

Indicaciones Especiales

Descripción

En este mini-juego encarnarás al guerrero legendario popularmente conocido como “The Hammer” debido a una gran batalla que ganó con únicamente un golpe. Nadie conoce su verdadero nombre ni nada más acerca de él. Siempre está cuando las cosas se ponen crudas tiene un sentido del olfato muy fino para localizar luchas cercanas.

Ábrete el paso por el continente de Orzammar realizando misiones para poder conseguir items que te serán útiles para luchar contra un poderoso gigante que amenaza con destruir todas las ciudades que encuentre a su paso.

Controles

WASD - Movimiento

N - Pausar

Click Izquierdo - Seleccionar/Atacar

E - Entregar

F - Desenfundar/Enfundar

Spacebar - Saltar

I - Abrir inventario

1,2,3,4,5 - Bailar

Tab - Cambiar cámara

Mecánica

Durante el juego se pedirá al usuario que recoja una serie de objetos, entre armas y coleccionables (pollos). Las armas que el usuario recoja se añadirán a su inventario, algunas de ellas con funciones especiales.

El usuario puede encontrarse también con enemigos que le lanzarán barriles, para eliminarlos deberá atacarlos con un arma.

La batalla final contra Shagazz es un duelo de baile en el que el usuario deberá repetir el baile que está haciendo dicho jefe. Para conseguir los bailes, el jugador tendrá que completar todas las misiones que le pedirá el maestro en el primer poblado.

Link a vídeo: <https://www.youtube.com/watch?v=QGAhQNVcAes>

Àlex Montaña

- **Un solo nivel**
- **Mundo abierto** o mezcla de mundo abierto y cerrado (**uso de Landscape obligatorio con material de al menos dos capas**)
- **Uso de la herramienta de vegetación**
- **Al menos un blendspace**
- **Vídeo**

Creación de un único nivel, formado por un landscape esculpido de manera que hemos intentado simular una parte de una isla donde encontraremos desde montañas nevadas hasta el mar e incluso lava.

Es un mundo abierto, el jugador tiene total libertad sobre cómo quiere ejecutar su partida, aunque las misiones tengan un orden, no hay ninguna restricción más. Uso de un material con 4 capas (piedra, tierra, césped e hielo). Cada una de estas tiene una ciudad especial, es decir, el pueblo del hielo, etc. Aunque la piedra también sirva para comunicarlas entre ellas mediante una serie de caminos.

Uso de la herramienta de vegetación para decorar el mapa conforme a la temática correspondiente. (ej. árboles nevados en la zona nevada)

Los blendspace los podemos encontrar en el personaje y en el maestro (npc te indica las misiones). El blendspace del personaje lo hace con las animaciones de idle y de correr. En cambio, el maestro cambia entre las 3 animaciones que tiene de idle para que no acabe siendo repetitivo.

Realización del video para enseñar el gameplay del juego.

Sergio Mora

- **Un menú principal básico y posibilidad de guardar y recuperar la partida (total o parcialmente).**
- **Al menos una instancia dinámica de material con un parámetro que se modifica en tiempo real.**
- **Al menos un blueprint de animación con su máquina de estados.**
- **Al menos una IA con Behaviour Tree en al menos un tipo de NPC.**
- **Una mecánica completa del nivel (objetivo, condición de victoria y condición de fracaso).**
- **Al menos un UMG con al menos dos variables a representar en tiempo real.**

Menú principal con opción play, cargar o salir; y menú de pausa con opción de continuar, guardar o salir. La función de guardado es parcial, pues sólo guarda algunos de los principales datos del juego. En el caso de las armas, en algunas ocasiones se buggea y estas se multiplican y/o desaparecen. Para las misiones, no se guardan las completadas, pero sí las variables que chequean su resultado, por lo que al cargar el player solo se tendrá que dirigir al maestro para conseguir todas las recompensas de las misiones realizadas al guardar.

La instancia dinámica de material se encuentra en el “Flotador de la luz” (espada de globo). Cuando el jugador se acerca a otra arma empuñando éste, el world position offset de los vértices varía según la distancia a dicha arma, creando un efecto de vibración a más cerca se esté de ella.

En el proyecto encontramos diversos blueprints de animación, el del maestro, el del tira barriles, el de Shagazz y el del jugador. Todos los dichos blueprints usan un sistema sencillo de bools y floats, excepto el del jugador que utiliza un nodo de blend per bone, para separar la animación de las piernas de las del torso y permitir así el ataque en movimiento.

La IA de este “juego” se encuentra básicamente en el tira barriles. Su comportamiento consiste en patrullar mientras no ve al player, seguirle hasta cierta distancia si lo vé, dispararle si lo tiene a rango y huir cuando éste se le acerca.

Como se ha explicado al inicio el objetivo principal es derrotar a Shagazz en un duelo de baile, completando las misiones se consiguen esos bailes. La condición de victoria es ganar a Shagazz y la de derrota, morir.

En cuanto a las variables de UMG, encontramos la vida y la stamina en la parte superior izquierda del HUD. La vida se resta cuando nos alcanzan los tira barriles y la stamina, cada vez que atacamos se baja. Mientras no nos ataquen y no atacemos, tanto la vida como la stamina se regeneran.

A parte de estos requisitos he hecho algunas cosas más:

- Implementación de un sistema de inventario en el que se establecen las armas que recogemos y que nos muestra el nombre y la descripción del arma que tenemos equipada en UMG, con el uso de Data Table y struct.
- UMG en la que se muestran los bailes y que se activan al completar misiones.
- Duelo de baile, gestión de las animaciones de ambos personajes y comparación para saber si hace la que se debe.
- Sistema de misiones a través de una Data Table que almacena todas las misiones y su descripción, para que el maestro las replique.
- Diferentes armas con diferentes funciones: El caballo aumenta la velocidad, el martillo es el único que puede arreglar la espalda del herido, la espada globo reacciona ante el resto de las armas.
- Implementación del socket para las armas, cambio de armas y enfundar o desenfundar a través del cambio entre 2 meshes.
- Capa de colisión en la que solo se encuentra el collider del player para gestionar como el resto de objetos reaccionan ante este.
- Integración de una música ambiental usando un Ambient Sound.

Jordi José Pascual

- **Al menos un pawn en primera persona y al menos uno en tercera persona (puede ser el mismo).**

El pawn es el personaje y puede alternar entre la 3a persona para ver el modelo del personaje y sus animaciones o la 1a persona, para tener la vista del arma.