

Visual C++

游戏开发技术与实例

四维科技 丁 展 编著

赛车游戏 ●
入侵者游戏 ●
精灵碰撞检测 ●
网络台球游戏 ●
冒险屠宰场游戏 ●
网络五子棋游戏 ●
对战坦克大战游戏 ●
面包圈大战游戏 ●
游戏设计和图形图像常用算法 ●



● 赠送代码
● 俄罗斯方块
● 地形编辑器
● 弹力球游戏
● 超级马里奥游戏

TP312
16021)

Visual C++

游戏开发技术与实例

四维科技 丁 展 编著

RJ5223/04

北方工业大学图书馆



00582058



人民邮电出版社
POSTS & TELECOM PRESS

图书在版编目 (CIP) 数据

Visual C++游戏开发技术与实例 / 四维科技, 丁展编著. —北京: 人民邮电出版社, 2005.2
ISBN 7-115-13023-X

I. V... II. ①四...②丁... III. ①C 语言—程序设计②游戏—应用程序—程序设计
IV. TP312

中国版本图书馆 CIP 数据核字 (2005) 第 006703 号

内 容 提 要

本书是一本介绍利用 Visual C++进行游戏开发的书籍, 能够帮助读者快速掌握游戏开发环境、流程、关键制作技术和相关算法。书中主要涵盖了游戏架构设计、2D 游戏开发、网络游戏开发、3D 游戏开发、游戏设计和图形图像常用算法等方面的内容。

书中涉及的主要实例包括冒险屠宰场游戏、入侵者游戏、赛车游戏、网络五子棋游戏、网络台球游戏、对战坦克大战和面包圈大战游戏 (Donuts4) 等, 内容全面且丰富。

书中附带的光盘提供了每个实例的完整的源代码, 方便读者举一反三, 开发出适合自己的游戏项目。

本书适合于有一定 C/C++编程基础的游戏开发爱好者, 对于有经验的游戏开发人员也有较高的参考价值。

Visual C++游戏开发技术与实例

- ◆ 编 著 四维科技 丁 展
责任编辑 汤 倩
- ◆ 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 010-67132692
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: 33.5
字数: 926 千字 2005 年 2 月第 1 版
印数: 1-5 000 册 2005 年 2 月北京第 1 次印刷

ISBN 7-115-13023-X/TP · 4412

定价: 58.00 元 (附光盘)

本书如有印装质量问题, 请与本社联系 电话: (010)67129223



入侵者游戏



赛车游戏

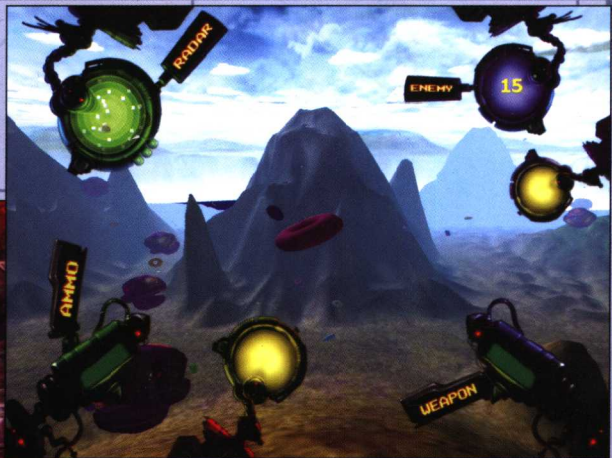


冒险屠宰场游戏



网络台球游戏

面包圈大战游戏
之战斗场景



面包圈大战游戏
之启动场景



前言 Foreword

电子游戏的出现代表了一种全新娱乐方式的诞生。目前，电子游戏正以其独特的魅力吸引着众多的玩家。随着游戏产业的蓬勃发展，越来越多的人加入到了游戏开发的行业。游戏开发涵盖了程序设计、文学、历史、音乐及美术等诸多方面的内容，因此，对开发者自身的要求较高。

► 为什么写本书

虽然已出版的游戏开发方面的书籍比较多，但大都没有将基础知识和实例进行有机的结合，要么只给出了游戏算法，要么缺少对实例核心代码的详细分析，给读者的学习带来不便。

本书主要介绍利用 Visual C++ 进行游戏开发的相关技术，并给出了几个综合性的游戏制作实例，便于读者真正了解游戏开发的各个环节，轻松掌握大型游戏中核心代码的编写方法。

