

RAE

1. TIPO DE DOCUMENTO: Trabajo de grado para optar por el título de INGENIERO DE SONIDO
2. TITULO: Desarrollo de prototipo de audio juego, aplicado a una instalación interactiva sonora con sistema de reproducción ambisonics
3. AUTORES: Alvaro Fernando Lagos Sanchez y Simón Quintero Rodriguez
4. LUGAR: Bogotá, D.C
5. FECHA: diciembre 2016
6. PALABRAS CLAVE: Audiojuego, instalación interactiva, Ambisonics, Kinect, Unity, FMOD
7. DESCRIPCIÓN DEL TRABAJO: El objetivo principal de este trabajo es desarrollar un prototipo de audiojuego para un sistema de reproducción ambisonics en el cual la interactividad del jugador se dé por medio de movimientos corporales sin necesidad de tener ninguna interfaz gráfica. En este trabajo se podrá ver todo el desarrollo del prototipo, en donde se contemplan los aspectos básicos desde la creación de guion técnico y literario, las herramientas necesarias para la interacción del jugador con la máquina hasta el sistema y configuración para la reproducción por medio del sistema envolvente Ambisonics.
8. LÍNEAS DE INVESTIGACIÓN: Tecnologías actuales y sociedad, Análisis y procesamiento de señales, Acústica y audio
9. METODOLOGÍA: Esta investigación posee un enfoque cualitativo.
10. CONCLUSIONES: El audiojuego presenta una eficacia en su trabajo del 80% debido a que alguna de las capturas corporales presenta dificultad en su reconocimiento. Con base en la población encuestada, el audiojuego es viable siempre y cuando la inteligibilidad de la palabra en la transmisión del mensaje sea buena, obteniendo así, una buena interpretación y entendimiento en el desarrollo del audiojuego en su estado gameplay. Además, se puede apreciar que el audiojuego es aceptado por los usuarios gracias a su canal de interactividad establecida por el arreglo de sonido envolvente y por su sistema de comunicación mediante movimientos corporales

**DESARROLLO DE PROTOTIPO DE AUDIOJUEGO, APLICADO A UNA  
INSTALACIÓN INTERACTIVA SONORA CON SISTEMA DE REPRODUCCIÓN  
AMBISONICS**

**ALVARO FERNANDO LAGOS SANCHEZ  
SIMON QUINTERO RODRIGUEZ**

**UNIVERSIDAD DE SAN BUENAVENTURA BOGOTÁ  
FACULTAD DE INGENIERIA  
INGENIERIA DE SONIDO  
BOGOTÁ D.C  
2016**

**DESARROLLO DE PROTOTIPO DE AUDIOJUEGO, APLICADO A UNA  
INSTALACIÓN INTERACTIVA SONORA CON SISTEMA DE REPRODUCCIÓN  
AMBISONICS**

**ALVARO FERNANDO LAGOS SANCHEZ  
SIMON QUINTERO RODRIGUEZ**

**PROYECTO DE GRADO**

**TUTOR:  
ING. MIGUEL OLIVARES**

**UNIVERSIDAD DE SAN BUENAVENTURA BOGOTÁ  
FACULTAD DE INGENIERIA  
INGENIERIA DE SONIDO  
BOGOTÁ D.C  
2016**

**Nota de aceptación**

**Firma del presente jurado**

**Firma jurado.**

**Firma jurado.**

## **AGRADECIMIENTOS Y DEDICATORIA**

Agradecemos de todo corazón a María Paula Ceballos, Alejandra Campo Martínez, Diana Martínez Muñoz, Valentina Vera, Laura Peña, Johnny Mejía, Laura Barrios y Miguel Olivares por su ayuda y constante cooperación en el desarrollo de este trabajo de grado. Sus aportes y constante ayuda fueron de vital importancia para seguir adelante y no desfallecer en la culminación de este proyecto de la mejor manera posible.

Este trabajo es dedicado a: Jorge Armando Lagos, Sandra Sanchez Torres, Santiago Lagos Sanchez, Alvaro Sanchez, María Torres, Rosalba Rodríguez, Carlos Quintero, Daniel Quintero, Juan David Quintero por creer en nosotros a brindarnos su apoyo incondicional, motivación y paciencia durante este que corto pero especial trayecto de nuestras vidas.

## TABLA DE CONTENIDO

1. INTRODUCCIÓN.....	15
2. PLANTEAMIENTO DEL PROBLEMA.....	16
2.1. Antecedentes .....	16
2.1.1. Estudios hechos en Colombia .....	16
2.1.2. Estudios internacionales.....	16
2.2. Descripción del problema.....	19
2.3. Formulación del problema .....	20
2.4. Justificación.....	20
3. OBJETIVOS.....	21
3.1. Objetivo general .....	21
3.2. Objetivos específicos .....	21
4. ALCANCES Y LIMITACIONES .....	22
4.1. Alcances.....	22
4.2. Limitaciones .....	22
5. MARCO METODOLOGICO.....	23
5.1. Tipo de investigación.....	23
5.2. Línea de investigación.....	23
5.3. Hipótesis .....	24
5.4. Variables .....	24
5.4.1. Variables dependientes .....	24
5.4.2. Variables independientes.....	24
6. MARCO REFERENCIAL .....	25
6.1. Marco conceptual .....	25
6.1.1. Audiojuego.....	25
6.1.2. Interacción hombre maquina .....	25
6.1.3. Motor de videojuegos .....	25
6.1.4. Interfaz de control .....	25
6.1.5. Técnicas de micrófonia.....	26
6.1.6. Técnicas de reproducción.....	26

6.1.7.	Plugin.....	26
6.1.8.	Diseño sonoro.....	27
6.2.	Marco Teórico .....	27
6.2.1.	Técnicas de grabación y reproducción .....	27
6.2.2.	Motor para el desarrollo de videojuegos .....	31
6.2.3.	Plugin.....	38
6.2.4.	Interfaz de control .....	39
6.3.	Marco normativo.....	40
6.3.1.	ESBR: Entertainment Software Rating Board.....	40
6.3.2.	Norma EBU R-128.....	41
7.	METODOLOGIA DEL DISEÑO .....	43
7.1.	Desarrollo ingenieril.....	43
7.1.1.	Sinopsis del audiojuego.....	43
7.1.2.	Niveles del audiojuego.....	43
7.1.3.	Clasificación del audiojuego .....	44
7.1.4.	Instalación interactiva y hardware del audiojuego .....	45
7.1.5.	Diseño sonoro y grabaciones de audio.....	47
7.1.6.	Acciones corporales .....	64
7.1.7.	Diseño del software .....	69
7.1.8.	Integracion Fmod-Unity .....	87
7.2.	Análisis ingenieril.....	102
8.	CONCLUSIONES .....	114
9.	RECOMENDACIONES.....	116
10.	REFERENCIAS.....	117
11.	BIBLIOGRAFIA.....	118

## LISTADO DE FIGURAS

Figura 1. Sistema Cuadrafónico Ambisonic. ....	28
Figura 2. Técnica de micrófonia Niimbus Halliday. ....	29
Figura 3. Técnica de micrófonia AB. ....	29
Figura 4. Técnica de micrófonia XY. ....	30
Figura 5. Fuente de audio y escucha. ....	31
Figura 6. Diagrama de flujo proyecto básico de FMOD Studio. ....	32
Figura 7. Evento en FMOD Studio. ....	33
Figura 8. Evento simple en FMOD Studio. ....	33
Figura 9. Evento Multitrack en FMOD Studio. ....	34
Figura 10. Archivo Multisound en FMOD Studio. ....	34
Figura 11. Uso de marcadores en un evento de FMOD Studio. ....	35
Figura 12. Tipos de Transition en FMOD Studio. ....	35
Figura 13. LoopRegion en FMOD Studio. ....	36
Figura 14. Barra de transporte FMOD Studio. ....	36
Figura 15. Valor numérico máximo y mínimo de un parámetro FMOD Studio. ....	36
Figura 16. Parámetro Ejemplo en FMOD Studio. ....	37
Figura 17. Evento Parametro Ejemplo en FMOD Studio. ....	37
Figura 18. Paneo y canales implementados dentro de un evento en FMOD Studio. ....	38
Figura 19. Oculus Audio configurado para una salida ambisonics de cuatro canales. ....	38
Figura 20. Características Kinect. ....	40
Figura 21. Reconocimiento corporal Kinect. ....	40
Figura 22. Logotipo de clasificación Everyone +10. ....	44
Figura 23. Plano de la instalación interactiva y su distribución. ....	46
Figura 24. Instalación interactiva ubicada en el estudio hibrido de la Universidad San Buenaventura Bogotá. ....	47
Figura 25. Parte inferior de la instalación interactiva. ....	47



Figura 26. Arreglo de micrófonos técnica Nimbus Halliday.....	48
Figura 27. Grabación de voz personaje Yacu .....	49
Figura 28. Grabación de voz personaje Ekko y Dante.....	49
Figura 29. Grabación splash de agua.....	50
Figura 30. Grabación de ambiente acuático .....	51
Figura 31. Grabación de estanque Universidad San Buenaventura con la técnica de microfoneo estero AB.....	51
Figura 32. Grabación de ruido blanco en un recinto reverberante .....	52
Figura 33. Grabación de ambientes en exterior .....	52
Figura 34. Ensamble en Reaktor correspondiente al efecto Fx_000 .....	53
Figura 35. Grabación de bongos y Palo de lluvia.....	54
Figura 36. Grabación de cadenas y plato de bicicleta para la puerta del castillo...54	
Figura 37. Grabación de motor de auto .....	55
Figura 38. Grabación de burbujas provocadas mediante el uso de licuadora y pitillo .....	55
Figura 39. Colisión de pecera de vidrio dentro de un espacio con reverberación alta .....	56
Figura 40. Grabación de objetos de vidrio colisionando contra el suelo .....	57
Figura 41. Grabación de colisión de tornillos y objetos de metal .....	57
Figura 42. Grabación de movimientos de agua .....	58
Figura 43. Grabación de objetos, rejas, y cuerdas de metal.....	58
Figura 44. Pasos en agua dentro de una ducha .....	59
Figura 45. Grabación en recipiente con agua .....	59
Figura 46. Grabación de Cadenas en ambiente 1 .....	60
Figura 47. Grabación de pasos en el Foley pit de ladrillo .....	60
Figura 48. Grabación de wosh mediante cuerdas de lana y de caucho.....	61
Figura 49. Grabación de splash con periódico y trapos mojados.....	61
Figura 50. Grabación de pasos de perro .....	62
Figura 51. Grabación de efecto de serpiente mediante palo de lluvia .....	62
Figura 52. Movimiento de hojas y rasgadura de tronco .....	63

Figura 53. Grabación de cinta magnética y láminas de madera .....	63
Figura 54. Grabación golpes con varas de madera y cinta de caucho .....	64
Figura 55. Parámetros Cubeman.....	65
Figura 56. Posición Tpose .....	65
Figura 57. Posición PSI.....	66
Figura 58. Posición LeftRighHand - RaiseRighHand .....	66
Figura 59. Posición Push vista frontal y vista lateral .....	67
Figura 60. Posición SwipeLeft.....	67
Figura 61. Posición SwipeRight .....	67
Figura 62. Posición SwipeLeft.....	68
Figura 63. Posición Jump.....	68
Figura 64. Posición StepRight en vista frontal y lateral .....	68
Figura 65. Plugin Ovr Goble Settings.....	70
Figura 66. Studio Listener .....	71
Figura 67. Parámetros Kinect Manager .....	71
Figura 68. Complementos CubeMan .....	72
Figura 69. Componente CubeMan Controller .....	72
Figura 70. Representación de cubos y líneas del usuario.....	73
Figura 71. Componente Gesture Listener .....	73
Figura 72. Avatar Kinect.....	73
Figura 73. Script Gesture Listener .....	74
Figura 74. Código detección de gestos.....	75
Figura 75. Función SetGestureJoint .....	75
Figura 76. Materiales creados en Unity .....	76
Figura 77. Simulación de ambiente en Unity.....	77
Figura 78. Recreación de espacio en Unity con ambientes ubicados.....	77
Figura 79. Objetos Nivel Entrenamiento .....	78
Figura 80. Script instrucciones en el Nivel Entrenamiento .....	78
Figura 81. Diagrama de secuencia Nivel de entrenamiento.....	80
Figura 82. Objetos escenario Introducción.....	81

Figura 83. Script introducción para el nivel de introducción.....	81
Figura 84. Diagrama de secuencia del nivel de introducción.....	82
Figura 85. Objetos Nivel Yacu.....	83
Figura 86. Script Yacu en el nivel acuático de Yacu.....	83
Figura 87. Diagrama de secuencia del nivel Yacu.....	85
Figura 88. Objetos Nivel Dante.....	86
Figura 89. Script Dante en el Nivel selvático de Dante.....	86
Figura 90. Integracion FMOD-Unity.....	87
Figura 91. Elementos Fmod Package.....	89
Figura 92. Asignación de eventos.....	90
Figura 93. Importar Assets.....	90
Figura 94. Diagrama de flujo Evento5.....	92
Figura 95. Interfaz Evento5 en FMOD Studio.....	93
Figura 96. Interfaz Evento1 con audios Amb_6 y Fx_54.....	94
Figura 97. Diagrama de flujo Evento19.....	95
Figura 98. Interfaz Evento19 en FMOD Studio.....	95
Figura 99. Interfaz Evento23 en FMOD Studio.....	96
Figura 100. Diagrama de flujo Evento41.....	97
Figura 101. Interfaz Evento41 en FMOD Studio.....	98
Figura 102. Diagrama de flujo Evento13.....	98
Figura 103. Interfaz Evento13 en FMOD Studio.....	99
Figura 104. Diagrama de flujo Evento14 y Evento 15.....	99
Figura 105. Interfaz Evento14 Propulsor en FMOD Studio.....	100
Figura 106. Interfaz Evento15 Orca en FMOD Studio.....	100
Figura 107. Diagrama de flujo Evento25.....	101
Figura 108. Interfaz Evento25 FMOD Studio.....	102
Figura 109. Jugador interactuando con el audiojuego.....	103
Figura 110. Jugador interactuando con el audiojuego.....	104
Figura 111. Resultados pregunta 1.....	107
Figura 112. Resultados pregunta 2.....	108

Figura 113. Resultados pregunta 3.....	108
Figura 114. Resultados pregunta 4.....	109
Figura 115. Resultados pregunta 5.....	109
Figura 116. Resultados pregunta 6.....	110
Figura 117. Resultados pregunta 7.....	110
Figura 118. Resultados pregunta 8.....	111
Figura 119. Resultados pregunta 9.....	111
Figura 120. Resultados pregunta 10.....	112
Figura 121. Resultados pregunta 11.....	112
Figura 122. Resultados pregunta 12.....	113

## LISTADO DE TABLAS

Tabla 1. Categoría de eventos en FMOD Studio .....	91
Tabla 2. Eventos de narración .....	91
Tabla 3. Eventos de ambiente 1 .....	94
Tabla 4. Eventos de ambiente 2 .....	94
Tabla 5. Eventos naturales .....	96
Tabla 6. Eventos compuestos en FMOD Studio .....	96
Tabla 7. Equipos y elementos usados en la instalación interactiva .....	103

## RESUMEN

En este proyecto de grado, se desarrolla y diseña un prototipo de audiojuego de tipo aventura, el cual será implementado dentro de una instalación interactiva conformada por un sistema de reproducción de sonido envolvente llamado *ambisonics*. Aquí, se dará a conocer el proceso que se llevó a cabo para el desarrollo del prototipo, evidenciando el trabajo que se hizo en áreas como: creación del guion literario, guion técnico, grabación de voces y efectos, programación del audiojuego, interacción de la maquina con el usuario y a integración con el middleware encargado de la reproducción por un sistema ambisonics.

## 1. INTRODUCCIÓN

La industria de los juegos en la actualidad está en auge, debido “a los avances tecnológicos del siglo XXI, la revolución digital y el desarrollo de las herramientas y programas informáticos que permiten al artista explorar un nuevo campo para el desarrollo de una actividad en específica” (Saura, 2010). El objetivo de un juego se fundamenta en alcanzar una meta, respetando ciertas reglas. La interacción que se realiza en este, se puede llevar a cabo de muchas maneras, esto depende del tipo el juego. Pero en este caso, se desarrollará un juego en el que la percepción sonora será el sentido fundamental por el cual, el usuario obtendrá información ya sea de un objeto, un lugar o un personaje con el cual tenga que interactuar.

En este proyecto se desarrollará el prototipo de un audio juego para una instalación interactiva con sistema de reproducción Ambisonics, en donde se hará uso del reconocimiento de movimientos corporales por medio del Kinect para la interacción del usuario con el juego, además de esto se contará un con sistema de reproducción envolvente Ambisonics por el cual el usuario obtendrá retroalimentación sonora, que lo ira guiando en los niveles establecidos. Para que esto tenga un buen funcionamiento se debe tener en cuenta un buen reconocimiento corporal y un buen diseño sonoro en los diálogos, Foleys, fx y ambientes.

## 2. PLANETAMIENTO DEL PROBLEMA

### 2.1. Antecedentes

#### 2.1.1. Estudios hechos en Colombia

- Luisa Fernanda Parra Espinel., En el año 2014 presenta su proyecto de grado “Desarrollo de un video juego con reconocimiento de voz para la identificación de colores y figuras geométricas”, en el cual desarrolla un viejo juego educativo que pretende reforzar el conocimiento por medio de la atención y repetición de un ejercicio a través de sistema no convencional de interacción, comandos de voz. Universidad de San Buenaventura.
- Jhosimar Andrés Aguacía Fiscó., Diego Armando Pardo Sabogal., Carlos Andrés Lozano Sarmiento., Presentan en el año 2004 su proyecto de grado en el cual pretenden recrear atmosferas sonoras por medio de sistemas Ambisonics, en donde realizan una producción audiovisual de video y audio. Implementan un arreglo de micrófonos que permiten simular señales en el B-Format de Ambisonics y también capturan fuentes puntuales que codifican en sistemas de tercer orden para su correcta reproducción en un sistema Ambisonics. Universidad San Buenaventura

#### 2.1.2. Estudios internacionales

- Mathias Kronlachner., Franz Zotter., Escriben un paper en el 2014 nombrado “Spatial transformations for the enhancement of Ambisonic recordings” en el que hablan de manera detallada sobre el funcionamiento de un sistema Ambisonics, sus formas de captura y reproducción además de las ventajas y flexibilidades que presentan los sistemas de reproducción Ambisonics.
- Mathias Kronlachner., En el año 2014 presenta su tesis de maestría titulada “Spatial Transformations for the Alteration of Ambisonic Recordings” en el



que habla sobre la producción y reproducción de escenas sonoras en un sistema Ambisonics, la ubicación de los altavoces para delimitar el área de escucha, y sus debidos procesos de decodificación para un buen resultado en la perspectiva sonora. Universidad de música y artes escénicas de Graz, Austria.

- Ásén Rickard, Publica en el 2013 “Towards a Theory on the roles and Functions of Sound in Audio Games”, La carencia de convenciones teóricas importantes que hay sobre audio juegos en comparación con el campo sonoro de los video juegos, es supremamente alta, ya que aún no se ha explorado el papel del sonido al actuar de una forma narrativa autónoma en un medio virtual interactivo , es por eso que en esta publicación se aborda la producción de audio juegos, basado en la experiencia sensitiva del oído y las condiciones que son necesarias tener en cuenta a la hora de diseñar un audio juego.
- Pablo Caloto Crespo., Manuel Moranchel Edras., Angel Ruiz Alonzo., En el Año 2009 para su proyecto de grado “Tecnologías de reconocimiento de voz y su aplicabilidad en video juegos” realizaron y presentaron una investigación, sobre los modelos acústicos de diferentes idiomas, con el objetivo de manejar un avatar por medio de los comandos de voz, descartando interfaces físicas como el teclado o ratón convencionales. Universidad Complutense de Madrid, España.
- Harris Frederick, Folmer Eelke, Realizan la publicación de su tesis de grado en la universidad de Nevada en el año 2009, titulada “Towards Generalized Accessibility of Video Games for the Visually Impaired”, en donde se hace un énfasis especial hacia los videojuegos más que una forma de entretenimiento, como una herramienta de educación gracias a las herramientas de interacción disponibles hoy en día. Se centralizan en el

diseño y desarrollo de estrategias de accesibilidad para los deficientes visuales ciegos.

- Zhihong Zeng., Jilin Tu., Brian Pianfetti., Ming Liu., Tong Zhang., Zhenqiu Zhang., Thomas S. Huang and Stephen Levinson., En el 2005 realizaron un artículo titulado “Bimodal HCI-related Affected Recognition” que trata acerca la interacción hombre máquina, en donde se habla sobre el desarrollo de un algoritmo para sensores de audio y de gestos que detecta el estado anímico de una persona. Universidad de Illinois, Urbana Champaign, EEUU.

## **2.2. Descripción del problema**

La búsqueda constante por logra satisfacer las necesidades del ser humano ha impulsado el avance en el desarrollo de nuevas tecnologías, dichas tecnologías se han aprovechado en mayor medida al diseño de herramientas y/o productos en diferentes campos como, por ejemplo, en el campo de los medios audiovisuales, lo cual actúa como medio de comunicación que busca transmitir mensajes dentro de una población a través de imagen y sonido.

Dichas tecnologías permitieron desarrollar la forma de transmitir mensajes en los medios audiovisuales a través de fotografías, imágenes impresas, publicidad, radiodifusión, etc. Pero este sistema está renovando gracias a la incorporación de herramientas digitales, como por ejemplo el computador, lo cual logra la creación de interfaces graficas (video, mapping, etc.) y sonoras (DAW) que dan como resultado una mayor interacción entre las personas a la hora de transmitir las ideas.

Uno de los aplicativos más trascendentes de los medios audiovisuales son los videojuegos, cuyo propósito es divertir, entretener, educa, etc. Al introducir de manera inmersa e interactiva al usuario en un mundo virtual y/o irreal, en donde, generalmente se debe seguir unas reglas establecidas para cumplir un objetivo y obtener una recompensa por esto.

La interacción entre el usuario y la maquina se da por diversos medios de entrada y salida en audio, imagen, comunicación, etc. lo cual da un valor agregado a una mejor inmersión, en especial en su campo grafico ya que su jugabilidad depende más de las acciones que se visualizan que de las acciones que se escuchan en el videojuego.

### **2.3. Formulación del problema**

Partiendo de lo anterior se plantea el siguiente interrogante:

¿De qué manera se puede desarrollar un audio juego, en la cual su jugabilidad solo este dada por una interfaz sonora?

### **2.4. Justificación**

Debido a que se quiere obtener un juego interactivo que dependa únicamente de una interfaz sonora, se plantea un prototipo de audiojuego en el cual, el sistema de interacción hombre-máquina será dada mediante estímulos netamente sonoros por parte del audiojuego y movimientos corporales por parte del jugador. Teniendo en cuenta que este audiojuego será implementado a una instalación interactiva, se busca obtener una buena recopilación de datos sobre la reacción del jugador hacia el audiojuego, esto se llevará a cabo mediante la integración de controladores de reconocimiento de gestos como el Kinect; Además, también se busca obtener una buena retroalimentación por parte del audiojuego. Para lograr la retroalimentación deseada, el audio juego será reproducido a través de un arreglo dedicado a sonido envolvente, en este caso sistema de reproducción ambisonic. Esto brindará al jugador una experiencia interactiva en donde tendrá que poner a prueba su destreza auditiva y su agilidad corporal, incorporando métodos no convencionales de sonido envolvente y sin depender de la parte gráfica.

### **3. OBJETIVOS**

#### **3.1. Objetivo general**

- Desarrollar el prototipo de un audio juego para una instalación interactiva con un sistema de reproducción Ambisonics, por medio de diferentes softwares de programación, en el que su jugabilidad se lleve a cabo por movimientos corporales.

#### **3.2. Objetivos específicos**

- Establecer la distribución del hardware para la instalación interactiva.
- Implementar sistemas de captura de sonido para la reproducción en Ambisonics
- Determinar y limitar la cantidad de acciones corporales que el usuario tendrá para interactuar con el audio juego.
- Diseñar un software para la recopilación de datos provenientes de la interacción corporal del usuario con la máquina.
- Compatibilizar los algoritmos del audiojuego y el proceso de codificación de señales.

## **4. ALCANCES Y LIMITACIONES**

### **4.1. Alcances**

- El prototipo de audio juego proporcionara una inmersión sonora de 360° grados, lo cual brindara un mayor realismo al jugador
- Unity y FMOD serán las plataformas principales donde se desarrollará el prototipo de audio juego
- Desarrollar un sistema de comunicación por el cual el usuario controlara el audio juego por medio interfaz de control kinect

### **4.2. Limitaciones**

- Espacio arquitectónico de la instalación sonora donde será implementado el prototipo audiojuego
- Debido al tipo de jugabilidad del audio juego, solo se permite el ingreso de un jugador por vez dentro de la instalación.
- El audio juego será controlado únicamente por movimientos de brazos y pies
- Solo se permiten jugadores con edad mínima de 14 años en adelante.

## **5. MARCO METODOLÓGICO**

### **5.1. Tipo de investigación**

Esta investigación posee un enfoque cualitativo, teniendo en cuenta que se basa en el desarrollo de un sistema que brinda una forma diferente de interacción con juegos digitales, basado en el uso del sentido de la audición humana; donde a su vez se pretende demostrar las cualidades que ofrece un audio juego en su jugabilidad.

### **5.2. Línea de investigación**

Tecnologías actuales y sociedad: El prototipo de audio juego controlado por medio de movimientos corporales es un producto el cual demuestra el desarrollo de la tecnología y el impacto positivo que puede llegar a tener este tipo de herramientas

Análisis y procesamiento de señales: La forma que tiene el usuario para interactuar con el audio juego es por medio de sus movimientos. Se hará un análisis de cada recopilación de datos MIDI, proveniente de la captura de los movimientos. Con esto nace la necesidad de desarrollar una plataforma que permita el procesamiento de estos datos que su vez permitirán en tiempo real procesar una señal sonora que será reproducida por medio del sistema de reproducción multicanal Ambisonics.

Acústica y audio: Se tendrá en cuenta los diferentes tipos de efectos y procesos aplicados a los audios que se reproducirán en el sistema, tanto digital como acústicamente, al igual que el correcto funcionamiento de envío y recepción de mensajes MIDI dentro de la plataforma.

### **5.3. Hipótesis**

Realizar un juego enfocado en la parte sonora y no en la parte grafica para una instalación interactiva con sistema de reproducción Ambisonics, es viable, siempre y cuando se aprovechen las características que tienen los sonidos, como el timbre, su amplitud y su espacialidad. Logrando captar la atención del usuario gracias al sistema de comunicación y retroalimentación implementado, teniendo como resultado una buena experiencia con el audiojuego.

### **5.4. Variables**

#### ***5.4.1. Variables dependientes***

- Guion del audiojuego
- Diseño sonoro del audiojuego
- Tamaño de hardware
- Ubicación de hardware
- Algoritmo

#### ***5.4.2. Variables independientes***

- Estado de los equipos de captura
- Entorno de programación
- Recinto de la instalación interactiva



## **6. MARCO REFERENCIAL**

### **6.1. Marco conceptual**

#### ***6.1.1. Audiojuego***

Es un juego que requiere un dispositivo electrónico para poder ser jugado (Por ejemplo, un computador o una consola de juegos) en donde el sistema de retroalimentación consiste principalmente o únicamente en el sonido, en donde la historia y la acción se desarrollan netamente de manera narrativa para aprovechar los potenciales y posibilidades que brinda el sonido. (AudioGame, s.f.)

#### ***6.1.2. Interacción hombre maquina***

Son sistemas informáticos interactivos o interfaces de usuario que tienen como fin ser usados por humanos para manejar herramientas en un sistema digital con mejor precisión, alcance, movilidad, comodidad etc. En otras palabras, es un medio de comunicación o relación que hay entre un ser humano y una máquina. Las características de la interfaz de usuario varían según las necesidades que el programa requiera. (Cueva, s.f)

#### ***6.1.3. Motor de videojuegos***

Es un software que está especializado en la creación, diseño y representación de juegos, por medio de un conjunto de rutinas de programación que proveen gráficos, sonidos, detección de colisiones, animación, inteligencia artificial, entre muchos otros que pueden ser usados o no, según el tipo y genero del juego que se desarrolle.

#### ***6.1.4. Interfaz de control***

Es el medio por el cual un usuario se comunica y relaciona con un dispositivo electrónico. La interacción se puede llevar por medio de:

- Interfaz de líneas de control: Método que permite al usuario ingresar instrucciones por medio de líneas de texto
- Interfaz gráfica de usuario: Programa gráfico que usa imágenes y objetos para representar información y acciones disponibles en la interfaz con las que el usuario puede interactuar a través de un dispositivo físico.
- Interfaz natural de usuario: Es el tipo de interfaz que permite relacionar a un usuario con un sistema o aplicación, sin necesidad de usar mandos de entrada, en su lugar se hace uso de movimientos corporales para la interacción del hombre con la máquina.

#### **6.1.5. Técnicas de microfonía**

Las técnicas de microfonía son la composición de dos o más micrófonos ubicados de manera específica para la grabación de sonidos. Con estas técnicas se pueden obtener imágenes estereofónicas, cuadrafónicas, envolvente, etc. Dependiendo del tipo de arreglo implementado. (G. Roy, s.f.).

#### **6.1.6. Técnicas de reproducción**

Las técnicas de reproducción son la composición de dos o más altavoces ubicados de manera específica para la reproducción de audios. Con estas técnicas se pueden obtener imágenes estereofónica, cuadrafónica, envolvente, etc. Dependiendo del tipo de arreglo implementado

#### **6.1.7. Plugin**

Un plugin es aquella aplicación que, en un programa informático, añade una funcionalidad adicional o una nueva característica al software. Es decir, es un subprograma que se asemeja a un complemento.

