

Audio Dinámico para Videojuegos

Jhony Steven Ubaque Guerrero

Universidad Nacional Abierta y a Distancia UNAD
Escuela de Ciencias Básicas, Tecnología e Ingeniería
Tecnología en Producción de Audio

2024

Bogotá D.C.

Dedicatoria

Dedico este logro a mi madre, por su apoyo incondicional, por brindarme la oportunidad de estudiar al asumir este esfuerzo, y por ser mi mayor motivación durante todo el proceso.

Y a mi novia, por su paciencia y constante ánimo, que fueron clave para culminar este proyecto.

Tabla de Contenido

Introducción	5
Objetivos	6
Objetivo General	6
Objetivos Específicos	6
Tarea 1: Análisis de un videojuego.....	7
Tarea 2: Diseño de eventos sonoros en audio middlewares.....	11
Tarea 3: Integración de audio middleware con motor de videojuegos	15
Tarea 4: Triggers, snapshots y modificación continua de parámetros	20
Tarea 5: Mezcla y finalización de un proyecto de sonorización de videojuego	24
Anexos	26
Conclusiones	27
Bibliografía	28

Lista de Figuras

Figura 1 <i>Menú del juego</i>	8
Figura 2 <i>FMOD Time Line Parameter</i>	11
Figura 3 <i>Pistas de audio</i>	12
Figura 4 <i>Scatterer instrument</i>	13
Figura 5 <i>Parámetros</i>	14
Figura 6 <i>Setup Wizard de FMOD</i>	15
Figura 7 <i>Asset Store</i>	16
Figura 8 <i>Objects spatializer</i>	17
Figura 9 <i>Room Tone en Unity</i>	17
Figura 10 <i>Ambientes</i>	18
Figura 11 <i>FootSteps Script</i>	19
Figura 12 <i>Snapshots</i>	20
Figura 13 <i>Automatizaciones</i>	21
Figura 14 <i>Triggers y Snaphots</i>	22
Figura 15 <i>Zonas invisibles con triggers</i>	23
Figura 16 <i>Trigger Once</i>	23
Figura 17 <i>Grupos de mezcla</i>	24
Figura 18 <i>Profile de rendimiento</i>	25

Introducción

El diseño sonoro dinámico es una herramienta fundamental en el desarrollo de videojuegos modernos, ya que permite crear experiencias inmersivas que responden en tiempo real a las acciones del jugador.

En el marco del Diplomado en Audio Dinámico para Videojuegos, este trabajo tiene como propósito presentar el proceso de creación de un diseño sonoro dinámico aplicado a una escena de un videojuego, utilizando *Unity* como motor de desarrollo y *FMOD* como middleware para la integración de audio dinámico, mostrando así la aplicación práctica de los conocimientos adquiridos durante el diplomado.

Durante el desarrollo del proyecto, se exploraron conceptos clave en el diseño sonoro, como los audios interactivos, adaptativos y variables, mostrando cómo se integran para enriquecer la experiencia del usuario en los videojuegos.

Objetivos

Objetivo General

Diseñar e implementar audio dinámico para una escena de videojuego, utilizando *Unity* como motor de desarrollo y *FMOD* como middleware, con el fin de mejorar la experiencia inmersiva del jugador mediante sonido interactivo y adaptativo.

Objetivos Específicos

Analizar las necesidades sonoras de la escena seleccionada, identificando los elementos interactivos que requieren audio dinámico.

Diseñar y desarrollar ambientes sonoros interactivos que respondan a las acciones y el contexto del jugador en tiempo real.

Implementar el sistema de audio dinámico en *Unity* utilizando *FMOD*, configurando eventos sonoros y parámetros adaptativos.

Realizar pruebas y ajustes en la integración del audio para garantizar un óptimo funcionamiento y que se adapte a la narrativa y jugabilidad de la escena.

Tarea 1: Análisis de un Videojuego

Se realizó el análisis al juego Hellblade: Senua's Sacrifice (Ninja Theory, 2017) por su innovador diseño sonoro, el cual destaca por la manera en que utiliza el audio para reforzar la narrativa, crear inmersión y transmitir las emociones y el estado mental de su protagonista. Este enfoque lo convierte en un excelente referente para estudiar cómo los sonidos adaptativos, interactivos y variables pueden integrarse de manera muy creativa en un videojuego.

Figura 1

Menú del juego



Nota. Menú de inicio del juego, tomado de "Steam Community" Steam, 2017,

<https://steamcommunity.com/sharedfiles/filedetails/?id=2148271090>

El análisis se realizó según las siguientes categorías:

Ambientes

Los ambientes en Hellblade son fundamentales para crear inmersión y transmitir la atmósfera oscura y opresiva del juego. Los sonidos ambientales incluyen viento, agua, ecos en cavernas y la naturaleza circundante.

Los sonidos cambian según el entorno en el que se encuentra Senua (bosques, cavernas, zonas abiertas).

Variables

La intensidad y textura del sonido ambiental fluctúan según las condiciones climáticas o el estado mental del personaje.

Efectos de sonido

Incluyen los pasos de Senua, el sonido de su espada al chocar con enemigos, y efectos de criaturas sobrenaturales. Estos sonidos refuerzan la acción y los momentos de tensión.

