



Escuela
Politécnica
Superior

Aplicación de realidad virtual para la mejora de condiciones motoras finas en personas con necesidades especiales



Grado en Ingeniería Robótica

Trabajo Fin de Grado

Autor:

Antonio Santo López

Tutor/es:

Miguel Ángel Cazorla Quevedo

Félix Escalona Moncholí

Junio 2020



Universitat d'Alacant
Universidad de Alicante

Aplicación de realidad virtual para la mejora de condiciones motoras finas en personas con necesidades especiales

Rehabilitación motora

Autor

Antonio Santo López

Tutor/es

Miguel Ángel Cazorla Quevedo

Ciencia de la computación e inteligencia artificial

Félix Escalona Moncholí

Ciencia de la computación e inteligencia artificial



Grado en Ingeniería Robótica



Escuela
Politécnica
Superior



Universitat d'Alacant
Universidad de Alicante

ALICANTE, Junio 2020

Resumen

En este Trabajo de Fin de Grado se desarrolla una aplicación de realidad virtual enfocada a la progresiva mejora de la motricidad fina de personas con necesidades especiales. La razón fundamental por la que llevar a cabo este proyecto es aportar alternativas en el ámbito de la rehabilitación médica con el propósito de que el conjunto de las diferentes técnicas existentes sea lo suficientemente amplio como para no caer en el desaliento del duro proceso de rehabilitación. Jugar, observar tu progreso semanal, la ambición de poder superarte semana tras semana en un marco tangible, físico, pueden ayudar positivamente de manera psicológica durante el proceso.

Antes del desarrollo de la aplicación de realidad virtual, se da un contexto general acerca de enfermedades cerebrovasculares, ictus o infarto cerebral y sus consecuencias motoras. Seguidamente, se explica la tecnología sensorial empleada para la detección de las extremidades, conocida como Leap Motion, y su consiguiente integración con el software Unity. Por último la explicación y desarrollo de la propuesta llevada a cabo.

Agradecimientos

En primer lugar, agradecer a mis tutores el tiempo invertido y la preocupación porque el proyecto saliera adelante. Desde la semana uno en la que me fue asignado, las conversaciones y reuniones fueron constantes, siempre ayudando en la medida de lo posible. Además, en estos tiempos que corren, con la situación tan delicada en la que nos hemos encontrado, no han relajado su interés, y por ello les dedico estas palabras de agradecimiento.

No puedo dejar pasar la oportunidad de plasmar unas palabras hacia las personas que me han acompañado durante este viaje, aunque en realidad me acompañaron durante toda la vida. A mis padres, por mirar siempre por mi futuro, incluso cuando yo no lo veía tan claro, y por ayudarme en todos los desafíos que me planteó la vida. Por último, a mi hermana, por ser el espejo en el que mirarme en muchos aspectos.

Finalmente, y no menos importante, a mis amigos. Todos comenzamos esta aventura de la Universidad con todo lo que ello implicaba. Sin embargo, no noté diferencia a cuando estábamos todos los días riendo en el instituto. Además, no creo que fuera lo que soy hoy sin ellos. Mención especial a un amigo, que se preocupó por el trabajo y me echó una mano en la búsqueda de documentación.

A todos vosotros por estar, y a ella especialmente por permanecer en mi desde el principio.

Índice general

| | | |
|----------|---|-----------|
| 1 | Introducción | 1 |
| 1.1 | Objetivos | 2 |
| 2 | Estado del arte | 3 |
| 2.1 | Introducción a la rehabilitación | 3 |
| 2.2 | Recuperación del miembro superior tras ictus | 5 |
| 2.3 | Neurorrehabilitación | 6 |
| 2.3.1 | Principios del aprendizaje motor | 6 |
| 2.3.2 | Normas de la Neurorrehabilitación | 7 |
| 2.4 | Gamificación | 8 |
| 2.4.1 | Gamificación en rehabilitación | 9 |
| 2.4.2 | Rehabilitación en pacientes oncológicos | 10 |
| 2.4.3 | Rehabilitación en pacientes de movilidad reducida | 10 |
| 2.4.4 | Autogestión de medicamentos y afecciones crónicas | 11 |
| 2.4.5 | Rehabilitación tras accidente cerebrovascular | 11 |
| 2.5 | Rehabilitación Virtual | 12 |
| 2.5.1 | Principales beneficios | 14 |
| 3 | Metodología | 17 |
| 3.1 | Unity | 17 |
| 3.2 | Leap Motion | 19 |
| 3.2.1 | Clase Bone | 20 |
| 3.2.2 | Clase Arm | 20 |
| 3.2.3 | Clase Finger | 20 |
| 3.2.4 | Clase Hand | 21 |
| 4 | Desarrollo de la propuesta | 23 |
| 4.1 | Integración de Leap Motion en Unity | 23 |
| 4.2 | Instalación de módulos | 23 |
| 4.3 | Leap Motion en Unity | 24 |
| 4.4 | Diseño del entorno del videojuego | 25 |
| 4.5 | Componentes del Videojuego | 28 |
| 4.5.1 | Personaje | 28 |
| 4.5.2 | Animaciones | 29 |
| 4.5.2.1 | Reposo | 29 |
| 4.5.2.2 | Correr | 30 |
| 4.5.2.3 | Saltar | 30 |
| 4.5.2.4 | Transiciones entre animaciones | 30 |
| 4.5.3 | Colisionadores | 31 |

| | | |
|----------|--|-----------|
| 4.5.4 | Puntos de guardado | 32 |
| 4.6 | Gestos | 33 |
| 4.6.1 | Clase Detector | 33 |
| 4.6.2 | Clase AbstractHoldDetector | 34 |
| 4.6.3 | Clases de detección de gestos | 34 |
| 4.7 | Traza del videojuego | 35 |
| 4.7.1 | Menú principal | 35 |
| 4.7.2 | Menú de opciones | 36 |
| 4.7.3 | Menú de gestión de gestos | 36 |
| 4.7.4 | Escenario principal del juego | 37 |
| 4.7.5 | Puntuación Final | 38 |
| 4.8 | Resultado | 38 |
| 5 | Conclusiones | 39 |
| | Bibliografía | 41 |
| | Lista de Acrónimos y Abreviaturas | 45 |

Índice de figuras

| | | |
|------|--|----|
| 2.1 | Técnicas de rehabilitación actuales | 4 |
| 2.2 | Ejercicios de rehabilitación | 5 |
| 2.3 | Sistema Octalysis. Extraído de: (<i>Yu-kai Chou sistema Octalysis</i> , 2020) | 9 |
| 2.4 | Re-mission: Nanobot's Revenge | 10 |
| 2.5 | Aplicación Glooko | 11 |
| 2.6 | Rehabilitación Virtual | 12 |
| 2.7 | Ejemplo de interfaz sensorial | 13 |
| 2.8 | Ejemplo de interfaz motora | 13 |
| 2.9 | Consolas usadas en rehabilitación | 14 |
| 2.10 | Ejemplo de Realidad Aumentada (AR) | 14 |
| | | |
| 3.1 | Hollow Knight realizado en Unity | 18 |
| 3.2 | Controlador Leap Motion | 19 |
| 3.3 | Representación en el Visualizador de la interfaz de Leap Motion. | 20 |
| 3.4 | Detección de la dirección de los dedos | 21 |
| 3.5 | Orientación de la mano a partir de la dirección y el vector normal a la palma | 21 |
| | | |
| 4.1 | Jerarquía de GameObjects para representar la mano en Unity. | 24 |
| 4.2 | Background con diferentes iluminaciones. | 25 |
| 4.3 | Conjunto de sprites individuales. | 26 |
| 4.4 | Comienzo del juego. | 26 |
| 4.5 | Nivel 1. | 27 |
| 4.6 | Nivel 2. | 27 |
| 4.7 | Nivel 3. | 27 |
| 4.8 | Parámetros asociados al personaje. | 28 |
| 4.9 | Sprite del personaje en reposo. | 29 |
| 4.10 | Conjunto de posturas de reposo. | 29 |
| 4.11 | Conjunto de posturas corriendo. | 30 |
| 4.12 | Postura de salto. | 30 |
| 4.13 | Estados y transiciones de animación. | 31 |
| 4.14 | Colisionador de una moneda. | 32 |
| 4.15 | Puntos de control. | 33 |
| 4.16 | Gestos a detectar | 34 |
| 4.17 | Scripts asociados al modelo de la mano. | 35 |
| 4.18 | Menú principal. | 35 |
| 4.19 | Menú de opciones. | 36 |
| 4.20 | Menu de gestión de gestos. | 37 |
| 4.21 | Escena principal. | 37 |

4.22 Puntuación Final. 38

1 Introducción

En la actualidad, los accidentes cerebrovasculares son uno de los motivos más frecuentes de asistencia neurológica urgente, y constituyen un grave problema de salud pública. En España, representan la primera causa de mortalidad específica en la mujer y la tercera en el hombre. Concretamente, en la Comunidad Valenciana, en 2015, existían 56.500 casos registrados de Daño cerebral adquirido (DCA), siendo un 80% de ellos debido a ictus o derrame cerebral (Conselleria, 2017).

Además, según datos de la Organización Mundial de la Salud, es la segunda causa de muerte en el mundo con 6 millones de personas, y la primera de discapacidad por lesiones cognitivas. Del 65% de los pacientes que sobreviven a un primer Accidente cerebrovascular (ACV) y que continúan viviendo de manera independiente, el 35% presenta una discapacidad significativa y requiere ayuda para la realización de las actividades de la vida diaria. A pesar de que muchos pacientes que han presentado un ACV tienen un deterioro sensitivo o motor primario, una gran proporción de estos pacientes presenta alteraciones muy importantes de las funciones visoespacial o sensorio-motriz (Newport, 2006).

Definimos el ACV como la interrupción del suministro de sangre a una parte del cerebro, o cuando un vaso sanguíneo del mismo se rompe, derramando sangre en espacios que rodean a las células cerebrales. Aun cuando un accidente cerebrovascular ocurre en los lugares recónditos del cerebro, los síntomas del mismo son fáciles de detectar. Entre éstos figuran los siguientes:

- Entumecimiento o debilidad repentina, especialmente en un lado del cuerpo.
- Confusión repentina o problemas con el habla o la comprensión.
- Problemas repentinos en la vista con uno o ambos ojos.
- Problemas repentinos en el andar, mareos o pérdida de equilibrio o coordinación.
- Dolor de cabeza severo repentino sin causa conocida.

Para entender el contexto del que partimos con esta enfermedad, ya hace más de 2.400 años, uno de los considerados como padre de la medicina, Hipócrates, reconoció y describió el accidente cardiovascular de manera sintomática como “el inicio repentino de parálisis”. Lo que antiguamente conoceríamos como apoplejía, término aplicado a cualquier persona con disfuncionalidades motoras.

Naturalmente, a lo largo de la historia se han producido grandes avances en el campo de la medicina. Sin embargo, no fue hasta hace dos décadas cuando realmente investigadores y clínicos adquirieron el suficiente conocimiento de nuestro cerebro como para ser capaces de detectar los principales factores de riesgo de esta condición médica (NINDS, 2017).

Entre ellos, destacan:

- Diabetes.
- Enfermedades cardíacas.
- Factores de riesgo genéticos.
- Hipertensión.
- Tabaquismo.

Como toda afección, los accidentes cerebrovasculares conllevan consecuencias directas en la cotidianidad de sus pacientes, con una intensidad variable dependiendo en gran medida de la rapidez y eficacia con la que se pueda llegar a tratar. Es así que resulta frecuente que se padezcan secuelas físicas, cognitivas y funcionales.

Las tratadas en este proyecto son las secuelas físicas que se pueden manifestar, como la parálisis en un lado del cuerpo, llamada hemiplejía. O puede no ser tan debilitante como la parálisis, tratándose de debilidad de un lado del cuerpo o hemiparesis. La parálisis o la debilidad puede afectar sólo a la cara, un brazo, o una pierna, o puede afectar a todo un lado del cuerpo y a la cara.

Una persona que sufre un accidente cerebrovascular en el hemisferio izquierdo del cerebro presentará parálisis del lado derecho. A la inversa, una persona que sufre un accidente cerebrovascular en el hemisferio derecho del cerebro presentará déficit en el lado izquierdo del cuerpo. Los déficits motores pueden resultar del daño de la corteza motora en los lóbulos frontales del cerebro o del daño de las partes inferiores del cerebro, tales como el cerebelo, que controla el equilibrio y la coordinación.

1.1 Objetivos

El objetivo principal del proyecto es el desarrollo de una aplicación de realidad virtual orientada a la rehabilitación de pacientes con dificultades en la actividad motora fina, normalmente dada por daño neurológico debido a un accidente cerebrovascular.

