INTRODUCCIÓN A LA PROGRAMACIÓN DE VIDEOJUEGOS

ANIMACIÓN EN UNITY

Ramón Mollá
rmolla at dsic.upv.es - ext. 73549
Grupo de Informática Gráfica
Departamento de Sistemas Informáticos y Computación

Objetivos de aprendizaje

Generar animaciones de cualquier parámetrop empleando el motor de videojuegos Unity

Presentar el sistema de animación de Unity y cómo trabajar con él

Emplear las diferentes partes que utiliza

Crear una Animación Reactiva

Índice

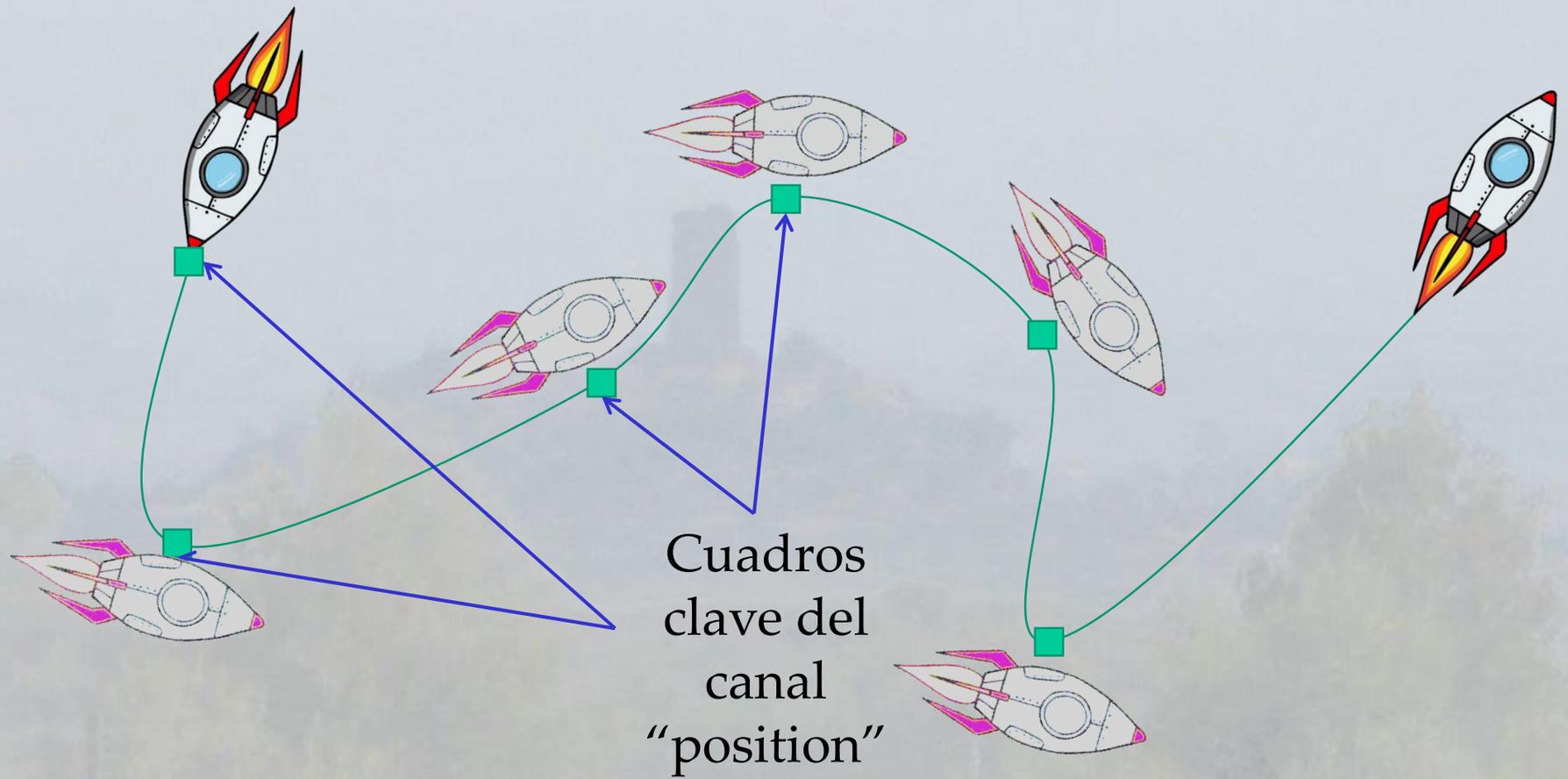
Cuadros clave

Una animación completa se compone de uno o más fotogramas clave

Un fotograma clave define el valor de un conjunto de propiedades en un momento dado en el tiempo

Cada propiedad se conoce como un canal

El motor de animación interpola posiciones intermedias (o tweens) a lo largo de la curva definida por los fotogramas clave



Sistema de Animación de Unity (I)

Sistema de animación equivalente a entornos de animación tipo 3D Studio Max, Maya, Softimage, Blender

Herramienta que diseñada para desarrollo de videojuegos pero que su potencia permite otros cometidos

https://unity.com/es/solutions/film-animation-cinematics

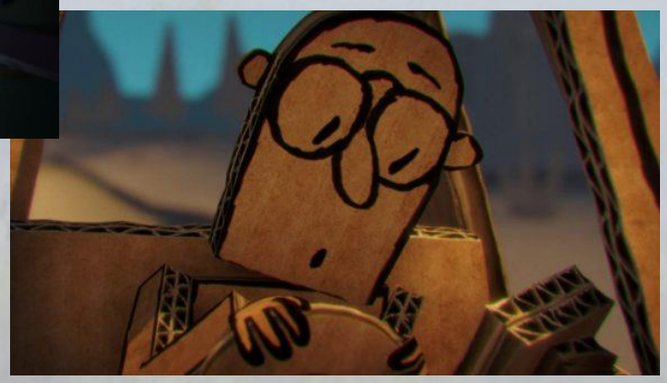
Demo Reel

Heretic

Adam

Book of the Dead

Sonder



Mr Carton

Sistema de Animación de Unity (II)

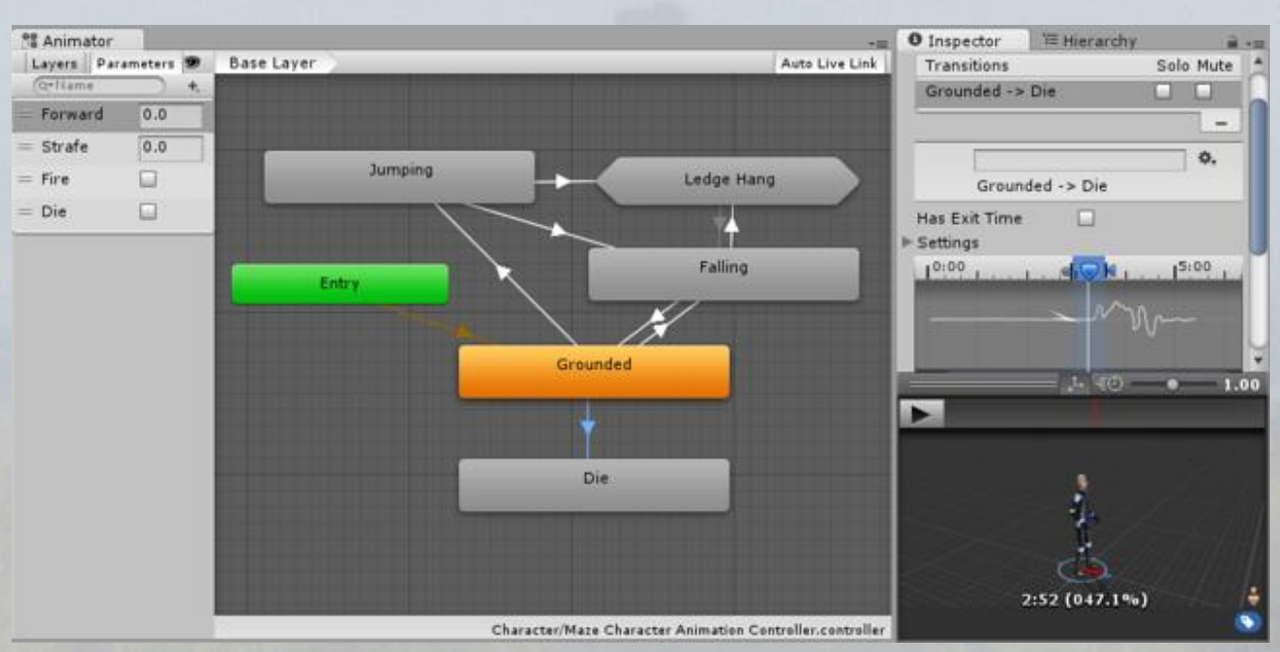
Se gestiona mediante máquinas de estado que se pueden usar para animar cualquier variable pública del juego

