



# **MÁSTER UNIVERSITARIO EN INGENIERÍA INFORMÁTICA**

## **REALIDAD VIRTUAL Y AUMENTADA**

### **Implementación de la técnica de interacción “Dual- Point World Manipulation” en un entorno de realidad virtual**

**Autores:**  
**Iván Larios Lopez**  
**José Ramón Martínez Riveiro**

**Febrero, 2021**





# **MÁSTER UNIVERSITARIO EN INGENIERÍA INFORMÁTICA**

## **REALIDAD VIRTUAL Y AUMENTADA**

### **Implementación de la técnica de interacción “Dual- Point World Manipulation” en un entorno de realidad virtual**

**Autores:**  
**Iván Larios Lopez**  
**José Ramón Martínez Riveiro**

**Febrero, 2021**



## Resumen

Gracias al avance tecnológico de la última década, los sistemas de realidad virtual han experimentado una enorme mejora. La inmersión que estos ofrecen es tan buena que sus funciones y utilidades se han expandido más allá de los videojuegos, llegando a campos como la biomedicina, la rehabilitación o la enseñanza.

Entre los sistemas de realidad virtual actuales destacan las gafas HTC VIVE (Vive, 2021) , las cuales posee una gran capacidad de inmersión en los entornos virtuales. Un concepto clave es la interacción que como usuario se puede tener con el entorno virtual. Una de las interacciones más importantes es aquella que permite al usuario moverse por el espacio digital, facilitando la exploración del entorno y aumentando el grado de inmersión.

Por consiguiente, este trabajo se centra en la implementación la técnica de interacción *Dual-Point World Manipulation*, una técnica propuesta en el libro *3D User Interfaces: Theory and Practice* (Joseph J LaViola, Krujiff, McMahan, Bowman, & Poupyrev, 2004). La premisa descrita por el libro es crear una “cuerda virtual” que el jugador puede manejar para moverse a través del espacio virtual.

Debido a que las oportunidades que la técnica ofrece, partiendo en todo momento de la premisa propuesta por el libro, se han realizado diferentes implementaciones dónde se puede observar las posibilidades que la técnica de interacción permite. Algunas de las cuales son la creación de una cuerda horizontal, una cuerda vertical, una escalera y un rocódromo. Además, debido a problemas detectados a la hora de utilizar los sistemas creados para la exploración de la interacción, se ha creado un sistema de anulación de la gravedad, activable y desactivable a través de un botón del mando.

Para una mejor exploración de la técnica propuesta y los sistemas diseñados, se ha creado un escenario y minijuego a través del motor Unreal Engine 4 (Epic Games, 2021) y el plugin SteamVR (Valve, 2021).



# ÍNDICE

<b>RESUMEN .....</b>	<b>V</b>
<b>ÍNDICE DE FIGURAS.....</b>	<b>IX</b>
<b>CAPÍTULO 1. INTRODUCCIÓN.....</b>	<b>1</b>
1.1 INTRODUCCIÓN.....	1
1.2 OBJETIVOS .....	1
<b>CAPÍTULO 2. DESARROLLO DEL SISTEMA .....</b>	<b>3</b>
2.1 ANTECEDENTES AL DESARROLLO.....	3
2.2 DESCRIPCIÓN DE DESARROLLO.....	3
2.2.1 <i>Implementación de la interacción</i> .....	3
2.2.2 <i>Movimiento y rotación de cuerdas</i> .....	7
2.2.3 <i>Control de gravedad</i> .....	9
<b>CAPÍTULO 3. DESCRIPCIÓN DEL SISTEMA .....</b>	<b>11</b>
3.1 DESCRIPCIÓN DE LA SOLUCIÓN.....	11
<b>CAPÍTULO 4. CONCLUSIONES Y TRABAJO FUTURO.....</b>	<b>15</b>
4.1 CONCLUSIONES.....	15
4.2 TRABAJOS FUTUROS .....	16
<b>BIBLIOGRAFÍA .....</b>	<b>17</b>





## ÍNDICE DE FIGURAS

FIGURA 1. <i>BLUEPRINT</i> INTERFACE BP_MOTIONCONTROLLER .....	4
FIGURA 2. MODIFICACIONES AL <i>BLUEPRINT</i> MOTIONCONTROLLERPAWN - PARTE 1.....	5
FIGURA 3. MODIFICACIONES AL <i>BLUEPRINT</i> MOTIONCONTROLLERPAWN - PARTE 2.....	5
FIGURA 4. BLUEPRINT BP_CLIMBABLECUBE – EVENTTICKER.....	6
FIGURA 5. BLUEPRINT BP_CLIMBABLECUBE – EVENTO CLIMB .....	6
FIGURA 6. BLUEPRINT BP_CLIMBABLECUBE – EVENTO LET GO .....	7
FIGURA 7. CREACIÓN DE CUERDA EN BASE A LA MIRADA DEL JUGADOR .....	8
FIGURA 8. LÓGICA PARA EL MOVIMIENTO Y ROTACIÓN DE LA CUERDA HORIZONTAL .....	8
FIGURA 9. LÓGICA PARA EL MOVIMIENTO Y ROTACIÓN DE LA CUERDA VERTICAL.....	9
FIGURA 10. LÓGICA PARA ANULAR LA GRAVEDAD.....	9
FIGURA 11. PANTALLA INICIAL DE JUEGO .....	12
FIGURA 12. PANTALLA DEL JUEGO - ESCALERA VERTICAL.....	13
FIGURA 13. PANTALLA FINAL DEL JUEGO .....	13



# CAPÍTULO 1. INTRODUCCIÓN

En este capítulo se realiza una breve introducción del proyecto y los objetivos que se buscan en este.

