

国外游戏开发丛书

OpenGL

游戏程序设计

[美] K. 霍 金 著
D. 阿斯特 著
田昱川 译



科学出版社
www.sciencep.com

国外游戏开发丛书

OpenGL 游戏程序设计

(美) K.霍金 著
D.阿斯特
田昱川 译

科学出版社

北 京

图字:01-2003-2024

内 容 简 介

本书重点介绍了如何使用高性能的 OpenGL 图形与游戏函数库来开发游戏的视频部分。为了让读者了解创建一个游戏的全部过程,书中还包含了 DirectX 的内容。本书是涵盖这两方面内容的极少书籍之一。

在本书中,首先介绍了 OpenGL 和 DirectX 的产生背景,对它们的工作机理给出一个总体的描述,然后着重介绍了 OpenGL 最有可能应用于游戏开发的技术。再后,介绍了如何把所有的技术结合在一起,搭建一个自己的游戏框架,用 OpenGL 做图形系统部分,用 DirectSound 和 DirectInput 做声音和设备输入系统部分。最后,本书将介绍如何利用这个游戏框架和本书所涵盖的各种技术来创建一个完整的 3D 游戏。

本书是学习使用 OpenGL 进行 3D 游戏设计的一个很实用的资料,非常适合打算进行游戏开发的程序员使用,也适合其他 3D 应用程序开发的程序员使用。

OpenGL Game Programming

First published by Premier Press, a division of Thomson Learning.

All Rights Reserved.

Authorized Simplified Chinese Edition by Thomson Learning and Science Press. No part of this book may be reproduced in any form without the express written permission of Thomson Learning and Science Press.

本书简体中文版由 Thomson Learning 授权科学出版社出版,未经出版者书面允许不得以任何方式复制或抄袭本书全部或部分内容。

版权所有,翻印必究

图书在版编目(CIP)数据

OpenGL 游戏程序设计/(美)K.霍金,D.阿斯特著;田昱川译. —北京:科学出版社,2006

(国外游戏开发丛书)

ISBN 7-03-016616-7

I.O… II.①霍…②阿…③田…III.图形软件,OpenGL IV.TP391.41

中国版本图书馆 CIP 数据核字(2005)第 147167 号

责任编辑:韩洁 李伟/责任校对:柏连海

责任印制:吕春珉/封面设计:耕者设计工作室

科学出版社出版

北京东黄城根北街 16 号

邮政编码:100717

http://www.sciencep.com

科学彩色印装有限公司印刷

科学出版社发行 各地新华书店经销

*

2006 年 3 月第 一 版 开本:787×1092 1/16

2006 年 3 月第一次印刷 印张:35 1/2

印数:1—3 000 字数:741 000

定价:59.00 元(含光盘)

(如有印装质量问题,我社负责调换<环伟>)

销售部电话 010-62136131 编辑部电话 010-62138978-8001 (BI01)

前 言

3D 视频游戏是很有趣的。这些游戏所创造的具有沉浸感的虚拟世界对于人类的感知是一个挑战，而对这些游戏的开发也同样是一个具有挑战性的工作。在此，我要告诉你，现在是学习 3D 游戏编程的最佳时机。如果你想要开发 PC 游戏，或者仅仅是想了解一下游戏开发，那么现在正是时候，这本书就可以满足你。

以我的观点来看，如今的游戏开发最令人难以置信的是在拥有最新的图形卡计算机上的那些众多的 3D 图形功能和性能。在过去，游戏图形硬件经常是在走两个极端。一方面是来自一些专业公司，如 Silicon Graphics（我的前任雇主）的 UNIX 工作站所提供的图形功能，它们的价格只有在大的组织机构中的科学家、工程师和艺术家才能够承受。尽管这些工作站中存在着一些很酷的游戏，但是没有人会仅仅为了玩其中的游戏而购买它！另一方面，游戏控制平台以游戏玩家们可以接受的价格提供图形硬件，但是控制平台编程却不是一个开放的技术，只有专业的游戏开发人员才能接触到所需的开发环境。

与经济的游戏控制平台不同，高端的图形工作站非常昂贵，因此就需要它们的 3D 编程接口必须非常的高效和简单。Silicon Graphics 开发了一种高性能的，简单的编程接口，称为 IRIS GL，其中 GL 是 graphics library 的缩写。以此为基础，IRIS GL 逐渐改善发展成为一个 3D 编程接口——OpenGL，UNIX 工作站厂商都以此为标准。有成千上万的 3D 工程、科学、医学和动画应用程序使用 OpenGL。微软公司也在它的 NT 工作站操作系统中将 OpenGL 作为标准；后来，微软公司在 Windows 95 以及其后的操作系统中都添加了对 OpenGL 的支持。

