是一个让人兴奋不已的时刻，棒极了。想像它将会如何工作的。试着想

像一下：浮现在你脑海中的是游戏机制是怎样与你的关卡、竞技场，或者球场融合在一起的。在你的大脑中的大屏幕上“观察”所发生的一切。

当你锻炼出这种技能，你就能够在你的大脑中“移动镜头”，有时候甚至还能以不同的角度观察关卡。第一次这样做会感觉困难，这是很正常的，请相信我。对于大多数人来说，尝试三维空间中想像是很难的（二维已经够难了）。我想你会掌握脑力映像技术，它在“察看”（因为我找不到更好的词了）游戏机制以及“察看”你的关卡时的确非常有用。在逐渐培养你的这种能力的过程中，你有可能一次“察看”到比刚开始更多的内容。你会看到竞赛系统的各部分是如何工作的，或者一个特别的、可爱的鲸脊柱形的微型高尔夫球场是怎样制作出来的——而且最重要的是，你还能看到它为什么适合于这个游戏。

■ 关卡的早期演变

这里当我谈及“关卡”一词，我指的是从二维开发平台上射手与安置在屏幕最右边的头目，到健壮的纯三维第一人称或者第三人称视角关卡之间所有的游戏类型。根据上下文，我甚至还会讨论到即时战略类游戏以及角色扮演类游戏的地图和使命。

关卡、地图以及使命这些概念的出发点都旨在为一个上设拱图的游戏视觉提供支持。从概念上讲，也许我们是在努力再现历史上的一些东西，比如说第二次世界大战中坦克大决战的一部分。也有可能我们的关卡是用简单的争夺机制或者快速动作机制驱动的。我们的任务就是努力构造这样的关卡：以尽可能最佳的方式和尽可能最有趣、最富有挑战性的方式支持这一事实。

我曾经提到并建议过规划的重要性。我们得要知道我们进行到哪里了。然而，在游戏开发过程中，最终目标的实现是一个不断演变的过程。如果我们能够保证事情会和设想的一样，完全按照游戏中一个给定的关卡规划执行的话，那就太棒了。情况看起来好像很少会这样。千万不要误解我的意思，我们仍然需要这里的每一个规划和概念。还有，游戏的基本原理也突然冒出来，并以数种方式显示出它们的重要性。关卡，作为整个游戏的大组件，也以数种方式演变着。

技术力量塑造了我们设计，并限制了我们的思维。作为一个设计团队，你们不能做那些代码做不了的事情。作为一个在生产环境中工作的设计团队，你们不能做那些纯粹的资源水平所无法承受的事情。也许你会有很好的“游戏支持的理由”去实现某一技能或者特征。事实上，由于技术的限制、难以解决的困难、经验上的问题，乃至其他的一

些因素，使得它难以付诸实施。

关卡的演变有两个重要的方式。第一个：它们有时根据你对游戏的洞察力进行演变。这是最有可能的情况。在给予足够多的精心改进的情况下，它们会按照你所希望的方式进行演变。你是基于纯粹的技术支持的角度对游戏思想进行测试和否定或者测试并加以改进的。第二个：它们以你绝不会选择的方式或者是你不喜欢的方式进行演变。在这些情况下，由于一些“软因素”（soft factors），比如说软件工具的限制、编辑器的限制、游戏引擎支持的限制，以及尚未解决的技术问题等等，你有时候就会完成不了你所追求的那种特殊的游戏玩法。优秀的关卡、地图和使命有幸在优良的技术“肥料”——一个功能十足的游戏引擎技术和工具高效集成的健壮平台的“呵护”下茁壮成长。

一个为个人电脑或者下一代控制台系统开发的游戏引擎，基本上就是大量复杂的C++代码模块集，用于处理诸如玩家输入/输出、作图或者渲染系统、动画系统、音效系统，以及大型机的主游戏逻辑循环这些东西。由它们通知其他系统要做什么以及怎样去做。软件工具是那些构造间或管理那些要整合到游戏引擎中去的图片、动画，以及音效资源的工具。这些工具的功能太强又通常会限制关卡的演变。至于游戏方面，一个游戏开发团队能够构造的游戏可玩性仅限于这些工具所支持的以及编程队伍所提供的那么多。程序员总是有许多并发的、面向一些附属功能的开发任务需要完成。没有人会选择把游戏可玩性扔在地上不管。

■ 早期信息反馈的重要性

如果你已经完成了基本的关卡或者全局图形，并且对关卡全面检查中所出现的空间流程问题、图形错误、栅格问题以及尺寸或者缩放比例问题都做了校正，那就到了向你说明交流反馈循环的重要性的时候了。

大多数游戏开发团队在开发过程中都会遇到一些技术问题、工具支持问题、以及基本执行问题，这些问题最好在早期就解决掉。在早期的构造阶段结束后，与此同时你的关卡也开始生长并成型，在这个时候，团队作为一个整体，如果设计组能将他们的相关反馈信息和参与其他会影响到设计细节的开发工作的其他组的成员交流和共享的话，那将会非常有价值。

如今对于大多数团队来说，在设计实现关卡方面的一个普遍的限制因素就是软件工具的支持。或许你正在使用一个成品游戏引擎或者是半成品游戏引擎。（成品引擎运行起来不会出现任何大的错误。半成品引擎运行中会有一些甚至许多个已知的错误，这对

于半成品引擎来说是很常见的情况，因为它的完善和修改是与你正在构造的游戏内容是同时进行的。)即使你的团队的程序员完成了成品游戏引擎整体构架，而且美工也在使用(和选择)成品的三维开发包，但是游戏的构造还需要太多的工作，而决不仅仅是把你的模型和动画资源简单地放入到游戏引擎中去(这同样有一定的挑战)。

问题就在于设计的实现或者游戏中心思想的“加入”通常都对工具的支持有很强的依赖性。如果你所使用的工具没有能够完成你的团队的游戏视觉这一功能，你很有可能会以失败而告终。为了完成设计目标，你的团队最重要的事情就是反馈信息到你的工具组(如果你够幸运，你的团队恰好有这个组)以尽早获得适当的支持。

基本的机制行得通吗？

在构造阶段的开始，请尽快问一下自己这个非同一般的问题，这一点很重要。而且要公正地看待这个问题。给你的游戏一个到达适合于问这个问题的状态的机会。如果它行不通，考虑一下为什么。是概念在实施过程中发生变化了？是你所使用的软件工具不能实现你的团队的设计思想？还是程序设计需求难以实现？

尽管是在做脑力劳动，从一开始的预视化、在脑海中构思游戏可玩性，你还是要想一想：游戏是不是不完全不像你所设想的那么有趣？每一个开发人员都会用大脑匆匆地做一遍这样的检查和质疑。你必须尽早提出这些问题。一旦由你的游戏中心思想所限定的基本游戏机制行不通，你就必须马上作出决策：立即改进或者否决。迫于开发的压力，多数开发团队为了让游戏向着成功的方向推进，为了获得更好的游戏体验，他们不得不想尽所有可能的办法修改所有可以改进的地方。这已经到了一个尽力而为的境地了。

■ 你的关卡目标是否增强你的游戏视觉？

在整个游戏开发过程中，团队都要时不时地检验着游戏的目标。不管你是在为你的游戏构造地图、关卡，还是使命，都要仔细看看你的关卡的“肉”(即目标)，看看它对增强你的游戏视觉或者支持你的游戏中心思想有怎样的作用。这样做你的团队一定会受益匪浅。如果你的关卡中有一些与游戏体验无关的内容，你一定想要把它们拿掉。作为一个团队，这就是为什么试着理解你的游戏将会成为什么样——是那么的重要……还有定义好的要素。

简单明了无论在游戏中还是其他一些地方都是一种美德。假如你的关卡有太多的子目标——如果每次当玩家完成某件事，就给玩家添加另一个小的子任务(“现在去做这

个……”)——关卡的这种“有目标”的感觉就趋于减弱。记住,总是尽可能保持目标的清晰。如果你没有这样做,玩家很快就会感觉不连贯,并会开始问自己“这又要让我做什么?”

不要迫使玩家做一些与他们努力尝试的乐趣毫无关联的事情而使得他们的游戏体验支离破碎。大多数游戏的可玩性都取决于情节要素,如果你每隔一分钟都要打断一次情节,这个游戏就会让人感觉不和谐。



2.4 关卡实施的案例研究评论

1997年,我正在为索尼 PlayStation 开发面向动作的即时战略游戏。这个游戏有许多使命,而且每个使命又有数个需要完成的目标。当时并没有觉得我们在每个使命中设置的目标数过了头。我们肯定不会有那么多的子目标。事实上,我们只是在每个使命中都设置一组目标,玩家必须完成后才能结束使命并继续玩后边的游戏。

有时,在这种风格的游戏中,为了让玩家在整个使命中都兴致勃勃,你好像需要精心构思许多目标。用目标来指引玩家的行动,这没有任何直接的错误。事实上,大多数玩家在领会应该向着哪里行动方面都有着很好的感觉。

这全在于一个平衡问题。许多游戏玩法的决策最终都归结为整体局势的平衡。你的伙伴是结合玩家的反馈来作出这样的决策。在这种基于玩家反馈的情况下,游戏的演变是从玩家最喜欢的动作要素的想法开始的。玩家们在执行他们的动作中仍旧想使用战术,但是动作本身才是他们的“享受”的中心。简言之,玩家们似乎并不需要一个精心设计的目标系统,他们也并不想要冗长的使命讲解。这其中的部分原因是即时战略类型的游戏在个人计算机上的过度流行,而且这类控制台游戏为了吸引玩家的热情仍然在不断提高其动作要素。

知道了这个事实,我们就不会再添加任何新的目标,事实上,我们还拿掉一些目标,并把我们的最后的努力集中在游戏玩家最可能喜欢的那一部分:面向群体的动作要素。我们并没有丢掉战略的一面,我们仍然在每个使命中都贯彻了基本的战略思想。我们只是用改善动作项目的方法来调整游戏的平衡,比如说构造数种新型的面攻击抛射武器、增强我们团队式的组合进攻能力等等。

我们团队面临的另一个大的负面因素,就是我们在对关卡进行修改时所陷入的“困境”。我们所用的工具不稳定,我们有一个错综复杂的制作过程:编辑关卡、用游戏代

码重建关卡、然后再检查修改后的结果。我怎么强调工具的重要性都不为过。差的、错误累累的，或者设计拙劣的工具都可能让一个游戏死于胎腹中。

JOHN KRENG 访谈录



John Kreng 是柬埔寨驻美国大使的儿子，他精通汉语、越南语和英语。他从孩提时代就接受了空手道训练，并获得了韩国唐手道（Tang Soo Do）和日本柔术（Te-Katana Jujitsu）的黑带三级（空手道的技能等级）。John 两年前就成为了美国第三级联赛的选手以及参加南美洲巡回比赛的美国职业空手道队的成员。他还上过《空手道内幕》和《空手道评论》杂志的封面。

在电影特技和打斗拍摄中，John 曾经合作过的人有斯蒂芬斯皮尔伯格（Steven Spielberg）、德比艾伦（Debbie Allen），香港杰出人物如徐克、李连杰，以及带有传奇色彩的凌空钢丝打斗大师袁祥仁。John 曾在电影 Enter the Grill Master、A Party Called Earth、Hard As Nails（刀锋武女）、The Haunting of Slaughter Studios、Shakedown，以及 LA Times（《洛杉矶时报》）的袁祥仁传略宣传片中担任过特技替身演员 / 武术指导。

在动作 / 武打序列编排中，他的风格和方式是更多地了解关于一个角色的个性、力量以及弱点，然后按照这个角色的个性特征定制动作。通过这样做，这些已编排好的动作场面将会围绕着情节主线的组织进行，并用肢体语言讲述一个故事，它在支持和加强画面的同时还牢牢吸引住观众的情感。

有着丰富知识的 John 打小就热爱空手道电影。他领头对动作影片制作人以及最著名的武打演员进行了深入的访谈，并为 World of Martial Arts（《空手道世界》）、Kung Fu / Qi Gung（《功夫 / 气功》）以及 Impact（《影响》）等杂志完成访谈录的撰写工作。他采访过的人物有成龙、洪金宝（《过江龙——他来自上海》）、李连杰、知念美智子（《我的幸运星》）、袁和平（《黑客帝国》、《卧虎藏龙》）和于仁泰（《白发魔女传》）等。他还是由 TC Media 公司出版的李小龙 25 周年纪念刊的协调编辑、项目总监及作者。

John 还是一位有造诣的相声演员，他四次在国家电视台登台演出，他是电影艺术科学院年奖的提名奖得主 Renee Tajima 所拍的一部美国公共广播公司的纪录片的主人公，并且是第一位出现在 BET's ComicView 的不是非裔的美国人。大约有 10 年的时间，他经常在好莱坞、加州 La Jolla 及拉斯维加斯的 Comedy Store 俱乐部演出，而且他还是

埃迪墨菲、路易安德尔森和波利肖尔的引荐人。

TM: 相声、世界级的空手道比赛、还有大量的武打片知识对你的游戏设计有什么样的影响?

JK: 让我们从我怎样进入电子游戏产业谈起吧! 那时候, 我正在旧金山拜访我的朋友 Jeremy Cantor (他和我一起获得黑腰带并一起进入武术学校), 我跟他说起 20 世纪 90 年代初期到中期这一段时间我该如何走过的喜剧生涯。这是因为全美国 95% 的喜剧俱乐部都关门了, 而且我几乎错过了下半年所有的预约, 那时候才刚刚六月份。再加上当时我还正在闹离婚, 我需要安定的生活来认真想想接下来的事情。那时候 Jeremy 在世嘉公司工作, 他问我是否想过做一个游戏测试员。这个职业我从来都没想过, 至于有没有这样的职业我也一无所知。他认为我是这个职位的最佳人选, 第一, 因为我是一个武术家, 而且打小我就是个香港动作影片的狂热爱好者, 而每个游戏都多少会牵扯到一些动作要素, 正是这样, 我知道怎样能使动作序列看上去更好以及如何制作动作序列; 第二, 作为一个相声演员, 我知道在各种场面制造一些有趣的事情; 第三, 我的写作经验对于做这项工作也是大有益处的。在熟练掌握写笑话的方法之外, 我还参加了美国加州大学洛杉矶分校的电影剧本作家的学习班。有了这些经历, 我就能够轻松地把游戏的样子和感觉描写、表达出来; 最后, 是因为我有美术背景, 我能很快地发现图形或者动画中错误的地方, 并帮助美术师进行修改, 使它们在游戏中更有效地执行。

为了寻找坐落在洛杉矶的游戏公司的地址, 当我返回到洛杉矶的时候, 我就去了一家大的计算机商店, 看了几乎所有的电子游戏盒的后盖。假如你是洛杉矶一家游戏公司的负责人, 你会收到我的一封热情洋溢的信——里边描述了我的各项技能以及我对游戏产业的这份儿工作的渴望。洛杉矶就只有那么几家游戏公司, 而我获得了两次面试机会, 其中一家雇用了我, 其他的就都成为了历史。

事实上, 就是我所提到的这些技能把我从“问与答的地狱”(指求职应聘)中拯救出来, 并让我以游戏助理设计师的头衔开始了我的第一个设计任务——SNES 系统(超级任天堂娱乐系统, Super Nintendo Entertainment System)上运行的 *The Mask* (《蒙面侠》)。有一天, 在我经过一位美术师的办公室的时候, 我听到他生气地嘟囔着。我问他出什么事了, 他告诉我他遇到了麻烦, 一个角色的踢腿动画很难做。我就说是否可以让我看看, 他把东西拿出来让我看。那是一个做跳转勾踢动作的角色, 他告诉我这个动作需要再快一点, 而且它看上去没有很大的威力。所以我告诉他在快要接触到目标的地方去掉几帧, 加长角色的臀部以使他的身体更高一些, 并缩回或者弯回那条不踢出去的腿,

给人以跳得很高的假象。因为当时我只是一个地位很低的测试员，他开始并不相信我的话。所以我在他办公室里向他示范了修改方法并向他说明了我所提出的几个问题后，我让他一个人待着，好让他消化消化我给他讲的这些东西。我还暗自琢磨：“在他眼里我只不过是个刚进门的测试员，他肯定不会听我的！”没过多久，他就过来把我拽到他的办公室。他向我展示了他更正后的动画，并问我是否和我向他解释的一致。哇！他听取了我的建议。他向其他美术师展示了他的修改，他们都很喜欢。这位美术师很正派，他告诉其他的组员是我帮助他解决这个问题的。再后来，其他的美术师也经常过来请教我了。

我还记得当我第一次被选为喜剧俱乐部（*The Comedy Store*）的正式演员的事儿：喜剧俱乐部像是一个喜剧学院，我只是靠听听喜剧演员们讲的故事在那儿混日子。我就像一块儿海绵，只管把所有东西都“汲”进来。像山姆金尼森（Sam Kinison）、阿森诺豪勒（Arsenio Hall）、戴蒙韦恩斯（Damon Wayans）、乔治卡林（George Carlin）、理查德普莱尔（Richard Pryor）等等这些喜剧演员。在听他们讲故事的过程中，我得到了很多你需要亲身体验才能获得的经历，不管你在做什么。只要有这样的体验，你就可以直接回想起你的经历，当你把它表达出来的时候，你的感想会更加清晰和强烈。如果你准备为你的游戏构造场景或者环境，如果没有真实的生活体验，你拿什么给玩家真实的感觉呢？

TM: 当进行游戏设计时，你会建议用什么样的总体“设计方法”？

JK: 我问自己下面几个问题：我想要玩家得到什么样的体验？玩家将会怎样感受他们的情感？如何把所有这些情感协调到单独一个游戏中？是什么让这个游戏不同于现有的其他游戏？什么要素会让玩家完全沉浸在你的游戏环境中？是什么东西让你的玩家再次光顾你的游戏？你是否要用这个游戏构造一个亚文化群？每一关卡或者每次胜利玩具能得到什么报酬？这些报酬对于玩家艰苦攻关值得吗？还是会让玩家感到失望？我喜欢以这种问问题的方式开始着手我的设计过程，因为你要围绕着你想要让玩家体验的情感来制作游戏。我感觉这才是更具有生命力的设计方法。

TM: 你做过大量的打斗类型游戏的开发工作（可想而知），请问哪些是打斗游戏设计首先要考虑的？你是怎样把你的丰富的武术知识融入到游戏中的？

JK: 我确实为游戏设计带来了普通人所不知道的对技术的一些理解。在武术这一行摸爬滚打的整整三十年里，我见过、读到过和学过许多不同的武术派别。用不了多久，你的大脑就能“识别出”在屏幕上看着很真实的东西。对那些不懂武术的人来说，这是

打败他的简单的方法。这一点非常重要。

TM: 在你所见过的下一代（现代的）格斗游戏中，你最喜欢的和最不喜欢的分别是什么？

JK: 不幸的是，它们还是老样子。看上去好像每个角色都是从成龙和袁和平的电影中摘出来的。图像要比以前的8字节和16字节年代更好看，而且也更加平滑。我看到他们在游戏中增加了更多的扭打移动，这为游戏增添了更多扩展空间，而且使得游戏更像UFC（Ultimate Fighting Championship，顶级搏击冠军赛）和Pride Fighting（擂台格斗赛）比赛了，但是格斗类游戏中的扭打移动不由你控制。目前它们还仅仅是动画，在将要受到攻击的时候你不能做转身/后退移动。

TM: 作为一个实实在在的搏击手，你希望在将来的格斗游戏中看到什么？什么内容最会让你感到兴奋？

JK: 说实话，我现在很少玩格斗类的游戏，仅仅是因为我更愿意去武术训练学校，通过自由搏击亲身体验格斗，对我来说这才是更好的实践和很棒的锻炼。我观看和研究过大量的游戏，但不幸的是，他们大体上都是一个模式。我更喜欢看基于实时搏击战略的游戏，理解更多的格斗中的策略以及如何才能通过夺取中心线从而操纵对手。不同类型的步法、做假动作以及拳打和脚踢结合声东击西，这样会很棒。这里还有一个战胜你自己的问题——即调整心理——以便在比赛中处于对手的上风。我所讨论的是一个人的思维过程，它可以使你失败，也可以使你势不可挡。这就是为什么你需要挖掘培养一个厉害且灵活的头脑从而超越自我。即使你能超越你自己，这也并不意味着你总是能打败你的对手，因为他可能会比你更厉害。这些就是一些我想要在格斗游戏中看到的东西，一些额外的维度。

TM: 你曾经说过，跟日本游戏开发人员合作和跟美国本土的游戏开发人员合作有很大的不同，这怎么讲？当你把你的经验应用到游戏设计上时，其优势是什么？劣势是什么？

JK: 和日本开发人员一块儿工作，他们会筹划好所有的事情——甚至一直考虑到生病的日子和家里办丧事的日子。而和这里的许多美国人在一起，通常大多数的制作人都不会有这样的一个游戏规划，这是一种“走着瞧”（run and gun）心态。特别是当市场部人员开始运作他们的“死亡”最后期限计划，这就显得有些无可奈何，因为他们有一个最晚的发布日期或者这个游戏与一部即将上映的电影有关。不巧的是，市场部人员并不知道游戏制作中涉及到哪些复杂的东西。带他们去参观质量保证设施以及走访制作

组成员也不会使他们获得全部的经验。这并不能让他们明白制作中会涉及哪些难题，而他们对此也确实一无所知。很多时候是游戏本身受到“惩罚”，因为制作人不得不将测试阶段中出现问题的要素拿掉，因为否则它们将不能满足最后期限。

TM: 以你对他们的观察，你认为美国本土的游戏设计人员应该向日本游戏设计人员学习些什么东西？

JK: 和我一块儿工作过的日本设计人员和制作组对待工作都一丝不苟，每一件事儿都计划到精确的细节。他们有一份儿他们赖以生存的游戏策划，所以我们没有必要为他们担心。而我们应该享受周末和夜晚的大多数情况下，我们很有可能在和我们的美国制作人忙碌着解决那些棘手的问题，如果我们事先做好计划的话，这些问题根本就不会出现。

TM: 如果有的话，你注意到现在的游戏中什么样的负面倾向可以通过更好的设计加以避免？

JK: 我感觉市场太多地主导了游戏产业。它们把游戏分门别类，或者是体育竞技游戏、打斗游戏、射击类游戏，或者是以一部电影为背景为了“制造”轰动效应开发的。有一条大多数游戏都要遵守的规则，它会使得开发出来的游戏特别地单调沉闷，就好像是克隆现有游戏的思想一样：“我喜欢卖得很好的热门新游戏，所以让我们以此为前提，靠我们自己做一些类似的但又与众不同的事情，这样我们才能赚到钱。”我希望看到更多不同以往的玩游戏体验。

TM: 游戏设计是一个竞争激烈的产业，一个人应该怎样做各方面的准备工作？

JK: 了解现在有些什么样的游戏；培养对这些游戏的洞察力，你可以带着挑毛病的心里去玩游戏；你可以把每一个游戏短处和长处都标记出来。李小龙曾经说过的一些话我至今都在受益。他说的好像是：“吸收所有有用的东西，摒弃所有没用的东西，把学过的都变成你自己的。”

TM: 好有影响力的论断啊！你以前为很多游戏做过武打编导，现在你在为电视节目和电影做武打编导。你可以简单地谈一谈它们各自的过程吗？

JK: 在电影中，我会紧跟着导演。我问他们这一场要什么类型的打斗，滑稽型的、严肃型的还是其他类型。我认真阅读剧本，更深入地了解打斗中各个角色，这样，我就能够为角色定制武打动作，从而补充和支持角色的发挥。比如说，如果角色在感情上很冷漠，我就不会让他们做任何的扭打或者近距离接触的打斗。除非它是故事或者角色安排的一部分，不然我会让他们保持一定的距离并使用朝鲜式的踢腿、长冲拳以及扎手法

交战。这样，我一旦明白了是什么样的角色参与打斗，我就在大脑中组合出一套打斗动作，想像它与屏幕上的风格有什么不同。我还会想出打斗后角色会发生怎样的变化。

为了使打斗场面能给玩家留下深刻的印象，你就得让打斗从一种情感开始，而以另外一种不同的情感结束。到要与演员会面的时候，我已经对料想要发生的事情有了一个总的想法。当我与演员第一次见面的时候，我会通过让他们做一下基本的练习来估量他们的身体技能。只要他们确实是用自己的身体或者大脑生活的话，我就可以在五到十分钟之内说出他们的身体技能。如果他们全身心投入，通常情况下他们更容易训练，因为他们把自己所有的感觉技能都开启了；如果他们没有全身心投入，事情就变得有些困难了，因为这种类型的人只利用了自己的一种感觉，他们的大脑到身体之间的连接信息传输不正常，或者是太慢。大约有三分之一的人没有全身心投入。通过第一次会面，我通常会让这些演员理解一套基础的打斗动作，我这样做的目的是让他们对自己所做的工作感觉得心应手。然后，到了要拍摄的时候，我需要确信他们是否已经熟悉了动作，因为他们应该练习很多遍了，他们看上去应该不再是自然的和自发的那种样子。这就是我在拍摄打斗场面时与演员或者特技师之间的合作方式。还有很多的其他细节我没有谈到，这些东西都可以写厚厚的一本书了。

为游戏做动作捕捉，我得先听一下武术师怎么说，并看看他们的基本要求，以了解他们是否对他们的动作有一个完整的基础和结构。然后再看看他们如何对拳打和脚踢作出反应。再然后我看他们把这些动作拼接起来，看他们是否能够由一项技术转向另外一项技术。在做动作捕捉的时候，确实需要重复性的工作，因为你不得不一而再、再而三地反复修改。幸运的是，其危险系数很低，因为我在同一时间只让一个人做动作。惟一可惜的就是为了完成动作捕捉镜头的所有动作，你需要耗费大量的时间。而在电影中，你做几次特技表演（这具体由特技本身而定），然后就可以走了。

对游戏产业我所不喜欢就是工作时间总是紧张忙碌，而在游戏完成后又不能马上获得报酬。对于电影和电视剧，我们有SAG（Screen Actors Guild，美国屏幕演员协会）和AFTRA（American Federation of Television & Radio Actors，美国广电艺术家联合会）的保护，而且每次插曲或者片断在电视上播放我们都获得重播报酬。游戏中没有一项和它类似，惟一能与它相提并论的就是红利支票了。

TM: 为什么制作精良的打斗类游戏对众多玩家有那么大的吸引力？

JK: 这是让你走出挫折的一种方式，这同样也是使你要暴打某人的幻想（你在电影中所看到的）变成现实的方法。因为真实的街头会在很短的时间内结束，而且并不好

看。它非常可怕，还有很多感情因素卷在里头。大量的打斗游戏所做的就是创造一个亚文化群，因为如果一个小孩能够通过某一组合和调整作出特殊的动作，他就会出乎意料地成为那些玩耍和关注这个游戏的孩子群体中受欢迎的“人物”。

TM: 对刚入门的游戏开发人员，你有什么告诫和建议吗？

JK: 找一份儿写得很好的设计文档，把其中涉及到的要素整理一个思路出来。尽可能多地学习关于游戏产品的其他方面的知识。最重要的是获得经验——深入到游戏产品中，看看它是不是就是你正想要做的。找工作时一定要坚持到底，并且要永不言不。为仿制游戏制作设计文档，添加那些你感觉会使你的游戏比现有游戏更好的内容，并让你的玩游戏的朋友诚实地评判它。

TM: 以你的经验，通常哪些基本的游戏设计规则应当遵守？

JK: 要了解目前有些什么东西，并以它们为基础开始构造。学习那些游戏大作的制作所用到规则，然后研究如何实实在在地（不是粗心地）打破这些规则。

TM: 游戏开发新手应该集中学习哪些技术和技能？打斗游戏设计人员呢？

JK: 对于打斗游戏设计人员，我觉得他们应当设计出一个拳击游戏，并引入其他一些能让游戏更加引人入胜的要素。许多的打斗游戏都非常的相似甚至完全一样。

TM: 作为一个游戏设计人员，还需要为技术以外的东西做准备（学习脚本和三维开发包），您认为对吗？

JK: 对。你所知道的每一样东西都会帮助你成为一个更加优秀的游戏设计人员。

TM: 下一代最好的游戏可能会出自哪里？

JK: 任何地方都有可能，只要能让开发组产生出一些与众不同的想法、除拳击赛之外、玩起来更加容易、有着始终如一的标准、可以使玩家沉迷（或者应当说上瘾）于游戏环境之中、并且总是支持游戏可玩性的组织。

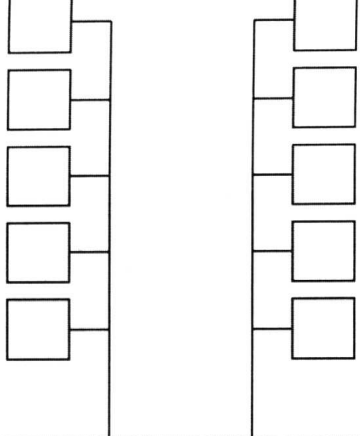
TM: 为了适应游戏设计的成长，游戏产业是不是需要作调整？

JK: 是的。像好莱坞，他们俗套的发展趋势以及只关心赚钱，而从不花时间研究和发展的思想。

TM: John，非常感谢你花费时间与我交谈！

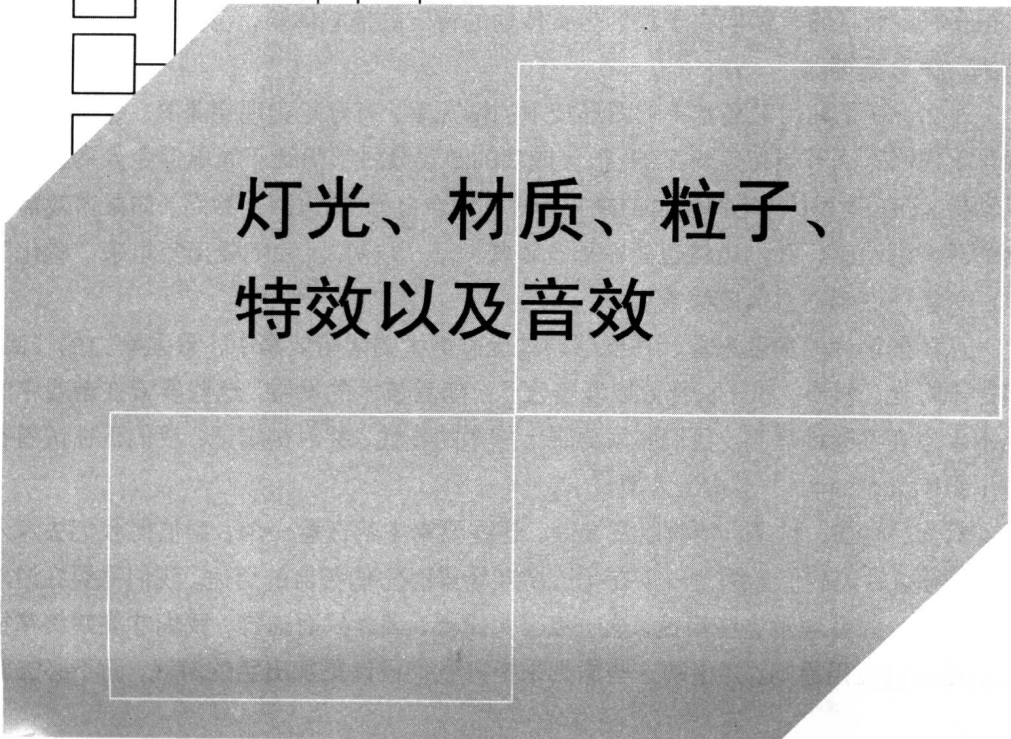
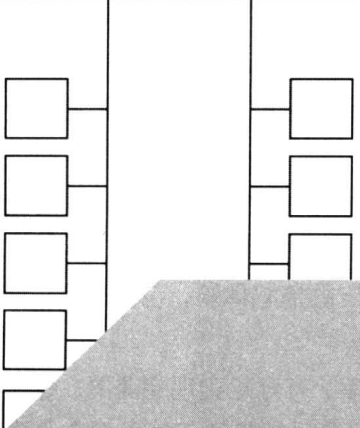
MEGA 提示

1. 在任何时候都要通过用模块化的思想设计关卡组件或者“关卡构造集”来使用并重用预制好的图形。如果你们在为自己制作工具集，一定要确保给这个思想留下施展空间。
2. 当你玩你所喜欢的游戏时做一个关卡日志。在你的日志中记下你所玩的游戏哪些东西好玩儿、哪些不好玩儿，以及为什么一个关卡或者游戏思想会成功或者失败。记住，设计过程是一个不断加工和改进的过程。
3. 在开始制作图形之前，为所有的主要构造问题统一商定好团队的总体方针（比如缩放比例、栅格尺寸，以及角色高度）。
4. 在投入大量的开发资源之前，在所有可能的地方进行游戏机制的原型设计。
5. 以一次完整的关卡测试结束关卡的占位设计阶段，标记出所有已知的问题或者争议。将这些问题和争议与适当的团队成员快速、及早地交换意见。
6. 制作过程中时刻注意游戏画面的整体效果。

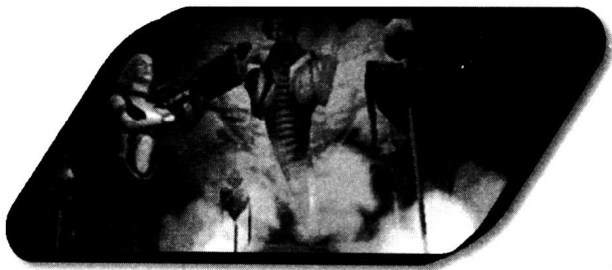


CHAPTER

3



灯光、材质、粒子、
特效以及音效



现如今，一个游戏要具有竞争力，其游戏环境必须有足够的视觉吸引力。我们希望通过玩家的眼球以及他们飞快动作的手指（或者甚至是对于节奏较慢的游戏时动作并不快的手指。在这些游戏中玩家根据线索和提示来进行角色构造、资源管理或者环境探索的）在玩家的情感和游戏之间架起一座桥梁。然而，一个能够从视觉上引人入胜的游戏环境要比简单的漂亮场景或者视觉享受意义丰富得多。漂亮的场景仅能提供……短暂的美感，但是最终还是不能满足玩家的需求。充分利用视觉效果可实现饶有趣味和意义丰富的游戏场景。在玩家开始注意到我们的游戏的时候，我们要牢牢地抓住他们的情感。我们希望游戏所能够提供的兴奋感和“连通性”上升到一个新的水平；其实现方式很多种，而且通常都取决于我们想要传输的特定的情感体验，包括恐慌、困惑、大难临头、惊奇等等。

那么，在不利用最高水平的图形技术的情况下，有可能实现完美的、令人满意的游戏可玩性吗？回答当然是肯定的。很多成功的游戏最后都归结于简单但令人陶醉的行为或动机。（在那个家伙吃掉你之前把它干掉，保全自己的性命！）然而，如果游戏需要丰富的环境沉浸感，我们当然也要在那一领域发动“攻势”。我们要用气氛来“感化”玩家。气氛往往牵动着情感线索。

从场景设计的角度来看，在能够调动玩家情绪的诸多要素中，最强有力的“武器”莫过于灯光、材质、粒子、特效以及音效了。随着技术的发展，这些要素在游戏中的应用水平也在不断地提高。我们必须跟上技术的步伐，并时刻牢记：我们正在试着在流沙（shifting sands）上构造大厦。

灯光、材质、粒子、特效以及音效，这些要素中的任意一个，都值得我们去深入地钻研。后边我们会探讨一下在构造游戏场景时它们各自的用途。我们需要在游戏的概念设计阶段就理解这些用途，这一点至关重要，因为只有这样，我们才能始终掌握着它们基本的应用原则，并借助这些原则来指导我们设计更加出色的游戏，而不必去管这

些技术如何是发展和变化的。这一章中，我们将要考察和探究这些要素中的每一块儿领地，我们要有明确的目标——深入地理解如何利用这些要素来营造气氛，从而支持游戏可玩性。



3.1 灯光

通常情况下，我们可以通过几个灯光参数来调整内部场景或者外部场景的灯光。如何准确地实现我们的灯光目标，还要取决于很多因素，比如说我们所用的游戏引擎软件工具的具体情况，或者是我们构造通用三维软件包的工具系统。本节我们将讨论几种类型的灯光控制方法。首先，还是先花点儿时间看看设计者们在着手设计游戏场景照明时所关心的高阶问题。

- **光源数目** 这一点通常会受到性能方面的约束，因此最好在开始时就确定能够满足光照目的物时所需的最少的光源数量。
- **光源类型** 光源类型的选择将取决于场景的细节以及你处理这些细节时所采用的方案。
- **光照强度** 光照强度需要进行频繁的调节，但是你应当对游戏场景中的环境细节进行光照处理（比如，室外的太阳光照，接近傍晚时的薄暮，办公室内的荧光灯等等）。
- **衰减效果（发散）** 这与光照的调节和改变有关，因此在早期阶段你就要考虑所有的关卡中在哪些地方应用衰减效果有助于实现光照目的。
- **颜色** 如果你有颜色概念设计板时，颜色就来源于此。如果没有的话，你就需要设置一个灯光的颜色模式作为出发点，然后再根据需要进行调整。

对于游戏的光照处理，我们还需要考虑的一些特殊因素：

- **位置变化时的光照细节** 不可避免地，你需要对场景的不同部分进行不同的光照处理。考虑一下当位置变化时需要场景中各部分所进行的光照处理，这样你就会知道当位置变化时对光照的要求将会发生哪些变化。不要仅仅是为了改变灯光而改变灯光，应该为了改变气氛而改变灯光。
- **突出点** 这些指的是道具、提示或者可见得分点的光照处理，即是你想要给玩家强调和着重部分。千万不要错过噢！
- **特殊情景下的光照动画** 考虑一下你是否需要一些特殊场景下的光照动画。举

例来说，是不是需要旋转的警报器发出的光线？闪电闪过的效果？探照灯扫过时的光照？这些都是特殊情景下的光照动画，它们能够用于场景中需要这些效果的区域。

- ▶ **使场景与将要出现的其他光照效果相匹配** 考虑一下你的游戏场景怎样才能和附加的光照效果相匹配。打个比方，如果你一开始就将整个场景设置得非常明亮，而有一个角色进入该场景后将引爆一枚巨型的核武器，该武器发出的光线将冲击到整个场景，它将掩盖掉你所有的光照效果。当该武器引爆时，它发出的所有光线将会叠加到已存在的光线中因而使得光照变得太亮，这时你就不得不把原来的光线变暗或者降低该武器的光照效果。

不管你是在构造一个什么样的游戏，一个滑雪场竞速游戏，其他类型的体育运动游戏或者中世纪的动作 / 打斗游戏，所需的光照特效和光照能力都根据场景的不同而有所差异。其中改变不大的是应用光照表达情感时的那种思想——这种情感就是你试图使玩家获得的情感，尽量利用光照效果来支持游戏可玩性的情感。

在滑雪类游戏中，赛道在晚上可能只需要极少量的光照（这在本章后面“光照强度和光源”一节中专门讲述）。事实上，“受限光照”可能是该过程中遇到的一个挑战。另一种情况可能是被明媚的阳光照耀得非常明亮，并且具有闪耀的光晕（迅速地观看明亮的光源所产生的“光圈”）和大量的边缘光（比如物体边缘的高亮度热光）。二者都能够帮助我们营造游戏氛围。

对于滑雪赛道竞速的游戏来说，其游戏可玩性最终归结为平滑的执行过程以及特技型的跃起、空翻、跳跃和翻越动作的实现，而且还要避免追赶过程中与其他物体（比如树木或者其他的滑雪运动员）相撞。这里“平滑的执行过程”通常是一个精巧的控制器的问题（利用程序调节游戏控制器对用户输入的响应方式——比如单击那个三角形按钮来完成一个滑冰 / 雪动作，比如跳跃）。游戏如何做出直接的响应完全由调整控制器来实现。

光照当然也有其实际的用途。你绝不会希望玩家在黑暗中碰碰撞撞，除非这就是你的目的！它只会导致游戏很快过时。当你为光照以及光照对情绪的影响“睁大眼睛”时，你就会得出几条规律。这需要针对不同的类型进行研究。

通过电影摄影技师那精确的视觉经验来研究光照是一个很好的想法。他们的日常工作中就涉及到光照及其构成要素。研究摄影技术来构造眼睛对光线的强度、对比度、饱和度、色调，以及重点照明的认识及感知能力，这也是很有帮助的。研究颜色理论，包

括颜色与情感的联系以及人对颜色的反应同样也是很有价值的。已经证实光线颜色本身就能改变人的生理反应，并因此而改变我们的情绪。对很多相关领域进行研究，这样才能成为一个更出色的游戏设计者，我喜欢将游戏设计者视为“专业的通才”。而且游戏开发者永远都是学生。

下面是一个概括性的示例。图 3-1 所示的是以最大亮度（“整体”最大光照强度）显示的一间小屋。很显然，该小屋内的光照单调无聊而且在情感上没有任何吸引力。实际上这种光照几乎没有任何的用处，除了一种实际应用——让我们可以清楚地看见周围的路。

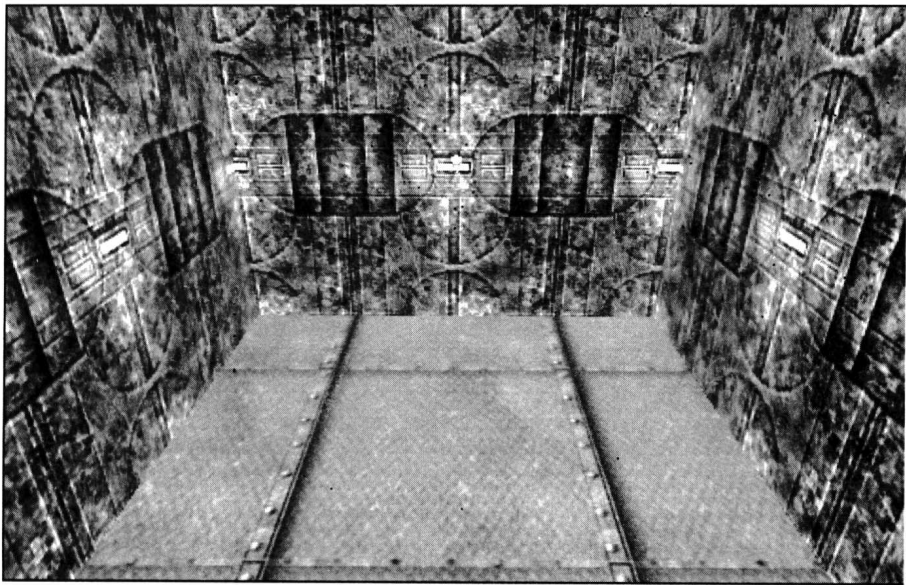


图 3-1 最大亮度的整体光照

作为工作流程上的一个步骤，在进行布局和基本构造阶段（在第 2 章中描述的）我实际上利用的就是“最大亮度”的光照。将上面这个简单的例子与图 3-2 相比较，图 3-2 给出的是一模一样的小屋，只是增加了一个单独的光源；它的亮度值已经发生了变化。这间小屋能够传递更多的情感，并且仅仅是传递情绪因素的开始。而所有我们能做的事情就是改变亮度的取值。这就打破了我们一开始使用一种单调的光线照射时的一律样式。除了亮度因素外，在设置中，我们还有大量的其他因素需要我们控制。

具有最先进技术的游戏引擎正开始表现出动态光照的特征。这为游戏中的光照处理开辟了全新的道路。很快光照就会产生更加快速和直接的效果。动态（或者实时）光照处理允许角色或者物体和光相互作用，因此就像在现实世界中一样，他们将会投下影子——这并不是预先计算光源并“假冒”物体的影子以此来暗指一个真正的光源。然而就像你能够想像的那样，动态光照会耗费大量的图形处理器时间。它需要坚实的图形硬件的支持，但是，它很快就会在诸多游戏中得到应用。

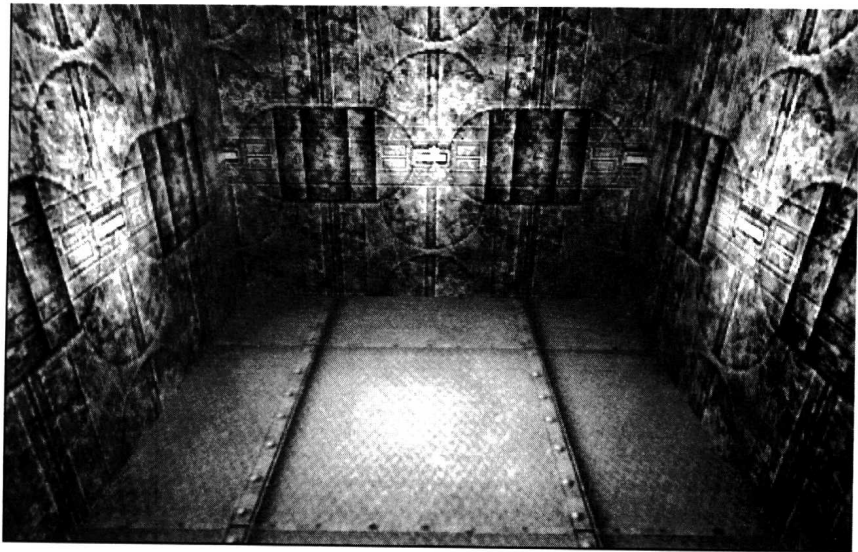


图 3-2 改变亮度后的光照效果

现在，我们来看一下为了提高游戏可玩性，如何通过调整光照来传递情感。在后面的几节中，我们将会看到各种各样能够控制光照的方法，所有这些方法都能够传递情感，以此来支持游戏可玩性。

■ 光源位置

场景中光源的布置或者定位必须同时满足几个目标：实用性（比如，为玩家照亮道路）、美观性（能够传达情感以支持游戏可玩性）和运行效率（在希望的光照效果和运行良好的光照效果之间作出平衡）。任何时候都尽量得到最佳的视觉效果而且运行的效果最好。另外，所用的光源数量满足目标需要即可，绝不多用，不要添加不必要的光照，

也不要应用对场景不起作用的东西，不要只是为了复杂而增加复杂的东西。在光照处理方面很容易走向愚蠢的一端，四处布置灯光看起来是很酷，然而，这就像是在圣诞树下的假日宴会一样到处都是灯光，你的游戏不会有多大的改善，而且在运行性能上也只有弊没有利。要随时准备好因为性能原因而去掉一些灯光。

一般情况下，你需要对光线进行定位以复现它们在现实世界中的位置或来源。如果你要再现一群乌合之众的头目办公室里的灯光，玩家可能希望出现某种类型的光照效果。也许你要模拟追踪时的光照效果，你绝不希望像飞机棚一样地照亮它，这再一次说明了具有充分的视觉参考素材的重要性。你同样希望能够以新的独特的方式对光线进行定位以获得你所希望的效果。对于室内光照处理，剪辑一些建筑杂志的灯光设置作为参考很有帮助。我有一个文件柜里装满了这些东西。

我发现一个很好的练习方法，它可以提高你的灯光技术到你开始实现你所希望的光照效果的水平。这个方法就是从一部科幻电影场景（*Cinefex* 杂志就是一个极好的资源）中摘取一幅场景照片，并尽量在你的工具软件中重建或者再现其中的光照效果。逐渐地，你就会对这种练习感到轻松自如了。灯光非常有趣，它是关于发现的全部艺术。对一个同样的场景，可以仅用侧面灯光照明，挑战你自己，接着只用高架灯光照明，然后仅用地面灯光照明，最后再把所有这些灯光混合并配合起来使用。完成这样的过程将有助于增进你的光照工具箱中的工具的功能。

■ 灯光颜色

场景中的灯光一般都会投射取决于 RGB 值（灯光颜色中的红色、绿色和蓝色含量）颜色的灯光。你可以利用这一点来产生巨大的情感效果。这就涉及一些颜色理论。如果你希望给场景施加紧张的气氛，或者展现一种阴冷的情感环境，又或者创造一个“冷酷”角色的藏身之处，那就在你的灯光中多加一些沉重的深蓝色。如果你想表现一个大家围坐在 12 世纪爱尔兰小屋的火炉边聊天的场面，那就多用一些橙色、紫色和黄色使灯光变暖。

正如前文提到的，人类对灯光色彩的情感反应其本身就是一个研究课题。从应用的目的出发，我们很容易理解为什么掌握人们对颜色的各种反应是一件很重要的事情，而接下来就是将这些知识应用到你的游戏场景中。这里有一些基本的规则，但是你应该考察一下它们的应用界限。你应当很自如地对这些想法加以实验，并尝试一些新的东西。有时你可能想利用一定数量的相同色。试着加入一些与情绪没有直接关系的光线颜色，

看看会发生什么。注意一下电影中摄影师是怎样利用颜色的，你可能会发现他们通过调整场景中灯光的颜色或色调来支持场景中的情感氛围；注意一下舞台灯光设计师在剧场中是怎样使用灯光颜色的。最重要的，对于怎样在环境场景中怎样利用灯光颜色来表达一定的气氛和情感，你要在工作中逐渐形成你自己的理解和认识。

■ 基本光照类型

一般来说，在大多数游戏的开发环境中，除了你所用来构造游戏的各种工具和技术以外，都有几种不同类型的光照供你使用。

点光源或环境光

点光源和环境光通常被用做“整体”灯光，这是由于它们从单个光源点发出射向各个方向的光线。你可以应用这种类型的光线对物理空间进行照明，用来模拟太阳光照，并且在很大的程度上能够为场景中的环境光或者“可见”光设置整体的光照感觉。应用点光源或者环境光的例子可以参考图 3-2。

聚光灯

就像在很多的真实情景下那样（比如在剧院的舞台上），聚光灯也用于为游戏场景中某些道具或角色提供特写和用强光突出——简而言之，就是以某种方式使对象或角色从场景中的其余部分或者背景中突出显示出来。你也可以使用聚光灯来暗示游戏可玩性。

在游戏场景中聚光灯可以意味着出口、秘密通道、线索和可毁坏的物品，或者只是简单地吸引玩家的眼睛，让他或她“不要错过这个东西”。也许你正在使用聚光灯来突出雕像的显示，比如在第 1 章的大教堂例子中被冻住的野兽雕像。当看到一束聚光灯照射到一个变了形的、呼着水汽的被冻住的野兽时，你会感觉到一种视觉上的享受，这有助于你了解所在关卡的诅咒背景故事。

聚光灯是以圆锥的形式显示灯光的，你可以控制很多显示特性以达到预期效果，调节最终灯光的半径，使聚光灯的边界模糊化等。图 3-3 是一个双光源照射下的场景，显示了在有顶部环境灯光照射的同时聚光灯作用下的道具情景。

定向光

定向光是一种柔和的填充光（fill lights），可用于显示细节。在场景中定向光能够对准或者指向某些道具或物品以更好地显示细节部位并提供高光显示。图 3-4 显示了

用定向光填充或者细节显示的效果。这也是一个双光源照射下的场景效果，一个是环境光，另一个是定向光。



图 3-3 会聚光的例子

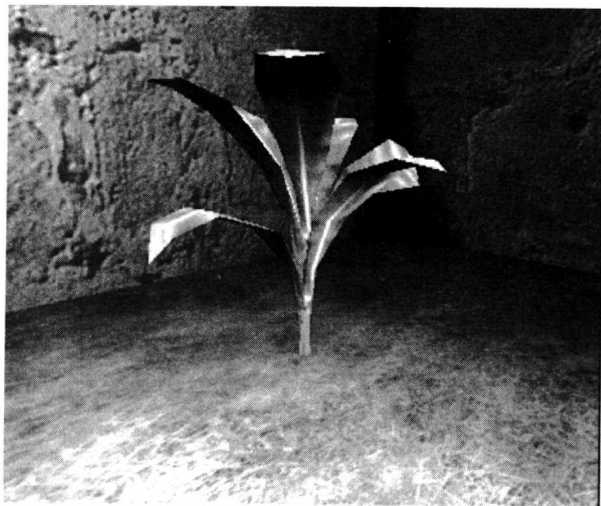


图 3-4 定向光的例子

■ 光照强度或倍增器

不论是什么类型的灯光，都可以通过调节光照的强度或饱和度以达到你所希望的光强效果。根据你所使用的工具软件的不同，控制光强的特定方法也有所不同，但最终效果是相同的。你可以在单个灯光或者整体灯光的基础上对场景中的灯光强度进行调节。降低光强通常会为你的场景添加戏剧性和悬念。

这里有一个小小的警告：不要过分地降低灯光强度。有些游戏将光强降得很低，以至于玩家在场景中导领角色都变得很困难。你不会希望仅仅是由于你自己喜欢极端微弱的灯光而牺牲“游戏可玩性”或者使玩家饱受挫折吧！（“我什么都看不见！”）这种类型的光照可能会在舞台场景中有用：此时舞台上需要强调的是角色的声音，而被动的观众则被要求将注意力集中在对话交互上。在主动的游戏环境中它并不起作用。

为了表达某种情感并实现光照目的，你并非一定要让玩家始终在黑暗中磕磕碰碰。如果你不会在关卡的行动、区域、目标和剧情的基础上利用光的对比度和变化（由亮到暗）来营造氛围，那么你是无论如何也不可能把对光照的使用发挥到极致的。

那么，怎样完美地平衡你的关卡光照呢？也许你希望某些部分十分黑暗，仅有柔和的光亮，而另一些部分则要有更加生动的光照效果，这一点通常需要在局部的环境光与点光源之间寻找一个恰当的平衡点来加以实现。我发现，通过将整体的环境光设置为比它看起来最合适的状况再稍微暗一些，然后回过头使用点光源和聚光灯来构造灯光，我就可以实现戏剧般的光照，从而获得和谐一致的效果。

■ 光的衰减

为了取得更好的视觉效果，我们通常都想要模拟光线离开光源时的消散效果。我们都知道光线离开光源以后会减弱，这通常称作衰减。控制光线的衰减就是简单地控制光线离开光源（场景中的灯光）后以多快或者多慢的速度消散。如果你要求快速衰减，就不要让光线远离光源，如果你希望光线从光源处非常慢地消散，就应该调节光强衰减范围的取值。衰减通常用来模拟由地面沿着墙壁射向空中的光线，也可以模拟路灯光。

■ 三个光照示例的设置

下面我们看一下怎样才能实现下面三个例子中的光照设置：阴森的城堡，欢乐的森林和恐惧的森林。

首先,考虑一下怎样对一座阴森的城堡进行光照处理。刚开始对这座阴森的城堡中一个房间进行光照处理时,我用了一个全局光或者说环境光和两个聚光灯。用深紫色(RGB: 128, 128, 255)加重环境光,使房间看起来很阴冷。针对角落里的两个聚光灯,让它们指向相反的方向:一个朝上,另一个朝下。如果天花板足够高的话,你就可以暗示光线是从场景中的好几个光源发出来的。图 3-5 给出了这些设置产生的结果。你可以对这个简单的例子进行试验,对城堡中所有其他的房间进行光照处理。



图 3-5 阴森的城堡

在开始对欢乐森林中的某些部分进行光照处理时,应当包含一个或两个已调节好距离的明亮的环境光(换句话说,在你需要照亮的区域内,对这些全局灯光的距离和高度进行试验调节)。利用大量的橙色和紫色对这些环境光的颜色进行平衡处理。我们想要的是被我叫做“麦片粥”的阳光——温暖、神奇并且充满生气。作为一个视觉参考点,我们可以回忆一下《魔戒》——*The Lord of the Rings*三部曲之一的影片《魔戒现身》——*The Fellowship of the Ring*中 Gandalf 坐在马车上到达夏尔时的情景。场景中神奇的阳光和色彩处理确实有助于传递一种奇妙和令人神往的思想感情。接下来我利用两束

很强的聚光灯来高亮度显示地形特征，比如一片充满生机的花圃，一棵会说话的大树，或者一只欢乐的动物的巢穴，再用一个定向光或者隧道光来散开太阳光柱中的一部分光。你还可以在光线通过的地方增加一些尘埃粒子，这看起来非常美妙。在本章后面的“粒子”一节中你将会看到怎样来实现尘埃粒子效果。

你会怎样处理一片充满恐惧的森林的光照呢？同样的道理，对于恐怖的森林中的某一部分（我通常都是一部分一部分地进行光照处理的），用一个“阻塞的”或者最小限度的环境光进行光照处理。环境光的颜色在蓝色和橙色之间进行调节。因为我仅用了一个全局灯光，接着我还要使用三束聚光灯。我倾向于使用红色和蓝色的灯光，并把这些光源放在粗糙的树根上或者这片恐怖森林中的其他奇异的道具上。这些放在粗糙的树根部（或者其他道具上面）的聚光灯投射向空中，会产生出极端的、使人感到害怕的阴影来。恐惧就存在于这些阴影，不是吗？



3.2 动态灯光

依赖于你的软件工具集，你可以通过对环境灯光进行动画处理，从而得到几种不同效果。比如，在一个12世纪的旅馆里，当主角们围着火炉聚在一起讲述背景故事或促成某个使命时，你可能想模拟一下局部发出的火光的情景。要做到这一点，你可以动态地改变点光源或者环境光的强度来近似模拟火光闪动时的情景，可以简单地为光的强度值设置一个动态变化的周期或者循环（换句话说，就是帧到帧的取值循环），该循环将不停地重复直到你认为应该停止的时候才结束。

动态灯光的另一个显而易见的用途是制作闪电闪过时的情景。简单地利用高亮度的“脉冲”值设置一个循环，然后将它进行随机重放就可以了。也许它只是每隔30~40秒重放一次，这里包含一个随机值，由这个取值就决定是否出现闪电。

在另一个例子中，你可以让灯光以接近于“光的速度”跳跃动作，这就会创造出光线的拖尾或者“闪光灯”的效果。动态灯光还有很多可能的情景。你可以用某种形式的动态灯光来模拟电流摇串着进入一根古老的矿井钻轴的情景，钻轴上的电光啪啪地响并且不停地闪耀着。你甚至还可以在玩家进入到场景的时候通过设置熄灭一些灯光。可以考虑一下怎样才能将一种技术应用到未来的游戏中以取得某种预期的效果，这永远都是一项有益的练习，这是积累你的思想知识库的一部分。



3.3 材质

材质（或称为“皮肤”或者“表面贴图”）就像是香皂包装盒上贴的标签一样。它们是包裹在物体表面上的图形。我们可以通过使用一张生了锈的盒子表面的材质贴图，来使一只盒子看起来像是真的生了锈。鉴于一个物体的位置以 X 坐标和 Y 坐标位置表示的，物体表面材质的“位置”可以用 U 坐标和 V 坐标位置来表示。这仅是一些数学上的术语，用 U 和 V 坐标代替了 X 和 Y 坐标。

构造材质本身就是一件极具艺术性的工作，资深的材质专家们在一个开发团队中简直就是“无价之宝”。如果我们的游戏中没有使用材质贴图，那我们将只能看到一些平直的、阴影的表面，这并不能提起人们的兴趣，不是吗？平直的、阴影的表面会让人感觉没有经历过风吹日晒或锤炼，也不能为玩家提供任何视觉暗示和线索，而这些都可以通过使用材质贴图来完成。更有趣的是，通过使用一个凹凸贴图，我们可以暗示某个表面具有凸起和凹陷，给人一种崎岖不平、带有波纹或者被侵蚀了的感觉。凹凸贴图改变了物体的表面，比如说使其具有了山丘和溪流。通过这种方法我们就可以使这个表面具有某种立体感。

通常情况下材质都是制作好了的，这样你就可以把它们铺在物体的表面上。铺的过程就好像是将一个瓷砖图案铺到地面上一样。我们想重复使用这些贴图，但是又不希望看到它们连接处的接缝。比如，我们正在“铺”有积雪的地面时，我们不希望看到区块之间相互拼接后的区块边缘线。

这些能够“铺”的材质，可以像七巧板一样沿着各自的边界拼装在一块儿。图 3-6 给出了一个“可铺的”材质例子。这种类型的材质可用于地板或者墙壁的表面。

其他类型的材质大都具有特定的用途，不能用来填充像地板和墙壁这样大片开阔的空间。它们更多地用于表现物理环境中的一些细节，如下水道的出口、带有污点的玻璃窗、或者管道系统。图 3-7 给出了一幅具有特定用途材质的例子。

材质可以制作为不同的像素尺寸、不同的分辨率和不同的色彩深度，这要根据你所使用的缩放系统的不同而不同。所有这些东西都是由你的游戏引擎的要求所决定。为了处理表面或者材质信息（有些时候甚至要以一种全新的不断变化的方式来处理），每一个游戏的引擎都要定制编写。然而，十分普遍的情况是，大多数引擎只处理特定尺寸及特定分辨率的材质。



图 3-6 可铺材质

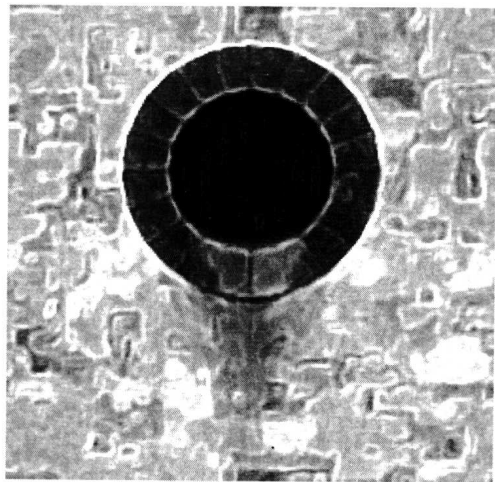


图 3-7 具有特定用途的材质

举例来说，用于像墙壁和地板这样大型表面空地的可重复使用材质，可以制作成 128×128 （128 个像素的宽度乘以 128 个像素的高度）大小。某些“细节性”的材质，如门上的装饰图案，或者特别适用于门框周围的材质，可以制作成 64×64 像素大小。确定所用材质的合适尺寸通常与预期的用途有关，当构造场景时，应慎重考虑在该场景

中如何使用材质。

有时制作能够很好地铺到一起的材质(不会出现任何的接缝和边界)确实需要一些技巧。如果你正在基于某些复杂的或者互锁的模式(考虑一下熔岩流或者玛雅神庙里的碑铭)来制作贴图,那么利用材质美工的技巧就十分重要。更为通常的情况是,在产品的制作过程中,先将一些占位用的或者临时的材质放置到场景中,等到材质美工提供出最后定稿的材质方案时才把它们替换掉。对于整体制作者或者关卡设计者来说,至少要能制作出这些占位用的材质或者他们自己的“基本的”材质,并且还要能够对它们进行基本的修改,这一点是很重要的。你学到的有关材质的知识越多,情况就越好!

当一个材质出现问题时(或者不能正确地铺在一起),你会很快地意识到这一点,你马上就会看到某些不希望的重复模式出现,这就意味着所用材质的边界处不能无缝拼接在一起。

■ 恰当地使用材质

制作自己的资源细目分类时(回忆一下在第1章大教堂例子中所述的精简版本),其部分工作就包括确定游戏中某个场景或某一部分会需要什么样的材质。从概念框架上说,你需要和材质美工密切配合以便精选和制作出一套材质集,然后你就可以用它来细化场景或者关卡。在制作材质集时你还要有相当程度的深谋远虑。你希望能以正确的方式制作出恰当的材质图片来实现自己的视觉目标,举例来说,128×128的材质用在地板和墙壁上可能效果会很好,但是如果你打算以另外的方式使用这些材质,你可能会遇到麻烦,你需要改变它们的尺寸、甚至根据需要定制。

如前所述,我们将材质放置到某个表面上,并且使用 U, V 坐标系统在网格或者几何体表面上定位贴图。但是,当你将材质“印”到表面上时,它们可能不能正确地对齐。你就会沿着表面“移动”或“滑动”它们,直到最终正确地对齐为止。大多数的游戏引擎工具都能够通过改变选定材质的 U, V 值来实现这个过程,这是将材质按照你所希望的形式对齐的一个好方法。

在有些情况下,你可能希望真正改变某个材质的绘图比例或尺寸,以使其能够更好地与场景相匹配(换句话说,就是相对于场景中其他的东西能保持一个合适的比例)。你可以通过改变 U, V 绘图值做到这一点,该取值将会根据你所做出的改变而放大或者缩小材质在那些方向上的尺寸。这样做的时候一定要小心,因为你可能会得到一些你不希望出现的效果。在所做的改变看起来有明显作用之前,你所能做的也仅限于对材质进

行放大或者缩小。这会把材质变得模糊不清或者压缩到没法使用的程度。这就是为什么在制作材质时，你就应该知道你要怎样使用它们。



3.4 粒子

在那些为游戏（和电影）制作的最有趣、最复杂而且最占用处理器时间的视觉效果中，有一些使用的就是粒子系统或者动态系统。当你想到粒子系统时，考虑一下那些在自然界中一直在发生的行为，例如当一大群鸟儿飞到一起时，这些大量的个体会形成一个更大的整体，或者说来自一片乌云的局部降雨淋到一个稻草人上。在一大群鸟飞到一起时每一只鸟就成为了系统中的一个粒子，同样在模拟来自一片乌云的降雨时，每一个雨滴也将变成该系统中的一个粒子。

以下是你可以利用粒子来实现的其他效果：

- ▶ 火把上面的火苗；
- ▶ 车辆排气管排出的尾气；
- ▶ 一群狂怒的异型生物攻击时的编队；
- ▶ 成群或者成串的物体；
- ▶ 运动动力学（滑冰者冰刀上的雪片，棒球棒上面脱落的粉末，滑行时飞扬的尘土，击高尔夫球时击起的草皮碎片，等等）。

在游戏中构造这些效果时开发者有很多种选择。构造火光时你可以使用精灵动画，就像传统动画那样的令人厌烦的单帧动画，或者利用粒子系统来对火光的舞动进行建模。为了保证游戏的运行速度，你必须在很多地方作出权衡。当然了，现在你也不能对所有的东西都精心地制作一个粒子系统，这只是因为它们的计算过程太复杂了。不管什么地方，只要可以利用粒子系统，而且其输出结果能够进行优化，它们就能表现出绚丽的视觉效果。

■ 粒子发射体的应用

你也可以像灯光和参与者（一个角色或者道具）一样在某个场景上放置发射体。发射体能够发射出粒子。（作为发射体的一个例子，可以考虑一下用在草坪上面的喷水装置。）你可以定义某个粒子系统将会发射出多少个粒子。通常情况下，在开始的时候最好只发射出少量的粒子，这是因为发射体都会占用大量处理器的执行时间。

假设我们开始的时候发射 50~100 个粒子，这些粒子可以从一点或者一个表面上发射出来。下面这个例子中，我选了一个全方向的点发射体。图 3-8 是一个点发射体的示例，它位于圆柱体的顶端（举例来说，比如像一个烟囱）。

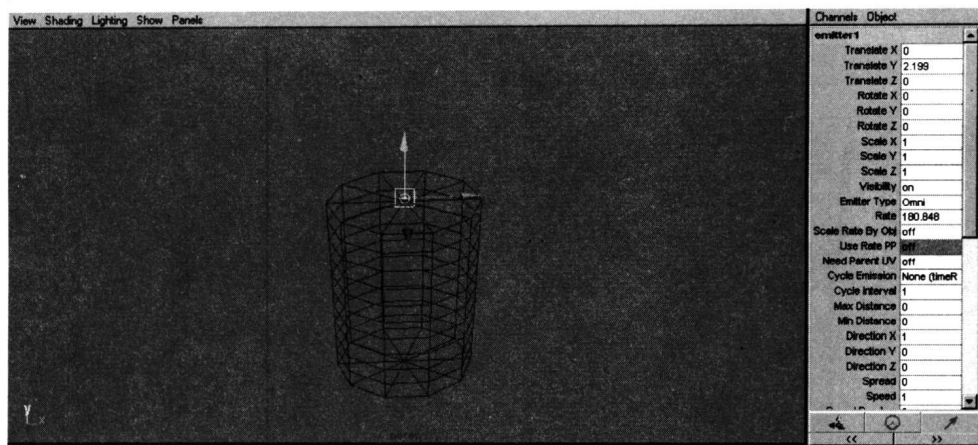


图 3-8 全方向的点发射体

在图 3-9 中你可以看到粒子的发射结果，粒子从一点发出射向四面八方。我将所有的粒子都渲染或者画成圆圈，这样你就能够更容易地看到这些粒子。在使你的粒子行为达到你所希望的效果之前，这只是一个开始成形的基本开端。

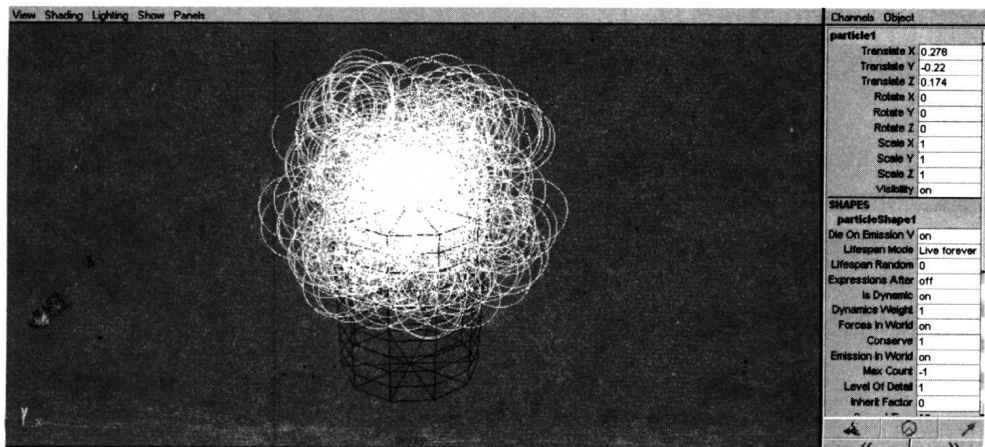


图 3-9 全方向发射的粒子

粒子的行为会受到作用于那些粒子上的力的影响，这些力称为场。你可以以上百种的组合方式来利用这些场，这样就能构造出很多有趣的视觉效果。其中重力场就是可以使用的一种力场。你可以为你的力场构造出各种吸引因子和排斥因子。

在图 3-10 中的动画序列中，我只是简单地增加了一个干扰场，这样当粒子从源点处翻腾而出时就开始向着一种类似于烟或者汽的方式发展。

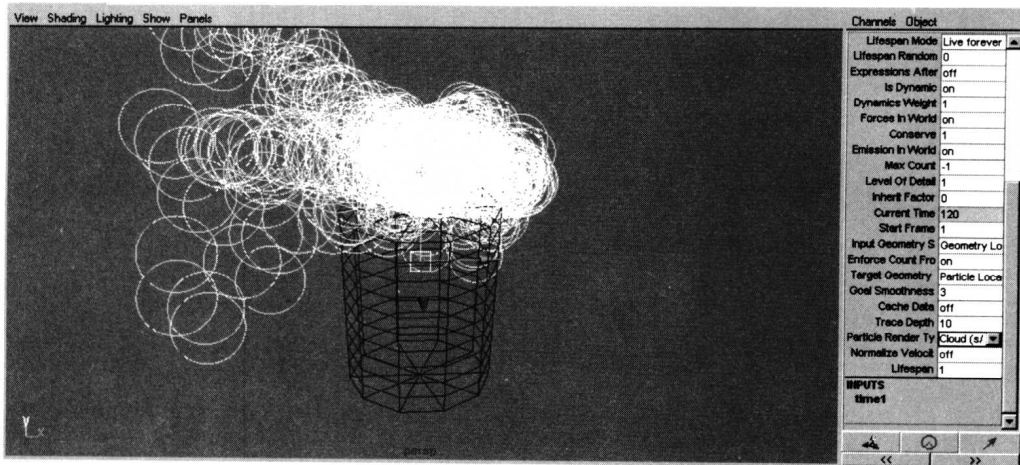


图 3-10 增加干扰场后的效果

在你知道了力场是怎样影响粒子之后，你就能够构造更为复杂的粒子行为了。也许你想创作出成群的蜜蜂在飞舞，或者想构造滑冰时冰屑从冰刀下飞出并远离冰刀的情景。你可能会构造的情景永无止境，每一种都具有某种类型的视觉冲击。这些都是很酷的素材。

现在我们来看一下一些游戏中的现实检验。一个特定的游戏引擎或者引擎工具包是怎样处理粒子系统的数据和粒子资源的呢？这完全取决于所使用的游戏引擎或工具包。每一个都具有其特定的数据传递方面的问题或者是资源导入及导出时的问题。

■ 粒子和动态系统信息的导出

在处理和使用粒子系统和动态系统时，游戏开发者需要作出几个方面的决定。一些常用的途经如下：

- 将完整的粒子系统和操作工具集成到你自己的编辑器中。

- 从玛雅或者 3ds max 中导入粒子系统的数据，然后在游戏引擎中重新构造这些数据。

对于第一种途经，其好处是能够共同分担某些粒子效果的构造工作，但是也就需要使用者学习这个特定的工具。

正如你可以想像到的那样，当前的很多游戏，其特点就是使用了各种具有粒子效果的元素：施加魔法、魔法发出光芒以及诅咒、神秘的花园中喷出的泉水样子、汽车排放的尾气、武器发射时的爆炸，等等。所有这些效果都可以利用粒子系统进行处理。所有这些单独的元素都可以作为一种专门为你的游戏所构造的资源。

很多的游戏都需要制作成百个（接近上千种）这种类型的元素。如果你有一种可靠的工具，它能够以很快的速度构造出这些可在游戏中直接使用元素并对它们进行改进，或者在构造这些元素所使用的工具本身能够在游戏引擎内部工作的话，你就可以避免任何在导入及导出过程中出现的麻烦。而且，在这种情况下，美工和设计者必须能够快速学会怎样来使用这些自己定制的工具。这并不总是一件很容易或者很有效的东西。

然而，很多的开发者并没有构造自己定制工具来构造粒子效果。他们依靠的是内置在像玛雅和 3ds max 这样的三维软件包中强大的粒子系统功能。这些粒子系统都制作得很棒，而且并经过了严格的测试。但是你在游戏中应该怎样利用在这些软件包构造的粒子效果呢？开发团队对此必须要作出一系列的选择。

假设一个 12 世纪的旅馆，里边有一个壁炉，你想为该壁炉中的火苗构造粒子效果。一种方案可能是从三维软件包中动态地导出这些粒子信息。这将会产生一个数据文件，文件中包含有在给定的帧数中（比如，从第 1 帧到第 30 帧）每一个粒子的位置或行为。使用这种方法，游戏引擎就能够读入这些数据并且重新构造出你在三维软件包中所制作出来的粒子效果。

另一种方案就是将粒子的行为渲染成动画序列，然后将这个动画粘贴到一个多边形的表面上。这种做法会给计算机带来一个很大的麻烦。我遇到过很多失败了的情况。当你使用计算机进行粒子系统的渲染时，一定要使粒子的数目少一些。开始进行实验时，要保留合理数目的粒子（要少于 100 个）并使保留的动画帧数也易于处理（大约 30 帧左右）。

一旦你能够使这些粒子的效果像火焰一样，并且颜色也像火焰，你就为该火焰构造了一个渲染序列。接下来，你可以构造一个简单的多边形平面并且把火焰的渲染序列粘贴到该多边形平面上。最后，将这个带有火焰的表面附着在（或者关联到）壁炉中燃烧

的干柴上，并把这个表面移动到适当的位置。现在，你就为这间小屋制作出了一个基于粒子的火焰效果！

这种思想可以应用到另一个示例中：如果你正在制作汽车排放的尾气效果，你可以使用与示例中相同的过程，然后尾气效果的平面贴到汽车的排气管上。

理想情况下，如果拥有一个允许你在场景中容易地放置发射体、然后可以实时调节粒子行为的工具包，这就太好了。图 3-11 和图 3-12 给出的场景是用 WildTangent Studio (WTStudio) 工具制作的。该工具能够让用户构造发射体并能够实时调节粒子的行为。

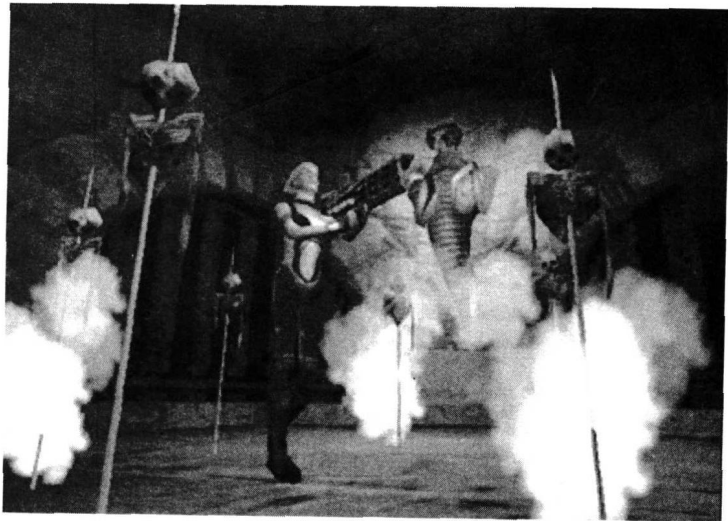


图 3-11 应用发射体制作的蒸汽效果

很多类型的粒子导向的效果（如闪电、下雪、下雨、瀑布、岩浆流和瘟疫云雾）都可以使用粒子系统来制作。为了得到这些效果合适的外观，需要涉及使用力场对粒子的行为的调节，并且还要对每一个粒子加上颜色和材质（或者是精灵贴图）。

到现在我们应该清楚地认识到，在游戏中构造粒子效果并对粒子进行处理时，我们有很多的方案和操作方法可以选择。比如说，目前的 Unreal 引擎就具有一个强大的内置系统，它能够应用粒子和发射体产生出这些效果。应用这种方法构造出来的效果能够为你的游戏环境增加视觉上的乐趣，有助于表达出故事的要点，而且还能够传递大量的情感信息。



图 3-12 应用发射体制作的火焰效果



3.5 效果

到这里你已经对构造粒子效果有所理解，我们就大致地看一下它们的应用。就像我们已经看到的那样，在实现很多的视觉效果时，粒子系统效果是一个很重要的方法。然而，它们很容易破坏系统的运行性能，因此使用粒子效果时，必须要进行规划、改进，并且通常情况下还要进行优化处理。

有些时候，某个效果可以使用一个简单的精灵效果来代替。一般来说，精灵 (*sprite*) 就是一幅的二维图形资源，它能够进行移动、缩放或者旋转等操作。有些时候，我们需要在精灵效果和粒子效果之间作出一个权衡。如果使用精灵效果就能够得到令人满意的效果，那么就使用精灵效果来代替类似的粒子效果，这样你就能够节省一些处理时间。如果使用精灵的效果看起来不太好的时候，你可能被迫使用粒子来制作这些效果，或者干脆就不制作这种效果。

甚至还有一些类型的发射体，它们发射的不是粒子而是精灵。这些发射出的精灵行为和粒子相类似。相比而言，这些精灵不会占用大量的处理器处理时间，而且还可以创造出一些有趣的效果，比如施魔咒。

图 3-13 给出的就是应用精灵粒子构造的魔法效果，精灵粒子从附着在雄蜂身上的发射体中发出。从位图的轮廓上面你可以清楚地看出来它们是以动画序列的形式发射出来的精灵。

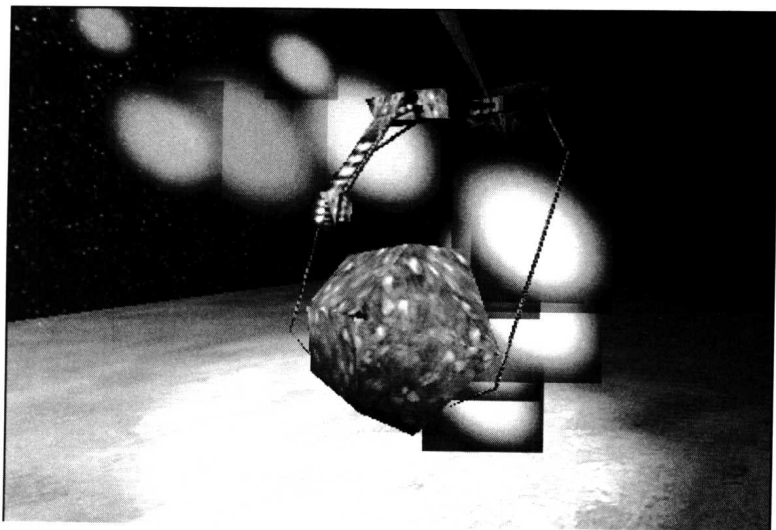


图 3-13 产生特效的精灵发射体

在你构造游戏环境的时候，可以通过文件引用的方式将各种道具引入到游戏场景中。也许你所使用道具之一就是武士的头颅和一个贴有所属派别的标志的长矛武器。这些道具可能分散在各处并戳在地面上，用以提醒玩家敌人就在附近。

你希望所有的道具都要符合正在构造游戏的视觉风格。通过使用精心制作的、边数较少的多边形模型并充分利用材质，你就可以做到这一点。将粒子效果应用到道具上就能够使之与众不同，以此来表达某些含义。还有，它会看起来更酷的！也许你想要构造一种效果，使贴在长矛道具中央的徽章上发出险恶的光亮来。处理的要点是你可以通过在道具上高效地使用粒子效果来增强视觉细节。并且试着将它们用作指示标志。要通过道具上增加的视觉效果细节告诉玩家一些信息。不要仅是为了出现某种效果而使用这种效果。在使用各种效果时你必须保持节约，这样你就不会减弱游戏的运行性能。

你必须要学会掌握对道具使用这些效果的频率及使用方式。归根到底，这又是一个涉及平衡取舍的问题。如果你在所有的道具中都使用同样的效果，这将很快失去对玩家本应该有的视觉冲击。你必须学会使用道具的效果来取得突出或者强调的效果，而并不

是游戏中的每一个道具都值得着重显示的。通过使用特殊的品质来显示一些道具，使它们与其他道具有所区别，这将有助于保持游戏环境的丰富多彩，而且会很明显地减少视觉上的厌倦感。



3.6 音效处理

在游戏开发中人们总是忽略了音效，而直到最后一刻才会想起来，这几乎成了一项惯例。音效有可能会对游戏作出重要的贡献，而由于几乎没有现成的资源可以供给音效开发者使用，这便削弱了这种可能性。长期以来，音效在游戏中的重要性在很大程度上都没有得到正确的评价，而且这种情况延续得太久了。然而，令人感到欣慰的是，对音效的处理正在朝着好的方向发展。游戏中的音效已经进入了一个新的时代。如果游戏发布商和开发商希望能够推出具有竞争力的游戏娱乐内容的话，就不得不继续对音效处理进行改进。对于我来说，对音效进行彻底而完全的支持出现得越快就越好。音效处理并不仅仅是作为一个“附件”在项目结束时加到游戏的后面；要发布令人感到兴奋的游戏内容，这是很关键的一个方面。

我们在前面所提到的一些设计工具就允许你将音效事件添加到模型中，或者在一幅场景中播放声音。有一些做得更好的是，允许你调节音效事件的“范围”，即在这个范围内主角可以察觉到声音。这种设置就决定了游戏主角（或者玩家的化身）在多远的地方能够“侦听到”一个给定的音效事件。

游戏中的音效有时用作线索元素。我已经发现，当音效和其他类型的视觉线索（例如血迹）一起使用时，其效果最好。比如说，当你听到了瀑布的水声，你会知道你已经离瀑布不远了，这将有助于玩家在地图上定位。

在传达情感方面，音效几乎和视觉效果是一样的，它也是你实现游戏可玩性最好的伙伴。下面列出了一些类型的音效，你可以使用它们来增加你的游戏可玩性。

- 能够刻画出电影艺术（电影片断）、音乐和关卡过渡的音效；
- 氛围或者环境音效；
- 特殊效果的音效；
- 突出游戏中激动人心的时刻的音乐；
- 角色的对话；
- 界面音效。

游戏中的音效通常是由游戏开发小组内部的处理音效的员工完成的,也可以由独立的承包商提供游戏中的音效。像在第2章中描述的那样,在游戏设计文档中,关于关卡或者使命的描述应当指明你对音效的要求和意图。开发团队需要向自己提出如下这个问题:我们打算怎样使用音效来支持游戏可玩性?对该问题的答案有的很明显,而有的则不十分明显。

一般情况下,游戏中音效的使用方法有下面两种,一种是伴随着场景,音乐直接从光盘或者硬盘“读入”或者播放的,另一种是将音乐效果或音乐片断加载到随机存储器硬件中。游戏控制台和个人计算机的声卡具有专门进行音效处理的处理器,这就可以使游戏开发者比以前能完成更多的工作。专用的音效处理器和随机存储器有助于为音效处理开辟更为广阔的发展空间。

通常情况下,那些必须快速响应甚至实时发生的音效事件,比如脚踢、拳击、射击和换挡,都是加载到音效随机存储器中,以便在游戏控制器或者个人计算机键盘上的按钮和按键输入时快速地进行响应。而场景或者游戏进行中的伴奏音效可能只是简单地从光盘或者硬盘上读取并播放。

从设计的观点出发,你应该注意一下你所喜爱的游戏中是在什么时候、怎样使用音效的。清脆响亮的射击声和沉闷的光束武器的轰鸣声听起来明显地能够增加游戏可玩性,但是也不要忘了周围的氛围!仅是单个角色的脚步声并不是什么氛围。某些“更为重要的”音效暗示总是应当设置音调并穿透背景音乐。可能是某个东西正在呼吸,也可能你听到了远处的敲钟声。反正只有脚步声是不够的。周围环境的音乐应当能够把你包围起来,并且给你一种立体的、沉浸的感觉。作为一种目标或者是理想,我们在任何时候都应该致力于多层次的音效的开发。就我个人来说,我一直都在努力寻找一些新的方法,通过使用音效来使玩家处于一种紧张状态,或者是让玩家体验一种全新的音效享受。

当我正在开发的棒球游戏的时候,我认为将球场上运动员的音效局部化将会更加有趣。换句话说,我希望根据球场上的位置来模拟棒球游戏的音效。在棒球大联盟游戏中,在玩的时候,你充当接球手和充当中场队员听到的音效是不同的。该游戏允许玩家选择扮演球场上特定位置的角色(比如,接球手或者内场手),并能为棒球队伍中的任意位置“增加角色特征”。我希望音效能够反映出这种思想。

游戏音效领域内有着充足的发展空间。目前的音效还不是真正的立体声效果,其中的部分原因是因为开发者们的主观臆断:他们认为玩家的音响系统好不到哪里去(一些顽固分子除外)。

如果你知道要在背景音乐或者音乐配音中播放几种武器的声音,你就应该在声谱中包含有所有的音频,从较低的低音到较高的高音。由于各种各样的原因,通常情况下是不可能包含所有音频的。

在音效的实现过程中,设计规划也起着极为关键的作用。现在,开发商们制作了很多授权游戏(*licensed games*)。授权游戏就是那些像《史酷比》——*Scooby Doo*和《星际大战》——*Star Wars* 这样的以著名角色和著名的领域为基础的游戏。如果你打算使用演员的声音来录制角色的对话,游戏中角色的对话最好是提前准备好并测试通过。你可能仅有一次机会来获得所需要的对话,而且你可能也只有三十分钟的时间来完成它。

