



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



Escola Tècnica  
Superior d'Enginyeria  
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica  
Universitat Politècnica de València

# ESTUDIO Y EXPERIMENTACIÓN DE DISPOSITIVOS DE INTERACCIÓN AVANZADOS SOBRE UNITY 3D: WIIMOTE

Trabajo Fin de Grado

**Grado en Ingeniería Informática**

**Autor:** Nicolás Tortosa Cerezo

**Tutor:** Manuel Agustí Melchor

2017-2018

## **Agradecimientos**

---

*A mi familia, amigos y tutor.*

*A mi familia, sobretodo padre y hermana, por su apoyo constante en la realización de  
este TFG.*

*A los Koalas, grandes amigos y mejores personas, por ayudarme en mis ratos libres a  
pensar en otras cosas que no fueran este TFG.*

*Al tutor por guiarme y ayudarme durante la redacción de mi TFG.*

Las formas de interacción entre los usuarios y los ordenadores siguen evolucionando, por este motivo podemos observar como aparecen nuevos dispositivos que proponen métodos más naturales que pueden ser usados fuera de la mesa de trabajo, en comparación a lo que venimos haciendo desde hace más de cincuenta años con el teclado y el ratón para la entrada de información.

Este trabajo nace del incremento en el mercado de dispositivos de interacción, especialmente en aplicaciones de realidad virtual y aumentada, que ofrecen más alternativas de entretenimiento para las personas, aunque también se pueden utilizar para la rehabilitación de personas con discapacidades o lesiones permanentes. Debido a dicho incremento nace de forma natural un afán por reutilizar algunos dispositivos que en muchas ocasiones se encuentran a nuestro alcance o se han vendido para comprarse otros más actuales.

En la octava conferencia de realidad virtual de Toronto se realizó un estudio sobre el uso de dispositivos de interacción humanos en el ámbito de realidad virtual. Los resultados fueron que solo el 10% de participantes indicó que el uso de realidad virtual con teclado y ratón fue satisfactorio. Como se dice, posteriormente, esto se debe a la necesidad de estar cerca de los dispositivos tradicionales, como el ratón y el teclado.

A partir del análisis del hardware de los dispositivos existentes se propondrá el uso del “Wii remote” para la realización de aplicaciones interactivas en el ordenador y se estudiará la conexión a bajo nivel de este dispositivo con un computador bajo diferentes sistemas operativos. También se investigarán algunas aplicaciones ya creadas que usan este dispositivo y el procedimiento necesario para poder usarlo en el ámbito del ordenador de sobremesa, ya que se necesitan accesorios adicionales para utilizar todas las funciones que proporciona. De ahí se derivará la realización hardware de estos accesorios para la creación de instalaciones personalizadas.

El trabajo consiste en implementar varios juegos para probar la eficacia del “Wii remote”, en otra plataforma diferente a la que le dio origen, como es el ordenador de sobremesa y para comprobar su eficacia frente a los dispositivos tradicionales del mismo. Para centrar la realización en el uso del dispositivo, el desarrollo de las aplicaciones se va a realizar sobre un motor de desarrollo de aplicaciones como es Unity 3D.

**Palabras clave:** Dispositivos de interacción, “Wii remote”, Unity3D, Reutilización, Ordenador de sobremesa.

## Resum

---

Les formes d'interacció entre els usuaris i els ordinadors segueixen evolucionant, per aquest motiu podem observar com apareixen nous dispositius per a proposar mètodes més naturals que poden ser emprats fora de la taula de treball, en comparació al que venim fent des de fa més de cinquanta anys amb el teclat i el ratolí per a l'entrada d'informació.

Aquest treball naix de l'increment en el mercat de dispositius d'interacció, especialment en aplicacions de realitat virtual i augmentada, que ofereixen més alternatives d'entreteniment per a les persones, tot i que també es poden utilitzar per a la rehabilitació de persones amb discapacitats o lesions permanents. A causa d'aquest increment naix de forma natural un afany per reutilitzar alguns dispositius que en moltes ocasions es troben al nostre abast o s'han venut per comprar-se altres més actuals.

En la vuitena conferència de realitat virtual de Toronto es va realitzar un estudi sobre l'ús de dispositius d'interacció humans en l'àmbit de realitat virtual. Els resultats van ser que només el 10% de participants va indicar que l'ús de realitat virtual amb teclat i ratolí va ser satisfactori. Com es diu, posteriorment, això es deu a la necessitat d'estar a prop dels dispositius tradicionals, com el ratolí i el teclat.

A partir de l'anàlisi del maquinari dels dispositius existents es proposarà l'ús del "Wii remote" per a la realització d'aplicacions interactives a l'ordinador i s'estudiarà la connexió a baix nivell d'aquest dispositiu amb un ordinador sota diferents sistemes operatius. També s'investigaran algunes aplicacions ja creades que usen aquest dispositiu i el procediment necessari per poder usar-lo en l'àmbit de l'ordinador de sobretaula, ja que es necessiten accessoris addicionals per utilitzar totes les funcions que proporciona. D'aquí es derivarà la realització maquinari d'aquests accessoris per a la creació d'instal·lacions personalitzades.

El treball consisteix en implementar diversos jocs per provar l'eficàcia del "Wii remote", en una altra plataforma diferent a la que li va donar origen, com és l'ordinador de sobretaula i per comprovar la seva eficàcia davant els dispositius tradicionals del mateix. Per centrar la realització en l'ús del dispositiu, el desenvolupament de les aplicacions es va a realitzar sobre un motor de desenvolupament d'aplicacions com és Unity 3D.

**Paraules clau:** Dispositius d'interacció, "Wii remote", Unity3D, Reutilització, Ordinador de sobretaula.

The ways of interaction between users and computers don't stop evolving, for this reason we can see how new devices appear to propose more natural methods that can be used outside the workspace, far from what we have been doing for more than fifty years with the keyboard and the mouse as a method to write information.

This work is born by the increase in the market of HID, especially in virtual and augmented reality applications, which offer more entertainment alternatives for people, and they can also be used for the rehabilitation of people with disabilities or permanent injuries. Due to this increase, there is a natural movement of curiosity to reuse some devices that are often within our reach or have been sold to buy a more recent one.

At the eighth virtual reality conference in Toronto, a study was conducted on the use of human interaction devices in the field of virtual reality. The results showed that only 10% of participants indicated that the use of virtual reality with keyboard and mouse was satisfactory. Subsequently, this study result shows that this lack of immersion is due to the need to be close to traditional devices, such as the mouse and the keyboard.

From the hardware analysis of the existing devices, the use of the "Wii remote" will be proposed for the realization of interactive applications in the computer and the low level connection of this device will be studied with a computer under different operating systems. Some applications that are already using this device and the necessary procedure to be able to use it in a PC will also be investigated, since additional accessories are needed to use all the functions it provides. From there, the hardware realization of these accessories will be derived for the creation of customized installations.

The work consists of implementing several games to test the effectiveness of the "Wii remote", in another platform different from the one given birth for, such as the desktop computer, and to check its effectiveness against the traditional devices of the same. To focus the realization of this work on the use of this device, the development of the applications will be done on an application development engine such as Unity 3D.

**Key words:** Interaction devices, "Wii remote", Unity3D, Reuse, Desktop.

# Índice

---

|  |    |
|--|----|
| <b>1. Introducción</b> .....   | 12 |
| 1.1. Objetivos.....  | 13 |
| 1.2. Metodología.....  | 13 |
| 1.3. Estructura de la obra.....  | 14 |
| <b>2. Estado del arte</b> .....  | 15 |
| 2.1. Análisis de dispositivos tradicionales.....                       | 20 |
| 2.1.1. Ratón.....  | 20 |
| 2.1.2. Teclado.....  | 21 |
| 2.2. Análisis de dispositivos avanzados.....                           | 22 |
| 2.2.1. Wii remote.....   | 23 |
| 2.2.2. Switch JoyCon.....  | 26 |
| 2.2.3. Leap Motion.....  | 27 |
| 2.2.4. PlayStation Move.....   | 29 |
| 2.2.5. HTC Vive Controller.....  | 32 |
| 2.2.6. Steam controller.....   | 34 |
| 2.2.7. Smart Phone.....  | 35 |
| 2.3. Análisis del software.....  | 37 |
| 2.4. Elección del dispositivo a utilizar.....                          | 39 |
| <b>3. Estudio avanzado del “Wii remote”</b> .....                      | 42 |
| 3.1. Fabricación hardware necesario.....                               | 42 |
| 3.2. Estudio de protocolo de comunicación del “Wii remote”.....        | 46 |
| 3.3. Estudio de conexión al ordenador.....                             | 48 |
| 3.3.1. Windows.....  | 48 |
| 3.3.2. Linux.....  | 49 |
| 3.4. Estudio de aplicaciones y librerías que usan el “Wii remote”..... | 49 |
| 3.4.1. HID Wiimote.....  | 49 |

|   |           |
|---|-----------|
| 3.4.2. CWiiD.....   | 50        |
| 3.4.3. Unity-Wiimote .....  | 50        |
| <b>4. Casos de estudio.....</b>   | <b>51</b> |
| 4.1. “Wii remote” como ratón.....   | 51        |
| 4.2. Implementación con la barra infrarroja en Unity3D.....   | 52        |
| 4.3. Implementación con el motion plus en Unity3D .....   | 54        |
| 4.4. Evaluaciones por parte de usuarios.....  | 58        |
| <b>5. Conclusiones .....</b>  | <b>60</b> |
| 5.1. Valoración personal .....  | 61        |
| <b>6. Trabajos futuros.....</b>   | <b>62</b> |
| 6.1. Uso de la cámara IR con el giroscopio para mejorar la detección del movimiento del “Wii Remote”..... | 62        |
| 6.2. Integración con Google Cardboard como sistema de realidad virtual de bajo coste                      | 62        |
| 6.3. Múltiples barras para un mejor control en realidad virtual.....                                      | 62        |
| <b>7. Referencias.....</b>  | <b>64</b> |
| <b>8. Anexos .....</b>  | <b>67</b> |



## Índice de anexos

---

|   |           |
|---|-----------|
| <b>Anexo 1: Descripción máquina virtual con distribución Linux .....</b>              | <b>67</b> |
| <b>Anexo 2: Capturas conexión “Wii remote” al PC en distintos sistemas operativos</b> | <b>68</b> |
| Anexo 2.1: Conexión “Wii remote” en Windows .....                                     | 68        |
| Anexo 2.2: Conexión de “Wii remote” en Linux .....                                    | 70        |
| <b>Anexo 3: Explicaciones código implementaciones .....</b>                           | <b>71</b> |
| Anexo 3.1: Rotación mediante posición relativa .....                                  | 71        |
| Anexo 3.2: Creación objetos simples .....   | 71        |
| Anexo 3.3: Adquisición desviación giroscopio y reinicio de este sensor .....          | 72        |
| Anexo 3.4: Filtrar resultados giroscopio sabiendo desviación “Wii remote” .....       | 73        |
| Anexo 3.5: Creación y limpieza “Wii remotes” en multijugador .....                    | 74        |
| <b>Anexo 4: Descripción “assets” externos usados en Unity3D .....</b>                 | <b>75</b> |
| <b>Anexo 5: Formulario online rellenado por sujetos reales .....</b>                  | <b>76</b> |
| <b>Anexo 6: Resultados del formulario .....</b>                                       | <b>81</b> |

## Índice de figuras

---

|   |           |
|---|-----------|
| <i>Figura 2-1. Captura de proyecto de realidad aumentada, Project North Star.....</i> | <i>15</i> |
| <i>Figura 2-2. Captura de Beat Saber.....</i>   | <i>16</i> |
| <i>Figura 2-3. Captura de estudio de Allyn Malyentano con el HTC Vive. ....</i>       | <i>17</i> |
| <i>Figura 2-4. Línea tiempo dispositivos de interacción avanzados.....</i>            | <i>18</i> |
| <i>Figura 2-5. Demostración de pizarra virtual por Johnny Chung Lee. ....</i>         | <i>19</i> |
| <i>Figura 2-6. Primer prototipo de raton de 1964. ....</i>                            | <i>20</i> |
| <i>Figura 2-7. Primera máquina de escribir. ....</i>                                  | <i>21</i> |
| <i>Figura 2-8. Teclado IBM modelo M.....</i>  | <i>22</i> |
| <i>Figura 2-9. Consola Wii y accesorios.....</i>                                      | <i>23</i> |
| <i>Figura 2-10 Mando Wii remote.....</i>  | <i>23</i> |
| <i>Figura 2-11. Interior Wii remote.....</i>  | <i>24</i> |
| <i>Figura 2-12. Interior Wii MotionPlus.....</i>                                      | <i>25</i> |
| <i>Figura 2-13. Switch JoyCon.....</i>  | <i>26</i> |
| <i>Figura 2-14. Consola Nintendo Switch con 2 “JoyCons”.....</i>                      | <i>26</i> |
| <i>Figura 2-15. Zona interaccion LeapMotion.....</i>                                  | <i>28</i> |
| <i>Figura 2-16. Interior LeapMotion.....</i>  | <i>29</i> |
| <i>Figura 2-17. Sistema de PS3 para PlayStation Move.....</i>                         | <i>29</i> |
| <i>Figura 2-18. Sistema de PS4 para PlayStation Move.....</i>                         | <i>30</i> |
| <i>Figura 2-19. Interior “PSMove”.....</i>  | <i>31</i> |
| <i>Figura 2-20. PlayStation Move.....</i>   | <i>31</i> |
| <i>Figura 2-21. Posicionamiento Lighthouse para HTC Vive Controller.....</i>          | <i>32</i> |
| <i>Figura 2-22. Interior HTC Vive Controller.....</i>                                 | <i>33</i> |
| <i>Figura 2-23.HTC Vive Controller.....</i>   | <i>33</i> |
| <i>Figura 2-24.Steam Controller.....</i>  | <i>34</i> |
| <i>Figura 2-25. Steam Controller.....</i>   | <i>35</i> |
| <i>Figura 2-26. Foto de IBM Simon. ....</i>   | <i>35</i> |
| <i>Figura 2-27. Foto de iPhone 1.....</i>   | <i>36</i> |
| <i>Figura 3-1. Barra sensor Nintendo Wii. ....</i>                                    | <i>42</i> |
| <i>Figura 3-2.Interior Barra sensor Wii.....</i>                                      | <i>43</i> |
| <i>Figuras 3-3 y 3-4. Interior barra Wii segundo modelo.....</i>                      | <i>43</i> |
| <i>Figura 3-5. Circuito Barra sensor Wii.....</i>                                     | <i>44</i> |
| <i>Figura 3-6. Circuito barra sensor.....</i>   | <i>44</i> |
| <i>Figura 3-7. Carcasa barra sensor.....</i>  | <i>45</i> |



|   |    |
|---|----|
| <i>Figura 3-8. Esquema circuito versión 2.</i>  | 45 |
| <i>Figura 3-9. Segunda versión circuito soldado.</i>  | 46 |
| <i>Figura 3-10. Carcasa pulida y pintada.</i>   | 46 |
| <i>Figura 4-1. Comando lswm con la dirección MAC de un "Wii remote".</i>                                    | 51 |
| <i>Figura 4-2. Captura de comando y edición de fichero de configuración Cwiid.</i>                          | 52 |
| <i>Figura 4-3. Comando para convertir "Wii remote" en ratón.</i>  | 52 |
| <i>Figura 4-4. Escena demo Unity-Wiimote.</i>   | 53 |
| <i>Figura 4-5. WiimoteShooter V0.8a.</i>  | 53 |
| <i>Figura 4-6. WiimoteShooter v1.0.</i>   | 54 |
| <i>Figura 4-7. WiimoteMenu V0. 8a.</i>  | 55 |
| <i>Figura 4-8. Instrucciones Wiimote Jedi v1.0</i>  | 55 |
| <i>Figura 4-9. WiimoteJedi v1.0.</i>  | 57 |
| <i>Figura 4-10. Wiimote Multiplayer v1.0.</i>   | 58 |
| <i>Figura 8-1. Configuración de Bluetooth y otros dispositivos, con el botón a pulsar marcado en rojo.</i>  | 68 |
| <i>Figura 8-2. Dispositivos e impresoras, con paso a seguir en rojo.</i>                                    | 68 |
| <i>Figura 8-3. Pantalla de elección de dispositivo a conectar, con paso a seguir en rojo.</i>               | 69 |
| <i>Figura 8-4. Pantalla donde se tiene que dejar el PIN vacío la correcta conexión con el "Wii remote".</i> | 69 |
| <i>Figura 8-5. Obtención dirección MAC del Wii remote.</i>  | 70 |
| <i>Figura 8-6. Configuración del fichero para usar Wii remote como ratón.</i>                               | 70 |
| <i>Figura 8-7. Conexión Wii remote en modo ratón.</i>   | 70 |
| <i>Figura 8-8. Pantalla conexión de "Wii remote" en wmgui.</i>  | 70 |

## Índice de fragmentos de código

---

|  |    |
|--|----|
| <i>Fragmento Código 8-1. Función UpdateRotationWiimote.</i>                                  | 71 |
| <i>Fragmento Código 8-2. Función de creación de un objeto normal.</i>                        | 72 |
| <i>Fragmento Código 8-3. Función para obtener desviación giroscopio e iniciar la escena.</i> | 73 |
| <i>Fragmento Código 8-4. Filtrado de resultados giroscopio.</i>                              | 73 |
| <i>Fragmento Código 8-5. Función para inicializar colección de "Wii remotes".</i>            | 74 |
| <i>Fragmento Código 8-6. Función OnDestroy.</i>  | 75 |

## Índice de tablas

---

|   |    |
|---|----|
| <i>Tabla 2-1. Comparativa dispositivos avanzados.....</i>   | 40 |
| <i>Tabla 3-1. Funciones disponibles “Wii remote”. .....</i> | 47 |

## Índice de abreviaturas

---

**3D:** 3 Dimensiones

**GUI:** Graphical User Interface

**HCI:** Human Computer Interaction

**HID:** Human Interface Device

**HTC:** High Tech Computer

**IR:** Infrared

**PC:** Personal Computer

**PS3:** PlayStation 3

**PS4:** PlayStation 4

**PSMove:** PlayStation Move

**UI:** User Interface

## 1. Introducción

La tecnología del mundo avanza y, actualmente, es casi imposible vivir el día a día sin hacer uso de uno o más dispositivos de interacción como pueden ser el ratón, el teclado o la pantalla táctil de un “Smartphone”, para relacionarnos con otras personas o trabajar.

Estos dispositivos de interfaz humana, conocidos también por sus siglas en inglés HID, son los encargados de permitirnos hablar con las máquinas sin necesidad de saber exactamente el lenguaje que esta usa para sus operaciones. Para esto, los HID utilizan interfaces fáciles de aprender que se transcriben internamente en operaciones complejas. Entre estos dispositivos se pueden encontrar los mencionados anteriormente para facilitarnos el uso de los aparatos más comúnmente usados.

Hay otro tipo de HID que intenta que estas interacciones persona ordenador, o sus siglas inglesas HCI, sean más naturales, en el sentido de que los humanos no estamos diseñados para escribir tecleando o a apuntar moviendo un ratón en una superficie plana. Por esto, surgieron los dispositivos de interacción que llamaremos avanzados porque intentan transcribir a la máquina acciones mucho más complejas, pero más naturales para los humanos, con el fin de minimizar los tiempos de aprendizajes y aumentar la inmersión en la actividad que haga uso de estos.

Este trabajo nace del incremento en el ritmo de evolución de estos dispositivos en el mercado, especialmente en aplicaciones de realidad virtual y aumentada, para ofrecer más alternativas de entretenimiento para las personas, pero también de rehabilitación para gente con discapacidades o lesiones permanentes. Con este incremento nace de forma natural un afán por reutilizar algunos dispositivos que en muchas ocasiones se encuentran a nuestro alcance o se han vendido para comprarse otros más actuales.

Los dispositivos que reutilizamos se pueden encontrar a un precio muy económico en comparación con los dispositivos punteros que van sacando las diferentes compañías, pero, como podremos observar en este trabajo, no son muy diferentes en cuanto a su hardware. Por este parecido entre los dispositivos de interacción, la mayoría de los secretos que tienen están detrás de un estudio en profundidad, ya que los más populares se basan en acercamientos distintos a las mismas tecnologías como veremos en este trabajo.

En este trabajo veremos cómo se pueden utilizar algunos de estos dispositivos para diseñar mini juegos que se puedan usar con fines lúdicos, de rehabilitación o artísticos.

## 1.1. Objetivos

El objetivo principal de este trabajo es el estudio de los dispositivos de interacción y de sus características para comprender las diferencias entre ellos y los dispositivos de interacción tradicionales, como el ratón y el teclado. Para lograr esto, este objetivo se ha dividido en los siguientes subobjetivos:

- Caracterizar los dispositivos de interacción avanzados y los clásicos.
- Análisis del hardware de los dispositivos avanzados.
- Estudio del software creado para los dispositivos anteriores en el PC de sobremesa.
- Elección del “Wii remote” para el resto de los objetivos.
- Estudiar el método de conexión al ordenador.
- Caracterizar hardware necesario para ser utilizado en el ordenador y en instalaciones de tamaño medio.
- Estudiar y validar las opciones de desarrollo a alto nivel para el Wii remote.
- Implementar mini juegos usando las características del Wii remote.
- Estudiar opiniones de mini juegos en usuarios reales.

Con estos subobjetivos realizados podemos estudiar los resultados y sacar conclusiones para determinar si estos dispositivos avanzados pueden ser mejores que los tradicionales para algunas actividades.

## 1.2. Metodología

Para conseguir estos objetivos, hay que primeramente conocer los dispositivos avanzados, ver de lo que son capaces y estudiar el software ya realizado para estos dispositivos en el ordenador.

Una vez hecho esto podemos explorar como conectar el “Wii remote” al PC para hacer los mini juegos y encontrar los distintos recursos disponibles en Internet que puedan facilitar el desarrollo de estos.

