**EX NO: 12 Simulate elliptical orbits in Pygame**

**Date :**

**AIM**

To simulate elliptical orbits in pygame.

**ALGORITHM**

**Step 1:**  Import and initialize pygame with **pygame.init()  
Step 2:** Create a graphical screen (Surface) with pygame.display.set\_mode().  
**Step 3:** Initialize earth and a particle (say moon).  
**Step 4:** Inside the infinite loop, apply\_gravity to the particle and redraw   
**Step 5:** Draw the ball in its next position on the screen using Surface.blit() method.   
**Step 6:** The pygame.display.flip() method makes everything we have drawn on the screen Surface become visible.  
**Step 7:** If user triggers quit() event (close button), the simulation stops.

**SOURCE CODE**

#Testing Orbits

import pygame

import math

import random

class Particle ():

def \_\_init\_\_ (self, x, y, colour=0x000000):

self.x = x

self.y = y

self.vx = 0

self.vy = 0

self.colour = colour

def apply\_gravity (self, target):

"""Accelerates the particle towards some mass at target."""

dsqd = (self.x - target.x) \*\* 2 + (self.y - target.y) \*\* 2

if dsqd == 0:

return

self.vx += -1 / dsqd \* (self.x - target.x) / dsqd \*\* 0.5

self.vy += -1 / dsqd \* (self.y - target.y) / dsqd \*\* 0.5

def update (self):

self.x += self.vx

self.y += self.vy

pygame.init()

screen = pygame.display.set\_mode ((600, 400))

main\_surface = pygame.Surface ((600, 400))

p = Particle (200, 100, 0x111111)

earth = Particle (200, 200)

while (True):

clock.tick(30)

for event in pygame.event.get():

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

pygame.draw.circle (main\_surface, 0x00FF00, (earth.x, earth.y), 5, 2)

p.apply\_gravity (earth)

p.update ()

pygame.draw.circle (main\_surface, p.colour, (int (p.x), int (p.y)), 5, 2)

screen.blit(main\_surface, (0, 0))

pygame.display.flip()

**OUTPUT**

**RESULT**

Thus the elliptical orbits in pygame was simulated successfully.