

Simulacion 2D

Juan Orozco, Diego Fernandez, Nayelis Jaimes

March 5, 2025

Abstract

Nuestro proyecto muestra una simulación interactiva que representa el movimiento de un punto dentro de un círculo, guardando y cuidando los límites entre si. Utilizando Python y la biblioteca matplotlib, se realiza la creación de clase llamada “Escena”, lo que nos deja modificar el tamaño y color del círculo, a su vez mover un punto en diferentes direcciones sin saltarse sus bordes establecidos.

Este código brinda métodos para así cambiar la configuración del círculo (cambiarescena), dibujar la escena (dibujarescena), mover el punto en una dirección específica (desplazarpuntos) y hacer la animación de movimiento (similar). Por medio de estos métodos, la simulación nos ayuda a visualizar de forma clara y sencilla el cómo el objeto se desplaza dentro de un espacio establecido.

Esta actividad no solo ayuda a reforzar conceptos de programación orientada a objetos y la manipulación gráfica en Python, sino que también muestra aplicaciones prácticas en áreas como el desarrollo de videojuegos, simulaciones físicas y modelado de trayectorias. Al probar con diferentes valores y configuraciones, los usuarios pueden entender mucho mejor cómo se realizan los movimientos y cómo las restricciones espaciales influyen en la dinámica de un objeto dentro de una zona definida..

0.1 Objetivos de la simulacion

el objetivo de esta simulacion es representar de manera sencilla el movimiento del punto dentro del circulo, atravez de este ejercicio logramos entender conceptos sencillos de la programacion de objetos.

0.2 Descripcion

el código crea una simulación sencilla donde un punto es movido dentro del circulo donde utilizamos la biblioteca matplotlib para eso se define con la clase llamada escena que representa como el punto en el interior del circulo esta clase permite modificar y mover el punto en varias direcciones el ejemplo de uso se crea una escena con el circulo de radio X luego este se desplaza en distintas direcciones y se vuelve a visualizar después que se ejecuta la simulación de movimiento hacia alguna direccion

0.3 Análisis de la simulacion

El código brinda una simulación del movimiento de un punto dentro de un círculo utilizando la librería matplotlib en Python. Por medio de la ejecución del programa, se observaron diferentes comportamientos en la escena, los cuales se detallan a continuación:

Escena Inicial: Se creo una visualización inicial con un círculo de color azul y un punto rojo en su centro. El círculo se programo con un radio de 5 unidades y se mantuvo estático a lo largo de la simulación. El punto rojo representa el objeto que se desplazará en la simulación.

Movimiento del Punto: La simulación consistió en desplazar el punto rojo en la dirección "derecha". Se realizaron 5 iteraciones, en la cual cada una de ellas el punto avanzó 1 unidad en el eje X. La observa que la escena se actualizó en cada paso, permitiendo ver el desplazamiento del punto en tiempo real. Se utilizó una pausa entre cada iteración para mejorar la percepción del movimiento.

Comportamiento General: - El círculo actuó como un área de referencia y no cambió de tamaño ni posición en ningún momento. - La función de desplazamiento verificó que el punto no excediera los límites del círculo, asegurando que el movimiento fuera controlado. - No se presentaron errores ni interrupciones inesperadas durante la ejecución del programa. - La animación del movimiento se realizó de manera progresiva y fluida.

Análisis de Resultados: El código logró simular correctamente un desplazamiento gradual dentro de un área definida. Se pudo visualizar claramente la transición del punto en la dirección especificada. La combinación de matplotlib y la función de tiempo permitió generar una animación sencilla pero efectiva. - Una mejora potencial sería la implementación de más opciones de movimiento, como diagonales o trayectorias curvas. - También se podrían agregar restricciones más estrictas para evitar que el punto salga de los límites del círculo.

0.4 Conclusion

Este código realizado en phyton nos ha dejado practicar con gráficos y animaciones donde utilizamos la biblioteca matplotlib, por medio de la creación un círculo y un punto en movimiento, creado en una escena. Trabajamos con conceptos muy puntuales de programación y vista organizada.

Las principales enseñanzas y conclusiones obtenidas de esto son: Utilización y organización del código. Se divide el programa en funciones específicas, lo cual nos ayuda a que sea más fácil entender, mejorar y reutilizar los proyectos en otros a futuro. Mejoramiento de los gráficos en phyton. La biblioteca Matplotlib nos deja mejorar, visualizar y dibujar las formas geométricas, lo que es clave a la hora de hacer animaciones simples. Simulaciones en movimiento. Cuando actualizamos y vamos haciendo el paso a paso las métricas y se vuelve a dibujar la escena, vemos como resultado, una animación sencilla, pero eficaz, lo que es super esencial en la programación básica. Se controla el tiempo en las animaciones. Con el uso de `time.sleep()`, es posible tomar descansos entre las actualizaciones, permitiendo así que el movimiento sea fluido. Mejoras para tener en cuenta con aplicaciones futuras: Esta la posibilidad de realizar mejoramiento del código sumando las colisiones con los bordes del círculo, trabajos mas complejos o agregar bibliotecas mas avanzadas como Pygame, lo que nos brinda animaciones mas detalladas. En conclusión, este código es un muy buen punto de inicio para aprender a trabajar con gráficos en Python y hacer animaciones simples. Su sistema modular y su enfoque práctico lo vuelve ideal para los que desean iniciar en la programación visual y en la simulación de objetos en movimiento.

0.4.1 graficos

se mostrara los graficos de la simulacion