Por tanto, se hace uso de tecnología hardware, como es Leap Motion, y software, Unity, que nos permitan capturar y detectar gestos con nuestros miembros superiores y plasmar una acción en un determinado entorno virtual. Este objetivo se puede descomponer en diferentes subtareas:

- Diseño del entorno virtual con el motor de videojuegos Unity.
- Instalación de Leap Motion.
- Integración de Leap Motion en Unity.
- Captura de los gestos y programación de las acciones en el entorno.

Por otra parte, existe la finalidad de concienciar sobre el arduo camino de rehabilitación que en muchas ocasiones deben llevar los pacientes. Para ello, se intentará introducir el concepto vanguardista de gamificación de la rehabilitación, muy presente en la actualidad como tema de investigación, por ser una alternativa eficaz y realista que puede optimizar la práctica clínica.

2 Estado del arte

2.1 Introducción a la rehabilitación

La rehabilitación es un proceso metódico y cooperativo entre paciente y equipo médico. Por tanto, ambos deben estar coordinados para el alcance de un objetivo común. Las terapias de rehabilitación actuales buscan la recuperación total de déficits tras lesiones neuromusculares, asegurando así una calidad de vida cercana a la disfrutada previamente. Sin embargo, no existe una única metodología concreta a implementar en los pacientes, ya que hay muchos tipos de terapia física y en solo algunas se dan evidencias clínicas de su eficacia. Y es que la fisioterapia en general se apoya, como es lógico, en aspectos teóricos asentados, pero también es un gran factor a tener en cuenta la formación y la experiencia del especialista, lo que provoca que el proceso no sea un camino pautado en todas las direcciones que se han de tomar para una satisfactoria recuperación. No obstante, a pesar de toda la subjetividad intrínseca en este marco de la medicina moderna, el proceso de rehabilitación debe cumplir una serie de principios básicos (García, 2016):

1. Establecimiento de un acuerdo entre paciente y equipo terapéutico.
2. Análisis de los déficits de comportamiento bajo principios de reorganización cerebral y comportamiento neuromuscular.
3. Medidas fiables de incapacidad, función y actividad antes, durante y después de la terapia.
4. Planificación del programa de rehabilitación de acuerdo a la pragnosis del paciente.
5. Administración de un nivel de terapia adecuado tanto en intensidad como en repetitividad.
6. Presencia de un entorno terapéutico adecuado orientado al aprendizaje motor.

Todo proceso de recuperación, además, consta de 4 fases (Conselleria, 2017):

- Fase aguda: Ocurre desde la aparición del daño hasta los primeros días desde el episodio. En esta fase el cerebro aún es vulnerable a cambios relacionados con la lesión. Se distinguen dos fases.
 - Fase crítica: La persona se encuentra en peligro y precisa de medidas avanzadas UCI.
 - Fase aguda: Tras superar la fase crítica, el paciente precisa de cuidados hospitalarios.

- Fase subaguda: Comprende desde la primera semana hasta el primer mes. Es en esta fase donde el cerebro comienza el proceso de neuroreparación, que comprende aquellos fenómenos que restauran la correcta estructura y/o función neuronal dañada. Suele ser una fase hospitalaria ya que requiere de unas condiciones controladas tanto físicas como ambientales.
- Fase postaguda: Se prolonga desde las primeras semanas hasta los primeros 6 meses. En esta fase el cerebro trata de establecer nuevas conexiones neuronales para compensar las funciones que son consecuencia de las áreas dañadas. El mecanismo por el que el cerebro se reorganiza es llamado neuroplasticidad.
- Fase crónica de mantenimiento: Representa la fase de post-rehabilitación, en la que los déficits siguen siendo parte activa de la vida del paciente. A pesar de ser consideradas secuelas, los mecanismos de neuroplasticidad siguen estando presentes, por lo que no se debe abandonar el entrenamiento para potenciar las funciones conservadas o bien para compensar las funcionalidades perdidas. El abandono asistencial, conlleva un retroceso en la funcionalidad de los pacientes y en su calidad de vida.

Como mencionaba anteriormente, no todas las técnicas aplicadas resultan ser efectivas. Actualmente, existen tres tipos de rehabilitación (Úbeda, 2018):

- Técnicas preparatorias: Intervenciones que preparan el sistema sensoriomotor para incrementar su plasticidad a través de la estimulación directa de los tejidos.
- Técnicas aumentativas: Intervenciones que incrementan los efectos de la interacción sensoriomotora durante la práctica.
- Entrenamiento de tareas específicas: Sesiones basadas en la práctica repetitiva de tareas específicas realizadas en un entorno real, con el fin de acelerar la generalización del aprendizaje en la vida real.

| Preparatorias | Aumentativas | Entrenamiento en tareas específicas |
|---|--|--|
| <ul style="list-style-type: none"> • Imaginación motora y visual • Estimulación táctil • Movilización de tejidos blandos • Movimientos pasivos • Observación de acciones • Terapia espejo • TMS, TDCS y TENS | <ul style="list-style-type: none"> • CIMT • Asistencia de robots • Biofeedback EMG • FES • Entrenamiento bilateral • Realidad virtual | <ul style="list-style-type: none"> • Habilidad con el miembro afectado • Habilidad con la limitación funcional |

Figura 2.1: Técnicas de rehabilitación actuales. Extraído de: (Úbeda, 2018)

2.2 Recuperación del miembro superior tras ictus

La limitación del brazo y de la mano es un problema muy frecuente que afecta al 75% de las personas que han sufrido un ACV. La mano es un órgano diseñado para realizar movimientos muy complejos y versátiles que permiten llevar a cabo múltiples actividades en nuestro día a día. Esto incluiría movimientos tales como los mostrados en la figura 2.2 y, por otro lado, movimientos de alcance, agarre, manipulación, control bimanual y movimiento individual de los dedos, entre otros. Este tipo de movimientos sofisticados están asociados a un control cortico-muscular específico. La planificación de su rehabilitación es un proceso muy complejo. En la mayoría de pacientes predomina la recuperación espontánea dentro de los 3 primeros meses, seguida de una recuperación por aprendizaje durante los 6 meses posteriores al ictus.

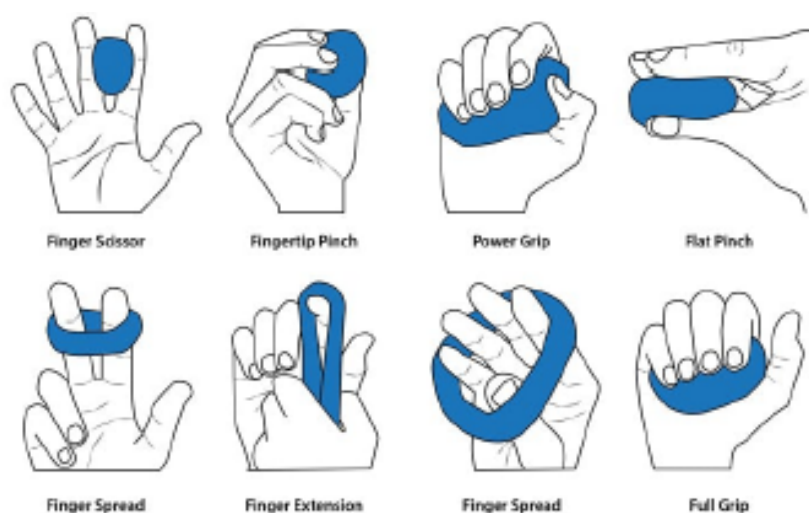


Figura 2.2: Ejercicios de rehabilitación. Extraído de: (*Ejercicios de rehabilitación, Flint Rehab, 2017*)

A diferencia de la marcha, la rehabilitación de miembro superior requiere una incorporación e integración muscular muy compleja que afecta a muchas articulaciones, desde el hombro hasta los dedos. En el caso de los miembros inferiores, un mínimo avance puede significar la obtención de la capacidad de marcha. Asimismo, la falta de estimulación en el miembro afectado puede conllevar que el paciente acabe por “olvidar” su uso, ya que si observamos el ejemplo en miembros inferiores ambos deben ser estimulados por igual para mantener el equilibrio o la marcha. Sin embargo, el paciente afectado en un solo brazo de parálisis tiende a inutilizarlo, siendo completamente dependiente de su miembro sano. De hecho, una de las terapias más efectivas tras la irrupción de un ictus en la vida del paciente es la Terapia de restricción-inducción de movimiento (CIMT). Se trata de una técnica de tratamiento específica (González, 2006) para personas que han sufrido un accidente cerebrovascular y que tienen como secuela una parálisis leve o moderada. Sus principios básicos incluyen la restricción del movimiento de la extremidad superior indemne durante un 90% del día, y la estimulación del uso de la extremidad superior afectada en sesiones terapéuticas enfocadas principalmente a aspectos funcionales.

En líneas generales, la mayoría de expertos terapéuticos en rehabilitación tras un accidente cerebrovascular dan por sentado la efectividad de los tratamientos físicos. Sin embargo, numerosos estudios (Ernst, 1990) muestran que no es tan sencillo de evaluar, aunque todos coinciden en que sí trae consigo beneficios la rehabilitación con fisioterapia. Enfoques fisioterapéuticos, que escapan al alcance de este proyecto, tales como Bobath, Brunnstrom, Clayton, Coulter, entre otros, constituyen diferentes escuelas de pensamiento sobre la recuperación motora, pero no necesariamente se ha demostrado un mejor resultado de alguna de ellas. Por tanto, la evidencia disponible hoy en día sugiere que no importa el enfoque del tratamiento, ya que si existe un método óptimo hasta ahora no lo hemos logrado identificar. Por tanto, la tendencia es el uso de terapias asequibles y que se puedan llevar a cabo por el mayor número de pacientes.

2.3 Neurorrehabilitación

La neurorrehabilitación se ha definido como el proceso clínico que tiene como finalidad minimizar los déficits funcionales en las personas como consecuencia de una lesión en el sistema nervioso y adquirir, por tanto, nuevas destrezas en la medida de lo posible. Esta subespecialidad se basa en la neuroplasticidad como principal factor para el aprendizaje motor del paciente.

La neuroplasticidad es, según la Organización Mundial de la salud, “la capacidad de las células del sistema nervioso para regenerarse morfológica y funcionalmente, después de estar sujetas a influencias patológicas, ambientales o del desarrollo, incluyendo traumatismos y enfermedades, permitiendo una respuesta adaptativa a la demanda funcional”(Rebolledo, 2003).

Generalmente, podemos distinguir dos tipos de aprendizaje:

- Aprendizaje implícito o no declarativo. La persona aprende en base a repeticiones de tareas que no precisan de un alto nivel de conciencia.
- Aprendizaje explícito o declarativo. Requiere de unos niveles de atención más altos que el aprendizaje implícito, y las tareas se realizan de forma consciente.

Aunque ambos tipos de aprendizaje son válidos, el más habitual en este tipo de rehabilitación es aquel que trata movimientos automatizados mediante procesos atencionales bajos.

2.3.1 Principios del aprendizaje motor

Muchos factores influyen en el aprendizaje motor, siendo algunos de ellos intrínsecos del individuo, como puede ser la edad, la raza o la predisposición genética (de la Fuente, 2013). Sin embargo, existen principios comunes y establecidos que permiten modelar el aprendizaje motor según estos factores:

- Intensidad del entrenamiento: Se cuantifica a partir del número de repeticiones de la tarea a entrenar. Se trata de uno de los requisitos fundamentales para la eficacia de la rehabilitación.
-

- **Variabilidad de la práctica:** Consiste en la elaboración ordenada y coherente de la organización de la práctica. Se han de tener en cuenta parámetros tales como, tiempo de actividad frente a tiempo de descanso, movimientos que describen una secuencia completa o actividades totalmente aleatorias, sesiones variables o sesiones repetitivas.
- **Retroalimentación o feedback:** Se trata de la información recibida por el sujeto durante y después de la realización de la terapia. Podemos distinguir dos canales de información de feedback: el procedente del propio sistema sensorial del paciente y el feedback procedente de una fuente externa. La calidad de esta información es un aspecto clave en el proceso de aprendizaje.
- **Motivación:** Es el principio de aprendizaje más personal y dependiente del sujeto, pero responde a condiciones externas que experimenta el paciente. Todo proceso de rehabilitación requiere de constancia, implicación y voluntariedad por lo que el paciente además de llevar consigo estos sentimientos debe verlos reflejados en su entorno, tanto familia como equipo médico.
- **Nivel de dificultad:** El correcto aprendizaje de una habilidad motora se debe plantear desde la perspectiva que su nivel de dificultad no sea ni sencillo ni muy complejo. Ya que ambos repercutirían negativamente en otros factores como el feedback o la motivación.

2.3.2 Normas de la Neurorrehabilitación

Cuando aparece una discapacidad, más allá de los condicionantes físicos que esta impone sobre la persona, el efecto se considera de tal magnitud que se aprecian de forma súbita alteraciones en su estilo de vida, proyectos, relaciones familiares y capacidad social. Es en este contexto donde la Neurorrehabilitación debe ser capaz de influir positivamente en la capacidad de aceptación de la discapacidad, en los objetivos para progresar en busca de una total autonomía y en una disposición emocional constructiva con el fin de lograr una reinserción total y satisfactoria (Conselleria, 2017).

