# МИНОБРНАУКИ РОССИИ

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**

**Кафедра МО ЭВМ**

**ОТЧЕТ**

**по лабораторной работе №1**

**по дисциплине «Компьютерная графика»**

**Тема: «Рисование геометрических объектов»**

|  |  |
| --- | --- |
| Студент гр. 6303 | Фиалковский М.С. |
| Преподаватель | Герасимова Т.В. |

## Санкт-Петербург 2020

**Задание.**

Реализовать программу с рисованием геометрической фигуры с использованием возможностей браузера и технологии современного OpenGL.

**Общие сведения.**

Под современным подходом, указанным выше, имеется ввиду разработка и использование спецификации некоторых этапов графического конвейера, т.е. шейдеров.

Помимо шейдеров в программе используются возможности преобразования С++ программ в WEB-контент посредством использования технологии Emscripten.

**Ход работы.**

Сборка и компиляция программы осуществляется с помощью утилиты emmake make. Графический интерфейс выполнен с помощью библиотеки [FLTK.](https://github.com/ocornut/imgui) В качестве GUI-совместимой с OpenGL библиотекой используется ImGUI.

Загрузка шейдеров происходит в несколько этапов. В самом начале их исходный код читается из указанных файлов в *vShaderCode* и *fShaderCode*. Затем этот исходный код компилируется:

vertex = glCreateShader(GL\_VERTEX\_SHADER); glShaderSource(vertex, 1, &vShaderCode, NULL); glCompileShader(vertex);

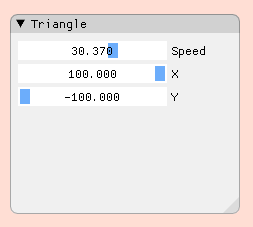
…

fragment = glCreateShader(GL\_FRAGMENT\_SHADER); glShaderSource(fragment, 1, &fShaderCode, NULL); glCompileShader(fragment);

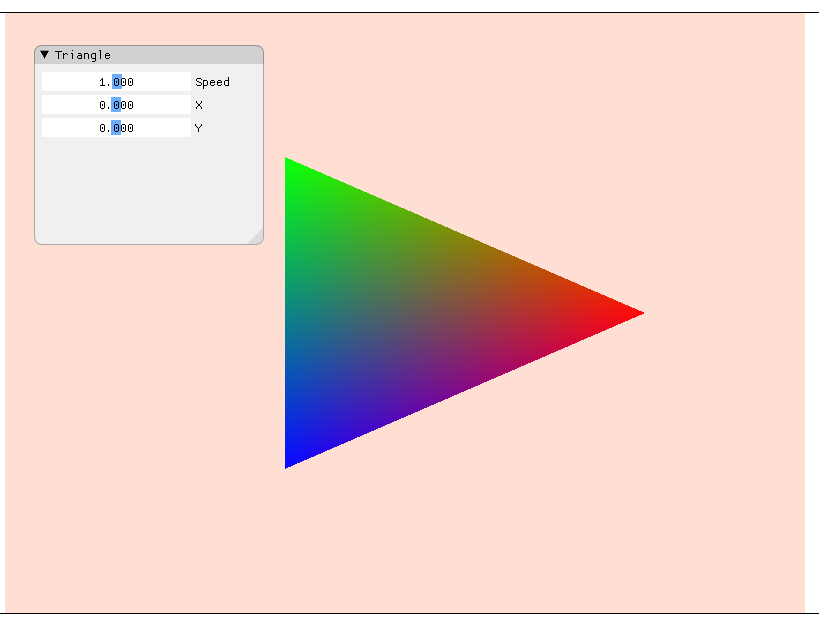
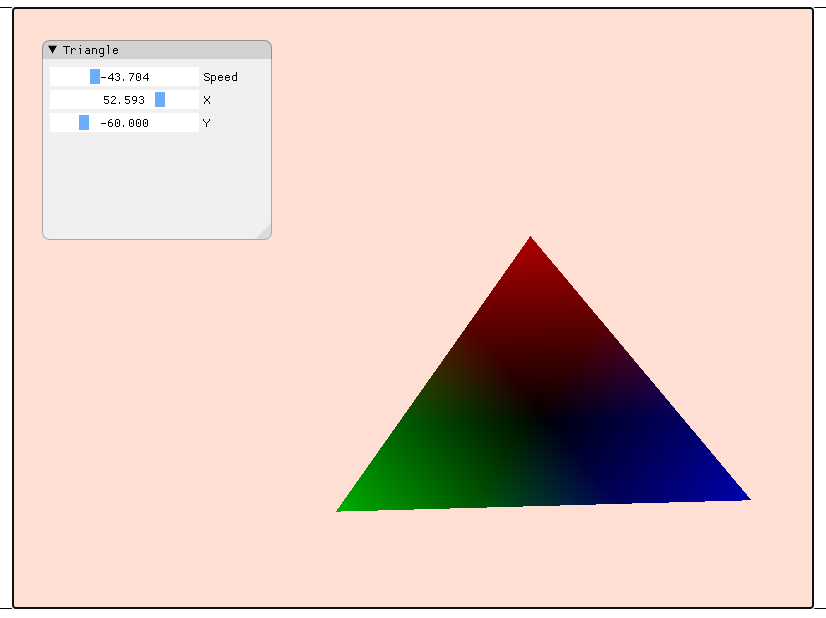
## При удачной компиляции полученные шейдерные программы линкуются в один шейдер с помощью команд:

this->Program = glCreateProgram(); glAttachShader(this->Program, vertex); glAttachShader(this->Program, fragment); glLinkProgram(this->Program);

Полученный таким образом шейдер мы затем используем в игровом цикле при отрисовке требуемой графики.

В области отрисовки находятся элементы UI, с помощью которых можно менять параметры треугольника: скорость и направление вращение, позицию фигуры по двум осям координат (ниже).

Вслед за изменением параметров с помощью мыши, треугольник изменяет скорость вращение, направление изменения цвета и позицию (ниже).



**Вывод.**

В процессе выполнения лабораторной работы была разработана программа, реализующая отрисовку OpenGL-контекста в браузере. При выполнении работы были приобретены навыки работы с графическим конвейером и шейдернымм программам из графической библиотеки OpenGL.

Исходный код проекта можно найти на Github:  
**https://github.com/makometr/CG\_9\_Emscripten/tree/main/Lab\_1**

Работающий прототип доступен по следующей ссылке (может не работать в некоторых браузерах, будет исправлено в будущем):  
**https://makometr.github.io/CG\_Lab\_1/EMSC.html**