

DDA ALGORITHM**AIM**

Write a program to draw a line using the DDA algorithm.

PROGRAM

```
import OpenGL

from OpenGL.GL import *
from OpenGL.GLU import *
from OpenGL.GLUT import *


import sys


def init(): # Clear screen and set origin
    glClearColor(0.0,0.0,0.0,1.0)
    gluOrtho2D(0,100,0,100)


def display_menu(): # Function to display menu
    print("-----MENU-----")
    print(f"1. Plot line")
    print(f"0. Exit")
    return int(input("Enter Choice: "))


def get_input(): # Function to get input from user
    x1, y1 = map(int, input("Enter initial coordinate seperated by space: (Eg. '0 0')").split(" "))
    x2, y2 = map(int, input("Enter final coordinate seperated by space: (Eg. '0 0')").split(" "))
    return x1, y1, x2, y2


def plot_line(x1, y1, x2, y2): # Function to plot line using DDA
    # Find deltaX and deltaY
    deltaX = abs(x2 - x1)
    deltaY = abs(y2 - y1)
```

```

if deltaX > deltaY:
    steps = deltaX
else:
    steps = deltaY
# Set the value to increment by
x_increment = deltaX/steps
y_increment = deltaY/steps

glClear(GL_COLOR_BUFFER_BIT)
glColor3f(0.0,1.0,0.0)
glPointSize(10.0)
glBegin(GL_POINTS)

for step in range(1, steps+1):
    # Round the values and plot the points
    glVertex2f(round(x1), round(y1))
    # Increment the points
    x1 += x_increment
    y1 += y_increment

glEnd()
glFlush()

```

```

def display_window(x1, y1, x2, y2): # Function to display window
    print("Creating Window...")
    glutInit(sys.argv)
    glutInitDisplayMode(GLUT_RGB)
    glutInitWindowSize(500,500)
    glutInitWindowPosition(50, 50)
    glutCreateWindow("Plot Line using DDA Algorithm")
    glutDisplayFunc(lambda: plot_line(x1,y1,x2,y2))
    # glutIdleFunc(lambda: plot_line(x1,y1,x2,y2))

```

```
init()
glutMainLoop()
```

```
def main():
    choice = 1
    while choice != 0:
        choice = display_menu()
        if choice == 1:
            # Checks if it's a valid input (i.e. present in dictionary)
            x1, y1, x2, y2 = get_input()
            display_window(x1, y1, x2, y2)
        elif choice == 0:
            # To handle exit from program
            print("Exiting Program...")
        else:
            # To handle invalid choice
            print("Invalid Choice! Try again.")

main()
```

ALGORITHM

- Input the initial and final coordinates and store it in x1, y1, x2 and y2 respectively.
- If $\text{abs}(x2-x1) > \text{abs}(y2-y1)$ then $\text{length} = \text{abs}(x2-x1)$, else $\text{length} = \text{abs}(y2-y1)$
- $\text{deltaX} = (x2-x1)/\text{length}$ and $\text{deltaY} = (y2-y1)/\text{length}$
- Now in a loop from 0 to lengthm plot (x,y) where $x = x + \text{deltaX}$ (x initialised from x1) & $y = y + \text{deltaY}$ where y is initialised from y1.

RESULT

Program to draw a line using DDA Algorithm was created and executed successfully.

INPUT/OUTPUT

```
(.venv) E:\College\S5\Computer Graphics\Experiment 1>python dda.py
-----MENU-----
1. Plot line
0. Exit
Enter Choice: 1
Enter initial coordinate seperated by space: (Eg. '0 0')0 0
Enter final coordinate seperated by space: (Eg. '0 0')15 13
Creating Window...
```

