

Date: 19/08/2025

Experiment No: 02

Experiment Name: Basic Line Drawing Using Points.

Introduction

In computer graphics, drawing basic shapes is the first step toward creating more complex images and designs. One of the most important tasks is line drawing, since lines are used to form shapes, graphs, and outlines of objects.

Using OpenGL (GL), we can draw lines by plotting individual points on the screen. A line is basically a collection of points placed very close together so that they appear as a straight path between two positions. By controlling the starting and ending coordinates, we can display a line in any direction.

This experiment focuses on drawing a basic line using GL points. It helps us understand how computer graphics represent continuous objects like lines using discrete pixels on the screen. Learning line drawing is important because it is the foundation for developing more advanced graphics algorithms, such as drawing polygons, curves, and 3D

Description:

This lab demonstrates basic 2D graphics using OpenGL and GLUT by drawing lines and points. The program opens a window and uses the `display()` function to render graphics. The coordinate system ranges from -1.0 to 1.0 along both x and y axes.

First, the color buffer is cleared to black using `glClear(GL_COLOR_BUFFER_BIT)`. Then, white coordinate axes are drawn with `GL_LINES` to provide a reference for the 2D space.

The main part of the lab focuses on drawing custom shapes using `GL_POINTS` in red color (`glColor3f(0.9, 0.3, 0.3)`) and varying point sizes with `glPointSize()`.

1. The letter "A" is drawn in the upper-right quadrant:

- a) Left diagonal: points along $x=y$.
- b) Right diagonal: points forming the other side.
- c) Crossbar: horizontal points at a fixed y-coordinate.

2. The letter "E" is drawn in the lower-left quadrant:

- a) Vertical backbone: points along a fixed x-coordinate.
- b) Top, middle, and bottom bars: horizontal points at different y-coordinates.

c) Finally, `glFlush()` is called to display all the points.

The `main()` function initializes GLUT, sets the window size to 640×640, creates the window titled "OpenGL Graphics," registers the `display()` function, and starts the GLUT event loop with `glutMainLoop()`, which keeps the program running and handles redraw events.

