**实验三 OpenGL绘制实体模型**

1. **实验目的**

1． 掌握GLUT库中的多面体函数的绘制方法；

2． 掌握GLUT库中的二、三次曲面的绘制方法；

3． 掌握绘制实体或线框模型的绘制方法；

4． 掌握显示列表的用法；

**二、实验要求**

1. 利用OpenGL绘制简单多面体示例。

1. 利用OpenGL绘制简单二次曲线示例。

1. 利用OpenGL绘制简单三次曲线示例。

**三、实验内容及代码解释**

#include <gl/glut.h>

static GLsizei iMode = 11;

static GLfloat xRot = 0.0f; //x方向旋转参数

static GLfloat yRot = 0.0f; //y方向旋转参数

GLUquadricObj \*obj; //二次曲面对象

void Initial(void)

{

glClearColor(0.0f, 1.0f, 1.0f, 1.5f);

glColor3f(1.0f, 1.0f, 0.0f);

obj = gluNewQuadric( );

gluQuadricDrawStyle(obj, GLU\_LINE); //以线框方式绘制二次曲面对象

}

void ChangeSize(int w, int h)

{

glViewport(0, 0, w, h);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

gluOrtho2D (1.5f, 1.5f, 1.5f, 1.5f);

}

void Display(void)

{

glClear(GL\_COLOR\_BUFFER\_BIT);

glMatrixMode(GL\_MODELVIEW);

glLoadIdentity();

glRotatef(xRot, 1.0f, 1.0f, 0.0f); //旋转图形

glRotatef(yRot, 1.0f, 1.0f, 0.0f); //旋转图形

//指定需要绘制的图元

switch(iMode) {

case 1:

glutWireTetrahedron(); break;

case 2:

glutSolidTetrahedron();break;

case 3:

glutWireOctahedron(); break;

case 4:

glutSolidOctahedron();break;

case 5:

glutWireSphere(1.0f,15,15); break;

case 6:

glutSolidSphere(1.0f,15,15); break;

case 7:

glutWireTeapot(1.0f); break;

case 8:

glutSolidTeapot(1.0f); break;

case 9:

gluSphere(obj, 1.0f, 15, 15); break;

case 10:

gluCylinder(obj,1.0f,0.0f,1.0f,15,15); break;

case 11:

gluPartialDisk(obj,0.3f,0.8f,15,15,30.0f,360.0f); break;

default: break;

}

glFlush();

}

void ProcessMenu(int value)

{

iMode = value;

glutPostRedisplay();

}

void SpecialKeys(int key, int x, int y)

{

if(key == GLUT\_KEY\_UP) xRot-= 5.0f;

if(key == GLUT\_KEY\_DOWN) xRot += 5.0f;

if(key == GLUT\_KEY\_LEFT) yRot -= 5.0f;

if(key == GLUT\_KEY\_RIGHT) yRot += 5.0f;

if(xRot > 356.0f) xRot = 0.0f;

if(xRot < -1.0f) xRot = 355.0f;

if(yRot > 356.0f) yRot = 0.0f;

if(yRot < -1.0f) yRot = 355.0f;

glutPostRedisplay();

}

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

{

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);

glutInitWindowSize(400,400);

glutInitWindowPosition(300,200);

glutCreateWindow("OpenGL绘制实体模型");

//创建菜单并定义菜单回调函数

int nGlutPolyMenu = glutCreateMenu(ProcessMenu);

glutAddMenuEntry("线框正四面体",1); //创建GLUT多面体绘制菜单

glutAddMenuEntry("实体正四面体",2);

glutAddMenuEntry("线框正八面体",3);

glutAddMenuEntry("实体正八面体",4);

int nGlutCurveMenu = glutCreateMenu(ProcessMenu); //创建GLUT曲面绘制菜单

glutAddMenuEntry("线框球面",5);

glutAddMenuEntry("实体球面",6);

glutAddMenuEntry("实体球面",7);

glutAddMenuEntry("实体茶壶",8);

int nGluCurveMenu = glutCreateMenu(ProcessMenu); //创建GLU曲面绘制菜单

glutAddMenuEntry("线框球面",9);

glutAddMenuEntry("线框圆锥面",10);

glutAddMenuEntry("线框圆环面",11);

int nMainMenu = glutCreateMenu(ProcessMenu); //创建主菜单

glutAddSubMenu("GLUT多面体", nGlutPolyMenu);

glutAddSubMenu("GLUT曲面", nGlutCurveMenu);

glutAddSubMenu("GLU曲面", nGluCurveMenu);

glutAttachMenu(GLUT\_RIGHT\_BUTTON);

glutDisplayFunc(Display);

glutReshapeFunc(ChangeSize);

glutSpecialFunc(SpecialKeys);

