**SIMULATE ELLIPTICAL ORBITS IN PYGAME**

**Aim:**

To write a Python program to simulate elliptical orbits using Pygame.

**Algorithm:**

1. Import the required packages

2. Set up the colours for the elliptical orbits

3. Define the parameters to simulate elliptical orbits

4. Display the created orbits

**Program:**

import pygame

import math

import sys

pygame.init()

screen = pygame.display.set\_mode((600, 300))

pygame.display.set\_caption("Elliptical orbit")

clock = pygame.time.Clock()

while(True):

for event in pygame.event.get():

if event.type == pygame.QUIT:

sys.exit()

xRadius = 250

yRadius = 100

for degree in range(0,360,10):

x1 = int(math.cos(degree \* 2 \* math.pi / 360) \* xRadius) + 300

y1 = int(math.sin(degree \* 2 \* math.pi / 360) \* yRadius) + 150

screen.fill((0, 0, 0))

pygame.draw.circle(screen, (255, 0, 0), [300, 150], 35)

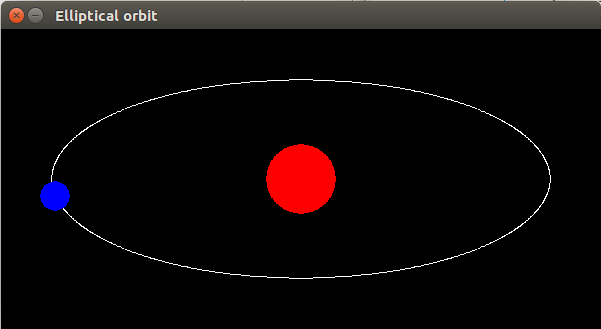
pygame.draw.ellipse(screen, (255, 255, 255), [50, 50, 500, 200], 1)

pygame.draw.circle(screen, (0, 0, 255), [x1, y1], 15)

pygame.display.flip()

clock.tick(5)

**Output:**



**Result:**

Thus the Python Program to simulate elliptical orbits using Pygame is executed successfully and the output is verified.

**SIMULATE BOUNCING BALL USING PYGAME**

**Aim:**

To write a Python program to simulate bouncing ball using Pygame.

**Algorithm:**

1. Import the required packages

2. Define the required variables

3. Define the screen space to display the bouncing balls in that space

**Program**:

**PROGRAM CODE:**

import sys, pygame

pygame.init()

size = [width, height] = [320, 240]

speed = [2, 2]

black = (0, 0, 0)

screen = pygame.display.set\_mode(size)

ball = pygame.image.load("ball.bmp")

ballrect = ball.get\_rect()

while 1:

for event in pygame.event.get():

if event.type == pygame.QUIT: sys.exit()

ballrect = ballrect.move(speed)

if ballrect.left < 0 or ballrect.right > width:

speed[0] = -speed[0]

if ballrect.top < 0 or ballrect.bottom > height:

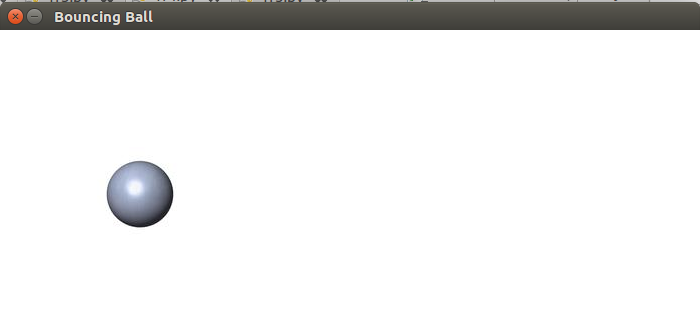
speed[1] = -speed[1]

screen.fill(black)

screen.blit(ball, ballrect)

pygame.display.flip()

**Output:**



**Result:**

Thus the Python Program to simulate bouncing ball using Pygame is executed successfully and the output is verified.