Лабораторная работа № 3

Белов ВА, гр. 4210

Листинг программы:  
#include <gl\glut.h>

#include <math.h>

#include <windows.h>

GLfloat angle = 0;

GLfloat resize = 0.5;

GLfloat posz = 0;

GLfloat dx = 0;

void init(void)

{

glClearColor(1.0, 0.5, 1.0, 0.0);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

gluPerspective(60, 1, 1, 10);

glMatrixMode(GL\_MODELVIEW);

glLoadIdentity();

}

void myDisplay()

{

glPushMatrix(); //Сохраняем VM = 1

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

glEnable(GL\_DEPTH\_TEST);

gluLookAt(3, 1, 1, 0, 0, 0, 0, 1, 0); // VM=Fwe позиция наблюдателя

GLfloat myLightPosition[] = { 100.0, 55.0, 6.0, 1.0 }; // Источник света в CKw

glLightfv(GL\_LIGHT0, GL\_POSITION, myLightPosition); /\*Позиция источника света будет преобразована в CKe\*/

glEnable(GL\_LIGHTING);

glEnable(GL\_LIGHT0);

glPushMatrix(); //Сохраняем VM=Fwe

//glRotatef(angle, -1, 0, 0); // VM=Fwe\*R

glRotatef(angle, 0, 0, 1); // VM=Fwe\*R\*R

//glRotatef(angle, 0, 0, -1); // VM=Fwe\*R\*R\*R

//glutWireCube(0.95);

glScalef(resize, resize, resize); //размеры октоэдра

glutSolidOctahedron(); //октоэдр

glPopMatrix(); // Восстанавливаем VM=Fwe

glDisable(GL\_LIGHTING); //Выключаем освещение

glBegin(GL\_LINES);

glColor3f(1, 0, 0); glVertex3f(0, 0, 0); glVertex3f(1, 0, 0);

glColor3f(0, 1, 0); glVertex3f(0, 0, 0); glVertex3f(0, 1, 0);

glColor3f(0, 0, 1); glVertex3f(0, 0, 0); glVertex3f(0, 0, 1);

glEnd();

//glBegin(GL\_LINES);//белая линия

//glColor3f(1, 1, 1); glVertex3f(100, 55, 6); glVertex3f(0.0, 0.0, 0.0);

//glEnd();

glPopMatrix();

glutSwapBuffers();

Sleep(30); //скорость вращения

}

void myResMB(int button, int state, int x, int y) // изменение размера фигуры нажатием клавиш мышки

{

if (state == GLUT\_UP)

{

if (button == GLUT\_LEFT\_BUTTON)

{

resize \*= 1.5;

}

if (button == GLUT\_RIGHT\_BUTTON)

{

resize \*= 0.5;

}

}

}

void myResKB(unsigned char key, int x, int y) // изменение вращения по клавишам Х и Y

{

switch (key)

{

case 'x':

// Наблюдатель вращается в пл. (осьZ, Е) вокруг объекта по ч.с.

glMatrixMode(GL\_MODELVIEW);

glTranslatef(0, 0, -3);

glRotatef(2, 0, 0, 1);

glTranslatef(0, 0, 3);

glutPostRedisplay();

break;

case 'y':

// Наблюдатель вращается в пл. (осьZ, Е) против ч.с.

while (true) {

glMatrixMode(GL\_MODELVIEW);

glTranslatef(0, 0, 0);

glRotatef(-2, 0, 0, 1);

glTranslatef(0, 0, 3);

glutPostRedisplay();

}

break;

default:

break;

}

}

void myReshape(int width, int height)

{

if (width / height < 1) glViewport(0, 0, width, width);

else glViewport(0, 0, height, height);

}

void myIdle()

{

angle += dx;

if (angle > 360.0) angle = 0;

glutPostRedisplay();

}

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

{

glutInit(&argc, argv);

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

glutInitWindowSize(800, 800);

glutInitWindowPosition(0, 0);

glutCreateWindow("my window");

glutDisplayFunc(myDisplay);

glutMouseFunc(myResMB); //нажатие мыши

glutKeyboardFunc(myResKB); //клавиатура

glutReshapeFunc(myReshape);

