

diseño sonoro dinámico para la sonorización del videojuego Free Low Poly Nature

Sofia Ayala Zaraza

Asesor

David Ricardo Agudelo Bernal

Universidad Nacional Abierta y a Distancia UNAD

Escuela de Ciencias Básicas, Tecnología e Ingeniería

Tecnología en Producción de Audio

Diplomado de profundización de audio dinámico para videojuegos

2025

Tabla de Contenido

Introducción.....	3
Objetivos.....	4
Objetivo general.....	4
Objetivos específicos.....	4
Fase 1 – Análisis de un videojuego.....	5-6
Fase 2 – Generación de un ambiente sonoro dinámico por medio de un middleware....	8-9
Fase 3 – Sonorización para un proyecto de videojuego.....	12-13
Fase 4 – Programación del diseño sonoro del videojuego.....	15-16-17
Fase 5 – Mezcla, pruebas y optimización del audio.....	21-22
Conclusiones.....	24-25
Referencias.....	26-27

Lista de Figuras

Figura 1. Mansión desde afuera	6
Figura 1.1. Visualización de los pasillos y Spooky	7
Figura 1.2. Personajes dentro del videojuego	7
Figura 2. Assets dentro de FMOD Studio	9
Figura 2.1. Eventos, carpetas y subcarpetas dentro de FMOD	10
Figura 2.2. Parámetro de lluvia configurado en FMOD Studio	11
Figura 2.3. Parámetro de intensidad aplicado en FMOD	11
Figura 3. Escena de bosque implementada en Unity	13
Figura 3.1. Integración de FMOD y Unity mediante Event Emitter	14
Figura 3.3. Atenuación por distancia para eventos 3D en FMOD	14
Figura 4. Cambio de estado de tensión mediante Trigger y zona de lluvia	18
Figura 4.1. Automatización de un filtro para simular oclusión en zona con Trigger ..	18
Figura 4.2. Activación de zona de reverberación mediante Snapshots	19
Figura 4.3. Activación de sonido de aves con Event Emitter	19
Figura 4.4. Creación de filtro y Snapshot en FMOD Studio	20
Figura 4.5. Parámetros de oclusión y filtro aplicados en FMOD	21
Figura 5. Vista general del Mixer en FMOD Studio	22
Figura 5.1. Ventana Profiler utilizada para la optimización	23
Figura 5.2. Proceso de exportación del videojuego a formato ejecutable en Unity ..	24

Introducción

El audio dinámico para videojuegos es un campo muy amplio y explorable en diferentes ámbitos y momentos, con la sonorización de un proyecto podemos crear un mundo inmersivo y diferente por medio del audio, con la gran posibilidad de crear experiencias inmersivas y realistas con el fin de que le jugador pueda sentirse dentro del juego, ya que el audio o sonorización en conjunto de la parte visual provocan emociones realistas que trascienden tras una pantalla lo cual es lo que buscamos por medio del diplomado de audio Dinámico para videojuegos.

En el proceso y desarrollo del diplomado exploramos y manejamos diferentes técnicas fortaleciendo habilidades necesarias para poder crear y realizar un proyecto dinámico para un videojuego, donde sale a flote nuestra parte creativa y contextual.

Entonces en este informe daremos a conocer cada una de las herramientas, técnicas y procesos utilizados para realizar una sonorización de una escena de un videojuego, para eso necesitamos dos cosas muy importantes y donde todo se desglosa para el trabajo en un middleware y motor de videojuegos, en este caso fmod el cual nos permite crear, controlar y optimizar un audio dinámico e interactivo y unity que nos permite, crear programar y ejecutar un videojuego.

Con esto finalizaríamos con la contextualización del proceso del audio dinámico de un videojuego donde se enfrentan diferentes adversidades durante el proceso, un camino experimental pero con un margen de aprendizaje exorbitante.

Objetivos

Objetivo General

Desarrollar un proyecto de diseño sonoro para videojuegos diseñando e implementando técnicas avanzadas de audio dinámico e interactivo mediante el uso de middleware y un motor de videojuegos además de herramientas especializadas, optimizando la inmersión y adaptabilidad con el fin de mejorar la calidad sensorial de la experiencia del jugador y la narrativa en los entornos virtuales.

Objetivos Específicos

Estudiar y reconocer la relevancia del audio Dinámico para videojuegos, jugabilidad, limitaciones e impacto.

Explorar y aplicar técnicas avanzadas de diseño sonoro para mejorar la adaptabilidad del audio dinámico.

Desarrollar e Implementar herramientas como middleware y motores de audio para optimizar la integración del sonido en entornos interactivos virtuales.

Integrar el middleware y el motor de videojuegos para la sincronización de eventos.

Implementar eventos RTPC para implementación en una escena.

Ajustar y programar cada uno de los eventos sonoros realizados dentro de la escena.

Realizar una mezcla y testeo final para optimizar condiciones que cumplan con estándares esperados.

Desarrollo del Diseño Sonoro del Videojuego Free Low Poly Nature

Fase 1

Análisis de un Videojuego

En la primera fase realizamos una previa investigación para analizar el comportamiento de un video juego, para analizar sus características componentes, como se ve reflejado en el videojuego su diseño sonoro, fecha de fabricación, proveedores y todos los datos técnico, diseño visual y constitución que puedan apoyarnos en nuestra investigación previa a nuestra propia realización de nuestro proyecto y como referencia utilizamos en videojuego spooopy's jumps scare mansión.

