BBM 414 - Computer Graphics Laboratory

HACETTEPE UNIVERSITY

Department of Computer Engineering

Programming Assignment 2
Fall 2023-2024
Assoc. Prof. Dr. Ufuk Çelikcan

Due Date: 23:59 pm on Friday, November 10th, 2023

WebGL2 Shape Drawing and Basic Shading

Part 1

In this part, you will modify the given project by downloading from Piazza resources and obtain the shape of form given in Figure 1.

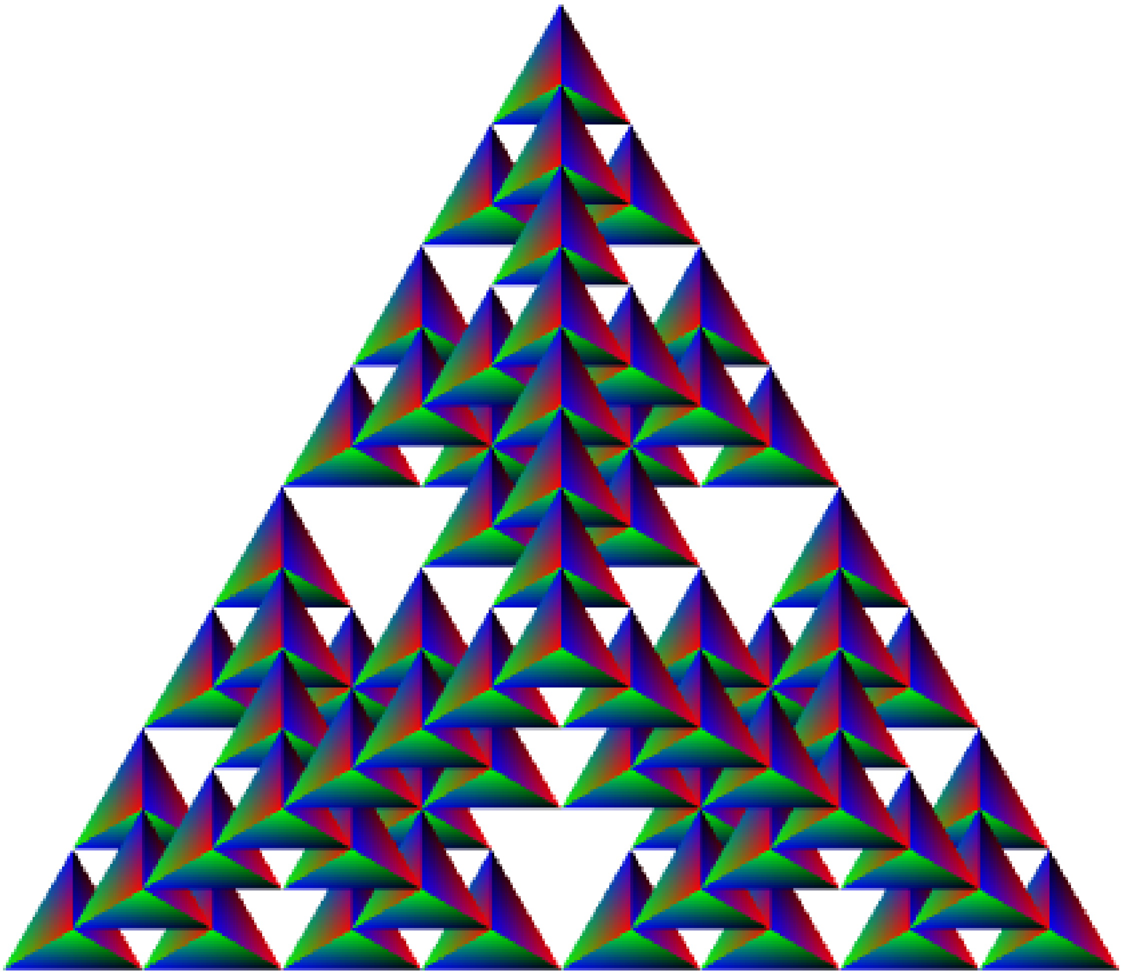

Figure 1: Hough Transform

Part 2

In this part, you will get familiar with simple translation and rotation using shader language. You will get keyboard keys from user and make an animation using the defined functions with respect to the given keys. The steps are explained in the following statements:

1. You will draw the shape from the previous assignment, snowflake (see Figure 2).

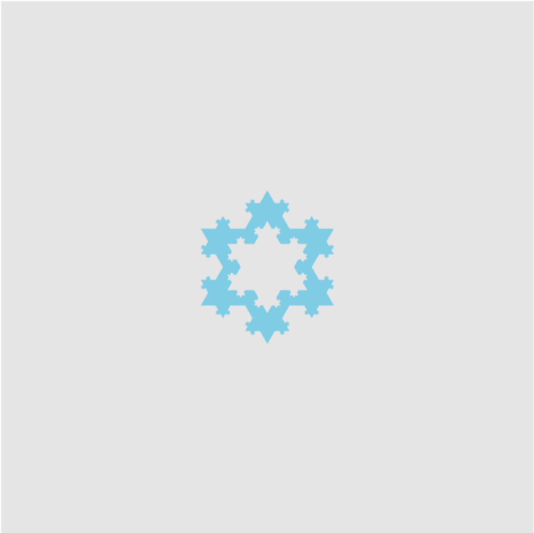

Figure 2: Snowflake centered in canvas.

2. You will translate 1 unit the shape on x and y axis using arrow keys (see Figure 3):
   up arrow: +y direction
   down arrow: -y direction
   left arrow: -x direction
   right arrow: +x direction.

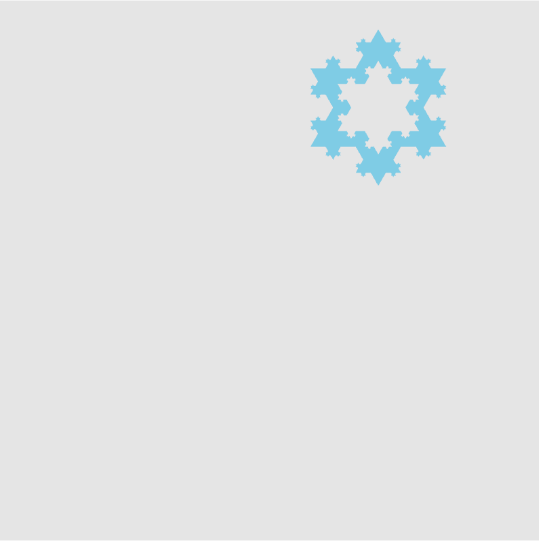

Figure 3: Translation on plane.

3. You will rotate 1 degree the shape around its center (not the center of canvas) using + and - keys (see Figure 4):
   +: clockwise rotation
   -: counterclockwise rotation.

4. (Step 1) Your shape will be at the default position given in Figure 1. Use ‘1’ key to switch to Step 1.

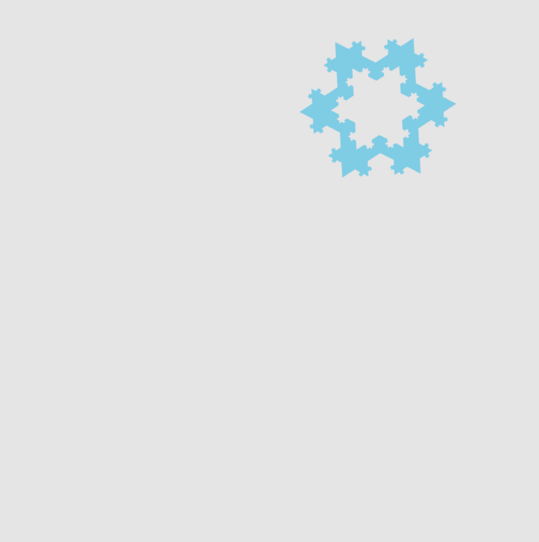

Figure 4: Rotation around its center.

5. (Step 2) You will change the vertex shader code to create swinging animation. The snowflake should not spin around its origin but should rotate [-80, 80]° to perform the swing animation (see Figure 5). Use '2' key to switch to Step 2.