Initial();

glutMainLoop();

return 0;

}

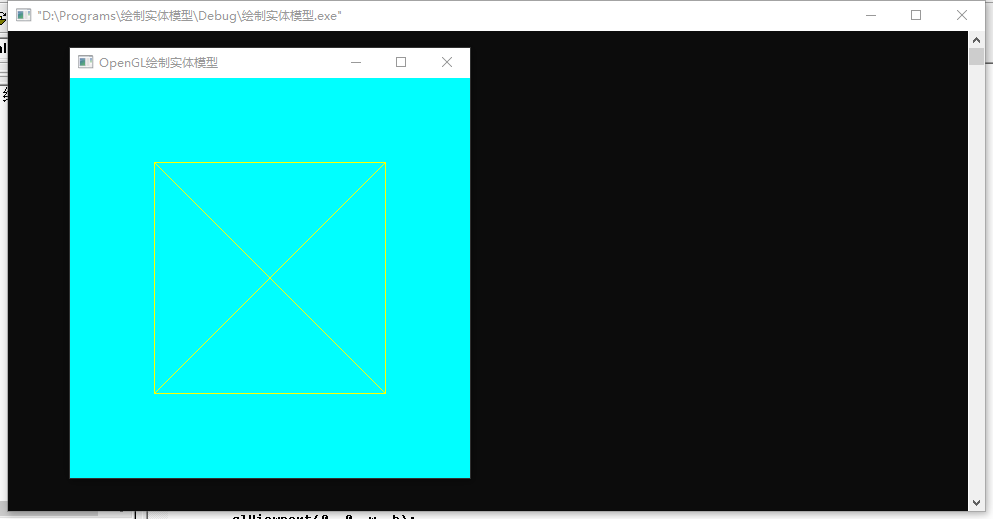
**四、实验总结**

1. 以线框方式绘制二次曲面对象，gluQuadricDrawStyle函数中的参数为GLU\_LINE。

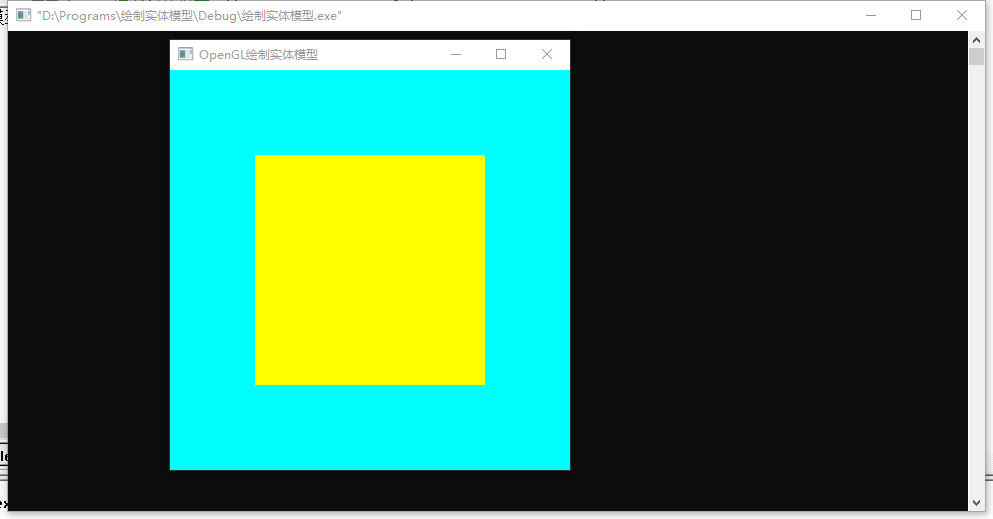
