

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL



Facultad de Ingeniería en Electricidad y Computación



**“DECORACIÓN DE INTERIORES DE OFICINAS
USANDO REALIDAD VIRTUAL”**

INFORME DE MATERIA DE GRADUACIÓN

Previa a la obtención del título de:

**INGENIERO EN CIENCIAS COMPUTACIONALES
ESPECIALIZACIÓN SISTEMAS MULTIMEDIA**

Presentado por:

**Luis Jairo Montesdeoca Bermúdez
Juan Arturo Hernández Peter**

Guayaquil – Ecuador

2009-2010

AGRADECIMIENTO

Agradecemos a Dios, a nuestros padres, familiares y amigos que estuvieron dándonos su apoyo incondicional, guiándonos y en muchas ocasiones motivándonos para seguir adelante. Al Ph. D. Sixto García, que nos proporcionó su apoyo durante mucho tiempo y supo corregirnos en donde estábamos equivocados y de esta manera culminar con este proyecto.

DEDICATORIA

A Dios

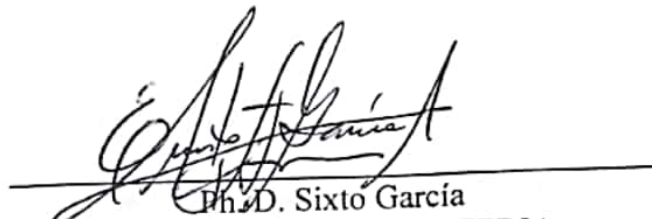
A nuestros padres

A nuestros familiares


A nuestros amigos

A nuestros profesores

TRIBUNAL DE SUSTENTACION



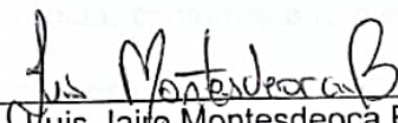
Ph. D. Sixto García
PROFESOR DE LA MATERIA

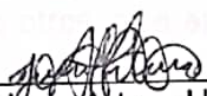


Ph. D. Xavier Ochoa
PROFESOR DELEGADO DEL DECANO

DECLARACIÓN EXPRESA

"La responsabilidad del contenido de este informe de materia de grado, nos corresponden exclusivamente; y el patrimonio intelectual de la misma a la ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL"
(Reglamento de Graduación de la ESPOL)


Luis Jaime Montesdeoca Bermúdez


Juan Arturo Hernández Peter

RESUMEN

El propósito de este proyecto de materia de graduación es mostrar la tecnología de realidad virtual en un sistema de decoración de interiores de oficinas, tomando en cuenta las ventajas de interactuar en tiempo real con el mayor realismo posible con objetos 3D y sentirse inmerso en un ambiente virtual, simulando la decoración en la vida real.

Teniendo en cuenta las necesidades actuales de los clientes que quieren ver el resultado final de decoraciones de oficinas surge la idea de realizar una aplicación que sea capaz de mostrar varios resultados de esa decoración en un ambiente de realidad virtual, permitiendo de esta forma realizar cambios de una manera más rápida, conforme a lo que el cliente desea y que así el diseñador pueda plasmar ese resultado en la vida real.

La decoración de interiores de oficinas es un proceso subjetivo, que se basa en ciertos conceptos como la ubicación de objetos en el entorno, el tipo de objetos, iluminación entre otros, que aplicarlo en la vida real depende mucho del gusto y habilidad del diseñador, pero que no cuenta con una herramienta virtual que le permita realizar cambios con rapidez que puedan ser apreciados por decoradores o clientes antes de hacerlo en la vida real.

Así, tenemos una aplicación que permite decorar el interior de un prototipo de una oficina usando dispositivos de realidad virtual como guantes de datos, trackers, sensores, gafas y un proyector 3D en conjunto con OpenSceneGraph, que es una herramienta para aplicaciones gráficas que maneja todos estos dispositivos. El prototipo de la oficina fue previamente diseñado con dos herramientas de modelado 3D que son: Google Sketchup y 3D Studio Max.

La aplicación, además consta con un menú de objetos que previamente han sido editados en las herramientas de modelado mencionadas y que nos permite realizar varias acciones como añadir, ubicar, borrar o cambiar objetos. Lo que no permite la aplicación es añadir cualquier tipo de oficina, es decir, para poner otro tipo de oficina diferente al prototipo se debería previamente editar el diseño y los elementos que se van a agregar en la misma.

El uso de la aplicación está orientado a aquellos profesionales en el diseño de interiores o aquellos arquitectos que están interesados en esta rama que son capaces de manejar herramientas gráficas de modelado y diseño y que les sería de mucha utilidad a la hora de mostrar su trabajo no solo en forma estática sino de una manera interactiva.

ÍNDICE DE GENERAL

RESUMEN	vi
ÍNDICE DE GENERAL.....	viii
ÍNDICE DE FIGURAS.....	xi
ÍNDICE DE TABLAS	xiii
INTRODUCCIÓN	xiv
1. Descripción del problema.....	1
1.1. Antecedentes.....	1
1.2. Justificación.....	3
1.3. Planteamiento del problema.....	4
1.4. Planteamiento de la solución.....	5
1.5. Alcance.....	6
1.6. Objetivo General.....	6
1.7. Objetivos Específicos.....	7
2. Realidad virtual y decoración de interiores.....	8
2.1. Realidad virtual	8
2.1.1. Definición de realidad virtual.....	9
2.1.2. Dispositivos de realidad virtual.....	11
2.2. Decoración de interiores.....	15

3.	Análisis y diseño de la aplicación.....	17
3.1.	Análisis de requerimientos.....	17
3.1.1.	Casos de uso	17
3.1.2.	Requerimientos funcionales.....	18
3.1.3.	Requerimientos no funcionales.....	20
3.2.	Arquitectura de la aplicación.....	21
3.3.	Análisis del diseño.....	22
3.4.	Diseño de la oficina mediante 3D Studio Max y Google Sketchup. .	22
3.5.	Diseño de clases.....	24
3.6.	Diseño de interacción de objetos.....	26
3.6.1.	Escenario 1: Conexión exitosa con dispositivos.....	26
3.6.2.	Escenario 2: Selección exitosa de objetos.....	27
3.6.3.	Escenario 3: Movimiento de cámara en escena exitosa.....	27
3.6.4.	Escenario 4: Movimiento exitoso de objetos.....	28
3.6.5.	Escenario 5: Escoger opciones del menú exitosamente.....	29
3.7.	Diseño de almacenamiento de objetos 3D.....	29
4.	Implementación de la solución.....	31
4.1.	Herramientas necesarias.....	31
4.1.1.	OpenSceneGraph	33
4.1.2.	Grafos de escena.....	34
4.1.3.	Arquitectura de OSG.....	36
4.2.	Implementación sin uso de dispositivos de realidad virtual.....	37
4.2.1.	Manejo de la cámara.....	37
4.2.2.	Selección de objetos.....	43

4.2.3.	Menú de objetos.....	44
4.2.4.	Movimiento y rotación de objetos.....	44
4.3.	Implementación usando dispositivos de realidad virtual.....	45
4.3.1.	Guantes de datos, trackers y sensores.....	45
4.3.2.	Manejo de la cámara.....	46
4.3.3.	Selección, menú, movimiento y rotación de objetos.....	47
4.3.4.	Manejo del cursor.....	48
4.4.	Prototipo de la implementación.....	49
5.	Plan de pruebas.....	54
5.1.	Pruebas, configuraciones y resultados.....	54
5.2.	Gestos predefinidos.....	55
5.3.	Pruebas con el guante de datos en la aplicación.....	56
5.4.	Configuraciones de gestos y resultados.....	57
5.5.	Configuraciones de tracker y sensores.....	58
CONCLUSIONES Y RECOMENDACIONES		61
BIBLIOGRAFIA		64

ÍNDICE DE FIGURAS

FIGURA 2. 1 GUANTE DE DATOS 5DT	11
FIGURA 2. 2 HMD I-GLASSES	12
FIGURA 2. 3 GAFAS 3D ACTIVAS	13
FIGURA 2. 4 POLHEMUS LIBERTY	14
FIGURA 2. 5 PROYECTOR DEPTH Q	14
FIGURA 2. 6 DECORACIÓN DE OFICINAS	15
FIGURA 3. 1 DIAGRAMA DE CASOS DE USO.....	18
FIGURA 3. 2 ARQUITECTURA DE LA APLICACIÓN.....	21
FIGURA 3. 3 GOOGLE SKETCHUP	23
FIGURA 3. 4 3D STUDIO MAX	24
FIGURA 3. 5 DIAGRAMA DE CLASES	25
FIGURA 3. 6 DIO: CONEXIÓN.....	26
FIGURA 3. 7 DIO: SELECCIÓN DE OBJETOS	27
FIGURA 3. 8 DIO: MOVIMIENTO CÁMARA	28
FIGURA 3. 9 DIO: MOVIMIENTO OBJETOS.....	28
FIGURA 3. 10 DIO: OPCIONES MENÚ	29
FIGURA 3. 11 ALMACÉN DE DATOS	30
FIGURA 4. 1 LA APLICACIÓN 3D EN CAPAS	34

