



IES VISTA ALEGRE
2ºDAM

LEGENDS OF THE AETHER

David Jiménez Pastor

José Manuel López Díaz

ÍNDICE

Presentación y motivación del proyecto.....	2
Género y Mecánicas.....	3
Historia.....	3
Mecánicas.....	4
Sistema de trabajos.....	4
Inventario.....	5
Habilidades.....	5
Misiones.....	6
Guardado y carga.....	7
Personajes y Entorno.....	8
Personajes.....	8
Entorno.....	9
Desarrollo del Proyecto.....	13
Herramientas Utilizadas.....	13
Unity.....	13
Visual Studio Code y C#.....	13
Blender.....	13
Mixamo.....	13
Adobe After Effects.....	13
Photoshop y Gimp.....	13
Proceso de Desarrollo.....	14
Planificación.....	14
Prototipado.....	14
Implementación.....	14
Pruebas.....	14
Aspectos Técnicos.....	15
Estructura del Código.....	15
Scripts Clave.....	15
Desafíos Técnicos.....	16
Resultados y Conclusiones.....	16
Logros.....	16
Feedback y Pruebas.....	16
Conclusiones.....	16

Presentación y motivación del proyecto

Nuestra principal motivación fue lo que suponía desarrollar un videojuego 3d en menos de 2 meses, es decir, todo un reto. Haciendo que este sea estable y funcional para que tenga una jugabilidad aceptable.

Además de ser un reto, pensamos que sería una buena idea como introducción al mundo del desarrollo, ya que la industria de los videojuegos es una de las más punteras.

Género y Mecánicas

Historia

Nuestro videojuego es de tipo rpg (**juego de rol**) de mundo abierto.



Teniendo que llevar a cabo ciertas misiones en él, las cuales van desenmarañando la historia principal a medida que las completas.

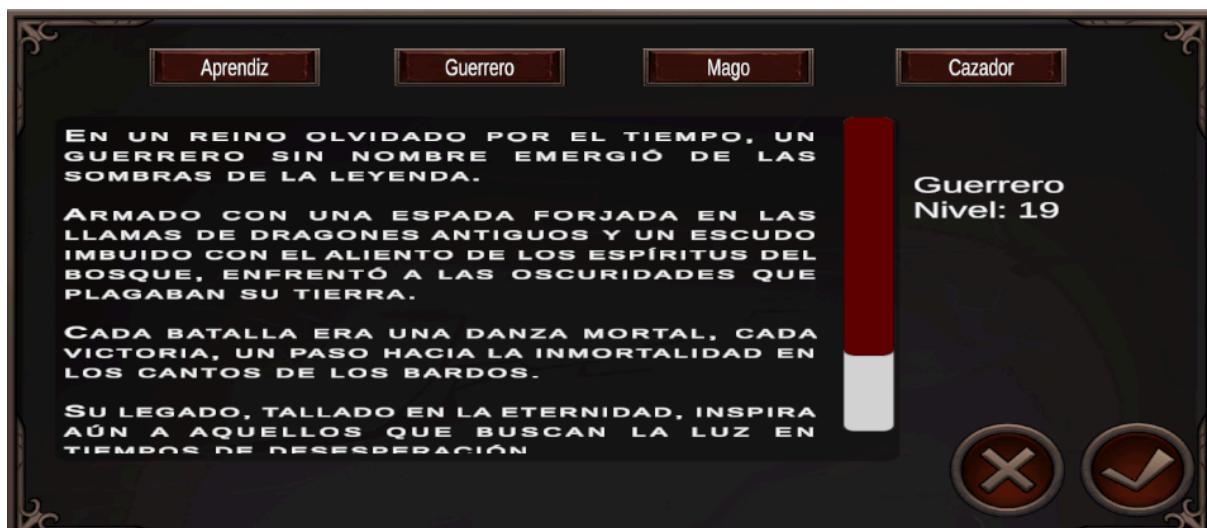
Esta historia trata de defender las amenazas del Gran Brujo Oscuro, nuestro enemigo principal el cual ha desatado una maldición sobre Terralia, nuestro mundo, dando vida a seres mágicos que tratan de entrometerse en nuestro camino.

La magia del Brujo Oscuro obtiene semejante poder del Aether, un mundo paralelo cuya energía oscura se expande sin control emanando una oscuridad que corrompe y destruye todo a su paso.

Mecánicas

Sistema de trabajos

Para llevar a cabo nuestra aventura hemos implementado un sistema de trabajos en el cual el jugador puede elegir diferentes clases como guerrero, mago, cazador o la inicial, aprendiz.



