



UNIVERSIDAD DE VALLADOLID
E.T.S.I. TELECOMUNICACIÓN

TRABAJO FIN DE GRADO

GRADO EN INGENIERÍA DE TECNOLOGÍAS DE
TELECOMUNICACIÓN

**Juego serio con sensores inerciales y de
electromiografía para rehabilitación física**

Autor:

Dña. Cristina Violeta Rodríguez Gutiérrez

Tutor:

Dr. D. Mario Martínez Zarzuela

Valladolid, 23 de septiembre de 2024

TÍTULO: **Juego serio con sensores inerciales
y de electromiografía para
rehabilitación física**

AUTOR: **Dña. Cristina Violeta Rodríguez
Gutiérrez**

TUTOR: **Dr. D. Mario Martínez Zarzuela**

DEPARTAMENTO: **Teoría de la Señal y
Comunicaciones e Ingeniería
Telemática**

TRIBUNAL

PRESIDENTE: **Miriam Antón Rodríguez**

SECRETARIO: **Mario Martínez Zarzuela**

VOCAL: **David González Ortega**

SUPLENTE 1: **Carlos Gómez Peña**

SUPLENTE 2: **Jesús Poza Crespo**

Agradecimientos

A mi tutor, Mario Martínez Zarzuela, por ofrecerme este proyecto y por su ayuda y guía durante el desarrollo del mismo.

A Javier Cantalapiedra por su trabajo previo que me ha servido como base para este proyecto.

A Carmen por estar siempre dispuesta a ayudar y aconsejarme en base a su propio trabajo.

A Javier González y por prestarnos su cinta de caminar, lo que nos simplificó enormemente la vida.

A mi familia y a D. por su apoyo y su confianza, incluso cuando yo misma no confiaba.

A mis amigos, especialmente a mis compañeros de grado, gracias a los cuales no dejé la carrera en tercero.

Resumen

El uso de biofeedback con electromiografía (EMG) está aumentando recientemente en terapias de rehabilitación. Este tipo de retroalimentación ofrece muchos beneficios tanto a pacientes como a terapeutas al devolver datos objetivos sobre las activaciones musculares a tiempo real.

Otra técnica que se utiliza cada vez más son los juegos serios, que además de hacer más amigable el proceso de la rehabilitación mejoran sus resultados.

La combinación de estas dos técnicas ha dado lugar a una nueva forma de rehabilitación que, aunque aún no está demasiado extendida, cada vez está tomando más fuerza debido a sus múltiples beneficios frente a terapias convencionales.

Este proyecto ha consistido en el desarrollo de un juego serio que, haciendo uso de biofeedback EMG, sirve para reeducar el ciclo de la marcha de pacientes con trastornos neuromusculares.

Para ello se recogen con dos sensores colocados en la pierna con patrón patológico datos inerciales (IMU) para caracterizar el ciclo de la marcha y EMG para analizar las activaciones de los músculos estudiados.

Todos estos datos se van mostrando por pantalla para que tanto el paciente como el terapeuta puedan ver a tiempo real las activaciones musculares durante la marcha.

Además, el juego incluye marcadores que indican el comportamiento deseado por parte del paciente. En todo momento aparecen señalados los momentos de la marcha donde se espera la activación de cada uno de los músculos y los momentos donde se espera que estén relajados.

Por último, se dispone también de un sistema de puntos que recompensa o penaliza al paciente en función de si se mantiene o no dentro del comportamiento esperado. Este sistema de puntos está apoyado por estímulos sonoros que incrementan la retroalimentación del juego, reforzando los comportamientos deseados y alertando de los que no lo son.

Palabras clave

Electromiografía (EMG), Ciclo de la marcha, Juego Serio, Rehabilitación, Trastorno neuromuscular

Abstract

The use of electromyography (EMG) in biofeedback is recently increasing in rehabilitation therapies. This type of feedback offers many benefits to both patients and therapists by returning objective data about muscle activations in real time.

Another technique that is increasingly being used is serious games, which in addition to making the rehabilitation process more friendly, improves its results.

The combination of these two techniques has given rise to a new form of rehabilitation that, although it is not yet very widespread, is increasingly gaining strength due to its multiple benefits compared to conventional therapies.

This project has consisted of the development of a serious game that, using EMG biofeedback, serves to reeducate the gait cycle of patients with neuromuscular disorders.

To this end, inertial (IMU) data are collected with two sensors placed on the leg with a pathological pattern to characterize the gait cycle and EMG to analyze the activations of the muscles studied.

All this data is displayed on the screen so that both the patient and the therapist can see the muscle activations during walking in real time.

Additionally, the game includes markers that indicate the patient's desired behavior. At all times, the moments of the gait where the activation of each of the muscles is expected and the moments where they are expected to be relaxed are indicated.

Finally, there is also a score system that rewards or penalizes the patient depending on whether they maintain the expected behavior or not. This score system is supported by sound stimuli that increase game feedback, reinforcing desired behaviors and alerting of those that are not.

Key words

Electromyography (EMG), Gait cycle, Serious games, Rehabilitation, Neuromuscular disorders.

ÍNDICE GENERAL

1.	Introducción.....	15
1.1.	Contexto y motivación.....	15
1.2.	Hipótesis y objetivos.....	20
1.3.	Materiales.....	21
1.3.1.	Hardware.....	21
1.3.2.	Software.....	22
1.4.	Fases del proyecto.....	25
2.	Fundamentación teórica.....	27
2.1.	EMG.....	27
2.2.	Ciclo de la marcha.....	32
2.3.	Actividad EMG en la marcha.....	35
3.	Métodos.....	39
3.1.	Mejora en la calibración del juego anterior.....	39
3.2.	Ideando el nuevo juego.....	42
3.3.	Nuevas necesidades en la recogida de datos.....	43
3.4.	Juego concepto final.....	46
3.4.1	Calibración.....	46
3.4.1.1	Primera etapa: Calibración estática.....	46
3.4.1.2	Segunda etapa: Calibración durante la marcha.....	48
3.4.2	Menú de opciones.....	53
3.4.2.1	Selección del modo de juego.....	54
3.4.2.2	Control de penalizaciones.....	54
3.4.2.3	Ajuste del umbral de activación.....	54
3.4.3	Pantalla de juego.....	56
3.4.3.1	Representación del momento de la marcha.....	57
3.4.3.2	Representación de activaciones musculares durante la marcha.....	61
3.4.3.3	Barras de activación acumulada.....	65
3.4.3.4	Recogida de recompensas y penalizaciones.....	67
3.4.4	Cámara.....	68
3.4.5	Datos exportados.....	70
4.	Pruebas y modificaciones.....	71
4.1	Reconocimiento en tiempo real del swing y el stand.....	71
4.2	Comprobación tiempos de paso y balanceo.....	72
4.3	Ajuste del valor del umbral.....	76

5. Conclusiones	79
5.1 Líneas futuras	80
6. Referencias.....	83

ÍNDICE DE FRAGMENTOS DE CÓDIGO

Fragmento de código 1. Cálculo de RMS de datos recogidos en el programa anterior	39
Fragmento de código 2. Cálculo de umbral (código simplificado).....	41
Fragmento de código 3. Obtención de datos de los sensores	43
Fragmento de código 4. Código para evitar el efecto del buffer de los sensores (código simplificado).....	45
Fragmento de código 5. Obtención de RMS de datos EMG en el juego nuevo.....	47
Fragmento de código 6. Código para la finalización de la primera fase de calibración	47
Fragmento de código 7. Obtención posibles tiempos de paso	48
Fragmento de código 8. Obtención tiempo de paso final	49
Fragmento de código 9. Obtención de tiempos de swing posibles	50
Fragmento de código 10. Obtención de datos EMG durante los periodos de activación en la marcha	52
Fragmento de código 11. Cálculo de umbrales EMG finales.....	52
Fragmento de código 12. Desplazamiento del fondo	53
Fragmento de código 13. Código para modificar el umbral EMG desde el menú de opciones.....	55
Fragmento de código 14. Paso a la pantalla de juego según el modo escogido.....	56
Fragmento de código 15. Ajuste de la sección rellena del círculo que representa el swing .	57
Fragmento de código 16. Fragmentación de la marcha a tiempo real	59
Fragmento de código 17. Representación gráfica del estado de la marcha a tiempo real ...	61
Fragmento de código 18. Desplazamiento del pájaro animado	61
Fragmento de código 19. Comienzo de nueva activación	65
Fragmento de código 20. Comprobación activación acumulada.....	68
Fragmento de código 21. Función AddHoja()	68
Fragmento de código 22. Configuración de cámara.....	69
Fragmento de código 23. Código para exportar a .csv	70
Fragmento de código 24. Segmentación de la marcha fuera del juego	73

ÍNDICE DE TABLAS

Tabla 1. Tabla comparativa tiempos de paso.....	74
Tabla 2. Tabla comparativa tiempos de balanceo	75
Tabla 3. Prueba umbrales 1	76
Tabla 4. Prueba umbrales 2	77
Tabla 5. Prueba umbrales 3	77
Tabla 6. Equivalencias entre X y sensibilidad en el juego.....	77

ÍNDICE DE FIGURAS

Figura 1. Componentes del sistema Trigno Avanti Lite de Delsys (Trigno® Lite System - Delsys Europe, s. f.).....	21
Figura 2. Interfaz de Trigno Discover	23
Figura 3. Interfaz de EMG Works Analysis.....	24
Figura 4. Representación circular del ciclo de la marcha (Vaughan & Davis, 1999).....	33
Figura 5. Velocidad angular en x medida desde un sensor colocado en el tibial	34
Figura 6. Puntos clave de la marcha sobre la señal de velocidad angular en x (Fraccaro et al., 2014)	34
Figura 7. Activaciones de diferentes músculos durante la marcha (Vaughan & Davis, 1999)	37
Figura 8. Señal EMG prueba original	40
Figura 9. Señal EMG prueba tras restar offset.....	40
Figura 10. Señal EMG prueba tras rectificación	40
Figura 11. Señal EMG prueba tras RMS	41
Figura 12. Ejes sensor Delsys (Trigno Wireless Biofeedback System User's Guide, s. f.)...	44
Figura 13. Primera fase de la calibración	46
Figura 14. Comparativa entre velocidad angular eje X (GYROX) y aceleración eje Z (ACCZ).....	48
Figura 15. División entre swing (1) y stand (0) a partir de los pasos por cero de la señal GYROX	50
Figura 16. Obtención del tiempo de swing final	51
Figura 17. Representación del error al tomar el paso por cero como división entre swing y stand.....	51
Figura 18. Menú de opciones	53
Figura 19. Selección del modo de juego	54
Figura 20. Selector de penalizaciones activadas o desactivadas	54
Figura 21. Control deslizante de la sensibilidad EMG	54
Figura 23. Vista general de la pantalla de juego.....	56
Figura 24. Representación del ciclo de la marcha en el juego	57
Figura 25. Ajustes en Unity para obtener figuras rellenables de forma radial.....	57
Figura 26. Diagrama de flujo del paso de una fase a otra de la marcha.....	58
Figura 27. Representación de las zonas de activación esperada en función del modo de juego.....	62
Figura 28. Círculos de activación durante el juego.....	63
Figura 29. Diagrama de flujo de tipos de activación y sus respuestas.....	64
Figura 30. Barras de activación acumulada	65
Figura 31. Activaciones en la marcha del tibial anterior (TA) y el gemelo (GL).	66
Figura 32. Configuración de cámara	69
Figura 33. Comparativa entre indicador y fase real de la marcha en video	71
Figura 34. Mid-Swing de cada paso en la señal de velocidad angular en X.....	73
Figura 35. Nueva señal producto de la combinación de todos los pasos.....	74

1. Introducción

1.1. Contexto y motivación

Trastornos de la marcha

La marcha humana necesita para su correcta ejecución que el sistema nervioso, el esquelético y el muscular funcionen de forma conjunta. Si el funcionamiento de alguno de estos sistemas esta alterado se producen los conocidos como trastornos de la marcha.

Hay una gran cantidad de patologías que pueden afectar a la marcha, pudiendo clasificarlas según su etiología en: metabólicas; farmacológicas; neurológicas; esqueléticas y musculares (Lorena Cerda, 2014).

Este proyecto se centrará en los desórdenes de la marcha producidos por trastornos neuromusculares, debido a su gran incidencia en la sociedad actual. Según la Sociedad Española de Neurología “Más de 60.000 personas padecen una enfermedad neuromuscular en España y más del 50% de los casos, la enfermedad debuta en la infancia”. El porcentaje en adultos y ancianos puede ser mayor debido a las enfermedades neurodegenerativas como el ELA y a otras patologías como la esclerosis múltiple o las neuropatías periféricas más frecuentes en ese grupo etario. (Pérez Menéndez, s. f.) En niños, por otro lado, lo más común son las distrofias musculares y otros trastornos genéticos y/o hereditarios (Muñoz Saavedra, 2018).

Juegos serios

Los juegos serios son videojuegos, o al menos programas que se basan en las mismas mecánicas y tecnologías, que buscan algún objetivo diferente al simple entretenimiento. Este objetivo puede ser desde que el jugador aprenda alguna nueva habilidad a diagnosticar problemas de salud. Debido a su amplio abanico de posibles utilidades muchos sectores se han interesado e invertido en ellos, utilizándose hoy en día juegos serios en campos como la educación, las organizaciones militares y gubernamentales, la empresa privada y, el que motiva este proyecto, la salud, concretamente el ámbito de la rehabilitación (Susi et al., 2015).

Los puntos más importantes de un juego serio son los siguientes (Garris et al., 2002; Michael, 2006):

- **Distracción:** Centrando la atención en el juego para el jugador el objetivo subyacente, que de primeras podría ser poco atractivo, o incluso desagradable, queda en segundo plano, haciendo la experiencia mucho más agradable y, en el mejor de los casos, deseable.
- **Simulación:** Los juegos serios están diseñados para simular, de forma más o menos abstractas, situaciones del mundo real para poder evaluar, aprender o practicar las respuestas del jugador de forma segura y controlada.

- **Feedback:** Todos juegos proporcionan realimentación, ya sea a tiempo real mientras se juega o periódicamente, por ejemplo, al terminar un nivel. Estos datos y estímulos, además de motivar al jugador porque fomentan la interactividad, le permiten ajustar su comportamiento para mejorar sus resultados y aprenderlo o practicar esta manera correcta de actuación en el proceso.
- **Motivación Intrínseca:** Tanto los juegos serios, como los demás videojuegos, proporcionan motivación extrínseca para captar al jugador en un principio, pero es la intrínseca la que suele conseguir atraparlos en el tiempo. Para ello se motiva al usuario a través de un ciclo de propuesta, acción y feedback repetitivo que fomenta las ganas del jugador de repetir dicho ciclo para obtener mejores resultados, o para enfrentarse cada vez a propuestas más complicadas, y sentirse satisfechos al superarlas.

Los juegos serios en la rehabilitación

Muchas enfermedades y lesiones requieren de rehabilitación para su correcta recuperación. Particularmente en este proyecto cabe destacar que mediante esta se puede conseguir mejorar los patrones de la marcha de pacientes patológicos. Pero la rehabilitación no es habitualmente un proceso rápido ni amigable, siendo muchas veces incómodo o aburrido para el paciente. Esto puede suponer un problema grave, ya que si el paciente se desmotiva puede llegar hasta a dejar la rehabilitación, lo que repercutiría negativamente en su salud.

Para ayudar con esta problemática se comenzaron a utilizar juegos serios durante el proceso de rehabilitación. Estos juegos persiguen el objetivo último de que el jugador realice los ejercicios necesarios para su rehabilitación, ya sea física o cognitiva, pero disfrazan estos ejercicios repetitivos y aburridos para hacer más amena la experiencia del paciente. Además, como se comentaba anteriormente, los juegos serios se preocupan por que el jugador desarrolle una motivación intrínseca, lo que ayuda a conseguir que el paciente quiera seguir jugando y no abandone la terapia.

Además de motivar al paciente y conseguir que se implique de forma más activa en el tratamiento, el uso de este tipo de juegos tiene otras ventajas:

- **Mejora del aprendizaje:** Según datos recogidos por psicólogos, el uso de juegos durante la terapia fomenta la plasticidad neuronal, lo que fomenta la retención a largo plazo de los mecanismos practicados durante el juego (Lohse et al., 2013a).
- **Feedback a tiempo real:** Muchos juegos utilizados en rehabilitación proporcionan al jugador datos de su rendimiento a tiempo real, lo que le permite modificar su comportamiento para acercarse más al recompensado. De esta manera el progreso es más rápido y se detectan rápidamente las acciones que pueden resultar perjudiciales, por lo que pueden evitarse lesiones o daños provocados por la mala realización de los ejercicios (Lohse et al., 2013a).

- **Seguimiento por parte del terapeuta:** Los datos recogidos por estos juegos, además de ayudar al jugador a realizar correctamente los ejercicios, permite a los terapeutas analizar el rendimiento del paciente y la eficacia de la rehabilitación. De esta manera los terapeutas podrán modificar los ejercicios propuestos para que los pacientes progresen de forma adecuada en la terapia sin llegar a sentirse superados (Herhkovitz, 2024).
- **Reducción de la ansiedad:** En ocasiones, sobre todo en el caso de niños, la rehabilitación supone una situación estresante. El uso de este tipo de juegos puede ayudar a distraer al paciente y ayudarlo a dejar de lado sus reticencias, haciendo la experiencia mucho más agradable y consiguiendo mejores rendimientos (Abd-alrazaq et al., 2022).

En conclusión, los juegos serios tienen, además de un beneficio físico evidente, un impacto positivo a nivel emocional, mental y cognitivo. Debido a que obligan al paciente a interactuar con el propio juego este se implica de forma más activa en su rehabilitación. El hecho de gamificar la experiencia sirve además para combatir la ansiedad que acompaña a este tipo de tratamientos. Si además el juego es entretenido los pacientes cumplirán más animosos con las exigencias del proceso de rehabilitación, lo que les ayudara a alcanzar los objetivos de cada sesión de entrenamiento y obtener mejores resultados a largo plazo.

El uso de EMG en juegos serios para la rehabilitación

El biofeedback basado en EMG es una técnica que consiste en mostrar al paciente su actividad muscular durante el movimiento, permitiéndoles ver cómo sus músculos se activan y relajan respondiendo a sus acciones a tiempo real. Esto tiene múltiples beneficios tanto para pacientes como para terapeutas:

- **Reentrenamiento de patrones de movimiento:** Al recibir feedback de las activaciones musculares los pacientes pueden ver por sí mismos el comportamiento de sus músculos y que acciones les acercan a los esperados en un sujeto sano, adquiriéndolos poco a poco como propios (*Manual práctico de modificación de conducta*, 1998).
- **Reducción del dolor:** Las activaciones musculares incorrectas continuadas en el tiempo pueden llegar a dar lugar a dolores crónicos. Las terapias que hacen uso de biofeedback puede ayudar a los pacientes a aprender a relajar los músculos cuando no sean necesarios para aliviar las tensiones añadidas que a menudo producen estos dolores (Angoules et al., 2008).
- **Monitoreo objetivo del progreso:** A diferencia de en otros tipos de rehabilitación donde el progreso del paciente se controla casi exclusivamente por la observación del terapeuta mediante el uso de electromiografía se obtienen datos objetivos sobre el rendimiento de este. Analizando estos datos se obtienen la medida real en que los ejercicios de la terapia están afectando al paciente y en base a este conocimiento se pueden adaptar para hacer la terapia más eficiente (Campanini et al., 2020).

Desgraciadamente todas las ventajas del uso de electromiografía en este tipo de terapias vienen también acompañadas por limitaciones, siendo las principales las siguientes (Campanini et al., 2020):

- **Precio y accesibilidad:** Los sensores de electromiografía son dispositivos caros y que, por tanto, no suelen estar disponibles para la rehabilitación, lo que está ralentizando el proceso de implementarlos de forma más extendida en este tipo de terapias.
- **Requiere formación especializada:** Este tipo de juegos pueden ser útiles por sí mismos pero la manera más productiva de utilizarlos es con el apoyo de un profesional formado que sea capaz de interpretar las señales electromiográficas obtenidas. El formar y emplear a estos profesionales para que puedan dar apoyo en este tipo de terapias es también muy costoso y a menudo es la causa por la que se tiende a aplicar terapias menos complejas.

Estas dificultades no opacan sin embargo todas las bondades del uso de señales EMG en este tipo de terapias. Superando estos contratiempos se obtiene una herramienta poderosa para la evaluación de la actividad muscular que permite un seguimiento de la terapia con datos objetivos y una mayor eficacia de los ejercicios propuestos al poder ver el paciente los resultados de estos a tiempo real. Si a todo esto se le añade además el factor de gamificación se puede avanzar hacia nuevas terapias atractivas para el paciente y mucho más efectivas que las terapias convencionales.

Un ejemplo de este tipo de juegos es el trabajo de fin de grado de Javier Cantalapiedra Artiaga en el que se planteó un juego serio controlado con EMG enfocado a la rehabilitación. Este se centraba en promover la activación y relajación voluntarias del bíceps mediante un juego en el cual se controlaba el movimiento de un pájaro contrayendo y relajando dicho músculo (Cantalapiedra Artiaga, 2022).

Biofeedback EMG para trastornos de la marcha en niños con parálisis cerebral

El uso de la técnica de biorretroalimentación con EMG está siendo de gran utilidad para mejorar el pronóstico y tratamiento de niños con parálisis cerebral, objeto de este estudio.

Según el estudio de Dursun (Dursun et al., 2004) a parálisis cerebral se origina por una lesión del sistema nervioso en la etapa de desarrollo con afectación predominante de la función motora. Los niños presentaran espasticidad con activación al unísono de músculos agonistas y antagonistas. Esta activación inadecuada producirá deformaciones en tejidos blandos y daños óseos y articulares, como la deformidad en equino de la articulación del tobillo, lo que conducirá a alteración de la marcha.

El biofeedback se utiliza como medida terapéutica ya que proporciona retroalimentación sobre la actividad neuromuscular voluntaria o autónoma en forma de señales auditivas o visuales.

En el proceso de rehabilitación el biofeedback EMG se utiliza para aprender a contraer y relajar músculos espásticos. Hay pocos estudios al respecto y si bien se ha comprobado que el método puede reducir la espasticidad, no hay evidencia palpable de que mejore la contracción muscular.

