



FACULTAD DE INGENIERIA Y CIENCIAS
APLICADAS

ADAPTACIÓN DE EJERCICIOS TERAPÉUTICOS EN LA FORMA DE JUEGOS
SERIOS PARA PACIENTES DE CIRUGÍA DE REEMPLAZO DE CADERA

Trabajo de Titulación presentado en conformidad con los requisitos establecidos
para optar por el título de Ingeniero en Sistemas de Computación e Informática

Profesor guía

PhD. Yves Philippe Rybarczyk

Autor

Patricio David Espinosa Alvarez

AÑO

2019

DECLARACIÓN DEL PROFESOR GUÍA

"Declaro haber dirigido el trabajo, adaptación de ejercicios terapéuticos en la forma de juegos serios para pacientes de cirugía de reemplazo de cadera, a través de reuniones periódicas con el estudiante Patricio David Espinosa Alvarez en el semestre 201920 orientando sus conocimientos y competencias para un eficiente desarrollo del tema escogido y dando cumplimiento a todas las disposiciones vigentes que regulan los Trabajos de Titulación".

Yves Philippe Rybarczyk

Doctor en Robótica

C.C. 175695097-6

DECLARACIÓN DEL PROFESOR CORRECTOR

"Declaro haber revisado este trabajo, adaptación de ejercicios terapéuticos en la forma de juegos serios para pacientes de cirugía de reemplazo de cadera, del estudiante Patricio David Espinosa Alvarez, en el semestre 201920, dando cumplimiento a todas las disposiciones vigentes que regulan los Trabajos de Titulación".

Jorge Luis Pérez Medina

Doctor en Informática

C.C. 175899312-3

DECLARACIÓN DE AUTORÍA DEL ESTUDIANTE

“Declaro que este trabajo es original, de mi autoría, que se han citado las fuentes correspondientes y que en su ejecución se respetaron las disposiciones legales que protegen los derechos de autor vigentes.”

Patricio David Espinosa Alvarez

C.C. 175342687-1

RESUMEN

El presente proyecto de titulación tiene como objetivo el desarrollo de un juego serio como soporte a la ejecución de ejercicios terapéuticos para pacientes de cirugía de remplazo de cadera utilizando el Motor gráfico Unity 3D. El resultado de este trabajo es facilitar el proceso de rehabilitación de los pacientes por medio de un ambiente didáctico y entretenido, el cual se estima permitirá que los ejercicios se hagan de manera constante y sigan el plan propuesto por el fisioterapeuta.

Se desarrollaron cuatro ambientes gamificados para los ejercicios de abducción de cadera, flexión lenta de cadera y rodilla, secuencia paso adelante, paso lateral y paso atrás, y extensión de cadera. Siendo cada ambiente un entorno diferente con un objetivo en particular como, patear una boya, patear una pelota, realizar pasos de baile e inflar un globo mediante una bomba.

Los pacientes realizarán los ejercicios utilizando un dispositivo de detección de movimiento que obtendrá los puntos de flexión de cada articulación de su cuerpo y los enviará mediante un protocolo UDP al servidor donde se estará ejecutando los ambientes. El ambiente lúdico consta de un avatar humanoide que imitará los movimientos del paciente utilizando los datos recibidos.

Para la experimentación del proyecto se realizaron pruebas del juego a diferentes personas, que realizaron el ejercicio de abducción de cadera en el ambiente desarrollado en Unity 3D. Posteriormente se les indicó realizar la misma actividad en un ambiente realizado en JavaScript.

Los resultados se estimaron por medio de encuestas realizadas a las personas que realizaron los ejercicios en ambos ambientes. Se logró comparar si el ambiente gamificado facilita el proceso de rehabilitación. El experimento presenta resultados satisfactorios en el uso del juego serio desarrollado.

ABSTRACT

The objective of this titulation project is the development of a serious game that supports the therapeutic exercises for hip replacement patients, using the Unity 3D Graphic Engine. The result of this project is to help in the rehabilitation process of patients in a didactic and entertaining environment, which is meant to reinforce the constancy of the proposed physiotherapist plan.

There were four gamification environments developed, which are hip abduction exercise, slow hip and knee flexion, forward step-back step-lateral step sequence and hip extension. Each environment has a particular objective like, kicking a buoy, kicking a ball, perform dancing steps and using a step pump to blow a balloon.

The patients will execute the exercises using a motion detection device which will capture the flexion points of each body articulation and send them via UDP protocol to the server where each of the environments mentioned before will be running. The playing environment displayed on screen consist of a humanoid avatar that mimics the patients movement using the received data.

For the project experiment, different subjects tested the game performing the hip abduction exercise in the environment developed in the Unity 3D engine. Then the subjects tested the same exercise in the environment developed in JavaScript.

There was a survey conducted to the subjects that tested both of the game environments, the one developed in Unity 3D and the one developed in JavaScript. The results showed if the gamificated environment helps in the rehabilitation process of hip replacement patients. The experiment presents satisfactory results with the use of the serious game.

ÍNDICE

1. INTRODUCCIÓN.....	1
1.1. Justificación del Trabajo.....	2
1.2. Propuesta de la solución.....	3
1.3. Objetivos	3
1.3.1. Objetivo General del Trabajo.....	3
1.3.2. Objetivos Específicos del Trabajo	3
2. MARCO TEÓRICO.....	4
2.1. ePHoRt.....	5
2.2. Ambiente terapéutico	5
2.3. Juegos serios.....	8
2.4. Herramientas de modelado 3D.....	9
3. DESARROLLO DE LA SOLUCIÓN.....	9
3.1. Unity 3D	10
3.2. Arquitectura	11
3.3. Diseño y mapeo de huesos y articulaciones del Avatar	12
3.4. Diseño de ambientes didácticos.....	16
3.4.1. Dojo de karate.....	17
3.4.2. Campo de fútbol.....	20
3.4.3. Salón de baile	22
3.4.4. Interacción con un globo	24
3.5. Adaptación de ejercicios terapéuticos	26
3.5.1. Abducción de cadera.....	26
3.5.2. Flexión lenta de cadera y rodilla.....	27
3.5.3. Secuencia paso adelante, paso lateral y paso atrás	28
3.5.4. Extensión de cadera.....	29
3.6. Transformación de datos UDP en puntos de movimiento.....	31
3.6.1. Desarrollo de función de obtención de datos	31
3.6.2. Obtención de datos entre el hilo principal y el hilo secundario.....	33
3.6.3. Transformación de Datos skeleton en puntos de movimiento	33
3.6.4. Integración con la aplicación ePHoRt.....	39
3.7. Filtrado de datos	39

4. EXPERIMENTO	40
4.1. Participantes	40
4.2. Dispositivos	40
4.3. Tareas	41
4.4. Procedimiento.....	41
5. RESULTADOS	41
5.1. Discusión	42
6. CONCLUSIONES Y RECOMENDACIONES	44
6.1. Conclusiones	44
6.2. Recomendaciones.....	44
REFERENCIAS	46

ÍNDICE DE FIGURAS

Figura 1. Sistema ePHoRT.	5
Figura 2. Abducción de cadera	6
Figura 3. Flexión lenta de cadera y rodilla.....	7
Figura 4. Extensión de cadera	7
Figura 5. Diagrama de arquitectura.....	11
Figura 6. Interfaz principal de una escena	12
Figura 7. Personaje femenino	13
Figura 8. Personaje masculino.....	13
Figura 9. Propiedades de la animación de los personajes	14
Figura 10. Ventana de Inspección.....	15
Figura 11. Objeto en pose T con sus huesos.....	16
Figura 12. Interfaz de mapeo manual de esqueleto a un objeto	16
Figura 13. Distribución de escenas	17
Figura 14. Panel con el material Wooden Floor 3	18
Figura 15. Sala de Dojo.....	18
Figura 16. Modelo de letrero de escritura.....	19
Figura 17. Modelo de bandera japonesa.....	20
Figura 18. Ambiente Dojo de Karate terminado	20
Figura 19. Modelo de cancha de fútbol	21
Figura 20. Comparación de tamaño entre objeto original y su modificación	21
Figura 21. Campo de futbol modificado.....	22
Figura 22. Ambiente de futbol terminado visto desde la cámara.....	22
Figura 23. Salón de baile	23
Figura 24. Ambiente Salón de baile terminado	24
Figura 25. Habitación para el globo.....	24
Figura 26. Modelo de globo con materiales	25
Figura 27. Ambiente de Globo aerostática terminado	25
Figura 28. Objeto boya.....	26
Figura 29. Curva de la animación Kick.....	26
Figura 30. Ambiente Dojo de karate en ejecución.....	27
Figura 31. Ambiente Campo de fútbol en ejecución.....	27
Figura 32. Ambiente Sala de baile en ejecución	28

Figura 33. Ambiente Interacción con un globo en ejecución	29
Figura 34. Objeto balanza.	30
Figura 35. Curva de animación BackKick.....	30
Figura 36. Curva de animación FrontalKick	30
Figura 37. Diagrama de obtención de datos mediante el protocolo UDP. Representación del hilo secundario	32
Figura 38. Representación de los joints del cuerpo humano.....	35
Figura 39. Orientation Quaternion.....	37
Figura 40. Diagrama de flujo de la aplicación	38
Figura 41 .Diagrama de resultados de la encuesta.....	42

1. INTRODUCCIÓN

A lo largo de los años se ha evidenciado el incremento de fracasos en la rehabilitación terapéutica de pacientes posteriores a cirugías. El limitado número de sesiones de ejercicios terapéuticos realizados, el costo elevado de las sesiones, la falta de doctores especializados en recuperación de su cirugía o el traslado de sus hogares a hospitales o centros médicos son algunas de las causas de los fracasos.

Debido a estas dificultades, en los últimos años se ha incrementado considerablemente el desarrollo e investigación de sistemas de tele-rehabilitación (Goldstein & Beers, 1998), los cuales buscan permitir a los pacientes realizar sus terapias desde la comodidad del hogar. Tecnologías que para ser aprobadas deben ser eficientes en todos los aspectos, para esto, deben ser confiables y altamente accesibles reduciendo el costo habitual de inversión de los pacientes. Por otra parte, debe existir un monitoreo constante de la actividad realizada, lo que evidenciaría el progreso y la correcta ejecución de los protocolos establecidos por el terapeuta. En este contexto ePHoRt, es una plataforma web innovadora que pretende dar soluciones eficaces a bajo costo para el desarrollo de actividades de tele-rehabilitación en pacientes que han sufrido una sustitución de cadera (Rybarczyk, 2019).

Los avances continuos de los dispositivos de reconocimientos de gestos han permitido la creación de sistemas médicos que involucran juegos serios. Sistemas creados con el principal objetivo de la recuperación o monitoreo de pacientes con distintos problemas de salud; los cuales además de ayudar a mejorar su estado físico, ofrece entretenimiento con diferentes actividades recreativas que mantienen el interés del individuo y lo impulsan a continuar regularmente con el plan de ejercicios dictado por el fisioterapeuta a cargo de su proceso de recuperación. ePHoRT incorpora dentro de su solución un juego serio con características limitadas, por lo que se requiere hacer estudios de manera de incorporar un mecanismo de creación de juegos más versátiles.

1.1. Justificación del Trabajo

La mayor parte de las personas después de una cirugía de reemplazo de cadera no tienden a recuperarse por completo dado que en mucho de los casos no finalizan un programa terapéutico. Un programa terapéutico está diseñado por un terapeuta y es un conjunto de ejercicios adaptado a las necesidades del paciente; el cual se ejecuta en varias sesiones de ejercicios agrupados en etapas durante un periodo aproximado de tres semanas (Murphy & Karlin, 2016).

Los juegos serios pueden apoyar a que los pacientes puedan realizar su actividad de recuperación de manera menos traumática apoyándose en ambientes interactivos capaces de hacer sentir a los pacientes cómodos y entretenidos (Mccallum, 2015). Una de las herramientas que ha apoyado a los juegos serios es Unity 3D, siendo un motor gráfico de software libre completo (Anticevic & Anticevic, 2014). Gracias a su accesibilidad se puede encontrar varios artículos de implementaciones de juegos serios con Unity.

Este proyecto busca ser un apoyo complementario para los pacientes durante la ejecución de un programa terapéutico con juegos serios. Los pacientes podrán ejecutar su set de ejercicios de rehabilitación de forma interactiva y entretenida desde sus hogares; lo que resolverá parte de los inconvenientes de traslado que tienen al movilizarse desde sus hogares a los hospitales y centros médicos. De esta forma, en un futuro con el juego integrado a un sistema el fisioterapeuta podrá monitorear las sesiones de ejercicio sin necesidad de tener la presencia física del paciente.

A través de un sistema de adquisición de los movimientos, por ejemplo, una cámara Kinect con acceso a internet, los pacientes podrán empezar a realizar las sesiones de ejercicio a través de un juego. Los datos de movimiento capturados pueden enviarse a un juego serio a través de un protocolo, como por ejemplo UDP. El juego serio procesará esa información y mantendrá al usuario en un ambiente amigable con el objeto de no aburrir ni hacer monótonas las sesiones de ejercicio. Este ambiente incentivará a los pacientes para completar su plan de ejercicios de forma efectiva para su recuperación (Mccallum, 2015).

Además, como se mencionó previamente los fisioterapeutas no necesitan estar

presentes cuando los pacientes realicen los ejercicios, ya que el juego podrá recopilar datos del paciente durante la ejecución de los ejercicios. Consecuentemente, en un futuro, la información recopilada podrá ser procesada y visualizada a través de un sistema donde tanto pacientes como fisioterapeutas podrán observar la ejecución del programa terapéutico.

1.2. Propuesta de la solución

Desarrollar un juego serio para acompañar al paciente durante la ejecución de ejercicios terapéuticos definidos por el fisioterapeuta luego de una cirugía de remplazo de cadera. El juego tendrá un ambiente didáctico e interactivo que motivan a los pacientes a comprometerse con su proceso de rehabilitación.

El juego utilizará gamificación para que el paciente obtenga diferentes bonificaciones al realizar correctamente los ejercicios. De esta manera, cambiar el ámbito rutinario y aburrido del programa terapéutico a un ambiente más dinámico.

1.3. Objetivos

1.3.1. Objetivo General del Trabajo

Desarrollar un juego serio, para la ejecución de ejercicios terapéuticos, en el motor gráfico Unity 3D que capte los movimientos a través de una cámara Kinect.

