

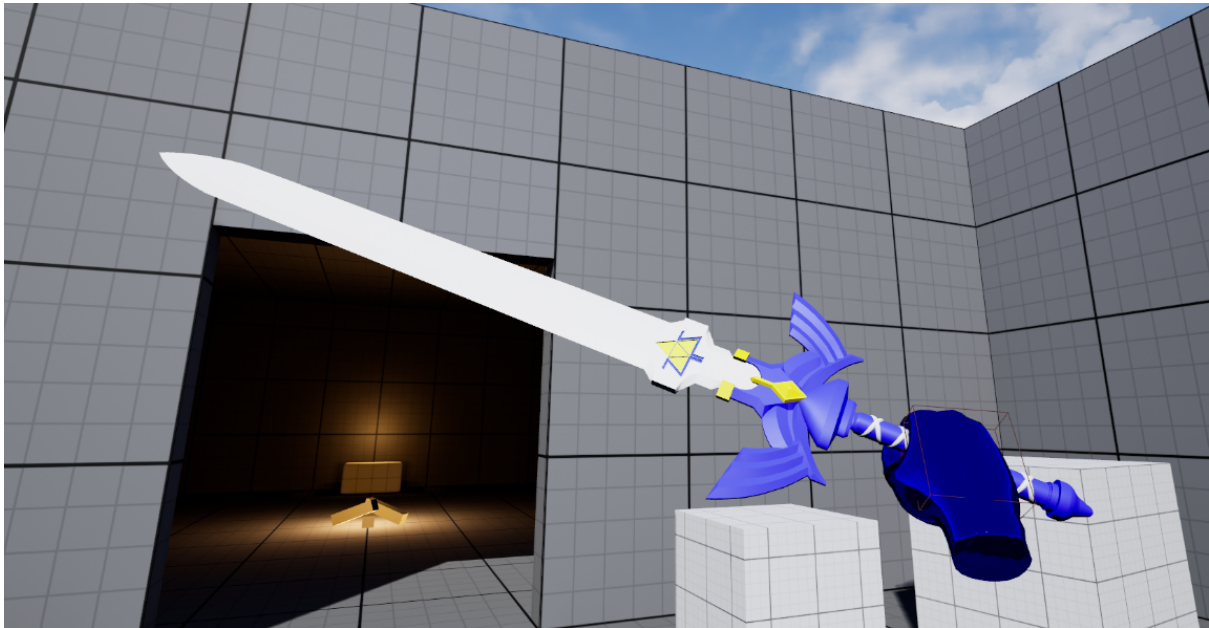
## InteracciónVR

- Pulsar botones:
  - Primero, se usa de base el botón pulsable de la plantilla de Oculus, PushButtonBaseBP. Este tiene funcionalidad en su interior establecida para reaccionar ante el contacto de una colisión, activando su tick que permite ser empujado hacia abajo una cantidad de profundidad definida.
  - Para modificar sus valores se ha hecho un Blueprint hijo llamado PushButton\_BP, que se ocupa de darle una forma con un Static Mesh, los sonidos que tiene al pulsarse, cuánto puede pulsarse... para ser colocado en la escena.
  - Se puede extender más allá su funcionalidad, utilizando una Blueprint hija de PushButton\_BP. En este caso se llama ToggleButton\_BP, que utiliza el Event Dispatcher OnButtonPress de PushButtonBaseBP asignándole un evento que cambia su material según se pulse.
  - Este último evento se lanza a través del evento ButtonPressed de PushButtonBaseBP. Cuando el tick es activado, este se lanza si el botón ha sido pulsado lo suficiente.