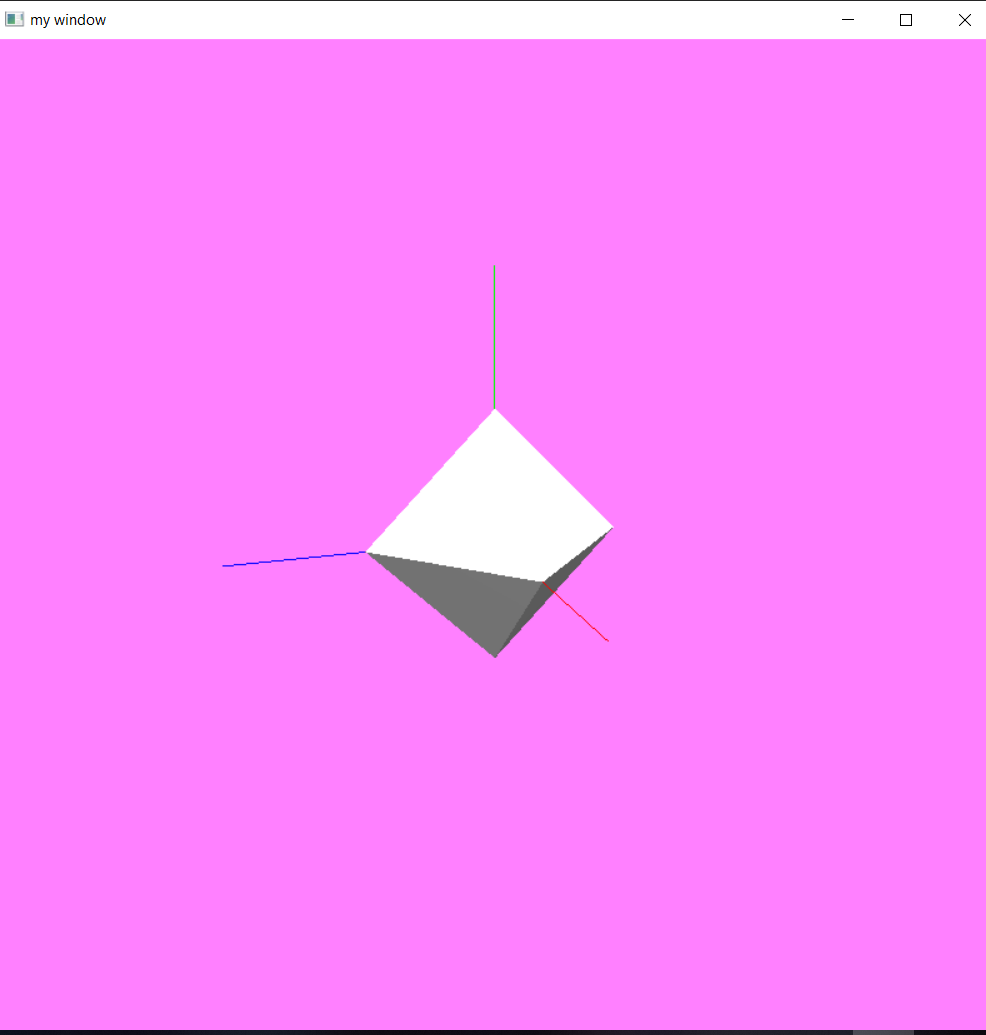
glutIdleFunc(myIdle);

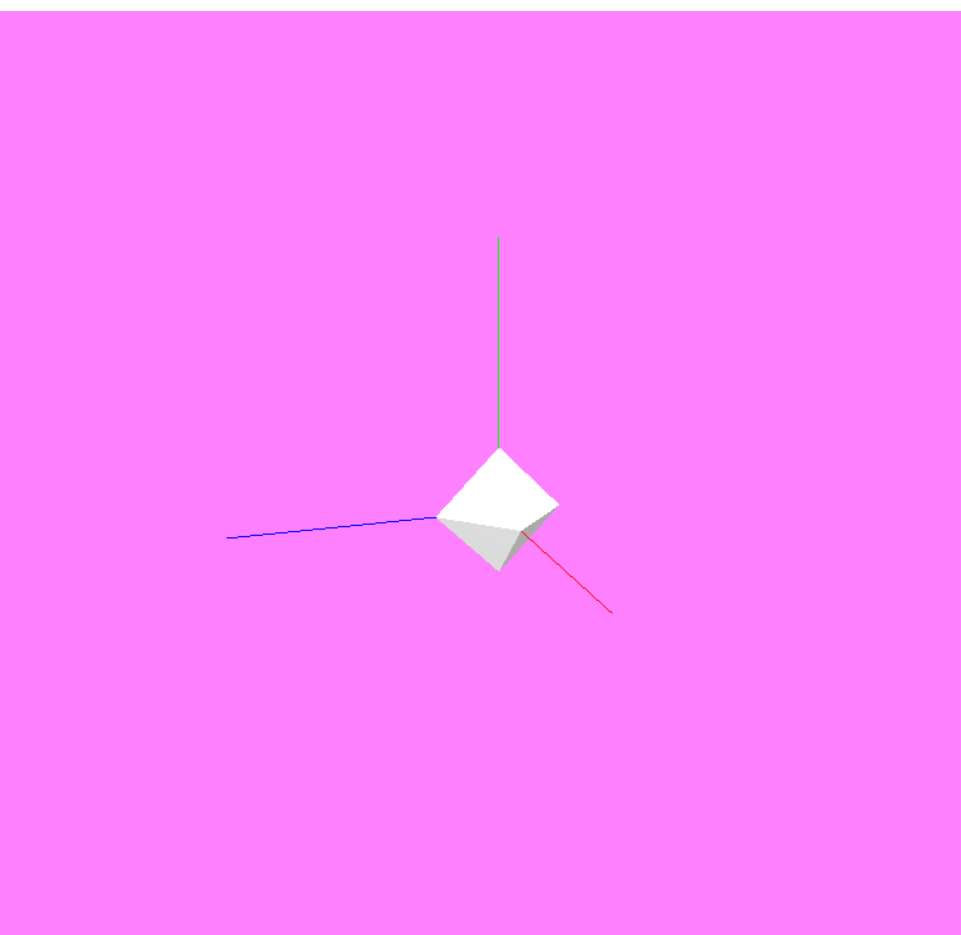
init();

glutMainLoop();

}

Работа программы:

Исходный вариант  


Уменьшение:  
  
Увеличение:  


Вращение наблюдателя по оси Z:

