1. INTRODUCCIÓN

La realización de estas prácticas se ha llevado a cabo en el Centro de Investigación en Tecnologías Software y Sistemas Multimedia para la Sostenibilidad (CITSEM) colaborando como investigador en el grupo GAMMA. Este centro, sigue una serie de objetivos que se centran en las tecnologías en un contexto de sostenibilidad.

El proyecto en el que he sido participante se basa en el diseño y desarrollo de juegos serios para la rehabilitación de personas con problemas de movilidad. En concreto, se trata del desarrollo del videojuego "*Phiby's Adventures*", iniciado en el 2019 y llevado a cabo por múltiples alumnos que han realizado sus prácticas externas o PFG para su puesta en funcionamiento. Este, es un juego dirigido para niños de diversas edades implementado en Unity y controlado a través de cámaras Kinect.

La historia se centra en *Phiby*, un dragón que, en su inicio, es encerrado en una jaula por su hermano más ambicioso, *Ra*, y debe superar diversos obstáculos para salvar el mundo de su control. El juego se distribuye en una isla donde van desencadenando diversos videojuegos donde el jugador debe realizar una serie de movimientos para superarlos.

Entre las funciones que he tenido que realizar, la principal ha sido la modificación del código del juego para su funcionamiento con un nuevo middleware. Este middleware fue implementado inicialmente por el alumno Cesar Luances en 2019 y entre septiembre de 2022 a junio de 2023, Filip Racki realizó una remodelación de este para que pudiese haber la posibilidad de intercambiar tipos de cámaras. Además, se han podido subsanar errores externos a esta implementación y mejorar la implementación de alguno de los minijuegos que lo componen. Para ello, ha sido necesario un estudio interno de alguno de los componentes que engloban la interfaz del juego, así como su código interno (lenguaje C#).

Esta interfaz se compone de un ordenador, donde se realizan todas las operaciones de funcionamiento del programa; una cámara Kinect v2 (Xbox One) o cámara Azure, la cual recoge los movimientos realizados por el jugador; el middleware K2UM, el cual sirve de intermediario para la comunicación entre las cámaras y el videojuego; y un servidor web, el cual sirve para que los terapeutas metan los parámetros de cada minijuego adaptados a cada paciente.

A través de esta memoria, se explica con más profundidad la situación actual del CITSEM y sus finalidades, los objetivos que se me impusieron, explicando los problemas y soluciones establecidas y las tecnologías y medios utilizados para su puesta en funcionamiento, entre otros.

2. INFORMACIÓN DE LA ENTIDAD COLABORADORA

El organismo para el cual se han llevado a cabo estas prácticas presenciales es el CITSEM (Centro de Investigación en Tecnologías Software y Sistemas Multimedia). Este centro se creó en 2011 con el fin de fortalecer y promover la I+D+i y la captación de talento investigador en el Campus Sur de la UPM, donde está situado. Sus principales objetivos se orientan hacia el aprovechamiento de las tecnologías software y multimedia en un contexto de sostenibilidad, siendo estos:

- Valorización y transferencia de los resultados de sus trabajos a entidades públicas y privadas del sector tecnológico.
- Realización de trabajos específicos, informes o labores de asesoría para empresas privadas o instituciones públicas.

- Colaboración con entidades (públicas o privadas) en la realización de proyectos de I+D+i que contribuyan al logro de sus objetivos empresariales o de servicio público.
- Difusión de los resultados de investigación alcanzados con el fin de incrementar el conocimiento científico.
- Organización de seminarios, congresos y conferencias y docencia de posgrado y doctorado en el ámbito de sus líneas de trabajo

En la actualidad, el centro está dividido en tres grupos de investigación: Grupo de Diseño Electrónico y Microelectrónico (GDEM), Grupo de Aplicaciones Multimedia y Acústica (GAMMA) y el Grupo de Tecnología Software y Sistemas (SYST).

Para fomentar el comienzo en la investigación, el CITSEM cada semestre oferta prácticas en los diversos departamentos. En este caso, se han llevado a cabo a través del grupo GAMMA, centrando la investigación en la gamificación y el desarrollo de interfaces inteligentes para personas con discapacidad física, mejorando la sensación de inversión en videojuegos serios.

Éste, se creó en 2015 y cuenta con investigadores especializados en ámbitos tan diversos como el procesado de señal, la acústica y la gestión de empresas y la lingüística aplicada a la ciencia y la tecnología. Además de la línea de investigación ya mencionada, también están envueltos en muchas otras, siendo algunas de ellas el procesado de imágenes médicas, la acústica virtual o el procesado de la señal de voz.

