МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра математического обеспечения и применения ЭВМ

ОТЧЕТ

по лабораторной работе №5

по дисциплине «Компьютерная графика»

Тема: Расширения OpenGL, программируемый графический конвейер. Шейдеры.

Студент гр.8382 _____ Нечепуренко Н.А.

Студент гр.8382 _____ Терехов А.Е.

Преподаватель _____ Герасимова Т.В.

Санкт-Петербург

2021

Цели работы.

С помощью программируемого графического конвейера выполнить преобразования над исходным изображением. Изучить язык шейдеров GLSL, описать вершинный и фрагментный шейдеры.

Задание.

Варианты 5_1_20:

Задание: 20. С помощью фрагментного шейдера сделать цветное изображение чёрно-белым. С помощью вершинного добавить волны на поверхность объекта. Требуется модифицировать координаты вершин и нормалей.

Выполнение работы.

Для отображения изображения и последующего применения эффекта волны необходимо сгенерировать сетку из вершин, к которой будет привязана текстура. Изменяя характеристики вершин сетки в вершинном шейдере, добьемся эффекта волны. Код генерации вершин приведен ниже.

```
float **generateVertices(int size) {
    float delta = 2.0 / (size - 2);
    float x_pos = 0.5;
    float y_pos = -0.5;
    float z_pos = 0.5;
    float x_tex = 1.0;
    float y_tex = 1.0;
    float **arr = new float *[size];
    for (int i = 0; i < size; ++i) {
        arr[i] = new float[5];
        arr[i][0] = x_pos;
        arr[i][1] = y_pos;
```

```
            arr[i][2] = 0;
            arr[i][3] = x_tex;
            arr[i][4] = y_tex;
            x_pos -= delta * (i % 2);
            y_pos = -y_pos;
            z_pos = -z_pos;
            x_tex -= delta * (i % 2);
            y_tex = std::abs(y_tex - 1);
        }
        return arr;
    }
```

В вершинный шейдер передаем координаты узла сетки и соответствующие ему текстурные координаты. Также передадим три матрицы: projection, view, model – с помощью которых будем вычислять новые координаты вершины.

```
glm::mat4 model = glm::mat4(1.0f);
glm::mat4 view = glm::mat4(1.0f);
glm::mat4 projection = glm::mat4(1.0f);
model = glm::rotate(model, glm::radians(-55.0f), glm::vec3
   (1.0f, 0.0f, 0.0f));
view = glm::translate(view, glm::vec3(0.0f, 0.0f, -3.0f));
projection = glm::perspective(glm::radians(45.0f), (float)
   SCR_WIDTH / (float)SCR_HEIGHT, 0.1f, 100.0f);
```

Код вершинного шейдера приведен ниже.

```
#version 330 core
layout (location = 0) in vec3 aPos;
layout (location = 1) in vec2 aTexCoord;

out vec2 TexCoord;
uniform mat4 model;
```

```
uniform mat4 view;
uniform mat4 projection;

void main()
{
    float pi = 3.1415;
    vec3 bPos = vec3(aPos.x, sin(4*pi*aPos.x)*0.2+aPos.y,
       aPos.z);
    gl_Position = projection * view * model * vec4(bPos, 1.0)
       ;
    TexCoord = vec2(aTexCoord.x, aTexCoord.y);
}
```

Из вершинного шейдера передаем во фрагментный текстурные координаты.

В фрагментном шейдере необходимо изменить цвет каждого пикселя изображения так, чтобы оно стало черно-белым. Серый цвет и его оттенки характеризуются тем, что значение всех каналов, кроме альфа-канала, одинаково. Исходя из этого, достаточно для каждого пикселя в каждый его цветовой канал положить среднее арифметическое его каналов. Код фрагментного шейдера приведен ниже.

```
#version 330 core
out vec4 FragColor;

in vec3 ourColor;
in vec2 TexCoord;

uniform sampler2D texture1;

void main()
{
    vec4 tex = texture(texture1, TexCoord);
```

```
    float grey = (tex.x + tex.y + tex.z) / 3.0;
    FragColor = vec4(grey,grey,grey,1);
}
```

Результат применения шейдера к тестовому изображению приведен на рисунке 1.