Además del nivel del propio personaje el cual aumenta a medida que obtienes experiencia ya sea matando enemigos o completando misiones, también tenemos nivel del trabajo, el cual aumenta nuestras estadísticas en función al trabajo para así potenciar este. Cada trabajo, además, tiene sus propias habilidades y para poder usarlas requieren un cierto tipo de equipamiento, como en el caso del cazador que requiere de arco y flecha.

Inventario

En cuanto al equipamiento del personaje, además de las armas ya mencionadas hemos incorporado 7 slots más en los cuales puedes equipar elementos como guantes, cascos, pecheras...



Estos mismos también aumentan tus estadísticas, como por ejemplo la armadura, en el caso de las armas, incrementa bastante el daño del jugador.

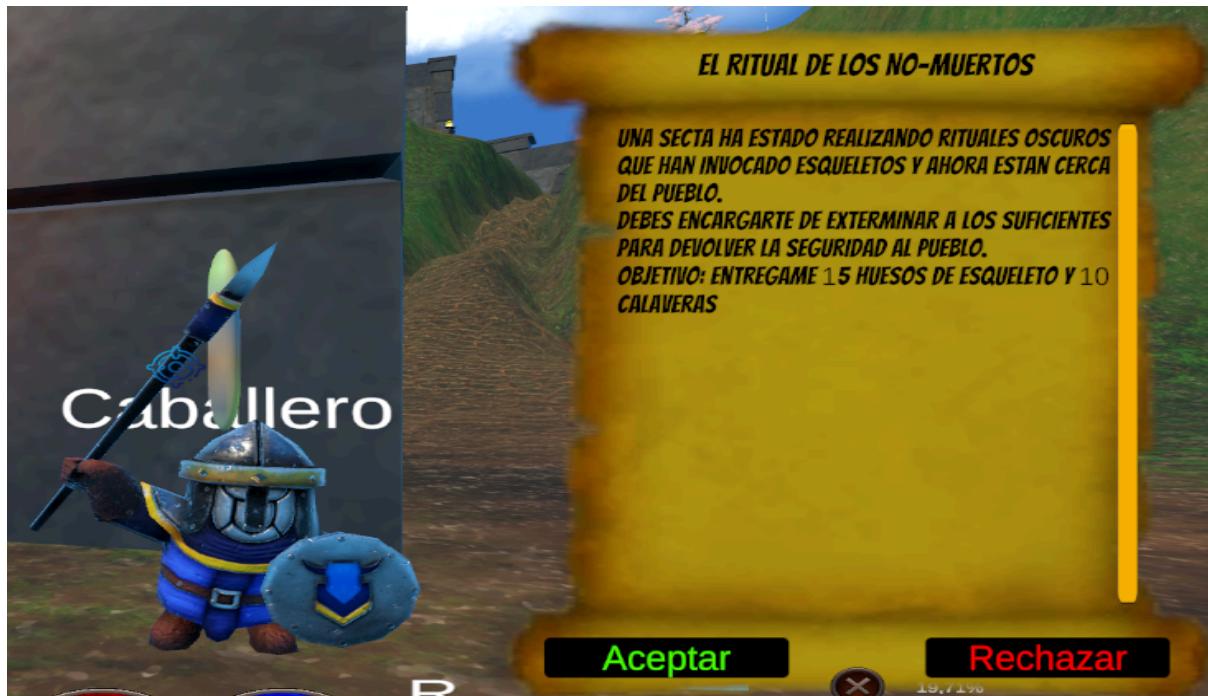
Habilidades

Como característica estrella, hemos agregado varias ramas de desarrollo para nuestro personaje, es decir, un árbol de habilidades en el que especializar a nuestro personaje en distintas direcciones:

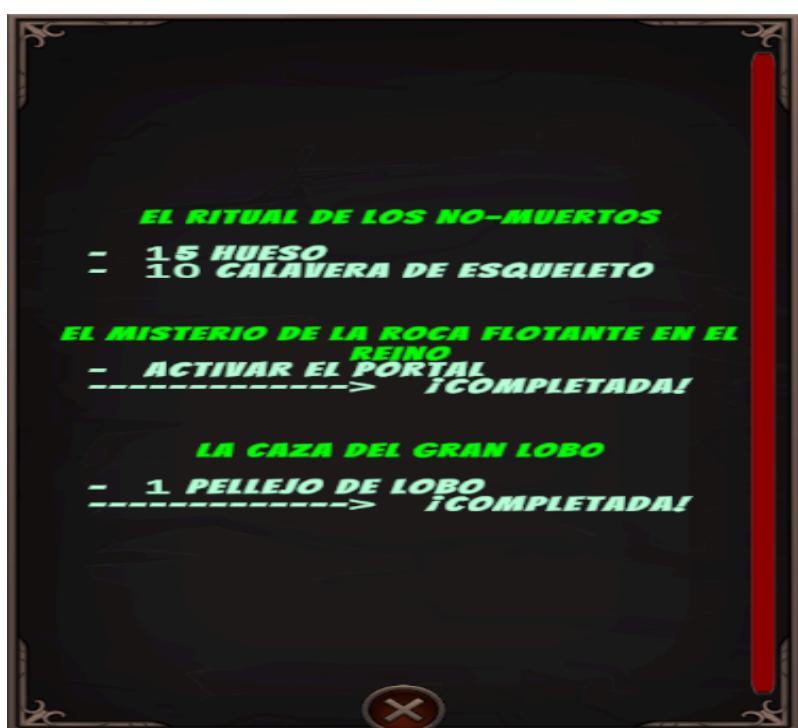


Misiones

En cuanto a misiones hemos incorporado menús de interacción con los npcs donde podemos aceptar, rechazar, abandonar o entregar nuestras misiones.



Para llevar la cuenta de las misiones que tenemos aceptadas y si las podemos entregar o no hemos creado esta interfaz:



Guardado y carga

Finalmente, a todos nos gusta guardar nuestro progreso, pues lo tenemos, lo hemos guardado por medio de archivos JSON en directorios seguros del proyecto, en nuestro menú principal podremos encontrar las opciones de cargar y guardar.



Y en nuestro menú de carga del juego podremos tanto iniciar una nueva partida como cargar nuestra partida anterior:



Personajes y Entorno

Personajes

➤ Protagonista:

- Gilgamesh, un joven aprendiz del mágico mundo de Terralia.



➤ Aliados:

- Los sabios del reino, mentores y guías de nuestro protagonista.



- Caballeros honorables del reino, los protectores del mismo.



➤ Enemigos:

- El Gran Brujo Oscuro, es el antagonista principal.



- Criaturas del Aether y guardianes corruptos, son los enemigos secundarios invocados por el mismísimo Brujo Oscuro

Entorno

- El portal mágico: zona en la que comienza nuestro protagonista, donde los sabios del reino le dan la bienvenida y le instruyen en su primera misión.



- Las ruinas embrujadas: Unas ruinas sometidas por el poder del Aether en la cuales los cofres han cobrado vida y son hostiles.



- El altar de los no muertos: Un altar custodiado por no muertos liderados por un nigromante.



- Eldoria: Capital de Terralia, es la ciudad poblada por los antiguos, los más sabios del reino.



- El poblado orco: Un poblado lleno de orcos que siguen la causa de su líder, Grulk el destructor.



Desarrollo del Proyecto

Herramientas Utilizadas

Unity

Fue el motor de desarrollo elegido para crear nuestro videojuego ya que proporciona mayor flexibilidad de cara al uso de ciertas herramientas incorporadas en él y facilita el acceso a assets que ayudan en el desarrollo.

Gracias a Unity implementamos una creación de terrenos avanzada junto con su manejo de físicas además de el audio integrado.

Visual Studio Code y C#

C# Fue el lenguaje de programación utilizado para desarrollar los scripts y las mecánicas del juego. Elegido por su compatibilidad con Unity, permitió desarrollar la lógica del juego, el movimiento de los personajes, la interacción con el entorno y más.

Blender

Utilizado para modelar diseños 3D

Mixamo

Utilizado para obtener y personalizar las animaciones de los personajes

Adobe After Effects

Empleado para crear efectos más especializados y de alta calidad.

Photoshop y Gimp

Utilizados para la edición de texturas y generación de sprites completos

Proceso de Desarrollo

Planificación

- I. Se realizó un brainstorming inicial para generar y organizar ideas.
- II. Durante esta fase se definieron la historia, mecánicas y diseño de niveles.
- III. Definición del concepto del juego, creación de un plan de desarrollo, establecimiento de objetivos y metas.

Prototipado

- I. Creación de un prototipo jugable para probar las mecánicas básicas del juego
- II. Pruebas iniciales, ajuste de mecánicas

Implementación

- I. Desarrollo completo del juego, incluyendo programación, diseño de niveles, arte y sonido.
- II. Programación de la lógica del juego, creación de gráficos y animaciones, integración de sonido y música, desarrollo de niveles y misiones.