Los **estados** representan una animación

Las **transiciones** representan un cambio de una animación a otra

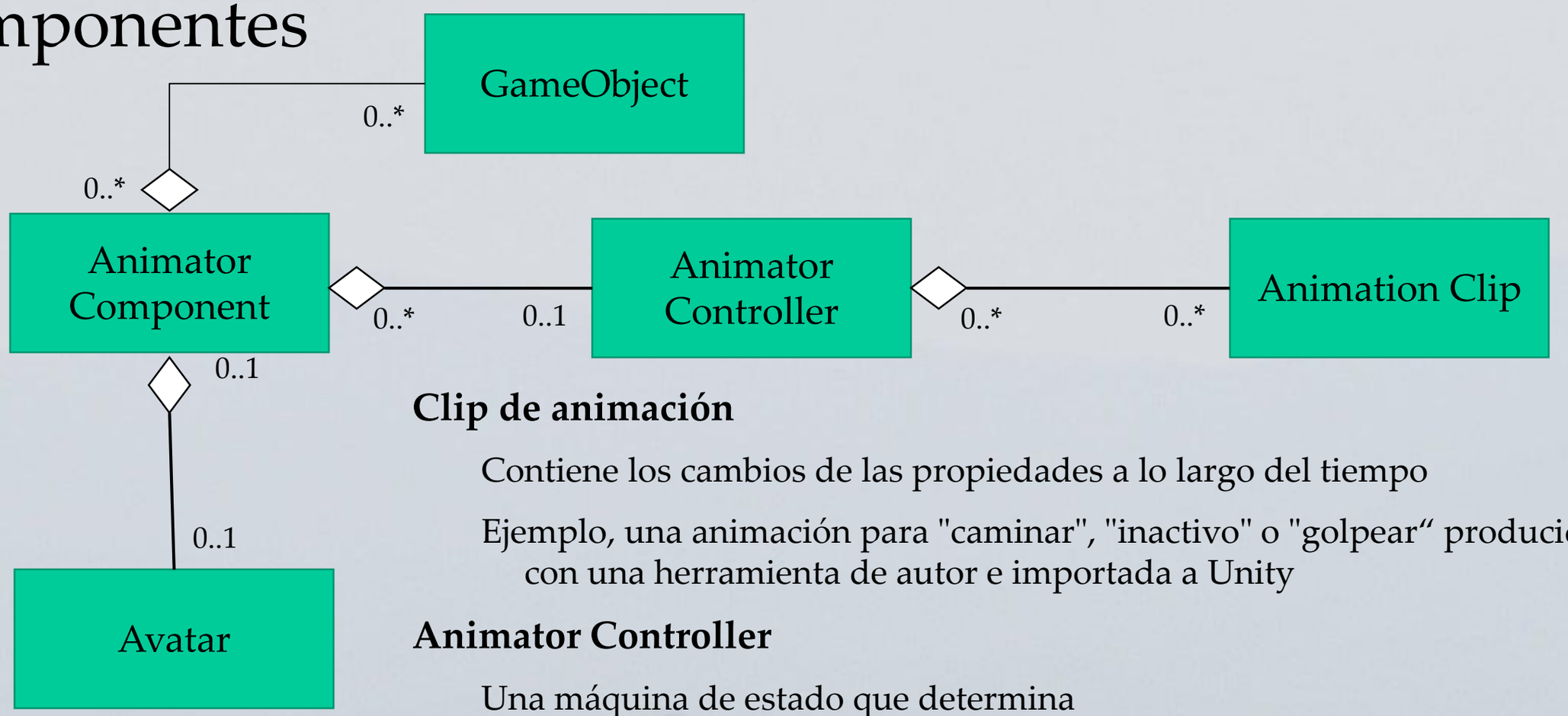
Se pueden combinar diferentes animaciones en un modelo (por ejemplo, para caminar y disparar un arma al mismo tiempo)

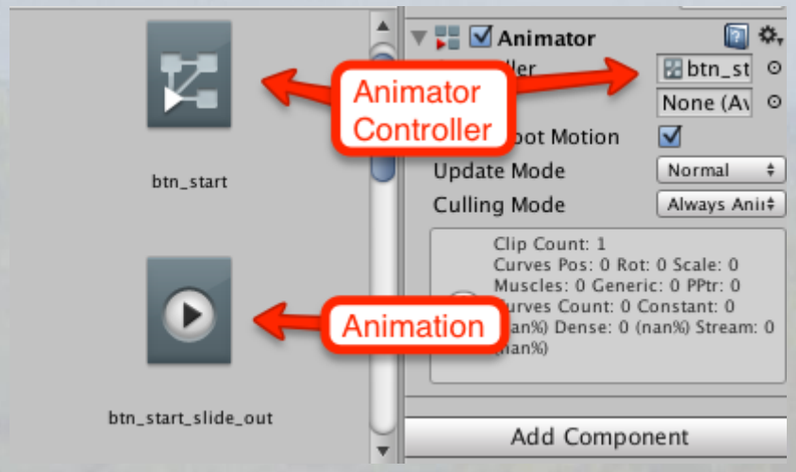
Las transiciones se pueden activar dependiendo de diferentes condiciones



Flujo de animación

Componentes





Clip de animación

Contiene los cambios de las propiedades a lo largo del tiempo

Ejemplo, una animación para "caminar", "inactivo" o "golpear" producida con una herramienta de autor e importada a Unity

Animator Controller

Una máquina de estado que determina

Qué clip de animación debe reproducirse

Cuándo debe cambiar la animación

Ejemplo, una animación simple de objetos o una animación compleja de varios clips de personajes

Animator Component

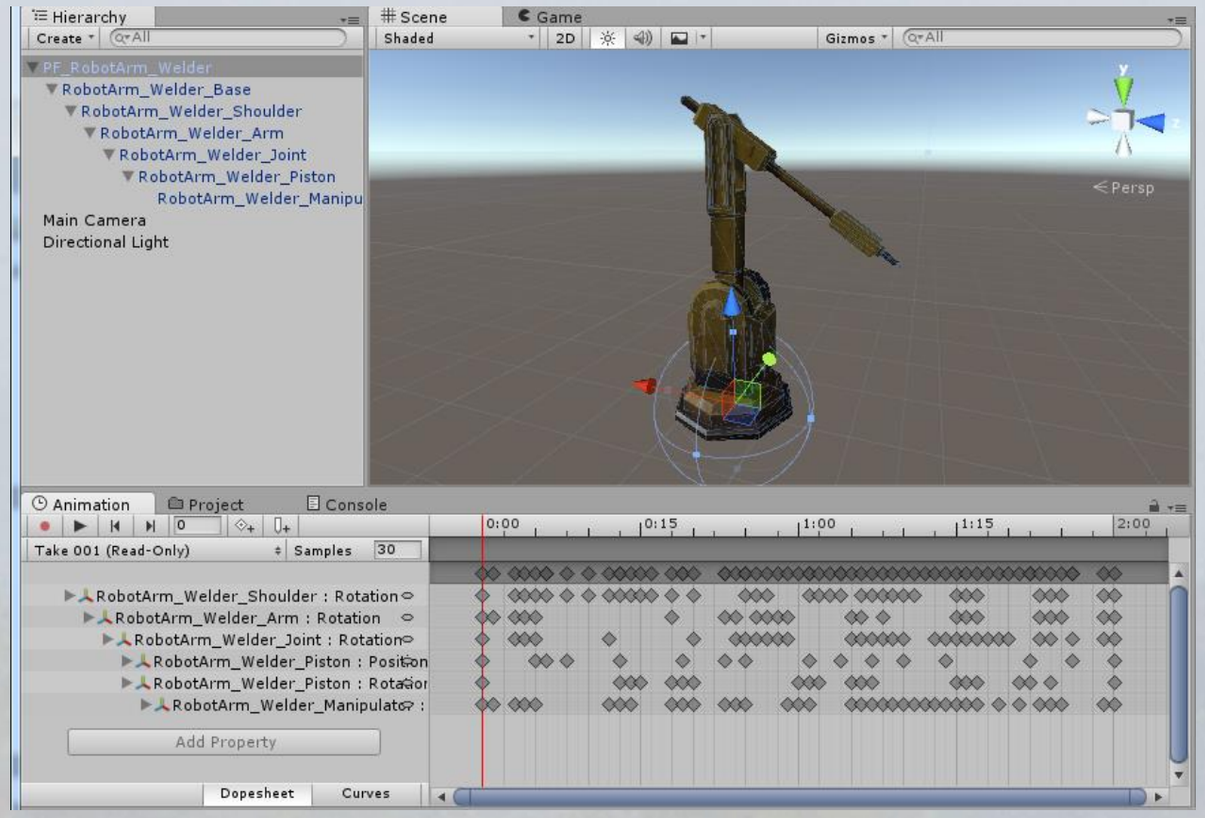
Componente que permite a un *GameObject* utilizar un Animator Controller

Avatar

El subsistema para realizar animación humanoide

Animación (I)

La ventana de animación muestra las animaciones asociadas con el GameObject seleccionado actualmente



La ventana de animación permite animar

- La posición, rotación y escala de los GameObjects
- Propiedades de los componentes tales como el color del material, la intensidad de una luz, el volumen de un sonido
- Propiedades dentro de sus propios scripts, incluidas las variables float, int, Vector y boolean

Animación (II)

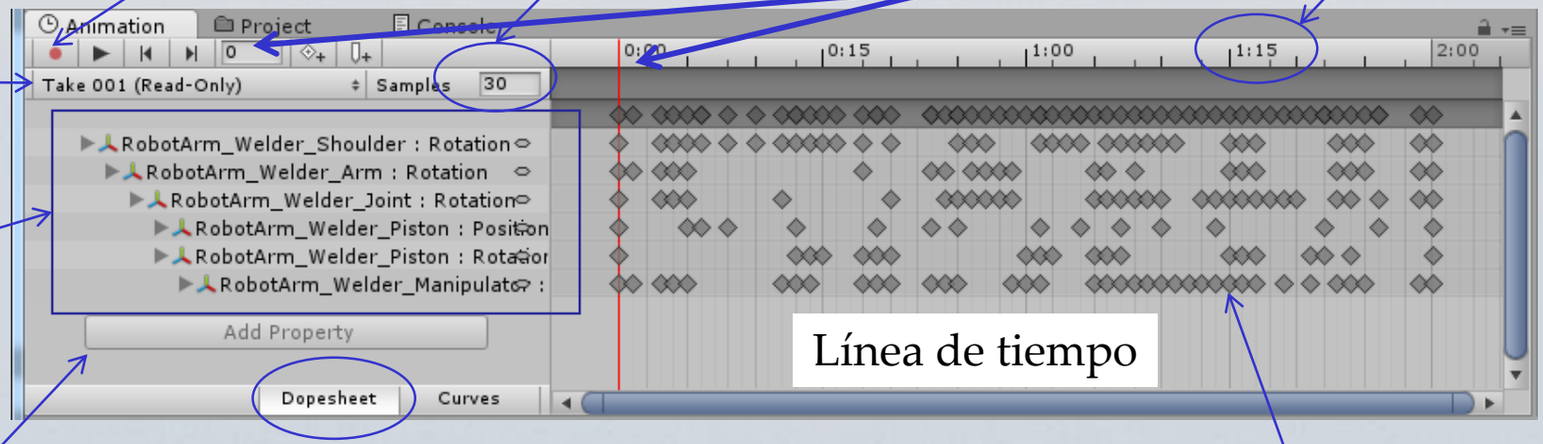
Clip de animación actual y creación de un nuevo clip

