

## EXPERIMENT – 2

Drawing a line using Bresenham Algorithm

CODE :

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

int x1, y1, x2, y2;

void myInit() {
    glClear(GL_COLOR_BUFFER_BIT);
    glClearColor(0.0, 0.0, 0.0, 1.0);
    glMatrixMode(GL_PROJECTION);
    gluOrtho2D(0, 500, 0, 500);
}

void draw_pixel(int x, int y) {
    glBegin(GL_POINTS);
    glVertex2i(x, y);
    glEnd();
}

void draw_line(int x1, int x2, int y1, int y2)
{
    int dx, dy, i, e;
    int incx, incy, inc1, inc2;
    int x,y;

    dx = x2-x1;
    dy = y2-y1;

    if (dx < 0) dx = -dx;
    if (dy < 0) dy = -dy;
    incx = 1;
    if (x2 < x1) incx = -1;
    incy = 1;
    if (y2 < y1) incy = -1;
    x = x1; y = y1;
```

```

        if (dx > dy) {
            draw_pixel(x, y);
            e = 2 * dy - dx;
            inc1 = 2 * (dy - dx);
            inc2 = 2 * dy;
            for (i=0; i<dx; i++) {
                if (e >= 0) {
                    y += incy;
                    e += inc1;
                }
                else
                    e += inc2;
                x += incx;
                draw_pixel(x, y);
            }

        } else {
            draw_pixel(x, y);
            e = 2 * dx - dy;
            inc1 = 2 * (dx - dy);
            inc2 = 2 * dx;
            for (i=0; i<dy; i++) {
                if (e >= 0) {
                    x += incx;
                    e += inc1;
                }
                else
                    e += inc2;
                y += incy;
                draw_pixel(x, y);
            }
        }
    }

void myDisplay() {
    draw_line(x1, x2, y1, y2);
    glFlush();
}

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

    printf( "Enter (x1, y1, x2, y2)\n");
    scanf("%d %d %d %d", &x1, &y1,

```

```
    glutInit(&argc, argv);  
    glutInitDisplayMode(GLUT_SINGLE);  
    glutInitWindowSize(500, 500);  
    glutInitWindowPosition(0, 0);  
    glutCreateWindow("Bresenham's  
    myInit());  
    glutDisplayFunc(myDisplay);  
    glutMainLoop();  
}
```

## OUTPUT :