Para desarrollar las prácticas, el centro cuenta con despachos de trabajo equipados con todos los recursos necesarios, así como ordenadores y sus periféricos, cámaras, conexiones a redes wifi etc. Además, se me proporcionó una cámara para poder seguir trabajando de forma remota.

3. MARCO DE LA ACTIVIDAD REALIZADA EN EL CONTEXTO DE LA ENTIDAD

Tal como se ha comentado previamente, el proyecto se ha llevado a cabo en el departamento GAMMA del CITSEM, basándose en el diseño y desarrollo de juegos serios para personas con problemas de movilidad, siendo en este caso el proyecto "*Phiby's Adventures*". Éste, es un pequeño videojuego integrado en Unity utilizando C# como lenguaje de código y cámaras Kinect v2 para el seguimiento de movimientos. Para que el juego funcione, es necesario la utilización del Middleware K2UM (Kinect to Unity Middleware), inicialmente desarrollado por César Luances y remodelado para el uso de cámaras Azure por Filip Racki. Todos los equipos necesarios para su implementación se han tenido disponibles en el despacho dedicado al proyecto.

La principal responsabilidad en este proyecto ha sido la adaptación del videojuego para que pudiese funcionar con la nueva versión del Middleware. Esto, engloba funciones como depuración y escritura de código o estudio de estructuras en Unity. De forma adicional, si se cumplía con lo propuesto inicialmente, el siguiente paso era proseguir con el desarrollo del proyecto y de sus diferentes minijuegos internos.

Como en el proyecto no están estrictamente establecidos los cargos y puestos de trabajo, las relaciones que se han mostrado han sido entre compañeros de despacho que trabajan en proyectos similares y con antiguos alumnos para resolución de problemas.

4. OBJETIVOS DE LAS PRÁCTICAS, TAREAS Y ACTIVIDADES REALIZADAS

En un inicio se me fijaron varios objetivos, siendo estos la adaptación del juego al nuevo middleware y la creación y mejora de funcionamiento de los minijuegos. El principal y más importante es el primero, ya que debía funcionar con las nuevas cámaras para poder seguir progresando en el juego.

Adaptación del videojuego al nuevo middleware K2UM

Como ya se ha mencionado con anterioridad, el alumno Filip Racki remodeló el antiguo middleware creado por Cesar Luances para que en la actualidad hubiera la posibilidad de jugar con cámaras Kinect de la Xbox One y la cámara Azure de Google. Para su adecuado funcionamiento, construyó clases nuevas para recoger los datos enviados por K2UM y poder sustituir alguna de las antiguas. Para poder empezar con las adaptaciones, el primer paso fue entender la dinámica del juego y su estructura.

Familiarización con entorno

Este juego es bastante extenso, por lo que no todas las partes de él debían ser modificadas para su nuevo funcionamiento. El juego está controlado inicialmente por una pantalla de menú en la que se puede acceder a la aventura general, los diferentes minijuegos que lo engloban, una calibración para ajustar el esqueleto de cada jugador y otros como créditos, checkpoints…

A parte, el juego en Unity se estructura a través de una serie de carpetas para que la búsqueda de los elementos sea más fácil y rápida.

