

数字化魅力·图像e时空丛书

1100101010011010101

1110101001010100101101001



OpenGL

三维图形系统 开发与实用技术

>>> (实用技术篇)

和平鸽工作室 编著

清华大学出版社
重庆大学出版社



数字化魅力·图像e时空丛书

OpenGL 三维图形系统开发与实用技术

(实用技术篇)

和平鸽工作室 编 著

清华大学出版社
重庆大学出版社

内 容 简 介

本书是“OpenGL 三维图形系统开发与实用技术”的实用技术分册。全书在基础编程分册的基础上,介绍了进行三维图形系统开发的一些实用技术,如外部三维模型的读入、地形模拟、粒子系统、LOD 技术、碰撞检测等,这些实用技术都是进行实际的三维图形系统开发所必需的。在本书的最后,介绍了 2 个 OpenGL 三维图形系统开发的综合实例。

本书深入浅出、内容广泛,既可供从事可视化系统、三维游戏、虚拟现实或其他图形应用程序开发的各大专院校学生、教师和研究人员参考,也可作为 OpenGL 三维图形编程的培训教程,或供其他相关专业人士和计算机爱好者阅读。

图书在版编目(CIP)数据

OpenGL 三维图形系统开发与实用技术. 实用技术篇/

和平鸽工作室编著. —重庆:重庆大学出版社,2003. 8

(数字化魅力·图像 e 时空丛书)

ISBN 7-5624-2951-0

I. O... II. 和... III. 图形软件,OpenGL IV. TP391.41

中国版本图书馆 CIP 数据核字(2003)第 055187 号

数字化魅力·图像 e 时空丛书

OpenGL 三维图形系统开发与实用技术

(实用技术篇)

和平鸽工作室 编 著

责任编辑:何 明 张 彬 谭 华 版式设计:陈 其
责任校对:蓝安梅 责任印制:秦 梅

*

清华大学出版社 出版发行
重庆大学出版社

出版人:张鸽盛

社址:重庆市沙坪坝正街 174 号重庆大学(A 区)内

邮编:400030

电话:(023) 65102378 65105781

传真:(023) 65103686 65105565

网址:<http://www.cqup.com.cn>

邮箱:fxk@cqup.com.cn (市场营销部)

全国新华书店经销

重庆科情印务有限公司印刷

*

开本:787×1092 1/16 印张:18.25 字数:455 千

2003 年 8 月第 1 版 2003 年 8 月第 1 次印刷

印数:1—5 000

ISBN 7-5624-2951-0/TP·424 定价:39.50 元(赠 1CD)

本书如有印刷、装订等质量问题,本社负责调换

版权所有 翻印必究

前言

OpenGL(即开放性图形库 Open Graphics Library)是近几年发展起来的一个性能卓越的三维图形标准,它源于 SGI 公司为其图形工作站开发的 IRIS GL,在跨平台移植过程中发展成为 OpenGL。SGI 在 1992 年 7 月发布 1.0 版,后来成为工业标准。1995 年 12 月批准了 1.1 版本,最新版规范是 1999 年 5 月通过的 1.2.1 版。OpenGL 作为一个性能优越的图形应用程序设计界面(API),具有广泛的可移植性,它独立于硬件系统、操作系统和窗口系统。OpenGL 适用于广泛的计算机环境,从个人计算机到工作站和超级计算机,用户都可以利用 OpenGL 创建漂亮的三维图形。由于许多在计算机界具有领导地位的计算机公司纷纷采用 OpenGL 作为三维图形应用程序设计界面,因此,OpenGL 是从事三维图形开发工作的技术人员所必须掌握的开发工具。

全书分基础编程篇和实用技术篇两册,主要介绍如何利用 OpenGL 开发三维图形应用系统。基础编程篇对 OpenGL 的体系结构和基础知识进行了详细的介绍,实用技术篇着眼于实际的工程应用,介绍了一系列的实用技术。

《OpenGL 三维图形系统开发与实用技术(基础编程篇)》共由 8 章组成,全面系统地介绍了进行 OpenGL 编程所需的基础知识,如三维基本图元的绘制、坐标变换、颜色、光照、纹理映射、动画和用户交互技术等。在附录中列出了所有的 OpenGL 库函数,便于读者在学习过程中查阅。

