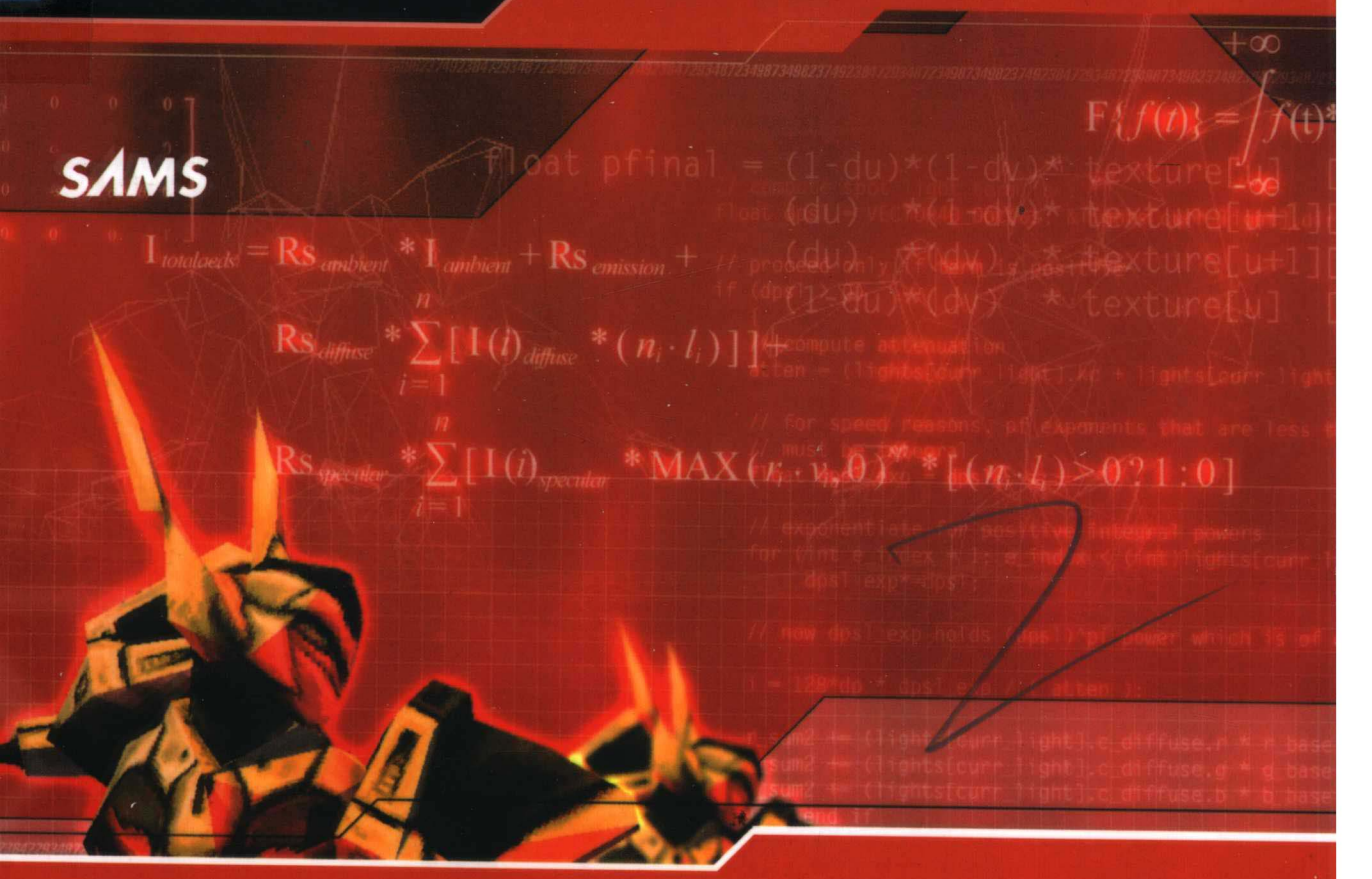
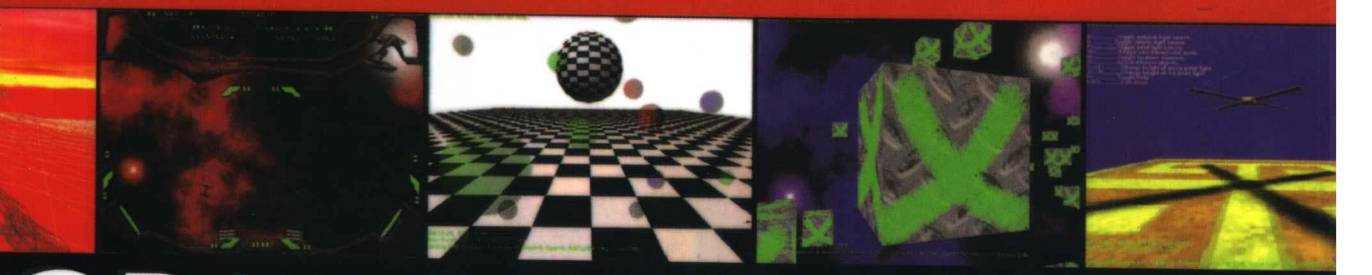
SAMS





3D游戏编程大师技巧

TRICKS OF THE 3D GAME PROGRAMMING GURUS

ADVANCED 3D GRAPHICS AND RASTERIZATION

附光盘

[美] André LaMothe 著
李祥瑞 陈 武 译

人民邮电出版社
POSTS & TELECOM PRESS

3D 游戏编程大师技巧

[美] André LaMothe 著

李祥瑞 陈武 译

人民邮电出版

图书在版编目（CIP）数据

3D 游戏编程大师技巧 /（美）拉莫泽（Lamothe, A.）著；李祥瑞，陈武译.
—北京：人民邮电出版社，2005.6

ISBN 7-115-13371-9

Ⅰ. 3… Ⅱ. ①拉…②李…③陈… Ⅲ. 游戏—应用程序—程序设计—教材 Ⅳ. G899

中国版本图书馆 CIP 数据核字（2005）第 026252 号

3D 游戏编程大师技巧

◆ 著 [美] André LaMothe
译 李祥瑞 陈 武
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

