**计算机图形学实验**

**姓 名：杨媛**

**学 号：20201050475**

**专 业：计算机科学与技术**

**教 师：钱文华**

实验九 三维图形几何变换实验

时间：2022.5.11

地点：信息学院2202

1、实验内容：

教材P222，三维图形旋转、缩放变换、平移变换、错切变换、对称变换等任意变换。

2、实验目的：

调用函数完成三维图形几何变换。

3、实验代码：

#include <windows.h>

#include <GL/glut.h>

float fTranslate;//平移因子

float fRotate;//旋转因子

float fScale=1.0f;//缩放因子

void Draw\_Cube(GLfloat x1, GLfloat x2, GLfloat y1,GLfloat y2,GLfloat z1,GLfloat z2)

{ //桌子由立方体组成，已知六个参数即可构造八个顶点

int i, j;

GLfloat vertex[8][3] = {

//八个顶点，从v1-v8

x1,y1,z2,

x2,y1,z2,

x1,y2,z2,

x2,y2,z2,

x1,y1,z1,

x2,y1,z1,

x1,y2,z1,

x2,y2,z1

};

GLint surface[6][4] = {

//v1对应0,以此类推

0,1,3,2,

4,5,7,6,

2,3,7,6,

0,1,5,4,

0,2,6,4,

1,3,7,5

glBegin(GL\_QUADS);//指定绘制方式

//将每个立方体分成六个面绘制

for (i = 0; i < 6; i++)

for (j = 0; j < 4; j++)

glVertex3fv(vertex[surface[i][j]]);

glEnd();

}

void Draw\_Table(){

//将桌子拆分成5个立方体绘制

Draw\_Cube(0.0, 1.0, 0.0, 0.8, 0.6, 0.8);//桌面

Draw\_Cube(0.1, 0.3, 0.1, 0.3, 0.0, 0.6);//四条腿

Draw\_Cube(0.7, 0.9, 0.1, 0.3, 0.0, 0.6);

Draw\_Cube(0.1, 0.3, 0.5, 0.7, 0.0, 0.6);

Draw\_Cube(0.7, 0.9, 0.5, 0.7, 0.0, 0.6);

}

void reshape(int width, int height){

//当窗口尺寸改变时改变尺寸

//三维变换——缩放

glPushMatrix();

glTranslatef(2.0f, 0.0f, -6.0f);//放在右边

glScalef(fScale, fScale,fScale);

Draw\_Table();

glPopMatrix();

//更新因子

fTranslate += 0.5f;

fRotate += 0.5f;

fScale -= 0.5f;

if(fTranslate > 0.5f)

fTranslate = 0.0f;

if (fScale < 0.5f)//缩小到一定程度之后变回原来大小

fScale = 1.0f;

glutSwapBuffers();

}

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

glutInit(&argc, argv);//对glut函数库进行初始化

glutInitDisplayMode(GLUT\_RGBA | GLUT\_DOUBLE);//指定glutCreateWindow函数将要创建的窗口显示模式 RGB 双缓冲

glutInitWindowSize(640,480);//窗口大小

glutCreateWindow("三维变换");//打开设置好的窗口，进入glutMainLoop之前这个窗口不会显示

glutDisplayFunc(redraw);//指定当前窗口需要重绘时调用的函数

glutReshapeFunc(reshape);//注册窗口大小改变时回调函数

glutIdleFunc(idle); //可以执行连续动画

glutMainLoop();//进入glut事件处理循环，永远不会返回

return 0;

}

if (height==0){

height=1;

}

glViewport(0,0,width,height);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

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

glMatrixMode(GL\_MODELVIEW);

glLoadIdentity();

}

void idle(){

glutPostRedisplay();

}

void redraw()

{

glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_LINE);

glClear(GL\_COLOR\_BUFFER\_BIT);//清除窗口

glLoadIdentity();

//三维变换——移动

glPushMatrix();//glpushmatrix与glpopmatrix配合使用可以消除上一次变换对本次变换的影响

glTranslatef(-2.5f, 0.0f,-6.0f);//放在左边

glTranslatef(0.0f, fTranslate, 0.0f);

Draw\_Table();

glPopMatrix();