Code:

```
#include <windows.h>
#include <GL/glut.h>

void display() {
    glClear(GL_COLOR_BUFFER_BIT);
    glBegin(GL_LINES);

    glVertex2f(1, 0.0);
    glVertex2f(-1, 0.0);

    glVertex2f(0.0, 1);
    glVertex2f(0.0, -1);
    glEnd();

    // A
    glPointSize(10);
    glBegin(GL_POINTS);
    glColor3f(1, 0.3, 0.3);
    for (float i = 0.3; i <= 0.9; i = i + (0.001)) {
        glVertex2f(i, 0.3);
    }
    glEnd();

    glPointSize(10);
    glBegin(GL_POINTS);
    glColor3f(1, 0.3, 0.3);
    for (float i = 0.2; i <= 0.6; i = i + (0.001)) {
        glVertex2f(i, i);
    }
    glEnd();

    glPointSize(10);
    glBegin(GL_POINTS);
    glColor3f(1, 0.3, 0.3);
    for (float i = 0.0; i <= 0.37; i = i + (0.001)) {
        glVertex2f(0.6 + i, 0.6 - i);
```

```

    }
    glEnd();

    // E
    glPointSize(10);
    glBegin(GL_POINTS);
    glColor3f(1, 0.3, 0.3);
    for (float i = 0.1; i <= 0.6; i = i + (0.001)) {
        glVertex2f(-0.6, -i);
    }
    glEnd();

    glPointSize(10);
    glBegin(GL_POINTS);
    glColor3f(1, 0.3, 0.3);
    for (float i = 0.6; i >= 0.1; i = i - (0.001)) {
        glVertex2f(-i, -0.6);
    }
    glEnd();

    glPointSize(10);
    glBegin(GL_POINTS);
    glColor3f(1, 0.3, 0.3);
    for (float i = 0.6; i >= 0.1; i = i - (0.001)) {
        glVertex2f(-i, -0.35);
    }
    glEnd();

    glPointSize(10);
    glBegin(GL_POINTS);
    glColor3f(1, 0.3, 0.3);
    for (float i = 0.6; i >= 0.1; i = i - (0.001)) {
        glVertex2f(-i, -0.1);
    }
    glEnd();

    glFlush();

}

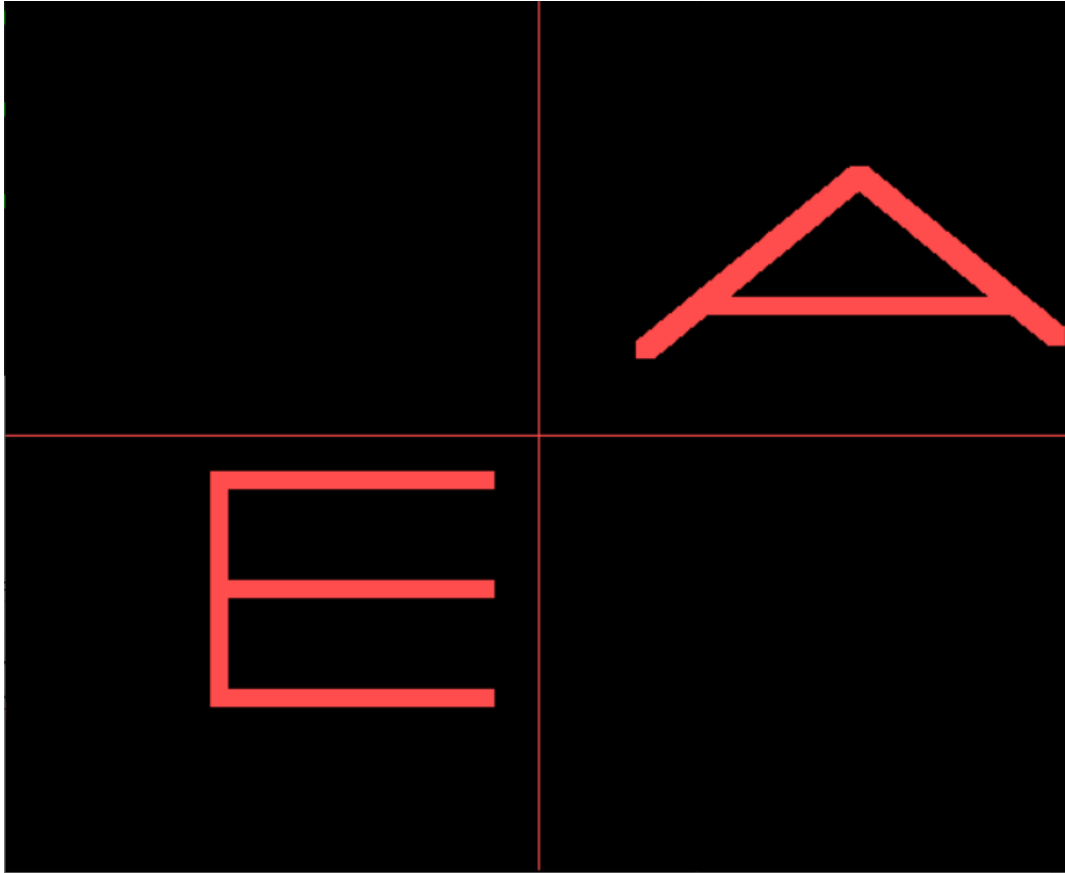
int main(int argc, char* argv[]) {

    glutInit(&argc, argv);
    glutInitWindowSize(440, 480);
    glutInitWindowPosition(0, 0);
    glutInitDisplayMode(GLUT_RGB | GLUT_SINGLE | GLUT_DEPTH);

```

```
glutCreateWindow(" Gult_Graphics");  
glutDisplayFunc(display);  
glutMainLoop();  
  
return 0;  
}
```

Output:



Conclusion:

In this experiment, we learned how to draw letters and shapes using GL_POINTS in OpenGL. By plotting points with specific positions, size, and color, we can form graphics like the letters "A" and "E." This shows how continuous objects are represented using discrete pixels in computer graphics.