Kathmandu University

Department of Computer Science and Engineering

Dhulikhel, Kavre

Computer Graphics Lab Report 02

on

‘Line Drawing Algorithms - Lab 02 Task’

Submitted By:

Reewaj Khanal (61)

Submitted to:

Mr. Dhiraj Shrestha

Assistant Professor

Department of Computer Science and Engineering

School of Engineering

Kathmandu University

Dhulikhel, Kavre

Submission Date: Thursday 16 May 2024

Question No. 1 Implement Digital Differential Analyzer Line drawing algorithm.

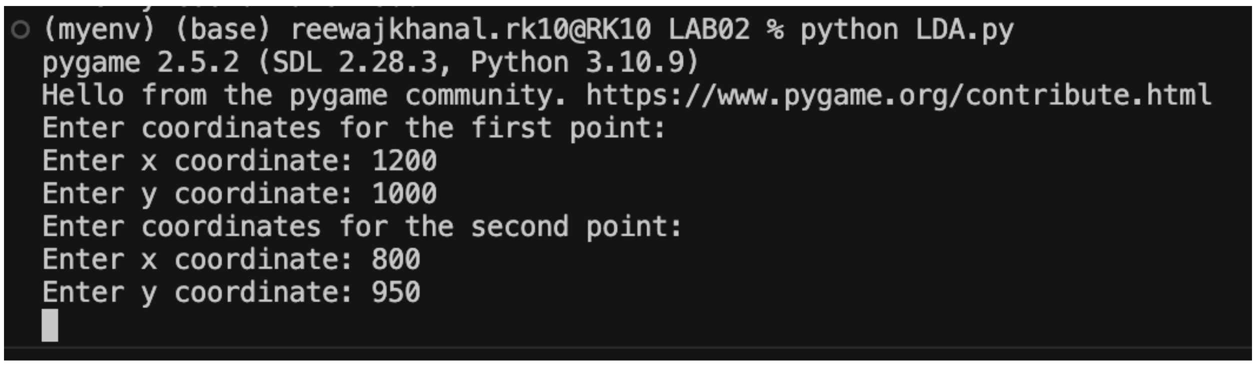
Answer:

```python
import pygame

from pygame.locals import *

from OpenGL.GL import *


def DDA_Line(x1, y1, x2, y2):

    dx = x2 - x1

    dy = y2 - y1


    steps = max(abs(dx), abs(dy))


    x_increment = dx / steps

    y_increment = dy / steps


    x = x1

    y = y1


    glBegin(GL_POINTS)

    glVertex2f(x, y)

    glEnd()


    for _ in range(steps):

        x += x_increment

        y += y_increment

        glBegin(GL_POINTS)

        glVertex2f(round(x), round(y))

        glEnd()
```

```python
def get_point():
   x = int(input("Enter x coordinate: "))
   y = int(input("Enter y coordinate: "))
   return x, y

def main():
   # Prompt user for coordinates of two points
   print("Enter coordinates for the first point:")
   point1 = get_point()
   print("Enter coordinates for the second point:")
   point2 = get_point()

   # Determine screen dimensions based on input points
   max_x = max(point1[0], point2[0])
   max_y = max(point1[1], point2[1])
   screen_width = max_x + 100  # Add padding
   screen_height = max_y + 100  # Add padding

   pygame.init()
   display = (screen_width, screen_height)
   pygame.display.set_mode(display, DOUBLEBUF | OPENGL)

   glOrtho(0, screen_width, 0, screen_height, -1, 1)

   while True:
       for event in pygame.event.get():
           if event.type == pygame.QUIT:
               pygame.quit()
               quit()
```