## 1.1 Introducción

El campo de la realidad virtual es un área sobre el que se ha realizado un gran avance tanto en tecnología como en técnica durante la última década. Avances en sistemas de tracking, motores de realidad virtual y otros han permitido que los usuarios puedan experimentar con nuevas técnicas de interacción en el mundo virtual, desde el mundo real.

En este trabajo se ha desarrollado en profundidad una de las técnicas de interacción propuestas en el libro *3D User Interfaces: Theory and Practice* (Joseph J LaViola, Krujiff, McMahan, Bowman, & Poupyrev, 2004) y la cual se denomina *Dual-Point World Manipulation*. Esta técnica puede ser interpretada de diferentes puntos de vista, ya que no hay una implementación específica derivada de la técnica, sino que cada desarrollador puede implementar su propia versión, adaptándose a las necesidades que pueda tener el usuario del sistema.

La implementación y premisa en la que en que este trabajo se basa es en la interacción de moverse tirando de dos cuerdas en horizontal y vertical, permitiendo que el usuario usarlas como punto de pivote para su desplazamiento en el mundo virtual.

## 1.2 Objetivos

El **objetivo principal** del proyecto es la implementación de la técnica de interacción dual-point manipulation en un entorno de realidad virtual. Este objetivo se puede desglosar en los subobjetivos:

- Implementar un sistema que permita al usuario el desplazamiento utilizando las dos manos.
- Facilitar el desplazamiento del usuario tanto en el eje horizontal como en el vertical, reduciendo las limitaciones que esta técnica pueda poseer.
- Implementar un prototipo de juego que explote el potencial de esta técnica y sirva como caso de uso para su implementación en otros sistemas.
- Minimizar el impacto visual de esta técnica para facilitar al usuario su inmersión en el mundo.

# CAPÍTULO 2. DESARROLLO DEL SISTEMA

En este capítulo se realiza una descripción del desarrollo del sistema, mostrando aspectos de implementación, centrándose tanto en la técnica de interacción como en la creación del entorno que permita explotar dicha técnica.

## 2.1 Antecedentes al Desarrollo

El sistema de realidad virtual utilizado fue *HTC VIVE* y, por consiguiente, para el correcto desarrollo de la solución se utilizó el motor de juego *Unreal Engine 4*, junto al plugin *Steam VR*. El proyecto partió de la plantilla que ofrece *Unreal Engine 4*, facilitando con ello aspectos como la implementación de las manos o la cámara, entre otros.

## 2.2 Descripción de desarrollo

Durante las primeras etapas, el proyecto se centró en explorar e implementar conceptos relativos a la técnica de interacción propuesta. Una vez que la técnica había sido implementada, se barajaron todas las posibilidades que ofrecía y se crearon diferentes sistemas que hicieran uso de la técnica. Por último, se creó un prototipo de minijuego con el objetivo de facilitar al usuario la comprensión de las técnicas y su gran abanico de posibilidades.

### 2.2.1 Implementación de la interacción

La interacción específica que se pensó, y la cual podía satisfacer las necesidades identificadas, era aquella que permitiera al usuario agarrarse a un punto y desplazarse hasta este, de tal manera que el usuario se desplace sobre un punto fijo de pivotaje.

Primeramente, se creó un *Blueprint Interface* con las funciones *Climb* y *Let Go*, que permitiera al usuario agarrarse a un objeto y soltarse. Continuadamente, se creó un actor que permitiera escalar, llamado *BP\_ClimbableCube*, y se agregó un cubo alargado (columna) con un material translúcido.

Seguidamente, se modificó el *blueprint* de de *BP\_MotionController*, en concreto, la función *GetActorNearHand* agregando la interface recientemente creada, permitiendo con ello detectar los objetos escalables. Además, se modificó la función *GrabActor* para que comience la escalada cuando agarramos un actor escalable (Figura 1). También, se agregó en la función *ReleaseActor* la lógica de caer cuando soltamos el objeto escalable.