Spooopy's Jump Scare Mansión

Este videojuego es del genero de horror psicológico y indie fue desarrollado por la empresa lag Studios su lanzamiento fue en 2014 libre para descargar en pc y gratis en la plataforma de Steam y como motor desarrollador utilizaron unity.

La historia de este videojuego se basa en una mansión controlada con una niña fantasma llamada spooko, el videojuego está ambientado en una mansión embrujada llena de pasillos y puertas con sustos repentinos que a medida que el jugador avanza se vuelve más intrigante y aterrador lo cual puede causar susto repentinos al jugador, el objetivo es cruzar y sobrevivir a las 1000 habitaciones de la mansión.

Este videojuego cuenta con una perspectiva en primera persona de caminar, explorar, recibir sustos, evadir enemigos y buscar resguardo en teoría sobrevivir a las 1000 habitaciones sin morir, con enemigos llamados Specimens.

Este videojuego cuenta con un diseño sonoro de música ambiental dinámica que puede cambiar de relajante a inquietante, efectos de sonido súbitos, ruidos característicos a los

personajes y ambientación de pasos, puertas ecos y demás, en cambio a su estilo visual tiene gráficos reto pero fase a lo que podemos llamarle cute cuenta con un diseño engañoso con cierto encanto.

El estudio de este videojuego nos da una apertura a cómo debemos hacer nuestro propio trabajo y como deberíamos trabajar, es como abre bocas a los resultados que nosotros mismos queremos conseguir.

Figura 1

Mansión desde afuera.



Fuente: Google fotos (2025)

Figura 1.1

Visualización de los pasillos y spooky.



Fuente: Google fotos (2025)

Figura 1.2

Personas dentro del videojuego.



Fuente: Google fotos (2025)

Fase 2

Generación de un Ambiente Sonoro Dinámico por Medio de un Middleware

Para realizar este ambiente sonoro como primer paso es identificar el tipo de escena que vamos a sonorizar en este caso la escena escogida tiene que ver con bosque, establecido esto, tenemos que descargar nuestro middleware para la actividad utilizamos FMOD es una plataforma muy completa y versátil para trabajar audio para un videojuego en su última versión 2.03.1.

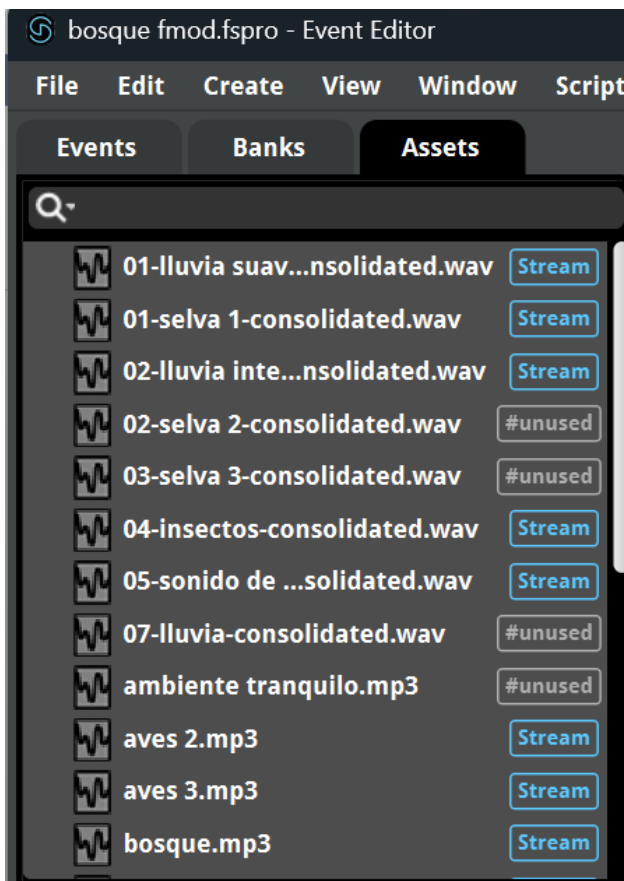
A continuación creamos nuestro ambientes sonoros que vayamos a utilizar como ambientes, efectos, música y demás, para este paso podemos crear nuestro propios sonidos por medio de técnicas como el Foley, o descargar sonidos libre de regalías por medio de bancos o librerías de sonidos gratuitos que fue lo que utilizamos en esta ocasión utilizar un banco de sonidos para adquirir nuestros sonidos a utilizar, ya que nuestra escena es un bosque, tenemos sonidos de lluvia, algunos animales, riachuelos y algunos sonidos ambientes, y con nuestros sonidos listos directo para utilizar en fmod.

Dentro de fmod se deben colocar los sonidos dentro de los assets para luego crear eventos sonoros y carpetas y subcarpetas según la distribución y organización que le queramos dar el proyecto, dentro de los eventos sonoros y los efectos sonoros que queremos conseguir se debe aplicar diferentes parámetros como por ejemplo la lluvia realizar un parámetro que me ayude a cambiar la intensidad de la lluvia esto lo podemos conseguir primero con varias muestras de sonido de lluvia se establece un parámetro de intensidad de 0 a 100 esto ayuda a que a que desde ese rango varíe la cantidad de lluvia que podemos escuchar dentro de una zona específica.

Con esto ya tendríamos el primer punto clave dentro de nuestro objetivo del diplomado lo cual tener los eventos sonoros dentro de un middleware de fmod organizados y listos para llevar al motor de videojuegos Unity.