■ 推动游戏音效的进一步发展

通常情况下,游戏中是如何使用音效呢?关于这一点大概我们都有自己非常好的想法。新的标准和新的技术正在推动着游戏音效更进一步地发展。事实上,当这些标准和新技术出现在大量的游戏玩家的家中后,游戏发布商才开始支持这些特性和功能。现在,有些游戏音效是扩展的立体声和环绕声配置的道尔贝编码(*Dolby-encoded*),但是有这些配置的发烧友在总的游戏玩家中只占很小的一部分。因此,游戏发布商就很难断定这种类型的支持工作是否必要、或者是否在市场上有竞争力。

游戏音效中还有很多可能的巨大商业空间没有得到开发。在场景中加入蟋蟀的叫声是一回事儿,但是在玩家头顶上跟随导弹飞行的声音,为玩家产生实际位置的音效反馈,这是一种令人兴奋的做法,而且在游戏音效开发中几乎完全没有人碰过这种做法。

多普勒频移(*Doppler shift*)指的是当声源移向观察者时声波频率变高,而在远离观察者时变低(可以考虑一下救护车从你身边呼啸而过的情景)。你可以在游戏场景中模拟出多普勒频移的音效。然而,公正地说,构造工具也只是使这种处理开始变得更为复杂。SoundMAX(www.audioforgames.com)正在试图改变这种状况。还有,很多最为有趣的声音现象都要求玩家使用耳机,而很多的玩家在玩游戏时就是不戴耳机。

很多游戏环境设计者对和位置有关的音效感应用到游戏中的可能性比较感兴趣,并且正在进行这种类型的研究。然而,在游戏中以全新的方法实现创新式的音效处理和音效应用又是一个牵涉到基本的软件工具问题。按照惯例,游戏程序员在写代码和扩展最新式的游戏引擎与工具包方面都有大量的工作要做。只是在相当近的一段时间里,各种工具的功能才变得强大起来,使那些仅是集中精力在游戏内容方面的开发者能够更加深入地进行试验。



3.7 设计工具集的变换

为游戏开发光照、材质映射、粒子效果和音效并不仅仅是意味着在商业化的三维软件包中构造这些特定的资源。同样，这也不仅仅是将三维软件包中的资源导出到游戏引擎中。其中所涉及的问题是：一旦这些资源经由游戏引擎的处理在你的游戏场景开始运作，你对这些资源有多大的控制权。设计者们总是受到工具集的束缚。

在这一点上，大多数游戏开发者都面临的一个生死攸关的设计、布局和执行方面的挑战，其中包括被我称为“工具集的变换 (*tools shift*)”的挑战，认识到这一点是很重要的。这个术语仅是指目前游戏开发中的一种工作现状：你没有必要知道在某个平台上开发某个游戏时将要使用什么样的工具集，而在另一个不同的平台上开发其他游戏时你又将会使用什么样的工具集。当你完成某个游戏，转而开发另一个游戏时，你可能需要很快地掌握另一套工具集，这才有助于你开发当前的游戏。甚至在你对当前项目进行收尾时你可能就需要为下一个项目的进行而学习并开发必要的工具集了。

对于大多数的独立开发者来说，制作一套定制的工具包是一项十分昂贵的做法。工具包制作、维护以及文档书写，都是非常昂贵的。也许这套工具仅适用在某一特定类型的游戏创作，但是你可能会转而开发另一个游戏，而这个游戏也许能够或者不能够利用你已经构造好的工具框架。唉，这种事情经常会发生。

你当前所使用的工具也许是写玛雅中的大量的 MEL (玛雅嵌入式语言) 脚本。MEL 允许你根据自己特定的游戏开发需要，以一种自由而灵活的方式对玛雅进行定制。还有可能，你打算使用 3ds max 中的 MAXScript 语言。它们中间有很多相似的地方，但也存在有显著的差别。

也许你会使用一款为特定的游戏引擎而专门编写的独立编辑器——所有这些编辑器都具有各自与众不同的操作方法和使用方法。所有这些可能性 (以及其他情况) 就促成了“工具集的变换”问题。工具集的变换问题，使得在资源有限、进度紧张的情况下，要完成一个有意义的、功能强大的游戏编辑工作变得异常困难。

如果你是处在一个不断演变的游戏开发环境中，我的建议是熟练掌握一个三维软件包、一种第一人称射击游戏编辑器和一种实时战略游戏编辑器。另外，也很重要的一点，就是能够熟练地掌握基于 C 语言的脚本语言 (兼有结构化和面向对象编程的语言) 或者约束性的脚本语言 (如 Python 和 Visual Basic)。

很多用于游戏开发的工具集都非常复杂，不可能一下子就掌握。掌握的过程通常就像是从发射体中发射出的粒子一样漂浮不定、难以捉摸；也好像是在黑暗中试图戴着手镜去抓一群鸟一样。大多数的游戏开发过程都是一个渐进的过程，而且有时是难以忍受的学习过程。如果你正在学习制作低精度多边形的物体，或者正在学习环境建模，即使你不能在十五分钟的电视广告内制作出泰姬玛哈陵（Taj Mahal）的模型，你也不用感到垂头丧气。坚持就是胜利。就像很多的技艺和训练一样，要想提高就必须不断地练习。只有练习你才会提高。提高的过程可能并不是按你自己制定的计划进行的，但是你可以看到你的进步。



3.8 基础元素的案例研究

从本章的讨论中我们已经看到，在构造游戏世界时，光照、材质、粒子、效果以及音效处理的作用是多么的重要。要制作某些令人身临其境的游戏场景，通常情况下就得使用这些类型的东西。随着三维游戏的到来，出现了一些更接近真实的光照（现在，游戏中的光照逐渐向着动态光效发展，这使得光照的标准变得更高了！）、粒子和真实场景模型的东西。在最近几年的二维游戏中，光照只是简单地“画”到精灵上，游戏开发者也没有见过真正的粒子系统，而且那些单调的“环境”也仅是由一些区块化的精灵粘贴拼接起来的。

重要的是要认识三维给游戏带来的一切。除了一些新的性能和更多的动作种类以外，它还和游戏开发者带来了在类型和二维游戏略有不同的工作流。

随着主流的游戏从二维过渡到三维，软件工具也试图跟上这个过渡的步伐。在二维游戏开发的日子里，所有的资源（美术和音效）都是由游戏程序员处理的。程序员们把这些资源安装到游戏中并制作它们的行为。虽然当前正在开发的游戏无论从广度上还是复杂度上都像冲天火箭般地猛增，但是这种情况仍然如此。现在就一个游戏本身来说，已不再是一个创新或者新奇的东西了（尽管现在的游戏中也要有大量的新奇的地方）。现在的游戏是在高度竞争的商业环境下进行开发的，游戏的内容才是最重要的。要开发出成功的内容来需要将有限的时间和精力集中在游戏的内容上。非常客观地说，对于大多数的独立开发商，如果他开发团队为了完成像样的游戏引擎和工具包而努力拼搏，那他们就无法将精力集中在主要的游戏内容上了。这就怂恿了大多数的游戏发布商和开发商倾向于使用授权的游戏引擎（比如 Quake、Unreal 等等）。

对于工作中的游戏设计者和关卡制作者来说，有必要通晓每一个基础领域（光照、材质、粒子、效果和音效处理）。要记住，对于一个开发团队来说，要让这些在游戏开发中发挥作用会遇到各种形式的挑战。假如，根据团队的指导方针，你已经使用一套新的或者刚开发出来的工具制作了整个或者一部分关卡。但不幸的是，游戏开发通常是变化无常的，因此这些指导方针也可能会改变。而指导方针趋向于在整个游戏项目开发中最不合时宜的时候发生改变，有很大的几率是在游戏开发的中期发生。不管在什么时候，你都需要向游戏发布商说明你的开发进度。然而还有一些情况下，根据另一套指导方针或者为满足对游戏引擎的某个关键的调整、增强、扩展以及为满足新技术的需要，你将不得不进行返工并从根本上重新构造某个关卡。

记住下面这一点是很重要的：制作游戏的过程实际上就是一个创新的过程。而对这些基础元素信心百倍是你跳进“无尽的烦恼”中的第一步。

■ 月光下的森林示例

现在我们来看一下这个示例了。从你喜欢的游戏中（比如，《环境 2》——*Unreal 2* 或者《绝地武士 2》——*Jedi Knights 2*）选择一个环境编辑器或者从 www.wildtangent.com（可以访问“Developer Central”部分并下载 WTStudio 和 WTStudio 示例工程，这样的话你就可以在这些资源的基础上进行工作了）免费下载一个编辑器。试着用这些给定的资源来构造一个简单的场景。我构造了一片十分简单的森林，月光下，一棵树被闪电击中，并从背景中冒出烟来。下面列出了一些需要注意的地方：

- 要理解基本的光照概念；
- 材质的使用、变换与缩放；
- 构造一个基本的粒子发射体；
- 添加声音；
- 在场景中加载一个角色。

怎样来学习有关本章中所讨论的这些领域的更多内容呢？图 3-14 给出了一个简单的例子。下面给出的就是我用 WildTangent Studio 拼装这个示例的步骤：

1. 从一个默认的房间开始并将其尺寸改得稍大一点；
2. 在周围的水平壁面上设置空间材质；
3. 在地面上设置草地材质；
4. 构造两个圆柱。将其中的一个缩小作为切口，并且将较小的圆柱沉入到地面

- 下并在其表面上附上一个水材质，而在其四周附上石头材质；
5. 加载一些“树木”道具，并进行缩放，安置到适当的位置；
 6. 构造一个发射体，并给它指定一幅烟雾精灵位图；
 7. 构造两个点光源和一个会聚光。第一个点光源是全局环境光，放置在场景右上角月亮所在的位置。第二个点光源放置在泉水的表面以模拟水面发生的反射。会聚光要从月亮所在的位置照射到泉水上。



图 3-14 月光下的树林

RICK SANCHEZ 访谈录



Rick 是一位非凡的美术家，他在游戏界以及其他的娱乐领域中具有长期而又丰富多彩的工作背景。他先前的客户和雇主包括：*Activision*、*3DO*、*Sony Pictures* 和迪斯尼。他曾经开发过的游戏有针对个人计算机开发的《终极战区》——*Battlezone*、《赌城风云 2000》——*Vegas Games 2000*、《魔法门 VII》——*Might & Magic VII* 和《魔法门之英雄无敌 III》——*Heroes of Might & Magic III*，以及为迪斯尼在线开发的 *Jasmine's Bazaar*、*The Toontown Avatar Project* 和其他各种各样的游戏。Rich 还曾经在各种电影和电视拍

摄中担任过角色设计者、美术指导、故事情节串连图板美工，比如《逃出魔幻纪》——*Jumanji*, *The Magic Mill*, *Cybercop*, *Health Nuts*，同时他还在很多商业领域中获得自由作家的良好声望。

TM: 你好，Rick！光照在三维场景中到底有多么重要呢？它在游戏中又起着什么作用呢？

RS: 光照也许是三维场景中惟一能称得上最关键、最重要的方面。好的光照处理使你所构造的游戏或者场景表现出某种情绪和感情——而这种情绪和感情就是所有的艺术指导都试图从其想像中所获取的东西。现在某些让人难以置信的游戏，比如 *Halo* 中，大家都注意到的很重要的一件事情就是：在产生某种气氛时，应该如何处理光照以及它与环境和材质相互作用。

TM: 你制作过很多的材质，你是怎样成功地构造出这些材质的呢？

RS: 嗯，这是一个有些复杂的过程。我制作这些材质所使用的技术是我使用的最棒的二维图形程序……就是使用 *Photoshop!* 的过程中所积累下来的。不管怎么说这也是我个人的观点（我并没有受雇于 Adobe 公司或者是该公司的发起人，我只是担心你会怀疑它）。我只是“一见到这些软件就为它们叫好”而已。话题还是回到材质上来。简言之，我通常会注意到很多现实生活中存在的材质。我还会特别注意到那些细节部分，比如光线和某个表面之间的相互作用；还有颜色以及物体图案之间的对比，等等。

需要牢记的几点是：要使你的材质能够相互拼接！在对比度上不要有极端的区域（有些游戏引擎在远处构造了一些闪光或者闪烁的不规则物体……这让人感到很恼火且毫无吸引力！）。为合适的调色板保留一些参考性的概念美术作品，以便于将来三维光照的调整。像这样微小的细节不计其数。只要符合基本的色调理论与色调协调原则就可以了。

TM: 游戏中的视觉风格是在多大的程度上受控于材质处理？

RS: 可以肯定地说是百分之一百一十！让我再想一下……这是受控于光照和材质处理两方面的。

TM: 在材质制作工作中你见到过哪些错误的做法呢？

RS: 有重复出现的材质样式，闪烁，材质的饱和度太大，材质的对比度太小而看起来像是掉色了一样，过多地使用那些材质光盘或者材质库中已加工好的材质。事实上只要你能够对这些东西进行修改以获得所希望得的效果或者外观，那么使用这些东西是没有任何过错的。我见到过在很多的游戏中使用相同的材质。其实最好的办法通常是从头开始构造自己专用的材质。

TM: 你是怎样将一个场景中的材质、光照和特效结合在一起的呢?

RS: 不管你所使用的材质有多么好,要是光照处理不正确的话,它们也不会出现你所期望的效果,这样你也仅是构造了一个看上去很普通的游戏,这个游戏也仅是保留了游戏的实际结构——游戏的可玩性(你的游戏的流程、故事线索是什么样的,最为重要的是……游戏引擎本身的规划)。我们都已经认识到现在游戏中,只要看上去好看,你就成功了一大半了。

TM: 在提高材质技术方面你有三个什么样的提示呢?

RS: 一定掌握 *Photoshop*! 不管细节看起来是多么的无关紧要,也要注意它们(有些时候即使是最小的瑕疵或者不足,也能损坏你的材质而使这些它们偏离真实的东西!)。当然了,你也应该拥有大量的可以参考的书籍、照片和绘画等。我向你推荐 Birn 和 Maestri 编著的《数字化光照与渲染》——*Digital Lighting & Rendering* 一书。这本书包含了十分精彩的素材资源,而不需要你去理解其中复杂的渲染算法和高阶的数学知识。

TM: 在调整光照、材质和特效的效果时,会遇到哪些团队生产的挑战?

RS: 嗯,坦白地说,就游戏资源的制作、调节和分配而言,我还没有遇到过真正很难的挑战。归根结底就是为你的动画设计者、材质美工、建模人员和美术指导提供必要的支持,嗯,而且,甚至编程人员也会给你一些关于游戏的效果和场面中哪些东西可行、哪些东西不行的非常宝贵的信息、建议和见解。在一个开发团队的环境中相互交流是至关重要的。在保持游戏整体观念方面经常性地召开一些带有创造性的并能解决纠纷的会议是十分必要的——要使游戏的整体观念保持生动而有活力,解决掉任何的需要解决的故障和问题。就是在这种会议上使一些最好的(并且有时候是最怪异的)思想变成了现实。这是集体力量的结晶;也说明了那些成功的游戏是怎样开发出来的。

TM: 在参与制作游戏的过程中,你是怎样逐渐成为一名美工人员的?

RS: 在为游戏项目工作的过程中,你会不由自主地发展和成长为一名美工。当你在美术处理上陷入僵局时,那些日常的工作和困难就会迫使你去寻找问题的答案,寻找解决方案,要有创新思维、并且能够开发一种所谓的视觉欺骗技术来获得某种效果或出现某种图像——赋予材质或者光照正确的视觉效果。所有这些事情都能够使你发展成为一名美工或者一个有创造力的个体。

TM: 在你的工作中哪些专业的美术技巧最为有用?

RS: 可以肯定地说是我的绘画技巧及颜色搭配技巧。所有人都能够在几个小时内

学会使用计算机，在几个星期内学会使用某个软件包，并在几个月之内可以掌握该软件包……但是绘画技巧和美工技巧的积累则需要花费好几年的时间。更为重要的是，你必须每天都要进行练习。这和骑自行车不一样……一旦学会了就永远不会忘。至于美术，则是一项日常的基本训练；你必须每天都要练习绘画，否则的话你会失去这些技巧。

TM: 在游戏开发工作中你最喜欢的事情是什么呢？

RS: 最喜欢的是一个充满乐趣的工作环境，你不用进行正规的着装，并且有时可以得到免费的食物和冰激凌……你不要对此不屑一顾！

TM: 那么在游戏开发中你最不喜欢的事情又是什么呢？

RS: 我惟一的困难就是每天乘车上下班。我讨厌开着车……不管到什么地方。这也是为什么我喜欢自由作家这种工作。我仅需要偶尔的到办公室一趟，提交一些美术作品或者类似的一些东西。我喜欢待在家里通过计算机终端远距离地进行工作。我睡醒以后……甚至不需要梳理一下头发或者穿上正规的服装。我通常是从床上起来，自我补充能量，然后坐在电脑桌前边吃早饭边工作。而在我意识到时间不早时，又是该睡觉的时候了。以这种方式工作，至少有 85% 的时间我都在较高效率地工作。

TM: 我同意这一点。随着时间的推移，新出现的技术对你的工作方式的改变影响有多大？

RS: 在该产业中工作只会变得越来越好而且越来越有趣……与刚过去的几年相比，现在仅仅会遇到的各种限制已经为数不多了。速度更快的 CPUs，不太昂贵的图形显示卡，更好的软件和硬件，在线游戏，宽带，等等——我可以说出很多很多这样的东西。

TM: 对于从事数字美工方面的工作，你有什么实用的建议吗？

RS: 不要从事这种工作！（开个玩笑）实际上不是这样的。我会鼓足干劲来发展我的各方面的技能，但是要把精力集中在所喜爱的产业上。是电影还是广播？是特效处理吗？是角色动画吗？在这些领域中你想从事哪一个？是建模还是动画？还是粒子效果？

很显然，你必须掌握各方各面的二维和三维美术的基本知识，但是还要专门地深入学习最适合你的那一方面。不要试图从事所有的方面。如果你喜欢动画，那就集中精力学习角色的操作（正运动学、逆运动学、机构和包络线）[正运动学指的是利用关节的转动确定角色的姿势；逆运动学指的是利用动作控制器（IK handle）来对一些关节进行控制；而机构和包络线指的是基于滑动的关节或者节点模型的运动控制等]和动画制作。如果你喜欢建模，那么也要集中在某个特定的领域，比如多边形建模。尽你之所能

去构造最简洁的模型。但是也不要忘了继续使用非均匀有理 B 样条 (*Non-Uniform Rational B-Splines*, *NURBS*) 和样条曲线, 原因是新的技术能够让我们制作出更为令人吃惊的东西。在不久的将来, 即使看到具有影片质量的高清晰度角色, 你也不用感到惊奇。嗯, 最为重要的是, 尽量不要掌握两个以上的三维软件程序! 不要像我一样, 浪费大量的宝贵时间去学习所有的程序, 最终的结果是我真正会使用的还不到其中的一半……精通地掌握一种三维软件比熟习或者会用很多种要强得多。

TM: 在即将出现的数字化美术中, 你对哪些感到兴奋?

RS: 老兄, 我就是希望你不要提问这个问题。我对很多的东西都感到兴奋, 以至于我都说不出来很多这样的东西了(你知道, 我签订了非公开协议 [*NDAs*, *nondisclosure agreements*])。未来太让人感到惊奇了。要求标准也在不断地提高。现在的数字化效果已经让人难以置信了。这些技术已经赋予了艺术家越来越多的创造性或富于表现性的自由空间, 他们只会受到自己想像的限制。哎呀, 老兄! 我对任何的事物都感到兴奋!

TM: 现在使你所掌握的技能专业化有多么重要呢?

RS: 我们的工作环境通常是由制作的各个领域内有专长的人组构成的。你要遇到概念设计(有些人称为产品设计)、美术指导、建模、动画、环境、粒子效果、创作与材质。你必须很专业才行。

TM: 将来你想要制作哪些类型的游戏呢?

RS: 喜欢制作那些所见到的技艺最高超、给人以幻觉、情节紧张、感观刺激、令人精神振奋的动作和图形。

TM: 现在你还对什么样的游戏感兴趣呢?

RS: 《光晕》——*Halo*、《细胞分裂》——*Tom Clancy's "Splinter Cell"*、《机甲袭击》——*Mech-Assault*、《二战特种兵》——*Commando*……这就是我最喜欢的几个游戏。

TM: 在游戏项目和电视或电影项目中工作会有什么样的不同呢?

RS: 嗯, 在电影中工作是最需要艺术天分的。你必须是你所工作的领域中最好的一员。电影图像是由于其最终结果而使人感到满意, 当所有的图像会聚到一块时其最终的结果往往是很令人吃惊的。进行合成的最终控制时所有的东西都是在各层的基础上完成的。电视不像电影那样激烈, 但是由于时间和预算的限制, 其压力比电影还要稍大一点。通常都希望每个星期都能制作出一段情节动画——并不总是有充足的时间来修改动画或者重新制作一个角色。

虽然工作在游戏开发中你仍然需要在最后期限之前完工, 但却有着很多的乐趣。通

常情况下是一种更为宽松的工作氛围，在这里你能够按照某个稳定的进度工作，并且不会受到任何人的督促。要是你有了一份这样的工作，通常情况下会有一些纨绔子弟羡慕你的工作或者做一些比较评论，或者诸如此类的酷玩意儿。

TM: 要让你的美工技巧有活力，最好的方法是什么呢？

RS: 要经常注意其他的美术家并注意他们正在做什么。请相信我，没有比这更令人感到惭愧的事情了。这就会强迫你更加努力地学习。也可能是，这会使你在挫折面前知难而退，不再从事该工作。你必须要信任自己的想像，要持之以恒，并不断地自问：“他是怎样实现的？”

TM: 在你的职业生涯中，美术方面的工作是发生变化的？

RS: 由于现在你所拥有的工具，作为一名美工只会做得越来越好而且越来越容易。我至今还记得手工剪贴广告牌、杂志排版、手工上色、粘贴刻字等等——尤其是粘贴刻字。天哪！太痛苦了！

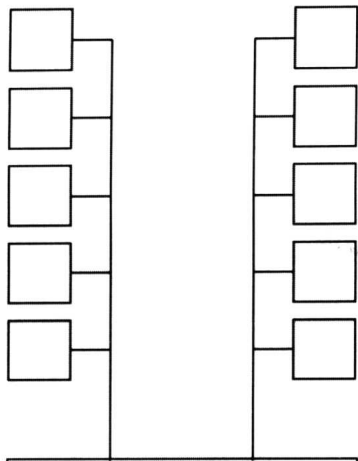
TM: Rick！感谢你花时间来和我们讨论这些事情。

MEGA 提示

1. 有关这些“基本”元素是怎样在游戏中发挥作用的，这方面的内容学得越多越好。要记住游戏世界是游戏可玩性的载体——而不是一个观看比赛的看台。
2. 当对关卡进行光照处理时，只使用能够满足要求效果的光源个数，一个都不要多用。使用节省下来的资源去考虑其他的关卡细节。在每增加一个光源前都要进行权衡比较。
3. 利用从建筑杂志上裁剪的方法构造丰富的环境方面的光照参考。
4. 研究灯光的颜色与情绪有怎样的联系。
5. 阅读照相和舞台灯光的书。参见附录 A。
6. 学习大师级的材质艺术家，像 Eni Oken 或者 Rick Sanchez (www.oken3d.com)。
7. 在制作材质之前就要想好打算怎样使用它。直到所有环境细节都确定下来再去处理最终的材质。
8. 制作效果时，尽可能多地使用精灵发射体而不是粒子发射体。（相对来说，精灵发射体不会占用大量的处理时间！）
9. 如果你要在你的游戏中使用粒子效果，而你的引擎工具系统没有这部分功能，

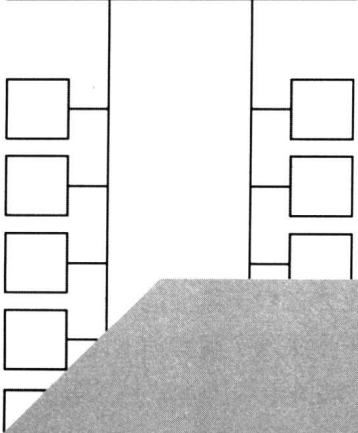
你得尽早作出决断：是导出数据（用专门的三维软件包）还是制作放在多边形平面的粒子动画序列。

10. 只在真正需要的地方才使用基于粒子的效果来突出某些道具或者特征。千万不要由于滥用而导致削弱了应有的效果。
11. 确信你的设计文档中的关卡或使命描述具体说明了大多数预期的音效制作计划。千万不要让音效成为一个附加的细节内容。体验一下使用更多的音效来推动玩家通过你的使命或关卡的方法。

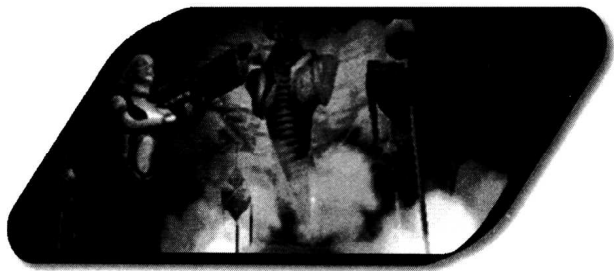


CHAPTER

4



**角色、道具、物品
以及镜头的细节**



你现在已经理解了从预可视化过程一直到开始使用照明、材质、粒子和音效细化你的关卡和地图的开发顺序。不同的地图 / 关卡 / 任务 / 游戏世界构造者都有着不同的工作风格，并且喜欢用不同的方式来安排他们自己的工作。个人的工作风格必须与整个团队的风格相吻合。毕竟，要“对付”目前这样大型的游戏的复杂度和广度，你往往需要与一个团队合作。作为团队中的一员，你的工作必须依靠某些工作要素才能完成，而这些工作要素也许会比整体的开发进度安排提前或者延后完成。通常，都会是后一种情况。

也许，你发现自己在等还没有完成的的工具，或者在等技术决议的制定。这种时候，你的工作就存在“依赖性”——你干实际工作的能力依赖于其他的易变条件、因素，或者资源问题。应该机智地利用这些等的时间，进一步深入了解其他的团队目标、解决或预见其他的问题。比如说，你可以利用这段时间扩展关卡和脚本软件工具的功能并增强其稳定性，或者为你正在使用的开发工具做注解（以便于进一步改进）。说不定用于将模型输出到游戏中的“模型导出工具”现在不能正常工作——别干等着有人来修理，还是转头做一些其他事情，比如改进动画或者制作更好的材质。

就我个人而言，我首先关心的是游戏的功能和流程。我问自己：“这幅地图看起来是否与构造它的意图相符合？”我并不是很担心灯光和材质这些细节问题，那是后面的事情。构造地图和关卡的大部分工作都和解决如下形式的问题密切相关：“怎样来完成这个或另一个关卡的设计目标？”在开始进一步选定最终要使用的光线、材质以及粒子系统等细节之前，尽量确保地图和关卡运行平滑自如。

在构造阶段，你会碰到许多新的资源——新的角色模型、新的道具、新的物品，等等。这一过程的绝大部分工作都是构造这些游戏中新加入的“零部件”。通常，游戏整合的过程和这些资源反反复复精心修改是同时进行的。

正如前面章节里讲到的那样，你经常需要用占位图符（placeholder graphics）来

表示游戏中的要素。例如，用一个立方体作为占位符来表示喷射酸液的喷泉。等到酸液喷泉的最终版本（模型和动画）建好以后，通过使用文件参考，用这个最终的作品来替代那个立方体占位图符。与此同时，你应该尽力去选定游戏的操作特征。选定游戏的这些特征的其中一部分工作包括布置角色（主角、敌人以及非玩家角色）、道具（环境特征，如椅子、箱状物以及圣坛等），物品（比如威力升级和收藏物）以及协调个别镜头细节。你的镜头细节决定了镜头或者玩家视角在游戏场景中怎样移动或者运行。因此，在本章中，我们将要深入探讨角色、道具和物品的布置以及有关镜头的一些考虑。



4.1 布置角色

一旦你所负责的关卡或使命中的游戏世界、环境外观（将来游戏比赛的主环境）通过前边描述的过程（或者其他什么类似的东西）构造完毕，接着就该让你的游戏逐渐成型了，这部分的主要工作有：布置对象，以及对它们进行测试或者书写脚本。书写脚本的方法细节将在后面的第6章中做详细介绍。现在，让我们先来看看着手布置不同类型的通用角色的方法。

■ 主角

一般来说，主角就是你的游戏中的明星。玛丽、Crash Bandicoot 和 Samus（《银河战士》——*Metroid*）都是主角的典范。当然，主角们通常都拥有最强大多变的技能和最复杂的与环境交互的能力。既然主角是由玩家来操纵进行闯关的，他的起始位置通常是在地图或者关卡的默认起点上（比如说关卡中心位置的一个花园）。这一点常被称作孵化点（*spawn point*），因为主角是在地图上的这一物理位置“出生”的或出现的。这一位置就是玩家控制角色在游戏中开始历险的地方。如果还没到达进一步闯关的“可存储位置”，而所有的“生命”又都耗尽，这一位置也是玩家重新闯关的出发点。

你给主角或者其他角色（对于多玩家游戏）布置的那一位置就是他们行动的原始起点。主角从起点处向任何一个方向都能通过关卡，这就是自然的多米诺效应（由某一事件引起的连锁反应）。

不同的工具，在处理主角的布置上所使用的方法也不尽相同，但本质上都是在场景中加载主角，并定义一个物理的开始位置作为他的出生地。通常，开始时会有关卡或者使命的介绍和 / 或电影片断来指导玩家闯关，然后，游戏画面切换到角色的等待状态动

画，表示主角正在等待玩家对它的控制。也许这时主角正弯起他的手臂、敲打他的脚。

主角通常都会有几套不同的等待状态动画。从哪个等待状态动画开始玩游戏可以是随机控制的，这样，玩家就时常会看到不一样的主角等待状态动画，或者也可以根据孵化点的固有特征设置特定的动画。必须在主角的人物特性中确定主角出生或者孵化时选择哪一种等待状态动画。

也许主角在一幅地图上只有一个孵化点，或者有几个对应于存储的游戏信息的孵化点。当然，这些点都可以调整和定制。

■ 敌方角色

毫无疑问，你希望你的主角以多种方式遭遇怪兽、接受挑战。这些怪兽和生物就是你的敌方角色。记住，“角色”并不一定是指“人”。敌方角色可以是植物、爆炸物或者爆炸舱、烫人的马铃薯、发疯的机器或者汽车零部件，以及被施了魔法的银器。

就像布置一间逼真的鬼屋，并决定哪些小动物从墙里面跳出来、从天上跳下来或者是爬上来访者的脚踝一样，你需要安排你的主角，让他去经历一个全数字化的历险过程。

当你把敌方角色布置到场景里以后，就可以改变他们的属性或者特征。它们会给主角造成多大程度的伤害呢？他们是如何向主角表现敌意和发动攻击的？他们是否会与另一派别或另一部分中的角色或者非玩家角色敌对？它们是战斗到全军覆没，还是在伤亡达到百分之八十时就逃跑？回答这些问题能帮助你确定如何编辑敌方人员的属性或特性设置。在这一过程中你将看到这些编辑工具有多重要。

怎样设置游戏中的敌方角色取决于好几个因素。一般来讲，设置得好与坏和设计人员的玩游戏的直觉（从你花费在玩游戏的数千个小时中获得的洞察力）、游戏测试、游戏的不断改进以及多次的试验有关。对于不同类型的游戏制作以及你所效力（试图提供的游戏体验）的团队的总体游戏目标，敌方角色的设置在很大程度上都不一样。

当你开始布置敌方角色的时候，有两个必须要考虑的因素，就是游戏的难度增加幅度值（*difficulty ramp*）以及冒险/奖励值（*risk/reward value*）。难度增加幅度值指的是游戏的难度随时间的变化程度。跟你一样（很有可能），我已经玩过太多的游戏以至于难以记清它们，也无法计算玩游戏所花的时间，我仍旧无法说清是从什么时候开始精通一个不会给我任何喘息机会的游戏。对我来讲，我没有从让我连喝茶的时间都忘记了的游戏中得到任何的享受。

大多数游戏都有如下特征：整个游戏的难度增加曲线明显呈拱形，从开始的简单关

卡到后来难度逐渐增大的关卡。每一关卡中同样也有一个难度增加曲线。有些关的难度增加得非常快，然而，其他关的难度却是随时间缓慢地增加。作为一个游戏的制作人员，你希望时刻都能意识到你的关卡或使命安排在整个游戏的什么地方合适。这是一个属于不断调整敌方人员的人数和力量的问题。当然，很多游戏都有可以由玩家设置的难度值，你可以根据从玩家那里返回的测试反馈意见相应地调整（选择）你的地图或关卡。把你的自我主义扔到一边，广泛地、诚恳地听取玩家的意见。

冒险 / 奖励值仅仅指的是以下这个前提：为了得到奖励，玩家必须参加冒险或者被置于危险境地。这种情况下，玩家就能得到奖励的分数。然而，你必须小心设计它，以保持游戏的平衡。如果有些物品太容易获得，玩家就不会有收获感和成就感；刚好相反，如果某些物品几乎就不可能获得，玩家会因为他们的努力没有得到回报而感到泄气。

图 4-1 用一个简单的示意图说明了冒险 / 奖励的概念。在没有与敌人对抗的情况下，受伤的主角可获得少量的生命值或健康能量（用 L 表示）；而在遭遇被三个敌人（用 E 表示）围攻的冒险活动，玩家获得的奖励却是一件宝物（用 C 表示）和一次武器升级（用 W 表示）。以上就是一个很经典的游戏情节。你可以将这种思想应用到你设计的多处游戏情节中。你一定要使用图注来将地形图上的各要素区别开，并且要记住，大多数要素的位置安排问题最终都要靠游戏的实战测试来指导解决。

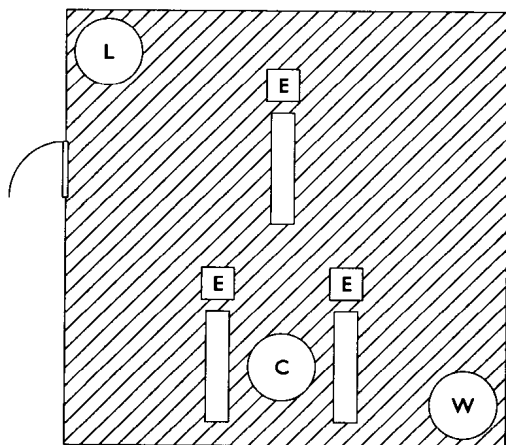


图 4-1 冒险 / 奖励游戏情节

冒险 / 奖励构成了经典的游戏对抗情节。从这一点来讲，这种思想就是要将“对抗”

一词扩展到一个新的、有趣的境界。每一位游戏设计人员都一直在努力实现好的情节。你随时可以在游戏中找到证据，比如做的更加复杂的非玩家角色、更加智能化的敌人以及部分可摧毁的环境障碍物——玩家可以通过摧毁这些障碍物为自己创造出入口。这种情况下，我们就有办法避免与敌人的对抗。需要逃跑路线吗？用手榴弹在墙上炸个洞。如果游戏环境中大量的墙可以开凿成路的话，就会增加额外的游戏实战测试问题。你完全不能预测玩家会选择在哪个地方做这种事情。这就是为什么很多游戏都已经采取了只允许关卡里的某些部分（使用一些特殊标记或细节说明，玩家可以从视觉上识别它们）可被玩家摧毁的对策。那样的话，你就知道玩家能在哪里的墙上炸开洞，也就可以实战测试这个地方，并且围绕它构造、筹备你的游戏场景。

敌对危险以及不可见数据

与设置敌方角色有关的区域也包括了设置“敌方”危险。危险可以采取多种形式：熔浆池、酸液坑、滚烫的热发生器；每一样我们都曾经在许多游戏中见到过。我们希望能够判断并编辑这些危险对于主角或者其他任何角色（那些我们希望它被危险所影响或者伤害的角色）的伤害程度。通常，我们都是使用所谓的不可见数据来完成这项设置工作。

我们可以用不可见数据标示或标明场景中的区域或物品，不可见数据是从美工开发包中导出，并由游戏引擎读取的。我们可以在不可见数据编辑器里设置不可见数据的定义，编辑器位于你所选择的 3-D 游戏开发包里。通常，我们给不可见数据一个名字和一个 ID 号，像 Damage024。我们可以把“熔浆池表面破坏”定为 Damage024。我们甚至可以更加明确规定甚至可以构造一个不可见数据的类型，如 lava_pool_d024。对于你们的团队来说，拥有文件和数据类型命名公约（商定文件命名规则）是非常重要的，这样的话，每个人只要一看文件的名字就能知道它的类型。

我们还希望将不可见数据与游戏场景中的建筑物要素绑定，比如多边形物体的表面、定点或者实体部分。对于熔浆池，我们希望它的表面有破坏力，于是，就把不可见数据与熔浆池表面（就是构成熔浆池表面的多边形）绑定。

下一步，将不可见数据 Damage024 与一种颜料颜色关联起来。通常，最好按类型保持它们之间的一致性。举个例子来说，同一种类型的破坏就尽量使用同一种颜色（例如表面破坏 = 红色，可产生破坏的对象 items = 桔红色，等等）。

图 4-2 中所所示的是不可见数据编辑器，图上左边视图中的附有标签或标记的所选表

面与 Damage024 的破坏类型（例子中的熔浆池表面破坏）相关联。那样，我们就可以为用户定制已经设定的多边形表面的标签的值。我们可以通过编辑这些值来控制当主角与多边形表面相接触时受到伤害的大小程度。

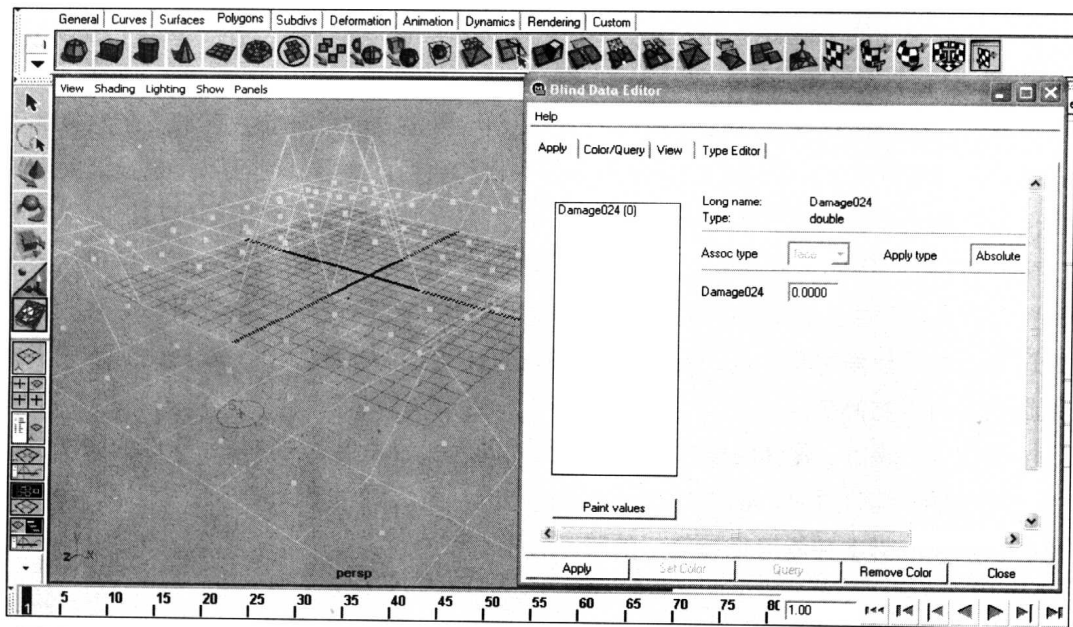


图 4-2 不可见数据编辑器

■ 非玩家角色 (NPC)

在前面的章节里，你已经看到了在游戏中非玩家角色的多种用途：作为背景故事的引出点以及提示主角沿着一条确定的路线游历游戏；用作参考点；或者是教玩家游戏控制以及展示游戏的独特之处。更多的情况下，非玩家角色正逐渐成为严密行动的不可缺少一部分。非玩家角色能够与玩家一起完成规定的目标。在单人游戏中，它们可以成为玩家的同盟。

将非玩家角色放入场景的方法和其他角色是一样的。它们的模型、材质以及动画数据在孵化点加载并生成。它们的特性、战斗状态和能力都可作为属性来设置。它们的行为可以用一个控制脚本（定义它们行为的脚本）来扩展或者用程序控制处理。要让非玩家角色以你所希望的方式行动，并按照你想像的那样表演，往往需要大量的脚本的书写、

修改和编辑工作。这些内容将会在第6章里详细介绍。

■ 头目角色 (boss)

头目就是游戏中“关键的”坏蛋。关于头目的想法从游戏严格按关卡来驱动之时就有了，完成游戏的关键就是击败关底头目或者坏蛋。击败小一些小的头目（小型头目，minibosses）是玩家闯关的关键。

我们现在仍然在游戏中使用头目角色，只是不再像以前那样非把它们放到屏幕的最右端。在三维游戏中，我们也一直使用小型头目和头目。小型头目可以用来强调关卡或者使命中某一个较为激烈的部分。也许，一关就是一场战斗，主角按顺序与兽穴里的吸血鬼战斗，而打败它们的头领就是一次与小型头目战斗的经历。击败关底头目仍然被作为整个行动的顶点、游戏的高潮、以及玩家玩游戏的最高成就。小型头目和关底头目的处理方法和其他角色基本是一样的，不同的是它们的个头比别人大、威力比游戏中你所见到的任何事物都要强。

设计以有趣的、具有挑战性的方式来行动的大头目角色，往往依赖于你的脚本管道的技术水平。换句话说，你的脚本必须是动态的，而且要功能强大，还要求你的游戏引擎必须能读取脚本并能够把编辑好的脚本转换成动作。



4.2 道具的类型以及用法

从功能上来讲，道具在整个游戏场景中有很多种的使用方法。任何一个随意摆放或有意布置在游戏环境中的物品和工艺品就是道具。所有的道具可分为三个基本的种类：

- ▶ **静态道具** 那些不能改变和毁坏的道具，比如酒吧场景里的桌子和凳子，或者牢狱场景中的小床、厕所和水槽。它们仅仅是按照你的愿望布置整个游戏环境，让环境看起来（希望是）更加真实。我们的游戏环境中不仅有家具，还可能有垃圾、脏乱的东西和乱放的物品，等等。
- ▶ **可毁坏道具** 那些在游戏环境中有多态的道具。它们不是静止的；它们随着周围环境的改变而改变。举个例子，一个潜在的可毁坏道具——天花板的支撑圆柱。如果它被摧毁，变成瓦砾（比如被导弹攻击），你需要一个粒子效果（尘埃云）、一个柱子倒塌过程的毁坏动画以及这个柱子的最后的新状态（倒在地上、碎成一大堆碎块的柱子）。这个圆柱道具，当它被击中时，由完好状态

变成了坍塌状态。想一想你玩过的游戏中几百个可毁坏道具的例子：盒子裂成碎片，气罐爆炸，等等。

- **脚本道具** 这是最复杂的道具，因为可毁坏道具所需要的每件事物脚本道具一般都需要，而且还需要更多。假设我们要向一个里面有果冻野兽的外星人孵化器射击。我们希望这个野兽被激怒并还击射击它的人。我们需要一切可毁坏道具所要求的东西（玻璃震裂的粒子效果、破碎的孵化器以及死去的野兽），而且还要写一个脚本来触发还击动作。

■ 道具的应用

利用这三种道具的每一样，你都可以做任何很酷的事。你可以用大量的组合方式把它们合起来使用。下面只是几种使用道具的方法：

- **布置场景** 对于很多场景来说，比如公园的地面、内景以及海底世界，要想让它们看起来感觉不错，就必须要用道具合理地布置它们。
- **打靶练习** 有些人把破坏东西作为他们的嗜好。对他们来说，在可毁坏道具上进行打靶练习就是医生的命令。这种毁坏通常都没有什么奖励（你也许能产生一些什么东西来），但它能够帮助玩家学习如何使用他们的武器——尤其是当道具在移动的时候。
- **破坏和揭密** 利用可毁坏道具来给玩家暴露隐藏的珍贵收藏品和秘密通道，或者引出地图上的捷径或隐藏的位置。
- **调用脚本动作** 道具被破坏时可能会让玩家的敌人发动进攻，或者会导致任何数量的其他种类的附带结果。
- **简化难题** 道具可以成为玩家解决难题的部分线索，比如要到达高处，玩家可以搬动梯子道具；或者作为平衡物（在平衡板物品上移动游码道具）；或者作为开关按钮（一个阀门道具）；或者用来阻塞敌方的入侵（破坏某物来封锁房间入口，从而把入侵者堵在外面）。
- **用作环境标记（路标）** 为了向玩家透露位置信息，可以采用如下道具形式：气势宏伟的雕像、火山熔浆源泉或者独一无二的关卡特征。这些道具可以帮助玩家明白他们当前所处的关卡位置。
- **提供幽默和情感要素** 道具可用来产生幽默，用作很酷的小把戏，用于引起情感反应，还可以用来使玩家心情放松或者紧张。

■ 安排道具

在既定的技术限制内，现在你可以在游戏场景中布置大量的道具。具体如何布置还要由它们的用途和功能决定。你可以想像，在大多数情况下，道具的布置都伴随着大量的改动和反复操作。你可以在布置场景的时候体验一下布置工作。牢记角色和敌人，并不时地考虑考虑。角色应该能够在场景中自由地游历。你可以利用道具来引导玩家通过关卡，也可以用它们来生成路径。



4.3 物品和威力升级的类型及其布置

物品和威力升级物的种类包括出现在大多数游戏中的几个其他的要素，如下所列：

- 健康或生命能量的补给物；
- 武器升级；
- 武器弹药；
- 可存盘的位置；
- 收藏物（例如戒指和星星）；
- 特殊物品（例如远程通信设备）；
- 特殊技能升级（例如，增加远距离射击的特异功能）；
- 特殊威力能量（例如，施魔咒的神力）；
- 特殊移动拾取物（例如，能够使角色完成独特的动作的拾取物）。

如何布置这些物品？通常，你的地形图的图例细节和设计文档已经详细说明这些东西已设计好的布局规划。怎样布置这些物品取决于你怎样考虑敌对冲突以及怎样帮助地图上的主角完成任务。对于大部分情况来说，这些物品会支持玩家，也会考验玩家。有时，规划好的物品布局会运行良好，而另一些情况下，这些规划可能会彻底失败。

寻找合适的物品安放位置基本上就是在地图上多次的改进过程。这是一个感觉问题，常识就是你的出发点。如果设计了高强度对抗的区域，你就会希望在附近布置健康或者生命能量的补给物。你还有可能希望利用局部区域内的武器升级和足够的弹药来帮助玩家对付棘手的敌人。用一些隐藏起来的宝物在地图中构造一些“探索”，让玩家来赚取这些特殊的物品和威力升级。如果把这些特殊的物品摆放在玩家面前的地面上，那么他们几乎不会有什么成就感。把得到并收藏这些物品的过程变成一种挑战，但是，别

把这种挑战设计得不可战胜。

听起来显然是这样，让这些收藏物在游戏世界里“生动”并且有意义，并确保它们在游戏中有效果。游戏中，没有人真的愿意拾起一些没头没脑的东西，它们对于玩家的功力只有很少的或者根本没有能感觉到的效果。物品必须要有一定的价值。

另外，请记住对于死亡竞赛、集体游戏、单人游戏以及合作游戏这些模式，布置这些物品的方法是不同的。研究你最喜欢的地图中每一个这样的区域，搞清楚别的游戏开发者在做什么，然后详细制作并精心修改。这样，你会想到一些布置的模式并尽力让它们新鲜而有趣。如果你的玩家一路苦战通过了关卡中的各部分并且冒了一些风险的话，给他提供一个可存盘的位置（为控制台）。



4.4 镜头的考虑

三维游戏使用程序控制（在游戏代码中）的镜头处理系统为玩家操纵三维场景中的视野。镜头和场景中的其他事物一样，可以沿着三个方向（ X 、 Y 和 Z 方向）运动。随着动作类型的不断改变，镜头的视角也不断地改变。比如说，也许你想让玩家能够拉近镜头做近距离（直接交手的）格斗，拉远镜头为实地行军的部队转移显示路径，然后在镜头到物体的极限距离内的任意距离之间拉近或者拉远镜头。

有时，玩家并不想让镜头连续跟踪（让镜头跟着某些东西来回移动）这些位置，而是只想简单地跳转过去。这种跳转动作在动作脚本或者玩家控制下迫使它跳到另一个位置时经常会出现。镜头从赛车司机座位上的第一人称视角的位置跳到跟踪或跟随那辆赛车的第三人称视角的位置上，这就是一个很好的跳转镜头的例子。

这些镜头典型的移动或跳转都是由镜头脚本来控制，这种脚本是可以编辑的，它定义了（说明了）给定的场景中的每个镜头如何运动或操作。镜头的妥善处理可以使得所有不同的情节环境都适合于玩耍，而在实时的环境中，这是一项艰巨的任务。如果镜头跟得太近，你就感觉（看）不到周围物体的运动。如果镜头离得太远，你又会和你的角色失去联系。如果你强迫镜头推进和拉远得太多，就会使玩家感到不舒服和注意力不集中。

注意你所玩过的那些游戏里的镜头是怎样动作的。那些游戏中一定会有大量的镜头跟踪（跟在某个东西后面）和镜头移动（从点A慢慢移动镜头到点B）。但是，比起这两种基本动作，镜头更多的是做跳转和切换动作。调整镜头来观察动作是构造关卡或者

使命中相当大的一部分工作。让我们来看一看一些通用的镜头模式并讨论它们的用途。

■ 界面细节

镜头扫描常常被用来增加用户界面 (UI) 的视觉冲击。例如, 玩家能够看到镜头扫过赛车的前面, 或者, 为了强调一个给人深刻印象的卓越的战士角色的优势地位, 镜头以较低的角度扫过他的前面。为了满足镜头的运行要求, 你可以对镜头扫描进行编辑, 设置镜头的开始位置和结束位置, 以及扫描的速率或定时系数 (多快或多慢)。这样, 你就可以给玩家一个立体的、引人入胜的观察角色或车辆的视角, 这要比单纯的平面图像生动得多。

角色视角实例

另一个有关利用镜头来详细描述界面的例子就是按照非玩家角色的视角来操作镜头 (例如, 有关你的司令官朝前射击的特写镜头)。随着三维面部动画中先进的对口形技术的出现, 你可以给非玩家角色设计一个特写镜头, 并使用游戏模式 (或者定制模式) 和玩家 “对话”。在很多游戏的设置中, 非玩家角色都会 “插嘴” 评论游戏性, 给玩家提供线索, 偶尔还会提示玩家或者给玩家指路。设置好一个用来操作这些非玩家角色的对话镜头后, 你就可以随心所欲地扩展和增强非玩家角色的作用。你甚至可以构造一个非玩家角色工作室 (例如, 一套非玩家角色在镜头中和玩家交流的画面)。使用呆板地走过屏幕的非玩家角色镜头的日子已经一去不复返了。更进一步来说, 利用一个简单的镜头脚本, 在你的非玩家角色窗口中, 你可以把镜头从一个非玩家角色摇向另外一个 (就像桥上两个在指挥的海军中校一样)。

■ 浮动镜头

许多游戏开始给玩家提供找出或者定位一个动态镜头的机会, 这样的镜头就可以在玩家的选择下由一个目标单元移向另一个单元, 或者从一个位置移向另一个位置。例如, 镜头可以快速地跳到该目标单元关卡中的格斗场景处, 还可以快速跳出该场景查看整个战场上发生的事。

有时候, 为了让玩家能够灵活地控制和沉迷于游戏中, 这些镜头移动都预先设定好了, 可以由键盘键或者按钮输入来进行简单地选择。如果你打算让玩家定制一个浮动镜头 (或者是几个), 那么你至少应该给游戏提供一些默认镜头设置——在你自己玩游戏

时工作情况良好的镜头设置。那些欣赏这种大位度的镜头控制并且想要进一步编辑镜头位置的玩家将可以从这些默认镜头设置开始。而那些对此不关心的玩家将会有可选择的这些默认值。

小心处理镜头的移动。过多的镜头跳动和转移容易使玩家变得找不到方向并感到迷惑，而且玩家被迫经常性地返回合适的视角这种杂事又会破坏游戏的娱乐性。

■ 固定镜头

固定镜头视角也许是玩家最常见、最熟悉的一种视角模式。对于第一人称视角游戏来说，固定镜头视角就是主角视野的平视视角。当然，这种叫法适用于当主角在地图上行走时“相对于”他或她的眼球的视角。如果让角色跑进一个角落或者岩石狭缝里，角落或狭缝就正好是你所看到的東西。

然而，在第三人称视角的动作类游戏中控制镜头，会带来独特的挑战。在很多第三人称视角的动作类游戏中，你以第三者的角度观察主角，这类游戏都有一个固定镜头跟踪主角，镜头在主角头顶后上方大约45度或更大的角度的位置。这种镜头在开阔的地方使用效果非常好，但在仅仅能容身的地方、墙壁之间或者狭窄的物理空间，有可能会产生很多问题。

在你玩第三人称视角游戏时（比如PlayStation 2平台上的《霸王神剑》——*the Mark of Kri*游戏就是一个镜头控制的好例子），注意一下主角进入空间狭小的角落时镜头是怎样动作的。通常，镜头会自动调整，为你提供一個观察主角较好的视角。在实时设置中，确定应该把镜头放在什么位置来保证主角视角是相当具有挑战性的——虽然有几款成功的游戏让我们感觉这个很容易。事实上并非如此。对于我们，在设计方面的挑战就是如何确信玩家的视角决不会被完全遮挡，并且确信镜头最大程度地配合游戏的动作。调整镜头系统的细节问题也是游戏设计员全部努力的一部分。

■ 特殊情况下的镜头

有时候，因为视觉差异和游戏可玩性，你要构造一些特殊情况的镜头来做一些特别的事。也许，你想让玩家在驾驶的同时还要移动一个六英寸高的机器人；也许，你想完成一个玩家弹跳到摩天大楼上的镜头；也许，你只是想为游戏过程中的某一特别部分的动作细节做一个独特的观察视角。依据这些特殊要求，你可能会相当轻松地实现这些特殊情况的镜头……也有可能就是一场恶梦。另外，作为一个游戏设计人员，实现基本的

镜头定制的能力基本上就是对你的镜头编辑工具能力的检验、以及它们怎样和游戏代码接口。如果用固定了的镜头作为特殊情况镜头的话，你消除了一些问题，但付出的代价是游戏的附加功能。

在这儿，处理镜头细节的编辑系统和脚本系统的能力非常重要。许多工具只能处理基本的或者有限的镜头移动编辑工作。要让镜头以精密而复杂的方式“执行任务”，你需要做游戏代码级的改进工作。

■ 常见镜头问题

你的镜头系统可能会冒出一大堆问题——事实上，这类问题多得不胜枚举。作为代替，我选择了针对许多镜头系统的两大类问题。

第一类问题出现在镜头被“卡住”的情况。你可以想像，当你正围绕场景移动一个镜头或者一组镜头时，它们可能会碰到物体（并且不知道怎样处理这个碰撞）。这会造成一种情况：镜头在场景中的某个物理位置卡住了或者跳到一边不能动了。造成镜头被卡的原因有好几种，维修它们也有几种办法，但是，镜头被卡实在是一个值得一提的常见问题。我所能提供的避免这个问题的最好办法是：从一开始就保持镜头编辑简单整洁（好好地计算规划，不要试图一次做得太多，让镜头按照简单的脚本步骤逐渐变复杂起来，而不是试图一次性就达到要求的最后效果）。另外，确信将镜头控制代码中旧的和/或不相关的镜头位置信息清除掉。

第二类常见问题出现在游戏动作的镜头视线被某个物体遮住的时候。在第三人称视角游戏中，当主角在关卡中某一墙角深处或者进入关卡中的地下墓地时，玩家的视线常常会被挡住。有些游戏解决这个问题的办法是：当主角被挡住时，挡主角的墙壁就会变得透明，或者通过使用X-射线技术来让玩家看见主角。

有些玩家在游戏中利用镜头遮挡这一点，故意让主角站到角落里，这样隔壁的墙就变成了透明的，他们就能看到等在那间屋子里的敌人了。这是你被迫给予玩家的功能性上的牺牲，因为在镜头被角落或小块区域挡住的时候，就需要在所有可能的情况下都有效的一定范围或者广度的透明区域。

其他的一些游戏并不是使用墙壁自动变透明的方法，而是让镜头自动移动，使主角以某种方式返回到视角中。这两种方法都很难实现，但确实能解决问题。采用这些解决方法，在解决问题的同时可能会带来一些严重的测试问题，因为地图上的每一个角落深处都要为它们的自动透明或镜头自动重定位甚至二者皆有的处理措施进行测试。

最好在一开始设计游戏时就计划好这些事情。这听起来简单，然而在创作的热情高涨以及许多其他的大量细节问题之下，它可能被适当地考虑过，但后来又被遗忘了。

如果你正在用周围都是墙的或者很紧凑的空间来制作一个第三人称视角游戏，尽量事先就预计出你会碰到什么样的镜头问题。用这个预计结果来影响你的早期的制作过程。一定要保证当你做了一个很糟糕的镜头移动、彻底解决这个问题时，你不需要花大量的时间和资源。

和很多其他的问题一样，镜头问题也是使得游戏设计员和技术领导之间的不断交流尤为重要的一个原因。如果你打算开始构造环境中的有用资源（就像是“堆砌”角落深处的墙壁块），你必须及早和镜头的实现达成一致。



4.5 有关演员加载和镜头调整的评论之案例研究

现在你理解了演员的加载（将角色、道具以及效果导入你的游戏的过程）是构造游戏的途径。同样，它也是游戏组合过程中至关重要的一部分。和我们前边所讲的一样，大多数游戏开发者用玛雅和 3ds max 来制作粗糙的多边形人物、道具和物品。我们已经简单介绍了材质、动画，还有可实施的光照技术。现在，我们要将它们从三维开发包中导出来，然后再导入我们的游戏引擎（授权的或者是游戏工作室内部专门定制开发的——也叫专用游戏引擎）。

软件导出器（从玛雅或者 3ds max 到我们的游戏引擎）其实就是一小段工具代码，它以适当的方式重新组织我们构造的有用资源，使得我们所使用的游戏引擎能够“理解”它们。或许，游戏引擎需要我们以某种方式构造多边形面部，或者以某种方式给动画接头定向。软件导出器必须要考虑这些情形，并且有时还需要尽量调整这些情形（得到混合的效果）。

通常情况下，我们的导出工具和正在创作的游戏是并行开发的。这就意味着当我们尝试用软件导出器来构造东西时，它本身正处于严格的调试阶段。就单独这一个事实就会给我们带来相当多的挑战，这并不奇怪。设计人员应该给工具程序员提供快速的、有用的反馈信息，以便导出器能够正常地工作、支持游戏开发。通常，用这些工具构造的可直接用于游戏的有用资源应归于游戏发行商，而与此同时，这些工具尚处于开发阶段。

在我个人的记录中，我参加过很多游戏的开发；在这些开发周期中（我不会专门选出一个来谈，它们有着共同的要素），都没有也没必要一开始把游戏引擎制作得功能非

常强大，那样会降低游戏的开发速度，并且会影响到游戏可玩性的开发目标——就是工具。准确地说，是缺乏这些工具。如果你连要加入场景中的可靠的有用资源都得不到的话，你怎么开始制作精彩的游戏？你将无从下手。

很多时候，这些软件导出器无法正常工作。那些在你所选用的三维开发包里看起来完美无缺、功能正常的东西（指有用资源），在（用你的软件导出器）转换给游戏引擎的过程中给毁坏了。最终你可能会发现，多边形面部翻转了、材质信息丢失了、光照信息不见了或者被严重改变了、动画数据没有了、还有接头处的局部转轴（运动的连接位置信息）也被破坏了。所有的这些加在一起足以让整个团队灰心丧气。

同样地，一旦角色能够可靠地加载到游戏中，如果你还没有强大的镜头调节系统，你仍然无法进行依赖于游戏的镜头移动编辑工作。为了制作你的团队向往的那种游戏，你必须能够以某种方式在游戏引擎中编辑和精心改进镜头移动。

游戏开发中的这个“挑战点”再次突出了开发强大的工具的重要性。实质上每个开发者都在为之努力奋斗的——或多或少。

小心计划并仔细考虑，对于如今的游戏市场来说，你的工具开发的选择在制作精彩和极具竞争性的游戏上绝对是生死攸关的。尽管游戏的种类和复杂程度还在不断上升，但是在很多个开发案例中，时间分配以及预算支持的数量已基本保持平稳甚至有所下降。这样，开发者就需要有令人难以置信的功能和多样化的构造工具。

你可能已经了解，这种思想正成为一个游戏开发界的事实：如今许多游戏开发者专门研究某一种游戏流派（运动类游戏、角色扮演类游戏、即时战略类游戏、第一人称视角类游戏，等等）。要从一种流派跳到另一种流派很难，因为每一种流派都有各自独特的开发需求（上至技术支持，下至工具细节）。

NATHAN HUNT 访谈录



Nathan Hunt 是 *Collective* 公司的首席游戏引擎程序员，*Collective* 就是大受欢迎的《魔法奇兵》——*Buffy the Vampire Slayer*、《印第安那琼斯》——*Indiana Jones* 以及 Xbox 平台上的《帝王古墓》——*Emperor's Tomb for Xbox* 游戏的开发商。在他以前开发的游戏当中，受到好评的有《虚拟台球 2》——*Virtual Pool 2* 和《虚拟台球 3》——*Virtual Pool 3*。目前，他正加紧为开发下一代游戏控制台发展新的游戏技术。他在 *UCLA*（美国加利福

尼亚大学洛杉矶分校) 获得了他的计算机科学学士学位。

TM: 你好, Nathan! 作为一个首席游戏引擎程序员, 能简单介绍一下您的工作任务吗?

NH: 我的工作就是共同监督并管理多平台、多游戏的游戏工具和技术代码库的开发, 并监管一个有 20 位工程师的部门。这包括技术设计、编写代码和很多核心技术系统的维护, 还有, 从管理这方面来说, 帮助确定我们的游戏有合理的时间范围和具有竞争力的特征集。

TM: 全职“游戏设计”在这个产业中还是个相对较新的职位。在工作流方面, 设计人员怎样才能和编程人员以及技术领导最好地配合?

NH: 要制作一个成功的游戏, 一个关键的因素就是设计人员和编程人员之间的有效交流。要做到这一点, 最重要的就是设计部门要确保他们对全部游戏有一个确定的、不变的想法。另一个关键要素就是设计人员必须能够为编程人员兑现一份儿低阶的、功能驱动文献资料——含糊不清的或者需要解释的资料很在第一次执行时有可能出错, 或者在执行以前就会遇到障碍, 甚至会导致信息匮乏。最终, 设计人员必须愿意妥协。当一个设计人员按照规格书来进行工作时, 也许他会很“熟练”, 但是当技术开发的时间可以减掉一半时, 他们的个性因素只损失百分之十, 通常这是一个划算的交易。

TM: 强大的软件开发工具直接关系到游戏内容的丰富程度。有些开发者使用 MAXScript 把工具构造到 3ds max 中, 有些使用玛雅的嵌入式程式语言, 有些编写他们自主开发的“单独的”软件工具, 而另外一些人必须使用带有授权引擎解决方案的工具。这就导致了我所说的“工具转换”, 而且当开发者在产业内频繁更换工作时, 这也会使得他的经验变得零零碎碎。你怎么看待近期的设计工具的实现?

NH: 我不认为这种工具的参差不齐在近期内会不复存在。到目前为止, 从一家游戏开发公司的设计师到另一家的设计师当中, 关于最有效的开发工具还没有一个的大致的统一意见。然而, 只要设计者能够变通并乐意听取意见, 他或她仍然能够在这样的环境中生存下去。即使制作工具的性能还很差, 创新性的环境构造的概念仍然可以不依赖于特定的工具而存在。

TM: 对于如今的设计员来说, 学习脚本语言有多重要? 你建议他们掌握哪一种语言? Python? Visual Basic? 还是其他的语言?

NH: 与编程人员和美术人员类似, 设计人员这一职业在产业中越来越专业化。对设计员的知识水平和掌握的技巧的期望值也在提高。虽然并不要求设计人员掌握一种脚

本语言，但是没有生产技术的设计人员（“纸上设计者”，就像他们有时说的那样）会很快就会被淘汰。设计人员最常见的两种工作就是构造环境和书写脚本。假定你选择了书写脚本，我想说的是用哪一种语言根本就无关紧要，因为每一个游戏都可能用与别的游戏稍有不同的方法（并且底层可能用的是不同的语言）来写脚本，最重要的是脚本人员/设计人员首先要理解书写脚本的工作流，其次是乐意适应（或许甚至是协作共同设计）任何最适合当前的游戏开发的脚本系统。而且，掌握任何一门现成的语言都能为进一步的培训打下良好的基础。

TM: 随着分配给游戏开发的时间的减少，而玩家对游戏内容的要求在不断提高，你对工具设计中的“模块化”思想有什么看法？你是否认为大多数开发者都将制作自己的工具，这样他们就可以像一个数字乐高拼装玩具系统一样组装游戏内容？

NH: 这是近来每个游戏开发工作室都渴望达到的目标。当然，他们还都有一定程度的不足——当工具多到足够可以使用构造模块类型系统“垒出”最先进技术游戏的时候，编程人员也就该被淘汰了。但是我坚信，直觉的工具设计对于在合理的时间范围内制作出一个高质量的游戏是绝对必要的。工具在开发期稳定得越早，设计组和美术组用来制作和改进游戏内容的时间就越多。

TM: 假定我们制作了一个面向动作的角色扮演游戏。我们想有一个简易的（而且完整的）脚本支持的拖放式结构系统。你是倾向于在玛雅（比如说）里安装它还是编写一个独立的编辑器？为什么？

NH: 关于“理想的”开发工具至今仍然没有统一的说法。随着游戏开发中的许多东西的发展，每一种选择都有正反两面。如果一个游戏的开发工作即将开始并主要由设计人员来执行，或者这个游戏拥有相当广泛的内容，也许编写一个独立的编辑器更有意义，当然要求这个编辑器重点放在简单的环境和游戏事件构造上，而不是复杂的建模能力。另一方面，如果打算让环境在风格上错综复杂，没有玛雅或者类似的程序，这一点是很难实现的。据我猜测，大体上，因为游戏继续在美学方面竞争得越来越厉害，我们将看到更多得开发者走结合玛雅（或者 3ds max）的路线。

TM: 你认为要把构造工具与游戏引擎完美地结合起来，其中最大的三个挑战是什么？

NH: 挑战 1：为了让开发团队有足够的时间来制作游戏内容，工具必须相对较早地完成（或者至少能够让用户制作出具有产品品质的作品）。但是这一点只能在有很固定的计划以及技术上和设计上的协同工作才能实现。

挑战 2: 在设计和技术之间保持一个稳固的交流渠道是必要的, 这个渠道也是开发过程中易于断开的地方。设计和技术双方必须是开放式的合作, 而且必须要乐于妥协、目标明确。

挑战 3: 当某个工具或工具特性不能正常工作或者需要改进的时候, 团队成员必须要能够作出判断, 同样重要的是, 他们还必须能够判断出何时工具特征“足够好”。如果对工具的修改工作一直延续到开发阶段中很晚的时间, 那么, 最后一旦设计人员得到完整的工具集以后, 就会有一个疯狂的、令人讨厌的匆忙阶段。另一方面, 如果太早就确定了工具而导致工具的功能很有限, 那么设计员将很难在没有大的痛苦和困难的情况下制作游戏。

TM: 对于不同平台 (比如 *Xbox*, *PS2*), 这些挑战有区别吗?

NH: 我认为它们都与开发平台没有关系。

TM: 在制作游戏中作为一个技术领导或指导, 你最想从游戏或者关卡的设计人员的日常基础中看到的是什么?

NH: 每个设计人员都必须认识到开发是由设计来驱动的。没有设计在背后支持, 即使是世界上最精彩的游戏技术也无法制作出很棒的游戏来。而且, 失败的或靠不住的设计总是会浪费掉程序员大量的“人月 (一个人在一个月时间内)” (或者在极端的情况下, 得用“人年”这个单位) 工作或重复性的工作而告终, 因此必须极度严肃地对待设计人员这项职责。没有完美地设计人员, 当然也没有一开始就很完美的游戏设计——然而, 处于任何水平的设计人员都应该始终努力提高他或她的设计技巧, 还必须尽早认识到什么时候游戏设计上的一个细微改变可能使游戏变得更加出色或者让团队的工作效率更高。

TM: 在内容开发过程中, 你是怎样调试工具的?

NH: 在工具开发的初期阶段, 团队必须要意识到制作理想的工具的过程就是一个学习的过程, 使用一件工具制作的最早的几件作品往往只能算是成功了一部分。但是一旦整个游戏的开发处于进行中, 改变或调试工具的过程必须非常小心地处理, 并且这个过程必须更加小心, 因为一旦出错可能会报废大量有价值的工作, 这就使得随意改变或调试工具的代价更加昂贵。

TM: 在提高游戏可玩性的过程中, 让设计人员整天修改行为脚本或者执行脚本是否有益呢?

NH: 当脚本管道高效地构造起来以后, 它可能是优秀的设计人员所掌握的强大工

具。因为高水平的脚本可以控制游戏的品质、感觉和个性，让一定数量的设计人员全天候地处理这些脚本绝对是有意义的。这些设计人员的数量跟项目的需求有关。

TM: 能告诉大家你在游戏技术开发方面的一些目标吗？

NH: 就我个人来讲，优秀的游戏要比高超的技术更重要。所以，我的第一目标就是开发工具管道和具有竞争力的技术来推动我们的团队制作出一个真正优秀的游戏，在此，我也广泛呼吁大家。技术是达到目的的手段，并理想地在一个已完成的游戏范围内高效地使用。

我的第二个目标是要实现一个游戏，它基于一个设计得非常好的最初计划，按时、按预算完成，而且开发组对它感到完全满意。制作游戏是一项商业活动，既然游戏的质量极其重要，所以这就是我的底线。

TM: 可以谈谈如今的设计人员设计的主要脚本“类别”（镜头脚本、道具行为，敌人行为）是什么吗？

NH: 不同的游戏对这个的定义有很大的区别。但是，我猜想基于设计人员的脚本形式，用得最多的是定义角色的特性。它的范围从动画序列到高层次的角色人工智能思考过程。从某种程度来讲，时间触发和对象交互也可以被认为是脚本的一种形式，即使它们是在可视化工具中制作的，例如玛雅，而不采用脚本语言来定义的。这就是今天游戏设计人员有那么大比例的时间用于游戏开发的原因。

TM: 调整游戏的“感觉”的要素是什么？游戏设计人员制作脚本时应具备的重要素质是什么？

NH: 成为一个优秀的编剧需要天赋，并不是每个设计人员都有这样的天资！首先，编剧要完全理解游戏所有想像的景象。就游戏的风格而言，什么是理想的最终效果？它是否与别的设计人员的作品风格一致？以及怎样才能让执行人员知道何时执行它？第二，成为一个优秀的编剧需要很大的耐性。要让一个东西打到预想的效果，几乎经常要做是无休止的重复、游戏测试以及改进工作。最后，编剧应该知道什么时候系统或者特征不能有效地工作，这一点只有极少数设计人员才能掌握，或者在什么时候一个合理的、全局性的改变能够戏剧性地提高整个游戏的感受——这类全局的决定需要与全体编程人员合作来完成。

TM: 你想给那些对制作游戏感兴趣却又不知道从哪里入手的游戏开发者提些什么建议？

NH: 第一，大量地玩游戏。要知道你将在哪方面和其他人竞争，知道这个不会有

什么坏处。尽量找出游戏中你感觉薄弱的要素，还有那些你感觉很强的要素。第二，保证你一直在自学关于游戏开发的最新的思想。有很多网址都包含了大量的关于游戏制作的各个方面的信息，新闻网址可以帮你指出游戏产业的发展趋势以及值得关注的东西。第三，保证你每天都在提高和磨炼你的交流技巧。书面交流的技巧在游戏开发中正一天天变得越来越关键，当然和队友进行简明地口头交流的能力也是绝对必要的。最后，一旦你感觉你在可以做一部分游戏开发的工作了，就尝试着去参加面试吧！很多公司都乐意培训那些渴望学习、虚心并表现出一定天赋的那些人。

TM: 原型设计游戏思想或者游戏机制有没有可能不会涉及大量的编程和美工资源？

NH: 这个问题很难回答。很多情况下，一些形式的原型设计会有这样的可能。例如，简单游戏构造工具包。或者，一些游戏可以用不同的手段制作其测试版本（例如，对于角色扮演类游戏或者回合制战略游戏，就有可能可以用“纸上谈兵”的游戏版本来测试游戏中一些要素）。可是，一个伟大的游戏远远超过了其各部分的总和，而且，在看见所有的要素按计划和在一起运转之前，很难说清楚一个游戏是否会具有真正的魅力——高出一切竞争者的魅力。

TM: 你认为游戏产业以外的那些人对游戏开发有些什么样的错误观念？

NH: 错误观念 1：“我喜欢玩游戏，所以不用动脑子去想，我一定会喜欢制作游戏”。这是一个对简单理由的误解。创作游戏很有趣，它也是一种娱乐，但是制作游戏却是一件工作，也是一种职业。这并不是说一个人不能热爱这个职业，而是说常常会有这样的时刻：作为游戏开发的一部分，一个游戏开发者不得不面对一些没有感染力或者令人难受的东西进行工作，仅仅是因为不得不这样去做。

错误观念 2：“我只在制作我喜欢玩的那类游戏时才感到快乐”。大多数人都对他们所玩的游戏非常挑剔。事实上，在市场上随意挑选的游戏中，真正为一般玩家所喜欢的不会超过百分之十。但是，每个游戏都有能够吸引人的东西，那也是值得它的创作者骄傲的东西，而且，每一个设计成功的游戏都将会吸引一些人。所以，一个有抱负的开发者应该抱着“我能学到什么？”的态度，以后等到他职业生涯中的经验逐渐丰富起来的时候，再去担心怎样制作完美游戏的问题。

错误观念 3：“在我看到一个游戏设计的时候，我能判断出它的好坏，因此我是一个设计者。”指出何时游戏设计得不好或者游戏设计不能正常运行是游戏开发中的一项很有价值的技能。但是，这只是作为一个好的设计者的开始。而识别出问题的下一步的

工作自然就是建议解决方案，但是不是每个人都能做到这一步的。

TM: 你认为在游戏产业的业内人士中关于游戏开发存在着什么样的错误观念？

NH: 错误观念 1：“我的经验丰富，所以我不愿意开发 X 型游戏了。”很不幸，高人一等的优越感 (*Elitism*) 在游戏开发者当中很普遍。很多情形下，考虑到单调乏味的工作远在他们的能力之下，大家都拒绝做这样的工作。但是职位较低或者资格较浅的雇员可能能够胜任给他们的特别任务，那些为自己的工作感到骄傲的人将会一直愿意做伟大的工作，并且会即时完成它。

错误观念 2：“我总是工作到深夜，所以我不一定要遵守常规的工作守则”。不错，游戏开发者们常常要长时间加班，因为各项任务规定的最终期限太紧张了。但是，游戏的制作应该在专业的环境中，环境中的每个个体都应该尽力跟上时间表和计划，以此来尊重和支持他们的同事，同时也获得他们的支持。当一个队员开了一整夜的夜车，然后在接下来的两天中不顾团队中的其他人而溜掉以恢复体力，这并不是开发游戏的最佳方法。

TM: 培养对游戏的直觉的最好方法是什么？

NH: 不同的人对这个问题有不同层次的倾向。但是，大家可以通过两个方法来不断改善和增强它。第一个方法就是玩大量的游戏，这也是最显而易见的方法。了解其他的游戏怎样运作，这是让这类知识变得根深蒂固的最好方法。第二个方法来自于开发游戏的经验。偶尔的挫折和失败是教会你游戏中哪些是“对”哪些“错”的最好的工具。

TM: 有什么“不为人知的”开发故事要和我们分享吗？

NH: 我想不起来了。

TM: 可以和我们一起分享你对自己成为前沿游戏开发者的成长之路的见解吗？

NH: 作为一个游戏开发者，我学到的关键一课是组建团队，这也许是制作优秀的游戏中最重要的一方面。如果团队由作为独裁的个人来主导——不管他有多强——都很难制作出由全体成员通力合作、相互支持的团队制作的同样品质的游戏。在过去的十年里，团队的规模一直在不断地扩大，这种趋势在短期内也不会停下来。我相信未来几年里最具优势的游戏开发公司会是那些能够把握怎样让团队成员互相合作的公司。

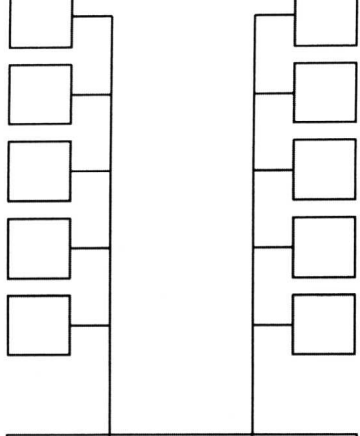
我还发现，如果游戏开发中的大部分工作都与创造性的、富有灵感和革新的事有关的话，那么，大量的时间就被用来做那些艰难的、通常还是紧张的工作，有时甚至还要做那些看起来非人的工作。我认为要成为一个对游戏开发作出贡献的很有价值的人，关键是他要接受这样的事实：游戏中的每一项开发任务都需要由某一个人来完成，并且为

了得到有趣的材料，我们每个人都不得不去经历痛苦的时刻！

我学到的另一课是——我感到对于任何单独的游戏开发者而言，最重要的、而且也许是被誉为高于其他因素的，是当他面对挑战时能够始终保持冷静和明智的能力。游戏开发领域改变非常频繁，因为它还是一个“年轻的”产业。那些始终需保持领先地位的人就是那些能够快速分解并解决新的、未预见到的问题的人。

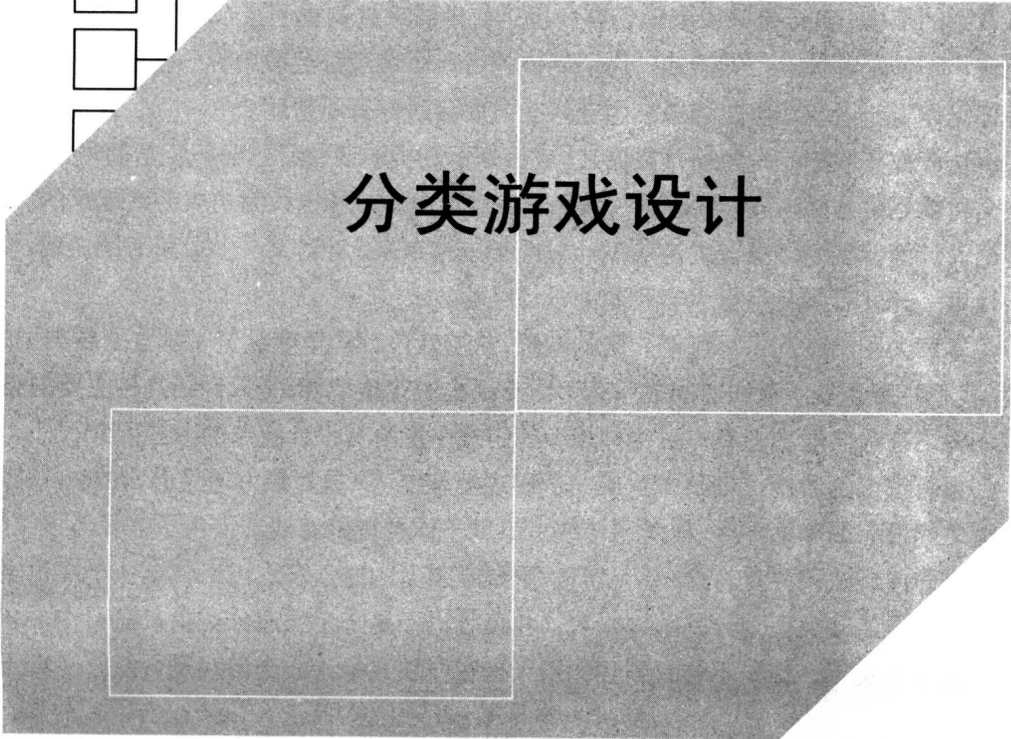
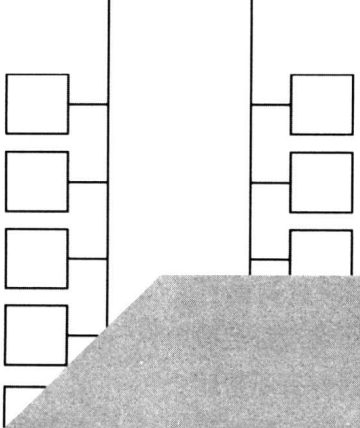
MEGA 提示

1. 早日计划你的工具需求并深入到细节问题。
2. 使用象征性的图注在地形关卡图上进行角色、道具、危险物和物品的第一次布置。确信整个团队都能理解这个图注。
3. 在你的游戏设计文件中的关卡规格书里写入“镜头注释”。这些注释应尽力事先说明镜头运动和镜头脚本可能会出现的问题种类。
4. 早一点计划你的非玩家角色，并尽量让它们灵活易变。为加入的行为的调节、新的或更好的游戏思想、重建以及为改善非游戏角色与玩家的接触体验需要做的调节留出余地。
5. 确信你的工具可以完成任何复杂的大头目的行为动作。
6. 按静态道具、可毁坏道具、或者脚本道具来区别道具并对它进行分类。你甚至可以通过文件命名的方式来构造这种区别（例如，*s_barrel* 为静态类型，*d_pillar* 为可毁坏类型，*sc_altar* 为脚本类型）。
7. 尽可能早地实现工具的稳定性。在游戏开发曲线上及早确信从三维开发包到游戏引擎的有用资源管道功能正常。

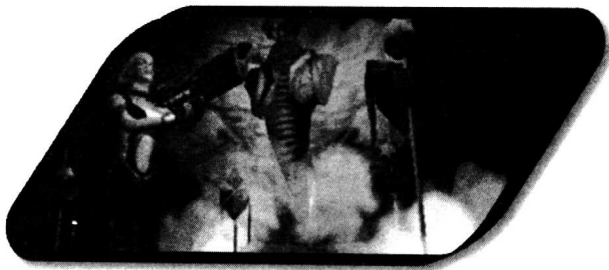


CHAPTER

5



分类游戏设计



到现在为止，我们已经讲述了相当数量的素材，在实际中构造游戏世界时会用到这些素材。我们所讨论的为数众多的元素都适用于构造各种类型的游戏。然而，每一种类型的游戏都有特殊的设计上的细微差别和要求，同时也有其特定的和独特的挑战。一个完全归属于某种特定类型游戏的设计者，能够针对任何一种主要类型的游戏的设计细节写一本完整的书，事实上确实已经有这样的书了（比如，*Swords & Circuitry: A Designer's Guide to Computer Role Playing Games*，作者是 Neal 和 Jana Hallford，2001 年由 Premier 出版社出版）。我鼓励你以任何可行的方法尽量地对一些类型的游戏进行更为深入的研究，只要你对这些类型的游戏具有强烈的爱好。正如你在第 4 章中所看到的那样，甚至是开发者他们自身也正变得越来越侧重于某个类型的游戏。请牢牢记住，不管是哪种类型的游戏，你所制作的每一个游戏都会教给你一些很重要的东西，在你制作下一个游戏时，这些东西在某些方面会对你有所帮助。

也许你喜欢的是角色扮演类游戏，或者是运动类游戏、模拟类游戏（常简称为 *sims*），还有可能是射击类游戏。我是通过编写文字冒险类游戏和街机射击类游戏起家的。没有哪个设计者能在各种类型的游戏设计方面都成为专家。针对某一个或所有的游戏，每位设计者 / 开发者都会有一大套不同的设计经验。

本章力图给你提供尽可能广阔的视野，从游戏设计的角度纵览了那些主要类型的游戏。我不打算涵盖所有知道的或所有可能出现的游戏类型，而是只介绍几种主要的类型。我们将站在最高层次上看一看每种类型的游戏中影响其设计特点的一些因素。

也有很多的游戏很难归属到某一个游戏类型中。这是对最好考虑而言的。各种游戏类型之间没有明确的界线，或者是传统上不属于该类型的游戏却借用了很多与该类型相关的概念或玩法，并且将这一切以新的更具吸引力的方法表现出来，这些一点也没错。当然我不是游戏类型划分的纯粹论者，并且我也不是一个强迫分类主义者。

通常情况下，你是不会看到我和别人坐在一块儿争论一个游戏是否能够“真正地”

称为某类游戏,比如说角色扮演类游戏,或者是这个游戏仅仅包含角色扮演类游戏的要素却混杂了一些其他类型游戏对它的影响。我从来不用游戏类型去衡量一个游戏,我只是对游戏体验的本身进行衡量,而不管除此以外的任何东西。你可以选择以任何方式把它们混合在一起——我只对汤的味道感兴趣。对于我来说(并且我相信对很多其他人也是如此),游戏体验决定一切。

简单的东西有时也具有欺骗性。它起作用吗?它是不是有趣?我是不是有那种在热烈的游戏“嗡嗡”飘然的感觉?太美妙了。我愿意被“粘”在椅子上一直玩游戏——直到我听到门前送来了早报吗?太精彩了。噢,我应该出去跑步了。

我们将纵览一些主要游戏类型的重要设计要素,并以考虑电影艺术、背景故事以及对白的创作结束,这就是这一章我们要展开讨论的。从整体来看,这些东西对所有的游戏类型都适用。我们将通过对游戏设计人员的日常工作工具进行一个总结以结束我们对游戏类型概观。



5.1 运动类游戏

有些玩家不喜欢运动类游戏,另一些人则百玩不厌。有一件事儿可以肯定:在提高控制台游戏技术方面它们比其他类型的游戏做得更多。足球、棒球、篮球、曲棍球、英式足球、田径赛、网球、橄榄球、赛车、冲浪、钓鱼、溜冰、乒乓球、自行车越野赛、高尔夫球……它们都被视为电子游戏。在游戏产业滞留足够长的时间后,你就可能想要开发一个有关运动类的游戏。下表中集中列出了一些有关运动类游戏设计方面的注意事项:

- 运动捕捉和手调动画;
- 已制定的规则;
- 模拟与想像;
- 授权;
- 大量的微调节;
- 防守脚本;
- 玩家和赛场的视觉与感觉;
- 数字统计。

运动类游戏经常要求游戏的角色(基于一些著名的职业运动员)既能够完成巧妙的

竞技动作，又能够完成那些完全个性化的肢体动作。这可以表现为一个独特的击球姿势和转身，一个独特的投掷传球动作或者是一种大力灌篮动作。在运动类游戏中玩家希望获得并且很看重控制这些动作的“能力”。这就使设计上需要考虑各种角色的动作。

开发者可以通过移动与三维角色模型相连的交点和骨架来“手调”完成这些动作的动画，也可以利用运动捕捉技术（有时甚至需要二者的结合）来完成。运动捕捉过程需要给参与者/运动员穿上一种上面带有小反光球或者反光点的衣服，通过追踪这些反光体的运动，扫描系统将它们（的运动信息）读入到计算机中。然后，就需要把这些数据有效地处理成为可用于游戏的动作数据。因此，运动类游戏设计的第一大障碍就是使玩家的动作在游戏中能够正常发挥作用的同时，还要让它看起来准确而熟悉。图 5-1 中给出了一个穿着动作捕捉服装的运动员，你可以看到服装上面的反光点。

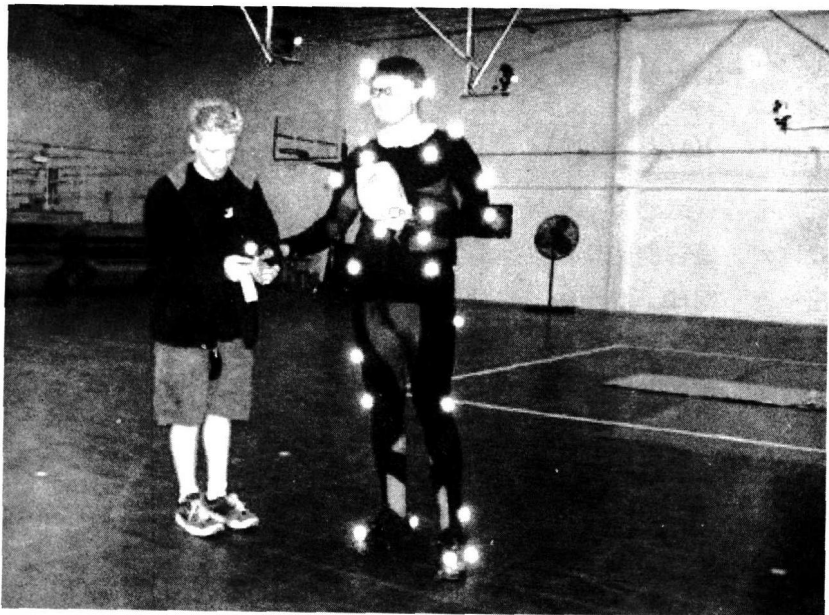


图 5-1 运动捕捉服装

运动类游戏有已成文的规则。在设计很多其他类型的游戏的时候，作为设计人员的你都肩负着游戏规则的构造任务。具有代表性的是，对于标准的运动类游戏，你需要把某一特定的运动规则融入到你的游戏中。就游戏设计方面而言，为了保持最佳的游戏体验，你得考虑一个典型的游戏流程以及它的规则是如何影响玩家的。你将会经常对这

样的抉择：是否给玩家提供禁止掉游戏中的一部分裁定规则的功能——因为这些规则可能会破坏了玩家的“动作感受”（比如，在冰上曲棍球游戏比赛中关掉“死球判定”（*icing call*，故意将球打出防守方区外的行为）选项）。我们应该尽自己的最大努力将对应的运动项目中的全套规则都引入到游戏中去。

因为“模拟与想像”二者都在运动类游戏中有所体现，所以我们经常要在其中寻找一个设计平衡点。那种试图实时再现体育运动中每一个微小细节的彻底的模拟游戏只可能在某些顽固派模拟爱好者中找到卖点，而会失去混合型动作类游戏玩家的拥护。比如说，棒球游戏中可能需要模拟这样的情节：接球手发信号要求传球，而投球手在投球前可以接受、也可以不理睬这个信号。对于游戏中简短的和近距离的“动作点”，你可能会把投球手发出信号的动作都减掉。简言之，如果你想吸引动作玩家，你就需要大量“浓缩”的动作点。这就是为什么在作出设计决策时紧记你的“目标玩家”那么重要的原因。你应该始终以你所预期的玩家为准来作出一个平衡的决策。

运动游戏体验富有想像的一面在于它公然对抗运动游戏模拟的一面。一个纯粹的棒球模拟游戏可能会让你花费掉九年的时间去玩一个AA / AAA棒球比赛！玩家希望比赛能较快地进行，进攻和防守都要快而且有好的效果，并且可以在午饭时间（译注：游戏比赛时间缩短，玩家很快就可以结束一场比赛）进行职业对抗比赛。这就是游戏过程中的部分乐趣所在。如果玩家被冗长烦闷的比赛所压抑，再加上很差的实际控制能力，那么你将会失去你的玩家。

授权，就是指许可使用著名的职业球队，选手和大型体育场和竞技场，它将影响到你设计的“范围”。在过去的一段时间里，不得到那些主要运动联盟的授权许可（在游戏中不使用他们的球队、球员……）你也可以开发运动类游戏，比如说美国职业棒球大联盟（MLB）或美国国家篮球协会（NBA）等等这些组织。从技术上讲，你可以这么做，但是在各种他们喜欢的运动项目的游戏中，大多数玩家更喜欢那些能够看到和控制他们所熟悉的天才队员的游戏。还有，将你自己置身于球队和选手的控制中，这也是想像的一部分。

运动类游戏需要大量的微调节。*微调节*是指对动作、功能或数据进行尽善尽美的编辑，直到游戏中的每一个动作或事件都“表现”正确。动画中从一个动作到另一个动作的转换是很复杂的，而且必须“无缝地”连接在一起。运动类游戏经常需要对各种细节进行编辑，如球的尺寸比例（球在各个不同表演地点的大小）、接力区（什么时候球从一个队员传到另一个队员？）和捕捉区域（控球或者丢球的范围有多大？），还有许多

其他的特定细节功能。

大多数运动类游戏包含一个玩家与 CPU (计算机) 对抗的模式, 因此还必须编制计算机与人类选手对抗的进攻和防守程序。这需要复杂的脚本和程序设计。有一个现成的例子, 如果有两个后卫和一个守门员守着曲棍球门, 同时侧翼两个边锋和一个中锋以某种独特的队形攻向球门, 计算机应该怎样让后卫占据最佳位置? 要将一个玩家的直觉判断写成代码是很一件很困难的事。

这就是为什么运动类游戏都要把比赛规则编制在脚本中的原因……这些规则能够被人类运动员很容易地发现和利用! 在进攻和防守脚本中的规则通常呈现出一种直观条件的形式 (比如, 如果冰球位于冰上这个象限或者区域内, 就将在这个范围内的这几名运动员的防守设置成某个形式, 并根据对手玩家的位置做一些微小调整)。

对于运动类游戏设计者来说, 另一个挑战就是让运动员和赛场从感观上都能恰到好处。人们会盯着这些运动员和场地看上几个小时。他们都知道这些运动员和场地是什么样子的。为了让这些东西能够取悦你的玩家, 而你必须以对授权人 (那些给予你使用著名运动员和赛场许可权的当事人) 来说是尽可能完美的角度来展现运动员和赛场。为了得到运动员正确的模型、材质和面部细节, 通常需要巨大的工作量和修改量。让那些“个性化”动作看上去准确可能是你面临的一个真正的挑战。大型体育场和竞技场也应该同样对待。在将它们引入游戏中之前需要进行相当规模的建模、材质和光照方面的修改。

你想要将某种体育运动的最新统计信息加入到你的运动游戏中去。然而, 就像游戏开发一样, 体育运动不是在真空中进行: 运动员要转会, 运动员的统计信息每天都在变化, 球队的标志图也会更改, 大型运动场得到新的合作资助商, 吉祥物会换成新的面孔, 等等。在你尽力给玩家提供最令人兴奋的和“真实”的体验时, 你必须尽你所能向玩家提供这些变化的信息。这需要相当细致的跟踪并且很不容易完成。



5.2 格斗类游戏

在各种不同“口味”的游戏中, 格斗类游戏还是很流行的, 其中不乏一些杰出的游戏, 像《铁拳 4》——*Tekken 4* 和《真人快打: 死亡盟约》——*Mortal Kombat: Deadly Alliance*, 都是非常经典的游戏。我们来看一下影响格斗类游戏的一些设计细节:

- 运动捕捉格斗动作;
- 控制器调节;

- 竞技场;
- 格斗动作及相关项; 组合动作;
- 道具的使用;
- 多种格斗设置;
- 扩展格斗模式。

与那些精心设计的体育运动动作相似,格斗类游戏需要悉心制作一些来自格斗者的身体动作。而且,为了保证动作的流畅性,在开始制作的时候经常会使用运动捕捉。在设计方面,为了构造格斗类游戏中所要求的动作,对于动作捕捉所需的每一个单独的动作都必须加以规划和说明。动作捕捉过程是非常昂贵的,因此你必须完成认真的方案设计,以保证对游戏中所需的所有动作都进行了捕捉。图 5-2 给出了单张电子数据表格的样本条目。你的游戏中可能会有成千上万条这样的动作和过渡动作(两个动作之间的动作,或者后续动作)。通过参观 *House of Moves* (一家动作捕捉的服务公司) 的互联网网址 (www.houseofmoves.com) 你能够学到有关动作捕捉的更多知识。

1	Asset ID#	Description	Hooks from	Into	Notes	Performer
2	jsk44781	Jump Sweep Kick	Fighting Stance	Fighting Stance	N/A	Yen Otsugo

图 5-2 电子表格的样本条目

很多团队都将每个需要进行动作捕捉的动作从游戏设计文档中导出到像微软的 Excel 这样的电子制表软件中,或者他们应用像 NXN alienbrain (www.nxn-software.com, 是一个完整的数码生产管理方案,曾为好莱坞特效业提供项目管理解决方案) 这样的资产追踪软件来构造一个动作捕捉列表。你可能还会考虑计划一个“弥补”过程来捕捉那些遗漏的、改变过的或者从原始的动作捕捉中替换出去的动作。

一个成功的格斗类游戏的很大一部分“感觉”在于游戏控制器的处理上,也就是说,按钮触发的“发射”或者“解释为”可控的格斗动作的速度有多快。游戏控制台通过游戏引擎每秒钟重复“扫描”输入很多次,而且,提高格斗响应时间往往可以降低动作执行过程中协调控制器与动画处理代码之间的滞后。对于游戏代码,当它把流畅的格斗角色控制读入到控制器中并转换为角色动作的时候,它就面临着一个系统性能工程的问题。接下来就是动画之间的协调,一定要保证格斗者模型动画间的过渡(从一个动作到另一个动作,比如从一种格斗姿势过渡到肢体抓取)没有干涉、没有掉帧并且没有不协

调的动画间的连接过渡(动作怎样开始和结束;是怎样从一个动作连续地变换为另一个动作的)。

现代格斗类游戏游戏中的竞技场也做了扩充,从十分流行的固定视角滚动式决斗游戏到在全方位竞技场中进行的决斗类游戏,这些竞技场中有精心设计的摄像机系统并可与环境交互。这就开阔了体能对抗的空间并使格斗游戏看起来规模更加宏大,但是这就会增加对背景资源新需求,要求具有挑战性的美术专家和设计者构造出各种类型广泛的、引人入胜的竞技场。如今的竞技场所已允许角斗角色在整个场景中漫游,随着战斗的进行。能从一个地方移动到另一个地方。这就强迫设计者必须构造出更加深思熟虑的战斗情节以及多种角斗场场景、各种天气效果等等进行配合使用。

对于同样的多按钮的组合进攻和独特的角色能力,格斗的技术却变得越来越复杂了。为了能够“最终通过漫长的回合”,玩家的战斗策略始终是具有挑战性的,玩家必须学习在会损失生命值的高能量攻击和经历更加庞大的战斗所要承受的损失之间保持平衡。决斗者具有各种各样的攻击手段,为了执行效果,每项攻击手段都必须精心制作。

当游戏中相互对抗的两个角色身体尺寸相差悬殊时(比如,想一下《星战前传II》——*Star Wars Episode II*中尤达与杜酷伯爵之战,或者一个侏儒与一个巨人之间的战斗),需要特别的机警才能攻击到对方角色的身体,这时的攫取、投掷和武器的使用、以及特殊情况下的动作都必须认真仔细地设计。比如说,一个侏儒猛击或抓取一个巨人的胫骨。巨人齐胸高度的一个抓击对一个侏儒来说可能毫发无损。

在程序编制方面,撞击块(*collision boxes*)或撞击区域作为你工具集的一部分,为了恰当地触发某类事件,你需要调整它的大小。撞击块决定出两个对抗者或角色在什么地方会发生冲突(被攻击判定)。这种情况的发生将触发一个事件。在其他的应用中,这些事件会触发对方身体上动作反应,以进行拳击、脚踢、抓取和武器的使用,或者显示像血液喷溅粒子、汗水的溅洒或者牙齿碎片这样的事件,哎唷!