### **6.1.8. Diseño sonoro**

El diseño sonoro es el campo que se encarga de brindar todo el aspecto sonoro dentro de un producto audiovisual, este debe complementar los campos visual e interactivo.

Dentro del diseño sonoro se encuentran cuatro aspectos fundamentales divididos de la siguiente manera: Diálogos, efectos, música y ambiente. Los diálogos son la parte principal dentro del esquema de diseño sonoro debido a que se encarga de transmitir el mensaje hacia el espectador. Los efectos se encargan de la reconstrucción sonora de objetos dentro de un entorno, pueden ser realizados mediante Foley o procesos de síntesis electrónica de sonido. Los ambientes son aquellos audios que hacen parte característica de un entorno, como, por ejemplo, el mar en la playa, los pájaros y el viento en el bosque, los autos en una ciudad etc. Y por último esta la música, la cual se encarga de dar la tensión emocional en el campo sonoro (Simon, 2005 ).

## **6.2. Marco Teórico**

### **6.2.1. Técnicas de grabación y reproducción**

#### **6.2.1.1. Técnica de reproducción ambisonics**

El sistema de reproducción Ambisonics es un arreglo que permite generar sonido envolvente a partir de un arreglo de parlantes, distribuidos alrededor del oyente. El primero de estos sistemas, fue el sistema *cuadrafónico*, desarrollado por Michael Gerzon (Elen, 2001), es un sistema en el cual se ubican cuatro parlantes de manera regular, formando un cuadrado donde cada parlante recibe un nombre específico de acuerdo a su localización (Lr-Lf-Rr-Rf).

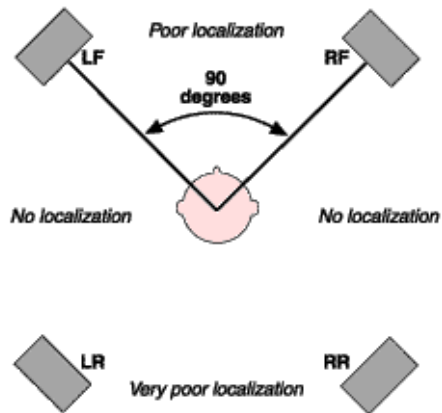


Figura 1. Sistema Cuadrafónico Ambisonic. Imagen tomada de: Ambisonic.net

Este arreglo de parlantes pretendía reproducir un sonido envolvente y permitir que el oyente experimentara una sensación de Surround. Sin embargo, el sistema tenía problemas de localización en los laterales, donde las fuentes eran difusas y difíciles de ubicar. Posterior a este vinieron otras configuraciones de parlantes, pero siempre manteniendo una regularidad en el arreglo, donde cada parlante tiene un opuesto en la distribución (Aguacia Andres, 2014).

### 6.2.1.2. Técnica de micrófona Ambisonics Nimbus Halliday

Para el registro de audio se experimentó varios arreglos de micrófonos para recoger la mayor cantidad de información en todos los ejes (arriba – abajo, frente - atrás, izquierda - derecha); a este tipo de arreglo se le llamo *A-Format*, pero dicho formato no se utiliza por si solo puesto que requiere ser codificado para los sistemas ambisonics; A dicha codificación se le denomino *B-Format*, compuesto por cuatro señales de entrada, una (1) omnidireccional y tres (3) figura ocho (Courville, 2007 - 2010). Una vez el arreglo *B-Format* se volvió un método estándar para el registro de audio ambisonic, la empresa Calrec Audio diseño un tipo de micrófono con el fin de facilitar la grabación de señales de audio que serán reproducidos en los sistemas ambisonic, a este micrófono se le llamado *Soundfield*.

El Doctor Jonathan Nimbus Halliday diseño un arreglo conformado por micrófonos convencionales con el objetivo de simular el micrófono *SoundField*. El arreglo nombrado *Nimbus Halliday* debe estar conformado por tres micrófonos, preferiblemente de diafragma pequeño, donde: el micrófono 1 (W) representa la salida de presión y debe tener un patrón polar omnidireccional y los micrófonos 2 y 3 (X, Y), deben ser bidireccional y representan la salida gradiente de presión en un plano horizontal, en el cual, el lado positivo de la capsula debe estar apuntando

hacia la parte izquierda y frente respectivamente en el arreglo. En la implementación de estos arreglos también se puede adicionar un cuarto micrófono con patrón polar bidireccional para complementar la salida gradiente de presión en un eje tridimensional (X, Y, Z). Las capsulas de estos micrófonos deben estar lo más cercanas posibles una de la otra (Elen, 2001).

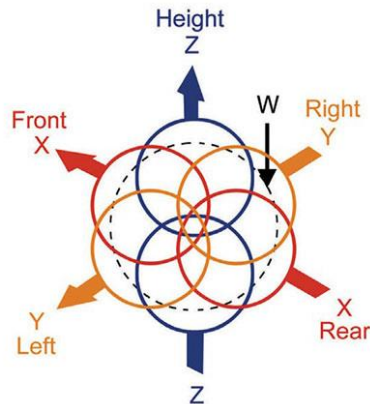


Figura 2. Técnica de micrófona Niimbus Halliday. Imagen tomada de: tvtechnology.com

### 6.2.1.3. Técnica de micrófona AB

La técnica de micrófona par espaciado o A-B, se encuentra entre las técnicas de micrófona no coincidentes. Es recomendable hacer uso de micrófonos con patrón polar cardiode u omnidireccional para la grabación de señales de audio. En esta clase de técnica se puede encontrar diferencias de tiempo o fase con respecto a la señal de audio grabada debido a la distancia que se encuentra de un micrófono a otro.

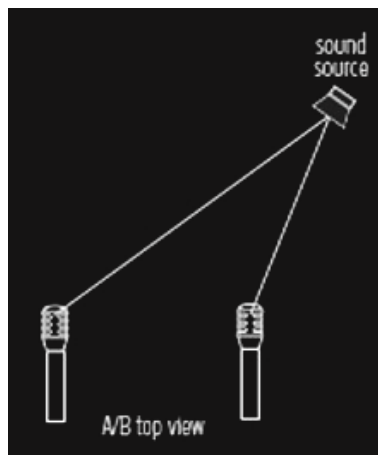
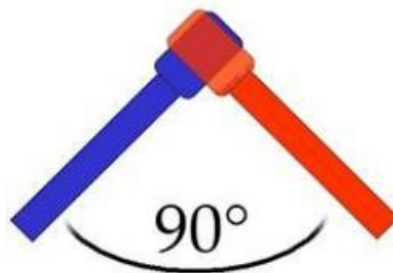


Figura 3. Técnica de micrófona AB. Imagen tomada de: shure.es

Una consideración importante cuando preparamos una grabación A-B estéreo es la distancia entre los micrófonos. Usando una distancia recomendada entre micrófonos de un cuarto de la longitud de onda del tono más bajo, y teniendo en cuenta la reducida capacidad del oído humano para localizar frecuencias por debajo de 150 Hz, llegamos a una distancia óptima entre 40 y 60 cm. Distancias menores se usan a menudo para captar fuentes de sonido próximas, para prevenir que la imagen del sonido de un instrumento concreto sea demasiado ancha y poco natural. Distancias por debajo de 17 o 20 cm son detectables para el oído humano porque es la separación equivalente a los oídos (G. Roy, s.f.).

#### **6.2.1.4. Técnica de micrófonia XY**

La técnica XY se encuentra entre las técnicas de micrófonia coincidentes. Se basa en el uso de dos micrófonos cardioides, preferiblemente del mismo fabricante, situados en un Angulo de  $90^\circ$  entre sus ejes, también se puede aplicar ángulos de cobertura entre  $120^\circ$  a  $135^\circ$ . Las capsulas de los micrófonos deben estar lo más próximas entre si una de la otra con una distancia máxima de 30cm (Shure, s.f.). Se recomienda que las capsulas se encuentre una encima de la otra, alineados verticalmente en el mismo punto para evitar problemas de fase con respecto a la separación de micrófonos.



*Figura 4. Técnica de micrófonia XY. Imagen tomada de: [analfatecnicos.net](http://analfatecnicos.net)*

Mientras que el A-B estéreo es un estéreo por diferencia de tiempo, el sistema XY estéreo es un estéreo por diferencia de amplitud. Pero como la atenuación por desviación del eje de un cardiode de primer orden es solamente de 6 dB en  $90^\circ$ , la separación del canal está limitada, y no son posibles amplias imágenes estéreo con este método de captación. Por tanto, el XY estéreo se usa a menudo cuando se necesita alta compatibilidad mono (por ejemplo, en emisiones radiofónicas donde la audiencia utiliza receptores monos para escucharlas) (G. Roy, s.f.).

## 6.2.2. Motor para el desarrollo de videojuegos

### 6.2.2.1. Unity 3D

Unity es un motor gráfico al framework de software, diseñado para desarrollar y crear videojuegos en donde convergen herramientas útiles y necesarias para realizar aplicaciones dinámicas en 2D o 3D. Tiene la posibilidad de expandir las herramientas para brindar un juego optimizado en rendimiento, fiabilidad, fluidez, poder de renderizado y ambientación, además cuenta con una interacción rápida para el desarrollo multiplataforma. En otros términos, es un ambiente o entorno de trabajo que facilita el desarrollo de aplicaciones con el soporte de programas, bibliotecas, máquinas virtuales, compiladores administración de recursos, rutinas de programación que están enfocados hacia: Scripting, Animación, Graficación, Audio, Optimización física, Detector de colisiones, Inteligencia artificial.

El audio cuenta con una versátil e interactiva configuración de sonido e importación de archivos en el proyecto que crean audio clips, fuentes de emisión y listeners, que gracias a esto es posible manipular características de mezcla en tiempo real en todo el campo sonoro, jerarquía entre mixers y capturas predefinidas de efectos que provee Unity.

Unity basa su teoría sonora en la vida real, los sonidos son emitidos por una fuente que es escuchada en una dirección y en una distancia, y además contiene un efecto doppler para lograr mayor naturalidad en el sonido.



Figura 5. Fuente de audio y escucha. Imagen tomada de: docs.unity3d.com

Es necesario que en Unity el sonido sea originado por una fuente de audio ligada a un objeto, y así ser “escuchada” o recibidas por el audio listener (micrófono interno que recibe señal de entrada de cualquier fuente de sonido en unity y reproduce esta señal por los altavoces del dispositivo) que tiene la capacidad de calcular la distancia y posición que tiene frente al objeto emisor.

### 6.2.2.2. FMOD Studio

FMOD Studio es un software Middleware (Lógica de intercambio de datos entre aplicaciones) dedicado a la programación de audio en videojuegos incluyendo capas, librerías y efectos que pueden ser útiles a la hora de diseñar un entorno de audio inmersivo. Se optó por utilizar este programa para el desarrollo del audiojuego debido a que, su interfaz gráfica es amigable y propone un mejor entendimiento del software a aquellas personas involucradas en el diseño sonoro con conocimiento en el manejo de programas de edición de audio. Además, su vinculación con el software de programación de videojuegos *Unity* ofrece una mejor respuesta, ya que, *Unity* brinda un archivo de integración exclusivo para FMOD Studio, lo cual permite estabilidad en el sistema durante el intercambio de datos.

Para la programación de audio dentro de FMOD Studio se debe tener conocimiento de las herramientas que brinda el programa. A continuación, se presentarán las herramientas que serán usadas constantemente para el desarrollo del audiojuego.

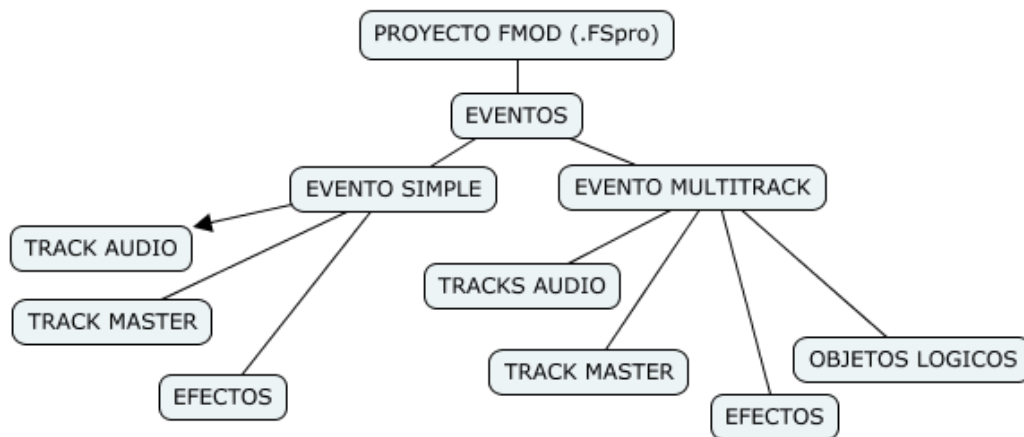


Figura 6. Diagrama de flujo proyecto básico de FMOD Studio, Imagen propia

- Evento

Evento es el objeto principal dentro de FMOD Studio. Un evento es la ventana de trabajo y edición donde se agregan los archivos de audio que serán reproducidos en el videojuego; Adicional a esto, también se pueden agregar objetos lógicos de programación de FMOD Studio tales como: Marcadores, LoopRegion, Transition, etc.



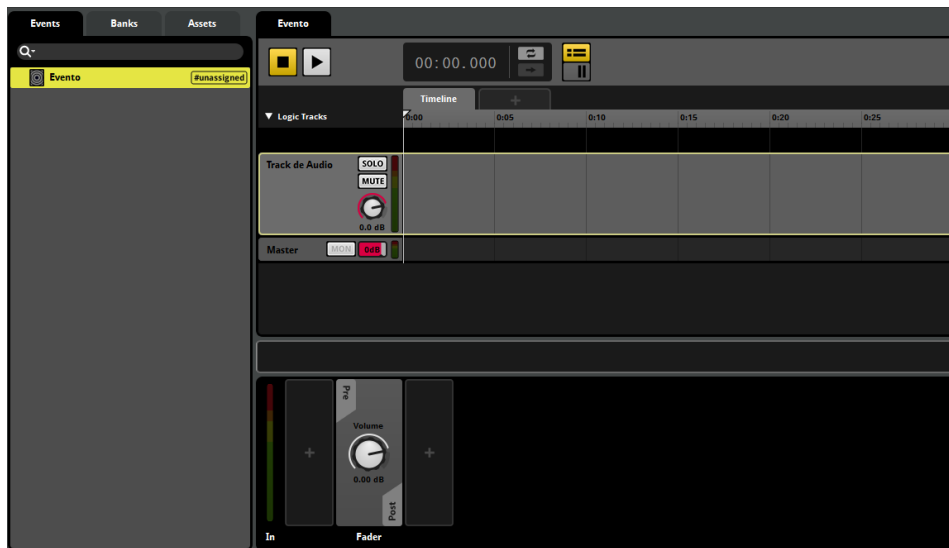


Figura 7. Evento en FMOD Studio, Imagen propia

Un evento, inicialmente, está compuesto por una barra de transporte (Play, stop, visualizador de tiempo, visualizador de ventana), capa de plugins y efectos, un track de audio y un track master; se pueden adicionar más tracks de audio, pero siempre manteniendo un único track master. En FMOD Studio los eventos están categorizados en dos grupos, eventos simples y eventos multitrack.

- Evento simple

Un evento simple contiene un solo track de audio, puede ser un archivo simple o un archivo multisound. No contiene objetos lógicos de programación.

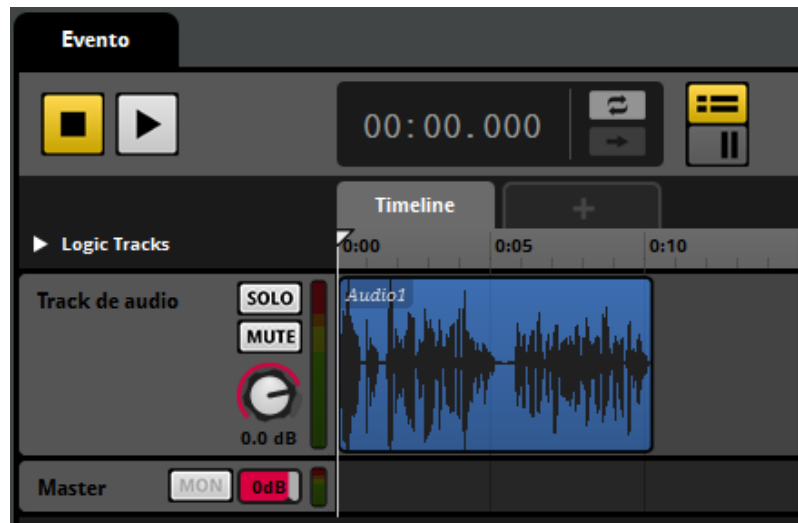


Figura 8. Evento simple en FMOD Studio, Imagen propia

- Evento Multitrack

Un evento Multitrack está compuesto de objetos lógicos de programación y uno o más tracks de audio (archivo simple o archivo multisound).

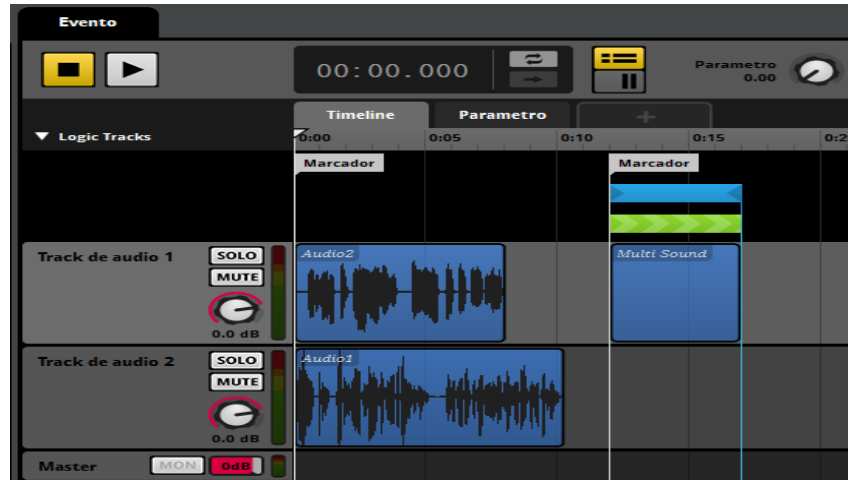


Figura 9. Evento Multitrack en FMOD Studio, Imagen propia

- Archivo Multisound

Los objetos multisound permiten almacenar varios archivos de audio en la misma ubicación de tiempo dentro de un evento. Los audios agregados a los objetos multisound son almacenados en una *Playlist* la que permite que sean reproducidos de manera aleatoria o lineal. No se puede reproducir más de un audio a la vez.



Figura 10. Archivo Multisound en FMOD Studio, Imagen propia

- Marcador

Los marcadores son puntos o “marcas” que almacenan ubicación de tiempo dentro de un evento. Suelen usarse para indicar el inicio o fin de un audio.

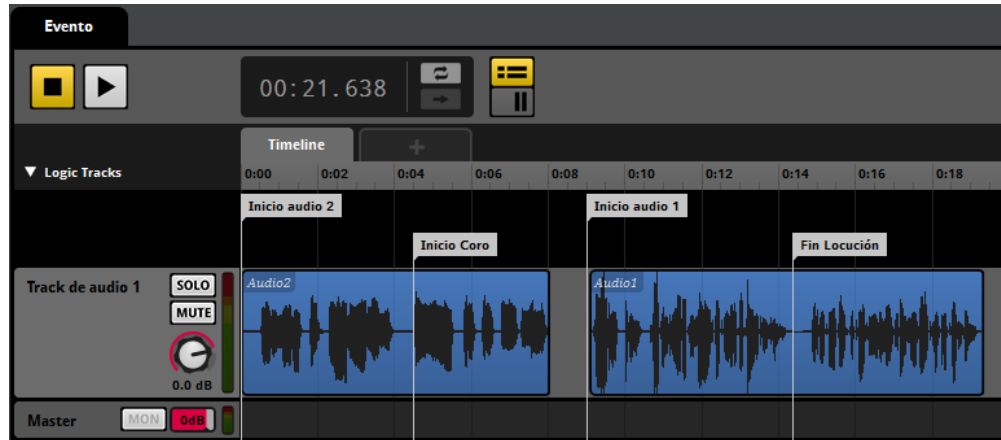


Figura 11. Uso de marcadores en un evento de FMOD Studio, Imagen propia

- Transition

Transition son puntos (TransitionPoint) o regiones (TransitionRegion) de color verde que tienen como función trasladar y ubicar el cursor de reproducción en los marcadores situados dentro de un evento.

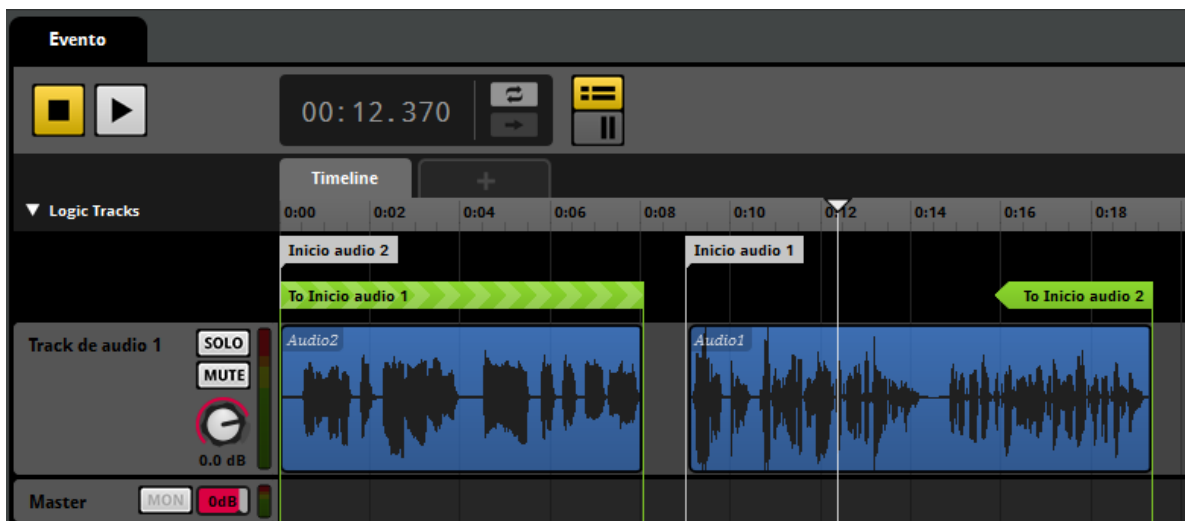


Figura 12. Tipos de Transition en FMOD Studio, Imagen propia

En el caso del TransitionRegion, el cursor será trasladado cuando se encuentre dentro de la región establecida; Y en el caso del TransitionPoint, el cursor de reproducción será trasladado cuando este mismo alcance el punto donde se encuentra situado el TransitionPoint.

- LoopRegion

LoopRegion son fragmentos de color azul, cuyo fin es activar un bucle/Loop de reproducción de uno o varios tracks dentro de un evento.

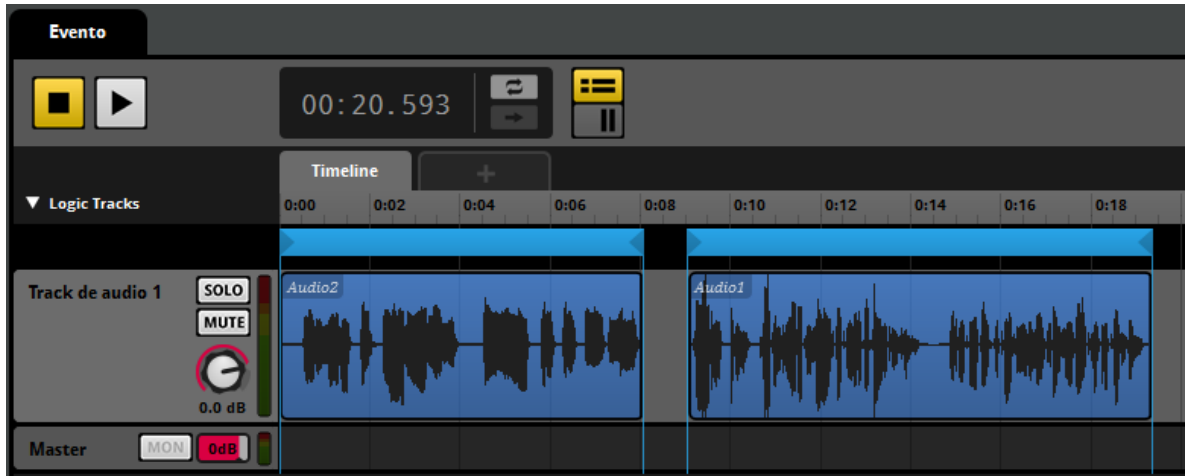


Figura 13. LoopRegion en FMOD Studio, Imagen propia

- Parámetro

Los parámetros son objetos lógicos en forma de Knob o potenciómetro, los cuales tienen como objetivo asignar diferentes acciones a otros objetos (Activación/desactivación de LoopRegion, activación/desactivación de Transition, activación/desactivación efectos, asignación de paneo, habilitar/deshabilitar archivos audio, etc.) dentro de un evento. Estas acciones son ejecutadas dependiendo del valor numérico asignado en el parámetro.

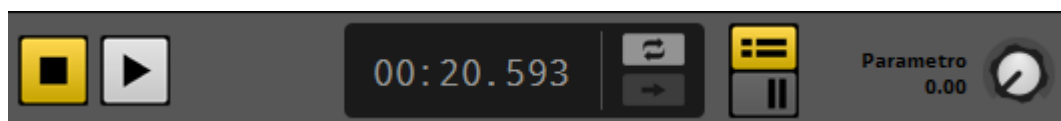


Figura 14. Barra de transporte FMOD Studio, Imagen propia

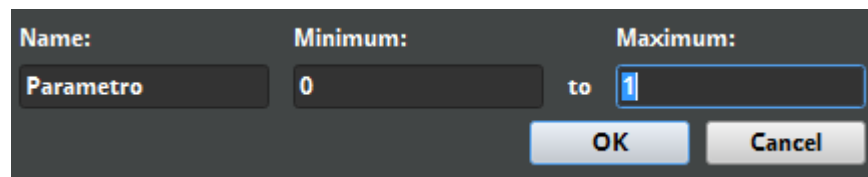


Figura 15. Valor numérico máximo y mínimo de un parámetro FMOD Studio, Imagen propia

Para entender mejor el funcionamiento de un parámetro, se expone el siguiente ejemplo:

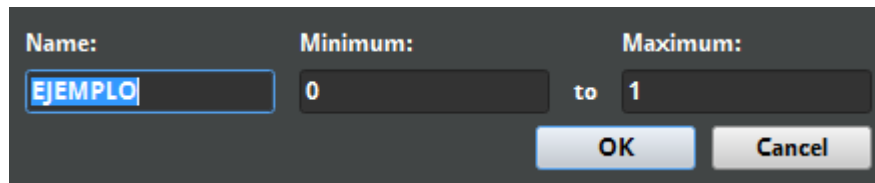


Figura 16. Parámetro Ejemplo en FMOD Studio, Imagen propia

Se crea un parámetro nombrado *Ejemplo*; A este parámetro se le asigna un valor mínimo de 0 y un valor máximo de 1. La función de este parámetro es activar y desactivar los TransitionPoint establecidos dentro del evento; cuando *Ejemplo* es igual a uno (1) se activa el TransitionPoint *Audio 2* y desactiva el TransitionPoint *Audio 1* pero cuando *Ejemplo* es igual a cero (0), se activa el TransitionPoint *Audio 1* y desactiva el TransitionPoint *Audio 2*.

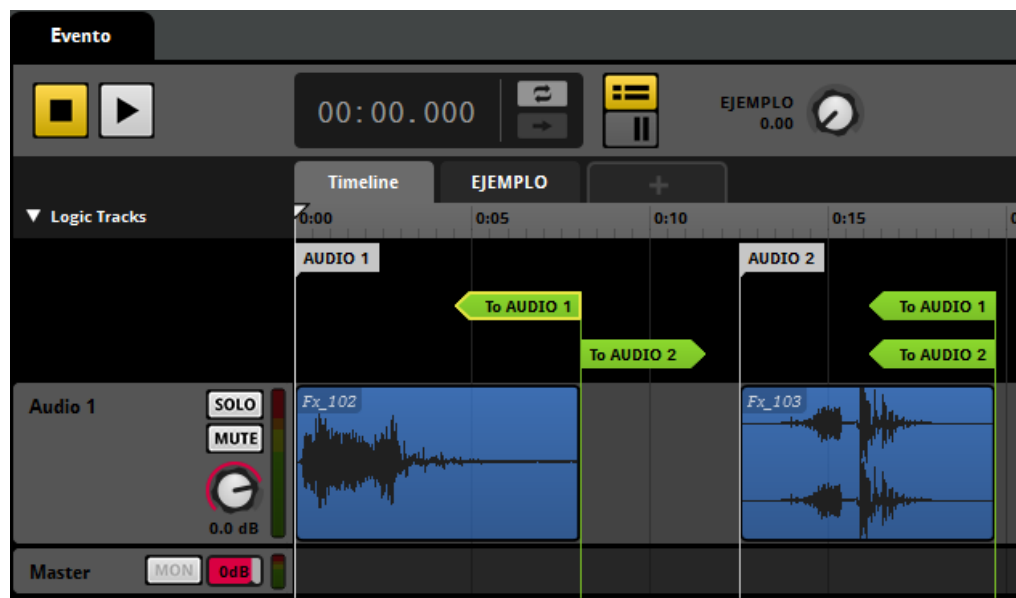


Figura 17. Evento Parametro Ejemplo en FMOD Studio, Imagen propia

Esto permite la reproducción de los audios ubicados en el evento, si TransitionPoint *Audio 1* está activado, reproducirá el audio *Fx\_102* pero cuando TransitionPoint *Audio 2* está activado, reproducirá el audio *Fx\_103*.

