

SAMS



game over

BEGINNING

GAME
PROGRAMMING

游戏编程入门

[美] Michael Morrison 著
徐刚 于健 薛雷 译

开发 7 个完整的游戏

- *Brainiac*
- *Light Cycles*
- *Henway*
- *Battle Office*
- *Meteor Defense*
- *Space Out*
- *Stunt Jumper*

人民邮电出版社
POSTS & TELECOM PRESS

游戏编程入门

[美] Michael Morrison 著

徐刚 于健 薛雷 译

人民邮电出版社

图书在版编目 (CIP) 数据

游戏编程入门/(美)莫里森(Morrison.M.)著;徐刚,于健,薛雷译.

—北京:人民邮电出版社,2005.9

ISBN 7-115-13660-2

I. 游... II. ①莫...②徐...③于...④薛... III. 游戏—软件设计 IV. TP311.5

中国版本图书馆 CIP 数据核字 (2005) 第 095156 号

版 权 声 明

Michael Morrison: Beginning Game Programming (ISBN: 0672326590)

Copyright © 2005 by Sams Publishing

All rights reserved.

本书中文简体字版由美国 **Sams Publishing** 公司授权人民邮电出版社出版。未经出版者书面许可,对本书的任何部分不得以任何方式复制或抄袭。

版权所有,侵权必究。

游戏编程入门

- ◆ 著 [美] Michael Morrison
- 译 徐刚 于健 薛雷
- 责任编辑 李岚
- ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京顺义振华印刷厂印刷
新华书店总店北京发行所经销
- ◆ 开本: 800×1000 1/16
印张: 24.5
字数: 592 千字 2005 年 9 月第 1 版
印数: 1—4 000 册 2005 年 9 月北京第 1 次印刷
著作权合同登记号 图字: 01-2004-2603 号

ISBN 7-115-13660-2/TP · 4780

定价: 49.00 元 (附光盘)

读者服务热线: (010) 67132705 印装质量热线: (010) 67129223

内 容 提 要

本书介绍如何设计和构建自己的计算机游戏。书中从零开始，引导读者开发一个“即插即用”的游戏引擎，并基于该引擎，循序渐进地开发 7 个完整的游戏。全书分为 8 个部分，共 24 章，内容包括游戏编程基础知识、如何与玩家交互、使用子画面动画、使用声音和音乐、高级动画、游戏人工智能、增添游戏的趣味性和附加练习。此外，在随书光盘中提供有附录，包括 C++ 语言和 Windows 编程的入门指导、游戏开发工具以及游戏图形创建的介绍。

本书适用于任何对 C++ 语言有基本了解的读者阅读，适宜作为读者进入游戏开发领域的技术入门学习用书。

作者简介

Michael Morrison 是一位作家、开发人员、玩具发明家，还是多本计算机技术书籍和交互式 Web 教程的作者。此外他还担任 Stalefish Labs (他和妻子 Masheed 共同创办的一家娱乐公司)的创意经理。Stalefish Labs 的首个商业产品是一个名为 *Tall Tales: The Game of Legends and Creative One-Upmanship* 的传统社交/琐事游戏 (www.talltalesgame.com)。当 Michael 没有坐在计算机前、打曲棍球、溜冰或者和妻子一起看电影时，他喜欢去鲤鱼池边闲逛。读者可以通过 www.michaelmorrison.com 拜访 Michael。

技术编辑简介

自 1990 起，David Franson 就是联网、编程以及计算机图形领域的一位专家。2000 年，他辞去了在纽约最大的一家娱乐法律公司的信息系统主管职位，开始从事游戏开发方面的全职工作。他编写过 *2D ArtWork* 和 *3D Modeling for Game Artists* 两本书，于 2002 年 11 月出版。

献辞

献给我已故的朋友 Randy Weems，我所懂得的有关游戏编程的所有知识实际上都是他教给我的，他将在我开发的每一个游戏中永生。

致谢

感谢 Mike Stephens、Loretta Yates、Mark Renfrow、David Franson 以及 Sams Publishing 的其他工作人员，是你们给了我这样一次愉快的写作经历。此外，我还要向我的妻子 Masheed 致以深切的谢意，你是我最好的朋友和最大的支持者。

