

孙波 编著

计算机最新技术培训教材

PEKING UNIVERSITY PRESS

OpenGL

编程实例

学习教程



- 理论与实例相结合讲解OpenGL编程
- 学习OpenGL编程的优秀教程

北京大学出版社
<http://cbs.pku.edu.cn>



计算机最新技术培训教材

OpenGL 编程实例学习教程

孙 波 编著

北 京 大 学 出 版 社

北 京

内 容 提 要

OpenGL 是一个工业标准的三维图形软件接口, 它的应用范围覆盖了 CAD/CAM、仿真、科学计算可视化、三维游戏、数字电影、广告动画创作等。

本书采用理论与实例相结合的方式讲解 OpenGL 编程, 全书共有 40 个实例, 讲述的基本内容有: OpenGL 初始化、基本几何对象的生成、颜色模式、坐标变换、显示列表、位图和图像、混合、反走样、雾化效果、光照和材质、纹理、求值器和 NURBS 曲线曲面等。书中实例全部在 Visual C++ 环境下调试通过, 这些实例多数具有很强的交互性, 便于读者对 OpenGL 编程概念的理解。

本书中的例子都是在 Visual C++ 中开发的, 所以需要读者对 Visual C++ 有一定的了解, 包括应用程序向导、对话框、事件响应和 ActiveX 控件的制作和使用等。本书条理清晰、讲解细致, 是一本学习 OpenGL 编程的理想教材。

图书在版编目 (CIP) 数据

OpenGL 编程实例学习教程/孙波编著. —北京: 北京大学出版社, 2000.12
(计算机最新技术培训教材)
ISBN 7-301-02876-8

I. O… II. 孙… III. 三维—动画—图形软件, OpenGL—技术培训—教材 IV. TP391.41

中国版本图书馆 CIP 数据核字 (2000) 第 45993 号

书 名: OpenGL 编程实例学习教程

著作责任者: 孙 波

责任编辑: 黄庆生 汉 明

标准书号: ISBN 7-301-02876-8/TP·258

出 版 者: 北京大学出版社

地 址: 北京市海淀区中关村北京大学校内 100871

网 址: <http://cbs.pku.edu.cn>

电子信箱: xxjs@pup.pku.edu.cn

排 版 者: 南方立德 (Leader) 信息技术中心

印 刷 者: 河北省滦县印刷厂

发 行 者: 北京大学出版社

经 销 者: 新华书店

787 毫米×1092 毫米 16 开本 22.875 印张 557 千字

2000 年 12 月第 1 版 2000 年 12 月第 1 次印刷

定 价: 36.00 元

前 言

随着计算机技术的飞速发展, 计算机动画 (Computer Animation)、科学计算可视化 (Visualization in Scientific Computer) 和虚拟现实 (Virtual Reality) 逐渐成为计算机图形学领域中三大重要技术, 而三维真实感图形又是这三大技术的核心内容。所以, 可以说目前计算机图形学已进入三维真实感图形的时代, 三维图形给我们带来的是与现实更接近的感觉。

OpenGL 是近几年发展起来的一个性能卓越的三维图形标准, 它是 SGI 等多家世界闻名的计算机公司, 以 SGI 的 GL 三维图形库为基础制定出的一个通用共享和开放的三维图形标准, 它性能好且方便灵活。

随着 Microsoft 公司在 Windows NT 和 Windows 95/98 中提供了 OpenGL 标准图形库及众多图形加速卡对 OpenGL 的支持, OpenGL 已经不再是图形工作站的专利了, 而将广泛应用在微机中。

与 Visual C++ 的结合, 更使得 OpenGL 编程得到编程人员的青睐, 本书就是在 Visual C++6.0 的编程环境下开发了大量实例, 采用实例的方式讲述 OpenGL 的编程实现。为了使读者了解 OpenGL 的基本编程模式, 从 C 语言的 Win Console 程序讲起, 然后过渡到使用 MFC 类库的 OpenGL 编程, 因为在 MFC 窗口进行 OpenGL 渲染需要进行一系列的设置, 为了方便使用 OpenGL, 本书建立了一个 OpenGL 控件, 使用这个控件, 用户就可以省去繁琐的设置, 同时, 还可以实现在对话框中的 OpenGL 绘图。

