



MEMORIA DE PRÁCTICA EXTERNA
ETS DE INGENIERÍA Y SISTEMAS DE
TELECOMUNICACIÓN
UPM

Laura Caballero Ruiz



ÍNDICE DE CONTENIDOS

DATOS IDENTIFICATIVOS DE LAS PRÁCTICAS EXTERNAS:.....	1
1. INTRODUCCIÓN:.....	2
2. OBJETIVOS DE LAS PRÁCTICAS, TAREAS Y ACTIVIDADES REALIZADAS	3
3. EVOLUCIÓN CRONOLÓGICA DE LAS ACTIVIDADES	7
4. TECNOLOGÍAS Y MEDIOS TÉCNICOS UTILIZADOS	7
5. COMPETENCIAS Y HABILIDADES ADQUIRIDAS CON LAS PRÁCTICAS	8
6. CONCLUSIONES	9

INSTRUCCIONES:

Se recomienda completar una cara por cada uno de los epígrafes a desarrollar, a excepción del nº2, "Objetivos de las prácticas...", que debe tener mayor contenido (mínimo dos caras)

1. INTRODUCCIÓN:

El Centro de Investigación en Tecnologías Software y Sistemas Multimedia para la Sostenibilidad (CITSEM) 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. Este centro se encuentra en el edificio La Arboleda de la Escuela de Ingeniería y Sistemas de Telecomunicación.

Cada semestre el CITSEM oferta diversas plazas de prácticas externas de iniciación a la investigación en diversos departamentos. En mi caso, el grupo de investigación es GAMMA, Grupo de Aplicaciones Multimedia y Acústica, y mi línea de investigación se centra en el diseño y desarrollo de juegos serios para su aplicación en entornos educativos.

Phiby's Aventures v2 es un videojuego serio diseñado para que, mediante la cámara Kinect, personas con debilidad muscular o discapacidad puedan hacer rehabilitación de forma atractiva viviendo las aventuras de un personaje. De esta forma, el paciente no identifica el uso del juego como una terapia para su lesión sino como algo divertido.

El juego tiene lugar en una isla en la que el jugador vive las aventuras de Phiby. Gracias a la Kinect, podrá moverse para explorar el terreno en el que tendrá que realizar múltiples ejercicios físicos en forma de mini juegos que le ayudarán a avanzar en la historia.

El objetivo de mis prácticas han sido mejorar la versión del juego Phiby's Aventures v2. Para ello, se han corregido errores anteriores, se han reestructurado en gran medida las carpetas del juego y además se ha introducido contenido nuevo enriqueciendo la experiencia del jugador. Para realizar las tareas, ha sido necesario profundizar en las herramientas que ofrece Unity y Blender, softwares que ya se conocían gracias a la asignatura Síntesis y Animación de Imágenes.

Durante estos meses, he formado equipo con dos compañeros, Dario Cavero Cabrera, alumno en prácticas al igual que yo y Henar Redondo Martín, alumna becada por el Ministerio. Debido a la situación sanitaria, hemos trabajado sobre todo desde casa. Esto nos ha permitido trabajar en diferentes horarios de forma autónoma, pero además que en múltiples ocasiones hayamos podido trabajar juntos vía Teams, ayudándonos en las tareas más difíciles. Además, nos hemos reunido todos los lunes con Martina Eckert, nuestra tutora profesional, encargada de coordinar las tareas a realizar, para contarle los avances de cada semana.

En los siguientes capítulos se explica con más detalle los objetivos a seguir, las tareas y actividades realizadas, así como las tecnologías y medios utilizados. Además, se exponen las competencias y habilidades adquiridas durante el desarrollo de las prácticas. Por otro lado, se describen las posibles mejoras para aquellos que sigan con el proyecto en el futuro.

2. OBJETIVOS DE LAS PRÁCTICAS, TAREAS Y ACTIVIDADES REALIZADAS

La creación del videojuego es un proyecto a largo plazo por lo que para estas prácticas se han fijado diversos objetivos.

El objetivo principal ha sido mejorar la versión del videojuego actual, “Phiby’s Adventures v2”, basado en captura de movimientos vía la cámara Kinect, que se encuentra en desarrollo. Para conseguir esto, se ha elaborado una lista de errores y de mejoras de jugabilidad.

