

**计算机图形学作业报告**

|  |  |
| --- | --- |
| 作业名称： | 透视变换下绘制三维动态后三位学号 |
| 姓 名： | 郭炅 |
| 学 号： | 3170105370 |
| 电子邮箱： | 3170105370@zju.edu.cn |
| 联系电话： | 18888923171 |
| 指导老师： | 张宏鑫 |

2019年 10月 16 日

实验名字

1. 作业已实现的功能简述及运行简要说明
   1. 功能简述
      1. 使用透视变化，绘制三维动态后三位学号（370）
      2. **（Bonus）立方体使用自定义的函数，而未使用 glutWireCube(dSize);**
      3. 左侧数字（3）沿y轴上下运动
      4. 中间数字（7）沿y轴正方向旋转
      5. 右侧数字（0）沿z轴缩放运动
   2. 运行简要说明
      1. 使用VS 2017在安装对应地windows SDK的前提下直接打开运行即可
2. 作业的开发与运行环境
   1. Visual Studio 2017 Enterprise
3. 系统或算法的基本思路、原理、及流程或步骤等
   1. 考察 glutWireCube 函数的实现，其绘制以原点为中心，面垂直于坐标轴的立方体，以相同的思路使用画线函数绘制立方体，即先画出上下底面，再将对应顶点连接即可
   2. 移动坐标原点以在不同位置图形
   3. 坐标系旋转以实现图形的旋转
   4. 缩放以绘制长方体
   5. 将缩放函数、旋转函数和坐标移动函数中的参数与时间参数关联，时间参数即通过不断累加变量并且取整实现，实现动态动画
4. 具体如何实现，包括关键（伪）代码、主要用到函数与算法等
   1. 使用 glBegin()和glEnd()函数开始绘制图形，使用 GL\_LINE\_LOOP和GL\_LINES两种模式用线条构建出立方体，使用glVertex3f()指定点，并进行连线，使用dSize作为立方体边长**（Bonus）**

*void* MyWireCube(*double* *dSize*)

{

*double* halfSize = dSize / 2;

    glBegin(GL\_LINE\_LOOP);

    glColor3f(1.0f, 1.0f, 1.0f);

    glVertex3f(halfSize, halfSize, halfSize);

    glVertex3f(halfSize, halfSize, -1.0\*halfSize);

    glVertex3f(-1.0\*halfSize, halfSize, -1.0\*halfSize);

    glVertex3f(-1.0\*halfSize, halfSize, halfSize);

    glEnd();

    glBegin(GL\_LINE\_LOOP);

    glColor3f(1.0f, 1.0f, 1.0f);

    glVertex3f(halfSize, -1.0\*halfSize, halfSize);

    glVertex3f(halfSize, -1.0\*halfSize, -1.0\*halfSize);

    glVertex3f(-1.0\*halfSize, -1.0\*halfSize, -1.0\*halfSize);

    glVertex3f(-1.0\*halfSize, -1.0\*halfSize, halfSize);

    glEnd();

    glBegin(GL\_LINES);

    glColor3f(1.0f, 1.0f, 1.0f);

    glVertex3f(halfSize, -1.0\*halfSize, halfSize);

    glVertex3f(halfSize, halfSize, halfSize);

    glVertex3f(halfSize, -1.0\*halfSize, -1.0\*halfSize);

    glVertex3f(halfSize, halfSize, -1.0\*halfSize);

    glVertex3f(-1.0\*halfSize, -1.0\*halfSize, halfSize);

    glVertex3f(-1.0\*halfSize, halfSize, halfSize);

    glVertex3f(-1.0\*halfSize, -1.0\*halfSize, -1.0\*halfSize);

    glVertex3f(-1.0\*halfSize, halfSize, -1.0\*halfSize);

    glEnd();

}

* 1. 使用glutPerspective()转为透视视图

    gluPerspective(45.0f, (GLfloat)width / (GLfloat)height, 0.1f, 100.0f);

* 1. 使用glTranslatef()移动坐标系原点

glTranslatef(-2.0f, yMovement, -6.0f);

* 1. 使用glRotatef()函数对坐标系沿某一轴进行旋转

glRotatef(timePara\*360, 0.0f, 1.0f, 0.0f);

* 1. 使用glScalef()对x、y或z轴进行放缩，绘制立方体

    glScalef(1, 0.25, 0.25);

    Draw();

    glScalef(1, 4, 4);

* 1. 使用sin()函数和全局变量作为参量控制动态缩放、移动、旋转，每次调用绘图函数均会令全局变量的值发生改变

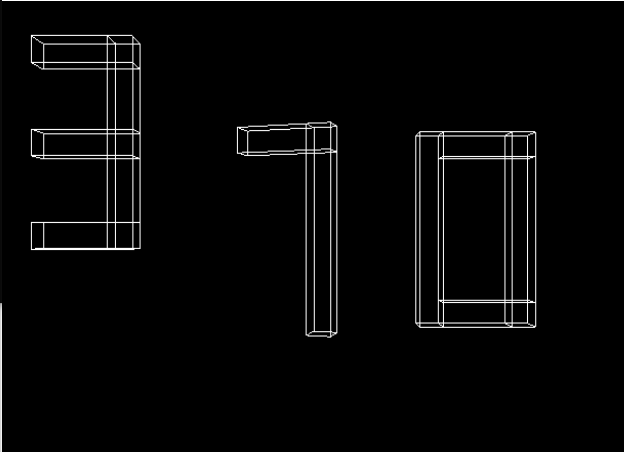
*double* timePara = 0;

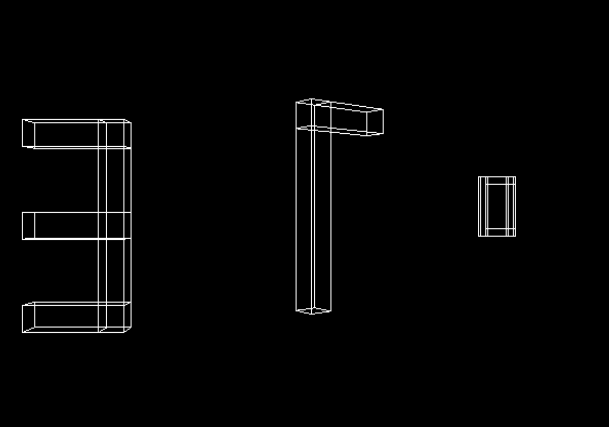
*double* yMovement = sin(timePara);

timePara += 0.0001;

glTranslatef(-2.0f, yMovement, -6.0f);

1. 实验结果与分析
   1. 实验结果





* 1. 分析

可以看到左侧数字3发生沿y轴上下移动，中间数字7沿y轴发生旋转，右侧数字0沿Z轴发生缩放

1. 结论与心得体会

初步学会了如何绘制立体图形，并且使用OpenGL库函数进行坐标系的缩放和移动，学到了如何使用透视视图

1. 参考文献

无