

Proceso de diseño sonoro dinámico para la escena del videojuego The Viking Village

Luis Felipe Padilla Restrepo

Asesor

Wilson Cárdenas Cristancho

Universidad Nacional Abierta y a Distancia UNAD

Escuela de Ciencias Básicas, Tecnología e Ingeniería ECBTI

Diplomado de Profundización Audio Dinámico para Videojuegos

2024

Tabla de contenido

Lista de Anexos.....	5
Introducción	6
Objetivos.....	7
Objetivo General	7
Objetivos Específicos	7
Desarrollo de la explicación del proceso de diseño sonoro dinámico de la escena del videojuego	
The Viking Village	8
Tarea 1: Análisis de un Videojuego	8
Tarea 2: Generación de un Ambiente Sonoro Dinámico.....	10
Tarea 3: Sonorización de un Proyecto de Videojuego.....	12
Tarea 4: Programación del Diseño Sonoro del Videojuego	21
Tarea 5: Mezcla, Pruebas y Optimización de Audio del Videojuego.....	29
Anexos	36
Conclusiones.....	37
Bibliografía.....	38

Tabla de Figuras

Figura 1 <i>Interfaz de inicio de Minecraft</i>	8
Figura 2 <i>Momento adaptativo e interactivo en Minecraft</i>	10
Figura 3 <i>Interfaz de Fmod Studio con vista de proyecto</i>	11
Figura 4 <i>Vista de la interfaz de Fmod con el parámetro de Intensidad</i>	12
Figura 5 <i>Imagen con versiones de Unity Hub y Fmod Studio</i>	13
Figura 6 <i>Guía de acceso a la assets store de Unity</i>	14
Figura 7 <i>Imagen del controlador de juego</i>	15
Figura 8 <i>Imagen del personaje Erika Archer provisto</i>	16
Figura 9 <i>Opciones de animación, tomado de www.maximo.com</i>	16
Figura 10 <i>Comandos de virtual Studio para funcionamiento del personaje</i>	17
Figura 11 <i>Eventos creados en Fmod para la escena The Viking Village</i>	18
Figura 12 <i>Proceso de creación de Studio Listener y Event Emitter</i>	19
Figura 13 <i>Vista de despliegue de opciones de Event Emitter</i>	20
Figura 14 <i>Vista final de la selección de objeto y creación de Studio listener y Event Emitter</i>	20
Figura 15 <i>Producción musical original para la scena de Unity The Viking Village</i>	21
Figura 16 <i>Forma de adherir componentes como trigger</i>	22
Figura 17 <i>Vista de los componentes trigger enter y trigger exit en el panel inspector</i>	23
Figura 18 <i>Secciones de música adaptativa en Fmod</i>	24
Figura 19 <i>Menús del logic track para la música adaptativa</i>	25
Figura 20 <i>Configuración “Add transition región to” en música adaptativa</i>	26
Figura 21 <i>Configuración de la condición de transición del área de transición</i>	26
Figura 22 <i>Configuración de la condición de transición del área de transición</i>	27

Figura 23 <i>Configuración de la condición de transición del área de transición</i>	28
Figura 24 <i>Personaje con espada activando trigger</i>	29
Figura 25 <i>Ventana de acceso al mixer de Fmod</i>	30
Figura 26 <i>Grupos de Fmod creados para simplificar la mezcla</i>	31
Figura 27 <i>Sincronización live Update</i>	32
Figura 28 <i>Ventana de búsqueda para la opción profiler</i>	32
Figura 29 <i>Creación de nueva sesión de profiler</i>	33
Figura 30 <i>Imagen de la muestra de perfil grabada en el Profiler de Fmod</i>	34
Figura 31 <i>Opción de búsqueda de recursos</i>	34
Figura 32 <i>Tabla de recursos del sistema Fmod</i>	35

Lista de Anexos

Anexo A <i>Carpets de Practica Académica</i>	36
---	----

Introducción

En el siguiente informe se consignará el proceso de aprendizaje obtenido y el resultado de cada uno de estos pasos para el desarrollo o creación del diseño sonoro para una escena de videojuego en Unity llamada *The Viking Village*.

Este proceso de aprendizaje tiene fundamento en el *diplomado de profundización en Audio Dinámico para Videojuegos* el cual consistió en cinco tareas con temáticas, de acuerdo a las guías, las cuales pueden agruparse de la siguiente manera: Tarea 1: Análisis de un videojuego, Tarea 2: Generación de un ambiente sonoro dinámico, Tarea 3: Sonorización de un proyecto de videojuego, Tarea 4: Programación del diseño sonoro del videojuego, Tarea 5: Mezcla, pruebas y optimización de audio del videojuego. Las dos tareas iniciales corresponden a fundamentos y usos del software de musicalización, sin embargo, las siguientes son el trabajo fuerte en cuanto a sonorización del entorno del videojuego.

Este informe lo que pretende mostrar es el desarrollo de las tareas y el aprendizaje demostrado a través de la creación de sonidos (música, ambientes y efectos), además del seguimiento de las instrucciones para obtener el resultado mostrado y obtenido en este informe durante la participación del *diplomado en Audio Dinámico para Videojuegos* de la Universidad Nacional Abierta y a Distancia UNAD.

Objetivos

Objetivo General

Proporcionar el diseño del Audio Dinámico en FMOD, para una escena de videojuego en Unity llamada The Viking Village

Objetivos Específicos

Usar y diseñar eventos sonoros en el Midleware FMOD.

Usar escena del motor de videojuegos: Unity, para la sonorización.

Integrar y Programar el Midleware FMOD con el motor de Videojuegos Unity.

Crear música, efectos, y ambientes para la escena de Unity *The Viking Village* para obtener un entorno deseado siguiendo las guías de trabajo.

Desarrollo de la explicación del proceso de diseño sonoro dinámico de la escena del videojuego The Viking Village

Tarea 1: Análisis de un Videojuego

Para esta tarea se seleccionó un video juego muy especial ya que se ha adentrado en los corazones de grandes y chicos, y que le ofrece al jugador un alto nivel de libertad además de sus otras categorías como supervivencia, pacifico entre otras. Este videojuego es Minecraft y tiene, como es común, un desarrollo amplio en diferentes plataformas que permite su accesibilidad desde cualquier dispositivo

Figura 1

Interfaz de inicio de Minecraft



Nota: El gráfico muestra la interface inicial del videojuego Minecraft de MoJang, versión 1.16.

Tomado de Minecraft, 2024, Copyright MoJang AB.

Maincraft es un videojuego lanzado el 17 de mayo de 2009 por la compañía Mojang Studio que fue comprado por Microsoft en el 2014, este juego es de tipo sandbox lo que permite una visual de mundo abierta, con disposición para la creación y el desarrollo de habilidades que se adquieren durante la participación en el juego.

Dentro del análisis del audio se tuvieron en cuenta varios momentos en cuento, el primero de ellos fue el momento inicial en el que se presentaron los sonidos de la interfaz que son sonido, luego se dio inicio al audio del video juego y encontramos los sonidos de los pasos y los efectos al interactuar con el entorno mientras que escuchábamos la música de fondo o la música de Minecraft, lo que nos permite tener categorías de música, efectos, sonidos de ambiente y de interfaz.

Luego adentrándonos un poco más en el juego encontramos audio interactivo y adaptativo en una zona de agua, ya que cada vez que cambia el ambiente el sonido cambia para darnos una sensación de que estamos en ese ambiente específico como por ejemplo el agua. A interactivo ya que el sonido nos avisa por ejemplo que nos encontramos en un momento de peligro al acabarse el aire dentro del agua con sonidos que nos dan a entender el aspecto de la escena en la que estamos, y la variabilidad en las escenas cambian de acuerdo con el clima o con el momento del día en el que nos encontremos, lo que lo hace adaptando a la naturaleza de lo que estamos viviendo en el momento.

Figura 2

Momento adaptativo e interactivo en Minecraft



Nota: El gráfico muestra escena del videojuego Minecraft de MoJang, versión 1.16. Tomado de *Minecraft*, 2024, Copyright MoJang AB.

Tarea 2: Generación de un Ambiente Sonoro Dinámico

Para esta tarea se eligió un modelo de ambiente sonoro independiente, es decir con música original y que llevará por nombre “*Espacio Sideral*” basado en un momento imaginario que simula el espacio, a lo mejor con un astronauta en un ambiente de ciencia ficción en un vehículo espacial, donde el ambiente general semeja videojuegos como *Aliens*, *Doom*, *Prometeo*, entre otros.