Para que sea efectiva y de calidad, la neurorrehabilitación debe ser:

- **Individualizada.** Desarrollo personalizado de estrategias centradas en el paciente.
 - **Integradora.** La planificación de la rehabilitación debe ser diseñada por equipos multidisciplinarios constituidos por profesionales con motivación y altamente capacitados.
 - **Participativa.** Se debe contar con la cooperación activa de pacientes y su familia; para ello, es imprescindible una correcta información y una óptima relación de confianza con el equipo terapéutico.
 - **Ecológica.** Se debe intentar dotar a la persona afectada de la máxima autonomía posible, procurando así evitar dependencias asistenciales o tecnológicas.
 - **Continuada.** Asegurar la continuidad asistencial desde la fase inicial hasta las posibles complicaciones que puedan surgir en fases posteriores.
-

- Resolutiva. La rehabilitación debe estar dotada de recursos tanto humanos como materiales para que pueda ser llevada a cabo de manera eficaz ante cualquier distorsión durante el proceso.
- Adaptable al entorno social. Deberán buscarse respuestas que se adapten a las características específicas del ambiente del paciente y, al mismo tiempo, se trabajará en minimizar el impacto de la reinserción social del paciente.
- Holística o integral. Debe tener en cuenta tanto los aspectos físicos y cognitivos como los psicológicos, sociales y culturales que inciden en la personalidad de la persona afectada, su etapa evolutiva, estilo de vida y ambiente familiar.

2.4 Gamificación

La gamificación es el arte de introducir los elementos divertidos de los juegos en ámbitos relativamente aburridos y rutinarios pero que, por lo general, son de vital importancia. Este concepto se debe aplicar en aquellos entornos en los que la motivación y la acción conductual son claves para el éxito. Mientras se pueda definir una conducta deseada, la gamificación puede mejorar esa conducta.

Yu-Kai Chou es el responsable del desarrollo de lo que se conoce como el sistema Octalysis (Chou, 2019), resultado obtenido a partir del estudio durante muchos años de los aspectos que hacen atractivo a un juego. Como resultado de ese estudio, consiguió definir los ocho motores principales de la motivación.

El sistema Octalysis consiste en un análisis basado en un octógono, mostrado en la figura 2.3, cuyos lados representan distintos motores principales de la motivación. La clave de esos motores principales que componen cada lado de la figura es que cada actividad que realizas mientras juegas se basa en uno o más de ellos. Es decir, si no hay ningún motor, no existe motivación, por lo que no habrá conducta. Los motores son los siguientes (Freitas y cols., 2017):

1. Significado y vocación épicas. Se trata de la motivación personal que nace al ser conscientes de ser parte de algo mayor que nosotros mismos. Un ejemplo muy claro es la participación de la gente a la hora de compartir conocimiento en la comunidad científica, debido a que sentimos la necesidad de proteger el conocimiento humano, que es mayor que nosotros mismos.
 2. Desarrollo y Logro. Es la parte de motivación procedente del dominio y progreso de la actividad. En un juego, las monedas o puntuaciones no son más que contadores, pero dan cierta sensación de progreso, así que más allá de estar siempre repitiendo la misma acción con el mismo final día tras día, al menos se producen variaciones en la barra de progreso que alientan a la persona a dar más de sí.
 3. Potenciación de la creatividad y la retroalimentación. Se da a los usuarios una cantidad de recursos y opciones por las que pueden optar y se trata de ajustar el resultado final mediante retroalimentación con el equipo de desarrollo.
-

4. Propiedad y posesión. Es la sensación de pertenencia. Cuando creemos que algo nos pertenece, tenemos la necesidad de protegerlo. No solo se trata de pertenencia en sí, sino también de la afección que puedas tener hacia ello. Por ejemplo, por el tiempo dedicado, a pesar de que existan nuevas maneras mejores o más eficaces de realizarlo.
5. Influencia social y afinidad. Incorpora todos los elementos sociales que motivan a las personas. En esta categoría se incluyen la aceptación, competencia o la nostalgia.
6. Impaciencia. Es el motor que aborda la motivación a partir de la obtención de una recompensa que es inalcanzable en ese preciso instante. Esto motiva a las personas a volver para comprobar la disponibilidad.
7. Imprevisibilidad. En los juegos, es un factor muy importante, y es aquel que da lugar a la incertidumbre que genera el no saber qué vendrá después.
8. Pérdida. Al igual que en la vida real, es mucho más probable que llevemos a cabo una acción para evitar una pérdida que para conseguir una ganancia, ya que una pérdida es más dolorosa.

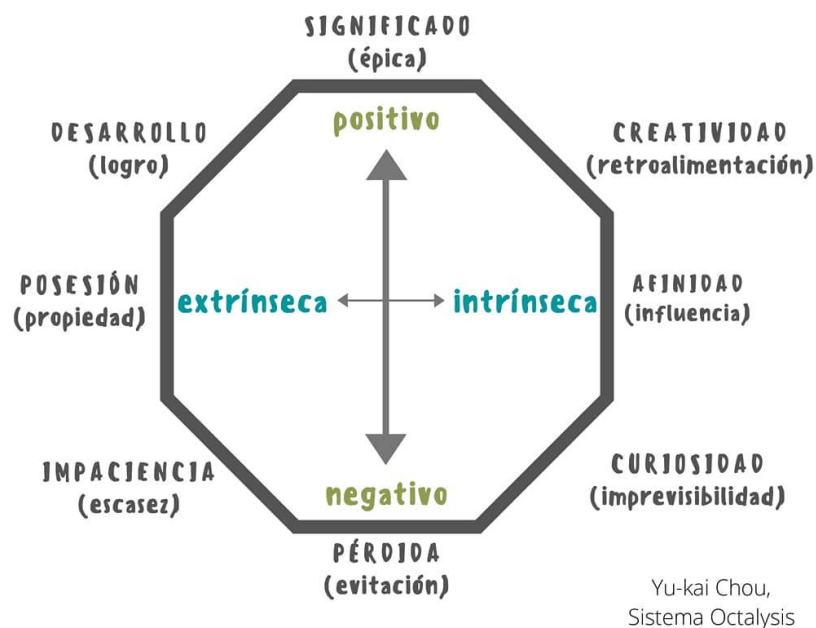


Figura 2.3: Sistema Octalysis. Extraído de: (Yu-kai Chou sistema Octalysis, 2020)

2.4.1 Gamificación en rehabilitación

En las últimas décadas se han desarrollado cambios en la manera de entender y de enfocar los procesos de rehabilitación en pacientes con ACV, como consecuencia a su vez del conocimiento de las enfermedades y del estudio de los componentes primordiales que se deben

dar para una recuperación satisfactoria. Es así como nace el concepto de gamificación en la rehabilitación, que de alguna manera es otra de las muchas actividades que se alejan de lo manual y optan por la tecnología.

Entendemos la gamificación como una técnica de aprendizaje que traslada las habilidades que una persona experimenta en un juego a otros ámbitos, como es el caso de la rehabilitación. El juego en sí mismo tan solo es el contenedor de aptitudes que fomentan los principios de aprendizaje motor mencionados anteriormente.

A pesar de tratarse de un concepto relativamente nuevo, en general, y muy novedoso particularmente en el ámbito de la medicina, ya existen aplicaciones con un impacto real en el estilo de vida de los pacientes. En las siguientes secciones veremos ejemplos de estas aplicaciones.

2.4.2 Rehabilitación en pacientes oncológicos

Re-mission: Nanobot's Revenge, figura 2.4, se trata de un videojuego destinado a niños o jóvenes adultos. Nace ante la necesidad de concienciar y ayudar a comprender la magnitud de una enfermedad que atenta contra sus vidas, ya que muchos de ellos no son conscientes de la importancia de tomar sus medicaciones regularmente.

El juego trata sobre un nanorobot, "Roxy", cuyo objetivo es la aniquilación del cáncer. Para ello usa armas derivadas de los tratamientos utilizados para eliminarlo como por ejemplo un cohete de antibióticos o armas radiactivas.



Figura 2.4: Re-mission: Nanobot's Revenge. Extraído de: (*Gamification in Healthcare: the Value of Fun*, 2020)

2.4.3 Rehabilitación en pacientes de movilidad reducida

Se trata de pacientes incapaces de mantener el equilibrio por sus propios medios. En estos casos, el nivel de actividad puede no llegar a ser el suficiente ya que sus actividades diarias no competen de manera directa en su nivel de actividad necesario para mantener en forma los miembros afectados.

Afortunadamente, durante los últimos años se han desarrollado interfaces que permiten la conexión entre la silla de ruedas y un ordenador. Los estudios que han probado este tipo de rehabilitación (O'Connor y cols., 2002) consistían en un videojuego de carreras en el que la velocidad del coche en el entorno era proporcional a la que el sujeto transmitía a su propia rueda. Se ha demostrado en pacientes que de esta manera se cumplen periodos de actividad más productivos y motivantes para el paciente.

2.4.4 Autogestión de medicamentos y afecciones crónicas

El tratamiento de afecciones crónicas es un camino escabroso, tedioso y repetitivo, por lo que aplicar técnicas de gamificación en este contexto ayuda a un mayor compromiso y motivación por parte del paciente.

Un ejemplo es la aplicación Glooko (Glooko, 2020). Se trata de una plataforma que proporciona una monitorización remota de la diabetes. Para ello, permite la conexión con medidores de glucosa, como se muestra en la figura 2.5. Alguno de los elementos que monitoriza son:

- Seguimiento de las medidas de glucosa, insulina, ingesta de alimentos.
- Atender a los posibles cambios de glucosa en sangre.



Figura 2.5: Aplicación Glooko. Extraído de: (Glooko, 2020)

2.4.5 Rehabilitación tras accidente cerebrovascular

Consiste en el diseño y programación de un videojuego que, además de concienciar a los pacientes sobre la complejidad del problema clínico que acarrearán, también les ayude de manera directa a superarlo o a, al menos, disminuir en la medida de lo posible sus déficits.

Un ejemplo sólido de este tipo de aplicación es la empresa suiza MindMize (MindMize, 2020). Desarrolla aplicaciones de realidad virtual con un dinamismo propio de un videojuego, tal y como se observa en la figura 2.6, para entrenar al cerebro en víctimas de derrames cerebrales. Este es el contexto sobre el que se aplica la gamificación en la propuesta de este proyecto, que se detalla más adelante.



Figura 2.6: Rehabilitación Virtual. Extraído de: (*MindMotion GO Increases Therapy Engagement and Adherence Through VR Games*, 2017)

2.5 Rehabilitación Virtual

Definimos rehabilitación virtual como la capacidad de integrar tecnologías tales como realidad virtual, videojuegos o realidad aumentada en un proceso clínico como es la rehabilitación.

La realidad virtual consiste en una forma de comunicación humano-ordenador. Es la recreación de una interfaz en 3 dimensiones como un espacio sintético que proporciona una simulación multisensorial del mundo en tiempo real, y en la que el ser humano se sumerge e interactúa a partir de la reacción a estímulos. Existen numerosos dispositivos que permiten el funcionamiento de esta tecnología, como pueden ser pantallas HMD, auriculares... y contribuyen en gran medida en la capacidad de inmersión del sistema.

En los últimos años, la realidad virtual se ha popularizado en numerosos ámbitos como la música, los videojuegos o la educación. Sin embargo, donde realmente ha demostrado un gran potencial es en su empleo para la rehabilitación tras accidentes cerebrovasculares, ya que solventa de manera directa algunos de los problemas que experimentan los métodos tradicionales de rehabilitación.

La forma de comunicación entre el usuario y el entorno virtual se da a partir de la interfaz, que puede ser de dos tipos:

- Interfaces sensoriales. Incluyen los elementos que permiten a nuestros sentidos ser capaces de sumergirse dentro del entorno virtual y darnos, por tanto, una sensación de presencia. La componen desde una simple pantalla de ordenador hasta los dispositivos montados en cabeza HMD, mostrado en la figura 2.7, la información auditiva para potenciar el ambiente, o incluso dispositivos hápticos que aportan un feedback sensorial de tacto o fuerza.
- Interfaces motoras. Se trata de la comunicación entre el usuario y el sistema a partir de movimientos. La interacción directa se basa en las conductas o gestos detectados por el sistema que conllevan una consecuencia dentro del entorno. Por ejemplo, manejar un objeto virtual. La interacción indirecta es la realización de esas mismas acciones a partir de hardware externo como puede ser un teclado, ratón o joystick, como se muestra en

la figura 2.8. Uno de los productos más conocidos de interacción directa es la Kinect, desarrollada por Microsoft.