Entra en Modo de Grabación de Animación

Fotogramas por segundo

Reproducción actual

Segundos:Fotogramas

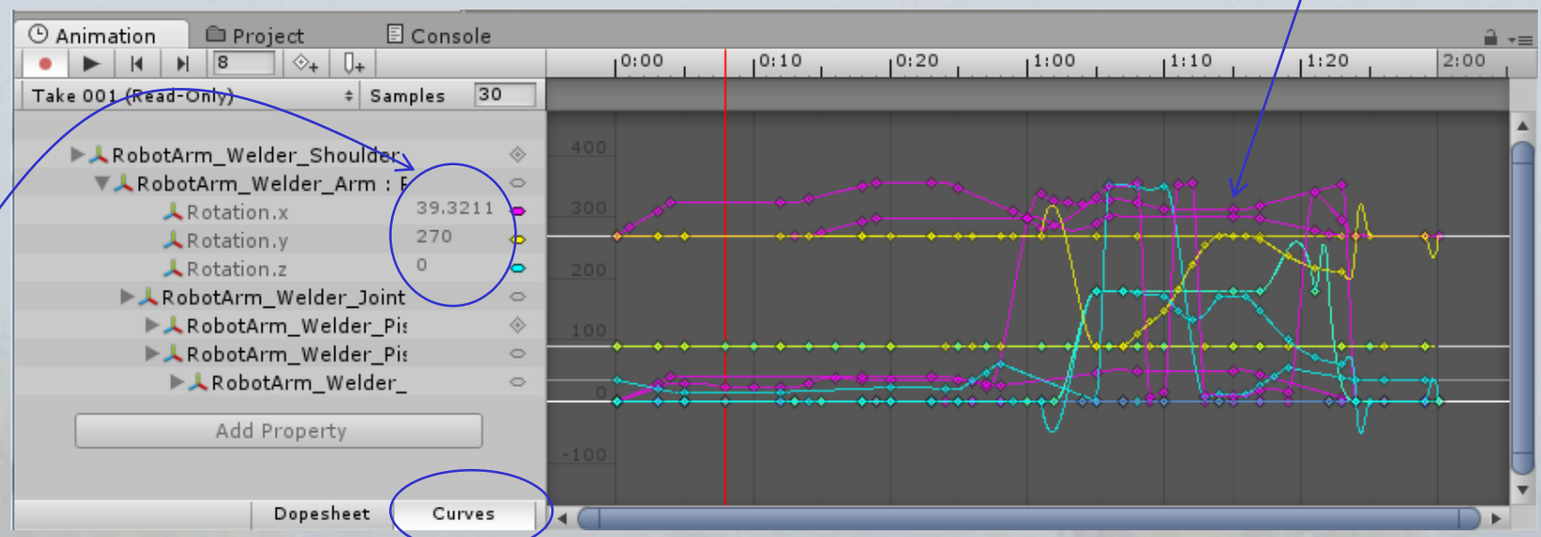


Propiedades animadas (jerarquía)

Añadir una nueva propiedad a animar

Cuadro clave

Valores interpolados en la posición de reproducción actual (si se editó agrega un nuevo fotograma clave)



Animación (III)

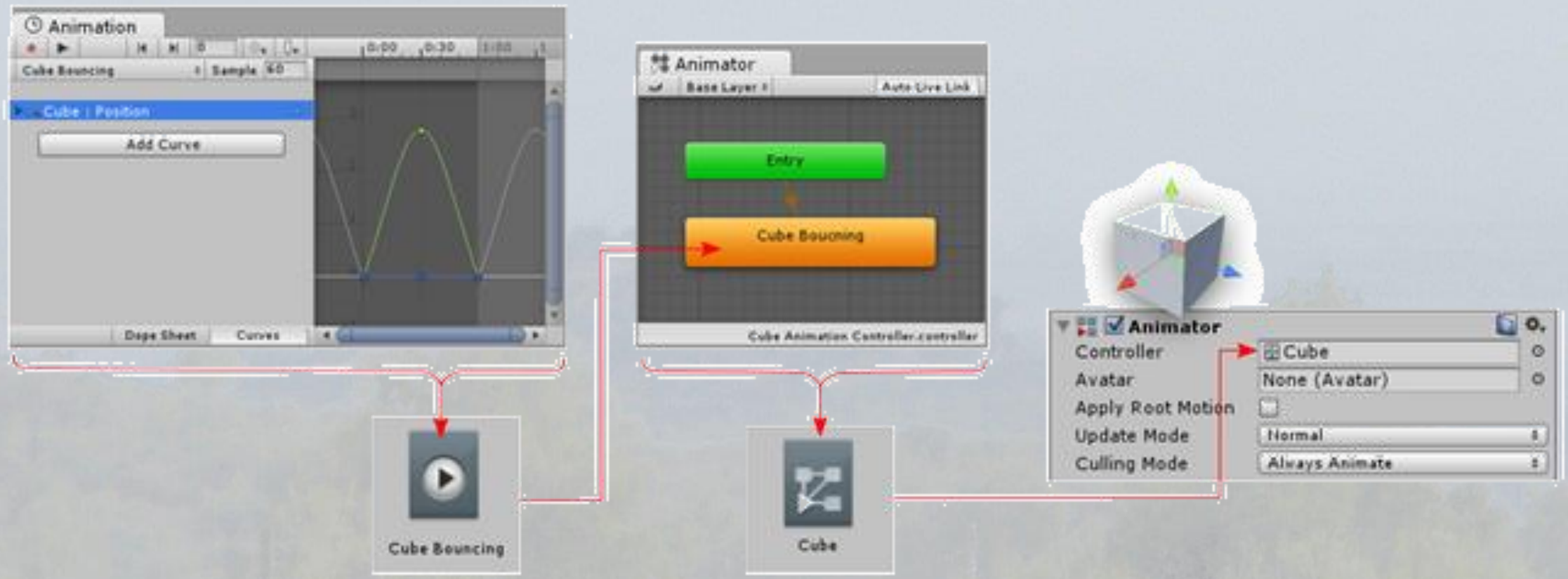
Al crear una animación para un GameObject que no tiene un componente Animator, Unity automáticamente

Crea un nuevo asset *Animator Controller*

Agrega el nuevo clip de animación en el *Animator Controller* como estado predeterminado

Agrega un componente *Animator* al *GameObject* que se está animando

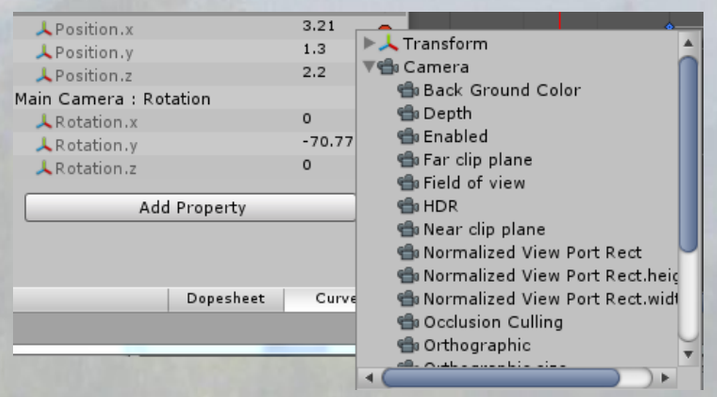
Asigna al nuevo *Animator Controller* al componente *Animator*



Animación (IV)