Interactivos

Responden directamente a las acciones del jugador, como los movimientos y los ataques.

Variables

Cambian en función de la distancia, dirección y contexto (por ejemplo, si Senua está corriendo o caminando).

Diálogos

Uno de los aspectos más destacados del juego son las voces que Senua escucha, que representan su estado mental y sus trastornos psicológicos. Estas voces están diseñadas en binaural para dar una sensación de inmersión total.

Adaptativos

Los diálogos se activan según las interacciones del jugador con el entorno o los eventos de la historia.

Interactivos

Las voces reaccionan a las decisiones del jugador, como resolver acertijos o entrar en combate.

Música

La música en Hellblade tiene un enfoque emocional, con pistas que varían entre sonidos melancólicos y ritmos intensos durante el combate.

Adaptativa

La música se ajusta al ritmo de la acción, aumentando en intensidad durante las batallas o disminuyendo en momentos de exploración.

Variable

Los arreglos musicales cambian según el desarrollo de la narrativa, reflejando las emociones de Senua.

Sonidos de interfaz

Interactivo

Los sonidos responden a las acciones del jugador, como navegar por las opciones del menú o seleccionar configuraciones.

Adaptativo

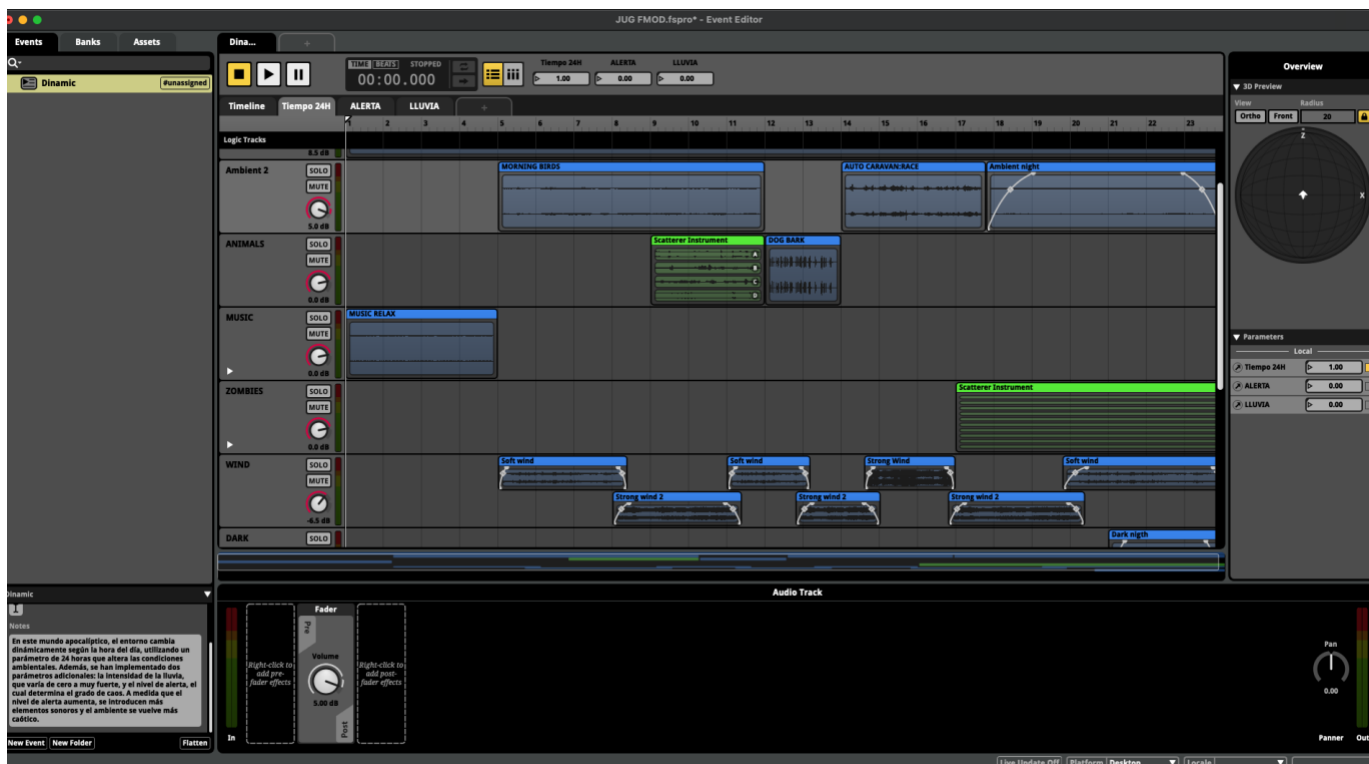
El sonido del menú puede variar dependiendo del momento en el que se accede, ajustándose al ambiente sonoro del lugar donde se pausa el juego.

Tarea 2: Diseño de Eventos Sonoros en Audio Middlewares

Para esta tarea, se diseñó un ambiente sonoro que representara un mundo apocalíptico con cambios durante las 24 horas del día utilizando *FMOD* como middleware. El proceso incluyó la creación y configuración de eventos sonoros dinámicos mediante el uso de diferentes clases de tracks, instruments y parámetros usando action sheet de *FMOD*.