//三维变换——旋转

glPushMatrix();

glTranslatef(0.0f, 0.0f,-6.0f);//放在中间

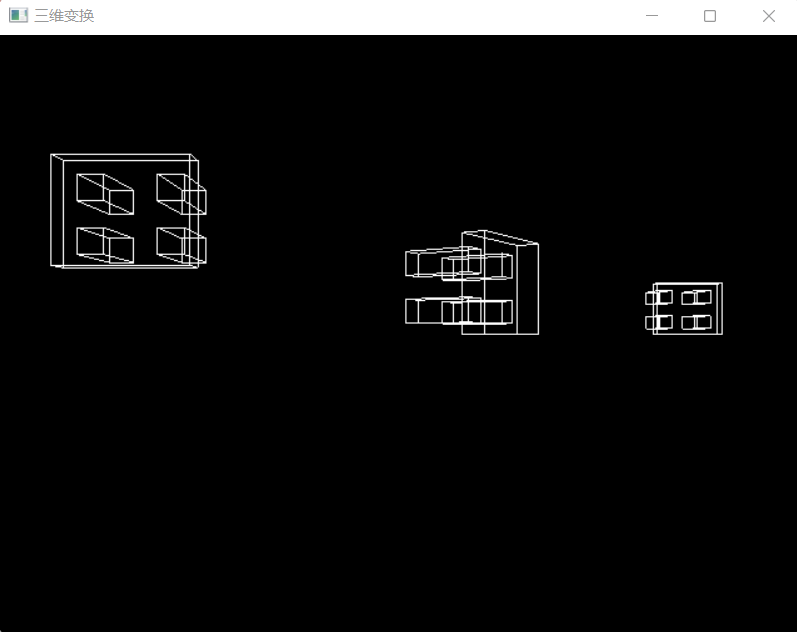
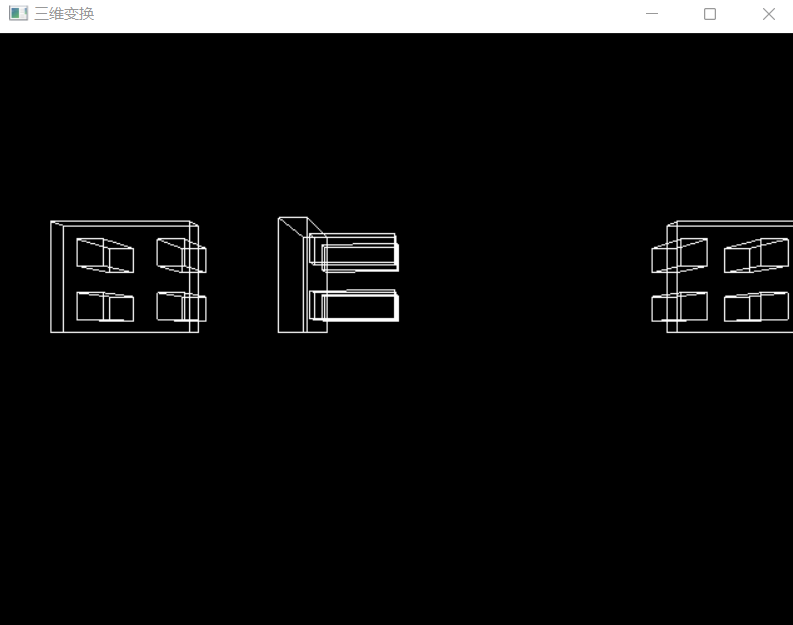
glRotatef(fRotate, 0, 1.0f, 0);

Draw\_Table();

glPopMatrix();

4、实验结果：

（从左到右依次为平移、旋转、缩放）



实验十 建模实验

时间：2022.5.18

地点：信息学院2202

1、实验内容：

使用opengl，教材P541,颜色编码建模显示。

2、实验目的：

调用函数完成颜色编码建模实验。

3、实验代码：

void mandelbrot(GLint nx,GLint ny,GLint maxIter){

complexNum z,zIncr;

color ptColor;

GLint iterCount;

zIncr.x = complexWidth/GLfloat(nx);

zIncr.y = complexHeight/GLfloat(ny);

for(z.x = xComplexMin; z.x<xComplexMax; z.x += zIncr.x){

for(z.y = yComplexMin; z.y<yComplexMax; z.y += zIncr.y){

iterCount = mandelSqTransf(z,maxIter);

if(iterCount>= maxIter)

ptColor.r = ptColor.g = ptColor.b = 0.0;

else if(iterCount>(maxIter/8)){

ptColor.r = 1.0;

ptColor.g = 0.5;

ptColor.b = 0.0;

}

else if(iterCount>(maxIter/10)){

ptColor.r = 1.0;

ptColor.g = ptColor.b = 0.0;

}

else if(iterCount>(maxIter/20)){

ptColor.b = 0.5;

ptColor.r = ptColor.g = 0.0;

}

else if(iterCount>(maxIter/40)){

ptColor.r = ptColor.g = 1.0;

ptColor.b = 0.0;

}

else if(iterCount>(maxIter/100)){

ptColor.r = ptColor.b = 0.0;

ptColor.g=0.3;

