EXPERIMENT NO: 2 DATE:08/09/2021

C) LINE DRAWING ALGORITHM: MIDPOINT LINE DRAWING ALGORITHM

AIM

Write a program to draw a line using Mid-Point Line Drawing algorithm.

ALGORITHM

```
• Input initial and final coordinates from the user and store it in x1, y1, x2 and y2
  respectively.
• Set x and y as x1 and y1 respectively. Set deltaX = x2 – x1 and deltaY = y2 – y1
• If deltaX > deltaY
    o x = x1 and y = y1
    o Set p = deltaY – deltaX
    o Loop till x is less than x2
        ▪ Increment x by 1
        ▪ If p < 0 add deltaY to p
        ▪ else increment y by 1 and add (deltaY – deltaX) to p
        ▪ Plot the point (x, y)
• Else
    o x = x1 and y = y1
    o Set p = deltaX – deltaY
    o Loop till y is less than y2
        ▪ Increment y by 1
        ▪ If p < 0 add deltaX to p
        ▪ else increment x by 1 and add (deltaX – deltaY) to p
        ▪ Plot the point (x, y)
```

PROGRAM

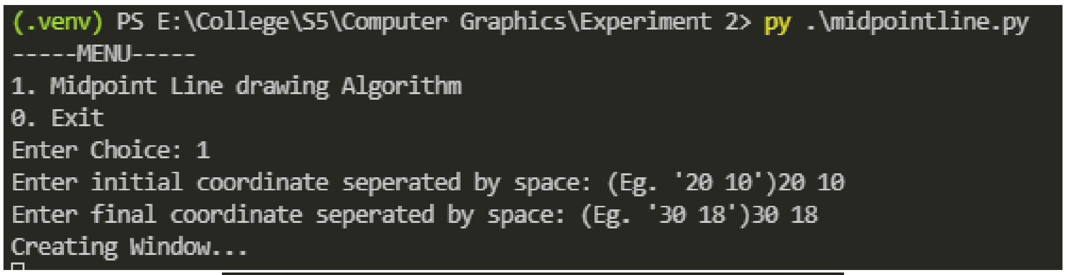
```
# Importing dependencies

import OpenGL # Standard interface for displaying

from OpenGL.GL import *

from OpenGL.GLU import *

from OpenGL.GLUT import *


import sys

import math


WINDOW_POSITION = 50
```

```
POINT_SIZE = 10


def init():                                             # Clear screen and set origin
    glClearColor(0.0, 0.0, 0.0, 1.0)                           # Set Background Color
    gluOrtho2D(0, WINDOW_POSITION, 0, WINDOW_POSITION)                 # Set the
Range of coordinate system (x1, x2, y1, y2)


def display_menu():
    # Function to display menu
    print("-----MENU-----")
    print(f"1. Midpoint Line drawing Algorithm")
    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. '20 10')").split("
"))
    x2, y2 = map(int, input("Enter final coordinate seperated by space: (Eg. '30 18')").split("
"))
    return x1, y1, x2, y2


def create_points(a, b, a2, b2, deltaA, deltaB, deltaY_greater):
    # Function to create points based on value of deltaX and deltaY
    points = []
    if deltaY_greater:
        points.append((b, a))
    else:
        points.append((a, b))
    p = deltaB - deltaA
    while a < a2:
        a += 1
        if p < 0:
            p += deltaB
        else:
            b += 1
            p += deltaB - deltaA
        if deltaY_greater:
            points.append((b, a))
        else:
            points.append((a, b))
    return points

def get_points(x1, y1, x2, y2):
    # Function to return points to plot
    # Points calculated using Midpoint Line Algorithm
    points = []

    deltaX = x2 - x1
    deltaY = y2 - y1

    if deltaX > deltaY:
        points = create_points(x1, y1, x2, y2, deltaX, deltaY, False)
    else:
        points = create_points(y1, x1, y2, x2, deltaY, deltaX, True)
    return points

def plot_line(x1, y1, x2, y2):
    # Function to the requeired plot line
    # Get points to plot
    points = get_points(x1, y1, x2, y2)
    glClear(GL_COLOR_BUFFER_BIT)
    glColor3f(1.0,0.0,0.0)
    glPointSize(POINT_SIZE)
    glBegin(GL_POINTS)

    # Plot the points
    for x, y in points:
        glVertex2f(x, y)

    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 Midpoint line drawing Algorithm")
    glutDisplayFunc(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()
```

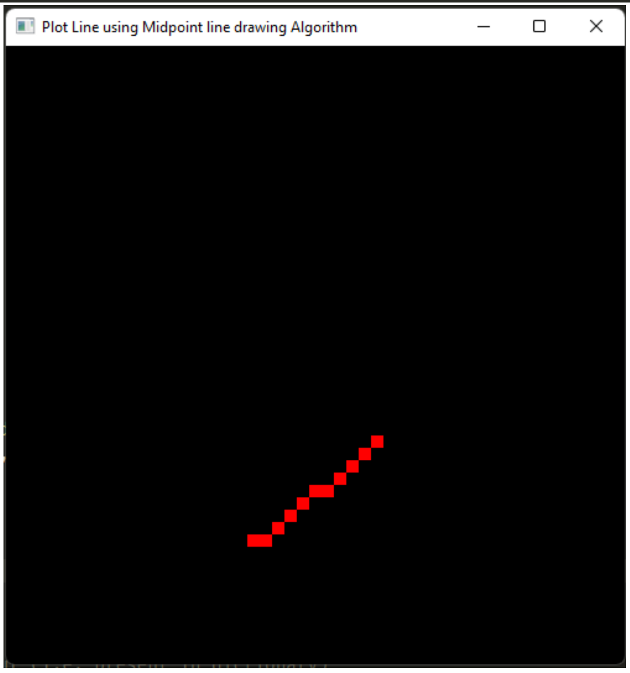
RESULT

Program to draw a line using Midpoint Line Drawing Algorithm was created and executed successfully.

OUTPUT/INPUT



```
(.venv) PS E:\College\S5\Computer Graphics\Experiment 2> py .\midpointline.py
-----MENU-----
1. Midpoint Line drawing Algorithm
0. Exit
Enter Choice: 1
Enter initial coordinate seperated by space: (Eg. '20 10')20 10
Enter final coordinate seperated by space: (Eg. '30 18')30 18
Creating Window...
```


Plot Line using Midpoint line drawing Algorithm