

UCS1712 - GRAPHICS AND MULTIMEDIA LAB

EX - 3: Drawing 2D Primitives –Line – Bresenham's Algorithm

NAME: KEERTHANAT

REGISTER NUMBER: 185001074

CLASS-SEC: CSE-B

DATE: 12/08/2021

AIM :

To plot points that make up the line with endpoints (x_0, y_0) and (x_n, y_n) using Bresenham's line drawing algorithm for the following cases:

- (i) $|m| < 1$
- (ii) $|m| \geq 1$

ALGORITHM :

- Start
- Import GL library as a header file
- Create a function void myInit()
- Provide the default conditions
- Create a function void draw_pixel(int x, int y) where x,y are the point coordinates.
- Plot the points using GL_POINTS
- Create a function void draw_line()
- Calculate dx,dy
- Calculate e which the decision parameter
- If $dx > dy$, then $e = 2(dy) - dx$ and while $e \geq 0$ $y = y + 1$, $e = e + 2(dy - dx)$ else if $e < 0$ then $e = e + 2(dy)$ and $x = x + 1$. With the calculated values pass it to the draw_pixel(x,y) function.

- If $dy > dx$, then $e = 2(dx) - dy$ and while $e \geq 0$ $x = x + 1$, $e = e + 2(dx - dy)$ else if $e < 0$ then $e = e + 2(dx)$ and $y = y + 1$. With the calculated values pass it to the `draw_pixel(x,y)` function.
- Create a function `void myDisplay()`
- Pass the points to the function `draw_line(x1,x2,y1,y2)`
- Create the main function
- Get the input points from the user
- Give basic details for the output window in the main function
- Apply the Bresenham algorithm on the user input by calling the `myDisplay` function.
- End

CODE :

```
#include <gl/glut.h>
```

```
#include <stdio.h>
```

```
int x1, y1, x2, y2;
```

```
int arrx1[500], arry1[500], arrx2[500], arry2[500];
```

```
void myInit() {
    glClear(GL_COLOR_BUFFER_BIT);
    glClearColor(1.0, 1.0, 1.0, 0.0);
    glMatrixMode(GL_PROJECTION);
    gluOrtho2D(-200, 200, -200, 200);
}
```

```
void draw_pixel(int x, int y) {
    glBegin(GL_POINTS);
    glColor3ub(200, 100, 200);
    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() {
    for (int i = 0; i < 2; i++)
    {
        x1 = arrx1[i];
        x2 = arrx2[i];
        y1 = arry1[i];
        y2 = arry2[i];
        draw_line(x1, x2, y1, y2);
        glFlush();
    }
}

```

```

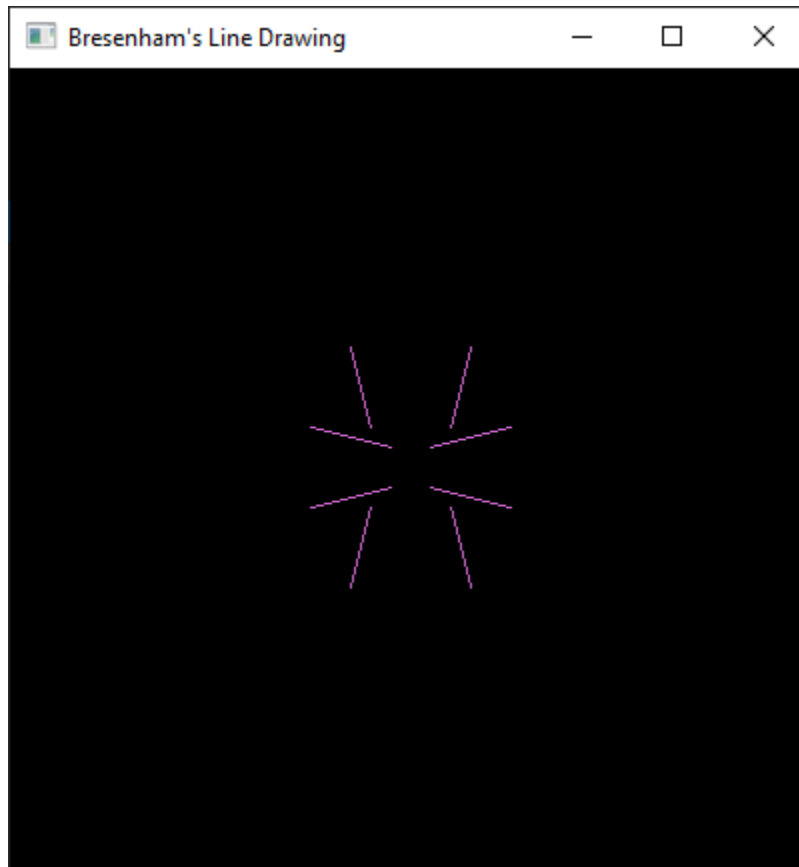
int main(int argc, char** argv) {
    for (int i = 0; i < 2; i++)
    {
        printf("Enter the points for line: %d\n", i + 1);
        scanf_s(" (%d,%d),(%d,%d)", &x1, &y1, &x2, &y2);
        arrx1[i] = x1;
        arrx2[i] = x2;
    }
}

```

```
    arry1[i] = y1;  
    arry2[i] = y2;  
}
```

```
glutInit(&argc, argv);  
glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);  
glutInitWindowSize(400, 400);  
glutInitWindowPosition(0, 0);  
glutCreateWindow("Bresenham's Line Drawing");  
myInit();  
glutDisplayFunc(myDisplay);  
glutMainLoop();  
  
return 0;  
}
```

OUTPUT:



RESULT:

Thus we have successfully implemented drawing lines using the Bresenham algorithm in OpenGL.