



UNIVERSIDAD DE VALLADOLID

E.T.S.I. TELECOMUNICACIÓN

TRABAJO FIN DE GRADO

GRADO EN INGENIERÍA DE TECNOLOGÍAS
ESPECÍFICAS DE TELECOMUNICACIÓN, MENCIÓN EN
TELEMÁTICA

**Juego serio para valoración de movimiento y
esfuerzo muscular con motor de juegos Unity3D
y sensores vestibles**

Autor:

D. Javier Cantalapiedra Artiaga

Tutor:

Dr. D. Mario Martínez Zarzuela

Valladolid, 4 de Julio de 2022

TÍTULO	Juego serio para valoración de movimiento y esfuerzo muscular con motor de juegos Unity3D y sensores vestibles
AUTOR:	D. Javier Cantalapiedra Artiaga
TUTOR:	Dr. D. Mario Martínez Zarzuela
DEPARTAMENTO:	Teoría de la Señal y Comunicaciones e Ingeniería Telemática

TRIBUNAL

PRESIDENTE:	Carlos Gómez Peña
SECRETARIO:	Mario Martínez Zarzuela
VOCAL:	David González Ortega
SUPLENTE 1:	Jesús Poza Crespo
SUPLENTE 2:	Miriam Antón Rodríguez

FECHA:	4 de Julio de 2022
CALIFICACIÓN:	

Agradecimientos

Le doy mis agradecimientos al Dr. D. Mario Martínez Zarzuela por pensar en mí para la realización de este Trabajo de Fin de Grado. Además, su ayuda ha resultado vital para la finalización del proyecto y la realización de las posteriores pruebas.

A todas las personas y compañeros que han aportado ideas y me han ayudado a mejorar la jugabilidad y utilidad del juego desarrollado para este Trabajo de Fin de Grado.

A mi familia y amigos por haberme llevado hasta aquí y haber supuesto un apoyo en los momentos más duros y difíciles de esta etapa.

Resumen

La rehabilitación siempre ha sido un proceso costoso para el ser humano. Para poder hacer de ella un proceso más liviano, y ayudar al paciente con una terapia complementaria a la habitual, se ha desarrollado un juego serio que utiliza la electromiografía para medir la fuerza muscular, como interfaz para el manejo del juego.

Para el desarrollo de Twynbird, inspirado en un juego para móviles llamado FlappyBird, se ha utilizado el motor de videojuegos multiplataforma Unity3D, que ha facilitado mucho el trabajo gracias a la cantidad de herramientas de la que dispone. Para la adquisición de la señal electromiográfica se han utilizado los sensores Trigno Avanti desarrollados por Delsys, una compañía pionera del sector.

Twynbird ayuda al paciente en su proceso de rehabilitación mediante un software que detecta en qué momento está haciendo fuerza. Para la detección, se obtienen los valores de la electromiografía en cada momento y cuando se detecta un incremento importante en los valores de la señal, el paciente obtiene un feedback visual por parte del juego.

Se han realizado una serie de pruebas con sujetos sanos y posteriormente se ha realizado una encuesta de satisfacción. Todos los sujetos que habían pasado por un proceso de rehabilitación afirman que utilizarían una herramienta como esta y consideran que puede ayudar a la rehabilitación.

Palabras clave

Electromiografía, Rehabilitación, Juegos serios, Unity3D, Sensores, Músculos.

Abstract

Rehabilitation has always been a difficult process for humans. In order to make it a more lightweight process and to help the patient with a complementary therapy to the usual, it has been developed a serious game that uses the electromiography to size the muscular strength, as a game management interface.

For Twynbird development, so it has been baptized, it has been used a videogames multiplatform engine Unity3D, that has facilitated the work a lot thanks to the amount of tools that this engine has. For the signal acquisition we have used Trigno Avanti sensors developed by Delsys, a pioneering company of the sector.

Twynbird helps the patient in his rehabilitation process using a software that detects the moment when the patient is straining. For the detection, the electromyographic values are obtained in every moment and when there is detected an important increase in the signal values, the patient obtains a visual feedback of the game.

A series of tests have been carried out with healthy subjects and subsequently a satisfaction survey has been carried out. All the subjects who had gone through a rehabilitation process assert that they would use a tool like this and consider that it can help rehabilitation.

Keywords

Electromyography, Rehabilitation, Serious Games, Unity3D, Sensors, Muscles.

Índice general

1. Introducción	16
1.1. Motivación y objetivos	16
1.2. Fases y métodos	19
1.3. Material utilizado	20
1.3.1. Hardware	20
1.3.2. Software	21
2. EMG y Juegos Serios en rehabilitación física	23
2.1. Electromiografía	23
2.1.1. Electromiografía en rehabilitación	27
2.2. Juegos Serios	29
3. Materiales y métodos	32
3.1. Requisitos y funcionalidades	32
3.2. Comienzo con Unity	33
3.3. Diseño del sistema desarrollado	36
3.3.1. Pantalla de calibración	36
3.3.2. Menú Principal	40
3.3.3. Pantalla de Juego	43
3.3.4. Menú de Configuración	46
3.4. Adquisición de la señal electromiográfica	49
4. Pruebas realizadas y resultados	52
4.1. Pruebas realizadas	52
4.2. Resultados de la encuesta	56
5. Conclusiones y líneas futuras	59
5.1. Conclusiones	59
5.2. Líneas Futuras	60
Referencias	62
Anexo A	66
Anexo B	69

Índice de figuras

1. Introducción	16
Figura 1. Número de estudios sobre electromiografía superficial.	16
Figura 2. Número de jugadores de videojuegos en el mundo.	17
Figura 3. Impacto económico de los videojuegos en Estados Unidos.	18
Figura 4. Sensor Trigno Avanti.	20
Figura 5. Logo de Unity.	21
Figura 6. Logo de Microsoft Visual Studio.	22
2. EMG y Juegos Serios en rehabilitación física	23
Figura 7. Ejemplo de Electromiografía.	23
Figura 8. Proceso de adquisición de la electromiografía.	26
Figura 9. Concepto sobre los juegos serios.	29
3. Materiales y métodos	32
Figura 10. Menú principal de Unity Hub.	33
Figura 11. Menú de creación de proyectos de Unity Hub.	33
Figura 12. Editor de Unity.	34
Figura 13. Pantalla de calibración cuando se inicia el juego.	36
Figura 14. Pantalla de calibración cuando se inicia un intento.	37
Figura 15. Pantalla de calibración al finalizar el primer intento.	37
Figura 16. Pantalla de calibración al finalizar la calibración.	38
Figura 17. Código necesario para comenzar un intento.	38
Figura 18. Código necesario para el movimiento del personaje.	39
Figura 19. Código necesario para terminar un intento.	39
Figura 20. Código necesario para crear el fichero csv.	40
Figura 21. Menú principal del juego.	40
Figura 22. Menú principal una vez encendidos los sensores.	41
Figura 23. Pantalla cuando no superas un obstáculo.	41
Figura 24. Código para la obtención de los valores almacenados en memoria.	41
Figura 25. Código necesario para cambiar a la pantalla de juego.	42
Figura 26. Código necesario para cargar la pantalla de derrota.	42
Figura 27. Código necesario para la actualización del récord.	42
Figura 28. Código necesario para la destrucción de los obstáculos.	43
Figura 29. Pantalla de juego.	43
Figura 30. Función que permite generar obstáculos.	44
Figura 31. Función que permite generar obstáculos de forma recurrente.	44
Figura 32. Código necesario para el movimiento de los obstáculos.	44
Figura 33. Función OnTriggerEnter2D().	45
Figura 34. Función OnCollisionEnter2D().	45
Figura 35. Código que controla el movimiento del personaje en el juego.	45
Figura 36. Menú de configuración.	46
Figura 37. Código que controla el menú de configuración.	47
Figura 38. Código que cambia los valores del juego.	47
Figura 39. Código necesario para cambiar de modo de juego.	48
Figura 40. Código para elegir el modo de funcionamiento de los sensores.	49
Figura 41. Código para crear los canales para obtener la información.	49
Figura 42. Código para añadir los datos necesarios a una lista de datos.	50
Figura 43. Bucle para añadir los nuevos datos a la lista de datos.	50

Figura 44. Código para crear el fichero con los datos de la electromiografía generada durante el juego.	51
Figura 45. Ecuación para obtener la función con la que trabajar en el juego.	51
4. Pruebas y resultados	52
Figura 46. Configuración del menú para las pruebas.	52
Figura 47. Electromiografía de un hombre de 29 años (bíceps).	53
Figura 48. Electromiografía de una mujer de 22 años (bíceps).	54
Figura 49. Electromiografía de una mujer de 22 años (tibial).	54
Figura 50. Porcentaje de personas que han realizado un proceso de rehabilitación (izqda.) y de los que sí han participado el porcentaje de los que se han aburrido (dcha).	56
Figura 51. Porcentaje de personas que conocían la electromiografía.	56
Figura 52. Porcentaje de personas que piensan que el juego puede servir en procesos de rehabilitación.	57
Figura 53. Valoración sobre los diferentes aspectos de la aplicación.	57
Figura 54. Valoración del funcionamiento con diferentes músculos.	58
Figura 55. Valoración general de la aplicación.	58

Índice de tablas

4. Pruebas y resultados	52
Tabla 1. Comparación de resultados entre sujetos (bíceps).	53
Tabla 2. Comparación de resultados entre sujetos (tibial).	54

1 Introducción

En este capítulo se explicará la motivación y los objetivos de la realización de este Trabajo de Fin de Grado, así como sus fases y métodos y finalizará con una breve descripción de los materiales que se han utilizado para su desarrollo.

Este proyecto ha sido realizado dentro del Grupo de Telemática e Imagen de la Universidad de Valladolid, que tiene una amplia experiencia en el desarrollo de aplicaciones de ayuda a las personas.

Mediante la realización de este TFG se intentará ampliar los conocimientos sobre la electromiografía superficial y su adquisición a través de los sensores Trigno Avanti desarrollados por la compañía Delsys.

1.1 Motivación y objetivos

Cada año más de un millón de europeos sufren enfermedades cerebrovasculares y aproximadamente dos tercios sufren parálisis en alguna de sus extremidades (Doumas *et al.*, 2021).

Dada la efectividad que tiene la electromiografía para evaluar las capacidades motrices entre los pacientes con este tipo de enfermedades, el número de estudios y publicaciones sobre esta se incrementa cada año. La electromiografía puede dar métricas que pueden pronosticar valores y predecir la funcionalidad muscular en un tiempo (Steele, Papazian and Feldner, 2020).

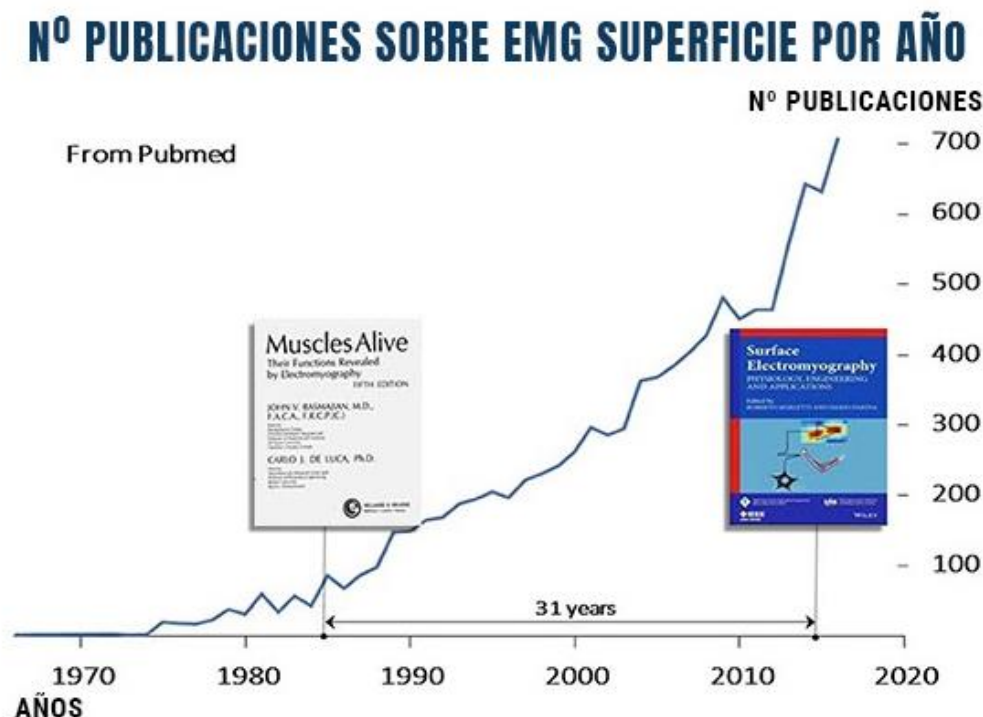


Figura 1. Número de estudios sobre electromiografía superficial (Campanini *et al.*, 2020; mDurance Studio, 2021)

La participación en los procesos de rehabilitación ayuda a incrementar los beneficios y su efectividad. Los juegos añaden un componente motivacional a los pacientes que sufren enfermedades cerebrovasculares para aumentar su participación, esta disminuye cuanto más largo es el proceso de recuperación, y así aumentar el nivel de recuperación de su movilidad corporal (Flores *et al.*, 2008).

El crecimiento de personas que juegan a videojuegos crece cada año. En 2021 se llegó a 2.81 billones de usuarios y se espera que esta cifra siga creciendo en el futuro, como se puede observar en la figura 2 (Gilbert, 2021).

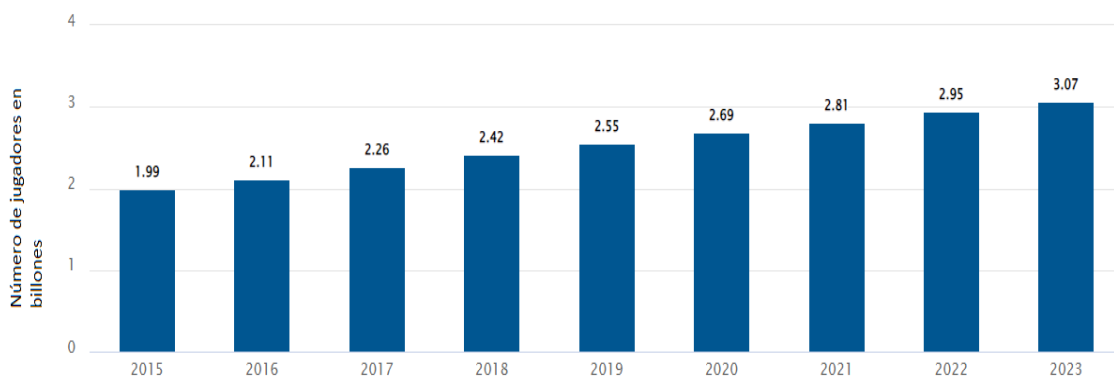


Figura 2. Número de jugadores de videojuegos en el mundo (Gilbert, 2021)

En los últimos años esta industria ha representado un incremento exponencial económicamente hablando (representa un 0.11% del PIB español). En 2017, el impacto en la producción del sector de los videojuegos fue de 3.577 millones de euros, 1.452 millones de euros más de valor añadido y generó 22.628 puestos de trabajo. Además, esta industria representa el 14.3% del sector de edición, un 9.6% del sector de producción audiovisual, un 3.8% del de programación y un 3.2% del sector de las telecomunicaciones (Collado, 2018).

