

**Diseño sonoro dinámico para la sonorización de un video juego trabajado en Unity.**

Leonardo Manuel Ramírez González

80795298

Asesor:

David Ricargo Agudelo Bernal

Universidad Nacional Abierta Y A Distancia – UNAD  
Escuela De Ciencias Básicas De Tecnología e Ingeniería – ECBTI  
Tecnología En Producción De Audio  
Diplomado En Diseño Sonoro Para Video Juegos  
Bogotá D.C. 2025

## Contenido

Introducción .....	7
objetivos .....	8
tarea 1- análisis de un video juego: genshin impact.....	9
tarea 2: eventos de diseño sonoro en audio middleware: generación de un ambiente dinámico selvático.....	15
tarea 3- integración de audio middleware con motor de video juegos.....	25
tarea 4- triggers, snapshots y modificación continua de parámetros de audio.....	36
tarea 5- mezcla y finalización de un proyecto de sonorización de video juego.....	43
conclusiones .....	49
referencias.....	50

## Índice de Figuras

<b>Figura 1</b> Presentación del video juego Gensin Impact .....	9
<b>Figura 2</b> Escena de diálogo de tipo opción múltiple presentado en el video juego Genshin Impact .....	11
<b>Figura 3</b> Vista del entorno o ambiente en el video juego Genshin Impact .....	12
<b>Figura 4</b> Escena con ambiente acción de batalla y uso de efectos SFX en partida del video juego Genshin Impact .....	13
<b>Figura 5</b> Vista del interfaz de ajustes del video juego Genshin Impact.....	14
<b>Figura 6</b> Imagen del programa Middleware Fmod Studio Versión 2.03.09.....	15
<b>Figura 7</b> Imagen de Fmod Studio en la pestaña Assets.....	17
<b>Figura 8</b> Imagen de Fmod Studio en la pestaña Events.....	18
<b>Figura 9</b> Imagen del evento personaje y sus diferentes pistas de audio con zonas de transición, marcadores, Single Instrument y loops junto con el parámetro Accion_Personaje.....	20
<b>Figura 10</b> Imagen del evento lluvia y sus diferentes pistas de audio con Single Instrument, Scatterer Instrument, Multi Instrument y loop, junto con el parámetro lluvia .....	20
<b>Figura 11</b> Imagen del evento Selva_Dia con Single Instrumentn, Scatterer Instrument, Multi Instrument, y Loop.....	21
<b>Figura 12</b> Imagen del evento Selva_Noche con Single Instruments y loop .....	21
<b>Figura 13</b> Imagen del evento principal Sonido_Selva, compuesta por Envent Instrument, Single Instrument y loop junto con los parámetros hora, Accion_Personaje y lluvia.....	22
<b>Figura 14</b> Imagen de la hoja de parámetros Acción Personaje .....	22
<b>Figura 15</b> Imagen de la Hoja de parámetros Hora .....	23
<b>Figura 16</b> Imagen de la hoja de parámetros para el evento lluvia .....	23

<b>Figura 17</b> Imagen de sitio web de Fmod Studio en su sección de descargas.....	25
<b>Figura 18</b> Imagen de librería de sonidos en pestaña asset de Fmod Studio.....	27
<b>Figura 19</b> Imagen de los grupos y eventos generados en Fmod Studio.....	27
<b>Figura 20</b> Imagen de la propiedad Fmod Studio Event Emitter en Unity y su configuración para el cubo Interior.....	32
<b>Figura 21</b> Imagen de la ubicación de los cubos Interior y Exterior en el juego. ....	32
<b>Figura 22</b> Imagen del objeto Radio y su configuración para la escucha a la distancia en 3D.....	33
<b>Figura 23</b> Imagen de la configuración gema y vista del total de gemas utilizadas en el juego ....	34
<b>Figura 24</b> Imagen del juego en acción.....	35
<b>Figura 25</b> Imagen de la configuración en Unity del área de Reverberación.....	37
<b>Figura 26</b> Imagen de la programación del Sanpshot en Fmod Studio .....	38
<b>Figura 27</b> Imagen de la ubicación del cubo de reverberación en el recinto.....	38
<b>Figura 28</b> Imagen de la configuración en Unity de la zona de alarma .....	39
<b>Figura 29</b> Imagen de la ubicación de cubo zona alarma en el recinto .....	40
<b>Figura 30</b> Imagen de configuración en Fmod Studio de la nueva pista de música utilizada en el juego.....	40
<b>Figura 31</b> Imagen de la configuración en Unity de las de cambio de tensión en el evento MusicaV2.....	41
<b>Figura 32</b> Imagen de las zonas de tensión para el evento MusicaV2 .....	42
<b>Figura 33</b> Imagen del mixer en Fmod Studio .....	43
<b>Figura 34</b> Imagen del Mixer en Fmod Estudio y mezcla de los niveles de audio de los eventos.....	44
<b>Figura 35</b> Imagen Surround 5.1 en Fmod Studio.....	44
<b>Figura 36</b> Imagen de mezcla de los eventos asignados a los grupos. ....	45

<b>Figura 37</b> Imagen del primero testeo de mezcla jugando en vivo en Unity .....	46
<b>Figura 38</b> Imagen de la herramienta Profiler de Fmod Studio .....	47

## Índice de Tablas

<b>Tabla 1</b>	Tabla de pre producción de audio dinámico para un ambiente selvático .....	16
<b>Tabla 2</b>	Tabla de referencia para la configuración del evento Música .....	28
<b>Tabla 3</b>	Tabla de referencia para la configuración del evento Golpe_Mueble .....	28
<b>Tabla 4</b>	Tabla de referencia para la configuración del evento Radio.....	29
<b>Tabla 5</b>	Tabla de referencia para la configuración del evento Recoger_Gema .....	29
<b>Tabla 6</b>	Tabla de referencia para la configuración del evento Interior – Exterior .....	30
<b>Tabla 7</b>	Tabla de referencia para la hoja de parámetro Interior – Exterior .....	30

## Introducción

La creación del diseño sonoro para video juegos, requiere ciertos conocimientos base para poder llevar a cabo un trabajo bien realizado sobre un juego, en donde se incluye el análisis del video juego, su historia, su intención, sus personajes entre otras muchas otras cosas y detalles.

Pero a lo que nos lleva este material de trabajo es sobre la herramienta que nos puede ayudar a prestar ese servicio y generar una buena banda sonora, si lo podemos decir así.

Fmod Studio es una herramienta ideal creada para este trabajo, nos permite generar diseño de audio dinámico, sincronizar efectos de sonidos, crear efectos de distancia 2D y 3D, y gracias a sus múltiples herramientas y con la visualización muy parecida a un DAW, puede llegar ser muy fácil su curva de aprendizaje.

Durante la próxima lectura daremos un vistazo rápido sobre el análisis de un video juego online, veremos herramientas del middleware Fmod Studio y como realizamos las diferentes configuraciones para poder generar un ambiente de selva, luego el diseño sonoro de un video Juego que nos ofrece Unity en su modo learning.

## **Objetivos**

### **Objetivo General**

Diseñar e implementar a través de la herramienta middleware Fmod Studio el diseño sonoro de un video juego desde cero en el motor de video juegos Unity, buscando generar una experiencia inmersiva y a su vez coherente con el juego.

### **Objetivos Específicos**

Diseñar el contenido sonoro dinámico para un video juego en Unity creando eventos de ambiente, música y de efectos especiales requeridos para el video juego

Aplicar de forma correcta herramientas como Single Instruments, Multi Instruments, Event Instruments y parámetros dentro del middleware Fmod Studio

Aplicar automatizaciones en los niveles de audio con Snapshots y la creación de grupos y canales de efectos con la herramienta mixer que ofrece Fmod Studio



## Tarea 1- Análisis de un video juego: Genshin Impact

El diseño sonoro de un video juego está constituido por diferentes tipos de audio que enriquecen la experiencia del usuario final o jugador

Estos tipos de audios se pueden clasificar según la función que tiene dentro del juego y de la que promete generar un tipo de sensación o sentimiento sobre el oyente buscando de tal forma que la partida sea más inmersiva, realista, atractiva, haciendo que la delgada línea de la ficción con la realidad pueda confundirse.

No hay mejor ejemplo que estudiar el diseño sonoro de un video juego que jugando una partida en tiempo real.

Para este caso analizaremos el diseño sonoro del video juego Genshin Impact cuya corporación se llama HoyVerse de origen chino fundada en 2012 por estudiantes japoneses.

**Figura 1** *Presentación del video juego Genshin Impact*



¿Qué tipo de juego es?

Es un RPG de mundo abierto online, se caracteriza por ser un video juego de aventura y acción fantásica dado a los diferentes personajes tipo anime que podemos encontrar. Al ser de mundo abierto se tiene la posibilidad de vagar por todo el mapa con plena libertad, lo que lo hace aún más enriquecedor ya que no se está atado a un camino fijo y donde se puede topar con diferentes criaturas o escenarios, sin embargo, se tiene misiones específicas que cumplir.

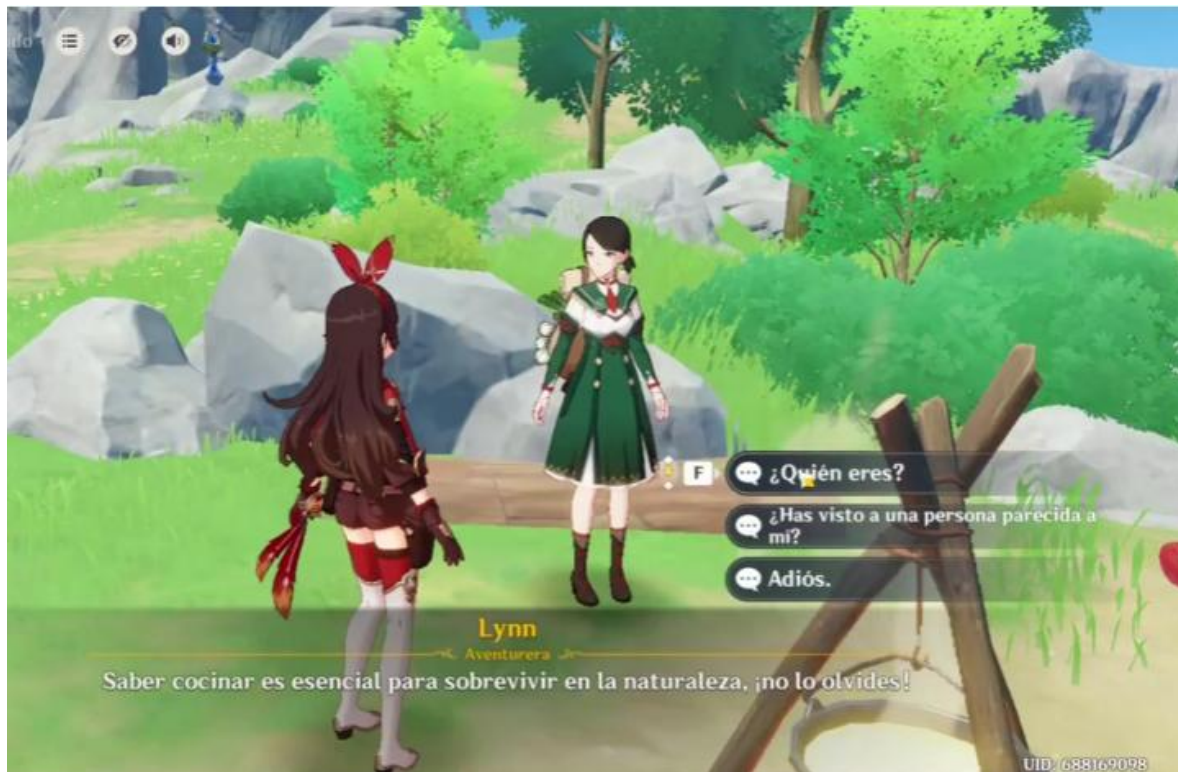
Este juego está disponible para plataformas como PS4, PS5, IOS, Android, y para PC lo podemos encontrar para Windows como es el caso que nos compete.

¿Qué tipo de audio encontraremos?

Durante el análisis se encontraron audios de tipo diálogo, tipo ambiente, efectos de sonido o FSX, música no diegética, audios de ambiente y de interfaz.

Audio de tipo diálogo: Para este caso el juego tiene escenas donde los personajes tienen diálogos o hablan entre sí. Este tipo de dialogo puede ser de tipo informativo o de solo entretenimiento, es decir, que existen diálogos que pueda darle pistas al jugador sobre temas específicos del juego o en su defecto es un diálogo que no tiene otra función que de entretener sin tener información valiosa o de referencia. Este tipo de momentos se presenta en este video juego en el formato tipo película, es decir que el jugador no ejecuta ningún comando mientras corre la escena y los diálogos pertinentes, pero también existe el de tipo interacción bajo opción múltiple, donde es posible escoger bajo varias opciones que queremos decir mientras se interactúa con otro personaje que no sea otro usuario humano.

**Figura 2** Escena de diálogo de tipo opción múltiple presentado en el video juego Genshin Impact



*Nota: Autoría propia.*

Ahora hablemos sobre aquellos sonidos que hacen únicos a los video juegos, estos son los Efectos o SFX: Este tipo de audio es diegético ya que pertenece o proviene del mundo del juego. En este caso se sospecha que está creado con evento 2D, ya que siempre lo estaremos escuchando frente a nuestro personaje y no cambia de distancia o posición. Lo podremos examinar cuando nuestro personaje, corre, salta, toma objetos, nada, grita, utiliza el arco y flecha, realiza poderes especiales, etc. Este tipo de audio es esencial que sea bien trabajado, ya que representa la acción que ejecuta el avatar y es el complemento a lo que vemos como jugador a través de nuestra pantalla. Si existiera incoherencias entre lo que vemos y escuchamos

generaría distracción y seguro pérdida de interés en el juego. Este ejemplo lo podremos observar en la figura 4.

Otro particular audio es el sonido ambiente: También es un audio diegético y se cree que está elaborado bajo eventos 3D, ya que podemos diferenciar de donde proviene y se percibe cierta distancia a la que nos encontramos del emisor. Estos sonidos reflejan el lugar donde nos encontremos dentro del juego, en este caso escucharemos sonidos típicos de la naturaleza como son los pájaros, el viento, los ríos entre otros. Su función principal es ubicarnos en el tiempo y espacio, este tipo de audio requiere de un trabajo muy elaborado, pues, así como los efectos de sonido, este tipo de audio coloca al oyente en el mundo. No sería correcto escuchar sonidos de naves espaciales o de un estadio de futbol si lo que estamos viendo es un bosque con muchos árboles y ríos. Aunque parece sencillo, lograr un diseño sonoro en ambiente también es un arte.

**Figura 3** Vista del entorno o ambiente en el video juego *Genshin Impact*



*Nota: Autoría propia*

Ahora pasemos acerca de la música en el video Juego. Este tipo de audio es no diegético ya que no pertenece al mundo del juego, pero acompaña o armoniza el estado del juego. Es creado en eventos tipo 2D, es decir que siempre lo estaremos escuchando durante toda la partida y no tiene un emisor de referencia por lo que no se aleja ni se acerca mientras estamos en la partida. En muchos casos este tipo de audio es dinámico o adaptativo ya que acompaña el momento o a la acción que está ocurriendo en el juego, por ejemplo si hay una batalla, el fondo musical cambiará a algo más tensionante o fuerte, con más dinámica o diferencias de tempo, esto permite al oyente diferenciar una escena tranquila a una escena de acción, no solo sirve como tipo ambiente sino que dinamiza y alienta al jugador a prepararse para la batalla, pone los sentidos más agudos y su atención incrementa; mientras que en estado de exploración o estado de calma la música será mucho más relajante y pasiva así mismo el jugador estará en otro estado de emoción de menos estrés.

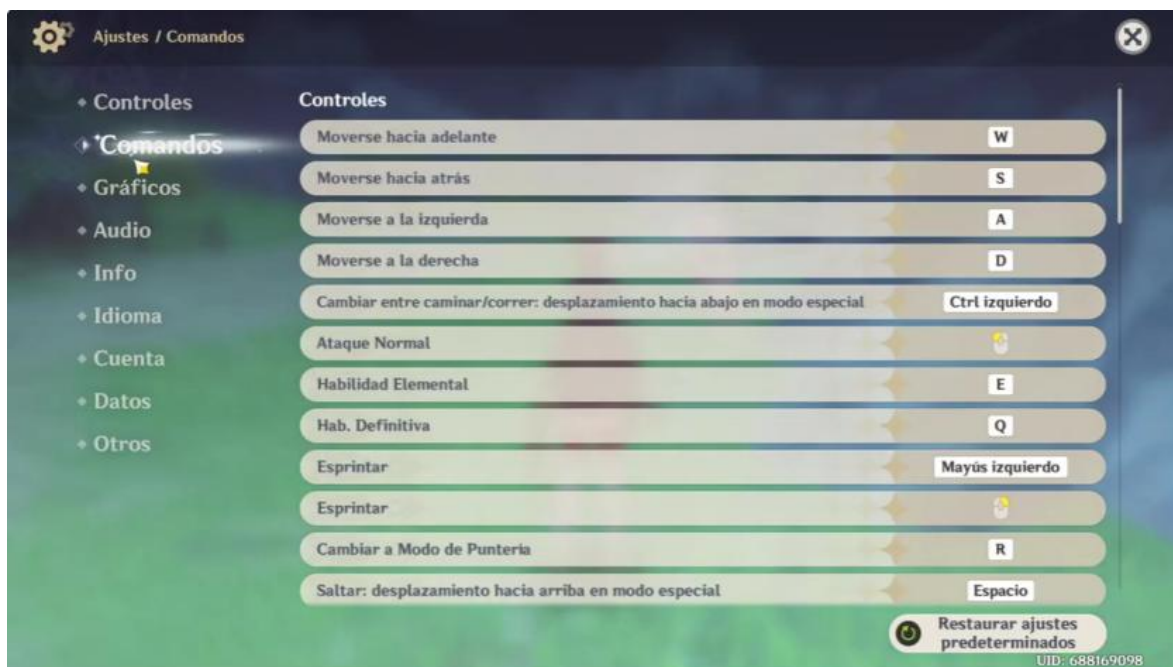
**Figura 4** *Escena con ambiente acción de batalla y uso de efectos SFX en partida del video juego Genshin Impact*



*Nota: Autoría propia*

Aunque no sea lo más primordial, sigue siendo parte del juego y por lo tanto darle sonoridad a la Interfaz del video juego da un toque de personalidad. Este tipo de audio es no diegético, es utilizado cuando existe una acción con alguna herramienta que tenga el juego, como ejemplo al abrir el menú, las configuraciones de juego u otras funciones o herramientas que este traiga. En este juego sus sonidos nos mantienen dentro del enfoque de aventura y anime, ya que mantiene su timbre tipo campana con algo de celeste lo que le da un toque o sensación de magia, esto permite al oyente estar aun en sintonía con el juego mientras se desplaza por los diferentes menús.

**Figura 5** Vista del interfaz de ajustes del video juego Genshin Impact



*Nota: Autoría propia.*

## Tarea 2: Eventos de diseño sonoro en audio middleware: Generación de un ambiente dinámico selvático

Un ambiente dinámico en términos de diseño sonoro para video juegos busca simular cambios en tiempo real, integrando de forma fluida y lo más acertado posible los diferentes efectos de sonido que hacen referencia al ambiente en donde transcurre el juego. Recordemos que el término ambiente es utilizado en este caso al lugar o sitio donde está ubicado nuestro jugador, que para este caso es la selva.

La herramienta que se utiliza es el Middleware Fmod Studio, que permite la programación en forma más sencilla de todos aquellos parámetros y dinamismos sonoros que tendrá nuestro ambiente selvático

**Figura 6** Imagen del programa Middleware Fmod Studio Versión 2.03.09



*Nota: Autoría propia*

Para la creación de este evento selvático, se debe tener en cuenta las características que la conforman, para lo cual se recurre a un análisis escuchando varias muestras de audio de selva encontrados en la red de internet e implementar una especie de esquema jerárquico que nos permite identificar los diferentes sonidos que allí lo confirman, de esta manera podemos mapear todos los sonidos y poder identificar las diferentes capas que lo conforman y esto permite saber que sonidos utilizar, en qué momento, con que efectos, que distancia, o si son audios que estarán sonando independientemente de la dinámica que se aplique, etc. Así mismo se plantea que tipo de herramientas se van a utilizar en Fmod studio para lograr nuestro objetivo realizando una tabla como guía base la cual puede ser modificada durante el diseño sonoro. Se puede decir que realizaremos algo parecido a una pre - producción de audio.

**Tabla 1** *Tabla de pre producción de audio dinámico para un ambiente selvático*

	<b>Eventos</b>	<b>Sonidos Act.</b>	<b>Tipo Instrumento</b>
<b>Base</b>	Selva Dia	Viento Suave	Simple
		Aves	Múltiple
<b>Base</b>	Selva noche	Viento suave	Simple
		Grillo	Múltiple
		Lechuza	simple
		Ranas	Múltiple
<b>Base</b>	Lluvia	Lluvia Suave	Simple
		Lluvia Fuerte	Simple
		Truenos	Múltiple
<b>Base</b>	Personaje	Caminar	Simple
		Correr	Simple

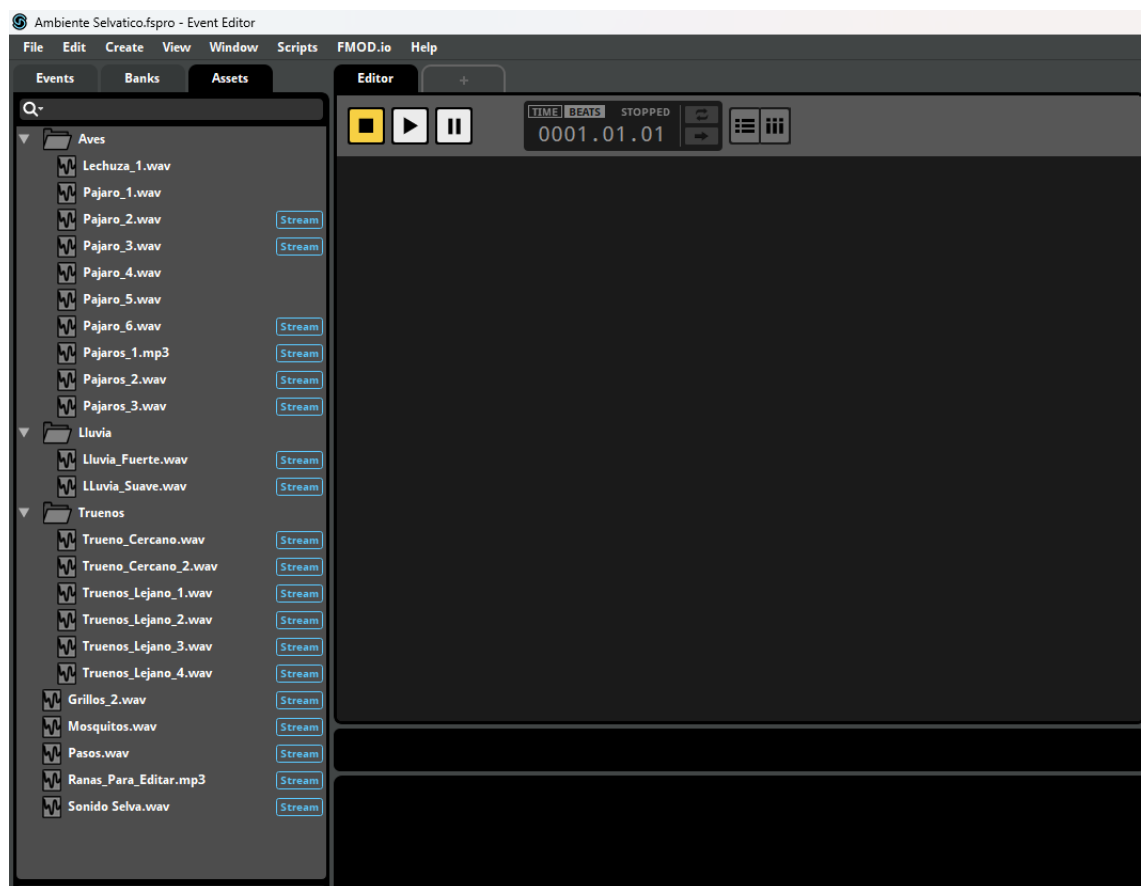


Eventos	Sonidos Act.	Tipo Instrumento
	Acercarse a mosquitos	Simple
	Quieto	Simpe

*Nota: Autoría propia*

Luego de tener más clara la idea de cómo va ser el diseño sonoro, se realiza la búsqueda o las grabaciones de los audios que serán la materia prima de nuestro proyecto las cuales podemos guardarlas y agruparlas según su relación en Fmod Studio en la pestaña Assets o activos.

