UNIVERSIDAD MIGUEL HERNÁNDEZ DE ELCHE
ESCUELA POLITÉCNICA SUPERIOR DE ELCHE
GRADO EN INGENIERÍA ELECTRÓNICA Y AUTOMÁTICA INDUSTRIAL

DESARROLLO Y EVALUACIÓN DE UN JUEGO SERIO PARA LA REHABILITACIÓN DEL MIEMBRO SUPERIOR ASISTIDO POR ROBOTS

TRABAJO FIN DE GRADO

Junio - 2026

AUTOR: Sonia Barreras Almarcha
DIRECTORES: José María Catalán Orts
David Martínez Pascual

AGRADECIMIENTOS

Gracias a mis padres, por estar siempre para mí y apoyarme de todas las maneras posibles. Por guiarme con su ejemplo, sostenerme y darme fuerzas cuando más lo necesitaba, incluso sin saber que lo hacían.

RESUMEN

La incidencia de ictus en Europa supone un importante problema para la salud pública, provocando la muerte de aproximadamente un tercio de los afectados, y multitud de secuelas cognitivas, motoras y/o de sensibilidad causantes de la pérdida de autonomía de muchos supervivientes.

Entre las más comunes se encuentran la hemiparesia y hemiplegia, que se traducen en la pérdida de fuerza o parálisis absoluta del lateral del cuerpo contrario al hemisferio afectado. Sin embargo, estas lesiones se pueden recuperar significativamente mediante el aprovechamiento de la plasticidad cerebral en las distintas fases tras el accidente a través de la rehabilitación, cuyos resultados mejoran haciendo uso de la asistencia robótica y juegos serios por su impacto en la motivación y la posibilidad de aumentar la repetitividad de los movimientos y ofrecer asistencia según sea necesario.

El enfoque de la terapia varía en función de la fase de la enfermedad, siendo inicialmente crucial poner el foco en la ejecución continúa de patrones de movimientos repetitivos. Durante la fase crónica, una vez las secuelas son menos limitantes, los juegos buscan estimular mentalmente al paciente con mayor intensidad, mediante pruebas de reacción a eventos impredecibles y arbitrarios, o actividades más complejas que involucren el razonamiento, la memoria o la toma de decisiones.

El presente TFG pretende desarrollar un juego serio para la rehabilitación en la fase crónica de los miembros superiores con dos modos posibles (horizontal y vertical), permitiendo la posibilidad de ejercitar mayoritariamente el movimiento del hombro o del codo en cada caso. El diseño del juego incluye una variedad de parámetros que permiten ajustar la dificultad según los requerimientos del paciente.

Los resultados obtenidos en la evaluación del impacto que tienen los parámetros de configuración, sugieren la validez de varios de ellos y la inefectividad de otros.

Índice

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

1. Introducción

El ictus es la segunda causa de muerte y la primera de discapacidad en Europa. Entre un 20 % y un 35 % de los pacientes fallece durante el primer mes tras el accidente cerebrovascular y muchos de los supervivientes pierden su autonomía debido a las secuelas de disfunción motora [1]. La incidencia del ictus muestra una tendencia de crecimiento rápido debido al aumento de la edad de la población.

Por todo esto, es necesaria la investigación en el campo de la rehabilitación, a la cual pretende contribuir este TFG. Concretamente, se centrará en el desarrollo de un juego serio con el propósito de servir de apoyo en las terapias de rehabilitación motora de los miembros superiores, asistidas por el robot Rubidium, en el que se profundiza en apartados posteriores.

1.1. Accidentes cerebrovasculares

1.1.1. Definición, consecuencias y alcance

El ictus, también conocido como Accidente Cerebrovascular (ACV), es un trastorno brusco del flujo sanguíneo cerebral que altera de forma transitoria o permanente el funcionamiento de una determinada región del encéfalo [2]. Existen de dos tipos:

- Ictus isquémico: constituye el 85 % de los casos. Se da por un déficit de riego en el encéfalo debido a un coágulo de sangre que tapona una arteria impidiendo que esta llegue a las neuronas. Este tipo de accidente cerebrovascular es el más común y cuenta con peores expectativas de recuperación.
- Ictus hemorrágico: la ruptura de un vaso sanguíneo, es decir, un derrame, provoca la compresión del tejido cerebral por la sangre. Suele ser más grave inicialmente, pero si el paciente sobrevive tiene un alto potencial de recuperación.

Las consecuencias más comunes tras un ictus son la hemiparesia o hemiplegia, llevando a déficits en el movimiento del lateral del cuerpo contrario al hemisferio del cerebro afectado [3]. Estos trastornos neurológicos se traducen en la disminución de fuerza o la parálisis total, respectivamente. Las principales características clínicas son la debilidad de ciertos músculos, alteración del tono muscular y postura anormal, falta de movilidad, pérdida de coordinación y de sensibilidad.

Aproximadamente el 13 % de las personas con dependencia, han sufrido un ictus, y de aquellas, un tercio presenta un grado de dependencia moderada, el 50 %, dependencia grave, y el 16 %, dependencia absoluta. Constituye por tanto una problemática muy extendida y altamente limitante para los pacientes. Sin embargo, en muchas ocasiones se puede tratar con el fin de mejorar tanto la condición neurológica, como la funcional.

1.1.2. Tratamiento y rehabilitación

La rehabilitación es un proceso limitado en el tiempo, cuyo objetivo es prevenir complicaciones y reducir el déficit neurológico a fin de conseguir la máxima capacidad funcional posible. Esto permite facilitar la autonomía en los diferentes ámbitos de la vida del paciente.

Tras la fase aguda, es decir, a lo largo de entre los 3 y 6 meses posteriores, la neurorrehabilitación representa la única oportunidad de mejora para los pacientes que presentan una discapacidad residual tras el ictus, estimándose que podría aplicarse aproximadamente al 40 % de todos los ictus (isquémicos y hemorrágicos) [4] . La importancia de aprovechar este periodo de tiempo para mejorar los déficits neurológicos y funcionales se debe al aprovechamiento de la plasticidad cerebral en los métodos empleados en la neurorrehabilitación. Las terapias más efectivas son aquellas que proporcionan una estimulación temprana, intensiva, de tareas específicas, repetitivas y multisensoriales [4]. De esta manera se pueden provocar adaptaciones neuronales y mejorar así, la recuperación motora y funcional en las extremidades superiores, tanto en términos de fuerza como de precisión.

Debido a la persistencia de una discapacidad en muchos supervivientes de ictus a pesar de los cuidados médicos y rehabilitadores habituales en los meses iniciales tras el accidente, ha surgido un creciente interés clínico e investigador por comprender el potencial de los tratamientos de rehabilitación más allá del periodo subagudo. Esto es la etapa crónica, definida frecuentemente a partir de los 6 meses post-ictus [4]. Estos tratamientos en constante evolución incluyen tanto terapias intensivas impartidas por un terapeuta especializado como aquellas administradas mediante un robot.

Durante las últimas dos décadas, diferentes robots han sido inventados para ayudar en los procedimientos de de rehabilitación [5]. El uso de robots se propuso para ayudar a los terapeutas precisamente a aumentar la intensidad de las terapias, producir la deseada estimulación multisensorial y reducir los costes durante su trabajo. Los

sistemas robóticos, cada vez más presentes en hospitales y clínicas, proporcionan terapia para periodos largos de tiempo permitiendo hacer un mayor esfuerzo y ahorrar tiempo. Están programados para medir y recoger datos de comportamientos correspondientes a diferentes aplicaciones terapéuticas, y pueden ser usados no solo para producir patrones de movimientos repetitivos y simples, sino para generar una estimulación más compleja y controlada.

La introducción de sistemas robóticos en la práctica clínica es útil promoviendo un uso efectivo de los recursos humanos y la estandarización de tratamientos de rehabilitación. De esta manera, se puede aliviar la carga de trabajo del fisioterapeuta, permitiéndole enfocarse en los aspectos funcionales de la rehabilitación durante entrenamientos individuales y supervisando varios pacientes de manera simultánea durante sesiones asistidas por robot [3].

1.2. Terapias mediante asistencia robótica

Habiendo presentado las ventajas de la utilización de robots en la rehabilitación, en esta sección se profundiza en diferentes aplicaciones de los robots durante los procedimientos. Las terapias se pueden clasificar en pasivas o activas según la implicación del miembro del paciente:

- **Terapia pasiva.** No requiere esfuerzo por parte del paciente y se aplica normalmente en los estados tempranos de los síntomas post-ictus, específicamente cuando no hay respuesta del miembro afectado. Este tipo de terapia está normalmente prescrita para pacientes que sufren de hemiplejia [5] , pues involucra el movimiento repetitivo del miembro afectado siguiendo una trayectoria específica, enfocada en estirarlo y contraerlo y permitiendo evaluar el rango de movimiento de la extremidad. Esta terapia se muestra efectiva reduciendo los espasmos y la rigidez de los miembros afectados.
- **Terapia activa.** Se prescribe a pacientes capaces de mover su miembro afectado hasta cierto límite, durante la fase crónica. Esta terapia se puede subclasificar como terapia activa asistida o terapia activa resistiva.
    1. **Terapia asistida:** implica la aplicación de una fuerza externa ejercida por un robot para ayudar al paciente a completar las tareas, que interviene únicamente cuando este no puede avanzar en la ejecución del objetivo. Este apoyo puede darse de una forma más sencilla, asumiendo que el paso del tiempo implica la necesidad de asistencia, o más compleja y precisa,

por ejemplo mediante la detección de bioseñales (como electromiogramas) que indiquen la intención del paciente de mover el miembro.

2. **Terapia resistiva:** implica la aplicación de una fuerza opuesta al movimiento de los miembros afectados que actúa como resistencia obligando al usuario a hacer más fuerza.

En definitiva, cada terapia está dedicada a condiciones específicas y depende principalmente de las necesidades del paciente.

En cuanto a los sistemas robóticos en sí mismos, durante las últimas dos décadas han surgido muchos diferentes para ayudar en los procedimientos [5]. Estos, se pueden clasificar en función del tipo de anclaje que tienen con el paciente de la siguiente manera: [6] (ver Figura 1.1.)

- **Robots de efector final:** mantienen el agarre del usuario en un solo punto, lo que simplifica el ajuste del robot al paciente, agilizando el proceso de usarlo. Sin embargo, este enfoque no permite una determinación completa de la posición de la extremidad que se está rehabilitando, sino únicamente del punto de unión, pudiendo llevar a cabo movimientos no deseables del resto del miembro.
- **Exoesqueletos:** son estructuras mecánicas con grados de libertad que están alineados con las articulaciones de la extremidad que requiere asistencia o rehabilitación, encapsulando el miembro con una férula o estructura biónica. Esta alineación permite un control directo sobre el movimiento de la extremidad. De esta manera, los exoesqueletos ofrecen la ventaja de una determinación completa de la posición del brazo, ya que el robot se ajusta específicamente a la extremidad que se está rehabilitando obligando a que esta lleve a cabo un buen patrón de movimiento.

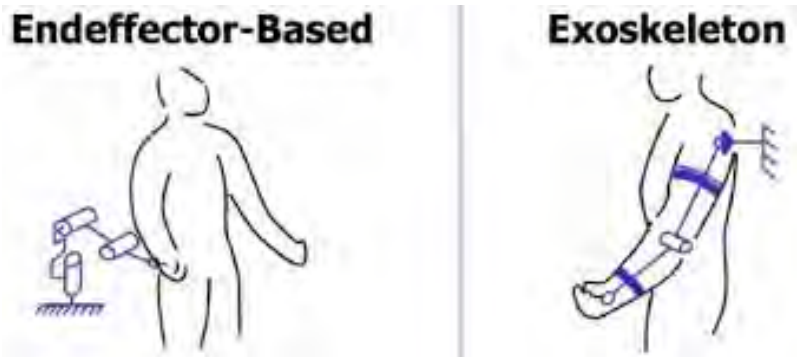


Figura 1.1: Robot basado en efector final vs exosesqueleto.

Estas formas de clasificación se verán reflejadas en una sección posterior sobre antecedentes robóticos.

1.3. Juegos serios en rehabilitación

El término "juegos serios", hace referencia a aquellos con un propósito principal diferente al puro entretenimiento [7]. Han sido desarrollados en distintos campos como la educación, publicidad, ecología, cuidado de la salud o rehabilitación.

En el campo de la salud y la rehabilitación, gran parte de los dispositivos robóticos de los que se habla de forma general en la sección anterior, y de aquellos que se mostrarán en la sección de antecedentes, hacen uso de juegos serios como elemento imprescindible de la terapia. Ofrecen la posibilidad de tener feedback en vivo de la actividad, mejorando el rendimiento motor. Además, al darle un sentido a la tarea, producen un incremento de la motivación [8], lo que permite realizar mayor número de repeticiones, maximizando la posibilidad de recuperación total. De acuerdo con unos estudios que grabaron los movimientos del paciente durante una sesión con juegos serios y otra tradicional, en la primera realizaban hasta 6 veces más ejercicio.

Para logar la motivación de la que se habla, es indispensable disponer de diferentes tipos de juegos. La repetición de la misma actividad puede resultar monótona, por lo que es crítico contar con un amplio repertorio. De esta forma los terapeutas pueden elegir entre los diferentes juegos disponibles en función de la habilidad del paciente, y este puede escoger también de acuerdo a sus preferencias.

Todos los juegos desarrollados con propósito terapeútico tienen que cumplir con ciertos patrones de diseño. Los puntos clave a nivel de diseño son los siguientes:

- **Dificultad adaptable:** si el juego supone un reto demasiado grande, se crea aversión. Si por el contrario es demasiado fácil no genera interés. El nivel de dificultad del juego debe poder adaptarse de forma independiente a cada paciente para que el uso de este sea eficaz. Esto se puede conseguir de diferentes maneras: mediante parámetros que ajusta el terapeuta en función de las necesidades del usuario, o bien a través de técnicas para regular el nivel de forma automática, como por ejemplo puede ser un aumento o disminución de la velocidad en función de si el nivel anterior se ha superado con éxito o no.
- **Características de la interfaz:** si bien se busca que esta sea atractiva, en función de la capacidad de atención del paciente la interfaz debe ser , en mayor

o menor medida, simple, ya que si es demasiado compleja puede confundir al usuario y no permitir que desarrolle los objetivos correctamente. Se debe tener en cuenta que las personas objeto de este tipo de terapias han sufrido daños neurológicos y que, por tanto, la complejidad adecuada de la interfaz debe ser, en general, baja.

- **Aspectos de la rehabilitación a nivel motor:** existen diferentes enfoques que puede tener un juego de acuerdo con el tipo de movimiento concreto que se desea entrenar: movimientos simplemente horizontales o verticales, otros para alcanzar objetos con precisión siguiendo diferentes patrones de movimiento, o aquellos libres y arbitrarios que sirven para entrenar la capacidad de reacción ante situaciones impredecibles para el paciente, en los que entra en juego su capacidad de decisión, lo cual se relaciona con el siguiente punto.

- **Aspectos de la rehabilitación a nivel neurológico:** en cualquier caso, el juego debe potenciar que el paciente ejerza también actividad neurológica a través de un intento activo de conseguir los objetivos. Puede llevarse a cabo mediante actividades dónde se ejercite la memoria, la identificación de objetos, la comprensión lectora, etc. Este elemento del diseño de los juegos es el que provoca resultados mucho mejores que una rehabilitación enfocada únicamente en el movimiento físico.

- **Número de jugadores:** los juegos pueden desarrollarse para un único jugador, o ser multijugador. El segundo caso puede involucrar competitividad y cooperatividad. La interacción social de juegos multijugador es un elemento potencial de aumento de la motivación, el disfrute y la intensidad del ejercicio de los pacientes, lo que maximiza los resultados de la rehabilitación [9]. Un modo de juego competitivo es similar a nivel de intensidad a uno individual de alta dificultad, pero el estrés provocado en el paciente es menor, más parecido a uno individual de baja dificultad, siendo por tanto más beneficioso.

La mayoría de juegos serios presentes en el campo de la rehabilitación clínica son de gamificación convencional [8]. Es decir, en la actualidad se usan prácticamente en su totalidad los juegos 2D y 3D, con los que el paciente interactúa mediante la conexión del robot con el PC. Sin embargo, para finalizar la sección es preciso señalar la reciente introducción de la realidad virtual (RV) y realidad aumentada (AV) en el ámbito de la investigación, siendo la segunda muy escasa e innovadora.

1.4. Antecedentes

Este TFG se centra en la rehabilitación de las extremidades superiores. A continuación, se comentan algunos antecedentes de robots destacados enfocados en distintas partes del brazo o del miembro completo, tanto exoesqueletos como robots de efector final. Más adelante en esta sección se presentan diversos juegos serios habituales en la actualidad.

1.1. AMADEO.

Este sistema desarrollado por TyroMotion está diseñado específicamente para la rehabilitación de la muñeca y los dedos de la mano. Su funcionamiento se basa en un mecanismo de deslizamiento lineal automatizado que se acopla magnéticamente a las puntas de los dedos del paciente, adherido junto con un soporte para el antebrazo a una base fija como se muestra en la Figura 1.2.

El dispositivo permite ejecutar terapias tanto pasivas (para pacientes con espasticidad severa o sin movilidad) como activas-asistidas, adaptando la resistencia y la fuerza de forma personalizada. El sistema integra sensores de fuerza y de rango de movimiento que registran de manera continua dichas variables, facilitando el seguimiento clínico del paciente. Además, la plataforma de software del dispositivo combina estas terapias con entornos virtuales de juegos interactivos (juegos serios), por los motivos que se comenta en la sección anterior.