Figura 3

Interfaz de Fmod Studio con vista de proyecto



Nota: El gráfico muestra la vista del proyecto de creación sonora de Fmod Studio, versión 2.02.23, *Imagen propia*, Copyright 2012 - 2024

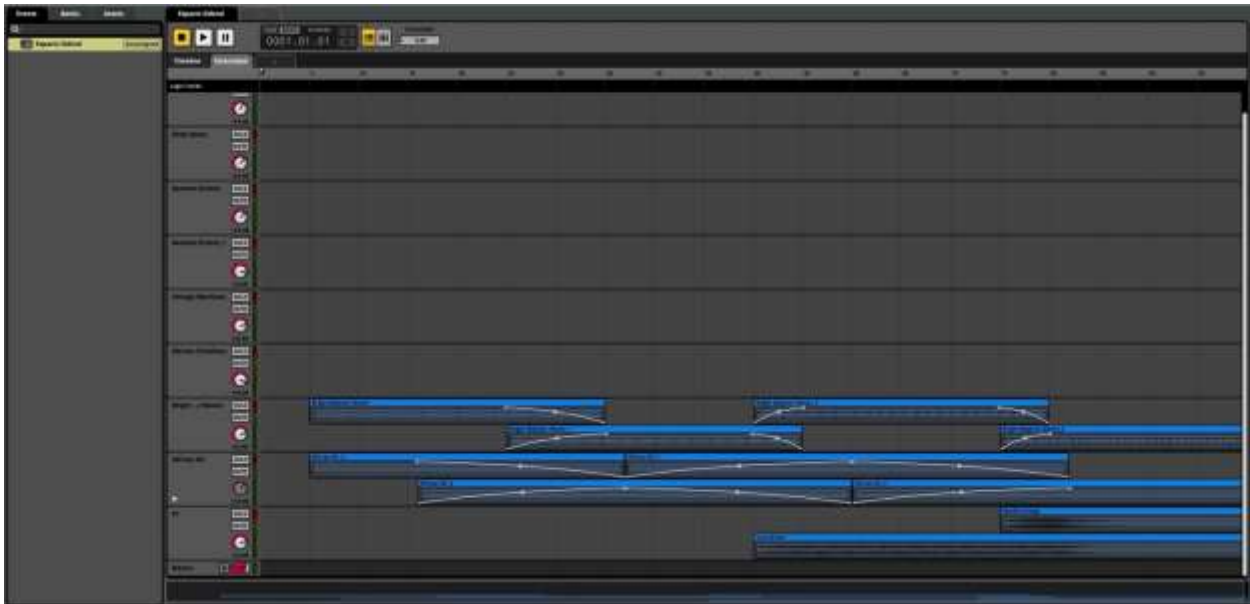
Además, se crea un evento sonoro con un parámetro de identidad que modifica dicho estado, que sumerge al oyente en un estado de desesperación o un momento confuso al enfrentarse a enemigos en la nave o el espacio donde se desarrolla la imagen ficticia e imaginaria que permite esta creación musical. Al mismo tiempo se construye un banco de assets con sonidos originales creado en el DAW Logic Pro, esto sonido tiene como característica que son mezclables y trocables entre sí, lo que los hace muy versátiles para superponerlos en cualquier momento para que no se hagan repetitivos y sea un poco más adaptable a las necesidades.

Luego en Fmod se prepararon alrededor de nueve instrumentos sonoros combinables entre sí, de los cuales cinco de ellos con multi -instrument para ajustar seis en el timeline, y tres

más usados en el parámetro de intensidad. Estos parámetros fueron verificados en funcionamiento para su correcta ejecución y con una descripción del evento.

Figura 4

Vista de la interfaz de Fmod con el parámetro de Intensidad



Nota: El gráfico muestra la vista del proyecto de creación sonora de Fmod Studio en la ventana de parametros, versión 2.02.23, *Imagen propia*, Copyright 2012 - 2024

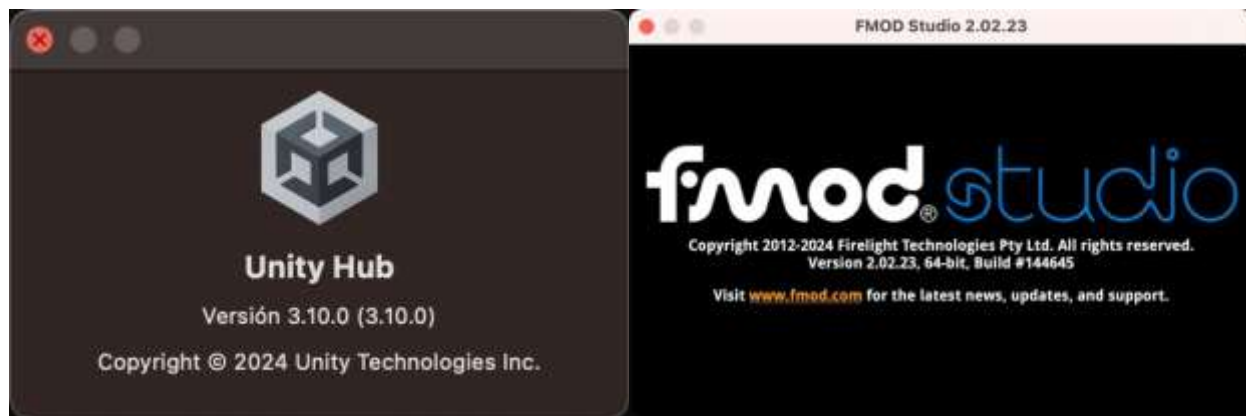
Tarea 3: Sonorización de un Proyecto de Videojuego

La tarea tres al igual que las demás, contiene varias actividades consecuentes con las que es necesario ya que depende de su orden para un correcto funcionamiento, y se habla específicamente de los programas FMOD y UNITY, lo cuales deben ser descargado con la guía de compatibilidad de ambos programas para su correcto funcionamiento.

En nuestro caso, las versiones fueron para UNITY la versión 3.10.0 del 2024, y para Fmod Studio la versión 2.02.23 de 2012-2024.

Figura 5

Imagen con versiones de Unity Hub y Fmod Studio



Nota: El gráfico muestra las versiones compatibles de Fmod Studio y Unity Hub, Fmod Studio versión 2.02.23, Version Unity Hub 3.10.0, *Imagen propia*, Copyright 2012 – 2024 y Copyright © 2024 Unity Technologies Inc.

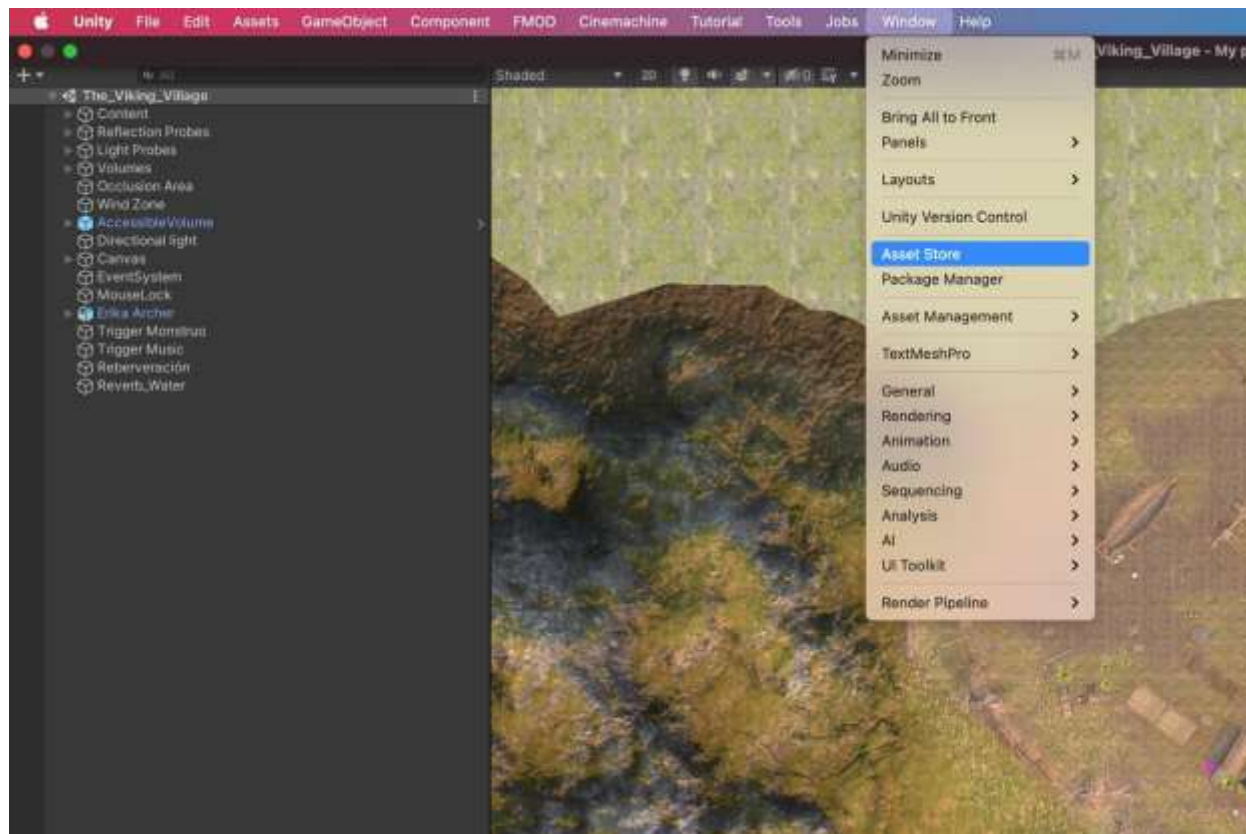
Es importante seguir la guía, ya que para que ambos funcionen no solo es necesario descargar las versiones apropiadas si no también un pequeño software que permitirá su integración.

Una vez realizada la integración exitosa de ambos programas se procede a dar inicio a la escena de Unity, debido a que la primera parte fue el reconocimiento de Fmod para la creación de sonidos y disposición de sonidos, para esto una vez abierto el motor de videojuegos se procedió a buscar una escena para el desarrollo de la actividad y la elegida fue “The Viking Village” que se encuentra alojada en la *Assets Store* de Unity, a la cual se puede acceder desde el programa y buscar un sinnúmero de posibilidades de escenas pagas hasta escenas de uso gratuito, las cuales se pueden editar y reubicar o simplemente adecuar a las necesidades del estudiante. Son

escenas de alta calidad y muy buenos gráficos ahorrando mucho tiempo en la preparación de nuestro trabajo.

