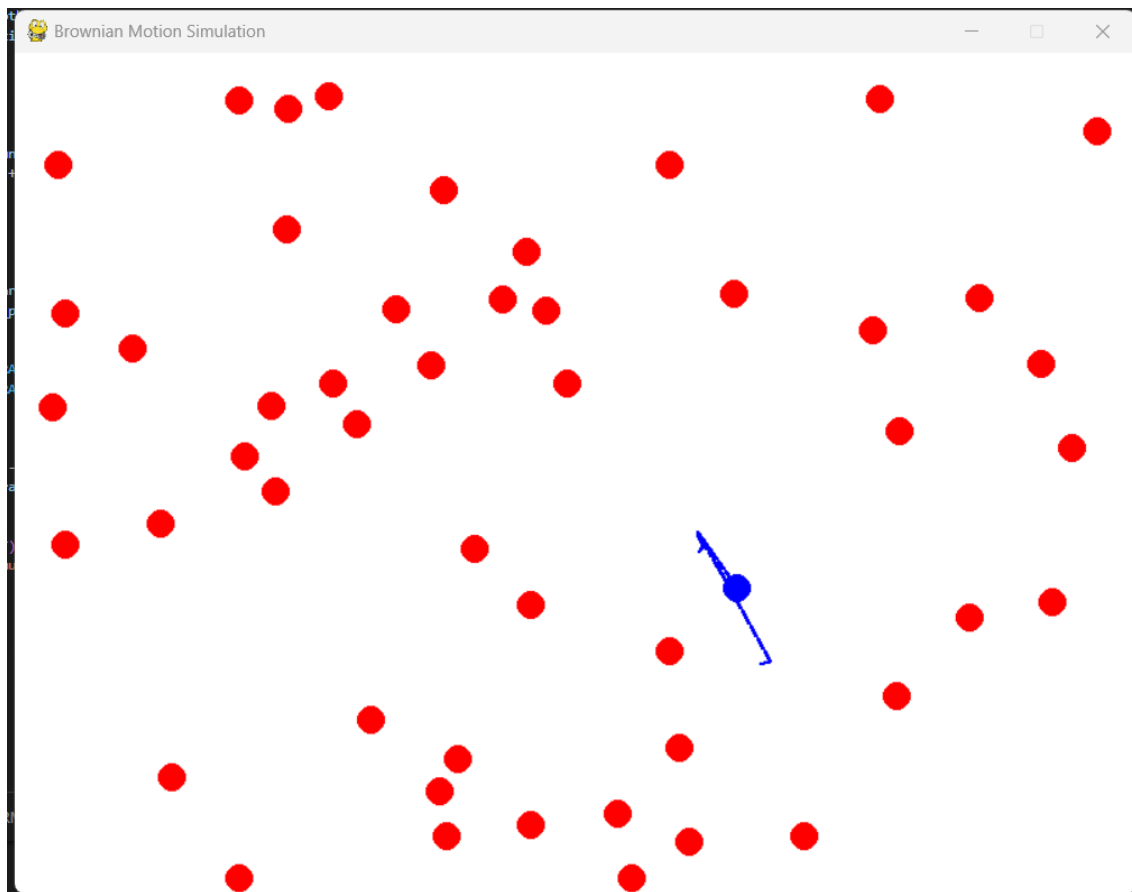


Brownian Motion

Vasco Pires & Duarte Gonçalves



1 Introdução

Neste projeto vais recriar o movimento browniano de um sistema com diversas partículas. Escolhe uma partícula na qual vais desenhar o seu trajeto e distingue-a das outras colocando a mesma com cor azul e o restante com cor vermelha. Vais ainda desenvolver a classe "Partícula" com os difentes atributos e métodos específicos.

