

HAT-EYELAND

Proyecto RV Interfaces Inteligentes

27 de enero de 2017

Diego Luis Afonso

Daniel Daher Pérez

Cristina Tosco González

En el siguiente enlace se encuentra el repositorio de github que contiene nuestro proyecto:

<https://github.com/dieg8/hateyeland>





En esta breve memoria se mencionan algunas cuestiones importantes del diseño, desarrollo e implementación del juego desarrollado para este proyecto de la asignatura Interfaces Inteligentes.

Descripción

Nuestro proyecto de realidad virtual se trata de un juego de búsqueda de una serie de elementos (tótems) en una isla encantada y deshabitada. Para ello, contamos con un personaje principal, perdido en esta isla desierta. Este personaje necesita recolectar una serie de tótems que se encuentran en unos determinados puntos de la isla. Una vez recolectados todos estos elementos, aparecerá el último ítem, la recompensa final, que le permitirá abandonar la isla en busca de civilización.

Objetivos

En nuestro juego, como hemos mencionado, el personaje tiene que encontrar en distintas zonas de la isla una serie de elementos que aparecen listados a la izquierda de la pantalla. Al adquirir uno de estos elementos, este desaparecerá del territorio y quedará constancia. Una vez obtenidos todos, el personaje tendrá acceso a una recompensa final que le permitirá abandonar la isla, con lo que finaliza el juego. Para llegar a este punto anteriormente oculto, dispondremos de unos rayos que señalarán el lugar en el que se encuentra.

Tótem fuego	Tótem aire
	
Este tótem se encuentra en lo alto de la montaña atravesando un puente sobre la lava.	Tenemos acceso a este elemento una vez pasamos el laberinto.
Tótem agua	Tótem tierra
	
Se encuentra en un pequeño islote dentro del lago.	Se encuentra dentro de la cárcel en ruinas que se esconde bajo la pirámide
Tótem Naturaleza	? > Recompensa Final

	
<p>Se encuentra en un poblado de setas y gnomos</p>	<p>En un principio no se puede acceder a él. Aparece tras recolectar todos los tótems. Suena un trueno y los rayos llevan hasta donde se encuentra.</p>

Contenidos

A continuación se recogerán los distintos hitos logrados relacionándolos con los contenidos que se han impartido en clase y las sesiones prácticas.

Assets

Para la realización del proyecto se han importado múltiples assets empleados para diseñar todo el escenario de la isla. Estos assets nos proveían de distintos modelos, prefabs, materiales, shaders y texturas variados que permitieron desarrollar el mundo de juego.

GameObjects

Todos los elementos que aparecen en la escena del juego son GameObjects. Los más destacables son:

- **Fuentes de luz.** La escena dispone de varios puntos de luz, **Directional Light** de color anaranjado y **Points light** para algunos tótems.
- Existe una plataforma origen (gameObject) que teletransporta al personaje directamente a otra zona del mapa completamente inaccesible de otra forma. El teletransportador tiene un Collider y un Script esperando a que nuestro RigidbodyFPSController entre en contacto con la plataforma para así convertir la posición del objeto chocante, es decir, la del personaje, en la posición del otro gameObject o plataforma destino.

- El **personaje** también cuenta con un script que lo hace interactuar con los diferentes tótems, haciéndolos desaparecer, reproduciendo el sonido y modificando el canvas.
- Elementos de **interfaz de usuario (UI)**:
 - Canvas: coloca una pantalla en la que poder mostrar información al jugador. Este canvas tiene como hijos unos gameObjects que informan de los tótems recogidos mediante UI text.
 - Botón de parada y continuación: Es un objeto que forma parte del juego, pero permite al jugador detener o reanudar el avance del personaje. Siempre se encuentra a los pies del personaje, haciendo posible detenerse o continuar simplemente mirando al suelo. Esto se consigue mediante un EventTrigger que detecta cuándo se mira al collider del botón y un script que recoge estos eventos y actúa en consecuencia.
 - Mensaje final: Una vez se han obtenido todos los tótems y el tesoro final, aparecerá un mensaje en el Canvas que informa al jugador de que ha acabado el juego.
- **Sistemas de partículas.** Se han colocado varios de estos elementos en distintos lugares del mapa para añadir efectos visuales que aumenten el grado de inmersión y el realismo:
 - Las nubes de la parte alta de la montaña.
 - La niebla que existe sobre la nube del laberinto.
 - Los tótems emanan diferentes partículas con efectos.
 - En la zona de lava encontramos humo.
 - Una llama de fuego sobre uno de los tótems, que también expulsa humo rojo.
- **Objetos vacíos:** En el istmo en el que aparece el tesoro se encuentra un objeto vacío. Se utiliza para definir la posición en la que aparece la instancia del tesoro final. Por otra parte, existen unas paredes formadas por emptys en lo alto de la montaña para que el usuario no se pueda caer a la lava.

Components

El personaje y la mayoría de los elementos que intervienen en nuestra escena poseen la combinación de los siguientes componentes Rigidbody - Collider. Sin embargo, existen algunos, como las setas, el agua o la lava, que no lo tienen para evitar que bloqueen el avance del personaje.