Figura 2

Assets dentro de fmod.



Fuente: elaboración propia (2025)

Figura 2.1

Eventos, carpetas y subcarpetas dentro de fmod.



Fuente: elaboración propia (2025)

Figura 2.2

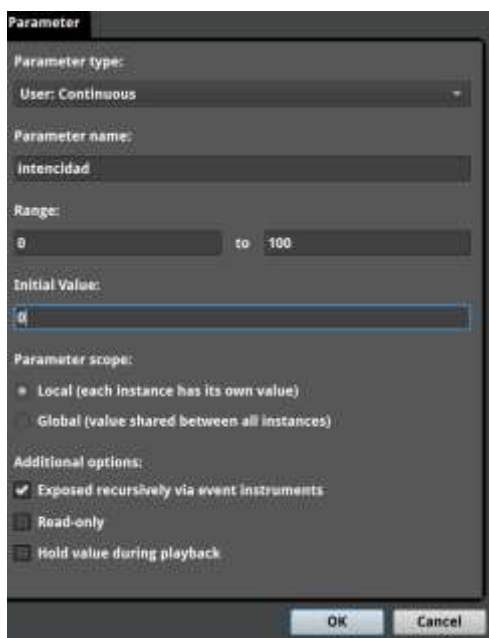
Parámetro de lluvia.



Fuente: elaboración propia (2025)

Figura 2.3

Parámetro de intensidad.



Fuente: elaboración propia (2025)

Fase 3

Sonorización Para un Proyecto de Videojuego

Para lo sonorización de un videojuego lo primordial es crear un proyecto desde cero por medio del unity hub, para esto utilizamos la versión de unity 6.0 para que fuera compatible con fmod ya que es la versión 2.03, en el unity hub creamos un proyecto nuevo uno tipo 3D, una vez el proyecto creado, tenemos que trabajar con una escena como el tema escogido fue bosque, desde la página principal de unity descargamos la escena de bosque de free low poly nature forest es una escena sencilla con algunos árboles, varios caminos, unos pequeños riachuelos y algunas zonas verdes.

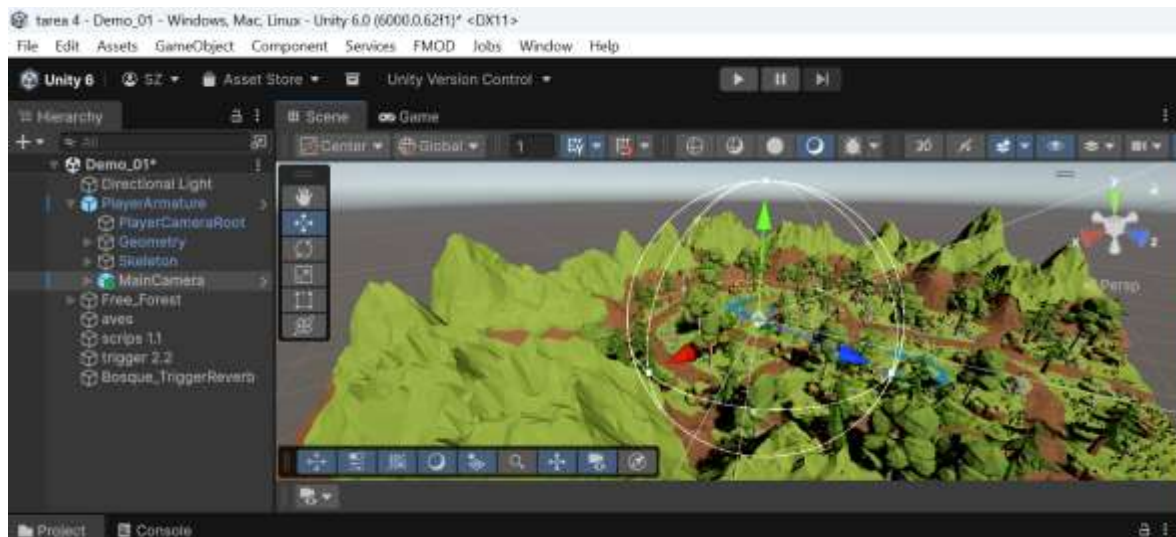
Con la escena montada y el proyecto en fmod hacemos una integración por medio del paquete Fmod for unity lo que nos ayuda hacer la integración directa con su versión 2.03.09 con estas versiones no tendremos problemas técnicos ya que todas trabajan de manera sincrónica al ser versiones de las más recientes, lo que hace el paquete fmod for unity es que nos ayuda que dentro de unity podamos trabajar directamente el listened de fmod y quitar el de unity, entonces toda la parte auditiva que queramos en unity está directamente integrada con el trabajo sonoro de fmod.

Con la integración echa podemos poner sonidos por medio del fmod studio listener que es el audio de Fmod y con el fmod studio event emitter el cual es nuestro principal elemento para poner la parte sonora ya que desde aquí podemos buscar cada uno de los eventos sonoros creados en fmod y además arreglar o establecer pequeños factores como las atenuación por distancia para cada uno de los sonidos dentro de la escena. Con esto ya tendríamos fmod con unity integrados

podemos trabajar de manera sincrónica y sencilla claramente reconociendo cada uno de los factores antes mencionados y con esto ya tenemos gran parte de nuestro trabajo hecho.