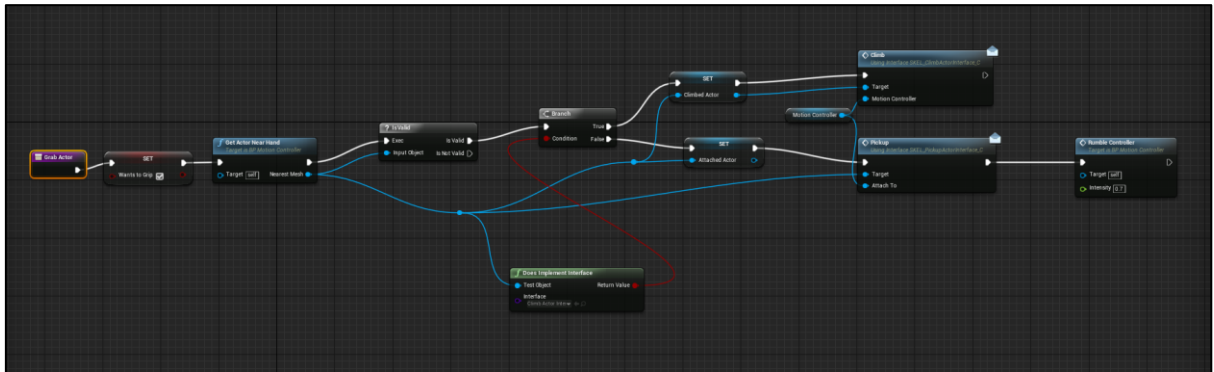
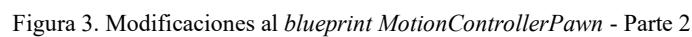
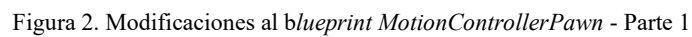


Figura 1. *Blueprint interface BP\_MotionController*

Una vez que ya se podría detectar los objetos escalables, se pasó a la implementación de la lógica de la interacción. En el *blueprint* *MotionControllerPawn* se agregaron las variables necesarias, tales como *IsClimbing* y *IsFalling* (*boolean*), la gravedad (*int* con el valor 980), la *FallingSpeed* (*int*) y la *MaxFallingSpeed* (*int* con el valor 300). Respecto al código, partiendo del *Event Ticker* se desarrolló la lógica que permita alternar entre escalar y no, y en caso de que el *pawn* estuviera escalando y se soltara, que se caiga al suelo con normalidad (Figura 2 y Figura 3).



La primera es el *eventTicker*, que implementa la lógica que mueve al actor respecto a un punto de pivotaje, en caso de que este se encuentre en fase de escalda (Figura 4).

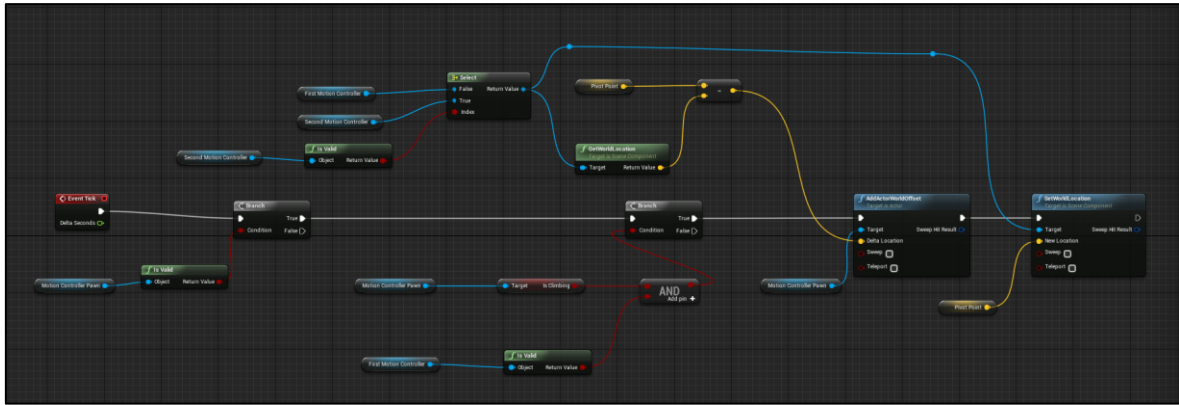


Figura 4. Blueprint BP\_ClimbableCube – EventTicker

La segunda corresponde con el evento Climb, que posee la lógica de interacción entre las manos y los objetos escalables, agregando y disminuyendo las manos que están escalando (Figura 5).

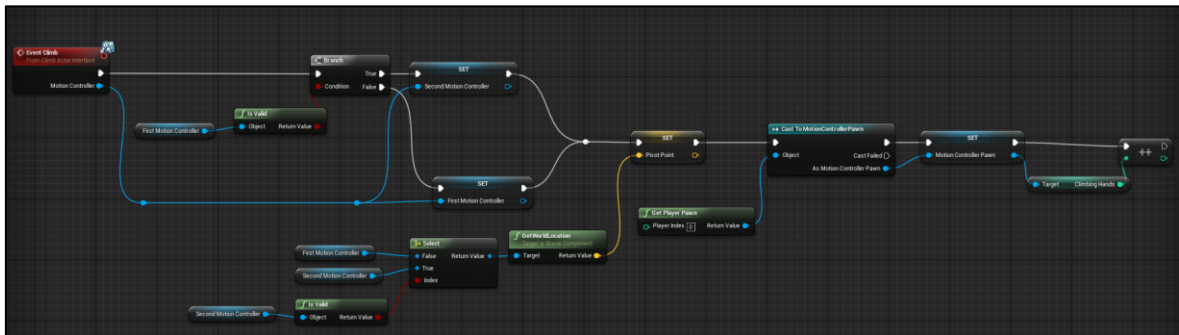


Figura 5. Blueprint BP\_ClimbableCube – Evento Climb

Por último, el evento Let Go, contrario al evento anterior, permite que el pawn se suelte del objeto escalable con la mano que corresponda (Figura 6).