Del estudio se infiere que los niños que recibieron biofeedback EMG en musculo dorso flexor y plantar flexor desarrollaban un mejor patrón que los niños que solo recibían terapia convencional de ejercicio.

Se estima que la terapia de biorretroalimentación, que requiere practica y repetición influye en la plasticidad neuronal contribuyendo a la rehabilitación. (Dursun et al., 2004).

1.2. Hipótesis y objetivos

A raíz de todo lo anterior se planteó la posibilidad de diseñar y desarrollar un juego serio que, haciendo uso de electromiografía de superficie para monitorizar la activación de los músculos de los miembros inferiores durante la marcha, ayude al paciente a reeducar la activación y relajación de estos músculos para conseguir que adquieran el comportamiento esperado durante la marcha.

El objetivo de este proyecto era diseñar un prototipo de dicho juego haciendo uso del entorno Unity3D y la API de Delsys asociada. Este juego debía ser capaz de segmentar a tiempo real las fases de la marcha y registrar los patrones de activación neuromuscular empleando sensores en el tibial anterior y el gemelo.

Para alcanzar este objetivo último se planteó la siguiente división en diferentes subobjetivos:

- Idear un sistema para monitorizar a tiempo real la marcha del paciente.
- Obtener datos EMG a tiempo real de los músculos más importantes para la marcha.
- Estudiar las activaciones musculares esperadas durante la marcha de los músculos escogidos.
- Comparar las activaciones obtenidas con las esperadas para detectar patrones anormales en la marcha.
- Representar de forma amigable todos estos datos y recompensar o penalizar al jugador según su desempeño.

El grupo demográfico objetivo se planteó en un principio que fuesen niños con parálisis cerebral, pero posteriormente se decidió desarrollar un juego menos específico que pudieran utilizar todo tipo de personas, dejando la puerta abierta a futuras modificaciones para adaptarlo a cualquier patología concreta que afectase a la activación de estos músculos.

También se incluyó un subobjetivo adicional y previo al trabajo propio de este proyecto que consistía en la mejora del juego serio desarrollado por Javier Cantalapiedra con el objetivo de depurar el tratamiento de la señal EMG realizado en dicho juego y poder aplicar este procesamiento depurado al nuevo juego objeto de este proyecto.

1.3. Materiales

1.3.1. Hardware

En este proyecto se ha hecho uso de los sensores Trigno Avanti de Delsys para obtener todos los datos necesarios del sujeto de estudio. Estos sensores son capaces de captar datos electromiográficos y medidas inerciales, lo que los hace ideales para este proyecto ya que, con los mismos sensores, se pueden captar todos los datos necesarios.

El paquete Trigno Avanti Lite incluye cuatro sensores, una base que actúa como cargador para los sensores y como punto de conexión intermedio entre los sensores y un USB receptor que se conecta al dispositivo con el que se esté trabajando (Figura 1. Componentes del sistema Trigno Avanti Lite de Delsys (Trigno® Lite System - Delsys Europe, s. f.)).



Figura 1. Componentes del sistema Trigno Avanti Lite de Delsys (Trigno® Lite System - Delsys Europe, s. f.)

Otro motivo que los hace adecuados es que, además del propio software de Delsys, la misma empresa proporciona APIs para facilitar el trabajo con sus sensores en diferentes plataformas y lenguajes de programación.

Por último, los sensores se colocan en el paciente adhiriéndolos a la piel por lo que, aunque pueden ocasionar alguna incomodidad, no son demasiado invasivos y su colocación es muy sencilla. Esto unido a la larga duración de su batería los hace perfectos para utilizar en rehabilitación.

Sus especificaciones son las siguientes (*Trigno® Avanti Sensor - Delsys, s. f.*):

Sensor Channels: 1x EMG, up to 6x IMU

Size (Body): 27 x 37 x 13 mm

Mass: 14 g

Contact Material: 99.9% Silver

Battery Life: 4-8 hours

Operating Range: 40 m in RF mode

EMG Sampling Rate (Max): 4370 sa/sec

Accelerometer Sampling Rate (Max): 963 sa/sec
Gyroscope Sampling Rate (Max): 741 sa/sec
Orientation Sampling Rate (Max): 222 sa/sec
On-Board EMG Filter
Real-Time IMU Orientation Filter
Quaternions or RPY (selectable)

1.3.2. Software

Dado que el proyecto se planteó en sus inicios como un juego creado con el software de desarrollo de videojuegos Unity este ha sido el más utilizado, pero a lo largo de todo el proceso se han utilizado softwares adicionales que se expondrán a continuación.

- UNITY

Unity es un motor de desarrollo de videojuegos multiplataforma muy popular hoy en día por su versatilidad y su amplia plataforma de recursos. Cuenta con un entorno gráfico con amplias posibilidades de diseño. Además, Unity se vale de la programación en el lenguaje C# para añadir funcionalidades más complejas y personalizadas.

La versión de Unity utilizada en este proyecto es la 2021.3.0f1 ya que es para la que se creó la API de Delsys más actualizada para trabajo con Unity.

- VISUAL STUDIO

Como se decía anteriormente parte del proceso de crear un juego en Unity es programando el lenguaje C#. Lo más habitual es vincular Unity con un editor externo, seleccionando este como predeterminado en las preferencias de Unity.

Visual Studio es un entorno de desarrollo de Microsoft que permite programar en diferentes lenguajes, ofreciendo multitud de herramientas para ayudar al programador como un explorador que permite administrar los archivos del proyecto, detección de errores de código e incluso sugerencias para ponerles solución. Es una solución muy completa que simplifica mucho el trabajo con scripts en Unity.

Además, Visual Studio ofrece la posibilidad de añadir paquetes externos como el de la API de Delsys, necesario para el trabajo en C# con los sensores de la compañía. Para ello se utilizó el administrador de paquetes NuGet Package Manager.

- SOFTWARE DE DELSYS

Delsys tiene su propio software para la recolección y el procesamiento de datos de sus sensores. Estos programas se utilizaron durante el desarrollo de este proyecto para visualizar los efectos de los procedimientos de tratamiento de las señales implementados en el juego de forma sencilla

El primero de ellos es Trigno Discover, un programa que recoge los datos de los sensores según el modo de muestreo escogido y que los muestra por pantalla mediante su interfaz interactiva (Figura 2). Esta permite dimensionar las señales a voluntad y compararlas visualmente entre sí con ayuda de un cursor deslizante.



Figura 2. Interfaz de Trigno Discover

Por otro lado, está EMG Works Analysis, un software que permite realizar operaciones sobre los datos recogidos mediante Trigno Discover. Para ello basta con crear un nuevo espacio de trabajo e importar los archivos de datos. Sin más que hacer doble clic sobre la señal deseada esta se mostrará en el visualizador y haciendo clic derecho sobre la gráfica se puede acceder a la lista de operaciones posibles sobre la señal, como se ve en la Figura 3.

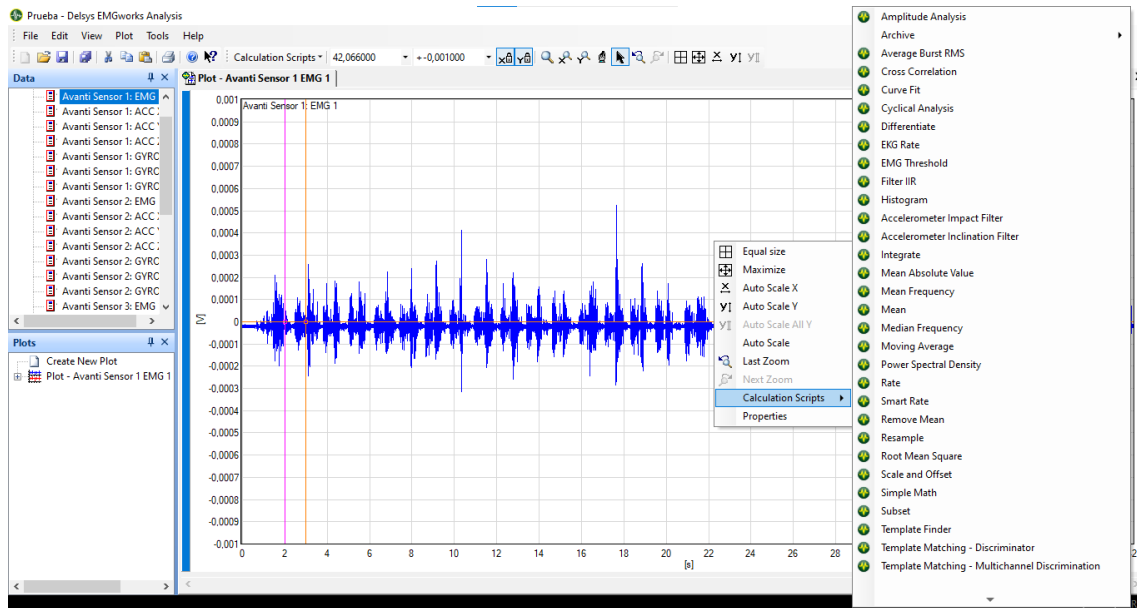


Figura 3. Interfaz de EMG Works Analysis

- **MATLAB**

El software de Delsys ofrecía operaciones limitadas sobre la señal, por lo que se hizo uso también de la plataforma MATLAB para hacer algunos tratamientos más complejos sobre la señal. Para ello se utilizó la opción de Trigno Discover de exportar los datos en formato csv, de forma que pudiese importarse en MATLAB.

1.4. Fases del proyecto

El proceso de desarrollo del proyecto se ha dividido en las siguientes fases:

1. **Cursos de introducción a Unity.** Unity proporciona de forma gratuita varios cursos de formación para aprender a trabajar con su software y con el lenguaje de programación C#. Estos cursos ofrecen todo el contenido necesario para aprender las bases del trabajo con Unity y proponen ejercicios para practicar los conocimientos adquiridos (*Unity Learn*, s. f.).
2. **Investigación teórica.** A continuación, se llevó a cabo una investigación sobre electromiografía y el ciclo de la marcha partiendo desde un conocimiento prácticamente nulo en ambos campos, por lo que hubo que comenzar por las bases antes de entrar en conceptos algo más complejos.
3. **Estudio de la API de Delsys.** Ya que el proyecto hace uso de la instrumentación de Delsys el siguiente paso fue familiarizarse con ella, comenzando por conocer todos los componentes y sus funciones. Se aprendió a colocar correctamente los sensores y a analizar los datos captados. Además, se estudiaron las posibilidades de la API de Delsys para Unity y la forma de utilizarla.
4. **Mejoras en juego anterior.** Después de probar el juego se estudiaron las posibilidades para mejorar su calibración y se retocó tanto su interfaz como su código para implementar el nuevo método elegido.
5. **Lluvia de ideas.** Antes de comenzar a desarrollar el nuevo juego se realizó una lluvia de ideas tratando de llegar al formato correcto que cumpliera todos los objetivos buscados por el juego.
6. **Desarrollo del juego.** Esta fase comprendió el diseño en Unity, la programación asociada y el trabajo con las señales recogidas fuera de Unity.
7. **Pruebas.** Durante el desarrollo del juego y al finalizar este se llevaron a cabo ciertas pruebas para evaluar su funcionamiento y, en caso necesario, hacer las modificaciones pertinentes.

2. Fundamentación teórica

2.1. EMG

De acuerdo con el Brigham and Women's Hospital “la electromiografía (EMG) es una prueba que permite medir la respuesta muscular o la actividad eléctrica en respuesta a la estimulación nerviosa de un músculo.” (*Electromiografía*, s. f.).

La medición de la actividad eléctrica muscular es útil en diferentes campos como la neurología, la fisioterapia y la rehabilitación. Los datos electromiográficos sirven tanto para estudiar la actividad muscular en sujetos sanos como para diagnosticar y controlar el tratamiento de sujetos patológicos (Konrad, 2005).

2.1.1. Principios físicos del movimiento

Para trabajar con electromiografía lo primero es conocer los principios físicos en los que se basa. Para comprender la activación muscular es necesario conocer los conceptos generales de la fisiología de los músculos y de la transmisión neuromuscular (Hernández & Ruiz, 2006).

- Fisiología muscular

Los músculos esqueléticos están formados por fascículos de fibras musculares. Cada fibra muscular no es más que una célula alargada y con múltiples núcleos que dispone en su interior de unas estructuras, las miofibrillas. Estas están a su vez compuestas por conjuntos de filamentos de actina y miosina llamados sarcómeros. La interacción de la actina y la miosina dentro del sarcómero será la responsable de la contracción muscular.

Los filamentos de actina y miosina se deslizan uno sobre otro, encogiendo y estirando el sarcómero. Al contraerse los sarcómeros se contraen a su vez las fibras musculares y por ende el músculo. La señal del sistema nervioso central para la activación muscular, en forma de impulso eléctrico, viajara por la medula espinal a través de las neuronas motoras hasta la placa motora del músculo para dar comienzo a este proceso.

- Transmisión neuromuscular

El impulso eléctrico generado en la corteza motora del encéfalo viaja a través de los cordones motores de la medula espinal hasta el músculo en cuestión. Pasa primero a través del axón de la primera neurona motora, cuyo cuerpo se encuentra en el sistema nervioso central. El axón de la segunda neurona motora hace sinapsis en la llamada placa motora, ubicada en el músculo. Es este estímulo eléctrico el que inicia la contracción muscular

El impulso nervioso que llega a la placa motora hace que se libere el neurotransmisor acetilcolina. Este neurotransmisor se une a los receptores que existen para él en la membrana muscular, llamada sarcolema, lo que provoca la despolarización de la membrana y la contracción de los filamentos de actina y miosina a través de una compleja sucesión de reacciones químicas. Este proceso es el conocido como acoplamiento excitación-contracción

- **Actividad eléctrica producida en los músculos**

La despolarización de la membrana muscular, con los cambios iónicos que origina, se traduce en una corriente eléctrica que será fácilmente medible y detectable a través de electrodos. La actividad eléctrica detectada será la suma de los potenciales de acción de cada fibra muscular implicada en la contracción.

2.1.2 Dispositivos para medir EMG

Para obtener los datos de activación se captura esa corriente eléctrica mediante electrodos. De los diferentes tipos de electrodos existentes los más utilizados son los de superficie y los de aguja. Cada uno de ellos tiene sus propias ventajas e inconvenientes y se utilizan, por tanto, en diferentes casos (Kimura, 2013).

- **Electromiografía de Superficie**

La electromiografía de superficie consiste en colocar electrodos en contacto con la piel para capturar las activaciones musculares de la zona.

Esta técnica es eficaz para medir la actividad de músculos grandes, ya que tienen una capacidad limitada en cuanto a la resolución espacial. Ante músculos demasiado pequeños el electrodo captará también actividad de los músculos que lo rodean, produciendo interferencias en la señal deseada. Tampoco tienen demasiada precisión frente a sus competidores y las señales recogidas pueden verse afectadas por el grosor de la grasa subcutánea, pero siguen siendo una gran opción para analizar la actividad de grandes músculos superficiales ya que es una técnica muy poco invasiva. Debido a sus ventajas y limitaciones se suele utilizar estos electrodos para estudiar la postura y los movimientos voluntarios.

En el trabajo con este tipo de sensores es importante colocarlos de forma correcta y fijarlos bien para que no se muevan durante la recolección de datos. Además, hay que limpiar bien tanto el electrodo como la zona de piel donde se va a colocar para conseguir la mejor conductividad eléctrica posible. También existe la posibilidad de utilizar cremas que mejoran la conducción, pero en la mayor parte de los casos esto no es necesario.

- **Electromiografía de Aguja**

En la electromiografía de aguja el electrodo se introduce en el músculo. Es un método mucho más invasivo que el anterior, pero, a cambio, tiene mucha mejor resolución

espacial dado que se centra en la fibra muscular en la que se introduce. Esto permite el estudio de músculos concretos, aunque estos tengan un tamaño pequeño.

Es también mucho más preciso que los sensores de superficie por lo que se pueden utilizar para diagnosticar afecciones de mucha menor magnitud. Esta diferencia de precisión y el hecho de que con ellos se puede acceder a músculos no superficiales hace a los electrodos de aguja la metodología perfecta para estudiar trastornos neuromusculares y servir como apoyo en intervenciones terapéuticas.

2.1.3 Aplicaciones Clínicas de la Electromiografía

Las aplicaciones de la electromiografía son amplias desde el punto de vista clínico. Para empezar, con ella es posible diagnosticar multitud de patologías neuromusculares. Se puede usar sin embargo en otros campos como por ejemplo el deportivo o la rehabilitación. Algunas de sus aplicaciones son:

- Diagnóstico de Trastornos Neuromusculares

La electromiografía es una herramienta muy útil a la hora de diagnosticar múltiples patologías nerviosas y musculares como las siguientes (Kimura, 2013), ya que la varían notablemente las señales obtenidas respecto a las de sujetos sanos.

a. Neuropatías

Las neuropatías afectan a los nervios periféricos y eso se traduce en una mala inervación muscular que puede provocar dolor, pérdida de sensibilidad o pérdida de fuerza. Por ejemplo, en la neuropatía axonal como se produce pérdida de fibras nerviosas, lo que puede observarse al tomar datos electromiográficos y analizarlos ya que potenciales de acción obtenidos tendrá una amplitud más pequeña de lo normal.

b. Miopatías

Las miopatías son enfermedades que afectan a los músculos, provocando debilidad y derivando en atrofia. Mediante electromiografía se pueden identificar las miopatías estudiando los potenciales de acción, que serán cortos y de poca amplitud, lo que refleja el bajo número de unidades motoras disponibles.

c. Enfermedades de la Motoneurona

Las motoneuronas son las encargadas de controlar la contracción de los músculos esqueléticos voluntarios. Los cuerpos de estas neuronas se encuentran en la corteza cerebral. Si el impulso no llega o llega con dificultad a la placa muscular el músculo se irá atrofiando y debilitando progresivamente.

La EMG puede detectar esta denervación y también los fenómenos de reinervación compensadora ineficaz que se traducirá en fibrilaciones y fasciculaciones en un intento

del organismo de volver a la situación inicial. Un ejemplo de esta patología en la esclerosis lateral amiotrofia (ELA).

d. Trastornos de la Unión Neuromuscular

El hecho de que no lleguen suficientes neurotransmisores a la placa motora para comenzar una despolarización eficaz o haya pocos receptores para estos neurotransmisores en el musculo se traducirá en debilidad y fatiga muscular. En las lecturas EMG esto quedará reflejado en una reducción en la amplitud de la señal a medida que se repite la contracción muscular por agotamiento de los receptores.

- Evaluación del Rendimiento Muscular

La electromiografía no sirve únicamente para detectar patologías sino también para ayudar en sus tratamientos y para prevenir lesiones (Escamilla et al., 1998).

Por un lado, es empleada en el ámbito deportivo. El análisis electromiográfico de los deportistas permite controlar su actividad muscular durante los ejercicios. El análisis de estos datos sirve para ajustar los ejercicios de forma que estos sean más eficientes y disminuir la posibilidad de lesiones.

En el caso de la rehabilitación también es muy útil analizar la actividad muscular. Comparando los datos electromiográficos durante la fisioterapia se pueden ir controlando los cambios en la activación. De esta forma se puede comprobar si la metodología de rehabilitación está resultando efectiva y modificarla si no es el caso.

- Biofeedback

Como se comentaba anteriormente, la EMG se está utilizando cada vez más en técnicas de biofeedback. Con esto se consigue no solo la recogida de datos sobre activaciones musculares para su posterior estudio, si no que los propios sujetos puedan interactuar con estos datos. El hecho de exponerles a sus propios datos de activación muscular los lleva a comprender mejor como estos afectan a su postura y movimiento, y a mejorar su control sobre ellos, siendo muy útiles estas técnicas en terapia de rehabilitación.

2.1.4 Factores externos que afectan a la señal EMG

La EMG es una técnica muy útil y eficaz, pero la interpretación de las señales obtenidas se dificulta por todas las variables que influyen en su análisis.

La señal eléctrica evaluada dependerá principalmente del número de fibras musculares y de la sincronización de sus potenciales de acción. Estos factores pueden verse afectados por diferentes circunstancias, entre ellas las descritas a continuación:

a. Edad y Género

La edad y el género son dos factores importantes en la electromiografía. Con el paso de los años los pacientes van perdiendo masa muscular y, por tanto, unidades motoras por lo que los pacientes más mayores originan señales de menor amplitud. Por otro lado, la distribución de grasa muscular, que varía notablemente en función del género, puede influir también en la señal, sobre todo con el uso de electrodos de superficie.

b. Estado del Músculo

A la hora de interpretar datos EMG es importante saber las condiciones del músculo en el momento de la medición. Si por ejemplo este estaba fatigado o presentaba espasmos o contracturas los datos no serán fiables. Incluso su temperatura puede variar notablemente las activaciones.

Cada uno de estos factores afectan de manera diferente a las mediciones. En el caso de la fatiga, por ejemplo, la frecuencia de las activaciones tiende a disminuir, pero puede aumentar su amplitud, ya que se sincronizan múltiples unidades motoras para tratar de mantener la fuerza de la activación.

c. Técnica de Registro

Más allá de la técnica escogida para recoger los datos de electromiografía es importante también llevarla a cabo correctamente. En el caso de los electrodos de superficie, por ejemplo, se debe tener especial cuidado en la correcta colocación de estos, asegurándose que se ponen en el lugar correcto y que quedan bien fijados. Además, estos no deben estar sucios ni desgastados y no se deben evitar entornos con interferencias electrónicas que puedan distorsionar las lecturas EMG (Basmajian & DeLuca, 1986).

d. Otras condiciones médicas

Hay que tener en cuenta las complicaciones médicas que, aunque sean aparentemente completamente independientes, pueden interferir en la lectura de los resultados de la EMG. Enfermedades metabólicas como la diabetes o el hipotiroidismo modifican los patrones de contracción muscular en algunos casos. Así mismo, medicamentos como los relajantes musculares o fármacos empleados en diversas patologías cardíacas pueden afectar la actividad eléctrica muscular, dando valores electromiográficos irreales que pueden llevar a diagnósticos o tratamientos erróneos. Por tanto, es importante conocer el historial clínico del sujeto antes de interpretar los datos obtenidos (Kimura, 2013).