Después de una fase de entender el proyecto y su contenido, se han comenzado a realizar dichas tareas, detalladas a continuación.

- **Estructurar de forma óptima las carpetas que contienen todo lo necesario para jugar:** en la versión del juego de principio del semestre, cada objeto de la isla tiene su propia carpeta en la cual tiene sus materiales y sus texturas por lo que hay multitud de carpetas de texturas y materiales. Esto provoca que, sin darse cuenta, haya la misma textura o material muchas veces. Se ha decidido crear una única carpeta de texturas y materiales en las cuáles estén todas las existentes, clasificadas por zonas de la isla, por ejemplo, jaula, bosque, montaña... De esta forma es mucho más rápido y eficiente encontrar lo que se necesita y verificar si, por ejemplo, una textura ya está en el juego y evitar importarla de nuevo.

Al igual que para las texturas y materiales, se ha realizado para los modelos; clasificándolos por zonas en la isla, los scripts; clasificados por su funcionalidad y las animaciones; clasificándolas según el personaje que la contenga.

En la siguiente figura, se muestra un ejemplo de esta estructura, seleccionando la carpeta de materiales. Tal y como se observa esta se divide en múltiples subcarpetas, correspondientes a diferentes zonas de la isla o a personajes. Está seleccionada la carpeta “characters” en la que a su vez está dividida en los distintos personajes que hay en el juego: personaje principal y personaje no jugador (animales, abuela...).

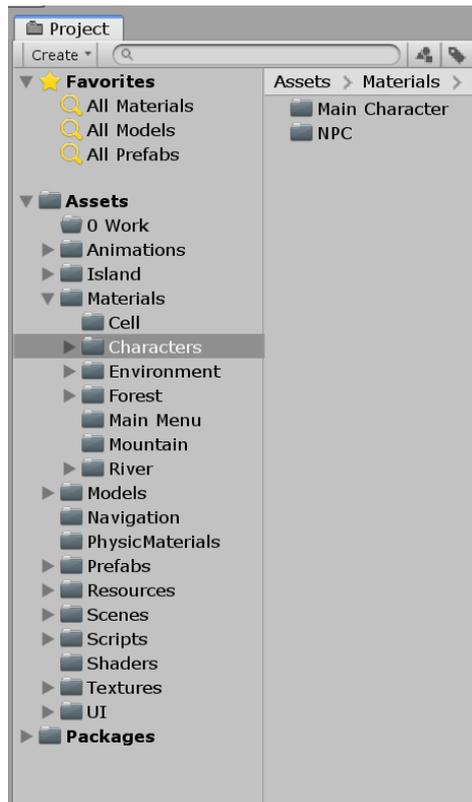


Figura 1. Estructura del proyecto

- **Estructurar de forma óptima la jerarquía del juego:** en una primera versión proporcionada al alumno, todo lo existente se clasifica según si es estático o interactivo, lo que es bastante positivo pero ineficiente pues causa una lista interminable de objetos en la que resulta difícil encontrar el deseado. Por ello, se opta por estructurar los objetos existentes según la zona de la isla en la que se encuentren y a su vez en estáticos o dinámicos.

En la siguiente figura vemos un ejemplo de aquellos objetos que se sitúan en la zona del bosque. Tal y como se ha mencionado, hay objetos interactivos como pueden ser los champiñones que dan energía al personaje y objetos estáticos que simplemente son decoración como las señales o los vegetales.

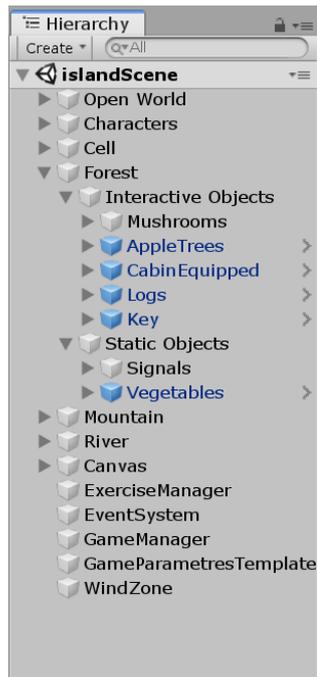


Figura 2. Estructura de los objetos en la escena

- **Mejorar la apariencia de la isla** realizando nuevos caminos, modelando objetos o reubicando algunos ya existentes.
 - **Modelado de señales.** Se han modelado tres señales distintas con ayuda del software Blender, las cuáles han sido diseñadas dándole distinta forma a diversos cubos. Estas señales permiten que cuando hay una bifurcación de caminos, el personaje sepa dónde se encuentran los lugares a los que quiere ir.