El impacto en la producción del sector de los videojuegos fue de 3.577 millones de euros, 1.452 millones de euros más de valor añadido y generó 22.628 puestos de trabajo (Collado, 2018).

Para que se vea la importancia de los videojuegos no sólo en España, se pueden ver los datos en otros países, por ejemplo, la gráfica de la figura 3 corresponde al ingreso por ventas de videojuegos en Estados Unidos.

Dado que la motivación es uno de los factores más importantes a la hora del aprendizaje autónomo y la felicidad del paciente, la electromiografía es una de las herramientas que más nos pueden ayudar en la caracterización de los músculos y los videojuegos están más en auge que nunca, hemos decidido combinar todo y realizar un videojuego cuyo propósito sea el de ayudar a las personas.

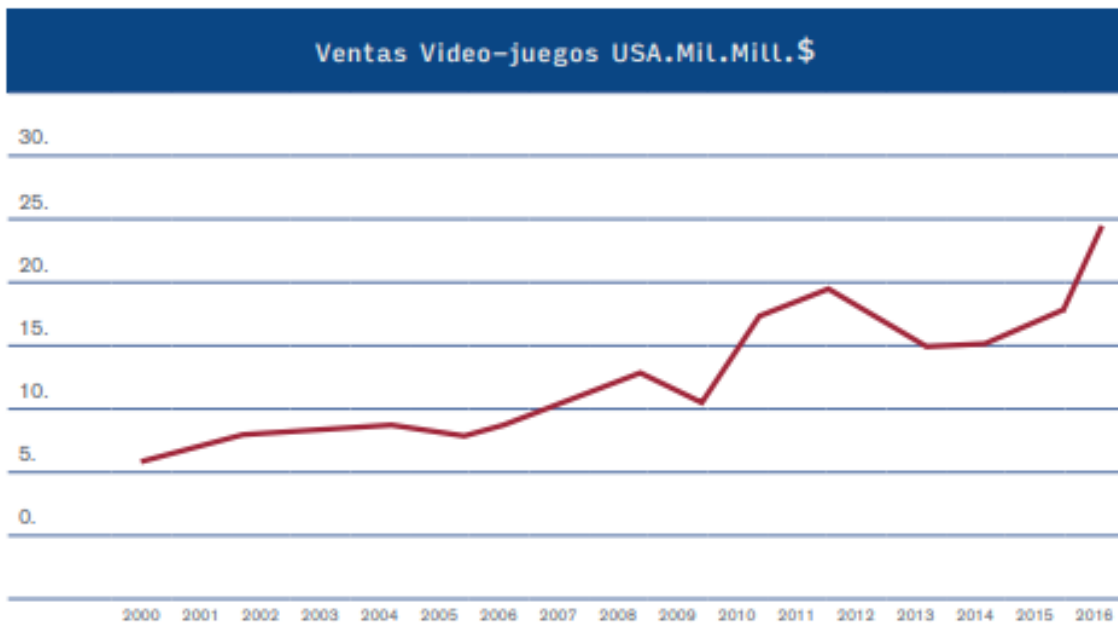


Figura 3. Impacto económico de los videojuegos en Estados Unidos (Collado, 2018)

Nuestros **objetivos** son varios:

- Proporcionar a los pacientes una forma alternativa de realizar su proceso de rehabilitación.
- Facilitar este proceso pudiéndose realizar en cualquier lugar siempre que se tenga un ordenador y un sensor, que son fácilmente transportables.
- Proporcionar información precisa y actualizada del estado de los músculos gracias a la señal electromiográfica.

1.2 Fases y métodos

En este apartado se comentarán las principales fases que ha seguido la elaboración de nuestro TFG:

1. **Realización de curso sobre Unity.** Se realizan una serie de tutoriales oficiales de Unity y del lenguaje de programación C#, además de un curso de formación intensiva en la plataforma de cursos online Udemy (Gomila, 2022).
2. **Estudio de los sensores.** Conocimiento del tipo de sensor a utilizar, así como de las diferentes funcionalidades que existen dentro de él y los diferentes datos que nos puede aportar.
3. **‘Lluvia de ideas’:** Se piensan propuestas sobre la temática, las funcionalidades y el desarrollo del videojuego.
4. **Investigación sobre la API de Delsys.** API proporcionada por la compañía encargada de la fabricación de los sensores que nos ayuda a acoplar todos los datos provenientes de los sensores con el motor de videojuegos Unity (Delsys, 2022a).
5. **Versión inicial.** Se realiza la primera versión de la aplicación que nos permite controlar de manera básica el personaje.
6. **Mejoras de la aplicación.** Se añaden funcionalidades a la aplicación, tales como un menú de configuración para poder modificar la dificultad del juego.
7. **Versión final:** Se finaliza la aplicación y se explica las funcionalidades que tiene.
8. **Pruebas:** Se realizan pruebas en pacientes sanos y se realiza una explicación sobre los resultados obtenidos.

1.3 Recursos disponibles

Durante la realización de este Trabajo de Fin de Grado se ha utilizado diferente instrumentación hardware y software que se va a explicar a continuación.

1.3.1 Hardware

Se ha utilizado un ordenador portátil con las siguientes características:

- Modelo del sistema: HP Pavilion Power Laptop 15-cb0xx
- Procesador: Intel(R) Core (TM) i7-7700HQ CPU @ 2.80GHz, 2808 Mhz, 4 procesadores principales, 8 procesadores lógicos
- Memoria RAM: 12,00 GB
- Tipo de sistema: PC basado en x64
- Almacenamiento: 222 GB
- Sistema operativo: Microsoft Windows 10 Home
- Versión: 10.0.19044 compilación 19044
- Tarjeta Gráfica: NVIDIA GeForce GTX 1050

Además, se han utilizado los siguientes sensores de movimiento:

- Sensor Trigno Avanti

Sensores capaces de capturar la actividad muscular y los datos de los movimientos (Delsys, 2022b).

- Peso: 14g
- Tamaño: 27x37x13 mm
- Batería: 4-8 horas
- Distancia entre electrodos: 10mm
- Ancho de banda de la electromiografía: 10-850Hz/20-450Hz
- Material de Contacto: 99.9 % plata
- Canales: 1 x EMG / 6 x IMU

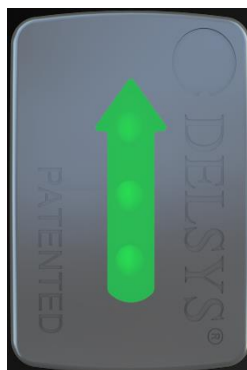


Figura 4. Sensor Trigno Avanti (Delsys, 2022).

1.3.2 Software

Se han utilizado las siguientes herramientas:

- Unity

Unity es la mejor plataforma de desarrollo en tiempo real, además de ser un robusto ecosistema para el diseño de videojuegos de software pudiendo desarrollar videojuegos desde destinados a ordenadores hasta consolas de última generación como PlayStation 5, incluso pudiendo desarrollar juegos en realidad virtual (VR) (Unity Technologies, 2022c).



Figura 5. Logo de Unity (Unity Technologies, 2022c)

- C#

C# es un lenguaje de programación que permite desarrollar aplicaciones que se ejecutan en .NET, un sistema de ejecución virtual. C# es un lenguaje de programación orientado a objetos orientado a componentes cuyo código fuente se compila en un lenguaje intermedio. (Wagner, 2022).

Gracias a C# se pueden crear aplicaciones sólidas y que perduran en el tiempo debido a que tiene diferentes características que lo facilitan (Wagner, 2022):

1. Control de excepciones
2. Sistema de tipos unificado
3. La sintaxis crea un patrón común para trabajar con datos de diferentes orígenes
4. Permite la asignación dinámica de objetos

C# insiste mucho en el control de versiones para cerciorarse de que las aplicaciones y bibliotecas sean compatibles durante mucho tiempo (Wagner, 2022).

- Microsoft Visual Studio

Visual Studio es un entorno de desarrollo integrado (IDE) con numerosas características para favorecer el desarrollo de software. Es compatible con numerosos lenguajes de programación como C# y .NET. Visual Studio facilita el trabajo gracias a la numerosa cantidad de herramientas de las que dispone. (Microsoft, 2022).



Figura 6. Logo de Microsoft Visual Studio (Microsoft, 2022)

2 EMG y Juegos Serios en rehabilitación física

En este capítulo se estudiará la electromiografía (EMG) y sus aplicaciones médicas, después se describirán los juegos serios y finalmente se analizarán los beneficios que provocan estos en los pacientes de forma conjunta y casos previos en el que esta forma de actuar haya sido eficaz.

2.1 Electromiografía

La electromiografía es una señal biomédica que mide la electricidad generada en los músculos durante su contracción. La señal electromiográfica es dependiente del sistema nervioso, que controla la actividad muscular, y de las propiedades anatómicas y fisiológicas de los músculos (Reaz, Hussain and Mohd-Yasin, 2006). La activación electrofisiológica de un músculo inicia la producción de fuerza mecánica. Esta actividad muscular refleja el grado de activación de un músculo y este se puede observar en la electromiografía (Disselhorst-Klug, Schmitz-Rode and Rau, 2009).

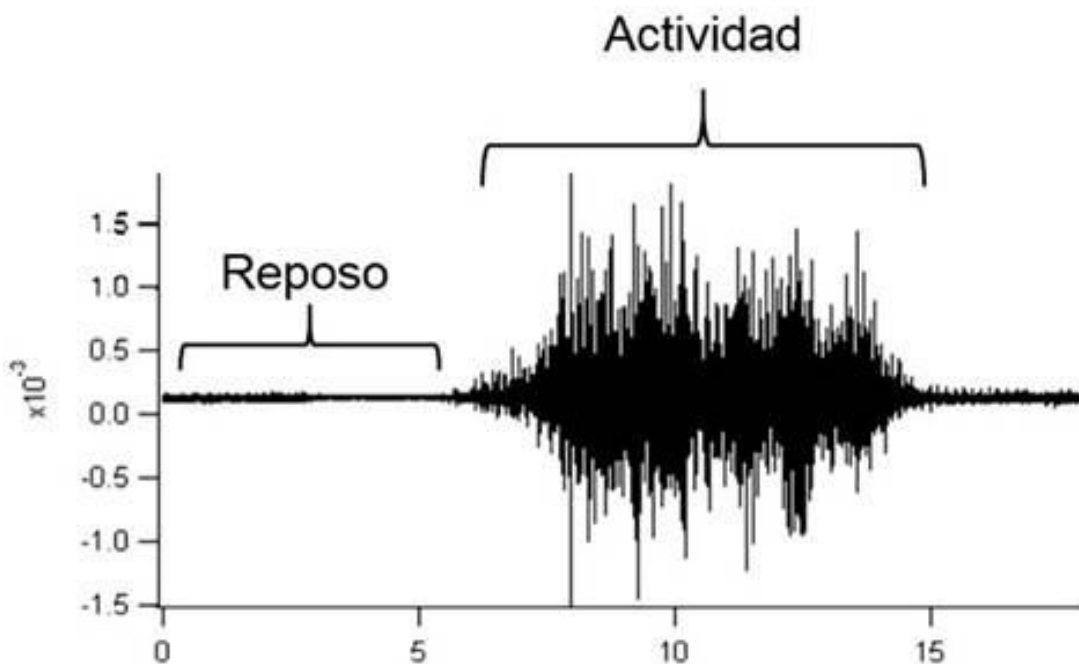


Figura 7. Ejemplo de Electromiografía (Guzmán-Muñoz and Méndez-Rebolledo, 2018)

La obtención de la electromiografía consta de varios procesos: adquisición de la señal, acondicionamiento de la señal y procesamiento de la señal (Norali, Mat Som and Kangar-arau, 2009).

Dependiendo del tipo de electrodo utilizado en la adquisición de la señal, la electromiografía puede ser de dos tipos: superficial o intramuscular. La diferencia entre ambas es que la primera se realiza con electrodos no invasivos, mientras que en la segunda se realiza con electrodos invasivos (Chowdhury *et al.*, 2013).

Se deben considerar diversos criterios para la elección entre los diferentes tipos de electrodos. El criterio más importante es la accesibilidad del músculo en cuestión. Por

ejemplo, sólo se puede detectar actividad en músculos como el ilio-psoas con los electrodos intramusculares. Por otro lado, la interferencia de estos en los patrones de movimiento es importante, ya que se ha demostrado que en pacientes con parálisis cerebral se reduce su distancia entre pasos (18%), su velocidad (23.5%) y su cadencia (7.9%). Además, aunque los electrodos superficiales pueden informar de manera más global de la actividad locomotora de los músculos más grandes, pueden provocar interferencias con los músculos adyacentes, lo que puede afectar de forma importante a la señal obtenida (Frigo and Crenna, 2009).

La mayoría de las aplicaciones utilizan la EMG superficial debido a que es más cómoda y tiene menor riesgo de infección.

En la fase de adquisición de la señal, también hay que tener en cuenta las propiedades, la localización, la orientación de los electrodos (Rodríguez-Tapia *et al.*, 2020).

Siempre ha habido mucha incertidumbre en cuánto a esto último, por ello la Unión Europea desarrolló el proyecto SENIAM (Surface Electromyography for the Non-Invasive Assessment of Muscles). SENIAM desvela una serie de recomendaciones que han sido probadas clínicamente por los miembros del proyecto (Stegeman and Hermens, 2007):

- Forma y tamaño

No existe ningún criterio sobre la forma del electrodo. Sí se define un tamaño máximo que debe ser de 10 milímetros en la dirección de la fibra muscular.

- Distancia entre electrodos

Esta distancia se define como la distancia entre las áreas conductivas de dos electrodos bipolares. Esta debe ser de 20 milímetros. En músculos relativamente pequeños, esta distancia debe ser de menos de un cuarto de la longitud del músculo para evitar inestabilidades asociadas a las zonas tendinosas.

- Material

El material debe de tener una baja impedancia entre el electrodo y la piel y un comportamiento estable en el tiempo. Se recomienda el uso de electrodos de plata pre-gelificados.

- Localización

Los electrodos deben ser colocados donde pueda obtenerse una EMG estable y de calidad. Estos lugares son aquellos en los que se encuentran los tendones musculares lejos de otros músculos activos que puedan interferir con la señal.

- Orientación y Fijación

Es recomendable que los electrodos estén colocados paralelamente a las fibras musculares. Además, es recomendable sujetar estos a la piel mediante cintas elásticas de manera que no se puedan mover.

La calidad de la señal de electromiograma viene influida por la relación señal-ruido (SNR). Esta examina el porcentaje de información que hay en una señal EMG. Para

maximizar la cantidad de información de una electromiografía, esta debe de tener la menor cantidad de ruido posible (Reaz, Hussain and Mohd-Yasin, 2006).

Las principales fuentes de ruido que pueden afectar a un electromiograma son (De Luca, 2002) (Chowdhury *et al.*, 2013) :

- Ruido Electromagnético

El ruido procedente de la radiación electromagnética, como la radio o la televisión. La superficie de nuestro cuerpo se inunda de radiación electromagnética y es imposible evitar la exposición a esta en la superficie de la tierra.

- Ruido generado por el equipamiento electrónico

Todo equipamiento electrónico genera ruido eléctrico. Este no se puede eliminar, aunque se puede reducir utilizando equipamiento electrónico de alta calidad.

- Movimiento de los dispositivos

Las fuentes principales de este tipo de ruido son el movimiento del cable que conecta los electrodos con el amplificador y el punto de contacto entre la superficie de los electrodos y la piel. Una posible solución a este ruido sería la utilización de una capa de gel conductor entre la superficie del electrodo y la piel.

- Inestabilidad de la señal

La amplitud de la EMG es prácticamente aleatoria. Los componentes con frecuencia entre 0 y 20 Hz son inestables debido a la frecuencia de disparo de las unidades motoras. El número de unidades motoras activas, la frecuencia de disparo de estas y la interacción entre las fibras musculares pueden modificar la información de la electromiografía.

- Actividad Electrocardiográfica

La principal causa de interferencias en la señal electromiográfica se debe a la actividad eléctrica del corazón. La actividad cardíaca contamina la EMG, principalmente cuando se examinan músculos situados cerca del corazón. Es muy difícil eliminar estas interferencias debido a que sólo es visualmente identificable en el 25% de las contracciones máximas voluntarias de la activación electromiográfica.

- Cross-Talk

El cross-talk es la parte de la EMG grabada por los electrodos situados en la piel que pertenece a un músculo situado cerca del músculo de interés. Este efecto limita la efectividad de la electromiografía en los casos en los que se quiera obtener información de un músculo específico (Mesin, 2020).

Este efecto puede ser eliminado filtrando las señales grabadas y aplicando umbrales de detección para eliminar las señales de los músculos más lejanos (Najarian and Splinter, 2012).

Es imposible eliminar ciertos tipos de cross-talk de las señales electromiográficas, sobre todo en los casos en los que las señales provienen de músculos muy cercanos

entre sí (Talib *et al.*, 2019).

La señal obtenida por los electrodos superficiales suele tener una amplitud de 10 mV (± 5 mV). Esta amplitud es pequeña para el procesamiento, por lo que en la mayoría de los casos debe pasar por un proceso de amplificación (Norali, Mat Som and Kangar-arau, 2009).

En la siguiente imagen se puede ver todo el procesado que sufre la electromiografía desde su adquisición.

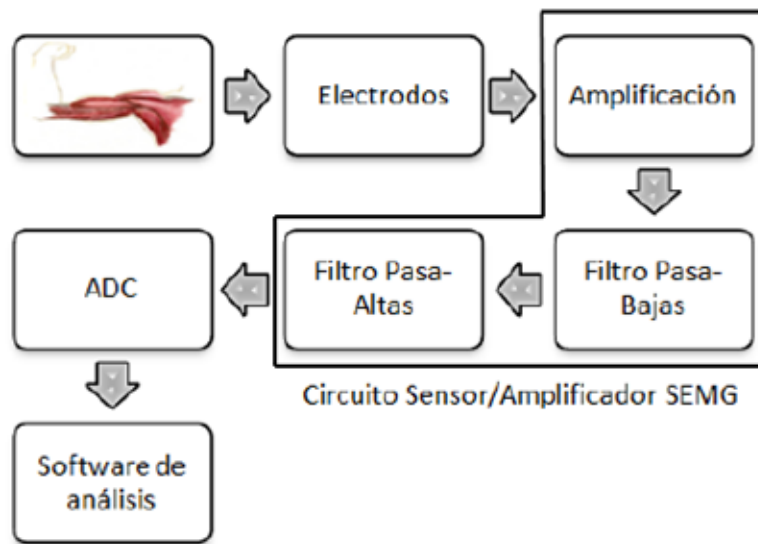


Figura 8. Proceso de adquisición de la electromiografía (Correa-Figueroa *et al.*, 2016)

Una vez realizado todo el proceso de adquisición de la señal, llega el momento de su análisis y extracción de características (Rangayyan, 2015):

- Características Temporales

Los valores de amplitud de la electromiografía siguen una distribución gaussiana. La amplitud puede variar en torno a los 10 mV pico a pico, pero todos los factores que hemos descrito anteriormente pueden hacer que esta varíe.

- Características Frecuenciales

El espectro de frecuencia se extiende desde los 5 Hz hasta los 500 Hz. Su frecuencia de muestreo suele estar entre los 1000 y 2000 Hz dependiendo de la aplicación (en nuestro caso utilizamos una frecuencia de 1740 Hz).

El espectro de un electromiograma está influenciado por la tasa de disparo de las unidades motoras en frecuencias bajas y por la forma de los potenciales de acción de las unidades motoras en frecuencias altas.

En la actualidad, la electromiografía es muy utilizada en muchos sectores. Uno de esos sectores es el de la medicina, debido a que la electromiografía es una herramienta muy útil para comprobar el correcto funcionamiento del sistema nervioso y el modelado musculoesquelético.

Algunas de las aplicaciones médicas de la electromiografía más relevantes son (Najarian

and Splinter, 2012):

- Visibilidad de la fatiga muscular

Los factores que pueden indicar fatiga muscular son: reducción de la amplitud, de la energía o del contenido a altas frecuencias de la EMG

- Fuerza muscular

La fuerza generada por un músculo afecta a la amplitud y la frecuencia del electromiograma

- Anestesia

La respuesta muscular a la estimulación eléctrica puede ayudar a determinar cuál es el nivel de anestesia local.

- Detección de enfermedades

La información obtenida del estudio de la electromiografía ayuda a identificar la conductividad de los músculos, así como posibles problemas en ellos.

2.1.1 Electromiografía en rehabilitación

El uso de la electromiografía como herramienta de diagnóstico ha hecho posible la caracterización de la contracción y relajación de los músculos. Dentro del campo de la rehabilitación la EMG se usa para medir la tensión experimentada por los músculos, y transformar esa medición en retroalimentación para el paciente. Esto les permite mejorar el control de sus músculos y genera una motivación extra en su proceso de rehabilitación (Kotov-Smolenskiy *et al.*, 2021).

Se sabe que los trastornos causados por un accidente cerebrovascular son acompañados por compensaciones motoras de los músculos que no han sido afectados. Para identificar estas compensaciones, se ha probado que la electromiografía superficial es más precisa que los sensores inerciales (Kotov-Smolenskiy *et al.*, 2021).

Un ejemplo de la utilización de señales electromiográficas en la rehabilitación puede ser un estudio (Moreland, Thomson and Fuoco, 1998) en el que se demostraba la eficacia de la electromiografía para la mejora del movimiento de dorsiflexión del tobillo.

El objetivo de este estudio era examinar la eficacia de la realimentación ofrecida por la electromiografía comparada con los mecanismos fisioterapéuticos convencionales en casos de pacientes que habían sufrido un accidente cerebrovascular.

Aunque las estadísticas no ofrecieron datos relevantes, si se pudo observar una mejora en el ángulo del tobillo y en la calidad de la marcha (Moreland, Thomson and Fuoco, 1998).

En otro estudio (Woodford and Price, 2007) en el que se realizaron pruebas sobre diferentes pacientes se observaban mejoras en el análisis de la marcha y en la restauración de la fuerza motora. Las pruebas no eran demasiado grandes, por lo que no se llegó a ninguna conclusión de que fuera beneficiosa en el tratamiento.

La electromiografía nos abre un universo de posibilidades dentro del mundo de la rehabilitación y del diagnóstico de los músculos, además de en el estudio del control del exoesqueleto (Kotov-Smolenskiy *et al.*, 2021).

En algunos casos, la electromiografía se combina con el electroencefalograma. Aunque actualmente el uso de esta técnica combinada ayuda a aclarar algunos mecanismos fisiológicos y patológicos para evaluar la efectividad de los tratamientos, lo cierto es que para explotar todo el potencial de ambas se deben desarrollar más estudios y mejorar las evidencias científicas (Brambilla *et al.*, 2021).

2.2 Juegos Serios

Los juegos serios pueden definirse como juegos cuyo propósito principal no es entretener ni divertir (Laamarti, Eid and El Saddik, 2014). Los juegos serios están diseñados para mejorar algunos aspectos de entrenamiento o aprendizaje, o para usarse en el área de la educación o la sanidad (Dias, Barbosa and Vianna, 2018).



Figura 9. Concepto sobre los juegos serios (Osyren, 2020)

El término “juego serio” fue usado por primera vez por Abt en 1970 (Abt, Clark C. , 1970). En sus inicios, los juegos serios se pensaron para entrenar a las personas para desarrollar diferentes tareas en su trabajo, como por ejemplo en el ejército de los Estados Unidos (De Gloria, Bellotti and Berta, 2014).

Durante los últimos años, se ha incrementado el uso en muchas áreas. Una de estas áreas en las que se ha empleado de forma satisfactoria es en la rehabilitación, sobre todo en casos de daño cerebral.

El éxito de la rehabilitación en muchos casos se debe a la realización de ejercicios en la propia casa de los pacientes, pero, para que esta tenga éxito, los ejercicios deben realizarse correctamente (Bonnehère *et al.*, 2016).

El problema es que un gran porcentaje de los pacientes deja el proceso a medias por diversas razones. Según un estudio realizado en la universidad de Bruselas, la participación de los pacientes se reducía cuanto más largo era el proceso de rehabilitación. Para poder encontrar una solución se preguntó a los pacientes el porqué de su abandono y estos respondieron que, o no sabían realizar los ejercicios de forma correcta o directamente se aburrían (Bonnehère *et al.*, 2016).

Los juegos serios pueden jugar un rol muy importante dando a los pacientes una motivación extra en la que sus tareas están adaptadas a lo que necesitan en cada momento. Además, ayudan a los terapeutas a seguir la evolución y supervisar a los pacientes (Hocine, Gouaïch and Cerri, 2014).

Los juegos serios se pueden clasificar siguiendo una serie de criterios. Estos son (Rego, Moreira and Reis, 2010):

- Área de aplicación
Son las diferentes áreas en las que el juego se puede aplicar. Se pueden considerar dos áreas generales de aplicación: cognitiva y motora. La cognición incluye facultades mentales como pueden ser la memoria, la concentración o el razonamiento mientras que la rehabilitación motora incluye las discapacidades físicas o enfermedades.
- Tecnología
Un aspecto muy importante es la capacidad de la tecnología usada por el paciente de interactuar con el sistema, desde un teclado o ratón hasta un sistema de realidad virtual.
- Interfaz
La interfaz usada en el juego (2D o 3D)
- Número de jugadores
Número de personas que pueden jugar a la vez.
- Género del juego
La influencia del juego puede variar dependiendo de su género. Los juegos pueden ser desde de acción, para evaluar el movimiento, o de estrategia, para evaluar las capacidades mentales, incluso llegar a ser una combinación de ambas.
- Adaptabilidad
Los pacientes deben ser capaces de cambiar la dificultad del juego a medida que van progresando. Esto puede llevarse a cabo por medio de niveles o incrementando la dificultad de llegar al objetivo.
- Feedback
El juego debe ser capaz de transmitir al paciente los resultados de sus acciones. Esto puede servir para evaluar el progreso de un paciente en su rehabilitación.
- Progreso
Es muy importante que el juego tenga la capacidad de guardar los resultados del paciente
- Portabilidad
La capacidad del sistema para usarse en diferentes lugares ya sea en casa, en el hospital o en una clínica de rehabilitación.

La efectividad de los juegos serios está ampliamente comprobada por diversos estudios. En uno de ellos (Wouters *et al.*, 2013) se explica claramente los beneficios de la utilización de los juegos serios.

Se llegó a la conclusión de que el conocimiento adquirido era superior que mediante los métodos de enseñanza convencionales. En segundo lugar, el nivel de retención era superior que mediante los métodos de entrenamiento convencionales. En tercer lugar, y el más importante en este caso, el nivel de atención y motivación era ampliamente superior que utilizando terapias convencionales.

3 Materiales y métodos

En este apartado se explicará el juego serio desarrollado en su totalidad. En primer lugar, se detallarán los requisitos y funcionalidades que ha de cumplir. Después, se comentará el desarrollo y la implementación de cada una de sus partes. En último lugar, se verá un ejemplo práctico de su utilización y configuración.

3.1 Requisitos y funcionalidades

Cuando se pensó la propuesta del juego, se propusieron una serie de requisitos que este debía de cumplir. Son los siguientes:

- Disponibilidad para los públicos de todas las edades.
- Interfaz amigable y fácil de usar.
- Posibilidad de entrenamiento con diferentes músculos.
- Posibilidad de cambiar de dificultad según se avanza el proceso de rehabilitación.

El juego está pensado para jugar en un ordenador. No se necesita que este tenga grandes capacidades ya que el juego no dispone de grandes efectos ni gráficos.

Esto es así ya que se puede trasladar fácilmente en un ordenador portátil y poder realizar la rehabilitación desde cualquier lugar, ya sea en una clínica como en la propia casa del paciente.

Una vez finalizada la implementación de nuestro videojuego, se van a valorar las funcionalidades que tiene:

- Se debe realizar un proceso de calibración previo al inicio del juego para ver el progreso de este en su proceso de rehabilitación
- El comienzo y la duración de los intentos durante el proceso de calibración es elegida por el paciente. Esto proporciona una gran flexibilidad a la hora de que se obtengan los valores más correctos.
- El paciente es capaz de obtener ficheros con los datos de su calibración y el valor de la señal electromiográfica en su intento con valor más alto y su intento con valor más bajo
- El paciente tiene indicadores de que debe hacer para conectar los sensores al juego e indicadores sobre lo que tiene en cada momento para iniciar el juego.
- El paciente tiene la posibilidad de cambiar entre modos de juego para cuando este se encuentra en las etapas iniciales de su proceso de recuperación.
- El paciente es capaz de modificar el umbral de dificultad que tiene que superar para conseguir el objetivo del juego.
- Se pueden extraer los datos de los valores de la señal electromiográfica durante el desarrollo del juego.
- El juego muestra el récord absoluto del juego para crear competitividad y generar una motivación extra en el paciente
- Se puede finalizar el juego en cualquier momento guardando los resultados

3.2 Comienzo con Unity

El motor gráfico elegido para el desarrollo del videojuego ha sido Unity. Los motivos de la elección son varios, y las facilidades que nos proporciona a la hora de generar objetos, animaciones o personajes son muchas.