本书讲述的基本内容有: OpenGL 初始化、基本几何对象的生成、颜色模式、坐标变换、显示列表、位图和图像、混合、反走样、雾化效果、光照和材质、纹理、求值器和 NURBS 曲线曲面等。书中的 40 个例子, 包括了以上的基本内容, 并且多数例子具有很强的交互性, 便于读者对 OpenGL 编程概念的理解。

本书中的例子都是在 Visual C++ 中开发的, 所以需要读者对 Visual C++ 有一定的了解, 包括使用应用程序向导、对话框、事件响应、ActiveX 控件的制作和使用等。

本书由孙波主编, 另外, 孙景利、刘天翠、张大志、陈宗秀、康孟霞、张小东、李宁、王强、赵四海、胡琼辉、关福宁、李晓峰、董团结、杨仕润、韩百、涂海滨、顾云飞、张旭、刘贤轶、张志明、朱黎、李民、朱小玉、张刚、刘一勤、张越男、吕顺强、于羽、张华开等也参加了本书编写工作。

由于编者水平有限, 书中错误在所难免, 敬请广大读者批评指正。

编 者

2000, 10, 1

目 录

第 1 章 OpenGL 初步	1
1.1 OpenGL 的基本流程.....	1
1.2 一个简单的 OpenGL 程序.....	2
1.3 本章小结	7
第 2 章 在文档/视图结构中使用 OpenGL	8
2.1 图形操作描述表.....	8
2.2 像素格式	9
2.3 在 MFC 中实现 OpenGL	13
2.4 本章小结	18
第 3 章 OpenGL 控件	19
3.1 建立 OpenGL 控件.....	19
3.2 本章小结	30
第 4 章 OpenGL 控件的使用.....	31
4.1 控件使用实例.....	31
4.2 本章小结	37
第 5 章 点的绘制	38
5.1 绘图前的一些准备工作.....	38
5.2 绘制点	38
5.3 绘制点的实例.....	40
5.4 本章小结	46
第 6 章 线的绘制	47
6.1 画线的基本函数.....	47
6.2 画线实例	47
6.3 本章小结	59
第 7 章 多边形的绘制.....	60
7.1 多边形基础知识.....	60
7.2 多边形实例	60
7.3 本章小结	71

第 8 章 投影变换	72
8.1 OpenGL 基本变换.....	72
8.2 投影变换	72
8.3 投影变换实例.....	74
8.4 本章小结	78
第 9 章 视区变换	79
9.1 视区变换实例.....	79
9.2 本章小结	86
第 10 章 模式变换	87
10.1 取景变换	87
10.2 模型变换	87
10.3 取景和模型变换的矩阵操作.....	87
10.4 模式变换实例.....	88
10.5 本章小结	95
第 11 章 显示列表.....	96
11.1 显示列表的基本概念.....	96
11.2 创建并执行一个显示列表.....	97
11.3 显示列表实例.....	97
11.4 本章小结.....	102
第 12 章 分层的显示列表.....	103
12.1 分层显示的基本内容.....	103
12.2 分层显示实例.....	103
12.3 本章小结	111
第 13 章 显示列表用于绘制字母.....	112
13.1 显示列表的指针.....	112
13.2 用显示列表绘制字母.....	112
13.3 本章小结	120
第 14 章 光源和材质	121
14.1 OpenGL 光照概念.....	121
14.2 光照处理的步骤.....	121
14.3 如何定义光源的特性.....	122
14.4 定义材料属性.....	122
14.5 光照和材质实例.....	123
14.6 本章小结	134