Figura 3. Ejemplo de señal modelada

- **Pájaros volando en círculo:** se ha creado un script llamado “Birds” para que los pájaros existentes no estén parados en el cielo, sino que hagan un movimiento circular lo suficientemente grande para simular el vuelo.
- **Creación de caminos:** se han creado dos caminos nuevos con el fin de aprovechar en mayor medida la isla, mejorar su apariencia y proporcionar al jugador más zonas por donde poder explorar. Además, se han suavizado algunas colinas que resultaban muy abruptas en las que la animación del personaje andando resultaba irreal. En las siguientes figuras se muestra como estaba en un primer momento y la última versión.



Figura 4. Primera versión de la isla

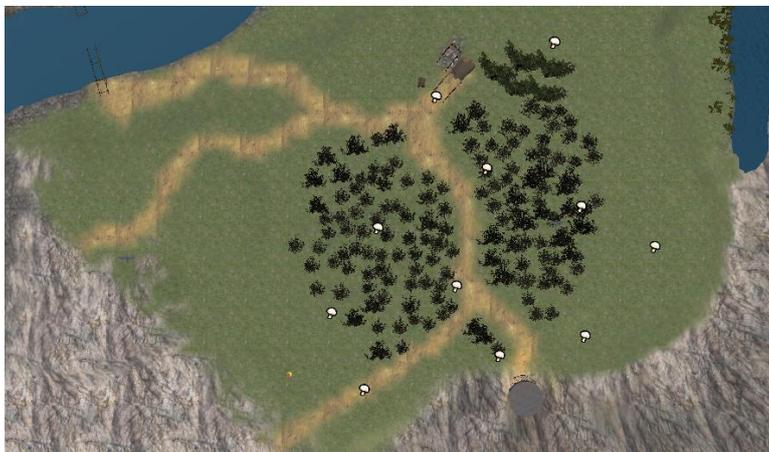


Figura 5. Última versión de la isla

- **Mejoras del juego** que favorecen la mejor experiencia del jugador:
 - **Llave más grande y con movimiento** para encontrarla. Phiby tiene que encontrar la llave de la cabaña para dejar las manzanas que recolecte. Sin embargo, este no sabe que tiene que buscarla hasta que no habla con su abuela. En un primer momento, la llave es pequeña y está en el suelo del bosque (Figura 6 izquierda), por lo que es difícil encontrarla, aunque no imposible y por ello, Phiby puede encontrarla si merodea por el bosque antes de hablar con su abuela. En caso contrario, se ha creado un script llamado “KeyController” para que cuando el personaje hable con su abuela, en forma de pista, esta llave aumente de tamaño y tenga un movimiento de rotación mejorando la visión de la misma (Figura 6 derecha).

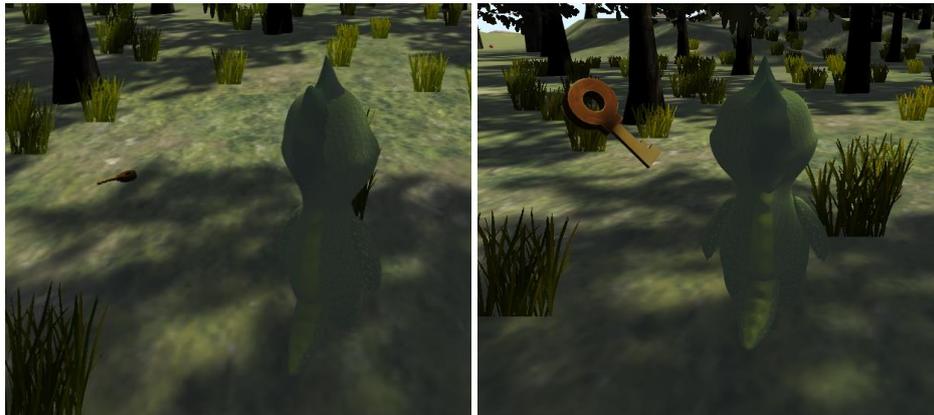


Figura 6. Llave al principio (izquierda) y después de que Phiby hable con la abuela (derecha)

- **Nuevo funcionamiento de Rocky.** En las primeras versiones, Phiby necesita la súper fuerza de Rocky, otro personaje, para realizar un mini juego y, por tanto, tiene que buscarle. Rocky está atrapado entre dos rocas y Phiby le ayuda a salir de allí para que le siga hasta el mini juego.