Pruebas

- I. Pruebas exhaustivas para detectar y corregir errores, así como ajustar el balance del juego.
- II. Pruebas unitarias, pruebas de integración, pruebas de jugabilidad, optimización del rendimiento, depuración de errores.

Aspectos Técnicos

Estructura del Código

Organización Modular: El código se dividió en módulos para facilitar el mantenimiento y la escalabilidad del proyecto. Cada módulo correspondía a una funcionalidad específica del juego, como el movimiento de personajes, el sistema de combate o la gestión de inventarios.

Scripts Clave

- **PlayerMovement:** Este script controlaba el movimiento y las animaciones del personaje principal, incluyendo el desplazamiento, saltos y ataques.
- **Habilidad, JobSystem y ProjecitlBehaviour:** Gestiona los ataques, habilidades especiales y la detección de colisiones, permitiendo un combate dinámico y estratégico.
- **InventorySystem y SistemaEquipamiento:** Permite al jugador recoger, utilizar y gestionar objetos en su inventario, incluyendo armas, pociones y recursos.
- **NPC, Mission y NPCAttributes:** Constituyen la estructura de las misiones.
- **EsferaController:** Gestiona todo el árbol de mejoras del personaje.
- **EnemySpawner:** Hace que la generación de enemigos sea más fluida, ya que los crea en una pool y los va reutilizando en vez de crear y destruir.
- **EsferaManager:** Se encarga de gestionar el guardado y cargado del juego.
- **UIController:** Gestiona todos los scripts de UI incorporados, tales como EsferasUI, InventoryUI...
- **GameManager:** Verifica la carga correcta del juego e inicia rutinas de spawn.

Desafíos Técnicos

- I. **IA de Enemigos:** Desarrollo de una inteligencia artificial convincente para los enemigos y jefes del juego, creando comportamientos complejos y patrones de ataque variados.

- II. **Balance de Juego:** Ajuste de habilidades y dificultad para mantener el juego desafiante pero justo, asegurando una experiencia equilibrada para los jugadores.

Resultados y Conclusiones

Logros

- I. **Desarrollo de un Juego Completo:** Creación de un juego funcional con una historia envolvente y mecánicas bien desarrolladas.
- II. **Aprendizaje y Aplicación:** Aplicación exitosa de conocimientos teóricos en un proyecto práctico, desarrollando habilidades técnicas en programación, diseño gráfico y gestión de proyectos.

Feedback y Pruebas

- I. **Mejoras Implementadas:** Basado en el feedback, se realizaron ajustes en la dificultad, se corrigieron errores y se optimizó el rendimiento del juego.
- II. **Comentarios Recibidos:** Se recibieron todo tipo de comentarios positivos y constructivos durante las pruebas, lo que permitió mejorar la jugabilidad y ajustar el balance del juego.

Conclusiones

Para finalizar, vamos a destacar un par de puntos claves que hemos aprendido/mejorado gracias a este proyecto y una breve conclusión para acabar:

Desarrollo de Habilidades: Mejora en áreas clave como la programación, diseño gráfico, sonido y gestión de proyectos.

Planificación y Gestión: Aprendizaje sobre la importancia de la planificación, la gestión del tiempo, la adaptación a imprevistos y sobre todo a establecernos pequeñas metas que vamos cumpliendo poco a poco hasta llegar al objetivo, gracias a estas pequeñas metas hemos evitado estancarnos y continuar avanzando sin pausa.

Este proyecto ha sido prácticamente un trabajo a tiempo completo, hemos tenido que llevar a cabo muchas tareas con una gran coordinación, trabajo en equipo y sobre todo comunicación.

La comunicación ha sido muy eficaz y nos ha evitado repetir tareas, además, cada apartado que íbamos desarrollando lo creamos de tal forma que luego hemos sido capaces de reutilizarlo y así no tener que crear tantas cosas de cero.

Hemos empleado un lenguaje que desconocíamos por completo y hemos sido capaces de crear 46 scripts cada uno de ellos con su propia funcionalidad con un código muy limpio y bien estructurado.

Incluso hemos creado modelos 3D por medio de herramientas como Blender y efectos con After Effects.

Ha sido un gran proyecto que nos ha dado una base muy sólida para el mundo del desarrollo, ya que gracias a él hemos aprendido a buscarnos las maneras para llevar a cabo este proyecto tan ambicioso.

FIN