Fausto Sánchez Hoya

Índice

[1 Introducción 2](#_Toc85201952)

[2 Diseño y estética del juego 3](#_Toc85201953)

# Introducción

En este trabajo se va a desarrollar el videojuego Dangerous Dungeons. Un videojuego 3D desarrollado con Unity 2020.3.18f1 en el que encarnaremos a nuestro personaje “Mike el Todopoderoso” que intenta liberar a su tierra de toda una serie de enemigos que la han conquistado.

Este videojuego será de tipo “Diablo” en el que se deberán de liberar una serie de escenarios de enemigos para poder ir avanzando.

¿Conseguirá Mike liberar a su tierra? ¿Conseguirá Fausto programar adecuadamente el mundo de Mike?

Lo iremos viendo…

# Diseño y estética del juego

Para empezar he elegido que la estética del juego será parecida a la del videojuego “Minecraft Dungeons”.

Por lo que he decidido utilizar assets creados con Voxel para dar esa apariencia de cubos o “pixel art”. La mayoría de estos recursos proceden del autor Maxparta[[1]](#footnote-1) excepto los relacionados con las armas que son de Unity-Fantom [[2]](#footnote-2).

 Imagen que contiene luz, tabla, pequeño, grupo

Descripción generada automáticamente

Imagen que contiene barco, iluminado, viendo, oscuro

Descripción generada automáticamente

Por otro lado, las animaciones de los personajes serán extraídas de Mixamo[[3]](#footnote-3).

# Desarrollo

## Importación de assets

Lo primero que he hecho ha sido importar todos los assets que me había descargado de varias páginas a mi proyecto de Unity. Para ello he creado una carpeta con el nombre del proyecto y dentro de ella otra carpeta llamada assets.

Texto

Descripción generada automáticamente con confianza baja

## Pruebas de animaciones en Mike

Para empezar con el proyecto quería probar las animaciones del personaje principal por lo que el primer paso es elegir a nuestro personaje de “Mike el Todopoderoso”. Al final me he decantado por este modelo (TPose\_Character05):

Imagen que contiene lego

Descripción generada automáticamente

He creado una carpeta dentro de personajes llamada Mike que contendrá todos sus recursos:

Texto

Descripción generada automáticamente con confianza baja

A continuación, he buscado en Mixamo animaciones de movimiento como correr y andar. Para ello he importado primero su modelo a la página web:

Interfaz de usuario gráfica

Descripción generada automáticamente

Seguidamente he descargado las animaciones de “Idle”, “Walking” y “Running”. Como en Mixamo te puede generar el prefab con el modelo y el esqueleto para directamente importarlo a escena, en la primera animación (Idle) he descargado estos recursos y por tanto serán utilizados como el modelo a importar a la escena de forma predefinida. Todos estos recursos estarán en la carpeta de “Animación” dentro de la carpeta de Mike:

Interfaz de usuario gráfica, Aplicación, Teams

Descripción generada automáticamente

Como Unity tiene problemas para reconocer las texturas del modelo 3D, he exportado las texturas y las he guardado en una carpeta llamada Materials dentro de la carpeta Mike:

Texto

Descripción generada automáticamente

Nuestro personaje ya está listo para ser importado por lo que sólo nos hace falta arrastrarlo a escena, colocarlo y aumentar su tamaño para que sea distinguible:

Imagen que contiene agua, vuelo, colgando, esquiando

Descripción generada automáticamente

A continuación, he creado un animator controller llamado “MikeAnimator” el cual he alojado dentro de la carpeta “Animations”, he hecho que el primer estado al que pasa Mike nada más empezar es al de reposo y se lo he añadido al GameObjet Idle de la escena.

Interfaz de usuario gráfica

Descripción generada automáticamente

Como la animación tiene un fin he activado la opción Loop Time en todas las animaciones para que se queden reproduciéndose, he activado también la opción Loop Pose para que haya menos problemas en las transiciones y las he añadido al animator controller.

Interfaz de usuario gráfica

Descripción generada automáticamente

Para controlar el paso de una animación a otra he creado un parámetro entero que representará los tres estados de velocidad que puede llevar:

* Si está en 0 está en reposo.
* Si está en 1 está andando.
* Si está en 2 está corriendo.

Teniendo este parámetro en cuenta he creado las transiciones de un estado a otro a las cuales les he desactivado la opción de “Has exit time” para asegurar que en cuanto el personaje comience a correr su animación también lo haga.

Imagen que contiene interior, tabla, computadora, juego

Descripción generada automáticamente

Ahora que sus animaciones básicas ya funcionan vamos a proceder con su activación a través de los controles del juego. Para el desplazamiento he decidido que el jugador podrá usar las teclas W A S D para moverse por el mapa y el L-Shift para aumentar su velocidad por lo que deberemos crear un script que recoja la activación de esas teclas y cambie el estado del integer “Velocidad” de la animación. Este script y todos los relacionados con Mike estarán alojados en una carpeta llamada Scripts dentro de la carpeta Mike:

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

De momento sólo puede estar en tres estados a la vez por lo que su controlador de movimiento es muy básico todavía:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

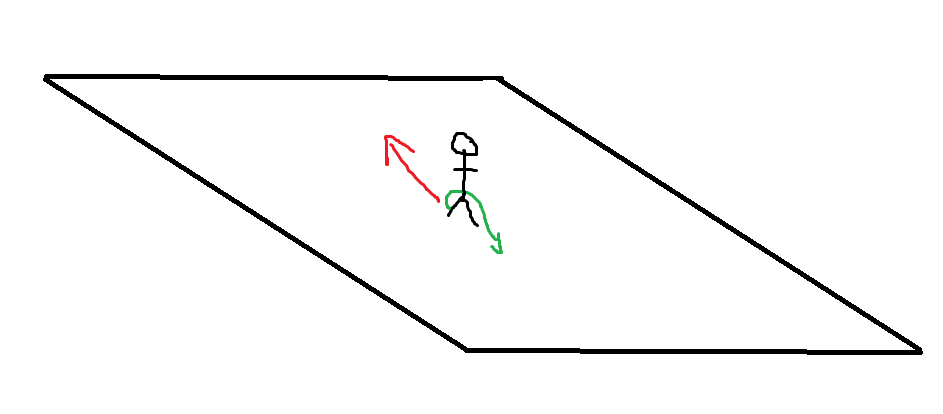
## El movimiento de Mike

Es hora de hacer que Mike pueda hacer movimientos más complejos como girar (todavía no me voy a centrar en los ataques). Para ello vamos a definir primero los movimientos que va a poder hacer nuestro personaje.

Mike podrá caminar en todas las direcciones, pero siempre andando hacia delante, es decir, si Mike está mirando hacia la parte de arriba de la pantalla y quiere andar hacia la parte de abajo no podrá andar marcha atrás, sino que dará un giro de 180 grados y comenzará a andar hacia delante.

Rojo: dirección actual.

Verde: camino que seguiría para ir en dirección contraria.



Para ello me voy a descargar una serie de animaciones de giros y de cambios de velocidad de movimiento.

Seguidamente he completado el animator de Mike con los estados nuevos. Para tener en cuenta los giros he creado otro parámetro integer el cual he llamado “Dirección” el cual tendrá los siguientes valores:

* Si está en 0 deberá hacer la animación de caminar/correr hacia delante
* Si está en 1 deberá hacer la animación de caminar/correr hacia la izquierda si está moviéndose o simplemente se girará si está quieto
* Si está en 2 deberá hacer la animación de caminar/correr hacia la derecha moviéndose o simplemente se girará si está quieto
* Si está en 3 hará la animación de darse la vuelta y su estado cambiará automáticamente a 0.

Diagrama

Descripción generada automáticamente

En cuanto a su controlador de movimiento he hecho que pueda cambiar a todos sus estados pulsando las teclas W A S D y L-Shift haciendo que por defecto haga que esté quieto y de ahí a cualquier estado modificando su dirección y velocidad.

A continuación, me enfrento a un problema: las animaciones mueven al personaje ligeramente y si se vuelven a ejecutar reinician su posición haciendo que experimente un movimiento brusco. Para solucionarlo lo primero que hago es en la parte de Rig de nuestro modelo “Idle” seleccionamos crear un avatar a partir de este modelo y seleccionamos como nodo raíz las caderas:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Haciendo esto se nos crea nuestro avatar que podremos ver dentro de el modelo “Idle”.

Interfaz de usuario gráfica

Descripción generada automáticamente

A continuación he configurado el avatar de avatar de cada animación que me daba problemas como el avatar que hemos generado:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Después he configurado la opción del “Root Motion” para que en aquellas animaciones que daban problemas se configure correctamente haciendo que su último estado se tome como inicio de la siguiente. Ejemplo de la animación de correr mientras se gira:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

En este caso como quiero que la animación controle la rotación del personaje mientras gira pero no su avance porque eso quiero controlarlo yo mediante código, activo la opción “Back Into Pose” para los ejes x y z.

He realizado esta acción en todas las animaciones que lo necesitaban y finalmente, para que esto se aplique, en el animator de Mike he adjuntado su avatar y he activado la opción de “Root Motion”.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

1. Recursos del autor en <https://maxparata.itch.io/> [↑](#footnote-ref-1)
2. Recursos del autor en <https://unity-of-fantom.itch.io/voxel-character-pack-v1> [↑](#footnote-ref-2)
3. Mixamo: <https://www.mixamo.com/#/> [↑](#footnote-ref-3)