Figura 7. Rocky en la primera versión atrapado entre dos rocas

Con el fin de mejorar la experiencia del jugador esta parte del juego ha sido modificada con mis ideas y las de mis compañeros de grupo. En primer lugar, como se habla de la súper fuerza de Rocky, se ha creado un halo de color amarillo para visualizarla. No tiene sentido que si Rocky tiene súper fuerza esté atrapado entre dos rocas y por ello, en la nueva versión ocurre lo siguiente. Rocky se ha quedado atrapado, pero no entre dos rocas sino en la cueva porque ha perdido su súper fuerza en el bosque (Figura 8). De esta forma, se aprovecha en mayor medida el mapa de la isla ya que la cueva estaba inutilizada y ahora, además, Phiby tendrá que ir al bosque a buscar la súper fuerza de Rocky. Tal y como se ha mencionado, esta súper fuerza se visualiza gracias a un halo amarillo y, por tanto, será fácil visualizar dónde está, en un champiñón con este halo (Figura 9). Al cogerlo, Phiby consigue el halo demostrando así que es él quien tiene la súper fuerza (Figura 10 izquierda) y ya puede liberar a Rocky de la cueva. Se ha creado una animación de las rocas de la cueva, simulando que Phiby, que dispone de la súper fuerza, las rompe para liberar a Rocky. Al sacarle, Rocky consigue su súper fuerza (Figura 10 derecha) y acompaña a Phiby a realizar el mini juego.

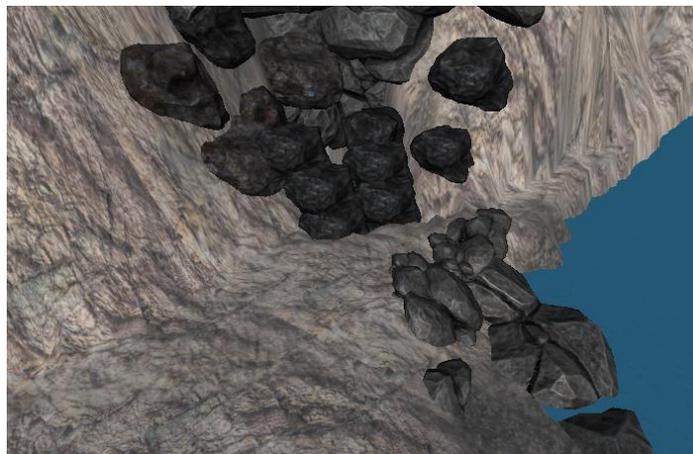


Figura 8. Rocky en la última versión atrapado en la cueva



Figura 9. Champiñón que corresponde a la súper fuerza de Rocky



Figura 10. Phiby y Rocky con la súper fuerza

- Mejora de la cámara:** se ha mejorado de forma que ahora, si el personaje pasa por delante de un objeto, este se ocluye acercando la cámara al personaje para no dejar de verle. Esto se ha realizado en el script “PhibyMixedCamera” en el que se ha usado la función de trazado de rayos, algoritmo que cuando detecta superficies, crea una nueva distancia para así ocluir estas superficies y acercarse al personaje. En las siguientes imágenes se muestra un ejemplo del funcionamiento antes y después de la mejora. En la Figura 11 únicamente se ve el árbol pues la cámara no lo ocluye, mientras que, en la siguiente figura, la cámara ignora el árbol acercando la cámara al personaje.



