



ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN

“IMPLEMENTACION DE UN PROTOTIPO DE UNA BATERIA MUSICAL VIRTUAL”

INFORME DE MATERIA DE GRADUACION

Previo a la obtención del Título de:

**INGENIERO EN COMPUTACIÓN
ESPECIALIZACIÓN SISTEMAS TECNOLÓGICOS
INGENIERO EN COMPUTACIÓN
ESPECIALIZACIÓN SISTEMAS MULTIMEDIA**

Presentado por:

**LAURA MARIANELLA ZAPATA ASPIAZU
DIEGO ANDRES HERRERA SAMANIEGO**

GUAYAQUIL - ECUADOR

2010

AGRADECIMIENTO

A Dios, que nos permite seguir avanzando cada día, a nuestros padres, que sin el apoyo de ellos nos hubiésemos rendido en el camino, y a todas las personas que colaboraron en especial, a nuestro director Ph.D. Sixto García por guiarnos en esta ardua tarea.

DEDICATORIA

*A nuestros padres, familiares y
amigos.*

TRIBUNAL DE SUSTENTACIÓN



Ph.D. Sixto García.

PROFESOR DE LA MATERIA DE GRADUACIÓN.



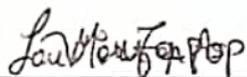
Ph.D. Xavier Ochoa.

PROFESOR DELEGADO DEL DECANO.

DECLARACIÓN EXPRESA

"La responsabilidad del contenido de este informe de materia de graduación, nos corresponde exclusivamente; y el patrimonio intelectual de la misma a la **Escuela Superior Politécnica del Litoral**"

(Reglamento de exámenes y títulos profesionales de la ESPOL)



Laura Zapata Aspiazu.



Diego Herrera Samaniego.

RESUMEN

Este documento presenta la implementación de un prototipo de una Batería Musical, por medio del uso de las tecnologías de Realidad Virtual y Multimedia. Se plantea crear un ambiente virtual 3D adecuado, que sea interactivo e inmersivo, que permita ejecutar un instrumento musical sencillo, con una adecuada sincronización entre las acciones del usuario, la retroalimentación visual y el sonido generado. La razón por la que se escogió a la Batería, es porque es una de las interacciones más fáciles de reproducir en un ambiente virtual además que es un instrumento que goza de nuestra preferencia.

Mediante el uso de un dispositivo sensor de posiciones (tracker), cuyos sensores se adjuntan a un par de palillos reales (baquetas), por medio de los cuales son capturados los movimientos motrices que hace el usuario y luego son generadas las imágenes que corresponden a estos movimientos, mostrándolas en una pantalla Head Mounted Display (HMD) que permite ver en tres dimensiones (3D).

Cada instrumento tiene asociado una imagen y un sonido en el escenario virtual, este sonido se reproduce mediante golpes que genera el usuario a través de las baquetas. Esta reproducción va asociada con diferentes niveles

de volumen, de tal forma que a mayor fuerza de golpe se eleva el volumen de reproducción.

En la primera parte de este documento se hace la introducción, se establecen los objetivos y justificativos, se detallan conceptos y definiciones concernientes a la realidad virtual y a los dispositivos usados. Luego se elabora el análisis de los distintos componentes que interactúan, se establecen los alcances y los requerimientos del sistema, se explica la arquitectura, herramientas de desarrollo, se detalla la etapa de implementación y las pruebas. Finalmente se presentan las conclusiones y recomendaciones del trabajo realizado.

ÍNDICE GENERAL

AGRADECIMIENTO	II
DEDICATORIA.....	III
TRIBUNAL DE SUSTENTACIÓN.....	IV
DECLARACIÓN EXPRESA	V
RESUMEN	VI
ÍNDICE GENERAL	VIII
ÍNDICE DE FIGURAS.....	X
ÍNDICE DE TABLAS.....	XI
INTRODUCCION.....	XII
1. JUSTIFICACION Y OBJETIVOS.....	1
1.2 Objetivo General	5
1.3 Objetivos Específicos.	5
2. FUNDAMENTOS TEORICOS	7
2.1 Realidad Virtual.	7
2.2 Despliegues Visuales Gráficos	8
2.3 Sistema de rastreo (Trackers)	14
3. REQUERIMIENTOS Y ANALISIS	20
3.1 Requerimientos Funcionales.	20
3.2 Requerimientos No Funcionales.	24
3.3 Diagrama General	24
3.4 Casos de Uso y Escenarios	27

3.5	Clases de análisis	30
3.6	Diagramas de Interacción de Objetos (DIO's) – Etapa Análisis.	32
4.	DISEÑO	36
4.1	Arquitectura General	36
4.2	Clases de diseño.....	40
4.3	Diagramas de Interacción de Objetos (DIO's) – Etapa Diseño.	42
5.	IMPLEMENTACION Y PRUEBAS	46
5.1	Herramientas de desarrollo.....	46
5.2	Instalación y Configuración de equipos.	48
5.3	Proceso de Implementación.....	50
5.4	Conexión física de los dispositivos de realidad virtual.....	54
5.5	Pruebas y Resultados.....	58
	CONCLUSIONES Y RECOMENDACIONES.....	63
	ANEXOS	67
	BIBLIOGRAFIA.....	73

ÍNDICE DE FIGURAS

Figura 2.1 Head Mounted Displays de marca eMagin [10].....	10
Figura 2.2 Head Mounted Display de marca 5DT [6]	10
Figura 2.3 Rotaciones en ejes X, Y y Z [6].....	15
Figura 2.4 Polhemus Liberty, 240/16 y 240/8 sensores [16].....	16
Figura 2.5 Polhemus Liberty, conexión del equipo [16].....	17
Figura 2.6 Panel posterior del Polhemus Liberty [16].....	17
Figura 2.7 Conexión de la fuente magnética [16].....	18
Figura 2.8 Conexión de un sensor [16]	18
Figura 3.1 Batería musical	21
Figura 3.2 Diagrama General.....	26
Figura 3.3 Diagrama de clases de análisis.....	31
Figura 3.4 Diagrama de interacción del Esterbil.....	32
Figura 3.5 Ejecución del Bombo	33
Figura 3.6 Ejecución de la vista panorámica.....	34
Figura 4.1 Arquitectura de la Batería Virtual.	37
Figura 4.2 Diagrama de clases de diseño	41
Figura 4.3 Ejecución del Esterbil.....	42
Figura 4.4 Ejecución del Bombo	44
Figura 4.5 Vista panorámica.	45
Figura 5.1 Activación de las configuraciones 3D en la Tarjeta de video.....	49
Figura 5.2 Clases desarrolladas.	50
Figura 5.3 Código para cargar la escena virtual.....	51
Figura 5.4 Escena virtual de la Batería	52
Figura 5.5 Código para establecer la salida en 3D.....	53
Figura 5.6 Polhemus, configuración de puertos	54
Figura 5.7 Diagrama de conexión de dispositivos	55
Figura 5.8 Sensores adjuntos a las baquetas	55
Figura 5.9 Sensor adjunto al head mounted display 5DT	56
Figura 5.10 Pedal USB usado para tocar el bombo	56
Figura 5.11 Ejecución del programa en 2D	57
Figura 5.12 Toma real de un usuario con el head mounted display	57
Figura 5.13 Toma real de un usuario con las baquetas.....	58
Figura 5.14 Latencia de la batería virtual	59
Figura A.1 Batería Virtual implementada en el año 2006	68
Figura A.2 Guitarra Virtual implementada en el año 2002.....	69

ÍNDICE DE TABLAS

Tabla 2.1 Estados de Polhemus Liberty [16]	19
Tabla 3.1 Rangos de frecuencias de los instrumentos	23
Tabla 5.1 Media, varianza y desviación estándar de las muestras de la latencia.....	61
Tabla 5.2 Los sentidos y la determinación de la inmersión.....	62
Tabla B.1 Tiempo entre la lectura de datos del Polhemus y el momento de colisión (muestras de la 1 -15).	71
Tabla B.2 Tiempo entre la lectura de datos del Polhemus y el momento de colisión (muestras de la.....	72
16-30).	72

INTRODUCCION

A lo largo de los tiempos, conforme la ciencia y la tecnología avanzan, se han desarrollado nuevas formas de expresarse musicalmente, desde los rústicos instrumentos de percusión y viento que usaban troncos y conchas para generar sonidos, hasta las avanzadas interfaces eléctricas y electrónicas utilizadas en la actualidad. Muchos artistas precisan de nuevas interfaces y controles que, aprovechando los avances tecnológicos, les permitan mayor libertad y les abran paso a nuevas expresiones musicales en su continuo afán de innovar en este arte. De la misma manera en que aparecieron los instrumentos eléctricos y revolucionaron la música con el rock, hoy se piensa que con los avances tecnológicos, los controles musicales podrían romper barreras que permitan al músico sumergirse en un ambiente donde no existan necesariamente las limitaciones de interacción con instrumentos físicos, y pueda aprovechar de mejor manera sus sentidos y movimientos para generar una nueva forma de expresión musical [14].

Algunas interfaces y controles musicales actuales, sobre todo aquellos de percusión, generan algunos problemas, porque no son ligeros y portables, su configuración y calibración en algunos casos puede

resultar costosa y consumir tiempo al momento de adaptarse a la comodidad y gustos específicos del músico. Algunos instrumentos ocupan mucho espacio o generan mucho ruido cuando se realizan prácticas. También podemos añadir que, para aquellas personas con alguna discapacidad, hay pocas alternativas de accesibilidad para tocarlos.

La implementación de instrumentos musicales con tecnología de realidad virtual no es nueva, existen baterías, guitarras, pianos [19]. En la figura A1 y A2 del anexo A, se puede observar a un usuario tocando la batería [12] y la guitarra virtual [13] respectivamente.

Este proyecto crea un prototipo de una Batería Virtual, en el cual se han aplicado los conocimientos adquiridos en la materia de graduación “Tecnologías de Realidad Virtual aplicada a la Multimedia”, gracias al cual se pudo experimentar con conceptos importantes de esta tecnología como lo son la inmersión, presencia e interacción.

La razón por la que se eligió a la Batería, es debido a que goza de nuestra preferencia. En el transcurso del desarrollo de este proyecto nos dimos cuenta que es un instrumento de fácil reproducción en el ambiente virtual, hubiese sido mas complejo implementar una guitarra o un piano virtual.

CAPÍTULO 1

1. JUSTIFICACION Y OBJETIVOS

1.1 Justificación.

Se están usando ampliamente las tecnologías de Realidad Virtual, para crear controles musicales, que no solamente replican las interfaces de los instrumentos reales, sino que además las extienden, al aprovechar las ventajas de interactuar con objetos que físicamente no están presentes, ofreciendo así mayor libertad, siempre y cuando la interacción con ellos resulte familiar y natural. Hoy en día se piensa en nuevas formas de expresión musical, que incluyan innovadoras formas de generar sonidos, siendo ahí donde la realidad virtual más que usarse para replicar una interacción con un instrumento real, se la está usando para crear ambientes inmersivos donde el usuario pueda

interactuar mas libremente en la generación musical. Artistas de vanguardia especialmente de música electrónica, experimentan continuamente con formas alternativas de hacer música en vivo, por lo que la realidad virtual puede significar una buena motivación para crear nuevos controles e interfaces musicales.

La realidad virtual permite al usuario sumergirse en un ambiente donde puede interactuar con objetos virtuales como si fueran reales, por medio de varios sentidos, especialmente la vista, el tacto y el oído. Al desarrollar interfaces usando la realidad virtual para el control musical, se puede mencionar algunas ventajas frente a los instrumentos reales y tradicionales:

- Mayor flexibilidad y facilidad para configurar y calibrar la interfaz a la comodidad y gustos del músico.
- Mejor control sobre los aspectos y variables musicales como afinación, tiempo, compás.
- Posibilidad de nuevas formas de expresión musical, donde los artistas puedan aprovechar de mayor libertad en sus movimientos y mejor retroalimentación de sus sentidos

- Se le puede brindar al usuario una retroalimentación visual o táctil de algunas características físicas de la reproducción del sonido (Ej.: la vibración de una cuerda).
- Se pueden crear aplicaciones colaborativas, donde personas en distintos lugares puedan tocar juntos.
- Crear aplicaciones para practicar y enseñar la ejecución de instrumentos.
- Disminución de espacio físico y del ruido generado por instrumentos reales.
- Creación de interfaces accesibles para que personas con alguna discapacidad, puedan expresarse musicalmente.

Entre las desventajas en cambio encontramos:

- Resulta complejo alcanzar un grado adecuado de precisión y sincronización dado el procesamiento que implica generar interacciones en realidad virtual.
- Modelos físicos del sonido ayudan a que las interfaces sean más sencillas y naturales, pero su implementación en un ambiente de realidad virtual complica la sincronización.

- Usuarios de instrumentos tradicionales pueden no encontrar interés en experimentar con controles virtuales
- La precisión espacial que ofrece la realidad virtual, puede no ser suficiente para ejecuciones musicales complejas.