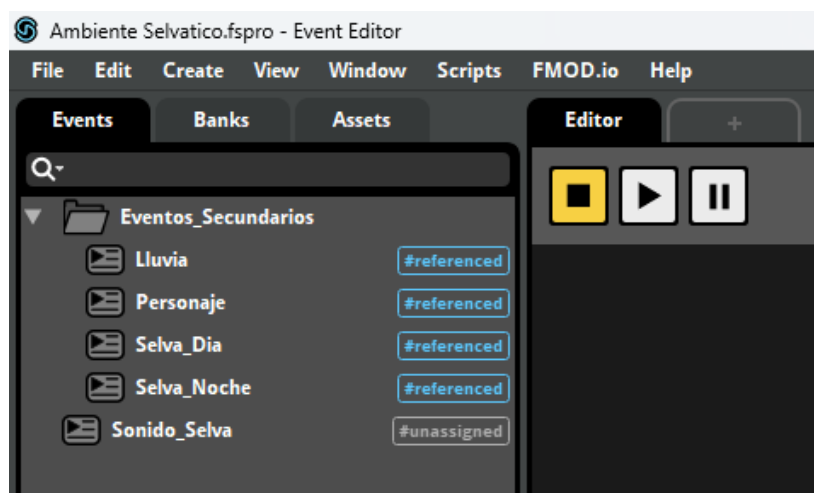
**Figura 7** Imagen de Fmod Studio en la pestaña Assets



*Nota: Autoría propia.*

Siguiendo nuestra tabla guía a la que se le llamó pre - producción, ver figura 6, generamos los llamados eventos base las cuales pueden ser 2D y 3D que contendrán los sonidos que son tomados de nuestro assets, y en ellas irán las mezclas, los efectos y dinamismos que debe tener cada audio.

**Figura 8** Imagen de Fmod Studio en la pestaña Events



*Nota: Autoría propia.*

Fmod Studio tiene diferentes herramientas para la manipulación de audio, aunque no es un DAW, se asemeja mucho a él, pero la edición de cada pista cumple una tarea muy diferente, puesto que su objetivo es la programación de eventos sonoros.

En este diseño sonoro dinámico de ambiente selva se utilizan las siguientes herramientas:

Single Instrument: Permite la inserción de un solo clip de audio.

Multi Instrument: Permite la inserción de múltiples clips de audio.

Scatterer Instrument: Permite la inserción de múltiples clips de audio generando dispersión aleatoria simulando espacio

**Zonas de Transición:** Permite saltar de una pista de audio a otra pista de audio de forma controlada a través de parámetros.

**Marcadores:** Permite el salto de una pista de audio a otra pista de audio en un punto en concreto de forma controlada a través de parámetros.

**Loop:** Permite la reproducción continua de una pista de audio.

**Event Instrument:** Permite agrupar dentro de una misma pista de audio otros eventos o audios. No es precisamente una pista de audio como tal sino es utilizado como llamado de otros eventos. Esto permite tener varios eventos dentro de un solo evento y ser activo a través de este.

**Hoja de parámetros:** Esta hoja permite crear controles parametrizados para la activación de las diferentes pistas de audio si estas cumplen los criterios establecidos programados.

**Automatización:** Permite realizar movimientos internos a través del tiempo como volúmenes de pistas de forma automática y efectos como reverb y ecualizaciones, mientras se reproduce el audio.

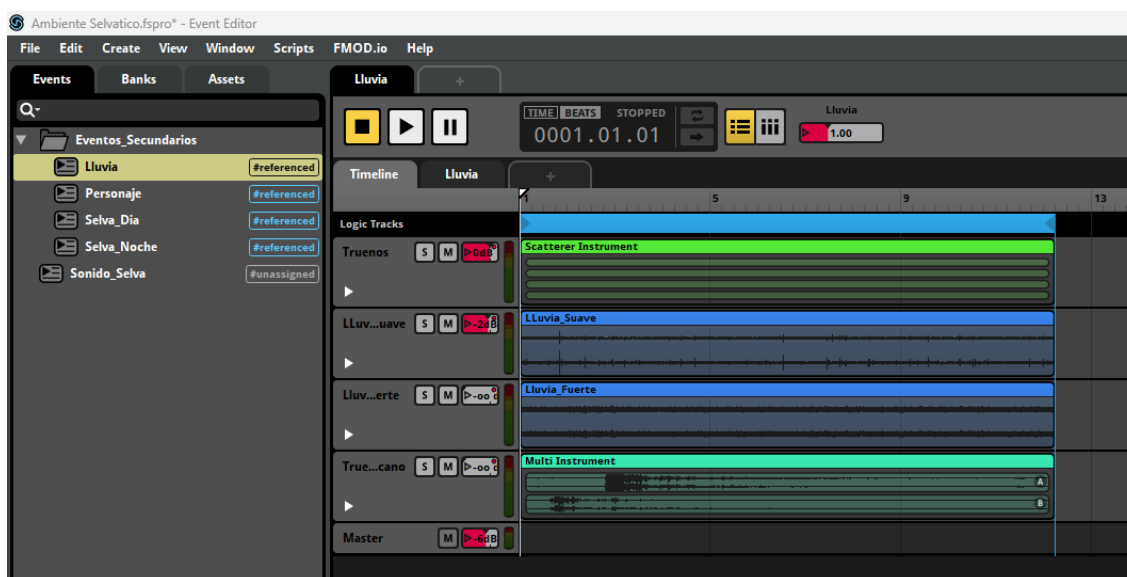
Ya teniendo claro cuáles son las herramientas de trabajo que vamos a utilizar procedemos a la creación de los eventos y a la edición y parametrización de cada uno de ellos logrando una simulación casi realista de un ambiente tipo selva.

**Figura 9** Imagen del evento personaje y sus diferentes pistas de audio con zonas de transición, marcadores, Single Instrument y loops junto con el parámetro Accion\_Personaje



*Nota: Autoría propia*

**Figura 10** Imagen del evento lluvia y sus diferentes pistas de audio con Single Instrument, Scatterer Instrument, Multi Instrument y loop, junto con el parámetro lluvia



*Nota: Autoría propia*

**Figura 11** Imagen del evento Selva\_Dia con Single Instrument, Scatterer Instrument, Multi Instrument, y Loop



*Nota: Autoría propia*

**Figura 12** Imagen del evento Selva\_Noche con Single Instruments y loop



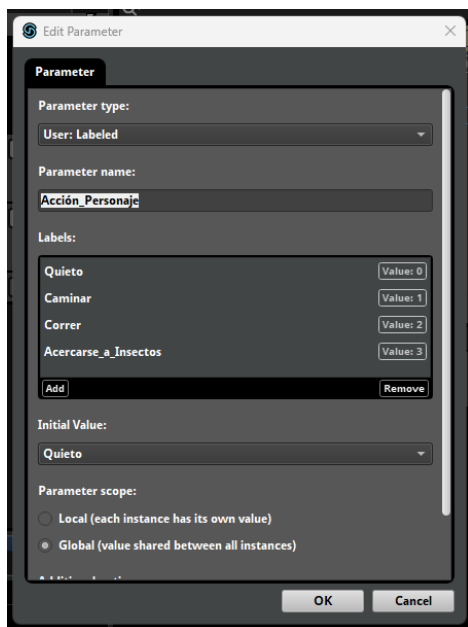
*Nota: Autoría propia*

**Figura 13** Imagen del evento principal *Sonido\_Selva*, compuesta por *Event Instrument*, *Single Instrument* y *loop* junto con los parámetros *hora*, *Accion\_Personaje* y *lluvia*



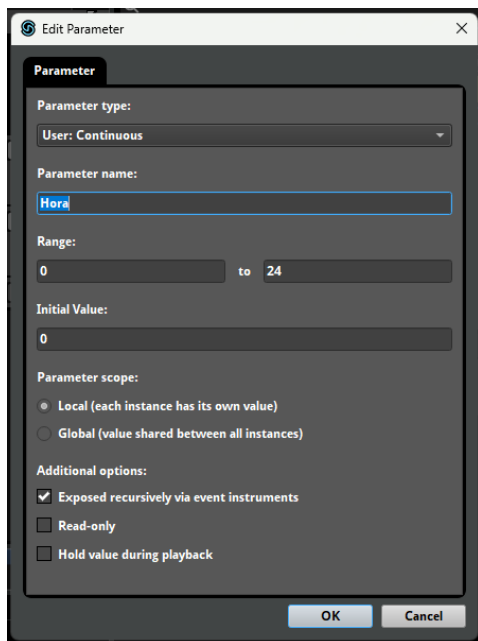
*Nota: Autoría propia*

**Figura 14** Imagen de la hoja de parámetros *Acción Personaje*



*Nota: Autoría propia*

**Figura 15** Imagen de la Hoja de parámetros Hora



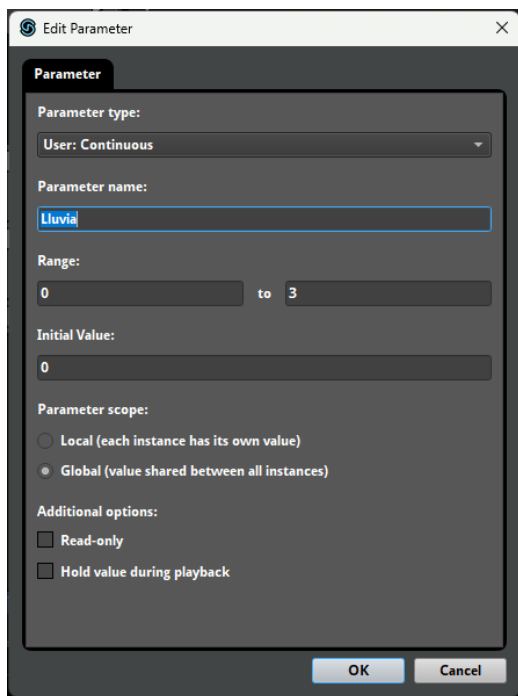
The screenshot shows a dialog box titled "Edit Parameter" with a close button (X) in the top right corner. The dialog is divided into several sections:

- Parameter type:** A dropdown menu showing "User: Continuous".
- Parameter name:** A text input field containing "Hora".
- Range:** Two input fields, the first containing "0" and the second containing "24", with the word "to" between them.
- Initial Value:** A text input field containing "0".
- Parameter scope:** Two radio buttons: "Local (each instance has its own value)" (selected) and "Global (value shared between all instances)".
- Additional options:** Three checkboxes: "Exposed recursively via event instruments" (checked), "Read-only" (unchecked), and "Hold value during playback" (unchecked).

