

Entorno de realidad virtual controlado en tiempo real por señales motoras para la rehabilitación del miembro superior en pacientes con daño cerebral

Hugo Kumpel del Castillo y Alejandro Scatton Lóndero

GRADO EN INGENIERÍA INFORMÁTICA

FACULTAD DE INFORMÁTICA

DEPARTAMENTO DE ARQUITECTURA DE COMPUTADORES Y AUTOMÁTICA

UNIVERSIDAD COMPLUTENSE DE MADRID



TRABAJO DE FIN DE GRADO EN INGENIERÍA INFORMÁTICA

Madrid, 17 de junio de 2016

Director: José Ignacio Hidalgo Pérez

Codirector: José Ignacio Serrano Moreno



U N I V E R S I D A D
COMPLUTENSE
M A D R I D

Autorización de difusión y utilización

Los abajo firmantes, alumno/s y tutor/es del Trabajo Fin de Grado (TFG) en el Grado en Ingeniería Informática de la Facultad de Informática, autorizan a la Universidad Complutense de Madrid (UCM) a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a su autor el Trabajo Fin de Grado (TF) cuyos datos se detallan a continuación. Así mismo autorizan a la Universidad Complutense de Madrid a que sea depositado en acceso abierto en el repositorio institucional con el objeto de incrementar la difusión, uso e impacto del TFG en Internet y garantizar su preservación y acceso a largo plazo.

Periodo de embargo (opcional):

- 6 meses
- 12 meses

TÍTULO del TFG: Entorno de realidad virtual controlado en tiempo real por señales motoras para la rehabilitación de miembro superior en pacientes con daño cerebral

Curso académico: 2015 / 2016

Nombre del Alumno/s:
Hugo Kumpel del Castillo
Alejandro Scatton Lóndero

Tutor/es del TFG:
José Ignacio Hidalgo Pérez
José Ignacio Serrano Moreno

Firma del alumno/s

Firma del tutor/es

Dedicatoria

A nuestras familias por darnos todo y apoyarnos para conseguir lo que nos proponíamos

A nuestras novias, Valentina y Ana, por su apoyo y siempre alentarnos a ser los mejores

A todos nuestros amigos que nos han acompañado y ayudado todo este tiempo

A todos ellos les dedicamos este trabajo porque sin ellos no habría sido posible

Agradecimientos

Queremos agradecer al CSIC y en especial a José Ignacio Serrano Moreno y a Lola del Castillo por la idea y por permitirnos elaborar este proyecto. De igual forma por su inestimable ayuda y colaboración en cada aspecto de este trabajo, por proveernos el material necesario y la posibilidad de trabajar en las instalaciones del CSIC de manera regular.

Agradecer también a José Ignacio Hidalgo por su ayuda e interés en siempre brindarnos una mano a lo largo de este proyecto.

Índice

Agradecimientos	VII
Índice de figuras.....	XI
Índice de abreviaturas	XIII
Resumen.....	XV
Abstract	XVII
Capítulo 1. Introducción	1
1.1 Daño Cerebral Adquirido	1
1.1.1 Causas del DCA.....	1
1.1.1.2 Ictus.....	2
1.1.1.3 Traumatismo Cráneo Encefálico.....	3
1.1.2 Déficits del DCA	3
1.1.2.1 Alteraciones motoras	3
1.1.2.2 Alteraciones cognitivas	4
1.1.2.3 Alteraciones conductuales	5
1.2 Rehabilitación del miembro superior	6
1.2.1 Terapia tradicional.....	6
1.2.2 Ejercicios e intervenciones basadas en movimiento funcional	7
1.2.3 Tecnologías asistidas	8
1.3 Realidad Virtual en rehabilitación.....	9
1.4 Captura de movimientos.....	13
1.5 Antecedentes de rehabilitación mediante RV	14
1.6 Objetivos del Trabajo de Fin de Grado	17
1.6.1 Objetivos específicos	18
1.6.2 Novedad del TFG.....	19
1.7 Introduction	20
Capítulo 2. Materiales y Métodos	21
2.1 Tecnologías empleadas	21
2.1.1 Entorno de desarrollo: Unity 3D	21
2.1.2 MakeHuman	26

2.1.2	Captura de Movimiento: Myo Armband	27
2.1.3	Dispositivo de RV: Oculus Rift DK2	29
2.2	Métodos	30
2.2.1	Escenas	31
2.2.1.1	Escena 1: Inicio	32
2.2.1.2	Escena 2: Elección brazo	32
2.2.1.3	Escena 3: Menú principal.....	32
2.2.1.4	Escena 4: Calibrado	32
2.2.1.5	Escena 5: Definir Terapia	33
2.2.1.6	Escena 6: Rehabilitación.....	35
2.2.1.6.1	Reproducción del alcance del brazo en la RV.....	36
2.2.1.6.2	Registro de actividad.....	37
2.2.1.7	Escena 7: Ayuda	39
2.2.1.8	Escena 8: Créditos.....	39
2.2.2	Captura de movimientos	39
Capítulo 3.	Resultados	41
Capítulo 4.	Conclusiones	51
4.1	Trabajo Futuro.....	51
4.2	Conclusions	52
Capítulo 5.	Distribución de trabajo.....	55
Apéndice	57
Bibliografía	67

Índice de figuras

<i>Figura 1.1: Tipos de ictus</i>	2
<i>Figura 1.2: Áreas funcionales corticales</i>	4
<i>Figura 1.3: Samsung Gear RV</i>	9
<i>Figura 1.4: Esquema del DAFO de la RV en rehabilitación en este TFG</i>	11
<i>Figura 1.5: Ejemplo de entorno de RV</i>	12
<i>Figura 1.6: Predicción de futuro del uso de RV</i>	13
<i>Figura 1.7: Rehabilitación con el sistema de captura de movimiento Kinect.</i> ..	14
<i>Figura 1.8: Paciente utilizando BioTrak</i>	15
<i>Figura 1.9: Movimiento de abducción y aducción</i>	16
<i>Figura 1.10: Hemisferios cerebrales y control motor</i>	18
<i>Figura 2.1: Logo Unity</i>	22
<i>Figura 2.2: Uso de Unity en el mundo</i>	23
<i>Figura 2.3: Cuota de mercado de Unity</i>	23
<i>Figura 2.4: Aspecto editor Unity</i>	24
<i>Figura 2.5: Logo MakeHuman</i>	26
<i>Figura 2.6: Aspecto de la aplicación MakeHuman. El humano base se puede ir moldeando para que adquiera distintos aspectos.</i>	26
<i>Figura 2.7: Myo Armband</i>	28
<i>Figura 2.8: Aspecto frontal y trasero Oculus Rift DK2</i>	29
<i>Figura 2.9: Diagrama OMT Singleton</i>	30
<i>Figura 2.10: Diagrama que representa la lógica de la aplicación</i>	31
<i>Figura 2.11: Concepto de Esfuerzo en la aplicación.</i>	33
<i>Figura 2.12: Aspecto fichero de terapia. Se visualiza primero el calibrado y luego dos niveles, el primero con 80% y 2 puntos y el segundo al 50% y 1 punto.</i>	34
<i>Figura 2.13: Lista bidimensional que contiene los niveles definidos.</i>	35
<i>Figura 2.14: Datos recogidos en el fichero</i>	38
<i>Figura 2.15: Se aprecia la problemática: cuando la aceleración es cero como resultado de que se ha parado, se observa que el espacio sigue levemente por</i>	

<i>encima de cero y por lo tanto se sigue desplazando. Aceleración en azul, velocidad en naranja y espacio en gris.</i>	40
<i>Figura 3.1: Escena Inicio</i>	41
<i>Figura 3.2: Escena Elige Brazo</i>	42
<i>Figura 3.3: Escena Menú Principal</i>	42
<i>Figura 3.4: Escena Ayuda</i>	43
<i>Figura 3.5: Escena Créditos</i>	43
<i>Figura 3.6: Escena Calibrado</i>	44
<i>Figura 3.7: Escena Define Terapia</i>	44
<i>Figura 3.8: Escena Rehabilitación</i>	45
<i>Figura 3.9: Escena Rehabilitación II</i>	45
<i>Figura 3.10: Escena Rehabilitación III</i>	46
<i>Figura 3.11: Escena Rehabilitación IV</i>	46
<i>Figura 3.12: Ejemplo Escenario I</i>	47
<i>Figura 3.13: Ejemplo Escenario II</i>	47
<i>Figura 3.14: Pruebas con usuarios no afectados</i>	48
<i>Figura 3.15: Pruebas con usuarios no afectados II</i>	49
<i>Figura 3.16: Adaptabilidad de paciente no afectado. Datos obtenidos del fichero que registra la actividad del paciente en la escena “Rehabilitación”</i>	49
<i>Figura 3.17: Variación de aceleración según el esfuerzo requerido. Datos obtenidos del fichero que registra la actividad del paciente en la escena “Rehabilitación”</i>	50

Índice de abreviaturas

- **TFG:** Trabajo de Fin de Grado
- **RehaV:** Nombre de la aplicación desarrollada
- **CSIC:** Consejo Superior de Investigaciones Científicas
- **RV:** Realidad virtual
- **ACV:** Accidente Cerebrovascular
- **DCA:** Daño cerebral adquirido
- **TCE:** Traumatismo craneoencefálico
- **Myo:** Brazaletes Myo Armband
- **EMG:** Electromiograma
- **GUI:** *Graphic User Interface* (Interfaz gráfica de usuario)

Resumen

Este trabajo tiene como meta el desarrollo de un sistema de neurorehabilitación mediante realidad virtual para pacientes con problemas motores en las extremidades superiores provocados por un daño cerebral.

Para entender y enfrentarse de manera correcta al problema, se han revisado todos los aspectos concernientes al daño cerebral y déficits motores derivados así como las terapias empleadas para su rehabilitación, como las que utilizan tecnologías asistidas como la realidad virtual. Se han expuesto las ventajas que han llevado a utilizar esta terapia en el proyecto y se han mencionado algunos estudios de investigación previos y con resultados favorables. Por último, se ha hecho hincapié en las novedades (*embodiment* y neuronas espejo) que aporta este proyecto con las que se podría acelerar la rehabilitación de los pacientes incidiendo en la plasticidad cerebral.

En lo que se refiere al desarrollo del entorno, se ha utilizado el motor de videojuegos Unity 3D, con el que se ha integrado el dispositivo Myo Armband, para capturar el movimiento, y las gafas de realidad virtual Oculus Rift DK2. Durante la terapia, el paciente tiene que superar una serie de niveles definidos por el terapeuta y, para ello, debe realizar un alcance y tocar un botón un número de veces por cada nivel. En cada nivel, el esfuerzo que tiene que realizar el paciente es distinto, dependiendo de la distancia a la que se encuentre el botón (relativa al calibrado previo del paciente) e igualmente, está definido por el terapeuta. También se lleva a cabo un registro de las aceleraciones y del porcentaje de acierto del paciente en un fichero, para llevar un control de su progreso.

Como conclusión, cabe mencionar que el entorno desarrollado es adaptable a las posibilidades de cada paciente, permitiendo una recuperación óptima. Se trata de un sistema de bajo coste, intuitivo y que lleva a cabo un registro de datos de la actividad del paciente, lo que posibilita, como trabajo futuro, el control de forma remota por parte del terapeuta. Todos estos aspectos dejan abierta la posibilidad de que el paciente lo utilice a domicilio, con las ventajas que conlleva. Del mismo modo, el sistema se podría adaptar fácilmente para la rehabilitación de las extremidades inferiores. Tras su análisis, los investigadores expertos en la materia del Consejo Superior de Investigaciones Científicas afirman que el sistema satisface con creces su propósito. Con todo esto, se concluye que este entorno tiene un excelente presente y un futuro muy prometedor.

Palabras clave: Ictus, Traumatismo Craneoencefálico, Plasticidad, Captura de movimientos, Myo Armband, Oculus Rift DK2, Unity 3D, Realidad Virtual, Neurorehabilitación, Miembro superior.

Abstract

The goal of this project is to develop a neurorehabilitation system with virtual reality for patients with motor problems in the superior limbs caused by brain damage.

Firstly, in order to understand and properly face the challenges of this subject, it has been covered several aspects related to brain damage and the motor deficits it brings and the employed therapies for the rehabilitation, such as those which use assisted technologies as virtual reality. It has also been exposed the advantages which made this type of therapies to be employed in this project and previous investigations with positive results. Finally, it has been stated the most novel aspects of this work (embodiment and mirror neurons) focusing on the brain plasticity with which the rehabilitation could be faster and more effective.

Concerning the development of the system, it has been employed Unity 3D game engine with which the Myo Armband movement tracker device and Oculus Rift DK2 virtual reality glasses have been integrated. During the therapy, the patient has to complete a number of levels defined by the therapist; to do so, the patient has to reach and push a button several times per level. On each level the effort required by the patient is different depending on the distance the button is located (it is relative to the previous calibration of the patient). This distance is decided by the therapist. It has also been decided to register the acceleration raw data from the Myo Armband and the success rate in a file to monitor the progress of the patient.

As a conclusion, it has to be mentioned that this system is flexible to each patient providing an optimum recovery. It is developed as a low cost system, with an intuitive interface and in addition it allows recording data from the rehabilitation session of each patient, this feature makes possible, as a future development, a remote control of the rehabilitation by the therapist. All these aspects leave an open door to the possibility to complete the rehabilitation from home, with all the advantages as cost or time. Also another prospective variety in a near future could be a rehabilitation system applied to the lower limbs. After evaluating the resulting system, the experts in this matter from the Superior Council for Scientific Investigations concurred that it fulfills its purpose. Finally, to sum up, this environment has showed an excellent present and a truly promising future.

Keywords: Stroke, Head Trauma, Neuroplasticity, Movement Tracker, Myo Armband, Oculus Rift DK2, Unity 3D, Virtual Reality, Neurorehabilitation, Upper Limb.

Capítulo 1. Introducción

La informática avanza a pasos agigantados hoy día, tal es así que uno de sus potenciales más inmediatos es su uso en el ámbito de la medicina moderna.

Con este trabajo se pretende contribuir y ayudar en ese campo realizando una interfaz de realidad virtual (RV) que permita rehabilitar el movimiento de alcance a pacientes con dificultades motoras en las extremidades superiores, como consecuencia de un daño cerebral. Con ello se aporta una solución aplicable a un problema real, lo que unido al potencial de las tecnologías empleadas fueron las principales fuentes de motivación para realizar este proyecto.

Este capítulo se divide en varios puntos. Primeramente se explicará lo que es el daño cerebral adquirido, sus causas y déficits más importantes, como son las alteraciones motoras en la extremidad superior, y cómo se intenta dar solución con el sistema desarrollado en este proyecto. Seguidamente se presentarán los distintos tipos de rehabilitación que han resultado ser beneficiosos en mayor o menor medida para este problema, desde terapias tradicionales hasta alguna más novedosa, como la implementada en este trabajo, la que utiliza RV. A continuación, se pasará a explicar las fortalezas y debilidades de este tipo de terapia y por qué se ha elegido. También, se revisarán algunos trabajos de investigación en los que se han obtenido mejoras significativas tras llevar a cabo neurorehabilitación mediante RV, para acabar presentando los objetivos y los aspectos distintivos y más novedosos de este proyecto respecto a otros anteriores, gracias a los que se podría acelerar la recuperación de los pacientes.

1.1 Daño Cerebral Adquirido

El Daño Cerebral Adquirido (DCA) se produce por una lesión cerebral surgida de manera brusca e inesperada. Su origen es, por orden de incidencia, un accidente cerebrovascular (ACV) y un traumatismo craneoencefálico (TCE). Figura como la tercera causa de muerte en nuestra población y la primera entre las mujeres. En el mundo occidental es la primera causa de incapacidad grave y crónica. Se calcula que en España se producen unas 100.000 lesiones cerebrales al año y el 25% de ellas afectan a menores de 30 años [1].

1.1.1 Causas del DCA

Las causas del DCA constituyen un concepto heterogéneo ya que su origen puede ser muy diferente y su recuperación variable dependiendo de factores diversos.

Estas lesiones pueden provocar déficits sensoriales, motrices, cognitivos y emocionales, que repercuten en la calidad de vida y en la capacidad de relación social de la persona.

A continuación, se describen las causas más comunes a las que va dirigido este proyecto: el ictus y el TCE.

1.1.1.2 Ictus

Se denomina ictus al trastorno agudo de la circulación cerebral que altera la función de una determinada área. Son sinónimos de ictus los términos accidente cerebrovascular, infarto cerebral, ataque cerebral o apoplejía. En España constituye la primera causa de invalidez permanente en población adulta. Son múltiples las causas que lo originan, pero con frecuencia es consecuencia de una serie de circunstancias personales que se denominan factores de riesgo.

Se diferencian dos grandes grupos (ver Figura 1.1) según su origen:

- Ictus hemorrágico: producido por la rotura de un vaso sanguíneo cerebral que sangra en el interior del cerebro. En ocasiones el paciente sufre de modo transitorio todas las manifestaciones con las que cursa un ictus establecido sin dejar ninguna secuela. Esta situación se denomina ataque isquémico transitorio y constituye una verdadera alarma ya que un tercio de las personas que lo experimentan sufren un ictus durante el siguiente año.
- Ictus isquémico: se produce por una disminución importante del flujo sanguíneo que recibe una parte del cerebro. El ictus aterotrombótico o trombosis cerebral está formado por un coágulo de sangre (trombo) formado en la pared de una arteria. En el ictus embólico o embolia cerebral el coágulo de sangre se ha formado en un lugar lejos de la obstrucción, usualmente el corazón o las grandes arterias del cuello. Este coágulo se denomina émbolo.

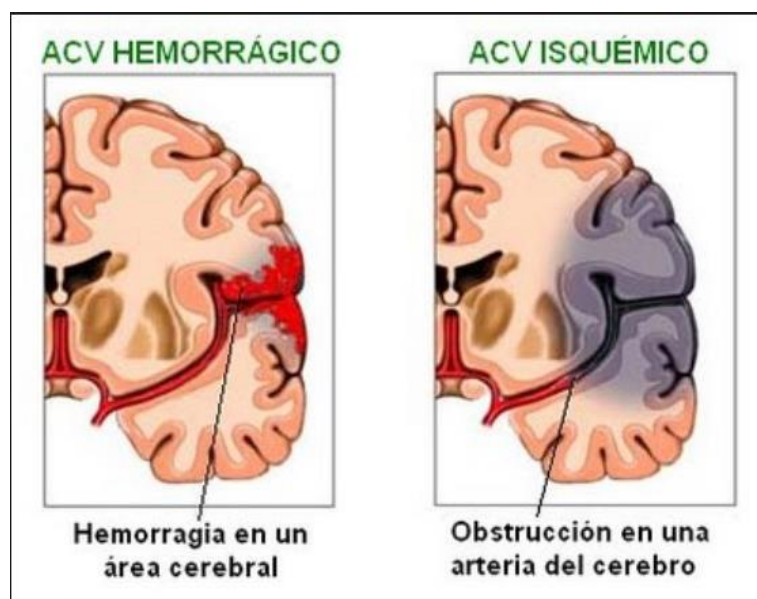


Figura 1.1: Tipos de ictus
Fuente: anamurillojarcia.blogspot.com.es

1.1.1.3 Traumatismo Cráneo Encefálico

El traumatismo craneoencefálico (TCE) se define como cualquier lesión física o deterioro funcional del contenido craneal provocado por un golpe. En los países industrializados constituye una de las principales causas de muerte y discapacidad del adulto joven.

Se engloban bajo el término de traumatismo craneoencefálico (TCE) a las alteraciones que sufre el encéfalo (hemisferios cerebrales, cerebelo y tallo encefálico), sus envolturas meníngeas (duramadre, aracnoides y piamadre), el estuche óseo o los tejidos blandos epicraneales. Esto produce una disminución de la conciencia, síntomas focales neurológicos y amnesia postraumática.

Las lesiones se producen por dos mecanismos:

- Lesiones primarias: lesiones nerviosas y vasculares que aparecen inmediatamente después y hasta las 6-24 horas del impacto. Se pueden producir de modo estático o dinámico. Si es estático, existe un agente externo que se aproxima al cráneo con una velocidad determinada hasta colisionar con él, causando lesiones focales. De este modo se producen hemorragias intracraneales (hematoma epidural, hematoma subdural, hemorragia intraparenquimatosa, contusión hemorrágica). El mecanismo dinámico consiste en fuerzas de aceleración-deceleración y es responsable de lesiones por daño axonal difuso, esto es, el daño ocurre sobre un área amplia más que en un punto focal del cerebro.
- Lesiones secundarias: dependiente o no del impacto primario se ponen en marcha una serie de alteraciones sistémicas e intracraneales que agravan o producen nuevas lesiones cerebrales.

1.1.2 Déficit del DCA

Las consecuencias del Daño Cerebral Adquirido son muy diversas y afectan de manera heterogénea a los diferentes pacientes. Depende principalmente de la extensión del daño y su localización, dada la especialización cerebral (ver Figura 1.2) y la amplia interconexión de las diferentes áreas cortico-subcorticales e inter-hemisféricas.

A continuación se describen los déficits más comunes.

1.1.2.1 Alteraciones motoras

Tras una lesión cerebral, una de las secuelas más importantes es la alteración en el control motor, que se define como la capacidad del sujeto de regular o dirigir los mecanismos esenciales del movimiento. Esto implica una cooperación entre la planificación del movimiento y la retroalimentación sensorial y es fruto de la interacción entre el individuo, la actividad y el ambiente.

La alteración en el control motor, que podría afectar al movimiento de la cara, brazo y pierna de un lado del cuerpo, afecta a un 80% de los supervivientes. Las alteraciones en la extremidad superior (brazo, mano y/o dedos) suelen ser persistentes e incapacitantes. Sólo del 20% al 56% de los supervivientes consiguen recuperar la funcionalidad de la extremidad superior pasados tres meses y de los que continúan con problemas, el 50% podría seguir igual hasta cuatro años después del accidente. Las funciones motoras de la extremidad superior están fuertemente ligadas a las actividades de la vida cotidiana, como puede ser calzarse, vestirse o asearse. Por lo tanto los afectados se ven abocados a una reducción en la calidad de vida, razón por la cual la rehabilitación de la extremidad superior en pacientes con DCA resulta determinante [2].