第 15 章 光源和材质的深入.....	135
15.1 光源的位置和强度的衰减.....	135
15.2 光源和材质的深入实例.....	136
15.3 本章小结	150
第 16 章 光源位置的应用.....	151
16.1 光源位置应用实例.....	151
16.2 本章小结	157
第 17 章 混合	158
17.1 混合操作的基本方法.....	158
17.2 混合操作的实例.....	159
17.3 本章小结	169
第 18 章 混合的深入	170
18.1 混合的深入实例.....	170
18.2 本章小结	179
第 19 章 三维物体的混合应用.....	180
19.1 三维物体混合实例.....	180
19.2 本章小结	186
第 20 章 反走样	187
20.1 反走样基本知识.....	187
20.2 反走样实例.....	187
20.3 本章小结	193
第 21 章 反走样深入	194
21.1 三维物体反走样实例.....	194
21.2 本章小结	199
第 22 章 雾效果	200
22.1 雾	200
22.2 雾实例	200
22.3 本章小结	209
第 23 章 位图和字体	210
23.1 位图和字体.....	210
23.2 位图实例	210
23.3 本章小结	217

第 24 章	图像	218
24.1	图像的定义和使用	218
24.2	图像实例	218
24.3	本章小结	223
第 25 章	纹理的定义和使用	224
25.1	初步使用纹理映射	224
25.2	定义一个纹理	224
25.3	纹理实例	225
25.4	本章小结	231
第 26 章	纹理的深入运用	232
26.1	对纹理进行放大、缩小	232
26.2	纹理深入运用实例	232
26.3	本章小结	240
第 27 章	纹理映射模式	241
27.1	纹理映射模式	241
27.2	纹理映射模式实例	241
27.3	本章小结	249
第 28 章	一维求值器	250
28.1	求值器	250
28.2	一维求值器实例	251
28.3	本章小结	258
第 29 章	二维求值器	259
29.1	二维求值器	259
29.2	二维求值器实例	260
29.3	本章小结	267
第 30 章	NURBS 曲面	268
30.1	使用 NURBS 绘制图形	268
30.2	NURBS 曲面实例	269
30.3	本章小结	274
第 31 章	在曲面中使用纹理	275
31.1	曲面纹理实例	275
31.2	本章小结	283

第 32 章	NURBS 曲面的修剪	284
32.1	NURBS 曲面的修剪	284
32.2	曲面修剪实例	284
32.3	本章小结	290
第 33 章	模板缓冲的应用	291
33.1	模板缓冲区	291
33.2	模板检验实例	291
33.3	本章小结	297
第 34 章	场景反走样的运用	298
34.1	累加缓冲区	298
34.2	场景反走样	298
34.3	场景反走样实例	298
34.4	本章小结	306
第 35 章	运动模糊	307
35.1	运动模糊实例	307
35.2	本章小结	313
第 36 章	景深	314
36.1	景深	314
36.2	景深实例	314
36.3	本章小结	322
第 37 章	选择	323
37.1	选择	323
37.2	选择实例	324
37.3	本章小结	331
第 38 章	拾取	332
38.1	实现拾取的基本步骤	332
38.2	拾取实例	332
38.3	本章小结	338
第 39 章	拾取的嵌套	339
39.1	拾取的嵌套	339
39.2	嵌套拾取实例	339
39.3	本章小结	346

第 40 章 反馈	347
40.1 反馈	347
40.2 反馈数组	347
40.3 在反馈模式下使用标记.....	348
40.4 反馈实例	348
40.5 本章小结	356

第1章 OpenGL 初步

OpenGL 原来是一个工业标准的三维计算机图形软件接口,是由 SGI 公司开发出来的。该公司后来为了改善其开放性,开发了 OpenGL。OpenGL 是开放的,适用于多种操作系统,用户可以方便地利用 OpenGL,创建出接近光线跟踪的高质量静止或动画的三维彩色图像,而且要比光线跟踪算法快一个数量级。

OpenGL 是一个三维图形渲染的软件接口,并且得到了硬件加速的支持,也就是说 OpenGL 实际上就是一个三维图形渲染库和三维模型库。由于它在渲染三维图形真实感方面显示出优秀的性能,得到各大公司和各个相关行业的认可,所以成为新一代的三维图形工业标准。从编程的角度讲,OpenGL 是一个图形函数库,最大特点是与硬件无关,可以在不同的硬件平台上得到实现。例如,用 AutoCAD 或 3DS MAX 等软件包可以方便地建立三维几何模型,但难以对其进行移植和控制。如果将这些模型转化为 OpenGL 程序,就可以随心所欲地控制这些模型。可使用 Microsoft Visual C++、Fortran PowerStation 等集成软件开发环境,利用 OpenGL 在微机上实现高品质、交互式的三维图像。