At the bottom of the dialog are two buttons: "OK" and "Cancel".

*Nota: Autoría propia*

**Figura 16** Imagen de la hoja de parámetros para el evento lluvia



The screenshot shows a dialog box titled "Edit Parameter" with a close button (X) in the top right corner. The dialog is divided into several sections:

- Parameter type:** A dropdown menu showing "User: Continuous".
- Parameter name:** A text input field containing "Lluvia".
- Range:** Two input fields, the first containing "0" and the second containing "3", with the word "to" between them.
- Initial Value:** A text input field containing "0".
- Parameter scope:** Two radio buttons: "Local (each instance has its own value)" (unchecked) and "Global (value shared between all instances)" (selected).
- Additional options:** Three checkboxes: "Read-only" (unchecked) and "Hold value during playback" (unchecked).

At the bottom of the dialog are two buttons: "OK" and "Cancel".

*Nota: Autoría propia*

Luego de las respectivas pruebas y ajustes necesarios, se consigue el objetivo deseado logrando de forma casi realista la inmersión en un ambiente selvático con diferentes dinámicas dependiendo de la hora del día que se ajuste (formato de 24 horas), así como si existe lluvia suave, moderada o fuerte y con la sonoridad del personaje simulando si está quieto, camina, corre, o si se acerca a un enjambre de insectos.



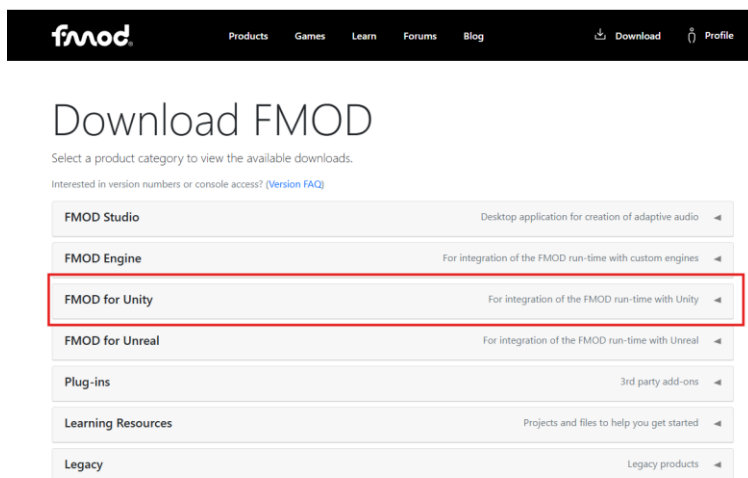
### Tarea 3 - Integración de audio middleware con motor de video juegos

Un motor de vídeo juegos es una plataforma donde en sí se programa y se realizan todas ediciones y configuraciones necesarias para la realización de cualquier video juego y aunque estos motores traen un editor de audio incorporado en ciertas ocasiones resulta complicado generar los diferentes tipos de sonidos y dinámicas necesarias, sea porque requiere de mucha programación o porque no tiene estándares de sonido ideales.

Actualmente contamos con plataforma como Unreal Engine, Godot, Game Marker, CryEngine, Unity entre otros para la creación de video juegos. Para nuestro caso utilizaremos el motor de video juego Unity versión 6000.0.58f2 en donde se podrá realizar la integración con nuestro Middleware Fmod Studio.

Antes de trabajar con Fmod Studio es necesario que nuestro Middleware se conecte con Unity, nuestro motor de videojuegos previamente instalado, para esto es necesario realizar la descarga directamente de la página web de Fmod y descargar el paquete integrador.

**Figura 17** Imagen de sitio web de Fmod Studio en su sección de descargas



Luego de realizar la integración es necesario desactivar el editor de audio de Unity y colocar como preferencia a Fmod Studio, esto habilitara internamente la conexión de estos dos programas y los recursos necesarios para su funcionamiento.

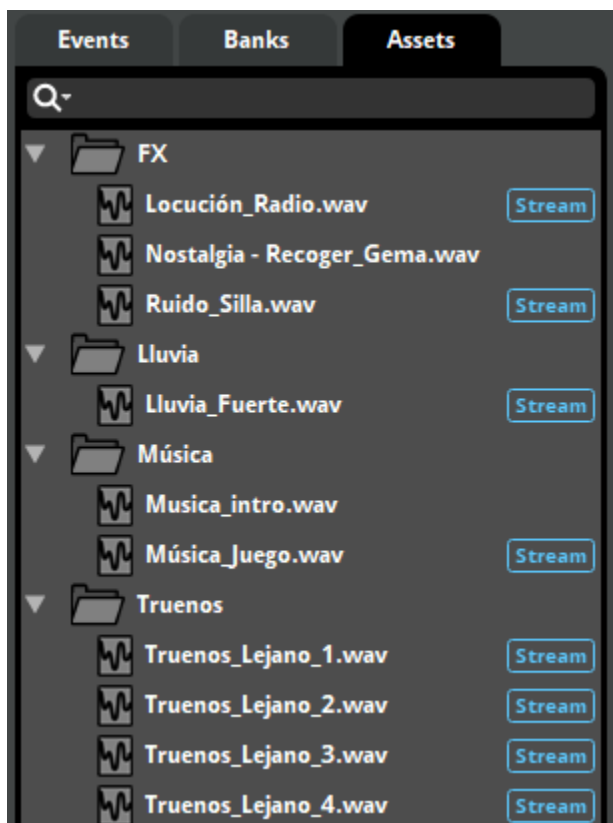
Ya todo listo, se procede a sonorizar nuestro video juego el cual fue seleccionado de las librerías de aprendizaje que trae Unity por defecto. Para nuestro caso se escoge un video juego del cual trata de una pequeña nave espacial encerrada en una sala de estar rodeada de muebles, sillas, un radio, una puerta animada que da al exterior de la habitación y cuyo objetivo del juego es poder recoger todas las gemas que se encuentra dentro de dicha habitación.

Como se menciona este video juego no tiene ningún tipo de audio por lo que se realiza el análisis respectivo dando como resultado cinco tipos de evento dividido en tres grupos los cuales tienen las siguientes características:

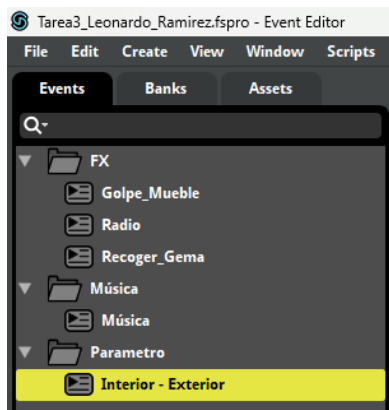
- Grupo1: Eventos de efectos el cual contiene los eventos de Golpe de mueble, Radio y Recoger Gema. Estos audios son diegéticos ya que pertenecen al mundo del juego.
- Grupo2: Evento de música no diegética el cual se genera un solo evento que contendrá la música del video juego. Esta música no pertenece al mundo del juego.
- Grupo3: Evento de parámetro, el cual contiene un audio de ambiente dinámico que modifica de forma automática la ecualización dependiendo si nuestra nave está fuera o dentro del recinto generando una sensación de oclusión. Este evento no afecta a la música no diegética.

Nuestra materia prima sonora es recolectada y cargada en el asset de Fmod estudio, teniendo las la librería de sonidos listas a disposición así mismo se crean los grupos y eventos mencionados anteriormente.

**Figura 18** Imagen de librería de sonidos en pestaña asset de Fmod Studio



**Figura 19** Imagen de los grupos y eventos generados en Fmod Studio



Se procede a realizar la inserción de cada uno de los clips de audio en los eventos creados, para lo cual se utilizaron las siguientes configuraciones:

- Evento música: Utilizamos dos regiones de Single Instrument en 2D, una para la introducción de la canción y la otra como la música en sí que sonará durante todo el juego, para lo cual utilizamos las regiones de loop

**Tabla 2** *Tabla de referencia para la configuración del evento Música*

<b>Evento</b>		<b>Música</b>		
<b>Tipo de Instrumento</b>	<b>Clip de Audio</b>	<b>Track Logic</b>	<b>Parámetros</b>	<b>Evento 2D/3D</b>
<b>Single Instrument</b>	Música de introducción	-	-	2D
<b>Single Instrument</b>	Música de desarrollo	Loop	-	2D

- Evento Golpe\_Mueble: Se utiliza una región tipo Single Instrument en 3D con el efecto de audio de golpe a mueble de madera sin la opción de loop

**Tabla 3** *Tabla de referencia para la configuración del evento Golpe\_Mueble*

<b>Evento</b>		<b>Golpe_Mueble</b>		
<b>Tipo de Instrumento</b>	<b>Clip de Audio</b>	<b>Track Logic</b>	<b>Parámetros</b>	<b>Evento 2D/3D</b>
<b>Single Instrument</b>	Golpe de mueble de madera	-	-	3D

- Evento Radio: El evento se crea de la forma tipo 3D, se utiliza una región de tipo Scatterer Instrument con la finalidad de generar diferentes distancias desde el punto de vista del jugador, como si en algún momento sonara con volumen alto o volumen bajo, así mismo tiene una región loop para dar continuidad al efecto sonoro de la locución.

**Tabla 4** *Tabla de referencia para la configuración del evento Radio*

<b>Evento</b>	<b>Radio</b>			
<b>Tipo de Instrumento</b>	<b>Clip de Audio</b>	<b>Track Logic</b>	<b>Parámetros</b>	<b>Evento 2D/3D</b>
<b>Scatterer Instrument</b>	Locución de radio	Loop	-	3D

- Evento Recoger\_Gema: El evento se crea con una región de audio tipo single Instrument en formato 2D, sin región de loop con una salida de audio a -6db debido a que el audio original tiene un volumen muy alto.

**Tabla 5** *Tabla de referencia para la configuración del evento Recoger\_Gema*

<b>Evento</b>	<b>Recoger_Gema</b>			
<b>Tipo de Instrumento</b>	<b>Clip de Audio</b>	<b>Track Logic</b>	<b>Parámetros</b>	<b>Evento 2D/3D</b>
<b>Single Instrument</b>	Recoger Gema	-	-	2D

- Evento Interior – Exterior: El evento se crea con dos Single Instruments, dos Multi Instruments, dos marcadores, dos zonas de traslado, loops, automatización en la ecualización y hoja de parámetro llamada Interior – Exterior

**Tabla 6** *Tabla de referencia para la configuración del evento Interior – Exterior*

<b>Evento</b>	<b>Interior - Exterior</b>			
<b>Tipo de Instrumento</b>	<b>Clip de Audio</b>	<b>Track Logic</b>	<b>Parámetros</b>	<b>Evento 2D/3D</b>
<b>Single Instrument</b>	Lluvia fuerte	<ul style="list-style-type: none"> <li>• Marker A</li> <li>• Loop</li> <li>• Transición a Marker B</li> </ul>	Interior - Exterior	2D
<b>Single Instrument</b>	Lluvia fuerte	<ul style="list-style-type: none"> <li>• Marker B</li> <li>• Loop</li> <li>• Transición a Marker A</li> </ul>	Interior - Exterior	2D
<b>Multi Instrument</b>	Truenos	<ul style="list-style-type: none"> <li>• Marker</li> <li>• ALoop</li> <li>• Transición a Marker B</li> </ul>	Interior - Exterior	2D
<b>Multi Instrument</b>	Truenos	<ul style="list-style-type: none"> <li>• Marker B</li> <li>• Loop</li> <li>• Transición a Marker A</li> </ul>	Interior - Exterior	2D

**Tabla 7** *Tabla de referencia para la hoja de parámetro Interior – Exterior*

<b>Hoja de parámetro</b>	<b>Interior - Exterior</b>
<b>Tipo Parámetro</b>	Label

<b>Nombre de Label</b>	<b>Valor</b>	<b>Herramienta a afectar</b>	<b>Valor Inicial</b>
<b>Interior</b>	0	Sin Afectación	X
<b>Exterior</b>	1	Subir volúmenes en ecualización en altos, medios y máster	-

Se realiza el respectivo test de audio en Fmod Studio verificando que sirven la dinámica esperada y que sean acordes al juego donde teniendo los resultados favorables nos adentramos en Unity para realizar las respectivas configuraciones para que el audio pueda ejecutarse como lo deseamos.