读者热线 010-67132705
北京鸿佳印刷厂印刷
新华书店总店北京发行所经销

◆ 开本：787×1092 1/16
印张：58.75
字数：1 921 千字 2005 年 6 月第 1 版
印数：1 –3 500 册 2005 年 6 月北京第 1 次印刷

著作权合同登记号 图字：01-2003-6127 号

ISBN 7-115-13371-9/TP • 4645

定价：118.00 元（附光盘）

本书如有印装质量问题，请与本社联系 电话：（010）67129223

内容提要

本书是游戏编程畅销书作者 André LaMothe 的扛鼎之作，从游戏编程和软件引擎的角度深入探讨了 3D 图形学的各个重要主题。全书共分 5 部分，包括 16 章的内容。第 1～3 章简要地介绍了 Windows 和 DirectX 编程，创建了一个 Windows 应用程序模板，让读者能够将精力放在游戏逻辑和图形实现中，而不用考虑 Windows 和 DirectX 方面的琐事；第 4～5 章简要地介绍了一些数学知识并实现了一个数学库，供以后编写演示程序时使用；第 6 章概述了 3D 图形学，让读者对本书将介绍的内容有大致的了解；第 7～11 章分别介绍了光照、明暗处理、仿射纹理映射、3D 裁剪和深度缓存等内容；第 12～14 章讨论了高级 3D 渲染技术，包括透视修正纹理映射、Alpha 混合、1/z 缓存、纹理滤波、空间划分和可见性算法、阴影、光照映射等；第 15～16 章讨论了动画、运动碰撞检测和优化技术。

本书适合于有一定编程经验并想从事游戏编程工作或对 3D 图形学感兴趣的人员阅读。

作者简介

André LaMothe 有 25 年的计算行业从业经验，拥有数学、计算机科学和电子工程等学位，是 20 岁时就在 NASA 做研究工作的少数几人之一。在 30 岁之前，他在硅谷的众多公司中从事过咨询工作，了解了公司运作，获得了多种领域的知识，如电信、虚拟现实、机器人技术、编译器设计、3D 引擎、人工智能以及计算和工程的其他领域的知识。

他创办的公司 Xtreme Game 公司一直是自成一体的游戏开发、发行商。后来他创办了 Xtreme Games Developer Conference（XGDC），为游戏开发人员提供了费用更低廉的 GDC 替代品。

最近，他参与了多个项目的开发工作，其中包括 eGamezone Networks——一个公平、有趣、没有任何广告的网络游戏分发系统。他还创建了一家新公司——Nurve Networks 公司，为在乎价格的消费者和业余爱好者开发手持设备上的视频游戏系统。最后，他还是世界上最庞大的游戏开发系列丛书的编辑。

技术审校人员简介

David Franson 从 1990 年起开始从事网络、编程、2D/3D 计算机图形学等工作。2000 年，他辞去了纽约最大的娱乐律师事务所之一的信息系统总监职务，全身心地投入游戏开发工作。他还是 *2D Artwork and 3D Modeling for Game Artists* 的作者，当前正在编写 *Xbox Hackz and Modz* 一书。

献　　词

谨以此书献给所有诚实正直的人们——人间正道是沧桑。您的举动不会默默无闻……

致　谢

首先要感谢 Sams 出版社的全体员工。这里有必要说明一下，Sams 出版社有一套非常严格的图书出版流程，“大师技巧”系列图书都是完全按这种流程成书的。

感谢出版人 Michael Stephens、组稿编辑 Kim Spilker、可信赖的开发编辑 Mark Renfrow，当然还有确保一切工作顺利进行的项目编辑 George Nedeff。书中还有很多具体细节，这首先要感谢文字编辑 Seth Kerney，当然还有技术编辑 David Franson，他提供了包括摩托艇在内的一些 3D 模型以及演示程序中使用的众多纹理。

最后，本书如果没有附带光盘将是不完整的，感谢媒体开发人员 Dan Scherf；还要感谢无名英雄 Erika Millen 为本书制作了详细而完备的索引——这对于读者查找内容至关重要。

有一个特殊人物这里不能不提，那就是 Angela Kozlowski；她与我一道完成了本书前面的 1/4，帮助确定了整体框架，让所有其他人明白本书有多么重要和与众不同。在本书完成之前，她被调往其他岗位工作。

接下来要感谢那些为我提供、租借或帮助我获得软件和硬件的公司和个人。Intel 公司的 David Hackson 提供了最新的高级 C++和 Fortran 编译器以及 VTune， Caligari 公司的 Kristine Gardner 提供了各种版本的 trueSpace，微软公司的老朋友 Stacey Tsurusaki 给我提供了微软公司内部使用的尖端工具。当然，还有一些公司给我提供了其产品的评估版或允许我将其软件放到本书的附带光盘中，如 JASC 公司允许我使用 Paint Shop Pro，Sonic Foundry 允许我使用 Sound Forge，Right Hemisphere's Mary Alice Krayecki 提供了 Deep Exploration and Deep Paint UV 的拷贝。

接下来要感谢那些在我撰写本书期间提供帮助的朋友。我这个人“爱憎分明”，但正是这一点让我在聚会中很有趣！

回到正题，感谢 Mike Perone 帮助我获得“软件”并处理网络方面的问题；感谢 Mark Bell 耐心地倾听我抱怨——只有创业者才能理解我们经历的困境；感谢参与举办 Vintage 计算机节的 Sellam Ismail 提供大量非常棒且价格低廉的东西；感谢 John Romero 经常同我交流，让我相信这个行业中有那么有趣的人！