El presente trabajo propone verificar la factibilidad del uso de la realidad virtual para controlar un instrumento musical, que resulte natural y familiar. Se plantea crear un ambiente virtual 3D adecuado, que sea interactivo e inmersivo, que permita ejecutar un instrumento musical sencillo, con una adecuada sincronización entre las acciones del usuario, la retroalimentación visual y el sonido generado. Además se analizarán los principales factores que permitan que la interfaz se convierta en una opción válida para el control musical.

El ambiente virtual consiste en una réplica de una batería, con la que se puede interactuar de manera similar que con una real, creando así un ambiente familiar para el control musical. La batería es una de las opciones más adecuadas para esta comprobación, dado que permite un amplio rango de movimientos, su interacción es sencilla ya que solo consiste en golpear las partes, cada una de las cuales genera un sonido diferente y a diferencia de otros instrumentos como por ejemplo

el piano, donde debe seguir sonando mientras se tiene presionada alguna tecla, la batería solo suena mientras es golpeada.

Para hacer inmersivo el ambiente, se usará un Head Mounted Display para visualizar la batería, además se usarán trackers para sensor los movimientos.

1.2 Objetivo General.

Implementar un ambiente de realidad virtual que permita al usuario interactuar con una Batería Musical.

1.3 Objetivos Específicos.

- Conseguir un grado adecuado de sincronización entre los movimientos y acciones del baterista, la retroalimentación visual y la generación del sonido.
- Lograr un equilibrio en la cantidad de trackers para no disminuir la precisión y sincronización que se requiere.
- Configurar adecuadamente el Head Mounted Display para visualizar correctamente la batería como si se tratase de una real.

- Crear una opción válida para ejecutar de manera natural y familiar un instrumento musical mediante la realidad virtual.

CAPÍTULO 2

2. FUNDAMENTOS TEORICOS

2.1 Realidad Virtual.

El término Realidad Virtual se refiere a la creación de reproducciones aceptables de objetos reales, con el objetivo de dar la ilusión de estar en un ambiente que puede ser percibido como real a tal punto que se pueda navegar e interactuar [1].

La navegación es la habilidad para moverse y explorar la escena, con interacción nos referimos a la selección y manipulación de objetos [1].

Hay dos factores que describen la experiencia de realidad virtual y son: la inmersión y la presencia.

La *inmersión* se refiere a la configuración física de la interfaz de usuario. No se tiene el mismo grado de inmersión al usar un proyector, un despliegue visual gráfico personal o una computadora. La inmersión de la puede clasificar en: completamente inmersiva, semi inmersiva, no inmersiva.

La *presencia* es un concepto subjetivo, está asociada a la psicología del usuario, se la puede describir como el estado de conciencia de estar en un ambiente virtual.

La interacción envuelve múltiples canales sensoriales, como lo son el visual, auditivo, táctil, olfato y gusto.

Para alimentar el sentido de la vista, se usan pantallas 3D estéreo, el audio por medio de sonidos interactivos. La retroalimentación táctil se la efectúa por medio de pantallas vibro táctiles, táctiles (*eléctricas, termales, mecánicas*) y kinestésicas; en cambio, para el caso de los sentidos del olfato y el gusto aun se están haciendo estudios y pruebas [20].

2.2 Despliegues Visuales Gráficos.

La visión fue uno de los primeros sentidos del ser humano en ser simulado en la realidad virtual debido a que es reconocido como la primera fuente de información hacia el mundo externo [1].

Además, este sentido es comúnmente usado para hacer más exacta la interpretación de otros, por ejemplo, miramos hacia una fuente inesperada de sonido, como la sirena de una ambulancia, para verificar su localización. Todos los sentidos son relevantes, pero la visión es uno de los más usados, la realidad virtual usa algunas tecnologías de presentación de gráficos, a los que se los denomina despliegues visuales gráficos.

Un dispositivo de despliegue visual gráfico muestra imágenes sintéticas a uno o muchos usuarios que interactúan con el mundo virtual, por ejemplo, los proyectores se usan para mostrar imágenes a una gran cantidad de personas de forma simultánea, hay otros equipos que se usan de forma individual como es el caso de los Head Mounted Display.

Las imágenes desplegadas pueden ser monoscópicas, es decir que no tienen profundidad, se las conoce como imágenes en dos dimensiones (2D). Las imágenes estereoscópicas, si tienen profundidad y se las conoce como imágenes en tres dimensiones (3D).

Despliegues Visuales Gráficos Personales:

Es un dispositivo que muestra una escena virtual a un solo usuario.

Dentro de la categoría de los displays personales se encuentran los head-mounted displays (*HMDs*) y los face-mounted displays (*FMDs*).

Head Mounted Displays (HMD) y Face Mounted Displays (FMD).

El Head Mounted Displays o Helmet Mounted Display, ambos se abrevian HMD, es un dispositivo que se usa en la cabeza o como parte de un casco [3], los Face Mounted Display se montan sobre el rostro [4].



Figura 2.1 Head Mounted Displays de marca eMagin [10]



Figura 2.2 Head Mounted Display de marca 5DT [6]

Los HMD monoculares tienen una sola pantalla en un ojo, en cambio los binoculares lo tienen en ambos ojos [2].

Un HMD típico tiene una o dos pantallas en miniatura con lentes y espejos semi-transparentes incrustados en un casco, gafas o visera. La tecnología más usada para la fabricación de las pantallas es la de cristal líquido (LCD), la cual es liviana, compacta y económica. Existen modelos antiguos que usan tubo de rayos catódicos (CRT), pero desafortunadamente son pesados y no son compactos [18].

Parámetros de funcionamiento:

Imágenes estereoscópicas.- Las imágenes estereoscópicas son producidas en pares, cada imagen muestra la misma escena del objeto desde diferentes ángulos de visión que corresponden a los ángulos de visión de ambos ojos del ser humano.

Y es posible solo en la visión binocular, el cual requiere que en el ojo izquierdo se muestre una imagen ligeramente diferente de la del

derecho. En el cerebro, las diferentes percepciones de la imagen son combinadas e interpretadas en términos de profundidad.

Campo de visión (Field of View FOV) – Una importante característica del sistema de visión humana es el campo de visión, esta es aproximadamente de 150° horizontalmente y 120° verticalmente cuando se usa 1 ojo, y crece a 180° horizontal y 120° vertical cuando se usan ambos ojos. Una parte central de este volumen de visión representa el área de estereopsis, donde se registra la misma imagen. La mayoría de HMD ofrece considerablemente menos que esto. Grandes campos de visión generan un gran sentido de inmersión [5].

Resolución.- es la cantidad de píxeles que tiene el display, el píxel (picture element) es la menor unidad homogénea de color que puede mostrar el display [5].

Granularidad.- expresada en arcminutos/píxeles o en píxeles por grado, es en otras palabras la densidad del píxel, es usada para determinar la agudeza visual (capacidad para ver claramente, para distinguir la forma y los detalles precisos de los objetos en condiciones de iluminación adecuados [7]), los HMD típicamente ofrecen de 10 a 20 píxeles/° [5].

Traslape Binocular.- O traslape, mide el área que es común a ambos ojos donde las imágenes se superponen. Es la base de la sensación de profundidad y estéreo, permitiendo al humano captar objetos cercanos y lejanos. Los humanos tenemos un traslape binocular de alrededor 100° (50° a el lado izquierdo de la nariz y 50° al derecho). Traslape es muchas veces especificado en grados o en porcentaje que indica cuanto del campo visual de cada ojo es común al otro [5].

Distancia de foco (Collimation).- Colimación es la alineación de los elementos de óptica de los binoculares a la mecánica del eje. Una buena colimación evita el cansancio ocular, dolores de cabeza, inferior y ver doble las imágenes al tiempo que mejora la resolución. Lamentablemente, la colimación adecuada es casi imposible de conseguir en muy bajo precio [5].

Distancia Entre Pupilas (Inter-Pupillary Distance IPD).- Es la distancia entre el centro de las pupilas de cada ojo, usualmente en milímetros, usada para especificar el tamaño del rango para sistemas ópticos [5].

La pupila es un orificio situado en la parte central del iris (parte de color) para el paso de la luz.

Periféricos del HMD.

Head Mounted Displays pueden incorporar un sistema de posicionamiento que rastrea las posición y el ángulo de la cabeza del usuario, de esta forma la imagen es congruente con el mundo externo.

Rastreo de cabeza (head tracking), son sensores que permiten el registro de cambios de ángulos y orientación. Puede ser usado para generar la imagen computarizada apropiada para el ángulo de vista en ese periodo particular, permite mirar 'al alrededor' del ambiente virtual.

Rastreo ocular (Eye tracking).- Mide la localización de la mirada, permitiendo a una computadora sensar hacia donde el usuario esta mirando. Esta información es útil en una amplia variedad de contextos, por ejemplo, se puede cambiar la información mostrada en pantalla brindando detalles adicionales.

2.3 Sistema de rastreo (Trackers).

Es el hardware usado en los sistemas de realidad virtual para medir un cambio de la posición y orientación en 3D de un objeto en tiempo real.

Un objeto en movimiento en el espacio 3D tiene 6 grados de libertad, 3 para traslaciones y 3 para rotaciones. Si un sistema de coordenadas cartesianas es adjuntado a un objeto en movimiento entonces sus traslaciones son a lo largo de los X, Y y Z ejes. Las rotaciones alrededor de estos ejes se llaman pitch, yaw, y roll respectivamente [5].

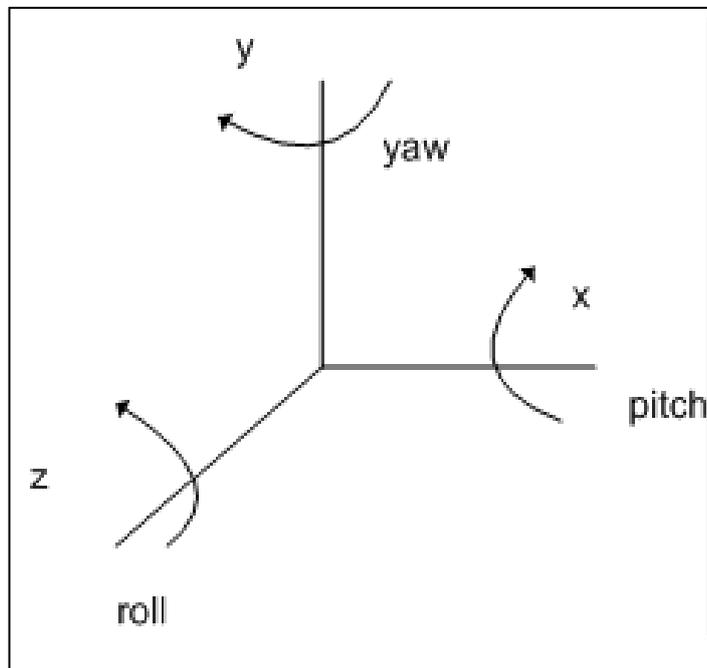


Figura 2.3 Rotaciones en ejes X, Y y Z [6]

Polhemus Liberty.

Es el nombre del equipo que usamos para desarrollar el aplicativo, es un sistema tracker en 3D. El modelo usado es el Liberty 240/8, el cual soporta el uso de 8 sensores cada uno opera hasta 240 Hz, existe otro modelo que se puede extender a tiene 16 sensores [8,17].



Figura 2.4 Polhemus Liberty, 240/16 y 240/8 sensores [16]

Este dispositivo se lo puede conectar al computador por medio de un cable USB o mediante un puerto de comunicaciones usando el conector RS-232. En la figura 2.5 le mostramos el panel posterior.

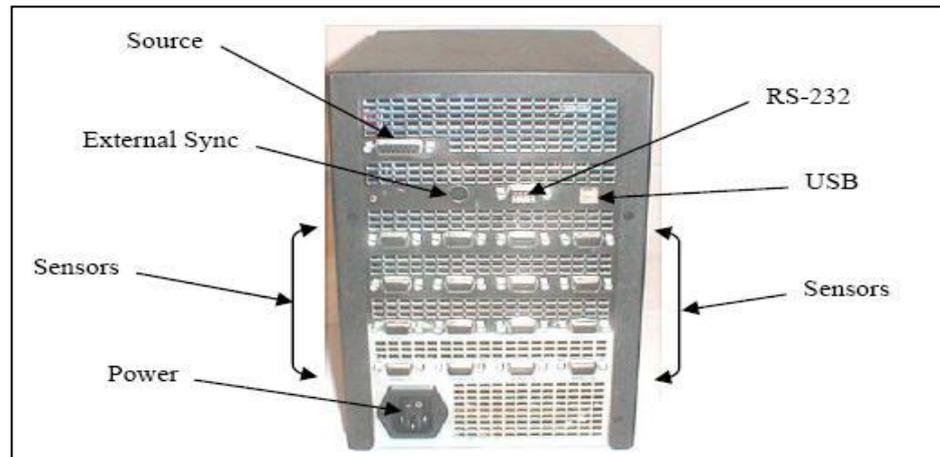


Figura 2.5 Polhemus Liberty, conexión del equipo [16]

En la figura 2.6 se muestra la conexión de los diversos componentes del Tracker.