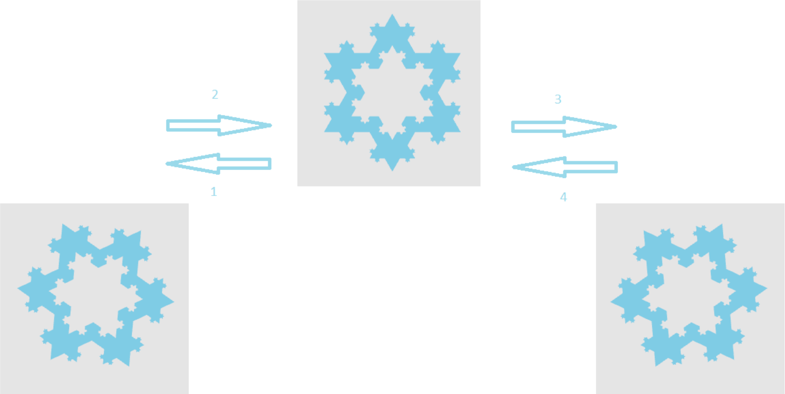

Figure 5: Rotation around its center.

6. (Step 3) You will change the fragment shader code to degrade the color of the shape according to its rotation. The shape should have its darkest color when its rotation is -80° and 80° and it should have its lightest color when its rotation is 0° (see Figure 6). Use '3' key to switch to Step 3.

The Implementation Details

1. Implement your homework using **WebGL2**. All programming assignments must use the shader-based functionality of **WebGL2**: at least one vertex shader and one fragment shader.

2. The assignment must be original work. Turning in someone else's work, in whole or in part, as your own will be considered as a violation of academic integrity. Please note that the former condition also holds for the material found on the web as everything on the web has been written by someone else. **Detection of such plagiarism in a submission will automatically void the submission and establish grounds to trigger an official disciplinary investigation.** General discussion of the assignment among peers is allowed, but do not share answers, algorithms or source codes. **Also using other resources (example source code, book, webpage etc.) as a code and javascript libraries (except jquery, Angel's book) are not allowed.**

3. Do not write the scripts into the html file. Reference your scripts in html.

4. You should use Netbeans or Webstorm as IDE for your projects.

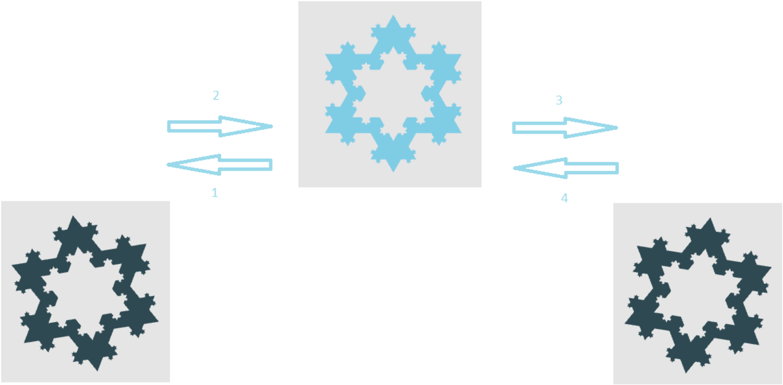

Figure 6: Rotation around its center.

The Report

You will write a report on latex for this assignment. You will explain the code parts and algorithm for part 1 and part 2.

What to Hand In

You should submit entire Netbeans or Webstorm project directory including javascript files and html file in a zip file extracted from IDE. Submission file structure is as given in below:

- b<studentNumber>.zip

  |–Experiment2_2023
  |——Part 1**(The whole Netbeans or Webstorm project)**
  |——Part 2**(The whole Netbeans or Webstorm project)**
  |–report.pdf

Archieve this folder as **b<studentNumber>.zip** and send via Piazza Private Post.

Grading

The assignment will be graded out of 100:

- PART 1 - CODE:0 (no implementation)
  20 (correct solution).

- PART 2 - CODE: 0 (no implementation)
  10 (shape),
  10 (translation - 4 directions),
  10 (rotation - clockwise and counter-clockwise),
  10 (step 1),
  10 (step 2),
  10 (step 3)

- REPORT: 20

Academic Integrity

All work on assignments must be done individually unless stated otherwise. You are encouraged to discuss with your classmates about the given assignments, but these discussions should be carried out in an abstract way. That is, discussions related to a particular solution to a specific problem (either in actual code or in the pseudocode) will not be tolerated. In short, turning in someone else's work, in whole or in part, as your own will be considered as a violation of academic integrity. Please note that the former condition also holds for the material found on the web as everything on the web has been written by someone else.

References

[1] https://xingangpan.github.io/projects/CULane.html