(a) Dispositivo robótico AMADEO.

(b) Acople al paciente.

Figura 1.2: Sistema robótico AMADEO para la rehabilitación de la mano y los dedos: (a) vista general; (b) acople magnético en el paciente [10].

1.2. NEEM.

El NEEM (NEUROExos Elbow Module) es un exoesqueleto robótico activo y vestible (wearable) para la rehabilitación del codo, desarrollado originalmente por el Wearable Robotics Lab de la Escuela Superior de Santa Ana en Italia en colaboración con la empresa IUVO (ver Figura 1.3).

Entre sus características de diseño principales destaca la integración de un actuador elástico en serie (SEA), el cual le permite monitorizar y regular el torque que ejerce con alta precisión para realizar la flexión y extensión del codo (único grado de libertad activo), así como un mecanismo pasivo de cuatro grados de libertad que permite alinear de forma automática los ejes mecánicos del robot con el brazo del paciente para garantizar una terapia segura y libre de presiones dañinas [11].

A diferencia de otros sistemas de rehabilitación comercializados, el NEEM está diseñado como una herramienta clínica orientada a medir y contrarrestar la rigidez articular y la espasticidad muscular. Por lo tanto, no utiliza juegos serios o entornos virtuales interactivos.

Figura 1.3: Exoesqueleto NEEM [12].

1.3. ARMIN Y ARMEO POWER.

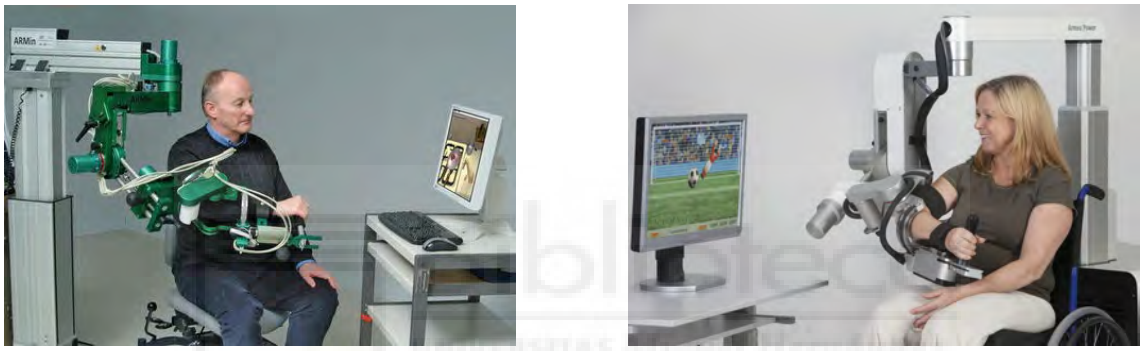
El sistema ARMin, representado en la Figura 1.4a es un exoesqueleto robótico motorizado de base fija que ofrece asistencia según sea necesario, diseñado originalmente en el entorno académico para la rehabilitación del miembro superior completo. Posteriormente, la empresa Hocoma lo comercializó y adaptó el diseño para la utilización de los terapeutas en entornos reales, bajo el nombre de ArmeoPower [13] (ver Figura 1.4b).

A diferencia de los sistemas basados en efectores finales, envuelve todo el brazo para controlar de forma precisa cada una de las articulaciones del miembro afectado. Cuenta con entre 6 y 7 grados de libertad, lo que le permite realizar casi todos los

movimientos del brazo humano: flexión/extensión y abducción/aducción del hombro, rotación interna/externa del brazo, flexión/extensión del codo, pronación/supinación del antebrazo y flexión/extensión de la muñeca [14].

Para garantizar una terapia segura y adaptada, el sistema cuenta con algoritmos de control que calculan y compensan el peso del propio exoesqueleto y, de ser necesario, el del brazo del paciente. Utiliza sensores de fuerza y posición que detectan la intención de movimiento del usuario para modular dicha asistencia.

El sistema integra una pantalla donde se muestran entornos virtuales inmersivos y tareas orientadas a objetivos (serious games), como alcanzar objetos virtuales o realizar actividades de la vida diaria. Esto fomenta la plasticidad neuronal gracias al biofeedback visual y acústico en tiempo real.



(a) Exoesqueleto ARMin [14]. (b) Versión comercial Armeo power [13].

Figura 1.4: Sistemas robóticos ARMin y Armeo power.

1.4. ARMEO SPRING PRO

También desarrollado por Hocoma, Armeo Spring Pro, representado en la Figura 1.5, se diferencia del Armeo power en que es un exoesqueleto pasivo, que el lugar de motores para asistir el movimiento, cuenta con muelles ajustables que permiten trabajar partiendo de la movilidad residual propia del usuario, soportando el peso del brazo y midiendo los movimientos durante los ejercicios. Esto es posible ya que está destinado a pacientes con impedimentos leves/moderados, tanto adultos como pediátricos [15].

Estos ejercicios motivadores son similares a juegos (también usan juegos serios) con el objetivo de aumentar el rango de movimiento, la fuerza y la coordinación del movimiento del brazo dañado.

Figura 1.5: Exoesqueleto pasivo Armeo Spring Pro [16].

1.5. InMotion ARM.

Siendo uno de los ejemplos más exitosos en los años recientes, usándose en 20 países alrededor del mundo [17], es la versión clínica del proyecto MIT-Manus del Instituto Tecnológico de Massachusetts, un robot de efector final activo que tiene el objetivo de la mejorar el rango de movimiento, fuerza, coordinación, velocidad y suavidad del movimiento. Con este sistema se puede realizar la flexión/extensión de codo, rotación interna/externa así como la flexión/extensión, aducción/abducción del hombro.

InMotion ARM, representado en la Figura 1.6, emplea un sistema de control que rastrea el rendimiento del paciente durante los primeros movimientos, ayudándole si fuera necesario, y modificando el ejercicio de forma automática según sea la respuesta, para que no sea monótono, por ejemplo reduciendo el tiempo asignado para realizar el movimiento según la evolución o variando ligeramente algún parámetro [18]. También hace uso de juegos serios para las sesiones.

Figura 1.6: Robot de efector final InMotioArm [19].

Habiendo presentado algunos de los robots que se mencionan dentro del panora-

ma actual de la rehabilitación de las extremidades superiores, se puede ver que la mayoría hace uso de la gamificación como técnica esencial para la rehabilitación neurológica.

Eodyne es un ejemplo de empresa que se dedica a desarrollar únicamente software que se conecta al hardware de terceros [20]. Han desarrollado una infraestructura tecnológica que permite cubrir todo el proceso de recuperación del paciente, desde que está ingresado en el hospital hasta que continúa la rehabilitación en su propia casa, donde los profesionales de la salud lo pueden monitorizar [21], conocido como RGS Ecosystem (o ecosistema Rehabilitation Gaming System). Ofrece así una solución de continuidad asistencial basada en la nube y la inteligencia artificial permitiendo aumentar el número de horas dedicadas a la terapia.

Durante la fase hospitalaria sirve de apoyo motivacional del hardware, mientras que cuando el paciente recibe el alta y recupera algo de movilidad voluntaria, se va a su casa y continúa su recuperación usando únicamente el software de Eodyne (en portátiles, tablets o smartphones) con la webcam, que detecta ese movimiento y mueve el avatar dentro del juego en tiempo real gracias a los algoritmos de visión por computador, permitiéndole realizar un entrenamiento de alta intensidad de forma autónoma, barata y diaria, como se ve en la Figura 1.7.

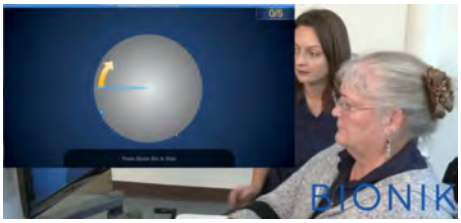
Figura 1.7: Juego de Eodyne para telerehabilitación [22].

Sin embargo, este TFG se centra en juegos diseñados para usar con robots. A continuación, se muestra una clasificación de estos con ejemplos concretos de cada tipo, tanto de aquellos que emplea la empresa Bionik y otros desarrollados por la Universidad Miguel Hernández. La clasificación sigue los criterios comentados en el apartado de reglas de diseño de los juegos:

- **Juegos de alcanzar objetivos:** en este tipo de juegos el paciente debe seguir trayectorias predefinidas, por ejemplo en actividades de trazar patrones como circunferencias o caminos rectos (ver Figuras 1.8b y 1.8c), o dibujar uniendo puntos, o en el juego de la ruleta (ver Figura 1.8a) en todas sus versiones, el

juego más típico y antiguo, ya inventado para el MIT-MANUS. Este consiste en un círculo con 8 objetivos cercanos al perímetro y un punto central. El paciente debe mover el cursor desde el centro hacia una de las luces que se ilumina y volver al centro, entrenando así en alcance en todas las direcciones del plano horizontal. En cualquiera de estos casos el robot puede prestar asistencia de manera que si la persona se desvía de la trayectoria, esta sea corregida, buscando entrenar el control fino y la suavidad del movimiento.

(a) Juego de tipo Ruleta [23].



(b) Trayectoria circular [23].

(c) Trayectorias rectas [24].

Figura 1.8: Ejemplos de juegos interactivos orientados a alcanzar objetivos.

- **Libres:** el paciente no debe seguir una trayectoria concreta preestablecida, sino que elije como ejecuta movimiento en función de si debe alcanzar o esquivar objetos con posiciones/desplazamientos cambiantes o arbitrarios. También hay muchos juegos de este tipo (ver Figura 1.9), al que pertenece el desarrollado en este TFG. Para esta categoría no se puede ofrecer asistencia total, al contrario que para la anterior, ya que forma parte del diseño que el usuario pueda equivocarse. Tienen como objetivo incrementar las velocidades y la intensidad del movimiento.

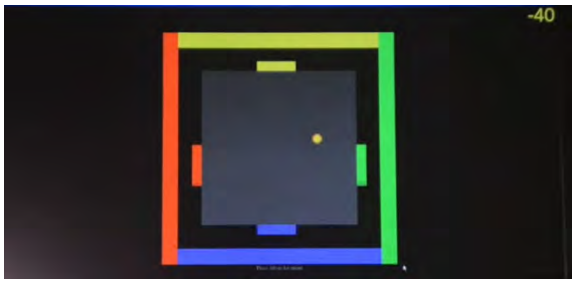


Figura 1.9: Ejemplo de juego libre.

- **Objetivo principal cognitivo:** existen otros juegos cuyo propósito principal

es ejercitar la mente más allá de los movimientos físicos, como por ejemplo laberintos en los que hay que encontrar la salida, identificar elementos o encontrar las parejas de imágenes entre varias cartas como se ve en la Figura 1.10.

Figura 1.10: Ejemplo de juego que entrena la memoria.

1.5. Problemática

Se ha mostrado que la robótica en el ámbito de la rehabilitación motora se lleva desarrollando durante años, buscando adaptarse a la recuperación con diferentes focos (miembros superiores o inferiores) y en las distintas fases post-ictus (aguda, subaguda y crónica). La gamificación, cuyo principal objetivo es aumentar la motivación del paciente, se emplea como elemento de apoyo para potenciar los resultados de mejoría, lo que ha llevado a la elaboración de una creciente variedad de juegos serios. Utilizarlos le da un sentido a la tarea.

Con el objetivo de contribuir aumentando la oferta de juegos para pacientes en fase crónica, este TFG se centra en el desarrollo de Arcade, un juego de tipo mayoritariamente «Libre». Esta categoría, que como se presenta en la sección anterior tiene el objetivo de entrenar el tiempo de reacción de los usuarios y la coordinación, requiere de un nivel mayor de destreza motora. Son los pacientes que ya han experimentado cierto nivel de mejoría y con un determinado grado de autonomía, aquellos que hacen uso de estos. Los necesitan para poder progresar incluso tras las fases aguda y subaguda de la enfermedad, siendo indispensable para su tratamiento.

1.6. Objetivos

Este Trabajo de Fin de Grado surge con el objetivo de desarrollar, para el robot de rehabilitación de miembro superior Rubidium, un nuevo juego serio para trabajar el movimiento horizontal, que involucra principalmente la abducción y aducción del hombro, o el vertical, implicando en su mayoría la flexión y extensión del codo.

A continuación, se exponen los objetivos establecidos en el desarrollo de este TFG:

- **Desarrollar el juego de «Arcade»** para Rubidium, sin asistencia y enfocado a pacientes con movilidad en la fase crónica de las lesiones.
- Se pretende **introducir parámetros que permitan la adaptabilidad de la dificultad a diferentes pacientes**, provocando en cada caso una terapia basada en el movimiento constante y la reacción cognitiva.
- **Se analiza y estudia cómo afectan dichos parámetros a la motivación y el rendimiento de los usuarios con datos objetivos**, como las posiciones y velocidades del efector final del robot durante las sesiones, o la puntuación obtenida en cada caso.

2. Materiales y métodos

En esta sección se presentan los diferentes componentes de hardware y de software que han intervenido en la realización de este TFG.

2.1. Rubidium

2.1.1. Descripción

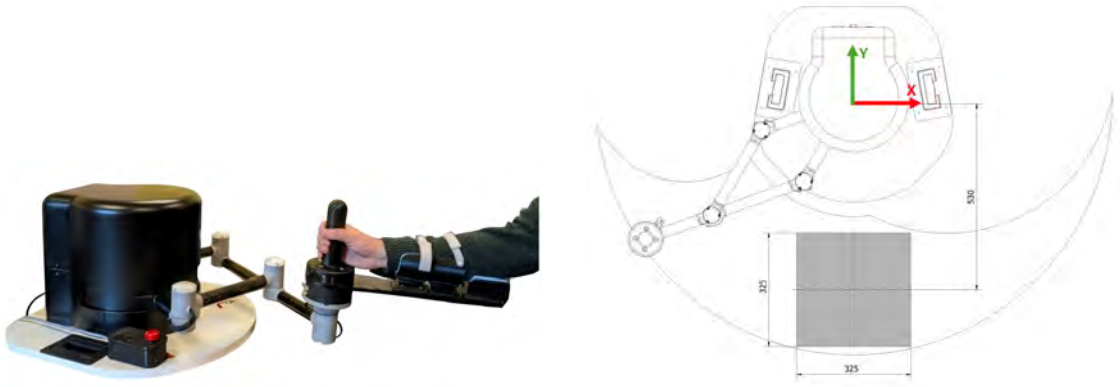
Rubidium es un dispositivo robótico de asistencia personal, diseñado para la terapia de rehabilitación de las extremidades superiores. Atendiendo a la clasificación presentada en la introducción, se trata de un sistema robótico de efector final, muy similar al InMotionArm de Bionik (ver Figura 2.1a).

Es un robot planar horizontal, es decir, que permite el movimiento en las dos dimensiones que forman el plano paralelo al suelo, abarcando así por una parte la flexión y extensión del codo, y por otra la abducción y aducción del hombro.

Está constituido por cuatro barras conectadas formando un paralelogramo articulado. Dos de estos eslabones se encuentran parcialmente en el interior de una carcasa que proporciona los dos finales de carrera físicos que delimitan el movimiento hacia los laterales. En la Figura 2.1b se puede ver el espacio de trabajo del robot.

Sobre cada articulación hay un indicador LED. Estos dan información acerca del estado de conexión en el que se encuentra el robot, así como del modo de control en el que está operando. Además, de la articulación entre las dos barras situadas al aire sale otro eslabón con el efector final en su extremo, elemento de unión entre la máquina y la persona constituido por un soporte para el antebrazo y un mango intercambiable. Esto permite liberar el peso del miembro del usuario y mejorar el agarre, proporcionando una rehabilitación más cómoda y eficaz.

El dispositivo está adherido a una base que dispone de agarres sobre su cara superior para el transporte, de manera que se pueda colocar sobre diferentes superficies planas, a distintas alturas y orientaciones en función de la disponibilidad espacial del lugar de la sesión, y de las necesidades que cada usuario pueda tener por su constitución.



(a) Aspecto real del dispositivo. (b) Espacio de trabajo útil.

Figura 2.1: Sistema robótico Rubidium y representación de su área de operación.

Además, cuenta con un mando de control que se conecta a la parte trasera del robot (ver Figura 2.2). Este mando dispone de una seta de emergencia, un LED de error y los pulsadores de rearme y de homing que permiten la conexión del robot con el PC.

Figura 2.2: Mando de control de Rubidium.

2.1.2. Modos de juego soportados

El robot posee varios niveles progresivos de actuación, desde la completa asistencia donde la persona no necesita realizar ningún movimiento para alcanzar los objetivos, hasta un nivel donde el dispositivo ejerce una resitencia al movimiento del usuario. De esta manera, se pueden realizar los tres tipos distintos de terapia de los que se habla anteriormente: pasiva, activa-asistida y activa-resistiva. Los modos de asistencia son los siguientes:

1. **Modo Libre:** no existe ningún tipo de asistencia ni apoyo para la consecución

de los objetivos, sino que lo único que produce el movimiento de Rubidium es la fuerza que el usuario ejerce sobre él, estando los motores apagados. No responde a ningún tipo de terapia de los mencionados, sino que al dejar total libertad de movimiento, está dirigido a personas con cierto grado de destreza motora.

2. **Modo Asistido (Total):** se indica la posición objetivo en el juego y el robot se desplaza de forma autónoma para alcanzarla, ejercitando el movimiento de la persona. Responde, por tanto, a la terapia pasiva.

3. **Modo túnel con extremos:** mediante dos puntos que definen la recta sobre la que se tiene que mover la persona dentro del juego, este modo controla las fuerzas que ayudan a impedir que se salga de la trayectoria. Se configuran unos extremos en la recta, que las fuerzas no permiten traspasar. Estos puntos pueden ser fijos o constrictivos, es decir que se van haciendo pequeños según se requiera.