FIGURA 4. 2 GRAFO DE ESCENA.....	35
FIGURA 4. 3 ARQUITECTURA DE OSG	37
FIGURA 4. 4 ESCENA PRINCIPAL	49
FIGURA 4. 5 MENÚ DE OBJETOS	50
FIGURA 4. 6 MENÚ DE SILLAS.....	51
FIGURA 4. 7 MENÚ DE ESCRITORIOS.....	51
FIGURA 4. 8 MENÚ DE BIOMBOS.....	52
FIGURA 4. 9 MENÚ DE ANAQUELES	52
FIGURA 4. 10 MENÚ DE OBJETO SELECCIONADO	53
FIGURA 5. 1 GESTOS PREDEFINIDOS.....	55

ÍNDICE DE TABLAS

TABLA 5. 1 PRUEBA DE GESTOS	57
TABLA 5. 2 CONFIGURACIÓN DE GESTOS	58
TABLA 5. 3 UMBRALES DE MOVIMIENTO	59

INTRODUCCIÓN

La decoración de interiores de oficina es la forma de adornar o distribuir, los espacios internos de un lugar de trabajo. La decoración de interiores, se logra por medio, de la colocación de distintos objetos, los cuales trabajan de forma armónica, para crear una sensación agradable a la vista. Pero esta sensación agradable es arbitraria, o sea, completamente subjetiva. Cada persona, en la decoración de interiores, logra su ambiente deseado. Por este motivo se dice que la decoración de interiores, es un proceso personal [1]. Por ello utilizando los equipos de realidad virtual se quiere dar un mayor realismo a lo que se quiere mostrar para que las personas se sientan inmersas y tengan una mejor interacción en ella.

La tecnología de realidad virtual nos permite de esta manera desarrollar una aplicación inmersiva, interactiva y navegable dentro de un entorno que incluye objetos 3D que pueden ser apreciados con un mayor detalle, en su forma, color o textura; apreciar los cambios que se realicen en el ambiente virtual y moverse dentro del mismo.

CAPÍTULO I

1. Descripción del problema.

El presente capítulo menciona el por qué de desarrollar una aplicación de este tipo, que sistemas se han desarrollado actualmente, cuál es el problema que se pretende solucionar y las soluciones a estos problemas existentes en la decoración de interiores de oficinas. Luego definimos el alcance y objetivos de la aplicación.

1.1. Antecedentes.

Actualmente existen personas con un alto poder adquisitivo, quienes confían la decoración de interiores, tanto de su casa, oficina u otro lugar, en las manos de expertos en este tema. Estos expertos son los que deciden, en base a sus conocimientos y preferencias de sus clientes, qué objetos convienen más, para el espacio que se desea

decorar. Dentro de la decoración de interiores, cabe todo objeto utilizable, para embellecer recinto interno. Estamos hablando de los muebles, lámparas, tapices, papeles murales, pintura, plantas, pisos de parquet, etc. En el fondo, todo aquello que se considera mueble, o sea, objeto inanimado que puede ser trasladado, por medio del hombre u otro medio [2].

Hoy en día, el Feng Shui es una de las corrientes orientales más utilizadas para la decoración de interiores, debido a su idea central de dejar circular libremente las corrientes positivas de energía en la casa y eliminar aquellas negativas. Esto se realiza por medio de una disposición especial de los objetos muebles [3].

En la decoración de oficinas, los colores deben transmitir la imagen de la empresa y deben ser acogedores para la vista; y además debe crear un buen entorno visual no solo para los clientes, sino también para los empleados. Hay que considerar que aquellos que trabajan dentro de la empresa pasan la mayor parte de su día en ella.

1.2. Justificación.

El lugar de trabajo también ocupa una parte importante de nuestra vida, y para desarrollar con éxito nuestra profesión, el ambiente que nos rodea juega un papel fundamental. No es lo mismo pasar ocho horas en un despacho pequeño, con poca luz, desordenado y sin una sola planta, que ir todos los días a un puesto de trabajo donde existe una buena iluminación, los muebles son ergonómicos y la amplitud es un hecho.

Las oficinas actuales se preocupan por el bienestar del empleado. Desde las compañías, cada vez se presta mayor atención a aspectos de carácter estético, ya que está demostrado que el ambiente influye directamente sobre la productividad. Por ello las empresas necesitan de una mejor herramienta para que puedan decidir y tener una apreciación más real de cómo quedará la oficina para poder satisfacer las necesidades previamente mencionadas.

Considerando los costos que conlleva realizar una tarea como la decoración de interiores de oficina podemos mencionar una estimación cualitativa en base a lo que costaría realizar una decoración de interiores sin un sistema de realidad virtual frente a lo que costaría utilizar un sistema de realidad virtual. Obviamente invertir

en un sistema de realidad virtual para decoración de interiores es significativamente más costoso que hacer decoración sin tener un sistema, pero tiene algunas ventajas como: el tiempo que le tomaría reflejar su trabajo que realizaría en el sistema en la vida real, la edición continua de objetos en tiempo real y poder mostrar muchos resultados a los clientes que les permita escoger la decoración de acuerdo a su preferencia.

1.3. Planteamiento del problema.

El sector de la construcción necesita la opción de promocionar sus servicios y/o construcciones. Para ello requiere que sus clientes tengan la posibilidad de ver y decorar su oficina con el máximo realismo y detalle, como si ellos estuvieran dentro de ella.

Es habitual que nos resulte difícil decidirnos por los elementos decorativos que incluiremos dentro de ella, o que dudemos a la hora de decantarnos por un color, para cambiar los tonos en los que está decorado, o que no nos atrevamos a renovar el ambiente, cambiando las tapicerías, etc., y es que muchas veces nos da miedo que los cambios no resulten bien, que no queden como nosotros habíamos imaginado.

1.4. Planteamiento de la solución.

La solución que proponemos es el desarrollo de un sistema que utiliza la realidad virtual y con la que podremos ganar mayor realismo al momento de que los usuarios interactúen con el sistema. El sistema es una inversión muy importante puesto que hay muchos casos en que los clientes no han apreciado sus construcciones como se lo imaginaban, por ello con nuestro sistema tendrán mayor noción que disminuya el riesgo de un cambio futuro.

El sistema también ayuda a poder entender la solicitud del cliente en cuanto a lo que él en realidad necesita puesto que se sentirá como si estuviera en ese mismo instante dentro de la decoración.

El sistema desarrollado va a permitir al usuario poder cambiar las texturas tanto en las paredes de la oficina así como cambiar los objetos que se encuentren en ella.

También podrá escoger los objetos que quiere que estén dentro, pero dichos objetos será dentro de una lista que el sistema les propone, no los que el usuario sugiera en ese momento.

Se va a permitir que por medio de los guantes utilizados en realidad virtual, hacer que el usuario pueda mover los objetos en ubicaciones que ellos elijan, para así poder tener mayor conocimiento de lo que desea.

1.5. Alcance.

Decorar una oficina en un ambiente virtual, recorriendo el ambiente usando los equipos respectivos para realidad virtual como guantes, trackers, sensores, gafas y proyector 3D para cambiar texturas, escoger objetos tales como mesas, sillas, biombos, entre otros que están dentro de una oficina, así como mover y rotar objetos. La aplicación no permitirá cambiar la arquitectura de la oficina ni añadir oficinas sin ser previamente editadas mediante una herramienta de modelado 3D, tan solo permitirá decorar el interior de un prototipo de una oficina previamente editada.

1.6. Objetivo General.

Permitir que el usuario interactúe en un ambiente de realidad virtual, decorando la oficina de manera tal que se sienta inmerso en este

ambiente virtual y así proyectar lo que se quisiera mostrar en el mundo real.

1.7. Objetivos Específicos.

Navegar en un ambiente virtual 3D que le permita al usuario el recorrido en todas las direcciones posibles con respecto a todos los ejes: X, Y, Z; así como la rotación con respecto a los ejes X y Z.

Permitir al usuario escoger de un menú opciones para decorar la oficina como agregar objetos de diferentes tipos, seleccionar objetos para moverlos, rotarlos o cambiarlos, así como eliminarlos.

Interactuar con la aplicación usando los dispositivos de realidad virtual como los guantes, el tracker, las gafas entre otras para obtener el resultado de inmersión en un ambiente virtual.