► 本书特色

- ✓ 全书共分 6 章，层次清晰，以“游戏架构设计→2D 游戏开发→网络游戏开发→3D 游戏开发→游戏设计和图形图像常用算法”为主线，涉及的内容全面且丰富。
- ✓ 以剖析经典实例的方式引入知识点，引导读者边干边学，书中涉及的主要实例包括：冒险屠宰场游戏、入侵者游戏、赛车游戏、网络五子棋游戏、网络台球游戏、对战坦克大战和面包圈大战游戏（Donuts4）等。
- ✓ 光盘中均提供了每个实例的完整的源代码，方便读者举一反三，开发出适合自己的游戏项目。

► 读者对象

本书适合于有一定 C/C++ 编程基础的游戏开发爱好者阅读，对于有经验的游戏开发人员也有较高的参考价值。

► 技术支持

本书主要由丁展编写，参加写作的人员还有吕蕴华、鲁秋萍和汪杰等。在编写过程中，我们力求精益求精，但难免存在一些不足之处，如果读者使用本书时发现差错或者遇到问题，敬请批评指正，并请发 E-mail 至 ibookcauchy@zj.com 或 tangqian@ptpress.com.cn 与我们联系。

编 者
2005 年 1 月

光盘说明

CD-ROM

为了方便读者学习，本书附带了一张光盘，下面对光盘内容及使用方法进行简要的介绍。

一、光盘的运行环境

硬件环境：CPU 的主频在 500MHz 以上、内存在 128MB 以上。

软件平台：操作系统为 Windows 98/Me/NT/2000/XP（推荐使用 Windows 2000/XP），调试环境为 DirectX 9.0、Visual C++ 6.0 及其以上版本。

二、光盘目录

1. 2D 游戏开发

精灵碰撞检测.....	\\chap3\\cHitChecker_src
冒险屠宰场游戏.....	\\chap3\\AdvInAbatSrc
入侵者游戏.....	\\chap3\\Invasion_src
赛车游戏.....	\\chap3\\racex_src

2. 网络游戏开发

Winsock 聊天室	\\chap4\\ChatRoom(Winsock)
CSocket 聊天室	\\chap4\\ChatRoom(CSocket)
网络五子棋游戏.....	\\chap4\\网络五子棋
网络台球游戏.....	\\chap4\\Billiards8.4
赛车游戏.....	\\chap4\\racex_src
对战坦克大战游戏	\\chap4\\Tanker

3. 3D 游戏开发

网络台球游戏.....	\\chap5\\Billiards8.4
面包圈大战游戏.....	\\chap5\\Donuts4

4. 游戏设计和图形图像常用算法

迷宫	\\chap6\\maze
二十四点.....	\\chap6\\二十四点
水波模拟.....	\\chap6\\CartonRipple
图像淡入淡出渲染	\\chap6\\FadeIn

赠送代码

俄罗斯方块.....	Appendix\russia
地形编辑器.....	Appendix\fractal
超级马里奥游戏.....	Appendix\smb
弹力球游戏.....	Appendix\BreakBall
Simulate-quake 游戏	Appendix\squake

三、光盘的使用方法及注意事项

将本书的源代码拷入硬盘，去掉其只读属性（否则可能无法正常使用这些源程序）即可编译运行。下面简单介绍两类程序的编译调试过程。

1. Visual C++程序

下面以“面包圈大战”程序为例，说明如何使用 Visual C++程序。首先将光盘中的第 5 章程序“chap5”拷贝到硬盘中。运行 Visual C++ 6.0，单击菜单【File】→【Open Workspace...】，选择该目录下的“donuts4.dsw”，然后选择菜单【Build】→【Rebuild All】编译该程序。按快捷键【Ctrl+F5】运行程序，程序运行界面如下图所示。



2. DirectX SDK

本书的大部分实例采用了 DirectX 开发技术，为了正确编译，读者需要在本地计算机上安装 DirectX SDK（如果只安装了 DirectX Runtime 库，程序是无法编译通过的）。DirectX SDK 9.0 版本可以从微软的官方网站 www.microsoft.com 下载。