1.3.2. Objetivos Específicos del Trabajo

- a) Recibir un grupo de puntos de articulación mediante UDP y transformarlos puntos de movimiento.
- b) Diseñar un avatar amigable y mapear los puntos de movimiento a sus huesos y articulaciones.
- c) Realizar un juego con un ambiente didáctico y entretenido en la herramienta Unity 3D.
- d) Adaptar cuatro ejercicios terapéuticos (abducción de cadera, flexión lenta de cadera y rodilla, secuencia paso adelante, paso lateral y paso atrás, y extensión de cadera) en la forma de juegos serios.

2. MARCO TEÓRICO

2.1. ePHoRT

ePHoRt es una plataforma creada con el propósito de ayudar a pacientes de cirugía de remplazo de cadera una forma de tele rehabilitación captando cada movimiento que realiza mediante un grupo de sensores. Tras la recolección de estos datos, el sistema evalúa si se realizaron de manera correcta y notificar tanto al paciente como al terapeuta después de cada sesión.

El sistema tiene tres componentes: el primero es un componente enfocado a la educación terapéutica en donde se encuentra un conjunto de recursos de aprendizaje. El segundo componente es un módulo de adquisición de los movimientos del paciente utilizando un dispositivo de tipo Kinect.

El tercer componente es un API RESTFUL que recibe los datos de movimiento, los procesa y da un resultado con ayuda de algoritmos de inteligencia artificial. El cuarto componente contiene un avatar sencillo de puntos que emula los movimientos del paciente en tiempo real con los datos proporcionados del sensor de la Kinect (Rybarczyk, 2019).

ePHoRt facilita a pacientes y fisioterapeutas interactuar y ejecutar los programas terapéuticos planificados (Rybarczyk, 2019). El fisioterapeuta solicita una reunión física con el paciente inicial para realizar una evaluación y planificar el programa. Luego, el paciente puede realizar los ejercicios propuestos desde su casa con ayuda del sistema. El paciente puede suspender el programa en cualquier momento en caso de que tengan alguna dificultad y entonces el fisioterapeuta recibirá una notificación y podrá realizar una reunión física para evaluar nuevamente al paciente.

De esta forma, el fisioterapeuta acompaña al paciente junto al sistema durante el proceso de rehabilitación. Los terapeutas también podrán observar los datos o resultados de sus pacientes, para ver su desempeño y poder modificar los ejercicios requeridos.

Para que los pacientes ejecuten los ejercicios en un ambiente gamificado, ePHoRt tiene un módulo de adquisición de datos en donde a través del sensor de la Kinect

recupera puntos en tiempo real y son enviados a Unity 3D a través del protocolo UDP (Segall, 2004). UDP es un protocolo de datagramas de usuario que permite enviar datagramas a través de internet entre dos dispositivos que hayan establecido una conexión (HALSALL, 1998).

ePHoRt utiliza un avatar sencillo de puntos en un ambiente de una cancha de futbol para acompañar al paciente durante la ejecución de sus ejercicios terapéuticos; como se ilustra en la Figura 1. El ambiente emula una cancha de futbol que se mantiene durante la ejecución de los ejercicios y el avatar no refleja la forma de un ser humano por lo que no es amigable al usuario; sin embargo, permite la evaluación de los ejercicios.

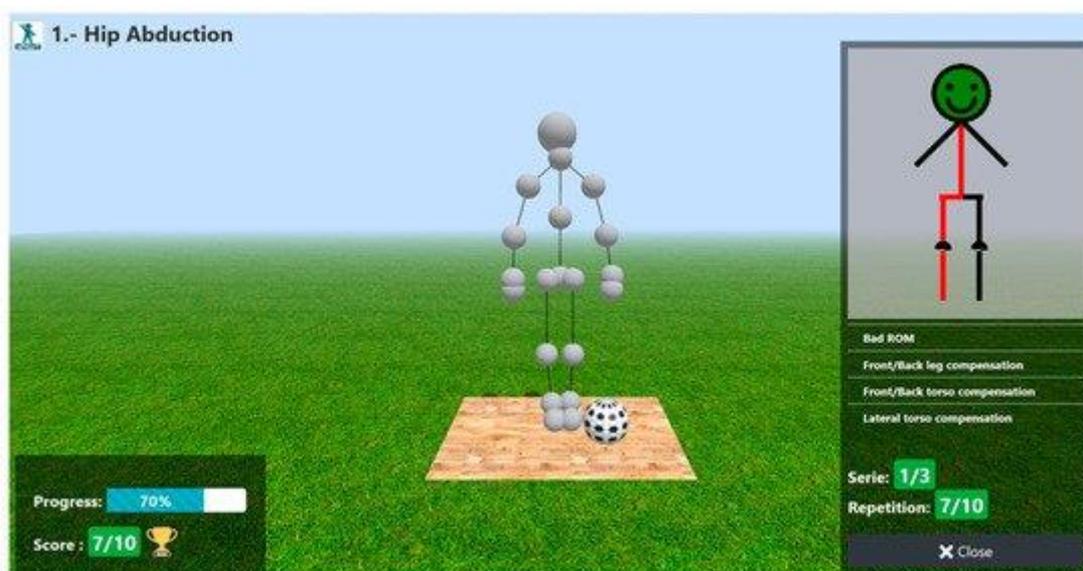


Figura 1. Sistema ePHoRT.

Adaptado de Perez Medina, 2019

2.2. Ambiente terapéutico

La rehabilitación del remplazo de cadera necesita que el paciente realice un conjunto de ejercicios continuos previamente planificados por el fisioterapeuta. Esto dado que el proceso de recuperación se ha considerado como uno de los más largos en el mundo de la medicina, ya que esta necesita de varias series de ejercicios diferentes para que el movimiento sea recuperado totalmente y el dolor sea disminuido al mínimo (Murphy & Karlin, 2016).

Los ejercicios de rehabilitación, luego de una operación de remplazo de cadera se basan principalmente en realizar movimientos en las extremidades inferiores, principalmente en piernas y rodillas. Entre los más conocidos se encuentra la abducción de cadera y la flexión de rodilla y cadera (Murphy & Karlin, 2016).

Los ejercicios también se pueden usar para conocimiento y seguimiento del estado del sistema nervioso de una persona. Por lo que su uso puede mejorar el funcionamiento total del sistema nervioso (Swartz, 2015).

2.2.1. Abducción de cadera

Este ejercicio consiste en mover una pierna lateralmente 30 grados, este nos permite generar movimientos en los que participa la pelvis en su totalidad, eso incluye los dos hemisferios laterales del hueso, un ejemplo del ejercicio se lo puede apreciar en la Figura 2 (Bernabé, 2018).

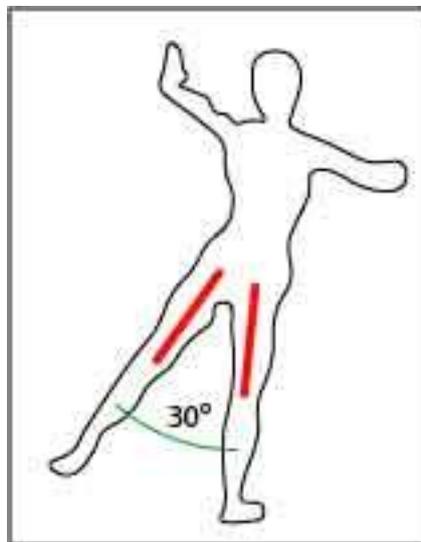


Figura 2. Abducción de cadera

2.2.2. Flexión lenta de cadera y rodilla

Este ejercicio consiste en mover lentamente la rodilla hasta levantarla a la altura del tronco, en este movimiento participa la rodilla y el hemisferio pélvico de la pierna que se está levantando, un ejemplo del ejercicio se lo puede apreciar en la Figura 3. (Bernabé, 2018)

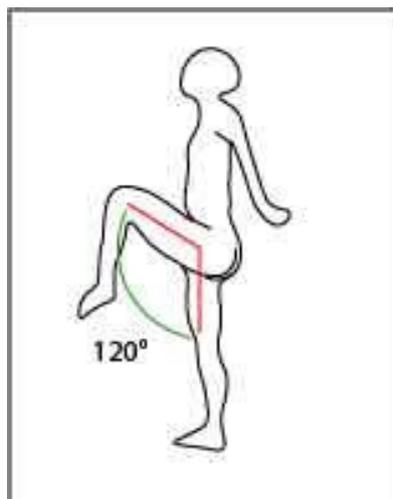


Figura 3. Flexión lenta de cadera y rodilla

2.2.3. Secuencia paso adelante, paso lateral y paso atrás

El ejercicio consiste en realizar tres secuencias diferentes, la que se utiliza un paso al frente, una abducción de cadera y una extensión de cadera. Siendo el ejercicio más complicado y largo. Dado que participa la pelvis en su totalidad.

2.2.4. Extensión de cadera

El ejercicio consiste en realizar un movimiento de pierna a una posición trasera, este se diferencia de una flexión normal de cadera por la amplitud, que tiende a ser mucho más corta. En este participa la parte lateral de la pelvis de donde se está moviendo la pierna, un ejemplo del ejercicio se lo puede apreciar en la Figura 4.

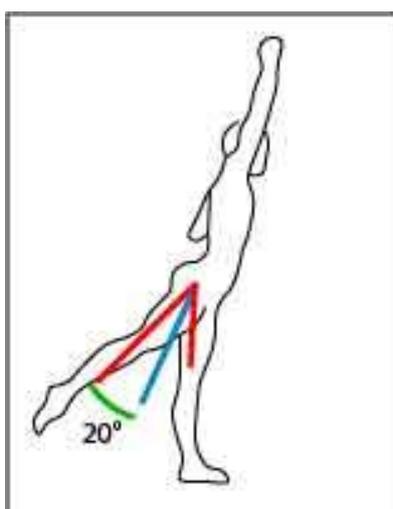


Figura 4 Extensión de cadera

2.3. Juegos serios

Los videojuegos (King, 2002) han sido una forma de entretenimiento bastante popular desde la aparición de las máquinas recreativas a inicios de los años 80. En un principio eran dirigidos a niños y a jóvenes, ahora son usados por muchas personas sin importar su edad. Su principal objetivo de los videojuegos es el del entretenimiento. Sin embargo en los últimos años han aparecido varios juegos que también optan por generar valor educativo o médico a varias personas (Belli, 2008).

Los juegos que generan valor de forma ajena al entretenimiento han aparecido desde mucho antes de que se supiera de su utilidad aparte de divertir. Por ejemplo, en el año de 1999 aparecieron muchos juegos que se consideraban así mismo como simuladores. Un juego simulador, trata de imitar un ambiente de tal manera que tenga gran similitud con la realidad, en sus ejemplos más conocidos está Sim City, que era un simulador de construcción de ciudades (Adams, 1998). Muchas personas que lo jugaron aprendieron varias cosas acerca de cómo mantener y construir una ciudad, incluso como hacer que esta progrese económicamente. Este tipo de juegos incentivaron la creación de varios sistemas similares, lo que apoyó la acogida de los videojuegos en diferentes ámbitos de estudio (Ute Ritterfeld, Cody, & Vorderer, 2009).

En el ámbito de la medicina los juegos serios se hicieron famosos en el 2004, apareciendo videojuegos que permitían conocer la salud mental de las personas por medio de diferentes ambientes, lo que permitió reconocer varias enfermedades como la esquizofrenia en varios pacientes mucho más rápido que con los tratamientos habituales (Mccallum, 2015).

Uno de los principales defectos en las rehabilitaciones es que para los pacientes que han probado las formas tradicionales afirman que estas tienden a ser repetitivas y aburridas, lo que causa que no se siga estrictamente como debe ser. Siendo los juegos serios una alternativa precisa para la resolución de este problema.

Por lo que la aparición de varios sistemas de videojuegos capaces de captar los movimientos de personas ha hecho que la evolución de los juegos serios sea

significativa. Fisioterapeutas especializados en rehabilitación física de pacientes hayan optado por cambiar su método tradicional a un método más tecnológico, dando resultados favorables en sus pacientes. El principal atractivo de un juego serio es que mantiene a los pacientes realizando los ejercicios con sistemas de recompensas y ambientes coloridos y amigables. Esto le da al paciente una sensación de comodidad y confort (Rego & Moreira, 2010).

2.4. Herramientas de modelado 3D

Una de las herramientas más usadas para crear objetos en 3D es Adobe Maya, que nos permite moldear y crear cualquier tipo de cuerpo y sus características principales. Con esta herramienta se han creado un sin número de animaciones que se basan en modelado 3D; incluyendo también efectos especiales de películas. Es así como se puede crear un Avatar que contenga esqueleto y articulaciones; con capacidad de animarlo a nuestra conveniencia (Labsch, Kr, & Grash, 2015).

Con esta tecnología han aparecido varias empresas especializadas en tecnología 3D basada en computación. Una de las más conocidas es Mixamo, con sede en San Francisco; la cual se encarga de diseñar y vender servicios basados en web para animación de personajes 3D. De esta página y de manera gratuita con una cuenta de Adobe se puede obtener modelados y animaciones para desarrollo, incluso, se puede subir propias para beneficio general de la comunidad.

Blender es otra de las herramientas de modelado, a diferencia de Maya esta es totalmente gratuita, al ser más ligera e intuitiva se optó por la utilización de esta plataforma al momento de modelar objetos no humanoides para el proyecto (Brito, s/f).

3. DESARROLLO DE LA SOLUCIÓN

La noción de juegos serios en el sistema ePHoRT, se encuentra actualmente desarrollado con un ambiente en JavaScript, específicamente haciendo uso de la librería Three.js. La figura 1 muestra la interfaz de usuario desarrollada en el ambiente Three.js. Esta solución presenta algunos inconvenientes, entre los que se destacan el rendimiento en la visualización en los movimientos del avatar ya que

funciona con los recursos computacionales del navegador del cliente. Si el computador del cliente no posee suficientes recursos el juego se ejecuta de forma lenta y esto dificulta la ejecución de los ejercicios.

Por tanto, se consideró el cambiar la tecnología por otro motor gráfico como: Unity 3D y Unreal. Para esto se procedió a realizar una investigación exhaustiva de ambos motores gráficos, que incluía sus ventajas, desventajas y las diferencias entre ellas. Priorizando el rendimiento de la herramienta que éstas tenían en un ambiente web.

Al ser las dos herramientas estudiadas muy parecidas en todos los aspectos de rendimiento, características, funcionalidad y facilidad, se procedió a escoger la herramienta con más información y documentación en el ámbito de los juegos serios. Para esto se determinó que Unity 3D es la más apropiada dado a la extensa documentación que posee para juegos serios y a su vinculación con el dispositivo Kinect.