Modo de grabación de animación

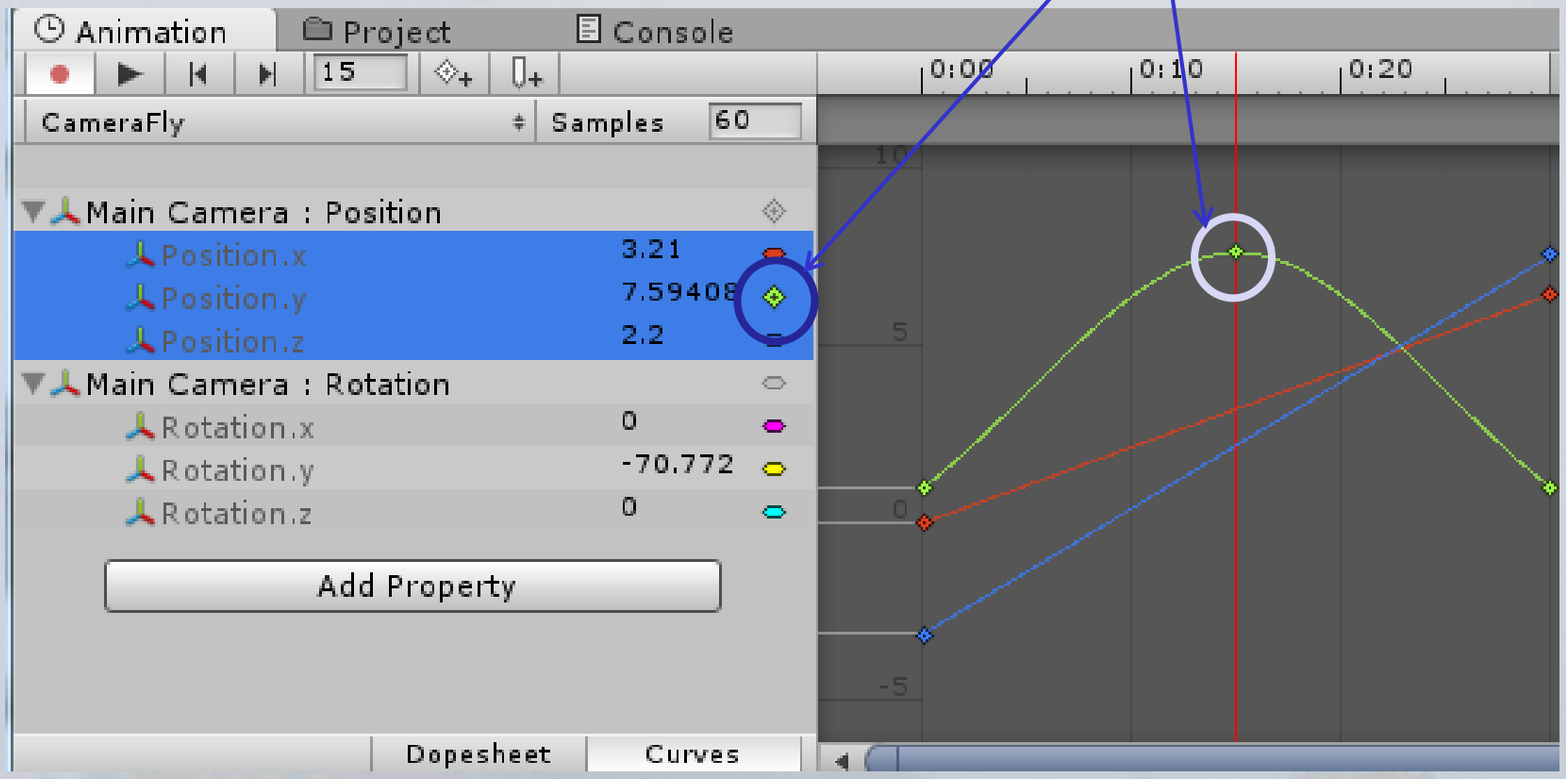
- Se inicia al crear un nuevo clip de animación o al hacer clic en el botón Guardar
- Cualquier cambio realizado en GameObject se registrará en el fotograma clave en la posición indicada por la línea roja en la línea de tiempo
- Transformar el objeto cambia las propiedades en el inspector o cambia los valores directamente en la vista de Animación
- Un clip de animación puede contener propiedades de los hijos actuales de GameObject
- Use el botón Agregar propiedad para agregar una curva
- Hacer clic en el botón Guardar nuevamente para salir de Registro modo



Animación (V)

Un diamante indica que la propiedad tiene una clave en el fotograma clave seleccionado

Solo las propiedades seleccionadas se muestran en la línea de tiempo



La curva pasa por todas las claves

El eje X es el tiempo, el eje Y es el valor de un atributo

Animación (VI)

Agregar claves

- Hacer doble clic en la curva en el punto donde se debe colocar la llave
- Hacer clic con el botón derecho en una curva y seleccionar Agregar clave en el menú contextual

Arrastrando claves

- Hacer clic en una clave para seleccionarla. Arrastrar la clave seleccionada con el ratón
- Mantener presionada la teca Control para ajustar a la cuadrícula mientras arrastra el ratón

Seleccionar múltiples claves

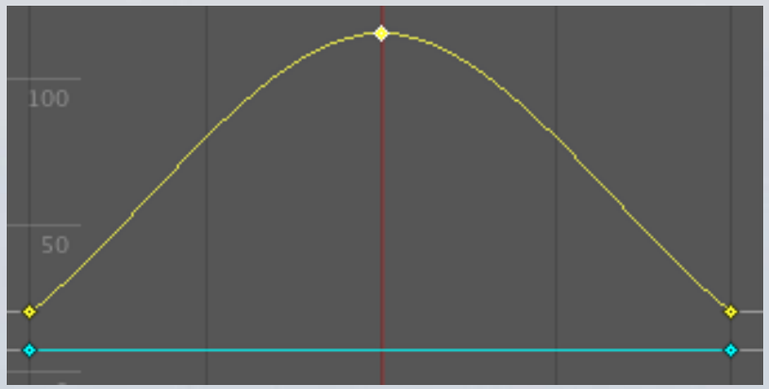
- Mantener presionada la tecla Mayús mientras se hace clic en las claves
- Hacer clic en un punto vacío y arrastrar para formar una selección rectangular

Eliminar claves

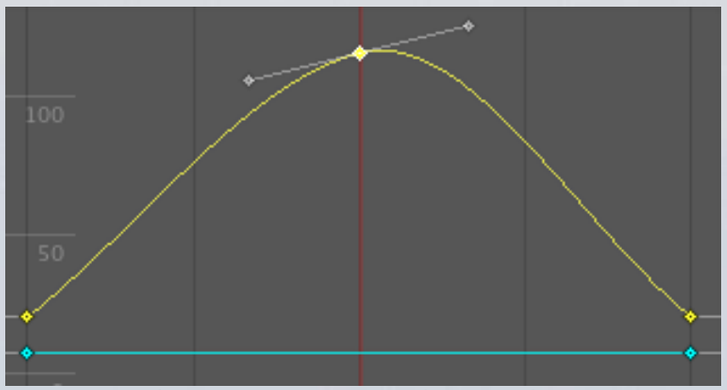
- Seleccionar las claves a borrar y presionar Eliminar
- Hacer clic derecho sobre ellas y seleccionar Eliminar clave en el menú contextual

Animator (I)

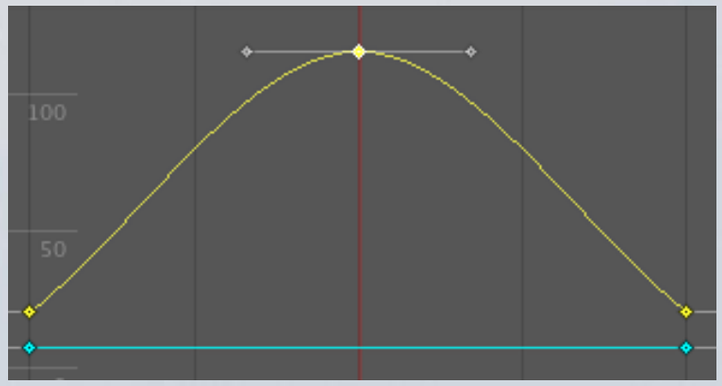
La forma de las curvas puede controlarse por medio de sus tangentes



Auto (smooth)

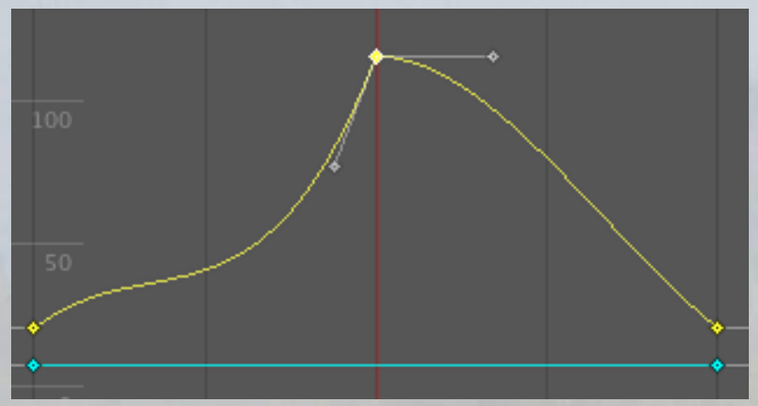


Libre (pero colineal)

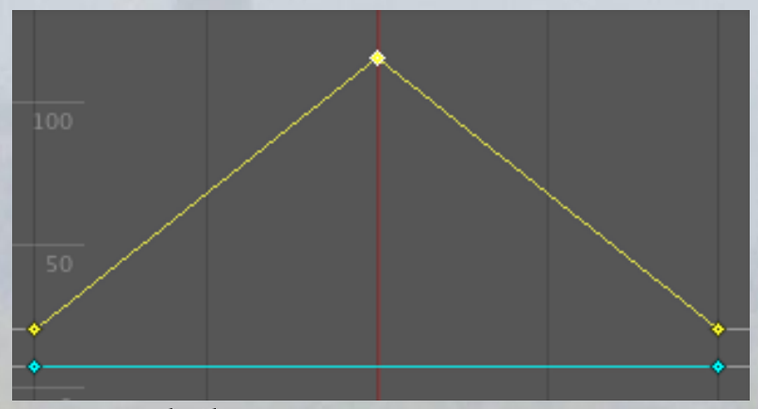


Plano (caso especial de libre)

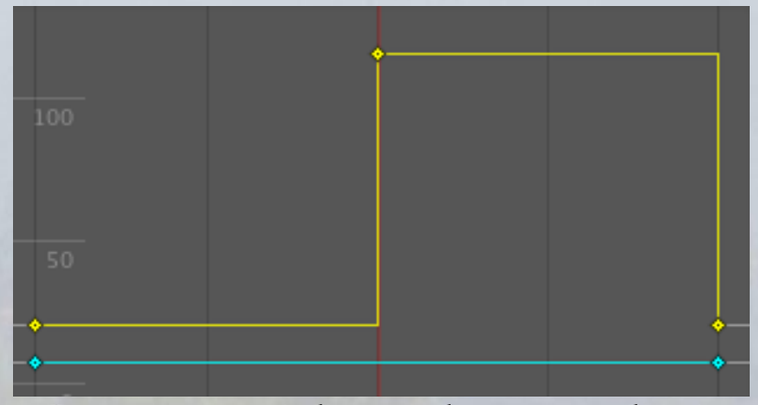
Si se necesita más control, se puede definir cada tangente (izquierda y derecha de cada fotograma clave) de manera diferente