前 言

像 *Junkyard Wars* 和 *American Chopper* 这样的电视节目获得了极大的成功，在很大程度上是因为它们揭示了有趣的机器（例如潜水艇、破城槌以及摩托车等）构造背后的创造性过程。自从有了电视以来，教育性的电视节目就一直存在，但是直到最近，教育才进入了正统娱乐的领域。这些节目之所以获得成功，是因为它们展示了在头脑中产生一个想法并将它们转化为现实是多么有趣和具有挑战性。是的，它们是教育性的，但是更重要的是，它们很有趣。本书中有同样有挑战性的乐趣在等待着读者，你将学习如何构建自己的视频游戏。不，你不能轰鸣着引擎骑着自己的游戏在街道上飞速行驶，但是你可以与朋友和家人分享，炫耀自己新获得的游戏开发技能。

虽然现在很难想像，但是就在不久之前，视频游戏的概念还像是“空中楼阁”。在企业里，计算机仅仅被视为有用的工具，而在娱乐方面是没有什么潜力的。技术可以改变我们看待事物的方式，就在不久前，早期的视频游戏赢得了人们的心和整整一代人中的四分之一。是的，我是 20 世纪 80 年代生人，我觉得自己很幸运，童年有一部分是在视频游戏从幻想发展成为一种文化现象的那 10 年里度过的。

视频游戏的有趣之处在于，它们代表了交互式数字娱乐的最早形式。或许更重要的是，视频游戏代表了最早的真实交互式的艺术形式。在考虑这一点时，要想到，没有什么绘画、雕刻或者音乐乐曲能够让你与它们交互或者以任何形式改变它们。视频游戏允许创建者更多地与游戏玩家分享某种东西，使玩家有机会将自己投入到游戏中，然后看看会有什么结果。事实上，最优秀的游戏允许玩家在玩游戏时表达他们自己独特的风格和技能，同时还从游戏所揭示出来的游戏设计者对一个虚拟世界的想像中得到乐趣。

如果读者认为这个简介让人感到这是一本充满了怀旧和附庸风雅的视频游戏话题的书，那就错了。虽然我很欣赏视频游戏的历史以及游戏设计中的艺术性，但是我知道，读者感兴趣的是创建自己的游戏。诀窍就在于从历史学习经验，同时使自己获得开创新未来的能力。因此，我将使用过去的视频游戏来解释游戏创作技术并讲授一些方法，但是我的最终目标是使读者具备游戏编程知识，以便实现自己对完美计算机游戏的设想。

我相信读者已经知道了，计算机编程是一件很有挑战性但是很有价值的工作，它使开发人员可以在计算机环境里做一些有趣的事情。不幸的是，在计算机编程中，游戏编程通常有一点神秘。虽然目前存在大量游戏，但是游戏程序员实际上并没有那么多。原因主要在于游戏编程是软件开发的一个独特的挑战性领域，需要掌握多个不同的领域。即使创建一个简单的游戏，也必须了解如何编写代码来绘制点阵图形、实现子图形动画、播放数字采样的声音效果，以及处理用户的键

盘、鼠标和（或）游戏杆输入。如果说这些任务听起来有点复杂，那是因为对于没有为此做好准备的人来说，它们确实很复杂。

本书一次介绍一个游戏编程领域，并且允许读者在以前学习的知识基础上学习另一种技术。最终结果就是，读者将一开始学习一点基本的游戏编程技术，然后逐步积累知识，从而能够开发越来越复杂的游戏。这种方法的好处在于它使读者能够逐步地慢慢进入游戏编程领域，重点是在学习本书的过程中积累知识，以便完成更有挑战性的项目。

这本书会使读者成为一位游戏编程高手吗？未必！本书的目的是帮助读者在游戏编程技术方面打下坚实的基础，使用这些技术来自己创建有趣的游戏。本书没有介绍 DirectX，它是 Microsoft 的高级游戏编程库。DirectX 是一项用来创建商业游戏的复杂技术。如果读者决定正式做一些游戏编程方面的工作，那么一定会很渴望学习 DirectX，但是本书的目的是向读者展示如何最轻松地创建有趣的游戏。学习 DirectX 是很困难的，因此我认为最好是将它留给更高级的游戏编程书籍。在学习本书的过程中，读者会了解到，不需要学习 DirectX 就可以自己创建有趣而引人入胜的游戏。

本书的结构

本书分为 8 个部分，每个部分都介绍游戏编程的一个不同方面：

- 第 1 部分“入门”——在这一部分，读者将学习视频游戏开发的基础知识以及创建游戏所涉及的各个方面。读者将创建一个可重用的游戏软件引擎，并且了解如何绘制基本的图形，包括点阵图像。