Figura 2.6 Panel posterior del Polhemus Liberty [16]

La conexión de la fuente magnética se la mostramos en la figura 2.7.



Figura 2.7 Conexión de la fuente magnética [16]

La conexión de un sensor en uno de los puertos se la mostramos en la figura 2.8.



Figura 2.8 Conexión de un sensor [16]

En la parte frontal y trasera del dispositivo se encuentra un LED, o foco indicador de los diversos estados en los que el dispositivo tracker se encuentra.

Color	Estado Operacional	Estado
Verde permanente.	Sistema operacional con una Buena calidad de señal magnética.	Sin distorsión.
Rojo/verde intermitente.	Sistema operacional con una señal magnética marginal.	Distorsión menor.
Rojo permanente.	Sistema operacional con señal magnética pobre.	Distorsión significativa.
Rojo intermitente.	Falla en la auto-comprobación y configuración.	

Tabla 2.1 Estados de Polhemus Liberty [16]

CAPÍTULO 3

3. REQUERIMIENTOS Y ANALISIS

3.1 Requerimientos Funcionales.

Se requiere el desarrollo de una aplicación de software que simule por medio de dispositivos de realidad virtual el funcionamiento de una batería real. A continuación detallamos las funcionalidades que el sistema debe cumplir:

Escena virtual.

Se debe crear un escenario virtual en tres dimensiones -3D- en el cual deben constar todos los instrumentos musicales que conforman la

batería. Estos instrumentos se agrupan en dos tipos: tambores y platillos, esto se lo puede observar en la figura 3.1.

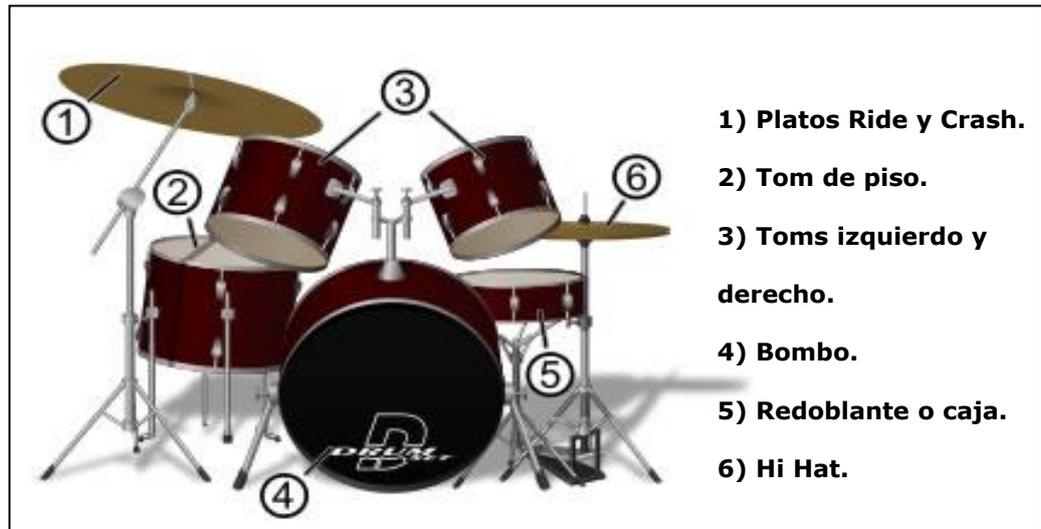


Figura 3.1 Batería musical

El escenario virtual debe constar de los siguientes elementos:

Baquetas (palillos): Son los palillos con los que se golpea a la batería musical y son dos: baqueta Izquierda y baqueta Derecha.

Tambores:

- Bombo (Bass Drum).
- Tom de Piso (Floor Tom).

- Tom Izquierdo (Left Tom).
- Tom Derecho (Right Tom).
- Caja o Redoblante (Snare Drum).

Platillos:

- Plato Crash (Crash Cymbal).
- Plato Ride (Ride).
- Esterbil (Hi-Hat).

Movimiento de las Baquetas.

Esta funcionalidad permitirá que las baquetas virtuales se muevan de la misma forma en que lo hacen las baquetas reales, se deberá usar un dispositivo de realidad virtual que determine las posiciones de las baquetas reales en el transcurso del tiempo.

Movimiento de la Cabeza.

Esta funcionalidad permitirá al usuario tener una visión panorámica de la escena virtual, la relación entre el movimiento de la cabeza y el movimiento de la escena deberá ser de 1 a 1, es decir, si el usuario mueve su cabeza 1 grado, la escena deberá moverse 1 grado.

Reproducción de sonidos musicales.

Al producirse un golpe con las baquetas sobre la batería, debe reproducirse el sonido asociado al instrumento golpeado y se debe mostrar una retroalimentación visual del golpe generado.

En la tabla 3.1 se muestra el rango de frecuencia de los instrumentos musicales que son parte de la batería, estos sonidos deben ser grabaciones de una batería real y no sonidos sintetizados o generados por la computadora.

Instrumento	Fundamental (Hz)	Armónicos (KHz)
Tambor	100-200	1-20
Bombo	30-147	1-6
Platillos	300-587	1-15

Tabla 3.1 Rangos de frecuencias de los instrumentos

Reproducción de Bombo.

El bombo es el único instrumento de la Batería que se lo toca con un pedal, se debe implementar su funcionamiento con un pedal de los que se usan para juegos de computadora.

3.2 Requerimientos No Funcionales.

Escalabilidad: El sistema debe ser escalable con respecto a las partes instrumentales de la batería, de tal forma que se puedan incorporar nuevos elementos de percusión a la escena.

Eficiencia en la reproducción de los sonidos musicales: Se debe garantizar un buen desempeño de la reproducción del audio, especialmente cuando dos o más instrumentos se golpean simultáneamente, por ejemplo: si se golpea el redoblante y el esterbil al mismo tiempo, ambos sonidos deben ser reproducidos simultáneamente y no se debe esperar que uno finalice para empezar el otro.

Eficiencia: el manejo de la visualización de la aplicación debe ahorrar recursos de hardware para su mejor desempeño.

Rendimiento: el sistema debe ser capaz de dar respuesta a los movimientos motrices del usuario minimizando el tiempo de respuesta de la aplicación.

3.3 Diagrama General

Hemos elaborado un diagrama general del aplicativo, lo hemos separado en tres niveles debido a que muchas actividades tienen un comportamiento similar, el de interface con el usuario, el de lógica de

la aplicación y el de acceso a datos, tal como lo podemos observar en la figura 3.2.

En el nivel de ***interface con el usuario*** se hace el uso de un dispositivo que permite determinar el desplazamiento de las baquetas, estos datos ingresan como entradas del siguiente nivel.

En el nivel de ***Lógica de la Aplicación*** vamos a desarrollar dos componentes, el visual y de reproducción de audio.

El componente visual hace uso de los datos provistos por la capa superior, realiza el movimiento de los objetos baquetas y se espera por una colisión con un objeto musical para hacer la retroalimentación visual.

El componente de reproducción de audio se ejecuta cuando ocurre una colisión, reproduciendo el sonido que nos da la capa de acceso a datos.

En el nivel de ***Acceso a Datos***, obtenemos el archivo requerido en la colisión.

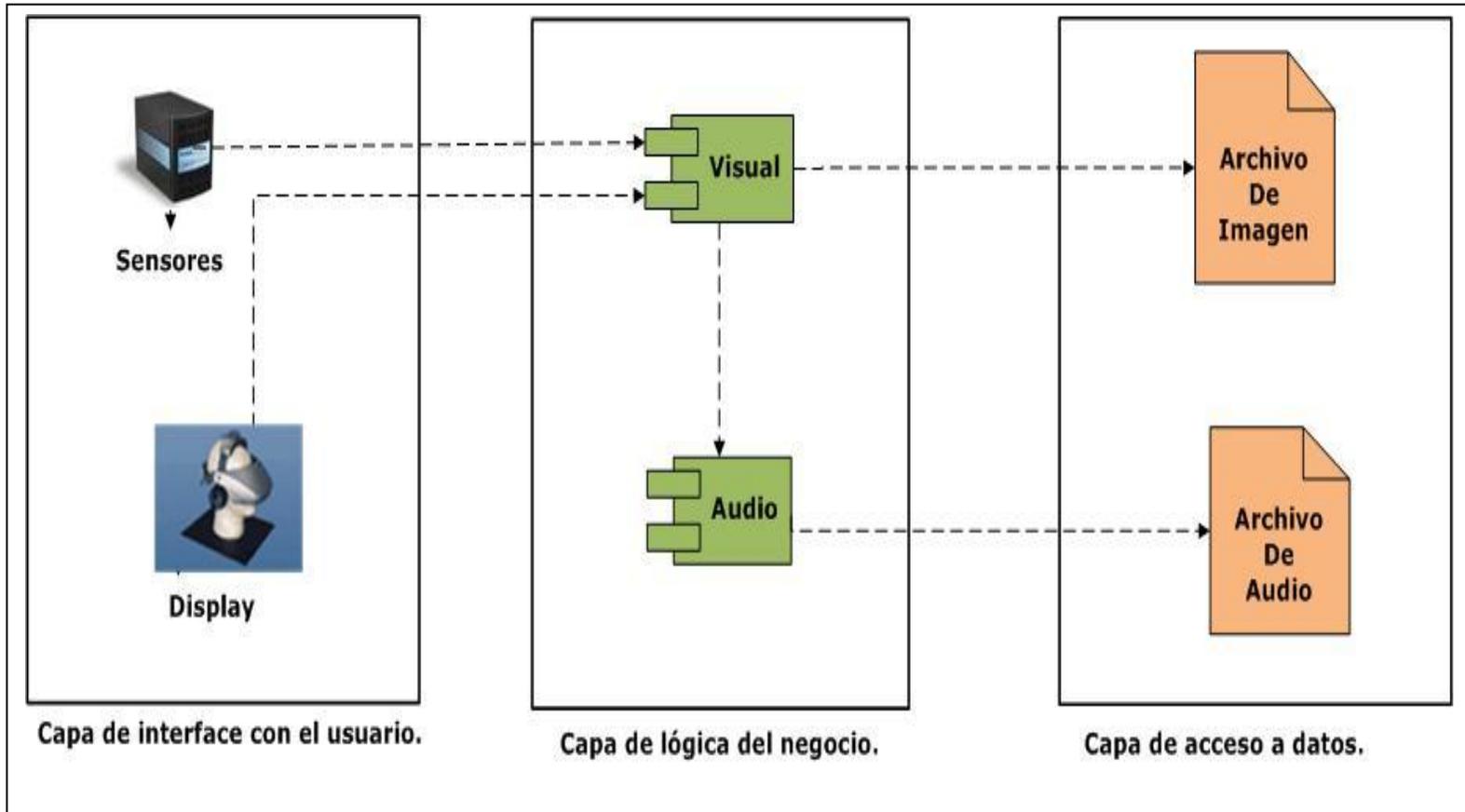


Figura 3.2 Diagrama General

3.4 Casos de Uso y Escenarios.

A continuación se presenta un listado de los casos de uso surgidos en la etapa de análisis.

1. Reproducción del Esterbil.
2. Reproducción del Crash.
3. Reproducción del Ride.
4. Reproducción del Tom de piso.
5. Reproducción del Tom izquierdo.
6. Reproducción del Tomb derecho.
7. Reproducción de la caja o redoblante.
8. Reproducción del Bombo.
9. Visión panorámica de la escena 3D.

De los nueve casos de uso, solo vamos a detallar tres (1, 8 y 9), debido a que los casos de uso del uno al siete son similares entre si.

Reproducción del Esterbil.

Reproducción del Bombo.

Visión panorámica de la escena 3D.

Caso de Uso: Reproducción del Esterbil.

Escenarios 1.1: Reproducción exitosa del esterbil de la batería.

Supuestos:

- El usuario mueve la baqueta con los sensores adjuntos a la misma.
- La baqueta colisiona con el objeto Esterbil.

Salidas:

- Se reproduce el sonido asociado al esterbil.
- Se esconde el objeto esterbil inicial y se muestra un esterbil que varia del primero solo en el color (retroalimentación visual).
- Se esconde el objeto esterbil asociado al golpe y se muestra el original.

Caso de Uso: Reproducción del Bombo.

Escenarios 1.2: Reproducción exitosa del bombo de la batería.

Supuestos:

- El usuario presiona el pedal.

Salidas:

- Se reproduce el sonido asociado al bombo.
 - Se esconde el objeto bombo inicial y se muestra un bombo que varia del primero solo en el color (retroalimentación visual).
 - Se esconde el objeto bombo asociado al golpe y se muestra el original.

Caso de Uso: Visión panorámica de la escena 3D.

Escenarios 1.3: Visión panorámica de la escena 3D exitosa.

Supuestos:

- El usuario tiene un sensor de movimiento adjunto al head mounted display.