Figura 2

FMOD Timeline parameter



Fuente. Elaboración propia

Desarrollo del ambiente sonoro

Clases de tracks

Se emplearon pistas de audio divididas en varias capas:

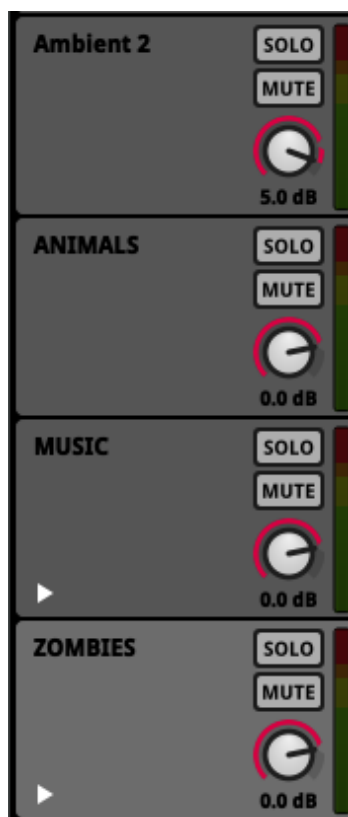
Ambiente base. Ruidos de viento variable, escombros cayendo y maquinaria distante.

Eventos especiales. Explosiones ocasionales, chirridos metálicos y ecos en el entorno.

Texturas sonoras. Capas de ruido granular y tonos bajos para reforzar la sensación de desolación.

Figura 3

Pistas de audio



Fuente. Elaboración propia

Clases de Instruments

Looping Instruments

Utilizados para sonidos continuos como el viento y demás ambientes.

Scatterer Instruments

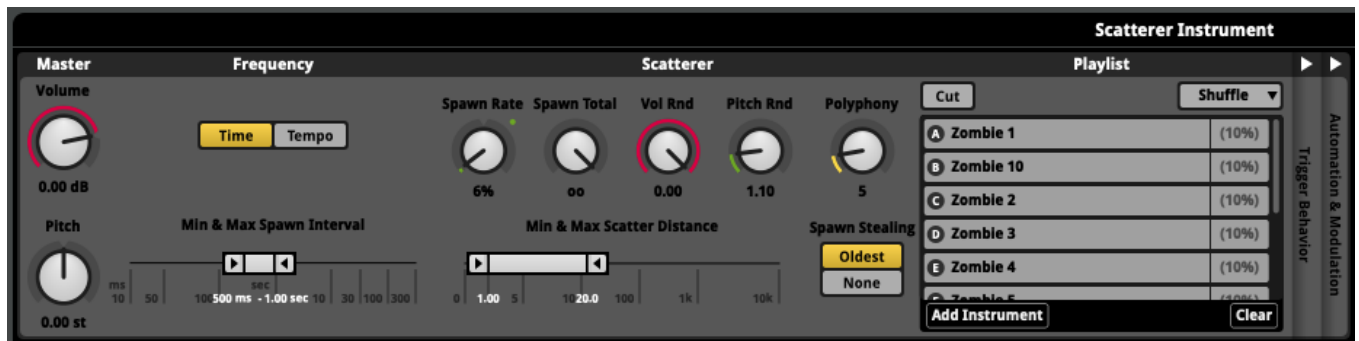
Configurados para sonidos aleatorios como explosiones o el sonido de hordas de zombies.

Multisound Instruments

Asignados a efectos variados, como disparos de armas o sonidos de estructuras colapsando.

Figura 4

Scatterer Instrument



Fuente. Elaboración propia

Parámetros

Se utilizaron parámetros dinámicos para ajustar la intensidad y características del ambiente según el contexto. Por ejemplo, la cercanía a un evento de peligro incrementa el volumen y la frecuencia de ciertos sonidos o según la hora del día