Después de recopilar todos estos datos, se realizarán los componentes necesarios a mano para poder utilizar todas las características de este dispositivo en el ordenador, dentro de los mini juegos que posteriormente se desarrollarán en un motor de videojuegos.



Por último, se probarán estos mini juegos en usuarios reales para recoger información sobre el uso de estos dispositivos de interacción en el ordenador frente al teclado y el ratón.

### **1.3. Estructura de la obra**

La estructura de la obra completa se desenvuelve en las siguientes partes:

Con la introducción hemos definido los dispositivos de interacción avanzados que analizaremos exhaustivamente en el segundo capítulo de este trabajo, en este capítulo también veremos las posibilidades que nos ofrecen en el PC estos dispositivos para finalizarlo con la elección del “Wii remote” para un análisis más exhaustivo en el tercer capítulo.

Con un dispositivo elegido pasamos a investigar con más detenimiento sus protocolos de comunicación y los recursos que podemos usar para el resto del trabajo, o fabricar para usar el dispositivo en el PC. Estos recursos nos serán cruciales para explotar las características del “Wii remote” con una “relativa” facilidad, ya que el giroscopio nos dará algunos problemas, en una plataforma para la que no estaba diseñado inicialmente este dispositivo.

Finalizando este último capítulo pondremos a prueba las aplicaciones realizadas en usuarios reales para comprobar su seguimiento de estos dispositivos y si estarán interesados cuando en un futuro salgan otros dispositivos de este tipo más económicos o con más diversidad de usos fuera del ámbito lúdico.

## 2. Estado del arte

En los últimos años, las investigaciones para mejorar los dispositivos de interacción avanzados han estado estrechamente ligadas a la realidad virtual y a la realidad aumentada, ya que permiten una mayor libertad de movimiento y un mejor uso de las tecnologías incorporadas en los mismos sin requerir al usuario que mire una pantalla fija. Permitiendo no solo la inmersión en entornos virtuales, sino también la posibilidad de empuñar objetos con las mismas acciones que en el mundo real.

Antes de entrar a ver algunos ejemplos de estas tecnologías y por qué es necesario el uso de estos dispositivos avanzados explicaremos brevemente lo que es la realidad virtual y aumentada.

Un sistema de realidad aumentada, según Hollerer and Feiner(2004), es aquel que combina información del mundo real y generada por un ordenador en el mundo real<sup>1</sup>, interactivo y en tiempo real, y que contiene objetos virtuales y reales, solo visible a través de gafas, como las vistas en la figura 2-1. En estos sistemas se ha popularizado el uso de las características de la mayoría de smartphones para generar estos entornos con fines educativos o comerciales, algunos ejemplos de estos sistemas pueden ser las aplicaciones creadas por la empresa Augment que realizan por contratación aplicaciones muy diversas, que pueden servir para que el cliente vea lo que quiere pedir con las características deseadas a través en un entorno real o para digitalizar objetos de libros para ayudar al alumno a visualizar en 3D y comprenderlos mejor, como es el caso de la Universidad Autónoma de Nuevo León de Méjico, que está usando una de las aplicaciones de esta empresa para algunos libros de textos de medicina, o aplicaciones de móvil como Sun Seeker que permite ver dónde está el sol.



*Figura 2-1. Captura de proyecto de realidad aumentada, Project North Star.<sup>2</sup>*

---

<sup>1</sup> Ver Feiner, T. H. (2004). Mobile Augmented Reality. Capitulo 1 Introduction.

<sup>2</sup> Captura extraída de <http://blog.leapmotion.com/north-star-open-source/> .

Por otro la realidad virtual, término acuñado por Jaron Laier en 1987, es definido por Chris Woodford (2018) como la experimentación de cosas a través de ordenadores que no existen en la realidad<sup>3</sup>. Por este motivo se usa principalmente para videojuegos como Beat Saber (2018), el cual se puede ver en la figura 2-2, o Duck Season (2017), o para ayudar a superar enfermedades mentales o fisiológicas, por ejemplo, el laboratorio clínico de King College (Londres) que, al cargo de Lucia Valmaggia, realiza rehabilitaciones en entornos virtuales que generan reacciones particulares a la enfermedad para acostumbrar al usuario a las mismas en un entorno controlado<sup>4</sup>.



*Figura 2-2. Captura de Beat Saber<sup>5</sup>*

Pero, para que estos sistemas sean convincentes no se pueden usar HID tradicionales como el ratón y el teclado, ya que estos pueden llegar a ser disruptivos si el usuario no está altamente acostumbrado a los mismos. Por esto, ahora describiremos los distintos acercamientos de los HID avanzados para proporcionar una mayor inmersión en estos sistemas.

En la octava conferencia de realidad virtual de Toronto se realizó un estudio en 50 individuos sobre lo dicho en el párrafo anterior. Los resultados en el apartado de respuesta libre fue que solo el 10% de participantes indico que el uso de realidad virtual con teclado y ratón fue satisfactorio. Como se dice posteriormente en la discusión de los resultados, esto se debe a la necesidad de estar cerca de los dispositivos tradicionales, como el ratón y el teclado, y de la pulsación de teclas que no pertenecen al juego del experimento,

---

<sup>3</sup> Ver <https://www.explainthatstuff.com/virtualreality.html> .

<sup>4</sup> Ver <https://www.ncbi.nlm.nih.gov/pubmed/27400919> .

<sup>5</sup> Imagen extraída de <https://www.flickr.com/photos/playstationblogeurope/28767480908/>.

debido a la imposibilidad de ver lo que pulsamos con un casco de realidad virtual, y pueden llegar a romper el flujo del usuario con el entorno<sup>6</sup>.

Desde el controlador de las gafas de realidad virtual de High Tech Computer Corporation (HTC) hasta el “Leap Motion” se pueden contemplar los distintos acercamientos para definir un dispositivo de interacción que requiera la mínima preparación del entorno para que funcione correctamente el mismo.

La mayoría de los dispositivos de interacción avanzados se basan en el uso de luz infrarroja (IR) para percibir los movimientos del dispositivo, como por ejemplo el HTC Vive Controller. Allyn Malventano (2015) explica que estos dispositivos se basan en unos fotodiodos que están instalados en los distintos dispositivos que porta el usuario, casco y mandos, los cuales tienen fotodiodos que reciben luz IR para determinar su posición respecto a unas balizas colocadas en los extremos del área de uso, llamados “Lighthouse”, que son las encargadas de emitir estos rayos. En la figura 2-3 se puede apreciar una demo que hizo donde se ve al usuario y cómo funcionan los sensores (fotodiodos) al recibir la señal de las balizas.



Figura 2-3. Captura de estudio de Allyn Malventano con el HTC Vive.<sup>7</sup>

---

<sup>6</sup> Ver Crystal S. Maraj, S. G.-U. (2016). Preliminary Review of a Virtual World Usability Questionnaire.

<sup>7</sup> Imagen extraída <https://www.pcper.com/reviews/General-Tech/SteamVR-HTC-Vive-depth-Lighthouse-Tracking-System-Dissected-and-Explored/SteamV>.

También se pueden observar otros acercamientos que no usan la luz IR para ninguna de sus funcionalidades como el “PSMove” que usa el color del mando para que sea detectado por la cámara o el “Steam Controller” que usa la tecnología convencional de un controlador de consola, es decir, joysticks y botones, junto a un giroscopio y trackpads para mejorar la respuesta a los movimientos del usuario e intentar eliminar la necesidad de teclado y ratón en un ordenador.

El último acercamiento que se va a presentar en este apartado es el del “Leap Motion” que usa una cámara que permite detectar las manos encima del dispositivo sin necesidad de ningún controlador, pudiendo incluso detectar cuando se pretende hacer fuerza con las manos sin necesidad de pulsar botones como en la mayoría de estos dispositivos anteriores. Esto lo consigue gracias a las imágenes obtenidas por la cámara y un procesamiento interno para detectar las posiciones de estas imágenes.

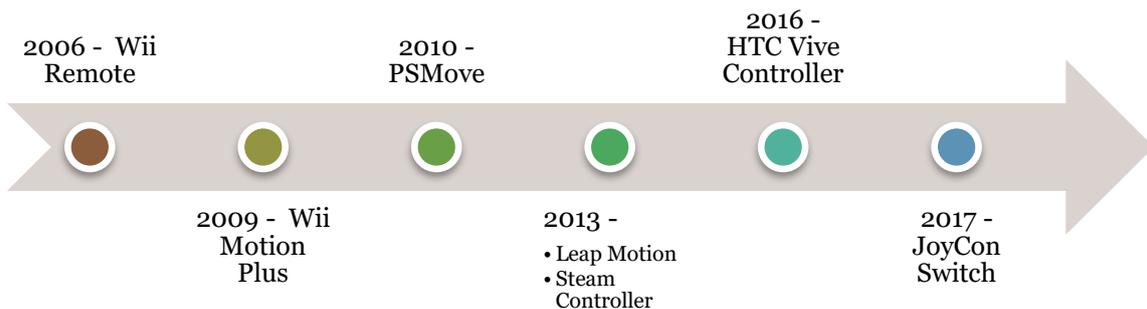
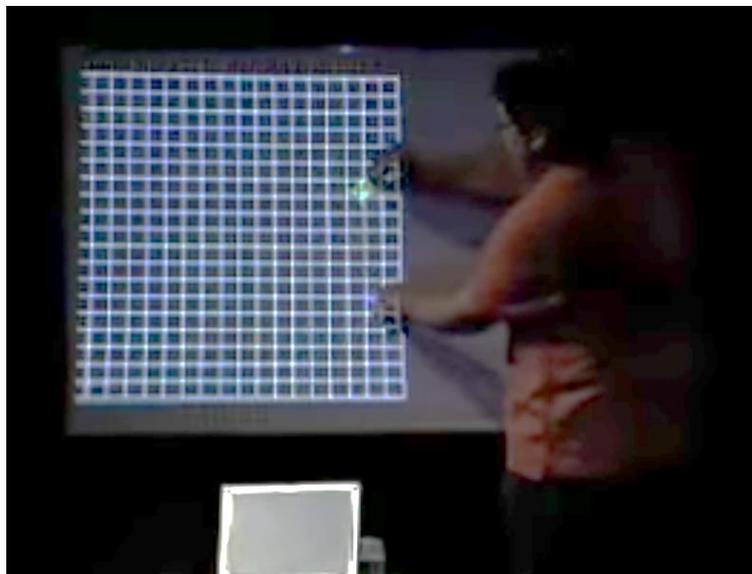


Figura 2-4. Línea tiempo dispositivos de interacción avanzados.

Pero la tecnología descrita anteriormente no se ha descubierto con el lanzamiento de estos dispositivos, como se puede ver en la Figura 2-4, sino que podemos remontarnos a dispositivos con más antigüedad que tienen usos similares, como el “Wii remote” que usa la luz IR emitida por la barra de la Wii para saber la posición a la que se apunta a la pantalla o el accesorio de este dispositivo, el “Wii remote plus” que incorpora varios giroscopios para saber en qué posición está el mando sobre la mano.

El nuevo dispositivo, lanzado por Nintendo, como remplazo del “Wii remote” es el “JoyCon” de la Switch, pero sin embargo este no ha generado grandes mejoras sobre el anterior dispositivo. La antigüedad y la popularidad que tuvo inicialmente el “Wii remote” han permitido que existan muchos estudios sobre “Wii remote” fuera de su consola nativa. Una de las primeras personas en realizar estos estudios fue Johnny Chung

Lee, que principalmente ha realizado proyectos que usan la cámara IR del “Wii remote” para rastrear puntos infrarrojos, como por ejemplo mover una imagen 3D con el movimiento de la cabeza o realizar una pizarra virtual usando de cursor un lápiz con un LED IR como se puede ver en la figura 2-5.



*Figura 2-5. Demostración de pizarra virtual por Johnny Chung Lee.<sup>8</sup>*

Pero los estudios que realmente marcan una diferencia importante para cualquiera de estos dispositivos avanzados son las que facilitan las comunicaciones a bajo nivel para crear una base para otros investigadores. De estas investigaciones hay una que se considera la quintaesencia de la ingeniería reversa del “Wii remote”, esta investigación fue llevada por la comunidad de WiiBrew<sup>9</sup> y documenta todas las comunicaciones “Bluetooth” que se pueden hacer con el mismo y el método con el que se puede vincular cualquier dispositivo al “Wii remote” mediante “Bluetooth”.

Gracias a esto, se han podido programar distintas librerías y “drivers” para facilitar la configuración inicial con el ordenador y poder centrarse en realizar juegos o aplicaciones que usen este dispositivo como veremos en apartados posteriores de este trabajo.

Otros dispositivos avanzados como el “PSMove” también se ha conseguido reutilizar fuera de su consola de origen, la PS3 y PS4, para usarse como controlador

---

<sup>8</sup> Imagen extraída de <http://johnnylee.net/projects/wii/> .

<sup>9</sup> Página principal en [http://wiibrew.org/wiki/Main\\_Page](http://wiibrew.org/wiki/Main_Page) .

alternativo en el PC, aunque las comunidades no han desarrollado nada tan difundido en parte por la escasa popularidad de este dispositivo fuera de su consola.

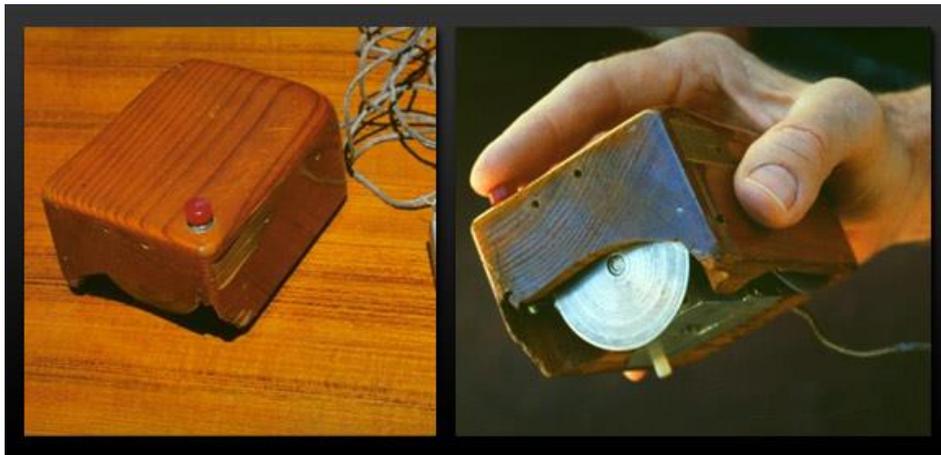
En los siguientes apartados de este capítulo veremos más especificaciones sobre cada uno de los dispositivos mencionados en este estado del arte y de los dispositivos clásicos como son el ratón y el teclado, para posteriormente ver porque se ha elegido el “Wii remote” porque principal dispositivo de estudio en este trabajo.

## 2.1. Análisis de dispositivos tradicionales

En este apartado veremos las características de los dispositivos de interfaz humana más antiguos y más usados en la sociedad para cualquier actividad que implique un PC.

### 2.1.1. Ratón

El primer prototipo de este dispositivo fue diseñado por Douglas Engelbart y fabricado por Bill English en 1964. Este prototipo, visto en la figura 2-6, constaba de dos ruedas perpendiculares conectadas a potenciómetros para medir el movimiento, cabe destacar que solo se podía mover en un eje a la vez y solo tenía un botón.



*Figura 2-6. Primer prototipo de raton de 1964.<sup>10</sup>*

En 1972, se hizo un ratón que tenía una bola conectada a las dos ruedas del primer prototipo, que permitía que el ratón pudiera transferir el movimiento de los dos ejes a la vez.

---

<sup>10</sup> Imagen extraída de <http://www.netprolive.com/img/first-mouse.jpg>

A partir de este momento solo se pueden destacar 2 cambios notables, el lanzamiento del primer ratón óptico patentado en 1988 por Lisa M. Williams y Robert S. Cherry y la adición de la famosa rueda central del ratón en 1995 en el ratón ProAgio.

### 2.1.2. Teclado

Aunque hay algunos precedentes de la máquina de escribir, consideramos la máquina de escribir como la primera máquina con relación directa con el teclado que usamos actualmente. Esto es porque la máquina de escribir fue el invento de Christopher Sholes con la ayuda de Samuel W. Soule y la idea de Carlos Glidden en 1868, que se puede ver en la figura 2-7, y la distribución QWERTY, usada en la mayoría de los teclados modernos fue creada por Christopher Sholes para su máquina de escribir en 1872, ya que si se escribía muy rápido en las primeras máquinas de escribir el mecanismo se podía atascar.



*Figura 2-7. Primera máquina de escribir.<sup>11</sup>*

En los años 70s se empezaron a crear los primeros teclados mecánicos, en los que cada activador de tecla, o “switch”, va soldada a una placa de forma individual, por encargo para grandes ordenadores. En los 80s IBM lanzó el icónico modelo M, visto en la figura 2-8.

Estos teclados mecánicos funcionan gracias a una matriz que está en la parte inferior del mismo. Cuando se pulsa una tecla el ordenador se cierra un circuito que emite un flujo

---

<sup>11</sup> Imagen extraída de <https://buffalohistorymuseum.files.wordpress.com/2015/01/typewriter.jpg> .

de electricidad, una vez hecho esto el procesador interno del teclado compara la tecla pulsada a un mapa de caracteres de solo lectura almacenado en la ROM del teclado. El sistema operativo, posteriormente puede sobrescribir la señal que le llega del teclado para tener un mapa de caracteres distinto.

Por último, en los años 90s para reducir el peso del teclado en los primeros ordenadores portátiles se crearon los teclados de membrana.



Figura 2-8. Teclado IBM modelo M.<sup>12</sup>

## 2.2. Análisis de dispositivos avanzados

Como hemos visto en el apartado anterior, los dispositivos tradicionales usan tecnologías muy básicas para cubrir las necesidades de las primeras interfaces gráficas en el caso del ratón y de la necesidad de conectar máquinas de escribir eléctricas en un ordenador, para poder comunicarse con los ordenadores sin la necesidad del uso de tarjetas perforadas.

Por esto, en este apartado veremos un análisis de los dispositivos avanzados presentados anteriormente y cómo buscan formas más naturales de comunicarse con los ordenadores que los dispositivos anteriores.

---

<sup>12</sup> Imagen extraída de

[https://upload.wikimedia.org/wikipedia/commons/f/f2/IBM\\_Model\\_M\\_1391403\\_keyboard.jpg](https://upload.wikimedia.org/wikipedia/commons/f/f2/IBM_Model_M_1391403_keyboard.jpg) .

### 2.2.1. Wii remote

Lanzado el 19 de noviembre de 2006 en Norteamérica junto a la consola Nintendo Wii, entre las características que lo hicieron destacar frente a los dispositivos de control de la competencia están el uso de IR y acelerómetro para detectar los gestos realizados con el mando y la dirección con la que se apunta a la barra sensor de la consola. Esta es una barra con LEDs IR que sirven de referencia a la cámara IR del mando “Wii remote”.



Figura 2-9. Consola Wii y accesorios.<sup>13</sup>

El sistema en el que va incluido este dispositivo se puede ver en la figura 2-9, está compuesto por la consola a la que se conecta la barra sensor para poder emitir la luz infrarroja que recibirá la cámara del mando. A este mando se le pueden añadir algunos accesorios, como el “nunchuck”, para mejorar los controles de algunos juegos gracias a su joystick y un botón trasero adicional.



Figura 2-10 Mando Wii remote<sup>14</sup>

---

<sup>13</sup> Imagen extraída de <https://www.konsolenkost.de/wii-konsole-weiss-mega-set-nr-06-inkl-remote-nunchuk-viel-zubehoer-gebraucht/a-9900041/>.

<sup>14</sup> Imagen extraída de <http://mod-center.com/wii-wii-u-accesorios/1714-comprar-mando-wii-remote-plus-protector-wii-negro-barato-accesorios-y-cables-oferta-mod-center.html>.

## Estudio hardware

Para saber más del “Wii remote” hay que recurrir a fuentes externas a Nintendo ya que las especificaciones del hardware no están publicadas, por suerte hay muchas fuentes externas que han investigado este dispositivo y la información contiene casi toda la que nos podría dar Nintendo.

Lo único que es visible desde el exterior, como se puede ver en la Figura 2-10, son sus 12 botones, de los cuales 4 forman un pad direccional, detrás de este pad hay un gatillo para el dedo índice. Los demás botones están repartidos por la parte frontal del dispositivo. También se pueden observar en la parte inferior del mismo 4 LEDs que nos indican el número de jugador en las consolas de Nintendo. Finalmente, en la parte de detrás, si levantamos la tapa, se puede observar que hay un botón para sincronizar el dispositivo, un PIN que sirve para la conexión con dispositivos externos a Nintendo y que el mando necesita 2 baterías AA para funcionar. También tiene un puerto de expansion para conectar los distintos accesorios de la consola, como el “motion plus” o el “nunchuk”.

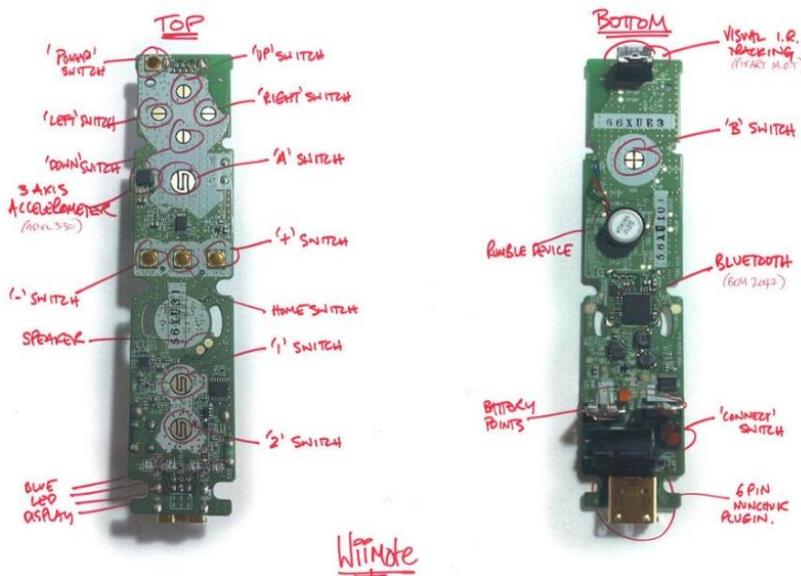


Figura 2-11. Interior Wii remote<sup>15</sup>

Investigando el interior de la carcasa del dispositivo, como se puede ver en el “Wii remote” desmontado de la figura 2-11, podemos observar algunos elementos que proporcionan retroalimentación a lo que haga en la consola el usuario como un altavoz

<sup>15</sup> Imagen extraída de <http://brettrufe.com/wii-mote-autops/>.