- Paneo

El paneo permite ubicar una fuente de sonido en el espacio estereofónico o multicanal dependiendo del número de parlantes incluidos en el arreglo y del plugin de mezcla implementado dentro de FMOD Studio, en este caso el *Oculus audio*. Cada evento tiene su panel de paneo independiente.



Figura 18. Paneo y canales implementados dentro de un evento en FMOD Studio, Imagen propia

### 6.2.3. Plugin

#### 6.2.3.1. Oculus Audio SDK

Oculus audio es un plugin que permite codificar señales de entrada para el desarrollo de audio envolvente para cualquier tipo de formato virtual. La ventaja de este plugin es que tiene como parámetro configurar arreglos de sonido envolvente como: 5.1, 7.1, audio 3D, sistemas ambisonics de cuatro (4) u ocho (8) fuentes, etc. Su funcionamiento se basa en emular parámetros físicos acústicos para lograr el sonido envolvente deseado.

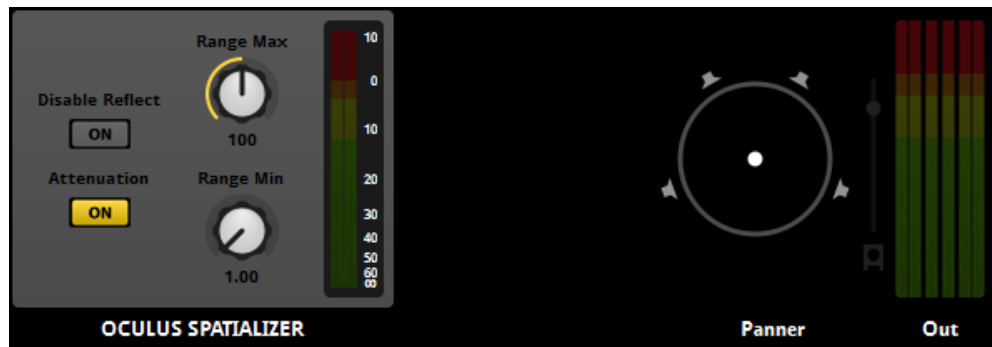


Figura 19. Oculus Audio configurado para una salida ambisonics de cuatro canales, imagen propia

- **Parámetros Oculus Audio**

- **Parámetros de track locales**

**Gain:** Incrementa el volumen de la señal procesada en decibeles (dB) en un rango de 0 a 24

**Near:** Establece la distancia entre el oyente y la fuente de sonido dada en metros en un rango de 0 a 175 al igual que la curva de atenuación la cual se aproxima a un cuadrado inverso y llega a la atenuación máxima cuando alcanza el parámetro FAR

**Far:** Define la distancia con respecto al oyente a la que una fuente de sonido alcanza el valor máximo de atenuación en metros en un rango de 0 a 175.

**X/Y/Z Pos:** Define la locación de la fuente de sonido con respecto al oyente

- ***Parámetros de track globales***

**Reflections:** Alterna el motor de reflexión, según la definición de los parámetros globales de reflexión, lo cual, mejora el efecto de especialización

**Reverb:** Cuando esta activada, una reverberación fija se mezcla en la salida, proporcionando un efecto de especialización de sonido más natural. Sobre la base de los valores de tamaño de habitación y de reflexión (X/Y/Z). Si los parámetros de X/Y/Z o Near, Far y Reflections alterara los valores de la reverberación establecida.

**X/Y/Z Size:** Define las dimensiones de un recinto teórico, usado para calcular las reflexiones. Las dimensiones son dadas en metros en un rango de 1 a 200.

**Left/Right - Forward/Backward – Up/Down Reflections:** Establece el porcentaje de sonido reflejado por cada pared en una habitación con las dimensiones especificadas por los parámetros X/Y/Z Size. Donde 0 es totalmente absorbente y 1 totalmente reflejante

#### **6.2.4. Interfaz de control**

##### **6.2.4.1. Kinect**

Kinect es un sensor de movimientos de interfaz natural de usuario extremadamente sensible, con el cual es posible controlar juegos o aplicaciones sin necesidad de tener contacto con un controlador físico tradicional, por medio de este dispositivo se envían órdenes a consola según los gestos utilizados proporciona la captura del movimiento corporal, también es posible reconocer por medio de este dispositivo los movimientos de la cara, la localización de la fuente acústica por medio de sonido y el reconocimiento de voz.

El sensor de Kinect se compone por una barra horizontal de 23 cm de largo que contiene una cámara RGB, un emisor infrarrojo (IR), que a su vez es combinado con un sensor CMOS monocromo que permite al dispositivo ver el recinto en cualquier condición de luz ambiental, el rango de detección de profundidad que posee se ajusta de manera automática basado en el ambiente físico del usuario. Además de eso posee un micrófono compuesto por varias matrices que permite conocer la ubicación del usuario.

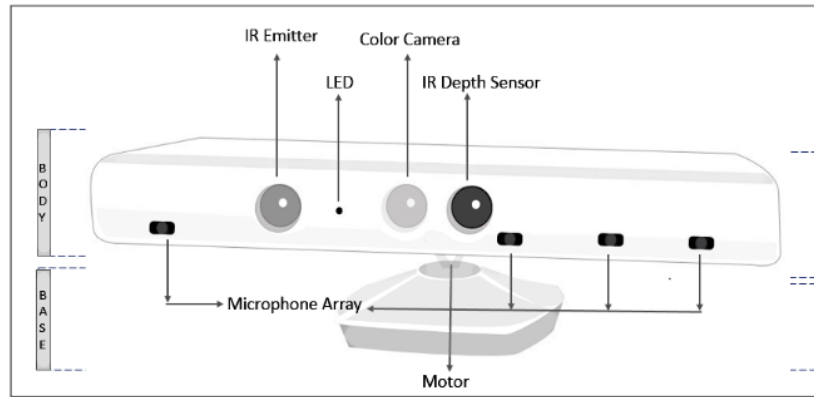


Figura 20. Características Kinect, Imagen tomada de: [doc.flashrobotics.com](http://doc.flashrobotics.com)

El sensor de movimientos basa su funcionamiento en el Tracking System, puede ser usado para rastrear el esqueleto o para el reconocimiento facial. Estas poses o gestos son reconocidos con ayuda de algoritmos específicos que son desarrollados según la jugabilidad que se desee.

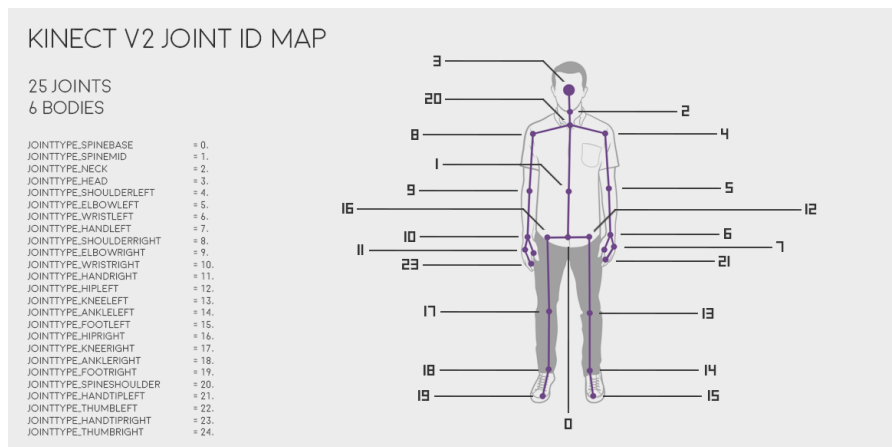


Figura 21. Reconocimiento corporal Kinect, Imagen tomada de: [org/documentation/kinect](http://org/documentation/kinect)

## 6.3. Marco normativo

### 6.3.1. ESRB: Entertainment Software Rating Board

La Junta de Clasificación de Software de Entretenimiento (ESRB, Entertainment Software Rating Board) proporcionan una información concisa y objetiva acerca del contenido de los juegos de video y las aplicaciones para que los consumidores, en especial los padres, puedan tomar decisiones informadas. Las clasificaciones de la ESRB constan de tres partes:



- **Categorías de clasificación:** Sugieren la edad adecuada para el juego.
- **Descriptor de contenido:** indican los elementos que pueden haber motivado la clasificación asignada y/o pueden resultar de interés o preocupación.
- **Elementos interactivos:** Informan acerca de los aspectos interactivos de los productos, incluida la capacidad de los usuarios de interactuar, si se comparte la ubicación de los usuarios con otros usuarios o si es posible que se comparta información personal con terceros.

### 6.3.2. Norma EBU R-128

Normalización de la sonoridad y nivel máximo permitido de las señales de audio.

La EBU R-128 considera que es necesario un método de nivelación de la señal de audio en producción, distribución y transmisión de programas, debido a que la normalización pico de las señales de audio que han generado los diferentes medios ha ocasionado considerable diferencias e inconsistencia de sonoridad entre programas y emisora, dando como resultado quejas por parte de los televidentes y oyentes.

La caracterización de la señal de audio se basa en la sonoridad (“Programme Loudness”), las mediciones de rango de sonoridad (“Loudness Range”) y el máximo nivel de pico verdadero (“Maximum True Peak Level”) para una buena normalización de la señal de audio y de esta manera cumplir y estar entre los límites técnicos de la señal de audio según sus requerimientos estéticos según el programa / radiodifusión, género y audiencia. También para aquellos programas que cuentan con una gran variedad de rangos sonoros da como ventaja mejorar la concordancia entre la sonoridad de las diferentes señales de audio. Para realizar la normalización de la señal de audio se debe hacer una medición general en su totalidad y no enfatizar o individualizar ningún elemento en específico. El nivel de normalizado debe ser de -23 LUFS con una desviación que no debe exceder más o menos, 1 LU, y su máximo nivel de pico verdadero sea -1 dBTP (“dB True Peak”).

### **Definiciones EBU R-128:**

- **Programa:** Elemento audiovisual o solo audio que vaya ser presentado en radio, televisión u otro medio electrónico.
- **Sonoridad de Programa:** El nivel de sonoridad de un programa es el valor de la sonoridad del programa (LUFS)
- **Rango sonoro:** Distribución de sonoridad dentro de un programa.
- **Máximo nivel de pico verdadero:** Valor máximo de la señal de audio de un programa en un dominio continuo del tiempo.

## 7. METODOLOGIA DEL DISEÑO

### 7.1. Desarrollo ingenieril

A continuación, se dará a conocer el proceso que se llevó a cabo para el desarrollo del audiojuego, explicando objetivo por objetivo y dando a conocer cuáles fueron las tareas que se realizaron para cumplir con el objetivo propuesto.

Teniendo en cuenta que el audiojuego será ubicado dentro de una instalación interactiva, se propone un juego de tipo *Aventura* en el cual, el jugador tendrá que realizar una serie de movimientos corporales para aprobar y superar retos con el fin de avanzar en los niveles establecidos.

#### 7.1.1. Sinopsis del audiojuego

Ekko, el titán del sonido, se encuentra encarcelado en un volcán por culpa de dos titanes, Yacu y Dante. Ekko pide ayuda a un ser terrenal (Jugador) para recuperar los amuletos que le harán recobrar su libertad. Una vez el jugador acepte ayudar a liberar a Ekko, este le brindara todos los poderes necesarios para poder conseguir los amuletos, cada amuleto custodiado por un titán (El jugador puede escoger que titán desea enfrentar primero). Para obtener los amuletos, el jugador debe superar una serie de obstáculos y trampas impuestos por los titanes Yacu y Dante en cada mundo (Acuático y selvático respectivamente); cuando estos obstáculos sean superados, el jugador tendrá que combatir con el titán del momento para poder destruirlo y conseguir el amuleto. Con los dos amuletos obtenidos, el jugador puede liberar a Ekko.

En el anexo E se puede encontrar el guion literario y técnico, donde se expone la jugabilidad, el diseño sonoro propuesto y los diálogos correspondiente a este audiojuego.

#### 7.1.2. Niveles del audiojuego

El audiojuego está compuesto por 4 niveles distribuidos de la siguiente manera:

- **Nivel 1:** *Escenario de entrenamiento*

En este nivel, y como su nombre lo indica, se encuentra toda la parte de entrenamiento. Acá, el jugador, ubicado en un volcán en reposo, deberá seguir las instrucciones brindadas por la voz guía para aprender a realizar los movimientos que serán requeridos a lo largo del audiojuego

- **Nivel 2:** *Escenario de introducción*

El escenario de introducción contiene la historia del personaje Ekko y el contenido sobre lo que tendrá que realizar el jugador en los niveles posteriores

- **Nivel 3:** *Escenario acuático de Yacu*

El nivel de Yacu es caracterizado por su entorno acuático, en este, el jugador dependerá de movimientos corporales enfocados en las manos. Este nivel está compuesto de dos (2) retos a lo largo del nivel.

- **Nivel 4:** *Escenario selvático de Dante*

El nivel de Dante es caracterizado por su entorno selvático, en este, el jugador dependerá de movimientos corporales enfocados en los pies. Este nivel está compuesto de dos (2) retos a lo largo del nivel.

### **7.1.3. Clasificación del audiojuego**

Tomando como referencia la junta ESRB, se establece la siguiente clasificación para el audiojuego:

- **Categorías de clasificación**

Se asignó en la categoría E+10 debido a que los diálogos y la narrativa del juego contiene características dadas por la organización.



Figura 22. Logotipo de clasificación Everyone +10, Imagen tomada de: esrb.org

E +10: *El contenido por lo general es apto para personas de 10 años o más. Puede que contenga más violencia de caricatura, de fantasía o ligera, lenguaje moderado o temas mínimamente provocativos (ESRB, s.f.).*

- **Descriptores de contenido**

El audiojuego presenta los siguientes descriptores de contenido sugeridos por la ESRB:

- *Violencia de caricatura*: acciones violentas que incluyen situaciones y personajes caricaturescos. Puede incluir violencia en la cual un personaje sale ileso después de que la acción se llevó a cabo.
  - *Travesuras cómicas*: representaciones o diálogo que impliquen payasadas o humor sugestivo.
  - *Violencia de fantasía*: acciones violentas de naturaleza fantástica que incluyen personajes humanos y no humanos en situaciones que se distinguen con facilidad de la vida real.
- **Elementos interactivos**

El audio juego no presenta ninguno de los elementos mencionado a continuación:

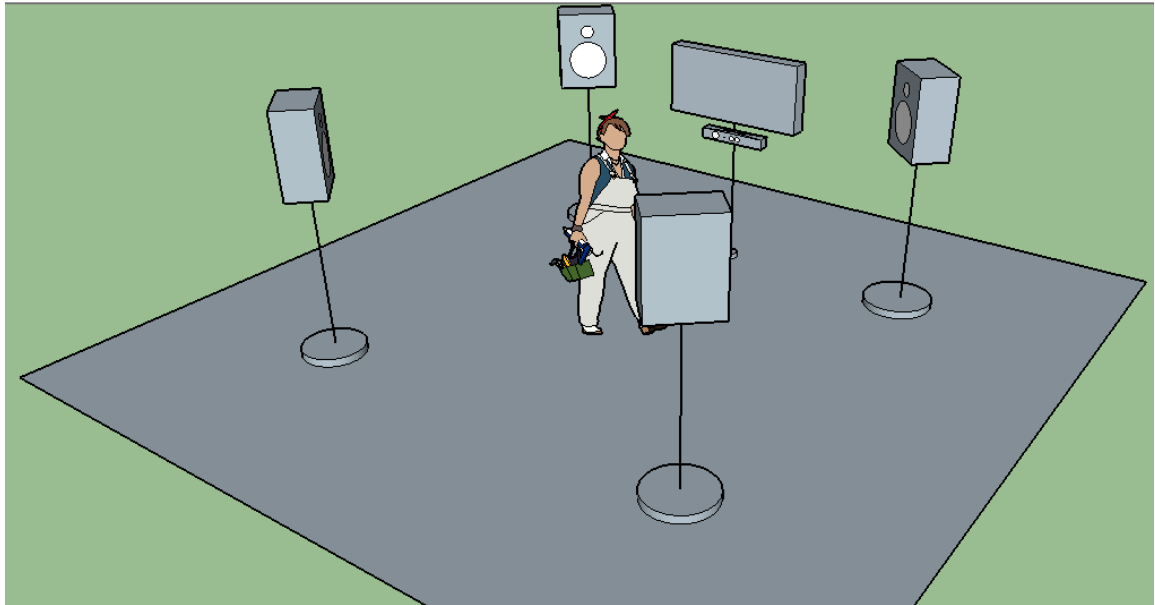
- *Información compartida*: Indica que la información personal brindada por el usuario (por ejemplo, dirección de correo electrónico, número de teléfono, información de tarjeta de crédito, etc.) se comparte con terceros.
- *Ubicación compartida*: Incluye la capacidad de mostrar la ubicación del usuario a otros usuarios de la aplicación.
- *Interacción de usuarios*: Indica una posible exposición a contenido sin filtro y sin censura generado por usuarios, que incluye comunicaciones y medios compartidos de usuario a usuario a través de medios y redes sociales.
- *Compras digitales*: Permite la compra de productos digitales directamente desde la aplicación (por ejemplo, compras de contenido adicional del juego, niveles, música que pueda descargarse, etc.).
- *Internet sin límites*: El producto brinda acceso a Internet.

#### **7.1.4. Instalación interactiva y hardware del audiojuego**

La instalación interactiva y el hardware del audiojuego se construye en los estudios de la universidad de San Buenaventura Bogotá. Dicha ubicación se determinó debido a que la universidad cuenta con los equipos requeridos para la instalación.

Se estableció un sistema de reproducción cuadrafónico ubicado de tal manera que emule el primer arreglo ambisonic establecido por Michael Gerzon (Aguacia Andres, 2014). No se establece un arreglo con un número mayor de fuentes debido a la falta de recursos y disposición de estos mismos.

El objetivo del hardware del audiojuego es brindar un espacio amplio, en el cual, el jugador tenga el beneficio de moverse libremente por la instalación, es por eso que se establece el siguiente diseño:



*Figura 23. Plano de la instalación interactiva y su distribución, imagen propia*

Se establece una distancia de metro y medio (1,5 m) entre el jugador y el Kinect, y de un (1) metro entre el suelo y la altura del kinect. Esta distancia se fija buscando la mejor posición en que el Kinect capte toda la superficie corporal del jugador en un rango de 150 cm a 190 cm de estatura.

Debido a que el jugador se debe posicionar en la mitad del arreglo ambisonic, se toma como base la distancia del jugador con el Kinect. Este valor es duplicado para instaurar la distancia que habrá de un parlante a otro dentro del arreglo, es por esto y como se puede observar en la figura 24 y 25, se fija una distancia de tres (3) metros entre fuente y fuente.



Figura 24. Instalación interactiva ubicada en el estudio híbrido de la Universidad San Buenaventura Bogotá, Imagen propia

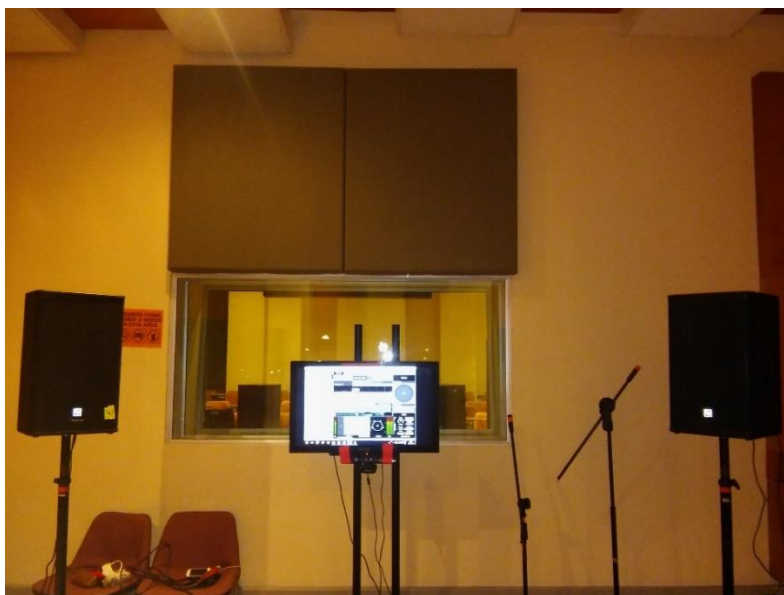


Figura 25. Parte inferior de la instalación interactiva, Imagen propia

### **7.1.5. Diseño sonoro y grabaciones de audio**

En el caso del objetivo *implementar sistemas de captura de sonido para la reproducción en ambisonics*, se estableció como primera medida un arreglo de cuatro (4) micrófonos teniendo como base el arreglo Nimbus Halliday para la grabación.

- **Grabación de voces**

La grabación de los personajes Ekko, Dante y Yacu se realizaron en el estudio digital de la Universidad de San Buenaventura sede Bogotá. Para esta práctica, se utilizaron un par de micrófonos AudioTechnica AT 4050 y un par de micrófonos AKG C414 dado que, son los micrófonos disponibles en la universidad con características más cercanas a las sugeridas para realizar el arreglo Nimbus Halliday (Courville, 2007 - 2010). En la página 27 se puede observar las características sugeridas para la implementación de este arreglo.

En el anexo C se puede encontrar la respuesta en frecuencia y el patrón polar de los micrófonos mencionados anteriormente.



*Figura 26. Arreglo de micrófonos técnica Nimbus Halliday, Imagen propia*

En este caso, tal y como se ve en la figura 26, se distribuyó el arreglo de micrófonos de la siguiente manera:

- Micrófono AKG C414 Omnidireccional (W)
- Micrófono AudioTechnica AT 4050 Bidireccional (X) paredes Laterales
- Micrófono AudioTechnica AT 4050 Bidireccional (Y) Paredes Frontales
- Micrófono AKG C414 Bidireccional (Z) Techo-Suelo





Figura 27. Grabación de voz personaje Yacu, Imagen propia



Figura 28. Grabación de voz personaje Ekko y Dante, Imagen propia

Inicialmente la codificación de señales a ambisonic se iba a realizar mediante el software de programación PUREDATA, pero este fue remplazado debido a que FMOD brinda la inclusión del plugin *Oculus Audio*. Este plugin permite realizar el trabajo de codificación con mayor facilidad para el desarrollo de audio envolvente, además, su integración con la plataforma Unity es más simple.

Debido a que la versión usada de *Oculus audio* no deja codificar 4 señales de entrada en un mismo canal, y solo permite codificar señales de entrada mono/estéreo a formato ambisonics, se tuvo que volver a realizar la edición de diálogos con el fin de obtener la mejor información de cada uno de los 4 canales grabados anteriormente, convirtiéndolos a un solo canal monofónico mediante el uso de plugins ATK y tener así un buen funcionamiento en FMOD. Esta edición se ejecutó por la falta de disponibilidad de los actores para una nueva grabación.

Además de realizar la edición de diálogos, se optó por hacer uso de técnicas de microfónica diferentes para la grabación de los audios restantes.

- **Grabación de ambientes**

En la grabación de ambientes se utilizaron técnicas convencionales estéreo. Para esto se empleó la grabadora Tascam DR-40, la cual, brinda mayor comodidad a la hora de grabar ambientes debido a que trae integrado un arreglo de micrófonos estéreo XY – AB. De igual manera, también se utilizaron los micrófonos AudioTechnica At4050 y Neumann KM 184 para obtener una mejor calidad en el audio en el mismo formato de grabación estéreo.

- *Amb\_00 y Amb\_05*

Este ambiente requería sonidos característicos de una playa, es por eso que para diseñar este ambiente fue necesario usar la librería online *Freesound.org* a falta de locaciones en la ciudad con este tipo de características sonoras. Aparte de las librerías usadas, se agregaron capas de audio conformadas por “Splash” de agua contra estanques y piscinas, Soundscapes desarrollados en instrumentos virtuales como Absynth y Massive, y también se generó ruido blanco en Protools para simular el sonido cuando la marea choca contra las rocas en la playa.



*Figura 29. Grabación splash de agua, Imagen propia*

- *Amb\_01 y Amb\_02*

Estos ambientes simulan el sonido debajo del agua, para el diseño de este ambiente se usaron sonido de librería que fueron modificados tonalmente con el plugin Pitch Shift de Avid, y además se realizó una grabación con la Tascam DR-40 con un arreglo XY, dentro de un recipiente con agua, generando movimientos de agua las manos, además se agregaron capas en baja frecuencia por medio del instrumento virtua Absynth.

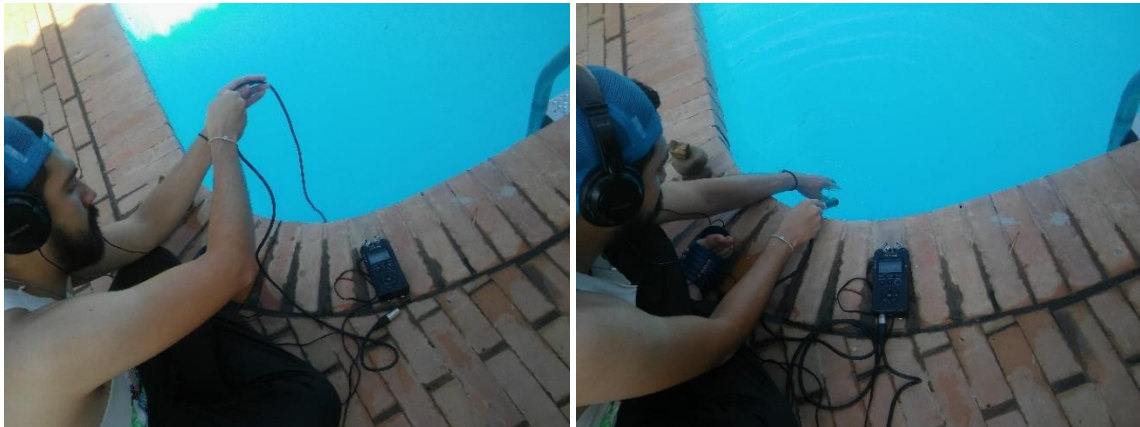


Figura 30. Grabación de ambiente acuático, Imagen propia

➤ *Amb\_03 y Amb\_04*

Para la grabación de un ambiente natural de estanque, se utilizó un arreglo estéreo AB, que posteriormente se ecualizó en software Protools.



Figura 31. Grabación de estanque Universidad San Buenaventura con la técnica de microfoneo estero AB, Imagen propia

➤ *Amb\_06*

El ambiente cueva se recreó utilizando varias capas de audio. Como primer paso se grabó ruido blanco reproducido mediante un amplificador en un recinto con mucha reverberación. Como segundo paso se adicionaron capas de audios conformadas por animales tales como: ratas y murciélagos (ambos audios obtenidos de la librería online *Freesound.org*), gotas de agua, arena, piedras, y piedras cayendo en un estanque. En cada uno del audio, excepto el ruido blanco,

se adicionaron tiempos de reverberación entre 2s a 5s dependiendo del audio elegido para recrear mejor el ambiente de cueva.



Figura 32. Grabación de ruido blanco en un recinto reverberante, Imagen propia

➤ *Amb\_07, Amb\_08, Amb\_09, Amb\_10, Amb\_11, Amb\_12, Amb\_13, Amb\_14*

Estos ambientes de naturaleza fueron grabados en una finca ubicada en la región de Sumapaz Tolima, se realizaron varias tomas en diferentes lugares y a diferentes horarios, obteniendo como resultado varias tomas de: viento, luciérnagas, matorrales, árboles, aves, estanques, insectos entre otros. Dichas grabaciones fueron realizadas en formatos XY y AB con los micrófonos internos de la grabadora Tascam DR-40.

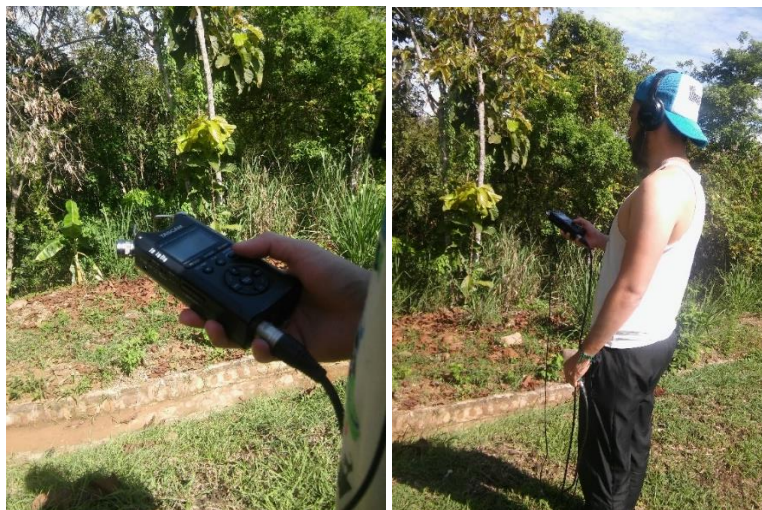


Figura 33. Grabación de ambientes en exterior, Imagen propia

- **Grabación de Foley y Efectos**

Para este caso se realizaron grabaciones con un solo micrófono o varios micrófonos obteniendo como resultado un archivo monofónico al final de la edición. Además de usar los micrófonos ya mencionados anteriormente, también se usaron los micrófonos 12Gauge Green y un Shotgun VidPro XM-55.