OpenGL 最根本的目的是为硬件加速的 3D 绘制而设计的。在高端的 Silicon Graphics 工作站上，每一个 OpenGL 调用都是直接依赖于其执行的图形硬件。在 OpenGL 最初被移植到 PC 机上的时候，OpenGL 命令必须依靠 CPU 来处理，因此 OpenGL 并没有展现出它应有的速度优势。幸运的是，很快 PC 机的图形硬件也能够实现 OpenGL 了。如今，PC 机的图形硬件，诸如 NVIDIA 的 GeForce 图形芯片，能够在硬件中直接执行 OpenGL 命令。半导体技术的持续发展以及图形芯片设计人员的聪明才智，使得现在的 PC 图形芯片的速度已经是几年前最好的 Silicon Graphics 系统的好几倍了（更别提相比而言非常便宜的价格了！）。随着 PC 图形硬件的发展，OpenGL 已经集成了大量的诸如多纹理映射这样的针对游戏的扩展功能。

这样就产生了一个令人惊奇的结合体，专业 3D 应用程序的最好的 3D 编程接口，同时也是游戏的最佳 3D 编程接口。而且 OpenGL 不仅仅适用于 Windows PC，它现在还是 Mac 的标准 3D 编程接口，包括了 Apple 的 OS X 版本。OpenGL 还可用于所有的 UNIX 工作站，包括 Sun、HP 和 IBM，当然还有 Silicon Graphics。OpenGL 同时还是开放源代码软件运动的主力；Mesa 和原来的 Silicon Graphics OpenGL 样例实现版本提供了对 Linux PC 的 OpenGL 支持。如果游戏控制平台也接受 OpenGL 的话，我是不会感到奇怪的。

如果你很熟悉最新的 3D PC 游戏,我敢确信你一定听说过,并且很可能玩过以 OpenGL 为基础的游戏。其中比较著名的包括: id Software 出品的所有的 Quake 系列游戏和即将推出的 Doom 3; Valve 公司的 Half Life; 来自 Raven Software 的 Soldier of Fortune 和 Star Trek 的 Voyager-Elite Force; Dynamix 的 Tribes 2, 以及 Croteam 的 Serious Sam。所有这些游戏背后的开发人员一度都是编程新手,但是他们拥有学习 3D 游戏编程的满腔热情,并实现了他们的游戏。也许你也会将你的游戏添加到上面的名单中,谁知道呢?

本书介绍了如何用高性能的图形和游戏函数库来开发游戏。本书是作者多年工作的积累,它是一个丰富的资料库,能很好地引导读者去创作精彩的游戏。

本书展示如何使用 OpenGL 开发游戏的图形系统部分。由于 OpenGL 只是图形函数库,而游戏还需要声音、音乐以及输入设备等其他要素,我们就把 DirectX 也涵盖进了本书。

对于期望成为游戏开发者的人来说,本书会填补相关资料中的一些空白。基于 DirectX 的游戏编程的书籍已有很多,同样基于 OpenGL 的图形学的书籍也不少,但到目前为止,还没有一本书同时涵盖了这两方面内容。

本书涵盖了创建一个游戏的全部工作。当然,图形是任何 3D 游戏的基石,因此我们在本书的开始对 OpenGL 的应用做了介绍。此外,本书还涉及了建立一个 PC 游戏所需的其他关键技术:使用 DirectX Input 获取用户的输入、用 DirectX Audio 处理声音和音乐、载入纹理和 3D 角色模型文件、处理真实物体碰撞和相互作用的基本物理概念,以及整体游戏体系结构设计。

虽然我们会在本书中同时涵盖 OpenGL 和 DirectX 的内容,但并不打算把本书写成它们的完全参考手册。在本书中只会论述那些游戏开发和本书演示程序所需的部分。在书的末尾,将列出一些资料,通过它们能更全面深入地学习这些功能强大的函数库。

本书分为三篇。

在第一篇中,我们先为后面所讲述的内容打好基础。回顾了 OpenGL 和 DirectX 的产生背景和历史,对它们的工作机理给出了一个总体的描述。因为我们把 Windows 作为开发平台,所以在这一部分简述了 Windows 编程的要点。同时为了使读者做好与 3D 图形打交道的准备,我们还讲述了 3D 的理论基础。