感谢 Nolan Bushnell 邀请我前往 uWink 公司并抽空与我讨论视频游戏，与创建 Atari 的人共度时光可是难得的机会！感谢他为本书作序。

另外，感谢我的新助手 Alex Varanese 容忍我喋喋不休地谈论尽善尽美、最后期限、超越极限、展望成功。

最后，感谢那些真正能够容忍我好斗、苛刻个性的人：母亲、父亲、耐心的女友 Anita，还有小狗 Ariel。

序

很荣幸应邀为这本向程序员介绍从零开始创建下一代视频游戏所需技能的重要著作做序。从绘制像素开始介绍创建实时 3D 引擎的著作并不多。以前，先进技术和 Atari 公司开发的粗糙游戏形成了鲜明的反差，对此进行反思时发现，我们确实曾致力于提高技术发展水平，但看起来收效甚微。

回顾过去，早期的游戏从技术的角度看根本算不上计算机游戏，它们不过是奇特的信号生成器，是使用计数器和基于布尔逻辑的移位寄存器的状态机，由拼凑而成的 MSI（中等规模集成）门组成。读者可能还记得我于上世纪 70 年代开发的第一款游戏——*Computer Space*，它面世的时间比 Intel 4004 早了 4 年，比 8080 早 6 年。我曾希望有微处理器来运行它！在那时候，对于任何重要的实时计算而言，典型的时钟速度太慢了。我们开发的第一款使用微处理器的游戏是 *Asteriods*，即使是这样，仍使用了大量的硬件来支持该程序，因为微处理器不能完成软件的所有工作。

当前，我们正在通向创建照片级图像的道路上迈进，虽然这种目标还未达到。能够动态地创建这样的图像确实令人兴奋。软件和硬件工具为游戏制作人员提供了非常强大的创建真实感游戏世界、环境和角色的功能。使用这些功能可以缩短游戏制作周期，增加财富，使开发新的游戏项目成为可能。

André LaMothe 不但谙熟这些技术，还有独特的“游戏感”。多年来，我见过很多精通尖端技术的专家，但缺乏编写优秀游戏必备的游戏感；而其他人有良好的游戏感，却是平庸的程序员。André不但是真正的游戏大师，还是软件大师，这一点在他撰写的每本著作中都表现得淋漓尽致。

在我们最近合作开发的一个项目中，André对尖端技术的精通、历史知识的广博以及对早期一些默默无闻游戏的了解之深，给我留下了深刻的印象。更值得一提的是，他竟然只花了 19 天的时间就为我编写了一款完整的游戏！知道一些成功案例很容易，但对败笔也了如指掌很难。是的，Atari 确实有一些糟糕的败笔，但如大家所知，我们也开发了很多著名的经典游戏。

希望读者喜欢本书，并以此为跳板，在未来开发出让我流连忘返的优秀游戏。

Atari 公司创始人
Nolan Bushnell

前　言

游戏编程原理和实践

很久以前，我编写了一本有关游戏编程的图书《Windows 游戏编程大师技巧》，终于实现了夙愿——为读者编写一本介绍如何制作游戏的图书。多年后，我在游戏编程方面的经验更加丰富，思想也更睿智，同时学会了更多游戏编程的技巧。读者即使没有阅读过《Windows 游戏编程大师技巧》，也能读懂本书；但需要提醒您的是，本书的内容更深，重点为 3D 游戏编程，要求读者具备众多的背景知识。

本书将续写和弥补《Windows 游戏编程大师技巧》没有涉及的内容，进一步探讨 3D 游戏编程的概念，在时间和篇幅允许的情况下，尽可能地涵盖 3D 游戏编程的每个重要主题。

当然，我不会假设读者是位大师级程序员且已经知道如何制作游戏。本书是为游戏编程新手编写的，针对的读者群是中高级程序员。如果读者不熟悉 C/C++编程，势必会在阅读本书的过程中深感迷惘。市面上有很多优秀的 C/C++图书，建议读者参考 Stephen Prata 或 Robert Lafore 的著作。在笔者看来，他们是世界上最棒的 C/C++图书作者。

当前是有史以来游戏行业最美好的时代。现在的技术足以让您创建出栩栩如生的游戏！想象一下即将出现的技术，PlayStation 2、Xbox 和 GameCube 都很酷。然而，这些技术掌握起来并不容易，您必须付出艰苦的努力。

当前，游戏编程的难度更高了，为制作游戏必须掌握更多的技能。然而，如果您正阅读这段文字，表明您乐于迎接挑战。算您找对了地方，阅读本书后，您将能够使用自己编写的软件光栅化模块，制作出支持纹理映射和光照效果的 3D PC 视频游戏。另外，您还将理解 3D 图形学的基本原理，更深入地认识与运用当前和未来的 3D 硬件。

内容简介

本书的内容极其丰富，涵盖了创建基于 Windows 9x/2000 的 PC 游戏所需的全部知识，其中包括以下主题：

- 《Windows 游戏编程大师技巧》中开发的引擎；
- Win32 编程和 DirectX 基础知识；
- 包括四元数在内的高等数学知识；
- 2D/3D 图形学和算法；
- 3D 投影和相机操控；
- 线框和实心模式渲染；
- 光照和纹理映射；
- 高级可见性算法；
- 3D 动画技术。

读者可能会问，本书介绍如何使用 3D 硬件还是使用代码实现 3D 软件光栅化模块？

答案是后者。只有懦弱的人才依赖于 3D 硬件，真正的游戏程序员能够从头开始编写 3D 引擎，对这样的工作充满激情，同时知道如何使用 3D 硬件。本书将介绍真正的 3D 游戏编程，具备这些知识后，读者将能够在两三周内学会使用任何 3D API。