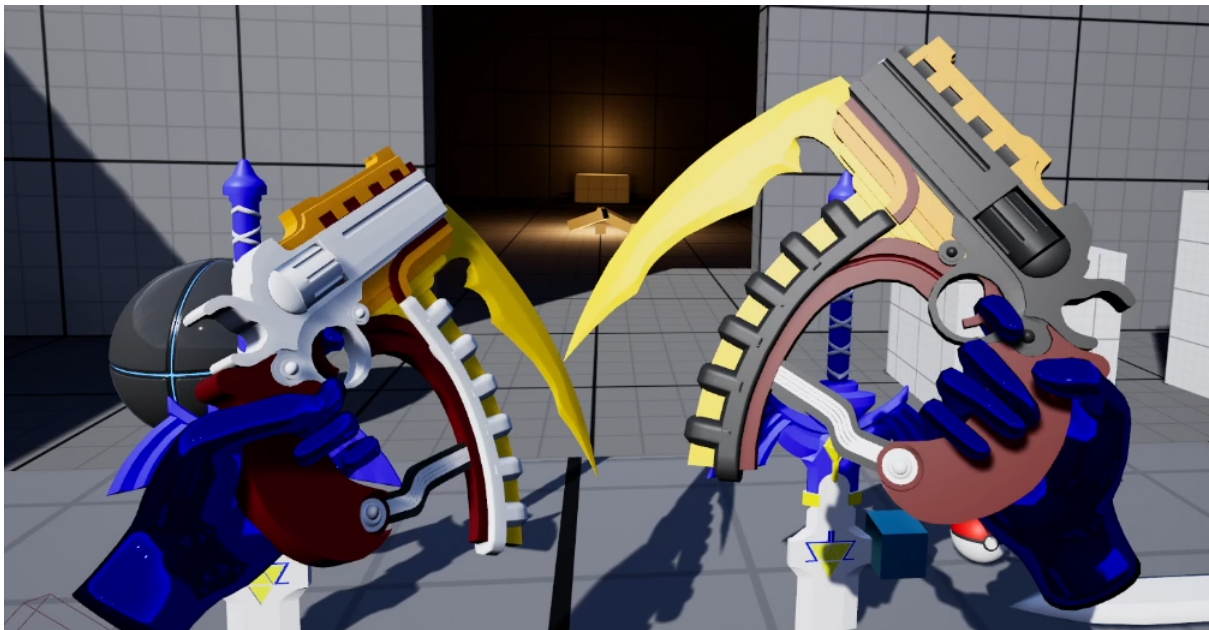
- Reconocer gestos (vídeo de muestra: <https://youtu.be/wfS-Nlf1iAY>)
  - Primero de todo, se crea una Blueprint extendida del Oculus Hand Component para modificar su comportamiento. Esta trabaja varias funciones:
    - Load Gestures: lee el archivo de guardado con los gestos guardados. Si no existe, se crea.
    - Record Gesture: guarda el gesto que tenga la mano en el archivo de guardado pulsando el botón R o L según la mano usada. Si ese gesto ya existía, no hace nada.
    - Does Gesture Exist: comprueba si el gesto existe en el archivo de guardado según el nombre.
    - Delete Gesture: Borra el gesto indicado por nombre del archivo de guardado.
    - Get Current Gesture: Lee la posición de los huesos de las manos del Hand-Tracking a través de un evento que se lanza cada 0,1 segundos. Si coincide con una de las guardadas en un Threshold de 3 cm de diferencia, acepta la posición como válida.
    - Equal Gesture: compara dos SGesture entre sí de forma directa.
  - Para hacer funcionar a los gestos, existe la Struct SGesture, que contiene un nombre y un array de vectores indicando la posición de cada uno de los huesos de la mano.
  - Así mismo, se ha cambiado el SaveGame por defecto del Game Mode para que acomode un Array de SGesture.

- Cuando el gesto coincide, se lanza el Event Dispatcher On Gesture Recognized. Este es asignado en MagicHands\_BP, que es el Pawn del jugador. En esta Blueprint se elige un efecto a crear en el entorno según la String de entrada desde el Event Dispatcher.
- Ese efecto es lanzado sobre el Effect\_BP, una Blueprint colocada en la escena que instancia un Emitter sobre el escenario según el evento lanzado.



- Agarrar objetos (vídeo de muestra: [https://youtu.be/zZUfbl\\_wsdw](https://youtu.be/zZUfbl_wsdw))
  - El objeto base de todo esto es la Blueprint PickupCube. Este incluye un Static Mesh modificable con el objeto que quieras, aparte de proporcionar variables modificables:
    - Snap Socket: indica el nombre del socket previamente creado sobre el Skeletal Mesh de la mano que agarra el objeto, utilizándolo como posición donde se coloca.
    - Left Hand Pose/Right Hand Pose: se refiere a la pose que mantiene el la mano indicada cuando agarras el objeto.
  - Antes de poder agarrar objetos se ha extendido la clase Oculus Hand Component con BP\_OculusHand. Esta se encarga de comprobar la forma de la mano.
    - En el begin play se recogen todos los huesos de la mano para poder buscarlos más adelante.

- Primero, consideramos que para agarrar se necesita cerrar por completo el puño. Esto se realiza en el método Detect Grabbing.
- El método Detect Grabbing utiliza los huesos de la punta de los dedos para saber si está cerrada. Si el total de distancia de todos respecto al hueso de la muñeca está entre 7,8 y 16,9, eso significa que está cerrado y puede proceder a activar el Event Dispatcher OnGrab. Si por lo contrario deja de estar entre esa distancia, eso significa que ha abierto su mano y puede llamar al Event Dispatcher OnRelease.



