

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

КОНТРОЛЬНА РОБОТА №1

Виконав:
Студент групи ІП-15
Мешков А.І.

Перевірів:
доц. каф. ІПІ
Родіонов П.Ю.

Київ 2023

ХІД РОБОТИ

1. Використовуючи графічний прикладний програмний інтерфейс WebGL створити новий документ та встановити довільний колір фону.
2. Написати програму, що виводить на екран зображення першої літери Вашого прізвища(Літера «М»).
3. Реалізувати використання довільних кольорів для літери.
4. Застосувати до літери анімацію з довільними налаштуваннями.

Лістинг 1 - Програмний код у файлі HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Контрольна робота 1</title>
  <script src="script.js" defer></script>
</head>
<body>
  <canvas id="myCanvas" width="500" height="500"></canvas>
</body>
</html>
```

Лістинг 2 - Програмний код у файлі JavaScript

```
const canvas = document.getElementById("myCanvas");
const gl = canvas.getContext("webgl");

if (!gl) {
  console.error("Unable to initialize WebGL. Your browser may not support it.");
}

const vertices = new Float32Array([
  0.0, 0.0,
  0.4, 0.5,
  0.5, 0.5,
  0.5, -0.5,
  0.4, -0.5,

  0.4, 0.0,
  0, -0.25,
  -0.4, 0.0,

  -0.4, -0.5,
  -0.5, -0.5,
  -0.5, 0.5,
  -0.4, 0.5,
]);
```

```

const vertexBuffer = gl.createBuffer();
gl.bindBuffer(gl.ARRAY_BUFFER, vertexBuffer);
gl.bufferData(gl.ARRAY_BUFFER, vertices, gl.STATIC_DRAW);

const numVertices = 12;

const vsSource = `
    attribute vec2 coordinates;
    void main(void) {
        gl_Position = vec4(coordinates, 0.0, 1.0);
    }
`;

const fsSource = `
    precision mediump float;
    uniform vec4 randomColor;
    void main(void) {
        gl_FragColor = randomColor;
    }
`;

const vertexShader = gl.createShader(gl.VERTEX_SHADER);
gl.shaderSource(vertexShader, vsSource);
gl.compileShader(vertexShader);

const fragmentShader = gl.createShader(gl.FRAGMENT_SHADER);
gl.shaderSource(fragmentShader, fsSource);
gl.compileShader(fragmentShader);

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

const coord = gl.getAttribLocation(shaderProgram, "coordinates");

gl.enableVertexAttribArray(coord);

gl.vertexAttribPointer(coord, 2, gl.FLOAT, false, 0, 0);

let yTranslate = 0.0;
let xTranslate = 0.0;
let xDirection = 1;
let yDirection = 1;

function getRandomColor() {
    return [
        Math.random(),
        Math.random(),
        Math.random(),
        1.0
    ];
}

function render() {
    gl.clear(gl.COLOR_BUFFER_BIT);
    gl.clearColor(0.2, 0.4, 0.6, 1.0);
    yTranslate += 0.01 * yDirection;

    if (yTranslate > 0.5 || yTranslate < -0.5) {
        yDirection *= -1;
    }
    xTranslate += 0.01 * xDirection;

```

```

    if (xTranslate > 0.5 || xTranslate < -0.5) {
        xDirection *= -1;
    }

    const colors = getRandomColor();
    gl.uniform4fv(gl.getUniformLocation(shaderProgram, "randomColor"),
colors);

    const vertices = [
        0.0 + xTranslate, 0.0 + yTranslate,
        0.4 + xTranslate, 0.5 + yTranslate,
        0.5 + xTranslate, 0.5 + yTranslate,
        0.5 + xTranslate, -0.5 + yTranslate,
        0.4 + xTranslate, -0.5 + yTranslate,

        0.4 + xTranslate, 0.0 + yTranslate,
        0 + xTranslate, -0.25 + yTranslate,
        -0.4 + xTranslate, 0.0 + yTranslate,

        -0.4 + xTranslate, -0.5 + yTranslate,
        -0.5 + xTranslate, -0.5 + yTranslate,
        -0.5 + xTranslate, 0.5 + yTranslate,
        -0.4 + xTranslate, 0.5 + yTranslate,
    ];

    gl.bufferData(gl.ARRAY_BUFFER, new Float32Array(vertices),
gl.STATIC_DRAW);
    gl.drawArrays(gl.TRIANGLE_FAN, 0, vertices.length / 2);

    requestAnimationFrame(render);
}

requestAnimationFrame(render);

```



Рисунок 1 – Результат виконання завдання