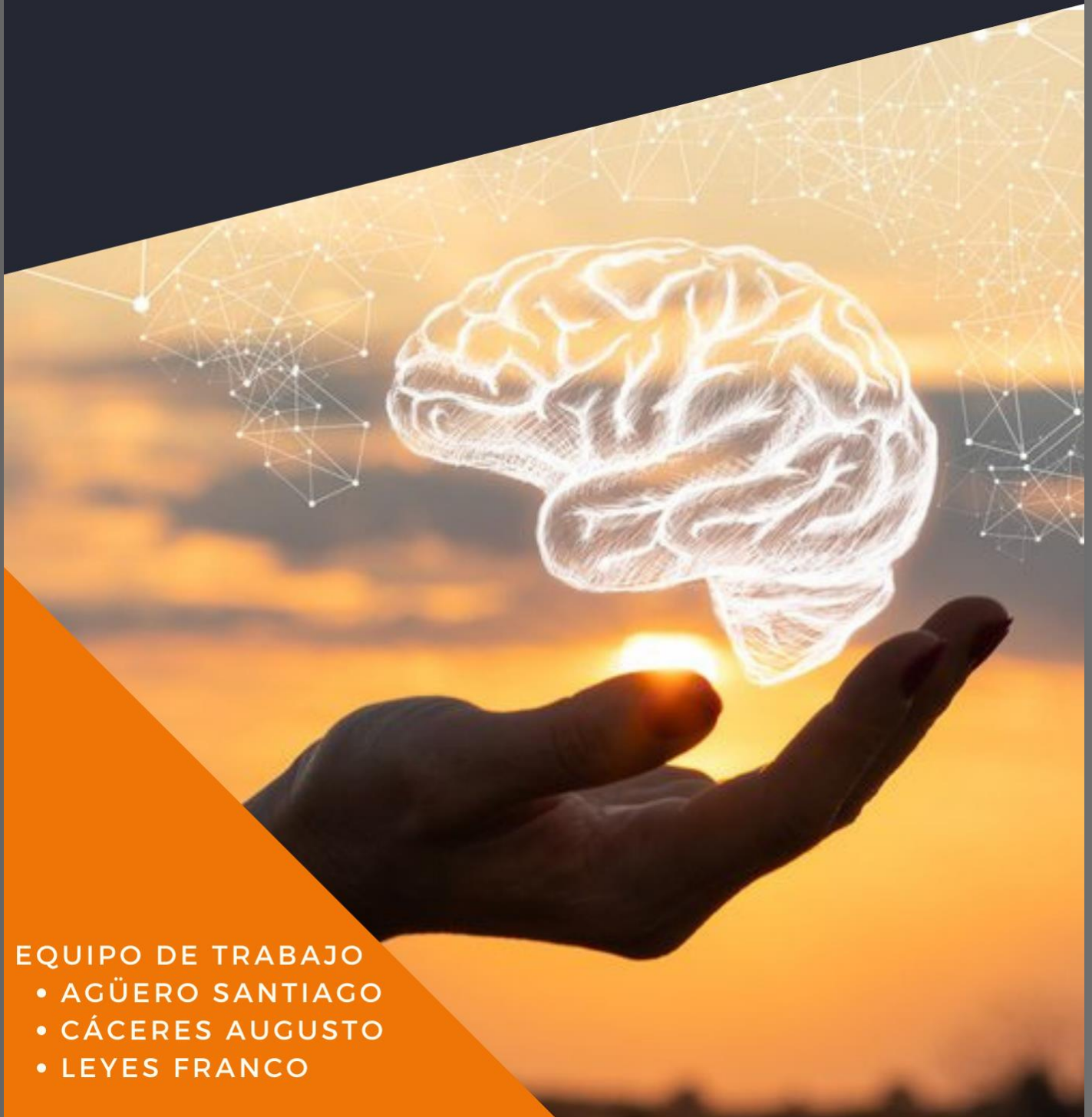


PROYECTO FINAL "REALIDAD VIRTUAL EN REHABILITACIÓN"

**DESTINADO A PERSONAS CON
ENCEFALOPATÍA CRÓNICA**



EQUIPO DE TRABAJO

- AGÜERO SANTIAGO
- CÁCERES AUGUSTO
- LEYES FRANCO

Resumen

El objetivo general del proyecto es la construcción de un juego 3D, implementado en realidad virtual, que sea capaz de correr en dispositivos de recursos medios. Se lleva a cabo el desarrollo de una interfaz de usuario desarrollada en Unity y de un dispositivo HMI, adaptados de modo que ambas partes interactúen entre sí. También se ha integrado visión artificial para poder detectar la postura del jugador, que será replicada por el personaje del juego.

