**西安电子科技大学**

**计算机学院**

**图**

**形**

**学**

**上**

**机**

**报**

**告**

**班 级：** 1403013

**姓 名：** 刘欢

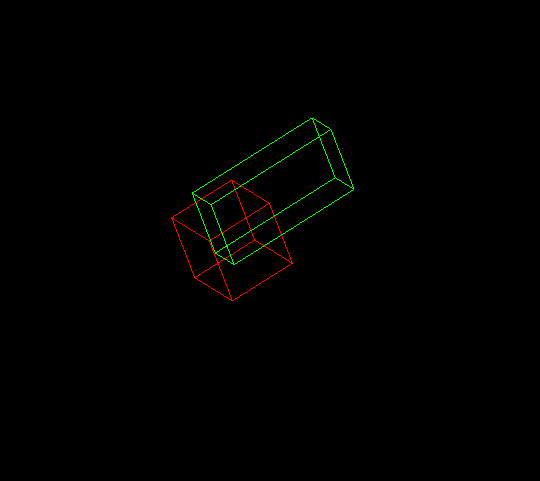
**学 号：** 14030130097

1. **实验目的**
2. 掌握图形的二维基本变换及其OpenGL实现。
3. 掌握图形窗口到视区的变换
4. 掌握图形的三维几何变换及其OpenGL实现

**二．实验内容及结果**

**题目一：**利用OpenGL实现一个立方体关于参考点（10.0,20.0,10.0）进行放缩变换，放缩因子为（2.0,1.0,0.5）

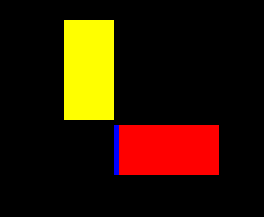
实验结果：



红色立方体为初始立方体绕着向量（1,1,1）旋转45°的图像，绿色立方体为变换后的结果。

**题目二：**利用OpenGL实现一个矩形关于y=x+5对称的新图形。

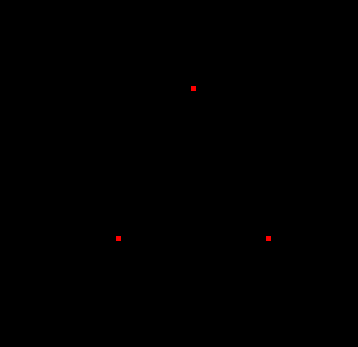
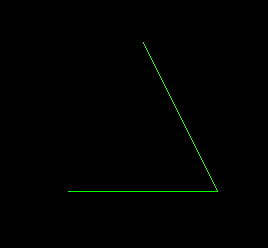
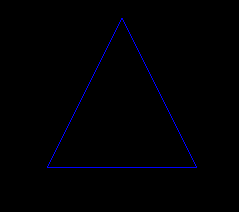
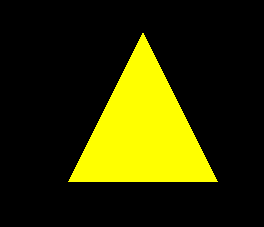
实验结果：

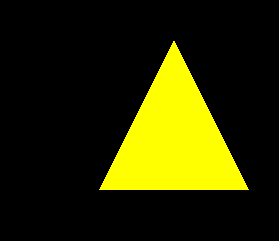
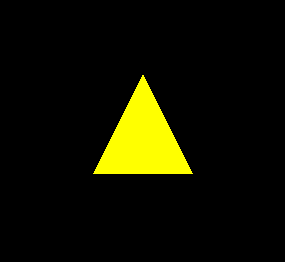
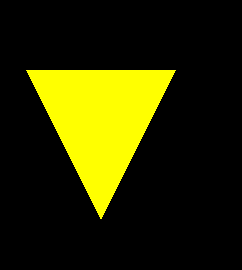
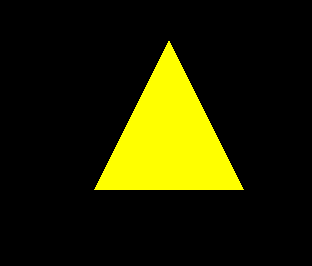


矩形两顶点为（50.0,0.0），（150,50），为蓝色，关于y=x+5对称，相当于将矩形先沿x轴正方向平移5（红色矩形），再绕空间向量（1.0,1.0,0.0）旋转180°（黄色矩形）。