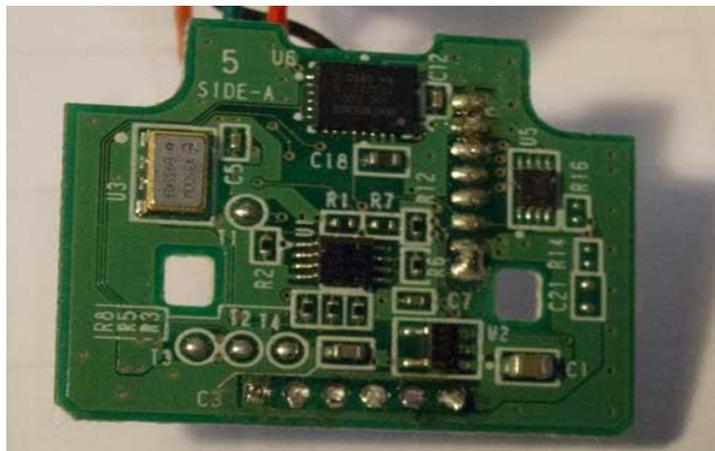
de 4KHz, un motor de vibración o la cámara infrarroja que cuenta con una resolución de 1024x768\_y un campo de visión de 45 grados.

La cámara sirve para detectar los LEDs infrarrojos de la barra de la Nintendo Wii y poder detectar a partir de la posición de estos LEDs a que parte de la pantalla se está apuntando, ya que la barra se debe colocar arriba o debajo de la pantalla. Cabe destacar que la barra, como veremos posteriormente, tiene más LEDs de los necesarios por si alguno se rompe o deja de funcionar correctamente.

Entre las características del “Wii remote” también tenemos un acelerómetro con una frecuencia de refresco de 100Hz y una sensibilidad de  $\pm 3G$  en los 3 ejes, comunicándose con 8 bits por eje<sup>16</sup>. Por último, cuenta con un chip Broadcom BCM2042 que permite las comunicaciones “Bluetooth” con otros dispositivos.

#### Accesorio Wii Motion Plus

Este complemento, visto en la figura 2-12, del “Wii remote” fue lanzado en junio de 2009 para permitir capturar movimientos complejos con el “Wii remote”, esto lo hace gracias a un giroscopio dual que captura el movimiento en 2 ejes y otro giroscopio para detectar la rotación del mando sobre sí mismo. Posteriormente se lanzó una versión del “Wii remote” que incorporaba este complemento por defecto.



*Figura 2-12. Interior Wii MotionPlus<sup>17</sup>*

---

<sup>16</sup> Información de acelerómetro extraída de [https://www.researchgate.net/figure/ADXL330-in-a-Wiimote\\_fig2\\_266597554](https://www.researchgate.net/figure/ADXL330-in-a-Wiimote_fig2_266597554) .

<sup>17</sup> Imagen extraída de <http://www.gloomy-place.com/wiigyro.htm> .

### 2.2.2. Switch JoyCon

Lanzado el 3 de marzo de 2017 con la consola Nintendo Switch es una evolución del Wii remote adaptada a las capacidades portátiles de la Nintendo Switch, como se puede ver en la figura 2-13.



Figura 2-13. Switch JoyCon<sup>18</sup>

A diferencia del “Wii remote”, la mayoría de los juegos de esta consola necesita de 2 “JoyCons” para jugar que, debido al diseño portable de esta consola, se pueden acoplar a los laterales de la consola para jugar sin necesidad de un soporte para la pantalla, como se puede ver en la figura 2-14.

También destaca una característica no muy usada por todos los juegos que han lanzado para esta consola, el JoyCon incorpora una “cámara de movimiento IR” en su JoyCon derecho. Esta cámara puede detectar objetos cercanos lanzando rayos IR y puede usarse de cámara nocturna y térmica, como demuestran una demo de Nintendo Labo<sup>19</sup>.



Figura 2-14. Consola Nintendo Switch con 2 “JoyCons”.<sup>20</sup>

---

<sup>18</sup> Imagen extraída de <http://i.imgur.com/ihAaMrO.jpg> .

<sup>19</sup> Demostración de la cámara y la demo en <https://kotaku.com/the-switch-joy-con-s-infrared-sensor-is-cooler-than-we-1822669059> .

<sup>20</sup> Imagen extraída de <https://www.falabella.com/falabella-cl/product/5587017/Consola-Switch-Azul-Roja#> .

### Estudio hardware

A diferencia del “Wii remote”, el “Joycon” no tiene cámara infrarroja ni altavoz, pero las otras características que comparte con el “Wii remote” las ha mejorado.

El acelerómetro sigue siendo en todos los ejes, pero la sensibilidad ha sido mejorada a  $\pm 8G$  con una frecuencia de actualización de 100 Hz, pero se puede incrementar hasta 416Hz por software, el giroscopio que lleva incorporado tiene una frecuencia de actualización de 208 Hz en los tres ejes<sup>21</sup>.

El MEMS, acrónimo de sistema micro electromecánico, controla el “Joycon” recogiendo datos del giroscopio y del acelerómetro a una frecuencia de 740Hz, o 1.35ms, pero solo manda a la consola la información cada 15ms, lo que indica que hay un procesamiento interno de estos datos recogidos antes de mandarlos<sup>22</sup>.

Por último, tenemos la “cámara sensor IR”, compuesta de 4 LEDs IR que emiten luz para rastrear los objetos o gestos que se hagan a una corta distancia del dispositivo<sup>23</sup>.

Aun no se tiene información completa del hardware por las políticas de Nintendo y la reciente salida a la venta de la consola, aunque ya hay comunidades, como ReSwitched, que se dedican a averiguar todos los detalles del hardware del “JoyCon”.

### **2.2.3. Leap Motion**

Lanzado el 22 de Julio de 2013 este dispositivo fue lanzado como un dispositivo que permite transformar el movimiento de la mano y los dedos en entradas del sistema al que esté conectado mediante USB.

El potencial de este dispositivo radica en que no se necesita contacto físico entre el usuario y el receptor, visto en la figura 2-15, por lo que una de las funciones que está intentando promover la empresa propietaria es el uso de este junto con un casco de realidad virtual o de realidad aumentada, que está siendo diseñado por la misma empresa

---

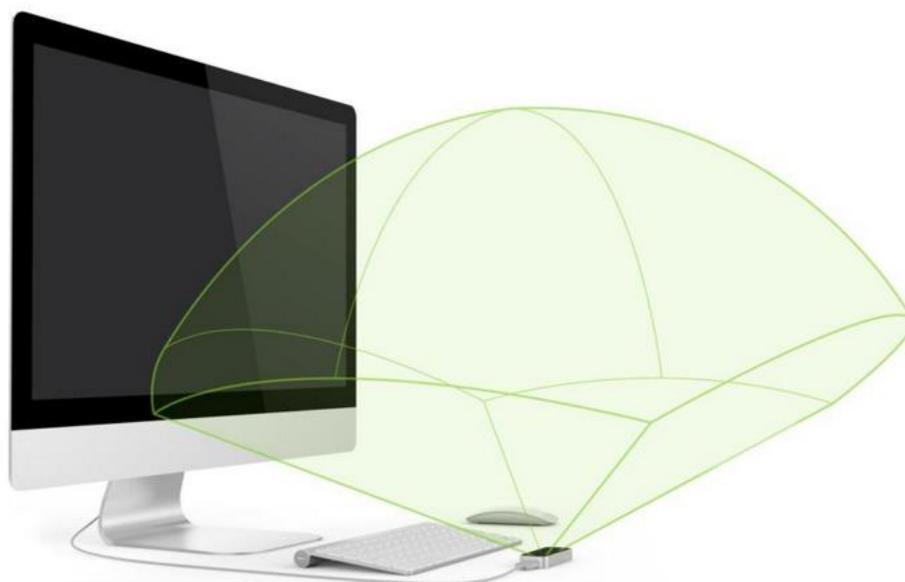
<sup>21</sup> Información del acelerómetro y giroscopio extraída de [https://github.com/dekuNukem/Nintendo\\_Switch\\_Reverse\\_Engineering](https://github.com/dekuNukem/Nintendo_Switch_Reverse_Engineering) .

<sup>22</sup> Cita extraída y traducida de <https://reswitched.tech/hardware/joycon> .

<sup>23</sup> Información cámara extraída de <https://es.ifixit.com/Desmontaje/Nintendo+Switch+Teardown/78263?lang=en> .



bajo el nombre de North Star y sus planos han sido publicados para permitir a cualquiera hacérselo en su casa<sup>24</sup>.



*Figura 2-15. Zona interacción LeapMotion<sup>25</sup>*

#### Estudio hardware

Los componentes del “Leap Motion”<sup>26</sup> son 2 cámaras y 3 LEDs infrarrojos, vistas en la figura 2-15, que permiten componer imágenes de un espacio alrededor del controlador, de 80 cm hacia arriba y los lados, limitado por el alcance de la luz infrarroja. Estas cámaras tienen un campo de visión de 150 grados y calcula la profundidad en el eje z, con esto pueden seguir el movimiento de los 10 dedos con una precisión de 1/100 mm y con una frecuencia de actualización de 200 Hz.

Con esto se genera una imagen de la que el software se ocupa de transformar en un espacio 3D, que gracias a las colisiones de los rayos infrarrojos en las manos detecta la posición de estas sobre este dispositivo.

---

<sup>24</sup> Cita extraída de <http://blog.leapmotion.com/north-star-open-source/> .

<sup>25</sup> Imagen extraída de <https://www.fabtolab.com/leap-motion-controller> .

<sup>26</sup> Información extraída de <https://learn.sparkfun.com/tutorials/leap-motion-teardown> .



Figura 2-16. Interior LeapMotion<sup>27</sup>

#### 2.2.4. PlayStation Move

Inicialmente lanzado por Sony en 2010 para competir el “Wii remote” sigue siendo utilizado en juegos de realidad virtual gracias a PlayStation VR. El dispositivo funciona de forma similar al de Nintendo ya que tiene sensores en el mando que detectan el movimiento y en la punta tiene una bola de color que es rastreada por una cámara en la consola llamada “PlayStation Eye”, que se puede ver en la figura 2-17.



Figura 2-17. Sistema de PS3 para PlayStation Move.<sup>28</sup>

---

<sup>27</sup> Imagen extraída de <https://www.fabtolab.com/leap-motion-controller> .

<sup>28</sup> Imagen extraída de <https://www.amazon.com/PlayStation-Eye-3/dp/B000VTQ3LU> .

Este dispositivo se empezó a vender junto a la consola PS3 y el PlayStation Eye pero posteriormente con el lanzamiento de la PS4, se ha empezado a vender con el sistema de realidad virtual de la misma, PlayStationVR y la nueva cámara, PS Camera. Para este último sistema no se necesita 1 PSMove sino 2, como se puede ver en la figura 2-18, ya que la realidad virtual suele usar las 2 manos del usuario para una mayor inmersión.



*Figura 2-18. Sistema de PS4 para PlayStation Move.<sup>29</sup>*

### Estudio hardware

Como podemos observar en la figura 2-19, en el interior del PlayStation Move (PSMove)<sup>30</sup> podemos encontrarnos un giroscopio de 3 ejes responsable de la detección del movimiento junto con el acelerómetro de 3 ejes. La frecuencia de actualización de los datos de los sensores con la consola es desconocida, pero sabemos que el casco de realidad virtual PlayStationVR tiene una velocidad de refresco de 90 a 120 Hz por lo que asumiremos que al ser un dispositivo más reciente este último marcará este límite el PSMove, por lo que en el resto del trabajo tomaremos esta velocidad de refresco para comparar este dispositivo con los demás.

---

<sup>29</sup> Imagen extraída de [https://i.blogs.es/b9ef3f/ps-camera/1366\\_2000.jpg](https://i.blogs.es/b9ef3f/ps-camera/1366_2000.jpg) .

<sup>30</sup> Información extraída de <https://www.ifixit.com/Teardown/PlayStation+Move+Teardown/3594> .



Figura 2-19. Interior “PSMove”<sup>31</sup>.

Aparte de lo anterior también se puede encontrar 10 botones repartidos por el dispositivo, que se pueden ver en los distintos perfiles del “PSMove” en la figura 2-20, un vibrador para retroalimentación al usuario y un indicador de estado. El color de la punta es intercambiable para la distinción entre jugadores y la comunicación con la consola es mediante tecnología Bluetooth.



Figura 2-20. PlayStation Move<sup>32</sup>

<sup>31</sup> Imagen extraída de <https://es.ifixit.com/Teardown/PlayStation+Move+Teardown/3594> .

<sup>32</sup> Imagen extraída de <http://forum.blu-ray.com/showpost.php?p=3012485&postcount=2> .

### 2.2.5. HTC Vive Controller

Dispositivo lanzado por Valve en 2016 para ser utilizado junto a los cascos de mismo nombre de realidad virtual. De los dispositivos presentados en este trabajo es el que requiere más preparación ya que requiere un espacio de 5x5 metros con un sistema de seguimiento llamado “Lighthouse”, que son unos dispositivos que lanzan luz infrarroja que recogen tanto los mandos como el casco para saber su posición relativa a estos.

Una de las cosas que llama la atención en el exterior es el uso de un trackpad, como se puede ver en la figura 2-22, usado también en otros dispositivos de la misma compañía, acompañado de 4 botones, para simular mejor el ratón de un ordenador. Este sistema de realidad virtual se completa con un casco de realidad virtual con una resolución de 2160 x 1200 y un campo de visión de 110 grados. Un ejemplo de cómo se diseñaría el espacio para este sistema lo vemos en la figura 2-21.

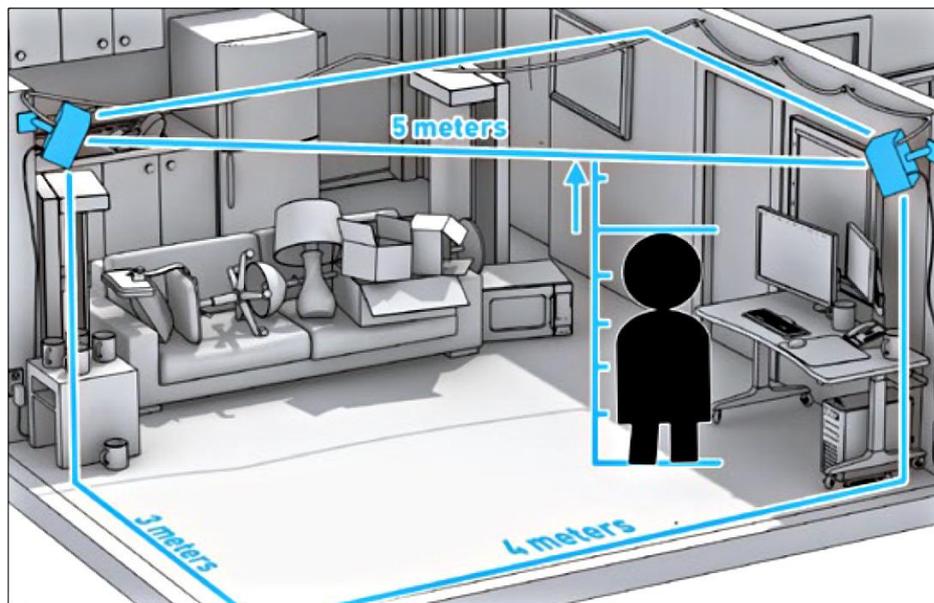


Figura 2-21. Posicionamiento Lighthouse para HTC Vive Controller.<sup>33</sup>

### Estudio hardware

En el interior, visto en la figura 2-23, nos encontramos 24 sensores que permiten determinar su posición exacta respecto a las dos estaciones “Lighthouse” y un

---

<sup>33</sup> Imagen extraída de <https://www.directcoops.com/product/complete-virtual-reality-system-htc-vive-fallout-4-vr-ready/>.

microprocesador Invensense MPU-6500 que incorpora un giroscopio y acelerómetro de seis ejes que reportan su información con una frecuencia de actualización de 250 Hz.



Figura 2-22. Interior HTC Vive Controller<sup>34</sup>

Este sistema de seguimiento emite luz infrarroja cada 8.33 ms, o con una frecuencia de 120 Hz, y también tiene incorporado un sistema de barrido con luz infrarroja que cada 16.6 ms intercala barridos horizontales y verticales, mientras que la otra estación hace los barridos contrarios a la primera.

Como hemos visto antes, este dispositivo tiene una frecuencia de actualización mucho mayor a la que funcionan estas estaciones. Esto es porque la posición calculada en cada actualización de estas estaciones reporta la posición exacta y entre actualizaciones de las estaciones, el HTC Vive Controller calcula internamente su posición aproximada con sus sensores.



Figura 2-23. HTC Vive Controller<sup>35</sup>

<sup>34</sup> Imagen extraída de <https://es.ifixit.com/Teardown/HTC+Vive+Teardown/62213> .

<sup>35</sup> Imagen extraída de <http://www.tomsguide.com/faq/id-3105264/install-custom-controller-skins-htc-vive-controllers.html> .

### 2.2.6. Steam controller

Dispositivo lanzado en 2013 por Valve que está orientado a controlar el PC sin necesidad de teclado o ratón gracias al modo “Big Picture” dentro de la aplicación Steam, que permite el uso de varias utilidades sin necesidad de acercarse al teclado o al ratón, como un navegador web, un reproductor de música o jugar a los videojuegos que se tengan registrados en la aplicación. Entre sus características más importantes se pueden ver las diferencias frente a otros dispositivos de consola parecidos como puede ser un mando convencional con solo joysticks y botones ya que el “Steam controller” cuenta con 2 “trackpads” para simular un ratón y giroscopio, como se puede ver en la figura 2-24. La conexión de este dispositivo es mediante “Bluetooth” al PC, gracias a una reciente actualización de este o mediante un cable USB.



*Figura 2-24. Steam Controller.<sup>36</sup>*

El hardware de Valve tiene grandes restricciones, ya que para usar todas las características del dispositivo tienes que tener una aplicación abierta en el PC, no existe todavía una comunidad que investigue a fondo el hardware de este dispositivo, visto en la figura 2-25, y Valve tampoco aporta nada sobre las características de sus dispositivos.

---

<sup>36</sup> Imagen extraída de <https://www.usgamer.net/articles/steam-controller-available-for-3499-once-more> .



Figura 2-25. Steam Controller<sup>37</sup>

### 2.2.7. Smart Phone

Para concluir con los dispositivos de interacción avanzados terminaremos analizando el dispositivo que intenta desbancar al teclado y al ratón de su trono.

Se considera Smartphone a un teléfono móvil que no solo tiene teclado, sino que tiene algunas características propias de un ordenador. El primer Smartphone fue presentado en 1992 bajo el nombre de Simon, visto en la figura 2-26. Por la empresa IBM, aunque no existía aun el termino Smartphone que fue acuñado por Ericsson en 1997. Simon cumple con la definición de Smartphone ya que era una fusión de PDA y móvil que tenía touchscreen para en vez de teclado para manejarse por el mismo.



Figura 2-26. Foto de IBM Simon.<sup>38</sup>

<sup>37</sup> Imagen extraída de <https://es.ifixit.com/Teardown/Steam+Controller+Teardown/52578> .

<sup>38</sup> Imagen extraída de <https://static.pisapapeles.net/uploads/2015/11/IBM-Simon-Worlds-first-smartphone-736x480.jpg> .

Pero no es hasta el lanzamiento del primer iPhone en junio de 2007, visto en la figura 2-27, que fue el primer Smartphone en implementar el acelerómetro y hacerlo popular, que estos dispositivos no tenían sensores para permitir una nueva forma de interacción con el entorno por parte del usuario.

Posteriormente al lanzamiento de este smartphone todas las compañías empezaron a lanzar móviles con este sensor y lo mismo pasó con el lanzamiento del iPhone 4 que popularizó el giroscopio con una presentación de Steve Jobs jugando al Jenga usando este sensor para visualizar los alrededores de la torre.



Figura 2-27. Foto de iPhone 1<sup>39</sup>

#### Estudio hardware

Por las características tan variables de este dispositivo según la compañía que lo fabrica no haremos un análisis de hardware del mismo y tan solo diremos que casi todos los smartphones de hoy en día cuentan con los siguientes sensores que pueden servir para mejorar las interacciones del usuario: “touchscreen”, giroscopio, acelerómetro, “GPS” y magnetómetro encargado de apagar la pantalla y el dispositivo con la cercanía de un imán y para medir el campo magnético de la Tierra y saber dónde está el norte.

El mayor defecto de estos dispositivos es la restricción de la forma por la pantalla que los hace poco ergonómicos para ser usados como un mando.

---

<sup>39</sup> Imagen extraída de <http://appmoviles.net/iphone-1-caracteristicas-y-especificaciones-tecnicas/> .