格斗类游戏已经开始应用更多的道具。格斗中的人物能够抓取场景中的物品(比如条形子)并将它们当作武器。这种类型的动作需要你可能在其他的第三人称动作游戏中使用过的同类脚本。

由于比赛要在一个环境中不停地换地方,这种多样化的格斗环境设置就需要更多的精心设计的战斗关卡。随着游戏类型的增多,视角相对固定的游戏现在看起来好像已经过时了。

扩展格斗模式允许在比赛中由玩家控制格斗者和计算机驱动的格斗者配对公开赛,

而且它还将单人动作要素引入到游戏之中,而不仅仅是一对一的单挑设置或者是双人车轮战分组。尽管决斗类游戏在今后的发展之中还有很多需要改进的地方,它仍然是一个十分受欢迎的游戏类型。决斗类游戏将来还有更加广阔的发展空间。



5.3 解谜类游戏

有时候让人感觉解谜类游戏或者说“智力测验游戏”好像已经被人遗忘了。你看到过多少次为新的解谜类游戏打出的巨幅广告?确实有过这样的事,但是很少见,而且所谓的“巨幅广告”的“巨”还要看你的定义。《俄罗斯方块》——*Tetris*是一个具有里程碑意义的游戏,并且现在是流行文化史上的一个亮点。对于那些所谓的“偶尔玩一次的”玩家,最为流行的游戏就是典型的解谜类游戏或是与解谜相关的游戏。

此时此刻,世界上的很多人都正在玩并且享受解谜游戏之乐。你也可能受此影响来创作或者是重新构思一个解谜游戏。从年龄上说,该类游戏的玩家可以是小孩、小青年、三十来岁或更大的人。你所开发的解谜游戏本意可能并不是针对某一年龄段的,但是总会在某种程度上有这方面的倾向。

玩解谜类游戏很容易上瘾,这已经不再是什么秘密了,因此这类游戏的设计都很成功。解谜类游戏中需要考虑的一些特性包括:

- ▶ 对美工资源的低要求;
- ▶ 游戏机理的简单假象;
- ▶ 集中精力地试验和修改;
- ▶ 吸取组合行为和危机行为。

我喜欢解谜类游戏的原因之一是因为它们对美工资源要求“很低”。这并不是说解谜类游戏就不应该使用复杂的美工资源。简单来说,与很多其他类型的游戏相比,美工资源的细目清单通常要“短”得多。图 5-3 举例说明了一个简单的但很过瘾的解谜游戏,它是为移动电话编制的。

换句话说,你需要很少的美工资源就能制作一个简单的解谜游戏。这对小的开发者和那些刚投身到游戏开发中的人来说,简直就是福音。一两个人就可以开发出一个世界级的解谜游戏。而要从头制作一个世界级的第三人称的动作类游戏的话(在千禧年结束之前完成),一两个人简直是天方夜谭。解谜游戏通常需要谜题组件,比如各种类型的方块、小器具,形状或者小物件等等。你一旦制作好这些东西,并使它们能在代码中正

常工作，开发新游戏就变得非常简单，而且完成这些工作不需要复杂的“有血有肉的”角色设计或者高级的三维建模技术。

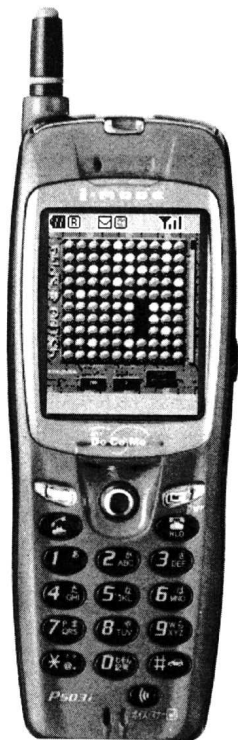


图 5-3 移动电话上的解谜游戏

大多数解谜类游戏都取决于一个根本的游戏机制——一个游戏的“脊梁”、一个令人心动的东西！考虑一下游戏《俄罗斯方块》中的纯朴之美：各种形状和色彩。我们所接受的挑战就是以一种最优化的方法对这些形状重新排列以实现某个目标，这就好像是一个小孩子在玩积木。正是我们这种孩提时代的天真无邪，孕育了这些解谜类游戏的思想。时间的流逝加重了这种怀旧的思想并且使那些想法得以实现。为什么我喜欢在沙滩上挖坑，而且一挖就是数个小时呢？为什么喜欢观看蚂蚁在一个塑料箱子里边钻洞呢？

解谜类游戏的机理可能会给我们一个简单的假象，并且我们经常都在冒险妄自揣测自己的想法。游戏设计者 1：“那太愚蠢了，我们不能仅仅弹出一些老套的东西；我们应该给玩家留下发挥自己思维能力的空间，让他们可以改进这些弹出角色。人们都喜欢

产生一些新的东西，而不是仅仅是这些弹出的“老”东西。”确实如此。游戏设计者2：“如果他们想要发展这些弹出角色，那我们就得开发出一个平衡系统或者权重系统，以此来确定他们发展或进步过程的弹出等级。”明白了我的意思了吧。

一些十分成功的游戏是依赖于匹配和弹出的，这其中没有任何神秘的东西。它们只是通过匹配和弹出一些东西来满足我们的愿望。我不知道为什么我们会对这些匹配和弹出的东西有一种基本的需要。我自己也喜欢将泡状小包歪曲来产生“机关枪”的效果。同样，我也喜欢运载火箭。是不是有这些火箭发射场一样的荒诞的想法就得承认自己玩过像《泡泡龙方块》——*Puzzle Bobble*和《泡泡龙》——*Bubble Bobble*这样的解谜游戏呢？我想我是会的。

开发一个解谜游戏通常需要一些集中精力的试验和修改。你可以从能抓住你的游戏中的一个基本点之的某一个特定想法开始试验。按照你认为可行的方法确定游戏的细节，并试着往下开发。代码开发中可能会有很多不好处理的地方。实际中，玩家可能喜欢你的游戏中“出人预料”的一面，而它根本就不是你先前设想的游戏要点。那也不错。继续开发，进行实验，多听意见，并注意一下它向哪个方向迅速发展。正是通过这种修改过程你才能不断改进你的游戏。

正如我所说的，对于你的解谜游戏试验中的核心部分，你必须乐于吸取那些组合行为和被我称之为“危机”的行为（那些使人上瘾的，反反复复的，令人满意的动作，比如弹出、挤碎、砸烂、打破、匹配和混合）。将这些组合行为和危机行为提供给玩家，并为他们设置一些挑战。不停地进行试验直到这个游戏确实能发挥作用。这时它就成功了！



5.4 即时战略游戏

制作即时战略游戏是一件非常复杂的工作。从设计的立场出发，在制定即时战略游戏的开发计划时，你需要作出很多决策，不仅如此，在游戏的开发过程中你将需要完成更多的决策。很好的参考例子如游戏《星际争霸》——*Starcraft*和《帝国时代》——*Age of Empires*，它们都是著名的即时战略游戏。

大多数的即时战略游戏都提供大量有效的运载工具、人物、和 / 或其他生物供玩家控制。玩家可以以数种方式指挥几大队人马或者个体单位努力作战。通常情况是，这些大队人马有目的地完成各种任务，玩家使用自己的策略去解决或者战胜一个给定的目标

或通过游戏中的一关，或者是击败另一个玩家。我们看一下即时战略风格游戏的设计中的首要难题：

- 需要极高的美工设计和创造性；
- 作战单位的定义；
- 作战单位的行为；
- 界面灾难；
- 路径搜索；
- 计算机对手；
- 简单的或者复杂的策略。

即时战略类游戏开发是极端的资产密集型工作，它们需要大量定制的美工、代码和工具的支持。用所需的大量素材来开发一个具有竞争性的即时战略类游戏需要一个相当大的团队相互合作。

这些由玩家控制的大量的人物、生物、交通工具和机动武器都需要精确地定义。对于每一个作战单位，不管多么的弱小或者多么强大，它们将掌握有什么样的技能，运用什么样的武器，相互之间存在什么联系，以及它们有什么样的动机、力量、弱点和倾向？对于设计者，每一个特定单位的这些细节都应该集中在一起放到数据表中。

下面是一个相当简单而通用的角色单位的细节列表，其中每一项的取值上限是 30：

力量	20 - 30
速度	5 - 18
智能	5 - 15
运气	10 - 20
耐力	20 - 30
心态	20 - 30
生命值	20
灵敏度	5 - 20
视野	8

可以看到表中定义了一个角色的常用特性，你可以根据需要改变（变大或变小）它的取值。对于一个即时战略游戏，你需要通过表格的形式非常详细地列出各种人物、交通工具以及攻守方面的能力和依存关系，资源系统的依存关系（举例来说，金子、水、

油、气、木头、香料), 普通武器的能力, 研发系统, 贸易系统, 政治关系等等。

你可能需要一张类似的表格来定义交通单位——比如说是一部装甲运兵车。你需要这个运兵车的概念产品以及对它的描述。它可能有三种风格: 标准的, 超级的和巨大的。它可能既有炮弹发射武器, 又有辐射武器; 它应该具有一定点数的生命值(能够承受损伤的程度)和一定的装甲等级; 它也有一定的运兵能力。每一个与众不同的单位都需要进行定义。一个标准的运载工具的定义和一个巨大的运载工具的定义看起来应该有所不同。

一个单位定义好后, 你就应该描述该单位的行为了。好的, 我们已经知道了关于这些单位的一些情况——那, 它们该怎样运作的呢? 发现敌人后他们会自动开火吗? 他们能够发现一些东西吗? 他们在寻找某个东西吗? 他们默认的运动模式是什么? 玩家能够改变这些模式吗? 怎样改变? 这些东西是怎样影响代码编制与接口的呢?

仍旧以我们的装甲运兵车为例, 我们需要定义这个单位的机动性(比如, 能从点 A 开到点 B; 能够安置到成在一个方形和圆形边界上进行巡逻, 等等)。而且, 我们的运兵车还具有炮弹发射和辐射攻击。它应该怎样工作呢? 是不是它只能沿直线发射? 它能不能热辐射制导和沿曲线飞行? 它的射程有多远(它的射程可以覆盖多少个单位栅格)? 这种炮弹发射武器能不能增强(升级)? 怎样增强? 对于辐射攻击我们也需要回答类似的问题。它是一个环形或“冲击波”式的攻击吗? 它看起来是什么样的? 它能够延伸多远或者说能够覆盖多少个单位网格? 它能够增强吗? 怎样增强? 它们武器的弹药可以再装填吗? 如果可以, 怎样装填? 这些单位必须返回再补给站吗? 补给站能不能飞行到该单位处进行装填弹药? 这个“飞行”动作是自动的还是由玩家指挥的? 现在, 我们需要进行一个相似的过程来定义和描述我们的补给站。

这些只是在开始定义单位的行为时需要回答的很多问题中的少数的几个。

在一个即时战略游戏中, 玩家会面临大量信息。我方各作战单位的状态如何? 敌方各作战单位的状态如何呢? 其中某一个作战单位(比如正在交火的那个)的状态又是怎样的呢? 同样的, 在游戏的后方, 玩家还要能够指挥大量的信息。继续构造这个单位, 尝试并保护那一个单位, 收获这些资源, 在这个地区巡逻。贸易路线是否畅通? 我们的捕鱼船沉了吗? 你应该知道这些情况。这些信息是怎样在玩家和游戏之间来回传递的? 就是通过游戏的界面来传递的。

游戏界面就是游戏中的操作外壳。构造一个外形整洁、易于理解且功能强大的界面并不是一件容易的事儿。随便找一个即时战略游戏的结构看看, 你会发现某些“界面障

碍”几乎无处不在——对于其他游戏类型也是如此，但对一个即时战略游戏尤其如此。多重菜单系统导致玩家必须层层进入才能访问到菜单的内部功能，这让玩家感到太慢并且厌烦。当界面设计得太复杂，你就会失去玩家。如果你为了追求界面的复杂性或多样化的特征而牺牲了界面的易于理解性和导航性，那么你就是自己在赶走用户。

我尽力地密切关注游戏界面，我更加注意研究那些没有使我感到失望的游戏。我会对一个界面中我认为处理较好的部分作记录，以及我认为它处理得好的原因。是布局上的因素吗？是不是它的自定义特征让我感觉到更加容易控制？

单位的行为必须能和游戏中地形和环境设置进行交互。在各种形式下，不管是对于个体还是群体，各单位必须依照地形活动。它们必须围绕着地形上的各种物体（比如建筑物、河流和其他单位）寻找道路。用来控制它们从点 A 运动到点 B 的代码称为路径搜索代码。程序员可以用来构造控制代码的算法有好几种，控制代码描述了单位应怎样运动，当和障碍物冲突时应该怎么办。你是怎样将一个很大的队伍通过一条狭窄的通道的？各单位挤在一块，相互遮挡、凑成一团，就像蚂蚁都争着进入蚁穴时的瓶颈作用一样。

在构造一个 CPU “选手”来挑战玩家时，你的脚本和游戏代码将必须平衡在人类玩家的“平均水平”（对于所有的开发者来说，很难精确地确定出一类游戏的中等或者平均游戏技巧水平是怎样的）。

举例来说，人类玩家可能会作出更好的策略转移，但是构造大量的战斗力量时相对较慢。而 CPU 通过以一个最佳速度不停地构造新的士兵可很快地构造战斗力量，但是可能作出一些策略较差的转移。你必须在所有这些东西之间作出折中，同时还要牢记住游戏体验。要是 CPU 很快并且毫不费力地构造完成一支庞大的部队，部队中都是那些游戏中可能拥有的最高级的士兵、运输船和武器，并且很快就干掉了人类玩家（如果赏给你一个机会再玩儿一次的话，还是重复同样一个过程），这将让你的玩家感到挫折并饱受羞辱。这正好与你打算传达给玩家的游戏体验背道而驰。虽然如此，很多游戏还在继续范这个低级的错误。

由于我们所构造的游戏其核心是策略的运用，那么我们要求玩家能够想到多少种策略呢？当构造战略情节时，我们既可以使用简单的也可以使用复杂的策略。简单的策略可能意味着做一些事来还击你的敌人。但是这还不够。这能给玩家一个令人兴奋的，使人沉浸其中的游戏体验吗？

需要玩家运用复杂的策略，这样玩家在它们自身所在的当前位置（就像在下象棋时

的那种策略和聚精会神)就必须考虑好下面几步的行动,这个游戏会有一些的难度。这会缩小这个游戏的玩家的数量。大部分的游戏只是围绕简单的策略构造游戏情节,这种策略要素通常以这样的形式出现的:“以某种方式使用这些武器或者这些工具来攻克一个普通类型的障碍或者是抵御计算机的部分攻击模式。”换句话说就是以某种策略来利用这些“事物”。

一旦你有了这些操作基础,当你改善单位的行为和调整或者说调节游戏的难度曲线时,即时战略游戏需要大量的发布前调试工作。如果你想要你的即时战略游戏尽可能地赢得最大数量的玩家时,游戏的难度曲线是一个十分重要的东西。你不能只是简单地利用一个即时战略游戏向导来调节游戏的难度,如果这样做了,你将会损失很多潜在的玩家。



5.5 角色扮演类游戏

和我们已讨论的其他类型的游戏一样,角色扮演类游戏(RPG)已真正进入一个振奋人心的新时代,并且每天都在赢得新的爱好者。现如今,角色扮演类游戏和其他类型游戏之间有很多东西都是共通的。角色扮演类游戏允许玩家“变成”游戏中的一个角色,接受该角色,并以该角色的身份在某个环境中完成任务和生存下去。流行的角色扮演类游戏包括《无尽的任务》——*Everquest*,《无冬之夜》——*Neverwinter Nights*,《末日危城》——*Dungeon Siege*。

我们来看一下与角色扮演类游戏相关的一些设计因素:

- 强调角色的制作和角色之间的关系;
- 资产密集型;
- 角色系统矩阵的开发;
- 角色与动作的融合;
- 女性游戏玩家 / 强调社会交际;
- 故事密集型。

在角色扮演类游戏中,玩家面临的挑战是为他们的角色构造很多物质的、精神的、迷人的、联盟导向的特征和喜好,接着还要构造它与其他角色之间的关系。一个很自然的设计要点就是怎样构造角色,还有构造一个角色在角色世界中意味着什么。

角色扮演类游戏的开发重点应该放在美术资源的制作上,因为需要多种角色类型、

各种特征、独特的能力、服装、设置、道具、物品和特定的界面才能为角色构造一个有意义的世界，在这个世界中各角色可以进行活动并互相交流。

构造一个角色系统矩阵有助于你构造一些具有“相互补充”的能力和技能的不同角色。不同的设计者可能会用几种不同的方法来完成这个矩阵。由于个人的爱好不同，很多玩家有自己最喜欢的角色类型，尽管如此，你还是要激励玩家试着构造不同的角色类型。

举例来说，你可能有近距离格斗的战士，并鼓励他们和有远距离攻击的魔法战士联合在一起。你可能有躲在战壕里的炸弹兵，他们进攻的时候需要弓箭手的保护。简而言之，你需要努力开发范围广大的各种各样的角色能力，它们彼此之间有着某种轻微的依赖关系。这可能是一个艰巨而复杂的任务。

很多角色扮演类游戏很成功地将角色构造与动作要素以一种令人振奋的方法融合在一起。以基本的角色构造系统和角色类型为基础，随着角色数目的增加，角色面临着各种各样的动作方面的挑战也越来越艰巨。这也和玩家的爱好基本上是一致的。有些玩家能在对抗中通过选择来简单地“增长”各种特征而感到满足，这和纸版游戏的起源以及大家所喜欢的游戏《龙与地下城》——*Dungeons and Dragons*有些类似。另一些玩家更喜欢通过直接参与到角色的活动中来使角色“成长”。此时最重要的是动作要素本身。

对于女性玩家，她们不会对杀气冲天的竞技场产生兴趣；通常她们喜欢玩那些侧重于构造、培养和交流方面的角色扮演类游戏。从内容角度来讲，我们都知道游戏市场并不能很好地为女性玩家提供服务。“编织”个人故事的角色扮演与角色成长，可能会成为将更多的女性玩家带入到游戏世界中的一座桥梁。游戏开发者们需要寻找更多的方法吸引女性玩家，这对于游戏产业的成长来说是至关重要的。

很多角色扮演类游戏是故事密集型的。为了让玩家角色体验它的游戏之路，这类游戏需要引入大量精美的故事细节。“故事”本身常常就是游戏开发者们进行重大争论的一个主题。有几个成功的角色扮演类游戏构造了漂亮的故事情节并将故事中的各要素发挥到了极致。像下面所给出的一些问题就是开发者所讨论的一些常见话题：怎样才能在游戏中将故事情节发挥出最好效果？在一个曲折的故事体验中，故事情节是不是非常地重要？在本章的末尾我们将更进一步地讨论这个问题。



5.6 第一人称和第三人称动作游戏

第一人称和第三人称动作游戏一直都很流行。大量这种类型的游戏都是以为数不多的非常大众化的东西为基础的，这让很多玩家感到失望。在很多杰出的游戏中，最为出色的第一人称游戏包括《雷神》——*Quake* 系列的游戏和《虚幻 2003》——*Unreal Tournament 2003*。我个人爱好的第三人称动作游戏是《异教徒 II》——*Heretic2*。

下面只是作为一个复习：“第一人称”是指从你的角色自己的眼睛来观看游戏，而“第三人称”是指就好像是一个可以看到角色的整个身体虚构的视野（比如，从一个跟着角色移动的普通“联动”摄像机的视野）。

下面是对第一人称和第三人称游戏设计因素的一个总结：

- ▶ 建模与材质密集型；
- ▶ 对关卡细节要求的增加；
- ▶ 需要难以对付的电脑人和脚本细化；
- ▶ 执行效率密集型。

大多数的第一人称和第三人称游戏中都需要在场景中安排数量众多的角色和道具，这导致整个游戏的开发需要大量的建模工作。跟据游戏内在模型所需要的复杂程度（所有优化过的东西转化成中低阶的多边形版本）和你的游戏引擎对关卡实现细节的处理，这些游戏的制作中需要完成数量相当巨大的模型自定义工作。它们还需要制作大量的材质来实现各种让人耳目一新的角色和环境。关卡细节（Level of detail, LOD）的实施是一个显示的过程：当镜头接近一个物体时要求显示这个角色的高分辨率模型，而当离摄像机有一定远的距离时要求应用低分辨率的显式版本。

在当前的一些游戏中，动画材质和多重材质表面有助于构造出那种像是一池荡漾的水面反射的光线一样的效果。你还可能看到过一些块状贴图的表面处理，它会使一个表面区域产生凹凸感（比如，腐蚀的墙壁或者是不平坦的砖墙表面）。

关卡细节设计的需求还在稳步地提高，更加强调对复杂光线和动画光线的使用，并出现了更为复杂的道具。很多这种作为“路标”的道具也需要定制粒子效果的要素。比如，在埃及城市这一关卡中，用一个太阳神雕像做路标通常情况下就已经足够了。现在太阳神雕像需要以一定的时间间隔从它的眼睛中闪烁出电光。还有可能，在电光效果正确工作并看起来效果神奇之前，你将必须花一些时间对电光效果的外观和行为进行

测试。

电脑人，即由计算机程序控制的“玩家”，近年来它们的“能力”也有了长足的进步。这归功于部分开发者在构造行为脚本系统时所付出的艰苦工作，以及后边他们利用这些系统试着“复制”那些高水平的第一人称射击手玩家的一些较高级的实战风格。虽然并不是很完美，但结果却给人留下深刻的印象。

在这些游戏中，能够帮助构造游戏可玩性的很多操作都依赖于触发器和脚本。通常情况下为了获得你某种希望的道具行为或动作要素需要花费很长的时间和付出大量的劳动。这会影响到设计，因为每个人都希望使足干劲，完成更多的内容并做一些前人没有做过的事情。如果你在任何一件道具和动作行为上花费了太多的时间，你就是在冒没有足够的时间来完成这一关卡后面的或者游戏后面的其他事情的危险。

这种类型的游戏特别的一点是执行效率密集型。通常情况下在玩家一方对硬件要求特别严格。这种类型的游戏以对机器的要求“位于前列”而著称，这些要求包括图形显示卡、内存和CPU速度。这些硬件需求能够让很多游戏的效果大增，但是从商业角度考虑，你得和你的发行商讨论讨论——一个普通玩家也需要这些最新的“各种硬件”，这可能是一种十分困难的销售境地。



5.7 模拟类游戏

很多流行的面向模拟的游戏都让玩家体验某种成长过程，你可能在经营、管理一个动物园、一个娱乐公园、一座未来主义的城市或者是一个创记录的商业联盟。一些开发人员甚至根本就没有将这种类型的东西当作是游戏，而是把它们视为一些与数字有关的东西或者是一个基于软件的玩具。显而易见，它们可以让一个玩家对身边的各种经历有足够的空间进行体验。

不管怎么说，这种类型的游戏一直都拥有大量的玩家需求。到目前为止，大多数成功的游戏好像都集中在一个成长过程的成长和成形上。玩家的决策会影响并表现出正面或者负面的结果。将赋税或票价抬得过高，就要经受它的负面影响。这些游戏教给玩家在它们进行交易时要追求一定程度的平衡，并且去寻找最“边沿”的利益或在没有“过载”的情况下最大化一个过程的方法。有些人已将它们作为一种娱乐性的教学工具。

现在我们来看一下设计时要考虑的几个因素：

- ▶ 单位密集型；

- 开放式或者白手起家;
- 易于扩展;
- 正面措施的负面影响;
- 规则和定义的依赖性。

模拟类游戏具有单位密集型的特点。它们需要成百上千的(甚至是成千上万的)美术资源,这些资源构成了游戏中的各种物件(比如建筑物、道路、动物、马匹、快餐厅、人物和交通工具)。这些游戏对美术资源的需求特别得大,这就变成了设计与生产的平衡问题,因为你的时间和预算仅允许你得到数量有限的资源。为了实现你的游戏思想,你该构造哪些东西呢?

这里存在有一种合乎情理的趋势:过多的任务实施和过度地设计(用太多的同步方向扩展游戏的范围)。在设计方面,它与实时战略游戏类似,你需要详细说明所有的单位组件,包括它们的功能和行为。最为重要的是,你需要定义出它们在一起相互生活并改变的规则。

模拟类游戏对于玩家通常是从“白手起家”的状态开始的。从某种程度上讲,这是一个开放式的过程的开端,尽管游戏的结局要求玩具达到一定的目标(比如,某个销售记录,出席人数的记录或者是身体的大小)。这些实际上和数字化的建筑集有些相似,它们都具有跟踪玩家编辑的价值与决策点数的能力。对于那些喜欢看到他们自己的决策与动作的作用效果的人来说,这是模拟游戏很具有感染力的一面。

相比较而言,模拟类游戏具有易于扩充的优点。一旦你的美术资源规格多到足以构造新的“物品”(假设这些物品在行为方面不需要编制复杂的代码),很多新的单位就能够加到游戏的新增加的主题中,从而增加它的重新玩耍的价值或将用于特殊目的。

对于那些没有必要寻求那种特殊动作巨大的杀伤能力的人来说,这些游戏给玩家提供了一个更加被动和需要深刻思考的经历。从可调整的数值出发做各种变化并观察出现的结果,构造或玩一个模拟情节可能令人感到精神放松。

模拟类游戏设计的核心很大程度上依赖于定义并实现一套规则系统,这个系统将对给定的成长过程加以引导和管理。在最基本的层面上,这可以利用一个标准的数据库程序或在纸上进行建模。从“数字”的角度来看,这可能是一个简单的任务,也有可能是异常复杂的任务。对于那些敢于有所作为的“对数字感到兴奋的”开发者,试着去完成经济关系、各种行为、对事物的态度等类似的特别复杂的数据模型、并将这些计算结果加入到玩家的模拟中,这些事情对他们有着相当大的诱惑力。

从商业角度上讲,开发者通常会用光所有的时间并且以被迫将规则加入到代码而告终,这听起来类似于“当赋税高于服务时无法令人满意的结果所采取的措施”或者“当入场费远远超过预料的价值时无法令人满意的结果所采取的措施”。我知道这是很平常的事儿,但这些只是形成这些引导和管理基于模拟的游戏体验的规则的起点。

基本上,这变成了一连串数据表格的比较。那些对你的综合性建筑进行参观的人是不是发觉没有足够的盥洗室和令人满意的交通工具?这变成了一个表格的比较,我想在我们能够看到其背后的观点,这正是你开始构造你的规则集的途径。这些规则分成等级并且具有优先级别。你在比较时,应该首先验证的“驱动”(driving)规则,然后才是那些驱动规则外围的或补充的规则。



5.8 构造电影动画

游戏中的电影动画是一些用来增强和帮助说明玩家经历的简单电影片断。游戏中电影动画的处理通常从两种方法中选择一种:一是将一个完全渲染好的电影动画片断压缩成 MPEG 格式,并使用一个集成到游戏代码中的多媒体数字信号的编解码器(编码器/解码器)进行播放,或者是由游戏引擎本身负责电影动画的渲染和播放。

对于第一种情况,将一个视频编解码器,如 DivX (www.divx.com) 或 Bink (www.radgametools.com),内置到游戏中来播放 MPEG 格式电影的片断。这些电影或者动画的要素是使用带有插件的三维商业软件包专门为游戏构造的,并且由于它们都进行过预渲染处理(相对于实时渲染),它们能够充分利用软件包中较高分辨率的模型、材质和令人眼花缭乱的特效。《魔兽争霸 III》——*Warcraft III* 中有一些引人入胜的优美的电影动画,可以说令人瞠目结舌,它们就是利用视频编解码器处理的。

在第二种情况下,一个开发者可能会选择利用游戏引擎本身来创作电影动画。像角色模型和已经存在或定制环境要用到的游戏资源,都能够在游戏的电影动画中直接应用。它们只是简单地利用绘制游戏本身的游戏引擎来制作电影动画。定制脚本通常用来“播放”电影动画并控制角色和环境细节。

不管是哪一种情况,你的电影动画(包含在该大类中的有开头介绍性的片断、变迁或过渡片断、特定目的或插入的片断以及结尾片断)都是用来叙述游戏背后的故事情节要素的,并通过占据玩家的视觉和心灵将他们从感情上融入到游戏情节中去。

电影动画是通过将分镜头细节(或者分镜头内容)分为多个场景并以情节串连图板

的形式串起来。情节串连图板技术在将书面上的或者口头上的概念思想转化为一个对玩家有意义的可视的故事逻辑中有着很重要的补充作用。制作复杂的电影动画需要相当多的资源，你应该不会对此感到吃惊。在对游戏完全理解而感到兴奋的冲动下，电影动画是开发者很容易做得过了头的另外一个地方。在你认识到这一点之前，你不但是在构造一个很复杂的游戏，同时你还在渲染完整的电影片断。就像游戏设计中众多其他的领域一样，所有的东西都应该保持在一个平衡位置。



5.9 开发背景故事

有关故事在游戏制作中的作用和地位，游戏开发人员之间已经进行了大量的讨论。如果你要对游戏本身找出什么证据，那么一些类型的游戏好像几乎就不需要什么故事。在他或她所喜欢的射击游戏、模拟游戏或者解谜游戏中，他或她真的被其中的“故事”吸引住了吗？我对此表示怀疑。

另一方面，其他类型的游戏，如角色扮演类游戏或即使战略类游戏以及很多其他的动作/冒险类游戏，它们都将故事叙述元素的作用发挥到了极致。但是我并不是真的想要根据游戏类型的不同对故事划分各种界线（换句话说，我并不是说有些类型的游戏在游戏中包含有很多故事情节而其他类型的却没有）。将一个感人肺腑的故事要素放到一个英雄射手的游戏中，这个有没有可能呢？我不会说这决无可能。

我对游戏和故事的需求是非常注重实际效果的。以我的经历，我觉得游戏设计者应该很有目的地去学习讲故事的技术与故事的编排，这是十分有益的。最少，他们应当理解那些传统的生动的故事连载以及它们是怎样应用在实际当中的：调用动作、冲突和分辨率。如果你对故事连载并不熟悉，也没有关系，而且很幸运的是，在这方面有许多很好的游戏可供参考。

在游戏中，讲故事和背景故事应该是用来支持和扩展游戏可玩性体验的，并不是用来取代电影产业。不管在什么地方，只要这个故事能够增加玩家的娱乐程度并使玩家在游戏中感到自由，我说就应该使用它！仅从这一点上来说，那些写在众所周知的书中的每一个技巧我们都应该加以利用。

从定义和侧重点上来讲，游戏和电影在很多方面有所不同。正像我所知道的那样，现如今在游戏和电影之间有各种各样相互启迪的灵感存在。但是毕竟游戏设计者并不是剧作家，而剧作家也并不是游戏设计者。

对于剧作家来说,构造出丰富的故事情节、能够让人接受的故事逻辑和复杂的故事角色也并不是一件轻松的事。与此类似,开发一个具有创新性、可玩性好、令玩家高度沉迷的并且长久的游戏机制对游戏设计者来说也是不容易的。因此,在游戏中讲述故事和背景故事这一点上,游戏设计者需要能够提供充实的并且具有娱乐性的故事来为游戏增光添色。他们并不需要和柯恩兄弟或者是斯蒂芬斯皮尔伯格来竞标下一项工作。



5.10 设计游戏对话

设计游戏对话是一项技巧性强、琢磨不透的工作。最主要的是,只有当你的故事细节确定下来后,你才会去构造对话。如果角色们还不知道故事中讲述的是什么内容,他们怎么能知道在故事叙述中应该说些什么呢?请记住,你的故事连载和故事细节应当支持你的游戏体验,而不是和你的游戏体验抢风头。因此,考虑一下什么样的故事将会为你的游戏体验添加广度,而这些游戏体验可能会影响到你的故事细节。

我们假设你已经设法确定好了故事的细节。在许多开发团队中,因为人们对故事思路的反应千差万别,所以这本身就是一个了不起的成就。将来自一个团队的很多想法结合在一起,对它们进行合成并加以处理,然后作出对这些想法结论并付诸于行动,将它们作为一种游戏创造性的激励,这需要极强的个性。

你有可能是需要单独一个人写出几种类型的对话,或者是和签约的职业作家一起完成,这取决于你们的预算和团队的风格。伴随着游戏介绍、情节的过渡、任务的完成、关卡的结束甚至整个游戏的结尾,角色将通过画外音进行讲述,并且有时作出指导性和有导向作用的评论。角色甚至可能教玩家如何玩游戏。

对于设计和应用对话,下面是一些更加实用的考虑因素:

- ▶ 当你要显示对话时(并且你希望当声音关闭了或者无法使用声音的时候游戏仍然可以继续玩),你将以怎样的形式把它呈现给玩家?
- ▶ 这些对话需要本地化或翻译给世界各地的观众吗?这在实践中将会对界面设计产生影响,因为对于每一种特定的语言,为了使它在计算机屏幕上清晰可辨,它的符号和字符都需要一个起码的像素尺寸的显示空间。
- ▶ 为了本地化的目的,所有的语言符号都应该是打印到屏幕上的(而不是嵌入和画在屏幕图形或界面元素中)。这在开发的早期就应该通过本地化的规划完成的,这样的话,在游戏代码中就可以切换你的各种语言的显示系统。

重要的是不要将所有的写作细节都堆在一边,这样到工程的最后会堆积如山(就像滚雪球一样)。事实上,你确实不能这样做,这是因为,在游戏的整个开发过程中,由于需要评估对话的效果并帮助构造游戏的细节,经常会需要一些起占位作用或“最终要完成的”对话。

和动作捕捉相似,规划录制对话的方案也需要精心的准备与计划。还有,你需要尽量避免大量的返工。也许你也不会有这样的机会(你可能已经被炒鱿鱼了)!你可能在制作游戏的整个过程中都要录制“声音”,当然这还要由几个产品因素来定。在其他情况下,你可能只需要短短几天或在一周时间内就能录制完游戏中90%的声音。

我们需要将这些限制加到这样一个事实中:现在的很多游戏都是围绕一些授权的资产构造的(像《终结者3》——*Terminator 3*和与《星际大战》——*Star Wars*有关的游戏)。那些来自游戏中露面的这些众所周知的道具的角色,可能需要那些为电影或电视配音的演员来配音。录制游戏对话时,来自这些演员和/或者工作室的报酬和工作量可能很高也可能很低。我还没有见过任何介于中间情况的。同样,对这些演员在时间上的要求显然也是很重要的。简言之,除非你已经做了相当多的规划,并且表现出一些非常老练的工程方面的深思远虑,否则你可能会正好错过一个绝好的机——找到一个演员,他表演的对话将会把你的游戏推向排行榜的首位。

考虑到一些特殊的情况:那些天演员们可能要为其他部门做其他类型的“具有奖励性”的录音工作(换句话说,你的游戏对话的录音工作本来有一个干劲十足的时间安排,却从某处被迫中断了),并且你可能只有30分钟或者是一个小时的时间和他们接触,尽管事实上你被告知你有两个小时的时间。规划是至关重要的。你必须经常当机立断地修改并评定录制的优先级。

正如前面所指出的,根据你的故事细节的不同,在要为游戏录制对话以前必须把这些对话写好。从内容上来说,你对游戏的里里外外知道得很清楚或者说比别人知道的更多。你知道在游戏中的什么时候、为什么以及什么地方需要角色说话。让那些疲惫不堪的(并且肯定脾气暴躁)演员作出与游戏相关的正确的表演是你的责任,即使这些演员可能不喜欢游戏,可能不知道为什么恰恰是由他们来做这件事,还有可能从来就没有玩过游戏。

其实,情况也并不总是这样。你也可能很幸运地找到一个活跃的演员,或者这个演员对于表述故事对话有一些特别的想法。在录音工作室中你应当经常带一个游戏的简单模型在身边,这样的话便于展示你所要求的故事情节和对话的“上下文”。你的身边应

该尽量地多带一些你找到的或者用得着的形象化的参考资料。你将会参照到其中的很多东西。

具备这些条件后，你将会很快地学会怎样引导一个演员他进行表演。有时候你将从音效师或者是从与演员一起工作或在工作室中的其他人员那里得到一些帮助。而有的时候你可能得不到任何的帮助。我发现来自演员中介机构的配音演员是很棒的。他们会根据你所提供给他们有关游戏中要表述的故事线索的角色与上下文环境的所有信息，尽自己最大的努力帮助你完成配音。



5.11 设计人员的工作工具总结

在下面的总结列表中，我对到目前为止我们讨论过的一些关于重要的游戏设计软件工具的建议做了整理。这些建议可以应用到很多类型的游戏中。请你记住，纵观整个游戏产业，你会发现不同的开发者对这些工具的使用方式差别很大，但是不管是这种方式还是那种方式，他们基本上都在使用：

- 三维软件包（比如像玛雅，3ds max, Lightwave, Mirai）；
- 微软的 Visio（用来画流程图、屏幕逻辑图和地形图）；
- Photoshop（用于基本的材质处理或者用作各种其他的用途）；
- HTML 编辑器（用于制作设计文档和产品文档）；
- BSP 编辑器（单独开发的编辑器，如 Unreal 和 Radiant 系列）；
- 脚本语言（Python、Visual Basic、自主知识产权的脚本语言）。



5.12 设计转型的案例研究评论

既然本章中我们已经研究了一些设计上的要素，看来到了该讨论一些有关“设计转型”过程的东西的时候了。设计转型指的是针对关键的设计改变（甚至是游戏类型选择上的彻底改变）而要求游戏开发者作出的一些转变，这些改变通常是依照最近一些成功游戏的热销或者只是对流行趋势的追求而定的。

现在最通常的情况是，当开发商决定投资开发一个游戏时，他们心目中都是认可了这个游戏。开发商的许可意味着它能够很快地得到观众的认可。我们都听说过游戏 *SpongeBob Squarepants*，其背后的思想是很简单的。如果人们已经喜爱 *SpongeBob*，他

们很可能会购买一个基于 SpongeBob 开发的游戏。当然了，我们也知道事情并非总是如此。

如果人们对我们“伟大的”新游戏一无所知，并且我们也不可能花那种钱来通过十分昂贵的广告让公众知道有关我们伟大的新游戏的情况，如果是这样的话，好啦，我们还是开发更多的有关 *Spongebob* 的游戏吧。

考虑一下现在出现的那些原创游戏的数量以及那些已注册的游戏的数量，你就会明白我们该朝哪个方向发展了。然而，不管你正在开发一个原创游戏还是一个已注册过的游戏，你都有可能遇到设计转型。

在当前的游戏发布大环境下，毫无疑问将会有人要求你的开发工作室去完成一份儿相当全面的概念验证 (*proof of concept, POC*)，以向他们说明在给你一个特定的游戏开发许可 (比如 *Spongebob Squarepants*) 的情况下，你将会制作出什么样的游戏，或者对于原创游戏的情况，你要展现你的原创游戏是怎样地像成品游戏一样地工作。这当然需要大量昂贵的探索性工作。

一般情况下，开发团队通过概念验证把他们要开发何种类型的游戏的思想呈现出来。这项工作通常会经历很多次的反复。有些时候，可能需要对开发何种类型的游戏作出一个决策后，才能开始执行在开发者和发布商之间达成的开发协议。那接下来如何按允诺好的条款完成游戏制作就是开发者的职责了。

游戏开始开发后，还有一点需要注意：游戏开发商需要对游戏销售的趋势作出反应，以保持其市场竞争力。为了响应某一正在增长或是处于顶峰的游戏销售趋势，开始有着特定设计目标 (比如说是一个综合的动作游戏) 的游戏很容易就变成其他样子。作为一个开发人员，这可能需要全盘重新设计，而且还可能需要一套全新的技术方法。游戏开发的“沙丘”迁移得很快。

这个过程是怎样影响游戏开发者的呢？嗯，游戏开发者必须时刻为这些突发事件做好准备。这就是为什么需要我们在某种程度上熟悉、了解和热爱尽可能多的游戏类型。尽管你昨天可能还在开发一个角色扮演类的游戏，今天你又可能要开发一个第三人称动作游戏。然而，即使情节本身会不断地变化，而开发者们却变得越来越类型专一化了。造成这种情况的部分原因是游戏的广度和复杂程度的增加。一旦你学会了支持构造即时战略游戏的各项技术，致力于即时战略游戏的开发对你来说更容易并且更能发挥出你的水平。在一个充满激烈竞争的产业里你真的能够做到事事精通吗？恐怕很难。

各种游戏是经由开发者而进入其生命周期的，并且在整个开发过程中它经常会发生

彻底的转型。有时，经过两三次的转型，整个游戏就可能被完全放弃，这样的话，你花费了16个月完成的工作就会被打包放进仓库里了。

要在游戏开发中保持有利的竞争地位，你需要一种能对不断发生的变化快速作出反应并保持获取知识的能力。

AARON ODLAND 访谈录



Aaron Odland 基于许多在迪斯尼频道 (Disney Channel) 和迪斯尼在线 (Disney Online) 中注册的大家耳熟能详的资源开发了很多游戏和网上娱乐节目。他在美国西北大学获得了广播、电视、电影专业的学士学位，随后在 Jellyvision 公司就职，Jellyvision 是以芝加哥为题材的系列游戏《你不了解杰克》——*You Don't Know Jack* 的创作公司。现在，他正在从事他所喜欢的园艺建筑工作，而且他断言说这项工作并不像听起来那样和游戏开发没有任何的关系。

TM: 致力于你所喜欢玩的游戏的开发工作和开发面向广大网上玩家的游戏之间有什么根本性的区别呢？

AO: 我发现当我在个人计算机上玩游戏时，根据游戏的开发平台，我会对游戏中声音、图像、操作等方面的质量有某些期望。当我玩一个控制台游戏时或者是玩一个老式的街机游戏时，同样会产生各种期望。这些期望通常是不一样的。我发现在线游戏方面的问题是：我通常在个人计算机平台上玩游戏，这种情况下我会有那些高的产品价值期望，而目前的数据连接确实不允许大量的经由服务器的数据传输，基于网络的游戏难以达到这样的期望。因此，我们被送回到了使用8位的任天堂时代，而在同样的个人计算机上都可以玩转 *Unreal II* 游戏。

TM: 你是怎样定位你的游戏设计来面向一个主要由年轻玩家构成的广泛的观众群呢？

AO: 在你讨论游戏设计时，你实际上不能将所有的年轻玩家都集中到同一组中。学龄前儿童和十几岁的儿童有着很大的差别。我认为开发一个游戏时你需要专门针对一个特定的年龄段，为该年龄段内的玩家制作出最好的游戏作品来。

TM: 当你使用一些授权的东西开发时，你所面临的是一些什么样的挑战？

AO: 在某些程度上，我发现在较短的产品期限中使用授权的角色是十分有效的，

这是我曾经效力的大多数在线游戏的共同特点。你不必再在角色设计上花费时间，这当然有助于后面的进展。当然，和第三方开发者合作时，需要花费时间教他们有关的道具的知识，这样他们才能对游戏有更好的理解。我还发现使用那些涉及到真人演员的授权内容要比使用那些动画角色更具有挑战性。

游戏制作中图形审订工作是必不可少的，并且由于预算和时间上的限制，专门为一项网上内容制作独特的、全新的图形通常情况下看起来是不可能的。因此，很有挑战性的工作就是努力搜索一些替代资源，包括图形、音频剪辑等能满足你需要的一切资源。

TM: 你是怎样通过网络来抓住并壮大你发布游戏的观众群呢？

AO: 我认为每个人都在不断地争取得到别人对他们所取得的成就或者他们所创造的成功的事物的赞誉，而且在线社区的确给你提供了一个使用这些反馈信息改进你的游戏的机会。记分板上的高分当然能说明你的游戏很成功，但是我更感兴趣的是给玩家提供一些工具来让他们构造关卡、构造角色的材质贴图等等。因特网提供了很多共享的时间和精力，如果你能在你的产品开发中好好利用它，我想它对你的帮助是非常大的。

TM: 为较年轻的玩家制作游戏时你面临的限制是什么？

AO: 父母都不喜欢他们的孩子和陌生人说话，但是因特网是一个用于交流的很强大的工具。因此，你怎样才能让你的玩家相互交流而不用说太多的话呢？当你和十几岁的少年打交道时这个问题变得令人备感挫折，这些人不断地使用消息速递（*Instant Messenger*）和电子邮件来交流，但是在游戏中通常你不能让他们进行同样的交流。

从整体上说，这是一个关于游戏的问题，我认为这些为小孩子设计游戏的必不可少的“限制”中，大多数都应该应用到所有的设计中。你不应该只是因为游戏是给小孩玩的，就将键盘控制做得简单直观，担心他们可能玩这些复杂的东西可能会有问题；应该是这样的：你将它们做得简单直观是因为对所有玩家来说这样有助于提高游戏的可玩性。

TM: 构造面向网络的游戏时，你最喜欢的东西和最不喜欢的东西分别是什么？

AO: 构造面向网络的游戏时，我最喜欢的一件事就是这个游戏可以由单个有创造性的人来完成，而不需要一些大型的游戏开发商和游戏发行商的支持。我有一个这样的爱好，就是通过个人的力量纯手工地开发一些小型的在线游戏，并将它们免费提供给世界各地的人们玩。

由于开发周期短，所以在线游戏还允许你的工作涉及相当广泛的内容而不必承受那么多的时间或者预算方面的压力。开发周期短带来的不利的一面是你常常没有大量的时

间去投入到具有独创性的新型游戏的开发中，因此，我所看到的很多这样的游戏大多数都是衍生出来的。

TM: 你认为未来大规模的网络游戏会是什么样的？

AO: 我认为网络游戏仍然会作为光碟版游戏的一部分继续发展，就像其主流游戏《模拟人生在线》——*The Sims Online* 一样。至于由服务器数据传输的游戏，我认为它未来的趋势很难预测。

我不知道未来网上娱乐整体上的情况会是什么样的。虽然像梦幻联盟棒球队 (*fantasy league sports*) 这样的完全的在线订购服务已经获得成功，但我不知道会兴起什么样的商业模式可以为更多的游戏开辟一条出路。

TM: 你的广播、电视和电影方面的背景是如何影响你的工作方式的？

AO: 在我的制作课中有一句话经常向我敲警钟，就是在制作前你花费了的预制作时间将会在实际制作过程中为你节省更多的时间。因此，我牢牢记住这句话，并且确保完成流程图、使用案例研究和其他指导方针，这不但帮我想到了将来可能会失败的地方，还使我在游戏体验方面思考得更加深入，同时还能够使我就能就游戏中的一些创造性的和技术性的细节同工程师和美术师进行更加深入的交流。

TM: 对于那些热衷于进行各种类型游戏开发的朋友们，你有好的什么建议吗？

AO: 如果你想从事游戏开发这个产业，我会建议你花一些时间分析一下你在该领域内的兴趣，并且尝试着寻找一个朝这个方面的明确的奋斗目标。如果你选择了这条路的话，这种有目标的学习在你找工作时会有助于把自己推销出去。

TM: 哪三个设计特性是你重点考虑的内容评估的关键问题？

AO: 用户界面、游戏机制以及“回头玩”能力。我认为基本用户界面在游戏产品中经常面临着被忽略的危险。考虑周到的游戏用户界面设计、菜单设计等确实有助于增加玩家的游戏体验的深度，因为这些要素自始至终都在和用户进行交互。

当然，如果游戏玩起来没有乐趣，即使你有世界上最好的游戏界面也没有用。

而且，如果你的游戏没有能够吸引玩家重新再玩的东西话，那么它将不会有很长的生命周期。

TM: 近期内，游戏中的哪些方面会让你感到兴奋？

AO: 我想一直持续扩大的在线游戏社区会让我感到兴奋。尽管人工智能在不断地向前发展，在网上和世界不同地区的真人玩策略竞赛模拟游戏进行对抗会更刺激。

TM: 你有什么建议要给其他一些对游戏设计感兴趣的人吗？

AO: 找到游戏设计中你所感兴趣的具体细节, 并且学习它。研究你喜欢游戏和你不喜欢的, 分析一下这些特别的地方。

TM: 当前基于各种平台的游戏你中你对那些更感兴趣?

AO: 对于围绕着游戏《无冬之夜》——*Neverwinter Nights* 的在线版本发展起来的社区, 我感到非常兴奋。假如你是从庞大的《龙与地下城》——*Dungeons & Dragons* 爱好者基地开始的, 但是我认为 *Bioware* 公司(《无冬之夜》和《龙与地下城》的开发公司)所构造的工具非常地棒, 这些工具允许爱好者塑造自己的游戏体验并创造他们自己的世界。

TM: 你是怎样衡量一个游戏的机制是否起作用呢?

AO: 我想游戏机理的成功与否完全是一个主观的看法。你可以不断地改进任何东西, 并且随着最后期限的到来, 有时候你只需要问自己“这是不是已足够好了?”

TM: 目前你在游戏方面的目标是什么?

AO: 我近来正在从事园艺建筑工作, 这使我想起了很多怎样才能将它与构造游戏环境联系起来。我认为在设计领域和在建筑与城市规划领域, 我还有很多东西需要学习。

反过来, 很多这样的领域已开始使用游戏工具来增进他们在现实世界中空间与建筑方面的发展。这是一个有趣的联系。

TM: 你对游戏产业发展趋势的哪些方面感到烦恼?

AO: 游戏产业中开发商与发行商之间的联合让我感到烦恼, 尽管这确实不是游戏产业中特有的趋势。我对不断开发游戏续集的做法也感到沮丧。

现在好像很少支持开发独立的原创游戏, 我希望这种情况在未来会有所改变。

TM: 在你看来, 游戏社区的最终形式应该是什么样的?

AO: 我想关于这一点我头脑中没有任何印象。看到大群的人玩《欢乐森巴》——*Sambe de Amigo* 游戏和《劲舞革命》*Dance Dance Revolution* 游戏, 那是一件令人着迷和备感愉悦的事情, 我想这也有几分社团的意思, 同时也会吸引了很多不玩游戏的人。

我的一个朋友和他的《无尽的任务》——*Everquest* 游戏战队举行了一个虚拟的劳动节烧烤宴会, 并且这使他们在终极游戏社区组织名单上的排名大大提前了。

TM: 好的游戏到底应该达到什么要求?

AO: 好的游戏会讲述一个故事, 把你带入一个围绕该故事的世界中并且允许你参与构造这个故事和世界。

TM: 对于那些刚刚进入游戏产业的人你将会给他们三条什么样的建议?

AO: 玩很多的游戏，在各种廉价柜台上寻找便宜的游戏，拿回家玩一玩，同样也要玩那些你通常情况下能够得到的新游戏。所有这些都教给你一些东西。

要是有可能的话，你那就制作大量的游戏。

随身携带一个记事本，并且每天都要试着去绘画，即使你认为自己不会绘画。

游戏是一种视觉的设计媒体，这有助于你从表现你周围的真实世界开始，到表现你在头脑中构思的世界。

TM: 到目前为止，在你的游戏工作中你最美好的一段记忆是什么？

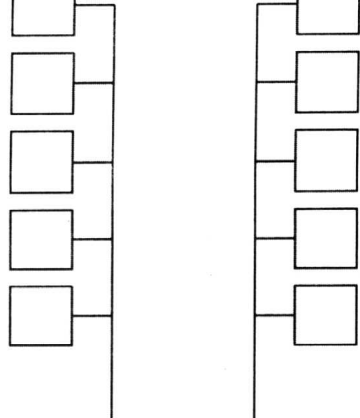
AO: 到目前为止，我在游戏工作中经历的最美好的一段记忆，是听到在早些时候开发控制台游戏时的一些令人吃惊的故事，就连一个 *Chuck E. Cheese* 店的维修工也能受雇为游戏开发工程师。在一天的网络游戏开发之后我也有一些有趣的回忆。

TM: 谢谢你参加我们的采访，Aaron。

MEGA 提示

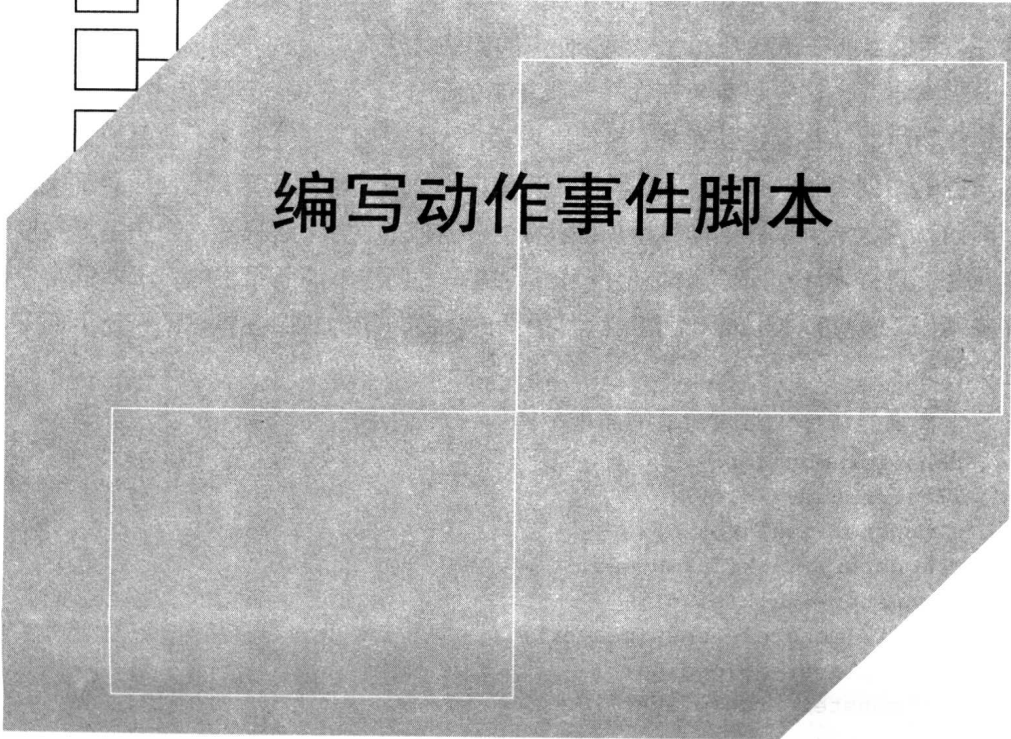
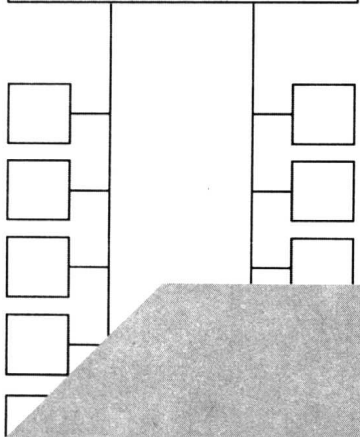
1. 集中精力深入地了解几个类型的游戏，同时继续完善你对所有主要类型的游戏的了解。
2. 玩一些你并不是特别喜欢的游戏。努力发掘出你为什么不喜欢这些游戏以及怎样做会使你更喜欢这些游戏。请记住，在玩游戏找乐子和受雇开发游戏，它们之间有很大的不同。
3. 至关重要：当你玩游戏时，要做一个设计记录和界面摘要，记录下各种注释、例子、图形、想法、问题和相互比较。
4. 一旦你找到了一些你喜欢的游戏类型，就研究研究那些和该类型有关的游戏。自己给自己提一些问题。对于运动类游戏，研究一下选手的动作和该运动项目的历史；确定出体育爱好者是怎样对不同的体育运动作出反应的。每一个风格的运动中最激动人心的一刻是什么时候？这些洞察能力是怎样融入到体育运动电子游戏中的呢？你是如何将运动的刺激融入到玩游戏中的呢？如果你喜欢角色扮演游戏，研究一下角色的构造，基于角色的社团以及角色的特征和细节。注意一下人们是怎样对角色的可视化构造作出不同反应的。你将会怎样对待这些反应？你能够预测到这些反应吗？要点是：无论是哪种类型的游戏，给你的设计构思注入了活力、增加的广度的通常就是这些相关的研究。

5. 充分利用游戏社区。访问一下你身边各种类型游戏的玩家，和你的所有朋友进行交流。试着找出为什么他们喜欢某种类型的游戏而你却可能不喜欢。仔细地聆听你所能听到的反馈。不要只是根据你自己对某类型游戏的爱好而忽略问题的答案。如果你认真听的话，你将会听到那些常见的严重问题。一旦你有机会的话，试着去修正你自己的游戏中出现的这些严重问题。
6. 走到游戏社区的外面，把“外边”新的设计观念带入到社区中。宽泛并概要地学习：考古学、世界历史、工业设计、图形绘制、布景设计、灯光设计、角色设计，工具设计、园艺设计、运输设计。从玩游戏的人群中走出去，并给他们带回一些新的东西。请记住，游戏设计需要你成为一个专业化的通才。



CHAPTER

6



编写动作事件脚本



游戏中的脚本给开发者提供了一种对游戏可玩性功能的许多关键方面直接加以控制的途经，这些方面中，有些很小，几乎察觉不到，而其他的则是相当大，如同上现蜃景。本书到了这里已经介绍了角色、道具和物品，你应该对构造一个多边形的游戏环境的过程（比如，可以利用像平面、球体、圆锥和圆柱这样的图形元素，而且还可以通过对这些元素进行“加和减”或合并运算以形成全新的图形）已经熟习了。现在，我们将把我们的注意力转移到学习一些有关如何控制这些脚本的知识，而所有这些脚本的目的都是为了构造出游戏的可玩性。

很多种类型的行为动作都可以由脚本系统来控制。总的来说，脚本系统大都是商品软件，而且当前在游戏开发者中间，他们的实施方法有很大的差别。很多游戏引擎都提供有脚本语言，这些语言允许你将一些复杂的动作行为编程到游戏中去。举例来说，一个怪物可能从谷仓走到农舍，再到储料垛，然后走回来再走过来，但是如果它看到玩家角色的话就会跑向玩家，如果玩家正流着血的话就会冲向玩家，而且如果玩家持有武器的话就会逃离玩家并走向储料垛。随着当今游戏的发展，这只是一个中等复杂度的动作行为，直接用 C++ 语言对这个行为进行编码将是很单调而冗长的。脚本语言使得为这些动作“编写脚本”成为可能，这样的话你就能将精力集中在具体的行为上而不是用在程序上。

下面便是这一个行为情节的简单示例：

```
behavior monster;
if (monster->idle) {
    monster->walk(barn);
    monster->walk(farmhouse);
    monster->walk(silo);
}elseif (monster->sees(player)) {
    if (player->bleeding) {
        monster->run(player);
```

```
    } elseif (player->has_weapon) {  
        monster->run(silo);  
    } else { monster->walk(player) }  
}
```

在没有计划好的脚本环境中，有时脚本系统只是简单地构造一些含有数值（速度 velocity = 44.6）的文件，你可以编辑这些数值来实现效果，然后，将这个文件读入到游戏引擎，并由游戏引擎控制的其中的行为内容。

在很多情况下，你所使用的脚本系统都能够通过接口连接你的游戏引擎。它可能有一个类似于一个我们熟习的标准的 Windows 程序的完整的图形用户界面（GUI），或者它可能有点像一个简单的允许我们对数值进行编辑、重新构造并对修改和编辑后的结果进行测试的文本编辑器。

在本章中，我们将仔细地看一下怎样利用脚本系统把一个多种多样的行为类型构造到游戏中去。我们还将考虑另一个强大的主要的脚本资源——触发器的使用！



6.1 脚本技术的选择

就像游戏开发中的很多其他的领域一样，现在开发人员在选择怎样处理游戏中的脚本时，也有好几种选择。在刚过去的一段时间里，游戏行为的脚本编制的方法一点也不具有通用性。对游戏可玩性的任何直接的改变，不管是什么类型的，都不得不由程序员直接将它编写代码到游戏中。脚本可以更有效地帮助游戏开发者对游戏行为（例如，某种武器的开火速度和作用范围，或者是运输工具和角色的移动属性，比如摩擦、重力和斜度）进行“连接”并加以试验。尽管如此，要想快速地构造出复杂的动作行为，我们还有一段很长的路要走。

现在，使用各种各样不断变化的硬件平台与设备的技术和工具来保持游戏引擎新鲜并且功能强大，这是程序员面临的一个相当大的持续性挑战。程序员最终决定了一个脚本系统将对游戏产生多大的控制作用和影响。一般情况下，游戏程序员与设计者通力合作，尽自己最大的努力为给定的脚本系统构造强大的编辑功能，他们的目标是构造一个强大而灵活的脚本系统，以使它能够产生出各种各样的游戏可玩性效果。这意味着在游戏开发过程中，你的游戏中的动作行为能够动态地对玩家的反馈作出反应并动态地聚焦问题。你能够很快地回答那些有关“游戏感觉”方面的问题，比如“如果我们加快导弹

的发射速度、减小重力并降低开火频率，那将会怎样呢？”

在当前的游戏市场竞争程度下，开发者们认识到只有最好玩的或者是最精彩的游戏内容才可能成功。很多游戏开发者都清楚，经由脚本工具，将团队成员专门集中在游戏可玩性内容的构造和监控将更有助于形成一个更加健壮的游戏。

至于脚本语言的选择，一些开发者会选择使用 Visual Basic (<http://msdn.microsoft.com/vbasic/>)、Python (www.python.org)、或者 JavaScript / Jscript (www.ngweb.biz/software/djsedit.shtml)，而其他的开发者会在完全自主版权的工具中定制自己的脚本系统。对于后一种情况，鼠标右击场景中的一个角色就可能会出现一系列“属性”编辑标签，这些编辑标签可以用来指定所选角色的一些细节。这些属性中的一些例子有：

- 角色的巡逻速度（某个角色在保护一个边界时的移动速度是多大？）；
- 角色的巡逻边界形状（这个角色是沿着一个方形边缘、半圆形还是一个自定义的形状移动？）；
- 角色对其他角色类型的敌对行为（朋友还是敌人？进攻还是防守？）；
- 角色缺省攻击武器的选择（激光炮？还是棍棒？）；
- 角色的缺省防守行为（使用防护罩、轧成一团还是自行毁灭？）；
- 角色缺省的等待状态（来回走动，坐下来，还是揉揉眼睛？）；
- 角色的物品清单（物品的使用顺序？武器的优先级？）；
- 角色对攻击的响应（一直战斗到死，战斗到伤害程度 $\text{damage} = 80\%$ ，还是一直战斗到没有弹药为止？）；
- 为角色规定好顺序。（寻找另一个角色、寻找地图上的位置、寻找关卡的中央、寻找弹药库、返回营地和最为相似的角色组合在一起、玩牌 / 掷骰子[或者其他一些有趣的内容]、请求帮助、掩护或者隐藏、尾随或跟着另一个角色、寻找娱乐、寻找某样东西一直到满足寻找条件为止？）

作为例子，这里仅列出一部分属性，同样，大量其他角色所具有的属性都能够以这些类似的方式加以编辑。因为到最后，在游戏重新编译或者重新构造时所有编辑的工作都需要反映到代码中，所以你会经常需要从工具设计程序员那里拿来的编程支持工具来构造你想要编辑的每一个属性。

为了给你举例说明不同的脚本语言看起来是什么样的，这一节将把一个简单的怪兽行为脚本翻译成 JavaScript、Visual Basic、Python 和 Perl 语言。在本章内容中，我列出的脚本大多数情况下都是使用 JavaScript 语言写的，这是因为 JavaScript 是支持良

好(对于不同平台)的脚本语言,JavaScript的编辑器可以免费获得,而目前 Visual Basic 和 Python 也变得越来越流行。

这个示例脚本要求怪物一旦看见玩家时就过去进攻,如果看见武器的话就拾起来,而在其他情况下它就简单地进行巡逻。

■ JavaScript 示例

```
function moveMonster() {  
    if(monster.canSeePlayer()) {  
        monster.attack();  
    } else if(monster.canSeeWeapon()) {  
        monster.pickupWeapon();  
    } else {  
        monster.patrol();  
    }  
}
```

图 6-1 显示了这些脚本代码在一个编辑器那看起来是什么样子的。

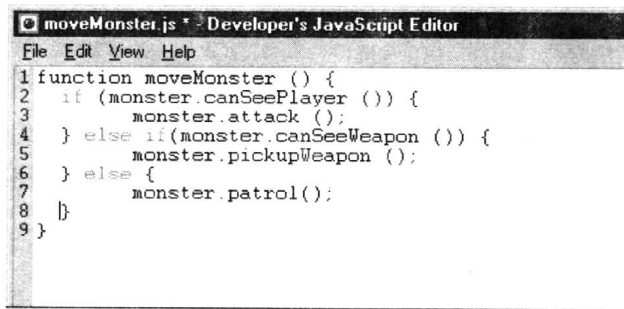


图 6-1 编辑器中的 JavaScript 代码示例

■ Visual Basic 示例

```
Sub moveMonster()  
    If monster.canSeePlayer() Then  
        monster.attack()  
    ElseIf monster.canSeeWeapon() Then  
        monster.pickupWeapon()  
    End If  
End Sub
```

```
Else
    monster.patrol()
End If
End Sub
```

■ Python 示例

```
def moveMonster():
    if monster.canSeePlayer():
        monster.attack()
    elif monster.canSeeWeapon():
        monster.pickupWeapon()
    else:
        monster.patrol()
```

■ Perl 示例

```
sub moveMonster {
    if( monster->canSeePlayer() ) {
        monster->attack();
    } elsif ( monster->canSeeWeapon() ) {
        monster->pickupWeapon();
    } else {
        monster->patrol();
    }
}
```



6.2 脚本应用示例

要说明脚本功能强大,最好办法之一就是分析几个常见的用于在游戏中控制动作行为的脚本案例。脚本可用来控制很多类型的自定义事件,因此这里绝对不可能对每一个可能的脚本控制的举例示范。基于脚本的控制是自己定义或用户可定义的,在这一点上,要和一个工具或者引擎技术的程序员或团队配合,这是非常重要的。双方(程序员和脚本编写人员)需要就行为动作的哪些方面有助于实现最为强大的游戏可玩性编辑功能上达成一致。关于怎样使用脚本来控制不同类型的游戏行为,下面的示例会让你有一个实实在在的理解。

■ 非玩家角色对话模板

非玩家角色“居住”在许多游戏世界中，并能以各种不同的方式加以使用。比如你可能想要控制某个非玩家角色如何完成与一个玩家角色的对话。我们假定 *actor_21* 是一个洞穴的守卫（一只防止他人进入洞穴的野兽），我们要让这个守卫洞穴的角色简单地将三条线索或者一些信息传达给玩家，这些线索和信息包含有洞穴里藏有什么东西。其脚本可能看起来会像下边这样：

```
If (cave_guard.isDisturbed()) {  
    playMessage(text1.txt);  
    waitForUser();  
    playMessage(text2.txt);  
    waitForUser();  
    playMessage(text3.txt);  
    waitForUser();  
}
```

这个例子中，洞穴守卫收到惊扰之后将会给玩家发送信息，同时等待玩家对收到的信息进行确认。如果玩家在看到前三条信息后返回来再一次和洞穴守卫进行对话，这一次你可能想要洞穴守卫随机地给出一些回答，你可以通过简单地对脚本进行扩充就能做到这一点，这样在第一次或者说初次接触后的每一次接触都将给出随机的答案，这些随机答案装载在一个文件中用来随机回放给玩家。

■ 射手飞行模式

对于街机风格的射击游戏，敌人通常可以用不同的方式、不同的速度和不同的飞行模式进攻。这些类型的动作也能够通过脚本系统进行编辑，下面我们看一个例子。

假设你正在构造一个射手。敌人可能只是从屏幕的最右端的远处向左侧移动着进攻，这几乎是所有经典的横向卷轴式射击类游戏的固定套路。也许我们想要控制那些敌人的攻击模式。这里需要大家像上数学课一样集中所有精力！哦！游戏中的运动是纯数学上的问题。所有这些漂亮得让人吃惊的三维操作是怎样产生的呢？它完全是数学问题，彻头彻尾的数学问题！

在一个开发团队中，编写脚本已经成为设计人员工作内容中至关重要的一部分，因此了解（这是最起码的要求）一些基本的程序设计和数学是非常有用的。如果你的基本

编程技巧比较生疏甚至根本就不会编程，那你就应该从网上（www.ngweb.biz/software/djsedit.shtml）下载一个免费版本的 JavaScript 编辑器开始练习吧。

下面是一个简单的脚本示例，其中定义了一个名称为“*enemyactor*”的敌方角色（比如，一个火星上的游牧部落或者是邪恶的环状飞行物）的进攻行为：

```
function moveEnemy() {  
    enemyactor.moveLeft();  
    // 将一个正弦波放大 100 倍后再加上 240，  
    // 画在 640×480 的屏幕中央位置  
    var x = enemyactor.getX()%360;  
    enemyactor.setY(100*sin(x*4)+240);  
}
```

图 6-2 以一个标准正弦波的形式给出了 *enemyactor* 的结果进攻模式。

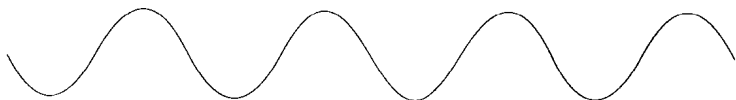


图 6-2 简单的正弦波形进攻模式

好的，这样 *enemyactor* 就会沿着正弦波形的路径发动进攻。一种进攻模式（我想大家以前应该见过）就构造完成了，但是我们想构造很多其他的攻击模式。即使是一个很简单的射击类游戏，我们也可能想要在每一个游戏关卡中构造至少一种不同的进攻模式，比如我们需要一个大的慢速斜线进攻模式。在开始构造进攻模式时，我们可以直接对正弦波的值进行调整。

实用的一点是：大多数的编写脚本工作仅涉及到调整或编辑变量值、保存以及重新构造。有些时候你仅在处理大量针对各种各样的行为调整好的脚本版本上就花费数天的时间。在有些情况下，当你看到这些脚本或者“行命令”时你甚至不用去编写它们；你或许只需要调整一些取值就能得到预期的效果。

现在，让我们通过改变这个正弦波的幅值来构造一个不同的进攻模式。这个例子有助于说明这样一个观点：就像这个简单的进攻行为一样，很多行为成形都是从对一些数值的修修补补开始的：

```
// 通过调节幅值来改变 enemyactor 的正弦波式的进攻模式  
function moveEnemy() {
```

```
enemyactor.moveLeft();  
// 通过另一个正弦波来修改正弦波幅值  
var x = enemyactor.getX()%360;  
enemyactor.setY((50*sin(x)+50) * sin(x*4) + 240);  
}
```

图 6-3 显示了通过修改幅值调节进攻模式后的结果。

现在我们的 *enemyactor* 有了一个修改后的攻击模式了。正如你所看到的，通过简单地改变这些取值和玩儿玩儿数字游戏，我们就能够制作出很多其他的进攻模式。通过改变正弦波的频率，我们还可以制作出更多类型的攻击模式来。

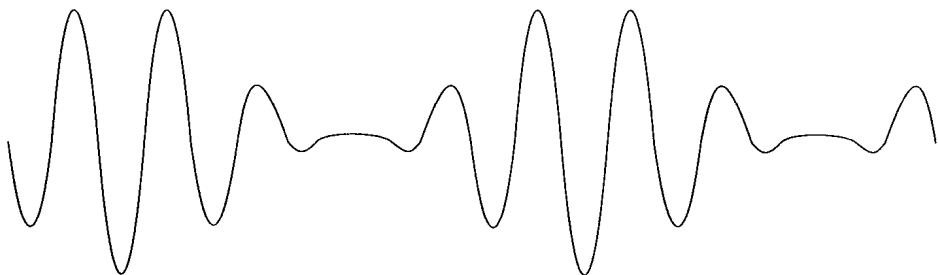


图 6-3 调节幅值后的正弦进攻模式

下面是一个修改频率来改变进攻模式的例子：

```
// 通过改变频率来改变 enemyactor 的正弦波攻击  
function moveEnemy() {  
    enemyactor.moveLeft();  
    // 利用另一个正弦波来改变正弦波的频率  
    var x = enemyactor.getX%360;  
    enemyactor.setY(100 * sin(x * (2*sin(x)+3) ) + 240);  
}
```

图 6-4 给出了通过变换或者说改变频率得到的修改后的攻击模式。



图 6-4 频率改变后的正弦进攻模式



6.3 应用触发器

正如我们已经看到的那样，一些行为是用脚本处理的，这些脚本可以用 JavaScript / JScript、Python、Visual Basic 或者其他自主知识产权的语言编写。然而，作为这些简单的脚本系统的补充，复杂的触发器系统的应用也越来越多。触发器就是一个简单的开关，当它被触发时会引发一个事件或者一系列的事件。这有时也称作事件处理 (*event handling*)。

构造并使用复杂的触发器系统在实践中有几大优点。第一，用于控制触发器属性的变量和参数编辑器能够很容易地内嵌到基于 Windows 技术的编辑器 (基于 GUI 的编辑器) 中。第二，设计者和美工人员很少有人能有中等偏上的编程能力 (如果会编程的话)，因此属性编辑功能以及构造触发器行为的工作流程对于非编程人员来说应该非常简单才行。

功能最为强大的通用编辑系统应该将触发器功能和行脚本编辑 (比如应用正弦形攻击模式的射手行为) 结合在了一起。在编辑器中，触发器通常放在一个三维场景中，并用一个图标表示。对玩家来说，它们是完全不可见的。触发器能够以各种有意思的方式用来实现游戏中的行为。

在大多数的场景编辑软件中，都可以通过设置触发器的属性来使其包含几种类型的相关信息，这些信息包括：

- True 和 false 模式 (用来设置触发器的逻辑状态或条件)；
- 值域 (用来输入半径、距离、力气和偏移量信息)；
- 文本区域 (为玩家屏幕上打印信息，或者是用于测试目的)；
- 文件指针 (指向一个文件资源，该文件资源是触发器事件的一部分)。

我们来看一个简单的例子，这个例子来自十分流行的游戏地图编辑器 *UnrealEd* (*version 3.0*)，它将向你展示这种类型的触发器编辑的强大功能。在很多游戏中，当一个角色被武器击中时，我们经常要加载某种类型的行为结果。对于一个射击触发器，被某种投射武器射中时会触发的触发器，我们要加载一个行为——可能是另一个脚本事件——也可能仅仅显示一条简单的文本信息。