2. 若要使模型显示的更具体，可以加入一些另外的函数，写入参数或者调视角。

**五、实验结果**

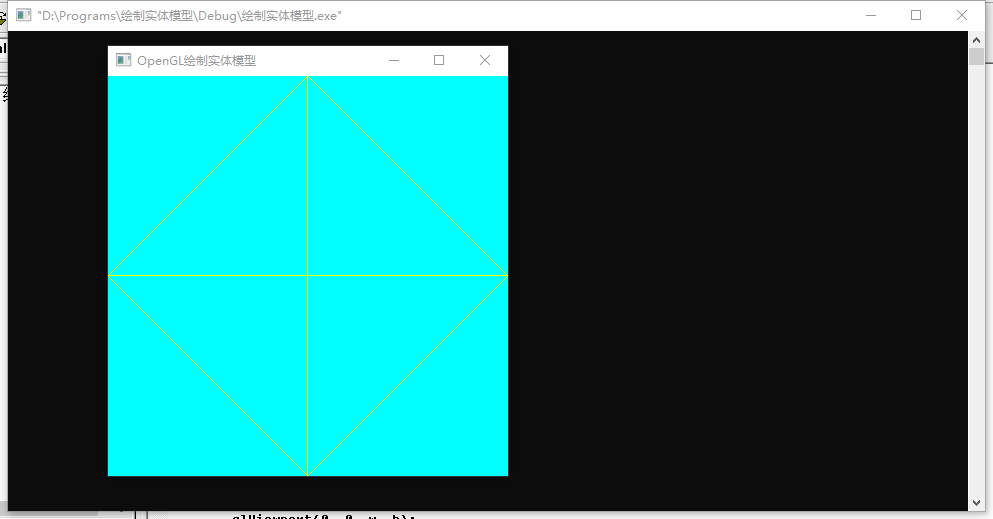
1. 线框正四面体



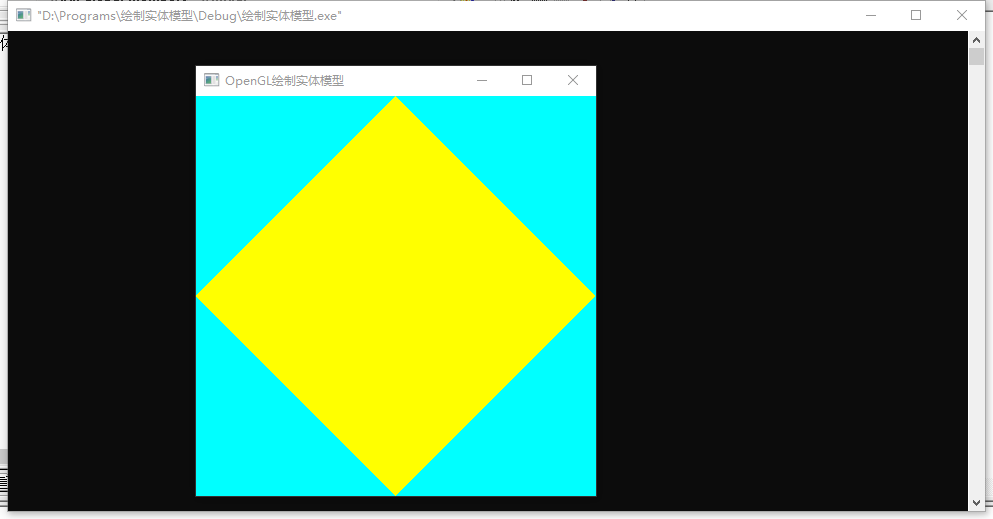
2. 实体正四面体



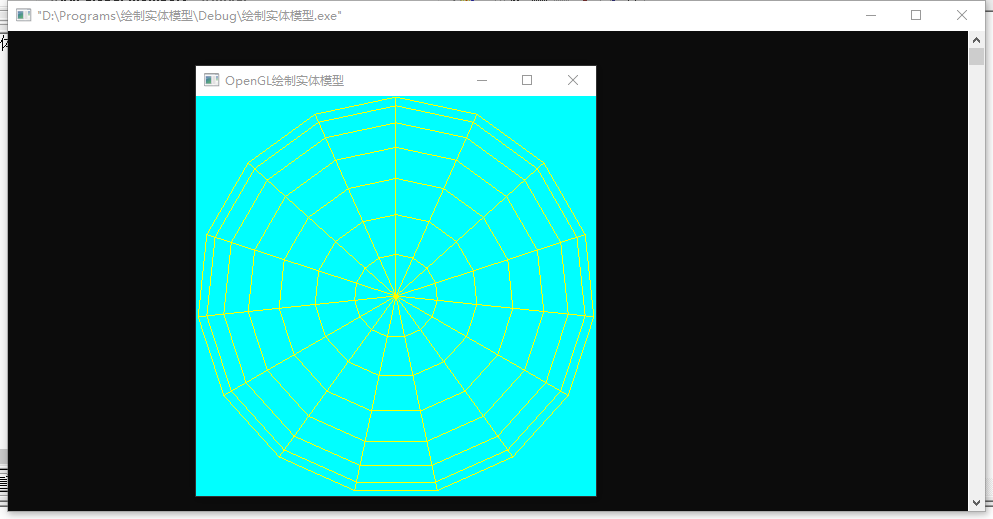