第二篇,介绍 OpenGL,着重于那些最有可能应用于游戏的技术。这一部分的所有章节都有演示程序,它们向你展示 OpenGL 是如何工作的,同时也给读者提供了自己体验和创新的機會。

第三篇,我们介绍如何把所有的技术结合在一起搭建一个你自己的游戏框架,用 OpenGL 做图形系统部分,用 DirectSound 和 DirectInput 做声音和设备输入系统部分。最后,我们将会向你展示如何利用这个游戏框架和本书所涵盖的各种技术来创建一个 3D 游戏。

除了你将从本书看到的内容外,我们还随书赠送一些资料,其中包括了大量的演示程序、游戏和指南,它是本书的一个很好的补充。

作为作者,本书的出版令我非常激动。本书是学习使用 OpenGL 来进行 3D 游戏设计的一个很实用的资料,这对于那些有此打算的 PC 程序员正合适。

即使读者不是以开发游戏为生的,对于 OpenGL 和游戏开发的学习也将会有助于你

鉴赏所玩游戏背后的技术，对于计算机编程技术也是一个充实。OpenGL 不仅仅是针对游戏的，这一事实使得它同时也是开发其他的 3D 应用程序软件的非常有价值的编程接口。那么就让我们开始吧！

Mark J. Kilgard

NVIDIA 公司图形软件工程师

OpenGL Utility Toolkit（也被称为 GLUT）的开发者

目 录

第 1 篇 OpenGL 和 DirectX 的介绍

第 1 章 而今迈步从头越——OpenGL 和 DirectX	1
1.1 开发游戏需要理由吗	1
1.2 3D 游戏世界	1
1.3 游戏的元素	2
1.4 开发工具	4
1.5 什么是 OpenGL	5
1.5.1 OpenGL 的历史	6
1.5.2 OpenGL 的体系结构	6
1.5.3 OpenGL 实用库	7
1.5.4 什么是 GLUT	7
1.5.5 轻松一瞥	8
1.6 什么是 DirectX	8
1.6.1 DirectX 的历史	9
1.6.2 DirectX 体系结构	9
1.7 OpenGL 与 DirectX	11
本章小结	12
第 2 章 Windows 操作系统平台上的 OpenGL 编程	13
2.1 Windows 编程介绍	13
2.1.1 Windows 应用程序基础	15
2.1.2 从 WinMain()开始	15
2.1.3 Windows Procedure	16
2.1.4 窗口类	17
2.1.5 窗口类的注册	20
2.1.6 窗口的创建	21
2.1.7 消息循环	23
2.1.8 一个完整的 Windows 应用程序	24
2.2 WGL 介绍	28
2.2.1 绘制环境	28
2.2.2 WGL 的使用	29
2.3 像素格式	30
2.3.1 nSize	31
2.3.2 dwFlags	31

2.3.3	iPixelFormat	32
2.3.4	cColorBits	32
2.4	OpenGL Windows 应用程序	33
2.5	全屏 OpenGL	40
	本章小结	41
第 3 章	3D 图形学理论概览	43
3.1	标量、点和矢量	43
3.1.1	矢量的大小	44
3.1.2	矢量的归一化	44
3.1.3	矢量加法运算	44
3.1.4	矢量与标量的乘法运算	45
3.1.5	点积运算	45
3.1.6	叉积运算	45
3.2	矩阵	46
3.2.1	单位矩阵	46
3.2.2	零矩阵	47
3.2.3	矩阵加法和减法运算	47
3.2.4	矩阵乘法运算	47
3.2.5	矩阵的总体实现	49
3.3	变换	51
3.3.1	平移变换	51
3.3.2	旋转变换	52
3.3.3	缩放变换	53
3.4	投影	54
3.4.1	平行投影	55
3.4.2	透视投影	55
3.5	3D 剪裁	57
3.6	光照	58
3.6.1	环境光	58
3.6.2	散射光	58
3.6.3	镜面反射光	59
3.7	纹理映射	59
	本章小结	60