Figura 2.7: Ejemplo de interfaz sensorial. Extraído de: (*Verizon's 5G Network Could Make Virtual Rehabilitation For Patients A Reality*, 2018)

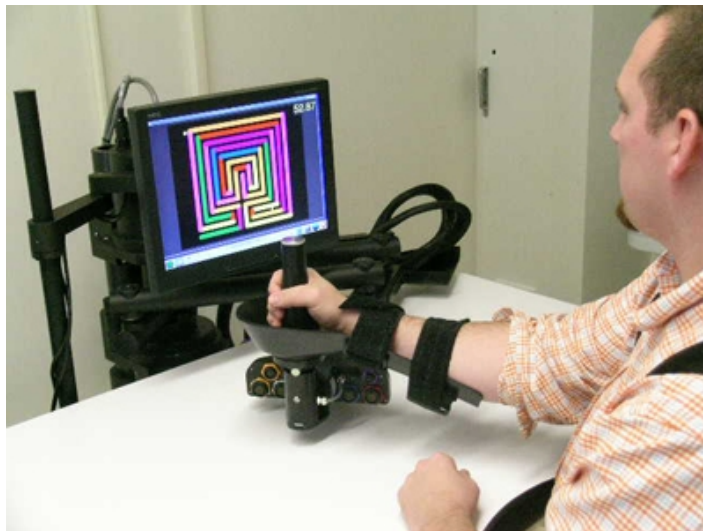


Figura 2.8: Ejemplo de interfaz motora. Extraído de: (Cogollor y cols., 2018)

Además, se cuenta con la ventaja que el mundo “gaming” se encuentra en constante desarrollo por lo que, aunque las grandes productoras de videojuegos o de consolas no se focalicen en el ámbito de la rehabilitación sino en el del entretenimiento, los sistemas desarrollados pueden ser adaptados para este uso. Como ejemplos recientes, tenemos las figuras 2.9a y 2.9b. Como consecuencia, cuanto más realismo se añade a la VR, acabamos derivando en el desarrollo de una tecnología en auge como es la Realidad Aumentada (AR).



Figura 2.9: Consolas usadas en rehabilitación. Extraídos de: (*Xbox360*, 2005), (*Nintendo wii*, 2006)

Se trata de una tecnología incipiente en muchos ámbitos, como en el de la rehabilitación. No es la recreación o la simulación de un mundo completamente virtual, sino que trata de ampliar y modelar el mundo real de nuestro alrededor a partir de información gráfica que se desarrolla en los dispositivos que la llevan a cabo. Un ejemplo de esta tecnología sería la figura 2.10.



Figura 2.10: Ejemplo de Realidad Aumentada (AR). Extraído de: (*A Brave New Augmented Reality Awaits*, 2018)

Existen estudios realizados (Mubin y cols., 2019) que comparan la progresión de actividades motoras en grupos de pacientes que aplican realidad aumentada con respecto a pacientes a los que no se aplica, y los resultados son esperanzadores. En definitiva, existen numerosas formas de aplicar el concepto de gamificación a la rehabilitación, ya sea en forma de videojuegos bidimensionales, juegos 3D, VR o AR. Ciertamente es que cada uno de ellos necesita de un cierto hardware que en parte dependerá de la condición física del paciente.

2.5.1 Principales beneficios

La rehabilitación en personas que padecen un cierto Daño cerebral adquirido (DCA) tiene como máxima el reaprendizaje de habilidades motoras que afectan en su calidad de vida de manera diaria. Para ello, se aprovecha la capacidad humana de moldear sus propias conexio-

nes cerebrales, lo que conocemos y ya definimos anteriormente como neuroplasticidad. Sin embargo, un mal enfoque de la tarea a realizar puede provocar una difícil reorganización en las cortezas sensoriales y motoras. El resultado de un enfoque erróneo conllevaría a un aprendizaje no deseado que en muchas ocasiones resulta ser un comportamiento difícil de reemplazar, y que podría llevar al paciente a una discapacidad a largo plazo. Sin embargo, ha sido probado que la rehabilitación virtual maximiza la eficacia de estos procesos neuroplásticos.

La realidad virtual es una tecnología flexible que apoya el entrenamiento repetitivo de alta intensidad, y que los pacientes encuentran motivador y atractivo. Son sistemas individualizados con múltiples opciones de configuración, con las que el paciente puede prevenir la frustración, el aburrimiento o la fatiga. Además, proporciona uno de los elementos más importantes de la rehabilitación como es el feedback. En sistemas inteligentes como videojuegos o realidad virtual este feedback es más objetivo, directo y rápido, lo que conlleva una acción correctiva por parte del sujeto.

Por último, una de las razones por la que la realidad virtual se ha convertido en una alternativa terapéutica es que genera un refuerzo positivo para motivar al individuo a realizar el movimiento. En contraposición con la terapia ocupacional, se han realizado estudios que hacen valorar esta hipótesis y es que la retroalimentación sensorial genera un mayor nivel de involucración que una realimentación directa por parte del terapeuta.

3 Metodología

3.1 Unity

Unity es un motor de desarrollo de videojuegos multiplataforma creado por Unity Technologies. Se trata de un conjunto de herramientas de desarrollo visual y componentes software que permiten crear videojuegos de manera simple. Nació en 2005 exclusivamente para generar proyectos en sistemas de plataforma Mac OS. Sin embargo, tras su éxito se continuó con el desarrollo del motor y sus herramientas. Actualmente, está disponible como plataforma de desarrollo para Microsoft Windows, Mac OS y Linux. A su vez estas tienen soporte de compilación para diferentes tipos de plataformas:

- Web
 - WebGL
- PC
 - Windows
 - OS X
 - LINUX
- Dispositivos Móviles
 - iOS
 - Android
- Consolas
 - Playstation
 - Xbox
 - Nintendo Wii/Switch
- Dispositivos de realidad aumentada
 - Oculus Rift
 - Playstation VR
 - HTC VIVE

Los componentes que posee un motor de videojuego pueden distinguirse en 5 áreas:

- Programa principal del juego. Lógica del videojuego.
- Renderización. Es el proceso por el cual se generan los gráficos por el ordenador. Se ocupa de mostrar escenarios, modelos, animaciones, texturas, sombras, iluminaciones.

- Audio: Se encarga de manipular las herramientas necesarias para la modificación y salida del audio a través de los altavoces del usuario.
- Motor físico. Responsable de recrear las leyes de la física de forma realista dentro del videojuego. Trata variables tales como la gravedad, masa, inercias, colisiones.
- Inteligencia artificial. Trata la definición de comportamientos, interacciones y toma de decisiones en el juego.
 - Scripting. Es la manera de crear las opciones que aporta el juego al jugador de forma controlada por parte del desarrollador. Por ejemplo, creando eventos.

El motor Unity aporta la capacidad de crear videojuegos 2D, ver figura 3.1, y 3D a través de una API de scripting primaria en C#, lenguaje que soporta de manera nativa. Este es el lenguaje de programación óptimo, aunque de por sí el entorno ha soportado a lo largo de su historia otros lenguajes como BOO y UnityScript, una versión particular de Java. El hecho de que C# sea el lenguaje dominante y que finalmente haya sido elegido de manera definitiva se debe a que es el lenguaje que prima en Unity, ya que le da un soporte especial y el motor basa toda su documentación en él. Además, ser el más utilizado por la comunidad de usuarios facilita el aprovechamiento de código de terceros a la hora de crear un juego propio. Sus características principales son:



Figura 3.1: Hollow Knight realizado en Unity. Extraído de: (*Hollow Knight*, 2017)

1. Entornos virtuales: Gráficos 2D como 3D presentan una calidad bastante buena.
 2. Curva de aprendizaje: Se trata de un motor gráfico cuyo aprendizaje durante la toma de contacto es relativamente complicado, pero a medida que te familiarizas con el entorno gráfico su aprendizaje se vuelve más intuitivo.
 3. Asset Store: Tienda Online para videojuegos. Es la más completa en la actualidad ya que presenta una gran cantidad de objetos y elementos que facilitan la tarea a los desarrolladores.
-

4. Multiplataforma: Aunque se ha mencionado anteriormente, el poder de realizar un código para diferentes plataformas es uno de los puntos fuertes que presenta Unity.

Por otra parte, es uno de los motores gráficos más utilizados por su política empresarial. El enfoque de la compañía es “democratizar el desarrollo de videojuegos”, siendo así posible que cualquier persona particular emprenda un proyecto desde cero sin costes por su utilización. Solamente en caso de que el juego supere los 100.000\$ de ingresos es necesario suscribirse a la licencia Pro de Unity.

3.2 Leap Motion

El sistema Leap Motion (*Leap Motion Controller*, 2012a) trata el reconocimiento y seguimiento de manos y dedos de manera que pueda interactuar con contenido digital a partir de un dispositivo hardware, como se puede ver en la figura 3.2, con sensores ópticos (dos cámaras IR monocromáticas) y tres LED infrarrojos. Los LED generan luz sin patrón y las cámaras generan casi 200 frames por segundo. Esta información es enviada al ordenador a través de un cable USB, donde es analizada por el software Leap Motion, cuyo funcionamiento la compañía ha guardado en secreto. Como consecuencia, el dispositivo reconoce el miembro superior en un cierto rango con gran precisión y, por tanto, simula la posición, gestos y movimiento realizados en un modelo de mano humana que recrea la interfaz de Leap Motion.

Los sensores se encuentran a lo largo de la dimensión longitudinal del dispositivo y colocados hacia arriba. Cuenta con una precisión de 0.7mm. Su campo de visión es aproximadamente de 150 grados y su alcance se encuentra aproximadamente entre 25-600mm por encima.

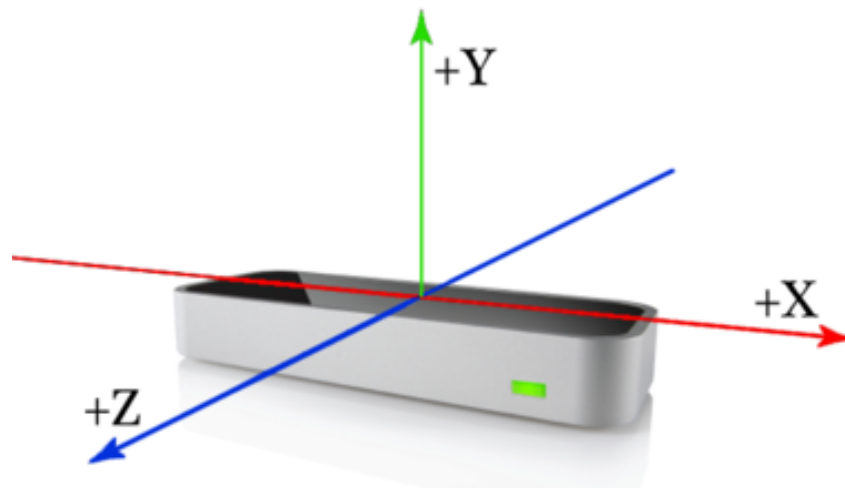


Figura 3.2: Controlador Leap Motion. Extraído de: (*Leap Motion Controller*, 2012b)

En lo pertinente al software, Leap Motion es capaz de discernir 27 elementos diferentes en una sola mano, incluidos huesos, articulaciones, dedos, palma y antebrazo. Para ello, cuenta con la implementación de clases en sus librerías que dan soporte al modelado de todos estos elementos, y de métodos que permiten gestionar adecuadamente esta información, véase la figura 3.3.

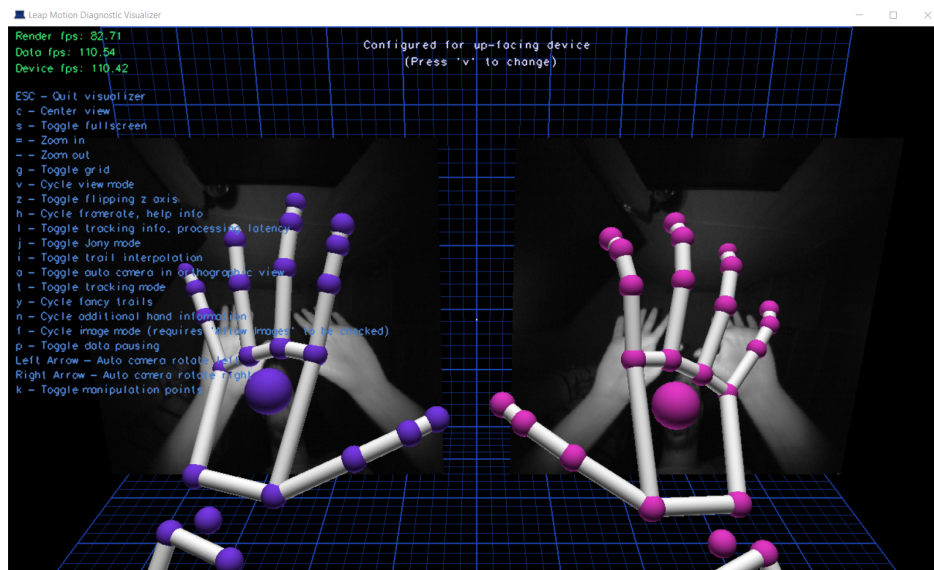


Figura 3.3: Representación en el Visualizador de la interfaz de Leap Motion.

