

Bangladesh University of Business and Technology



Lab no: 1

Course Name: Computer Graphics Lab

Course Code: CSE 342

Submitted By:

Name: Iftekhar Ahamed Siddiquee

ID : 19202103239

Intake : 44

Section: 6

Submitted To:

Ms. Sweety Lima

Lecturer

Department of CSE

BUBT

Submission Date : 17/01/2023

Lab No: 01

Lab Task Name: DDA Line generation

AlgorithmObjective:

DDA stands for Digital Differential Analyzer. It is an incremental method of scan conversion of line. In this method calculation is performed at each step but by using results of previous steps.

Suppose at step i, the pixels is (x_i, y_i)

The line of equation for step i

Algorithm:

```
    dx=x2-x1;
    dy=y2-y1;
    m=dy/dx;
if(m>0 && m<=1)
{
    while(x1<=x2 && y1<=y2)
    {
        x1=x1+1;
        y1=y1+m;
        glVertex3f(x1/100,y1/100,0.0);
        printf("%f %f",x1,y1);

    }
}
else if(m>1)
{
    while(x1<=x2 && y1<=y2)
    {
        x1=x1+(1/m);
        y1=y1+1;
```

```
        glVertex3f(x1/100,y1/100,0.0);
        printf("%f %f",x1,y1);
    }
}
```

```
else if(m>-1 && m<=0)
{
    while(x1>=x2 && y1>=y2)
    {
        x1=x1-1;
        y1=y1-m;
        glVertex3f(x1/100,y1/100,0.0);
        printf("%f %f",x1,y1);
    }
}
else if(m<-1)
```

```
{

    while(x1>=x2 && y1>=y2)
    {
        x1=x1-(1/m);
        y1=y1-1;
        glVertex3f(x1/100,y1/100,0.0);
        printf("%f %f",x1,y1);
    }
}
```

Program:

```
#include<stdio.h>
#include <GL/gl.h>
#include <GL/glut.h>
float x1,y1,x2,y2,m,i,j;
float dx,dy;
void display(void)
{
    glClear
(GL_COLOR_BUFFER_B
IT);

    glEnd();

    glColor3f (0.0, 1.0, 0.0);
    glBegin(GL_POINTS);

    if(m>0 && m<=1)
    {
        while(x1<=x2 &&
y1<=y2)
        {
            x1=x1+1;
            y1=y1+m;

            glVertex3f(x1/100,y1/100,
0.0);
            printf("%f
%f",x1,y1);

        }
    }
    else if(m>1)
    {
        while(x1<=x2 &&
y1<=y2)
        {
            x1=x1+(1/m);
            y1=y1+1;
```

```
glVertex3f(x1/100,y1/100,  
0.0);
```

```
    printf("%f  
%f",x1,y1);  
    }  
}
```

```
    else if(m>-1 && m<=0)  
    {  
        while(x1>=x2 &&  
y1>=y2)  
        {  
            x1=x1-1;  
            y1=y1-m;
```

```
glVertex3f(x1/100,y1/100,  
0.0);
```

```
    printf("%f  
%f",x1,y1);  
    }  
}
```

```
    else if(m<-1)
```

```
    {
```

```
        while(x1>=x2 &&  
y1>=y2)  
        {  
            x1=x1-(1/m);  
            y1=y1-1;
```

```
glVertex3f(x1/100,y1/100,  
0.0);
```

```
    printf("%f  
%f",x1,y1);  
    }  
}
```

```
glEnd();
```

```

    glFlush ();
}
void init (void)
{
    glClearColor (0.0, 0.0,
0.0, 0.0);

    glMatrixMode(GL_PROJE
CTION);
    glLoadIdentity();
    glOrtho(0.0, 1.0, 0.0,
1.0, -1.0, 1.0);
}
int main(int argc, char**
argv)
{

//glVertex3f(x1/100,y1/100
,0.0);write your code here
    printf("Enter value of X1
:");
    scanf("%f",&x1);
    printf("Enter value of y1
:");
    scanf("%f",&y1);
    printf("Enter value of X2
:");
    scanf("%f",&x2);
    printf("Enter value of Y2
:");
    scanf("%f",&y2);
    dx=x2-x1;
    dy=y2-y1;
    m=dy/dx;

    glutInit(&argc, argv);

