|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **计算机图形学实验报告（四）** | | | | | | |
| 学 号 | 姓 名 | | 班 级 | | 报告日期 | |
| 180400715 | 杨卓宸 | | 1804102 | | 2020/11/6 | |
| 实验内容 | (1) 绘制三个立方体，位置如图所示。要求绘制出坐标系的3 个坐标轴。  (2) 三个立方体一起沿着坐标轴(x 或 y 或 z)旋转起来。  说明：旋转可以通过设置时钟事件来实现，提供的程序里已经实现旋转的功能。打开菜单： 编辑-rotate about x 等即可旋转，可以参考其实现。   1. 三个立方体分别沿 X 轴，Y 轴和 Z 轴旋转。 | | | | | |
| 实验目的 | 1.熟练掌握OpenGL编程的基本方法。  2.学会图形绘制，使用基本图元：点、线、多边形来绘制三维图 形和场景。  3. 变换操作，从不同的角度和距离观察物体。 | | | | | |
| 实验预备知识 | 1.OpenGL 提供了3个函数库: 基本库 实用库 辅助库  基本库是OpenGL的核心函数库，提供了115个 函数，函数以“gl”为前缀。   主要功能包括物体描述、平移、旋转、缩放、 光照、纹理、材质、像素、位图、文字处理等。  2. OpenGL中的变换  OpenGL中的变换矩阵：  glRotatef(2,2,3); //变换矩阵  M draw\_triangle(); //定义点 P  OpenGL中的多种变换（几何变换、投影变换等） 是由矩阵的乘积实现的。   矩阵操作模式：  void glMatrixMode(Glenum mode);  参数取值 GL\_PROJECTTION  表示该语句后面的变换是针对投影变换的。  3.OpenGL中的矩阵堆栈  矩阵堆栈的使用：  void glPushMatrix(): 复制栈顶矩阵并压入栈。  void glPopMatrix(): 移除栈顶矩阵。 | | | | | |
| 实验过程描述 | 1.仔细研读实验材料，结合ppt，理解明白每个函数的作用。  2.补充RenderScene()函数：  画出坐标系：  glLineWidth(4.0f);  glBegin(GL\_LINES);  glColor3f(0, 0, 1);  glVertex3f(0, 0, 0);  glVertex3f(2, 0, 0);  glColor3f(1, 0, 0);  glVertex3f(0, 0, 0);  glVertex3f(0, 2, 0);  glColor3f(0, 1, 0);  glVertex3f(0, 0, 0);  glVertex3f(0, 0, 2);  glEnd();  通过判断不同的按键信息来画方块;  首先是三个立方体看做一个整体的情况：  if (m\_rotate\_axis == XX)//如果想要按x轴旋转  {  glRotatef(m\_angle, 1, 0, 0);  }  else if (m\_rotate\_axis == YY) //如果想要按y轴旋转  {  glRotatef(m\_angle, 0, 1, 0);  }  else if (m\_rotate\_axis == ZZ) //如果想要按z轴旋转  {  glRotatef(m\_angle, 0, 0, 1);  }  else if (m\_rotate\_axis == NOROTATE) //默认状态，直接按照一定角度画出三个立方体即可  {  glRotatef(315, 0, 1, 0);  glRotatef(40, 0, 1, 1);  glRotatef(310, 0, 1, -1);  glRotatef(350, 1, 1, 0);  glRotatef(355, 0, 0, 1);  }  假设按下的是想要三个立方体分别旋转，则需要一步一步地画三个立方体，达到不让三个立方体朝一个方向转的的目的：  glPushMatrix();  glTranslatef(0, 1, 0);  glColor3f(1, 0, 0);  if (m\_rotate\_axis == CYLINDER)  {  glRotatef(m\_angle, 0, 1, 0);  }  draw\_cube();  glPopMatrix();  glPushMatrix();  glTranslatef(1, 0, 0);  glColor3f(0, 0, 1);  if (m\_rotate\_axis == CYLINDER)  {  glRotatef(m\_angle, 1, 0, 0);  }  draw\_cube();  glPopMatrix();  glPushMatrix();  glTranslatef(0, 0, 1);  glColor3f(0, 1, 0);  if (m\_rotate\_axis == CYLINDER)  {  glRotatef(m\_angle, 0, 0, 1);  }  draw\_cube();  glPopMatrix();  3.不断调试，通过调整角度使效果最佳。 | | | | | |
| 实验结果 | 1.直接运行程序画出的立方体：    2.绕x轴旋转：    3.绕y轴旋转    4.绕z轴旋转     1. 绕xyz三轴旋转   捕获1捕获   1. 停止：   捕获2 | | | | | |
| 实验当中问题  及解决方法 | 1.问题：  起初没有仔细阅读材料，在原有的基础上再画两个立方体即可，但是期初没有想到直接调用画立方体的函数，而是通过寻找立方体每个点的坐标一个面一个面地画。  解决：直接调用draw\_cube()函数。  2.问题：  没有很好的理解题目要求3。  解决：  每个方块都要旋转但不是一起朝一个方向旋转，而是沿各自的轴转。通过判断按下的事件来控制旋转。 | | | | | |
| 成绩（教师打分） | 优秀 | 良好 | | 及格 | | 不及格 |