El daño sufrido puede afectar a diferentes zonas del cerebro como el córtex motor y/o el somatosensorial. Los déficits iniciales sensorimotores pueden prolongarse en el tiempo provocando que el paciente tienda a compensar los movimientos con la extremidad sana con el consiguiente desuso de la extremidad afectada por el daño cerebral. Tanto la inutilización de la extremidad de la que se carece de control, como el uso excesivo para compensar de la extremidad intacta, pueden dar lugar a que se elimine la representación en el córtex motor y somatosensorial de la extremidad afectada [3].

Con el presente Trabajo de Fin de Grado (TFG) se busca incidir en este tipo de carencia para ayudar a los pacientes a neurorrehabilitarse más rápido mediante técnicas de RV “inmersiva” con las que se lleva a cabo la reorganización cortical y, consecuentemente, la actividad funcional, como veremos posteriormente.

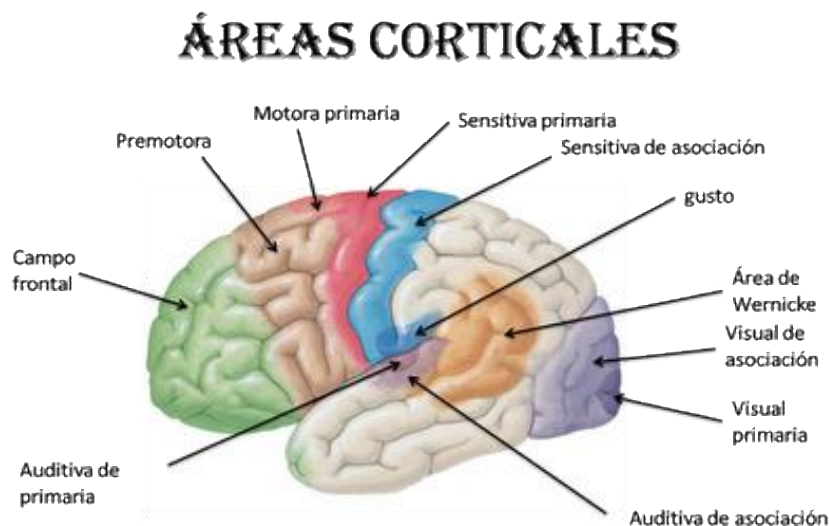


Figura 1.2: Áreas funcionales corticales

Fuente: neurofisiologia.jimdo.com/funciones-del-cerebro/areas-corticales/

1.1.2.2 Alteraciones cognitivas

Este tipo de alteraciones pueden aparecer tras un daño cerebral y comprende:

- Desorientación y confusión.
- Atención y velocidad de procesamiento: alteraciones en la capacidad para atender a la información relevante y evitar las distracciones, mantener la concentración en una tarea durante periodos prolongados de tiempo, así como una adecuada velocidad de respuesta, etc.
- Memoria: dificultades en la adquisición de nuevos aprendizajes (amnesia anterógrada), la recuperación de información antigua (amnesia retrógrada), y la memoria prospectiva u orientada al futuro, fabulaciones o intrusiones en el recuerdo, etc.
- Percepción: alteraciones en el reconocimiento de objetos, en las distintas modalidades sensoriales (visual, auditiva, etc.), conocidas como agnosias.
- Praxias: alteraciones en la ejecución de movimientos voluntarios previamente aprendidos, problemas en la secuenciación motora, etc.
- Funciones ejecutivas: alteraciones en los procesos necesarios para la conducta inteligente y dirigida a metas (capacidad de anticipar y planificar, resolver problemas, inhibir las respuestas automáticas, autorregular el comportamiento, etc.).
- Conciencia de las limitaciones: alteraciones en la percepción de las propias dificultades y de sus repercusiones en la vida diaria. Ajuste de las expectativas de futuro tras el daño cerebral.

Del mismo modo que sucede con las alteraciones motoras, este proyecto también pretende contribuir a la mejora de este tipo de alteraciones, ya que en muchos aspectos están relacionadas con las anteriores. Tras producirse daño cerebral, el paciente podría tener problemas cognitivos en mayor o menor medida. Estas alteraciones podrían acrecentarse al ver que la extremidad no responde como se quiere, por lo que desaprendería todos los patrones de movimiento que realmente aún no ha perdido, adaptándose a las nuevas y mermadas capacidades. Por ello, se va a intentar mediante la RV que el paciente vea que está haciendo correctamente el movimiento y no sufra tanto desorden. Es importante empezar el tratamiento en la fase aguda en la que aún el paciente no haya cogido malos hábitos para que se recupere antes. Si esto no es posible, este sistema de rehabilitación también puede ayudar a reaprender mediante las neuronas espejo (aprendizaje por imitación), ya que al recibir una realimentación visual posturalmente corregida a través de la RV, se activarían zonas del cerebro que han caído en desuso y que viendo el movimiento real de su propio brazo no activa. Todo lo anterior está relacionado con el concepto de *embodiment* (control o conciencia del cuerpo) que más adelante se explicará en detalle.

1.1.2.3 Alteraciones conductuales

Tras una lesión cerebral frecuentemente se producen alteraciones emocionales y conductuales que, en ocasiones, pueden llegar a provocar cambios de personalidad. Estos trastornos pueden ser orgánicos, consecuencia directa de la disfunción cerebral, o bien, una consecuencia del suceso vivido.

Entre las alteraciones más frecuentes pueden observarse irritabilidad, impulsividad, agresividad, egocentrismo, desinhibición, euforia, conducta inapropiada, rigidez, oposicionismo, apatía, labilidad emocional, depresión o infantilismo.

Con la rehabilitación que se propone en este trabajo, también se podría favorecer de algún modo la conducta del paciente, ya que al ver que está realizando el movimiento correctamente se reduciría su frustración y podría motivarle.

1.2 Rehabilitación del miembro superior

Los profesionales responsables de la rehabilitación de la extremidad superior suelen ser comúnmente fisioterapeutas y terapeutas ocupacionales. Sin embargo, puede haber otros profesionales como enfermeras y médicos, además de cuidadores y miembros de la familia, que juegan también un papel importante en la recuperación. La rehabilitación debería empezar tras el accidente o cuando el paciente se encuentre en la fase aguda. Los profesionales encargados de la rehabilitación tendrán en cuenta los objetivos personalizados de cada paciente tras pasar una evaluación que mida la discapacidad del miembro superior. Sin embargo, el paciente también puede rehabilitarse realizando ejercicios en grupo o mediante un circuito de entrenamiento. Existe otro tipo de intervenciones adicionales como, por ejemplo, el uso de dispositivos comerciales de videojuegos o de máquinas de *fitness* con las que realizar ejercicio ya sea en casa o en algún centro especializado.

Existe un rango bastante amplio de intervenciones que pueden llevarse a cabo para mejorar la función del miembro superior tras el daño. Dichas intervenciones pueden estar dirigidas a alteraciones particulares (debilidad muscular) o a habilidades funcionales (agarre y alcance). Clínicamente, sin embargo, los déficits de diversa índole y las complicaciones secundarias derivadas requieren una intervención compleja que integra un conjunto de técnicas para resolver estos problemas [2]. A continuación se muestran algunas de ellas.

1.2.1 Terapia tradicional

El terapeuta se encarga de mover el brazo y las articulaciones con el objetivo de mantener la movilidad. Realiza movimientos en la muñeca y dedos para que el paciente mejore y amplíe su movimiento y se reduzca el dolor. La percepción sensorial y posicional del paciente también puede estimularse con movimiento asistido así como acariciándole y tocándole. La movilización y la estimulación táctil han sido usadas para mejorar el movimiento voluntario aunque no se conoce la dosis óptima.

Existen numerosas formas de mejorar la realimentación posicional y somatosensorial y de motivar al paciente. La facilitación neuromuscular propioceptiva es una técnica de rehabilitación en la que el terapeuta proporciona asistencia o resistencia a unos determinados patrones de movimiento y utiliza la motivación verbal con el objetivo de incrementar el rango

de movimiento y la masa muscular en la extremidad tratada. Debido a que la plasticidad cerebral permite mejoras neuronales que se traducen en mejoras funcionales, el foco del tratamiento se ha desplazado a mejorar el aprendizaje motor asegurándose de que el balance y la masa muscular son suficientes para la práctica del movimiento.

1.2.2 Ejercicios e intervenciones basadas en movimiento funcional

Existen distintas formas de realizar ejercicios, un terapeuta puede asistir en los movimientos o mostrar resistencia a éstos al igual que una máquina de gimnasio. Alternativamente pueden realizarse en clases grupales dirigidas por un terapeuta, utilizar alguna máquina o realizar un circuito de entrenamiento. También es posible llevar a cabo un entrenamiento que comprenda las tareas cotidianas más importantes. El entrenamiento repetitivo consiste en la práctica repetida de tareas funcionales. El entrenamiento bilateral simultáneo de los brazos se basa en la realización de actividades en las que los dos brazos ejecutan movimientos idénticos al mismo tiempo. El aprendizaje motor, reaprendizaje motor o la ciencia del movimiento implica el entrenamiento funcional o de una tarea específica, y, a menudo, es complementado con otras modalidades, como las tecnologías asistidas.

En el entrenamiento de movimiento inducido-restrictivo o terapia de uso forzado, el brazo que no tiene afección se coloca en un cabestrillo o, más comúnmente, en un guante que impida su movimiento, mientras que se van realizando tareas con el brazo afectado e incrementando la dificultad a medida que se van completando. En ocasiones esta intervención se considera una tecnología asistida. El uso activo del brazo afectado en tareas funcionales completas, hace de ésta, una terapia clave.

También existen técnicas de estimulación que no implican esfuerzo motor por parte del paciente. Una de estas técnicas es la práctica mental o visualización, que no conlleva realmente ningún movimiento. Se pueden centrar en reducir la ansiedad, pero lo que se usa más a menudo en rehabilitaciones tras sufrir un DCA implica un ensayo cognitivo imaginando la realización de tareas concretas.

Otra intervención que consigue estimular sin realizar esfuerzo motor es la terapia del espejo. Esta terapia se basa en la estimulación visual mediante la colocación de un espejo en el plano sagital del paciente reflejando el brazo no afectado como si fuera el afectado, haciendo que con los movimientos del brazo no afectado se obtenga la ilusión de que se mueve el brazo afectado. En este tipo de entrenamiento cobran importancia las neuronas espejo, que se activan cuando el paciente ve realizar un movimiento, ya que permiten aprender por imitación.

Precisamente, son las neuronas espejo las que se pretende estimular con la aplicación derivada del presente proyecto, pero, en lugar de hacerlo de manera tradicional como puede ser con la terapia del espejo, se haría con RV. En siguientes capítulos se explicará con detalle en qué consiste y la dinámica del juego/terapia. De manera preliminar, el planteamiento implica que el paciente tendrá que realizar un movimiento de alcance para pulsar un botón con su brazo afectado. El movimiento, que seguramente no lo realice de manera correcta, se traducirá en un

movimiento posturalmente corregido en la RV. Por lo tanto, el paciente asociará la intención y el esfuerzo motor de realizar el alcance con la realimentación visual que recibe, y de esta manera, la influencia será mucho mayor que si simplemente imagina que mueve bien el brazo o ve un movimiento correcto mediante la terapia del espejo.

1.2.3 Tecnologías asistidas

Se puede usar un amplio número de modalidades asistidas, o dispositivos para rehabilitar la extremidad superior, entre las que se encuentran:

- Realimentación electromiográfica: se recoge información sobre la actividad muscular a través de electrodos colocados en la piel o insertados en el músculo. El paciente percibe esta actividad a través de señales eléctricas que se mostrarán en una pantalla o por señal auditiva.
- Dispositivos electromecánicos y robotizados: dispositivos que pueden mover pasivamente las extremidades y proporcionar asistencia o resistencia al movimiento de una única articulación o movimiento coordinado de varias de ellas.
- Estimulación eléctrica: estimulación aplicada a los músculos, ya sea utilizando electrodos superficiales o electrodos percutáneos. Se puede programar la frecuencia, ancho de banda e intensidad apropiadas con el fin de controlar la duración de la estimulación y la duración de intervalo entre estimulaciones. La estimulación podría activarse por la iniciación de la actividad muscular en el músculo a ser estimulado. La estimulación eléctrica se aplica a toda la mano por medio de un guante que puede proporcionar estimulación sensorial.
- Estimulación magnética transcraneal: estimulación del cerebro aplicada a través de una bobina colocada en la cabeza, sobre el área sensorimotora.
- Estimulación transcraneal de corriente directa: similar a la anterior, pero aplicado a través de dos electrodos colocados en la superficie de la cabeza.
- Ortesis: dispositivos externos aplicados a codo, muñeca o articulaciones del dedo para prevenir, limitar o ayudar al movimiento. Se pueden usar solos o con una neuroprótesis que proporcione estimulación eléctrica (un dispositivo ortótico con electrodos preposicionados que ayudan a la función). La subluxación del hombro tradicionalmente se ha tratado con este tipo de aparatos.
- Realidad virtual: software de simulación interactiva ejecutado en un ordenador que proporciona realimentación a la ejecución de un movimiento y/o la consecución de objetivos.

En la siguiente sección se explican las ventajas y motivaciones de la técnica de RV, que han llevado a realizar este proyecto para neurorrehabilitar a pacientes con DCA.

1.3 Realidad Virtual en rehabilitación

La realidad virtual es una tecnología que se lleva desarrollando desde el final de la década de 1950, mediante un casco colocado en la cabeza en el que se situaban imágenes de entornos artificiales para los usuarios [4]. Desde ese primer acercamiento por parte de Philco Corporation en 1958, hasta la RV usada hoy ha habido muchos avances, hasta tal punto que actualmente es posible tener unas gafas de RV simplemente con tener un móvil de última generación y un visor especial determinado (ver Figura 1.3).



Figura 1.3: Samsung Gear RV
Fuente: <http://www.samsung.com/>

La RV en la actualidad puede definirse como una simulación de un entorno real generado por un ordenador en la que a través de una interfaz hombre-máquina se va a permitir al usuario interactuar con ciertos elementos dentro del escenario simulado.

Los ambientes y los objetos virtuales de la RV proporcionan al usuario información visual (a través de distintos dispositivos como pueden ser unas gafas, un sistema de proyección o una pantalla) y en algún caso información auditiva, táctil, olfativa y de movimiento. Existe una gran variedad de dispositivos para interactuar con el entorno virtual, desde los más comunes (ratón, teclado, joystick) hasta alguno más sofisticado con sistemas de captura de movimientos (sensores inerciales). Estos últimos, que son los utilizados para este proyecto, permiten darle al usuario la sensación de que está manipulando objetos reales. Más adelante se analizará en detalle (sección 1.4 Captura de Movimientos).

Los ambientes virtuales pueden proporcionar diferente nivel de inmersión al usuario [5]. El término inmersión se define como "grado de percepción del usuario que le lleva a creer que se encuentra físicamente en el mundo virtual en lugar de en el mundo real". Está relacionado con el diseño del programa y del equipo. Teniendo en cuenta este aspecto, hay dos tipos de dispositivos/sistemas de RV:

- Sistemas de RV “inmersivos”, donde el usuario está integrado totalmente dentro del ambiente virtual viendo solo las imágenes generadas por el ordenador, bloqueándose el resto del mundo físico. En la actualidad, los sistemas “inmersivos” más utilizados son Glasstron, IREX, PlayStation EyeMotion o las gafas Oculus Rift, que son las que se utilizan en este proyecto. Interesa que el sistema a utilizar en este trabajo sea de este tipo para acelerar la recuperación del paciente, ya que, como se mencionó anteriormente, entra en juego el fenómeno de *embodiment* y el aprendizaje gracias a las neuronas espejo, en los que la inmersión cobra un papel importante.
- Sistemas de RV “semiinmersivos” o no “inmersivos”, en los cuales el usuario percibe parte del mundo real y parte del mundo-entorno virtual. No hay una inmersión total en el entorno virtual. En la actualidad, los sistemas no “inmersivos” más utilizados son Virtual Teacher, Cyberglobe, Virtual Reality Motion, Pneumoglobe y Nintendo-Wii.

Como se ha comentado en la sección previa, uno de los principios en los que se basa este proyecto es en la utilización de la RV con el objetivo estimular zonas del cerebro en pacientes para su recuperación funcional tras un DCA. Existen evidencias empíricas que muestran que la efectividad de este entrenamiento viene dada por la reorganización cerebral, activándose nuevas áreas corticales para ayudar al movimiento del brazo afectado. Es destacable que los mayores beneficios aparecieron en pacientes con el accidente reciente, ya que aún no habían adquirido hábitos negativos. Además, es en el periodo próximo al accidente cuando surge la denominada reorganización espontánea del córtex motor, que se extiende aproximadamente tres meses en pacientes que han sufrido un ictus y seis meses en pacientes con traumatismo craneoencefálico [6]. En este tiempo tiene lugar una serie de mecanismos debido al DCA en los que se activan estructuras de aprendizaje y otras partes del cerebro, como puede ser la homóloga contralesional (ipsilateral) para contrarrestar el déficit que supone la muerte de las neuronas debido al daño en la parte lesionada. Esta actividad se iría desplazando hacia su zona natural del hemisferio ipsilesional (contralateral) y se distribuiría alrededor del área dañada, lo que se traduciría en una mejora motora. Durante este periodo es cuando más efecto puede tener la terapia de RV, debido a la activación de esas redes de aprendizaje (córtex motor primario, lóbulos parietales, zona motora suplementaria y cerebelo), que únicamente se da durante la etapa de desarrollo de cada ser humano y resurge para facilitar el reaprendizaje y la plasticidad.

Para visualizar de manera más clara los pros y contras de la RV se cuenta con un análisis DAFO en el que se pone de manifiesto las fortalezas, debilidades, oportunidades y amenazas de esta tecnología en el campo de la rehabilitación [7] (Figura 1.4).

<p>Fortalezas</p> <ul style="list-style-type: none"> • Validez Ecológica • Control de Estímulos y Consistencia • Realimentación en tiempo real • Aprendizaje sin errores • Registro de actividad • Entrenamiento y entorno seguro • Bajo coste y fácil distribución • Factores de juego para aumentar la motivación 	<p>Debilidades</p> <ul style="list-style-type: none"> • Limitaciones de interacción • Limitaciones respecto a pantallas utilizadas • Efectos secundarios adversos • Procesos ingenieriles poco definidos • Usabilidad en interfaces destinadas al terapeuta
<p>Oportunidades</p> <ul style="list-style-type: none"> • Mejora de pantallas y dispositivos • Sistemas de RV inalámbricos • Inteligencia artificial para adaptarse a cada paciente 	<p>Amenazas</p> <ul style="list-style-type: none"> • Conflictos éticos • Competencia medicina tradicional

Figura 1.4: Esquema del DAFO de la RV en rehabilitación en este TFG

Entre las fortalezas más destacadas está la validez ecológica, es decir, la posibilidad de emular entornos reales con facilidad y, de este modo, marcar objetivos reales y similares a tareas cotidianas. El control de estímulos en la RV la convierte en una tecnología ideal, ya que permite adaptar la dificultad de la actividad y a su vez proporcionar una realimentación en tiempo real más natural y funcional. Otra ventaja está en la posibilidad de adaptar el entorno de la RV estableciendo límites o reglas según la necesidad (aprendizaje sin errores) para ayudar y guiar al usuario y que éste no cometa errores. Del mismo modo, permite capturar datos de manera automática y así el propio sistema de RV puede registrar el avance del usuario en la terapia. Al tratarse de una simulación por ordenador es una actividad segura tanto para el paciente como el terapeuta, con bajo coste de desarrollo y fácil distribución entre pacientes. Por último, la introducción en el sistema de características propias de videojuegos incrementa la motivación del paciente.

En la Figura 1.5 se presenta un entorno de RV que aporta gran realismo e inmersión para el usuario.



Figura 1.5: Ejemplo de entorno de RV

Fuente: <http://riftinfo.com/>

Por el contrario, la RV cuenta de igual manera con debilidades, a destacar: la RV es una tecnología que aún no es 100% “inmersiva” lo que hace necesario el uso de periféricos para interactuar con la interfaz, con la complicación que esto conlleva y, aún más, con usuarios con limitaciones físicas. Por otro lado, en la actualidad los sistemas de RV no cuentan con una tecnología refinada en el ámbito de los visores o pantallas, por lo que muchos usuarios no disfrutan de la situación “inmersiva” y, en casos extremos, pueden llegar a marearse y sufrir malestar general después de su uso. Al ser un campo de desarrollo muy fructífero, está constantemente evolucionando con los problemas que esto acarrea, como no contar con procesos ingenieriles claramente definidos y que los desarrolladores deben adaptarse a cada proyecto y controlar el uso de diversos periféricos de manera empírica. Por último, es necesario desarrollar interfaces que permitan a los profesionales clínicos visualizar los resultados de las terapias propuestas con RV de manera intuitiva y útil.

A pesar de estos aspectos controvertidos, las oportunidades de futuro que ofrece la RV invitan al optimismo y perfilan una herramienta con un gran papel en el mundo de la rehabilitación. Su rápida evolución y el desarrollo en el mercado global de pantallas de mayor calidad contribuyen a ello. Junto a estos avances, sería deseable contar con sistemas inalámbricos de RV así como con métodos de inteligencia artificial que den lugar a modelos capaces de adaptarse a cada usuario.

Todo ello hace de la RV una herramienta con un inmenso potencial y una apuesta segura de futuro. Por último, las amenazas sobre la RV son también un factor importante, como son los conflictos éticos que plantea o la competencia con el campo de la medicina tradicional y su marco de negocio. Sin embargo, haciendo un balance general son muchas más las posibilidades de futuro que los problemas que pueda plantear por lo que se augura un desarrollo y un gran

crecimiento como una herramienta estándar en el campo de la rehabilitación del presente y futuro.

La Figura 1.6 presenta una predicción sobre el uso de la RV para el año 2025. El primer gran mercado será el de los videojuegos, al fin y al cabo la RV se ideó inicialmente para ello y hoy día es ya el más desarrollado. Sin embargo, en segundo lugar aparece el uso de la RV dedicado a la medicina con un valor total aproximado de 5.1 billones de dólares, muy por encima de entretenimiento por vídeo o eventos en vivo.