Para empezar, se puede elegir entre descargar una versión de Unity o descargar Unity Hub. Unity Hub es una herramienta que nos permite administrar las cuentas, crear proyectos, descargar o actualizar las diferentes versiones de Unity o incluso comentar en foros con otros creadores (Unity Technologies, 2022b).

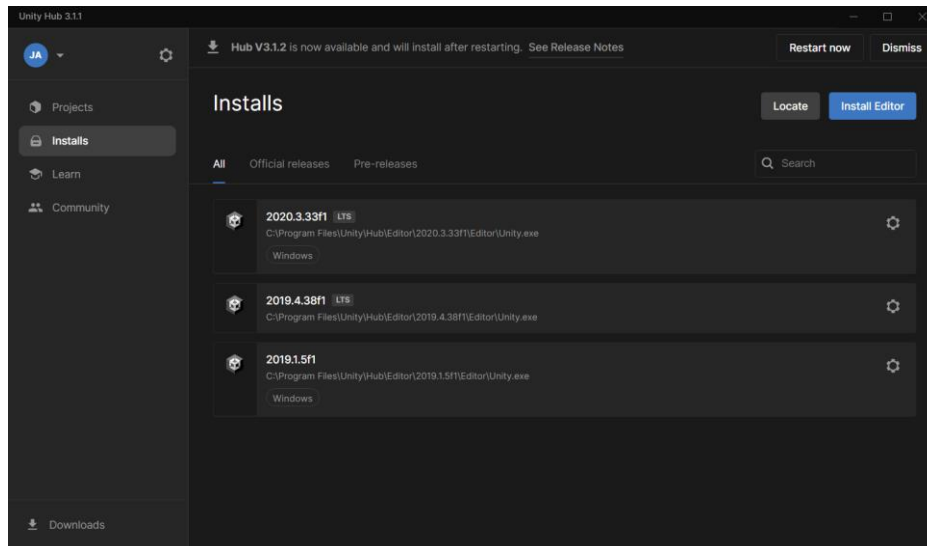


Figura 10. Menú principal de Unity Hub

Una vez instalada esta aplicación, se podrá empezar a crear proyectos. Para ello se debe pulsar en la pestaña *Projects* a la izquierda de la pantalla y después *New Project* y aparecerá la siguiente pantalla. Como se puede ver podemos elegir el tipo de juego que queremos desarrollar, desde 2D, 3D hasta Realidad Virtual. En este caso se ha optado por desarrollar un juego en 2D.

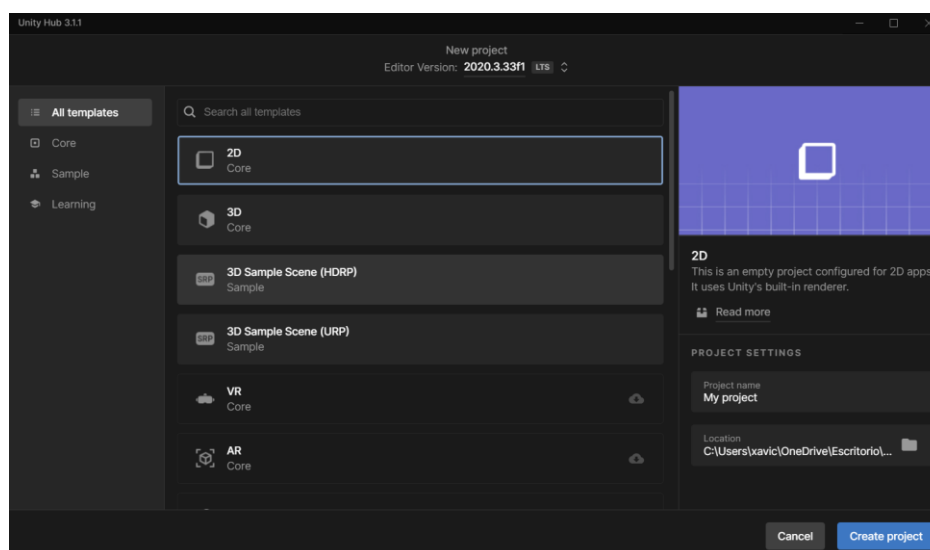


Figura 11. Menú de creación de proyectos de Unity Hub.

Una vez instalada esta aplicación, se podrá empezar a crear proyectos. Para ello se debe pulsar en la pestaña *Projects* a la izquierda de la pantalla y después *New Project* y aparecerá la siguiente pantalla. Como se puede ver podemos elegir el tipo de juego que queremos desarrollar, desde 2D, 3D hasta Realidad Virtual. En este caso se ha optado por desarrollar un juego en 2D.

Una vez creado el proyecto nos aparecerá la siguiente pantalla:

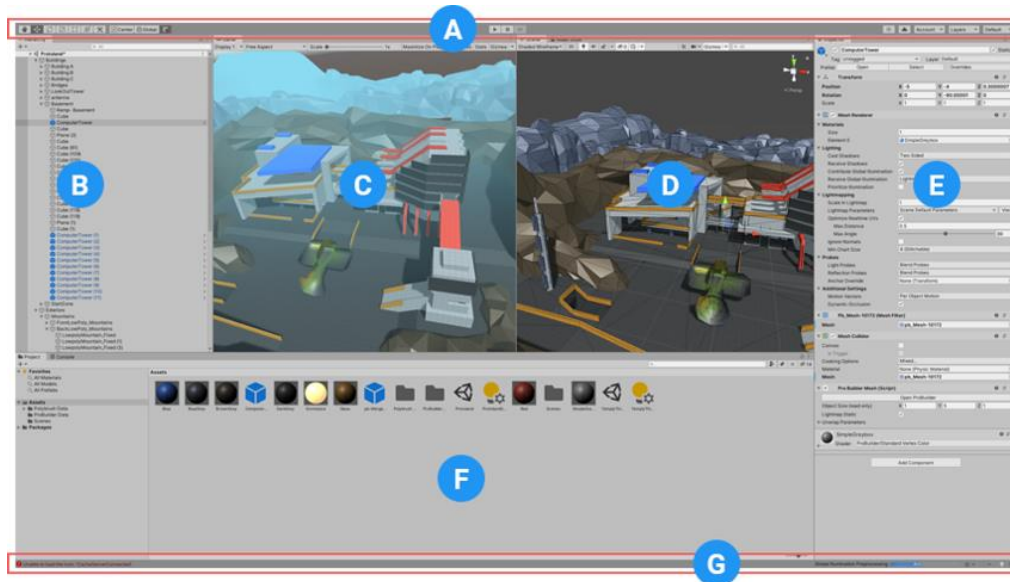


Figura 12. Editor de Unity (Unity Technologies, 2020)

En ella se pueden ver diferentes ventanas que se van a explicar a continuación (Unity Technologies, 2020):

A. Barra de herramientas

Da acceso a las herramientas de edición más importantes. A la izquierda tenemos las herramientas más básicas para editar la vista de escena y los objetos que se encuentran en ella. En el centro se encuentran los botones de inicio y pausa de ejecución. Los botones de la derecha dan acceso a la cuenta de Unity, a las capas en las que se agrupan los objetos y el diseño de la ventana de edición

B. Ventana de jerarquía

Representación jerárquica de cada objeto que hay en la escena. Cada elemento en la escena se encuentra en la ventana de jerarquía, por lo que ambas están relacionadas.

C. Vista de juego

Simula como se verá el juego a través de las cámaras de la vista de escena. Cuando se pulsa el botón de play, se comienza la simulación

D. Vista de escena

Permite navegar y editar la escena. Puede mostrar una perspectiva en dos o tres dimensiones dependiendo del tipo de proyecto en el que estes trabajando.

E. Inspector

Permite editar las propiedades del objeto que esté seleccionado en ese momento. Cada objeto tiene sus propias propiedades, por lo que esta ventana cambia cada vez que se selecciona un objeto.

F. Ventana de proyectos

Aporta información sobre los assets que están disponibles para usar en el proyecto.

G. Barra de estado

Barra en la que aparecen notificaciones que aportan información sobre los procesos de Unity, procesos de compilación o estado de las herramientas

3.3 Diseño del sistema desarrollado

El juego está compuesto por diferentes escenas: la pantalla de calibración, el menú principal, la pantalla de juego y el menú de configuración.

3.3.1 Pantalla de calibración

Cuando se inicia el juego se empieza con una pequeña calibración para ver el estado del paciente en ese momento. Esto se hace para poder ajustar el nivel del umbral en el juego en el menú de configuración.

En primer lugar, se tiene que poner en funcionamiento los sensores. La inicialización de estos se lleva a cabo de manera automática una vez se inicia el juego.

Antes de empezar con la calibración, el paciente podrá introducir su nombre y apellidos si lo desea. Es recomendable introducir los datos personales para poder nombrar los archivos que se generan con los datos del juego.

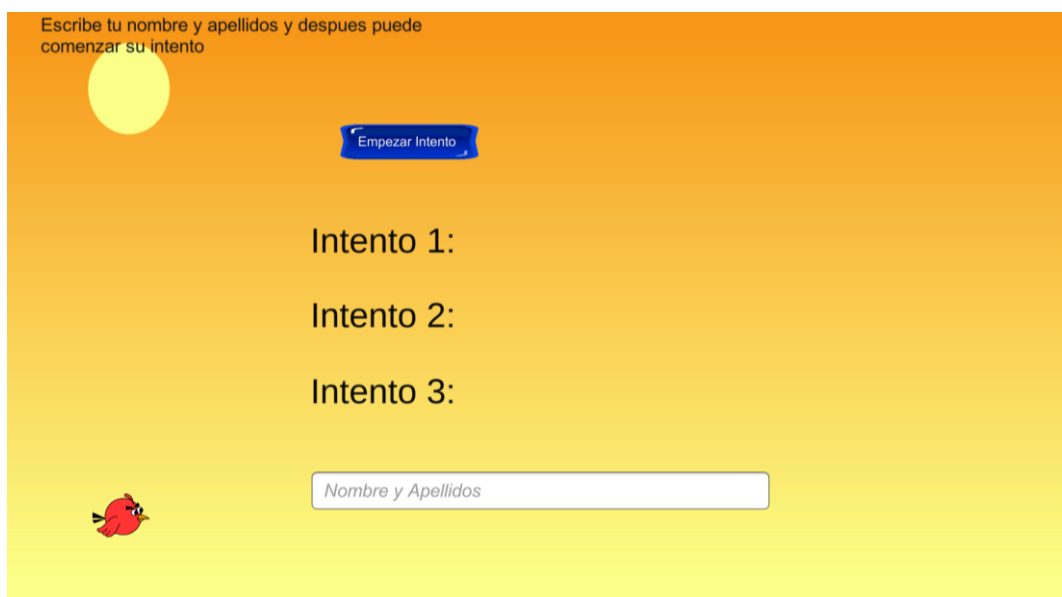


Figura 13. Pantalla de calibración cuando se inicia el juego.

La calibración consta de tres intentos en los que el paciente tiene que hacer la mayor fuerza posible. La duración de cada intento no está determinada, por lo que el paciente puede finalizarlo cuando sea conveniente. Durante los tres intentos de la calibración se propone un objetivo extra. Este es el de intentar que el pájaro llegue a tocar el sol. El motivo de este “minijuego” no es otro que intentar que el paciente tenga una motivación adicional para realizar la calibración correctamente.

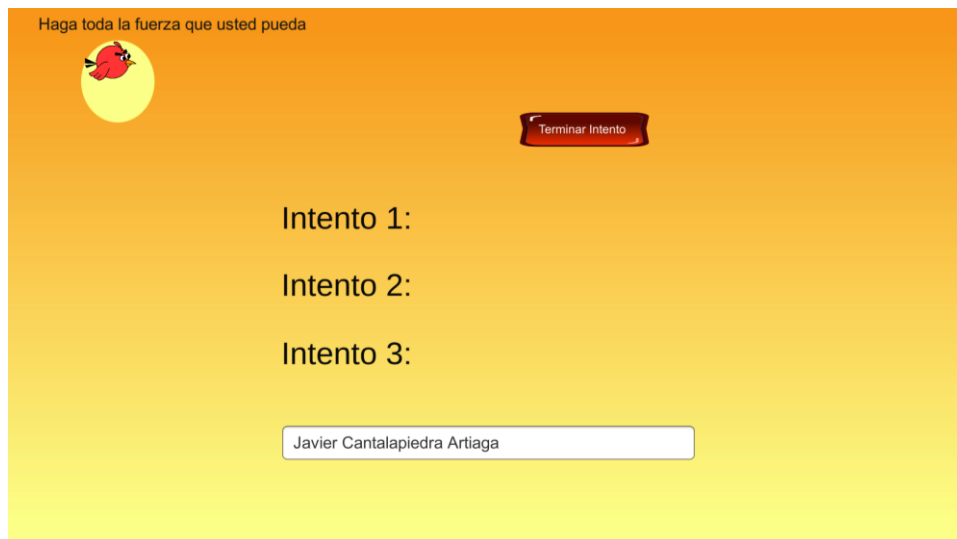


Figura 14. Pantalla de calibración cuando se inicia un intento.

En la interfaz de usuario se puede ver en todo momento lo que hay que hacer para seguir avanzando con la calibración, además del valor máximo de la señal electromiográfica en cada intento realizado. Los botones irán apareciendo solo en el momento en el que el paciente tenga que pulsarlos, como se puede ver en las diferentes imágenes.

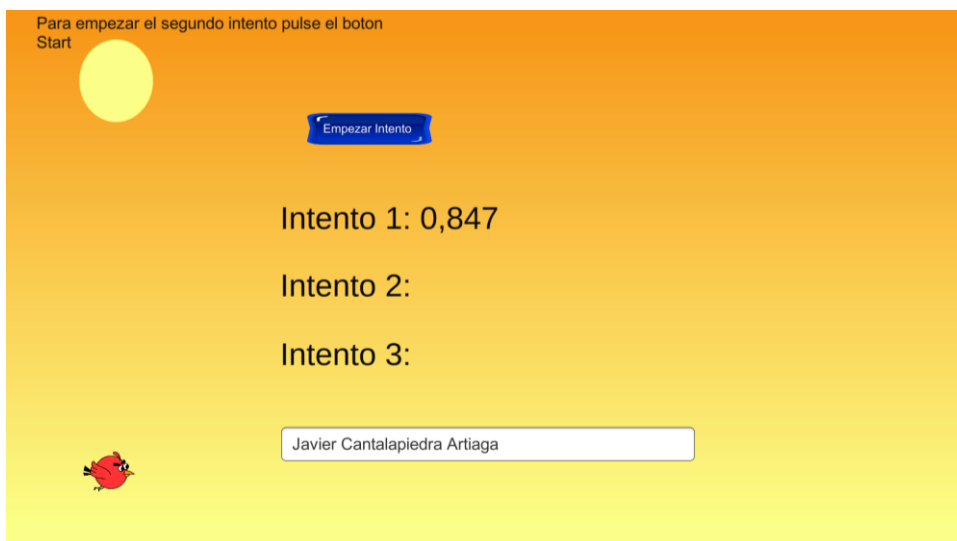


Figura 15. Pantalla de calibración al finalizar el primer intento.