Figura 3

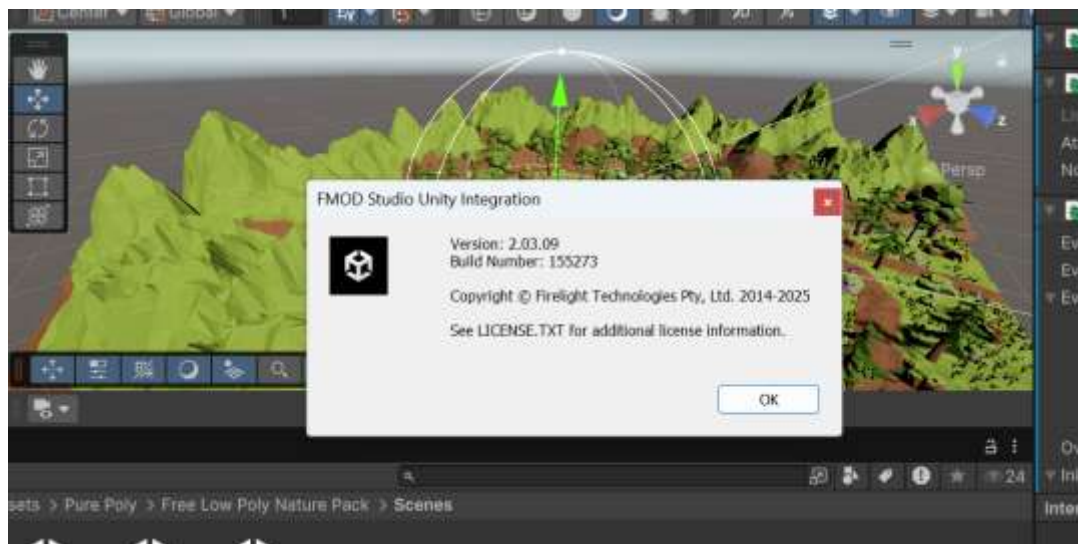
Escena de bosque unity.



Fuente: elaboración propia (2025)

Figura 3.1

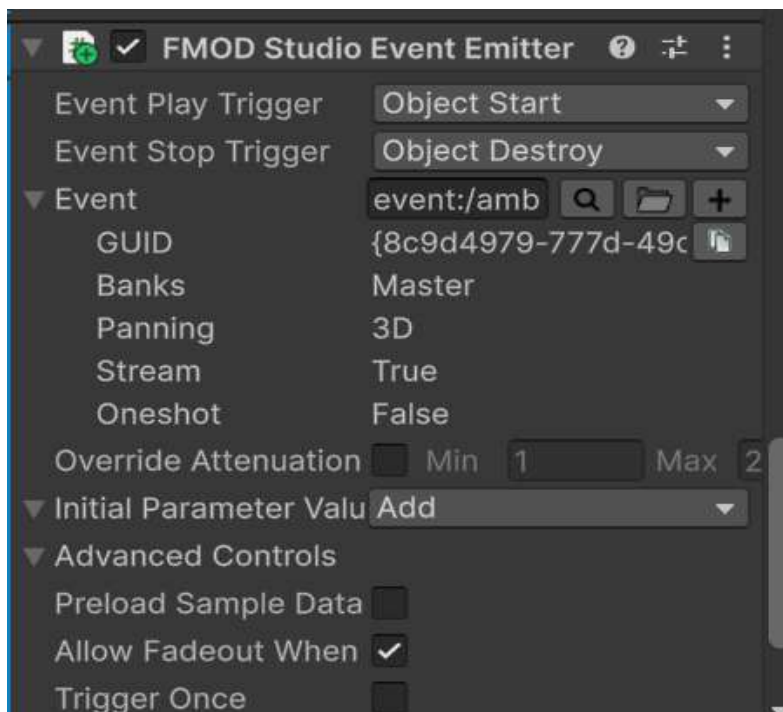
Integración de fmod y unity.



Fuente: elaboración propia (2025)

Figura 3.2

Fmod studio event emitter.



Fuente: elaboración propia (2025)

Figura 3.3

Atenuación por distancia.



Fuente: elaboración propia (2025)

Fase 4

Programación del Diseño Sonoro del Videojuego

Para la programación de un diseño sonoro de videojuegos necesitamos un factor importante el cual es un player, un jugador que podamos mover dentro de toda la escena y recorrerla y activar los parámetros que se van a programar, para este proyecto trabajamos con el paquete de starter assets third person controller, este paquete lo podemos adquirir directamente desde la página principal de unity el cual trae un controlador en tercera persona, a lo que nosotros llamamos un player para mover dentro de la escena que va dentro de la cámara de nuestra escena la misma cámara donde se encuentra fmod.

Con lo mencionado anteriormente tuve una experiencia muy particular nosotros estamos estudiando para ser productores de audio, a nosotros se nos sale de las manos conocimientos de programación por ejemplo porque no es nuestra área de trabajo durante esta fase en particular tuve un problema técnico de acuerdo a programación o relación con la parte visual de la escena donde primero la cámara de la escena no aceptaba nuestro controlador en tercera persona, el player descendía o flotaba o simplemente toda la escena desaparecía en el modo Game y viene la pregunta que voy hacer si esto sale fuera de mi alcance y mis conocimientos, pero en el momento tenemos muchas herramientas que nos pueden ayudar como son las inteligencias artificiales como copilot, gemini o chat gpt que son una gran fuente de apoyo si las utilizamos correctamente para solucionar el inconveniente por medio de estas IA les comente la situación paso por paso con envío de imágenes hasta que ellas me ayudaron a resolver el problema y mi cámara de la escena por fin acepto el personaje y lo puede mover por la escena según lo

indicado, comento esto porque en un campo de trabajo real tendremos un técnico o un programador que nos resuelva el problema, pero mientras estamos en este proceso de aprendizaje podemos utilizar herramientas como las IA que nos facilitan el aprendizaje y nos ayudan a resolver inconvenientes durante el proceso.

