Kathmandu University

Department of Computer Science and Engineering

Dhulikhel, Kavre

A Report on '**Lab Work 2'** [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 18, 2023

Qlab2

1. **Implement Digital Differential Analyzer Line drawing algorithm.**
2. **Implement Bresenham Line Drawing algorithm for both slopes (|m| <1 and |m| >=1).**

Solution:

https://flagwebgl.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

DDA Algorithm:

```
function drawDDALine() {
  let vertexData = [];
  let X0 = 200;
  let Y0 = 200;
  let X1 = 400;
  let Y1 = 400;

  let dx = Math.abs(X1 - X0);
  let dy = Math.abs(Y1 - Y0);
  let steps = dx > dy ? dx : dy;

  let Xinc = dx / steps;
  let Yinc = dx / steps;
```

```
  let X = X0;
  let Y = Y0;
  for (let i = 0; i < steps; i++) {
    vertexData.push(normalise(X, canvasWidth));
    vertexData.push(normalise(Y, canvasHeight));
    vertexData.push(0);
    X += Xinc;
    Y += Yinc;
  }
  let fragCode = `void main() {gl_FragColor = vec4(1, 0, 0, 1);}`; //red color
  DrawObject(gl.POINTS, 1, fragCode, vertexData, 0, vertexData.length);
}
```

BLA Algorithm:

```
function drawBLALine() {
  let vertexData = [];
  let X0 = 200;
  let Y0 = 200;
  let X1 = 400;
  let Y1 = 400;

  let dx = Math.abs(X1 - X0);
  let dy = Math.abs(Y1 - Y0);
  let slope = dy / dx;

  let X, Y, steps, p;
  if (slope < 1) {
    // m<1
    if (X0 > X1) {
      [X0, Y0, X1, Y1] = [X1, Y1, X0, Y0];
    }
    X = X0;
    Y = Y0;
    vertexData.push(normalise(X, canvasWidth));
    vertexData.push(normalise(Y, canvasHeight));
    steps = X1 - X0;
    p = 2 * dy - dx;
    for (let i = 0; i < steps; i++) {
      if (p >= 0) {
        Y++;
        p += 2 * dy - 2 * dx;
```

```
      } else {
        p += 2 * dy;
      }
      X++;
      vertexData.push(normalise(X, canvasWidth));
      vertexData.push(normalise(Y, canvasHeight));
    }
  } else {
    // m>1
    if (Y0 > Y1) {
      [X0, Y0, X1, Y1] = [X1, Y1, X0, Y0];
    }
    X = X0;
    Y = Y0;
    vertexData.push(normalise(X, canvasWidth));
    vertexData.push(normalise(Y, canvasHeight));
    steps = Y1 - Y0;
    p = 2 * dx - dy;
    for (let i = 0; i < steps; i++) {
      if (p >= 0) {
        X++;
        p += 2 * dx - 2 * dy;
      } else {
        p += 2 * dx;
      }
      Y++;
      vertexData.push(normalise(X, canvasWidth));
      vertexData.push(normalise(Y, canvasHeight));
    }
  }
  let fragCode = `void main() {gl_FragColor = vec4(1, 0, 0, 1);}`; //red color
  DrawObject(gl.POINTS, 1, fragCode, vertexData, 0, vertexData.length);
}
```

Output and Screenshots:

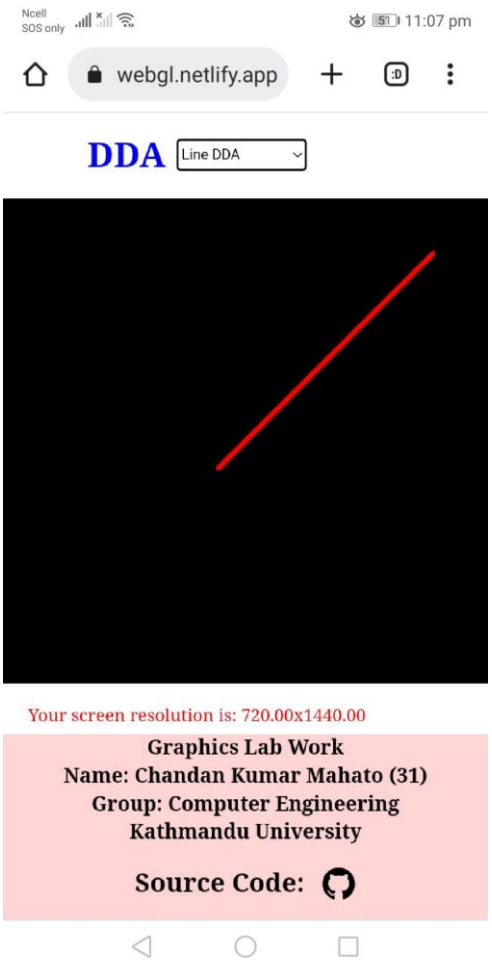

Fig 1: DDA

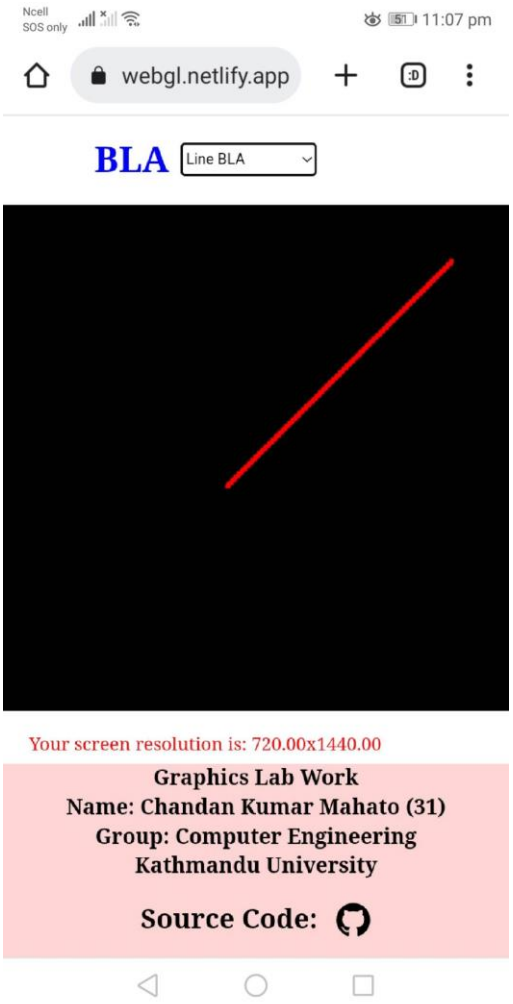

Fig 2: BLA

Conclusion

By getting familiar with the coordinate system and utilizing the graphics geometrical functions and classes provided by our chosen graphics library, I was able to successfully draw the line using Digital Differential Analyzer algorithm and Bresenham line drawing algorithm.

This exercise helped me to develop a better understanding of the coordinate system and the use of graphics geometrical functions and classes to create visually appealing graphics.