在作者看来，如果您知道如何编写纹理映射函数和观察系统，使用硬件时将更为得心应手。另外，您不能假设每台计算机都配置了优秀的 3D 硬件，这样的时代还没有到来；只因为计算机没有 3D 硬件就将其排除在目标市场之外是非常糟糕的，尤其是在您没有数百万的资金，又想进入游戏市场时。在这种情况下，您将从非 3D 加速的软件市场着手。

最后，读者肯定对“Windows-DirectX”有些担心。只要方法得当，Windows 编程实际上非常简单、有趣，DOS32 编程中的很多问题都不再会出现。不要将 Windows 编程视为障碍——它让我们能够将更多的时间花在游戏代码而不是诸如 GUI、I/O 和图形驱动程序等细节上。如果想为市面上所有的 2D/3D 加速硬件编写图形驱动器，就是日夜不停地干也干不完，况且还有声卡、游戏杆等硬件呢。

读者必须具备的知识

本书假定读者有很强的编程技能。如果您不懂如何编写 C 语言代码，不知道怎样使用编译器，将会在阅读本书时感到相当迷惘。本书还使用了一些 C++代码，这可能会让 C 语言程序员感到有些担心。不过也不用怕，我在做任何怪异的事情前将提醒读者。如果您需要 C++程序设计的速成课程，可参阅附录 D。基本上，本书只有有关 DirectX 的范例偶尔使用了 C++。

然而，我还是决定在本书中稍微多用些 C++，因为在游戏编程中，很多东西都是面向对象的，如果将它们设计成 C 语言风格的结构，简直是悖理逆天。总之，如果您能够使用 C 语言进行编程，那很好；如果能够使用 C/C++进行编程，阅读本书将完全不成问题。

众所周知，计算机程序是由逻辑和数学计算组成的。3D 视频游戏的重点在数学运算，数学在 3D 图形学中几乎无处不在。幸运的是，读者只需具备一些基本的代数和几何学知识即可；本书还将介绍有关向量和矩阵的知识。读者只要知道加、减、乘、除，就可以理解 90%以上的内容，虽然不能亲自推导。毕竟，最终的目的只是要能使用其中的代码。

本书的组织结构

本书由 6 部分组成。

- 第一部分——3D 游戏编程简介。简要地介绍游戏、Windows 和 DirectX 编程，将建立一个虚拟计算机接口，用于创建所有的演示程序。
- 第二部分——3D 数学和变换。介绍各种数学概念，创建一个供本书使用的完整数学库。这部分的最后几章涉及 3D 图形学、数据结构、相机和线框模式渲染等。
- 第三部分——基本 3D 渲染。讨论光照、基本着色、隐藏面消除和 3D 裁剪。
- 第四部分——高级 3D 渲染。讨论纹理映射、高级光照、阴影以及 BSP 树、入口等空间划分算法等。
- 第五部分——高级动画、物理建模和优化。介绍动画、运动、碰撞检测、简单物理建模等。另外，还将讨论层次型建模、加载大型游戏世界和众多的优化技术。

附带光盘的内容

附带光盘包含本书全部的源代码、可执行文件、范例程序、素材、软件程序、音效和技术文章，其目录结构如下：

```
Tricks 3D\

SOURCE\
      T3DIICHAP01\
```

```
        T3DIICHAP02\
                .
                .
        T3DIICHAP16\

TOOLS\
GAMES\
MEDIA\
        BITMAPS\
        3DMODELS\
        SOUND\


DIRECTX\


ARTICLES\
```

每个主目录都包含您所需的特定数据，具体情况如下。

- Tricks 3D：包含其他所有目录的根目录。请阅读 README.TXT 文件以便了解最后的修改。
- SOURCE：按章节顺序收录了书中所有的源代码。只需将整个 SOURCE\目录拷贝到硬盘上就可以使用。
- TOOLS：收录了各公司慷慨地允许我放入本光盘的演示版程序。
- MEDLA：可在您的游戏中随便使用的图像、声音和模型。
- DIRECTX：最新版本的 DirectX SDK。
- GAMES：大量演示了软件光栅化的共享版 2D、3D 游戏。
- ARTICLES：由 3D 游戏编程领域的许多老手撰写的启迪性文章。

附带光盘包含各种程序和数据，因此没有统一的安装程序，您需要自行安装不同的程序和数据。然而，在大多数情况下，只需将 SOURCE\目录拷贝到硬盘中就可以了。至于其他程序和数据，可以在需要时安装它们。

安装 DirectX

附带光盘中最重要的、必须安装的部分是 DirectX SDK 及其运行阶段文件。安装程序位于 DIRECTX\目录中，该目录中还有一个 README.TXT 文件，阐明了最后的修改。

注意：必须安装了 DirectX 8.1 SDK 或更高版本（附带光盘中提供了 DirectX 9.0）才能使用本书的源代码。如果不能确定系统中是否已经安装了最新版本的 DirectX SDK，请运行安装程序进行确认。

编译程序

本书的程序是使用 Microsoft Visual C++ 6.0 编写的。然而，多数情况下，也可以任何与 Win32 兼容的编译器进行编译。尽管如此，我还是推荐使用 Microsoft VC++或.NET，因为用它们做这类工作最有效率。

如果您不熟悉您的编译器集成开发环境（IDE），编译 Windows 程序时肯定会遇到麻烦。因此，编译程序之前，请务必花些时间来熟悉编译器，至少达到知道如何编译控制台（console）程序“Hello World”的程度。

要编译生成 Windows Win32.EXE 程序，只需将工程的目标程序设置为 Win32.EXE，再进行编译。然而，要创建 DirectX 程序，必须在工程中包含 DirectX 导入库。您可能认为只要将 DirectX 库添加到包含路径（Include path）中即可，但这样不行。为避免麻烦，最好手工将 DirectX .LIB 文件包含到工程中，.LIB 文件位于 DirectX SDK 安装目录中的 LIB\目录下。这样将不会出现链接错误。在大多数情况下，需要下面这些文件。