}

else{

ptColor.r = 0.0;

ptColor.g = ptColor.b = 1.0;

}

#include <windows.h>

#include <GL/glut.h>

GLsizei winWidth = 500,winHeight = 500;

GLfloat xComplexMin = -2.00,xComplexMax = 0.50;

GLfloat yComplexMin = -1.25,yComplexMax = 1.25;

GLfloat complexWidth = xComplexMax - xComplexMin;

GLfloat complexHeight = yComplexMax - yComplexMin;

class complexNum{

public:

GLfloat x,y;

};

struct color{ GLfloat r,g,b;};

void init (void){

glClearColor(1.0,1.0,1.0,0.0);

}

void plotPoint(complexNum z){

glBegin (GL\_POINTS);

glVertex2f(z.x,z.y);

glEnd();

}

complexNum complexSquare (complexNum z){

complexNum zSquare;

zSquare.x = z.x\*z.x-z.y\*z.y;

zSquare.y = 2\*z.x\*z.y;

return zSquare;

}

GLint mandelSqTransf(complexNumz0, GLint maxIter){

complexNum z = z0;

GLint count = 0; while((z.x\*z.x+z.y\*z.y<=4.0)&&(count<maxIter)){

z = complexSquare(z);

z.x += z0.x;

z.y+=z0.y;

count++;

}

return count;

}

glColor3f(ptColor.r,ptColor.g,ptColor.b);

plotPoint(z);

}

}

}

void displayFcn(void){

GLint nx = 1000,ny = 1000,maxIter = 1000;

glClear(GL\_COLOR\_BUFFER\_BIT);

mandelbrot (nx,ny,maxIter);

glFlush();

}

void winReshapeFcn(GLint newWidth,GLint newHeight){

glViewport(0,0,newHeight,newHeight);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

gluOrtho2D(xComplexMin,xComplexMax,yComplexMin,yComplexMax);

glClear(GL\_COLOR\_BUFFER\_BIT);

}

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

glutInit (&argc, argv); // 初始 GLUT.

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

glutInitWindowPosition (50, 50); // 设定窗口位置

glutInitWindowSize (winWidth,winHeight); // 设定窗口大小

glutCreateWindow ("Mandelbrot Set");

// 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

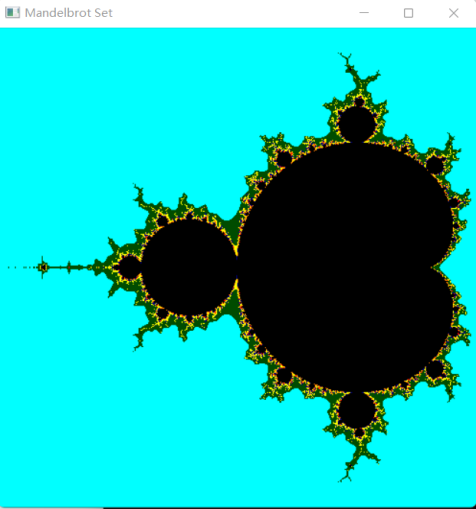
glutDisplayFunc(displayFcn); // 指定绘制的回调函数

glutReshapeFunc(winReshapeFcn);

glutMainLoop ( );

return 0; // 进入无穷循环，等待事件处理

}

4、实验结果：

实验十一 交互控制实验

时间：2022.5.18

地点：信息学院2202

1、实验内容：

使用opengl，完成鼠标、键盘交互操作。

2、实验目的：

熟悉鼠标、键盘交互。

3、实验代码：

#include<GL/glut.h>

#include<stdio.h>

GLsizei winWidth = 500,winHeight = 500;

char sixel;

float thera=0;

float x=0,y=0,z=0;

void init(void){

glClearColor(1.0,1.0,1.0,0.0);

}

void displayWirePolyhedra(float x,float y,float z,float thera){

glClear(GL\_COLOR\_BUFFER\_BIT);

glColor3f(0.0,0.0,1.0);

gluLookAt(5.0,5.0,5.0,0.0,0.0,0.0,0.0,1.0,0.0);

glScalef(1.0,1.0,1.0);

glTranslatef(1.0,2.0,0.0);//下一个图形坐标

glutSolidTeapot(1.5);

//glutWireTeapot(1.5);//放大倍数

glScalef(1.0,1.0,1.0);//缩放比

glTranslatef(-1.0,-5.0,0.0);//下一个图形坐标

glRotatef(thera,x,y,z);

glutWireTeapot(1.5);

//glutSolidTeapot(2.0);

glFlush();

}

void display(){

displayWirePolyhedra(x,y,z,thera);