La finalidad es lograr que el usuario final, un niño con encefalopatía crónica, pueda jugar y entrenar motivado en sus sesiones de rehabilitación por los objetivos de cada nivel en el juego. Además, se prevé que la ejecución de juego se realice bajo la supervisión de un profesional kinesiólogo o similar.

Índice

Resumen.....	2
Índice de Figuras	4
Introducción	5
Herramientas	6
Hardware.....	6
• Wii Balance Board	6
• Arduino MEGA.....	6
• Placa HX711	6
• Módulo Bluetooth HC-06.....	7
Software	7
• Unity 3D.....	7
• Python	8
• Visual Studio Code.....	8
• Arduino IDE.....	8
• Plastic CSM.....	8
Desarrollo de la aplicación	9
Interfaz	9
Visión Artificial	9
Balance Board	10
Unity	11
Menú Inicio	11
Niveles	11
Conclusión	13
Bibliografía	14
Repositorio.....	14

Índice de Figuras

Figura 1: Paciente N°1.

Figura 2: Paciente N°2.

Figura 3: Wii Balance Board.

Figura 4: Galga extensométrica.

Figura 5: Puente de Wheatstone.

Figura 6: Microcontrolador ARDUINO MEGA 2560.

Figura 7: Amplificador HX711.

Figura 8: Módulo HC-06.

Figura 9: Diagrama de conexión hardware.

Figura 10: Mapa de Salida de MediaPipe.

Figura 11: Captura de Visión Artificial en funcionamiento.

Figura 12: Menú Inicio.

Figura 13: Fotografía del hardware.

Figura 14: Captura de “Balance Ball”.

Figura 15: Captura de “El Cielo de Cae”.

Figura 16: Captura de “Ruta 40”.

Introducción

La *Encefalopatía Crónica* es un desorden del desarrollo de la postura y movimiento de la persona, que causa limitaciones, atribuidas a una alteración no progresiva del cerebro fetal o inmaduro, que también puede afectar sensaciones, percepciones, aspectos cognitivos, de comunicación y conductuales. Es una lesión no progresiva, tal cual como lo define su nombre, pero cambia con el crecimiento.



Figura 1



Figura 2

En la ciudad de Mendoza existe un centro de rehabilitación llamado “**KINETIC**”, en el departamento de Godoy Cruz, donde desde muy pequeños asisten niños hasta 14/15 años que padecen encefalopatía crónica.

En una entrevista, los profesionales que allí trabajan, nos plantearon que los recursos didácticos con los que cuentan no son diversos, esto cada vez aburre más a los niños, provocando un mal desempeño en la terapia.

Así entonces, teniendo como objetivo brindarles a los profesionales del centro una herramienta más para trabajar con los chicos, desarrollamos un escenario de rehabilitación que integra: Realidad Virtual, Visión Artificial y Electrónica. El cual contiene diversos juegos cuyos mandos están orientados a movimientos que los kinesiólogos buscan que los niños realicen en sus terapias.

Herramientas

Hardware

- Wii Balance Board

Es fundamental en terapia lograr que los niños apoyen por completo la planta del pie en el suelo, para tal evaluación, en el juego, necesitamos ubicar el punto en donde ejercen el peso y saber hacia dónde se corre el centro de gravedad del apoyo. Es por eso que decidimos reutilizar una “Balance board” de la Nintendo Wii.



Figura 3.

La misma está compuesta por 4 celdas de carga, una en cada esquina. La celda de carga es un transductor que convierte la fuerza aplicada sobre ella en una señal eléctrica medible, en su interior alberga a una resistencia llamada galga extensiométrica, la cual tiene la capacidad de comprimirse y estirarse según sea necesario, variando su valor óhmico. Las variaciones en la resistencia son muy pequeñas por lo tanto se utiliza un puente de Wheatstone que da como salida una tensión.