- 第 2 部分“与游戏玩家交互”——在这一部分，读者将学习如何通过输入设备（例如键盘、鼠标以及游戏杆）与玩家交互。本书的这一部分还将引导读者开发第一个完整的游戏，这是一个名为 *Brainiac* 的锦标赛记忆游戏。不过，不会就此止步，还会在这一部分开发第 2 个完整的游戏，它名为 *Light Cycles*，用来怀念经典的街机游戏（arcade game）*Tron*。

- 第 3 部分“在游戏中使用子画面”——在这一部分，读者将学习子画面（sprite）动画的一系列知识。子画面动画是二维游戏编程的基础。读者将了解游戏动画的基础知识，然后创建支持在游戏中实现动画的类。还将在本书的这一部分开发第 3 个完整的游戏，这个游戏名为 *Henway*，它受到了经典的街机游戏 *Frogger* 的影响。

- 第 4 部分“使用声音和音乐”——在这一部分，读者将熟悉游戏中的音乐和声音效果。更具体地说，将了解如何创建波形声音效果，然后在游戏中播放它们，还将学习如何播放 MIDI 音乐。在具有了有关声音的丰富基础知识之后，将开始创建 *Battle Office*，这是第 4 个完整的游戏。

- 第 5 部分“高级动画”——在这一部分，将学习一些比子画面动画更高级的动画技术，它们在动作游戏中是非常有用的。读者将学习如何创建子画面的外观动画以及如何创建动画背景。然后，将在另一个完整的游戏上应用这些新知识，这个游戏名为 *Meteor Defense*，有一点类似于街机游戏 *Missile Command*。

- 第 6 部分“让游戏拥有大脑”——在这一部分，将学习人工智能（Artificial Intelligence, AI）

的基础知识，并了解它对于游戏的重要性。AI 可能是一个令人感到畏惧的主题，因此我介绍的重点是很容易理解且易于应用在自己游戏中的一些基本 AI 技术。本书的这一部分还包括了另一个完整游戏的开发过程，这个游戏名为 *Space Out*，是一个太空射击游戏，几乎结合了在整本书中学习的所有技术。

- 第 7 部分“增添游戏的趣味性”——在这一部分，将学习一些有趣的游戏编程技术，它们使游戏显得更专业。读者将学习如何创建闪屏、演示模式以及存储在磁盘上的高分列表。在此过程中，将使用各种新技术来增添 *Space Out* 游戏的趣味性，使之成为一个更具有专业水准的游戏。这是一个很好的游戏，可以将它作为自己的游戏编程项目的起点。

- 第 8 部分“附加练习”——在这一部分，将设计和开发一个冒险模拟游戏以及一个更完整的游戏，以此来结束本书的学习。冒险模拟游戏 *Wanderer* 利用了滚动的背景，使玩家可以引导一个冒险者在虚拟世界中移动。*Stunt Jumper* 游戏是一个 side-scrolling 摩托车跳跃游戏，玩家骑在一辆特技摩托车上并尝试跳过公共汽车。

对读者的要求

本书假设读者拥有 C++ 程序语言方面的知识。虽然在本书中开发的程序示例和游戏是 Windows 程序，但是并不需要 Windows 编程的背景知识。实际上，我确实没有使用任何复杂的 C++ 编程概念，因此只要基本了解 C++ 语言就完全足够了。如果读者对 C++ 技术感到有一点生疏，那么可以阅读一下光盘中的附录 B“C++ 编程入门”来温习一下。同样，光盘中的附录 C“Windows 游戏编程入门”提供了 Windows 编程中与游戏有关的基础知识，如果以前不了解 Win32 API (Application Programming Interface, 应用程序编程接口)，那么这个附录是绝对值得阅读的。

虽然事先了解 Windows 编程并不重要，但是理解如何在 Windows 平台上使用开发环境（例如 Visual C++、Borland C++Builder 或者 Dev-C++）来编译 C++ 程序是非常重要的。如果已经有了一个喜欢的 C++ 开发环境，那很好。如果还没有自己喜欢的开发环境，那么可以看看光盘中的附录 A“选择游戏开发工具”，这里描述了一些比较流行的 Windows C++ 开发工具。

可以在本书的配套光盘上获得书中的所有例子以及用于 Visual C++、C++Builder 和 Dev-C++ 的项目文件。由于为 Windows 游戏从头创建一个新项目通常是比较麻烦的，因此我希望读者尝试和使用其中一个工具以及我提供的项目。

