



游戏开发与编程系列

# 游戏架构与设计

(第2版)

GAME  
ARCHITECTURE  
AND DESIGN  
A NEW EDITION

北京希望电子出版社 总策划  
[美] Andrew Rollings, Dave Morris 编 著  
付 煜 庄晓雷 等 译  
爱亿尔(北京)国际游戏开发院 审 校

New  
Riders

红 旗 出 版 社



北京希望电子出版社  
Beijing Hope Electronic Press  
www.bhp.com.cn

NRG



游戏开发与编程系列

# 游戏架构与设计

(第2版)

GAME  
ARCHITECTURE  
AND DESIGN  
A NEW EDITION

北京希望电子出版社 总策划  
[美] Andrew Rollings, Dave Morris 编 著  
付 煜 庄晓雷 等 译  
爱亿尔(北京)国际游戏开发院 审 校

New  
Riders

红 旗 出 版 社



北京希望电子出版社  
Beijing Hope Electronic Press  
www.bhp.com.cn

NRG

## 图书在版编目 (CIP) 数据

游戏架构与设计/(美) 罗林斯 (Rollings, A.), (美) 莫里斯 (Morris, D.) 著; 付煜等译. —北京: 红旗出版社, 2005.3

书名原文: Game Architecture and Design

ISBN 7-5051-1024-1

I. 游... II. ①罗...②莫...③付... III. 游戏—软件设计  
IV. TP311.5

中国版本图书馆 CIP 数据核字 (2004) 第 097321 号

## 内 容 简 介

本书系统、全面地讲述了游戏设计流程的各个方面。全书共分为三个部分和两个附录, 第一部分介绍了游戏设计中的一些基本概念, 如游戏组成、游戏可玩性、游戏平衡、游戏观感等; 第二部分介绍了游戏设计过程中的团队组建和管理, 以及游戏开发的各个步骤; 第三部分讨论了游戏的体系结构, 同时探讨了游戏大师的设计方法学; 附录给出了几个游戏的详细设计文档, 很有实用价值。

本书由浅入深, 讲解透彻, 是一本非常好的游戏开发书籍。本书内容广泛, 适合于游戏设计流程中的各个环节的游戏设计人员, 包括游戏策划师、项目经理、程序员、美工等。本书是游戏设计的经典之作, 是所有游戏工作者的案头必备之书。

## 版 权 声 明

本书英文版名为 Game Architecture and Design, 作者是 Andrew Rollings, Dave Morris, 由 New Riders 出版, 版权归 New Riders 所有。未经出版者书面许可, 本书的任何部分不得以任何形式或手段复制或传播。

|           |                                                                                          |
|-----------|------------------------------------------------------------------------------------------|
| 系 列 名     | 游戏开发与编程系列                                                                                |
| 书 名       | 游戏架构与设计                                                                                  |
| 编 者       | [美]Andrew Rollings Dave Morris                                                           |
| 译 者       | 付煜 庄晓雷等                                                                                  |
| 总 策 划     | 北京希望电子出版社                                                                                |
| 责 任 编 辑   | 但明天 雷铎                                                                                   |
| 出 版 发 行   | 红旗出版社 北京希望电子出版社                                                                          |
| 地 址       | 红旗出版社 北京市沙滩北街 2 号 (100727) 电话 (010) 64037138<br>北京希望电子出版社 北京市海淀区上地 3 街 9 号金隅嘉华大厦 C 座 610 |
| 经 销       | 各地新华书店 软件连锁店                                                                             |
| 排 版       | 希望图书输出中心 姜艳                                                                              |
| 印 刷       | 北京媛明印刷厂                                                                                  |
| 版 次 / 印 次 | 2005 年 3 月第 1 版 2005 年 3 月第 1 次印刷                                                        |
| 开 本 / 印 张 | 787 毫米×1092 毫米 1/16 33.375 印张                                                            |
| 字 数       | 762 千字                                                                                   |
| 印 数       | 0001~5000 册                                                                              |
| 书 号       | ISBN 7-5051-1024-1                                                                       |
| 定 价       | 58.00 元                                                                                  |

# 从游戏中学会生存（总序）

——献给游戏创造者

## 游戏为什么受到重视

今天，越来越多的人不再把游戏看作小儿科，也不仅仅视其为一门艺术了。这大概要归功于全球每天数亿美元的市场业绩，抑或在金融危机中使一个国家经济转危为安的力挽狂澜作用。作为一种生存途径，日本、韩国深得游戏产业振兴经济的非凡要领。无数名不见经传的小人物靠游戏制作成为百万富翁，游戏经典之作的原创者则受到了超级明星般的待遇而被全世界的游戏玩家追捧着。

当游戏被视为艺术时，人们常常拿他与传统艺术形式作比较。认为游戏作为一门娱乐艺术形式，不同于诗歌、小说、戏剧、电影、电视等等传统艺术形式的根本点是“交互”，是“参与”。不错，游戏作为一个产业勃然兴起，确实离不开计算机和互联网。但就其对人类的根本作用而言，要比交互、参与这类技术层面的问题远为原始得多。

## 游戏比艺术起源更早

诗歌、小说、戏剧、电影、电视等等艺术的形成，依赖于语言文字的产生，因此其历史见于壁画与甲骨等人类考古学证据的发现，大约可以上溯到距今 5000 年前。

游戏却不仅仅属于人类。小猫玩线球，人们看到的只是小猫的兴趣盎然和乐此不疲，却不见在这种练习中小猫捕捉能力的提高：通过追逐线球，小猫强壮了自己的利爪，敏捷了全身的器官和肌腱。这种对后代的训练形式普遍存在于动物界，其历史至少可以追溯到对哺乳动物的考古学发现，并非最早的证据距今也已经有 6000 万年了。

## 从生存描述到生存模拟

作为对生存过程的一种描述，诗歌、小说、戏剧、电影、电视等等艺术，是把受体放在媒体之外，通过媒体的艺术感染力，使受体产生共鸣，令其产生沉浸感。

作为对生存过程的一种模拟，游戏则是创造一种虚拟环境，把受体放在媒体之中，使受体自己判断、决策和动作，通过反复操作媒体获得某种能力。数千万年以来，游戏使用着更直接的沉浸手段，让后代在沉浸中得到更实际的生存锻炼。