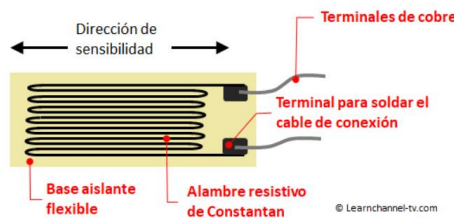


Figura 4

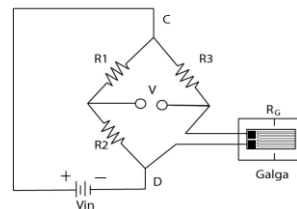


Figura 5

Decidimos entonces solo hacer uso de los cuatro mencionados sensores de presión y además de la estructura de la tabla, desligándonos de todos los demás componentes que ésta contiene.

- Arduino MEGA

El microcontrolador empleado es un Arduino Mega 2560, el cual posee la algoritmia correspondiente a la calibración y utilización de la tabla. Es el encargado de recibir los datos generados por los sensores y enviarlos hacia la computadora.



Figura 6

- Placa HX711

Cada celda de carga fue conectada a un amplificador HX711, que es capaz de adaptar la información para ser comunicada hacia el microcontrolador mediante dos hilos (reloj y datos).

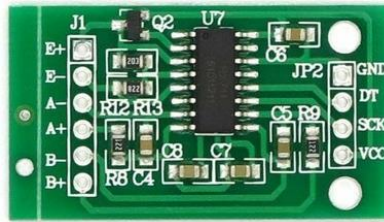


Figura 7

- Módulo Bluetooth HC-06

La comunicación se realiza mediante protocolo Bluetooth, haciendo uso del módulo HC-06, que nos permite establecer comunicación serie con el ordenador.



Figura 8

Diagrama de conexión final

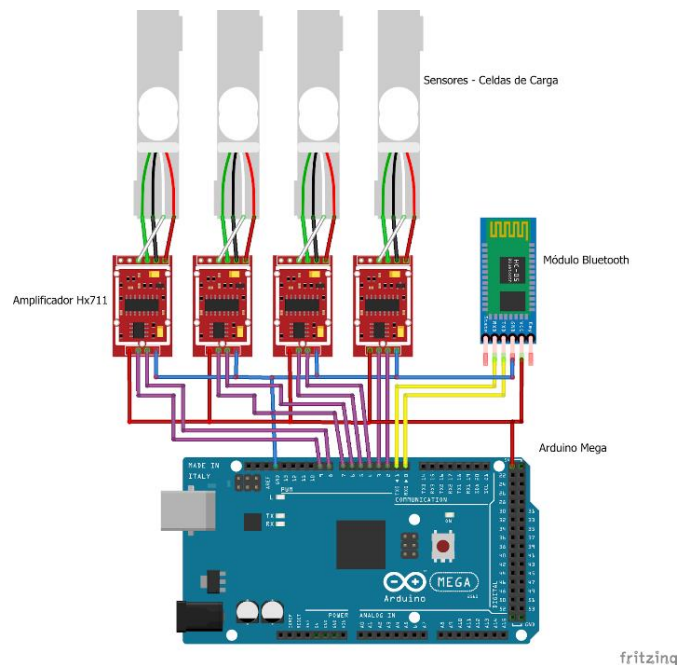


Figura 9

Software

- Unity 3D

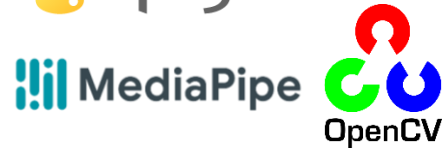
Unity es uno de los motores de creación de juegos más usados en la actualidad. Nosotros empleamos sus herramientas para diseñar los diferentes juegos en un entorno 3D, empleando fundamentalmente sus motores de física, los cuales logran que las acciones y movimientos dentro de la aplicación sean más realistas, aportando gran valor al entorno de realidad virtual.



La versión empleada en este proyecto es 2022.10.f1.

- Python

Aquí es donde se realiza la **Visión Artificial** como tal, en esta sección empleamos CVZone, un paquete de visión artificial que integra “Mediapipe Pose” y “OpenCV”.



