Міністерство освіти і науки України

Державний університет ,,Житомирська політехніка”

Кафедра КН

Група: КН-22-3

Комп'ютерна графіка та конструювання графічних інтерфейсів

Лабораторна робота №1

«OpenGL»

Виконав: Ткачук Н. О.

Прийняв: Власенко О. В.

**Мета роботи:** ознайомитися з основами роботи з бібліотекою OpenGl. Підготувати усе необхідне для майбутнього проекту.

Виконання роботи:

1. Що таке OpenGL?

OpenGL (Open Graphics Library) - це крос-платформена бібліотека для реалізації графіки та 3D-зображень на комп'ютерах. Вона надає програмістам інтерфейс для створення візуалізаційних програм, які можуть працювати на різних операційних системах, включаючи Windows, macOS і Linux.

OpenGL широко використовується у галузях від ігор до наукової візуалізації, комп'ютерної графіки, медицинської візуалізації, симуляції та інших областях, де потрібна обробка та відображення графіки. Багато популярних графічних бібліотек та ігрових движків, такі як Unity і Unreal Engine, базуються на OpenGL або його відкритому відповіднику, яким є OpenGL ES для мобільних платформ.

2. Підготовка до першого проекту OpenGL: налаштування GLFW, та GLAD

Для підключення GLFW та GLAD в середовищі Visual Studio для роботи з OpenGL було проробленно наступні кроки:

1. Створення проекту в Visual Studio:

- Відкрийте Visual Studio та створіть новий проект C++ (наприклад, "Empty Project").

- Додайте файли програми Python у вашому проекті.

2. Встановлення GLFW:

- Відвідайте офіційний веб-сайт GLFW і завантажте файл для Windows.

- Розпакуйте завантажений архів і скопіюйте файли `glfw3.lib` та `glfw3.dll` до вашого проекту.

- В Visual Studio перейдіть до властивостей проекту. Для цього клацніть правою кнопкою миші на проекті в Solution Explorer та виберіть "Properties".

3. Налаштування залежностей і додавання GLFW:

- У властивостях проекту перейдіть до розділу "VC++ Directories".

- В розділі "Include Directories" додайте шлях до папки `include` GLFW.

- У розділі "Library Directories" додайте шлях до папки `lib` GLFW.

- У розділі "Input" в "Additional Dependencies" додайте `glfw3.lib`.

4. Встановлення GLAD:

- Відвідайте веб-сайт GLAD і налаштуйте параметри GLAD так, як ви хочете.

- Натисніть кнопку "Generate" та завантажте zip-архів із згенерованим кодом.

- Розпакуйте завантажений архів та додайте файли `glad.c` та `glad.h` до вашого проекту.

5. Додаємо файли GLAD у Visual Studio:

- В Visual Studio, правою кнопкою миші клацніть на вашому проекті в Solution Explorer та виберіть "Add" > "Existing Item".

- Виберіть файли `glad.c` та `glad.h`, які ви додали до вашого проекту.

6. Включаємо файл glad.h та glfw3.h:

- В Visual Studio включіть файл `glad.h`, додавши наступний код в верхній частині вашого коду С:

#include <glad/glad.h>

#include <GLFW/glfw3.h>

3. Перший проект у OpenGL — Створення вікна

Лістинг програми:

#include <glad/glad.h>

#include <GLFW/glfw3.h>

#include <iostream>

void framebuffer\_size\_callback(GLFWwindow\* window, int width, int height);

void processInput(GLFWwindow\* window);

//Константи

const unsigned int SCR\_WIDTH = 800;

const unsigned int SCR\_HEIGHT = 600;

int main()

{

// glfw: ініціалізація та конфігурування

glfwInit();

glfwWindowHint(GLFW\_CONTEXT\_VERSION\_MAJOR, 3);

glfwWindowHint(GLFW\_CONTEXT\_VERSION\_MINOR, 3);

glfwWindowHint(GLFW\_OPENGL\_PROFILE, GLFW\_OPENGL\_CORE\_PROFILE);

// glfw: створення вікна

GLFWwindow\* window = glfwCreateWindow(SCR\_WIDTH, SCR\_HEIGHT, "OpenGLдля тесту", NULL, NULL);

if (window == NULL)

{

std::cout << "Не вдалося створити вікно GLFW" << std::endl;

glfwTerminate();

return -1;

}

glfwMakeContextCurrent(window);

glfwSetFramebufferSizeCallback(window, framebuffer\_size\_callback);

// glad: завантаження всіх покажчиків на OpenGL-функції

if (!gladLoadGLLoader((GLADloadproc)glfwGetProcAddress))

{

std::cout << "Не вдалося ініціалізувати GLAD" << std::endl;

return -1;

}

// Цикл візуалізації

while (!glfwWindowShouldClose(window))

{

// Обробка введення

processInput(window);

// Виконання рендерингу

glClearColor(0.76f, 1.0f, 0.407f, 1.0f);

glClear(GL\_COLOR\_BUFFER\_BIT);

// glfw: обмін вмістом front-і back-буферів. Відстеження подій введення/виводу (чи

//була натиснута / відпущена кнопка, переміщений курсор миші тощо)

glfwSwapBuffers(window);

glfwPollEvents();

}

// glfw: завершення, звільнення всіх раніше задіяних GLFW-ресурсів

glfwTerminate();

return 0;

}

// Обробка всіх подій введення: запит GLFW про натискання/відпускання клавіш на

//клавіатурі в даному кадрі та відповідна обробка даних подій

void processInput(GLFWwindow\* window)

{

if (glfwGetKey(window, GLFW\_KEY\_ESCAPE) == GLFW\_PRESS)

glfwSetWindowShouldClose(window, true);

}

// glfw: щоразу, коли змінюються розміри вікна (користувачем або операційною

//системою), викликається дана callback - функція

void framebuffer\_size\_callback(GLFWwindow \* window, int width, int height)

{

// Переконуємось, що вікно перегляду відповідає новим розмірам вікна.

// Зверніть увагу, висота вікна на Retina-дисплеях буде значно більшою, ніж

//зазначено в програмі

glViewport(0, 0, width, height);

}

Результат виконання програми:



**Висновок:** на даній лабораторній роботі було ознайомленно з базовими функціями OpenGL , що становлять основу для розробки графічних програм з використанням цієї бібліотеки. Ми навчилися створювати вікна, ініціалізувати контекст OpenGL та організовувати основний цикл малювання. Це є першим кроком у розвитку навичок роботи з 3D-графікою та візуалізацією.