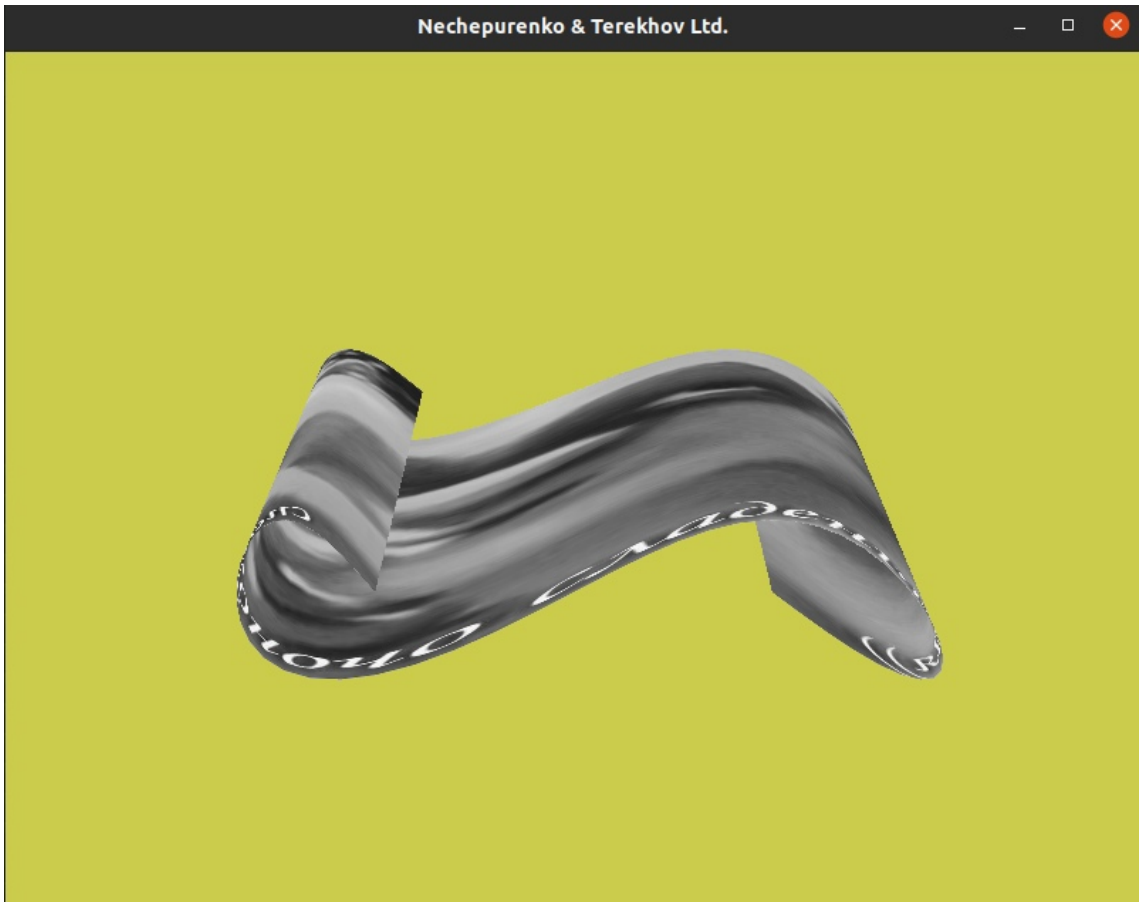

Рисунок 1 – Результат применения шейдеров к изображению

Выводы.

В результате выполнения лабораторной работы была реализована программа, позволяющая добавлять волны на поверхность изображения и делать его черно-белым. Это было реализовано путем программирования графического конвейера OpenGL с помощью вершинного и фрагментного шейдеров, код которых приведен выше.