《OpenGL 三维图形系统开发与实用技术(实用技术篇)》共由 10 章组成,介绍了一些具体的利用 OpenGL 进行三维图形系统开发的实用技术,如外部三维模型数据的读入、真实感地形的模拟、自然环境的模拟、碰撞检测、LOD 技术和 OpenGL 图形系统后处理技术等。在最后两章中介绍了两个 OpenGL 开发的综合实例,一个是三维台球游戏,另一个是俄罗斯方块,读者可以分析其中的代码,提高图形系统的开发能力。

为了便于读者使用,在书后均附有包含该书全部应用实例的源代码光盘。全部程序在 Visual C++6.0 中编译通过,运行环境为 Windows 98 和 Windows 2000。读者可以充分利用其中的代码,有些只需稍许改动便可应用到自己的系统中。

拓智文化 CG 实验室网站 www.cglab.info 上有关于本书和 OpenGL 的精彩内容版块,读者可以访问

这个网站来探讨交流有关问题!

本书由和平鸽工作室全体成员齐心协力编写完成,是大家精诚团结的结果。感谢对本书的出版给予大力支持的重庆拓智文化发展有限公司和重庆大学出版社,以及其他提供支持的所有老师、朋友。

和平鸽工作室

2003 年 3 月

目 录

第 1 章 外部三维模型的读取与绘制	1
1.1 3DS 模型的读取与绘制	2
1.1.1 3DS 文件格式简介	2
1.1.2 3DS 编辑程序块	2
1.1.3 3DS 模型的读取与绘制实例	8
1.2 OBJ 模型的读取与绘制	27
1.2.1 OBJ 文件格式简介	27
1.2.2 OBJ 模型的读取与绘制实例	32
1.3 ASE 模型的读取与绘制	44
1.3.1 ASE 文件格式简介	44
1.3.2 ASE 模型的读取与绘制	47
1.4 MD2/MD3 模型的读取与绘制	66
1.4.1 MD2/MD3 文件格式介绍	66
1.4.2 MD2 模型的读取与绘制	67
1.4.3 MD3 模型的读取与绘制	79
1.5 MS3D 模型的读取与绘制	99
1.5.1 MS3D 文件格式简介	99
1.5.2 MS3D 模型的读取与绘制	101
1.6 转换 3DS 模型为 OpenGL 源文件	112
1.6.1 3DS 模型转换	112
1.6.2 转换模型的读取与显示	113
第 2 章 三维地形的模拟	115
2.1 地形模拟概述	116
2.2 地形模拟实例介绍	117
2.2.1 地形数据的获得与地形的渲染	117
2.2.2 地形漫游的实现	120
2.2.3 天空的绘制	126
2.2.4 场景绘制	129
第 3 章 粒子系统编程技术	132
3.1 粒子系统概述	133
3.1.1 过程模型	133