Salidas:

- Se muestra la escena en una posición diferente a la inicial, dependiendo del ángulo en que movió la cabeza.

3.5 Clases de análisis

Las clases participantes en la interacción son:

CPDIDev.- Es una clase que pertenece a las librerías PDI, Polhemus Device Interface, esta fuera de nuestro diseño. Representa al dispositivo físico.

PDIconsole.- Esta clase representa la entidad que establece la comunicación entre el dispositivo Polhemus Liberty y nuestro sistema de batería virtual.

Osg.- Es la clase encargada de establecer la escena 3D y todos los métodos que se pueden ejecutar en la interfaz gráfica, como por ejemplo las colisiones.

MyOsgNodoMusical.- Entidad que representa a cada instrumento musical como tambores y platillos e inclusive a las baquetas. Tiene asociado un sonido y su malla (imagen) correspondiente, si es un elemento baqueta no tiene sonido.

En la figura 3.3 se muestra la relación entre las clases.

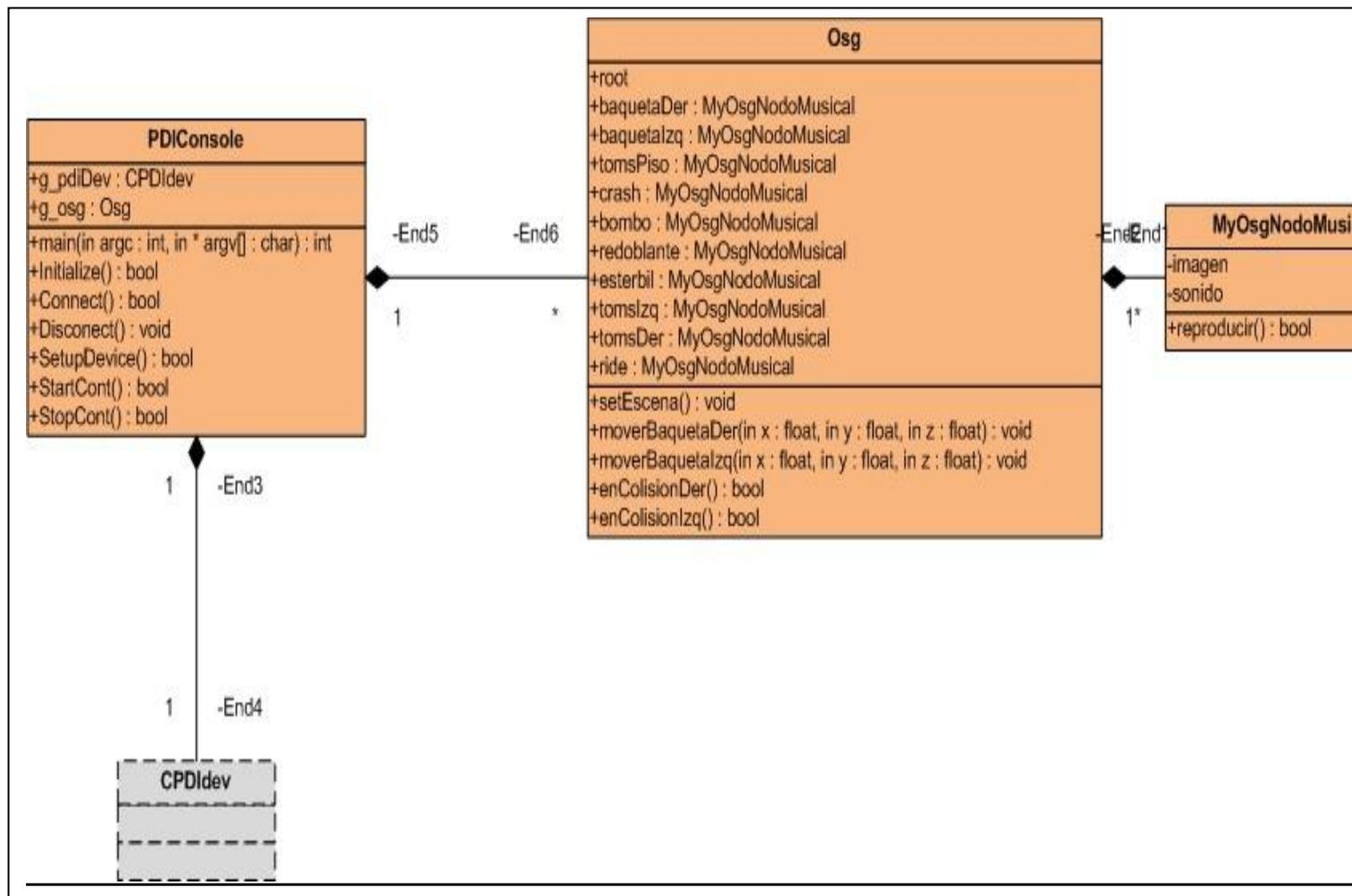


Figura 3.3 Diagrama de clases de análisis

3.6 Diagramas de Interacción de Objetos (DIO's) – Etapa Análisis.

Los siguientes diagramas de interacción de objetos están basados en los escenarios de la sección anterior (3.5).

Escenario 1.1: Ejecución exitosa del esterbil.

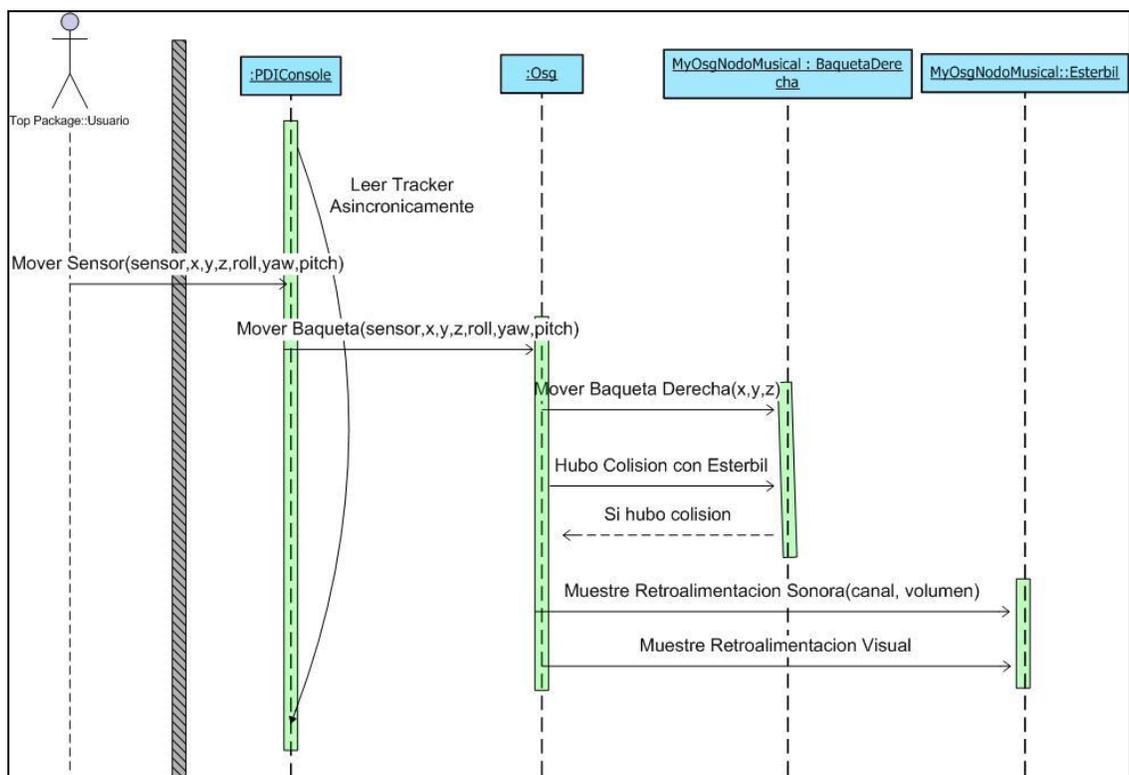


Figura 3.4 Diagrama de interacción del Esterbil

Se ejecutan lecturas continuas del tracker, las cuales registran los sensores conectados y las posiciones sobre las coordenadas 'x', 'y', 'z', y

sus giros respectivos 'roll', 'yaw', 'pitch'. Cuando un usuario hace un movimiento de la baqueta -sobre la cual está pegado un sensor del tracker-, se realiza el cambio de posición geométrica de la imagen asociada a la baqueta, luego se ejecuta un proceso para determinar si hubo una colisión de la imagen de la baqueta con la imagen de un instrumento, si se da la colisión se reproduce el sonido que corresponde al instrumento musical y se muestra una retroalimentación visual que representa el golpe dado.

Escenario 1.2: Ejecución exitosa del bombo

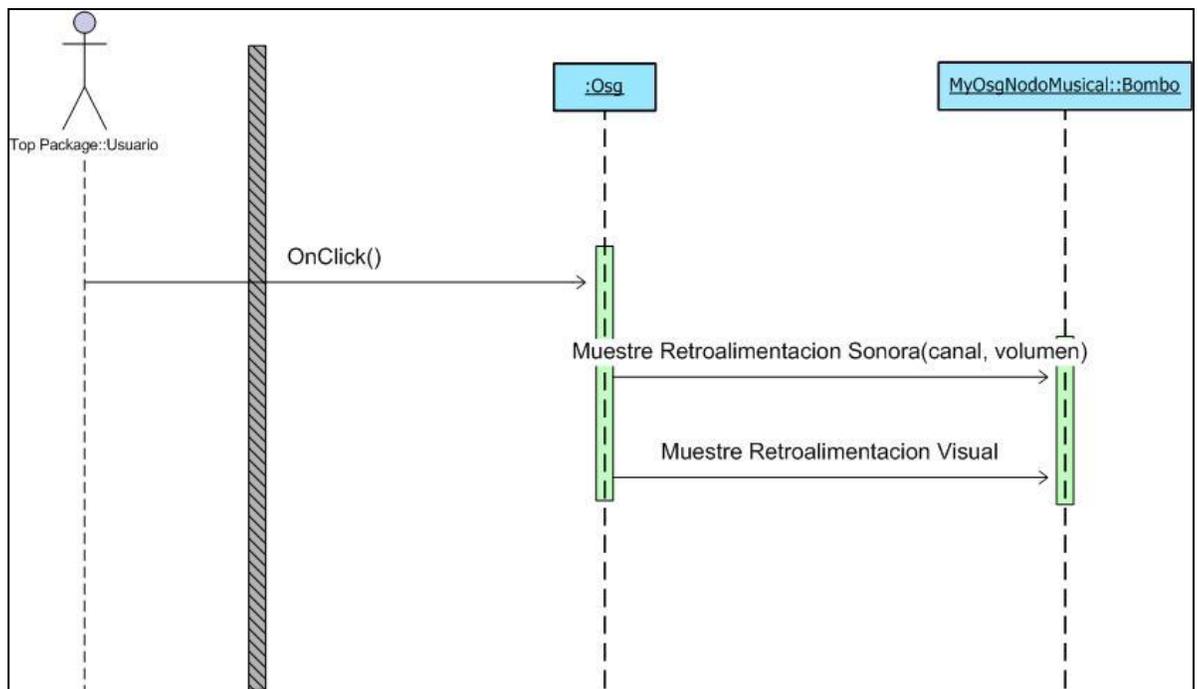


Figura 3.5 Ejecución del Bombo

La ejecución del bombo difiere de los otros instrumentos debido a que se la realiza por medio de un pedal.

Al presionar el pedal, este emite una señal al computador, que actúa como un click, entonces al producirse el evento se reproduce el sonido del bombo y se muestra la retroalimentación visual.

Escenario 1.3: Visión panorámica de la escena 3D.