2.2. Ciclo de la marcha

La marcha es la sucesión de un ciclo de movimientos que permite el desplazamiento erguido. Cada ciclo consta de una fase de apoyo (pie en contacto con el suelo) y una de balanceo (pie en el aire). Ambas piernas realizan este ciclo durante la marcha, de manera sincronizada pero desfasadas una respecto a la otra, permitiendo así el avance hacia delante.

El ciclo empieza con el pie en el suelo y acaba cuando el mismo pie toca el suelo después del balanceo. En realidad, al ser cíclico se podría considerar cualquier otro instante como comienzo y final del paso, pero generalmente se considera este (Pifarre, 2021).

2.2.1 Fases de la Marcha

Para comprender la marcha de manera más profunda es clave conocer no solo las anteriormente mencionadas si no sus subfases (Perry & Burnfield, 2010).

1. Fase de Apoyo (Stand)

La fase de apoyo es el periodo de la marcha en que el pie se encuentra en contacto con el suelo. Esta se divide en diferentes subfases:

- **Contacto Inicial (Heel Strike o HS):** Considerado el inicio de la marcha, es el instante en que el pie hace entra en contacto con el suelo. Lo primero que toca el suelo es el talón, que es la parte del pie que recibe y absorbe el impacto inicial.
- **Respuesta a la Carga (Loading Response):** Tras el contacto inicial el peso del cuerpo pasa de recaer únicamente en el talón a distribuirse más según el resto de la planta del pie va entrando también en contacto con el suelo. Durante esta fase se termina de absorber el impacto con el suelo y se va estabilizando el cuerpo.
- **Apoyo Medio (Midstance, Foot Contact o FC):** El apoyo medio hace referencia al periodo durante el cual la planta del pie está completamente en contacto con el suelo. Es este pie en el que recae el peso completo del cuerpo, en preparación para que el otro pie pueda levantarse del suelo durante esta misma fase.
- **Apoyo Terminal (Terminal Stance):** En esta parte del paso el peso del cuerpo se desplaza hacia delante, hacia la puntera del pie. También se empieza a elevar el talón en preparación para la fase de balanceo.
- **Prebalanceo (Preswing, Toe Off o TO):** Esta es la última fase del stand, en la que el pie se eleva por completo del suelo para dar inicio al balanceo.

2. Fase de Balanceo (Swing)

La fase de balanceo abarca el periodo del paso en que el pie no está en contacto con el suelo. También se divide en subfases:

- **Balaneo Inicial (Initial Swing):** Esta fase comienza justo después de que el pie se despegue por completo del suelo. A partir de entonces comienza su balaneo hacia delante para lo que es importante la flexión de la rodilla y le tobillo, de forma que se eviten tropiezos.
- **Balaneo Medio (Midswing o MS):** El balaneo medio comprende el instante en que la pierna en balaneo adelanta a la de apoyo, es decir, cuando la pierna que no está en contacto con el suelo pasa justo por debajo del tronco del sujeto. A partir de ese momento la pierna vuelve a estirarse para prepararse para el siguiente contacto con el suelo.
- **Balaneo Terminal (Terminal Swing):** En esta fase la pierna ya está completamente estirada y el pie se coloca en preparación para el contacto con el suelo, de forma que este se produzca con el talón.

Para este proyecto es importante recalcar que el ciclo de la marcha es, como su propio nombre indica, cíclico. Como se puede ver de forma muy gráfica en la Figura 4. La sucesión de las diferentes fases de la marcha puede representarse fácilmente como un círculo. Por lo tanto, a la hora del desarrollo del juego se optó por ofrecer una visión circular de la marcha parecida a la de la figura, de forma que el flujo este fuese continuo y no hubiese saltos visuales entre paso y paso.

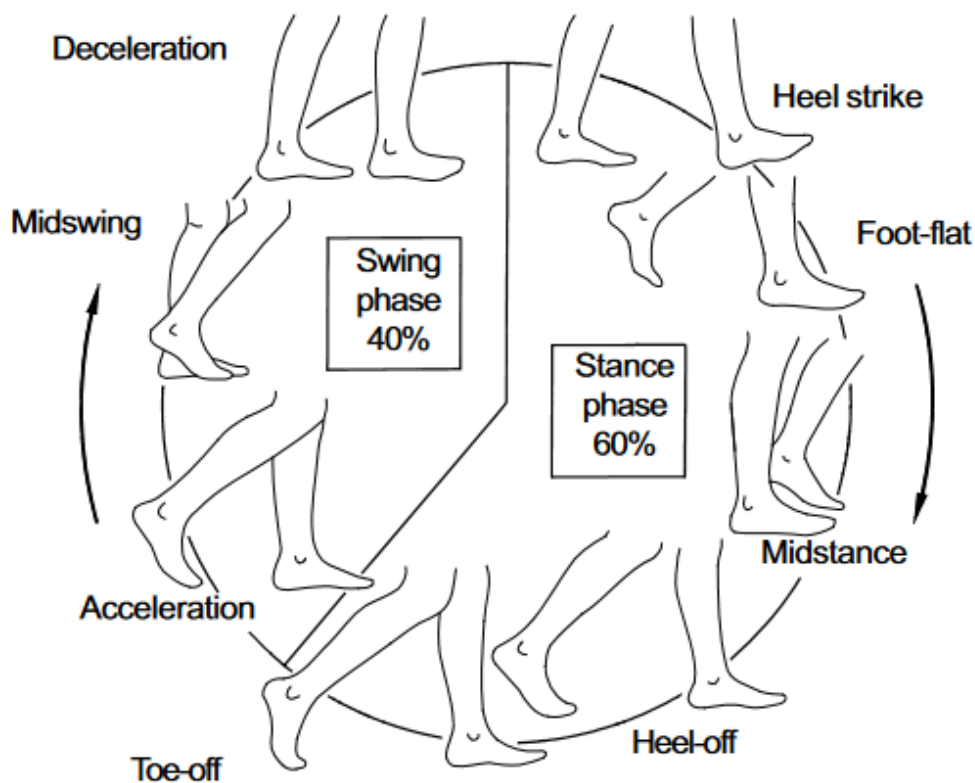


Figura 4. Representación circular del ciclo de la marcha (Davis & Vaughan, 1999)

2.2.2 Detección de la marcha

Para identificar estas fases en la marcha de un sujeto real una de las formas más sencillas es colocar un sensor en parte anterior de la pierna, sobre la tibia, que capte la velocidad angular del eje orientado en la dirección de la marcha como se hace en el trabajo de fin de grado de Carmen Fernández González (Fernández González, 2024). Esa señal, en un sujeto sano, tendrá un aspecto similar a la de la Figura 5.



Figura 5. Velocidad angular en x medida desde un sensor colocado en el tibial

Para caracterizar la marcha a partir de esta señal se utilizan cuatro los puntos de referencia del paso que se ven en la Figura 6, siendo:

1. Balanceo medio o Mid Swing (máximo por encima de cierto umbral)
2. Contacto con el suelo o Heel Strike (mínimo local tras un MS)
3. Apoyo medio o Foot Contact (máximo local tras un HS)
4. Prebalanceo o Toe Off (mínimo local tras un FC)

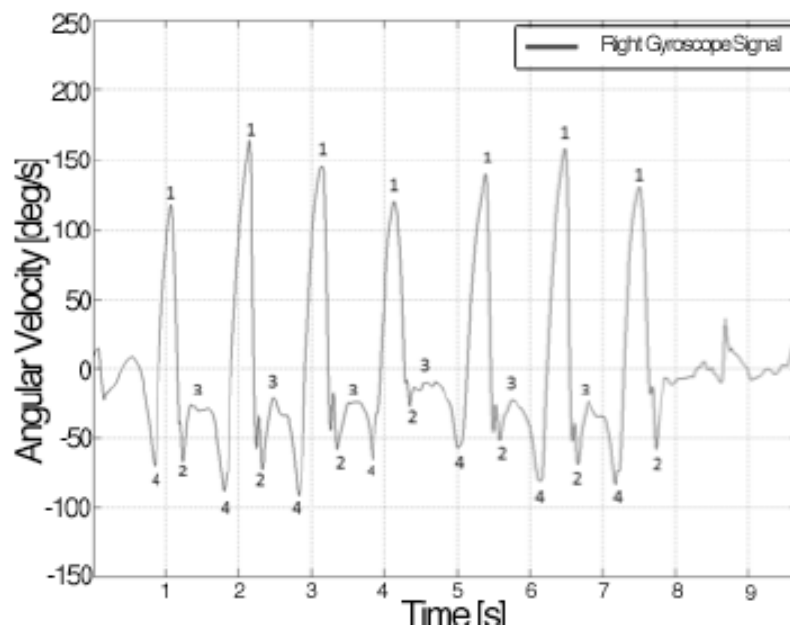


Figura 6. Puntos clave de la marcha sobre la señal de velocidad angular en x (Fraccaro et al., 2014)

2.3. Actividad EMG en la marcha

Muchos músculos son partícipes de la marcha, entre ellos destacan el gemelo y el tibial anterior. Estos dos son músculos opuestos en cuanto a localización espacial y complementarios del uno del otro. Cada uno se encarga de funciones diferentes pero imprescindibles durante la marcha.

El tibial anterior se localiza en la parte delantera de la pierna, siguiendo al hueso con el mismo nombre. Se encarga de la dorsiflexión del tobillo, una parte crucial de la marcha. Al comienzo del balanceo se tiene que producir una dorsiflexión, es decir, la puntera del pie tiene que levantarse para evitar que se arrastre por el suelo y provoque tropiezos (Whittle, 2008).

El gemelo se encuentra en la parte posterior de la pierna, en la parte de arriba de la pantorrilla. Su función principal es de empuje contra el suelo, lo que vale para propulsar el cuerpo y levantar el pie del suelo (Neptune et al., 2001).

Según su función cada uno de ellos se activa o no en cada una de las distintas fases de la marcha.

1. Fase de Apoyo

- Contacto Inicial (Heel Strike):

Tibial Anterior: Dado que el contacto inicial con el suelo tiene que producirse con el talón el tobillo tiene que estar en dorsiflexión en el instante de contacto con el suelo. Por tanto, el tibial anterior, que es el encargado de levantar la puntera del pie, se activa de forma muy intensa antes y durante el impacto del pie con el suelo.

Gemelo: El gemelo no se activa durante el contacto inicial. A parte del tibial se activan otros músculos para absorber el impacto y estabilizar el cuerpo, pero el gemelo no es uno de ellos.

- Apoyo Medio (Mid Stance):

Tibial Anterior: Durante esta fase la actividad del tibial disminuye poco a poco hasta desaparecer para permitir que el tobillo se estire.

Gemelo: Mientras tanto el gemelo se activa progresivamente para estabilizar el cuerpo controlar su avance hacia delante, cambiando el peso del cuerpo hacia la puntera del pie de forma progresiva.

- Despegue (Toe Off):

Tibial Anterior: El tibial anterior comienza a activarse en esta fase, preparando la dorsiflexión para levantar el pie correctamente y evitar que los dedos sigan tocando el suelo cuando comience el swing.

Gemelo: Es esta etapa el gemelo alcanza su pico de activación ya que tiene que propulsar el pie lo suficiente para terminar de cambiar el peso del cuerpo a la puntera y levantar del suelo primeramente el talón y a continuación el pie completo.

2. Fase de Balanceo

- Aceleración (Early Swing):

Tibial Anterior: La activación del tibial anterior continúa durante esta etapa. Esto es esencial para mantener el tobillo dorsiflexionado y la puntera del pie elevada del suelo para evitar tropiezos.

Gemelo: El gemelo se desactiva durante esta fase ya que la propulsión hacia arriba deja de ser necesaria.

- Balanceo Medio (Mid Swing):

Tibial Anterior: Durante el balanceo medio el tibial permanece activado para mantener el tobillo dorsiflexionado.

Gemelo: El gemelo sigue inactivo en esta fase.

- Desaceleración (Late Swing):

Tibial Anterior: Durante esta última fase del balanceo aumenta la activación del tibial anterior en anticipación al contacto con el suelo, incrementando la dorsiflexión del tobillo para asegurar que la primera parte del pie que entrará en contacto con el suelo será el talón.

Gemelo: El gemelo sigue inactivo en esta fase.

En la siguiente tabla (Davis & Vaughan, 1999) se pueden observar de forma simple y clara las activaciones de ambos músculos durante el paso.

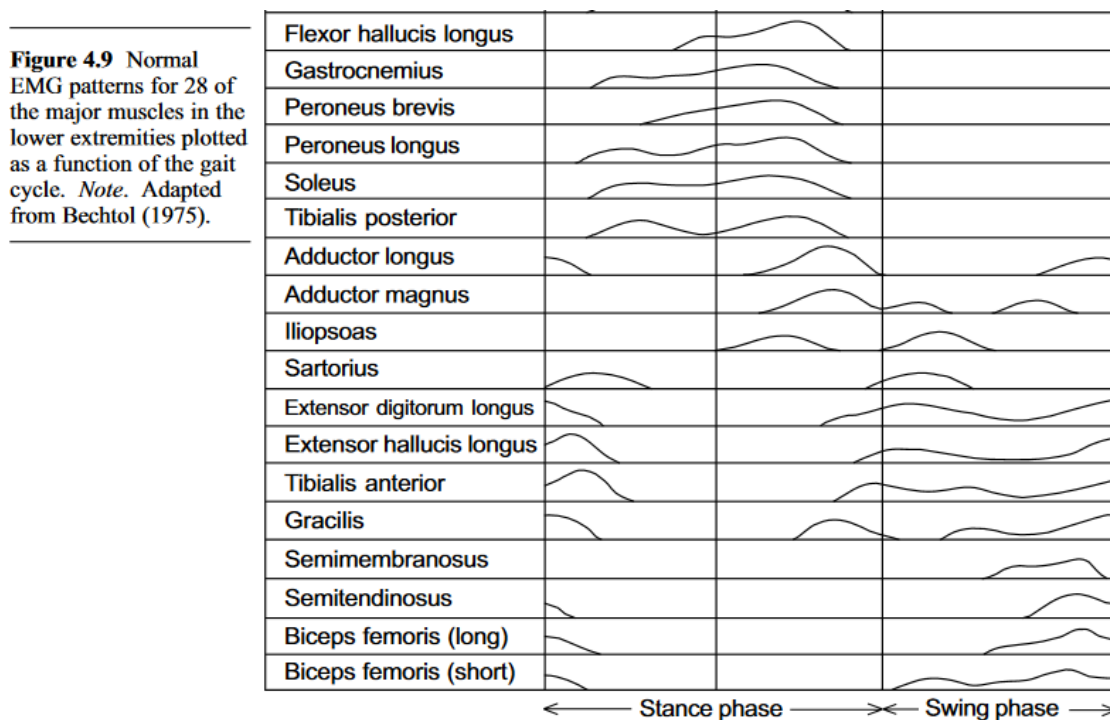


Figura 7. Activaciones de diferentes músculos durante la marcha (Davis & Vaughan, 1999)

En vista de todo lo anterior se puede afirmar que, en principio, se esperarían las siguientes activaciones:

Gemelo: Durante la mayor parte del stand

Tibial: Durante el swing completo y la primera parte del stand.

Este sería el comportamiento esperado según la función que cumple cada músculo durante la marcha, pero lo cierto es que, como se explicaba anteriormente, las activaciones musculares pueden variar bastante de una persona a otra por diferentes motivos, aunque ambas sean sujetos sanos.

En la práctica se ha comprobado que hay diferentes patrones de activación durante la marcha de sujetos sanos, tanto para el gemelo como para el tibial anterior. La diferencia radica en el número de activaciones que se producen a lo largo del paso, que pueden variar de una a cuatro para el gemelo y de dos a cinco para el tibial. Según el número de activaciones las zonas de activación de cada músculo se distribuyen de forma diferente durante el paso, no coincidiendo en muchos casos con las activaciones esperadas según la teoría (Di Nardo et al., 2013).

Observando los resultados de estos experimentos se puede afirmar que, independientemente del número de activaciones, el tibial anterior presenta activación durante la mayor parte del swing y la activación del gemelo se concentra principalmente en el stand. Según el patrón de activación concreto del sujeto se producirán otras activaciones durante otros instantes del paso, pero siempre que se trate de sujetos sanos se cumplirán las anteriores afirmaciones por lo que serán sobre las que se basará este proyecto.

3. Métodos

3.1. Mejora en la calibración del juego anterior

Como primera aproximación al tratamiento de datos EMG en Unity se partió del trabajo de fin de grado de Javier Cantalapiedra con la intención de mejorar la sensibilidad del juego desarrollado en dicho trabajo (Cantalapiedra Artiaga, 2022)

El juego comenzaba con una calibración en la que se fijaba un umbral de activación EMG en función de la activación máxima posible del músculo, pero esto daba lugar a umbrales demasiado altos, lo que dificultaba al jugador controlar el juego. Durante esta calibración se solicitaba al sujeto hacer toda la fuerza posible con el músculo en cuestión y se recogían los datos EMG correspondientes para, posteriormente, tomar el máximo de estos valores como umbral. El problema era que al quedarse con el valor máximo se producían desbalances dado que, a menudo, el valor que se tomaba era un valor de sobre pico.

Para evitar este suceso se implementó el uso del valor cuadrático medio (RMS). El uso de RMS en el tratamiento de señales EMG es un proceso muy extendido ya que suaviza los datos y se deshace de los posibles sobre picos erróneos de una forma muy sencilla (*EMG Signal Analysis*, s. f.; Konrad, 2005).

Para implementarlo se optó por utilizar el propio funcionamiento de la obtención de los datos en el juego. Se comprobó que cada paquete recibido incluía 26 datos EMG y se cambió el funcionamiento de la calibración de forma que en vez de almacenar los 26 datos de cada paquete se almacenase su RMS. De esa lista de valores cuadráticos medios ya podía utilizarse el máximo como umbral sin temor a introducir errores.

A continuación se presenta el desglose del código de la obtención de los valores cuadráticos medios y demostración una gráfica de los efectos del procesado (Fragmento de código 1).

```
foreach (var dato in DataValid) //A cada dato le restamos
la media del paquete (eliminamos el offset) y tomamos el
valor absoluto (rectificamos)
{
1.   var datoMinusMean = dato - DataValid.Mean();
2.   var absdato = Math.Abs(datoMinusMean);
    DataPositive.Add(absdato);
}

//Calculamos la rms del paquete
3. rms = Math.Sqrt(promedioCuadrados. RootMeanSquare());
```

Fragmento de código 1. Cálculo de RMS de datos recogidos en el programa anterior

Señal original:

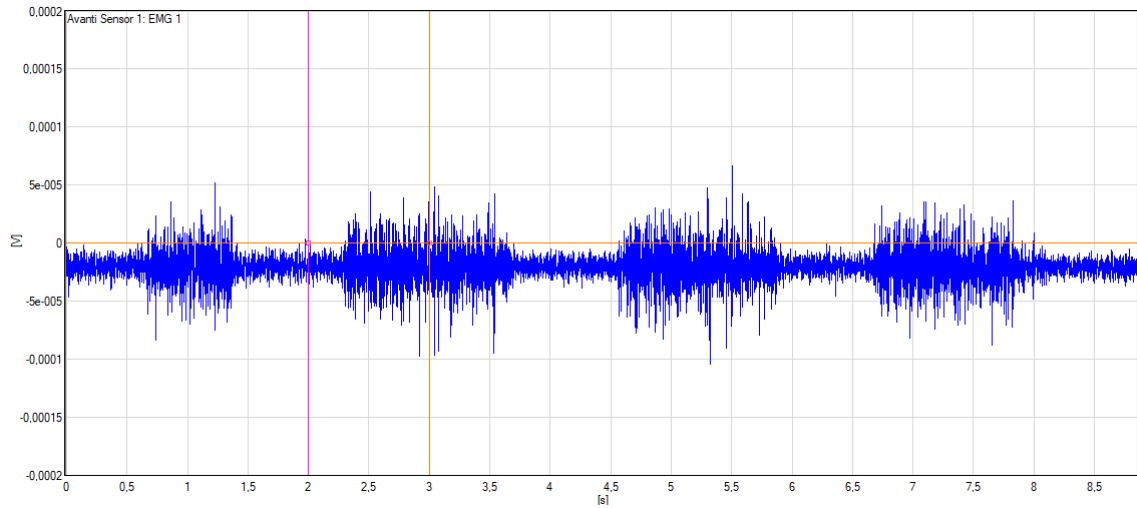


Figura 8. Señal EMG prueba original

1. Restar el offset (Remove mean):

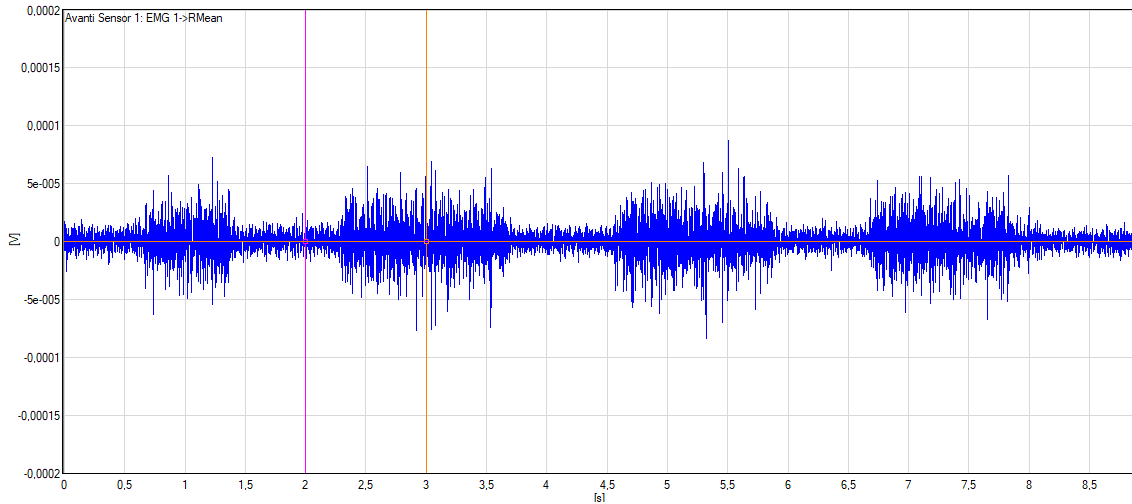


Figura 9. Señal EMG prueba tras restar offset

2. Rectificar (Simple Math -> Abs):

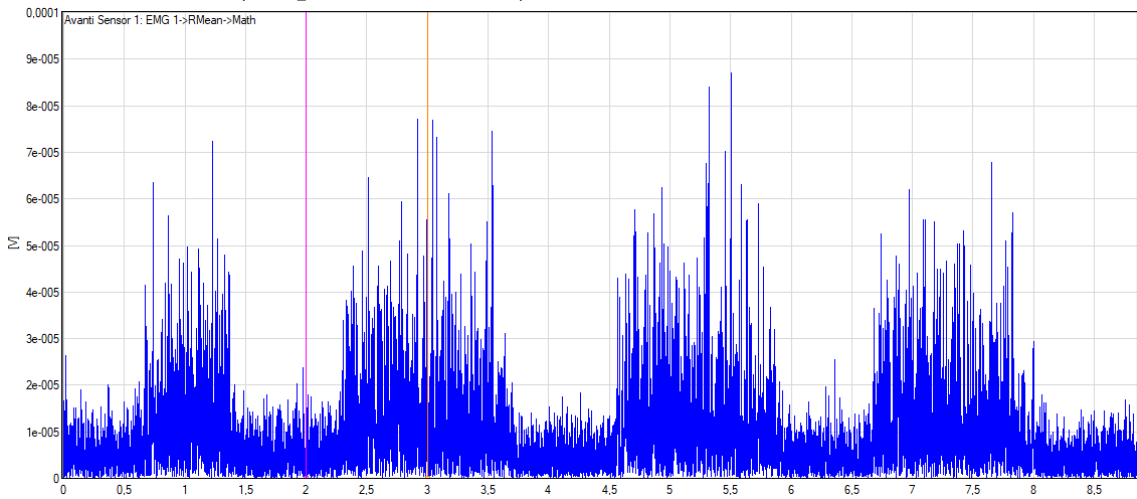


Figura 10. Señal EMG prueba tras rectificación

3. RMS (Root Mean Square):

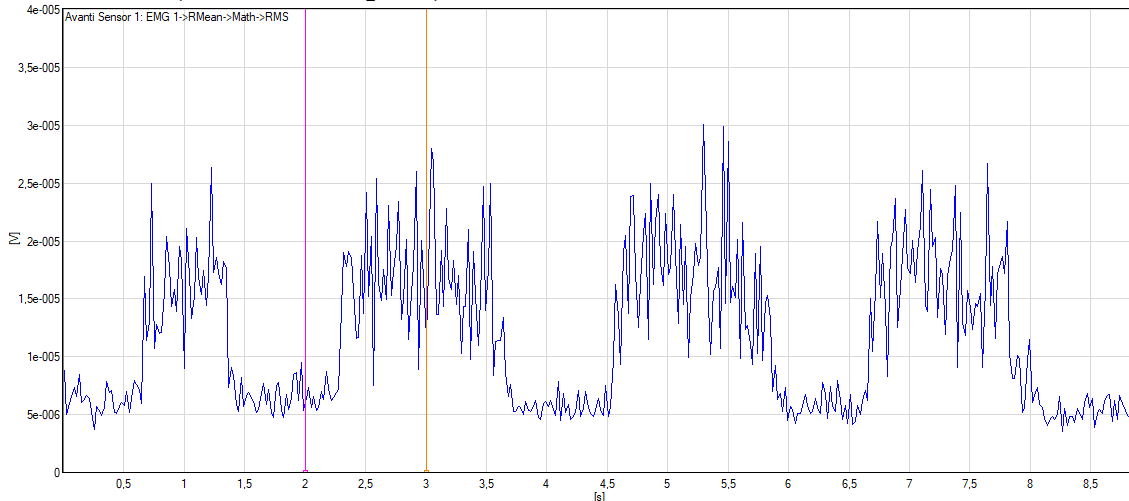


Figura 11. Señal EMG prueba tras RMS

Sin embargo, esto no mejoró notablemente la jugabilidad pues los valores máximos seguían siendo difíciles de repetir durante el juego dado que habían sido obtenidos a partir de la máxima fuerza que el jugador puede realizar, algo que no puede mantenerse en el tiempo durante el juego. Por ello hubo que reconsiderar la estrategia de calibración completa.

Se planteó como nueva idea no tener en cuenta solamente el valor de activación máxima si no el del músculo en reposo también. Con ambos valores se establece un nuevo umbral más balanceado de la siguiente forma:

```
umbral = (EMGintentomaximo - EMGintentominimo) * 0.35 +
EMGintentominimo;
```

Fragmento de código 2. Cálculo de umbral (código simplificado)

Así logramos un umbral con valor mayor que el valor de reposo en un 35% de la diferencia entre este y el valor máximo. Con esto se asegura por un lado que no se tome como activación un momento de reposo y por otro que el umbral no esté demasiado próximo al máximo de activación para que se pueda controlar el juego ejerciendo una fuerza moderada.

3.2. Ideando el nuevo juego

Al enfrentarse al planteamiento del nuevo juego hay que enfrentarse a dos aspectos muy importantes.

3.2.1 Funcionalidad

El juego busca dos funcionalidades principales, caracterizar el ciclo de la marcha a tiempo real y las activaciones musculares durante este de una forma visual y clara.

Para ello se tenía que obtener el punto del estado de la marcha en que se encuentra el jugador en cada momento e idear una forma de representarlo gráficamente.

Por otro lado, había que plantear un método de procesado de los datos de electromiografía con el que determinar con la mayor precisión posible si el músculo en cuestión estaba o no activo en cada momento.

Por último, había que buscar la forma de mostrar tanto estas activaciones como las partes del paso donde se espera cada una de la forma más clara posible.

3.2.2 Disfrute

Un juego serio es, al fin y al cabo, un juego y, aunque persigue una finalidad más provechosa, debe ser entretenido para el jugador.

Al ser el ciclo de la marcha la base del juego este tenía que ser irremediabilmente cíclico, lo que puede resultar monótono de por sí. Pero, además, al ser el hecho de caminar la forma de interactuar con el juego, no es demasiado interactivo dado que modificar la forma de caminar es un proceso complejo que puede resultar frustrante.

Primeramente, se trató de crear una interfaz atractiva, incluyendo un pájaro animado para ayudar a señalar el ciclo de la marcha y ambientando lo demás de acuerdo con esa temática.

Para motivar al jugador se introdujo un sistema de recompensas. A la representación gráfica de la marcha y las activaciones musculares durante esta se le añadió la recolección de puntos cuando estas activaciones fuesen las esperadas. Los puntos recolectados son en realidad hojas y palos para integrarlo con la temática de los pájaros y a cada punto recolectado lo acompaña un sonido que sirve de refuerzo positivo para el jugador.

3.3. Nuevas necesidades en la recogida de datos

Cabe destacar el hecho de que la API de Delsys fue actualizada desde la realización del juego anterior, por lo que la forma de recoger datos de los sensores variaba ligeramente, el nuevo juego trajo nuevas necesidades:

3.3.1 Incorporar dos sensores

El nuevo juego incluía la dificultad añadida de trabajar con más de un sensor, dado que se necesitaba captar datos electromiográficos del tibial y el gemelo de forma simultánea. Por lo tanto, se hizo necesario conectar dos sensores y obtener y procesar los dos flujos de datos.

Del ejemplo que acompañaba a la nueva API de Delsys se utilizó el siguiente código que es el encargado de almacenar los datos que envían los diferentes sensores en listas con las que trabajar en C#.

```
for (int k = 0; k < e.Data.Count(); k++)
{
    // Loops through each connected sensor
    for (int i = 0; i < e.Data[k].SensorData.Count(); i++)
    {
        // Loops through each channel for a sensor
        for (int j = 0; j <
e.Data[k].SensorData[i].ChannelData.Count(); j++)
        {

            data.Add(e.Data[k].SensorData[i].ChannelData[j].Data);

        }
    }

    if (i != 2)
    {
        text = "Conecte los dos sensores por favor";
    }

    //Add frame data to entire collection data buffer
    AllCollectionData.Add(data);
}
```

Fragmento de código 3. Obtención de datos de los sensores

En cada elemento de “data” se almacenan, por tanto, los datos recibidos de un canal distinto.

Cada sensor dispone de ocho canales por los que enviar paquetes por lo que al primer sensor le corresponden los canales almacenados en las posiciones de la cero a la siete, al segundo los almacenados de la ocho a quince y así sucesivamente.

3.3.2 Captar IMU

Como en este juego se pretendía caracterizar el ciclo de la marcha de forma independiente a las señales de electromiografía es necesario obtener también la aceleración o la velocidad angular, aunque para asegurar la robustez del juego se tienen en cuenta las dos.

Esta tarea no fue trivial dado que la nueva API no incluía una lista accesible de los modos de muestreo posibles por lo que fue necesario descompilar la API haciendo uso del descompilador gratuito (dorPeek de JetBrains). Entre los archivos obtenidos al descompilar se encontraba un fichero HTML con una lista de todos los modos posibles, sus nombres y descripciones, de donde se obtuvo la cadena de texto para utilizar el deseado en el juego:

"EMG raw (1259 Hz), skin check (74 Hz), ACC 2g (148 Hz), GYRO 250 dps (148 Hz), +/-5.5mV, 20-450Hz"

Con esto se reciben por los ocho canales disponibles para un sensor y en ese orden las señales de electromiografía, contacto con la piel (sí o no), aceleración y velocidad angular. Estas dos últimas ocupan tres canales cada una, correspondientes a los tres ejes espaciales del sensor, distribuidos de la siguiente forma (Figura 12):

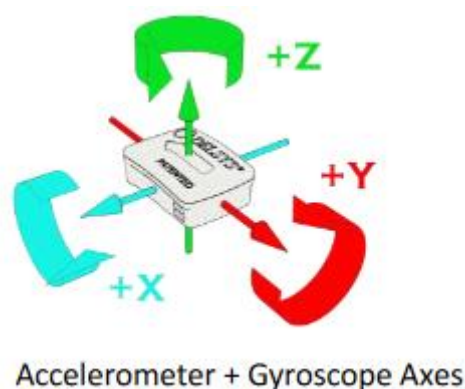


Figura 12. Ejes sensor Delsys (Trigno Wireless Biofeedback System User's Guide, s. f.)

3.3.3 Conocer la frecuencia de datos

A la hora de trabajar a tiempo real es importante conocer la frecuencia de llegada de los datos. Esta frecuencia se obtuvo de manera experimental mediante temporizadores entre llegadas ya que la documentación de Delsys solo cubría lo relativo a su propio software y la información no se correspondía con la realidad del trabajo en Unity.

Se obtuvo que, sin importar el modo de muestreo utilizado, se producían llegadas a una frecuencia de 36 paquetes de datos por segundo. Lo que variaba de modo a modo era el número de datos por paquete. En el escogido para el juego cada paquete de datos EMG presenta 17 datos y cada paquete ACC y GYRO presenta 2 datos.

Además, se descubrió la existencia de un buffer que no se mencionaba en la documentación. Para evitar los efectos de este se comprueba en código que la frecuencia de llegada sea la correcta, desechando los datos hasta el momento en que esa se estabiliza. Esto se consigue mediante el siguiente código (Fragmento de código 4), que descarta el paquete si el tiempo entre su llegada y la del anterior es menor que 1/36 segundos.

```
public virtual void CollectionDataReady(object sender,
ComponentDataReadyEventArgs e)
{
    //Parar el temporizador para obtener el instante de tiempo
    en que llegó este paquete
    stopwatch.Stop();
    //Guardar el tiempo entre paquetes
    TimesTest.Add(stopwatch.Elapsed);
    // Reiniciar el cronometro para el siguiente paquete
    stopwatch.Start();
    //Comprobamos que la frecuencia ya es la deseada (que el
    buffer ya se ha vaciado)
    TimeSpan periodoPaquetes = new TimeSpan(0, 0, 0, 0, 20);
    if (TimesTest.Count() > 1)
    {
        if ((TimesTest[TimesTest.Count() - 1] -
TimesTest[TimesTest.Count() - 2]) > periodoPaquetes)
        {
            ...
        }
    }
}
```

Fragmento de código 4. Código para evitar el efecto del buffer de los sensores (código simplificado)

3.4. Juego concepto final

3.4.1 Calibración

El juego comienza con una etapa de calibración en la que se establecen de forma personalizada para el jugador los valores de ciertos parámetros necesarios para el juego. Esta calibración consta de dos fases.

3.4.1.1 Primera etapa: Calibración estática

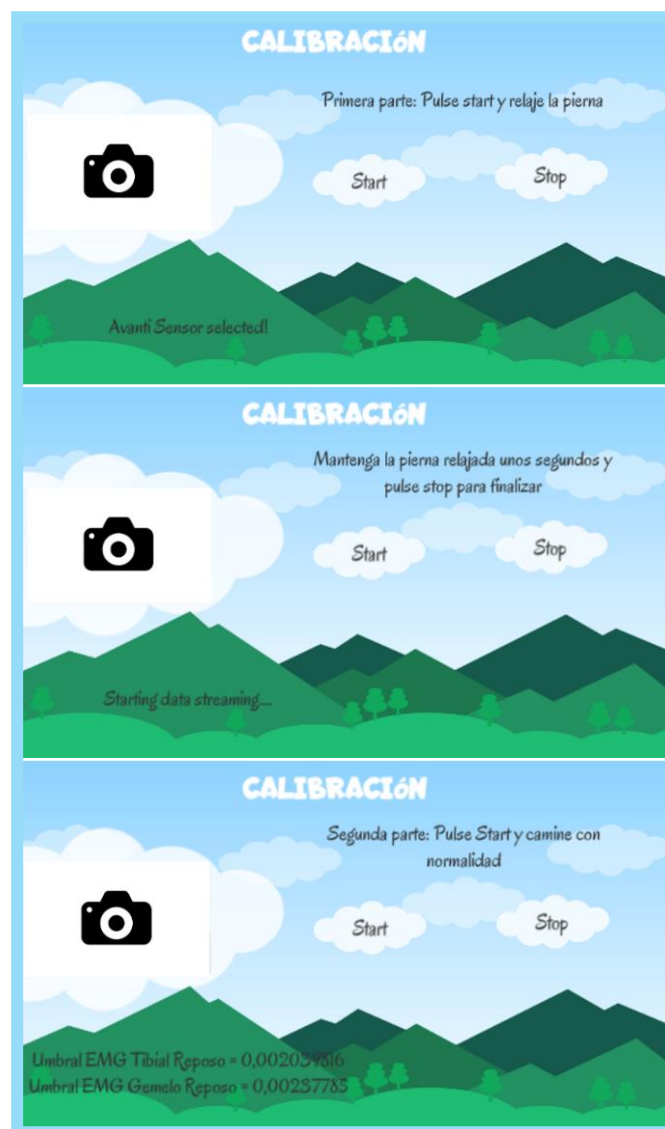


Figura 13. Primera fase de la calibración

La primera parte de la calibración sirve para obtener un primer valor EMG que más tarde servirá para calcular los umbrales de activación usados en el juego. Para ello simplemente se solicita al jugador que permanezca de pie lo más quieto posible durante unos segundos y se recoge la señal de electromiografía.

En las capturas de la Figura 13 se puede observar el proceso completo de esta parte de la calibración. En la primera imagen se muestra el momento en que los sensores se han conectado con éxito y se puede empezar la calibración pulsando el botón “Start”. En ese punto se comienzan a recoger datos y se habilita el botón de “Stop” para que, después de unos segundos, el jugador pueda finalizar la primera fase de la calibración.

Durante la recogida de datos se van almacenando los valores cuadráticos medios de cada paquete recibido, de forma análoga a como se hacía en el otro juego.

```
foreach (var dato in PaqueteEMGPositivo) //A cada dato
le restamos la media del paquete (eliminamos el offset) y
tomamos el valor absoluto (rectificamos)
{
    var datoMinusMean = dato - PaqueteEMGPositivo.Mean();
    var absdato = Math.Abs(datoMinusMean);
    DatosEMG1.Add(absdato);
}

EMGTibial.Add(DatosEMG1.RootMeanSquare() / 0.7);
```

Fragmento de código 5. Obtención de RMS de datos EMG en el juego nuevo

Los botones pueden habilitarse y deshabilitarse de la siguiente manera:

```
Habilitar:           StartButton.enabled = true;
Deshabilitar:       StartButton.enabled = false;
```

Una vez se pulsa el botón “Stop” se calculan las medianas de los datos recogidos y se muestran los umbrales en reposo obtenidos. Además, se proporcionan las instrucciones para continuar a la siguiente etapa de la calibración y se habilita el botón “Start”.

```
if (EtapaCalibración == 0) //EMG reposo
{
    UmbralReposoTibial = (float) (EMGTibial.Median());
    UmbralReposoGemelo = (float) (EMGGemelo.Median());

    text = "Umbral EMG Tibial Reposo = " + UmbralReposoTibial
+ "\nUmbral EMG Gemelo Reposo = " + UmbralReposoGemelo;
    text2 = "Segunda parte: Pulse Start y camine con
normalidad";
}
```

Fragmento de código 6. Código para la finalización de la primera fase de calibración

3.4.1.2 Segunda etapa: Calibración durante la marcha

La segunda parte de la calibración está destinada a obtener los tiempos medios de la marcha del sujeto y los valores electromiográficos de la activación muscular durante la marcha.

Primeramente se obtiene el tiempo que dura un paso y el tiempo de swing, con lo que se puede calcular también el tiempo de stand simplemente restando uno del otro. Es importante que el juego se juegue en una cinta de caminar, además de para que el sujeto pueda ver la pantalla en todo momento, para poder afirmar que estos tiempos de paso se mantienen prácticamente constantes.

El tiempo de paso se obtiene a partir de la señal de aceleración en el eje Z captado en el tibial que tiene el aspecto de la primera señal de la Figura 14.

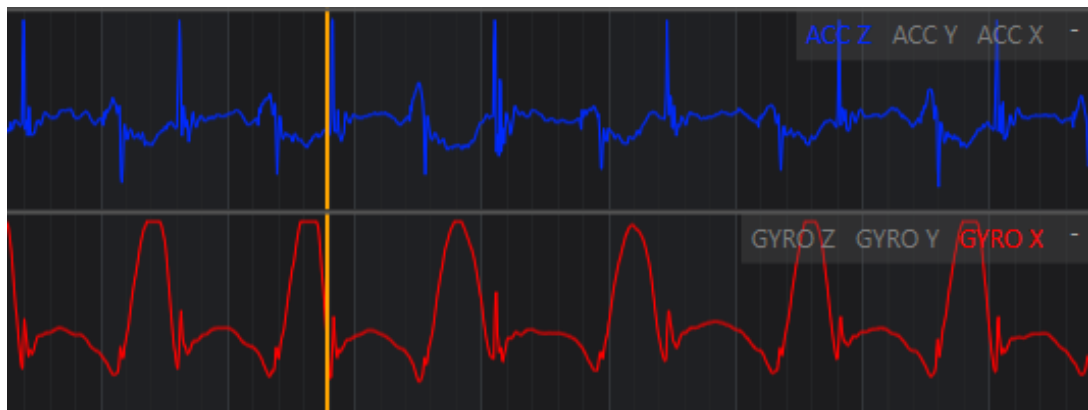


Figura 14. Comparativa entre velocidad angular eje X (GYROX) y aceleración eje Z (ACCZ)

A simple vista se pueden observar unos grandes picos en la señal, uno de ellos marcado por el cursor amarillo. Estos picos coinciden con el contacto con el suelo de cada paso por lo que, localizando los picos de la señal y restando el tiempo entre ellos, podemos obtener el tiempo de cada uno de los pasos.

```
//OBTENCION TIEMPO PASO
// Encuentra los picos y guardamos sus tiempos de llegada
asociados

int i = 0;

for (i = 1; i < datosACCZ.Count() - 1; i++)
{
    if (datosACCZ[i] > datosACCZ[i - 1] && datosACCZ[i] >
datosACCZ[i + 1] && datosACCZ[i] > 0.1)
    {
        peaks.Add(Times[i]);
    }
}
```

Fragmento de código 7. Obtención posibles tiempos de paso

Después de recoger alrededor de 10 pasos para evitar errores, se toma como tiempo de paso la mediana de los tiempos obtenidos.

```
//Calculamos la mediana de los tiempos de paso

TimeSpan Tpaso = new TimeSpan(0, 0, 0, 0);
for (i = 1; i < peaks.Count() - 1; i++)
{
    TpasoPosibles.Add(peaks[i] - peaks[i - 1]);
}

TpasoPosibles.Sort();
if (TpasoPosibles.Count() % 2 != 0)
{
    Tpaso = TpasoPosibles[TpasoPosibles.Count() / 2];
}
else
{
    Tpaso = (TpasoPosibles[TpasoPosibles.Count() / 2 - 1] +
TpasoPosibles[TpasoPosibles.Count() / 2]) / 2;
}
```

Fragmento de código 8. Obtención tiempo de paso final

El tiempo de swing, por otro lado, se obtiene de la señal de velocidad angular.

Como se ha visto en los fundamentos teóricos el paso del stand al swing y viceversa coinciden con mínimos locales de la señal de velocidad angular en el eje x captada en el tibial. Lo ideal sería localizar estos mínimos para conseguir la máxima precisión, pero a tiempo real esto se vuelve bastante complicado.

Por ello surgen técnicas alternativas que toman diferentes puntos de referencia para identificar las fases de la marcha. No todos estos métodos se limitan a dividir el ciclo de la marcha en swing y stand, si no que algunos dividen estas fases en otras más pequeñas, caracterizando la marcha de manera más precisa (Pérez-Ibarra et al., 2020). En el caso de este juego solo se consideró necesario dividir la marcha en sus dos principales fases para no complicarlo en exceso.

Al final la técnica escogida fue la propuesta por Allseits (Allseits et al., 2017). Esta técnica consiste en tomar como puntos clave los pasos por cero de la señal, en lugar de los mínimos locales. Cada paso por cero se toma como un Heel Strike, en el caso de que se pase de valores positivos a negativos, o como un Toe Off, si pasa de negativos a positivos. Para ello se ha utilizado el siguiente código durante la adquisición de los datos (Fragmento de código 9):

```

if (datosGYROX[datosGYROX.Count () - 1] > 0 &&
datosGYROX[datosGYROX.Count () - 2] < 0) //Posible TO
{
    TstartSwing = Tactual;
    text = "swing";
}
else if (datosGYROX[datosGYROX.Count () - 1] < 0 &&
datosGYROX[datosGYROX.Count () - 2] > 0) //Posible HS
{
    TendSwing = Tactual;
    text = "stand";
    TswingPosibles.Add(TendSwing - TstartSwing);
}

```

Fragmento de código 9. Obtención de tiempos de swing posibles

El código anterior divide la señal de una forma similar a la Figura 15, correspondiendo los periodos donde se detecta swing en el código con los de valor 1 en la imagen.