如果决定使用这 3 个开发工具之外的工具，则需要为各个例子创建一个新项目，然后添加源代码文件。读者的开发工具可能允许导入我提供的项目，因此在手工创建新项目之前可以检查一下。如果在编译项目时遇到了问题，请确保链接了正确的标准库，在大多数程序示例和游戏中都必须导入两个库 (winmm.lib 和 msimg32.lib)。此外，还要确保将项目设置为“Windows 应用程序”而不是“控制台应用程序”。

除了一些 C++ 知识和一个优秀的 C++ 开发工具之外，读者真正需要的就是敞开的心扉以及一点创造力，这样才能从本书中获得最大的收益。当你开始进入游戏创作领域时，这两点必将使你受益。祝你学得开心！

目 录

第 1 部分 人 门

第 1 章 学习开发游戏的基础知识	3	2.3.2 GameEngine 类	22
1.1 认识视频游戏	3	2.4 开发 Blizzard 示例	33
1.1.1 为什么人们如此沉迷于开发 视频游戏	4	2.4.1 编写程序代码	33
1.1.2 视频游戏的类型	5	2.4.2 测试完成的程序	37
1.2 学习游戏设计的要素	7	2.5 小结	38
1.2.1 提出基本思想	7	2.6 游戏大变身	38
1.2.2 整理剧情	8	第 3 章 学习绘制基本图形	39
1.2.3 可视化图形	9	3.1 图形基础	39
1.2.4 为游戏选择正确的声音	10	3.1.1 理解图形坐标系	40
1.2.5 使用控制器控制游戏	10	3.1.2 学习颜色的基础知识	40
1.2.6 确定游戏模式	11	3.2 查看 Windows 中的图形	42
1.3 面向对象的编程和游戏	12	3.2.1 使用设备环境	42
1.3.1 理解 OOP	12	3.2.2 使用画笔写	43
1.3.2 在游戏中应用 OOP	13	3.2.3 使用画刷绘制	44
1.4 研究相关工具	14	3.2.4 使用位图绘制图像	44
1.4.1 编译器和开发环境	15	3.2.5 使用调色板管理颜色	44
1.4.2 选择图形工具	15	3.3 绘制窗口	45
1.4.3 选择声音和音乐工具	15	3.3.1 绘制文本	46
1.5 小结	16	3.3.2 绘制图元	47
1.6 亲身实践	16	3.3.3 使用画笔和画刷	49
第 2 章 创建游戏引擎	17	3.4 开发 Crop Circles 示例	50
2.1 什么是游戏引擎	17	3.4.1 编写程序代码	51
2.2 考虑游戏引擎的作用	18	3.4.2 测试完成的程序	54
2.2.1 将游戏分解为事件	18	3.5 小结	54
2.2.2 建立游戏的计时机制	19	3.6 亲身实践	55
2.3 开发游戏引擎	21	第 4 章 绘制图形图像	56
2.3.1 游戏事件函数	21	4.1 位图图像的基础知识	56
		4.2 深入学习位图	58
		4.3 开发位图类	58
		4.3.1 位图类的工作原理	59