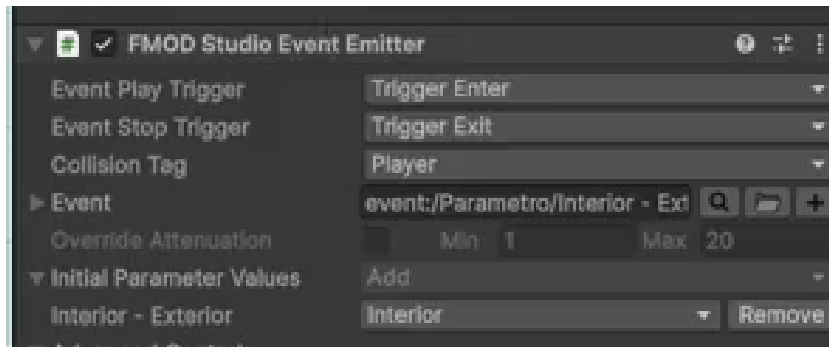
Primera mente crearemos dos cubos de tipo Trigger, uno que cubra toda la parte interior del recinto y la segunda estará fuera de ella. El objetivo con esto es crear dos zonas que permitan disparar a través de nuestro parámetro Interior – Exterior la forma como escucharemos el audio del ambiente del juego. Recordamos desactivar la opción de renderizado para que no veamos el cubo en sí durante el juego.

Cuando el jugador esté dentro del cubo interior sonará el ambiente con oclusión y cuando salga de le el audio se escuchará con mejor nitidez como si se estuviera fuera de él.

Para que Unity reproduzca nuestro audio es necesario adjuntarle una propiedad de audio la cual es llama FMOD Studio Event Emitter y seleccionamos nuestro evento Interior – Exterior lo que permite a Unity traer la información que hemos realizado en Fmod Studio incluyendo las opciones o valores de los parámetros Interior y Exterior. Es necesario que en esta opción se seleccione como valor inicial Interior que tiene la configuración de audio con la ecualización necesaria para sentirnos en un ambiente de oclusión o de sentir el audio como si estuviéramos

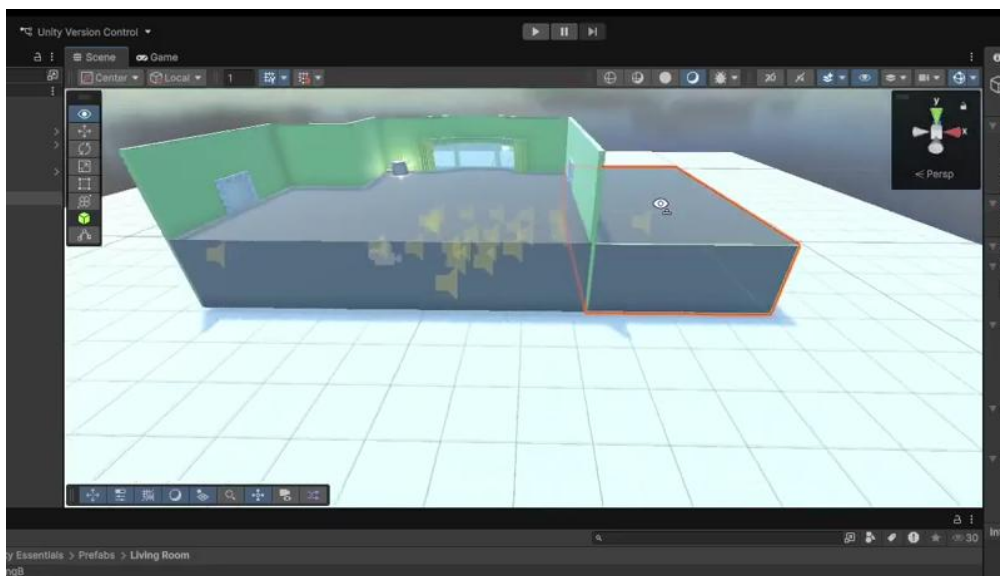
dentro del recinto. El audio se activará o reproducirá apenas Unity crea el objeto jugador durante el juego, es decir nuestra nave espacial, que también ha sido configurado desde esta propiedad.

**Figura 20** Imagen de la propiedad *Fmod Studio Event Emitter* en Unity y su configuración para el cubo Interior



A si mismo realizaremos la programación para nuestro segundo cubo que será nuestra zona exterior del recinto.

**Figura 21** Imagen de la ubicación de los cubos Interior y Exterior en el juego.



Seguimos con la programación para el efecto 3D de nuestro radio, el cual se le asigna el evento Radio que está configurado en 3D desde Fmod Studio y en Unity adicionaremos a este objeto un Fmod Studio Event Emitter, buscamos nuestro evento de Radio y automáticamente nos



trae los valores de este. Al ser un evento 3D Unity nos activa para el objeto el parámetro de distancia el cual configuraremos directamente desde los radios o circunferencias que salen del objeto permitiéndonos visualmente decidir a qué distancia queremos que la radio sea escuchada por el jugador.

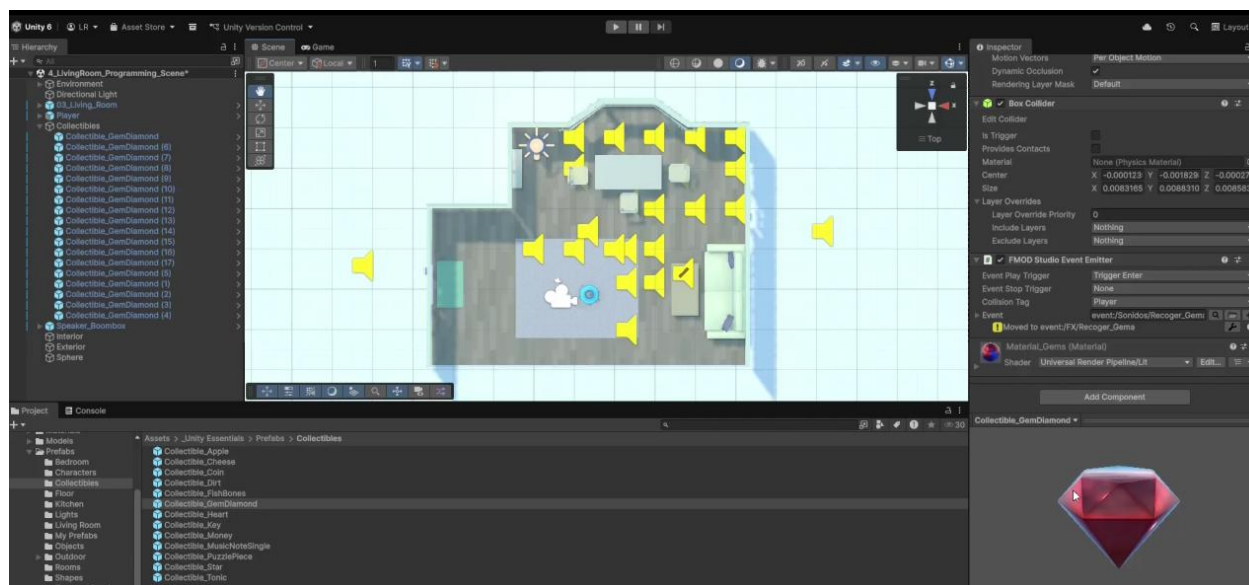
**Figura 22** Imagen del objeto Radio y su configuración para la escucha a la distancia en 3D



Pasamos a la configuración del audio cuando nuestra nave espacial recoge las gemas, así como en los dos casos anteriores realizamos la misma operación añadiendo un Fmod Studio Event Emitter seleccionado nuestro evento recoger\_Gema que está configurado como evento 2D al objeto gema. Configuramos el objeto como Trigger y con la activación cuando el jugador colisione con él. A diferencia de los cubos y la radio, en nuestro juego tenemos muchas gemas por lo que sería engorroso añadir audio a cada gema por individual, por lo que se opta la técnica de aplicar desde el objeto madre esta configuración de audio y de Trigger la cual será replicada a cada gema sin necesidad de hacerlo una por una.

Con esta técnica aplicada no solo lograremos agilidad en la configuración de audio para cada objeto gema, sino que permite rendimiento en la PC, reduciendo consumo de memoria y lectura de datos por individual.

**Figura 23** Imagen de la configuración gema y vista del total de gemas utilizadas en el juego



Exactamente igual se realiza la configuración para los objetos sillas que tendrán como evento seleccionado Golpe – Mueble, el cual sonará cuando nuestra nave choque con cualquiera de las tres sillas ubicadas dentro del juego.

Para la música del juego, recordando que no es diagético, utilizamos en este caso la creación de una esfera sin renderizar ubicada fuera del recinto como emisor de este evento, al ser de tipo 2D siempre lo estaremos escuchando sin ningún tipo de distancia y con el mismo volumen. Cuando Unity crea los objetos y al crear esta esfera automáticamente reproducirá el evento música. Este evento será modificado más adelante en la tarea 4 optimizando rendimiento en nuestra PC

Terminada las configuraciones en Unity se procede a la reproducción del video juego realizando un testeo del mismo donde se cuadran volúmenes y las distancias sonoras de los objetos 3D como radio y sillas para mejorar la experiencia del jugador

**Figura 24** *Imagen del juego en acción*



#### **Tarea 4 - Triggers, Snapshots y modificación continua de parámetros de audio**

Anteriormente hemos visto cómo aplicar un diseño sonoro a un video juego utilizando las herramientas más esenciales de Fmod Studio como Single Instruments, Scatterer Instrument, Event Instrument entre otros en el motor de video juego Unity, aunque el proyecto funciona correctamente es probable que nos encontremos con proyectos más complejos, en donde es necesario crear dinamismo en la música, por ejemplo, sitios que tendrán reverberación, o la creación de zonas dentro del juego que activen alarmas entre otras opciones.

En el siguiente apartado vamos aplicar otras características de Fmod Studio para mejorar o aplicar nuevas opciones dentro del video juego que ya se trabajó en la tarea anterior.

Los cambios que vamos a realizar son los siguientes:

- Crear una zona en específico dentro de la sala de estar para que se escuche con reverberación todos los efectos menos la música no diegética.
- Crear una zona donde dentro de la sala de estar para que se active una alarma.
- Crear dinamismo en la música no diegética donde el tema musical presente cambios de tensión cuando avance en ciertas zonas del recinto de estar.

Para lograr nuestros objetivos anteriormente descritos, vamos a utilizar los siguientes elementos que nos proporciona Fmod Studio:

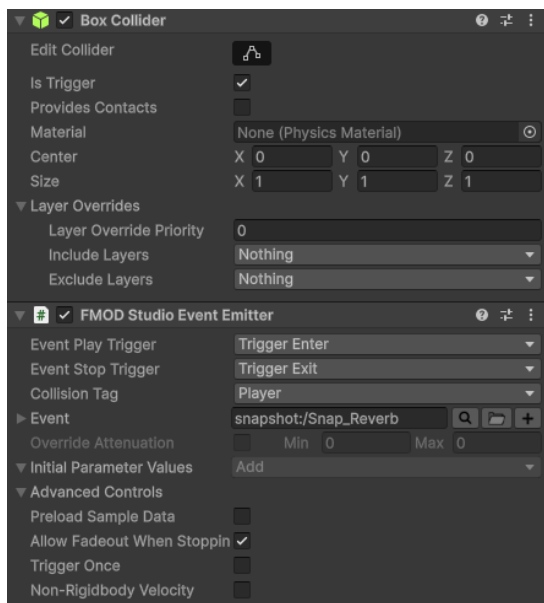
- Triggers: Actúa como disparadores mediante las condiciones lógicas de un parámetro asignado. Esto permite activar o desactivar clips de audio, realizar saltos entre clips a través de las zonas de transición, realizar cambios en la ecualización entre otros.
- Snapshots: Se puede decir en términos simples que es la automatización de las mezclas de audio en el mixer que tiene Fmod Studio. A partir de ciertos parámetros podemos

programar como queremos que se comporte la mezcla para determinados momentos en el juego.

