计算机图形学实验报告
OpenGL 实验

一、要求：

设计一个室内三维环境, 并利用OPEN GL展示它的三维效果。

要求：

（1）包含基本的实体元素：球、多面体、锥体、柱体、曲面等；

（2）有全局光照效果和纹理功能；

（3）程序具有交互功能.

二、编程环境：

（1）Visual C++ 2008(包含基本库和实用库)及包含opengl API的几个文件 glut32.dll、glut32.lib、glut.h,

（2）将glut32.dll文件复制到Windows的system32的系统目录下。glut32.lib文件复制到VC 6.0的lib目录下。glut.h复制到VC的include\GL目录下。

（3）在Visusl C++中创建一个空的Win32的控制台应用程序(Win32 Console Application)；在Project/Settings对话框中的Link选项卡中的Object/library modules一栏中加入opengl32.lib, glu32.lib和glut32.lib。

三、设计

（1）创建一个窗口中，在窗口中显示三维图形，图形可以进行旋转，达到从不同角度观察的目的。

```
glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB | GLUT_DEPTH);
glutCreateWindow("opengl jokerlee");
MessageBox(NULL, "点击右键可以进行图形选择 \n 使用键盘的方向键对图形进行旋转", "操作说明", MB_OK);
```

（2）创建菜单（即右键的选择按钮），进行不同三维模型的选择。

为了能在视口中显示不同的三维图形，及对不同的三维图形能够从不同的角度进行观察，我进行了菜单的创建与运用，这样可以在窗口中只显示一个三维图形，并且可以利用鼠标点击右键后进行不同三维图形的显示（默认显示一个实心球）。在主函数中，利用glutCreateMenu，glutAddMenuEntry，glutAddSubMenu三个函数函数创建菜单。

（3）设置窗口视口的初始大小，开启灯光等。

在函数 void OnResize(int w, int h)中对视口大小，光源等进行设置。

（4）建立对三维图形进行旋转的键盘响应函数。

glutSpecialFunc(OnKeyDown) 添加键盘交互的回调函数

OnKeyDown(int key, int x, int y)函数接受键盘的输入（方向键），对于输入的不同的方向键设置不同的 xRot与 yRot值，以对当前显示的三维图形进行不同方向的

旋转操作。

（5）设置显示的回调函数。

glutDisplayFunc(OnRender) 设置显示的回调函数

OnRender()函数进行了清空一个窗口，设置矩阵，在视口显示三维图形，三维图形的旋转等操作。

（6）对一些状态进行初始化。

Start函数主要进行窗口背景色的设置，光照明模型的设置（光源、光照模式、材质等），具体实现方法见源程序。

glLightfv(GL_LIGHT1, GL_AMBIENT, ambientLight)；设置环境泛光
glLightfv(GL_LIGHT0, GL_DIFFUSE, diffuseLight)； 设置漫反射
glLightfv(GL_LIGHT0, GL_SPECULAR, specular)； 设置镜面反射
glEnable(GL_LIGHT0); // 开启光照

glShadeModel(GL_FLAT); // 设置材质类型
glEnable(GL_COLOR_MATERIAL);// 开启材质
glColorMaterial(GL_FRONT, GL_AMBIENT_AND_DIFFUSE); //设置颜色材质

（7）三维图形是借助于opengl的辅助函数画出的。

用到的几个函数如下：

glutSolidSphere(1.0f, 50, 50);	绘制球
glutSolidCube(1.0f);	绘制立方
glutSolidCone(0.30, 1.1f, 30, 25);	绘制圆锥
glutSolidTorus(0.3f, 1.0f, 25, 50);	绘制环形圆面
glutSolidDodecahedron();	绘制十二面体
glutSolidOctahedron();	绘制八面体
glutSolidTetrahedron();	绘制四面体
glutSolidIcosahedron();	绘制二十面体
glutSolidTeapot(1.0f);	绘制茶壶