图 6-5 显示了一个设定好的触发器类型 (*TT_Shoot*)，当它被触发时，将显示一条文本信息：“哎唷，你打中我了！”

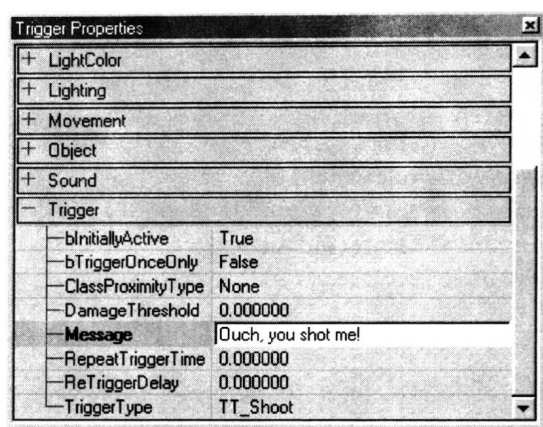


图 6-5 使用触发器显示触发信息

当然，这是一个非常简单的例子，但它揭示了在处理角色行为时拥有一个域编辑触发器系统其功能是多么的棒。为了便于编辑基于触发器的行为，很多游戏开发者都开发了类似的触发器系统。

需要强调的一点是：触发器能够在具有可编辑域的用户图形界面型式工具（比如 *UnrealEd* 示例）中进行编辑，也能在脚本代码中直接编辑，或者这两种方式都使用。

现在，我们选择了一些通用的触发器类型来看一下。而且，为了增加你的游戏可玩性，你可以制作或者自定义触发器类型。

■ 恐惧点触发器

“恐惧点”触发器可以构造“汲入点”，敌人不敢踏入此区域内。你可以将“恐惧点”触发器安放在地图上的一个巨大神像旁边——一个充满敌意且具有至高无上权力的神像，他会让你的敌人产生发自内心的恐惧。这种类型的触发器可以是半径驱动的，这样敌人就必须距离神像超过定义好的“恐惧点”半径属性（比如，在 12 个全局单位的半径范围内）。在游戏可玩性方面，这些区域可以作为玩家角色的一个安全区域或者补给点。战争可能会在这种“恐惧点”的触发区域外的附近发生。

玩家可以躲藏在这个神像附近并将它作为一个自由活动区域，因为一些敌人（那些设置为对“恐惧点”有反应的敌人）不能进入“恐惧点”半径区域内。应用这个触发器，当敌人进入“恐惧点”触发区域的边界或半径时，你可以通过一条命令行脚本（就像前

面的正弦波攻击范例)来播放一段“恐吓或者吓唬”动画。这种行为所产生的效果是:当这些敌人在追逐你的过程中触发了触发器时,他们便会畏缩(他们用胳膊遮住脸,甚至畏惧到不敢看这尊神像)并夺路而逃(这只是一个可能的例子)。

下面的脚本范例是:当敌人触发“恐惧点”触发器时,他们将会因恐惧而畏缩并逃离。

```
// 支持“恐惧点”触发器的 JavaScript 范例
if(monster.inside(fearThisTrigger.getBoundary())) {
    monster.playAnimation(fearAnim);
    monster.reverseDirection();
}
```

■ 可视距触发器

可视距在游戏术语中有几个不同的用法和意义。通常,它是指角色能够看到的東西。它可以指代表游戏中玩家的某个特定角色所看到的東西。通常情况下,它是指一个角色能不能看到一个特定的物体或者另外一个角色。可视距可以以集中不同的方式进行计算,并且并不总是指从角色的角度出发在物理上可以看到的東西。其语义非常丰富。

当玩家进入可视距半径时,或者说在地图上当玩家距离可视距触发器的物理距离足够近时,可视距触发器将会执行它的事件。这可以用来引发一块巨石危险的滚动动画(想一下《夺宝奇兵》——*Raiders of the Lost Ark*和《印地安那琼斯》——*Indiana Jones*)。当玩家触发可视距触发器时,就会播放一个以一定运动路径的巨石滚动动画(换句话说,巨石从哪里开始滚动,它滚动的路径是什么样子的,还有它在什么地方毁灭或者停止滚动?)。

你也许注意到了很多触发器都依赖于一个半径。将事件之间的相互关系调到“感觉”良好的时候多半都落在了对半径的调整上。

如果玩家可以看到这块巨石,下面的脚本就会让巨石滚动并将镜头聚焦到这个动作上。

```
// 支持可视距触发器的 JavaScript 代码
if(objectVisible(player.currentPosition(), boulder)) {
    boulder.startRolling();
    playSound(boulderSound);
    camera.focus(boulder);
}
```

■ 群体触发器

群体触发器触发时会给一大群角色发送动画行为信息。举个例子，你可能进入了一个部落村庄，并有几个武士列队在村的路口处等你。在村庄里当你拾取或者是试图偷一些东西，就可能触发一个群体触发器。这可能会使所有这些武士，他们原本是在欢迎你来的，都带着杀气在地图上到处找你。群体触发器使得这些武士从欢迎你的状态变成了搜索玩家的进攻状态。

你也可以让一个群体触发器的有效范围更小一些。玩家一旦触发了触发器，会引起这些围成一圈的武士开始敲击他们的战鼓。在这种情况下，群体触发器只是简单地告诉武士角色从坐在鼓边的动画状态变成击鼓的动画状态。这个群体触发器还会触发一个声音触发器（武士击鼓的声音）。

很多触发器都像这样“串连”在一起。一个群体触发器触发一个动画和声音触发器，而动画和声音触发器又可能触发另一个特定事件（比如说，当武士开始击打他们的鼓乐时火开始越燃越旺）。

下面的示例脚本用来控制一定数量的武士，使他们得知玩家从他们的村庄中拿走了一件神圣的东西，并使他们想找出玩家并杀掉他：

```
// 支持群体触发器的 JavaScript 代码
if(player.pickedUp(sacredObject)) {
    for(var n=0;n<numWarriors;n++) {
        warrior[n].seek(player);
        warrior[n].setState(HUNTING_STATE);
    }
}
```

■ 天气效果触发器

如果你的游戏中没有内建一个动态的天气系统（一个随机改变和更替天气模式的系统），那么你就可能想利用一个天气效果触发器来控制天气。也许在一个关卡任务概述之后，当玩家开始一个新的征程时，你也许想要增加一些雨和 / 或雾的效果。这就可以通过一个天气效果触发器来实现。

当触发该触发器时，在你的地图上的一个易于编辑的区域内便开始下雨。单一的世界定位单位便可以定义地图上的任何区域。你甚至还可以确定降雨的强度、范围和方向！你也可以用类似的方法为某个区域设置雾的参数。然而，这种类型的效果必须尽量少用，

因为它和其他的粒子系统效果一样，计算量很大。

下面的示例脚本用来控制，在任务简介一旦结束时，两个区域以不同的强度开始降雨：

```
// 实现“天气”触发器的 JavaScript 示例
if(player.finishedBriefing()) {
    // 使“region1”降小雨，等级为“20”

    weatherSystem.startRain(region1, 20);
    // 使“region2”降雨增加到等级为“50”
    weatherSystem.startRain(region2, 50);
}
```

■ 计数触发器

计数触发器只是简单地对一些时间单位进行递减计数，然后执行其事件。计数触发器完成一次事件要等待的时间数可以在它的属性中进行设置，这就是让你和时间比赛。你将会留给玩家多长的时间来逃跑呢？等待下一个事件触发需要多长时间呢？因此，如果玩家触发了一个计数触发器，可能十秒钟以后一个巨大的机器人蜘蛛将在他们身后出现。你还打算调整蜘蛛的进攻速度和进攻的性质以给玩家一个生还的机会。

你可以利用地图上一边的一个计数触发器来引发地图上另一边的一个计数触发器。这样当整个地图上的计数触发器触发事件时会产生一种“爆米花”的效果。这些对于动作类游戏和环境射击类游戏是非常有用的。

下面的示例脚本展示了你可以怎样来警告玩家他已经进入机器人蜘蛛的巢穴，给玩家一个逃跑的机会，然后让蜘蛛追捕玩家：

```
// 支持“计数”触发器的 JavaScript 代码
if(player.intersects(spider.getBoundary())) {
    alertUser(spiderWarningText);
    waitSeconds(10);
    spider.startHunting();
}
```

■ 材质触发器

当材质触发器被玩家的位置触发器触发时（记住“爆米花”效果），场景中的材质信息会发生改变。举例来说，玩家可能触发了一个位置触发器，它使覆盖有灰尘粒子效果的地面上浮现出一种冰晶状物质。当这些冰晶物质出现时，会触发光效触发器，使空

间内的光线增强（产生生动的视觉冲击效果！）。

利用墙的材质或墙的材料以某种方式来解决那些从冰冷的冰晶上反射回来光线的问题，这难道不是更好吗？不错，你可以为墙的材质制作一个动态的“光线反射”版本，并在材质触发器触发时显示它。现在，当冰晶出现时，就会在你自定义的墙的材质或材料上显示这些动态的光线反射了！太酷了。如果没有对材质触发器指定新的“材质”信息，它通常情况下就什么也不做。如果没有新的材质信息它就不知道该做什么了。

下面是为出现的冰晶设置材质信息的脚本示例：

```
// 支持“材质”触发器的 JavaScript 代码
if(iceCrystal.rising()) {
    iceCrystal.setTexture(animatedLightTexture);
}
```

■ 音效触发器

音效触发器包含了所有的音效事件。如果一个音效触发器触发后，它可能会使背景音乐变成“战斗场景”的音乐。它可能只是简单的开始一个新的背景音效的事件（就像山洞中滴水的回音），或者是引起话外音叙述，还有可能产生某种特定“一次性的”声音，就像一个定制的爆炸声同时伴着一个垂死的角色在远处喃喃地说着他的最后一句话。

下面的示例脚本会在战争爆发时将背景音乐设置成战争配音：

```
// 支持“音效”触发器的 JavaScript 代码
if(player.startedFighting()) {
    audioSystem.setBackgroundMusic(fightingTrack);
}
```

■ 消息触发器

消息触发器可以用来直接给玩家打印一条消息，向玩家的界面系统传送一条消息或者引起一个非玩家角色事件。消息触发器还可以用作测试目的的信息打印，通知玩家或者测试人员关于地图上某物的状态。

下面的示例脚本打印了一条奖金等级的信息：

```
// 支持“消息”触发器的 JavaScript 代码
if(player.hasEntered(bonusArea1)) {
    messageSystem.prompt(bonusLevelMessage);
}
```

■ 光效触发器

光效触发器可以用来产生自定义的光效环境。也许你打算从视觉上试图表现“结束时间”的场景，就好像在一座巨大的宫殿内从我们的视角所看到的阳光日复一日地照射到外面的花园里一样。我们可以使用一个光效触发器来实现这种效果，当它被触发时引发一个循环播放的光线轨迹的动画来模拟天空中太阳位置的变化。

再简单一点，我们可能只是想让一缕光线洒进屋内，并照射到遗物上、一把充满魔力的剑上，或者是照射到一个主角的助手角色上。我们也可以使用光效触发器来实现这种行为。

下面的脚本示例实现了如下功能：当玩家靠近主角的助手时，怎样产生一缕光线照在助手身上：

```
// 支持“光效”触发器的 JavaScript 代码
if(player.nearHeroHelper()) {
    createSpotlight(heroHelper);
}
```



6.4 构造行为

现在你已经看过了一些基本的脚本示例，并体验了触发器带来的一些强大功能。构造行为通常需要同时使用脚本和触发器。重要的是，你需要学习足够多的关于脚本和触发器的知识，这样才能将它们的强大功能结合在一起构造行为。我们来看一下另外一个例子，它演示了你应该如何使用触发器信息来控制整组的生物。

■ 动物构造器

有时候为了纯粹的视觉效果，你想使用触发器来控制道具生物的行为，这可能包括成群的昆虫、成群的鸟或者成群的鱼。根据所用脚本系统的不同，有时候你要使用命令行入口（就像正弦波进攻模式），而其他一些情况你只要在工具中进行简单的属性设置——用这些属性定义触发器的工作方式。

这样的话，假定我们定义了一个类型为 *creature_creator* 的触发器，我们可以对它的行为进行编辑，其属性种类包括（但并不局限于此）如下内容：

- 可以构造生物的最大数量（比如说 10）；

- 哪些类型的生物将会产卵？（我们可能有 15 种选择，bug_1、bug_2、fish_1、fish_2，等等。有时候我们甚至使用同一个触发器来产生两种或更多种类型——就像从同一个蜂房里出来的两类不同的虫子。我们可以将这个触发器藏在场景中的一个蜂房角色里边。）；
- 以单位计量，产卵半径是多大？（在距离触发器多远的范围内这些生物会产卵？）；
- 产卵的频率或者说速度是多大？（这些动物产卵的速度有多快？）；
- 这些卵的寿命有多长？（在它们死亡或者消失以前这些卵将在屏幕上“存活”多长时间？）；
- 触发器的触发半径是多少？（当玩家距离触发器有多远时触发器开始产卵？）；
- 哪些因素会影响这些卵？（它们远离触发器还是紧跟着触发器？它们是不是按照噪音模式随机地从触发器旁边散去呢？它们是不是就像是在寻找什么东西似地很快地散开并又返回来呢？）；
- 这些卵出生后的运动路径是什么？（昆虫可能会飞、蜚人以及盘旋；鸟儿可能会飞以及集聚成一群；鱼可能会聚成群和巡游——在这里我们可以使用命令行给这些卵指定行为的路径或曲线，来定义它们随时间变化的行为方式。通常情况下，从本质上说，这些路径可以定义为线性函数或者三角数学函数。）。



6.5 引擎实现与 Unreal 引擎

《魔域幻境》——Unreal 游戏的引擎 (www.epicgames.com) 是很多游戏开发者所选择的一种杰出的技术解决方案，他们利用 Unreal 的引擎技术制作他们自己的游戏内容。选择这种开发方式对开发者来说既有利也有弊。如果游戏开发者选用像 Unreal 引擎这样的健壮的解决方案，而且该引擎是不断地增强和改进，游戏开发者可以缩短开发时间和减少开发费用；否则的话，他们就得开发自己的游戏引擎。如果他们不得不自手起家制作一个原创的游戏引擎的话，就可能会由于资源问题导致游戏工程的无法完成；而用现成的引擎技术能为开发者的开发工作节省大量的资源，使他们敢于承担原本完成不了的工程。

使用授权的游戏引擎，比如像 Unreal 引擎，当你所开发游戏工程在内容方面与 Unreal 引擎所设计的相互兼容时，就能够取得最佳的效果。如果你正在开发一个第一

人称竞技射击游戏，Unreal 就是很自然的选择，并本身也很适合。另一方面，如果你正在开发格斗竞赛游戏或者体育竞赛游戏，应用 Unreal 引擎就不是一个很好的选择。从定义上讲，游戏引擎是为了处理某种特定类型的游戏中的问题而设计的。这就是为什么拥有一个完整而健壮的游戏引擎会给游戏开发者带来很显著的成效。如果资源允许的话，还是制作自己原创的游戏引擎吧，资源一个是很好的理由。你是不是处在一种很微妙的境地，不是吗？

游戏开发者们专门为他们自己的游戏工程构造游戏引擎，这样做所带来的价值和灵活性正是这些长期从事开发工作的开发者们需要的。一旦他们把一种类型游戏的引擎打造完工，并且已经变得稳定而可靠，就可以通过增强游戏引擎来使之跟上发展趋势，并且可以用它来制作新的游戏。

然而，按照产业内的认识，游戏引擎是独立的实体。当你从一个开发岗位换到另一个开发岗位，本质上也就是从一个引擎换到另一个引擎，你将发现所遇到的游戏引擎在功能、运行方面、工具支持以及健壮性方面存在着很大的差异。

这种环境会帮助我们使用可靠的游戏引擎前必须作出切实的选择。

Unreal 引擎是一个很好的解决方案，因为它是一个有着强大的游戏社区支持的健壮的、跨越几个时代的游戏引擎。Unreal 技术为游戏开发提供了完整的引擎代码平台和工具包。发行商 / 开发者在注册或者付费后才能使用这个引擎技术开发游戏，这就节省了引擎开发的费用，并且，在大多数情况下，极大地缩短了制作具有竞争力的游戏内容所需的时间。

就像我以前说过的，你可以有一个相当不错的游戏引擎，它只是没有工具管道或者没有工具支持，还有，在构造很好的游戏可玩性方面其功能又十分有限。Unreal 引擎在这方面的支持做得也很好，它有一个详细的、功能强大的工具系统，而且随着一代一代的新引擎的推出，其工具系统也越来越好。

为了能够在上下文中使我们看到怎样使用本章所讲述的触发器来构造游戏的可玩性，我们来快速地浏览一下 Unreal 引擎是怎样处理这些事件的。图 6-6 展示了 Unreal 编辑器在以整洁而有效的方式设置基本触发器属性时的强大功能。图中显示的是：我们正在设置能够触发一个接近触发器 (TriggerType = TT_ClassProximity) 的角色类型 (士兵)，这样，任何一个“士兵”类型的角色都能够触发这个触发器。请注意一下使用该工具时内容生成器中所有的有用的属性和配置选项。

当你考察这些各种各样的用于商业游戏开发的引擎时，你很容易就发现其中一些引

引擎有着特别的扩展功能,这些引擎的工具和脚本系统会给游戏可玩性的实现带来很大的方便。当选择游戏引擎技术时,这也许是需要考虑的最为重要的方面之一。

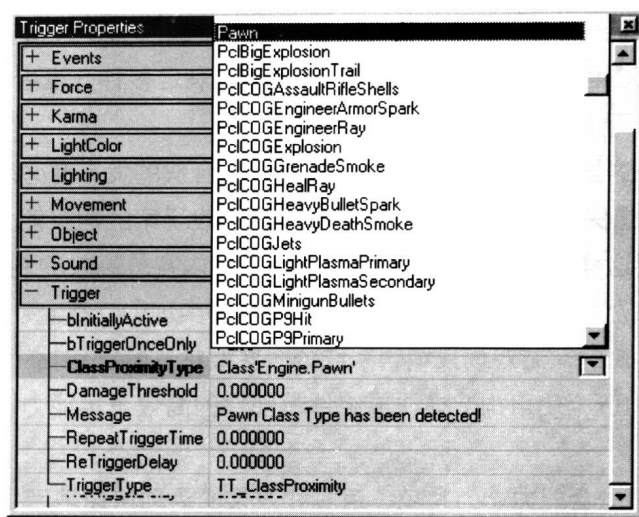


图 6-6 在 UnrealEd 中设置触发器属性



6.6 有关脚本 / 编辑系统的因素

就如你可以看到的那样, 在游戏中的脚本和触发器系统之间还有很大的发展空间。始终阻碍这些工具发展的实际原因是有预算、时间表安排紧凑、还有与资源方面的考虑。(在预算和进度表允许的情况下, 你能找到愿意全身心地投入到这些工具的开发的优秀程序员吗?) 由于运行于当前控制台和个人计算机的游戏的复杂性和广泛性, 开发经费的增长非常缓慢, 而同时游戏开发的时间进度表却趋于紧缩。

玩家希望他们所喜爱的游戏系列或原创游戏中不断涌现出新的内容, 这一点对于整个游戏产业都是非常有利的, 这是可以理解的。而且, PlayStation2 已经进入了世界各地的千家万户之中, 并且伴随着像 PlayStation2 这样的控制台获得的广泛成功, 市场对新的游戏内容的需求正在迅猛增长。上述这些因素, 再加上大量多玩家在线游戏 (MMOGs) 的发展以及这些游戏本身也要求不断地推出续集并持续地更新, 便产生了一种情景——开发者能不能快速地开发出精心制作的游戏内容决定了他们是开始繁荣还

是开始衰败。

游戏内容本身的粒度和生命力依赖于适合使用的健壮脚本和构造工具。到目前为止,仅有少量的开发者开发出了令人印象深刻的脚本工具集和技术,但据我所知,没有任何一个能够提供一种统一的解决方案。这种统一的解决方案还应该成为整个游戏引擎技术的一部分,这里提到的引擎技术要足够灵活以满足多种开发者的需求,并在整个产业中得到广泛使用(授权的)。

为了满足我在这里所提到的一些要求,开发者与合伙的发布商必须要进行相当规模的投资。如果他们想保持市场竞争力,并能够根据游戏玩家的购买能力即时而恰当地作出反应,他们需要这样的构造系统——能够支持游戏内容最高效的制作和测试。当前,这对很多开发者来说,既是一个要解决的问题也是要实现的目标。

就像我们所讨论的,游戏场景编辑系统的一些概括性特征包括:

- 快速图形构造(使用构造模块或组件、部件的高效重用能力以及方便地添加附件的能力);
- 强大的动画融合能力(为角色、道具和效果的运动快速地产生中间帧的能力);
- 大量的内建或引擎内含的效果编辑功能,包括角色效果、道具效果和自定义环境效果(天气);
- 在角色和触发器定义中(在有用的地方)的命令行编辑功能;
- 音效标签功能(将音效事件绑定到角色上的能力)。

拖放构造系统和即时脚本功能(右键点击屏幕上的一个角色不只是设置属性而且还能“定义”新的行为)是怎样结合在一起而成为一个技术解决方案的呢?它也许能够成为每种类型游戏的一个很好的解决方案。请记住还有另一个很重要的问题需要处理:根据定义,每种类型的游戏都有其自身特定的构造和脚本的简化需求。

到目前为止,一些开发者已经在这方面取得了很好的成就。然而,是不是这些系统中的任何一个都能够具有我所建议的这种类型的功能和扩展性呢?许多的游戏界人士是否同意这个看法,我没有把握作肯定回答。



6.7 棒球游戏脚本评论的案例分析

如果足够幸运的话,你将从游戏的设计阶段进入到游戏的开发状态。这是一个不错的突破,因为在游戏的期望与挑战方面,这是对一个给定的游戏工程的历史赋予真实感

的最为有效的途经。

在整个开发周期中,开发团队审议游戏开发工具事项的时间越早越好,而不应该在没有或者是很少考虑游戏工具方面的事情的情况下就开始游戏开发,并在开发过程中以某种方式对工具进行查漏补缺。这一点很重要。

除非开发团队非常仔细地规划了一个坚实的脚本和触发器系统,否则的话可能会没有一个这样的系统供你使用。程序设计人员已经面临着不断出现的时间和功能方面的挑战,他们自己将会在游戏引擎底层的角色代码部分对游戏代理的角色行为进行简单的编码。“暴露”这些变量(或者将这些变量提供给脚本系统)会增加复杂性、规划和执行方面的需求。有些情况下,并不需要一个完整的脚本系统的,这些时候,直接对游戏代码本身进行编辑可能会更简单一些。其判断准则应当是——什么能够更佳地满足游戏的需求,什么就是最好的。

然而,大多数的现代游戏都具有覆盖面广以及潜在的可扩展能力(具有附加的关卡或者任务包,续集或者游戏交付定购)的特点。脚本系统通常被证明是必要的且是有价值的工具,它支持当前游戏的构造工作并且足够灵活,能够满足游戏未来开发的需要。记住,决定如何更好地对待和修正未来设计项目脚本的要求这一点往往是很难做到的。不同类型和风格的游戏需要独特的、专门的脚本系统的支持。

一旦游戏的设计细节设定好了,而且基本的设计/特征确定下来了(不能再增加新的关键要素了),你的游戏的可玩性目标在概念层上应当是清晰可见了。在构造脚本系统时你需要清楚这里头的每一个细节。如果你还在设计你的游戏细节,那就不要在构造脚本系统上浪费资源了。当然也有例外,如果你的团队对游戏可玩性想法的高层概念很有信心的话,那就完全可以考虑你的脚本需求了。

每一种类型的游戏都有各自基本的需求和专门的需求,并且你自己的游戏可玩性想法也会产生各种需求。你可能想在开始时拥有一个详细的列表,来确切地描述出你需要控制哪些类型的参数。运动类游戏通常都是富有挑战性并且具有脚本密集型(需要编写大量的脚本)的特点。它们背后所使用的脚本工具也经历了多次快速的改进。我们来看一下在为棒球游戏制作脚本工具时你应该怎样开始这个过程。

你可以这样开始:把一个棒球游戏分解成各种场地操作角色或各种场地上的行为,接着围绕这些种类来实现工具的功能。我们把脚本分成“通用”、“投球”、“防守”、“击球”和“接球”五类开始设计。为增强游戏的可玩性,在每一类中我们打算控制哪些要素呢?这些细目分类开始不断演变,你也找到了制作脚本工具的一个起始点。

■ 通用

通用类别就是我们的基本的或者说多用途的类别。从广义上讲，该类别包括我们所有的多用途脚本和游戏控制要素，应该包括如下内容：

- **群体的反应** 为群体音效以及动画提示设置触发器；
- **比赛情节的镜头细节** 设置摄像机惯例。摄像机镜头如何从多个视角跟随每一个动作呢？很多游戏都允许用户自己配置游戏视角，但这些视角必须事先为玩家确定下来并精心设计；
- **戏剧亮点或游戏高潮** 为全垒打、漂亮的接球、或者精彩的实况设置特写画面和细节回放；
- **即时回放调整** 配置游戏过程怎样进行内存缓冲并在玩家请求时进行回放；
- **道具控制** 定义道具的行为，如魔法行为或空中小型飞船的飞行；
- **每种定位类型的人工智能控制器** 在玩家与 CPU 对抗的模式中，为 CPU 玩家配置“最佳的”行动以对抗人类对手；
- **天气状况** 在没有动态天气系统的情况下，你必须处理天气对环境的外观和感觉的影响（比如，在球的飞行过程中风对它的影响，小雨能使棒球场草地的摩擦力增加多少）。

■ 投球

我们想使用脚本和改变取值来对棒球中不同类型的投球进行很好的控制，这些投球有突破球、曲线球、上升快速球、下降快速球、慢速变化球、以及其他类型的投球。我们也能构造出几种全新的投球方式（其中的一些可能连人类选手都无法完成的）。我们实际想要做的就是为玩家重现当其作为一名击球手面对各种各样的投球时所面对的感觉和情景。在投球这个类别中，我们从一些基本内容开始，包括：

- **速度** 球冲向玩家的速度；
- **球的旋转** 球旋转的速度和方向；
- **上升期** 球在飞向接球员的过程中上升的幅度；
- **下降期** 球在飞向接球员的过程中下降的幅度；
- **偏向左垒边界** 球向着本垒板左边界偏斜的“引力”有多大；
- **偏向右垒边界** 球向着本垒板右边界偏斜的“引力”有多大；

- **随机性** 将所有这些值都设置成一个随机数，看看到底会发生什么！
- **坏球种子值** 控制坏球的出现频率。

这些值中的绝大部分均可以通过一个可进行数值编辑的简单的图形用户接口工具进行调整，并不是必须要用脚本才能完成。然而，根据你的开发工具配置的不同，你可能根本就没有图形用户接口工具，那你有可能要亲自在游戏代码或者脚本中直接编辑这些值了。

■ 防守

构造防守队员的防守行为将是一项复杂的任务，并且永远都做不到完美的程度，因为棒球从其最根本的本质就要求对某种环境中的任何给定状态进行快速实时的分析。（难道我要冒出界的危险吗？我想要双杀还是轻松出局？）对于防守我们可以从下面几点入手：

- **接球半径** 在该半径以内我们可以认为接住了球；
- **投球速度** 球投向垒包或者传向另一名队员的速度；
- **投球距离** 球在空中飞行了多少个单位的距离；
- **奔跑速度** 防守队员完成一个接球、俯冲或跳跃动作时的速度有多快；
- **球的阴影大小** 球的阴影有多大？在设置接球时很有用。

和以前一样，在这儿你也可以用图形用户界面工具来修改这些数值，或者你可以直接在脚本中进行修改，如同下面的例子：

```
// 编辑“防守”数值的 JavaScript 代码
function processOutfielder(outFielder) {
    if(outFielder.nearestBall()) {
        outFielder.seekBall(20); // 20 在“可修改”值域内
    }
    if(outFielder.ballRadius(5)) { // 外场手和球之间的距离小于5个单位距离吗？
        outFielder.doCatch();
    }
    if(outFielder.hasBall()) { // 接住了球
        // 将这个队员的最大投球距离设为“60”
        outFielder.throwToNextPlayer(60);
    }
}
```

■ 击球

棒球游戏中的击球大致上就是确定在几个力作用下的细微平衡。你不能使击球太容易了，但也不能做得太难了。如果你觉得由于一些不可避免的原因肯定会背离难易之间的平衡的话，那就朝着玩家所偏爱的方向（指得是让击球容易些）“偏离”吧。构造一个打击力强的选手总比一个倍受挫折的选手要好，后者会让玩家十分厌恶地扔掉你的游戏。这并不仅仅适用于击球行为，而且，此类型思维方法也适用于很多其他类型的游戏。请相信，我见过很多人，他们通过使玩家感到茫然而获得某种程度的兴奋。我从来就不理解这种想法。下面是构造击球行为时需要考虑的几个游戏因素：

- **球的大小** 球在靠近你的过程中，其尺寸应当增加；
- **挥动速度** 击球手能以多快的速度挥动球棒，根据身体类型的不同而有所不同，体格较大的选手挥动速度较慢，但接触瞬间的力道更大；
- **挥动球棒的弧线** 击球手挥动球棒的路径并不是完全平行于地面的。挥动弧线能够帮助设置地滚球和高飞球的出现比例；
- **撞击速度** 选手设置的碰撞接触的速度有多快；
- **击球角度** 棒球离开球棒时的角度会对击球结果产生多大的影响；
- **擦棒接球的比例** 球擦棒进入接球手手中的出现比例；
- **出界因素** 通过调节增加或减少出界的次数；
- **平直因素** 通过调节增加或减少平直球出现的次数；
- **高飞因素** 通过调节增加或减少高飞球出现的次数。

为击球所写的脚本示例代码如下：

```
// 支持“击球”的JavaScript代码
function hitBall() {
    // 确定一下击球时会发生什么。
    // 下面的脚本只考虑击球的角度，还应该考虑的有球速和选手
    // 击球的角度，挥棒速度，球棒弧线等因素。
    if(ball.getAngle()>75) {
        ball.popupFoul();
    } else if(ball.getAngle()>60 || ball.getAngle()<20) {
        ball.baseHit();
    } else if(ball.getAngle()>50 || ball.getAndle()<25) {
        ball.doubleBaseHit();
    } else if(ball.getAngle()>40 || ball.getAngle()<35) {
```

```
        ball.tripleBaseHit();
    } else {角度在 35 度到 40 度之间
        //
        ball.homeRun();
    }
}
```

■ 接球

在真实的棒球比赛中，接球手是信息的绿洲，它给出了任何游戏比赛中下一步将要发生的事情。接球手能够帮助形成事件并创造出有利于他们球队的情景。到目前为止在绝大多数的棒球游戏中还没有遇到过这种动态情景，我们所见到的如此之多的接球动作还是很基本的。我们可以从下面这些因素开始：

- **接球半径** 接球手能够成功地完成接球动作的区域或范围；
- **漏球比例** 球跑到接球半径的边缘上但没有被接球手接住而成为漏球的比例；
- **掷球速度** 接球手以多大的力量将球掷向对方垒包以还击对方的盗垒企图；
- **默认姿势** 接球手应该以怎样的姿势应付不同的投球。

下面是用于调整接球手的接球区域的脚本范例：

```
// 支持“接球”的JavaScript代码
function doCatcher() {
    // “30”是根据游戏的准确度的高低确定的一个可以调节的数字
    if(catcher.radiusFromBallToMit(<30) {
        catcher.catchBall();
    } else {
        catcher.runAfterBall();
    }
}
```

在这个棒球例子中，你可以把这些条目中的任何一个作为构造脚本系统的功能的出发点。对于其他类型的游戏，你将会需要各种不同的以及特定的角色控制、敌方控制、非玩家角色控制、道具控制等等。这个过程出发点是相同的。为了构造游戏的可玩性，这个过程是从你想要实现的行为和技能的分类列表开始的。

ANDREW FORSLUND 访谈录



Andrew Forslund 是迪斯尼在线(Disney Online)的前任总工程师,他为 Disney's Club Blast (具有成千上万个会员的游戏内容预定服务)和 Disney.com 构造了很多成功的游戏。在迪斯尼工作时,他为网站创作了很多 Java 游戏,其中包括单人游戏、点对点游戏和客户机/服务器游戏,这些游戏包括《近敌》——Circuit Breaker、Bug Bot、《泰山丛林探险》——Tarzan's Jungle Adventure、Pop-Up、Rat's Domino、Aladdin Experience for WildTangent,还有很多其他的游戏。Andrew 在 Riverside 的加利福尼亚大学获得了他的 BSCS/EE 学位。我和 Andrew 是 Magnet Island 股份有限公司(www.magnetisland.com)的合伙创始人,这家公司是基于多路无线设备的游戏和娱乐内容的开发商。他还是一名顾问,他所感兴趣的领域包括分布式网络应用和网络服务。

TM: 游戏开发者在脚本语言选择上(JavaScript/JScript、Python、Visual Basic、面向对象 C++或者其他语言)为何缺乏一定的标准或者侧重?

AF: 这些类型语言的选择通常要涉及到程序员的个人爱好、他们熟悉什么样的语法或者以前的开发经验。这个选择有时还是由于一个组织内部的政策动机而强制使用某种脚本语言。另一个因素是公开使用的很多脚本语言都十分成熟,因此当选择一种语言时有很大的选择空间。实际上还没有一门专门为游戏开发的脚本语言,这也是为什么一些游戏公司在制作自主知识产权的脚本语言的原因。

TM: 对于通用的行为和道具的动作用的脚本编写来说,有没有一种语言优于另一种语言的情况?

AF: 现在我们用的大多数脚本语言,如 Visual Basic、Python 和 JavaScript 等都是高级语言,并且都已经很成熟了,用它们来完成这些任务的效果是旗鼓相当的。这再一次归结于程序员最熟悉哪一种语言以及哪一种语言他们能够快速入门。

TM: 人们都认为基本的编程熟练程度或者说深入程度对游戏脚本编写者是至关重要的,你有什么建议要给这些学生吗?

AF: 也许熟悉编程的最重要的一点就是自发地并自愿地去学习新东西并去钻研一门语言。确定一个你希望完成的特定的任务,比如说构造一个简单的射击类游戏,并且作为一个真实的产品从始到终把它完成,这不但是你学习编程的一条很重要的途经,也

是你向未来的雇主展示一下你的才能的资本。我们可以从像 *WildTangent* 这样的东西开始学习。这些类似 *WildTangent* 的东西提供了一个完整的脚本环境，并且你可以结合使用它们所提供的教程和网上的编程资源来构造一个完整的游戏，这将会给你以后的游戏开发工作打下一个良好的基础。

TM: 很多脚本程序员都只是简单地编辑命令行或者修改变量值。对于 workflow 方面怎样才能够保持这种直接对应关系呢？

AF: 一个很好的办法就是以某种资源知识库的形式来保存一个脚本的所有部分。脚本编程人员为了调节好某些东西可能会对一个命令行或单个变量进行很多次的修改，但是这些修改应该定期地登记到资源知识库中，而且要让其他开发人员也能得到这些数值。将这一点与不断地构造和不停地测试结合在一起就能够保障所有的“修改”都适用于系统，而不会产生不良影响。

TM: 工具都是很昂贵的，并且开发工具要冒一定的风险。大多数的游戏工具的功能都非常有限，其主要原因是不是就是这个原因？

AF: 随便拼凑出一个基本的工具，用它来完成某一项任务，这相对简单一点。但是要继续开发这个工具，直到我们认为它的功能好到可以作为一个产品的话，这就要困难得多了。在很多情况下，开发者由于其他方面的任务已经负重不堪了，就只能利用休息时间随便拼凑一个工具进行这些基本的编辑工作了。大多数的游戏开发预算和开发进度表中都没有为系列工具的开发留出足够的时间。

TM: 将一个拖放式角色系统捆绑到拖放式脚本系统，其可实施性有多大？

AF: 确实可以这么做，但是需要一个成熟的集成环境。很多情况下，当打算用一些高度成熟的工具时，却由于它们没有提供某种特性而最终没有采用，而这些特性正好是美术工作者或者设计者可能在他们所钟爱的动画软件包中用到过。当然可以这样做，但是要做得让人使用起来自然而简单，并且和现有的一些工具集成在一块，那是很困难的。

TM: 在游戏内容开发中，你是否认为工具系统是其中一个限制因素？

AF: 在游戏内容开发时，有一个平滑的工作流程显然是极其重要的。并不是说缺少成熟的工具系统就一定会限制游戏的开发；如果美工——设计者——开发者三者之间的工作循环能够有效地进行，可能会很少需要那些高级工具。如果开发人员没有以前开发游戏系统的工作经验，而且开发的内容上要求推陈出新，这时候工具确实就变得很重要了。这个时候，一个设计良好的工具对于此人来说，可能就意味着在开发时间上几个

星期的差距。

TM: 当预算中没有考虑脚本工具时你会怎么做呢?

AF: 现在可用的大多数脚本语言并不需要什么工具,而且可能只需要一个简单的文本编辑器就可以开发和调试。还有,经常会有一些免费的或者只需要付少量费用的工具,它们可以帮助脚本编写者以一种更简单的方式将更为复杂的脚本拼接在一起。

TM: 我将在游戏中看到的行为“逆编”成简单的脚本逻辑,试着多学习一些关于脚本的知识。这样做你看有什么意义吗?

AF: 绝对有意义。这又回到了那个问题上,即以足够大的动力去走更多的路程。观察这些行为并试着模仿这些行为是一个很好的练习,而且很有可能会增加一些实战的经验。还有,对这些行为进行评价并试着用几种不同的方式来实现同样也是很有帮助的。

TM: 运动类游戏脚本是最难写的。在一个英式足球比赛中,用脚本描述守门员的行为时需要考虑的两个因素可能是什么呢?

AF: 涉及到对场景的评估以及怎样做会更有技巧性,对这些东西进行编程要比其他类型的脚本编写,如触发器或者系列谜题,要困难得多。对英式足球中的行为进行编程,你需要考虑球离球门有多远、谁在控球、对方球队的其他球员在什么地方、球正在朝哪个方向运动以及从得分角度考虑这个球看起来有没有危险等因素。

TM: 在什么地方就应该不再进行脚本编写而开始编写程序代码?

AF: 编写程序,从编写一个游戏引擎的意义上可以定义为开发所有的独立模块,这些模块能够被其他人重复使用,比如存取屏幕、声音支持、对三维场景的通用支持等等。脚本编写部分是用来和这个基本的构架进行交互,以此来控制它的游戏可玩性方面的情况。脚本编写绝对不会涉及到复杂的数学,也不会和输出最大数目的多边形有任何的关系;为了控制所显示的内容,脚本是在这个构架之上进行构造的。

TM: 我认为设置标志位(true/false)、触发器和判断条件都是脚本处理的一部分。为什么只有极少数几个开发者拥有能够满足需要的图形用户界面的脚本工具(*UnrealEd 3*是个例外)呢?

AF: 这个问题追根究底就是在时间和预算上缺乏对这些工具的坚持不懈的投入。为了使设计出来的工具有用性和功能尽量地最佳化,其开发过程照样需要某些恰当的、认真的连续性的创造性见解。遗憾的是,考虑怎样才能最佳地开发一个工具所需的时间很可能在进度表中没有安排。

TM: 现在大多数的游戏内容都是完全依赖于脚本,你同意这种观点吗?

AF: 脚本已经成为大多数当前的游戏内容中的十分重要的组成部分, 尤其是对于投资数百万美元的大型项目, 而且脚本在开发中将会继续发挥其中心作用。对脚本的期望值和有效性随着项目的复杂性的增加而增加, 因为与带有诸如自动内存管理这样的技术的传统的程序设计相比, 脚本编程相对更安全。较小的项目, 比如移动设备的游戏, 或者是运行在只有少量内存而使得开发收到很大限制的平台上的游戏, 制作者可能会为了节省资源而不使用脚本。因此, 在脚本编程继续发挥重要的作用并且其应用持续地增长的同时, 还存在有一些不使用脚本开发的游戏。

TM: 你对所见到的哪些游戏行为 (可能是应用脚本编写的, 也可能是直接用代码编写的) 的印象最深?

AF: 像《帝国时代》——*Age of Empires* 这个游戏中的路径搜索和分组算法给我的印象最深。多个代理在一个环境中相互作用, 比如说一支足球队, 如果队员之间能非常协调地工作, 也会给人留下深刻的印象。使这些代理在一块为共同的目标而协调工作是一件很复杂的事, 而要做到让它们始终如一地协调一致就更难了。

TM: 对于这种行为你是怎样尝试和编写脚本的呢? 通常情况下你会规划这些行为的逻辑关系吗?

AF: 这要根据不同的行为内容和目标而区别对待, 但是通常情况下, 我是先从限定行为的最终目标入手, 然后依据该目标逐步推进。应该考虑尽可能多的情况, 而且在这些情况下什么样的行动是最优的, 一直到构造出一张相当完整的地图为止。接着对这些情况进行实战测试, 并对行动进行调整, 直到找出最好的解答方案。举一个例子, 让一个个体在观察它周围其他所有个体的位置、方向和速度之后再采取相应的行动。在足球比赛游戏中, 让一个前锋跑到中场位置, 或者让一个球员调整其占位, 和其他球员排成一排。

TM: 对于那些害怕脚本编程的技术人员, 你有什么好的建议吗?

AF: 脚本编程在技术上可能确实有挑战性, 但是克服对脚本编程技术的恐惧的惟一办法就是对它进行深入研究并设法掌握它。这需要动力和自律才能学到足够扎实的脚本知识, 从而扫除这些最开始的障碍, 除此之外还有很多的在线示例可以参考, 以及大量的脚本引擎可以使用。

TM: 哪些游戏给了你激励和启发?

AF: 我是一个游戏迷, 我喜欢那些让玩家进行思考的游戏。我最喜欢的游戏之一是《萨尔达》——*Zelda* 系列游戏, 对于我来说它将动作、探索、冒险和谜题求解完美

地融合在了一起。我对第一人称射击类游戏和没头没脑的动作游戏不那么感兴趣，但是喜欢那些由玩家控制着自己所处的环境，而且允许他们选择自己想要做的动作类型的游戏。

TM: 脚本编程能变得更加可视化吗？

AF: 或许脚本工具可以做到允许编程人员直接拖放脚本元素，并以一种可视化的方式观看拖放后对游戏产生什么样的影响。构造这样的工具极具挑战性，因为脚本和可视化显示之间的无缝连接将是一个很难解决的问题。脚本也许不可能变得完全可视化，原因是大多数的游戏都需要自定义的脚本，只有那些默认行为可以构造到可视库为大家所用。

TM: 你是不是对无线游戏（又称移动游戏）感兴趣？

AF: 到目前为止，我已经用了好几年无线技术应用程序接口（API）了，而且当我最终看到完全使用这些 API 的电话机面世时，着实让我兴奋了好一阵子。现在我们所拥有的技术实际上只能支持很有限的游戏，但是现已卖出的移动电话的总量以及在未来几年内将会卖出的数量表明，将会使移动电话成为游戏开发中最为流行的游戏平台之一。

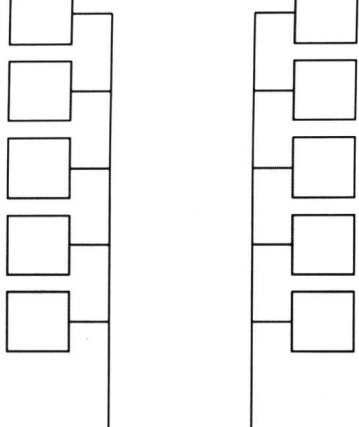
TM: 通常情况下，较小规模的游戏需要脚本吗？

AF: 这要依据游戏的类型而定，但一般情况下较小的游戏不需要编写脚本。较小的游戏通常是在资源很有限的平台上运行的，而且脚本语言往往要消耗大量的内存，所以它们对小游戏不太合适。也存在这样的情况，有些小游戏也许更适合用脚本编写，特别是当编写脚本就意味着能够从外部资源加载变量的情况。

MEGA 提示

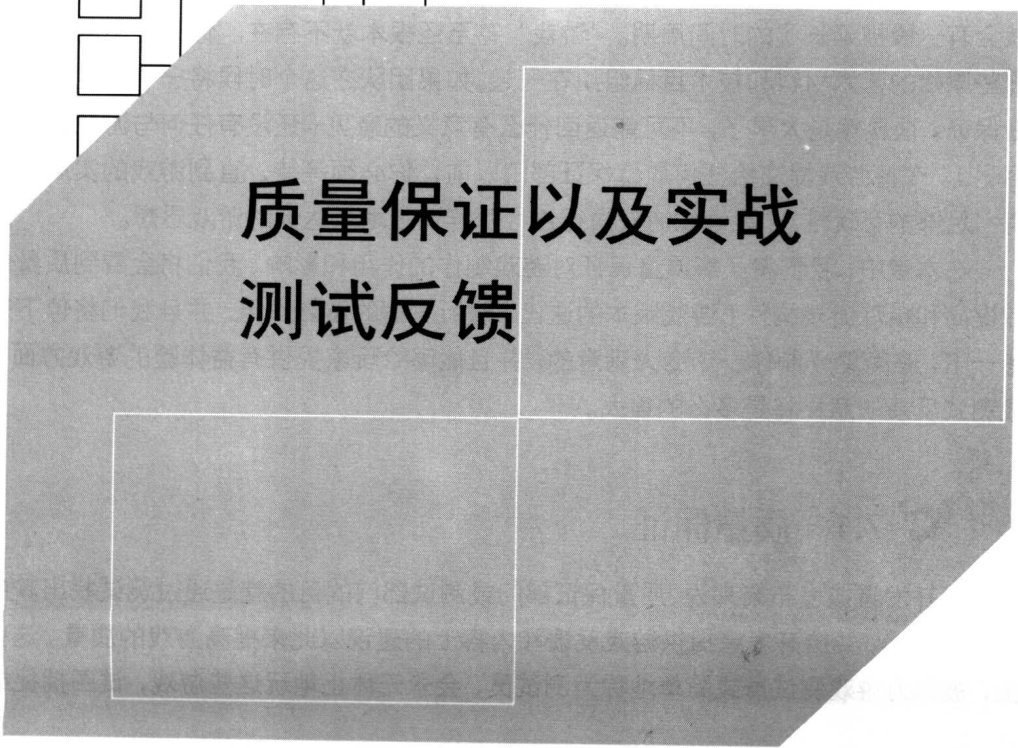
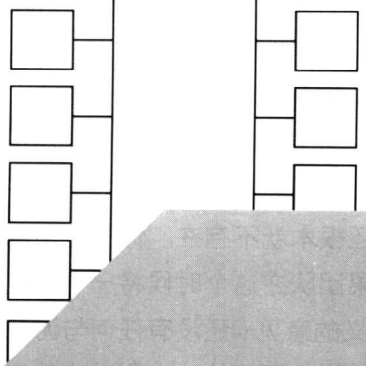
1. 要尽可能多地学习有关基本的编程、物理和数学方面的概念。Python 是一门很好的入门的语言（它还是免费的！），而且有一些游戏开发者就是在它的基础上构造自己脚本系统的。
2. 将你所喜爱的游戏中的角色行为和道具行为施以“逆向工程”。对于自己感兴趣的任何行为，都要问一下自己怎样用脚本来实现。控制这样的一个行为的背后隐藏着什么样的因素？为了构造这样的行为（比如说控制角色的详细情况、重力、武器选择和武器的攻击范围），你需要控制哪些特定的属性？

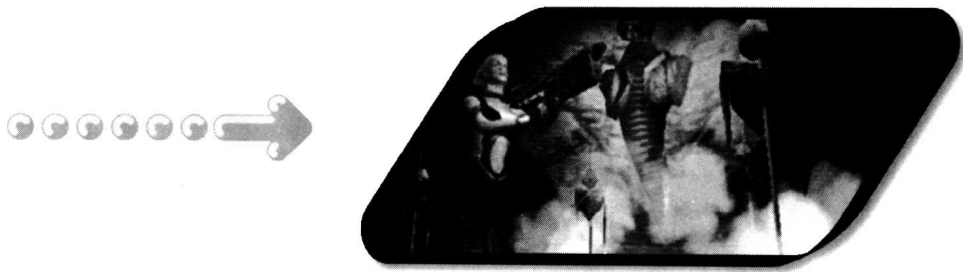
3. 留意一下你所喜爱的游戏中触发器系统的使用（就像本章所描述的那样）。问一下自己执行这样的事件或行为需要触发器的哪些属性。
4. 把通用触发器作为一个出发点来开发自己的触发器，以此来检验自己的游戏想法。你可以构造一个触发器，使它只有当一个特定的角色出现时才会产生大量的爆炸性的带刺的武器；另一个触发器则可以产生一些在附近搜索特定的角色并给该角色补充魔法能量的产物。设法想出一些使用触发器的新思路。
5. 在有了游戏设计思想的情况下，思考一下将单位的行为按照操作类型分门别类，就像前边的棒球游戏范例一样。试着定义那些你需要用到的特定的脚本控制，把它作为一项挑战来做。



CHAPTER

7





一旦你的游戏环境结合在一起形成了一个紧密的整体，并通过脚本、触发器、代码或者任何它们的组合体将所有的基本动作、行动以及受控的行为嵌入到环境中，你就应该准备好接收一些玩家的反馈意见了。对于将整个游戏、关卡、任务、地图、竞技场以及游戏情节提交给质量保证（Quality Assurance）或游戏测试部门的确切时间，开发者们有不同的工作风格和观点。

按常规，提交给质量保证的游戏的实质内容应当足够稳定并且功能齐全，游戏的可玩性（基本的导航、功能和操作——但并不是那么的完美）才能被认同。请记住，当你正在制作的游戏中没有哪一部分具有可玩性的时候，在你的游戏工程的早期开发阶段中就会有一段被延长了的时间周期。“游戏”甚至还根本就不存在。你只是刚刚开始要把那些原始的艺术材料和技术目标组织在一起。如果团队在这个时候将一个游戏提交给质量保证，很显然是太早了，不可能返回什么有意义的意见（还没有任何与游戏有关的内容呢）。在将游戏提交给任何质量保证部门以前，你必须等待，直到游戏的实质内容具有“足够的可玩性”，能够像游戏执行过程那样清楚地表达你的游戏思想。

在本章中，我们将了解质量保证对游戏制作的作用和影响。我们将会看到质量保证在提高和精炼游戏构件（游戏版本的迭代）时所发挥的重要作用，并且我们将停下来思考一下，在有助于制作一个令人满意的，并且能够给玩家提供有益体验的游戏方面，实战测试反馈的意义将是多么的重大。



7.1 质量保证

对开发者或发布商来说，质量保证部门或测试部门的目的是通过测试找出其中的缺陷和错误，并给开发者提供游戏反馈和内容上的建议以此来提高游戏的质量。这些人员，被称为游戏测试员或简单地称为测试员，会永无休止地玩这些游戏，试图捕捉或找

出游戏中的所有缺陷和错误,而这些缺陷和错误会对游戏体验产生负面影响。这是你的游戏投入发行之前的最后一道防线。如果你的开发团队认为质量保证的测试员对他们的游戏过于挑剔的话(至少这些测试员玩你的游戏是要得到一定报酬的),那就只能等待决不仁慈的公众对你的游戏进行强烈的批评吧。测试员是不会关心开发团队所遇到的特定技术障碍,人物的冲突,或资金预算困难的,这些也不是他们该关心的。他们拿了薪水就是来玩你的游戏的。要取得成功你必须交出一种引人注目的游戏体验。

一些人把游戏测试员的生活当作他们梦寐以求的生活。整天玩游戏还能取得报酬,这就是他们想要的生活!任何一个测试员都能告诉你,为了娱乐而广泛地玩各种游戏(这首先有助于你得到游戏测试这份工作)和通过连续地玩上14个小时(多数时候都是这样)来从头到尾测试一个游戏,两者之间有着巨大的差异。作为一名测试员,要求你立即就能充满激情地监测所有类型的游戏操作元素。

当要考虑游戏质量时,你应该认识到质量保证部门和每个特定的游戏之间关系的重要性。特别是,质量保证部门将会给你的团队提供大量的有价值的反馈意见,这些意见都是关于游戏玩得怎样了。尽一切可能去加强开发团队与质量保证部门之间的联系是一件极其重要的事情。你必须让这两个组织在任何时候都有保持有大开方便之门的双向交流。

交流的目的在于使双方来往的信息能够进行一种无缝的交换。通常情况下,游戏的改进正是依赖于这种信息的交换。测试部门应该得到最起码的尊重,在我们的心目中应该将他们当作为相同目的而奋斗的合作者:制作出伟大的游戏,这样的话,就能够做到这样的信息交换。当他们帮助你解决了重大问题时,就像是其他的开发者一样,他们也应该立刻得到回报。

并不是每一个游戏开发者都能够直接进入到质量保证部门。构造并维持一个完整的测试部门其代价高得令人望而却步。质量保证部门的形式在产业内也是多种多样的。大型游戏发布商(比如Electronic Arts[战场1942]和索尼公司)都有很多专职的测试员。而另一方面,一些独立的游戏开发者可能由于某些原因而没有全职的测试员。他们可能会有一些产品助理来帮助他们解决测试问题,但是在通常情况下他们都是依靠发布商来提供详细测试信息的。

有些情况下,即使是发布商自己也不愿意成立一个昂贵的测试部门,所以他们就与独立的第三方测试机构或测试服务部门签订条约将他们的测试要求交给这些人进行。这种服务正在世界各地开始流行。这对游戏开发者会产生 workflow 问题,因为对于游戏测试员来说,开发者们几乎是遍布全世界的。在这种情况下你就不可能简单地走到游戏测试

部门去查看一下出现的那个特别令人讨厌的错误，接着就返回你的工作室去修改它。如果测试部门位于地球的另一面，解决一个问题可能就需要发送几个电子邮件，通过很多企业的内部互联网，打许多电话，还要考虑时区问题，以及利用很多的动画文件解决其中的一个问题，而这个问题只不过是成千上万个问题中的一个而已。游戏开发者将游戏构件（*game builds*）（游戏的功能版本）发送给测试机构，接着就要对这些测试反馈进行处理并采取措施。

如果预算允许的话，最好是为自己的团队雇用全职的测试员。那样的话，他们和程序员、设计者的物理距离将很近，如果愿意，你可以定期的将新的游戏构件交给他们测试。如果预算资金较少的话，那就给新闻组发电子邮件寻找志愿测试员。请记住，总有人要处理那些反馈意见并与你所选择的测试员合作。很多游戏玩家都愿意无偿的工作，只要你的游戏发布后送给他们一份免费的游戏就行了。

■ 质量保证组织

质量保证部门是怎样把他们的反馈意见组织起来提供给游戏开发者的？一般来说，对于控制台游戏测试，在一些大型的房间内或在一些开放的工作场地上有很多控制台连接到录像机（VCR）上。在测试员测试游戏的过程中，录像机就会记录所有的动作。有些游戏错误很难描述或者重现，在捕捉这样的错误时这项技术就十分有用。将录像机磁带上出现错误的位置记录下来，并写成错误报告交给适当的开发者进行分析。对于个人计算机上的游戏测试，可采取类似的方式用带有视频输出的视频卡来进行测试。

■ 错误跟踪

当在游戏构造中发现一个错误时，就要把它详细地写成错误报告并指定一个错误标识号码。这个错误标识号码将那个错误与错误数据库中某个与它相应的描述细节联系起来。测试部门应用大量的、各种各样的应用程序来构造错误数据库。错误报告试图全面地描述某个错误并提供有关这个错误出现情况的尽可能详细的信息，因此开发者就知道应该怎样动手进行修复或解决这个错误。通常情况下，大多数的错误报告包含下面的这些内容：

- ▶ 错误的严重等级（较轻型、较重型、系统崩溃型）；
- ▶ 错误的类型（文本、图形、音频、可玩性、人工智能、界面、外围设备）；
- ▶ 错误描述（一个包含有错误出现的具体位置的书面描述）；

- 重复出现还是间断出现（能否重现该错误或者只是偶尔或间断性地出现）；
- 观察到错误的相关媒介物（录像机的磁带位置、磁带编号、动画文件等等）。

好了，你也许要问，所有这些都和游戏设计之间有什么关系呢？关系密切。至少你的开发和设计团队能够获得详细地描述出现的大量各种各样问题的错误报告。

我们再进一步地看一下这些错误的类型：

- **文本错误** 要求对拼写错误，单词冗长、杜撰，或者使用地方语言描述（将游戏文本翻译成不同的语言）进行修改；
- **图形错误** 可能是很严重的低层（位于支持代码的底层）错误。比如，角色可能突然失去了他们的纹理信息；纹理可能被扭曲并出现不合适的缩放比例；模型的图形显示可能不正确；或者某个动画动作失败（比如，角色悬浮在梯子的上方）；
- **游戏可玩性错误** 这是玩游戏的界面。例如，角色的武器在应当开火时没有开火或者是错误地发射了炮弹，某个角色开火两次后突然停止了开火，或者玩家无法从某一关卡中退出；
- **人工智能错误** 出现了古怪的、无法预料的和不希望出现的角色行为（比如，当对方射门时，守门员总是向着球门的一个角上移动）。路径探测问题（中间具有障碍物时角色怎样从一点运动到另一点）往往是人工智能型错误的一个主要来源。一个普通的例子就是，对方玩家戏弄怪物，它就朝着不通路的围墙或边界跑过去，而不是聪明地绕过去；
- **界面错误** 这是图形错误的一个子集，通常包括游戏的界面出现功能性的失败。有可能是收集的或拾取的物品没有更新或没有出现在相应的目录中，或者是拾取的时候在界面上显示了错误的物品；
- **外围设备错误** 与游戏外围设备的应用和功能有关，如方向盘，连接线和音响混合器；
- **安装错误** 涉及到在安装阶段出现的错误，包括从插入游戏光盘这一刻开始到成功、完整地将游戏安装到你的硬盘上；
- **引导错误** 与界面错误或游戏可玩性错误有关，经常是，玩家成功地安装上了游戏但却不知道怎样玩这个游戏。对于那些针对新手或者小孩的游戏，界面和提示必须十分简洁和清楚（没有哪一条游戏的导航信息被额外臆造的内嵌图形隐藏到游戏内部不易察觉的地方）。

我们在此处提到的每一种错误类型都能影响到游戏正常的可玩性,认识到这一点是很重要的。这些错误无论如何都要改正。然而,和这些正式的错误一起出现的是那些不太正式的错误,他们通常就变成了游戏出错目录中的“建议”。测试员会提出很多的建议的。很通常的情况是,他们提出的建议比开发者能改正的要多得多。要紧的是:对于游戏设计者来说关注这些建议的类型是非常重要的。

如果测试反馈信息开始表现出有个别的测试员在关卡难度、导航性、谜题求解、物品的位置、事件顺序等方面存在问题的话,这就到了要从设计的角度密切地关注这些反馈信息的时候了。你必须始终关注玩家向你报告的这些基本问题。这并不是说要关注测试员发表的评论如“我希望这个装甲的颜色更蓝一些”;而是要寻找这样的一些评论,如“每次我到了地图上水下设施这一节的时候,我都找不到回去的路”或者“如果不死几次的话,我就不能走完通向第二关的道路的四分之一”。这些都是潜在的危险信号。

就像游戏玩家的水平参差不齐一样,测试员在游戏技术方面也有所不同。最好是测试部门有多种层次的玩家,有极高技术水平的玩家,也有中等技术水平的玩家,还有相对来说是新手的玩家。请记住,随着玩游戏的人群不断的壮大,游戏披露与技术水平之间的日渐扩大之差距也相应地扩大了。你总是希望你的游戏能够有机会尽可能被最大范围的玩家所钟爱。

你必须和你的测试员之间构造起良好的关系并且依靠他们。即使设计者是从测试部门提升起来,知道怎样测试游戏,他们通常都不会是好的测试员候选人。他们对游戏进行过程的特性太熟习了,并且通常会认为其他人也达到了类似的熟习程度(但实际上他们没有达到这种程度)。还有,他们的自负通常是和游戏内容联系在一起的,他们反复检查、反复玩游戏的某些部分,而其他人有可能在这些部分中发现问题并感到麻烦。

■ 技术支持

设计方案的一部分在范围上应该扩展到技术支持问题上。对于控制台游戏,发布商会要求开发者提供一些信息(比如,游戏秘笈的代码,关卡的排练等)来帮助玩家,这些都是和玩游戏有关的一些信息。对于个人电脑游戏,游戏的补丁和游戏更新需要网络支持和发行控制。发布者也可能要求你利用扩展包来扩展你的游戏,包括具有新式功能、新式武器的新角色或新工具以及场地。

与玩家保持良好关系的部分内容涉及到要提供有用的即时的信息来支持你的游戏(像是定期的更新 1~800 个玩家顾问,提供切实的网络支持)。如果你打算给已经发

布的游戏增加新的特性和功能或者你预测到发布商可能要求你这么做,那么不要忘了你选择的设计方案对所需的技术支持的冲击。

维持一种可靠的客户关系有助于保证为你的游戏争取一些未来的玩家。很多开发者和发布商因公众认可的游戏而著称,因此,一旦玩家买了你的游戏,就绝不能让他们在信息方面搁浅。

■ 测试大型多人在线游戏

随着大型多人在线游戏(Massive Multiplayer Online Game, MMOG)的出现,人们提出了一整套全新的测试需求。既然将控制台游戏移到了网上,这些游戏也会提出新的测试挑战。单人或双人游戏的控制台游戏能够在最低限度的硬件(电视、控制台、录像机、数据库)上进行测试。在客户机/服务器游戏中突然出现了一套全新的测试要求和挑战。

外部测试实验室具有良好装备就能够测试控制台、个人电脑和掌上电脑的单人或者双人游戏。但只有极少数的实验室能够测试大型多人在线游戏。这就导致了很多的开发者/发布商,即大型多人在线游戏的构造者,将游戏玩家本身作为一种测试力量。很多流行的游戏都允许来自游戏社区的 *beta* 版游戏测试员(开发中的 *beta* 阶段,就是紧挨着游戏发布前的那个阶段)参加测试,他们在测试和构造游戏内容方面发挥着积极作用。这种方法能使开发者获得能支付得起的(免费的)测试帮助,还能把他们的游戏运用到大范围的真实个人电脑系统上,而这些系统都有众多的硬件组合(视频卡、CPU 速度、内存配置等等)。

这种类型的合作还要为 *beta* 版测试员们证明修改完后的游戏是值得玩的。如果所测试的大型多人在线游戏也位于定购之列的话,大型多人在线游戏的制作人希望 *beta* 版测试员能够成为游戏的定购者,而这已经不再是什么秘密了。

对大型多人在线游戏进行彻底的测试是一件很复杂的工作。你要测试的不仅仅是游戏的设计要素,还包括硬件需求、带宽,潜在的问题,错误的检查与修正,服务器超时和连接掉线。根据我们自身的目的,我们把精力主要集中在一些游戏设计要素上。

由于大型多人在线类型的游戏还是一种新兴的游戏,还缺乏对它深入的研究,致使很多设计上的问题变成了部分的社会问题。很多彼此陌生的人在相互交互,这是一件既令人着迷又使人感到灰心的事。那些大型多人在线游戏的构造者在开发这类游戏的同时,也揭示着开发这种类型的游戏所应遵循的规则和/或指导方针!在作出游戏决策时,

玩家的本领和单个玩家个人的能力对他们周围世界的影响将处于中心地位。

我们同时还有其他的目的。在保护其他玩家的游戏体验的同时，我们想让某些玩家拥有足够的能力以一种有意义的、充满乐趣的、有意思的方式来影响游戏的环境。有些玩家善于隐姓埋名在大型多人在线游戏中玩得发疯。他们不顾任何劝阻。他们试图毁掉任何其他玩家可能会喜欢的正面的游戏体验，并以此而发迹。作为设计者（和世界警察），我们必须认识到并对这种现象负责。我们职责的一部分就是提供一个有趣的、安全的、可靠的、令人沉迷的，且大家都喜爱玩的游戏体验。那种围绕着一个大型多人在线游戏世界而东奔西跑的能力将形成并影响（有些可能甚至说是指示我们作出决定）我们的设计决定。

■ 包含三个阶段的完整过程

完成一个游戏或软件产品，要经历三个重要的开发阶段，一般情况下称作 *alpha*，*beta* 和最终或“黄金”阶段。单独的一个游戏的每一开发阶段确切的定义多少总有些模糊。通常来说，*alpha* 阶段的定义是指一个游戏的基本特征已经完成，但存在有大量的错误需要修改。当所有 *alpha* 阶段的重大错误都已经被修改了，只剩下相对较小的错误时就到达了 *beta* 阶段。当每一个小错误都经过了修改并解决好了的时候，游戏就进入了它的最终阶段或则叫“黄金”阶段。黄金是指刻有游戏的最终光盘或磁带能够作为产品进行销售和复制了。这些阶段两两之间的时间长短通常取决于你的产品开发计划以及离产品的交付日期还有多长。这可能会是几个月或者大概是几个星期。

■ 书写测试计划

每次向质量保证部门提交一个具有实质意义的新测试的同时，都应该提交一个测试计划。测试计划应该尽可能详细地准确指出：在所提交的测试版本中，测试员应当密切关注游戏哪一部分的功能。在提交给质量保证部门的早期测试中，测试计划可能有点广泛，原因是你要测试能够想像到的任何类型的问题。随着质量保证阶段的进行，很多错误都已经修复好了，你希望测试队伍将精力集中在游戏中那些做过重大的修改的地方。你的测试计划应该详细、准确地指出要集中精力测试游戏的哪一部分（例如，特定的多玩家部件或者客户端 / 服务器连接问题，目录特性，角色能力的改变等等）。

你能提供的有关在什么地方可能会出现潜在新问题的信息越多，测试员就越能够集中精力来提供有价值的且节省时间的功能性反馈意见。

一旦测试计划开始进行（当游戏确实存在错误时，这个过程是必须的），就会出现几种非正式但仍在使用的测试方法：

- **“只管玩！”** 测试员就像普通的玩家玩游戏那样，只注意到游戏是难还是简单，是有趣还是无聊。他们将记录花费了多少时间来完成游戏中的每一关卡或每一个区域，在这一关卡或这一区域中他们喜欢或比喜欢哪些东西，玩的过程中对游戏的步调和平衡性感觉如何，以及花了多少时间玩完了整个游戏；
- **“看看能不能把它搞垮！”** 测试员尽量以设计者不希望的方式来玩你的游戏，看看能不能找到一种会削弱游戏的可玩性的情况，比如陷入到房间和关卡的某一部分而无法逃出来。他们还要尝试同时发射多种武器，将按钮疯狂地进行组合，并尝试其他异常的动作方式，企图使游戏由于输入的数据过量而崩溃；
- **“看看你能不能摆脱这种情景！”** 将测试员置于一种异常的并且通常是很难存盘的游戏位置来对其进行挑战，看看他们能不能“摆脱这种情景”。



7.2 实战测试反馈

对你的设计团队来说，测试反馈有几种来源。怎样理解这些反馈信息，怎样将它们反馈给设计团队，以及怎样在游戏开发中加以利用，这将是很重要的。在其他可能的来源中，实战测试反馈信息主要来源于以下几个方面：

- 授权人代表；
- 发布商的制作者和测试团队；
- 代表市场行为的集中群体；
- 开发团队的成员；
- 美术指导和其他的设计人员；
- 杂志和网站的预告；
- 你自己的独立测试团队。

请注意，授权人代表是指那些负责保护某项工程的图像和质量的人，该项工程为其知识产权所有者（IPRH, Intellectual Property Rights Holder）所拥有；换句话说，他们就是你游戏中任何内容的拥有者。举例来说，如果你正在制作一个明星游戏《辛普森一家》——*The Simpsons*，你期望 Fox 和 Matt Groening 这两个角色的模型非常地与众不同。

正像你看到的那样，实战测试反馈有几个来源。你应该怎样管理所有这些测试数据呢？好了，从实用的观点出发，能够跟踪每一个重大问题并相应地作出响应是很重要的。当然了，最终惟一实际有用的测试反馈类型是那种能够切实地改善你游戏的可玩性的反馈类型。你要像一个善于呵护小孩的母亲一样去关爱你的游戏。如果这些反馈看上去似乎不能改善你的游戏，那它通常对你的游戏是有害的而不是有利的。这就是为什么在任何时候你都需要对你游戏中那些使人心跳的东西保持一种清醒的想像（整个团队共同的想像是在产品开发刚开始时构造的，并且随着它的进化发展它应该还是保持它原来的样子）。

这一点需要牢牢记住，还有一些商业因素也需要考虑。最好是授权人对你的游戏感到满意，否则的话他们可以选择终止这种许可，这样也就终止了你使用他们的著名角色或事物的权力。这可以轻松地毁掉你的游戏。你的发布商可能不愿意出资开发一个其中没有著名的角色的游戏。有些时候授权人对于游戏、游戏类型或者玩家的期望根本就不熟悉，尽管如此，他们对于你所开发的游戏还是有着很大的权力的。（毕竟，你的游戏正在使用他们的那些受保护的角色和事物！）

如果你正在开发一个游戏，它不知为什么而不是独立投资的，那么你就有可能正在为一个发布商开发游戏。发布商负责开发的费用，因此作为一名独立的开发者，发布商就是你的实际客户。在任何商业活动中你都必须满足你的客户的需求。发布商对游戏的想像可能和你的开发团队对游戏的想像稍有不同或者是大相径庭。这种“想像分歧”在开发的早期就已经很好地解决了，但通常在开发中期会引起摩擦、内讧和利己主义的出现。

市场营销部门也对现代游戏的开发产生重大的影响作用。实际上，为提高游戏的知名度而进行市场预算可能会远远超过了开发游戏的所有费用。销售者有时会投入数百万美元用在电视和印刷广告上，以此来销售一个游戏。如果他们引入正式的或者是非正式的中心团体以获取玩家的反馈，并开始听取普通的意见或使他们失望的事情，你可以打赌，这些潜在的问题一定会通过正确的渠道返回到设计和生产团队中。

开发团队成员对设计指导通常也有很坚决的看法。这些建议是以一种激情的方式表达出来的，并且如果没有一个制作人立刻进行处理的话就会引起“设计内讧”和团队破裂，该制作人要能够在人和思想之间架起一座桥梁并要关注他们所关的问题。

艺术指导和其他设计者同时还参加了其他的游戏工作，他们在提供各种批评和建议的同时通常就表达了他们的意见。当从一个全新的角度利用新的见解来解决问题时，这些评论将是十分有用的。这些评论同时也会引起分心和分裂，它们会导致在开发团队之

间出现不必要的紧张关系。

当一个游戏接近它的黄金阶段时，游戏杂志和网站的作者、评论家和编辑会写很多的有关游戏的预测以供参考。市场营销者经常尽量让杂志和网站为即将出现的游戏制造声势以使销售量增加。对这些游戏预测的负面反应将会抢先一步反馈给设计团队。

最后，作为一件奢侈的事你可能有机会接触内部测试团队。对于你自己的个人测试和你已经考虑过的来自各个方面评论，这个团队将进行一次实际的检测。

既然知道了你的设计团队将接受来自各方面的玩家反馈，你就应该做好准备来处理所有这些问题。重要的是，处理这些问题时候应该考虑优先级问题。你应该对反馈回来的信息做一个优先级排序。开发者处理每一个反馈流程的方式有所不同，有些人将会或多或少地强调我们这里所提到的每一种类型。

作为一个开发者，你在处理被授权的角色和事物时必须给它们的授权人留下深刻的印象。你必须尽可能地与发布者之间构造起最好的关系。他们是你的客户，是关系到游戏成功或者失败的合作者，将来也是更多游戏开发的发起者。如果你不能交付游戏，他们会找其他的开发者。所有来自其余类别中的反馈、需求、批评、指导和要求都必须一个接一个的加以考虑。这里就是必须对反馈内容（和你的游戏可玩性方面有关的建议）的顺序和优先级加以度量并按顺序和优先级处理它们的地方。

■ 处理反馈信息

人们似乎要在通常的基础上要求一些不可能的事情。也许你根本就没有注意到，根据定义进入你的反馈列表末尾时，这种情况就非常明显。你是怎样处理所有的这种信息的？我利用微软的 Excel 软件构造了一个模板来输入和跟踪这些来源不同的反馈，当它们直接反馈给设计和产品团队时就输入和跟踪它们。这些反馈和错误数据库是分开保存的，除非给定的反馈本身就是一个错误。在这种情况下，就要写一个完整的错误报告并把它输入到错误的数据库中。

希望现在你已经认识到将会有数量庞大的实际信息反馈给你的设计团队。我已经建议你在顶层用一个具有优先级、分顺序的、分类别的系统来处理这些反馈。在这同时，开发过程中还存在着一个并行的更加密切的反馈循环。这个循环集中在实际开发团队内部所讨论的游戏反馈上。

开发团队的反馈循环

程序编写者、美术工作者、设计者、测试员、音效工程师、制作者和作家都在展现他们的才能，但他们也在测试游戏。他们看着游戏慢慢生成，一天天地一直等了好几年（有些情况下就是这样！）。他们想把游戏引导上正确的方向——设想游戏时就构造了的方向。这个过程本身就成为一个内部反馈循环或者团队循环，并且可能是功能最为强大的一个循环。如果一个团队试图要取悦每个人的随意要求，那么这个游戏就永远也完成不了——即使完成了，也不具备内聚性。

你必须要过虑掉所有毫不相关的反馈并使它与团队循环相协调。这不容易办到。下面这些类型的问题必须在团队循环层上加以处理：

- 条目的布置需要改进；
- 条目的功能不能像预想的那样工作；
- 内置资源的不平衡；
- 各种类型角色的速度和导航问题；
- 某项能力、功能、武器或者技能没有按预想的那样工作；
- 角色、交通工具或武器的物理过程很脆弱；
- 出现了沉闷的界面，难以轻易或快捷地进行操作；
- 错误的动画元素，比如从一个动作到另一个动作错误的运动过渡；
- 不连续的、难以读懂的或者不合逻辑的地图、竞技场和关卡布局；
- 敌人太多或者太少；敌人的分布密度不平衡；
- 玩的过程太困难或者太容易了；
- 玩的过程没有乐趣或者毫无吸引力；
- 角色或者装置需要更加平衡地补充能量和能力；
- 进攻、防守的水平太高或是太低（比如摧毁对手太容易或是太难了）；
- 控制器的反应“感觉”太慢了；
- 镜头移动时丢掉画面，产生摇晃或者使游戏玩耍的体验受阻/难以理解；
- 角色、效果、道具或环境特性在视觉上不具有吸引力；
- 需要更好的人工智能（角色显得太“傻”了）；
- 武器的类型或种类以及发射能力需要更加多样化；
- 应用控制器按钮组合或者鼠标/键盘做动作时感到太混乱了；

- ▶ 音效支持好像有黏滞性，不连续或者太假了；
- ▶ 头目角色的动作需要具有冲击性，使玩家感到兴奋，但实际上做得太简单了；
- ▶ 电影、音乐和开始与结尾动画已经超出了有效的资源范围；
- ▶ 试图将游戏特征和功能做得更复杂，而不是把简单的事情做好（这是所有人都容易犯的一个常见错误；将所谓的“简单事情”做好，你就能够制作出一个十分充实的游戏——控制器调节、简易的操作、游戏目标难易程度的平缓地“滑坡”或增加、精彩的玩家得分点等等）；
- ▶ 游戏的特征复杂，但从一个阶段到另一阶段的内容多少有点空洞；
- ▶ 游戏设计本身在范围或跨度上太广泛了并被迫出现不平衡的游戏情景。

这里所提到的这些条目仅是一个游戏开发团队通常情况下要处理的各种考虑要素与反馈意见的一个概述。这些条目中很多特定的例子都被添加到正式的错误中去了，而其他一些则认为具有专题性或概念性。尽管有些人试图来解决这些问题，但在任何一个给定的游戏的整个开发周期中，它们却继续使开发者陷入灾难之中或者是饱受折磨。

■ 听取反馈意见

反馈意见如果不能找到一条通道返回到游戏中那它就不会那么有用了。硬反馈是以切实的明显错误的报告的形式出现的，具有最高的优先级。如果你遇到一个能够使游戏崩溃的错误，遇到角色由于不明原因而粘到墙上或地板上的错误，或者遇到一个条目更新错误，那么为什么要解决这些问题的原因是显而易见的：它会破坏基本的游戏可玩性。

正是那些在形式上相对较“软”的反馈常常由于多种原因被人遗忘。这些较软形式的反馈包括：玩家对游戏整体难度的不同反应；特定关卡的导航问题以及对其喜好问题；针对很多关卡布局、步调和连续性的多少有点个人或主观性的反应。开发者们耗费了大量的时间来为游戏构造游戏体验，而很多人发现很难接受那些言词激烈、摇摆不定的批评意见。这完全是可以理解的。你连续花了五个周末的时间来工作，却只是听说游戏还没有测试好，并且还不能从测试员那里得到满意的反馈（或者，情况更坏，从其他一些不负责任的批评家那里得到了反馈意见）。就像我想要尽量表达清楚的那样，有些批评和反馈是有用的，但有些是没有用的。你必须要学会做一个好的反馈意见的过滤器。好的制作人通常会尽力保护他们的团队，使他们不被那些毫无效果的或者令人难受的“与任务毫无关系”的反馈意见所影响。

尽可能地，开发者本人不应该直接接受模式批评（这些批评中的大多数评论都是有

关同一个的潜在问题的)。模式批评有可能是一个潜在的危险信号，因此应该尽快地对它加以评估。如果有一些人对游戏体验表现出了相似的担忧或者受到了相似的挫折，你就必须要扭转这些负面的模式批评。如果在某一关卡情节中你所钟爱的某个想法确实没有获得成功，那并没有什么。这并不能证明你不是一名称职的设计员。游戏设计本来就是和实验、改进和冒险联系在一起的。更可能的是，这意味着你抓住了一次学以致用的机会，从而可指导你将来的设计思想。在查看玩家反馈意见的过程中你可以学着增加一些有关游戏的直觉。

目前尽我所知，现在还没有这样一套规则：如果遵循它就一定能开发出令人高度满意的游戏来。构造游戏的过程从头到尾就是一种实时的实验过程：作为一项工程，一项技术难题，以及内容的形式而言都是如此。你必须亲自绘画，反复摸索各种素材，并尝试新的思想，以此来得到你最终想要的游戏效果。这做起来绝不容易，并且会不断要求团队的支持。在某些情况下，你可能会有一个想法，它基本上能够在某些方面提高游戏的可玩性，但是由于技术上的障碍和资源问题却无法实现你的想法。

■ 实战测试反馈对日常设计任务的冲击

在整个开发周期中，一个设计团队将有助于构思下列问题的细节：

- 角色的历史 / 背景、能力、行为、成长曲线和相互作用；
- 环境细节、环境特征和地形；
- 游戏可玩性思想；
- 对角色、武器、能力和行为的挑战和定义；
- 电影、音乐、开始和结尾的故事情节。

这些类别中的每一项通常都将会因实战测试反馈所修改。甚至在游戏开发进行到就要送交测试部门以前，设计者通过将游戏定义信息送到开发团队以避免出现瓶颈问题。如果程序员正在等待基本的敌人行为细节，或者是美工用来构造模型和动画的角色描述模糊不清而且不完整，那就会产生设计瓶颈问题。设计团队一直在努力工作，力求避免这种情况的出现。

当实战测试反馈循环呼啸而至发挥作用时，设计者通常也必须要超负荷工作。游戏中有了新的或者改进过的功能，通常这是测试反馈的结果，这些功能可能需要进一步的定义并进行适当调整（例如，出现新的界面元素或新的角色能力）。设计团队将面临挑战要及时地找到解决以下这些问题的答案；如由于其物理构造方面的原因，角色还没有

完成测试或者是角色提出了独特的挑战；某些场景被认为是太容易了，太讨厌了，太短或者是太长，或者太难了，或者是对游戏整体步调的感觉变得越来越迟钝。

日常游戏设计任务的要素已从帮助定义游戏可玩性的各种功能转变成避免给队友制造瓶颈问题，变成了要在让人发疯的修改过程中有助于应付和指导将要出现的各种变化。



7.3 质量保证影响的案例研究

现在你对质量保证在修复游戏中的错误时所发挥的至关重要的作用应该是很清楚的了，他们的不但以报告的形式提出了显而易见的错误，而且还提供了很多为游戏团队所需的有关游戏可玩性方面的反馈意见。在这种意义上，质量保证通过提供玩家对于游戏细节的反应来帮助形成和影响游戏设计的细节，而这些细节通常都需要进行一些基本的调整和修改。再说一遍，质量保证将是你的游戏在面向广大公众之前的最后一次机会和最后一道防线。尽管测试员们已经花费了数百小时在测试上，但他们还是为日益逼近的游戏发布日期到来而感到兴奋，同时他们也知道其他的玩家也会陶醉于即将发布的游戏中，这时候，你就知道你的游戏中确实有一些很不错的东西。

质量保证部门还存在另一个重要的冲击因素。它还是未来游戏开发者的培育基地。很多游戏开发者（程序员、美工、设计者、制作人、音效工程师等等）就是从质量保证部门起步的。实际上，有些公司甚至要求每位员工都要在质量保证部门不间断地工作上一段时间。这是一项重要的教育。

通常，那些对游戏开发感兴趣并想以此作为职业的人都想知道他们怎样才能获得一份游戏开发工作。我想，在游戏产业工作的每一个人都会坦白地告诉你——这是一个竞争十分激烈的产业。这听起来像是陈词滥调，毕竟，根据定义有哪一个产业不是“竞争十分激烈”？是的，的确是这样的。但是大多数的其他产业是以另一种类型的基本商业模式来运转的。它们不需要为一系列的失败而遭受重大的打击。

游戏主要是那些年轻的、富有热情的、相对来说也经得起失败的天才们创造出来的。看着你周围的那些人花了好几年的时间在某个游戏上，而这个游戏却可能由于多种原因不能投入市场，这是一种多么残酷的经历啊。这种经历可以最大限度的测试你对游戏开发的热情。一种结局是很多开发者只在游戏产业中工作了很少几年接着就转行脱离这个产业了。游戏产业还要努力地解决这个问题，因为通常这都会导致经验丰富的、有创造能力的开发者流失而进入到与之竞争的产业中去。

开发新手可能还不确定在游戏开发中应该从事什么样的特定职业。质量保证部门是一个很好的起点。如果你想要增长你在程序设计、美工、音效和设计技能方面的经验，在质量保证部门的工作经历将使你直接获得观察一个游戏的整体内容的有价值的经验。它还可以帮你更加清楚地认识到你到底喜欢从事什么样的工作？

由于现代游戏的硬件已经变得如此的功能强大并具有兼容性，每个团队成员之间的技术和能力障碍也就相应地出现了。现在，开发者通常是找那些具有计算机科学和数学学历且擅长三维图形设计的程序员。美工者必须被证明其具有高超的传统图解能力和设计技术以及真正了解数字软件工具。

尽管有了这些准备，由于以前没有直接的开发经验，很多开发新手都还不能找到程序员或者美工的职位。这并不是说你刚出校门的时候就不要试图去找程序设计工作。说不定你会找到一个很好的职位。一般来说程序员还是很缺乏的。如果你还在继续选择的话，那就别害怕在职业道路上先做一个游戏测试员。在大多数的游戏测试部门里，你会遇到一些十分有趣、多姿多彩的、充满热情的，并令人振奋的同伴。请记住，很多公司就是直接从测试部门将开发者提升到那些理想岗位上去的，制作纹理或模型，编写脚本，甚至是编写程序。通过参加测试工作，公司将会对你有进一步的了解。一定要坚持，这样他们就会看到你的奉献精神，看到你的愿望和能力。

当开发团队和设计团队连续地与测试部门合作以此来最后确定一个游戏时，你就可能有机会直接碰到那些伴随游戏设计与游戏生产过程而出现的各种挑战与挫折。很多开发新手只是太理想主义化了。我们中的那些仅在游戏产业待了很短一段时间的人也就是这样的。不管怎么样，这确实会对游戏产业产生有利的影响。我们需要一些令人震惊的想法。

在测试部门工作将会使你成为一名更加出色的游戏设计者。简而言之，不管最终你决定怎么做，这都可能使你成为一名更好的开发者。甚至，如果你没有体验过一个游戏从开始到结束的过程，即使你的可视化设计技巧高得令人难以置信，并且你在顶尖的设计学校受过良好的训练，那么你的经历也是有所欠缺的。别担心，通过这样或那样的方式，随着你制作游戏的时间的增加，你会弥补这种经历上的欠缺。

当你在质量保证部门工作时，你几乎能够利用全天的时间来注意什么会起作用，什么不起作用，以及为什么会这样。这就是我成功的秘诀。作为游戏构造者，我们经常要知道什么东西有用，什么东西没有用以及为什么。这在你将来制作游戏时将是非常有用的信息。

即使你拥有成功产品设计,拥有无懈可击的设计方案,和一支经验丰富开发团队,你也无法保证一定会有一个畅销的、能够获胜的游戏出炉。构造一个成功的游戏是一项杂乱的、需要反复修改的、令人烦恼的试验过程。这就会直接导致出现“要求昨天的它更便宜”这样的矛盾。这通常是一项相当大的冒险。它并不适合那些软弱无力的人。每一个开发者都知道:要想切实的做好游戏需要时间和资源。概念上是这样的。而绝大多数的开发者是既没时间也没资源。

在技术方面忙得团团转,这就是所谓的游戏开发,这好比是你骑上马时才去发明骑术,你怎样安排游戏的确切发布日期,以此来赶上与游戏相关联的电影的上映时间?唉,你现在还没有发晕吗?你会的。

幸运的是,从某种程度上,在质量保证部门工作你就能够大致上了解所有这些是怎么发生的。不管是由于什么原因,如果你正在质量保证部门工作,尽量地利用这一段长时间吧。利用这一段时间锻炼或提炼你对游戏机制的认识,并利用这一段时间来构造你对游戏的直觉。如果你坚持选择从事游戏开发,在质量保证部门度过的这段时间将是物有所值的。

MELINDA WHITE 访谈录



Melinda White 是 *US Recruiting* 的负责人,擅长企业应用程序集成技术,任职于 *SeeBeyond* 技术有限责任公司,该公司是位于加利福尼亚州蒙罗维亚市的一个软件开发商。她在 *Recruiting* 工作的十年中,她为很多游戏开发者、娱乐公司和软件公司雇用了大量的技术员工。她拥有加利福尼亚州克莱尔蒙特的斯克里普斯大学的心理学和法语学士学位。我们坐在一起讨论在游戏产业中怎样寻找职业机会以及公司怎样寻找职位候选人。

TM: 为大多数游戏开发商雇用雇员的过程是怎么样的呢?

