Kathmandu University

Department of Computer Science and Engineering

Dhulikhel, Kavre

A Report on '**Lab Work 4'** [COMP 342]

Submitted by:

Chandan Kumar Mahato (31)

III-year, II semester

Submitted to:

Mr. Dhiraj Shrestha

Department of Computer Science and Engineering

Submission Date: April 29, 2023

Qlab1

1. Write a Program to implement:

1. **2D Translation**
2. **2D Rotation**
3. **2D Scaling**
4. **2D Reflection**
5. **2D Shearing**

Solution:

https://ckmwebgl.netlify.app

I have used JavaScript as the programming language and WebGL as the graphics library. **JavaScript** is a popular programming language that is used for web development and has the capability of creating interactive graphics and animations on the web. **WebGL** is a graphics library that enables high-performance 3D graphics library that enables high-performance 3D graphics rendering in web browser using JavaScript. These technologies provide a powerful platform for creating engaging and interactive graphics, which is essential for my lab work.

The code snippets for setting graphics environment in my chosen graphics library and programming language and display system resolution are as follow:

Source Code

https://github.com/ChandankMahato/Graphics_Lab_6th_Sem

Matrix Multiplication:

```
function matrixMultiplication(Transformer, coordinates) {
  let result = [];
  for (let i = 0; i < 3; i++) {
    let sum = 0;
    for (let j = 0; j < 3; j++) {
      sum += Transformer[i * 3 + j] * coordinates[j];
    }
    result.push(sum);
  }
  return result;
}
```

2D Triangle:

```
let vertexA = [0.5, 0.1, 1];
let vertexB = [0.5, 0.5, 1];
let vertexC = [0.1, 0.1, 1];
function drawTriangle() {
  let vertexData = [];
  vertexData.push(...vertexA, ...vertexB, ...vertexC);
  DrawObject(gl.TRIANGLES, 3, Red, vertexData, 0, vertexData.length);
}
```

2D Translation:

```
function translateObject() {
  let vertexData = [];
  let translationMatrix = [
      ...[1, 0, -0.6],
      ...[0, 1, -0.6],
      ...[0, 0,    1]
      ];
  vertexData.push(...matrixMultiplication(translationMatrix, vertexA));
  vertexData.push(...matrixMultiplication(translationMatrix, vertexB));
  vertexData.push(...matrixMultiplication(translationMatrix, vertexC));
  console.log(vertexData);
  DrawObject(gl.TRIANGLES, 3, White, vertexData, 0, vertexData.length);
}
```

2D Rotation:

```
function rotateObject() {
  let vertexData = [];
  let rotationMatrix = [
      ...[0, -1, 0],
      ...[1, 0,  0],
      ...[0, 0,  1]
      ];
  vertexData.push(...matrixMultiplication(rotationMatrix, vertexA));
  vertexData.push(...matrixMultiplication(rotationMatrix, vertexB));
  vertexData.push(...matrixMultiplication(rotationMatrix, vertexC));
  DrawObject(gl.TRIANGLES, 3, White, vertexData, 0, vertexData.length);
}
```

2D Scaling:

```
function scaleObject() {
  let vertexData = [];
  let scalingMatrix = [
      ...[1.5, 0, 0],
      ...[0, 1.5, 0],
      ...[0, 0,   1]];
  vertexData.push(...matrixMultiplication(scalingMatrix, vertexA));
  vertexData.push(...matrixMultiplication(scalingMatrix, vertexB));
  vertexData.push(...matrixMultiplication(scalingMatrix, vertexC));
  DrawObject(gl.TRIANGLES, 3, White, vertexData, 0, vertexData.length);
}
```

2D Reflection:

```
function reflectObject() {
  let vertexData = [];
  let reflectionMatrix = [
      ...[1,  0, 0],
      ...[0, -1, 0],
      ...[0,  0, 1]];
  vertexData.push(...matrixMultiplication(reflectionMatrix, vertexA));
  vertexData.push(...matrixMultiplication(reflectionMatrix, vertexB));
  vertexData.push(...matrixMultiplication(reflectionMatrix, vertexC));
  DrawObject(gl.TRIANGLES, 3, White, vertexData, 0, vertexData.length);
}
```

