**Міністерство освіти і науки України**

**Запорізький національний технічний університет**

кафедра програмних засобів

**ЗВІТ**

з лабораторної роботи №2

з дисципліни «Методи та алгоритми комп'ютерної графіки» на тему:

"Малювання тривимірних об’єктів. Обертання"

Варіант 22

Виконала:

студентка групи КНТ-415 В.С. Хохлова

Прийняв:

асистент Ж.К. Камінська

2018

1. Мета роботи

Навчитися працювати з тривимірними фігурами та їх положенням.

1. Завдання

Виконати побудову перших букв імені і прізвища (англійською) у тривимірному вигляді, використовуючи функції повороту примітиву.

1. Хід роботи

#include "my\_paint.h"

#include <QtOpenGL>

#include <GL/gl.h>

#include <GL/glut.h>

#include <iostream>

const GLfloat z1 = 0.2f;

const GLfloat z2 = 0.3f;

const GLfloat vertices\_V[7][2] = {

{0.3f, 0.1f},

{0.1f, 0.7f},

{0.2f, 0.7f},

{0.35f, 0.25f},

{0.5f, 0.7f},

{0.6f, 0.7f},

{0.4f, 0.1f},

};

const GLfloat vertices\_K[13][2] = {

{0.75f, 0.1f},

{0.75f, 0.7f},

{0.85f, 0.7f},

{0.85f, 0.45f},

{0.9f, 0.45f},

{1.05f, 0.7f},

{1.15f, 0.7f},

{0.975f, 0.4f},

{1.15f, 0.1f},

{1.05f, 0.1f},

{0.9f, 0.35f},

{0.85f, 0.35f},

{0.85f, 0.1f},

};

void **drawFilledSquare**(const GLfloat vertices[][2], int v1, int v2, int v3, int v4, GLfloat z1, GLfloat z2) {

glVertex3f(vertices[v1][0], vertices[v1][1], z1);

glVertex3f(vertices[v2][0], vertices[v2][1], z1);

glVertex3f(vertices[v3][0], vertices[v3][1], z2);

glVertex3f(vertices[v4][0], vertices[v4][1], z2);

}

void **DrawLineStrip**(const GLfloat vertices[][2], int n, GLfloat z) {

// (GL\_LINE\_STRIP)-лінія з 2х точок

glBegin(GL\_LINE\_STRIP);

for(int i = 0; i < n; i++) {

glVertex3f(vertices[i][0], vertices[i][1], z);

}

glVertex3f(vertices[0][0], vertices[0][1], z);

glEnd();

}

void **DrawLineZ**(const GLfloat vertices[][2], int n){

// (GL\_LINE\_STRIP)-лінія з 2х точок

for(int i = 0; i < n; i++) {

glBegin(GL\_LINE\_STRIP);

glVertex3f(vertices[i][0], vertices[i][1], z1);

glVertex3f(vertices[i][0], vertices[i][1], z2);

glEnd();

}

}

My\_Paint::**My\_Paint**(QWidget \*parent) // конструктор

: QGLWidget(parent) {

resize(1000, 600);

//resize(1000, 600);

xRotation = 0;

yRotation = 0;

zRotation = 0;

scale = 1;

}

void My\_Paint::***initializeGL***() {

qglClearColor(QColor(245, 245, 208)); // заполняем фон

glEnable(GL\_DEPTH\_TEST); // задаем глубину проверки пикселей - будут ли видны противоположные грани

glShadeModel(GL\_FLAT); // убираем режим сглаживания цветов

glPolygonMode(GL\_BACK, GL\_FRONT\_FACE); // фигуры будут закрашены с обеих сторон

// Дозволяємо очищення буфера глибини

glClearDepth (1.0); // Дозволяємо очищення буфера глибини

// Тип тесту глибини

glDepthFunc (GL\_LEQUAL);

// Поліпшення в обчисленні перспективи

glHint (GL\_PERSPECTIVE\_CORRECTION\_HINT, GL\_NICEST);

// Дозволяємо змішування

glEnable (GL\_BLEND);

// Вказуємо спосіб змішування

glBlendFunc (GL\_SRC\_ALPHA, GL\_ONE\_MINUS\_SRC\_ALPHA);

}

void My\_Paint::***resizeGL***(int nWidth, int nHeight) {

glViewport(0, 0, nHeight, nWidth); // установка точки обзора

glMatrixMode(GL\_PROJECTION); // установка режима матрицы

glLoadIdentity(); // загрузка матрицы

}

void My\_Paint::***paintGL***() {

*resizeGL*(800, 1000);

glClear(GL\_COLOR\_BUFFER\_BIT | GL\_DEPTH\_BUFFER\_BIT); // очистка экрана

glMatrixMode(GL\_MODELVIEW); // задаем модельно-видовую матрицу

glLoadIdentity(); // загрузка единичную матрицу

glScalef(1, 1, 1); // масштабирование

glTranslatef(-0.5, -0.5, 0.0); // смещение

glRotatef(xRotation, 1.0f, 0.0f, 0.0f); // поворот по X

glRotatef(yRotation, 0.0f, 1.0f, 0.0f); // поворот по Y

glRotatef(zRotation, 0.0f, 0.0f, 1.0f); // поворот по Z

scene();

