PEC 3 – Animación de robots

Joaquín López Soriano

Media para videojuegos

Universidad Oberta de Cataluña

Contenido

[Resumen de las entregas 3](#_Toc156985084)

[Visión de futuro 4](#_Toc156985085)

[Explicación sobre los Animator controllers 5](#_Toc156985086)

[Resumen de la creación de las animaciones 6](#_Toc156985087)

[Bibliografía 7](#_Toc156985088)

[Software 7](#_Toc156985089)

[Exportar animaciones 7](#_Toc156985090)

[Realizar animaciones con Maya 7](#_Toc156985091)

[Unity Asset 7](#_Toc156985092)

[Movimiento y rotación 7](#_Toc156985093)

[Terminología 7](#_Toc156985094)

# Resumen de las entregas

El desarrollo incremental de las diferentes prácticas para entender como funciona el proceso de creación y animación de personajes 3D ha sido adecuado.

La evolución desde una escena vacía en Maya hasta ver tus personajes animados interactuando resulta reconfortante, más todavía tras todos los problemas superados durante cada práctica.

Respecto a esta práctica, para animar a cada robot se ha continuado usando la aplicación [Maya](https://www.autodesk.es/products/maya/overview?term=1-YEAR&tab=subscription).  
Al principio, la realización de la animación ha resultado complicado hasta entender como se realiza; ubicar el marcador de frame en un frame, modificar las propiedades de las articulaciones del esqueleto y grabar el nuevo estado con la tecla *‘S’*.

El animar los robots no ha sido una tarea fácil porque al mover las articulaciones dejaban huecos entre los elementos que forman el objeto 3D u obtenido un objeto 3D no muy verosímil, restringiendo y delimitando la animación de los mismos.

Para realizar el proyecto, se ha recreado el escenario de una ciudad donde dos [mecha](https://es.wikipedia.org/wiki/Mecha_(ciencia_ficci%C3%B3n)) luchan.  
Para ello, se ha usado el asset de Unity [CITY package](https://assetstore.unity.com/packages/3d/environments/urban/city-package-107224), que ofrece gran cantidad de objetos para construir una cuidad.

Tras ello, se ha implementado el [animator controller](#_Explicación_sobre_los), explicando en un punto posterior.

Por último, se ha definido el movimiento de cada robot. Para ello, se ha usado el componente *Character controller*, para definir el movimiento y rotación de cada robot.  
Para la captura de teclado, se ha usado el *asset* de *Unity Input System*.

Para su implementación, se ha programado el script *PlayerController*.

# Visión de futuro

Realizar estas tareas a comprender mejor la visión sobre el funcionamiento de la animación tanto en videojuegos como en películas y entender el trabajo que existe detrás de trabajos como el de [diseñadores gráficos](https://www.unir.net/ingenieria/revista/disenador-3d/), [riggers](https://www.notodoanimacion.es/que-es-y-como-hacer-un-buen-rigging/) o [animadores gráficos](https://www.unrealengine.com/es-ES/explainers/animation/what-is-a-3d-animator).

Respecto al proyecto, es un punto de partida para continuar desarrollando la aplicación y convertirlo en un juego del género de [lucha](https://es.wikipedia.org/wiki/Videojuego_de_lucha) o en uno [Beat 'em up](https://es.wikipedia.org/wiki/Videojuegos_Beat_%27em_up) individual o cooperativo.

# Explicación sobre los Animator controllers

Para realizar la animación de los robots, se han definido 3 estados en la máquina de estados finita:

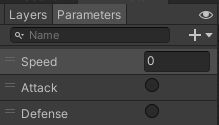
* *Locomotion*: árbol de mezclas (blend tree), donde según el valor de la velocidad, (speed), se ejecuta una animación.
* *Attack*: animación de ataque
* *Defense*: animación de defensa.