Se creó un proyecto Unity 3D que contiene cuatro escenas gamificadas que adaptan cada ejercicio terapéutico de una manera entretenida para el paciente. Estas escenas captan los movimientos en tiempo real de un paciente. Los movimientos son reproducidos en una escena lúdica que incorpora un avatar, este elemento tiene como objetivo imitar los movimientos realizados por el paciente, de manera que le sirvan de guía cuando el paciente realiza los movimientos. Esto permite que personas sin importar su edad logren mantenerse jugando por varios lapsos de tiempo sin aburrirse o sentir monotonía en su rehabilitación (Anticevic & Anticevic, 2014).

3.1. Unity 3D

El motor gráfico de Unity 3D ofrece varias funcionalidades al momento de crear un sistema de entretenimiento, la herramienta nos brinda opciones personalizadas y especializadas en cada aspecto necesario dentro del desarrollo de un videojuego, animación, etc. Al ser especializado en videojuegos es muy fácil agregar contenido y adaptarlo a las necesidades que conllevan alcanzar un objetivo.

Unity 3D en su tienda virtual ofrece de manera gratuita varios tipos de materiales y objetos gratuitos. Estos se pueden descargar desde el proyecto de Unity 3D e

importarlos automáticamente a nuestras soluciones. Todos los datos importados se guardan en la carpeta Assets del proyecto.

3.2. Arquitectura

Se creó una aplicación en Unity 3D que interactúa con el sistema adquisición y el API RESTFUL de ePHoRt. En la figura 5 se puede observar el diagrama de arquitectura del juego.

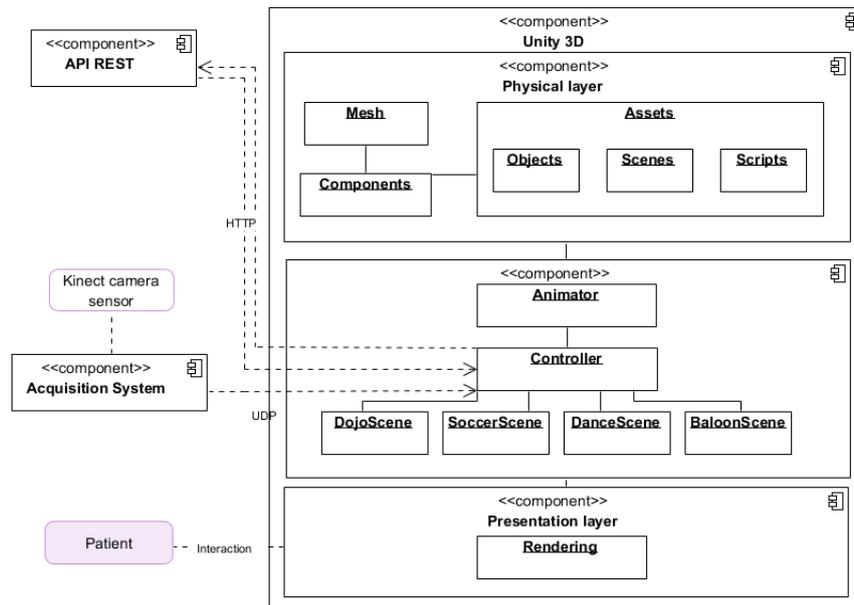


Figura 5. Diagrama de arquitectura

El sistema de adquisición de ePHoRt mediante el uso de una cámara Kinect capta la posición y orientación de cada articulación del cuerpo humano y con una comunicación UDP envía los datos al controlador (servidor) de Unity 3D.

El controlador de Unity consume diferentes servicios del API REST de ePHoRt para inicialización y para la evaluación de los movimientos.

La aplicación Unity 3D tiene tres componentes: capa física, capa de control y capa de presentación.

En la capa física se encuentran todos los archivos necesarios para el funcionamiento correcto del juego. Estos archivos se conocen comúnmente como Assets. En esta capa también se pueden encontrar los diferentes objetos que se importaron al

proyecto para cada escena. Estos objetos corresponden a materiales modelos. Los scripts y las animaciones también son almacenadas en la capa física.

La capa de control posee el avatar, el controlador y las escenas. El avatar es un objeto humanoide, el cual posee una animación de respiración. El controlador recibe datos mediante UDP y los transforma en puntos de movimiento del objeto avatar. Las cuatro escenas comparten un avatar, el cual tiene asociado un script de control que permite la actualización constante de la escena.

La capa de presentación que viene siendo el juego en funcionamiento en un entorno HTML5. En esta capa se presentan todos los componentes de interfaz que visualiza el usuario. La figura 6 muestra la interfaz principal de una escena. Las siguientes subsecciones detallan como fue construcción de un juego.



Figura 6. Interfaz principal de una escena

3.3. Diseño y mapeo de huesos y articulaciones del Avatar

Uno de los principales desafíos de la solución propuesta es imitar movimientos de una persona en tiempo real y reflejarlo en un Avatar virtual. Unity 3D permite hacerlo a través de un objeto 3D incorporado en la capa de control.

Utilizando Mixamo se descargó dos personajes. Un personaje diferente para ambos géneros (masculino que se puede apreciar en la figura 7 y femenino que se puede apreciar en la figura 8) de las personas que van a utilizar el servicio ePHoRt. Ambos avatar cuentan con una animación personalizada de respiración, las propiedades de

la animación se las puede observar en la figura 9. Se hizo de Mixamo dado que este sitio nos facilita una serie de avatar de uso gratuito, lo que disminuyendo el esfuerzo de crear este personaje.

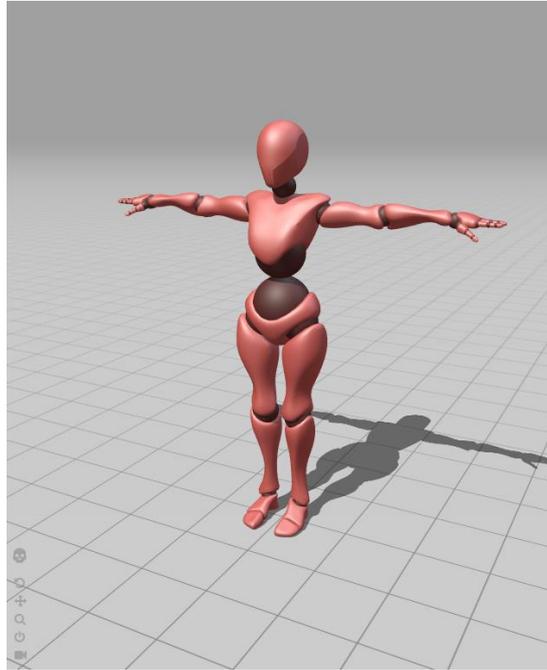


Figura 7. Personaje femenino

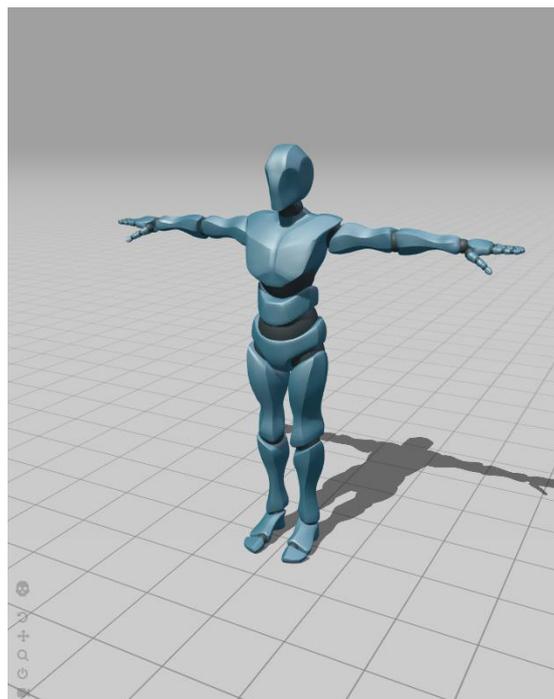


Figura 8. Personaje masculino

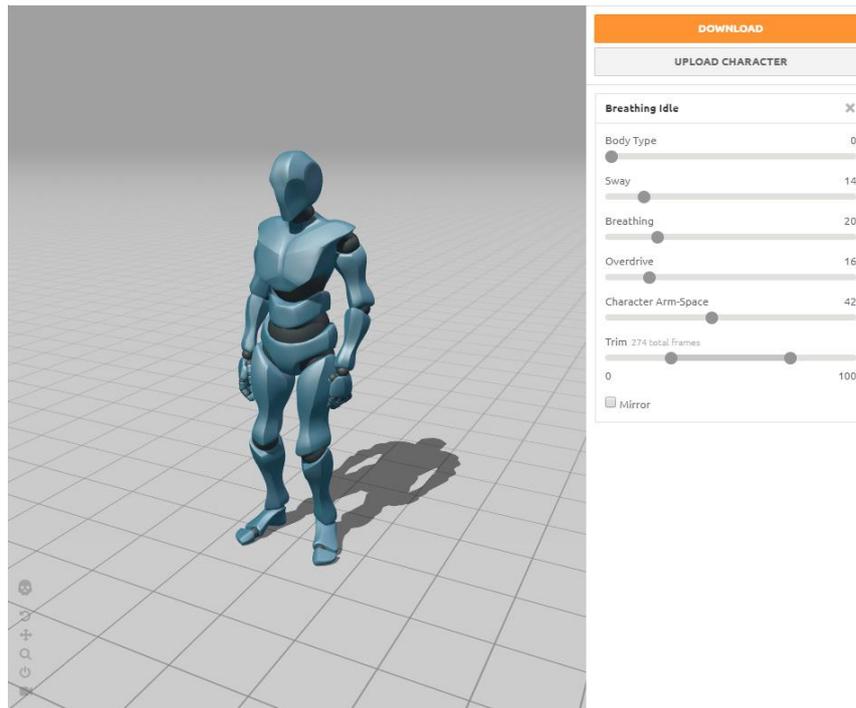


Figura 9. Propiedades de la animación de los personajes

Los personajes descargados de la página Mixamo, pueden ser de dos tipos: Flimbox o FBX (especializado para Unity 3D) o Collada. Para integrarlo a nuestro sistema se creó una carpeta de Objetos llamada personajes y se alojaron los dos modelos. Por sí solos, los modelos son objetos que no tienen físicas, y para nuestro sistema es necesario que imiten comportamientos y movimientos de un humano.

Para que nuestros Avatar asemejen los movimientos humanos se debe cambiar el tipo de animación del objeto, para esto, se lo debe seleccionar dentro la carpeta Assets el lugar donde se guardó el objeto. En la ventana de Inspección se debe seleccionar la opción de Plataforma y cambiar el tipo de animación, el tipo por defecto es “Genérico” al que debemos sustituir por “Humanoide” como se muestra en la figura 10. El tipo de animación humanoide nos garantiza operar los movimientos de las extremidades del avatar y sus articulaciones.



Figura 10. Ventana de Inspección

Otra propiedad que se debe configurar es la definición de nuestro Avatar, lo que permite que Unity 3D reconozca en donde se encuentra cada hueso y articulación en el objeto. Para esto, seleccionamos la opción configurar que se encuentra en la ventana de Inspección, como se muestra en la figura 10. En la mayoría de los casos, cuando un objeto no tiene un esqueleto definido, o es difícil reconocer, se debe realizar un mapeo manual de cada uno, la interfaz del mapeo manual se puede apreciar en la figura 12. Unity 3D nos permite mapear ojos, nariz y boca para usos más específicos de un personaje.

Los modelos humanoides descargados de Mixamo ya contienen un esqueleto definido, por lo que el mapeo es automático, el esqueleto de nuestro avatar se puede apreciar en la figura 11. En nuestro caso, al descargar nuestro objeto con una animación, la postura inicial es semejante a su primer cuadro; lo que puede causar

problemas al momento de integrarlo a una escena. Por lo que directamente en nuestra ventana de configuración de definición forzamos que tenga una pose T que se la pueda apreciar en la figura 11.

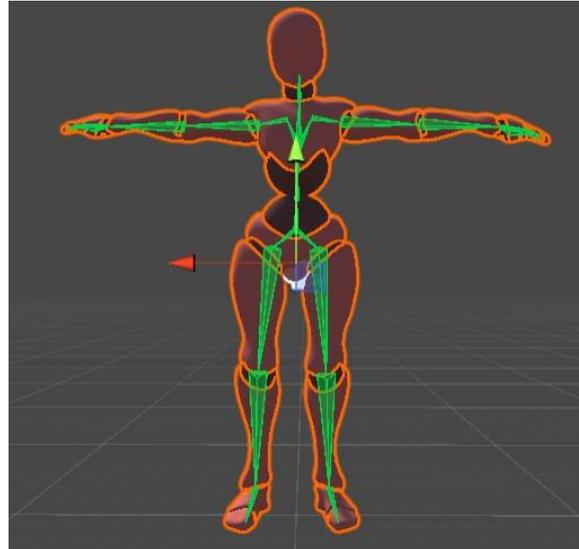


Figura 11. Objeto en pose T con sus huesos

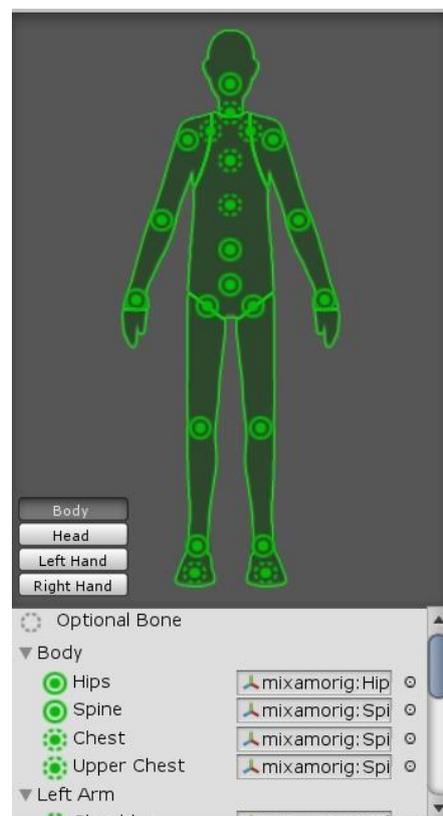


Figura 12. Interfaz de mapeo manual de esqueleto a un objeto

3.4. Diseño de ambientes didácticos

Para la lista de ejercicios que se presenta a continuación, se creó un ambiente didáctico totalmente diferente. Cada ambiente funcionará en una escena distinta e independiente de las otras, para apreciar de mejor manera la distribución de escenas se puede observar la figura 13.