2D Shearing:

```
function shearObject() {
  let vertexData = [];
  let shearMatrix = [
      ...[1, 0.5, 0],
      ...[0, 1,   0],
      ...[0, 0,   1]];
  vertexData.push(...matrixMultiplication(shearMatrix, vertexA));
  vertexData.push(...matrixMultiplication(shearMatrix, vertexB));
  vertexData.push(...matrixMultiplication(shearMatrix, vertexC));
  DrawObject(gl.TRIANGLES, 3, White, vertexData, 0, vertexData.length);
}
```

Output and Screenshots:

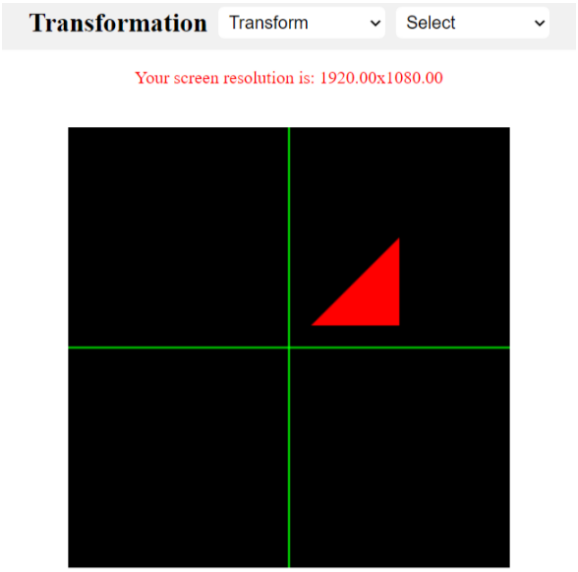

Fig 1: Triangle

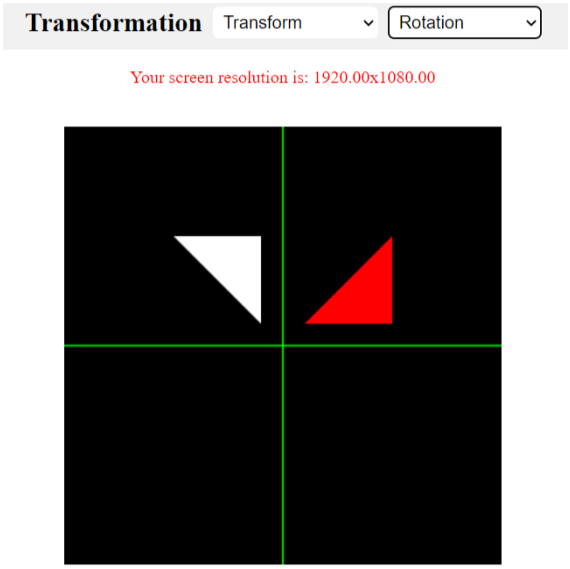

Fig 2: Rotation

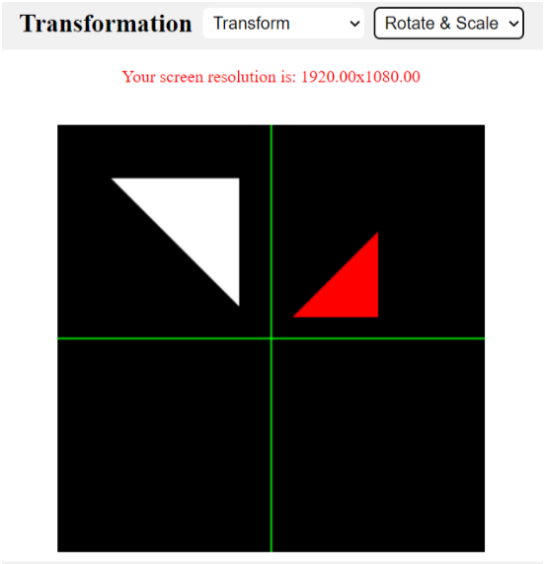

Fig 1: Rotation & Scaling

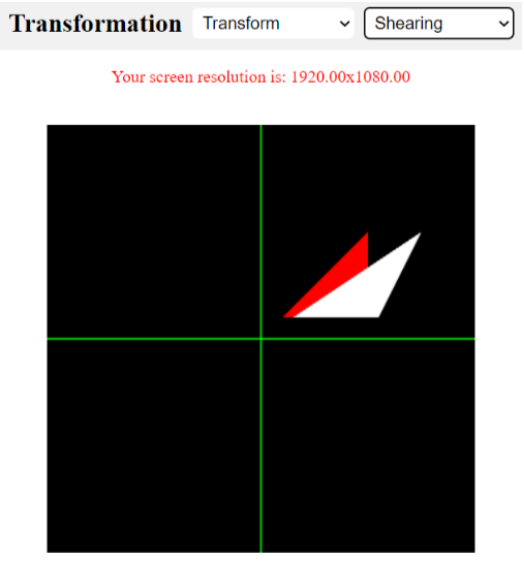

Fig 2: Shearing

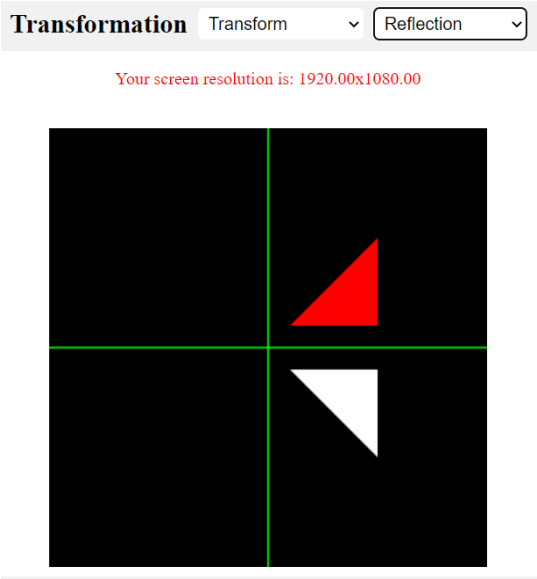