En el anexo C se puede observar las características de estos micrófonos

- *Fx\_000*

El audio Mosca fue obtenido mediante la librería online *Freesound.org*, complementado con osciladores y LFO del instrumento virtual Reaktor.

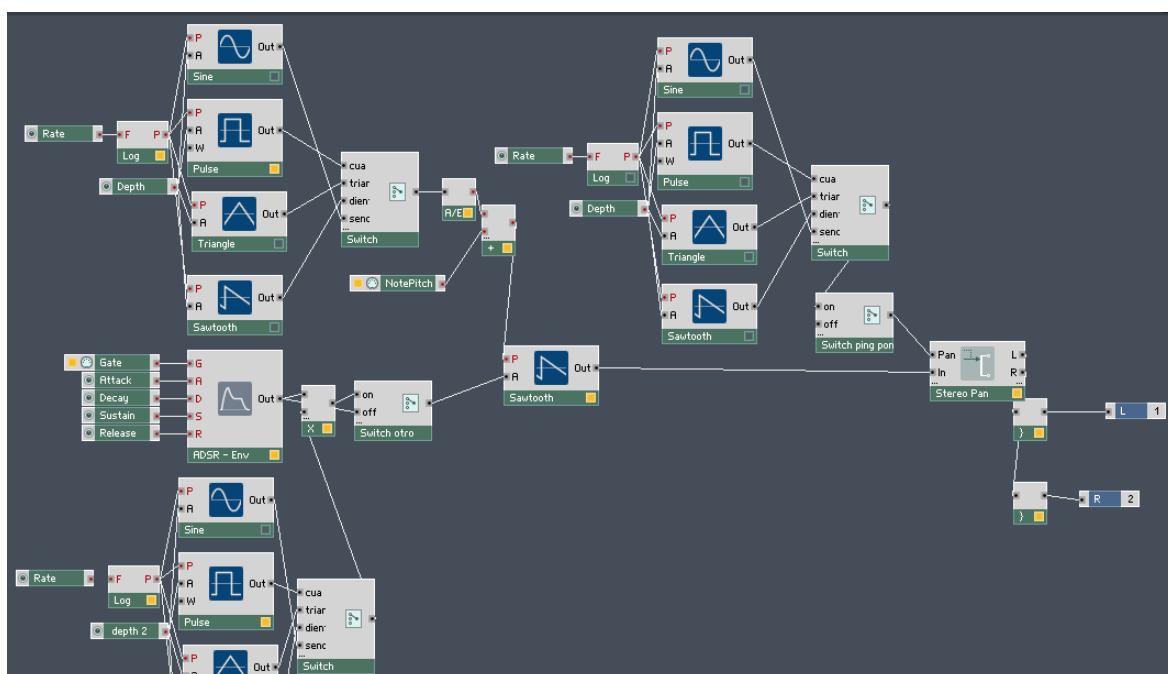


Figura 34. Ensamble en Reaktor correspondiente al efecto *Fx\_000*, Imagen propia

- *Fx\_001*

Para la creación del efecto - portal de tele transportación - se grabaron instrumentos tales como un palo de lluvia y bongos, que a su vez fueron mezclados con el uso de atmósferas sintéticas incluidas en el instrumento virtual Absynth.



Figura 35. Grabación de bongos y Palo de lluvia, Imagen propia

➤ *Fx\_002*

La puerta del castillo de yacu del nivel 1 se obtuvo mediante varias capas de Foley para lograr el sonido deseado. Se grabaron objetos tales como: Puertas metálicas, platos de bicicleta, choque entre piedras, cadenas y el bombo de una batería.



Figura 36. Grabación de cadenas y plato de bicicleta para la puerta del castillo, Imagen propia

➤ *Fx\_004, Fx\_005, Fx\_006, Fx\_007, Fx\_008, Fx\_009*

El sonido del propulsor se logró mediante la grabación de diferentes motores (Auto, Moto, Batidora, licuadora, Ventilador), asignando variaciones en pitch y en amplitud para lograr naturalidad en el sonido. El efecto que tiene el propulsor en el agua, se hizo grabando burbujas provocadas en un recipiente mediante un pitillo.



Figura 37. Grabación de motor de auto, Imagen propia

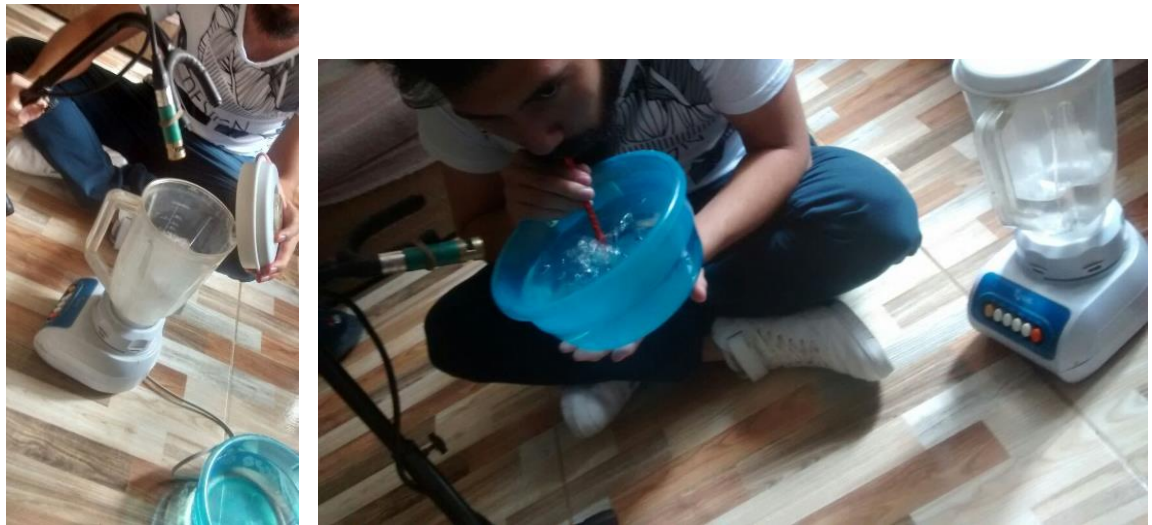


Figura 38. Grabación de burbujas provocadas mediante el uso de licuadora y pitillo, Imagen propia

➤ *Fx\_010*

El sonido para la activación del reto, se diseñaron varias capaz de sonidos sintéticos mediante el uso de los siguientes instrumentos virtuales: Absynth, FM8 y Massive. Además, se incluyó un sonido con efecto de reversa correspondiente al rompimiento y colisión de objetos de vidrio dentro de una bañera-ducha con tiempo de reverberación alto.



*Figura 39. Colisión de pecera de vidrio dentro de un espacio con reverberación alta, Imagen propia*

➤ *Fx\_012 al Fx\_21*

En el caso de la orca, se tuvo en cuenta el uso de librerías online debido a la complejidad de tener una grabación directa sobre este tipo de animal. Una vez obtenido el audio correspondiente, se procedió a realizar variaciones, con el uso de sintetizadores, en pitch, reverberación y modulaciones en amplitud para lograr un sonido característico. Se incluyeron capas de golpes a diferentes objetos para el efecto de mordisco de dicho animal.

➤ *Fx\_022*

Para la avería del propulsor, no se enfocó como tal en el objeto, sino en las cosas que están ocurriendo alrededor de este objeto. Para eso se procedió a incluir capas de audio correspondientes a: Explosiones, tornillos, motores, agua, orcas, burbujas, Woosh y switch, colisión de objetos. En el caso de la destrucción del propulsor se grabó explosiones con diferentes objetos como bombas, bombo de batería, tablas y tornillos, objetos de metal colisionando entre sí. En el caso del motor se cargó el audio en una CDJ 850 de Pioneer con el fin de disminuir el pitch de este audio sin la activación de la función Master Tempo y un rango de pitch amplio, lo cual generó el efecto de lentitud en el motor.





Figura 40. Grabación de objetos de vidrio colisionando contra el suelo, Imagen propia



Figura 41. Grabación de colisión de tornillos y objetos de metal, Imagen propia

➤ *Fx\_023*

Para lograr el sonido deseado con respecto al poder de uno de los personajes principales, se procedió a realizar la grabación de choques de agua contra paredes y suelos, se adicionaron audios de movimiento de agua, ruido blanco, movimiento de piedras y sonidos sintéticos y Soundscapes realizados mediante el software Absynth.



Figura 42. Grabación de movimientos de agua, Imagen propia

➤ Fx\_024 a Fx\_026

Con respecto a la red metálica se grabó el golpe a varios objetos como: cuerdas de guitarra eléctrica, tensores de poste, cercas metálicas, columpios y puertas de metal. Adicionalmente se incluyeron sonidos sintéticos por medio del software FM8 para la creación de los golpes a la red metálica dentro del audio juego.



Figura 43. Grabación de objetos, rejas, y cuerdas de metal, Imagen propia

➤ Fx\_027 a Fx\_037

Los pasos entre agua se realizaron dentro de una ducha en las instalaciones de la universidad, debido a que el tiempo de reverberación que genera este recinto es el indicado para generar la percepción del espacio deseado dentro del audiojuego.



*Figura 44. Pasos en agua dentro de una ducha, Imagen propia*

➤ *Fx\_038 a Fx\_048*

El braceo debajo del agua correspondiente al jugador principal, se realizó mediante la introducción de un micrófono protegido con un condón a un recipiente con agua mientras se hacía el respectivo movimiento con la mano.



*Figura 45. Grabación en recipiente con agua, Imagen propia*

➤ *Fx\_050*

En el caso de las cadenas de Ekko, se realizaron grabaciones a 3 tipos de cadenas diferentes en 2 ambientes diferentes. Una vez obtenida las grabaciones, se realizó la mezcla con la inclusión de algunos efectos.



Figura 46. Grabación de Cadenas en ambiente 1, Imagen propia

➤ Fx\_055 a Fx\_064

Los pasos en piedra se realizaron con el Foley pit de piedras y ladrillos de la universidad San Buenaventura Bogotá

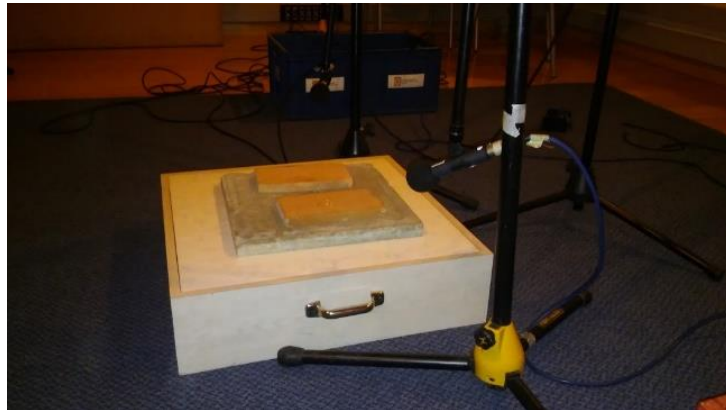


Figura 47. Grabación de pasos en el Foley pit de ladrillo, Imagen propia

➤ Fx\_070 a Fx\_075 y Fx\_101

En los audios de pisada de bichos correspondientes a los Fx\_70 al Fx\_75, se desarrollaron mediante la grabación de whoosh y swish (Efectos de transición rápida de sonido) para el efecto del movimiento del pie, para dicho efecto se utilizaron herramientas tales como: raqueta de tenis y una vara de plástico, cuerdas y cauchos. Adicional, se agregó una capa de grabaciones conformadas por sonidos “húmedos”, creados mediante la colisión de toallas y papel periódico mojado contra una lámina de granito, con el fin de crear el efecto de un bicho aplastado.



Figura 48. Grabación de wosh mediante cuerdas de lana y de caucho, Imagen propia



Figura 49. Grabación de splash con periódico y trapos mojados, Imagen propia

➤ Fx\_76 a Fx\_85

Estos efectos corresponden a los animales que se encuentran dentro del mundo de Dante. Los efectos Fx\_76 a Fx\_81 corresponden a los insectos de la primera holeada, creados mediante la grabación de perros y gatos caminando en baldosa, ladridos, canto de aves y gritos de voz humana con alteración en el pitch mediante

sintizadores, movimientos de arena y sintetizadores LFO para el efecto de aleteo de los insectos.



*Figura 50. Grabación de pasos de perro, Imagen propia*

Para los efectos Fx\_82 a Fx\_84 corresponden a los sonidos de ranas, en este caso el croar de las ranas fueron obtenidas mediante librerías online, estos audios fueron alterados con sintetizadores virtuales. Estos fueron complementados con whoosh y swish como efecto del salto de la rana.



*Figura 51. Grabación de efecto de serpiente mediante palo de lluvia, Imagen propia*

El Fx\_85 corresponde a la serpiente, dicho audio se realizó con herramientas como: palo de lluvia para el efecto de cascabel de la serpiente, ruido blanco como base del rugido de la serpiente, voz humana con uso de efectos virtuales para el rugido de la serpiente y con capas de movimiento de tierra para el efecto de desplazamiento de la serpiente.

➤ *Fx\_86 y Fx\_87*

El efecto de Telekinesis está compuesto de varias capas de audio tales como: movimiento de ramas, rasgadura de cortezas de árbol, caída y colisión de piedras y movimiento de tierra; Estos audios representan el movimiento de los árboles al ser desprendidos del suelo. Por último, se adicionaron un grupo de tracks que conforman el efecto de sonido mágico de la activación del poder de dante, este efecto fue creado mediante el uso del sintetizador Absytnh.



*Figura 52. Movimiento de hojas y rasgadura de tronco, Imagen propia*

➤ *Fx\_88 y Fx\_89*

Para la grabación de la destrucción de los árboles se tuvo en cuenta el uso de láminas de madera y baldosas, con el fin de colisionarlas entre sí y obtener el crujido de éstas al romperse. También se adicionaron el audio de golpes a un bombo, cinta y una tambora para simular el efecto de golpe hacia los árboles.



*Figura 53. Grabación de cinta magnética y láminas de madera, Imagen propia*

➤ *Fx\_90*

El ataque de la onda expansiva del jugador fue creado mediante el uso de sintetizadores virtuales tales como: Absynth, FM8, Massive y Razor.

➤ *Fx\_100*

El *Fx\_100* corresponde a los golpes que recibe el jugador. Este archivo fue creado a partir de golpes a objetos tales como sillas, almohadas, puff y suelo con una vara de madera y cuerdas de caucho, además incluye varias capas de sintetizadores obtenidas de Absynth y massive para dar un sonido más ficticio.



*Figura 54. Grabación golpes con varas de madera y cinta de caucho, Imagen propia*

NOTA: En el anexo D se puede encontrar la tabla de los audios usados en el audiojuego, allí se podrá apreciar nombre del archivo, el audio implementado y el escenario en el que se usa.

### **7.1.6. Acciones corporales**

Las acciones corporales que se van a usar para interactuar con el audiojuego en el caso del objetivo tres (3), se denominarán desde ahora como gestos y se fundamentan en el reconocimiento de partes específicas del cuerpo ejecutando un movimiento determinado. La librería de Kinect para Unity que se usó realiza el tracking de las siguientes partes corporales.





Figura 55. Parámetros Cubeman, Imagen propia

Apenas se realiza el tracking, estas partes toman automáticamente su ubicación en el avatar que provee la librería.

Los gestos que se pueden reconocer pueden ser infinitos ya que la librería usada, es de código abierto y permite crear nuevos gestos según las necesidades.

Para este prototipo de audio juego interactivo se escogieron 9 gestos que desempeñan diferentes funciones, dependiendo de lo que el videojuego requiera en determinado momento, este requerimiento es retro alimentado por la voz guía hacia el usuario.

- **Tpose:** Los brazos deben ser extendido hacia los lados, perpendicularmente al cuerpo durante 1 segundo.

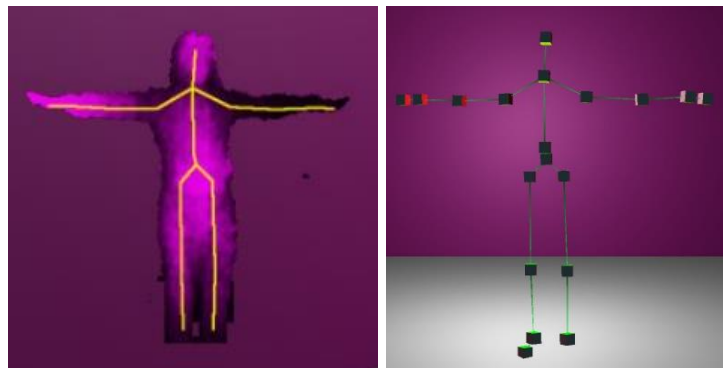


Figura 56. Posición Tpose, Imagen propia

- **PSI:** Ambas manos deben estar por encima de los hombros por 1 segundo

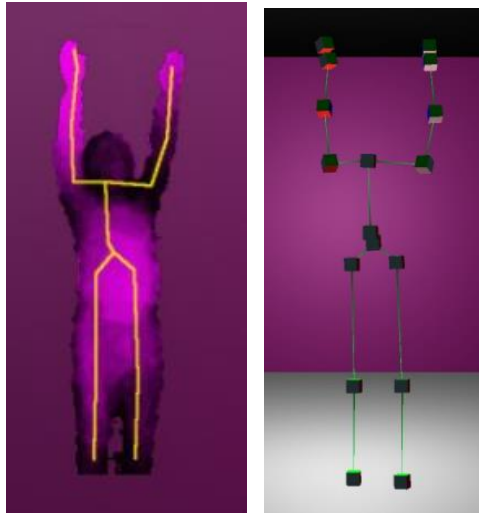


Figura 57. Posición PSI, Imagen propia

- **RaiseRightHand / RaiseLeftHand:** La mano derecha o la mano izquierda debe permanecer por encima del hombro por 1 segundo.

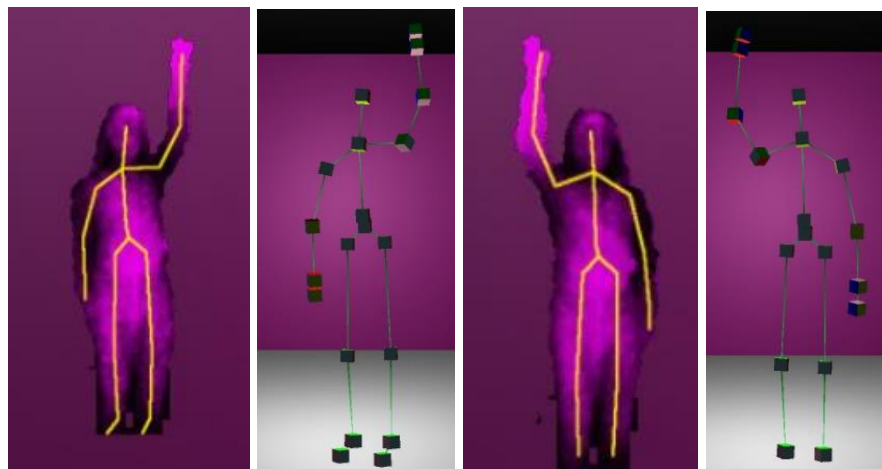


Figura 58. Posición LeftRightHand - RaiseRightHand, Imagen propia

- **Push:** Estirar el brazo izquierdo o derecho hacia al frente durante 1.5 segundos

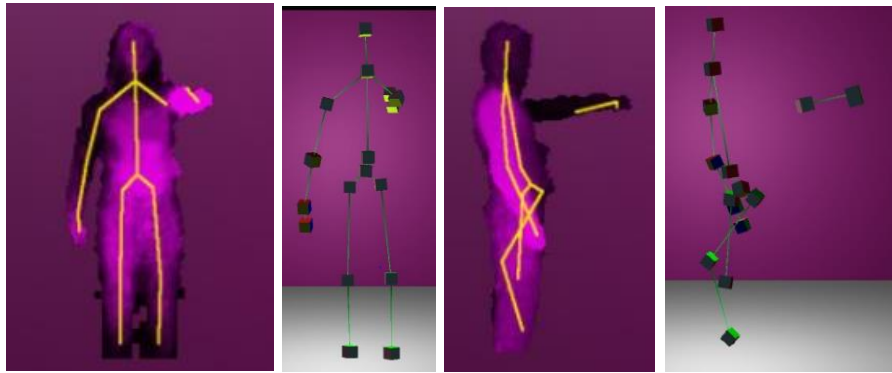


Figura 59. Posición Push vista frontal y vista lateral, Imagen propia

- **SwipeLeft – SwipeRight:** Deslizar la mano derecha hacia la izquierda o la mano izquierda hacia la derecha

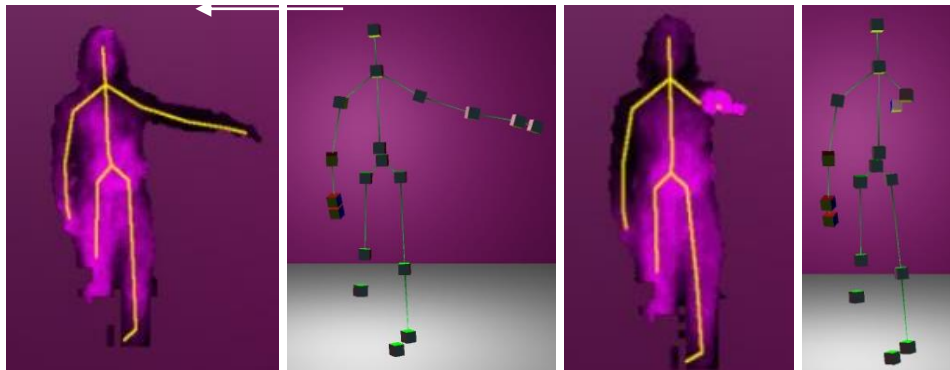


Figura 60. Posición SwipeLeft, Imagen propia

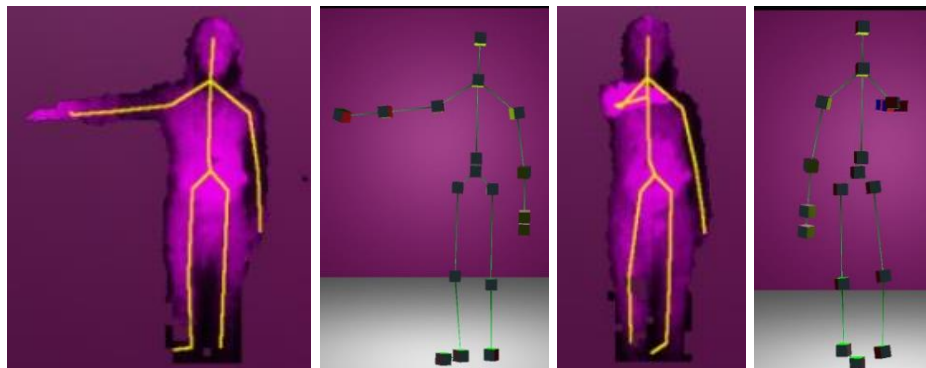


Figura 61. Posición SwipeRight, Imagen propia

- **Squad:** La cintura debe estar 15 centímetros por debajo de la última posición hace 1.5 segundos.

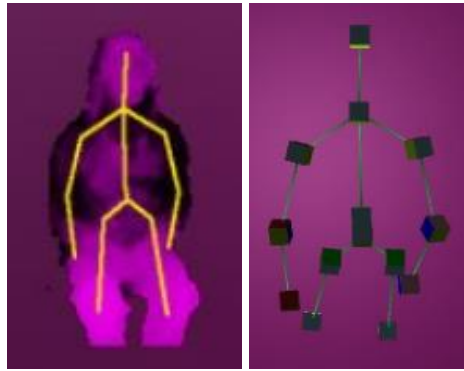


Figura 62. Posición SwipeLeft, Imagen propia

- **Jump:** La cintura debe estar 15 centímetros por encima de la última posición hace 1.5 segundos

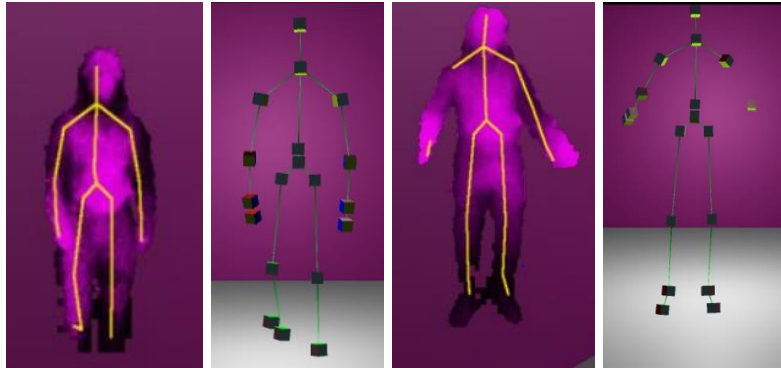


Figura 63. Posición Jump, Imagen propia

- **StepRight – StepLeft:** Levantar la rodilla derecha o izquierda 20 centímetros

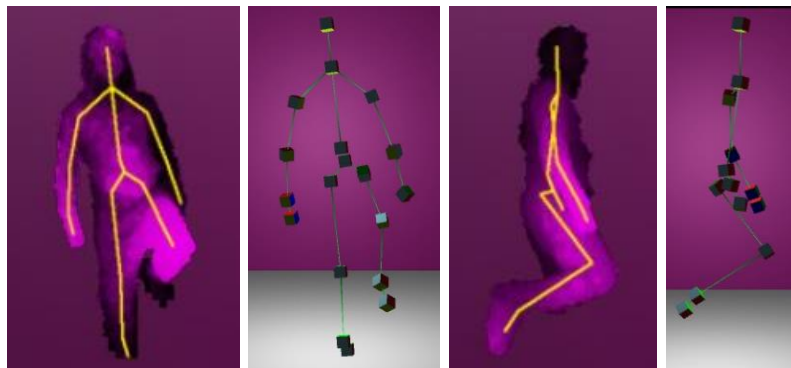


Figura 64. Posición StepRight en vista frontal y lateral, Imagen propia

### **7.1.7. Diseño del software**

#### **7.1.7.1. Elementos usados**

A continuación, se presenta las versiones de los softwares y licencias implementados para el desarrollo del audiojuego:

- Reaper 5, Licencia 30 días, Vencida
- Drivers Kinect 1.8
- Kinect For Windows SDK v1.8
- Kinect For Windows Runtime v1.8
- FMOD (Non Commercial/Educational) Licencia Gratuita para la mayoría de las personas, el precio depende del proyecto y su perfil.
- Unity 5 Pro, Licencia gratuita para todos los desarrolladores, se paga cuando se desea comercializar el producto.
- Waves Plugins - Licencia de prueba 30 días, Vencida
- Izotope Licencia de prueba 30 días, Vencida
- Absynth 5 Licencia de prueba 30 días, Vencida
- Protools 10.8: Licencia adquirida por la Universidad
- Reaktor Razor 5 Licencia de prueba 30 días, venida
- Plugins atk gratuitos

#### **7.1.7.2. Desarrollo en Unity**

El desarrollo en Unity se basa principalmente en las características del juego, en este caso es proporcionar al usuario una forma alternativa de jugar, donde su interfaz de usuario será audible por medio de un sistema de reproducción Ambisonics, en vez de un sistema gráfico y visual, también se proporcionara al jugador una interacción por medio de una interfaz natural de usuario, en donde se podrá controlar el juego por medio de gestos y movimientos corporales sin necesidad de usar algún dispositivo físico clásico (teclado, ratón, etc...) en la instalación.

Una vez las características del juego son claras, se procede a realiza un planteamiento de las herramientas que van a ser necesarias para desarrollar este prototipo de audio juego, en este caso esas herramientas van a ser FMOD (Versión 1.08.08) que estará encargado de la reproducción y decodificación de audio para un sistema de reproducción Ambisonics de primer orden por medio del Plug-in Oculus Spatializer, y para la interacción del usuario por medio de gestos y

movimientos corporales, se va a usar el Kinect v1.0 por medio del “Kinect With MS-SDK”, un paquete gratuito que se encuentra en el Asset Store (Tienda virtual) de Unity, el cual brinda un reconocimiento de movimientos y gestos limitados, además de proporcionar un recreación virtual del usuario en la parte gráfica para controlar un avatar.

### 7.1.7.3. Diseño del audio juego en Unity

Desde que el audio juego fue planteado en el guion técnico y literario se dejaron claros cuales iban a ser los personajes, los ambientes, las acciones y las funciones por nivel con las que el usuario iba a tener que interactuar durante todo el audio juego, es por eso que principalmente se hizo un diseño lógico de funcionamiento, que depende de la interacción del usuario con el juego, además se realizó un diseño básico para la recreación de ambientes en la parte gráfica, con el propósito de poder obtener de manera más rápida una retroalimentación del comportamiento del jugador en el juego al momento de diseñarlo.

Todos los niveles del audio juego cuenta una configuración básica de funcionamiento:

#### **Audio**

Al crear el proyecto 3D en Unity, se debe realizar la integración correspondiente de FMOD con Unity, y además de esto se deben cargar los siguientes scripts para culminar la integración:

Asignación de componente de especialización de audio en un objeto nativo del proyecto y de la escena (Main Camera) para poder escuchar los otros objetos que están alrededor del espacio virtual.

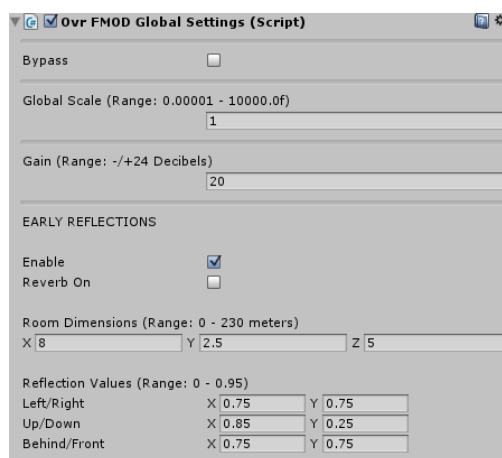


Figura 65. Plugin Ovr Gobla Settings, Imagen propia

De igual manera se agrega y activa el Studio Listener que provee FMOD, este componente actúa parecido a un micrófono, pues recibe lo que es reproducido por las fuentes de audio en el proyecto de unity, para después reproducirlo por el sistema de reproducción Ambisonics en este caso.