Los componentes también permitieron modificar la visualización de la cámara, dando el aspecto de isla encantada al atardecer, así como dotar al juego de efectos de sonido, lo que lleva a una mayor inmersión en el mismo.

Scripts

Importamos a nuestro proyecto el paquete de unity de Google CardBoard VR. Con los conocimientos previos de prácticas anteriores y documentación en el campus virtual, aplicamos el plugin Google VR a nuestro proyecto, añadimos los prefab necesarios e implementamos y añadimos los scripts correspondientes para un correcto funcionamiento.

Para el desarrollo de HatEyeland hemos hecho uso de 3 scripts principales:

- **Tp.cs:** Script encargado del teletransporte del personaje que colisiona con la plataforma teletransporte a lo alto del volcán a la plataforma objetivo en lo alto de la nube. En este Script vinculamos mediante el editor de Unity el RigidbodyFPSController, y la plataforma teletransporte objetivo al script como variables public para poder acceder dentro del script a las posiciones de dichos gameObject. Tp.cs es un componente de la plataforma teletransporte del volcán. Cuando el rigidbody colisiona con la plataforma, esta se encarga de acceder al vector posición del personaje y modificarla con la posición de la plataforma objetivo.
- **Pause.cs:** Script componente de “Boton”, botón que tenemos como hijo del RigidbodyFPSController y que siempre va a seguir a nuestro personaje en su movimiento a lo largo de nuestro terreno. El botón tiene la función de que cada vez que lo miremos con la retícula detendrá o reanudará el movimiento de nuestro personaje. El script accede a otro script, el predefinido de movimiento del Rigidbody, para modificar cada vez que es “mirado” cambie el valor de movimiento de nuestro personaje, 0 si queremos que esté quieto o 1 para iniciar o reanudar la marcha (el valor predefinido es 0 ya que no nos interesa que el juego empiece sin la directa actuación del usuario, cuando esté listo). Este Script utiliza los métodos onGazeEnter() para modificar esta variable ya nombrada.
- **Totems.cs:** Este es el script principal de nuestro proyecto. Totem.cs tiene distintas funcionalidades, pero siempre como centro el hecho de que se encarga de monitorizar los totems que se van consiguiendo y las acciones que se realizan a consecuencia.
 - Como primera tarea nos encargaremos de ambos UI.Text, correspondientes al panel lateral en el que nos muestra la lista de los objetivos actuales, los que hemos conseguido y los que no. Además de modificar el panel lateral con los progresos que vayamos consiguiendo, cuando llegue el momento en el que hemos logrado obtener todos los tótems necesarios nos saldrá un mensaje de felicitaciones por haber completado el juego.
 - El otro punto central de este script es la instanciación y destrucción de objetos. Cada vez que interactuamos con un objeto, si la tag de ese objeto es “totem”, veremos de qué tipo de tótem es y realizaremos la acción acorde, después destruiremos este GameObject. En el caso final en el que se consigan los 5 tótems “principales” se instanciará el prefab Cofre, sería el tótem final objetivo, en un pequeño islote detrás de el volcán, la localización de la instancia se consigue debido a un Empty object que tenemos localizado en dicho islote.

Desarrollo del juego

El desarrollo del juego, desde el diseño de nuestra isla, a la implementación de cada uno de los Scripts utilizados, la creación de la apk, etc, se ha explicado en el **vídeo** que se adjunta en el siguiente [enlace](#).

Para ver una partida de nuestro juego: [Playthrough](#).

Metodología de trabajo

Este proyecto de realidad virtual se ha comenzado a desarrollar a principio del cuatrimestre. Sin embargo, la asignatura en la que nos encontramos está dividida en dos proyectos y en clase se abandonó el proyecto tras el primer mes. Por tanto, este trabajo se ha desarrollado desde el último mes y medio hasta la fecha de entrega. Una vez conseguimos instalar y hacer funcionar Unity Collaborate, trabajamos perfectamente de forma conjunta.

En un principio se comenzó con la creación de la isla en la que se desarrolla la experiencia de realidad virtual. Una vez diseñado la mayor parte del mapa se decidió dónde se colocarían los diferentes objetivos del juego (tótems). Además, se desarrollaron los entornos donde se encuentran de forma que se adapten a ellos. Por ejemplo, el tótem de fuego se colocó en la cima de un volcán activo.

También se añadieron un conjunto de Components a la cámara para producir los efectos ya comentados (luz al atardecer, skybox, sonidos...). Tras acabar con todas las tareas relacionadas con la estética del mapa, se comenzó a desarrollar el comportamiento de los diferentes objetos de la escena mediante el uso de Scripts.

Por último incluimos algunos elementos para mejorar la experiencia del usuario como son el desarrollo de una interfaz de usuario (canvas) en la que se muestran los detalles de los logros del juego, y el botón de parada/continuación que nos permite una mejor interacción, permitiendo al personaje intercambiar entre sus dos modos, andar o parar, con sólo mirarlo.

Por otra parte, no nos podemos olvidar de la adición de los paquetes necesarios para su correcta visualización en las cardBoard, así como del cambio a la plataforma Android para el funcionamiento en los dispositivos móviles.