**题目三：**通过定义键盘回调函数，每按一次空格键，让三个点依次完成画点、画线、画三角形、让三角形平移和缩放，并让三角形沿三角形中心旋转起来。

实验结果：

**二．心得体会**

通过本次上机实验，掌握了图形的二维变换和三维变换的算法实现，并且加深了课堂上学过的关于图形二维和三维变换的原理的理解，巩固了理论知识。

**三．源代码**

**题目一：**

**#define GLUT\_DISABLE\_ATEXIT\_HACK**

**#include <GL\glut.h>**

**#include<math.h>**

**void init()**

**{**

**glClearColor(0.0, 0.0, 0.0, 0.0);**

**}**

**void RenderScene()**

**{**

**glClear(GL\_COLOR\_BUFFER\_BIT);**

**glColor3f(1.0, 0.0, 0.0);**

**glMatrixMode(GL\_MODELVIEW);**

**glLoadIdentity();**

**glRotatef(45.0, 1.0, 1.0, 1.0);//绕着向量（1,1,1）所指定的轴旋转45°**

**glPushMatrix();//保存矩阵状态**

**glutWireCube(10.0);**

**glColor3f(0.0, 1.0, 0.0);**

**glTranslatef(-10.0, -20.0, -10.0); //将参考点移到原点**

**glScalef(2.0, 1.0, 0.5); //在x、y和z轴放大2倍**

**glTranslatef(10.0, 20.0, 10.0); //再平移回参考点**

**glutWireCube(10.0);**

**glFlush();**

**}**

**void ChangeSize(GLsizei w, GLsizei h)**

**{**

**GLfloat aspectRatio;**

**if (h == 0)**

**h = 1;**

**glViewport(0, 0, w, h);**

**glMatrixMode(GL\_PROJECTION);**

**glLoadIdentity();**

**aspectRatio = (GLfloat)w / (GLfloat)h;**

**if (w <= h)**

**glOrtho(-50.0, 50.0, -50.0 / aspectRatio, 50.0 / aspectRatio, -50.0, 50.0);**

**else**

**glOrtho(-50.0\*aspectRatio, 50.0\*aspectRatio, -50.0, 50.0, -50.0, 50.0);**

**glMatrixMode(GL\_MODELVIEW);**

**glLoadIdentity();**

**}**

**void main()**

**{**

**glutInitDisplayMode(GLUT\_RGB | GLUT\_SINGLE);**

**glutCreateWindow("1");**

**init();**

**glutDisplayFunc(RenderScene);**

**glutReshapeFunc(ChangeSize);**

**glutMainLoop();**

**}**

**题目二：**

**#define GLUT\_DISABLE\_ATEXIT\_HACK**

**#include <GL/glut.h>**

**void init()**

**{**

**glClearColor(0.0, 0.0, 0.0, 0.0);**

**}**

**void RenderScene()**

**{**

**glClear(GL\_COLOR\_BUFFER\_BIT);**

**glColor3f(0.0, 0.0, 1.0);**

**glRectf(50.0, 0.0, 150.0, 50.0);**

**glColor3f(1.0, 0.0, 0.0);**

**glTranslatef(5, 0.0, 0.0);**

**glRectf(50.0, 0.0, 150.0, 50.0);**

**glLoadIdentity();**

**glColor3f(1.0, 1.0, 0.0);**

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

**glRectf(55.0, 0.0, 155.0, 50.0);**

**glFlush();**

**}**

**void ChangeSize(GLsizei w, GLsizei h)**

**{**

**float ratio;**

**if (h == 0)**

**h = 1;**

**ratio = (GLfloat)w / (GLfloat)h;**

**glViewport(0, 0, w, h);**

**glMatrixMode(GL\_PROJECTION);**

**glLoadIdentity();**

**if (w <= h)**

**gluOrtho2D(-200.0, 200.0, -200.0 / ratio, 200.0 / ratio);**

**else**

**gluOrtho2D(-200.0\*ratio, 200.0\*ratio, -200.0, 200);**

**glMatrixMode(GL\_MODELVIEW);**

**glLoadIdentity();**

**}**

**void main()**

**{**

**glutInitDisplayMode(GLUT\_RGB | GLUT\_SINGLE);**

**glutInitWindowPosition(50, 50);**

**glutInitWindowSize(600, 400);**

**glutCreateWindow("Geometric transformation 2");**

**init();**

**glutDisplayFunc(RenderScene);**

**glutReshapeFunc(ChangeSize);**

**glutMainLoop();**

}  