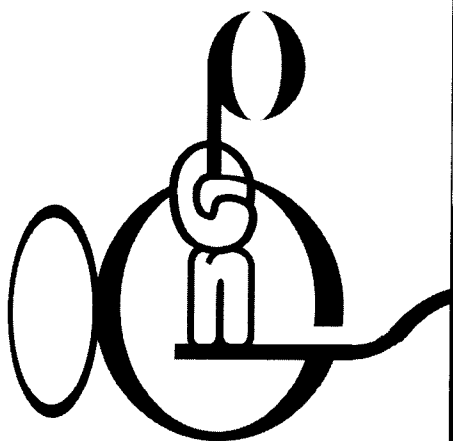
3.1.2	粒子系统	133
3.2	爆发的喷泉实例介绍	134
3.2.1	粒子属性的定义	135
3.2.2	粒子的生成	135
3.2.3	粒子的运动	136
3.2.4	粒子的死亡	137
3.2.5	粒子的绘制	138
3.2.6	场景绘制	139
3.3	五色的彩虹实例介绍	141
3.3.1	粒子属性的定义	141
3.3.2	粒子的纹理定义	143
3.3.3	粒子与场景的初始化	144
3.3.4	粒子与场景的绘制	145
3.3.5	键盘交互	148
第4章	碰撞检测技术编程	150
4.1	碰撞检测概述	151
4.2	碰撞检测实例介绍	152
4.2.1	各数据结构的定义	152
4.2.2	场景的建模	155
4.2.3	碰撞检测	158
4.2.4	场景绘制	161
4.2.5	用户交互	165
第5章	自然现象的模拟	167
5.1	模拟方法概述	168
5.2	动态云彩的模拟	168
5.2.1	云彩纹理图像的获得	169
5.2.2	纹理参数的定义	170
5.2.3	动态云彩的显示	171
5.3	移动雾效的模拟	173
5.3.1	雾效图像的获得	173
5.3.2	纹理参数的定义	174
5.3.3	场景的绘制	175
5.4	海面波浪的模拟	177
5.4.1	海面纹理图像的获取	177
5.4.2	纹理参数的定义	178
5.4.3	海面波浪的绘制	179
5.4.4	用户交互	181
第6章	LOD 技术编程	183
6.1	LOD 技术概述	184

6.2	LOD 编程实例一	187
6.2.1	球体的数据结构与球体的纹理	187
6.2.2	球体的绘制	188
6.2.3	与用户交互	190
6.3	LOD 编程实例二	192
6.3.1	获取地形高程数据	192
6.3.2	设置地形的层次结构	194
6.3.3	LOD 地形的绘制	195
6.3.4	与用户交互	195
第 7 章	高级界面编程技术	198
7.1	OpenGL 界面编程实例	199
7.1.1	菜单项数据结构的定义	199
7.1.2	菜单项的初始化	199
7.1.3	菜单的绘制	200
7.1.4	菜单项的响应	203
第 8 章	OpenGL 三维图形后处理技术	205
8.1	屏幕图形的打印	206
8.1.1	DDB 转换为 DIB	206
8.1.2	将屏幕位图转换成 DIB	209
8.1.3	屏幕图形打印预览	210
8.1.4	实现应用程序打印预览	211
8.1.5	实现应用程序打印	213
8.2	屏幕图形的保存	214
8.2.1	将 DIB 数据写入到文件	215
8.2.2	实现保存功能	216
8.3	屏幕动画的录制	217
8.3.1	写入 AVI 文件的头信息	217
8.3.2	写入 AVI 文件的帧信息	220
8.3.3	实现应用程序动画录制	221
第 9 章	综合开发实例一——三维台球	225
9.1	游戏简介	226
9.2	编程思想	226
9.3	技术基础	227
9.3.1	向量与矩阵操作基础	227
9.3.2	读入 TGA 纹理数据	232
9.4	关键技术	235
9.4.1	基本数据结构的定义	235
9.4.2	场景的初始化	237
9.4.3	静态场景的绘制	240

9.4.4	动态对象的绘制	241
9.4.5	场景的绘制	242
9.4.6	鼠标交互方式	243
9.4.7	键盘交互方式	246
9.4.8	游戏参数的初始化	248
9.4.9	力的计算	250
9.4.10	计算方法	252
9.4.11	碰撞检测	253
9.4.12	碰撞处理	256
9.4.13	游戏模拟	257
第 10 章	综合开发实例二——俄罗斯方块	261
10.1	游戏简介	262
10.2	编程思想	263
10.3	编程技术	263
10.3.1	屏幕文本的绘制	263
10.3.2	基本数据结构的定义	264
10.3.3	场景初始化	265
10.3.4	游戏封面的绘制	266
10.3.5	游戏主界面的绘制	268
10.3.6	新方块的产生	272
10.3.7	游戏进度的控制	276
10.3.8	下落方块的控制	279
10.3.9	设计菜单	282