- Para agarrar los objetos primero se han creado las dos manos por separado. Una es BP\_Hand, y la otra es BP\_RightHand, hija de BP\_Hand. Todas tienen las variables correspondientes puestas según sea la mano derecha o la izquierda, indicando al motion controller cual es la creada. Internamente, se realizan varias tareas:
  - Evento Begin Play: activa los eventos OnGrab y OnRelease de BP\_OculusHand.
  - Evento Grab: cambia la mano real por una poseable, poniendo en efecto la que el objeto agarrado utilice
  - Evento Release: suelta el objeto y restaura la función de la mano real, quitando la mano poseable de enmedio.
  - Evento OnAttachedObjectPickup: permite la transferencia de un objeto ya agarrado a otra mano.

- Evento SetHandPose: coloca la mano poseable según la indicada en el objeto agarrado.
- Evento RestoreTracking: reinicia el movimiento de la mano real de Oculus.
- Función SetPoseableMeshPose: coloca los huesos de la mano poseable de la misma forma que la mano real.
- Función Poseable&SkeletalEqual: Comprueba si la mano real y la poseable están colocadas de la misma forma en base a las puntas de los dedos.
- Función GetSkeletalMeshBoneTransforms: obtiene los huesos de la mano real para compararlos con los poseables.
- Función InterpolatePoseableToSkeletal: cambia entre la mano poseable y la real de forma interpolada.
- Función Poseable&OculusHandEqual: comprueba si la mano poseable y la de Oculus son iguales.
- Función InterpolatePoseableToOculusHand: cambia la mano poseable a la real de forma interpolada.
- Estas manos se instancian en la Blueprint BP\_VRPawn asignando su movimiento a cada una de las manos.

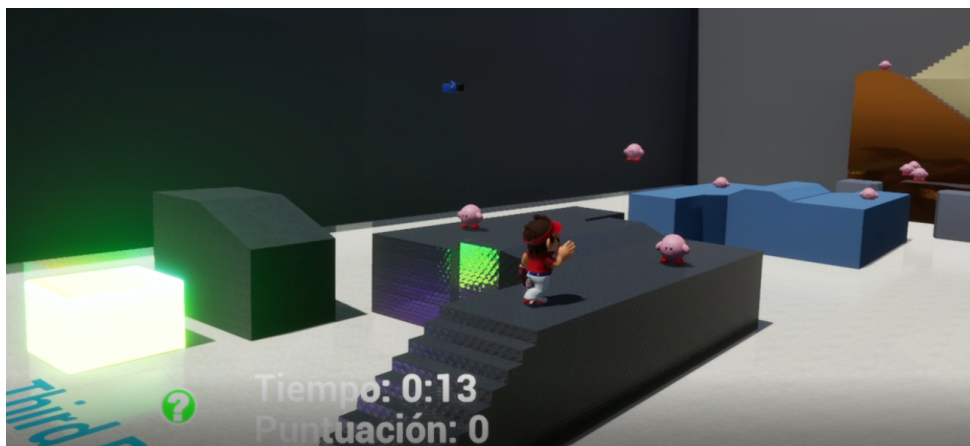


DLCs Unreal (vídeo de muestra: <https://youtu.be/qsnLW1ZbrGQ>)

- El juego empieza con la Blueprint de widget Interfaz Inicial, donde primero de todo se crea un array con los niveles y luego comprueba si el jugador tiene los DLCs, en caso de que sí, activa los elementos seleccionables del DLC



- Cada uno de los checkbox tiene su estado bindeado al mismo evento, Set Check Box, y después asigna la variable Selected Level a el número correspondiente.
- Cuando pulsas el botón iniciar, esos datos deciden el personaje que se va a usar establecido en el GamelInstance, y luego establece el color del suelo. Una vez hecho, abre el nivel elegido
- SetCheckBox elige según un parámetro de String qué grupo de checkbox debe modificar, pero en general desactiva todos y activa el seleccionado
- Al cargar uno de los niveles, es imprescindible activar el input para el jugador, que había sido quitado en la interfaz anterior.
  - Es importante también cambiar el gamemode a uno custom, para poder modificar el personaje jugable.
  - Después, se comienza a buscar el player start del nivel para obtener la ubicación de inicio, y junto al GamelInstance donde se encontraban los valores de la pantalla anterior, se spawnnea el character seleccionado.
  - Cuando eso termina, se obtienen todos los static mesh y se cambia su material en base al equipo seleccionado. Para evitar desactivar el vídeo, se comprueba si es un vídeo por cada uno.
- El evento de Spawn character es una función del Player Controller customizado al poner el PruebasDLCsGameMode\_BP como gamemode. Su función es spawnear el actor indicado y poseerlo para que sea controlable



PlataformasVR (vídeo de muestra: [https://youtu.be/OGCLB\\_VduM4](https://youtu.be/OGCLB_VduM4))

- BP Interface
  - Funcionalidad: la Blueprint funciona de interfaz para el jugador dando información sobre los puntos obtenidos y el tiempo transcurrido de la partida.

