BBM414 Computer Graphics Lab.
Assignment 1 - Report

Berk Bubuş
21945939
Department of Computer Engineering
Hacettepe University
Ankara, Turkey
b21945939@cs.hacettepe.edu.tr

Overview

In this assignment, we are expected to draw a snowflake that is similar to Koch Snowflake but not same.

1 Part 1 - Using Examples as a Starter Code

I used some parts of the lesson pdf's. These are some parts:

```
<!DOCTYPE html>
<!--
Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
Click nbfs://nbhost/SystemFileSystem/Templates/Other/html.html to edit this template
-->
<html>
    <head>
        <title>Assignment 1 - Berk Bubuş</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <script src="js/initialize.js" defer></script>
        <script src="js/shaders.js" defer></script>
        <script src="js/app.js" defer></script>
    </head>
    <body>
        <canvas id="glCanvas" width="840" height="840"></canvas>
    </body>
</html>
```

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/ClientSide/javascript.js to edit this template
 */

const vsSource = `
    attribute vec4 a_position;
    void main(){
        gl_Position = a_position;
    }
`;

const fsSource1 =  `
    void main(){
        gl_FragColor = vec4(0.0,0.0,1.0,1.0);
    }
`;

const fsSource2 =  `
    void main(){
        gl_FragColor = vec4(1.0,1.0,1.0,1.0);
    }
`;
```

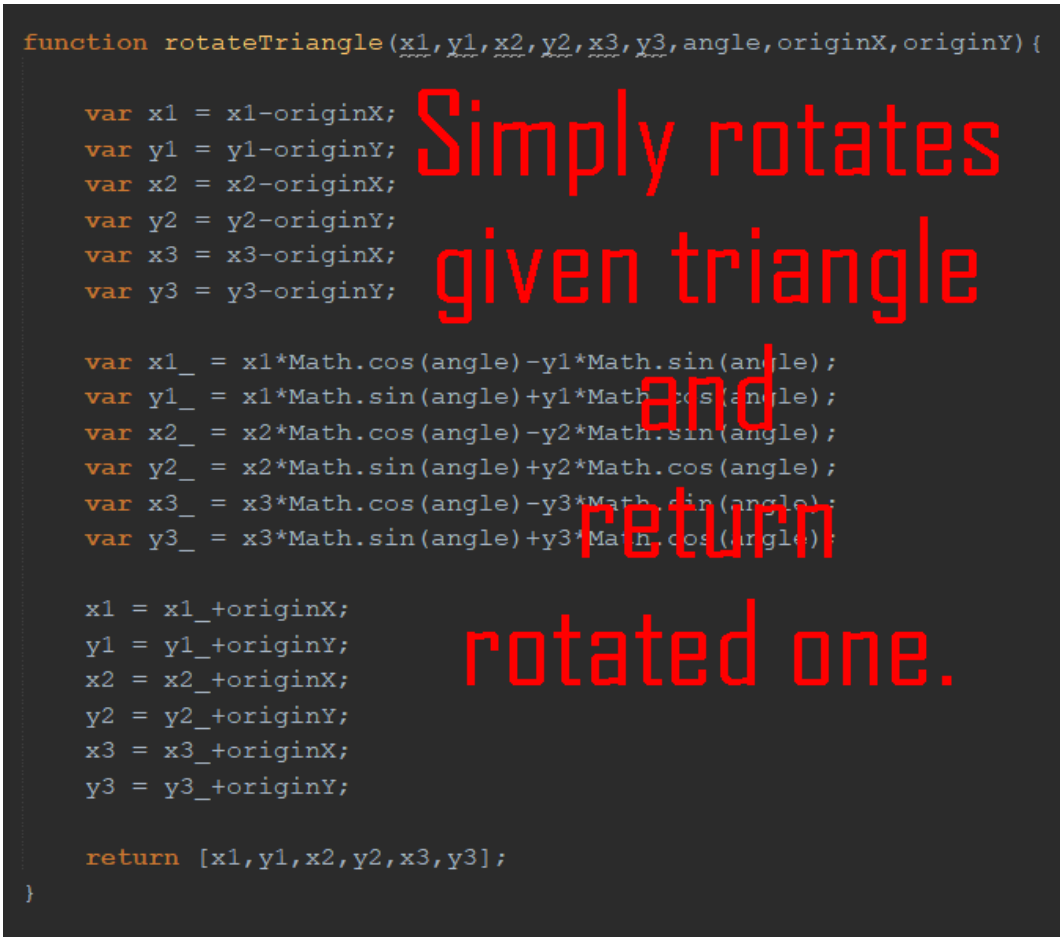
There was just one fsSource in this starter code. I added one more to use for second snowflake. I changed colors too.

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/ClientSide/javascript.js to edit this template
 */


/**
 * Comment
 */
function loadShader(gl, type, source) {
    const shader = gl.createShader(type);

    gl.shaderSource(shader, source);
    gl.compileShader(shader);

    if(!gl.getShaderParameter(shader,gl.COMPILE_STATUS)){
        alert('An error occurred compiling the shaders: ' + gl.getShaderInfoLog(shader));
        gl.deleteShader(shader);
        return null;
    }

    return shader;
}

function initShaderProgram(gl, vsSource, fsSource) {
    const vertexShader = loadShader(gl, gl.VERTEX_SHADER, vsSource);
    const fragmentShader = loadShader(gl, gl.FRAGMENT_SHADER, fsSource);

    const shaderProgram = gl.createProgram();
    gl.attachShader(shaderProgram, vertexShader);
    gl.attachShader(shaderProgram, fragmentShader);
    gl.linkProgram(shaderProgram);

    if(!gl.getProgramParameter(shaderProgram, gl.LINK_STATUS)){
        alert('Unable to initialize the shader program: '+ gl.getProgramInfoLog(shaderProgram));
        return null;
    }
    return shaderProgram;
}

function initBuffer(gl, positions){
    const positionBuffer = gl.createBuffer();

    gl.bindBuffer(gl.ARRAY_BUFFER, positionBuffer);

    gl.bufferData(gl.ARRAY_BUFFER, new Float32Array(positions), gl.STATIC_DRAW);

    return {
            position: positionBuffer};
}
```

2 Part 2 - App.js

In app.js file, I have 3 functions one of which is main(), I'll explain them part by part:

2.1 rotateTriangle()



2.2 makeSnowflake()

2.3 main()