Kathmandu University

Department of Computer Science and Engineering

Dhulikhel, Kavre

Computer Graphics Lab Report 03

on

‘Circle and Ellipse Drawing Algorithm” - Lab 03 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

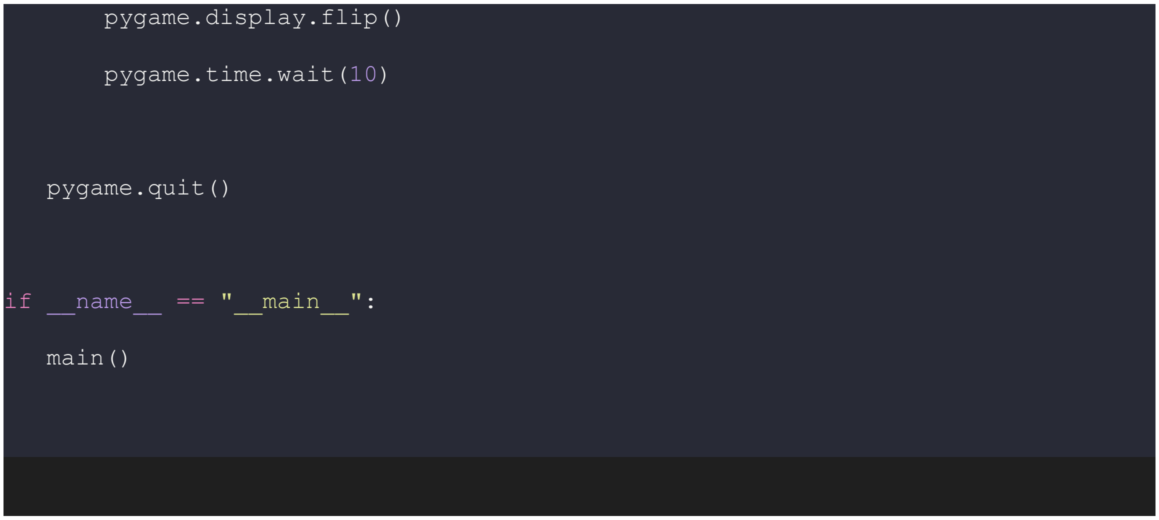
Submission Date: Friday 31 May 2024

Question No. 1 Write a Program to implement mid- point Circle Drawing Algorithm

Answer:

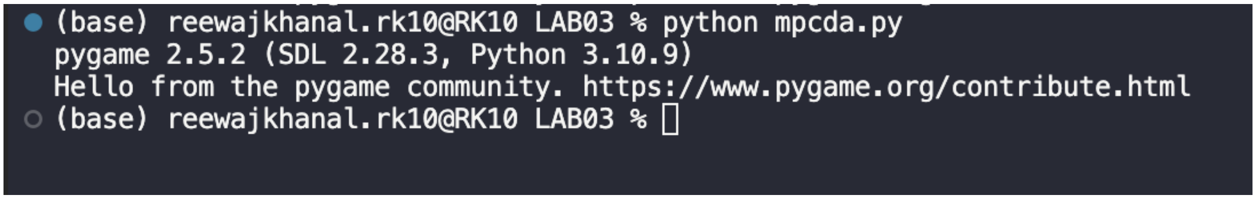
```
import pygame
from pygame.locals import *
from OpenGL.GL import *
from OpenGL.GLUT import *
from OpenGL.GLU import *

# Function to plot points in all octants
def plot_circle_points(x_center, y_center, x, y):
   glBegin(GL_POINTS)
   # Reflecting the points in all octants
   # Octant 1: Right Top Up
   glVertex2i(x_center + y, y_center + x)
   # Octant 2: Right Top
   glVertex2i(x_center + x, y_center + y)
   # Octant 3: Right Bottom Up
   glVertex2i(x_center + y, y_center - x)
   # Octant 4: Right Bottom
   glVertex2i(x_center + x, y_center - y)
   # Octant 5: Left Bottom
   glVertex2i(x_center - x, y_center - y)
   # Octant 6: Left Bottom Up
   glVertex2i(x_center - y, y_center - x)
   # Octant 7: Left Top Up
   glVertex2i(x_center - y, y_center + x)
   # Octant 8: Left Top
   glVertex2i(x_center - x, y_center + y)
```

```
    glEnd()


# Mid-Point Circle Drawing Algorithm

def midpoint_circle(x_center, y_center, radius):

    x = 0

    y = radius

    d = 1 - radius  # Decision parameter

    plot_circle_points(x_center, y_center, x, y)


    while x < y:

        if d < 0:

            # Move to the right

            d = d + 2 * x + 3

        else:

            # Move to the right and down

            d = d + 2 * (x - y) + 5

            y -= 1

        x += 1

        plot_circle_points(x_center, y_center, x, y)


def draw_circle(x_center,y_center,radius):


    glClear(GL_COLOR_BUFFER_BIT)

    glColor3f(1.0, 1.0, 1.0)  # Set color to white

    # Draw the axes

    draw_axes()

     midpoint_circle(x_center, y_center, radius)  # Draw circle with radius 100 at origin

    glFlush()


def draw_axes():
    glBegin(GL_LINES)

    # Draw X axis

    glVertex2i(-400, 0)

    glVertex2i(400, 0)

    # Draw Y axis

    glVertex2i(0, -300)

    glVertex2i(0, 300)

    glEnd()



def get_input():

    x_center=int(input("Enter x coordinate of origin"))

    y_center=int(input("Enter x coordinate of origin"))

    radius=int(input("Enter radius of circle"))

    return x_center,y_center,radius



def main():

    pygame.init()

    display = (800, 600)

    pygame.display.set_mode(display, DOUBLEBUF | OPENGL)

    gluOrtho2D(-400, 400, -300, 300)  # Set up 2D coordinate system

    # x_center,y_center,radius=get_input()

    running = True

    while running:

        for event in pygame.event.get():

            if event.type == pygame.QUIT:

                running = False


        draw_circle(x_center=50,y_center=50,radius=100)
```