Continuando por la parte de esta fase el principal objetivo es aplicar automatizaciones, efectos so oros por medio de Triggers y Snapshots estos creados tanto en fmod como en unity los trigger en unity y los Snapshots en fmod y cada uno de los eventos sonoros que utilizaremos en los Triggers, esto lo establecemos con el fin de darle más dinamismo en las escenas y hasta realismo ya que son mecanismos de interacción dentro del videojuego.

Entonces se programan eventos sonoros con dos scripts principalmente el trigger enter y el trigger exit, por medio de estos podemos realizar procesos en cambios de tensión, automatización por medio de filtros, poner zonas de reverberación y demás y activar trigger es nuestro event emitter de fmod.

Los parámetros programados están distribuidos de la siguiente manera:

Cambio de Estado de Tensión

Para el cambio de estado de tención utilice el parámetro de lluvia antes mencionado lo que hicimos fue seleccionar una parte de nuestra escena por medio de un fmod studio event emitter seleccionamos el evento lluvia y el parámetro de intensidad, en nuestra jerarquía se crea un cubo y se pone exactamente sobre la parte seleccionada que tiene nuestro evento sonoro entonces en el cubo por medio del inspector ponemos el trigger enter y el trigger exit con el parámetro seleccionado con el fin de que cuando nuestro personaje entre a la zona la lluvia se escuche suave y cuando salga de esta se escuche fuerte.

Automatización de un filtro que Simule Oclusión

Como primer paso es seleccionar que sonido vamos a utilizar en este caso el sonido de riachuelo y en la escena la parte donde está el riachuelo, entonces en fmod creamos un parámetro de oclusión para nuestro sonido de aves con cambiante de 1 a 2 y además ponemos un filtro de low pass lo que esto hace en nuestra escena es que cuando nuestro personaje entre dentro del riachuelo se escuche suave y cuando salga se escuche mas fuerte pero se vaya desvaneciendo y se escuche más lejos y esa es la función el filtro y el parámetro.

Activar una Zona de Reverberación Mediante Snapshots

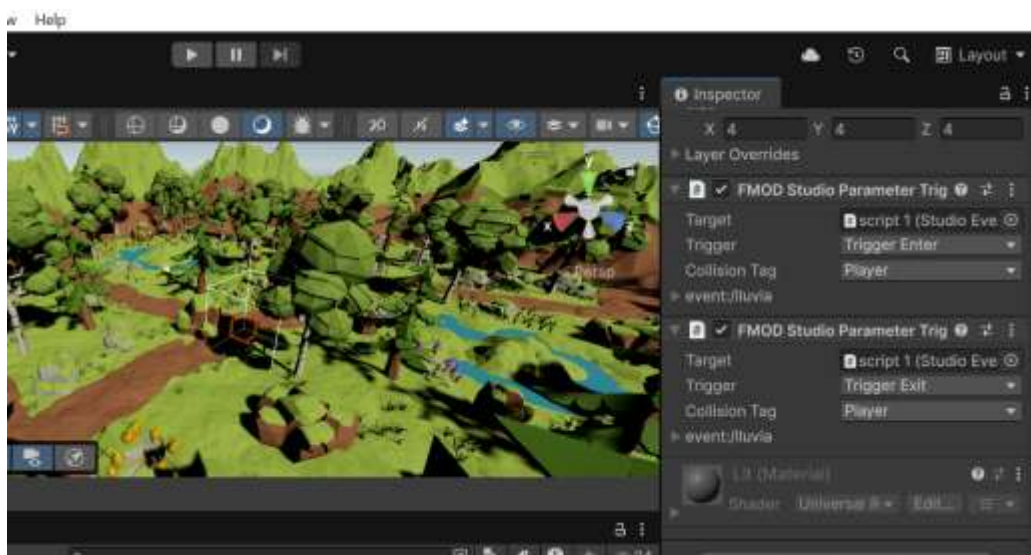
Para esto en fmod studio seleccionamos cada uno de los sonidos con los cuales trabajamos menos la música en la parte de routing creamos un grupo con los sonidos y adicional fuera del grupo creamos un retorno el cual aplicamos una reverb, y en la parte de Snapshots todo esto en la parte de la mixer de nuestro proyecto de fmod tenemos la Snapshots la cual nos va a aplicar el reverb por medio del retorno a cada uno de los eventos seleccionados, ya en unity creamos un cubo que cubra la zona donde están los sonidos que serán afectados por la reverb con la aplicación de los Triggers de enter y exist ya que cada vez que el personaje entre o salga de la zona de reverberación active la zona.