Figura 66. Studio Listener, Imagen propia

Es agregado en el objeto “Head” para hacer que la escucha sea lo más natural posible ya que este objeto será la cabeza del usuario.

#### 7.1.7.4. Kinect

Los Scripts usados para el reconocimiento, son los que brinda la librería “Kinect With MS-SDK”, en donde únicamente se accedió al código para activar reconocimientos de gestos, para agregar el gesto de caminar basado en la misma lógica y para cambiar el valor de una variable propia de la cual se pudiera tener acceso público en otros Scripts

El Script “Kinect Manager” es el componente general necesario para poder interactuar con el sensor y así poder obtener los datos que el sensor registra del espacio, la profundidad, los cuerpos, la posición de las articulaciones corporales. Este componente es agregado en un objeto nativo del proyecto y de cada escena.

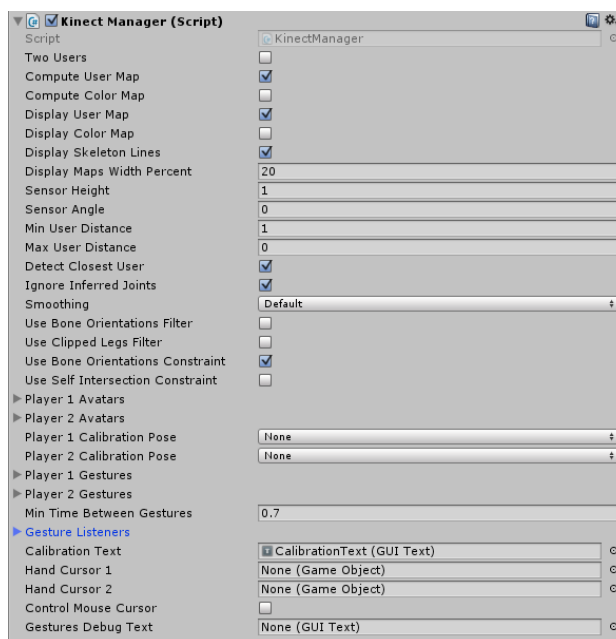


Figura 67. Parámetros Kinect Manager, Imagen propia

Para complementar el componente Kinect Manager y poder visualizar la posición de las articulaciones y sus movimientos, se agrega un objeto llamado “Cubeman” que cuenta con 20 sub-objetos (Hijos) que están nombrados como cada una de las articulaciones que los sensores reconocerán.



Figura 68. Complementos CubeMan, Imagen propia

Por medio del script “Cubeman Controller” agregado al objeto anterior, se procede a hacer la asignación de cada una de las articulaciones al script para que este realice el Tracking y haga la representación de cubos y líneas de las articulaciones y el esqueleto del usuario.

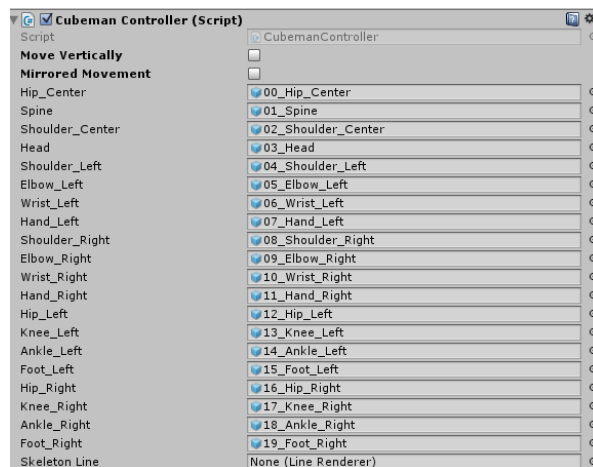


Figura 69. Componente CubeMan Controller, Imagen propia



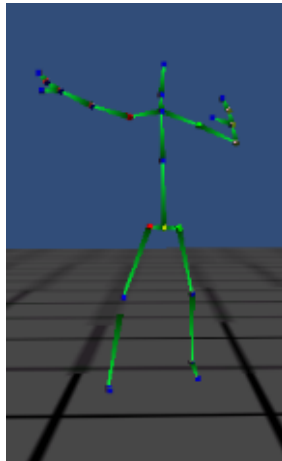


Figura 70. Representación de cubos y líneas del usuario, Imagen tomada de: [Psychic-vr-lab.com](http://Psychic-vr-lab.com)

Adicional a esto se desea reconocer gestos o movimientos corporales como acciones relevantes en el juego, para esto se agrega el componente “Gesture Listener” que estará encargado de reconocer movimientos y realizar alguna acción o cambio en el código del juego. Se agrega en un objeto nativo de cada escena del proyecto.



Figura 71. Componente Gesture Listener, Imagen propia

El objeto Gesture Info que contiene el componente Gesture Listener, este será el encargado de mostrarnos el gesto realizado en el display del juego.



Figura 72. Avatar Kinect, Imagen tomada de: [Psychic-vr-lab.com](http://Psychic-vr-lab.com)

En este componente están establecidos los gestos que va reconocer el juego, en este caso son 8, en donde cada uno de los movimientos cambia el valor de la variable “pose”. Para que de esta manera se puedan realizar acciones según como interactúe el usuario, el valor que toma la variable, es la que se puede ver en color verde en la siguiente imagen.

```
public int pose = 0;

private bool swipeLeft; //(1)
private bool swipeRight; //(1)
private bool psi; //(2)
private bool rh; //(3)
private bool lh; //(4)
private bool push; //(5)
private bool jump; //(6)
private bool sq; //(7)
private bool kr; //(8)
private bool kl; //(8)
```

Figura 73. Script Gesture Listener, Imagen propia

Cada una de estas variables indica respectivamente lo siguiente:

1. swipeLeft / swipeRight = Deslizar a la izquierda o deslizar a la derecha.
2. psi = Levantar ambas manos.
3. rh / lh = Levantar mano derecha o levantar mano izquierda.
4. push = Puño izquierdo o derecho al frente.
5. jump = Saltar.
6. sq = Agacharse.
7. kr / lr = levantar rodilla derecha o izquierda.

La manera en que se hace el reconocimiento de este gesto o movimiento no solamente depende del script “Gesture Listener”, este script únicamente recibe información lógica del script “Kinect Gesture”, el cual está encargado de verificar si el gesto o movimiento se realiza de manera correcta o no. A continuación se explicara la lógica con que se desarrolló el gesto “kr”, la cual es muy similar para el resto de gestos que la librería brinda.

```

case Gestures.kright:
    switch (gestureData.state)
    {
        case 0: // gesture detection
            if (jointsTracked[rightkIndex] && jointsTracked[hipCenterIndex] &&
                (jointsPos[hipCenterIndex].y - jointsPos[rightkIndex].y) <= 0.35f)
            {
                SetGestureJoint(ref gestureData, timestamp, rightkIndex, jointsPos[leftkIndex]);
                gestureData.progress = 1.0f;
            }
            break;

        case 1: // gesture complete
            if ((timestamp - gestureData.timestamp) <= 1.5f)
            {
                bool isInPose = jointsTracked[rightkIndex] && jointsTracked[hipCenterIndex] &&
                    (jointsPos[hipCenterIndex].y - jointsPos[rightkIndex].y) <= 0.35f;

                Vector3 jointPos = jointsPos[gestureData.joint];
                CheckPoseComplete(ref gestureData, timestamp, jointPos, isInPose, 0f);
            }
            else
            {
                SetGestureCancelled(ref gestureData);
            }
            break;
    }
    break;

```

Figura 74. Código detección de gestos, Imagen propia

Cada uno de los gestos está dividido por casos, para este será respectivamente el caso “kright”, en donde accedemos al estado de los datos del gesto para poder realizar la detección. Inicialmente se presenta un condicional en donde se verifica si la rodilla derecha se encuentra a una distancia menor de 0.35 [m] en el eje Y (Altura). En caso de que se cumpla el condicional se procederá a asignarle valores a las variables de posición y tiempos en la función “SetGestureJoint”

```

private static void SetGestureJoint(ref GestureData gestureData, float timestamp, int joint, Vector3 jointPc
{
    gestureData.joint = joint;
    gestureData.jointPos = jointPos;
    gestureData.timestamp = timestamp;
    gestureData.state++;
}

```

Figura 75. Función SetGestureJoint, Imagen propia

En esta función aparte de asignar valores a las variables, es necesario resaltar que los registros de tiempo son de vital importancia para poder analizar y controlar comportamientos internos de los objetos. Continuando con la explicación del código, una vez se hacen los registros de valores, se procede a comprobar por medio de un caso con un condicional si el gesto se completa en un tiempo menor a 1.5 segundos por medio de la diferencia que hay en el registro actual de tiempo con el tiempo

registrado previamente en la función “SetGestureJoint”. Al cumplirse este condicional se procede a verificar de manera lógica si el gesto se encuentra en pose (true) o no (false), al encontrar que el gesto está en pose se procede a enviar la variable lógica “true” a la función “CheckPoseComplete” en donde nuestro script “Gesture Listener” tendrá acceso para indicar en el display si el gesto o movimiento se realizó correctamente, de lo contrario, se enviara un falso a la función “SetGestureCancelled” que se encargará de borrar cada uno de los valores previamente guardados.

Al tener todo lo anterior configurado correctamente, se procede al diseño y desarrollo del juego en unity como tal. Se cuenta con diferentes materiales para diferenciar los objetos en la parte gráfica:

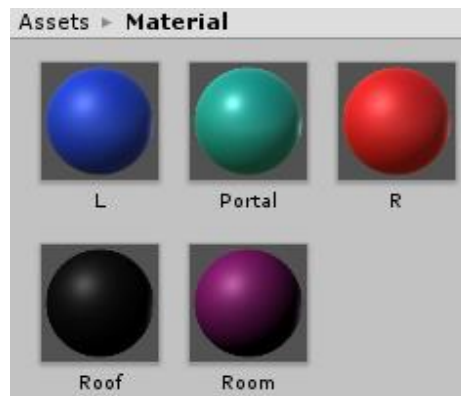


Figura 76. Materiales creados en Unity, Imagen propia

Adicional a la integración en de FMOD con unity y a la detección de gestos por medio del Kinect, las escenas presentan objetos en común en la parte gráfica, que son:

1. Point light: Recrea un punto de luz que permite visualizar los objetos que hay en la escena.
2. Main Camera: Cámara virtual que se ubica en la escena de unity para permite observar el diseño y comportamiento del juego de manera gráfica.
3. Room: Elemento que se compone por la recreación virtual de un cuarto simétrico compuesto por 4 paredes, techo y suelo. Las paredes son de color morado, y el piso y el techo son de color negro, estos colores simplemente son para ubicarse de manera correcta en el escenario.

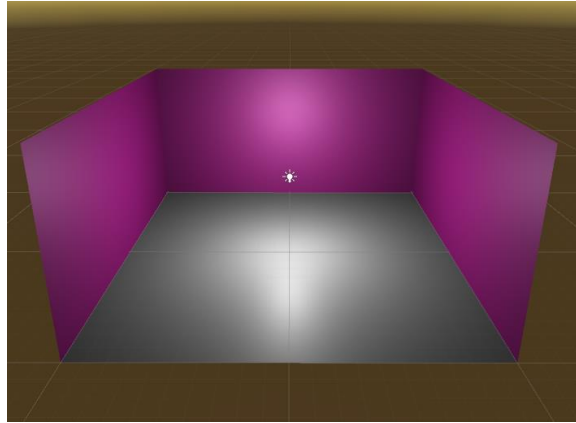


Figura 77. Simulación de ambiente en Unity, Imagen propia

4. AMB: 4 Objetos destinados para ubicar cada ambiente en un lugar en específico, son de color azul y rojo para distinguir entre izquierda (Azul) y derecha (Rojo)

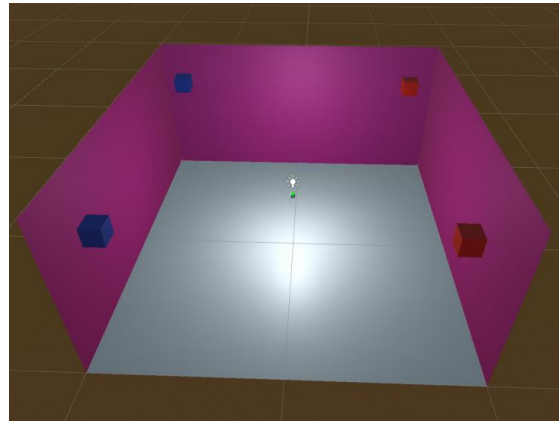


Figura 78. Recreación de espacio en Unity con ambientes ubicados, Imagen propia

Los personajes y elementos varían según lo establecido en el guion técnico y literario. Es por eso que procederá a realizar un análisis detallado de cada nivel. Cabe mencionar en esta parte que el juego fue diseñado para poder seguir la historia hasta que se realice el gesto sin la necesidad de perder, ya que se tuvo en cuenta la dificultad que presenta para algunas personas, y el margen de error que puede presentar la detección de movimientos.

- **Entrenamiento**

Este nivel es para realizar una explicación al usuario de la manera en cómo se debe interactuar con el juego por medio de los gestos según lo que la voz del personaje Guía vaya indicando.

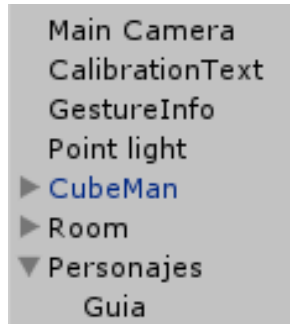


Figura 79. Objetos Nivel Entrenamiento, Imagen propia

Los eventos que se reproducirán son el 44,45 y 6 los cuales irán guiando al usuario, para poder avanzar en el juego se debe seguir las indicaciones que la voz guía indicara.

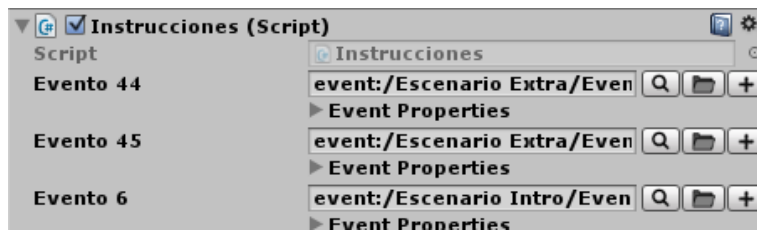
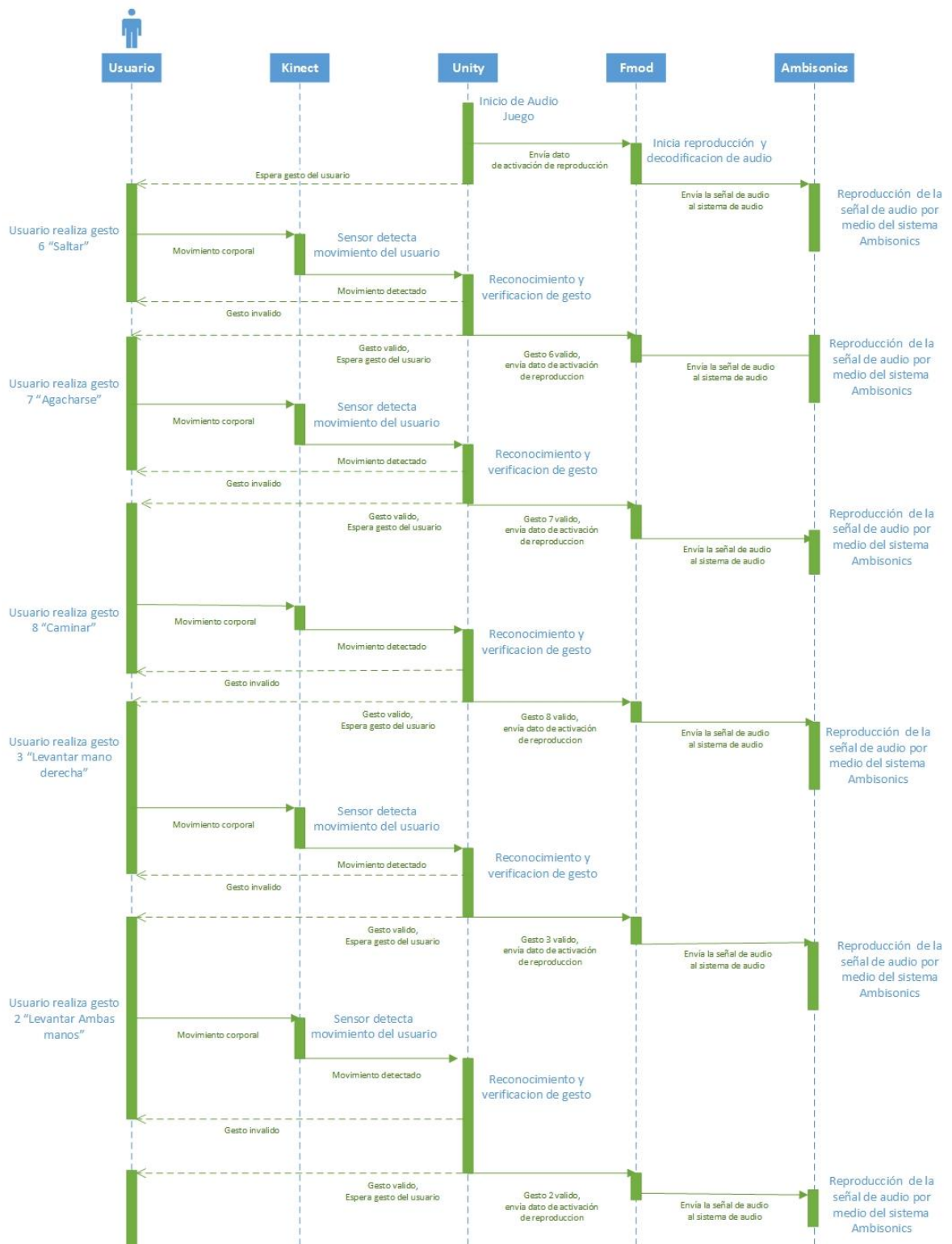


Figura 80. Script instrucciones en el Nivel Entrenamiento, Imagen propia

En el Script Instrucciones aparte de contener los eventos que se van a reproducir también se puede observar el funcionamiento lógico según la interacción del usuario con el juego que será explicado en el siguiente diagrama de secuencia.

## Diagrama de Secuencia Nivel 1 "Instrucciones"



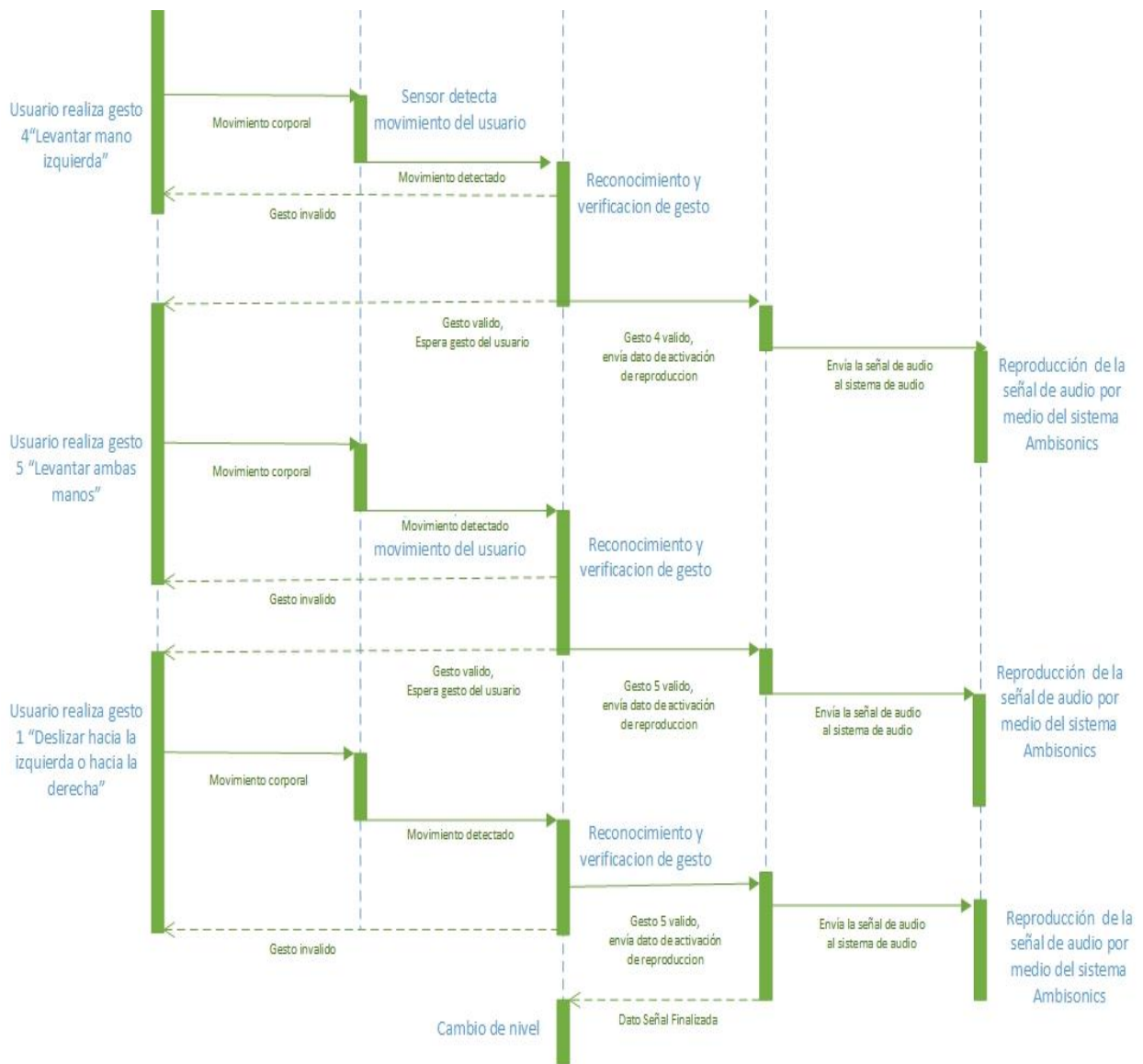


Figura 81. Diagrama de secuencia Nivel de entrenamiento, Imagen propia



- **Introducción**

En el nivel introducción, hay un breve relato para contextualizar el usuario con el juego, recibirá información del personaje principal Ekko y recibirá ayuda del personaje Guía

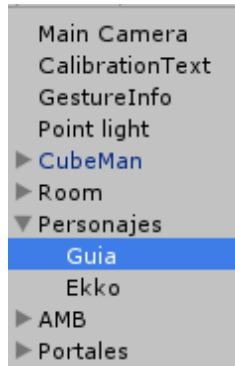


Figura 82. Objetos escenario Introducción, Imagen propia

Los eventos 5,6 y 42 que son los que se reproducirán y la parte lógica del comportamiento en este escenario se encuentra en el script “Introducción”.

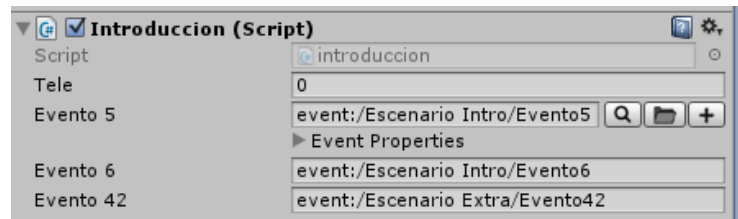


Figura 83. Script introducción para el nivel de introducción, Imagen propia

## Diagrama de Secuencia Nivel 1 "Introducción"

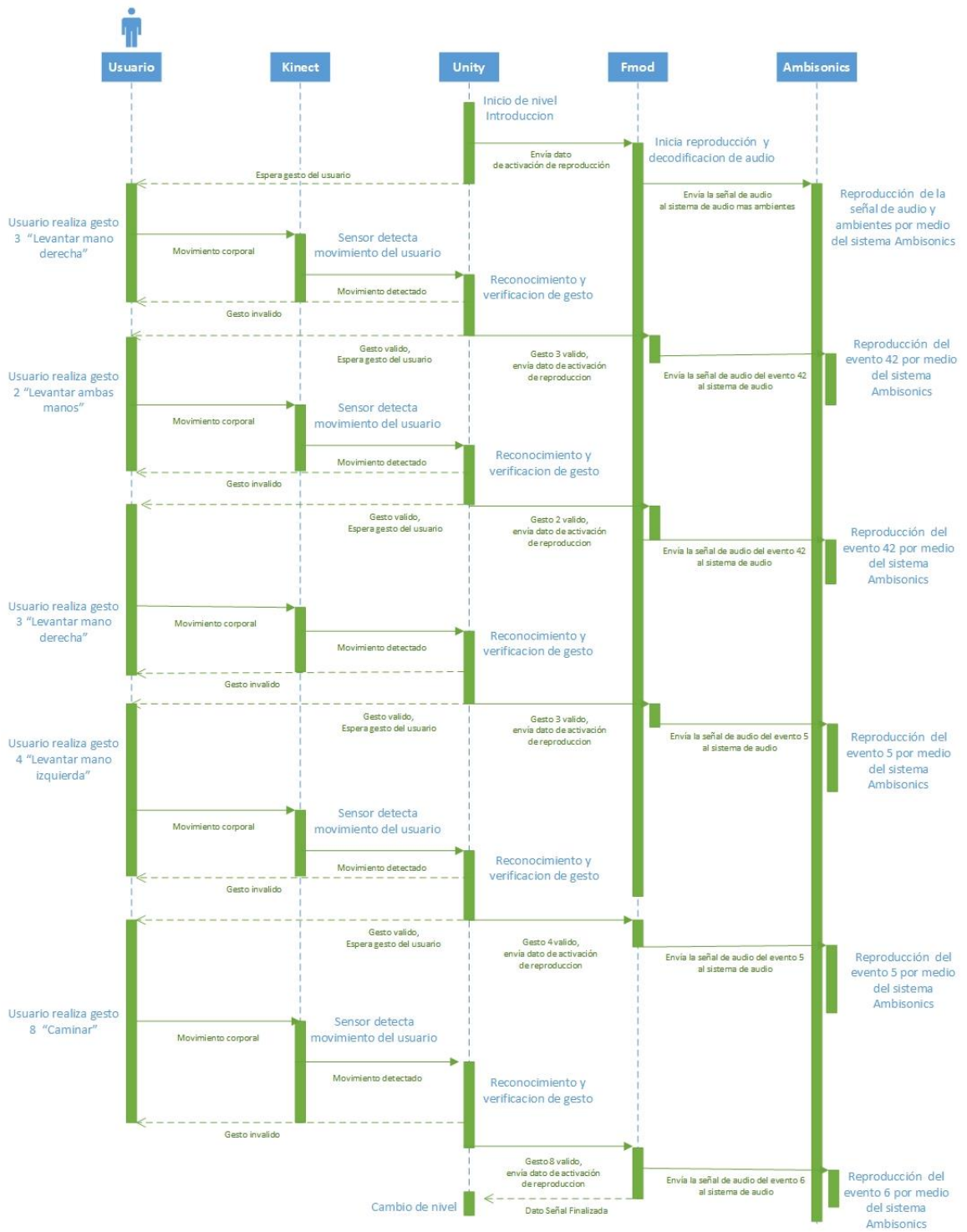


Figura 84. Diagrama de secuencia del nivel de introducción, Imagen propia

- **Yacu**

En este nivel el usuario pondrá a prueba lo aprendido en el nivel instrucciones ya tendrá que estar atento a lo que los personajes le digan y además de todo también habrá presencia sonora de objetos con los cuales tendrá que interactuar ayudado por la voz guía como siempre, tal cual como está especificado en el guion técnico y literario

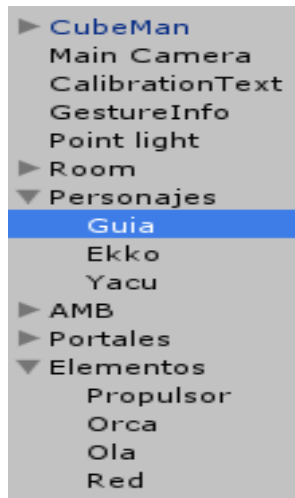


Figura 85. Objetos Nivel Yacu, Imagen propia

Como es posible evidenciar, el usuario tendrá que interactuar con más personajes, objetos y ambientes, es por eso que en el script Yacu están presentes los eventos del 11 al 18 junto con los eventos 41,42,43 que se irán reproduciendo a medida que se avance en el juego.