3.2.1 Clase Bone

La clase Bone representa la detección y el seguimiento de un hueso de la mano. Puede ser cualquiera de los 4 diferentes:

- Falanges distales.
- Falanges medias.
- Falanges proximales.
- Metacarpianos.

A partir de su detección, existen parámetros que aportan información tal como dirección, centro o longitud.

3.2.2 Clase Arm

Es una clase que nace como consecuencia de la modularización del software ya que trata la detección y seguimiento del hueso del antebrazo, pero se implementa aparte para diferenciarse de los huesos de la mano.

3.2.3 Clase Finger

La clase Finger implementa el seguimiento y detección de un dedo. Como en el caso de los huesos, es capaz de modelar cualquiera de los dedos de la mano y contiene atributos como dirección, longitud o posición.



Figura 3.4: Detección de la dirección de los dedos. Extraído de: (*API overview*, 2014)

3.2.4 Clase Hand

Se trata de la clase que instancia en su constructor al resto de clases mencionadas, y es capaz de representar un modelo de la mano en su totalidad desde el codo hasta la punta de los dedos. El software de Leap Motion utiliza un modelo interno para proporcionar un seguimiento predictivo, incluso cuando algunas partes no son visibles. De este modo, utiliza las partes visibles y las observaciones pasadas para calcular la posición más probable de las partes de la mano que actualmente no son visibles.

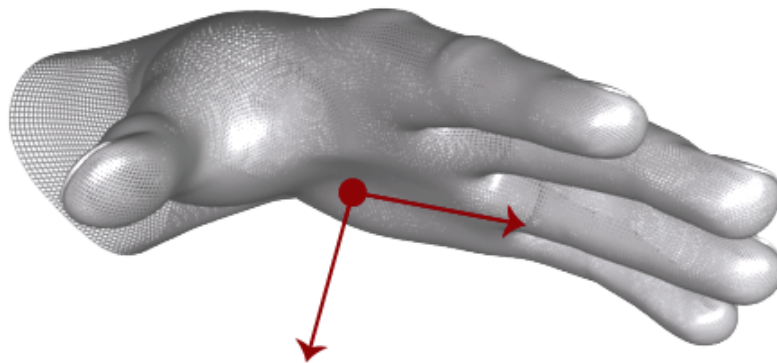


Figura 3.5: Orientación de la mano a partir de la dirección y el vector normal a la palma. Extraído de: (*API overview*, 2014)

4 Desarrollo de la propuesta

En este apartado se explican las distintas etapas llevadas a cabo para realizar una aplicación de realidad virtual enfocada a la progresiva mejora de la motricidad fina de personas con necesidades especiales.

Por tanto, se desmenuza el proyecto desde la obtención del software necesario, pasando por la creación del arte del videojuego, y acabando con la implementación de los gestos y sus consecuencias en la jugabilidad.

4.1 Integración de Leap Motion en Unity

Unity es un motor de desarrollo de videojuegos multiplataforma que de manera nativa no ofrece las herramientas necesarias para la detección de gestos y su uso como entrada en el control de una aplicación. Sin embargo, la empresa Ultra Leap, con su última actualización Leap Motion Orion 4.0.0, proporciona un Kit de desarrollo de software (Leap Motion's Unity SDK) que permite al desarrollador crear una aplicación informática para un sistema concreto. En este caso se trata de ciertos paquetes/módulos de software que permiten la compatibilidad y el uso de Leap Motion en Unity. Estos paquetes los podemos encontrar en (*Unity Assets for Leap Motion Orion Beta*, 2018) y únicamente soportan:

- Unity 5.6.2
- Unity 2017.1-4
- Unity 2018.1

4.2 Instalación de módulos

- Core Assets: Es la dependencia para cualquier módulo de los posteriores. Contiene las herramientas necesarias para obtener los datos de la mano en Unity.
- Interaction Engine: Implementa los controladores de realidad virtual que ofrece una API de interacción con funciones como: agarre, lanzamiento, retroalimentación de colisión.
- Graphic Renderer: Pretende ser un sistema que permite tratar una colección de objetos como un grupo, en lugar de como objetos individuales. Así se pueden aplicar muchas optimizaciones como, por ejemplo, aplicar sombras al acercar la mano a un objeto en concreto. Con este módulo se podría hacer de manera directa sin tratar a cada objeto de manera individual.
- Hands Module: Se trata del software que permite importar tus propios modelos de manos y darles vida a través de Unity.

4.3 Leap Motion en Unity

Para este proyecto se ha utilizado la versión 5.6.2 de Unity y la versión 4.4.0 del módulo Core Assets. Para su integración, a parte de la instalación del controlador de Leap Motion en Windows y el calibrado del dispositivo para una correcta medición y detección de los miembros superiores, se debe importar el paquete a Unity.

Una vez Unity ya cuenta con las herramientas necesarias para la detección de miembros superiores, entonces se procede a la creación de los objetos que formarán parte del juego, asignándoles los componentes necesarios (scripts, propiedades físicas...). La figura 4.1 muestra los elementos necesarios para modelar en Unity una mano.

En el primer lugar de la jerarquía de los objetos que se precisan para la detección, se encuentran:

- Controlador. En general, cualquier clase que necesite datos de la mano del sensor necesitará una referencia a un LeapServiceProvider para obtener esos datos. Es la clase que se comunica con el servicio Leap que se ejecuta en la máquina cliente.
- Partes físicas. Se trata de la incorporación de forma modular de los gameobjects que modelan la mano, tales como dedos, huesos, palma, antebrazo y sus propiedades como cuerpos rígidos.
- Representación de datos. Conjunto de esferas y cilindros para representar las manos utilizando todos los datos en bruto disponibles en una mano Leap. En nuestro proyecto no se ha precisado de esta capacidad del software ya que se trata de un juego 2D en el que no se hace falta representar el modelo 3D de la mano.

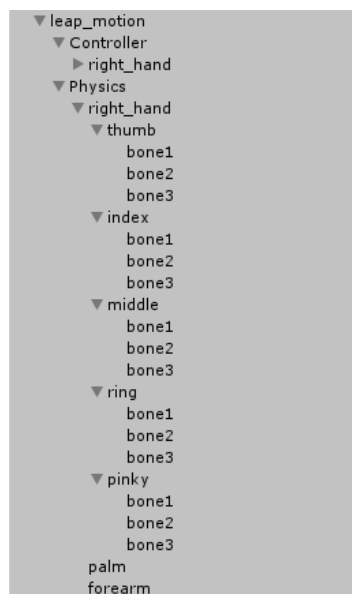


Figura 4.1: Jerarquía de GameObjects para representar la mano en Unity.

4.4 Diseño del entorno del videojuego

El diseño del juego desarrollado no dista mucho del de cualquier juego clásico de plataformas. Se trata de un conjunto de sprites, obtenidos de la tienda virtual de Unity, Asset Store, dispuestos sobre la escena de manera coherente con el objetivo de maquetar las plataformas y elementos necesarios que darán una cierta funcionalidad y sentido al entorno de nuestro videojuego.

En cuanto al background, se puede observar un mismo escenario a lo largo del videojuego, pero con distintos tonos de iluminación, dando la sensación de seguir una línea temporal, en la que el personaje comienza con la luz del día y continúa por la noche. Estos escenarios se muestran en la figura 4.2.

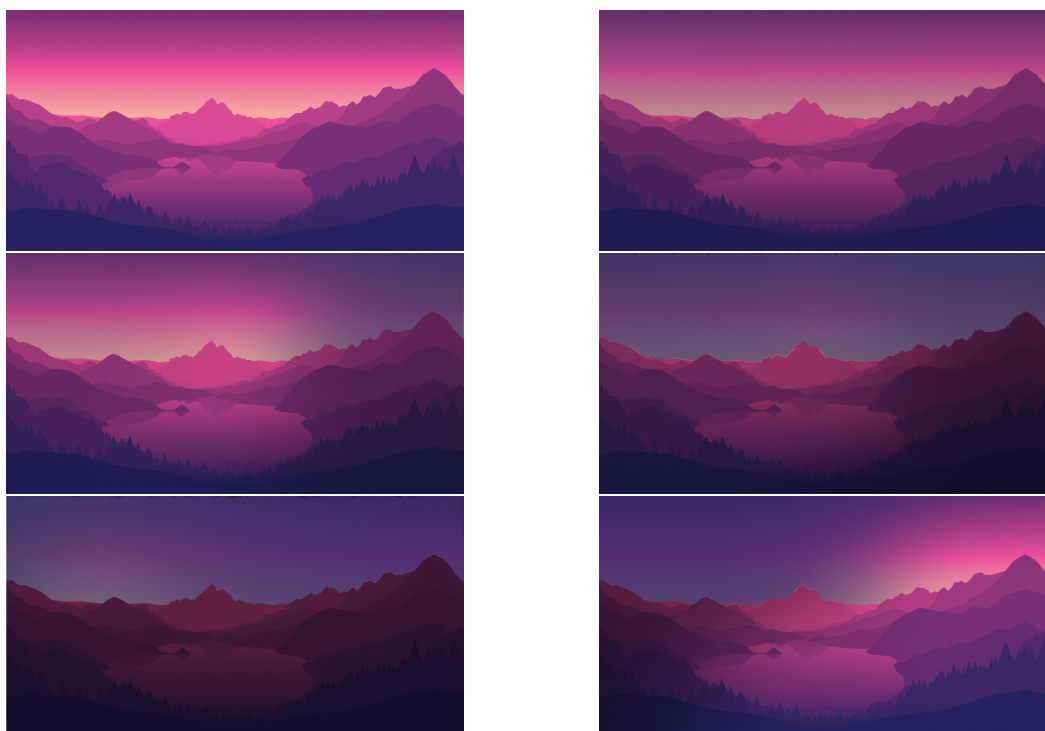


Figura 4.2: Background con diferentes iluminaciones.

Una vez establecido el fondo sobre el que transcurre la acción, debemos diseñar las plataformas. Para ello, antes de nada, individualizamos por cuadrículas la imagen que contiene todos los elementos que usaremos como unidades únicas en nuestro entorno. Unity cuenta para tal efecto con la opción "Sprite Editor", que permite dividir la imagen en tantos elementos individuales como hayan en ella, siempre y cuando hayamos indicado que se trata de una imagen contenedora de múltiples sprites, tal y como aparece en la figura 4.3.

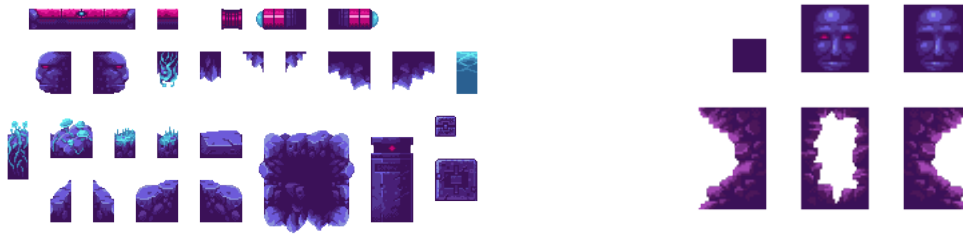


Figura 4.3: Conjunto de sprites individuales.

Por tanto, en este punto se trata de realizar una coherente e imaginativa disposición de estos elementos, como aparece en figura 4.4. Para simplificar la organización inicial del entorno, lo más sensato es agrupar en diferentes entidades todos aquellos sprites que representen una sola cosa. Además, se ha de tener en cuenta qué entidades se encontrarán en contacto con el personaje para así añadir una colisión a lo largo de la superficie de contacto.

El diseño de las plataformas es manual, a partir de la agrupación y repetición de sprites, para así formar el suelo, las paredes y techo de la cueva.