Activación de un Event Emitter

Para este utilizamos sonidos de aves en un punto específico seleccionamos el sonido por medio del fmod event emitter mediante trigger, seleccionamos la zona ponemos atenuación por distancia y cada vez que el personaje se acerque o se aleje de la parte seleccionada el canto de las aves disminuye y aumenta además en fmod el sonido de a veces esta hecho por medio de un scatterer instrumen que reúne varias pistas y las mezcla y reproduce aleatoriamente lo que ayuda a dar dinamismo

Figura 4

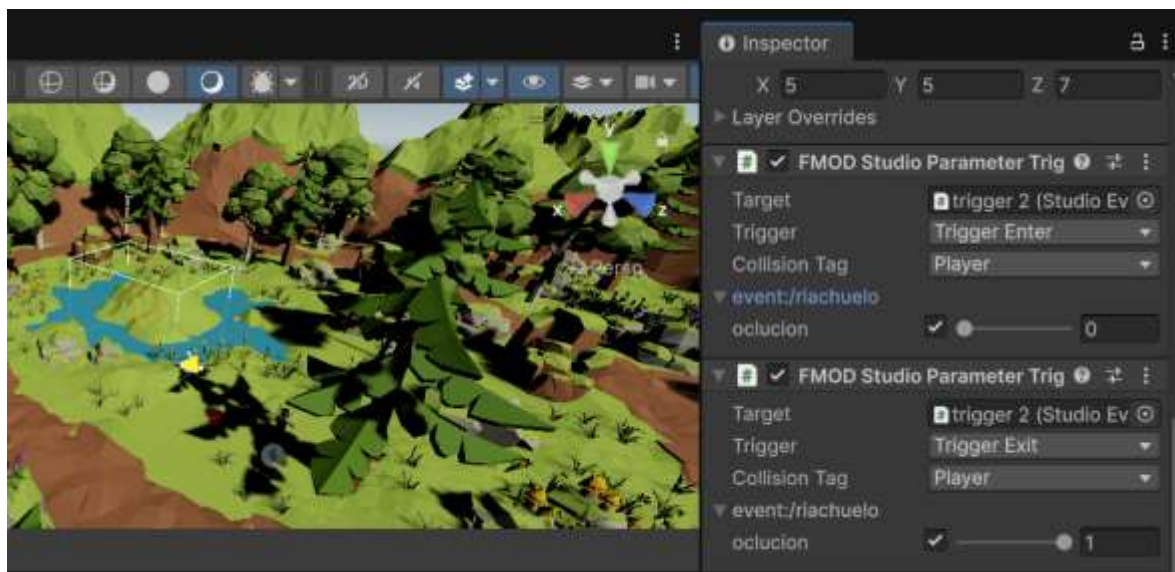
Cambio de estado de tención trigger y zona seleccionada, lluvia.



Fuente: elaboración propia (2025)

Figura 4.1

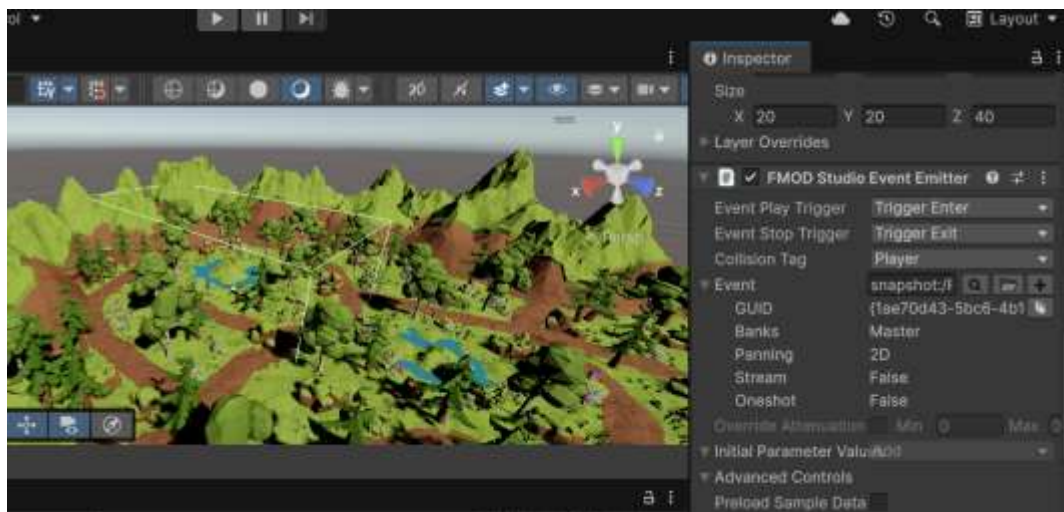
Automatización de un filtro que simule oclusión, zona y trigger, riachuelo.



Fuente: elaboración propia (2025)