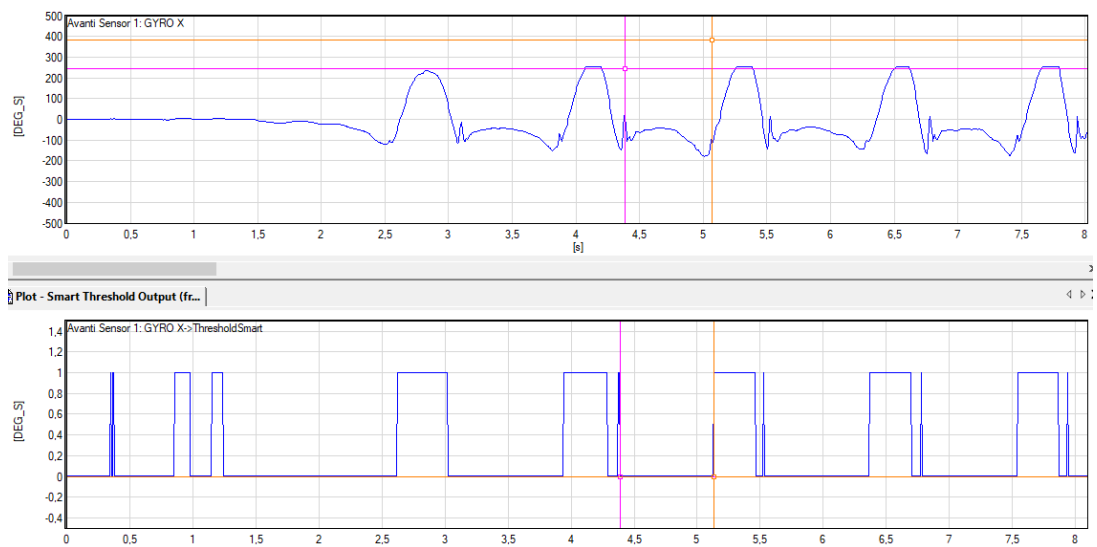


Figura 15. División entre swing (1) y stand (0) a partir de los pasos por cero de la señal GYROX

Este método presenta el problema de que en muchas ocasiones se produce un pico justo después del contacto con el suelo que supera el cero, por lo que se captan dos cambios de estado que no son reales.

En la figura se muestra marcado con el cursor rosa uno de estos picos y su resultado tras el procesado. Como se ve da lugar a un “falso swing” que provocaría tiempos de paso erróneos.

En este juego se optó por ignorar este falso swing descartando de entre todos los tiempos de swing posibles aquellos menores que una décima parte del tiempo total del paso:

```

TswingPosibles.RemoveAll(n => n < Tpaso/10);

if (TswingPosibles.Count() % 2 != 0)
{
    Tswing = TswingPosibles[TswingPosibles.Count() / 2];
}
else
{
    Tswing = (TswingPosibles[TswingPosibles.Count() / 2 - 1] +
TswingPosibles[TswingPosibles.Count() / 2]) / 2;
}

```

Figura 16. Obtención del tiempo de swing final

Una vez solucionado este problema se realizó una comprobación de si los desajustes que provoca tomar el paso por cero como punto de corte son aceptables. Experimentando con diferentes señales se descubrió que la diferencia entre el instante real de cambio entre estados y el instante escogido es menor de 0.1 segundos, como se puede ver en la Figura 17. Este tiempo es lo suficientemente pequeño para que el jugador no aprecie el desajuste por lo que es viable utilizar este método.

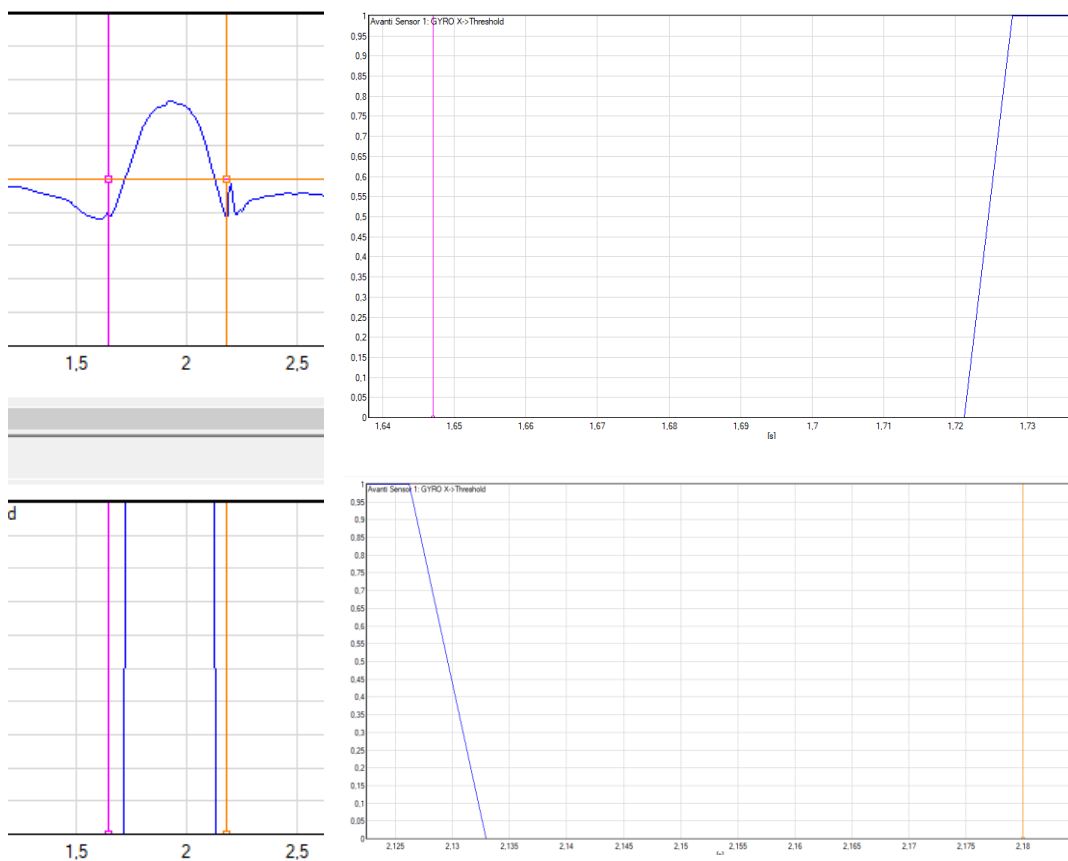


Figura 17. Representación del error al tomar el paso por cero como división entre swing y stand

En cuanto a la parte de electromiografía se recogen los datos de forma diferente a como se hacía en la primera etapa de la calibración. En esta segunda etapa se recogen solo los datos EMG correspondientes a las zonas de activación esperada de cada músculo (tibial durante el swing y gemelo durante el stand) de la siguiente forma:

```

if (datosGYROX[datosGYROX.Count () - 1] > 0 &&
datosGYROX[datosGYROX.Count () - 2] < 0) //Posible TO
{
    TstartSwing = Tactual;
    text = "swing";
    EMGTibialSwing.Add (DatosEMG1.RootMeanSquare () / 0.7);
}
else if (datosGYROX[datosGYROX.Count () - 1] < 0 &&
datosGYROX[datosGYROX.Count () - 2] > 0) //Posible HS
{
    TendSwing = Tactual;
    text = "stand";
    TswingPosibles.Add (TendSwing - TstartSwing);
    EMGGemeloStand.Add (DatosEMG1.RootMeanSquare () / 0.7);
}

```

Fragmento de código 10. Obtención de datos EMG durante los periodos de activación en la marcha

De estas listas se saca la mediana y con ese valor se calculan los umbrales de forma análoga a como se hacía en el otro juego (

Fragmento de código 11) y se guardan mediante las `PlayerPrefs` para poder acceder a ellos desde otros scripts de forma sencilla.

El valor del umbral se obtiene de la suma del valor EMG en reposo más una fracción de la diferencia entre el valor EMG durante la marcha y el de reposo.

De forma experimental se decidió que esta fracción fuese de un 15%, aunque durante el juego se da la opción de variar este valor para ajustar la sensibilidad del juego según las necesidades del jugador.

```

float UmbralMarchaTibial = (float) (EMGTibialSwing.Median());
float UmbralMarchaGemelo = (float) (EMGGemeloStand.Median());

float UmbralTibial = (float) (UmbralReposoTibial +
(UmbralMarchaTibial - UmbralReposoTibial) * 0.15);
float UmbralGemelo = (float) (UmbralReposoGemelo +
(UmbralMarchaGemelo - UmbralReposoGemelo) * 0.15);

PlayerPrefs.SetFloat ("Umbral1EMG", UmbralTibial);
PlayerPrefs.SetFloat ("Umbral2EMG", UmbralGemelo);

```

Fragmento de código 11. Cálculo de umbrales EMG finales

3.4.2 Menú de opciones

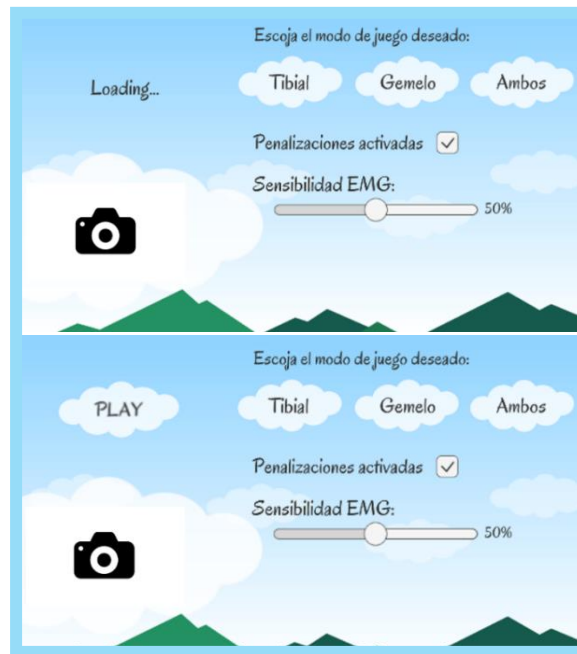


Figura 18. Menú de opciones

Al terminar la calibración y pulsar “Jugar” se presenta un pequeño menú de opciones mientras se configuran en segundo plano los pipelines de Delsys. Durante este tiempo se ajusta el fondo para la siguiente parte del juego mediante una pequeña animación que baja las montañas para simular que se está subiendo al cielo (Fragmento de código 12). De esta forma se consigue más espacio libre en la pantalla y se crea la ilusión de que el pájaro animado que aparecerá más tarde durante el juego está volando por el cielo.

```
public IEnumerator background()
{
    for (int i = 0; i > -30; i--)
    {
        yield return new WaitForSeconds(0.05f);
        Background1.transform.position = new Vector3(0,
Background1.transform.position.y - 1,
Background1.transform.position.z);
        Background2.transform.position = new Vector3(0,
Background2.transform.position.y - 1,
Background2.transform.position.z);
        Background3.transform.position = new Vector3(0,
Background3.transform.position.y - 1,
Background3.transform.position.z);
        ImagenCamera.transform.position = new
Vector3(ImagenCamera.transform.position.x,
ImagenCamera.transform.position.y - 1,
ImagenCamera.transform.position.z);
    }
}
```

Fragmento de código 12. Desplazamiento del fondo

Una vez finalizadas estas configuraciones aparece el botón “PLAY” que permite continuar y comenzar a jugar.

3.4.2.1 Selección del modo de juego

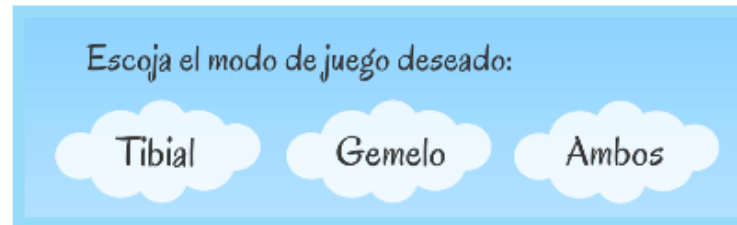


Figura 19. Selección del modo de juego

Para simplificar la experiencia del usuario cuando aún no está familiarizado con la interfaz o simplemente para centrar la atención en el trabajo con uno de los dos músculos se implementa la opción de escoger entre tres modos de juego. La elección es entre trabajar con solo tibial, solo gemelo o ambos a la vez, mostrándose durante el juego únicamente la parte de la interfaz asociada al músculo o músculos seleccionados.

La opción predeterminada es la de ambos músculos y es la que se presentará si no se escoge otra antes de continuar.

3.4.2.2 Control de penalizaciones

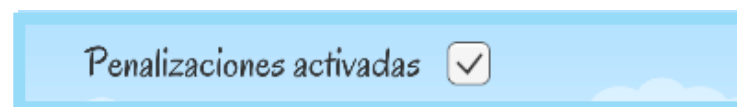


Figura 20. Selector de penalizaciones activadas o desactivadas

El sistema de recolección de puntos del juego ofrece la posibilidad de restar puntos cuando se produzcan activaciones fuera de las franjas esperadas. Esta opción puede ser muy útil si el objetivo con ese sujeto en concreto es fomentar la relajación del músculo, pero también puede llegar a ser frustrante. Por tanto, se dan ambas opciones y se deja a criterio del profesional a cargo el hecho de si penalizar los errores o no.

3.4.2.3 Ajuste del umbral de activación

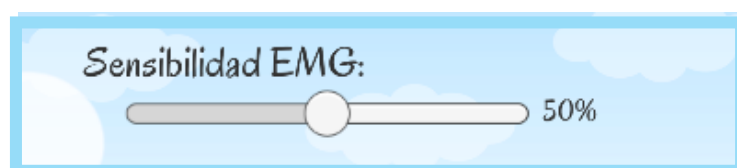


Figura 21. Control deslizante de la sensibilidad EMG

El umbral de activación se calcula durante la calibración de forma que quede ajustado para el jugador, pero es posible que durante el juego este se quiera modificar para conseguir que el juego sea más o menos sensible de lo habitual a la actividad muscular.

Para ello se implementa esta funcionalidad en el menú de opciones mediante el control deslizante de la Figura 21 y el siguiente código (Fragmento de código 13):

```
void SliderValueChanged()
{
    float UmbralReposoTibial =
    PlayerPrefs.GetFloat("Umbral1EMGReposo");
    float UmbralReposoGemelo =
    PlayerPrefs.GetFloat("Umbral2EMGReposo");

    float UmbralMarchaTibial =
    PlayerPrefs.GetFloat("Umbral1EMGMarcha");
    float UmbralMarchaGemelo =
    PlayerPrefs.GetFloat("Umbral2EMGMarcha");

    float UmbralTibial = (float)(UmbralReposoTibial +
    (UmbralMarchaTibial - UmbralReposoTibial) * SliderEMG.value);
    float UmbralGemelo = (float)(UmbralReposoGemelo +
    (UmbralMarchaGemelo - UmbralReposoGemelo) * SliderEMG.value);

    PlayerPrefs.SetFloat("Umbral1EMG", UmbralTibial);
    PlayerPrefs.SetFloat("Umbral2EMG", UmbralGemelo);

    //Valor a mostrar:
    int valor = (int)(100 - (SliderEMG.value - 0.05) * 1000 /
2);
    textSlider.text = valor + " %";
}
```

Fragmento de código 13. Código para modificar el umbral EMG desde el menú de opciones

Cada vez que se cambia el valor del control deslizante, este se utiliza para recalculer el umbral a partir de los valores EMG en reposo y durante la marcha.

De forma experimental se determinó que la diferencia de los valores EMG de marcha y reposo influyese en el umbral en un 15%, pero también se decidió que este valor debería poder cambiarse dentro del juego para poder ajustar la sensibilidad. El control deslizante permite variar el porcentaje entre el 5% y el 25%.

En realidad, al deslizar el control hacia la derecha para aumentar la sensibilidad lo que se hace es disminuir ese porcentaje. Al disminuir el porcentaje baja el valor del umbral por lo que se considera activación valores más bajos de EMG, es decir, se vuelve más sensible el juego.

3.4.3 Pantalla de juego

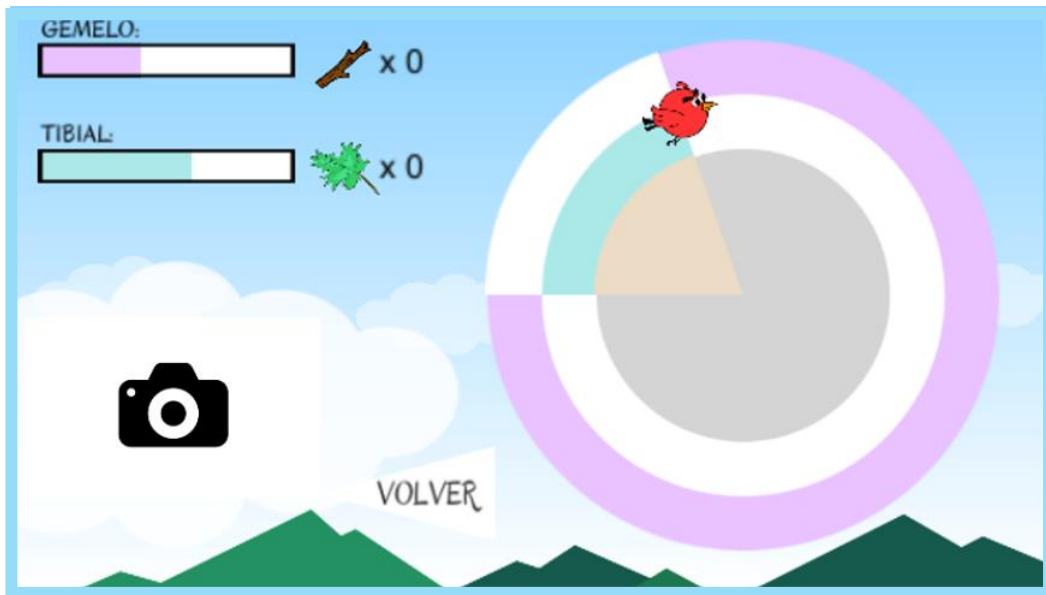


Figura 22. Vista general de la pantalla de juego

Al pulsar el botón “PLAY” del menú de opciones se accede directamente a la pantalla de juego (Figura 22). Para ello se deshabilita el canvas correspondiente al menú y se muestran los correspondientes al juego de la siguiente forma:

```

modoJuego = PlayerPrefs.GetInt("ModoJuego");

CanvasMenuInicio.SetActive(false);
CanvasGame.SetActive(true);
CanvasGait.SetActive(true);

switch (modoJuego)
{
    case 0: //Ambos
        CanvasTibial.SetActive(true);
        CanvasGemelo.SetActive(true);
        break;

    case 1: //Tibial
        UnityEngine.Debug.Log("Tibial");

        CanvasTibial.SetActive(true);
        CanvasGemelo.SetActive(false);
        break;

    case 2: //Gemelo
        UnityEngine.Debug.Log("Gemelo");
        CanvasTibial.SetActive(false);
        CanvasGemelo.SetActive(true);
        break;
}

```

Fragmento de código 14. Paso a la pantalla de juego según el modo escogido

3.4.3.1 Representación del momento de la marcha

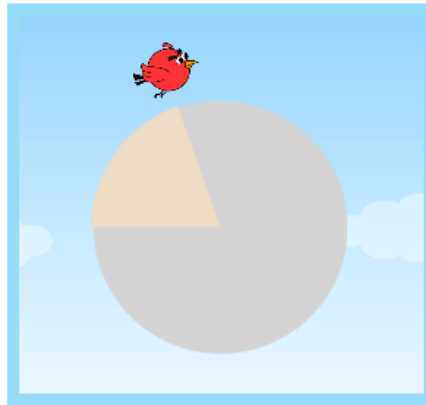


Figura 23. Representación del ciclo de la marcha en el juego

Como se explicaba anteriormente, debido a la naturaleza cíclica de la marcha, se planteó su representación en el juego como algo similar a un reloj, que gira al mismo ritmo que el sujeto camina en la cinta.