Libre (tangentes independientes)



Lineal (las tangentes apuntan hacia las claves vecinas)



Constante (el canal no cambia)

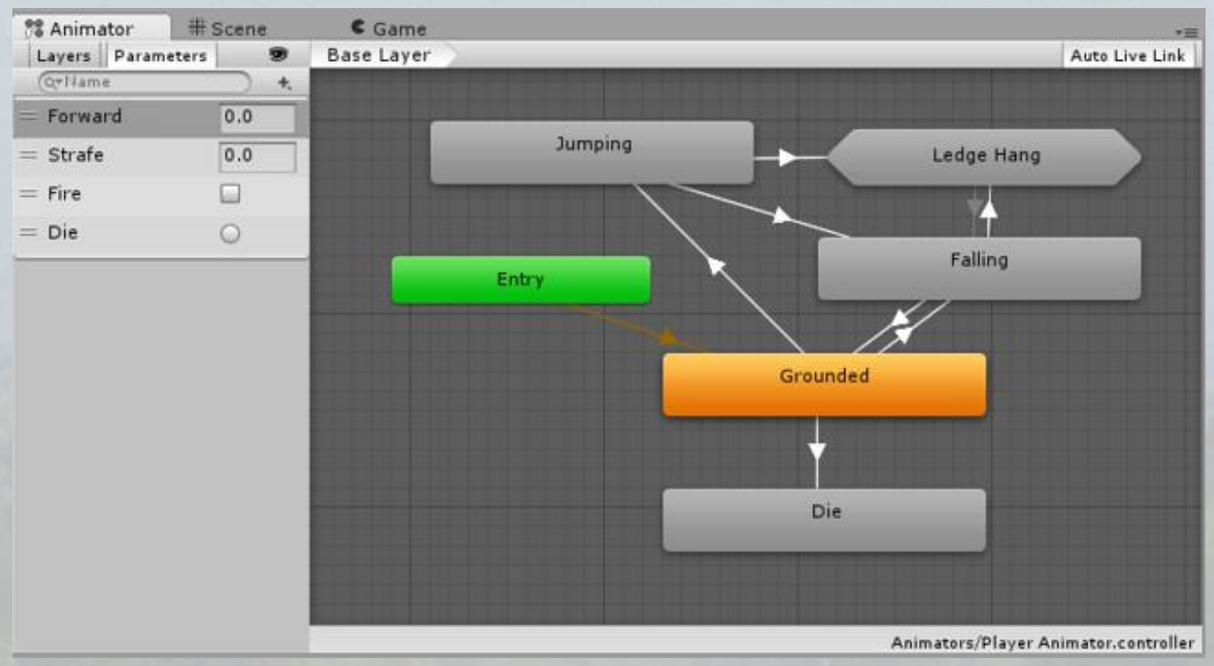
Animator (II)

Mantiene un conjunto de animaciones para un GameObject animado

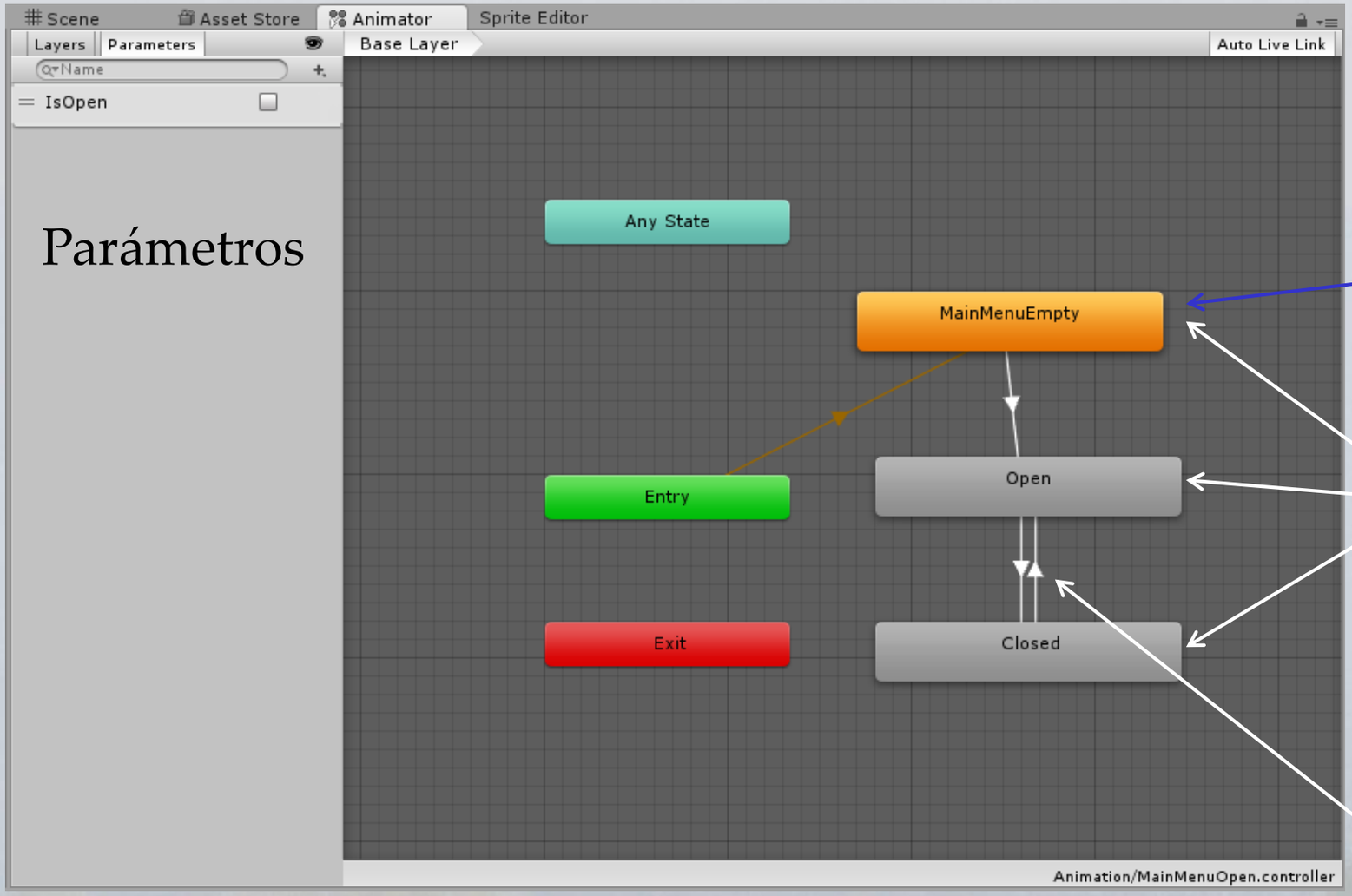
El controlador

- Tiene referencias a los clips de animación usados dentro de él
- Gestiona los diversos estados de animación
- Gestiona las transiciones entre ellos usando una máquina de estado

El *Animator Controller* se configura en la *Animator Window,* no en la ventana de animación



Animator (III)



Parámetros

Animation Clip por defecto

Animation Clips

Transiciones

Animator Controller (I)

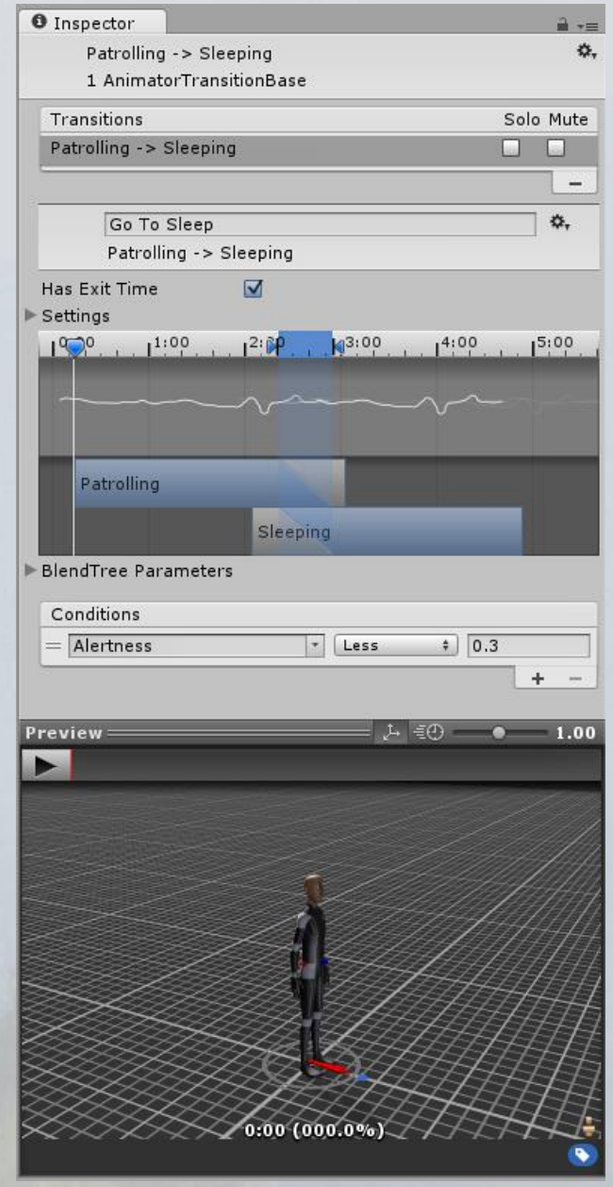
Parámetros de animación

- Variables definidas en el *Animator Controller*
- Son entradas para la máquina de estados, que pueden leerse y escribirse mediante scripts
- Tipos de Parámetros
  - Int, Float, Bool
  - Trigger - un parámetro booleano que se restablece cuando se consume por una transición
- Los parámetros se pueden modificar a partir de un script usando funciones en la clase *Animator: SetFloat, SetInt, SetBool, SetTrigger y ResetTrigger*