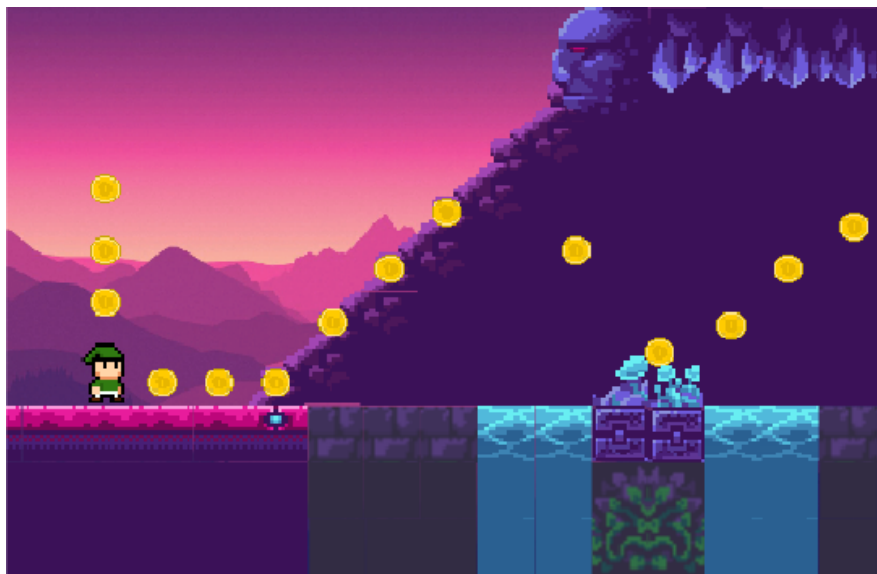


Figura 4.4: Comienzo del juego.

Sin embargo, como ya se explicará más adelante, el juego no es de extensión fija, si no que su duración varía acorde a un parámetro introducido en las opciones del mismo. Por tanto, resultaba lógico implementar un nivel infinito en el que se generaran partes de forma aleatoria.

Se trata de tres fragmentos diferentes, mostrados en las figuras 4.5, 4.6 y 4.7, con distintas plataformas que se suceden entre sí, siendo posible acceder de unas a otras por medio de un algoritmo. Este algoritmo instancia cada una de las plataformas, contenidas en una lista, de forma aleatoria, basándose en la distancia que calcula entre la posición del personaje y

una etiqueta colocada al final de cada una. Si esta distancia es menor que una constante declarada, significa que se encuentra relativamente cerca del final e instancia un nuevo nivel. De esta forma, conseguimos obtener un nivel infinito.

Las figuras 4.5, 4.6 y 4.7, se tratan de los diferentes niveles instanciados, a lo largo del bucle infinito, de forma totalmente aleatoria. En ellas se pueden observar las diferentes monedas que el avatar deberá recoger y los diferentes puntos de control explicados más adelante.

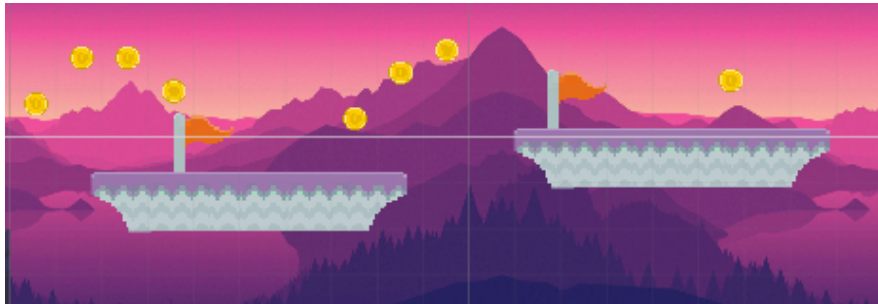


Figura 4.5: Nivel 1.

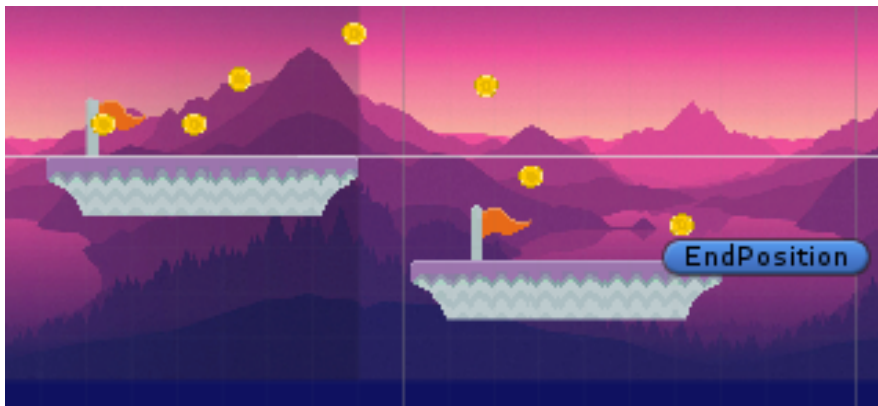


Figura 4.6: Nivel 2.

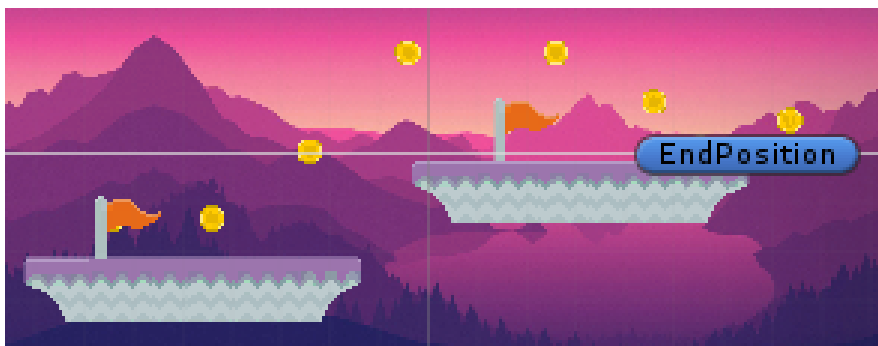


Figura 4.7: Nivel 3.

4.5 Componentes del Videojuego

4.5.1 Personaje

Es el eje principal de nuestro juego, la única parte que interactúa con todo. Visualmente, se trata de un sprite más. Sin embargo, el código que se le asocia, implementa todas las funcionalidades que tiene nuestro videojuego. Define gran cantidad de parámetros como se puede ver en la figura 4.8.

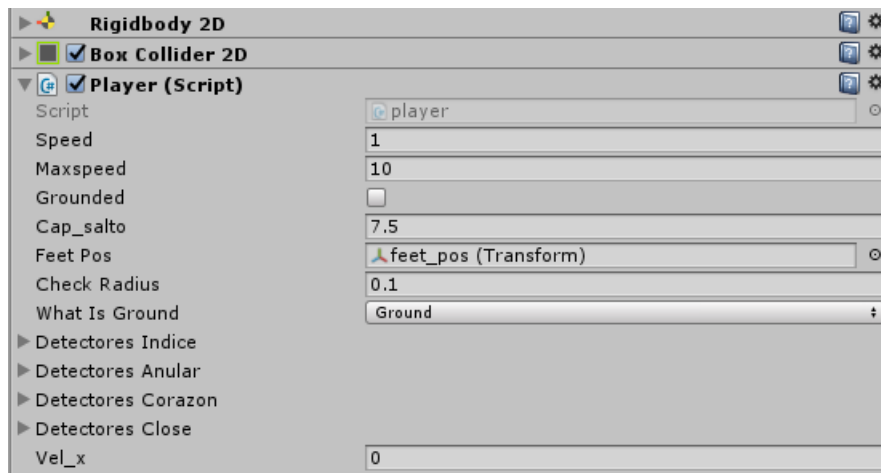


Figura 4.8: Parámetros asociados al personaje.

A continuación, se incluye una breve descripción de cada una de estas variables:

1. Speed. Es la velocidad horizontal inicial del personaje.
2. MaxSpeed. Es la velocidad máxima que puede adquirir el personaje.
3. Grounded. Variable booleana que se comprueba constantemente para saber si el personaje está en el aire.
4. Cap_salto. Determina el impulso que tomará el personaje en caso de tener que saltar.
5. Feetpos, checkRadius, whatsIsGround. Se trata de los elementos que determinan si un área circular definida por feetpos entra en contacto con un colisionador. En definitiva, gestionan las colisiones del personaje con el suelo.
6. Detectores. Se trata de las instancias de las clases de detección de los gestos.
7. Vel_x. Define el incremento de velocidad que se dará a lo largo del tiempo.

En este caso, el personaje, mostrado en la figura 4.9, tiene como objetivo conseguir el máximo número de monedas. Para ello, se le dota de una velocidad inicial horizontal, muy pequeña en un principio, que aumentará con el transcurso del tiempo. También debe ser capaz de interactuar con el dispositivo Leap Motion, pues si el jugador realiza los gestos implementados, que serán explicados más adelante, el personaje debe ser capaz de saltar.

Para ello, instancia objetos de las clases que implementan la detección de gestos y, según el resultado de los métodos de detección implementados, que indican si el gesto ha empezado o si se sigue realizando, se modifican distintos parámetros de la velocidad del personaje.



Figura 4.9: Sprite del personaje en reposo.

4.5.2 Animaciones

La animación es el proceso utilizado para dar la sensación de movimiento a imágenes, dibujos u otro tipo de objetos inanimados. En este proyecto se trata del conjunto de movimientos del personaje; implican 3 acciones en concreto:

- Reposo.
- Correr.
- Saltar.

Sin embargo las animaciones también tienen que definir la lógica que existe entre la transición de estas acciones para así mostrar una secuencia racional de las mismas.

4.5.2.1 Reposo

Conjunto de posturas sucesivas entre sí, que realiza el personaje cuando su velocidad es nula. Visualmente se trata de un movimiento vertical en el mismo lugar dando sensación de respiración y parpadeo, tal y como se aprecia en la figura 4.10.



Figura 4.10: Conjunto de posturas de reposo.

Para concatenar cada una de estas imágenes, Unity cuenta con la herramienta "Animation". Esta herramienta permite colocar las imágenes una detrás de otra en intervalos de tiempo muy pequeños, generando así esa secuencia siempre que se den las condiciones que se imponen en la herramienta que gestiona la lógica de las animaciones.

4.5.2.2 Correr

Correr es la animación más larga, como se muestra en la figura 4.11. Implica el diseño de las posturas intermedias que se dan a lo largo del movimiento. Primero, los dos pies en el suelo paralelos. Después uno de ellos se levanta y avanza. Mientras este no toca el suelo del todo el pie de atrás se alza, y así sucesivamente simulando la acción de correr.



Figura 4.11: Conjunto de posturas corriendo.

Normalmente las animaciones acaban y terminan con la misma postura. Esto es así para que el bucle de la animación al terminar no quede muy brusco ya que, si no, la última de ellas se ejecutaría en tan poco tiempo que ni siquiera sería capaz de apreciarse.

4.5.2.3 Saltar

Este es el caso más simple, como se muestra en la figura 4.12. Se trata únicamente de una sola imagen, que se mostrará cuando el personaje no esté en contacto con el suelo.



Figura 4.12: Postura de salto.

4.5.2.4 Transiciones entre animaciones

La gestión de animaciones y la manera de enlazarlas a lo que está sucediendo realmente en el juego es tan importante como el diseño de estas. Se trata de dar un sentido lógico a su uso.

Para ello, Unity cuenta con la herramienta "Animator", que gestiona las animaciones como estados y transiciones supeditadas a condiciones, según las variables que intervienen en el juego.

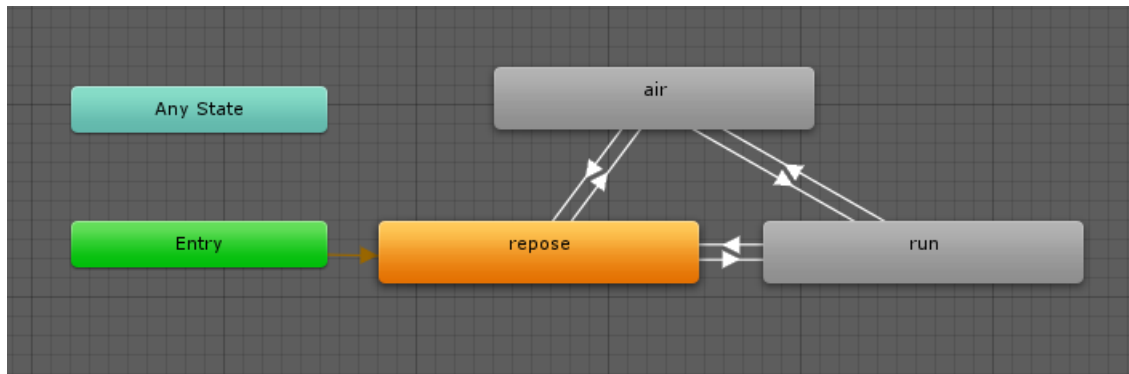


Figura 4.13: Estados y transiciones de animación.

La figura 4.13 muestra el conjunto de animaciones explicadas anteriormente, como los bloques de estado y las condiciones de transición para ir de una a otra.