Figura 6

Guía de acceso a la assets store de Unity



Nota: El gráfico muestra la forma de acceso a assets store del software Unity, versión 3.10.0,

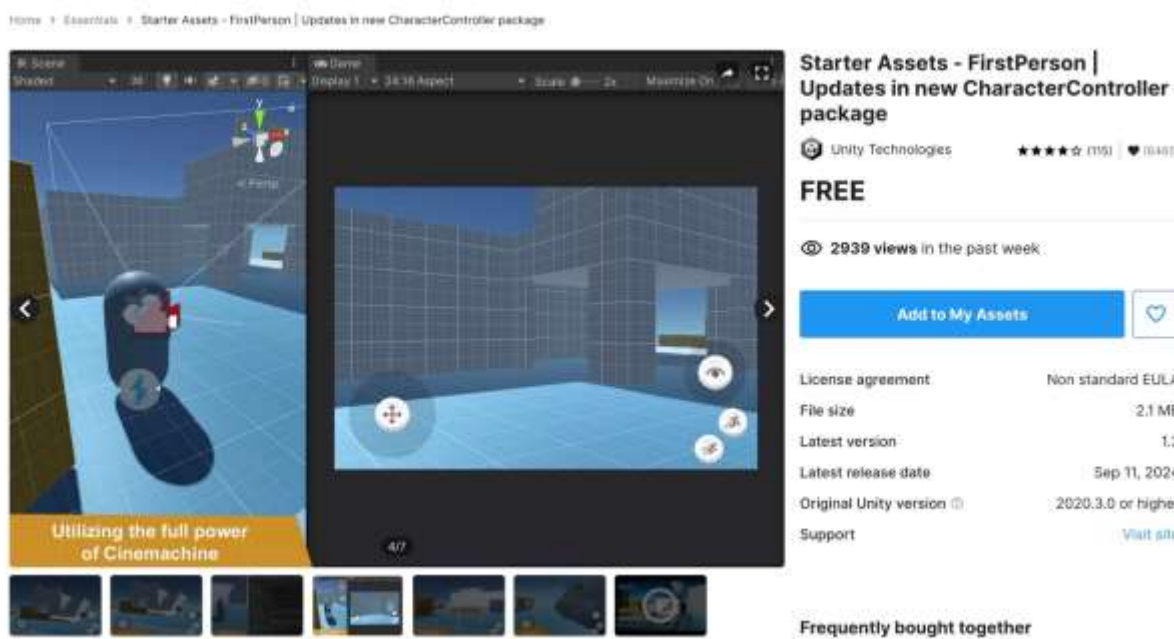
Imagen propia, Copyright © 2024 Unity Technologies Inc.

Para esta escena faltaba un personaje, es importante aclarar que la descarga de la escena no incluye el personaje ya que este debe ser tu creación, en la assets store solo te proveen de la escena, pero esta misma se descarga el Starter Assets – First Person, que es la figura, el controlador sin diseño de un personaje, y sobre se realizó el trabajo de creación.

Para nuestro personaje se escogió una arquera llamada Erika, la cual quedaba perfecta para nuestra escena de carácter medieval ya que porta un atuendo muy de la época con colores tierra y texturas que semejan el cuero, el fique y demás de este periodo de la historia. Para este Maximo, empresa creadora de estos personajes tiene muchos por ofrecer de la misma manera que Unity provee las escenas, gratuitas y de pago, pero los movimientos son otro tema que se tuvo en cuenta ya que tomamos el personaje mas no su forma de moverse en el juego lo cual fue programado también para su uso en The Viking Village.

Figura 7

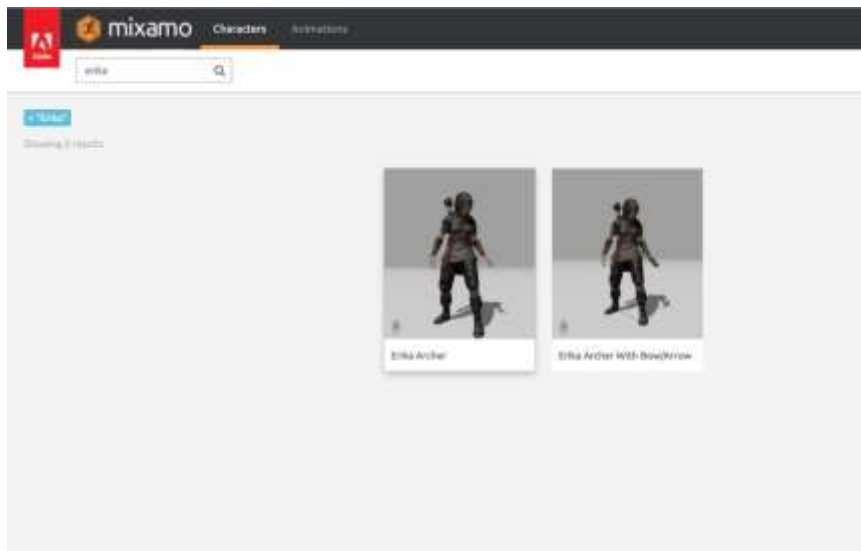
Imagen del controlador de juego



Nota: El gráfico muestra la forma de acceso al personaje controlador del software Unity, versión 3.10.0, Tomado de *Assets Store de Unity*, Copyright © 2024 Unity Technologies Inc.

Figura 8

Imagen del personaje Erika Archer provisto



Nota: El gráfico muestra la forma de acceso al personaje controlador del motor de videojuegos

Unity, versión 3.10.0, Tomado de: www.maximo.com.

Figura 9

Opciones de animación, tomado de www.maximo.com



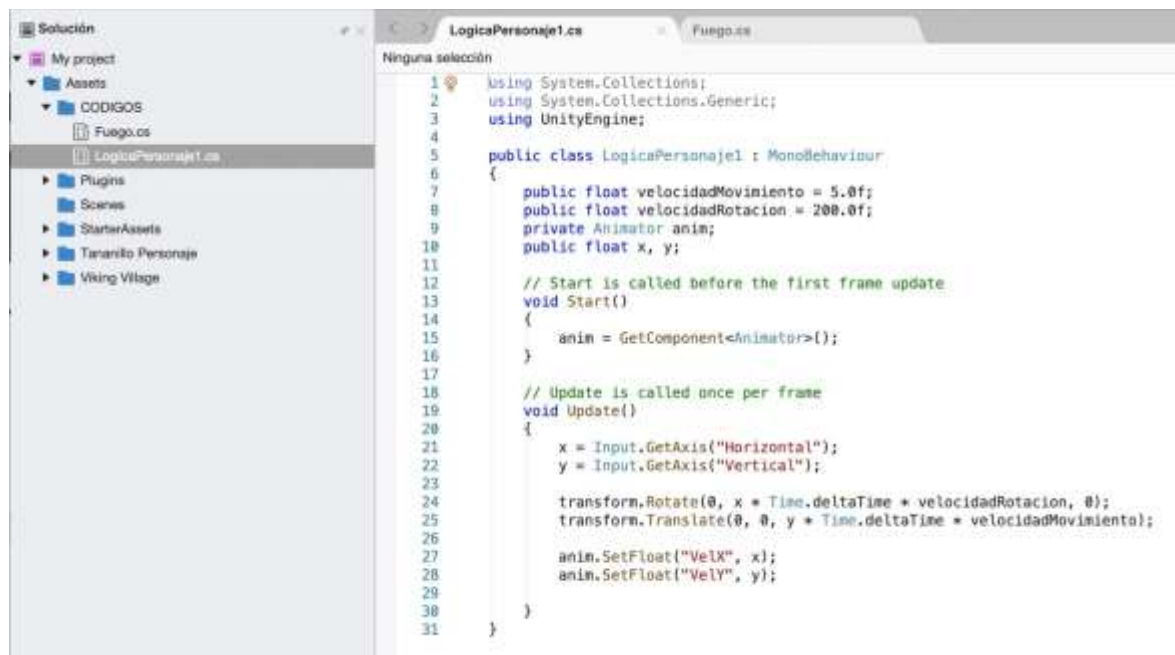
Nota: El gráfico muestra los skins de uso para el personaje controlador del motor de videojuegos

Unity, versión 3.10.0, Tomado de: www.maximo.com.

Para el funcionamiento del personaje en el video juego se insertaron algunos comandos que permitieron la sincronización de este con el entorno de la escena, esto fue en el *Virtual Studio de Unity*.

Figura 10

Comandos de virtual Studio para funcionamiento del personaje



```

1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class LogicaPersonaje1 : MonoBehaviour
6  {
7      public float velocidadMovimiento = 5.0f;
8      public float velocidadRotacion = 200.0f;
9      private Animator anim;
10     public float x, y;
11
12     // Start is called before the first frame update
13     void Start()
14     {
15         anim = GetComponent<Animator>();
16     }
17
18     // Update is called once per frame
19     void Update()
20     {
21         x = Input.GetAxis("Horizontal");
22         y = Input.GetAxis("Vertical");
23
24         transform.Rotate(0, x * Time.deltaTime * velocidadRotacion, 0);
25         transform.Translate(0, 0, y * Time.deltaTime * velocidadMovimiento);
26
27         anim.SetFloat("VelX", x);
28         anim.SetFloat("VelY", y);
29
30     }
31 }

```

Nota: El gráfico muestra los códigos para el funcionamiento del controlador de personaje de Visual Studio para Unity, Imagen propia.

Una vez solucionado este paso continuamos con el diseño sonoro para este proyecto, en el que inicialmente se incluyeron pocos sonidos dentro de los cuales se destacan tres grupos como lo son: ambientes, efectos sonoros y música. Dentro de la música se buscó una que pudiera tener un aspecto de cultura celta o escoces, para el efecto se utilizó un sonido de fuego que fue puesto en la antorcha, y en los ambientes se pusieron sonidos de pájaros y sonido de agua

Los sonidos fueron inicialmente como eventos en Fmod, para luego de la sincronización de ambos programas pasar a ubicarlos en Unity en el lugar que los necesitamos, por ejemplo, la música fue puesta para el personaje de Erika, ya que este debe cambiar luego como música adaptativa para las siguientes tareas, el fuego se puso en una antorcha cercana con efecto de cercanía, para que se pudiera escuchar más fuerte conforme el personaje se acercara. Y el ambiente tiene esa misma característica del efecto.