}

void winReshapeFcn(GLint newWidth,GLint newHeight){

glViewport(0,0,newWidth,newHeight);

glMatrixMode(GL\_PROJECTION);

glFrustum(-1.0,1.0,-1.0,1.0,2.0,20.0);

glMatrixMode(GL\_MODELVIEW);

glClear(GL\_COLOR\_BUFFER\_BIT);

}

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

glutInit(&argc,argv);

glutInitDisplayMode(GLUT\_SINGLE|GLUT\_RGB);

glutInitWindowPosition(100,100);

glutInitWindowSize(winWidth,winHeight);

glutCreateWindow(" ");

init();

printf("请选择绕哪一个轴旋转x,y,z \n");

scanf("%c",&sixel);

getchar();

if(sixel=='x'){

x=1.0;

y=0.0;

z=0.0;

printf("请输入旋转的角度\n");

scanf("%f",&thera);

}else if(sixel=='y'){

x=0.0;

y=1.0;

z=0.0;

printf("请输入旋转的角度\n");

scanf("%f",&thera);

}else if(sixel=='z'){

x=0.0;

y=0.0;

z=1.0;

printf("请输入旋转的角度\n");

scanf("%f",&thera);

}else{

printf("输入有误\n");

}

glutDisplayFunc(display);

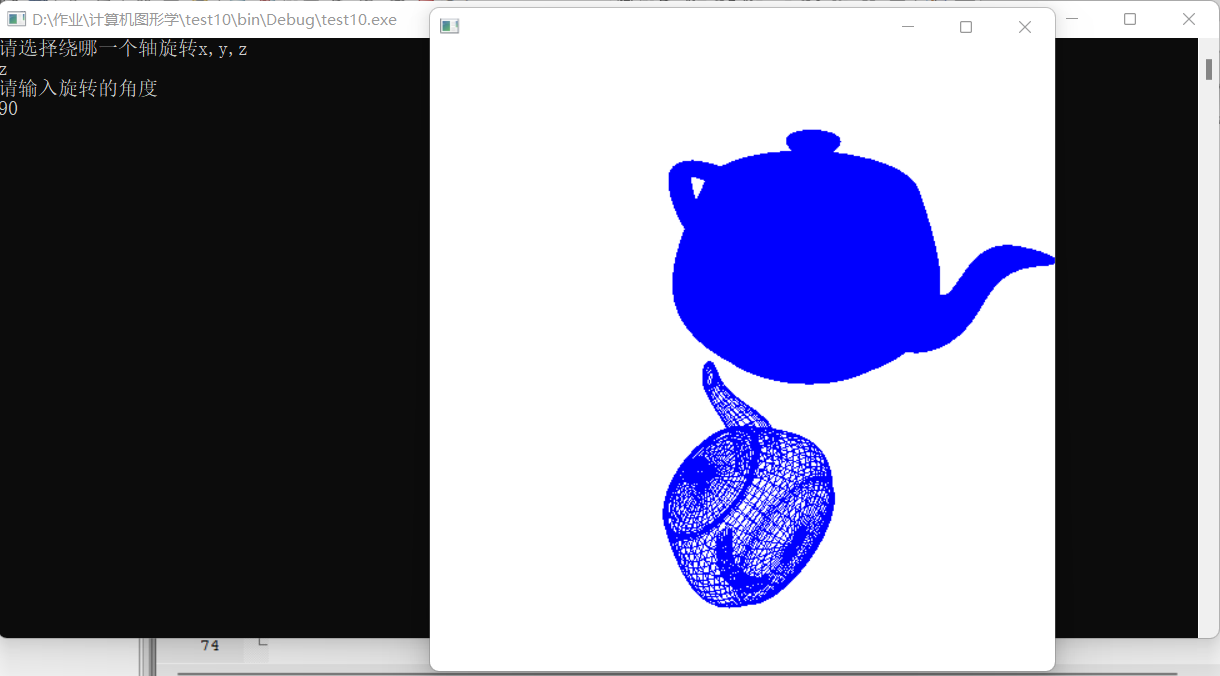
glutReshapeFunc(winReshapeFcn);

glutMainLoop();

return 0;

}

4、实验结果：



实验十二 三维观察实验

时间：2022.3.23

地点：信息学院2202

1、实验内容：

使用opengl,完成投影变换等实验（P264）。

2、实验目的：

熟悉三维观察相关内容。

3、实验代码：

#include <windows.h>

#include <GL/glut.h>

GLint winWidth=600,winHeight=600;

GLfloat x0=100.0,y0=50.0,z0=50.0;

GLfloat xref=50.0,yref=50.0,zref=0.0;

GLfloat Vx=0.0,Vy=1.0,Vz=0.0;

GLfloat xwMin=-40.0,ywMin=-60.0,xwMax=40.0,ywMax=60.0;

