

Компьютерная графика

Лабораторная работа №1.

Построение изображения флага с помощью пиксельного шейдера

Задание

Изучите исходный код программы, написанный на языке GLSL, и создайте изображение флага согласно вашему варианту задания с помощью пиксельного (фрагментного) шейдера.

Ход выполнения работы

Шаг 1. Настройка окружения.

Создайте и сохраните файл с исходным кодом программы, представленный в конце этого документа, в отдельной папке.

Откройте сохраненный файл с помощью текстового редактора Atom, поддерживающий множество различных плагинов.

Для работы с пиксельным шейдером установите на Atom плагины language-gls, glsl-preview и autocomplete-gls. (Эти плагины можно установить с помощью консоли, написав команду: `apm install language-gls glsl-preview autocomplete-gls`).

Для предпросмотра результата работы пиксельного шейдера нажмите сочетание клавиш Ctrl+Shift+G.

Шаг 2. Изучение языка GLSL.

Изучите указанную литературу и сделайте краткий конспект изученного материала, как минимум содержащий развернутые ответы на следующие контрольные вопросы:

1. Что такое шейдеры? Какие виды шейдеров вы знаете? Каково их назначение?
2. Как расшифровывается аббревиатура GLSL?
3. На чем основан язык GLSL, в чем заключается его особенность?
4. Какие типы встроенных переменных в язык GLSL имеются? В чем их преимущество по сравнению со встроенными переменными языка Си?
5. Что такое uniform-переменные? Для чего они нужны?
6. Для чего нужны переменные `gl_FragCoord` и `gl_FragColor`?
7. Существуют ли встроенные функции? Если да, то приведите примеры таких функций.

Список основной литературы:

- 1) Приложение. Язык GLSL. // Боресков А.В. Программирование компьютерной графики. Современный OpenGL. - М.: ДМК Пресс, 2019. - 372 с.
- 2) Главы: Введение и Алгоритмическое рисование. // The Book of Shaders (авторы: Патрицио Гонзалес Виво и Джен Лав) // <https://thebookofshaders.com/?lan=ru>

Список дополнительной литературы:

3) <https://habr.com/ru/post/313380/>

4) <https://webglfundamentals.org/webgl/lessons/ru/webgl-shaders-and-gsl.html>

5) <https://www.khronos.org/registry/OpenGL/specs/gl/>









Шаг 3. Построение изображение флага страны согласно варианту.

Выберите флаг страны, который необходимо вам нарисовать, согласно вашему варианту из представленной ниже таблицы. Помните, что необходимо корректно определить цвета, соотношение кривых на флаге и размеры флага.

Используя, полученные знания из предыдущих шагов, модифицируйте код пиксельного шейдера так, чтобы добиться желаемого результата.

Table 1: Таблица с вариантами заданий

| | | | |
|--|---|---|--|
| 1. Флаг страны Грузия  | 2. Флаг страны Гайана  | 3. Флаг страны Лаос  | 4. Флаг страны Сан-Томе и Принсипи  |
| 5. Флаг страны Сент-Люсия  | 6. Флаг страны Тонга  | 7. Флаг страны Великобритания  | 8. Флаг страны Сент-Винсент и Гренадина  |
| 9. Флаг страны Гренландия  | 10. Флаг страны Исландия  | 11. Флаг страны Танзания  | 12. Флаг страны Израиль  |

| | | | |
|--|--|--|---|
| <p>13. Флаг страны Македония</p>  | <p>14. Флаг страны Багамские острова</p>  | <p>15. Флаг страны Сейшельские остр-а</p>  | <p>16. Флаг страны Восточный Тимор</p>  |
| <p>17. Флаг страны Ямайка</p>  | <p>18. Флаг страны ЮАР</p>  | <p>19. Флаг страны Катар</p>  | <p>20. Флаг страны Республика Корея</p>  |

Текст файла "cg_p1_flags.glsl" с исходным кодом программы

```
//Use ctrl+shift+G for live preview a fragment shaders
```

```
uniform vec2 u_resolution;
```

```
vec4 RussianFlag(vec2 position)
```

```
{
    vec4 color = vec4(0);
    if (position.y <= 0.22)
        color = vec4(1,vec2(0),1);
    else if (position.y <= 0.44)
        color = vec4(vec2(0),vec2(1));
    else if (position.y <= 0.66)
        color = vec4(1);
    return color;
}
```

```
void main()
```

```
{
    vec2 position = gl_FragCoord.xy/u_resolution.xy;
    gl_FragColor = RussianFlag(position);
}
```