2 Imports

Antes de escreveres o teu código é importante fazeres os seguintes imports para poderes usar as bibliotecas necessárias:

```
import pygame
import sys
import random
import math

pygame.init()
```

3 Constantes

```
# Constants
WIDTH, HEIGHT = 800, 600
FPS = 60
PARTICLE_RADIUS = 10
NUMPARTICLES = 50
MAXSPEED = 2

# Colors
WHITE = (255, 255, 255)
RED = (255, 0, 0)
BLUE = (0, 0, 255)
```

4 Funções

Para o código em si, vais desenvolver a classe "Partícula" que terá o seguinte formato:

```
# Particle class
class Particle:
def __init__(self, x, y, is_tracer=False):
    self.x = x
    self.y = y
    self.radius
    self.color
    self.speed
    self.angle
    self.is_tracer
    self.path

def move(self):
    return

def check_collision(self, other_particle):
    return
```

4.1 Atributos

1. x: posição no x
2. y: posição no y
3. radius: raio da partícula
4. color: cor da partícula
5. speed: velocidade da partícula
6. angle: ângulo em que a partícula se move
7. is_tracer: se é a partícula na qual querem guardar a trajetória
8. path: caminho da partícula (apenas da específica)

4.2 Métodos

1. move(): método que atualiza a posição da partícula em função da colisão com partículas ou com a caixa
2. check_collision(): método que verifica se houve uma colisão entre partículas

5 Plot

Para poderes ver o trajeto das tuas partículas a aparecerem numa interface gráfica no teu computador, coloca as seguintes linhas de código no final:

```
screen = pygame.display.set_mode((WIDTH, HEIGHT))
pygame.display.set_caption("Brownian Motion Simulation")
clock = pygame.time.Clock()

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

    # Move particles and check collisions
    for particle in particles:
        particle.move()

    # Draw particles and paths
    screen.fill(WHITE)
    for particle in particles:
        pygame.draw.circle(screen, particle.color, (int(particle.x), int(particle.y)), particle.radius)

    # Draw path for the tracer
    if particle.is_tracer and len(particle.path) >= 2:
        pygame.draw.lines(screen, particle.color, False, particle.path, 2)

    pygame.display.flip()
    clock.tick(FPS)
```

6 Exemplo

Podes pegar no exemplo seguinte e completá-lo:

```
1  import pygame
2  import sys
3  import random
4  import math
5
6  # Initialize Pygame
7  pygame.init()
8
9  # Constants
10 WIDTH, HEIGHT = 800, 600
11 FPS = 60
12 PARTICLE_RADIUS = 10
13 NUM_PARTICLES = 50
14 MAX_SPEED = 2
15
16 # Colors
17 WHITE = (255, 255, 255)
18 RED = (255, 0, 0)
19 BLUE = (0, 0, 255)
20
21 # Particle class
22 class Particle:
23     def __init__(self, x, y, is_tracer=False):
24         self.x = x
25         self.y = y
26         self.radius
27         self.color
28         self.speed
29         self.angle
30         self.is_tracer
31         self.path
32
33     def move(self):
34         return
35
36     def check_collision(self, other_particle):
37         return
38
39
40 # Create particles
41 particles = []
42
43 # Choose one particle as a tracer
44 tracer_index = random.randint(0, NUM_PARTICLES - 1)
45 particles[tracer_index] = ?
46
47 # Set up Pygame screen
48 screen = pygame.display.set_mode((WIDTH, HEIGHT))
49 pygame.display.set_caption("Brownian Motion Simulation")
50 clock = pygame.time.Clock()
51
52 # Main loop
53 while True:
54     for event in pygame.event.get():
55         if event.type == pygame.QUIT:
56             pygame.quit()
57             sys.exit()
58
59     # Move particles and check collisions
60     for particle in particles:
61         particle.move()
62
63     # Draw particles and paths
64     screen.fill(WHITE)
65     for particle in particles:
66         pygame.draw.circle(screen, particle.color, (int(particle.x), int(particle.y)), particle.radius)
67
68     # Draw path for the tracer
69     if particle.is_tracer and len(particle.path) >= 2:
70         pygame.draw.lines(screen, particle.color, False, particle.path, 2)
71
72     pygame.display.flip()
73     clock.tick(FPS)
74
```