MW: 这要取决于开发商的规模大小,一些开发商有十分正式的雇用过程,而有一些则一点也不正式。像 *Electronic Arts* 这样的大型公司应用网络系统来处理投递的简历。当有一个空缺职位时,他们能够搜索整个数据库来寻找匹配的关键词以此寻找该职位的候选人。较小的游戏开发商则倾向于更多地在个体的层次上选择职位候选人,他们持有一些简历并且主要是雇用他们内部职员的熟人以及合作者。

TM: 作为一名游戏开发新手在努力寻找第一份工作时, 你认为最好是先从质量保证部门开始呢还是直接去寻找一个游戏开发的职位?

MW: 这实际上取决于你的背景。通常的情况是: 你可以先从质量保证部门开始, 而随着时间的推移你就可以进入到公司的任何岗位上了。事情也并不总是这样。开发商也在寻找一些人员, 他们能够具有非常专业的技术水平并且能够一进公司就独立从事开发工作。程序员需要切实的代码案例, 美工需要准备好引人注目的作品, 而设计者则需要准备好已完成的关卡并附上设计文档。如果你已经准备好了这些材料, 那就可以给自己一些时间去寻找一个符合自己专业的职位。如果你还正在努力地学习这些技术, 你最好先进入质量保证部门来展现一下你能做什么, 并从这里走向你梦寐以求的工作职位上。

TM: 你经常看到招聘测试员的信息吗?

WM: 这似乎是季节性的, 并多少有点零零星星。很显然, 当大量的游戏快要结束时就要雇用更多的测试员。我更多的是注意寻求软件工程师, 这只是因为需求的缘故。

TM: 你是不是听说过有很多的测试员就在游戏产业内部转到了其他的职位上?

MW: 是的。很多人已经转到其他职位上了。对很多人来说测试部门通常会成为他们的“试验场地”。现在的不同是大量的专业技术人员需要摆脱测试阶段而进入到其他种类的工作中。

TM: 质量保证部门的报酬和薪水问题怎么样?

MW: 很多测试员都是临时雇用的; 因此他们是按小时付费的(10~14美元)而且没有津贴, 尽管质量保证部门中也有一些人是全职的, 并且是具有固定职位并有津贴。与主流软件质量保证部门中的员工相比, 这样的报酬是没有竞争性的, 而且实际拿到的报酬多少还与你所在的地区有关。对大多数人来说这个职位仅是你迈向其他职位的一块垫脚石。

TM: 程序员、美术工作者、制作人和设计者他们的工资怎么样呢?

MW: 我听说过有一些本来在主流软件公司工作的开发者, 他们就丢下了工资跑来做游戏开发者, 接着又相当迅速地又跑回到了主流软件中去了。开发游戏的工作计划令人筋疲力尽, 但很公平地说这种情况在主流软件中也会出现。游戏公司往往倾向于最重视程序员。这种重视的范围非常广泛。根据个人的经验和专业技术的不同, 他们在任何地方的工资范围都在3.5万~10万美元之间。美工们挣得要少一点, 大约在3万~7万美元之间。要记住, 由于规模大小不同, 开发商、开发商/发布者等等的薪水差别很大。制作者可以挣到3万~9万美元, 设计者大约平均在3万~6万美元之间, 但话又说回来

了,其中有很多因素需要考虑。你可能挣到的要比我说的这个薪水范围少一些或者是多一些。

TM: 在现在的市场状况下,你是怎样为在游戏产业中工作而作出最好的准备的?

MW: 有很多东西显而易见且必不可少的,如尽可能多地玩你手边的游戏。如果你想成为一名程序员那就要获得一个计算机科学或者是数学专业的学位。如果你想成为美术工作者那就要获得美术学士学位。设计者也是一样的。越来越多的开发商都要求学位,仅是因为求职者能够做得这一点。求职的人很多,所以他们能够在其中挑挑拣拣。你需要证明你能够在具有挑战性的环境下快速地学习。

TM: 你已经为各公司雇用了为数众多的开发者,根据这些人的状况你是不是可以说他们在游戏产业中工作得很开心?

MW: 有些人是很开心的但有些也就不那么开心了。有些人在里面试了一段时间就走人了。有些人工作了很多年并且拥有了他们需要的一切。有一些真正的先驱者也离开了,这太不幸了,因为他们离开了他们自己构造出来的产业。尽管游戏产业增长得很快,但它还是一个十分年轻的产业。在很长的一段时间里,游戏公司都没能真正地知道他们应该怎样雇用雇员。它已成为一个十分令人迷惑的过程。坦白地说,我见到的很多雇用过程都无任何逻辑可言,但这在任何一个产业中也并不是什么不寻常的事。那些表面上看起来是最佳的候选人,他们也具有最多的相关背景知识,但由于一些政治上的,微不足道的,很无聊的或者是无法说明的原因却没能找到工作。

TM: 将整个游戏开发事项送到国外去进行开发,这种趋势不断地增长,你是怎样看待这种趋势的?

MW: 这是一个很严重的问题。它已经影响到这本书的每一位读者。可能正好本应是他们的工作现在却流失了。然而,这并不令人震惊。很多的产业,并不单只是游戏产业,由于财政方面的原因都走上了相似的道路。

TM: 作为一个每天都要雇用开发者的雇主,除了一份简历和相关的有用的经历以外你是怎样来挑选职位候选者的?是什么原因让你雇用了某些人?

MW: 一份简历并不能准确地代表一个人。其中还有很多因素会影响到一个人的成功与失败;比如,失业,经济状况,政治见解或者一个公司的行政机构。一个人的“软技术”以及面试时表现出的技巧在简历中体现得并不明显。这个人能不能很有效地和别人交流呢?他们能不能简洁地表达出自己的观点呢?他们的写作能力如何?他们能和团队合体吗?能领导一个团队吗?他们有文化认同吗?他们遇到压力时是怎么处理

的？如果一个人受过正规的大学教育，根据他们的学识，通常他们就会具有完成任务的良好纪律并且具有清楚地思考分析的能力。

TM: 当你走出校门寻找第一份游戏开发工作时，你会不会做一份甚至没有薪水的工作？

MW: 如果你能够承受得起的话那就这样做吧，因为这是一种经验。接着当然就可以在寻找下一份游戏工作时用上这些经验了。在游戏开发中是没有太多的实习期的，所以这些没有给你支付工钱的经验可能还是很公平的或者毫无坏处的。

TM: 你在游戏产业中任职很长时间了，你对此有什么看法呢？

MW: 你必须不断地更新自己的技能；不管你是一名程序员、美工、制作人还是设计员，公司并不总是支付教育费用以使你拥有最新的技术，因此你经常处于这样的一种位置：你必须自己采取主动。在这同时，你必须要跟得上那些即将出现的平台、工具和产业趋势，因为这些因素将会影响你自己的“市场推荐能力”——就像听起来那样粗俗，这就是现实。对待游戏产业中的人才流失和截止日期驱动的特性应该保持开放的思想并灵活多变。战场上的竞争是特别激烈的！你能做任何尝试以及使自己脱颖而出都是很有帮助的。这并不那么简单。仅仅是说你是门撒国际的一名成员和我想要讨论的东西没有关系。在你的背景中应该有自己可以度量的独特的一面来突出你。即使有先前的经验，这可能还是不够，因为这太具有主观性和特殊性了。仅仅为了挑选几个工作所要进行的竞争现实就是这样。

TM: 在艺术方面你有没有发现很多相似的艺术进入到了游戏开发中，尤其是在电影、电视和戏剧上？

MW: 确实发现了。存在有很多相似的部分。当你为游戏制作的艺术类型开始看起来像是在电影和电视上见过的艺术类型时，每天你都会看到更多的相似部分。这也是感觉的稳定性方面的事。电视和电影中的艺术工作已经过时或者仅能持续几个月。你是知道这种情况的，尽管，在游戏中也是这样。所有这些艺术类型都不稳定。如果你要寻求稳定，这个产业也许不适合你。

TM: 关于技术方面的多样性是怎样？积极进取的游戏开发者应当紧记要保持多样性吗？我自己通常是鼓励多样性的。

MW: 明确地说应该这样。理想情况下，你需要具有一整套广泛的能力，通过一时的注意你就能够获得这些能力。如果你想让工作进行的快一些，你就应该这样。如果你只是以一种方式去做同一件事情，你就会遇到麻烦。当很多的二维美工没有或者不愿意

转向三维时就会发生这样的事情。它为那些愿意转化的人创造出了机会。尽管我们已经看到一旦某个领域饱和后,需求就会缩小并且人们对于自己的评价和期望就会变得合情合理。在互联网炸弹时代也会发生相同的事情。

TM: 对于现在这些热情高涨的开发者你给他们的最佳建议是什么?

MW: 我将建议他们去一个好的学院或大学中完成四年的本科学位,因为他们将会拥有一生的基础,而且在漫长的人生中它将会给你提供更多潜在的赚钱机会以及专业成长,并且将使你在那些职业经理人面前脱颖而出,他们都尊重并/或者要求教育。要玩大量的游戏,跟上它们的发展,对待游戏要有自己的见解,要有特别的热情。

TM: 在这样一个竞争特别激烈的产业中你是怎么度过严重的经济危机的?

MW: 这又回到了要有一套多样化的技术并拥有一个构造在良好教育上的基础。尽管在经济萎靡时期仅有这些因素还不能够保证你不会失业,但在雇主要挑选的众多的候选人中你将会显得很突出。具有了多种技术和良好的教育,你就能合理地将你的背景知识应用到游戏产业中的不同角度中去。或者,你也许能够转移到主流软件中去。美工可能无法轻而易举地做到这一点,但是程序员能够做到,而制作人能够成为程序或项目负责人。

TM: 从事游戏开发事业的最好理由是什么?

MW: 如果你是一名程序员,假如你能够在游戏产业中坚持下去地话,我就认为对工作的满意是你从事游戏开发事业的一个原因。而且人们选择游戏是因为他们具有共同的爱好。这是一个年轻、动态发展的产业,它给一些人提供了一个学习使用工具以及进行商用方面的实习的机会,而这种实习是基于市场的并且充满了乐趣。这个产业通常还是一个能够自由掌握时间的临时性的环境。

TM: 游戏产业中你最不喜欢的东西是什么?

MW: 两件事——他们的职业习惯和商业本身。但是这些也适用于其他商业;尽管如此,游戏产业具有典型的政治性和流失性。在他们的招聘习惯方面,游戏公司通常会就工具和编程语言对求职的人有不切实际的要求,这些东西在市场很少见,或者他们要求求职的人对某些特定类型的游戏有着很深的理解(比如,需要聘请一位某运动类游戏专业或半专业开发队员来当他们的制作人)。尽管他们要求得很高,但还不失为一种伙伴关系。在我经历过的很多情况下,我并不认为开发者作出了很好的雇用选择。至于商业活动本身,雇用团队方面的情况是一个圆圈,有用时就雇一支团队,而当游戏完成或者是取消以后就没必要再启用这个团队了。这种使用资源的方式很不明智,作为一项长

远的商业战略也很不合理。他们也有可能请顾问,但在游戏开发中这是令人感到很新奇的想法。游戏开发工程可能由于任意一个什么原因而被取消,这最终会影响到所有参加该游戏开发的人员的事业。我们能够认识到:当游戏被取消的真实原因经常远非开发团队所能控制时,他们[游戏开发团队]对于游戏的失败是绝对负有责任的。你的身份是构造在你所成功开发或者开发失败的游戏上的。你仅仅和你最近一次成功开发的游戏是一样的好,假定这个游戏开发成功了。这个产业已经将最富有热情和想像力的游戏开发者逐出圈外,并且变得过分地以魅力为导向。最后,对于面向女性的市场(《芭比娃娃》——*Barbie* 和《古墓丽影》——*Lara Croft* 对于女性游戏玩家并不是健康的游戏)和在该产业工作的女性这一点还不具备应有的敏感。

TM: 在游戏产业寻找工作时也要关注其他产业的工作或者网络,这一点的重要性有多大呢?

MW: 这是很重要的。游戏产业并没有必要回报绝大多数的合格人才,因此他们会忽略掉许多极具天赋并受过良好教育的人,这些人对游戏事业怀有极大的热情并能为游戏事业的发展作出很大的贡献。应聘的成功在一定程度上与你认识的人有关。很多的职位根本就没有在广告中提到过。

TM: 对于一个正在寻找游戏工作的新手你给他的最好的建议是什么?

MW: 做好准备迎接游戏产业残酷的现实!你将会遇到很多和你有着相同兴趣的人,并且你将会和那些有自己的办事章程的人一起工作,和那些可能本不应该在游戏界的人一块工作。要跟上游戏的发展并更新自己的技术,要提醒你自己为什么你要从事游戏产业的工作,要做到灵活机动,并且要尽可能多的吸收信息。

MEGA 提示

1. 练习写详细的游戏评论。一定要区分出其中好的与不好的部分。尤其是要说明你将会怎样改进游戏来解决这些问题。
2. 即使你只是在朋友中间开发过一个小游戏,当他们答应你的要求给你测试反馈后,你还一定要他们给你包含测试方案的信息。他们想要的东西是什么?你希望他们将注意力集中在什么地方?作为一个设计者,你应该知道你实际的关卡或任务中哪一部分给人感觉比较薄弱。看一看参加测试的玩家是否同意这些观点。
3. 这一点看起来是显而易见的,但一定要记住:如果测试反馈对于你构造一个更

好的游戏有所帮助的话，你就一定要处理这些反馈并采取相应的行动。傲慢无礼地将这些模式反馈忽略掉，这将会使你的游戏付出代价。

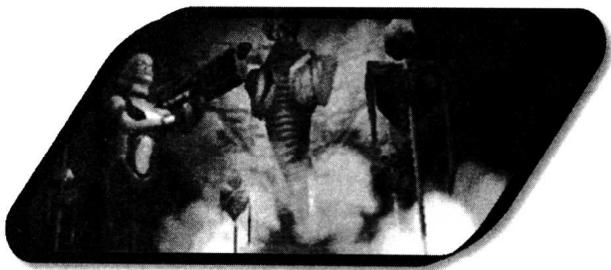
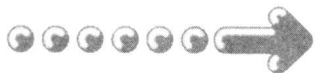
4. 通过给大型多人在线游戏的开发者提供帮助，充分地利用他们提供的开放性 beta 版游戏测试。这是获得早期经验的一个重要的方法，并且有些职业生涯就是从这里开始的。
5. 当你发现游戏中存在一个错误时，根据我们在这一章中讨论过的错误报告的标准详细地把它记录下来。证明你是能够书写出清楚的错误报告的，这可能有助于你在当地找到一份质量保证工作。
6. 当涉及到内容上的细节时，一定要让自己思想开放并忘记个人主义。作为一个设计者或设计团队，一定要知道质量保证反馈有助于充实的你的游戏内容。最后被反馈否定掉的设计思想并不能作为设计者个人失败的有力证据。每一个人都受过这种时刻的折磨，不管是个人还是整个团队。这是游戏开发的一部分。我发现，这些“失败”被证明正是学习和反思的重要时刻。



CHAPTER

8

大型多人在线游戏
的设计考虑



一种全新的游戏领域已经以大型多人在线游戏（MMOG）的形式出现了。这些游戏能让大量玩家在一个持续的环境中共同合作并相互竞争。正如你想像的那样，在电子游戏相对较短的历史中，开发这类游戏是任何一个游戏开发者所面临的最为复杂的游戏开发挑战之一。

到目前为止，我们已经从控制台和个人计算机上的单人和多人游戏的这两方面来讨论了游戏的制作。在纯粹的制作方面，MMOG 游戏“分享”了很多别的游戏制作所面临的相同的挑战，但是它也对所有的开发者提出了一些全新的挑战和知识。

在本章中，我们将介绍一下在构造这种特殊类型游戏的时候，所涉及到的制作和设计的众多问题和要素。现在，很多游戏开发者都在努力钻研这种类型的游戏。他们摸索着往前走。可以这样认为，很多从 MMOG 游戏开发者那里收集到的信息和经验都还不确定，还在不断地变化，并且每天还有很多信息源源不断地在“工作前线”产生出来。我们不要忘记 MMOG 游戏还处在它们的幼年时期。对于世界各地的开发者，还有很多的東西需要去学习。

现在大型多人在线角色扮演游戏（MMORPG）是一种很常见的游戏类型，并且是一种很流行的 MMOG 游戏形式。我使用了通用的术语“MMOG”是因为大型多人游戏中所面临的挑战和需要考虑的设计要素并不仅仅局限于角色扮演类游戏。



8.1 MMOG 的制作挑战

MMOG 游戏纯粹的广度、规模以及复杂度有助于构造出一个带有各方面的制作挑战和问题游戏开发环境。由于制作细节是设计观念在现实世界中的实现，所以当我们开始创作一个 MMOG 游戏时，最好停下来从一个纯粹的制作角度严密地思考一下：我们制作的目标是什么？

在下面各节中，我们将对很多这样的问题进行一个详细的分析。

■ 定义游戏

在MMOG游戏短暂的历史中，一部分很好的游戏有《网络创世纪》——*Ultima Online* (www.uo.com)、《无尽的任务》——*Everquest* (<http://everquest.station.sony.com/>) 以及《亚瑟龙的召唤》——*Asheron's Call* (www.microsoft.com/games/zone/asheronscall/)。这些游戏举行了大量的玩家体验活动，以此来吸引玩家，并让他们在持续的游戏世界中尽展自己的个性而感到激动。到现在为止，最为成功的游戏都是完全面向角色的。这些游戏提供的社会网络和社会特征任何其他类型的电子游戏所无法比拟的。对于不太熟悉这些游戏的人，它们提供了广阔的空间让单个玩家或者那些与他人合作的玩家去探索，主要集中在角色的成长、定位、能力的细微差别、功效和面向社会/权力的等级上。

■ MMOG 游戏玩家和控制台玩家的区别在什么地方？

一般来说，认为控制台游戏玩家和PC/MMOG游戏玩家之间还是存在着相当大区别的。其基本原因是：控制台游戏玩家更喜欢快速动作游戏和体育运动类游戏，而个人计算机游戏玩家则更喜欢第一人称射击类游戏、角色扮演类游戏和面向策略的游戏。这种观点是有重要的销售证据支持的。个人计算机的玩家使MMOG游戏获得了成功。然而，要把那些已被证明在个人计算机上获得成功的MMOG游戏要转移到大控制台上还是有办法的，通向控制台世界的途径，主要是将*Xbox*和*PlayStation 2*转移到在线世界之中。

在刚刚过去的那段时间里，玩家确实有转向喜欢控制台游戏或者个人计算机游戏的趋势，同时还有很多顽固的玩家对这两类游戏具有同样的热情。现在，年纪较小的玩家并不惧怕个人计算机的复杂操作，因此，对他们来说，没有理由只在简单易用的控制台上玩游戏。两种类型游戏玩家之间的分隔好像正在缩小，但是也不会完全消失。

你照样还能够遇到很多对平台有着强烈偏好的玩家。很多喜欢控制台游戏的玩家对于玩个人计算机游戏并不那么热衷，其中的原因是多种多样的。现在发布的MMOG游戏做得非常棒，个人计算机的游戏玩家很满意，而且也很容易掌握。但是，还不能确定这些游戏会不会转向在线控制台游戏。

从设计的前景来说，任何一个游戏，只要想设计成MMOG游戏，都必须更好地处理好两方面的问题：首先是游戏必须适合PC/MMOG游戏玩家，这一点是显而易见的，他们是你忠实的玩家；其次是要更多地准确了解在线控制台玩家会怎样玩这个游戏。

■ MMOG 游戏的饱和问题考虑

大多数的 MMOG 游戏使用订购服务模式。玩家购买零售的游戏、安装游戏，但必须每个月付费才能够继续玩这个游戏。这有一个明显的限制。每一游戏玩家能够承担多少份游戏订购？零份？一份到三份？你不能把玩家得到每一份单独的 MMOG 游戏所需要的订购费增加得太多。这就涉及 MMOG 游戏的饱和问题，因为只有极少数最为忠实的玩家会订购少量的基于订购服务的游戏。

可行的解决方案之一是：将几个 MMOG 游戏订购“捆绑”在一起，并以“捆绑”价格向玩家提供一种选择，让他们可以选出自己喜欢玩儿的游戏类型。在这方面人们思索很久，并得出了几种解决方案。为了支付 MMOG 游戏的开发费用，发布商急于寻找一些办法来增加一些面向订购服务的附加费用，但是这将有可能意味着只有少数几个游戏才能在激烈的竞争中生存下来。

尽管存在着这样的事实，很多的发布商还是试图利用 MMOG 游戏来吸引玩家。可玩的游戏由于各种各样的原因在质量上差别较大，而且很多游戏苦苦挣扎只是为了不陷入经济困难之中。有件事似乎是可以肯定的：没有人会真的相信游戏玩家会为大量地订购游戏支付费用。

■ MMOG 游戏的纯产品风险

开发、部署和维护一个 MMOG 游戏的费用可能会高得让你无法估计。其开发风险等级是非常高的。如果一个游戏发布商将数十亿美元倾注到某个 MMOG 游戏中，结果却没有吸引到一个玩家，那么这个游戏发布商以后可能就不会再开发任何类型的游戏了。看起来游戏发布商通常看重的是游戏的成功，比如说《无尽的任务》的成功，并希望复制出一个由成功的订购游戏所带来的源源不断的收入。然而，制作这样一个游戏，其风险和费用都是巨大的。只有相当规模的财政预算和团队联合起来冒这个风险，《无尽的任务》的开发者才可能成功发布这个游戏。这不是在一夜之间发生的。订购者也花费不少时间才找到并喜欢上这个游戏，而其他的游戏可能没有能力花费得起这种时间。如果他们不能很快地找到玩家，他们将会很快地宣告破产。

《无尽的任务》现在是展翅高飞的 MMOG 游戏的成功范例，但是要想成为能够盈利的游戏之一，其面临的竞争还是很激烈的。考虑一下 Electronic Arts 开发的《模拟人生在线》——*The Sims Online* (<http://thesimsonline.ea.com/>)。这是一个期望值很高的

游戏，它花费了相当大的一笔费用来进行开发，甚至还在《新闻周刊》——*Newsweek* 的封面上做过广告，目的是给玩家提供《模拟人生》——*Sims* 游戏（在个人计算机上十分流行的游戏）的 MMOG 版本。这是一项巨大的冒险行为，并且遭遇了错综复杂结局。

当开发费用达到这样高的程度时，运行你出错的空间是很小的，但是错误是难免的。我们也不要再毫不留情地批评开发商和发布商试图开发一个 MMOG 游戏了。不要忘了 MMOG 游戏的开发就是拿我们看不到的事物的发展前景来做赌注。试验和进步并不会导致可预测的结果。

很多因素都会影响游戏的成败，包括基于订购模式的 MMOG 游戏也会受到影响。这些影响因素中的一部分如下：

- **价格**：游戏和平台的价格是多少？记住“饱和”问题。
- **可用性**：对于玩家来说，平台的确定性或者说可用性如何？
- **内容的独特性**：这是一个很棒的游戏吗？它能够找到玩家吗？在众多竞争者中它能够占据一席之地吗？
- **熟悉程度**：玩家对这种类型或者风格游戏的熟悉程度怎样？

很多游戏发布商都十分谨慎，害怕过量地开发 MMOG 游戏。他们更愿意看到其他的发布商花费金钱并承担收益和损失。根据这种观察的结果，他们可能选择进入或者放弃进入该领域开发 MMOG 游戏。再一次说明，开发 MMOG 游戏是一种特别冒险并且昂贵的计划。一些发布商宁愿在比较成熟的游戏平台上更为轻松自如地操作（如在 PlayStation 2 平台上开发游戏）。

■ MMOG 游戏的费用和支持方面需要考虑的事项

是哪些风险领域以及花费导致了该种类型游戏的开发要冒如此大的风险？其中有很多的因子都需要仔细地加以考虑，下面几节中将讨论其中的少数几个。

服务器的安装与冗余

由于大多数的 MMOG 游戏都是基于客户端 / 服务器的模式（其中有一个游戏服务器作为游戏主机，而客户端 / 玩家都通过网络连到它上面），配置游戏服务器的硬件（个人计算机）来运行这些游戏通常要花费很多的费用。如果你的游戏要设计成能够同时连接几百个或者数千个玩家，将会需要多个服务器来运行和容纳大量的服务器端游戏实例，这是应付这些可能同时在线的玩家所必需的。

如果你打算在某个地区、全国区域或者国际上大量推出你的游戏，你可能需要在全国范围内分布服务器或者寻找一家提供有“区域性”服务器的公司，他们的服务器随时可以作为你的游戏服务器。像 Akamai 这样的公司 (www.akamai.com) 提供分布式服务器和相关的支持服务。假如说你要为美国的游戏玩家推出一个 MMOG 游戏时，就可以利用它们在整个美国范围内战略性地分布你的游戏服务器。微软公司将为 Xbox Live 游戏提供所有的宿主服务。微软公司当前拥有非常大的网络供应服务，仅以其全部容量的一小部分就能够容纳很多的 Xbox Live 游戏。

带宽支持问题

在游戏服务器和客户端（玩家）之间进行游戏数据传输需要开通巨大的带宽通道，这将花费相当大的一笔资金。这是很大一笔支持费用，必须把它考虑到维持游戏环境持续发展的费用中去。

硬件支持与维护

与基于客户端的游戏不同，比如一个简单的控制台游戏，MMOG 游戏需要大量的服务器硬件支持、升级、配置和维护。这其中不仅包含有硬件的费用，还需要足够的维护人员来使它稳定地运行。

数据库 / 后端的构造与维护

那些所有保存下来的角色和物品清单信息是从什么地方进入到 MMOG 游戏中的呢？它存储在一个与游戏服务器接口的数据库或者“后端”。流行的数据库解决方案包括运行在高级终端的 Oracle (www.oracle.com)，和更为便宜的 MySQL (www.mysql.com)，但在其他的解决方案中也有很多功能十分强大。游戏服务器定期地将这些角色信息发送到数据库中。这有助于防止玩家作弊，维护角色的完整性，以及并构造起一定的安全特性。开发和维护这样的数据库后端是一件很辛苦、需要深思熟虑、并且很耗费资金的工作。

质量保证费用

MMOG 游戏的质量保证是十分困难的事情。并不是说把新刻在光盘上的游戏演示一下就可以了。你需要的是一份深思熟虑的测试计划和相当数量的员工来帮助测试所有的游戏功能，这是使一个 MMOG 游戏达到稳定性要求所必需的。玩家社区的 beta 版测试者对游戏测试能起到很大的帮助作用，但是为了使他们的反馈信息发挥出应有的价值，就不得不对反馈信息加以管理并排出优先考虑级别来。

你想要做的最后一件事就是将一个有很多错误的 MMOG 游戏投放市场，假如因为数据库阻塞丢掉了游戏信息，并且系统崩溃了，而使玩家失去了他们来之不易的经验值，那么你将失去你的玩家。客户端 / 服务器游戏应用程序的所有方面，包括数据库接口、连通性问题、订购功能，以及角色或游戏数据完整性，都必须进行彻底地、系统地测试。很显然，这是一项艰巨的任务，至今只有极少数游戏界的专业人士在这方面具有丰富的经验。

如果你打算配备一支专业的测试团队，用来弥补由感兴趣的玩家免费提供的 beta 版测试反馈信息的不足，这也会在员工支持方面消耗掉相当数量的资金，仅仅是因为 MMOG 游戏的测试非常困难而且特别消耗时间。完成测试阶段并保证游戏某些程度上的可靠性将花费数千小时的时间。

世界范围内的客户服务员工

由于 MMOG 游戏的持续不断性以及实时运行，我们已经进入了“游戏管理员”的时代，这就使我们需要世界范围内的员工来为我们的游戏提供客户服务。游戏管理员就好像是游戏世界仲裁者。在生动的游戏世界中，他们扮演着警察的角色，试图解决玩家遇到的问题并帮助维持游戏世界的和平。玩家在游戏中经常会陷入到各种痛苦的斗争之中。一些玩家则试图破坏其他玩家的游戏体验，他们很“不辞辛劳”地去制造各种麻烦。作为游戏发布者，你希望为游戏的消费者确保一个充实而愉快的游戏经历。在 MMOG 游戏世界中，雇用游戏管理员或者世界范围内的客户服务人员来支持我们的游戏，这通常是十分必要的。这显然是一种全新的、特殊的人员需求。

安全与攻击问题

既然成功的 MMOG 游戏是依赖于订购模式的，这就会使你的游戏中存在电子商务的要素。玩家需要管理他们的账号，并为游戏服务支付费用。这就涉及那些敏感信息的交易，如信用卡号码等。MMOG 游戏需要充分的，有时候也是很昂贵的软件安全策略，确保玩家能够非常安全地支付他们的费用。而且当然会常常有接二连三的黑客攻击问题，他们试图搞垮你的游戏服务器，毁坏你的账号信息系统、窃取玩家的记录或者商业信息。

《儿童网络隐私保护法》问题

如果你的游戏是面向较为年轻游戏玩家的，显然还要关心这些年轻玩家的安全问题，他们是网上容易受到伤害的群体。对于那些面向美国玩家的游戏，你就应该咨询《儿

童网络隐私保护法案》——COPPA (www.ftc.gov/bcp/online/pubs/buspubs/coppa.htm) 指南, 它从 2001 年 4 月就已经生效执行了。



8.2 MMOG 游戏的构造要素与解决方案

现在我们已经看到了一些伴随 MMOG 游戏一起出现的产品方面的挑战。需要进一步考虑的问题以及可能的解决方案是什么呢? 在下面的几节中, 我们将介绍一下构造 MMOG 游戏时需要考虑的一些特定要素。这些要素中的任何一个对 MMOG 游戏的设计来说都是很重要的。在所有的游戏开发中, 特别是在 MMOG 游戏的制作中, 技术上的限制在确定设计定位时有着巨大的作用。如果游戏开发的技术支持结构像我们想像的那样广泛, 我们就几乎没必要详细考虑这些要素了。当然了, 事情显然不是这样的。技术上的限定因素通常很“残忍”地决定了游戏设计的细节。

■ MMOG 游戏的总体结构

如果我们考虑一下 MMOG 游戏的结构, 会有什么发现呢? 我们知道, 我们希望几百个玩家、甚至也许是几千个玩家一起连接到运行在每一个游戏服务器上的游戏世界中每一个实例上。我们知道玩家会将游戏加载到他们的个人计算机上, 并把他们的个人计算机变成游戏的客户端, 接着通过一个 ISP (Internet 服务提供者) 连接和一个网络云团 (通过 Internet 连接时位于 ISP 和游戏服务器之间的东西), 他们就能够和游戏服务器进行数据交换。图 8-1 以图形的方式表达了这种思想。

从图 8-1 中你能够看出, 网络云团和 ISP 之间存在从属关系, 游戏数据的传输或流动将会经过这层关系。当游戏数据从玩家的机器 (客户端) 上出发, 经由传输路径到达游戏服务器时, 发送的数据就会作为输入数据被游戏服务器感知并做相应的处理, 然后再返回到玩家的机器上。对于实时的游戏, 尽管这种物理过程是非常迅速的, 但花在数据传输上的时间还是相当多的。这种情况就引出了 MMOG 游戏设计中最重大的问题之一, 通常称作延迟 (*latency*)。图 8-2 说明了当数据在玩家的客户端与服务器之间来回传递从而产生延迟时, 是怎样在一个网络云团的内部进行从路由器到路由器之间的跳转的。在本章后面“延迟”一节中我们将进一步讨论到底是什么原因造成了这样的问题。

如果这就是一个典型的 MMOG 游戏结构, 那么从设计的角度出发, 我们就会知道我们的设计思想应该围绕什么东西进行设计或要考虑哪些因素。很重要的一点是要理解对

于游戏事件的更新（比如确定一个玩家发射的炮弹是不是“击中”了敌人，并播放结果动画和对破坏程度进行更新），你的消息处理系统必须穿过这条稍微有点慢而且错综复杂的高速通道从 A 点到达 B 点，接着再返回到 A 点。

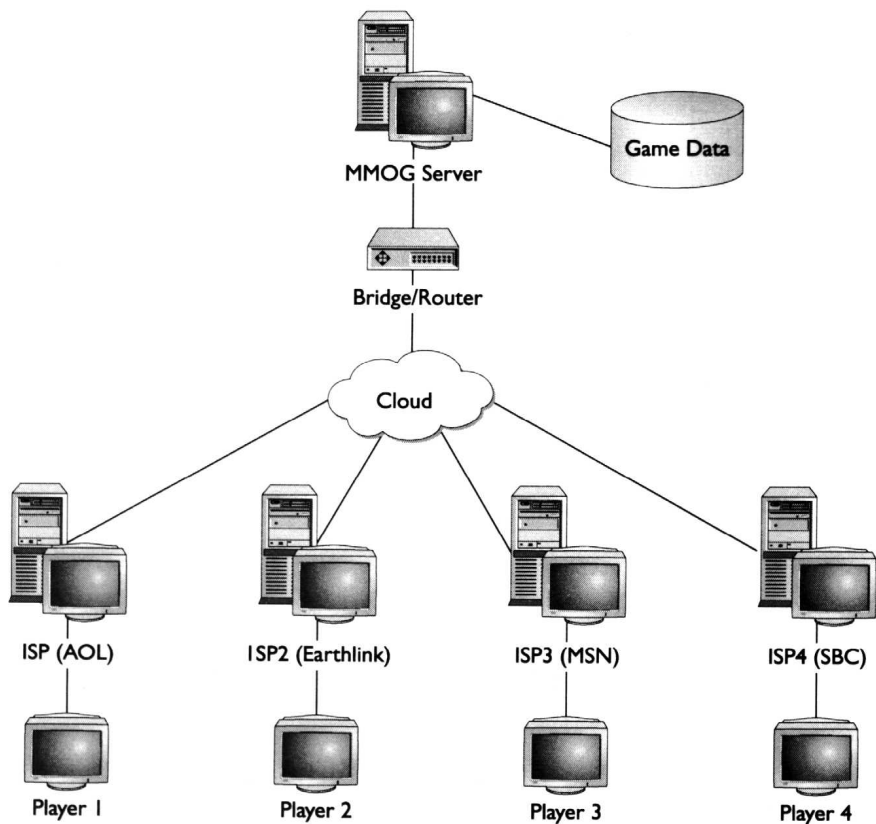


图 8-1 基本的 MMOG 游戏结构

很多开发商选择编写他们自己的 MMOG 游戏引擎、数据库和支持工具。有的开发者可能没有时间或者资源来编写一个完整的 MMOG 游戏引擎以及所有的支持代码，对于他们来说还有一些产品级的解决方案可以使用。LithTech (www.lithtech.com) 就提供有这样的解决方案。对于标准个人计算机和控制台游戏开发来说，LithTech 引擎是一种很流行的引擎方案，但是 LithTech 的员工们更进一步地提供了一种称作“LithTech Discovery System”的 MMOG 技术方案。这是一个完整的代码库，它带有支持工具，游

戏开发者能够在它的基础商构造 MMOG 游戏。

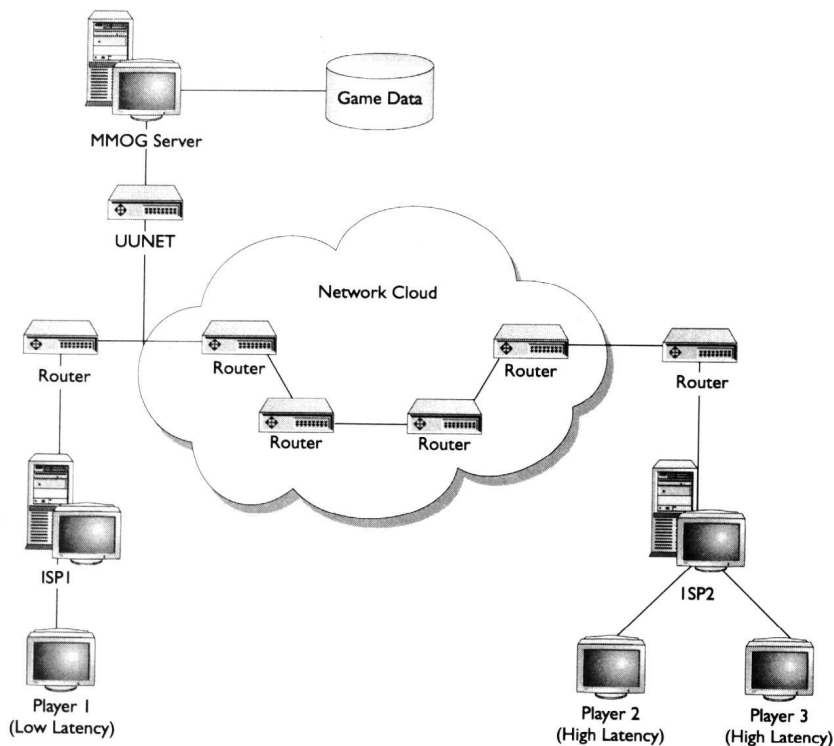


图 8-2 由玩家机器到游戏服务器产生的延迟

Terra Zona (www.zona.net) 是另一个引擎方案的供应商。开发者在应用任何一个这些解决方案的时候,通常喜欢先调查一下其他开发者的体验,这一点丝毫也不让人感到奇怪。“预”构造的或是产品级的方案很难考虑到开发者在构造多种多样的内容时会有什么样的内容需求以及需要什么样的工具支持。对于任何一个“未成品”的解决方案通常都需要进行大量的技术细化,这就导致一些开发者害怕使用产品级的 MMOG 框架,而断然决定从头开始开发他们 MMOG 游戏的所有内容。选择这样的开发路线,就像我们说过的那样,很可能会破坏游戏开发的时间限制和财政预算。

如果你选择自己编写 MMOG 游戏引擎,其费用和时间上的冲击太激烈了,以至于等到你所使用的技术成熟的时候,你的游戏内容的思想或者是游戏内容的方向已经由于其他竞争者的缘故而失去了它原有的吸引力。构造一个 MMOG 游戏需要认真地做好一系列

的准备工作、市场调研和技术评估。如果这里面没有商业秘密或者泄密问题存在的话，那么把你的 MMOG 游戏构造方案交给具有 MMOG 开发经验的其他开发商进行对等的检查，将是很有好处的。



8.3 MMOG 游戏设计因素

以上我们已经看到了技术上的考虑因素是怎样影响设计方案的选择的。现在我们应该看一下在构造、增强和维护 MMOG 游戏内容方面发挥作用的几个设计因素。这又是一个挑战世界各地的游戏开发者的最新、最大的技术和设计领域。我们期待的放之四海皆准的 MMOG 游戏设计规则还正在它的生长期。每一位开发者都在寻找解决设计问题的答案。（而且并不只是针对 MMOG 游戏！）每一位开发者都试图了解：怎样才能最好地利用平台的所有优势（并且完全避开其中的不足）向玩家挑战并使他们加入到 MMOG 游戏环境中。没有一个人真的想简单地重复同样的游戏思想，尽管这些思想可能在非 MMOG 游戏很管用。这些思想中的某些部分在任何情况下都与 MMOG 游戏平台极不匹配。

在本节中，我们将集中讨论一些关键的设计因素，它们会影响到任何类型的 MMOG 游戏开发。这些问题适用预所有的游戏开发，并不局限于特定的游戏类型。

■ 延迟

正如我们前面提过的，延迟是设计人员在制作令人感兴趣（在动作快速的意义上）的游戏时所面临的最重要的问题之一。这就是为什么由角色驱动的 MMOG 游戏或者 MMORPG 游戏能够运行得如此好的原因。必须注意到，成功地由角色驱动的 MMOG 游戏设计要点的是 MMOG 游戏平台能够做好什么样的工作，而不是开发者希望平台能做好什么样的工作。这就是敏捷设计。在这些游戏中也会有“战斗”发生，但不是玩家在快速动作游戏中推崇的那种类型的“战斗”。将其他类型的游戏扩展成 MMOG 游戏也会是什么样的呢？为什么我们看不到 MMOG 类的体育运动、竞赛、射击等游戏呢？从根本上讲，这是因为实时操作、判断、计算和延迟不能够很好地融合在一起。

由于延迟的问题，最好的 MMOG 游戏也会避免做“快速动作”的游戏特征，因为这太难完成了，并且即使完成了，其结果也不会令人满意。还要记住，你的游戏玩家或客户端是以不同的速度连接到游戏服务器上的，于是就会产生不同的从服务器到客户端的更新速度。这就意味着玩家 1 能够完成一个动作并跑到拐角处，而此时玩家 2 甚至还

不知道玩家 1 做了什么动作。这不利于构造一个公平的或者具有可玩性的游戏，难道不是吗？延迟问题经常会产生滞后，它让人感到心烦意乱，以至于使一个游戏变得基本上没有什么可玩性了。这是我们在本章中要讨论的关键的一点。

■ 模块化

由于大多数的 MMOG 游戏都使用订购模式，游戏开发商们希望能够在—个标准的基础上增加新的地图、关卡或者整个续集来升级游戏。这就需要模块化设计。准确地说你所设计的游戏和游戏环境必须是可扩充的。最为重要的是，你所用的工具包必须支持你以这种方式来制作游戏。如果你不得不取消以前的游戏信息细节（换句话说，就是图形更新）来增加更多的游戏区域，那么你将陷入麻烦之中。所以，你会希望你的游戏环境是彻底实现了模块化的。

■ 监视

你需要在你的游戏中规划并构造一个监视系统。你还需要一个离线测试环境来测试你对游戏的改动和模块的扩充，该环境要能够再现现实世界中玩游戏的各种情况（包括延迟问题）。

■ 工具支持

用于 MMOG 游戏的工具支持并不是什么奢侈品，而是一种需要。你所用的制作工具的功能完全能够促进游戏开发或者是搞砸你的游戏开发。如果你的游戏环境具备了监视功能，你所用的游戏监视工具应该能够对玩家提出警告，甚至可以消除错误。

■ 特殊事件

MMOG 游戏最酷的特征之一是在游戏内部运行特殊事件的可能性。同样，我们需要强有力的工具支持，来使得游戏管理者或者监视器具备这种能力。锦标赛、特殊的秘密泄漏，或者是与假日有关的事情，都能够在游戏世界中发生。这种类型的事件确实是游戏中的新事件，并且只在一些已上市的有活力的 MMOG 游戏中开始出现。

■ 步调与平衡

适当的步调与平衡是 MMOG 游戏中的一个重大问题。显而易见的是，由于订购这一关键原因，开发商希望你能够在游戏中长时间地处于忙碌状态。然而，这会、而且必定

会走向极端。在保持玩家活跃、高兴并订购游戏，和制作那些单调并且令人乏味的游戏任务，在它们两者之间，必须要找到一个平衡点。从我个人的观点看，我不希望花费数百小时让我的角色来完成那些最简单的任务，或者去进行那些永无止境的升级活动。由于MMOG游戏还要利用那些匆忙甚至仓促的活动来吸引更多的主流玩家，所以在一个较短的时间，可视化游戏进展的需求也将会增加。如果你愿意每星期利用40~60个小时的时间来玩一个MMOG游戏那就太好了，但是很多的玩家希望玩MMOG游戏的过程并不需要每星期都花费40~60个小时的时间。MMOG游戏设计若能够满足这种现状的话将会赢得大量的玩家。

■ 玩家中断 / 掉线

在MMOG游戏制作中，由于在玩家的个人计算机和游戏服务器之间失去物理连接而发生的玩家掉线问题是必须考虑到的。由于玩家突然关掉了个人计算机或者局部供电中断引起的突然退出游戏，就会影响到MMOG游戏中角色的数据以及角色之间的依存关系。如果你正在和其他玩家合作完成某些事情或者行动，并在其中发挥着极为重要作用，但是你却突然掉线了，很显然游戏会受到不利的影响，这种影响对你和你的伙伴都存在。你的设计团队与技术团队必须要处理下面的一些问题：

- ▶ 如果丢失了一个数据包，你的游戏怎么处理？
- ▶ 如果数据包接收顺序不对，游戏会怎么处理？
- ▶ 如果一个玩家的计算机突然断电了，你的游戏怎么处理？
- ▶ 如果在游戏过程中一个游戏服务器崩溃了，你的游戏怎么处理？
- ▶ 你的游戏怎样验证玩家正在运行的是正式版的游戏客户端软件，而不是给黑客带来完美目标以及非公平竞争优势的黑客客户端软件？

MMOG游戏所用的网络协议通常是UDP协议而不是TCP协议。简而言之，你做的设计必须要考虑到这些潜在的问题，要快速完整地保存角色的信息并对游戏中玩家与玩家之间的依存关系进行认真地分析。要注意大多数成功的MMOG游戏都是通过对玩家与玩家之间复杂的依存关系进行限制来处理这种情况的。如果某个玩家在和别人合伙去消灭一群怪物的时候掉线了，肯定会产生一些不良的影响，但是如果某个玩家在一个MMOG体育运动游戏中掉线了，将会对游戏过程产生破坏性的影响（比如，棒球游戏中投手在某一局比赛中投球时掉线了）。

■ MMOG 游戏的游戏机制

当涉及构造 MMOG 游戏的游戏机制时，设计者必须要经常考虑到那些比如延迟这一类“重大”问题所产生的预料之中的影响。你无法真正地解决延迟的问题，因此你被迫围绕它进行设计。再说一遍，这使得想要在 MMOG 游戏中产生令人满意的快速动作的游戏体验基本上没有可能了。你在一个 MMOG 游戏中想要做的任何事情都必须要考虑到延迟的问题。有些时候，某个团队会做得过了头，他们试图以最初看起来有可能解决延迟问题的一些想法来创作游戏。

那些曾经看上去可能实现的游戏思想并不总是能够获得成功，并且如果没有一个适当的技术解决方案的话，你就不要再在它上面浪费宝贵的时间和资源了，把时间和资源花在你所知道的、能在平台上实现的地方也许会更好。例如，游戏《无尽的任务》围绕着延迟问题做了大量的工作，甚至也为低速连接的用户做了大量的工作。作为学习游戏设计的一个参考，你留意一下《无尽的任务》中的游戏机制、动作和技能是怎样和 MMOG 游戏平台很好地结合在一起的。比如说，混战机制在 MMOG 游戏中应该尽可能地快速、流畅。这些战斗机制在任何方面都不会像控制台游戏中的斗士那样地流畅和迅速，但是，由于 MMOG 游戏平台本身的特征，你压根儿就别指望在 MMOG 游戏中出现那种样子的流畅和迅速反应。

我想说，游戏设计是游戏技术的最重要的伙伴，同时也是它的俘虏。游戏设计经常会被技术所“束缚”，不过我的意思不是消极的。我的意思是设计约束显然是一个对技术依赖程度的函数。在 20 世纪 90 年代早期，那些由于盲目的资金投入和天才们大量地涌入到游戏产业，很多人似乎都认为没有游戏做不到的事情，束缚它的只会是人们的想像力。人们似乎都忘了“技术束缚”的概念了。



8.4 MMOG 游戏与设计取向

我们都知道，MMOG 游戏在游戏的舞台上还是相对较新的角色。学习怎样为 MMOG 玩家设计游戏（并吸引玩家）显然是非常重要的。这不仅对于很多将要开发的 MMOG 游戏重要，而且对于这种类型游戏未来的成长同样也是非常重要的。现在我们把注意力转移到某些设计知识上面，很多游戏界的人士都正在为这些问题而奋斗。

➤ MMOG 游戏流派的发展；

- 非玩家角色 (NPC) 及宠物;
- 独立的 MMOG 游戏的力量;
- MMOG 游戏玩家的分类;
- 深层的社会因素;
- 当前面临的挑战。

■ MMOG 游戏类型的增加

如今最成功的 MMOG 游戏都是由角色发展来驱动的。打算玩面向角色的游戏玩家会有几种不同的选择。有一些玩家抱怨说：“看起来似乎只有面向角色的游戏可玩了。”我认为像出自索尼在线娱乐的《行星边际》——*PlanetSide* (www.planetside.com) 这样的游戏正在试图改变这种情况。我们还没有发现有哪个流行的控制台或个人计算机平台类型的游戏成功地进入到 MMOG 游戏世界中。限制 MMOG 游戏类型可用性的部分原因是制作任何 MMOG 游戏时所固有的那些技术方面的限制。另外，正如我们已经说过的，是巨大的财政风险。尽管存在这些原因，开发商还是一直在为昂贵的游戏寻找开发出路。问题就变成：你要围绕哪一种游戏类型制作你的游戏？为什么？

看起来 MMOG 游戏将成长为新的游戏类型，因为开发者们开始领会到怎样围绕着本章中讨论过的众多的技术依赖关系来制作其他类型（或者其他类型版本的）的游戏。只有当公众能够很好地接受这些游戏，并且当开发、升级和维护的费用能够承受的时候，这些游戏才能够生存下来。

■ 非玩家角色及与宠物

很多 MMOG 游戏为实现某些功能而使用了非玩家角色 (NPC)，但通常都是些毫无生气的、空洞的角色。宠物或者“小动物”却在游戏得到了生龙活虎地应用，而且很多玩家似乎都很喜欢它们。《半条命》——*Half-life* (非 MMOG 游戏) 的确将第一人称射击类游戏中应用非玩家角色更上一层楼。这些正是那些尚未完成或者正在开发的游戏区域，也就是你想让自己的游戏涉足的区域。哪一个 MMOG 游戏会像《半条命》将第一人称射击类游戏中应用非玩家角色更上一层楼一样，把 MMOG 宠物的应用更上一层楼呢？不要忘记你的设计焦点应当考虑到所有的竞争者都做得很好的那些东西。当你理解了为什么它们能够在游戏中发挥作用时，你就能够试着发展和提炼自己的游戏观念，制作出令人振奋的、大家都会喜欢玩的游戏。

■ 孤立的 MMOG 游戏的力量

到现在为止，我们已经知道了 MMOG 游戏的很多缺点（比如延迟）。我们也必须展示一下 MMOG 游戏的力量。按定义，其力量之一就是可以将大量的玩家集中到某一种类型的游戏世界中。你可能希望利用社会力量为你的游戏机制提供灵感，这一点也不奇怪。带有传奇色彩的游戏设计师 Chris Crawford 远在 MMOG 游戏出现之前就编写了游戏“*Balance of Power*”，而他凭借的是对人性冲突和戏剧性事件的理解，这些都有助于塑造以及传达游戏可玩性。问自己一个问题：你该怎样应用这些我们认为是理所当然的社会力量，使得你的游戏既令人兴奋又有趣？你该怎样应用社会竞争来制造冲突？

■ MMOG 游戏玩家的分类

MMOG 游戏还能够帮助消除以往那种刻板的玩家分类方式。各种年龄和性别的人们（从老太太到各个年级的学生）都投入了极大的热情来玩最为流行的 MMOG 游戏。在某些游戏中——MMOG 游戏可能就是一个很好的例子——商人们要想清楚地区分“游戏人”和“玩家”变得越来越难了。我预测它将变得一天比一天困难。这个事实本身就为游戏制作者开辟了新的蓝图与发展空间。

同样，试图为不明确的、成长中的或者处于变化中的玩家（换句话说，所有的事都是试图为了满足所有的玩家）创作游戏也会有负面的设计效果。试图取悦广泛的、各种各样的玩家的设想会削弱你的游戏能力，以至于那些本来可能会使你的游戏获得成功的情节也在这个过程中消失了。你不能简单地将游戏制作成为适合所有的人的游戏。请相信我，你需要将游戏设计成：对年纪较大的玩家来说，游戏不要节奏太快；对年纪较小的玩家来说，游戏不要太复杂；对十来岁的儿童来说，游戏不要太枯燥；对女性玩家来说，游戏不要太暴力，等等。

■ 深层的社会因素

在玩 MMOG 游戏时所涉及的深层的社会因素都是一些最有趣的、非正式的社会经历，而设计的取向就是围绕这些经历进行观察并创作。玩家对他们所扮演的角色在心理上有着极大的主权感。我想他们对 MMOG 游戏世界中的非角色物品（所挣得的交通工具、房子、财富或者地位等）也一样有着极大的主权感。MMOG 游戏中的社会性方面，是很多其他类型的游戏所不具有的，与那些内容相似但具有很少的或者根本就不具有社会特征

的游戏相比,看起来它能吸引更多的女性玩家。孩子们也喜欢聊天,但是要扮演一个真实角色的话他们还需要培养、教育以及与社会化。

有些 MMOG 游戏,比如《模拟人生在线》和《无尽的任务》,构造了一整套的虚拟经济。你可能阅读过有关 Norrath (游戏《无尽的任务》中的一个地名) 经济状况的新闻文章。在那里,通过收集虚拟货物并将它们变卖成实际的钱币,玩家真的能够每小时赚到 3.42 美元。这就导致了一种奇怪的结果,如果 Norrath 是一个真正的国家,那么它将成为世界上排名第 77 位的富国。我们仅仅刚开始看到这些种类的虚拟经济对 MMOG 游戏的影响。

借助使用与分配那些能够表征地位的物品,地位问题看起来能够很顺利地将人的情感与游戏思想连接在一起。通过提供能让玩家在游戏中“爬上”显赫位置的方法同样有这样的作用,因为某些玩家认为他们无法超越自己在日常生活中的地位。还记得我们在什么时候讨论过那些迷人的情感的重要性吗?在某些情况下,MMOG 游戏常常能够很好地把握玩家的情感,并引导他们情绪高度膨胀。

不能忽视获取代表游戏中地位的物品的能力。在现实社会中,人们倾向于无休止地对社会地位的评判,玩家往往非常乐意展示代表他们在游戏中地位的物品。这种倾向,从开发的角度来说,只是开发了一些表面的东西。在 MMOG 游戏中还有数量庞大的社会力量的现象有待于开发。

■ 当前的挑战

现在 MMOG 游戏面临的最大的设计学习的挑战是:试图准确地理解怎样才能最好地利用媒体为玩家提供娱乐,而与此同时也要满足商业要求。这当中涉及一个极为巨大的风险,就是任何错误都将是致命的。就像我们已经知道的那样,MMOG 游戏需要大量的资源。但是如果是大量的资源带来的是一个失败的设计思想,其结果对于游戏发布商来说绝对是一个致命的打击。

尽管这和商业电影之间有相似之处,但游戏发布商并没有在一个与之完全相同的商业模式下操作,该模式下,一次巨大的成功足以抵消许多次的失败。再一次说明,是与游戏销量相关的订购收入使一个 MMOG 游戏变成一个“巨大的成功”的。然而,像我们前面讨论的那样,有多少种“巨大的成功”订购模式能够存在下去下去?饱和问题又是怎样的呢?成功的电影你想看多少就有多少,可你不应该每个月都花钱去看这些电影。

MMOG 游戏必须做到将巨大的财政风险与一个感觉上“切实可行”的设计思想有机

地结合起来,在这一点上,要求 MMOG 游戏比其他任何类型的游戏都要做得更好。完美的设计思想很少等价于完美的设计实现。因此,MMOG 游戏为玩家提炼出了很多新的游戏方法,在这一点上它还算独树一帜。



8.5 体育爱好者或者体育团体与 MMOG 游戏

如果过去能够塑造未来的话,只要能够解决几个技术性的难题,运动类游戏最有可能在 MMOG 游戏的成长过程中发挥重要的作用。不管你喜不喜欢它们,它们都一次又一次地促进了控制台游戏硬件和软件的发展。把这种力量与所有主要的控制台游戏结合起来,到目前为止这些控制台游戏已经成功地移植成为在线游戏,而且你能够为运动类游戏的开发创作出十分有趣的情节。在 MMOG 游戏环境中,在线控制台上的运动类游戏开发肯定会有助于改革技术解决方案,这将有助于为玩家制作其他的,甚至更新的游戏类型。

由于各方面的原因(比如在游戏手柄和键盘之间控制器的差别),运动类游戏确实没有制作出与个人计算机上同样的效果来,但是推动个人计算机持续发展的第一人称设计游戏也没能在控制台上做得十分成功。

由于在控制台上玩运动类游戏的社区如此的庞大,很多发布商都密切地关注着这一块市场,当各方面的力量集聚在一起时,使得面向体育运动的 MMOG 游戏成为现实。我并不是无视开发道路上巨大的技术障碍,确实存在不少的障碍。但是对于所有主要的运动类游戏,如棒球、足球、篮球、曲棍球、高尔夫球和英式足球(还有很多其他的运动),游戏玩家正拭目以待下一代的运动游戏哪些方面会吸引他们。

最近,我有幸参加了一个游戏的开发工作,它似乎将成为世界上第一个 MMOG 棒球游戏。鉴于我已经说过的众多的原因,这是一项复杂而艰巨的任务,其中的每一个角落都隐藏着相当大的技术障碍。游戏产业常常因那些只求安全、基于套路的游戏而备受指责。这种评论不无道理。如果你仅仅制作那些安全、俗套的游戏,那你就不能创造出新的、能够赢得玩家的机会。这又是一件涉及平衡与执行的问题。你也不可能仅仅是因为某个 MMOG 游戏看起来很热门或者是很有潜力,就跑出来冒着巨大的风险去制作一个它。

图 8-3 是从游戏《顶级棒球在线》中截下来的屏幕快照,它展示的是通过一台游戏服务器把几个玩家连接在一起充当棒球场上的运动员角色。和我们经常遇到的情景一样,这里也有许多设计上的挑战因素有待考虑。当玩家在各自的场地等待行动时,你是怎样避免他们闲着没事儿?如果玩家正在投球的时候他或者她掉线了,你将怎么处理?

你怎样处理那些到处跑动因而扰乱了游戏秩序的不守规矩的玩家？当然还有大量的其他因素也需要考虑。



图 8-3 《顶级棒球在线》

这是我所知道的制作 MMOG 体育运动（任意一种类型）游戏的首次尝试。为什么没有更多的这样的游戏投入制作呢？其中的原因是多方面的。首先，最重要的原因是，当发布商承担的任务其技术难题非常大以及有可能要消耗大量资金时，他会犹豫不决。然而，单纯从游戏内容的角度来看，毫无疑问这个游戏尝试了革新。在极可能失败的情况下还要有革新的勇气，以及对棒球本身的热爱，就是最初吸引我参加该游戏开发的动力。不管这个游戏到最后是成功还是失败，它理所当然的会试出一些新的东西。不管结局是哪一个，我都感到很幸运。现在有多少个游戏是在尝试一些新东西呢？

我提到的所有这些仅仅是因为它们暗示了 MMOG 游戏在不久的将来将会出现的一种可能性。任何一种类型的 MMOG 游戏取得成功（当然包括体育运动游戏！）都会在某种意义上有助于另一个游戏也取得成功。大批的专注于运动类游戏的玩家正等待着下一代平台上的下一个《疯狂橄榄球 2003》——*Madden 2003* 游戏，等待着控制台游戏转向在线，随着它们的出现，一个运动类游戏的风暴即将到来。更为重要的是，这次风暴也将

会进入到其他类型的游戏中。

将传统的运动类游戏的爱好者、狂热爱好者、动作 / 战斗类游戏爱好者和科幻类游戏爱好者全部融入到个人计算机或者在线控制台的 MMOG 游戏大舞台上，这正是游戏发展所需要的。作为一名设计者，应该考虑一下——并仔细检查一下——这些充满热情的爱好者们所组成的每一个社区。请记住，在 MMOG 游戏设计中，我们一直希望能够利用社会力量来为各个社区构造出可玩的情节。

每一个社区的动态、其特定的影响都会给 MMOG 游戏世界带来广阔的空间。考虑一下你是怎样制作一个 MMOG 游戏，以将这些社区中的某一队玩家带到一个持久稳固的游戏环境中的。其中的社会问题是什么？又是什么使人们对它如此着迷呢？哪一种“快速动作”或者思考的行为是最引人注目或者最激动人心的？它们值得反复地玩吗？什么行为能从社会、情感，以及行为的角度最好地满足那些狂热的爱好者呢？

■ MMOG 游戏的机遇

随着 MMOG 游戏的成熟，而且技术也越来越可靠，制作这种类型的游戏将会变得更加容易。制作游戏的预算也将会降低到人们能够承受的范围，开发时间也会相对变短。MMOG 游戏算不上是沉浸式 / 耐玩儿类型的游戏。这种观念对于很多纯粹停留在概念层的人极具吸引力。概念世界与游戏技术的碰撞是 MMOG 游戏成长中最令人兴奋的一个方面。即将面市的游戏《星球大战银河篇》——*Star Wars Galaxies* 就是说明这种观念的一个极好的例子；它用到了很多在构造和支持《无尽的任务》游戏时获得的开发经验来构造游戏世界，其中用到一个迄今为止最著名且很受欢迎的虚拟世界。

那些高度发展领域的著名实体 / 知识产权拥有者 (IPRHs)，决定通过构造 MMOG 版本来丰富玩家的世界，并在很早以前就把产权提供给了控制台游戏制造商。他们这样做，有助于促进控制台游戏引擎的发展与革新。尽管他们对所有最初构造的游戏还存在一些犹豫并具有很大的偏好，但接触到可靠的 MMOG 游戏技术后将会为游戏制造商创造出大量的新机会。

不久的将来，广告很可能在 MMOG 游戏中发挥一定的作用。广告产生的矛盾是：一方面对不断上涨的资金投入感到不堪负重；而另一方面又渴望玩家能够“生活在他们的广告世界中”。这些游戏根本不需要订购模式。这听起来像是歪门邪道，但是你可以制作一个耐玩儿的《星际迷航记》——*Star Trek* 游戏世界，简单得就像是制作麦当劳一样。很多开发者对这种想法感到畏惧。理所当然，我理解他们为什么感到畏惧。就我个

人来说,我关心的是看到游戏制作中出现新的技术方案和机遇——不管它们来源于哪里。



8.6 挑战《顶级棒球在线》的案例研究

《顶级棒球在线》(UBO)游戏是一个创新的游戏。该游戏的开发者选择了承担制作一个完整的MMOG棒球游戏这样的巨大挑战。与很多其他相关软件的开发领域一样,在游戏开发过程中,任何需要实际创新的时候,绊脚石的出现几乎是不可避免的。作为一个开发团队,当你在失败中汲取教训,在成功中总结经验的时候,很难预测将面临的麻烦的真实本质是什么(就更难为一项能够节约费用的工程方案限定准确的时间期限和费用了)。毫无疑问,当进行创新时,你可以设想:即使是最好的分析与预测技术也将会“错位”,并且时间和预算资源也会受到重大影响。

与MMOG游戏的开发一样,新技术领域的探索也需要花费极多的时间、金钱和耐心(对于商业游戏开发来说,这里的每一项都是特定的恩惠)。为什么那么多的发布商都选择谨慎地开发MMOG项目,这些只是其中的少数几个原因。这就给小型到中型的开发商和发行商留下了发展的空间,他们可以通过努力工作创造出奇迹,并为他们自己赢得一些MMOG游戏的玩家。

由于没有办法进行事后全面的总结(因为游戏还在开发之中),我将花费一些时间来对某些已知的富有挑战性的困难进行评价,这些困难是《顶级棒球在线》游戏开发者在制作游戏的过程中所遇到的。

首先,这不是一个授权游戏。它没有得到美国职业棒球大联盟(MLBPA)的授权。就像你可能指出的那样,这些授权的费用高得惊人。由大型游戏发布商开发支付了高额的授权费,才使得他们的高画质的棒球游戏成为授权游戏。这样的结果是:《顶级棒球在线》游戏不能在游戏中使用那些著名的棒球选手、球队和体育场。很多玩家在玩一个单人或者双人游戏时可以接触到了这些东西。不管怎么说,这是一个MMOG游戏。玩这个游戏的乐趣就是让自己的角色在棒球大联盟中担任角色。在过去的一段时间里,如果你制作了一个丰富多彩的游戏,即使没有这些“授权要素”,游戏照样能够表现得很完美。但现在由于很多原因,事情看起来似乎不再是这样了。

众所周知,用成熟的客户端/服务器架构制作一个三维游戏引擎,让它能够同步处理实时的棒球事件,这已经被证明是一项无比艰巨的任务。解决不同玩家互联状态下的延迟是其中最主要的问题。当然,我们还必须设法把所有称为棒球运动的规则和玩法的

游戏功能都包含进去。

单纯的引擎开发费用也是相当大的,并且高额的支持与维护费用给预算细节施加了更大的压力。索尼公司的《无尽的任务》游戏和微软公司的《亚瑟龙的召唤》游戏都有索尼和微软公司强大的技术支持作为后盾。这就使一些独立的开发商在此领域中的竞争变得格外困难,而对于每一个开发商来说都是极具风险的领域,除非是能够拿出数百万美元作为资金底线的最大的发行商。

由于很多原因,其中包括费用的原因,游戏引擎的开发都是在韩国完成的。这就为预计在美国进行部署的游戏开发商带来了许多的制作方面的困难,比如语言障碍和本地化问题。要注意到将整个游戏送到海外进行开发呈现出一种不断上升的趋势。这可与为了尽量节省产品费用而将动画开发项目送到世界各地去开发做一个比较。

即使拥有经验丰富的开发团队、明确的目标、充足的资源,远程开发通常仍然是一种挑战。虽然它能够被成功地完成,但通常都涉及重大的挑战性难题,而当你制作一个全新的游戏时,负担就相对较轻。我不推荐对这样一个新类型的、具有创新性的游戏进行远程开发,即使是交给一个具有扎实的MMOG游戏创作经验的团队进行开发。

由于《顶级棒球在线》游戏开发者手头上有很多的挑战性工作,他们仅完成了这样一个有魄力的游戏的一些基础部分,就在工具的实现上陷入困难之中。没有强劲的工具,没有大量的持续的项目资源,游戏的运作就会陷入困难之中。大量必需的工具改进工作虽已纳入计划,但这些计划因更加紧迫的引擎方面的困难而没有得到缓解,被搁置到一边了。

至于资源方面,MMOG游戏开发需要充足的资源。毕竟,MMOG游戏是至今我们所见到的最复杂最深邃的游戏类型。我们必须面对的是:解决这些技术上的挑战将会花费大量的资金。极少中小型的开发商具有足够的资金实力来承担MMOG游戏的开发。紧张的资源将会像任何其他因素一样对游戏产生很大的阻碍作用。并且它已经确实实地阻碍了《顶级棒球在线》游戏的开发。

他们宁可承担所有的风险,同时也可能获得最大的潜在收益。《顶级棒球在线》游戏的很多发布细节都是很灵活的,开发商希望使用一种类似于《混乱在线》——*Anarchy Online*的方式自己发布游戏。自己发布游戏会有巨大的风险和创新(我们不要忘了*id Software*就是以这种方式起家的……),但却有一个明显的不利因素——想要你的名字迅速而高效地被潜在的玩家所熟悉是十分困难的。对于《顶级棒球在线》游戏,哪一种发布方式能够最好地照顾到各个方面,在这一段时间里还是没有找到答案。

参加《顶级棒球在线》游戏的开发能够得到大量的益处。如果你参加游戏开发的时间足够长，就会经常反思怎样来改进一个项目，而每一项都是一次学习的机会。这个游戏所表现出来的创新性吸引我去参加这个项目。而每一位开发者都会尽自己最大的努力来实现这个游戏。

对于开发者来说，开发任何类型的新游戏，尤其是为新的或者不熟悉的平台开发新游戏，通常都是一段很痛苦的经历，《顶级棒球在线》也不例外。我认为自《无尽的任务》和《亚瑟龙的召唤》之后，每一个进入游戏市场的MMOG游戏都经历过同样的痛苦。随着时间的流失，随着对平台的更加熟悉和团队知识的积累，这种痛苦会有所减轻。

《顶级棒球在线》邀请了美国职业棒球联盟的明星 Darrell Evans 作技术顾问，来帮助实现职业大联盟的游戏体验。这种内容上的合作以前也出现过，但是由于MMOG游戏需要为体育运动游戏提供一种前所未有的真实性，这有可能形成令人期待的全新的合作关系。

我通常对给游戏设计加入新的细节很感兴趣。这些细节可以使游戏更加好玩，并使游戏在市场竞争中获胜。我们应该不断地围绕一种游戏观念来寻找其实现方法，但是，我们也希望为游戏注入新的前沿技术。不管是什么类型的游戏，如果开发者始终将同样的细节“塞”到游戏中去的话，那么游戏产业实际上是不会有发展的，我们得到的只是版本更加鲜明或者更加漂亮、而内容却很陈旧的游戏。

纵观《顶级棒球在线》开发的整个过程，访问棒球选手（比如是击球手或投球手的全明星选手）对我们是很有帮助的。他们能够帮助你揭示游戏中出现的很微妙的差别，这些差别有助于形成游戏设计的细节，从而提供最好的游戏体验。他们没有必要明白为什么一个开发团队要作出这样的权衡和预留游戏动作选择，但是他们的洞察力将有助于给热情高涨的玩家传递玩儿棒球的体验。

《顶级棒球在线》游戏另一个出色的地方是它尽可能地延伸了MMOG游戏类型的类别。如果基于MMOG的游戏能够生存下去的话，它们必须在角色扮演游戏的核心上更进一步。已有的MMOG游戏没有任何问题——我认为它们是辉煌的游戏杰作——但是围绕它们所提供玩家的内容种类，还有很多的工作需要完成。

不管《顶级棒球在线》游戏是成功还是失败，它都不会改变开发团队尽其所能为MMOG游戏进行类型的延伸这样一个事实。创新是一件很痛苦的事情，崎岖不平的发展经历是很正常的。一些开发者认为：如果不能完成每一个你所接触过的游戏，那你对你所做的工作一定不是特别在行。事实胜于雄辩，与最佳制作及最佳销售的游戏的创造者

交谈一下，你将会听到另一个故事。如果碰巧他们心情好的话，他们会告诉你所有的那些没有开发成功的游戏。这些只是成长过程中的一部分。

在制作《顶级棒球在线》游戏的过程中所获得的大量技术能够以其他一些方式加以应用，这对于其他MMOG游戏的开发者也是一样的。他们可以重复利用这些技术来为其他类型的游戏或者相关游戏的制作作出贡献。

在一个MMOG游戏开发的曲折过程中，最重要的是所获得的开发经验，这些经验能够直接用到下一个开发项目中。事实也确实如此，对于非MMOG类型的游戏也是一样的，但对于MMOG游戏尤为重要，因为它们的范围极广。

如果你正在开发一个MMOG游戏或者正在头脑中做计划，你应该放放松一点；要记住，我们的情况都是一样的。一天到晚，我们所寻找的东西实际上就是游戏发展的新机会。MMOG游戏就是未来该计划中十分重要的一部分。

商务课程、计算机课程，所有这些游戏的方法我们都要准备好。没有什么比雇用一个人正好胜任这项工作的人更能令雇用经理感到开心的了。

MICHAEL WEINER 采访录



Michael Weiner从1989年至今一直在游戏产业中工作，最开始的时候他加入称为*Buena Vista Software*（现在的*Disney Interactive*）的公司，开始在客户支持和质量认证两个部门工作。在*Disney*工作了几年以后，*Michael*转到了*Sierra Online*继续工作（那时候还在加利福尼亚的*Oakhurst*）并在*Sierra*的子公司*The Sierra Network*里第一次担任了质量保证经理的职位。从那时起他在*Sega of America*，*Acclaim Entertainment*，*Disney Online*，*3DO*和*Hasbro*公司里担任了一系列的职位。*Michael*还担当过*Studio Search*公司的招聘主管，该公司坐落于内华达州的拉斯维加斯，他现在担任《顶级棒球在线》开发团队的顾问，在团队中，他的工作是保证正确的质量保证实施以及实施过程，同时还要管理远程测试团队。*Michael*曾两次被选为游戏开发者大会（2000年和2002年）的特邀发言人。

TM: 公司是怎样挑选游戏设计人员的？

MW: 大多数情况下，各公司看重的是经验和热情。有很多的“赶超崇拜者的人”都试图进入游戏产业，这是很好的——这使得游戏产业充满活力——招聘部门的经理寻找的是能把最大精力用到工作上的人……和那些已经有了一系列成功开发记录的人。而

这种成功是指曾经开发了至少一个游戏，而且是成功地开发了一个游戏，这就证明他们已经能够胜任某些工作，能够在一个团队中工作，长时间地工作，当然还证明更多的问题。

TM: 招聘人员应该怎样帮助游戏开发商呢？

MW: 我们给开发商提供了一种新办法来寻找最具天才的人。大多数的公司不会把太多的时间用在寻找人才的问题上，他们不得不依靠一些传统的方法，比如：招工表、展览活动、人才市场或者口头信息等。作为招聘人员，我们把所有的时间用在寻找那些有经验的开发人员上，把他们推荐给开发商。

TM: 在和招聘人员个别接触时，哪些应该做哪些不应该做呢？和招聘人员接触前应该掌握多少经验呢？

MW: 很明确地说，你不能做的一件事就是不断地烦扰招聘人员。大多数的招聘人员只有在招聘到人才时才会得到报酬。因此他们不需要你整天给他们打电话或者写信问“情况怎么样了？”“情况怎么样了？”“情况怎么样了？”，一个星期问一次就足够了。但请相信我，当招聘人员有适合你的工作时，他们一定会通知你的。

至于说，你应该具有多少经验，当然是越多越好了。如果你的工作经验低于两年，即使是你参加过大型游戏的开发，招聘人员也不会录用你。为什么呢？因为在游戏界中工作两年是最起码的要求！有很多在游戏界工作过更长时间的人，他们也可能在找新的工作（或者机会），因此当某个公司寻找有经验的开发人员时……你自己掂量掂量吧。

TM: 在招聘与解雇问题上，游戏市场看起来能不稳定到什么程度？

MW: 这是我能够想像的最不稳定的产业之一。我几乎每个星期都要和那些刚刚被解雇的人谈话。我在游戏界已经快十五年了，而我自己也被解雇过两次！这个产业最为注重的是“现在谁最红”。这样一种产业，各公司必须不断地对他们的位置进行评估和调整，以便保持在：即使不是在最顶层，至少也要能够支撑下去。这里面没有任何特权，即使对那些“大人物”也是如此。我们面对的是那些反复无常的浮躁的顾客，他们的经济状况不断地改变，技术也要不断改变，而且还要在所有的这些条件下拿出一个合理的方案来，不能出现一点差错……这是我所见到过的最艰苦的产业。

TM: 当一个游戏设计新手在这个产业中寻找工作时，你能给他提一些建议吗？

MW: 首先，要做好准备“该付出的就付出”。有很多很多人，他们虽然现在是在设计人员，但是开始时是在质量保证部门做测试工作，但也并不是每一个人都这样，有一些人就 very 幸运，但从质量保证部门开始起步是一条最保险的途径。另外一个优势就是：能够让你得到第一手的资料，知道事情是怎样运作的。到了最后，一个想成为设计者的

人可能会发现需要做的工作比他想像的要多，于是，就转而做其他工作了。谁知道会不会出现这种情况呢？一旦你进入到游戏界以后，就要多结交朋友，而且不要局限在质量保证部门。在不至于引起工作中断的情况下，多与开发团队中的人员交谈，问一下他们正在做什么事情。如果你有一些做好的样本，可以在午餐时间带上它，请别人提一些批评意见。但是不要去打扰那些正忙于工作的人、正在最后期限的逼迫下工作的人，或者是正疯狂工作的人，这个产业太小了，我们不能和任何人为敌。但是只要你花了时间，该付出的就付出，有耐心，有信心，就一定会成功！

TM：大多数的公司都需要游戏设计应聘人员提供哪些资料呢？

MW：这根据设计者申请的职位类型的不同而略有细微差别。对于关卡设计人员职位，他们通常会要求看看你构造好的关卡的屏幕快照；对于游戏设计人员职位，和那些设计外观感觉、地图、角色、动作以及游戏中其他所有的人，他们可能会要求你写作的样本。这可能意味着他们会给你一个简短的思想，看一下你是怎样将它在游戏中实现的。这些都是要确定你是否能胜任这项工作，而不仅仅是你要做这项工作。

TM：大部分被成功录用的设计方面的应聘者通常具有什么样的背景（教育程度和工作经验）？

MW：第一条，也是最重要的一条：游戏公司需要游戏人。这一点好像是不必多说，理所当然就应该如此。但是，当你知道我见到过多少从来就没有玩过一个游戏，甚至根本就不知道最流行的游戏是什么的人的时候，包括那些面试寻求游戏测试工作的人，你一定会感到吃惊。如果对游戏都没有热情，那你怎样才能对你将要完成的工作充满热情呢？更重要的是，他们怎么能期望你知道如何把游戏做得更好呢？当公司把数百万美元投入到一个游戏中时，需要你制作的是一个具有冲击性的游戏。至于教育方面，很多的学院和大学开设了游戏设计方面的课程。受教育的情况通常能够很好地说明你学过什么东西。所有的商业课程、计算机课程以及所有的精良方法我们都要学。雇用的人完全能够胜任该项工作，可以爬上更高的职位，因为他们已经在学校里学到了技术与商业部分的知识，没有什么能比这更令招聘部门的经理感到高兴的了。

TM：在游戏设计产业工作，我们应该做好哪些准备呢？

MW：先要玩一些游戏，就是要玩！并要阅读所有你能得到的游戏评论，密切关注每一个评论中的批评内容。没有一个称职的评论家会写一个评论说：“对，嗯……太好了！太有趣了！我喜欢这个游戏。”这根本就不是评论——这仅仅是个人观点而已！但是一篇真正的评论将会使你了解到：到底是游戏中的哪一部分使游戏这么成功！而这就

是你学习的东西——也是你要带到公司里的东西。接着就是自己写一些评论并把它们寄给发布商和各种网站。这可以使你的名字出现在那里，当你出去找工作时把它写到简历上，理所当然是一件好事。这就向招聘经理表明你是很严肃地对待这个产业的，并且正变得越来越投入。最后，也是最重要——就是聆听！当你听到有人谈到他们在工作中需要什么样的人时——把耳朵竖起来！如果你有实力胜任这项工作，就要试着和他们交谈！但是不要仅仅告诉他们你想要他们知道的事情，也要告诉他们你的事情！如果你不聆听，你就做不到这一点。接着，当找到第一份工作后（测试人员、设计者或者其他职位），你要继续聆听。大多数情况下，那些发表言论的人就是能使你继续前进的人。

TM: 你认为在不久的将来对游戏设计人员的需求会是什么样的呢？

MW: 我喜欢看到对设计人员的需求量增加，而且，我认为这是一件很自然的事情。当我们长期从事设计工作时，不断地把新的思想注入到该产业中也是很重要的。在不久的将来，对人员的需求将随着经济的变化而变化，经济形势越好，人们花费在游戏上的钱就越多，对游戏的需求量就会增长，也就需要更多的游戏设计人员设计更多的游戏来满足人们的需要。我们所从事的产业是一个被经济学家称为“可支配性的收入”型的产业。我们需要人们将“多余”的钱花在我们所制作的东西上——而这就需要一个良好的经济形势。

TM: 根据你的经验，在游戏开发产业工作有什么正面和负面的因素？

MW: 正面的因素——制作游戏可以得到报酬——还有什么比这更好的呢？我在这个产业内外都参加过各种工作，但只有在游戏公司工作时才体会到有如此多的乐趣。在这里工作的人都很伟大，其中的氛围通常具有创造性、活跃、令人兴奋并且其乐无穷。而在商店里看到你曾经参与创造的东西时，是很让人激动的。它正躺在那里等着别人把它带回家娱乐呢。与此相似，当看到为你游戏做的广告时，或者读到一篇相关评论时，都会让人激动。这通常都是一次冲击！

负面的因素也有很多。这还是一个很不成熟的产业。我这么说的意思是：我们一年到头都要继续学习怎样来运作这个产业，还要学习各种不断发展的技术。游戏公司通常要做好几年以后的展望和准备，并将数百万美元投资到游戏控制台、宽带连接或个人计算机系统上，而这些东西有时还处于方案规划阶段，这是一种疯狂的商业运行模式！尽管如此，就像该产业中那些成功的事迹所表明的那样，它能够一直运行下去，但是它有时候会使卷入其中的人付出巨大的牺牲，有时候甚至会最好的将人甩在后面。

MEGA 提示

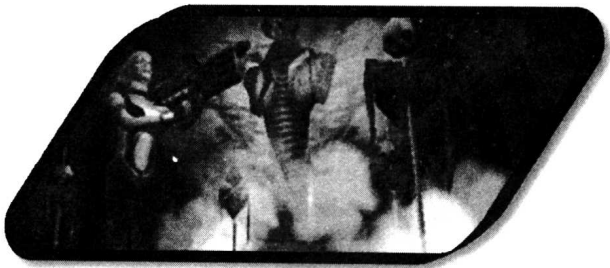
1. 在进行任何类型的 MMOG 开发之前，就要对你们所使用的技术基础在同行之间做一次彻底的评审（必要时可以利用顾问的经验）。
2. 对 MMOG 游戏设计所需的精确范围、规模和复杂程度做尽可能周密的考虑。设计阶段的一个基本的估算错误将会使你的开发团队在实施该设计方案的过程中陷入重重困难。
3. 要成为 MMOG 游戏分析师。密切关注你的竞争对手，在新闻组或者爱好者网站上不断地进行搜寻，仔细聆听玩家对现有游戏中出现的各种模式的反应。
4. 显而易见，作为一个刚刚进入游戏产业的开发商，要尝试开发一个 MMOG 游戏，很可能不是一个好的主意。应该先制作几个好的客户端网络游戏来测试一下你产品的销售渠道。
5. 要仔细地检查这一章中提到的关于 MMOG 游戏的所有需要考虑的费用和支持因素。如果不能满足这些需求，就可能会开发出一个很差的游戏，并大大减少你的成功几率。
6. 在假定某种类型的 MMOG 游戏的可玩性机制是否可行之前，要先和你的团队一起调查一下延迟和掉线（还要其他因素）的影响，以确定你的游戏思想的可行性。
7. 要记住，MMOG 游戏是游戏发展的另一条基本途径，而创新的过程通常都是很痛苦的，但最终还是很值得的。



CHAPTER

9

移动电话和无线游戏



另一个新的游戏开发领域出现在无线设备和移动电话上。这是一个崭新的游戏平台，在这里还有很多至关重要的经济因素尚未得到解决。尽管如此，如果能让人们认识到，到目前为止在游戏的制作和发布中所取得的（骄人的）进步的话，则无线游戏有着令人极为激动的广阔前景。我个人认为，我们应该把每一个可能的游戏平台都视为游戏未来发展的契机，而无线设备或者移动电话设备也不例外。

那些不愿意在公共场合投入地玩类似任天堂的游戏小子改进版（Gameboy Advance, GBA）这样的掌机（一块彩色的塑料盒子）而被逮个正着的成年朋友们，可他们会很乐意地噤里啪啦地去玩一个移动电话游戏。就是由于游戏小子改进版看起来像是一个玩具，而移动电话看起来像是一部电话，所以成年朋友们不会介意被别人看到他们正在使用移动电话。

为这些设备开发的游戏，对于玩家来说便于携带是一个显而易见的因素，它们还是壮大游戏玩家阵营的一个令人振奋的新手段。到本书的这一章为止，我们已经讨论了游戏控制台和个人电脑开发平台上的设计过程。这些平台依旧是最大数量的游戏开发者们开发工作的“面包和黄油”。然而，为了能够继续发展并保持繁荣昌盛，独立的游戏开发者们始终在不断地寻找开发收入的新来源。很多开发者的职业生涯以及继续进行游戏开发的能力都依赖于这些收入。

游戏产业发展的任何一个新机遇都不能被认为是游戏产业的一次衰退。不要忘了，尽管你那最流行的新的游戏机上拥有最大马力的硬件，但如果你不能以某些基本的方式向玩家提供一个能引人注目的游戏体验的话，你手中所拥有的仅仅是一个价格昂贵的技术“表演”。对于很多其他类型的娱乐媒体，情况也是类似的。

国际象棋绝不会通过一些向屏幕上冲击的多边形来激发我们的预测能力、让我们成长或者考验我们。换句话说，技术并不能决定游戏的可玩性价值。在我之前已经有很多人这么说了，而我仅能很保守地同意这一点：如果你希望知道怎样才能将你的一个特定

的游戏设计与玩家联系起来,你必须要试着去了解怎样才能在你的游戏内容之中迎合和激励人的基本愿望并与之周旋——即使你在技术上受到挫折也要如此。

玩家都有自己的想法和愿望,你的游戏设计必须要以某种方式来满足这些想法和愿望。玩家把这些愿望投射到你那充满悬念的游戏之中了,他才不管其中的运输工具是完全三维沉浸式的还是缓慢的二维精灵式的。

本章中,我们将要研究无线电话和移动电话上的游戏,集中讨论在这个新的游戏平台中出现的几个实践问题、技术要点以及设计因素。



9.1 日益逼近的繁荣

根据将来会有大量的无线电话用户的预测(根据你所问的人的不同,或者根据你相信谁的统计数字最为可靠的不同,这个预测结论可能会不一致),无线游戏开发的前景看起来很有希望。我想我可以很自信地说,通信中的无线电变革毫无疑问会带来一场游戏类型的变革。这场变革会有多大呢?时间将会告诉我们一切。我不喜欢引用统计数字或者使用条形图与饼形统计图,但是已经有数十亿潜在的玩家正在期待着一个伟大的游戏的出现。

有一些人认为无线设备和移动电话本身硬件的缺陷太多,所以不可能吸引众多玩家。这种批评似乎有点短视,因为设备的硬件始终都在快速地改进着,并在几年以后,很有可能会成为一个不再是值得考虑的问题了。

在下面的讨论中,我们所提到的无线设备指的就是像 Palm 系统(www.palm.com)或者像新式的诺基亚 N-Gage 游戏平台(www.n-gage.com)这样的硬件,这是一个游戏平台与移动电话的结合物。N-Gage 游戏手机是一个非常有趣的试验,诺基亚公司在它上面投入了大量的资金,而且在开发的早期阶段就有大批的发布商签约明确表示对该试验最起码的支持。图 9-1 给出了诺基亚 N-Gage 游戏手机的样子。

提及移动电话,我是指多家制造商生产的任何一部电话,只要它符合无线二进制运行环境(由美国高通公司推出,即 Binary Runtime Environment for Wireless,简称 BREW, www.qualcomm.com/brew2001/)或者 J2ME(www.sun.com)标准即可。

BREW 是一个操作系统,它允许你将游戏下载到支持 BREW 的无线电话上。如果下载下来的应用程序不需要访问网络,你就能够在任何地点使用它而不用占用你的通话时间。BREW 允许开发者使用 C、C++或者 Java 语言编制应用程序,并且相对于 J2ME 具有

运行速度上的优势。随着一些电信运营商们纷纷转向基于 BREW 的应用程序，BREW 开始获得了巨大的推动力。然而，对于很多小的开发商，高额的启动费用可能会让它们对 BREW 望而却步。



图 9-1 诺基亚 N-Gage 游戏手机

Java 和 J2ME 都是 Sun 微系统公司开发的程序设计语言，而且 J2ME 是 Java 的一个版本，是专门针对在像移动电话这样的小型设备上使用而开发的。J2ME 允许你将游戏下载到支持 J2ME 的移动电话上。图 9-2 是在 J2ME 开发环境中进行编程或者说制作的一个解谜类游戏程序的示例。

其他的开发选择包括为 Symbian 移动操作系统 (www.symbian.com) 和为智能手机系统 (Smartphone) 开发的应用程序。这两个系统运行的是微软 Windows 的移动版本。图 9-3 给出了一个目前已在英国上市的智能手机。