CAPÍTULO II

2. Realidad virtual y decoración de interiores

El presente capítulo hace una introducción a lo que es la realidad virtual así como los diferentes dispositivos que son utilizados en una aplicación. Por último se describe brevemente sobre el concepto de decoración de interiores.

2.1. Realidad virtual

En la actualidad vivimos en un mundo muy avanzado en cuanto a tecnología, equipos y software de realidad virtual que permiten la representación del mundo real en un mundo virtual con características similares. Gracias al modelado y a la animación por computadora podemos representar objetos del mundo real con

excelentes resultados, también podemos aplicar leyes físicas: como mover, rotar, colisionar, etc., en fin podemos realizar un sinnúmero de transformaciones de manera similar como lo hacemos en el mundo real [4].

Existen muchas aplicaciones de realidad virtual que son de mucha utilidad como simuladores utilizados en el campo de la medicina e ingeniería, aplicaciones en el campo de la educación con software virtuales para la enseñanza de niños, en marketing y comercio electrónico con visitas virtuales para observar productos, así como en arquitectura para la creación de modelos virtuales y la navegación dentro de ellos.

2.1.1. Definición de realidad virtual.

Existen diferentes definiciones de realidad virtual, debido a que es una tecnología en constante evolución; sin embargo, se considera a criterio propio la más aproximada la siguiente definición:

La realidad virtual [5] es simulación por computadora, dinámica y tridimensional, con alto contenido gráfico, acústico y táctil, orientada a la visualización de situaciones y variables complejas,

durante la cual el usuario ingresa, a través del uso de sofisticados dispositivos de entrada, a "mundos" que aparentan ser reales, resultando inmerso en ambientes altamente participativos, de origen artificial.

La realidad virtual puede ser de dos tipos: inmersiva y no inmersiva. La realidad virtual inmersiva se refiere a aquellos métodos relacionados con un ambiente tridimensional creado por un computador, utilizando dispositivos como cascos, guantes de datos, sensores de movimiento, etc., que nos permiten realizar movimientos y transformaciones en un mundo virtual. La realidad virtual no inmersiva en cambio se crea en un ambiente virtual generado por el computador, al igual que la realidad virtual inmersiva, sin embargo no necesita de dispositivos de adicionales como los mencionados en los que utiliza la realidad virtual inmersiva [6].

La ventaja de la una sobre la otra es en la relación costo-interacción/inmersión ya que la realidad virtual inmersiva al utilizar dispositivos genera más costos pero de igual manera obtenemos una mayor inmersión e interacción con el ambiente virtual, lo que

obviamente no ocurre con el método de realidad virtual no inmersiva.

2.1.2. Dispositivos de realidad virtual.

Existe una gran variedad de dispositivos electrónicos y mecánicos que nos permiten la interacción con el mundo virtual, de los cuales mencionaremos aquellos que consideramos más importantes para el desarrollo de aplicaciones de realidad virtual.

2.1.2.1. Guantes de datos

El guante de datos [7] se utiliza para manipular múltiples objetos en ambientes virtuales que incluyen evaluación de prototipos digitales, biomecánica y animación.



Figura 2. 1 Guante de datos 5DT

2.1.2.2. HMD o Visor 3D

El visor o “Head Mounted Display” [8] por sus siglas en inglés, abarca el área de visión del usuario. Proporciona el efecto de inmersión por medio de un visor, un sistema de audio y un sensor de orientación. Los cascos proporcionan una experiencia inmersiva en ambientes virtuales, transmitiendo impacto y realismo al usuario.



Figura 2. 2 HMD I-Glasses

2.1.2.3. Gafas estereoscópicas

Las gafas estereoscópicas [9] requieren un sensor infrarrojo, que permita sincronizar las imágenes alternativas de la pantalla con las lentes LCD de las gafas. Las gafas activas llevan una batería, un sensor infrarrojo, cristales LCD y

circuitería, por lo que son algo pesadas y caras. El proyector emite las imágenes sin ningún tipo de filtrado, siendo las gafas a través del sensor las que se obturan y desobturan a la misma frecuencia que la emisión de fotogramas.



Figura 2. 3 Gafas 3D activas

2.1.2.4. Sistema de Sensores de movimiento

El sistema de sensores consta de un tracker [10] electromagnético y soporta cierta cantidad de sensores conectados. Es un dispositivo de seguimiento de alta velocidad con cierto grado de libertad (por lo general 6) y trabaja a cierta frecuencia independientemente del número de sensores conectados (de 120 Hz a 240 Hz), con una latencia cercana a cero. Es ideal para aplicaciones del movimiento de disciplinas deportivas y simuladores de entrenamiento.



Figura 2. 4 Polhemus Liberty

2.1.2.5. Proyector 3D

Es un proyector estéreo de alta definición 3D [11] que proyecta imágenes en un rango de 50 Hz a 120 Hz. Incluye un emisor estereográfico y es ideal para presentaciones en 3D.



Figura 2. 5 Proyector Depth Q

2.2. Decoración de interiores.

La decoración de interiores [12], es la forma de adornar o distribuir, los espacios internos de un hogar, departamento u oficina. La decoración de interiores, se logra por medio, de la colocación de distintos objetos, los cuales trabajan de forma armónica, para crear una sensación agradable a la vista. Pero esta sensación agradable, es arbitraria. O sea, completamente subjetiva. Cada persona, en la decoración de interiores, logra su ambiente deseado. Por lo mismo, es que se dice, que la decoración de interiores, es un proceso personal.



Figura 2. 6 Decoración de oficinas

La decoración de oficinas [13] no es un tema menor. Sobre todo en aquellas oficinas en donde se interactúa mucho con clientes y

visitantes. La decoración de oficinas transmite toda una imagen corporativa u organizacional.

En la actualidad se privilegia aquella decoración de oficinas amplia y moderna. No solo para los clientes sino también para los empleados. Ya que hay que pensar que aquellos que trabajan dentro de la empresa pasan la mayor parte de su día en ella. Por lo que el mobiliario usado para la decoración de oficinas debe ser agradable a la vista y de la misma manera práctica de ocupar.

CAPÍTULO III

3. Análisis y diseño de la aplicación.

El presente capítulo describe el análisis y el diseño de la aplicación, detallando la ingeniería de requerimientos, casos de uso, el diseño de clases e interacción de objetos, el modelado de objetos en 3D, así como la manera como se almacenaran los objetos 3D en la aplicación.

3.1. Análisis de requerimientos

3.1.1. Casos de uso

Los casos de uso definen las acciones que existen entre el usuario o cliente, conocido como el actor principal y el sistema; de

esta manera mostramos su comportamiento, como en el siguiente diagrama de casos de uso:

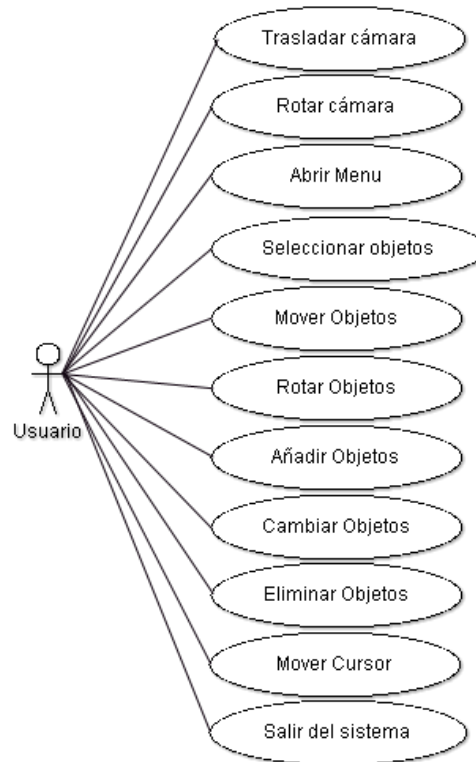


Figura 3. 1 Diagrama de Casos de Uso

3.1.2. Requerimientos funcionales.

El sistema debe ser capaz de realizar todas las acciones descritas anteriormente en los casos de uso, pero para ello debe interactuar con los dispositivos de realidad virtual tomando sus acciones para obtener una salida deseada. Consideramos las

entradas de nuestro sistema a los dos sensores del tracker, considerados par e impar de acuerdo a como están conectados y los gestos de los guantes derecho e izquierdo de acuerdo a la configuración de gestos definida en el capítulo V.

Para podernos trasladar o rotar en la escena se debe: leer los valores del sensor impar manteniendo un gesto del guante derecho; los valores correspondientes a esta acción se detallan en el capítulo V. La traslación se realiza en dirección de los 3 ejes X, Y, Z y la rotación con respecto a X y Z.

El movimiento del cursor se obtiene a partir de la lectura de los valores del sensor impar y manteniendo un gesto del guante derecho; el movimiento es en dos dimensiones, en este caso en el eje X y en el eje Z.