Se comienza por representar las fases de la marcha en el círculo. Para ello se parte de los tiempos obtenidos en la calibración, calculando el tanto por uno del paso correspondiente al swing. El swing se representa con un círculo de un color diferente al del stand que se llena el sector correspondiente a dicho valor mediante el siguiente código:

```
CircleSwing.fillAmount = TswingFloat / (TpasoFloat);
```

Fragmento de código 15. Ajuste de la sección rellena del círculo que representa el swing

Para que esto funcione `CircleSwing` tiene que ser un elemento `Image` configurado de la siguiente forma:

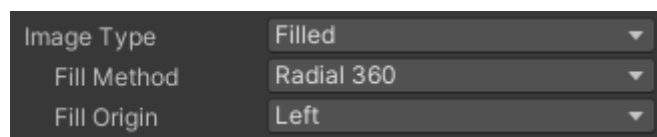


Figura 24. Ajustes en Unity para obtener figuras rellenables de forma radial

Una vez comienza el juego se empiezan a recolectar los datos de velocidad angular. Cada vez que se recibe un paquete se actualiza la fase de la marcha en que se encuentra el sujeto en ese momento de acuerdo con el siguiente diagrama de flujo (Figura 25):

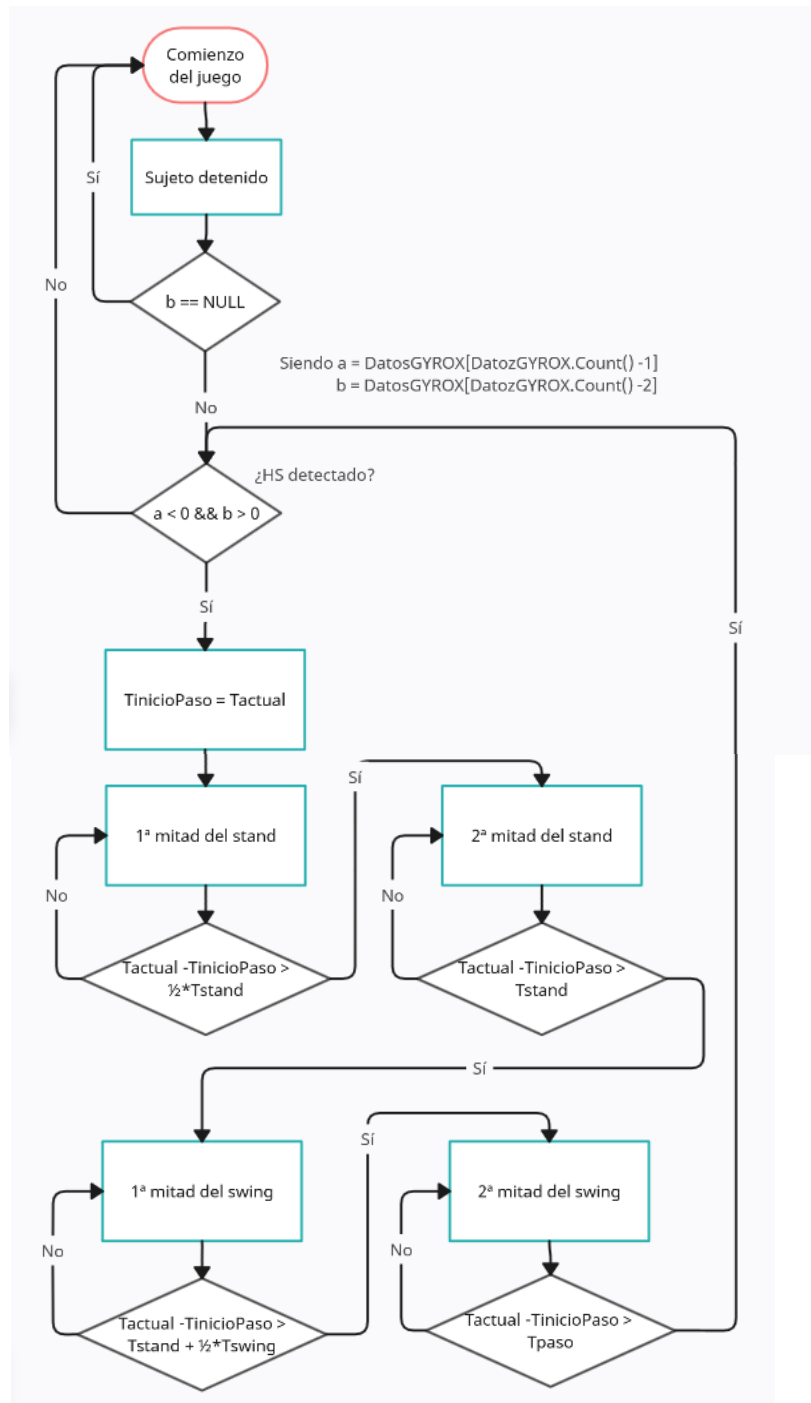


Figura 25. Diagrama de flujo del paso de una fase a otra de la marcha

El paso de una fase a otra dependerá de los tiempos de swing y stand calculados en la calibración, excepto en el caso del paso del swing al stand.

El stand comienza en el momento en que se detecta un contacto con el suelo a partir de la señal de aceleración, de forma similar a como se hacía en la calibración, y se guarda ese instante de tiempo como el comienzo del paso. En el caso de que haya transcurrido un tiempo mayor a la duración de paso estimada desde el comienzo del paso actual se considera que el jugador ha parado de caminar y el juego se detiene hasta detectar el siguiente contacto con el suelo.

Se determina la fase actual según el tiempo transcurrido desde el comienzo del paso siendo las fases en las que se divide la marcha en el programa 1ª mitad del stand, 2ª mitad del stand, 1ª mitad del swing, 2ª mitad del swing, por ese orden. Dado que se calcularon los tiempos de swing y stand en la calibración tan solo hay que compararlos con el tiempo transcurrido desde el comienzo del paso para determinar la fase de la marcha.

```

TimeSpan TimeSpanActualPaso = Tactual -
TInicioPaso[TInicioPaso.Count()-1];
double TActualPaso = TimeSpanActualPaso.TotalSeconds;

if (datosGYROX.Count() > 1)
{
    //2ª mitad swing
    if (TActualPaso > (1 / 2) * TswingFloat + TstandFloat &&
TActualPaso < TstandFloat + TswingFloat && TInicioPaso.Count()
> 1)
    {
        estado.Add(1);
    }
    //1ª mitad stand
    else if (TActualPaso > 0.5 * TswingFloat + TstandFloat &&
datosGYROX[datosGYROX.Count() - 1] < 0 &&
datosGYROX[datosGYROX.Count() - 2] > 0 && TInicioPaso.Count()
> 1)
    {
        estado.Add(0);
        TInicioPaso.Add(Tactual); //Nuevo paso
    }
    //1ª mitad stand del primer paso
    else if (datosGYROX[datosGYROX.Count() - 1] < 0 &&
datosGYROX[datosGYROX.Count() - 2] > 0 && TInicioPaso.Count()
== 1)
    {
        estado.Add(0);
        TInicioPaso.Add(Tactual); //Nuevo paso
    }
    //2ª mitad stand
    else if (TActualPaso > (1 / 2) * TstandFloat &&
TActualPaso < TstandFloat && datosGYROX[datosGYROX.Count() -
1] > 0 && TInicioPaso.Count() > 1)
    {
        estado.Add(0);
    }
    //1ª mitad swing
    else if (TActualPaso > TstandFloat && TActualPaso < (1 /
2) * TswingFloat + TstandFloat && TInicioPaso.Count() > 1)
    {
        estado.Add(1);
    }
}

```

Fragmento de código 16. Fragmentación de la marcha a tiempo real

El código anterior permite dividir el paso en cuatro fases, pero tal y como está planteado el juego solo se utilizan dos de ellas.

Para mostrar este ciclo de la marcha de forma gráfica se crean las funciones “GiroReloj”, “GiroRelojTibial” y “GiroRelojGemelo”, una para cada modo de juego.

Estas funciones son llamadas desde “FixedUpdate”, una variante de la función “Update” que se ejecuta cada 0.02s. De esta manera se actualiza la interfaz de forma constante y rápida, lo que permite reflejar con precisión los datos recibidos, dado que se conoce la frecuencia de refresco de la interfaz y la frecuencia de llegada de los datos desde los sensores.

Para representar gráficamente el instante del paso en que se encuentra el jugador se utilizan otros dos círculos auxiliares de un tono más oscuro que se van llenando de forma radial. Para obtener la sección del círculo que se debe llenar en cada iteración del “FixedUpdate” se divide 0.02s entre el tiempo de paso calculado en la calibración, de forma que el círculo completo tarde exactamente ese tiempo en llenarse. Los detalles de cómo se lleva a cabo este proceso pueden verse en el siguiente código:

```
public void GiroReloj ()
{
    if (estado.Count () > 0) //Se ha detectado el primer HS
    {
        float maxCircleSwing = TswingFloat / (TpasoFloat);
        double UpdateFillAmountDouble = 0.02 / (TpasoFloat);
        float UpdateFillAmount =
        (float)UpdateFillAmountDouble;

        //swing
        if (estado[estado.Count () - 1] == 1 &&
        CircleSwingHighlighted.fillAmount + UpdateFillAmount <
        maxCircleSwing)
        {
            updates.Add(0);
            if (updates[updates.Count () - 2] == 1)
            {
                CircleStandHighlighted.fillAmount =
                maxCircleSwing;

                CircleSwingHighlighted.fillAmount =
                CircleSwingHighlighted.fillAmount + UpdateFillAmount;
                //UpdateFillAmount será lo que se vaya llenando el círculo
                //en cada update.
            }

            else if (estado[estado.Count () - 1] == 1 &&
            CircleSwingHighlighted.fillAmount + UpdateFillAmount >
            maxCircleSwing)
            {
                CircleSwingHighlighted.fillAmount =
                maxCircleSwing;
                updates.Add(2);
            }
        }
    }
}
```

```

//stand
    else if (estado[estado.Count() - 1] == 0 &&
CircleStandHighlighted.fillAmount + UpdateFillAmount < 1)
    {
        .....

        CircleSwingHighlighted.fillAmount = 0;

        CircleStandHighlighted.fillAmount =
CircleStandHighlighted.fillAmount + UpdateFillAmount;
    }

    else if (estado[estado.Count() - 1] == 0 &&
CircleStandHighlighted.fillAmount + UpdateFillAmount > 1)
    {
        updates.Add(1);

        CircleStandHighlighted.fillAmount = 1;
    }
}
}
}

```

Fragmento de código 17. Representación gráfica del estado de la marcha a tiempo real

Además de esta representación del ciclo de la marcha se decidió introducir un pájaro animado para hacer el juego más atractivo para el jugador. Este pájaro gira realizando una circunferencia en torno al centro del círculo que representa las fases de la marcha a la vez que este se va llenando de forma radial.

Giro pájaro durante el stand:

```

BirdPivot.transform.rotation = Quaternion.Euler(0, 0, -
(CircleSwingHighlighted.fillAmount) * 360 + 90);

```

Giro pájaro durante el swing:

```

BirdPivot.transform.rotation = Quaternion.Euler(0, 0, -
(CircleStandHighlighted.fillAmount - CircleSwing.fillAmount)
* 360);

```

Fragmento de código 18. Desplazamiento del pájaro animado

Como se puede apreciar en el código Unity trabaja con cuaterniones, pero dado que trabajar con ellos es bastante complejo se hace uso de una función que transforma a cuaterniones ángulos de Euler, que son mucho más sencillos de manejar.

3.4.3.2 Representación de activaciones musculares durante la marcha

Alrededor del círculo que muestra el momento de la marcha se presentan los correspondientes a la activación muscular, pudiendo estar presentes ambos o solo uno de ellos, dependiendo del modo de juego escogido como se ve en la Figura 26.

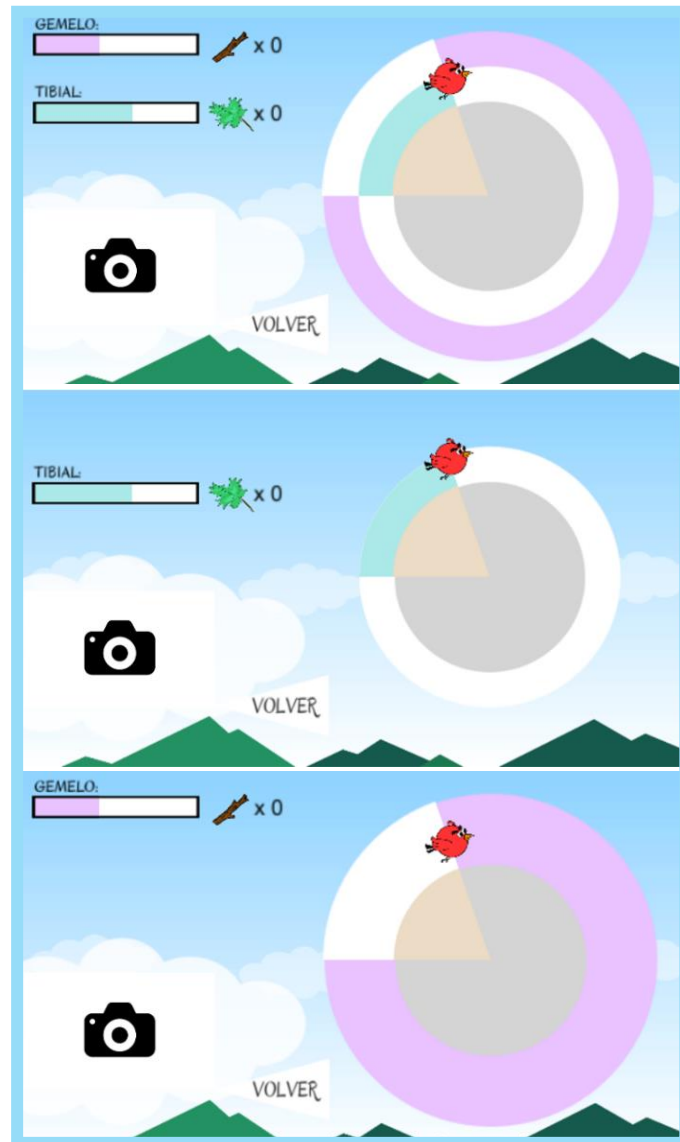


Figura 26. Representación de las zonas de activación esperada en función del modo de juego (ambos, tibial y gemelo por ese orden)

Cada uno de estos círculos tiene una zona coloreada, correspondiente a la zona del paso donde se espera activación, y una zona blanca, que representa la zona donde se espera que el músculo esté en reposo o tenga menos activación. La distribución de estas zonas se realiza en la función “start()” de forma similar a como se representaba el swing en el círculo que representa la marcha.

Ambos círculos se van “coloreando” de un color diferente sección a sección de forma simultánea al del ciclo de la marcha siempre y cuando el músculo esté activado. En caso contrario esa sección quedará sin colorear, viéndose durante el juego situaciones similares a las de la Figura 27.

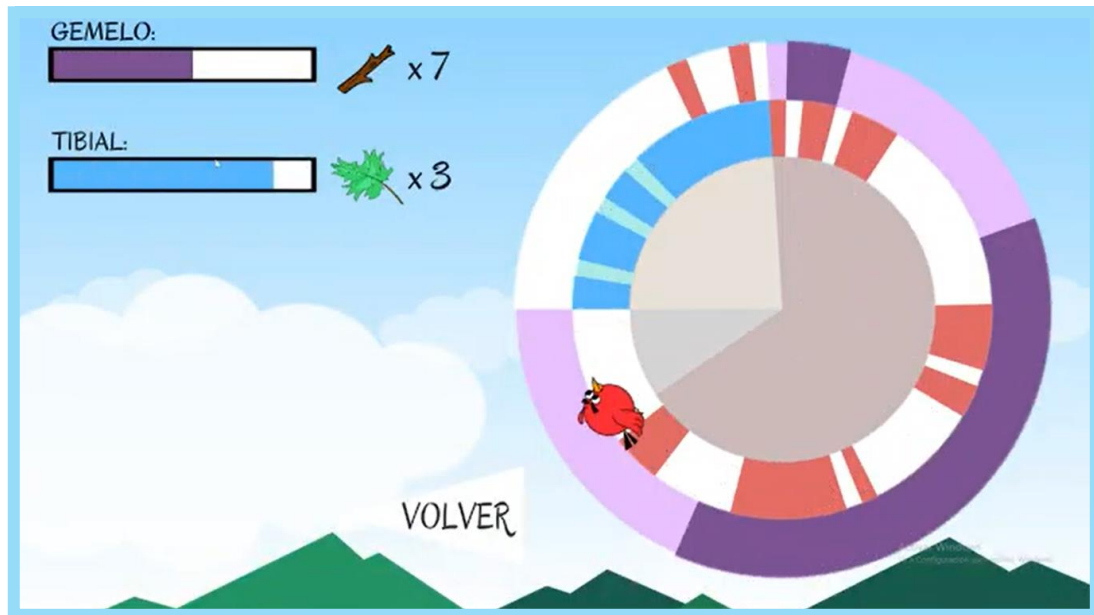


Figura 27. Círculos de activación durante el juego

Como se puede observar en la imagen anterior hay activaciones marcadas en las zonas no deseadas, lo que podría en principio parecer un problema pero que no tiene por qué serlo necesariamente. Como se vio en el apartado teórico diferentes personas tienen diferentes patrones de activación, aunque todos sean sujetos sanos, por lo que lo que se buscará durante el juego no será activación nula en las zonas marcadas en blanco, si no que esta sea no sea demasiado extendida en el tiempo.

En el caso de corresponder la sección donde se detectó la activación a una zona donde se espera activación dicha sección se rellena de un color similar al que ya había, pero más intenso, pero en el caso de ser una sección donde no se esperaba activación esta se rellena de rojo.

Para determinar si el músculo está o no activo en un instante determinado se compara el valor cuadrático medio del último paquete EMG recibido con el umbral de activación del músculo correspondiente.

En cada llamada a la función del giro del reloj se presentan tres opciones en cuanto a la activación de acuerdo con el siguiente diagrama de flujo:

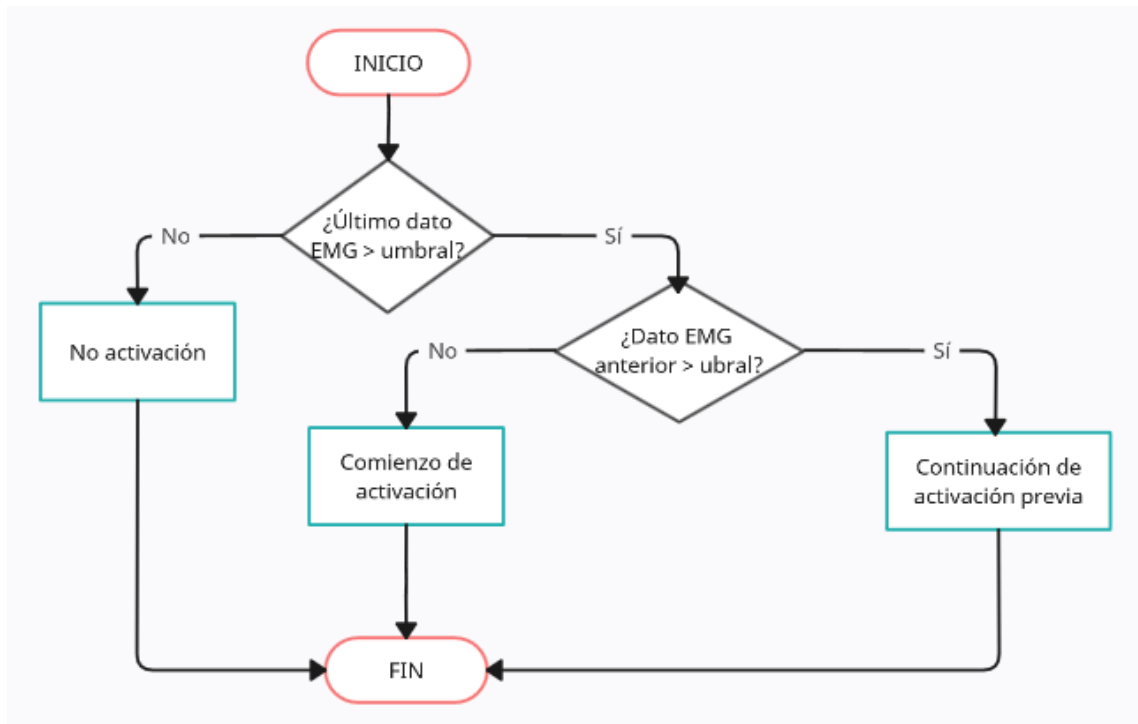


Figura 28. Diagrama de flujo de tipos de activación y sus respuestas

En cada uno de los tres estados posibles se dan comportamientos diferentes:

1. **No activación.** No se modifica el estado de los círculos indicadores de activación.
2. **Continuación de activación previa.** Como el instante de activación viene precedido de otro instante en que el músculo estaba activo simplemente se llenará el siguiente sector del círculo, de tamaño igual al de una actualización de los círculos de la marcha ($0.02s/T_{\text{paso}}$).
3. **Comienzo de activación.** En el caso de que el instante de activación no venga inmediatamente precedido por otro instante de activación el proceso es más complejo.

Lo primero es guardar la activación previa en un nuevo objeto del juego para que se siga mostrando en la pantalla. Para ello se crea una nueva imagen con las mismas características que el círculo original con el que se muestra la activación con el siguiente código, donde `CircleEMGTibialHighlighted` hace referencia al círculo original de activación.


```

PastCircleEMGTibialHighlighted =
(Image) Instantiate (CircleEMGTibialHighlighted,
CircleEMGTibialHighlighted.transform.position,
CircleEMGTibialHighlighted.transform.rotation);
PastCircleEMGTibialHighlighted.transform.SetParent (Cana
sTibial.transform, true);
PastCircleEMGTibialHighlighted.fillAmount =
CircleEMGTibialHighlighted.fillAmount;
PastCircleEMGTibialHighlighted.rectTransform.localScale
= CircleEMGTibialHighlighted.rectTransform.localScale;
PastCircleEMGTibialHighlighted.tag = "clon";

```

Fragmento de código 19. Comienzo de nueva activación

Es importante la última línea de código en la que se etiquetan a las nuevas imágenes como “clon” para poder eliminarlas todas al final de cada paso.

Una vez guardada la activación anterior hay que representar el comienzo de la nueva y, para ello, hay que rotar el círculo de activación original de forma que quede posicionado un sector antes de la posición de la marcha en ese instante. De esta forma, el sector que se llena debido esta nueva activación quedará alineado con el sector del círculo representativo marcha que se llena en esa misma llamada a la función GiroReloj correspondiente.

Al final de cada paso todos los círculos de activación se rotan de vuelta a su posición original y se vacían por completo de forma que se pueda comenzar de cero en el nuevo paso. Además, se eliminan todas las imágenes etiquetadas como “clon”.

3.4.3.3 Barras de activación acumulada

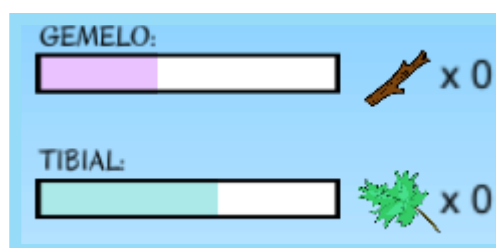


Figura 29. Barras de activación acumulada

En cada llamada a la función de reloj se actualiza las variables correspondientes a la activación acumulada, habiendo una para la activación en zona deseada y otra para la zona no deseada. Ambos valores pueden variar de 0 a 1, siendo 0 que no se ha producido activación en ningún momento y 1 que el músculo ha estado activo durante todo el periodo.

Este valor de 0 a 1 determina también el estado de las barras de activación, que sirven como representación visual más clara de los porcentajes de activación en el paso.