## 2.3. Análisis del software

En este apartado veremos, de forma superficial el software desarrollado para el ordenador de sobremesa de los dispositivos expuestos anteriormente ya sea de forma oficial o por comunidades. El software que se expondrá a continuación puede ser en forma de librerías, de interfaces, que se encargan de manejar las comunicaciones a bajo nivel con el dispositivo, o de aplicaciones, para simular el ratón y el teclado con estos dispositivos. Especialmente se mostrará interés por los motores de videojuegos ya que será lo que utilicemos posteriormente para las implementaciones en el capítulo siguiente.

Empezaremos hablando sobre los dispositivos que tienen soporte oficial en el ordenador y que se pueden usar sobre aplicaciones hechas específicamente por la empresa propietaria o por compañías externas con fines comerciales para estos dispositivos. Y posteriormente hablaremos de los dispositivos que no tienen este soporte oficial y las interfaces y aplicaciones que “intentan” darle este soporte por parte de terceros, en muchos casos, de forma gratuita y con fines de estudio.

El primer dispositivo con soporte oficial es el Steam Controller. Este fue lanzado para que los videojuegos, que tradicionalmente teclado y ratón, pudieran ser jugados gracias a sus 2 trackpads, que usan de teclado como en la figura 2-29, y un giroscopio que muchas veces se configura para maximizar la precisión. Aunque no tiene soporte oficial directo, este dispositivo es muy fácil configurarlo para manejar el escritorio, como se puede ver en figura 2-28, gracias al modo Big Picture de la aplicación Steam que te permite configurar los controles del Steam Controller para controlar el equipo, como si fuera el mapeo de un juego, y minimizando este modo se puede seguir usando esta configuración para controlar el equipo con bastante soltura.



Figura 2-28 y 2-29. Mapeo de Steam Controller<sup>40</sup> y Teclado partido por trackpads Steam Controller<sup>41</sup>.

<sup>40</sup> Imagen extraída de <https://www.pcp.com/image/view/65882?return=node%2F64704> .

<sup>41</sup> Imagen extraída de [http://cdn2.expertreviews.co.uk/sites/expertreviews/files/2015/12/steam\\_controller\\_onscreen\\_keyboard.png?itok=zem-r1Bh](http://cdn2.expertreviews.co.uk/sites/expertreviews/files/2015/12/steam_controller_onscreen_keyboard.png?itok=zem-r1Bh) .

El HTC Vive es un dispositivo para un casco de realidad virtual y por lo tanto las aplicaciones que permiten la simulación del escritorio del equipo necesitan de estos cascos. Esto lo que nos permite es no solo competir contra los HID tradicionales, sino llevarlos a una dimensión totalmente distinta que es la de la realidad virtual. Este dispositivo de realidad virtual nos permite controlar un escritorio 3D gracias a aplicaciones como VirtualDesktop, visto en la figura 2-30.



Figura 2-30. Virtual Desktop para HTC Vive.<sup>42</sup>

El dispositivo con menos popularidad con soporte oficial tenemos el “LeapMotion” que tiene aplicaciones como “Touchless” para controlar el equipo y se usa sobre todo para aplicaciones pequeñas en el ámbito de la educación sobre realidad aumentada y en medicina<sup>43</sup>.

Los tres dispositivos anteriores tienen librerías públicas para los motores de videojuegos más populares, como pueden ser Unity3D y UnrealEngine. También hay que destacar que estos 3 dispositivos simulan bien el ratón del equipo, pero solo el “Steam Controller” proporciona una fuerte alternativa al teclado, aunque la curva de aprendizaje sea muy alta para el usuario normal.

El mayor problema de los dispositivos sin soporte oficial es manejar las comunicaciones “Bluetooth” con entre el PC y estos dispositivos. Para esto, en todas las

<sup>42</sup> Imagen extraída de <https://i.ytimg.com/vi/bjE6qXd6Itw/maxresdefault.jpg> .

<sup>43</sup> Más ejemplos en <http://blog.leapmotion.com/category/education/> .

implementaciones que explicaré a continuación usan ligeras modificaciones de la librería hidapi<sup>44</sup> para facilitar estas comunicaciones.

El dispositivo para el que más se han realizado implementaciones en el pc es el “Wii remote”, para este dispositivo explicaremos en el capítulo siguiente algunas más a fondo, pero de forma superficial podemos decir que dispone de librerías bastante completas para Unity3D, como Unity-Wiimote<sup>45</sup> que se usará posteriormente para las implementaciones del estudio de este dispositivo, y tiene interfaces que permiten usar algunas de sus características para controlar el ordenador tanto en Windows como en Linux.

El siguiente dispositivo con falta de soporte oficial es el PSMove. Para este dispositivo podemos destacar paquetes para Unity3D y UnrealEngine que utilizan las capacidades de este dispositivo, aunque el de este último está obsoleto. Este dispositivo también tiene librerías para el estudio del comportamiento del dispositivo en el ordenador, como pueden ser psmoveapi<sup>46</sup> y PSMoveService<sup>47</sup>. El desarrollo de este último no está completo y planea lanzar paquetes para Unity3D y UnrealEngine en el futuro.

El último dispositivo que discutiremos brevemente en este apartado, debido a la reciente salida de este al mercado, es el “JoyCon” que ya cuenta con modificaciones de paquetes adaptados de la librería explicada anteriormente para el “Wii remote” para Unity3D<sup>48</sup>.

## 2.4. Elección del dispositivo a utilizar

Con lo visto en los apartados 2.2 y 2.3 y con el objetivo en mente de poder reutilizar alguno de estos dispositivos fuera de su ámbito normal, podemos extraer una tabla con la información de cada uno de los dispositivos para ver cual tiene la mejor relación hardware precio e investigarlo más a fondo.

Así pues, elaboramos la siguiente tabla comparativa contemplando las distintas características que nos permitirán determinar cuáles son los dispositivos más aptos para realizar las implementaciones para este estudio.

Para entender completamente esta tabla, hay que decir que la fila de precio de accesorios es el precio de los accesorios indispensables para el uso de las características

---

<sup>44</sup> Página principal en <https://github.com/signal11/hidapi> .

<sup>45</sup> Página principal en <https://github.com/Flafla2/Unity-Wiimote> .

<sup>46</sup> Página principal en <https://github.com/thp/psmoveapi> .

<sup>47</sup> Página principal en <https://github.com/cboulay/PSMoveService> .

<sup>48</sup> Página principal en <https://github.com/Looking-Glass/JoyconLib> .



del dispositivo, como puede ser el “PSEye” en el “PSMove” o el casco de realidad virtual del “HTC Vive”, sin el cual no se activan las “Lighthouse” del sistema y no se puede detectar bien la posición del dispositivo<sup>49</sup>. Otra explicación que hacer es el uso de BT como sigla de “Bluetooth” para ganar espacio en la misma, que la frecuencia de actualización del “PSMove” a 120Hz por las causas explicadas anteriormente en este capítulo y que la fila de rastreo de posición significa si lleva giroscopio o acelerómetro por defecto.

| Nombre   | Wii remote | Joycon   | Leap Motion | PSMove | Vive Controller    | Steam Controller | Smartphone          |
|--|------------|----------|-------------|--------|--------------------|------------------|---------------------|
| Precio   | 40€        | 44€      | 70€         | 45€    | 145€               | 55€              | +150€ <sup>50</sup> |
| Precio Accesorios                                  | 7€         | 0€       | 0€          | 10€    | 500€ <sup>51</sup> | 0€               | 0€                  |
| Rastreo posición integrado <sup>52</sup>           | X          | X        |             | X      | X                  | X                | X                   |
| Rastreo Posición Externo (Fuente IR) <sup>53</sup> |            |          | X           | X      | X                  |                  |                     |
| Frecuencia de actualización                        | 100 Hz     | 66.66 Hz | 200Hz       | 120Hz  | 250Hz              | 175Hz            | Variable            |
| Posición basada en luz IR                          | X          | X        | X           |        | X                  |                  |                     |
| Emulación ratón y teclado                          |            |          | X           |        | X                  | X                | X                   |
| Comunicación                                       | BT         | BT       | USB         | BT     | BT                 | BT               | BT                  |
| Soporte oficial en Windows                         |            |          | X           |        | X                  | X                |                     |
| Librerías públicas para Unity3D                    | X          | X        | X           | X      | X                  | X                | X                   |

Tabla 2-1. Comparativa dispositivos avanzados.

En la tabla 2-1 podemos ver que el dispositivo más barato es el “Wii remote”, seguido de cerca con el “Joycon” y el “PSMove”. Como el “Joycon” no tiene una librería de Unity 3D que utiliza todas las características de este se descarta de entre los candidatos para ser investigado a fondo, aunque cabe destacar que con el tiempo que lleva en el mercado este dispositivo se han hecho grandes avances para desentrañar como se

<sup>49</sup> Información extraída de <https://community.viveport.com/t5/Technical-Support/How-to-use-controllers-without-headset/td-p/11760>.

<sup>50</sup> Los móviles muy baratos suelen abaratar costes quitándoles sensores como el giroscopio.

<sup>51</sup> Precio de paquete completo, el casco no se vende por separado.

<sup>52</sup> Giroscopio y/o acelerómetro integrado.

<sup>53</sup> Rastreo basado en algún dispositivo externo al dispositivo que controla el usuario.

comunica con otros dispositivos y en el futuro próximo se podrán utilizar todas sus características.

Esto nos deja con dos candidatos, el “PSMove” y el “Wii remote”. Se podrían haber desarrollado los próximos apartados con cualquiera de estos dos ya que a nivel de hardware y de requerimientos son parecidos. Finalmente se ha decidido optar por el “Wii remote” por la popularidad de este, la posibilidad de encontrarlo en el mercado a menor precio que el mencionado en la tabla y la mayor información adquirida del “Wii remote” frente a la que se pueden encontrar del “PSMove”.

### 3. Estudio avanzado del Wii remote

En los próximos apartados se explicará cómo se puede conectar el “Wii remote” al PC, cómo fabricar una barra casera de la Wii para diseñar sistemas a medida y cómo comprobar la efectividad de esta frente a sus usos convencionales. Aprovecharemos estas implementaciones hechas en Unity3D para explicar cómo usar las funciones de las librerías ya creadas para controlar las funciones “normales” de un PC.

#### 3.1. Fabricación hardware necesario

Uno de los principales problemas que tenemos al querer utilizar todas las funcionalidades del “Wii remote” es la imposibilidad de conectar la barra infrarroja de la Nintendo Wii en el PC.

Esta barra, vista en la figura 3-1, sirve para generar las referencias para la cámara IR del “Wii remote” que servirán para que este dispositivo pueda saber a qué punto de la pantalla se está apuntando o saber cuándo se ha llegado a un borde de la pantalla, si esta barra está correctamente colocada encima o debajo de la misma. También sabemos que esta lo único que hace es darle referencias luminosas al “Wii remote” y que no se comunica de ningún método con el dispositivo al que está conectado.



Figura 3-1. Barra sensor Nintendo Wii. <sup>54</sup>

Con esto, podríamos cambiar el conector que viene por defecto a la barra con un USB y conectarlo al PC para que le proporcione la corriente necesaria para el funcionamiento de esta. Pero como estamos investigando este dispositivo a fondo

---

<sup>54</sup> Imagen extraída de <https://www.pcpur.com/image/view/68071?return=node%2F65117>.

optamos por abrirlo para ver que está compuesto en el interior y fabricar uno de cero para no necesitar comprar una barra si no se tiene la consola o si diseñar un sistema a medida.

Abriendo este dispositivo observamos que es un circuito compuesto por 6 LEDs infrarrojos y 2 resistencias de 22 ohmios. Fijándose en la Figura 3-2 se puede observar también como están ligeramente inclinados hacia los lados los LEDs de los laterales para que el “Wii remote” funcione desde un cono mayor respecto a la barra.

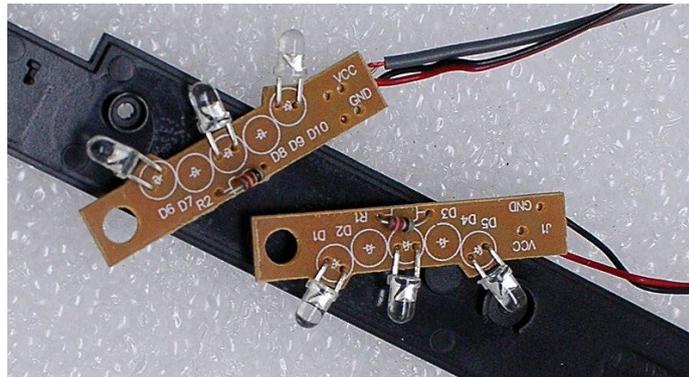
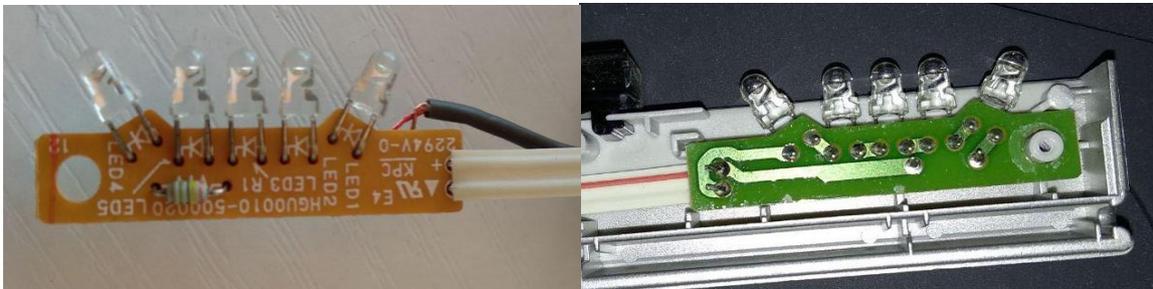


Figura 3-2. Interior Barra sensor Wii<sup>55</sup>

También se puede observar que hay otro modelo más reciente de la barra que consta de 10 leds en total y de 2 resistencias de 24,60 ohmios como se puede ver en las figuras 3-3 y 3-4, extraídas al abrir una barra sensor propia.



Figuras 3-3 y 3-4. Interior barra Wii segundo modelo.

Para este estudio decidimos hacerla totalmente inalámbrica, para poder posicionarla sin el problema de tener cables por el medio de la zona en la que estará el controlador del “Wii remote”.

---

<sup>55</sup> Imagen extraída de [https://www.reddit.com/r/mildlyinteresting/comments/4png8z/whats\\_inside\\_a\\_wii\\_sensor\\_bar\\_disassembly\\_photos/](https://www.reddit.com/r/mildlyinteresting/comments/4png8z/whats_inside_a_wii_sensor_bar_disassembly_photos/).

Lo primero fue diseñar el circuito eléctrico necesario. En primer lugar, partí del voltaje inicial de 6V que quería utilizar y de que los LEDs de la barra infrarroja tienen una longitud de onda de 950nm. Con estos datos diseñé el siguiente circuito en la aplicación web de CircuitLab:

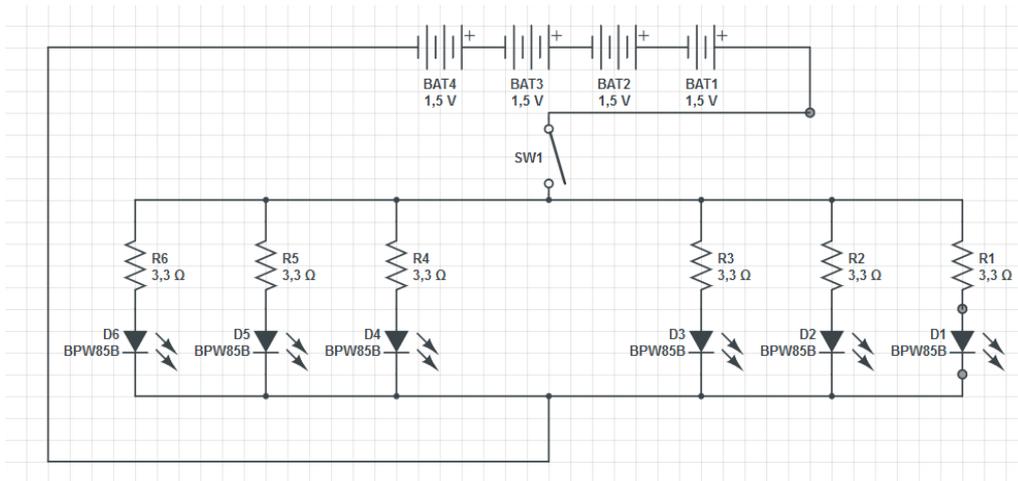


Figura 3-5. Circuito Barra sensor Wii.

Como veís en la figura 3-5 el circuito se puede hacer de forma casera con componentes comprados por menos de 10 euros en una tienda de electrónica.

El circuito lo he elaborado de cero recordando algunos conceptos básicos que ya había olvidado, como por ejemplo que la pata larga de los diodos es la positiva o como se reparte la intensidad en paralelo y en serie por los distintos componentes del circuito. También se han soldado entre sí todos los componentes del circuito para hacerla más robusta a los golpes y al tiempo, como se puede ver en la figura 3-6.

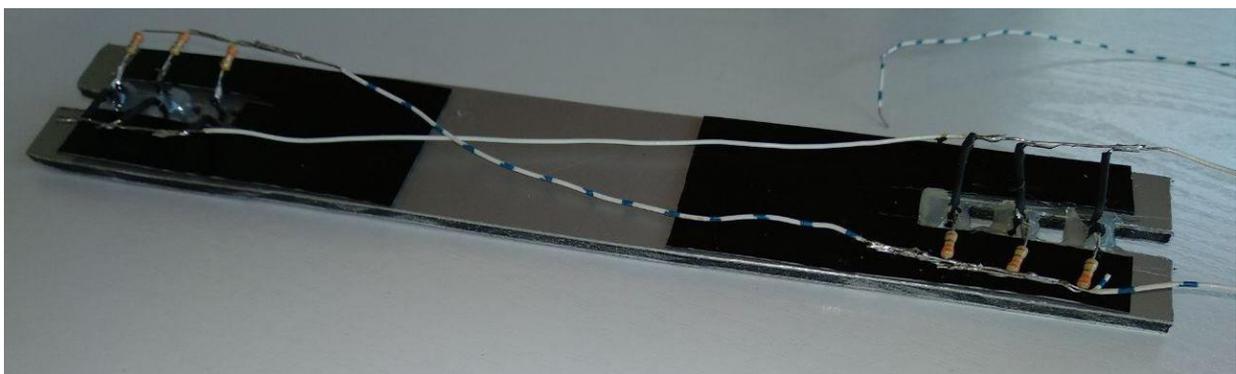


Figura 3-6. Circuito barra sensor.

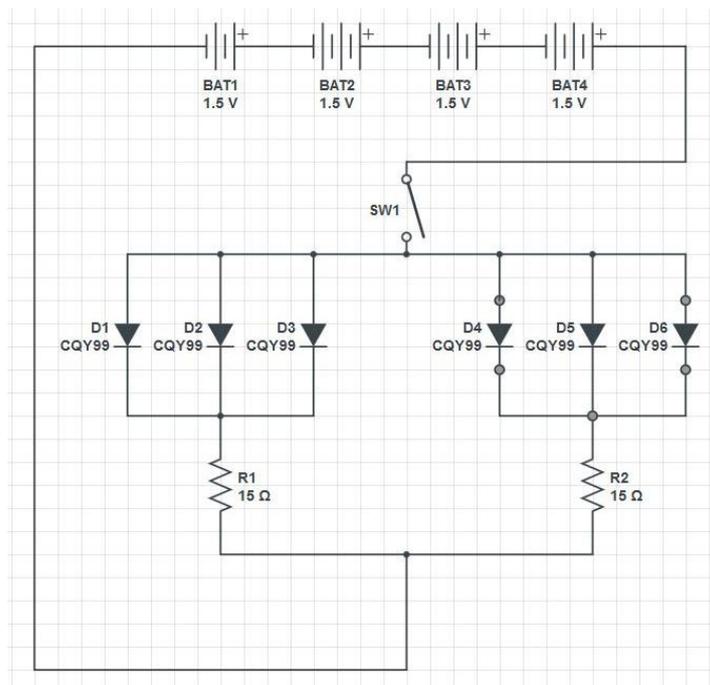
El último paso de la construcción ha sido crear la caja en la que se ubicará el circuito posteriormente. La caja de la figura 3-7 se ha realizado con aluminio con un recubrimiento interior aislante para no interferir con el circuito, también se le han hecho unos agujeros

para que se vean los LEDs desde el exterior, a diferencia de la barra original que usa plástico opaco.



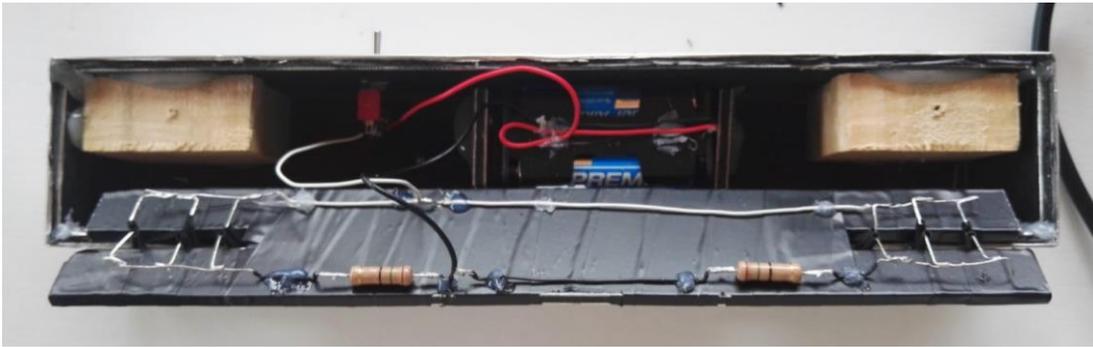
*Figura 3-7. Carcasa barra sensor.*

Finalmente, se le puso a prueba a este dispositivo dejándolo encendido varias horas seguidas y se fundieron algunos de los LEDs. Esto fue debido a que no habíamos considerado un voltaje de LED erróneo en el cálculo de las resistencias necesarias y la alta intensidad del circuito los había fundido.



