实验六 二维图形几何变换实验

时间：2022年4月27日

地点：信息学院机房

1、实验内容

实验一：使用opengl，对二维几何图形进行几何变换

实验二：鼠标函数实验

2、实验目的

验证二维几何变换实验，平移、比例、旋转、错切、对称变换，其中三种

3、实验代码

实验一：

//初始化

void init(void) {

glClearColor(1.0, 1.0, 1.0, 1.0);

gluOrtho2D(-400, 400.0, -400, 400.0);

}

//平移变换

void translation(int \*x,int \*y,int xdistance,int ydistance){

int now[1][3]={\*x,\*y,1};

int by[3][3]={1,0,0,

0,1,0,

xdistance,ydistance,1};

int after[1][3]={0};

int i,j,k;

for(i=0;i<3;i++){

for(j=0;j<3;j++){

after[0][i] += now[0][j]\*by[j][i];

}

}

\*x=after[0][0];

\*y=after[0][1];

}

//比例变换

void proportion(int \*x,int \*y,float xmultiple,float ymultiple){

int now[1][3]={\*x,\*y,1};

float by[3][3]={xmultiple,0,0,

0,ymultiple,0,

0,0,1};

float after[1][3]={0};

int i,j,k;

for(i=0;i<3;i++){

for(j=0;j<3;j++){

after[0][i] += now[0][j]\*by[j][i];

}

}

\*x=after[0][0];

\*y=after[0][1];

}

//旋转变换

void myrotate(int \*x,int \*y,float degree){

int now[1][3]={\*x,\*y,1};

float by[3][3]={cos(degree),sin(degree),0,

-sin(degree),cos(degree),0,

0,0,1};

float after[1][3]={0};

int i,j,k;

for(i=0;i<3;i++){

for(j=0;j<3;j++){

after[0][i] += now[0][j]\*by[j][i];

}

}

\*x=after[0][0];

\*y=after[0][1];

}

void setPixel(int x,int y){

glPointSize(1.0f);

glBegin(GL\_POINTS);

//translation(&x,&y,200,100);

//proportion(&x,&y,0.4,0.4);

//myrotate(&x,&y,PI/4);

glVertex2i(x,y);

glEnd();

glFlush();

}

void myBreseham(int x0,int y0,int xEnd,int yEnd)；中点画线算法（省略）

void myDisplay(void){

glClear(GL\_COLOR\_BUFFER\_BIT);// 清空显示窗口

glColor3f(0.0,0.0,0.0);// 指定前景色（当前绘制颜色）为黑色

myBreseham(-200,60,300,-90);

glFlush();// 使绘制立即反映到屏幕上

}

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

{

glutInit (&argc, argv); // 初始 GLUT.

glutInitDisplayMode (GLUT\_SINGLE | GLUT\_RGB); //设定显示模式

glutInitWindowPosition (500, 200); // 设定窗口位置

glutInitWindowSize (300, 300); // 设定窗口大小

glutCreateWindow ("An Example OpenGL Program");

// 用前面指定参数创建窗口，并定义窗口名称

init( ); // 进行一些初始化工作

glutDisplayFunc(myDisplay); // 指定绘制的回调函数

glutMainLoop ( ); // 进入无穷循环，等待事件处理

}

实验二：

#define \_STDCALL\_SUPPORTED

#include <GL/glut.h>

#include <math.h>

#include <stdlib.h>

#include <windows.h>

#include <stdio.h>

GLsizei winWidth = 400,winHeight = 300;

void init(void){

glClearColor(0.0,0.0,1.0,1.0);

glMatrixMode(GL\_PROJECTION);

gluOrtho2D(0.0,200.0,0.0,150.0);

}

void displayFcn(void){

glClear(GL\_COLOR\_BUFFER\_BIT);

glColor3f(1.0,0.0,0.0);

glPointSize(3.0);

}

void winReshapeFcn(GLint newWidth, GLint newHeight){

glViewport(0,0,newWidth,newHeight);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

gluOrtho2D(0.0,GLdouble(newWidth),0.0,GLdouble (newHeight));

winWidth = newWidth;

winHeight = newHeight;

}

void plotPoint(GLint x, GLint y){

glBegin(GL\_POINTS);

glVertex2i(x, y);

glEnd();

}

void mousePtPolt (GLint button, GLint action,GLint xMouse, GLint yMouse){

if(button == GLUT\_LEFT\_BUTTON && action == GLUT\_DOWN)

plotPoint(xMouse,winHeight - yMouse);

glFlush();

}

int main(int argc, char\*\* argv){

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);

glutInitWindowPosition(100,100);

glutInitWindowSize(winWidth, winHeight);

glutCreateWindow("Mouse Plot Points");

init();

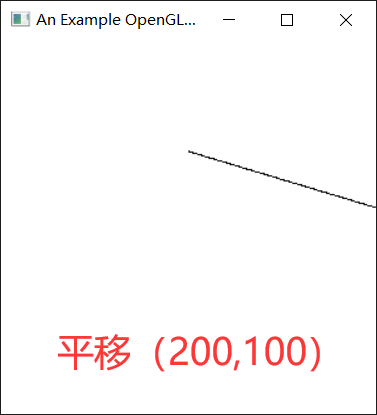
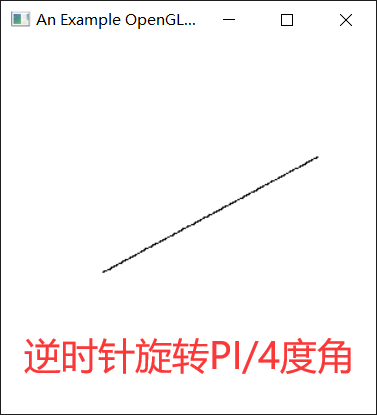
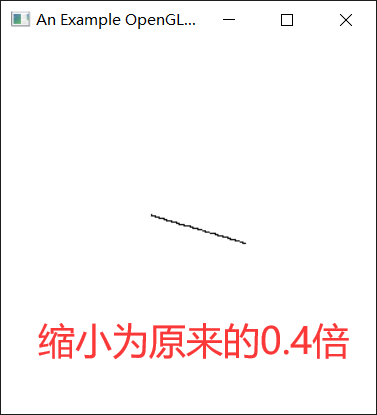
glutDisplayFunc(displayFcn);

glutReshapeFunc(winReshapeFcn);

glutMouseFunc(mousePtPolt);

glutMainLoop();

}

4、实验结果

实验二：

背景图案

低可信度描述已自动生成

5、实验总结

对图形进行变换，我们首先需要考虑的是把图像上的每一个点进行相应的变换，那么画线函数就应该是以点为元素，这就可以用到实验三中的Bresenham函数，对图形中的每一个点，与相应的矩阵进行乘积计算，即可得到变换后新点的坐标，再将新得到的点进行像素输出，即可得到变换后的图形。