Animator Controller (II)

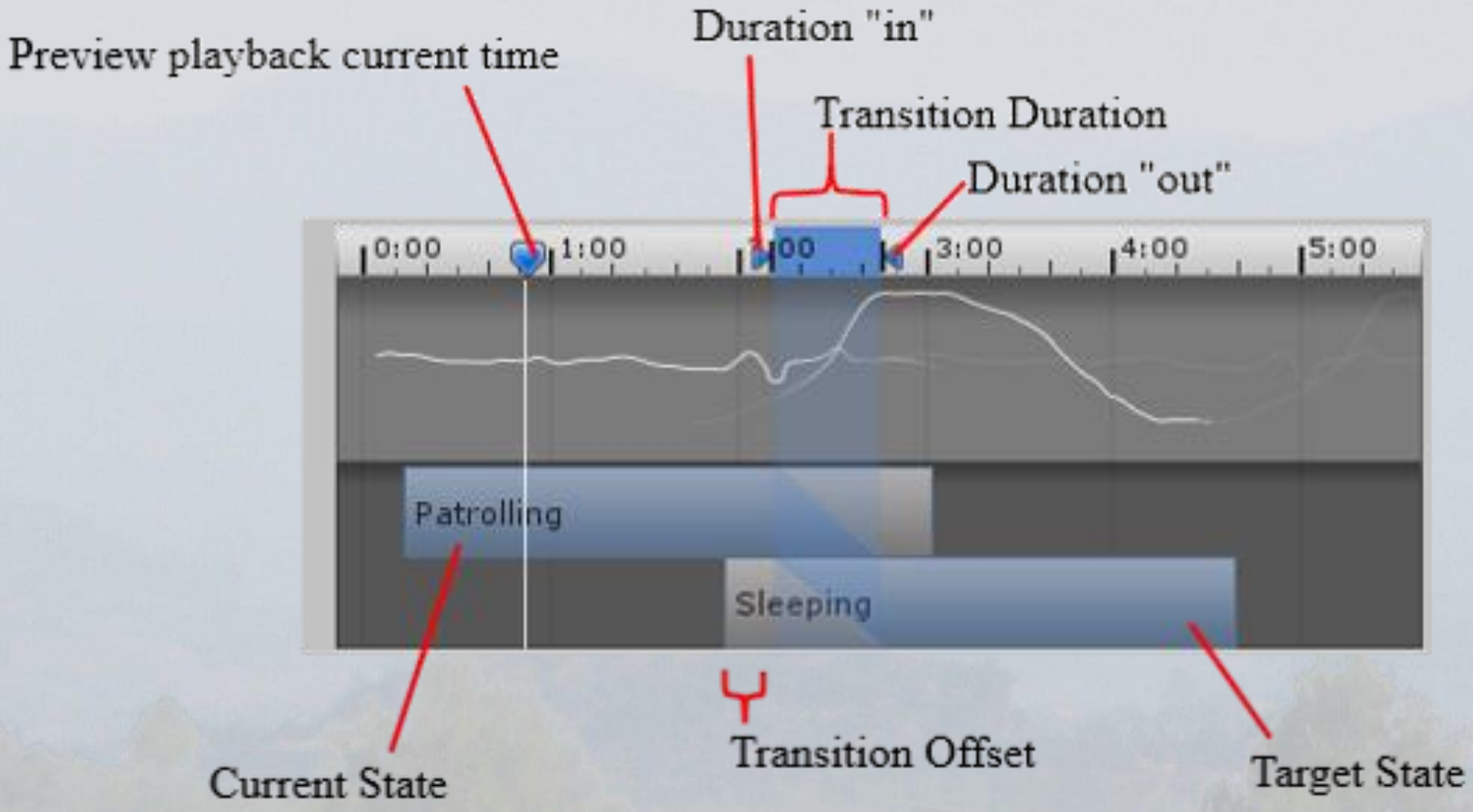
Transiciones

- Permitir que la máquina de estado cambie de un estado de animación a otro o los mezcle
- Definir cuánto tiempo debe durar la combinación entre estados y qué condiciones lo activan, según los valores de los parámetros del *Animator Controller*
  - Tiene tiempo de salida. Determina si la condición de la transición puede tener efecto en cualquier momento, o solo durante el tiempo de salida del estado
  - Tiempo de salida: porcentaje de tiempo (0..1) que debe pasar para permitir la transición si se cumplen las condiciones



Animator Controller (III)

Transiciones

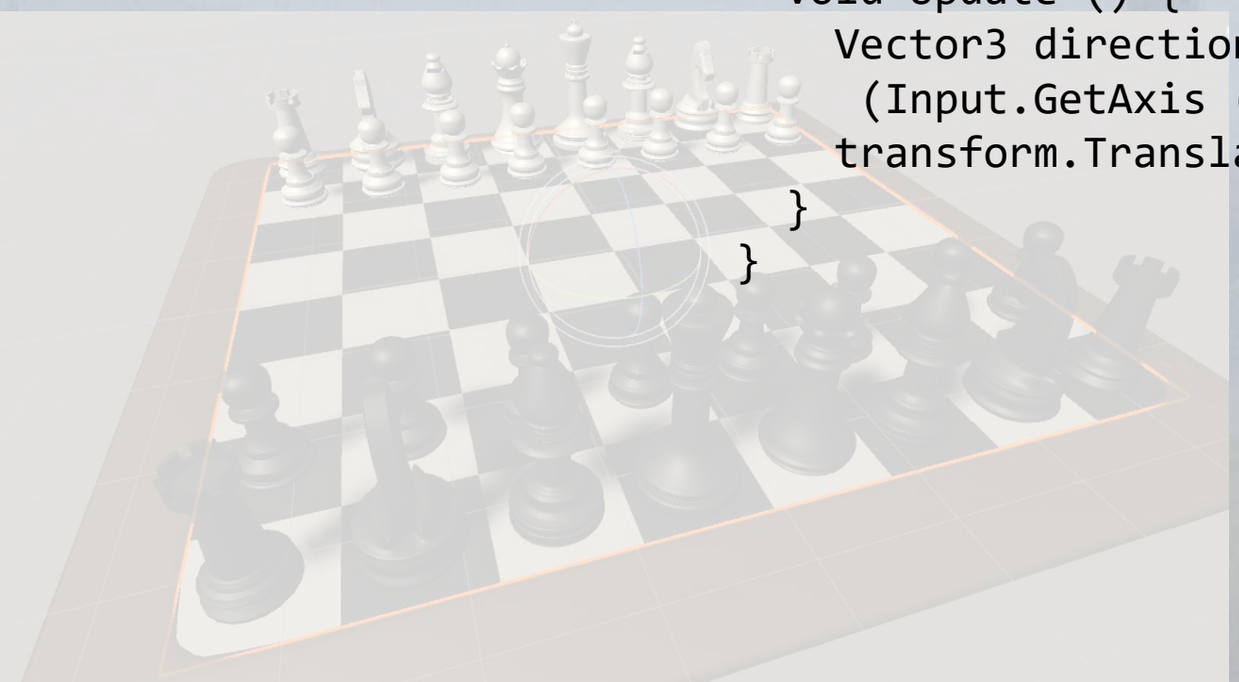


Una Animación Reactiva (I)

A partir del proyecto de ajedrez, cargar la escena chess.fbx

- Seleccionar al rey blanco
- Etiquetarlo como Jugador
- Añadirle un componente RigidBody
- Ajustar la cámara como una hija del rey y situarla para que presente un adecuado punto de vista de la escena