GLfloat dnear=25.0,dfar=125.0;

void init (void){

glClearColor(1.0,1.0,1.0,0.0);

glMatrixMode (GL\_MODELVIEW);

gluLookAt (x0,y0,z0,xref,yref,zref,Vx,Vy,Vz);

glMatrixMode (GL\_PROJECTION);

glFrustum(xwMin,xwMax,ywMin,ywMax,dnear,dfar);

}

void displayFcn(void){

glClear (GL\_COLOR\_BUFFER\_BIT);

glColor3f(0.0,1.0,0.0);

glPolygonMode(GL\_FRONT,GL\_FILL);

glPolygonMode(GL\_BACK,GL\_LINE);

glBegin(GL\_QUADS);

glVertex3f(0.0,0.0,0.0);

glVertex3f(100.0,0.0,0.0);

glVertex3f(100.0,100.0,0.0);

glVertex3f(0.0,100.0,0.0);

glEnd();

glFlush();

}

void reshapeFcn (GLint newWidth, GLint newHeight){

glViewport (0,0,newWidth,newHeight);

winWidth=newWidth;

winHeight=newHeight;

}

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

glutInit (&argc,argv);

glutInitDisplayMode (GLUT\_SINGLE|GLUT\_RGB);

glutInitWindowPosition (50,50);

glutInitWindowSize(winWidth, winHeight);

glutCreateWindow ("Perspective View of A Square");

init();

glutDisplayFunc(displayFcn);

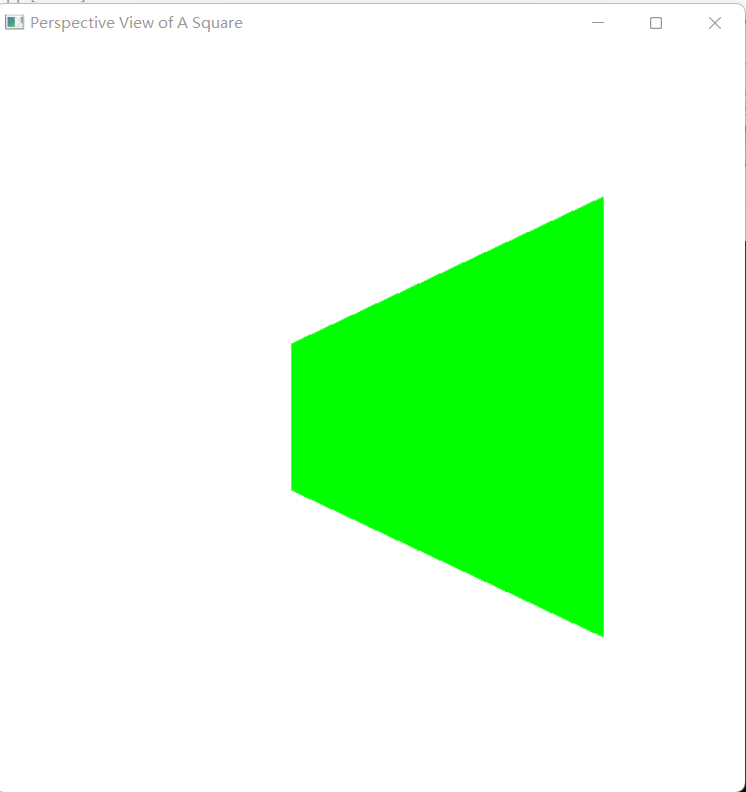
glutReshapeFunc(reshapeFcn);

glutMainLoop();

return 0;

}

4、实验结果：



实验十三 多面体实验

时间：2022.6.1

地点：信息学院2202

1、实验内容：

生成多面体线框图，P300、P307。

2、实验目的：

熟悉三维线框图相关内容。

3、实验代码：

#include <windows.h>

#include <GL/glut.h>

//P300

GLsizei winWidth=500,winHeight=500;

void init (void){

glClearColor (1.0,1.0,1.0,0.0);

}

void displayWirePolyhedra (void){

#include <windows.h>

#include <GL/glut.h>

//P307

GLsizei winWidth=500,winHeight=500;

void init (void){

glClearColor (1.0,1.0,1.0,0.0);

}

glClear (GL\_COLOR\_BUFFER\_BIT);

glColor3f (0.0,0.0,1.0);

gluLookAt (5.0,5.0,5.0,0.0,0.0,0.0,0.0,1.0,0.0);

glScalef (1.5,2.0,1.0);

glutWireCube (1.0);

glScalef (0.8,0.5,0.8);

glTranslatef (-6.0,-5.0,0.0);

glutWireDodecahedron();

glTranslatef (8.6,8.6,2.0);

glutWireTetrahedron ();

glTranslatef (-3.0,-1.0,0.0);

glutWireOctahedron ();

glScalef (0.8,0.8,1.0);

glTranslatef (4.3,-2.0,0.5);

glutWireIcosahedron();

glFlush();