外部三维模型的读取与绘制

第1章



在实际的三维图形程序中,需要构建的三维模型一般来说都是比较复杂的,但是 OpenGL 的建模功能并不是很强大,只提供了一些绘制简单图元的函数,如圆锥、圆柱和球体等类似的简单物体。如果利用这些函数来构建一个复杂的模型,如一架飞机或一辆坦克就显得非常困难,另外一次性地使用基本的绘图语句编写程序也不现实。因此,必须与其他功能强大的三维建模软件结合,利用一定的算法,将其他三维建模软件创建的模型数据读出来,然后利用这些数据在 OpenGL 环境中重新构建三维模型,从中可以免去构建复杂三维模型这一枯燥复杂的工作。

1.1 3DS 模型的读取与绘制

3D Studio 是 Autodesk 公司开发的一套用于在微机上制作三维动画的应用程序,它所生成的图形文件格式是 3DS 文件格式。该软件的功能非常强大,许多精美的电视广告都是用它制作的,现在已经发展到了 3D Studio MAX 5.x。由于 3DS 是一种非常普遍的数据格式,以 3DS 格式保存的三维图形文件非常丰富。各种三维图形素材光盘以及内容丰富的网上图形站点,都有非常丰富的 3DS 模型库。在微机上直接利用 3DS Studio 软件制作 3DS 格式的三维图形文件也非常容易。因此,学会读取和操作 3DS 文件对于建立比较实用的三维图形应用软件显得非常重要。

1.1.1 3DS 文件格式简介

3DS 文件由许多块组成,每个块首先描述其信息类别,即该块是如何组成的。块的信息类别用 ID 来标识,块还包含了下一个块的相对位置信息。因此,即使不理解一个块的含义,也可以很容易地跳过它,因为该块中指出了下一个块相对于该块起始位置的偏移字节数。与许多文件格式一样,3DS 二进制文件中的数据也是按低位在前、高位在后的方式组织的。例如,2 个十六进制字节 4A 5C 组成的整型数,表明 5C 是高位字节,4A 是低位字节;对于长整型数,如 4A 5C 3B 8F,表明 5C 4A 是低位字节,而 8F 3B 是高位字节。

下面描述块的具体定义。块的前两项信息分别是:块的 ID 和块的长度(也即下一个块相对于该块的字节偏移量),块的 ID 是一个整型数,而块的长度是一个长整型数。每个块实际上是一个层次结构,不同类型的块,其层次结构也不相同。3DS 文件中有一个基本块,其 ID 是 4D 4D,每一个 3DS 文件的开头都是由这样一个块构成。基本块内的块称为主块。

为了对块的层次结构有一个初步的认识,下面给出一个图表来说明不同类型(ID)的块及其各自在文件中的位置,如图 1.1 所示。这样为每个块起了一个名字,以便于理解或直接转换为源程序。

1.1.2 3DS 编辑程序块

最开始出现的主块是基本块,包含了整个文件。因此这个块的大小就是文件的大小减去主块的大小。还有两种主块:3D 编辑程序块和关键帧块,前者的 ID 是

3D3D, 后者的 ID 是 B000。3D 编辑程序块表明编辑程序数据开始, 也就是说物体的形体数据定义从此处开始。关键帧块表明即将开始定义关键帧信息。

下面介绍 3DS 格式文件中的编辑程序块。

• 3D(编辑程序块)的子块

ID	含义
0100	配置的一部分
1100	未知
1200	背景色
1201	未知
1300	未知
1400	未知
1420	未知
1450	未知
1500	未知
2100	环境颜色块
2200	雾
2210	雾
2300	未知
3000	未知
3D3E	编辑程序配置主块
4000	一个物体的定义
FFFF	材质列表开始

• AFFF(材质列表块)的子块

ID	材质名称
A000	用一个零结尾的字符串定义

• 3D3E(编辑程序配置块)的子块

ID	定义
7001	视口指示器
7011	视口定义(类型 2)
7012	视口定义(类型 1)
7020	视口定义(类型 3)