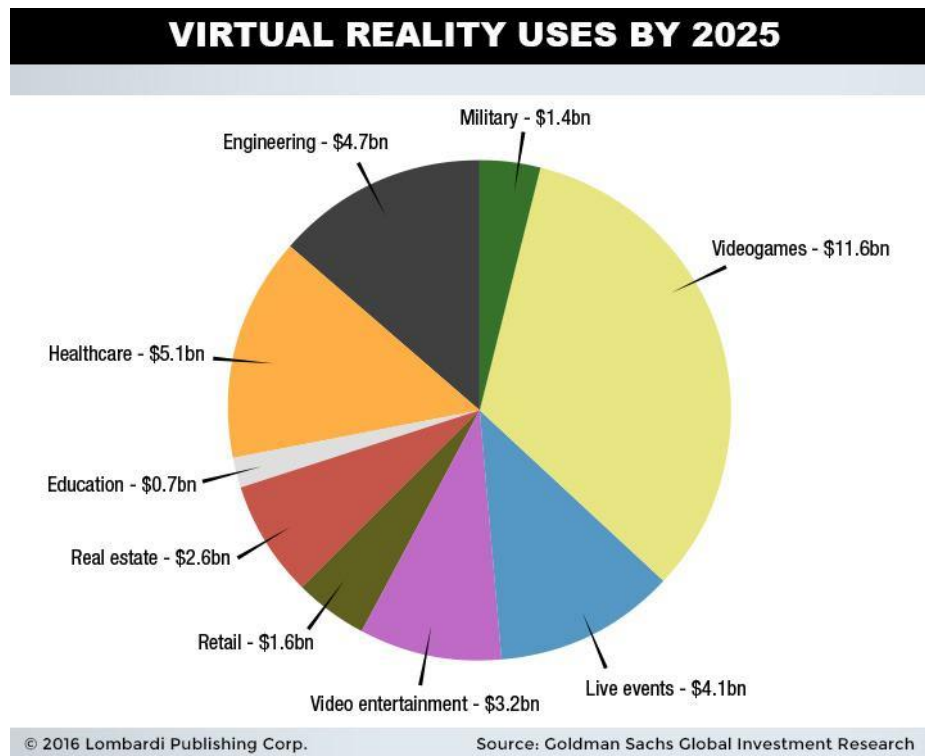


Figura 1.6: Predicción de futuro del uso de RV
Fuente: <http://www.profitconfidential.com/>

1.4 Captura de movimientos

Uno de los elementos más importantes en la experiencia de RV es su capacidad de emular un entorno real. Tal y como se mencionó anteriormente, uno de los inconvenientes de esta tecnología es la necesidad de usar periféricos para conseguir la inmersión de los usuarios como, por ejemplo, un mando para controlar objetos. Sin embargo, gracias al desarrollo del campo de los videojuegos, existe un amplio abanico de posibilidades en cuanto a dispositivos para registrar los movimientos de los usuarios. Por ello, ya no es preciso utilizar periféricos especializados del ámbito sanitario con el alto coste que conlleva.

Hay multitud de posibilidades tales como joysticks o mandos tradicionales. Sin embargo, dada la naturaleza de este proyecto y en general de todas las terapias de rehabilitación mediante RV, los pacientes pueden sufrir discapacidades motoras o cognitivas que les impida utilizar correctamente estos dispositivos menos intuitivos. Por ello, se ha popularizado el uso de otros periféricos tales como Kinect [8], PlayStation Eye [9], Wii Remote Plus [10], o en nuestro caso, el brazalete Myo Armband [11]. Todos ellos con una característica común: abstraer al usuario de que está utilizando un periférico para capturar sus movimientos y acciones (Figura 1.7).



Figura 1.7: Rehabilitación con el sistema de captura de movimiento Kinect.

Fuente: <http://www.cosasdesalud.es/kinect-en-la-rehabilitacion-de-la-esclerosis-multiple/>

De este modo, una persona afectada por un problema cognitivo que le impida memorizar botones o colores, del estilo de un mando tradicional, al utilizar un dispositivo de los anteriormente mencionados puede interactuar mediante sus movimientos naturales y realizar la terapia sin más impedimentos que los generados por su propia afección.

Las ventajas del uso de estos dispositivos en la rehabilitación son numerosas, siendo una de las más importantes la ya mencionada, de facilitar la terapia a pacientes con problemas motores o cognitivos, pero también nos encontramos ante un sistema divertido que motiva al usuario a superarse a la vez que realiza los ejercicios que le corresponden. Al utilizarse en otros ámbitos como los videojuegos, los usuarios están más familiarizados con su uso y permite monitorizar en tiempo real las mediciones obtenidas e incluso que el terapeuta lleve un control de la terapia de manera remota abaratando así costes en un futuro [12].

1.5 Antecedentes de rehabilitación mediante RV

La RV se aplica actualmente en rehabilitación motora con el objetivo de mejorar las capacidades funcionales del paciente. Al utilizar sistemas de RV los ambientes y objetos

virtuales proporcionan al usuario información visual, auditiva, táctil e incluso olfativa. Existe una gran variedad de interfaces y elementos que proporcionan realimentación al usuario para darle mayor realismo a la experiencia tales como tener la sensación de que manipula objetos de la interfaz.

En la actualidad, el campo donde más se aplica RV para rehabilitación es en pacientes con patologías de origen neurológico ya sea DCA, Parkinson o lesiones medulares. Los resultados son positivos y motivan a los pacientes a alcanzar el máximo nivel dentro de un juego que, traducido en la realidad, sería el movimiento funcional ideal.

Uno de los aspectos más importantes de la rehabilitación mediante RV es la terapia de imitación de la que se tienen indicios de que la actividad generada en la corteza prefrontal y en el cerebelo es la responsable de la recuperación motora asociada a este tipo de terapia. Sin embargo, una de las conclusiones principales a las que llegaron Viñas-Diz y Sobrido-Prieto en su investigación es que se necesitan más estudios que profundicen en los posibles cambios corticales producidos por la RV en rehabilitación [5].

Un ejemplo de terapia mediante RV es el trabajo de investigación BioTrak, un sistema de RV para la rehabilitación del equilibrio en pacientes con daño cerebral [13]. En él se investigó si utilizando RV se podía mejorar el equilibrio de una persona con daño cerebral para lo cual sometieron a 10 pacientes a 20 sesiones de RV (ver Figura 1.8). Los resultados confirmaron una mejora durante, al final y un tiempo después de la terapia en el equilibrio de la población a estudio.



Figura 1.8: Paciente utilizando BioTrak

Fuente: <http://inforuvid.com/>

Otro trabajo interesante se ha llevado a cabo en la Universidad Carlos III de Madrid (UC3M). La investigación tenía como objetivo rehabilitar una lesión física de hombro sin daño

neuroológico y obtuvo igualmente buenos resultados tras su aplicación a los pacientes afectados [14]. La inutilización del brazo lesionado, también puede provocar que empeore la representación en el mapa neuronal de esta extremidad aunque en un principio no se tenga daño neuroológico. Como ya se ha explicado, en pacientes con DCA, a los que va dirigido el presente trabajo, influyen dos factores en la alteración neuronal, uno es el propio daño y otro es la falta de uso de la extremidad afectada y consiguiente compensación con la extremidad sana. En el trabajo de la UC3M sólo entraría en juego el segundo factor.

Utilizaron unas gafas Oculus Rift DK2, iguales a las utilizadas en este proyecto, y un detector de movimiento Intel RealSense, similar a un Kinect. Con estos dos periféricos desarrollaron un juego mediante Unity que le permite al paciente realizar rehabilitación sobre el hombro afectado. El concepto es sencillo, simula un juego en el que el usuario actúa de portero de fútbol en una interfaz y el objetivo es atrapar los balones que se dirigen hacia él. De este modo realiza, de manera amena y mediante objetivos, movimientos de aducción y abducción del hombro afectado (Figura 1.9)

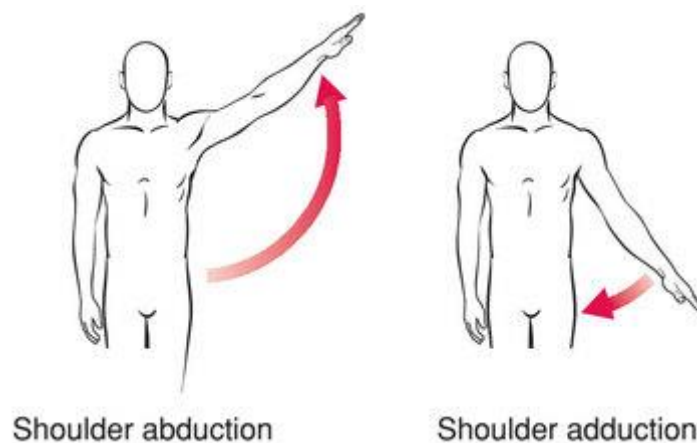


Figura 1.9: Movimiento de abducción y aducción
Fuente: www.nolanlee.com/blog/anatomy-language-part-ii

Un aspecto importante de ese trabajo es el hecho de que mejora la propiocepción del paciente, que se define como la capacidad del ser humano para conocer la posición, movimiento y aceleración de las distintas partes del cuerpo y su relación con el entorno [15]. Para potenciar el sistema propioceptivo, el paciente solo es capaz de ver la posición de su mano en el entorno virtual pero no el resto del cuerpo, ni siquiera el brazo, y, de esta manera el usuario hace un esfuerzo extra al realizar el movimiento. Tras concluir su desarrollo la terapia fue evaluada por expertos en rehabilitación llegando a conclusiones muy positivas sobre su efectividad.

En conclusión, en ambos trabajos, tanto BioTrak como el de la UC3M, se probó la efectividad y el potencial de la rehabilitación tradicional apoyada en técnicas de RV. Los pacientes la acogieron con entusiasmo y, a falta de realizar más pruebas y ampliar el espectro de evaluación en más pacientes, sus resultados fueron positivos y aún mejor, se mantuvieron hasta 3 meses

después de finalizar el tratamiento, lo cual invita al optimismo y perfila la RV como un pilar importante en nuevas técnicas de rehabilitación.

En resumen, los resultados obtenidos en trabajos similares a este, son un estímulo positivo y auguran este proyecto como una herramienta muy útil y eficaz.

1.6 Objetivos del Trabajo de Fin de Grado

El objetivo principal del TFG es el diseño y desarrollo de un entorno, software y hardware, basado en RV para la realización y seguimiento de terapias de neurorrehabilitación en el movimiento funcional de alcance del miembro superior.

El trabajo objeto de esta memoria está basado en tecnología de RV gracias a muchas de las ventajas que aporta en el ámbito de la rehabilitación y que ya hemos mencionado, como su capacidad de hacer sentir al usuario que se encuentra en un entorno distinto al de su realidad, es decir, su capacidad “inmersiva” y lo que esto aporta, de acuerdo a diversas evidencias científicas, a la recuperación del cerebro del paciente.

Todos los estudios de rehabilitación mediante RV revisados se centran en simular los movimientos del paciente, es decir, en mostrar exactamente lo que hace, por lo que el cerebro recibe un estímulo visual igual al de la realidad. Sin embargo, si el movimiento que realiza el paciente no es el deseado debido a sus limitaciones pero en la RV lo percibe de manera correcta, el estímulo visual que llega al cerebro activaría el área cortical del movimiento visualizado y le permitiría reaprender el gesto. Como se ha explicado con anterioridad, en este reaprendizaje están relacionados varios factores, como son la intención del paciente de realizar el movimiento de alcance, el esfuerzo motor que esto provoca, y la realimentación visual corregida. Como el estímulo visual va a ser controlado, la realización motora tenderá a igualarse, ya que el cerebro tiende de manera natural a corregir contradicciones entre ambas cosas, y de esta manera el paciente irá mejorando. Por lo tanto, son muchos los condicionantes de la terapia que propone el sistema desarrollado en este proyecto que pueden acelerar la neurorrehabilitación del paciente.

Cuando una persona sufre un daño cerebral, las consecuencias vienen dadas por la zona afectada del cerebro y las interconexiones con redes neuronales de otras zonas que se pierden. Es decir, la pérdida de movilidad del brazo no viene dada por un problema de la propia extremidad sino porque las conexiones del cerebro con la extremidad se han dañado gravemente tras el DCA.

El principio básico sobre el que gira la efectividad de aplicar la terapia basada en el sistema desarrollado en este proyecto es el de la plasticidad cerebral. Esta se define como la capacidad adaptativa funcional que tienen las neuronas, es decir, que las células nerviosas que inicialmente estaban dedicadas a una función puedan adaptarse para realizar otra tarea de

manera progresiva. Por lo tanto, este proyecto está enfocado en neurorrehabilitar y no en el desarrollo de una terapia motriz.

La neurorrehabilitación es una subespecialidad de la neurología que incide en la plasticidad cerebral, con la que se regeneran estructuras neuronales y/o se reorganiza se reorganiza la función de las neuronas como respuesta al daño y como resultado de la actividad motora, incluyendo ejercicios de coordinación y entrenamiento funcional. Con el inicio precoz del tratamiento se pueden minimizar las posibilidades de adquirir malos hábitos o posturas [16].

Al sufrir un DCA las neuronas de la zona lesionada mueren de manera irreversible y, por tanto, la función que desempeñaban queda afectada en mayor o menor medida. Por ejemplo, una persona que sufra un accidente cerebrovascular en el hemisferio izquierdo de su cerebro tiene altas probabilidades de sufrir problemas motores en la parte derecha de su cuerpo, contralateral al daño (Figura 1.10) ya sea el brazo, pierna o cara.



Figura 1.10: Hemisferios cerebrales y control motor

Fuente: sites.google.com/site/alanrosales209/home/hemisferio-derecho-e-izquierdo-del-cerebro

1.6.1 *Objetivos específicos*

Además de los objetivos generales existe una serie de objetivos específicos:

- Rehabilitación mediante juego, que facilitará su ejecución además de potenciar el entretenimiento y la motivación.
- Terapias de libre configuración mediante unos parámetros. Adicionalmente a esto existirá posibilidad de guardar y cargar terapias ya existentes.
- Sistema fácilmente aplicable a distintos pacientes.
- Entorno usable e intuitivo tanto para el médico como para el paciente.
- Realizar un registro de la actividad realizada a modo de diagnóstico extra.
- Ejecución fluida sin necesidad de contar con equipos especiales.
- Posibilidad de utilizar el sistema en el domicilio del paciente (Telemedicina).

- Escalable a rehabilitación de extremidades inferiores.
- Sistema de neurorrehabilitación de bajo coste.

1.6.2 Novedad del TFG

Mediante la plasticidad cerebral, las neuronas no afectadas adyacentes podrían ser capaces de adaptarse para realizar la función perdida y así recuperar la autonomía evolucionando funcionalmente a un estado similar anterior al accidente.

Aquí es donde la RV de este proyecto juega un papel vital, ya que mediante el estímulo visual, el paciente creerá observar el movimiento de su brazo de manera ideal, aunque su brazo no lo realice así. De este modo el cerebro reaprenderá el movimiento correcto progresivamente y las células nerviosas podrán reaprender la función que se perdió al sufrir el DCA.

Por otra parte, debido al fenómeno conocido como *embodiment* [17], tras el daño cerebral, el paciente va a tener la sensación de estar en un cuerpo que no controla y de ahí que se activen los mecanismos que alteren su representación neuronal. Si el paciente siente como suyo el brazo representado en la RV, y que éste está respondiendo posturalmente de forma correcta, se podría contribuir a la no alteración y reordenación neuronal. Nuestro cuerpo y nuestra morfología determinan el mapa sensorimotor de nuestro córtex cerebral, es decir, "el cuerpo moldea la mente".

Del mismo modo, es posible influir en una recuperación más rápida gracias a las neuronas espejo, ya que éstas participan en procesos tales como el aprendizaje de conductas complejas a través de la observación. Estas neuronas se activan tanto al realizar una acción como al ver a otro realizar esa misma acción [18]. De esta manera, cuando el paciente intente realizar un movimiento y vea que su ejecución es la correcta incrementará su aprendizaje.

Con el fin de potenciar todos estos fenómenos (plasticidad, *embodiment*, neuronas espejo) se decidió que la mejor alternativa para la neurorrehabilitación era desarrollar un sistema de RV "inmersiva".

Gracias a ello, la terapia de rehabilitación a realizar con esta plataforma será diferente a todas las anteriores que emplean RV ya que el objetivo es influir en las conexiones cerebrales mediante el estímulo visual de la RV y los fenómenos explicados, que pueden acelerar la recuperación, y no únicamente sobre la función motriz del paciente como suelen hacer tanto las terapias tradicionales como las que emplean nuevas tecnologías.

En las siguientes secciones se analizan los materiales empleados y metodología seguida para el desarrollo del sistema de neurorrehabilitación. Posteriormente se exponen los resultados obtenidos y, por último, se concluye analizando aspectos relevantes como el impacto del proyecto y el trabajo futuro.

1.7 Introduction

Nowadays the computer science field advances with big steps. It is so much so that one of their immediate potentials is related with modern medicine.

The intention of this project is to contribute and help on this field developing a virtual reality (VR) interface, which allows to rehabilitate the arm reach movement in patients with problems in the upper limbs as a consequence of brain damage. This provides an applicable solution to a real problem and, attached to that, the possibility to develop with modern technologies were a great motivation source.

This chapter is divided in several points. Firstly, it is explained what is an acquired brain damage, their causes and the most important deficits it brings such as those related with the upper limbs motor skills and how this system pretend to provide an effective solution. It has also been exposed the different types of rehabilitation which have been helpful for this issue, from traditional therapies to the most innovative such as the one developed in this project with VR. Then, after that, it is explained the strengths and weakness of this type of therapy with a SWOT diagram. Also, it will be reviewed previous investigations, which use neurorehabilitation and VR, with positive results. Finally, to conclude, they are presented the most novel aspects of this project, thanks to those it could accelerate the recovery of patients.

Capítulo 2. Materiales y Métodos

En este capítulo se presenta la metodología empleada y los materiales necesarios para realizar este proyecto. A continuación se detallan a nivel técnico los dispositivos y tecnologías empleadas para, posteriormente, pasar a explicar la metodología seguida en el desarrollo del proyecto.

2.1 Tecnologías empleadas

En cuanto a Software se utilizó principalmente el motor de desarrollo de videojuegos Unity 5.3 [19] junto con el editor MonoDevelop [20] para la programación de Scripts en C#. La creación de la persona con esqueleto a utilizar en Unity para la rehabilitación se realizó con la aplicación MakeHuman [21].

A nivel de dispositivos Hardware fue necesario contar con unas gafas de RV, en este caso Oculus Rift Development Kit 2 [22], y un dispositivo de captura de movimientos como el Myo Armband [11]. Sin embargo, el desarrollo del proyecto se llevó a cabo en ordenadores portátiles personales sin grandes especificaciones técnicas más allá del soporte gráfico para la RV.

2.1.1 Entorno de desarrollo: Unity 3D

El objetivo del proyecto es proporcionar un entorno de RV “inmersivo” para realizar terapias de rehabilitación en pacientes. Por lo que, en un principio, se plantearon dos posibilidades para desarrollar la interfaz: Unreal Development Kit (UDK) o Unity, ambos en la versión gratuita.

Tras valorar las opciones que ofrecían ambos motores de desarrollo se eligió Unity por varios motivos:

- Basado en los conocimientos previos que se contaban en el uso de la herramienta, lo cual nos parecía determinante a la hora de realizar un trabajo de calidad y consistente.
- Otro motivo importante, por el cual la balanza se inclinó por este motor, fue el hecho de la integración que ofrecía con los periféricos necesarios para el desarrollo del proyecto. Tanto las gafas de RV como el brazalete Myo Armband se acoplaban mejor en Unity y su funcionamiento prometía ser más automático que en la opción de UDK.
- Por último, pero no menos importante, la cantidad de recursos e información sobre el motor de desarrollo de Unity era muy extensa y la multitud de proyectos realizados con este motor invitaba a formar parte de una comunidad enorme y proactiva.

Un motor de juegos o “game engine” se define como la rutina de programación que permite el diseño, creación y representación final de un videojuego. En nuestro caso, se adapta perfectamente ya que se pretende hacer un entorno para rehabilitación pero que tenga motivaciones similares a un videojuego marcado por objetivos.

La funcionalidad de un motor de juegos se caracteriza por otorgar al desarrollador un motor de renderizado para el apartado gráfico, tanto en 2D como 3D, un motor de detección de colisiones y de física, animaciones, sonidos o scripting, entre otros.



Figura 2.1: Logo Unity
Fuente: <https://unity3d.com>

Unity es un motor de videojuegos programado esencialmente en C, C++ y C# y desarrollado por la compañía Unity Technologies.

La primera versión fue lanzada al mercado en el año 2005 y tuvo un gran impacto en el mercado del desarrollo de videojuegos ya que se perfilaba como una alternativa más que válida a Unreal, clásico dominador del campo del desarrollo de videojuegos, en cuanto a funcionalidad y con un factor diferenciador determinante: era gratuito.

Bajo el lema “Unity se creó con la visión de democratizar el desarrollo de juegos y nivelar el campo de juego para desarrolladores de todo el mundo” [23] radica el éxito de este motor de juegos y así se abrió paso de manera exponencial en el mercado, proponiendo un software de calidad, barato y que permitía el acceso al mundo del desarrollo de videojuegos a todos de manera global.

Unity cuenta con una versión Professional, de pago, y una personal, gratuita. La versión de pago cuenta con características adicionales que afectan al renderizado del motor, por ejemplo, mientras que la versión gratuita es más limitada y promociona la marca Unity con su logo a modo de publicidad.

Se han sucedido muchas versiones y cambios hasta hoy con la versión 5, lanzada en 2015. Este tipo de desarrollo tan positivo se consigue gracias a la inmensa comunidad detrás de Unity trabajando para su mejora a diario y eso lo ha convertido en claro dominante a nivel mundial como vemos en la comparativa de la Figura 2.2.



Figura 2.2: Uso de Unity en el mundo
 Fuente: <https://unity3d.com>

Se puede apreciar que casi la mitad de desarrolladores globales usan Unity en sus proyectos, cuentan con 4.5 millones de desarrolladores registrados y su participación de mercado es del 45% y como se ve en la Figura 2.3 esto implica que dobla en porcentaje a su inmediato competidor.