Mediapipe es una solución de machine learning de alta precisión, que permite obtener 33 puntos de referencia 3D de todo el cuerpo, teniendo como entrada, imágenes en rgb. Además, esta solución alcanza resultados en tiempo real en la mayoría de computadoras, e incluso en la web.

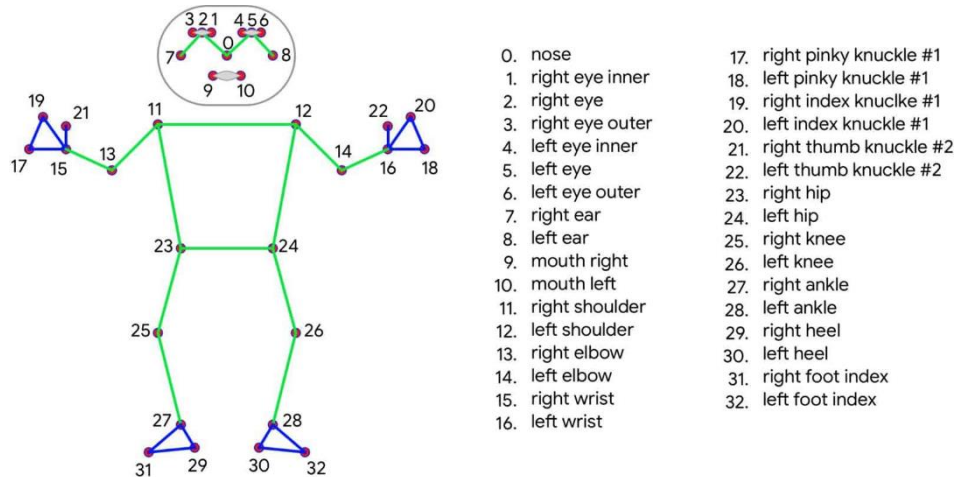


Figura 10

También se encuentra integrado *OpenCV*, que es una biblioteca libre de visión artificial originalmente desarrollada por Intel. Desde 1999 se ha utilizado en una gran cantidad de aplicaciones, y hasta 2020 se la sigue mencionando como la biblioteca más popular de visión artificial. En la actualidad es capaz de realizar detección de movimiento, reconocimiento de objetos, reconstrucción 3D a partir de imágenes, y mucho más.

- Visual Studio Code

Visual Studio Code es un editor de código fuente desarrollado por Microsoft para Windows, Linux, macOS y Web. Lo usamos en conjunto con Unity para editar los archivos en lenguaje C#. Es favorable ya que todas las características de C# son compatibles.



- Arduino IDE

Un IDE es un entorno de programación que consiste en un editor de código, un compilador y un depurador. Además, en el caso de Arduino incorpora las herramientas para cargar el programa ya compilado en la memoria flash del hardware.



- Plastic SCM

Unity Plastic SCM es una herramienta de gestión de código fuente y control de versiones, utilizada para desarrollar proyectos de manera colaborativa, similar a GitHub.



Desarrollo de la aplicación

Interfaz

Visión Artificial

Esta solución se realizó en Python. Se adaptó el funcionamiento del algoritmo junto al de la librería para poder obtener, desde la imagen obtenida por la cámara web del ordenador, con muy bajo retraso, los 33 puntos anteriormente mencionados en formato string. Este string es transmitido por un Socket datagram, basado en el protocolo UDP y ofrecen un servicio de transporte IP sin conexión. Es decir, podemos mandar información a un destino sin necesidad de realizar una conexión previa. La gran ventaja de este tipo de sockets es que los retrasos introducidos son mínimos, lo cual los hace especialmente interesantes para aplicaciones en tiempo real, como la transmisión de audio y vídeo sobre Internet. El proyecto incluye un script de C# capaz de ejecutar este archivo Python y destruirlo una vez concluida la sesión de juego.

La información es recibida en Unity por un Cliente UDP, **UDPRecieve.cs*, que realiza la detección de errores que el protocolo no provee. Finalmente, **MotionCapture.cs*, es capaz de extraer la información tridimensional de los 33 puntos detectados y posicionar al *GameObject* correspondiente en Unity. La visual del personaje está basada en un personaje palitos tipo “Sticky Figure”, al mismo se le han asignado físicas necesarias para detectar colisiones y poder realizar movimientos de manera correcta.