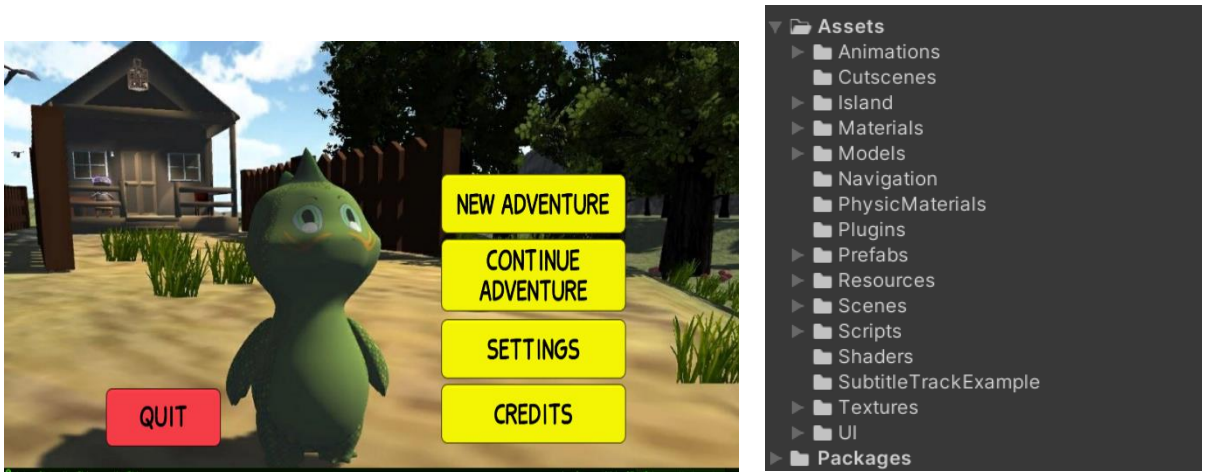

Fig. 1: Menú del videojuego "Phiby's Adventures" (izq.); estructura interna del juego en Unity (dcha.)

Para que la depuración del juego fuera más fácil, se comenzó estudiando una escena de uno de los minijuegos, en concreto el minijuego de escalar árboles.

Cada escena cuenta con el jugador, en este caso *Phiby*, el cual se compone de dos partes: *PhibyV1*, que se corresponde con la malla y esqueleto en si del personaje; y *KinectReceiver3.0_rotacionesiniciales*, que es lo que llamaremos *KinectAsset* y que se corresponde con las referencias del jugador en la Kinect. Esta parte era la que era necesario modificar para que pudiese funcionar con el nuevo middleware.

Fig. 2: Estructura en Unity del videojuego ClimbAppleTree

Para comprender que había que modificar, fue necesario estudiar la estructura de una escena modelo que realizó el creador del nuevo middleware en la que solo se mostraba a phiby y sus movimientos. La principal diferencia se hallaba en el objeto *kinectReceiver*, en donde se había suprimido completamente los antiguo scripts `Rotation.cs` y `launcher.cs` por uno nuevo llamado `kinectReceiver.cs`. Además, los scripts Con esto entendido, se comprendió que el primer paso a realizar era modificar el *KinectAsset* (*KinectReceiver3.0_rotacionesiniciales*) del antiguo juego con la estructura del nuevo. Para ello, se creó una nueva escena de prueba en la que se introdujo solo al personaje y se trasladó la estructura propuesta por Filip cambiando los nombres a como se muestran en el juego base.

Fig.3: estructura en Unity de la escena modelo de Filip (izq.); estructura en Unity de la escena de prueba (dcha.);

Entendido esto, el siguiente paso fue adaptar el juego grande.

Adaptación de Phiby's adventures

Para empezar, se trasladó lo creado en la escena de prueba al minijuego de escalar árboles, solucionando los errores que mostraba Unity. Era necesario borrar completamente los scripts ya mencionados (`rotation.cs` y `launcher.cs`), para implementar el nuevo (`kinectReceiver.cs`). Como muchos otros dependían de estos, surgieron cuatro tipos de errores:

- Error en la obtención de los valores x e y con la función `transform()`: había cambiado la estructura del esqueleto y el diccionario de articulaciones. Se solucionó cambiando la expresión a

la estructura adecuada.

- Error en función no encontrada `finishAndDestroy()`: en el nuevo script no existía esta función, por lo que se añadió, ya que es bastante importante a la hora de destruir el *kinectAsset* al pasar de una escena a otra.
- Error en función no encontrada `moveAllow()`: en el nuevo script no existía esta función, porque no era necesaria ya que tenía un funcionamiento algo retórico porque que se eliminó su aparición en otros scripts.
- Error en la obtención de la configuración del usuario: se solucionó tras una reunión con Filip, en donde se aclaró que se estaba utilizando una versión antigua del middleware. Los otros

Tras solucionar estos errores, apareció el error que más tiempo llevó, ya que fue algo difícil de encontrar.

Está relacionado con la recepción de los datos del usuario que inicia sesión en el middleware.

Error con configuración de usuario

Cuando un usuario inicia sesión en la aplicación, esta coge los datos de una página web y los traslada al juego. El problema se encontraba en que cuando se iniciaba sesión y se iniciaba el juego, éste mostraba que no había ningún inicio de sesión. En un principio se pensó que no estaba habiendo comunicación entre el middleware y el juego. Para comprobar si esto era cierto, se utilizó la aplicación Wireshark para visualizar los mensajes que se compartían. Fue necesario estudiar el funcionamiento de los puertos udp.

