

# Programming Assignment #1

Assigned: 10.03.2017 Due: 27.03.2017

Updated: 17.03.17

## 1. Objective

This is a WebGL programming homework. You will learn the basics of WebGL programming, how to pass information from application to shaders, different shader variables, color properties, etc. You will also add some interaction to control the parameters. You will just put all the necessary pieces we covered in the class together.

## 2. Specification

You will tessellate a triangle and twist it by a specified angle (See Figure 1). A 2D point can be rotated about the origin using the following equations:

$$x' = x \cos\theta - y \sin\theta$$

$$y' = x \sin\theta + y \cos\theta$$

where  $x$  and  $y$  are the original coordinates of the point, and  $x'$  and  $y'$  are the coordinates of the point after rotation. Now let amount of rotation depend on distance from origin giving us **twist**:

$$x' = x \cos(d\theta) - y \sin(d\theta)$$

$$y' = x \sin(d\theta) + y \cos(d\theta)$$

$$d = \sqrt{x^2 + y^2}$$

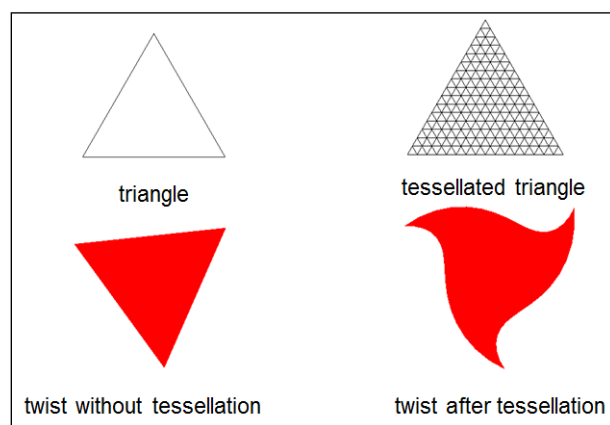


Figure 1: Left – Twisted triangle without tessellation. Right – Twisted triangle after tessellation.

In addition, assign a different color to each vertex. You can assign a fixed or a random color for each vertex. The final output of the program should look like the triangles in Figure 2.

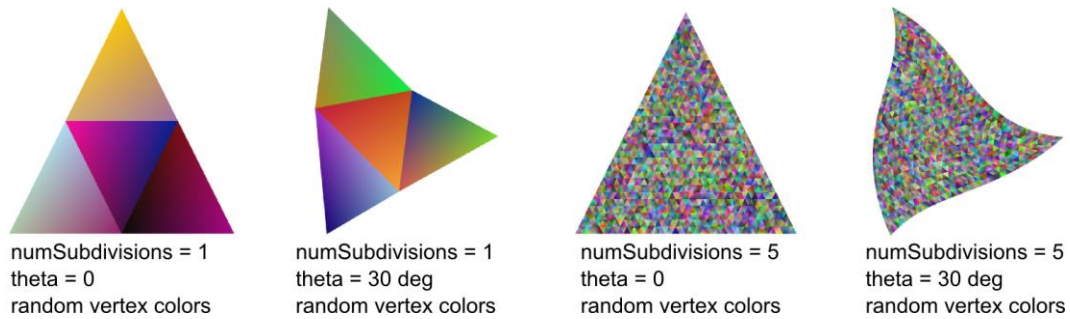


Figure 2: Different outputs from the final program, for different parameters.

### Interaction:

Now add some user interaction to control the parameters. You can control the following parameters: **rotation angle**, **twist or not twist option**, and **number of subdivisions**. Controlling the rotation angle and twist or not twist option is OBLIGATORY for the assignment. Controlling the number of subdivisions is more difficult and it is OPTIONAL with bonus points. (See Figure 3)

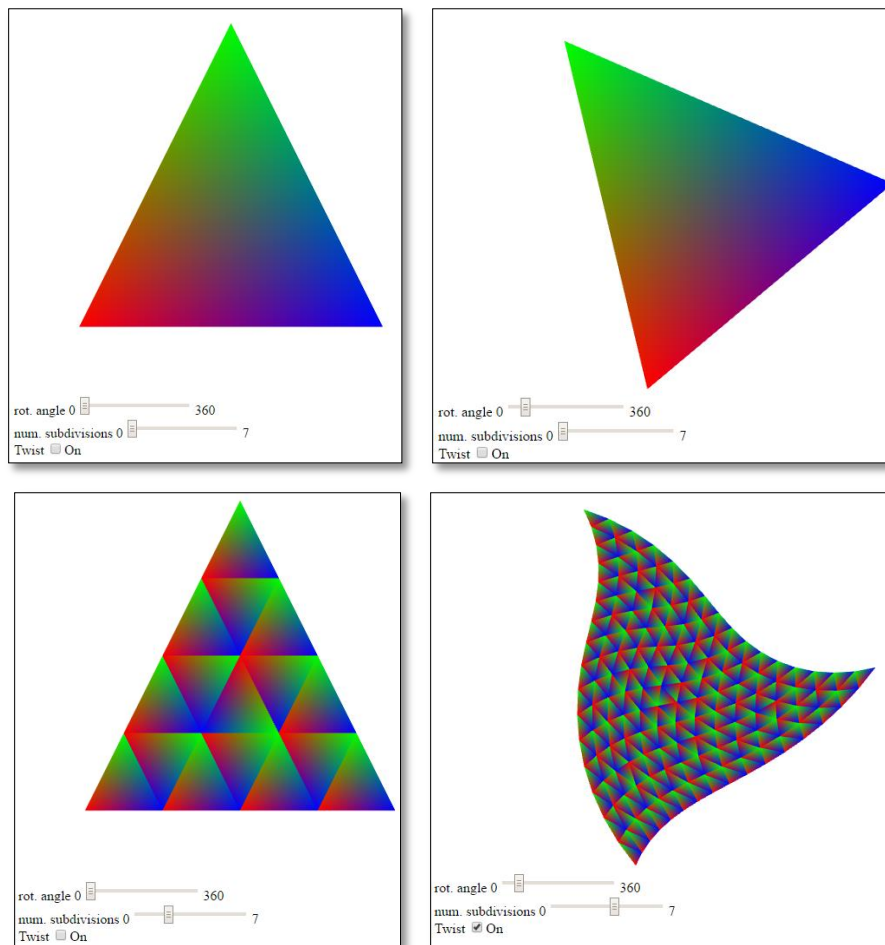


Figure 3: Sample outputs from the program.

**Notes:**

1. For tessellation, you can get help from the 2D Sierpinski gasket example of the book.
2. You should send the rotation angle ( $\theta$ ) from the application to the shader.
3. Trigonometric functions (sin, cos, tan, etc.) in GLSL, take parameter in radians.
4. Restrict number of subdivisions up to at most 10, otherwise too many triangles will be generated and rendering will be too slow.
5. In Google Chrome, you can right click on the browser and click on “inspect” and see the errors on console.
6. You can also use “console.log()” function in your javascript code for debugging purposes.

### 3. Submission

- This homework can be done individually or in pairs.
- Place all your source files in a zip archive with name **PA1\_Surname\_Name.zip** and submit through the Moodle submission module.
- If you have further questions, you can send me an e-mail or come to my office.

### 4. Late Submission Policy

Deadline for homework submissions is **23:59 pm** at the specified date. For each additional day, **25% cut-off** will be applied.

Assist. Prof. Dr.  
Zeynep ÇİPİLOĞLU YILDIZ