提示：每个程序文件夹里都有一个“readme”文件，将具体说明如何使用和运行程序。

目录 Contents

第 1 章 Windows 游戏编程	1
1.1 电子、电脑游戏的历史	1
1.1.1 从头谈起	1
1.1.2 图形硬件的革命	3
1.1.3 著名的游戏开发公司	5
1.1.4 著名的游戏设计大师	6
1.2 游戏软件开发工具	8
1.3 Windows 编程	9
1.3.1 Windows 消息机制	9
1.3.2 图形设备接口 (GDI)	12
1.3.3 定时器 (Timer)	27
1.3.4 回调机制	29
1.3.5 多线程	30
1.3.6 动态链接库 (DLL)	33
1.3.7 COM (Common Object Model)	35
1.3.8 声音	36
1.4 MFC 与游戏设计	42
第 2 章 游戏架构设计	43
2.1 游戏分类和常用术语	43
2.2 游戏策划	47
2.3 游戏模型	49
2.3.1 游戏的情感模型	50
2.3.2 游戏的行为系统模型	51
2.4 RPG 游戏的设计	52
2.5 美工与动画设计	56
2.6 音乐与音效设计	57
2.7 游戏引擎	58
第 3 章 2D 游戏开发	60
3.1 DirectDraw 游戏开发基础	60
3.1.1 DirectDraw 图形编程概述	61
3.1.2 DirectDraw 的架构	62

3.1.3 DirectDraw 的开发原理与方法	63
3.2 DirectInput 游戏开发基础	81
3.2.1 DirectInput 键盘	81
3.2.2 DirectInput 鼠标	84
3.3 等角贴片 (Isometric Tiles)	87
3.4 一个精确的 2D 碰撞检测算法	93
3.5 一个简单的精灵碰撞检测游戏	94
3.6 实例——冒险屠宰场游戏	107
3.6.1 系统框架设计	108
3.6.2 程序框架设计	109
3.7 实例——入侵者游戏	119
3.7.1 奖子的设计	120
3.7.2 子弹 (Bullet) 的设计	126
3.7.3 卷屏 (Scroll) 的设计	126
3.8 实例——赛车游戏的 2D 编程	139
3.8.1 cMain 游戏库	140
3.8.2 游戏类和元素	149
3.8.3 游戏的流程	160
第 4 章 网络游戏开发	179
4.1 TCP/IP	179
4.1.1 ISO 的 OSI 模型	179
4.1.2 TCP/IP 结构	180
4.2 SOCKET 编程	182
4.2.1 WINSOCK 的基本特点	182
4.2.2 WINSOCK 基本的 API	183
4.2.3 WINSOCK 异步模式	187
4.2.4 WINSOCK 网络程序实例——聊天室	188
4.2.5 CAsyncSocket 类	198
4.2.6 CSocket 类	200
4.2.7 CSocket 类网络应用实例——聊天室	201
4.3 小型网络游戏实例——网络五子棋	206
4.3.1 主程序结构和流程	206
4.3.2 实例演示	206
4.3.3 实例设计	207
4.3.4 代码详细分析	207
4.4 CSocket 线程安全性	222
4.5 服务器同步	223
4.6 航位推测法	224
4.7 用 DirectPlay 开发网络游戏	226

4.7.1	DirectPlay 的特点	227
4.7.2	DirectPlay 的开发过程和方法	229
4.7.3	DirectPlay 大厅的开发	236
4.8	实例——网络台球游戏的网络编程	238
4.8.1	刚体运动与碰撞	239
4.8.2	网络部分的代码实现	245
4.9	实例——赛车游戏的网络编程	249
4.9.1	cMultiplayer 类的声明	249
4.9.2	cMultiplayer 类的实现	252
4.10	实例——对战坦克大战	274
4.10.1	对战坦克大战的服务器程序	274
4.10.2	对战坦克大战的客户端程序	279
第 5 章	3D 游戏开发	286
5.1	2D 与 3D	286
5.2	Direct3D	287
5.2.1	绘制三角形	287
5.2.2	旋转立方体	292
5.2.3	深度缓冲	296
5.2.4	纹理	297
5.2.5	灯光与材质	301
5.2.6	索引缓冲	306
5.2.7	有纹理的圆柱体、锥体和球体	314
5.2.8	载入模型	322
5.3	OpenGL 3D 游戏开发基础	324
5.3.1	OpenGL 工作流程	325
5.3.2	OpenGL 开发基础	326
5.3.3	OpenGL 辅助库使用方法	329
5.3.4	OpenGL 建模	333
5.3.5	OpenGL 变换	336
5.3.6	OpenGL 光照	344
5.3.7	OpenGL 图像和位图	356
5.3.8	OpenGL 纹理映射	359
5.4	实例——网络台球游戏的 3D 编程	366
5.4.1	在 MFC 中使用 OpenGL	367
5.4.2	处理调色板	368
5.4.3	CBilliardsPlayView 视图类	370
5.5	实例——Donuts4 (面包圈大战游戏)	397
5.5.1	Donuts4 的游戏任务	402
5.5.2	Donuts4 的基本构成	403

