



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ _____ «Информатика и системы управления»

КАФЕДРА _____ «Теоретическая информатика и компьютерные технологии»

Лабораторная работа № 1 по курсу «Компьютерная Графика»

Студент группы ИУ9-42Б Павлов И. П.

Преподаватель Цалкович П.А.

Москва 2024

1 Задача

- Реализовать любой графический примитив
- Добавить любое геометрическое преобразование
- Добавить обработку события

2 Основная теория

Для инициализации библиотеки используется функция `glfwInit()`. Для создания окна используется `glfwCreateWindow(...)`, в аргументах которой указываются высота и ширина окна. Для считывания нажатий клавиш нужно установить callback хэндлер, передав в `glfwSetKeyCallback(...)` void-функцию, которая кастится к типу `GLFWKeyFun`. Далее нужно в цикле выводить на экран полигон, который нужно создать по вершинам с помощью `glVertex2f(...)` (изображение двумерное). При закрытии программы нужно вызвать `glfwDestroyWindow(...)` и `glfwTerminate()`.

3 Практическая реализация

3.1 Реализация main.cpp

```
#include "main.h"

GLfloat angle = 30.0;

int main() {
    if (!glfwInit()) {
        return 1;
    }

    GLFWwindow* window = glfwCreateWindow(900, 900, "Lab1",
                                           nullptr, nullptr);

    if (window == nullptr) {
        glfwTerminate();
        return 1;
    }

    glfwMakeContextCurrent(window);
    glfwSetKeyCallback(window, key_callback);

    while(!glfwWindowShouldClose(window)) {
```

```

        display(window);
    }

    glfwDestroyWindow(window);
    glfwTerminate();

    return 0;
}

void key_callback(GLFWwindow* window, int key, int scancode, int action, int mods) {
    if (action == GLFW_PRESS) {
        if (key == GLFW_KEY_RIGHT) {
            angle -= 15;
        }
        if (key == GLFW_KEY_LEFT) {
            angle += 15;
        }
        if (key == 'Q') {
            glfwTerminate();
            exit(0);
        }
    }
}

void display(GLFWwindow* window) {
    glClear(GL_COLOR_BUFFER_BIT);
    glLoadIdentity();
    glClearColor(0.0, 0.0, 0.0, 1.0);
    glPushMatrix();
    glRotatef(angle, 0, 0, 1);

    glBegin(GL_POLYGON);
    glColor3f(1.0, 0.0, 1.0);
    glVertex2f(-0.5f, -0.5f);

    glVertex2f(-0.5f, 0.3f);
    glVertex2f(-0.3f, 0.5f);

    glVertex2f(0.2f, 0.5f);
    glVertex2f(0.5f, 0.2f);

    glVertex2f(0.5f, -0.3f);
    glVertex2f(0.3f, -0.5f);
    glEnd();

    glPopMatrix();
    glfwSwapBuffers(window);
    glfwPollEvents();
}

```

3.2 Реализация main.h

```

#ifndef PROGRAM_MAIN_H
#define PROGRAM_MAIN_H

#include <stdlib.h>
#include "GLFW/glfw3.h"

```

```
int main();  
void key_callback(GLFWwindow*, int, int, int, int);  
void display(GLFWwindow*);  
  
#endif //PROGRAM_MAIN_H
```

4 Заключение

Для выполнения лабораторной работы не потребовалось дополнительно устанавливать никакие библиотеки, так как на маняго все необходимое для работы с OpenGL установлено по умолчанию. Во время выполнения лабораторной работы был получен навык работы с графическими примитивами OpenGL, а также отображения их при помощи оконной программы.