La acción de selección la obtenemos con un gesto del guante derecho que en esta aplicación la llamamos “pick”, que es útil para el menú y para mover o rotar objetos. Esta misma acción permite quitar la selección de un objeto.

El menú se obtiene a partir de un gesto del guante izquierdo y muestra las opciones en la parte superior izquierda de la aplicación. A partir del menú y con la acción de selección podemos escoger las acciones de añadir, cambiar y eliminar objetos; así como la acción de salir del sistema.

Para trasladar y rotar objetos obtenemos la lectura del sensor par con el gesto del guante izquierdo. La traslación es en el eje Y y en el eje X. La rotación es sobre el eje Z.

3.1.3. Requerimientos no funcionales.

Entre los requerimientos no funcionales podemos mencionar la facilidad de uso considerando que tenemos dos guantes y 5 gestos que memorizar entre los dos guantes.

El costo es un requisito muy significativo en la aplicación debido a la adquisición de los equipos de realidad virtual.

La aplicación es extensible aunque no escalable, en vista de que se sacrifica el rendimiento al agregarle más funcionabilidad a la misma.

3.2. Arquitectura de la aplicación.

La aplicación consta de 4 partes fundamentales: La interfaz gráfica, la lógica de la aplicación, el manejo de archivos y el almacén de modelos 3D, como se muestra en el siguiente gráfico:

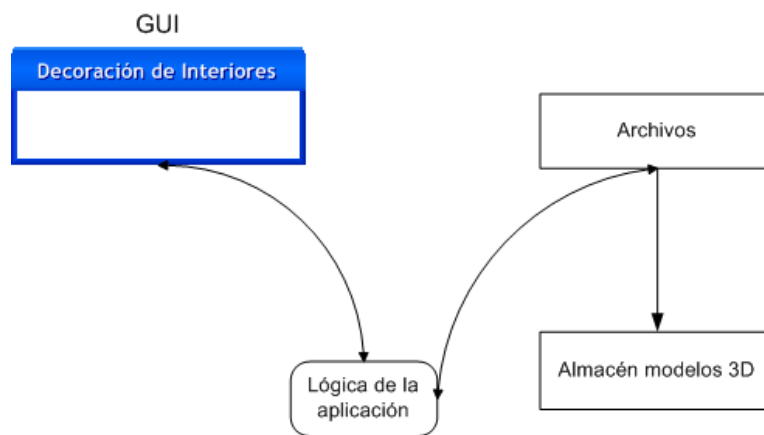


Figura 3. 2 Arquitectura de la aplicación

La aplicación lee desde archivos que están debidamente clasificados según el tipo de objeto que se vaya a obtener del almacén de objetos 3D y esta ruta es retornada para que pueda ser cargada en el GUI; y así de esta forma los objetos que se quieran añadir o cambiar pasan a través de la lógica de la

aplicación para ser leídos desde archivo que envía la ruta del almacén de datos.

3.3. Análisis del diseño.

El diseño de la aplicación consta de 3 partes: El modelado de la oficina utilizando las herramientas disponibles, que posterior a esto se exporta a OSG, el almacenamiento e importación de este modelo en partes a OSG y el diseño de clases. Cada parte de la oficina debe ser modelada por separado y exportada de la misma manera. De igual manera al importar estas partes en OSG, se lo debe hacer con el nombre correspondiente al que se lo exportó. Esto con la finalidad de que cada parte sea tratada por separado y poder realizar las transformaciones correspondientes.

El diseño de clases y objetos nos permitirá aplicar orientación a objetos y de esta manera tener un código mas entendible, y de fácil mantenimiento.

3.4. Diseño de la oficina mediante 3D Studio Max y Google Sketchup.

3D Studio Max y Google Sketchup son las herramientas que nos facilitan el modelado, transformación y exportación de objetos 3D.

Cada modelo obtenido en Google Sketchup es exportado para ser transformado y adecuado en la escena que se está modelando en 3D Studio Max.



Figura 3. 3 Google Sketchup

Como se muestra en la figura 3.3 cada modelo es obtenido en Google Sketchup, se lo selecciona y se lo exporta; aquí se pueden obtener un sinnúmero de modelos de oficina y de cualquier tipo.

Posterior a esto cada modelo tomado de Google Sketchup es importado desde 3D Studio Max para su posterior transformación, como se muestra en la figura 3.4.

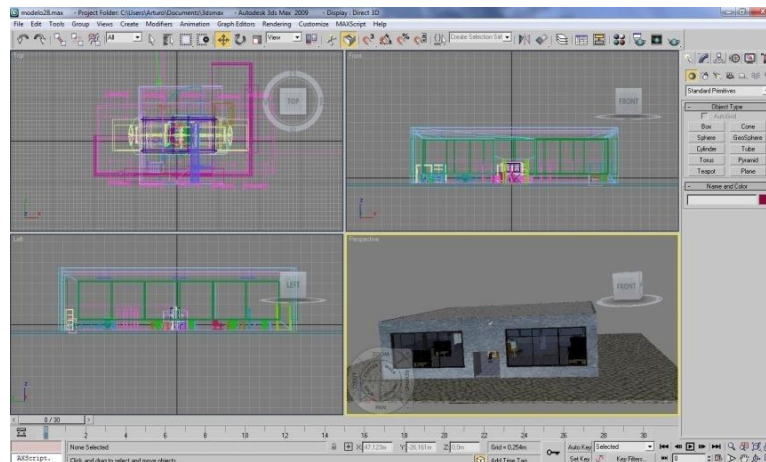


Figura 3. 4 3D Studio Max

3.5. Diseño de clases.

El diseño de clases se basa en la orientación a objetos aplicados a c++, ya que OSG está basado en este lenguaje, de ahí que hemos estructurado la aplicación en diferentes clases dependiendo de las diferentes interacciones en la aplicación.

De esta manera se estructura la solución con 8 clases: la clase *Scene* que es donde se encuentra el método main que es el que ejecuta la aplicación; la clase *PickHandler* es la encargada de realizar todos los métodos relacionados con la selección y transformación de objetos; la clase *CameraManipulator* es la encargada de generar los métodos para mover y rotar la cámara en la escena; la clase *Menu* es la encargada de mostrar las diferentes opciones que tendrá la

aplicación; la clase *Glove* es la que maneja la interacción de gestos con el guante; la clase *Tracker* es la encargada de generar las distintas posiciones con los sensores para la cámara, movimiento de objetos y para el cursor; la clase *Display* es la encargada de la configuración estéreo, y las diferentes distancias para el monitor o proyector; y por último la clase *Cursor* es la que nos permite obtener las distintas posiciones del cursor en la aplicación.

La figura 3.5 muestra el diagrama de clases y los distintos atributos y métodos de cada clase.

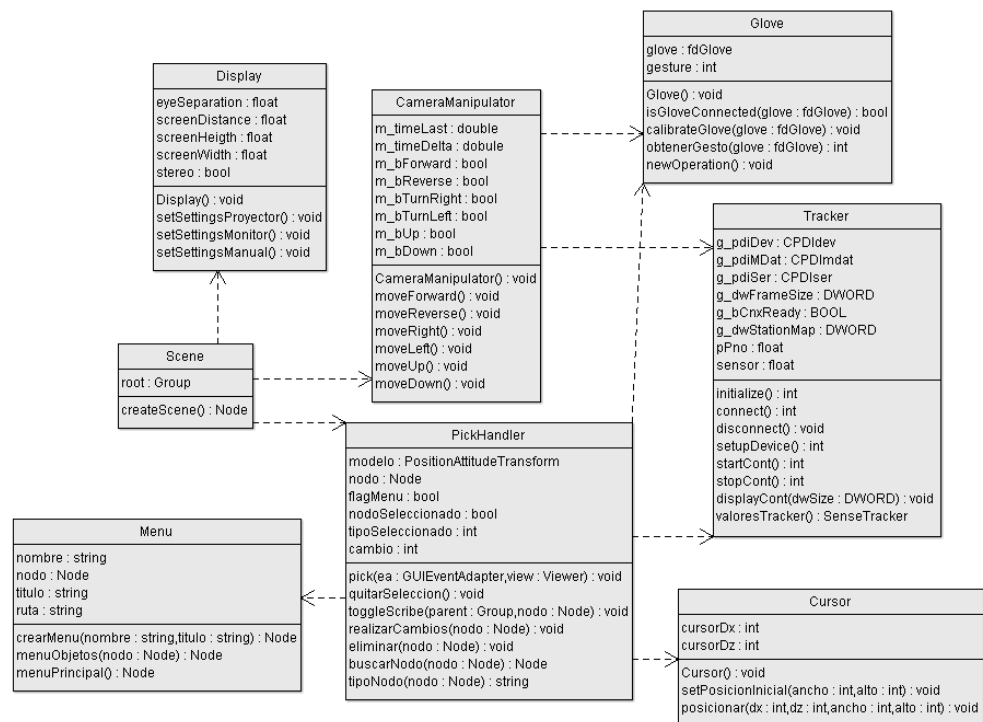


Figura 3. 5 Diagrama de clases

3.6. Diseño de interacción de objetos

En la interacción de objetos de la aplicación hemos considerado cinco escenarios importantes que engloban todas las acciones, considerando la conexión con los dispositivos de realidad virtual.