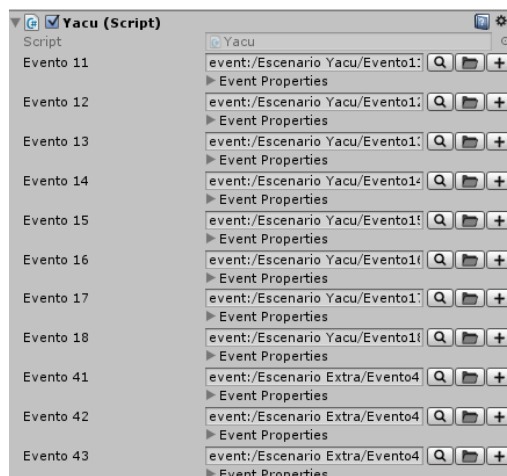
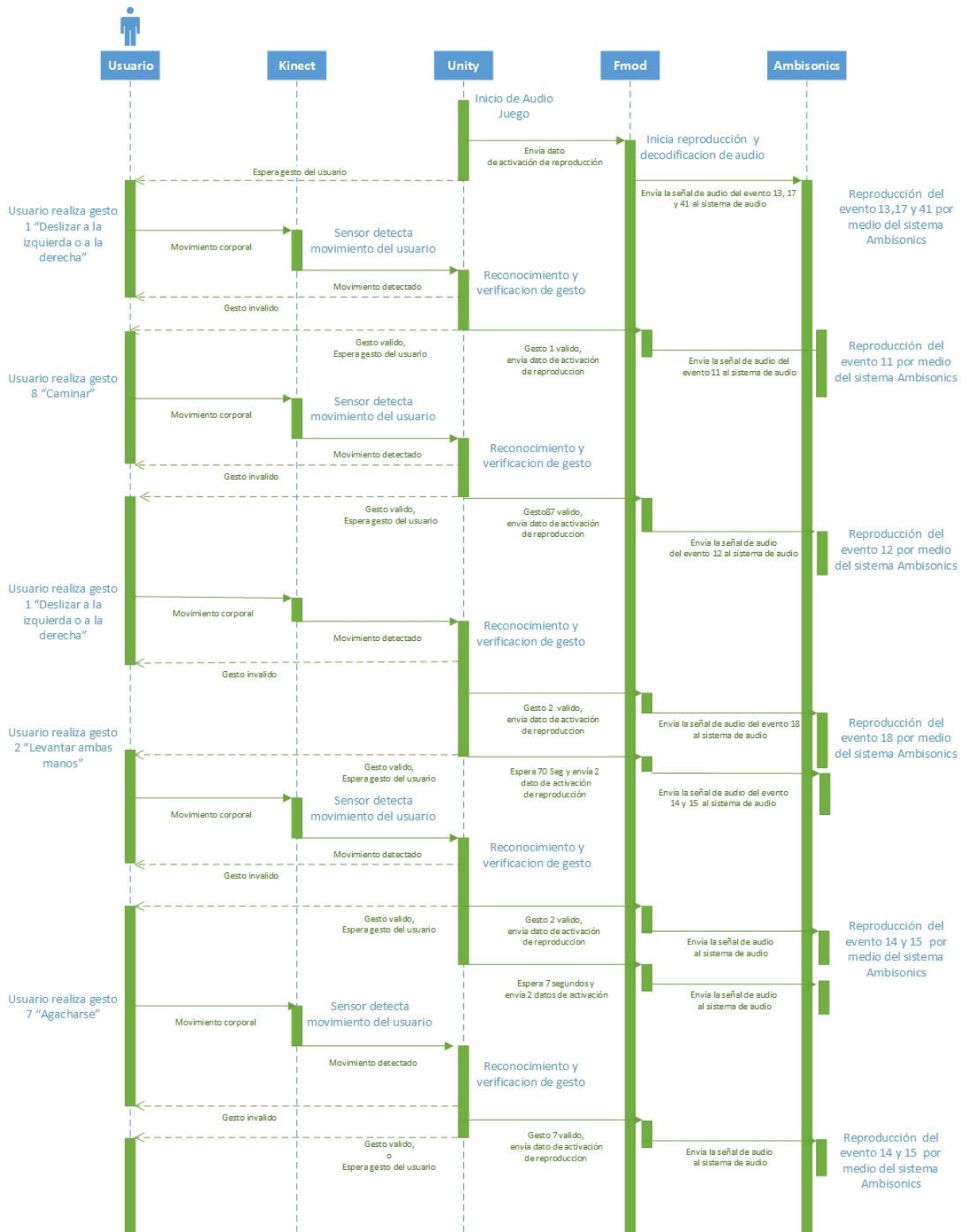


Figura 86. Script Yacu en el nivel acuático de Yacu, Imagen propia

## Diagrama de Secuencia Nivel 2 "Yacu"



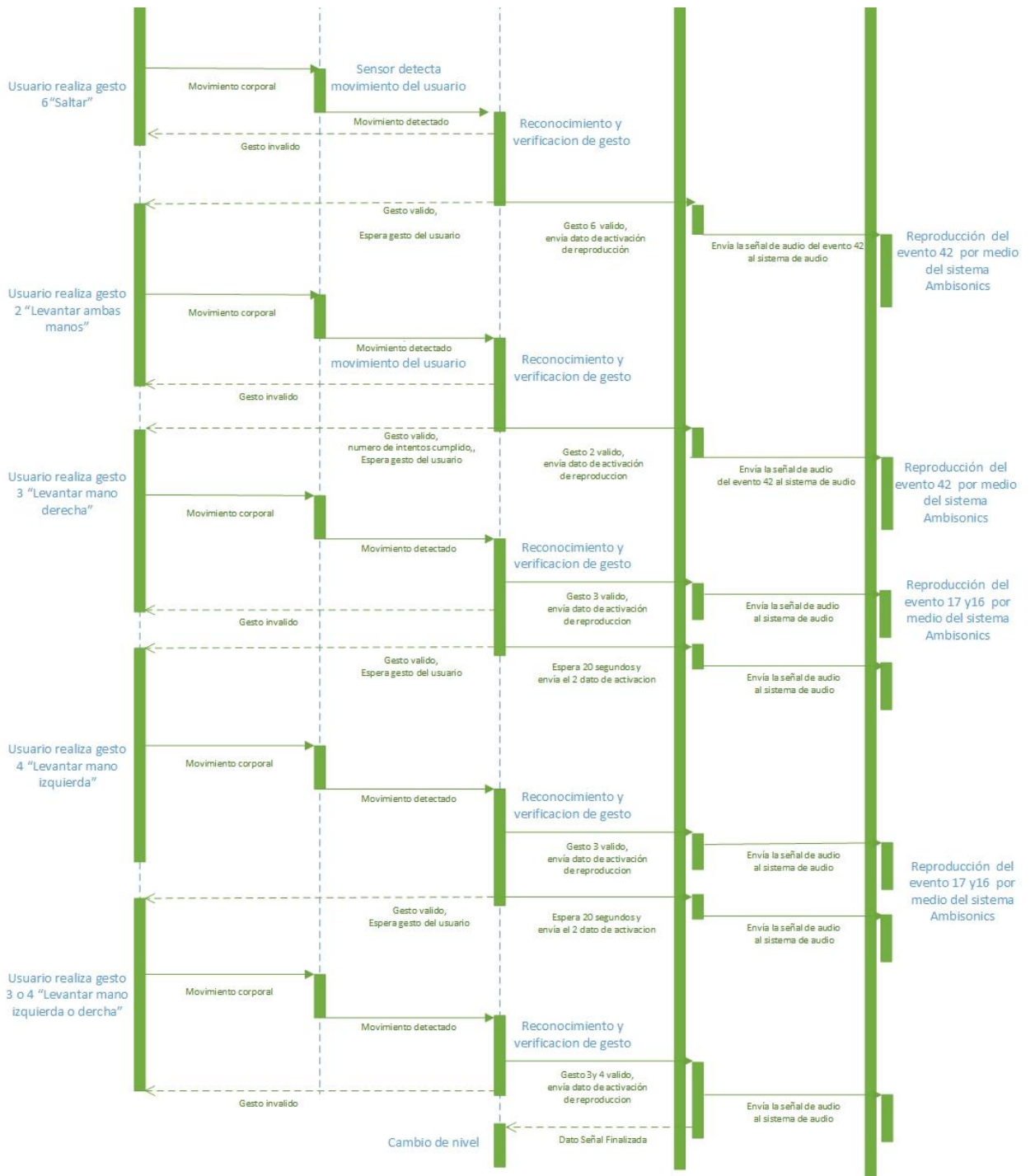


Figura 87. Diagrama de secuencia del nivel Yacu, Imagen propia

- **Dante**

Este es el escenario final, en el que el usuario desafiará sus capacidades al tener que estar atento de todo lo que escuche para reaccionar de manera rápida ya que la reproducción en este último nivel de los eventos subirá su velocidad con el propósito de aumentar la complejidad.



Figura 88. Objetos Nivel Dante, Imagen propia

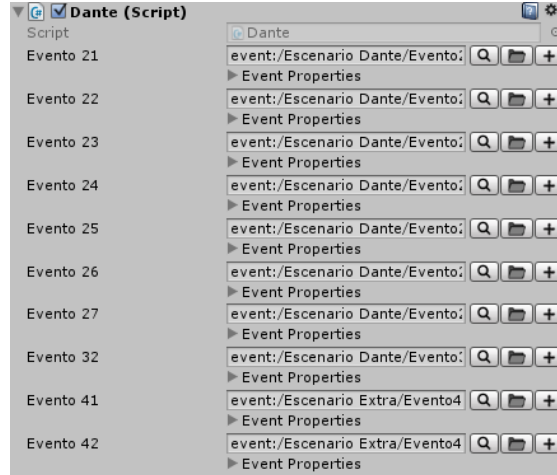


Figura 89. Script Dante en el Nivel selvático de Dante, Imagen propia

## Diagrama de Secuencia Nivel 2 "Dante"

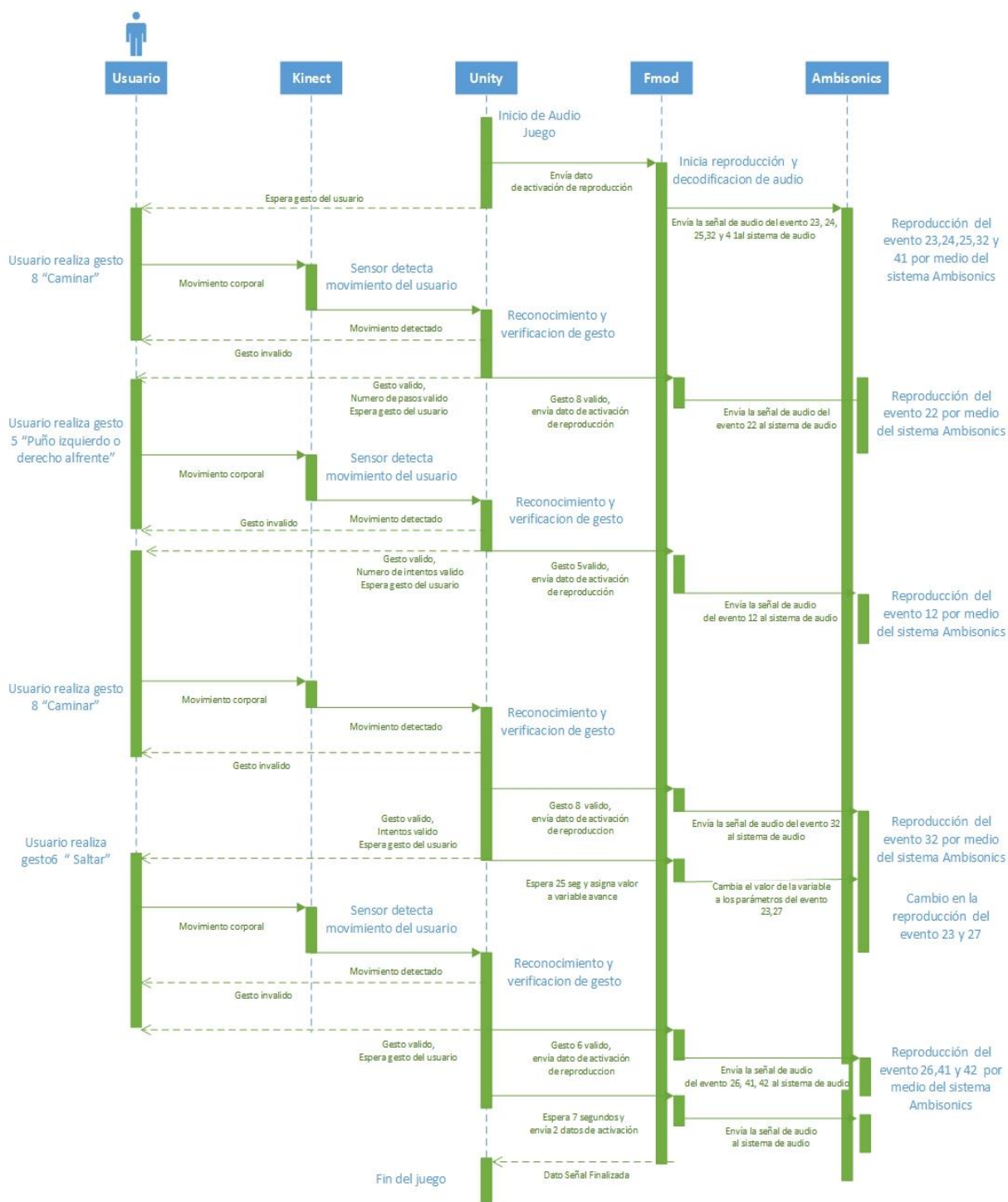


Figura 90. Compatibilidad de algoritmos, Imagen propia

### **7.1.8. Integración FMOD-Unity**

#### **7.1.8.1. Integración**

La Integración se realiza debido a qué es necesario una buena respuesta para un buen rendimiento y funcionamiento del audio juego puesto que se usan dos softwares por aparte, Unity un motor de video juegos que está a cargo de la parte interactiva entre el usuario y la máquina. Por otro lado, FMOD Studio un software middleware dedicado a la programación de audios para la reproducción de eventos, según la interacción que el usuario tiene con la máquina. Para que el audio juego funcione de manera correcta, es necesario hacer la integración de los 2 software, en este caso es necesario que se mantenga actualizada la versión de FMOD STUDIO, que además requiere Unity 4 Pro o Unity 5, para que se presente comunicación entre ambos softwares sin problemas o retrasos. Se sugiere tener todo el contenido del proyecto en una sola carpeta para evitar alguna pérdida de archivos que puedan interferir en la integración de FMOD Studio con Unity.

Se debe seguir una serie de pasos de manera ordenada para realizar la integración de manera correcta, a continuación, se desglosarán y enumerarán para una fácil comprensión:

- 1- Descargar el paquete de integración “FMOD Studio Unity Integración” y mantenerlo actualizado para evitar problemas de funcionamiento. En este paquete se encuentran todas las librerías que requiere Unity (classes, functions, callbacks, structures, defines, enumerations,) para lograr establecer comunicación con FMOD y poder tener control sobre las características y todos los parámetros programados en FMOD para el audio juego. Se encuentra en la página de descargas de FMOD (<http://www.fmod.org/download/>).
- 2- Abrir el proyecto de Unity en el que se ha desarrollado toda la parte Interactiva del audio juego e importar el paquete de integración completo al proyecto, para esto debe ir a la pestaña de Assets en la parte superior, Import Package y dar click en “Custom Package”, Al seleccionar el paquete de importación se abrirá una ventana en donde están seleccionados todos los elementos del Package que va a importar.



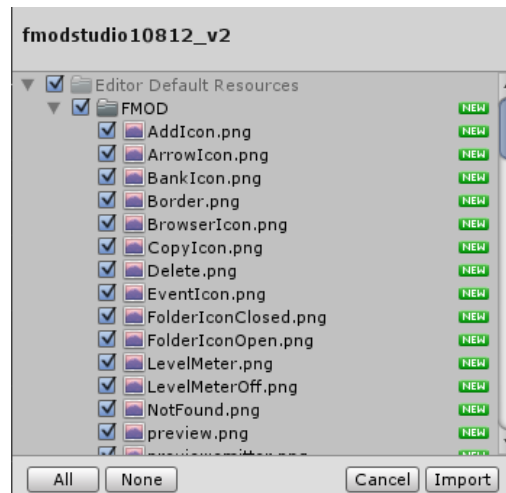


Figura 91. Elementos Fmod Package, imagen propia

Una vez sea importado el paquete de integración, se podrá observar en la barra de herramientas de Unity, el menú de FMOD en el cual podremos tener acceso a la configuración de FMOD Studio, un buscador de eventos, una consola de errores, una herramienta para encontrar y reemplazar archivos del proyecto, ayuda (manuales, documentación, foros, soporte, etc..) e información sobre la versión de integración.

Adicional a esto podrá encontrar Scripts de FMOD que pueden ser agregados en los componentes de los objetos en Unity. Se procederá a asignar el “FMOD Studio Listener” al objeto “Head” de nuestro proyecto, al agregar este Script a este objeto, estará en función de recibir el sonido emitido por los otros objetos en Unity en un espacio 3D, Amplitud – Distancia y Espacialidad. Este componente se asigna en este objeto para buscar el mayor realismo, ya que el Tracking que hará el Kinect para este objeto siempre será la cabeza del usuario en el audio juego. Además, También se agregará el componente “Ovr FMOD Global Settings” que se encargará de inicializar parámetros generales para la escena.

- 3- Antes de importar el proyecto de FMOD STUDIO a Unity, se debe verificar en FMOD que todos los eventos programados hayan sido asignados al banco del proyecto, en este caso se hizo uso el “Master Bank” que FMOD establece por defecto. En este banco se incluirán todos los eventos que interfiera de manera significativa el audio juego.

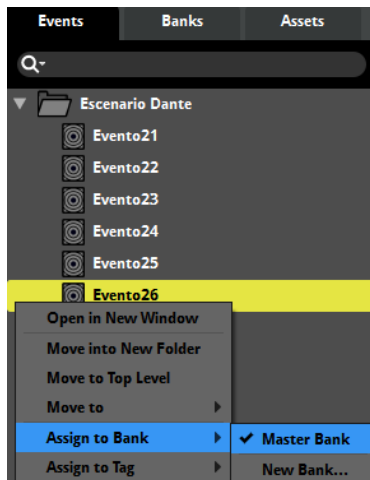


Figura 92. Asignación de eventos, imagen propia.

- 4- Una vez asignados todos los eventos que van a ser usados al “Master Bank”, se procederá a realizar el “Build” en el menú “Fille” de FMOD, esto es para convertir todos los eventos activos en “Master Bank” a Audio Assets que puedan ser cargados y usados en el juego. Esta conversión la realizara para el tipo de plataforma que se esté usando, en este caso PC.
- 5- Una vez realizados todos los pasos anteriores, se procederá a importar el archivo generado por el Build a Unity. para esto se debe acceder al menú de FMOD en Unity, escoger la opción “Edit Settings “y en la pestaña que se abre en el Inspector de Unity, se selecciona el botón “Project” y después se buscara en la opción “Studio Project Path” el archivo con extensión “. fspro” para importar los Assets generados por FMOD.

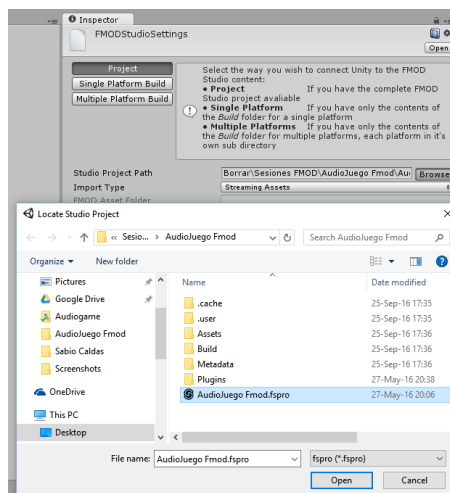


Figura 93. Importar Assets, imagen propia.

Apenas se haya escogido la ruta del archivo, se podrá observar en el “Event Browser” todos los eventos asignados al “Master Bank”, que posteriormente podrán ser asignarlos al objeto que funcionará como fuente sonora.

### 7.1.8.2. Decodificación y programación de señales

El audio juego está compuesto por un total de 36 eventos, los cuales están distribuidos en diferentes carpetas, cada carpeta representa un escenario en el audio juego.

En el anexo B se encuentra la tabla de eventos, en ella se puede observar el número total de eventos, su distribución por niveles, su categoría (Evento simple – Evento multitrack) y el contenido de cada evento.

Para entender mejor la organización de los eventos en este proyecto, se estableció cinc (5) categorías; Estas categorías fueron diseñadas en base al modo de trabajo de los eventos.

CATEGORIA	EVENTOS INCLUIDOS
Narración	5 - 17 - 32 - 44
Natural	6 - 11 - 12 - 18 - 21 - 22 - 23 - 24 - 26 - 27 - 42 - 43 - 45
Ambiente 1	1 - 2 - 3 - 4
Ambiente 2	9 - 10 - 16 - 19 - 20 - 28 - 29 – 30 - 31
Compuesto	13 - 14 - 15 - 16 - 25 - 41

Tabla 1. Categoría de eventos en FMOD Studio

- **Eventos de Narración**

Estos eventos corresponden a las instrucciones y a los diálogos de cada nivel en el audio juego.

EVENTO	CONTENIDO
5	EKKO – GUIA
17	EKKO – YACU -
32	EKKO - DANTE
44	GUIA

Tabla 2. Eventos de narración

En el caso de los eventos 5, 17 y 32 tienen como adición un track de audio con efectos que complementan la narrativa del juego. Dichos tracks de audio no influyen en la operación lógica de cada evento.

Para poder entender la operación lógica y modo de trabajo, se tomará como ejemplo uno de los eventos mencionados anteriormente, en este caso el *evento5*.

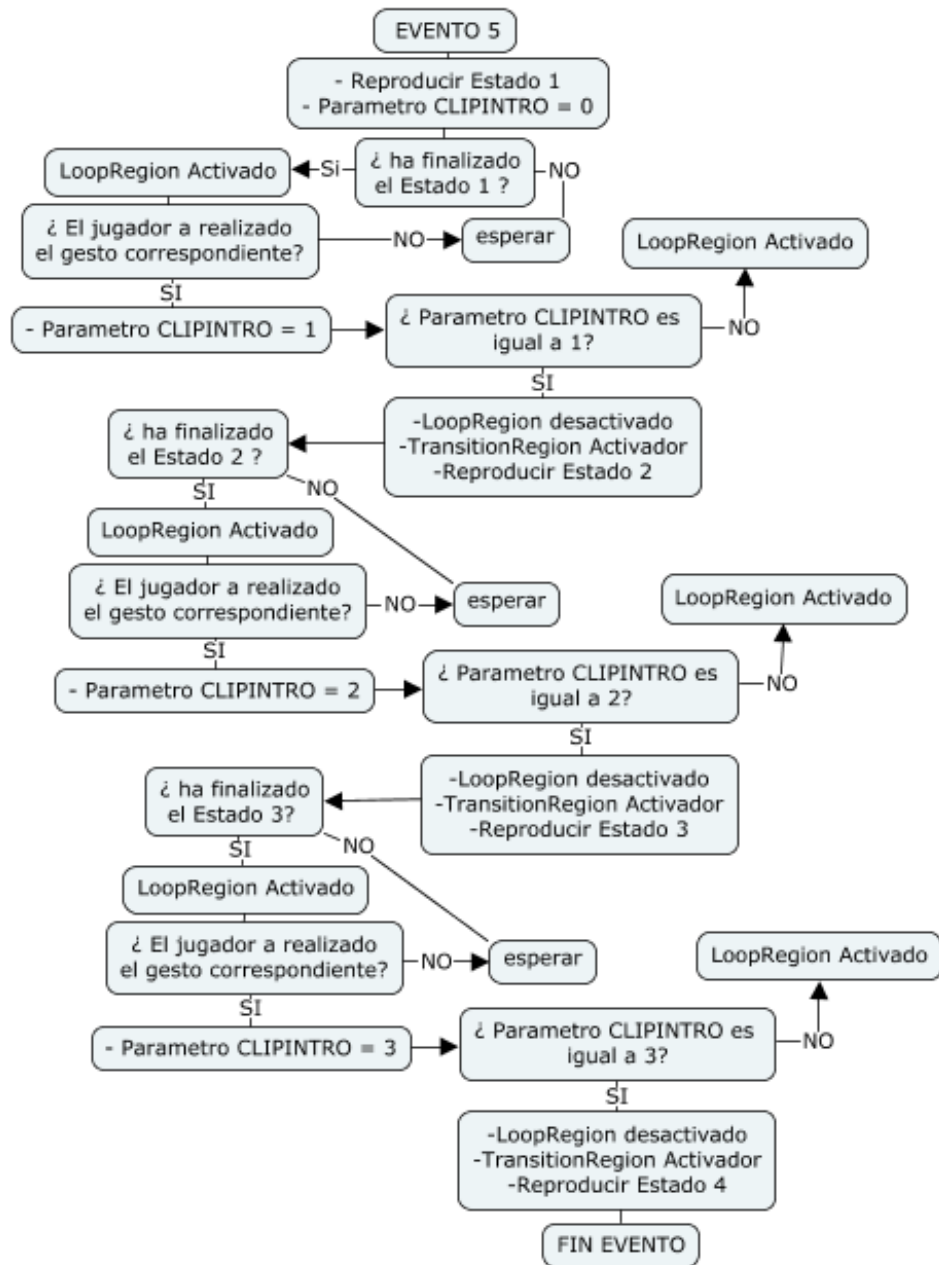


Figura 94. Diagrama de flujo Evento5, Imagen propia

El evento5 está constituido por cuatro tracks de audio, el primero correspondiente a los clips de narración del personaje principal *Ekko*, el segundo a la voz guía, el tercer corresponde al efecto *Fx\_050* y por último el *Fx\_000* correspondiente a la mosca en

el cuarto track. En este caso y como se puede observar en la figura 94, se pueden encontrar 4 marcadores que indican el inicio de los estados.

Al final de cada estado se encuentra ubicado un LoopRegion con el fin de evitar el progreso de la narración y un Transitionregion con el objetivo de ubicar el cursor en el siguiente marcador y así permitir la continuidad en la reproducción hasta su fin.

Una vez el audiojuego detecte el gesto correspondiente por parte del jugador, el parámetro *CLIPINTRO* activara los TransitionRegion dependiendo del valor numérico que el parámetro tenga, es decir, si parámetro *CLIPINTRO* es igual a 1, el cursor de reproducción se ubicara en el marcador *Estado 1* y así sucesivamente hasta pasar por todos los marcadores y finalizar el evento.

Este mismo procedimiento se aplica en los demás eventos mencionados anteriormente, la diferencia radica en la duración de los diálogos y en la cantidad de marcadores, LoopRegion, TransitionRegion y gestos de cada evento.

En el Anexo B se puede encontrar las imágenes correspondientes a los eventos *Evento17*, *Evento32* y *Evento44*

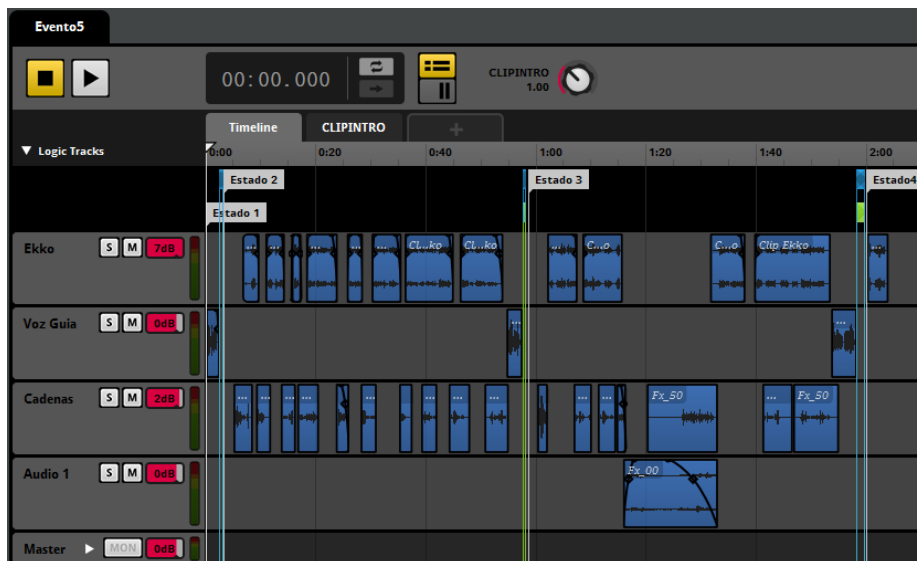


Figura 95. Interfaz Evento5 en FMOD Studio, Imagen propia

- **Eventos de Ambiente**

En estos eventos se pueden encontrar los audios destinados a los ambientes del audio juego. Para este caso dividimos en dos categorías los ambientes, los cuales se encuentran de la siguiente manera:

➤ **Evento Multitrack Ambiente 1**

EVENTO	CONTENIDO
1	Ambiente con animales
2	Ambiente con agua
3	Ambiente con rocas
4	Ambiente con gotas

Tabla 3. Eventos de ambiente 1

Estos eventos se encuentran en la categoría de ambiente 1 debido a que su operación lógica se basa en el uso de LoopRegion, lo cual permite escuchar los audios un número de veces necesario hasta que allá cambio de nivel.

Los archivos Fx\_51, Fx\_52, Fx\_53, Fx\_54 y Amb\_6 correspondientes a los audios complementarios de los ambientes en el nivel de introducción, se encuentra distribuidos en los eventos ya mencionados respectivamente.