第 2 篇 使用 OpenGL

第 4 章	OpenGL 状态和图元	61
4.1	状态函数	61
4.2	图元的处理	67

4.2.1	3D 空间中点的绘制	68
4.2.2	3D 空间中线的绘制	70
4.2.3	3D 空间中的多边形的绘制	72
4.2.4	图元的使用	76
	本章小结	76
第 5 章	坐标变换和 OpenGL 矩阵	77
5.1	理解坐标变换	77
5.1.1	视点和眼坐标	78
5.1.2	视图变换	79
5.1.3	模型变换	82
5.1.4	投影变换	83
5.1.5	视区变换	84
5.2	OpenGL 和矩阵	84
5.2.1	模型视图矩阵	84
5.2.2	平移变换	85
5.2.3	旋转变换	86
5.2.4	缩放变换	86
5.2.5	矩阵堆栈	87
5.2.6	机器人例程	89
5.3	投影变换	99
5.3.1	正交投影变换	99
5.3.2	透视投影变换	100
5.3.3	视区的设置	101
5.3.4	投影变换实例	101
5.4	自定义矩阵的使用	103
5.4.1	自定义矩阵的载入	103
5.4.2	矩阵的乘法运算	104
5.4.3	自定义矩阵的实例	104
	本章小结	105
第 6 章	添加颜色、混合与光照	106
6.1	颜色是如何工作的	106
6.2	在 OpenGL 中使用颜色	107
6.2.1	颜色深度	107
6.2.2	颜色立方体模型	108
6.2.3	OpenGL 中的 RGBA 模式	108
6.2.4	OpenGL 中的颜色索引模式	109
6.3	明暗处理	110
6.4	OpenGL 中的光照	111
6.4.1	OpenGL 光照和现实世界	111

6.4.2 材质	112
6.4.3 法线	112
6.4.4 OpenGL 光照的使用	116
6.5 颜色混合	136
透明度	137
本章小结	142
第 7 章 位图、图像与 OpenGL	143
7.1 OpenGL 位图	143
7.1.1 位图的定位	144
7.1.2 位图的绘制	144
7.1.3 一个 OpenGL 位图例程	145
7.2 图像的使用	147
7.2.1 图像数据的绘制	147
7.2.2 屏幕的读取	148
7.2.3 屏幕数据的复制	149
7.2.4 放大、缩小和翻转	149
7.3 像素存储的管理	149
7.4 Windows 位图	150
7.4.1 BMP 文件格式	150
7.4.2 BMP 文件的载入	151
7.4.3 BMP 文件的输出	152
7.5 Targa 图像文件	154
7.5.1 Targa 文件格式	154
7.5.2 Targa 文件的载入	155
7.5.3 Targa 文件的输出	156
本章小结	158
第 8 章 纹理映射	160
8.1 纹理映射的概览	160
8.2 纹理图的应用	166
8.2.1 2D 纹理	166
8.2.2 1D 纹理	166
8.2.3 3D 纹理	167
8.3 纹理对象	167
8.3.1 纹理名字的生成	167
8.3.2 纹理对象的创建和使用	168
8.4 纹理的过滤处理	168
8.5 纹理模式	169
8.6 纹理坐标	169
8.6.1 重复和夹持纹理	170

8.6.2 mipmap 和细节层次	172
8.6.3 mipmap 的自动生成	173
8.7 例程——飘动的旗帜	173
8.7.1 例程的解释说明	173
8.7.2 例程的实现	174
8.8 例程——高程纹理地形	184
8.8.1 例程的解释说明	184
8.8.2 例程的实现	188
本章小结	196
第 9 章 高级纹理映射	197
9.1 多纹理映射	197
9.1.1 多纹理支持的验证	198
9.1.2 扩展函数的访问	199
9.1.3 纹理单元的建立	199
9.1.4 纹理坐标的设置（指定）	201
9.1.5 综合应用	202
9.2 环境映射	210
9.3 纹理矩阵	213
9.4 光照映射	215
9.5 多通道多纹理映射	221
本章小结	225
第 10 章 显示列表与顶点数组	226
10.1 显示列表	226
10.1.1 显示列表的创建	227
10.1.2 向显示列表中填充命令	227
10.1.3 显示列表的执行	228
10.1.4 显示列表的注意事项	229
10.1.5 显示列表的销毁	229
10.1.6 显示列表与纹理	230
10.1.7 例程：使用显示列表的机器人演示程序	231
10.2 顶点数组	232
10.2.1 顶点数组的启用	233
10.2.2 数组的使用	234
10.2.3 顶点数组与多纹理映射	236
10.2.4 数组的锁定	237
10.2.5 例程：重建地形演示程序	237
本章小结	240
第 11 章 文本的显示	241
11.1 位图字体	241