Fig 1: Reflection

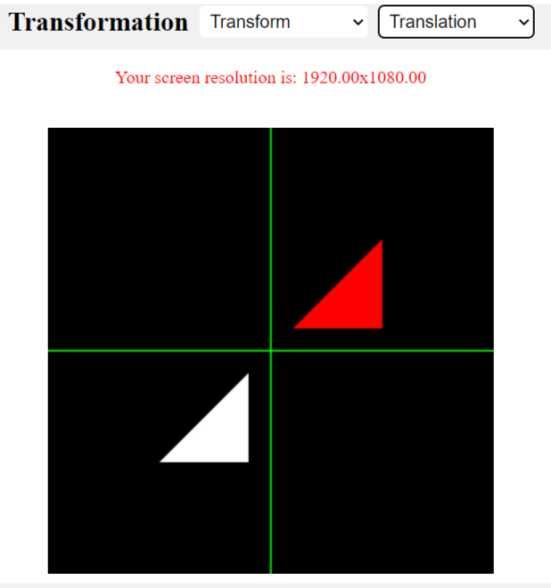

Fig 2: Translation

Conclusion

After implementing the 2D Translation, Rotation, Scaling, Reflection, and Shearing programs, I have gained a deeper understanding of the fundamental concepts of computer graphics. By utilizing matrix transformations and basic mathematical operations, I was able to manipulate and transform 2D objects in various ways.

This exercise has helped me to develop a better understanding of the concepts involved in 2D graphics transformations and how to implement them using programming languages. Overall, this exercise has enhanced my skills in computer graphics and has prepared me for more advanced graphics programming tasks.