Fuente: Informe inédito de McKinsey

Figura 2.3: Cuota de mercado de Unity
 Fuente: <https://unity3d.com>

Otro de los aspectos más destacados de Unity y su comunidad es el Asset Store. Lanzado en el año 2010, es un recurso del editor en el que se pueden descargar multitud de paquetes, incluyendo modelos 3D y contenidos para implementar en cualquier proyecto. Este es otro punto a favor de este software y el cual se ha utilizado para mejorar y perfeccionar el diseño 3D de la interfaz, al no contar con un experto diseñador.

Por todos los motivos mencionados la opción de desarrollar con Unity fue la elegida y, dadas las características del proyecto, la licencia ideal era la personal o gratuita.

El primer paso fue descargar la última versión (5.3.2 con su posterior actualización a la versión 5.3.4) del software de la página oficial e instalarlo en los equipos de desarrollo y familiarizarnos con el editor con ayuda de la documentación oficial de Unity [24], un recurso muy completo y que fue una herramienta primordial en el proyecto junto al foro de la comunidad [25].

El aspecto del editor de Unity se presenta en la Figura 2.4.

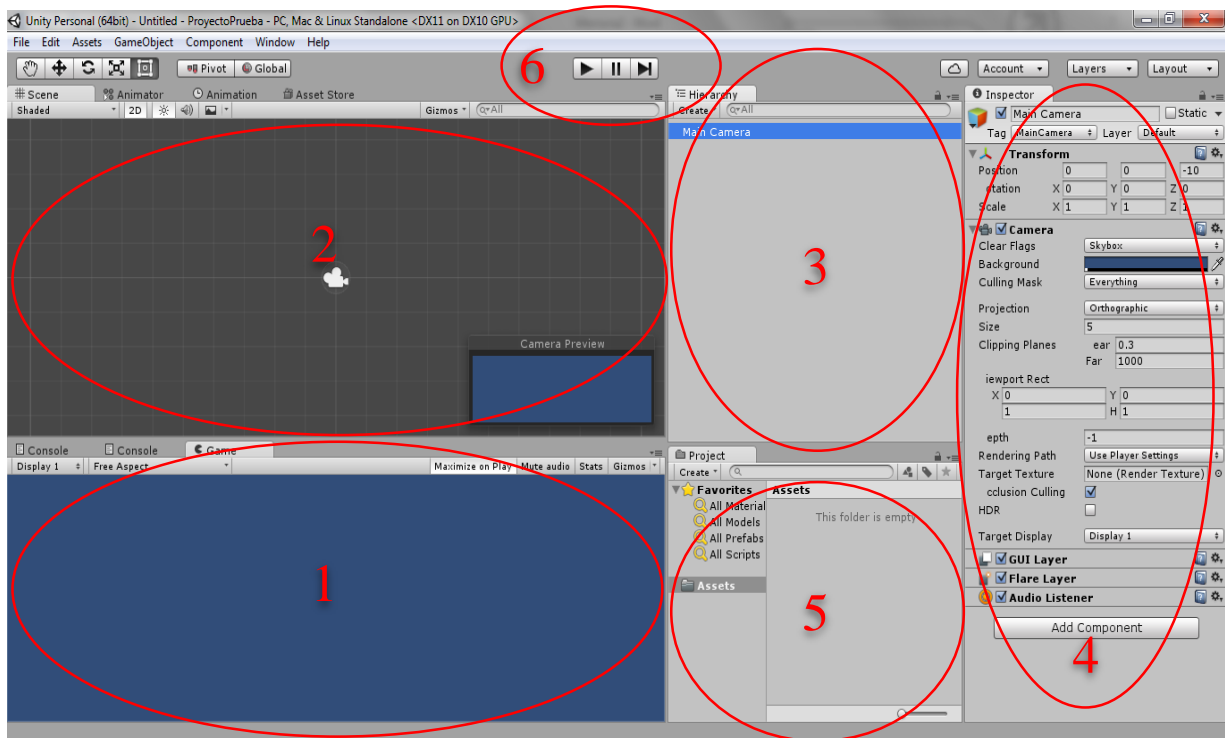


Figura 2.4: Aspecto editor Unity

Como se aprecia cuenta con diversas ventanas incluyendo una vista del juego actual: la azul inferior *Game*, número “1”, en la que se visualiza el aspecto del juego reproducido por la cámara según se va trabajando sobre él.

La organización básica de un juego en Unity es mediante escenas. Una escena representa una determinada pantalla o escenario distinto del videojuego, por ejemplo, el menú principal y la interfaz de rehabilitación son dos escenas diferentes. En la imagen vemos la ventana de la escena, *Scene*, número “2”. La diferencia principal entre la vista de *Game* y la vista de *Scene* es que en *Game* se visualiza el juego tal y como se verá en modo de ejecución según la cámara que esté definida, por lo que no es editable. Sólo refleja en el juego los cambios que se realizan en la escena, mientras que en la vista de la escena se puede visualizar toda la situación y es donde se realizan los cambios sobre los objetos y elementos que forman la escena concreta.

Cada escena está compuesta por diversos objetos definidos en una jerarquía, *Hierarchy* número “3” en la imagen, entre los que se incluyen por defecto un objeto cámara que permite visualizar el entorno actual. En dicha jerarquía se establecen todos los objetos que interactúan en la escena

como sonidos, luces, objetos 3D e incluso objetos de tipo GUI para definir interfaces dentro de la aplicación.

La siguiente ventana es el inspector, *Inspector* número “4”, que se utiliza para ver y editar propiedades de los objetos. Al seleccionar un objeto en la jerarquía, en este caso la cámara, el inspector mostrará sus propiedades y los componentes que contiene, como por ejemplo su posición en el espacio, dada por el componente *Transform*, un campo de colisión física o un script de programación.

Por último la ventana “5” *Project* muestra el directorio del proyecto actual donde se crean los scripts, se importan paquetes o demás elementos asociados a su desarrollo.

El modo de juego, *Play Mode* “6”, nos permite controlar como se verá el juego publicado. Al pulsar *play* el juego entrará en ejecución en el modo de edición, donde se podrá ver su comportamiento y se podrá editar mientras se ejecuta pero los cambios no se guardarán una vez finalizada la ejecución [24].

Una vez familiarizados con el editor, el siguiente paso fue decidir qué lenguaje de programación utilizar para los scripts de los objetos del juego. La programación de scripts es esencial en cualquier juego, ya sea para controlar eventos de entrada del usuario o eventos de la propia lógica del juego.

Unity soporta dos lenguajes:

- C#: un lenguaje de la industria estándar similar a Java o C++.
- UnityScript: un lenguaje diseñado específicamente para Unity y basado en JavaScript.

Tras estudiar qué opción sería la más acertada se llegó a la conclusión de que C# se adaptaba mejor a la aplicación ya que no estaba orientada a web además de ser un lenguaje de propósito general con más posibilidades que su competidor [26]. Por otro lado, la posibilidad de aprender un nuevo lenguaje de programación popular y basado en otros ya conocidos como C++ o Java fue un gran incentivo para tomar la decisión.

La estructura básica de un script en Unity consta de dos métodos por defecto: *Start* y *Update*. El primero se ejecuta al crearse el objeto que contenga el script, similar a un constructor de Java, mientras que el segundo se ejecuta una vez cada *frame*.

Hay que tener claro que el concepto de un videojuego se podría definir como el de una aplicación que tiene un gran bucle que actualiza el estado de todos los elementos que intervienen en el mismo, de una manera muy rápida, como el *Update* en cada *frame*, y así conseguir engañar al ojo humano y que parezca todo una acción continua y natural.

Esta particularidad del método *Update* era perjudicial ya que no es un método que se ejecute de manera constante en el tiempo sino que depende de los *frames* que se generen de manera dinámica. Afortunadamente Unity cuenta con el método *FixedUpdate* que se ejecuta cada 20 milisegundos exactos, lo cual lo hace ideal para programar teniendo en cuenta leyes físicas (el tiempo transcurrido siempre es el mismo).

Tras haber resuelto dudas iniciales mediante la documentación y la comunidad de Unity empezó el grueso del desarrollo del proyecto.

2.1.2 MakeHuman

MakeHuman es una aplicación de software libre que sirve para crear prototipos realistas de humanos en 3D para ilustraciones, animaciones y videojuegos [21]. El contenido creado con MakeHuman obtiene la licencia CC0 (dominio público), lo que supone una libertad total de uso.



Figura 2.5: Logo MakeHuman

Fuente: patryglez.blogspot.com.es/2014_01_01_archive.html

El modelo inicial de humano estándar que suministra la aplicación puede irse modificando mediante controles intuitivos hacia un humano masculino o femenino, cambiando la estatura, anchura, edad, entre otros aspectos (Figura 2.6).

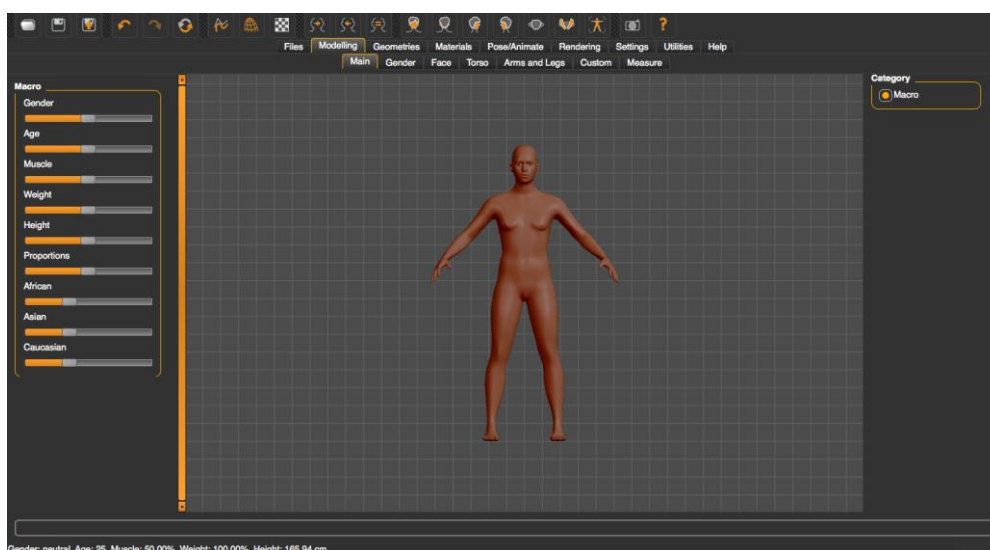


Figura 2.6: Aspecto de la aplicación MakeHuman. El humano base se puede ir moldeando para que adquiera distintos aspectos.

En un principio se pensó en adquirir un humano o brazos ya hechos a través del Asset Store de Unity. Sin embargo como el paciente debía sentirse inmerso para acelerar su recuperación, se descartó la posibilidad de que fueran sólo unos brazos y se eligió un humano entero para que pudiera mirar hacia abajo y ver el resto del cuerpo y así sentirse más dentro de la escena.

Tras decidir que sería un cuerpo entero el utilizado se realizó una investigación para obtener un prototipo de calidad. Al ver la facilidad de uso y la corta curva de aprendizaje que permitía MakeHuman, se decidió crear el humano con esta aplicación, ya que permitía que éste tuviera esqueleto y resultaba fundamental tener huesos en el brazo y antebrazo para reproducir el movimiento de alcance como se explicará más adelante en la sección "2.2.1.6.1 Reproducción del alcance del brazo en la RV". Otro aspecto destacado para la elección de MakeHuman para el diseño del personaje fueron las facilidades que otorgaba en cuanto a exportar el prototipo a Unity.

2.1.2 Captura de Movimiento: Myo Armband

El brazalete Myo Armband es una pulsera que permite interactuar con diversas aplicaciones de manera intuitiva gracias a las señales que recoge del usuario que la utiliza.

Se escogió para la realización de este proyecto por ofrecer un dispositivo de calidad a un precio más económico que los utilizados tradicionalmente en el campo de la medicina. Gracias a esto se podría esperar en un futuro que fuera asequible y que permitiese a los pacientes tener, junto con la aplicación y unas gafas de RV, un sistema de rehabilitación propio en su casa.

Desarrollada por la compañía Canadiense Thalmic Labs, este periférico salió al mercado a mediados del año 2014 con una gran expectación en el público por su aspecto y funcionalidades de carácter futurista (guarda cierta similitud con el dispositivo utilizado por el protagonista de la obra cinematográfica Ironman [27]).

Se presenta como un dispositivo *wearable* (usable) cómodo de llevar, y su premisa básica es controlar diversas aplicaciones mediante gestos o movimientos sencillos por parte del usuario. Para ello cuenta con un acelerómetro, un giroscopio y un total de 8 sensores de electromiografía (EMG) capaces de registrar mediante la actividad eléctrica los movimientos y alteraciones musculares producidas.

En el caso que nos ocupa, solo se tomarán datos del acelerómetro pero los sensores EMG apuntan a una versión posterior de este trabajo que permita al usuario una mayor interacción con el entorno además del movimiento de alcance del brazo afectado.

Se puede apreciar el aspecto de la pulsera en la Figura 2.7. Cada bloque lleva incorporado un sensor EMG mientras que el bloque central cuenta con el giroscopio y el acelerómetro antes mencionado.



Figura 2.7: Myo Armband
Fuente: <https://www.thalmic.com/>

Las funcionalidades de este dispositivo son inmensas y van desde videojuegos o control de drones hasta el campo de la medicina o del entretenimiento en vivo como los conciertos.

Uno de los mayores potenciales de esta tecnología es la posibilidad de descargar el SDK (Software Development Kit) de manera gratuita y sencilla, tanto para Windows, Mac, iOS y Android. De este modo, se abre la puerta a los desarrolladores y a la implementación de multitud de aplicaciones utilizando el Myo Armband, como por ejemplo la desarrollada en este proyecto.

El SDK contiene un paquete de tipo Unity por defecto, el cual permite probar funcionalidades básicas del brazalete controlando un martillo con la mano y haciéndolo cambiar de color mediante gestos. Esto puso de relevancia la fácil integración de Myo Armband con Unity y, de este modo, se utilizó para leer los datos que recogía la pulsera.

Entrando en el apartado más técnico del dispositivo, es muy importante reseñar que funciona con una frecuencia de 50Hz, es decir, recoge datos cada 20 milisegundos. Esto, aunque parezca muy rápido, no lo es si se quiere simular movimientos de manera continua como se pretende en este proyecto tal y como se explica más adelante en la sección "Captura de Gestos y Movimientos". Por fortuna, el método antes comentado FixedUpdate de Unity se ejecuta cada 20 milisegundos por lo que es posible sincronizar la actualización de los objetos de la interfaz con la obtención de nuevas mediciones de la pulsera en cada intervalo.

Por ello, gracias a sus características positivas como su relación calidad precio y las posibilidades que ofrece su SDK, el Myo Armband es el dispositivo utilizado en el proyecto para registrar movimientos.

2.1.3 Dispositivo de RV: Oculus Rift DK2

El periférico elegido para tener la experiencia mediante RV fueron las gafas de RV Oculus Rift DK2 (Figura 2.8). Los motivos son varios. Entre ellos destacan su calidad y experiencia en el campo de la RV, su fácil integración con Unity y su extenso material incluido en el SDK.

Esta versión, que sustituye a la DK1, lanzada a mediados del año 2014, cuenta con un nuevo *display* (pantalla) *Low Persistence* OLED que disminuye considerablemente los efectos que se describieron en el DAFO del capítulo anterior que hablaban sobre malestar general y mareos, mientras que con su dispositivo de posicionamiento mejorado contribuye a la experiencia de RV de manera positiva.

El funcionamiento básico esta tecnología es el de fraccionar la imagen que se visualiza de modo que se obtiene en bifocal. De esta manera, el usuario puede mirar con ambos ojos tal como lo haría en la realidad. A esto se le suma su diseño ergonómico que le aísla del exterior y aumenta la sensación de inmersión para la persona que lo utiliza.



Figura 2.8: Aspecto frontal y trasero Oculus Rift DK2

Fuente: <http://es.engadget.com/>

Integrar las gafas de RV en Unity fue un proceso relativamente sencillo, no así conectarlas en los equipos de desarrollo. Es necesario contar con una tarjeta gráfica de ciertas características para poder visualizar todo correctamente orientado y sin cortes en las lentes de RV. Habiendo conseguido esto, el editor de Unity cuenta con unos aspectos de configuración donde aparece la opción de habilitar para RV el proyecto que se esté diseñando. Tras activar dicha configuración, al ejecutar el juego, se puede visualizar mediante las gafas de RV el entorno diseñado desde la perspectiva de la cámara principal de la escena de Unity.

Por todo esto, el aspecto de integrar el dispositivo de RV con la aplicación resultó no muy complicado y se realizó en fases avanzadas de la implementación.

2.2 Métodos

La metodología de trabajo empezó por abordar el primer problema: decidir el motor de juegos para desarrollar el entorno de rehabilitación. La elección de Unity vino dada por varios motivos, como ya se explicó anteriormente, pero uno de los más importantes fue las facilidades que otorgaba a la hora de integrar el brazalete Myo Armband y las gafas de RV Oculus Rift DK2. Tras comprobar que se accedía correctamente a los datos en crudo generados por la pulsera, se empezó el desarrollo del entorno. Para ello se ha seguido la metodología típica de desarrollo de Unity que consiste en la consecución de diferentes escenas, cada una destinada a un propósito.

En cuanto a metodologías aplicadas cabe destacar los controladores de escenas. Gracias a la abstracción de un objeto vacío que se denomina "ControladorEscena". Este objeto cuenta con un script asociado programado en C# que se encarga de gestionar todos los eventos incluyendo la entrada del usuario y la lógica de la escena en cuestión.

Otro método empleado ha sido la implementación de una clase llamada "Manager", que contiene variables globales para gestionar el juego. Tales como la escena actual o la distancia máxima a la que el paciente puede estirar el brazo. Esta clase está basada en el patrón de diseño "Singleton" (véase Figura 2.9) cuyo propósito es garantizar que solo exista una instancia de una clase y proporcionar un único punto de acceso global a la instancia. Con ello se garantiza que solo exista una instancia de la clase "Manager", es decir la existencia singular de las variables globales, y se obtiene un método de acceso global para su modificación o lectura.

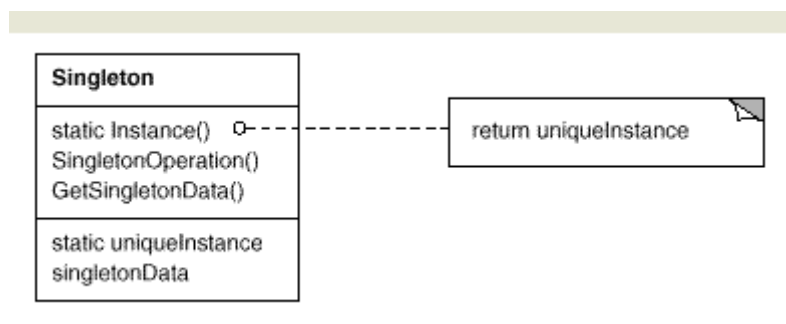


Figura 2.9: Diagrama OMT Singleton

Fuente: <https://msdn.microsoft.com/es-es/library/bb972272.aspx>

Para la transición entre una escena y otra se implementó una mecánica para simular un *fade*, que se logra gracias a un fondo negro que se va oscureciendo progresivamente según unos parámetros y así conseguimos una transición visualmente más suave entre pantallas diferentes.

2.2.1 Escenas

Como se ha comentado anteriormente, los juegos en Unity se dividen en escenas. En este proyecto se cuenta con un total de 8 escenas, algunas dirigidas para el personal clínico y otras para el paciente, siendo solo visible en RV la escena de rehabilitación y las demás en formato plano para facilitar así la interacción con la interfaz por parte del personal médico.

Este entorno está pensado inicialmente para ser utilizado de manera asistida con el médico junto al afectado ya que el médico deberá definir la terapia concreta a realizar. La terapia consiste en una serie de niveles que deberá completar el paciente: cada nivel consiste en tocar un botón en el entorno virtual a cierta distancia del usuario, es decir un esfuerzo mayor o menor, y un número de repeticiones que se representan como puntos. Para otorgar mayor libertad al médico de organizar la rehabilitación como considere conveniente se ha aplicado una abstracción que le permite crear todos los niveles que quiera independientemente del número de escenas creadas. Esto se consigue gracias a que el cambio de un nivel a otro viene representado por el desplazamiento del botón en el entorno, lo cual se ha desarrollado para generarlo de manera dinámica recargando siempre la misma escena y así crear un número de niveles tan grande como se quiera.

Gracias a la abstracción mencionada y tras realizar varias revisiones y depurar ideas el flujo de ejecución de la aplicación que ilustra la interacción entre escenas se presenta en la Figura 2.10 es el siguiente, siendo “Rehabilitación” la escena que se cargará tantas veces se requiera.

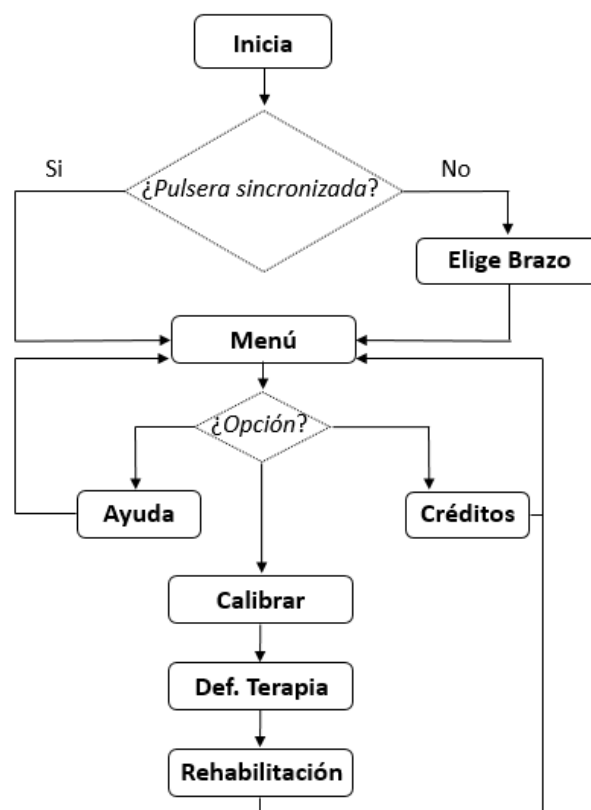


Figura 2.10: Diagrama que representa la lógica de la aplicación

2.2.1.1 Escena 1: Inicio

Consiste en la escena inicial del juego. Es una interfaz de tipo GUI que contiene un mensaje de bienvenida y pide el nombre del paciente para continuar. En esta pantalla se inicializan las variables globales creando la instancia, que será única gracias al patrón de diseño “Singleton”, de la clase “Manager”. De igual manera, se inicializa la pulsera a utilizar en la aplicación. Para ello se crea en la jerarquía de la escena el prefab, objeto predefinido, "Hub - 1 Myo" que contiene un hijo "Myo" que representa a la pulsera. De este modo, se obtienen los primeros datos, como su estado actual, que permite conocer, en caso de estar sincronizada correctamente en el paciente [28], el brazo en que se está utilizando. En el supuesto de obtener un valor válido provocará un salto a la escena del Menú principal sin pasar por la escena de Elección brazo.