5.5.3 Donuts4 的初始化	404
5.5.4 Donuts4 的运行过程	426
第 6 章 游戏设计和图形图像常用算法	445
6.1 启发式搜索和 A* 算法	445
6.2 在 A* 路径搜索中使用堆	451
6.3 迷宫算法	453
6.4 24 点算法	466
6.5 五子棋走棋算法	481
6.6 象棋走棋算法	484
6.7 运动模糊算法 (Motion Blur)	489
6.8 瀑布粒子系统	493
6.9 基于粒子引擎的简单点精灵	494
6.10 水波效果算法	498
6.11 火焰效果算法	502
6.12 像素混合技术	503
6.13 镜头眩光特效	504
6.14 图像合成技术	508
6.15 图像淡入淡出渲染	513
6.16 云的实现	516
6.17 图形学中的 3D 数学	520
6.17.1 向量	521
6.17.2 线和平面	522
6.17.3 齐次坐标	523
6.17.4 矩阵	523
参考文献	526

第 1 章 Windows 游戏编程

游戏发展至今已经有 30 多年,在这个短暂的时期里,随着硬件水平的提高,游戏开发新技术层出不穷,经典游戏比比皆是。

本章将介绍游戏的发展历史、Visual C++开发游戏的优势以及 Windows 游戏编程基础。

1.1 电子、电脑游戏的历史

1.1.1 从头谈起

真正的电子游戏机产生于 20 世纪 70 年代。1971 年,麻省理工学院的学生 Nolan Bushnell 设计了世界上第一个业务用游戏机(俗名街机),叫做《电脑空间》(Computer Space)。《电脑空间》的主题是两个玩家各自控制一艘太空战舰围绕着具有强大引力的星球向对方发射导弹进行攻击。两艘战舰在战斗的同时还必须注意克服引力,无论是被对方的导弹击中还是没有成功摆脱引力,飞船都会坠毁。这台游戏机用一台黑白电视机作为显示屏,用一个控制柄作为操纵器,不过由于市场因素这款游戏以失败告终。

虽然初次尝试失败,但 Nolan Bushnell 仍然坚信电子游戏的发展前景。他在《电脑空间》推出的次年,和朋友 Ted 用 500 美金注册成立了自己的公司 Atari(雅达利)。他们获得了成功,世界上第一台被接受的业务用机就是 Atari 推出的以乒乓球为题材的游戏。

1976 年 10 月,Atari 发行了名为《夜晚驾驶者》的模拟业务机游戏,这个游戏为黑白屏幕,自带了相关操纵器(方向盘、油门和刹车等)。在这个游戏里,玩家需要扮演一个黑夜里驾车在高速公路上狂奔的疯狂车手。这是游戏史上第一个 3D 游戏,它用简单的透视效果,专业术语是 LOD(Level of Detail,层次细节)来表现汽车的前进和道路景物的后退。这款游戏成为 3D 游戏的始祖。除此之外,它还是历史上第一个主视角(First Personal View)的游戏。

雅达利公司在开发街机市场之后又把目光转向了家用机市场。1977 年,雅达利公司推出了 Atari 2600 型游戏主机,这是世界上第一台家用专业游戏机。同年 Bushnell 把自己的公司以 2800 万美元的价格卖给了时代华纳。不久,任天堂和世嘉迅速崛起,彻底击败了雅达利。

上面介绍的是专用机游戏的历史,而最早的电脑游戏可以追溯到 1972 年,一个叫 Crowther 的工程师用当时最流行的主机——DEC 公司的 PDP-10 编写了一段简单的 FORTRAN 程序(FORTRAN 是一种专用工程数学编程语言,当今最流行的工程模拟软件 MATLAB 的早期版本就是由 FORTRAN 语言开发出来的)。在这个程序里,Crowther 设计了一张地图,地图上不规则的分布着陷阱,游戏者必须寻找路径避开陷阱。这个程序被公认为是最早的电脑游戏程序。1976 年,Don Woods 编写了一个类似的程序,更为重要的是他加入了幻想成分(Imagination)和谜题(Puzzle)。Woods 的这个程序直接导致了电脑游戏的诞生。他的程序被传播到各处,让所有的计算机高手们都为之惊喜。这个作品同时也启发了其他人的思维,紧接着,各式各样的游戏程序就诞生了。比较著名的有麻省理工学院的几个学生编写的 Zork(直到现在还有续集在发售)和亚当斯 1978 年编写的 Adventure Land(注,此 Adventure