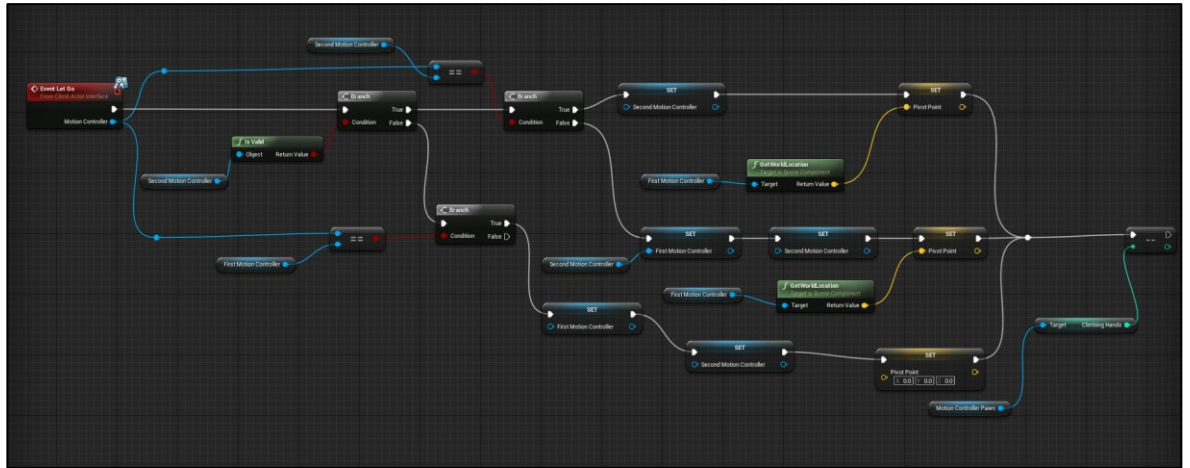


Figura 6. Blueprint BP\_ClimbableCube – Evento Let Go

Con lo desarrollado hasta este momento, quedaba implementada la interacción que permitía al *pawn* moverse pivotando sobre objetos que agarrase. Para la implementación de otros elementos que no fueran cubos, se crearon *blueprint* específicos para cada objeto a partir del *blueprint ClimbableCube*.

### 2.2.2 Movimiento y rotación de cuerdas

Primeramente, se llevó a cabo la implementación del movimiento de las cuerdas. Con la interpretación de la técnica de interacción utilizada en esta memoria, ambas cuerdas horizontal y vertical debían aparecer siempre delante de la vista del usuario. Por tanto, mirara donde mirara el usuario, al pulsar el botón de Grip (lateral), la cuerda apropiada aparecería delante de él.

Junto al movimiento, es importante resaltar que la cuerda horizontal debía estar orientada hacia donde mirara el usuario, y por tanto, a la vez que la cuerda se movía, esta debía ser orientada para ser la proyección en un plano horizontal del vector que une el centro del punto de vista del usuario con el sus ojos, aunque para ser más usable, debía estar desplazada con su punto medio en el usuario y su distancia recortada para evitar cuerdas infinitas. En la Figura 7 se puede observar este concepto, dónde el primer paso es la proyección del rayo de visión del usuario en un plano horizontal, y el segundo es el desplazamiento al centro del usuario.

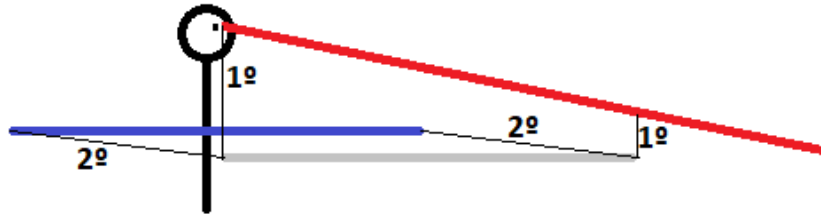


Figura 7. Creación de cuerda en base a la mirada del jugador

Con este concepto en mente, se implementaron los correspondientes bloques en los *blueprints*. Para la rotación, se tomó como base la orientación del casco de realidad virtual y nos quedamos con la rotación en el eje Z, ya que este eje es el vertical. De esta manera, manteniendo la rotación XY a 0, la cuerda se encontraría siempre en el plano horizontal, lo que se podría traducir como el primer paso de la Figura 8. Por otro lado, para el movimiento de la cuerda al usuario, se tomó la posición de la cámara como base, y se restó una altura fija para que esta quedase siempre a la altura de sus manos.