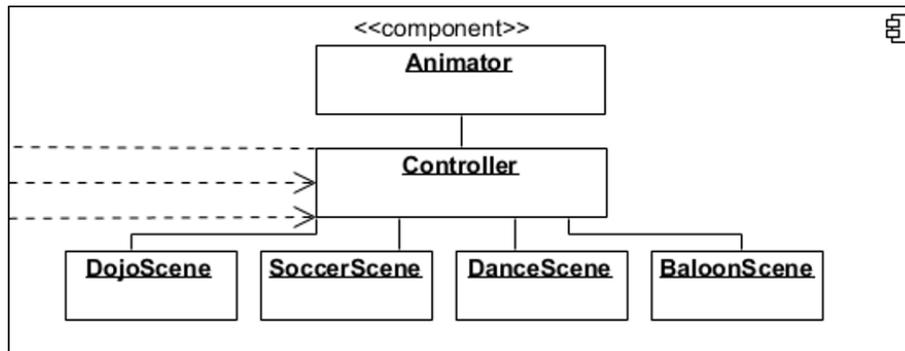


Figura 13. Distribución de escenas

1. Abducción de cadera
2. Extensión lenta de cadera y rodilla
3. Secuencia paso adelante, paso lateral y paso atrás
4. Extensión de cadera

3.4.1. Dojo de karate

Se creó una escena con el nombre DojoScene, ésta debe imitar un Dojo de karate. Los Dojos son lugares donde las personas pueden entrenar un tipo de arte marcial, como karate, judo, kendo, etc. Originalmente provienen de Oriente, por lo que en su mayoría tienden a tener estéticas asiáticas.

Se procedió a crear 4 paneles donde 3 funcionarían como paredes y uno como el suelo, en su mayoría estos sitios de entrenamiento tienen pisos y paredes de madera, conocidos también como pisos flotantes. Para semejar esto, se debe agregar un material y usarlo en los paneles, en este caso el material Wooden Floor 03 obtenido de la tienda Unity 3D, esto se lo puede apreciar en la figura 14.

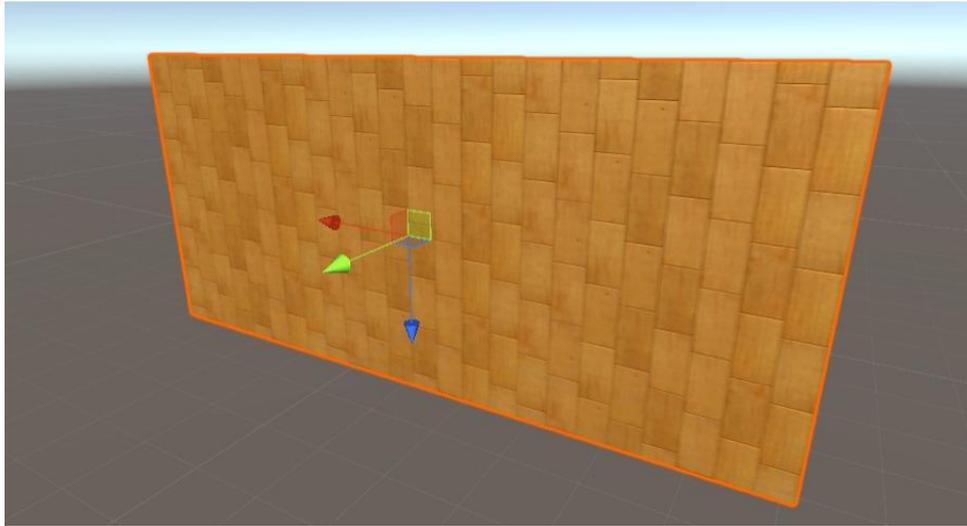


Figura 14. Panel con el material Wooden Floor 3

Colocamos los 4 Paneles asemejando una sala pequeña, a la que también agregamos una luz, esta luz es de tipo direccional y color blanca. Este objeto lumínico está colocado en la parte superior de los paneles y apuntando a la intersección del panel central que funciona como pared y que funciona como piso, esto se lo puede apreciar en la figura 15.



Figura 15. Sala de Dojo

Para darle un toque un poco más cómodo y familiar, se procedió a crear ornamentos que nos permitan definir este ambiente como un Dojo de Karate, para esto, se crearon dos letreros de escritura japonesa colocados en los extremos de la pared trasera, y una bandera de Japón colocada en el centro de la sala.

Para el diseño de los letreros de escritura se utilizó la herramienta Blender. Este

modelo consiste en dos paneles rectangulares, uno un poco más grande, el primero tiene un color negro que asemeja a un cuadro. El segundo panel de color blanco se encuentra sobrepuesto, lo que da la sensación de ser un lienzo listo para ser escrito con las hazañas de sus estudiantes. Se colocaron dos líneas en la parte superior de los paneles para que tengan forma de un cordel que los sostiene, esto se lo puede apreciar en la figura 16.

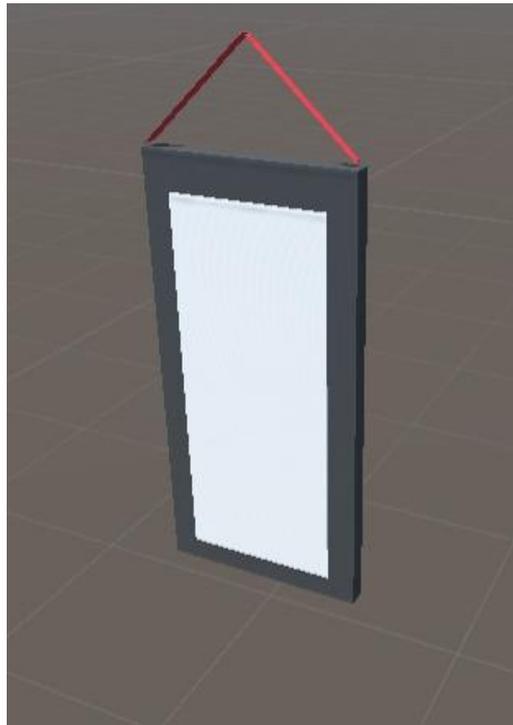


Figura 16. Modelo de letrero de escritura

Se modeló directamente desde la herramienta Unity una bandera de Japón, la que consiste en un panel blanco y una esfera plana en el centro, para darles color se procedió a crear un material diferente para cada uno de los objetos. El primer material contiene un color blanco y el segundo uno rojo, esto se lo puede apreciar en la figura 17.

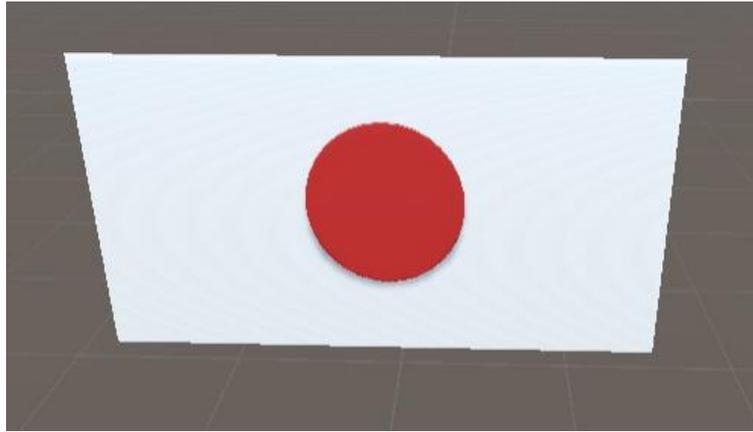


Figura 17. Modelo de bandera japonesa

Para terminar, se colocó una cámara frente a todo el ambiente, ésta con una proyección de tipo perspectiva, la que nos permite apreciar la escena en 3D. Su campo de visión es del 50%, y mostrar solo puntos esenciales, esto se lo puede apreciar en la figura 18.

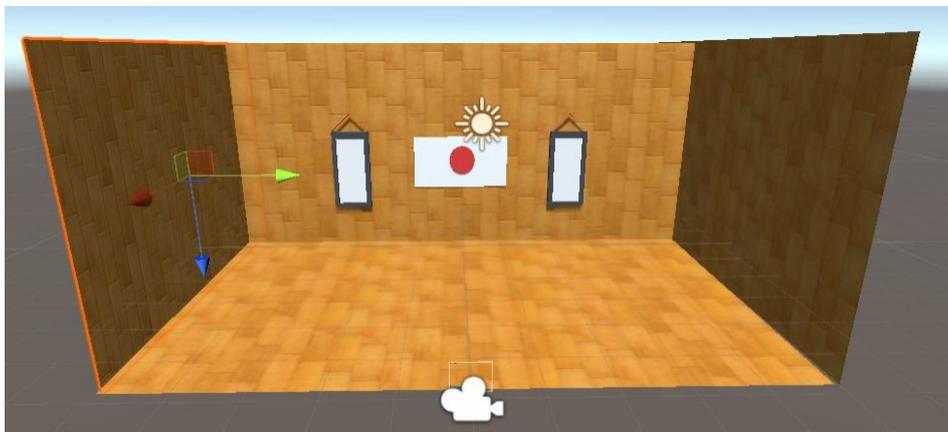


Figura 18. Ambiente Dojo de Karate terminado

3.4.2. Campo de fútbol

El siguiente paso fue crear una escena llamada SoccerScene. Esta escena que consiste en la creación de un ambiente deportivo que asemeja una cancha de fútbol, con elementos característicos como un campo de césped y un arco para anotaciones.

La figura 19 presenta la escena que se recuperó directamente de la tienda Unity 3D. Esta escena fue reducida dado que era muy grande para el escenario del juego. Posteriormente fueron retirados algunos elementos que se consideraron innecesarios.

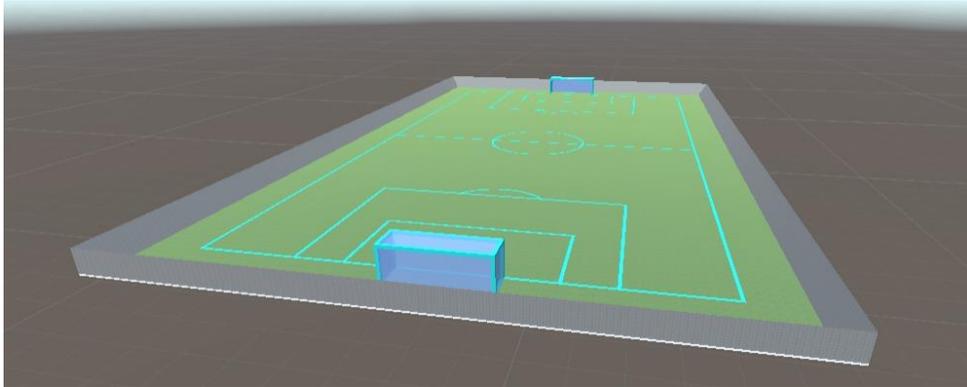


Figura 19. Modelo de cancha de fútbol

Para reducir el tamaño se hizo uso de la herramienta de escalado incorporada en Unity 3D. La figura 20 muestra en el área izquierda el tamaño reducido del campo de fútbol en comparación con la imagen original.

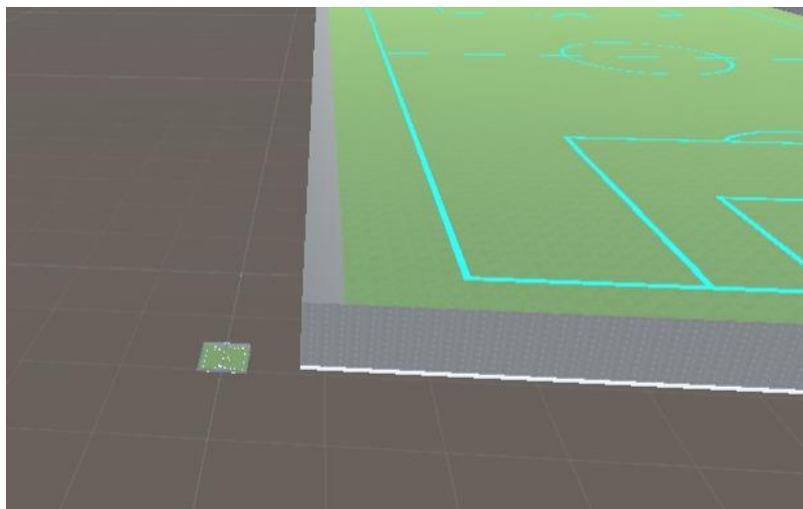


Figura 20. Comparación de tamaño entre objeto original y su modificación

Se añadió una cámara con una proyección de tipo perspectiva con el objetivo de apreciar la escena en tres dimensiones. Su campo de visión es del 50% para mostrar sólo puntos esenciales. Colocamos la cámara en diferentes puntos de la cancha, buscando un punto esencial, en la que se pueda apreciar de mejor manera el ambiente, encontrando un punto ideal justo al inicio de una de las zonas de anotación. La figura 21 presenta en color verde resaltado los elementos eliminados de la escena. Estos elementos fueron eliminados de manera de garantizar una menor sobrecarga y procesamiento en el servidor de juegos.

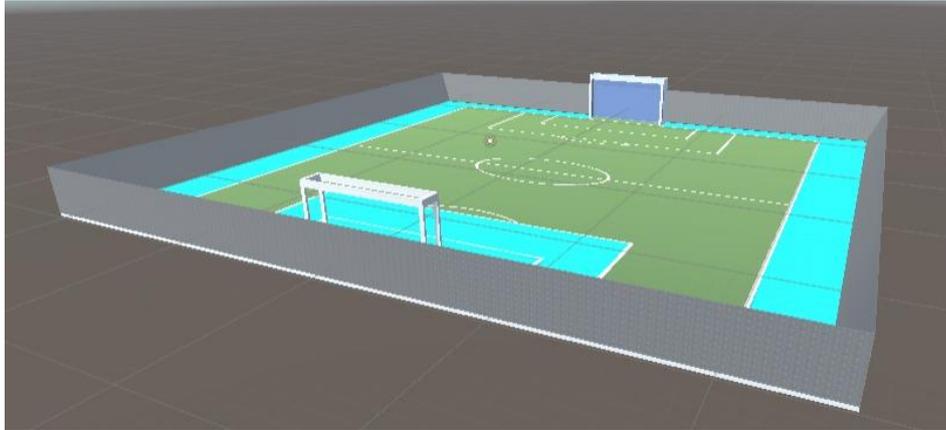


Figura 21. Campo de futbol modificado

Para finalizar, agregamos una luz unidireccional de color blanco en la parte superior del campo, apuntando al centro de la cancha, esto se lo puede apreciar en la figura 22.