La estructura de la máquina de estados del controlador de animación es la siguiente:

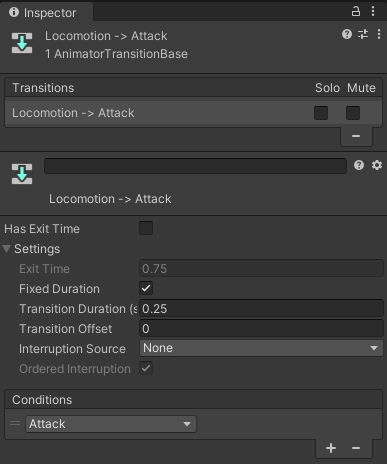
|  |  |
| --- | --- |
|  |  |
| *Controlador de animación Robot1* | *Controlador de animación Robot1* |

Se ha usado un controlador con una estructura diferente para cada uno para comprobar que de las dos formas se consigue el mismo resultado. La diferencia estriba en el hecho de que en el primero controlador, para pasar a la animación de ataque o defensa, se ha de estar antes en la animación de locomoción, pero en la segunda, da igual en que animación se éste para pasar a la animación de ataque o defensa.

DEFINIR EL BLEND TREE

La forma de pasar a la animación de ataque o defensa se realiza mediante un disparador o trigger. Se ha definido un disparador para cada animación:

* Attack: para la animación de ataque.
* Defense: para la animación de defensa.

La condición de cada transición que termina en la animación *locomotion* es la siguiente, tanto para la animación de ataque o defensa, con la salvedad de cambiar el valor del disparador.

Se ha de deseleccionar la opción de *has exit time* para que se ejecute en el momento en que se activa el disparador y en *conditions*, definir que se ha de realizar la transición con el disparador adecuado, en este caso el asociado con la animación de ataque.

# Resumen de la creación de las animaciones

Las animaciones de los robots se han intentando hacer que se parezcan a movimientos similares a los humanos, además de armónicos y natural, pero como se ha explicado en el punto [Resumen de las entregas](#_Resumen_de_las), las limitaciones técnicas han impedido realizar unas mejores animaciones.

Se han creado 4 animaciones para cada robot, una para cada acción:

* *Parado*: cuando el robot no realiza ninguna acción
* *Movimiento*: cuando el robot se mueve.
* *Ataque*: cuando el robot realiza un movimiento de ataque.
* *Defensa*: cuando el robot realiza un movimiento de defensa.

En el primer robot, el que tiene orugas por piernas, solo se ha animado el tronco superior, brazos y cabeza.

El segundo robot, el arácnido, se ha buscado que el movimiento de las patas sea similar al de una araña. Este robot ha constado más animarlo debido a la complejidad de sincronizar el movimiento de cada par de patas y que de forma global el movimiento resultase fluido.

# Bibliografía

## Software

* Maya: [https://www.autodesk.es/products/maya](https://www.autodesk.es/products/maya/)
* Unity: <https://unity.com/es>

## Exportar animaciones

* <https://www.youtube.com/watch?v=KiSZ6J0VqS8>
* <https://www.youtube.com/watch?v=i8cmDK_O-1g>

## Realizar animaciones con Maya

* <https://www.youtube.com/watch?v=1wvdQy2Fdhw>
* <https://www.youtube.com/watch?v=-KcwtJSAz4g>
* <https://www.youtube.com/watch?v=3UR9r-tb6KQ>

## Unity Asset

* <https://assetstore.unity.com/packages/3d/environments/urban/city-package-107224>

## Movimiento y rotación

* <https://docs.unity3d.com/ScriptReference/CharacterController.html>
* <https://unity.com/es/features/input-system>
* <https://www.youtube.com/watch?v=BJzYGsMcy8Q>
* <https://www.youtube.com/watch?v=UJsaEVPntMg>

## Terminología

* Género beat em up: <https://es.wikipedia.org/wiki/Videojuego_de_beat_%27em_up>
* Género de lucha: <https://es.wikipedia.org/wiki/Videojuego_de_lucha>
* Definición de rigger: <https://www.notodoanimacion.es/que-es-y-como-hacer-un-buen-rigging/>
* Definición de diseñador gráfico[: https://www.unir.net/ingenieria/revista/disenador-3d/](:%20https:/www.unir.net/ingenieria/revista/disenador-3d/)
* Definición de animador gráfico: <https://www.unrealengine.com/es-ES/explainers/animation/what-is-a-3d-animator>
* Definición mecha: <https://es.wikipedia.org/wiki/Mecha_(ciencia_ficci%C3%B3n)>