

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра МО ЭВМ**

**ОТЧЕТ**  
**по лабораторной работе №5**  
**по дисциплине «Компьютерная графика»**  
**Тема: Расширения OpenGL, программируемый**  
**графический конвейер. Шейдеры**

Студентка гр. 0382

Здобнова К.Д.

Преподаватель

Герасимова Т.В.

Санкт-Петербург

2023

## ЗАДАНИЕ

Вариант 27.

Эффект линзы / рыбий глаз (Magnify Effect).



## ХОД РАБОТЫ

В ходе выполнения лабораторной работы был написан фрагментный шейдер:

```
uniform sampler2D texture;
uniform vec2 resolution;
uniform vec2 mouse;

void main() {
    vec2 uv = gl_FragCoord.xy / resolution.xy;
    vec2 center = mouse / resolution.xy;
    float radius = 0.1;
    float strength = 0.5;
    vec4 color = texture2D(texture, uv);
    float distance = length(uv - center);
    if (distance < radius) {
        float delta = smoothstep(radius, radius * (1.0 -
strength), distance);
        gl_FragColor = mix(color, texture2D(texture, center + (uv
- center) * (1.0 + delta)), delta);
    } else {
```

```

        gl_FragColor = color;
    }
}

```

Этот шейдер принимает текстуру и два юниформа: *resolution* и *mouse*. *Resolution* — это вектор, содержащий ширину и высоту экрана в пикселях. *Mouse* — это вектор, содержащий координаты x и y положения, где должен располагаться эффект линзы.

Шейдер вычисляет UV-координаты каждого пикселя изображения, используя *gl\_FragCoord.xy / resolution.xy*. Затем он вычисляет расстояние между каждым пикселем и положением круга, используя *length(uv - center)*.

Если расстояние меньше определенного радиуса, он применяет эффект увеличения к этому пикселю, смешивая его цвет с цветом близлежащего пикселя, используя *mix(color, texture2D(texture, center + (uv - center) \* (1.0 + delta)), delta)*. Величина увеличения контролируется двумя переменными: *radius* и *strength*, значения которых задаются в самом шейдере.

Если расстояние больше радиуса, шейдер просто выводит исходный цвет пикселя.

Чтобы использовать этот шейдер в программе нужно установить юниформы *resolution* и *mouse* перед рендерингом изображения:

```

int resolutionLocation = glGetUniformLocation(shaderProgram,
"resolution");
glUniform2f(resolutionLocation, 1000, 1000);

```

В данном фрагменте кода устанавливаются значения юниформы *resolution* – ширина 1000, высота 1000.

```

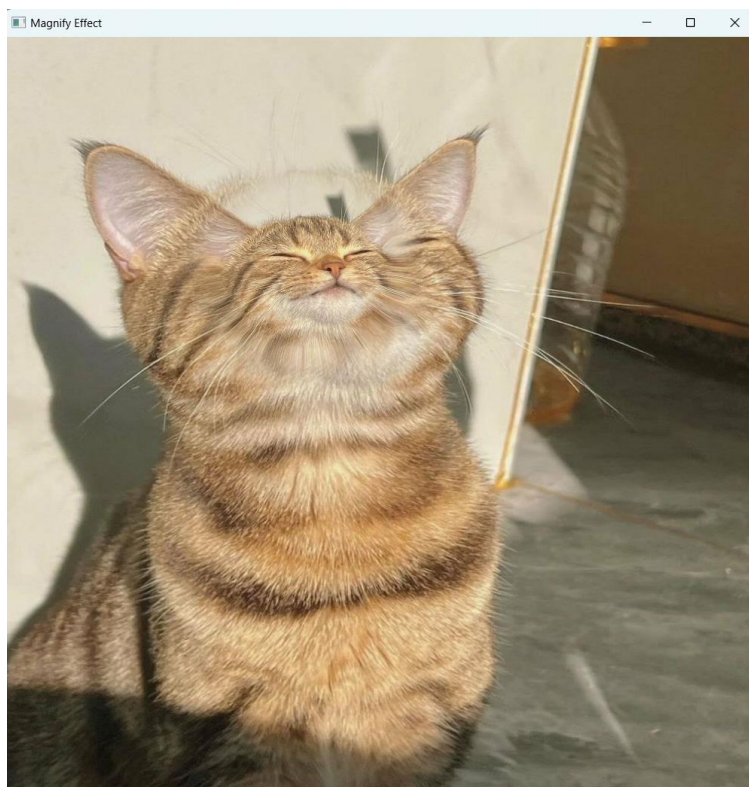
int mouseLocation = glGetUniformLocation(shaderProgram, "mouse");
glUniform2f(mouseLocation, 400f, 700f);

```

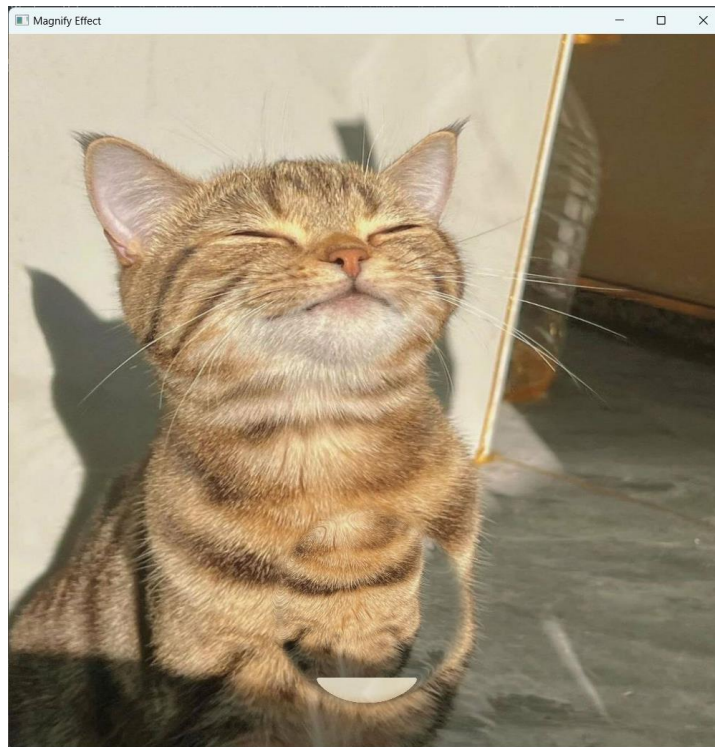
Во втором фрагменте – значения юниформы *mouse*. Таким образом, в изображении 1000x1000 круг с эффектом рыбьего глаза будет располагаться с центром в точке (400; 700) и радиусом 0.1 (это значение задается в шейдере).

Для загрузки изображения используется библиотека *ByteBuffer* – изображение попиксельно загружается в буфер, затем с помощью функции *glTexImage2D* происходит текстурирование, которое позволяет шейдерам читать элементы массива изображений.

## ТЕСТИРОВАНИЕ



Эффект рыбьего глаза с параметрами *resolution* (1000, 1000), *mouse* (400, 700), *radius* = 0.19, *strength* = 0.5.



Эффект рыбьего глаза с параметрами *resolution* (1000, 1000), *mouse* (500, 200), *radius* = 0.15, *strength* = 0.2.

## ВЫВОД

В результате выполнения лабораторной работы была разработана программа, применяя на изображении эффект рыбьего глаза. Программа работает корректно. При выполнении работы были приобретены навыки работы с графической библиотекой OpenGL и шейдерами.