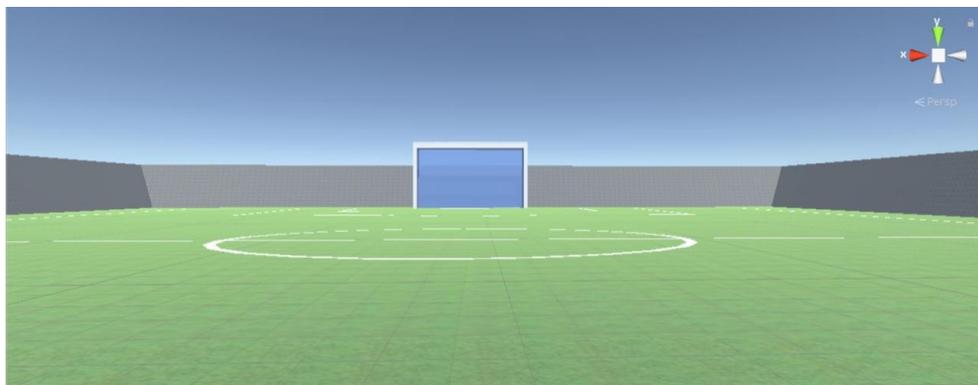


Figura 22. Ambiente de futbol terminado visto desde la cámara

3.4.3. Salón de baile

Se creó una escena con el nombre DanceScene, con la finalidad de imitar una sala de baile. Inspirada en las salas de baile de los años 50 en la que las personas bailaban en un salón totalmente oscuro en una pista con paneles lumínicos.

Para la creación del salón se diseñaron cuatro paneles en los cuales tres servirán como paredes y uno como el piso. Para darle un toque oscuro, se creó un material que tenga un color negro mate el cual fue agregado a los cuatro paneles.

Seguidamente, se unieron los cuatro paneles de manera que asemejen un salón de baile. Asimismo, se crearon varias luces unidireccionales tenues, y de diferentes colores a lo largo de la parte superior del salón. Entre los colores que se escogieron para las luces están: verde, rojo, amarillo y azul. La figura 23 permite apreciar el salón de baile.

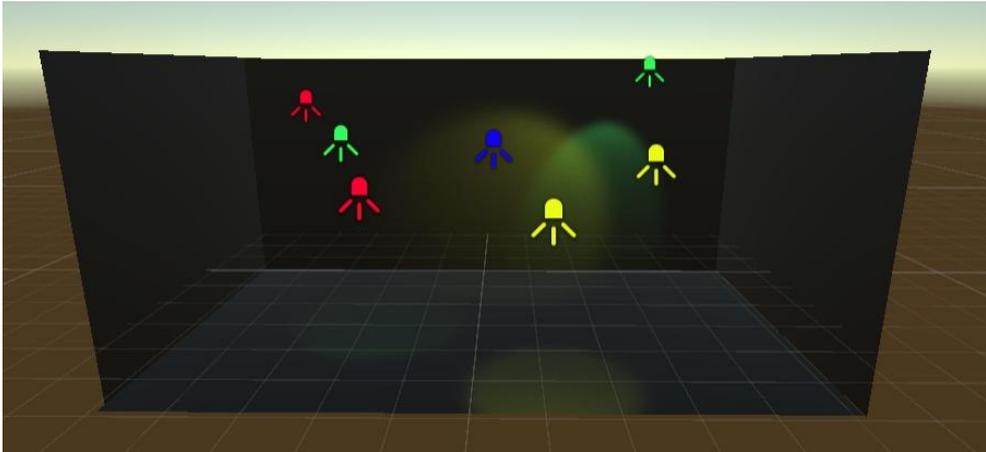


Figura 23. Salón de baile

Para darle una ornamenta que nos permita diferenciarlo como una pista de baile de los años 50, se decidió crear paneles lumínicos a lo largo de suelo. Para llenar todo el salón se necesitaron crear 35 paneles cuadrados, todos del mismo tamaño y colocados en forma matricial con una separación igual entre cada panel. La sensación de iluminación se creó a partir de un material de color blanco, con la propiedad de luminiscencia muy leve, para evitar opacar las luces direccionales ambientales de la parte superior.

Para concluir, se colocó una cámara frente a todo el ambiente con una proyección de tipo perspectiva, la que nos permite apreciar la escena en 3D. Su campo de visión es del 50% para mostrar solo puntos esenciales, esto se lo puede apreciar en la figura 24.

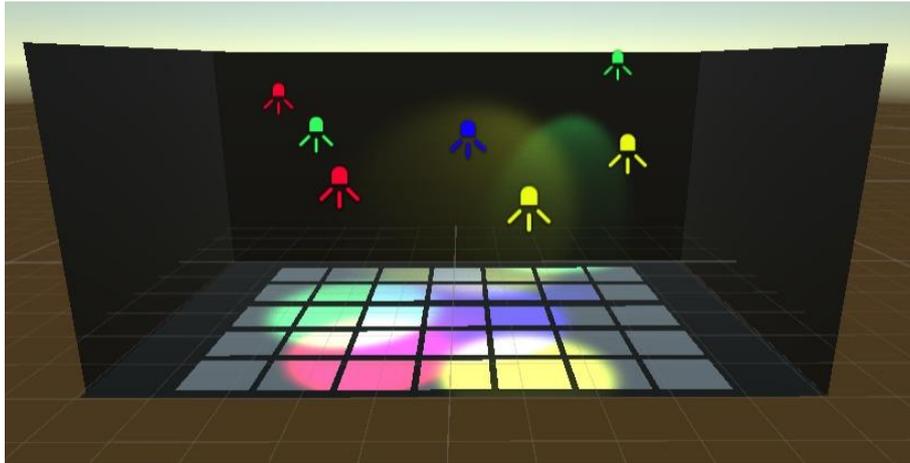


Figura 24. Ambiente Salón de baile terminado

3.4.4. Interacción con un globo

En el cuarto y último ambiente se propuso un ambiente de gimnasio en donde el paciente podía inflar un globo aerostático por medio de una bomba mientras el mismo va realizando las secuencias del ejercicio extensión de cadera. Del ambiente de gimnasio se conservó la idea de que sea en una habitación amplia con colores claros, para ello se creó cuatro paneles, tres que funcionarían como paredes y uno como piso. Para darle color claro se creó un material de color celeste mate. Se unieron los cuatro paneles de tal manera que parezca una habitación, se agregó también una luz unidireccional fuerte en el centro superior del ambiente, apuntando directamente al centro de la habitación, en donde se juntan los paneles del piso y la pared central, esto se lo puede apreciar en la figura 25.

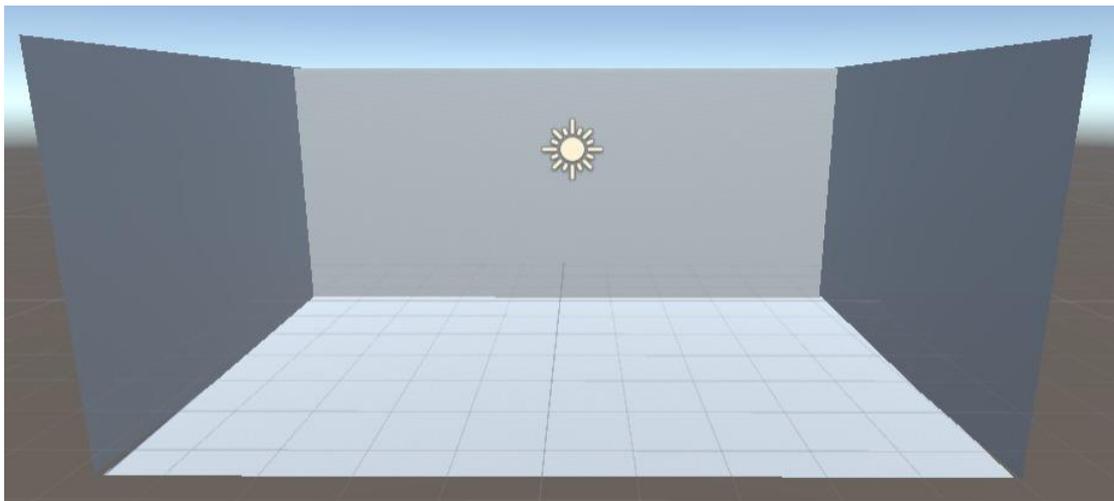


Figura 25. Habitación para el globo

Para la creación del globo aerostático se realizó una búsqueda de un modelo en la tienda de Unity 3D que consistía en un globo aerostático sin material alguno y uno con el material y colores ya establecidos. Para el ambiente se procedió a modificar el globo para acoplarle al objetivo, quitándole algunos objetos innecesarios para el proyecto. En la figura 26 se puede en primera instancia el objeto sin materiales, seguidamente del objeto con los materiales para darle realismo y color. Finalmente se encuentra el globo adaptado para nuestro ambiente, en el que se le sustrajo la canasta y el sistema de combustión.



Figura 26. Modelo de globo con materiales

Para finalizar, se colocó una cámara frente a todo el ambiente, ésta con una proyección de tipo perspectiva, la que nos permite apreciar la escena en 3D. Su campo de visión es del 50% y mostrar solo puntos esenciales. El resultado final del ambiente de globo aerostático se puede apreciar en la figura 28.

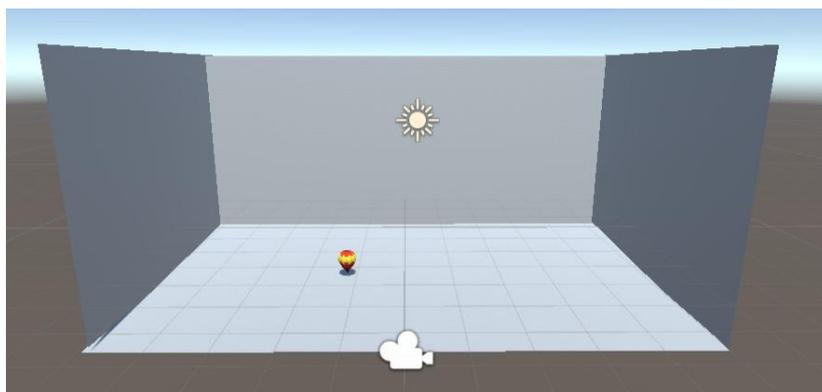


Figura 27. Ambiente de Globo aerostática terminado

3.5. Adaptación de ejercicios terapéuticos

3.5.1. Abducción de cadera

Para el ejercicio de abducción de cadera se escogió utilizar el ambiente de Dojo, en el cuál el paciente tiene que golpear una boya de entrenamiento con su pierna para realizar la secuencia de ejercicios.

La boya fue realizada con la herramienta Blender y consta de una pirámide con la base y punta redondas. Esta representación se incorporó al proyecto con una extensión .fbx. La boya se puede apreciar en la figura 29.

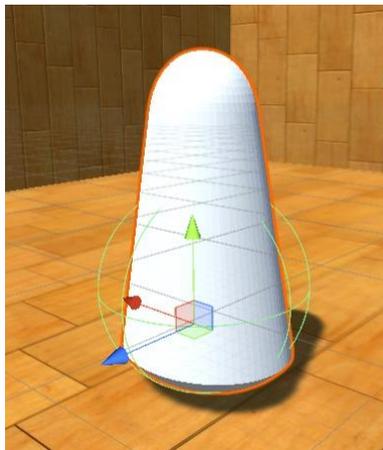


Figura 28. Objeto boya.

Al objeto se le agregó una animación llamada Kick en la que su rotación varía en un intervalo de tiempo. A este objeto se le configuró un controlador de animación. Este controlador contiene la animación en la que el objeto cambia su rotación en el eje Z cada 0.10 segundos. Este eje se resta en 10 hasta llegar a -30. A partir de este punto regresa de igual manera aumentando 10 hasta su posición original. La curva de animación se puede apreciar en la figura 29.

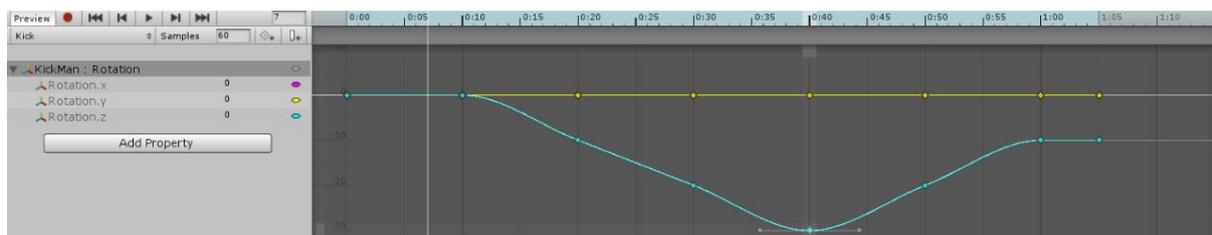


Figura 29. Curva de la animación Kick.

Para simular el movimiento al momento de una colisión entre el avatar y la boya se creó un objeto de tipo animator en el Script del Avatar. Este objeto se enlazó con la animación Kick. Básicamente lo deseado es que se inicie la animación cuando la posición de la pierna del avatar sea igual a la posición de la boya. La figura 30 muestra el ambiente terminado y en ejecución.



Figura 30. Ambiente Dojo de karate en ejecución.

3.5.2. Flexión lenta de cadera y rodilla

El ejercicio de flexión lenta de cadera y rodilla se realizó considerando el ambiente de fútbol. El juego consiste en que el paciente deba golpear una pelota levantando la rodilla. La pelota debe moverse verticalmente hacia arriba unos momentos y luego caer para iniciar otra sesión del ejercicio. La figura 31 presenta la escena del juego.