2.2.1.2 Escena 2: Elección brazo

Tras pasar por la escena inicial, si la pulsera no está correctamente sincronizada es imposible saber qué brazo se pretende rehabilitar. Para ello, en esta escena, se pide de manera muy intuitiva seleccionar sobre qué extremidad superior se debe efectuar la terapia.

2.2.1.3 Escena 3: Menú principal

Esta escena representa el menú principal de la aplicación, en la cual se puede decidir si empezar la terapia de rehabilitación, visualizar una pantalla de ayuda, los créditos del sistema o salir.

Al igual que las dos anteriores es una interfaz gráfica de usuario con botones para seleccionar la opción deseada.

2.2.1.4 Escena 4: Calibrado

Al pulsar en Iniciar en el Menú principal pasamos a la escena de Calibrado. El propósito de esta escena es medir el alcance máximo del brazo del paciente para que, de esta manera, el terapeuta pueda pedirle esfuerzos relativos a su máximo y tenerlo en cuenta al generar la dificultad de la terapia.

El aspecto de la interfaz es sencillo para hacerlo intuitivo al paciente, al que se pedirá que estire el brazo lo máximo que pueda, con la pulsera puesta correctamente, un total de 5 veces para así obtener la media de los alcances. Una vez completado el calibrado, la distancia máxima se habrá guardado en una variable global de la instancia “Manager” y se pasará a la siguiente escena donde el personal médico definirá la terapia a realizar.

2.2.1.5 Escena 5: Definir Terapia

Tras obtener en la escena anterior el alcance máximo que tiene el usuario se pasa a una escena de tipo GUI igualmente en la que el clínico ajustará ciertos parámetros para realizar la rehabilitación.

Lo que se persigue conseguir mediante esta terapia fijada por el clínico es mejorar el alcance del miembro superior afectado. Para ello, se le pedirá que realice una actividad con una mayor o menor dificultad, que es la que se define en este punto de la aplicación.

El objetivo es estirar el brazo a cierta distancia cada vez. Para realizar esta rehabilitación de manera más dinámica y divertida, y que el paciente se motive, está estructurada como un videojuego por niveles, de forma que hay que completar todos para terminar la terapia. Cada nivel tiene una dificultad y un número de puntos necesarios para superarlo, de modo que estos dos parámetros son los que definirá el médico en esta escena.

La dificultad, en porcentaje, se define como el esfuerzo necesario que debe realizar el paciente para alcanzar el objetivo. Lo ideal al realizar esta rehabilitación sería que el afectado consiguiese un alcance de su brazo igual al de una persona sana. Por tanto, para obtener una escala de esfuerzo se fija a la distancia máxima obtenida en la escena anterior de calibrado como el 100% de alcance. Así, conociendo la distancia de alcance que realiza una persona sana se obtiene un porcentaje a lograr por el paciente (ver Figura 2.11).

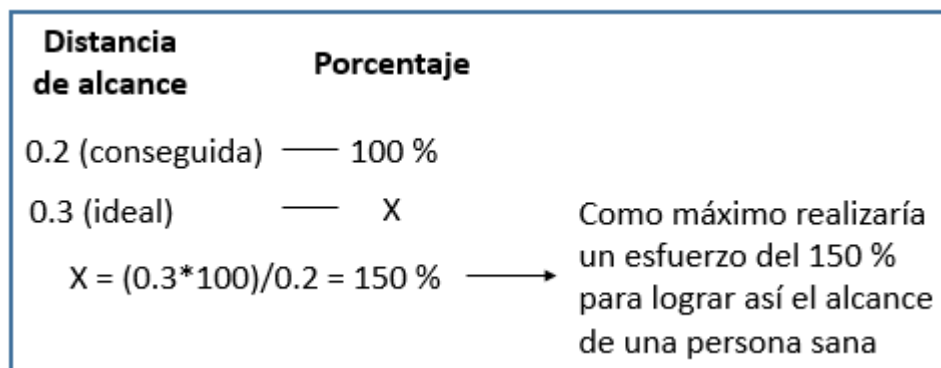


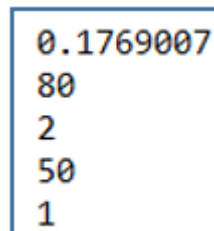
Figura 2.11: Concepto de Esfuerzo en la aplicación.

Por otro lado se define la cantidad de veces que debe realizar dicho esfuerzo en una escala de 1 a 10, es decir los puntos que se deben conseguir para superar el nivel.

De este modo cada nivel se compone de una dificultad (esfuerzo) y un número de veces (puntos) que se debe realizar el esfuerzo definido.

El clínico tiene dos formas de definir una terapia: creándola partiendo de cero, con la posibilidad de guardarla, o cargando una ya existente. Las terapias se guardan en ficheros con extensión “.txt” en un directorio “Terapias”, que se crea en el momento de guardar una rutina de rehabilitación. El formato con el que se guardan las terapias es simple: en la primera línea

se guarda la distancia de calibrado con la que se realizó dicha terapia. Posteriormente tras insertar un salto de línea se guardan los niveles. Cada nivel se guarda con un esfuerzo y un número de puntos por lo que se almacenan dichos datos. Primero el porcentaje de esfuerzo como un entero y, tras un salto de línea, el número de puntos como otro entero. En caso de contar con más de un nivel se continúa con el siguiente par de enteros (ver Figura 2.12).



```
0.1769007
80
2
50
1
```

Figura 2.12: Aspecto fichero de terapia. Se visualiza primero el calibrado y luego dos niveles, el primero con 80% y 2 puntos y el segundo al 50% y 1 punto.

El motivo principal de registrar el calibrado con el que se guardó la terapia existente es para no sobre esforzar al paciente. En el ejemplo de la Figura 2.11 el paciente podía como máximo realizar un sobre esfuerzo del 150% para así alcanzar la distancia de una persona sana. Sin embargo, si un paciente con una distancia conseguida en el calibrado igual a 0.25 (en vez de 0.2 como el anterior), pedirle un esfuerzo del 150% es ilógico ya que será superior a la distancia ideal (0.3) de una persona sana y, por tanto, inalcanzable. Por ello, para adaptar una terapia a cada paciente, se tiene en cuenta su distancia máxima de alcance y así se consigue no sobre esforzar a ningún usuario.

Esta posibilidad de cargar una rutina ya creada previamente es muy interesante ya que permitiría a un médico guardar determinadas terapias, que sean especiales por su duración o esfuerzo, y así aplicarlas a distintos pacientes dentro de las posibilidades de cada uno.

Volviendo a la escena en cuestión, cada vez que se define una dupla esfuerzo-puntos se agrega como un nivel en una ventana a la derecha de la interfaz. En dicha ventana se visualizan numerados los niveles a realizar, ya sean recién creados o una terapia cargada. En caso de querer modificar la terapia es posible eliminar los niveles ya definidos y, de igual manera, se puede guardar la nueva terapia que se esté definiendo que tendrá el formato antes comentado.

Entrando en un apartado más técnico de implementación, la aplicación controla los niveles con una lista bidimensional definida en la clase “Manager”, por lo que al crear la instancia se crea el listado vacío que en esta escena se rellena. La lista se podría visualizar como se muestra en la Figura 2.13, en la que cada posición representa un nivel y está compuesta por otra lista de dos posiciones que contienen el porcentaje de esfuerzo y los puntos a alcanzar:

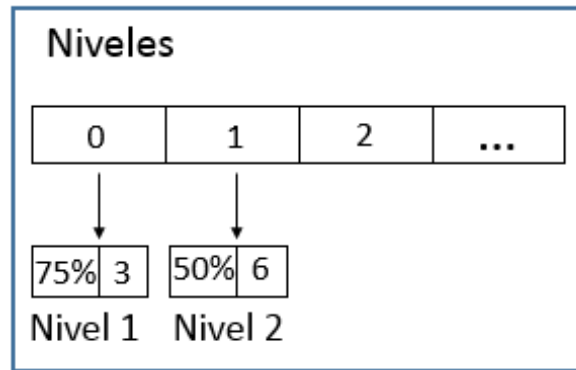


Figura 2.13: Lista bidimensional que contiene los niveles definidos.

2.2.1.6 Escena 6: Rehabilitación

Tras haber definido la terapia a realizar, se pasaría a la escena más importante del juego, la encargada de llevar a cabo la rehabilitación mediante RV.

La dinámica del juego consiste en superar los niveles definidos de manera sucesiva hasta completar todos. El objetivo de esta rehabilitación es, como ya se ha dicho, mejorar el alcance del brazo del paciente, por lo que se busca simular un entorno que le obligue a estirar el brazo. La manera en que se ha implementado permite realizar la terapia de manera satisfactoria. El objetivo es un botón o pulsador que debe ser tocado para obtener un punto. La dificultad radica en lo lejos que esté el pulsador del paciente y la cantidad de veces que deba hacerlo.

El entorno básico consiste en un cuerpo humano creado con el software MakeHuman e importado en Unity y una cámara que está ubicada de manera que, al visualizar en RV, simula el propio cuerpo del usuario. Este personaje se encuentra sentado en una silla y con los brazos formando un ángulo aproximado de 90 grados sobre una mesa. En dicha mesa se encuentra el botón, que se acercará o alejará del paciente según la dificultad de cada nivel, y que al ser pulsado emitirá un sonido a modo de realimentación auditiva aparte del sensorial emitido por la vibración de la pulsera.

Para amenizar la terapia y convertirla en una experiencia realista y divertida en cada nivel se creó un añadido que consiste en cambiar la ambientación 3D del entorno. Esto incluye paisajes de tipo boscoso o un entorno de playa con grandes gráficos con animaciones y sonidos de calidad. Tras completar un nivel se pasará al siguiente, que como se ha comentado será la misma escena, y con dicho cambio se cambiará igualmente el entorno gráfico, lo que le permitirá al paciente divertirse y estar motivado para ver más niveles añadiendo un factor motivacional importante a la actividad de rehabilitación.

Para lograr el efecto de cambio de ambientación, a pesar de ser la misma escena la que se carga sucesivamente, se crea un objeto llamado “GeneradorEscenarios”. Este objeto vacío cuenta con un script que se encarga de realizar dicha función. Para ello contiene un *array* de terrenos en

el que cada posición está ocupada por un terreno en 3D y se instancia de manera aleatoria en cada nivel (teniendo en cuenta el del nivel anterior para no repetir).

Al empezar esta escena el paciente visualizará, con ayuda de las gafas de RV, el entorno definido y un mensaje de información donde se le explica cómo empezar a “jugar”, recentrar la vista de la cámara de RV o la opción de ocultar dicho mensaje sin empezar la terapia para que pueda observar con detenimiento el ambiente. Tras empezar un nuevo nivel, se le mostrará nuevamente el mensaje y, por último, al completar todos los niveles volverá automáticamente al menú principal.

El desarrollo de esta escena fue el más complejo, como cabía esperar. Primero, como ya se comentó al comienzo de este apartado 2.2.1, se tomó la decisión de que para simular tantos niveles como el médico definiese, sólo se tendría una escena, que se cargaría tantas veces como fuese necesaria únicamente cambiando su aspecto gráfico y la dificultad para aparentar ser otra distinta. Con esta abstracción se otorga libertad al médico y se gana en eficiencia de la aplicación.

Al no contar con diseñadores gráficos, el apartado de modelar y diseñar el entorno en 3D se llevó a cabo con ayuda de paquetes disponibles, todos ellos gratuitos, en el Asset Store de Unity. Contar con este recurso fue clave para obtener entornos realistas con buenos gráficos para la RV. Cada ambiente se crea como un objeto de tipo terreno en Unity y, tras modelarlo y agregarle los componentes necesarios como sonido ambiente u objetos extra animados, se agrega al *array* del objeto “GeneradorEscenarios” que más tarde es la fuente de donde se obtiene de manera aleatoria el ambiente de cada nivel.

Por último, otros aspectos destacados de esta escena son la forma en la que se reproduce un alcance con el brazo en la RV y la recogida de datos que se realiza cuando el paciente lleva a cabo la rehabilitación. Estos aspectos se detallan a continuación.

2.2.1.6.1 Reproducción del alcance del brazo en la RV

Uno de los problemas que se presentaba era determinar el modo de reflejar el movimiento de alcance del brazo en la RV, dado que solamente se cuenta con un sensor inercial colocado en el antebrazo del paciente. El hecho de utilizar MakeHuman para crear la persona, posibilitaba que ésta tuviera un esqueleto de forma muy similar a un ser humano real, con lo cual era posible contar con un hueso para el antebrazo y otro hueso para el brazo. El humano creado se exportó a Unity, desde donde se permitían realizar rotaciones en las articulaciones del codo y hombro. Por ello, se estimó oportuno imitar la forma en la que se hace un alcance real, esto es, una rotación positiva en el eje X para el hombro y una rotación negativa en el mismo eje para el codo. Para hacer el movimiento de retroceso se harían rotaciones igualmente en las mismas articulaciones y los mismos ejes, pero en sentido contrario.

La aceleración generada por el Myo Armband en el eje Z que se produce al realizar un alcance se traduce, mediante leyes físicas, en espacio recorrido en este mismo eje. Al contar únicamente con un acelerómetro situado en el antebrazo, se tenía que traducir del alguna forma dicha

distancia en rotaciones en el codo y hombro de manera que se simule el alcance, como se ha comentado anteriormente. Por ello, se realizaron pruebas experimentales con diferentes transformaciones proporcionales del espacio recorrido, como rotaciones en ambas articulaciones, hasta que se halló el valor óptimo de los parámetros tanto para el hombro (espacio recorrido en el eje Z x 200) como para el codo (espacio recorrido en el eje Z x15), los cuales reflejan fidedignamente el alcance.

De cómo se captura la aceleración y se transforma en espacio recorrido se habla con más detalle posteriormente (sección 2.2.2 Captura de movimientos).

2.2.1.6.2 Registro de actividad

Como se mencionó en la introducción, utilizar la RV para neurorrehabilitar a pacientes con DCA posibilita obtener de alguna manera sus progresos. Por lo tanto, además de su función principal de rehabilitar, la RV también podría ser usada como herramienta de diagnóstico, como se ha hecho en este proyecto.

Esto se implementa recogiendo los datos de aceleración del paciente mediante el script "Thalmyo", cuando está jugando en su escena de "Rehabilitación", tanto para movimientos de avance como para movimientos de retroceso. En un fichero de tipo ".csv" cuyo nombre tendrá el formato "NombrePaciente Fecha_Acel.csv", se guarda lo siguiente (Figura 2.14): las aceleraciones sobre los tres ejes de coordenadas, si se trata de un movimiento de avance o de retroceso, el nivel en el que se encuentra de los que haya establecido el terapeuta y el esfuerzo en forma de porcentaje relativo al máximo alcance del usuario para ese nivel. Además en una línea a parte se guarda el porcentaje de acierto del paciente para cada nivel. El fichero se guardará en una carpeta llamada "Aceleraciones" en la misma ruta en la que se encuentra el ejecutable del juego.

Aceleración X	Aceleración Y	Aceleración Z	Movimiento	Nivel	Esfuerzo (%)
0.328898	0.1405549	0.08886208	avance	1	80
0.1396058	0.07308733	0.1973568	avance	1	80
-0.0894464	0.0006431341	0.1611775	avance	1	80
-0.219785	-0.0686335	-0.03236796	avance	1	80
-0.177548	-0.1318273	-0.1863757	avance	1	80
-0.1672677	-0.04755366	-0.1981923	avance	1	80
-0.04641894	0.05852139	-0.1508963	avance	1	80
Fin nivel 1. Puntos a alcanzar : 1 - Intentos necesarios: 1 -> Porcentaje de acierto: 100%					
0.2956095	0.1341451	0.07890819	avance	2	95
0.2899423	0.09558177	0.2704895	avance	2	95
-0.1531175	0.09815288	0.2074668	avance	2	95
-0.3227775	-0.09039021	-0.01539575	avance	2	95
-0.3047226	-0.1264631	-0.2172489	avance	2	95
-0.1441726	-0.05876762	-0.3212845	avance	2	95
-0.01508757	0.0616976	-0.2162071	avance	2	95
-0.1229573	0.1375406	-0.08239959	retroceso	2	95
-0.2995854	0.0867157	-0.3518182	retroceso	2	95
-0.1899361	0.04771054	-0.0938779	retroceso	2	95
-0.04465053	0.01459241	0.0680727	retroceso	2	95
0.2165053	-0.1359311	0.2162227	retroceso	2	95
0.2246694	-0.03297079	0.2636526	retroceso	2	95
0.3204854	0.1294534	0.03041831	avance	2	95
0.4451746	0.04786873	0.2084658	avance	2	95
-0.0182202	0.06514442	0.1993115	avance	2	95
-0.2194286	0.03590226	0.139847	avance	2	95
-0.402043	-0.02672851	-0.02221173	avance	2	95
-0.1685379	-0.2232374	-0.2544309	avance	2	95
-0.1384455	-0.08418757	-0.3221295	avance	2	95
-0.135858	0.1128119	-0.3282853	avance	2	95
Fin nivel 2. Puntos a alcanzar : 2 - Intentos necesarios: 2 -> Porcentaje de acierto: 100%					

Figura 2.14: Datos recogidos en el fichero

De este modo, además de la forma más evidente que ya existía y permitía ver el progreso del paciente, que era simplemente ver cómo éste iba superando niveles, con estos nuevos datos se puede hacer un análisis más profundo acerca de la calidad del movimiento del paciente. El juego trata de alcanzar y retroceder por lo que se deberían producir aceleraciones fundamentalmente en el eje Z, y no en el eje X ya que significaría que se está desviando del objetivo y que aún no controla el movimiento en su totalidad. Este tipo de progresos se pueden apreciar observando los datos de las aceleraciones del fichero. Además, con el porcentaje de acierto de cada nivel, se puede ver si el paciente a medida que va avanzando va teniendo más aciertos y, de esta manera, conocer si progresa favorablemente y saber cuál es la terapia óptima para su recuperación.

Como posible ampliación a futuro, esta idea se puede trasladar al ámbito de la telemedicina, es decir, que el paciente pueda rehabilitarse desde su casa y se envíen los datos al terapeuta remotamente para que lleve un control.

2.2.1.7 *Escena 7: Ayuda*

La escena de Ayuda es accesible desde el menú principal y se visualiza como una interfaz sencilla con un mensaje de ayuda en el que se le explica al usuario el funcionamiento básico de la aplicación y resuelve posibles dudas que puedan surgir.

2.2.1.8 *Escena 8: Créditos*

Por último, la escena de créditos que, al igual que la de ayuda sólo es accesible desde el menú, se visualiza como una interfaz en la que aparecen los créditos desplazándose de abajo hacia arriba con un movimiento suave. Aquí se incluyen los créditos tanto a los desarrolladores, los impulsores y a los autores de las obras con licencia utilizadas en el proyecto.

2.2.2 **Captura de movimientos**

Como se ha mencionado anteriormente, en un inicio, este entorno ha sido pensado para neurorrehabilitar a pacientes con daño cerebral en una tarea de alcance con el brazo. Por ello, el paciente, en su escena del juego "Rehabilitación", tendrá como meta estirar el brazo para llegar a pulsar un botón. Para obtener la distancia de desplazamiento, el paciente llevará puesto en el brazo afectado el *movement tracker* Myo Armband, que recogerá los datos de aceleración en tiempo real. Desde el script "Thalamic Myo" se tomarán los datos relativos a la aceleración en el eje Z para hallar el espacio recorrido en este eje. El script "Controlador Brazo" utilizará esta distancia para reflejar el movimiento del brazo real en el brazo de la RV pero, en este caso, de manera ideal, por las razones dadas en la sección 1.6.1 relativas al *embodiment* y las neuronas espejo.

En lo referente a reflejar movimientos reales del brazo en la RV con datos de la aceleración tomados del *tracker* Myo Armband no ha resultado un asunto banal. El hecho de que el dispositivo emita la información a una frecuencia de 50Hz presenta un problema a la hora de simular un movimiento continuo. Por ejemplo, para reflejar con exactitud en la RV un simple estiramiento del brazo hacia adelante, se necesitaría recibir, para mostrar que el brazo queda parado al final, exactamente la misma aceleración positiva que negativa. Esto no ocurre así debido a la frecuencia con la que trabaja el Myo Armband. Por lo tanto, lo que sucede es que la aceleración y la deceleración no se compensan y, como consecuencia, el brazo de la RV seguiría un movimiento hacia adelante o hacia atrás, cuando el brazo real ya se encuentra en reposo.

En la gráfica de la Figura 2.15 se puede apreciar el problema. Se observa como en un movimiento de alcance se produce primero una aceleración y tras ello una deceleración hasta que la aceleración es cero. En ese momento el brazo está parado, pero sigue habiendo una velocidad constante positiva y, por lo tanto, se sigue avanzando en el espacio. Esto es provocado porque la aceleración no se ha compensado totalmente con la deceleración debido a la obtención de datos discretos por la frecuencia del Myo.