Land《冒险地带》非 FC 上《高桥名人的冒险岛（美版）》Hudson's Adventure Island）。

后来用电脑编制电子游戏就开始在程序员之间流行起来。当时的电子游戏大多数还都是编程高手们做出来自娱自乐的“玩具”，这是因为当时接触计算机的还只是科技精英，普通人是接触不到的。说到这就不得不提苹果电脑，这个中国人眼里普遍认同的奢侈品。

Apple II 推出于 1976 年，设计者是史蒂夫·乔布斯（Steve Jobs），Apple II 是许多人的入门计算机，同时也是一台功能强大的真正的个人电脑。Apple II 和它内置的 Basic 程序为电脑界培养了无数的人才。Apple II 当时的内存只有 64KB，资源的缺乏使得当年的电脑程序精密严整得可怕。可以说，当年的程序员们才是一些真正懂得狂想乐趣的人，他们能够在一个匣子里完成一个世界。更重要的是，Apple II 代表了一种自由开发的精神，当年的电脑用户和程序员之间的差别是如此之小，可以说每一个 Apple II 的用户都是一个不错的 Basic 程序员。可以毫不夸张地说，苹果电脑造就了一大批游戏程序设计高手。而现在的家用电脑变得更像一台家用电器。

1989 年，Broderbund 公司的设计师乔丹·麦克纳（Jordan Mechner）根据阿拉伯民族的古老传说《一千零一夜》在 Apple 平台上制作了一部动作冒险相结合的电脑游戏——波斯王子。这个游戏获得了空前的成功，它在全球共卖出了 200 万份。“波斯王子”绝对可以说是电脑游戏动作冒险类（ACT）的第一作，它代表了当时电脑技术的最高水平。当年的玩家看到栩栩如生上窜下跳的王子都不禁惊叹不已，故事讲述的是王子为了拯救公主获得爱情而和罪恶的苏丹王斗争，在游戏中王子需要跑过各种奇怪诡秘的机关，小心地避过上下活动的柱子和深坑里的尖刺，用一把阿拉伯弯刀杀死各种敌人。

1983 年，日本的任天堂（Nintendo）和世嘉（Sega）分别推出了自己的家用游戏主机，世嘉推出的两台游戏主机名字叫做 SG1000 和 SG3000。而 Nintendo 推出的主机名字是 Family Computer（FC）。图 1.1 所示是 FC 的外观图。



图 1.1 任天堂的 FC

当时的 FC 采用 6502 芯片作为主 CPU，还有一块专门处理图像的 PPU。FC 可以显示 52 种颜色，全屏可以显示最多 13 种颜色，内存合计为 64KB，矩形波 2 音，三角波 1 音，杂音 1 音。同时任天堂采用了“以软件为主导”的指导思想，不断推出有趣的软件吸引玩家。1983 年底，FC 售出 44 万台，第二年伴随着一些先锋游戏公司分别加入 FC 制作者阵线，FC 销量大增。1984 年底 FC 总销售量达到 150 万台。

1985 年 9 月 13 日，任天堂公司发售了一款真正的游戏巨作——超级马里奥（Super Mario），这个

游戏讲述了一个意大利管子工打败魔王拯救世界迎娶公主的故事。任天堂凭借这款游戏确立了自己在游戏界霸主的地位。到了1985年,任天堂在家用机市场上的占有率高达98%。同年,任天堂向海外发售了FC的出口型NES(Nintendo Entertainment System),当年的销量就突破了500万台。图1.2所示是超级马里奥的动画原型。



图1.2 超级马里奥动画原型

20世纪80年代IBM PC(Personal Computer)兼容机的出现打破了Apple公司的垄断地位。虽然在最初几年里,PC上的软件数目还无法和Apple II抗衡,不过PC有一个最大的优势,那就是它是完全自由和开放的,所以更多的“穷”程序员可以选择PC机作为自己的开发平台而不是价格昂贵的Apple机。这时代比较著名的电脑游戏有Origin Systems公司的RPG游戏《创世纪》(Ultima)系列。光荣(Koei)公司推出了第一版《三国志》。当年的《三国志1》画面很简陋,不过到了后来《三国志4》系列开始大受欢迎,直到现在的《三国志9》。