Figura 11. Cámara antes de la mejora

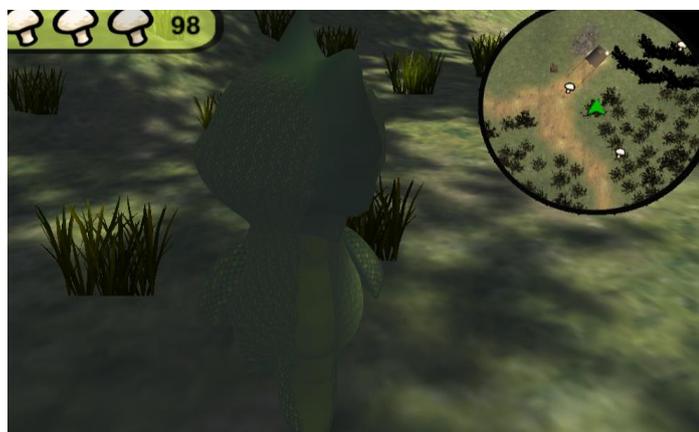


Figura 12. Cámara después de la mejora

- **Nueva escena** de ejercicio en la que el personaje debe escapar de una jaula: se ha creado la nueva escena para implementar el mini juego de la jaula. Sin embargo, no ha dado tiempo a implementarla de forma correcta con Kinect, únicamente la posición del personaje y la línea de script que cargará esta nueva escena.

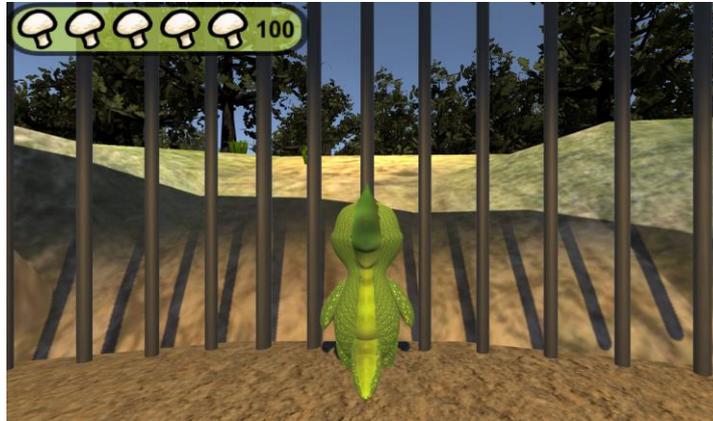


Figura 13. Nueva escena de escalar la jaula

- **Solución de diversos errores:**
 - **El personaje principal se sube encima de los animales.** Debido a la física de los animales, Phiby al acercarse para que le hablen, se sube encima de ellos. Esto es debido a la física que tienen y al funcionamiento del script al acercarse. Para solucionar esto, se opta por eliminar el rigid body de los animales que no es necesario si no se producen colisiones. Por lo tanto, se crea un script llamado “Animals” en el que en vez de que los animales hablen cuando Phiby choque con ellos, hablen cuando entren en un trigger predeterminado que les rodea. En la siguiente figura se muestra un ejemplo del funcionamiento antes y después de solucionar el error:

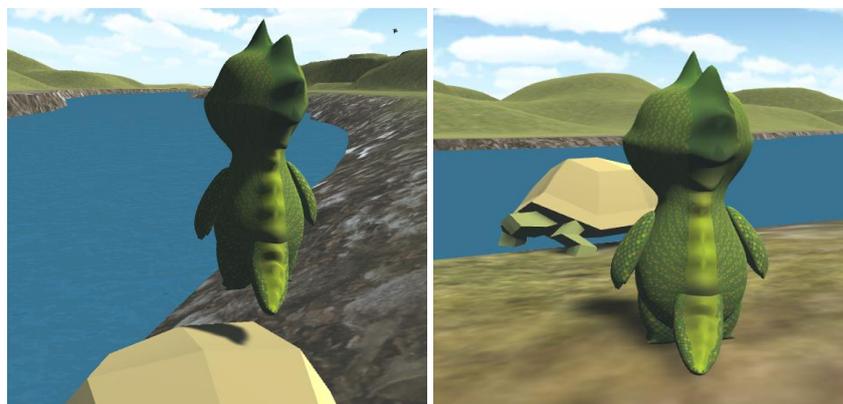


Figura 14. Phiby subido a la tortuga (izquierda) y Phiby al lado de la tortuga sin subirse (derecha)

- **Se atraviesan objetos:** existen algunos objetos como el puente o determinados animales que no disponían de un trigger y por tanto podían ser atravesados. Para solucionar esto basta con dotarles de un trigger que les cubra de forma completa.
- **Mejora del juego a nivel de rendimiento**

La solución de los diferentes errores, así como la reestructuración de las carpetas eliminando todo lo repetido o no necesario han mejorado en gran medida el rendimiento del juego. Sin embargo, se han realizado otras tareas para optimizarlo.