3D3E 块中包含了许多冗余数据, 其中较重要的块是 7020 块, 这个块定义编辑程序中 4 个活动的视口。假设在编辑

```

MAIN3DS (0x4D+D)
+--EDIT3DS (0x3D3D)
+--EDIT_MATERIAL (0xAFFF)
+--MAT_NAME01 (0xA000) (See mli Doe)
+--EDIT_CONFIG1 (0x0100)
+--EDIT_CONFIG2 (0x3E3D)
+--EDIT_VIEW_P1 (0x7012)
+--TOP (0x0001)
+--BOTTOM (0x0002)
+--LEFT (0x0003)
+--RIGHT (0x0004)
+--FRONT (0x0005)
+--BACK (0x0006)
+--USER (0x0007)
+--CAMERA (0xFFFF)
+--LIGHT (0x0009)
+--DISABLED (0x00010)
+--BOGUS (0x00011)
+--EDIT_VIEW_P2 (0x00001)
+--TOP (0x0001)
+--BOTTOM (0x0002)
+--LEFT (0x0003)
+--RIGHT (0x0004)
+--FRONT (0x0005)
+--BACK (0x0006)
+--USER (0x0007)
+--CAMERA (0xFFFF)
+--LIGHT (0x0009)
+--DISABLED (0x00010)
+--BOGUS (0x00011)
+--EDIT_VIEW_P3 (0x7020)
+--EDIT_VIEW1 (0x7001)
+--EDIT_BACKGR (0x1200)
+--EDIT_AMBIENT (0x2100)
+--EDIT_OBJECT (0x4000)
+--OBJ_TRIMESH (0x4100)
+--TRI_VERTEXL (0x4110)
+--TRI_VERTEXOPTIONS (0x4111)
+--TRI_MAPPINGCOORDS (0x4140)
+--TRI_MAPPINGSTANDARD (0x4170)
+--TRI_FACELI (0x4120)
+--TRI_SMOOTH (0x4150)
+--TRI_MATERIAL (0x4130)
+--TRI_LOCAL (0x4160)
+--TRI_VISIBLE (0x4165)
+--OBJ_LIGHT (0x4600)
+--LIT_OFF (0x4620)
+--LIT_SPOT (0x4610)
+--LIT_UNKNWN01 (0x465A)
+--OBJ_CAMERA (0x4700)
+--CAM_UNKNWN01 (0x4710)
+--CAM_UNKNWN02 (0x4720)
+--OBJ_UNKNWN01 (0x4710)
+--OBJ_UNKNWN02 (0x4720)
+--EDIT_UNKNW01 (0x1100)
+--EDIT_UNKNW02 (0x1201)
+--EDIT_UNKNW03 (0x1300)
+--EDIT_UNKNW04 (0x1400)
+--EDIT_UNKNW05 (0x1420)
+--EDIT_UNKNW06 (0x1450)
+--EDIT_UNKNW07 (0x1500)
+--EDIT_UNKNW08 (0x2200)
+--EDIT_UNKNW09 (0x2201)
+--EDIT_UNKNW010 (0x2210)
+--EDIT_UNKNW011 (0x2300)
+--EDIT_UNKNW012 (0x2302)
+--EDIT_UNKNW013 (0x2000)
+--EDIT_UNKNW014 (0xAFFF)
+--KEYF3DS (0xB000)

```

图 1.1 3DS 文件结构

程序布局中使用了 4 个视口,编辑程序配置中包含 5 个 7020 视口定义块和 5 个 7011 视口定义块。但事实上只有开始的 4 个 7020 块对用户的视口外观有影响,其他视口中只包含一些附加信息。该块的第 6,7 字节表明视图的类型,合法的 ID 及其对应的视图如下所示:

ID	含义
0001	顶视图
0002	底视图
0003	左视图
0004	右视图
0005	前视图
0006	后视图
0007	用户视图
FFFF	相机视图
0009	光源视图
0010	无效

• 4000(物体描述块)的子块

第一项是物体名称,为一个零结尾串,这里所指的物体可以是一个相机或一个光源,也可以是物体的形体。

ID	含义
4010	未知
4012	阴影块
4100	三角形列表块
4600	光源块
4700	相机块

• 4100(三角形列表)的子块

ID	含义
4110	顶点列表
4111	顶点选项
4120	面列表
4130	面材质
4140	纹理映射坐标
4150	面平滑组
4160	平移矩阵
4165	物体可见性
4170	标准映射