## 假如游戏不仅以娱乐为目的

由此可见，作为功能的界定，与游戏更接近的社会部类是教育，而不仅仅或不主要是娱乐。无怪乎国际游戏开发者协会（IGDA）活动负责人杰森·德拉·罗卡（Jason Della Rocca）在 2003 CHINAJOY 大会上的主题发言是《“严肃”游戏：游戏对社会经济的潜在影响》。他把“严肃”游戏（Serious Game）定义为“不以娱乐为主要目的游戏”，例举了训练市长的《模拟城市》、训练董事长的《虚拟领导》、训练员工的《直言者》、训练海军陆战队员的《DOOM》等等经典游戏作品。在这一方面，美国确实走在了全世界的前面。早在 1994 年，美国海军陆战队就成立了世界上第一个 Game 军事训练机构；1995 年，美国空军和陆军紧随其后，把游戏作为军队训练的有效辅助手段。总结这些经验，罗卡先生列出了游戏的适用领域依次为：

- 教育
- 国家政策
- 卫生保健
- 企业培训
- 其他

### 游戏可以创造未来

随着计算机的更加移动化、个性化，虚拟社会与现实社会的界限愈加模糊了。如今网游交易市场上的商业链，与现实市场上的买卖规则何其一致！试想，当我们的手机、腰带、手表甚至纽扣全都可以无线上网之后，谁还能够分清现实世界与虚拟世界的截然界限呢？

在虚拟世界与现实世界的直接联系中，虚拟世界的创造者可以把游戏里发展起来的生存新规则，通过游戏渠道传播到更大的范围，影响更多的人。换言之，游戏的创造者可以创造更加理想的未来。这就是：

- 改变工作生活方式
- 影响现有科学技术
- 增强综合国力
- 造福于社会

### 创造者在游戏中永生

回首 12 年前，中国最早的一批拓荒者进入了游戏开发领域。那时他们相信凭借坚定的信念和努力付出就可以确保成功。然而，游戏也是一种商业，他像其他行业一样受到财务、市场和管理等等条件的制约。在软件盗版和决策失误、管理失措的多重夹击下，许多人付出了惨重代价。然而这些付出换来的是生存经验的积累和生存能力的提高。在经过最初的游戏狂热到现在国际游戏市场的供不应求之后，更多的游戏开发从业者学会了在激烈竞争和快速变化的市场中生存。在游戏产业中，似乎并不遵守“胜者王侯败者寇”的法则，美酒的甘甜和血汗的腥涩都可以收入博物馆，正因为他们是历史的拓荒者。

今天，互联网给游戏带来了蓬勃发展的春天。大量资金裹挟着人才和技术进入游戏开发领域。然而，仿效、重复也跟着多起来了，类同题材和近似玩法的游戏产品充斥市场。巨大的风险潜藏随之，二八铁律决定了绝大多数仿效者的时空有限。因此，要想在游戏中获得永生，唯有创造，别无他途。

希望这套出自世界游戏创造高手的丛书能对梦想创造者有所帮助。

杨南征

爱亿尔（北京）国际游戏软件开发院院长  
工商管理博士、作战模拟研究员

## 译者序

如果你是一名游戏爱好者，那么你多半玩过紧张得令人透不过气的《星际争霸》、阴森恐怖的《地下城守护者》或者逼真刺激的《半条命》。享受这些游戏带给你乐趣的同时，难道你不想了解一些关于这些游戏的幕后故事吗？就像观赏一部为之惊叹的电影后，我们多半喜欢了解电影的制作花絮，我相信你也愿意了解在这些出色的游戏背后，天才的程序员为之付出努力的一些细节。

本书不仅仅是一本讲述如何设计游戏的教科书，同时也是一部游戏发展史的见证。从最初的《宇宙入侵者》、《吃金豆》，到后期的《模拟城市》、《上帝也抓狂》，从街机游戏、控制台游戏到 PC 游戏，内容包罗万象。如果你想了解游戏的发展史，那么本书就是你不错的选择。

如果你是一名游戏开发者，那么这本书更是你案头的必备之书。本书没有艰深晦涩的理论，有的只是一个个生动的案例。很多例子都是取自游戏开发公司、甚至是作者本人的亲身经历。阅读这些案例，并将从中汲取的养分付诸于你的开发实践中，你就可以节省大量的开发时间、避免犯下那些会导致项目流产的严重错误，在目前激烈的游戏市场竞争中游刃有余。而如果你是一名游戏开发的项目经理，你将从本书的第二部分获益良多，这部分着重介绍了团队组建和管理中会遇到的各种问题，并提出了合适的解决办法。

暴雪公司、牛蛙公司、ID software、Elite，业界中一个个如雷贯耳的名字，这些公司的一些开发经历都将在本书中多次阐述，希望能对你有所裨益。本书还特别包含了 *The Sims*《模拟人生》的开发者 Will Wriht、*Populous*《上帝也抓狂》和 *Dungeon Keeper*《地下城守护者》的开发者 Peter Molyneux、Firaxis 公司的 Brian Reynolds、*Alpha Centauri*《阿尔法人马行星》的设计师以及 Bill Roper（暴雪公司的设计师）这些游戏业巨擘的采访实录，从中你可一窥天才们的开发理念。

本书共分为三部分和两个附录。

第一部分“游戏设计”共包括 8 章，这部分从最初的游戏创意讲起，内容主要涉及游戏开发的基础知识。第 1 章“初步概念”主要讲解了游戏类型、故事情节、创意以及创意的可行性。第 2 章“核心设计”内容主要涉及逐步建立概念、添加游戏可玩性、开发游戏规格说明以及游戏的原型制作。第 3 章“游戏可玩性”主要讲述了策略的使用以及游戏中的互动性。第 4 章“详细设计”主要讲述了设计文档的作用以及如何利用设计文档。第 5 章“游戏平衡”主要是针对即时战略类游戏中的战斗平衡性因素的阐述。而第 6 章“外观和感觉”强调的是游戏中的界面和画面问题。第 7 章“包装”包含了一个游戏业访谈录，从中你可了解游戏天才的开发历程和理念。第 8 章“游戏设计的未来”则是作者对游戏业未来设计趋势的预测。