Vale aclarar, que la información correspondiente a la posición ha sido tratada para que todos los puntos estén referenciados al centro de la cadera, escalado a metros para independizarnos de la distancia del sujeto a la cámara. La posición del personaje respecto al mundo es controlada por la posición referencia, esto es realizado en **Character.cs*, que recibe información de teclas o periféricos, que le permiten realizar movimientos hacia adelante-atrás o hacia los costados.

Para esta solución también evaluamos el funcionamiento de otros dos softwares de visión artificial, “Open Pose” y “Barracuda”. Descartamos la primera por tener prestaciones inferiores en cuanto a precisión y retardos temporales. En cuanto a la segunda, esta opción resultaba de fácil implementación en Unity, pero requiere muchos recursos computacionales, de todas maneras su desventaja recae en la poca documentación que lo soporta, y aunque resultaba útil poder aplicar la información a modelos 3D, era compatible con formatos muy poco utilizados en el occidente y de imposible acceso desde este lado del mundo.

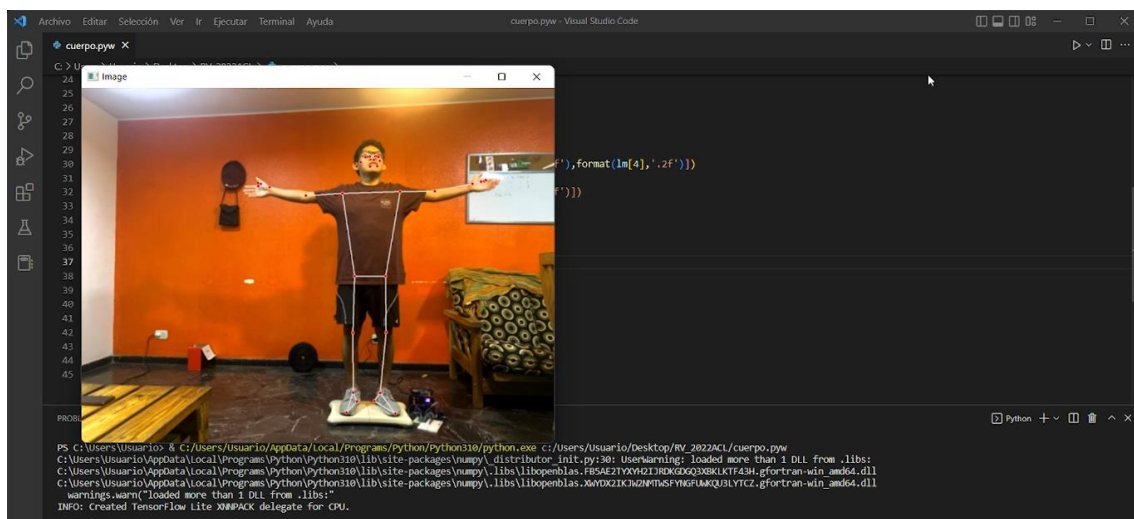


Figura 11

Balance Board

Inicialmente se adquiere la señal de las celdas de carga en Arduino, a través del módulo Hx711, los datos son tratados y formateados en el microcontrolador para ser enviados, a través del módulo HC-06, en formato string hacia el puerto serie de la computadora, **balance.ino*.

Por otro lado, en Unity, **SerialManagerScript.cs* recibe y extrae la información de las 4 fuerzas en formato float y posteriormente calcula la posición del centro de gravedad (en el plano XY), esta es comparada con umbrales para generar señales de avance o movimiento lateral.

La conexión, reconexión y calibración ha sido configurada para ser accesible desde el menú implementado en Unity, de esta manera el usuario no tendrá que abandonar el juego para poder continuar con la diversión.