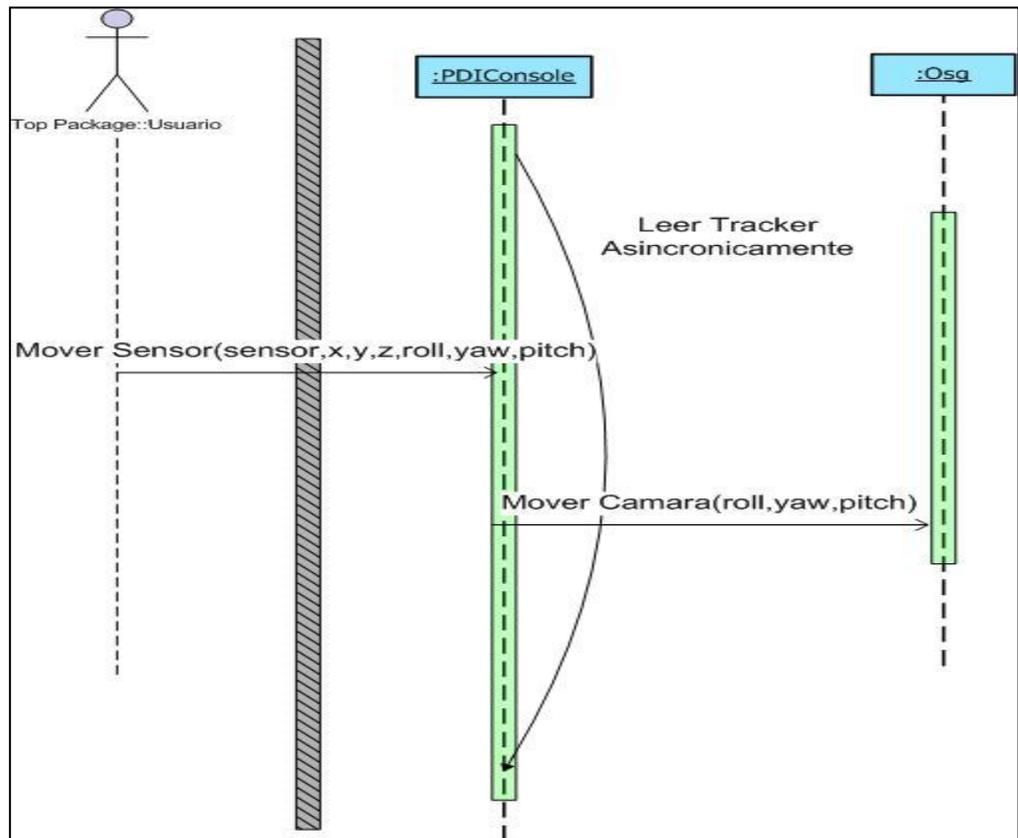


Figura 3.6 Ejecución de la vista panorámica

Se ejecutan lecturas continuas del tracker, las cuales registran los sensores conectados y las posiciones sobre las coordenadas 'x', 'y', 'z', y sus giros respectivos 'roll', 'yaw', 'pitch'.

Cuando un usuario hace un movimiento de la cabeza -sobre la cual está pegado un sensor del tracker-, se realiza el cambio de posición geométrica de la cámara de la escena. El campo de visión que tiene el usuario dentro de la escena virtual es de 140 grados en la horizontal y 140 grados en la vertical y la relación del movimiento real y el generado es de 1 a 1, es decir, si el usuario gira 1 grado, la escena se mueve 1 grado.

CAPÍTULO 4

4. DISEÑO

4.1 Arquitectura General

La figura 4.1 muestra la arquitectura de nuestro sistema, donde se pueden observar los componentes de cada nivel que permiten cumplir con los requerimientos funcionales establecidos en la etapa de análisis.

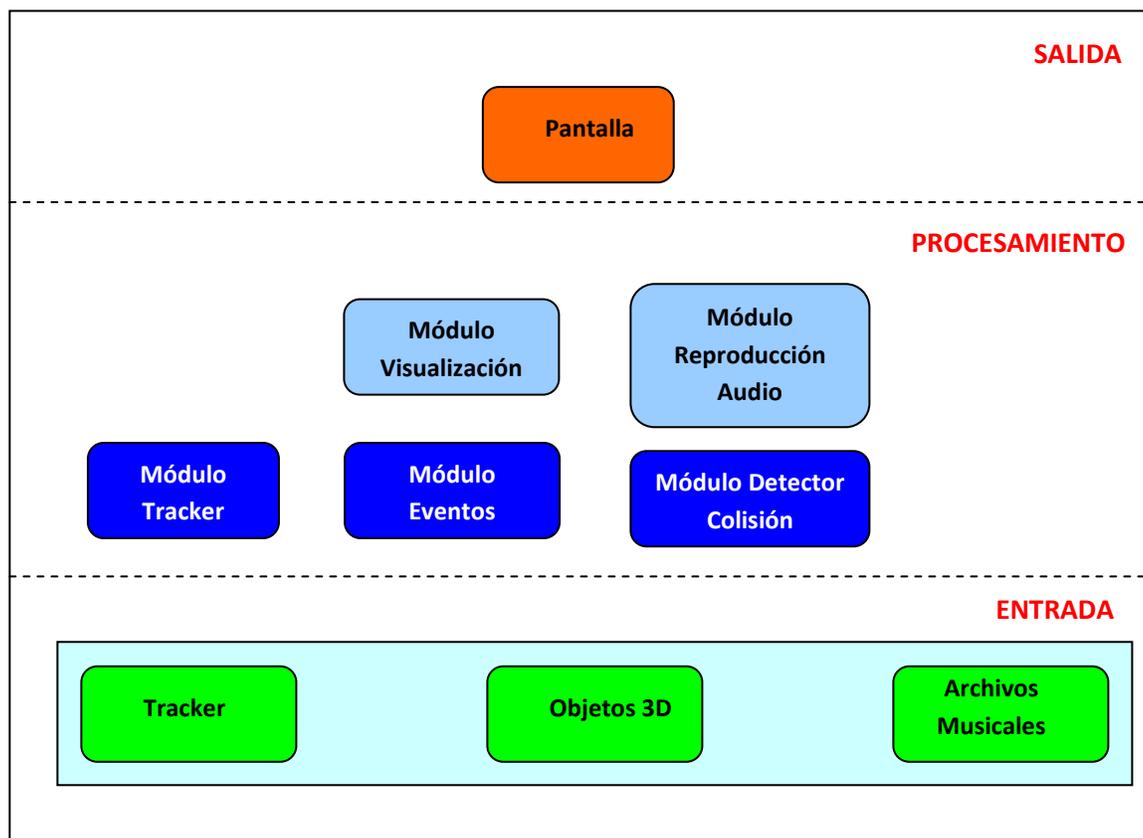


Figura 4.1 Arquitectura de la Batería Virtual.

A continuación detallamos los procesos que se realizan en cada nivel.

Modulo de Visualización

Carga de la Escena 3D en el ambiente virtual.

Por medio de la librería Open Scene Graph (OSG) cargamos la escena virtual con las imágenes de los objetos musicales y las baquetas, además se hace la iluminación del ambiente. Se

establecen los parámetros de configuración 3D para que el dispositivo de despliegue visual pueda mostrar imágenes en tres dimensiones y se ubica la cámara en posición inicial.

Retroalimentación Visual

Al momento de producirse una colisión entre la baqueta y la batería, además de reproducir el sonido musical correspondiente, se cambia de color al objeto que ha sido golpeado, de esta forma el usuario recibe una retroalimentación visual del objeto con el que interactúa.

Módulo Tracker

Con la librería propia del dispositivo de posicionamiento “Polhemus Liberty” (PDI), obtenemos las coordenadas X, Y y Z que registran los cambios de movimiento de las baquetas; estas coordenadas son usadas para mover sus correspondientes imágenes en la escena virtual.

Módulo de Eventos

La función principal de este módulo es poder tocar el bombo por medio de un pedal, que actúa como evento click del ratón del computador.

Además adiciona la funcionalidad necesaria para poder tocar la batería completa sin el dispositivo de rastreo "Tracker".

Módulo Detector de Colisiones

Este módulo es usado para detectar colisiones, las cuales ocurren cuando hay una intersección entre dos objetos. Cuando existe una colisión entra en funcionamiento el modulo de reproducción de audio.

Módulo de Reproducción de Audio

Reproducimos el sonido asociado al objeto musical que el usuario golpea por medio del uso de colisiones de imágenes. Esta reproducción va asociada con diferentes niveles de volumen, de tal forma que a mayor fuerza de golpe elevamos el volumen de reproducción.

Cada instrumento musical tiene un canal de reproducción individual, con lo que se logra independencia entre los sonidos, por lo que no hay que esperar que finalice uno para iniciar otro sonido.

4.2 Clases de diseño.

En la figura 4.2 se muestran las clases participantes en el diseño, a continuación se detallan las *nuevas* clases que surgieron después del análisis.

ElMouseEventHandler.- Hereda de `GUIEventHandler`, la utilizamos para capturar el evento `onClick()`, que se dispara cuando el usuario presiona el pedal del bombo.

PlayAudio.- Esta clase utiliza la librería `SDL_Mixer` la cual abre el dispositivo de audio con los parámetros deseados, además usamos un canal diferente para reproducir cada sonido, también se le envía un nivel de volumen que va de 0 a 128.

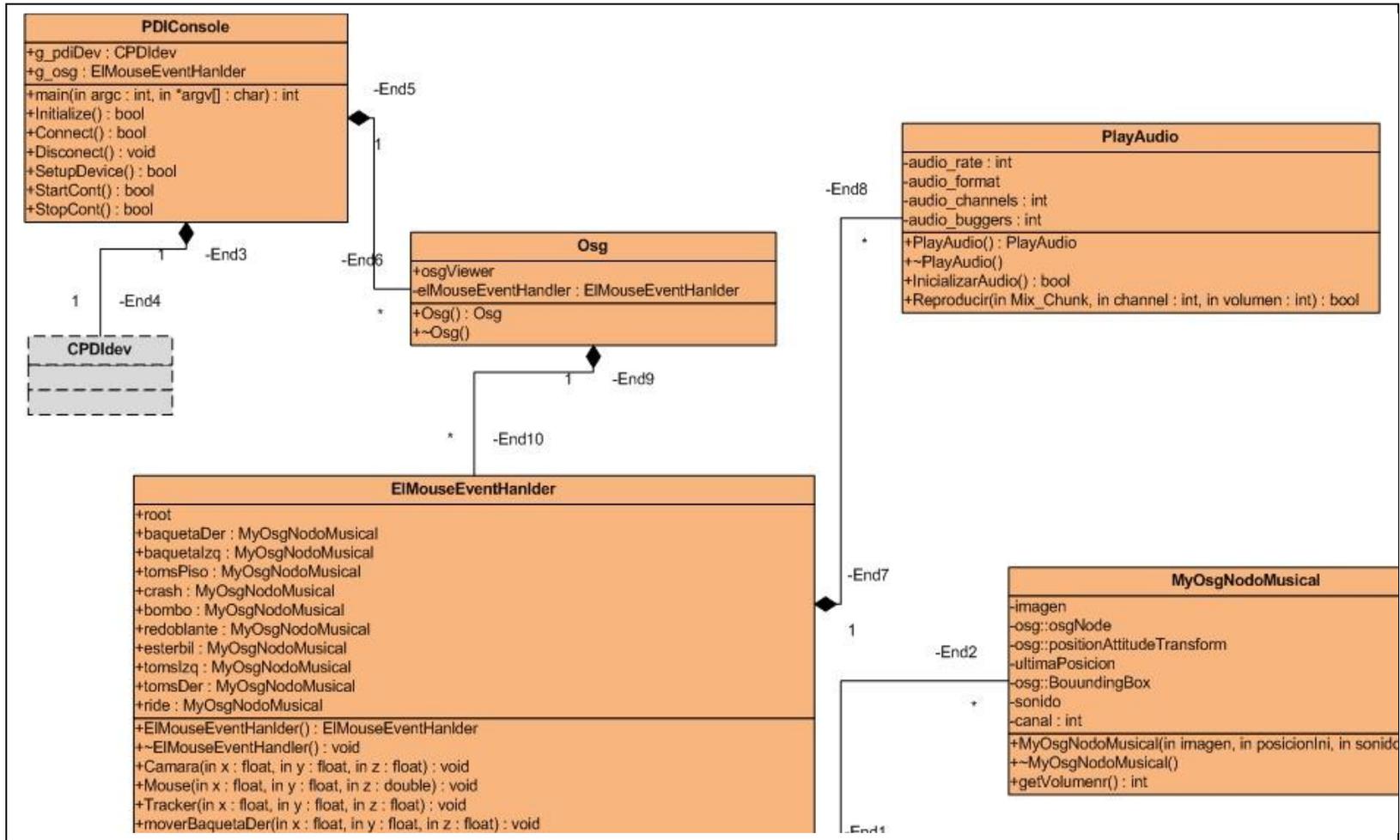


Figura 4.2 Diagrama de clases de diseño

4.3 Diagramas de Interacción de Objetos (DIO's) – Etapa Diseño.

A continuación se presentan los diagramas de interacción de objetos de diseño, obtenidos a partir de los diagramas de análisis presentados en la sección 3.5.

Escenario 1.1: Ejecución exitosa del esterbil.