 - **Transform a (0,0,0):** El transform es utilizado para almacenar la posición, rotación, escala y el estado de parenting de un GameObject y, por tanto, es muy importante.

Cuando un GameObject es un Parent de otro GameObject, el GameObject Child va a moverse, rotar y escalar exactamente como su padre lo haga, mostrando su transform relativamente a los valores del Transform de su padre. Es muy útil, por tanto, establecer la ubicación del padre siempre en el origen antes de añadir el hijo. Sin embargo, en la versión proporcionada al alumno al principio del semestre, muchos de los padres e hijos existentes en la jerarquía no tienen su transform en el origen, lo que ocasiona que, en cada frame, el juego tenga que recalcular la posición, rotación y escala de cada uno, resultando ser un peso bastante grande a nivel de rendimiento. Aunque ha sido muy trabajoso, se opta por modificar el transform de todos los padres e hijos de tal forma que estén centrados en el origen. Esto va a mejorar muchísimo el rendimiento y, además, al estar todos emparentados al mismo transform, es decir, al origen, crear nuevas escenas es algo automático pues basta con copiar y pegar el objeto siendo seguro que va a estar en la posición correcta.
 - **Árboles con menos resolución, menos hojas.** Se baja la resolución de los árboles para mejorar el rendimiento, ya que tenían un nivel de detalle excesivo, aproximadamente 800 hojas cada uno. Al estar en movimiento, recalcular este para cada una de las hojas de cada uno de los árboles era un trabajo tedioso que ralentizaba el juego. Por ello, se han reducido el número de hojas aproximadamente un 80% quedando cada uno con 200 hojas sin perjudicar apenas la visualización de dichos árboles.

En las siguientes figuras se muestra la disminución del número de hojas visualizando el árbol para apreciar que, efectivamente, la apariencia de los árboles no ha sido perjudicada.



Figura 15. Diminución de las hojas de los árboles, de 820 a 204.

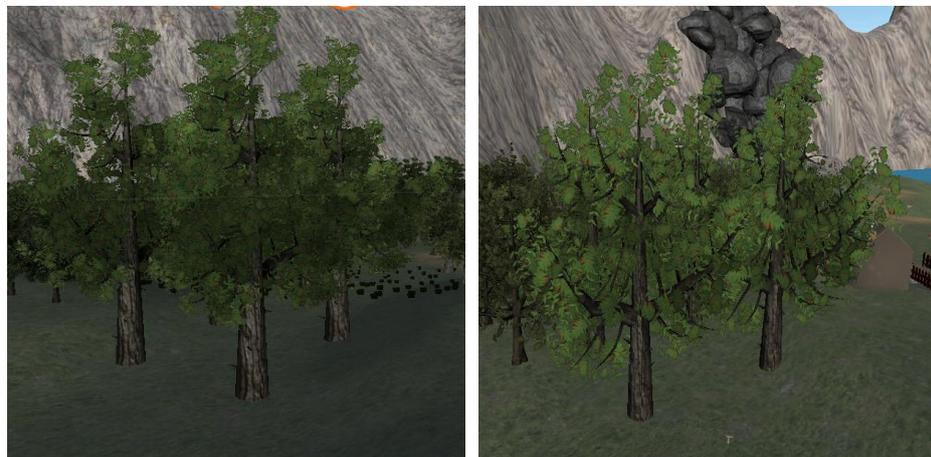


Figura 16. Árboles antes (izquierda) y después (derecha) de la disminución del número de hojas

3. EVOLUCIÓN CRONOLÓGICA DE LAS ACTIVIDADES REALIZADAS Y LOGROS OBTENIDOS DURANTE EL PERÍODO,

Se realiza en forma de tabla y de diagrama de Gantt el desarrollo de las actividades efectuadas durante todo el semestre explicadas en el capítulo anterior.