*Figura 3-8. Esquema circuito versión 2.*

Para resolver esto se rehízo el circuito usando 2 resistencias de 15 ohmios, visto en la figura 3-8, con las que se cumplen los requisitos para que funcione correctamente, resultado final en la figura 3-9. Finalmente, una vez soldado todo el circuito de nuevo, se le dio un acabado en negro para evitar los posibles reflejos sobre el aluminio y enmascarar los defectos menores producidos durante el montaje, como vemos en la figura 3-10.



*Figura 3-9. Segunda versión circuito soldado.*

Con esto tenemos la posibilidad de elaborar artesanalmente una barra infrarroja necesaria para el uso del “Wii remote” fuera de su ámbito normal.



*Figura 3-10. Carcasa pulida y pintada.*

### **3.2. Estudio de protocolo de comunicación del Wii remote**

El protocolo usado por el “Wii remote”, como hemos visto en otras partes del trabajo, es el protocolo de corto alcance Bluetooth. Este protocolo puede soportar velocidades de transferencia de hasta 25 MBs por segundo en su versión 4.0, lo que es suficiente para soportar las comunicaciones con el “Wii remote”.

Para conectarnos a este dispositivo tenemos que poner el “Wii remote” en modo descubrimiento presionando el botón “sync” que está al lado de las pilas del dispositivo. Una vez hecho esto se puede establecer la conexión desde el host a este dispositivo sin necesidad de emparejarse con clave.

Sin embargo, si uno se quiere emparejar al “Wii remote” necesita, desde el host, enviar un comando HCI, de las siglas de Host Controller Interface que es un comando de transporte entre el host y el dispositivo objetivo en un protocolo Bluetooth, con el nombre de “Authentication\_Requested”. Una vez hecho esto el “Wii remote” pide una clave al host que deberá ser denegada para que pida un código PIN, el cual es la dirección binaria del “Wii remote” al revés. Esta dirección se puede obtener por varios métodos explicados en el apartado siguiente.

Una vez hecho esto se puede establecer una conexión al “Wii remote” a través del multiplexor de servicio, o PSM, 0x11 para la tubería de control y el PSM 0x13 para los datos usando el protocolo Bluetooth L2CAP, siglas de Logical Link Control and Adaptation Protocol (Protocolo de control y adaptación del enlace lógico).

Para mandar comandos al “Wii remote” se debe usar la tubería de datos con una de las funciones que presentaremos en la tabla 3-1 con un prefijo de (a2) que el “Wii remote” interpreta internamente como un reporte de output, mientras que este dispositivo enviara un reporte de input de respuesta con prefijo (a1).

No exploraremos en profundidad cada una de estas funciones ya que para la experimentación que se realizara en apartados posteriores se usara una librería, hidapi, que maneja estas comunicaciones internamente, para poder centrarnos mejor en el objetivo del trabajo.

| I/O | ID(s)     | Tamaño | Función  |
|-----|-----------|--------|--|
| O   | 0x10      | 1      | Desconocida  |
| O   | 0x11      | 1      | LEDs de los jugadores  |
| O   | 0x12      | 2      | Modo de reporte de datos                                       |
| O   | 0x13      | 1      | Activar cámara IR  |
| O   | 0x14      | 1      | Activar altavoz  |
| O   | 0x15      | 1      | Solicitud de información de status                             |
| O   | 0x16      | 21     | Escribir Memoria y Registros                                   |
| O   | 0x17      | 6      | Leer Memoria y Registros                                       |
| O   | 0x18      | 21     | Datos Altavoz  |
| O   | 0x19      | 1      | Mutear Altavoz   |
| O   | 0x1a      | 1      | Activar Cámara IR 2  |
| I   | 0x20      | 6      | Reporte Información  |
| I   | 0x21      | 21     | Lectura Memoria y Registros                                    |
| I   | 0x22      | 4      | Responde a un Reporte tipo Output, devuelve función resultada. |
| I   | 0x30-0x3f | 2-21   | Reporte de datos   |

Tabla 3-1. Funciones disponibles “Wii remote”.<sup>56</sup>

<sup>56</sup> Tabla extraída y traducida de <http://wiibrew.org/wiki/Wiimote> .



### 3.3. Estudio de conexión al ordenador

El método de conexión del “Wii remote” es la tecnología Bluetooth. Por tanto, será necesario un adaptador Bluetooth en el PC si no se tiene por defecto. Cabe destacar que los ordenadores de sobremesa no suelen tener de fábrica estos adaptadores, pero los ordenadores portátiles sí.

Para este estudio, como mi principal PC de trabajo es de sobremesa y no suelen llevar Bluetooth integrado por defecto me compre un dispositivo Bluetooth 4.0 de la marca iAmotus<sup>57</sup>.

Para la conexión del dispositivo al PC, el “Wii remote” se mostrará como un dispositivo de interfaz humana-computador estándar y se pedirá un PIN una vez inicie la conexión, explicado en el apartado 3.2. Esto no es necesario si no se quiere vincular el mando al PC para que siempre que se encienda el “Wii remote” intente conectarse al PC.

#### 3.3.1. Windows

Para conectar un “Wii remote” a un PC con sistema operativo Windows 10 se deberán de realizar los siguientes pasos:

1. Busca en el panel de control “Configuración de Bluetooth y otros dispositivos”.
2. Con la ventana maximizada, pulse “Dispositivos e impresoras”.
3. Pulsando el botón de agregar dispositivo te saldrá una ventana que buscará automáticamente los dispositivos.
4. Mientras busca el PC los dispositivos, se ha de pulsar los botones 1 y 2 en el “Wii remote” simultáneamente hasta que aparezca, puede ser necesario más de una búsqueda.
5. Seleccionar el “Wii remote” en la ventana de búsqueda y dejar el PIN vacío.

Cabe tener en cuenta que, para realizar la desconexión del dispositivo de este sistema operativo hay que quitar el dispositivo del menú de “Bluetooth” manualmente, en el menú del paso 1, sino se quedará emparejado incorrectamente el “Wii remote”

---

<sup>57</sup> Comprado de [https://www.amazon.es/Adaptador-iAmotus-Inal%C3%A1mbrico-Compatible-Auriculares/dp/B01J3AMITS/ref=sr\\_1\\_1?ie=UTF8&qid=1536076726&sr=8-1&keywords=iAmotus](https://www.amazon.es/Adaptador-iAmotus-Inal%C3%A1mbrico-Compatible-Auriculares/dp/B01J3AMITS/ref=sr_1_1?ie=UTF8&qid=1536076726&sr=8-1&keywords=iAmotus) .

aunque se apague el equipo, y habrá que desconectarlo la siguiente vez antes de poder conectar correctamente.

### 3.3.2. Linux

Para conectar el “Wii remote” a un PC con distribución Linux, una máquina virtual de la que se ofrece más detalles en el [anexo 1](#), se realizan los siguientes pasos a través de la consola “bash” que lleva por defecto el sistema operativo:

1. Instalar cwiid con el siguiente comando:

```
sudo apt-get install libcwiid1 lswm
```

2. Lanzar la GUI del programa recién instalado con el siguiente comando:

```
wmgui
```

3. Selecciona “connect” en el menú de archivo y pulsa los botones 1 y 2 del “Wii remote” hasta que el programa lo encuentre.

Una vez hecho estos pasos se puede observar en este programa las pulsaciones de los botones y los datos de la cámara.

Para finalizar decir que en el [anexo 2](#) se pueden observar capturas de los pasos explicados para conectar el “Wii remote” tanto en Windows como en Linux.

## 3.4. Estudio de aplicaciones y librerías que usan el “Wii remote”

### 3.4.1. HID Wiimote

En Windows no existe un controlador que sepa manejar las señales que envía el “Wii remote” al PC, por lo que el autor de este proyecto, Julian Löhr, decidió crear un controlador llamado HID Wiimote<sup>58</sup> que se instala automáticamente al conectar cualquier “Wii remote” y que permite usarlo como un mando normal, aunque con funcionalidad limitada del sensor infrarrojo. Junto a programas que mapean los inputs de controladores sin soporte oficial, como JoyToKey<sup>59</sup>, se puede usar “Wii remote” para jugar a juegos usando el “Wii remote” con su “Nunchuck”, soportado también por este controlador.

---

<sup>58</sup> Página principal en <https://www.julianloehr.de/educational-work/hid-wiimote/> .

<sup>59</sup> Página principal en <https://joytokey.net/en/> .

Este controlador también sirve para otros dispositivos de Nintendo como el “Wii Controller” lanzado para la Wii U.

### 3.4.2. CWiID

Como en Linux tampoco existe una forma de leer las señales del “Wii remote”, el paquete de librerías CWiID<sup>60</sup> sirve para reconocer y utilizar el “Wii remote” para distintas aplicaciones, como utilizarlo como controlador como el “driver” del apartado anterior, ver datos sobre el “Wii remote” como la dirección única que tiene cada “Wii remote” o las señales que recibe cuando pulsas teclas a partir de una sencilla UI o incluso usarlo como ratón del PC, como realizaremos en el siguiente capítulo.

### 3.4.3. Unity-Wiimote

Unity-Wiimote es un paquete creado por flafra2 para el motor Unity3D. Proporciona una interfaz para el uso del “Wii remote” y algunos de sus accesorios para realizar implementaciones de forma sencilla ya que casi todas las comunicaciones del “Wii remote” con el PC las tiene mapeadas gracias al uso de una versión modificada de hidapi. Este paquete incorpora una “demo” para ver los datos que recoge del “Wii remote” y proporciona clases para acceder a las funcionalidades del “Wii remote” y de sus accesorios.

---

<sup>60</sup> Página principal en <https://github.com/abstrakraft/cwiid> .

## 4. Casos de estudio

Después de estudiar todo lo expuesto anteriormente sobre las posibilidades que nos otorga reutilizar el “Wii remote” fuera de su ámbito natural, llega el momento de utilizar lo aprendido para demostrar en qué casos puede llegar a ser mejor que los dispositivos tradicionales, como el ratón y el teclado, para hacer cosas cotidianas como navegar por las carpetas de tu ordenador o para llevar al límite la compatibilidad fuera de su ámbito en videojuegos hechos en Unity3D con el paquete expuesto en el apartado 3.4.3.

### 4.1. “Wii remote” como ratón

En este caso comprobaremos como el “Wii remote” puede llegar a ser utilizado como un ratón, pero sin el defecto de necesitar una superficie donde apoyarse en el mundo real, en un sistema operativo Linux. Puesto que las alternativas de Windows en lo referente a drivers, HID Wiimote, utilizan el “Nunchuk” como joystick para eliminar la necesidad de la barra infrarroja.

Como en esta implementación quería comprobar la eficacia de la barra infrarroja hice la prueba usando CWiiD, introducido en el capítulo anterior, en un entorno con sistema operativo Ubuntu 18.04.

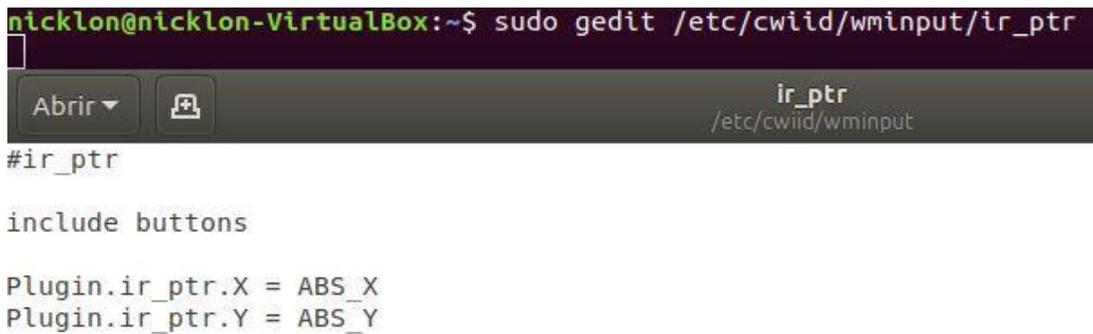
Para poder empezar el estudio hay que tener conectado un “Wii remote” al PC con los pasos descritos en el capítulo 3.2, una vez hecho esto habrá que seguir los pasos que se describen a continuación desde la consola de sistema operativo:

1. Averiguamos la dirección MAC del “Wii remote” pulsando los botones 1 y 2 del “Wii remote” y poniendo lswm en la consola “bash”.

```
nicklon@nicklon-VirtualBox:~$ lswm
Put Wiimotes in discoverable mode now (press 1+2)...
E0:0C:7F:8E:C2:2A
```

Figura 4-1. Comando lswm con la dirección MAC de un “Wii remote”.

2. Abrimos el fichero de la siguiente ruta, vista en la figura 4-2, y quitamos el carácter “~” de las siguientes líneas:



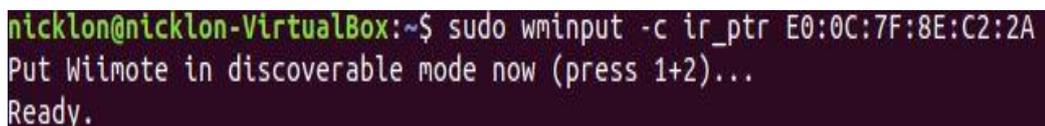
```
nicklon@nicklon-VirtualBox:~$ sudo gedit /etc/cwiid/wminput/ir_ptr
#ir_ptr

include buttons

Plugin.ir_ptr.X = ABS_X
Plugin.ir_ptr.Y = ABS_Y
```

Figura 4-2. Captura de comando y edición de fichero de configuración Cwiid.

3. Con esto ya es posible usar el “Wii remote” como ratón pulsando los botones 1 y 2 del “Wii remote” y usando el siguiente comando:



```
nicklon@nicklon-VirtualBox:~$ sudo wminput -c ir_ptr E0:0C:7F:8E:C2:2A
Put Wiimote in discoverable mode now (press 1+2)...
Ready.
```

Figura 4-3. Comando para convertir “Wii remote” en ratón.

Una vez hecho esto si desconectamos el ratón, podremos ver que podemos manejar el “Wii remote” cómo si fuera un ratón, ya que si se deja conectado el ratón puede haber “choques” entre las lecturas de dos “ratones” en un solo PC, usando el botón A para clic izquierdo y manteniendo este botón para hacer clic derecho.

## 4.2. Implementación con la barra infrarroja en Unity3D

El propósito de esta implementación es aprender a usar la librería hidapi a través de la interfaz de flafla2 Unity3D presentada anteriormente y crear un mini juego para probar la eficacia del “Wii remote” sin el uso del accesorio motion plus que usaremos en el apartado siguiente.

Para esto he decidido hacer un mini juego de tiro al blanco en el que el “Wii remote” apunta a la posición relativa de donde se está apuntando a la barra y con el botón A de este se puede disparar para conseguir puntos antes de que se acabe el tiempo. Para hacer esto he partido del modelo de “Wii remote” que lleva la interfaz de flafla2 y de la escena de demo de esta, vista en la figura 4-4, en la que se explican algunos básicos sobre el uso de esta, como aplicar la posición a la que se apunta a un “canvas” en la escena y ver donde se está apuntando.

A partir de esto, he aplicado una rotación sobre el “Wii remote” para dar la sensación de que se está moviendo el dispositivo como lo esté moviendo el usuario sin necesidad de usar el giroscopio ni el acelerómetro de este, explicación del código

utilizado en [anexo 3.1](#). También se ha hecho uso de los eventos de pulsación de botones del mando para hacer un menú de pausa, el botón de home, y el evento de disparo, disparado por el botón A del mando.

Con esto, ya tenemos la base de la implementación hecha que se ha concluido con una escena formada por un escenario parecido al visto en las ferias, y unos objetivos que se mueven por las filas de este.

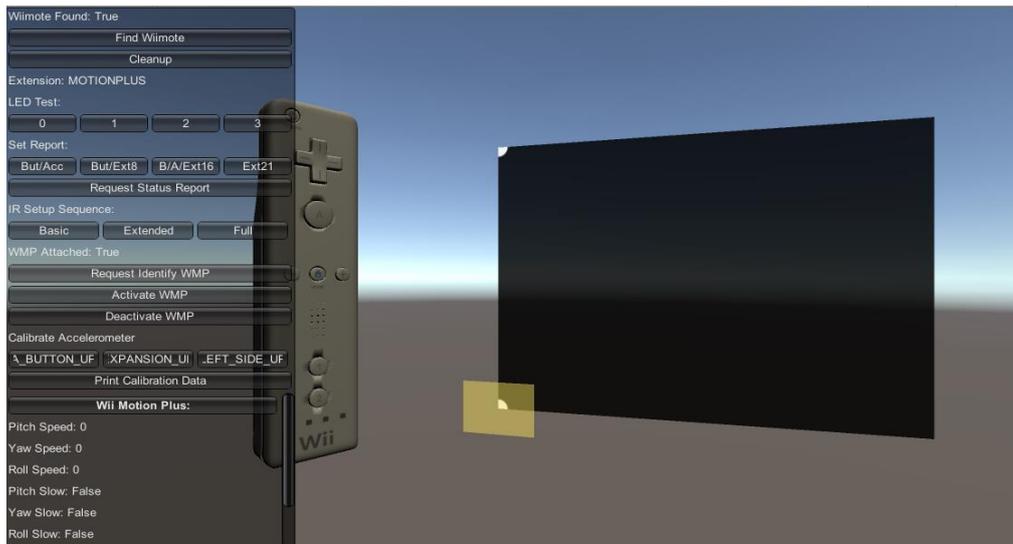


Figura 4-4. Escena demo Unity-Wiimote.

Estos objetivos son cubos primitivos que pueden ser de dos tipos, normales o especiales, según una variable que cambia aleatoriamente. La única diferencia entre ambos es el tamaño, la velocidad y los puntos que dan, ya que el objetivo especial es más pequeño, va más rápido y da más puntos, ejemplo de creación de objetivo en [anexo 3.2](#).

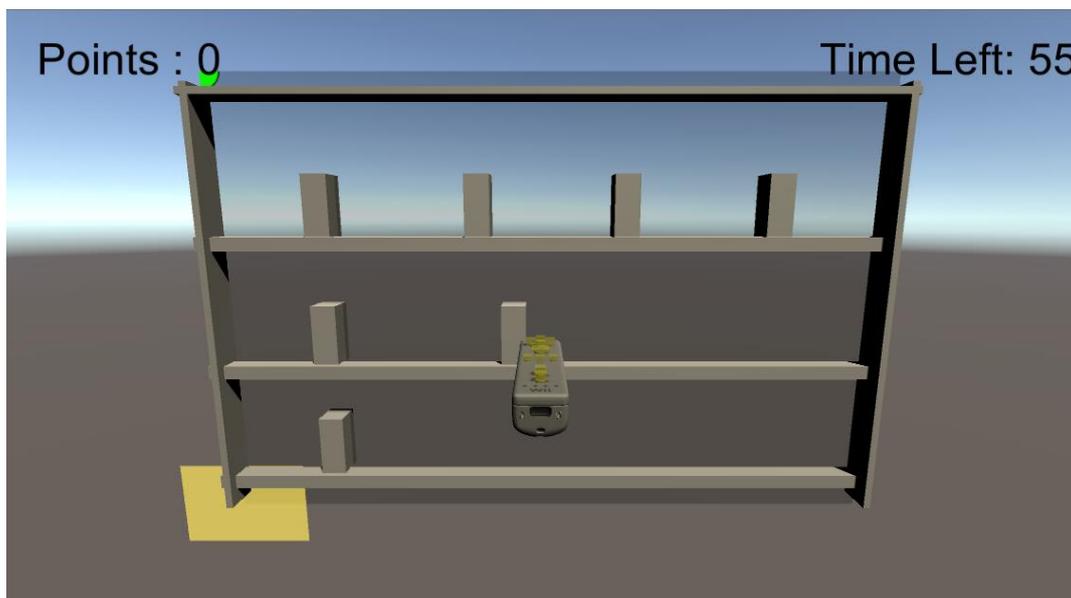


Figura 4-5. WiimoteShooter V0.8a.

A esto, y para finalizar la explicación de la implementación de esta escena se ha añadido un temporizador y un marcador con los puntos que se llevan en esta ronda, recibidos cuando los objetivos son destruidos a través de mensajes internos. En la figura 4-5 se puede ver una compilación de prueba con todo lo explicado en este apartado.

Para mejorar el aspecto visual del mismo se le añadió un terreno y materiales a los objetos, como podemos ver en la figura 4-6. Los dos puntos rojos cercanos al verde son la posición de los LEDs que toma como referencia la cámara, y el punto verde a donde se está apuntando en referencia a los LEDs anteriores.

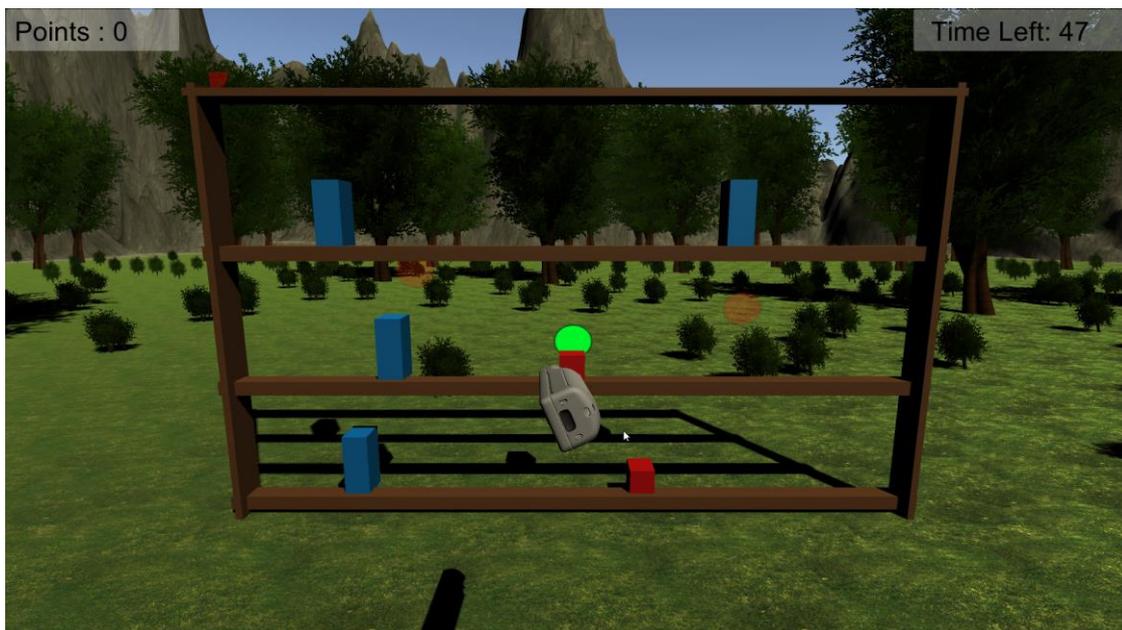


Figura 4-6. WiimoteShooter v1.0.