Estas barras presentan además una zona sombreada que determina el porcentaje necesario para que se considere suficiente activación como para puntuar. El tamaño de estas zonas se estableció a partir del estudio mencionado en la fundamentación teórica sobre los múltiples patrones de activación durante el paso en sujetos sanos. (Figura 30)

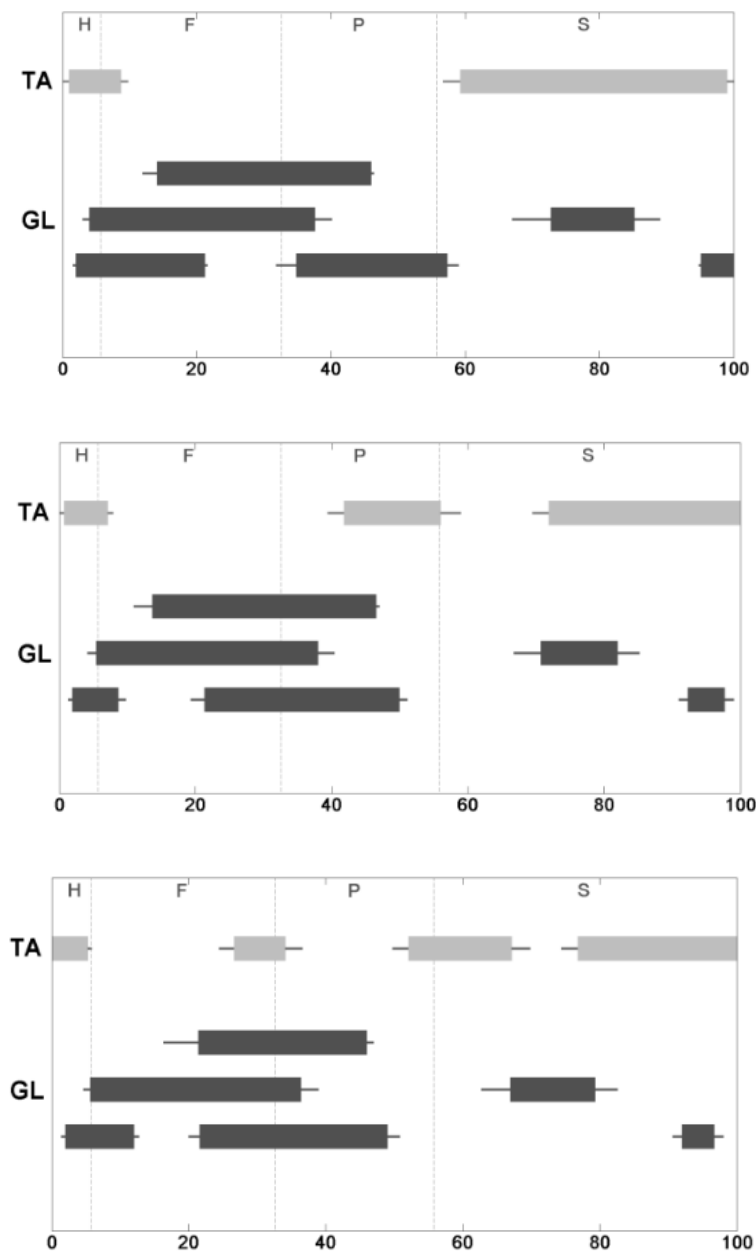


Figura 30. Activaciones en la marcha del tibial anterior (TA) y el gemelo (GL).
Stand: H, F y P. Swing: S (Di Nardo et al., 2014)

Se observó que las activaciones compartidas por todos los sujetos eran al menos durante tres cuartas partes del swing para el tibial y durante algo menos de la mitad del stand para el gemelo, como se puede ver en la Figura 30.

Para dejar algo de margen al jugador se establecieron los umbrales a partir de los cuales la activación se considera correcta en el 60% del swing para el tibial y el 40% del tiempo del stand para el gemelo.

En cuanto a las zonas en las que no se espera activación el comportamiento de un grupo de personas a otro era bastante variado por lo que se decidió poner un umbral que cumpliera el que imponía las condiciones más restrictivas, es decir, aquellos que más activación presentaban en zonas donde habíamos determinado que no se iba a esperar activación. Al igual que en el caso anterior se intentó dar algo de margen al jugador estableciendo el margen algo por encima de lo necesario, fijándolo en un 60% de la respectiva zona de no activación.

El juego solo presenta las barras visuales de activación, sin embargo, para las activaciones correctas con la intención de no reforzar los estímulos negativos y frustrar demasiado al jugador. Esto se decidió así a raíz de estudios como el de Lohse (Lohse et al., 2013b) que sugiere que los pacientes experimentan mayores progresos ante juegos donde el feedback positivo predomina sobre el negativo.

3.4.3.4 Recogida de recompensas y penalizaciones

En cada interacción del FixedUpdate se comprueba si dichos valores de activación acumulada son mayores a los de umbral y si es así se llama a la función correspondiente ya sea para añadir o restar un palo, en el caso del gemelo, o una hoja, en el caso del tibial.

Todas estas llamadas están implementadas con un condicionante adicional que impide que se llame a ninguna de ellas más de una vez por paso (Fragmento de código 20).

```
if (updates[updates.Count() - 2] == 1 &&
updates[updates.Count() - 1] == 0) //Si acaba de comenzar un
nuevo paso
{
    pasoTibial = 1;
    pasoGemelo = 1;
    pasoTibialWrong = 1;
    pasoGemeloWrong = 1;
}

if (BarTibial.fillAmount > BarTibialEsperado.fillAmount &&
pasoTibial == 1)
    AddHoja();
    pasoTibial = 0;
}

if (BarGemelo.fillAmount > BarGemeloEsperado.fillAmount &&
pasoGemelo == 1) // (por ejemplo)
{
    AddPalo();
    pasoGemelo = 0;
}
```

```

if (DropTrue == 1)
{
    if (FillAmountTibialWrong > 0.6 * (1 -
(CircleSwing.fillAmount + (1 - CircleSwing.fillAmount) / 2))
&& pasoTibialWrong == 1)
    {
        DeleteHoja();
        pasoTibialWrong = 0;
    }

    if (FillAmountGemeloWrong > 0.6 * (CircleSwing.fillAmount)
&& pasoGemeloWrong == 1)
    {
        DeletePalo();
        pasoGemeloWrong = 0;
    }
}

```

Fragmento de código 20. Comprobación activación acumulada

Al superar el porcentaje de activación en zona deseada se llama a una función que añade un elemento, es decir, aumenta en uno el contador correspondiente y hace el sonido correspondiente a la recolección de una recompensa (Fragmento de código 21):

```

public void AddHoja()
{
    NumHojas++;
    ContadorTibial.text = "x " + NumHojas;
    CollectAudio.PlayOneShot(collect);
}

```

Fragmento de código 21. Función AddHoja()

Al superar el porcentaje de activación en la zona no deseada la función a la que se llama es similar, solo que el contador disminuye y se reproduce un sonido diferente.

3.4.4 Cámara

Una vez planteado el juego surgió la problemática de que la persona que supervisa el progreso del jugador no podía ver a la vez el caminar del jugador y la pantalla del juego, dado que esta se colca para que pueda verla el jugador. Para superar este inconveniente se incorporó imagen captada a tiempo real para poder ver al jugador por pantalla. Antes de comenzar el juego se debe situar la cámara externa a un lado de la cinta de caminar, de forma que capte las piernas del jugador mientras camina.

Para configurar la cámara se introdujo un desplegable en la etapa de calibración que permite escoger la cámara deseada de entre los dispositivos disponibles como se ve en la Figura 31. Una vez escogida se mostrará de forma continuada durante todo el juego por pantalla para facilitar la experiencia de la persona supervisora.

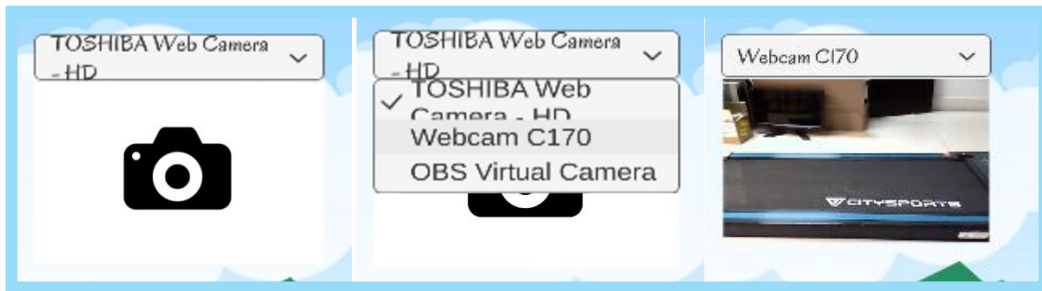


Figura 31. Configuración de cámara

Todo esto se implementa con el siguiente código (Fragmento de código 22). En la función `start` se obtienen los dispositivos disponibles y se comienza a mostrar la imagen del dispositivo por defecto. Además, se configura el desplegable para que muestre como opciones los nombres de estos dispositivos. Cuando se selecciona un dispositivo diferente en el desplegable se ejecuta la función adicional que cambia la fuente de la imagen mostrada por pantalla.

```
void Start()
{
    //Obtener dispositivos disponibles
    WebCamDevice[] devices = WebCamTexture.devices;

    //Inicializar con el primero de ellos
    text = new WebCamTexture(devices[0].name);
    display.texture = text;
    text.Play();

    //Configurar Dropdown
    m_Dropdown.onValueChanged.AddListener(delegate {
        DropdownValueChanged(m_Dropdown);
    });

    m_Dropdown.ClearOptions();
    foreach (WebCamDevice device in devices)
    {
        devicesNames.Add(device.name);
    }
    m_Dropdown.AddOptions(devicesNames);
}

void DropdownValueChanged(TMP_Dropdown change)
{
    text = new WebCamTexture(devicesNames[change.value]);
    display.texture = text;
    text.Play();
}
```

Fragmento de código 22. Configuración de cámara

3.4.5 Datos exportados

El juego proporciona datos exportables para su posterior estudio mediante ficheros .csv. Los datos exportados son los siguientes:

Calibración:

- **EMG Calibracion.csv:** Umbrales EMG obtenidos en la calibración, tanto para el tibial como para el gemelo.
- **Tiempos.csv:** Tiempo de paso, tiempo de apoyo y tiempo de balanceo obtenidos en la calibración.

Juego:

- **Tibial.csv:** Porcentaje de activación en zona deseada del tibial en cada paso.
- **TibialWrong.csv:** Porcentaje de activación en zona no deseada del tibial en cada paso.
- **Gemelo:** Porcentaje de activación en zona deseada del gemelo en cada paso.
- **GemeloWrong:** Porcentaje de activación en zona no deseada del gemelo en cada paso.

Estos datos se exportan mediante fragmentos de código similares al siguiente:

```
using (StreamWriter channelOutputFile6 = new
StreamWriter ("Tiempos" + ".csv"))
{
    channelOutputFile6.WriteLine ("TPaso:");
    channelOutputFile6.WriteLine (Tpaso.ToString());
    channelOutputFile6.WriteLine ("Tswing:");
    channelOutputFile6.WriteLine (Tswing.ToString());
    channelOutputFile6.WriteLine ("Tstand:");
    channelOutputFile6.WriteLine (Tstand.ToString());
}
```

Fragmento de código 23. Código para exportar a .csv

4. Pruebas y modificaciones

A lo largo del desarrollo del juego se realizaron diferentes pruebas para ir comprobando su correcto funcionamiento y realizar los cambios oportunos. De entre ellas destacan las siguientes tres.

4.1 Reconocimiento en tiempo real del swing y el stand

En la segunda fase de la calibración se recogen datos IMU para caracterizar la marcha. Concretamente a partir de la señal de aceleración en el eje X se reconocen los HS y los TO, como se explicaba anteriormente. De esta forma se va haciendo un seguimiento a tiempo real de los cambios entre la fase de balanceo y la de apoyo (cuando se produce un HS) y viceversa (cuando se produce un TO).

Para comprobar que se estaba haciendo correctamente este seguimiento se implementó en el programa un indicador que mostrase durante la segunda etapa de la calibración en cuál de las dos fases de la marcha se encontraba la pierna estudiada. Gracias a la cámara incorporada en la interfaz del juego resultó muy sencillo comparar los cambios de este indicador con el movimiento de la pierna deseada, comprobando que, en efecto, la fase indicada en el juego se correspondía en todo momento con la fase en que se encontraba la pierna del sujeto como se puede ver en las capturas de video de la Figura 32.

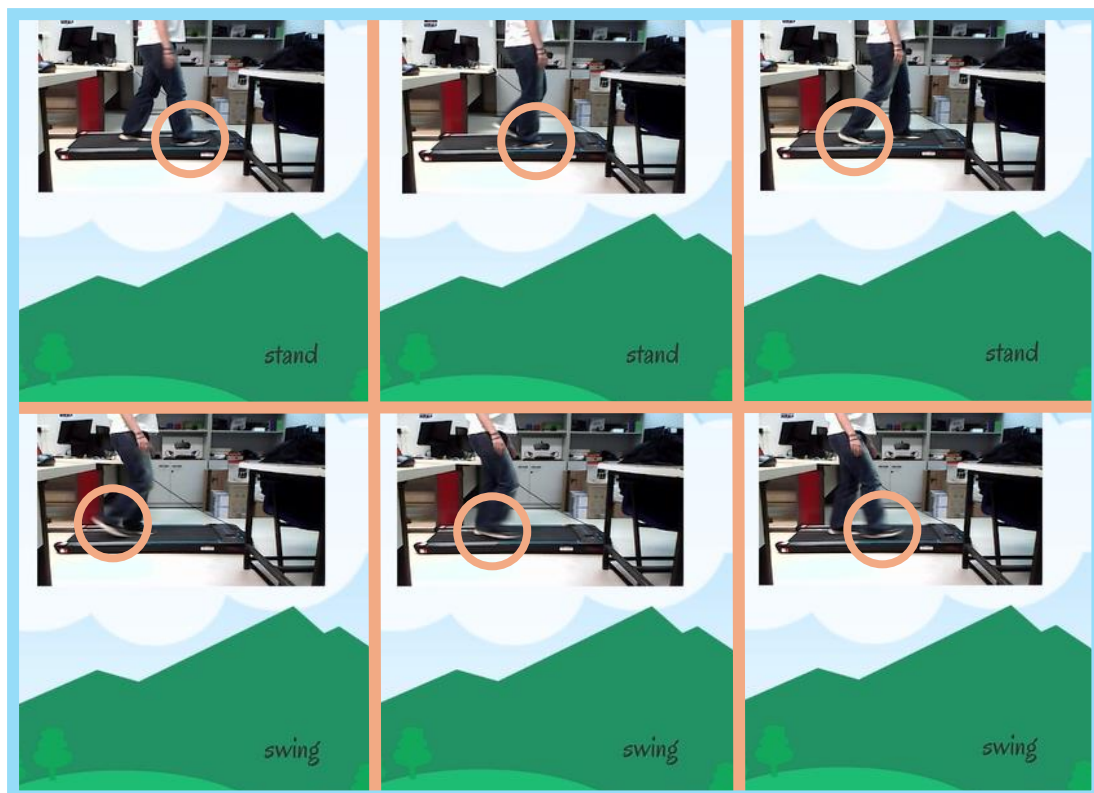


Figura 32. Comparativa entre indicador y fase real de la marcha en video

4.2 Comprobación tiempos de paso y balanceo.

Durante la calibración se calculan los tiempos de paso y de oscilación a partir de las señales IMU y, particularmente para el caso del tiempo de balanceo, se realiza la aproximación de tomar el paso por cero de la señal de velocidad angular en X como contacto con el suelo y como instante en que el pie se separa de este. Aunque teóricamente ya se había comprobado que el error introducido por esta aproximación era prácticamente despreciable para el funcionamiento de este juego, se realizó adicionalmente esta prueba para comprobar que todo funcionaba como debía.

Para ello se tomaron datos IMU de un sujeto caminando a tres velocidades diferentes en la cinta y a partir de ellos se obtuvieron los tiempos medios de paso y de balanceo para cada una de las velocidades, de forma que fuese posible compararlas con los tiempos obtenidos durante la calibración del juego a esas mismas velocidades.

La segmentación de la marcha de las señales captadas fuera del juego se llevó a cabo con el código presentado en el Fragmento de código 24. Con él se localiza el Mid-Swing de cada paso (Figura 33) y se calcula el tiempo de paso como el promedio entre estos.

Posteriormente, y a partir de ese tiempo de paso, se construye una nueva señal producto de la superposición de todos los pasos recogidos (Figura 34) y, a partir de esta, se obtiene el tiempo de balanceo.

```
%Tpaso
figure
findpeaks(Trial2.GYROX,Trial2.times,"MinPeakProminence",100)
xlabel('times')
ylabel('GYROX')
title('Find Prominent Peaks')

[pks, locs] =
findpeaks(Trial2.GYROX,Trial2.times,"MinPeakProminence",100);
peakInterval = diff(locs);
Tpaso = mean(diff(locs))

%Tswing
[pks, locs] = findpeaks(Trial2.GYROX,"MinPeakProminence",100);
peakInterval = diff(locs);
AverageDistance_Peaks = mean(diff(locs));

for j=1:1:AverageDistance_Peaks
    for i=locs(1:end-1)
        x = Trial2.GYROX(i + j);
    end
    Paso(j) = mean(x);
end

figure
findpeaks(-
Paso,Trial2.times(locs(1):AverageDistance_Peaks+locs(1)-1)-
Trial2.times(locs(1)))
xlabel('times')
```



```

ylabel('GYROX')
title('Find Prominent Peaks')

Times = Trial2.times(locs(1):AverageDistance_Peaks+locs(1)-1)-
Trial2.times(locs(1));
cond = Paso(1:end-1).*Paso(2:end);
puntos_clave = find(cond<0);
TswingJuego = Times(puntos_clave(1))+(Tpaso-
Times(puntos_clave(end)))

[pks, locs] = findpeaks(-
Paso,Trial2.times(locs(1):AverageDistance_Peaks+locs(1)-1)-
Trial2.times(locs(1)));
peakInterval = diff(locs);
Tswing = locs(1)+(Tpaso-locs(end))

```

Fragmento de código 24. Segmentación de la marcha fuera del juego

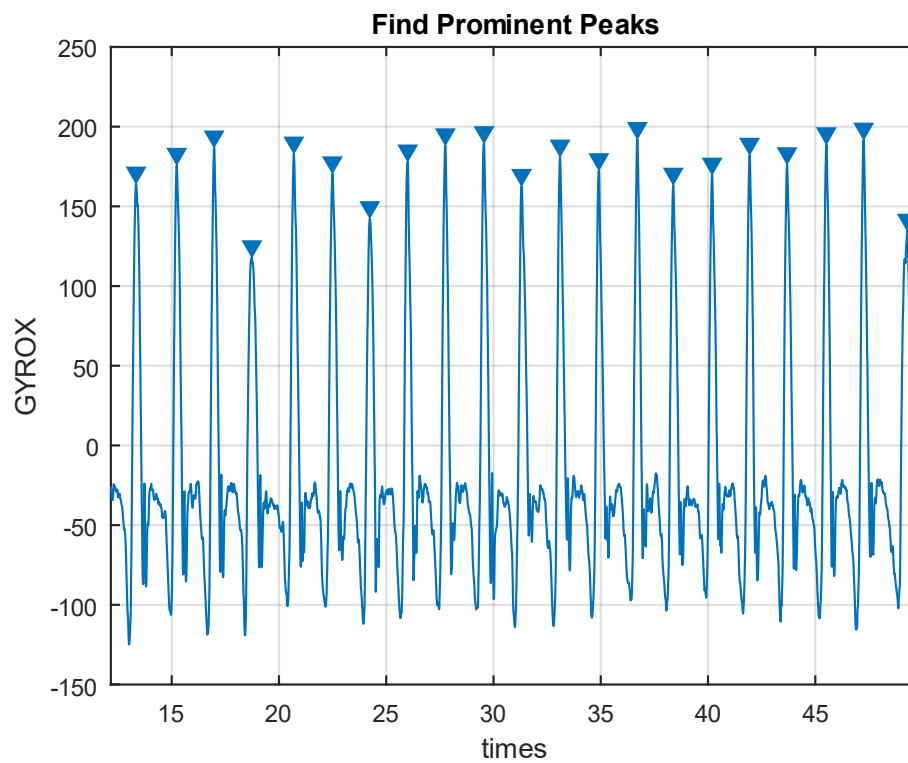


Figura 33. Mid-Swing de cada paso en la señal de velocidad angular en X

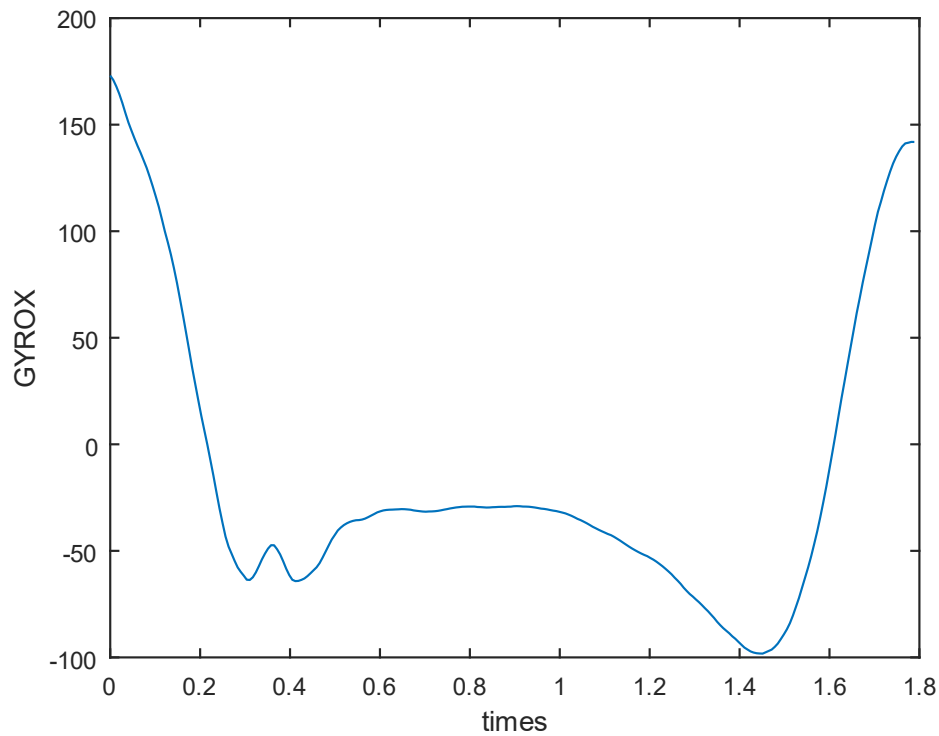


Figura 34. Nueva señal producto de la combinación de todos los pasos

Una vez realizado este proceso para cada una de las tres velocidades de estudio se obtuvieron los siguientes resultados:

	Velocidad de la cinta (km/h)		
	1.5	2	2.5
Tiempo de paso calculado fuera del juego (s)	1.7975	1.5839	1.46
Tiempo de paso obtenido en el juego (s)	1.7547	1.5919	1.4601
Diferencia entre ambos (s)	0.0428	0.0080	0.0001
Error relativo (%)	2.38	0.51	0.01

Tabla 1. Tabla comparativa tiempos de paso

	Velocidad de la cinta (km/h)		
	1.5	2	2.5
Tiempo de swing calculado fuera del juego (s)	0.6565	0.5781	0.5487
Tiempo de swing calculado fuera del juego tomando los pasos por cero (con la aproximación) (s)	0.4003	0.3621	0.3395
Tiempo de swing obtenido en el juego (s)	0.3852	0.3505	0.3353
Diferencia entre ambos (sin aproximación)	0.2713	0.2276	0.2134
Error relativo (%) (sin aproximación)	41.3252	39.3704	38.8919
Diferencia entre ambos (con aproximación)	0.0151	0.0116	0.0042
Error relativo (%) (con aproximación)	2.300	2.01	0.77

Tabla 2. Tabla comparativa tiempos de balanceo

A raíz de las tablas anteriores (Tabla 1 y Tabla 2) se puede observar como el error de alrededor de 0.1 segundos por cada aproximación de un HS o un TO a su paso más cercano por cero se cumple, ya que la diferencia aproximada entre el tiempo de balanceo calculado tomando mínimos frente al calculado tomando pasos por cero, varía en valores cercanos a los 0.2 segundos (la suma del error en el HS y en el TO).