```

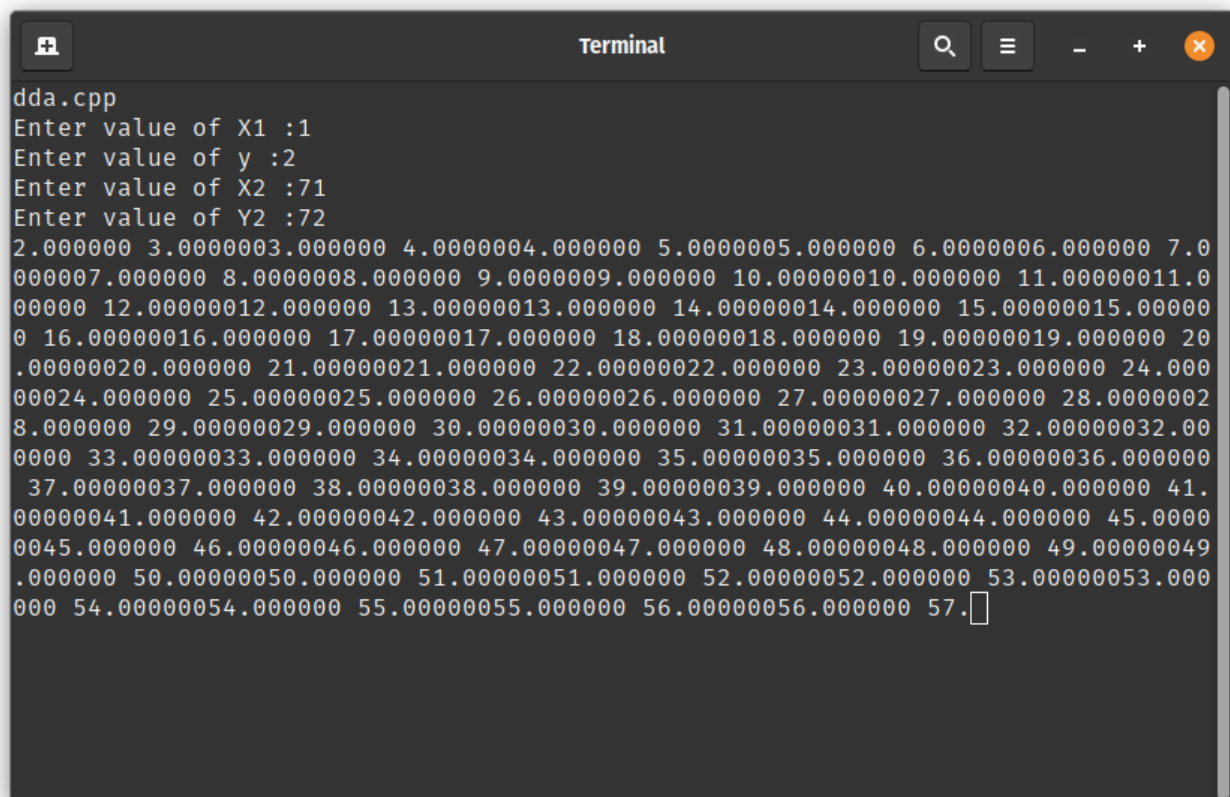
```

    glutInitDisplayMode
(GLUT_SINGLE |
GLUT_RGB);
    glutInitWindowSize
(500, 500);
    glutInitWindowPosition
(100, 100);
    glutCreateWindow
("hello");
    init ();

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

```

Input & Output:



```

Terminal
dda.cpp
Enter value of X1 :1
Enter value of y :2
Enter value of X2 :71
Enter value of Y2 :72
2.000000 3.000000 3.000000 4.000000 4.000000 5.000000 5.000000 6.000000 6.000000 7.0
000007.000000 8.000000 8.000000 9.000000 9.000000 10.000000 10.000000 11.000000 11.0
00000 12.000000 12.000000 13.000000 13.000000 14.000000 14.000000 15.000000 15.00000
0 16.000000 16.000000 17.000000 17.000000 18.000000 18.000000 19.000000 19.000000 20
.000000 20.000000 21.000000 21.000000 22.000000 22.000000 23.000000 23.000000 24.000
000 24.000000 25.000000 25.000000 26.000000 26.000000 27.000000 27.000000 28.000000 2
8.000000 29.000000 29.000000 30.000000 30.000000 31.000000 31.000000 32.000000 32.00
0000 33.000000 33.000000 34.000000 34.000000 35.000000 35.000000 36.000000 36.000000
37.000000 37.000000 38.000000 38.000000 39.000000 39.000000 40.000000 40.000000 41.
000000 41.000000 42.000000 42.000000 43.000000 43.000000 44.000000 44.000000 45.0000
00 45.000000 46.000000 46.000000 47.000000 47.000000 48.000000 48.000000 49.000000 49
.000000 50.000000 50.000000 51.000000 51.000000 52.000000 52.000000 53.000000 53.000
000 54.000000 54.000000 55.000000 55.000000 56.000000 56.000000 57.

```