3.6.1. Escenario 1: Conexión exitosa con dispositivos.

Se considera aquí el momento en que la aplicación inicia y debe establecer una conexión con el dispositivo del guante de datos y con el dispositivo del tracker, de tal manera que nos muestre un resultado de conexión satisfactoria. El siguiente diagrama muestra lo descrito en la figura:

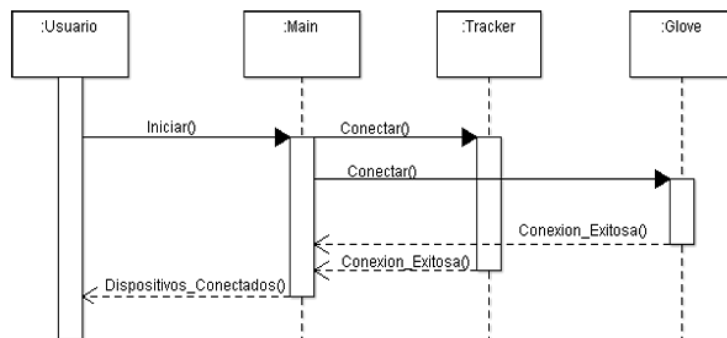


Figura 3. 6 DIO: Conexión

3.6.2. Escenario 2: Selección exitosa de objetos.

Aquí se muestra como los objetos que deben ser seleccionados deben ser apreciados por el usuario al momento de ejecutar la acción de selección. El siguiente diagrama muestra lo descrito en la figura:

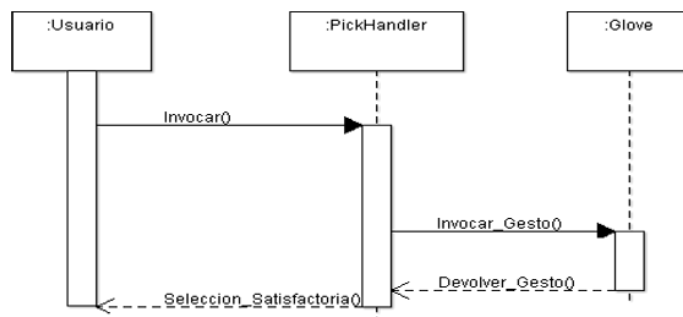


Figura 3. 7 DIO: Selección de Objetos

3.6.3. Escenario 3: Movimiento de cámara en escena exitosa.

Aquí consideramos el escenario en el cual se deben realizar los diferentes movimientos de traslación y rotación de la cámara en la escena de una manera tal que el usuario aprecie correctamente cada uno de estos movimientos. El siguiente diagrama muestra lo descrito en la figura:

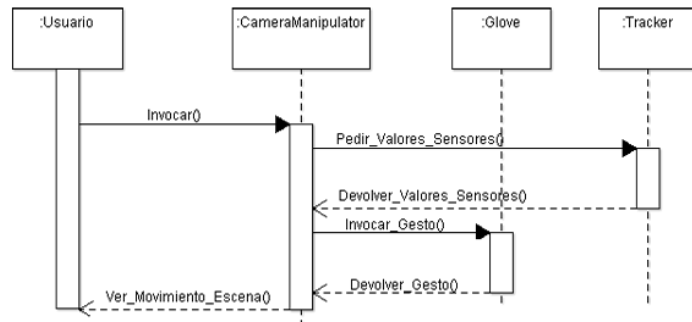


Figura 3. 8 DIO: Movimiento Cámara

3.6.4. Escenario 4: Movimiento exitoso de objetos.

Este escenario muestra las acciones que se deben realizar para los diferentes movimientos de traslación y rotación de los objetos presentes en la escena de una manera tal que el usuario aprecie correctamente cada uno de estos movimientos. El siguiente diagrama muestra lo descrito en la figura:

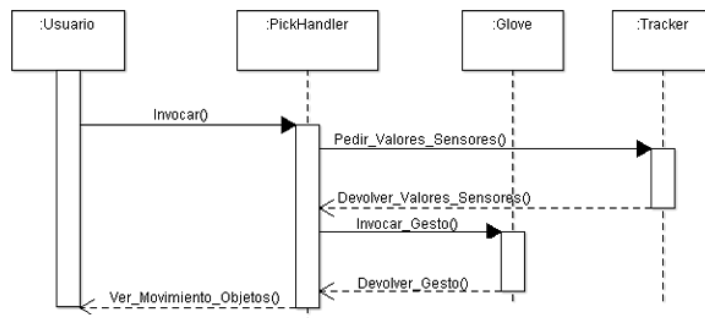


Figura 3. 9 DIO: Movimiento Objetos

3.6.5. Escenario 5: Escoger opciones del menú exitosamente.

Este escenario muestra la interacción del usuario al momento de escoger las opciones del menú, de tal manera que al realizar una acción el pueda apreciar la opción correspondiente. El siguiente diagrama muestra lo descrito en la figura:

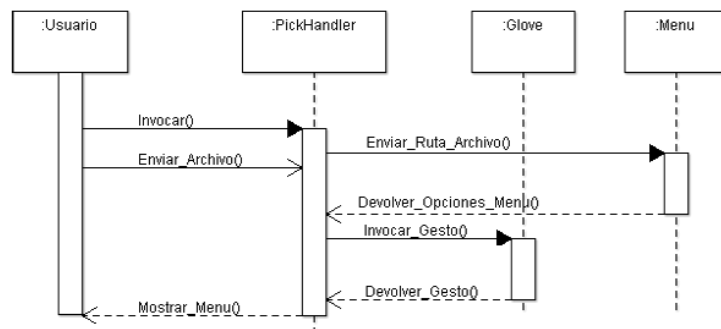


Figura 3. 10 DIO: Opciones Menú

3.7. Diseño de almacenamiento de objetos 3D.

Para almacenar los objetos 3D que se mencionan en el análisis del diseño es necesario establecer un almacén de datos que podría ser una base de datos en particular, pero para facilitar el acceso y disminuir la complejidad los objetos 3D se guardaran en carpetas con una estructura de árbol y serán accedidos mediante archivos, los cuales están distribuidos de acuerdo al tipo de objeto que se exportó

previamente. La figura 3.11 muestra la distribución del almacenamiento de objetos 3D.

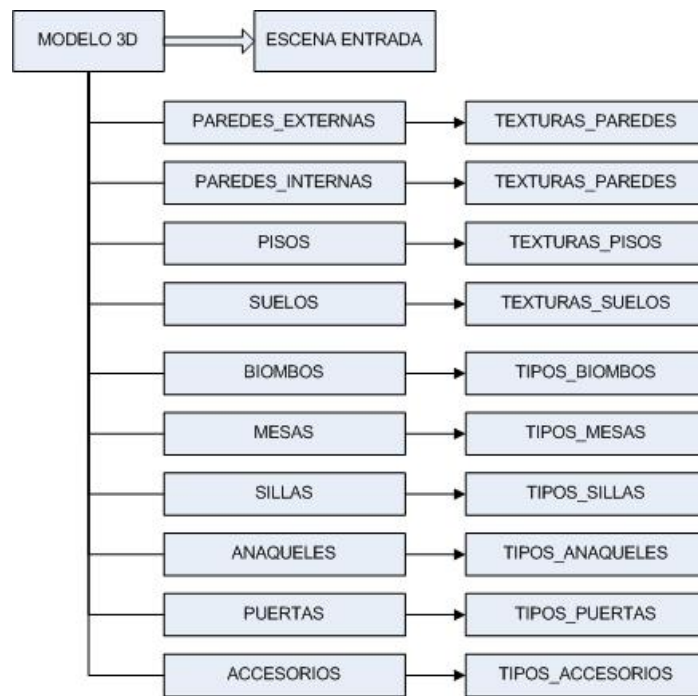


Figura 3. 11 Almacén de datos

CAPÍTULO IV

4. Implementación de la solución.

El presente capítulo describe los detalles de la implementación de la solución, las herramientas disponibles a utilizar para la misma y hace una diferencia en cómo se implementó la solución sin los dispositivos de realidad virtual y con el uso de los dispositivos, lo primero debido a que no se contaba con los dispositivos de realidad virtual, con esto se realizaron pruebas con el teclado y luego toda la interacción se mapea con los dispositivos mencionados. Por último muestra un prototipo de la solución.