移动信息设备描述 (Mobile Information Device Profile, MIDP) 技术规范也值得一提。它可以用来告诉你的应用程序当前正运行在什么类型的设备上。MIDP 能够为移动应用程序——比如游戏提供所需要的基本的应用功能；这些功能包括用户界面配置、网络连接、局部数据存储和应用程序生命周期的管理等。所有这些控制功能都封装在了带有一整套 Java API 接口的标准 Java 运行环境中。

令人兴奋的是我们看到，在开发者们才刚刚开始为这些设备编制游戏，而一些有远见的公司已经为在这些无线设备上进行三维开发做好了准备。当前在这一领域中处于领先地位的是 Fathammer (www.fathammer.com)。目前这一代移动电话的处理能力还不足以处理三维应用程序，因此这些硬件还需要一段时间才能跟上软件的发展。

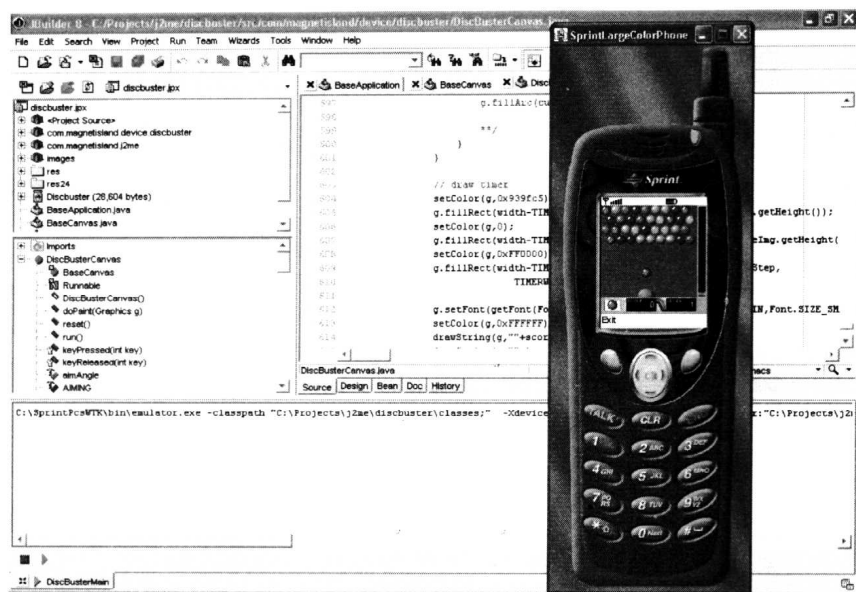


图 9-2 在 J2ME 开发环境中进行游戏开发

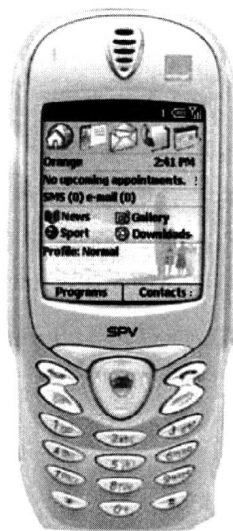


图 9-3 智能手机

■ 全球竞争

当前世界上的不同地区的无线网络作业的能力差别很大。那些提供无线网络作业服务的公司通常被称为移动通信运营商，比如美国的 Sprint PCS (www.sprintpcs.com) 公司。没有一个坚实的无线系统基础设施，你就无法让数十亿部移动电话以最快的传输速度进行游戏。我可以肯定地说：在这个领域没有专家，但看起来亚洲要处于领先地位，其次是欧洲，然后才是美国。这是一个处处都高速增长的领域。

NTT DoCoMo (www.nttdocomo.com) 是日本最大的移动通信运营商，它向公众提供了大范围的无线服务和应用程序。在欧洲，Orange (www.orange.com) 和 Vodafone (www.vodafone.com)，还有其他一些公司，为给用户相同类型的服务而相互竞争着。亚洲和欧洲都进行了相当程度的财政投资来构造自己的无线网络，这已经不再是什么秘密了，而尽管美国最近取得了很大的发展，但是已经被落在后面了。

全球良好的迹象以及所有移动通信运营商们的服务延伸，对于参与到游戏供给产业中的任何一个人来说都是一件十分令人兴奋的事情。很显然这些移动通信运营商拥有大量的用户，当我们以合理的方式、在能够提供合理运行速度的硬件上为他们提供适当类型的游戏时，他们便会成为潜在的玩家。

■ 开发中需要考虑的因素

如果试图要跟上每一款新的移动电话和它的特定硬件技术规格，同时还要密切关注像诺基亚 N-Gage 这样的混合设备，那简直就是发了疯。就像在其他平台上进行游戏开发一样，游戏开发者希望制作出来的游戏能够在各种广泛的和通用的游戏平台上运行。在其他一些原因中，这个因素就是为什么如此多的游戏开发者集中在 PlayStation 2 上进行开发的原因——世界各地的很多家庭中共有数十亿台这样的游戏机。

这里，我将集中讨论适用于各种移动电话的基于 J2ME 的游戏内容开发，尽管在方法上和你在开发一个基于 BREW 的游戏或者用于 N-Gage 手机的游戏比较相似。我们先来看一下进行开发时一些需要考虑的相关因素：

- **移动电话有限的处理能力** 现在，很多适用于 J2ME 规范的移动电话只能提供非常有限的处理能力。移动电话硬件没有能力完成复杂的处理任务，而仅是这一条就严重地限制了开发者所能够编制的游戏类型。移动电话内部的处理器会随着下一代移动电话的发行而具有更强的处理能力，但是它们还是远远不能满足

处理十分复杂的游戏应用程序的需求。

- **有限的可视屏幕尺寸** 移动电话都只有很小的可视屏幕范围。随着时间的推移这些可视屏幕会变大，但是它们还是很小。对开发者更为不利的是，还没有一个“标准”的屏幕尺寸，因此你的游戏程序所使用的屏幕尺寸就依赖于你开发的游戏是用哪一种型号的移动电话。这样当你制作的游戏要在几种类型的移动电话上都要很好地运行时就会出现可移植的问题。定制的图像和 / 或定制的代码通常都需要处理各种不同型号的移动电话之间所存在的屏幕尺寸上差别的问题。
- **有限的内存配置** 游戏通常都很消耗内存，而移动电话上的物理内存又很有限。为了降低制造成本，移动电话生产厂商不能够将特别多的资金投入他们的产品中，而且也没有一个现实的理由要求他们这么做，除非移动通信运营商也进入到游戏供应商的产业之中。移动电话的内存限制也在相当大的程度上限制了游戏的设计。
- **数据传输的速度** 数据传输速度会成为设计上的一个障碍，尤其是对于网络游戏应用程序。如果你开发的是一个单机版的游戏，玩家一旦下载后，他就能够在一段时间内或者永久地使用该游戏，这不会出现这种问题。然而，如果你开发的游戏应用程序要传递更多的内容而需要向服务器发送信息，那么数据传输速度就是一个极大的限制因素。而且这实际上不是处理速度太慢了——而是由于信息传输时必须经过的传输路径，即信息必须通过所有的网络路由器才能达到服务器并从服务器返回到移动电话中。
- **帧速率** 对于积极型的游戏程序这可能是一个很严重的性能问题，但是你可以制作出一个令人满意的基于小精灵的游戏程序那根本就不需要考虑帧速率问题。
- **颜色以及 alpha 通道管理** 你必须很小心地去处理复杂颜色的应用以及遮罩问题，因为你不希望扩大资源文件的大小。只有很少数量的内存能够用于加载你的游戏程序。
- **音效问题** 一些较新款式的移动电话，其声音确实十分迷人，但它还是不能够和其他平台上的高级音响相比较的。它可以很有效地支持简单的游戏程序。
- **授权问题** 授权也成为一个问题，特别是从 BREW 系统要求看，只有你的应用程序是签约或者批准了的才能够使用。
- **移植性问题** 现在对于大量的移动电话游戏，你需要在 J2ME 游戏版本和 BREW 游戏版本之间来回辗转，这确实是一个很大的麻烦。制作一个游戏版本并让它

能够在 J2ME 设备上运行的同时也能够在 BREW 设备上运行，这一点很不容易做到，更不用说能够在他们所希望的各种类型的硬件上直接正确地运行了。

- **同时按键问题** 大多数的游戏玩家已经习惯了能够同时进行按键。当前的移动电话不支持这一点。这就明显地限制了你能制作的游戏类型，即使你的处理速度很快能够利用同时按键的优势。有些游戏开发者正努力试图使用单个按键，应用单个按键的按下、保持和释放来决定某些事情，比如打高尔夫球时的力量控制。

这些只不过是众多因素中的少数几个，但是每一个因素所构成的限制都对确定你可以制作哪些类型的游戏起作用。目前是移动电话游戏发展的一个很有趣的阶段，因为对于很多着手于这种游戏开发的人来说，他们现在的感觉很像是在游戏开发的早期，那时候的游戏设备，如早期的控制台（Intellivision，Atari 2600），只有十分有限的处理能力和内存容量可供使用。



9.2 移动电话上的设计问题

现在是时候了，我们应该将 20 世纪 70 年代留下来的资料上面的灰尘抹掉，坐下来重新审视一下这些资料了。我不是在开玩笑。较新款式的移动电话带有小的彩色屏幕。嘿，起码我们能够使用一些颜色了。它们的处理能力和内存容量还很有限。而且，用于在移动电话上进行开发的程序设计 API（应用程序编程接口，简称 API，是由操作系统或者应用程序提供的一种方法，它能够让编写游戏的程序员向操作系统或者其他应用程序作出请求）也是有限的。它们现在还没有包含很多在其他平台上能够使用的功能，如缩放、图像翻转以及一些复杂的数学函数。

这意味着什么呢？这就意味着你所制作的那种类型的游戏不能够是针对特定处理器的，不能够占用大量的内存，而且还必须在小的彩色屏幕上能够看到。哎呀！你必须时刻牢记，尽管如此，这只是移动电话游戏发展的早期，而且这些日子终究会过去的。用不了多久，就会出现比现在复杂得多的游戏程序。只有到了那个时候，你才需要在给定的限制条件下尽可能地满足游戏玩家的要求并吸引更多的游戏玩家。

图 9-4 和图 9-5 给出了两个游戏示例，这两个游戏是针对与 J2ME 规范相兼容的移动电话制作的。图 9-4 给出的是为高彩 I-Mode（富士通生产的 F503i 型号）移动电话设计的一个解谜游戏，这款移动电话是专门针对亚洲的无线通信市场的。图 9-5 给出的

是一个平台风格的游戏示例。这种类型的游戏将移动电话的硬件发挥到了极点，因为大多数的移动电话都要求应用程序能够在 64K~100K 大小或者更小的范围内运行。新款式的移动电话不断地冲击着市场，使得像游戏这样的应用程序可用的内存有所增加。

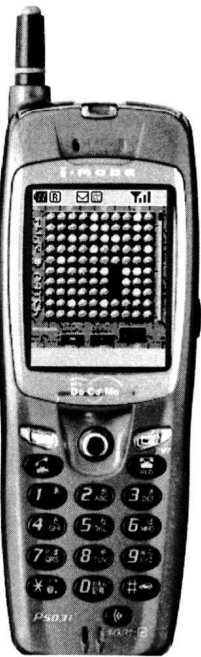


图 9-4 Magnet Island 公司开发的 SlideNmatch 解谜游戏

尽管可用的内存增加了一些，但我们所熟悉的大多数游戏都要求有足够的处理能力，因此在移动电话上开发一个运行速度令人满意的平台类游戏并不是一件容易的事情。

请记住，开发移动电话游戏的人员必须能够访问到“无线电波”（移动通信运营商向电话用户提供服务的电波），只有这样才能够使游戏的制作和升级更为方便、容易。如果一个住在美国的开发者为只在亚洲市场上发行的移动电话开发游戏，那么情况会是什么样子呢？当你带着你的移动电话都站在美国的土地上的时候，你根本无法访问到亚洲的移动通信运营商，因此对游戏内容进行修改时，你就不得不把电话的硬件反复地运来运去。这并不是一个十分有效的开发方式。

考虑一下到目前为止移动电话上的哪些游戏能够很好地运行，这一点很重要。下面的内容绝不会是详尽无遗的，但那些能够很好运行的游戏都包括下面几种：



图 9-5 Magnet Island 公司开发的 Kanga Kaos 平台游戏

- ▶ **低激烈程度的射手游戏** 与性能有关的问题决定了射手游戏的速度或者是激烈程度。现在还不能指望移动电话上会有与游戏小子改进版上同样的游戏体验。要记住游戏小子改进版是一个便携式的游戏平台，具有为运行游戏而专门设计的专用硬件。移动电话连在屏幕上跳出一些小精灵并把它们快速地来回移动（就像在游戏小子改进版里那样）都做不到。简而言之，即使是将一个射手游戏做得像在 8 位的任天堂上的老式射手游戏那样好都是一件特别困难的事情。
- ▶ **简单的街机游戏** 如果不要求很高速度的话，这些游戏能够在移动电话上很好地运行。只要是由你引导着游戏主角在屏幕上穿梭，速度不要求很快，并且让玩家接受一些简单的挑战或者是收集目标物，这样的游戏都可以作为它的好范例。你会想要围绕这一考虑进行设计的。
- ▶ **运动类“情景”游戏** 这些游戏最为典型的就是从某项运动中选取某种情景或者动作事件，并围绕它来制作游戏。其中的例子有曲棍球或者篮球的“得分”游戏。当前，这些游戏只能简单地使用运动场景，因为要将整个棒球或者足球游戏放到一个移动电话上并使人的眼睛看到整个游戏，同时还要运行得足够好，

能提供令人满意的游戏可玩性，这将是一件极其困难的事情。

- **简单的运动类游戏** 这些运动类游戏不需要高质量的图像处理就能够很好地表现出来。推圆盘游戏和保龄球游戏就是很好的例子。那些既使人着迷又不需要多少图像处理的体育运动就特别适合在移动电话上开发游戏。
- **解谜游戏或简单的匹配游戏** 只要这些游戏不要求剧烈的运算或者极快的速度，它们就更适合于移动电话。通常情况下，这些限制因素会直接驱使你设计解谜游戏。图 9-6 显示了另一个解谜游戏的例子。这些游戏具有的优点是便于绘制，并且在各种不同类型的移动电话上，你都能够使用提供的独特的按键布置来配置简单的控制。



图 9-6 Magnet Island 公司开发的解谜 / 匹配类游戏

- **主题测验游戏** 如果问题和答案都能做得很短，并且通过简单地按一个数字键就能作出回答，那么这些小游戏（可以是关于体育运动、电影、流行文化等内容的）就会运行得很好。如果能结合准确的资源内容来制作问题和答案的话，

这种类型的游戏会流行起来并具有知识性。

哪些游戏不能够在移动电话上很好地运行呢？嗯，就是那些大多数游戏者已经习惯的在这样或者那样便携式平台上玩的游戏。还有就是，三维游戏的开发虽然起步很早，但当前的可见屏幕尺寸问题却产生一些很大的开发障碍。这些障碍用不了多久就会被克服掉，但现在大多数的开发者都在集中精力开发一些能够在当前硬件上运行得很好的游戏内容，并提供一些令人满意的玩法。制作这些游戏的目的是让人们在车管所排队等候或者坐在机场终端等待航班时，能够用移动电话来消磨时间。

任何为这些最新款式的硬件开发小型游戏的经验都将能够应用在这些游戏的开发中。我甚至见到过这样的“从头再来”：经验老道的游戏程序员们从早期的 Atari 时代的阴影中走出来，转而开发移动电话游戏，或者是将他们的一些古老的程序库升级换代。

■ 移动电话的对战游戏

数家移动通信运营商在他们成功地开发了某些单玩家单机版游戏后，正在推动开发者们进行移动电话的对战游戏开发。目前，这些游戏都是回合制的，能够让玩家相互配对进行比赛。为了着手制作这种类型的游戏，大多数的回合制的游戏作为测试题材都能够很好地运行。这些游戏中的大多数都是很早以前的老游戏了，因此根本就没有办法确定这种类型的游戏会流行到什么程度，但是合适的游戏类型好像还是能吸引不少玩家。

移动电话正开始使用蓝牙技术——Bluetooth (www.bluetooth.com)，这样玩家能够找到其他的玩家。比如说某个玩家在火车上想玩一个游戏，但是距离其他的玩家很远，或者不知道他们是谁、他们在什么地方，这个玩家就可以用移动电话向那些潜在的玩家广播一条消息：“有没有人想玩《真人快打》——*Mortal Kombat*？”

你可以猜测到多家移动通信运营商都会寻找到合适的游戏类型，以使他们的订购者对所提供的游戏感到满意。大量的游戏将会集中吸引那些十来岁的无线订购者并让他们感到快乐。

对于游戏开发者来说，拥有一支庞大的、非常精通游戏的玩家队伍是一件很幸运的事情，因为无线订购服务的增值部分将取决于游戏的内容以及其他一些东西，要始终使订购者对他们所提供的服务价值感到满意。

■ 无线玩具网络

对于某些开发者来说，这是一个扩展出来的领域，尤其是对于那些利用玩具授权而

制作的游戏。为了提供本地的无线玩具和游戏娱乐，玩具生产厂商似乎更加频繁地与游戏开发者进行合作。有些玩具的特点是将特定的游戏单元以某些方式制作到玩具中，而另一些则作为本地无线玩具网络的一部分在玩具中提供游戏互动。

思考一下适用于“带有本地无线命令中心”的下一代玩具的各种游戏开发经验。这些命令中心能够通过 USB 接口连接到个人计算机上实现快捷地在线更新，并且会向那些“识别的”玩具单元发布简单的游戏命令。对于动作角色，各种类型的运输舰队等等，这样的例子无穷无尽。

该领域向玩家奉献的另外一件礼物是像 Cybiko Xtreme (www.cybikoxtreme.com/) 这样的专用无线游戏单元。这些单元能够传输便携式的游戏和局域网功能，用于与朋友聊天或者是玩儿游戏。图 9-7 给出的是便携式的 Cybiko Xtreme 设备。



图 9-7 Cybiko Xtreme

这些设备的共同缺点是它们的零售价格，它们的扩展能力与升级问题，以及由于缺乏一个庞大而持久的玩家基本队伍而不能吸引开发者和有吸引力的游戏内容。还有就是，有些平台上还没有形成一个足够大的群体或者没有任何玩家会快速增长的迹象，开发者们也很不愿意为这样的平台开发游戏。很多的玩具公司和游戏公司使用各种类型的硬件已开发出了便携式的游戏，其最终结果也多种多样。Cybiko 公司能够生存下来并不断地改进其产品，其秘密就在于该公司开发出了具有吸引力的产品。

其他的一些产品，包括根据《口袋小精灵》——*Pokemon* 游戏开发的电子 Pikachu

玩具，它们也很流行，但在游戏和无线功能方面却十分有限。玩具制造商冒着巨大的风险，他们得要收回他们用于研究和制造精心设计的无线玩具所花费的费用——尽管这也涉及到兴趣方面的因素！一些开发者在致力于使这些开发费用最低的同时还要使他们开发的带有简单而有趣的互动和游戏要素的产品的功能最大化的研究。

■ 打造移动电话的游戏市场

正像在控制台游戏领域中一样，很多流行的授权游戏也正在大量的涌入移动电话游戏领域。那些擅长于开发移动电话游戏的游戏发布商们开始大规模地计划向世界各地的移动通信运营商提供各种类型的游戏，他们已经崛起了。这些发布商包括 JAMDAT (www.jamdatgames.com)、Digital Bridges (www.digitalbridges.com) 和 Unplugged 公司 (www.unplugged-inc.com)。

一些高档次的授权游戏也开始出现在移动电话上了，其中包括 JAMDAT 公司的《魔戒：双城奇谋》——*The Lord of the Ring: The Two Towers* (见图 9-8)，JAMDAT/Activision 的 *Tony Hawk Pro Skater 4* 和 EA Sports 的 *Tiger Woods PGA Tour Wireless Golf* (见图 9-9)。

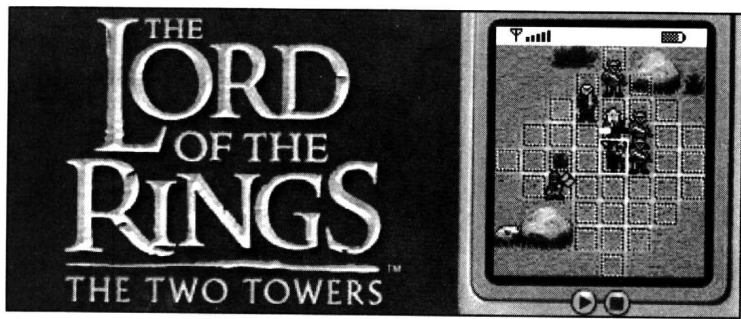


图 9-8 《魔戒·双城奇谋》

随着数以亿计的新订购者签订了无线服务包，移动游戏发布商希望他们众所周知的授权产品会激励玩家购买他们的游戏，并将这些游戏下载到自己的电话上。那些较年轻的移动电话的用户和玩家群体，他们完全不会被无线技术吓倒，对所有这些游戏甚至更多的游戏都是一个很具吸引力的潜在的人群。尽管如此，就像我前面提到的那样，决定其价格的经济模型仍处于实验阶段，当然这个模型要能够为所有的合作伙伴都提供尽可

能好的结果的同时还要将游戏提供给最大数量的用户。

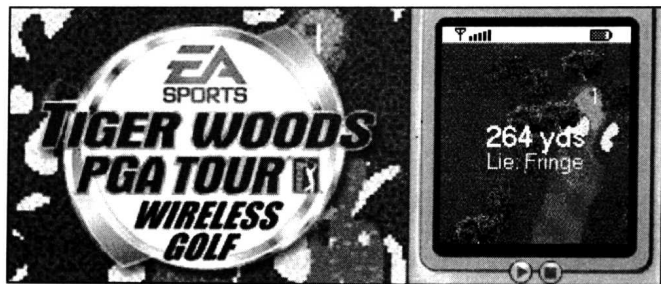


图 9-9 Tiger Woods PGA Tour Wireless Golf

就像其他平台上的游戏那样,会根据玩家的爱好和年龄段将这些游戏多多少少地划分为一些类型,在无线游戏领域中也存在着类似的分类,这样,确定类型的游戏和娱乐程序就会到了它们“固有的”玩家手中。为数众多的游戏仍然有着广阔的设计空间,以满足大量的、不断增长的玩家人群的需要。在未来的几个月或者几年内,我们将会十分高兴地看到有多少的人将会因为移动电话游戏的诱惑而成为新的玩家,而且不管他们偏好什么样的,他们都将会在众多类型的游戏中找到自己的所爱。

当我们看到那么多的新玩家在全新的游戏平台上以全新的方式体验玩游戏所带来的乐趣时,为什么会很兴奋? 其中的一个原因就是移动电话游戏市场的发展。游戏产业的这一方面为游戏开发商提供了不断的刺激与挑战,但是随着我们身边的新游戏领域的发展,这也需要同样程度的耐心和耐力。对于那些喜欢游戏的人来说,他们总能以令人感兴趣的新方法提供游戏给玩家。由于玩游戏本身就是以人的体验为中心的,提供全新的游戏有助于培养全新型的游戏玩家,一想到这些各式各样的可能性就会让人感到激动。由于无线游戏玩家是如此的广泛,这个领域在某种程度上对具体的人口统计数据的依赖要比其他领域小。

■ 无线游戏的机遇

正像你在本章中已经看到的那样,在将几种类型的无线技术(局部的或者是基于移动通信运营商)与游戏和娱乐结合在一起的过程中,存在有大量的机会,而且这种机会还在不断地增加。在无线技术中,这些机会并没有受到移动电话硬件的约束,也没有受到任何其他特定的平台所束缚,反而扩大了,在无线技术中,游戏开发者们的才干可以

应用于全新的游戏体验。实际上，全新的游戏领域还有待于进一步的界定。

在所有这些无线游戏发展方向上，我们很难准确地预测游戏到底会走上哪一条道路，但是对我来说，有一个想法却一天天地变得更加清晰了：无线游戏将会以其特有的方式进入到一个比我们以前所认为的还要庞大得多的游戏世界中。对于开发者来说这是一个需要下注的领域，在这里，游戏的目的是那些争取潜在的玩家，而其数目之大我们以前从来没有听说过。就像是在该产业短暂的历史上，游戏开发者们的设计方面的知识在很大程度上就是进行比较、发展以及改进一样，玩家对无线游戏的反馈意见将继续向游戏设计者提出挑战，要求他们在提供能够满足玩家的愿望和要求的游戏上能够快速地作出响应。让游戏超越每个人的最开阔的思维，这是一个多么好的机会啊！

在无线游戏中出现的第一个游戏决不会是最好的游戏，这一点不会让人感到吃惊。出现在任何新平台上的最好的游戏通常是第二代或者第三代游戏。开发商通常都会疯狂地学习怎样才能花费最少的力气而完成更多的工作，以及怎样才能突破所有的限制。一些业界的评论员看起来甚至是根本就不相信无线游戏能够或者是将会成功。关于这一点，只有耐心和时间才能说明一切。

就我个人而言，我认为我们应该尽可能充分地利用每一次制作游戏的时机。我们不要仅仅因为处理器的功能太弱、屏幕尺寸太小以及没有标准的屏幕尺寸就放弃或者取消无线游戏的开发。这就要求我们要有耐心。

在我的印象中，能够发展新游戏的每一次机会都应该充分地加以利用，其目的是为该产业注入我们急需的新鲜血液以及伴随而来的新的游戏激情，而这个产业有时会处于停滞状态——尽管我们会遇到持续不断的难题。如果我们还继续在两个类型的平台上为“一些核心部分”制作三种类型的游戏的话，那么作为一个产业或者开发商我们就不能很好地成长，而且摆在每个人面前的机遇照样也会消失。



9.3 无线游戏初期开发因素的案例研究

当提及移动游戏开发时，“未成年”几乎就暗示着快要成熟了。就目前为止，无线游戏开发还处于一个疯狂的世界之中。如果你已经注意到我们身边很多这类型的游戏的话，我敢肯定你所见到的那些为移动电话制作的、并由移动通信运营商发布的游戏，它们仅是刚刚能够运行而已。

就像前面已经提到的那样，任何类型的游戏进行世界范围的开发都是很困难的，原

因是对任意给定的移动电话其有权访问的移动通信运营商都是位于某个区域的。你可能会发现,你使用来自另外一个国家的移动电话工作时,它根本就不能测试你的游戏或者将一个游戏完整地下载到电话中。为了测试你的游戏,你需要将移动电话运回到原来的所在国,只有该国才能够为这款移动电话提供有效的无线服务,并请人在电话上下载并测试你的游戏。就像你能够看到的那样,这根本就不是一个最佳的开发过程,这也是为什么居住在一个地区内的开发者始终能够获得最终要在该地区使用的、重要的游戏的开发工作合约。

对于移动电话制造厂商来说,要求他们提供在其移动电话上应用程序开发的对接设备,这看起来是很合理的,这使得开发者能够传送应用程序信息到位于其服务区外的移动电话上,但是我还从未见到过像这样的任何东西呢(这并不意味着不存在这样的解决方案)。当涉及到游戏时,移动通信运营商他们自己也正在涉足一个未知的领域。当测试部门“打扫干净房间”准备好要测试像游戏或者其他类型的移动电话应用程序时,这些程序却还没有安装好,更别想它们能够运行了。

测试加载好游戏后的移动电话也会遇到几个特定的挑战。如果你是为一系列型号的移动电话(好像通常情况下很多移动游戏发布商都是这样要求哦)编制的游戏,作为一名开发者,你可能没有机会能轻而易举地接触到所有这些机型。每种型号的移动电话其技术规格都不相同,而且游戏的表现各种型号的移动电话上看起来也是不同的。那些能够在一种型号的移动电话上毫无漏洞地运行、甚至是运行得相当完美的游戏,在另一种型号的移动电话上运行时很有可能会崩溃。

经济方面的因素是什么样的呢?由于移动游戏发布商不能够肯定他们会从任何一个给定的游戏中获得多大的利润,可以理解他们没有必要冒险将大量的资金投入移动到游戏开发中去。很多急于进入该市场领域中去的开发商,都必须很仔细地处理这些“实验”的预算费用。移动游戏发布商和开发商一样,都迫切地想知道到底怎样才能把这些移动游戏推销出去,以及这对未来的移动游戏开发意味着什么。

很显然,只有出现了强壮而清晰的市场行情时,所有的这些开发商才会继续为这个平台推出游戏或者其他应用程序。就像在游戏开发中的任何其他领域一样,那些开发过强大的游戏并有很好的销售业绩的开发商才会成为任何发布商的具有吸引力的合作伙伴。发布商通常情况下都去寻找已证明具有实力的开发商,把游戏交给他们进行开发。当然,这对于新的游戏开发商来说是一个很大的障碍。

很多的移动游戏发布商都赌注未来对移动游戏的需求会远远超过现在,并对当前的

项目投入了相对少量的资金,他们认为这些资金已经足够以合理的速度来不断的扩大这个市场,而只有对游戏提出了新的需求时才值得投入更大数量的资金。

很多看起来非常适用于低功耗的移动电话的游戏正在不断地涌现,它们将有助于考察市场对类似游戏的需求。简而言之,随着很多具有竞争力的技术开始到位,接着就要开始大型的试验了。

然而,对潜在市场的一个相当公平的分析应该能够证明:对这个领域进行的初始的投资是非常值得的。极少数的人可能会认为每天都将会有更多的人获得无线服务方案,将其视为一件一次性投入的奢侈品,这绝对是很普通的事情。作为一种交付机制,游戏产业还从来没有见到过像这样具有巨大潜力的前景。它将会把每一个人都变成玩游戏的人吗?不会的。但是实际上你不能忽略了这样的想法,只要提供了合适类型的游戏,由于接触到游戏的机会增加了,许多新的玩家就会相应地产生。

现在给大家讲一些好消息!在本章较靠前的部分中,我们看到了一些与制作无线游戏相关的特定的开发挑战。尽管这些挑战在目前还明确地存在着,但我们也有足够的理由相信这些游戏在未来将会得到快速的发展。下面是与无线游戏相关的一些积极因素:

- ▶ **即将到来的图形解决方案** 像 ATI (www.ati.com) 和 NVIDIA (www.nvidia.com) 这样的图像芯片制造商正在密切地关注着移动通信市场,而且毫无疑问他们会为移动电话制造商提供图形芯片的解决方案,使得开发更加强健的图形游戏成为可能。这将允许开发商在展现游戏内容的可视化方面做更多的工作。
- ▶ **更大、更清晰的可视屏幕** 利用精巧的设计,移动电话制造商正在寻找方法来制造物理尺寸更大的可视屏幕,而不用大幅度地增加移动电话的重量和尺寸。这个因素结合对物理可视屏幕的技术上的改进,就能够提供更好的显示效果和相当清晰的细节显示。
- ▶ **具有吸引力的多玩家应用程序** 即使在单机版游戏产品不断增长的情况下,一些发布商已经开始寻找某种能使人们能够通过多玩家游戏而连接在一起的方法。开发商们正在尽自己的努力开发一些游戏来满足这些要求,以壮大基于移动电话的多玩家游戏群体。这就为开发商提供了一些新的机遇。
- ▶ **与无线游戏相关的新的游戏玩家与文化** 某些类型正确的无线游戏能够成功地吸引住玩家,围绕这些游戏的整个亚文化群就会悄悄地萌芽。在刚过去的游戏历史中,同样的现象在个人计算机上发生了:数种限定类型的游戏发展起来了大型的社区以及亚文化群体。

DAVE WARHOL 访谈录



Dave Warhol 是游戏业界的先驱者，并且是 *Blue Sky Rangers*（早期的 Intellivision 开发商）的成员。他是 *Realtime Associates* 公司的构造者和董事长，该公司在他的领导下已经为众多的游戏发布商开发了超过 80 个游戏，其中包括 Bug!，这是为 *Sega Saturn* 开发的一个大众性游戏。我想得知 Dave 对几个业内问题的看法。

TM: 从 *Intellivision* 到现在，你对什么东西的印象最好？

DW: 我们所处的这个环境中的每一个人都感到非常的幸运。我们得到了一个游戏开发机会，那个时候还没有游戏产业。那些“老前辈们”也只有一年时间的开发经验。这就构造出这样一种情景，其中的每一个人都对基本的游戏设计问题和技术构造感到兴奋，并对这些问题进行学习和试验。这是一个进行发明的阶段。根据现在标准这并不是一个很大的封套，但那时却出现了大量的封套！我最喜欢的一段广泛的回忆是向其他的程序员进行学习；最喜欢的一段特定的记忆是当我走进一个大厅听到有一群人正在玩我的游戏——只因为有趣！

TM: 你在游戏界遇到的几个最大的变化是什么？

DW: 这几年中最大的变化是技术上的变化以及产品的范围与团队的规模。在 1980 年，仅仅一个人，加上一个美工几个星期的工作和一个负责声音的人几天的工作，就能在三到四个月内编制一个完整的游戏。这就意味着大约有 5% 到 10% 的精力是用来连接这些东西的，而其他的 90% 到 95% 是用来进行游戏的设计、写代码和调试。在开发的过程中我们能够不停地和另一个玩家一起玩或者进行游戏对抗。这些游戏太小了以至于那时还没有集成技术设计，而现在集成技术是必不可少的；设计就像玩游戏时的自然序列一样演变着。考虑一下现在项目中的情况，其中仅是“使游戏能够运行”，或者“将游戏连接起来”，再或者是“增加一些效果”就要多花费很多很多的时间。是的，这些就构成了一个充满竞争的时代中游戏体验的所有部分，但是从游戏可玩性的角度讲，所有用在该部分上的时间所占的比例相对于所有花费的精力来说已经下降了。

TM: 游戏开发的风气是怎样随时间变化的？

DW: 过去所有的那种新奇的东西现在已不存在了。过去曾经有很多不眠之夜，但现在由于与投资者们的相互依赖性，这种情况作用也不太大了。因此这种风气就变成了

一种计划编制、联合作业以及相互交流，而过去则是“一头扎进来就开始工作。”

TM: 现在游戏制作“生意”对设计方案的影响有多大呢？

DW: 作为一个商人，我希望游戏制作产业能够百分之百地影响到一切。其最低限就是一个游戏应该能够赚钱，否则的话将没有人愿意制作游戏。而且那些最为成功的游戏通常都是在商业上获得成功的游戏。

我们也为那些对赚钱没有兴趣的人留了出路；这些人被称作所谓的“艺术家”。如果某人想要抱怨游戏产业内的财政压力的话，那他就应当不再从游戏产业中拿钱；在威尼斯买一个小房子，不要再洗澡了，在当地的咖啡店里找一个自己喜欢的座位，并开始创造纯艺术的活动吧，就像世界各地的那些穷苦的音乐家、画家和诗人一样。

认真地说，商业上的压力要求游戏设计能够在相当大的程度上保持一致性。电视和电影在同种程度上都变成了同一种形式，游戏也是这样。不幸的是，在另一方面，游戏也创造了一个能维持的、可预测的市场。也许交互的方式是很有限的，并且这种方式已经得到很好的利用；也许所有的方式都是基于“安全的市场选择”的，它只允许提供这些选择。不管是哪一种方式，消费者还在买我们的游戏，因此情况还不错。一个无法归入到已存在的任何类别中的成功游戏通常会有所突破；而它只能成为一种漂亮的点缀。

TM: 与早期的真枪实弹的开发时代相比较，现在你是怎样进行游戏设计的？

DW: 现在做得特别好的一件事就是你可以从理论层次很高的基础开始——从“二次开发”开始——做你的游戏设计中需要完成的工作。对于当代的程序设计者来说，他们有整套很广泛的技术和描述性更强的语言可以使用。在此之前，设计者们仅有少数几个小模子可供使用：你要开发一个体育运动游戏、迷宫游戏还是其他类型的游戏？现在，你可以谈论你想要让玩家有什么样的“感受”，尽管你的任务只是开发一个有吸引力的游戏而已。

在过去由于预算较少，你能够“担负”更多重复性的工作。要在一两个星期内将所有的东西理顺好，测试游戏，找出哪些地方能够正常工作而哪些地方不能，并作出修改。

TM: 什么样的游戏思想还会使你感到激动？

DW: 具有大量玩家的游戏是很有趣的；它们现在快完成了它们的第一代，就像 20 世纪 80 年代中期的控制台游戏。开放型的游戏和非引导型的游戏还有很大的开发空间；五年内最典型的可能是自我引导型的。我喜欢冒险的设计。

TM: 游戏产业中哪些好的或者是不好的趋势使你感到不安？

DW: 不好的趋势主要是“雷同”综合症。（第一人称射击类游戏比 Phil Collins 的

诗歌还要多——他的诗歌就够多的了——它们都做得很漂亮使我们很难把它们区分开!)发布商希望能够通过开发已经成熟类型的游戏来最小化风险投资,因此他们能够参考一些市场方面的数据并确信开发的游戏能够卖出足以收回成本的份数。而同时,他们也希望有一些不同的东西;因此作出来的游戏既与另一个成功开发的游戏有所区别,又和它很类似。

另一个令人不安的趋势是游戏要做得很庞大才能卖出去。好莱坞拍摄一部电影其花费约在两百万到两千万美元之间——这是一个相当大的范围。但是在游戏中看起来像是发布商仅对制作少量更大的游戏感兴趣。

我喜欢的趋势就是继续去吸引那些非中坚的游戏玩家,就像《模拟人生》——*the Sims* 那样。

TM: 对游戏近来一段时间的发展你会关注哪些东西?

DW: 我希望这一天的到来,那时电视网络的经历会在某个特定的晚上由于其等级的下降而牢骚不止,原因是一些因特网上提供的内容在这个特定的时刻开始发行,并在那一周挖走了他们的观众。

TM: 游戏以后将会走向何方呢?

DW: 开放型游戏,能动态更新的游戏,多玩家和大量玩家的游戏。

TM: 你是怎样为现在的游戏产业做准备的?

DW: 玩啊玩,不停地玩!

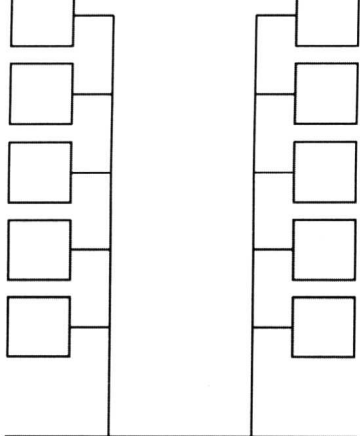
在享受游戏的乐趣之余,仔细地在头脑中消化整个游戏。要指出来它的每一部分都是什么,作出了什么样的设计方案与权衡。对游戏进行分解,不要只顾着玩。可以根据自己的见解写出评论。

说到职业方面的机会,就是测试,测试,再测试。找到一份游戏测试工作是进入到游戏产业的一条重要的途经。

MEGA 提示

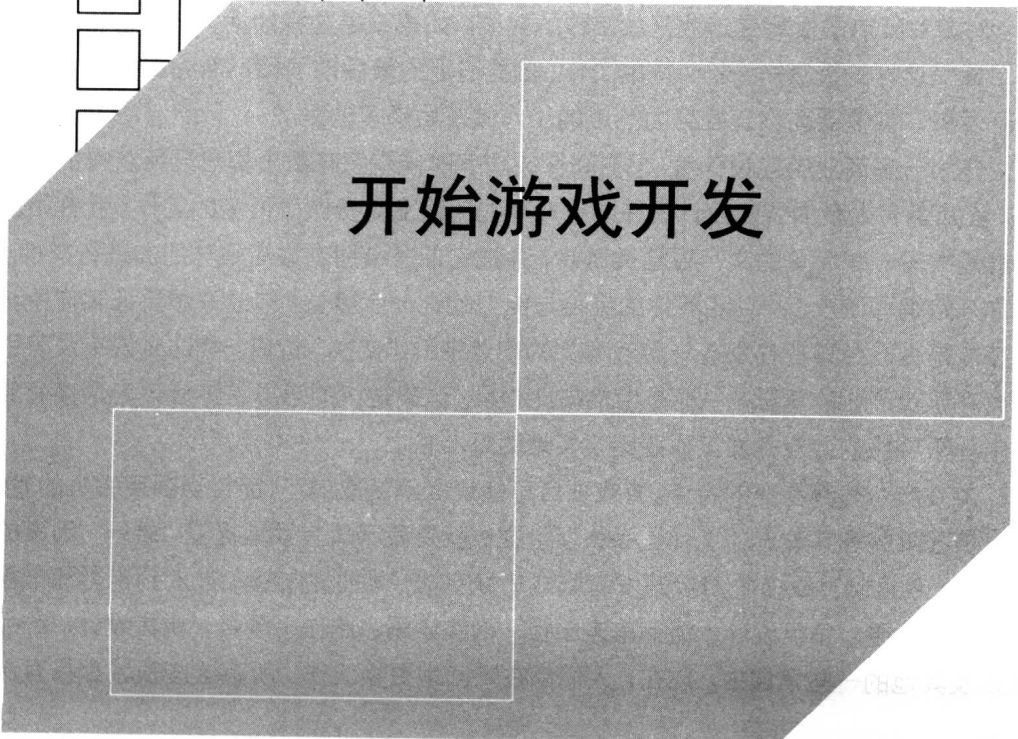
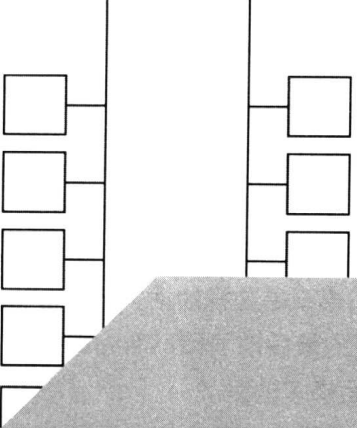
1. 就像移动电话上被迫开发的游戏一样,简单的设计通常是很好的设计练习。要记住,不管游戏是在什么样的平台和场景下运行,都要尽力去履行满足玩家愿望的意图。
2. 在所有你自己的游戏想法中,留意一下哪些想法可能最适合于便携式移动电话。

3. 便携式设备的限制提出了好的设计挑战。这些挑战迫使设计者集中精力去开发一个既简单又让人着迷的游戏。这说起来容易但做起来却很难。
4. 要密切关注全球无线游戏设计的趋势并对此做好准备。
5. 参考一下自己的游戏库和自己喜爱的游戏。试着将你喜爱的游戏“改编”成一个不需要很健壮的便携版本的游戏。如果没有技术支持，这个游戏还能不能继续下去？你可能感到吃惊。

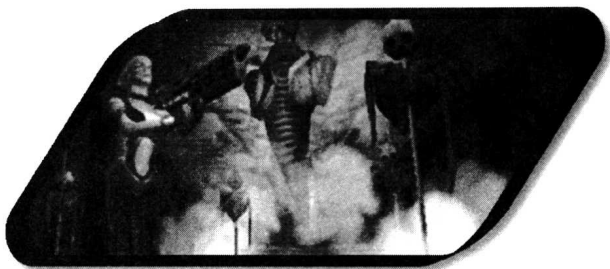


CHAPTER

10



开始游戏开发



对于那些对游戏设计要素充满热情的人来说，眼前有一个广阔且还在继续增加的机会。在书写脚本、舞台设计、光照、编程、建模、布局和“游戏测试”等方面，你掌握的知识越多对你来说越有利。我已经假设你玩过每一种类型的游戏，并且对它们作了比较。

当你掌握了多种专业基础技能，你就为自己创造的新的机遇。在这里我无意给你增加任何的压力，但是游戏或者娱乐产品设计是一项竞争非常激烈的专业化工作。很多的开发者为了获得惟一的游戏开发职位而相互竞争的事儿时有发生。雇主也知道有很多的人都希望从事游戏设计工作。通常，从事游戏开发的人一般都不愿意从事任何其他的工作。对游戏狂热的爱好是游戏开发工作的需要。如果不是这样的话，那你为什么还要让自己遭受这样的折磨——长时间的工作、日益迫近的最后期限和不断缩减的预算？在游戏开发中，你需要为你正在努力完成的工作储备足够多的爱。

作为一名商业游戏设计者，想要获得成功有时候就意味着出卖自己所获得的特殊技能。我知道有很多的游戏设计者在寻找成功的道路上免费做了很多的工作。我开始的时候就是编写一些简单的文字冒险类游戏，并把它们送给朋友或者是与朋友进行交换。一旦你开始编写游戏，并且随着你的经验与能力的进一步增长，你或许想要证实你所掌握的技能是否已经具备在那些与游戏有关的产业中的“交换”价值。我这么说决没有引导你离开游戏产业的意思——而是恰恰与此相反。这条建议的意思是你应该为迎接商业环境中基于项目的游戏开发工作踏踏实实地做好准备。

游戏产业有着易变的特点。游戏项目可能说取消就取消了，而且当项目丢失的时候，很多情况下就意味着你的工作也丢失了。这也就是我不厌其烦地提及“转行”因素的根本原因。我见到过很多的游戏开发者都成功地完成了职业的转变，进入到电影和电视 / 图形界面效果、角色设计、情节串连图板、网络娱乐、制作、漫画、玩具设计、产品设计以及其他的一些领域中。现在，人们都在来回地更换工作。所有这些领域都具有自己

特定的生产需求和工作流程，需要设计者学习各种新的项目知识。

而现实情况是：在任何一个时候投产的游戏数量都是固定。如果你没有参与其中的任何一个项目，你就想要继续培养你在游戏项目中工作的能力。在游戏产业中工作的一个重要的因素以及一个严峻的挑战就是要要求自己不停地培养和改进自己的技能。

同时也不要忘了，广泛地参与各种平台上的类型各异的游戏项目的开发能够使你在解决项目中遇到的问题时具有敏锐的洞察能力，这些问题涉及游戏的种类、预算和平台的特定细节。

由于我们已经讨论完了游戏内容的构造，本章中我将讨论现代的游戏设计人员需要将会在游戏设计领域遇到的几种不同类型的机遇，以及让自己掌握多种技能的重要意义。



10.1 为什么要掌握多种技能

我们应该使自己的技能或者甚至可以说是进入游戏产业的手段多样化，这有很多重要的实际原因和面向发展的原因。在游戏产业内工作是异常艰苦的！获得第一份游戏开发工作可能是特别困难的事情。我家里装满了几个鞋盒子的拒绝信就可以证明这一点。问一下其他的开发人员，你也会听到同样的故事……获得头一份工作不是很容易的，但是在被拒绝后你还能够保持乐观对你来说就是好消息。

即使是有多年的知名游戏开发经验的游戏开发者，在寻找工作时也会面临着被拒的可能。很多时候你都难逃此劫，但是却不能因此而给自己施加巨大的压力。我的建议是：要好好考虑你在游戏开发中所制定的目标（就是那些为自己明确定义的东西），并在任何时候都不能放松对技能的追求。要想更好地提高自己的技能，你就需要不断地进行练习。尽量不要对任何一种产业感到沮丧，同时如果感到满足的话，就可以回想一下在你之前还有很多人在追求自己的梦想都有过同样的焦虑。

如果你对游戏编程有着强烈的爱好，那就要尽自己的所有努力去从事这项工作。通常情况下，好的游戏编程人员总是供不应求。以我的经验来说，美工和设计人员在雇用方面的情形则并非如此。据我所知，只有那些广泛地掌握了各种各样技能的人才能够长期一直从事该工作。如果他们不在游戏项目中工作，他们可以制作情节串连图板，为杂志、光盘或者图书封面做插图，为音乐电视制作后期动画，参与网络制作，进行概念设计和角色设计等等。简而言之，他们的工作不是完全“依赖于”游戏产业或者任何其

他的单个产业的，而是尽量摆脱仅作为一名材质美工或者仅一名建模人员的尴尬境地。即使现在他们（指材质美工或者建模人员）正在为某个很棒的游戏而努力工作，日子过着也很快乐，但是他们还要调整自己的技能，为以后随时可能的改行做好准备。

很显然，游戏业是一个正在发展的产业，从一开始就处于跌跌撞撞的状态下。游戏产业在某种程度上讲是不会衰退的，原因是人们总是想玩游戏，而且玩儿游戏的人的数量每年都在增加。最好的游戏开发商总能够抓住那些出现在他们面前的机遇。在某个方面做得最好确实是很重要的。你也许会很幸运！我仅见到过少数的游戏美工和设计人员经历过相对比较稳定的工作。我试图为你指出一条正确的方向，但你最好是两手做准备。

可以肯定地说：美工和游戏设计者总是在寻找各种机遇。第一点，也是最重要的一点，是你想要制作游戏，整天整夜都在思考着要制作游戏，但是你也要准备好接受其他类型和其他种类的新式游戏娱乐工作。完全没有必要把自己限制到单一的游戏类型中或者单一的开发平台上。你参与开发的游戏类型越多，你对游戏的制作过程就懂得越多。这并不是说你不需要掌握特定平台上的核心开发过程；如果有机会的话你就应该试着掌握这个核心开发过程。

如果你居住的地方离本地游戏开发商的办事处很远的话，你可能需要重新换个地方居住。在搬迁之前试着和一个“不错”的开发商构造关系。也不要太主动了，就用通过网络发送邮件联系就可以，但要持之以恒。在某种程度上，只有在你证明你曾经参与制作成功的商业游戏，你才有机会展露出你在设计方面的才能。不要过于没有耐心。就像我说过的那样，既然你已经开了头，你就要让自己成为一名游戏开发人员，接着向他们展现一下你所能做到的。你可以把附录 A 作为你的出发点对照一下。

■ 游戏设计者的职责定义

作为游戏设计者，你为你的游戏开发商的贡献在很大程度上取决于你自身具有的技巧、目标、背景知识、天赋以及兴趣爱好。要成为游戏开发团队中的设计人员、甚至是设计要员，我们到底需要掌握哪些类型的技能，关于这一点，不同的游戏开发工作室（从最大的工作室到最小的工作室）具有完全不同的看法。一些工作室像传统设计一样会选择把重点放在美术技巧方面，而另一些则选择把重点放在脚本语言能力上。甚至还有一些认为关键是具有充足的美术和脚本方面的技能，同时还要能够兼任角色操纵或者技术指导（为动画构造角色或者道具模型）工作。

就职位描述而言，你有三个常见的类别：设计主管、游戏设计者和关卡设计者。这

些类别之间通常具有大量的重叠的工作内容。尽管游戏和关卡的设计者也是较高的职位，但通常设计主管的职位要高一些。个人资历是构造在个人经验和表现的基础上的，这没有什么好奇怪的。

设计主管的任务是定义、构造以及支持所有的游戏可玩性功能细节。他们帮助协调设计人员集中精力制作流畅、刺激的游戏。他们要从游戏可玩性和视觉设计的角度出发，确保游戏具有竞争力。

游戏设计者也可以是设计主管，反过来也同样成立。游戏设计者的主要任务是定义和发展游戏可玩性细节的整体描述，当然，这个游戏的开发通常都是应用各种各样的方法从零开始的。他们中的许多人都有着足以胜任的美术和脚本编程技能。对于任意一个给定的游戏，他们都能写出一套完整的游戏设计文档，详细说明所有的相关设计细节和制作中需要考虑的因素。

关卡设计者的主要工作是构造各个关卡，这与构造整个游戏的支撑框架结构不同，但并不是说他们不能构造完整的游戏支撑框架结构。一般情况下，他们都忙于构造并管理与构造极具可玩性的关卡有关的各种细节。

某些游戏设计者具有核心程序设计的背景，因此更喜欢脚本和设计细节。而另一些则具有更多美术方面的背景，因此更喜欢环境建模、角色设计和文字工作。

现在有几件事情是可以肯定的：你需要掌握基本的绘画与设计技巧；掌握扎实的写作技巧；需要彻底地理解“玩”和“感觉”的动态变化；需要理解某个布局图中道具、角色和物品的摆放。最好还要掌握一些脚本编程知识。

你是如何来掌握这些技能的呢？这绝非一日之功，并且没有人能够不通过学习就可以提高自己在这方面的技能。我总是试着进一步提高自己的游戏开发技能，而我总是感觉到我又落后了。所有的游戏开发者，包括游戏设计者，始终都在努力地学习以增进自己的技能。学习是永无止境的。我打算告诉你一些需要学习的领域和重点，我觉得它们会对你提高自己的游戏设计技能有所帮助。

- 工业设计；
- 写作和表达；
- 脚本编程；
- 三维建模；
- 领导团队的能力。

工业设计强调的是视觉概念 (*Visual Concept*) 的开发，这些概念能够使产品的功

能、价值和外观更为完美。非常凑巧，构造游戏中的各种道具或者对象并组装成游戏场景时，这也正是你想要实现的东西。我试图收集有关工业设计内容的各种书籍。在美国工业设计师协会（*Industrial Designers Society of America*）的网站上（www.idsa.org）有关于该研究领域的更多的信息。

Johnston 编写的被誉为经典之作的《星球大战写生》——*Star Wars Sketchbook*（*Ballantine* 图书公司，1977 年出版发行）是我所钟爱的与工业设计和车辆设计有关的书籍之一。幸运的是，现在有更多的这种类型的参考书籍可以使用，并且大多数都是很有用的。

对于职业游戏设计者来说，写作和口头表达能力是非常重要的“工具”。正像前面所讨论的那样，作为一名高效的游戏设计者，在一定程度上要有将大量的口头与视觉设计信息综合起来的能力，编写成一篇连贯的并且易于理解的游戏设计文档。某些设计者就犯了这样的错误：写成了冗长的、长篇大论型的设计文档，将阅读者的视线转向了一些无关紧要的细节上面，因此使每一个人在阅读这些文档都感到十分困难。在这一点上要相信我——一个繁忙的程序员或者美工是无论如何都不会将他或者她在仓促的一天中挤出来的有限时间花费在阅读那些写得很差且难以理解的设计文档。没有人愿意找这个麻烦。

随着时间的推移和参与开发的次数的增多，你就会更好地理解到底哪些设计细节是有用的，以及哪些设计细节是很重要的。你需要非常正式地将这些设计细节写入到设计文档中，并且让团队的其他成员也可以通过企业内部的互联网看到这些文档。

然而，单靠这些写作、表达和交流方面的技巧并不能结束设计文档的整理工作。你会发现经常不得不亲自将游戏设计思想陈述给各种类型的观众。你可能需要向管理队伍展示开发团队的思想，以保证足够的开发预算。你可能需要将设计的综合结果讲述给自己的团队或者其他在场的开发者。你可能要将自己的游戏开发思想“推销给”一个公司内的很多部门。所有的这些打算都需要你的写作和口头表达技巧尽可能的好。从某些形式上或者在其他形式上讲，你的工作依赖于你要成功地将自己的游戏设计思想“销售”出去。

现在我们肯定已经知道游戏开发过程中掌握脚本技能是多么关键、多么有意义了。花费一些精力和资金来购买几本书籍学习一下怎样使用各种脚本，对我们来说是非常值得的，比如 *Alex Martelli* 编著的《Python 开发手册》——*Python in a Nutshell*（*O'Reilly & Associates* 出版社 2003 年出版，中译本由中国电力出版社出版）或者 *Paul Lomax*、

Matt Childs 和 Ron Petrusa 共同编著的《VBScript 开发手册》——*VBScript in a Nutshell* (O'Reilly & Associates 出版社出版, 中译本由中国电力出版社出版)。在学习脚本编程和展示你的脚本技巧方面你作出的努力越多, 对你来说越有利。

学习构造精准而有用的低精度多边形的模型需要花费相当长的时间和相当多的精力。我总是尽量把它做好, 偶尔也会获得成功。我花费过大量的时间和很多的三维美术家进行讨论, 现在让我们来体验一下: 坐在某些三维软件包前, 让自己感觉到就好像驾驶着太空船要着陆一样。如今的三维应用程序的交互界面做得越来越好, 但是要使这些程序更易于使用, 还有很长的路要走。

如果你是一个三维新手, 我建议从一个免费的应用软件开始起步, 比如 Wings 3D (www.wings3d.com)。我发现该软件作为三维设计的一个起点是很直观的, 而且功能特别强大。它倾向于以一种更接近雕刻的方式来构造模型, 这种方法与进行一系列永无止境的表面拉伸与缩放正好相反, 尽管后一种方法的功能显然也很强大。我是 Bay Raitt 的忠实爱好者, 他在三维应用程序中的贡献就相当于 Nendo 和 Mirai 在早些时候推出了 Wings 3D 一样。图 10-1 是 Bay 为了演示他的雕刻建模法而构造的一个模型样例。

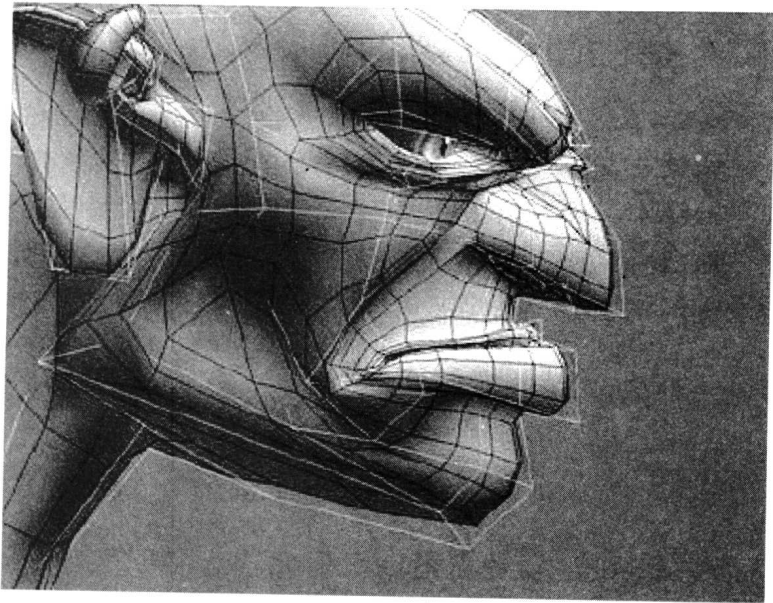


图 10-1 Bay Raitt 的雕刻建模技术

随着你在游戏设计的工作中不断地进步,有一天你可能发现自己能够领导一个设计团队了。我建议你在发展过程中要积累一些纯制作方面的经验。学会怎样分派任务和资源跟踪,能够有效地与不同开发人员进行合作,并且要能够保证游戏及时交付,这些是充当领导设计角色时所需的非常有用的经验。一定要记住其他的设计者和你有着相同的目的。他们希望能够不断学习并参与游戏整体设计所需的各个方面。作为团队的领导,你需要尽自己的最大能力来支持同事们的学习与开发,同时还要维持产品的商业目标和预算。

我这里所提及的每一条建议,都是为了帮助你开始成为或者继续做一名游戏设计人员。在实践中,你一定不要忘记,对于从事游戏制作的每一位开发人员来说,在开发团队中设计人员的确切任务是什么,你都会有与众不同的定义。最好是尽自己最大的努力掌握多种多样的技术,这样就能够迎接出现在自己面前的任何一个机会。这并不是说你必须掌握制作游戏所需的方方面面的技术。你需要集中精力培养自己定义游戏核心的能力。除非你是在编写一个非常小的游戏,否则的话,你将会和一支高度职业化的团队进行合作。



10.2 增长的区域与新的机遇

使自己的技能保持多样化有助于你朝着成为一名独立的游戏设计者的方向发展,你能够独自地创作游戏,而且还能注意到任何平台方面的特定细节。你能够利用现有的各种纸张、纸牌、小块塑料、网格玩具、木偶玩具、钥匙链、无线装置以及现在最流行的游戏硬件控制台来工作。从理论上讲,这为你提供了一个非常广阔的空间,使你能够综合运用各种游戏技能。开发一个纸牌游戏时你并不需要高级的脚本技术,也不需要深入了解作用于粒子系统的各种力场,但是你应当时刻注意到那些能够使任何一个游戏都获得成功的东西——使用所有可能的手段来实现游戏的体验。你在这个领域内的专业知识只是你所掌握的技能一部分。

记住这一点以后,让我们看看一些正在迅速发展的领域以及新的机遇。

■ 微软、梅泰 (Mattel)、英特尔和 LeapFrog

进入到游戏产业中,随着你游戏设计生涯顶峰的到来,你可能想都没想过能在微软开发像 *Barney* 和 *Teletubbies* 这样的玩具,但是作为一名专职的商业设计人员,你会

对任何一个能够展现自己才能的机会和可以继续发展的机遇而感到兴奋。如果这些项目需要游戏构造元素或者管理，你就成为一名游戏制作者：索尼的 PlayStation 2、微软的 Xbox、个人计算机、任天堂的 GameCube、便携机、主题乐园中的亭子、基于位置的娱乐场所（location-based entertainment, LBE）、网络、纸牌、玩具、智能卡、机灵鬼——这些都无关紧要。时时都要牢记，做一名狂热的游戏爱好者和作为商业游戏开发者，它们之间是有天壤之别。更可能的是，你也许发现自己所参与开发的游戏并不是自己最爱玩的游戏，尤其是在你的开发生涯的早期，这种事情更为普遍。要想在游戏产业中持续发展，你所钟爱的游戏种类必须多于一到两种。

就像我们已经看到的那样，玩具与游戏的联姻给我们提供了新的机遇。玩具制造商都知道游戏的力量，而且他们也具有这种开发能力。他们始终都在寻找一些方法将游戏元素添加到玩具中去。微软的 ActiMates 系列玩具，包括 Barney, Teletubbies 和 Arthur（来源于美国公共广播公司的 Arthur 节目），仅是少数的几个例子，它们只是将极其简单的游戏元素添加到了玩具开发中。这些玩具是为年龄较小的小孩开发的，提供了一种哄小孩的游戏躲猫猫，并且还能玩一些简单的游戏。可以肯定地说这些都是非常简单的游戏物品，但是下一代交互式玩具看起来要复杂得多，它们带有完整的游戏程序，同时也为年龄较大的儿童提供了几种游戏。

梅泰利用游戏来引起公众的注意并且支持各种著名游戏的爱好者所组成的整个社团，这些游戏如《风火轮玩具轨道车大赛》——Hot Wheel 和《芭比》——Barbie。它已经注册了这些标准的游戏开发所有权，也围绕着这些著名的游戏成立了基于网络的大型游戏社团。这种类型的网络游戏通常情况下编写成 Java 小应用程序，或者编写成 Flash / Shockwave 应用程序。

尽管英特尔公司已经不再开发基于芯片的 Intel Play (www.intel.com/intelplay/) 系列的玩具，这些类型的个人计算机游戏在某些阶段也使游戏制作者感到震惊。这种玩具包括一个通过 USB 接口连接到计算机上的显微镜，一套虚拟游戏系统，一个语音播放器和一个数字电影制作器，这些类型的个人计算机玩具为游戏制作商提供了几个不同层次的希望。这个市场上还存在着一些相当大的挑战和稳定性问题，但是很多个人计算机玩具的制造商还在设法积累家庭用这方面的游戏和教育素材。

在这种普遍的开发类型中一个非常成功的传奇故事就是 LeapFrog (www.leapfrog.com) 公司。LeapFrog 为登陆到它的 LeapPad 学习系统 (LeapPad Learning System) 上的少年儿童提供了一些简单的基于游戏的教育内容。LeapFrog 非常成功地为少年儿童

构造一个相当规模的知识库。游戏市场中具有教育性的部分通常总是具有一些增值的教育价值,尽管原因是多方面的,这一点却对很多的家长和购买者都表现出了极大的吸引力。对很多开发者来说,这是一个难度很大的跨越,这些开发者要跨进入教育内容之中或者通过开发一些教育方面的内容来弥补他们的收入。

这里所提到的这些具有发展机遇的领域都说明了这样一个简单的事实:独立的游戏开发世界始终都需要寻找新的发展途经。作为一个独立的游戏开发商,不管你是否在开发一个顶级的控制台游戏(要记住,根据发布商的判断力不同,这些游戏的开发资金可能会很快地就被削减掉),你都必须不停地去寻找几种资金来源,来运作自己的开发工作室。很多的开发工作室找不到足够的资金来源。一个开发工作室在其开始发展的前两年就关门大吉了,这是很平常的事情。为什么游戏界的很多开发商都在寻找新的方法以将游戏开发技术应用到各种可从这项技术中获益的产业之中,原因就在于此。

■ 个人计算机玩具所面临的挑战

游戏开发商在向个人计算机玩具制造商制作游戏中会有一些相当艰巨的挑战,这些挑战会限制游戏开发商能力的发挥。首先,在研制与开发过程中个人计算机玩具的设计费用非常高。一旦设计出来后,它们的加工成本也是很昂贵的。这就意味着很难把零售价格定得很低,这样就不会有很好的销量。仅是这一点就使很多的玩具制造商倍感挫折,甚至使他们不敢进行尝试去开发成功的个人计算机/游戏玩具。

当然还有其他方面的不利因素。由于这些游戏通常要比很多其他类型的游戏贵得多,你无法轻松地把它们开发成一个系列并且以系列化的方式销售角色或者玩具。极少有顾客愿意对每个类型的玩具都购买一个,尤其是以现在大多数的个人计算机玩具的成本或价格购买。

另一个实际中存在的很有意思的限制因素和认识上的问题是由于这样的一种观念:爸爸妈妈认为没有必要将 *Interactive Barney* 链接到他们的个人计算机桌面上来增加孩子玩的时间。而且普通家庭的计算机也不是放在年龄最小的小孩的卧室里。如果把这些因素加起来,你可以看出:要想成功地销售出大量的个人计算机玩具,还有一些重要的问题需要解决。这并不是说玩具制造商不会设法用新式产品来打开这一市场;毫无疑问他们会这样做的,并且,大多数的独立开发商并不介意承担这种类型的游戏开发以弥补他们在生产更为传统的电子游戏时的开支。



10.3 广告游戏

对于很多开发者来说，这是一个令人心痛的地方，我也有同样的感觉。你也许不想或者不曾想过进入游戏产业干广告制作的工作，对吗？你深深地喜欢上了《银河战士》——*Metroid*，并且还对自己说：“就是它！它就是我要做的东西！”，是吗？这些就是你想要制作游戏的极好理由。每一个人都希望能够参与到下一代创新性游戏的设计中。事实上，这也是为什么要成为一名游戏开发人员就首先必须经受重重考验的原因。

现在制作的很多（有的人说是“绝大部分”）游戏都是基于授权的。你可以基于某项授权制作一个精彩的游戏，但是对于产权的持有人来说，游戏在本质上就是一个商业宣传。至少，这是一种基于授权的品牌商业宣传形式。

某些类型的公司（这些公司大都是为特定年龄和特定性别的顾客提供产品或服务的）都迫切希望使用游戏来推销他们的产品。如果这种想法会使你感到吃惊的话，这实际上是大可不必的，因为从游戏刚开始出现的时候这种思想就产生了。鉴于为下一代控制台开发游戏的风险和投入都在不断地增加，发布商始终都在寻找各种方法来弥补他们的开支并降低开发的风险。面对单个游戏开发的风险，众多公司大宗买进对大多数发布商来说是一个很有吸引力的想法，而且很有可能成为未来发展的一种强大的模式。

举例来说，本田汽车每天都要在电视广告上花费很多钱，而它在开发基于一系列当前以及即将出现的街头电动车的非常棒的 PlayStation 2 平台游戏也没有投入这么多钱。一个很棒的游戏具有很大的“重玩”价值。业内的很多开发商都不喜欢以这种方式思考问题，但是在玩游戏的过程中，你始终都能体验到这种商业宣传效应。还记得 *Sega Genesis* 平台上的《七喜小子》——*Cool Spot* 游戏吗？这是一款很成功、很有趣的游戏。它是构造在当时最高水平的 16 位 *Genesis* 引擎上的，最终目的是为无酒精饮料“七喜”做了广告。如果你为在电视网络上播放两天时间的广告而花费了两百万美元，这很正常。你曾经有过一次代价昂贵的印象，而现在差不多都忘记了。在不久的将来，当你再去本田汽车的经销商那里购买摩托车时，你也许能够带一个游戏回来，这样你就可以想什么时候就什么时候体验你所喜爱的骑车时的感觉了。

《美国陆军》——*America's Army* 是军方广告商与游戏开发商合作的产物，这正是提到“广告游戏”时我总是要讨论的东西。考虑一下美国陆军的广告预算，开发一个游戏所需的费用对于做好招募工作来说，简直就不值一提。这对于美国陆军来说是无关紧

要的一笔开支。对于急需资金的独立游戏开发商来说，这是一个很重要的资金来源。这也能进一步证明，仅在授权允许的范围内你也能开发出一个很好的游戏来。

我们期待在不久的将来有更多的游戏会借鉴像《美国陆军》游戏这样的成功开发经验，尤其是随着游戏逐渐成为广告的主流、并且更多传统的公司开始相信游戏的强大魅力。在未来几年中，发布商和开发商都将会寻求各种途经与大型的公司进行合作，通过开发游戏来宣传他们的品牌并吸引更多的观众。考虑一下为宣传英特尔与麦当劳而推出的游戏《模拟人生在线》——*The Sims Online*，在这个游戏世界中包含了使用英特尔CPU的个人计算机、大麦克和麦当劳的授权。

现在还没有理由为此而争论……这一点是必然的。原先的游戏产业是非主流的，是为那些在本地的比萨饼餐馆的人群制作的小东西，能够让人感到开心并能消磨时间；而现在游戏产业则成长为一个法人组织的庞然大物。从电视上开始直播超级杯职业足球联赛的实况，到带有“EA Sports”标志的《疯狂足球》——*Madden Football*游戏出现在众多的新式控制台中，游戏已经形成一个很大的产业了。

那么我们做些什么呢？在这种状况下我们怎样开发游戏呢？作为开发者，我们应该把精力放在那些仍然很重要的东西上面、以及那些始终都是最重要的东西上面——开发完美的游戏。游戏中也许不再是只有我们自己的初始角色处于单一的游戏世界中（某些游戏可能还将是这样），但是我们还是能够尽自己的最大努力制作出全新类型的游戏，并且为角色找到进行交互的新机制。角色和环境都是非常重要的，但是当与它们涉及到的各种动作和游戏机制相比时，它们的形式最终还是很随意的。我个人认为，在游戏产业中，改进游戏设计过程的本身就是下一个要进行革新的重要领域之一。甚至是在我已经给你描述过的那种条件下，我们也是始终都需要一个实用的游戏可玩性思想来俘获玩家的情感。

■ 未来有趣的趋势

谈到在不久的将来游戏的发展时，我认为我们需要再看看数家公司以几种方式的“疯狂的”合作。其中某些合作可能使大家都感到很吃惊。就像现代商业中其他各个领域一样，企业影响将会继续扩大，并且大多数的独立开发商将不得不与数量正在减少的游戏发布商联合起来。我们在任何地方都能看到这种事情发生。

一旦有了机会，大多数独立的游戏开发商都会快速地与大型游戏发布商进行深入的合作。从一个独立游戏开发商的角度来看待这个问题，你就不能责备他们的。加入到这

种类型的合作中会有众多的好处,甚至有可能被发布商收购。对于游戏工作室的业主或者合作伙伴来说,这样做是很有好处的,但是而在创造机遇方面,广大的开发商并不总是形势一片大好。但是独立游戏开发商的最终目的并不是创造机遇,而是能够给雇员发放更多的工资。在现在的市场情况下,如果你参与的游戏开发工作没有和前五位的游戏发布商联系起来,那就太悲惨了。

仅依靠游戏内容的可能性越来越小,而与那些希望能将自己的品牌加入到游戏中的公司寻求共担风险资金的可能性毫无疑问倒是增加了。

如果游戏产业还要继续成长、还要构造更多的机会的话,它还需要与旅行服务公司、连锁旅店、推广部门、教育部门甚至是政府部门构造进一步的关系,并且还需要继续加强与其他类型更为传统的娱乐媒体供应商之间的联系。



10.4 具有现实对照物的网上游戏娱乐

在所有的远离网络娱乐的公司破产之前,一些侧重于游戏开发的娱乐公司就已经开始通过“现实对照物”寻求各种与网上游戏有关的想法。这些现实的对照物可能包括自己制作的与游戏特征有关的木偶、玩具、游戏片断、象征符号、宝物、纸牌等等,但不仅限于这些东西。

网络游戏娱乐还要尝试一些其他的想法,比如为玩家支付“工资”,而这些工资在现实中也是可以进行买卖的。作为一个示例,假设有人鼓励你玩某一类型的多玩家网络游戏(很多类型的游戏都满足这一条件),围绕着玩家社区构造某种联赛环境,玩家通过玩游戏累积点数和增加自己在联赛中的声望。这些点数可用于玩游戏,还可以兑换奖品和小礼物。这会引发几个安全方面的问题,但那完全是另外一回事。

玩家挣到的点数可以兑换成奖金或者转换成零售店中一定的购买力。这种想法试图为参与游戏的玩家创造出额外的价值。谁不愿意在享乐的同时又赢得了很酷的物品呢?这种想法几乎是好到都没法实现了,这也许就是问题的所在。

这种类型的实验在某些网站还在进行中,如 *pogo* (www.pogo.com), 它们欢迎所谓的偶尔玩一次的玩家也加入到游戏之中。当看到 32 655 个玩家在纸牌游戏中努力对抗的时候,我自己也感到很着迷。你确实需要了解你的玩家。那么运行在 PlayStation 2 平台上的《极品纸牌游戏》——*Extreme Solitaire* 会怎样呢?

不管你是哪一类的玩家制作游戏,你的想法都是要扩大自己的玩家阵营。现实对照

物能够扩大游戏领域吗？它们可能仅仅会吸引部分的玩家。这正是很多人试图回答的问题，而我不仅仅是说在你玩游戏的同时还将《无尽的任务》——*Everquest* 中的木偶放到你的显示器上的事。我是在讨论现实的游戏元素——这些元素直接与游戏体验相关联，并且有助于增进、支持和扩充这种体验。

■ 走向在线的玩具和纸牌游戏

玩具正在把游戏作为其在线部分，当然纸牌游戏也一样。其中某些游戏变得越来越流行，就像纸牌游戏在纸牌游戏玩家中那样，而另一部分则是在构造在某些家喻户晓小道具上的纸牌游戏和网络游戏，比如《哈利波特》——*Harry Potter* 和《龙与地下城》——*Dungeons and Dragons*。现在的游戏通常都是一个授权软件包的一部分，认识到这一点非常重要。如果你举办了一个风行一时的展览，如《方块小子大冒险》——*SpongeBob Squarepants*，那么由该展览产生的主要收入将来源于产品的授权。你需要获得从夜明灯和装裱画卷的画框一直到视频游戏的每一种产品制作的授权。

对于自己最为成功的纸牌游戏，你希望能够通过提供在线版本来维持并壮大自己的玩家阵营。一个很好的例子就是《魔法风云会》——*Magic The Gathering*。在线设置为纸牌玩家游戏提供了各种构造游戏的机会。这种游戏面向的是较小的玩家群体，因此你不会看到大量的在线纸牌游戏出现。

发布一个像《魔法风云会》这样一个实实在在的游戏会遇到很多新奇而独特的挑战。那些被玩家认为是现实而便携式的游戏换成另外一种在线形式是不是也会获得巨大的成功呢？这一点尚不明确。只要有很多的玩家利用自己的实际购买能力来支持这种游戏的开发，那么人们将会继续把精力投入到这个领域中。但不能保证那些游戏会获得成功，而且不管它制作得有多么好。

同时还要牢记，在数字时代编写游戏的众多成功的游戏设计者都能够归结为纸上游戏和纸牌游戏。而另一些现代的游戏设计者则很少或者根本就没有这方面的经验。

在刚过去的一段时间里，很多盛行的玩具特许权都打算与游戏和在线元素通过某一点结合起来。到了后来，由于当前经济条件的限制，这方面所做的大量工作都放慢了步伐或者完全取消调了，因为各公司都将精力集中在知名游戏和可重复的游戏开发上，而减少了进行尝试的耐心。

像丹麦的乐高这样的玩具经销商，它在经济衰退之前成功地开发出了“智力风暴系统”（*Mindstorms*）系列的游戏（<http://mindstorms.lego.com/>），能够利用它们的在线商

品发展和壮大它的爱好者阵营。

在未来几年中,游戏开发者将会有新的机会参与支持各种类型的玩具游戏和纸牌游戏的游戏要素的构造。毫无疑问,需要游戏开发者进行技术支持的各种极为成功的玩具游戏和纸牌游戏的数量和种类也将会不断增长。

■ 新机遇对开发者的重要意义

控制台、个人计算机和 MMOG 游戏的开发总是充满挑战性并且令人兴奋,这些都是游戏主题中的“蓝筹股”。为了争夺在这些平台上进行开发的合同,其竞争是非常激烈的,并且很多开发商都开始注意到自己能够争取到的开发服务机会正在不断地减少。

对于发布商就是这么一个事实。伴随着一个价值 3,500,000 美元的游戏开发和升级过程会出现的巨大风险,由于受到这些风险的限制,仅有少数的游戏能够投入生产,而且那些确实投入生产的游戏在范围、轮廓和规模上也正变得越来越大。发布商现在是在选择开发四个高质量的游戏和很少几个开发实验,而不是去开发三个高质量的游戏和七到十个一般的游戏——不管由于什么样的原因,他们认为其他的游戏可能存在着问题。

为游戏开发商寻找新的机遇对于游戏产业的成长和未来的发展都是极其重要的。我敢肯定你能够理解我在这一点上的反应。寻找一个能够创收的“轻量级”的游戏开发工作与几个控制台游戏的开发一起进行,这可以使参与开发的员工完成一个大型的游戏之后在较小的项目开发上得到放松。

构造游戏是很艰苦的一件事情,在这一点上不要有任何误解。如果你完成了一个开发周期长达二十六个月的游戏开发项目,中间没有休假,周末也要工作,个人关系很紧张,身体状况还好,并且精神毅力较差,那么你就需要利用一周的时间到斐济去自己给自己搞一个庆功活动了。

要是游戏开发没有向新的领域发展,那就会没有足够的机会来扩大那些游戏爱好者的数量,这些热衷于游戏开发的人们就无法指望把开发游戏作为一种职业。



10.5 游戏开发前线的趣闻轶事

有一件事可以肯定:当你在游戏开发产业中呆一段长时间的话,你肯定会遇到一些特别怪异的事情。在这一点上我先澄清一下,我并不是说这些“狂热的创造者”作出了离谱的或愚蠢的事情:在庆祝自己三十岁生日时他们先在大厅里面骑着单轮脚踏车兜风

或者带着 Nerf 玩具在市场狂喊大叫，然后再坐着他们的宝马车去吃寿司。我说的是确实很怪异的员工——你是知道的，比如精神不正常、口吐白沫、用玩具口哨吹着“*goofy goes goo-goo*”的那样的员工。

这样的好故事我半个都不可能告诉你，我是不能这样做的。有些时候我是乐于这样做的，但是不是现在。他们涉及到的问题确实妙趣横生，因此好像不能在这样一个公共论坛上进行讨论。会有人不记名投票时反对我的。不开玩笑了（“小乖乖，你会幸运地找到把披风系到蝙蝠侠木偶娃娃的工作”）。恐怕那些最为离奇的事情你自己也会经历的。

毫不夸张地说，单单就制作游戏这一件事情所产生的压力，就足够使任何一个正常的人感到发病。在游戏开发的火坑里面工作很长一段时间后，你就开始很严肃地讨论水晶的医疗作用和神秘的远古时期的宗教。你会想知道为什么不能把制作木偶作为业余爱好。你会发现自己在去看牙科医生时仅是坐在鱼缸前发呆。在你完成讲演（讲演的主题是《菲力猫》——*Felix the Cat* 中的某一片断，其中想要表达一下对资本主义、道斯托耶夫斯基以及社会结构的看法）之前，有人就离开你的办公室时你会感到恼火。当你身边的人把一些你从来没有见过的化学物质塞给你时，你会毫不怀疑地喝下去。这其中的部分原因是来源于你处于一种单纯的沉睡缺乏警惕的状态。

下面是我能够和你分享的几个小故事。作为一名游戏开发人员，有时候觉得就像是在黑夜里用自己湿漉漉的手指去摸索一个带电的插座一样。

■ 考虑钓鱼游戏吧

1994 年，在加利福尼亚的马利布大火正值高峰时期，我们团队正在开发一个钓鱼游戏。一天，当我正在办公室里把一个 *Sega Genesis* 游戏烧到可擦写可编程只读存储器中时，我听到从大厅中某处传来了低沉的嚎啕声。我决定过去看一下。当我走近的时候，那声音变成了一种痛苦的呼叫声，从一阵阵牢骚和混乱的咒骂声中传了过来。我们的办公室在山林之中，因此某种野兽发出这种声音也没有什么奇怪的……尽管我很快就作出判断这家伙竟然骂得像我的 *Uncle Burt* 一样难听。当我到了那声音发出的地方，我才认识到这根本就不是什么被困的野兽；是一个崩溃了的程序员。我很快就明白一个道理：崩溃了的程序员就像被困的野兽那样脾气暴躁、浅薄。如果我们是在基督教青年会（YMCA）的游泳池里，他会试图把我淹死。“小伙子，发生什么事情了？”我问道。在陷入如此窘境，他那苍白而高傲的脸对着我，小声嘀嘀咕咕地说到“我快死了”。

嗯，这简直太糟糕了。我让自己平静下来。我急冲冲地去找办公室经理——我吃了太多的 *Ding Dongs*，以至于一直坚持到凌晨三点。同时对游戏的全身心投入也在某种程度上削弱了我的“冲刺”能力。我现在就打算多到体育馆去锻炼锻炼。我们打了 911。911 的调度员说，“先生，所有的救生队员都忙着救火呢。”我快速地吸了一口气并说了类似这样的话：“听着，如果我们的程序员死了的话，我们就无法完成我们的游戏，而且我们已经落后了好几个阶段了。我认为他注释的代码也不是特别的好。有时人们会提到意大利面条，可现在还不到午餐时间。请派一个人到这儿来一下吧！在我和你说话时他的脸色变得更难看了。”调度员向我保证：救生人员一从火灾现场赶回来就会马上过来的。

我把团队中的几个成员集中在一块商量了一下，然后四处寻找看看有没有放弃医学院的学习机会而成为材质美工的人。也许有人以前研究过脸色苍白这样的症状。没有一个，奇怪吗？

在我们寻求帮助的过程中，那个崩溃了的程序员还在不停地嚎叫、呻吟，这就说明他也许不是在故意假装成这样的。大约四十分钟后，办公室的门被冲开了，接着一些身材高大、身着黄色工作服、满身都是烟灰、弥漫着焦油味的消防队员闯了进来，抱着鲜红色的氧气罐。我们立刻带着他们去看那位脸色苍白的朋友。

他们把我们那位脸色苍白的同事放到担架上，并装了吊瓶和监视器。正当我们在办公室的前门走来走去的时候，产品开发的副经理吃完午饭回来了，还很可能打了一两局高尔夫球。他看了一下轮床上那位脸色苍白的程序员，接着又看着我（制作人的那种目光），眉毛竖起的姿势好像在说“我们还没有落后了几个阶段，是吗？”他又看看那位脸色苍白的程序员，程序员现在已经安置好了，正准备进行治疗。他拍了拍坐在氧气罐上的消防员，拍打的莫尔斯电码意思是：“各位，把他抬到一边去。”同时只是简单地说到，“考虑钓鱼游戏吧。”

结论是：你可能快要死了，但这也不是你停止思考钓鱼游戏代码的理由。我张大了嘴，这是不是太残忍了？要是有人能挤出一点儿时间把那些过期的桔子汁扔掉，我们是完全可以避免这种混乱场面的。很明显是那些变了质的桔子汁搞得他脸色苍白。

■ 惨剧

在游戏开发工作室的办公室周围，你通常都会听到各种各样的噪音。你可能会听到正在编辑或者正在合成的音效或音乐播放，与可能会听到美工人员与其他重要人物进行

一些很有趣的交谈。你也可能会听到罐装可乐打开的声音。任何时候，都会有人蜷缩在一个角落里面或者钻到桌子底下睡觉。这里的卫生状况一直值得怀疑。甚至你可能会听到脑袋后面的某个阀门的漏气声。不要紧张。你并不是沙滩上的一个“装饰品”，你只是确实需要和某个人出去度假了。当我赶上度假的时候，我通常都会冲出这个牢笼的。所有的开发人员都需要知道，当压力表的指针到达了那一小块红色信号区时应该怎么做。要密切关注团队中的成员。如果你看到有人开始“融化”了，那就把他们放到电冰箱里再重新冻到一块。

在某个特定的日子里，在某一位程序员和另一位制作人之间发生了激昂的吼叫声与爆炸般的声音相对抗。程序员的任务就是为游戏构造代码资源，这不值得我们羡慕。制作人的任务就是处理游戏开发中方方面面的事情，这些事情都是程序员或者美工根本就懒得关心的，这也不值得我们憧憬。这些总的“方方面面”通常都会掉在地上，弄得地上一片狼藉，除非是有一个好心的制作人时不时地收拾一下。可以肯定地说：所有的开发人员之间都存在一些摩擦。程序员可能会有这样的心态：“我是负责代码的小人物，我只要看看我的任务就行了”。而制作人会有这样的想法：“我理解这个，今年我希望你能够抽出时间为 *Star Trek* 起草协定，但是我们还有商业债务”。

随着两个派别整天不断地发生冲突，有一天当问题最终好像得以解决，事态就会平息一小会儿。刚平静了不久，我们突然听到一些轮胎发出的刺耳声，这不像是有人在大厅尽头的游戏室里玩“暴力赛车”——*Twisted Metal Black* 游戏。我们还不知道出了什么事，我们就听到玻璃、木头和瓷器的破碎声，我们在想为什么整个墙开始摇动。那声音听起来就像是一枚导弹击中了这座建筑物的前部。我冲出了自己的工作区，跑下走廊并绕过墙脚。我看见一个咖啡壶在地上滚来滚去。事情还没有到此结束。我自己没有 *Underdog* 杯。我想：“该死的家伙，这简直就是浪费了。你们会把地毯上的绒丝弄到咖啡中去的！——待会儿我们会需要这些咖啡，因为今天晚上我在离开前必须要完成另一套概念验证。你们这些蠢货。”

确实是一枚导弹击中了在建筑物前面了。后来证明这是坐在角落里的卷发头程序员发射的。很明显，他那被压抑的情感爆发出来了，正在播放的那种立体声的竞技音乐充满了他头脑中的不满。他已经受够了这个游戏开发的生活，因此他决定开着自己的吉普车直接穿过接待处冲进咖啡馆里。天哪。我的意思是，我们并不是真的关心接待处，会有保险公司赔款的，但是难道他不能转弯绕到咖啡馆吗？这显然是不怀好意。

他伤得并不很重，在经过了几次心理调整后，他又回到游戏开发中，并在很多公司

工作过。但是以后的开发商要注意了：这可能不是你努力想要的那种工作流程。



10.6 对设计人员起决定作用的想法的案例研究

要是你在游戏产业呆的时间很长的话，你将会经历几个繁荣与萧条的交替循环，而在你之前的任何人也都会经历同样的事情。在该产业发展的初期，没有任何一个人可以肯定你一定能够靠游戏制作而维持生计。白天，你为某个城市的周末打折程序编写数据库，而夜里则进行游戏开发。

