

ELLIPSE DRAWING ALGORITHMS**AIM**

Write a menu driven program to draw an ellipse using

- a)Polar ellipse drawing algorithm
- b)Non Polar ellipse drawing algorithm

ALGORITHM

1. Display a menu and get user input to create ellipse according to any of the two algorithms
2. Input the center of the ellipse and major and minor radius of the ellipse and set it to xc, yc, rx and ry respectively
3. If choice = 1 (Polar Ellipse Generation Algorithm)
 - Set theta as 0 and target as (pi)/4
 - Loop till theta < target
 - Set x as $r \cdot \cos(\theta)$ and y as $r \cdot \sin(\theta)$
 - Plot the 4 points according to symmetry
 - Add some small value to theta (<0.2)
4. If choice = 2 (Non-polar Circle Generation Algorithm)
 - Set x as xc - rx and target as xc + rx
 - Loop till x < target
 - Compute offset as $ry \cdot \sqrt{1 - (x-xc)^2 / rx^2}$
 - Plot points (x, yc + offset) and (x, yc - offset)
 - Increment x by small value (<0.2)

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 = 5
```

```

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 Ellipse Algorithm")

    print(f"2. Polar Ellipse")

    print(f"3. Non-Polar Ellipse")

    print(f"0. Exit")

    return int(input("Enter Choice: "))


def get_input():

    # Function to get input from user

    xc, yc = map(int, input("Enter Coordinates of center of the ellipse seperated by space: (Eg. '0 0')").split(" "))

    rx, ry = map(int, input("Enter rx and ry seperated by space: (Eg. '5 10')").split(" "))

    return xc, yc, rx, ry


def get_points_polar_ellipse(xc: int, yc: int, rx: int, ry: int):

    # set color of points

    theta = 0

    factor = 500

    incr = 1 / factor

    target = math.pi / 2

    points = []

    while (theta <= target):

```

```

x = rx * math.cos(theta)
y = ry * math.sin(theta)
points.append((x + xc, y + yc))
points.append((-x + xc, -y + yc))
points.append((-x + xc, y + yc))
points.append((x + xc, -y + yc))
theta += incr

```

```

return points

```

```

def get_points_nonpolar_ellipse(xc: int, yc: int, rx: int, ry: int):

```

```

    # set color of points
    points = []
    x = xc - rx
    target = xc + rx
    points.append((x, yc))
    points.append((target, yc))
    factor = 500
    incr = 1 / factor
    x += incr
    while x < target:
        offset = ry * math.sqrt(1 - (math.pow(x - xc, 2) / math.pow(rx, 2)))
        points.append((x, yc + offset))
        points.append((x, yc - offset))
        x += incr

```

```

return points

```

```

def plot_ellipse(xc: int, yc: int, rx: int, ry: int, choice: int):

```

```

    # Function to plot the points

```

```

# Get points to plot
points = []
if choice == 1:
    points = get_points_polar_ellipse(xc, yc, rx, ry)
else:
    points = get_points_nonpolar_ellipse(xc, yc, rx, ry)

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(xc: int, yc: int, rx: int, ry: int, choice: int, title: str):
    # Function to display window
    print("Creating Window...")
    glutInit(sys.argv)
    glutInitDisplayMode(GLUT_RGB)
    glutInitWindowSize(500,500)
    glutInitWindowPosition(50, 50)
    glutCreateWindow(f"Plot Line using {title}")
    glutDisplayFunc(lambda: plot_ellipse(xc, yc, rx, ry, choice))
    init()

```

```
glutMainLoop()
```

```
def main():
```

```
    choice = 1
```

```
    titleList = {
```

```
        1: "Polar ellipse drawing algorithm",
```

```
        2: "Non-Polar ellipse drawing algorithm",
```

```
    }
```

```
    while choice != 0:
```

```
        choice = display_menu()
```

```
        if choice == 1 or choice == 2:
```

```
            # Checks if it's a valid input
```

```
            xc, yc, rx, ry = get_input()
```

```
            display_window(xc, yc, rx, ry, choice, titleList[choice])
```

```
        elif choice == 0:
```

```
            # To handle exit from program
```

```
            print("Exiting Program...")
```

```
        else:
```

```
            # To handle invalid choice
```

```
            print("Invalid Choice! Try again.")
```

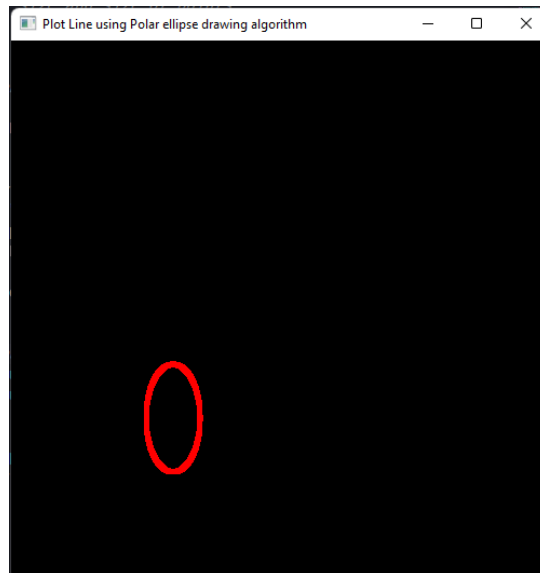
```
main()
```

RESULT

Program to draw a circle using a menu driven program using Mid point circle drawing algorithm, Polar circle generation algorithm, Non-Polar circle generation algorithm

OUTPUT/INPUT

```
-----MENU-----  
1. Polar Ellipse  
2. Non-Polar Ellipse  
0. Exit  
Enter Choice: 1  
Enter Coordinates of center of the ellipse seperated by space: (Eg. '0 0')30 30  
Enter rx and ry seperated by space: (Eg. '5 10')5 10  
Creating Window...
```



```
-----MENU-----  
1. Polar Ellipse  
2. Non-Polar Ellipse  
0. Exit  
Enter Choice: 2  
Enter Coordinates of center of the ellipse seperated by space: (Eg. '0 0')30 30  
Enter rx and ry seperated by space: (Eg. '5 10')5 10  
Creating Window...
```