Figura 11

Eventos creados en Fmod para la escena The Viking Village



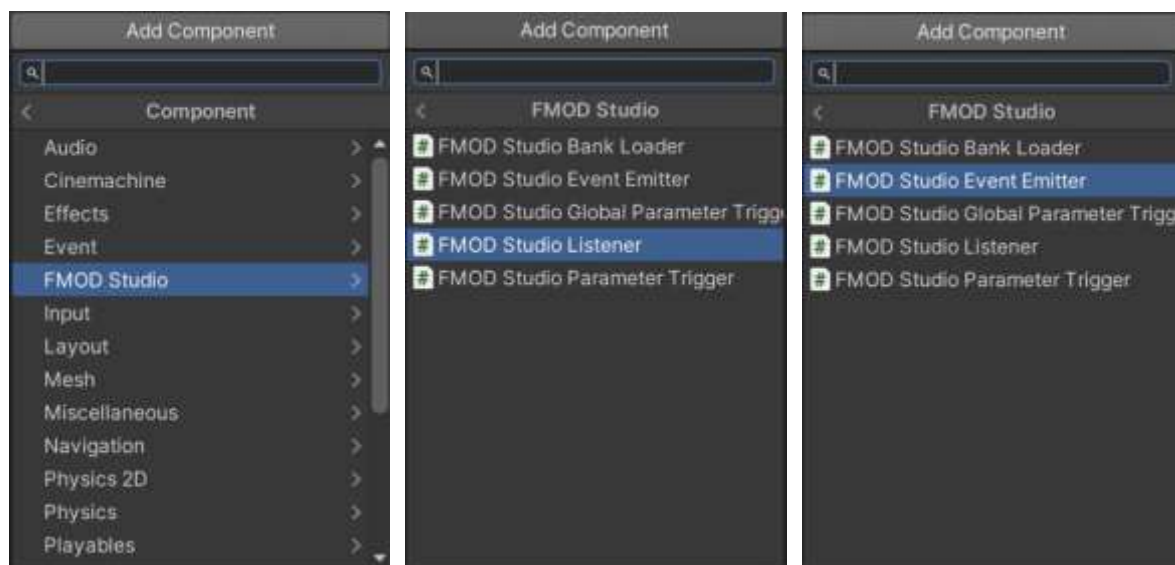
Nota: El gráfico muestra la interface de Fmod con los eventos sonoros creados para la escena de Unity, versión 3.10.0, *Imagen propia.*

La imagen a continuación muestra el proceso en el que se ubicó la música, inicialmente se selecciona el personaje, *Erika Archer*, que se encuentra en el panel de jerarquías al extremo

izquierdo de Unity, luego de esta selección aparece el inspector en el extremo derecho que nos muestra todas las cualidades del objeto seleccionado anteriormente, una vez en el *inspector* se procede a seleccionar con un clic *add component*, seleccionamos *FMOD studio*, luego *Studio Listener*, luego repetimos los pasos para *Event Emitter*.

Figura 12

Proceso de creación de Studio Listener y Event Emitter

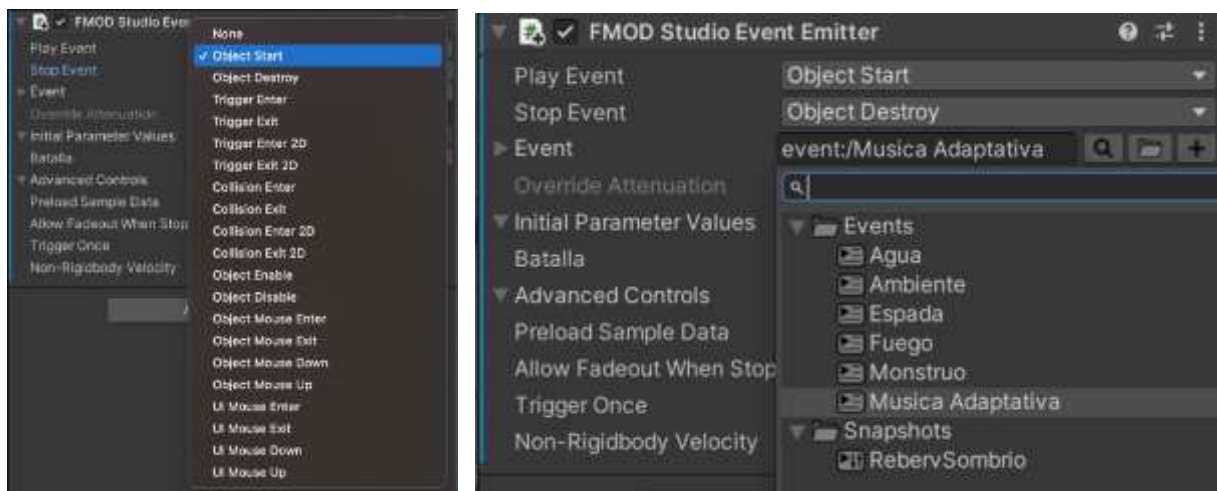


Nota: El gráfico muestra la secuencia de creación de un studio listener y un event emitter para introducir un evento sonoro en Unity. *Imagen propia.*

Una vez creado estas opciones procedimos a su configuración que consiste en que el *Studio Listener* quede así tal cual, ya que este nos permitirá escuchar lo que vamos a configurar con el *Event emitter* que es quien carga el sonido a escuchar. Ahora bien, el *event emitter* debe tener dependiendo de la necesidad un *Objet Start* en *Play Event*, y un *Object Destroy*, es decir que las opciones pueden cambiar a *trigger enter* y *trigger exit* o *collision enter* y *collision exit*.

Figura 13

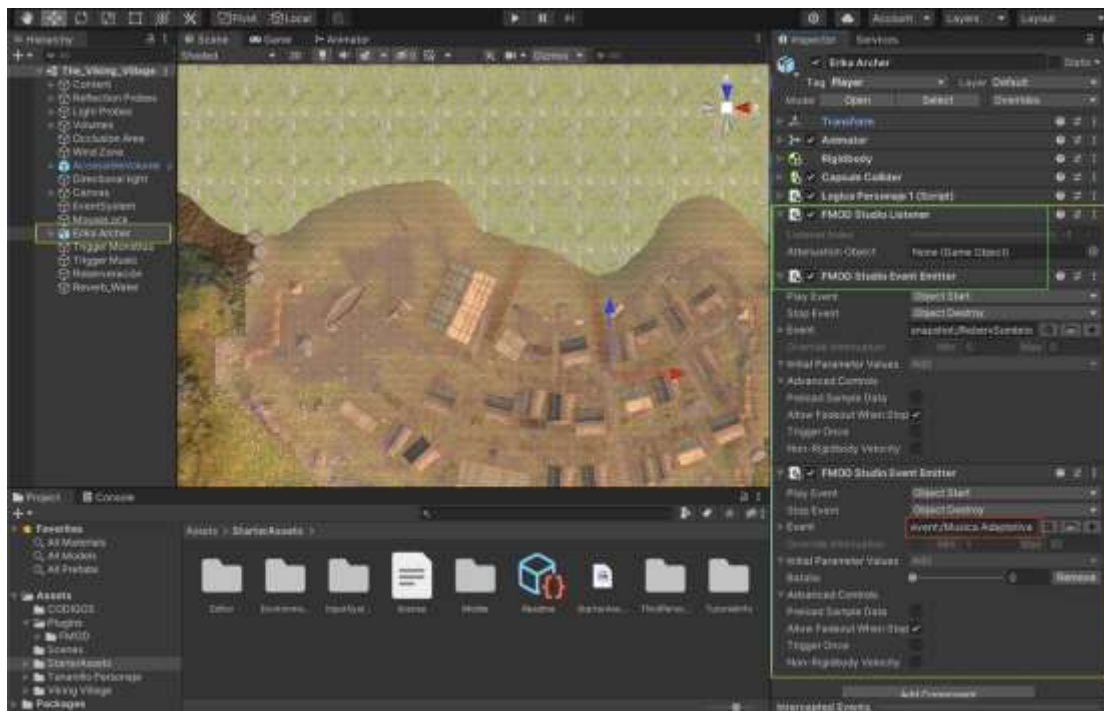
Vista de despliegue de opciones de Event Emitter



Nota: El gráfico muestra la secuencia para la selección y asignación de un evento emitter para introducirlo en Unity. *Imagen propia.*

Figura 14

Vista final de la selección de objeto y creación de Studio listener y Event Emitter



Nota: El gráfico muestra la secuencia la visual en líneas de colores de un objeto, de un studio listener, y la selección del evento emitter en Unity. *Imagen propia.*

Tarea 4: Programación del Diseño Sonoro del Videojuego

Esta tarea fue la más exigente del diplomado y consiste en primer lugar en crear música adaptativa, la cual fue compuesta en el software *Logic Pro X*, siendo una producción original para escena. Este paso tomo tiempo en la composición y luego el ingreso de los datos musicales a Unity a través de Fmod.

La música constó de percusión y sonidos con características inglesas, celtas y escocesas, se grabó en la escuela de música de Titiribí – Antioquia y fue el modelo para construir la música adaptativa de este proyecto, la melodía tiene un aire muy de la zona y preciso para esta escena.