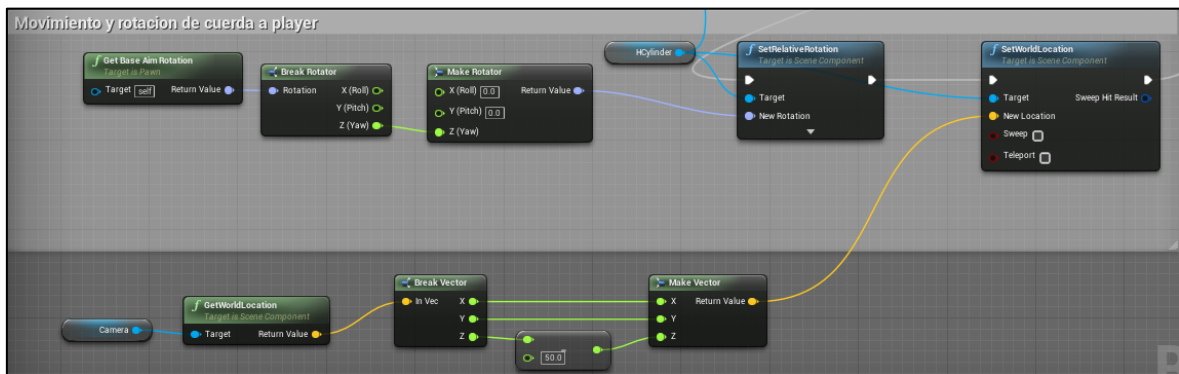


Figura 8. Lógica para el movimiento y rotación de la cuerda horizontal

Por razones de usabilidad, el control de la cuerda queda bajo el usuario, el cual puede elegir cuándo hacerla aparecer y cuando hacerla desaparecer pulsando el mismo botón. Para ello, lo primero que se comprueba tras detectar que se ha pulsado el botón grip del mando derecho, es si la cuerda es visible actualmente o no. Si la cuerda es visible, esta se hace invisible e intangible. Si, al contrario, esta era invisible, se hace tangible, se desplaza, y se hace visible en este orden, aunque es transparente al usuario puesto que es simultáneo. También es importante mencionar que, tras hacer una cuerda visible, se inicia un contador de 30 segundos que al acabar se torna invisible e intangible. De esta manera, se intentó reducir la intrusión de la técnica en el mundo al dejar cuerdas que ya no se estaban usando visibles.

En cuanto a la cuerda vertical, mientras se utilizará también la técnica del párrafo anterior para los motivos de usabilidad, se gestionó de manera ligeramente diferente el movimiento y rotación. Esto se debe a que la cuerda vertical no es rotada como tal, sino que siempre tiene la rotación (0,0,90)XYZ. Por tanto, el punto más importante fue el movimiento de esta. Ya que la idea era que siempre apareciera delante de donde el usuario estuviera mirando, se utilizó la orientación del casco de realidad virtual como referencia. En la Figura

Se retrata lo mencionado, ya que utilizando esta rotación se obtiene el vector que va desde la cámara hacia la dirección en que el usuario mira. Este vector se normaliza y multiplica para mantener la cuerda a una distancia fija del usuario siempre que esta es invocada. Una vez calculado el desplazamiento ya mencionado, se obtiene la posición central aproximada del usuario como se ve en la parte inferior de la figura X. Para ello se toma la posición de la cámara y se resta 50 unidades en el eje vertical. Para calcular a dónde se debe desplazar, ambos vectores (posición y desplazamiento) se suman para establecer la posición final de la cuerda.

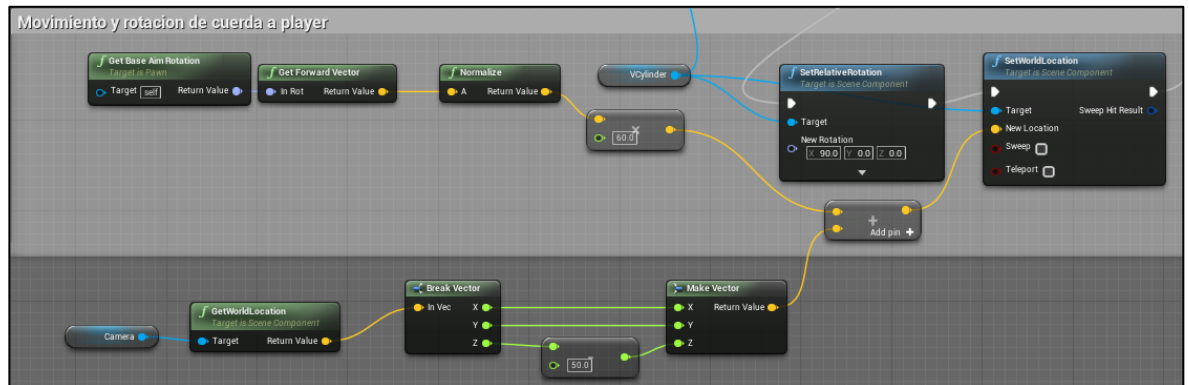


Figura 9. Lógica para el movimiento y rotación de la cuerda vertical

### 2.2.3 Control de gravedad

También se ha interpretado que, en los casos en los que el usuario pueda tener una mano ocupada, desplazarse en el eje vertical podría ser complejo. Para ello, se estableció un botón que permite activar o desactivar la gravedad de manera que, si el usuario trepa, puede hacerlo con una sola mano sin caer mientras se suelta y se vuelve a agarrar. Esto se ha implementado de manera simple, ya que teniendo en cuenta el valor actual de la gravedad (0 o 981) se establece el valor opuesto (Figura 10).