第二部分“团队组建和管理”共包括 7 章。内容涉及“团队管理方法”、“角色和部门”、“软件工厂”、“里程碑和截止日期”、“步骤和流程”以及“游戏业的未来”几个方面，这部分面向的读者主要是项目管理人员，项目经理可以从中了解和学习游戏开发中的常见问题以及解决办法。当然，开发人员也能从中获益不浅。

第三部分“游戏体系结构”共包括 9 章。内容涉及“当前的开发方法”、“初步设计”、“技术的使用”、“建立模块”、“初始体系结构设计”、“开发”、“发布前的预备阶段”、“事后剖析”、“游戏开发的未来”几个方面。这部分面向的读者主要是游戏架构师和游戏设计师，其中详细

介绍了重用技术的使用，并对游戏开发的未来发展方向做了大胆的预测。

全书还包含两个附录，附录 A 中包含了几游戏的详细设计文档，附录 B 则是本书写作时的参考书目。

“临川羡鱼，不如退而织网”。如果羡慕天才程序员们的成就，请先汲取营养。希望本书能使你的游戏设计水平更上一层楼。

本书由付煜、庄晓雷主译，参与翻译工作的还有盛海燕、李明、冷风雪、薛赛男、李闻、李双行、李明等，谢小花、杨小梅初排了本书，在此一并感谢！本书篇幅较大，涉及面广，翻译中有些仓促，加之译者水平有限，错误之处在所难免，请读者谅解！

译者

# 前言

## Andrew Rollings 对新版的介绍

我必须承认自己非常惊奇于《游戏架构和设计》这本书的成功，当我们在 1999 年开始有这个打算时，我们大约向十家不同的出版商寄出了建议。只有 Coriolis，更具体地说，是 New Riders Publishing 的 Stephanie Wall 作了回复。我敢打赌其余的出版商正在为此捶胸顿足。

尤其高兴的是，尽管 Coriolis 陷入了麻烦，随之而来的法律问题涉及确保将此书的版权归还给我们，Stephanie Wall 还是接下了这本书，但这次是在 New Riders。我想她和我这样麻烦的作者打了 5 年交道后已经受够了。

这样我们的《游戏架构和设计》第二版就诞生了。事情在这 4 年中发生了改变，尽管我们不太喜欢这样。自 1999 年以来我们已经走了很长一段路，但仍有很长一段路要走。我想最终我们会到达目的地。

我希望这本书的新版能和第一版一样，成为那些有追求的职业游戏开发人员的一本有益的参考书。

愿你享受阅读的乐趣！

— Andrew Rollings

## Dave Morris 对新版的介绍

几年过去后，作者重新修订一本著作经常会有冲动就是稍微改写一点历史。总会有一些令人尴尬的预测困扰着你。而只需轻敲几键，似乎对新一代的读者而言我们就是一贯正确的了，这是个多么诱人的场景啊！

事实上，我们完整保留了在 20 世纪所做的大多数预测。这是因为在很多情况下，例如，中间件的兴起——证实了我们是正确的。坦白地说，像所有其他人一样，我们喜欢这样说：“我们早就这样告诉你了！”

在出错的情况下，我们继续前进并试图从错误中学到东西。某种意义上，这正是我们的设计体系的核心。仅仅有一个方法学并不能永远阻止你犯错误，但它能在很大程度上确保你不会再犯同样的错误。

案例分析都是从我们和同事的共有经历中精选出来的，有人问这些案例研究是否的确是真的。不，事实上不是，因为每个案例真正的实际情况还要糟得多！

但令人鼓舞的是游戏产业正在改变。4 年前，我们建立形式化开发方法学的战斗口号就像旷野中孤独的呼声。现在游戏开发与过去相比，正在成为一个更加结构化的过程。发行商对过程所承担的东西也有了更好的理解。在又一个 4 年，一名偶然看到首版《游戏架构与设计》的开发人员对开发过程如此摇摇欲坠而吃惊。我们想到这点就很高兴，我们为这个产业的进化作出了贡献。

更好的是，随着制作过程得到更好的理解和自动化程度的加强，开发人员消耗的时间减少了。因而释放的额外创造力就可用于游戏内容本身。我们已经看到游戏不再是无声电影的等物，游戏正变得富于深度、和谐和有情感。

托尔斯泰写道：“艺术工作不是手工业，它是将艺术家经历过的情感传达出来的过程。”伟大的艺术不仅仅有娱乐功能——它是有这个功能，但它所能做的要更多。艺术会改变你的生活。在下一个 10 年，我们将看到 *Birth's of a Nation*《国家的诞生》的电视游戏版和 *Citizen Kane*《公民凯恩》的控制台时代版。

为游戏业还处于生命的黎明状态而祝福！

—Dave Morris

## 第一版前言

这本书蕴藏着简单明了但非常正确的哲理：如果计划失败，那就为失败作计划。

当然，游戏是独特的，游戏开发总会产生一些意想不到的问题。与加速的技术竞争危险性非常大。更糟糕的是，有时客户会在项目进行到中途修正其需求。

但这些不能成为抛弃事先计划的理由，当变更不可避免时，良好的设计会为你提供一个要达到的目标。很多时候，设计至少可以预见到未来变化的领域。一个完整的项目计划将建立一个框架应对变更情况。开发阶段的开始并不意味着计划过程的结束。更恰当的看法是，在任何必要的变更情况发生时仍然要维持计划，这样，尽管目标可能变动，但你也永远不会在你正前往何处这点上迷失方向。

不错，游戏开发是很特别，正因为如此，它才需要特别方式的计划，这就是本书中展示的方法学。为阐明这些观点，我们提供了丰富的案例研究。这些案例研究都建立在这个行业的共同环境之下，但都是虚构的——任何与实际存在的公司或产品名称的类似均非有意造成，除非是明确阐述的。[另外，我们还参考了几个注册过的游戏，对注册过的典型游戏，如 *Pac-Man*（由 *Namco* 拥有）则重复提及多次。包含这些参考文献是为了增强指导意义以及更直观地传达本书的游戏观念，本书中组织的这些参考文献绝非对这些注册产品发起挑战。

