Documentación del Código de Simulación de Escena

]ROBBYEL ELIAS, CARLOS JERONIMO, ANDRUS LOPEZ]

Abstract

Este documento describe la implementación de una simulación de escena en Python utilizando clases y animaciones. Se explican los métodos principales y sus funcionalidades, incluyendo la generación de escenas con rectángulos y puntos, su modificación, desplazamiento y animación.

1 Introducción

Este documento presenta la documentación detallada del código de simulación de escena. La simulación consiste en la representación gráfica de un área rectangular con puntos aleatorios y su animación mediante desplazamientos en distintas direcciones.

2 Descripción del Código

El código está estructurado en una clase llamada `Escena`, que encapsula todas las funcionalidades necesarias para manejar y modificar una representación gráfica.

2.1 Inicialización de la Escena

El constructor de la clase `Escena` inicializa los parámetros básicos:

- `ancho`, `alto`: dimensiones del área rectangular.
- `color`: color del rectángulo de fondo.
- `n_puntos`: cantidad de puntos generados aleatoriamente.

2.2 Método para Dibujar la Escena

El método `dibujar_escena` representa gráficamente el rectángulo y los puntos generados. Se usa la librería `matplotlib` para trazar el rectángulo y los puntos dentro de la escena.

2.3 Desplazamiento de Puntos

El método `desplazar_puntos(valor, direccion)` permite mover los puntos en diferentes direcciones:

Parámetros:

- `valor`: cantidad de desplazamiento.
- `direccion`: dirección del movimiento (`'derecha'`, `'izquierda'`, `'arriba'`, `'abajo'`).

2.4 Animación de la Escena

El método `animar(velocidad, direccion)` genera una animación de los puntos desplazándose en la dirección especificada. Se usa la función `FuncAnimation` de `matplotlib.animation`.

3 Código Fuente

```
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.patches as patches
import matplotlib.animation as animation
from IPython.display import HTML, display

class Escena:
    def _init_(self, ancho, alto, color, n_puntos):
        self.ancho = ancho
        self.alto = alto
        self.color = color
        self.puntos = np.random.rand(n_puntos, 2) * [

    def cambiar_escena(self, ancho, alto, color, n_pu
        self._init_(ancho, alto, color, n_puntos)
```

```
def dibujar_escena(self):
    fig, ax = plt.subplots()
    ax.set_xlim(0, self.ancho)
    ax.set_ylim(0, self.alto)
    rect = patches.Rectangle((0, 0), self.ancho, 
    ax.add_patch(rect)
    ax.scatter(self.puntos[:, 0], self.puntos[:,
    ax.legend()
    plt.show()

def desplazar_puntos(self, valor, direccion):
    desplazamientos =
        'derecha': np.array([valor, 0]),
        'izquierda': np.array([-valor, 0]),
        'arriba': np.array([0, valor]),
        'abajo': np.array([0, -valor])

    if direccion in desplazamientos:
        self.puntos += desplazamientos[direccion]

def animar(self, velocidad, direccion):
    fig, ax = plt.subplots()
    ax.set_xlim(0, self.ancho)
    ax.set_ylim(0, self.alto)
```

```
            rect = patches.Rectangle((0, 0), self.ancho, 
            ax.add_patch(rect)
            scatter = ax.scatter(self.puntos[:, 0], self.

            desplazamiento =
                'derecha': np.array([velocidad, 0]),
                'izquierda': np.array([-velocidad, 0]),
                'arriba': np.array([0, velocidad]),
                'abajo': np.array([0, -velocidad])
            .get(direccion, np.array([0, 0]))

            def update(frame):
                self.puntos[:] += desplazamiento
                scatter.set_offsets(self.puntos)
                return scatter,

            ani = animation.FuncAnimation(fig, update, fr
            return HTML(ani.to_jshtml())

    if _name_ == "_main_":
        escena = Escena(10, 5, 'blue', 10)
        escena.dibujar_escena()
        escena.desplazar_puntos(1, 'derecha')
        escena.dibujar_escena()
```

```
display(escena.animar(0.5, 'arriba'))
```

4 Conclusiones

Se presentó un código en Python para simular una escena gráfica con puntos en movimiento. Se explicaron las funciones principales y su implementación, proporcionando una visión clara del proceso de desarrollo.