- Al construirse se obtiene el player controller para tenerlo a mano más adelante.
- Tiempo marca el tiempo transcurrido, tiene el texto bindeado a una función llamada GetTime la cual usa el tiempo de juego en segundos para obtener los segundos y los minutos, formateando la frase a mm:ss
- Puntuación marca los puntos obtenidos, leyendo los puntos del Personaje cuando cambian.
- BP Goal
  - Funcionalidad: utiliza una hitbox para comprobar si el jugador llega, marcando el final del nivel
  - Al iniciar el overlap con un objeto, comprueba si es el jugador casteando al tipo Twinblast, si lo consigue recarga el nivel
- BP Kirby Collectible
  - Funcionalidad: aparece como coleccionable en el entorno. Sirve de objetivo a recoger por el jugador
  - Al iniciar el overlap con un objeto, comprueba si es el jugador casteando al tipo Twinblast, si lo consigue suma a los puntos actuales 100 puntos más y los settea
- BP Dynamic Camera System
  - Funcionalidad: son las cámaras que el jugador puede usar para cambiar su vista.
  - Al iniciar la partida, el tracking se resetea al origen, y después comprueba si la cámara que se quiere utilizar es una cámara inicial, es decir, la que se utilizará como primera cámara aparte de activar el input para el jugador. Si no lo es, la visibilidad de la mirilla se quita hasta que se active.
  - En el tick se realiza una secuencia de procesos. Primero, se comprueba si la cámara actual usada por el jugador es la misma que ella misma. Cuando lo es:
    - Inicia a realizar tracking según sea estacionaria, si lo es, comprueba si puede rotar mirando al jugador, si puede hacerlo entonces mira directamente al jugador constantemente.
    - En caso de no poder rotar, se queda mirando fijamente con sólo el movimiento del jugador. Si no es estacionaria, sigue al jugador pero sólo en la dirección indicada en el vector

TrackPlayer de forma normalizada. En el caso de este juego, lo hace en vertical

- Segundo, realiza un raycast desde la situación de la cámara a lejos de su posición. Donde ese raycast choca con un objeto colisionable, se coloca la posición de la mirilla (esfera) y si el objeto puede castearse a BP Camera, se guarda esa cámara como a la que debe moverse al pulsar el
  - InputAction ChangeCamera.
- El InputAction ChangeCamera cuando es presionado obtiene la otra cámara a cambiar, si no es nulo cambia la cámara a ese objetivo, activa su input, activa su mirilla, quita la mirilla de la anterior y se pone a null la cámara a la que se ha cambiado.



FútbolVR (vídeo de muestra: <https://youtu.be/FrUvrGfGGJw>)

- BP Football
  - El único propósito de esta Blueprint es identificar a una pelota en el escenario. Se le ha asignado un Physics Material específico para su uso que contiene propiedades de rozamiento. El Static Mesh que utiliza lleva una colisión esférica por defecto, y en su interior lleva una colisión extra que se utiliza en el jugador para reconocer cuando está cerca de él.
- BP Dynamic Camera System
  - Evento BeginPlay: configura el control para dejarlo a las gafas VR y cambia a la cámara actual si es la Start Camera.



- Evento Tick: lo primero de todo, comprueba si está la cámara activa. Si es así, comienza a realizar un SphereCast invisible hacia delante. Si ese cast choca con otro objeto del mismo tipo, cambia su material para notificar al jugador y se lo guarda para cambiar la vista cuando se pulse el botón LB o la tecla C.
- Acción ChangeCamera: si al accionar este evento existe otra cámara registrada para cambiarse, realiza el cambio con un segundo de duración y activa su input, eliminando el material brillante y quitando la otra cámara de la anterior.
- BP Portería
  - Este objeto comprueba con una hitbox si los objetos que entran dentro son BP\_Football. Si lo son, suma un punto a la puntuación del Widget BP\_Interface
- BP MarioFutbol
  - Evento ActorBeginOverlap: comprueba si el objeto es una BP\_Football para activar la posibilidad de golpearla.
  - Evento ActorEndOverlap: comprueba si el objeto es una BP\_Football para desactivar la posibilidad de golpearla.
  - Evento Tick: primero de todo, comprueba si el jugador se está moviendo. Si no lo hace, suma fuerza a su barra de resistencia. Si se mueve, resta fuerza. En los dos casos comprueba si tiene la pelota a su alcance. En caso de que así sea, mira si puede golpearla (dato indicado en la blueprint de animación). Si lo consigue, añade fuerza a la pelota en la dirección entre la mirada del jugador y la posición de la pelota.
  - El resto de eventos son de control del personaje, por defecto de la plantilla de personajes de Paragon de Epic Games.
- BP Interface
  - Esta interfaz está presente en todas las cámaras. Informa al jugador sobre los goles que ha metido y sobre el tiempo de juego. La barra inferior es la resistencia del jugador.