Imagen que contiene primate, mamífero, perro, agua

Descripción generada automáticamenteDocumentación DEV: Project Western

Alex.-

Contenido

[Sistema de Apuntado 1](#_Toc162609950)

[Sistema de Equipar Armas 4](#_Toc162609951)

[Animaciones 5](#_Toc162609952)

# Sistema de Apuntado

Imagen que contiene interior, con baldosas, pequeño, computadora

Descripción generada automáticamente

Imagen de la pantalla de un video juego

Descripción generada automáticamente con confianza media

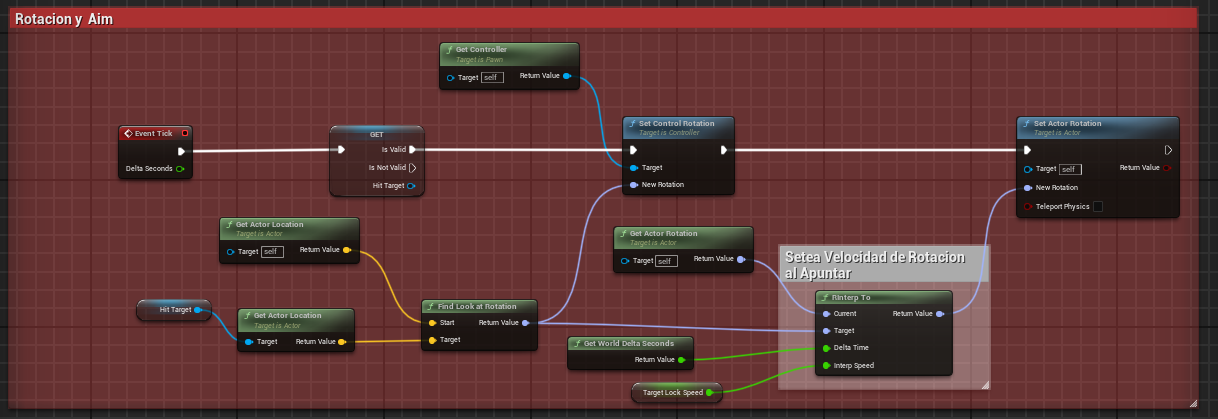
Dentro del blueprint del Player principal: BP\_ThirdPersonCharacter ubicado en Content\ThirdPerson\Blueprints\Player se crearon dos apartados de nodos para lo que es el sistema de apuntado. El primero “Raycast al enemigo”. Es el calculo del rayo hacia un objeto especifico. Esto se ejecuta cuando se realiza la acción de evento “IA\_Aim” (se ejecuta al presionar clic derecho del mouse). Cuando esto sucede, se dispara un rayo de 9 metros (900) con un radio de 1.25m (125). Si detecta un objeto y este concuerda con el filtro de “Pawns” devuelve el objeto y nos da una opción de tratamiento para cada caso. Por el momento nos interesa pegar generalmente con el personaje así que el bucle finaliza con “Hit Actor” y esto va a una variable de método llamada “Hit Target”. Por otro lado, si se deja de accionar el action evento (Completed), se llama a la variable “Hit Target” para vaciarla, es decir para que deje de apuntar.

Escala de tiempo

Descripción generada automáticamente con confianza bajaCaptura de pantalla de un celular

Descripción generada automáticamente

Estas variables se pueden llamar in-game para modificarlas y probar *cositas*. Ejecutado el juego, tienen que ir al componente del personaje y buscar “Aim”.



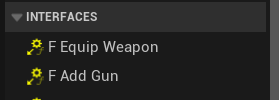
Una vez que se apunte al target, también nos interesa que el personaje rote hacia él. Es decir que lo mire a la cara. Para esto realizamos un eventtick cada 5 segundos, tiempo razonable para cambiar entre targets. Es decir, si el personaje toma un nuevo target va a rotar hacia el cada ese lapso. Se probo inGame y funciona correctamente. Se deja variable también para poder modificar si se requiere (TargetLockSpeed).

# Sistema de Equipar Armas

Pantalla de un video juego

Descripción generada automáticamente con confianza media

También dentro del blueprint del personaje, se encuentra esta sección de nodos en la cual se crearon dos eventos custom para equipar y desequipar arma. Estos eventos están asociados a una interfaz que se llama y se utiliza tipo inventario. Se crea un bool para las condiciones de tener o no el arma. En el caso del evento EquipWeapon, se toma por referencia una variable “EquippedWeapon” que según lo que traiga dentro va a devolver las condiciones tener o no arma. Esta variable la veremos a continuación en la parte de animaciones.



Las interfaces para almacenar y alojar armas entre blueprints.

### Animaciones

Pantalla de un video juego

Descripción generada automáticamente con confianza media

La forma de acceder a animaciones es posicionarnos sobre la raíz del player y luego hacer doble clic en el dibujo del MESH del Player.

Captura de pantalla de un video juego

Descripción generada automáticamente con confianza media

Una vez dentro tendremos todas las funcionalidades de las animaciones. En este caso primeramente fuimos a la parte de esqueleto del Player y luego en el hueso de la mano derecha, agregamos un socket e incrustamos el skeletal mesh del rifle que viene por defecto.

Pantalla de juego de computadora

Descripción generada automáticamente con confianza media

Una vez hecho esto, Damos clic en los nodos de arriba a la derecha para acceder en “Locomotion State Machine” que guarda las animaciones principales.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Ingresamos dentro de Idle.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Y por el momento creamos dos estados de animaciones. Idle, que ya viene por defecto y Idle\_Pistol el cual agregamos como animación y posicionamos el arma en las manos. Esta animación venia en el paquete de inicio. A su vez se va a tomar que numero de equipamiento fue accionado por el player. Como veremos en el próximo cuadro, se creo una variable para guardar en el caso de 0 desequipado y en el caso de 1 la pistola.

Imagen que contiene sostener, mujer, jugador, agua

Descripción generada automáticamente Imagen que contiene grande, sostener, pelota, agua

Descripción generada automáticamente

Captura de pantalla de computadora

Descripción generada automáticamente

En el animgraph se creó este custom evento que setea si el personaje esta desarmado o armado.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Volviendo al blueprint del personaje, se creo un pequeño Branch el cual cuando estamos apuntando bloquea el movimiento del personaje y al dejar de apuntar podemos movernos nuevamente.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Finalmente, en el blueprint del player controller tenemos las funciones para equipar o desequipar y que pase los respectivos valores para el apartado de animaciones.