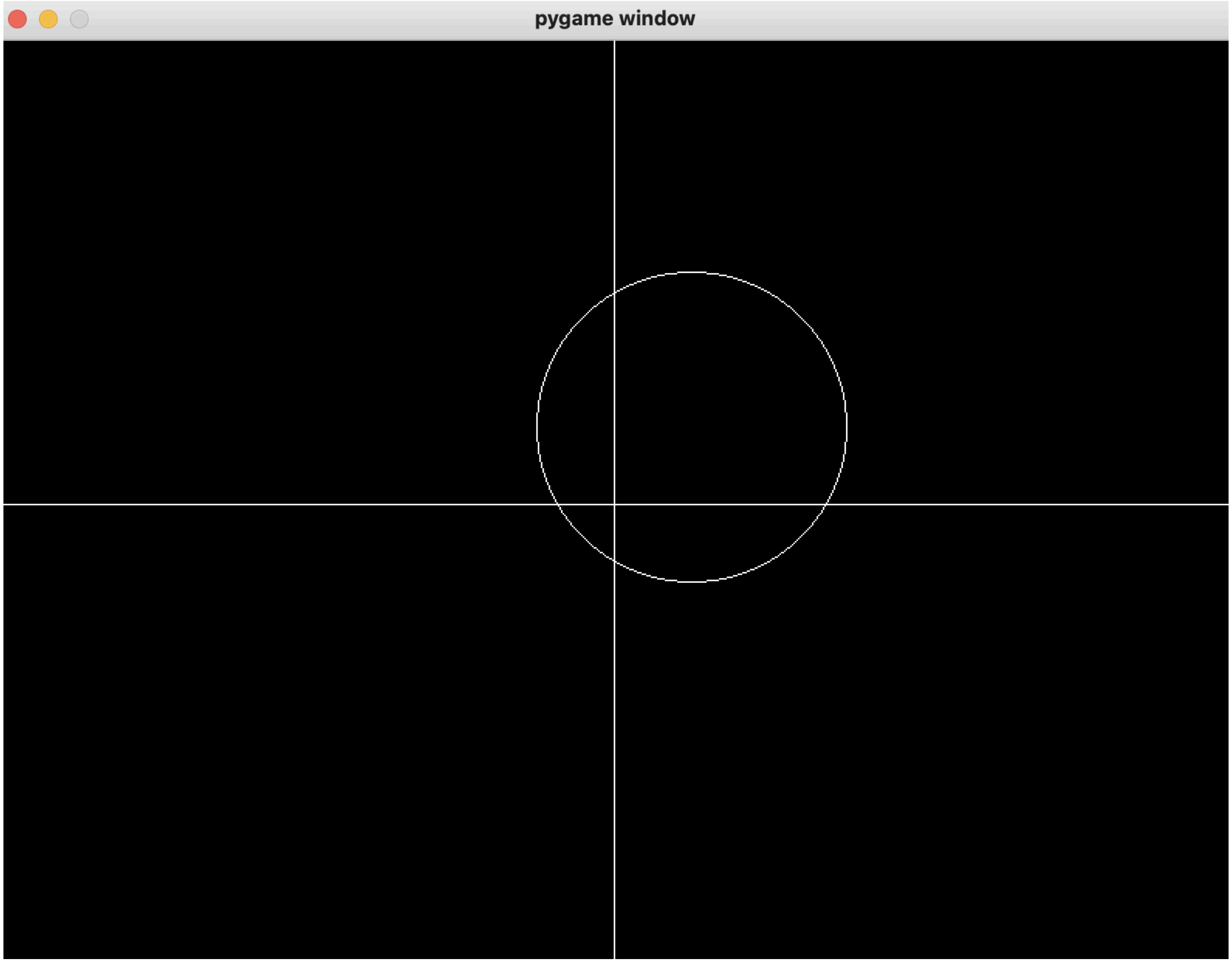
```
        pygame.display.flip()

        pygame.time.wait(10)


    pygame.quit()


if __name__ == "__main__":

    main()
```

Input:



```
● (base) reewajkhanal.rk10@RK10 LAB03 % python mpcda.py
  pygame 2.5.2 (SDL 2.28.3, Python 3.10.9)
  Hello from the pygame community. https://www.pygame.org/contribute.html
○ (base) reewajkhanal.rk10@RK10 LAB03 % ▯
```

Output Generated:

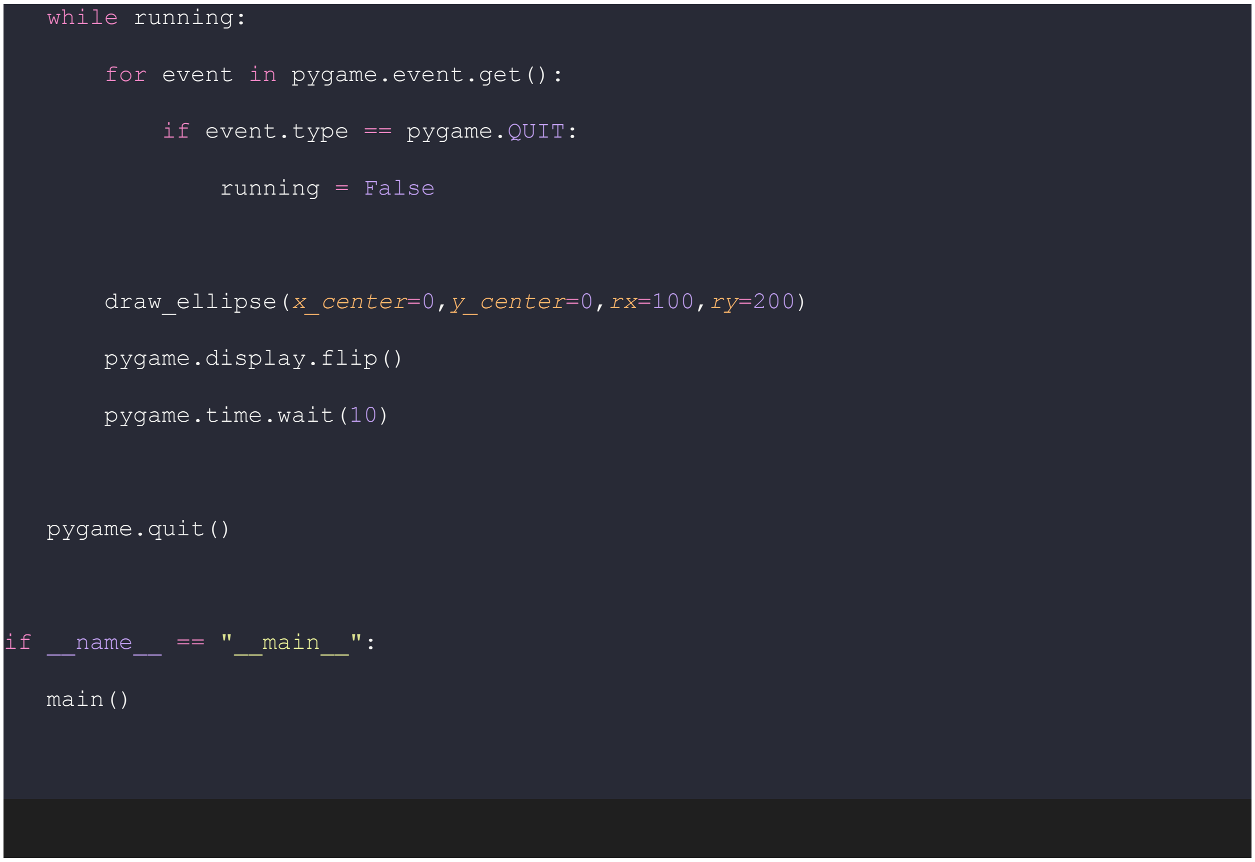


Question No. 2 Write a Program to implement mid- point Ellipse Drawing Algorithm

Answer:

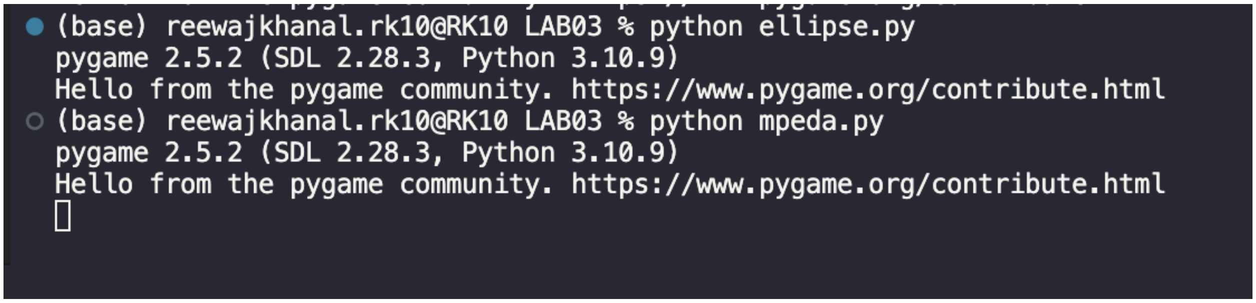
```
import pygame

from pygame.locals import *

from OpenGL.GL import *

from OpenGL.GLUT import *

from OpenGL.GLU import *
```

```
# Function to plot points in all four quadrants
def plot_ellipse_points(x_center, y_center, x, y):
   glBegin(GL_POINTS)
   # Quadrant: 1 Right Top
   glVertex2i(x_center + x, y_center + y)
   # Quadrant: 2 Right Bottom
   glVertex2i(x_center + x, y_center - y)
   #Quadrant: 3 Left Bottom
   glVertex2i(x_center - x, y_center - y)
   # Quadrant: 4 Left Top
   glVertex2i(x_center - x, y_center + y)
   glEnd()

# Mid-Point Ellipse Drawing Algorithm
def midpoint_ellipse(x_center, y_center, rx, ry):
   x = 0
   y = ry
   rx2 = rx * rx
   ry2 = ry * ry
   tworx2 = 2 * rx2
   twory2 = 2 * ry2
   p1 = ry2 - (rx2 * ry) + (0.25 * rx2)  # Decision parameter for region 1
   dx = twory2 * x
   dy = tworx2 * y

   # Region 1
   while dx < dy:
       plot_ellipse_points(x_center, y_center, x, y)
       if p1 < 0:
           x += 1

           dx += twory2

           p1 += dx + ry2

       else:

           x += 1

           y -= 1

           dx += twory2

           dy -= tworx2

           p1 += dx - dy + ry2


   # Region 2

   # Decision Parameter for region 2

   p2 = (ry2 * (x + 0.5) * (x + 0.5)) + (rx2 * (y - 1) * (y - 1)) - (rx2 * ry2)

   while y >= 0:

       plot_ellipse_points(x_center, y_center, x, y)

       if p2 > 0:

           y -= 1

           dy -= tworx2

           p2 += rx2 - dy

       else:

           x += 1

           y -= 1

           dx += twory2

           dy -= tworx2

           p2 += dx - dy + rx2


def draw_axes():

   glBegin(GL_LINES)

   # Draw X axis

   glVertex2i(-400, 0)
   glVertex2i(400, 0)
   # Draw Y axis
   glVertex2i(0, -300)
   glVertex2i(0, 300)
   glEnd()

def draw_ellipse(x_center=0,y_center=0,rx=100,ry=50):
   glClear(GL_COLOR_BUFFER_BIT)
   glColor3f(1.0, 1.0, 1.0)  # Set color to white
   # Draw the axes
   draw_axes()

   # Draw the ellipse
   midpoint_ellipse(x_center, y_center, rx, ry)  # Draw ellipse
   glFlush()

def get_input():
   x_center = int(input("Enter x coordinate of origin: "))
   y_center = int(input("Enter y coordinate of origin: "))
   rx = int(input("Enter x radius of ellipse: "))
   ry = int(input("Enter y radius of ellipse: "))
   return x_center, y_center, rx, ry

def main():
   pygame.init()
   display = (800, 600)
   pygame.display.set_mode(display, DOUBLEBUF | OPENGL)
   gluOrtho2D(-400, 400, -300, 300)  # Set up 2D coordinate system
   # x_center,y_center,rx,ry=get_input()
   running = True
```



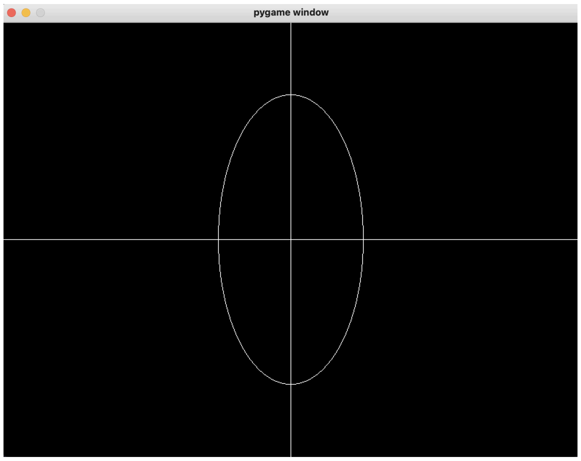
```
    while running:
        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                running = False

        draw_ellipse(x_center=0,y_center=0,rx=100,ry=200)
        pygame.display.flip()
        pygame.time.wait(10)

    pygame.quit()

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

Input:



```
● (base) reewajkhanal.rk10@RK10 LAB03 % python ellipse.py
  pygame 2.5.2 (SDL 2.28.3, Python 3.10.9)
  Hello from the pygame community. https://www.pygame.org/contribute.html
○ (base) reewajkhanal.rk10@RK10 LAB03 % python mpeda.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