Figura 12

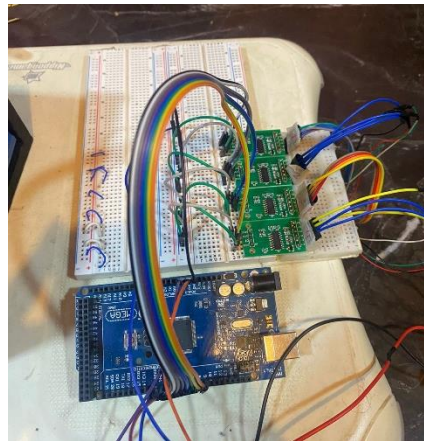


Figura 13

La formulación empleada para el cálculo del centro de presión es:

$$x_{cp} = \frac{(F_{sd} + F_{id}) - (F_{si} + F_{ii})}{F_{total}} * \frac{L_x}{2}$$

$$y_{cp} = \frac{(F_{sd} + F_{si}) - (F_{id} + F_{ii})}{F_{total}} * \frac{L_y}{2}$$

$$F_{total} = F_{si} + F_{sd} + F_{id} + F_{ii}$$

Donde:

$$L_x = 45cm ; L_y = 26.5cm$$

F_{st} ; F_{sd} ; F_{id} y F_{ii} : son recibidas desde Arduino.

Unity

Menú Inicio

Es la pantalla principal donde en un inicio se debe calibrar la balance y posteriormente acceder a cada uno de los juegos, desde los cuales se puede retornar a este mismo.

Luego de haber finalizado el juego se puede salir de la aplicación.

Niveles

1ro: Balance Ball

El nivel hace uso únicamente de la tabla de balance, esta se utiliza para controlar el movimiento lateral de una bola en un entorno 3D.

El jugador debe recorrer un camino evitando caer o colisionar con obstáculos, recogiendo la mayor cantidad posible de monedas antes de llegar a la meta. Cada una de estas le brindará la capacidad de destruir los objetos con los cuales colisione posteriormente.

Está pensado para trabajar el equilibrio de los usuarios, obligando a los mismos a distribuir su peso sobre la tabla, apoyando y realizando fuerzas de distinta intensidad en partes específicas del pie.



Figura 14

2do: ¡El cielo se cae!

En este nivel se suma la utilización de la visión artificial, por lo que el protagonista es un humanoide que copia en tiempo real la postura del jugador. El juego está ambientado en 3 dimensiones, sin embargo, el personaje solo puede realizar movimientos laterales en lo que llamaremos “recta de movimiento”.

Cada cierto periodo, aparecen objetos en el plano vertical que contiene a la recta de movimiento, algunos son objetivos y otros son obstáculos. Para ganar el jugador debe agarrar los objetivos indicados en pantalla, evitando chocar con los obstáculos que disminuyen la vida del personaje, cuando la barra de vida llegue a 0 el jugador perderá el nivel.

Aquí se busca familiarizar al jugador con la tecnología de detección de pose, su sensibilidad y velocidad de respuesta. A su vez se agrega la dificultad de moverse en el plano haciendo uso del equilibrio por medio del balance board.



Figura 15

3ro: Ruta 40

En este nivel el grupo de trabajo consideró necesario mejorar la inmersión del usuario en el juego, brindándole la capacidad de moverse en el espacio por medio de la balance board. Haciendo uso esta vez la posición del centro de gravedad en los cuatro cuadrantes del plano de la tabla.

A lo largo de la partida el jugador debe ir recogiendo las diferentes monedas que se encuentran distribuida en la pista. La única forma de que puedan ser capturadas es si el humanoide las colisiona con manos, pies o cabeza. La complejidad radica en que tiene solo 300s para recoger las 10 monedas, de no lograrlo perderá el nivel. Para esta aventura cuenta con diferentes beneficios en forma de signos de pregunta, los cuales les permitirán aumentar su velocidad de traslación y llegar más rápido a recoger las todas monedas.

Con este juego se busca captar por completo la atención y postura del niño, logrando que el mismo el apoye toda la planta del pie, objetivo fundamental en terapia.