### 4.3. Implementación con el motion plus en Unity3D

Para este caso de estudio he decidido hacer un mini juego que utilice el movimiento de las manos del usuario de la forma más exacta posible. Para esto se han utilizado los giroscopios incorporados en el accesorio del “Wii remote” de motion plus, que viene incluido por defecto en las versiones más actuales del mismo.

Para realizar este mini juego se ha reutilizado la interfaz usará para el caso anterior, ya que en el menú de pausa, visto en la figura 4-7, usar solo el giroscopio podía llevar a confusión en como activar los botones, y por este motivo se ha creado una aplicación común con menús para moverse por estos dos casos sin tener que salir en este menú el giroscopio no funciona y se activa el método de rotación por la cámara infrarroja, y que permite reiniciar el mini juego o salir al menú para ver los dos mini juegos posibles.



Figura 4-7. WiimoteMenu V0. 8a.

En las pruebas iniciales de las lecturas dadas por dos “Wii remotes” con el accesorio plus incluido por defecto se podían observar bastantes errores de datos si se inicializaba este accesorio en mitad de un movimiento para permitir un comienzo del minijuego simultaneo a la carga de la escena, el “Wii remote” se quedaba girando sin parar y reaccionaba levemente a los movimientos del dispositivo. Para resolver este problema se ha implementado un temporizador que arranca con la escena y unas instrucciones que le muestran al usuario como iniciarlo correctamente. Estas instrucciones, extraídas del juego Wii Sports Resorts, le indican al usuario que deje el mando en una superficie plana con los botones mirando a la superficie, como se puede ver en la figura 4-8.



Figura 4-8. Instrucciones Wiimote Jedi v1.0

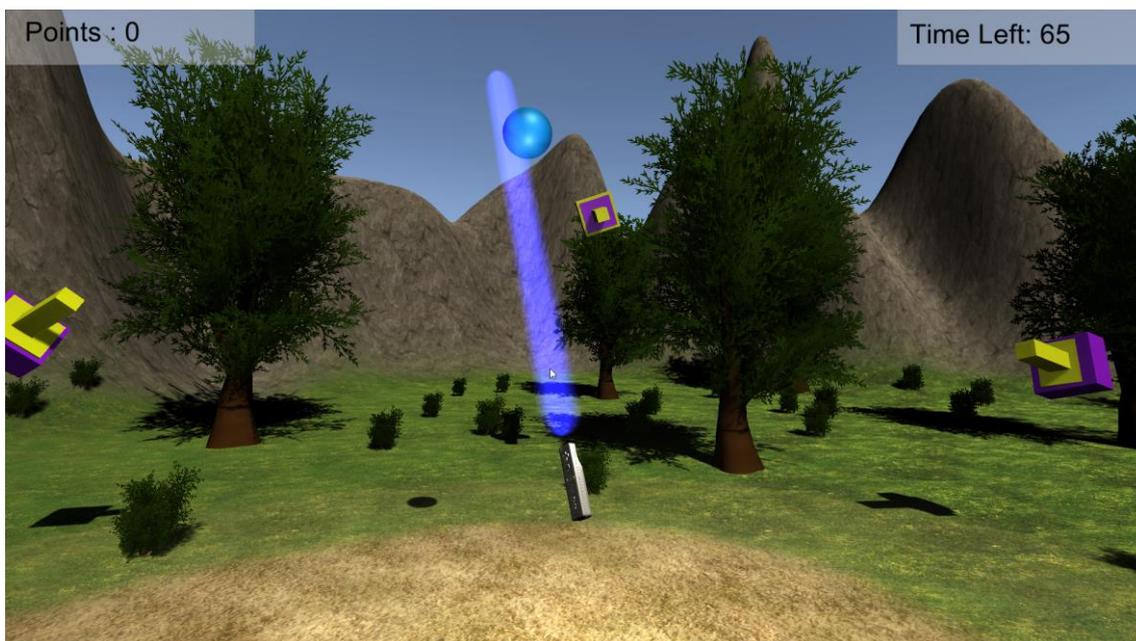
Una vez resuelto este problema, analizando nuevamente las lecturas recibidas por el giroscopio, nos encontramos con una desviación inicial que produce errores a largo plazo en las rotaciones del “Wii remote”, que rota en la escena de acuerdo con este vector. Por este motivo, se ha elegido restar este vector de desviación a todas las medidas que llegan del “Wii remote” si esta medida es exactamente igual a la desviación, lo que quiere decir que está quieto, para conseguir unos datos más fiables, explicación del código en [anexo 3.3](#).

Después de esto se llega a una versión estable de rotación del objeto en la escena de acuerdo con los movimientos de la mano del usuario, aunque con movimientos muy bruscos puede llegar a ser errónea, posiblemente debido a una baja frecuencia de actualización por parte del “Wii remote”. Para resolver estos casos en los que el accesorio quedaba siempre en una posición errónea se ha creado un método interno para reiniciar los datos del accesorio siguiendo las instrucciones del inicio y pulsando los botones 1 y 2 del “Wii remote”, explicación del temporizador de inicio y reinicio forzoso en [anexo 3.4](#).

Con este objeto configurado correctamente se pasa a analizar el resto de los componentes que enmarcan este mini juego para ver si a largo plazo el “Wii remote” sigue siendo efectivo. Para esto se ha añadido un láser al “Wii remote” para que se asemeje a una espada laser de la saga de películas de “Star Wars” y se han añadido unos cañones que lanzan pelotas al jugador.

Cada vez que la hoja de la espada toca una pelota esta desaparece y le otorga puntos al jugador. También se ha aplicado, como en el caso anterior, la posibilidad de que los cañones disparen pelotas más rápidas y pequeñas que otorgan más puntos, estas bolas al igual que los cubos del caso anterior se comunican con el marcador de puntos cuando son destruidas para sumar su valor.

Para acabar el mini juego, se ha añadido un temporizador que arranca cuando se configura el accesorio correctamente. En la figura 4-9 se muestra una captura del juego en su versión final.



*Figura 4-9. WiimoteJedi v1.0.*

Para la realización de estos mini juegos se han usado varios “assets” externos, como la espada laser, los árboles y arbustos y las texturas. Estos se explicarán con más detalles en el [anexo 4](#).

Estos mini juegos se han compilado solamente para Windows ya que Unity-Wiimote solo funciona en Windows y macOS, pero no disponemos de equipos con macOS.

#### **4.4.Implementación multijugador**

Para finalizar estos casos hemos realizado un mini juego en el que pueden participar hasta 4 “Wii remotes”. Este mini juego es parecido al WiimoteShooter, visto en el capítulo 4.2, pero sin modelos del “Wii remote” para cada participante para no estorbar la visibilidad de los demás, la foto del mini juego la podéis observar en la figura 4-10.

También se han modificado las funciones de búsqueda de “Wii remotes” y de limpieza al final del juego, vistos en el anexo 3.5. Para distinguir a los jugadores se han utilizado colores que se asignan al inicio del juego, estos colores se utilizan para cambiar del indicador de donde apuntan y de los paneles donde está su puntuación.

Para la creación del objetivo se han situado 4 paredes y se van creando objetivos normales y especiales, como en los restos de casos, que desaparecen cuando un jugador presiona el botón A encima de ellos.

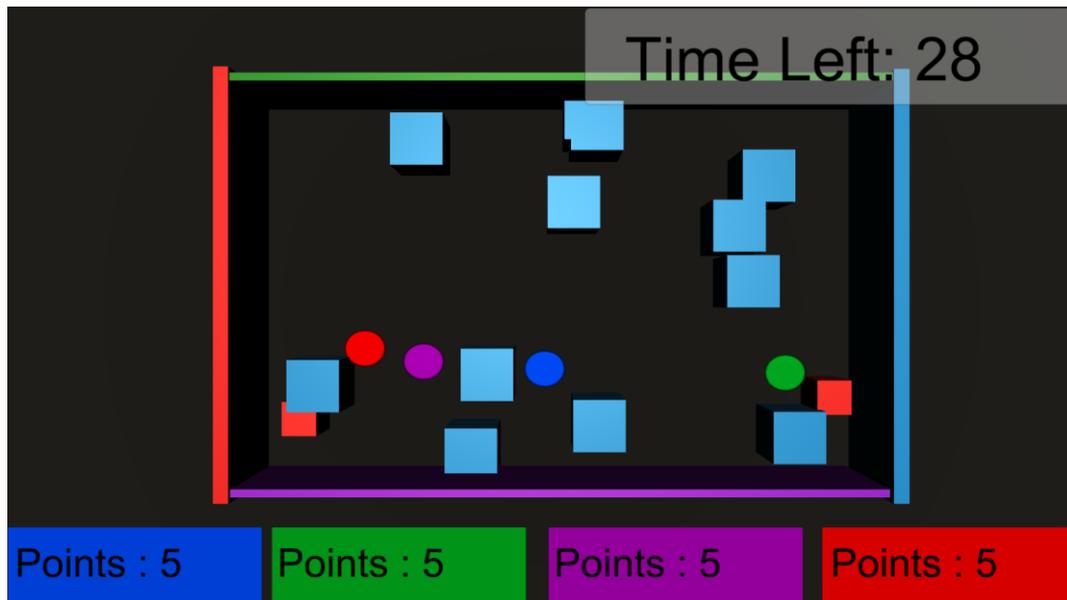


Figura 4-10. Wiimote Multiplayer v1.0.

El juego está limitado a 4 jugadores por el tamaño de los componentes en la interfaz gráfica, con un código más escalable se podría reutilizar fácilmente para  $n$  jugadores.

Por último, cabe indicar que debido a la complejidad de las conexiones no se ha podido reutilizar el menú de los otros mini juegos y se tiene que hacer un “build” específico de este nivel.

#### 4.5. Evaluaciones por parte de usuarios

Con lo realizado en el apartado anterior, comprobamos en usuarios reales el grado de aceptación de este sistema. Para esto, primero se les invita a usar cada uno de los casos vistos en este capítulo para posteriormente rellenar un formulario, visto en el [anexo 5](#). Como podemos observar en dicho anexo el formulario de Google<sup>61</sup> consta de 4 apartados: el primer apartado trata sobre información general relevante para el estudio como por ejemplo problemas de salud, la inversión que realizaría en un sistema de realidad aumentada, etc. El segundo apartado podemos encontrar una serie de preguntas relacionadas con la utilización de otros dispositivos avanzados, con el añadido del Oculus Rift. En el tercer apartado vemos preguntas sobre los casos, su usabilidad y efectividad. En el último apartado podemos observar unas preguntas a modo de conclusión donde se les pide que valoren las posibilidades que tendrán dispositivo en el mundo real.

---

<sup>61</sup> Enlace al formulario en [https://docs.google.com/forms/d/e/1FAIpQLSeOuiOnJeptqzEvUa\\_KDYfVL-JOdY3TN1duEDtNMwjMOAtQ-Q/viewform](https://docs.google.com/forms/d/e/1FAIpQLSeOuiOnJeptqzEvUa_KDYfVL-JOdY3TN1duEDtNMwjMOAtQ-Q/viewform).

Los resultados de este formulario se analizarán de forma conjunta sin desvelar los nombres de los usuarios. En el anexo 6 se puede observar las gráficas de los resultados obtenidos de los 8 sujetos que aceptaron voluntariamente a participar en esta evaluación, de los cuales solo se analizarán los más relevantes en este capítulo.

Antes de realizar las pruebas de los casos presentados se les pregunta la experiencia previa con el “Wii remote” para dar una pequeña formación sobre el mismo y se les avisa de las posibles desviaciones del giroscopio ante los movimientos bruscos en el mini juego que usa el motion plus. Después de esto, los sujetos prueban cada uno de los casos para, posteriormente pedirles que rellenen un formulario de forma anónima, pidiéndoles el correo para evitar duplicados, pero no pudiendo ver los resultados individuales de los sujetos. Una vez explicado el formulario y el procedimiento seguido para su realización procederemos a analizar los resultados más relevantes de este formulario.

Una de estas preguntas sería sobre su límite de presupuesto que para la mayoría es de menos de 600 euros, en las preguntas sobre la experiencia en dispositivos en el que la gente tiene más experiencia es el “Wii remote”, esto puede ser debido a la popularidad que tuvo en España este dispositivo cuando se lanzó<sup>62</sup>. No sorprende que la gente no tenga experiencia con los sistemas más caros, el Oculus Rift o el HTC Vive debido su alto coste, en relación con las respuestas obtenidas a la primera pregunta. A las preguntas de los casos realizados en este capítulo, después de experimentar el control del ratón a través del “Wii remote”, la mayoría de gente sigue prefiriendo el ratón. Aunque en un entorno más lúdico, como los dos otros casos, WiimoteShooter y WiimoteJedi, la gente ha valorado positivamente la experiencia, aunque la eficacia del giroscopio ha sido un poco peor a la de la cámara IR del “Wii remote”, como se indica en la pregunta de qué dispositivo se prefiere para esta actividad de la conclusión.

Finalmente, la gente valora positivamente la posibilidad de usar este dispositivo fuera del PC, no usarían teclado y ratón si hubiera una alternativa capaz de equipararse a estos y, por último, todos los encuestados compraría un sistema similar para el PC por 40 euros.

---

<sup>62</sup> Mas información en <http://www.elmundo.es/elmundo/2010/02/11/navegante/1265884279.html> .

## 5. Conclusiones

Después de realizar un estudio sobre de los dispositivos clásicos y los avanzados podemos decir que los objetivos marcados en el capítulo 1-2 se han cumplido. A continuación, recordaré cada uno de los objetivos que planteé en dicho capítulo y cómo se han resuelto.

El primer objetivo consiste en describir que es un dispositivo de interacción avanzado y en lo que se diferencia de los clásicos, como el ratón y el teclado. Para conseguir este objetivo, se ha buscado una definición de un término que aún está en un estado de evolución constante y no se ha asentado y mostrar sus diferencias con los clásicos.

Después de esto, se han analizado algunos de los dispositivos de interacción avanzados más innovadores de este campo, tanto su hardware como software, para ver sus diferencias con los clásicos y poder elegir posteriormente el mejor dispositivo de interacción avanzado, el “Wii remote” en este estudio, por su número de características, de recursos posibles y por su bajo precio.

Una vez elegido el “Wii remote” nos adentramos en explorar los requisitos para conectarlo al PC. Uno de estos requisitos es como comunicarnos con protocolo “Bluetooth”, desde el momento en el que lo detecta hasta como enviar funciones predeterminadas a este. Una vez conectados hemos diseñado y fabricado desde cero la barra sensor de la Nintendo Wii, con un intento fallido en este proceso, para poder ser usado en un futuro esta barra en otras instalaciones sin necesidad de consola Nintendo Wii.

Con todos los requisitos estudiados y preparados, llegó el momento de estudiar como conectarnos al PC y posteriormente como desarrollar una serie de mini juegos que exploran las posibilidades que nos brindan los sensores del “Wii remote”, creando no solo estos mini juegos sino una base sólida para futuros trabajos que lleven al límite este dispositivo. Y sin olvidarnos de poder simular un ratón en el sistema operativo Linux a través de una aplicación que puede ser útil para sistemas remotos en ámbitos lúdicos o comerciales.

Finalmente, lo explicado en este trabajo se ha probado en sujetos reales para comprobar que este sistema goza de mucha popularidad y verlo reutilizado en el PC con un buen rendimiento ha sido recibido de buen grado por los encuestados, que comprarían un sistema parecido debido a su coste reducido respecto a la competencia. Otra conclusión

importante de estos formularios ha sido que los usuarios reemplazarían el teclado y el ratón si existiera una alternativa viable.

Como conclusiones de este trabajo se pueden destacar la creación de un sistema relativamente barato para poder divulgar las posibilidades que nos otorgan estos dispositivos y que lleguen a la mayor cantidad sea posible, para que el día de mañana puedan estar interesados en esto e investigar más sobre este campo, obteniendo sistemas más baratos y menos privados.

## **5.1. Valoración personal**

Creo que este trabajo puede servir a mucha gente como base por el estudio en profundidad que se hace de los dispositivos de interacción avanzados. También creo que alguien que quiera realizar un mini juego o aplicación en Unity3D puede partir fácilmente de los mini juegos míos ya que incluyen código para regular los posibles errores en las mediciones de este dispositivo.

También tengo que destacar que gracias a algunas asignaturas este trabajo ha sido posible ya que han tenido influencia directa en mi decisión de proponer y hacer este trabajo. Entre estas puedo mencionar la asignatura de "CAD CAM", en la que hice un mini juego con el "Leap Motion" como proyecto final de esta; "Introducción a gráficos de usuario", que me dio las ganas de aprender Unity3D al haber realizado mini juegos en OpenGL y, por último, "Introducción de Medios Digitales", en la que aprendí lo que se podía hacer en un ordenador sin el teclado ni el ratón y a utilizar lo aprendido en la carrera para desarrollar un proyecto de elección propia.

En lo personal creo que en este trabajo he aprendido bastante de estos dispositivos y se ha reafirmado mi sentimiento de que en estos se halla el futuro una vez los programadores descubran sus posibilidades y las sepan explotar para realizar aplicaciones que puedan substituir al teclado y al ratón de un PC, ya que yo creo que este paso es fundamental para el posible avance de estos dispositivos en cualquier ámbito.

## 6. Trabajos futuros

En este apartado discutiré algunos posibles trabajos que pueden derivar de este usando tanto lo explicado en este estudio como lo realizado en Unity3D como base para la programación.

### 6.1. Uso de la cámara IR con el giroscopio para mejorar la detección del movimiento del “Wii remote”

Para mejorar la precisión del giroscopio se puede utilizar la rotación por IR que se usaba en el mini juego del tiro al blanco para afianzar las medidas de este sensor. Esto se basa en el sistema que usa el HTC Vive con las “Lighthouse” explicado en el apartado 2.2.5.

### 6.2. Integración con Google Cardboard como sistema de realidad virtual de bajo coste

Las Google Cardboard son las gafas realizadas por Google como alternativa barata de realidad virtual para la plataforma móvil. Hay aplicaciones en el ordenador que pueden simular un casco de realidad virtual, TriniusVR por ejemplo, como el HTC Vive.

Con este casco simulado correctamente, se podrían realizar aplicaciones en Unity3D usando lo visto en capítulo 4 para integrar el “Wii remote” en algún juego que use un casco de realidad virtual.

Con esto se crearía un sistema de realidad virtual por menos de 60 euros, ya que las Google Cardboard son muy baratas en el mercado por su distribución pública de cómo construirlas.

### 6.3. Múltiples barras para un mejor control en realidad virtual

Una barra sensor de la consola Wii, o casera como hemos visto en el capítulo 3.1, puede cubrir bastante bien la zona a la que se esté apuntando. Esto reduce muchas las posibilidades de movimiento rotatorio del usuario en los mini juegos realizados para este “Wii remote”.

Por esto, en este apartado se propone la realización de un sistema de 4 barras, una en cada eje cardinal, para poder cubrir toda el área alrededor del usuario. Para resolver el problema de que el “Wii remote” no sabe cuándo debe cambiar de barra, o de eje cardinal, se podría usar el acelerómetro de este o realizar alguna forma de comunicación entre las

Nicolás Tortosa Cerezo

barras y la plataforma para la que se realice esta aplicación para poder detectar hacia donde mira el usuario.

Se podrían incorporar a estas barras una cámara IR y que el usuario portara un LED en el “Wii remote”, como en algunos experimentos de Chung Lee vistos en el apartado 2-1, aunque esto incrementaría bastante el coste del sistema.

