**Міністерство освіти і науки України**

**Запорізький національний технічний університет**

кафедра програмних засобів

**ЗВІТ**

з лабораторної роботи №1

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

"Створення простого зображення за допомогою бібліотеки Open GL

у середовищі розробки Qt"

Варіант 22

Виконала:

студентка групи КНТ-415 В.С. Хохлова

Прийняв:

асистент Ж.К. Камінська

2018

1. Мета роботи

Ознайомитися з бібліотекою OpenGL у середовищі розробки Qt та створити просте зображення.

1. Завдання

Використовуючи примітиви «точка», «лінія», «трикутник», «чотирикутник» та «многокутник» зобразити фігуру, вказану у варіанті.

1. Хід роботи

#include "my\_paint.h"

#include <QtOpenGL>

#include <GL/gl.h>

My\_Paint::My\_Paint(){}

void My\_Paint::initializeGL() {

//Обрати фоновий (очищуючий) колір

glClearColor(0.0, 0.0, 0.0, 0.0);

//glClearColor(r,g,b,t); Де, r-червоний колір, g-зелений, b-голубий, t-прозрачність.

//Задаємо режим обробки полігонів - передню та задню частини,

//полігони повністю зафарбовані

//(можна просто відображувати обрамлення)

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

}

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

//Встановлюємо точку огляду. Останні два параметра однакові -

// щоб не порушувати пропорції у широких экранів

glViewport(0, 0, nWidth, nHeight);

glMatrixMode(GL\_PROJECTION); //Встановлюємо режим матриці

glLoadIdentity(); //Завантажуємо матрицю

}

void My\_Paint::paintGL() {

this->resize(600, 400);

glClear(GL\_COLOR\_BUFFER\_BIT | GL\_DEPTH\_BUFFER\_BIT); //Очищуємо екран

glViewport(0, 0, 600, 400); //Встановити проекцію:

glMatrixMode(GL\_PROJECTION); //Задаємо режим матриці

glLoadIdentity(); //Завантажуємо матрицю

glOrtho(0.0,600.0,400.0,0.0,-1.0,1.0); // L, R, B. T, Near, Far

glMatrixMode(GL\_MODELVIEW); // return to the model matrix

glLoadIdentity();

scene();

swapBuffers(); //выводимо на екран

}

void drawFilledCircle(GLfloat x, GLfloat y, GLfloat radius){

int i;

int triangleAmount = 20; //# of triangles used to draw circle

GLfloat twicePi = 2.0f \* 3.14f;

glBegin(GL\_TRIANGLE\_FAN);

glVertex2f(x, y); // center of circle

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

glVertex2f(

x + (radius \* cos(i \* twicePi / triangleAmount)),

y + (radius \* sin(i \* twicePi / triangleAmount)));

}

glEnd();

}

void drawFilledRectange(GLfloat x1, GLfloat y1, GLfloat x2, GLfloat y2, GLfloat z=0.0) {

//Починаємо відрисовку, аргумент означає відрисовку прямокутника.

//Кожний виклик glVertex3f задає одну вершину прямокутника

glBegin(GL\_POLYGON);

glVertex3f(x1, y1, z);

glVertex3f(x2, y1, z);

glVertex3f(x2, y2, z);

glVertex3f(x1, y2, z);

glEnd();

}

void My\_Paint::scene() {

// Малюємо небо

//Задаємо колір зображення

qglColor(QColor(0x39, 0x2d, 0xe1));

drawFilledRectange(0.0, 0.0, 600.0, 80.0);

//Зіроньки

qglColor(QColor(0xf0, 0xf3, 0x0f));

drawFilledRectange(50.0, 50.0, 70.0, 70.0);

// Основа будиночка

qglColor(QColor(0xd4, 0x30, 0x89));

drawFilledRectange(110.0, 200.0, 490.0, 360.0);

// Основне вікно та двері

qglColor(QColor(0xe3, 0xd3, 0x1b));

drawFilledRectange(150.0, 240.0, 190.0, 280.0);

drawFilledRectange(220.0, 270.0, 280.0, 360.0);

// Вікно у душовій

qglColor(QColor(0xe0, 0x9b, 0x15));

drawFilledRectange(360.0, 230.0, 450.0, 270.0);

// Дах

qglColor(QColor(0x7e, 0x61, 0x07));

glBegin(GL\_POLYGON);

glVertex3f(70.0,200.0,0.0);

glVertex3f(190.0,60.0,0.0);

glVertex3f(550.0,200.0,0.0);

glEnd();

// Гірлянда на даху

qglColor(QColor(0xa6, 0x13, 0x8f));

drawFilledCircle(135, 155, 15);

}

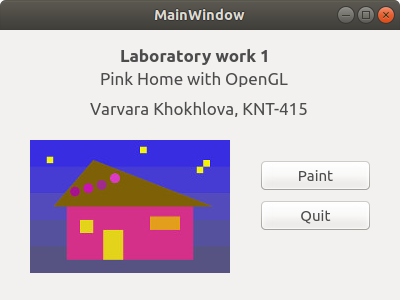


Рисунок 3.1 – Головне вікно програми

Рисунок 3.2 – Малюнок з використанням OpenGL

1. Висновки

Виконуючи лабораторну роботу, я ознайомилась з бібліотекою OpenGL у середовищі розробки Qt та створила просте зображення рожевого будинку вночі.