四、代码

main.c

```
class GLShapes
{

public:
    GLShapes();
    void Start();

private:
```

```
    void SetupRC();
    void SetupMenu();

// 回掉函数
public:
    static void OnMenu( int value );
    static void OnRender( void );
    static void OnKeyDown( int key, int x, int y );
    static void OnResize( int width, int height );

public:
    enum shapt{ SPHERE=1, CUBE, CONE, TORUS, DODECAHEDRON, OCTAHEDRON, TETRAHEDRON,
ICOSAHEDRON, TEAPOT };

};

main.cpp

#include <windows.h>
#include <gl/glut.h>
#include "main.h"

int iShape = 1;
GLfloat xRot = 0.0f;
GLfloat yRot = 0.0f;
GLfloat ambientLight[] = { 0.5f, 0.5f, 0.5f, 1.0f };
GLfloat diffuseLight[] = { 1.0f, 1.0f, 1.0f, 1.0f };
GLfloat specular[] = { 1.0f, 1.0f, 1.0f, 1.0f };
GLfloat specref[] =  { 1.0f, 1.0f, 1.0f, 1.0f };

GLShapes::GLShapes()
{

}

void GLShapes::OnMenu( int value )
{
    iShape = value;
    glutPostRedisplay();
}

void GLShapes::OnRender( void )
{
```

```
// 清空窗口
glClear( GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT );
// 保存矩阵与进行旋转
glPushMatrix();

glRotatef( xRot, 1.0f, 0.0f, 0.0f );
glRotatef( yRot, 0.0f, 1.0f, 0.0f );

// 重新绘制图形
glColor3ub( 128, 128, 128 );
switch( iShape )
{
case SPHERE:
    glutSolidSphere( 1.0f, 50, 50 );
    break;
case CUBE:
    glutSolidCube( 1.0f );
    break;
case CONE:
    glutSolidCone( 0.30, 1.1f, 30, 25 );
    break;
case TORUS:
    glutSolidTorus( 0.3f, 1.0f, 25, 50 );
    break;
case DODECAHEDRON:
    glutSolidDodecahedron();
    break;
case OCTAHEDRON:
    glutSolidOctahedron();
    break;
case TETRAHEDRON:
    glutSolidTetrahedron();
    break;
case ICOSAHEDRON:
    glutSolidIcosahedron();
    break;
case TEAPOT:
     glutSolidTeapot( 1.0f );
    break;
default:
     break;
}
```

```
    glPopMatrix();
    glutSwapBuffers();

}
void GLShapes::SetupRC()
{
    //设置窗口背景色
    glClearColor( 0.0f, 0.0f, 0.0f, 0.0f );
    glEnable( GL_DEPTH_TEST );
    // 开启光照
    glEnable( GL_LIGHTING );
    // 开启并设置光源
    glLightfv( GL_LIGHT0, GL_AMBIENT, ambientLight ); // 环境泛光
    glLightfv( GL_LIGHT0, GL_DIFFUSE, diffuseLight ); // 漫反射
    glLightfv( GL_LIGHT0, GL_SPECULAR, specular );    // 镜面反射
    glEnable( GL_LIGHT0 );
    // 开启并设置材质
    glShadeModel( GL_FLAT ); // GL_SMOOTH
    glEnable( GL_COLOR_MATERIAL );
    glColorMaterial( GL_FRONT, GL_AMBIENT_AND_DIFFUSE );
    glMaterialfv( GL_FRONT, GL_SPECULAR, specref );
    glMateriali( GL_FRONT, GL_SHININESS, 128 );
}


void GLShapes::OnKeyDown( int key, int x, int y )
{
    // 处理旋转
    if ( key == GLUT_KEY_UP )
        xRot -= 5.0f;
    else if ( key == GLUT_KEY_DOWN )
        xRot += 5.0f;
    else if ( key == GLUT_KEY_LEFT )
        yRot -= 5.0f;
    else if ( key == GLUT_KEY_RIGHT )
        yRot += 5.0f;

    if ( key > 356.0f )           // 超过360度则置零
    {
        xRot = 0.0f;
        yRot = 0.0f;
    }
    else if ( key < -1.0f )       // 小于-1度则置355
```

```
    {
        xRot = 355.0f;
        yRot = 355.0f;
    }
    /// 重绘
    glutPostRedisplay();
}


void GLShapes::OnResize( int w, int h )
{
    GLfloat  lightPos[] = { -50.f, 50.0f, 100.0f, 1.0f };
    GLfloat nRange = 1.9f;
    if (h == 0) h = 1;
    glViewport( 0, 0, w, h );
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    // 设置修剪空间的范围，缩放
    if (w <= h)
        glOrtho (-nRange, nRange, -nRange*h/w, nRange*h/w, -nRange, nRange);
    else
        glOrtho (-nRange*w/h, nRange*w/h, -nRange, nRange, -nRange, nRange);
    glMatrixMode( GL_MODELVIEW );
    glLoadIdentity();
    glLightfv( GL_LIGHT0, GL_POSITION, lightPos ); // 重新设置光源
}

void GLShapes::SetupMenu()
{
    // 创建右键菜单
    int nMainMenu;
    nMainMenu = glutCreateMenu( OnMenu );

    // 添加子菜单
    glutAddMenuEntry( "Sphere", 1 );
    glutAddMenuEntry( "Cube", 2 );
    glutAddMenuEntry( "Cone", 3 );
    glutAddMenuEntry( "Torus", 4 );
    glutAddMenuEntry( "Dodecahedron", 5 );
    glutAddMenuEntry( "Octahedron", 6 );
    glutAddMenuEntry( "Tetrahedron", 7 );
    glutAddMenuEntry( "Icosahedron", 8 );
    glutAddMenuEntry( "Teapot", 9 );
```

```
    glutAttachMenu( GLUT_RIGHT_BUTTON );
}

void GLShapes::Start()
{
    glutInitDisplayMode( GLUT_DOUBLE | GLUT_RGB | GLUT_DEPTH );
    glutCreateWindow( "计算机图形学 opengl by jokerleee@gmail.com" );
    MessageBox( NULL, L"点击右键进行图形选择\n键盘的方向键对图形进行旋转", L"交互操作说明",
 MB_OK );
    SetupMenu();                  // 设置右键菜单
    glutReshapeFunc( OnResize );  // 处理窗口大小变化
    glutSpecialFunc( OnKeyDown ); // 添加键盘交互
    glutDisplayFunc( OnRender );  // 设绘制回调函数
    SetupRC();                        // 自定义的初始化例程
    glutMainLoop();                   // 让GLUT框架开始运行，处理交互事件
}

int main()
{
    /// 隐藏控制台
    FreeConsole();
    GLShapes app;
    app.Start();
    return 0;
}
```

五、实验总结

通过本次实验，初步熟悉了用OpenGL进行3D绘图的方法，了解了OpenGL API的结构、参数和使用方法，并锻炼了编码能力。

姓名：李劼
班级：07409
学号：071202
邮箱：jokerleee@gmail.com