运行 OpenGL 的软件及硬件配置如下所述:

- 操作系统: Windows 95 (OEM Service Release 2) 或 Window NT 4.0 版本。
- 软件开发环境: Microsoft Visual C++4.0 及以上版本。
- 硬件: 奔腾级微机,最好配有支持 OpenGL 硬加速的显示卡。
- 相关文件: windows.h 是在微软 Windows 95 或 Windows NT 环境下使用的窗口头文件,gl.h 是 OpenGL 核心函数的头文件,glu.h 是应用程序库,glaux.h 是辅助库函数的头文件,对于所有的 OpenGL 应用程序,前 3 个头文件是必需的。在创建执行文件时,要另外连接 opengl32.lib、glu32.lib 和 glaux.lib 等 3 个函数库。运行执行文件时,在 Windows95\system 目录下要有 opengl32.dll 和 glu32.dll 两个动态链接库。

1.1 OpenGL 的基本流程

OpenGL 的操作顺序(如图 1-1 所示):

- OpenGL 首先对顶点和像素进行坐标、纹理、法向向量的计算等操作,其中顶点可能来源于基本几何元素、求值器、显示列表、图像和纹理坐标等,像素来源于纹理和图像操作,在显示列表中也可能产生像素的操作。
- 剔除掉取景体以外的几何元素,若使用了裁剪平面,还要实行裁剪。
- 根据点、线、面的属性,OpenGL 生成相应的像素段,确定其颜色、深度值等。

同时，也要对像素进行类似的操作：解压和传输像素，写入纹理内存，使用纹理映射，最后光栅化成像素段。

- 对像素段进行操作，做雾效果计算、反走样处理、模板检验、深度检验、抖动处理等帧缓冲区操作。

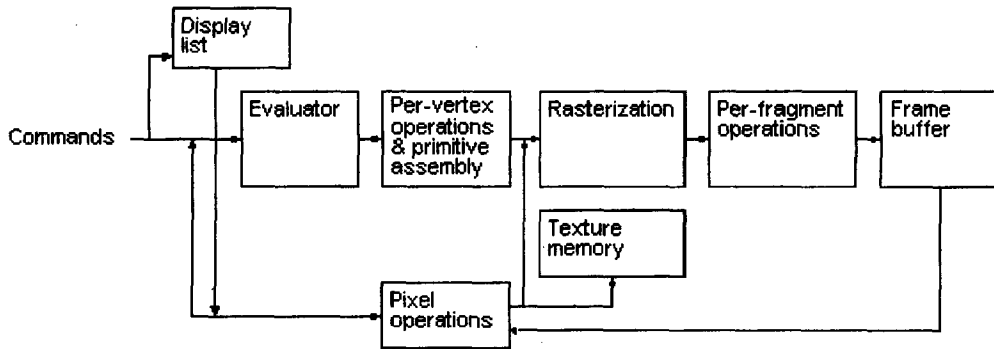


图 1-1 OpenGL 的操作顺序

1.2 一个简单的 OpenGL 程序

1. 建立新项目 Exam1

在 Visual C++ 中，用鼠标点取“File”菜单，选取“New”命令。在弹出的对话框中，选择“Projects”面板，在列表框中选“Win32 Console Application”，在“Project name”文本框中输入 Exam1，单击“OK”，进入 Visual C++ 的应用程序向导。第一步中，确定选中“An empty project”，单击“Finish”。向导创建了一个空的项目（如图 1-2 所示）。

2. 添加头文件

打开 StdAfx.h 文件，在文件末尾添加 gl.h、glu.h、glaux.h 及 MFC 集合类头文件 afxtempl.h，它包含了对 CArray 的定义，如下所示：

```

#include <gl/gl.h>
#include <gl/glu.h>
#include <gl/glaux.h>
#include <afxtempl.h>
  