到了20世纪90年代,游戏业才真正成熟起来,成为了一种产业。

由于PC机价格非常低而且硬件速度越来越快,游戏逐渐成为人们生活中不可缺少的一部分。游戏产业也逐渐发展成熟。这个时代诞生了许多史诗般的巨著:Quake(雷神之锤)、Doom(毁灭战士)、StarCraft(星际争霸)和FFT(最终幻想)等。

如图1.3所示是Quake中的一个游戏画面。如图1.4所示是Doom2中的游戏画面。

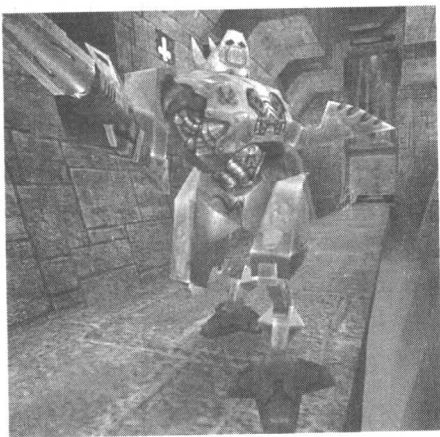


图1.3 Quake 游戏画面

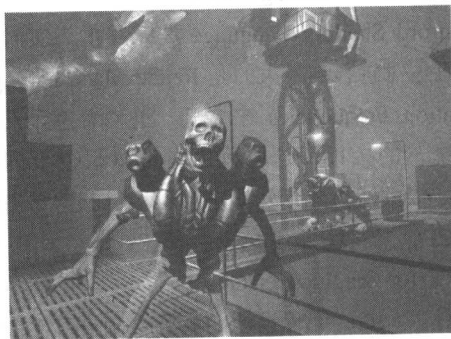


图1.4 Doom 游戏画面

1.1.2 图形硬件的革命

图形硬件的飞速发展是近些年的事情,部分原因是来自工业方面的压力,例如在军事和医疗方面对于实时图形的需求很强烈,而交互娱乐产业也极大地推动了图形硬件的发展。技术上的因素同样也推动着图形硬件的发展,许多图形算法可以很容易地表达为并行方式,这样硬件执行的效率变得很高。摩尔定律也起了作用,越来越多的晶体管可以集成到一块单独的芯片上。

在所谓的GPU(图形处理器)概念出现以前,特殊的图形硬件只出现在诸如SGI和E&S系统里面,这些硬件价格太昂贵,不过这些公司提供了第一代基于硬件的顶点变换和纹理映射的解决方案。

第一代消费级的GPU出现在1998年后期,主要代表为NVIDIA的TNT2,ATI的Rage和3DFX Voodoo3。这些处理器着重处理图形的光栅化操作,其中有些芯片支持多纹理,这意味着可以在光栅化过程中完成多幅纹理的融合操作,为程序开发者提供了一些固定的融合函数。这些处理芯片上大约

集成了 10×10^6 数量的晶体管。

在 1999 年后期和 2000 年初, 工业界开始正式出现 GPU 的概念, 即在 GPU 上集成了变换和光照计算, 从而减轻了 CPU 的负担。GPU 会自动的对送入的顶点矩阵进行变换和光照计算, 但这个时候还没有出现真正的可编程性。这一代典型的处理器有 NVIDIA GeForce256、GeForce 2、ATI 的 Radeon 7500 和 S3 的 Savage3D 等。这些处理器的处理芯片已经达到 25×10^6 数量的晶体管。

而第三代 GPU 将图形硬件的流水线作为流处理器来解释。顶点级出现了可编程性, 而在像素级出现有限的可编程性。顶点程序的指令数达到 128 条, 四元组参数达到 96 个。在像素级程序中, 访问纹理的方式和格式都受到一定限制, 依赖纹理严格受限, 只有定点数可用。另一个局限性就是缺乏程序流控制, 不允许分支, 最大程序尺寸也很小。这一代 GPU 另一些特征还包括支持阴影映射和 3D 纹理。主要代表有 NVIDIA 的 GeForce3、GeForce4 和 ATI 的 Radeon 8500。这些处理芯片达到 60×10^6 数量的晶体管, 采用 GPU 来做通用计算开始出现。

第四代 GPU 的像素和顶点可编程性更通用化, 可以包含上千条指令。依赖纹理更为灵活, 可以用作索引进行查找。GPU 现在具备了浮点功能, 纹理不再限制在 $[0,1]$ 范围, 从而可以用作任意数组, 这一点是对通用计算的另一个主要贡献。主要代表为 NVIDIA GeForce FX 系列和 ATI 的 Radeon 9700、9800, 集成的晶体管数达到 100M 的规模。