Figura 15

Producción musical original para la escena de Unity The Viking Village



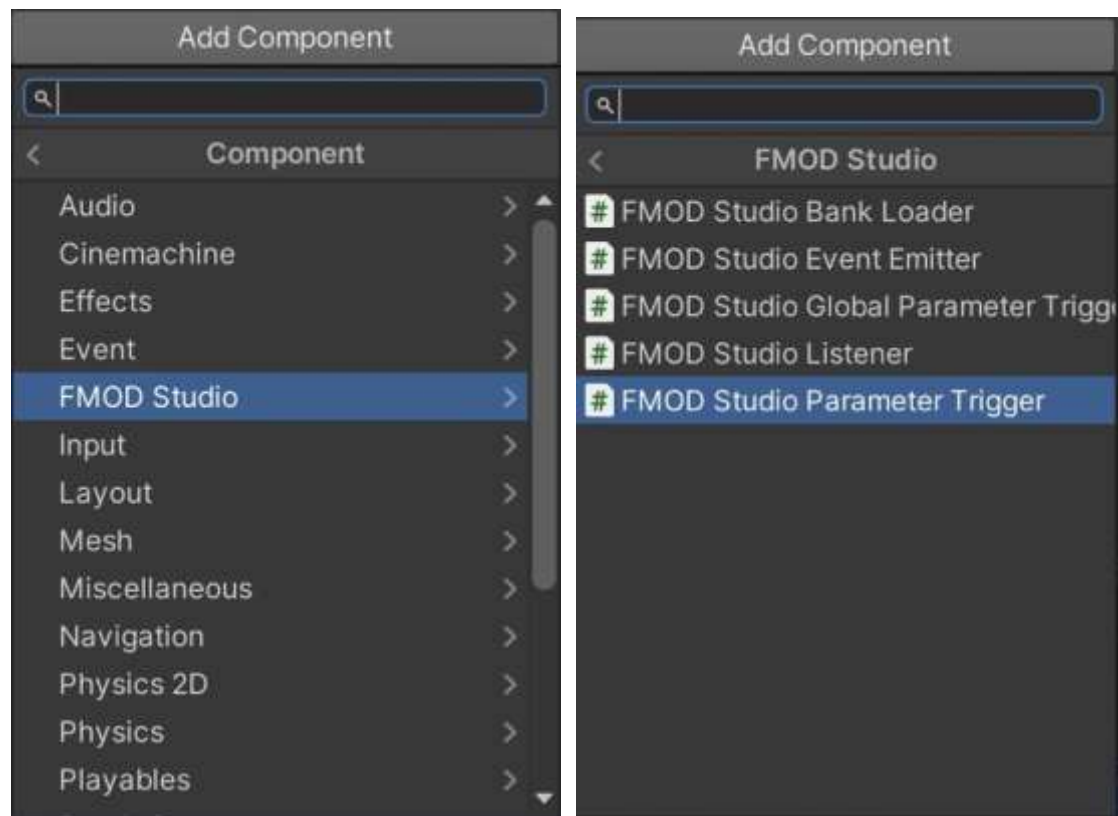
Nota: El gráfico muestra las secuencias de audio propio creadas en el DAW Logic Pro X, para la musicalización de la escena de Unity a ingresadas en Fmod Studio. *Imagen propia.*

Esta música tiene una base que permanece durante el cambio del proceso adaptativo del audio, pero su intensidad e instrumentación cambia drásticamente a pesar de continuar con el mismo ritmo. Este sonido fue ubicado como lo vimos anteriormente en el personaje y cambia cuando se aproxima a una edificación la cual fue programada para tener dentro un momento de batalla.

Este parámetro se activa mediante *trigger* el cual permite que cuando el personaje ingrese a la zona crítica la música cambie, en Unity su programación se ve de la siguiente manera, se debe ir al panel de jerarquía en el extremo izquierdo, seleccionar el objeto, regresar al extremo derecho al inspector para adherir un par de componentes: los *trigger enter* y *trigger exit*, de esta manera, uno para permitir la entrada y cambiar la música y otro para salir de la zona de tensión.

Figura 16

Forma de adherir componentes como trigger

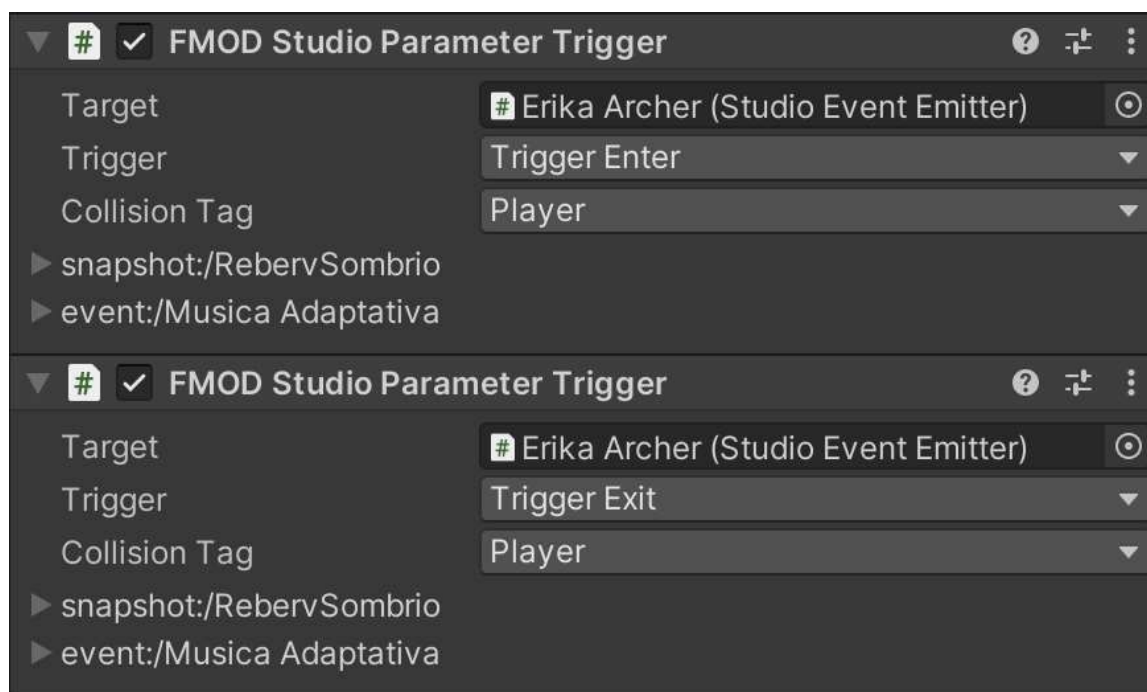


Nota: El gráfico muestra la secuencia de creación de un studio parameter trigger en Fomd para introducir un disparador en un evento sonoro en Unity. *Imagen propia.*

Una vez realizado este paso se debe configurar como trigger enter y exit debidamente y en *Collision Tag*. Se debe poner *player*, sin antes seleccionar al personaje y etiquetarlo como el *player* también, ya que este es quien será reconocido al momento de colisionar con el espacio que programamos anteriormente

Figura 17

Vista de los componentes trigger enter y trigger exit en el panel inspector



Nota: El gráfico muestra la vista recortada del inspector de Fmod Studio donde se aprecia un studio parameter trigger con su respectiva entrada y salida. *Imagen propia.*

A diferencia de Unity en Fmod tenemos que configurar también la forma en la que se cambia la música al momento en que el personaje llegue a la zona de batalla. Este debe tener dos

secciones separadas y marcadas debidamente, esto es de suma importancia ya que gracias a la marca podremos dar instrucciones al programa para que haga sus entradas respectivas.

Figura 18

Secciones de música adaptativa en Fmod



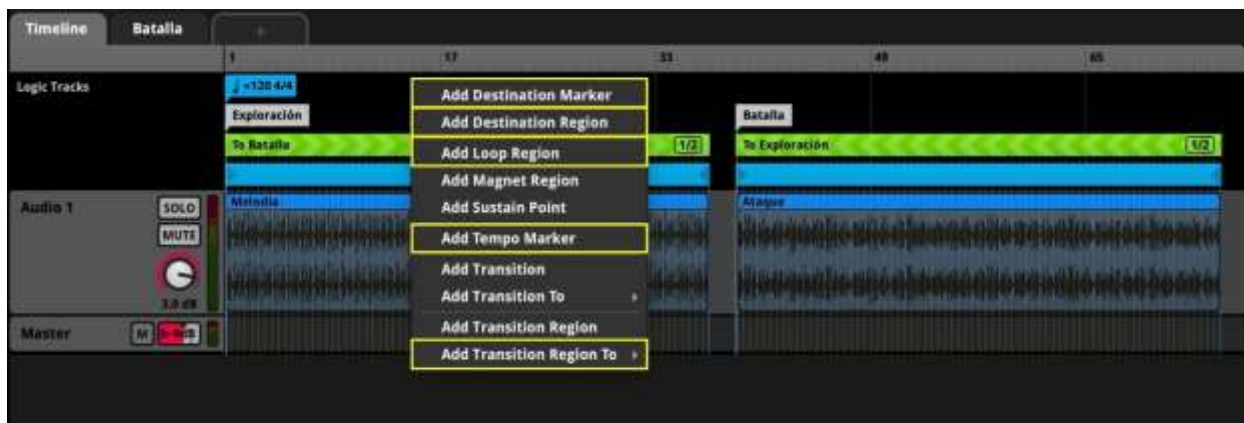
Nota: El gráfico muestra un evento sonoro de música adaptativa de Fmod para Unity. *Imagen propia.*

Buscamos la marca de tiempo o el tiempo en el que fue compuesta la pieza musical, esto en Fmod se llama *Tempo Maker*, una vez hecho esto procedemos a marcar las dos secciones usando *Add Destination Marker*, de ser posible escribamos como se llamarán las partes para no confundirnos, creamos luego el *Loop Region* que nos permitirá repetir permanentemente la música para que no se agote hasta que cambie de nuevo y viceversa, y luego necesitamos crear

las transiciones para que el personaje una vez entre en el modo de batalla pueda cambiar la música, esto lo hacemos con *Add Transition Region*. Estos menús salen de oprimir al frente del *Logic Track* con el clic derecho del mouse del computador como se muestra en la siguiente figura.