• 4110(顶点列表块)的子块

起始位置	结束位置	大小	数据类型	名称
0	1	2	无符号整数	物体的顶点数目
2	5	4	浮点数	X 坐标值
6	9	4	浮点数	Y 坐标值
10	13	4	浮点数	Z 坐标值

从第 2 个字节到第 13 个字节定义了一个顶点的坐标,重复这项定义 VertexNum

次,便得到了所有顶点的坐标。

●4111(顶点选项块)的子块

该子块由一些整型数组成,第一个整型数表明顶点个数,然后对每个顶点用一个整型数表示一些位信息。其中,0~7位和11~12位影响物体的可见性,8~10位是随机信息,13~15位表明该顶点是否在某个选择集中被选中。顶点选项块不是很重要,即使将其删除,3DS也能正确地将物体装入。

●4120面列表块

起始位置	结束位置	大小	数据类型	名称
0	1	2	无符号整数	物体中的三角形数 (PolyNum)
2	3	2	无符号整数	顶点 A 的序号
4	5	2	无符号整数	顶点 B 的序号
6	7	2	无符号整数	顶点 C 的序号
8	9	2	无符号整数	面信息

重复2~9定义PolyNum次,就给出了所有的三角形。2~7所给出的3个整型数是三角形面的3个顶点序号,序号为0的顶点表示顶点列表中定义的第一个顶点。顶点的顺序影响着面的法向量方向。一般情况下三角形应按逆时针方向定义,但有些3DS文件使用顺时针方向,这时就需要在程序中将其调整过来。面信息是一个整型数,其中前3个二进制位给出了三角形每条边的顺序,可以根据它们判别三角形是以逆时针还是顺时针顺序给出的。这3个数字要么全是0,要么全是1。若这3位全为1,其二进制方式为111,意味着三角形的3个顶点的正确顺序应是A→C→B,即A→B→C给出的是顺时针方向。反之,这3位全是0,则意味着A→B→C给出的就是逆时针方向。

下面给出面信息中每一位的具体含义:

bit0	AC边顺序,	1:A→C	0:C→A
bit1	BC边顺序,	1:B→C	0:C→B
bit2	AB边顺序,	1:A→B	0:B→A
bit3	纹理映射(如果有)		
bit4~8	保留(0)		
bit9~10	随机数		
bit11~12	保留(0)		
bit13	面在第3选择集中被选中		
bit14	面在第3选择集中被选中		
bit15	面在第3选择集中被选中		

●4130面材质块

材质块给出了物体中使用的每一种材质,但并不是每个物体都有材质块,只使用默认材质的物体就没有材质块。每一个4130面材质块以一个0结尾的材质名称字符串开始,接着跟一个由数字表示的与该材质相关的面的数目,然后就是一个一个的面。0000表示4120面列表中的第一个面。

• 4140 纹理映射坐标

前 2 个字节表示定义的顶点数,然后为每个顶点定义 2 个浮点型的纹理映射坐标。也就是说,如果一个顶点的纹理映射在纹理平面的中心,那么其映射坐标为(0.5,0.5)。

• 4150 面平滑组块

块的大小为面的个数 $\times 4B$,即每个基本数据是一个长整型数字,第 n 个数字表示该面是否属于第 n 个平滑组。

• 4160 局部坐标轴块

开始的 3 个浮点数定义了物体的局部坐标轴在绝对坐标系内的位置坐标,最后的 3 个浮点数是物体的局部中心。

• 4170 标准映射

前 2 个字节表明映射的类型,其中,0 表明平面映射或指定映射,当指定映射时,与这个块中的信息就无关了,1 表明圆柱映射,2 表明球映射。接着用 21 个浮点数来描述这个映射。

• 光源块

ID	含义
0010	RGB 颜色块
0011	24 位颜色
4610	点光源
4620	光源被关闭

起始位置	结束位置	大小	数据类型	名称
0	3	4	浮点数	光源位置 X 坐标
4	7	4	浮点数	光源位置 Y 坐标
8	11	4	浮点数	光源位置 Z 坐标