消费级图形硬件最大的成就就是图形流水线的可编程性, 这点是通过特定的程序来代替流水线的某些部分来实现的。这样就出现一个尖锐的问题, 如何来描述这些程序。汇编语言由硬件厂商提供, 可移植性是个问题, 而且不太好编程。不同的 GPU 的指令集不一样, 在一个 GPU 上可以编译的程序在另一个 GPU 上可能就不行。而且, 这个问题必须解决, 要做到与各厂商无关。

➤ Stanford Shading Language

Stanford Shading Language 是最早的为实时图形设计的高级绘制语言之一, 它借鉴了很多语言的思想, 但是对它影响最大的是 RenderMan Shading Language。Stanford Shading Language 围绕着不同的 Computation Frequencies 来展开, 过程着色元素可以跟体素结合起来, 顶点或者像素。绘制语言提供了一个统一的接口, 这样程序员可以组合不同的 frequencies。不同于其他语言, Stanford Shading Language 很关注将图形资源虚拟化, 这就意味着如果图形卡不能完成某项渲染程序指定的操作, 它就会采用 CPU 来实现。

➤ C for Graphics (CG)

CG 是为 GPU 编程设计的高级绘制语言, 由 NVIDIA 公司开发。CG 极力保留 C 语言的大部分语义, 并让开发者从硬件细节中解脱出来。CG 语言主要参照 ANSI C 建模, 但也从 C++ 和 Java 以及早期的绘制语言, 如 RenderMan and the Stanford Shading Language 中吸取了一些思想。这些使得很容易写程序然后由编译器进行优化, 提高了可读性。而且 CG 的设计考虑了 GPU 的体系结构, 如可编程多处理器单元 (顶点处理器, 像素处理器, 外加不可编程单元)。这些部分和应用都是通过数据流连接起来。CG 语言允许分别为顶点和像素写程序。

➤ Microsoft HLSL

DirectX 9.0 中一个最强有力的组件就是 HLSL (High Level Shading Language)。渲染器首次引入是在 DirectX 8 中的 Direct3D。那个时候, 定义了几个虚拟渲染机, 每一个大致对应一个领先的三维图形厂商的一个独特的处理器。对于每一个虚拟渲染机都定义了一套汇编语言。在 DirectX 8.0 和 DirectX 8.1 中, 这些渲染器模型支持的程序都很短小, 通常由开发者直接采用汇编语言来写。在 DirectX 9.0 中, 应用程序通过 D3DXCompileShader API 将 HLSL 渲染器传递给 D3DX, 得到一个二进制的表示然后传给 Direct3D。另外, 伴随着 D3DX 的 HLSL 编译器的发展, DirectX 9.0 也引入了另外的汇编级渲染器

模型 (vs_2_0, vs_3_0, ps_2_0 and ps_3_0) 来展现最新的三维图形硬件的功能。实际上 CG 和 HLSL 是同一类语言, 主要由于它是由 NVIDIA 和 Microsoft 联合开发的。HLSL 是 Microsoft DirectX API 的一部分, 只能编译成 DirectX 的代码, 而 CG 可以编译成 DirectX 和 OpenGL 的代码。如图 1.5 所示是 DirectX 的 LOG。

➤ OpenGL 2.0 shading language

OpenGL 2.0 首先由 3D Labs 开发的, 已提交 OpenGL ARB 评议 (注, 在本书创作期间, OpenGL 2.0 已经被官方正式发布), 其中包括了 shading language 规范。其语法同前面几种类似, 但是它包含在 OpenGL 标准中, 极有可能成为第一个跨平台的绘制语言。它跟其他语言最大的不同在于其编译器紧密集成在图形流水线中, 提供了一个更为易用的接口。如图 1.6 所示是 OpenGL 的 LOG。



图 1.5 DirectX LOG

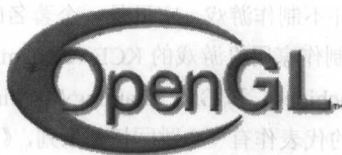


图 1.6 OpenGL LOG

1.1.3 著名的游戏开发公司

➤ 雅达利 (Atari)

游戏制作、开发的开山鼻祖, 现在只能在历史的档案中找到该公司的足迹。但是雅达利留下了许多成熟的开发技术, 这些开发技术至今还在应用, 甚至影响了计算机图形学中的某些分支。

➤ 任天堂 (Nintendo)

FC、超级马里奥 (Super Mario)、魂斗罗、最终幻想……这些响当当的名字足以震撼整个游戏界, 过多地介绍只会显得冗繁。

➤ 世嘉 (Sega)