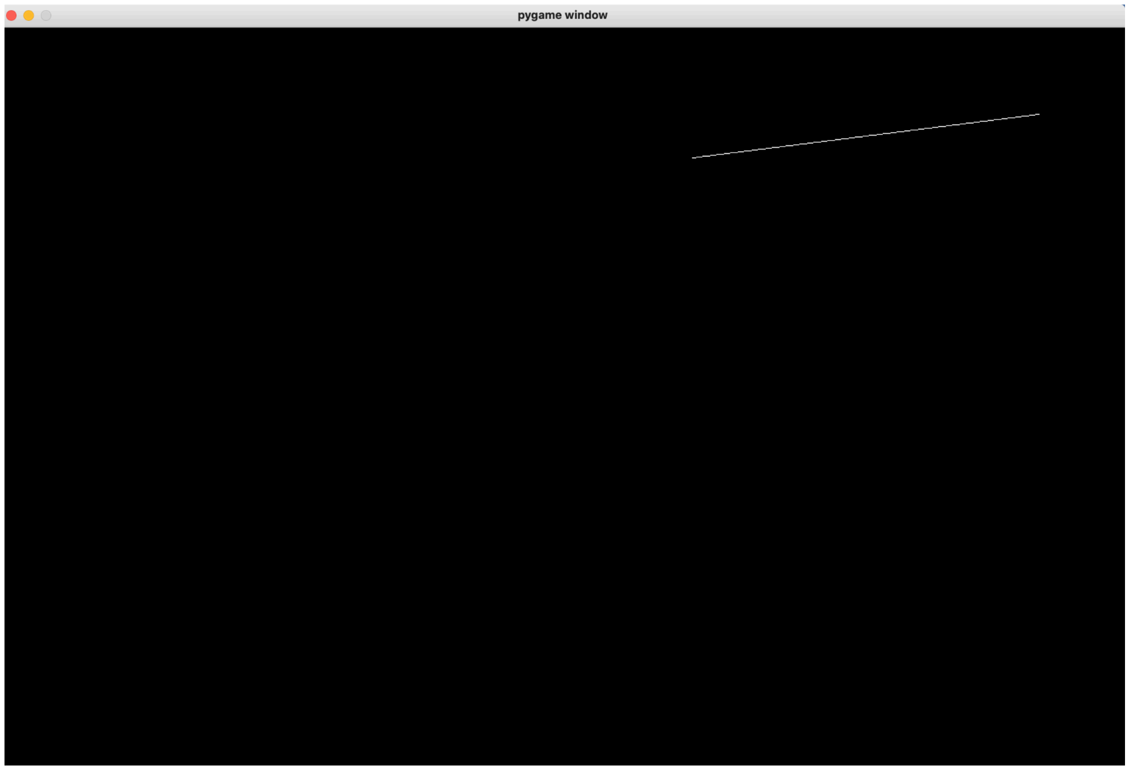
```python
        glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)


        glColor3f(1, 1, 1)  # Set line color to white


        # Drawing the line using DDA algorithm
        DDA_Line(*point1, *point2)


        pygame.display.flip()


if __name__ == "__main__":
    main()
```

Input:



```
(myenv) (base) reewajkhanal.rk10@RK10 LAB02 % python LDA.py
pygame 2.5.2 (SDL 2.28.3, Python 3.10.9)
Hello from the pygame community. https://www.pygame.org/contribute.html
Enter coordinates for the first point:
Enter x coordinate: 1200
Enter y coordinate: 1000
Enter coordinates for the second point:
Enter x coordinate: 800
Enter y coordinate: 950
```

Output Generated:



Question No. 2 Implement Bresenham Line Drawing algorithm for both slopes(|m|<1 and |m|>=1).

Answer:

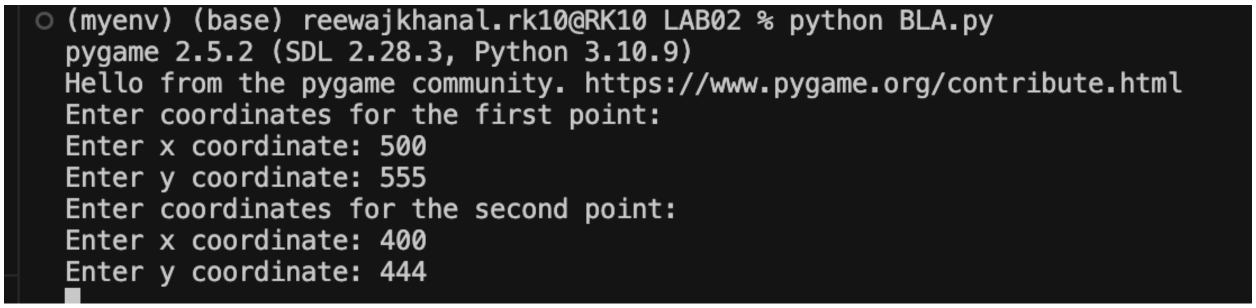
```
import pygame

from pygame.locals import *

from OpenGL.GL import *


def Bresenham_Line(x1, y1, x2, y2):

   dx = abs(x2 - x1)

   dy = abs(y2 - y1)
```

```
slope_error = dx - dy


x, y = x1, y1


x_increment = 1 if x2 > x1 else -1

y_increment = 1 if y2 > y1 else -1


glBegin(GL_POINTS)

glVertex2f(x, y)


if dx > dy:  # |m| < 1

    slope_double_error = slope_error * 2

    while x != x2:

        x += x_increment

        if slope_error >= 0:

            y += y_increment

            slope_error -= slope_double_error

        slope_error += dx * 2

        glVertex2f(x, y)

else:  # |m| >= 1

    slope_double_error = slope_error * 2

    while y != y2:

        y += y_increment

        if slope_error >= 0:

            x += x_increment

            slope_error -= slope_double_error

        slope_error += dy * 2

        glVertex2f(x, y)


glEnd()
```

```python
def get_point():
   x = int(input("Enter x coordinate: "))
   y = int(input("Enter y coordinate: "))
   return x, y

def main():
   # Prompt user for coordinates of two points
   print("Enter coordinates for the first point:")
   point1 = get_point()
   print("Enter coordinates for the second point:")
   point2 = get_point()

   # Determine screen dimensions based on input points
   max_x = max(point1[0], point2[0])
   max_y = max(point1[1], point2[1])
   screen_width = max_x + 100  # Add padding
   screen_height = max_y + 100  # Add padding

   pygame.init()
   display = (screen_width, screen_height)
   pygame.display.set_mode(display, DOUBLEBUF | OPENGL)

   glOrtho(0, screen_width, 0, screen_height, -1, 1)

   while True:
       for event in pygame.event.get():
           if event.type == pygame.QUIT:
               pygame.quit()
               quit()
```

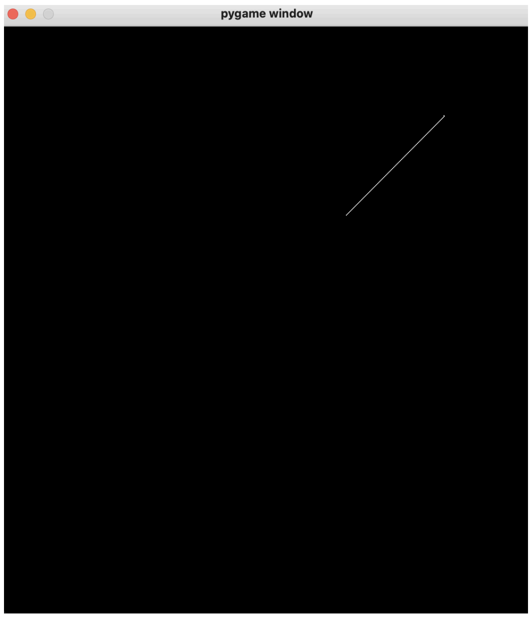
```
        glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)

        glColor3f(1, 1, 1)  # Set line color to white

        # Drawing the line using Bresenham algorithm
        Bresenham_Line(*point1, *point2)

        pygame.display.flip()

if __name__ == "__main__":
    main()
```

Input:



```
(myenv) (base) reewajkhanal.rk10@RK10 LAB02 % python BLA.py
pygame 2.5.2 (SDL 2.28.3, Python 3.10.9)
Hello from the pygame community. https://www.pygame.org/contribute.html
Enter coordinates for the first point:
Enter x coordinate: 500
Enter y coordinate: 555
Enter coordinates for the second point:
Enter x coordinate: 400
Enter y coordinate: 444
```

Output:



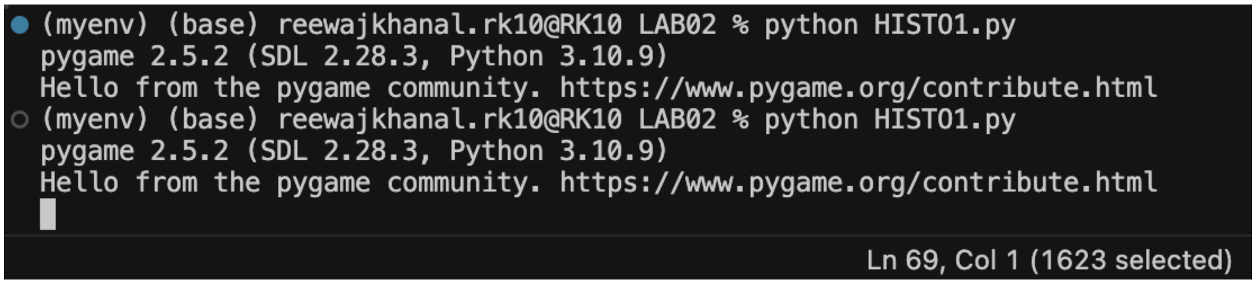
Question No. 3 Implement the given line drawing algorithm to draw a line histogram for any given frequency inputs.

Answer:

```
import pygame

from pygame.locals import *

from OpenGL.GL import *

import random  # Add this line to import the random module


X_start = 400

Y_start = 600

cost = 40

width = 60




class Histogram:

   def __init__(self, frequencies):

       self.frequencies = frequencies

       pygame.init()

       self.screen = pygame.display.set_mode((1280, 720), DOUBLEBUF|OPENGL)

       self.clock = pygame.time.Clock()

       self.show_screen()


   def show_screen(self):

       glClearColor(1, 1, 1, 1)

       glOrtho(0, 1280, 720, 0, -1, 1)


       color = [0]*(len(self.frequencies))

       for i in range(len(self.frequencies)):
           color[i] = (random.random(), random.random(), random.random())


       while True:

           for event in pygame.event.get():

               if event.type == pygame.QUIT:

                   pygame.quit()

                   quit()


           glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)


           draw_axis(X_start, Y_start, X_max=1000, Y_max=100)


           for i, frequency in enumerate(self.frequencies):

               x = X_start + i * width

               draw_bar(x, Y_start, width, frequency * cost, color=color[i])


           pygame.display.flip()

           self.clock.tick(60)



def draw_bar(x, y, width, height, color):

   glColor3f(*color)


   glBegin(GL_QUADS)

   glVertex2f(x, y)

   glVertex2f(x + width, y)

   glVertex2f(x + width, y - height)

   glVertex2f(x, y - height)

   glEnd()
```