4.3.2 整合代码	59	7.4 向游戏引擎添加游戏杆支持	103
4.4 开发 Slideshow 示例	62	7.4.1 访问 Win32 多媒体功能	103
4.4.1 编写程序代码	62	7.4.2 开发游戏杆代码	104
4.4.2 汇集资源	65	7.5 开发 UFO 2 示例	108
4.4.3 测试完成的程序	67	7.5.1 编写程序代码	108
4.5 小结	67	7.5.2 测试完成的程序	110
4.6 游戏大变身	68	7.6 小结	111
第 2 部分 与游戏玩家交互		7.7 亲身实践	112
第 5 章 使用键盘和鼠标控制游戏	71	第 8 章 示例游戏: <i>Light Cycles</i>	113
5.1 游戏与用户输入	71	8.1 游戏的玩法	114
5.2 了解用户输入设备	73	8.2 设计游戏	114
5.2.1 接受键盘输入	73	8.3 开发游戏	116
5.2.2 响应鼠标	74	8.4 测试游戏	127
5.2.3 使用游戏杆交互	75	8.5 小结	128
5.3 评估游戏的键盘输入	75	8.6 游戏大变身	128
5.4 跟踪鼠标	76	第 3 部分 在游戏中使用子画面	
5.5 向游戏引擎添加输入支持	77	第 9 章 使用子画面动画移动对象	131
5.5.1 添加键盘支持	78	9.1 理解动画的基础知识	131
5.5.2 添加鼠标支持	78	9.1.1 动画和帧频	132
5.5.3 修改 Bitmap 类	79	9.1.2 了解计算机动画	132
5.6 开发 UFO 示例	81	9.2 2D 动画与 3D 动画	133
5.6.1 编写程序代码	82	9.3 理解 2D 动画的类型	134
5.6.2 测试完成的程序	85	9.3.1 基于帧的动画	134
5.7 小结	86	9.3.2 基于形状的动画	134
5.8 亲身实践	86	9.4 将子画面动画应用于游戏	138
第 6 章 示例游戏: <i>Brainiac</i>	87	9.5 设计通用的子画面	139
6.1 游戏的玩法	87	9.6 创建 Sprite 类	141
6.2 设计游戏	88	9.6.1 创建和破坏子画面	142
6.3 开发游戏	89	9.6.2 更新子画面	143
6.4 测试游戏	95	9.6.3 绘制子画面	145
6.5 小结	96	9.7 开发 Planets 示例程序	146
6.6 游戏大变身	97	9.7.1 编写程序代码	146
第 7 章 使用游戏杆改进输入	98	9.7.2 测试完成的程序	150
7.1 了解游戏杆的基础知识	98	9.8 小结	151
7.2 校准游戏杆	99	9.9 亲身实践	152
7.3 追踪游戏杆的移动	101	第 10 章 管理子画面	153

10.1 了解管理子画面的需求	153	12.8.2 组合资源	194
10.2 设计子画面管理器	154	12.8.3 测试完成的游戏	195
10.3 向游戏引擎添加子画面管理器	155	12.9 小结	196
10.3.1 改进 Sprite 类	155	12.10 亲身实践	196
10.3.2 增强游戏引擎	157	第 13 章 播放 MIDI 音乐	198
10.4 使用双重缓存消除闪烁	159	13.1 感受 MIDI 音乐	198
10.5 开发 Planets 2 示例	161	13.2 理解 Windows 媒体控制接口	200
10.5.1 编写程序代码	161	13.3 使用 MCI 播放 MIDI 音乐	201
10.5.2 测试完成的程序	165	13.3.1 打开 MIDI 设备	201
10.6 小结	165	13.3.2 播放 MIDI 歌曲	202
10.7 游戏大变身	166	13.3.3 暂停 MIDI 歌曲	203
第 11 章 示例游戏: Henway	167	13.3.4 关闭 MIDI 设备	203
11.1 游戏的玩法	167	13.4 向游戏引擎添加 MIDI 音乐支持	203
11.2 设计游戏	169	13.5 开发 Henway 2 示例程序	205
11.3 开发游戏	170	13.5.1 编写程序代码	205
11.4 测试游戏	177	13.5.2 测试完成的游戏	209
11.5 小结	179	13.6 小结	210
11.6 游戏大变身	179	13.7 亲身实践	210
第 4 部分 使用声音和音乐		第 14 章 示例游戏: Battle Office	211
第 12 章 播放数字声音效果	183	14.1 游戏的玩法	211
12.1 理解数字声音	183	14.2 设计游戏	212
12.2 了解 Windows 波形声音	185	14.3 开发游戏	215
12.3 了解声音工具	186	14.4 测试游戏	222
12.3.1 Sony Sound Forge	187	14.5 小结	223
12.3.2 Adobe Audition	187	14.6 亲身实践	224
12.3.3 廉价的声音编辑器	187	第 5 部分 高级动画	
12.4 创建和编辑声音	187	第 15 章 实现子画面外观动画	227
12.5 查找声音和音乐	188	15.1 再次介绍帧动画	227
12.6 访问波形声音	188	15.2 设计动画子画面	228
12.7 播放波形声音	189	15.3 向游戏引擎添加动画子画面支持	230
12.7.1 播放文件中的波形声音	190	15.3.1 只绘制位图的一部分	230
12.7.2 播放作为资源的波形声音	190	15.3.2 对 Sprite 类实现动画	231
12.7.3 循环播放波形声音	191	15.4 开发 Battle Office 2 示例程序	233
12.7.4 停止播放波形声音	191	15.4.1 编写程序代码	233
12.8 开发 Brainiac 2 示例程序	192	15.4.2 测试完成的游戏	235
12.8.1 编写程序代码	192	15.5 小结	235