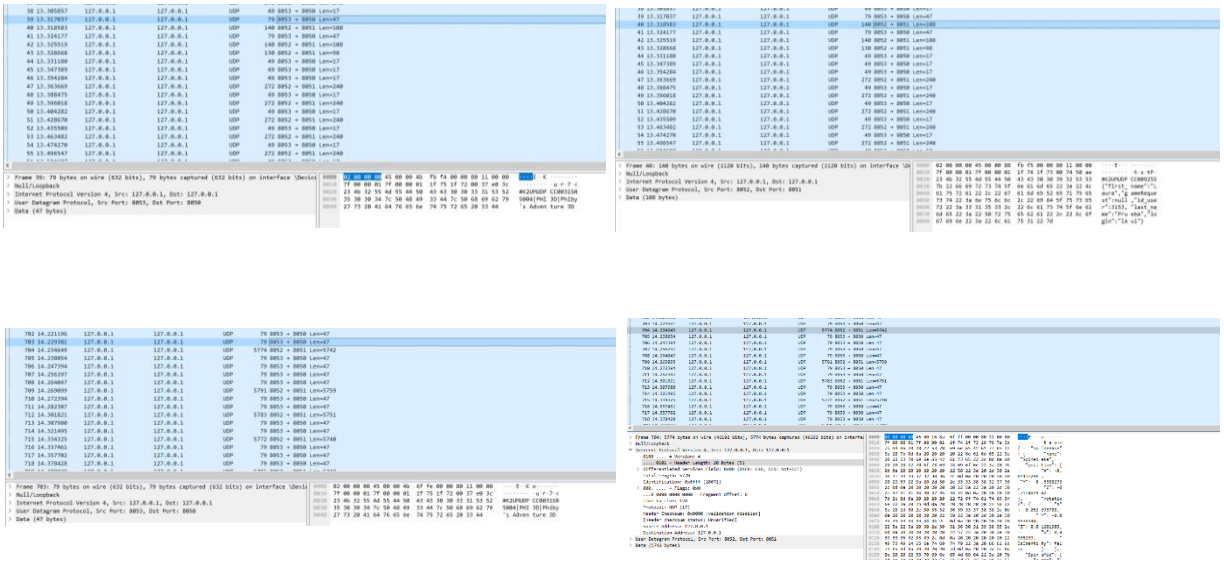

Fig.4: capturas de Wireshark del mensaje Setting Request versión Antigua (arriba izq.) y su respuesta en versión antigua (arriba dcha.); mensaje Setting Request versión nueva (abajo izq.) y su respuesta en versión nueva (arriba dcha.).

Se comprobó que, en la versión antigua del juego, este mandaba una petición de configuración (*Setting Request*) y el middleware la mandaba, además del envío de la rotación de todas las articulaciones. En el nuevo, se visualizó que, sí que había comunicación, ya que se recibían los valores de las rotaciones, pero cuando se mandaba la petición de configuración, el middleware no la mandaba. Es por esto por lo que se pensó que el problema estaba en el middleware.

Tras esto, se estudió el funcionamiento interno del middleware, incluyendo otra reunión con Filip para mejor entendimiento. Finalmente se encontró que el error residía en que la aplicación no encontraba la referencia del juego por un error gramatical (alumnos antiguos).

Tras comprobar que, si se mandaba la configuración, era necesario diferenciar entre datos de articulaciones y datos de usuario a través del código.

Otros errores

Con la estructura del juego y los scripts adaptados, se procedió a arreglar nuevos errores que surgían independientemente de la versión. El principal residía al pasar de escenas en el menú, el cual se solucionó modificando la posición de la función `finishAndDestroy()` en el script `kinectGamesManager.cs`.

Minijuegos

Al comprobar que todo funcionaba como lo establecido y no aparecían errores, se procedió a seguir con el desarrollo de los minijuegos, en concreto se prosiguió con el minijuego de cortar troncos. Como era algo antigua la versión que había de este minijuego, se adaptó a la estructura actual que muestran los demás, eliminando scripts que estaban obsoletos.

Fig. 5: Estructura en Unity del videojuego Chop Wood antiguo (izq.); Estructura en Unity del videojuego Chop Wood nuevo (dcha.);

Los principales fueron `EvntMng.cs` que ya no se usaba, y `ChopMovementControler.cs`, ya que la función de este script está contenida en `movementControlles.cs`. Fue necesario cambiar y añadir código de este script y otros como `kinectGamesManager.cs`. El siguiente paso por realizar sería arreglar el algoritmo de suma de los tablones cortados para arreglar el puente, ya que por falta de tiempo no se cuentan cómo está decretado en el GDD.