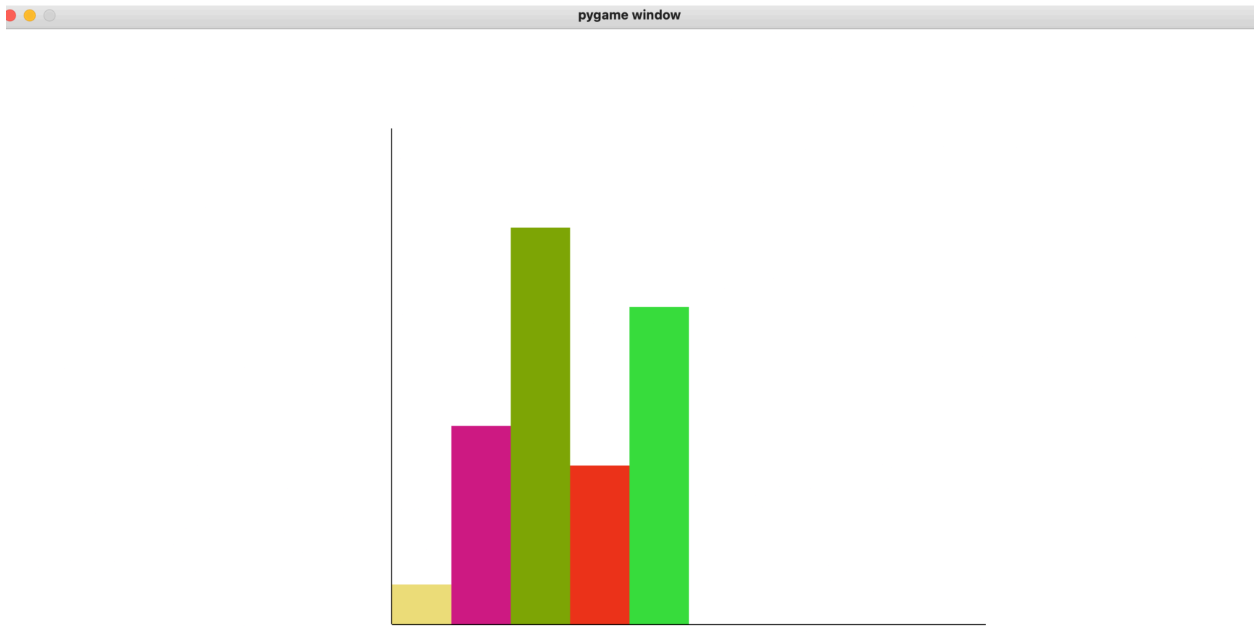
```python
def draw_axis(x, y, X_max, Y_max):
    glColor3f(0, 0, 0)

    glBegin(GL_LINES)
    glVertex2f(x, y+1)
    glVertex2f(X_max, y+1)
    glEnd()

    glBegin(GL_LINES)
    glVertex2f(x, y)
    glVertex2f(x, Y_max)
    glEnd()


if __name__ == "__main__":
    freq = [1, 5, 10, 4, 8]
    hist = Histogram(freq)
```

Input:



```
● (myenv) (base) reewajkhanal.rk10@RK10 LAB02 % python HIST01.py
  pygame 2.5.2 (SDL 2.28.3, Python 3.10.9)
  Hello from the pygame community. https://www.pygame.org/contribute.html
○ (myenv) (base) reewajkhanal.rk10@RK10 LAB02 % python HIST01.py
  pygame 2.5.2 (SDL 2.28.3, Python 3.10.9)
  Hello from the pygame community. https://www.pygame.org/contribute.html
  █
                                                Ln 69, Col 1 (1623 selected)
```

Output:

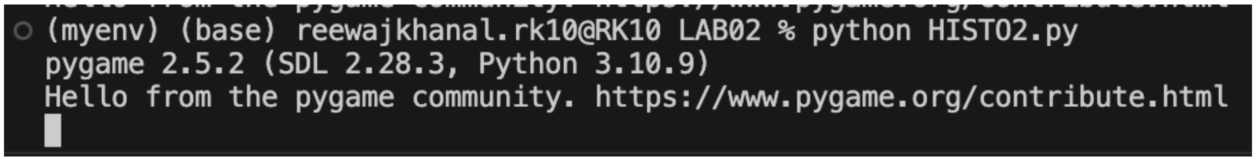


Answer [From BLA Approach]:

```
import pygame

from pygame.locals import *

from OpenGL.GL import *

import random  # Import the random module


X_start = 400

Y_start = 600

cost = 40

width = 60



class Histogram:

   def __init__(self, frequencies):
       self.frequencies = frequencies

       pygame.init()

       self.screen = pygame.display.set_mode((1280, 720), DOUBLEBUF|OPENGL)

       self.clock = pygame.time.Clock()

       self.show_screen()


   def show_screen(self):

       glClearColor(1, 1, 1, 1)

       glOrtho(0, 1280, 720, 0, -1, 1)


       color = [0]*(len(self.frequencies))

       for i in range(len(self.frequencies)):

           color[i] = (random.random(), random.random(), random.random())


       while True:

           for event in pygame.event.get():

               if event.type == pygame.QUIT:

                   pygame.quit()

                   quit()


           glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)


           draw_axis(X_start, Y_start, X_max=1000, Y_max=100)


           for i, frequency in enumerate(self.frequencies):

               x = X_start + i * width

               draw_bar(x, Y_start, width, frequency * cost, color=color[i])


           pygame.display.flip()

           self.clock.tick(60)
```