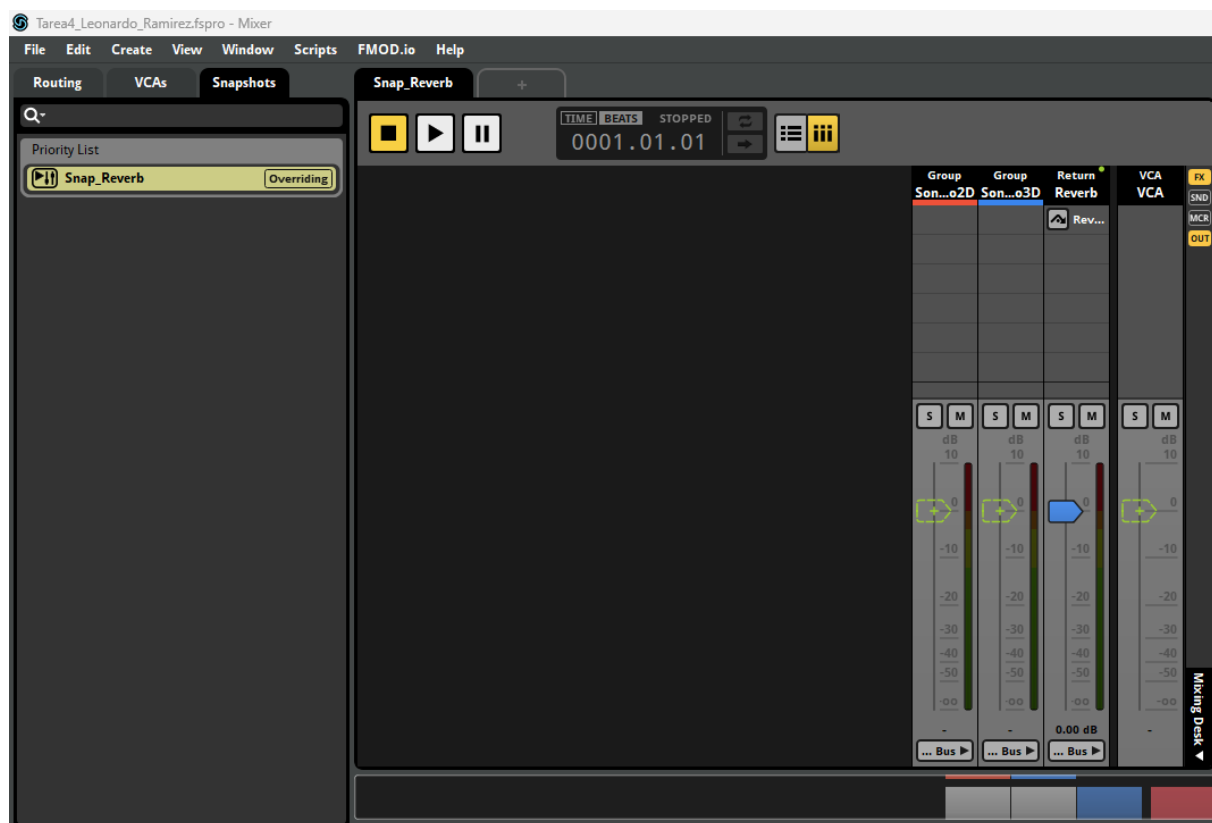
Es momento de crear las diferentes zonas dentro del recinto para incluir la zona de reverberación y la zona de alarma, así como también las tres zonas que van a generar un audio dinámico en la música del video juego.

Nuestra zona de reverberación estará ubicada cerca de la lámpara y cubrirá 1/3 parte del área del recinto, para esto creamos un cubo sin renderizar y será activada la opción *is Trigger* en el componente *Box Collider*, se utiliza un complemento para este cubo de *Fmod Studio Event Emitter* lo que permite la emisión de audio. Dentro de sus opciones indicaremos a Unity que el evento a llamar será el *Snapshot Snap\_Reverb* el cual fue configurado en *Fmod Studio*. El evento se activará cuando el jugador ingrese al cubo y se desactivará cuando salga de él.

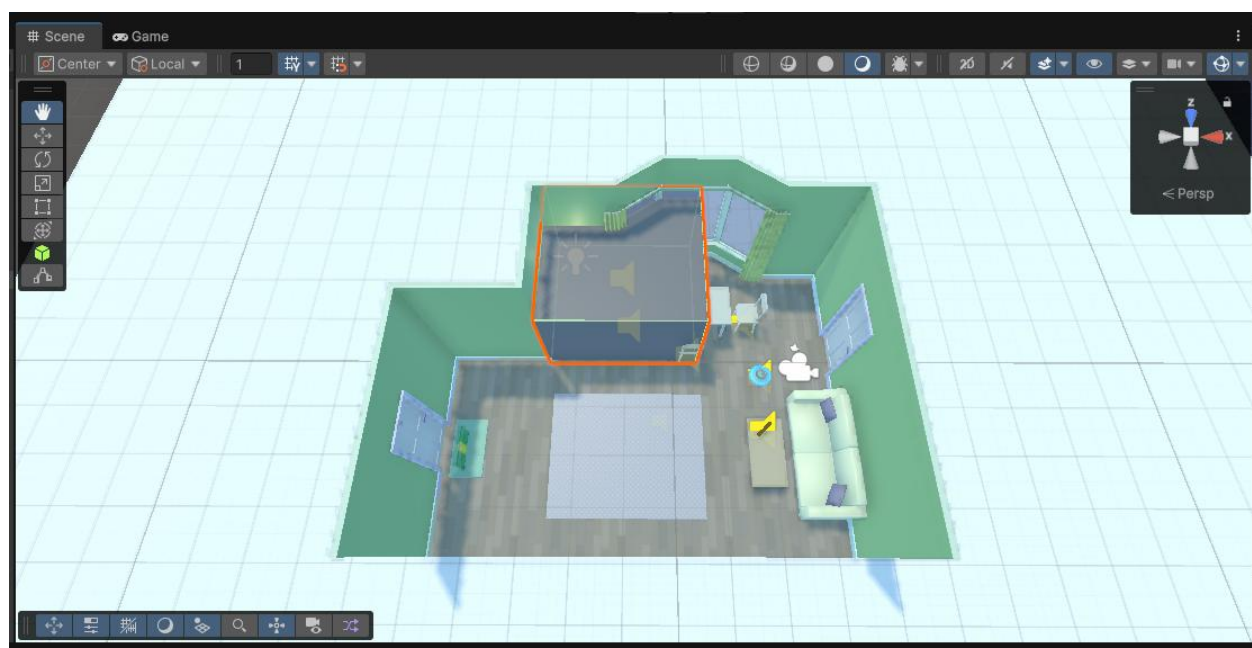
**Figura 25** Imagen de la configuración en Unity del área de Reverberación



**Figura 26** Imagen de la programación del Snapshot en Fmod Studio



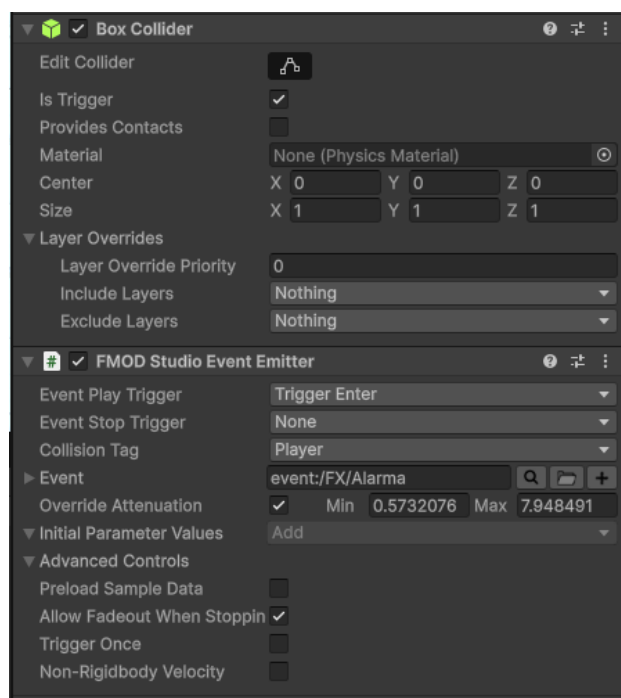
**Figura 27** Imagen de la ubicación del cubo de reverberación en el recinto.



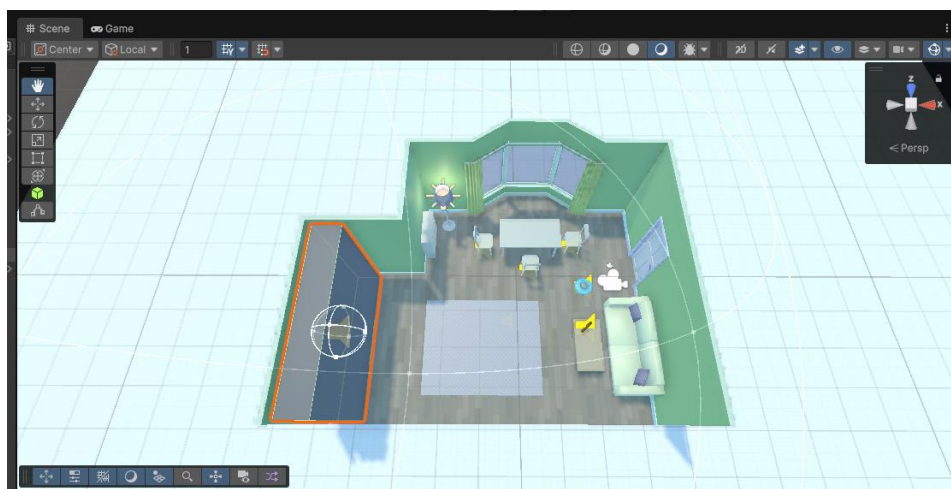
Ahora realizaremos la ubicación de la zona de alarma. Esta zona se activará apenas el jugador ingrese al área establecida. A diferencia de la zona de reverberación que su efecto termina a salir de la zona, esta no parará de sonar así el jugador se retire de ella, pero tendrá la particularidad que al ser un evento 3D, escucharemos como nos alejamos o nos acercamos al sonido dependiendo de la distancia a la que nos encontremos.

Igual que en el caso anterior creamos un cubo sin renderizar, con la opción Is Trigger activada en el componente Box Collider y añadimos el componente Fmod Studio Event Emitter, el cual activará el evento Alarma cuando el jugador ingrese a la zona, pero no podrá ser desactivado a partir del momento de su reproducción. Para lograr esto simplemente seleccionamos en la parte que dice Event Play Trigger como Trigger Enter y en la opción Event Stop Trigger colocamos none, esto configura la reproducción, pero no el detener de la pista de audio.

**Figura 28** Imagen de la configuración en Unity de la zona de alarma



**Figura 29** Imagen de la ubicación de cubo zona alarma en el recinto



Para la creación de la música dinámica dentro del juego, fue necesario cambiar el estilo de música que se tenía en la tarea 3, lo que nos permite sentir los cambios de tensión mientras el jugador pasa por las diferentes zonas que activarán el dinamismo de la música. También se requiero ajustar los cambios en Fmod Studio con las pistas de audio y realizar los ajustes en la hoja de parámetros para que pudiéramos tener el efecto deseado.