Figura 4.2

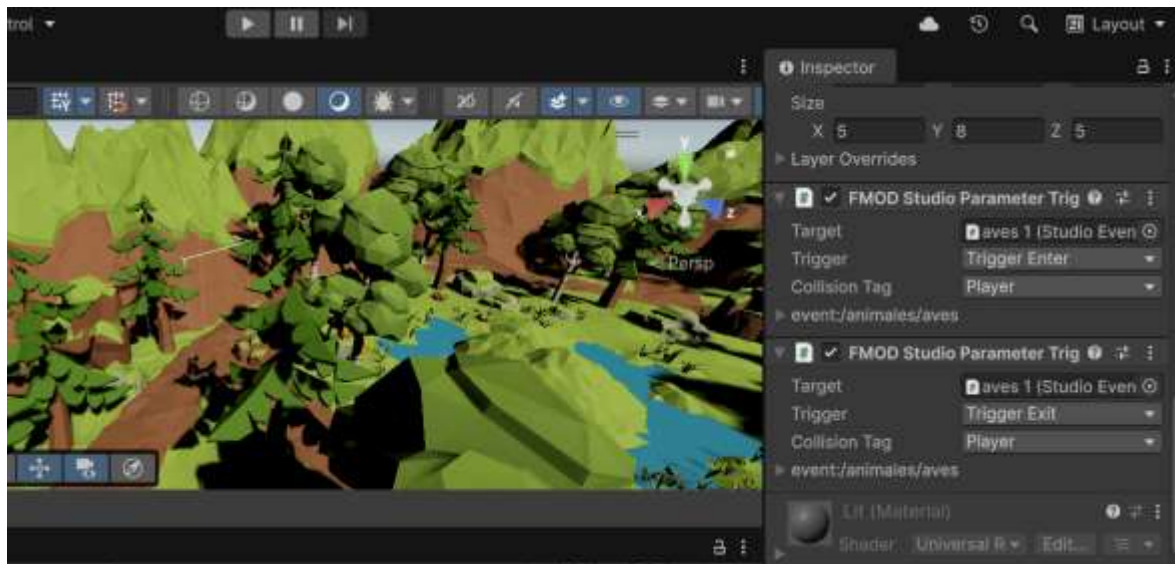
Activar una zona de reverberación mediante Snapshots, zona y Triggers.



Fuente: elaboración propia (2025)

Figura 4.3

Activación de un event emitter, zona y trigger, aves.



Fuente: elaboración propia (2025)

Figura 4.4

Creación de filtro y Snapshots fmod.



Fuente: elaboración propia (2025)

Figura 4.5

Parámetro de intensidad de la lluvia en fmod.



Fuente: elaboración propia (2025)

Figura 4.6

Parámetro de oclusión y filtro.



Fuente: elaboración propia (2025)

Fase 5

Mezcla, Pruebas y Optimización de Audio del Videojuego Free Low Poly Nature

En la fase 5 se realiza un análisis general de cada uno de los sonidos y como es su comportamiento grupal durante la reproducción del videojuego todo esto con el fin de poder realizar una optimización del producto sonoro en este caso, cada uno de los sonidos trabajados en fmod studio, en la optimización lo que hacemos es realizar grupos y sub grupos en la mixer y los eventos sonoros, realizar limpieza de las cosas que ya no utilizamos, un vez los grupos creador utilizamos el live update reconnecting esta función nos permite ver el comportamiento de los sonidos mientras están en el modo Game dentro del proyecto de unity, esta función nos ayuda para poder realizar la mezcla y saber que efectos o buses debemos agregar para dar una mejor calidad del sonido, además de esta función tenemos otra manera de optimizar y saber que se debe corregir en la mezcla con la ventana de profile donde podemos hacer un testeo por medio de la

grabación de los sonidos mientras se reproducen en el modo Game, esta grabación se hace para realizar el testeo y conocer el estado de CPU que se consume cada uno de los sonidos, una vez conociendo el estado por medio de las funciones , y ya teniendo los faders de cada grupo realizamos una mezcla de lo que consideremos conveniente y con esto en unity ya podemos construir un archivo ejecutable exe. Que nos da una vista de como se ve el videojuego exportado listo para su comercio en plataformas ya sea Windows, Mac o para teléfono y con esto concluimos la realización de crear un diseño sonoro dinámico para videojuegos.

Figura 5

Vista de la mixer en fmod.



Fuente: elaboración propia 2025

Figura 5.1

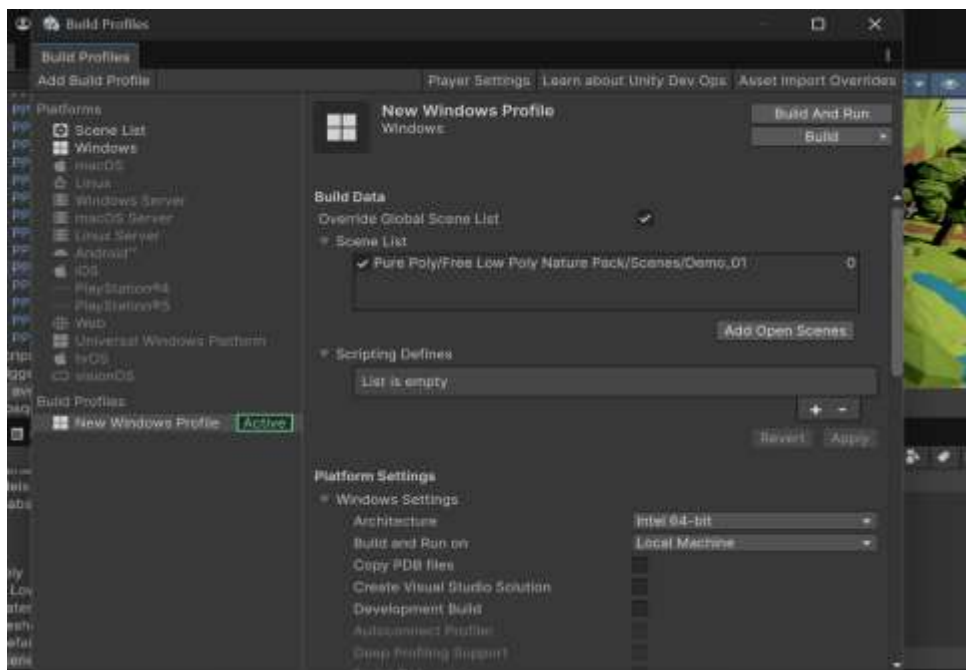
Ventana profiler en fmod.