}

void winReshapeFcn (GLint newWidth,GLint newHeight){

glViewport (0,0,newWidth,newHeight);

glMatrixMode (GL\_PROJECTION);

glFrustum(-1.0,1.0,-1.0,1.0,2.0,20.0);

glMatrixMode (GL\_MODELVIEW);

glClear(GL\_COLOR\_BUFFER\_BIT);

}

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

glutInit (&argc,argv);

glutInitDisplayMode(GLUT\_SINGLE|GLUT\_RGB);

glutInitWindowPosition (100,100);

glutInitWindowSize(winWidth,winHeight);

glutCreateWindow ("Wire-Frame Po1yhedra");

init();

glutDisplayFunc (displayWirePolyhedra);

glutReshapeFunc (winReshapeFcn);

glutMainLoop();

}

void wireQuadSurfs (void){

glClear (GL\_COLOR\_BUFFER\_BIT);

glColor3f (0.0,0.0,1.0);

gluLookAt (2.0,2.0,2.0,0.0,0.0,0.0,0.0,0.0,1.0);

glPushMatrix();

glTranslatef (1.0,1.0,0.0);

glutWireSphere (0.75,8,6);

glPopMatrix();

glPushMatrix();

glTranslatef (1.0,-0.5,0.5);

glutWireCone (0.7,2.0,7,6);

glPopMatrix();

GLUquadricObj\* cylinder;

glPushMatrix();

glTranslatef (0.0,1.2,0.8);

cylinder=gluNewQuadric();

gluQuadricDrawStyle(cylinder,GLU\_LINE);

gluCylinder(cylinder,0.6,0.6,1.5,6,4);

glPopMatrix();

glFlush();

}

void winReshapeFcn (GLint newWidth,GLint newHeight){

glViewport (0,0,newWidth,newHeight);

glMatrixMode (GL\_PROJECTION);

glOrtho (-2.0,2.0,-2.0,2.0,0.0,5.0);

glMatrixMode (GL\_MODELVIEW);

glClear(GL\_COLOR\_BUFFER\_BIT);

}

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

glutInit (&argc,argv);

glutInitDisplayMode (GLUT\_SINGLE|GLUT\_RGB);

glutInitWindowPosition (100,100);

glutInitWindowSize (winWidth, winHeight);

glutCreateWindow("wire-Frame Quadric Surfaces");

init ();

glutDisplayFunc (wireQuadSurfs);

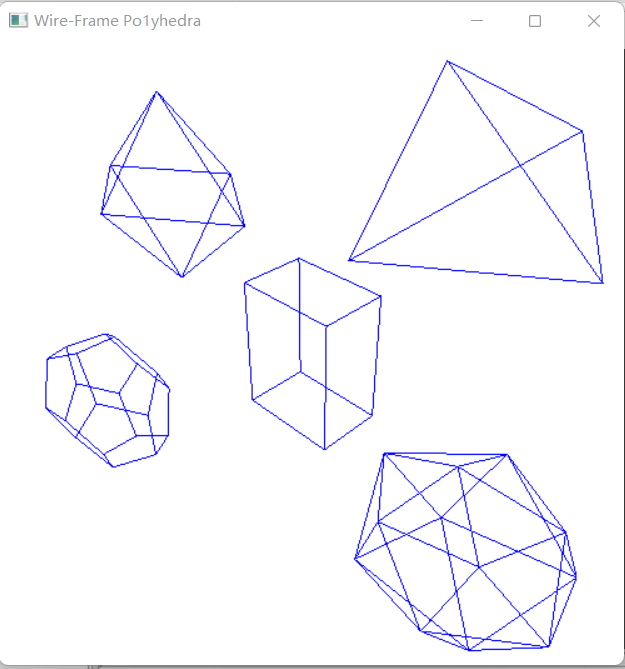
glutReshapeFunc (winReshapeFcn);