Figura 19

Menús del logic track para la música adaptativa



Nota: El gráfico muestra el menú para la asignación de diferentes parámetros en líneas de color amarillo, necesarios para el adecuado funcionamiento de la música adaptativa. *Imagen propia.*

En el menú de *Add Transition Region* es importante configurar la transición de acuerdo con las marcas creadas, es decir desde *exploración* en mi caso, a batalla y viceversa ya que esta configuración la que nos va a permitir que el personaje pueda salir y regresar de estos dos modos y que la música haga su entrada o cambio al estado normal del juego.

Figura 20

Configuración “Add transition región to” en música adaptativa



Nota: El gráfico muestra la configuración de Fmod en el timeline para las regiones de transición en la música adaptativa de Fmod para Unity. *Imagen propia.*

Ahora bien, la configuración debe tener un parámetro para que este pueda sincronizarse con Unity y que este último nos de la opción para poder transformar ese paso, entonces para esto damos clic izquierdo sobre la región de transición para encontrar en la parte inferior las condiciones de transición, por lo que cada una debe tener un parámetro diferente ya que son momentos diferentes, nosotros lo hicimos con 0 y 1 de la siguiente manera.

Figura 21

Configuración de la condición de transición del área de transición



Nota: El gráfico muestra la configuración de las condiciones de las áreas de transición para un adecuado funcionamiento y cambio de ambientes en la música adaptativa del videojuego.

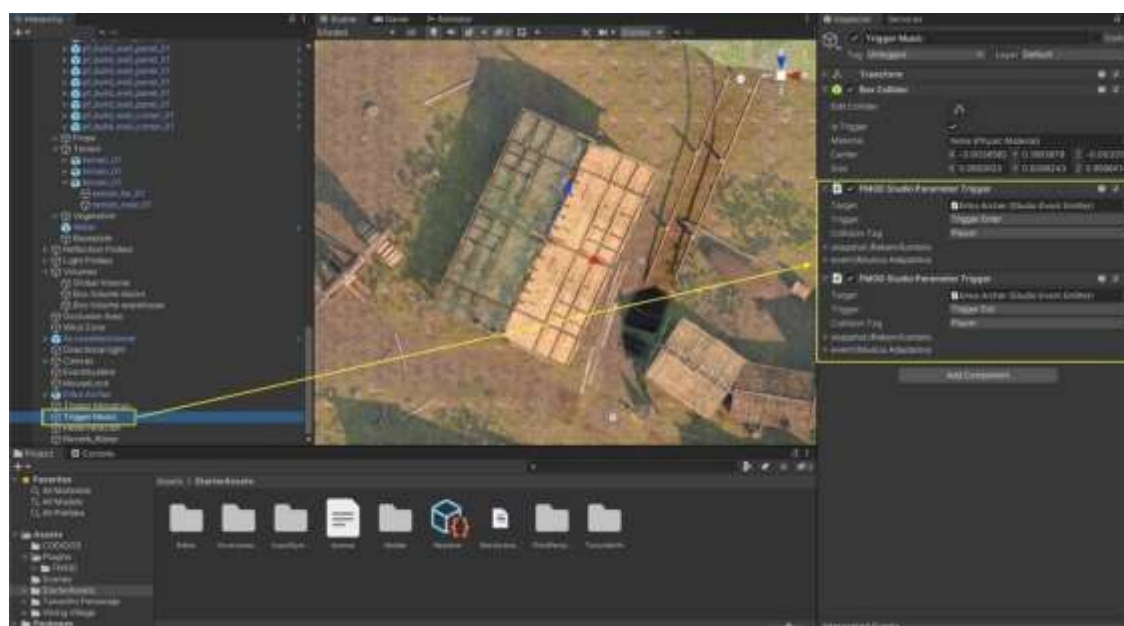
Imagen propia.

Ya en el motor del videojuego se crea un objeto para ubicarlo en un edificio o en el lugar donde se desea, para que se active con el *box collider* al momento del contacto con el jugador o el personaje. Este *parameter trigger* se configura como un *trigger enter* y un *trigger exit* para que cambie de modo al momento en el que el personaje entre o salga del área crítica.

Para que todo lo demás ocurra entonces al personaje que es el objeto que carga con la música se crea un *event emitter* con la música adaptativa y con el parámetro de *Batalla*, para que al momento de ingresar al área y colisionar con el *box collider* pueda hacerse el cambio de música y generar el ambiente deseado para crear la atmosfera que queremos darle a nuestro video juego.

Figura 22

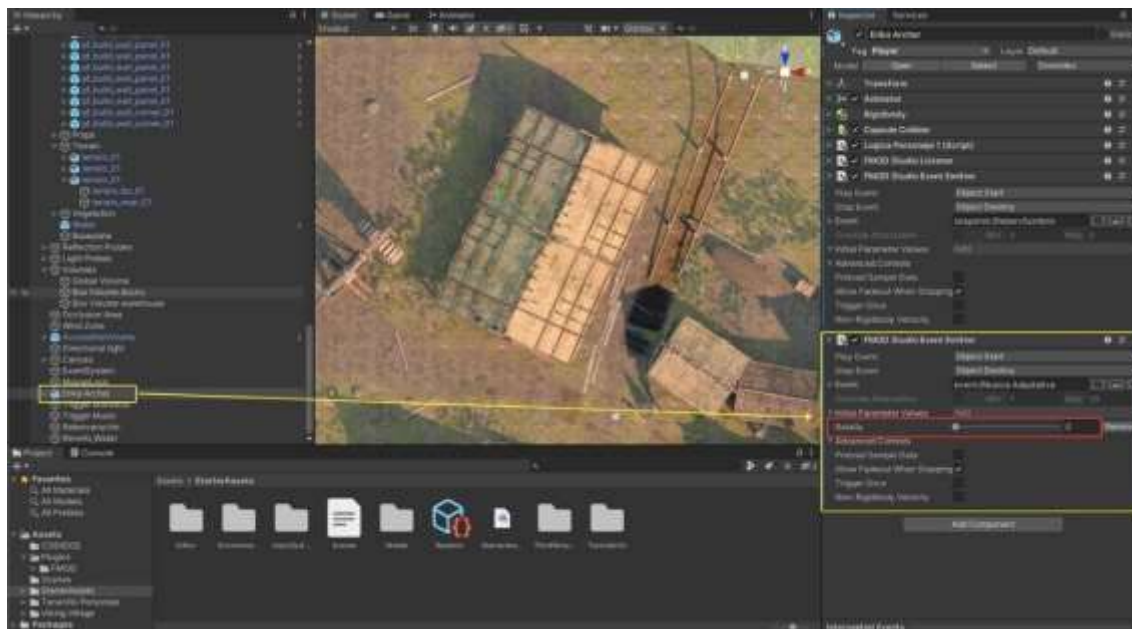
Configuración de la condición de transición del área de transición



Nota: El gráfico muestra la configuración de las áreas de transición en Unity que permitirán el adecuado funcionamiento de la imagen de Fmod en la música adaptativa. *Imagen propia.*

Figura 23

Configuración de la condición de transición del área de transición

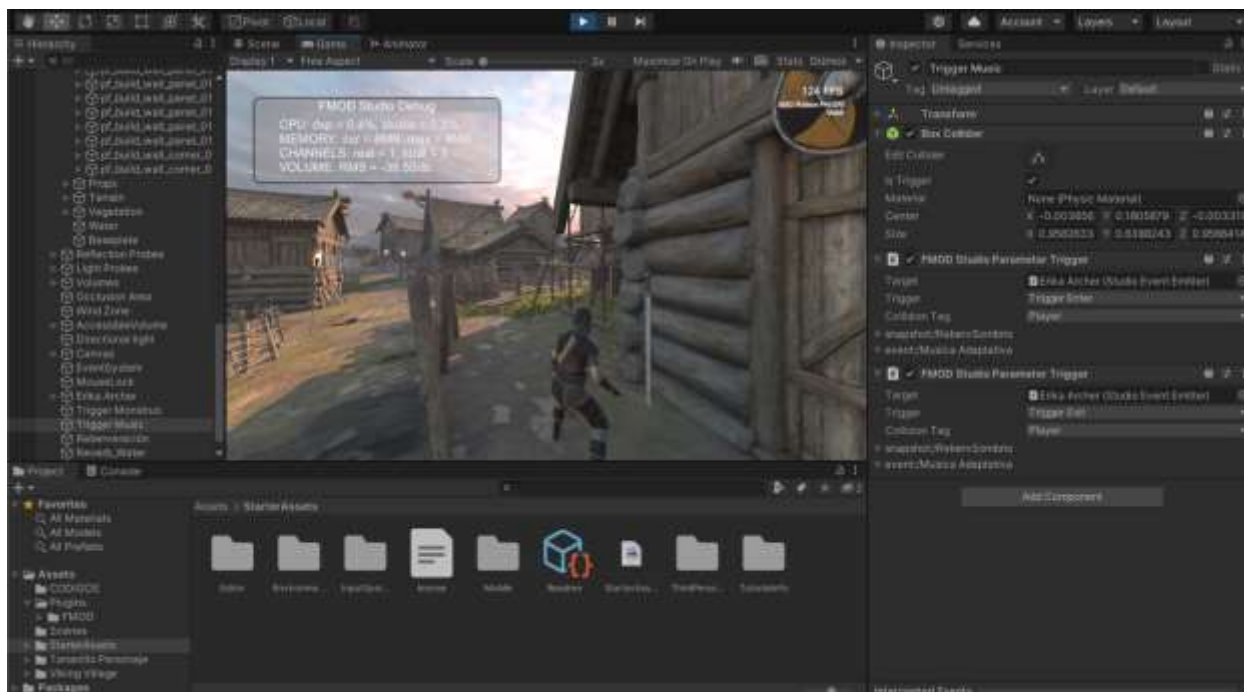


Nota: El gráfico muestra las condiciones para la transición de las áreas en Unity. *Imagen propia.*

Continuando con el uso de los disparadores (*triggers*) se ubica de la misma manera un sonido de espada en una cercana al jugador para que se active de esta manera al momento del contacto, simulando tener ese objeto como parte de sus herramientas de batalla.