Además de esto, se añadió algo de vegetación al entorno. No se pudo añadir gran cantidad porque al hacerlo la carga del juego era muy grande y los movimientos del personaje eran muy bruscos.

5. TECNOLOGÍAS Y MEDIOS TÉCNICOS UTILIZADOS

Para llevar a cabo este proyecto se han utilizado elementos tanto como físico como de software.

Como núcleo central de control se encuentra el ordenador y todo el equipamiento relativo a su funcionamiento (teclado, ratón, altavoces). Éste, es lo suficientemente potente para manejar gráficos y con alta memoria para que los programas dentro de él funcionen con fluidez.

Además, como el proyecto se basa en la captura de movimientos físicos para su posterior

reproducción en el videojuego, se ha utilizado una cámara Kinect v2 (Xbox One). Es utilizada para la captura del esqueleto del jugador a través de un conjunto de cámaras y sensores de infrarrojos.

En cuanto al equipamiento de software se han utilizado varios programas:

- **Unity**: Motor principal para el desarrollo del videojuego. La estructura principal del juego se encuentra recogido en este programa, en donde se adaptó el KinectAsset al nuevo middleware.
- **Visual Studio**: Herramienta de programación que se utilizó para adaptar el código del juego para la recepción de mensajes procedentes del middleware y después depuración de errores y creación de nuevas funcionalidades. Éste, se utiliza en toda la parte de codificado del videojuego, utilizando el lenguaje C#.
- **K2UM**: Middleware remodelado por Filip Racki basándose en la versión inicial de alumnos anteriores. En esta nueva versión, se permite la utilización de dos nuevas cámaras (Kinect v2 (Xbox One) y Azure), cuando en su versión inicial solo era posible utilizar una cámara, ya se la antigua (Kinect v2(Xbox 360), obsoleta actualmente en este proyecto) o la de la One. Como descripción general de la aplicación, trata del software que permite la comunicación entre las cámaras, el motor de desarrollo del juego y la web.
- **Wireshark**: Analizador de protocolos utilizado para observar la comunicación entre el videojuego y el middleware. Entre estos, se envían y reciben paquetes UDP a través de unos puertos específicos, por lo que se observó lo que se recibía para la detección de errores en el código del juego.
- **Antiguos proyectos de Phiby y otros**: Se utilizaron proyectos de Phiby antiguos para observar la implementación de ciertas funciones y otros proyectos más simples para ver la estructura de los elementos en Unity.

En cuanto a documentación, principalmente se han utilizado PFG y PE de alumnos anteriores para entender al principio el funcionamiento del juego. Además, se usaron tutoriales de youtube para ciertos errores.

6. COMPETENCIAS Y HABILIDADES ADQUIRIDAS CON LAS PRÁCTICAS

Las competencias y habilidades que se han adquirido a través de las prácticas externas son:

- **Capacidad de organización, planificación y de toma de decisiones.**

  Para llevar a cabo este proyecto de forma satisfactoria se han tenido que definir los objetivos, analizarlos y organizar que orden se debía seguir para el correcto funcionamiento del código. Con esto, la planificación a lo largo de las semanas ha sido un punto clave en la resolución de problemas.

- **Capacidad para aprender de manera autónoma nuevos conocimientos y técnicas adecuados para la concepción, el desarrollo o la explotación de sistemas y servicios de telecomunicación.**

  Todos los cambios que los alumnos han ido realizando y el por qué en numerosas ocasiones no quedaban redactados en ningún informe, por lo que ha sido necesario investigar las técnicas y funciones utilizadas para entender el funcionamiento del juego y poder seguir adelante con ello.

- **Capacidad de realizar programación en tiempo real, concurrente, distribuida y basada en eventos, así como el diseño de interfaces persona-computador.**

  Mi función principal ha sido adecuar el código del juego al nuevo middleware, por lo que he tenido que trabajar continuamente con scripts de código.

- **Habilidades para la utilización de las Tecnologías de la Información y las Comunicaciones.**

  Se han utilizado equipos relacionadas con estas tecnologías.

- **Capacidad de abstracción, de análisis y de síntesis y de resolución de problemas.**

  A lo largo de las prácticas surgían nuevos errores de código y en unity que a través de estudiarlos y buscar información se han sabido resolver.

