**OpenGL的消隐处理**

日期 2020/6/8

**一、实验目的**

1、掌握OpenGL消隐技术；

2、了解基本消隐算法；

3、熟练OpenGL中多边形剔除关键代码。

**二、实验要求**

1、利用消隐技术对绘制的茶壶进行多边形剔除；

2、注释给定代码中的剔除操作。

**三、实验内容（代码及注释）**

#include "stdafx.h"

#include <windows.h>

#include <gl/glut.h>

void Initial()

{

glEnable(GL\_DEPTH\_TEST);

glFrontFace(GL\_CW);

glClearColor(1.0, 1.0, 1.0, 0.0);

}

void ChangeSize(int w, int h)

{

if(h == 0) h = 1;

glViewport(0, 0, w, h);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

if (w <= h)

glOrtho (-4.0f, 4.0f, -4.0f\*h/w, 4.0f\*h/w, -4.0f, 4.0f);

else

glOrtho (-4.0f\*w/h, 4.0f\*w/h, -4.0f, 4.0f, -4.0f, 4.0f);

glMatrixMode(GL\_MODELVIEW);

glLoadIdentity();

}

void Display(void)

{

glClear(GL\_COLOR\_BUFFER\_BIT | GL\_DEPTH\_BUFFER\_BIT);

glColor3f(1.0,0.0,0.0);

glPushMatrix();

/\* 第一个茶壶使用了剔除\*/

glEnable(GL\_CULL\_FACE);

glCullFace (GL\_BACK); //剔除茶壶的后向面

glTranslatef(-2.0f, 0.0f, 0.0f);

glRotatef(180.0f, 0.0f, 1.0f, 0.0f );

GLdouble equ[4] = { -1.0f, 2.3f, 2.3f, 2.3f }; // equ中保存平面方程的系数

glClipPlane(GL\_CLIP\_PLANE0, equ); //glClipPlane定义裁减平面

glEnable(GL\_CLIP\_PLANE0);

glutSolidTeapot(1.0);

glPopMatrix();

/\* 第二个茶壶关闭了剔除操作\*/

glDisable(GL\_CULL\_FACE);

glTranslatef(2.0f, 0.0f, 0.0f);

glRotatef(180.0f, 0.0f, 1.0f, 0.0f );

glClipPlane(GL\_CLIP\_PLANE0, equ);

glEnable(GL\_CLIP\_PLANE0);

glutSolidTeapot(1.0);

glPopMatrix();

glDisable(GL\_CLIP\_PLANE0);

glFlush();

}

int main(int argc, char\* argv[])

{

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB );

glutInitWindowSize(400,400);

glutCreateWindow("茶壶的剔除操作");

glutReshapeFunc(ChangeSize);

glutDisplayFunc(Display);

Initial();

glutMainLoop();

return 0;

}



1. **实验总结**

通过实验掌握了对多边形的消隐处理。

多边形区域排序算法可适用于任意多边形构成场景的交易处理，例如它可以正确消隐循环遮挡的情况。多边形剔除在多边形表面模型中，一个面包括正面和反面，通常正面会被观察，而反面通常看不见，这种看不见的面可以直接进行消隐处理，这种处理可以使用opengl中的多边形剔除函数glEnable(Gl\_ CULL\_FACE)和glCULLFace(mode)。