11.2 轮廓字体	245
11.3 纹理映射字体	248
本章小结	254
第 12 章 OpenGL 缓存	256
12.1 什么是 OpenGL 缓存	256
12.1.1 像素格式的设置	256
12.1.2 缓存的清理	259
12.2 颜色缓存	259
12.2.1 双缓存系统	260
12.2.2 立体缓存	260
12.3 深度缓存	261
12.3.1 深度比较程序	261
12.3.2 深度缓存的使用	261
12.4 模板缓存	271
12.5 累积缓存	279
本章小结	280
第 13 章 OpenGL 二次曲面	282
13.1 OpenGL 二次曲面的基础	282
13.1.1 绘制风格	283
13.1.2 法线	283
13.1.3 指向	283
13.1.4 纹理坐标	283
13.1.5 清理	284
13.2 圆盘	284
13.3 圆柱	285
13.4 球	286
13.5 例程：飞掠一个二次世界	287
本章小结	290
第 14 章 曲线与曲面	291
14.1 曲线与曲面的描述	291
14.1.1 参数方程	292
14.1.2 控制点与连续性	292
14.2 求值程序	293
14.3 曲面	297
14.4 NURBS	303
本章小结	307
第 15 章 特殊效果	309
15.1 标志板	309
15.2 粒子系统的使用	312

15.2.1	粒子	313
15.2.2	粒子系统	315
15.2.3	粒子系统管理器	318
15.2.4	实例	318
15.2.5	粒子系统的效果	321
15.2.6	例程：暴风雪	322
15.3	雾	326
15.3.1	OpenGL 的雾	326
15.3.2	体积雾	327
15.4	镜像	328
15.4.1	光照的镜像	328
15.4.2	深度缓存的处理	328
15.4.3	使用模板缓存来处理有限的平面	329
15.4.4	对于非常规镜像表面的处理	330
15.4.5	对任意指向的平面的处理	330
15.5	阴影	331
15.5.1	静态阴影	331
15.5.2	投射阴影	332
15.5.3	模板阴影体	334
15.5.4	其他的方法	335
15.5.5	例程：镜像与阴影	335
	本章小结	338

第 3 篇 建立一个游戏

第 16 章	DirectX 的使用：DirectInput	339
16.1	为什么要使用 DirectInput	339
16.1.1	Windows 消息	339
16.1.2	Win32	341
16.1.3	DirectInput	343
16.2	DirectInput 的初始化	344
16.3	DirectInput 的使用	345
16.3.1	设备的添加	346
16.3.2	输入的获取	353
16.3.3	关闭	354
16.4	动作映射	355
16.5	输入子系统的建立	355
16.6	输入系统例程	364
	本章小结	366

第 17 章 DirectX Audio 的使用	367
17.1 声音的基础	367
17.2 什么是 DirectX Audio	370
17.2.1 DirectX Audio 的功能部件	371
17.2.2 音频数据流	373
17.3 使用 DirectMusic 加载和演奏音频	374
17.3.1 初始化 COM	375
17.3.2 创建与初始化演奏对象	375
17.3.3 创建 loader 对象	376
17.3.4 载入一个 segment 对象	376
17.3.5 下载波段	377
17.3.6 演奏 segment	377
17.3.7 停止一个 segment 的播放	378
17.3.8 segment 是否正在播放	378
17.3.9 segment 循环的控制	379
17.3.10 清理	379
17.4 一个简单的例程	380
17.5 Audiopath 的使用	391
17.5.1 默认的 Audiopath	391
17.5.2 标准 Audiopath	392
17.5.3 在 Audiopath 上播放声音	392
17.5.4 从 Audiopath 获取对象	394
17.6 3D 声音	394
17.6.1 3D 声音坐标	395
17.6.2 感知	395
17.6.3 DirectSound 3D 缓存	395
17.6.4 3D 参数的设置	396
17.6.5 DirectSound 3D 听者	398
17.6.6 3D 音效文本例程	399
本章小结	412
第 18 章 3D 模型的处理	414
18.1 3D 模型文件格式	414
18.2 MD2 文件格式	415
18.2.1 MD2 的实现	417
18.2.2 MD2 的载入	421
18.2.3 MD2 的显示	425
18.2.4 添加纹理	427
18.2.5 模型的动画	429
18.2.6 开发一个 CMD2Model 类	434