4. **Asistido según sea necesario:** contando con las restricciones del modo túnel, además ayuda al desplazamiento del miembro ejerciendo fuerza hacia el objetivo si al cabo de cierto tiempo no hay avances. Se puede considerar una mezcla entre los dos tipos de terapia activa: asistida y resistiva.

El juego desarrollado en este TFG está diseñado para el modo libre, ya que tiene que permitir a la persona equivocarse y fallar dándole al paciente poder de decisión para que este sea más entretenido y tenga un mayor efecto motivacional.

2.2. REVIRE

REVIRE (Realidad Virtual para Rehabilitación) es un software específicamente diseñado para inicializar la conexión con el hardware y ejecutar las actividades que el paciente debe realizar (lanza los juegos). Además se encarga de gestionar toda la información clínica con la que trabajan los terapeutas: el registro de los diferentes pacientes, sus sesiones y las actividades asociadas a dichas sesiones.

La aplicación no permite únicamente seleccionar distintos usuarios, sino también establecer la conexión con diferentes dispositivos. Esencialmente, REVIRE busca la integración con robots y sensores: se comunica con sistemas robóticos de asistencia como Rubidium y dispositivos vestibles (wearables) como las gafas de realidad virtual Meta Quest 3, que guían y monitorizan los movimientos del usuario.

2.2.1. Interfaz del programa

Puesto que como se comenta, es un software diseñado para uso de los terapeutas, la interfaz debe ser clara, intuitiva y funcional. El flujo de navegación se muestra en la Figura 2.3 y se comenta a continuación.

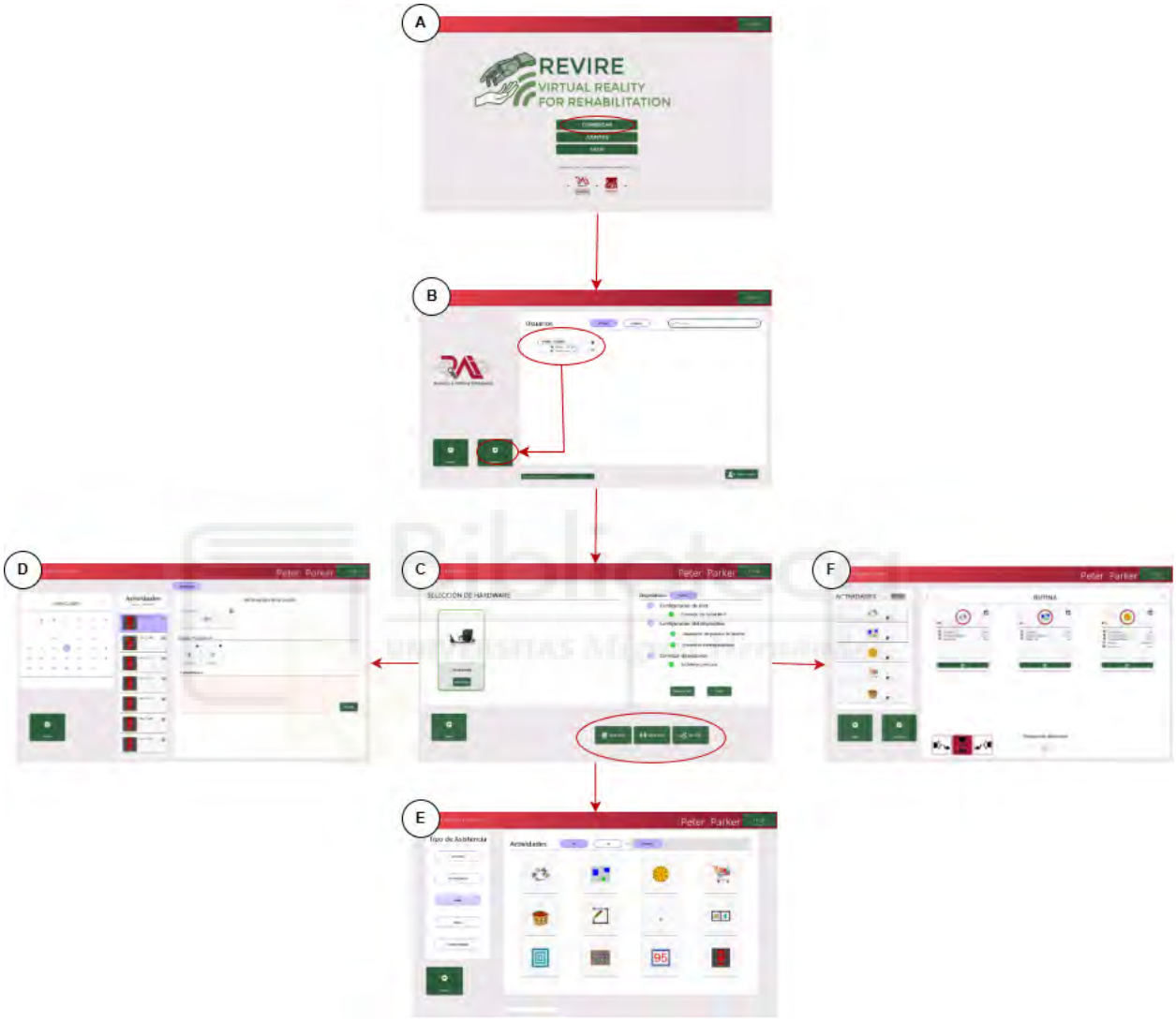


Figura 2.3: Flujo de navegación de REVIRE entre las diferentes pantallas.

La selección de usuarios y dispositivos, son los dos primeros pasos dentro del programa, como se ve en las Figuras 2.3A y 2.3B.

Una vez establecida la conexión con el dispositivo, se permite visualizar todos los juegos disponibles y lanzar actividades de forma aislada desde la pantalla E de la Figura 2.3 (seleccionando el botón de «Ejercicio») u organizar sesiones compuestas por varios juegos desde la pantalla F (seleccionando el botón de «Rutina»). Al preparar los juegos, el terapeuta puede modificar de forma intuitiva los parámetros

involucrados en cada actividad, así como el tiempo de descanso entre juegos en el caso de las sesiones.

Además, la aplicación cuenta con un registro de los entrenamientos en forma de calendario (botón de «Sesiones»), como se ve en la pantalla D del diagrama. Ahí se registran las actividades pasadas de cada usuario y se pueden consultar con facilidad.

2.2.2. Obtención y gestión de datos

Es esencial recoger información acerca de la actuación de los pacientes. Los terapeutas se basan en escalas clínicas subjetivas para evaluarlos, lo que inevitablemente implica sesgos en la valoración. El objetivo es evitar esto y disponer de evaluaciones objetivas y medibles mediante el almacenamiento y análisis de multitud de datos, que se registran durante las sesiones.

REVIRE cuenta con una base de datos SQLite, ligera y relacional, a la que se copian los datos de cada sesión al finalizar, almacenando y organizando la información sobre los entrenamientos de todos los pacientes. Algunos ejemplos de datos relevantes son:

- De tipo personal y clínico: como el nombre, la edad, el género o el historial de actividades.
- Acerca de los juegos y su configuración: ya sea el modo de asistencia empleado, el tiempo límite para la tarea a completar, la distancia objetivo a desplazar en función de las capacidades del paciente, el número de repeticiones u otros parámetros. Se añaden a la base de datos también la puntuación final y el tiempo que ha durado la partida.
- Métricas de rendimiento físico: como fuerzas, trayectorias, velocidades y aceleraciones. Dan información acerca de los movimientos del brazo a través de los movimientos del robot. Cabe destacar que estas métricas no se guardan en la base de datos de REVIRE debido al peso de la información, sino que se almacenan localmente en la carpeta de datos del proyecto de Unity, como se muestra más adelante.

2.3. Unity como software de creación de videojuegos

Para comenzar la parte del TFG entorno al juego creado para Rubidium, es conveniente hablar de Unity, muy utilizado por la comunidad de desarrolladores para

proyectos grandes.

Es un motor de videojuegos con su propia interfaz de desarrollo y una amplia oferta de herramientas que permiten ahorrar una enorme cantidad de tiempo. Un ejemplo son las leyes físicas, como puede ser la gravedad, las cuales no es necesario programar desde cero. Además, cuenta con herramientas de renderizado, gestión de sonido y detección de colisiones. También cuenta con otras específicas de RV muy útiles para el ámbito de la salud, ya que permiten identificar, por ejemplo, donde están las manos del paciente o hacia dónde está mirando.

Características como las mencionadas, sumadas a la facilidad de exportación del juego para que funcione en diferentes soportes, como un móvil, un ordenador, o unas gafas de realidad virtual Meta Quest, hacen de Unity una buena opción para la creación de videojuegos de rehabilitación. Ese es uno de los motivos por los que todos los juegos de REVIRE han sido elaborados con este programa.

En la Figura 2.4 se muestra la disposición de la interfaz y sus diferentes secciones principales.

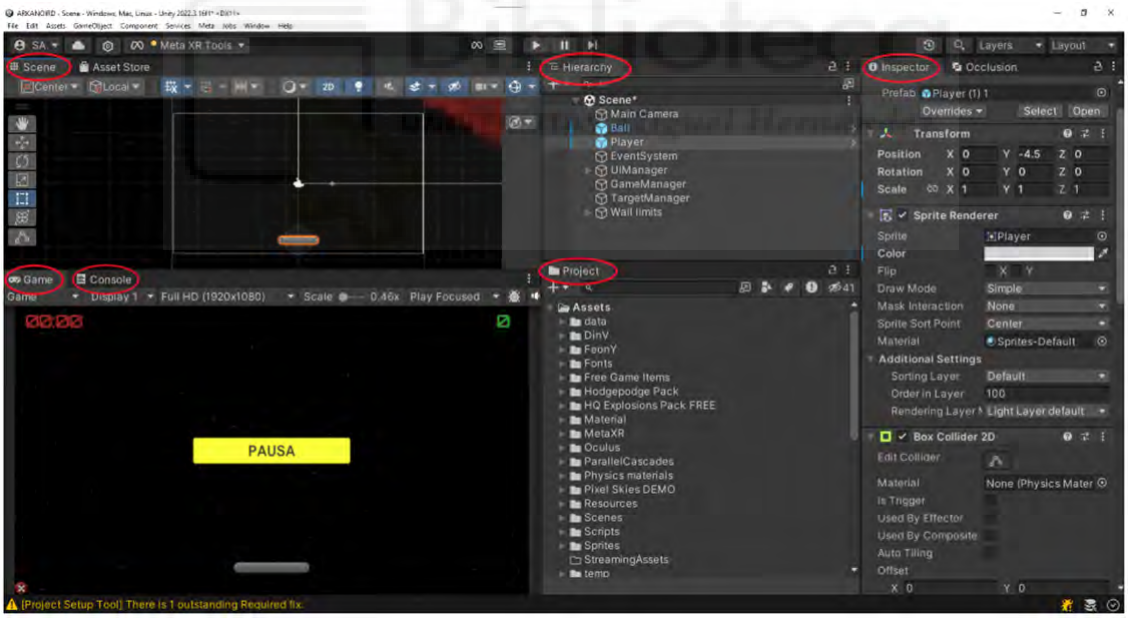


Figura 2.4: Interfaz de Unity donde se visualizan las principales secciones.

Secciones de la interfaz:

- Scene: es el espacio de creación de la escena. Ahí se conforma la escena con todos sus objetos y se crea la cámara que delimita el encuadre visual del juego.
- Hierarchy: en la jerarquía se encuentran todos los objetos presentes en las

escena en ese mimso instante. Se pueden agrupar y desglosar.

- Inspector: esta sección permite visualizar, asignar, eliminar y modificar todos los atributos y métodos del objeto seleccionado en la jerarquía.
- Game: como su nombre indica es la ventana en la que se juega. Simula la visualización real del ejecutable.
- Console: permite visualizar los errores, las advertencias y mensajes del código.
- Project: la carpeta del proyecto agrupa todos los recursos de este: sprites, scripts, datos, etc.

2.4. Intercomunicación de elementos

En este apartado se representa una visión general de cómo Rubidium, REVIRE y Unity se relacionan entre sí, como se muestra en la Figura 2.5.

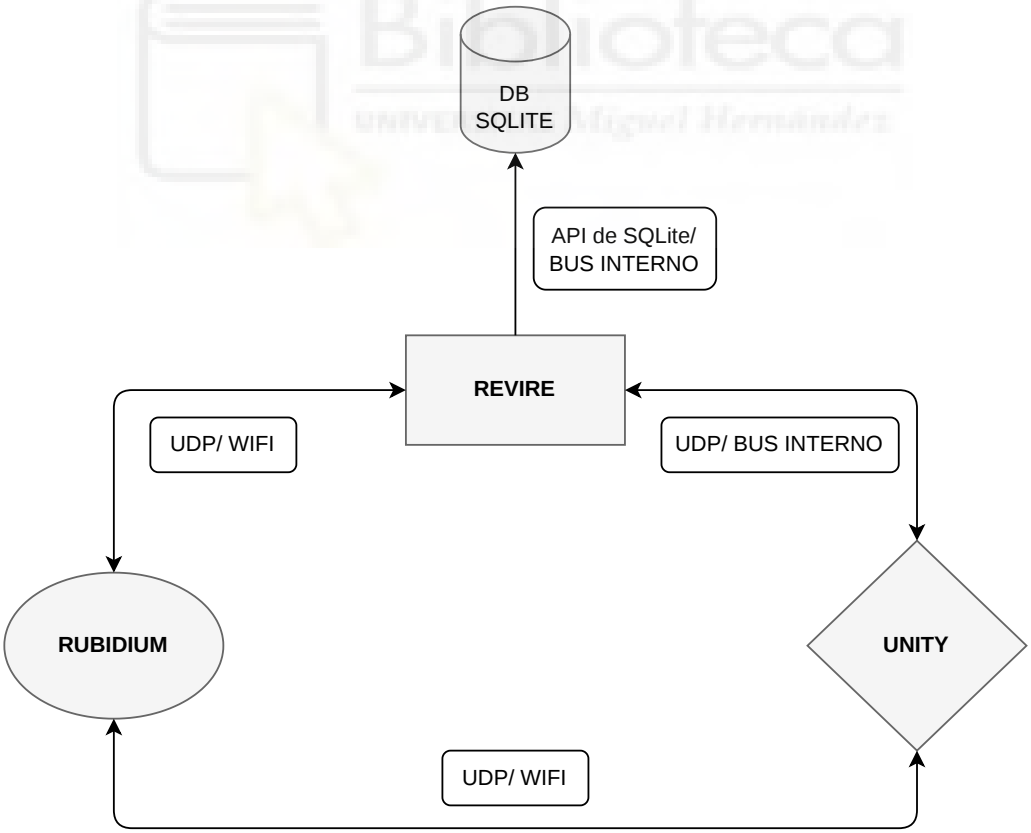


Figura 2.5: Esquema de intercomunicaciones del sistema. Comunicaciones bidireccionales entre Unity, Rubidium y REVIRE, y unidireccional con la base de datos.

Como se mencionaba anteriormente, es necesario establecer la comunicación con Rubidium y configurarlo a través del gestor mediante un proceso. Una vez encendido el robot, este crea una red WIFI local a la que se debe conectar el PC. Tras realizar el rearme y homing siguiendo las indicaciones de REVIRE, el robot está conectado y listo para ser usado. Para saber cuando desconectar Rubidium, abre un hilo de recepción UDP que queda en espera de recibir la señal del robot que se lo indique.

Desde REVIRE se seleccionan los parámetros del juego y se ejecuta. El lanzamiento del juego se lleva a cabo mediante un proceso que a su vez abre la comunicación con Unity y otro hilo de recepción en segundo plano para el momento en que la partida termine y reciba los datos enviados del juego para guardar en la base de datos.

A su vez, entre Unity y Rubidium existe una comunicación, constantemente activa en ambos sentidos debido a la necesidad de traducir el movimiento del efector final dentro del juego, para detectar si se está siguiendo la trayectoria adecuada o si es necesario suministrar fuerzas en caso de utilizarse algún modo de asistencia.

Es conveniente destacar el motivo de que las comunicaciones que se establecen tengan protocolos UDP. De esta forma, se garantiza una respuesta rápida del juego a los movimientos del hardware y viceversa (en función del modo de asistencia) gracias a que el protocolo envía la información lo más rápido posible sin esperar confirmación. Esto es esencial para asegurar la sincronización entre los dispositivos robóticos y el juego. En este ámbito, importa la velocidad a la que se envían los datos. En caso de perderse alguno, no es esencial recuperarlo, sino detectar el siguiente lo antes posible para subsanar el retraso de la manera más inmediata posible.

2.5. Desarrollo de Arcade

Se iniciará este apartado con una breve descripción del juego objetivo del desarrollo. Este, es de tipo Arkanoid (ver Figura 2.6). Originalmente, se basa en una barra que corresponde al jugador y se desplaza lateralmente. De esta forma, se dirige mediante el rebote una pelota que se mueve en el plano con el objetivo de destruir ladrillos dispersos en el escenario. La bola cambia su trayectoria al colisionar tanto con los ladrillos como con tres de las cuatro paredes de la escena. En caso de contacto con la pared inferior, sobre la que se mueve el jugador, se pierde una vida. Al finalizar todas las vidas, el juego termina.