**题目三：**

**#define GLUT\_DISABLE\_ATEXIT\_HACK**

**#include <stdlib.h>**

**#include <stdio.h>**

**#include <GL/glut.h>**

**int currentMode = 0;**

**const int ModeNums = 8;**

**void init()**

**{**

**glClearColor(0.0, 0.0, 0.0, 0.0);**

**}**

**void myKey(unsigned char key, int x, int y) //响应ASCII对应键，鼠标的当前x和y位置也被返回。**

**{**

**switch (key)**

**{**

**case ' ': currentMode = (currentMode + 1) % ModeNums;**

**glutPostRedisplay();**

**break;**

**case 27: exit(-1);**

**}**

**}**

**void RenderScene()**

**{**

**glClear(GL\_COLOR\_BUFFER\_BIT);**

**switch (currentMode)**

**{**

**case 0: glPointSize(5);**

**glBegin(GL\_POINTS);**

**glColor3f(1.0, 0.0, 0.0);**

**glVertex2f(-3.0, -3.0);**

**glVertex2f(3.0, -3.0);**

**glVertex2f(0.0, 3.0);**

**glEnd();**

**glFlush();**

**break;**

**case 1: glBegin(GL\_LINE\_STRIP);**

**glColor3f(0.0, 1.0, 0.0);**

**glVertex2f(-3.0, -3.0);**

**glVertex2f(3.0, -3.0);**

**glVertex2f(0.0, 3.0);**

**glEnd();**

**glFlush();**

**break;**

**case 2: glBegin(GL\_LINE\_LOOP);**

**glColor3f(0.0, 0.0, 1.0);**

**glVertex2f(-3.0, -3.0);**

**glVertex2f(3.0, -3.0);**

**glVertex2f(0.0, 3.0);**

**glEnd();**

**glFlush();**

**break;**

**case 3: glBegin(GL\_TRIANGLES);**

**glColor3f(1.0, 1.0, 0.0);**

**glVertex2f(-3.0, -3.0);**

**glVertex2f(3.0, -3.0);**

**glVertex2f(0.0, 3.0);**

**glEnd();**

**glFlush();**

**break;**

**case 4: glBegin(GL\_TRIANGLES);**

**glColor3f(1.0,1.0,0.0);**

**glVertex2f(-0.0, -3.0);**

**glVertex2f(6.0, -3.0);**

**glVertex2f(3.0, 3.0);**

**glEnd();**

**glFlush();**

**break;**

**case 5: glColor3f(1.0, 1.0, 0.0);**

**glBegin(GL\_TRIANGLES);**

**glVertex2f(-2.0, -2.0);**

**glVertex2f(2.0, -2.0);**

**glVertex2f(0.0, 2.0);**

**glEnd();**

**glFlush();**

**break;**

**case 6: glRotatef(180.0, 0.0, 0.0, 0.0);**

**glColor3f(1.0, 1.0, 0.0);**

**glBegin(GL\_TRIANGLES);**

**glVertex2f(-0.0, -3.0);**

**glVertex2f(6.0, -3.0);**

**glVertex2f(3.0, 3.0);**

**glEnd();**

**glFlush();**

**break;**

**case 7: glRotatef(180.0, 0.0,0.0,0.0);**

**glColor3f(1.0, 1.0, 0.0);**

**glBegin(GL\_TRIANGLES);**

**glVertex2f(-0.0, -3.0);**

**glVertex2f(6.0, -3.0);**

**glVertex2f(3.0, 3.0);**

**glEnd();**

**glFlush();**

**break;**

**}**

**}**

**void ChangeSize(GLsizei w, GLsizei h)**

**{**

**float ratio;**

**if (h == 0)**

**h = 1;**

**glViewport(0, 0, w, h);**

**glMatrixMode(GL\_PROJECTION);**

**glLoadIdentity();**

**ratio = (float)w / (float)h;**

**if (w <= h)**

**gluOrtho2D(-10.0, 10.0, -10.0 / ratio, 10.0 / ratio);**

**else**

**gluOrtho2D(-10.0\*ratio, 10.0\*ratio, -10.0, 10.0);**

**glMatrixMode(GL\_MODELVIEW);**

**glLoadIdentity();**

**}**

**void main()**

**{**

**glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);**

**glutInitWindowPosition(0, 0);**

**glutInitWindowSize(500, 500);**

**glutCreateWindow("KeyboardFunc");**

**init();**

**glutDisplayFunc(RenderScene);**

**glutReshapeFunc(ChangeSize);**

**glutKeyboardFunc(myKey); //为当前窗口设置键盘回调函数。**

**printf("Press space to continue,press escape to exit!\n");**

**glutMainLoop();**

**}**