- DDRAW.LIB：DirectDraw 导入库。

- DINPUT.LIB：DirectInput 导入库。
- DINPUT8.LIB：DirectInput8 导入库。
- DSOUND.LIB：DirectSound 导入库。
- WINMM.LIB： Windows 多媒体扩展库。

具体使用上述文件时，将更详细地介绍它们；当链接器指出“未知符号（Unresolved Symbol）”错误时，请检查是否包含了这些库。我不想从新手那里再收到有关这方面的电子邮件。

除 DirectX .LIB 文件外，还需要将 DirectX .H 文件放到头文件搜索路径中。另外，请务必将 DirectX SDK 目录放在搜索路径列表的最前面，因为很多 C++编译器带有旧版本的 DirectX，编译器可能在其 INCLUDE\ 目录下找到旧版本的头文件，而使用这些头文件是错误的。正确的位置是 DirectX SDK 的包含目录，即 DirectX SDK 安装目录中的 INCLUDE\目录。

最后，如果读者使用的是 Borland 产品，请务必使用 Borland 版本的 DirectX .LIB 文件，它们位于 DirectX SDK 安装目录中的 BORLAND\目录下。

目　录

第一部分　3D 游戏编程简介

第 1 章　3D 游戏编程入门 ……2

1.1　简介 ……2
1.2　2D/3D 游戏的元素 ……3
1.2.1　初始化 ……3
1.2.2　进入游戏循环 ……3
1.2.3　读取玩家输入 ……4
1.2.4　执行 AI 和游戏逻辑 ……4
1.2.5　渲染下一帧 ……4
1.2.6　同步显示 ……4
1.2.7　循环 ……4
1.2.8　关闭 ……5
1.3　通用游戏编程指南 ……7
1.4　使用工具 ……9
1.4.1　3D 关卡编辑器 ……12
1.4.2　使用编译器 ……13
1.5　一个 3D 游戏范例：*Raiders 3D* ……15
1.5.1　事件循环 ……33
1.5.2　核心 3D 游戏逻辑 ……34
1.5.3　3D 投影 ……35
1.5.4　星空 ……36
1.5.5　激光炮和碰撞检测 ……37
1.5.6　爆炸 ……37
1.5.7　玩 *Raiders3D* ……37
1.6　总结 ……37

第 2 章　Windows 和 DirectX 简明教程 ……38

2.1　Win32 编程模型 ……38
2.2　Windows 程序的最小需求 ……39
2.3　一个基本的 Windows 应用程序 ……43
2.3.1　Windows 类 ……43
2.3.2　注册 Windows 类 ……47
2.3.3　创建窗口 ……47
2.3.4　事件处理程序 ……48
2.3.5　主事件循环 ……52
2.3.6　构建实时事件循环 ……55
2.4　DirectX 和 COM 简明教程 ……56
2.4.1　HEL 和 HAL ……57
2.4.2　DirectX 基本类 ……58
2.5　COM 简介 ……59
2.5.1　什么是 COM 对象 ……60

2.5.2 创建和使用 DirectX COM 接口……61
2.5.3 查询接口……62
2.6 总结……64

第 3 章 使用虚拟计算机进行 3D 游戏编程……65

3.1 虚拟计算机接口简介……65
3.2 建立虚拟计算机接口……66
3.2.1 帧缓存和视频系统……66
3.2.2 使用颜色……70
3.2.3 缓存交换……71
3.2.4 完整的虚拟图形系统……73
3.2.5 I/O、声音和音乐……73
3.3 T3DLIB 游戏控制台……74
3.3.1 T3DLIB 系统概述……74
3.3.2 基本游戏控制台……74
3.4 T3DLIB1 库……79
3.4.1 DirectX 图形引擎体系结构……79
3.4.2 基本常量……79
3.4.3 工作宏……81
3.4.4 数据类型和结构……81
3.4.5 函数原型……84
3.4.6 全局变量……88
3.4.7 DirectDraw 接口……89
3.4.8 2D 多边形函数……92
3.4.9 数学函数和错误函数……97
3.4.10 位图函数……99
3.4.11 8 位调色板函数……102
3.4.12 实用函数……104
3.4.13 BOB（Blitter 对象）引擎……106
3.5 T3DLIB2 DirectX 输入系统……112
3.6 T3DLIB3 声音和音乐库……116
3.6.1 头文件……117
3.6.2 类型……117
3.6.3 全局变量……117
3.6.4 DirectSound API 封装函数……118
3.6.5 DirectMusic API 封装函数……121
3.7 建立最终的 T3D 游戏控制台……124
3.7.1 映射真实图形到虚拟接口的非真实图形……124
3.7.2 最终的 T3DLIB 游戏控制台……126
3.8 范例 T3LIB 应用程序……134
3.8.1 窗口应用程序……134
3.8.2 全屏应用程序……135
3.8.3 声音和音乐……136
3.8.4 处理输入……136
3.9 总结……139