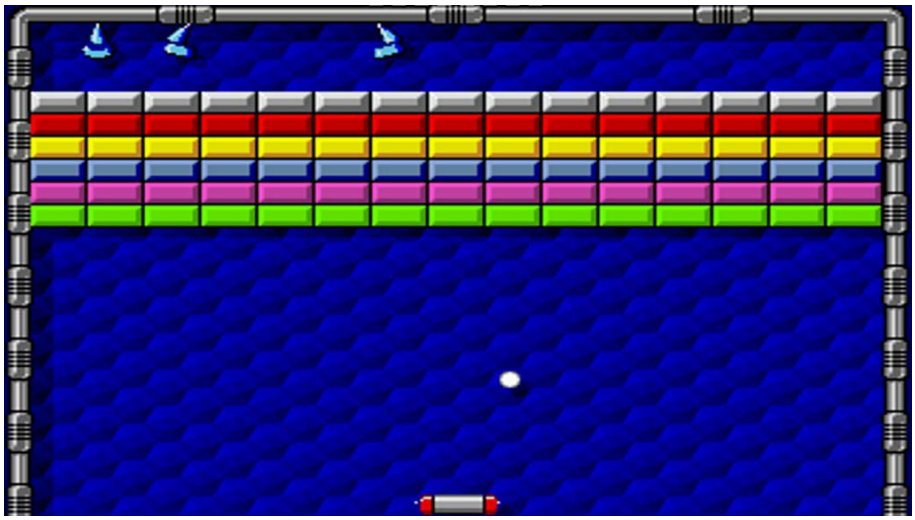


Figura 2.6: Juego tipo Arkanoid.

Se ha trabajado sobre este concepto, suprimiendo y añadiendo diversas funcionalidades y elementos extra que lo adaptan a su finalidad terapéutica. En todo momento, se ha tenido en cuenta la necesidad de simplicidad y claridad que se argumentaba anteriormente de las características gráficas del juego, como el contraste de colores entre el fondo de pantalla y los diferentes elementos, o el tamaño y disposición de todos los objetos y componentes del canvas para facilitar así la comprensión, teniendo en mente al público al que va dirigido.

2.5.1. Elementos del juego. Motivación.

Como elementos principales y básicos que componen el juego se encuentran: la figura del jugador (que constituye una barra alargada), la pelota, y los ladrillos/bricks, que forman los distintos escenarios presentes en diferente cantidad y disposición en la pantalla. El objetivo del juego consiste en destruir todos los ladrillos con la pelota, dirigiéndola a través del rebote con la figura del jugador. Este solo puede seguir una trayectoria lineal dentro del juego, sujeta a la recta sobre la que se encuentra, a pesar de no ser estrictamente recto el movimiento del robot. El rebote contra el centro de la barra es perpendicular, y el ángulo varía hacia los laterales a medida que la pelota se acerca a los extremos de la barra. Destruir un ladrillo da un punto, y que la pelota caiga en la zona bajo la línea sobre la que se desplaza el jugador (DeadZone), lo quita.

Adicionalmente al juego original están los elementos de las rocas y los diamantes. Las rocas caen desde fuera de la escena hacia la línea de desplazamiento del jugador y son perjudiciales: quitan 5 puntos en caso de contacto. Por otra parte, los

diamantes aparecen de forma aleatoria al destruir algunos bricks y el objetivo es alcanzarlos con el jugador cuando caen, obteniendo así 5 puntos. Ambos elementos han sido introducidos para favorecer la necesidad de movimiento continuo del jugador y suprimir los tiempos largos de espera en los que la pelota se queda rebotando entre varios ladrillos o las paredes.

En la parte superior derecha e izquierda se encuentran, respectivamente, el contador de la puntuación y el contador descendiente del tiempo restante, que como se comenta más adelante es lo que marca el paso de un nivel a otro. Todos los elementos se pueden ver en la Figura 2.7, que representa la escena del juego durante una partida.

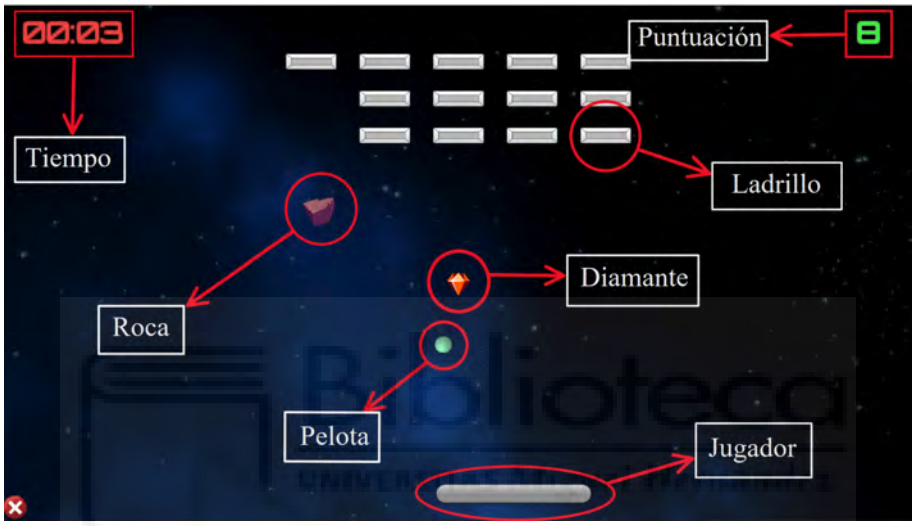


Figura 2.7: Escena del juego desarrollado donde se indentifican cada uno de los elementos de los que está compuesto.

A continuación, se describen las diferentes consideraciones que se han tenido durante el desarrollo del juego con el objetivo de explotar su potencial motivacional.

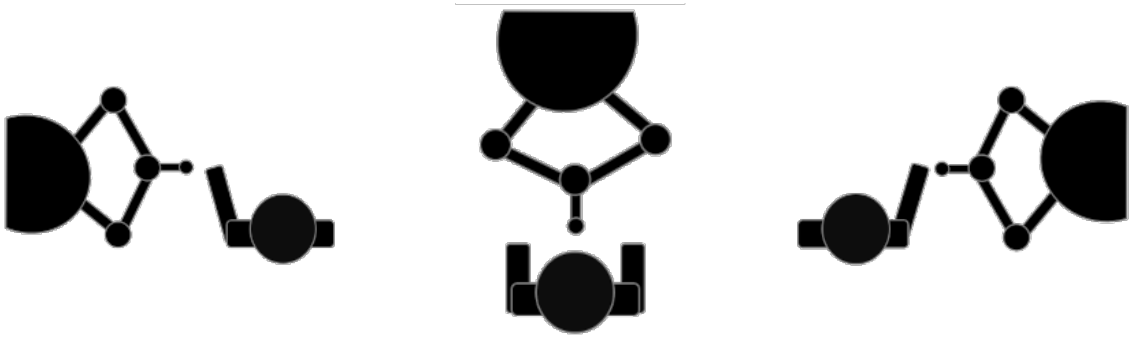
- **Supresión de las vidas:** puesto que el objetivo es ejercitar el movimiento de forma constante y repetida durante un tiempo concreto y no de acuerdo a unos intentos limitados, se elimina el elemento de las vidas. El juego finalizará, por tanto, pasado un número determinado de escenarios y un tiempo límite establecido por el terapeuta.
- **Formato del contador del tiempo:** un contador decremental, fácilmente visible, que se reinicia para cada escenario fomenta una mayor implicación por parte del paciente por aversión a la pérdida. Por otra parte, garantiza el paso del escenario a uno diferente en caso de dificultad excesiva para completarlo, evitando que la frustración se convierta en una limitación en el proceso de rehabilitación.

- **Sistema de puntuación:** es otro elemento clave en la implicación del paciente. Habiendo comentado las acciones que suman y restan puntos, se puede entender como el dinamismo en el marcador motiva al usuario. Asimismo, el límite inferior está en 0, de forma que no se pueda obtener puntuación negativa y eso tenga un efecto desmoralizante.
- **Variabilidad de escenarios:** no existe una distinción clara entre la dificultad que implica la diferente disposición de bricks del banco de escenarios creados. El objetivo de la variedad de escenas no es proporcionar niveles de dificultad. No obstante, es necesario para evitar la monotonía y estimular la participación del usuario. Los escenarios aparecen de manera aleatoria, pero sin poder repetirse en una misma sesión.

2.5.2. Elementos de Adaptabilidad

Arcade cuenta con una serie de parámetros presentes en todos los juegos desarrollados para Rubidium. Por otro lado, incorpora otros parámetros específicos. Como ya se ha mencionado, el hecho de intoducir parámetros está ligado a la capacidad de adaptabilidad. A continuación, se listan y explican todos aquellos inputs relevantes en el juego y de qué manera son importantes para ajustarse al paciente:

- **Layout:** como su nombre indica, hace referencia a la disposición del Rubidium. Se utiliza para saber cómo está colocado físicamente el robot respecto de la pantalla donde se visualizan los juegos. Las diferencias en la colocación pueden ser requeridas según las distintas necesidades espaciales del sitio en que se lleva a cabo la terapia. Hay tres distribuciones posibles, representadas en la Figura 2.8.



(a) Layout = 1 (b) Layout = 0 (c) Layout = 2

Figura 2.8: Disposiciones posibles del robot frente a la pantalla según el parámetro «Layout».

- **Amplitud:** se refiere a la distancia física en milímetros que abarca el dispositivo, es decir, el rango que cubre el robot correspondiente a los límites de la pantalla. Este campo de trabajo se varía en función de cuánto se desea que mueva el brazo el paciente.

  La amplitud es necesaria para poder realizar la conversión de coordenadas reales a coordenadas del mundo de Unity, que cuenta con unidades propias. Para ello se realiza la interpolación lineal, manteniendo una relación proporcional constante.

  En primer lugar, es necesario hallar el factor de escala que relaciona el rango de la escena con el rango del dispositivo:

  $$m = \frac{\text{Rango escena}}{\text{Amplitud}} \tag{2.1}$$

  De esta manera podemos traducir el desplazamiento del robot a unidades de Unity:

  $$\text{PosUnity} = m \cdot \text{PosRobot} + b \tag{2.2}$$

  Donde b es el desplazamiento que se debe tener en cuenta entre los centros de cada sistema de coordenadas para los ejes x e y, es decir la diferencia entre el punto destino en el que debería aparecer el robot según su posición física, y el punto origen en el que aparece en Unity.

  $$\text{b} = \text{Punto Destino} - \text{Punto Origen} \tag{2.3}$$

  Estas fórmulas se aplican para el cálculo de las coordenadas en x e y, variando también en función del valor de Layout, ya que afecta a la manera en la que el movimiento se traduce dentro del juego. Por ejemplo, si la persona quiere desplazarse hacia la derecha en la pantalla y el robot está frente a esta (ver Figura 2.8b), los ejes corresponden y mueve el brazo hacia la derecha. En cambio, si el robot se encuentra a la derecha de la pantalla (ver Figura 2.8c), moverá el brazo hacia arriba en el eje y del robot para conseguir el mismo desplazamiento.

- **NivelAsistencia:** mediante este parámetro se seleccionan los modos de asistencia del robot, siendo el valor para el Modo Libre: 1.0f.

- **TimeTotal:** es el tiempo límite que se establece para cumplir cada escenario.

Es único e igual para todas las escenas de la sesión.

- **Lateralidad:** es un indicador para determinar cuál es el brazo con que se ejecutan los ejercicios y así poder distinguir y clasificar los datos de ambos miembros. El valor 1 corresponde al derecho y el 2 al izquierdo.

- **Repeticiones:** mediante este input se escoge el número de escenarios/niveles que formarán parte de la sesión. Únicamente se escoge la cantidad, puesto que estos se dan de manera aleatoria.

- **RockDensity:** como elemento complementario para estimular la continuidad del movimiento, se han añadido las rocas, que caen hacia el jugador y restan puntos en caso de contacto con él, es decir, se deben evitar. Este parámetro permite modificar la cantidad de rocas que caen durante las partidas y variar así su dificultad. Hay 4 opciones estándar con distinta densidad de rocas:

  0 = Sin Rocas.

  1 = Densidad Baja.

  2 = Densidad Media.

  3 = Densidad Alta.

- **InitialSpeed:** es la velocidad de la pelota elegida para la sesión. Cuando el paciente pase a la siguiente pantalla antes de que acabe el tiempo establecido, habiendo roto todos los bricks, la velocidad aumenta ligeramente dentro de un rango, y en caso contrario, baja en la misma magnitud. Es decir, cada vez que hay un cambio de escenario, la velocidad de la pelota sufre un cambio de $\pm 0{,}5f$ hasta una variación de $\pm 1{,}0f$ sobre la velocidad inicial. De esta manera, el juego se adapta ligeramente a la actuación del paciente con el objetivo de facilitar que alcance los objetivos y a su vez pruebe sus límites.

- **ItemsSpeed:** es la velocidad de los objetos que caen (diamantes y rocas).

- **Movement:** en el juego original, la plataforma que corresponde al jugador se desplaza horizontalmente por el límite inferior entre los laterales derecho e izquierdo. Esto promueve principalmente el movimiento de apertura y cierre del hombro.

Para trabajar también la flexión y extensión del codo, se crean los modos de juego laterales, en los que todos los objetos del juego se rotan 90º, de manera que el jugador se desplaza en vertical en la pantalla, entre los límites superior e inferior.

Este parámetro, se utiliza, por tanto, para acompañar visualmente el movimiento del jugador en función de los diferentes objetivos. Se elige la orientación y disposición del elemento del player, los bricks y la pelota dentro del juego. Las posibilidades se muestran en las imágenes a continuación y son las siguientes:

0 = Horizontal. (ver Figura 2.9)

1 = Vertical Derecha. (ver Figura 2.10)

2 = Vertical Izquierda. (ver Figura 2.11)

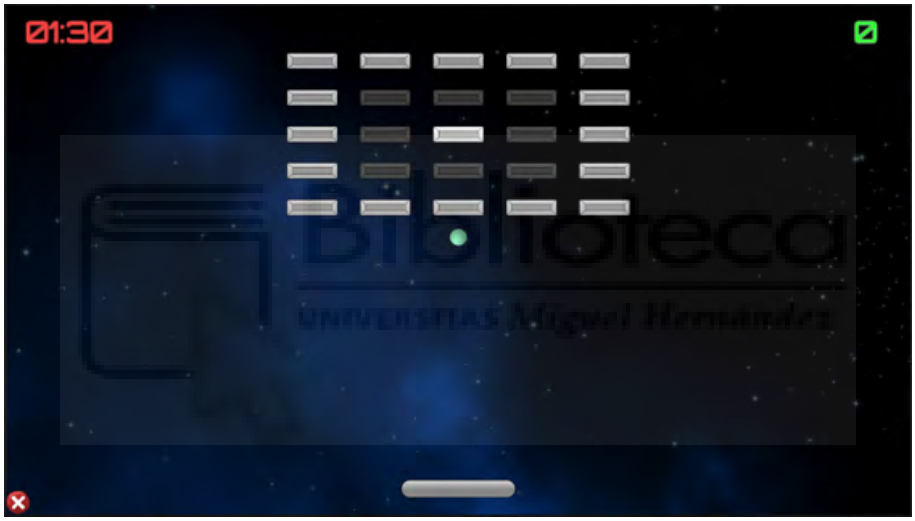


Figura 2.9: Disposición de la escena cuando Movement = 0.

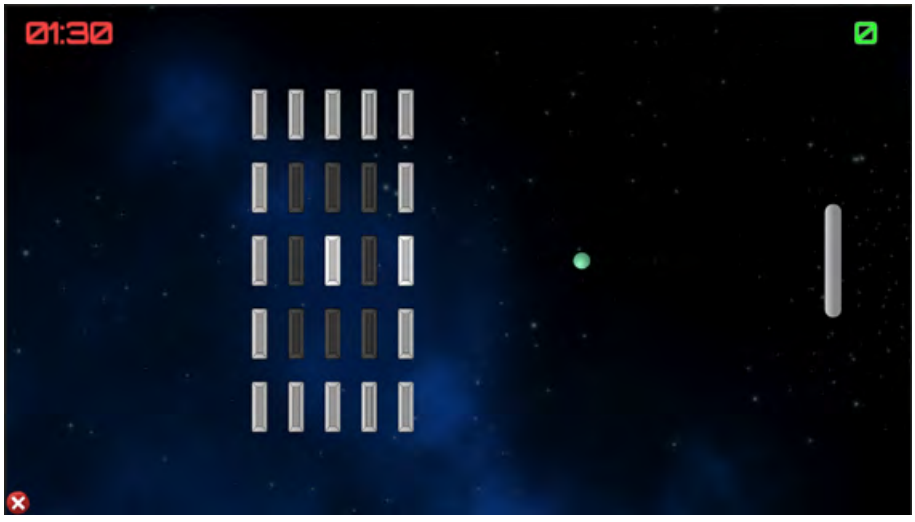


Figura 2.10: Disposición de la escena cuando Movement = 1.

Figura 2.11: Disposición de la escena cuando Movement = 2.

- **PlayerSize:** Para tratar de añadir otro elemento que modifique la dificultad del juego, se han creado dos tamaños estándar del jugador. Debido a la diferencia del rango que hay que cubrir en función del parámetro «Movement», las dos opciones de tamaño se ajustan de manera proporcional, cubriendo el mismo rango relativo en modo horizontal y vertical. De esta forma, aunque la distancia en vertical sea más pequeña, se fuerza a la persona a hacer más recorrido para cubrirla. La diferencia de tamaño entre las dos opciones, y del rango que cubren en horizontal y en vertical se puede apreciar en la Figura 2.12.