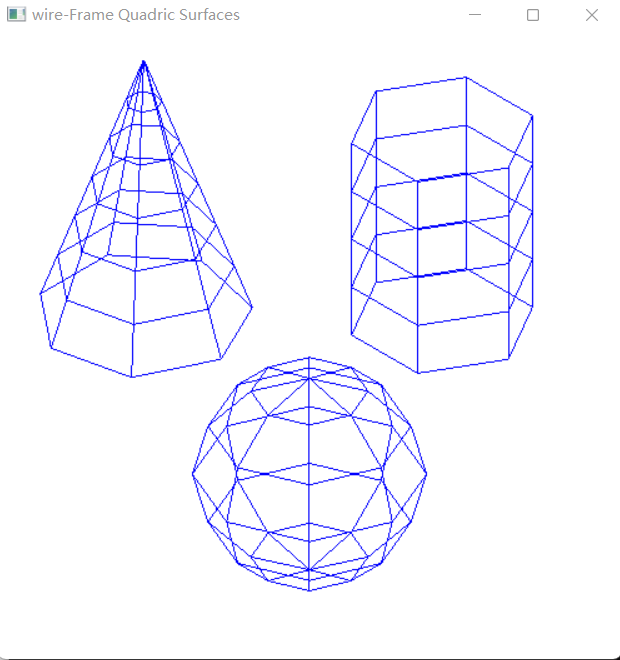
glutMainLoop ();

}

4、实验结果：

P300：

P307：



实验十四 曲线曲面生成实验

时间：2022.6.8

地点：信息学院2202

1、实验内容：

生成曲线或者曲面，P323。

2、实验目的：

熟悉Bezier、样条等相关内容。

3、实验代码：

#include <windows.h>

#include <GL/glut.h>

#include <cstdio>

#include <cmath>

const GLfloat Pi = 3.1415926536f;

//定义点集

struct data{

GLfloat x;

GLfloat y;

}Point[4];

void init() //初始化函数

{

glClearColor(1.0, 1.0, 1.0, 0.0); //设置背景颜色

glMatrixMode(GL\_PROJECTION); // 设置投影参数

gluOrtho2D(0.0, 100.0, 0.0, 100.0); // 设置场景的大小

}

void Bernstain() //Bernstain基函数

{

glClear(GL\_COLOR\_BUFFER\_BIT);

glTranslatef(30.0f, 30.0f, 0.0f); //平移图形

glScalef(0.1f, 0.1f, 0.1f); //缩小图像0.5倍

//四个点作为控制顶点

Point[0].x = 0.0;

Point[0].y = 0.0;

Point[1].x = 200.0;

Point[1].y = 100.0;

Point[2].x = 300.0;

Point[2].y = 100.0;

Point[3].x = 400.0;

Point[3].y = 0.0;

glColor3f(0.0, 0.0, 1.0); //设置线条颜色

glPointSize(2); //设置点的大小

glBegin(GL\_LINES);

glVertex2f(Point[0].x, Point[0].y);

glVertex2f(Point[1].x, Point[1].y);

glVertex2f(Point[1].x, Point[1].y);

glVertex2f(Point[2].x, Point[2].y);

glVertex2f(Point[2].x, Point[2].y);

glVertex2f(Point[3].x, Point[3].y);

glEnd();

glFlush();

glColor3f(1.0, 0.0, 0.0); //设置线条颜色

glPointSize(2); //设置点的大小

glBegin(GL\_LINE\_STRIP);

for (int i = 1; i <= 200; i++){

GLfloat t = i / 200.0;

GLfloat b0 = pow(1 - t, 3.0);

GLfloat b1 = 3.0 \* t\*pow(1 - t, 2.0);

GLfloat b2 = 3.0 \* t\*t\*(1 - t);

GLfloat b3 = t\*t\*t;

//运用三次Bezier曲线

GLfloat x = Point[0].x\*b0 + Point[1].x\*b1 + Point[2].x\*b2 + Point[3].x\*b3;

GLfloat y = Point[0].y\*b0 + Point[1].y\*b1 + Point[2].y\*b2 + Point[3].y\*b3;

glVertex2f(x, y);

}

glEnd();

glFlush();

}

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

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_RGB | GLUT\_SINGLE);

glutInitWindowPosition(100, 100);

glutInitWindowSize(400,400);

glutCreateWindow("Bezier曲线");

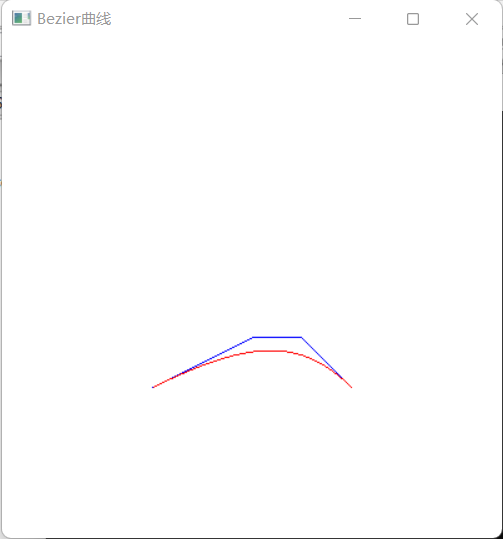
init();

glutDisplayFunc(&Bernstain);

glutMainLoop();

return 0;