第二部分 3D 数学和变换

第 4 章 三角学、向量、矩阵和四元数……142

4.1 数学表示法……142
4.2 2D 坐标系……143
4.2.1 2D 笛卡尔坐标……143
4.2.2 2D 极坐标……144
4.3 3D 坐标系……147
4.3.1 3D 笛卡尔坐标……147
4.3.2 3D 柱面坐标……149
4.3.3 3D 球面坐标……150
4.4 三角学……151
4.4.1 直角三角形……151
4.4.2 反三角函数……153
4.4.3 三角恒等式……153
4.5 向量……154
4.5.1 向量长度……155
4.5.2 归一化……155
4.5.3 向量和标量的乘法……155
4.5.4 向量加法……156
4.5.5 向量减法……157
4.5.6 点积……157
4.5.7 叉积……159
4.5.8 零向量……160
4.5.9 位置和位移向量……160
4.5.10 用线性组合表示的向量……161
4.6 矩阵和线性代数……161
4.6.1 单位矩阵……162
4.6.2 矩阵加法……163
4.6.3 矩阵的转置……163
4.6.4 矩阵乘法……164
4.6.5 矩阵运算满足的定律……165
4.7 逆矩阵和方程组求解……165
4.7.1 克来姆法则……167
4.7.2 使用矩阵进行变换……168
4.7.3 齐次坐标……169
4.7.4 应用矩阵变换……170
4.8 基本几何实体……176
4.8.1 点……176
4.8.2 直线……176
4.8.3 平面……179
4.9 使用参数化方程……182
4.9.1 2D 参数化直线……182
4.9.2 3D 参数化直线……184
4.10 四元数简介……189
4.10.1 复数理论……189
4.10.2 超复数……193
4.10.3 四元数的应用……197
4.11 总结……200

第 5 章 建立数学引擎……201

5.1 数学引擎概述……201
5.1.1 数学引擎的文件结构……201
5.1.2 命名规则……202
5.1.3 错误处理……203
5.1.4 关于 C++的最后说明……203

5.2 数据结构和类型……203
5.2.1 向量和点……203
5.2.2 参数化直线……204
5.2.3 3D 平面……206
5.2.4 矩阵……206
5.2.5 四元数……209
5.2.6 角坐标系支持……210
5.2.7 2D 极坐标……210
5.2.8 3D 柱面坐标……211
5.2.9 3D 球面坐标……211
5.2.10 定点数……212
5.3 数学常量……213
5.4 宏和内联函数……214
5.4.1 通用宏……218
5.4.2 点和向量宏……218
5.4.3 矩阵宏……219
5.4.4 四元数……220
5.4.5 定点数宏……221
5.5 函数原型……221
5.6 全局变量……224
5.7 数学引擎 API 清单……225
5.7.1 三角函数……225
5.7.2 坐标系支持函数……226
5.7.3 向量支持函数……228
5.7.4 矩阵支持函数……235
5.7.5 2D 和 3D 参数化直线支持函数……245
5.7.6 3D 平面支持函数……248
5.7.7 四元数支持函数……252
5.7.8 定点数支持函数……259
5.7.9 方程求解支持函数……263
5.8 浮点单元运算初步……265
5.8.1 FPU 体系结构……266
5.8.2 FPU 堆栈……266
5.8.3 FPU 指令集……268
5.8.4 经典指令格式……270
5.8.5 内存指令格式……271
5.8.6 寄存器指令格式……271
5.8.7 寄存器弹出指令格式……271
5.8.8 FPU 范例……271
5.8.9 FLD 范例……272
5.8.10 FST 范例……272
5.8.11 FADD 范例……273
5.8.12 FSUB 范例……275
5.8.13 FMUL 范例……276
5.8.14 FDIV 范例……278
5.9 数学引擎使用说明……279
5.10 关于数学优化的说明……280
5.11 总结……280

第 6 章 3D 图形学简介……282

6.1 3D 引擎原理……282
6.2 3D 游戏引擎的结构……282
6.2.1 3D 引擎……283
6.2.2 游戏引擎……283
6.2.3 输入系统和网络……284
6.2.4 动画系统……284
6.2.5 碰撞检测和导航系统……287
6.2.6 物理引擎……288
6.2.7 人工智能系统……289
6.2.8 3D 模型和图像数据库……289
6.3 3D 坐标系……291
6.3.1 模型（局部）坐标……291
6.3.2 世界坐标……293
6.3.3 相机坐标……296
6.3.4 有关相机坐标的说明……302
6.3.5 隐藏物体（面）消除和裁剪……303
6.3.6 透视坐标……308
6.3.7 流水线终点：屏幕坐标……315
6.4 基本的 3D 数据结构……321
6.4.1 表示 3D 多边形数据时需要考虑的问题……322
6.4.2 定义多边形……323
6.4.3 定义物体……327
6.4.4 表示世界……330
6.5 3D 工具……331
6.6 从外部加载数据……332
6.6.1 PLG 文件……333
6.6.2 NFF 文件……335
6.6.3 3D Studio 文件……338
6.6.4 Caligari COB 文件……343
6.6.5 Microsoft DirectX .X 文件……345
6.6.6 3D 文件格式小结……345
6.7 基本刚性变换和动画……345
6.7.1 3D 平移……345
6.7.2 3D 旋转……346
6.7.3 3D 变形……347
6.8 再看观察流水线……348
6.9 3D 引擎类型……349
6.9.1 太空引擎……349
6.9.2 地形引擎……350
6.9.3 FPS 室内引擎……351
6.9.4 光线投射和体素引擎……352
6.9.5 混合引擎……353
6.10 将各种功能集成到引擎中……353
6.11 总结……353

第 7 章 渲染 3D 线框世界……354

7.1 线框引擎的总体体系结构……354
7.1.1 数据结构和 3D 流水线……355
7.1.2 主多边形列表……357
7.1.3 新的软件模块……359
7.2 编写 3D 文件加载器……359
7.3 构建 3D 流水线……367

7.3.1 通用变换函数 ……367
7.3.2 局部坐标到世界坐标变换 ……372
7.3.3 欧拉相机模型 ……375
7.3.4 UVN 相机模型 ……377
7.3.5 世界坐标到相机坐标变换 ……387
7.3.6 物体剔除 ……390
7.3.7 背面消除 ……393
7.3.8 相机坐标到透视坐标变换 ……395
7.3.9 透视坐标到屏幕（视口）坐标变换 ……399
7.3.10 合并透视变换和屏幕变换 ……403
7.4 渲染 3D 世界 ……405
7.5 3D 演示程序 ……408
7.5.1 单个 3D 三角形 ……408
7.5.2 3D 线框立方体 ……411
7.5.3 消除了背面的 3D 线框立方体 ……413
7.5.4 3D 坦克演示程序 ……414
7.5.5 相机移动的 3D 坦克演示程序 ……416
7.5.6 战区漫步演示程序 ……418
7.6 总结 ……421