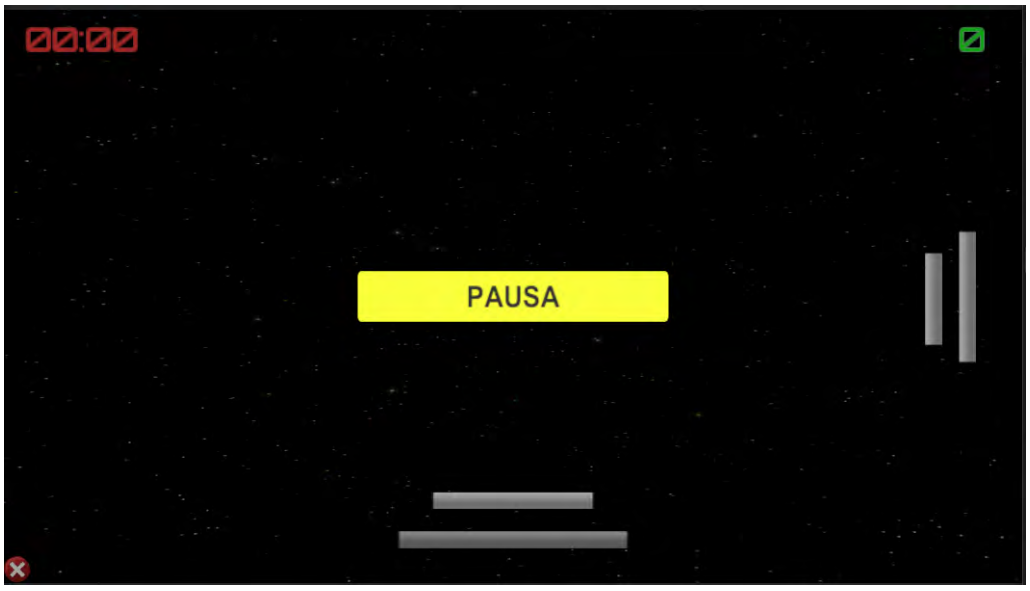


Figura 2.12: Escena del juego donde se aprecian los tamaños pequeño y grande del jugador para el modo horizontal y vertical, y el rango que cubren.

2.5.3. Módulos del software

La clave de la arquitectura del software es la modularidad. Así, se permite conseguir mayor claridad y mejor organización, necesaria por el nivel de complejidad del proyecto, que debe integrar las diferentes funcionalidades de los elementos del juego, el almacenamiento local de los datos y la comunicación con Rubidium y con REVIRE. Mediante la separación de tareas en diferentes bloques, se consigue que el código sea más fácilmente adaptable a cualquier cambio que se pueda presentar en el futuro, como por ejemplo podría ser utilizarlo con otro robot, en cuyo caso solo sería necesario desarrollar un nuevo bloque encargado de esa parte que sustituyera el de Rubidium.

En la Figura 2.13 se observa la estructura modular general de los diferentes bloques y como se relacionan las principales clases, sin representar aquellas referentes a los objetos del juego, que modelan su comportamiento y como interactúan entre sí, como el jugador, la pelota o los bricks, entre otros. En la Tabla 2.1, se muestran las clases principales y sus funciones más importantes.

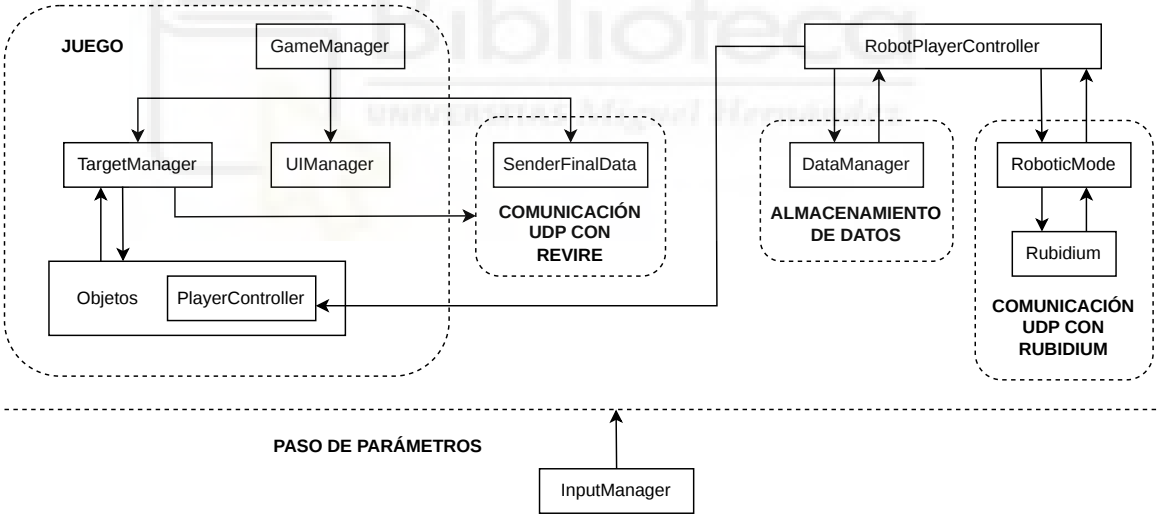


Figura 2.13: Diagrama de módulos general de los diferentes scripts principales del juego. InputManager contiene los valores de los parámetros que utilizan las diferentes clases de todos los módulos. GameManager llama a SenderFinalData cuando la partida acaba antes del lo previsto para que los datos no se guarden en REVIRE, y coordina las clases de UIManager y TargetManager, que a su vez llama a SenderFinalData para mandar el tiempo y la puntuación a REVIRE al final de la partida, y gestiona los diferentes objetos del juego y sus clases. RobotPlayerController establece la comunicación UDP con Rubidium a través de las clases RoboticMode y Rubidium, y envía datos a DataManager para almacenarlos en la carpeta del juego.

Tabla 2.1: Clases principales del sistema y sus funciones.

Clase	**Funcionalidades**
GameManager	Coordina el juego mediante la gestión de diferentes estados, desde que se inicia hasta que termina, contemplando pausas y el modo de seguridad por errores. También lleva el conteo del tiempo total de la sesión.
TargetManager	Gestiona las tareas específicas dentro de la partida mediante diferentes subestados, como por ejemplo el paso entre niveles o la puntuación.
UIManager	Se encarga de la visualización de los elementos de la interfaz, como el contador del tiempo, de la puntuación y mensajes temporales al empezar la partida o entre niveles.
InputManager	Declara y asigna los parámetros de entrada.
DataManager	Incluye todo lo relativo al almacenamiento local de datos desde Unity: la apertura de carpetas y archivos, su escritura y su cierre.
SenderFinalData	Se encarga de iniciar la comunicación con REVIRE para la transferencia de datos al final de la partida.
Rubidium	Es la capa de comunicación con el hardware. Gestiona toda la comunicación UDP con el robot: abre los sockets de envío y recepción, lanza un hilo de recepción en segundo plano para no bloquear el hilo principal de Unity, parsea la información que llega de él y le envía comandos al robot (el modo de control en que debe operar, comunicación de errores, y si fuera necesario fuerzas o posiciones).
RoboticMode	Es la clase base del modo de control. Define las implementaciones comunes de cualquier modo: inicializarlo, actualizar la posición, gestionar errores y la lógica de recuperación. Abstrae el hardware para que el resto del juego no dependa de él.
FreeMode	Es una subclase de RoboticMode que implementa el modo concreto en el que el robot no aplica ninguna fuerza (libre). Inicializa el modo de control del Rubidium y registra la posición actual.
RobotPlayerController	Se ejecuta en cada frame y llama a los scripts necesarios para actualizar la posición del jugador en la pantalla, ya que contiene la lógica de conversión de coordenadas del robot a coordenadas de la escena, actuando así como puente entre el robot físico y el jugador en Unity. Guarda datos.

2.5.4. Proceso de inicialización

En cuanto al proceso de inicialización, es aquel que sucede desde que se le da al botón de «Play» hasta que la escena está lista y el juego comienza. La relación entre scripts y su secuencia de llamadas durante el proceso se ven en el diagrama de la Figura 2.15.

Todo comienza en el método Awake de la clase RobotPlayerController, la central del sistema, que mediante llamadas a diferentes métodos del propio script y de otros, provoca la asignación de los parámetros necesarios para el juego, que se establecen con valores por defecto o mediante asignación por línea de comandos, en función de entorno desde el que se ejecute (ver Figura 2.14). También establece el modo de control del robot, calibra su punto de partida y abre el canal de comunicación UDP por WIFI con el Rubidium para envío y recepción de datos, para lo cual crea un hilo de recepción. Además, prepara las carpetas y archivos para el almacenamiento local de datos en el ordenador.

Una vez realizado todo esto, en el método Start de las diferentes clases se establece el estado y subestado de inicio, la interfaz propia y la inicialización correspondiente a todos los objetos del juego.

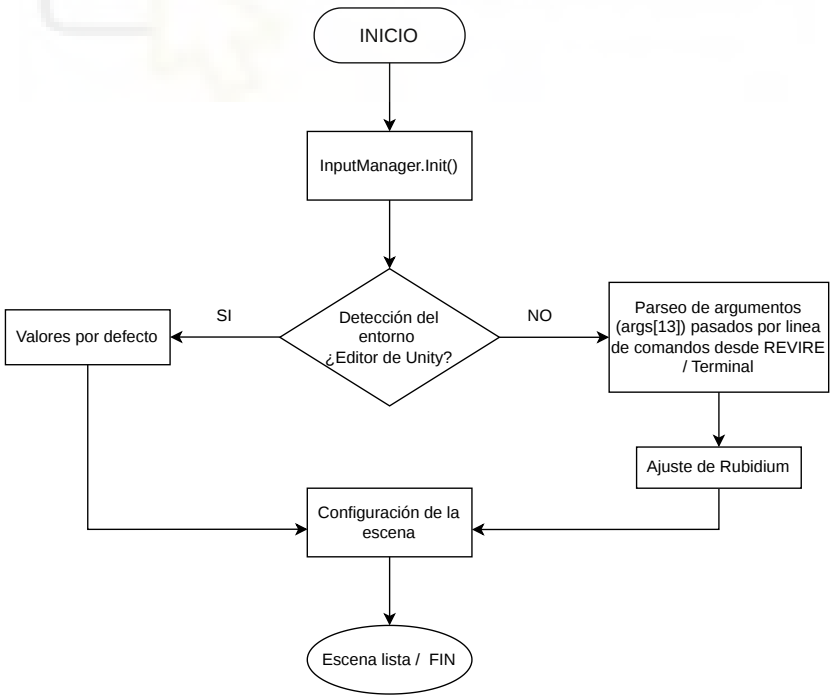


Figura 2.14: Diagrama de flujo de la asignación de argumentos en función del entorno en el que se ejecuta en juego.

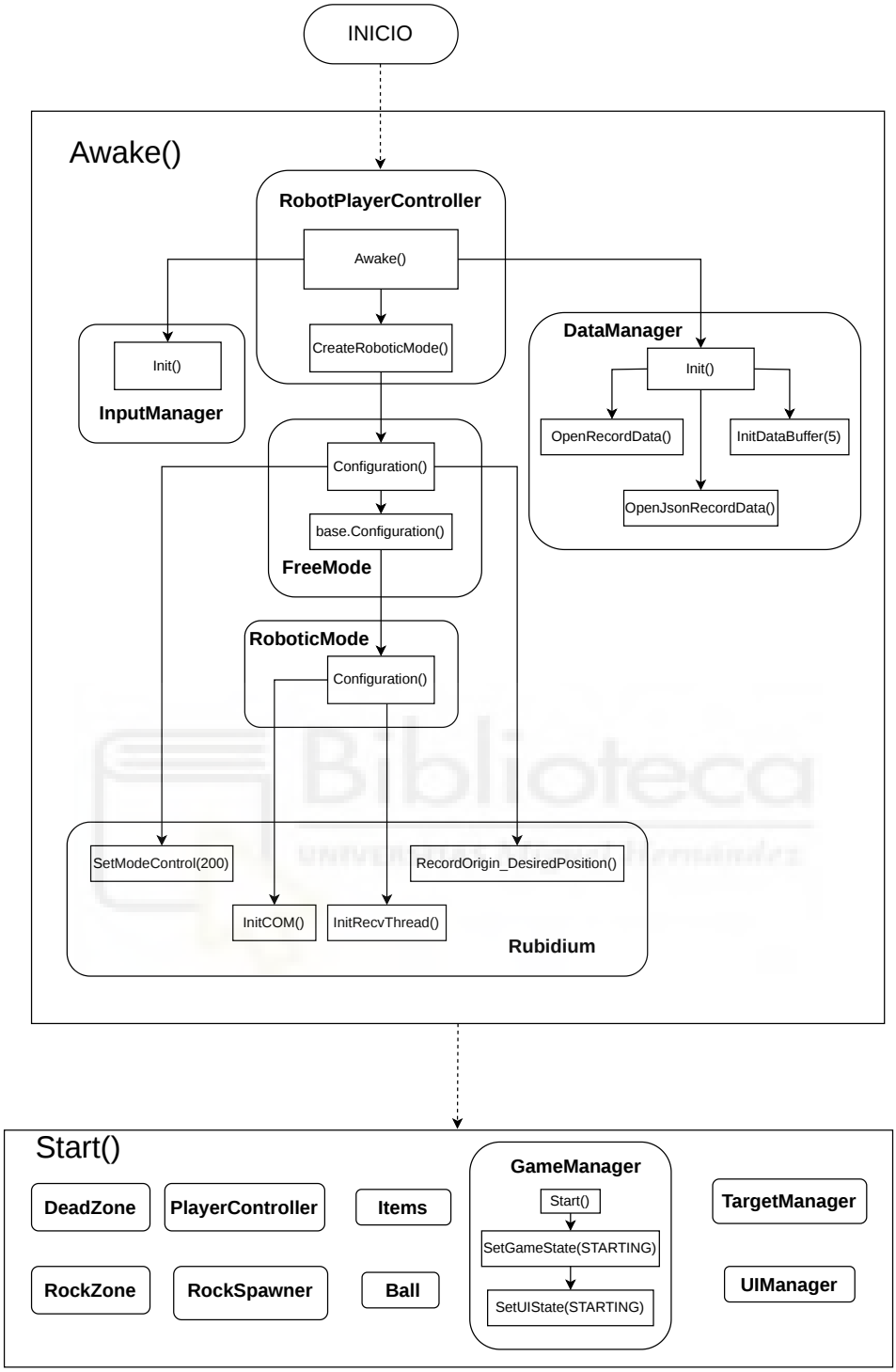


Figura 2.15: Diagrama de módulos y llamadas de inicialización. Todo comienza en el método Awake de RobotPlayerController que en primer lugar asigna los parámetros mediante una llamada a InputManager. Después llama a la clase FreeMode y esta a su vez a RoboticMode, para crear un modo de control para el robot, libre en este caso, a partir del cual se configura mediante diferentes métodos pertenecientes a la clase Rubidium. Por último RobotPlayerController llama a DataManager para abrir las carpetas y archivos .bin y .json destinados a almacenar la información de la partida. Posteriormente se ejecuta el método Start de las diferentes clases.

2.5.5. Estados del juego

El flujo general del juego, tiene una estructura que lo coordina y gestiona mediante dos niveles de estados y subestados. La implementación se basa en una Máquina de Estados Finitos (FSM), es decir, que cuenta con un número concreto de fases, pudiendo encontrarse solo en una a la vez y transicionando entre ellas a través de eventos concretos que producen el cambio. Este patrón de diseño permite que el sistema se comporte de manera diferente según la situación actual de la sesión de una forma más organizada, separando las diferentes tareas y responsabilidades.

Existen dos niveles en la coordinación del juego. Por una parte, el GameManager se encarga de los estados globales del juego independientemente de las mecánicas de este. Simplemente gestiona momentos de preparación, que se lleve a cabo la ejecución de la partida, tiempos de pausa o de seguridad, y cierre. Cuenta por tanto, con los siguientes estados del juego que hacen referencia a las mencionadas tareas respectivamente: STARTING, INGAME, PAUSE, SAFETYMODE y FINISH.

Por otro lado, el TargetManager solo actúa cuando el juego está activo durante el estado global INGAME, que lo descompone en subestados que modulan la lógica interna de la partida y sus fases, tales como la carga de escenarios, la lógica de victoria/derrota, etc. Los subestados son: INIT, PLAYING, SUCCESS, FAIL y WAITFORREMOVE.

- INIT: el único que se establece antes de INGAME. Solo tiene una función de clasificación pero es prescindible.
- PLAYING: envuelve toda la lógica de la partida desde que esta empieza hasta que se termina el nivel. (ver Figura 2.16)
- SUCCESS: cuando termina un nivel de forma éxitosa (eliminando los bricks antes de que el tiempo termine), provoca la aparición del mensaje "Bien hecho" por pantalla y el aumento de la velocidad de la pelota en 0.5 unidades de cara al siguiente nivel.
- FAIL: de la misma manera que SUCCESS, gestiona el mensaje "Probemos algo diferente" y la disminución de la velocidad de la pelota cuando el nivel no se supera dentro del tiempo establecido.
- WAITFORREMOVE: es el subestado siguiente a SUCCESS o FAIL que deja un segundo para la lectura de los mensajes antes de cargar el siguiente nivel y volver de nuevo al subestado PLAYING.

El diagrama de flujo que muestra las transiciones entre estos estados y subestados se muestra en la Figura 2.17.