Figura 24

Personaje con espada activando trigger



Nota: El gráfico muestra el personaje en escena junto a espada junto antes de activar un trigger en el motor de videojuegos Unity. *Imagen propia.*

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

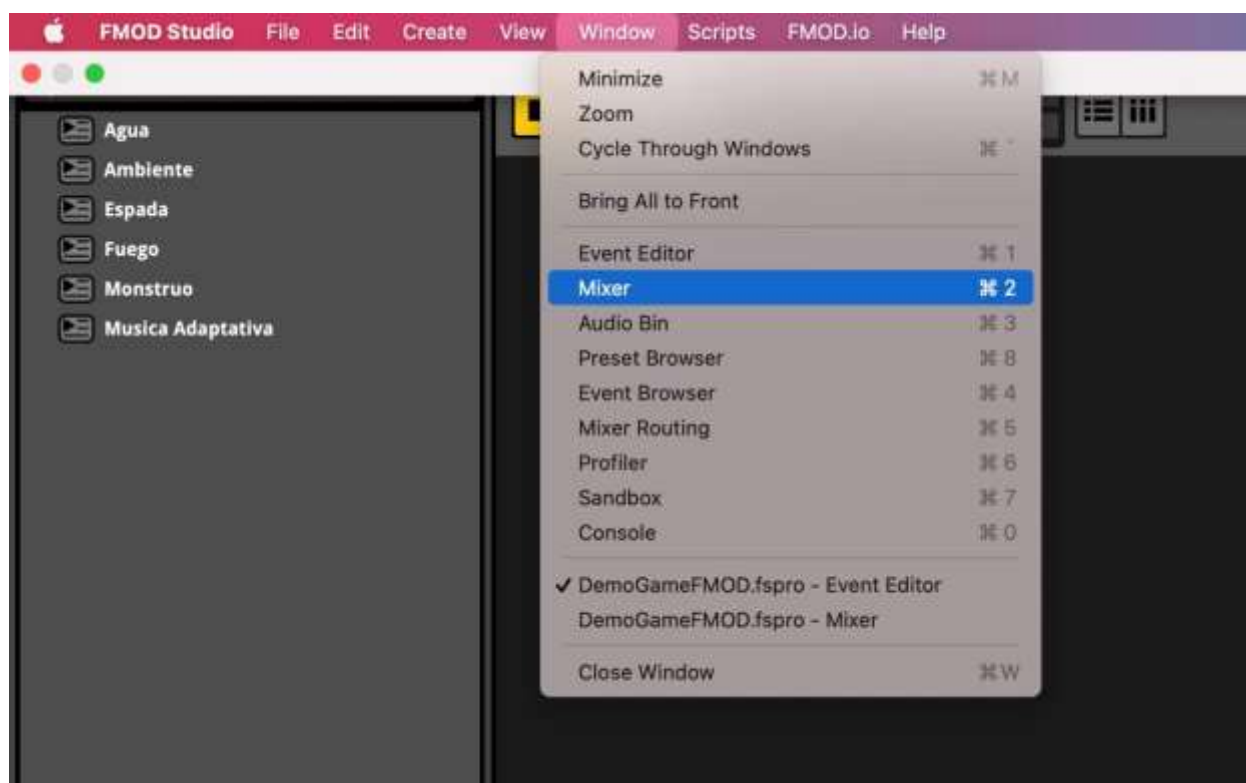
Esta actividad consistió en hacer de todo el audio creado para las diferentes partes del videojuego una mezcla y un test o prueba que permita tener todos los sonidos en orden, con calidad, con un buen acabado, con niveles adecuados y con todos los aditamentos necesarios para que el espectador del videojuego pueda tener una experiencia única en el desarrollo de este.

Esta mezcla se desarrolla especialmente desde Fmos, sin embargo, para esto debemos correr el juego para tomar una captura y realizar una grabación de audio *in situ*, que nos permitirá ver cómo se comporta el audio durante el desarrollo del videojuego.

Para esto entonces iniciamos con la mezcla creando grupos de acuerdo con la cercanía o a las características de los sonidos, para que puedan compartir parámetros y así optimizar de mejor manera los envíos al bus master del Fmod. Para ello entonces accedemos la ventana del mixer yendo a la opción de ventana (window) y buscamos la opción como se muestra en la imagen siguiente, allí nos aparece otro panel en el que trabajaremos adecuadamente nuestros grupos, y tenemos la opción de visualizar todas las perillas de cada uno de los instrumentos para realizar los ajustes deseados, por lo que nosotros realizamos los nuestros para obtener el mejor resultado.

Figura 25

Ventana de acceso al mixer de Fmod



Nota: El gráfico muestra la forma de acceso al mixer del software Fmod Studio para la mezcla del audio de Unity. *Imagen propia.*

Figura 26

Grupos de Fmod creados para simplificar la mezcla

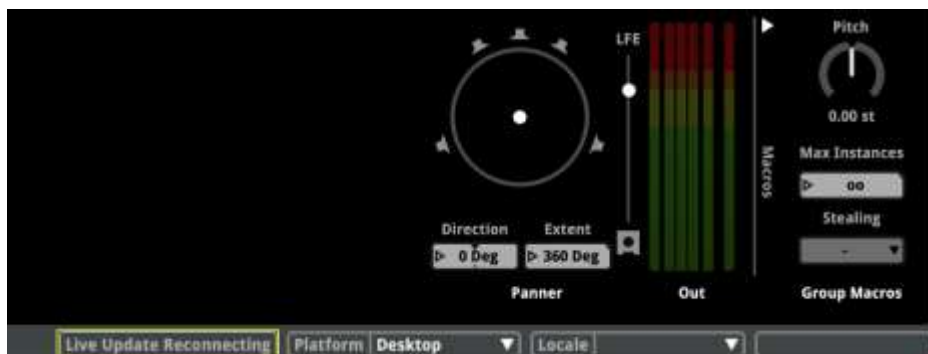


Nota: El gráfico muestra la agrupación de los sonidos para maximizar los procesos de mezcla sonora de la imagen en Fmod. *Imagen propia.*

Una vez organizados los grupos, que ninguno de nuestros sonidos nos muestre sobresaltos inadecuados y de haber realizado la mezcla respectiva nos dirigimos de nuevo al editor de eventos para visualizar en la parte inferior una pequeña pestaña en gris con la opción *live Update off*, la presionamos para encender la opción.

Figura 27

Sincronización live Update.



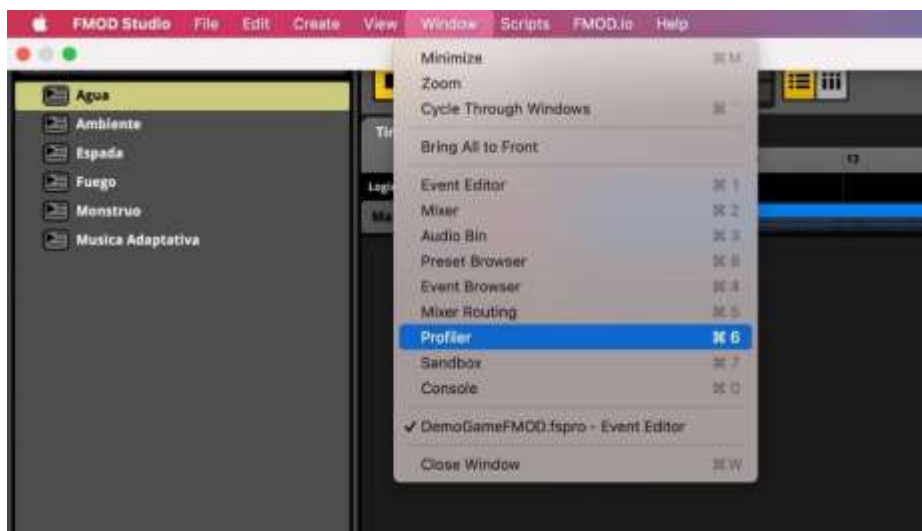
Nota: El gráfico muestra en líneas de colores la activación de Sincronización de live update.

Imagen propia.

Esta opción nos permite sincronizar en tiempo real el middleware con el motor de videojuegos Unity en tiempo real, así mismo nos permite ver en Fmod en la Opción Profiler el recurso consumido durante el juego en los momentos de uso del sonido y tomar capturas de esto para tomar mejores decisiones en cuanto a la optimización y ecualización y mezcla del audio.

Figura 28

Ventana de búsqueda para la opción profiler



Nota: El gráfico muestra la ventana de acceso al Profiler, para generar una imagen sonora en Fmod studio. *Imagen propia.*

Figura 29

Creación de nueva sesión de profiler

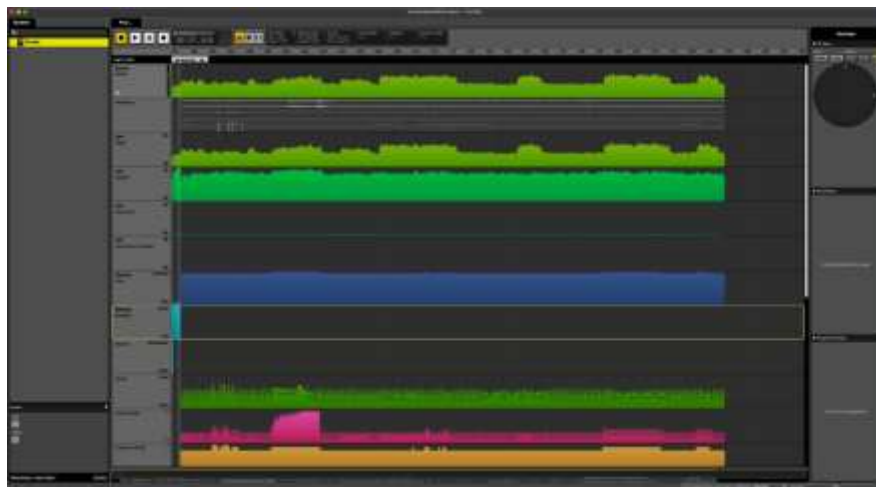


Nota: El gráfico muestra la creación de una nueva sesión para las imágenes o perfiles del consumo de recursos por el diseño sonoro en Fmod Studio. *Imagen propia.*

Una vez creada la sesión del perfil que tomaremos procedemos a grabar mientras jugamos con el motor de videojuego la escena que hemos programado, pasando por todas las áreas de sonido, y estas quedarán grabadas en la sesión en la que nos encontramos grabando con una visual como la imagen siguiente.