3. 线框正八面体



4. 实体正八面体



5. 线框球面



6. 实体球面



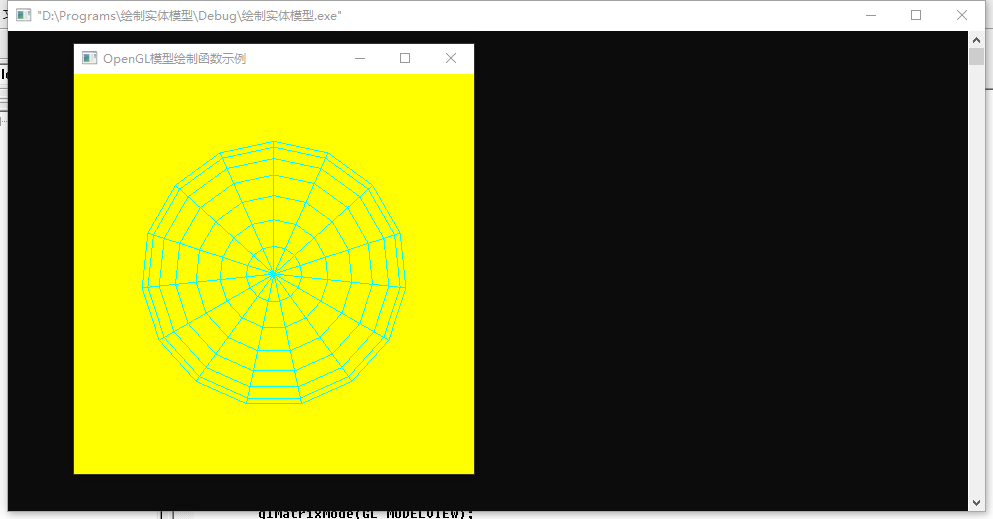
7. 线体茶壶



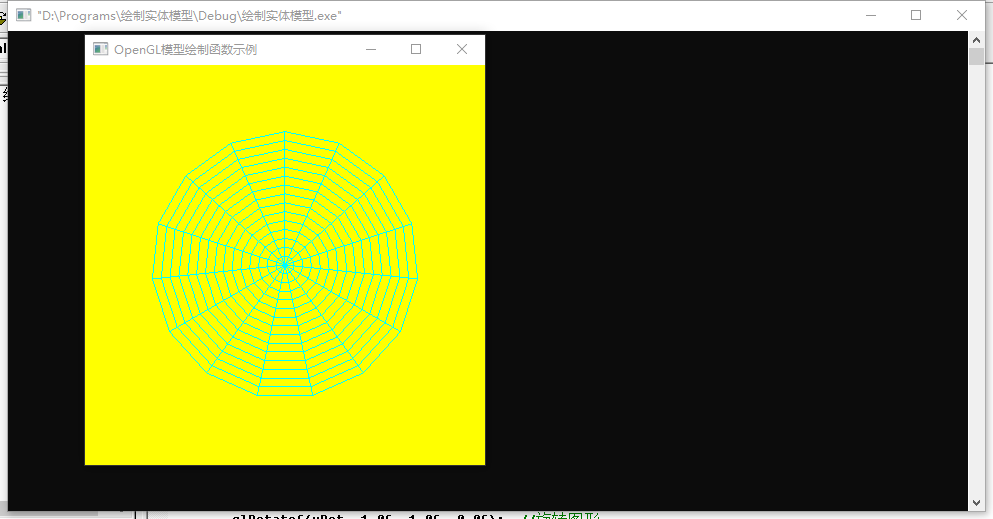
8. 实体茶壶



9. 线框球面



10. 线框圆锥面



11．线框圆环面