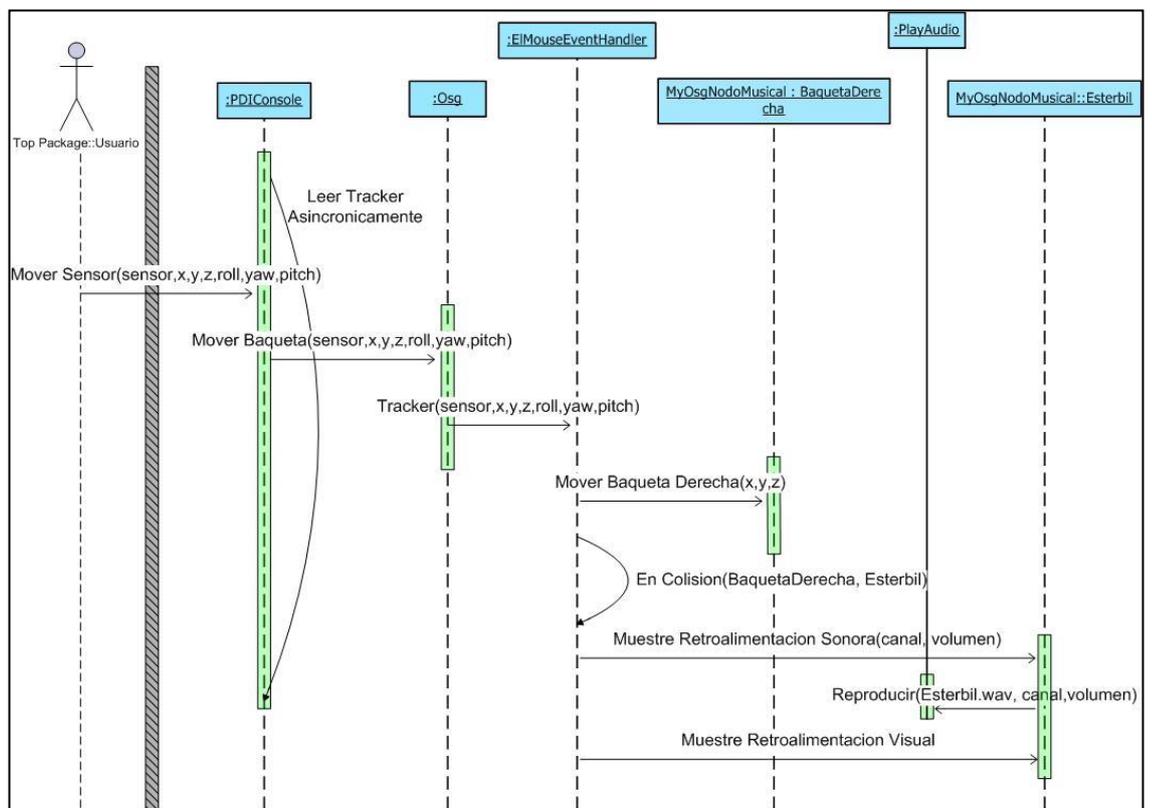


Figura 4.3 Ejecución del Esterbil

En la etapa de diseño surgieron 2 clases nuevas, MouseEventHandler y PlayAudio, ver figura 4.3.

Añadimos la clase `ElMouseHandlerEvent`, debido a que la ejecución del programa no solo se la realiza con los sensores, sino que también se la puede realizar con los eventos del Mouse, esto lo hicimos debido a que no siempre contábamos con la disponibilidad de los equipos de realidad virtual.

En etapa de análisis se estableció que cada instrumento iba a reproducir el archivo de audio asociado al mismo, pero debido a que la librería de audio abre flujos de datos para poder realizar la reproducción de la música, hicimos una sola instancia de la clase `PlayAudio`, la cual abre un solo flujo al inicio y cierra el mismo al salir de la aplicación, esta es una forma optima de su uso.

Escenario 1.2: Ejecución exitosa del bombo

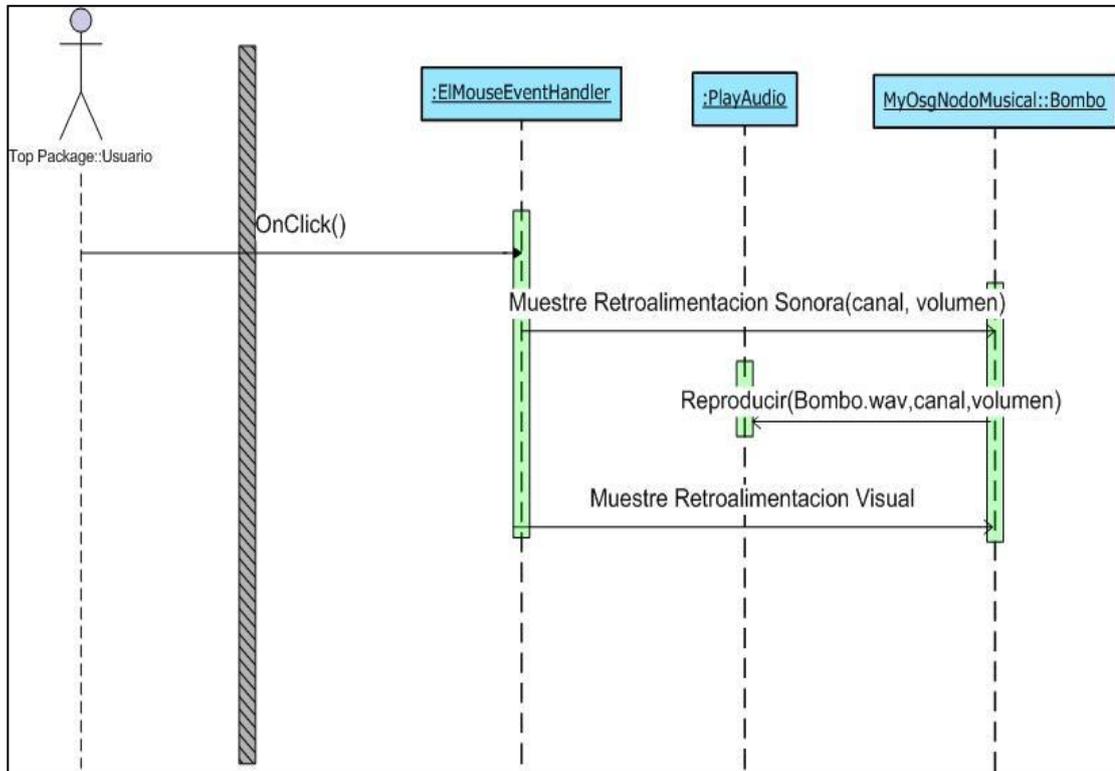


Figura 4.4 Ejecución del Bombo

Si comparamos los diagramas de análisis (figura 3.3) y diseño (figura 4.4), la clase Osg no interviene en esta interacción debido a que el pedal actúa como un evento click, el cual es manejado por la clase `ElMouseEventHandler`.

Escenario 1.3: Visión panorámica de la escena 3D

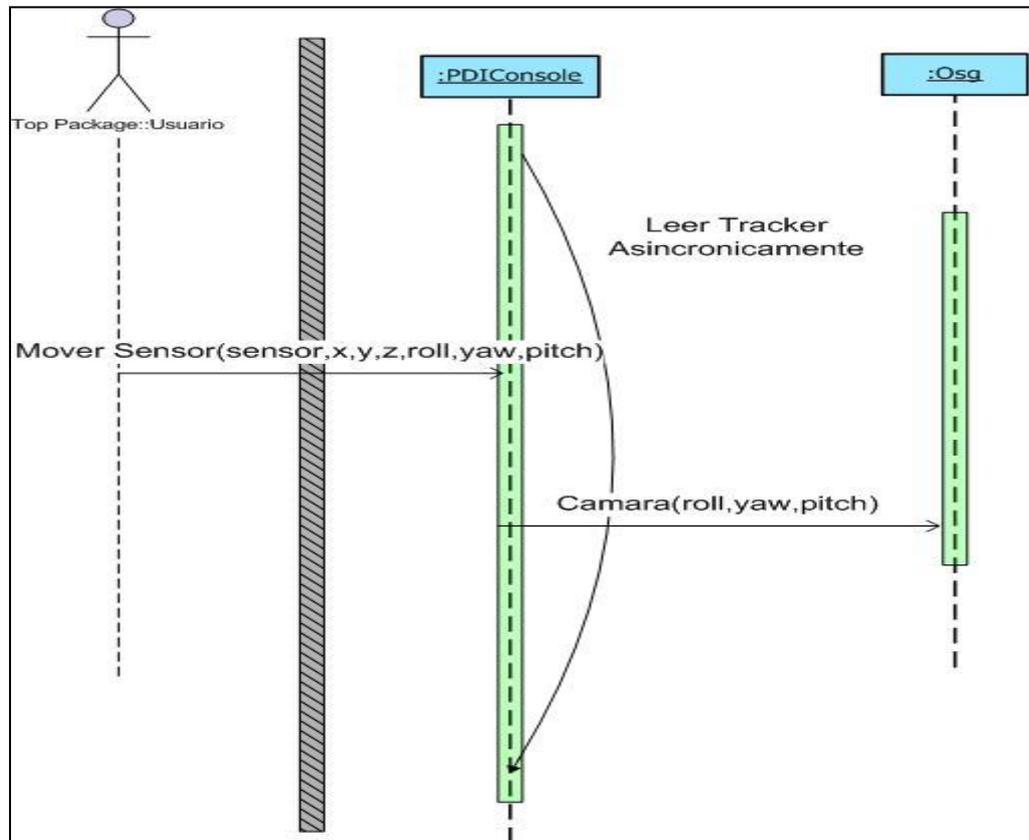


Figura 4.5 Vista panorámica.

Este diagrama de interacción se mantuvo igual al que surgió en la etapa de análisis, esto se debe a que no intervienen eventos del ratón de la computadora y tampoco existe preproducción musical.

CAPÍTULO 5

5. IMPLEMENTACION Y PRUEBAS

5.1 Herramientas de desarrollo.

A continuación vamos a detallar las herramientas de software que usamos para desarrollar esta aplicación, algunas dependen de los equipos de realidad virtual que usamos.

Polhemus Developer Interface (PDI)

Es una librería usada para administrar y capturar datos desde los sistemas: Polhemus Liberty [™], Patriot [™], MinuteMan [™] y Liberty LATUS [™], que son sistemas electro magnéticos de rastreo (Tracking).

PDI contiene encabezados de archivos C++, archivos de librerías, el cual es compatible con MS Visual Studio 6, Visual Studio .NET, Visual Studio 2005, y las siguientes versiones.

Microsoft Visual C++ 2008 Express Edition (MSVC)

MSVC cuenta con una versión Express, llamada Microsoft Visual C++ Express Edition, la cual es gratuita y se puede descargar desde el sitio de Microsoft.

OpenSceneGraph-2.8.1

Es una herramienta con código abierto para una alta ejecución de gráficos 3D, esta escrita totalmente en C++ y en OpenGL.

SDL-1.2.13 y SDL_mixer-1.2.8

SDL_mixer es una librería mezcladora de audio multi-canal. Soporta un numero de reproducción simultanea en diversos canales de 16 bit de audio estéreo. Usa librerías SDL.

Blender

Es un kit de código abierto de creación de contenido 3D, disponible bajo licencia GNU (General Public License).

Instalación de OpenSceneGraph-2.8.1, SDL-1.2.13, SDL mixer-1.2.8 y Blender

La instalación de las librerías OpenSceneGraph no es muy sencilla, seguimos los pasos indicados en el sitio web dwightdesign [9]:

El SDL y DSL_Mixer se descomprimen y se añaden las referencias de las carpetas lib y bin en el IDE del Visual C++.

El Blender tiene un asistente de instalación por lo que el proceso es sencillo.

5.2 Instalación y Configuración de equipos.

Instalación del dispositivo de rastreo Polhemus Liberty:

El software y los manuales de instalación y uso vienen con el equipo, en el mismo CD viene el instalador del SDK, que es la librería usada para poder comunicarnos con este dispositivo, los pasos para usar el SDK con el Visual C++ los explican en la ayuda del mismo.

Instalación del Display 5DT (Fifth Dimension Technologies):

Al inicio intentamos usar el visor **eMagin Z800** [10] (<http://www.emagin.com>), que viene con un sensor para el movimiento de la cabeza, pero lamentablemente no logramos ver las imágenes en 3D, al parecer la tarjeta de video (NVIDIA QUADRO FX3700) usada no era compatible con el visor para el 3D, pero el laboratorio cuenta

con otros equipos de display, elegimos el Head Mounted Display **5DT** (Fifth Dimension Technologies), y no el proyector, debido a que deseábamos que sea más inmersiva la aplicación.

DEI 5DT no necesitamos instalar los drivers debido a el computador lo reconoció con los drivers estándares con los que cuenta.

Para poder ver las imágenes en el Head Mounted Display 5DT, hubo que cambiar la configuración de la tarjeta de video para que muestre la salida en dicho display, por ejemplo se activó el modo 3D, se cambió la resolución a 800x600 y la frecuencia a 75Hz, estas configuraciones están indicadas en el manual del equipo, la figura. 5.1 muestra algunos de estas configuraciones mencionadas.



Figura 5.1 Activación de las configuraciones 3D en la Tarjeta de video.

5.3 Proceso de Implementación.

Como ya indicamos anteriormente, como plataforma usamos el lenguaje de programación **Microsoft Visual C++ 2008 Express Edition (MSVC)**, para poder interactuar con el SDK del Polhemus, tuvimos que añadir las siguientes DLL's al sistema operativo: mfc71.dll, mfc71d.dll, msvcp71d.dll, msvcr71d.dll. En la figura 5.2 se muestra el grupo de clases que desarrollamos para la Batería 3D.