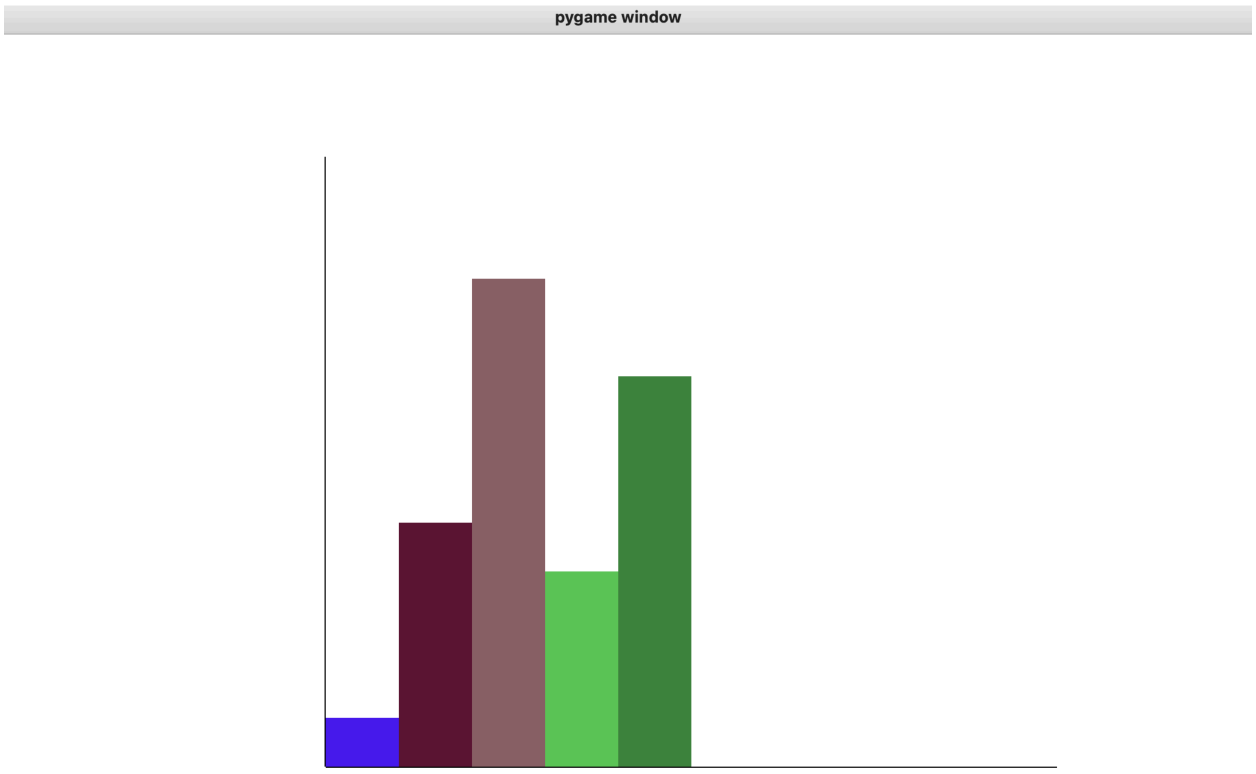
```
def draw_bar(x, y, width, height, color):
   glColor3f(*color)

   glBegin(GL_QUADS)
   glVertex2f(x, y)
   glVertex2f(x + width, y)
   glVertex2f(x + width, y - height)
   glVertex2f(x, y - height)
   glEnd()


def draw_axis(x, y, X_max, Y_max):
   glColor3f(0, 0, 0)

   glBegin(GL_LINES)
   glVertex2f(x, y+1)
   glVertex2f(X_max, y+1)
   glEnd()

   glBegin(GL_LINES)
   glVertex2f(x, y)
   glVertex2f(x, Y_max)
   glEnd()


if __name__ == "__main__":
   freq = [1, 5, 10, 4, 8]
   hist = Histogram(freq)
```

Input:



```
(myenv) (base) reewajkhanal.rk10@RK10 LAB02 % python HIST02.py
pygame 2.5.2 (SDL 2.28.3, Python 3.10.9)
Hello from the pygame community. https://www.pygame.org/contribute.html
```

Output:


pygame window