15.6	游戏大变身	236	18.5	小结	281
第 16 章	创建子画面背景	237	18.6	亲身实践	281
16.1	了解游戏背景的重要性	237	第 19 章	示例游戏: <i>Space Out</i>	282
16.2	了解游戏背景的类型	238	19.1	游戏的玩法	282
16.2.1	纯色背景	238	19.2	设计游戏	283
16.2.2	图像背景	239	19.3	向游戏引擎中添加另一个子画面特性	286
16.2.3	动画背景	239	19.4	开发游戏	287
16.2.4	滚动背景	239	19.5	测试游戏	297
16.3	向游戏引擎添加背景支持	240	19.6	小结	298
16.3.1	创建基本的背景类	240	第 7 部分 增添游戏的趣味性		
16.3.2	创建动画背景类	242	第 20 章	使用闪屏增添游戏的活力	303
16.4	开发 Roids 示例	245	20.1	闪屏的重要性	303
16.4.1	编写程序代码	245	20.2	了解闪屏	304
16.4.2	测试完成的游戏	248	20.3	创建 <i>Space Out 2</i> 游戏	305
16.5	小结	248	20.3.1	编写游戏代码	305
16.6	亲身实践	249	20.3.2	测试完成的游戏	309
第 17 章	示例游戏: <i>Meteor Defense</i>	250	20.4	小结	309
17.1	游戏的玩法	250	20.5	亲身实践	309
17.2	设计游戏	251	第 21 章	使用演示模式展示游戏	311
17.3	增强游戏引擎中的子画面	254	21.1	什么是演示模式	311
17.4	开发游戏	256	21.2	演示模式的细节	312
17.5	测试游戏	264	21.3	开发 <i>Space Out 3</i> 游戏	313
17.6	小结	265	21.3.1	编写游戏代码	313
17.7	游戏大变身	265	21.3.2	测试完成的游戏	318
第 6 部分 让游戏拥有大脑			21.4	小结	319
第 18 章	教游戏思考	269	21.5	游戏大变身	320
18.1	理解人工智能	269	第 22 章	记录高分	321
18.2	了解游戏 AI 的类型	271	22.1	记录得分的重要性	321
18.2.1	漫游 AI	272	22.2	确定高分数据模型	322
18.2.2	行为 AI	274	22.3	存储和检索高分数据	323
18.2.3	策略 AI	275	22.4	开发 <i>Space Out 4</i> 游戏	325
18.3	开发 AI 策略	276	22.4.1	编写游戏代码	325
18.4	开发 Roids 2 示例程序	276	22.4.2	测试完成的游戏	330
18.4.1	编写程序代码	277	22.5	小结	331
18.4.2	测试完成的程序	280	22.6	亲身实践	332

第 8 部分 附加练习

- 第 23 章 使用滚动背景更改远景** 335
- 23.1 什么是滚动背景 335
- 23.1.1 横向滚动的背景 336
- 23.1.2 纵向滚动的背景 336
- 23.1.3 视差滚动背景 337
- 23.2 理解滚动背景的工作原理 337
- 23.3 向游戏引擎添加滚动背景支持 338
- 23.3.1 创建背景图层类 339
- 23.3.2 创建滚动背景类 344
- 23.4 开发 Wanderer 示例 346
- 23.4.1 编写程序代码 346
- 23.4.2 测试完成的程序 354
- 23.5 小结 355
- 23.6 游戏大变身 355
- 第 24 章 示例游戏: Stunt Jumper** 357
- 24.1 游戏的玩法 357
- 24.2 设计游戏 359
- 24.3 开发游戏 361
- 24.4 测试游戏 374
- 24.5 小结 376
- 24.6 游戏大变身 376
- 24.7 亲身实践 377
- 配套光盘上的附录**
- 附录 A 选择游戏开发工具** CD:381
- A.1 Microsoft Visual C++ CD:382
- A.2 Borland C++ Builder CD:383
- A.3 Bloodshed Dev-C++ CD:383
- A.4 DJGPP CD:383
- 附录 B C++编程入门** CD:384
- B.1 C++语言基础 CD:384
- B.2 C++开发周期 CD:385
- B.3 创建一个简单的 C++程序 CD:385
- B.4 使用注释说明代码 CD:387
- B.5 使用空白分割代码 CD:387
- B.6 将代码组织到函数中 CD:387
- B.7 使用表达式操作数据 CD:388
- B.8 增强程序示例 CD:389
- B.9 变量和常量 CD:390
- B.9.1 变量的类型 CD:391
- B.9.2 区分大小写 CD:392
- B.9.3 关键字 CD:392
- B.9.4 字符串和数组 CD:392
- B.10 if 语句 CD:393
- B.11 关系运算符 CD:393
- B.12 函数 CD:395
- B.13 全局变量 CD:396
- B.14 模块 CD:397
- B.15 执行多次操作 CD:397
- B.16 至少执行一次操作 CD:398
- B.17 执行 0 次或多次操作 CD:398
- B.18 for 循环 CD:399
- B.19 switch 语句 CD:399
- B.20 加 1 和减 1 CD:400
- B.21 学习类的基础知识 CD:401
- B.21.1 类与实例 CD:401
- B.21.2 理解方法 CD:402
- B.21.3 初始化和清理对象 CD:402
- B.21.4 析构函数 CD:403
- B.22 进一步学习 CD:403
- 附录 C Windows 游戏编程入门** CD:404
- C.1 Windows 编程基础 CD:404
- C.1.1 事件驱动编程 CD:405
- C.1.2 使用消息通信 CD:405
- C.1.3 理解设备独立性 CD:406
- C.1.4 将程序信息存储为资源 CD:406
- C.1.5 处理陌生的数据类型 CD:406
- C.1.6 非传统的编码约定 CD:407
- C.2 深入了解 Windows 程序 CD:408
- C.2.1 入门 CD:408
- C.2.2 窗口类 CD:408
- C.2.3 创建窗口 CD:409