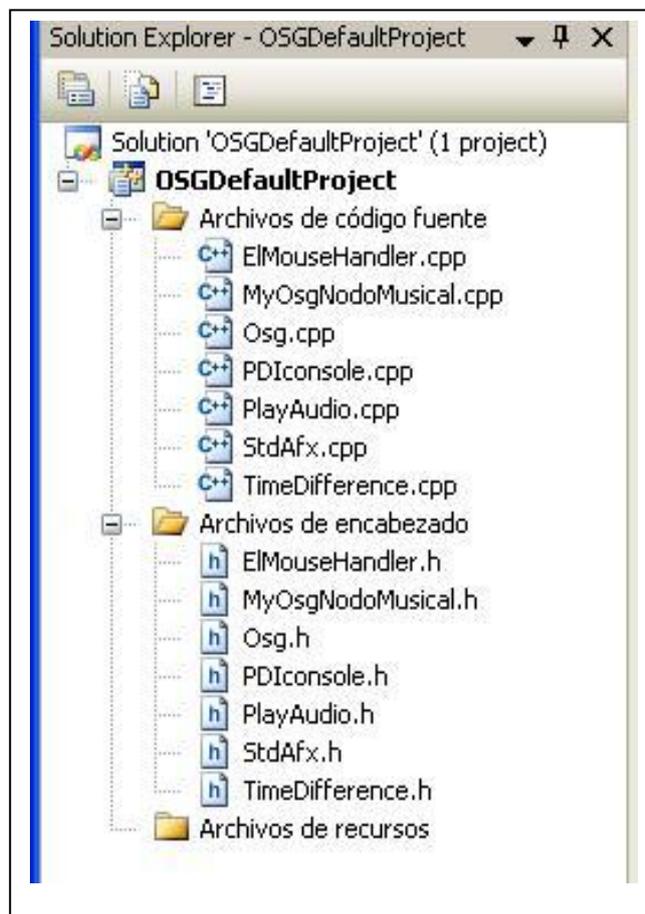


Figura 5.2 Clases desarrolladas.

Carga de escena 3D

Hicimos una clase que representa a cada parte de la batería, la llamamos MyOsgNodoMusical, al crear una instancia de esa clase le asociamos la imagen que va tener, la posición geométrica en la escena, el archivo de sonido que debe reproducir y el canal del audio.

El canal de audio es manejado por la librería SDL, su función es reproducir más de un sonido al mismo tiempo. En la figura 5.3, hemos añadido una porción de código que muestra la carga de los instrumentos en la escena.

```
void elmouseEventHandler::setEscena(void){
    try{
        //z es arriba
        //y adelante atras
        //x izq o der
        baquetaDer = new MyOsgNodoMusical("bataca.obj",osg::Vec3(-
10,20,30),"",-5);
        baquetaIzq = new MyOsgNodoMusical("bataca.obj",osg::Vec3(-
10,20,30),"",-5);

        tomsPiso = new MyOsgNodoMusical( "FLOOR_TOM_V6.obj"
                                           ,osg::Vec3(7.320,4.688,15.395)
                                           ,"floor_tom.wav"
                                           ,0);
        tomsPisoPunch = new MyOsgNodoMusical(
"FLOOR_TOM_POUNCH_V1.obj"
```

Figura 5.3 Código para cargar la escena virtual

Note que las baquetas no tienen sonido asociado, y al canal de audio le establecemos el valor de -5.

La figura 5.4 muestra la escena completa donde están cargadas todas las partes de la batería.



Figura 5.4 Escena virtual de la Batería

Además de las configuraciones en la tarjeta de video, en el código también encendimos la salida en 3D que nos brinda el OpenSceneGraph, esto se puede apreciar en la figura 5.5.

A los parámetros: separación de ojos, alto y ancho de la pantalla, les dejamos las configuraciones por defecto, que son de 6 cm (`osg::DisplaySettings::instance()->setEyeSeparation(0.06)`), 1.50

metros (osg::DisplaySettings::instance()->setScreenHeight(1.50)) y 2 metros (osg::DisplaySettings::instance()->setEyeSeparation(2)) respectivamente.

```
void elmouseEventHandler::set3D(void){
    osg::DisplaySettings::instance()->setStereo(true);
    osg::DisplaySettings::instance()-
>setStereoMode(osg::DisplaySettings::StereoMode::QUAD_BUFFER);
    osg::DisplaySettings::instance()->setScreenDistance(1.50);

    osg::DisplaySettings::instance()->setRGB(true);
}
```

Figura 5.5 Código para establecer la salida en 3D

Lectura de coordenadas de las baquetas

La clase PDIconsole es la que tiene el método main el cual desencadena la ejecución, lo primero que se realiza es Inicializar, Conectar y Configurar el dispositivo de rastreo, luego se configura en el dispositivo el modo continuo de lectura, esto es para obtener las lecturas de las coordenadas X, Y y Z del sensor, una vez que tenemos las lecturas, empezamos a mover los objetos baquetas, en el caso de que haya colisión, se reproduce el sonido asociado a la clase MyOsgNodoMusical.

Cada clase tiene un método destructor, que se ejecuta antes de salir del programa, cerrando las conexiones de la clase PlayAudio, del PDIconsole, y otras.

5.4 Conexión física de los dispositivos de realidad virtual.

En el capítulo 2, sección 2.3 detallamos el panel posterior del equipo Polhemus Liberty, razón por la cual solo vamos a describir la forma en que nosotros establecimos la configuración de los sensores.

El sensor de la cabeza lo insertamos en el primer puerto, en cualquiera de los puertos restantes impares (3, 5, 7) conectamos la baqueta derecha y en cualquiera de los puertos pares (2, 4, 6, 8) conectamos la baqueta izquierda, ver figura 5.6.

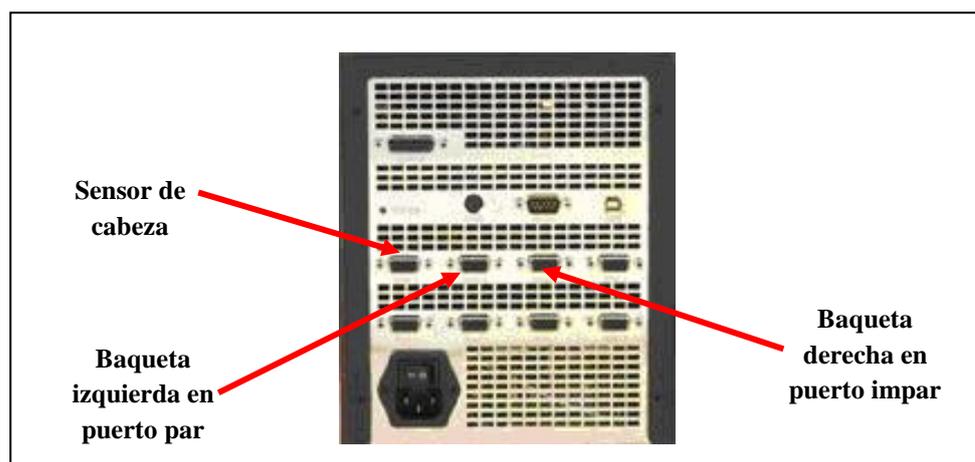


Figura 5.6 Polhemus, configuración de puertos

En la figura 5.7 mostramos un diagrama de conexiones de los dispositivos usados.

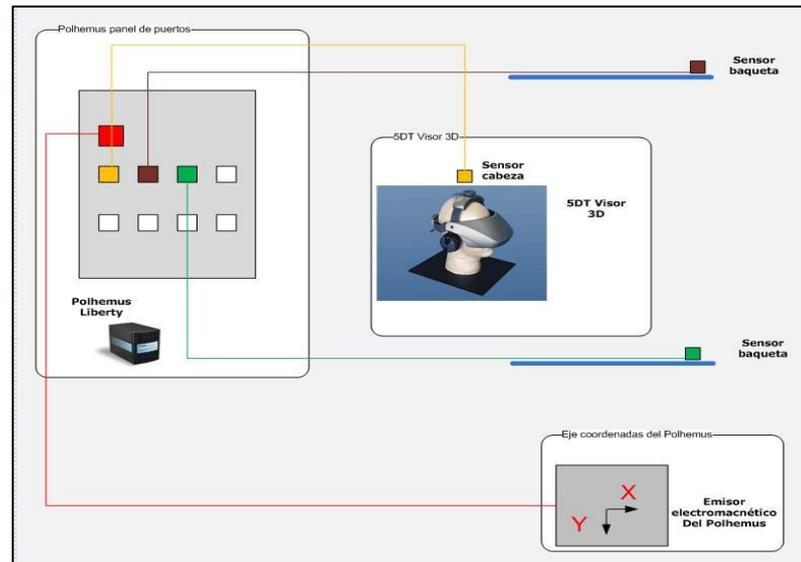


Figura 5.7 Diagrama de conexión de dispositivos

En la figura 5.8 mostramos las baquetas con los sensores adjuntos a las mismas.

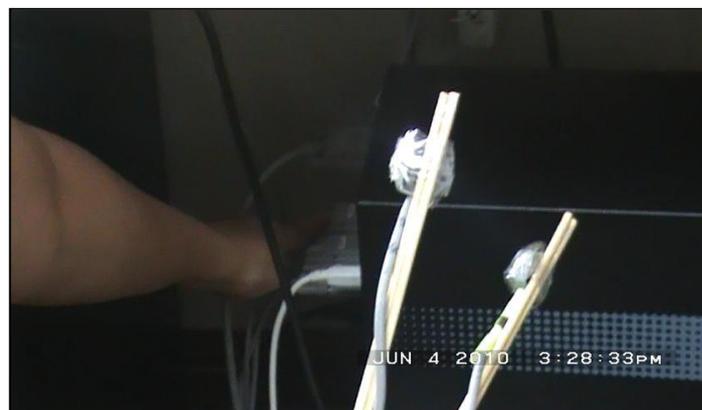


Figura 5.8 Sensores adjuntos a las baquetas

En la figura 5.9 mostramos el sensor adjunto al head mounted display.



Figura 5.9 Sensor adjunto al head mounted display 5DT

En la figura 5.10 mostramos los pedales que se usan para tocar el bombo, la conexión es mediante un cable USB.



Figura 5.10 Pedal USB usado para tocar el bombo

En la figura 5.11 le mostramos una vista de lo que el usuario puede apreciar en el monitor del computador.



Figura 5.11 Ejecución del programa en 2D

En la figura 5.12 vemos a un usuario probando los visores 3D y en la 5.13 usando las baquetas.



Figura 5.12 Toma real de un usuario con el head mounted display



Figura 5.13 Toma real de un usuario con las baquetas

5.5 Pruebas y Resultados.

Para las pruebas se ha elegido los siguientes indicadores:

Latencia.

La capacidad de acoplar los movimientos motrices del usuario con su representación gráfica es la base de muchos sistemas de realidad virtual, la sincronización exacta de los movimientos motrices y la representación gráfica es crítica: retrasos mayores a 100 ms puede causar discrepancias entre lo real y virtual, llegando incluso a causar mareos [11]. Esta es una de las razones por las que hemos incluido a la latencia como un indicador a ser considerado.

Se estableció la latencia como el tiempo transcurrido entre el movimiento motriz de la baqueta por parte del usuario y el instante de colisión con la batería, ver figura 5.14.

Para estas mediciones hicimos solo golpes simples, usamos un temporizador “timer” a nivel de código fuente [15], lo encerramos cada vez que es tomada una lectura de posiciones del Polhemus, tomamos el tiempo hasta que hay un evento de colisión y sumamos el retardo del dispositivo de rastreo Polhemus.

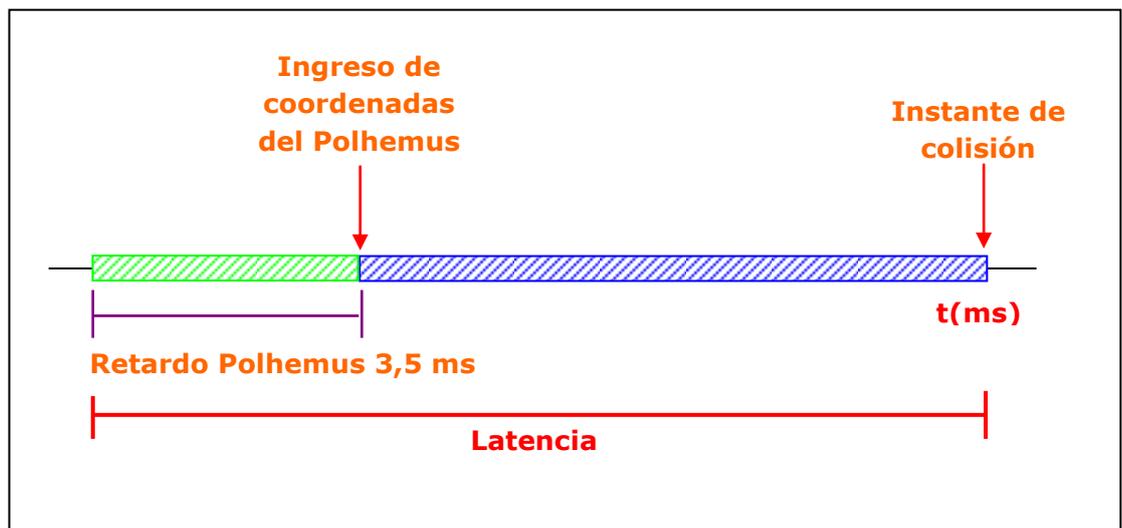


Figura 5.14 Latencia de la batería virtual

Tomamos 30 muestras, basados en el teorema del límite central [17], en el anexo C, en las tablas B.1 y B.2 se encuentran las muestras analizadas.

