

CSO351: Computer Graphics

Lab Assignment 1.(b): Line Generation using Bresenham's Algorithm

Objective:

Write a program in C/C++ for implementation of line generation using Bresenham's algorithm.

Algorithm:

- **Step 1:** Get the input of two end points : (x1, y1) and (x2, y2).

- **Step 2:** Calculate the difference between two end points.

$$\begin{aligned}dx &= x2 - x1 \\ dy &= y2 - y1\end{aligned}$$

- **Step 3:** Calculate initial decision parameter:

$$p_0 = 2*dy - dx;$$

- **Step 4:** Calculate the increment in x and y coordinates.

```
if (x2 > x1)
{
    x = x1;
    y = y1;
}
else
{
    x = x2;
    y = y2;
}
```

- **Step 5:** If $dx > dy$ i.e. $|\text{slope}| < 1$ so compute $p_0 = 2*dy - dx$ else go to step 1.

- **Step 6:** Starting from $k = 0$ to dx do:

```

if ( $p_k < 0$ )
{
     $x_{k+1} = x_k + 1;$ 
     $y_{k+1} = y_k;$ 
     $p_{k+1} = p_k + 2*dy;$ 
}
else
{
     $x_{k+1} = x_k + 1;$ 
     $y_{k+1} = y_k + 1;$ 
     $p_{k+1} = p_k + 2*dy - 2*dx;$ 
}
plot( $x_k + 1, y_k + 1$ );

```

- **Step 7:** If $dx < dy$ i.e. $|\text{slope}| > 1$ so compute $p_0 = 2*dx - dy$ else go to step 2.

- **Step 8:** Starting from $k = 0$ to dy repeat:

```

if ( $p_k < 0$ )
{
     $x_{k+1} = x_k;$ 
     $y_{k+1} = y_k + 1;$ 
     $p_{k+1} = p_k + 2*dx;$ 
}
else
{
     $x_{k+1} = x_k + 1;$ 
     $y_{k+1} = y_k + 1;$ 
     $p_{k+1} = p_k + 2*dx - 2*dy;$ 
}
plot( $x_k + 1, y_k + 1$ );

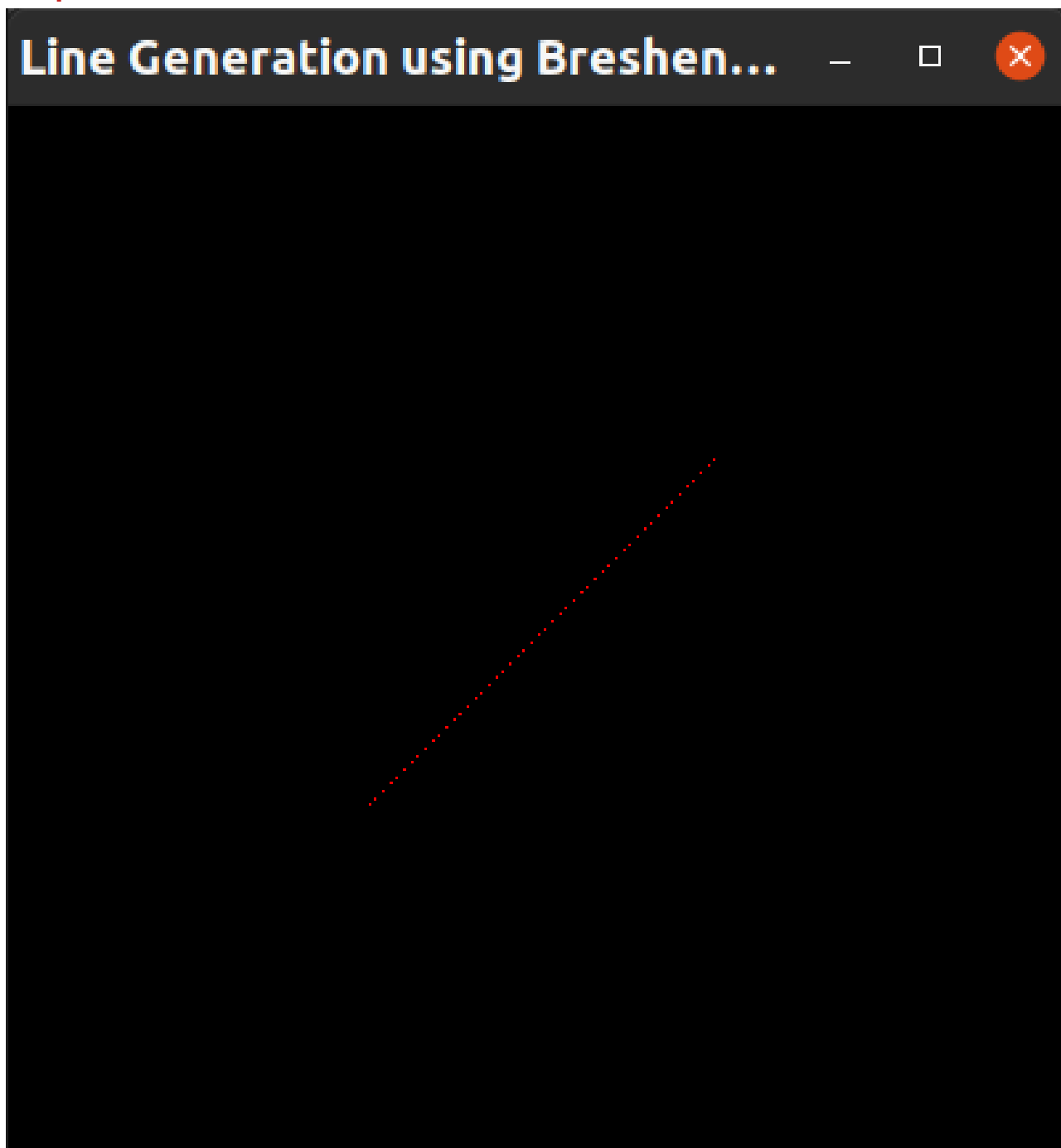
```