哪些人应该读这本书呢？最简单的答案是那些与游戏开发相关或打算涉足这个领域的人。团队中所有人都会因了解到其他人的职责领域而受益匪浅。书中每一章节强调的问题都是特定于开发团队的某一部分。我们推荐你首先阅读与自己主攻方向联系最紧密的章节。

## 第一部分 游戏设计

本书将纯粹的游戏设计从体系结构和形式化设计分离出来，游戏设计包含了最终产品的特征描述，可用这些特征描述作为整个团队的共享创造性思维的产物。随着开发的进展以及不可避免的变更出现，设计师的任务就转为逐渐演化设计，使得项目的意图保持清晰。

我们不认同那种游戏可玩性是完全无法预知、因而是无法设计的说法。在这个问题上，为了在连续迭代开发的基础上得到一个精确的开发模型，那些由游戏理论得到的技巧和数学知识以及富于想象力的作品都可以在设计过程中使用。

很明显，如果设计师对游戏可玩性的基本问题能更好地理解，就可以将很多游戏包含的可玩性智慧的火花合成一团火焰。第一部分向你展示了如何在自己的设计中达到这个目标。

## 第二部分 团队组建和管理

本书提倡采用形式化过程，因为我们发现这样做很有效果。很多开发人员对采用形式化开发的态度非常谨慎，因为他们担心这样会导致官僚作风和过度管理。事实上，采用形式化方法

得到的正好与此相反。

考虑这样一个类比。在日本的剑道中，事先定义好一定顺序的动作，训练的很多重点放在对这些动作的重复上，这称为 *kata*。学员可能会奇怪形式化的一系列对头部、胸部和腹部的打击怎么可能在实际生活中运用，因为真实生活中几乎有无限多的动作组合。但这样训练之后的一天，有人持剑站在你面前，你的胳膊就会立即摆出防卫姿态。这个动作对你而言简直无需思考。

类似地，我们赞成采用形式化过程正好是因为它可以减轻管理负担。只不过这里的重点放在了个人作为团队一部分的技术的磨炼，开发人员本人负责对自身的管理。第二部分详细阐述了简洁、通用意义上的步骤，这些步骤易于采用并很快成为开发人员的第二本能。这样做的好处是提高了效率、可靠性以及团队士气。

### 第三部分 游戏体系结构

项目的体系结构计划的关键在于将概念性因素、艺术性因素或游戏可玩性因素与技术需求联系在一起。将设计设想为游戏所要达到的一种完美形式。体系结构就是制订将现实状态转化为完美形式的策略。

一个完美的体系结构应试图达到如下目标：

- **模块性** 将项目分成完全封装的模块的好处是每个模块都可独立地测试、修改或被替代而不会影响系统的其余部分。
- **重用性** 每次都重新发明轮子毫无意义。应将模块设计成可扩展、可移植的，这样就可将它们插入到其他项目中。
- **健壮性** 只有建立的体系结构解放了模块间的相互依赖才能获得高度的可靠性，设计的终极目标是体系结构中可以建立这样的游戏：它们可在各种预想不到的环境下运行——简而言之，它们可经受崩溃考验。尽管这样的目标很大程度上已经超出了本书的范畴，我们还是使用面向对象的设计模式介绍性地论述了这个主题。
- **可跟踪性** 项目计划是直接从体系结构集成而来的，它产生的进度表允许对项目过程进行实际度量。

项目失败是由于拙劣的体系结构设计未能帮助设计出想要的游戏，或者，更糟的是，体系结构以一种不适合的形式限制了开发。

我们展示了设计如何逐渐地转换为体系结构，而体系结构又如何转换为技术设计和文档，后者又如何转换为代码——这个有内聚性的连续过程一直朝着游戏设计所描述的（演化）场景前进。

### 第四部分 附录

这里我们使用现实中的项目作为案例来阐明本书中使用的技巧，我们展示了如何使用一个所期待的最终产品的特征化描述——我们称之为游戏可玩性设计的设想文档来开始一个项目，当这个设计映射到游戏环境的逻辑抽象中，你就得到了软件计划，其中描述了游戏的分立模块以及它们如何相互作用。最后，还增加了实施细节以产生项目的技术设计。然后编码阶段开始。