在走完游戏开发的漫长道路的过程，你只有一个盟友：各种各样的方法。而自己掌握的相关技能必须是用于游戏开发的（有点一语双关），但与此同时，你需要保持很强的适应能力，并能够将自己的技能应用到数个并行的工作中。繁荣与萧条总是相互联系在一起，因此你可以感觉到这两种极端情况都是很特殊的。当你不断地看到一些新闻中说，就和你在当地同一个街区的游戏开发商赚了数十亿美元，你不要被它欺骗了。这些数字都是个人的想入非非，开玩笑的，甚至是值得怀疑的，情况就是这样。游戏开发人员并没有分享到游戏产业收入中的最大的一块“蛋糕”。

别误会了我的意思；确实有一些开发商赚到了很多的钱，但这也是相对而言的。据我所知，有很多的开发商工作非常努力，为所有你能够叫的上名字的各种平台开发了大量的游戏，并不像你所说的那样他们家里堆满了钞票、一到暑假便开着法拉利去打高尔夫球。游戏开发是一项艰苦的工作。如果某一“小撮”人拥有法拉利，他们通常会开着它到办公室，连续紧张地工作十四个小时去开发一个 BSP 剖析器或者类似的一些东西。如果你想在游戏产业中成为一名具有竞争力的开发商，那你就不能有任何的疏忽大意。

请记住，作为一名游戏设计人员或者游戏开发中的其他角色，你是在为游戏开发工作室奉献自己的技巧、服务和能力（不管这些工作室与游戏发布商之间有没有直接的关系）。你自己的命运在很大程度上是与游戏开发工作室的命运关联在一起的。不管由于什么原因（并且还有很多可能的原因），如果游戏工作室丧失了一份开发合同，随着合同的丢失，给你的工作岗位所提供的资金也会“飞走”。在像游戏产业这样竞争激烈的产业中，这种情况会接二连三地发生。

我曾经获知，有些开发商连续参与了三个被取消了的的游戏开发项目。如果这件事发生在你身上，就会出现一些问题，因为对于那些现在已经是你的潜在雇主的开发商，他们会怀疑是不是你在被取消的游戏项目中出了某些差错（如果有的话）。最奇怪的是，

他们开始责备你，即使是在可能导致游戏被放弃的各种原因、动机、个人利益的冲突和商业决策方面绝对地和你没有任何的关系。

某些情况下，即使你已经尽了自己最大的努力，你可能还会发现，在实现最佳质量的游戏时自己的技能会受到一些负面影响，甚至走到了穷途末路。好了，遇到这种情况，你需要为自己的生存而变通一下了。我也见过一些游戏设计人员，他们在其他领域也过得相当快乐。而另外一些人则可能跟着游戏产业的发展随波逐流。对他们来说，负面影响胜过正面影响的日子到来了。如果自己四十岁了，就很难过得像是才二十二岁那样，即使你自己能够这样做，但你身边的人也不会接受的。这也并不意味着过了四十岁就没有机会了，就不能成为一名成功的游戏开发人员了。你能够做到的，但需要作出一些牺牲。

还有另外一个因素需要考虑。当项目预算很紧张的时候，独立的开发商很少会在职工培训上投钱，这已经不再是什么秘密了。在很多情况下，如果你想使自己的技能跟上形势的发展而使自己有竞争力——要是你打算继续工作的话，你最好还是要做到这一点——就需要你个人交钱参加培训了。不要再购买赛车了，将这笔钱从车库里面省出来购买短期国库券吧。你还是将其他事情放在一边，加强自己在技能方面的培训吧。

一个游戏设计人员的工作将会是什么样的呢？在不同的场合，这会有不同的含义。有些团队进展得相当顺利，他们相信自己的设计人员能够非常及时地完成一些关键性的设计细节。有的团队运作时，各个成员都各自负责很多的设计细节，而设计人员更像是处理其他任务的一个设计协调人员。随着时间的推移，开发团队会对他的团队成员构造起信任关系。那些能够联合在一起并且知道应该怎样相互合作的团队是很受欢迎的，其原因就在于此。

将特定团队的设计相互协调在一起简直就是一个剧烈的“化学反应”。要知道，作为一个商业游戏设计人员，你的工作就是维护并帮助定义游戏的设计目的。这通常是一个特别强调合作的过程。你不是在一个孤岛上工作。可能会需要你做一些以前没有人做过的工作，而这时对于一个特定的项目或小型项目你可能会感觉到自己是在一个孤岛上工作，但是一般情况下你都需要与自己身边的每一个人通力合作。你并不是在创造个人的工艺作品。在很多的游戏开发商相互竞争的过程中有些自私自利也没有什么坏处，而作为设计人员你最重要的一项技能就是走出自私自利的雷区。

你必须朝着貌似相反的方向为设计思想找到一个快速的解决方案以及解决方法。（比如，对于某个给定的授权，*Herb* 可能希望制作一个面向动作的角色扮演游戏，因为那是他最喜欢的游戏类型，而 *Virgil* 则希望将一些解谜活动加入到第一人称射击类游

戏中，作出一个解谜射击游戏。)你需要能够快速意识到：这些设计派别会把你的设计团队分解成一些小碎片，而且自己要尽最大努力不要作出片面的选择。这没有什么偏向这个、偏向那个的问题。你必须要以体育运动队伍的管理理念体系来处理这种情况。你必须做各种宣传，并且要加强这种思想——思想不统一就会导致失败——而你决不能失败。我见到过无数的团队在游戏开发的道路上拖拖拉拉，原因就是“团队”成员中间存在着很荒谬的思想不统一的情况。

为什么会出现这些很荒谬的思想不统一的状况呢？嗯，你可能不会相信有多少的专业游戏开发人员会认为游戏开发是展示个人表达能力的天地，是为某些怪异的毫无意义的游戏竖立声望的一个个人特有的微小机会(成功的作品是不需要进行介绍的——作品本身就能说明问题)。这种“自我展示”的思想再不能和商业界线及现代游戏开发情况相抵触了。

而另一方面，某些开发人员确实克服了这种情况。他们知道这样做要付出什么样的代价，并且已经做好要为团队付出一切的准备。他们从来都不是只顾自己，也从不陷入片面追求之中而不顾其他的一切。在管理游戏工作室时，这些开发人员就是你需要留下来的。我会要一个任劳任怨、多才多艺的成员，任何时候都具有团队道德规范而不作自私世界中的所谓“超级明星”式天才的成员。误用一个“超级明星”能够在一顿饭的时间里毁掉整个的游戏开发项目。而五位多才多艺的具有团队规范的开发人员能够连续辅助完成五个游戏的成功开发，而且还能在凌晨三点半的时候踢会儿足球娱乐一下。

实际工作环境可能会与你所加入的开发商的规模大小有关。较小的开发商，比如说十五个到三十个人，可能会在某个狭小的区域里工作，并且可能处于开发工作室早期的发展阶段。他们随时都有可能用完自己的资金。你甚至都不想问一下自己的薪水将何处得来，最好也不要知道是从哪里来的，这样你就可能不会被传唤了。

规模较大的开发商或者与大型发布商联合的开发商可能具有更为宽敞的工作环境。然而，和其他产业中的大型公司一样，在一个多变的市场环境中快速地作出决定和反应以确定一套新的指导路线是非常困难的。大型公司可能在完成一个最为简单的任务时也要召开几次大型会议，进行多层次的论证，以及大量的机器模拟。这通常都是在作出某种权衡，这也没什么。

小型开发商提供的工作环境通常具有这样的优点：一个组织严密的、有着家庭氛围的环境。你有可能忘了这些。在很多情况下，小型开发商为了维持生计不得不拼命挣扎。你可能不会得到任何额外的补助，而且你的薪水可能会上下波动很大。你也可能根

本就得不到任何报酬，他们可能只是告诉你把你的电脑搬回家并称其为结清两记。最有可能的是，你会具有更大的自由和空闲去发展自己的兴趣。

规模较大的公司可能会有额外的补助，而且你的薪水也会比较稳定，但是你会丧失某些自主权。还有，在这些规模较大的公司中，有人可能会要求你对一个项目作出某个方面的工作，而你在其他项目中的工作将会受到严密的关注和监视，因为项目的预算并不是按正常情况进行分配的，虽然完全可以按正常情况进行分配。在内部开发一个项目时不管你是为哪一个团队效力的，支付给你的薪水都不包括在他们的预算之内。如果你是在一个规模较小的开发商那里工作，你可能需要完成几个相关领域（场景布局，角色操纵）的工作，甚至同时为几个相关的游戏出力。

和在很多其他的产业中一样的，你所处的地理位置会影响你拿到的薪水。当你在世界各地开发游戏时，你可能不会、也可能会得到自己生活调整时所需要的费用，这样就能保持一个恒定的生活水平。你可能在饱受饥饿之后又过上富裕的生活，或者正好相反。你将会遇到很多很多伟大的人，他们充满激情，并且和你有着共同的兴趣。

这样做值吗？当然值了。我们可以去卖保险。（卖保险这职业没什么不好的，只是我没有必要这么做。）

要集中发展自己的技能，要保持多样性并且要灵活多变。如果你喜欢角色操纵工作，那就成为本国最厉害的角色操纵人员，要记住，当你做这项工作的时候就不必再用其他的角色操纵人员了。如果你是一名很好的程序员，那就恭喜你了。比起我们中的其余几个外，你可能不太熟悉本国那些处于失业状态的开发部门。要坚持下去，但不要令人感到讨厌。我知道这是一个很好的平衡，大多数事情都是这样的。为了维持自己的游戏设计生涯，上一个法律夜校还是一个不错的选择。实实在在的木工家具制作经验也不是一件坏事，除了完成一个医学学位外还具有一些动人的其他经历也是很不错的。

如果你是一个七年级的学生，喜欢游戏，并且还要处理那些“七年级学生需要做的事情”，一定要坚持下去。要集中精力将自己的兴趣爱好和激情转变成各种技能，这样就可以增加你进行游戏开发的雄心壮志了。我在七年级的时候就忙于制作游戏，而且现在还是这样。你将能够为各种控制台以及对我来说难以理解的设备开发一些游戏。他们可能会在两个游戏任务之间发动你的汽车和加热一下咖啡。如果你七十岁了，游戏产业也会需要你。也许没有人会去开发你所喜爱的那种类型的游戏，但控制台游戏也并不是惟一值得去制作的游戏。过去所完成的某些最好的游戏并不是从那些结构性很差的设计文档开始的，他们起始于一种思想，什么类型的动作操纵起来更有趣，他们源于一种感

觉以及某个人说过的话——“我喜欢制作一个什么什么样的游戏……”。

在二十四岁时你不要停止开发游戏，而且也不必到了二十四岁才开始开发游戏——不要管在找工作面试时他们和你说了些什么事情。

七年级的学生，七十岁以及位于二者之间的任何一个年龄段，我都为你们那种努力学习更多的游戏设计知识的兴趣而喝彩。我不能再等了——我要玩你为我们制作的游戏。

■ 形成自己的观点、保持自己的热情

如果你开始进军游戏产业了，那么最为重要的就是给自己制定一定的目标。“在游戏产业中找到一份工作”并不是我所要讨论的那种类型的目标，但很明显，这是这个过程的一个组成部分。如果你能持之以恒、保持灵活多变并且充满热情，这样的话，要是你还没有进入游戏产业中，你会找到进入该产业的途径的。当我谈到目标时，我并不是说要作出某种选择——是成为一名程序员还是成为一名材质美工。我是指自己的个人决定，比如你要在游戏产业中要完成什么样的工作、你是怎么进行自我定位的、以及在游戏产业中工作一段时间后你希望得到什么。

你必须自己决定要在游戏产业中要完成什么样的工作，但是也要记住享受一下这个“旅程”。就是这种旅程会使你回忆起过去，并且会让你比在交货那天笑得还要开朗。而在各个里程碑之间将是野餐时的烧烤和路途上的旅行。如果你的游戏经过十九个月的开发还没有完成，如果该游戏被取消了的话，你的团队将会经历一小段伤心的时期，但是要相信我，你会经受并能度过这个时期的。我见到过一些开发人员为了完成游戏而搞得精神崩溃。不管怎么说，这么做都是不值的，而且当你带着极大的希望和兴奋出发，这其实是一个很悲惨的结局。

我可以告诉你，很多开发商进入游戏产业就是希望能够开发出“历史上”最为成功的游戏版本。一些人在他们游戏开发生涯中的大部分时间都在思考同一个特定的游戏。这是一个很自然的开端。其他的一些游戏开发新手梦想着有一天能够拥有自己的游戏开发工作室。这也是另一种特别自然的激励方式。要确保在游戏产业中你希望完成的理想要在自己力所能及的范围内。

有些开发人员利用自己在游戏产业中的合作关系或工作对自己加以“定位”。如果在一个鸡尾酒会上不让他们提及他们是在这样的产业或者这样的游戏公司里工作，他们可能会显得无精打采并且要去死。你是绝对可以为自己的工作引以为豪的，

但是别犯这样的错误，利用游戏对自己进行定位。当你在十九或者二十二岁的时候找到第一份游戏开发工作的时候，由于能为自己的梦想而生活，你就有了一种很兴奋的感觉。到了你为自己的第十个或者第二十个公司效力的时候，比如说你为“*The New Adventures of Bozo*”制作材质的时候，可能听起来就不再那么动听了。

要记住，不管你是一名程序员、美工、设计者、制作人、工作室主管、音效工程师还是测试人员，你都不应该只用自己与游戏之间的联系对自己进行定位。要是不这样做的话你会成为一名更好的开发人员。有时候，你的生活必须要能够脱离游戏并把游戏开发以外有用的东西带回到游戏开发中。很多开发人员会犯这样的错误，他们仅是依靠或者围绕着游戏和游戏工作而生活着，但这并不利于游戏开发的发展。

要留意自己的未来。你应当给自己制定明确的理想和志向，但要记住，即使是最专注、最热情的游戏开发人员，也不能仅限于此。你也不能仅限于做一个游戏开发人员。我见到过太多太多的开发人员，他们为了完成一个游戏冥思苦索而作出了太大的牺牲。为了一个在你决定离婚后第四天就被取消的游戏而牺牲了自己的婚姻，那是很不值得的。有些开发人员能够开发出非常成功的游戏，这些做得最好的游戏开发人员（并不都是这样）都知道平衡才是最为重要的——要在开发目标、时间期限、各种关联等问题上作出权衡。如果一个很协调的游戏其精髓就是作出平衡选择的话，那么一个充满活力的游戏开发人员其精髓也是能够在各种问题之间作出平衡处理。

如果自己的期望太高，那你可能会受到挫折的。我总是喜欢强调这样的思想——在技术上和在进入该产业的途经上都要保持“多样性”，这样做的另一原因就是让你在纠正平衡上能够快速作出改变。

一旦你在游戏产业中工作了一段时间后，你可能会遇到这样的挑战：如何使自己总是充满激情？你可能知道一句古老的谚语：“如果你真的要抹杀自己的热情，那就把它作为谋生的手段。”这其中是有一定道理的。如果你每天（周末也如此）都花一整天的时间来开发游戏，那么即使你能够挤出一些“空闲”时间，你可能也不想再玩游戏了，你可能希望能够到外面去跋山涉水。

游戏产业中很重要的一件事情就是：即使你开发了很多的游戏，但随着各种平台的出现和不断发展，只有一件事情是不变的——你总是能够从与游戏制作有关的领域中学习到更多的东西。为实现自己特定的游戏目的，你可以将范围广阔的各种主题交叉在一起。我发现要保持对游戏充满激情的最好方法就是用新游戏的探索任务向自己和自己身边的人提出挑战。根据自己的计划，我每年都会选择开发三到四个或者更多的游戏。在

你开发游戏的时候你很快就会发现自已又落后了几年。

有时候,这些在游戏开发中进行的探索会偏离游戏的开发。那也没有有什么关系;要知道我们一直都希望能使问题返回到原来的道路上。我要参加一个角色设计班;我会参加一个重要的建模班;我要尽力学习更多的脚本知识;我要参加一个绘画与光照处理班;我要利用自己宝贵的假期时间去参观一个苏格兰城堡。我喜欢那些涉及到旅行的游戏调查。我很羡慕迪斯尼公司的动画师,他们被送到世界各地为影片制作起草纲要、研究和做准备工作。

我发现很多地方都与游戏之间存在有某种联系,如果你还没有发现的话,将来你也会发现这一点的。在进行体育活动、参加音乐会和参观博物馆时我都会联想到游戏。我喜欢参观博物馆书店并购买一些书籍,比如“*Pieter Saenredam: The Utrecht Work*”,作者是十七世纪教堂内部结构的绘画大师(译注:见附录 D)。我考虑将所有这些“调查工作”都用到游戏开发中。我可能很快就需要一座漂亮的小教堂了!我始终都知道在学习上我还有很长的路要走。

你可能会发现,你最终完成的那些最令人兴奋的作品其灵感就来源于自己在增长游戏技能时所做的工作。我鼓励你去寻找全新的方式来发展自己的游戏爱好。我知道你会对自己的发现感到很兴奋的。

与我共同工作过的那些最杰出的游戏开发者都近乎疯狂,他们对自己身边的所有东西都感兴趣,并且你将有机会见到这些人中的一部分。我记得一名美工,他在建模和材质处理方面绝对是才华横溢的;我很坦白地告诉他,他是一个非常了不起的数字美术家。他犹豫了一会儿,视线离开他的画笔——*Wacom tablet*,抬头看着我,又完成了一次鼠标移动的动作说到,“我不是一个数字美术家,我是一名画家。”而我已经忘了他曾经是一名画家——接受过标准的培训。“太绝妙了”,我心里想。他没有把自己完全定位到游戏中,但却把游戏以外的所有事物都带入到游戏开发中了。

BILL ROPER 访谈录



Bill Roper 是 Blizzard Entertainment North 公司的副总裁。作为《星际争霸》和《暗黑》系列这些最畅销的游戏的缔造者,以及即将推出的《魔兽世界》——*Worlds of Warcraft* 的创造者,我想不需要我对 Blizzard Entertainment 详细介绍了吧。我有幸就一些有关怎

样进行游戏设计的问题采访了 Bill。

TM: 你的游戏设计思想是怎样产生的呢?

BR: 你是否曾经冲上高速公路并做着白日梦: 要穿过拥挤的交通, 从路边出发, 一直冲到本地的 7-11 停车场上才刹住车, 这样的话在等待停车的过程中你还可以抓紧时间喝一杯苏打水? 欢迎大家来到《侠盗猎车手 III》——*Grand Theft Auto III*)。你在生活中所接触到的所有事物都可以作为游戏设计一个潜在的出发点。音乐、艺术、文学、电视、漫画册、电影、体育运动、食物和基本的社会之间的交互关系——所有这些事情都能够激发出一个游戏设计者的灵感火花。其中关键的问题就是要找到自己感兴趣、喜欢或者充满激情的那些东西, 并把它们加入到自己的游戏中。不管是哪种类型的游戏主题, 一位特定的游戏机制, 或者甚至是一种艺术的表述, 你所需要的都是指导性的想像。然而, 这种想像, 其焦点必须是真正构造在自己的兴趣之上的。

很多很多的游戏设计人员都知道在制作游戏时需要具备最前沿的技术, 或者是最漂亮的美术作品, 或者有史以来最聪明的人工智能, 或者使用了最奇异、最具创造性的手法, 但却没有看到游戏中也需要具有乐趣。有个观点总是正确的: 制作一个你自己都喜欢玩了再玩的游戏, 同时不要忘了游戏的设计应该是人们能够理解的, 易于介入并且能够投入情感的东西。即使是像俄罗斯方块这样的一个“低技术含量的”游戏也能够使你感到快乐, 着急和局促不安, 因此玩过以后还想再玩, 并且想要打破自己的最好纪录。

还有, 玩一下别人开发的游戏也是非常关键的, 特别是和你的开发队伍一起玩。最好的独奏乐就来自将独奏者的激情与技巧和迷唱伴奏乐队有机地融合在一起。他们聆听独奏者的演奏, 然后为他和音以及弹奏相反的曲调。反过来, 独奏者通过自己的直觉以迷唱或者变调进行回应。当第二个独奏者加入演奏以后, 如果他将自己的演奏构造在前面的过程上, 他的演奏将仅仅是把他刚才所听到的演奏所放大。作为游戏开发人员, 我们不应该害怕玩其他公司制作的游戏并向他们学习, 就像我们不应该害怕玩自己的游戏并从自己犯下的错误中学习知识一样。应该关注那些曾经出现过的东西, 看看它们现在又在引导新的思想朝何处发展。

TM: 你是怎样对游戏的可玩性思想或者可玩性机制进行测试或原型制作的?

BR: 一个你能采用的最佳途径就是找到某种方法在屏幕上显示那些能够代表游戏运行机制的核心思想。这并意味着你必须首先制作一个最好的图形引擎——只要能够测试核心的概念, 即使是最简单的形状图形也是一个很好的起点。根据游戏类型的不同, 可能有各种方法能够对已有的产品和工具加以利用, 对概念模型进行快速的初始论证。

举例来说,《军团要塞》——*Team Fortress*,就是一个使用已有的游戏和一套工具包构造出来的游戏,并且可利用它们来检查那些在记事本里勾画出的思想是否玩起来有趣。除了声音设计概念和哲理以外,还有一些东西你也不知道它们玩起来是否真的有趣,除非你实际地玩上一次,因此尽早实现这个阶段——哪怕是一个最基本的实现——是很必要的。

TM: 你过去作出的设计决策是如何影响到现阶段的设计决策的?

BR: 历史学家说过,那些不学习历史的人注定要重演历史。游戏设计和开发领域当然也是这样。从游戏机制到技术实现,从进度安排到测试程序,到艺术素材的构造,设计一个新游戏时其所有的方面都要和其他已完成项目中所对应的方面相互比较。如果给最后三个项目中的每一个项目构造一个基本的、可玩的游戏原型要花费四个月的时间,那我们就没有理由假定这一次它将只需两个月的时间就能建好。如果你在最后一个角色扮演游戏中包含了一个内置的自动导航系统,而玩家对此反应良好,他们也就希望在你开发的下一个游戏中也包含一个内置的自动导航系统。还有,如果你进行过的某项尝试而且失败了,那它就是一个必须记住的非常重要的教训。如果在单人游戏战役的故事线索完成以前,就进行关卡构造的尝试,并因而消耗了项目中最后的、为返工所准备的时间和资金,也无法再修改关卡或者重新录制对话了,这就要求确保下一次在作出相同的决策时一定得记住这次的教训。

在每一个项目结束时都应该做一次经验总结,其中既要指出游戏设计和制作中所有决策成功的一面,也要指出其失败的一面,这对未来的设计工作将是一个很重要的工具。还有,要和日常的游戏社区保持密切的接触,观察他们喜欢你游戏中的什么,不喜欢什么,这些在你作出设计决策时将是非常有益的。然而你可能倾向于使用某种特定的游戏机制,如果没有玩家喜欢它(更别说使用它),那么你就需要重新考虑是否还要使用这种机制。这并不是说你会为了将来而自动地改变你过去所做的事情,但是你必须意识到其中的问题并对这些问题进行切实的考虑。当那些制作好的游戏能够作为游戏改进过程中很好的参考时,我们不仅能够从分析这些游戏,还可以从不断地玩这些我们自己开发的以及别人开发的游戏中学到很多有关如何决策的知识。

TM: 你是怎样处理设计团队中的多种设计偏好以及多种设计个性的?

BR: 在这个产业中作为一个制作人或者设计主管就像是在变戏法一样。你需要将众多才华横溢的、相互之间高度依赖的人员投入到游戏的协调开发中去,而且,你的责任是把他们组织起来,从中提取出最好的思想用到你的游戏中去。这和通常意义上的提

取最好的思想有着至关重要的区别；在所有的设计决策中是游戏，而不是团队成员中个人的个性或愿望，具有超强创新性或者需要使用前沿技术，这样整个过程才得以进行，这是很重要的一点。不管在与团队成员交流时你是否老练而精明（而且这两方面都应该具有），只要在他们明白了你的思想和你心目中的最佳兴趣时，他们就会愿意和你一同工作。这也意味着你要能够通过例子来加以引导。如果你有一个设计想法对于游戏来说不如团队中另一个成员提交的想法好，那么你就应该放弃你的想法，你必须能够把自己的想法放在一边，采用他人的想法。

TM: 设计团队会对工具的特性和功能提出一些要求，编程团队要构造这些工具，在处理他们之间的交流时你有什么建议吗？

BR: 如果你能使编程人员和设计人员都相信：在开发一个成功的游戏时工具是其最重要的一部分，那你就已经成功了一半。还有，问题的焦点应该集中在游戏上而不是集中在编程人员“个人的”兴趣或者设计人员的“最小的”要求上。设计人员有责任考虑利用他们所要求的工具打算完成什么工作，而不是仅仅提出要求：“我们需要某个特性”。编程人员有责任和设计人员一起编制一个工具包，它可以使游戏的制作尽可能的方便和快捷。其中一个很大的好处是如果有一个很强大的工具包能够使设计人员把自己所有的思想都搬到屏幕上，那么对于编程人员来说项目的完成就会容易得多，因为他们不必花费时间编写特殊事件的处理代码来处理那些能够为编写得很棒的工具所处理的东西。

作为一个例子，对于《星际争霸 III》来说环境编辑器的设计与实现对于角色扮演这一部分的成功实现是极其重要的，在游戏的单个玩家战役这一部分能够找到。关卡设计人员与编程人员联合开发了一套工具，它不仅功能强大而且在他们的日常应用中也非常简单。尽管很少有工具在构造时没有更多地为最终用户考虑，但环境编辑器的成长就是制作它的人与使用它的人之间通力合作的一次大胆尝试，这是很重要的，不仅仅是因为我们倾向于把它作为游戏开发的一部分，还因为我们知道我们希望能使编程人员尽可能地得到解放，从不得不为战役游戏编写特殊事件的处理代码中解放出来。

通过将大量的精力用在设计一个强大的工具上，关卡设计人员就能够用它来修改他们的作品并在游戏中进行快速预览，这样我们不仅可以使开发流程更为简单而高效，而且还能够在缩减所需的质量认证测试时间的情况下给自己更多的机会对这些设计进行反复修改。据说工匠们最多只能将作品做得和自己所使用的工具一样好，因此给你的工匠提供尽可能好用的工具将有助于他们制作出自己力所能及的最佳作品。

TM: 你有什么好的建议要送给那些游戏设计新手吗? 你是怎样进行准备的?

BR: 开发一个你自己想玩的游戏。和一个不喜欢体育运动的体育类游戏设计人员进行交谈, 或者和一个只玩第一人称射击游戏的开发人员, 即使他是战略游戏开发人员进行谈话, 都会让人倍感挫折。尽管你不可能总是选择那个你愿意开发的项目, 但至少要尽自己的最大努力涉足那些能激起你的兴趣与激情的领域中。同时还要做到不管你正在设计哪一种类型的游戏, 一定要密切关注下面几点:

易玩

这是一个使玩家加入到游戏中的阶段。游戏必须很容易入门并且要具有一个非常容易理解的界面。随着玩家在玩游戏时不断地进步, 他们被给定更多的任务去完成、更多的空间去探索时, 这时他或她就会被引入更为复杂的游戏机制。到游戏的最后, 玩家就能够达到一个比游戏开始时要高得多的水平, 利用一个倾斜角度缓慢增长的难度值曲线, 很容易开发出来这样的游戏。

难精

这是游戏中真正的实质性内容。当玩家知道了怎样进行界面操纵, 怎样到处乱转等操作以后, 什么东西能够吸引他们返回来再玩一次呢? 高级的策略, 一系列的多玩家选项, 以及随机的或者由玩家来构造的地图, 所有这些因素都使得一个游戏具有可玩性, 并能够使游戏可以重复玩上几个月或几年, 而不是几天或几个星期。

有趣

除此以外, 游戏必须有趣。尽管这可能是一句非常深奥的话, 但是当看到设计人员忽视了这些最为基本的设计基石时, 一定很令人感到吃惊。具有高级智能的敌人, 最先进的图形引擎, 或者完美的数字环境环绕声, 这些能够增强一个游戏的功能, 而就其本质来说, 就是使玩起来有趣。然而, 这些东西自身是不能使游戏玩起来有趣的。这也就是设计人员的主要贡献; 要确保游戏设计人员的首要工作是设计一种引人注目的、充满乐趣的游戏体验。

TM: 随着游戏范围和复杂程度的增加, 游戏设计的作用也在增加, 你是怎样看待这个问题的?

BR: 我们要能够在不断进步的技术和不断增加的游戏规模、范围和复杂度之间作出选择, 使它们都在我们的接受能力之内。我们很容易被所有这些铃声、光线和哨声所

吸引,但却认识不到这样的一个事实,某个人坐下来第一次玩游戏时,他对你所设计的东西绝对是一点概念都没有的。你应该让游戏在一开始的时候简单轻松,即使到最后你是打算让玩家敲入大量的文字,不停地移动鼠标,疯狂地点击众多的对象,并平衡各种资源。

还有,我们应该将精力集中在游戏设计中那些使人愉悦、令人着迷的关键元素上,接着就构造这些领域,而不是简单地创造更多的不同思想。作为开发人员,我们的工作就是定义一个游戏体验并真实地实现这种想法。如果我们在这一点上做得很好,这个游戏对于玩家来说就会始终充满乐趣,他们会对我们游戏中所提供的那种类型的游戏体验感到兴奋和愉悦。

最后,我们需要制作并维护一些工具,这些工具不但能够帮助我们构造游戏,还能够跟踪游戏设计开发过程中的进度情况。如果你有一个巨大型的角色扮演游戏,比如说它具有数百个咒语,上千种需求,上万种不同的物品,你不但要能够设计并生产出所有这些内容,还要能够对这些元素进行反复的权衡、分配与使用。游戏的规模越大,就有越多的东西要遵循并要进行安装,因此我们需要开发出一套卓越的工具,以便能以一种即时而有组织的方式更为方便地处理这些事情。说得更通俗一点,如果你有10个交往的朋友,你可以记住他们的名字和各自的电话号码。如果你拥有50~100个朋友,你就需要一支笔和一个记事本,如果你拥有1 000 000个朋友,你就需要一个电话本,一种组织各种操作以方便调用的方法。

TM: 在游戏的近期发展中哪些方面使你感到很兴奋?

BR: 游戏设计人员能够完成的事情以及游戏产业作为一个主流的娱乐产业始终都处于上升趋势,而且没有迹象表明在不久的将来它们的增长会放慢。随着这一代人逐渐地成长起来,他们可能会把个人计算机和控制台游戏作为他们文化生活中的很大一部分,就像过去的一代人把美术、文学、音乐、电视和电影作为他们文化生活的一部分那样。这就意味着人们对游戏的兴趣会不断增加。对那些具有大型游戏创作激情的,多才多艺,专注投入的开发人员将会有一种持续的需求,并且越来越多的人将会体验到游戏的极大乐趣。我们创作并用来运行这些游戏的系统将会继续变得更加强大,而且玩家的体验也会变得越来越好。最好的一面就是在五年后可能会创造出某些现阶段还处于想象中的东西。

TM: 对于大多数的游戏项目,游戏在完成时会偏离最初的游戏概念有多远?

BR: 游戏中的两类相反的却互相补充的力量是保持思想的持久不变的同时能

够放开这些想法。一旦你为某个项目构造起某种思想后——我指的是主要的目的、感觉、游戏机制等等——在进行游戏设计时你需要把它们牢牢地记在心里。尽管相对于初始设计来说将会有不计其数的改变，大的小的都有，但你必须将游戏概念作为讨论的焦点。

在游戏设计的创造性过程中，你将会提出数千种伟大的想法，也有很多不太伟大的想法。有时候你会提出一种注定要成为令人真正感到吃惊的、惊天动地的概念。偶尔，这些想法就是实现不了。不管由于什么原因，你认识到了这些想法需要改变或者就此搁置不用。最简单的一种情况就是，当你最终在游戏中尝试这种方法时立刻就意识到它根本就没有任何乐趣。其他情况下，这可能是一个极其艰难的决策，因为你知道怎样能够让你的想法发挥作用并且乐于尝试寻找各种途径来实现它。

不管在哪种情况下，抛弃掉一种自己构思的、具有培养前途的甚至是能够实现的想法是一个很痛苦但却是必须的过程。并不是所有的东西都能够像计划的那样发挥作用，而且当你自以为会对游戏产生重大变革的游戏机制被证明不是一项重大冒险而是一些烦闷的工作时，接纳这种思想就是一件很困难的事情。当你面临需要将游戏中的某个漏洞堵住的时候——不管是由于什么样的原因——你必须快速地、毫不痛苦地去完成这件事。不用害怕抛弃某些想法，因为很多情况下你会有比它要好得多的方法来代替它，或者只是简单地发现根本就没有必要使用它。

放眼于项目的整体思路，你甚至可以对游戏中的关键元素作出重大的改变，同时知道你最终会用一个内聚力更强更有趣的游戏设计来代替它。只要你能够认识到这两种力量的存在并且愿意把它们两个都作为工具进行使用，这就是它们相遇并发生实际的相互作用的地方。

TM: 对不同的开发人员来说“设计”角色的含义不同。设计人员应该将精力集中在技术方面、艺术方面上，或者是完全不同的其他方面上？

BR: 最重要的是要找出能够激发你创作源泉的东西，接着就集中精力在那个领域中做得最好。看一下自己感兴趣的其他东西，并要知道作为一个设计人员怎样才能把自己生活中的这些元素与自己的工作结合起来。如果你喜欢体育运动并且还对人工智能感兴趣，那就开始制作一个最好的足球训练游戏，它能够进行数字足球比赛。如果你喜欢西部和冒险，那就开始制作一个游戏，其中的英雄人物或者坏人是一个性格怪僻的枪手。如果你有一种想法可以开发出一种从未见过的游戏，那就选用其他游戏的图形和片断把它作出来，或者用其他的计算机游戏的编辑器构造出能够表现出自己的思想的一关。作为一个设计人员，你应该将自己的热情与专业技术集中在最有益于游戏开发的领域内。

MEGA 提示

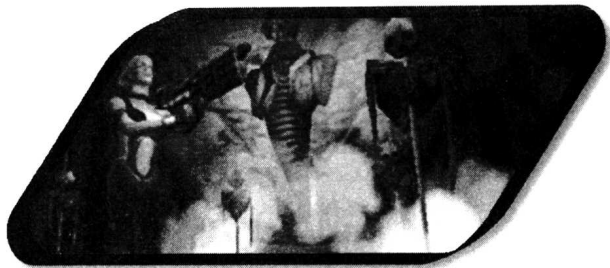
1. 要使自己掌握的方法、技能和概念体系保持多样化。游戏产业是一个艰苦而动荡的产业。
2. 做得最好的开发人员会珍惜各种机会，会“带回”他们个人和集体共同使游戏受益的探索结果。
3. 以个体以及游戏开发人员的身份把精力集中在自己的技能增长途径上，而不要仅仅把自己当作一个游戏开发人员。
4. 你认为哪些领域能够构造出新的游戏思想，或者通过对这些领域的拓展能够提高自己的相关游戏技能，那你现在就可以开始对这些领域进行研究和探索。
5. 要确保将精力集中在游戏设计的基础要素上，包括工业设计、写作和表达、脚本、建模、舞台设计、动态效果、光照处理，以及（也许是最为重要的）要学会怎样真正地成为一个团队的成员。
6. 要摆脱各种约束。不要害怕在新地点尝试玩游戏。
7. 要把精力集中在游戏的有趣性上。作为设计人员，不要由于技术方面的原因而使自己分心。游戏要么就是很有趣因而自然会有重玩的价值，要么就很没趣而没有重玩的价值。在游戏设计最基本的水平上，你必须能够处理这个问题。
8. 不要忽略各种能够扩大自己游戏领地的机会。你必须能够为自己，有可能的话，也为其他人创造机会。
9. 坚持玩游戏。



CHAPTER

11

在游戏开发产业
中择业



当你的游戏开发技术提高到一定的高度时，你必须就你的游戏职业生涯的发展方向作出选择。在今天的游戏产业中，要想成为一名美工、程序员或者音效工程师都是相当困难的，因为每一个学科对专业知识的深度要求都是如此的严格。从真正意义来说，对于许多游戏开发团队，请记住一句有用的话：团队的每一个成员都是一个游戏设计者，或者至少是一个重要的游戏设计贡献者。“游戏设计者”是一个相对较新的职位，而且很多游戏制作者都在尽力确切地弄清楚一个强有力的游戏设计者应掌握哪一套技术。很多程序员、美工和制作人都在游戏设计方面拥有很强的能力，并且愿意和那些承担游戏设计任务的同事紧密合作。另外一些开发人员则希望尽量少地陷入繁琐的游戏设计细节中，他们更愿意专注于自己专长的而且感兴趣的领域。

在本章中，我们将关注几种主要的游戏开发职业的发展历程，并讨论一些选择游戏职业时应该考虑的因素。



11.1 程序设计

要成为一名游戏程序员并不那么容易。那么，再说一次，要得到游戏开发中的任何一个职位都不容易。我经常说，如果你是一个真正意义上的程序员的话，那你就会知道了。这是什么意思呢？它的意思就是说你无法让自己成为一个真正的程序员。也许你有几分喜欢这个职业，或者你一点都不喜欢它。我知道随着时间的推移代码编辑器和程序调试器已经变得越来越好，但是当我学习编程的时候，我感觉并不像是在编程。我感觉好像是为了让程序正确地编译，我把绝大部分的时间都浪费在了寻找放错了位置的分号。如果有一天我没有和计算机上方的“橡胶鸡”挥手示意就将代码编译通过，那一天我就“完蛋”了。

在开始体验游戏编程的时候，我使用 Basic 语言编写文字游戏并把它们录制到放在

TRS-80 Model 1 型号的盒式磁带录音机的录音带上。如果我把磁带卷弄乱了，游戏就不能存盘了。我和一个朋友一起写了几个小游戏，而且我们还动用了街机射击游戏的 Basic 代码，并将这些代码进行修改以获得新的结果。

我想成为职业游戏程序员的强烈愿望在学习大学计算机图形学编程课程的早期就消失了。我用 Pascal 语言编制一个飞镖靶子。我自己几乎不能让飞镖射到靶子上，而坐在我旁边的那个家伙用快速动画说明飞镖射到靶子上的什么位置，而且他使用的是彩色飞镖。该死。刚刚体验完我的编程职业的结束，我知道是该回到哲学系的时候了。我甚至不能想像出为什么笛卡尔看起来是很有意义的，而在屏幕上投掷彩色的飞镖却毫无意义。

事实上，你可以“使”自己成为一个程序员，并强迫自己做编程工作；我就认识几个试图这样做的人。很不幸，你也许没有激励你每天更加深入地工作所必需的热情。作为任一层次的游戏开发者，理解编程非常重要（如果仅仅是概念性地理解）。即使你不想成为一个专业的游戏程序员，花一些时间在编程的课程上也是很值得的。你需要在一个非常基础的层面上去理解面对一个编程挑战会是怎样的情形，即使你面临的挑战是写一个计算你在火星上的重量的程序。或者，老天帮忙，你面临的挑战是制作一个飞镖游戏！

就我的经验来说，游戏程序员需要具备两个核心要素。他们需要深入理解怎样应付三维编程的挑战（用代码、数学和物理），以及怎样通过代码为构成游戏的不同的要素注入活力。他们还需要玩大量的游戏来了解他们所追求的是什么样的游戏内容。我曾经看见过几乎没有玩过游戏的工程师尝试着制作游戏。这不是什么好现象。他们有时会有了了不起的工程直觉，却对如何用代码来实现游戏的娱乐性毫无概念。

收集很多游戏的知识是制作游戏这项工作中容易的一部分，它也是构造你对游戏的理解的工作中有意思的部分。最困难的一部分似乎是对纯粹的基础数学和物理的充分理解，只有这样你才能让你的代码得到想要的效果。

很多人和我讨论过以下的观点：我敢说为了从事游戏编程，在那些日子里，你是不得已才去完成计算机科学或者数学的学士学位（或者自学获得同等水平的知识）。它确实变得太复杂。在和处于巨大的开发压力之下的很多团队合作过以后，我直接了解了他们之间的不同之处。我看到有扎实开发背景知识的开发者，他们在短短两个月内就从头开始完成了三维引擎的开发。我也见到那些没有扎实的开发背景知识的开发者，他们为了要在屏幕上移动一个东西就“努力奋斗”了数个月。

■ 游戏编程作为一种职业选择

游戏编程也许是游戏产业里最佳的职业选择。很多游戏工作室的创立者都出生于程序员，而且他们当中的一部分人仍然喜欢从事代码编写工作。有些人则把编程丢到了一边，而选择专注于游戏开发工作室的运作。按照传统，作为个体的游戏程序员是游戏产业的焦点。在过去，常常是他们赋予了游戏的全部体验。让我们直面这个事实：没有很棒的程序设计，即使有最好的美工、设计或者音效，你也不可能拥有一个很棒的游戏。

如果你是一位有天赋的程序员并且爱好编程的话，那么在成为一个程序员的这条路上你不会有什麼麻烦。这是个伟大的工作。然而，你懂得，作为一位游戏程序员，你的任务是帮助控制游戏项目的代码基础。任何需要在代码中作出的改变都将由你和你的工作组来完成。游戏程序员的工作时间很长，而且这些工作至少来说都是脑力挑战。你会不间断地以最高的效率来解决问题、试验和工作。这是一项棘手的工作，而且你只能喜欢它，不然的话你就不能持续很长时间。好的游戏程序员有相当不错的报酬，但是他们赚的钱出乎你的想像。在多数情况下，他们的薪水比其他产业与他们同等地位的人要少。我认识一些程序员，他们在刚刚开始游戏编程工作时年薪为两万五千美金，并且在短短五年内他们的年薪已经远远超过十万美金。像所有的游戏产业中的薪水一样，对于各种不同的情况，这些数字上下变动也是很大的。

找工作的最好的方法是带上一个游戏的运行良好的代码例子或者是针对游戏的材料。可能你和朋友写了一个很棒的游戏工具，或者你在上大学期间写了一个非常有趣的游戏。好多人就是通过这种方法找到工作的。如果你不具有可以和《毁灭战士Ⅲ》——*Doom III*匹敌的游戏引擎代码基础的话，也不必担心。如果你可以做到，那我们可以交流一下。

游戏编程作为职业的另一个好的方面是你最终可以接触到很多方面的内容。你可以把精力集中于你感兴趣的多个领域：图形或渲染、音效、工具、压缩算法、路径寻找或人工智能，或者软件开发管理。

既然游戏编程是各种类型的编程工作中最难的一种，那么如果你不喜欢从事这种工作了，跳槽到别的部门是很容易的。对程序员的需求可能会时而增加时而减少，但是对于能写高效代码的程序员的需求越来越不受外界条件的影响。

好的游戏程序员总是供不应求。但是你不能对许多其他的开发角色说同样的话。



11.2 美工

游戏需要创作很多类型的美工资源（在第 1 章中我们已经讨论过了）。这里所说的美工资源包括（但不限于）：角色模型、角色材质、场景建模、场景材质、道具模型、道具材质、角色和道具动画、特效、完整的序幕电影、过渡电影片断和结尾电影，当然，还有大量的界面和威力升级细节。

我已经尽量不再讲粗话了，但是曾经有个老板对我说：“如果我是一个程序员我就会走到街上破口大骂并且会抨击一个美工。”确实有点儿夸张，但是有许多有抱负的游戏开发者想成为游戏美工。游戏公司的办公室里总是塞满了演示录像带。为什么有人认为他们可以通过提交一个在银幕上抢夺带有铬黄色材质球体的作品就能获得游戏美工的工作，我永远也搞不明白。

我曾递交过一份用木刻打印机打印的简历和一张没有任何花哨内容的名片。游戏开发商并不是要找美术展示者；他们寻找的是具有美工技艺，并且能够创作出有助于游戏销售的游戏资源的人。就是这样。这并不意味着你不能是一个美工。事实上，你应该是一个美工，但是你应该知道怎样走出你的艺术世界并在游戏产业的商业世界中“拼搏”。

首先，作为一个游戏美工，游戏开发商需要你要有建模、材质和动画技能。好的美工应该对于如何准备用在游戏中的低精度多边形图片有完整的理解，开发商通常都需要这样的美工。但是，学习怎样灵活地制作这种材料要花费很多的时间。

作为一个游戏美工，别人会要求你提供制作用在游戏中的多种类型的美工资源，因此你要能把自己表现得多才多艺。想一想，所有你喜爱的游戏和你曾见到的制作好的游戏类型，它们有什么共同点？它们的特点可能是一些幻想风格或者卡通风格的英雄和敌人。它们可能提供很多种类的交通工具。它们可能布置在一个游戏环境中。

那个老板还告诉我：“每个游戏的背景都可以追溯到外层空间、沙漠、丛林、城堡或者城市。”我想你可能已经知道游戏需要什么样的美术需求。集中精力于图画、建模、材质处理或者动画制作这些具体的要素。如果你想有一个长久而深入的职业，而且你还想着转行干点儿别的工作，你应该开始就打好坚实的基础。去取得艺术或设计的文学学士学位或者美术学士学位。绘制大量的人物、车辆和动物。你将永不停止地学习，由此构造一个坚实的开始确实很好。

■ 游戏美工作为一种职业选择

游戏开发商总都是需要好的美工。现在对这个位置有兴趣并且为这职位作好了准备的人数已经开始超出了这个职位实际需求。这是一个竞争激烈的领域。如果你在吃饭、呼吸和睡觉的时候都在想着制作面向游戏的美工资源，你可能会把这份儿工作做好。

如果你在一个游戏开发商那里找到了工作，成为一个游戏美工是一个了不起的工作。如果你喜欢画图或者建模和动画，做你喜欢做的事情而获得报酬是值得的。不错，这种感觉很快就会过去，而且，与别的游戏开发角色一样，你必须要喜欢你所做的工作，这样才能为它做出必要的牺牲。

游戏美工主要负责制作美工资源。现代游戏开发需要大量的美工资源。完整的软件包为管理所有这些资源的工作流程做了不少的贡献。这些资源通常都需要修改、重做甚至是全部删除。有时一个小的美工组要负责创作数百的或者数千的美工资源：每一个小图标、光标或者屏幕要素，每一个单独的威力升级、炮弹类型和炮弹效果。

创作所有这些要素是一项非常辛苦的工作，而且还要在截止日期前全部做完。游戏美工常常要工作到深夜。作为一个游戏开发商，会对他的个人关系产生不良的影响。游戏美工收入一般来说比程序员低。一些美工在刚开始的时候是免费为游戏开发商工作的。有些人开始的年薪在两万五千美金左右，最多可达到七万五千美金。有些人可能会赚得更多。对于美工的薪金分配，其中包含了许多因素。他们是否参与创作资源和管理工作组？他们是否帮助培训别人？这些“软”因素构成了能被感知的价值的一部分。

这是一个经常有人失业、瞬息万变的世界。在这些受糟糕经济的影响最严重的娱乐公司之外，你能在哪儿找到工作？你或许会找到几次机会，但是很难。这就是为什么我用整章的篇幅来让你保持多样化的技能和思维的原因。

如果你能够在你的热情和对怎样为你的长远目标而努力奋斗的理解之间保持平衡，你就会发现作为一名游戏美工会是非常值得的，而且很有成就感。



11.3 游戏设计

随着游戏范围的不断扩大，就得有人留意游戏设计流程问题和游戏可玩性问题，并且提供设计到制作的信息。对于大多数游戏开发商来说，游戏设计角色是一个有趣的野兽。一般来说，游戏设计人员并不直接被游戏公司所雇用。在多数情况下，你首先在游

戏开发的其他领域工作，之后才会进入游戏设计行列。在某种程度上讲，在别人给你这个赚钱的职位之前，你必须要向别人证明你的设计能力。

关卡设计人员的工作是制作游戏关卡。进入关卡设计领域最好的方法是展示你在学习如何制作你自己的游戏时所构造的关卡。许多开发商会定期地或者偶尔到专为他们的游戏制作关卡而构造的网站上“转转”，从一帮最好的关卡设计天才中寻找最好的。我不是欺骗你；要引起别人的注意并不简单。当你的关卡开始在产业内受到欢迎时，向别人展示自己是引起别人注意的一个很好的方式。进入关卡设计领域的途径非常简单直接——只要制作高水平的关卡！

另一方面，要找到一份游戏设计的工作，你需要展现你以前的设计工作。这可以是完整游戏的形式、设计文档或者设计素材等等。这个领域内存在一些专业性，因为开发商喜欢雇用的是那些对他们准备建构的这种游戏有真正的热情和爱好的设计者。如果你是一位发烧级的体育游戏设计者，你可能不会对设计一个完完全全的角色扮演类游戏有任何的兴趣，反过来也一样。

如果你不能直接进入游戏设计的行列，最好是从游戏开发的另一个领域开始，进而当某个职位出现空缺时再试着转到设计上来。

■ 游戏设计作为一种职业选择

如果你能长期地坚持并且决心已定，而且有能力实现你的追求，那么你在游戏设计领域就会有希望。我的希望是随着游戏产业规模的不断扩大，不久将会出现更多的机会。如果你曾经参与过好几个游戏的设计，展示你的一些工作将是非常简单的事情。然而，得到一份重要的设计任务永远不是一件容易的事儿。竞争总是非常激烈。让我们面对它，对于许多游戏爱好者来说，还有其他更好的选择吗？

再考虑一下其中包含的纯粹的数字吧。目前，任何给定的开发团队，都需要有二十五名游戏开发人员。然而，大多数这些开发职位都是为程序员和美工保留的。这就意味着在游戏制作中，一般都有着比游戏设计人员更多的程序员和美工。所以，如果你竞争这些紧俏的游戏设计职位，竞争就会非常激烈，并且可能降低你的薪水。简单地说，其他人可能只要求比你更低的报酬就愿意干同样的工作。

对于那些想接受挑战的人来说，游戏设计这种工作可能是一种很好的职业。游戏设计会有什么样的挑战呢？这是一种实现关于艺术和技术的交叉创新的内容的挑战。这种努力很少会是简单的或者容易的。

至于游戏设计者们的薪金范围就像许多其他开发职位一样差别很大。初入门者的年薪大概在两万五千美金到三万美金，而那些相对有经验的设计者能够挣到六万美金左右。如果你很精通三维工具、软件包和脚本编写，你就明显地为你的角色增加了价值和灵活性。“超级”设计者显然能挣到比上面提到的多得多的薪金，但是几乎没有开发人员能达到这一水平。



11.4 游戏制作

游戏开发商和发行商的职员中都有游戏制作人。在其他和游戏不相关的软件开发队伍中，这些人被称作项目经理或者软件产品经理。游戏产业使用了更多的面向娱乐的所谓的“制作人”。制作人的工作就是通过任务分配、跟踪、协调、预算和提供便利等手段来管理一个游戏产品的所有方面。他们具有计划制定人员和项目协调人员双重身份。通常情况下，他们有很深厚游戏开发底子，并且对游戏开发很熟悉（原因显而易见）。

按照传统，制作人是质量管理部门走出来的。他们由助理制作人开始，逐渐成为副制作人，进而成为完完全全的制作人。助理制作人常受副制作人监管，而副制作人被制作人监管。

如果你喜欢高度的专业化、面向细节的工作，将在你最不期望它们的时候向你投射曲线球，那你可能会喜欢在游戏制作中工作。你将常常向高级管理层汇报工作的进度，保证游戏项目的速度在日程上、开支在预算中，同时要保持已经和发行商构造起来的稳固关系。这种工作把各种细节和交际工作与对游戏中的技术任务的扎实的理解结合了起来。在正式的游戏设计职位出现以前，许多制作人同时也是游戏设计者，许多程序员和美工同时也是游戏设计者。现在，一些人已经决定留在游戏设计这方面继续发展，而其他的人不再从事面向设计这个工作、而是从事一种纯粹的制作工作。

不同于在其他领域，你可以跟着一些熟练的人员工作、学习；要成为一个出色的游戏制作人，实在没有什么好的学习办法。对于制作工作来说，这里没有真正的标准和尺度，因此一些“可疑的角色”有时会混进制作队伍里来。

你的日常事务可能包括了团队计划、进度跟踪、游戏制作评估、设计协调、艺术评价和培训支持，然后你就是处理那些其他人没有时间去处理的细节方面的工作。这些可能包括市场、PR 连接、特殊事件、食物供应、事件规划、咨询服务以及在取得突破性进展的那天早晨为大伙做早餐。

■ 游戏制作作为一种职业选择

游戏开发商通常通过一家公司培养他们自己的制作人。这并不是说公司不会从他们当前队伍之外雇用制作人——事实上他们确实这样做了。但是，自从制作人最终对关于开发工作室或出版商的商业游戏计划有了初步认识以后，相互信任因素就需要放在一个恰当的位置。对于开发工作室来说，要想从外面找到一个他们满意的制作人是很困难的。另一方面，为了促进内部发展，从来不或者很少雇用外部人员会导致团体的停滞发展，因为开发商仅仅知道以前他们是怎样做的，而并不能看到别人先前犯的是什么错误，并从中学习受益。

游戏制作人的职位也因数目有限而很难获得。即使是在相对较大的、独立的开发工作室，也仅仅有两三个游戏制作人。和众多的程序员和美工的空缺职位相比，可能仅仅有两到三个空缺的全职制作人职位。这就会降低你获得制作人职位的几率。

游戏制作角色是非常令人满意的职业，特别是当你在克服一系列制作上的困难并最终完成一项为大众提供娱乐的游戏后。按照经验的不同，游戏制作人的年薪范围从两万八千美金到九万美金不等。另外，薪金范围通常都要考虑工作的方方面面。一些开发商喜欢把他们的制作人几乎完全安排在商务和工作室发展的事务中，而其他的开发商则喜欢让他们的制作人拥有更多的技术方面的倾向。好多成功的制作人看起来是两方面兼而有之。他们把对游戏结构技术的深刻理解和一定程度的商业感觉和经验结合在一起。



11.5 音效

音效是整个游戏中传达感情的重要组成部分。如果你不相信我，你只需要试着关掉音乐去玩一下你最喜欢的游戏，感受会大不一样。现在的游戏有很严格的音效要求，还要支持像杜比环绕音响 (Dolby Surround sound) 那样的扩展音效格式。游戏要求创作大量的和众多特殊效果配合的音乐或福氏音效 (Foley audio)。一些游戏开发人员同时也是音乐家，这些人通过为游戏制作音效而看到了他们自己有兴趣的理想结合物。

学习怎样为游戏准备音效，以及在技术局限和内存限制的情况下进行开发，就是一些很大的挑战。许多音效工程师和音乐作曲家都是在游戏之外的音乐领域工作。

■ 音效创作人和工程师作为一种职业选择

对大多数开发商来说，音效部门都是非常“小巧的”。一些开发商选择跟音效专家签订有关他们的音效需求合同，这些音效专家知道怎样为各种游戏平台配音乐。如果他们和开发人员一起在室内工作，那么这个音效部门肯定会很小。这就意味着每个公司仅雇用少数几个人在音效部门中全职创作音效。这也意味着对于这少数几个职位的竞争是非常激烈的。

对于那些热爱音乐创作和声音效果，并且逐渐喜欢上游戏的人来说，这可能是一份理想的工作。时刻牢记，你得常常整理很多能够满足听觉要求的每一个声音版本。你也要针对一个音乐主题或运动创作不同的版本，并把它们集成到游戏中去。

游戏开发中的全职音效工程师收入大概是三万美金到六万美金或者更多。作为一个独立的承包人，你有时可以通过从事游戏音效以及其他内容的工作从而赚得更多。总之，只要你能够在具有挑战性的预算、时间期限和技术限制下为游戏开发出限定的音效内容，为游戏制作音效是有价值的和富有挑战性的。



11.6 质量保证

许多开发商视质量保证部门为他们事业的跳板。确实如此。当你知道有多少现在的工业界权威在某个时候曾经在质量保证部门工作过，你一定会感到吃惊的。我认为质量保证部门对于其他娱乐公司是邮箱，而对于游戏来说它不再是邮箱了。

质量保证部门能够让开发商明白：一个游戏是如何最终获得成功的，以及各种不同种类的决策是如何帮助成型一个游戏的。如果你在质量保证部门工作，并且参与过游戏制作的一些过程，你就会明白。一些开发人员成为了测试员，但是他们真正想成为的是美工、程序员、设计人员或者音效工程师。这是很好的。事实上，如果你的职业生涯是从质量保证开始的，那相对于从事其他任何工作，你会有更多的跳槽机会。这些工作有助于你确定你的目标。

还要再说一次，如果你无法得到其他方面的工作，你可以先进入测试部。我就是这样做的，并且其他很多人也是这样做的。

一些人选择一直待在质量保证部门，那样也是很好的。如果你成为质量保证部门的经理并且帮助经营整个测试部门，你就可以在游戏的制作期间休息，只有在测试这些游

戏的时候你才需要工作。

■ 质量保证作为一种职业选择

许多开发人员只想在质量保证部门“混上”两年然后改行，而别的人却选择它作为自己的职业。在质量保证部门中有全职和兼职两种，而实际上其他任何工作都只有全职的。质量保证测试员可以按小时付钱，每小时 8 美元到 13 美元，兼职没有奖金。作为一个开发新手，测试是很好的进入游戏产业的途径。全职的质量保证工作人员带奖金的年薪在两万五千美金之内，或者根据他们的职责和经验还要多一些。拥有全额的薪水和奖金的全职测试员是很少的，他们一般都是高级的测试员。质量保证部门经常依赖大量的兼职者的帮助。在最忙的时候，他们可以找很多兼职者来帮忙，在一个大的游戏做完后再让这些兼职者离开。质量保证经理能够挣到五万美金或者更多，但是他们的作用扩展到了包含支持整个测试工作的许多责任。



11.7 创办独立的游戏工作室

对于许多新的老的开发人员，构造自己的开发室是一直以来的梦想。联合一帮乐于奉献的、有才能的朋友来一起开发游戏，这表面上看像是一个绝好的想法——甚至有大量产业经验的人而言。它可能确实是个好的想法（而且这个“独自的”想法会因许多情有可原的因素而发生变化），但是作为一个独立的游戏工作室，其成功的路上布满荆棘。为了开发最前沿的游戏而创办独立的工作室的年代已经被淹没在历史的长河之中。甚至那些曾经在先前控制台上开发过很长时间的开发者们也发现，要跨越到未来的游戏开发上是一件有挑战性的工作。

发行商只愿意和拥有深厚的商业经验的开发者签订游戏开发协议。与产业内权威人士和代表人物构造良好的关系、向别人展示你自己拥有的全部游戏技术，并且拥有一个有着丰富经验的一流开发团队，在这样的基础上，也才可能得到一个报酬很少的低优先级的游戏开发项目，但是即使这样也变得越来越难以保障了。

作为缺乏经验的初级开发商，走到发行商面前然后说：“喂，我们想为你开发一个游戏！”你不会得到什么结果。事实上，这种情节甚至不可能发生。这对我们游戏爱好者来说是残酷的，但是发行商想在知名的开发商中寻求合作。在刚开始的时候，他们已经见过太多和他们差不多的不知名的开发商。简而言之，他们只想把风险投给知名的

开发商。现在，随着游戏的发展，即使是知名的开发商也很快会变成了不知名的开发商，但是前边的开发工作留下来的“遗产”使得你更容易拿到合同。他们想“知道”并且“相信”你能为他们制作出一个成功的游戏。他们怎样去逐渐“知道”或“相信”你的这种能力呢？可能就是因为你已经为别人制作过一个成功的游戏。必须要有人在某个时候给你一次机会，否则是没人愿意给你这第一次制作成功游戏机会的，从而又有机会再制作下一个，是吗？就是这么回事儿！

如果你考虑过，最近时兴起来的独立游戏工作室实际上是从大的发行商那里派生出来的。这就是说一个团队在发行商的支持下走到了一起，学习怎样在一起协调工作、完成一个或两个卖得火爆的游戏，然后他们就可以独立出来，从而可以不仅仅为一个发行商（这个发行商就是他们最初形成的那个）开发游戏。

构造独立工作室另一个常见的方法是：找一些以前曾经为数家开发商或者开发商 / 发行商制作过游戏的开发人员，把他们集合在一起快速地组建一个新的开发团队。通常他们会为获得工作而运用他们与以前发行商雇主的关系。这也能行得通。

这些告诉我们，如果你想创办一个独立的工作室，在尝试以前你可能会先进入产业内并且获得一些丰富的经验。你也可以选择直接开始，去争取一些创业资金和风险投资（如果你运气不错的话），但是在现在的商业环境下，这将是一个非同寻常的困难之路。

这意味着你创办独立的工作室来开发游戏的梦想破灭了？不对，有许多方法让你走向自己的目标。你可以在每年的游戏开发者大会（GDC）上参加独立游戏节；你可以从事共享软件的开发工作。嘿，能为 *id Software*（著名游戏厂商）工作也很不错。在不久的将来一定会出现其他的交付系统，通过这种系统你可以直接交付到你的潜在玩家手中。你不要因为风险承受度以及对大规模的游戏发行商的忧虑而对此失去信心。

让我们弄清楚游戏开发商和游戏发行商的区别。游戏开发商构造游戏内容并且雇用程序员、美工、设计人员、制作人员以及音效师。相比较而言，游戏发行商负责包装、制造、市场、促进以及支持游戏的销售。制作游戏的开发合同是游戏发行商给开发商的。

■ 参与到游戏开发中

得到有价值的开发经验的最佳方法是成为一个游戏开发人员。通过参与游戏开发工作，学习游戏开发的规则以及游戏产业的微妙之处（在试着面对独立开发的茫然以前），大多数的程序员、美工、设计者、音效工程师、制作人以及质量保证人员都一下子在他们的职业生涯中迈出了一大步。

要去理解整个游戏开发周期的运作以及在日常的游戏工作环境中发行商怎样与开发商合作进行游戏制作的,这个很有必要。通过作为一名开发者来锤炼自己,你将学会解决许多你可能在开发游戏中遇到的关键问题。

■ 产业经济

如果你观看近一段历史时期的游戏产业,你有可能看到:几年的时间当中,游戏产业中有很多空缺的职位,而也就是近一两年,在游戏产业中找工作变得非常困难。游戏产业,就像其他许多产业一样,在整个经济环境下随着经济的起伏作出反应。

在某种程度上讲,游戏产业几乎不会受到外界经济状况的影响,但是有一些时候产业内会明确表示不招聘人。如果能预料到这些上升和下跌的势头以便于你能有所准备,那是最好不过了。而且,这是保持多变的心态的一部分。



11.8 游戏工作室的分工

接下来的部分细分了通常意义上的专业角色,并简要描述了各个职位的分工。记住,一个独立的游戏开发商制作游戏的内容,而一个游戏发行者包装、市场化,支持并销售游戏。

开发商和发行商也可以合在一起,事实上,游戏发行商下设有自己的开发商,致力于制作自己的游戏。然而,可能最普遍的安排是发行商把他的游戏承包给一个开发商开发。

让我们看看开发商/发行商混和型公司的角色分类。图 11-1 展示了一个混和型公司通常的组织方式。在不同的公司,这些角色的搭配具有极大的灵活性。

■ 行政部门

总经理/创始人也许是一个具有纯商业背景的管理人或者是一个制作游戏的程序员。他们的管理能力存在着极大的差异。作为首席执行官,这个人向公司的所有人或者股东负责公司的财政运转和投资方向。

产品开发副经理负责指导游戏产品开发的所有方面。这些人通常在市场团队的旁边运作以搜寻并获取有吸引力的游戏制作授权。他们也与开发商商讨开发协议,并且全权负责有利可图的游戏的成功交付。总的来说,他们一般都具有制作和商业的背景。

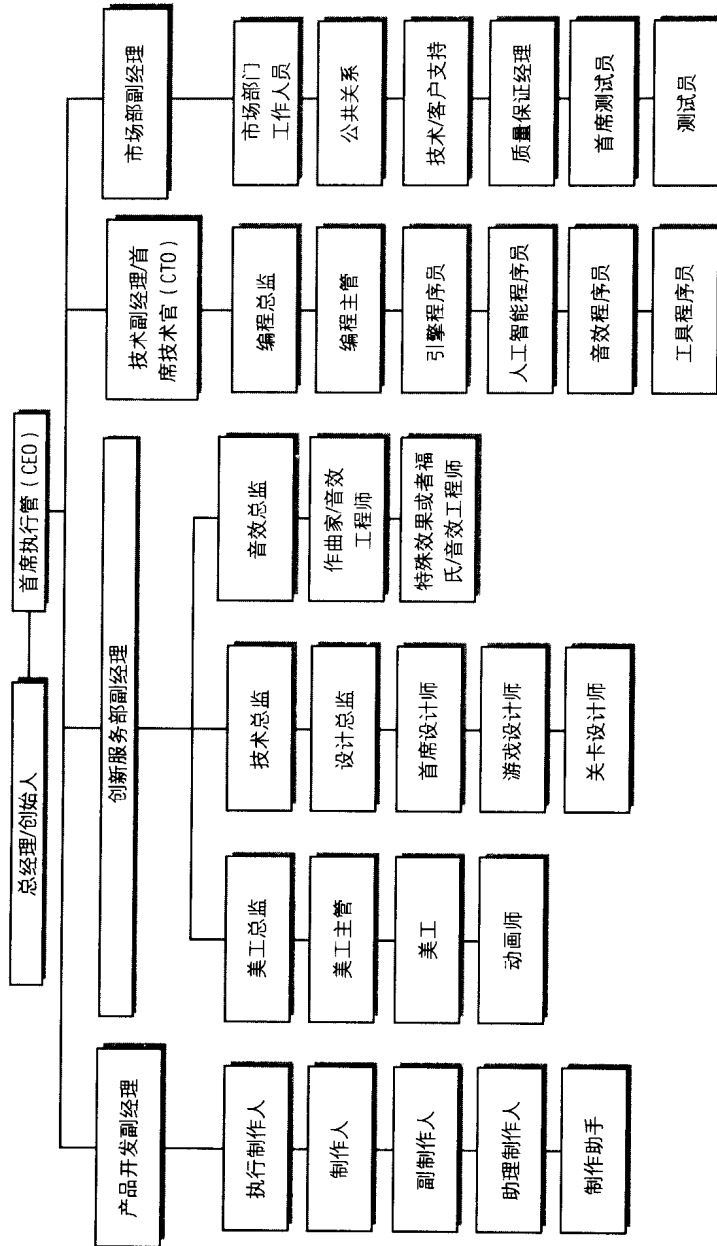


图 11-1 开发商 / 发行商混和型公司的组织方式

创新部副经理是所有项目的有创新领导。这些人通常具有曾经帮助指导过一系列各种类型带有创新成果的游戏项目，这意味着他们指导的创造材料已经受到市场的好评。不是每一个组织机构都设有创新部副经理。很多公司由游戏设计主管来运作这个职位或者相似职位的工作。

技术部副经理或者首席技术官 (CTO) 负责考虑和指导公司里的所有技术问题。通常情况下，在这个岗位上的每个人都在开发、制作和维护数种可持续发展的技术方面具有很丰富的经验。他们也具有管理和团队组建的经验。他们帮助制定影响了公司的技术方向的指示。我们应该在这个平台还是那个平台上进行开发？我们应发展什么样的内部技术？这些都包括在他们可以帮助回答的问题之中。他们也许还领导自主游戏引擎的开发计划、专利技术等等。任何有关公司的核心技术的决策都经过首席技术官的深思熟虑。

市场部副经理主要负责把已完成的游戏推销带给游戏玩家。他们的工作方式有很多种，而且他们还和很多其他组织有合作，有助于确信影响游戏销售的几个因素：游戏受到关注程度、宣传和适当的广告。很显然，如果你有很多游戏都没有很好的销量，你也就不会再开发游戏了。

■ 产品开发部

执行制作人是公司里的制作人领导，负责其他制作人和他们团队的工作。他们与产品开发副经理联合起来，帮助权衡某一授权游戏的价值以及与外来开发商确定合同时协助作出决策。他们还能够参与监督所有现场的游戏制作工作。

开发商帮助管理整个游戏团队开发的所有方面，并负责确保一个或者多个游戏项目能够按时完成而且没有超出预算。他们尽力确保他们组所有的开发人员都有完成他们的开发目标所必需的工具和资源，最终制作出一个成功的游戏。对他们来说，团队组建、任务分配、进度跟踪、做预算以及游戏产品视觉和清晰的维护都是很重要的工作内容。

副制作人通常在一个或者多个游戏项目上和制作人一块儿工作。一些副制作人甚至全权负责较小的项目，比如说便携平台上的游戏开发，而不是大控制台上的游戏开发。公司喜欢通过逐渐增加副制作人负责项目的范围来培养他们成为制作人。优秀的制作人对副制作人传授他们自己因犯错（或观察到别人犯错）而积累起来的所有来之不易的教训，以此来引导副制作人。

助理制作人通常是新近从质量保证部门提升上来的制作人。他们这些开发者才刚刚开始学习一线的游戏开发过程知识。他们通常与副制作人和制作人一块儿工作，帮助完

成一大堆分派给他的细节工作。

制作助手是没有工资或按小时付报酬的实习生，他们对学习更多的关于游戏开发的过程有着浓厚的兴趣。他们帮助完成次要的项目细节，比如为美工收集参考书或者是项目中的跑腿人员（“去拿这个或者取那个……”）。

■ 创新部

所有的开发人员都具有创造才能，但是一般只有美工、设计和音效才被划分到这个范围或类别。就像其他所有提到过的类别一样，虽然大多数开发人员都要求在极少的技术支持下完成分配到的任务，这个类别仍然分了高级、初级和一些入门级几种职位。在紧张的日期限制下，几乎没有时间把一个人技术上的缺陷降低到最低。如果你想提高你的技术那是可以的；你只需要停止工作，待在幕后，直到你的技术发展到你把你推向最前沿的工作。这是一个商业现实。

美工总监与副经理或者创新部总监互相配合，确保游戏做到高的视觉标准，以及制作出的材料能够让知识产权拥有者的授权人满意。大的公司往往有许多美工总监，同时为多个游戏工作。一般地，美工总监负责鉴别美工所做工作的效果，以及为他们的美工参与组织一些类型的培训课程，以提高美工们的技术竞争力。

正如我们以前多次讨论过的那样，美工们在构造游戏资源方面的贡献是很大的。一些美工仅致力于模型构造、材质制作、概念绘图或者动画制作，其他美工则对所有的这些都涉及一点儿（或者一大堆包含所有内容的东西）。美工主管一般指高级美工，在游戏开发中他不仅负责指导美术设计（比如构造模型和制作材质），而且还负责管理方面的工作，也就是帮助那些缺乏经验的美工制作富有内涵、游戏需要的美工资源。在最好的情况下，美工主管是团队中其他美工的参考对象和有价值的资源。

美工要制作低精度多边形的模型、材质、界面要素、情节串联图板、电影片断等等。

动画师主要是定义物体和角色的动作。动画制作者可以处理任何需要复杂类型动作的游戏要素。一些动画师也是模型制作者和材质美工。但是，随着数字动画制作技术变得越来越复杂，动画师就只能集中精力于动画制作了。事实上，在我们讨论过的许多美工角色中，专业化程度正在逐步提高。

技术总监的工作是为动画师构造制作动画构造模型。它包括用逆运动学（*Inverse Kinematics, IK*）和正运动学（*Forward Kinematics, FK*）装配角色，用于动画制作中的控制处理。还有，根据需要用到的动画技术，技术总监可能会构造基于拖动的动画系统，

这样动画师就可以控制模型的顶点集合。

设计总监在所有员工中属于高级游戏设计师。许多公司没有专门的设计总监，但是当较大的设计团队组合在一起制作游戏时，这个角色就开始变得很普遍了。

总设计师是设计细节管理者，他主要关心的是为玩家设计一个尽可能好玩儿的游戏。所有与设计有关的事，故障、问题、挑战以及工作环境都是由总设计师和他手下的团队成员共同负责。

游戏设计师的工作是充实或定义所有与给定游戏有关的游戏设计和游戏操作细节。每一个独立的要素将怎样呈现在游戏中以及如何以极完美的细节与另一个要素联系？这些细节的结果玩起来有趣吗？如果不是，就可能需要加以调整甚至要重新制作。

关卡设计师把不同的游戏资源放到关卡中，他们一般都使用专门为关卡装配制作的工具。关卡设计师关心游戏的整体结构，但是主要还是处理从关卡与关卡之间的游戏细节和结构。

音效总监负责设计所有支持游戏的音效内容的创作。他们帮忙管理由音频工程师、音效专家和作曲家组成的团队。作为他们管理内容的一部分，他们常常帮助制作人和团队安排录音时间、为游戏配音，等等其他的工作。

作曲家和音效工程师创作为游戏伴奏的配乐曲目。他们为序幕电影、过渡电影片断、结尾电影创作音乐。只要游戏需要，作曲家可以创作任何原创的、定制的音乐。许多公司与独立的作曲家签订合同，给他们的游戏创作音乐。

特殊音效或 Foley 工程师的工作是为游戏创造声音效果。所有的这些奇异的敲击声、咀嚼声、踢腿声以及迫击炮声都是这些天才们的杰作。

■ 程序设计或技术部

编程总监通常是高级软件工程师，负责几方面的游戏开发的代码支持工作。他们可能领导游戏引擎开发任务，管理有限的研究内容，为代码共享和跨平台开发问题提出建议或做出改进，或者帮助保持整个公司内代码书写的指导方针和代码资源最新。

程序主管在团队中通常是高级程序员。他在整个项目的中以团队为单位指挥编程工作。大多数基于项目的代码细节则由程序主管以某种重要的风格来实现。他们还帮忙评价、辅导以及培训新来的程序员。

引擎程序员主要完成游戏引擎的开发、修改以及改进。他们可能不参与具体的游戏项目，而是致力于代码库的编写，这样，其他游戏内容就可以在此基础上进行开发。要

牢记，许多在某种程度上为公司的游戏引擎编程任务做出贡献的程序员（如程序主管、编程团队、编程总监等）并非都是引擎程序员。作为一名游戏程序员，你可能在做许多其他类型（音效、界面、工具）的编程工作，也可能在做引擎代码开发方面的工作。

人工智能（AI）程序员集中精力于与人工智能有关的编程问题上，比如路径搜索、智能演员或者代理系统，以及构造复杂的体育运动游戏或者头目行为。

音效程序员的工作是编写程序来支持游戏中的音效要求。他可以编制智能音效脱机系统（*intelligent audio spooling systems*）用以播放音乐片断作为游戏的背景音乐，改写固有音效在储存器中存放的方式，或者为处理输出专业的音效输出格式开发自定义的代码。

我最喜欢的职位是工具程序员。工具程序员的工作是制作支持所有类型的复杂游戏创作需求的工具。这些工具可以用来支持音效处理、关卡制作、演员属性系统或者在大类型多玩家在线游戏中支持玩家组织比赛。

■ 市场部

市场销售人员由许多人组成，这些人在每一个与把游戏推向市场有关的领域内都有专门的联系。对于每一件可能有助于构造和通知游戏玩家关于每一个新发布的游戏信息的事情，他们都将全力以赴。

公共关系工作人员想尽办法帮助在作为消费者的玩家和游戏公司之间架起一座沟通的桥梁。他们组织专门的游戏竞赛、试着去解决玩家的一些疑问，并且使开发人员了解公共意见和对某些问题的反应。

从一个游戏发布到后边长期很长的一段时间，技术 / 客户支持员工的工作就是为玩家提供支持。他们提供在线帮助或者建议热线、网站，常见问题解答（FAQs）技术支持等等。他们回答问题并帮助玩家解决安装、加装补丁程序，或者兼容性问题。如果没有他们的工作，那么就会丧失一个向客户传达有趣的玩游戏体验的重要途径。如果玩家得不到某个游戏的支持或帮助，他们下次在某个地方看到你的标志时就会想起前边的经历。

质量保证（QA）经理管理各种测试服务和资源，以确保最大程度地把无错误游戏体验交付给玩家们。

测试主管就是一个关于任何特殊的游戏题目测试组中的联络人。他们从测试团队的反馈信息中剔除掉大部分信息，这样开发人员就能够对所有明显的错误有一个清晰的概念。

测试员的工作就是发现游戏错误，并“中断”游戏，将错误报告给游戏开发团队并由他们来修改。

■ 分工总结

希望你已经发现角色分工对开发商 / 出版商混合型的公司很有用。最好牢记许多角色在不同的场合代表不同的事情。角色细节的好多方面都是由公司规模、资源获取和许多其他因素所决定的。大多数这些角色在整个游戏产业中很通用，因此在这里提一下是值得的。在你追求你自己的目标时，你起码要对在游戏产业中什么样的职位适合你有一个认识。



11.9 关于测试途径案例研究评论

对我们大部分人来说，记住质量保证是通向游戏开发的途径这一点是很重要的。你在测试部的第一个星期对你来说可能是一个震惊，那就刚刚好。它会让每个人感到震惊。你不会相信你就是通过玩新游戏来获得报酬，而且你也不会相信这也会使你的脑袋紧张、脖子酸痛和眼睛疲劳。一定要注意休息。

你将可能会遇到和你一起在测试部工作的各种各样有趣的人。我遇到过有商业学位的人，有法律学位的人，儿童演员，摔跤狂热者，厨师，神经系统科学家，职业赌徒和因杀人而正在接受审判的家伙。每个人都分享着对游戏的热爱（如果不爱它们，做这件事会使你精神错乱），而且大多数人都在寻找成为游戏开发工作室专职人员的途径。你甚至可能遇到一些“职业的”或追求竞争奖金的游戏玩家。这些游戏玩家中的一些已经赢得了由游戏发行商组织的游戏比赛，而且他们已经跻身于他们所在领域中最好的玩家。如果你没有达到这样的水平，别担心。测试部门所需要的游戏玩家范围很广，从总体职业选手到近乎新手都要有。如果你能设法在某一点上压倒一个游戏冠军（或者是相关的任何一件事），那在测试部就会给你发一枚荣誉徽章。

你得要玩各种各样的游戏。你可能立刻就喜欢上了它们中的一些，但却讨厌其余的。当你一整天都做同一件事情时，很容易你就会厌倦它。你将开始学会哪些是真正的程序错误而哪些不是。但愿你能把书写完整的、描述性的程序错误报告学到家。甚至在这里，你也想要发展你的交际能力。

在那里你必须夜以继日地睁大眼睛和竖起耳朵。像一块“戴着隐形眼睛”的大海绵

一样,吸收你所能吸收的一切。要问大量的问题,但不要问得太多。你想要搜集信息和表达你的兴趣而不至于使人神经错乱。毫无疑问,你会交到一些好朋友。如果你喜欢质量保证过程,那你可能已经找到了你的位置。如果你想进入游戏开发,那就开始留意眼前的机会。这可能有难度,而且很可能不会一夜间就冒出这样的机会。和任何其他职业一样,通过制作游戏来谋生,你需要作出好多种牺牲。学你能学会的,还要自娱自乐。记住:开发商不是要寻找能很快掌握知识的人;他们要寻找的是真正喜欢制作游戏的人。

采访你自己



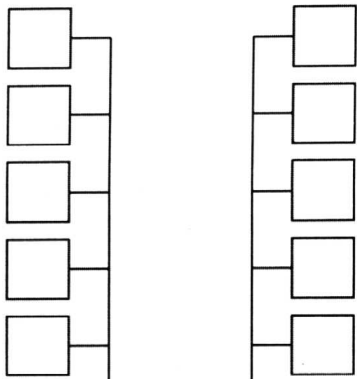
在前面的章节中我们已经听了很多有才能的游戏开发人员介绍构造游戏的一般过程(再一次感谢他们),但是不同的是,这一章讲的是职业,讲的是你自己。现在大概该问一下自己一些关于游戏产业的问题和在游戏产业中你想在哪个方面发展。基于此,我总结了一些问题,用以测试你自己你在游戏产业中想要追求的目标。

- 为了实现我在游戏产业中的目标,我需要哪些基础知识?
- 游戏开发的哪个方面最能引起我的注意?
- 除了游戏外,我还想做别的事情吗?
- 我可以灵活地把握可能降临到我头上的机会吗?
- 在我头脑里有多种多样的想法吗?
- 为要获得一些相关经验,现在我应该做些什么?
- 我怎样才能贯彻在这里提到的好多关于游戏设计的重要主题?
- 哪儿需要你,你就愿意去哪儿吗?
- 我对游戏的热情能帮助我度过困难时期吗?
- 要成为我所在领域或者科目中最好的,我能做些什么?

希望你能充分利用这里提供的信息来帮助你在你自己的方向上制作出令人兴奋的新游戏。感谢你和我们一起走完这个旅程。祝你所有的游戏梦都能梦想成真!

