

## Создание треугольной мозаики на языке шейдера

*Черкашин Александр Михайлович*

*Приамурский государственный университет имени Шолом-Алейхема*

*Студент*

### **Аннотация**

В данной статье описан процесс создания треугольной мозаики. В процессе работы использовался язык шейдера GLSL.

**Ключевые слова:** шейдер, OpenGL Shading Language, GLSL, текстура, мозаика.

## Creating a triangular mosaic in the shader language

*Cherkashin Alexander Mihailovich*

*Sholom-Aleichem Priamursky State University*

*Student*

### **Abstract**

This article describes the process of creating a triangular mosaic. In the process of work, the GLSL shader language was used. As a result, the image was displayed in the form of a triangular mosaic.

**Keywords:** shader, OpenGL Shading Language, GLSL, texture, mosaic.

### **1 Введение**

#### *1.1 Актуальность исследования*

Данная статья описывает возможность использования языка шейдера для создания мозаики.

#### *1.2 Цель исследования*

Целью работы является написание программы на языке шейдера для получения изображения в виде мозаики.

#### *1.3 Обзор исследований*

Д. Босиков описывает пакет программ для визуализации мозаики 3D с использованием OpenGL (GLSL) в реальном времени [1]. В работе Е. Мазза, де М. Й. М. Писон представляют мультиплатформенную программу с открытым исходным кодом, предназначенную для программирования в реальном времени и визуального программирования. Приложение на базе openFrameworks [2]. А. Дения показывает использование языка шейдера для мозаики больших размеров флага, для визуализация анимационных эффектов для спортивных трансляций [3].

## 2. Рабочий процесс

В данной статье использовалось изображение (рис. 2.1) для демонстрации программы шейдера. В работе использовалась программа glslViewer [4].

Данная программа написана на языке OpenGL Shading Language (GLSL).



Рисунок 2.1. Исходное изображение

Листинг 2.1. Исходный код программы для создания эффекта капли.

```
1 #version 330
2 #ifdef GL_ES
3 precision mediump float;
4 #endif
5
6 uniform vec2 u_resolution;
7 uniform float u_time;
8 uniform vec2 u_mouse;
9
10 uniform sampler2D u_tex0;
11 uniform vec2 u_tex0Resolution;
12
13 void main (void) {
14     vec2 uv = gl_FragCoord.xy / u_resolution.xy;
15     vec2 muv = u_mouse.xy / u_resolution.xy;
16     vec2 tile_num = u_resolution.xy * muv / 4.0;
```

```

17     vec2 uv2 = floor(uv * tile_num) / tile_num;
18     uv -= uv2;
19     uv *= tile_num;
20     gl_FragColor = texture(u_tex0,
21         uv2 + vec2(step(1.0 - uv.y, uv.x) / (2.0 * tile_num.x),
22         step(uv.x, uv.y) / (2.0 * tile_num.y)
23     ));
24 }

```

В строке 1 (листинг 2.1) выбрана версия шейдера 3.30.

В строках 6 — 11:

u\_resolution - разрешение экрана,

u\_time — время в секундах,

u\_tex0 — текстура (рис 2.1) и u\_tex0Resolution — размер текстуры (рис 2.2).

u\_mouse — координаты курсора, система координат от 0 до размера экрана.

В строках 14 — 17:

Переменная uv — экранное пространство от 0 до 1. muv — пространство курсора от 0 до 1, tile\_num — размер решетки, зависит от координат курсора, uv2 получаем размер квадратного пикселя.

В строках 18 — 23: текстурные координаты для получения треугольника и изображения в виде мозаики треугольников.

Команда для запуска программы:

```
glslViewer main.frag Рис2.1 -w 800 -h 600
```

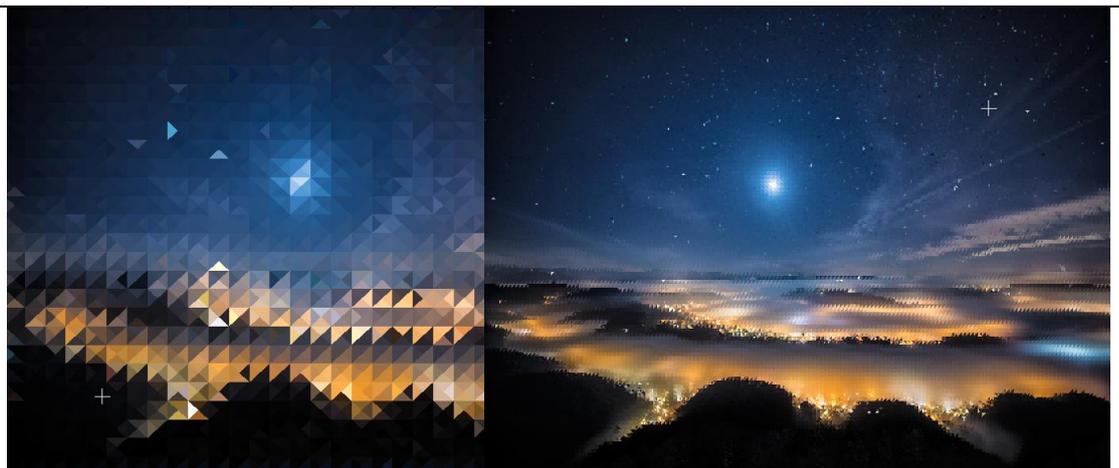


Рисунок 2.2. Результаты работы программы, в зависимости расположения курсора

### 3 Выводы

В результате работы написана программа на языке шейдера для получения изображения в виде мозаики.

**Библиографический список**

1. Bosikov D. Developing a Comprehensive Software Toolkit for Creating Digital Mosaic Artwork. Thesis for the degree Master of Computer Science. City University of New York, 2012.
2. Mazza E., de Pisón M. J. M. Mosaic, an openFrameworks based Visual Patching Creative-Coding Platform //Mosaic. Accedido. 2020. Т. 15. URL: <https://iclc.toplap.org/2019/papers/paper50.pdf>
3. Denia A. et al. Low Cost Virtual Animation Effects for Sports Broadcasting: Mosaics, Flags and Big-Sized Flags // 2011 International Conference on Computational Science and Its Applications. IEEE, 2011. С. 10-16.
4. patriciogonzalezvivo/gslViewer: Console-based GLSL Sandbox for 2D/3D shaders [shaders. // GitHub](https://github.com/patriciogonzalezvivo/gslViewer) URL: <https://github.com/patriciogonzalezvivo/gslViewer> (дата обращения: 2022-09-04).