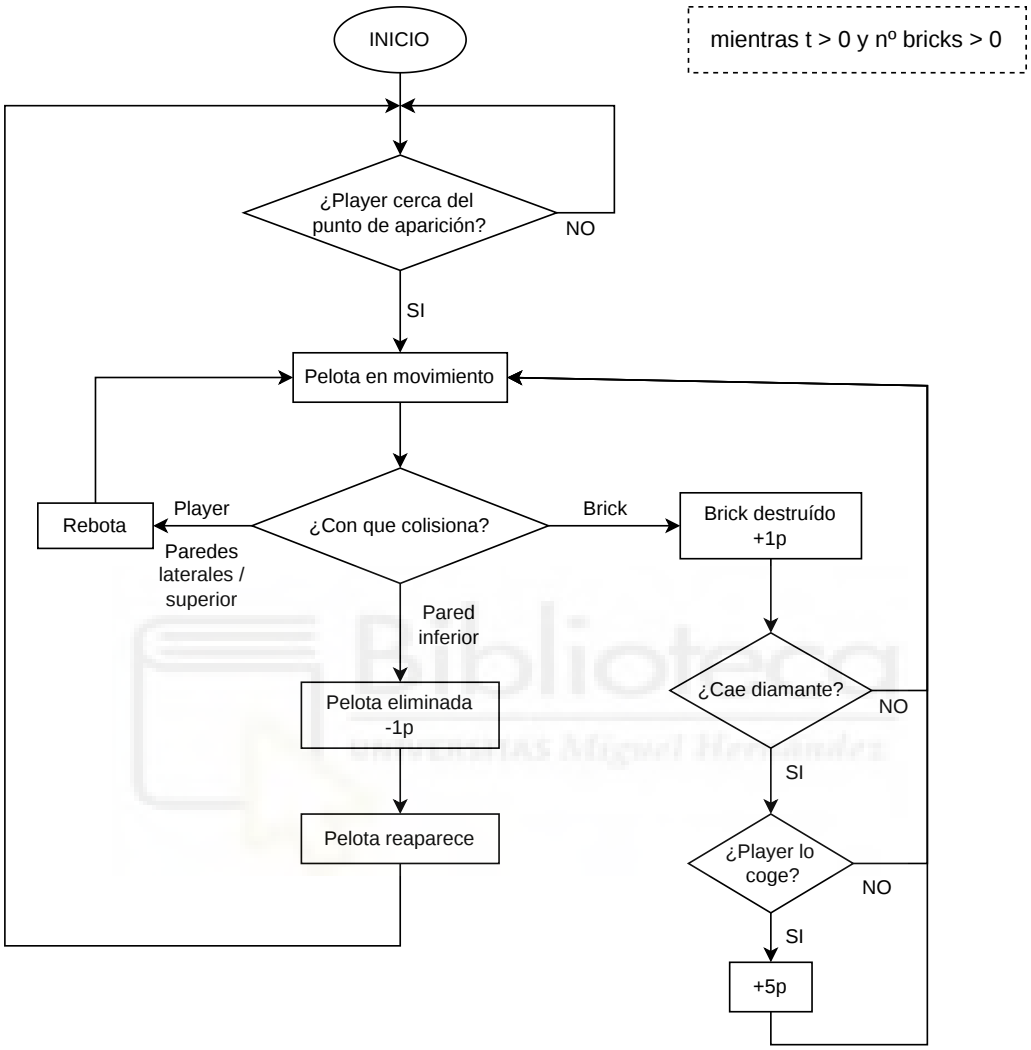


Figura 2.16: Diagrama de flujo de la partida durante el subestado PLAYING. Además, cuando el jugador toca las rocas se restan 5 puntos del marcador, valiendo como mínimo 0.

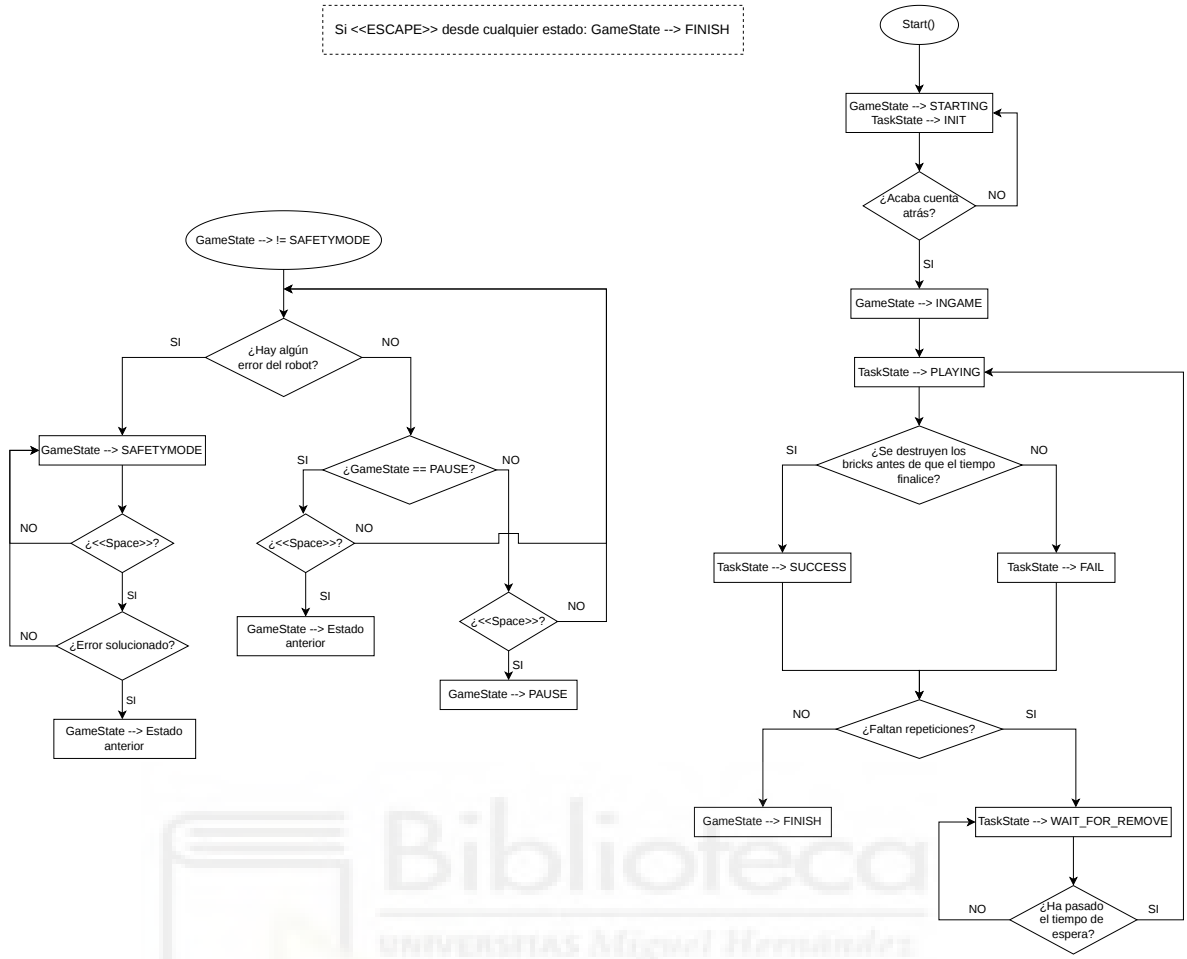


Figura 2.17: Diagrama de flujo de los estados y subestados del juego y de sus transiciones.

2.5.6. Almacenamiento local de datos de Unity

Respecto al almacenamiento local de datos llevado a cabo desde Unity, necesario para el análisis de los resultados, se compone de la información sobre el propio robot almacenada en formato .bin (posiciones y velocidades del efector final a lo largo de la partida, etc.), y de aquella respectiva al juego en sí mismo, en formato .json (puntuación y tiempo total finales, diferentes eventos como la caída de la pelota en la deadZone, cuando el jugador coge diamantes o rocas, etc.). Todo este proceso de grabado y cierre de los archivos se implementa una vez inicializado, principalmente en la clase DataManager, como se ve en la Figura 2.18.

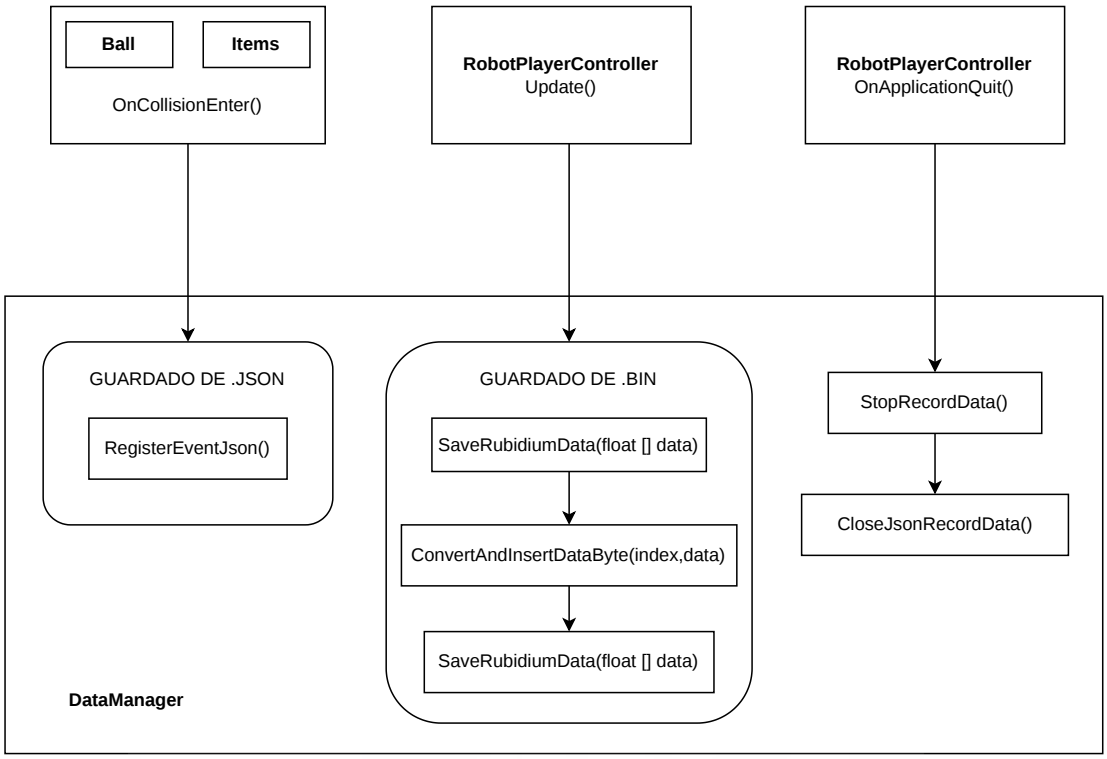


Figura 2.18: Diagrama de módulos y llamadas del almacenamiento local de Unity. Se guardan en formato .json los datos referentes a eventos (cuando cae la pelota o el jugador la coge, cuando alcanza o pierde un diamante y al esquivar o tocar una roca). Estos se detectan mediante las colisiones de los objetos, que llaman a RegisterEventJson para guardar la hora exacta con el evento al instante. Los datos del robot se almacenan en formato .bin. Cada frame, RobotPlayerController los almacena en una lista y llama a DataManager, que separa el array de datos en sus variables correspondientes en el método SaveRubidiumData y llama a ConvertAndInsertDataByte para transformarlas a binario y guardarlas en el buffer. Cuando se han tomado 5 muestras de cada variable se pasan al fichero temporal. RobotPlayerController detecta cuando la aplicación está a punto de cerrarse y llama a StopRecordData, que libera el escritor y el archivo binario temporal, el cual copia a continuación en el definitivo. Además, llama a CloseJsonRecordData para guardar el tiempo y puntuación finales antes de liberar el escritor.

2.5.7. Comunicación con REVIRE

Como se comenta anteriormente, existe además una transferencia de datos a REVIRE. Esta información, también se almacena en una base de datos local.

A diferencia del almacenamiento local que se escribía directamente en el disco duro desde Unity, la comunicación con REVIRE tiene una estructura Cliente-Servidor con un flujo unidireccional. El juego de Unity es el cliente, encargado iniciar el envío de datos y subir la información al final de la partida (ver Figura 2.19). REVIRE

es el Servidor, que inicia la escucha, recibe los datos y los guarda en la base de datos del disco duro. Si bien las pruebas realizadas en este TFG se hacen con pocos usuarios y con objetivos de investigación, cuando se lleva a cabo la aplicación clínica con mayor número de usuarios, tener un servidor facilita la correcta ordenación de la información y su seguridad, ya que no se puede acceder al archivo y modificarlo, sino únicamente extraer los datos.

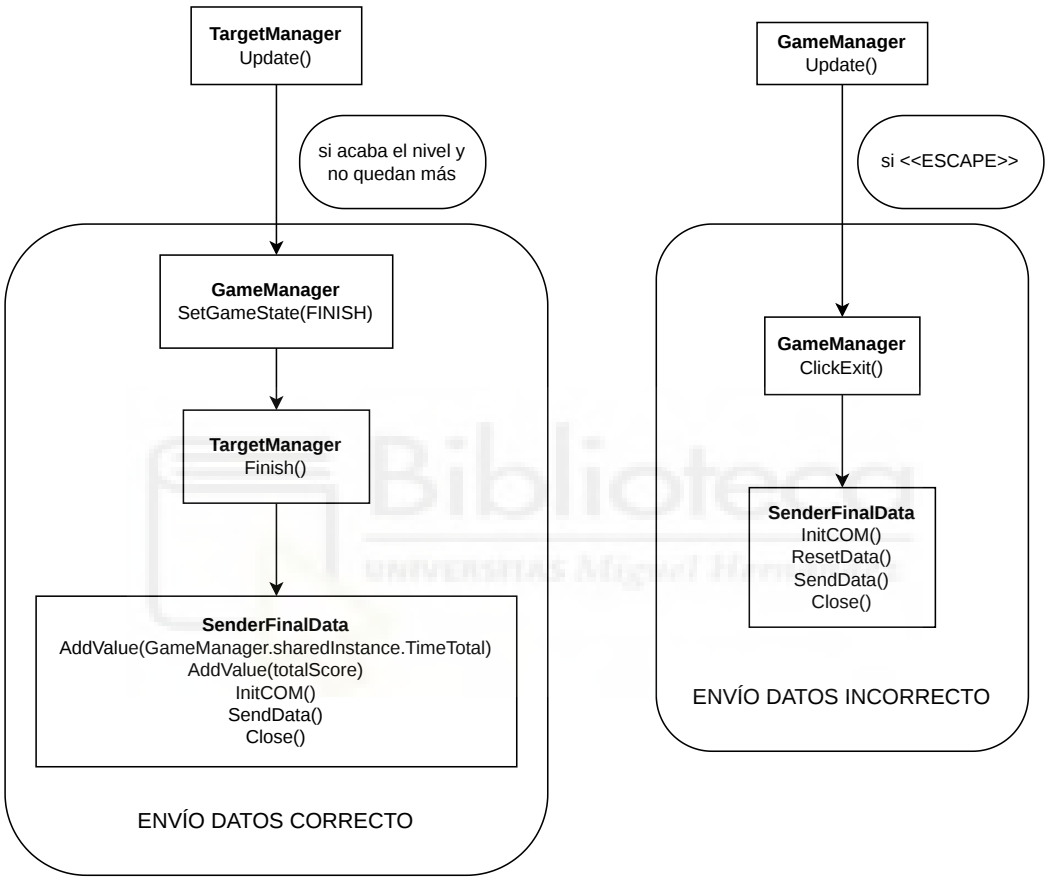


Figura 2.19: Diagrama de comunicación con REVIRE para el envío de datos. Cuando la partida se desarrolla con normalidad, TargetManager comprueba cada frame si se ha pasado de nivel y si ya no quedan más, en cuyo caso cambia el estado del GameManager a FINISH, lo cual produce la llamada al método Finish del TargetManager y este a SenderFinalData para añadir a la lista que se envía, el tiempo y la puntuación de la partida. En caso de finalizar antes de lo esperado al presionar «ESCAPE», detectado en el Update del GameManager, se llama a SenderFinalData para borrar el contenido de la lista en caso de tener algo y enviar "-1.0f" al programa.

2.5.8. Comunicación con Rubidium

Por último, se explica el bloque de la comunicación con Rubidium durante la partida, una vez está ya está inicializada. Sabiendo que para este juego el modo de asistencia es libre, no es necesario realizar ningún cálculo de fuerzas, sino que lo más importante es la conversión de la posición del efector final del robot a coordenadas del objeto del jugador en la pantalla dentro de Unity. Como se observa en la Figura 2.20, dicho proceso de conversión se lleva a cabo en el método FixedUpdate del RobotPlayerController que se ejecuta a una frecuencia fija de de 50Hz (cada 20ms) independientemente de los frames. Esto es muy importante para comunicarse con el hardware físico, ya que si la frecuencia con la que se gestiona los datos de posición del robot variase, se producirían saltos e inconsistencias en el jugador de la pantalla.

Si no se detecta el robot, se pasa al control del jugador por teclado, implementado en la clase PlayerController. Por el contrario, si se detecta comienza la secuencia en la que se obtienen los datos enviados por el robot y se guardan en variables de RobotPlayerController que se utilizan para actualizar la posición del jugador mediante el método UpdatePose. También se llama al método UpdateForces, que por ser el modo de asistencia libre, devuelve un array de ceros. Finalmente, se registran todos los datos restantes de la lista dataRb, que se busca almacenar mediante DataManager, en variables separadas que se agrupan en una lista diferente en Update del RobotPlayerController para evitar que estos valores cambien mientras se trabaja con ellos.