Generar el siguiente script

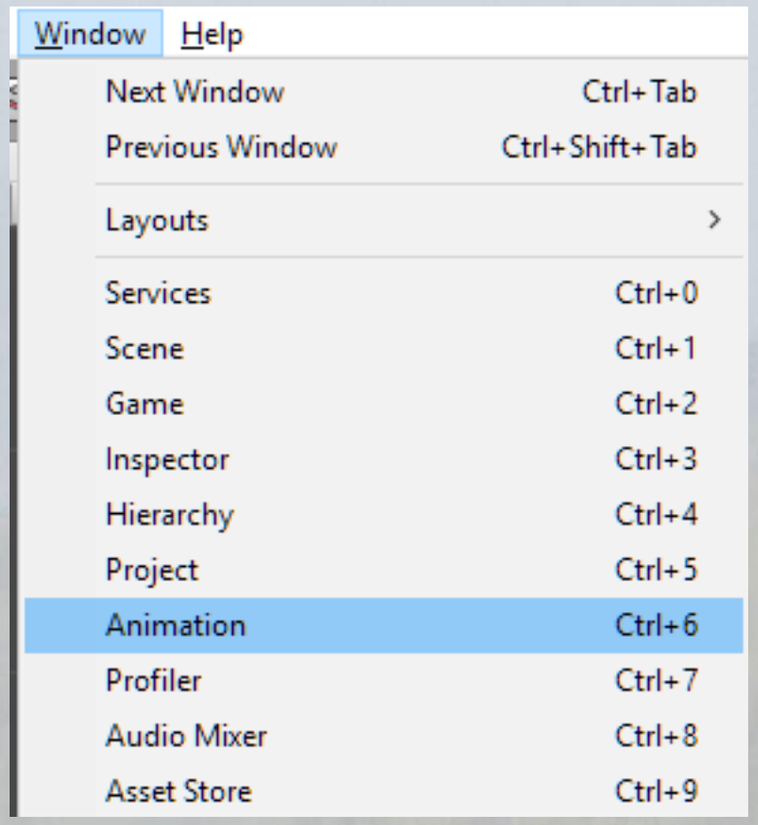


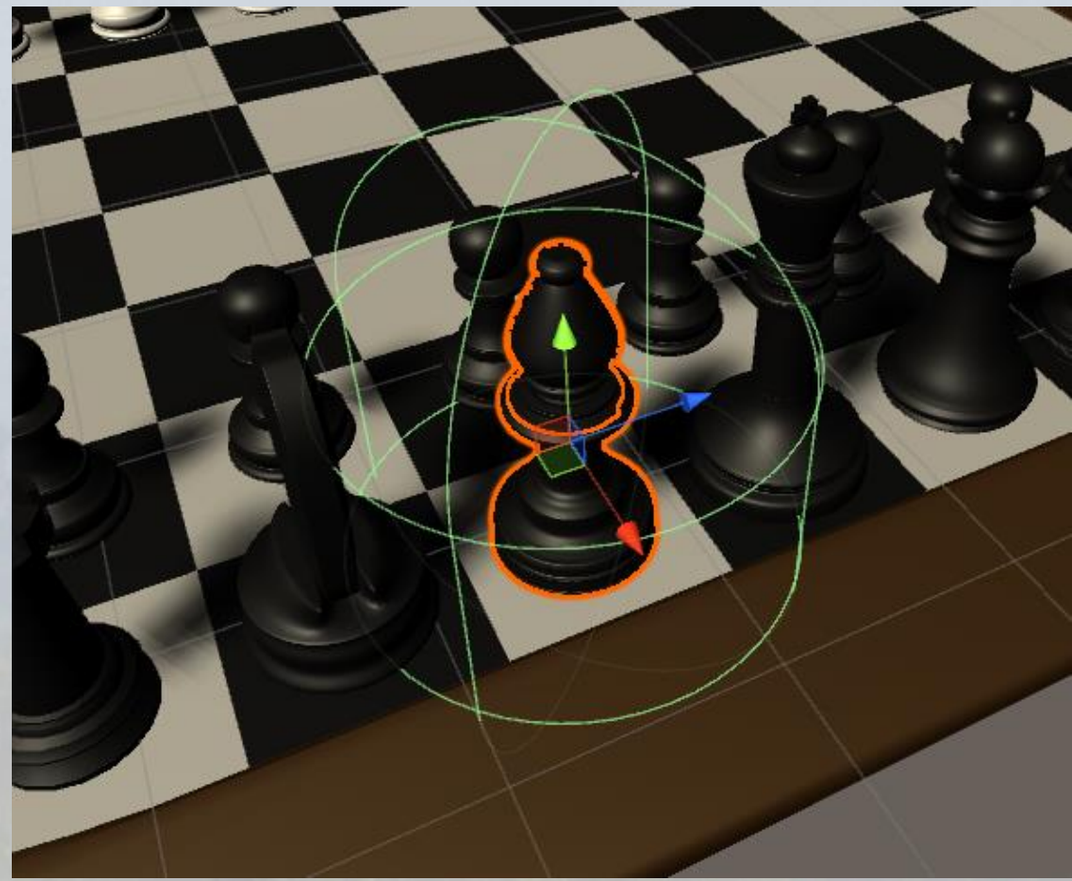
```
public class Move : MonoBehaviour {
  public float speed = 3.0f;
  void Update () {
    Vector3 direction = new Vector3
     (Input.GetAxis ("Horizontal"), 0.0f, Input.GetAxis ("Vertical"));
    transform.Translate (direction.normalized * speed * Time.deltaTime);
  }
}
```

Una Animación Reactiva (II)

Seleccionar al alfil negro izquierdo

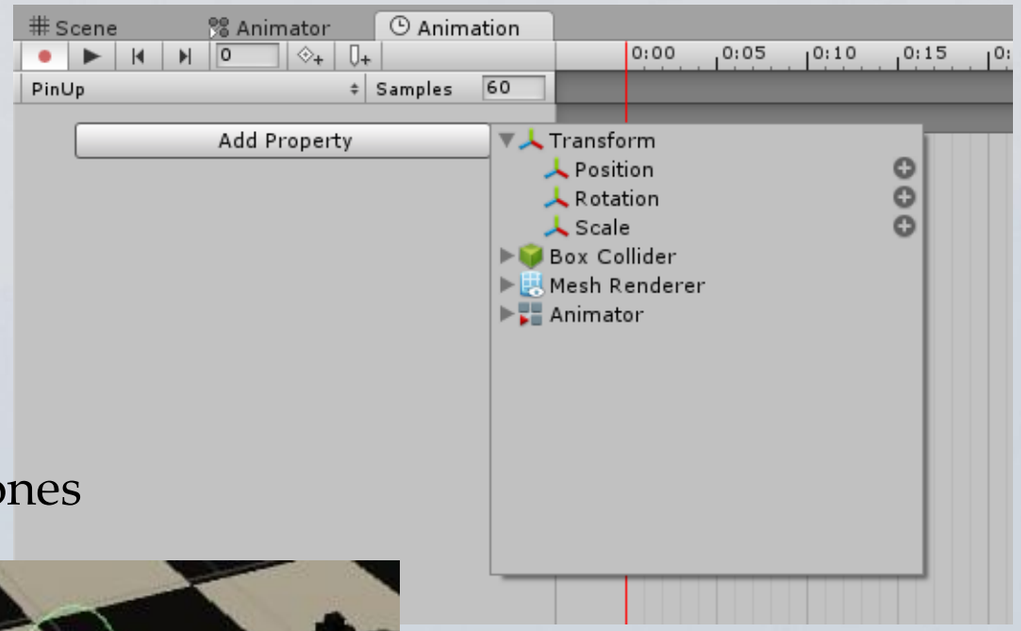
- Añadirle un colisionador de tipo cápsula y aumentar su tamaño más allá de su malla para que detecte una colsión en cuanto el rey blanco se acerque
- Activar la propiedad "Is Trigger" del Box Collider porque se desea invocar un script tan pronto como el jugador entre en contacto con el colisionador
- Crear una animación abriendo la ventana de animación (Window\Animation) o Ctrl+6

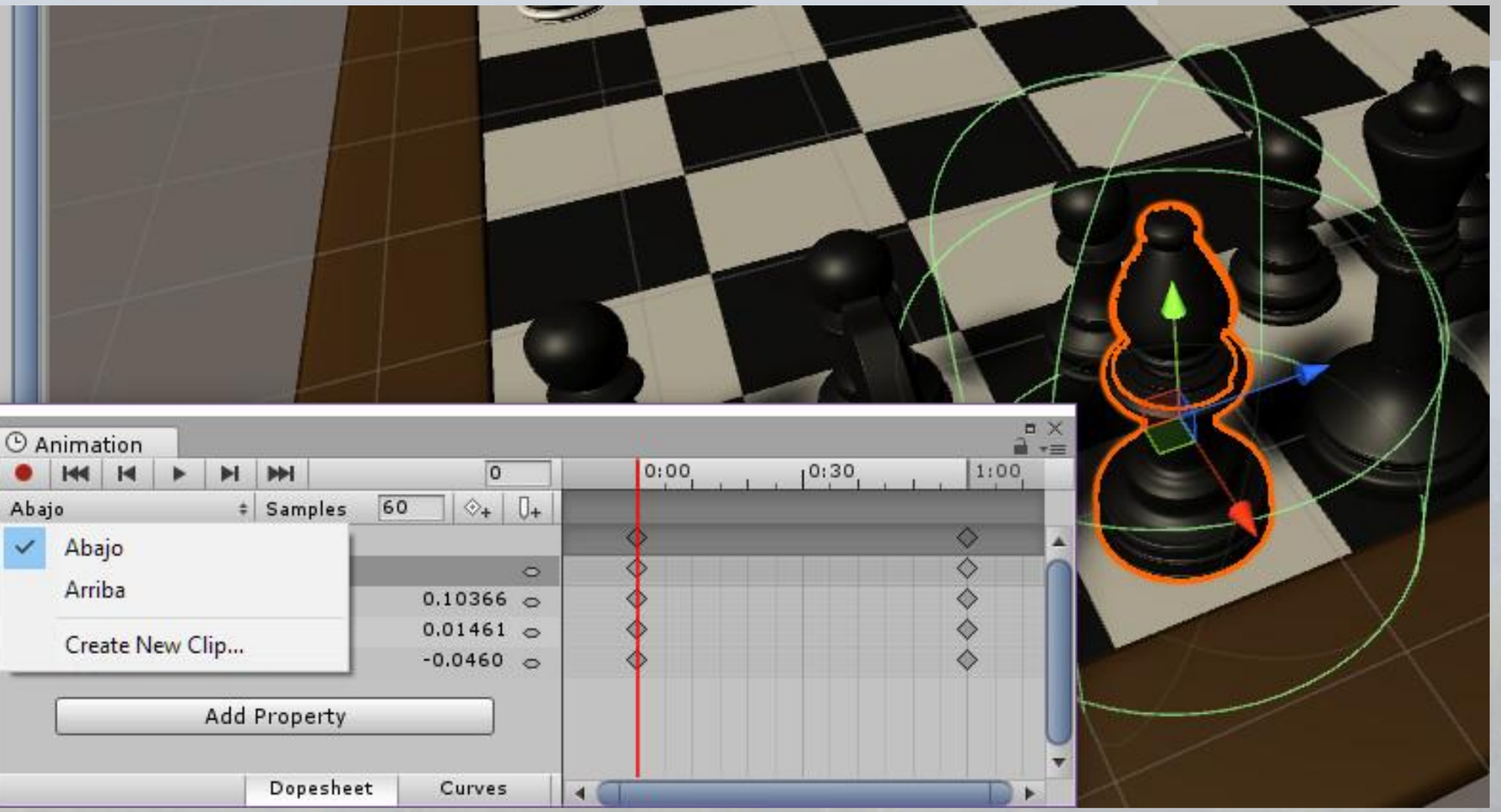




Una Animación Reactiva (III)