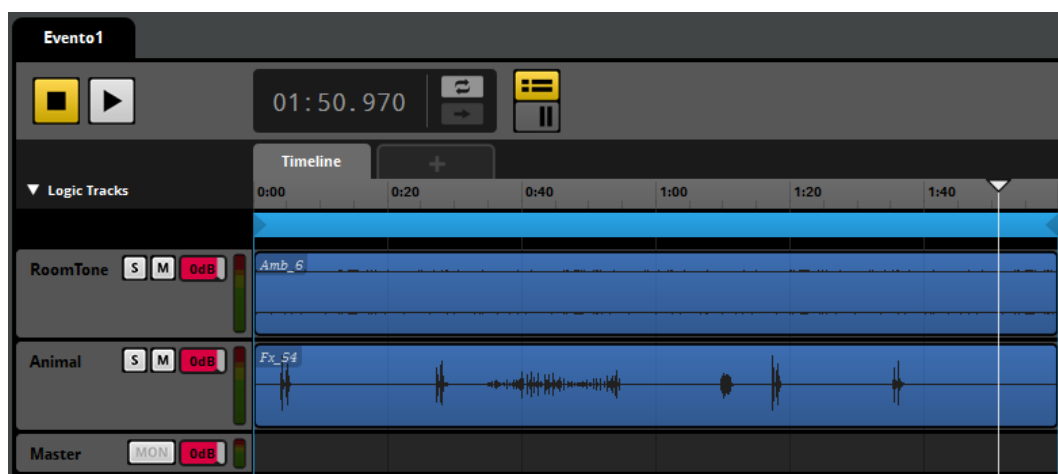


Figura 96. Interfaz Evento1 con audios Amb\_6 y Fx\_54

En el Anexo B se puede encontrar las imágenes correspondientes a los eventos 2, 3 y 4

➤ **Evento Multitrack Ambiente 2**

EVENTO	CONTENIDO	EVENTO	CONTENIDO
9	Ambiente Submarino 1	28	Ambiente Selvático 1
10	Ambiente Submarino 2	29	Ambiente Selvático 2
16	Ambiente red metálica	16	Ambiente Selvático 3
19	Ambiente Submarino 3	9	Ambiente Selvático 4
20	Ambiente Submarino 4		

Tabla 4. Eventos de ambiente 2

El funcionamiento de estos eventos es diferente al encontrado en los Ambiente 1. Para entender mejor su funcionamiento, se tomará como ejemplo el *Evento19*

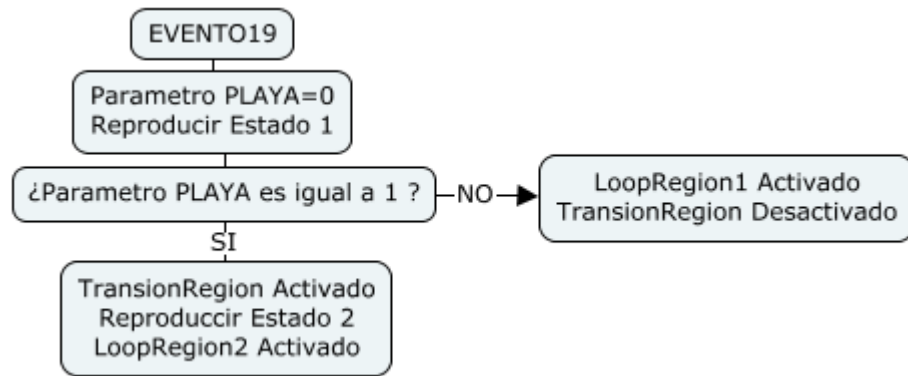


Figura 97. Diagrama de flujo Evento19, Imagen propia

La operación lógica de este tipo de eventos consiste en mantener uno o varios audios en reproducción dentro de un LoopRegion mientras transcurre el escenario. Cuando se lleva a cabo el cambio de ambiente dentro del mismo nivel, el valor numérico del parámetro *PLAYA* será igual a 1; Cuando esto ocurra, el TransitionRegion será activado y ubicara en cursor en el marcador *Estado 2*. Esto permite reproducir los audios correspondientes al siguiente ambiente.

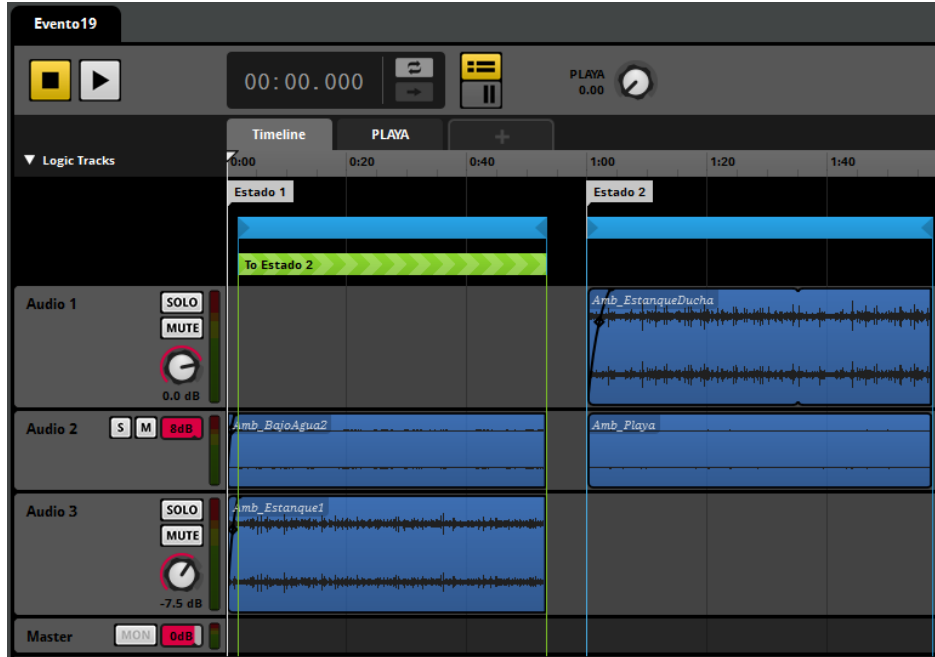


Figura 98. Interfaz Evento19 en FMOD Studio, Imagen propia

Los Imágenes de los *Eventos 20, 10, 9, 16, 28, 29, 30 y 31* se pueden observar en el anexo B.

- **Eventos Naturales**

EVENTO	CONTENIDO	EVENTO	CONTENIDO
6	Pasos jugador en ladrillo	23	bichos
11	Braceo jugador	24	Rana
12	Pasos jugador en agua	26	Onda expansiva
18	Sirvientes Yacu	27	Serpiente
21	Bichos Aplastados	42	Activacion reto
22	Pasos jugador en cesp�ed	43	Golpe jugador acertado
		45	Wosh push jugador

Tabla 5. Eventos naturales

En esta categor a podemos encontrar eventos simples. En el caso de los eventos 23, 24 y 27, se les adiciono un par metro llamado *PANEO* (*m nimo=0, m ximo=3*), con el fin de variar el paneo de dichos audios durante la reproducci n del nivel de Dante.

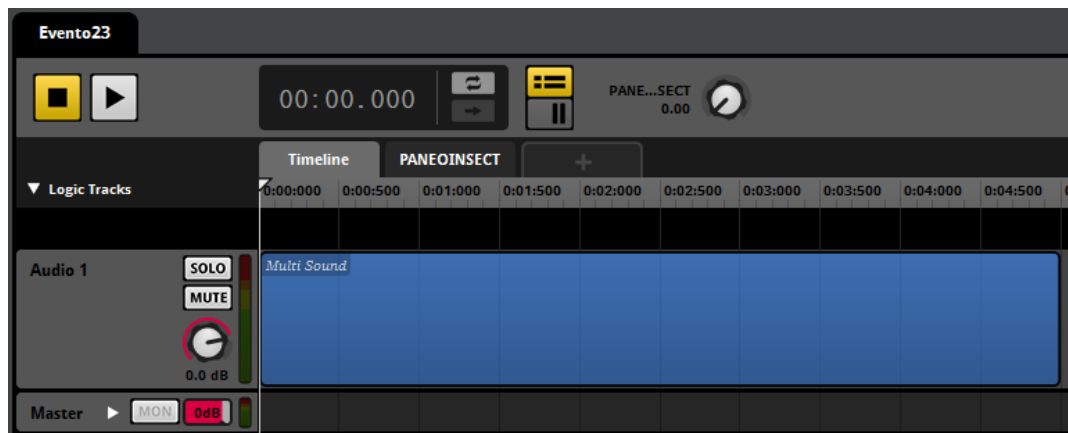


Figura 99. Interfaz Evento23 en FMOD Studio, Imagen propia

En el Anexo B se puede encontrar las im genes correspondientes a los eventos 6, 11, 12, 18, 21, 22, 24, 26, 27, 42, 43, 45.

- **Eventos Compuestos**

EVENTO	CONTENIDO	EVENTO	CONTENIDO
13	Poder ola Yacu	16	Red metalica
14	Propulsor jugador	25	Lanzamiento de arboles
15	Orca asesina	41	Portal de teletransportacion

Tabla 6. Eventos compuestos en FMOD Studio

En esta categor a se pueden encontrar los eventos de tipo Multitrack que tienen una  nica operaci n l gica para su modo de trabajo.



➤ *Evento41*

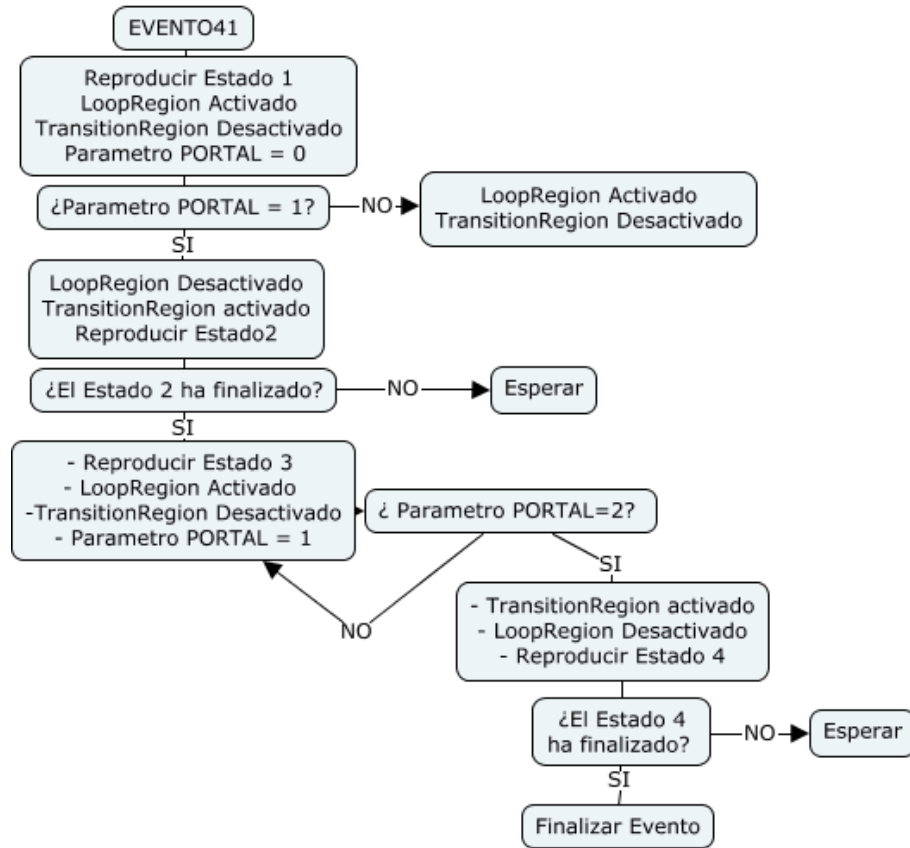


Figura 100. Diagrama de flujo Evento41, Imagen propia

En el *evento41* tiene como objetivo reproducir el archivo de audio Fx\_011 (portal de tele transportación) al inicio y al fin de cada nivel en el audiojuego. El evento se encuentra dividido en 4 estados. El estado 1 es un loop vacío que permite tener en stop el evento durante la reproducción de un escenario, el estado 2 corresponde al inicio del audio Fx\_011, el estado 3 se encarga de mantener el audio en un loop hasta que el jugador realice el gesto correspondiente y el estado 4 corresponde al Fx\_001 (Portal de tele transportación 2) y la parte final del Fx\_011.

El cursor de reproducción va pasando al siguiente dependiendo del valor numérico asignado por el parámetro *Portal* (mínimo=0 y máximo=2)

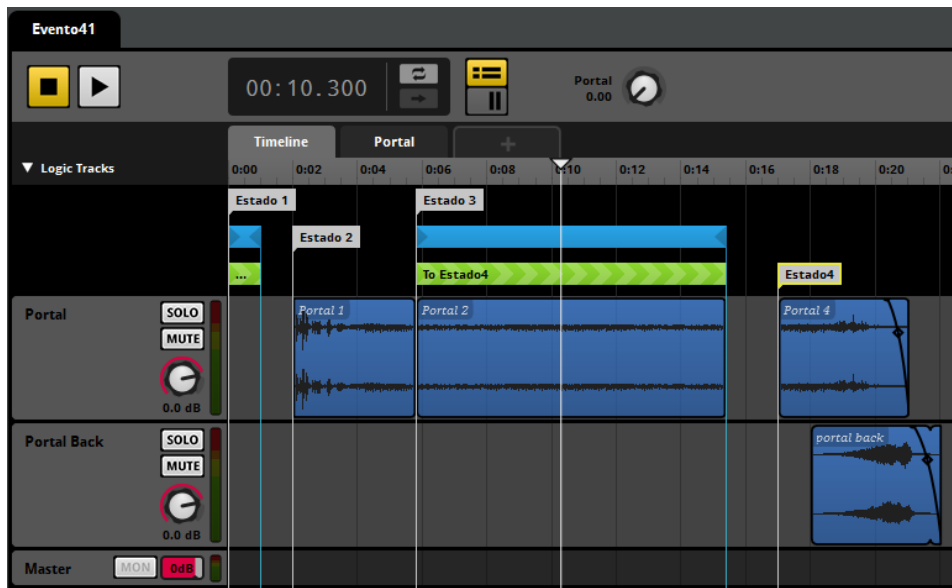


Figura 101. Interfaz Evento41 en FMOD Studio, Imagen propia

➤ *Evento13*

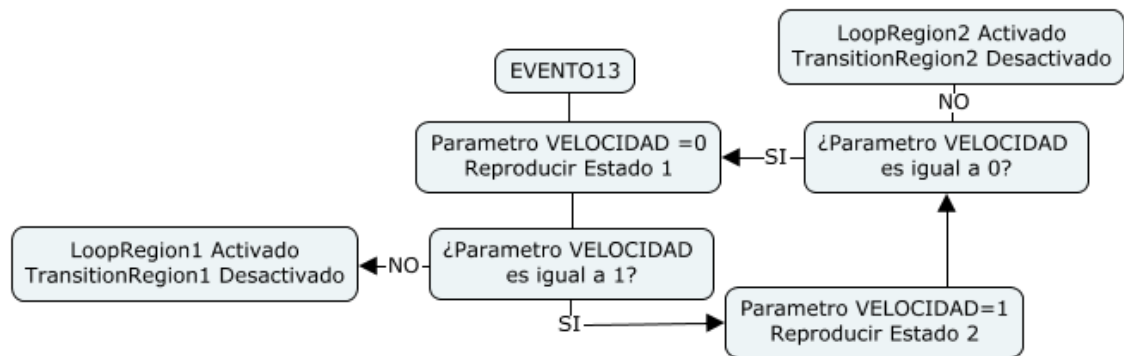


Figura 102. Diagrama de flujo Evento13, Imagen propia

El *Evento13* corresponde al poder de olas de Yacu, el cual será reproducido en el transcurso de un reto del nivel 3. En este evento encontramos 2 estados. El primero corresponde a un loop vacío que permite tener en stop el evento durante la reproducción de un escenario y el segundo corresponde a un objeto multisound que contiene 3 audios correspondientes al poder de olas de yacu (Fx\_023).

El cursor de reproducción va cambiando al siguiente estado dependiendo del valor numérico asignado por el parámetro *Velocidad* (mínimo=0 y máximo=1)

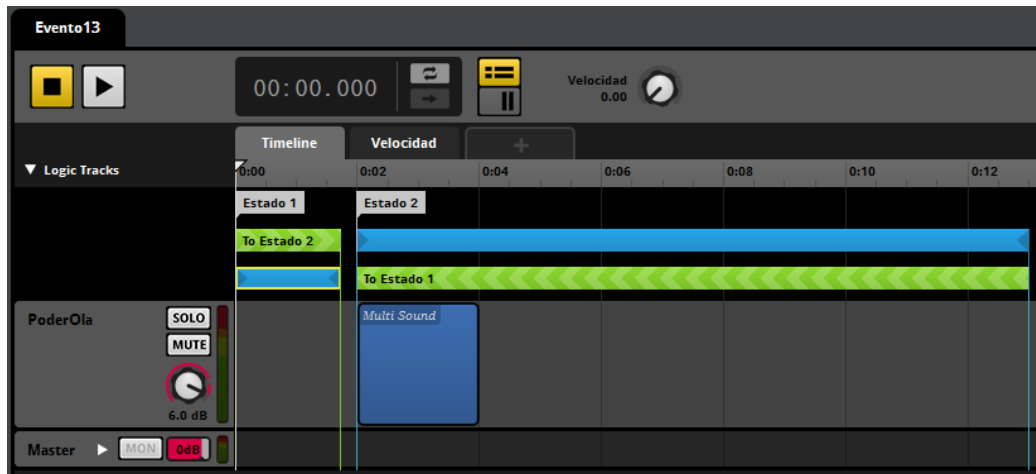


Figura 103. Interfaz Evento13 en FMOD Studio, Imagen propia

➤ *Evento14 y Evento 15*

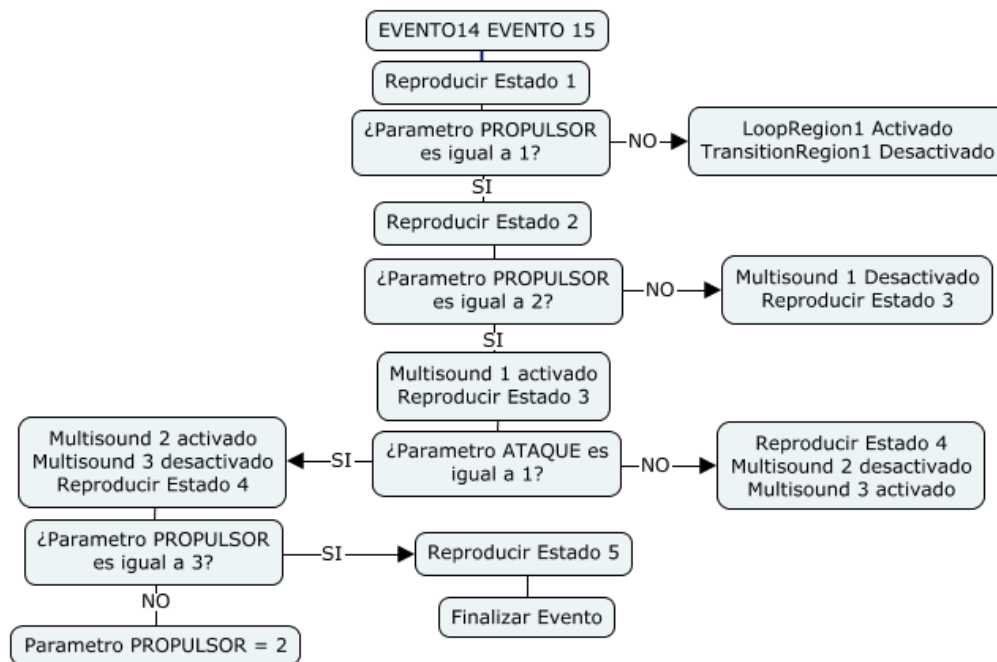


Figura 104. Diagrama de flujo Evento14 y Evento 15, Imagen propia

Se optó por explicar el funcionamiento de estos dos eventos como uno solo debido a que, ambos eventos contienen parámetros lógicos iguales. El *Evento14* correspondiente al propulsor del jugador y el *Evento15* correspondiente a los audios de la Orca. Ambos eventos incluidos en el nivel 3.

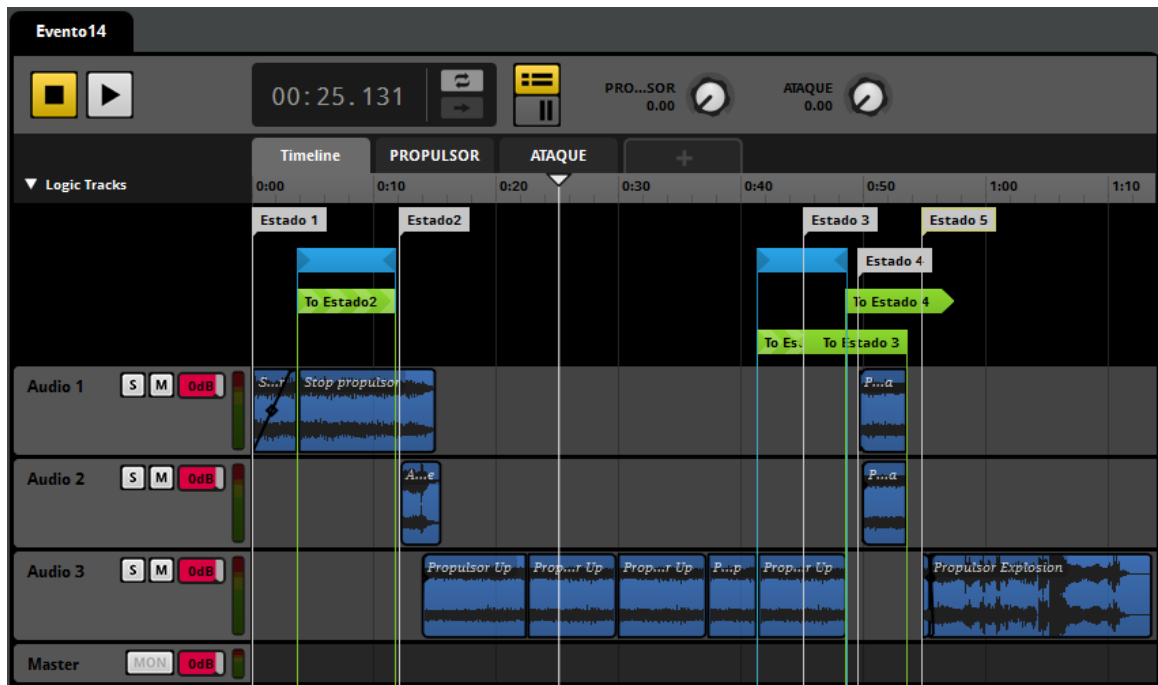


Figura 105. Interfaz Evento14 Propulsor en FMod Studio, Imagen propia

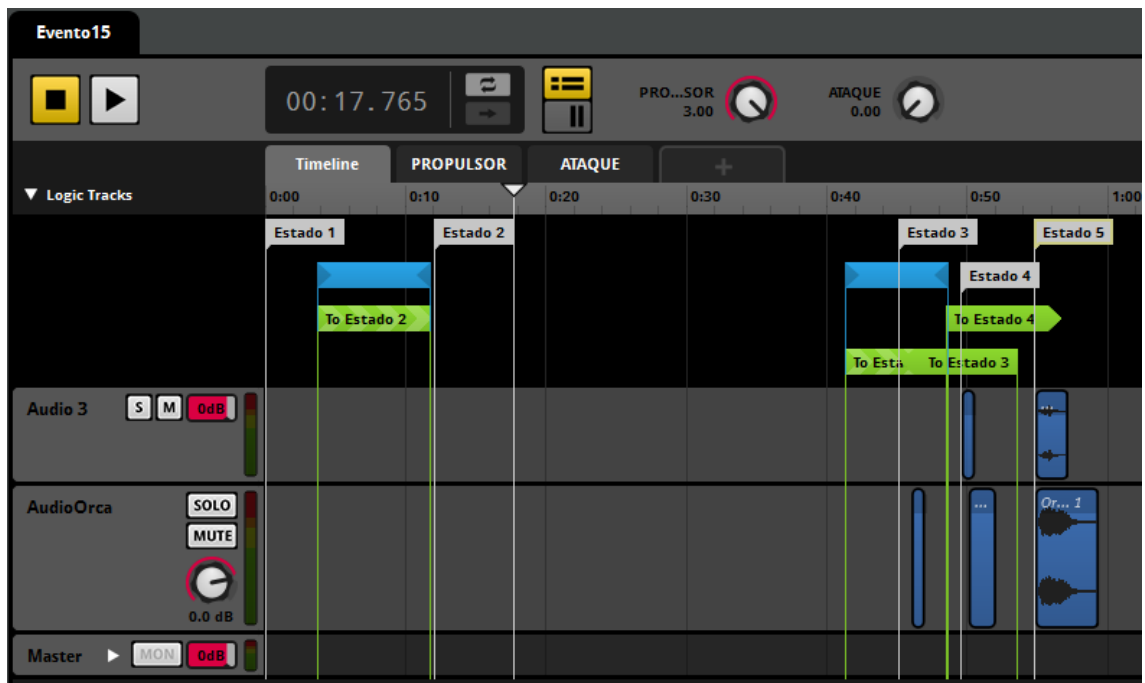


Figura 106. Interfaz Evento15 Orca en FMod Studio, Imagen propia

En este caso, los eventos están divididos en 5 estados. Estado 1 es el efecto del propulsor apagado correspondiente al Fx\_004, estado 2 al efecto de encendido del propulsor correspondiente Fx\_006, estado 3 al propulsor encendido y estado 4

correspondientes a los efectos de la orca (Fx\_012 a Fx\_021) y movimientos del propulsor (Fx\_007 y Fx\_008).

En el estado 4 la reproducción de los audios depende del gesto realizado por el jugador; si el jugador realiza el gesto correcto, se reproducirán audios correspondientes a reto superado (Fx\_007, Fx\_020, Fx\_021), de lo contrario reproducirá audios correspondientes a reto fallado (Fx\_008, Fx\_017, Fx\_018, Fx\_019). Los audios de la orca están incluidos en objetos multisound en el evento 15 y del propulsor en el evento 14. El comportamiento del estado 4 dependen de los valores asignados por el parámetro *Ataque* (*mínimo=0, máximo=1*)

Por último, el estado 5 corresponde a la avería y explosión del propulsor (Fx\_022) y llanto de la orca (Fx\_012 y F\_013). El comportamiento de estos eventos depende de los valores asignados por el parámetro *Propulsor* (*mínimo=0, máximo=3*)

➤ *Evento25*

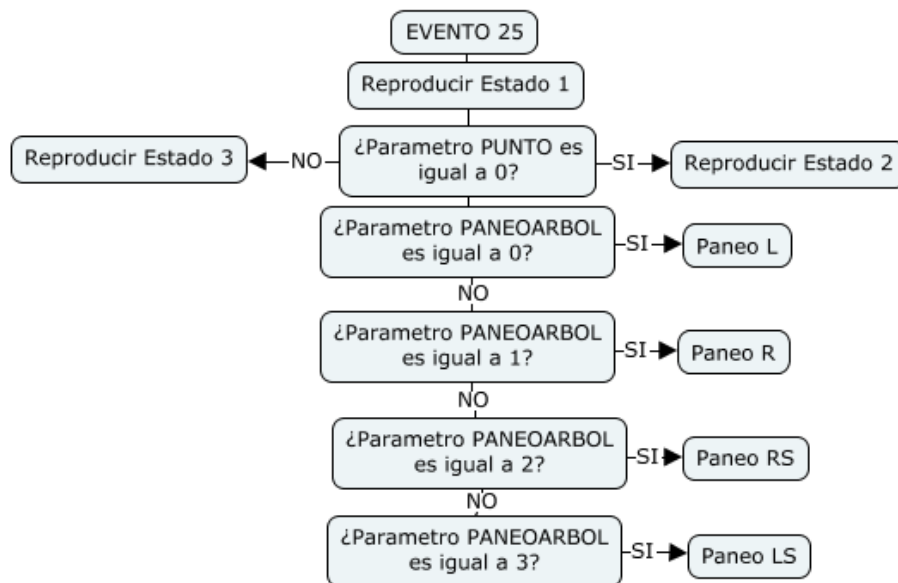


Figura 107. Diagrama de flujo Evento25, Imagen propia

El evento 25 está compuesto por 3 estados, el estado 1 corresponde a los archivos de audio Fx\_086 y Fx\_087 (lanzamiento de árboles), el estado 2 a los golpes acertados a los árboles incluidos también en los audios Fx\_086, Fx\_087 y el estado 3 al efecto Fx\_100. Este evento será reproducido en uno de los retos establecidos en el nivel de Dante.

En este caso, el estado 2 o el estado 3 será ejecutado dependiendo del gesto realizado por el jugador; Si el jugador realiza el gesto correcto, se reproducirá el

estado 2, sino, se reproducirá el estado 3. El comportamiento de esta característica depende de los valores asignados por el parámetro *Punto* (*mínimo=0, máximo=1*).

Adicional al parámetro *punto*, también se encuentra el parámetro *Paneo* (*mínimo=0, máximo=3*). Este parámetro permite ubicar la fuente de sonido en cualquiera de los parlantes establecidos del arreglo.

