

Inputs: Q,A W,S E,D R,F 1: animación 1, 2: animación 2

Inicialización:

Inicializo las longitudes de cada joint, calculando la distancia inicial de los puntos y calculo los ángulos de rotación que tienen al inicio de la escena los ángulos.

Cálculo Iterativo de Posiciones y Rotaciones:

- Para cada articulación, la función:
 - Calcula el ángulo que hay respecto al siguiente joint
 - Calcula la **rotación acumulativa** aplicando el ángulo actual de la articulación sobre la rotación acumulada previa.
 - Calcula la **posición nueva** de la articulación desplazándola desde la anterior según la dirección del eslabón y la longitud correspondiente.

Transformaciones en el Espacio 3D:

- **Rotaciones acumulativas:**
 - Las rotaciones se calculan con Quaternion.Euler(), que toma ángulos en grados y los convierte a un Quaternion.
 - La rotación acumulativa para una articulación se obtiene multiplicando la rotación previa por la rotación de la articulación actual.
- **Posiciones acumulativas:**
 - La nueva posición de cada articulación se calcula usando la fórmula:

$$\text{Posición } i = \text{Posición } i-1 + (\text{Rotación acumulada} \times \text{Dirección base} \times \text{Longitud}).$$

Aquí, Dirección base es un vector que apunta en la dirección deseada

Actualización de las Transformaciones:

- Una vez calculadas las nuevas posiciones y orientaciones, estas se aplican a las transformaciones (Transform.position y Transform.rotation) de las articulaciones.

La animación funciona con posiciones fijas colocadas y ordenadas.