

Transformaciones en una Escena 2D

[carlos jeronimo, robbyel elias, andrus lopez]

March 17, 2025

1 Introducción

Este documento presenta una implementación en Python para manejar una escena 2D que incluye un círculo y puntos simulando ojos. Se incluyen funciones para escalar, desplazar y rotar la escena.

2 Código Fuente

A continuación, se muestra el código en Python:

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 import matplotlib.patches as patches
4
5 class Escena:
6     def _init_(self, radio=5, color='darkblue'):
7         """
8         Inicializa la escena con un círculo centrado en el origen
9         y dos puntos simulando ojos.
10        """
11        self.radio = radio
12        self.color = color
13        self.puntos = np.array([[-1.5, 2], [1.5, 2]]) # Posiciones
14        de los ojos
15
16    def dibujar_escena(self):
17        """
18        Dibuja la escena con el círculo, los ejes y los puntos.
19        """
20        fig, ax = plt.subplots()
21        ax.set_xlim(-self.radio - 1, self.radio + 1)
22        ax.set_ylim(-self.radio - 1, self.radio + 1)
23
24        # Dibujar ejes cartesianos
25        ax.axhline(0, color='black', linewidth=1)
26        ax.axvline(0, color='black', linewidth=1)
27
28        # Dibujar círculo
29        circle = patches.Circle((0, 0), self.radio, color=self.
30        color)
31        ax.add_patch(circle)
```

```

29
30     # Dibujar puntos (ojos)
31     ax.scatter(self.puntos[:, 0], self.puntos[:, 1], color='red', s=100)
32
33     plt.gca().set_aspect('equal', adjustable='datalim')
34     plt.show()
35
36     def escalar(self, factor):
37         """
38         Escala el círculo y los puntos por un factor dado.
39         """
40         self.radio *= factor
41         self.puntos *= factor
42
43     def desplazar(self, dx, dy):
44         """
45         Desplaza los puntos y el centro del círculo.
46         """
47         self.puntos += np.array([dx, dy])
48
49     def producto_interno(self, punto):
50         """
51         Calcula el producto interno entre los puntos y otro punto
52         dado.
53         """
54         return np.dot(self.puntos, punto)
55
56     def calcular_norma(self):
57         """
58         Calcula la norma de los puntos respecto al origen.
59         """
60         return np.linalg.norm(self.puntos, axis=1)
61
62     def rotar(self, angulo):
63         """
64         Rota los puntos un cierto ángulo en grados alrededor del
65         origen.
66         """
67         theta = np.radians(angulo)
68         rot_matrix = np.array([[np.cos(theta), -np.sin(theta)],
69                                [np.sin(theta), np.cos(theta)]])
70         self.puntos = self.puntos @ rot_matrix.T
71
72 # Ejemplo de uso
73 escena = Escena()
74 print("Dibujando escena inicial...")
75 escena.dibujar_escena()
76
77 print("Escalando la escena por un factor de 1.2...")
78 escena.escalar(1.2)
79 escena.dibujar_escena()
80
81 print("Desplazando la escena en (2, -1)...")
82 escena.desplazar(2, -1)
83 escena.dibujar_escena()

```

```
83 print("Rotando la escena 30 grados...")
84 escena.rotar(30)
85 escena.dibujar_escena()
```

Listing 1: Código Python para manejar la escena

3 Conclusión

Este código permite manipular una escena en 2D mediante transformaciones geométricas básicas, lo que facilita su visualización y análisis.