4.1. Herramientas necesarias.

Para la implementación de la solución es necesario mencionar las herramientas que se van a utilizar, las que están disponibles y las que se podrían utilizar en caso de no tener las mismas versiones.

Para el desarrollo del modelado de la oficina son necesarios herramientas que nos permitan manipular y transformar objetos en 3D, para ello utilizamos la herramientas ya mencionadas en el capítulo anterior, como son 3D Studio Max 2009 y Google Sketchup Pro 7. Cabe recalcar que no son las únicas herramientas que nos permiten modelar objetos 3D, por mencionar unas cuantas herramientas 3D [14] que nos permiten manipular gráficos por computadora como: Maya [15], Blender [16], LightWave [17], Cinema 4D [18], Houdini [19], etc., pero nos inclinamos por estas por ser de mayor facilidad de uso. Además para retocar ciertas imágenes hemos utilizado Adobe Photoshop CS3, aunque se podría haber utilizado cualquier versión anterior o actual.

Para el desarrollo en lo que se refiere a programación utilizamos Visual Studio 2008, instalando y configurando las variables de entorno y de sistema para su efecto en conjunto con el toolkit gráfico OpenSceneGraph.

En lo que se refiere a hardware, para implementar la solución sin dispositivos se requiere una computadora que soporte todos estos tipos de programas que consumen altos recursos gráficos, de preferencia una dual core o core2duo se recomienda. Ya para el desarrollo de la aplicación utilizamos una computadora que tiene una tarjeta de video Quadro, que nos permite ver la aplicación en 3D, utilizando los equipos correspondientes, mencionados en el capítulo III.

4.1.1. OpenSceneGraph

OpenSceneGraph [20] (OSG) es un toolkit gráfico de código abierto para el desarrollo de aplicaciones gráficas tales como simuladores de vuelo, juegos, realidad virtual o visualización científica. Está orientado a objetos y construido a partir de la librería gráfica OpenGL y C++ estándar.

OpenSceneGraph es prácticamente un middleware entre las aplicaciones 3D y el rendering que hace bajo OpenGL, como se muestra en el siguiente gráfico:

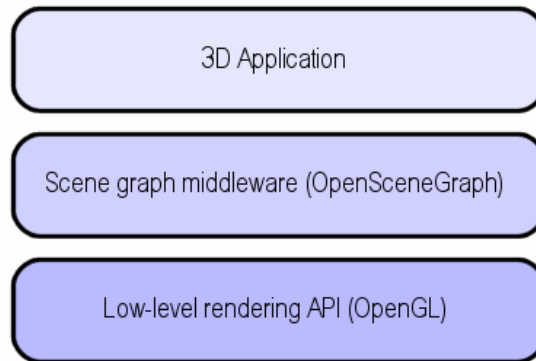


Figura 4. 1 La aplicación 3D en capas

Entre sus principales características tenemos: rendimiento, portabilidad, escalabilidad, soporte para diferentes tipos de archivos de imágenes como .rgb, .gif, .jpg, .png, .tiff, .pic, .bmp, .tga, modelos 3D como LightWave (.lwo), Alias Wavefront (.obj), OpenFlight (.flt), 3D Studio MAX (.3ds), Peformer (.pfb), AutoCAD (.dxf), VRML 1.0 (.wrl), Designer Workshop (.dw), AC3D (.ac); así como soporte multi-lenguaje tales como Java, Lua y Python [21].

4.1.2. Grafos de escena

Un grafo de escena [22] es un grafo dirigido acíclico de nodos que contienen los datos que definen un escenario virtual y controlan un proceso de dibujo. Contiene descripciones de bajo nivel de la geometría y apariencia visual de los objetos, así como

descripciones de alto nivel referentes a la organización espacial de la escena.

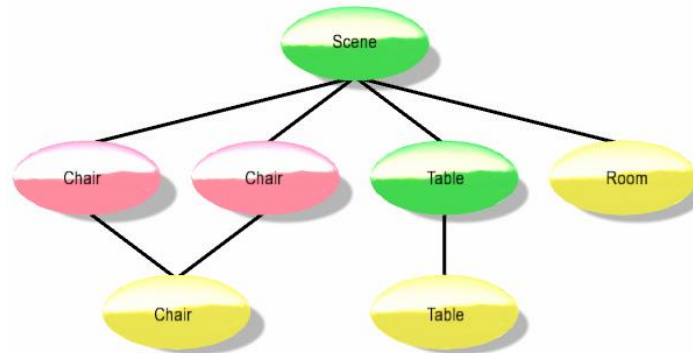


Figura 4. 2 Grafo de escena

El grafo de escena tiene como funciones principales:

- Contribuir a establecer una organización lógica de la escena.
- Establecer dependencias jerárquicas entre distintos sistemas de referencia.
- Permitir el proceso automático de culling (eliminación automática de objetos).
- Facilitar el control de la escena por parte del usuario.

4.1.3. Arquitectura de OSG

La arquitectura de OSG [23] está compuesta por el núcleo de OSG(osg core), los nodeKits y los plugins de OSG; así como aplicaciones y ejemplos.

El núcleo de OSG es donde están las librerías más útiles como osg, osgDB que permite el manejo de plugins, osgUtil, osgViewer que maneja la escena, osgGA que permite el manejo de eventos.

Los NodeKits son librerías extendidas del núcleo de OSG que permite el manejo de efectos, manipular objetos, terrenos, sombras, etc.

Los plugins son para el soporte de entrada/salida en una aplicación para imágenes 2D o formatos 3D

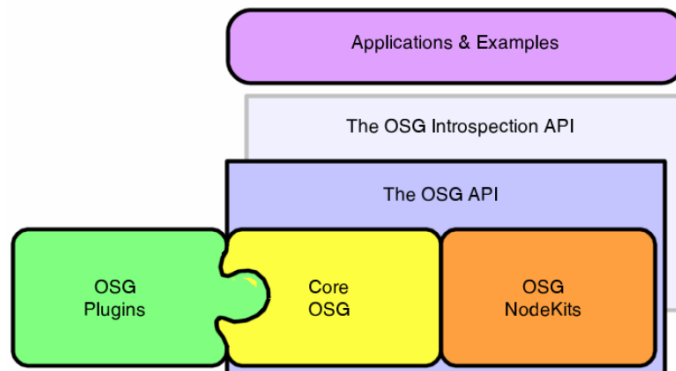


Figura 4. 3 Arquitectura de OSG

4.2. Implementación sin uso de dispositivos de realidad virtual.

4.2.1. Manejo de la cámara

La clave para el manejo de la cámara está en la manipulación de matrices y considerar 3 aspectos importantes que son: el centro, la rotación y la distancia:

```
osg::Vec3d m_c; // center
osg::Quat m_q; // rotation
float m_d; // distance
```

El centro de la cámara es hacia donde apunta la cámara, en otras palabras es el ojo de la misma y como notamos es un vector que

tiene los 3 ejes de referencia: X, Y, Z y al cambiar este centro con respecto a cada frame en relación con el tiempo, al presionar una tecla obtenemos el movimiento deseado. La rotación es un vector de 4 dimensiones que considera las operaciones de una matriz que ha sido transformada para obtener la rotación requerida. La distancia solo es considerada como punto de ubicación de la cámara en la escena.

El siguiente algoritmo muestra como se implementó la manipulación de la cámara:

Si Presiona tecla "w" Entonces Activo movimiento hacia adelante

Si Presiona tecla "s" Entonces Activo movimiento hacia atrás

Si Presiona tecla "q" Entonces Activo movimiento hacia arriba

Si Presiona tecla "w" Entonces Activo movimiento hacia abajo

Luego se tiene el evento por cada frame:

Para cada frame:

Si `m_timeLast` es igual a cero entonces

 Tomo el tiempo actual: `m_timeLast = ea.getTime();`

 Considero un tiempo delta: `m_timeDelta = 0.0;`

Sino

Saco la diferencia de tiempos: $m_timeDelta = t - m_timeLast$;

Al tiempo actual se lo asigno a t: $m_timeLast = t$;

Fin Si

Esto se lo hace para que en cada frame con el paso del tiempo se ejecute una acción. Esto es similar a un loop en una ventana en c por ejemplo.