C.2.4 处理消息.....	CD:410	D.2 了解图形工具.....	CD:420
C.2.5 窗口过程.....	CD:411	D.2.1 Image Alchemy.....	CD:420
C.2.6 使用资源.....	CD:412	D.2.2 Paint Shop Pro.....	CD:421
C.3 开发 Skeleton 示例.....	CD:413	D.2.3 Graphic Workshop.....	CD:421
C.3.1 编写程序代码.....	CD:414	D.3 创建和编辑图形.....	CD:421
C.3.2 组合资源.....	CD:416	D.3.1 艺术线条图形.....	CD:421
C.3.3 测试完成的程序.....	CD:417	D.3.2 3D 渲染图形.....	CD:422
附录 D 创建游戏图形.....	CD:418	D.3.3 扫描的照片和视频捕捉的 图形.....	CD:422
D.1 评估游戏图形.....	CD:418	D.3.4 背景图形和纹理.....	CD:423
D.1.1 确定游戏屏幕大小.....	CD:418	D.3.5 动画图形.....	CD:423
D.1.2 考虑目标观众的影响.....	CD:419	D.4 查找图形.....	CD:423
D.1.3 确定游戏环境和情绪.....	CD:419		
D.1.4 坚持一种图形风格.....	CD:420		

第 1 部分

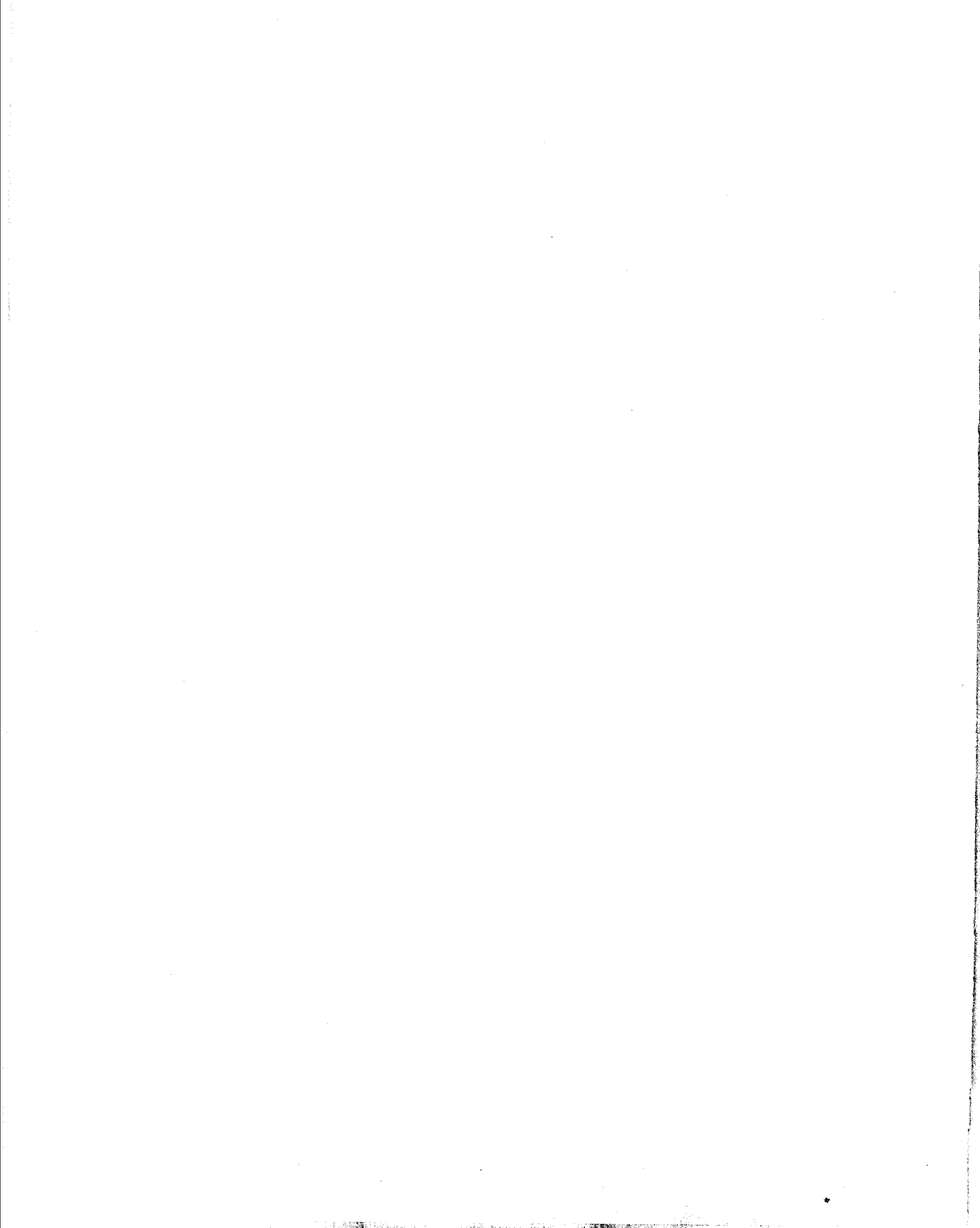
入 门

第 1 章 学习开发游戏的基础知识

第 2 章 创建游戏引擎

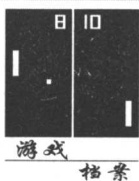
第 3 章 学习绘制基本图形

第 4 章 绘制图形图像



第 1 章