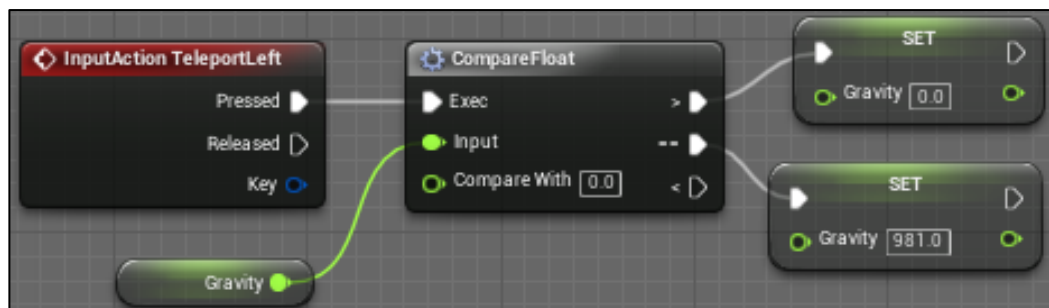


Figura 10. Lógica para anular la gravedad

El código completo de la interacción y de la demo se encuentra público y está alojado en GitHub para tanto consultarlo como descárgalo en su equipo a través del enlace: [https://github.com/JoseRamonMartinez/VR\\_DualPoint\\_World\\_Manipulation](https://github.com/JoseRamonMartinez/VR_DualPoint_World_Manipulation).



# CAPÍTULO 3. DESCRIPCIÓN DEL SISTEMA

En este capítulo se describe como se han integrado las diferentes implementaciones detalladas con anterioridad. Durante el capítulo, se observará como encajan todas las técnicas en un entorno virtual bajo el contexto de un minijuego.

## 3.1 Descripción de la Solución

Los objetivos del videojuego son dos; llevar un cubo de un altar a otro y realizar el trayecto en el menor tiempo posible. Para llevar esto a cabo, el jugador tendrá que valerse de su agilidad para trepar rocas y desplazarse rápidamente tirando sus cuerdas tanto verticales como horizontales.

El juego se desarrolla en un terreno ambientado como un valle, en el cual aparece el jugador. En la zona inicial, el jugador tiene un cartel con indicaciones de cómo desplazarse y controlar las cuerdas, así como controlar la gravedad (se introduce al usuario la técnica de interacción) y el objetivo del juego (Figura 11). Una vez explicado, el usuario tendrá que subir a la montaña, donde se encuentra el cubo a transportar.

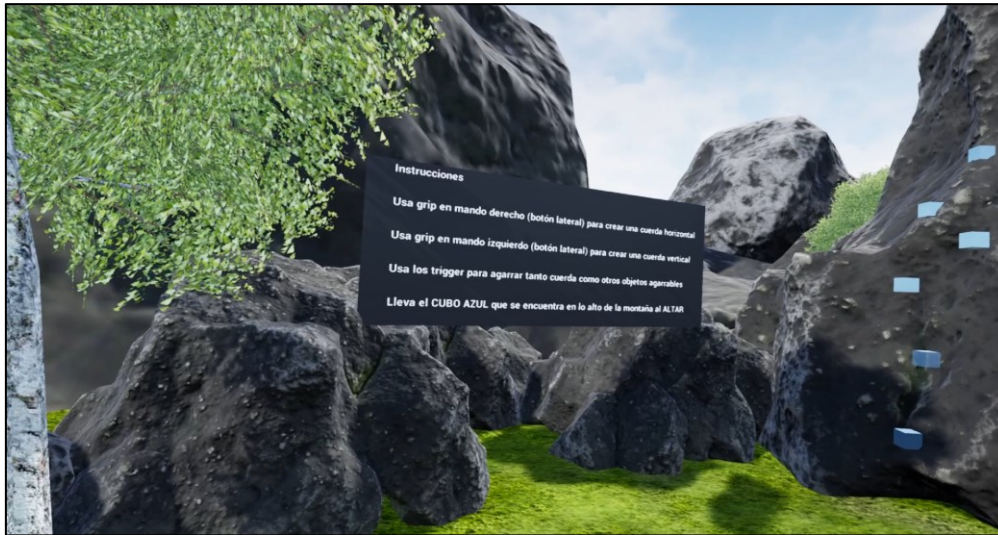


Figura 11. Pantalla inicial de juego

Una vez con el cubo en la mano, al pasar por delante del altar comenzarán el recorrido y el contador.

Para implementar el comienzo del recorrido, se ha diseñado un *trigger* que acota toda la zona después del altar hacia la meta. Este detectará el momento en el que el jugador lo atraviesa y almacenará la hora del sistema en ese momento.