Luego para los eventos con teclado se tiene

Si Activo movimiento hacia adelante entonces ejecuto acción de mover hacia adelante

Si Activo movimiento hacia atrás entonces ejecuto acción de mover hacia atrás

Si Activo movimiento hacia arriba entonces ejecuto acción de mover hacia arriba

Si Activo movimiento hacia abajo entonces ejecuto acción de mover hacia abajo

Si Activo movimiento hacia derecha entonces ejecuto acción de mover hacia derecha

Si Activo movimiento hacia izquierda entonces ejecuto acción de mover hacia izquierda

Para el manejo adecuado de la cámara se utiliza el teorema de Pitágoras el cual establece la relación entre la hipotenusa y los catetos, y estableciendo los signos correctos obtenemos el movimiento deseado. Además se ha considerado tener diferentes velocidades debido a que el movimiento en las diferentes direcciones no se debe realizar de la misma manera, así consideramos el movimiento hacia adelante o hacia atrás debe ser más rápido, que el movimiento hacia los costados, y la rotación debe ser la más lenta de todos los movimientos, como se muestra a continuación:

Las ecuaciones para el movimiento en general son:

$$\text{hipotenusa} = m_timeDelta * \text{velocidad};$$

$$\text{adyacente} = \text{hipotenusa} * \cos(\text{AnguloRadianes}(\text{anguloZ}));$$

$$\text{opuesto} = \text{hipotenusa} * \sin(\text{AnguloRadianes}(\text{anguloZ}));$$

Donde hipotenusa es el cambio de movimiento con respecto a cada frame; adyacente y opuesto es lo que se obtiene al haber rotado la cámara un anguloZ para que de esta manera se obtenga un movimiento resultante. Luego para obtener los movimientos hacia las diferentes direcciones, tenemos que modificar el centro de la cámara como se muestra a continuación:

Movimiento hacia adelante:

$m_c.x() -= \text{opuesto};$

$m_c.y() += \text{adyacente};$

Movimiento hacia atrás:

$m_c.x() += \text{opuesto};$

$m_c.y() -= \text{adyacente};$

Movimiento hacia la derecha:

$m_c.x() += \text{adyacente};$

$m_c.y() += \text{opuesto};$

Movimiento hacia la izquierda:

$m_c.x() += \text{adyacente};$

$m_c.y() += \text{opuesto};$

Cabe recalcar que las velocidades establecidas para el movimiento hacia adelante/atrás y el movimiento hacia la derecha/izquierda son distintas ya que en la aplicación al moverse hacia adelante o hacia atrás necesitamos hacerlo de una manera más rápida, mientras que hacia los lados, el

movimiento debe ser más lento, de ahí que se decidió que la velocidad hacia adelante/atrás debe ser el triple de la velocidad hacia los lados. Esto en base a prueba y error.

Por último tenemos el algoritmo para la rotación de la cámara. Para ello necesitamos una matriz de rotación y establecer los diferentes parámetros con lo cual se va a permitir la rotación, en este caso solo utilizaremos parámetros para rotar con respecto a X y Z:

Para la rotación en general definimos lo siguiente:

Matriz de rotación: `Matrixd orbitRotation;`

Rotación con respecto al eje Z positivo (rotación hacia izquierda):

`rotateZ += m_timeDelta*velocidad;`

Rotación con respecto al eje Z negativo (rotación hacia derecha):

`rotateZ -= m_timeDelta*velocidad;`

Rotación con respecto al eje X positivo (rotación hacia arriba):

`rotateX += m_timeDelta*velocidad;`

Rotación con respecto al eje X negativo (rotación hacia abajo):

`rotateX -= m_timeDelta*velocidad;`

Por último realizar la rotación con respecto a los parámetros establecidos:

```
orbitRotacion.Rotar(rotateX, rotateZ);
```

4.2.2. Selección de objetos

Para seleccionar objetos y realizarles cambios como moverlos, rotarlos, cambiar un objeto por otro, implementamos una clase que maneja un evento llamado pick. El algoritmo siguiente describe como se seleccionan los objetos:

Obtenemos la posición del objeto que esta sobre el puntero.

Calculamos la proyección del objeto que esta sobre el puntero.

Realizamos una intersección con el objeto señalado. Si hay más de un objeto, tomar el más próximo.

Para mostrar cómo se selecciono el objeto lo marcamos con un sinnúmero de líneas sobre el mismo.

Este método utiliza proyecciones para realizar las intersecciones con los objetos y de esta manera permitir la selección de los mismos. La proyección es normalizada con respecto al tamaño del viewport.

4.2.3. Menú de objetos

Para la implementación de un menú de objetos se utilizó una clase llamada Camera y utilizando proyecciones y matrices se pudo mantener un menú fijo en la escena 3D, agregándole a unos quads(figura de cuatro lados) imágenes para poder ser apreciado como un menú. De esta manera se posiciona cada quad de acuerdo a una referencia en la pantalla de la escena. Cada imagen es cargada de acuerdo a la ruta de un archivo.

4.2.4. Movimiento y rotación de objetos

La implementación del movimiento y rotación de objetos conlleva saber en qué ubicación se encuentra el objeto actualmente y guardar las posiciones a las cuales fueron movidos, de igual manera con la rotación. Además debe considerarse para la rotación de objetos la colocación del pivote, o sea con respecto a que ejes va a rotar, de ahí que es necesario ubicar el pivote en el centro del objeto. El siguiente algoritmo detalla lo explicado aquí:

Si esta el nodo seleccionado entonces:

Obtener la posición del nodo

Obtener el centro del nodo

Establecer un pivote en el centro del objeto (nodo)

Establecer la posición del objeto de acuerdo al tipo de movimiento que realice el usuario.

Fin Si

Es así como obtenemos la traslación y rotación de objetos a partir del centro del pivote.

4.3. Implementación usando dispositivos de realidad virtual.

4.3.1. Guantes de datos, trackers y sensores.

Los dispositivos de realidad virtual como los guantes de datos y trackers son dispositivos que nos permiten trabajar con nuestra aplicación a través de las librerías de OSG. Los guantes de datos se conectan mediante puerto usb y se distinguen mediante dos cadenas como "usb0" para el guante derecho y "usb1" para el guante izquierdo; de esta manera podemos obtener los valores de cada uno de estos mediante una función `gesto=fdGetGesture(glove)` que nos retorna un número entero

entre 0 y 15 para poder determinar que gesto es de acuerdo a una tabla de gestos predefinidos en el capítulo V.

Los dos sensores conectados al tracker están dispuestos de manera que se distingan para qué guante es cada sensor; así establecimos que el sensor conectado en un puerto impar en el tracker son para el guante derecho, y el sensor conectado en un puerto par son para el guante izquierdo.

4.3.2. Manejo de la cámara

La implementación del manejo de la cámara usando los dispositivos de realidad virtual solo cambia en el sentido de que deben mapearse los movimientos con los trackers y con los guantes, es así que al realizar un gesto particular obtenemos el movimiento de traslación y rotación de la cámara. Para el manejo de la cámara solo utilizamos el guante derecho. El algoritmo siguiente muestra lo descrito:

Si el guante derecho y el tracker están conectados entonces:

Obtenemos el gesto

Si el gesto es el 15 entonces:

Establecemos los umbrales de movimiento: $X=3$, $Y=2$, $Z=6$, $\text{Alpha}=20$ y $\text{Beta}=20$.

Obtenemos los valores del sensor.

Para cada valor del sensor obtenido inicialmente lo comparamos con el valor obtenido actualmente mas el umbral entonces realizamos el movimiento deseado. Esto es trasladarse hacia adelante/atrás, izquierda/derecha, arriba/abajo rotación hacia arriba/abajo, izquierda/derecha.

Fin Si

Fin Si

Si el gesto es diferente de 15 entonces se inicializan los valores para el sensor. De esta manera el movimiento se lo realiza a partir de que nuevamente mantenga el gesto 15.

4.3.3. Selección, menú, movimiento y rotación de objetos

De igual manera como en la implementación anterior se mapea las diferentes acciones como la selección de objetos, es decir con algún gesto en particular del guante; de igual manera el menú. El movimiento y rotación de objetos se lo hace con un gesto en

particular para poder sentir los valores del tracker. En este caso determinamos las acciones de selección de objetos (pick) con el guante derecho, el gesto 14 y el con el guante izquierdo con el gesto 9 el menú y con el gesto 0 el movimiento de objetos.