学习开发游戏的基础知识



在视频游戏的变革期长大成人是我的幸运，最初是 *Pong* 让我认识了这种交互式数字艺术形式，现在我们称之为视频游戏。*Pong* 是由 Atari（它是最早兴起硅谷技术非传统文化的公司之一，这种文化点亮了计算机行业）的创办者 Nolan Bushnell 于 1972 年开发的。*Pong* 在设计上非常简单，不过它仍然令一代年轻人为之着迷，他们为了能够通过电子媒介相互较量而狂喜不已。虽然 *Pong* 最初是作为街机游戏，但是它直到作为第一个家庭视频游戏发布之后才变得家喻户晓。虽然许多人都认为 *Pong* 是第一个视频游戏，但其实这项殊荣应该属于一个不那么知名的游戏 *Computer Space*，它于 1971 年发布，也是 Nolan Bushnell 开发的。

如果读者熟悉 *Pong*，那么肯定会注意到过去 30 多年来所取得的进步，是这些进步让我们得以拥有在视听吸引力方面可以与电影匹敌的视频游戏。如果读者曾经对这些游戏的工作原理感到好奇，那么你现在真是来对了地方。我开始涉足视频游戏开发也是始于这种惊讶和好奇心，感觉我有能力创建按照自己的规则（我的规则）运转的小型虚拟世界。在有了一些编程知识之后，带着看看我的想像能够产生什么结果的渴望，我学习了开发视频游戏的基本知识。不幸的是，我的学习过程非常艰难，不得不靠自己构建解决方案，甚至解决一些最简单的问题也要经过不断地尝试。读者是幸运的，不必再经历这种痛苦的过程。本章将介绍开发游戏的基础知识。

本章内容包括：

- 视频游戏的不同类型；
- 在设计新游戏时考虑的主要方面；
- 为什么面向对象的编程（OOP）对于游戏开发很重要；
- 作为一名游戏开发人员，将使用哪些类型的工具。

1.1 认识视频游戏

在开始这里的讨论之前，有必要指出一点，视频游戏并不是一直都像今天这样受到尊重，被