La tabla 5.1 muestra los resultados obtenidos, el bombo es el elemento con mayor latencia 93 ms, este tiempo es muy cercano al límite del tiempo recomendado para sistemas de realidad virtual (100ms) [11]. La latencia del crash es de 16 ms, del redoblante 17 ms, del esterbil es 18 ms, del bombo es 93 ms, del tom izquierdo es 17 ms, del tom derecho es 16 ms, del tom de piso es 17 ms, del ride es 16 ms. La varianza de cada instrumento es muy pequeña.

	crash	redoblante	esterbil	bombo	tom izquierdo	tom derecho	tom de piso	ride
Sumatoria (s)	0.394316	0.410011	0.442566	2.707233	0.41739	0.398028	0.412817	0.380119
Media (s)	0.013143867	0.013667033	0.0147522	0.0902411	0.013913	0.0132676	0.013760567	0.012670633
Varianza	1.41E-08	3.5E-08	5.29538E-08	1.4E-05	1.68E-07	1.04E-08	2.4E-08	2E-08
Desviación estándar	0.00012	0.0002	0.0002	0.0038	0.0004	0.00010	0.00015	0.00015
Retardo Polhemus (s)	0.0035	0.0035	0.0035	0.0035	0.0035	0.0035	0.0035	0.0035
Resultado (s)	0.016643867	0.017167033	0.0182522	0.0937411	0.017413	0.0167676	0.017260567	0.016170633
Resultado (ms)	16.64386667	17.16703333	18.2522	93.7411	17.413	16.7676	17.26056667	16.17063333

Tabla 5.1 Media, varianza y desviación estándar de las muestras de la latencia.

Nota: A los tiempos promedios le sumamos 3.5 ms (0.0035 s) del retardo del Polhemus [16], con lo que tenemos el tiempo total desde que se ejecuta un movimiento hasta que colisiona.

Inmersión.

La **inmersión** es un factor que describe, desde el punto de vista físico, la experiencia en la realidad virtual [1].

Inmersión.- Este indicador está relacionado a la configuración física de la interfaz de usuario. La cantidad de sentidos que se alimentan establece los niveles de inmersión: Completamente Inmersivo (todos los sentidos), Semi-Inmersivo (la mitad o menos), No Inmersivo (ninguno) [1].

Sentido	Simulación	Elemento
Visión	SI	Gráfico 3D
Audición	SI	Audio
Tacto	NO	--

Tabla 5.2 Los sentidos y la determinación de la inmersión.

En una batería real, por lo regular son tres sentidos los que son experimentados por el baterista, debido a que sólo simulamos dos (la visión y la audición) ver figura 5.2, podemos afirmar que el ambiente virtual es *semi-inmersivo*.

CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES

1. El tiempo de respuesta es una variable que hay que tener muy en cuenta en los sistemas de realidad virtual, ya que juega un papel crucial para dar la sensación de ser más real y no causar discrepancias entre lo real y virtual.
2. La latencia de los instrumentos que conforman la batería varia desde 16 a 18 ms, a excepción del bombo que tuvo un tiempo de 93.74 ms, como resultado tenemos que este instrumento musical reduce la sensación de presencia que experimenta el usuario en el ambiente virtual.
3. No se puede “sentir la batería” debido a que no se implementó una retroalimentación táctil, esto dificulta el manejo de las baquetas para generar golpes en los instrumentos.
4. El dispositivo de despliegue gráfico personal (Head Mounted Display) nos permitió aislarnos del mundo real incrementándose la sensación

de inmersión en el ambiente virtual, este efecto no se hubiese alcanzado con el proyector 3D o con el monitor.

5. La implementación es escalable, dejando abierta la posibilidad de que se añadan más instrumentos de percusión en la escena.

6. En una batería real el sonido generado depende de la forma en que el baterista hace colisionar la baqueta, en el caso de esta implementación existe esta limitante, debido a que los sonidos son grabaciones de una batería real, por lo que el sonido generado va a ser siempre el mismo independientemente de la forma en que se produzca la colisión.

RECOMENDACIONES

1. Debido a que el bombo tiene una alta latencia por el uso del pedal, recomendamos que se implemente su funcionamiento con otro sensor del equipo de rastreo (tracker).
2. Tomando ventaja del diseño escalable de este prototipo, recomendamos que en trabajos futuros se implemente la opción de escoger el tipo de batería que se va a tocar, por ejemplo una de rock, una con tambores africanos, un estilo japonés.
3. Se recomienda que se implemente la retroalimentación táctil para hacer más real la interacción con la batería virtual y de esta forma incrementar la sensación de inmersión en el ambiente virtual.

ANEXOS

ANEXO A

Trabajos similares.

En el sitio Web <http://www.virtual-drums.com/>, encontramos un programa desarrollado a inicios del año 2006, ellos crearon su propio sistema de posicionamiento (tracker) para un robot de la Universidad Laval en Francia y luego usaron ese sistema para la implementación de una batería virtual.

El nombre inicial del proyecto fue “DrummersRevolution”, en la actualidad se llama “Virtual Drums” [12]. Puede ver un video de su ejecución en la siguiente url:

http://www.youtube.com/watch?v=G4QAc1_2JFY



Figura A.1 Batería Virtual implementada en el año 2006

En el sitio: <http://airguitar.tml.hut.fi/>, encontramos una guitarra virtual, fue desarrollada en 2002 en el proyecto que se denominó ALMA del programa Fondos de la Unión Europea (European Union funds) en colaboración con participantes internacionales. Los movimientos del usuario son captados con una cámara Web y pasados a través de un programa de reconocimiento de gestos, luego de esto, ellos fundaron una empresa, cuyo sitio Web es: www.virtualairguitar.com, y se dedican a hacer juegos de video de realidad virtual [13].

Puede ver un video de su ejecución en la dirección:

<http://www.youtube.com/watch?v=FIAmyoEpV5c>



Figura A.2 Guitarra Virtual implementada en el año 2002.

ANEXO B

Datos.

Se tomaron 30 muestras para ser analizadas, las primeras quince están registradas en la tabla B.1 y las restantes en la tabla B.2, estos datos corresponden a los tiempos de retardo producidos desde que se leen las coordenadas del dispositivo de rastreo (tracker) hasta el momento en que se produce una colisión.

n	crash	redoblante	esterbil	bombo	tom izq	tom der	tom de piso	ride
1	0.013206	0.013987	0.014321	0.090144	0.014104	0.013277	0.013934	0.012608
2	0.013088	0.013517	0.014841	0.086124	0.014259	0.013438	0.013852	0.012781
3	0.013262	0.014079	0.014722	0.090413	0.014378	0.013135	0.013628	0.012712
4	0.013099	0.013465	0.014718	0.092635	0.013486	0.013277	0.013579	0.012700
5	0.013206	0.013542	0.014983	0.096344	0.014424	0.013277	0.013820	0.012582
6	0.013153	0.013603	0.014852	0.084157	0.014205	0.013276	0.013983	0.012159
7	0.013202	0.013625	0.014706	0.094392	0.013682	0.013215	0.013874	0.012347
8	0.013252	0.013524	0.014722	0.087827	0.014198	0.013212	0.013780	0.012725
9	0.013004	0.013624	0.014685	0.092993	0.014273	0.013424	0.014112	0.012679
10	0.013031	0.013532	0.014675	0.092189	0.013366	0.013243	0.013918	0.012791
11	0.013094	0.013513	0.014570	0.089908	0.014150	0.013302	0.013961	0.012619
12	0.013154	0.013598	0.014745	0.092360	0.013444	0.013339	0.013684	0.012864
13	0.013115	0.014085	0.014167	0.094260	0.014321	0.013299	0.013643	0.012681
14	0.012923	0.013666	0.014804	0.090707	0.013613	0.013144	0.013492	0.012760
15	0.013036	0.013632	0.014584	0.085323	0.014095	0.013335	0.013487	0.012705

Tabla B.1 Tiempo entre la lectura de datos del Polhemus y el momento de colisión (muestras de la 1 -15).

n	crash	redoblante	esterbil	bombo	tom izq	tom der	tom de piso	ride
16	0.013481	0.013625	0.014869	0.088500	0.014467	0.013442	0.013626	0.012691
17	0.012938	0.013698	0.014622	0.092302	0.013462	0.013145	0.013668	0.012595
18	0.013037	0.013807	0.014375	0.091361	0.013513	0.013149	0.013706	0.012925
19	0.013081	0.013536	0.014971	0.088798	0.014335	0.013184	0.013817	0.012572
20	0.013299	0.013446	0.014910	0.091923	0.013394	0.013296	0.013987	0.012749
21	0.013127	0.013804	0.015130	0.091828	0.013307	0.013380	0.013682	0.012782
22	0.013081	0.013671	0.015176	0.087003	0.013545	0.013514	0.013594	0.012596
23	0.013145	0.014178	0.014944	0.085326	0.014355	0.013160	0.013767	0.012829
24	0.013131	0.013623	0.014420	0.087251	0.013993	0.013233	0.013770	0.012722
25	0.013083	0.013547	0.014927	0.094695	0.014345	0.013327	0.013880	0.012511
26	0.013231	0.013638	0.014938	0.086673	0.013419	0.013239	0.013639	0.012578
27	0.013098	0.013577	0.014580	0.086169	0.013344	0.013203	0.013678	0.012801
28	0.013147	0.013695	0.014912	0.097707	0.014215	0.013218	0.013796	0.012661
29	0.013361	0.013609	0.014814	0.094728	0.013552	0.013092	0.013822	0.012686
30	0.013251	0.013565	0.014883	0.083193	0.014146	0.013253	0.013638	0.012708

Tabla B.2 Tiempo entre la lectura de datos del Polhemus y el momento de colisión (muestras de la 16-30).

BIBLIOGRAFIA

- [1] Mario A. Gutiérrez A, Frédéric Vexo, Daniel Thalmann, “Stepping into Virtual Reality”, 2008.
- [2] William Welling, Photography in America: The formative years, 1987.
- [3] A Three Dimensional Helmet Mounted Primary Flight Reference for Paratroopers, disponible en:
<http://www.stormingmedia.us/35/3525/A352534.html>, último acceso junio 21, 2010.
- [4] Evaluation of Face-Mounted Binocular Video Display for Laparoscopy, disponible en:

<http://www.liebertonline.com/doi/abs/10.1089/109264203769681754>, último acceso junio 21, 2010.

[5] Grigore C Burdea, Philippe Coiffet, Virtual Reality Technology, 2003.

[6] 5DT “Fifth Dimension Technologies” disponible en:

<http://www.5dt.com/hardware.html#hmd>, ultimo acceso junio 21, 2010.

[7] Facultad de Medicina UNAM, Salud en el escolar: Agudeza visual, disponible en:

www.facmed.unam.mx/deptos/salud/periodico/agudeza/ques.html, último acceso junio 21, 2010.

[8] Motion Tracking, Polhemus Liberty disponible en:

http://www.polhemus.com/?page=Motion_Liberty

[9] Installing OpenSceneGraph 2.8.0, disponible en:

<http://dwrightdesign.com/2009/05/installing-openscenegraph-280/>, ultimo acceso junio 21, 2010.

[10] Making virtual imaging a reality, disponible en:

<http://www.emagin.com>, último acceso junio 21, 2010.

- [11] M. Friedmann, T. Starner, A. Pentland, Synchronization in Virtual Realities, Enero 1991.
- [12] Virtual Drums, 2006, disponible en:
<http://www.virtual-drums.com>, último acceso junio 21, 2010.
- [13] Virtual Air Guitar, disponible en:
<http://airguitar.tml.hut.fi/>, último acceso junio 21, 2010.
- [14] Federico Miyara, La música por computadora, 1995.
- [15] How do I evaluate the time difference between two events?, disponible en:
<http://www.codeguru.com/forum/showthread.php?t=379581>,
último acceso junio 21, 2010.
- [16] Liberty Manual.pdf, user manual, Octubre 2005.
- [17] MSc. Gaudencio Zurita, PROBABILIDAD Y ESTADÍSTICA: Fundamentos y Aplicaciones, 2008.
- [18] How Virtual Reality Gear Works, disponible en:
<http://electronics.howstuffworks.com/gadgets/other-gadgets/VR-gear1.htm>, último acceso junio 21, 2010.

- [19] Michael Kriegel, Musical instruments in virtual environments using optical tracking systems, 2005.
- [20] Grigore C. Burdea, Haptic Feedback for Virtual Reality, 1999.