Figura 5

Parámetros*Fuente.* Elaboración propia

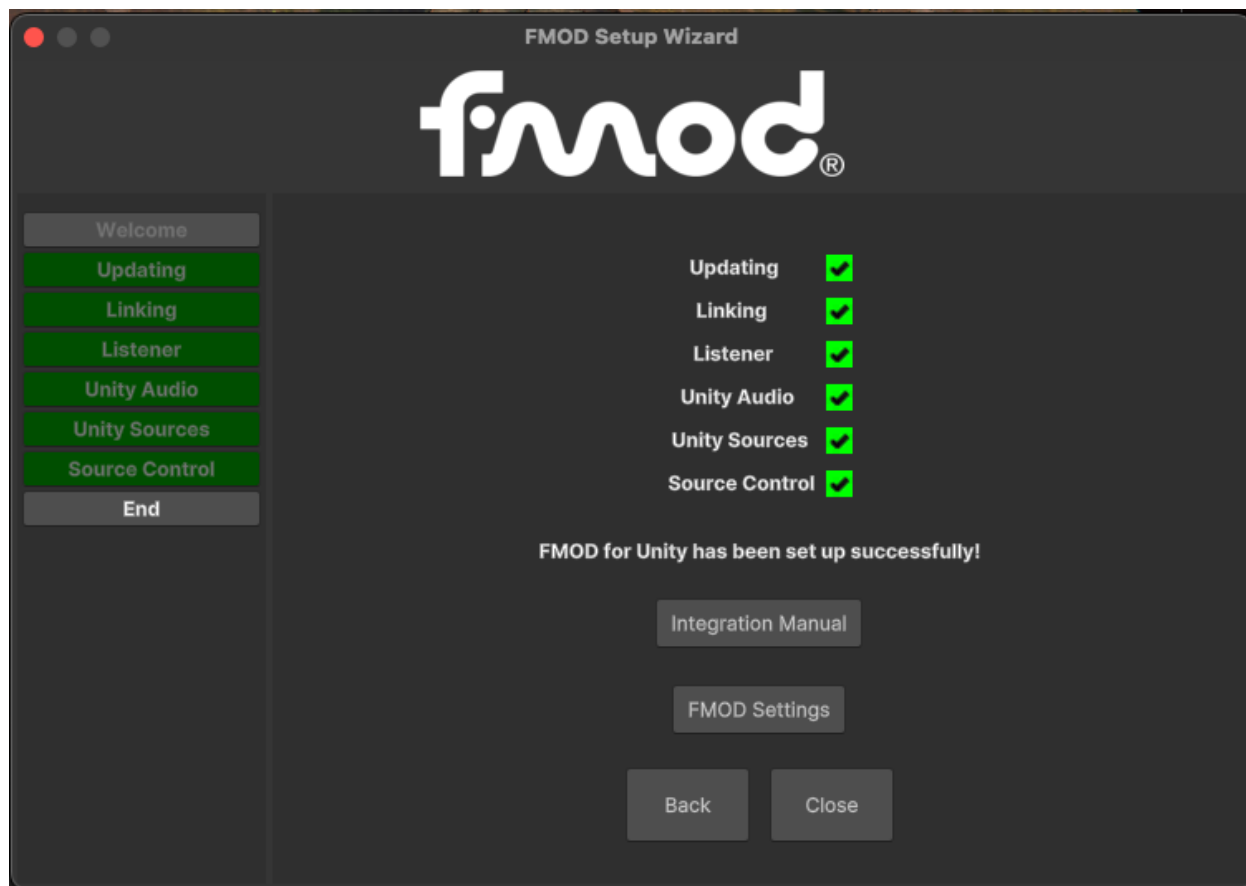
El diseño sonoro combina una atmósfera dinámica y detallada que se adapta al contexto del jugador. La integración de tracks, instruments, y parámetros en *FMOD* permitió crear un paisaje sonoro convincente de la narrativa apocalíptica.

Tarea 3: Integración de Audio Middleware con Motor de Videojuegos

En esta tarea, se llevó a cabo la integración del middleware *FMOD Studio* con el motor de desarrollo *Unity*, siguiendo las pautas y conceptos vistos en la unidad 5. El propósito fue crear una escena interactiva con diseño sonoro dinámico, incorporando sonido ambiente, efectos sonoros y música, configurados con parámetros de interactividad y variabilidad.

Figura 6

Setup Wizard de FMOD



Fuente. Elaboración propia

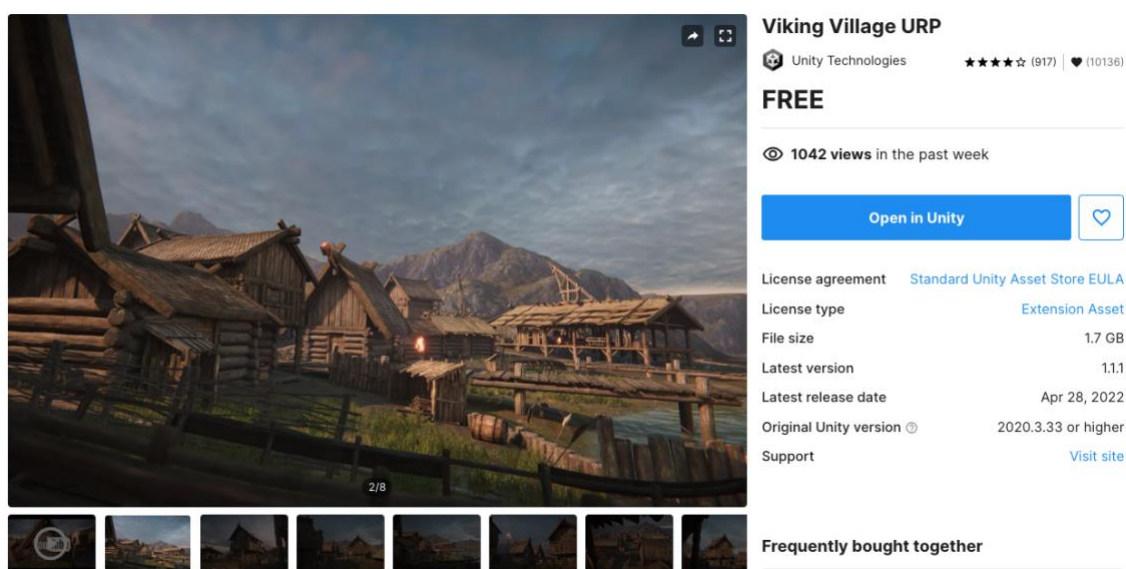
Integración con la escena

Se descargó e instaló el integrador oficial de *FMOD Studio Unity Integration*, asegurándose de que coincidiera con la versión de *FMOD* previamente instalada.