A continuación se explica la secuencia lógica que seguirán las animaciones:

1. El personaje comenzará en reposo siempre.
2. En cada frame, comprueba sus dos posibles saltos a nuevas animaciones: "Air" que implica la animación de salto y "Run" que implicaría la de correr.
 - En caso de que el personaje no esté en contacto con un colisionador y, por tanto, la variable "grounded" sea falsa, se dará la animación de salto.
 - Por otro lado, en caso de que la variable "grounded" sea verdadera y la velocidad del personaje sea superior a 0, se dará el salto al estado "Run".
3. Si el personaje está en la animación de salto, volverá a reposo cuando esté en contacto con el suelo.
4. Si el personaje está corriendo, volverá a reposo cuando la velocidad sea 0 y esté en contacto con el suelo.
5. Si el personaje está corriendo pero se precisa de la animación saltar en movimiento, bastará con que en los siguientes frames se detecte que el personaje no está sobre el suelo.
6. Estando en la animación de salto, se precisa seguir con la animación de correr. Entonces se comprueba que el personaje ya se encuentre sobre el suelo y que la velocidad que lleva sea mayor a 0.

4.5.3 Colisionadores

Los colisionadores son áreas que engloban por su perímetro a aquellas plataformas o elementos del videojuego que pueden colisionar con el personaje.

A la hora de realizar las colisiones, Unity establece pequeñas diferencias entre dos tipos de colisionadores:

- Colliders: Definen una superficie en la que se detectarán colisiones. Son objetos sólidos y no pueden ser atravesados. En caso de que se quiera reaccionar a la colisión, Unity implementa un sistema de eventos que da soporte a este tipo de colisiones.
- Triggers: Al igual que los colliders, también definen una superficie. La única diferencia es que sí pueden ser atravesados por otros triggers o colliders.

Ambos dos se definen en Unity de igual forma, pero para aportarle la capacidad de trigger hay que marcar la casilla "IsTrigger", como se muestra en la figura 4.14.

En este proyecto se han usado ambos. Se han definido elementos que precisan de una colisión directa con consecuencias cinemáticas, y otros que, sin embargo, no generan ninguna respuesta física, como las monedas o los puntos de guardado, por lo que en estos casos cuando un colisionador entra en el espacio de otro colisionador, un disparador llamará a un evento de nombre "OnTriggerEnter2D", instanciado en el script correspondiente.

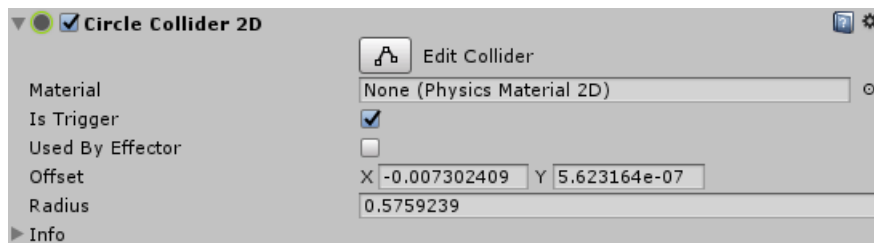


Figura 4.14: Colisionador de una moneda.

4.5.4 Puntos de guardado

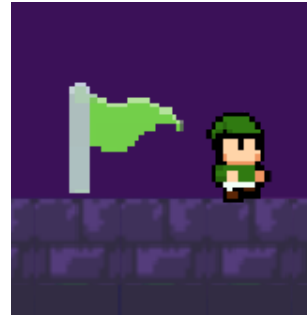
Todo videojuego tiene forma de poder guardar la partida que se está jugando, o bien en cualquier momento que el jugador desee o bien en determinados puntos de control que establece el programador. De esta manera, se iniciará de nuevo la partida sin necesidad de perder el progreso que ya se había hecho, y así comenzar por donde el jugador se había quedado en ese momento. Sin embargo, en este proyecto no tenía sentido establecer puntos de guardado manual, ya que se trata de un juego de una duración determinada para un proceso de rehabilitación, en la que no es interesante seguir una historia.

Por tanto, el enfoque seguido en este proyecto ha sido colocar puntos de control que, una vez alcanzados, guardan la posición actual del personaje. En caso de que el jugador no llegue al siguiente punto de control porque no realice bien el gesto que le permite acceder a la plataforma siguiente, no perderá todo su progreso, sino que reaparecerá en la ubicación del último punto de control que alcanzó.

La implementación de esta funcionalidad se basa en los eventos de colisión explicados anteriormente. Los puntos de control, mostrados en la figura 4.15, son triggers que lanzan un evento al tener una colisión. Este evento conserva la posición y en caso de que se detecte una caída, asigna al personaje dicha posición.



(a) Punto de control sin alcanzar.



(b) Punto de control alcanzado.

Figura 4.15: Puntos de control.

4.6 Gestos

En este apartado, se explicará la implementación de la detección de los gestos. Para ello, deberemos importar al entorno Unity, como mínimo, el primero de los módulos, llamado Core Assets. Este paquete permite la detección de la mano y proporciona cierta información sobre sus componentes. De este modo, se hará uso del cálculo de distancias entre los dedos en cada frame del videojuego y, en caso de que esas distancias superen cierto valor umbral, se habrá detectado uno de los gestos en cuestión, y el personaje actuará en consecuencia saltando.

Partiendo del paradigma de la programación modular, con el fin de hacer más manejable y legible el código implementado, la detección de gestos se ha implementado a partir de 3 clases distintas relacionadas entre sí por medio de herencia. Las clases implementadas son:

- Clase Detector.
- Clase AbstractHoldDetector.
- Clases de detección de gestos.

4.6.1 Clase Detector

En primer lugar y situada en lo más alto de esta jerarquía como clase base, se encuentra la clase Detector.

Un detector es un objeto que observa aspectos de una escena e informa de que son verdaderos cuando se cumplen las condiciones especificadas. En nuestro proyecto, estas condiciones involucrarán información con respecto a la mano. Esta clase podríamos decir que trata la detección del gesto de manera muy superficial, implementando los métodos genéricos que, combinados con métodos de las clases derivadas, permiten la detección. Su funcionalidad es realizar un seguimiento del estado de detección (verdadero o falso) y enviar eventos cuando ocurre un cambio.

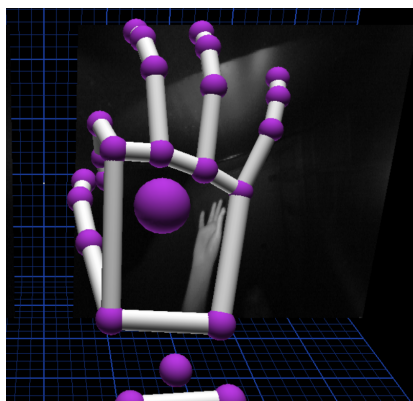
4.6.2 Clase AbstractHoldDetector

Se trata de una clase abstracta, derivada de la clase Detector. Implementa métodos virtuales que gestionan la detección de gestos de manera menos superficial que la clase de la que deriva, pero todavía se trata de métodos genéricos, ya que la personalización del gesto se ejecuta en un método todavía no definido que se halla en la clase derivada de esta.

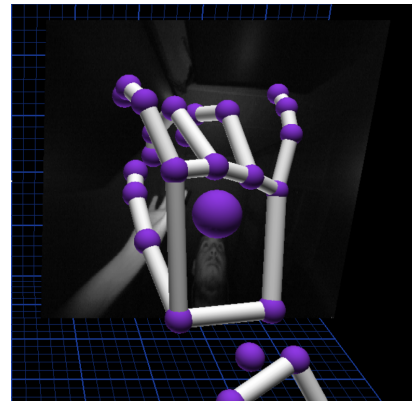
4.6.3 Clases de detección de gestos

Si bien las dos clases anteriores son genéricas y establecen la plantilla de funcionamiento para los detectores de gestos, esta última clase de la jerarquía es la que implementa la detección de cada gesto concreto, como se puede ver en la figura 4.16.

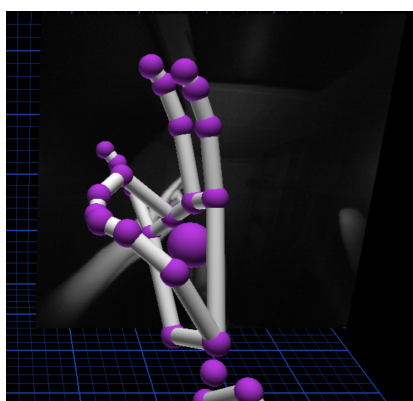
Se trata de las clases que acceden directamente al modelo de la mano y realizan cálculos de distancias, posiciones y rotaciones de distintos elementos de la misma.



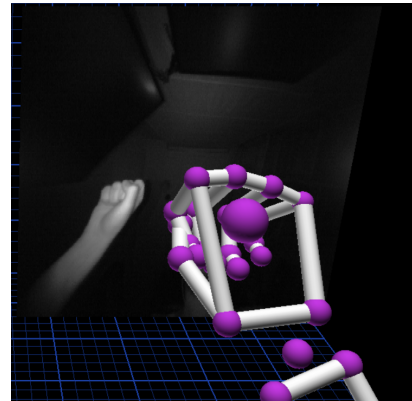
(a) Pellizco pulgar-índice.



(b) Pellizco pulgar-corazón.



(c) Pellizco pulgar-anular.



(d) Puño cerrado.

Figura 4.16: Gestos a detectar

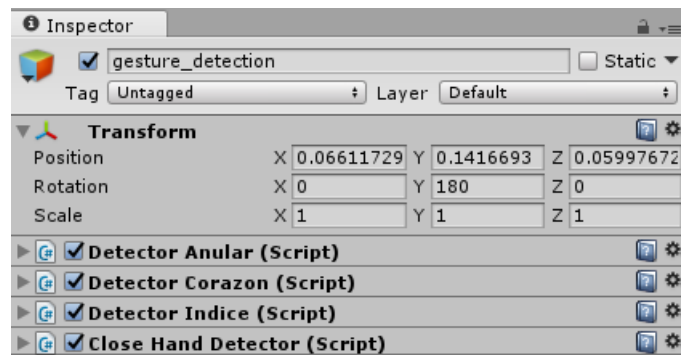


Figura 4.17: Scripts asociados al modelo de la mano.

4.7 Traza del videojuego

En este apartado se explican los pasos a seguir para poder jugar, desde su comienzo en el menú principal, seguido del menú de opciones, hasta la pantalla final donde se recoge la puntuación del jugador.

4.7.1 Menú principal

El menú principal, mostrado en la figura 4.18, es la primera escena que compone el videojuego. Se presentan ante el jugador 3 botones, y cada uno de ellos ejecuta una función diferente:

- Play: Ejecuta el administrador de escenas de Unity y da paso al juego.
- Options: Abre el menú de opciones.
- Quit: Sale de la aplicación.



Figura 4.18: Menú principal.

4.7.2 Menú de opciones

El botón que permite el acceso a este menú tiene asignado una función que activa o desactiva cualquier elemento del juego (GameObject) que se le asocie. Por tanto, mientras no se presiona, muestra el menú principal, y al presionarlo activa la transición al menú de opciones tal y como se aprecia en la figura 4.19.

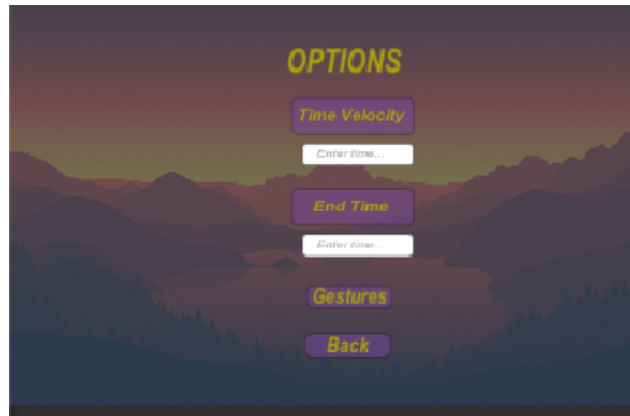


Figura 4.19: Menú de opciones.

Este menú permite adaptar la terapia a las necesidades del paciente por medio de la modificación de diferentes parámetros del juego, controlados por el equipo médico encargado de la supervisión de la rehabilitación. Para ello, cuenta con las siguientes opciones:

- Time Velocity. Esta variable representa el intervalo de tiempo en el que aumentará la velocidad horizontal del personaje, siempre sin superar la velocidad máxima definida. El personaje comienza sin apenas velocidad y, cada X tiempo, la velocidad aumenta para intensificar la dificultad.
- End Time: Este parámetro define el tiempo de duración de la terapia. Para ello, se ha implementado un cronómetro en la escena principal del juego. Tras cumplirse el tiempo se dará por finalizado el juego, y aparecerá la puntuación obtenida por el paciente.
- Gestures: Abre el menú de gestos.
- Back: Vuelve al menú principal.

4.7.3 Menú de gestión de gestos

Al igual que sucede con el botón "Options" del menú principal, el botón "Gestures" tiene asignado, al pulsarlo, la activación del menú gestos, como se muestra en la figura 4.20.



Figura 4.20: Menu de gestión de gestos.