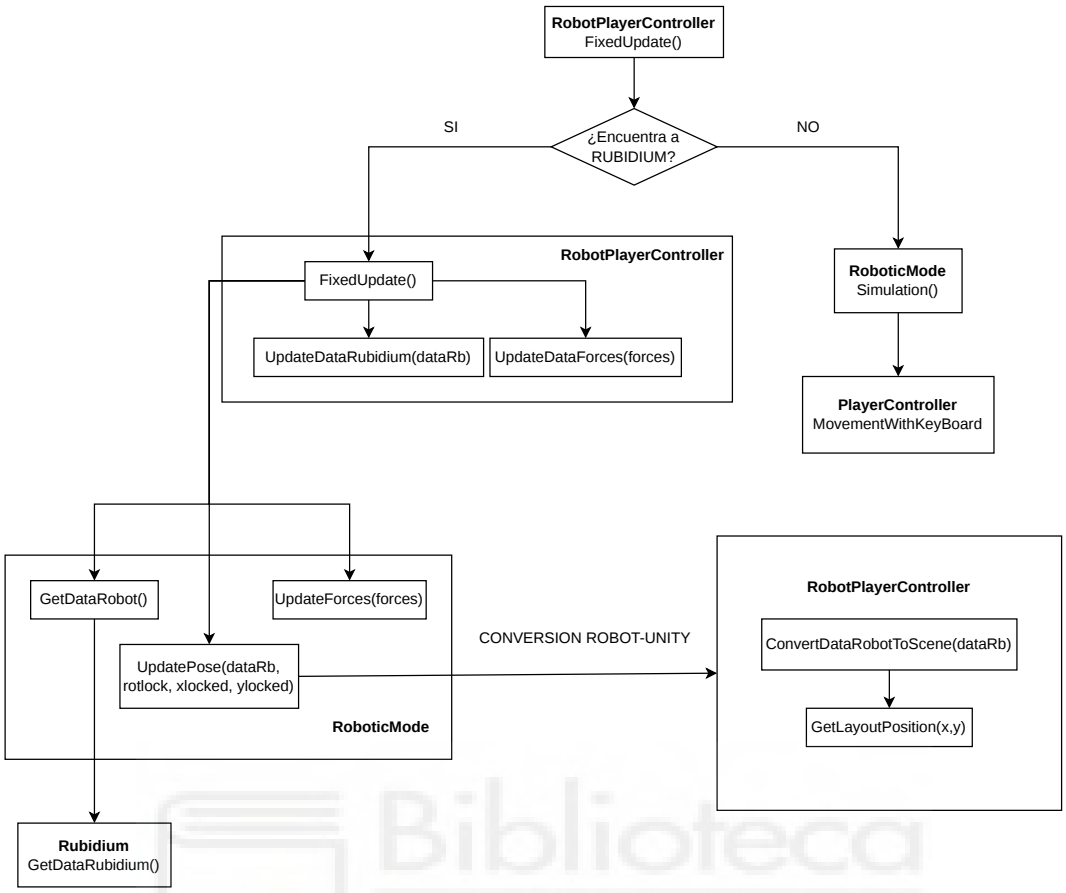


Figura 2.20: Diagrama de comunicación con Rubidium. GetDataRobot guarda los datos del robot en el array dataRb. UpdatePose llama ConvertDataRobotToScene que pasa las coordenadas a mm siendo los argumentos de GetLayoutPosition, que hace la conversión como se explicaba anteriormente, en función del Layout. Los valores se ajustan en caso de sobrepasar los límites visuales. UpdateForces devuelve un array de 10 ceros por ser modo libre. Después, dentro de RobotPlayerController se llama a UpdateDataRubidium y UpdateDataForces para guardar los datos en las variables de las listas creadas por Update para su almacenamiento. Si hay un error encontrando al robot, lanza el modo de simulación y activa el control por teclado, implementado en la clase PlayerController.

2.6. Experimentación.

Como se ha comentado, se han introducido varios parámetros con el objetivo de poder modificar la dificultad del juego de acuerdo con los requerimientos de cada persona. Para comprobar el efecto que tiene la variación de estos parámetros, se han realizado dos tipos de pruebas, cada una con tres sujetos sanos.

Una vez recogidos los datos necesarios de cada prueba, se ha procedido al analizarlos.

Para el procesamiento y análisis de la información, se utiliza Python junto con el entorno de Jupyter Notebooks integrado en VS Code, elegido por la capacidad que ofrece de combinar bloques de código con celdas de texto y gráficos, lo que facilita la organización y comprensión mediante una estructura modular clara del flujo de trabajo que permite ejecutar y visualizar las distintas celdas por separado.

A continuación, se explica en qué consisten las pruebas mencionadas y qué se espera de ellas.

2.6.1. Evaluación de la repercusión de las rocas en la actividad del paciente

Tiene como objetivo específico comprobar las repercusiones de añadir rocas (parámetro «RockDensity») como elemento adicional al juego, para fomentar el movimiento del brazo del paciente. Por una parte, busca aprovechar al máximo el rango de movimiento del brazo, es decir, provocar que aunque la pelota o los diamantes se concentren mayoritariamente en la misma zona de movimiento del jugador, este se vea obligado a desplazarse por todo el recorrido de la pantalla para evitar las rocas, generando así movimientos más amplios y repartidos. Además, trata de evitar que el paciente quede inmóvil esperando mientras la pelota rebota con otros objetos, provocando así un ejercicio continuo, que aumente el movimiento en términos generales. Para ver los resultados se analizarán distintos datos del robot.

En la prueba, todos los parámetros se han de mantener constantes con los valores mostrados en la Tabla 2.2, salvo RockDensity, que varía del 0 al 3 entre todas sus posibilidades de dificultad, en diferente orden para cada sujeto. Por tanto, cada uno ha realizado 4 ejercicios. Cada sesión tiene dos escenarios con un tiempo límite de dos minutos por nivel.

Tabla 2.2: Configuración de la evaluación de las rocas.

Parámetro	**Valor**	**Parámetro**	**Valor**
TimeTotal	2:00 min	InitialSpeed	5
Repeticiones	2	ItemsSpeed	5
Amplitud	200	PlayerSize	1
Layout	0	Movement	0
Lateralidad	0		

2.6.2. Evaluación de la repercusión de la variación de los parámetros en la dificultad

Consiste en evaluar, mediante la puntuación obtenida, de qué manera se ve influida la dificultad de la actividad cuando se modifican los diferentes parámetros que se han introducido con tal fin, con el objetivo de estudiar de qué forma podrían definirse diferentes niveles de dificultad con estos parámetros, y ver como afectan en la actividad del usuario, tanto para el juego en horizontal (Movement = 0) como en vertical (Movement = 1). Estos parámetros a evaluar son el tamaño del jugador («PlayerSize»), y la velocidad de la pelota («InitialSpeed») y de los objetos («ItemsSpeed»), valiendo estas dos lo mismo entre ellas. Esto se comprueba mediante la puntuación obtenida en cada caso.

Para evaluar los parámetros mencionados, cada ejercicio tiene un único escenario igual para todos, y 3 minutos de tiempo límite, estimado suficiente para realizar con éxito la prueba. Se establecen dos niveles de velocidad para cada modo, siendo mayores los del vertical, teniendo en cuenta que el recorrido de ida y vuelta de la pelota y la distancia que deben recorrer los objetos que caen, es mayor que en el horizontal. Estas dos velocidades se combinan con los dos tamaños del jugador. De esta manera, se obtienen 8 ejercicios, que se identifican de la A a la H como se observa en la Figura 2.21. En la Tabla 2.3, aparecen los valores de los parámetros constantes en todos los ejercicios.

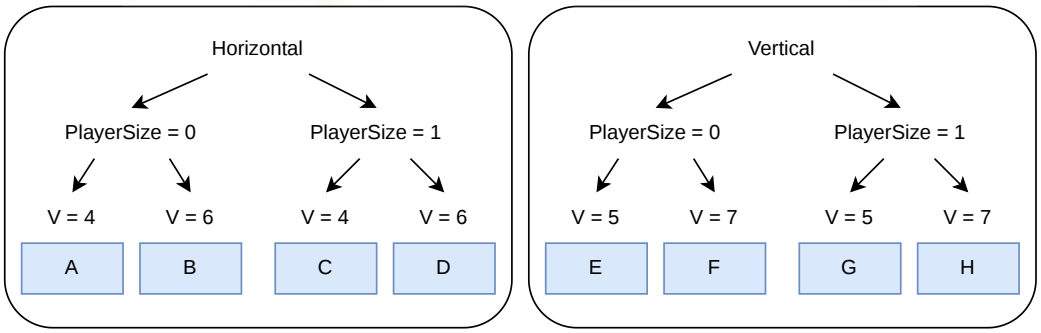


Figura 2.21: Ejercicios de la evaluación de la dificultad según la variación de «PlayerSize» y «InitialSpeed» = «ItemsSpeed» = V, con su identificador de la A a la H.

Tabla 2.3: Configuración de la evaluación de la dificultad en función de la variación de los parámetros.

Parámetro	**Valor**	**Parámetro**	**Valor**
TimeTotal	3:00 min	Layout	0
Repeticiones	1	Lateralidad	0
Amplitud	200	RockDensity	2 (medio)

3. RESULTADOS Y DISCUSIÓN

Es esta sección se presentan los datos y gráficas obtenidos a partir de la experimentación de las dos pruebas y se comentan exponiendo posibles justificaciones y mejoras para trabajos futuros.

3.1. Evaluación de la repercusión de las rocas en la actividad del paciente

Para comprobar los objetivos mencionados de esta evaluación, se han visualizado las trayectorias del efector final del robot en el plano XY realizadas por cada sujeto para los distintos modos de rocas como se muestra en la Figura 3.1.

Observando las trayectorias se puede ver como al aumentar la dificultad generalmente aumenta la dispersión de puntos en el eje y, pese a que el movimiento busca ser horizontal. Esto plantea la necesidad de adaptar este juego para un modo de asistencia robótica, en lugar de únicamente libre, restringiendo el movimiento en el eje y mediante un túnel (o en el eje x en el caso del modo de movimiento vertical). De esta manera, el terapeuta también podría aprovechar la fijación de la trayectoria para ejercitar distintos tipos de movimientos en función de la distancia entre el paciente y el robot.

Además, el rango horizontal supera siempre los 200 mm de amplitud del juego elegidos para las pruebas. Esto quiere decir que los sujetos continúan desplazando el efector final más allá de los límites visuales de la pantalla, lo que podría deberse a movimientos incontroladamente amplios por reacciones rápidas. Para evitar que eso sucediera con pacientes reales provocando un daño mayor en los brazos, convendría también introducir extremos en la asistencia.

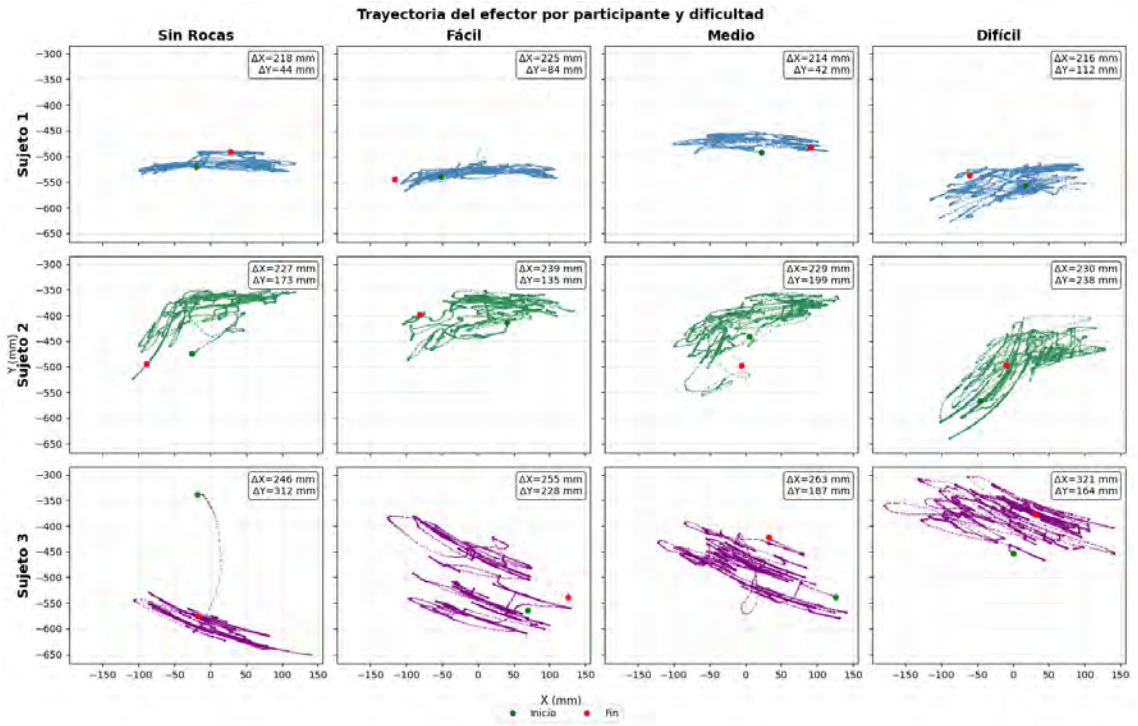


Figura 3.1: Trayectorias del efector final realizadas por cada sujeto durante los diferentes modos de configuración del parámetro «RockDensity», y el rango de movimiento en los ejes x e y en cada caso.

Además, estas trayectorias se han analizado generando mapas de calor con el máximo global de cada sujeto para visualizar la concentración de puntos del robot en el eje x del plano (puesto que el modo de juego es horizontal), pudiendo observar así si el movimiento del paciente ha sido homogéneo o se ha desplazado por unas zonas más que por otras de forma significativa en función del nivel de dificultad (ver Figuras 3.2, 3.3, 3.4).

La concentración de puntos en el eje x depende en gran parte de la estrategia personal de cada sujeto en cada caso. Sin embargo, en todos los mapas de máxima dificultad, la densidad del movimiento es mayor en una o varias zonas del centro, lo que puede indicar que se obliga al usuario a volver a una posición central constantemente para poder reaccionar de forma más rápida a cualquier desplazamiento necesario hacia los laterales.

No siempre se consigue aumentar el desplazamiento por los extremos de la pantalla de acuerdo con la densidad de las rocas como se pretendía. Esto puede deberse a que, siendo el objetivo esquivarlas, solo provocan la necesidad de hacerlo en zonas dónde el jugador se mueve para alcanzar la pelota o los diamantes. Por tanto, si estos no caen en los extremos no se estimula a pasar por ahí. Para resolverlo y aprovechar de forma efectiva el rango del juego, los diamantes deberían caer constantemente de la

parte superior al igual que las rocas, en lugar de depender de que la pelota golpee los ladrillos, los cuales pueden quedar repartidos de forma concentrada y escasa.

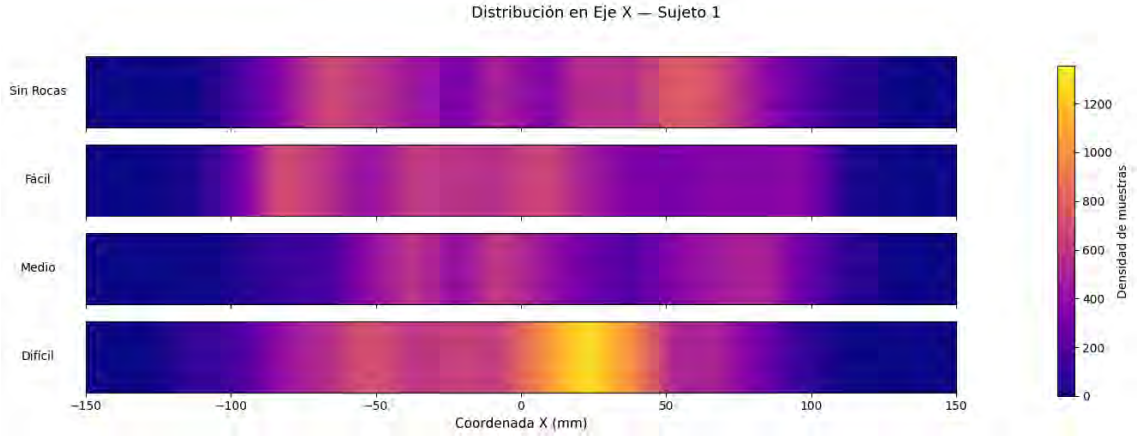


Figura 3.2: Concentración del movimiento del efector final en el eje x a lo largo del espacio de trabajo para los diferentes modos de configuración de «RockDensity» para el Sujeto 1.

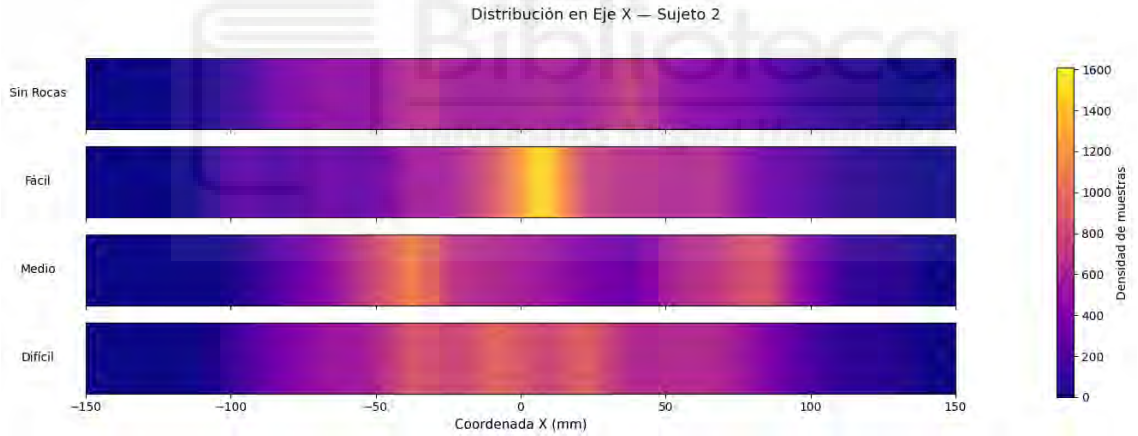


Figura 3.3: Concentración del movimiento del efector final en el eje x a lo largo del espacio de trabajo para los diferentes modos de configuración de «RockDensity» para el Sujeto 2.