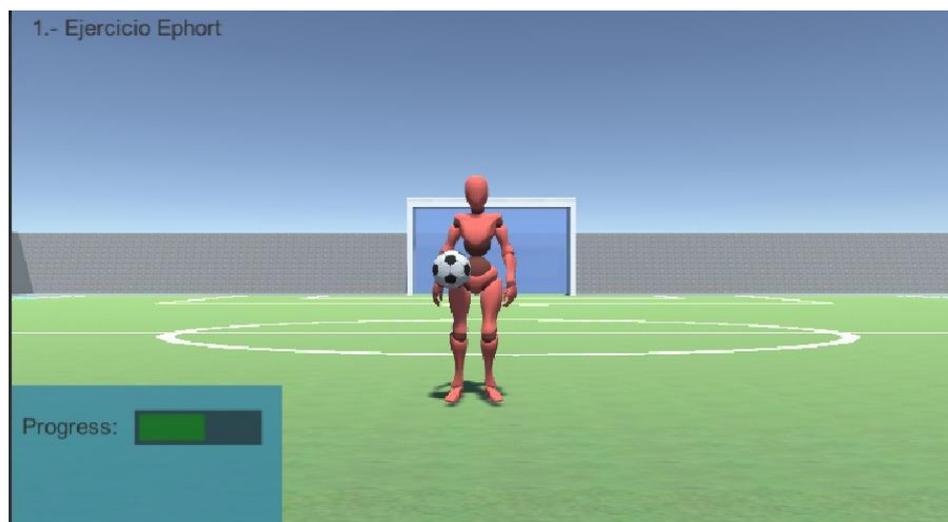


Figura 31. Ambiente Campo de fútbol en ejecución

En esta escena al ser la posición de la pelota una variable, ya que para cada persona el ejercicio puede variar, se debió crear una animación de cambio de posición en el eje Y. Para esto creamos una variable global con la posición inicial de la pelota. Asimismo, creamos una sentencia de colisión con las siguientes condiciones:

1. Cuando la rodilla del avatar tenga la misma posición que un extremo de la pelota, la variable global debe incrementarse en 0.10 en cada Frame.
2. Un contador se incrementa en uno cada vez que la condición anterior se cumpla.
3. Cuando el contador alcance el valor de 5, la condición número 1 en lugar de incrementar en 0.10 se reducirá en 0.10 hasta llegar a la posición inicial.

3.5.3. Secuencia paso adelante, paso lateral y paso atrás

El ejercicio de secuencia de paso adelante, paso lateral y paso atrás se realizó haciendo uso de la escena de baile. El juego consiste en que el paciente tiene que realizar la secuencia del ejercicio pisando pequeños paneles que se encuentran en la escena. Como los paneles ya fueron creados al diseñar la escena, basta con escoger los paneles necesarios y cambiarlos de color. La figura 32 muestra la escena para este ejercicio.



Figura 32. Ambiente Sala de baile en ejecución

Los paneles que el paciente debe pisar fueron identificados por otro color. Los paneles fueron enlazados con el Script del Avatar mediante tres objetos tipo MeshRenderer. Al iniciar la aplicación cambiamos el color con el método `material.color` del panel

frontal a rojo, tal y como lo apreciamos en la Figura 33. Asimismo, creamos tres sentencias de colisión, una para cada panel, haciendo uso de las condiciones siguientes:

1. Al momento de que la posición del pie del avatar sea igual al panel frontal, este panel obtiene el color original, no obstante, el panel lateral obtiene el color rojo.
2. Al momento de que la posición del pie del avatar sea igual al panel lateral, este panel obtiene el color original, no obstante, el panel trasero obtiene el color rojo.
3. Al momento de que la posición del pie del avatar sea igual al panel trasero, este panel obtiene el color original, no obstante, el panel frontal obtiene el color rojo.

3.5.4. Extensión de cadera

El ejercicio de extensión de cadera se realizó mediante la escena del globo. El objetivo del juego es que el paciente debe cambiar la rotación de un objeto que se asemeja a una balanza. La rotación permite inflar un globo presente en la escena. La figura 33 muestra el ambiente en ejecución.

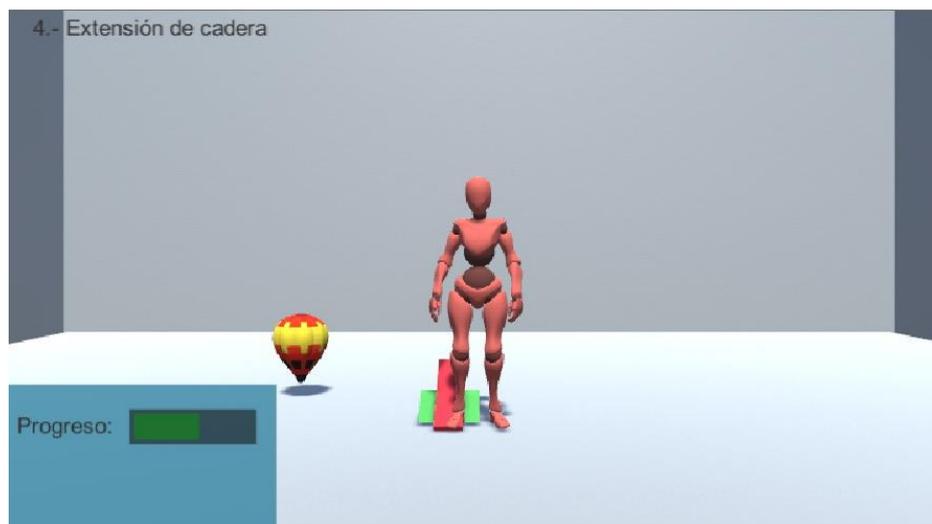


Figura 33. Ambiente Interacción con un globo en ejecución

Para diseñar la balanza, se utilizó la herramienta Blender. La balanza consiste en dos triángulos perpendiculares que contienen un objeto plano conectado en sus puntas. La figura 34 permite apreciar el diseño del objeto de la balanza.

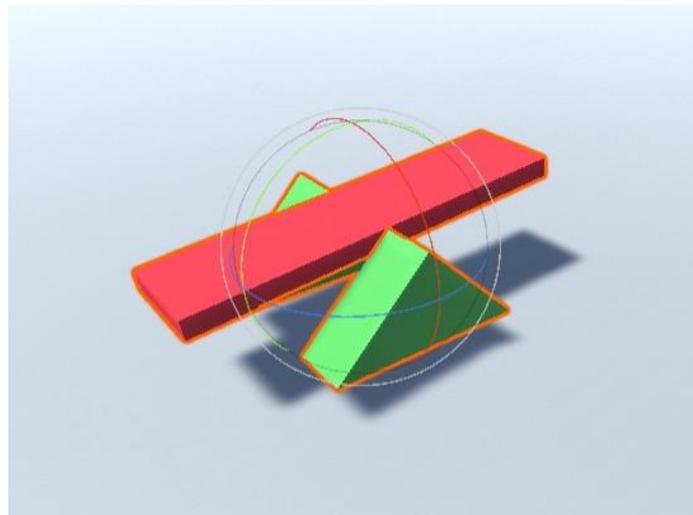


Figura 34. Objeto balanza.

Una vez definido el objeto balanza se procedió a realizar dos animaciones llamadas FrontalKick y BackKcick, en la que su rotación de la tabla central varía en un intervalo de tiempo. Para que el objeto funcione con las animaciones se tuvo que añadir de igual manera un controlador de animación, en total, uno para cada animación. La primera animación le suma 5 puntos al eje Y cada 0.10 segundos, esto durante 10 veces. La segunda animación hace lo mismo que la primera con la diferencia de en vez de aumentar, resta los 5 puntos. La figura 35 y 36 muestra las curvas de animación.

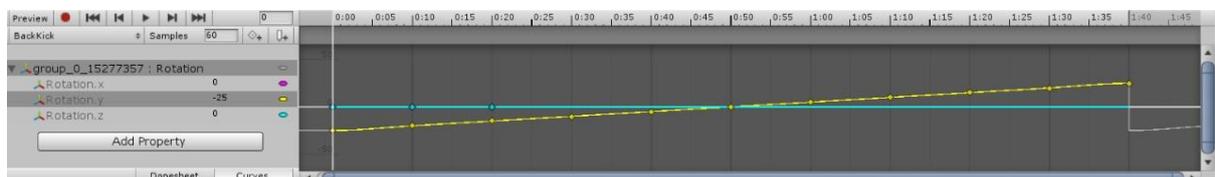


Figura 35. Curva de animación BackKick

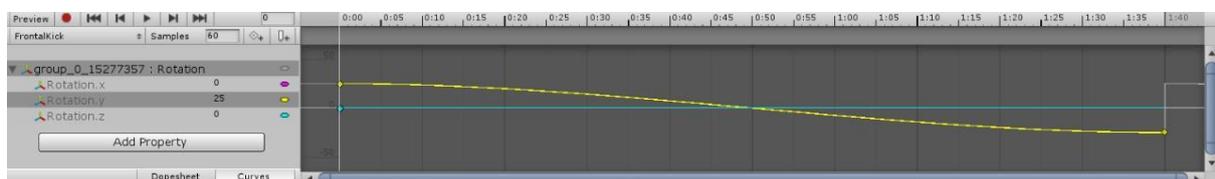


Figura 36. Curva de animación FrontalKick

Dentro del script del avatar en la función Update creamos dos sentencias de colisión. En la primera cuando la posición del pie toque sea igual al extremo frontal de la tabla

empieza la animación Frontalkick. La segunda iniciará la animación BackKcick cuando la posición del pie sea igual a la posición trasera de la tabla.

Para dar la sensación de inflar al globo dentro de cada sentencia de colisión creada cambiamos la escala del objeto. Para esto creamos una variable de tipo GameObject y la igualamos al globo. Cambiamos la escala de x, y, z aumentándoles 5 puntos en cada colisión.

3.6. Transformación de datos UDP en puntos de movimiento

3.6.1. Desarrollo de función de obtención de datos

Para implementar una solución de recolección de datos mediante el protocolo UDP, desde la plataforma ePHoRt hacia el juego serio, lo primero que se debe realizar es crear un hilo para controlar la recepción de los datos. Un juego serio posee un hilo principal de ejecución que controla todos los componentes que intervienen en el juego. El hilo de recepción de datos debe funcionar al mismo tiempo que el hilo principal. De esta manera si una función de cada hilo intenta modificar la misma variable para un componente existente en el juego, no habría conflictos y el hilo secundario esperaría a que finalice la modificación del hilo principal.

Se creó una clase llamada UDPReciver, que contiene los siguientes atributos:

1. Un objeto de tipo Thread responsable del control de los hilos.
2. Un objeto de tipo UDPClient responsable de controlar la conexión entre la plataforma ePHoRt y el juego serio.
3. Un entero que funcionará como el puerto de obtención de datos, esta variable es requerida por el objeto UDPClient.
4. Un String que recibirá los datos del skeleton proveniente de la plataforma ePHoRt.

La lógica para la recepción de los datos provenientes de la plataforma ePHoRt consiste en crear una instancia del objeto UDPReciver. Una vez instanciada la clase se debe inicializarla, generándose el nuevo hilo de recepción. El nuevo hilo constará con la función ReciveData, en la que se ejecutará de forma infinita para recibir los datos. Dentro del ciclo infinito con la función Recive de la clase UDPClient, obtenemos los datos codificados provenientes de la conexión UDP. Estos datos son decodificados en tipo UTF8 antes de asignarlos a la variable global de tipo String.

Es importante conocer que, una vez iniciado el hilo, este debe ser cerrado de forma manual, ya que, si se olvida de cerrar, este seguirá funcionando, aunque la aplicación este en pausa o deshabilitada. Para esto se creó una función que destruye el hilo de recepción. Para lograr esto se hizo uso de la llamada DisableUDPRecive que se aprecia en el diagrama de actividades de la figura 37.

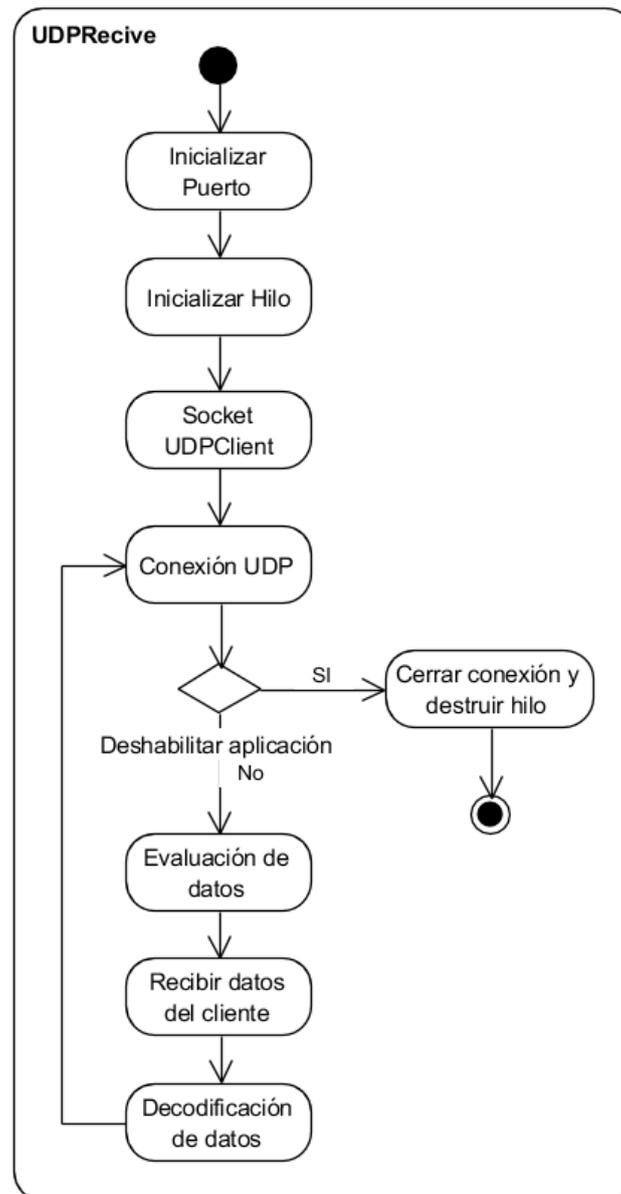


Figura 37. Diagrama de obtención de datos mediante el protocolo UDP. Representación del hilo secundario

Con el objetivo de que otras clases accedan a los datos que son recibidos mediante el protocolo UDP, se creó una variable llamada LastRecivedUDPPackets que recibe

los datos provenientes del protocolo UDP. Esta función se le dio el nombre de `GetLastestUDPPacket`.

3.6.2. Obtención de datos entre el hilo principal y el hilo secundario

Se agregó un Script que va a funcionar para desarrollar todos los métodos necesarios para que nuestro juego serio funcione. Unity 3D posee los métodos automáticos `Start` y `Update`. El método `Start`, se activa una vez que la aplicación es iniciada y solo lo hace una vez. El método `Update` es un método que sigue un ciclo infinito hasta que la aplicación sea deshabilitada o pausada.

Se creó un objeto llamada `UDPObject`, del tipo `UDPRecive`, la clase anteriormente creada. Esta es inicializada en el método `Start`, y llamamos a la función de inicialización para que el hilo secundario de recolección de datos empiece a funcionar al mismo tiempo que el hilo principal de la escena.