# 目 录

|                                       |    |                            |    |
|---------------------------------------|----|----------------------------|----|
| 第一部分 游戏设计.....                        | 1  | 2.4.1 特色.....              | 26 |
| 第1章 初步概念.....                         | 3  | 案例分析 2.4 一个浮现的实例.....      | 27 |
| 1.1 新技术的震撼.....                       | 3  | 2.4.2 游戏可玩性.....           | 28 |
| 1.2 创造性指导方针.....                      | 4  | 2.4.3 界面.....              | 29 |
| 1.3 形成创意.....                         | 5  | 案例分析 2.5 优美界面.....         | 29 |
| 1.3.1 灵感.....                         | 5  | 2.4.4 规则.....              | 29 |
| 1.3.2 合成.....                         | 6  | 案例分析 2.6 规则必须为特色服务.....    | 30 |
| 1.3.3 共鸣.....                         | 7  | 2.4.5 关卡设计.....            | 30 |
| 1.3.4 汇聚.....                         | 7  | 案例分析 2.7 有趣的关卡设计.....      | 31 |
| 1.4 修正创意.....                         | 8  | 2.5 游戏规格说明举例.....          | 32 |
| 戏剧效果.....                             | 8  | 案例分析 2.8 游戏说明.....         | 32 |
| 1.5 论述.....                           | 10 | 2.5.1 原型的价值.....           | 34 |
| 1.6 审查.....                           | 11 | 2.5.2 文档的必要性.....          | 35 |
| 1.6.1 分析.....                         | 11 | 第3章 游戏可玩性.....             | 36 |
| 1.6.2 评估.....                         | 11 | 3.1 游戏可玩性.....             | 37 |
| 1.6.3 论证.....                         | 11 | 3.1.1 游戏可玩性的实现.....        | 37 |
| 案例分析 1.1 一页的展示.....                   | 12 | 3.1.2 支配性策略问题.....         | 38 |
| 1.7 可行性.....                          | 13 | 3.1.3 准支配.....             | 38 |
| 1.7.1 商业因素.....                       | 13 | 案例分析 3.1 环境+规则=游戏的可玩性..... | 41 |
| 1.7.2 技术因素.....                       | 13 | 3.1.4 投资支持.....            | 43 |
| 1.7.3 开发因素.....                       | 13 | 3.1.5 多功能性.....            | 43 |
| 1.8 写下你的想法.....                       | 14 | 案例分析 3.2 出乎意料的多功能性.....    | 44 |
| 案例分析 1.2 <i>Conquerors</i> 的初始论述..... | 14 | 3.1.6 补偿因素.....            | 45 |
| 第2章 核心设计.....                         | 22 | 案例分析 3.3 平衡补偿因素.....       | 45 |
| 2.1 游戏是什么.....                        | 22 | 3.1.7 非永久性.....            | 46 |
| 2.1.1 酷的特色.....                       | 22 | 3.1.8 影子成本.....            | 46 |
| 2.1.2 奇特的图形.....                      | 22 | 案例分析 3.4 《帝国时代》的影子成本.....  | 46 |
| 2.1.3 谜题.....                         | 23 | 3.1.9 协作.....              | 47 |
| 2.1.4 背景和故事.....                      | 23 | 3.1.10 游戏可玩性的最后评论.....     | 47 |
| 2.2 游戏不是一切.....                       | 23 | 3.2 互动.....                | 48 |
| 案例分析 2.1 故事与可玩性.....                  | 24 | 3.2.1 互动的种类.....           | 48 |
| 2.3 游戏意味着游戏可玩性.....                   | 24 | 案例分析 3.5 不同种类的互动.....      | 49 |
| 案例分析 2.2 一个错过的机会?.....                | 24 | 3.2.2 为什么,与什么?.....        | 50 |
| 2.4 创建游戏规格说明.....                     | 26 | 第4章 详细设计.....              | 52 |
| 案例分析 2.3 集成游戏目标.....                  | 26 | 4.1 设计师的作用.....            | 52 |

|                                            |    |
|--------------------------------------------|----|
| 案例分析 4.1 开发时限.....                         | 52 |
| 4.2 设计文档.....                              | 55 |
| 4.2.1 游戏可玩性规格说明.....                       | 55 |
| 4.2.2 设计师的备注.....                          | 55 |
| 案例分析 4.2 针对规格说明编制文档.....                   | 55 |
| 4.3 使用设计文档.....                            | 56 |
| 4.4 使设计适应开发.....                           | 58 |
| 层和试验台.....                                 | 58 |
| 案例分析 4.3 计划详细规格说明以适应<br>体系结构.....          | 60 |
| 4.5 为什么要使用文档.....                          | 61 |
| <b>第 5 章 游戏平衡</b> .....                    | 62 |
| 5.1 玩家/玩家平衡.....                           | 62 |
| 对称.....                                    | 63 |
| 5.2 玩家/游戏可玩性平衡.....                        | 65 |
| 案例分析 5.1 这样有趣吗?.....                       | 65 |
| 5.2.1 奖赏玩家.....                            | 66 |
| 5.2.2 让机器来做这件事情.....                       | 67 |
| 5.2.3 创作可玩性游戏.....                         | 67 |
| 案例分析 5.2 保存游戏的问题.....                      | 67 |
| 5.3 游戏可玩性/游戏可玩性平衡.....                     | 68 |
| 5.3.1 组件和特性平衡.....                         | 69 |
| 案例分析 5.3 Dungeon Keeper 中的组件和<br>特性平衡..... | 70 |
| 5.3.2 非传递游戏力学保证平衡.....                     | 71 |
| 案例分析 5.4 使用 SPS 的特性平衡.....                 | 74 |
| 案例分析 5.5 使用游戏理论分析来<br>实现平衡.....            | 78 |
| 5.4 游戏平衡性核对清单.....                         | 82 |
| <b>第 6 章 外观和感觉</b> .....                   | 83 |
| 6.1 氛围.....                                | 83 |
| 6.1.1 声音.....                              | 84 |
| 案例分析 6.1 最佳音效.....                         | 84 |
| 6.1.2 视觉.....                              | 85 |
| 案例分析 6.2 一个荒腔走板的摘记.....                    | 86 |
| 6.1.3 触觉.....                              | 86 |
| 6.2 界面.....                                | 87 |
| 案例分析 6.3 把界面、外观和感觉结合<br>在一起.....           | 87 |

|                            |     |
|----------------------------|-----|
| 案例分析 6.4 有时少就是不好.....      | 88  |
| 6.3 故事讲述.....              | 89  |
| 6.3.1 故事讲述技术工具箱.....       | 90  |
| 案例分析 6.5 外观和感觉文档的例子.....   | 92  |
| 6.3.2 转折点.....             | 95  |
| 案例分析 6.6 预料之外的情节发展.....    | 95  |
| 6.3.3 悬念.....              | 97  |
| 6.3.4 对话.....              | 97  |
| 6.3.5 主题.....              | 98  |
| 6.3.6 结局.....              | 98  |
| 案例分析 6.7 不满意的结局.....       | 99  |
| 6.3.7 变化.....              | 99  |
| 6.4 结束语.....               | 100 |
| <b>第 7 章 包装</b> .....      | 101 |
| 专家.....                    | 101 |
| 7.1.1 游戏概念.....            | 102 |
| 7.1.2 为变化做计划.....          | 103 |
| 7.1.3 技术.....              | 106 |
| 7.1.4 开发.....              | 107 |
| 7.1.5 团队.....              | 109 |
| 7.1.6 成本和时间期限.....         | 110 |
| 7.1.7 游戏可玩性.....           | 111 |
| 7.1.8 未来展望.....            | 113 |
| <b>第 8 章 游戏设计的未来</b> ..... | 115 |
| 8.1 设计的必要性.....            | 115 |
| 8.1.1 别害怕作计划.....          | 115 |
| 案例分析 8.1 设计节省了时间.....      | 116 |
| 8.1.2 为什么进行设计是很好的行为.....   | 117 |
| 案例分析 8.2 及时更新设计.....       | 118 |
| 8.2 游戏设计的本质.....           | 118 |
| 8.2.1 它是原创吗?.....          | 119 |
| 8.2.2 它有一致性吗?.....         | 119 |
| 8.2.3 它是互动的吗?.....         | 119 |
| 8.2.4 它吸引人吗?.....          | 120 |
| 8.2.5 它富于趣味吗?.....         | 120 |
| 8.3 设计的未来.....             | 121 |
| 8.3.1 让设计更具共性.....         | 121 |
| 8.3.2 非象征性的设计.....         | 122 |
| 案例分析 8.3 比较象征性和非象征性        |     |

