EXPERIMENT NO: 2 DATE:08/09/2021

A) LINE DRAWING ALGORITHM: BRESENHAM’S ALGORITHM

AIM

Write a program to draw a line using Bresenham's 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 and p = 2*deltaY - deltaX.
- Display this point (x, y)
- Loop till x < x2
    - Increment x by 1
    - If p < 0 add 2*deltaY to p
    - Else increment y and add (2*deltaY – 2*deltaX) to p
    - Plot the point (x, y)

PROGRAM

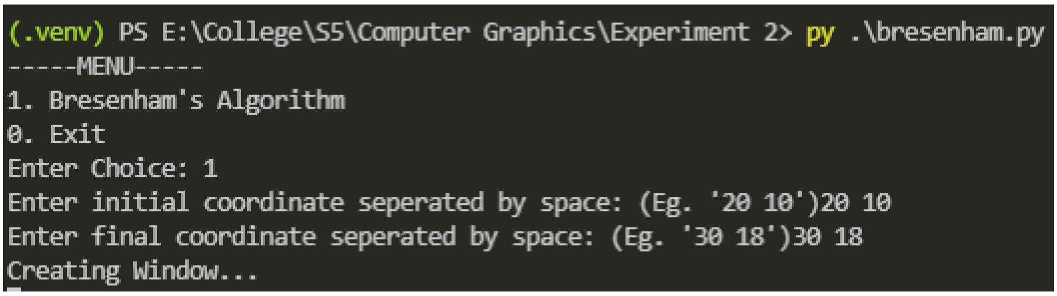
```
# Importing dependencies
import OpenGL
from OpenGL.GL import *
from OpenGL.GLU import *
from OpenGL.GLUT import *


import sys
import math


# Constants to set window size and size of points
WINDOW_POSITION = 100
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. Bresenham's 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 get_points(x1, y1, x2, y2):
    # Function to return points to plot
    # Points calculated using Bresenham's Algorithm
    points = []

    x, y = x1, y1

    deltaX = x2 - x1
    deltaY = y2 - y1

    points.append((x, y))

    p = 2*deltaY - deltaX

    while x < x2:
        x += 1
        if p < 0:
            p += 2*deltaY
        else:
            y += 1
            p += 2*deltaY - 2*deltaX
        points.append((x, y))
    return points



def plot_line(x1, y1, x2, y2):
    # Function to plot the points
    # 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 Bresenham's 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()
```

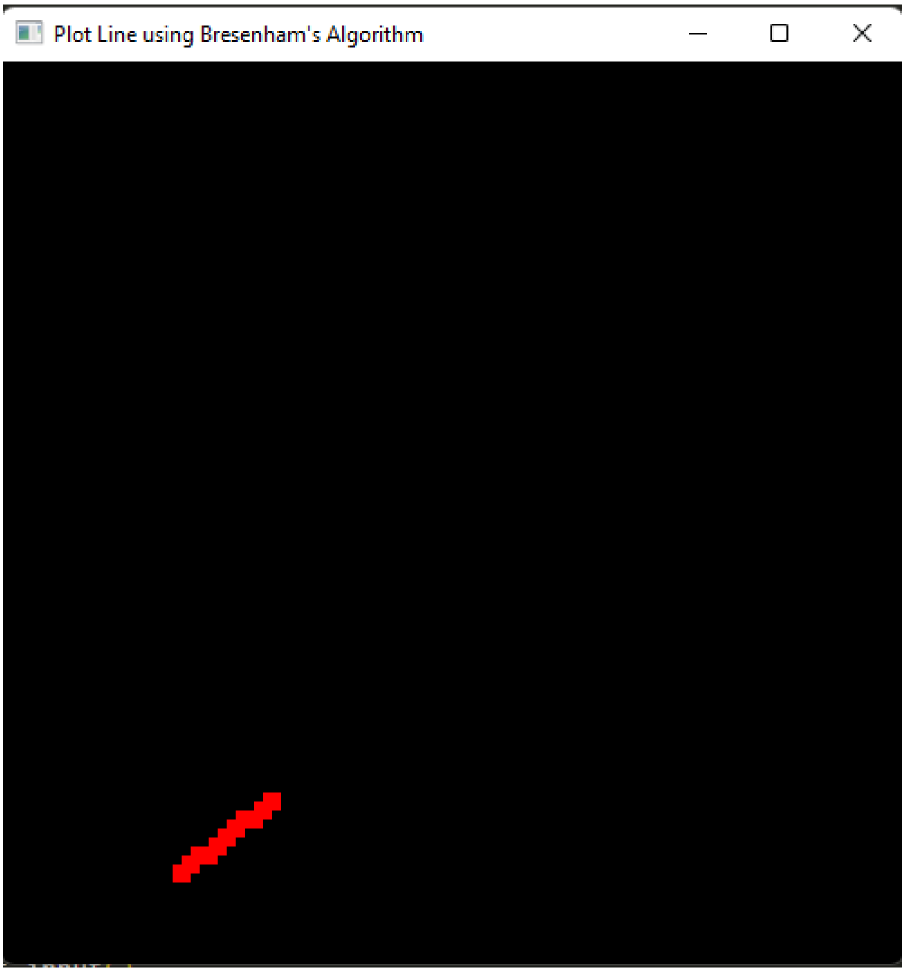
RESULT

Program to draw a line using Bresenham's Algorithm was created and executed successfully.

OUTPUT/INPUT



```
(.venv) PS E:\College\S5\Computer Graphics\Experiment 2> py .\bresenham.py
-----MENU-----
1. Bresenham's 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 Bresenham's Algorithm