7. CONCLUSIONES

La realización de estas prácticas me ha favorecido tanto personal como académicamente. He podido aprender nuevas técnicas y programas y fortalecer mis conocimientos en programación y manejo de estructuras de Unity, además de aplicar conceptos estudiados a lo largo de la carrera. Además, he aprendido a planear las tareas a lo largo de la semana de una forma más ordenada, así como a ser autodidacta en algunos aspectos. Otro de los puntos clave que he podido desarrollar es la constancia y el no rendirse a pesar de que los errores encontrados a veces han sido bastante duros de solucionar.

Los objetivos que se me impusieron han sido completados con éxito, no pudiendo extender mucho más el juego por falta de tiempo. Además, se ha adquirido la capacidad de poder explicar a antiguos alumnos y conjuntamente intentar resolver los errores más importantes.

En relación con mi tutora de prácticas, Martina, he estado muy a gusto durante el periodo que han durado mis prácticas. Me brindó la posibilidad de realizar mis horas de la manera que yo prefiriese, siendo bastante flexible. Además, me puso en contacto con alumnos que habían trabajado con anterioridad en este proyecto para que me pudieran explicar el funcionamiento de ciertas partes de código. Por otro lado, he estado en contacto con ella prácticamente de continuo, ya que dos días a la semana se presentaba en el despacho y la contase cual era mi progreso.

Además de todo lo aprendido durante la realización de estas, me ha abierto la puerta a realizar un trabajo de fin de carrera en el que pueda plasmar estos conocimientos y culminar mis estudios de una forma comprometedora.

8. DIARIO DE PRÁCTICAS

Semana 1 (2-6 Oct.)

- Primera toma de contacto con el proyecto
- Entender principales funcionalidades de este y del modelo realizado por Filip Racki
- Primera puesta en funcionamiento y análisis de los principales errores

Semana 2 (9-13 Oct.)

- Entender la sustitución de los principales scripts
- Cambio de la estructura del *KinectAsset* del *Phiby*
- Sustitución de funciones que con el nuevo middleware no son necesarias
- Entender como es la nueva estructura del esqueleto del jugador

Semana 3 (16-20 Oct.)

- Seguir adaptando el código para la recepción de los datos del nuevo middleware.
- Primer adelanto, *Phiby* realiza los movimientos correctos.
- Error con la configuración del juego

Semana 4 (23-27 Oct.)

- Reunión con el autor del nuevo middleware (Filip Racki) (Se estaba utilizando una versión errónea del middleware)
- Primer adelanto, *Phiby* realiza los movimientos correctos.
- Error con configuración resuelto
- Nuevo error en la recepción de parámetros del usuario.

Semana 5(30 Oct.-3 Nov.)

- Error en la recepción de parámetros del usuario.
- Estudio paso a paso de ciertas funciones

Semana 6 (6-10 Nov.)

- Error en la recepción de parámetros del usuario.
- Reunión con antiguo alumno con PFG basado en el juego (Juan Fuentes-Pila Martín)
- Análisis del código interno del middleware
- Uso de Wireshark para visualizar los paquetes enviados y recibidos
- Reunión con Filip Racki (entender las funciones que gestionan el envío y recepción de los parámetros del usuario)

Semana 7 (13-17 Nov.)

- Entender uso de los puertos de entrada y salida para la recepción de información
- Uso de Wireshark y analizar que mensajes se mandan, su contenido y cada cuanto.
- Arreglo del error en la recepción de parámetros del usuario (nombre del juego se estaba mandando mal)

Semana 8 (20-24 Nov.)

- Adaptar script `KinectReceiver.cs` para la distinción entre información de usuario y datos de las articulaciones.
- Aumentar tiempo de espera entre inicio y recepción de información
- Error cuando desde el menú entra en una escena y sale con escape (se bloquea)

Semana 9 (27Nov.-1 Dic.)

- Arreglo de error de cambio entre escenas (función `finishAndDestroy()` no está bien colocada).
- Modificación de funciones de cambio de escenas

Semana 10 (4-5 Dic.)

- Optimización de terreno
- Vegetación añadida

Semana 11 (11-15 Dic.)

- Adaptación estructura de minijuego (*Chop Wood*) a nuevo middleware
- Modificación de scripts relacionados con este para su correcto funcionamiento.

Semana 12 (18-22 Dic.)

- Añadir botón de juego *Chop Wood* en sección de minijuegos del menú
- Actualización de GDD