Se implementó una variable global de tipo `String`, la cual recibe los datos mediante en protocolo UDP. A esta variable se le dio el nombre de `readData`, y es modificada en el método `Udpate`. Es en este método que se hace el llamado a la función `GetLastestUDPPacket`. De esta manera ya podemos utilizar la información en el hilo principal de la aplicación.

Unity 3D dispone del método `OnDisable` para realizar acciones al momento de deshabilitar la aplicación. Dentro de este método llamamos a la función de cerrar el hilo y deshabilitar el cliente del objeto `UDPObject`, esta secuencia de acciones se puede apreciar de mejor manera en el diagrama de actividades de la figura 37.

3.6.3. Transformación de Datos skeleton en puntos de movimiento

Los datos recibidos provenientes del protocolo UDP están formateados bajo una estructura JSON. Para hacer uso de este formato esta estructura se debe deserializar en un objeto. Para realizar esta labor se crearon 3 clases diferentes. La primera llamada `Skeleton`. La segunda llamada `SkeletonList`. Por último, una clase llamada `Joints`.

La clase Skeleton recibe todos los datos que transmite el Kinect V2, entre estos se encuentran el bodyIndex, righHandState y leftHandState. Asimismo, esta clase cuenta con un booleano llamado tracked, un char llamado trackingId y dos double llamados record_starttime y record_timestamp. Estos atributos tienen el objetivo de marcar el inicio y el fin de la sesión con el dispositivo Kinect. Finalmente, el dato más importante para nuestro proyecto es un arreglo de tipo Joints (clase creada anteriormente).

La clase de tipo Joints, tiene los datos necesarios para realizar que el avatar se mueva imitando los movimientos de la persona que se encuentra frente a la Kinect V2. Todos los datos que tiene esta clase son de tipo Double, es decir, números que contienen decimales. La posición de cada elemento del cuerpo está dada por los atributos cameraX, cameraY y cameraZ, mientras que rotación de cada elemento del cuerpo están dada por los atributos orientationX, orientationY, orientationZ y orientantionW.

Es importante resaltar, que los datos recibidos están serializados. Es por ello por lo que las clases creadas deben heredar atributos que extiendan de la clase System.Serializable. Una vez creadas estas clases, se puede empezar a recibir los datos y hacer que un avatar se mueva. Finalmente, se crearon variables que representan cada Joint de los Avatar. La figura 38 muestra cada una de las posiciones de los joints del cuerpo y su número de identificación en el arreglo (Lisa Jamhoury, 2018).

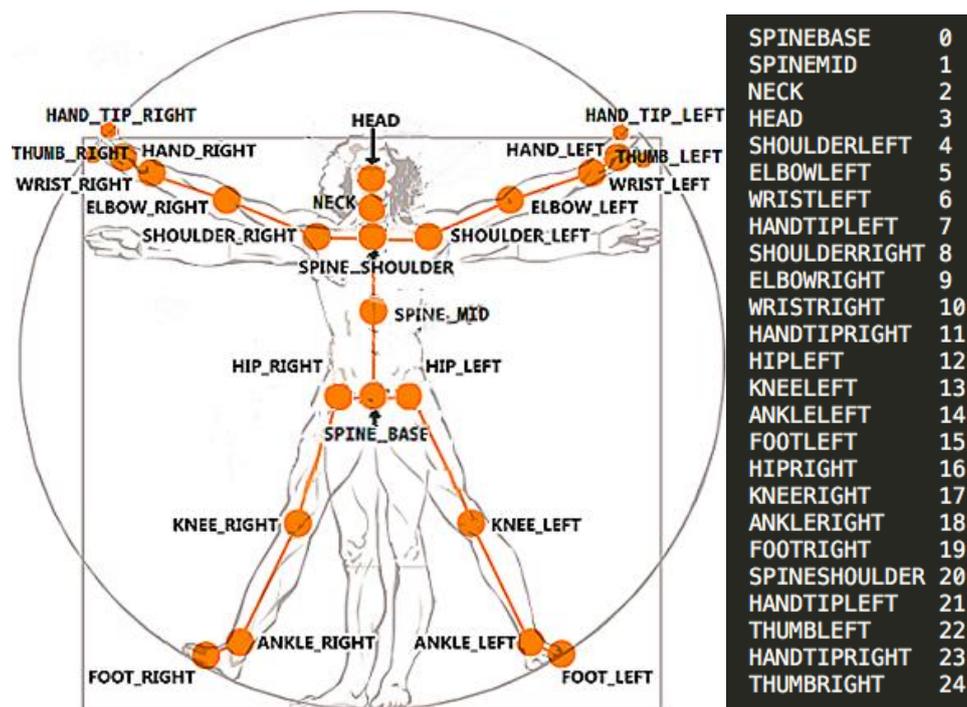


Figura 38. Representación de los joints del cuerpo humano.

Adaptado de Lisa Jamhoury, 2018

El Script que se realizó para los movimientos se encuentra anclado como un componente para cada Avatar, por lo que, para obtener sus componentes se creó un objeto de tipo Animator. Este objeto se inicializa al inicio de la aplicación. Asimismo, al inicio de la aplicación inicializamos cada objeto creado que representa cada parte del cuerpo, con un hueso obtenido del objeto Animator. De esta manera, podemos mover el Avatar, simplemente modificando la posición de cada uno de estos objetos.

Unity 3D, mediante la librería Kinematics, permite realizar movimientos de extremidades con solo mover su posición. Con esta librería se pueden realizar movimientos más fluidos y naturales a objetos humanoides. Consta aclarar que solo funciona para extremidades. Para realizar los movimientos de las secciones restantes del cuerpo debemos proceder de manera manual.

Para realizar los movimientos con la librería Kinematics, lo primero que se debe hacer es revisar que el objeto humanoide tenga una animación. En nuestro caso particular, el Avatar tiene la animación de respiración. Seguidamente, creamos un controlador para la animación. Este control permite modificar las propiedades como la velocidad, la repetición del movimiento, la secuencia de acciones de la animación. También se

creó una máscara que muestra los puntos que utilizará la librería. Esta máscara se agrega al controlador de la animación.

Dentro del Script del Avatar debemos crear una función que llama a las operaciones Kinematics, esta función tiene el nombre de OnAnimatorIk. Dentro de esta función se realizarán las transformaciones de los puntos del skeleton en movimientos del Avatar. Se hizo uso del método SetIkPosition que tiene el objeto animator, el cual necesita dos datos; el Objetivo IK y la posición. El objeto IK se encarga de realizar el movimiento de los elementos, tales como: las manos, los pies, los codos y las rodillas. Por su parte, la posición es la referencia donde se generará el movimiento. Para esto debemos transformar el Joint de cada extremidad en un vector de 3 puntos que recibe los datos de las variables cameraX, cameraY y cameraZ.

Con el objetivo de garantizar una fluidez y un movimiento natural del Avatar, se realiza un filtrado utilizando el método SetIkPositionWeight del objeto Animator. Este método recibe dos datos el Objetivo IK y el tipo de filtro. El tipo de filtrado puede ser un entero entre 1 y 0. El número 1 indica que se filtra desde la anterior posición mientras que el número 0 utiliza la posición de la animación.

Para los demás huesos del cuerpo, se debe realizar la transformación manualmente, esto se lo hizo directo al objeto que tiene mapeado a los puntos del avatar, para esto transformamos la posición del objeto en un vector 3 donde utilizamos los puntos de cada Joint, en este caso cameraX, cameraY y cameraZ.

También se agregó rotación para el cuerpo del Avatar, para darle naturalidad a los movimientos. Los datos que nos devuelve la Kinect V2 son de orientación y no rotación, por lo cual debemos transformarlos para luego convertirlos en un Quaternion. Para eso se utilizaron tres métodos de transformación de orientación y rotación como se muestra en la figura 39. Para rotar en el eje X usamos el método Pitch, para rotar en el eje Y el método Yaw y por último para rotar en el eje Z el método Roll. (Vangos Pterneas, 2017)

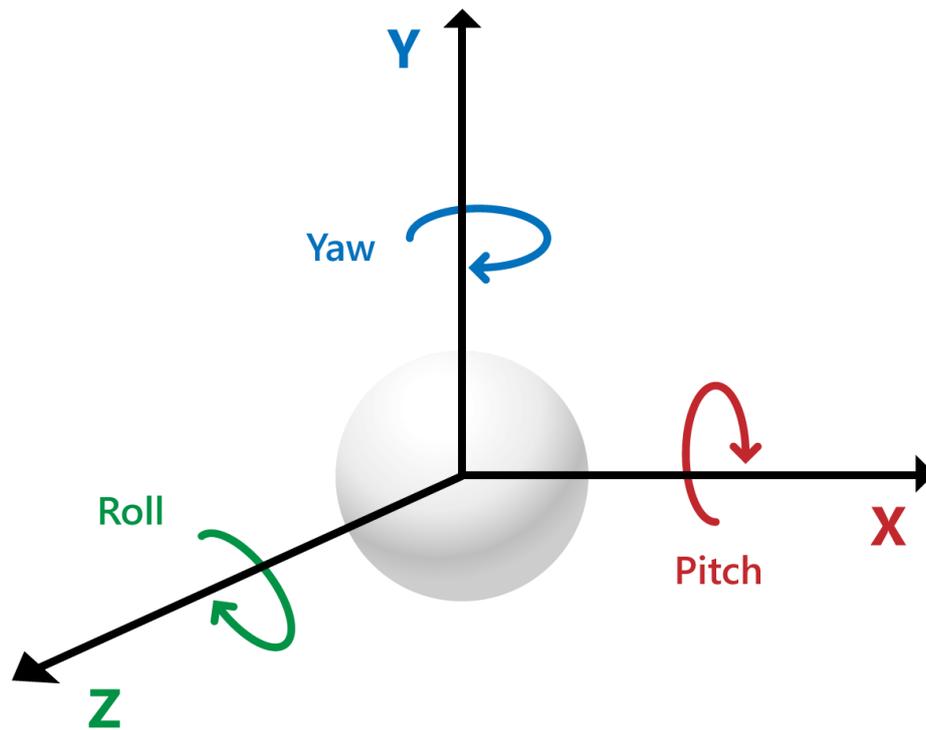


Figura 39. Orientation Quaternion.

Adaptado de Vangos Pterneas, 2017

Al realizar la prueba de rotación, hubo varios problemas con la obtención de datos UDP, ya que varios datos se perdían, y la rotación es sensible. Los movimientos muy bruscos en las manos y pies causaban distorsión, por lo que se tomó la decisión de no usar rotación para los movimientos de estos puntos. El flujo de la aplicación se puede apreciar en el diagrama de actividades de la figura 40.

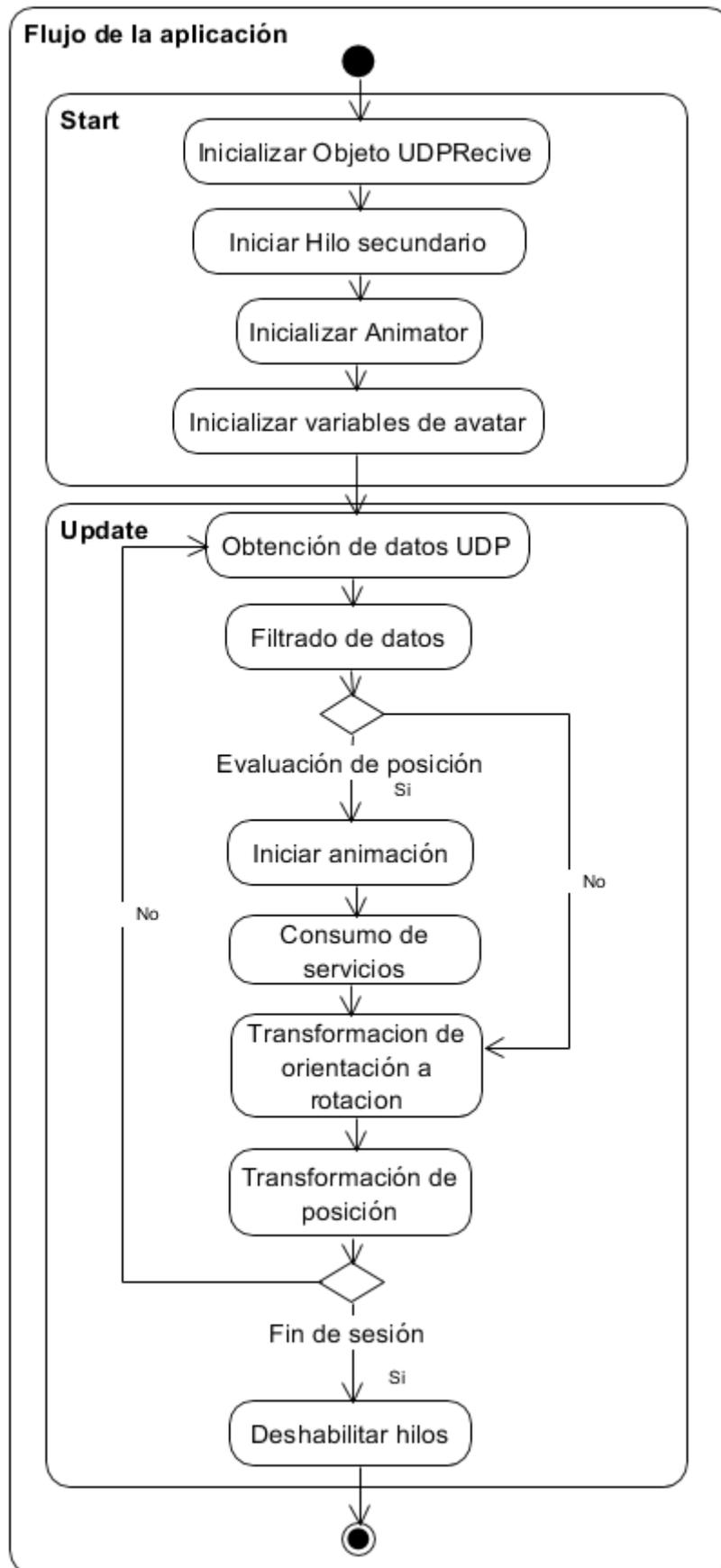


Figura 40. Diagrama de flujo de la aplicación

3.6.4. Integración con la aplicación ePHoRt

Como se mencionó previamente, ePHoRt también cuenta con un módulo API RESTFUL. Este permite evaluar los ejercicios con algoritmos de Inteligencia artificial a través de peticiones HTTP que aceptan puntos de una secuencia de ejercicios del ejercicio de Hip Abduction en particular. La evaluación para los otros ejercicios está en desarrollo (Rybarczyk, 2019).