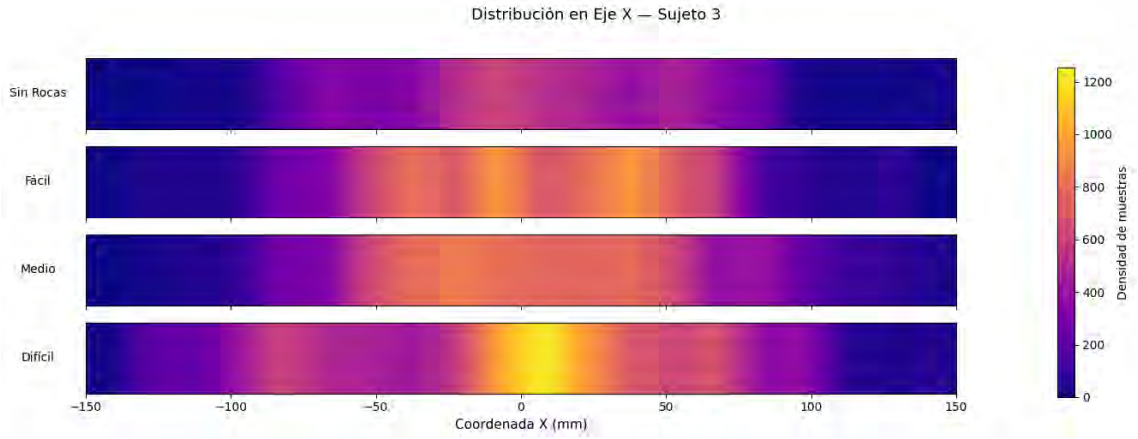


Figura 3.4: Concentración del movimiento del efector final en el eje x a lo largo del espacio de trabajo para los diferentes modos de configuración de «RockDensity» para el Sujeto 3.

Otro factor importante en la evaluación del desempeño de los usuarios bajo el efecto de las rocas es la velocidad que ejercen en cada caso. Se ha calculado la velocidad media mediante el valor absoluto de la velocidad instantánea, evitando la cancelación de desplazamientos en sentidos opuestos, y la desviación estándar en cada caso como se ve en la Figura 3.5.

Los resultados de la velocidad media global aumentan con los niveles medio y difícil, aunque así lo hace también la desviación estandar, por lo que no se puede hablar de una tendencia clara. A pesar de eso, el aumento en la media se puede explicar por la necesidad de movimientos más bruscos y rápidos para sortear mayor cantidad de obstáculos. En cuanto al nivel fácil, los resultados sugieren que, al menos para sujetos sanos, no implica ningún aumento de la dificultad.

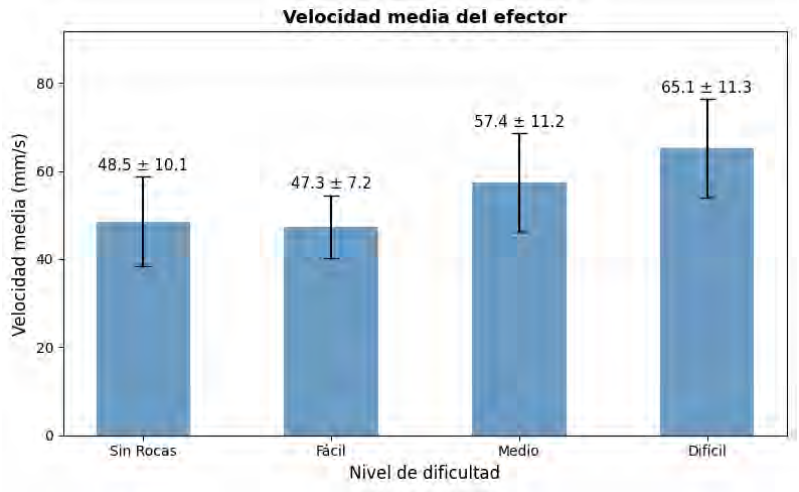


Figura 3.5: Velocidad media global para cada configuración de «RockDensity» y su desviación estándar.

3.2. Evaluación de la repercusión de la variación de los parámetros en la dificultad

Con el fin de visualizar los efectos sobre la dificultad de variar los diferentes parámetros, se genera un gráfico de barras para cada modo (horizontal y vertical) con la media de la puntuación relativa obtenida por los usuarios para cada configuraión y su desviación estándar (ver Figura 3.6). Es necesario representar la puntuación relativa ya que el número máximo de puntos que se puede obtener varía en función de la cantidad de diamantes que caigan, que es aleatoria.

Puesto que todos son sujetos sanos, se observan puntuaciones bastante parejas y altas, habiendo completado las partidas sin dificultades significativas. Se observa que para el mismo tamaño del jugador (configuraciones A, B, E y F para el pequeño, y C, D, G, H para el grande), la media de los resultados siempre es mejor para la velocidad lenta (configuraciones A, C, E, G), como era de esperar.

Sin embargo, aunque en el modo vertical el efecto de variar el tamaño del jugador es el esperado, obteniendo mayor puntuación para la misma velocidad cuando este es más grande, en el modo horizontal sucede lo contrario. Esta inconsistencia en los efectos de variar el «PlayerSize» puede explicarse porque a pesar de que aumentar el tamaño parecía ventajoso para alcanzar la pelota y los diamantes con más facilidad, la mayor cobertura del rango también dificulta el sorteo de las piedras. Este parámetro no cumple, por tanto, con la funcionalidad propuesta.

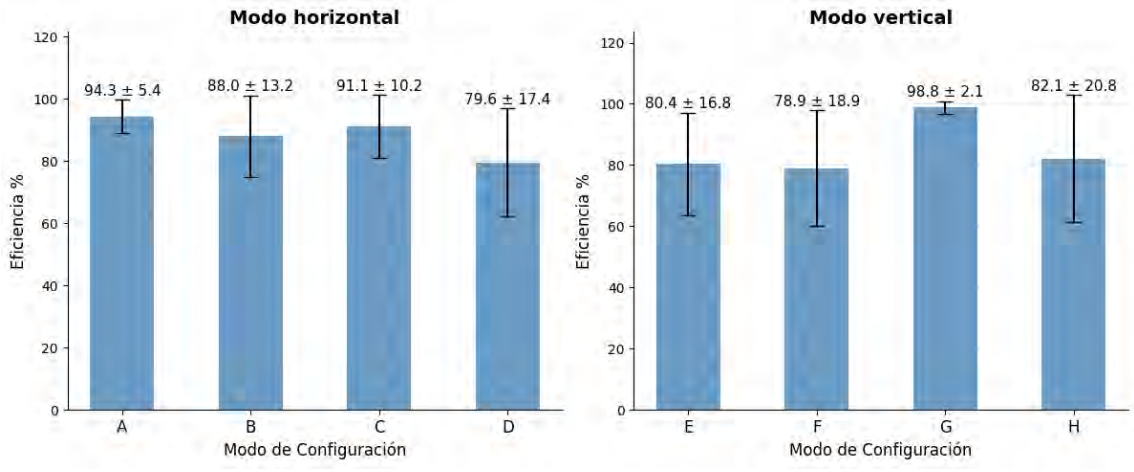


Figura 3.6: Puntuación relativa global y desviación estándar para cada configuración de los parámetros «PlayerSize», «InitialSpeed» y «ItemsSpeed», siendo estas dos iguales entre sí en cada modo. La designación de las configuraciones es la representada en la Figura 2.21.

Cabe destacar que la puntuación se ve afectada, además de por la habilidad del usuario, por la toma de decisiones. En ocasiones, este debe decidir entre salvar la pelota y esquivar una roca cercana o alcanzar un diamante que se encuentra lejos. Puesto que es un juego que involucra la toma de decisiones de los sujetos, se han registrado distintos eventos de las partidas para poder representar en gráficos circulares a qué se han debido los fallos de cada uno de ellos (ver Figura 3.7).

En el resumen global de los fallos, se puede observar como en general hay una tendencia a dejar caer las pelotas por coger diamantes, mayor que el número de las que se caen exclusivamente por motivos motores, sin necesidad de elección. Esto tiene sentido, pues los diamantes dan más puntos que los que quita perder la pelota. Por el contrario, en ningún caso se dejó caer por esquivar rocas.

Figura 3.7: Resumen global de los fallos para todas las configuraciones de «Rock-Density» para cada sujeto y para todos juntos.

4. CONCLUSIONES

En el presente TFG se ha desarrollado un juego serio para la rehabilitación del miembro superior, principalmente libre y pensado para terapias sin asistencia por parte del robot. Se han introducido distintos parámetros con el objetivo de modificar la dificultad mediante su variación, ofreciendo así la posibilidad de adaptación a las necesidades de cada paciente.

Los parámetros más relevantes en este aspecto son la velocidad de la pelota, la velocidad de los objetos y el tamaño del jugador, evaluadas en conjunto mediante distintas combinaciones, y la densidad de las rocas, evaluada de forma independiente.

Los resultados sugieren que el aumento o disminución del tamaño del jugador no tiene un papel eficaz en la adaptación de la dificultad. Tampoco hay diferencia aparente entre el modo fácil de rocas y cuando no cae ninguna. En cualquier caso, los demás parámetros parecen afectar como se esperaba, aunque no de forma significativa.

Por otra parte, se ha observado como el juego no promueve el movimiento del brazo a lo largo de todo el rango del espacio de trabajo de forma homogénea. Además, en las gráficas de las trayectorias del robot obtenidas, la dispersión de muestras de la posición del efector final fuera de la trayectoria deseada es significativa y aumenta con la dificultad.

Por último, cabe destacar que si bien se menciona que el juego es principalmente libre, también estimula la actividad cognitiva mediante la toma de decisiones.

5. TRABAJOS FUTUROS

Habiendo comentado los resultados obtenidos, cabe recalcar que estos son muy preeliminares. Las pruebas se han realizado con un escaso número de sujetos sanos, por lo que sería necesario continuar evaluando la configuración del juego en pacientes con lesiones reales para obtener resultados concluyentes.

Con el objetivo de promover el movimiento a lo largo de todo el rango del juego, se propone la modificación del elemento de los diamantes, cambiando su comportamiento de manera que aparezcan también de forma continua en cualquier punto, como las rocas. Para optimizar el efecto de los elementos, ambos podrían ser configurados para caer en una posición u otra tratando de homogenizar el paso del efector final por todos los puntos, en lugar de aleatoriamente.

Además, se plantea el desarrollo del juego para el modo de asistencia de túnel con extremos, evitando así la dispersión demasiado alejada del eje del movimiento deseado y fuera de la amplitud establecida.

BIBLIOGRAFÍA

[1] Á. Soto, F. Guillén-Grima, G. Morales, S. Muñoz, I. Aguinaga-Ontoso y R. Fuentes-Aspe, «Prevalencia e incidencia de ictus en Europa: revisión sistemática y metaanálisis,» en *Anales del sistema sanitario de Navarra*, SciELO Espana, vol. 45, 2022.

[2] D. SANIDAD, «Guía de Práctica Clínica sobre el Manejo del Ictus en Atención Primaria,»

[3] P. Poli, G. Morone, G. Rosati y S. Masiero, «Robotic technologies and rehabilitation: new tools for stroke patients' therapy,» *BioMed research international*, vol. 2013, n.º 1, pág. 153 872, 2013.

[4] M. Murie-Fernández, P. Irimia, E. Martínez-Vila, M. J. Meyer y R. Teasell, «Neurorrehabilitación tras el ictus,» *Neurología*, vol. 25, n.º 3, págs. 189-196, 2010.

[5] H. M. Qassim y W. Wan Hasan, «A review on upper limb rehabilitation robots,» *Applied Sciences*, vol. 10, n.º 19, pág. 6976, 2020.

[6] E. Oña, R. Cano-de La Cuerda, P. Sánchez-Herrera, C. Balaguer y A. Jardón, «A review of robotics in neurorehabilitation: Towards an automated process for upper limb,» *Journal of healthcare engineering*, vol. 2018, n.º 1, pág. 9 758 939, 2018.

[7] B. Bonnechère, «Serious games in physical rehabilitation,» *DOI*, vol. 10, págs. 978-3, 2018.

[8] P. S. Ardakani, H. Moradi, F. Bahrami y M. Akbarfahimi, «A web-based gamification of upper extremity robotic rehabilitation,» en *2021 International Serious Games Symposium (ISGS)*, IEEE, 2021, págs. 43-47.

[9] J. M. Catalán, A. Blanco-Ivorra, J. V. García-Pérez, Y. Vales, D. Martínez-Pascual, S. Ezquerro, A. Garrote, T. Costa, L. D. Lledó y N. García-Aracil, «Patients' physiological reactions to competitive rehabilitation therapies assisted by robotic devices,» *Journal of NeuroEngineering and Rehabilitation*,

vol. 20, n.º 1, pág. 41, abr. de 2023. DOI: 10.1186/s12984-023-01163-2 dirección: https://doi.org/10.1186/s12984-023-01163-2

[10] Tyromotion GmbH, *AMADEO: Finger-Hand Rehabilitation Device for Clinics and Therapists*, https://tyromotion.com/en/amadeo-for-clinics-and-therapists/, Accedido el: 1 de junio de 2026, 2023.

[11] S. Crea, M. Cempini, S. Mazzoleni, M. C. Carrozza, F. Posteraro y N. Vitiello, «Phase-II Clinical Validation of a Powered Exoskeleton for the Treatment of Elbow Spasticity,» *Frontiers in Neuroscience*, vol. 11, pág. 261, 2017. DOI: 10.3389/fnins.2017.00261 dirección: https://www.frontiersin.org/journals/neuroscience/articles/10.3389/fnins.2017.00261/full

[12] ReHyb Project Consortium, *ReHyb–SSSA–IUVO Exoskeletons for Upper Limb Rehabilitation*, https://rehyb.eu/2020/07/rehybsssa-iuvo-exoskeletons-for-upper-limb-rehabilitation/, Accedido: junio de 2026, jul. de 2020.

[13] Hocoma / MedicalExpo, *Exoesqueleto de rehabilitación de miembro superior: ArmeoPower*, https://www.medicalexpo.es/prod/hocoma/product-68750-438436.html, Accedido el: 1 de junio de 2026, 2026.

[14] T. Nef, M. Guidali y R. Riener, «Arm Exoskeleton ARMin,» en *Neurorehabilitation Technology*, V. Dietz, T. Nef y W. Z. Rymer, eds., Springer, Cham, 2016, págs. 303-319. DOI: 10.1007/978-3-319-28603-7_17 dirección: https://link.springer.com/chapter/10.1007/978-3-319-28603-7_17

[15] Hocoma AG, *Pretraining material UserScript - ArmeoSpring Pro*, Versión en español (13/12/2023), Hocoma Knowledge Platform, dic. de 2023. dirección: https://knowledge.hocoma.com/wp-content/uploads/2023/12/Pretraining-material_UserScript_-ArmeoSpringPro_13122023-1_Spanish.pdf

[16] iF International Forum Design GmbH, *ArmeoSpring Pro - iF Design Award Winner*, https://ifdesign.com/es/winner-ranking/project/armeospring-pro/568617, Accedido: junio de 2026, 2026.

[17] Plant Automation Technology, *InMotion ARM - Interactive Rehabilitation Device by Bionik Laboratories*, Consultado el 23 de abril de 2024, 2024. direc-

ción: https://www.plantautomation-technology.com/products/bionik-laboratories-corp/inmotion-arm

[18] AEMEDI - Asociación de Empresas de Medicina, *Sistema de Rehabilitación Interactivo InMotion ARM*, https://www.aemedi.es/inmotion.htm, Accedido: junio de 2026, 2026.

[19] MedicalExpo / Bionik Laboratories, *Aparato de rehabilitación de brazo / de codo / de hombro InMotion ARM*, https://www.medicalexpo.es/prod/bionik-laboratories/product-119369-821823.html, Catálogo técnico en línea. Accedido: junio de 2026, 2026.

[20] Eodyne Systems, *Eodyne: Science-Based Neurorehabilitation Solutions RGS*, https://eodynesystems.com/en/, Sitio web oficial. Accedido: junio de 2026, 2026.

[21] Eodyne Systems, *RGS+ — Rehabilitation Gaming System at Home*, https://eodynesystems.com/en/rgs-plus/, Sección oficial de telerehabilitación domiciliaria. Accedido: junio de 2026, 2026.

[22] Eodyne Systems, *Rehabilitation Gaming System (RGS@home)*, Vídeo de YouTube, Captura de pantalla utilizada como Figura. Accedido: junio de 2026, 2020. dirección: https://www.youtube.com/watch?v=4na0OtdDqS0

[23] BIONIK, *InMotion ARM Robot: Evaluation Overview*, Vídeo de YouTube, Disponible en el canal oficial de BIONIK. Accedido: junio de 2026, 2023. dirección: https://www.youtube.com/watch?v=WeUHM_fQWxU

[24] BIONIK, *BIONIK InMotion ARM/HAND - Therapy Activities*, Vídeo de YouTube, Muestra de videojuegos interactivos y tareas de alcance. Accedido: junio de 2026, 2019. dirección: https://www.youtube.com/watch?v=dA2nvZX9fN0