第三部分 基本 3D 渲染

第 8 章 基本光照和实体造型 ……424

8.1 计算机图形学的基本光照模型 ……424
8.1.1 颜色模型和材质 ……426
8.1.2 光源类型 ……432
8.2 三角形的光照计算和光栅化 ……437
8.2.1 为光照做准备 ……441
8.2.2 定义材质 ……442
8.2.3 定义光源 ……445
8.3 真实世界中的着色 ……449
8.3.1 16 位着色 ……449
8.3.2 8 位着色 ……450
8.3.3 一个健壮的用于 8 位模式的 RGB 模型 ……450
8.3.4 一个简化的用于 8 位模式的强度模型 ……453
8.3.5 固定着色 ……457
8.3.6 恒定着色 ……459
8.3.7 Gouraud 着色概述 ……472
8.3.8 Phong 着色概述 ……474
8.4 深度排序和画家算法 ……475
8.5 使用新的模型格式 ……479
8.5.1 分析器类 ……479
8.5.2 辅助函数 ……482
8.5.3 3D Studio MAX ASCII 格式.ASC ……484
8.5.4 TrueSpace ASCII.COB 格式 ……486
8.5.5 Quake II 二进制.MD2 格式概述 ……494
8.6 3D 建模工具简介 ……495
8.7 总结 ……497

第 9 章 插值着色技术和仿射纹理映射 ……498

9.1 新 T3D 引擎的特性 ……498
9.2 更新 T3D 数据结构和设计 ……499
9.2.1 新的#defines ……499
9.2.2 新增的数学结构 ……501
9.2.3 实用宏 ……502
9.2.4 添加表示 3D 网格数据的特性 ……503
9.2.5 更新物体结构和渲染列表结构 ……508
9.2.6 函数清单和原型 ……511
9.3 重新编写物体加载函数 ……517
9.3.1 更新.PLG/PLX 加载函数 ……517
9.3.2 更新 3D Studio .ASC 加载函数 ……527
9.3.3 更新 Caligari .COB 加载函数 ……528
9.4 回顾多边形的光栅化 ……532
9.4.1 三角形的光栅化 ……532
9.4.2 填充规则 ……535
9.4.3 裁剪 ……537
9.4.4 新的三角形渲染函数 ……538
9.4.5 优化 ……542
9.5 实现 Gouraud 着色处理 ……543
9.5.1 没有光照时的 Gouraud 着色 ……544
9.5.2 对使用 Gouraud Shader 的多边形执行光照计算 ……553
9.6 基本采样理论 ……560
9.6.1 一维空间中的采样 ……560
9.6.2 双线性插值 ……561
9.6.3 u 和 v 的插值 ……563
9.6.4 实现仿射纹理映射 ……564
9.7 更新光照/光栅化引擎以支持纹理 ……566
9.8 对 8 位和 16 位模式下优化策略的最后思考 ……571
9.8.1 查找表 ……571
9.8.2 网格的顶点结合性 ……572
9.8.3 存储计算结果 ……572
9.8.4 SIMD ……573
9.9 最后的演示程序 ……573
9.10 总结 ……576

第 10 章 3D 裁剪 ……577

10.1 裁剪简介 ……577
10.1.1 物体空间裁剪 ……577
10.1.2 图像空间裁剪 ……580
10.2 裁剪算法 ……581
10.2.1 有关裁剪的基本知识 ……581
10.2.2 Cohen-Sutherland 裁剪算法 ……585

10.2.3 Cyrus-Beck/梁友栋-Barsky 裁剪算法……586
10.2.4 Weiler-Atherton 裁剪算法……588
10.2.5 深入学习裁剪算法……590
10.3 实现视景体裁剪……591
10.3.1 几何流水线和数据结构……592
10.3.2 在引擎中加入裁剪功能……593
10.4 地形小议……611
10.4.1 地形生成函数……612
10.4.2 生成地形数据……619
10.4.3 沙地汽车演示程序……619
10.5 总结……623

第 11 章 深度缓存和可见性……624

11.1 深度缓存和可见性简介……624
11.2 z 缓存基础……626
11.2.1 z 缓存存在的问题……627
11.2.2 z 缓存范例……627
11.2.3 平面方程法……630
11.2.4 z 坐标插值……631
11.2.5 z 缓存中的问题和 1/z 缓存……632
11.2.6 一个通过插值计算 z 和 1/z 的例子……633
11.3 创建 z 缓存系统……635
11.4 可能的 z 缓存优化……649
11.4.1 使用更少的内存……649
11.4.2 降低清空 z 缓存的频率……650
11.4.3 混合 z 缓存……651
11.5 z 缓存存在的问题……651
11.6 软件和 z 缓存演示程序……652
11.6.1 演示程序 I：z 缓存可视化……652
11.6.2 演示程序 II：Wave Raider……653
11.7 总结……658