El primer servicio que debemos consumir es el de MovementType, el cual con una sentencia GET obtenemos los datos necesarios para inicializar el ejercicio. El segundo servicio es el DataReference que nos retorna los datos de referencia del ejercicio. Estos datos son necesarios para realizar el siguiente consumo del servicio MovementType. Este último servicio de tipo POST nos permite enviar un JSON que contiene la referencia que obtuvimos anteriormente, el experimento, la repetición y la serie que realiza el paciente. Por último, hacemos el llamado POST al servicio Skeleton el cual nos retorna un JSON con un ID, movement, j_data y j_data_result. Estos datos corresponden a los resultados de la evaluación de la calidad de movimiento.

3.7. Filtrado de datos

El uso de tecnología UDP, nos permite enviar datos entre varios dispositivos conectados a una red utilizando un puerto abierto. Sin embargo, una de sus desventajas es que, al mandar una cantidad muy alta de datos en un periodo de tiempo, tienden a dañarse o perderse en el camino. Lo que causa que haya incongruencia de datos y nuestro modelo de avatar tenga movimientos erráticos.

Para solucionar esto se procedió a implementar un filtro para suavizar estos datos. Este tipo de filtro se conoce como Low-pass Filter (Li, 2007). El filtro utiliza tres datos para funcionar. Un punto inicial, uno secundario y una referencia de filtrado. De esta manera puede suavizar una señal y disminuir el ruido de esta.

En la primera versión del proyecto ePHoRt la señal del skeleton se evalúa en el módulo API REST que fue desarrollado en Python en donde se utilizó un filtro para

un dato. Este filtro viene de una librería llamada ButterWorth, que tiene algoritmos para C, Python y Java. El problema es que Unity 3D no permite agregar librerías externas, por lo que se utilizó librerías propias de la plataforma Unity 3D para realizar el filtrado. En consecuencia, el filtrado de los datos se realizó mediante la clase Mathf. Específicamente, se utilizó la función Lerp.

El filtrado consistió en realizar un ciclo dentro del método Update que recorre todo el arreglo de Joints del skeleton una vez recibido mediante el protocolo UDP. Se usaron los puntos de posición mediante una llamada al objeto skeletonFilter. En caso de retornar un valor nulo, se asigna el valor del skeleton. En caso contrario utilizamos el valor retornado como dato inicial del filtrado y el valor del skeleton como dato secundario. Para la referencia de filtrado, utilizamos una variable global con un valor de 0.5. Una vez filtrados todos los datos igualamos objeto skeletonFilter con los datos filtrados y lo usamos para los movimientos. Así al momento de ingresar nuevos datos, estos se filtrarán con los datos guardados anteriormente.

4. EXPERIMENTO

Esta sección presenta los resultados de un estudio realizado haciendo uso de la plataforma ePHoRt y el juego serio desarrollado.

4.1. Participantes

Para el experimento se seleccionaron un total de 12 participantes sin tomar en cuenta su nivel profesional o género. Los participantes se encontraban en el rango entre 20 y 23 años de edad. Entre los participantes se encontraban 8 hombres y 4 mujeres.

4.2. Dispositivos

Los dispositivos utilizados para realizar el experimento fueron:

1. Computadora portátil marca Hp con procesador Intel Core I5, 12 GB en RAM y tarjeta de video Intel HD Graphics. En la computadora corría el juego serio y la aplicación ePHoRt.
2. Dispositivo de reconocimiento Kinect V2.

4.3. Tareas

Las tareas realizadas por los participantes de la encuesta fueron:

1. Probar la funcionalidad del sistema ePHoRt.
2. Probar la funcionalidad del juego serio en Unity 3D.

4.4. Procedimiento

El experimento constó de sesiones de aproximadamente 20 minutos. Iniciando con una pequeña introducción explicando el funcionamiento y como interactuar con los dos ambientes (juego serio en Unity 3D y ePHoRt). Esta introducción duró aproximadamente 5 minutos por cada participante.

Seguidamente, los participantes procedieron a interactuar con el sistema ePHoRt durante 5 minutos, para luego continuar de igual manera con el juego serio desarrollado en Unity 3D.

Se concluyó el experimento realizando una encuesta de aceptación para cada interacción con cada ambiente. De esta manera se podrá obtener información necesaria para corroborar el éxito o fracaso de cambiar el ambiente en el cual los pacientes ejecutan sus ejercicios terapéuticos.

5. RESULTADOS

Se realizó una encuesta basada en SUS de valoración de experiencia del sistema; la cual consta de 5 preguntas que nos dan datos acerca de si la aplicación es viable como una forma entretenida de realizar rehabilitación de manera sencilla.

Se evaluó cada pregunta con una métrica en donde la persona encuestada debe calificar del con una escala Likert. En donde 1 significa que está muy en desacuerdo y muy de acuerdo con el enunciado. Las preguntas de la encuesta fueron:

Pregunta 1. Me gustaría hacer uso frecuente de este sistema

Pregunta 2. Encuentro al sistema muy difícil de usar

Pregunta 3. Creo que podría usar este sistema en mi hogar sin la necesidad de un profesional.

Pregunta 4. Encuentro entretenido el uso del sistema

Pregunta 5. Considero que el sistema se acerca más a un juego que a un tratamiento médico.

En la figura 36 se puede apreciar los resultados de la encuesta en gráfico de barras, donde la barra de color rojo representa las respuestas al ambiente ePHoRt y la barra azul el ambiente desarrollado en Unity 3D.

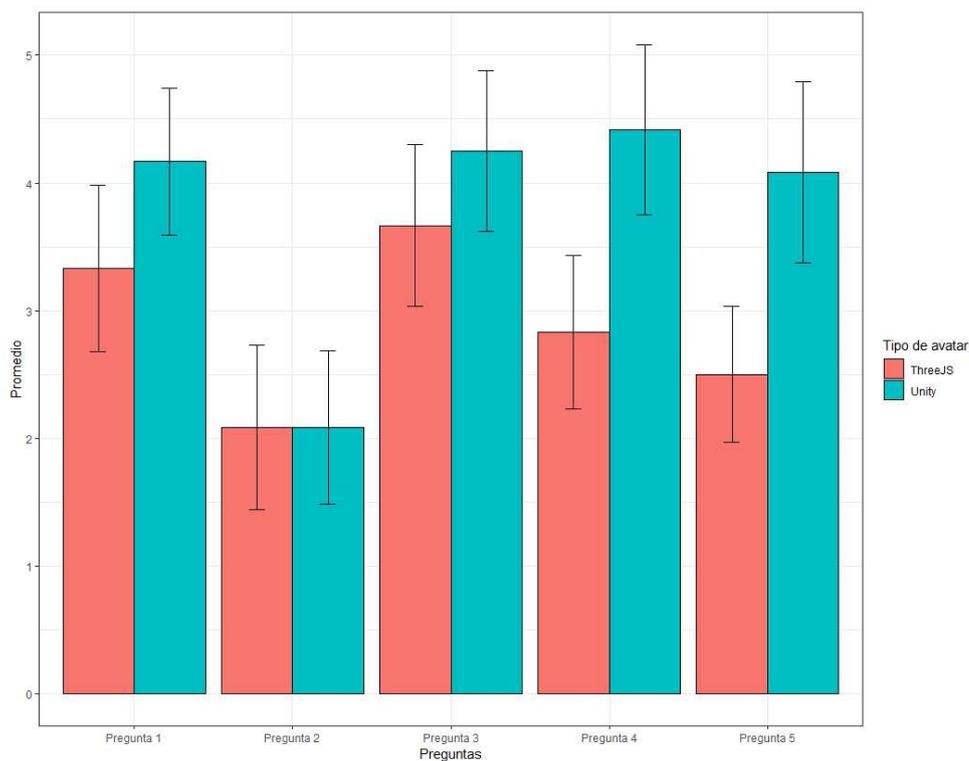


Figura 41. Diagrama de resultados de la encuesta

5.1. Discusión

Los resultados muestran que de forma general las personas encuentran atractiva la forma de realizar los ejercicios por medio de un entorno gamificado, lo que permite que utilicen el sistema de manera frecuente y así seguir con su rehabilitación de manera correcta.

La dificultad de la aplicación no es un gran problema ya que las personas lograron entender el sistema de manera rápida, lo que nos demuestra que es bastante intuitivo. Esto se puede observar en la respuesta obtenida de la segunda pregunta que hace énfasis a la dificultad de la aplicación. Esto evidencia que los usuarios se sintieron cómodos y entretenidos realizando sus diferentes ejercicios con el avatar y ambiente del juego serio que con el avatar de puntos inicial.

Se comparando los resultados de la encuesta para ambas plataformas se deduce que un ambiente con un avatar amigable y un objetivo incentivan a los pacientes al uso constante de la plataforma sin caer en la monotonía ni el aburrimiento.

6. CONCLUSIONES Y RECOMENDACIONES

6.1. Conclusiones

El juego serio que permite a los usuarios realizar sus ejercicios de rehabilitación tras una cirugía de remplazo de cadera de manera entretenida y confortable mediante el uso de una cámara Kinect la cuál capta sus movimientos en tiempo real y proporciona un objetivo en un ambiente gamificado.

El envío de datos captados con una cámara Kinect a través de UDP permitieron que la arquitectura del sistema inicial sea más modular. Los puntos captados por el sensor, los cuales contienen posición y orientación de cada articulación del cuerpo de una persona, podrían ser capturados a través de algún otro dispositivo distinto a la Kinect.

El ambiente gamificado es más agradable para el usuario ya que permitió que mediante comunicación UDP se reciban datos del API REST de la aplicación Django ePHoRT. Estos datos fueron transformados en puntos de posición y rotación de un avatar virtual que asemeja un humanoide.

Por otro lado, se crearon ambientes capaces de adaptar a cada ejercicio planteado a una escena gamificada. Cada ejercicio cuenta con una escena diferente que asemeja diferentes situaciones del mundo real. Esto para que el paciente evita esa sensación de monotonía y aburrimiento típica en la rehabilitación tradicional.

6.2. Recomendaciones

Se recomienda que, al momento de escoger una plataforma para el desarrollo de proyectos basados en juegos serios, se realice una investigación sobre la herramienta para optar siempre por la que contenga mayor cantidad de documentación e implementaciones de aplicaciones similares. Esto acelera el desarrollo e implementación de soluciones basadas en juegos.

Probar la solución con otros dispositivos de adquirió diferentes. Esto nos permitirá observar el comportamiento de la solución con esqueletos obtenidos con cámaras

distintas al sensor de la Kinect.

Adoptar una solución para solventar los problemas de rotación de las manos y pies. Las personas tienden a mover mucho estas partes del cuerpo por lo que esto causo gran perdida y distorsión de la información al recibirla vía UDP en Unity 3D. Se podría estudiar alguna solución para corregir la calidad de los datos.

REFERENCIAS

- Adams, P. C. (1998). *Teaching and learning with simcity 2000*. *Journal of Geography*.
<https://doi.org/10.1080/00221349808978827>
- Anticevic, M., & Anticevic, M. (2014). Creación de juegos serios en Unity 3D.
 Recuperado el 20 de mayo de 2019 de <https://eprints.ucm.es/26476/>
- Belli, S. (2008). Breve historia de los videojuegos *A brief history of videogame*.
 Athenea Digital. Revista de pensamiento e investigación social.
- Bernabé, Á. (2018). La biomecánica de la cadera. Recuperado el 13 de junio de 2019
 de <http://www.tecnicadecarrera.com/biomecanica-de-cadera/>
- Brito, A. (2007). *Blender 3D*. Novatec.
- Goldstein, G., & Beers, S. R. (1998). *Introduction to Rehabilitation Assessment*. *En
 Rehabilitation*. Boston, MA: Springer US. https://doi.org/10.1007/978-1-4757-6392-8_1
- HALSALL, F. (1998). COMUNICACION DE DATOS, REDES DE COMPUTADORES
 Y SISTEMAS ABIERTOS. (S. A. A. MEXICANA, Ed.). MEXICO.
- King, L. (2002). *Game on : the history and culture of videogames*. Universe Publishing,
 Incorporated.
- Labsch, M., Kr, K., & Grash, F. (2015). *Content Creation for a 3D Game with Maya
 and Unity 3D Content Creation for a 3D Game with Maya and Unity 3D*. Institute
 of Computer Graphics and Algorithms, Vienna University of Technology.
- Li, Z. (2007). *Design and analysis of improved Butterworth low pass filter*. *En 2007 8th
 International Conference on Electronic Measurement and Instruments, ICEMI*.
<https://doi.org/10.1109/ICEMI.2007.4350554>
- Lisa Jamhoury. (2018). *Understanding Kinect V2 Joints and Coordinate System*.
 Recuperado el 10 de Abril de 2019 de
<https://medium.com/@lisajamhoury/understanding-kinect-v2-joints-and-coordinate-system-4f4b90b9df16>
- Mccallum, S. (2015). *Gamification and Serious Games for Personalized Health*,
 (Septiembre 2012). <https://doi.org/10.3233/978-1-61499-069-7-85>
- Murphy, K. P., & Karlin, A. M. (2016). Tratamiento de las lesiones
 musculoesqueléticas: Espondilólisis. En Elsevier. *Nelson, Tratado de Pediatría*.
- Rego, P. A., & Moreira, P. M. (2010). *Serious Games for Rehabilitation: A survey and
 a classification towards a taxonomy*.IEEE.

- Rybarczyk, Y., Luis, J., Leconte, L., Jimenes, K., Gonz, M., & Esparza, D. (2019). *Implementation and Assessment of an Intelligent Motor Tele-Rehabilitation Platform..* <https://doi.org/10.3390/electronics8010058>
- Segall, A. (2004). Distributed network protocols. *IEEE Transactions on Information Theory*. <https://doi.org/10.1109/tit.1983.1056620>
- Swartz, M. H. (2015). Sistema nervioso. En Elseiver, *Tratado de semiología*. Madrid.
- Ute Ritterfeld, Cody, M., & Vorderer, P. (2009). *Serious Games: Mechanisms and Effects*. Routledge, Ed.
- Vangos Pterneas. (2017). *Kinect Joint Rotation - The definitive guide*. Recuperado el 23 de Mayo de 2019 de <https://pterneas.com/2017/05/28/kinect-joint-rotation/>