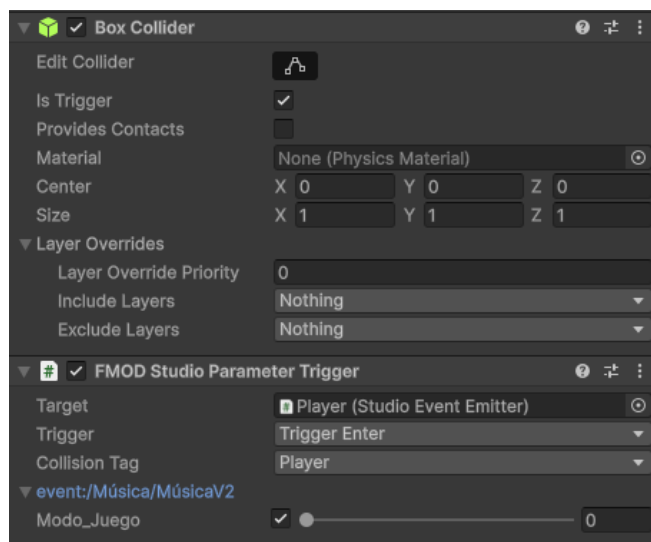
**Figura 30** Imagen de configuración en Fmod Studio de la nueva pista de música utilizada en el juego



Como podemos ver en la figura 30, se realizaron ajustes en la música, teniendo tres pistas de audio que tienen diferentes tipos de tensión, así como la inclusión del parámetro Juego con tres valores diferentes. Las pistas de audio tienen configurado una transición programada de un intervalo de cuantización de 1 Bars, esto permite un cambio suave sin que se oiga cortes o interrupciones o cortes bruscos entre cada pista. También para lograr este cambio suave y llamativo utilizamos el contador en Beats y no en Time, lo que facilita visualmente donde se genera los cambios y lo más importante es colocar el tempo en que está la canción para que fluya sin novedad. En este caso lo tenemos en 95 bpm.

Para Unity hemos utilizado 3 cubos que tienen una configuración diferente, ya que estas están con el módulo Fmod Studio Parameter Trigger, lo que significa que activará el evento dependiendo del parámetro que nosotros le asignemos a ese cubo. Es importante recordar que este parámetro es el que tenemos ya configurado en Fmod Estudio por lo en Unity no podremos seleccionar otra opción demás.

**Figura 31** Imagen de la configuración en Unity de las de cambio de tensión en el evento MusicaV2



Como se puede observar en la figura 31, tenemos tres áreas que serán las encargadas de activar los tres diferentes tipos de tensión de música que están programadas en Fmod Studio.

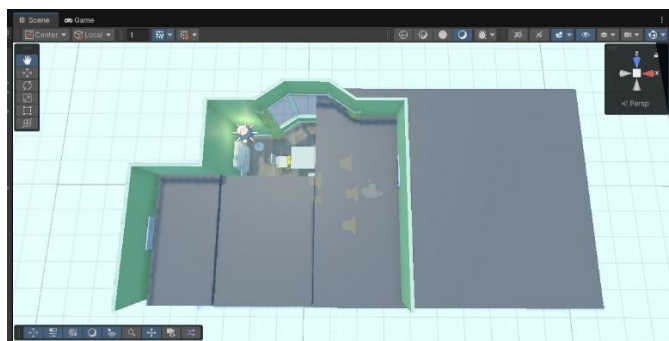
La primera área, la más grande, tenemos la música de desarrollo, o la que se escuchará en casi todo el video juego, es la música principal por lo que tiene la mayor envergadura de área cubierta.

La segunda área será la que tenemos en medio de la sala de estar, el de segundo tamaño, el que encontramos en la mitad de dos cubos. Esta área activará la pista de audio con tensión 1, o de primer aviso que nos indica que estamos cerca algo que nos puede interesar.

La tercera área que es la más pequeña, la de costa izquierdo, activara el audio de tensión 2, esta música ya tiene un acompañamiento adicional que nos da a entender que estamos en el punto de acción. Es en esta área es donde encontramos la zona de alarma, por lo que nos genera una sensación de violación de espacio privado.

Para el espacio que no tiene cubo, la música que escucharemos será dependiente del cubo o zona de donde provenga el jugador. Es decir que, si se proviene del cubo mayor, escucharemos en esta zona la música de desarrollo, pero si viene del cubo o área que está justo en la mitad entonces escucharemos la tensión de la música Tensión 1.

**Figura 32** Imagen de las zonas de tensión para el evento MusicaV2

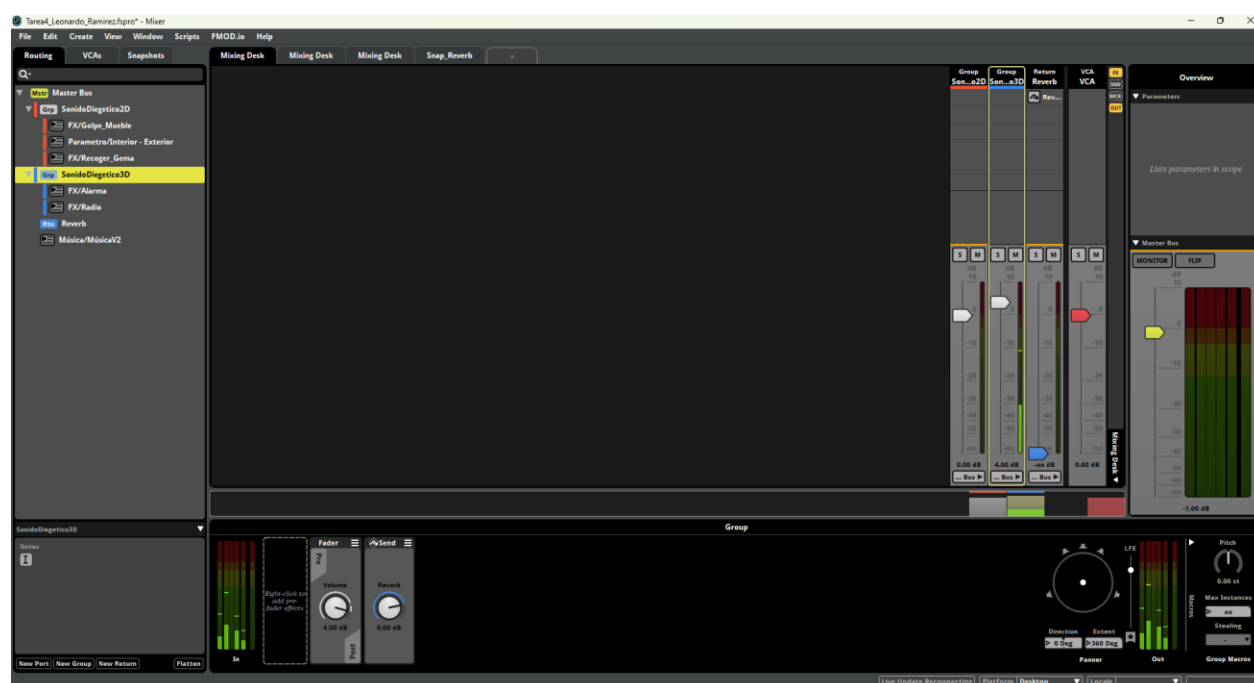


## Tarea 5 - Mezcla y finalización de un proyecto de sonorización de video juego

En este punto del trabajo vamos a revisar los temas de niveles de audio y ruteo de señal, algo muy parecido a lo que hacemos en un DAW.

En el caso de Fmod Studio, este Middleware trae su propio mezclador y forma de realizar los ruteos de señal, en el podemos crear grupos y envíos de señales de audio a otros canales, como por ejemplo a canales de efectos o en nuestro caso al canal de reverberación.

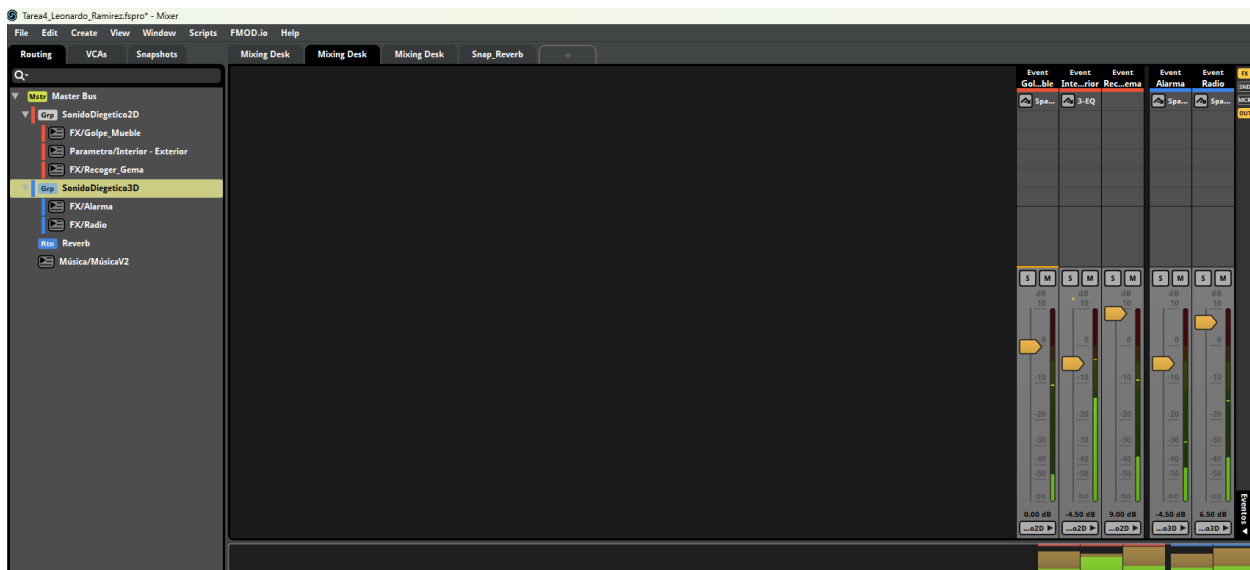
**Figura 33** Imagen del mixer en Fmod Studio



En principio generamos dos grupos en donde se enrutarán las señales de los diferentes eventos. Para el grupo llamado Sonido Diegético2D involucraremos los eventos 2D como los son Golpe\_Mueble, Parámetro Interior – Exterior y el evento recoger Gema. Para el otro grupo nombrado Sonido Diegético3D3 se asignaron los eventos Alarma y Radio. Las señales de audio de cada uno de estos eventos serán enrutadas al grupo asignado, así mismo estos grupos les hemos configurado un envío de señal al canal de reverberación, el cual tiene configurado una

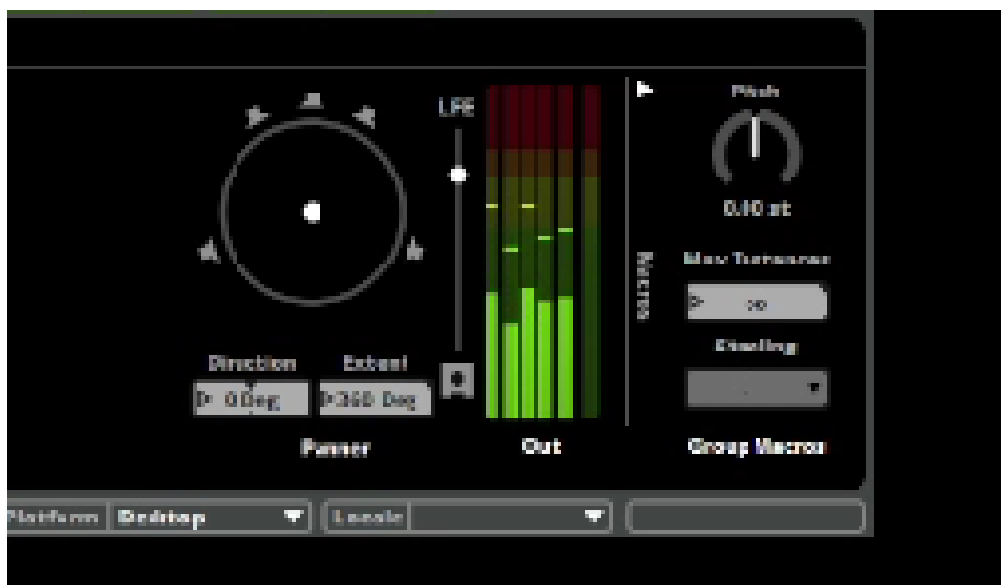
reverberación que afectara las señales que a él ingresen y posteriormente lo envía al master. Por otro lado, los grupos estarán enviando otra señal al master a parte del envío al canal de reverberación, esto con el fin de tener una señal que es de solo para efectos y otra limpia.