Tabla 1. Desarrollo de actividades

	Actividad	Fecha inicio	Duración en días	Fecha fin
	Entender el proyecto y su contenido	07-sep	8	15-sep
	Elaboración de lista de errores y mejoras de jugabilidad	16-sep	7	23-sep
Corrección de errores	Personaje encima de los animales	24-sep	6	30-sep
	Se atraviesan objetos	01-oct	3	04-oct
	Mejora de la cámara	05-oct	7	12-oct
Mejora de la apariencia de la isla	Modelado de señales	13-oct	6	19-oct
	Script vuelo de pájaros	20-oct	2	22-oct
	Creación de caminos y suavizado de colinas	23-oct	5	28-oct
	Estructuración óptima de las carpetas	29-oct	11	09-nov
	Estructuración óptima de la jerarquía	10-nov	5	15-nov
Mejora del juego para el jugador	Mejora de la llave: más grande y con rotación	16-nov	5	21-nov
	Nuevo funcionamiento de Rocky	22-nov	14	06-dic
Mejora del rendimiento	Transform al origen	07-dic	12	19-dic
	Bajar resolución árboles	02-ene	4	06-ene
	Nueva escena	07-ene	4	11-ene



Gráfica 1. Planificación de actividades durante el semestre de prácticas

4. TECNOLOGÍAS Y MEDIOS TÉCNICOS UTILIZADOS

Para el desarrollo de estas prácticas, los softwares utilizados han sido:

- **Unity:** es un software de desarrollo de videojuegos en tiempo real. Ha sido creada por Unity Technologies y está disponible tanto para Microsoft Windows, Mac OS y Linux permitiendo el desarrollo de videojuegos para más de 25 plataformas distintas como plataformas móviles, ordenadores y consolas.
Unity es una herramienta tan potente que no solo puede crear videojuegos, sino que también es capaz de crear experiencias de realidad virtual, entornos virtuales interactivos, aplicaciones para móviles y miniseries.
Para la edición y depuración de código destinada a crear scripts mediante el lenguaje C#, se utiliza **Visual Studio Tools**, una extensión gratuita de Visual Studio.
Todas las tareas llevadas a cabo descritas en el capítulo anterior a excepción del modelado en blender ha sido desarrollado utilizando dicha plataforma.
- **Blender:** es un software libre utilizado para el modelado, iluminación, animación y renderizado de gráficos tridimensionales. Además, incluye un motor de gráficos para videojuegos con el Blender Game Engine.
Se ha utilizado para el modelado de objetos conformados por diversas formas geométricas. Por ejemplo, si se observa la figura 3, se comprueba que las señales están conformadas por cubos con distintas formas.
- **Middleware K2UM:** fue desarrollado por alumnos anteriores en dichas prácticas. Permite la comunicación entre Unity y la cámara Kinect.

Además, se ha necesitado el siguiente equipamiento:

- **Ordenador personal**
- **Kinect:** controlador de juego libre y entretenimiento desarrollado por Microsoft para la videoconsola Xbox 360 y desde 2011 para PC a través de Windows. Permite a los usuarios controlar e interactuar con la consola sin necesidad de tener contacto físico mediante una interfaz natural de usuario que reconoce gestos, comandos de voz y objetos e imágenes.
Gracias a su cámara de 640x480 píxeles de resolución y su sensor de profundidad, es capaz de detectar a más de una persona creando un esqueleto con hasta veinte articulaciones de cada una. Esto permite que, en este juego, se detecten todos los movimientos necesarios para moverse por la isla, así como los de los mini juegos.

Por otra parte, se han utilizado diversos recursos:

- **Youtube:** se han visto distintos tutoriales, tanto para modelar en blender como para realizar diversas funciones en unity.
- **Documentación** de la asignatura Síntesis y Animación de Imágenes.
- **Scripting API:** documentación que Unity provee en la que se muestra las clases disponibles para scripts, métodos, propiedades, ejemplos y cualquier otra información relevante para su uso.
- **Proyectos de fin de grado** de alumnos anteriores consultados para un mejor entendimiento del juego.