Se configuró el proyecto en *Unity* y se importó la escena Viking Village desde la tienda de assets de *Unity*, sirviendo como base para el diseño sonoro.

Figura 7

Asset Store



Fuente. Elaboración propia

Los eventos sonoros fueron diseñados en *FMOD Studio*, usando diferentes instrumentos para crear sonidos dinámicos.

Se configuraron parámetros como distancias mínimas y máximas para eventos 3D, con curvas de atenuación, y se ajustaron niveles de volumen para eventos 2D como la música de background.

Se construyen los bancos de sonido desde *FMOD* para su implementación en *Unity*.

Figura 8

Object spatializer



Fuente. Elaboración propia

Se integraron los events de *FMOD* en *Unity* mediante Studio Event Emitters en objetos 3D no solidos cuando son ambientes y directo a los mismos objetos cuando son efectos. El Studio Listener se integró a la cámara.

Los sonidos ambiente incluyeron un roomtone general para mantener la continuidad sonora, además de elementos específicos como viento, lluvia y el mar.

Los efectos sonoros se aplicaron a objetos interactivos, como antorchas, moscas, ladridos de perros y sonidos de descubrimiento en diferentes puntos de la escena.

Figura 9

Room tone en Unity



Fuente. Elaboración propia

La música se configuró como un evento 2D asignado a la cámara principal, ajustando sus niveles mediante automatizaciones para transiciones suaves entre momentos de intensidad y fondo.

Se implementaron principios de interactividad y variabilidad en los sonidos, permitiendo cambios dinámicos según la posición del jugador y el progreso en la escena como por ejemplo cambio del viento según la zona del mapa o al tener elementos cerca como las antorchas o animales.

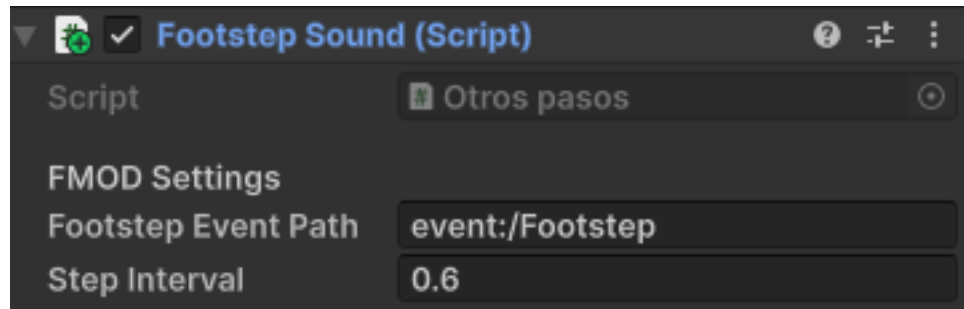
Figura 10

Ambientes



Fuente. Elaboración propia

Se creó un script para agregar sonidos de pasos al controller/cámara. Este permite que los sonidos de footsteps se reproduzcan de manera dinámica en función de los movimientos del personaje sobre madera que es la mayor superficie en la escena

Figura 11*Footstep script**Fuente.* Elaboración propia

Tarea 4 - Triggers, Snapshots y Modificación Continua de Parámetros de Audio

En esta tarea, se realizó un análisis de las mecánicas de interacción del videojuego creado en la Unidad 3 y se programaron los eventos sonoros necesarios para responder a dichas interacciones. La implementación se llevó a cabo utilizando Snapshots, Triggers y automatizaciones de audio.

Figura 12

Snapshots



Fuente. Elaboración propia

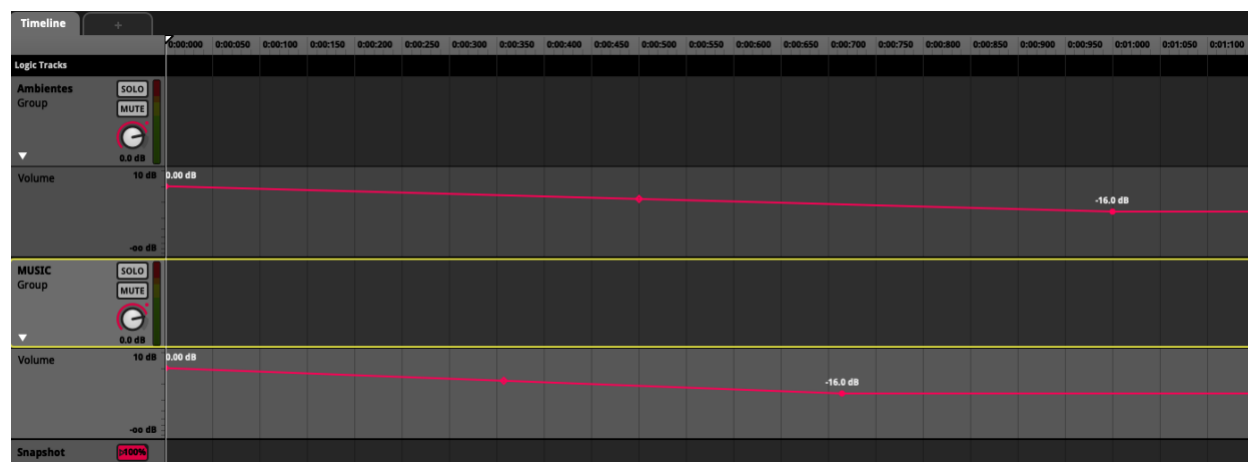
Cambio de Estados de Tensión en la Música (música adaptativa)