18.2.7 模型动画的控制	445
18.3 最后的一点小资料	448
本章小结	452
第 19 章 OpenGL 物理建模	453
19.1 物理学的回顾	453
19.1.1 时间	453
19.1.2 距离、位移和位置	454
19.1.3 速度	456
19.1.4 加速度	457
19.1.5 作用力	458
19.1.6 动量	459
19.1.7 摩擦力	461
19.2 真实世界的模拟	463
19.2.1 任务的分解	464
19.2.2 定时	464
19.2.3 矢量	469
19.2.4 平面	474
19.2.5 物体	478
19.2.6 物体碰撞的处理	480
19.2.7 一个例程：空中曲棍球	486
本章小结	509
第 20 章 建立一个游戏引擎	510
20.1 SimpEngine 的设计	510
20.1.1 利用 CNode 管理数据	511
20.1.2 对象的处理：CObject	515
20.2 引擎核	520
20.2.1 输入系统	522
20.2.2 CEngine 类	523
20.2.3 游戏循环	524
20.2.4 输入的处理	525
20.2.5 SimpEngine	526
20.3 视点	527
20.4 场景	530
20.5 模型的添加	530
20.6 音频系统	531
20.7 粒子系统	533
本章小结	534
第 21 章 创建一个游戏：杀戮时刻	535
21.1 初始设计	535

21.2 游戏场景	535
21.3 敌人	538
21.4 火箭与爆炸	542
21.5 用户的交互	543
21.6 运行游戏	544
21.7 建立 EXE 可执行文件	544
本章小结	545

第 4 篇 附 录

附录 A 在线资源	546
A.1 游戏开发	546
A.2 OpenGL	547
A.3 DirectX	547
A.4 其他的资源	548
附录 B 随书所附资料的使用	549
B.1 用户界面	549
B.2 文件结构	549
B.3 系统要求	549
B.4 安装	549
B.5 其他相关问题与疑难解答信息	550
B.6 是否仍旧需要帮助	550

第1篇 OpenGL 和 DirectX 的介绍

第1章 而今迈步从头越——OpenGL 和 DirectX

在享用游戏开发这道大餐之前，需要对将要用到的工具有一个清楚的认识，它们来自何方，人们是如何设计他们的。整本书中，我们会用到两类应用程序编程接口（API）。OpenGL API 用来生成 3D 图形和其他可视内容，而 DirectX API 用来处理设备输入、声音和网络游戏功能。本章以介绍这两类应用程序编程接口的历史和设计的基本背景作为开始，简述了本书所要涉及的内容。知道了这些后，让我们开始演出吧！

本章我们将要讨论：

- 什么是游戏
- OpenGL 的基础知识
- DirectX 组件
- OpenGL 和 DirectX 的比较

1.1 开发游戏需要理由吗

过去十年来，互动娱乐有了飞跃式的发展。被认为是孩子们的玩具的计算机游戏已经开拓出数十亿美元的市场。最近几年其发展势头更是越来越猛，并且在可预见的未来这种势头还将继续下去。互动娱乐工业是一个蓬勃发展的产业，它及时地利用最新的计算机技术，促进了在诸如图形学和人工智能这样的领域中的研究，而这样的研究通常是需要广泛的合作才能完成，不是单个的厂商就能应付的。正是由于这巨大的动力和广阔的发展前景吸引了大量的人进入到这一行业，但什么是人们创作游戏的真正原因呢？

通过与整个游戏业中的形形色色的人士的交流，可以得知似乎只有一件事促使他们学习游戏开发艺术并在其中获得成功，那就是兴趣。游戏开发已经成为软件开发中较为有创造性的形式之一，最近几年发布的令人惊叹的游戏就是很好的证明。像 Valve Software 推出的半条命（Half Life）这样的游戏已经将单机游戏推向了无法再更进一步的顶峰。游戏开发者们转而希望创造他们自己的虚拟世界，成千上万的不同人，也可能是上百万的人，将可同时体验这个世界。游戏开发者们努力地迎接挑战，发现新的技术和领域。用一个独立游戏开发者——Michael Sikora 的话来说就是“那是一条不归路”这就是游戏创作的全部。

1.2 3D 游戏世界

大约十年前，id Software 推出了一个叫 Wolfenstein 3D 的小游戏，Wolfenstein 3D 以其具有实时的光线计算的 3D 图形和其所创造的具有沉浸感的世界成为游戏发展史上的