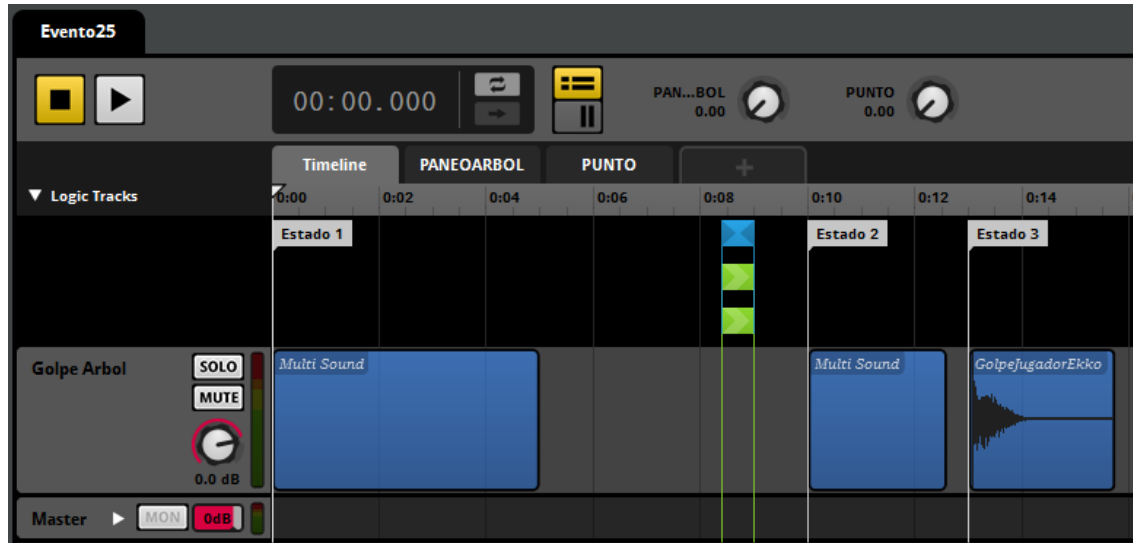


Figura 108. Interfaz Evento25 FMOD Studio, Imagen propia

## 7.2. Análisis ingenieril

Como se había mencionado anteriormente y tomando como base la norma ESBR para audiovisuales, se establece una población conformado por jóvenes entre 14 y 35 años de edad. La muestra está conformada por jóvenes estudiantes de la universidad San Buenaventura Bogotá. Se escogió esta muestra debido a que el audiojuego se encuentra ubicado dentro de las instalaciones de la universidad, permitiendo realizar con facilidad la ejecución de las pruebas por cuestión de tiempo y disponibilidad de los estudiantes, los cuales serán denominados de ahora en adelante como *los jugadores*.



Figura 109. Jugador interactuando con el audiojuego, Imagen propia

El audio juego fue probado en el “Live Room” del estudio híbrido de las instalaciones de la Universidad de San Buenaventura Bogotá, este espacio se escogió debido a que cuenta con las dimensiones óptimas para ubicar el arreglo de parlantes de primer orden de Ambisonic, sin que este interfiriera, ni fuese incómodo para la interacción del jugador por medio de movimientos corporales con el audio juego.

Equipos y elementos utilizados:

Equipos	Cantidad	Especificaciones	
Computador	1	Procesador	AMD FX-8120 Eight-Core Processor 3.10 GHz
		RAM	8 GB
		Sistema operativo	Windows 10 Pro, 64 Bits
		Placa Base	Asus M5A78L – Tarjeta de audio 5.1 Integrada
Kinect	1	Microsoft Kinect Version 1	
Parlantes Electro Voice	4	ELX 112 p Respuesta en frecuencia -3dB 60 – 18000 Hz -10dB 50 – 20000 Hz	
Trípodes	4		
Cables	2	Plug 1/8 Estereo a 2 XLR hembra	
Cables	4	XLR	
Cajas directas	4	Whirlwind IMP2	

Tabla 7. Equipos y elementos usados en la instalación interactiva

Para poder evaluar el rendimiento del audiojuego, se optó por diseñar una encuesta en la cual se busca que los jugadores den una calificación de su experiencia dentro de la instalación. La encuesta contiene preguntas dirigidas a los dos temas fundamentales del audiojuego: respuesta efectiva en interacción y percepción de la espacialidad sonora.

La encuesta está conformada por un total de 12 preguntas incluyendo preguntas sobre el sexo y edad de los jugadores. En el anexo G se puede observar el formato de la encuesta.



*Figura 110. Jugador interactuando con el audiojuego, Imagen propia*

El audio juego fue probado por un total de 10 jugadores, de diferentes edades y estaturas, con el fin de obtener una retroalimentación variada dada por cada una de estas, además también se adquirió información relevante con el funcionamiento del audio juego. Todos los jugadores tuvieron que pasar por cada uno de los niveles desarrollados para entender el funcionamiento del audio juego, y así poder interactuar con la interfaz sonora por medio de movimientos corporales para finalizar el audio juego.

- **TIEMPO DE EJECUCIÓN DEL AUDIOJUEGO**

Tiempo estimado para completar el audiojuego: 10minutos

- Jugador 1

Tiempo promedio: 12m 53s - Estatura: 1,72cm



El jugador tiene dificultad para entender el movimiento Swipeleft/SwipeRight en el escenario de entrenamiento. Una vez entendido el movimiento, supera los tres siguientes niveles sin dificultad.

- Jugador 2

Tiempo promedio: 22m 17s - Estatura: 1,83cm

El jugador tiene dificultad para entender el movimiento Swipeleft/SwipeRight en el escenario de entrenamiento. El nivel de introducción es superado sin dificultad. En el nivel de Yacu el jugador tiende a “bracear” en los límites de la instalación debido a su altura, esto hace que el Kinect no reconozca el movimiento de sus brazos y lo asimila como “push”. En el nivel de Dante el jugador vuelve a desplazarse por toda la instalación

- Jugador 3

Tiempo promedio: 16m 44s - Estatura: 1,74cm

El jugador tiene dificultad para entender el movimiento Swipeleft/SwipeRight en el escenario de entrenamiento. El nivel de introducción es superado sin dificultad. En el nivel de Yacu el jugador tiende a tener dificultad con el gesto “squat” ya que no se levanta inmediatamente, sino que se queda agachado, esto hace que pierda tiempo en la superación del reto. En el nivel de dante el audiojuego tiene dificultades en reconocer el gesto “push”, una vez superado el reto, el jugador supera el nivel sin dificultad.

- Jugador 4

Tiempo promedio: 15m 26s - Estatura: 1,57cm

El jugador tiene dificultad para entender el movimiento Swipeleft/SwipeRight en el escenario de entrenamiento. Una vez entendido el movimiento, supera los dos siguientes niveles sin dificultad. En el nivel de Dante el jugador se desplaza al lado de una base de cabina, en este caso, el Kinect reconoce el trípode como el nuevo cuerpo a reconocer, teniendo como resultado la no detección del cuerpo del jugador. Para solucionar el problema, el jugador tuvo que salir de la instalación y volver a ingresar para que le audiojuego reconociera al jugador como el nuevo cuerpo.

- Jugador 5

Tiempo promedio: 18m 20s - Estatura: 1,67cm

El jugador supera los niveles de entrenamiento e introducción sin dificultad. En el nivel de Yacu el jugador tiene problemas a la hora de ejecutar el movimiento

SwipeLeft/SwipeRight teniendo como resultado pérdida de tiempo en la primera parte del nivel. El nivel de Dante es cumplido satisfactoriamente.

- Jugador 6 - Estatura: 1,75cm

Tiempo promedio: 11m 37s

El jugador cumple satisfactoriamente los 4 niveles sin dificultad.

- Jugador 7 - Estatura: 1,58cm

Tiempo promedio: 15m 03s

En los niveles de entrenamiento, Yacu y Dante el reconocimiento del gesto "Jump" es interpretado como el gesto "Squat", lo cual tiene como resultado pérdida de tiempo en la ejecución del audiojuego. A diferencia de este gesto, los demás son ejecutados satisfactoriamente.

- Jugador 8 - Estatura: 1,79cm

Tiempo promedio: 26m 12s

El jugador tiene dificultades en la gran mayoría de los gestos sobre todo en los niveles de Yacu y de Dante, además el jugador tiende a caminar por los límites de la instalación debido a su altura teniendo como consecuencia la pérdida de reconocimiento corporal por parte del Kinect.

- Jugador 9 - Estatura: 1,72cm

Tiempo promedio: 13m 47s

El jugador tiene dificultad para entender el movimiento "Jump" en el escenario de entrenamiento. Una vez entendido el movimiento, supera los tres siguientes niveles sin dificultad.

- Jugador 10 - Estatura: 1,64cm

Tiempo promedio: 15m 33s

El jugador tiene dificultad para entender el movimiento Swipeleft/SwipeRight en el escenario de entrenamiento. En el nivel de introducción y Dante son superados sin dificultad. En el nivel de Yacu toma su tiempo para superar la parte final del nivel debido a que el audiojuego reconoce el gesto "jump" como "Squat" en 4 ocasiones.

Se pudo observar que cada una de las personas que participo en el audio juego, se tomó una cantidad de tiempo diferente, esto se debe a que las instrucciones dadas

por la voz guía no eran acatadas en el momento por el usuario, en ocasiones también se presentaba confusiones con los movimientos que debía ejecutar y por último el reconocimiento hecho por el Kinect no era el correcto, es por eso que el tiempo de ejecución varía en cada una de las personas.

De igual manera se pudo observar que al tener una voz guía, el usuario necesita instrucciones muy precisas de la manera en que debe interactuar con sus movimientos, ya que en diferentes momentos el usuario no reaccionaba de la manera correcta y era necesario brindarle un poco más de ayuda para que pudiera continuar, pues la voz guía que contiene el audiojuego para muchas de las personas no era lo suficientemente precisa.

El reconocimiento de gestos presenta algunas fallas en la captura del movimiento JUMP debido a que su funcionamiento se basa en el seguimiento y comparación de posición de diferentes articulaciones de manera simultánea, muchas veces los jugadores al realizar dicho movimiento, el software lo puede interpretar como otro gesto en este caso SQUAT, lo cual, de manera interna, el audiojuego lo descifra a como gesto incorrecto impidiendo la continuidad de este.

- **RESULTADOS DE LA ENCUESTA**

1. ¿Qué edad tiene?

22
22
22
27
27
24
24
25
20
23

Figura 111. Resultados pregunta 1, Imagen propia

Se puede observar que el rango de edades de los encuestados esta entre los 20 a los 27 años. Con este rango se cumple los límites establecidos dentro de la población

2. ¿Cuál es su tipo de sexo?

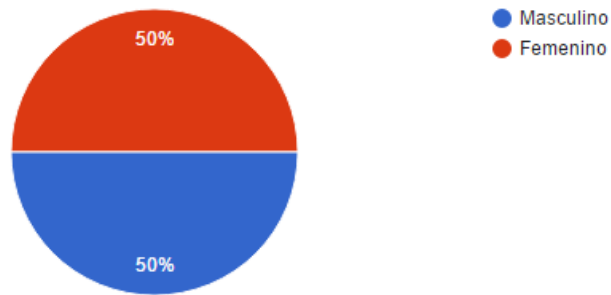


Figura 112. Resultados pregunta 2, Imagen propia

La muestra está dividida en partes iguales en cuanto a tipo de sexo, 5 mujeres – 5 hombres

3. ¿Acostumbra a jugar videojuegos?

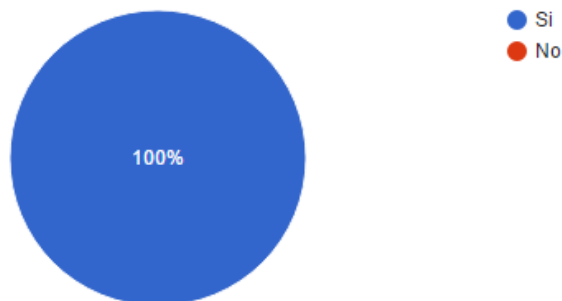


Figura 113. Resultados pregunta 3, Imagen propia

El total de los jugadores acostumbra a jugar videojuegos

4. ¿Ha jugado alguna vez sin interfaz gráfica?

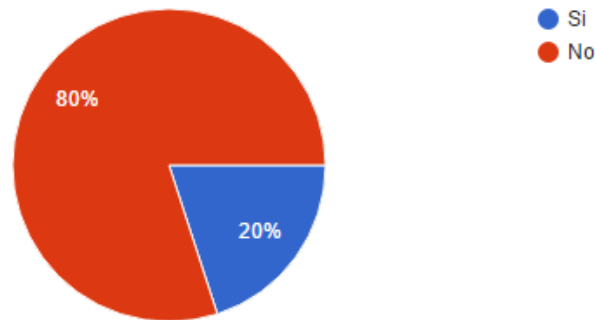


Figura 114. Resultados pregunta 4, Imagen propia

Dentro de la muestra escogida, se evidencia una superioridad en el uso de la interfaz gráfica. Aun así, por el número de personas encuestadas no es preciso decir que hay un déficit en el uso de audiojuegos

5. Elija de 1 a 5, tomando como 5 el mayor grado de dificultad, ¿qué tan difícil le pareció jugar con una interfaz sonora?

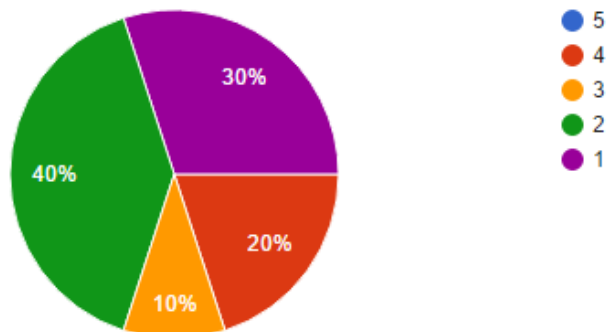


Figura 115. Resultados pregunta 5, Imagen propia

El 80% de los encuestados piensa que jugar únicamente con una interfaz sonora no es complicado.

6. Elija de 1 a 5, tomando como 5 el mayor grado de dificultad, ¿qué tan difícil le pareció jugar con movimientos corporales?

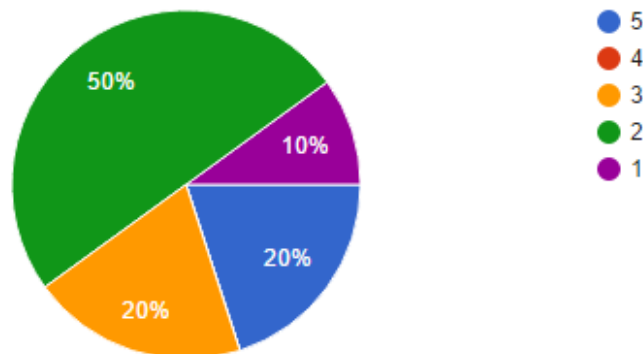


Figura 116. Resultados pregunta 6, Imagen propia

De los encuestados, el 20% piensa que jugar mediante gestos corporales es muy difícil, el 20% normal, el 50% fácil y 10% muy fácil. Con un total de 80% se demuestra que utilizar el cuerpo como medio de control del audiojuego es fácil para superar los niveles del audiojuego

7. Elija de 1 a 5, tomando 5 como el mejor valor, si ¿las instrucciones brindadas por el audiojuego fueron claras y concisas?

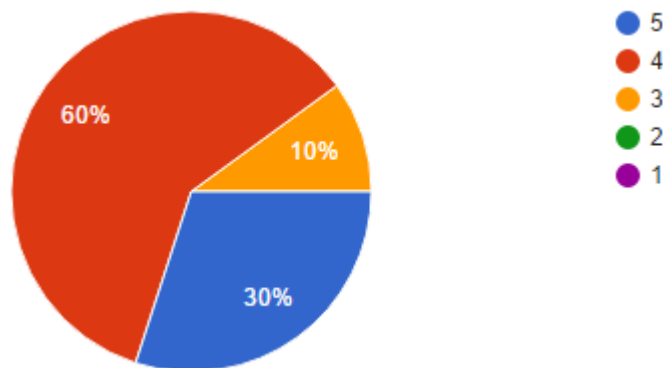


Figura 117. Resultados pregunta 7, Imagen propia

De los encuestados, el 30% piensa que las instrucciones fueron excelentes, el 60% sobresaliente y el 10%. Con un total de 100% tomando 60% como máximo valor, se demuestra que las instrucciones son lo suficientemente claras para superar niveles en el audiojuego.

8. Elija de 1 a 5, tomando 5 como el mejor valor, si ¿el audiojuego reconoció, en el momento que era, los movimientos efectuados por usted?

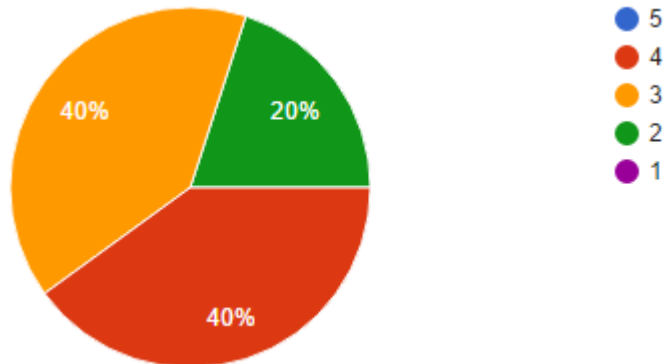


Figura 118. Resultados pregunta 8, Imagen propia

De los encuestados, el 40% piensa que el audiojuego reconoce sobresalientemente los gestos efectuados, el 40% aceptable, 20% insuficiente. Con un total de 80% se demuestra que el audiojuego sobresale a la hora de reconocer los gestos, pero de igual manera se presentan fallos por inconvenientes mencionados anteriormente.

9. ¿Fue fácil recordar los movimientos explicados en el nivel de introducción?

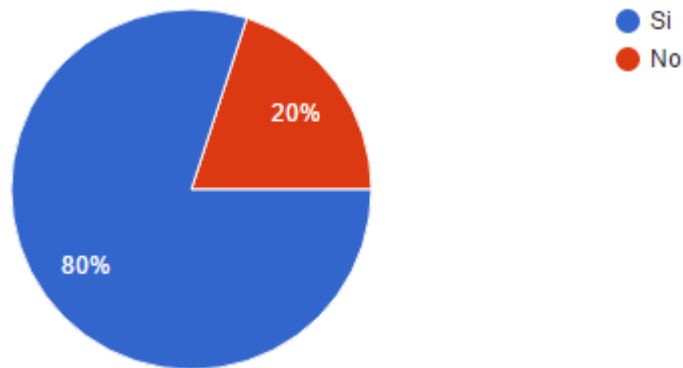


Figura 119. Resultados pregunta 9, Imagen propia

Con un total del 80%, se demuestra que los movimientos seleccionados para el control del audiojuego son fáciles de recordar, lo cual permite una mayor facilidad a la hora de superar los niveles.

10. ¿Fue fácil identificar la ubicación de los sonidos dentro de la instalación?

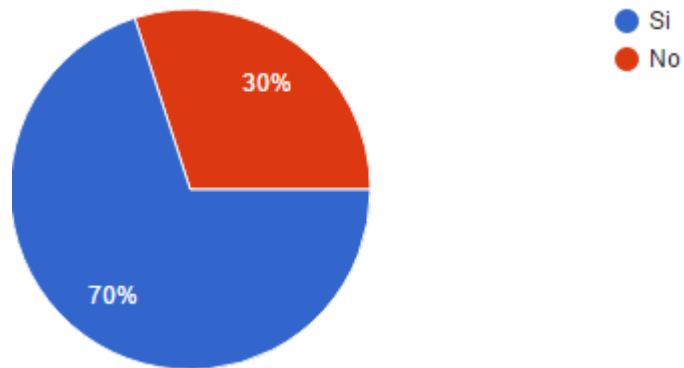


Figura 120. Resultados pregunta 10, Imagen propia

Con un 70%, los jugadores piensan que la distribución y la especialización sonora dentro del audiojuego es sobresaliente

11. ¿Siente usted que el audiojuego brinda una buena experiencia de audio 3D?

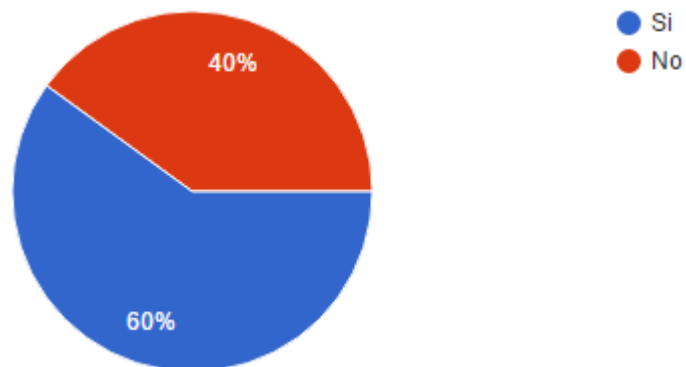


Figura 121. Resultados pregunta 11, Imagen propia

Con un 60%, los jugadores piensan que la inmersión y el sonido envolvente que brinda el audiojuego es aceptable, demostrando que en este campo el audiojuego presenta un déficit a la hora de brindar realismo en su jugabilidad.



12. ¿Qué mejoraría o recomendaciones daría sobre el audiojuego?

en los ambientes algunos audios enmascaran otros

alguos audios no se diferencian de otros

se debe mejorar el reconocimiento de movimientos

podría cambiarse la forma en que el video juego reconoce los gestos

mejorar como se reconocen los movimientos

TOCA MEJORAR EL RECONOCIMIENTO DE GESTOS

MEJORAR EL RECONOCIMIENTO

*Figura 122. Resultados pregunta 12, Imagen propia*

Tal y como se venía apreciando anteriormente, se presentan algunas fallas a la hora de reconocer los gestos ya sea por factores humanos o del audiojuego

## 8. CONCLUSIONES

- Con base en los jugadores encuestados, el reconocimiento de gestos por parte del audiojuego funciona con una eficacia del 90% en personas con una altura promedio de 1,50m a 1,75m.
- Se presentan inconvenientes en el reconocimiento de gestos para aquellas personas con altura superior a las mencionadas anteriormente, como lo es en el caso de los jugadores 2 y 8, debido a que el tamaño del sweet spot establecido para la interacción con el audiojuego es pequeño.
- Basados en las preguntas 5, 7 y 9 de la encuesta, cuyo énfasis está orientado en conocer la aceptación por parte del usuario al medir características como: la dificultad de interacción con un sistema sonoro por medio de una interfaz natural, el entendimiento de las instrucciones brindadas por el audiojuego y la facilidad de recordar gestos que actúan como canal de comunicación, arrojaron un valor promedio del 77.5%, lo que demuestra que más de  $\frac{3}{4}$  partes de la población encuestada le dieron el visto bueno a esta forma de juego e interacción.
- Aunque en la configuración general dentro de Unity 3D se puede omitir movimientos de otro usuario, se presentaron algunos inconvenientes con superficies externas, como en el caso del jugador 4, los trípodes usados en la instalación influyeron en el reconocimiento de gestos asimilando estas estructuras como parte corporal del jugador.
- Tomando como referencia los resultados obtenidos de las preguntas 10 y 11, se evidencia que el 65% de los encuestados le dieron una calificación aceptable al diseño y especialización sonora, sin embargo, este resultado no es contundente ya que demuestra que el audiojuego queda corto al brindar una buena inmersión sonora.
- Se puede observar que los parámetros usados y la versión del plugin *oculus*, implementado para la reproducción de audio en sistema ambisonic en la integración FMOD Studio – Unity 3D, afecta con pequeñas variaciones de amplitud en la mezcla general del audiojuego.

- Se puede observar que el 80% de los jugadores puede recordar con facilidad la totalidad de los gestos implementados. El gesto con mayor grado de dificultad de entendimiento fue el *SwipeRigth* y *Swipeleft*. En el audiojuego este movimiento fue denominado *Braceo* lo cual provoco que cada jugador ejecutara este movimiento de acuerdo a la interpretación y al conocimiento previo que cada uno tiene sobre esta palabra. Solo el 30% de los jugadores superó satisfactoriamente este movimiento en el primer intento.
- La altura de cada jugador influye en la distancia que hay entre los objetos de reproducción de audio y el audiolistener establecido en la cabeza del avatar dentro de Unity. Debido a esto, el audiojuego tiene como resultado variaciones en el volumen general.
- Se logró realizar un audiojuego manejado por medio de movimientos corporales que cumpliera con los objetivos planteados.
- Para el desarrollo del audiojuego, FMOD Studio es un motor de programación de audio procedimental que facilita la creación de un entorno sonoro para cualquier tipo videojuego, haciendo posible instaurar atmosferas, ambientes, efectos, etc. En este caso se logró crear diferentes escenarios sonoros para el propósito del audiojuego, permitiendo al jugador controlar los parámetros que este brinda y así cambiar, en tiempo real, el entorno sonoro que lo rodea
- La implementación de Unity 3D como motor de programación para el desarrollo de este audiojuego, facilito el mecanismo de comunicación entre el audiojuego y el usuario debido a las múltiples herramientas que brinda este entorno, en este caso, el uso de SDK Kinect.
- Debido al uso de plugin dedicados al uso de la interfaz de comunicación Kinect, el audiojuego puede ser jugador únicamente en sistemas operativos Windows.

## 9. RECOMENDACIONES

- Se recomienda realizar una configuración minuciosa en el software para la reproducción de audio en un sistema de primer orden Ambisonics.
- El arreglo de parlantes debe tener una altura adecuada según la estatura de cada jugador.
- El Kinect debe ubicarse de manera correcta para que los movimientos de las articulaciones del usuario sean registrados correctamente.
- Se debe tener en cuenta el entorno en el que se instala el audiojuego, se busca que el entorno tenga superficies planas con una mínima cantidad de objetos para que disminuya el riesgo de reconocimiento.
- El usuario debe mantener su posición central, para que el reconocimiento hecho por el Kinect sea lo mejor posible y además siempre este adquiriendo información sonora por el arreglo de los 4 parlantes.
- Usar en preferencia un sistema que cuente con un sistema de reproducción 5.1 integrado si se va a usar en Windows, debido a que, al usar una interfaz de audio externa, se presenta problemas de latencia y Drops out y en muchas de las ocasiones no da más que una reproducción en estéreo.

## 10. REFERENCIAS

- Aguacia Andres, P. D. (2014). *Diseño de atmosferas sonoras tridimensionales utilizando sistemas Ambisonic*. Bogota Dc, Colombia.
- AudioGame. (s.f.). *What is Audiogame*. Obtenido de [www.audiogames.net](http://www.audiogames.net)
- Courville, D. (2007 - 2010). *Ambisonic Studio: B-Format recording*. Obtenido de [www.radio.uqam.ca](http://www.radio.uqam.ca)
- Cueva, J. M. (s.f.). *Interacción hombre-Maquina*. Obtenido de [di002.edv.uniovi.es](http://di002.edv.uniovi.es)
- Elen, R. (2001). *Ambisonics: The surround alternative*. Obtenido de [www.ambisonic.net](http://www.ambisonic.net)
- ESRB. (s.f.). *Guia de clasificaciones de la ESRB*. Obtenido de [www.esrb.org](http://www.esrb.org)
- G. Roy, J. J. (s.f.). *Técnicas de microfónica estéreo*. Obtenido de [www.analfatecnicos.net](http://www.analfatecnicos.net)
- JAM. (s.f.). *AudioGame Jam*. Obtenido de [jams.gamejolt.io/AGJam](http://jams.gamejolt.io/AGJam)
- Shure. (s.f.). *Técnicas de microfónica estéreo*. Obtenido de [www.shure.es](http://www.shure.es)
- Simon, P. I. (2005 ). *El diseñador de sonido: Funcion y esquema de trabajo*. Obtenido de <http://www.pabloiglesiassimon.com/>

## 11. BIBLIOGRAFIA

- Aguacia Andres, P. D. (2014). *Diseño de atmosferas sonoras tridimensionales utilizando sistemas Ambisonic*. Bogota Dc, Colombia.
- Alonzo, P. C. (2009). *Tecnologías de reconocimiento de voz y su aplicabilidad en video juegos* . Madrid.
- AudioGame. (s.f.). *What is Audiogame*. Obtenido de [www.audiogames.net](http://www.audiogames.net)
- C., U. (s.f.). *Unity 3D*. Obtenido de [www.unity3d.com.es](http://www.unity3d.com.es)
- Courville, D. (2007 - 2010). *Ambisonic Studio: B-Format recording*. Obtenido de [www.radio.uqam.ca](http://www.radio.uqam.ca)
- Cueva, J. M. (s.f.). *Interacción hombre-Maquina*. Obtenido de [di002.edv.uniovi.es](http://di002.edv.uniovi.es)
- Definicion. (s.f.). *Definicion de Plugin*. Obtenido de [www.definicion.de](http://www.definicion.de)
- Elen, R. (2001). *Ambisonics: The surround alternative*. Obtenido de [www.ambisonic.net](http://www.ambisonic.net)
- Espinel, L. F. (2014). *Desarrollo de un videojuego con reconocimiento de voz para la identificacion de colores y figuras geometricas*. Bogotá DC.
- ESRB. (s.f.). *Guia de clasificaciones de la ESRB*. Obtenido de [www.esbr.org](http://www.esbr.org)
- Freesound. (s.f.). *Free Library*. Obtenido de [www.freesound.org](http://www.freesound.org)
- G. Roy, J. J. (s.f.). *Técnicas de microfónica estéreo*. Obtenido de [www.analfatecnicos.net](http://www.analfatecnicos.net)
- JAM. (s.f.). *AudioGame Jam*. Obtenido de [jams.gamejolt.io/AGJam](http://jams.gamejolt.io/AGJam)
- Kronlachner, M. (2014). *Spatial Transformations for the alteration of ambisonic Recordings*. Austria.
- Microsoft. (s.f.). *Xbox: Kinect documentation*. Obtenido de [www.xbox.com](http://www.xbox.com)
- Oculus. (s.f.). *Oculus Audio SDK*. Obtenido de [ww.developer3.oculus.com](http://ww.developer3.oculus.com)
- S., Z. Z. (2005). *Bimodal HCI-Related affectec recognition*. Universidad de Illinois - EEUU.
- Shure. (s.f.). *Técnicas de microfónica estéreo*. Obtenido de [www.shure.es](http://www.shure.es)

- Sigchi. (s.f.). *Computer-Human Interaction*. Obtenido de <http://www.sigchi.org/>
- Simon, P. I. (2005 ). *El diseñador de sonido: Función y esquema de trabajo*. Obtenido de <http://www.pabloiglesiassimon.com/>
- Studio, F. (s.f.). *Guía de introducción a FMOD Studio*. Obtenido de [www.fmod.org](http://www.fmod.org)
- Zotter, M. K. (2014). *Spatial Transformations for the enhancement of ambisonic recording*. Austria.