Se configuraron triggers en *Unity* para diferentes zonas del videojuego, activando cambios en la música según el estado del jugador por ejemplo en la aldea, la música mantiene un tono relajado. Al salir de la aldea hacia áreas inseguras, se activa un snapshot que incrementa la tensión de la música, ajustando parámetros como el volumen y el pitch.

Estos cambios se lograron mediante curvas de automatización en *FMOD*, asegurando transiciones suaves entre los estados musicales y se activan por medio de snapshots en objetos 3d invisibles.

Figura 13

Automatizaciones



Fuente. Elaboración propia

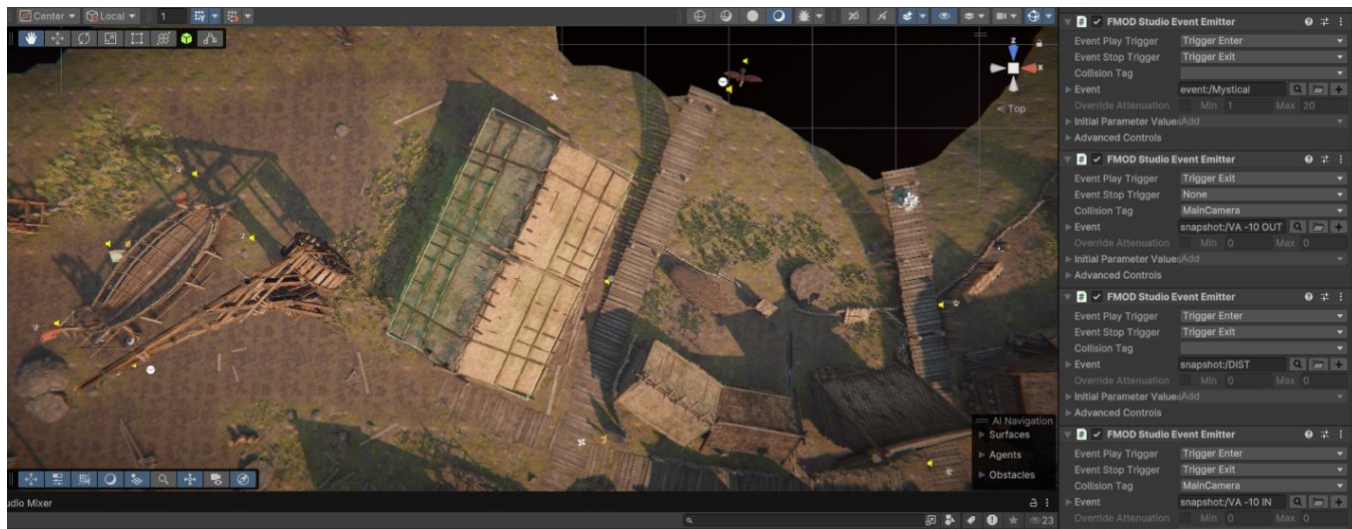
Simulación de Oclusión Mediante Automatización de Filtros

Se implementó un filtro en *FMOD* para simular la oclusión del sonido al ingresar a zonas cerradas. Este filtro, un Low-Pass Filter, se aplicó a los efectos ambientales para crear la sensación de aislamiento acústico, simulando cómo el sonido disminuye y cambia al entrar en estructuras que bloquean parcialmente el ambiente exterior.

Esto se configuró usando trigger enter y trigger exit en *Unity*, vinculados a los snapshots correspondientes.

Figura 14

Triggers y snapshots



Fuente. Elaboración propia

Activación de una Zona de Reverberación

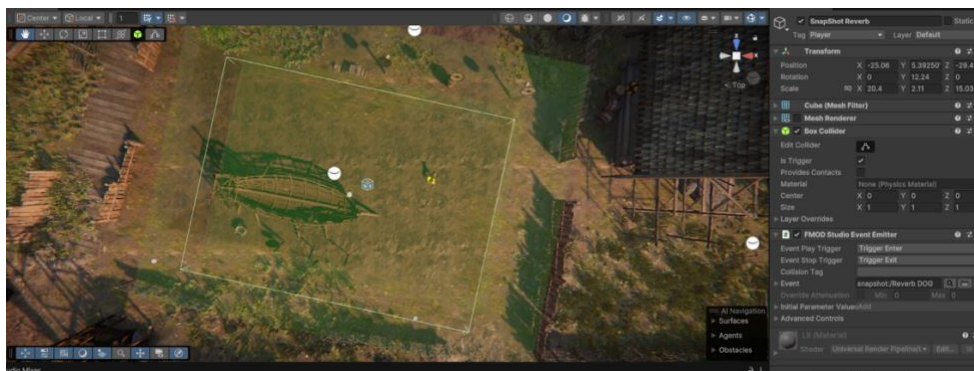
Se creó una zona con un snapshot que aplica un efecto de reverberación al entrar en áreas específicas.