}

void My\_Paint::**scene**() {

// створюємо букву V квадратними полігонами

glBegin(GL\_QUADS);

qglColor(QColor(247, 104, 151));

// V – передня частина

drawFilledSquare(vertices\_V, 0, 1, 2, 6, z1, z1);

drawFilledSquare(vertices\_V, 3, 4, 5, 6, z1, z1);

// V – задня частина

drawFilledSquare(vertices\_V, 0, 1, 2, 6, z2, z2);

drawFilledSquare(vertices\_V, 3, 4, 5, 6, z2, z2);

// V - бокові частини

drawFilledSquare(vertices\_V, 0, 1, 1, 0, z1, z2);

drawFilledSquare(vertices\_V, 1, 2, 2, 1, z1, z2);

drawFilledSquare(vertices\_V, 2, 3, 3, 2, z1, z2);

drawFilledSquare(vertices\_V, 3, 4, 4, 3, z1, z2);

drawFilledSquare(vertices\_V, 4, 5, 5, 4, z1, z2);

drawFilledSquare(vertices\_V, 5, 6, 6, 5, z1, z2);

drawFilledSquare(vertices\_V, 6, 0, 0, 6, z1, z2);

glEnd();

// координати меж літери V

glLineWidth(2.0);

qglColor(QColor(180, 8, 65));

DrawLineStrip(vertices\_V, 7, z1);

DrawLineStrip(vertices\_V, 7, z2);

DrawLineZ(vertices\_V, 7);

// створюємо букву K квадратними полігонами

glBegin(GL\_QUADS);

qglColor(QColor(247, 104, 151));

// K – передня частина

drawFilledSquare(vertices\_K, 0, 1, 2, 12, z1, z1);

drawFilledSquare(vertices\_K, 3, 4, 10, 11, z1, z1);

drawFilledSquare(vertices\_K, 4, 5, 6, 7, z1, z1);

drawFilledSquare(vertices\_K, 7, 8, 9, 10, z1, z1);

drawFilledSquare(vertices\_K, 4, 7, 10, 4, z1, z1);

// K – задня частина

drawFilledSquare(vertices\_K, 0, 1, 2, 12, z2, z2);

drawFilledSquare(vertices\_K, 3, 4, 10, 11, z2, z2);

drawFilledSquare(vertices\_K, 4, 5, 6, 7, z2, z2);

drawFilledSquare(vertices\_K, 7, 8, 9, 10, z2, z2);

drawFilledSquare(vertices\_K, 4, 7, 10, 4, z2, z2);

// K - бокові частини

for(int i = 0; i < 12; i++) {

drawFilledSquare(vertices\_K, i, i+1, i+1, i, z1, z2);

}

drawFilledSquare(vertices\_K, 12, 0, 0, 12, z1, z2);

glEnd();

// координати меж літери K

glLineWidth(2.0);

qglColor(QColor(180, 8, 65));

DrawLineStrip(vertices\_K, 13, z1);

DrawLineStrip(vertices\_K, 13, z2);

DrawLineZ(vertices\_K, 13);

}

void My\_Paint::***mousePressEvent***(QMouseEvent\* event) {

// запоминаем координату нажатия мыши

mousePos = event->pos();

}

void My\_Paint::***mouseMoveEvent***(QMouseEvent\* event) {

xRotation += 180/scale\*(GLfloat)(event->y()-mousePos.y())/height(); // вычисляем углы поворота

zRotation += 180/scale\*(GLfloat)(event->x()-mousePos.x())/width();

mousePos = event->pos();

*updateGL*(); // обновляем изображение

}

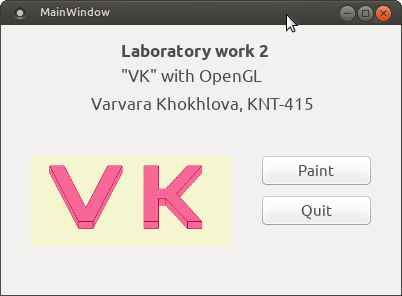


Рисунок 3.1 – Головне вікно програми



Рисунок 3.2 – Малюнок з використанням OpenGL

1. Висновки

Виконуючи лабораторну роботу, я навчилася працювати з тривимірними фігурами та їх положенням, намалювала літери “V” та “K”.