## 7. Referencias

- Anónimo. (Marzo de 2010). *MCVUK : Full Tech Specs PlayStation Move*. Obtenido de <https://www.mcvuk.com/development/full-tech-specs-playstation-move>
- Anónimo. (s.f.). *Wiibrew - Wiimote*. Obtenido de <http://wiibrew.org/wiki/Wiimote>
- Augment. (s.f.). <https://www.augment.com/>.
- Bellis, M. (24 de Febrero de 2018). *ThoughtCo*. Obtenido de <https://www.thoughtco.com/history-of-the-computer-keyboard-1991402>
- Colgan, A. (9 de Agosto de 2014). *Leap Motion Blog*. Obtenido de <http://blog.leapmotion.com/hardware-to-software-how-does-the-leap-motion-controller-work/>
- Computer Hardware Explained. (s.f.). *Computer Hardware Explained*. Obtenido de <https://www.computer-hardware-explained.com/history-of-computer-keyboards.html>
- Crystal S. Maraj, S. G.-U. (2016). Preliminary Review of a Virtual World Usability Questionnaire. En R. S. Stephanie Lackey, *Virtual, Augmented and Mixed Reality: 8th International Conference, VAMR 2016, Held as Part of HCI International 2016, Toronto, Canada, July 17-22, 2016*. (págs. 35-45). Springer.
- Detwiler, B. (27 de Septiembre de 2010). *ZDNet : PSMove Teardown*. Obtenido de <https://www.zdnet.com/pictures/playstation-move-navigation-controller-teardown/>
- Developers Android. (14 de Agosto de 2018). *Developer.Android*. Obtenido de [https://developer.android.com/guide/topics/sensors/sensors\\_overview](https://developer.android.com/guide/topics/sensors/sensors_overview)
- Edwards, B. (9 de Diciembre de 2008). *Macworld*. Obtenido de <https://www.macworld.com/article/1137400/input-devices/mouse40.html>
- Feiner, T. H. (2004). Mobile Augmented Reality. En H. A. Karimi, *Telegeoinformatics: Location-Based Computing and Services* (págs. 392-421). CRC Press.
- Follet, J. (20 de Octubre de 2008). <https://www.uxmatters.com>. Obtenido de <https://www.uxmatters.com/mt/archives/2008/10/toward-a-more-human-interface-device-integrating-the-virtual-and-physical.php>

- Horde, M. (27 de Junio de 2013). *Leap Motion Teardown*. Obtenido de <https://learn.sparkfun.com/tutorials/leap-motion-teardown>
- HTC Corporation, Valve. (2016). *Manual usuario VIVE*. Obtenido de [https://www.htc.com/managed-assets/shared/desktop/vive/Vive\\_PRE\\_User\\_Guide.pdf](https://www.htc.com/managed-assets/shared/desktop/vive/Vive_PRE_User_Guide.pdf)
- HTC Corporation, Valve. (s.f.). *Página oficial Vive*. Obtenido de <https://www.vive.com/us/accessory/controller/>
- JJules. (18 de Enero de 2017). *Foro Unity*. Obtenido de <https://forum.unity.com/threads/unity-and-the-accelerometer-vs-the-gyroscope-a-complete-guide.451496/>
- Kreylos, O. (25 de Mayo de 2016). *Doc-Ok*. Obtenido de <http://doc-ok.org/?p=1478>
- Leap Motion. (6 de Junio de 2018). *Blog Leap Motion*. Obtenido de <http://blog.leapmotion.com/north-star-open-source/>
- Lee, J. C. (15 de Julio de 2008). Hacking the Nintendo Wii Remote. *IEEE Pervasive Computing*, págs. 39-46.
- Luis, A. J. (7 de Marzo de 2013). *Gizmodo*. Obtenido de <https://es.gizmodo.com/la-historia-del-raton-en-seis-disenos-663577893>
- Malventano, A. (5 de Abril de 2016). *PC Perspective*. Obtenido de <https://www.pcper.com/reviews/General-Tech/SteamVR-HTC-Vive-depth-Lighthouse-Tracking-System-Dissected-and-Explored/SteamV>
- Mazzanti, D. (2 de Diciembre de 2015). *Steam Controller TearDown*. Obtenido de <https://www.ifixit.com/Teardown/Steam+Controller+Teardown/52578>
- Mccarty, B. (6 de Diciembre de 2011). *The next web*. Obtenido de <https://thenextweb.com/mobile/2011/12/06/the-history-of-the-smartphone/>
- Newgenapps. (22 de Junio de 2017). *Newgenapps*. Obtenido de <https://www.newgenapps.com/blog/augmented-reality-apps-ar-examples-success>
- Newgenapps. (27 de Abril de 2018). *Newgenapps*. Obtenido de <https://www.newgenapps.com/blog/top-5-virtual-reality-examples-real-world>



- Nield, D. (23 de Julio de 2017). *Gizmodo*. Obtenido de <https://gizmodo.com/all-the-sensors-in-your-smartphone-and-how-they-work-1797121002>
- ReSwitched. (31 de Marzo de 2017). *ReSwitched Wiki*. Obtenido de <https://reswitched.tech/hardware/joycon>
- RobotShop inc. (s.f.). *RobotShop*. Obtenido de <https://www.robotshop.com/media/files/pdf/datasheet-leap-motion-controller.pdf>
- Sunil. (15 de Enero de 2011). *Mobile Gyros*. Obtenido de <http://mobilegyros.blogspot.com/>
- u/iamonlyoneman. (s.f.). [https://www.reddit.com/r/mildlyinteresting/comments/4png8z/whats\\_inside\\_a\\_wii\\_sensor\\_bar\\_disassembly\\_photos/](https://www.reddit.com/r/mildlyinteresting/comments/4png8z/whats_inside_a_wii_sensor_bar_disassembly_photos/).
- Veljko B.Petrović, D. I. (2014). Active Learning Attributes of Wii Peripherals in CS Education. En J. G. Laborda, *4th World Conference on Learning Teaching and Educational Leadership* (págs. 447-452). Obtenido de <https://www.sciencedirect.com/science/article/pii/S1877042814035022>
- Virtual Reality Society. (s.f.). *Virtual Reality Society*. Obtenido de <https://www.vrs.org.uk/virtual-reality/who-coined-the-term.html>

## 8. Anexos

### Anexo 1: Descripción máquina virtual con distribución Linux

Esta máquina se ha construido a través del programa Oracle VM VirtualBox63, instalándole una distribución Ubuntu 18.04 y los paquetes de características adicionales del programa llamados “Guest additions”, para esto se puede usar el comando siguiente en la consola “bash” del sistema:

```
sudo apt-get install virtualbox-guest-additions-iso
```

Una vez hecho esto y para la correcta reproducción de los casos descritos en este trabajo hay que usar adecuadamente el “Bluetooth” a través de este programa. Para conseguir esto, hay que vincular el dispositivo USB en la configuración de la máquina, visto en la figura 8-1, y desconectarla del equipo principal para dejarle acceso a este a la máquina virtual.

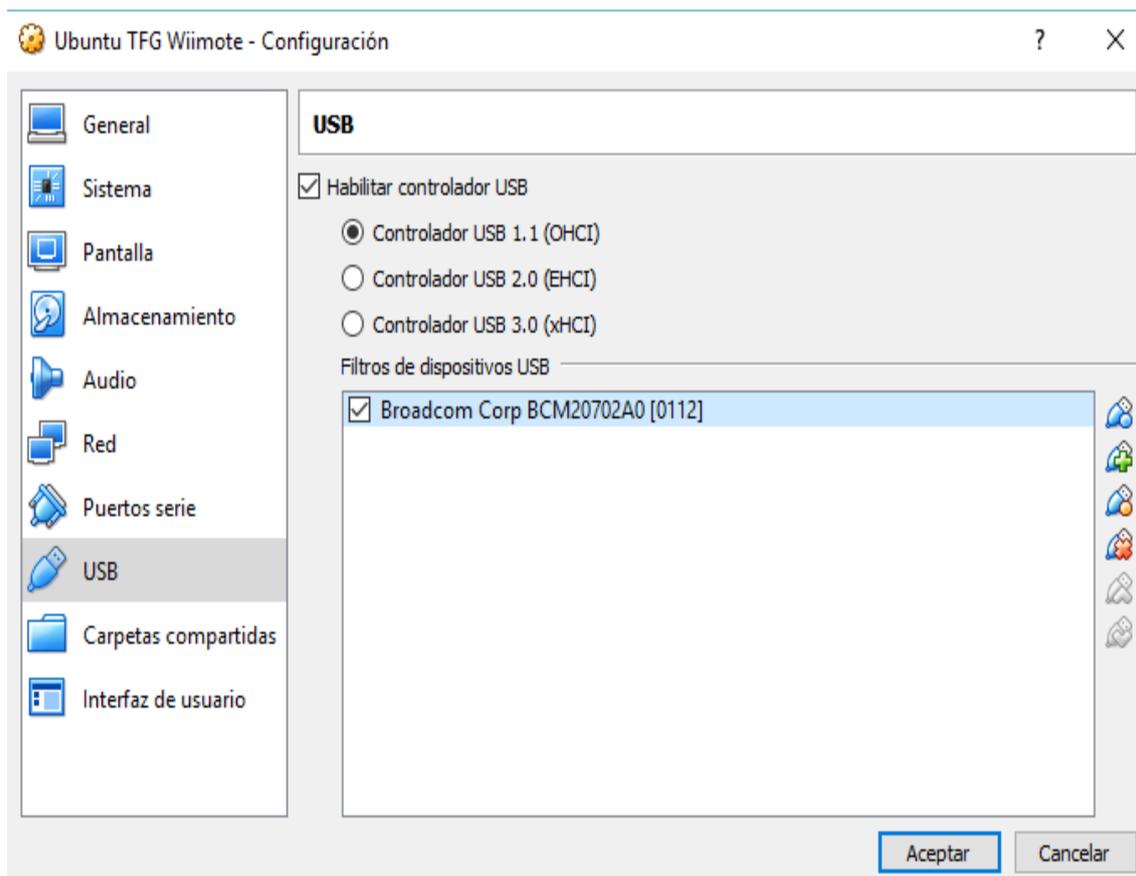


Figura 8-1. Pantalla donde aparecen los USBs reconocidos por la máquina virtual, con posibilidad de añadir más.

<sup>63</sup> Página principal en <https://www.virtualbox.org/>.



## Anexo 2: Capturas conexión “Wii remote” al PC en distintos sistemas operativos

En este anexo se muestran las capturas realizadas para los pasos que se han descrito en el capítulo 3.3.

### Anexo 2.1: Conexión “Wii remote” en Windows

#### Paso 1:

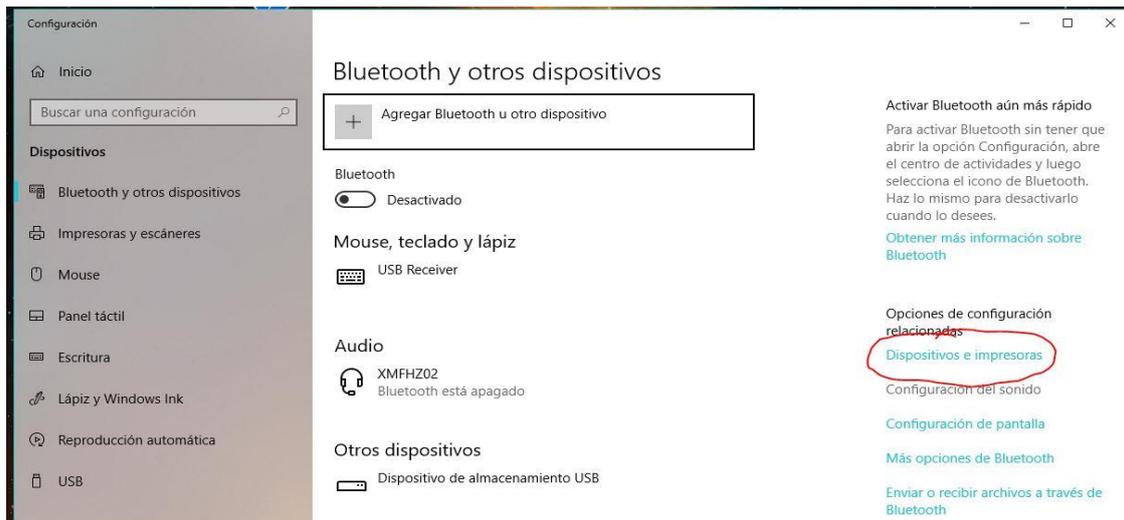


Figura 8-1. Configuración de Bluetooth y otros dispositivos, con el botón a pulsar marcado en rojo.

#### Paso 2:



Figura 8-2. Dispositivos e impresoras, con paso a seguir en rojo.

**Paso 3:**

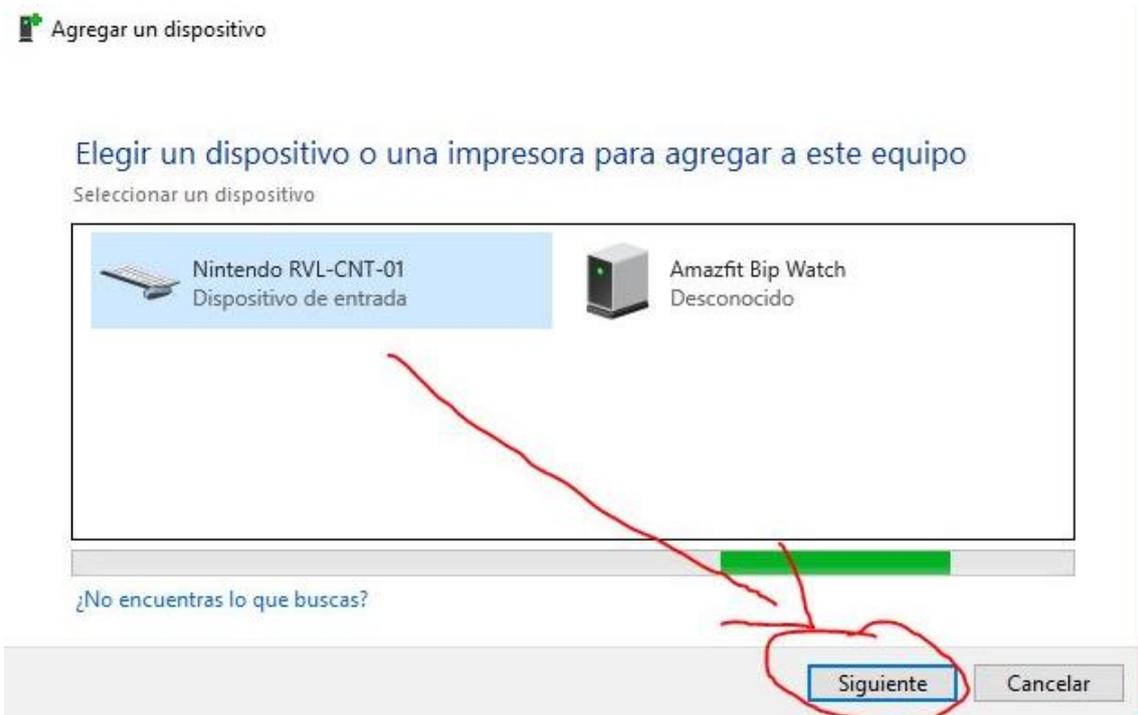


Figura 8-3. Pantalla de elección de dispositivo a conectar, con paso a seguir en rojo.

**Paso 4:**



Figura 8-4. Pantalla donde se tiene que dejar el PIN vacío la correcta conexión con el "Wii remote".

## Anexo 2.2: Conexión de “Wii remote” en Linux

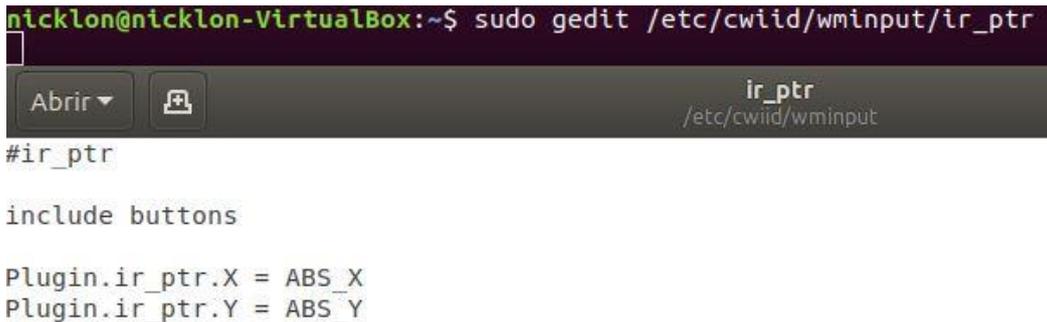
### Paso 1:

```
nicklon@nicklon-VirtualBox:~$ lswm
Put Wiimotes in discoverable mode now (press 1+2)...
E0:0C:7F:8E:C2:2A
```

Figura 8-5. Obtención dirección MAC del Wii remote.

### Paso 2:

```
nicklon@nicklon-VirtualBox:~$ sudo gedit /etc/cwiiid/wminput/ir_ptr
```



```
#ir_ptr

include buttons

Plugin.ir_ptr.X = ABS_X
Plugin.ir_ptr.Y = ABS_Y
```

Figura 8-6. Configuración del fichero para usar Wii remote como ratón.

### Paso 3:

```
nicklon@nicklon-VirtualBox:~$ sudo wminput -c ir_ptr E0:0C:7F:8E:C2:2A
Put Wiimote in discoverable mode now (press 1+2)...
Ready.
```

Figura 8-7. Conexión Wii remote en modo ratón.

### Paso 4:



Figura 8-8. Pantalla conexión de “Wii remote” en wmgui.

## Anexo 3: Explicaciones código implementaciones

### Anexo 3.1: Rotación mediante posición relativa

La función encargada de esto se llama en cada “frame” del juego en la función Update().

```
void UpdateRotationWiimote()
{
    model.rot.rotation= Quaternion.FromToRotation
        (
            ir_origin.transform.position,
            ir_pointer.position
        ) * Quaternion.Euler(new Vector3(0, 180, 0));
}
```

*Fragmento Código 8-1. Función UpdateRotationWiimote.<sup>64</sup>*

En el fragmento de código 8-1 vemos la función encargada de la rotación según dos puntos, “ir\_origin” e “ir\_pointer”. El primer punto es una modificación del modelo del “Wii remote” que viene con el paquete, Unity-Wiimote, por defecto.

Este punto se usa para disparar los proyectiles y para tener una referencia de la posición en la que está la cámara infrarroja, que por defecto no tiene nada en esta posición.

El último punto “ir\_pointer” es el obtenido del procesamiento de la cámara IR del “Wii remote” como punto al que se está apuntando sobre la pantalla. Junto al punto del párrafo anterior, “ir\_origin”, se genera un Quaternion, que son operadores matemáticos que se usan para manipular vectores sin los cuales se podrían producir bloqueos en los ejes<sup>65</sup>, que deja al modelo en la escena mirando hacia el punto objetivo, “ir\_origin”, pero con los botones dando la espalda al usuario, por lo que se le aplica un Quaternion de 180 grados en el eje y para girarlo y que los botones miren hacia el usuario, que es la posición normal en la que se sujeta este dispositivo.

### Anexo 3.2: Creación objetos simples

Para la creación de los múltiples objetos que otorgan puntos en los mini juegos expuestos en el trabajo se utilizan variaciones de esta función, vista en el fragmento de código 8-2. Las cosas más importantes que hace es asignarle una transformación de

---

<sup>64</sup> Código extraído de la clase WiimoteShooter.cs

<sup>65</sup> Más información en <https://www.allaboutcircuits.com/technical-articles/dont-get-lost-in-deep-space-understanding-quaternions/> .



acuerdo con la posición en la que se quiera situar inicialmente, una fuerza hacia donde se quiera llevar al objeto y un tamaño según el tipo de objetivo.

También se le asigna un script, responsable de los mensajes cuando es destruido, que sirve para contar los puntos de cada mini juego.

```
GameObject createNormalObjective ()
{
    GameObject objective = GameObject.CreatePrimitive (PrimitiveType.Cube);
    var objScale = new Vector3 (0.4f, 1f, 0.5f);

    objective.transform.localScale = objScale;
    objective.transform.SetParent(gameObject.transform);
    objective.transform.localPosition = new Vector3 (rightBorder, height, 0);

    var objectiveScript = objective.AddComponent<ObjectiveScript>();
    objectiveScript.setPoints(20);
    objective.AddComponent<Rigidbody> ();

    Rigidbody objectiveRigidbody = objective.GetComponent<Rigidbody> ();
    objectiveRigidbody.useGravity = false;
    objectiveRigidbody.AddRelativeForce (
        new Vector3 (-objectiveSpeed, 0f,
                    );

    return objective;
}
```

*Fragmento Código 8-2. Función de creación de un objeto normal.*<sup>66</sup>

### **Anexo 3.3: Adquisición desviación giroscopio y reinicio de este sensor**

Esta función, vista en el fragmento de código 8-4, lo que hace es coger los datos del giroscopio y los comprueba con un vector auxiliar en el que se guardara la desviación. Si no coinciden estos dos vectores significa que el “Wii remote” se está moviendo y no hay que iniciar la cuenta atrás para iniciar la escena. Una vez pasan 10 segundos sin que estos 2 vectores sean distintos entre ellos se asume que es porque el “Wii remote” está quieto y se guarda esa desviación para el resto de la partida.

---

<sup>66</sup> Código extraído de clase createObjectives.cs

En caso de reinicio del sensor esta desviación no se pierde, simplemente se reinicia la rotación del “Wii remote” a la inicial indicada en las instrucciones, sobre una superficie plana con los botones de espaldas al usuario.

```

var precision = 0.2f;

wmpDeviation = new Vector3(-wiimote.MotionPlus.PitchSpeed,
                           wiimote.MotionPlus.YawSpeed,
                           wiimote.MotionPlus.RollSpeed) / 95f;

if ((wmpDeviation.x - plusSetupVector.x) < precision &&
    (wmpDeviation.y - plusSetupVector.y) < precision &&
    (wmpDeviation.z - plusSetupVector.z) < precision)
{
    if (plusSetupStill > 1000)
    {
        InitializeScene();
    }
    else
    {
        plusSetupStill++;
    }
}
else
{
    wmpDeviation = plusSetupVector;
    plusSetupStill = 0;
}

```

*Fragmento Código 8-3. Función para obtener desviación giroscopio e iniciar la escena.<sup>67</sup>*

### Anexo 3.4: Filtrar resultados giroscopio sabiendo desviación “Wii remote”

Esta función, vista en el Fragmento Código 8-3, lo único que hace es después de recibir en offset los datos actuales del giroscopio comprobar si son iguales que la desviación asignada al inicio de la escena, visto en el [anexo 2.3](#), si es igual a esta desviación no rota el modelo.

```

Vector3 offset = new Vector3(-wiimote.MotionPlus.PitchSpeed,
                             wiimote.MotionPlus.YawSpeed,
                             wiimote.MotionPlus.RollSpeed) / 95f;

if (offset - wmpDeviation != Vector3.zero)
{
    model.rot.Rotate(offset - wmpDeviation, Space.Self);
}

```

*Fragmento Código 8-4. Filtrado de resultados giroscopio.<sup>68</sup>*

<sup>67</sup> Código extraído de clase WiimoteMotionPlus.cs

<sup>68</sup> Código extraído de clase WiimoteMotionPlus.cs



### Anexo 3.5: Creación y limpieza “Wii remotes” en multijugador

Empezaremos analizando la función de “SearchAndInitialize”, código de la cual se puede ver en fragmento código 8-5, usada desde la función “Update” principal si no hay ningún “Wii remote”. En esta función vemos como inicialmente se actualiza el “WiimoteManager”, encargado de administrar los “Wii remotes” en una capa más interna.