5. COMPETENCIAS Y HABILIDADES ADQUIRIDAS CON LAS PRÁCTICAS

Durante estos meses de prácticas he adquirido distintas competencias y habilidades:

- **Capacidad de búsqueda y selección de información, de razonamiento crítico y de elaboración y defensa de argumentos dentro del área:** se han consultado multitud de documentos, tanto para entender la funcionalidad del juego en el comienzo, como para realizar tareas después. Por tanto, se han consultado PFGs, el Scripting API de Unity, documentación de la asignatura Síntesis y Animación de Imágenes...
- **Capacidad de abstracción, de análisis y de síntesis y de resolución de problemas:** a lo largo de las prácticas han tenido lugar multitud de problemas que en un principio no se han sabido resolver y tras trabajar horas en ellos y buscar información se ha conseguido solucionarlos.
- **Capacidad de trabajo en equipo y en entornos multidisciplinares:** se ha trabajado como se mencionó en la introducción, con dos compañeros más y, por tanto, ha sido de vital importancia el trabajo en equipo; escuchar las ideas que surgían, proponer nuevas y tener en cuenta la retroalimentación recibida. Esto ha permitido avanzar de manera óptima con el juego, la mayoría del tiempo dividiéndonos el trabajo pero además en distintas ocasiones ayudarnos los unos a los otros resolviendo aquellos problemas que surgían.
- **Capacidad de organización, planificación y toma de decisiones:** de la misma forma que antes, ha sido muy importante organizar el equipo y planificar los objetivos para llevar adelante el proyecto.
- **Capacidad para crear, codificar, gestionar, difundir y distribuir contenidos multimedia, atendiendo a criterios de usabilidad y accesibilidad de los servicios audiovisuales, de difusión e interactivos:** se han modelado nuevos objetos en blender y se ha programado en el lenguaje C# en unity logrando así una mejora del videojuego.
- **Habilidad para la utilización de las Tecnologías de la Información y las Comunicaciones.**
- **Autoaprendizaje y aplicación de los conocimientos adquiridos en diversas asignaturas:** durante el desarrollo de las prácticas se ha trabajado con distintos recursos para el autoaprendizaje como han podido ser vídeos de youtube o el manual de Unity. Además, he profundizado mi conocimiento en los softwares de Unity y Blender los cuáles fueron impartidos en la asignatura Síntesis y Animación de Imágenes.
- **Expresión oral** de mis conocimientos y avances para la puesta en común del trabajo realizado además de **expresión escrita** elaborando distintos documentos.

6. CONCLUSIONES

Estoy muy agradecida por la oportunidad que se me ha brindado de poder trabajar en este proyecto. No solo he adquirido nuevos conocimientos a nivel académico, sino que también he podido vivir de primera mano lo que será un entorno real de trabajo, mejorando mis habilidades y valorando lo realmente importante, la eficiencia cuando no trabajas solo, la actitud frente al trabajo, la toma de decisiones... por todo ello, solo tengo palabras buenas para valorar esta experiencia.

En cuanto al equipo de trabajo, fue una auténtica sorpresa enterarme de que, de nuevo, iba a formar equipo con Dario y Henar. Un semestre atrás, en la asignatura Síntesis y Animación de Imágenes formamos equipo y creamos un juego de escape. Esto ha hecho que estuviéramos muy compenetrados a la hora de trabajar y que el entorno haya sido familiar, no solo entre nosotros sino con nuestra tutora también. Martina ha sido un apoyo constante. Sin ella, habríamos tenido muchos momentos de altibajos. Ha habido momentos de desesperación cuando algo no salía, momentos de enfrentarme a dificultades como el lenguaje de programación o la creación de animaciones y ella no ha parado de dar ideas, opiniones y sobre todo ánimos para que siguiéramos adelante. Sin duda, desafiar los problemas te hace mejorar y crecer como persona, tanto a nivel personal como profesional.

Desde el principio, mis compañeros y yo teníamos claro uno de los objetivos y quizá el más importante: darle una buena base al juego. Queríamos aprovechar que los tres teníamos conocimientos de Unity para arreglar errores, poner comentarios en scripts y dejar todo documentado. Esto favorece a todos aquellos que se incorporen dentro del grupo de investigación sin experiencia con este software, permitiéndoles trabajar de forma más cómoda, fijándose en funciones ya creadas o consultando la documentación.

Para terminar, me gustaría resaltar que he disfrutado muchísimo trabajando en este proyecto. Phiby's Adventures v2 es un proyecto muy ambicioso, en el que, desde el principio he estado muy comprometida e ilusionada. Y por supuesto, aunque mis prácticas acaben aquí, siempre estaré disponible para ayudar a Phiby a conseguir su objetivo, encontrar a toda su familia.

A todos aquellos que se incorporen al proyecto, disfrutad y recordad que para mejorar este proyecto no hacen falta tantos conocimientos, sino, sobre todo, ganas, compromiso e ilusión.