

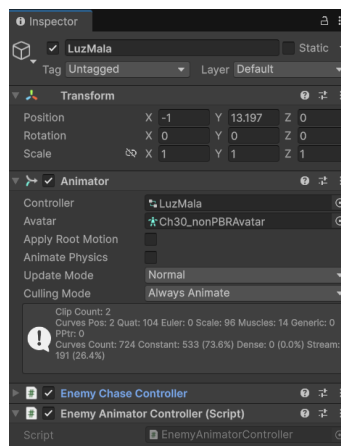
## Script EnemyChaseController

Cómo usarlo

1. **Agregá el script** al GameObject que realizará la persecución (Por defecto está asociado al LuzMala [GameObject vacío del que cuelgan la animación y efectos]).
2. En el **Inspector**, configura:
  - Si quieres detección automática del objetivo, asegurate de que ese GameObject tenga el tag "Player".
  - **Detection Radius:** el rango en el que el GameObject detecta al objetivo.
  - **Move Speed / Rotation Speed:** ajustá la velocidad de movimiento y rotación.
3. En la sección **Eventos**, puedes agregar acciones como:
  - Reproducir una animación cuando entra en rango.
  - Emitir un sonido.
  - Cambiar color, estado o activar un FX.

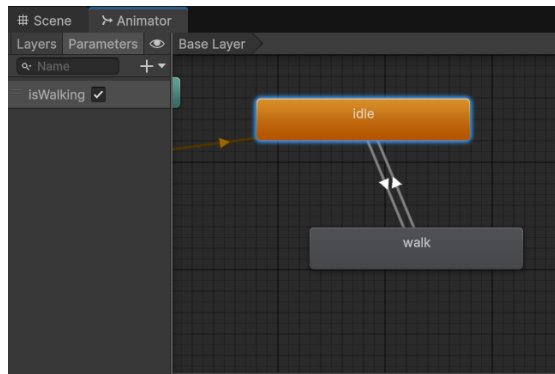
## Ejemplo Uso general de EnemyChaseController y EnemyAnimatorController

1. En **LuzMala (GameObject en la jerarquía)**, notarás que están ambos scripts:
  - EnemyChaseController
  - EnemyAnimationController



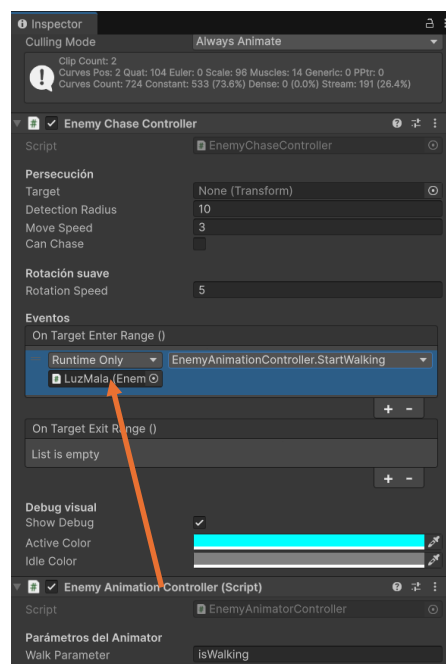
2. **LuzMala** también tiene un componente **Animator**. En el **Animator** del modelo (ubicado en Assets -> LuzMala -> Animation), encontrarás el Animator. Puedes asegurarte de tener el parámetro **isWalking** y las transiciones:


- Idle → Walk con condición **isWalking = true**
- Walk → Idle con **isWalking = false**

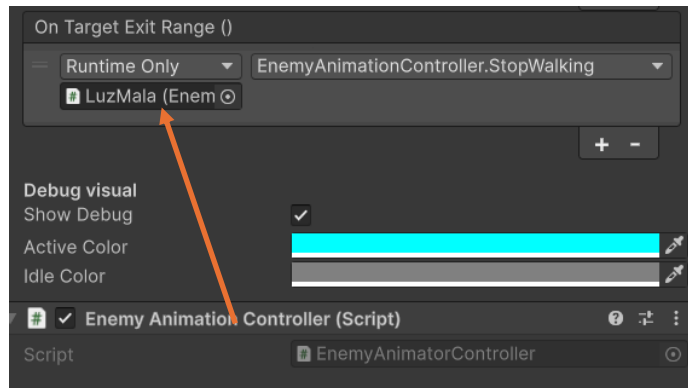


3. Al seleccionar el GameObject **LuzMala**, selecciona en el inspector el componente **EnemyChaseController** y despliega la sección **Eventos**:

- OnTargetEnterRange: **+** arrastra **LuzMala** → **EnemyAnimationController → StartWalking()**



- OnTargetExitRange:  → arrastra LuzMala → EnemyAnimationController → StopWalking()



- Activa canChase = true desde el Inspector o mediante SetChaseState(true) en tiempo de ejecución.

### Probando que funciona la persecución

Selecciona LuzMala.. y en el inspector verifica que tiene asignado un GameObject como Target (Aquí es donde puedes cambiar por otro, por defecto apunta a un cubo). Establece los valores de radio, velocidad y canChase debe estar activado)