```

3. 添加静态库

用鼠标点取“Project”菜单，选取“Settings...”命令，选择“Link”面板，在“Object/library Modules:”文本框中输入 opengl32.lib、glu32.lib、glaux.lib，中间用空格隔开。

4. 输入代码

用鼠标点取“File”菜单，选取“New”命令。在弹出对话框中，选择“File”面板，在列表框中选“C++ Source File”，在“File”文本框中输入 Main，单击“OK”，输入以下代码：

```
#include <windows.h>
#include <math.h>
#include <gl/gl.h>
#include <gl/glu.h>
#include <gl/glaux.h>

void init();
void CALLBACK reshape(GLsizei w, GLsizei h);
void CALLBACK display();
GLfloat s, h;
```

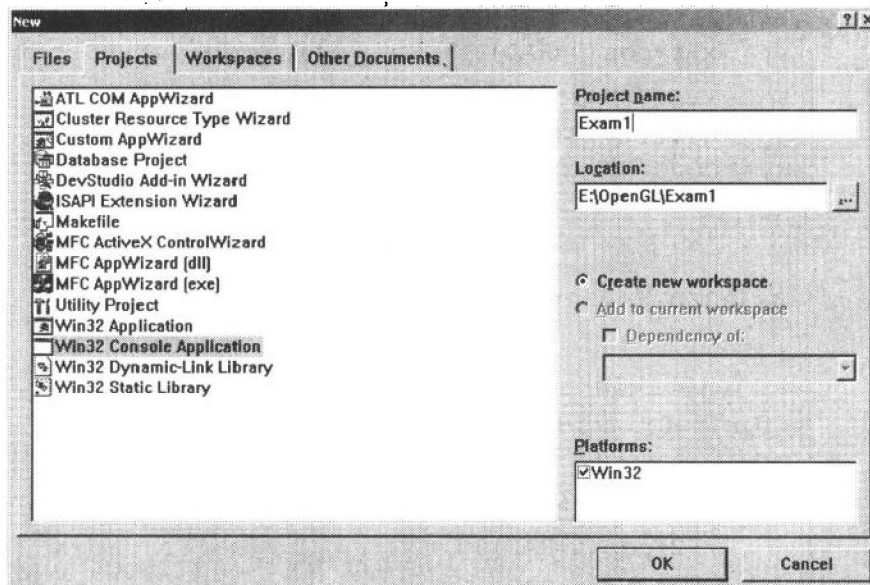


图 1-2 Win32 Console Application

```
//回调函数，绘制窗口时调用
void CALLBACK display()
{
    //清空窗口，设置背景色为白色
    glClearColor(1,1,1,1);
    glClear(GL_COLOR_BUFFER_BIT);
    //取景变换
    glLoadIdentity();
    gluLookAt(5,5,h,s,0,0,0,1,0);
    //glRotatef(30,1,1,0);
    //设置前景色为黑色
    glColor3f(0,0,0);
```

```
//绘图开始, 两条水平平行线
GLfloat RAD=3.1415926/180;
GLfloat x,y,z,r;
int i,j;
for(i=0;i<180;i+=5)
{
    glBegin(GL_LINE_LOOP);
        r=2*sin(i*RAD);
        z=2*cos(i*RAD);
        for(j=0;j<360;j+=10)
        {
            x=r*cos(j*RAD);
            y=r*sin(j*RAD);
            glVertex3f(x,y,z);
        }
    glEnd();
}
for(j=0;j<360;j+=10)
{
    glBegin(GL_LINE_LOOP);
        for(i=0;i<=180;i+=10)
        {
            r=2*sin(i*RAD);
            z=2*cos(i*RAD);
            x=r*cos(j*RAD);
            y=r*sin(j*RAD);
            glVertex3f(x,y,z);
        }
    glEnd();

}
//清空帧缓存
glFlush();
}

//OpenGL 初始化, 设置颜色为单一着色模式
void init()
{
    glShadeModel(GL_FLAT);
```

```
s=0;
h=5;
}

//回调函数，窗口初始化和大小改变时，调用此函数
void CALLBACK reshape(GLsizei w,GLsizei h)
{
    //设置当前矩阵为投影变换矩阵
    glMatrixMode(GL_PROJECTION);
    //设置投影变换
    glLoadIdentity();
    gluPerspective(30,1,-3,3);
    //设置当前矩阵为模式变换矩阵
    glMatrixMode(GL_MODELVIEW);
    //设置视区变换
    glViewport(0,0,w,h);
}

void CALLBACK Left()
{
    s+=0.1;
}
void CALLBACK Right()
{
    s-=0.1;
}
void CALLBACK Up()
{
    h-=0.1;
}
void CALLBACK Down()
{
    h+=0.1;
}
//主函数
void main()
{
    //设置 OpenGL 的显示模式：单缓存、RGB 模式
    auxInitDisplayMode(AUX_SINGLE|AUX_RGB);
```

```
//设置窗口位置、大小和标题
auxInitPosition(0,0,300,300);
auxInitWindow("OpenGL Demo");
init();
//设置回调函数
auxKeyFunc(AUX_LEFT, Left);
auxKeyFunc(AUX_RIGHT, Right);
auxKeyFunc(AUX_UP, Up);
auxKeyFunc(AUX_DOWN, Down);
auxReshapeFunc(reshape);
auxMainLoop(display);
}
```