Además, se observa que, aunque los valores obtenidos en el juego, tanto para el tiempo de paso como para el tiempo de balanceo no son iguales, sí son muy cercanos a los esperados. Es lógico que haya pequeñas variaciones entre una muestra y otra de una misma persona caminando en la cinta ya que la marcha de un ser humano no es un proceso exacto, por lo que variaciones de centésimas o incluso milésimas de segundo, como las que se ven en los anteriores resultados, son esperables.

Por otro lado, se comprueba que la aproximación realizada introduce un error que, a pesar de ser prácticamente indetectable para el jugador, supone una variación considerable del tiempo de swing y, por tanto, del porcentaje del paso que abarca la balanceo. Este error se supe a efectos del juego con el comportamiento porcentual del análisis de las activaciones, pero lo ideal sería hacerlo desaparecer o disminuirlo.

Un comportamiento interesante que se detecta es como los errores disminuyen al aumentar la velocidad de la cinta, lo que plantea la posibilidad de que exista una velocidad óptima de juego.

4.3 Ajuste del valor del umbral.

Como se explicaba anteriormente el valor de los umbrales EMG se obtiene a partir de un valor de activación en reposo y otro valor durante la marcha de la siguiente manera, siendo X un valor escogido para obtener la sensibilidad deseada:

$$\text{EMGReposo} + (\text{EMGMarcha} - \text{EMGReposo}) * X$$

En un principio se probó dar a X el mismo valor que tenía en el juego anterior (0.3) pero se comprobó que apenas detectaba las activaciones. Por ello, tanto para la realización de esta prueba como para la creación del control deslizante, solo se tomaron valores por debajo de este. También se descartaron los valores por debajo de 0.05 por detectar como activaciones variaciones demasiado pequeñas de la señal EMG.

Para comprobar cuál era el valor idóneo dentro de este intervalo se realizaron diferentes pruebas y se compararon los resultados. En dos de los sujetos se probó con X = 0.05, 0.10, 0.15, 0.20 y 0.25. Para el tercero, por el contrario, se decidió probar únicamente con los valores centrales, ya que los extremos no parecían aptos para ser el valor escogido.

Cada prueba consistía en hacer caminar al sujeto en la cinta durante 30 segundos sin permitirle ver la pantalla de juego, para que no modificase su conducta y variase los resultados.

A continuación, se recolectaron los datos de los puntos recolectados y perdidos en ese tiempo para poder compararlos y ver bajo que valores de X se obtenían los mejores resultados.

Cabe destacar el hecho de que según la longitud del paso del sujeto su tiempo de paso será ligeramente mayor o menor para una misma velocidad por lo que el número de pasos que da en 30s puede variar de una prueba a otra.

Sujeto 1:

	Activaciones correctas		Activaciones erróneas	
	Gemelo	Tibial	Gemelo	Tibial
0.05	16	6	0	11
0.10	15	11	1	6
0.15	17	14		
0.20	9	9	0	0
0.25	9	7	0	0

(Pasos esperados en 30s: 17)

Tabla 3. Prueba umbrales 1

Sujeto 2:

	Activaciones correctas		Activaciones erróneas	
	Gemelo	Tibial	Gemelo	Tibial
0.05	9	12	0	0
0.10	15	11	0	0
0.15	16	13	0	0
0.20	10	6	6	0
0.25	9	12	7	2

(Pasos esperados en 30s: 16)

*Tabla 4. Prueba umbrales 2***Sujeto 3:**

	Activaciones correctas		Activaciones erróneas	
	Gemelo	Tibial	Gemelo	Tibial
0.10	16	11	0	0
0.15	17	13	0	0
0.20	15	10	0	1

(Pasos esperados en 30s: 17)

Tabla 5. Prueba umbrales 3

En vista de los resultados anteriores se escogió 0.15 como el valor de X predeterminado, ya que es para el que se obtuvieron mayores puntuaciones durante los experimentos. Para valores mayores la sensibilidad era demasiado baja y se perdían activaciones, no sumándose puntos en algunos pasos en donde el comportamiento era correcto. Por otro lado, para valores menores la sensibilidad resultaba a veces demasiado alta, produciéndose pérdidas de puntos en el patrón normal de los sujetos que se suponían sanos.

Equivalencias entre valores de X y los porcentajes de sensibilidad en el juego (control deslizante):

X	Sensibilidad
0.05	100%
0.10	75%
0.15	50%
0.20	25%
0.25	0%

Tabla 6. Equivalencias entre X y sensibilidad en el juego

5. Conclusiones

Los objetivos propuestos en este trabajo consistían en la creación de un juego serio que basado en la obtención de datos mediante sensores medidores de la actividad electromiográfica contribuyera a la rehabilitación de pacientes con trastornos de la marcha producidos por patologías neuromusculares

Tras un estudio de la marcha en sujetos sanos se desarrolló un juego basado en C# que, utilizando datos EMG e inerciales obtenidos mediante dos sensores Trigno Delsys colocados en las piernas, representa el ciclo de la marcha y las activaciones de los músculos estudiados a tiempo real. También se muestran por pantalla los momentos del paso donde se esperaría activación en un sujeto sano. De esta forma, y con ayuda de un sistema añadido de recompensas, el paciente puede ver de manera sencilla su patrón de activación y que comportamientos lo acercan más al patrón no patológico.

El juego serio creado expone de forma visual y auditiva tanto al paciente como al terapeuta los progresos en la variación de la marcha patológica hacia una marcha más normalizada derivada de los cambios tanto voluntarios como involuntarios que el paciente realiza en su caminar para cumplir los objetivos del juego.

A lo largo del desarrollo de este proyecto se ha ido descubriendo que el hecho de caracterizar la marcha mediante señales IMU y evaluar las activaciones musculares de un sujeto mediante EMG durante esta no es una labor tan sencilla como se presentaba en un principio, pero a pesar de ello, se han llegado a resultados bastante satisfactorios y que pueden servir como base para proyectos futuros.

En cuanto a la caracterización de la marcha surgió la complicación imprevista de que, a pesar de caminar en una cinta a velocidad constante, la duración de los pasos varía de uno a otro. Este problema fue atajado realizando promedios de los tiempos obtenidos durante la calibración, obteniendo tiempos medios que permiten el correcto funcionamiento del juego.

Durante el juego se realiza un seguimiento de la fase de la marcha en que se encuentra la pierna estudiada. Para ello se tomó una aproximación de uso muy extendido que simplifica mucho la tarea de segmentar la marcha a tiempo real, demostrándose posteriormente que los efectos de dicha aproximación son indetectables para el jugador.

Por otro lado, el análisis electromiográfico no resultó tan sencillo. La señal EMG es una señal biológica y, como tal, ruidosa y con mucha variabilidad, no solo entre sujetos, si no para una misma persona en diferentes instantes. Realizando algunos procesados sencillos a la señal EMG se pueden suavizar los efectos del ruido y eliminar los posibles datos erróneos, pero no luchar contra esa variabilidad.

Hay muchos factores que pueden influir en las señales EMG e, incluso entre sujetos sanos, los patrones de activación del tibial anterior y el gemelo varían notablemente. Esto provoca que no se pueda establecer un patrón estándar de activaciones esperadas, una de las hipótesis en las que se sustentaba este proyecto.

Para solventar este problema se decidió, en vez de establecer momentos concretos del paso donde se esperase o no activación de determinado músculo, definir zonas más amplias en las

que se espera activación durante un determinado porcentaje de estas. Con esta generalización se cubren los distintos tipos de patrón de la marcha según su número de activaciones y, aunque no es tan preciso como se deseaba en un principio, aún puede cumplir la función de transmitir a grandes rasgos las zonas donde se debe tratar de relajar o contraer el músculo durante la marcha.

Es importante por tanto a la hora de utilizar el juego ser consciente de que las zonas marcadas como activación esperada o no esperada no son unas directrices que seguir a rajatabla, si no unas guías sobre en qué momentos el sujeto debe centrarse más en relajar o contraer el músculo. Que se produzcan activaciones en zonas no esperadas, o que haya fragmentos de las zonas en las que se espera activación en las que no se ha activado el músculo, son comportamientos completamente normales y esperados, siempre y cuando sean en una cantidad razonable, cuantificada por el propio juego y representada mediante las barras de activación y el sistema de puntos.

A pesar de todas estas dificultades el producto final de este proyecto es un juego funcional que hace un seguimiento a tiempo real de la marcha del paciente, recoge y evalúa sus activaciones musculares, representa todos estos datos de forma visual y proporciona una realimentación adicional centrada en transmitir si las activaciones recogidas se mantienen o no dentro de los límites establecidos por el juego, para considerar como normal su patrón de marcha.

El hecho de representar todos estos datos de forma visual puede ser muy beneficioso tanto para el paciente como para el terapeuta, ya que permiten ver a tiempo real de forma muy clara el comportamiento de los músculos estudiados, permitiendo ver los efectos inmediatos de cualquier variación de la marcha del paciente.

Además, el sistema de puntos y el entorno gráfico amigable pueden ayudar al paciente a recibir el feedback de la rehabilitación como algo positivo y agradable, mejorando su experiencia durante el tratamiento.

Por último, a pesar de no haberse hecho pruebas objetivas sobre el efecto del juego en el comportamiento de los jugadores, se ha observado que su marcha varía cuando reciben la retroalimentación del juego. Al caminar, observando la pantalla, los jugadores tienden a modificar, de forma consciente o inconsciente, su marcha para obtener mejores resultados por lo que todo apunta a que el juego está cumpliendo con su intención educacional.

5.1 Líneas futuras

Este juego tiene un gran potencial a futuro en terapias de rehabilitación de la marcha si se refinan ciertos aspectos.

Realizando un estudio más profundo de los distintos patrones de actividad muscular en sujetos sanos durante la marcha, quizá se puedan adaptar de forma más personalizada las zonas de activación esperada. En el caso de que esta personalización no sea posible, tras estudiarla en profundidad, quizá baste con ajustar de forma más rigurosa los porcentajes esperados de activación dentro de las zonas ya definidas, para lograr resultados lo más precisos posibles.

Por otro lado, y como ya se ha dicho anteriormente, este juego está diseñado para ser una herramienta de reeducación de la marcha para el público general. En el caso de querer aplicarla a alguna patología concreta, como pueden ser los niños con parálisis cerebral que tienen dificultades para relajar los músculos, quizá convendría adaptarlo en consecuencia, dando más importancia, por ejemplo, en el caso mencionado, a fomentar la relajación de algún músculo concreto.

En general considero además que con un mayor conocimiento de las señales EMG durante la marcha se podría desarrollar una calibración más específica que proporcionase umbrales de activación más rigurosos.

Por otro lado, en cuanto a la problemática de los tiempos de paso variables para un mismo sujeto a una misma velocidad de la cinta, podría ser una idea interesante incorporar música durante el juego, con la intención de ayudar al paciente a seguir un ritmo constante en sus pasos. Esta adición pasaría por desarrollar una pequeña base de datos con canciones cuyo ritmo se correspondiese a las distintas velocidades de la cinta y solicitar durante el juego que se escogiese la velocidad deseada. En función de la velocidad elegida se mostrarían las canciones disponibles cuyo ritmo se adapta a ella y el jugador podría escoger la que más le gustase, haciendo así además la experiencia más gratificante para él.

También sería recomendable en el futuro ajustar los tiempos de swing y stand durante el juego. Es cierto que el jugador no lo nota, al ser tiempos muy pequeños, pero el desajuste en los porcentajes de apoyo y balanceo que introduce la aproximación puede dar lugar a errores en la interpretación de las activaciones musculares, por lo que sería conveniente intentar disminuir ese error.

Por último, queda pendiente desarrollar un manual de usuario para el terapeuta, que indique los pormenores de la preparación y la ejecución del juego, y quizá también mejorar las instrucciones durante la calibración, que pueden no resultar demasiado claras para el paciente.

6. Referencias

- Abd-alrazaq, A., Alajlani, M., Alhuwail, D., Schneider, J., Akhu-Zaheya, L., Ahmed, A., & Househ, M. (2022). The Effectiveness of Serious Games in Alleviating Anxiety: Systematic Review and Meta-analysis. *JMIR Serious Games*, *10*(1), e29137. <https://doi.org/10.2196/29137>
- Allseits, E., Lučarević, J., Gailey, R., Agrawal, V., Gaunaurd, I., & Bennett, C. (2017). The development and concurrent validity of a real-time algorithm for temporal gait analysis using inertial measurement units. *Journal of Biomechanics*, *55*, 27-33. <https://doi.org/10.1016/j.jbiomech.2017.02.016>
- Angoules, A., Balakatounis, K., Panagiotopoulou, K., Mavrogenis, A., Mitsiokapa, E., & Papagelopoulos, P. (2008). Effectiveness of Electromyographic Biofeedback in the Treatment of Musculoskeletal Pain. *Orthopedics*, *31*. <https://doi.org/10.3928/01477447-20110525-10>
- Basmajian, J. V., & DeLuca, C. J. (1986). *Muscles alive: Their functions revealed by electromyography* (5. ed., 2. print). Williams & Wilkins.
- Campanini, I., Disselhorst-Klug, C., Rymer, W. Z., & Merletti, R. (2020). Surface EMG in Clinical Assessment and Neurorehabilitation: Barriers Limiting Its Use. *Frontiers in Neurology*, *11*. <https://doi.org/10.3389/fneur.2020.00934>
- Cantalapiedra Artiaga, D. J. (2022). *Juego serio para valoración de movimiento y esfuerzo muscular con motor de juegos Unity3D y sensores vestibles*. Trabajo Fin de Grado. Universidad de Valladolid. <https://uvadoc.uva.es/handle/10324/57414>
- Davis, B. L., & Vaughan, C. L. (1999). *Dynamics of Human Gait*. Kiboho Publishers.
- Di Nardo, F., Ghetti, G., & Fioretti, S. (2013). Assessment of the activation modalities of gastrocnemius lateralis and tibialis anterior during gait: A statistical analysis. *Journal of*

- Electromyography and Kinesiology*, 23(6), 1428-1433.
<https://doi.org/10.1016/j.jelekin.2013.05.011>
- Di Nardo, F., Mengarelli, A., Ghetti, G., & Fioretti, S. (2014). Statistical Analysis of EMG Signal Acquired from Tibialis Anterior during Gait. En L. M. Roa Romero (Ed.), *XIII Mediterranean Conference on Medical and Biological Engineering and Computing 2013* (pp. 619-622). Springer International Publishing. https://doi.org/10.1007/978-3-319-00846-2_153
- Dursun, E., Dursun, N., & Alican, D. (2004). Effects of biofeedback treatment on gait in children with cerebral palsy. *Disability and Rehabilitation*, 26(2), 116-120.
<https://doi.org/10.1080/09638280310001629679>
- Electromiografía*. (s. f.). Recuperado 9 de septiembre de 2024, de <http://healthlibrary.brighamandwomens.org/spanish/TestsProcedures/Neurological/92,P09194>
- EMG Signal Analysis*. (s. f.). Recuperado 9 de septiembre de 2024, de <https://delsys.com/downloads/TECHNICALNOTE/103-emg-signal-analysis.pdf>
- Escamilla, R. F., Fleisig, G. S., Zheng, N., Barrentine, S. W., Wilk, K. E., & Andrews, J. R. (1998). Biomechanics of the knee during closed kinetic chain and open kinetic chain exercises. *Medicine & Science in Sports & Exercise*, 30(4), 556.
- Fernández González, C. (2024). *Utilización de señal de electromiografía para complementar el análisis cinemático de la marcha en paciente pediátrico e implementación en un servicio de rehabilitación hospitalario*. Trabajo Fin de Grado, Universidad de Valladolid.
- Fraccaro, P., Walsh, L., Doyle, J., & O'Sullivan, D. (2014). *Real-world Gyroscope-based Gait Event Detection and Gait Feature Extraction*. 247-252.
- Garris, R., Ahlers, R., & Driskell, J. (2002). Games, Motivation, and Learning: A Research and Practice Model. *Simulation & Gaming*, 33, 441-467.
<https://doi.org/10.1177/1046878102238607>

- Hershkovitz, M. (2024, marzo 6). *The Role of Gamification in Enhancing Patient Engagement and Outcomes in Neurorehabilitation*. <https://www.linkedin.com/pulse/role-gamification-enhancing-patient-engagement-mike-hershkovitz-s9kgc>
- Hernández, J. A. F.-T., & Ruiz, C. A. (2006). *Fisiología humana*. McGraw-Hill Interamericana de España.
- Kimura, J. (2013). *Electrodiagnosis in diseases of nerve and muscle: Principles and practice* (4th ed). Oxford university press.
- Konrad, P. (2005). The abc of emg. *A practical introduction to kinesiological electromyography*, 1. https://www.researchgate.net/publication/270895853_The_abc_of_emg
- Lohse, K., Shirzad, N., Verster, A., Hodges, N., & Van der Loos, H. F. M. (2013a). Video Games and Rehabilitation: Using Design Principles to Enhance Engagement in Physical Therapy. *Journal of Neurologic Physical Therapy*, 37(4), 166. <https://doi.org/10.1097/NPT.0000000000000017>
- Lohse, K., Shirzad, N., Verster, A., Hodges, N., & Van der Loos, H. F. M. (2013b). Video Games and Rehabilitation: Using Design Principles to Enhance Engagement in Physical Therapy. *Journal of Neurologic Physical Therapy*, 37(4), 166. <https://doi.org/10.1097/NPT.0000000000000017>
- Lorena Cerda, A. (2014). Manejo del trastorno de marcha del adulto mayor. *Revista Médica Clínica Las Condes*, 25(2), 265-275. [https://doi.org/10.1016/S0716-8640\(14\)70037-9](https://doi.org/10.1016/S0716-8640(14)70037-9)
- Manual práctico de modificación de conducta* (with Vallejo Pareja, M. Á., & Ruiz Fernández, M. Á.). (1998). Fundación Universidad Empresa.
- Michael, D. (2006). *Serious games: Games that educate, train and inform*. Thomson Course Technology.
- Muñoz Saavedra, E. J. (2018). Actualización en: El trastorno neuromuscular, un reto diagnóstico. *Pediatr. Panamá*, 34-42.

- Neptune, R. R., Kautz, S. A., & Zajac, F. E. (2001). Contributions of the individual ankle plantar flexors to support, forward progression and swing initiation during walking. *Journal of Biomechanics*, 34(11), 1387-1398. [https://doi.org/10.1016/s0021-9290\(01\)00105-1](https://doi.org/10.1016/s0021-9290(01)00105-1)
- Pérez Menéndez, A. (s. f.). Día Nacional de las Enfermedades Neuromusculares. *Sociedad Española de Neurología*, 2.
- Pérez-Ibarra, J. C., Siqueira, A. A. G., & Krebs, H. I. (2020). Real-Time Identification of Gait Events in Impaired Subjects Using a Single-IMU Foot-Mounted Device. *IEEE Sensors Journal*, 20(5), 2616-2624. *IEEE Sensors Journal*. <https://doi.org/10.1109/JSEN.2019.2951923>
- Perry, J., & Burnfield, J. M. (Eds.). (2010). *Gait analysis: Normal and pathological function* (2. ed). SLACK.
- Pifarre, F. (2021). *Física y biomecánica clínica para fisioterapeutas y podólogos*. Edicions de la Universitat de Lleida.
- Susi, T., Johannesson, M., & Backlund, P. (2015). *Serious Games—An Overview*.
- Trigno® Avanti Sensor—Delsys. (s. f.). Recuperado 9 de septiembre de 2024, de <https://delsys.com/trigno-avanti/#techspecs>
- Trigno® Lite System—Delsys Europe. (s. f.). Recuperado 9 de septiembre de 2024, de <https://delsyseurope.com/trigno-lite/>
- Trigno Wireless Biofeedback System User's Guide. (s. f.). <https://delsys.com/downloads/USERSGUIDE/trigno/wireless-biofeedback-system.pdf>
- Unity Learn. (s. f.). Recuperado 7 de septiembre de 2024, de <https://learn.unity.com/pathways>
- Whittle, M. W. (2008). *Gait analysis: An introduction* (4th ed., reprinted). Butterworth-Heinemann, Elsevier.