|                              |     |                              |     |
|------------------------------|-----|------------------------------|-----|
| 的设计 .....                    | 123 | 案例分析 11.4 有效处理问题的方法.....     | 162 |
| 8.4 游戏的未来 .....              | 123 | 案例分析 11.5 工具重用的好处.....       | 164 |
| 8.4.1 下一个 10 年.....          | 124 | 11.4 实施软件工厂结构和方法 .....       | 166 |
| 8.4.2 软件的强项.....             | 125 | 11.4.1 开始.....               | 166 |
| 8.4.3 创意的十字路口.....           | 126 | 11.4.2 知道什么时候使用哪个团队          |     |
| 案例分析 8.4 舞台剧的范例.....         | 128 | ——并行开发时间表 .....              | 167 |
| 8.5 游戏如娱乐 .....              | 130 | 11.4.3 轮换、重新分配组员.....        | 168 |
| 8.6 前进之路 .....               | 131 | 案例分析 11.6 必不可少的注释.....       | 168 |
| <b>第二部分 团队组建和管理</b> .....    | 133 | 11.5 软件工厂的适用性 .....          | 169 |
| <b>第 9 章 当前的团队管理方式</b> ..... | 135 | 11.6 小一些的团队 .....            | 169 |
| 当前的开发模式 .....                | 135 | 11.7 结束语 .....               | 169 |
| 9.1.1 游戏业的起源.....            | 135 | <b>第 12 章 里程碑和截止日期</b> ..... | 170 |
| 9.1.2 游戏开发者的难题.....          | 137 | 12.1 现在的里程碑是如何工作的 .....      | 170 |
| 9.1.3 存在问题的开发者.....          | 138 | 案例分析 12.1 模糊的里程碑会给           |     |
| 9.1.4 加班时间过长意味项目的失败.....     | 142 | 项目带来什么影响? .....              | 172 |
| 9.1.5 规则的例外.....             | 143 | 12.2 模糊的里程碑 .....            | 173 |
| 案例分析 9.1 《雷神之锤》、《星际          |     | 12.3 里程碑和小里程碑 .....          | 173 |
| 争霸》与《幽浮——突击者》 .....          | 143 | 12.4 什么时候使用里程碑 .....         | 174 |
| <b>第 10 章 角色和部门</b> .....    | 145 | 12.5 制定精确的里程碑 .....          | 174 |
| 10.1 分派人员 .....              | 145 | 案例分析 12.2 取消项目的花费.....       | 175 |
| 10.1.1 管理和设计部门.....          | 146 | 检查点 1.0 收集基本需求.....          | 176 |
| 10.1.2 编程部门.....             | 147 | 检查点 1.1 收集技术需求.....          | 177 |
| 10.1.3 美工部门.....             | 147 | 检查点 1.2 收集资源需求.....          | 177 |
| 10.1.4 音乐和杂项部门.....          | 148 | 检查点 2.0 基本可行性研究.....         | 178 |
| 10.1.5 支持和质保部门.....          | 149 | 检查点 2.1 技术可行性研究.....         | 179 |
| 10.2 提高士气和工作环境 .....         | 150 | 检查点 2.2 资源可行性研究.....         | 180 |
| 10.2.1 提高士气和改善工作环境.....      | 150 | 检查点 3.0 起草体系结构规范.....        | 180 |
| 10.2.2 士气高涨的告诫和警告.....       | 153 | 检查点 3.1 项目初始化.....           | 180 |
| 10.3 分散风险 .....              | 154 | 12.6 定义里程碑 .....             | 181 |
| <b>第 11 章 软件工厂</b> .....     | 155 | 12.6.1 差的里程碑.....            | 182 |
| 11.1 关于软件工厂 .....            | 155 | 12.6.2 好的里程碑.....            | 184 |
| 11.2 为什么使用软件工厂 .....         | 156 | 案例分析 12.3 一个发生在现实生活中的        |     |
| 解决游戏开发问题.....                | 156 | 里程碑.....                     | 185 |
| 案例分析 11.1 失去核心成员的影响.....     | 157 | 12.6.3 研究以及截止日期.....         | 186 |
| 案例分析 11.2 代码重用.....          | 158 | 12.6.4 里程碑的评价.....           | 186 |
| 11.3 组织软件工厂 .....            | 159 | <b>第 13 章 步骤和流程</b> .....    | 187 |
| 11.3.1 软件工厂的结构.....          | 159 | 13.1 步骤.....                 | 187 |
| 11.3.2 小组责任和沟通.....          | 160 | 13.1.1 检查.....               | 188 |
| 案例分析 11.3 没有有效地处理问题.....     | 161 | 13.1.2 总体测试.....             | 190 |