Este menú permite a los especialistas seleccionar la actividad que tendrá que llevar a cabo el paciente de manera regular a lo largo de la terapia. Con un solo click serán capaces de escoger el gesto que tendrá que hacer el paciente para realizar el salto del personaje. Las opciones implementadas son:

1. Thumb-Index: Establece el contacto entre el dedo pulgar e índice como entrada de control al juego.
2. Thumb-Middle: Establece el contacto entre el dedo pulgar y corazón como entrada de control al juego.
3. Thumb-Ring: Establece el contacto entre el dedo pulgar y anular como entrada de control al juego.
4. Close Hand: Establece el cierre de la mano como entrada de control al juego.
5. Back: Vuelve al menú de opciones.

4.7.4 Escenario principal del juego

Una vez se ha personalizado la terapia para el paciente, ya estaría todo listo para empezar a jugar. Para ello, se pulsa el botón Play, y el flujo del programa accede al escenario principal del juego, como se muestra en la figura 4.21. En esta figura únicamente se muestra una captura parcial de la escena ya que, tras la última plataforma, el juego genera una repetición de las distintas plataformas que aparecen en las figuras 4.6 y 4.7.

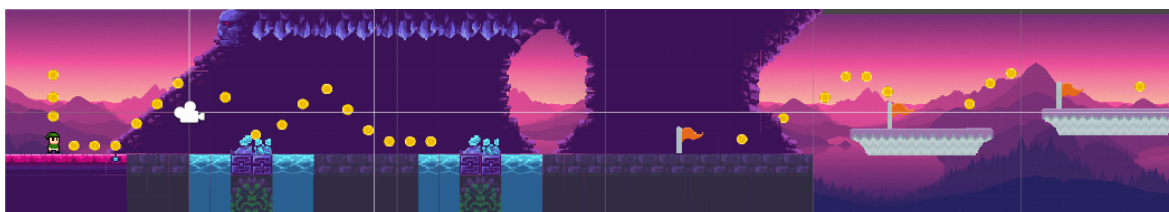


Figura 4.21: Escena principal.

La jugabilidad es bien sencilla. El personaje se desplaza horizontalmente de manera automática y el paciente se encarga de ejecutar la acción de salto. El objetivo del juego es conseguir el máximo número de monedas saltando entre las distintas plataformas que aparecerán a su paso.

4.7.5 Puntuación Final

Tras superar el tiempo de juego, introducido en el menú de opciones, el juego termina y aparece la escena final, mostrada en la figura 4.22, que presenta la puntuación final junto con dos botones:

- **Retry:** Permite volver a jugar sin la posibilidad de cambiar las variables que se habían personalizado durante el primer intento.
- **Quit:** Cierra la Aplicación.

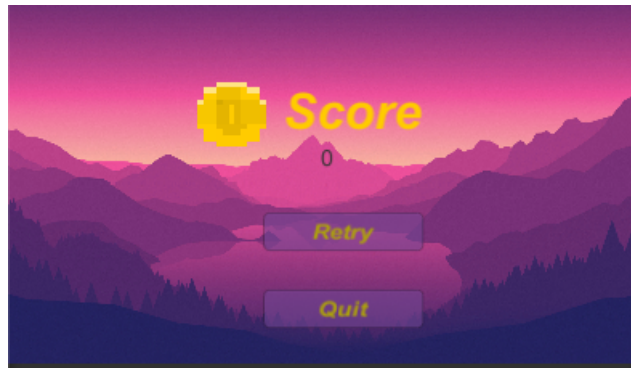


Figura 4.22: Puntuación Final.

4.8 Resultado

La integración de todos los conceptos de rehabilitación tratados anteriormente junto con el inicial desarrollo de la figura de la gamificación han sido las bases fundamentales de este proyecto. Por lo que una vez explicados todos los componentes del videojuego que en conjunto intentan dar lugar a una experiencia rehabilitadora, y la secuencia lógica que se sigue para su experimentación, solo falta mostrar el resultado obtenido:

<https://www.youtube.com/watch?v=xipYUNgFq3E&t=23s>

5 Conclusiones

A lo largo de esta memoria se ha presentado la situación de las personas con Daño cerebral adquirido con respecto a la rehabilitación tradicional, y lo innovador y beneficioso que puede resultar el uso de tecnologías en la terapia. La propuesta desarrollada está en consonancia con estas nuevas terapias innovadoras y, a la vista de los resultados, podemos afirmar que se han cumplido los objetivos planteados al inicio del proyecto.

En primer lugar y para situar un punto común de partida, se ha realizado una reseña del contexto histórico y actual de los accidentes cerebrovasculares. Así pues, hemos introducido el ámbito clínico de la rehabilitación, y más concretamente el de la rehabilitación de miembro superior tras ictus.

Para entender esta área de la medicina, hemos introducido los principios del aprendizaje motor y su relación con la neuroplasticidad, ya que ambos conceptos son la base para una rehabilitación satisfactoria.

Dados los problemas que acarrea una terapia al uso debido a la carga psicológica que conlleva sobre el paciente, en este proyecto hemos procurado aportar un enfoque vanguardista en este ámbito, y hemos introducido el concepto de gamificación al ámbito de la rehabilitación.

Toda la propuesta ha focalizado su interés en incluir de forma intrínseca alguno de los motores que motivan al paciente a generar una conducta determinada, de acuerdo con el sistema Octalysis. En este caso, consideramos que hemos conseguido incluir los siguientes:

- Significado y vocación épicas: El paciente forma parte de una comunidad trascendente a él mismo y se siente integrado.
- Desarrollo y logro: La persona observa su propio rendimiento a partir de la puntuación final. La motivación nace como consecuencia de querer superarse a sí mismo.
- Potenciar la creatividad y la retroalimentación: El juego por medio de sus opciones se ajusta a las necesidades del paciente.
- Pérdida: El hecho de jugar crea un hábito saludable para el sujeto, por lo que la posible pérdida del progreso que conlleva este hábito, motiva y hace continua su práctica.

Por tanto, de alguna manera hemos tratado de ayudar al paciente a mantener su motivación durante el tratamiento de rehabilitación, uno de los elementos más importantes para evitar el abandono y, por tanto, favorecer la adherencia del paciente a la terapia.

Como líneas futuras de investigación, se podría aumentar el conjunto de gestos que se pueden detectar para servir como entrada al sistema. De esta manera, se podría aplicar esta

rehabilitación a pacientes que sufran otro tipo de afecciones y que no puedan realizar los gestos que hemos definido.

Por otra parte, consideramos extender la jugabilidad para evitar la monotonía de estar siempre en la misma escena principal sin variar el modo de juego. Es decir, ahora mismo el desarrollo de la propuesta es totalmente lineal, una única pantalla de juego donde el personaje se traslada a lo largo de la escena. Como línea futura sería el hecho de aportar nuevas pantallas con diferente diseño y dificultad.

Por último, una muy buena línea de investigación para enriquecer este proyecto sería obtener el grado de precisión con el que el paciente realiza el gesto. De esta forma, compararíamos el gesto modelado con el gesto realizado por el paciente y obtendríamos una medida de similitud, que nos serviría para evaluar cómo de bien se ha realizado.

Bibliografía

- Api overview.* (2014, May). Descargado 26/5/2020, de https://developer-archive.leapmotion.com/documentation/v2/csharp/devguide/Leap_Overview.html
- A brave new augmented reality awaits.* (2018, May). Descargado 21/5/2020, de <https://www.electronicdesign.com/industrial-automation/article/21806766/a-brave-new-augmented-reality-awaits/>
- Chou, Y.-k. (2019). *Actionable gamification: Beyond points, badges, and leaderboards*. Packt Publishing Ltd.
- Cogollor, J. M., Rojo-Lacal, J., Hermsdörfer, J., Ferre, M., Waldmeyer, M. T. A., Giachritsis, C., ... Sebastián, J. M. (2018). Evolution of cognitive rehabilitation after stroke from traditional techniques to smart and personalized home-based information and communication technology systems: literature review. *JMIR rehabilitation and assistive technologies*, 5(1), e4.
- Conselleria, S. (2017, May). *Estrategia para la atención al daño cerebral adquirido en la comunitat valenciana*. Descargado 17/5/2020, de http://www.san.gva.es/documents/156344/6977187/DCA_castellano_imprimible_dos_caras.pdf
- de la Fuente, M. J. L. (2013). Teorías del control motor, principios de aprendizaje motor y concepto bobath. a propósito de un caso en terapia ocupacional. *Revista electrónica de terapia ocupacional Galicia, TOG*(18), 13.
- Ejercicios de rehabilitación, flint rehab.* (2017, May). Descargado 18/5/2020, de <https://www.flintrehab.com/es/>
- Ernst, E. (1990). A review of stroke rehabilitation and physiotherapy. *Stroke*, 21(7), 1081–1085.
- Freitas, S. A. A., Lacerda, A. R., Calado, P. M., Lima, T. S., y Canedo, E. D. (2017). Gamification in education: A methodology to identify student's profile. En *2017 ieee frontiers in education conference (fie)* (pp. 1–8).
- Gamification in healthcare: the value of fun.* (2020, May). Descargado 20/5/2020, de <https://belitsoft.com/custom-application-development-services/healthcare-software-development/gamification-in-healthcare-just-for-fun-or-has-real-value>
- García, E., Soto. (2016, May). *Rehabilitación tras un ictus*. Descargado 18/5/2020, de <https://serpacienteexpertoactivo.files.wordpress.com/2016/07/rehabiliatacion-tras-ictus.pdf>
- Glooko.* (2020, May). Descargado 20/5/2020, de <https://www-int.glooko.com/>

- González, O. C. (2006). Terapia de restricción-inducción de movimiento (cimt), una técnica de utilidad para terapeutas ocupacionales que trabajan en neurorehabilitación. *Revista Chilena de Terapia Ocupacional*(6), ág-9.
- Hollow knight*. (2017, May). Descargado 22/5/2020, de <https://unity.com/es/madewith/hollow-knight>
- Leap motion controller*. (2012a, May). Descargado 22/5/2020, de <https://www.ultraleap.com/product/leap-motion-controller/>
- Leap motion controller*. (2012b, May). Descargado 24/5/2020, de <https://www.ultraleap.com/product/leap-motion-controller/#whatsincluded>
- Mindmize*. (2020, May). Descargado 20/5/2020, de <https://www.mindmaze.com/>
- Mindmotion go increases therapy engagement and adherence through vr games*. (2017, December). Descargado 26/5/2020, de <https://fitnessgaming.com/news/health-and-rehab/mindmotion-go-increases-therapy-engagement-and-adherence-through-vr-games.html>
- Mubin, O., Alnajjar, F., Jishtu, N., Alsinglawi, B., y Al Mahmud, A. (2019). Exoskeletons with virtual reality, augmented reality, and gamification for stroke patients' rehabilitation: Systematic review. *JMIR rehabilitation and assistive technologies*, 6(2), e12010.
- Newport, R. (2006). Ventajas de la rehabilitación asistida mediante robot en la recuperación de las funciones motriz y visuoespacial en pacientes en fase de recuperación de un accidente cerebrovascular. *Revista Española de Geriatria y Gerontología*, 41, 66-73.
- NINDS. (2017, May). *Accidente cerebrovascular. esperanza en la investigacion*. Descargado 17/5/2020, de https://espanol.ninds.nih.gov/trastornos/accidente_cerebrovascular.htm
- Nintendo wii*. (2006, May). Descargado 21/5/2020, de <https://www.xbox.com/es-ES/consoles/xbox-one>
- O'Connor, T. J., Fitzgerald, S. G., Cooper, R. A., Thorman, T. A., y Boninger, M. L. (2002). Kinetic and physiological analysis of the game (wheels) system. *J Rehabil Res Dev*, 39(6), 627-634.
- Rebolledo, F. A. (2003). Neuronal plasticity. part 1. *Revista Médica del Instituto Mexicano del Seguro Social*, 41(1), 55-64.
- Úbeda, A. (2018). Presentaciones teoría-robótica de servicios. *Robótica de Servicios*.
- Unity assets for leap motion orion beta*. (2018, May). Descargado 26/5/2020, de <https://developer.leapmotion.com/unity#5436356>
- Verizon's 5g network could make virtual rehabilitation for patients a reality*. (2018, May). Descargado 21/5/2020, de <https://tbirehabilitation.wordpress.com/2018/06/23/web-site-verizons-5g-network-could-make-virtual-rehabilitation-for-patients-a-reality/>
-

Xbox360. (2005, May). Descargado 21/5/2020, de <https://www.xbox.com/es-ES/games/xbox-360>

Yu-kai chou sistema octalysis. (2020, May). Descargado 20/5/2020, de <https://www.picuki.com/media/2255814240984260992>

Lista de Acrónimos y Abreviaturas

| | |
|-------------|---|
| ACV | Accidente cerebrovascular. |
| AR | Realidad Aumentada. |
| CIMT | Terapia de restricción-inducción de movimiento. |
| DCA | Daño cerebral adquirido. |
| HMD | Head mounted displays. |
| UCI | Unidad de cuidados intensivos. |
| VR | Realidad Virtual. |