Fuente: elaboración propia (2025)

Figura 5.2

Vista para la exportación del archivo en EXE.



Fuente: elaboración propia (2025)

Conclusiones

Destacando que durante el desarrollo del diplomado pudimos comprender y aplicar de manera clara y de forma práctica conceptos y fundamentos del audio dinámico para videojuegos en los cuales fueron técnicos, creativos y de forma metodológica, durante todo el proceso para lograr construir un diseño sonoro de carácter interactivo que lograra responder coherentemente con la escena y satisfacer las necesidades de la industria y jugadores y expectativas planteadas en los objetivos.

Podemos evidenciar por medio de nuestros resultados obtenidos utilizando el middleware de fmod con el motor de videojuegos de unity que unidos, son dos herramientas donde podemos crear experiencias, por medio de la creación de eventos sonoros, parámetros, filtros, atenuaciones, optimizaciones, lo cual llamamos la creación del ambiente sonoro capaz de responder y adaptarse a sensaciones de realismo y dinamismos dentro de un entorno virtual.

De igual manera el proceso de crear un diseño sonoro de carácter dinámico, identificamos factores importantes como la forma correcta de organizar el proyecto para poder evaluar el rendimiento, los niveles y poder reducir el consumo de recursos, además durante el trabajo interdisciplinario podemos tener desafíos técnicos, de integración y programación que por medio de consulta de documentación y recursos digitales y herramientas de inteligencia artificial, se puede conseguir fortalecer el aprendizaje y resolución de problemas.

Finalmente con este proyecto podemos afirmar que el diseño sonoro para video juegos es un campo diversamente explorable, un campo muy importante y esencial dentro de la industria interactiva ya que brinda un impacto emocional al jugador y fomenta a la capacidad de enriquecer la narrativa y construir mundos inmersivos, concluyendo podemos decir que para futuras líneas de trabajo, explorar sistemas adaptativos mas complejos, ir avanzando en las

nuevas versiones con las que podemos trabajar y el estudio de nuevas técnicas que nos brinden más especialización, con esto a futuro podemos formar líneas de trabajo más integrales, crear nuevas habilidades técnicas y fortalecer las conocidas con el fin de crear ejercicios más hacia lo profesional y requerido en el ámbito del diseño sonoro dinámico para video juegos.

Referencias Bibliográficas

- Berndt, A., & Hartmann, K. (2020). *Adaptive game music: Theoretical and practical considerations*. *Journal of Game Sound and Music*, 3(2), 45–63.
<https://doi.org/10.xxxx/yyyy>
- Castillo Ávila, R. A. (2023). *Diplomado de profundización en audio dinámico para videojuegos* [Informe de diplomado, Universidad Nacional Abierta y a Distancia]. Repositorio Institucional UNAD.
<https://repository.unad.edu.co/jspui/bitstream/10596/56589/1/racastilloav.pdf>
- Collins, K. (2008). *Game sound: An introduction to the history, theory, and practice of video game music and sound design*. The MIT Press.
- Ekman, I., & Lankoski, P. (2009). Meaningful noise: Understanding sound effects in video games. *Game Studies*, 9(1). https://www.gamestudies.org/0901/articles/ekman_lankoski
- Game Jolt. (2014). *Spooky's Jump Scare Mansion* (Akuma Kira).
<https://gamejolt.com/games/spooky-s-jump-scare-mansion/36962>
- Garzón Henao, Y. (2023). *Audio dinámico para videojuegos* [Trabajo de grado, Universidad Nacional Abierta y a Distancia]. Repositorio Institucional UNAD.
<https://repository.unad.edu.co/bitstream/handle/10596/60080/ygarzonhe.pdf?isAllowed=y&sequence=1>
- 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 de integración de FMOD con una escena de Unity*. Game Audio.
https://ramorenov.github.io/gameaudio/01_IntegracionfmodUnity.htm

Moreno, R. (2021). *Tutorial emisoras 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 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 Viasus, R. (2017, marzo 10). *FMOD Studio: Event Instrument* [Archivo de video].

<http://hdl.handle.net/10596/11794>

Moreno Viasus, R. (2017, marzo 10). *FMOD Studio: Introducción* [Archivo de video].

<http://hdl.handle.net/10596/11795>

Moreno Viasus, R. (2017, marzo 10). *FMOD Studio: Multi Instrument* [Archivo de video].

<http://hdl.handle.net/10596/11788>

Moreno Viasus, R. (2017, marzo 10). *FMOD Studio: Single Instrument* [Archivo de video].

<http://hdl.handle.net/10596/11789>

Steam. (2015). *Spooky's Jump Scare Mansion* (original). Valve Corporation.

https://store.steampowered.com/app/356670/Spookys_Jump_Scare_Mansion/

Steam. (2017). *Spooky's Jump Scare Mansion: HD Renovation*. Valve Corporation.

https://store.steampowered.com/app/577690/Spookys_Jump_Scare_Mansion_HD_Renovation/

TV Tropes. (s. f.). *Spooky's Jump Scare Mansion*.

<https://tvtropes.org/pmwiki/pmwiki.php/Videogame/SpookysJumpScareMansion>