Cuando se finaliza el tercer intento de la calibración, se generará un archivo .csv en el que se guardarán los datos de los intentos realizados durante la calibración.

Una vez acabada la calibración, se podrá comenzar con el juego.



Figura 16. Pantalla de calibración al finalizar la calibración.

Código necesario para la implementación de la pantalla de calibración

Cuando se pulse el botón “Empezar intento” se ejecuta el código que se puede ver en la figura 17, el cual realiza las siguientes tareas:

1. Actualiza el número de intento.
2. Configura la pipeline para la correcta transmisión de datos entre los sensores y el juego. Este proceso se explicará en el siguiente apartado.
3. Arranca la pipeline que hemos configurado anteriormente.
4. Muestra los comentarios necesarios al paciente para que sepa lo que hacer en cada momento.

```
1. trial++;
2. bool success = ConfigurePipeline();
   if (success)
   {
3.     RFPipeline.Start();
4.     APIStatusText.text = "Haga toda la fuerza que
       usted pueda";
   }
   else
   {
       APIStatusText.text = "Ha ocurrido un error";
   }
```

Figura 17. Código necesario para comenzar un intento.

Cuando se comienza con el intento se empiezan a recoger datos sobre la electromiografía. Como se puede ver en el código de la figura 18, si los valores de la electromiografía superan el umbral marcado, el personaje se elevará y si no lo supera, el personaje descenderá. Además, existe otra condición para el movimiento del personaje, que es que se encuentre activo, es decir, que se este mostrando por pantalla. Esto evita que el personaje cambie de posición antes de iniciar el juego.

```
if (DataPositive.Max() > 0.080f)
{
    if (player.activeSelf == true)
        player.transform.Translate(0, maxSpeed , 0);
}
else
{
    if (player.activeSelf == true)
        player.transform.Translate(0, maxSpeed , 0);
}
```

Figura 18. Código necesario para el movimiento del personaje.

Cuando se termina un intento, se desarma la pipeline que se había configurado durante el comienzo del intento. Posteriormente, se almacena en una lista en valor máximo de la electromiografía durante el intento y se vacía la lista donde se han almacenado los datos de los sensores, para no mezclar los datos de varios intentos. Todo este proceso se puede ver en el código de la figura 19.

```
RFPipeline.Stop();
RFPipeline.DisarmPipeline();
intentoMaximoBD.Add(DataPositive1.Max());
DataPositive1.Clear();
```

Figura 19. Código necesario para terminar un intento.

Por último, se muestra por pantalla el valor máximo del intento realizado en ese momento. En el último intento se guarda en el fichero .csv mencionado anteriormente el valor máximo y el valor mínimo de los tres intentos.

```

if (trial == 3)
{
    intento3BD.text = "Intento 3: " +
    Math.Round(intentoMaximoBD[2],3).ToString();
    using (StreamWriter channelOutputFile = new
    StreamWriter("./" + fecha + "_" + nombreYApellidos.text +
    "_calibracion"+"".csv"))
    {
        channelOutputFile.WriteLine("Intento maximo: " +
        intentoMaximoBD.Max());
        channelOutputFile.WriteLine("Intento minimo: " +
        intentoMaximoBD.Min());
    }
}

```

Figura 20. Código necesario para crear el fichero csv.

3.3.2 Menú Principal

Cuando se pulsa el botón “Empezar Juego” del menú de calibración, nos lleva al menú principal de *Twynbird*, que como se ha mencionado anteriormente es el nombre del juego.

En este menú se pueden observar diferentes componentes:

1. Título del juego.
2. Caja de texto informativa.
3. Record del juego con la puntuación y el nombre de la persona que lo consiguió.
4. Botón para abrir el menú de configuración.
5. Botón para salir del juego.
6. Botón para encender los sensores. Cuando se pulsa este botón se pasa de la imagen de la figura 21 a la de la figura 22 y se podrá comenzar con el juego.



Figura 21. Menú principal del juego

7. Botón para empezar a jugar.



Figura 22. Menú principal una vez encendidos los sensores.

Cuando se acaba la partida, se muestra la pantalla de la figura 23, en la que aparece un botón para poder volver al menú principal.

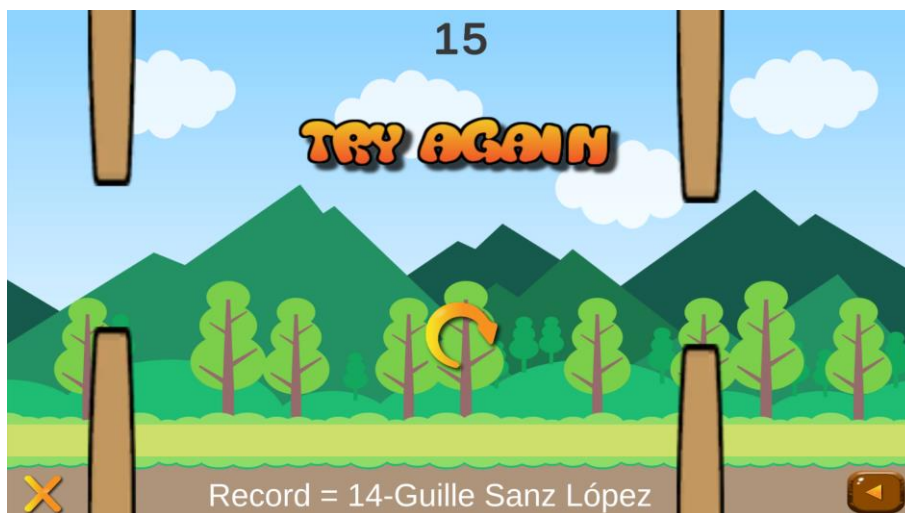


Figura 23. Pantalla cuando no superas un obstáculo.

Código necesario para la implementación del menú principal

Cuando se inicia el juego, lo primero que hay que obtener son los datos almacenados en la memoria del ordenador, que son el récord y el nombre de la persona que lo consiguió. Estos datos tienen que haber sido almacenados en memoria previamente. Para ello utilizamos la clase `PlayerPrefs` de la API de Unity, que permite guardar datos entre sesiones de juego. Se pueden guardar cadenas de caracteres (strings), valores enteros (int) y decimales (float) (Unity Technologies, 2022a)

```
puntajeRecord = PlayerPrefs.GetInt("scoreRecord");
nombreRecord = PlayerPrefs.GetString("nombreRecord");
```

Figura 24. Código para la obtención de los valores almacenados en memoria.

Cuando se pulsa el botón para empezar a jugar se ejecuta el código que está en la figura 25 en la que se ocultan los diferentes elementos que forman parte del menú principal y se

muestran los componentes del juego. También se coloca al personaje en la posición correcta para que no haya ningún problema.

Cuando el paciente empieza a jugar, se ocultan todos los elementos pertenecientes al menú principal y se muestran todos los pertenecientes a la pantalla de juego. También se sitúa al personaje en la posición inicial, por si ha habido algún problema.

```
configuracion.SetActive(false);
player.SetActive(true);
reiniciar.SetActive(true);
score.SetActive(true);
canvasEmpezar.SetActive(false);
player.transform.position = positiony;
```

Figura 25. Código necesario para cambiar a la pantalla de juego.

Cuando el personaje se choca con un obstáculo (si las colisiones están activadas), se para el juego y se muestra la pantalla correspondiente de haber perdido la partida (que contiene un botón de reinicio para volver a empezar), se congela la imagen y se escucha un efecto de sonido de derrota.

```
canvasPerdiste.SetActive(true);
Time.timeScale = 0;
audioPerder.Play();
```

Figura 26. Código necesario para cargar la pantalla de derrota.

Si la puntuación obtenida durante el juego es mayor que la del récord que había en ese momento, se guarda la puntuación con el nombre que se había introducido durante la calibración y se actualiza en el menú.

```
if(ScoreController.puntaje > puntajeRecord)
{
    puntajeRecord = ScoreController.puntaje;
    nombreRecord = PlayerPrefs.GetString("nombre");
    PlayerPrefs.SetInt("scoreRecord", puntajeRecord);
    PlayerPrefs.SetString("nombreRecord", nombreRecord);
}
```

Figura 27. Código necesario para la actualización del récord.

Por último, se destruyen todos los obstáculos creados en la escena hasta ese momento. Para ello se buscan todos los objetos etiquetados como obstáculos y se hace un bucle llamando a la función Destroy().

```

GameObject[] obstacles =
GameObject.FindGameObjectsWithTag("Obstacle");
foreach (GameObject obstacle in obstacles)
    Destroy(obstacle);

```

Figura 28. Código necesario para la destrucción de los obstáculos.

3.3.3 Pantalla de Juego

Cuando se pulsa el botón para empezar a jugar del menú principal, se cambia a la pantalla de juego, la cual podemos ver en la figura 29. En ella se tiene que intentar evitar los obstáculos que se van generando, usando la señal electromiográfica.

Como se puede ver, la pantalla está formada por diferentes elementos:

1. Personaje
2. Obstáculos
3. Puntuación
4. Botón para salir de la aplicación
5. Botón para volver al menú principal
6. Récord

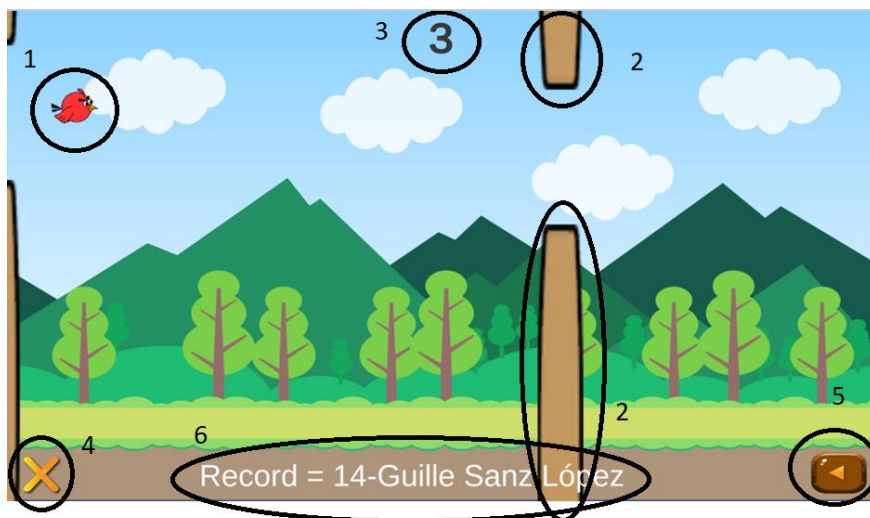


Figura 29. Pantalla de juego.

Código necesario para la implementación de la pantalla de juego

Lo primero de todo es generar los obstáculos. Para crear un obstáculo utilizamos la función generarObstaculo() , la cual crea una instancia de un obstáculo en la posición derecha, a una altura aleatoria entre un rango estudiado (para que el obstáculo salga dentro de la pantalla) y con la rotación del objeto inicial.

```

private void generarObstaculo()
{
    Instantiate(obstaculo, new Vector3(derecha,
        Random.Range(alturamin, alturamax),
        0), obstaculo.transform.rotation);
}

```

Figura 30. Función que permite generar obstáculos.

Para la generación de obstáculos de forma recurrente se utiliza la generación `InvokeRepeating()`, que invoca el método “generarObstaculo” en *time* segundos cada *repeatTime* segundos.

```
InvokeRepeating("generarObstaculo", time, repeatTime);
```

Figura 31. Función que permite generar obstáculos de forma recurrente.

Una vez generados los obstáculos, hay que hacer que estos se muevan y se dirijan hacia el personaje. Para ello modificamos la posición de los obstáculos dependiendo de su velocidad. Si estos han atravesado la pantalla, se procede a eliminarlos para que no ocupen memoria.

```

transform.position += Vector3.left * speed * Time.deltaTime;
if (transform.position.x < -15)
{
    Destroy(gameObject);
}

```

Figura 32. Código necesario para el movimiento de los obstáculos.

Durante el juego, hay que detectar el momento en el que el personaje supera o se choca con un obstáculo. Para ello se usan las funciones `OnTriggerEnter2D()` y `OnCollisionEnter2D()`.

En el caso de superar un obstáculo se llama a la función `OnTriggerEnter2D()` y se diferencian dos casos:

1. Colisiones activadas

En este caso todo el obstáculo pertenece a la misma capa. Los obstáculos están formados por 3 elementos, los dos objetos visibles que son con los que puede chocarse el personaje y un objeto invisible con el trigger activado para detectar el momento en el que se supera un obstáculo. En este caso esta función sólo será llamada cuando se supere un obstáculo.

2. Colisiones desactivadas

En este segundo caso, los dos objetos visibles pertenecen a una capa distinta. En este caso se puede llamar a la función cuando se supere un obstáculo, lo que sumará un punto, o cuando atraviere uno de los obstáculos, lo que restará un punto.

```

private void OnTriggerEnter2D(Collider2D collision)
{
    if (collision.gameObject.layer == 6)
    {
        ScoreController.puntaje++;
        AudioSource.Play();
    }
    if (collision.gameObject.layer == 9)
    {
        ScoreController.puntaje--;
    }
}

```

Figura 33. Función OnTriggerEnter2D().

En el caso de que las colisiones estén activadas, si se choca con un obstáculo, se llamará a la función OnCollisionEnter2D() que significa que has perdido la partida.

```

private void OnCollisionEnter2D(Collision2D collision)
{
    if (collision.gameObject.layer == 6)
        sceneController.Perdiste();
}

```

Figura 34. Función OnCollisionEnter2D().

Para el movimiento del personaje se utiliza un mecanismo similar al minijuego del menú de calibración.

```

if (DataPositive1.Max() > umbral)
{
    if (player.activeSelf == true)
        player.transform.Translate(0, maxSpeed, 0);
}
else
{
    if (player.activeSelf == true)
        player.transform.Translate(0, -maxSpeed, 0);
}

```

Figura 35. Código que controla el movimiento del personaje en el juego.

3.3.4 Menú de configuración

En el menú principal si se pulsa el botón para abrir el menú de configuración, se abre el menú de la figura 36 en el que se pueden cambiar algunas de las funcionalidades que ofrece la aplicación:

1. Umbral

El umbral es el valor que tiene que superar la señal electromiográfica para que el personaje ascienda. Cuanto más alto es este valor más difícil es superar los obstáculos y más fuerza muscular hay que realizar. Se puede ver el valor del umbral actual a la derecha del deslizador correspondiente.