Durante el recorrido del juego, la mayoría de los usuarios encontrará que la manera más rápida de completar el nivel es lanzar el cubo para dejar ambas manos libres para desplazarse, aunque la manera de lanzar el cubo es importante puesto que, si se lanza inadecuadamente, el usuario tendrá que realizar un recorrido más largo para recogerlo.

En la parte final, hemos aprovechado la implementación de las cuerdas para implementar una escalera, la cuál se basa en una “cuerda invisible” con las dimensiones de esta. De esta manera, se puede agarrar no solo los escalones, sino cualquier punto. Esto aumenta la usabilidad de la escalera y facilita al usuario su uso, a la par con el botón de quitar gravedad, que le permitirá subir la escalera con una mano sin caer mientras transporta el cubo con la otra. Esto es visible en la Figura 12, donde el jugador está utilizando una mano para agarrar el cubo, mientras que la otra está suelta para agarrar la escalera. Todo esto se produce mientras flota en el aire, facilitando la acción de desplazarse en el eje vertical.



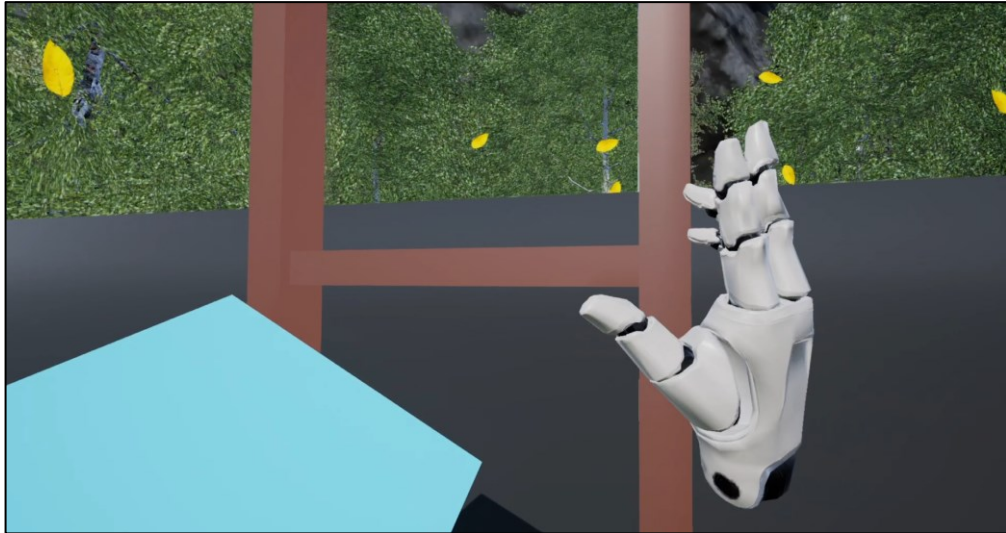


Figura 12. Pantalla del juego - Escalera vertical

Finalmente, el usuario se ve forzado a utilizar la cuerda vertical, que hasta este punto ha sido prescindible. Esto se debe a que, para ascender al altar, debe subir a una plataforma colocada a una demasiado elevada como para simplemente avanzar con una cuerda horizontal, y sin ningún agarre en el mapa que facilite su ascensión.

Una vez el jugador consiga trepar a la plataforma superior, depositará el cubo sobre el altar y un *trigger* detectará esto y declarará el final del juego (tomando el tiempo del sistema y realizando la resta de ambos tiempos, para así saber el tiempo utilizado para realizar el recorrido y comenzar el reinicio del nivel), mostrando delante del jugador el tiempo que ha tardado, junto a un mensaje de felicitación y un aviso de que el nivel se va a reiniciar en 30 segundos, visibles estos tres últimos (Figura 13).

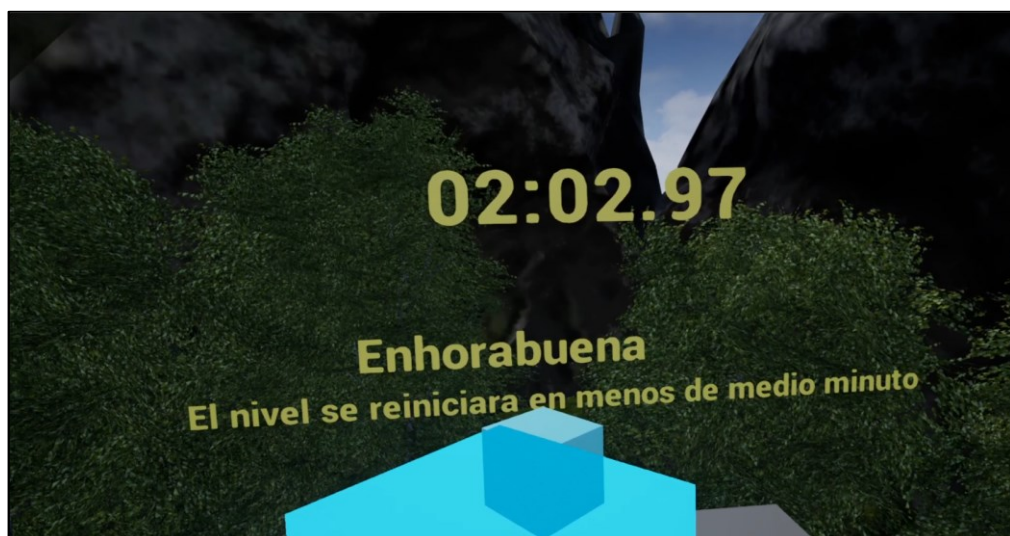


Figura 13. Pantalla final del juego

Es caso de que se quiera obtener una mejor idea del *gameplay*, se puede acceder a un video con esta demo a través del enlace: <https://www.youtube.com/watch?v=u6hWpus1zto>.