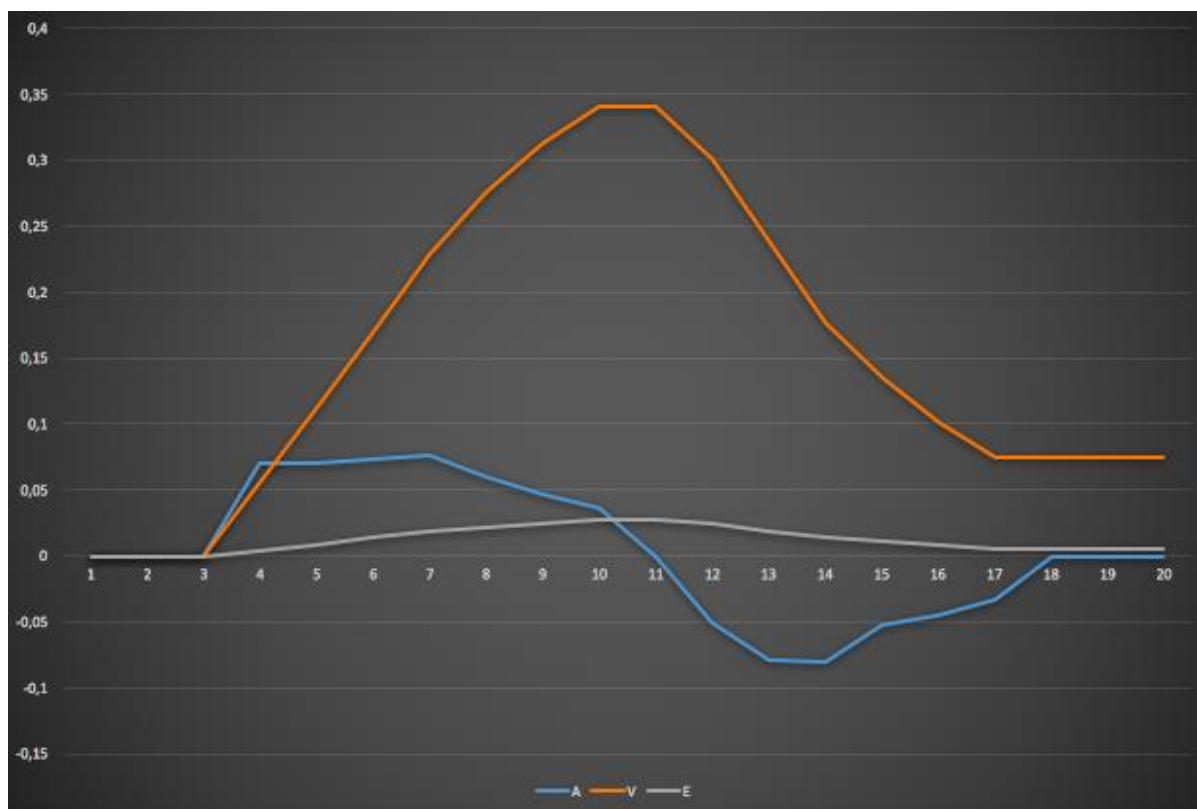


Figura 2.15: Se aprecia la problemática: cuando la aceleración es cero como resultado de que se ha parado, se observa que el espacio sigue levemente por encima de cero y por lo tanto se sigue desplazando. Aceleración en azul, velocidad en naranja y espacio en gris.

Como solución se ha procedido a transformar de alguna manera los datos de aceleración así como la dinámica del juego para adaptarlo a las limitaciones impuestas por esta tecnología. Se ha ido tomando la aceleración como la media de cinco en cinco aceleraciones (filtro de media acumulada). De ese modo, se consigue suavizar los valores discretos y así compensar algo mejor la aceleración positiva con la negativa, aunque no totalmente. Por ello, se ha acotado la libertad de mover hacia adelante y hacia atrás el brazo, cuando el paciente quiera, para hacer más abordable el problema. Se ha decidido dividir el movimiento que puede realizar el paciente con el brazo en dos fases síncronas: avance y retroceso, reiniciando al final de ambos esfuerzos la aceleración acumulada. En el movimiento de avance el paciente deberá estirar el brazo para llegar a presionar el botón y en el movimiento de retroceso, el paciente volverá a poner el brazo en la posición inicial. Antes de realizar cualquiera de los dos movimientos se mostrará por pantalla una cuenta atrás en la que se le indicará al paciente si tiene que avanzar o retroceder al término de ésta. Con todo esto, se consiguen paliar los inconvenientes anteriormente mencionados.

Se ha dicho que algunas de las ventajas del Myo Armband es que es un dispositivo comercial no excesivamente caro y por lo tanto accesible, pero al no estar dirigido específicamente a rehabilitación no queda exento de algún inconveniente como el que se acaba de describir.

Capítulo 3. Resultados

El resultado final de la conjunción de las metodologías antes mencionadas, junto con las tecnologías empleadas da a lugar a la aplicación desarrollada: un entorno de RV controlado en tiempo real por señales motoras para realizar terapias de neurorrehabilitación a pacientes con problemas de movilidad en los miembros superiores. Tras plantear varias opciones el nombre elegido para el sistema es “RehaV”, haciendo un juego de palabras con rehabilitación o *rehab* en inglés junto con RV.

Para ejecutar de manera correcta RehaV y sus periféricos son necesarios ciertos requisitos del sistema. La pulsera Myo Armband no exige un gran rendimiento al igual que los juegos en Unity, que prometen ser ejecutados en casi cualquier dispositivo. Sin embargo las gafas de RV si necesitan un rendimiento superior por lo que se tendrán en cuenta sus requisitos [29]:

- NVIDIA GTX 970 / AMD 290 equivalente o superior
- Intel i5-4590 equivalente o superior
- 8GB+ RAM
- Puerto HDMI 1.3 de salida de video
- 2 puertos USB 3.0 y 1 puerto USB 2.0
- Windows 7 SP1 64 bit o superior, más la actualización de plataforma DirectX

RehaV consiste, como ya se mencionó anteriormente en un total de 8 escenas, 7 de ellas accesibles sin necesidad de dispositivos de RV y 1 que representa la rehabilitación en la que sí es necesario contar con unas gafas de RV.

La escena inicial se presenta, como ya se comentó en la sección 2.2.1.1, como una pantalla sencilla, en la que se debe introducir el nombre del paciente que realizará la terapia (Figura 3.1).



Figura 3.1: Escena Inicio

Tras avanzar se pasará en caso de no estar sincronizada la pulsera a la escena de elección de miembro superior a rehabilitar (Figura 3.2).

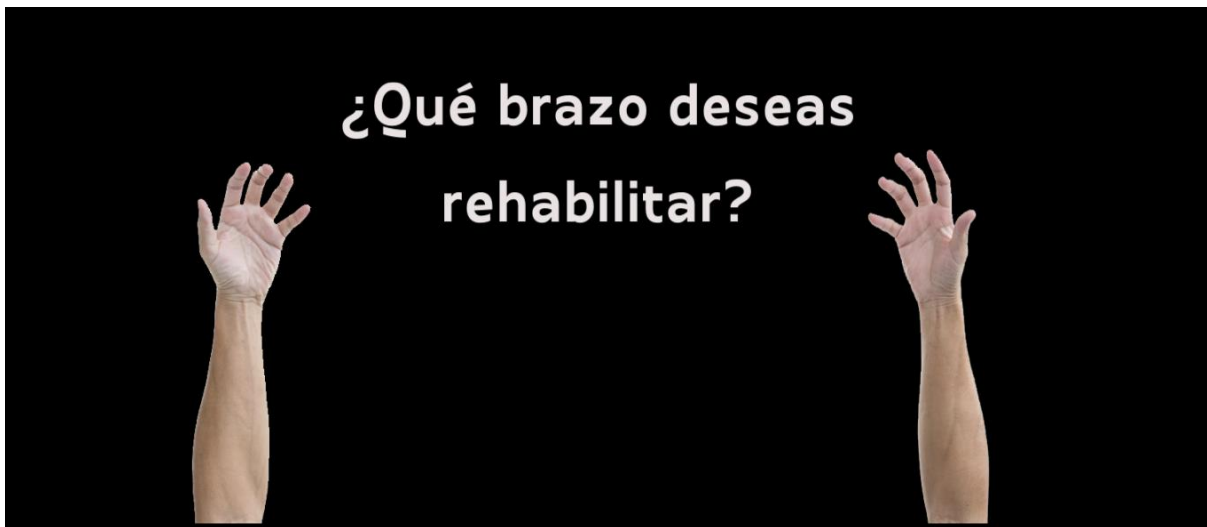


Figura 3.2: Escena Elige Brazo

En cambio, en caso de tener el brazalete correctamente sincronizado, el sistema ya interpreta que brazo es el designado para realizar la terapia por lo que se carga la escena correspondiente al Menú, donde se visualizan botones que permiten navegar por la aplicación y el botón para salir de la misma (Figura 3.3).

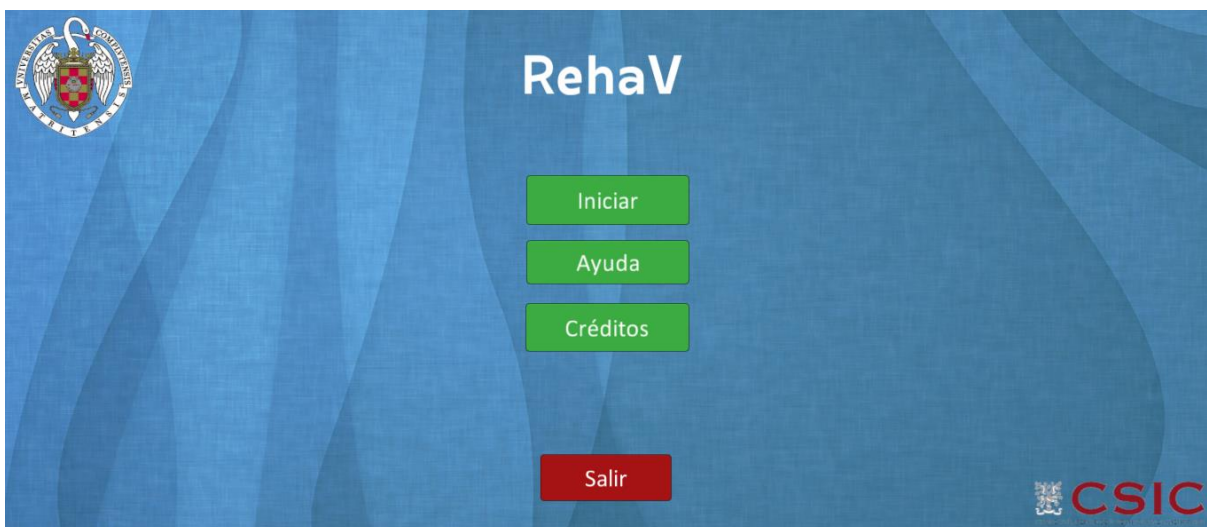


Figura 3.3: Escena Menú Principal

Al pulsar en Ayuda se accede a la escena con el mismo nombre, en la que se visualiza un mensaje de ayuda para orientar al usuario de la aplicación (Figura 3.4).

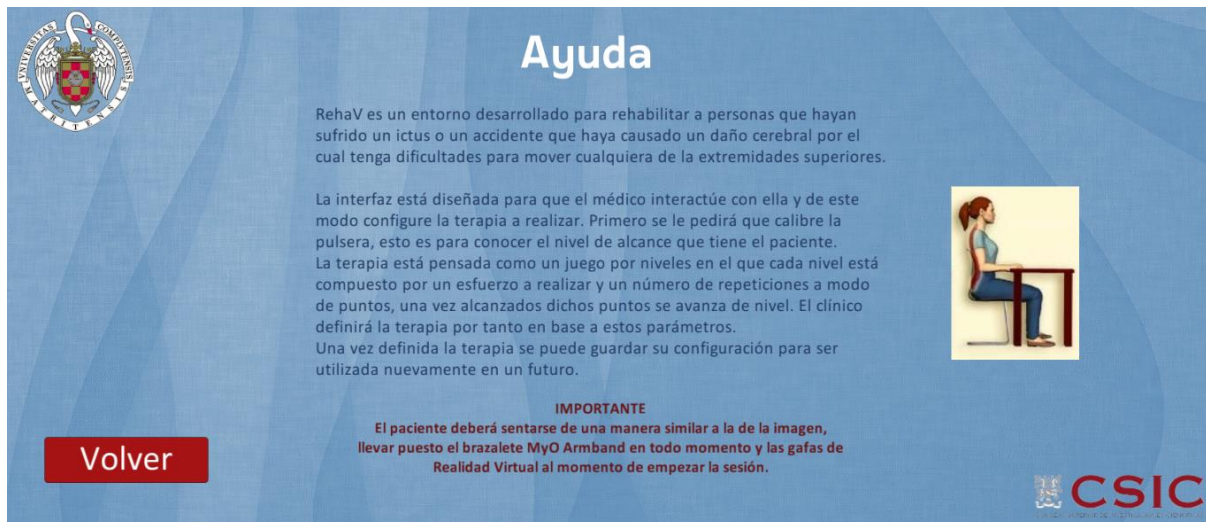


Figura 3.4: Escena Ayuda

Si se vuelve al Menú y se pulsa el botón de Créditos se visualizan los créditos de la aplicación y el proyecto, incluyendo desarrolladores o materiales, con un diseño similar al planteamiento inicial que se comentó en la sección de metodologías (Figura 3.5).



Figura 3.5: Escena Créditos

Volviendo al Menú, si se pulsa en Iniciar se accede a la pantalla de calibrado, en la que se solicitará al usuario que realice el movimiento de alcance, hasta lo máximo que pueda, un total de 5 veces para generar un registro con su capacidad (Figura 3.6).



Figura 3.6: Escena Calibrado

Tras superar la escena de Calibrado, se pasa a la escena de definición de terapia (Figura 3.7) donde se puede apreciar que es posible cargar una terapia existente o guardar la actual que se crea de manera intuitiva, con ayuda de los dos *sliders* (diales) y el botón de añadir central.



Figura 3.7: Escena Define Terapia

Por último, está la escena principal del juego, la que permite realizar la rehabilitación mediante RV. En ella, como ya se ha comentado, el objetivo es alcanzar un botón un número determinado de veces y así ir avanzando de nivel. Al iniciar la escena, el aspecto que tiene en la pantalla del ordenador es el mostrado en la Figura 3.8 (el ambiente se genera mediante un algoritmo aleatorio por lo que puede variar cada vez).

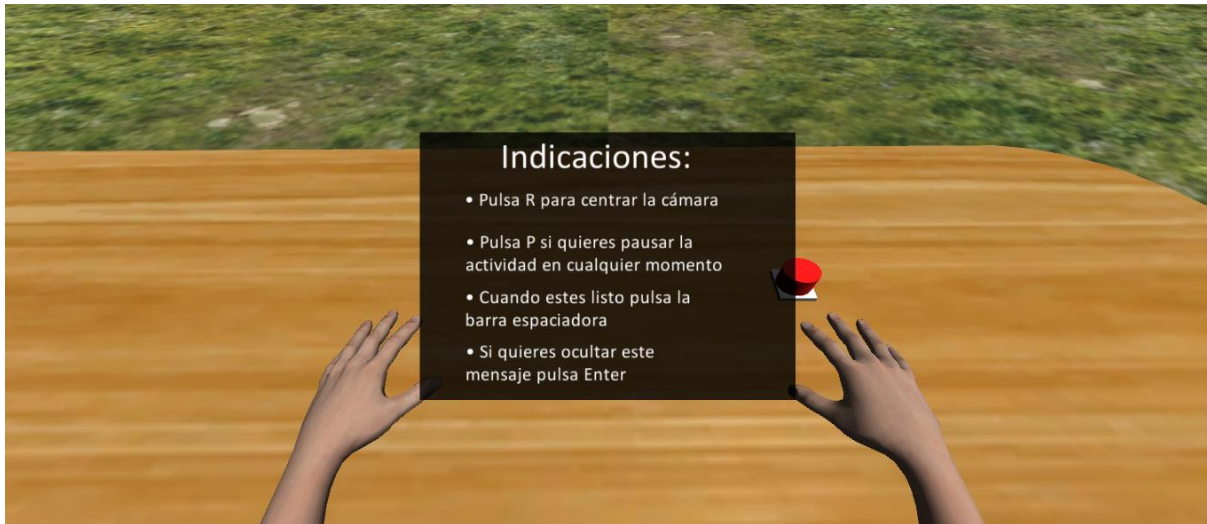


Figura 3.8: Escena Rehabilitación

El paciente ve los mismos elementos pero a través de las gafas de RV en modo bifocal, activándose así la sensación “inmersiva”. Aparece un mensaje inicial de indicaciones que puede ocultar para observar el entorno antes de empezar.

Al empezar se le pedirá que estire el brazo hacia adelante mediante una cuenta regresiva (Figura 3.9).



Figura 3.9: Escena Rehabilitación II

Se visualiza un campo de texto que le indica cuándo debe empezar el movimiento de alcance hacia adelante y además se perfila ya el marcador de puntuación, que aquí se superpone por encima de la cuenta regresiva porque debe estar centrado en la pantalla para visualizarlo mediante RV (en la vista bifocal esta todo debidamente ordenado).

Tras realizar el movimiento hacia adelante se le pedirá, en caso de no haber completado el nivel, que realice el movimiento hacia atrás de manera análoga (Figura 3.10).



Figura 3.10: Escena Rehabilitación III

Por último, en caso de completar el nivel, se mostrará un mensaje que le informa del fin del nivel unido a la realimentación que siempre emite el botón, tanto de sonido como de vibración de la pulsera (Figura 3.11).



Figura 3.11: Escena Rehabilitación IV

Tal y como se ha mencionado anteriormente, en cada nivel, además de modificar la distancia del botón, se cuenta con un algoritmo que instancia escenarios diferentes de manera aleatoria,

lo que añade un factor sorpresa para el paciente. Se han implementado un total de 5 ambientaciones distintas (véase Figuras 3.12 y 3.13), que se visualizarán a medida que se avanza de nivel. Esta dinámica está ideada para potenciar la motivación del paciente y hacer la experiencia de rehabilitación más divertida.



Figura 3.12: Ejemplo Escenario I



Figura 3.13: Ejemplo Escenario II

Al completarse el desarrollo de la aplicación se realizaron diversas evaluaciones por expertos en la materia. Personal del CSIC (Consejo Superior de Investigaciones Científicas) e investigadores expertos en temas de neurorrehabilitación y nuevas tecnologías se mostraron

satisfechos con el resultado obtenido e hipotetizan que será una herramienta muy útil y potencialmente eficaz para pacientes neurológicos afectados por problemas motores en los miembros superiores.

Su valoración concluye que existen suficientes evidencias que argumentan que la RV, tal y como se aplica en este trabajo, ayudará a acelerar el proceso de recuperación por sus propiedades, que inciden en la reorganización neuronal, y recalcaron que el método empleado es bastante eficiente y correcto gracias a los procedimientos empleados en el desarrollo del proyecto.

De igual manera, el CSIC ha conseguido pactar la realización de unas pruebas con el prototipo desarrollado con personas afectadas en el hospital Beata María Ana de Madrid.

Sin embargo, para obtener una primera impresión más inmediata se han realizado pruebas con usuarios no afectados por problemas motores y así apreciar su comportamiento (ver Figura 3.14 y 3.15). Los resultados fueron positivos, los usuarios afirmaron haber sentido estar dentro del entorno y su ambientación, y de igual manera encontraron entretenido el juego en cuestión. Estos resultados son representativos ya que, a pesar de no ser el público objetivo al no contar con problemas motores, permiten verificar la sensación de inmersión de la RV con la vital importancia que tiene esto para el futuro del proyecto y la neurorrehabilitación.



Figura 3.14: Pruebas con usuarios no afectados



Figura 3.15: Pruebas con usuarios no afectados II

Además, se ha podido apreciar la capacidad adaptativa de los usuarios al juego, es decir, a medida que iban avanzando en los niveles definidos para la terapia, conseguían mayor porcentaje de acierto al pulsar el botón (ver Figura 3.16). De este modo, la facilidad de uso del sistema permite que el usuario se adapte rápidamente a él, ajustándose más al movimiento de alcance que se ve en la RV. Esto también supone un factor importante y a tener en cuenta en la recuperación de pacientes afectados.



Figura 3.16: Adaptabilidad de paciente no afectado. Datos obtenidos del fichero que registra la actividad del paciente en la escena "Rehabilitación"

De igual manera, se ha analizado la variación de aceleración en el eje Z para cada nivel (Figura 3.17). Por ello para el nivel más fácil, en el cual el botón está más cerca, la aceleración imprimida por el usuario es menor que en niveles superiores en los que el botón se encuentra más alejado. Gracias a esto se puede intuir la adaptación del movimiento por parte del paciente y aún más importante, la eficacia de la sensación de inmersión en la propiocepción y *embodiment* al realizar el ejercicio.

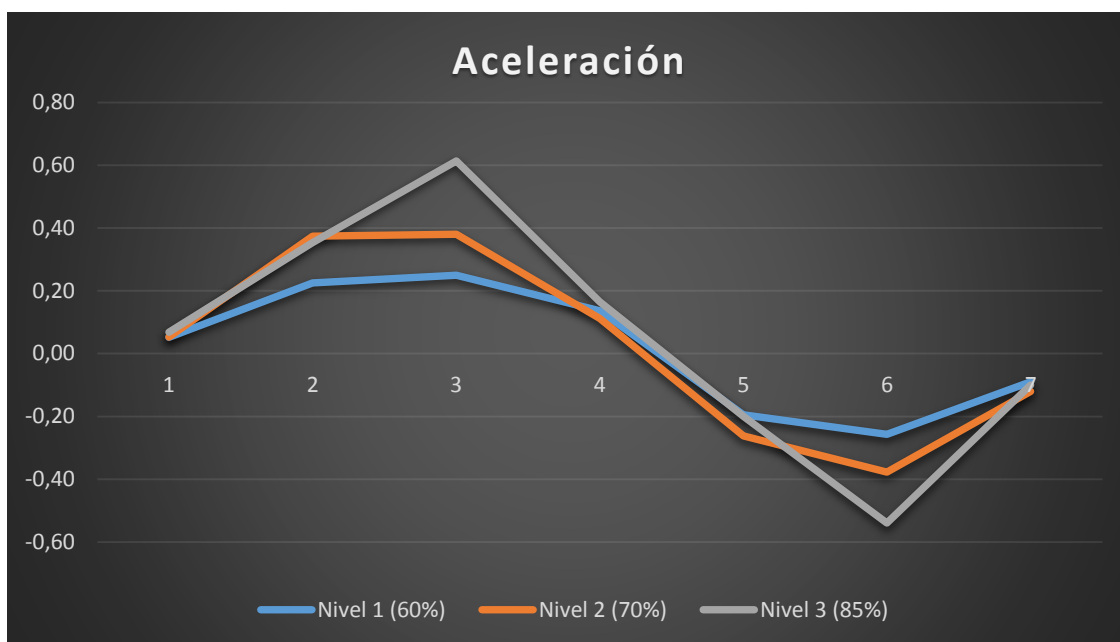


Figura 3.17: Variación de aceleración según el esfuerzo requerido. Datos obtenidos del fichero que registra la actividad del paciente en la escena "Rehabilitación"

Capítulo 4. Conclusiones

La recuperación motora del paciente está asociada a la reorganización cortical, es decir, tras el DCA se activará la zona ipsilateral y a medida que el paciente se vaya recuperando tenderá a activarse la zona contralateral, alrededor de la zona dañada, que es la que de forma natural tenía el control. Lo ideal es empezar la terapia en la fase aguda que es cuando surge la denominada plasticidad espontánea del cerebro, en la que se activan estructuras de aprendizaje que sólo lo hacen durante la edad temprana de crecimiento.