2. Volumen:

Aumenta o disminuye el volumen del juego (música de fondo y efectos de sonido).

3. Velocidad del personaje

Para poder aumentar o disminuir la dificultad del juego, se puede modificar la velocidad del personaje, para que este ascienda o descienda más rápido o más despacio.

4. Activación de las colisiones

Se pueden desactivar las colisiones con los obstáculos para que los pacientes en su etapa inicial de rehabilitación no se frustren ni dejen de jugar. En este modo de juego al chocar con un obstáculo no se acaba el juego, sólo se resta un punto a la puntuación total que se haya conseguido en este momento.

También existe un botón para volver al menú principal del juego.

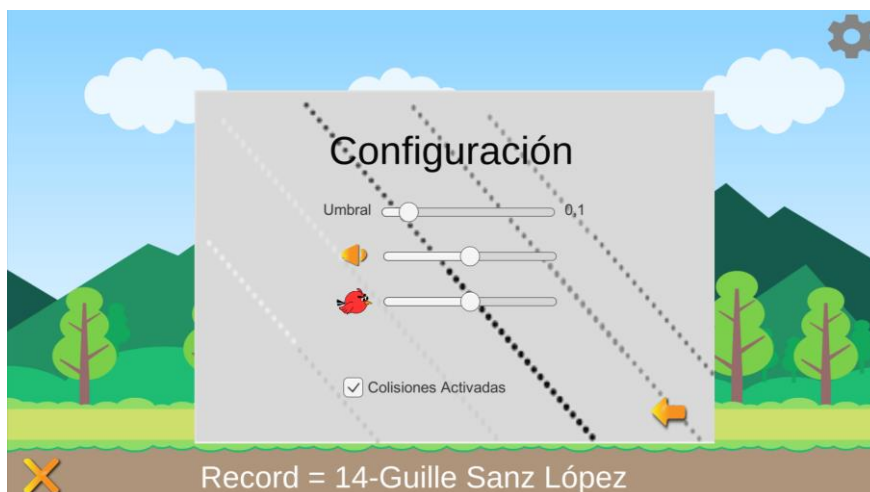


Figura 36. Menú de configuración.

Código necesario para la implementación del menú de configuración

Al comenzar el juego, se ejecuta la porción de código que se muestra en la figura 37. En ella se realizan diferentes acciones que van a ser importantes en el resto del juego:

1. Se asigna a las variables *arriba* y *abajo* los objetos correspondientes al obstáculo superior e inferior
2. Se obtiene la velocidad del personaje y se guarda en la variable *playerSpeedGame*.
3. Se guarda en la variable *audioMuestra* el volumen del juego al iniciarse.
4. Se almacena en la variable *umbrallimit* el valor del umbral al iniciar el juego.
5. Se configuran los obstáculos para que no sean triggers y pertenezcan a la capa número 6. Esto se hace para que el jugador pueda chocarse con los obstáculos y no atravesarlos.

```

1.  arriba = obstaculo.transform.GetChild(0).gameObject;
    abajo = obstaculo.transform.GetChild(1).gameObject;

2.  playerSpeedGame = sensorController.maxSpeed

3.  audioMuestra = audioListener.volume * volume.value;
4.  umbrallimit = sensorController.umbral * umbral.value;

5.  arriba.GetComponent<BoxCollider2D>().isTrigger = false;
    abajo.GetComponent<BoxCollider2D>().isTrigger = false;
    arriba.layer = 6;
    abajo.layer = 6;

```

Figura 37. Código que controla el menú de configuración

Para modificar las variables en tiempo de juego, se ejecuta el código que se muestra en la figura 38. El código realiza la siguiente serie de acciones:

1. Se multiplica el valor inicial por el valor de los deslizadores y el valor del umbral redondeado con 3 decimales se muestra por pantalla. Como el código se encuentra dentro de la función `Update()`, el cambio se realiza de forma prácticamente instantánea.

```

sensorController.maxSpeed = playerSpeedGame * playerSpeed.value;
audioListener.volume = audioMuestra * volume.value;
sensorController.umbral = umbrallimit * umbral.value;
textoUmbral.text =Math.Round(sensorController.umbral,3).ToString();

```

Figura 38. Código que cambia los valores del juego.

2. Si las colisiones están activadas no cambia nada respecto a la función `Start()` pero si estas están desactivadas, se cambian de capa y se activan los triggers para que se puedan atravesar. Se asignan a una capa diferente porque el trigger que hay que atravesar para aumentar la puntuación en caso de superar un obstáculo también

pertenece al obstáculo, pero el personaje no debe chocarse con él.

```
if (toggleColisiones.isOn == true)
{
    arriba.layer = 6;
    abajo.layer = 6;
    arriba.GetComponent<BoxCollider2D>().isTrigger = false;
    abajo.GetComponent<BoxCollider2D>().isTrigger = false;
}
else
{
    arriba.layer = 9;
    abajo.layer = 9;
    arriba.GetComponent<BoxCollider2D>().isTrigger = true;
    abajo.GetComponent<BoxCollider2D>().isTrigger = true;
}
```

Figura 39. Código necesario para cambiar de modo de juego.

3.4 Adquisición de la señal electromiográfica

La adquisición de la señal electromiográfica se realiza gracias a los sensores Trigno Avanti de la compañía Delsys. Esta compañía proporciona una API para poder acoplar los sensores con Unity y poder utilizar los datos extraídos. En este apartado se va a explicar como es este proceso de adquisición de datos una vez ya están conectados los sensores al juego.

Lo primero es configurar la pipeline, que se crea en el proceso de encendido de los sensores y permite el transporte de datos entre los sensores y el juego. Si la pipeline se configura correctamente, la aplicación nos permitirá arrancarla y empezar con la transmisión de datos.

Después se configura el modo de funcionamiento de cada sensor conectado. En este caso se configura con el modo “EMG raw (1778 Hz), skin check (74 Hz), OR 16 bits (74 Hz), +/-11mv, 20-450Hz“. Gracias a este modo se obtiene la señal electromiográfica con una frecuencia de 1778 Hz, además de otra serie de datos que no vamos a utilizar en este juego.

```
foreach (var somecomp in RFPipeline.TrignoRfManager.Components.Where
(x => x.State == SelectionState.Allocated))
{
    string[] sm = somecomp.Configuration.SampleModes;
    somecomp.Configuration.SelectSampleMode("EMG raw (1778 Hz), skin
check (74 Hz), OR 16 bits (74 Hz), +/-11mv, 20-450Hz");
}
```

Figura 40. Código para elegir el modo de funcionamiento de los sensores.

Por último, se realiza la configuración de los canales de entrada y salida para cada sensor (existe un canal de salida del que se leen los datos por cada uno de entrada por donde se escriben los datos).

```
foreach (var component in RFPipeline.TrignoRfManager.Components)
{
    if (component.State == SelectionState.Allocated)
    {
        for (int k = 0; k < component.TrignoChannels.Count; k++)
        {
            var chin = component.TrignoChannels[k];
            var chout = new ChannelTransform(chin.FrameInterval,
chin.SamplesPerFrame, Units);
            TransformManager.AddInputChannel(t0, chin);
            TransformManager.AddOutputChannel(t0, chout);
            outconfig.MapOutputChannel(channelIndex, chout);
            channelIndex++;
        }
    }
}
```

Figura 41. Código para crear los canales para obtener la información.

Cuando existen datos en el canal de entrada, se produce una llamada a la función `CollectionDataReady()`, en la que se vacía la lista que contenía los datos de la señal previos a la nueva entrada de datos. Después guardamos el valor absoluto de todos los datos de la señal en la lista *DataPositive1* .

```
List<List<double>> Data = new List<List<double>>();
List<double> DataPositive1 = new List<double>();

public void CollectionDataReady(object sender, ComponentDataReadyEventArgs e)
{
    if (DataPositive1.Count != 0)
    {
        DataPositive1.Clear();
    }

    foreach (var dato in e.Data.First().Data)
    {
        var absdato = Math.Abs(dato);
        if (absdato<5)
        {
            DataPositive1.Add(absdato);
        }
    }
}
```

Figura 42. Código para añadir los datos necesarios a una lista de datos.

Después, se guardan todos los datos obtenidos de los sensores en la lista *Data* para posteriormente poder representar la señal electromiográfica.

```
for (int j = 0; j < e.Data.Count(); j++)
{
    var channelData = e.Data[j];
    Data[j].AddRange(channelData.Data);
    dataPoints += channelData.Data.Count;
}
}
```

Figura 43. Bucle para añadir los nuevos datos a la lista de datos.

Finalmente, se genera un fichero con todos los datos guardados en la variable *Data[0]* que es la lista correspondiente a los datos que forman el electromiograma.

```

using (StreamWriter channelOutputFile = new StreamWriter("./" +
fecha + "_" + nombreYApellidos + "_EMGgame"+" .csv"))
{
    foreach (var pt in Data[0])
    {
        channelOutputFile.WriteLine(pt.ToString());
    }
}

```

Figura 44. Código para crear el fichero con los datos de la electromiografía generada durante el juego

La señal resultante con la que se trabaja en el juego tiene la siguiente forma:

$$f'(t) = \max |abs(f(t))|$$

Figura 45. Ecuación para obtener la función con la que trabajar en el juego.

4 Pruebas realizadas y resultados

Durante la exposición de este apartado se van a analizar las diferentes pruebas realizadas para probar el correcto funcionamiento del juego. Se han realizado múltiples pruebas a varios sujetos sanos que han trabajado con diferentes grupos musculares. Lo ideal hubiera sido haber realizado un proceso de rehabilitación entero con un paciente con algún problema de movilidad, pero esto no ha sido posible por falta de tiempo. Una vez completada la prueba, los pacientes han realizado una encuesta sobre los conocimientos previos de las tecnologías usadas en el proyecto y algunas preguntas sobre la usabilidad del juego.

4.1 Pruebas realizadas

Aunque se han realizado pruebas a un grupo más amplio de gente, el análisis se va a centrar en un grupo de cinco personas, las que más han probado nuestro juego.

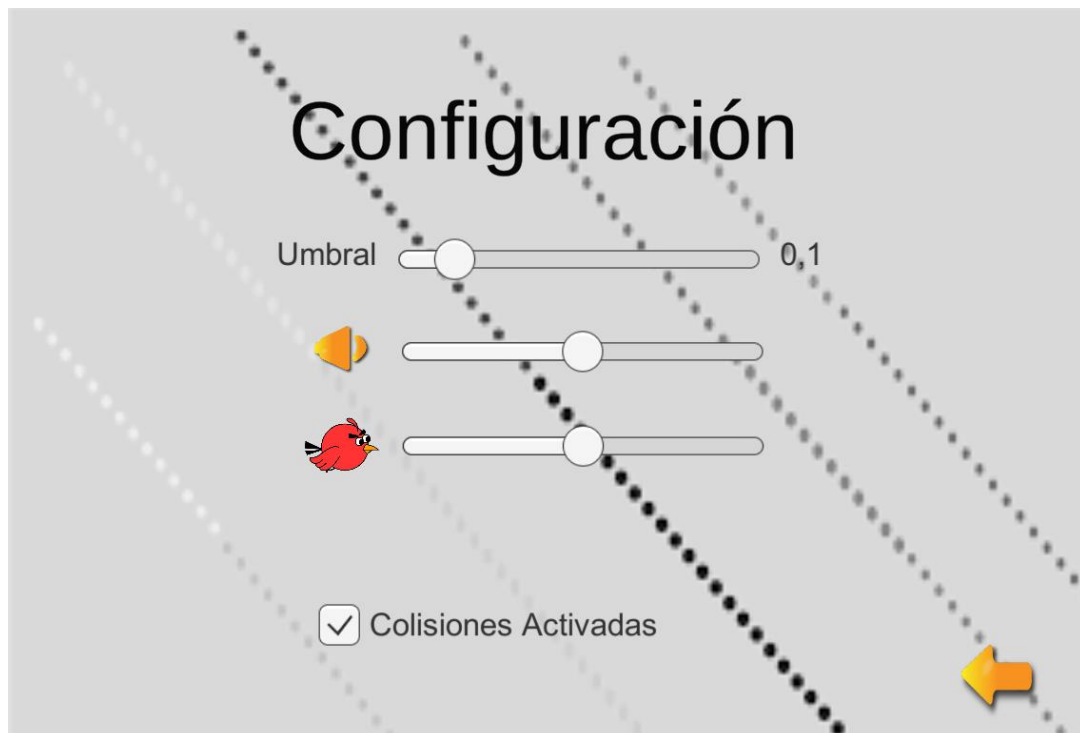


Figura 46. Configuración del menú para las pruebas.

En primer lugar, se ha realizado una prueba en la que el sensor estaba colocado sobre el bíceps del sujeto. En esta prueba, el paciente tenía que intentar mover el personaje con diferentes niveles umbrales. En la tabla 1 se muestran los resultados de esta prueba.

Personas/Umbral	0.100	0.200	0.300	0.400	0.500
Mujer, 22 años	✓	✓	✓	✗	✗
Hombre, 22 años	✓	✓	✓	✓	✗
Hombre, 28 años	✓	✓	✓	✓	✗
Hombre, 29 años	✓	✓	✓	✓	✓
Hombre, 21 años	✓	✓	✓	✓	✗

✓ : Se ha conseguido superar el umbral durante bastante tiempo

✓ : Se ha conseguido superar el umbral durante poco tiempo.

✗ : Apenas se ha conseguido superar el umbral.

Tabla 1. Comparación de resultados entre sujetos (bíceps)

Como se puede ver en la tabla 1, todos han podido superar los niveles de umbral más bajos, mientras que sólo uno ha podido superar, aunque de manera no duradera, el último nivel de umbral. Esto significa que se cumple el objetivo de que el nivel de dificultad se va incrementando a medida que crece el nivel del umbral.

Para saber si realmente esta diferencia entre umbrales significaba un incremento en el esfuerzo muscular se ha realizado un intento con el umbral más bajo e inmediatamente se ha realizado otro intento con el umbral más alto. Gracias a la señal electromiográfica obtenida durante el juego, hemos podido observar que el nivel de esfuerzo realizado para los umbrales más altos es mayor que para el de los umbrales más bajos.

En la figura 47 se observa esta diferencia de forma gráfica. La electromiografía corresponde al hombre de 29 años.