}

4、实验结果：

实验十五 消隐实验

时间：2022.6.15

地点：信息学院2202

1、实验内容：

完成消隐实验，采用Z-buffer算法完成消隐。

2、实验目的：

熟悉Z-buffer、画家算法等相关内容。

#include<windows.h>

#include<GL/glut.h>

GLfloat pos[]={-2,4,5,1},amb[]={0.3,0.3,0.3,1.0} ;

GLfloat front\_amb\_diff[]={0.7,0.5,1.1,1.0};

GLfloat back\_amb\_diff[]={0.4,0.7,1,1.0} ;GLfloat spe[]={0.25,0.25,0.25,1.0} ;

GLfloat theta=0, dt=1.5, axes[3][3]={{1,0,0},{0,1,0},{0,0,1}} ;

int axis=0;

void display(void){

glClear(GL\_COLOR\_BUFFER\_BIT|GL\_DEPTH\_BUFFER\_BIT);

glPushMatrix() ;

if(axis<3)

glRotated(theta, axes[axis][0], axes[axis][1], axes[axis][2]);

else{

glPushMatrix () ;

glRotated(theta,0,0,1);

glLightfv(GL\_LIGHT0,GL\_POSITION, pos);

glPopMatrix ();

}

glutSolidTorus(4,1,48,96);//图片调用函数可以设置图片的属性

glPopMatrix () ;

glutSwapBuffers () ;

}

void idle(void){

if(theta>=360)

axis=(axis+1)%4;

theta=(theta<360)?theta+dt : dt;

glutPostRedisplay () ;

3、实验代码：

}

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

glutInit(&argc, argv) ;

glutInitDisplayMode(GLUT\_DOUBLE|GLUT\_RGB|GLUT\_DEPTH);

glutInitWindowSize (500,500);

glutInitWindowPosition(200,100) ;

glutCreateWindow ("z-buffer" ) ;

glClearColor(1,1,0,0);//背景颜色黄色

glMatrixMode(GL\_PROJECTION) ;

glLoadIdentity() ;

gluPerspective (45,1.0,2,8) ;

glMaterialfv(GL\_FRONT,GL\_AMBIENT\_AND\_DIFFUSE,front\_amb\_diff);

glMaterialfv(GL\_BACK,GL\_AMBIENT\_AND\_DIFFUSE, back\_amb\_diff);

glMaterialfv(GL\_FRONT\_AND\_BACK,GL\_SPECULAR,spe);

glMaterialf(GL\_FRONT\_AND\_BACK,GL\_SHININESS,75);

glLightfv(GL\_LIGHT0,GL\_AMBIENT, amb);

glLightModeli(GL\_LIGHT\_MODEL\_TWO\_SIDE,GL\_TRUE);

glMatrixMode (GL\_MODELVIEW);

glLoadIdentity() ;

glTranslated(0,0,-5);

glLightfv(GL\_LIGHT0,GL\_POSITION,pos) ;

glEnable(GL\_LIGHTING);

glEnable(GL\_LIGHT0) ;

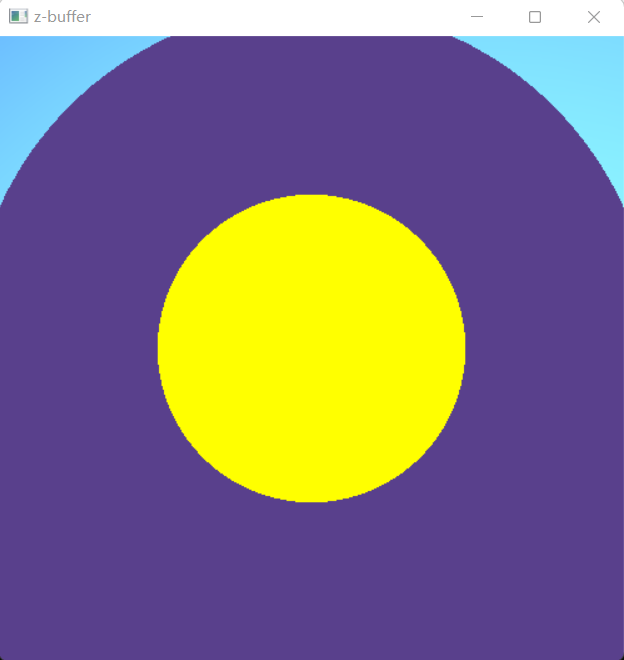
glutDisplayFunc(display);

glutIdleFunc(idle);

glutMainLoop () ;

return 0;

}

4、实验结果：