• 4610 点光源块

起始位置	结束位置	大小	数据类型	名称
0	3	4	浮点数	目标位置 X 坐标
4	7	4	浮点数	目标位置 Y 坐标
8	11	4	浮点数	目标位置 Z 坐标
12	15	4	浮点数	热点
16	19	4	浮点数	发散

• 0010 GB 颜色块

起始位置	结束位置	大小	数据类型	名称
0	3	4	浮点数	红
4	7	4	浮点数	绿
8	11	4	浮点数	蓝

• 0011 24 位 RGB 颜色块

起始位置	结束位置	大小	数据类型	名称
0	1	1	浮点数	红
1	1	1	浮点数	绿
2	1	1	浮点数	蓝

• 相机块

起始位置	结束位置	大小	数据类型	名称
0	3	4	浮点数	相机位置 X 坐标
4	7	4	浮点数	相机位置 Y 坐标
8	11	4	浮点数	相机位置 Z 坐标
12	15	4	浮点数	相机目标 X 坐标
16	19	4	浮点数	相机目标 Y 坐标
20	23	4	浮点数	相机目标 Z 坐标
24	27	4	浮点数	相机倾斜(旋转角)
28	31	4	浮点数	相机镜头

下面介绍关键帧块的 ID 及其含义:

ID	含义
B00A	未知
B008	帧信息块
B009	未知
B002	物体信息开始块

• B008 帧信息块

起始位置	结束位置	大小	数据类型	名称
0	3	4	无符号长整型	开始帧
4	7	4	无符号长整型	开始帧

• B002 物体信息开始块

物体信息开始的子块 ID 及其含义如下所示。

子块 ID	含义
B010	名称与层次结构描述
B011	哑元物体命名
B013	未知
B014	未知
B015	未知
B020	物体中心点
B021	未知
B022	未知

• B010 名称与层次结构描述表

起始位置	结束位置	大小	数据类型	名称
0	未确定	2	0 结尾字符串	物体名称
未确定	未确定	2	unsigned int	未知
未确定	未确定	2	unsigned int	未知
未确定	未确定	2	unsigned int	物体的层次结构

物体的层次结构并不复杂,场景中给予每个物体一个数字以标识其在场景树中的顺序。相应地,3DS 文件中也用相同的方法标识了物体在场景树中的位置,作为根物体给予了数 -1(FFFF)作为其数字标识。当读取文件的时候,就会得到一系列的物体数字标识。



对于 3DS 文件,如果当前数字标识比前一个大,那么当前物体是前一个物体的子物体;反之,当数字标识比前一个数字标识小,则又回到上一层结构。

1.1.3 3DS 模型的读取与绘制实例

下面具体介绍一个读入 3DS 文件并将其绘制出来的实例,该程序是基于 Windows 编程,详细的源代码请参考本书所附光盘,部分关键源代码介绍如下。

(1) 定义 3DS 文件的读入类 CLoad3DS

这个类主要用于处理 3DS 文件中的各种块,如对象块、纹理块、材质块等,类声明的部分源代码如下:

```
// 基本块(Primary Chunk),位于文件的开始
#define PRIMARY          0x4D4D
// 主块(Main Chunks)
#define OBJECTINFO       0x3D3D      // 网格对象的版本号
#define VERSION          0x0002      // .3ds 文件的版本
#define EDITKEYFRAME     0xB000      // 所有关键帧信息的头部
// 对象的次级定义(包括对象的材质和对象)
#define MATERIAL         0xAFFF      // 保存纹理信息
#define OBJECT           0x4000      // 保存对象的面、顶点等信息
// 材质的次级定义
#define MATNAME          0xA000      // 保存材质名称
#define MATDIFFUSE       0xA020      // 对象/材质的颜色
#define MATMAP           0xA200      // 新材质的头部
#define MATMAPFILE       0xA300      // 保存纹理的文件名
#define OBJECT_MESH      0x4100      // 新的网格对象
// OBJECT_MESH 的次级定义
#define OBJECT_VERTICES  0x4110      // 对象顶点
#define OBJECT_FACES     0x4120      // 对象的面
#define OBJECT_MATERIAL  0x4130      // 对象的材质
#define OBJECT_UV        0x4140      // 对象的 UV 纹理坐标
```