世嘉和任天堂是同时起步的公司, 两家都专注于家庭机市场。很长一段时间内任天堂一直是市场的主宰, 直到《真人快打》的出现。那一年, 世嘉五代主机 (Sega Genesis) 上《真人快打》的销量远远超出了任天堂的合家欢版。北斗神拳、战斧、太空哈利等都是世嘉的名作。

➤ 光荣 (KOEI)

KOEI 最擅长于历史游戏。KOEI 有四大金牌系列游戏:《信长的野望》系列、《三国志》系列、《太阁立志传》系列和《无双》系列。除此之外,《大航海时代》系列也是 KOEI 的杰作。

➤ 西屋 (Westwood)

Westwood 是《沙丘》系列、《命令与征服》(C&C) 系列、《大地》(黑暗王座) 系列等最具盛名的经典游戏的缔造者。西屋现已被 EA (电子艺界) 收购。

➤ 暴雪 (Blizzard)

Blizzard 是一个以精品和跳票而闻名, 创造了《星际争霸》、《暗黑破坏神》、《魔兽争霸》等经典游戏的公司。如图 1.7 所示是暴雪的 LOG。

➤ 电子艺界 (EA, Electronic Arts)

“EA Sports! Enter the game!”。EA 在自己的每款体育类游戏的宣传片中都会有这个浑厚的男声。EA 的擅长项目就是体育类游戏。《FIFA》系列、《NBA》系列是 EA 的代表作。如图 1.8 所示是 EA 的 LOG。

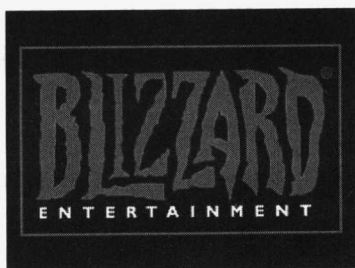


图 1.7 暴雪 LOG



图 1.8 EA LOG

➤ KONAMI

KONAMI 并不制作游戏，它只是一个著名的游戏厂商品牌、一个集团，而负责具体制作游戏的有 3 大部门：专门制作家用机游戏的 KCE (Konami Computer Entertainment)、专门制作商用游戏机的 AM (Amusement Machines) 和 GM (Gaming Machines)。

KONAMI 的代表作有《心跳回忆》系列，《幻想水浒传》系列，《合金装备》系列以及《实况足球》系列等。

➤ CAPCOM

《洛克人》、《龙之战士》、《吞食天地》以及《生化危机》都是 CAPCOM 的名作。

➤ id Software

id Software 是业界有名的小公司。id Software 成立十几年来，它的前任员工加上现任员工的数量还不到 20 人，就是这样一个小公司，创造了 3D 游戏的奇迹——《德军司令部 3D》，《毁灭战士》系列，《雷神之锤》系列是 id Software 的经典作品。

1.1.4 著名的游戏设计大师

在游戏设计界有一些令人敬仰的游戏设计大师，他们创造并推动了这个产业的发展。从某种意义上说，他们还在改变着我们的生活方式。更为重要的是，一大批游戏程序设计爱好者都是被这些设计大师的传奇经历和设计理念所打动，从而加入游戏开发的领域。

➤ John Carmack

John Carmack 在游戏界被尊称为 3D 游戏之父（见图 1.9）。作为整个游戏界最知名的程序设计师，Carmack 的设计技巧是超乎常人的。在业界里，Carmack 的编程技巧是所有程序员的典范。他是 Bill Gates 心目中为数不多的天才之一。Carmack 从来没有上过大学，他的编程技巧都是靠自学和钻研得来的。1990 年，年轻的 Carmack 便研究出了用 EGA（16 色的显示模式）使屏幕平滑滚动的方法，这在当时是不可思议的。同年，id Software 成立，发行了第一款游戏——《Wolfenstein 3D》，该游戏获得了空前的成功。它那优秀的不依靠任何硬件加速功能的帧速率和严密的迷宫设计无一不体现着 Carmack 的心血和技术。

他的专注造就了 Quake（《雷神竞技场》）和 Doom（《毁灭战士》）。

➤ John Romero

John Romero 号称王牌设计师（见图 1.10）。John Romero 的成功也是从 id Software 开始的。Romero 擅长于游戏策划、关卡设计和音效美术设计，而 Carmack 擅长于游戏引擎设计。这两个人造就了 Quake 和 Doom，游戏史上的巨著。