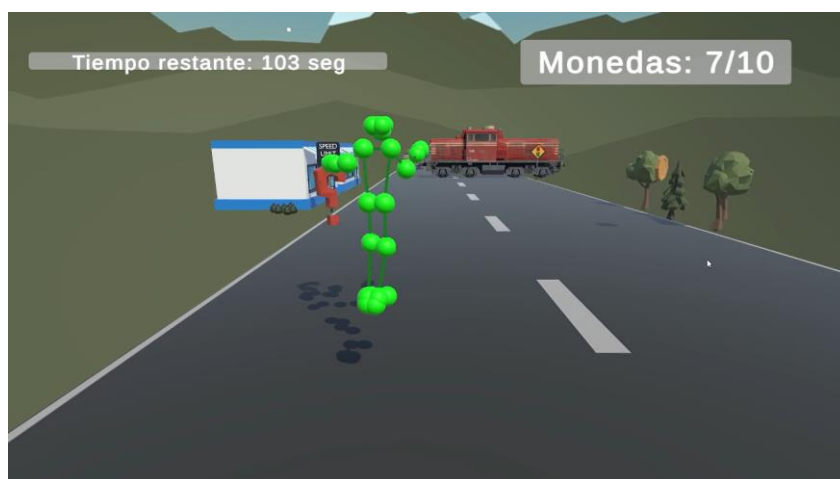


Figura 16

Conclusión

A lo largo del trabajo hemos encontrado ciertas dificultades, ya que no traíamos experiencia previa con software de realidad virtual ni de visión artificial.

Inicialmente el proyecto se veía ambicioso, y de hecho fue un gran desafío, pero gracias a un excelente trabajo colaborativo se pudo llevar adelante. Comenzamos familiarizándonos con la tecnología Unity, y de manera paralela cada uno profundizó en una característica específica del juego.

El resultado final nos ha dejado una sensación muy grata, ya que hemos superado nuestras expectativas. El juego funciona de forma fluida, esto debido a que tuvimos el tiempo necesario para reparar cualquier bug que se generase y así salvar cualquier problemática.

Queda pendiente la exportación y el tratamiento de los datos provenientes de la visión artificial, esta sería una gran ayuda para el profesional que está evaluando la postura del paciente, ya que podría obtenerse por ejemplo un promedio del ángulo de apertura de las rodillas.

Bibliografía

- AVIA SEMICONDUCTOR. (s. f.). Hx711 datasheet [PDF]. https://cdn.sparkfun.com/datasheets/Sensors/ForceFlex/hx711_english.pdf
- Bogde, B. (s. f.). GitHub - bogde/HX711: An Arduino library to interface the Avia Semiconductor HX711 24-Bit Analog-to-Digital Converter (ADC) for Weight Scales. GitHub. <https://github.com/bogde/HX711>
- Cvzone, C. (s. f.). GitHub - cvzone/cvzone: This is a Computer vision package that makes its easy to run Image processing and AI functions. At the core it uses OpenCV and Mediapipe libraries. GitHub. <https://github.com/cvzone/cvzone>
- Encefalopatía traumática crónica. (2022, 19 octubre). Middlesex Health. <https://middlesexhealth.org/learning-center/espanol/enfermedades-y-afecciones/encefalopat-a-traum-tica-cr-nica>
- Peek, B. (s. f.). GitHub - BrianPeek/WiimoteLib: A library for using a Nintendo Wii Remote (Wiimote) from .NET. GitHub. Recuperado 8 de abril de 2022, de <https://github.com/BrianPeek/WiimoteLib/>
- UNITY FORUM. (s. f.). de <https://forum.unity.com/>
- Digital-Standard. (s. f.). GitHub - digital-standard/ThreeDPoseUnityBarracuda: Unity sample of 3D pose estimation using Barracuda. GitHub. <https://github.com/digital-standard/ThreeDPoseUnityBarracuda>
- Lshachar. (s. f.). Lshachar/WiiBalanceWalker. GitHub. <https://github.com/Lshachar/WiiBalanceWalker/releases>
- Unity. (s. f.). Documentación Unity. <https://docs.unity.com/>
- Wagner. (s. f.). C# docs - get started, tutorials, reference. Microsoft Learn. <https://learn.microsoft.com/en-us/dotnet/csharp/>
- Wikipedia contributors. (2022, 26 septiembre). Load cell. Wikipedia. https://en.wikipedia.org/wiki/Load_cell

Repositorio

https://github.com/francoleyes/fuera_de_realidad

Link video demostrativo: <https://youtu.be/Ybh1e2eE0xI>