Figura 30

Imagen de la muestra de perfil grabada en el Profiler de Fmod



Nota: El gráfico muestra la imagen del perfil del diseño sonoro. Imagen propia.

En nuestro caso, se realizaron dos muestras, una antes y otra después de la optimización y mezcla, luego con las opciones siguiente en la imagen podemos ver la tabla de recursos consumidos durante el proceso de uso de sonido del videojuego.

Figura 31

Opción de búsqueda de recursos



Nota: El gráfico muestra en líneas de colores la forma de acceder a la información generada por el perfil tomado en Fmod Studio. *Imagen propia.*

Figura 32

Tabla de recursos del sistema Fmod

Nombre	CPU Mixer (Self)	CPU Mixer (Total)	CPU Update (Self)	CPU Update (Total)	CPU Streaming (Total)	CPU Calculation Threads (1)	CPU Calculation Threads (2)	Levels (RMS)	Levels (Peak)
System	1.00%	1.00%	1.70%	1.70%	0.00%	0.00%	0.00%	-22.47 dB	-16.44 dB
Bank	0.00%	0.00%	0.00%	0.00%	-	-	-	-80.00 dB	-80.00 dB
Musica	0.01%	0.04%	0.00%	0.01%	-	-	-	-22.48 dB	-16.44 dB
Musica2	0.00%	0.00%	0.00%	0.00%	-	-	-	-80.00 dB	-80.00 dB
Master Bus	0.02%	0.14%	0.00%	0.02%	-	-	-	-22.47 dB	-16.44 dB
Efectos	0.02%	0.06%	0.00%	0.02%	-	-	-	-75.26 dB	-43.44 dB
Ambiencs	0.00%	0.00%	0.00%	0.00%	-	-	-	-80.00 dB	-80.00 dB
Reverberación	-	-	-	-	-	-	-	-	-
Musica Adaptativa	0.00%	0.00%	0.00%	0.00%	-	-	-	-22.48 dB	-16.44 dB
Muestras	0.00%	0.00%	0.00%	0.00%	-	-	-	-80.00 dB	-80.00 dB
Fuente	0.04%	0.04%	0.00%	0.04%	-	-	-	-75.34 dB	-43.44 dB
Esada	0.00%	0.00%	0.02%	0.02%	-	-	-	-80.00 dB	-80.00 dB
Ambiencs	0.00%	0.00%	0.00%	0.00%	-	-	-	-80.00 dB	-80.00 dB
Agua	0.00%	0.00%	0.00%	0.00%	-	-	-	-80.00 dB	-80.00 dB

Nota: El gráfico muestra la tabla de recursos consumidos por el diseño sonoro. *Imagen propia.*

Y es así como termina este proceso que hemos mostrado a modo de tutorial para que pueda servir en algún momento como ayuda a las personas que se enfrenten a este exigente, pero divertido Diplomados en Audio Dinámicos para Videojuegos. Esperamos sea de utilidad este informe para el futuro de nuestros compañeros y colegas.

Anexos

Anexo A

Carpetas de Practica Académica

<https://drive.google.com/drive/folders/1dO8ZR97iuUaS7HBA4sXPt1VVqXnsok0j?usp=sharing>

https://drive.google.com/drive/folders/1dO8ZR97iuUaS7HBA4sXPt1VVqXnsok0j?usp=drive_link

Conclusiones

El reconocimiento y adaptación a los middleware para la creación de audio y permitir que estos sean una experiencia cada vez más envolvente para los jugadores se dio mediante la presentación y uso del motor de videojuegos Unity y el middleware Fmod Studio, basando en sus usos y complejidades las cuales se adentran en un mundo bastante complejo que cumple con todas las exigencias del mercado y la creación

Las necesidades de aprender a controlar estos softwares permiten al tecnólogo en producción de audio ser una ficha clave en la creación y hacer parte de los equipos de trabajo más productivos por el conocimiento adquirido permitiendo superar grandes retos de la industria.

Para esto el aprendizaje debe ser continuo y de la mano de las tutorías y los formadores, participar y consultar por lo que no es solo un trabajo de lo básico, es fundamental adentrarse y ser un empedernido con el manejo de estos motores para una creación más profesional cada vez.

Ahora no basta solo con realizar este proceso ni no también seguir adelante con la capacitación o profundización de los temas y habilidades adquiridas en este diplomado.

Bibliografía

- Brandon, A. (2006). *Game Audio Integration*. Mix, 30(3),
56. <https://bibliotecavirtual.unad.edu.co/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=bsu&AN=20042238&lang=es&site=eds-live&scope=site>
- Eraso, C. (2022). *Audio dinámico en los videojuegos*. [Objeto_virtual_de_aprendizaje_OVA]. Repositorio Institucional UNAD. <https://repository.unad.edu.co/handle/10596/53360>
- Fammà, A. (2021). *Implementing Footsteps with FMOD in Unity*.
<https://alessandrofama.com/tutorials/fmod-unity/footsteps/>
- 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>
- FMOD (2021). *Studio User Manual 2.01*. [Manual de usuario].
<https://fmod.com/resources/documentation-studio?version=2.1&page=mixing.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>
- 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

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)

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)

Lanham, M. (2017). [*Game Audio Development with Unity 5.X*](#). Packt Publishing, 205-
221. [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_205](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_205)

Lanham, M. (2017). [*Game Audio Development with Unity 5.X*](#). Packt Publishing, 230-
237. [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_230](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_230)

Lanham, M. (2017). [*Game Audio Development with Unity 5.X*](#). Packt Publishing, 365-
373. [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_365](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_365)

px?direct=true&db=e000xww&AN=1547031&lang=es&site=eds-live&scope=site&ebv=EB&ppid=pp_365

Martínez, I. (2021). [Música y sonido para videojuegos](https://press-music.com/musica-y-sonido-para-videojuegos/). Press Music. <https://press-music.com/musica-y-sonido-para-videojuegos/>

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 Introducción](http://hdl.handle.net/10596/11795). [Archivo de video]. <http://hdl.handle.net/10596/11795>

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 Programmer, command, snapshot and Plug-in Instruments](http://hdl.handle.net/10596/11777). [Archivo de video]. <http://hdl.handle.net/10596/11777>

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 Single Instrument](http://hdl.handle.net/10596/11789). [Archivo de video]. <http://hdl.handle.net/10596/11789>

Moreno Viasus, R. (15,02,2017). [Audio Middlewares y Herramientas de Diseño Sonoro para Videojuegos](http://hdl.handle.net/10596/11604). [Archivo de video]. <http://hdl.handle.net/10596/11604>

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

Moreno, R. (2021). [Tutorial de Integración de Fmod con una escena de Unity](https://ramorenov.github.io/gameaudio/01_IntegracionfmodUnity.html). Game Audio.

https://ramorenov.github.io/gameaudio/01_IntegracionfmodUnity.html

Moreno, R. (2021). [Tutorial efectos sonoros y pasos de personaje en Fmod para una escena de Unity](https://ramorenov.github.io/gameaudio/05_Fmod_efectos_sonoros_y_pasos_de_personaje.html). Game Audio.

https://ramorenov.github.io/gameaudio/05_Fmod_efectos_sonoros_y_pasos_de_personaje.html

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

Moreno, R. (2021). [Tutorial música adaptativa en FMOD Studio](https://ramorenov.github.io/gameaudio/FMOD/02_Fmod_musica_adaptativa.html). Game Audio.

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

Moreno, R. (2021). [Tutorial música en Fmod para una escena de Unity](https://ramorenov.github.io/gameaudio/06_Fmod_musica.html). Game Audio.

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

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>

Rodríguez, J. A. (2023). [*Música para medios interactivos*](#).

[Objeto_virtual_de_aprendizaje_OVA]. Repositorio Institucional UNAD.

<https://repository.unad.edu.co/handle/10596/56066>

Sello editorial, Beltrán Bríñez, J. A., Mejía Cuevas, M., & Torres, M. (2014). *Análisis de parámetros objetivos y subjetivos en Pre-Amplificadores de Audio*. Publicaciones E

Investigación, 8(1), 13-24. <https://doi.org/10.22490/25394088.1276>

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>

Winifred Phillips. (2014). [*A Composer's Guide to Game Music*](#). The MIT

Press. [https://bibliotecavirtual.unad.edu.co/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=e000xww&AN=699337&lang=es&site=eds-](https://bibliotecavirtual.unad.edu.co/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=e000xww&AN=699337&lang=es&site=eds-live&scope=site&ebv=EB&ppid=pp_185)

[live&scope=site&ebv=EB&ppid=pp_185](https://bibliotecavirtual.unad.edu.co/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=e000xww&AN=699337&lang=es&site=eds-live&scope=site&ebv=EB&ppid=pp_185)

[live&scope=site&ebv=EB&ppid=pp_185](https://bibliotecavirtual.unad.edu.co/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=e000xww&AN=699337&lang=es&site=eds-live&scope=site&ebv=EB&ppid=pp_185)