Todos los eventos diegéticos, como pasos, diálogos y efectos ambientales, son afectados por la reverberación. La música no diegética permanece excluida de este efecto. Un ejemplo claro es en una de las cabañas de la escena, al entrar, se activa un snapshot que incrementa la reverberación de los ladridos de un perro.

Al salir, la reverberación se reduce gradualmente mediante una curva de automatización en *FMOD*

Figura 15

Zonas invisibles con triggers



Fuente. Elaboración propia

Activación de un Event Emitter con Trigger

Se diseñó un evento sonoro para señalar al jugador el descubrimiento y recolección de un coleccionable o easter egg dentro del videojuego. Al acercarse al objeto, una música 3D que indica su presencia desaparece mediante un trigger en un objeto transparente alrededor del coleccionable (en este caso, una pizza). Este mismo trigger activa un efecto sonoro que confirma la recolección del ítem, el cual se reproduce una única vez.

Figura 16

Trigger once



Fuente. Elaboración propia

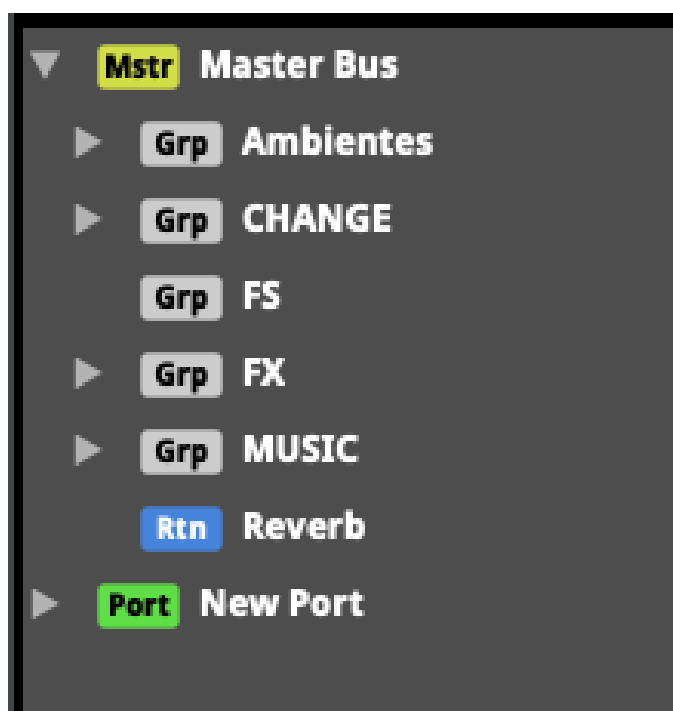
Tarea 5: Mezcla, Pruebas y Optimización de Audio del Videojuego

En la Tarea 5, se llevó a cabo el proceso de mezcla, prueba y optimización del diseño sonoro para el videojuego desarrollado en las etapas anteriores. Este trabajo incluyó una revisión de todos los event emitter y todas las interacciones de *FMOD* con *Unity*.

Para la mezcla de audio, se utilizó la ventana Mixer en *FMOD*, organizando los eventos sonoros en canales de grupo específicos, como ambientes, efectos y música. Esto permitió un control más detallado de la mezcla y facilitó la aplicación de envíos a buses auxiliares y el uso de efectos compartidos.