第四部分　高级 3D 渲染

第 12 章 高级纹理映射技术……660

12.1 纹理映射——第二波……660
12.2 新的光栅化函数……667
12.2.1 最终决定使用定点数……667
12.2.2 不使用 z 缓存的新光栅化函数……668
12.2.3 支持 z 缓存的新光栅化函数……670
12.3 使用 Gouruad 着色的纹理映射……671
12.4 透明度和 alpha 混合……677
12.4.1 使用查找表来进行 alpha 混合……678
12.4.2 在物体级支持 alpha 混合功能……688
12.4.3 在地形生成函数中加入 alpha 支持……694
12.5 透视修正纹理映射和 1/z 缓存……696
12.5.1 透视纹理映射的数学基础……696
12.5.2 在光栅化函数中加入 1/z 缓存功能……702
12.5.3 实现完美透视修正纹理映射……707
12.5.4 实现线性分段透视修正纹理映射……710
12.5.5 透视修正纹理映射的二次近似……714
12.5.6 使用混合方法优化纹理映射……718
12.6 双线性纹理滤波……719
12.7 Mipmapping 和三线性纹理滤波……724
12.7.1 傅立叶分析和走样简介……725
12.7.2 创建 Mip 纹理链……727
12.7.3 选择 mip 纹理……734
12.7.4 三线性滤波……739
12.8 多次渲染和纹理映射……740
12.9 使用单个函数来完成渲染工作……741
12.9.1 新的渲染场境……741
12.9.2 设置渲染场境……743
12.9.3 调用对渲染场境进行渲染的函数……745
12.10 总结……753

第 13 章 空间划分和可见性算法……754

13.1 新的游戏引擎模块……754
13.2 空间划分和可见面判定简介……754
13.3 二元空间划分……757
13.3.1 平行于坐标轴的二元空间划分……758
13.3.2 任意平面空间划分……759
13.3.3 使用多边形所在的平面来划分空间……760
13.3.4 显示/访问 BSP 树中的每个节点……762
13.3.5 BSP 树数据结构和支持函数……763
13.3.6 创建 BSP 树……765
13.3.7 分割策略……767
13.3.8 遍历和显示 BSP 树……775
13.3.9 将 BSP 树集成到图形流水线中……784
13.3.10 BSP 关卡编辑器……785
13.3.11 BSP 的局限性……793
13.3.12 使用 BSP 树的零重绘策略……794
13.3.13 将 BSP 树用于剔除……795
13.3.14 将 BSP 树用于碰撞检测……802
13.3.15 集成 BSP 树和标准渲染……802
13.4 潜在可见集……807
13.4.1 使用潜在可见集……808
13.4.2 潜在可见集的其他编码方法……809
13.4.3 流行的 PVS 计算方法……810

13.5 入口……811
13.6 包围体层次结构和八叉树……813
13.6.1 使用 BHV 树……815
13.6.2 运行性能……816
13.6.3 选择策略……817
13.6.4 实现 BHV……818
13.6.5 八叉树……825
13.7 遮掩剔除……825
13.7.1 遮掩体……826
13.7.2 选择遮掩物……826
13.7.3 混合型遮掩物选择方法……827
13.8 总结……827

第 14 章 阴影和光照映射……828

14.1 新的游戏引擎模块……828
14.2 概述……828
14.3 简化的阴影物理学……829
14.4 使用透视图像和广告牌来模拟阴影……832
14.4.1 编写支持透明功能的光栅化函数……833
14.4.2 新的库模块……835
14.4.3 简单阴影……837
14.4.4 缩放阴影……839
14.4.5 跟踪光源……841
14.4.6 有关模拟阴影的最后思考……844
14.5 平面网格阴影映射……845
14.5.1 计算投影变换……845
14.5.2 优化平面阴影……848
14.6 光照映射和面缓存技术简介……848
14.6.1 面缓存技术……850
14.6.2 生成光照图……850
14.6.3 实现光照映射函数……851
14.6.4 暗映射（dark mapping）……853
14.6.5 光照图特效……854
14.6.6 优化光照映射代码……854
14.7 整理思路……854
14.8 总结……854

第五部分 高级动画、物理建模和优化

第 15 章 3D 角色动画、运动和碰撞检测……858

15.1 新的游戏引擎模块……858
15.2 3D 动画简介……858
15.3 Quake II .MD2 文件格式……859
15.3.1 .MD2 文件头……861
15.3.2 加载 Quake II .MD2 文件……868
15.3.3 使用.MD2 文件实现动画……874
15.3.4 .MD2 演示程序……882
15.4 不基于角色的简单动画……883
15.4.1 旋转运动和平移运动……883
15.4.2 复杂的参数化曲线移动……885
15.4.3 使用脚本来实现运动……885
15.5 3D 碰撞检测……887
15.5.1 包围球和包围圆柱……887
15.5.2 使用数据结构来提高碰撞检测的速度……888
15.5.3 地形跟踪技术……889
15.6 总结……890

第 16 章 优化技术……891

16.1 优化技术简介……891
16.2 使用 Microsoft Visual C++和 Intel VTune 剖析代码……892
16.2.1 使用 Visual C++进行剖析……892
16.2.2 分析剖析数据……893
16.2.3 使用 VTune 进行优化……894
16.3 使用 Intel C++编译器……899
16.3.1 下载 Intel 的优化编译器……900
16.3.2 使用 Intel 编译器……900
16.3.3 使用编译器选项……901
16.3.4 手工为源文件选择编译器……901
16.3.5 优化策略……902
16.4 SIMD 编程初步……902
16.4.1 SIMD 基本体系结构……903
16.4.2 使用 SIMD……903
16.4.3 一个 SIMD 3D 向量类……912
16.5 通用优化技巧……918
16.5.1 技巧 1：消除_ftol()……918
16.5.2 技巧 2：设置 FPU 控制字……918
16.5.3 技巧 3：快速将浮点变量设置为零……919
16.5.4 技巧 4：快速计算平方根……919
16.5.5 技巧 5：分段线性反正切……920
16.5.6 技巧 6：指针递增运算……920
16.5.7 技巧 7：尽可能将 if 语句放在循环外面……921
16.5.8 技巧 8：支化（branching）流水线……921
16.5.9 技巧 9：数据对齐……921
16.5.10 技巧 10：将所有简短函数都声明为内联的……922
16.5.11 参考文献……922
16.6 总结……922