***ЛАБОРАТОРНАЯ РАБОТА №5***

**По дисциплине «Алгоритмические средства компьютерной графики»**

***ТЕМА. Каркасные трехмерные модели***

**Цель работы:** Попрактиковаться в задании и отображении трехмерных каркасных объектов в OpenGL

**Порядок работы**

**Задание 1.**

* 1. В среде Visual Studio создайте пустое решение (solution) с именем PW6. В нем создайте консольный проект на языке C++ Win32 Console Application. Назовите его PW6\_1.
  2. В проект добавьте пустой файл с исходным кодом source.cpp
  3. Наберите следующий текст программы:

#include<Windows.h>

#include <GL/glut.h>

#include<iostream>

#include<fstream>

#include<vector>

using namespace std;

struct Point

{

float x, y,z;

};

vector<Point> v;

vector<vector<int>> faces;

int point\_count, face\_count;

void reshape(int w, int h);

void display();

void processNormalKeys(unsigned char key, int x, int y);

void processSpecialKeys(int key, int x, int y);

void readfromfile()

{

float x, y,z;

Point p;

ifstream f("object\_data.txt", ios::in);

f >> point\_count >> face\_count;

for (int i = 0; i < point\_count; i++)

{

f >> p.x >> p.y >> p.z;

v.push\_back(p);

}

faces.resize(face\_count);

for (int i = 0; i < face\_count; i++)

{

int n;

f >> n;

for (int j = 0; j < n; j++)

{

int p;

f >> p;

faces[i].push\_back(p);

}

}

f.close();

}

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

{

readfromfile();

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_DOUBLE | GLUT\_RGBA);

glutInitWindowSize(800, 600);

glutCreateWindow("OpenGL lesson 7");

glutReshapeFunc(reshape);

glutDisplayFunc(display);

glutKeyboardFunc(processNormalKeys);

glutSpecialFunc(processSpecialKeys);

glutMainLoop();

return 0;

}

void reshape(int w, int h)

{

glViewport(0, 0, w, h);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

glOrtho(0, 10, 0, 10,-10,10);

glMatrixMode(GL\_MODELVIEW);

glLoadIdentity();

}

void display()

{

glClearColor(1, 1, 1, 0);

glClear(GL\_COLOR\_BUFFER\_BIT);

glColor3f(1.0, 0.0, 0.0);

for (int i = 0; i < face\_count; i++)

{

glBegin(GL\_LINE\_LOOP);

for (int j = 0; j < faces[i].size(); j++)

{

glVertex3f(v[faces[i][j]].x, v[faces[i][j]].y, v[faces[i][j]].z);

}

glEnd();

}

glutSwapBuffers();

}

void processNormalKeys(unsigned char key, int x, int y)

{

if (key == 27)

exit(0);

if (key == 65)

{

glMatrixMode(GL\_MODELVIEW);

glTranslated(2, 2, 0);

display();

}

}

void processSpecialKeys(int key, int x, int y) {

switch (key) {

case GLUT\_KEY\_UP:

glMatrixMode(GL\_MODELVIEW);

glTranslated(0, 2, 0);

display();

break;

case GLUT\_KEY\_DOWN:

glMatrixMode(GL\_MODELVIEW);

glRotated(5,1, 1, 1);

display();

break;

}

}

* 1. Создайте текстовый файл object\_data.txt в папке проекта. Заполните файл следующим данными (количество вершин, количество граней, координаты вершин, количество вершин в грани, номера вершин):

8 6

1 1 0 9 1 0 9 5 0 1 5 0

1 1 1 9 1 1 9 5 1 1 5 1

4 0 1 2 3

4 4 7 6 5

4 4 5 1 0

4 2 6 7 3

4 5 6 2 1

4 0 3 7 4

* 1. Запустите программу. Нажмите клавиши UP, DOWN и ESC. Проанализируйте работу программы. Изучите текст программы.

**Задание 2.**

* 1. Используйте разработанную в задании 1 программу для отображения Ваших инициалов (первых букв Вашего имени и фамилии) в 3Д в виде каркасных объектов. Например, как показано на рисунке 1а. На рисунке 1б. показано как изменится изображение буквы при вращении объекта.

|  |  |
| --- | --- |
|  |  |
| 1а | 1б |

Для этого в папке проекта создайте новый текстовый файл. Внесите в него соответствующим образом координаты вершин и информацию о гранях для Ваших объектов (инициалов). В тексте программы измените название файла.

* 1. Запустите и протестируйте работу программы
  2. Задайте обработку следующих клавиш:
     + LEFT – движение влево,
     + RIGHT – движение вправо,
     + + - увеличение объектов,
     + - - уменьшение объектов,
     + HOME – вращение объектов против часовой стрелки вокруг оси OX,
     + END – вращение объектов против часовой стрелки вокруг оси OY,
     + DELETE – вращение объектов против часовой стрелки вокруг оси OZ,
     + PG UP – вращение объектов по часовой стрелки вокруг всех трех осей одновременно,
     + PG DN – вращение объектов против часовой стрелки вокруг всех трех осей одновременно

Запустите приложение и проанализируйте его работу.

* 1. Измените текст программы таким образом, что бы отображаемая проекция объекта была центральной. Сравните полученный результат работы программы с предыдущими запусками.