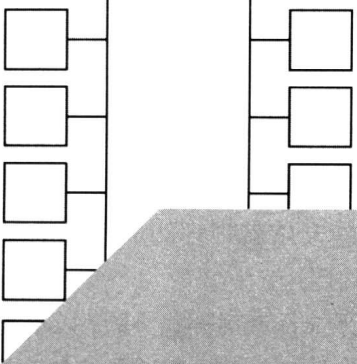
MEGA 提示

1. 质量保证通常都是你进入游戏开发的途径；
2. 现在就开始为你的游戏开发兴趣打基础。不要忘记，你游戏兴趣会随着时间的流逝而改变。你可以不停地调整你为你的基础设置的目标；
3. 问问你自己你制作的游戏关卡的强度这个难以解决的问题；
4. 充分利用这里提供的材料，把它们作为跳板，从而设计你的路线；
5. 理解各种开发角色，并且让直觉和兴趣引导你找到那些可遇不可求机会。

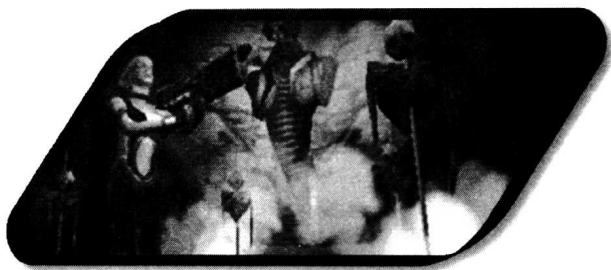


APPENDIX

A



参考信息



附录 A 涉及了多个种类的有价值的参考信息，方便研究游戏产业。但是，它决不是探索游戏产业的所有可能发展方向的完整列表。在这个动态领域里每天都会有新的机遇出现。



A.1 教育

许多学校、学院和大学已经开始向学生提供与游戏设计相关的课程了。在多数情况下，学校提供的这些相对较新的课程只能教一些基础的内容。任何东西都无法替代在开发团队中的游戏构造过程中获取的经验（即使你开始就把你的朋友聚集起来集思广益！），不过，你还是应该把握住所有可能的教育机会。

如果可能的话，我鼓励你试着向一位游戏开发人员学习一些经验，以平衡你所接受的游戏教育。记住，在质量保证部门接收锻炼仍然是迈入游戏领域最好的途径之一。对于业余时间的教育机会，你可以经常和当地的大学校园的“教育推广”部门核查落实，或者也可以在网上搜索。

要牢记，面向游戏编程的普通教育是计算机专业或者主攻三维图形编程的数学的工学本科学位，而面向美工和设计的是传统艺术和 / 或设计文学学士学位或者美术学生学位（多媒体设计、工业设计，等等）。

我的建议是要你完成所选择的全部的文学学士或者工学学士学位教育。这会帮助你构造起你的游戏开发基础，而且会证明你可以开始做工程，去面对各种形式的挑战，甚至更重要的是完成一项工程。很多成功的游戏设计人员都有着海洋生物学、人类学、航空航天工程以及剧院布景设计等等的背景。

开始出现了一些在线教育的游戏培训班和机会。下面是一些专门的游戏设计培训供你选择：

- **DigiPen 技术学院 (www.digipen.edu)**: 最早提供真正的游戏开发学位课程的学院之一。
- **ed2go (www.ed2go.com)**: 我定期在这里给学生讲授在线游戏产业课程, 包括几门儿专门面向开发中的游戏设计。
- **Full Sail (www.fullsail.com)**: Full Sail 提供有面向贸易的游戏开发课程。
- **国际美术学院 (Art Institutes International, www.ait.edu)**: 在美国, 大多数的美术院校现在都提供有游戏开发课程。

这里需要强调的是, 作为一个研究和探索课题, 游戏设计已经成为很多有名望的大学的传统课程的一部分, 这些大学包括麻省理工学院 (Massachusetts Institute of Technology, MIT)、罗切斯特理工学院 (Rochester Institute of Technology, RIT)、卡耐基-梅隆大学 (Carnegie-Mellon University) 以及乔治理工学院 (Georgia Institute of Technology, Georgia Tech) 等等。



A.2 活动

电子游戏产业中每年都有几个主要的产业盛会, 下边列出其中最大的盛会:

- **娱乐游戏产品展览会, 也称为 E3 (Electronic Entertainment Exposition, www.e3expo.com)**: 由美国主办的主要的年度产业贸易展览会。
- **游戏开发者大会 (The Game Developers Conference, www.gdconf.com)**: 游戏开发者大会上的独立游戏节 (The Independent Games Festival) 是展现你独立开发的游戏作品、并让全世界的开发商和发布商都能够看到它的绝好方法。
- **SIGGRAPH (www.siggraph.org)**: 最大的计算机图形和图形技术公司年度集会。
- **电子消费者贸易展览 (EElectronic Consumer Trade Show, www.ects.com)**: 欧洲最早的交互式产业贸易展览。
- **MILIA (www.milia.com)**: 在法国康城举行的一个国际游戏大型展览会。



A.3 产业杂志

《游戏开发者》 (*Game Developer*, www.gdmag.com) 杂志是一份必读的材料。我建议你认真地读读每一期杂志中的 “Post-mortems” 栏目。杂志中还有大量与设计相关

的文章。每年一次的“薪资调查 (salary survey)” 栏目向你提供了薪金范围信息。



A.4 与游戏产业相关的网站

网上突然出现了众多与游戏产业相关的网站，其中有几个网站我经常访问。下面是我向你推荐的几个：

- Gamasutra (www.gamasutra.com);
- Blues News (www.bluesnews.com);
- Game Daily (www.gamedaily.com)。



A.5 组织

要会见其他的游戏开发人员、那些对学习更多有关游戏开发的人，最好的方法之一就是参加当地国际游戏开发者组织分会的会议。在它的网站上列出了遍布世界各地的分会。如果你所在的地方没有分会，而当地又有足够多感兴趣的人，通常你可以着手筹办自己的分会。

- 国际游戏开发者组织 (International Game Developers Association, www.igda.org);
- 美国互动艺术与科学学会 (Academy of Interactive Arts & Sciences, www.interactive.org);
- 美国互动数字软件协会 (Interactive Digital Software Association, www.idsa.com)。



A.6 跨入游戏产业！

当你“走出校门”，如果鉴于你的代表作或者你的对打游戏演示 / 代码而没有被雇用，那你可以把质量保证部门作为你职业生涯的跳板。你可以经常订购一些《游戏开发者》杂志，它上面包含许多有关进入游戏产业的文章和故事。

GameDev.net 有一些在线文章，向读者提供了各种各样跨入游戏产业的建议：www.gamedev.net/reference/list.asp?categoryid=68。



A.7 代理商和招聘人员

如果你想要进入游戏产业,在找代理商和招聘人员商量之前你必须获取一些工作经验。如果你已经有了产业工作经验, *Studio Search* (www.studio-search.com) 是很不错的代理商。在 www.gamasutra.com 网站的 *Companies/Business/Legal* 路径下列出了许多其他的代理商供你参考。



A.8 工作网站

你可以使用重要的工作公告板寻找与游戏相关的工作,甚至入门级的工作。我以前偶尔看到过招聘测试员和质量保证支持人员的列表,这两种工作都是进入游戏产业很好的途径。

- *Game jobs* (www.gamejobs.com);
- *Monster* (www.monster.com);
- *Gamasutra* (www.gamasutra.com)。



A.9 设计文档参考

请牢记,在开始准备游戏设计文档时,每一个开发商都有他自己的风格和编排格式的偏好。不过, *GameDev.net* 网站 (www.gamedev.net) 是一个开始看设计文档的好地方。在这个网站的“*Articles and Resources* (文章及资源)”下,你能找到数个游戏设计文档的模板。这些模板向你提供了文档布局和内容的建议,你可以用来书写和编排你自己的游戏设计文档。



A.10 TV 编程

现在,很多地区的有限电视都能收看到 *G4TV* (www.g4tv.com),它是一个每天二十四小时开放的游戏频道,每天不间断地播出游戏相关的程序设计。



A.11 《儿童网络隐私保护法》方针

如果你对进一步学习如何制作符合《儿童网络隐私保护法》(Children's Online Privacy Protection Act, COPPA)的在线游戏感兴趣,请访问 www.ftc.gov/bcp/online/pubs/buspubs/coppa.htm。



A.12 自己发行

对于那些有兴趣试着自己发行已经完成的、且具备市场竞争力的商业游戏的朋友们,在 www.gamasutra.com 网站的 Companies/Business/Legal 路径下列出几家公司,它们可以为你提供发行支持方面的服务,比如 *Tri-Synergy* 公司 (www.trisynergy.com)。



A.13 外包测试服务

如果你需要外包或者“户外”测试服务,查看一下美国国家技术系统 (National Technical Systems, www.xxcal.com)。



A.14 游戏产业的市场研究和报告

时刻了解游戏产业的发展趋势和销售模式是很有好处的。有数家公司提供这方面的报告,比如:

- DFC Intelligence (www.dgcint.com);
- NPD Group (www.npdfunworld.com)。



A.15 推荐网站

对于任何有兴趣研究什么样的概念参考资料能够帮助人们了解并会影响他们的早期游戏设计过程的人来说,我下面推荐的每一个网站,都可以作为你的参考点和灵感来源。

- Bay Raitt (<http://cube.phlatt.net/home/spiraloid/>): 建模大师 Bay Raitt (一定要看看 Bay 在《指环王》——*Lord of the Rings* 电影中的作品) 在这个网站中提供了一些关于角色构造、建模和雕刻技术的应用等很棒的忠告;
- Doug Chiang (www.dougchiang.com): Doug 是《星球大战》——*Star Wars* 前传的设计总监, 一定不要错过他在《机器人》——*Robota* 中的精美作品;
- Syd Mead (www.sydmead.com): 作为《银翼杀手》——*Blade Runner*、《异形》——*Aliens*、《电子争霸战》——*TRON* 等的传奇设计者, Syd 的作品总是让人感到兴奋和刺激;
- Craig Mullins (www.goodbrush.com): Craig Mullins (美国著名职业画家) 的令人瞠目结舌的概念画;
- Gnomon (www.gnomon3d.com): 学习操作各种玛雅技术的很好的选择;
- Eni Oken (www.oken3d.com): 对于那些有兴趣研究材质构造的人来说, 这是一个绝好的网站。



A.16 推荐读物

绝大多数游戏开发人员都在书架上摆满了许多方面的范围极广的参考资料。在这里, 我列出一些读物, 我想你会觉得它们在不断延伸你对游戏产业和游戏设计基础的研究方面非常有用。

- A Brush With Disney: Herbert Dickens Ryman 通过谈话和作品讲述了一位艺术家的旅程。由 Gordon 和 Mumford 编辑, Camphor Tree 出版发行。
- *The Art of Maya* (2003)。Alias Wavefront Education 编著 (中译本为《Maya 的艺术》, 由中国青年出版社出版)。
- Ching, Francis (1997)。 *Design Drawing*。 John Wiley & Sons 出版社。
- Hanks & Belliston (1992)。 *Rapid Viz*。 Crisp 出版发行公司。
- Hogarth, Burne (1996)。 *Dynamic Figure Drawing*。 Watson-Guptill 出版发行。
- Mencher, Marc (2002)。 *Get in the Game! Careers in the Game Industry*。 New Riders 出版发行。
- *Pieter Saenredam: The Utrecht Work*。 Liesbeth M. Helmus 编辑 (2002), Getty 出版发行。

- Roberts, J.M. *The Illustrated History of the World* 一共十卷。牛津大学出版社。
- Thomas, Frank 等 (1995)。 *The Illusion of Life: Disney Animation*。Hyperion 出版社。
- Thompson, Frank (2002)。 *Tim Burton's Nightmare Before Christmas* (蒂姆伯顿的圣诞城之夜)。Hyperion 出版社。
- *Visions of Light: The Art of Cinematography* (光影的魅力: 电影摄影艺术)。DVD。Image Entertainment 公司发行。

我推荐你阅读任何关于游戏作品和参考绘画的视觉概念的书籍,比如那些专门描写《星球大战》、《指环王》和《人猿星球》——*Planet of the Apes*的书。



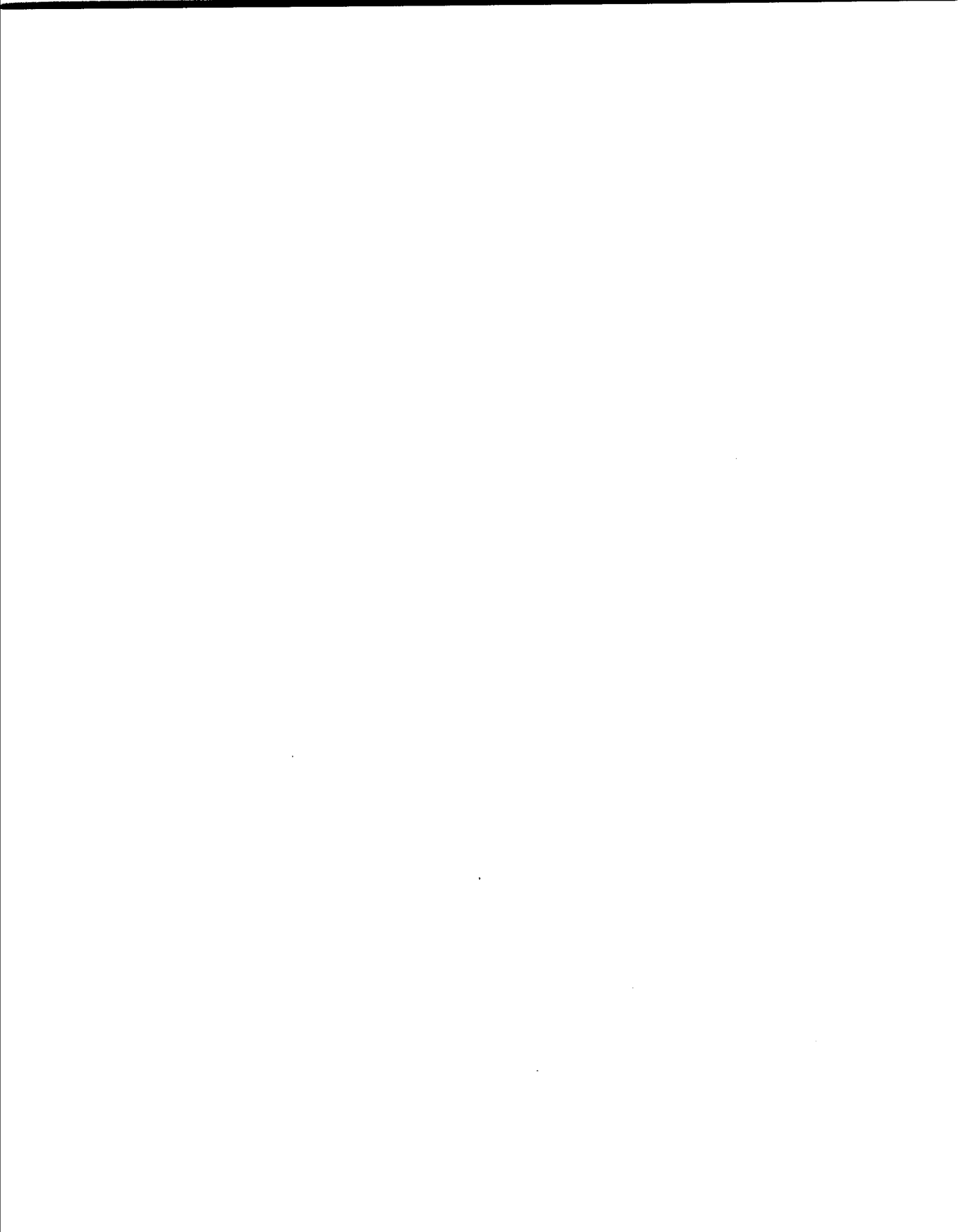
A.17 进一步研究和推荐的推荐主题

下面的列表包含了许多与游戏设计相关的主题,供你做进一步的研究。这里的每一个领域,包括除此之外的其他一些领域,都会对有关游戏设计的多个方面产生正面的影响。在帮助规划新游戏的制作时,游戏设计人员常常不得不做大量不同方面的研究工作。

下面是一些你可以开始研究的方面:

- 光照 (戏剧、电影、电视);
- 写作 (描述、戏剧、技术、滑稽);
- 世界历史;
- 军事历史 (Osprey 出版社出版的军事历史是极棒的参考材料:
www.ospreypublishing.com);
- 体育运动和体育运动历史;
- 基本的演技;
- 车辆制图和构造 (最好用爆炸视图);
- 建筑架构 (内部架构和外部架构);
- 景观建筑;
- 布景构造技术 (戏剧和电影);
- 模块化制作;
- 动作电影“热点”和角色动作 (你可以构造你自己的动作库);

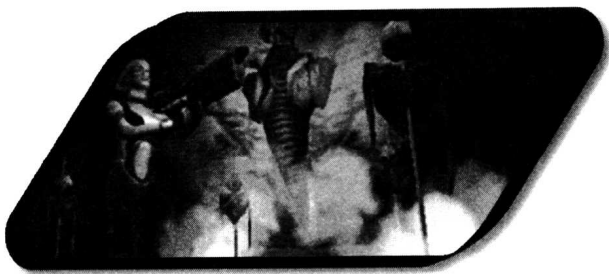
- 人类学;
- 绘画和图表;
- 博物馆收藏书籍 (大英博物馆, www.thebritishmuseum.ac.uk/, 还有许多主要的博物馆出版了这类书籍);
- 世界神话、寓言以及传奇故事(比如 Joseph Campbell 的《神话的力量》——*Power of Myth*系列);
- 美术和手工艺品的书籍 (尤指那些有很棒的照片参考资料的书籍);
- 服装设计;
- 界面设计;
- 写生、钢笔速写、视觉传达;
- 所有幻想或者科幻导向的事物, 所有经典的或者暂时的事物;
- 情节串连图板;
- 喜剧;
- 工业设计;
- 产品设计;
- 角色设计和构造;
- 电影特效 (www.cinefex.com);
- 生物雕刻。



APPENDIX

B

讨论过的工具



在前边贯穿各个考虑主题中，我们讨论了许多类型的软件工具，而且在这里，我整理出一些关于这些工具的参考资料。你最好牢记：在游戏开发中除了一些老工具（像 3ds max 或者玛雅）以外，游戏工具开发的发展还有很大转变和空间。



B.1 三维建模软件包

每一个三维建模软件包都有一帮支持者和诋毁者。许多美工和设计者都极端地热衷于他们自己选择的软件包，其实这里关键是“熟悉”这两个字。每一个软件包都有其长处和短处（或者是感觉上的长处和短处）。最好的做法是根据你的制作需要及目标挑选最适合的工具，而不是试着去——指出它们的优劣长短。不要害怕和你的同事分享知识。培训独立的开发人员需要很少的费用，甚至不需要。开发人员通常都会不断地学习并努力提高使用各种软件包的技能。

名称	网址	注释
Maya (玛雅)	www.aliaswavefront.com	玛雅是我个人最喜欢的三维软件包 玛雅个人学习版网上可以免费获得
3ds max/gmax	www.discreet.com	找一份儿 gmax 的免费版本！这是一个非常棒的软件包，很多开发者都在使用
LightWave 3D	www.newtek.com	另一个可选择的软件包
SOFTIMAGE\3D	www.softimage.com	另一个有很多支持者的软件包
Wings 3D	www.wings3d.com	一个开始学些三维构造很棒的软件包
Rhino3D	www.rhino3d.com	一个设计项目非常不错的应用软件

续

名 称	网 址	注 释
<i>DOGA</i>	<i>www.doga.co.jp</i>	一个为“新手”(从来没有接触过三维软件)准备的应用软件。这个免费的程序可以帮助你三维场景中构造物体时找准方向
<i>MilkShape 3D</i>	<i>www.swissquake.ch/chumbalum-soft/</i>	另外一个你可以选择的免费应用软件



B.2 美工工具

用于支持游戏美工资源创作的美工工具有很多种。下面列表中仅列出几个最受欢迎的。

名 称	网 址	注 释
<i>Adobe Photoshop</i>	<i>www.adobe.com</i>	是一个受欢迎的多用途的软件,可作为备用的材质构造软件包
<i>Adobe Illustrator</i>	<i>www.adobe.com</i>	许多美工人员都喜欢用它来画图
<i>Painter 7</i>	<i>www.corel.com</i>	用于处理材质细节以及很多其他的功能
<i>Wacom graphic tablets</i>	<i>www.wacom.com</i>	对于要在计算机上工作的传统美工人员,这是一个必备的软件
<i>ACDSee image viewer</i>	<i>www.acdsystems.com</i>	一个非常棒的图像查看和图像管理程序
<i>Flash</i>	<i>www.macromedia.com</i>	很棒的向量美工工具,可以完成惊人的视觉效果,而且文件又很小。这个软件包不仅适合于网页制作,也可满足许多游戏界面的设计需要
<i>Bryce 5</i>	<i>www.corel.com</i>	很不错的快速地形图构造选择
<i>Polytrans</i>	<i>www.okino.com</i>	在各种格式和软件包之间转换模型资源



B.3 关卡编辑

许多流行的游戏（包括多种类型）都附带有完整的关卡编辑器。这些编辑器允许玩家构造自己的关卡和使命。学习怎样把这些编辑器玩儿得很转并加入到 mod 社区中去，这是一种（当你开始能够构造精细的地图时）被开发商发掘（并被雇用）的很好方法。我建议你好好地学习一个第一人称射击游戏环境编辑器、一个即时战略游戏工具，当然还要熟练掌握一个流行的商业三维软件包。当这些编辑器你都用过了，你就会明白为什么从一个软件包彻底转向另一个软件包是很容易的。

名 称	网 址	注 释
<i>Unreal 2/</i> <i>UnrealEd3.0</i>	www.epicgames.com	<i>UnrealEd3.0</i> 关卡编辑器包含有零售版的 <i>Unreal Tournament 2003</i> 和 <i>Unreal 2</i>
<i>Radiant</i>	www.qeradiant.com	基于 <i>Quake</i> 引擎技术， <i>Radiant</i> 工具可以用于制作很多流行游戏的关卡，比如《绝地武士 2》和《半条命》
<i>Serious Editor for</i> <i>Serious Sam</i>	www.croteam.com	一个有意思的游戏和一个不错的引擎，并附带一个免费的编辑器
<i>The DEdit tool is</i> <i>part of the Jupiter</i> <i>System behind No One</i> <i>Lives Forever 2 and</i> <i>Tron 2.0</i>	www.touchdownentertainment.com	<i>DEdit</i> 是一个专为和 <i>Jupiter</i> 系统引擎技术一块儿使用的关卡编辑工具。如果你正在使用这个引擎的话，你的关卡工具就应该是 <i>DEdit</i>
<i>The Warcraft 3</i> <i>editor comes</i> <i>included with the</i> <i>game</i>	www.blizzard.com	使用流行游戏附带的编辑器是体验关卡构造很不错的方法。用 <i>Warcraft 3</i> 编辑器来开始研究地图构造吧！你可以使用在前边几章所讲的大多数的工作流建议

名 称	网 址	注 释
<i>Neverwinter Nights</i> <i>includes the Aurora</i> <i>Neverwinter Toolset,</i> <i>allowing players to</i> <i>create custom</i> <i>content</i>	www.bioware.com	学习关于游戏的事件脚本很好的选择。不过，你的脚本应该是游戏特有的。不要忘了用像 JavaScript 或者 Python 这样多用途的脚本语言试验一下

这些列表仅是那些允许玩家进行编辑的游戏中抽样出来的，还有许多其他的附带有用于用户内容创作的工具的游戏没有列在这里。这些工具在稳定性和质量上有很大的区别。你没有必要花费大量的时间去学习所有的这些工具，根据自己的游戏兴趣选择它们中的几个集中精力学好就行了。



B.4 中间件

中间件在你的开发工具和游戏硬件之间形成一个技术层，有了它，开发人员就可以把更多的精力集中在游戏内容的创作上而不是支撑技术的构造上。

名 称	网 址	注 释
<i>Renderware</i>	www.renderware.com	支持游戏开发工作和降低外部依赖性的一整套工具和技术解决方案



B.5 制作工具

下面这些应用软件是几种使用频繁的通用制作工具，它们能以数种不同的方式支持游戏开发工作。

名 称	网 址	注 释
<i>Microsoft FrontPage</i>	www.microsoft.com/frontpage/	我选择的超文本标记链接语言编辑器，用它来制作基于内部网的设计文档。任何一款你熟悉的编辑器都可以很好地为你工作

续

名称	网址	注释
Microsoft Project	www.microsoft.com/ office/project/default.asp	通用的工程任务分配和跟踪应用软件，用于帮助管理和预测游戏工程开发中的许多方面。现在，其企业版允许管理队伍和开发人员基于网络进行工程协作和文件共享
Microsoft PowerPoint	www.microsoft.com/office/ powerpoint/default.asp/	快速组合设计演示材料的好工具
Microsoft Visio	www.microsoft.com/ office/visio/default.asp	了不起的框图和流程图应用软件。对于帮助定义游戏流程逻辑和界面逻辑非常有用
NXN Alienbrain	www.nxn-software.com	资源管理软件，极其适合于跟踪大量的资源创作和编辑 workflow



B.6 声音编辑工具

对于繁重的音效工作，许多声音设计者和音效工程师都喜欢使用 Macintosh 机上的 Pro-Tools 系统。对于与游戏设计相关的音效设计，Sound Forge 极其有用。SoundMAX 在扩展游戏中的音效范围上是一个有趣的尝试。我的感觉是，游戏中的音效工作做得太少了！如果以我的方式开发游戏的话，在帮助定义游戏体验中，音效与视觉应该并重。

名称	网址	注释
Sound Forge	www.sonicfoundry.com	很棒的音效编辑工具！使用简便且功能极其强大（即使所有的工具可以使用……）
SoundMAX	www.audioforgames.com	帮助拓宽游戏中的音效限制



B.7 3-D 网页构造

对那些有兴趣研究网络的开发人员，下面的列表为他们提供了研究基于 3-D 网页内容的机会。

名 称	网 址	注 释
<i>WildTangent</i>	<i>www.wildtangent.com</i>	很棒的构造基于 3-D 网页的应用软件。包括免费的 <i>WTStudio</i> 工具，它是开始学习三维环境编辑很好的工具
<i>Anark</i>	<i>www.anark.com</i>	另一个交互工具和基于 3-D 网页的解决方案，它可以和数个流行的三维开发包对接



B.8 脚本语言

现在你已经知道，一般来说你最终得使用专为项目挑选的那门儿脚本语言，所以，最好能换着多学习几门儿脚本语言。简单的熟悉也是很有用的。在一个项目中连续使用其中一种语言达十五个月后，你就会对它非常精通。

名 称	网 址	注 释
<i>Microsoft Visual Basic</i>	<i>http://msdn.microsoft.com/vbasic/</i>	常见的行为脚本语言选择
<i>JavaScript</i>	<i>www.ngweb.biz/software/djsedit.shtml</i>	另一个常见的脚本语言选择
<i>Python</i>	<i>www.python.org</i>	开源 (<i>open source</i>) 语言，正逐渐流行起来的一门可选的脚本语言



B.9 游戏动态库

这些基于软件的库加入到特定的游戏引擎构架之中，从而向开发人员提供数种动态的解决方案。

名称	网址	注释
<i>Havok 2</i>	www.havok.com	游戏物理的 / 动态的解决方案
<i>Karma</i>	www.mathengine.com	另一个非常棒的游戏动态解决方案



B.10 动作跟踪

下面两个例子中，一个是动作跟踪工作室，另一个是动作跟踪的硬件供应商。

名称	网址	注释
<i>House of Moves</i>	www.moves.com	位于加利福尼亚州 <i>Santa Monica</i> 常用于动作捕捉的地方
<i>PoIhemus</i>	www.poIhemus.com	动作跟踪硬件解决方案



B.11 控制台 / 个人计算机编程语言

控制台和个人计算机游戏是 C++ 开发的世界。

名称	网址	注释
<i>C++</i>	www.research.att.com/~bs/C++.html	用于控制台和个人计算机开发的面向对象程序设计语言选择



B.12 MMOG 框架解决方案

随着数家公司正在努力创作成功的 MMOG 游戏，两家喜欢冒险的公司开始提供支持这种类型游戏构造的技术。

名称	网址	注释
<i>LithTech Discovery</i>	www.lithtech.com 或者 www.touchdownentertainment.com	<i>Jupiter</i> 系统 (用于控制台和个人计算机游戏开发) 的制作人提供的 MMOG 解决方案
<i>Tera Zona</i>	www.zona.net	MMOG 游戏制作者的开发解决方案 供应商



B.13 无线游戏开发

无线游戏和娱乐程序仍然发展的初期。下面为你列出的是目前的两个最主要的开发方向。

名称	网址	注释
<i>J2ME</i>	www.borland.com/jbuilder/	对于 <i>J2ME</i> 开发, <i>JBUILDER</i> 开发环境是一个流行的选择
<i>BREW</i>	www.qualcomm.com/brew/	你可以在这里下载 <i>BREW</i> 软件开发包并开始学习 <i>BREW</i> 开发



B.14 向孩子们介绍游戏设计

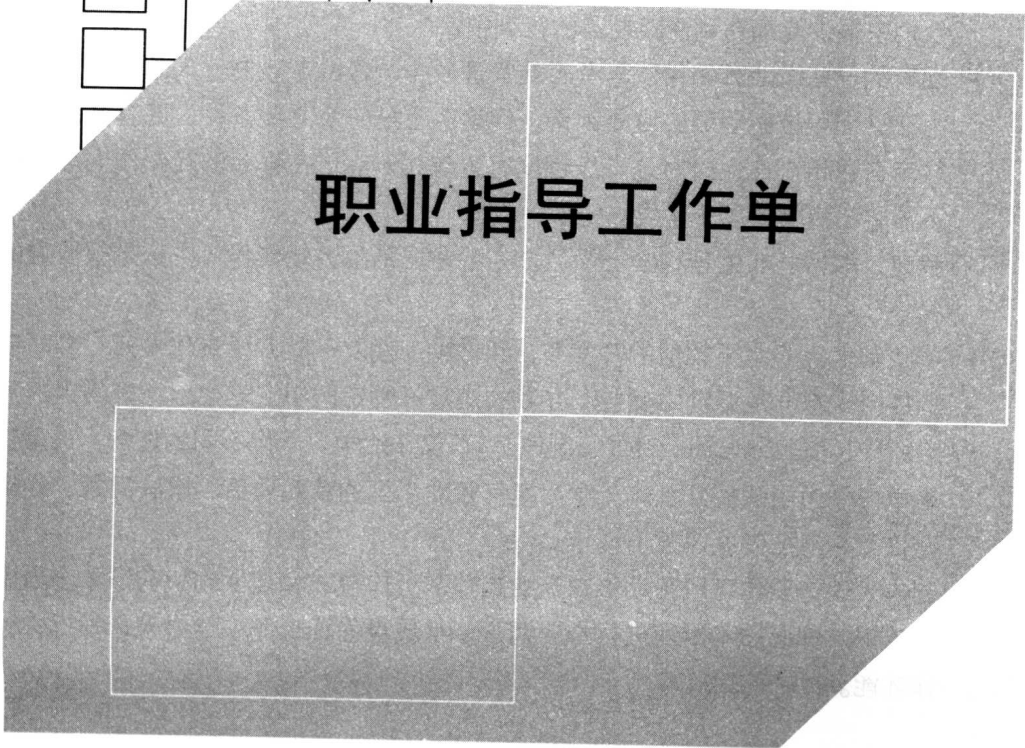
对于那些有兴趣帮助指导孩子们成为未来的游戏设计者的人来说,这个工具是向孩子们展示游戏设计思想很棒的方法。

名称	网址	注释
<i>Klik 'N' Play</i>	www.clickteam.com/ English/klikplay.php	一个适合于孩子们的很酷的游戏构造系统

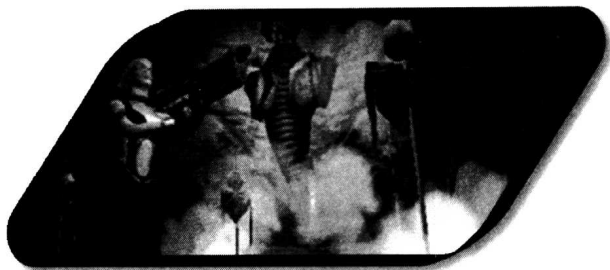


APPENDIX

C



职业指导工作单



为了帮助你进一步明确在游戏开发中的目标，我整理出一份儿简单的职业工作清单，并按照工作类别分别给了一些建议。这些并不是用来作为工作的准则，它们只是一些简单的指导条款，两者最终的导向是不一样的。希望这份儿材料对你有一定的价值。但愿，这份儿材料能够作为一个出发点来帮助你在游戏产业内找到你所喜欢工作。



C.1 如果你想要寻找游戏制作工作

- ▶ 许多游戏制作人是从小质量的游戏测试员做起的。没有扎实的质量保证经验，他们中很少能加入到游戏制作的行列中。
- ▶ 现有的一些游戏公司喜欢具有工商管理硕士学位的游戏制作人职位应试者，因为在这个职位有明显的商业因素，但这决不是一个必要条件，而且开发商或发行商在聘用制作人时对这一点也没有达成共识。现在许多游戏制作人没有或不需要工商管理硕士学位。
- ▶ 我建议取得扎实的理科学士学位。在我所遇到的游戏制作人中这是很普遍的情况。
- ▶ 寻找其他经验水平相似的开发商，和你的新朋友一起开始制作游戏！参与制作的每个人都会从经验中受益。在我的游戏开发部门中，学生承担开发组的角色（编程、美工、设计、音效、制作），不仅开始学习承担不同的任务，更重要的是学习作为团队相互合作。学习如何管理小型的游戏项目、解决问题、作出设计权衡并承担多项任务。
- ▶ 制作人这个职位可以帮助你了解产业趋势、运作模式和商业动机有一个全面深入的认识。你必须不断地进行产业调查。从质量保证工作中走出来，发挥你的制作才能。



C.2 如果你想要寻找游戏美工工作

- 集中精力培养你的传统技能（素描、油画和雕刻）。
- 完成艺术或设计类的文学学士学位或美术学士学位。
- 尽早开始把你正在培养的传统技能应用到数字世界中。开始使用你在外面已学到的传统技术，在你喜欢的软件包中进行试验。
- 开始把精力集中到你喜欢的领域。如果喜欢动画，那就大胆地从事动画制作。如果喜欢造型，学习你能学到的关于造型的一切。如果对材质绘画感兴趣，努力使自己成为优秀的材质大师。现在所有开发角色越来越严重地专业化了。
- 记住，如果你不能直接获得游戏开发中的美工工作，可以从质量保证工作开始，不断地向你身边的开发商展示你的技能和渴望。质量保证始终是一条通向游戏开发工作的途径。
- 保持你的思维和技能多样化。不要因为你明白你将成为一名动画制作人，就忽视这些情节串连图版课程。保持思路开阔，随时准备吸收一切。以后你会惊奇地发现这些确实很有用。你也可以一直为非游戏公司做兼职来增加你的收入，即使你正在寻找每一个你所获得的游戏工作的机会。



C.3 如果你想要寻找游戏编程工作

- 从自学编程开始。记住，控制台和个人计算机开发是用 C++ 完成的，它是你打好编程基础很棒的语言。为了帮助你入门，网站 www.gamedev.net 上列出了许多好的参考书目。用自己的技术编写自己的游戏（如果能得到朋友的帮助的话当然更好了）。卡片游戏、益智游戏、街机版设计游戏或者非常简单的角色扮演类游戏都是很好的选择。
- 这是一个三维的世界。要作出让人感兴趣的三维作品需要有强硬的数学和编程能力。尽量独立地培养自己对三维图形编程的深层次理解。
- 获得数学或计算机科学理科学士学位。
- 获得编程工作的最好方法之一是拥有大量的代码实例，这些你可以从以前所做的游戏工作亲手编写的代码中获得。



C.4 如果你想要寻找质量保证工作

- ▶ 大量地玩游戏！这是你的课外作业，游戏迷们一定会高呼万岁！！
- ▶ 根据你居住的地理位置，调查你所在的地区是否有游戏发行商。你可以在 www.gamasutra.com 网站上找到这些清单。与发行商洽谈时要表现得很有礼貌——他们是非常忙的！询问一下他们现有的或预计将要实施的测试需求。可在 www.monster.com 网站中查找“游戏测试员”的工作。
- ▶ 获得你选择的任何专业的学位。如果你希望做质量保证的管理工作，学位一定能帮你进入较大的公司（大公司的运作正在向规范化的方向发展）。



C.5 如果你想要寻找游戏音效工作

- ▶ 成为一个优秀的作曲家和音效制作人。你可以在互联网上得到许多免费的声音播放和编辑软件。开始调整枪炮声、身体撞击声、恐龙尖叫声以及湿乎乎的拍打声。努力用占用空间最小的文件实现最美妙的声音。你可以在网站 www.fatman.com 中查找一些关于游戏声音大师狂热的生活方式的信息。
- ▶ 做音效工作并不需要接受专门的教育，但是有也不碍事儿——至少可以构造自己的朋友关系网和音乐人关系网，以后可以召集他们彼此帮助完成工作！因此去获得艺术学士学位。去学习计算机音乐（据毕达哥拉斯[Pythagoras，古希腊著名的哲学家和数学家]说，音乐可以陶冶情操）。许多学校现在开设计算机音乐和音乐技术课程。
- ▶ 依我看来，你可以在网站 www.sonicfoundry.com 上找一些最好的计算机音效工具。学习这些工具对开始学习游戏音效是非常好的。
- ▶ 注意你的竞争对手！！聆听你所玩的每一个游戏中的最棒的音效，不要管它是什么类型的（界面、气氛、立体声、效果等等）。试着重新创作出这些你喜欢的感觉和声音。最好是能在当地找到需要制作游戏声效的人，给他们提供帮助。



C.6 如果你想要寻找商业关系或者市场工作

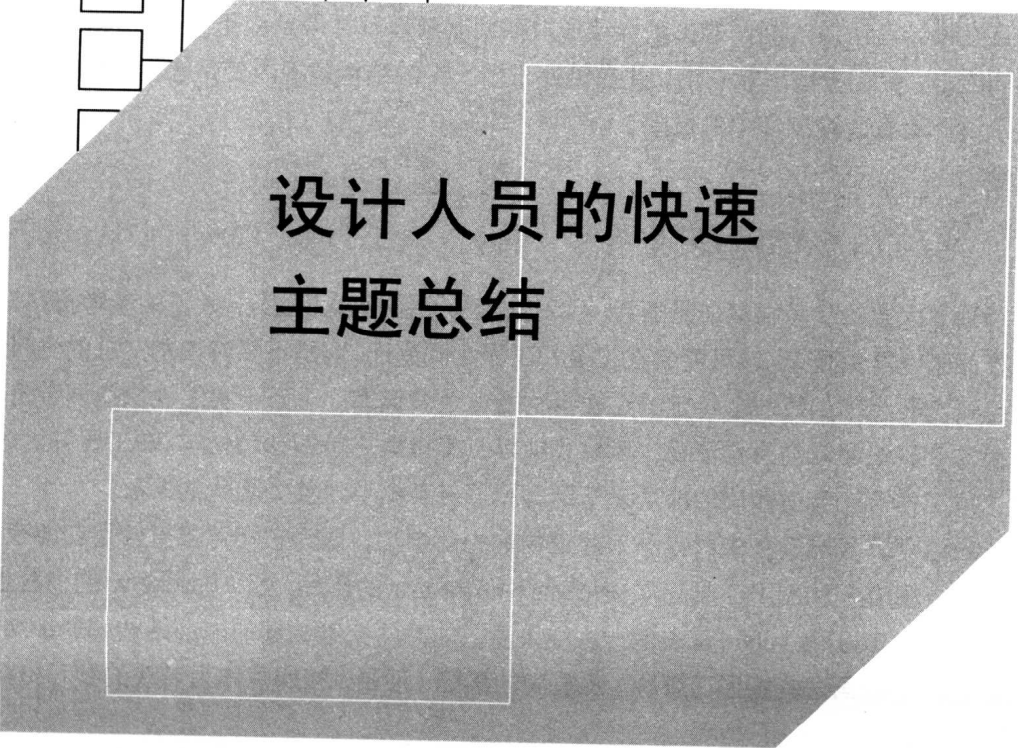
- 大量地玩游戏！你要能直接和你的观众——游戏玩家取得联系。没有更好的办法可以帮助确保你知道什么地方能找到你的观众，同时也没有更好的办法让你像你的观众在购买一个游戏时所考虑的那样去思考。
- 让自己熟悉游戏竞技社区的各个方面，包括玩家对游戏类型的偏好、经常访问的网站、以及所有相关产业的杂志。
- 有一些直接的游戏开发知识是很有用的。花点儿时间在质量保证部门和开发团队中转转。尽可能多的弄清楚开发人员在交付游戏时所面临的挑战。
- 获得你希望的任何专业的学位。
- 一些游戏发行商的公共关系部和市场部都有实习生，这都是进入一个公司很好的途径。寻找需要你的发行商，咨询一下有关实习生的职位。



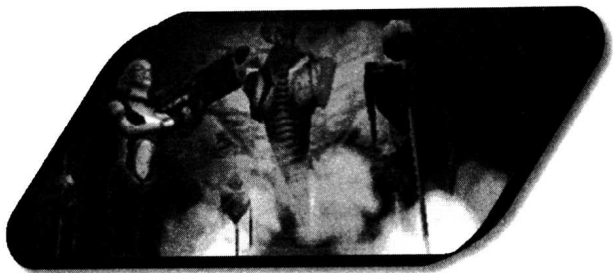
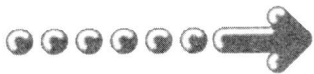


APPENDIX

D



设计人员的快速
主题总结



在这一附录中，我将我们讨论过的一些重要的概念整理出一份儿大纲。接下来的每一条目都是设计人员在团队中的作用的的重要组成部分。设计人员必须保持灵活性并随时准备迎接挑战。这种涉及深层次、特定领域知识和专业化的任务的思想，正好和大多数开发商所遵循的“少花钱、多办事儿”的经营方式相反。游戏工作室努力寻找着具有多方面经验的开发人员，这一点完全可以理解，但是，这些愿望必须和下面的做法相一致：面向利于特定游戏项目，以合适的方式寻找具有深厚的专门领域技能的开发人员。

随着游戏的内涵逐渐明朗集中，设计任务也在逐渐演变。这些材料在摘要中提供，并且在设计者对游戏制作的贡献标准形成期间逐渐理解并得到应用。这些材料以总结的形式给出，我们应该理解它们，并在仍处于增长中的结构范围内使用它们（这些结构定义了设计者在游戏制作中的贡献）。



0.1 参考资料

拥有大量的参考材料总是有用的！对此我怎么强调都不过分。大量的参考资料对游戏产品来说至关重要。你可能会在工程和工程之间变迁，但是不要害怕构造总的资料库，而且一定要为你的每一个特定的工程都构造一个资料库。千万不要忽视快速调查阶段，因为这个阶段常常会为构造视觉风格和纯粹灵感搜集充足的参考材料。在工程开始时不做这些工作可能会为最后的成功设下绊脚石，甚至会导致整个团队的失败。

在这里，我准备集中讨论一下建筑物的内部结构，因为它们可以直接用在游戏中。图 D-1、图 D-2 和图 D-3 显示了 Pieter Saenredam 的作品：St. Bavokerk 的小教堂。Saenredam (1597 ~ 1665) 花去他一生中大多数的时间描绘教堂的内部结构和外部结构，并以他对透视画法的把握而著称。建筑物的素描、油画、和照片作为游戏的参考材料是

非常有用的。最好的办法就是留意所有可以添加到你的游戏参考库中的资料。但有一件事儿是肯定的：你需要大量的有关建筑物内部结构和外部结构的参考资料。

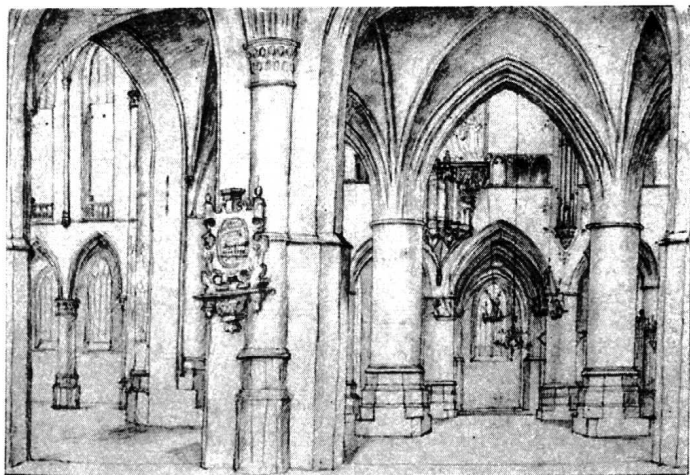


图 D-1 St. Bavokerk's Chapel, Haarlem, 彼得萨恩列达姆 (Pieter Saenredam) 作品 (场所素描, 1634 年)

注: Haarlem, 荷兰。



图 D-2 St. Bavokerk's Chapel, Haarlem, 彼得萨恩列达姆作品 (建筑素描, 1634 年)

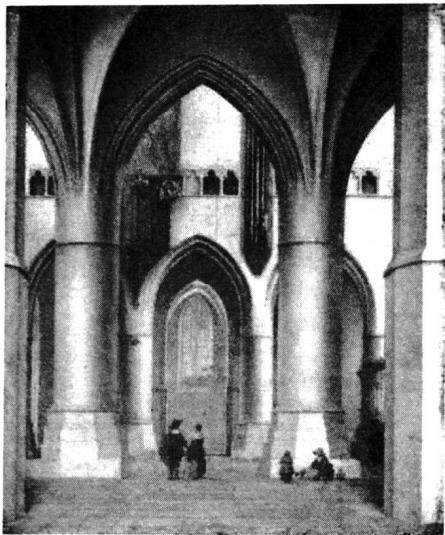


图 D-3 St. Bavokerk's Chapel, Haarlem, 彼得萨恩列达姆作品 (油画, 1635 年)

我最喜欢的一些素描出自 Giovanni Battista Piranesi 和 engraver (1720~1778) 之手, Piranesi 不但是位考古学家, 还是一位建筑师。我个人很幸运地看到他们的一些素描作品。Piranesi 在剧院设计和布景设计方面对我产生了深远的影响。他能用非常简单的方法影响游戏视觉效果。为了强调参考资料在设置视觉风格和激发灵感方面的重要性, 我想和你分享这些素描作品。

图 D-4 向你展示了素描是如何不费吹灰之力就启发了你的道具设计和道具细节设计。注意观察那些靠近该油画顶端左侧的链条和悬臂梁。像这样的素描也可能会影响到舞台设计 (道具之间的相互关系)、整个场景的布局, 甚至会影响关卡中某一部分的道具放置细节。你可以简单地把这张图作为你的设计文档的关卡说明书中的参考。

图 D-5 可能会对你构造和布置关卡中的桥梁结构的方式产生一定的影响, 包括楼梯结构设计, 就像在图上我们看到的那种环绕柱子的楼梯结构一样。注意一下这幅作品的中央位置是怎样聚焦在两个悬索桥上的。这种结构形式甚至可以启发你解决一些小问题。这幅图可能会启发人们创造出一些有趣的桥梁和路径布置, 或者像图中柱子上悬挂的灯笼一样, 仅仅可能影响简单的灯光道具的想法。

图 D-6 提供了有趣的壁龛、隧道和桥梁结构的参考材料。



图 D-4 皮拉内西 (Piranesi) 的 Carceri d'Invenzione, 出自第二版中的插图 plate from
注: 皮拉内西 (1720~1778), 意大利建筑师、艺术家, 他的罗马及其废墟的版画为新古典主义复兴作出了贡献。



图 D-5 皮拉内西的 Carceri d'Invenzione, 第二版版画的插图 VII



图 D-6 皮拉内西的 Carceri d'Invenzione, 第二版版画的插图 XIV

图 D-7 是我非常喜欢的另一幅作品，它的弯曲的拱门，蕴涵戏剧效果的灯光，突兀的墙面浮雕和突出的桥梁结构，都对我产生了很多明显的影响。

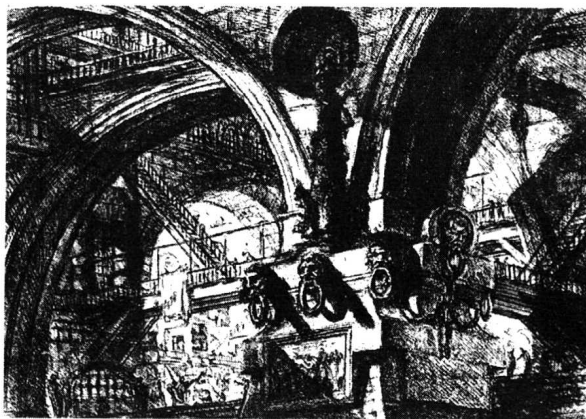


图 D-7 皮拉内西的 Carceri d'Invenzione, 第二版版画的插图 XV

图 D-8 是一个有关建筑物内部结构的例子，图中的道具以及道具细节设计的参考更加优秀——最值得一提的是图上狮子护卫的石雕壁画，它雕刻在墙壁上。直到前不久，把这种复杂设计细节溶入游戏环境中才成为可能。在制作游戏中，即使能得到这样的美感，但常常还需要折中考虑游戏的当前运行性能，你需要从大多数道具之中获取足够的

资料，同时必须明智地作出选择。

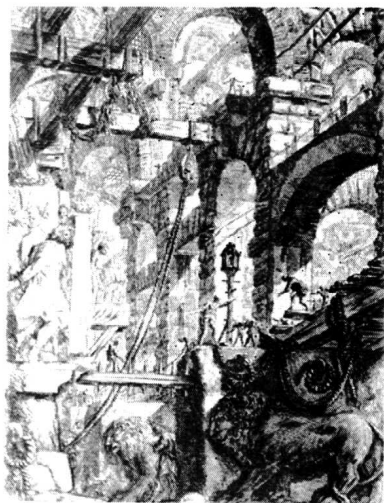


图 D-8 皮拉内西的虚构监狱，监狱 VI，狮子石雕壁画

图 D-9 中展现了更多潜在的道具、细节和舞台设计影响：钉有大钉子的轮子、大型铁架、锁链和悬挂设备。



图 D-9 皮拉内西的虚构监狱，扉页



0.2 设计文档的书写

www.gamedev.net 网站上有许多文档模板可供你使用。这些文档模板都可以作为你的有用的出发点。一份设计文档要快速地为你的开发团队提供动态且有用的信息。一份全面的设计文档应当把游戏的所有特征和功能都加以定义。这些游戏的特征和功能可能会改变，但是游戏的各个基本的方面应该有一个合理的出发点，这些在设计文档都作了说明。如果在设计文档中：过多地讨论了总的设计方向、设计者对一些事情的个人观点展开叙述了数段甚至数页、每隔一页就有关于最后要制作的游戏什么出错了的冗余叙述、无休止地与市场上其他游戏进行比较、或者熟悉的游戏背景故事长篇大论的细节描述，你的团队将会不愉快。

根据游戏类型或游戏风格的不同，大多数设计者设计出的文档也多少略有不同。游戏设计文档的大小由游戏风格，游戏平台等等因素决定，其类型因此也是各种各样的。我写过四百页的设计文档，也写过二十页设计文档。现在，一个超过二百页的设计文档很常见，而一个大型多玩家在线游戏的设计文档很容易就会达到五百页。我为一个第三人称动作游戏提供设计信息和为一个竞速游戏文档有很大的不同。但是，我的设计文档在格式上通常都有一个共同点，就是使用一个左边的工具条来包含一系列带有标题信息的主题，选定这些主题后其信息就会显示在主“框架”视图中。这种文档结构看起来很方便，而且开发人员通过标题很快就能找到他们所需要的特定设计细节。

你得快速有效地给你的开发团队提供设计信息。游戏美工和动画师应该知道人物角色应该去怎样构造、怎样材质贴图以及假设它怎样运动。人工智能程序员需要通过实例了解游戏为人物设定好的行为方式。事实上，每个人都应该知道一些关于打算怎样使用物品和它的特征细节。

如果这些可以用于权衡游戏制作的内容在文档中没有写清楚，你的团队就会迷失方向。根据设计文档所说明的游戏种类的不同，我会作出不同格式的设计文档。我竭尽所能使我写的文档中炫耀的信息让每个人都能够容易地理解。要使用表格，还要总结。使用简单的列表清单。不要让团队中的成员自己去“挖掘”那些他们需要的信息。

正如我所提到的，我使用 HTML 写代码，把那些带有信息的长工具条与一个框架相互关联。通过选择标题，你只需要简单地浏览左边的工具条，并通过简单快速地选择主题，就能从中获取你想得到的任何信息。“做成框架干什么？”一名专家或者程序员会

有疑惑。这时只要轻轻一点，“哦，是这么回事儿”。



0.3 快速建模

根据你的团队的组织结构，作为一名设计者或者头头，你可能得大量建模。起码你最好是能够构造临时模型，这样你就能够在美工完成最终模型之前尝试游戏。掌握一些基本的建模技能非常重要，而且你的这些技能会随着时间和接触次数增多而逐渐提高。

可以为每个需要资源的主要类别构造一个基本的多边形预算，这些资源由你的技术团队构造。那么就往你的场景中丢临时道具、临时环境细节和临时人物吧！你的人物角色造型大概需要一千五百多个多边形，如果你的游戏中有三十个人物角色，就是四万五千个多边形；道具可能平均需要三百到五百个多边形。场地或者环境的多边形预算还要高一些，而且由许多其他因素所决定。为了知道这些多边形预算数据，你的团队一定要尽早和你的技术团队一起工作。

不要等到最终的美工作品完工后才把它放到游戏中！即使是很粗糙的模型，先建起来再说，并立刻把它放到游戏中，越早越好！有关哪些能用、哪些不能用的问题，你要尽你之所能找出来。使用临时模型可以帮助你发现一系列存在的问题，这就为你在美工制作中节省出时间。

对于设计新手来说，有许多关于三维建模的好书和录像带供你选择。不要再犹豫了，赶快去 www.aliaswavefront.com 网站下载玛雅个人学习版，还有 www.discreet.com 网站的 *gmax* 或者 www.wing3d.com 网站的 *Wings 3D!* 然后，试着用这些软件创作一些简单的人物造型和道具。



0.4 布局与舞台设计

正如第 1 章到第 3 章中所讨论的，你会把地形图和关卡规格说明写到你的设计文档中，不是吗？根据这些信息，你就能设计出你的关卡基本布局和舞台布置。你会创造一个适当的游戏环境，并用正面角色、反面角色、道具和物品丰富游戏的内容。设计者们花费大量的时间管理关卡布局 / 舞台布置的构造和改进过程。

你将随时随地获取可用到游戏中的新的美工、代码和音效等资源。资源管理 (www.nxn-software.com) 和文件参考将很快成为你最好的“朋友”。*NXN* 是数字资产管

理软件的主要供应商。通过访问该站点,你可以学习到更多的关于资产跟踪软件的知识。你想要拥有快速寻找游戏资产的能力,这样你就能够以最快的速度有效地用新的、其他的或者更新后的部件替换已有的游戏部件。或许,在你的关卡中有三十五根街灯柱,但是美工部刚刚才把新改进的街灯柱送来。你肯定不乐意手工或者通过文件名去逐个找到关卡中的这三十五根街灯柱;你希望能够通过简单地修改文件参考指针一次性地把它们都改过来。

由于布局和舞台设计要求及其他的相关事宜经常会发生变化,设计者常常需要得花费大量的时间构造和改进游戏场景。增进你在三维环境中构造场景的能力。

如果你对构造场景过程不熟悉,从 www.doga.co.jp 下载一份儿 *DOGA*, 用这个共享软件包中的免费部分开始构造场景。虽然在实际游戏制作中你会用到 *DOGA* 软件,但是它将帮助你熟悉三维创作。*DOGA* 软件会帮助你学习在三维环境中使用预制件构造场景



0.5 脚本书写

不要忘记,现在对于大多数游戏设计职位,脚本书写能力已经成为一项基本要求;所以应该至少精通一门脚本书写语言,诸如 *JavaScript*、*Python* 或者 *Visual Basic*。在你书写和编辑简单的脚本时使用这些语言越多越好。

一个设计者不需要书写大量的错综复杂的脚本,但是他或她必须知道应该怎样简单地用编辑器去修改基本的脚本数值。



0.6 地图或者关卡制作

无论你是一个制作关卡和地图的关卡设计人员,还是个游戏中从头到尾的细节的设计人员,为了实现游戏的功能,你都会用到一系列的软件工具。熟悉使用这些工具制作来制作游戏的惟一方法就是反复——反复地经历制作、测试、改进、再构造、再测试、再改进,如此循环地做。现在和以前比较起来,玩家和开发商拥有更多的可用工具。这样人们会很容易得到游戏应用工具,但是不利于人们掌握这些工具。

我不知道你使用的是什么工具软件,你希望使用什么工具软件,或者你将要使用什么工具软件。是的,我们想要的是拥有强大功能的游戏开发工具,拥有强大功能的、有着惊人的灵活性的编辑器,但是,这些工具最终能够发挥出的真实威力就是你在这些工

具中体现的那些——你自己通过无法计量的时间玩游戏所构造起来的对游戏的洞察力。

作为一般性的建议，我建议你了解一个第一人称涉及游戏的编辑工具，了解一个实时战略游戏或角色扮演类游戏方面的工具，还有一个三维软件包。这会帮助你打好基础，同时也会为你创造机会。



0.7 音效

根据团队的人员配置，你的制作人可能会为团队聘请一个音效总监。作为设计者和制作人，我们总是尽力使各部分的音效获得最大的感染力。千万不要忽略音效细节。你最好知道怎样去编辑基本的声音、怎样创作占位声音以及效果。有时候你也许会创作出一些有用的东西来！要充分利用 www.sonicfoundry.com 网站上提供的那些优秀工具软件。

尽早为音效做准备。在你设计考虑中一定要把音效放在重要的位置。只有游戏画面才是最重要的，但是这种想法是错误的！音效在帮助你传达思想和情感的时候往往能打到事半功倍的效果。



0.8 测试

在质量保证部门，你的工作就是测试。在成为官方正式公布的错误之前，尽力去把它们改正过来。你可能会不喜欢从质量保证部门反馈回来的一些信息。听好了——失败往往都是共同分担的。如果游戏玩家对你的游戏感觉很差，你就必须解决一些实时问题去改善这种情况。这将会给你带来更多的工作。这就是改进游戏的代价；否则就是失败（把你的游戏随便处置）。一定要让这种测试在你的工作中继续下去。



0.9 支持软件

为了改进你们的游戏设计和游戏制作过程，不要忘记去尝试一些新的软件。我一直以来使用的软件有 *Word*、*FrontPage*、*Visio*、*Project* 和 *PowerPoint*。随着游戏产业向“中间件”（www.renderware.com）发展的趋向，在实际的制作软件和工具中，可能很快就会产生一些标准化的软件。到今天为止，这种情况还没有任何苗头。

虽然 *Visio* 在构造界面特征和功能、为游戏内核绘制视图流程以及完成其他图表任务方面，的确是一个很好的工具，但是很少有人使用这个软件。



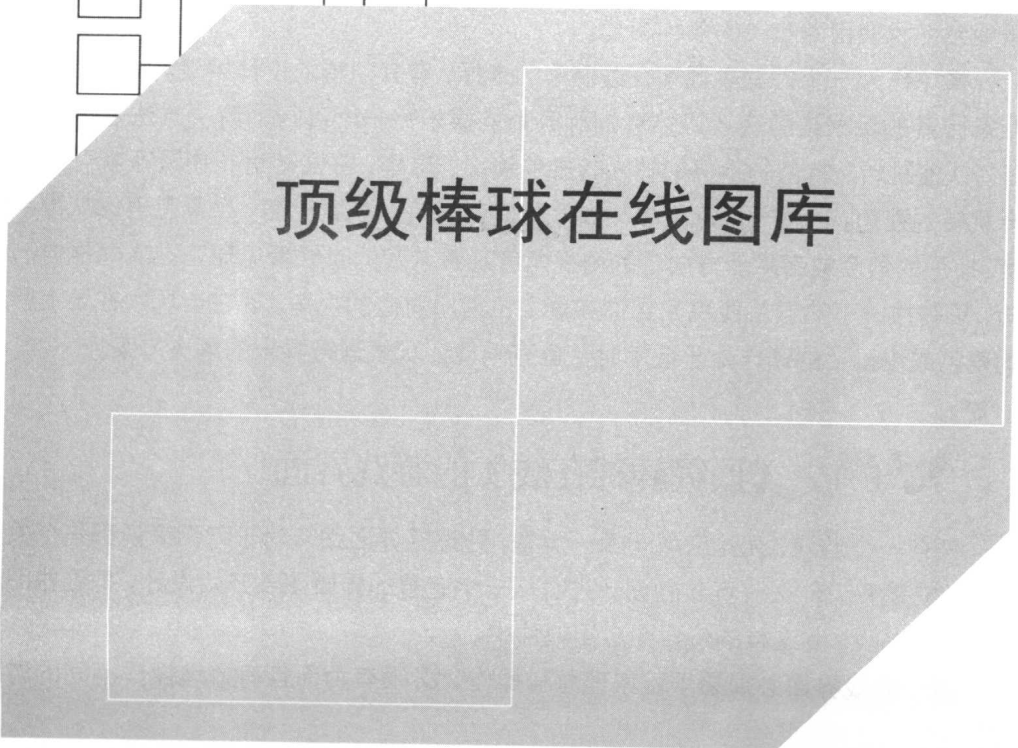
0.10 聚焦团队

面对问题，你应该按照计划反复查对游戏的核心内容，还要支持团队的工作。让自己成为一名解决问题的专家，并努力使自己周围的事情变得越来越好。自己一定要有原则。一定要求知若渴。不要使你的团队丧失斗志，同时不要忽略团队的重要事情。

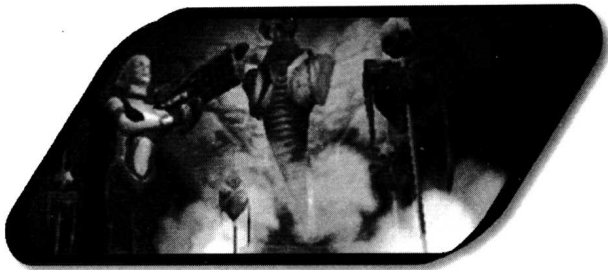


APPENDIX

E



顶级棒球在线图库



作为一份儿参考资料和关于游戏中美术资源的快速讨论，我在《顶级棒球在线》——*Ultimate Baseball Online* 中收集了几幅屏幕快照。这个游戏我们在第 8 章中已经介绍过了。我为每一幅屏幕快照都作了一些评论，希望这对你会有所帮助。就如我早些时候提到过的，这个游戏项目不仅在技术方面需要大量的创新，而且在设计方面也一样——作为顶级棒球制作商的设计和 产品方面的顾问，我发现这一因素从一开始就令人十分兴奋。尝试去开发、制作并销售有史以来最好的大型多玩家在线运动游戏，必然会伴随着许多风险和挑战。我为每一个对顶级棒球作出贡献的开发人员以及他们在游戏的制作中作出的努力喝彩。我要感谢 *Andy Wang* 和 *Netamin* 公司的每一个人，感谢他们慷慨地允许我使用游戏中的各种图片。

棒球作为一个大型多玩家在线游戏的选择，有其独特的设计挑战。如何使在场地上准备比赛的玩家或者在场边休息的玩家都能娱乐？一个可行的解决方法是提供一种具备“快速跳转”特色的不停的转来转去的镜头，这样，在游戏进行中玩家就可以自由选择跳转的视角，以便于玩家在参与游戏的同时也可以观看比赛。如果正当你从投手的角度、捕手的角度或者从上方向下的俯视角度观看游戏的时候看见球正在飞向你自己的方向，只需按一下热键就能跳回到你在场上位置所对应的视角。现在当玩家在场上所对应的角色或位置上的动作处于低潮时，也有其他让玩家进行娱乐的解决方案。



E.1 《顶级棒球在线》的游戏界面

任何一个游戏，无论它属于哪一类型，处理好界面细节对于它们来说都是至关重要的。这就是我为什么一直鼓励设计人员从一个完善的界面图表开始设计，把这些图表放在像微软的 Visio 这样的流程图应用软件中。

每个游戏界面必须易于玩家理解和操作，必须有力而且有效地操作不同的游戏细

节。这是一个一直都很难以实现的平衡点。随着玩家和开发者不断提供的反馈信息，游戏开发也越来越深入，大多数的界面细节会从根本上发生改变。在最初的时候，必须从一个制作良好的界面图表开始。一定要仔细的画出你的界面逻辑、界面与界面之间的相互转换以及每一个单独界面的逻辑细节，以避免大量的重复性工作。

■ 玩家构造界面

图 E-1 是一个基本的玩家构造界面（这是游戏的很早期的一个版本），允许玩家配置大多数基本的玩家细节，但是几乎所有的游戏细节都不能够配置。这个玩家构造界面还没有制作完。记住，你的每一个界面的设计在开发过程中都要考虑到尽可能多的扩展以及调整。

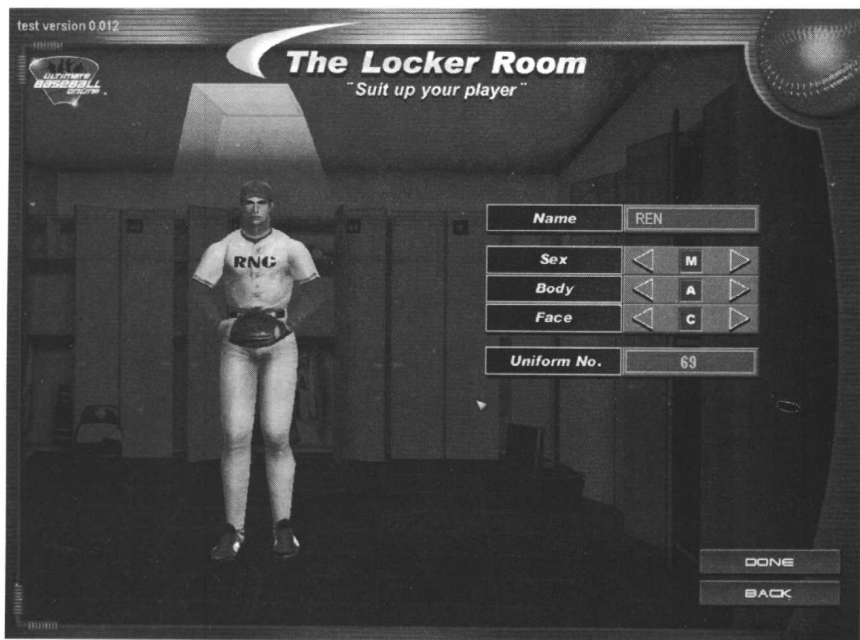


图 E-1 更衣室 / 构造玩家

当你发现有更好而且更简单的控制玩家与游戏之间的信息传递方式的时候，你常常会对界面进行修改。始终要尽力避免为改进过程中把物理屏幕空间消耗光的情况出现。

■ 前期的游戏界面设置

图 E-2 是一个游戏管理界面的例子，它允许玩家加入或者构造游戏，并在等待游戏开始的时候在游戏大厅里同其他玩家进行交谈。最好不要将物理屏幕的尺寸或者背景嵌入到界面图形中，否则会很难进一步修改。

相反，你可能希望这些尺寸可以调整，以取得界面上的最佳平衡。就像你可能想到的，游戏中经常会提供一个或者两个界面类型，根据这些界面的定义，它们仅仅需要更多的显示空间。

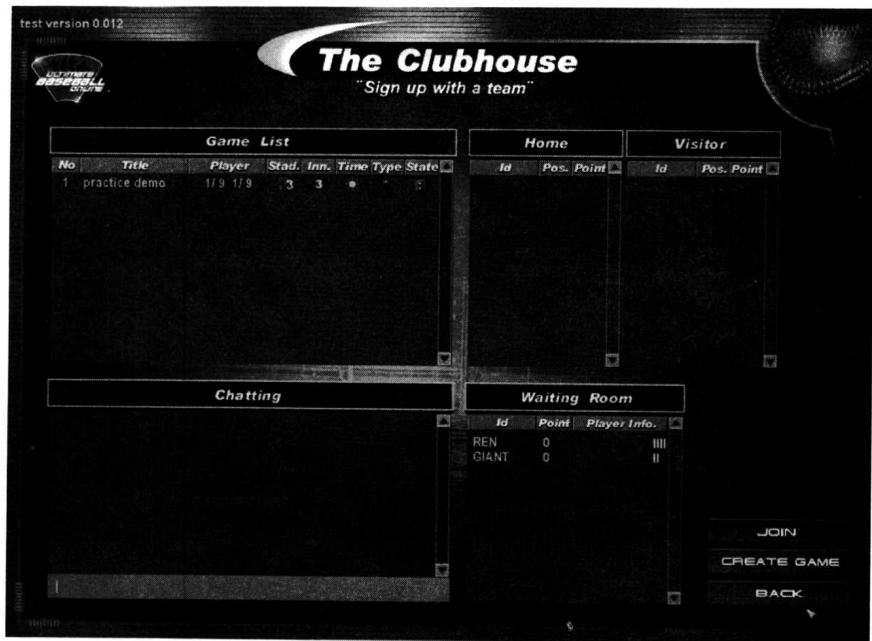


图 E-2 俱乐部会所 / 游戏设置

■ 游戏细节

图 E-3 种给出的例子展示了一个早期版本的游戏界面细节。一旦玩家构造了游戏，许多游戏的设置都可以根据玩家的偏好作调整，比如游戏场地的选择、游戏长度，以及 / 或者一些比赛条件等等。此外，在开发的早期阶段，最好能尽量保持这些界面的灵活性。

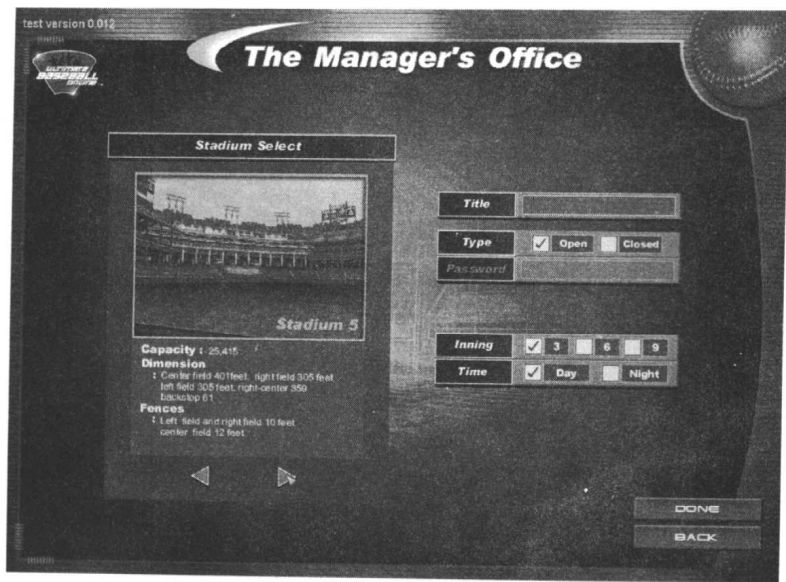


图 E-3 管理者办公室 / 游戏的一些细节

■ 位置选择和击球顺序

到目前为止,你都可以从此书所展示的每一个界面中看到:在这些界面和它们所提供玩家的选项之间,存在一种“界面逻辑”和一种自然的联系。这些都应该在设计文档中清晰地说明。图 E-4 展示了玩家如何设置他们在场上的位置以及如何被加入到一定顺序的击球阵容之中。



E.2 基本角色的构造

在角色开发过程中有四个重要的角色制作阶段:概念图的绘制、线条框架的模型、渐变表面渲染模型、全材质模型。图 E-5 展示了一个角色的线条框架模型、渐变表面渲染模型以及全材质模型。

对于每一个角色、道具、物品以及环境细节,多边形预算、材质贴图的大小、解析度的规格说明等信息都要提供给美工。在这些预算的限制下他们的任务就是创造出最引人注目并且最具吸引力的角色。

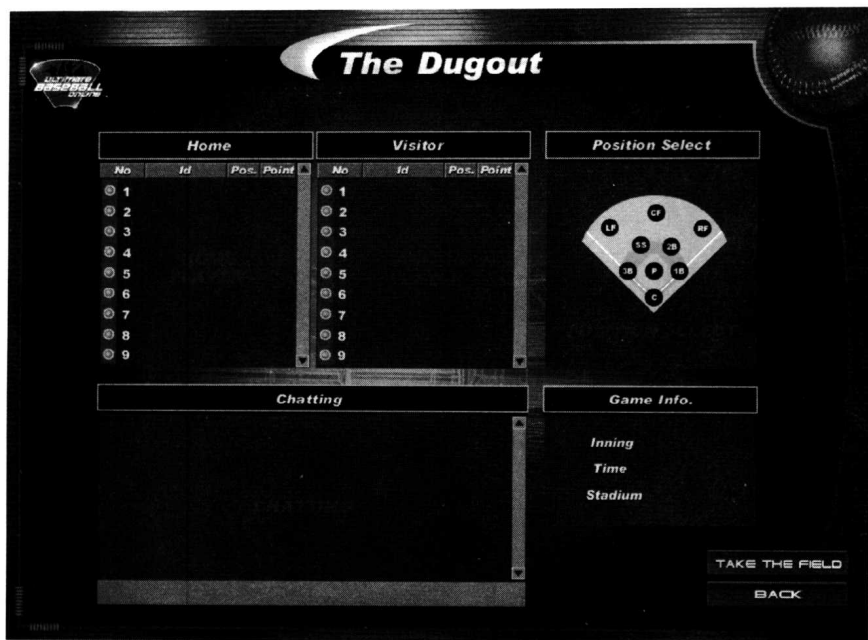


图 E-4 休息室 / 球员占位选择



图 E-5 线框模型、表面渲染模型以及全材质模型

图 E-6 从线框模型和全材质模型两种方式展示了一个防守角色的动态姿势。记住，

在三维软件包中，动作捕捉的数据信息应用到这个模型来构造动画。

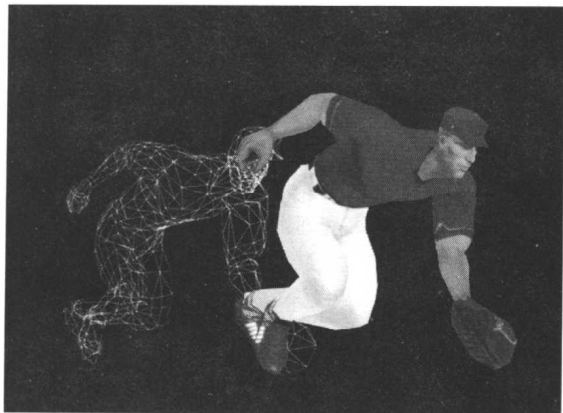


图 E-6 做动作的外野手

特定的动作可以作为游戏中独立的动作或者移动来进行动作捕捉，或者也可以从现有的动作库中提取出来。动作捕捉库为那些不想承担制作定制动作所需花费的开发商提供了以前已经捕捉过的通用动作（比如走或者跑）。当然，并不能确保你能够在任何一个标准库中找到你所需要的每一个动作。

许多动作捕捉设备都提供有标准动作库供使用者购买。这些库在质量和可用性上差异很大。

■ 动作捕捉过程

这里有一些取自于《顶级棒球在线》的动作捕捉过程的照片。

图 E-7 显示的是美国职业棒球大联盟（MLB）的球星 Darrell Evans（站在中间的人）正在指导动作捕捉演员 Derek。头顶上方的三个数字照相机能够反射回系在演员身上的小球反射出来的发光二极管（LED）的光线，用以记录他的动作细节。

能够同一个像 Darrell 这样的球星合作是一个非常明显的有利条件，他可以为游戏开发人员带来前所未有的真正的棒球知识以及对棒球游戏的深入了解。

图 E-8 显示了 Darrell Evans 正在同运动捕捉演员做动作。照片中从左到右分别是 Derek、Darrell 和 Tom。在这幅图片中可以清晰地看到系在演员的表演服装上的反光小球。

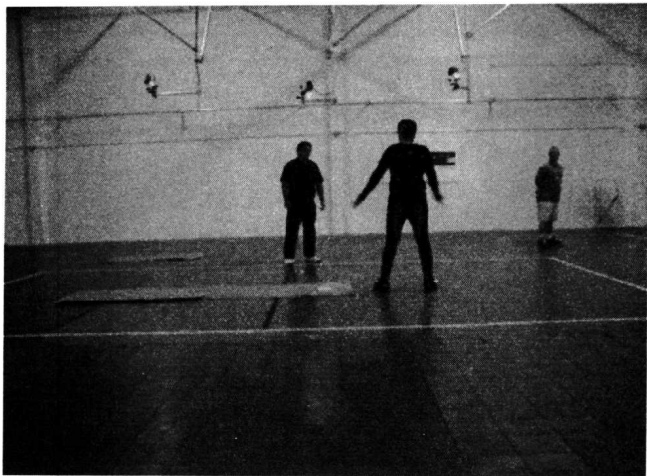


图 E-7 运动捕捉指导

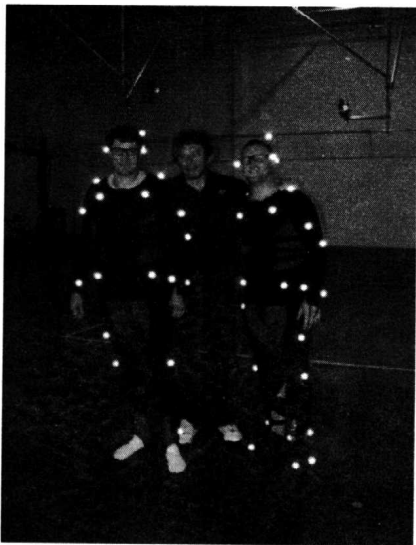


图 E-8 Darrell 和运动捕捉演员们在一起

最后，图 E-9 显示了运动捕捉工作团队正在处理一个运动捕捉过程。大多数游戏开发人员在运动细节方面都依赖于与有经验的运动捕捉工作室和工作团队合作。从这个控

制中心设备，运动捕捉团队能够监视和处理由演员在运动捕捉阶段所产生的所有运动数据。

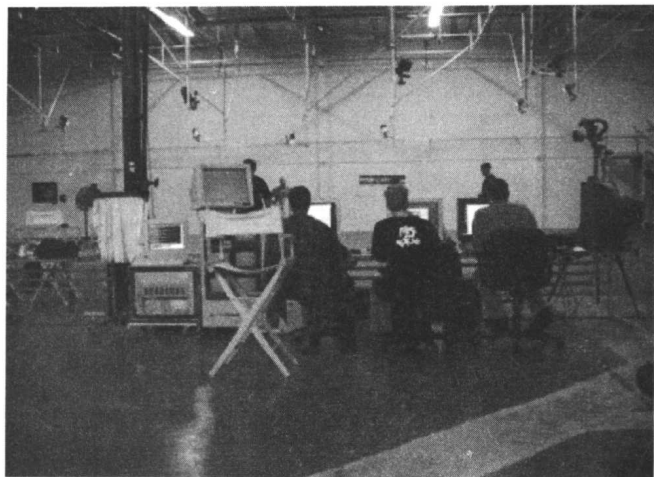


图 E-9 运动捕捉工作组



E.3 棒球运动场

这一节为您介绍一些为《顶级棒球在线》专门定制的完整的新式体育场。如果你在得到美国职业棒球大联盟 (Major League Baseball, MLB) 授权下按照真正的美国职棒大联盟棒球体院馆构造体育场，每一步的制作都要以再现真实的棒球场外观目标，包括最小的细节在内。但是在这里，你有真正的自由来为玩家构造一个造型独特的新棒球场。

图 E-10 展示的是一个从三垒线的角度看过去的一个室外体育场。

图 E-11 是这个体育场从左外野方向看过去的外观。

图 E-12 显示了取自于《顶级棒球在线》的另一个新型室外体育场，它展示了典型棒球场建筑的外观和感觉（注意建筑物的影子以及草地上的磨损的痕迹）。

图 E-13 展示了一个室内的或者有圆顶 / 覆盖的体育场的例子。对于这些体院馆来说，不管是室内的还是室外的，要把所有的建筑细节都为玩家处理得恰到好处常常都是富有挑战性的（灯光、外观和感受、观众、磨损和破损的细节）。如果有一个因素不协调，这些幻想就会破灭。开发者总是尽最大努力避免这一情况的发生。

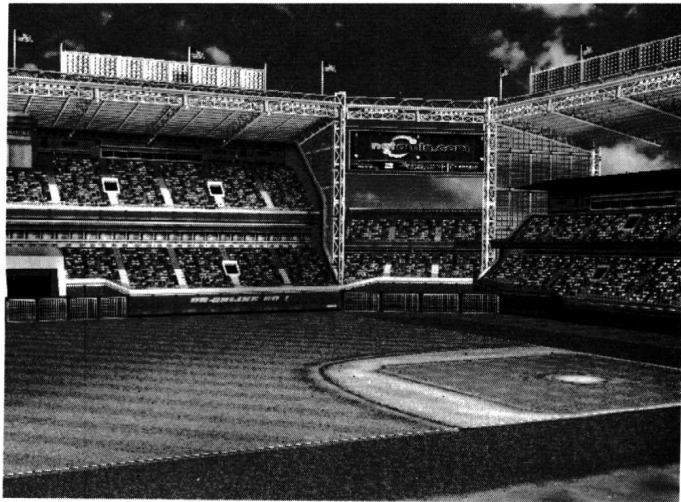


图 E-10 室外体育场

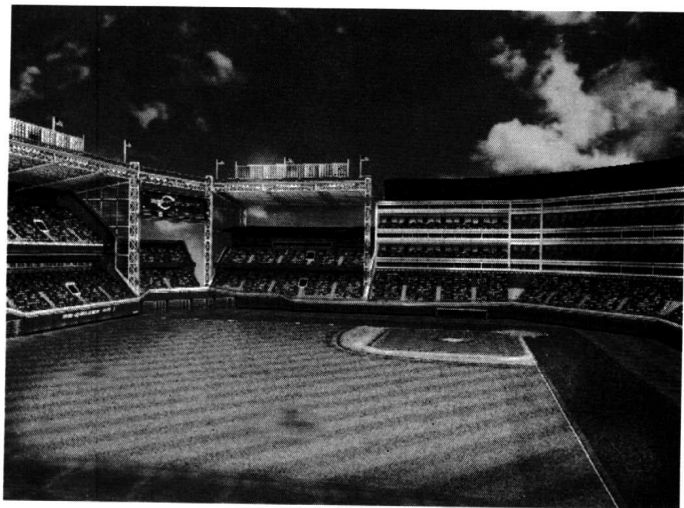


图 E-11 室外体育场（左外野视角）

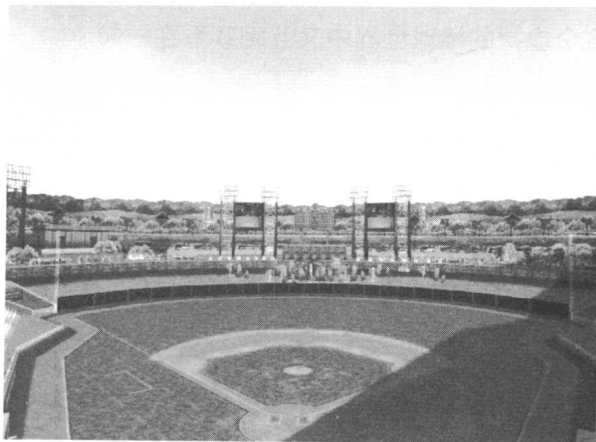


图 E-12 典型的棒球场风格的室外体育场

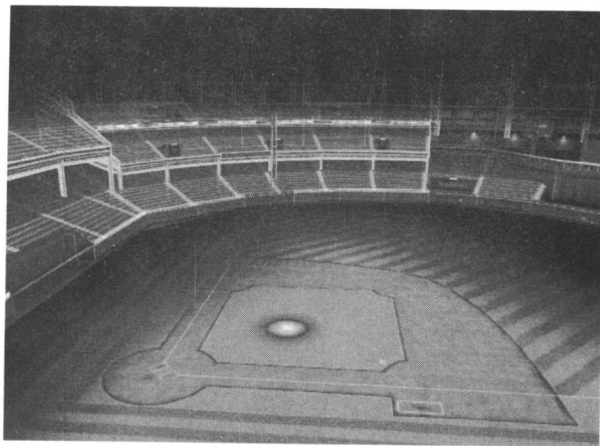


图 E-13 室内体育场



E.4 游戏动作

这里讨论到的游戏中的所有因素都归结为一件事情——游戏动作！在很多情况下，大部分的设计和开发时间都花费在努力保证游戏细节的正确性上。

图 E-14 显示的是游戏动作开发的早期阶段的效果，是从投手的视角看到的场景。

当然,《顶级棒球在线》很有特色的地方是——场上每一个位置都由一个单独的玩家控制,而这些玩家在这个虚拟的棒球世界中又互相联系在一起。

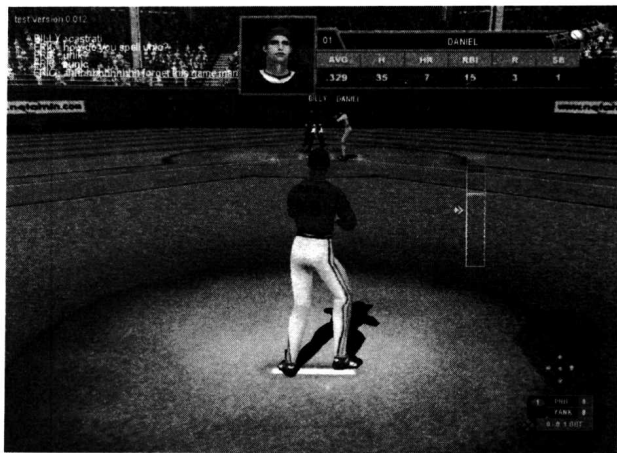


图 E-14 《顶级棒球在线》中投手视角

图 E-15 给出的是从观众视角看的一次击球动作(一次挥棒和一次击空)。谁会不相信大型多玩家在线体育运动游戏前途一片光明!

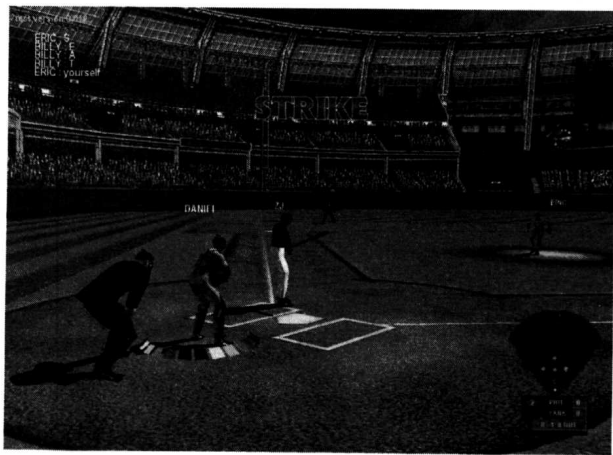


图 E-15 《顶级棒球在线》中观众视角

顶级游戏设计

→ 构造游戏世界

通过应用这本独一无二的著作中所讲述的专家技术，可规划、设计和构造带有栩栩如生的角色和物品的迷人的游戏环境。游戏开发高手Tom Meigs向你讲述了游戏设计的基本知识，包括预视化技术、关卡占位设计和布局、光照、材质映射、行为脚本编程和粒子使用，并且详细介绍了游戏开发和设计的每一个阶段。书中每一章都有对游戏产业内专家的深入采访，以及Tom在Sony PlayStation游戏及其他项目开发中的实战经验的案例研究。此外，本书还提供有关如何进入游戏开发产业，并从事其中各项不同工作的宝贵建议。

- 为你的游戏项目进行一个全面的预视化过程
- 为关卡规划和布局构造形图和参考图形
- 用光照、材质映射、粒子、特效以及音效构造游戏场景
- 布置角色、道具和物品，并调整特定的镜头细节
- 学习各种类型游戏的设计—运动类游戏、格斗类游戏、即时战略类游戏、角色扮演类游戏、第一人称射击类游戏、模拟类游戏、大型多人在线游戏等等
- 用JavaScript、Visual Basic、Python、Perl等工具编写动作事件脚本
- 为掌上电脑和移动电话等无线设备开发和设计游戏
- 实施测试循环，学习如何避免犯错误以及如何为成功做好准备



Tom Meigs是一位独立的游戏顾问，他曾经在The Walt Disney公司、THQ以及其他开发公司担任过游戏设计师和制作人。他参与了数个获奖游戏的开发，包括Madden Football '95、Sports Illustrated Golf、Sea Quest DSV、Akira、Bassmaster's Classic、The Mask、Youngblood，以及许多其他游戏的许多游戏。他还是"Game Developer Magazine"的撰稿人。

ISBN 7-5053-9801-6



9 787505 398016 >



责任编辑：顾慧芳
封面设计：崔云霞

OSBORNE
www.osborne.com

本书贴有激光防伪标志，凡没有防伪标志者，属盗版图书。
ISBN 7-5053-9801-6 定价：35.00元

[General Information]

书名=顶级游戏设计：构造游戏世界

作者=(美)梅格斯(Meigs,T.)著 陈贵敏 杜敬利 韩琪译

页数=360

SS号=11239280

出版日期=2004年05月第1版

出版社=电子工业出版社

封面
书名
版权
前言
目录
目录

第1章 预视化技术

- 1.1 预视化过程的介绍
- 1.2 预视化的逐步实现
- 1.3 功能
- 1.4 关卡场面纸上设计
- 1.5 预视化案例研究评论

MEGA提示

第2章 关卡的规划及制作

- 2.1 规划你的关卡工作
- 2.2 游戏原型设计
- 2.3 关卡占位设计的后处理考虑
- 2.4 关卡实施的案例研究评论

MEGA提示

第3章 灯光、材质、粒子、特效以及音效

- 3.1 灯光
- 3.2 动态灯光
- 3.3 材质
- 3.4 粒子
- 3.5 效果
- 3.6 音效处理
- 3.7 设计工具集的变换
- 3.8 基础元素的案例研究

MEGA提示

第4章 角色、道具、物品以及镜头的细节

- 4.1 布置角色
- 4.2 道具的类型以及用法
- 4.3 物品和威力升级的类型及其布置
- 4.4 镜头的考虑
- 4.5 有关演员加载和镜头调整的评论之案例研究

MEGA提示

第5章 分类游戏设计

- 5.1 运动类游戏
- 5.2 格斗类游戏

- 5.3 解谜类游戏
- 5.4 即时战略游戏
- 5.5 角色扮演类游戏
- 5.6 第一人称和第三人称动作游戏
- 5.7 模拟类游戏
- 5.8 构造电影动画
- 5.9 开发背景故事
- 5.10 设计游戏对话
- 5.11 设计人员的工作工具总结
- 5.12 设计转型的案例研究评论

MEGA提示

第6章 编写动作事件脚本

- 6.1 脚本技术的选择
- 6.2 脚本应用示例
 - 6.3 应用触发器
 - 6.4 构造行为
 - 6.5 引擎实现与Unreal引擎
 - 6.6 有关脚本 / 编辑系统的因素
 - 6.7 棒球游戏脚本评论的案例研究

MEGA提示

第7章 质量保证以及实战测试反馈

- 7.1 质量保证
- 7.2 实战测试反馈
- 7.3 质量保证影响的案例研究

MEGA提示

第8章 大型多人在线游戏的设计考虑

- 8.1 MMOG的制作挑战
- 8.2 MMOG游戏的构造要素与解决方案
- 8.3 MMOG游戏设计因素
- 8.4 MMOG游戏与设计取向
- 8.5 体育爱好者或者体育团体与MMOG游戏
- 8.6 挑战《顶级棒球在线》的案例研究

MEGA提示

第9章 移动电话和无线游戏

- 9.1 日益迫近的繁荣
- 9.2 移动电话上的设计问题
- 9.3 无线游戏初期开发因素的案例研究

MEGA提示

第10章 开始游戏开发

- 10.1 为什么要掌握多种技能
- 10.2 增长的区域与新的机遇
- 10.3 广告游戏
- 10.4 具有现实对照物的网上游戏娱乐
- 10.5 游戏开发前线的趣闻轶事
- 10.6 对设计人员起决定作用的想法的案例研究

MEGA提示

第11章 在游戏开发产业中择业

- 11.1 程序设计
- 11.2 美工
- 11.3 游戏设计
- 11.4 游戏制作
- 11.5 音效
- 11.6 质量保证
- 11.7 创办独立的游戏工作室
- 11.8 游戏工作室的分工
- 11.9 关于测试途径案例研究评论

MEGA提示

附录A 参考信息

附录B 讨论过的工具

附录C 职业指导工作单

附录D 设计人员的快速主题总结

附录E 顶级棒球在线图库