Una vez actualizado este objeto, se asigna a la variable global el número de “Wii remotes” encontrados por el “WiimoteManager”, también se guardan en una colección interna todos los “Wii remotes” de este objeto y se configuran adecuadamente su forma de comunicarse con el PC y la forma en la que detectan los LEDs infrarrojos, que se iluminan con el número de jugador que tendrán durante la partida.

```
void SearchAndInitialize()
{
    WiimoteManager.FindWiimotes();

    if (!WiimoteManager.HasWiimote())
    {
        return;
    }
    else
    {
        playersActive = WiimoteManager.Wiimotes.Count;

        for (var i = 0; i < WiimoteManager.Wiimotes.Count; i++)
        {
            wiimotes.Add(WiimoteManager.Wiimotes[i]);
            wiimotes[i].SendDataReportMode(InputDataType.REPORT_BUTTONS_ACCEL);
            wiimotes[i].SetupIRCamera(IRDataType.EXTENDED);
            wiimotes[i].SendPlayerLED(i >= 0, i >= 1, i >= 2, i >= 3);

            ir_pointer_list[i].GetComponent<Image>().color = playerColors[i];

            object[] changeColorUI = { i, playerColors[i] };
            SendMessage("SetColorPanel", changeColorUI);
        }
        for(int i = playerColors.Count-1; i >= playersActive; i-- )
        {
            SendMessage("HidePlayerPanel", i);
        }
    }
}
```

*Fragmento Código 8-5. Función para inicializar colección de “Wii remotes”<sup>69</sup>.*

---

<sup>69</sup> Extraído de la clase WiimoteShooterMultiplayer.cs

También se envían mensajes al controlador de la interfaz gráfica para cambiar los colores de los paneles de los jugadores, para no tener variables repetidas con el puntero del color que se asigna en esta clase también, y se ocultan los jugadores que no están presentes.

La función que nos queda por discutir en este anexo es la limpieza en la función “OnDestroy” de la clase principal, que se llama cuando se cambia de escena o cuando se sale de la aplicación por una llamada en código a esta función. Esta función recorre la colección de “Wii remotes”, rellena anteriormente en la función “SearchAndInitialize” con los “Wii remotes” que había encontrado el “WiimoteManager”, y los limpia llamando al “WiimoteManager” pasándole de argumento el objeto de la lista. Posteriormente a este bucle se limpia la colección de la clase principal que tiene los “Wii remotes”.

```
void OnDestroy()
{
    foreach (var wiimote in wiimotes)
    {
        if (wiimote != null)
        {
            WiimoteManager.Cleanup(wiimote);
        }
    }

    wiimotes.Clear();
}
```

*Fragmento Código 8-6. Función de limpieza de “Wii remote”.<sup>70</sup>*

#### **Anexo 4: Descripción “assets” externos usados en Unity3D**

Los “assets” utilizados para hacer el terreno del juego, los árboles y los arbustos son del paquete gratuito de la tienda de Unity3D Nature Starter Kit<sup>71</sup>, junto al skybox<sup>72</sup> usado en el menú principal encontrado en internet de forma gratuita también. La textura de la roca de las montañas fue realizada por hhh315 para DevianArt<sup>73</sup>.

---

<sup>70</sup> Extraído de la clase WiimoteShooterMultiplayer.cs

<sup>71</sup> Enlace a la tienda <https://assetstore.unity.com/packages/3d/environments/nature-starter-kit-2-52977>.

<sup>72</sup> Sitio Web en <http://www.custommapmakers.org/skyboxes.php>.

<sup>73</sup> Enlace en <https://www.deviantart.com/hhh316/art/Seamless-Rock-Face-Texture-271675185>.



La espada laser usada junto al modelo del “Wii remote”, proporcionado con Unity-Wiimote, ha sido realizada por Jeff Johnson<sup>74</sup>. Incluía más efectos de los usados en el mini juego ya que quería centrar la atención en la implementación.

Todos los demás “assets” han sido realizados con formas primitivas de forma propia.

## Anexo 5: Formulario online rellenado por sujetos reales

En este anexo se muestra el contenido del formulario rellenado para los datos recogidos en el apartado 4-4, para que en caso de pérdida del enlace al formulario se siga pudiendo ver.

# Formulario Wiimote Unity

Este formulario se ha realizado para saber el grado de satisfacción de los usuarios al usar un Wii Remote en un PC en los casos estudiados en este trabajo.

\*Obligatorio

Has tenido alguna lesión relacionada con el uso de teclado y ratón? \*

Sí

No

Cuánto invertirías en un sistema de realidad aumentada o virtual? (En euros) \*

1 2 3 4 5 6 7 8 9 10

0           1000

---

<sup>74</sup> Sitio Web en <http://www.digitalruby.com/unity-plugins/volumetric-lightsaber-unity/> .

Indica la experiencia que tienes en los siguientes dispositivos:

Wiimote \*

|         | 1                     | 2                     | 3                     | 4                     | 5                     |          |
|---------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|----------|
| Ninguna | <input type="radio"/> | Bastante |



Steam Controller \*

|         | 1                     | 2                     | 3                     | 4                     | 5                     |          |
|---------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|----------|
| Ninguna | <input type="radio"/> | Bastante |



### JoyCon Switch \*

|         | 1                     | 2                     | 3                     | 4                     | 5                     |          |
|---------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|----------|
| Ninguna | <input type="radio"/> | Bastante |



### HTC Vive \*

|         | 1                     | 2                     | 3                     | 4                     | 5                     |          |
|---------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|----------|
| Ninguna | <input type="radio"/> | Bastante |



Oculus Rift \*

|         |                       |                       |                       |                       |                       |          |
|---------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|----------|
|         | 1                     | 2                     | 3                     | 4                     | 5                     |          |
| Ninguna | <input type="radio"/> | Bastante |



Leap Motion \*

|         |                       |                       |                       |                       |                       |          |
|---------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|----------|
|         | 1                     | 2                     | 3                     | 4                     | 5                     |          |
| Ninguna | <input type="radio"/> | Bastante |



## Wiimote en el PC

A continuación deberás responder preguntas basandote en lo visto sobre este trabajo.

Cómo calificarías la experiencia de controlar el PC sin ratón? \*

- Prefiero ratón
- Prefiero Wii Remote
- Indiferente

Cómo calificarías la precisión del Wiimote frente a un ratón convencional (Wiimote Shooter) ? \*

|               |                       |                       |                       |                       |                       |                |
|---------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|----------------|
|               | 1                     | 2                     | 3                     | 4                     | 5                     |                |
| Bastante Mala | <input type="radio"/> | Bastante Buena |

Cómo calificarías el uso del giroscopio (WiimoteJedi) frente al uso de otros dispositivos de interacción avanzados? \*

|               |                       |                       |                       |                       |                       |                |
|---------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|----------------|
|               | 1                     | 2                     | 3                     | 4                     | 5                     |                |
| Bastante Mala | <input type="radio"/> | Bastante Buena |

## Conclusiones

Qué dispositivos prefieres para este tipo de actividades? \*

- Teclado y ratón
- Wiimote
- Otro: \_\_\_\_\_

Cómo valoras la experiencia del uso del Wii remote fuera de la Wii? \*

|      |                       |                       |                       |                       |                       |       |
|------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-------|
|      | 1                     | 2                     | 3                     | 4                     | 5                     |       |
| Mala | <input type="radio"/> | Buena |

Si existieran alternativas al teclado y el ratón en el PC las usarías? \*

- Sí
- No
- Tal vez

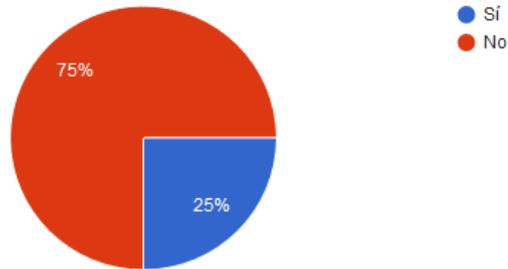
Gastarías 40 euros (aproximadamente) por poder usar este sistema para el ordenador? \*

- Si
- No

## Anexo 6: Resultados del formulario

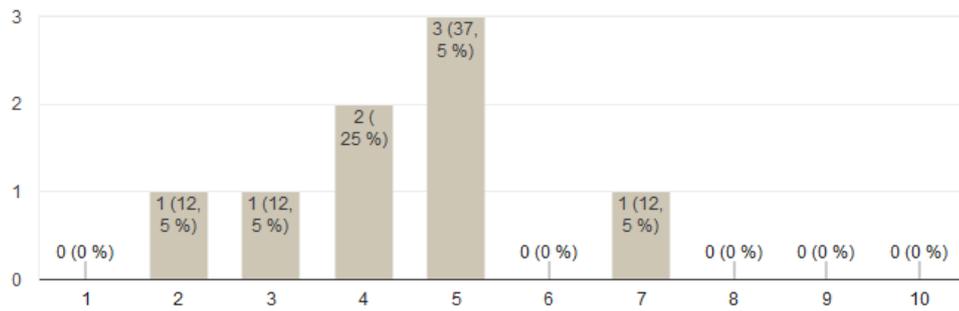
Has tenido alguna lesión relacionada con el uso de teclado y ratón?

8 respuestas



Cuánto invertirías en un sistema de realidad aumentada o virtual? (En euros)

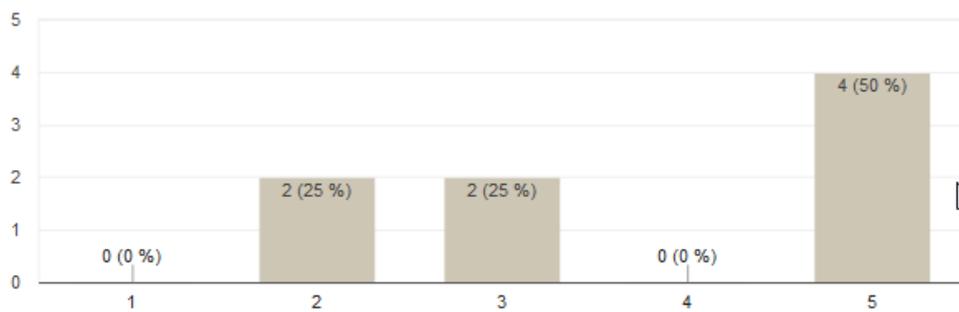
8 respuestas



Indica la experiencia que tienes en los siguientes dispositivos:

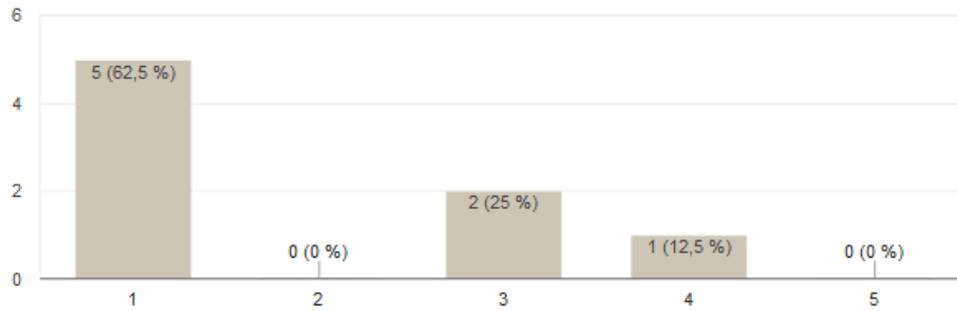
Wiimote

8 respuestas



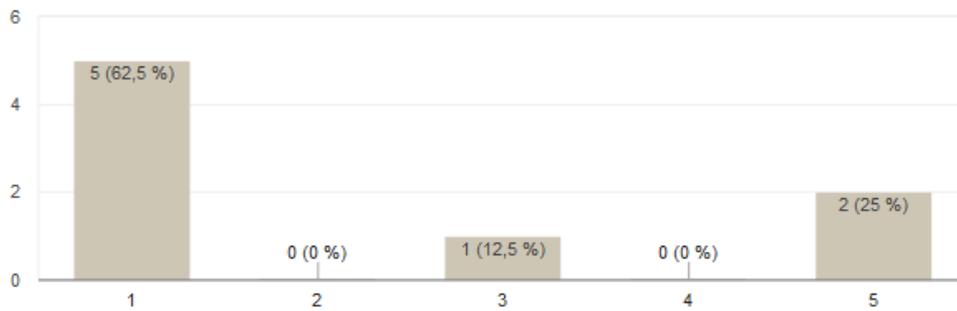
### Steam Controller

8 respuestas



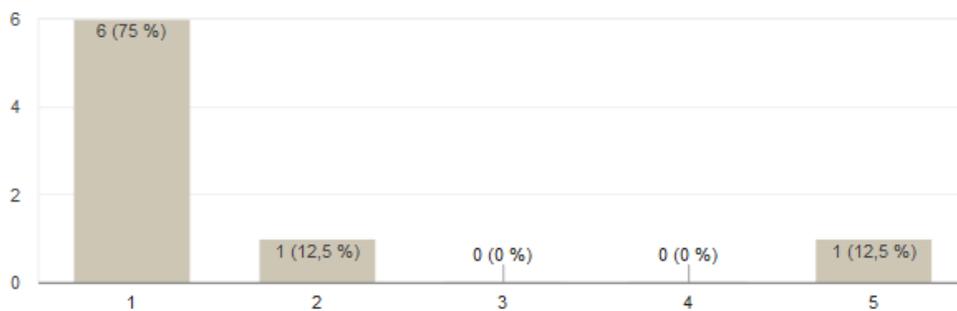
### JoyCon Switch

8 respuestas



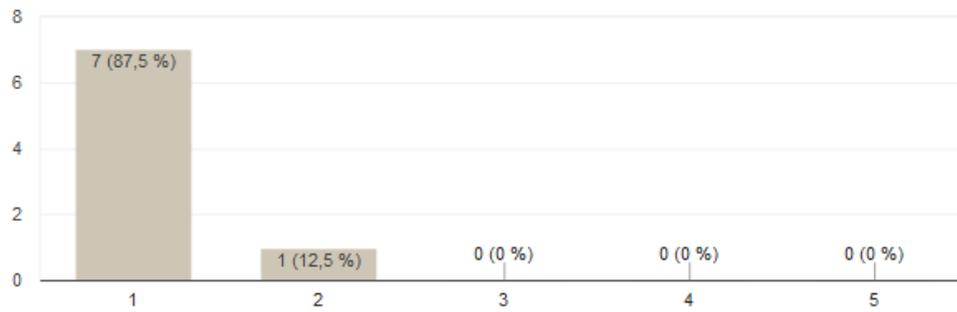
### HTC Vive

8 respuestas



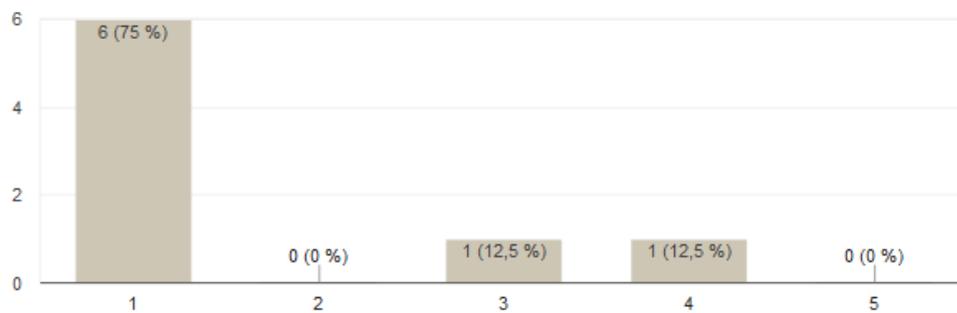
### Oculus Rift

8 respuestas



### Leap Motion

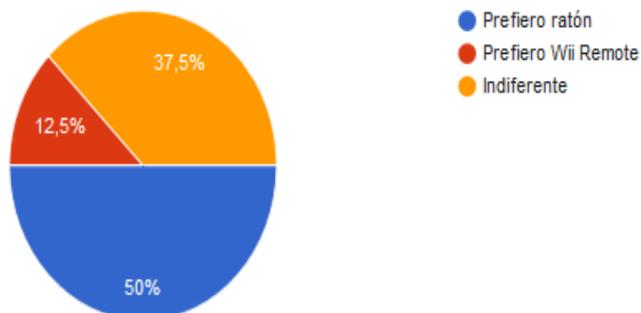
8 respuestas



### Wiiote en el PC

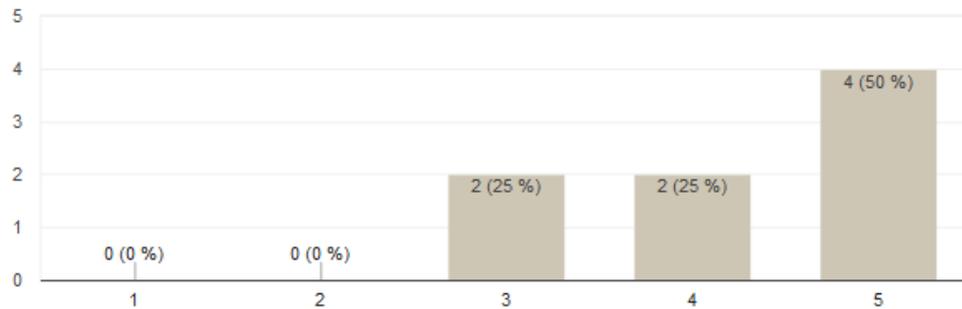
### Cómo calificarías la experiencia de controlar el PC sin ratón?

8 respuestas



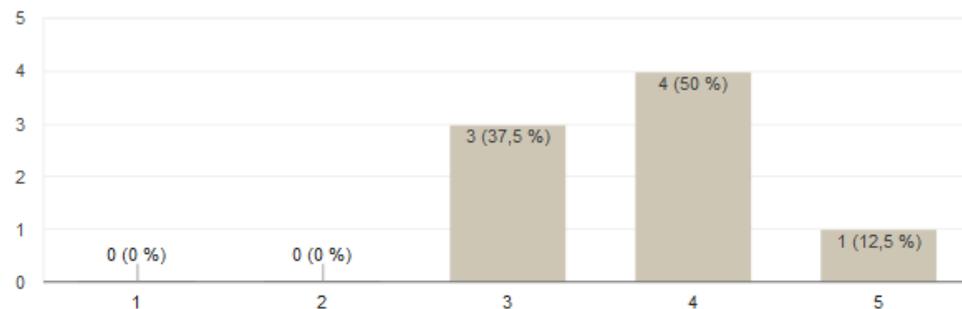
### Cómo calificarías la precisión del Wiimote frente a un ratón convencional (Wiimote Shooter) ?

8 respuestas



### Cómo calificarías el uso del giroscopio (WiimoteJedi) frente al uso de otros dispositivos de interacción avanzados?

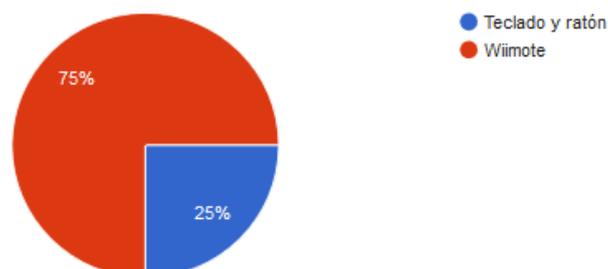
8 respuestas



## Conclusiones

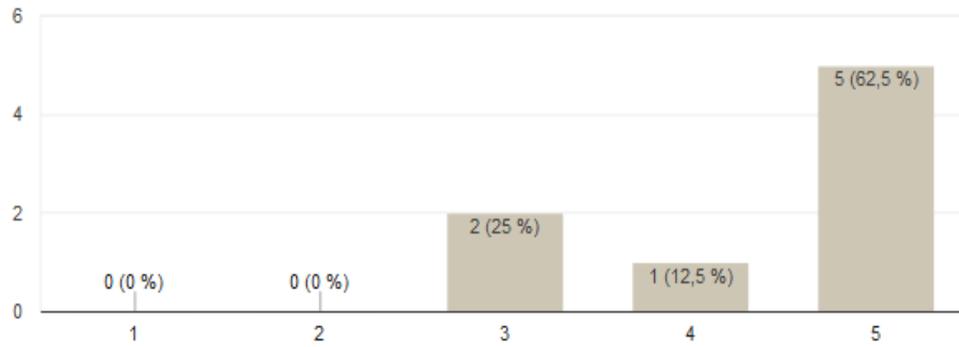
### Qué dispositivos prefieres para este tipo de actividades?

8 respuestas



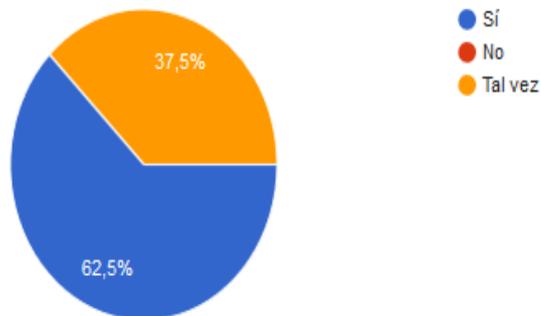
### Cómo valoras la experiencia del uso del Wii remote fuera de la Wii?

8 respuestas



### Si existieran alternativas al teclado y el ratón en el PC las usarías?

8 respuestas



### Gastarías 40 euros (aproximadamente) por poder usar este sistema para el ordenador?

8 respuestas