|                          |     |                                    |     |
|--------------------------|-----|------------------------------------|-----|
| 13.2 流程.....             | 194 | 复用能力.....                          | 255 |
| 案例分析 13.1 愚蠢的流程.....     | 197 | 第 17 章 初步设计.....                   | 257 |
| 13.3 步骤: 什么时候使用.....     | 198 | 17.1 开始.....                       | 258 |
| 13.3.1 设计阶段.....         | 199 | 案例分析 17.1 《雷神之锤 II》中的<br>抽象.....   | 259 |
| 13.3.2 开发阶段.....         | 201 | 17.2 硬件抽象.....                     | 260 |
| 13.3.3 测试阶段.....         | 202 | 17.2.1 图形硬件抽象.....                 | 260 |
| 13.4 源控制和代码检查: 协同.....   | 202 | 17.2.2 声音硬件抽象.....                 | 263 |
| 案例分析 13.2 不需要源控制.....    | 203 | 17.2.3 其他硬件问题.....                 | 264 |
| 源控制用来干什么?.....           | 204 | 17.2.4 “不构造在这里”可能更好.....           | 268 |
| 13.5 信息传递的重要性.....       | 204 | 17.2.5 模糊地带.....                   | 269 |
| 前摄的信息传递和被动的信息传递.....     | 206 | 17.3 问题域.....                      | 270 |
| 第 14 章 疑难解答.....         | 208 | 什么是游戏? (回顾).....                   | 270 |
| 风险.....                  | 210 | 17.4 以 token 的方式考虑.....            | 271 |
| 14.1.1 设计和体系结构问题.....    | 212 | 17.4.1 Pong 游戏的 token 化.....       | 272 |
| 案例分析 14.1 不听劝告的管理人员..... | 214 | 17.4.2 Pac-Man 的 token 化.....      | 277 |
| 14.1.2 进度表威胁.....        | 218 | 17.4.3 状态转换及属性.....                | 282 |
| 案例分析 14.2 重新调整进度表.....   | 221 | 案例分析 17.2 缺乏灵活性产生的陷阱.....          | 283 |
| 14.1.3 组织机构的问题.....      | 222 | 第 18 章 技术的使用.....                  | 288 |
| 14.1.4 合同问题.....         | 223 | 18.1 美工的状态.....                    | 290 |
| 14.1.5 人员问题.....         | 224 | 18.1.1 3D 引擎的起落.....               | 290 |
| 14.1.6 开发问题.....         | 225 | 18.1.2 技术的理解.....                  | 295 |
| 14.1.7 过程问题.....         | 227 | 案例分析 18.1 第一印象.....                | 295 |
| 第 15 章 游戏业的未来.....       | 229 | 18.2 不保险的研究.....                   | 299 |
| 15.1 游戏业现状.....          | 229 | 18.2.1 研究的类型.....                  | 300 |
| 15.1.1 第一时期.....         | 229 | 案例分析 18.2 忽略了可能存在的<br>机会.....      | 301 |
| 15.1.2 第二时期.....         | 230 | 案例分析 18.3 俄罗斯方块: 一个告诫.....         | 304 |
| 15.1.3 第三时期.....         | 230 | 案例分析 18.4 《时空英豪》: 技术的<br>很好运用..... | 305 |
| 15.1.4 游戏中的暴力.....       | 232 | 18.2.2 做好工作日志.....                 | 305 |
| 15.2 新模块开发者.....         | 235 | 18.3 重新发明轮子.....                   | 306 |
| 案例分析 15.1 对开发者来说很艰难..... | 236 | 18.4 使用对象技术.....                   | 307 |
| 15.3 在线革命.....           | 238 | 抽象方法的利弊.....                       | 310 |
| 15.3.1 在线交付游戏.....       | 238 | 第 19 章 建立模块.....                   | 312 |
| 15.3.2 在线玩游戏.....        | 238 | 软件中的代码重用.....                      | 313 |
| 第三部分 游戏体系结构.....         | 241 | 19.1.1 代码重用.....                   | 313 |
| 第 16 章 当前的开发方法.....      | 243 | 案例分析 19.1 引擎重用.....                | 313 |
| 16.1 游戏开发技术的历史.....      | 244 | 19.1.2 设计重用: 模式.....               | 314 |
| 16.1.1 最初的游戏创意的兴衰.....   | 245 |                                    |     |
| 16.1.2 开发环境.....         | 248 |                                    |     |
| 16.2 现在的情况.....          | 254 |                                    |     |

