Graphics

Lab 3

Name / Ahmed Abdallah Aboeleid

ID / 19015274

Problem statement:

You should implement an OpenGL application that writes your first name using line drawing algorithms (DDA, Bresenham)

(your drawing should include all possible slopes).

You should use VAOs to store vertices data.

Set point size to 5.

Code:

```
#include <cmath>
#include <iostream>
#include <GL/glew.h>
#include <GL/freeglut.h>
#include <vector>
using namespace std;

int Choice = 0;
static unsigned int buffer[2];
vector<float> vertices(0);
```

⇨ First define global variable as choice to take input from user to determine which algorithm will used,

⇨ Second define vector vertices used to store all points which will used to draw name.

```
void Bresenham(int x0, int y0, int xl, int yl)
{
    bool steep = false;
    if (abs(xl-x0) < abs(yl-y0)){
        swap(x0, y0);
        swap(xl, yl);
        steep = true;
    }
    if (x0 > xl){
        swap(x0, xl);
        swap(y0, yl);
    }
    int dx = xl - x0;
    int dy = yl - y0;
    int derror2 = abs(dy) * 2;
    int error2 = 0;
    int y = y0;

    for (int x = x0; x <= xl; x++)
    {
        if (steep){
            vertices.push_back(y);
            vertices.push_back(x);
            vertices.push_back(0.0);
        }
        else{
            vertices.push_back(x);
            vertices.push_back(y);
            vertices.push_back(0.0);
        }
        error2 += derror2;
        if (error2 > dx){
            y += (yl > y0 ? 1 : -1);
            error2 -= dx * 2;
        }
    }
}
```

⇨ Here , I implement bresenham algorithm which take 2 points and draw line between them and put points in global vector I define before.

```
void DDA(int x0, int y0, int x1, int y1)
{
    float dx = x1 - x0;
    float dy = y1 - y0;
    float steps = max(abs(dx), abs(dy));
    float xIncr = dx / steps;
    float yIncr = dy / steps;
    float x = x0;
    float y = y0;

    for (int i = 0; i < steps; i++)
    {
        vertices.push_back(x);
        vertices.push_back(y);
        vertices.push_back(0.0);
        x += xIncr;
        y += yIncr;

    }
}
```

⇨ Here , I implement DDA algorithm which take 2 points and draw line between them and put points in global vector I define before.

```
// Initialization routine.
void setup(void)
{
    drawname();
    glGenVertexArrays(1, &vaoID);
    glBindVertexArray(vaoID);

    float colors[vertices.size()];
    glClearColor(0.0, 0.0, 0.0, 0.0);
    glGenBuffers(2, buffer); // Generate buffer ids.
    // Bind vertex buffer and reserve space.
    glBindBuffer(GL_ARRAY_BUFFER, buffer[0]);
    glBufferData(GL_ARRAY_BUFFER, vertices.size()*sizeof(float) + sizeof(colors), NULL, GL_STATIC_DRAW);
    // Copy vertex coordinates data into first half of vertex buffer.
    glBufferSubData(GL_ARRAY_BUFFER, 0, vertices.size()*sizeof(float), &vertices[0]);
    // Copy vertex color data into second half of vertex buffer.
    glBufferSubData(GL_ARRAY_BUFFER, vertices.size()*sizeof(float), sizeof(colors), colors);
    // Enable two vertex arrays: co-ordinates and color.
    glEnableClientState(GL_VERTEX_ARRAY);
    glEnableClientState(GL_COLOR_ARRAY);
    // Specify vertex and color pointers to the start of the respective data.
    glVertexPointer(3, GL_FLOAT, 0, 0);
    glColorPointer(3, GL_FLOAT, 0, (void *)(vertices.size()*sizeof(float)));

    glutTimerFunc(3, animate, 1);
}
```

⇨ I take this function from code which is given in pdf of lab and begin to change in to be suitable for my code.

```
void drawname()
{
    switch (Choice) {
        case 1:{
            // Letter A.
            Bresenham(0.0, 30.0, 12.5, 90.0);
            Bresenham(12.5, 90.0, 25.0, 30.0);
            Bresenham(6.25,60.0, 18.75, 60.0);
            // Letter H.
            Bresenham(30.0, 30.0, 30.0, 90.0);
            Bresenham(30.0, 60.0, 55.0, 60.0);
            Bresenham(55.0, 90.0,55.0, 30.0);
            // Letter M.
            Bresenham(60.0, 30.0, 60.0, 90.0);
            Bresenham(60.0, 90.0, 72.5, 60.0);
            Bresenham(72.5, 60.0, 85.0,90.0);
            Bresenham(85.0,30.0, 85.0,90.0);
            // Letter E.
            Bresenham(90.0,30.0, 90.0,90.0);
            Bresenham(90.0,30.0, 115.0,30.0);
            Bresenham(90.0,90.0, 115.0,90.0);
            Bresenham(90.0,60.0, 115.0,60.0);
            // Letter D.
            Bresenham(120.0, 30.0, 120.0, 90.0);
            Bresenham(120.0, 30.0, 145.0, 45.0);
            Bresenham(120.0, 90.0, 145.0, 45.0);
            break;
        }case 2:{
            // Letter A.
            DDA(0.0, 30.0, 12.5, 90.0);
            DDA(12.5, 90.0, 25.0, 30.0);
            DDA(6.25,60.0, 18.75, 60.0);
            // Letter H.
            DDA(30.0, 30.0, 30.0, 90.0);
            DDA(30.0, 60.0, 55.0, 60.0);
            DDA(55.0, 90.0,55.0, 30.0);
            // Letter M.
            DDA(60.0, 30.0, 60.0, 90.0);
            DDA(60.0, 90.0, 72.5, 60.0);
            DDA(72.5, 60.0, 85.0,90.0);
            DDA(85.0,30.0, 85.0,90.0);
            // Letter E.
```

⇨ This function used for creating lines from some points I give for algorithm to draw lines.

