Practical no.02

Name:-Ujawala Sinha
Roll no.S554

Title:-Bresenham Line Drawing algorithm

code:

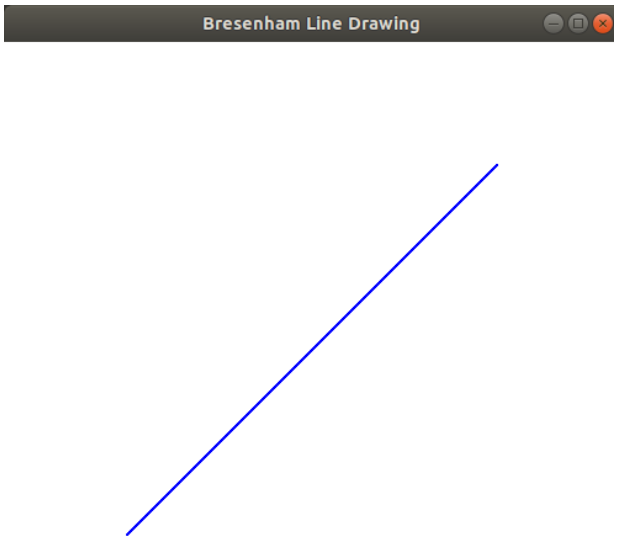
```
#include <GL/glut.h>
#include <iostream>

int lineType = 1;  // Default: Simple Line

// Fixed Start and End Points
const int x1 = 100, y1 = 100, x2 = 400, y2 = 400;

// Function to draw a pixel at (x, y)
void drawPixel(int x, int y) {
    glBegin(GL_POINTS);
    glVertex2i(x, y);
    glEnd();
}

// Bresenham's Line Algorithm (Different Styles)
void drawBresenhamLine(int style) {
    int dx = abs(x2 - x1), dy = abs(y2 - y1);
    int sx = (x2 >= x1) ? 1 : -1;
    int sy = (y2 >= y1) ? 1 : -1;
    int err = dx - dy, e2;

    int count = 0;
    int x = x1, y = y1;

    while (true) {
        switch (style) {
            case 1:  // Simple Line
                drawPixel(x, y);
                break;
            case 2:  // Dotted Line
                if (count % 2 == 0) drawPixel(x, y);
                break;
            case 3:  // Dashed Line
                if (count % 10 < 6) drawPixel(x, y);
                break;
        }

        if (x == x2 && y == y2) break; // Stop when end point is reached

        e2 = 2 * err;
        if (e2 > -dy) { err -= dy; x += sx; }
        if (e2 < dx) { err += dx; y += sy; }
```

```
        count++;
    }
}

// OpenGL Display Function
void display() {
    glClear(GL_COLOR_BUFFER_BIT);
    drawBresenhamLine(lineType);
    glFlush();
}

// OpenGL Initialization
void initOpenGL() {
    glClearColor(1, 1, 1, 1); // Black background
    glColor3f(0, 0, 1);       // White line
    glPointSize(2);           // Pixel size
    gluOrtho2D(0, 500, 0, 500); // 2D coordinate system
}

// Get user choice
void getUserChoice() {
    std::cout << "Select Line Type:\n";
    std::cout << "1. Simple Line\n2. Dotted Line\n3. Dashed Line\n";
    std::cout << "Enter your choice: ";
    std::cin >> lineType;

    if (lineType < 1 || lineType > 3) {
        std::cout << "Invalid choice, defaulting to Simple Line.\n";
        lineType = 1;
    }
}

// Main Function
int main(int argc, char** argv) {
    getUserChoice();  // Get user input before creating the window

    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
    glutInitWindowSize(500, 500);
    glutCreateWindow("Bresenham Line Drawing");

    initOpenGL();
    glutDisplayFunc(display);
    glutMainLoop();
    return 0;
}
```

Output

```
svpm@svpm-HP-EliteDesk-800-G2-SFF:~$ g++ breshline.cpp -lGL -lGLU -lglut
svpm@svpm-HP-EliteDesk-800-G2-SFF:~$ ./a.out
Select Line Type:
```

1. Simple Line
2. Dotted Line
3. Dashed Line
Enter your choice: 1



svpm@svpm-HP-EliteDesk-800-G2-SFF:~$ g++ breshline.cpp -lGL -lGLU -lglut
svpm@svpm-HP-EliteDesk-800-G2-SFF:~$ ./a.out
Select Line Type:
1. Simple Line
2. Dotted Line
3. Dashed Line
Enter your choice: 2

```
svpm@svpm-HP-EliteDesk-800-G2-SFF:~$ g++ breshline.cpp -lGL -lGLU -lglut
svpm@svpm-HP-EliteDesk-800-G2-SFF:~$ ./a.out
Select Line Type:
1. Simple Line
2. Dotted Line
3. Dashed Line
Enter your choice: 3
```