El objetivo general de este proyecto era el desarrollo de un sistema que permitiera la neurorrehabilitación del miembro superior en pacientes con DCA influyendo en la plasticidad cerebral. Se puede afirmar que dicho objetivo se ha cumplido, como se muestra en el capítulo de resultados. Además, se incide en los fenómenos de *embodiment* y neuronas espejo en la escena de rehabilitación, lo que podría acelerar la recuperación, llevando a cabo con éxito la parte más novedosa de este proyecto.

Del mismo modo, se han cumplido los objetivos específicos planteados. Se trata de un sistema de rehabilitación mediante juego, lo cual potencia el entretenimiento y la motivación, como se aprecia en la escena de rehabilitación, en la que se ha de superar un número determinado de niveles. Permite la configuración de terapias específicas dependiendo del esfuerzo gracias a la escena en la que se define la terapia, lo cual hace que se pueda adaptar a las necesidades y posibilidades de los distintos pacientes. Se trata de un entorno intuitivo tanto para el terapeuta como para el paciente, como por ejemplo muestran las instrucciones de ayuda de la escena de rehabilitación. Se realiza un registro de actividad durante la misma escena, tanto de las aceleraciones producidas por el movimiento del brazo, como del porcentaje de acierto, y se guarda en un fichero que podrá consultar el terapeuta para hacer un seguimiento del progreso y evolución del paciente. Además, no es necesario utilizar dispositivos específicos del campo de la medicina, con el coste que ello conlleva. De igual manera, RehaV se ejecuta en ordenadores personales con requisitos de rendimiento asumibles.

4.1 Trabajo Futuro

Este proyecto tiene un potencial inmediato en el campo de la telemedicina ya que, como se ha visto, permite registrar datos del paciente a medida que éste realiza la terapia y estos datos se le pueden enviar al terapeuta de forma remota para que lleve un seguimiento de su progreso. Además, la creciente comercialización de dispositivos que capturan el movimiento y de dispositivos de RV en el mundo de los videojuegos potencia su distribución entre pacientes, al ser productos más asequibles y más baratos que los dedicados específicamente a rehabilitación y a la vez más sencillos de utilizar al estar familiarizados con ellos. Por ejemplo, se podría desarrollar la aplicación para algún *smartphone* que admita RV y así el paciente pueda desarrollar la terapia desde su propio domicilio, con todos los beneficios que ello conlleva. El paciente puede tener problemas de movilidad, debido a los déficits generados por el daño

cerebral, por lo que realizar la recuperación desde casa puede suponerle una ayuda importante. Otra ventaja sería que puede dedicar más tiempo a realizar de manera intensiva la terapia, lo que podría acortar los plazos de recuperación.

En cuanto a las tecnologías utilizadas, el Myo Armband demuestra ser una herramienta con potencial en terapias de rehabilitación mediante RV debido a sus virtudes como su precio y multitud de funciones. Abarcando esas funciones se podría plantear un trabajo futuro en el que, además de rehabilitar el alcance del brazo, se tratase también mediante terapia con RV el agarre, es decir, la acción de coger objetos, ya que gracias a los sensores EMG del brazalete se pueden obtener valores que permitan simular la acción de agarre y aplicar el mismo principio que en este proyecto.

También cabe decir que este trabajo se centra en la neurorrehabilitación del miembro superior, pero se podrían emplear las mismas tecnologías en la recuperación de la extremidad inferior. Se decidió que el enfoque fuera sobre las extremidades superiores porque son las que más les cuesta rehabilitar a los pacientes, además de la importancia que tienen en las actividades de la vida cotidiana. Sin embargo, cuando un paciente sufre daño cerebral, es posible que repercuta en varias partes del lado contralateral al daño en el cuerpo, tanto la extremidad superior como la inferior. Es por ello que se podrían neurorrehabilitar de igual manera las piernas del paciente colocando, por ejemplo, el Myo Armband en su tobillo y utilizando técnicas de realimentación visual positiva, como las que se usan en este proyecto, que incidan en los mismos fenómenos.

Por otra parte, la terapia de RV desarrollada en este proyecto puede ir de la mano de las terapias tradicionales y ambas se pueden complementar. En cuanto a la efectividad de esta vertiente de terapias mediante RV es necesario seguir realizando trabajos de investigación en los que se demuestre de manera fehaciente, mediante fMRI (imagen por resonancia magnética funcional) y otras técnicas, si este tipo de terapias son realmente más efectivas que las tradicionales por sí solas. De acuerdo a las investigaciones realizadas hasta ahora, esta es una hipótesis plausible, y, por lo tanto, probarla otorgaría una mayor consideración a las terapias basadas en RV.

Por todo ello, se concluye que el presente trabajo de fin de grado ha cumplido con creces los objetivos planteados, tanto por los desarrolladores como por el CSIC. Su hipótesis de partida y su naturaleza metodológica le confieren un excelente presente y un futuro muy prometedor.

4.2 Conclusions

The recovery of the patient is associated with the cortical reorganization. After suffering the DCA, the ipsilateral area will be activated and as the patient is recovering, the contralateral area around the affected area tends to activate too, which is the one that originally had the control. The best results come when the therapy started during the acute phase when the spontaneous brain plasticity emerge, in that case is where the learning structures became active again (which are only activated during the growth young age).

The main goal of this project was the development of a system which allows the neurorehabilitation of the upper limb in patients who has suffer a DCA influencing on brain plasticity. It can be affirmed that this main objective has been achieved, as it can be seen in the results chapter. Besides, its effects on the embodiment and mirror neurons phenomena in the rehabilitation scene, could accelerate the recovery and at the end successfully carrying out the newest part of this project.

In the same way, the specific goals have been achieved with excellent results. It is a rehabilitation system based on game concepts, which enhances the entertainment and motivation, as seen in the rehabilitation scene, which has to overcome a certain number of levels. It allows the configuration and personalization of therapies according to each patient and their capabilities. Also it is developed as an intuitive platform even for the patient and the medical staff, such as it can be seen in the instructions and help panel on each level. Additionally, an activity record is saved for each patient which includes the acceleration raw data produced by the arm movement and the success ratio. These data are written in a file which can be consulted any time by the therapist to track the progress and evolution of the patient. Another important aspect to have in consideration is that the system is developed as a low cost system. The devices needed are cheaper than the traditional related with the field of medicine and in the same way, RehaV, can be executed in personal computers with assumable performance requirements.

Capítulo 5. Distribución de trabajo

En el presente TFG la distribución de trabajo por parte de los miembros del equipo de desarrollo ha sido bastante equitativa.

En una primera instancia fue necesario realizar una investigación conjunta sobre el campo de trabajo. Éste comprende el uso de la informática y nuevas tecnologías aplicadas a la medicina moderna, que al ser un tema totalmente novedoso para los desarrolladores, implicó un estudio del estado del arte actual.

Debido a las características del trabajo y los dispositivos necesarios para su desarrollo, al contar con recursos limitados, es decir, con únicamente un brazalete Myo Armband y unas gafas de realidad virtual Oculus Rift DK2 se antojaba difícil el desarrollo en paralelo. Por ello se adoptó la política de establecer reuniones de manera frecuente (normalmente cada dos días), en las que se trabajaba de manera intensiva en tareas conjuntas que permitieran la integración en el sistema de los dispositivos mencionados. Estas reuniones tuvieron lugar en la Facultad de Informática de la Universidad Complutense de Madrid y en el Centro de Automática y Robótica (CAR) del CSIC cuando se necesitaba la orientación y validación de los progresos por parte de los investigadores expertos en el área de la neurorrehabilitación.

El objetivo general del proyecto era el desarrollo de un sistema para la neurorrehabilitación de del alcance de la extremidad superior en pacientes con daño cerebral. Para ello, la intención era mostrar mediante la realidad virtual un movimiento mejorado respecto del que el paciente realmente hace, con el fin de incidir en los fenómenos explicados (*embodiment* y neuronas espejo). Inicialmente, se desconocía como se iba a llevar a cabo este objetivo global y abstracto, ya que requería la comprensión profunda del dispositivo Myo Armband y sus posibilidades. Uno de los aspectos más relevantes del proyecto y a la vez con más complicación, ha sido todo lo relacionado con la captura de movimientos, debido a los problemas que ha supuesto reflejar el movimiento del brazo en la realidad virtual con la baja frecuencia de muestreo de aceleraciones del brazalete. Por ello, si no se conseguía implementar esto de manera correcta, siendo la esencia del proyecto, no tenía sentido continuar con el resto del desarrollo, ya que, como después se ha visto, las limitaciones impuestas por este dispositivo han determinado la dinámica del juego. Este es otro de los motivos por los cuales ha sido necesario trabajar de manera conjunta, debido a la complejidad que imponía el uso del Myo Armband.

Tras conseguir traducir correctamente las aceleraciones registradas por el Myo Armband en desplazamiento (mediante leyes físicas) sobre un objeto vacío en Unity 3D, se procedió a traducir esta distancia en un movimiento de alcance en el humano articulado creado con MakeHuman. Para ello se simuló el alcance del brazo de una persona real con rotaciones en hombro y codo. Al igual que en el apartado anterior, el hecho de reflejar correctamente el movimiento de alcance con precisión fue una tarea compleja que requirió horas de esfuerzo conjuntas entre los dos desarrolladores.

Una vez conseguido el efecto de simular el movimiento de alcance realizado por el usuario, se procedió a desarrollar el flujo de ejecución de escenas mediante Unity. Tras un breve periodo inicial de aprendizaje en Unity ambos desarrolladores contaban con los conocimientos necesarios para abordar el juego, por lo que se pudo realizar cierto reparto de tareas pero siempre atado a la limitación de contar con un único brazalete y gafas de RV. Dichas tareas se caracterizaban por tener elementos comunes como los controladores de escena, encargados de gestionar la entrada del usuario y la lógica de la escena en cuestión, siendo muy similares entre sí. Para integrar todo esto en el sistema con los dispositivos utilizados y comprobar su correcto funcionamiento era necesario seguir practicando la política de reuniones constantes.

Por último, en la fase de pruebas del sistema desarrollado ambos componentes del grupo se encargaron de pactar y organizar dichas evaluaciones con usuarios sin problemas motores, estando presentes durante las mismas.

Apéndice

En el presente apéndice se adjuntan fragmentos representativos del código en C# de la aplicación RehaV.

En primer lugar se puede ver el aspecto general de la clase “Manager”, una de las más importantes ya que permite controlar variables únicas gracias al patrón de diseño Singleton.

```

/// <summary>
/// Clase Manager que representa la clase basada en el patron singleton para controlar las variables globales del juego
/// Se definen todas las variables globales aqui junto con métodos getter y setter para cada una
/// </summary>
public class Manager : Singleton<Manager> {
    protected Manager() {}

    //Instancia de la clase
    private static Manager instance;
    //Atributos de la clase que representan las variables globales
    private static float distanciaMax;
    private static float maximoNormal;
    private static string ultimaEscena;
    private static Thalmic.Myo.Arm brazoActual;
    private static int ultimoTerreno;
    private static List<List<int>> arrayNiveles;
    private static int nivelActual;
    private static string nombrePaciente;
    private static int numIntentos;

    /*----Ficheros-----*/
    private static StreamWriter writeFile;
    /*-----*/

    /// <summary>
    /// Método que se encarga de crear la instancia Manager con valores por defecto si no existe o devolver la ya existente
    /// </summary>
    /// <value>The instance.</value>
    public static Manager Instance {
        get {
            if (instance == null) {
                GameObject variablesGlobalesObject = new GameObject ("VariablesGlobalesObject");

                DontDestroyOnLoad (variablesGlobalesObject);
                ultimaEscena = SceneManager.GetActiveScene ().name;
                distanciaMax = 0.3f; //Del calibrado de la persona
                maximoNormal = 0.3f; //Máximo de una persona sana
                brazoActual = Thalmic.Myo.Arm.Right;
                ultimoTerreno = -1;
                arrayNiveles = new List<List<int>> ();
                nivelActual = 0;
                nombrePaciente = "";
                numIntentos = 0;
                writeFile = null;
                instance = variablesGlobalesObject.AddComponent<Manager> ();
                Debug.Log ("Se ha creado una instancia de Variables Globales.");
            }
            return instance;
        }
    }
}

```

Tras el método anterior que devuelve la instancia, la existente o si es la primera vez la crea, se encuentran diversos métodos para acceder a las variables y modificar su valor.

A continuación se aprecia un fragmento del controlador de la escena “Inicio”, todos los controladores de escena tienen un aspecto y funcionalidad similar dependiendo de cada escena y su interfaz.

```

/// <summary>
/// Clase del controlador de la escena Inicio del juego
/// En ella se inicializan multitud de variables y se piden datos del paciente
/// </summary>
public class ControladorEscenaInicio : MonoBehaviour {

    private GameObject myo;
    public InputField user;
    public Button aceptar;
    public AudioSource click;
    private bool aceptaActivado= false;
    //private List<char> caracteresProhibidos = {'/',':','<','>', '?', '*', '|', ''}; No puedo ha
cerlo así

    /// <summary>
    /// Método que se ejecuta primero que todos, controlamos que no se active la vista en RV
    /// </summary>
    void Awake(){
        UnityEngine.VR.VRSettings.enabled = false;
        UnityEngine.VR.VRSettings.loadedDevice = UnityEngine.VR.VRDeviceType.None;
    }

    /// <summary>
    /// Se activa el objeto de la clase Manager por lo que se crea la instancia que será única
    en toda la ejecución
    /// </summary>
    void Start () {
        Manager.Instance.gameObject.SetActive(true);
        myo = GameObject.Find ("Myo");
        Manager.Instance.setUltimaEscena (SceneManager.GetActiveScene ().name);
    }

    //En el método Update se realizan comprobaciones para habilitar o deshabilitar botones en
    base a la entrada del usuario
    void Update(){
        if (!acceptaActivado && user.text.Length > 0 && (!user.text.Contains("/") && !user.text
.Contains("<") && !user.text.Contains(">") && !user.text.Contains("?") && !user.text.Contains(
"**) && !user.text.Contains(":") && !user.text.Contains("|")) ) {
            aceptar.enabled = true;
            aceptaActivado = true;
        }
        if ((acceptaActivado && user.text.Length == 0) || (user.text.Contains("/") || user.text
.Contains("<") || user.text.Contains(">") || user.text.Contains("?") || user.text.Contains("**
") || user.text.Contains(":") || user.text.Contains("|"))) {
            aceptar.enabled = false;
            aceptaActivado = false;
        }

        if(user.text.Length > 0 && Input.GetKeyDown (KeyCode.Return) && !user.text.Contains("/
")){
            OnClick();
        }
    }
}

```

Se observa que desde este Script se gestiona la entrada del usuario y el control de errores. El método inicial *Awake()* se encarga de gestionar el tipo de interfaz ya sea en RV o plana. La llamada al método *OnClick()* indica que el usuario ha pulsado el botón y la aplicación reacciona de cierta manera a esa interacción.

Las demás escenas de tipo GUI cuentan con controladores similares al anterior por lo que no se adjuntan en esta memoria.

Otro código interesante es el encargado de definir la terapia. Se realizan escrituras en ficheros, actualizaciones en la lista de niveles entre otros. Las variables de tipo público se encuentran así definidas para facilitar su asignación en el editor de Unity.

```

/// <summary>
/// Clase ControladorEscenaDefine encargada de gestionar la escena donde se define la rehabilitación a realizar
/// La escena de DefineTerapia se genera de manera dinámica ya que los valores son dependientes del paciente concreto
/// Cuenta con dos sliders uno para el esfuerzo que se genera de manera dinámica teniendo en cuenta el esfuerzo de una persona sana
/// y el segundo slider es fijo de con el número de puntos a alcanzar.
/// Luego cuenta con un botón para agregar ese nivel al array de niveles que se visualiza a la derecha de la pantalla en un scroll view
/// Desde este código se gestionan todos los botones de la interfaz incluyendo todo lo relacionado con guardado y cargado de terapias
/// </summary>
public class ControladorEscenaDefine : MonoBehaviour {

    public AudioSource click;
    private GameObject audioFondo;
    private GameObject myo;
    public Slider porcentaje;
    public Slider puntacion;
    public Text puntosText;
    public Text porcenText;
    public Button iniciar;
    private bool aceptarActivado = false;

    private int i=0;
    public RectTransform content;
    public Text datos;
    public Text nivel;
    private int distanciaH=0;

    public Dropdown dropfile;
    public InputField inputText;
    public Button guardar;
    private bool guardarActivado = false;
    public Button borrar;
    private bool borrarActivado = false;
    public Button cargar;
    private bool cargarActivado = false;

    /// <summary>
    /// Método que se ejecuta primero que todos, controlamos que no se active la vista en RV
    /// </summary>
    void Awake () {
        UnityEngine.VR.VRSettings.enabled = false;
        UnityEngine.VR.VRSettings.loadedDevice = UnityEngine.VR.VRDeviceType.None;
    }

    void Start () {
        myo = GameObject.Find ("Myo");
        audioFondo = GameObject.Find ("AudioFondo");
        Debug.Log (Manager.Instance.getDistanciaMax());
        Manager.Instance.setUltimaEscena (SceneManager.GetActiveScene ().name);
        Debug.Log (Manager.Instance.getUltimaEscena());

        //Añade listener al slider
        porcentaje.onValueChanged.AddListener (delegate {
            actualizaSliderPorc (porcentaje.value);
        });

        //El porcentaje va en base a su calibrado
        porcentaje.maxValue = (Manager.Instance.getMaximoNormal () / Manager.Instance.getDistanciaMax ()) * 100f;

        //Añade listener al slider
        puntacion.onValueChanged.AddListener (delegate {

```

```

        actualizaSliderPunt (puntacion.value);
    });

    //Creamos el directorio para guardar las terapias si no existia
    string path = @"./Terapias/";
    if(!Directory.Exists(path))
    {
        Directory.CreateDirectory(path);
    }

    //Carga en la interfaz las terapias anteriores disponibles
    actualizaFicheros ();
}

//En el método Update se realizan comprobaciones para habilitar o deshabilitar botones en
base a la entrada del usuario
void Update() {

    if (!guardarActivado && Manager.Instance.getArrayNiveles ().Count > 0 && inputText.tex
t.Length > 0 && (!inputText.text.Contains("/") && !inputText.text.Contains("<") && !inputText.
text.Contains(">") && !inputText.text.Contains("?") && !inputText.text.Contains("*") && !input
Text.text.Contains(":") && !inputText.text.Contains("|")) {
        guardar.enabled = true;
        guardarActivado = true;
    }
    if (guardarActivado && inputText.text.Length == 0 || (inputText.text.Contains("/") ||
inputText.text.Contains("<") || inputText.text.Contains(">") || inputText.text.Contains("?") |
| inputText.text.Contains("*") || inputText.text.Contains(":") || inputText.text.Contains("|"
)) {
        guardar.enabled = false;
        guardarActivado = false;
    }

    if(!aceptarActivado && Manager.Instance.getArrayNiveles().Count>0){ //Activamos el bot
on de iniciar si cargamos algun nivel
        iniciar.enabled = true;
        aceptarActivado = true;
    }

    if(aceptarActivado && Manager.Instance.getArrayNiveles().Count == 0){ //Activamos el b
oton de iniciar si cargamos algun nivel
        iniciar.enabled = false;
        aceptarActivado = false;
    }

    if (!borrarActivado && Manager.Instance.getArrayNiveles ().Count > 0) {
        borrar.enabled = true;
        borrarActivado = true;
    }

    if (borrarActivado && Manager.Instance.getArrayNiveles ().Count == 0) {
        borrar.enabled = false;
        borrarActivado = false;
    }

    if (!cargarActivado && dropfile.value >= 0) {
        cargar.enabled = true;
        cargarActivado = true;
    }

}

/// <summary>
///Método encargado de cargar una terapia ya existente.
/// Se borra el array actual de la terapia y se lee del fichero seleccionado, se calculan l
os porcentajes adecuados
/// y se empieza a llenar el array de niveles uno a uno hasta llegar al final y se llama al
método de pintar en pantalla
/// para mostrarlos en el scroll view
/// Por último se cierra el descriptor de fichero
/// </summary>

```

```

public void OnClickCargar(){
    OnClickBorrar ();
    StreamReader loadFile;
    loadFile = File.OpenText("./Terapias/" + dropfile.captionText.text);
    float calibradoAnterior = float.Parse(loadFile.ReadLine());
    int porcentajeMaxAnterior = (int)((Manager.Instance.getMaximoNormal () / calibradoAnterior)*100f);
    int porcentajeMaximoActual = (int)((Manager.Instance.getMaximoNormal () / Manager.Instance.getDistanciaMax ())*100f);
    while (!loadFile.EndOfStream){
        //cada dos lineas es un elem del Array, el 0 y el 1 de cada index

        string linePorc = loadFile.ReadLine();
        string linePunt = loadFile.ReadLine ();
        List<int> a = new List<int> ();
        a.Add ((Int32.Parse(linePorc)*porcentajeMaximoActual)/porcentajeMaxAnterior); //porcentaje normalizado para el paciente actual
        //Aunque basado en la terapia del txt con un calibrado actual
        a.Add (Int32.Parse(linePunt));
        Manager.Instance.getArrayNiveles ().Add (a);
        pintaEnPantalla ();
    }

    loadFile.Close ();
}

/// <summary>
/// Método del botón de añadir una dupla esfuerzo-puntos a la terapia actual
/// Agrega al array de niveles un nuevo elemento por tanto y posteriormente llama a pintaEnPantalla
Pantalla
/// </summary>
public void OnClickAñadir(){
    click.Play ();
    List<int> a = new List<int> ();
    a.Add ((int)porcentaje.value);
    a.Add ((int)puntacion.value);
    Manager.Instance.getArrayNiveles ().Add (a);
    pintaEnPantalla ();
}

/// <summary>
/// Método del botón de borrar la terapia definida actualmente
/// Elimina todos los elementos del array y actualiza la interfaz
/// </summary>
public void OnClickBorrar(){
    click.Play ();
    Manager.Instance.getArrayNiveles().Clear();
    int Nhijos = content.transform.childCount;
    Debug.Log (Nhijos);
    for (int j = 0; j < Nhijos; j++) {
        Destroy (content.GetChild (j).gameObject);
    }
    i = 0;
    distanciaH = 0;
}

/// <summary>
/// Método encargado de actualizar el listado de niveles actuales de la terapia
/// Lo hace instanciando un prefab predefinido con un formato el cual edita para insertar el detalle de cada nivel.
/// Tras pintarlo actualiza la distancia entre texto y texto y el número de niveles i
/// </summary>
private void pintaEnPantalla(){
    Text childObjectNivel = Instantiate(nivel, new Vector3(nivel.transform.position.x,nivel.transform.position.y-distanciaH,nivel.transform.position.z), nivel.transform.rotation) as Text;
    Text childObjectText = Instantiate(datos, new Vector3(datos.transform.position.x,datos.transform.position.y-distanciaH,datos.transform.position.z), datos.transform.rotation) as Text;
    childObjectNivel.text = (i + 1).ToString () + " ";
    childObjectText.text = Manager.Instance.getArrayNiveles () [i] [0].ToString () + "% - " + Manager.Instance.getArrayNiveles () [i] [1].ToString () + " puntos";
    childObjectNivel.transform.SetParent(content.transform,false);
}

```

```

        childObjectText.transform.SetParent (content.transform, false);

        content.offsetMin = new Vector2 (content.offsetMin.x, datos.transform.position.y-
distanciaH );
        distanciaH = distanciaH + 49;
        i++;
    }

```

Se aprecian los métodos encargados tanto de cargar una terapia ya existente, guardar es análogo pero escribiendo en fichero y no en el listado, agregar niveles y el método encargado de mostrar en pantalla de manera correcta el nivel recién creado.