⇨ This function is call in setup function to do what I want.

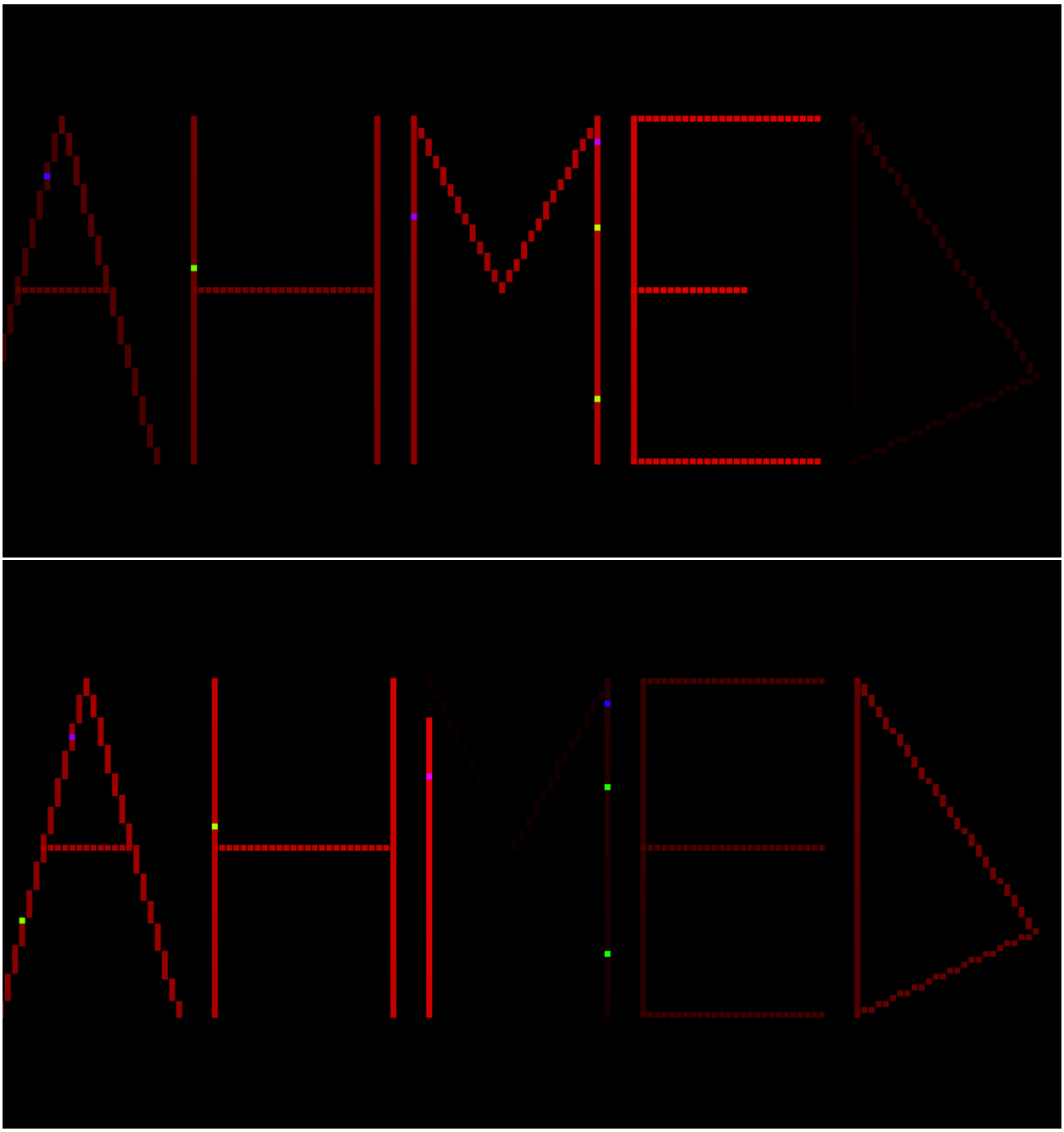
```
float color = 0.0f;
GLuint vaoID;
// Drawing routine.
void drawScene(void)
{
    glClear(GL_COLOR_BUFFER_BIT);
    glPointSize(5.0f);
    float* bufferData = (float*)glMapBuffer(GL_ARRAY_BUFFER, GL_WRITE_ONLY);
    float incre = 0.0011526;
    for (int i = 0; i < vertices.size()/3; i++)
    {
        bufferData[vertices.size() + i*3] = color;
        color += incre;
        if (color >= 0.9f ) {
            color = 0.0f;
        }
    }

    glUnmapBuffer(GL_ARRAY_BUFFER);
    glBindVertexArray(vaoID);
    glDrawArrays(GL_POINTS, 0, vertices.size() / 3);
    glutSwapBuffers();
}
```

⇨ I use for loop to Apply simple text gradient animation

Sample run :

⇨ DDA algorithm:

⇨ Bresenham



⇨ Link for animation video:
https://drive.google.com/file/d/1RjTolRVZ30L0vDbTAuvoOkN7G2LtzRvJ/view?usp=share_link