Carla Lek Boada

Para poner la vegetación he migrado el asset de hierba que usamos en el proyecto de Icaro. Con el pincel de vegetación, empecé a poner y borrar vegetación por encima del terreno (hecho por mi compañera Meri) pintado de verde, minimizando su presencia en zonas de tenue verde.

Lo siguiente fue buscar un árbol para completar la decoración. Realmente me llevó más tiempo del esperado encontrar un modelo preparado para Unreal ya que muchos no incluían sus propias texturas hacía imposible crear el material. Después de poner el modelo, metí las texturas en un material para poder ser aplicado. Después de estar listo, simplemente lo fui colocando por el escenario.

Para el mar importé el que hice en clases. Se ajustó el tamaño y el color para que encajaran con el escenario.

Para el sangrado del personaje hice uso de un material dinámico combinado con un Fresnel.

Para hacerlo edité el material principal del personaje. Intenté hacer algunas funciones usando PostProceso y otros recursos pero no salió bien. Tampoco conseguí (o no entendí muy bien) crear máscaras de texturas que funcionaran para lo que quería. Después de varios intentos fallidos sin llegar a ningún sitio, me acordé de la función Fresnel.

Quería hacer que lo que cambiara fuera una silueta del personaje, así que probando vi que me servía. En el nodo de material emisivo del material del personaje hice una interpolación entre la función de Fresnel, la textura base y un color rojo que simularía el recibir daño. Al terminar, creé una instancia de dicho material.

Para implementarlo en el personaje, fui a la función de “recibir daño” del personaje (hecha por mi compañero Àlex) y le agregué unas referencias a la instancia del material. Añadí una timeline para regular el “encendido” y “apagado” del material emisivo.

Para las plataformas simplemente agregamos unas que ya creamos de antemano en clases y les cambiamos algunas variables.

Meritxell Vázquez Venturo

En primer lugar, he creado un terreno en el que las montañas estaban en los alrededores para delimitar el mapa excepto por un lado que se ha hundido para hacer un mar (que se encargó Carla). Al terreno le puse erosión solar en las zonas montañosas y a la zona del mar erosión de agua.
A continuación creé un material landscape con nuevas texturas encontradas en Textures Com de hierba, barro y roca pero hubo un problema de que se veían mal aplicados en el terreno así que cogí el material de la clase y le cambié las texturas.
Después creé los menús a partir de la game mode ya creada en la clase y corrigiendo algunos errores como, por ejemplo, borrar el Widget de pausa cuando va al menú principal y que en el menú principal el juego esté en pausa.
A continuación he hecho migrate de los coleccionables, la puerta y las plataformas para incorporarlas y cambiar los casts para el blueprint del pawn en primera/tercera persona. También Àlex corrigió que las puertas se quedaran abiertas una vez se consumían las estrellas necesarias para abrirlas. Después creé el HUD del gameplay con un Widget en el que las estrellas recogidas se muestran en números y la vida en una progression bar. Se actualizan mediante su respectivo binding, en el que el número de las estrellas pasa por texto el valor guardado en el inventario mediante la Game Instance y el de la vida coge la vida actual del inventario de la Game Instante y la divide por la vida máxima que es 100. Para que siempre saliera independientemente del State del juego, creé un blueprint tipo HUD en el que crea el widget explicado previamente y lo añade, y también se ha cambiado el HUD por defecto del Project Settings por el nuevo HUD.

Después he aplicado las condiciones de victoria y fracaso en forma de Widgets que te avisan cuando has ganado y cuando has perdido. La condición de derrota está en el evento del character de perder vida por estar cerca de un zombie en el que comprueba si su vida está a 0. En caso que lo esté, pasa al estado de derrota y se muestra un texto diciendo que has perdido y no puedes hacer nada más. En cuanto a la victoria, el objetivo del jugador es llegar a la estatua rara de cristal que está en medio de la escena y, para ello, debe buscar las estrellas para pasar por las puertas. Cuando llega a la estatua mediante su collider, se muestra un widget de que has ganado y finaliza el juego.

Àlex Vidal Domingo

En el proyecto la primera tarea a llevar a cabo fue la creación del personaje, para ello me base en el controlador creado en clase y añadiendo algunas funcionalidades como ocultar la malla para que no moleste en primera persona pero que sea visible mientras esté en tercera.

A continuación desde Mixamo.com conseguí una malla con texturas, un esqueleto y un set de animaciones de idle, correr y saltar. Para el blendspace de una dimensión use las animaciones de idle y correr dependiendo de la velocidad del player.
Para la inteligencia artificial use Mixamo.com otra vez para conseguir una malla con texturas, un esqueleto y una animación de correr.
La inteligencia artificial cuenta con una blackboard que guarda referencias al jugador, a uno mismo y a si ha visto al jugador, un controlador y una máquina de estados que cambia entre dos comportamientos distintos dependiendo si ha visto al player o no.
Los dos comportamientos son el de patrullar donde el zombie busca un punto navegable dentro de un radio delimitado, cuando llega al punto espera un momento y busca otro. El segundo estado es el de perseguir al player mientras este esté en el campo de visión del zombie, cuando lo pierde de vista vuelve a patrullar.

En el juego hay unos obstáculos representados como puertas, basadas en los ejercicios de clase se abren mediante estrellas recogidas por el jugador, con la funcionalidad añadida de que una vez abiertas se pueden pasar las veces que el jugador precise sin coste de estrellas adicional.

El jugador posee un inventario en el cual se almacenan las variables de la vida que tienen actualmente y el número de estrellas actual. Si el jugador colisiona con una estrella se añade a su inventario y si se acerca a una puerta y tiene suficientes estrellas se restan de este.
Para finalizar realice el guardado y la carga de partida junto a Meritxell creando un blueprint de tipo SaveGame donde se almacenan la posición del jugador, la cantidad de vida y estrellas que tiene. Para cargar la partida seteamos los valores anteriormente guardados en el game slot *SaveSlotMJ.*