**Figura 34** Imagen del Mixer en Fmod Estudio y mezcla de los niveles de audio de los eventos



Por otro lado, vemos también la configuración de salida de audio en Surround 5.1

**Figura 35** Imagen Surround 5.1 en Fmod Studio.



Para la mezcla de los niveles de audio de los eventos se realizaron diferentes configuraciones hasta encontrar el punto más equilibrado con el objetivo de que cada sonido se escuchara de forma clara sin generar enmascaramiento entre ellos.

**Figura 36** Imagen de mezcla de los eventos asignados a los grupos.



Ya realizado la mezcla y el enrutamiento de las señales de audio, procedemos a realiza la prueba en vivo entre Fmod Studio y Unity jugando en tiempo realiza y mirando los niveles de audio y realizando ajustes necesarios. Para esto utilizamos la opción del Live Update que proporciona Fmod para realizar este primer testeo.

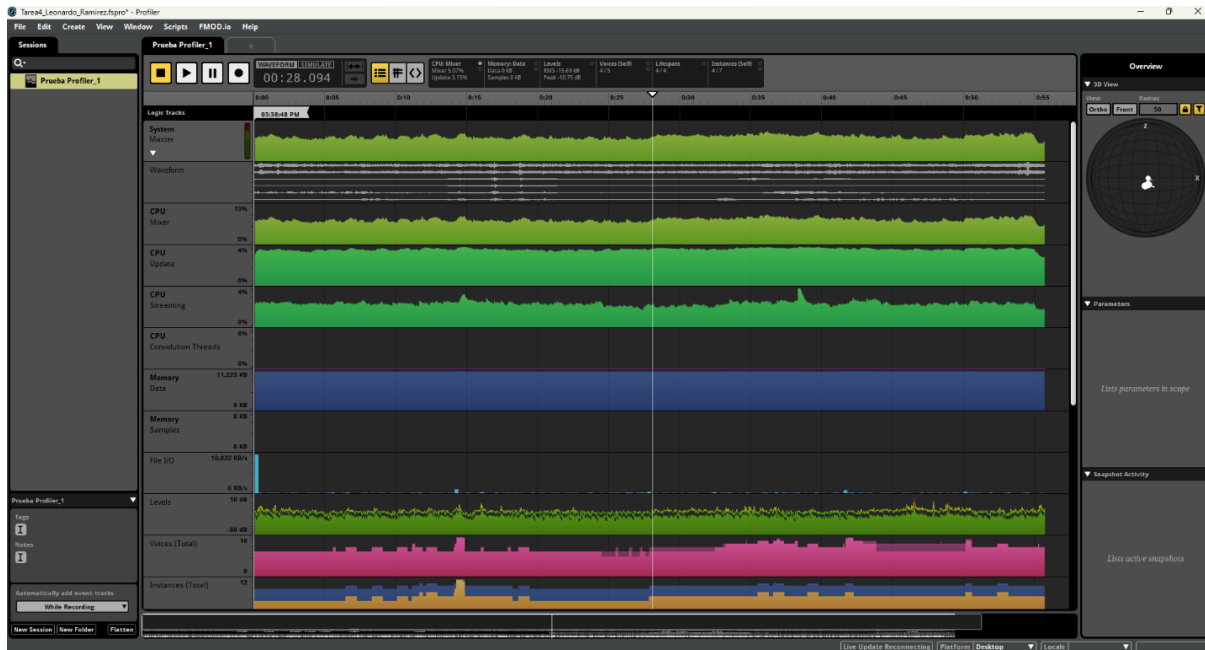
**Figura 37** Imagen del primero testeo de mezcla jugando en vivo en Unity



Ya verificado el testeo inicial, seguimos con la prueba de rendimiento general que nos proporciona Fmod Studio con la ventana Profiler. Con esta herramienta podemos analizar rendimientos en CPU, memoria, niveles de audio reportado durante la ejecución del juego en vivo, entre otros datos el cual nos ayudara a realizar las mejoras necesarias para el video juego.

Todas estas herramientas proporcionan una ayuda adicional con el que podemos mejorar nuestro diseño sonoro para video juegos para hacerlo mas efectivo y con buen rendimiento.

Figura 38 Imagen de la herramienta Profiler de Fmod Studio



Se adjunta link del video sobre el trabajo final diseño sonoro para video juegos

<https://youtu.be/8kIV8ObT8rE>

Se adjunta link del One Drive

[Tarea 5 Leonardo Ramirez](#)





## Conclusiones

Fmod Studio como middleware para la creación y programación de diseño sonoro para video juegos, nos proporciona una gran variedad de herramientas que facilita la programación de eventos que serán aplicados a los diferentes motores de video juego, en este caso y como se ha visto, aplicado a Unity, motor de video juegos de tamaño mundial.

Durante su uso se pudo evidenciar que la curva de aprendizaje no es tan larga, y que se requiere es de mucha práctica diaria para descubrir otra infinidad de posibilidades y configuraciones posibles que se puede llegar adquirir, ampliando aun mas el potencial de Fmod Studio.

Durante lo aprendido se logró entender, manipular e interiorizar temas como los diferentes instrumentos que nos ofrece este programa, como los Single Instruments, Multi-instruments, Event Instruments en otros, así como lograr la automatización de niveles de audio, la inserción de efectos y la programación a través de parámetros para un audio dinámico profesional. Por otra parte la ventana de mezcla que nos ofrece Fmod Studio trae la esencia parecida a un DAW, lo que hace amigable para quienes ya trabajan con este tipo de programas y lo pueden encontrar familiar, así mismo como funciones muy parecidas cuando se trabaja en la edición y mezcla en un programa de estos, como encontrar la opción de hacer grupos y envíos para canales de efectos, poder tener control de los niveles de audio de varios grupos con la opción VCA, mutear o dejar en solo los canales que uno desea... realmente es cómodo su interfaz.

## Referencias

- ¿Qué son las dinámicas de video juegos?, Tomado de <https://www.tokioschool.com/noticias/que-son-dinamicas-videojuegos/>
- Martinez, I. (2021). *Música y sonido para videojuegos*. Press Music. <https://press-music.com/musica-y-sonido-para-videojuegos/>
- Karen Collins. (2008). *Game Sound : An Introduction to the History, Theory, and Practice of Video Game Music and Sound Design*. The MIT Press. [https://bibliotecavirtual.unad.edu.co/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=e000xww&AN=237756&lang=es&site=eds-live&scope=site&ebv=EB&ppid=pp\\_123](https://bibliotecavirtual.unad.edu.co/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=e000xww&AN=237756&lang=es&site=eds-live&scope=site&ebv=EB&ppid=pp_123)
- Rehren, C., & Cárdenas, J. (2011). *Motores de Audio para Video Juegos*. Síntesis Tecnológica, 4(2), 81-99. <http://revistas.uach.cl/html/sintec/v4n2/body/art09.html>
- Horowitz, S., & Looney, S. (2014). *A Guide to Game Audio Middleware*. Electronic Musician, 30(7), 64–72. <https://bibliotecavirtual.unad.edu.co/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=edsgao&AN=edsgcl.379089739&lang=es&site=eds-live&scope=site>
- Lanham, M. (2017). *Game Audio Development with Unity 5.X*. Packt Publishing, 191-193. [https://bibliotecavirtual.unad.edu.co/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=e000xww&AN=1547031&lang=es&site=eds-live&scope=site&ebv=EB&ppid=pp\\_191](https://bibliotecavirtual.unad.edu.co/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=e000xww&AN=1547031&lang=es&site=eds-live&scope=site&ebv=EB&ppid=pp_191)

- Moreno Viasus, R. (10,03,2017). [\*FMOD Studio Introducción\*](http://hdl.handle.net/10596/11795). [Archivo de video].  
<http://hdl.handle.net/10596/11795>
- Moreno Viasus, R. (10,03,2017). [\*FMOD Studio Single Instrument\*](http://hdl.handle.net/10596/11789). [Archivo de video].  
<http://hdl.handle.net/10596/11789>
- Moreno Viasus, R. (10,03,2017). [\*FMOD Studio Multi Instrument\*](http://hdl.handle.net/10596/11788). [Archivo de video].  
<http://hdl.handle.net/10596/11788>
- Moreno Viasus, R. (10,03,2017). [\*FMOD Studio Event Instrument\*](http://hdl.handle.net/10596/11794). [Archivo de video].  
<http://hdl.handle.net/10596/11794>
- Moreno Viasus, R. (10,03,2017). [\*FMOD Studio Scatterer Instrument\*](http://hdl.handle.net/10596/11793). [Archivo de video].  
<http://hdl.handle.net/10596/11793>
- Moreno Viasus, R. (10,03,2017). [\*FMOD Studio Programmer, command, snapshot and Plug-in Instruments\*](http://hdl.handle.net/10596/11777). [Archivo de video]. <http://hdl.handle.net/10596/11777>
- Lanham, M. (2017). [\*Game Audio Development with Unity 5.X\*](#). Packt Publishing, 222-257. [https://bibliotecavirtual.unad.edu.co/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=e000xww&AN=1547031&lang=es&site=eds-live&scope=site&ebv=EB&ppid=pp\\_222](https://bibliotecavirtual.unad.edu.co/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=e000xww&AN=1547031&lang=es&site=eds-live&scope=site&ebv=EB&ppid=pp_222)
- Moreno, R. (2021). [\*Tutorial de diseño de eventos sonoros dinámicos en FMOD Studio\*](https://ramorenov.github.io/gameaudio/FMOD/01_Fmod_evento_din%C3%A1mico.html). Game Audio.  
[https://ramorenov.github.io/gameaudio/FMOD/01\\_Fmod\\_evento\\_din%C3%A1mico.html](https://ramorenov.github.io/gameaudio/FMOD/01_Fmod_evento_din%C3%A1mico.html)

FMOD (2021). [\*Studio User Manual 2.01\*](#). [Manual de usuario]

<https://www.fmod.com/resources/documentation-studio?version=2.1&page=parameters.html#parameter-trigger-conditions>

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](https://ramorenov.github.io/gameaudio/04_Fmod_triggers_y_parametros.html)

Moreno, R. (2021). [\*Tutorial de Integración de Fmod con una escena de Unity\*](#). Game Audio.

[https://ramorenov.github.io/gameaudio/01\\_IntegracionfmodUnity.html](https://ramorenov.github.io/gameaudio/01_IntegracionfmodUnity.html)

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

[https://ramorenov.github.io/gameaudio/FMOD/02\\_Fmod\\_musica\\_adaptativa.html](https://ramorenov.github.io/gameaudio/FMOD/02_Fmod_musica_adaptativa.html)

Moreno, R. (2021). [\*Tutorial música en Fmod para una escena de Unity\*](#). Game Audio.

[https://ramorenov.github.io/gameaudio/06\\_Fmod\\_musica.html](https://ramorenov.github.io/gameaudio/06_Fmod_musica.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](https://ramorenov.github.io/gameaudio/07_Fmod_postproducci%C3%B3n_y_testeo.html)