El siguiente Script se denomina “ArmController” ya que se encarga de realizar el movimiento del brazo, ya sea derecho o izquierdo, según lo realice el paciente.

```

/// <summary>
///Clase encargada de controlar el movimiento del brazo del jugador.
///Se asigna el brazo y antebrazo a rehabilitar al upper arm y fore arm correspondiente.
///A partir de ahí se tiene un ángulo para cada uno. Aplicando rotaciones con angulos de euler
///se consigue el efecto del brazo moviendose hacia adelante y hacia atrás.
/// </summary>
public class ArmController : MonoBehaviour {

    private GameObject myo = null;
    public GameObject upper = null;
    public GameObject fore = null;

    private float xAngleUpper = 0.0f;
    private float xAngleFore = 0.0f;
    private float xAngleIniUpper, yAngleIniUpper, zAngleIniUpper, xAngleIniFore, yAngleIniFore
, zAngleIniFore;

    private Quaternion rotationUpper, rotationFore;

    private const float max = 0.3f;
    private float razon;

    void Start () {
        myo = GameObject.Find ("Myo");

        //Asignamos los UPPER y FORE segun el brazo que rehabilitará
        if (Manager.Instance.getBrazoActual () == Thalmic.Myo.Arm.Right) {
            upper = GameObject.Find ("RUpper");
            fore = GameObject.Find ("RFore");
        } else if (Manager.Instance.getBrazoActual () == Thalmic.Myo.Arm.Left) {
            upper = GameObject.Find ("LUpper");
            fore = GameObject.Find ("LFore");
        }

        //razon = Manager.Instance.getDistanciaMax() / 0.1f;
        razon = 1.0f;
        //razon = Manager.Instance.getDistanciaNivelActual ();
        //upper
        xAngleIniUpper = upper.transform.rotation.eulerAngles.x;
        yAngleIniUpper = upper.transform.rotation.eulerAngles.y;
        zAngleIniUpper = upper.transform.rotation.eulerAngles.z;

        //fore
        xAngleIniFore = fore.transform.rotation.eulerAngles.x;
        yAngleIniFore = fore.transform.rotation.eulerAngles.y;
        zAngleIniFore = fore.transform.rotation.eulerAngles.z;
    }

    void FixedUpdate () {
        xAngleUpper = xAngleIniUpper - ((razon * myo.transform.position.z) * 200.0f);
        xAngleFore = xAngleIniFore + ((razon * myo.transform.position.z) * 15.0f);

        //xAngleUpper = xAngleIniUpper - (myo.transform.position.z * 200.0f);

```

```

rotationUpper.eulerAngles = new Vector3 (xAngleUpper, yAngleIniUpper, zAngleIniUpper);
upper.transform.rotation = rotationUpper;

//xAngleFore = xAngleIniFore + (myo.transform.position.z * 15.0f);
rotationFore.eulerAngles = new Vector3 (xAngleFore, yAngleIniFore, zAngleIniFore);
fore.transform.rotation = rotationFore;
}
}

```

Se pueden apreciar las rotaciones en el brazo y antebrazo del ser humano del juego para simular el movimiento de alcance deseado.

Por último se adjunta un fragmento del código central de la aplicación, el encargado de registrar datos de la pulsera Myo Armband y transformarlos en movimiento para la rehabilitación. Es el código más complejo y largo de RehaV, por la aplicación de leyes físicas y por la necesidad de aplicar diversas medidas para paliar las consecuencias derivadas de tratar con un producto low cost.

```

//Guardamos la posición inicial para cuando se realice el movimiento de vuelta se pueda volver a la posición inicial y no
//se acumule error. Esta variable es utilizada por el script ControladorEscenaBrazos.
void Start() {
    posIni = transform.position;
}

void Update() {
    lock (_lock) {
        armSynced = _myoArmSynced;
        arm = _myoArm;
        xDirection = _myoXDirection;
        if (_myoQuaternion != null) {
            transform.localRotation = new Quaternion(_myoQuaternion.Y, _myoQuaternion.Z, -_myoQuaternion.X, -_myoQuaternion.W);
        }
        if (_myoGyroscope != null) {
            gyroscope = new Vector3(_myoGyroscope.Y, _myoGyroscope.Z, -_myoGyroscope.X);
        }
        pose = _myoPose;
        unlocked = _myoUnlocked;
    }
}

//Método que se ejecuta continuamente con el que se trata la aceleración del Myo Armband
void FixedUpdate() {
    lock (_lock) {
        if (_myoAccelerometer != null) {
            //Se recibe la aceleración de en sus tres componentes
            accelerometer = new Vector3 (_myoAccelerometer.Y, _myoAccelerometer.Z, -_myoAccelerometer.X);

            //Se acumula el tiempo de cada FixedUpdate (20ms) para utilizarlo en las formulas de velocidad y espacio
            t += Time.deltaTime;

            //Transformamos las coordenadas locales de la aceleración del Myo Armband en globales
            globalAccel = transform.TransformVector (accelerometer);
        }
    }

    //Control del movimiento de alcance
    if (inicia) {
        //Los primeros datos que se reciben de aceleración al activar el script están alterados por lo
        //que se retiene antes de empezar a tomar datos
        if (retiene < 5) {
            retiene++;
        }
    }
}

```

```

    } else {
        acumAcel += globalAcel.z;
        contadorAcel++;
    }
    //Se toma aceleraciones de 5 en 5 para hacer una media y que de esa manera se suavice por lo problemas
    //que hay con el muestreo de baja frecuencia a 50Hz
    if (contadorAcel == Nelem) {
        azmedia = acumAcel / (float)Nelem;
        //En el caso de que la aceleración este al revés para controlar que el sentido sea el correcto
        if ((Mathf.Abs (azmedia) > 0.03f) && !signoFijado) {
            if ((azmedia < -0.03f)) {
                Debug.Log ("CambiaSigno");
                cambiaSigno = true;
            }
            signoFijado = true;
        }

        if (cambiaSigno) {
            azmedia = -azmedia;
        }

        //Umbral de 0.03 debido a la sensibilidad del Myo Armband. Si se sobrepasa el umbral quiere decir que
        //se ha empezado a relizar el movimiento, con lo cual, pasa a tener en cuenta todas las aceleraciones
        if ((azmedia > 0.03f) || empieza) {
            empieza = true;
            if (Manager.Instance.getDescriptor () != null && Manager.Instance.getUltimaEscena () == "Box On A Stick") {
                Manager.Instance.getDescriptor ().Write (globalAcel.x + " " + globalAcel.y + " " + azmedia + " " + "avance" + " " + (Manager.Instance.getNivelActual () + 1) + " " + Manager.Instance.getPorcentajeNivelActual () + System.Environment.NewLine);
            }

            signoFijado = true;

            //Para saber cuando ha acelerado y decelerado
            if (azmedia > 0.0f && !acelera) {
                acelera = true;
            } else if (azmedia < 0.0f && !decelera) {
                decelera = true;
            }

            //Cálculo de la velocidad
            vz0 = vz0 + (azmedia * 9.8f) * t;

            //Control para averiguar cuando ha parado y pasar al movimiento en el otro sentido
            if ((acelera && decelera) && ((vz0 < 0.0f) || (azmedia >= 0.0f))) {
                acelera = false;
                decelera = false;
                empieza = false;
                vuelve = true;
                inicia = false;
                Reset ();
                if (Manager.Instance.getUltimaEscena () == "Box On A Stick") {
                    Manager.Instance.incrementaNumIntentos ();
                }
            }

            //Cálculo del espacio
            ez = (vz0 * t);

            //limitamos el maximo por su hay algún fallo y se continúa avanzando
            if ((transform.position.z + ez) > 0.4f) {
                acelera = false;
                decelera = false;
                empieza = false;
                vuelve = true;
                inicia = false;
                Reset ();
                if (Manager.Instance.getUltimaEscena () == "Box On A Stick") {
                    Manager.Instance.incrementaNumIntentos ();
                }
            }
            transform.position = new Vector3 (transform.position.x, transform.posi

```

```

tion.y, 0.4f);
        } else {
            transform.position = new Vector3 (transform.position.x, transform.posi
tion.y, transform.position.z + ez);
        }

    }
    t = 0.0f;
    acumAcel = 0.0f;
    contadorAcel = 0;
    azmedia = 0f;
    ez = 0.0f;
}

}
//Control del movimiento de vuelta
else if (vuelve) {
    //Los primeros datos que se reciben de aceleración al activar el script están alte
rados por lo
    //que se retiene antes de empezar a tomar datos
    if (retiene < 5) {
        retiene++;
    } else {
        acumAcel += globalAccel.z;
        contadorAcel++;
    }
    //Se toma aceleraciones de 5 en 5 para hacer una media y que de esa manera se suav
ice por lo problemas
    //que hay con el muestreo de baja frecuencia a 50Hz
    if (contadorAcel == Nelem) {
        azmedia = acumAcel / (float)Nelem;
        //En el caso de que la aceleración este al revés para controlar que el sentido
sea el correcto
        if (cambiaSigno) {
            azmedia = -azmedia;
        }

        //Umbral de -
0.03 debido a la sensibilidad del Myo Armband. Si se sobrepasa el umbral quiere decir que
//se ha empezado a relizar el movimiento, con lo cual, pasa a tener en cuenta
todas las aceleraciones
        if ((azmedia < -0.03f) || empieza) {
            empieza = true;
            //Registro de las Aceleraciones en el fichero
            if (Manager.Instance.getDescriptor () != null && Manager.Instance.getUltim
aEscena()=="Box On A Stick") {
                Manager.Instance.getDescriptor ().Write (globalAccel.x + " " + globalA
ccel.y + " " + azmedia + " " + "retroceso" + " " + (Manager.Instance.getNivelActual () + 1) +
" " + Manager.Instance.getPorcentajeNivelActual () + System.Environment.NewLine);
            }

            //Para saber cuando ha acelerado y decelerado
            if (azmedia > 0.0f && !acelera) {
                acelera = true;
            } else if (azmedia < 0.0f && !decelera) {
                decelera = true;
            }

            //Cálculo de la velocidad
            vz0 = vz0 + (azmedia * 9.8f) * t;

            //Control para averiguar cuando ha parado y pasar al movimiento en el otro
sentido
            if (((acelera && decelera) && ((vz0 > 0.0f) || (azmedia <= 0.0f)))) {
                ResetMov ();
                Reset ();
                ReseteaSigno ();
            }

            //Cálculo del espacio
            ez = (vz0 * t);

            //limitamos el minimo por su hay algún fallo y se continúa retrocediendo
            if ((transform.position.z + ez) < -0.07f) {

```

```

        ResetMov ();
        Reset ();
        ReseteaSigno ();

        transform.position = new Vector3 (transform.position.x, transform.position.y, -0.07f);
    } else {
        transform.position = new Vector3 (transform.position.x, transform.position.y, transform.position.z + ez);
    }

    }
    t = 0.0f;
    acumAcel = 0.0f;
    contadorAcel = 0;
    azmedia = 0f;
    ez = 0.0f;
}

}

//Para pasar al movimiento de sentido contrario para ambos sentidos
public void Reset (){
    retiene = 0;
    azmedia = 0f;
    acumAcel = 0.0f;
    contadorAcel = 0;
    vz0 = 0f;
    ez = 0f;
    t = 0f;
    accelerometer = new Vector3 (0f, 0f, 0f);
    globalAccel = new Vector3 (0f, 0f, 0f);
}

//Para pasar al movimiento de alcance
public void ResetMov(){
    transform.position = new Vector3 (0f, 0f, 0f);

    acelera = false;
    decelera = false;

    empieza = false;
    inicia = true;
    vuelve = false;
}

//Control para realizar correctamente el movimiento
public void ReseteaSigno(){
    cambiaSigno = false;
    signoFijado = false;
}

```

Destaca el método *FixedUpdate()*, claramente dividido en dos ramas, una para iniciar el movimiento y otra para retroceder. Su extensión se debe a la multitud de cálculos físicos que se realizan cada 20 milisegundos exactos y las medidas de control establecidas para manejar correctamente los datos del Myo. De igual manera se visualiza la escritura del fichero de aceleraciones donde se registra la actividad de cada paciente.

Al final se definen métodos de reinicio de variables denominados *Reset*, en ellos se reestablecen los valores de las variables encargadas de gestionar el movimiento. Se llama a estas funciones para evitar arrastrar errores o datos residuales de cálculos anteriores.

Bibliografía

- [1] Centro Lescer, «Centro Lescer: centro de tratamiento de la lesión cerebral,» [En línea]. Available: <http://www.centrolescer.org/>. [Último acceso: 10 Mayo 2016].
- [2] A. Pollock, S. E. Farmer, M. C. Brady, P. Langhorne, G. E. Mead, J. Mehrholz y F. v. Wijck, «Interventions for improving upper limb function after stroke (Protocol),» *The Cochrane Collaboration*, pp. 1-20, 2013.
- [3] S. H. You, S. H. Jang, Y.-H. Kim, M. Hallett, S. H. Ahn, Y.-H. Kwon, J. H. Kim y M. Y. Lee, «Virtual Reality-Induced Cortical Reorganization and Associated Locomotor Recovery in Chronic Stroke. An Experimenter-Blind Randomized Study,» *Stroke by American Heart Association*, vol. 36, nº 6, pp. 1166-1171, 2005.
- [4] Gonzalo Mendez, Pozo; Facultad de Informática UCM, «PhD Thesis Gonzalo Mendez Pozo,» Septiembre 2008. [En línea]. Available: <http://nil.fdi.ucm.es/sites/default/files/GonzaloMendez-PhDThesis.pdf>. [Último acceso: Mayo 2016].
- [5] S. Viñas-Diz; M. Sobrido-Prieto; Departamento de Fisioterapia, Universidad de A Coruña, «Realidad virtual con fines terapéuticos en pacientes con ictus: revisión sistemática,» *ELSEVIER*, 2015.
- [6] B. Haiwen Chen, M. Jane Epstein y M. Emily Stern, «Neural Plasticity After Acquired Brain Injury: Evidence from Funcitonal Neuroimaging,» *American Academy of Physical Medicine and Rehabilitation*, vol. II, pp. 306-312, 2010.
- [7] MIT; Rizzo, Albert "Skip"; Jounghyun Kim, Gerard, «A SWOT Analysis of the Field of Virtual Reality Rehabilitation and Therapy,» *Presence*, vol. 14, nº 2, 2005.
- [8] Microsoft Corporation, «Kinect: desarrollo de aplicaciones de Windows,» [En línea]. Available: <https://developer.microsoft.com/es-es/windows/kinect>. [Último acceso: 01 Junio 2016].
- [9] Sony Playstation, «PlayStation Eye - Accesorios oficiales de PlayStation,» [En línea]. Available: <http://es.playstation.com/ps3/accessories/detail/item78899/PlayStation%C2%AEEye/>. [Último acceso: 01 Junio 2016].
- [10] Nintendo, «Acciones básicas | Wii | Atención al cliente | Nintendo,» [En línea]. Available: <https://www.nintendo.es/Atencion-al-cliente/Wii/Utilizaci-oacute-n/Mando-de-Wii/Acciones-basicas/Acciones-basicas-243993.html>. [Último acceso: 01 Junio 2016].

- [11] Thalmic Labs, «Present | Myo Gesture Control Armband,» [En línea]. Available: <https://www.myo.com/present>. [Último acceso: 01 Junio 2016].
- [12] B. Lange, S. Koenig, C.-Y. Chang, E. McConnell, E. Suma, M. Bolas y A. Rizzo, «Designing informed game-based rehabilitation tasks leveraging advances in Virtual Reality,» *informahealthcare.com by University of Southern California, Institute for Creative Technologies, Los Angeles, California, USA*, 2012.
- [13] R. Lloréns, C. Colomer-Font, M. Alcañiz y E. Noé-Sebastián, «BioTrak: análisis de efectividad y satisfacción de un sistema de realidad virtual para la rehabilitación del equilibrio en pacientes con daño cerebral,» *ELSEVIER DOYMA*, 2011.
- [14] Alejandro Baldominos; Yago Saez; Cristina García del Pozo; Computer Science and Engineering Dept. Universidad Carlos III, «An approach to physical rehabilitation using state-of-the-art virtual reality and motion tracking technologies,» *ELSEVIER Procedia Computer Science*, 2015.
- [15] Enciclopedia Salud, «Enciclopedia Salud: Definición Sistema Propioceptivo,» Febrero 2016. [En línea]. Available: <http://www.encyclopediasalud.com/definiciones/sistema-propioceptivo>. [Último acceso: Mayo 2016].
- [16] M. Murie-Fernández, P. Irimia, E. Martínez-Vila, M. J. Meyer y R. Teasell, «Neurorehabilitación tras el ictus,» *ELSEVIER DOYMA*, 2009.
- [17] Eucognition, «Tutorial on Embodiment: Eucognition,» [En línea]. Available: <http://www.eucognition.org/index.php?page=tutorial-on-embodiment>. [Último acceso: 14 Mayo 2016].
- [18] L. Sallés, X. Gironés y J. V. Lafuente, «Organización Motora del Córtex Cerebral y el papel del Sistema de las Neuronas Espejo. Repercusiones Clínicas para la Rehabilitación,» *Medicina Clínica*, vol. 144, nº 1, pp. 30-34, 2010.
- [19] Unity Technologies, «Unity - Game engine, tools and multiplatform,» [En línea]. Available: <https://unity3d.com/es/unity>. [Último acceso: 01 Junio 2016].
- [20] Unity Technologies, «Unity Documentation - Monodevelop,» [En línea]. Available: <http://docs.unity3d.com/es/current/Manual/MonoDevelop.html>. [Último acceso: 01 Junio 2016].
- [21] MakeHuman TM, «MakeHuman index,» [En línea]. Available: <http://www.makehuman.org/index.php>. [Último acceso: 22 Mayo 2016].
- [22] Oculus, «Oculus Rift Development Kit 2 (DK2) | Oculus,» [En línea]. Available: <https://www.oculus.com/en-us/dk2/>. [Último acceso: 01 Junio 2016].

- [23] Unity Technologies, «Acerca de nosotros: Unity Technologies,» [En línea]. Available: <https://unity3d.com/es/public-relations>. [Último acceso: Mayo 2016].
- [24] Unity Technologies, «Documentación: Unity,» [En línea]. Available: <http://docs.unity3d.com/es/current/Manual/index.html>. [Último acceso: Mayo 2016].
- [25] Unity Technologies, «Comunidad: Unity,» [En línea]. Available: <https://unity3d.com/es/community>. [Último acceso: Mayo 2016].
- [26] BiggShark, «BiggShark,» [En línea]. Available: <http://biggshark.com/why-using-c-with-unity-is-better-than-boo-and-js-for-your-next-mobile-game/>. [Último acceso: Mayo 2016].
- [27] L. Ulanoff, «Mashable,» Julio 2014. [En línea]. Available: <http://mashable.com/2014/07/29/myo-motion-control-armband-hands-on/#GB7pnVa5ViqZ>. [Último acceso: 2016 Mayo].
- [28] Thalmic Labs, «Myo Support: how to perform the sync gesture,» [En línea]. Available: <https://support.getmyo.com/hc/en-us/articles/200755509-How-to-perform-the-sync-gesture>. [Último acceso: 10 Mayo 2016].
- [29] Oculus, «Documentation - Recommended Specifications : Oculus,» [En línea]. Available: <https://product-guides.oculus.com/en-us/documentation/dk2/latest/>. [Último acceso: 01 Junio 2016].