Result:

Input:

```
swaraj@shiv-raj-75:~/Documents/Assignments/Sem5/CG$ ./1.ii  
Enter x1 and y1 : 1 1  
Enter x2 and y2 : 50 70
```

Output:



Conclusion:

- In Bresenham's line generation algorithm, next pixel calculated is that one which has the least distance from true line, i.e. more accurate than DDA.
- It involves only integer arithmetic, i.e. easy and fast.
- It uses fixed points only.

Appendix: Code

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

int x1, x2, y1, y2;

void display (void)
{
    int dx = x2 - x1;
    int dy = y2 - y1;
    int p0 = 2*dy - dx;
    float x = x1, y = y1;
    glClear(GL_COLOR_BUFFER_BIT);
    glColor3f(1.0, 0.0, 0.0);
    glBegin(GL_POINTS);
    glVertex2i(x, y);
    int p = p0;
    int k;
    for(k = 0 ; k < dx ; k++)
    {
        if(p < 0)
        {
            x = x + 1;
            glVertex2i(x, y);
        }
        else
        {
            x = x + 1;
            y = y + 1;
            glVertex2i(x, y);
        }
    }
    glEnd();
    glFlush();
}

void init(void)
{
    glClearColor(0.0, 0.0, 0.0, 0.0);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluOrtho2D(-50, 100, -50, 100);
}
```

```
}
```

```
int main(int argc, char** argv)
{
    printf("Enter x1 and y1 : ");
    scanf("%d %d", &x1, &y1);
    printf("Enter x2 and y2 : ");
    scanf("%d %d", &x2, &y2);
    glutInit(&argc, argv);
    glutInitDisplayMode (GLUT_SINGLE | GLUT_RGB);
    glutInitWindowSize (400, 400);
    glutInitWindowPosition (10, 10);
    glutCreateWindow ("Line Generation using Bresenham's
Algorithm");
    init();
    glutDisplayFunc(display);
    glutMainLoop();
    return 0;
}
```