5. 编译执行

执行结果如图 1-3 所示。

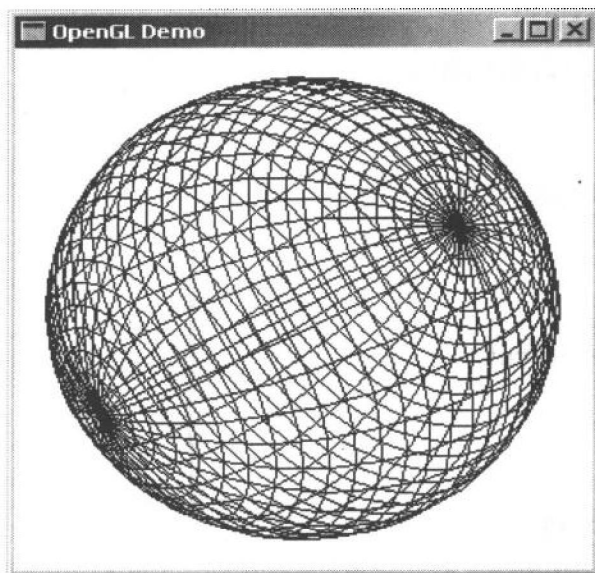


图 1.3 绘制一个线框图

6. 程序分析

主程序中设置 OpenGL 窗口的显示模式，打开窗口的左下角坐标为 (0,0)，右上角坐标为 (300,300)；auxInitWindow 函数给出窗口的标题；init 函数设定 OpenGL 的明暗处理方式；auxShapeFunc 函数在窗口移动、变形后，调用 myReshape 函数重新计算模型、取景变换及投影变换；auxMainLoop 循环调用 displny 函数，设定颜色、投影交换、描述几何要素；auxKeyFunc(AUX_LEFT,Left)设定响应方向键左键的回调函数；auxKeyFunc(AUX_RIGHT,Right)设定响应方向键右键的回调函数；auxKeyFunc(AUX_UP,Up)设定响应方向键向上键的回调函数；auxKeyFunc(AUX_DOWN,Down)设定响应方向键向下键的回调函数。

Display 函数首先设置 OpenGL 窗口的背景及绘图颜色, 然后对物体、场景进行旋转、缩放、平移操作。glBegin 与 glEnd 之间的语句用于给定顶点绘制圆球的经线和纬线, 其中 x 、 y 、 z 为顶点坐标, 由 glVertex 函数定义所描述几何要素的顶点。

1.3 本章小结

本章介绍了一个用纯 C 语言编写 OpenGL 应用程序的基本步骤, 在解释例子程序后, 讲解了 OpenGL 编程所必需的概念、库函数等。下一章将介绍在 MFC 类库的基础上如何编写 OpenGL 应用程序。