# CAPÍTULO 4. CONCLUSIONES Y TRABAJO FUTURO

En este capítulo se muestran las conclusiones, obtenidas a través de la realización del trabajo, y definiendo el avance realizado en el campo en el que el proyecto se enmarca. Además, se proponen diferentes líneas o mejoras para seguir el proyecto en un futuro.

## 4.1 Conclusiones

La realización del trabajo ha supuesto la consecución del objetivo principal, implementando la técnica de interacción dual-point manipulation en un entorno de realidad virtual. Además, se han abordado los diferentes subobjetivos en los que se segmentaba. Se ha implementado una *cuerda* que permite al usuario realizar un desplazamiento, logrando el subobjetivo *Implementar un sistema que permita al usuario el desplazamiento utilizando las dos manos*. Simultáneamente, se logró el subobjetivo *Facilitar el desplazamiento del usuario tanto en el eje horizontal como en el vertical, reduciendo las limitaciones que esta técnica pueda poseer*, mediante la implementación de un botón que permite eliminar la gravedad, facilitando el transporte de objetos en el eje vertical; y la creación de diferentes botones para crear *cuerdas* verticales y horizontales. El subobjetivo *Implementar un prototipo de juego que explote el potencial de esta técnica y sirva como caso de uso para su implementación en otros sistemas* fue abordado a través de la creación de un circuito en realidad virtual situado en un bosque que obliga al usuario a usar los diferentes sistemas implementados, obteniendo conocimientos acerca de su aplicación en un sistema real. Por último, el subobjetivo *Minimizar el impacto visual de esta técnica para facilitar al usuario su inmersión en el mundo* fue abordado a través de la implementación de un sistema que hace que las cuerdas pasado un tiempo desaparezcan, además de ser estas translucidas, lo cual permite al usuario ver el entorno por donde se mueve de una mejor manera.

Desde una perspectiva académica, este trabajo supone de movimiento e interacción con el entorno, que sin ser quizás el más eficiente, aumenta notoriamente el grado de

inmersión si se utiliza en el momento y entorno correcto. Un ejemplo es la escalera, la cual implementa la interacción desarrollada de una forma no invasiva en el entorno virtual de la demo.

Por último, desde el punto de vista personal, este trabajo ha permitido obtener competencias sobre el trabajo en equipo y el desarrollo de un trabajo que se aproxima a la investigación, partiendo de una hipótesis y una primera idea, y explorando los límites de esta. Además, se han obtenido conocimiento sobre los entornos en realidad virtual y la importancia de la interacción en estos, mejorando con ello la inmersión y sensación de transporte a un mundo diferente.

## **4.2 Trabajos Futuros**

Aunque el trabajo es completo y resulta finalizado, se han identificado diversos trabajos futuros que podría seguir el proyecto.

### Cuerda como liana

Resulta interesante el aspecto de crear una cuerda, tal y como lo es la actual cuerda vertical, y que esta se mueve de la misma manera que lo hace una liana, pudiendo con ello implementar aspectos reales en juegos en entorno de realidad virtual

### Cuerda como látigo

La implementación de una cuerda como látigo, similar a lo que se puede observar en las películas de Indiana Jones resulta interesante desde la perspectiva de atravesar barrancos o subir a sitios altos.

### Físicas de cuerda

Este aspecto consta de la implementación de físicas realistas de cuerda, lo cual permitiría aumentar el grado de inmersión que esta ofrece.

### Exploración de otros minijuegos

Crear minijuegos que exploren la técnica desde diferentes perspectivas es un punto muy a tener en cuenta. Una idea que se baraja es la implementación del típico juego donde dos grupos de personas tiran de la cuerda desde ambos.

## BIBLIOGRAFÍA

Epic Games. (28 de 1 de 2021). *Unreal Engine*. Obtenido de <https://www.unrealengine.com/en-US/?sessionInvalidated=true>

Joseph J LaViola, J., Krujiff, E., McMahan, R. P., Bowman, D., & Poupyrev, I. P. (2004). *3D User Interfaces: Theory and Practice*. (D. A. Bowman, Ed.) Manhattan, Nueva York, EE. UU.

Valve. (28 de 1 de 2021). *Steam VR*. Obtenido de [https://valvesoftware.github.io/steamvr\\_unity\\_plugin/](https://valvesoftware.github.io/steamvr_unity_plugin/)

Vive. (28 de 1 de 2021). *Vive*. Obtenido de <https://www.vive.com/eu/product/vive/>