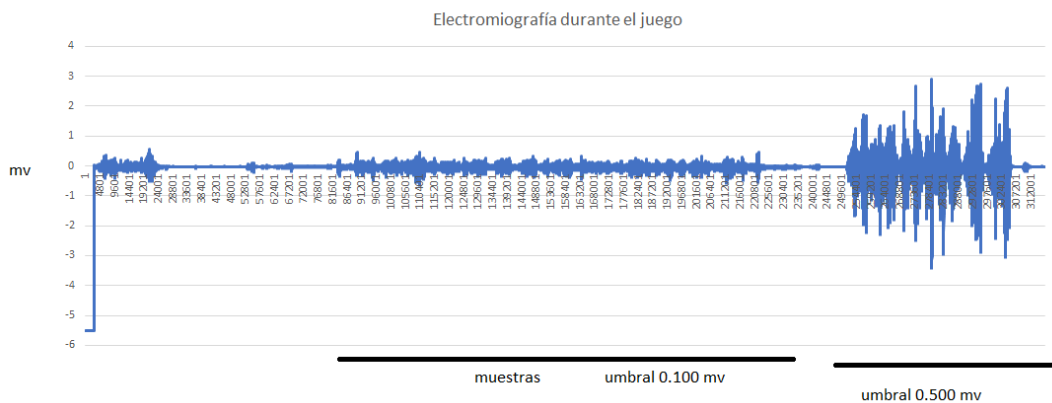


Figura 47. Electromiografía de un hombre de 29 años (bíceps).

Además, también se puede ver la diferencia de fuerza entre los sujetos. La figura 48 muestra la electromiografía de la mujer de 22 años, la que más problemas ha tenido a la hora de manejar el personaje.

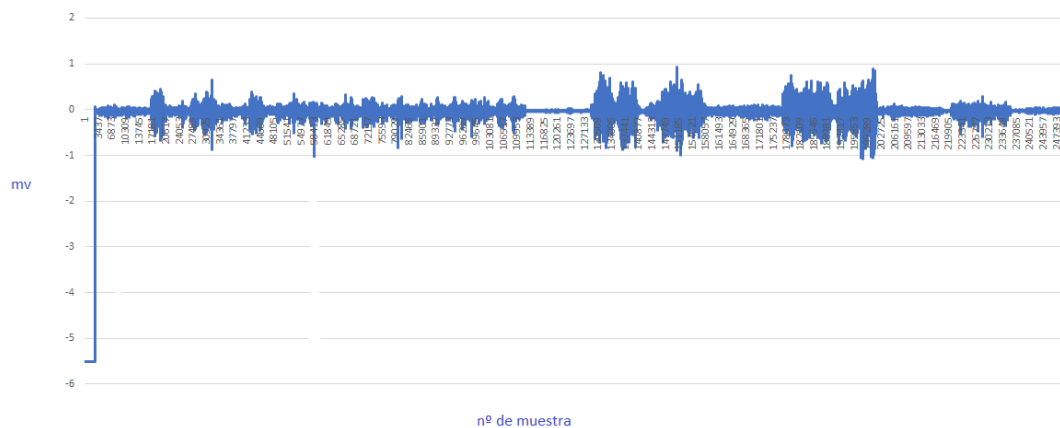


Figura 48. Electromiografía de una mujer de 22 años (bíceps).

Mientras que en la señal de la figura 47 se pueden ver picos de 3 milivoltios, en la figura 48 se ven picos de 1 milivoltio. Esto explica el porqué de la dificultad para superar los obstáculos con los umbrales más altos.

En la segunda prueba realizada, se ha procedido con la misma metodología, pero esta vez situando en el sensor en el tibial anterior de la pierna del paciente.

Mujer, 22 años	✓	✓	✗	✗	✗
Hombre, 22 años	✓	✓	✓	✗	✗
Hombre, 28 años	✓	✓	✓	✗	✗
Hombre, 29 años	✓	✓	✓	✓	✗
Hombre, 21 años	✓	✓	✓	✗	✗

✓ : Se ha conseguido superar el umbral durante bastante tiempo

✓ : Se ha conseguido superar el umbral durante poco tiempo.

✗ : Apenas se ha conseguido superar el umbral.

Tabla 2. Comparación de resultados entre sujetos (tibial)

Para comparar la diferencia que ha podido haber entre los dos músculos se analizará la diferencia entre las señales electromiográficas de la mujer de 22 años.

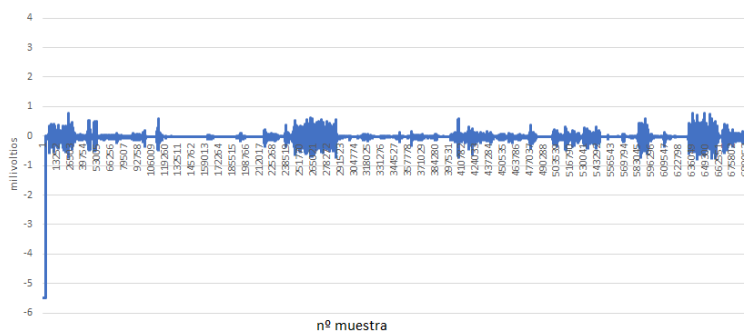


Figura 49. Electromiografía de una mujer de 22 años (tibial).

Esta vez se puede ver en la figura 49 que los picos de la señal del tibial son inferiores a la señal correspondiente al bíceps (figura 48), por lo que la tensión de ambos músculos es diferente. Esto explica las diferencias en la dificultad de superar los obstáculos con los mismos umbrales.

Gracias a la realización de las pruebas, se pueden ver las diferencias que hay entre los diferentes grupos musculares y entre los diferentes sujetos, de ahí de la importancia de la calibración para poder configurar el juego al potencial de cada uno.

4.2 Resultados de la encuesta

Antes de realizar las pruebas, se han preguntado algunas cuestiones básicas sobre las tecnologías que se utilizan en nuestro juego serio. El cuestionario entero se puede encontrar en el anexo A. Se va a comentar las respuestas que nos han dado los encuestados.

La primera pregunta ha sido si alguna vez habían tenido que realizar algún proceso de rehabilitación. Se puede comprobar viendo la figura 50 que un 70% ha respondido que nunca se ha enfrentado a este proceso, y un 30% que sí. De este 30%, todos han respondido que encontraron este proceso duro y aburrido, por lo que se cumple el primero de los propósitos del desarrollo de nuestro videojuego.

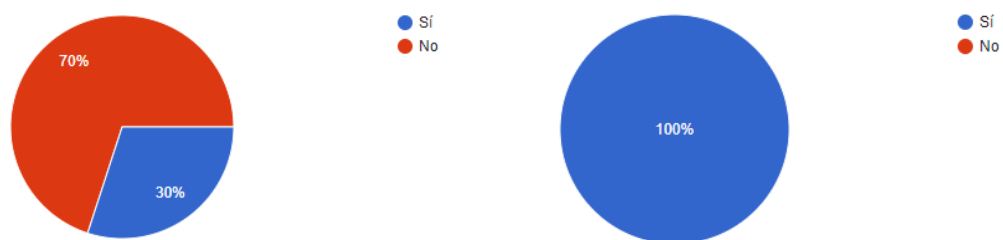


Figura 50. Porcentaje de personas que han realizado un proceso de rehabilitación (izqda.) y de los que sí han participado el porcentaje de los que se han aburrido (dcha).

Uno de los conceptos más innovadores de este proyecto ha sido la utilización de la electromiografía. Nos hemos quedado sorprendidos viendo, en la figura 51, que sólo un 40% de los encuestados conocía la existencia de la electromiografía antes de haber probado el juego.

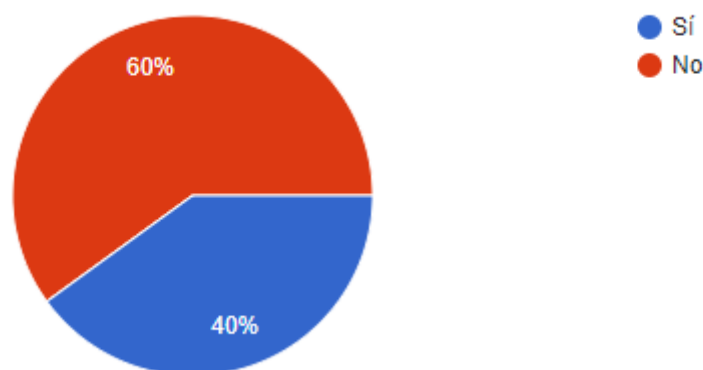


Figura 51. Porcentaje de personas que conocían la electromiografía.

Una vez realizadas las pruebas, se ha procedido a evaluar algunos de los aspectos y funcionalidades que debería tener la aplicación. Sobre el propósito principal de la aplicación que inicialmente era si podía servir como alternativa en los procesos de rehabilitación, el 100% de los encuestados ha respondido de forma afirmativa, como se ve en la figura 52.

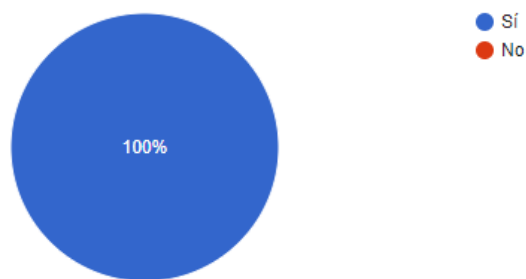


Figura 52. Porcentaje de personas que piensan que el juego puede servir en procesos de rehabilitación

Nuestro trabajo no termina aquí, así que hemos preguntado sobre la opinión hacia los diferentes aspectos de la aplicación, en busca de mejoras.

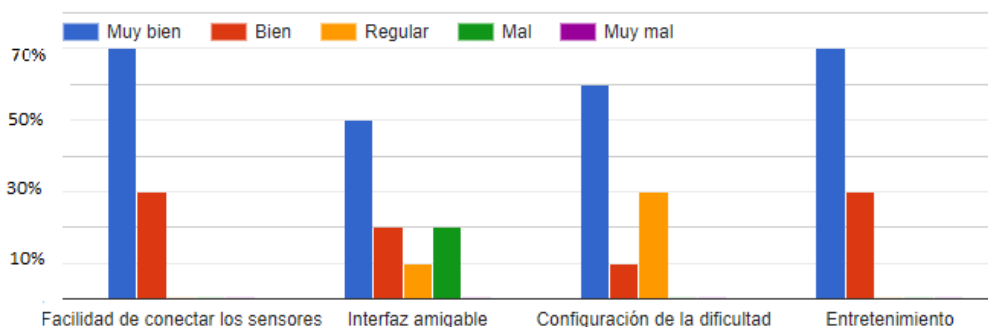


Figura 53. Valoración sobre los diferentes aspectos de la aplicación.

En la figura 53 se pueden ver las diferentes cuestiones valoradas para mejorar nuestro juego. En general, los usuarios han valorado de forma favorable el entretenimiento y la facilidad para la conexión de los sensores. Por el contrario, no han valorado de forma demasiado favorable la amigabilidad de la interfaz y la configuración de la dificultad. Los dos factores que hay que mejorar están relacionados ya que, aunque es verdad que se puede cambiar la dificultad, el menú de configuración no es lo suficientemente claro. También, han afirmado que el modo de juego sin colisiones le quita el sentido al juego porque se elimina el objetivo principal. Si se eliminara este modo de juego, muchos pacientes no podrían llegar a superar ningún obstáculo, por lo que sirve como incentivo para no abandonar el juego y no perder la competitividad.

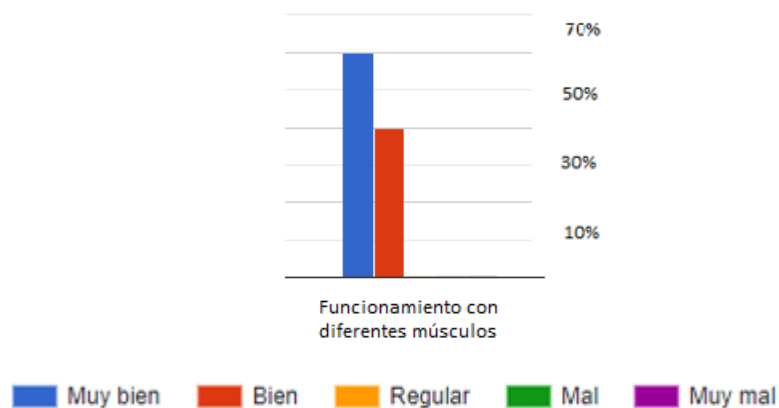


Figura 54. Valoración del funcionamiento con diferentes músculos.

Como punto más favorable, los encuestados valoraron de forma muy favorable la posibilidad de entrenamiento con diferentes grupos musculares. La posibilidad de configurar el umbral nos facilita el trabajo con los músculos con menor fuerza.

Finalmente, preguntamos sobre una valoración numérica general, la cual ha sido de un destacado 8,8 de media, como podemos ver en la figura 55.

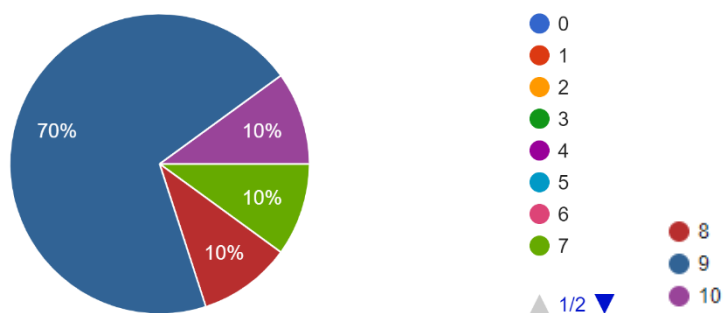


Figura 55. Valoración general de la aplicación.

5 Conclusiones y Líneas Futuras

En este último apartado se van a exponer las diferentes conclusiones que se han extraído durante el desarrollo del Trabajo de Fin de Grado y se va a explicar las vías de mejora y uso de nuestro juego a largo plazo.

5.1 Conclusiones

El uso de los videojuegos y el número de personas que sufren enfermedades que conllevan rehabilitación está en continuo aumento. Por ello se ha pensado en la realización de un juego serio que permita trabajar con diferentes músculos del cuerpo.

Los objetivos iniciales que eran proporcionar a los pacientes una forma alternativa de realizar su proceso de rehabilitación, facilitar este proceso pudiéndose realizar en cualquier lugar y proporcionar información precisa y actualizada del estado de los músculos gracias a la señal electromiográfica se han cumplido con creces.

Para el desarrollo del juego se ha utilizado el motor de videojuegos Unity3D, uno de los más usados en el mercado para el desarrollo de videojuegos. Su gran catálogo de herramientas ha permitido crear diversas escenas de forma sencilla.

El uso de la electromiografía ha permitido el uso de la fuerza muscular para el control del juego. Gracias a esto, se ha podido saber que la tensión entre los diferentes sujetos de pruebas es diferente. También hemos podido conocer que la tensión entre los distintos músculos de un mismo sujeto es diferente.

Las dificultades que hemos sufrido durante la realización del proyecto han sido varias, la compañía proveedora de los sensores tuvo que desarrollar una API distinta no disponible al público para poder conectar los sensores correctamente. Debido a la licencia de uso de los sensores, cada vez que se realizaba una prueba del juego teníamos que construir una nueva versión. Además, nunca habíamos utilizado Unity ni el lenguaje de programación C# por lo que hemos tenido que realizar cursos de aprendizaje.