Figura 17

Grupos de mezcla



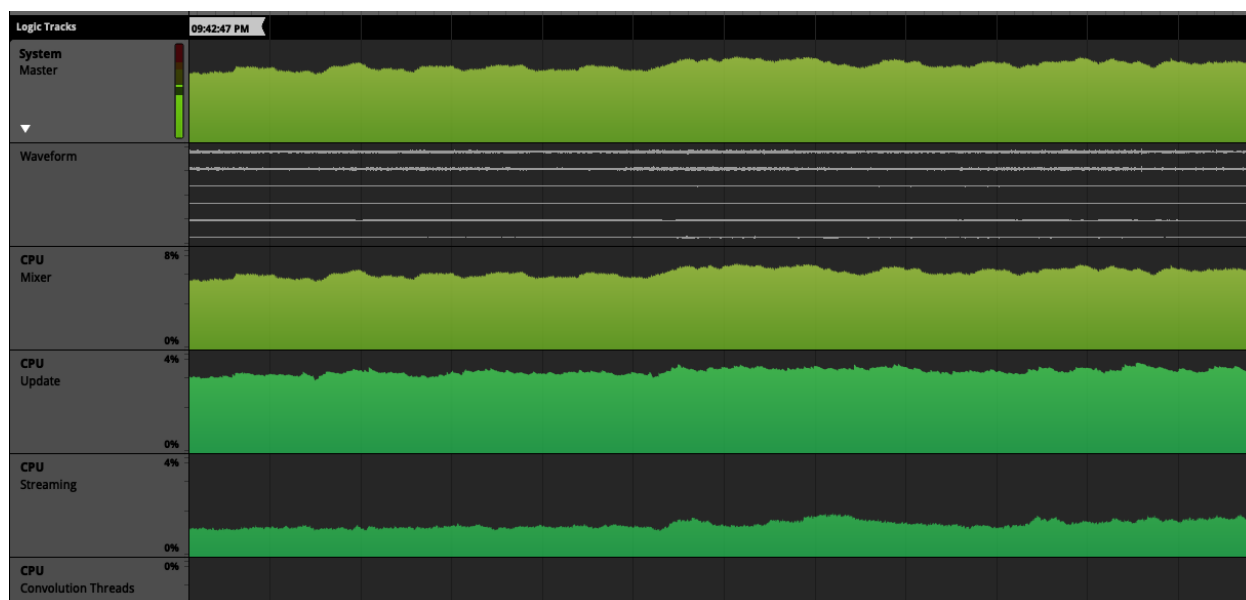
Fuente. Elaboración propia

Durante el desarrollo, se empleó la función de Live Update para sincronizar *FMOD* con *Unity* y monitorear los niveles sonoros en tiempo real mientras el videojuego estaba en ejecución. Esto permitió realizar ajustes precisos en la mezcla y ajustar cualquier inconsistencia de los eventos sonoros dentro del entorno de juego.

Por último, en la vista Profile de *FMOD*, se efectuó una grabación del rendimiento del proyecto para identificar posibles problemas, como clipping o un alto consumo de recursos de CPU y memoria.

Figura 18

Profile de rendimiento



Fuente. Elaboración propia

Anexos

Link sobre explicación tarea 5 y resultado final:

<https://youtu.be/XzvcPNdT4HY>

Juego ejecutable para MacOS Apple silicon:

<https://unadvirtualedu->

[my.sharepoint.com/:u:/g/personal/jsubaqueg_unadvirtual_edu_co/EcXciP4eB_VBn9nngg](https://unadvirtualedu-my.sharepoint.com/:u:/g/personal/jsubaqueg_unadvirtual_edu_co/EcXciP4eB_VBn9nngg)

[PWDzMBpiQrVSL9OMZ4qTZFFy9enw?e=fS4kUJ](https://unadvirtualedu-my.sharepoint.com/:u:/g/personal/jsubaqueg_unadvirtual_edu_co/EcXciP4eB_VBn9nngg-PWDzMBpiQrVSL9OMZ4qTZFFy9enw?e=fS4kUJ)

Conclusiones

El presente informe abarcó el desarrollo del diseño sonoro para una escena de un videojuego integrando diversos elementos sonoros y dinámicos, desde efectos ambientales y música adaptativa hasta sistemas de interacción y automatización mediante *FMOD Studio*.

El presente ejercicio destaca la importancia de la interactividad y la variabilidad en los elementos sonoros de un videojuego. Para futuros ejercicios, se recomienda prestar especial atención a la sincronización y la relación de los efectos sonoros con la jugabilidad.

El uso de *FMOD Studio* permitió una mayor comprensión sobre cómo los elementos sonoros pueden transformar la jugabilidad de un videojuego. La implementación de filtros, snapshots y el enrutamiento de canales para controlar el audio, proporcionó una nueva perspectiva sobre el diseño sonoro y su impacto directo en la experiencia del jugador.

Bibliografía

Moreno, R. (2021). *Tutorial de diseño de eventos sonoros dinámicos en FMOD Studio*. Game Audio.

https://ramorenov.github.io/gameaudio/FMOD/01_Fmod_evento_din%C3%A1mico.html

Moreno, R. (2021). *Tutorial emisores de eventos 3D con Fmod en una escena de Unity*. Game Audio. https://ramorenov.github.io/gameaudio/03_Fmod_emisores_de_eventos.html

Moreno, R. (2021). *Tutorial música adaptativa en FMOD Studio*. Game Audio.

https://ramorenov.github.io/gameaudio/FMOD/02_Fmod_musica_adaptativa.html

Moreno, R. (2021). *Tutorial postproducción y testeo de eventos sonoros en FMOD para una escena de Unity*. Game Audio.

https://ramorenov.github.io/gameaudio/07_Fmod_postproducci%C3%B3n_y_testeo.html

Moreno, R. (2021). *Tutorial sonido ambiente y eventos 3D con FMOD en una escena de Unity*. Game Audio.

https://ramorenov.github.io/gameaudio/03_Fmod_emisores_de_eventos.html

Moreno, R. (2021). *Tutorial triggers y parametros en FMOD para una escena de Unity*. Game Audio. https://ramorenov.github.io/gameaudio/04_Fmod_triggers_y_parametros.html

Villa, L. (2023). *Fundamentos del audio dinámico en videojuegos: Interactividad, adaptabilidad y variabilidad*. [Objeto_virtual_de_aprendizaje_OVA]. Repositorio Institucional UNAD.

<https://repository.unad.edu.co/handle/10596/58248>