- Crear dos clips de animación
  - Arriba y abajo
  - Agregar la propiedad Posicion para la animación
- Eliminar el último fotograma clave
  - Definir sólo una posición por clip: arriba y abajo
  - Unity interpolará entre las posiciones en las transiciones





Una Animación Reactiva (IV)

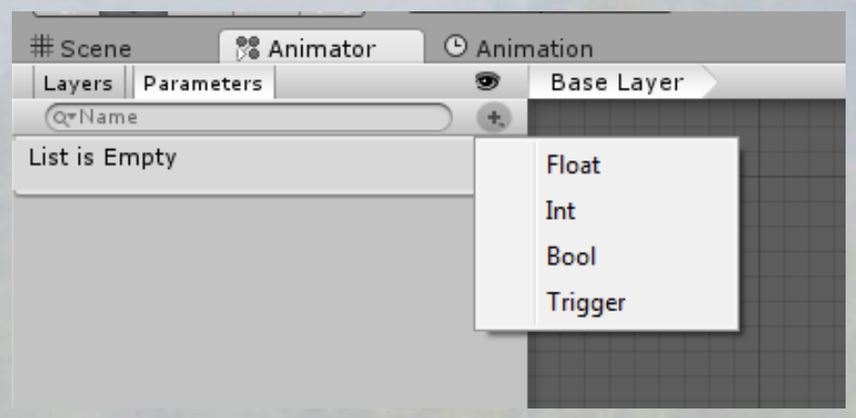
Abrir el controlador de animación. Debería mostrar ambos clips

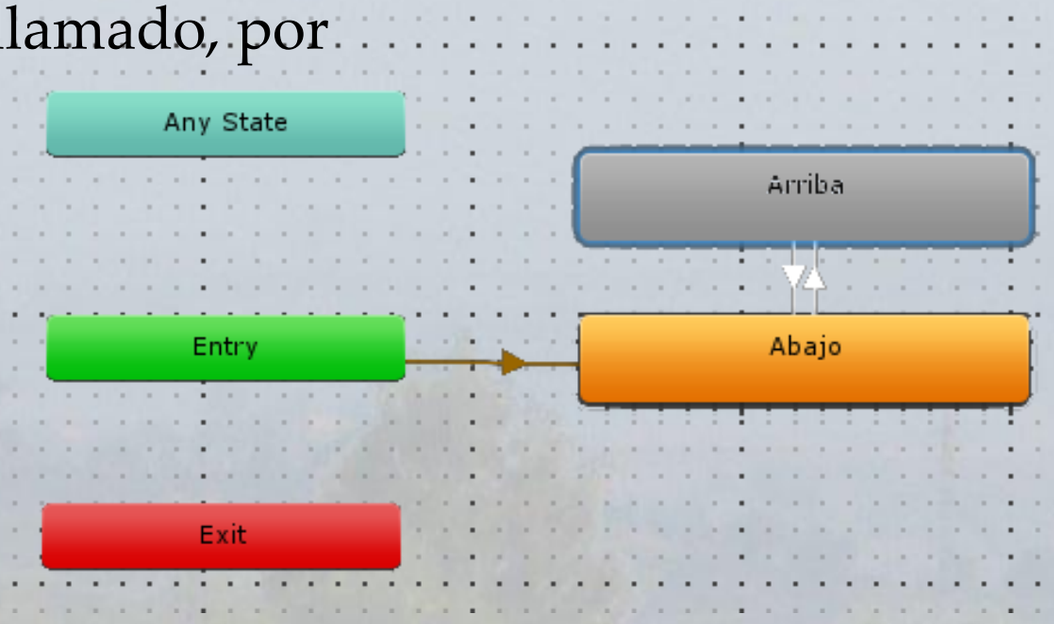
- El clip Abajo debe ser el predeterminado (marcado en naranja)
- Si no, hacer clic con el botón derecho y seleccionar "Establecer como estado predeterminado de capa" en el menú emergente

Crear dos transiciones de estado *Arriba* a estado *Abajo*, y viceversa

- Hacer clic con el botón derecho en estado origen
- Seleccionar hacer transición en menú emergente
- Seleccionar el destino

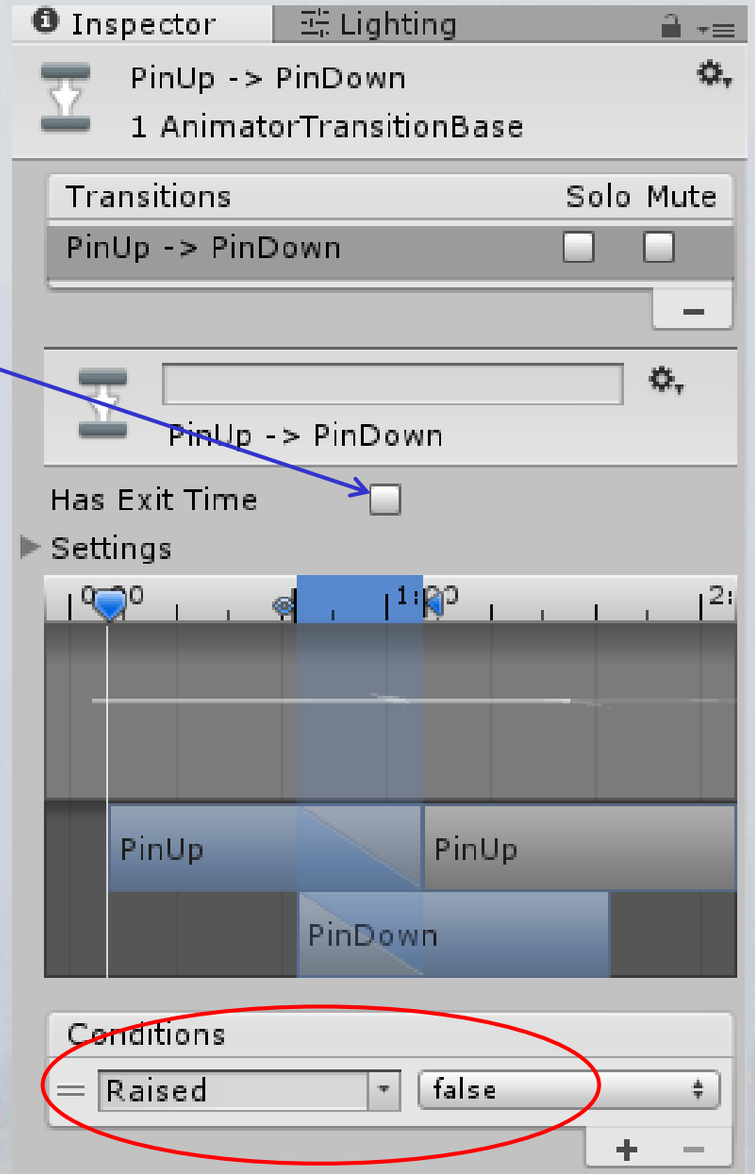
Crear un parámetro booleano en la ventana *Animator* llamado, por ejemplo, *Elevado*





Una Animación Reactiva (V)

- Hacer que cada transición dependa del valor apropiado del parámetro
  - Desmarcar la casilla de verificación *Has Exit Time*
  - Agregar una nueva condición (*Elevado* es verdadero o falso, dependiendo de la transición)
- Ajustar la duración de la transición usando las flechas azules en la línea de tiempo
- Comprobar la propiedad *Elevado* en el *Animator*, ya que el álfil debería estar normalmente abajo, salvo cuando se acerque el rey enemigo



Una Animación Reactiva (VI)

Ahora hay que cambiar el valor del parámetro *Elevado* para iniciar las transiciones

Cambiar el valor del parámetro usando un Script en el álfil, siempre que el jugador entre o salga de su Box Collider

Crear un script y asignarlo al álfil

```
using UnityEngine;
using System.Collections;
public class Alfil : MonoBehaviour {
  private Animator anim;

  void Awake () {
    anim = GetComponent<Animator> ();
  }
  void OnTriggerEnter(Collider other) {
    if (other.gameObject.tag == "Player")
      anim.SetBool (“Elevado", true);
   }
  void OnTriggerExit(Collider other) {
    if (other.gameObject.tag == "Player")
      anim.SetBool ("Elevado", false);
   }
}
```

Una Animación Reactiva (VII)

El álfil ahora debería ocultarse cuando la esfera se acerca, y subir automáticamente cuando la esfera se va

Preparar un prefab para hacer más álfiles

Envolver el álfil en un Empty (así la animación se aplicará con respecto a la posición definida en Empty)

Crear una carpeta Prefab en la ventana de proyecto y arrastrar el empty

Crear tantas instancias del prefab como sea necesario

Cada uno debe reaccionar de manera autónoma

Bibliografía

Manuales en línea de Unity 3D
http://docs.unity3d.com/Manual/AnimationOverview.html

INTRODUCCIÓN A LA PROGRAMACIÓN DE VÍDEOJUEGOS

Documentación generada por
Dr. Ramón Mollá Vayá
Sección de Informática Gráfica
Departamento de Sistemas Informáticos y Computación
Universidad Politécnica de Valencia