|                               |            |                              |            |
|-------------------------------|------------|------------------------------|------------|
| 19.1.3 特定于游戏的模式.....          | 347        | 22.1 后期评估.....               | 397        |
| <b>第 20 章 初始体系结构设计.....</b>   | <b>348</b> | 22.1.1 最终分析.....             | 397        |
| 20.1 体系结构的诞生.....             | 348        | 22.1.2 游戏达到标准了吗?.....        | 398        |
| 体系结构的概念.....                  | 349        | 案例分析 22.1 一起自食其果的灾难.....     | 398        |
| 20.2 层式系统.....                | 353        | 案例分析 22.2 一份恢复计划.....        | 400        |
| 20.2.1 0 层: 原型.....           | 354        | 案例分析 22.3 许可协议地狱.....        | 403        |
| 案例分析 20.1 一个数据库驱动的方法..        | 357        | 案例分析 22.4 最后的疯狂.....         | 403        |
| 20.2.2 一层及其上各层.....           | 357        | 22.2 后期的本地化.....             | 404        |
| 20.3 体系结构设计.....              | 360        | 22.2.1 许可协议.....             | 405        |
| 20.3.1 将基于层的方法用于架构设计....      | 362        | 22.2.2 语言.....               | 405        |
| 案例分析 20.2 讨论 Warbots 的架构..... | 363        | 22.2.3 Demos.....            | 406        |
| 20.3.2 体系结构的正交性.....          | 364        | 案例分析 22.5 把游戏拱手交出去.....      | 406        |
| <b>第 21 章 开发.....</b>         | <b>365</b> | 案例分析 22.6 留一手.....           | 407        |
| 21.1 开发过程.....                | 365        | 22.3 玩法测试.....               | 407        |
| 21.2 代码质量.....                | 367        | 案例分析 22.7 他们怎么会没有            |            |
| 编码标准.....                     | 367        | 发现呢? ! .....                 | 408        |
| 21.3 编码优先级.....               | 385        | 22.4 焦点小组.....               | 409        |
| 21.3.1 速度.....                | 385        | 22.5 网站.....                 | 410        |
| 21.3.2 规模.....                | 386        | 22.6 为 Gold Master 准备就绪..... | 411        |
| 21.3.3 灵活性.....               | 386        | 22.7 补丁.....                 | 411        |
| 21.3.4 可移植性.....              | 387        | <b>第 23 章 事后剖析.....</b>      | <b>414</b> |
| 21.3.5 可维护性.....              | 387        | 案例分析 23.1 两个项目的故事.....       | 415        |
| 21.4 调试和模块完成.....             | 387        | 23.1 团队动力.....               | 417        |
| Bug 的类型.....                  | 388        | 案例分析 23.2 所有可怕的错误.....       | 417        |
| 案例分析 21.1 是 A 类 bug 吗? .....  | 389        | 23.2 概念.....                 | 419        |
| 21.5 7 个黄金法则.....             | 392        | 23.2.1 趋势.....               | 420        |
| 21.5.1 重用.....                | 392        | 案例分析 23.3 对趋势的错误判断.....      | 420        |
| 案例分析 21.2 可重用的体系结构.....       | 393        | 23.2.2 亲和力.....              | 422        |
| 21.5.2 文档.....                | 393        | 23.3 开发阶段.....               | 423        |
| 21.5.3 先设计.....               | 393        | 23.3.1 软件计划.....             | 424        |
| 21.5.4 进度表.....               | 393        | 案例分析 23.4 地下密牢.....          | 424        |
| 21.5.5 在项目进行过程中捕捉错误.....      | 394        | 23.3.2 编码.....               | 425        |
| 21.5.6 控制研发程度 (R&D) .....     | 394        | 23.3.3 测试.....               | 426        |
| 21.5.7 知道什么时候划定最后界限.....      | 394        | 23.4 商业方面.....               | 426        |
| 21.6 3 种毫无作用的行为.....          | 394        | 案例分析 23.5 保证你的资金流.....       | 426        |
| 21.6.1 糟糕的管理.....             | 395        | 23.5 事后剖析本身的事后剖析.....        | 427        |
| 21.6.2 特性蔓延.....              | 395        | <b>第 24 章 游戏开发的未来.....</b>   | <b>428</b> |
| 21.6.3 编码人员的僵化.....           | 395        | 24.1 基于背景开发.....             | 428        |
| <b>第 22 章 发布前的预备阶段.....</b>   | <b>396</b> | 24.2 未来开发.....               | 431        |

|                            |     |
|----------------------------|-----|
| 24.2.1 市场营销.....           | 431 |
| 案例分析 24.1 市场意味着定位.....     | 432 |
| 24.2.2 内容.....             | 433 |
| 案例分析 24.2 无战略的开发.....      | 433 |
| 24.2.3 计划.....             | 435 |
| 24.2.4 开发人员.....           | 436 |
| 24.3 虽小仍美.....             | 437 |
| 24.4 未来团队的建立.....          | 437 |
| 24.4.1 角色.....             | 437 |
| 24.4.2 动机.....             | 439 |
| 24.4.3 士气.....             | 440 |
| 24.5 开发中的新方向.....          | 441 |
| 24.5.1 整体方法.....           | 441 |
| 24.5.2 “侏罗纪公园”软件.....      | 443 |
| 24.5.3 固有的及超然的世界.....      | 444 |
| 24.6 未来事物的定形.....          | 446 |
| <b>第四部分 附录</b> .....       | 449 |
| <b>附录 A 游戏设计文档样例</b> ..... | 451 |
| A.1 详细设计讨论.....            | 451 |
| A.1.1 弹珠游戏简介.....          | 451 |
| A.1.2 游戏玩法概述.....          | 451 |
| A.1.3 平台.....              | 452 |
| A.1.4 时间比例.....            | 453 |
| A.1.5 为什么迷宫类游戏今不如昔? .....  | 453 |
| A.1.6 迷宫类游戏魅力.....         | 454 |
| A.1.7 为什么弹珠游戏如此出色? .....   | 454 |
| A.1.8 游戏设计: 用户界面元素.....    | 456 |
| A.1.9 弹珠的物理学.....          | 459 |
| A.1.10 方块.....             | 464 |
| A.1.11 特殊例子——方块之间的碰撞... .. | 466 |
| A.1.12 玩游戏.....            | 468 |

|                                       |     |
|---------------------------------------|-----|
| A.1.13 更多的修饰.....                     | 469 |
| A.2 初步设计和样例设计.....                    | 472 |
| A.3 芝加哥黑帮: 在 20 年代繁荣的美国<br>地下社会 ..... | 472 |
| A.3.1 概述.....                         | 472 |
| A.3.2 游戏目标.....                       | 473 |
| A.3.3 图形.....                         | 474 |
| A.3.4 进入游戏.....                       | 476 |
| A.3.5 角色类型.....                       | 477 |
| A.3.6 角色个性.....                       | 480 |
| A.3.7 命令.....                         | 481 |
| A.3.8 战斗.....                         | 482 |
| A.3.9 游戏世界.....                       | 483 |
| A.3.10 营业场所.....                      | 485 |
| A.3.11 消息.....                        | 488 |
| A.3.12 战役指南.....                      | 488 |
| A.3.13 目标平台.....                      | 490 |
| A.3.14 营救者.....                       | 490 |
| A.3.15 游戏元素.....                      | 491 |
| A.3.16 游戏怎么玩? .....                   | 495 |
| A.4 技术规约.....                         | 497 |
| 技术规约: 弹珠游戏完全 3D 置入式<br>图形模块 .....     | 497 |
| A.5 代码评审表.....                        | 512 |
| A.6 测试脚本.....                         | 513 |
| <b>附录 B 参考文献</b> .....                | 514 |
| B.1 参考书目 .....                        | 514 |
| B.2 期刊 .....                          | 516 |
| B.3 新闻组 .....                         | 516 |
| <b>术语表</b> .....                      | 517 |



# 第一部分 游戏设计

- 第1章 初步概念
- 第2章 核心设计
- 第3章 游戏可玩性
- 第4章 详细设计
- 第5章 游戏平衡
- 第6章 外观和感觉
- 第7章 包 装
- 第8章 游戏设计的未来