Finalmente, la realización de la encuesta de uso del juego ha permitido extraer una serie de afirmaciones bastante importantes para un futuro.

La mayor parte de la gente no conoce lo que es la electromiografía. Esto es un problema porque el uso que se le ha dado a esta señal en nuestro proyecto no es el único que tiene. La electromiografía ayuda a detectar múltiples enfermedades y sería importante conocer sus aplicaciones desde edades más tempranas.

Los procesos de rehabilitación son largos y aburridos. Esperamos que la realización de este proyecto permita amenizar la rehabilitación y rebajar su tasa de abandono, además de incrementar su eficacia, como se había planteado en el inicio.

Nuestro juego puede servir como alternativa en los procesos de rehabilitación. Si se consiguen pulir los defectos y mejorar las funcionalidades, se conseguirán todos los objetivos que se habían propuesto al inicio del desarrollo.

5.2 Líneas Futuras

A largo plazo, se tiene mucho margen de mejora. La interfaz en algunos momentos es difícil de interpretar, aunque en gran parte esto se debe al proceso de integración de los sensores de la aplicación.

La interfaz no es amigable, como ya han comentado los usuarios de pruebas en la encuesta de uso del juego. Se mejorará centrando el texto que explica lo que hay que hacer en cada momento. Es cierto que está algo esquinado y hay veces que no se centra la vista en el texto.

Con respecto a las tecnologías usadas, los sensores no son los más baratos del mercado, por lo que limita su uso a centros de rehabilitación o laboratorios de investigación.

En un futuro, sí que vemos posible el uso de nuestro juego en clínicas de rehabilitación, aunque sí que es cierto que sería conveniente pulir algunos defectos vistos y añadir más funcionalidades, no sólo trabajando con la electromiografía, sino combinando su uso con otras metodologías.

Adicionalmente, se desarrollará otra versión del juego, ya que, como retroalimentación proporcionada por terapeutas de un centro de rehabilitación, lo ideal sería que el juego funcionase durante la marcha de un paciente sobre una cinta.

Referencias

- Abt, Clark C. *Serious Games*. New York: Viking, 1970, 176 pp., L.C. 79-83234' (1970) *American Behavioral Scientist*, 14(1), pp. 129–129. doi:10.1177/000276427001400113.
- Bonnechère, B. *et al.* (2016) 'Do Patients Perform Their Exercises at Home and why (not)? A Survey on Patients' Habits during Rehabilitation Exercises', *THE ULUTAS MEDICAL JOURNAL*, 2, p. 1. doi:10.5455/umj.20160210060312.
- Brambilla, C. *et al.* (2021) 'Combined Use of EMG and EEG Techniques for Neuromotor Assessment in Rehabilitative Applications: A Systematic Review', *Sensors*, 21(21). doi:10.3390/s21217014.
- Campanini, I. *et al.* (2020) 'Surface EMG in Clinical Assessment and Neurorehabilitation: Barriers Limiting Its Use', *Frontiers in Neurology*, 11. doi:10.3389/fneur.2020.00934.
- Chowdhury, R.H. *et al.* (2013) 'Surface Electromyography Signal Processing and Classification Techniques', *Sensors*, 13(9), pp. 12431–12466. doi:10.3390/s130912431.
- Collado, J.C. (2018) 'El sector de los videojuegos en España: impacto económico y escenarios fiscales', *Asociación Española de Videojuegos*, January. Available at: <http://www.aevi.org.es/la-industria-del-videojuego/en-espana/> (Accessed: 1 June 2022).
- Correa-Figueroa, J.L. *et al.* (2016) 'Sistema de Adquisición de Señales SEMG para la Detección de Fatiga Muscular', *Revista mexicana de ingeniería biomedical*, 37, pp. 17–27.
- De Gloria, A., Bellotti, F. and Berta, R. (2014) 'Serious Games for education and training', *International Journal of Serious Games*, 1(1). doi:10.17083/ijsg.v1i1.11.
- De Luca, C.J. (2002) 'SURFACE ELECTROMYOGRAPHY: DETECTION AND RECORDING', p. 10.
- Delsys (2022a) *Delsys API Documentation*. Available at: <http://data.delsys.com/DelsysServicePortal/api/web-api/index.html> (Accessed: 21 June 2022).
- Delsys (2022b) 'Trigno® Avanti Sensor', *Delsys*. Available at: <https://delsys.com/trigno-avanti/> (Accessed: 7 June 2022).

- Dias, L.P.S., Barbosa, J.L.V. and Vianna, H.D. (2018) ‘Gamification and serious games in depression care: A systematic mapping study’, *Telematics and Informatics*, 35(1), pp. 213–224. doi:<https://doi.org/10.1016/j.tele.2017.11.002>.
- Disselhorst-Klug, C., Schmitz-Rode, T. and Rau, G. (2009) ‘Surface electromyography and muscle force: Limits in sEMG–force relationship and new approaches for applications’, *Clinical Biomechanics*, 24(3), pp. 225–235. doi:<https://doi.org/10.1016/j.clinbiomech.2008.08.003>.
- Doumas, I. *et al.* (2021) ‘Serious games for upper limb rehabilitation after stroke: a meta-analysis’, *Journal of NeuroEngineering and Rehabilitation*, 18(1), p. 100. doi:[10.1186/s12984-021-00889-1](https://doi.org/10.1186/s12984-021-00889-1).
- Flores, E. *et al.* (2008) ‘Improving Patient Motivation in Game Development for Motor Deficit Rehabilitation’, in *Proceedings of the 2008 International Conference on Advances in Computer Entertainment Technology*. New York, NY, USA: Association for Computing Machinery (ACE ’08), pp. 381–384. doi:[10.1145/1501750.1501839](https://doi.org/10.1145/1501750.1501839).
- Frigo, C. and Crenna, P. (2009) ‘Multichannel SEMG in clinical gait analysis: A review and state-of-the-art’, *Clinical Biomechanics*, 24(3), pp. 236–245. doi:<https://doi.org/10.1016/j.clinbiomech.2008.07.012>.
- Gilbert, N. (2021) *Number of Gamers Worldwide 2022/2023: Demographics, Statistics, and Predictions*, *Financesonline.com*. Available at: <https://financesonline.com/number-of-gamers-worldwide/> (Accessed: 1 June 2022).
- Gomila, J.G. (2022) *Curso de desarrollo de videojuegos con Unity 2021*, *Udemy*. Available at: <https://www.udemy.com/course/draft/4151304/> (Accessed: 7 June 2022).
- Guzmán-Muñoz, E. and Méndez-Rebolledo, G. (2018) ‘Electromiografía en las Ciencias de la Rehabilitación’, *Revista Salud Uninorte*, 34(3), pp. 753–765.
- Hocine, N., Gouaïch, A. and Cerri, S.A. (2014) ‘Dynamic Difficulty Adaptation in Serious Games for Motor Rehabilitation’, in Göbel, S. and Wiemeyer, J. (eds) *Games for Training, Education, Health and Sports*. Cham: Springer International Publishing, pp. 115–128.
- Kotov-Smolenskiy, A.M. *et al.* (2021) ‘Surface EMG: Applicability in the Motion Analysis and Opportunities for Practical Rehabilitation’, *Human Physiology*, 47(2), pp. 237–247. doi:[10.1134/S0362119721020043](https://doi.org/10.1134/S0362119721020043).
- Laamarti, F., Eid, M. and El Saddik, A. (2014) ‘An Overview of Serious Games’, *International Journal of Computer Games Technology*, 2014. doi:[10.1155/2014/358152](https://doi.org/10.1155/2014/358152).

- mDurance Studio (2021) '▷ ¿Es fiable la electromiografía de superficie? ¿Tiene contraindicaciones?', *mDurance estudio*, 14 June. Available at: <https://blog.mdurance.eu/academia/fiabilidad-electromiografia-de-superficie/> (Accessed: 14 June 2022).
- Mesin, L. (2020) 'Crosstalk in surface electromyogram: literature review and some insights', *Physical and Engineering Sciences in Medicine*, 43(2), pp. 481–492. doi:10.1007/s13246-020-00868-1.
- Microsoft (2022) *Visual Studio: IDE y Editor de código para desarrolladores de software y Teams, Visual Studio*. Available at: <https://visualstudio.microsoft.com/es/> (Accessed: 7 June 2022).
- Moreland, J.D., Thomson, M.A. and Fuoco, A.R. (1998) 'Electromyographic biofeedback to improve lower extremity function after stroke: a meta-analysis.', *Archives of physical medicine and rehabilitation*, 79(2), pp. 134–140. doi:10.1016/s0003-9993(98)90289-1.
- Najarian, K. and Splinter, R. (2012) *Biomedical Signal and Image Processing*. 2nd edn. Boca Raton: CRC Press. doi:10.1201/b11978.
- Norali, A.N., Mat Som, M.H. and Kangar-arau, J. (2009) 'Surface electromyography signal processing and application: A review', in *Proceedings of the International Conference on Man-Machine Systems (ICoMMS)*, pp. 1A4-1.
- Osyren (2020) 'Juegos Educativos y Serious Games', *OSYREN - Jugar es aprender*, 21 May. Available at: <https://osyren.com/juego-educativo-y-serious-games> (Accessed: 27 May 2022).
- Rangayyan, R.M. (2015) *Biomedical signal analysis*. John Wiley & Sons.
- Reaz, M.B.I., Hussain, M.S. and Mohd-Yasin, F. (2006) 'Techniques of EMG signal analysis: detection, processing, classification and applications', *Biological Procedures Online*, 8(1), pp. 11–35. doi:10.1251/bpo115.
- Rego, P.A., Moreira, P. and Reis, L. (2010) 'Serious Games for Rehabilitation: A survey and a classification towards a taxonomy', in *5th Iberian Conference on Information Systems and Technologies (CISTI)*, pp. 1–6.
- Rodríguez-Tapia, B. *et al.* (2020) 'Myoelectric Interfaces and Related Applications: Current State of EMG Signal Processing—A Systematic Review', *IEEE Access*, 8, pp. 7792–7805. doi:10.1109/ACCESS.2019.2963881.
- Steele, K.M., Papazian, C. and Feldner, H.A. (2020) 'Muscle Activity After Stroke: Perspectives on Deploying Surface Electromyography in Acute Care', *Frontiers in neurology*, 11, pp. 576757–576757. doi:10.3389/fneur.2020.576757.

- Stegeman, D. and Hermens, H. (2007) ‘Standards for surface electromyography: The European project Surface EMG for non-invasive assessment of muscles (SENIAM)’, 1.
- Talib, I. *et al.* (2019) ‘A review on crosstalk in myographic signals’, *European Journal of Applied Physiology*, 119(1), pp. 9–28. doi:10.1007/s00421-018-3994-9.
- Unity Technologies, (2020) *Unity - Manual: Unity’s interface*. Available at: <https://docs.unity.cn/2020.1/Documentation/Manual/UsingTheEditor.html> (Accessed: 13 June 2022).
- Unity Technologies (2022a) *Unity - Scripting API*: Available at: <https://docs.unity3d.com/ScriptReference/index.html> (Accessed: 15 June 2022).
- Unity Technologies (2022b) *Unity Hub – Administra las versiones de Editor y colabora con otros creadores | Unity*. Available at: <https://unity.com/es/unity-hub> (Accessed: 9 June 2022).
- Unity Technologies (2022c) *Plataforma de desarrollo en tiempo real de Unity | Motor de VR y AR en 3D y 2D*. Available at: <https://unity.com/es> (Accessed: 7 June 2022).
- Wagner, B. (2022) *Un paseo por C#: información general*. Available at: <https://docs.microsoft.com/es-es/dotnet/csharp/tour-of-csharp/> (Accessed: 8 June 2022).
- Woodford, H. and Price, C. (2007) ‘EMG biofeedback for the recovery of motor function after stroke.’, *The Cochrane database of systematic reviews*, 2007(2), p. CD004585. doi:10.1002/14651858.CD004585.pub2.
- Wouters, P. *et al.* (2013) ‘A Meta-Analysis of the Cognitive and Motivational Effects of Serious Games’, *Journal of Educational Psychology*, 105, p. 249. doi:10.1037/a0031311.

ANEXO A. Cuestionario de uso

Encuesta sobre Twynbird

Twynbird es un juego serio orientado a la rehabilitación muscular cuyo propósito es amenizar estos procesos. Se basa en la adquisición de la electromiografía (señal que mide la actividad muscular y puede ayudar a detectar ciertos trastornos).

Correo *

Correo válido

Este formulario registra los correos. [Cambiar configuración](#)

Nombre y Apellidos

*

Texto de respuesta corta

Rango de edad *

- Menos de 18 años
- 18-30
- 31-45
- 46-60
- 61-75
- más de 75 años

¿Has tenido que realizar algún proceso de rehabilitación o de recuperación de movilidad? *

- Sí
- No

Solo si has respondido Sí a la pregunta anterior, ¿Has encontrado el proceso aburrido?

Sí

No

¿Habías oído hablar de la electromiografía antes? *

Sí

No

Valora los siguientes aspectos *

	Muy bien	Bien	Regular	Mal	Muy mal
Facilidad de co...	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Interfaz amigab...	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Configuración ...	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Entretenimiento	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Funcionamient...	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Valoración de la aplicación *

1. 0

2. 1

3. 2

4. 3

5. 4

6. 5

7. 6

8. 7

9. 8

10. 9

11. 10

¿Qué es lo que más te ha gustado? *

Texto de respuesta larga

¿Qué es lo que menos te ha gustado? *

Texto de respuesta larga

¿Crees que Twynbird puede ser útil en procesos de rehabilitación? *

Sí

No

Algún aspecto a mejorar sobre Twynbird

Texto de respuesta larga

ANEXO B. Presupuesto del proyecto

En este anexo se van a explicar los costes originados durante el desarrollo del proyecto. Se puede dividir el coste del proyecto en dos partes. Primero, se van a agrupar los costes correspondientes al hardware y por otro lado los costes de desarrollo software.

Por la parte de los componentes hardware se puede ver el coste en la tabla 3.

COMPONENTE	PRECIO	PRECIO (IVA)
ORDENADOR PORTÁTIL HP PAVILION POWER LAPTOP 15-CB0XX	867.77 €	1050 €
TRIGNO LITE SYSTEM	4132.23 €	5000 €
	5000 €	6050 €

Tabla 3. Coste componentes hardware.

La parte del desarrollo software se puede ver en la tabla 4.

TRABAJO	TIEMPO	COSTE
CURSO DE FORMACIÓN	60 horas	13 €
DESARROLLO DE LA APLICACIÓN	300 horas	4500 €

Tabla 4. Coste desarrollo software.

En los costes se ha supuesto que el salario medio de un ingeniero software junior es de 15 euros brutos por hora.

Por lo tanto, el presupuesto final del proyecto combinando los costes es de **10550 €**