4.3.4. Manejo del cursor.

Para la implementación del manejo del cursor se tomo en consideración un diferencial en dos ejes para poder mapear lo que sienten los tracker y de esta manera reflejar el movimiento respectivo del cursor. De igual manera el cursor empieza a moverse con un gesto del guante derecho y lo reseteamos con el mismo gesto pero con el guante izquierdo. El siguiente algoritmo muestra lo descrito anteriormente:

Si el gesto derecho es igual a 1

Obtengo los valores para el sensor correspondiente al cursor

Calculo un diferencial de acuerdo a las posiciones iniciales y actuales del cursor, como se muestra a continuación:

$dx = (\text{PosXSensorFinal} - \text{PosXIniCursor}) * 300;$

$dz = (\text{PosZSensorFinal} - \text{PosZIniCursor}) * 300;$

Por último establecemos la posición del cursor:

```
PosCursor(ancho+dx,alto+dz);
```

Fin Si

4.4. Prototipo de la implementación.

El prototipo mostrado es una versión beta de la solución, que muestra todo lo descrito anteriormente. Mostramos en cinco gráficos el prototipo con las acciones más importantes:

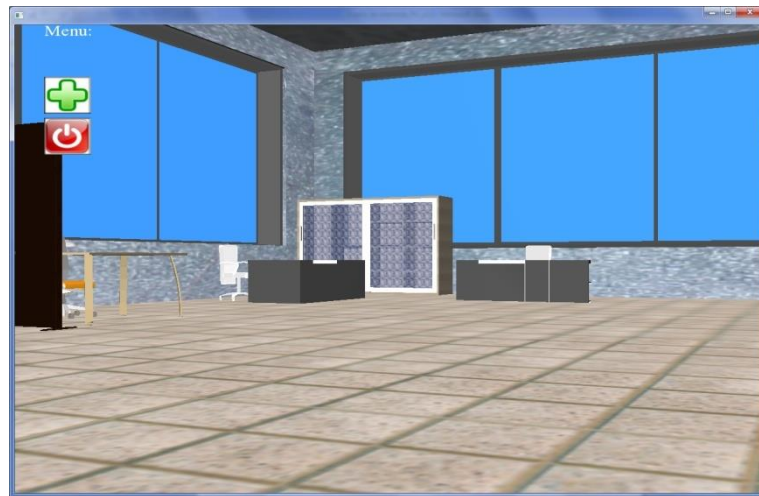


Figura 4. 4 Escena Principal

La figura 4.4 muestra la escena principal de la aplicación como vemos es un espacio compuesto por ventanas, piso y una mini sala de juntas. Además se muestra el menú principal, que como vemos el

icono del signo más es para agregar objetos, el otro icono es para salir de la aplicación.

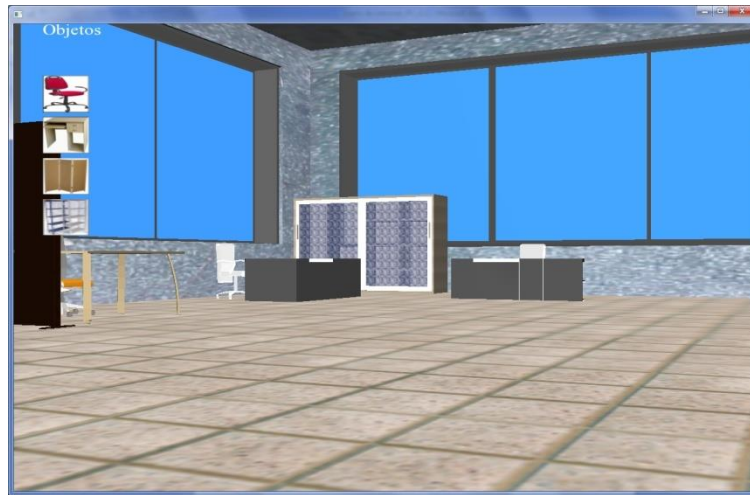


Figura 4. 5 Menú de Objetos

La figura 4.5 muestra el menú de los objetos que incluye: sillas, escritorios, biombos y anaqueles.

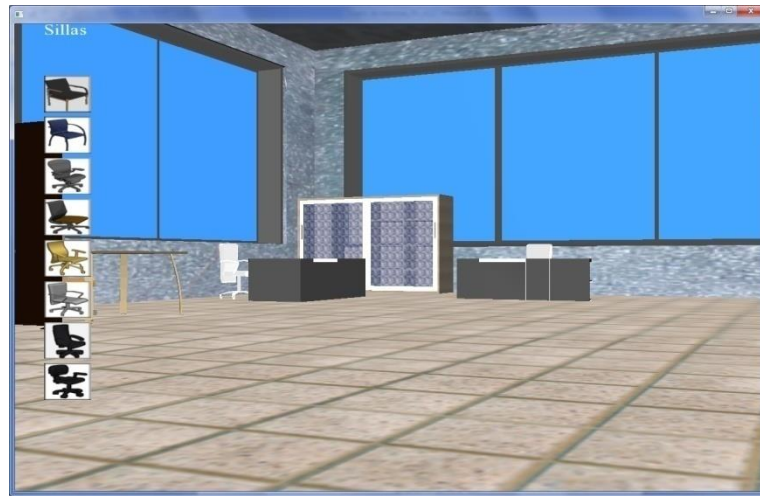


Figura 4. 6 Menú de Sillas

Luego podemos seleccionar cualquiera de esos ítems del menú y colocarlos en la escena. En este caso en las figuras 4.6 y 4.7 tenemos el menú de sillas y escritorios respectivamente.

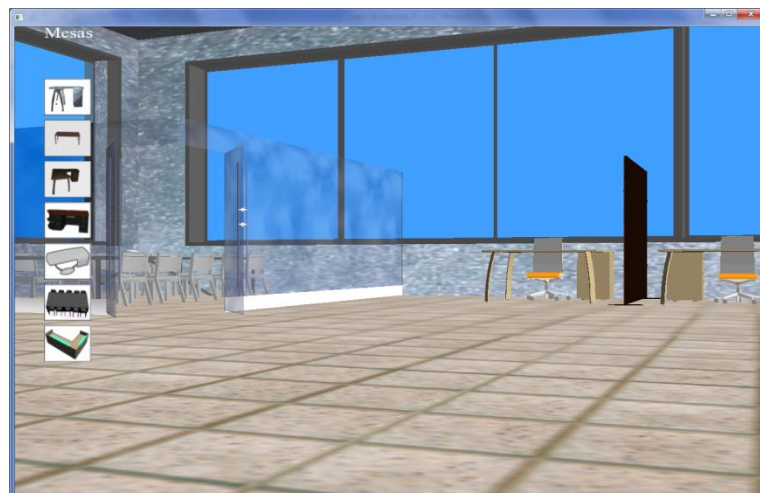


Figura 4. 7 Menú de Escritorios



Figura 4. 8 Menú de biombos

Las figuras 4.8 y 4.9 muestran el menú de biombos y el menú de anaqueles respectivamente.



Figura 4. 9 Menú de Anaqueles



Figura 4. 10 Menú de objeto seleccionado

La figura 4.10 indica que al hacerle pick al objeto, se muestra el objeto seleccionado y podemos ver un menú diferente a los anteriores que nos permite realizar diferentes acciones, el primer ícono desde la parte superior es para quitar la selección del objeto, el segundo ícono es para cambiar un objeto por otro del mismo tipo, el tercer ícono es para cambiar el objeto por otro de cualquier tipo y el último ícono es para eliminar el objeto seleccionado.

CAPÍTULO V

5. Plan de pruebas.

El presente capítulo describe las pruebas realizadas con los dispositivos de realidad virtual como lo son el guante de datos y los sensores. La configuración de gestos a partir de las pruebas realizadas.

5.1. Pruebas, configuraciones y resultados.

Las pruebas realizadas se basan en los gestos predeterminados del guante y la facilidad con que el usuario puede interactuar con la aplicación, de tal manera que puedan realizarse de una manera correcta y precisa.

Las configuraciones son el resultado de estas pruebas realizadas que determinan qué gestos son los más apropiados para que el usuario interactúe con la aplicación.

5.2. Gestos predefinidos.

La siguiente figura muestra los gestos predefinidos en la librería del guante: fdglove.lib que nos proporciona 16 gestos para poder interactuar con la aplicación.



Figura 5. 1 Gestos predefinidos

5.3. Pruebas con el guante de datos en la aplicación.

Para poder determinar que gestos son los apropiados para interactuar con la aplicación y obtener los mejores resultados se procedió a probar cada uno de los gestos predefinidos tomando en cuenta varios aspectos entre ellos: la facilidad con que se realiza el gesto midiendo su nivel de dificultad: 0-ninguna, 1-moderada, 2-alta, el tipo de guante si es derecho o izquierdo, y si el gesto que se está realizando se confunde con otro gesto similar. Las pruebas se realizaron con varios usuarios, en este caso se las realizó con 10 usuarios diferentes en el tamaño y grosor de su mano, que conocen el manejo de sistemas y se obtuvo la siguiente tabla:

GD: Guante derecho

GI: guante izquierdo

Gesto	Dificultad (0-2)		Problema con gesto	
	GD	GI	GD	GI
0	0	0	-	-
1	0	0	-	-
2	1	1	6	6
3	1	1	11	11
4	2	2	12, 8	12, 8
5	2	2	13	13
6	2	2	7, 14	14

Tabla 5. 1a Prueba de Gestos

Gesto	Dificultad (0-2)		Problema con gesto	
	GD	GI	GD	GI
7	0	1	-	3
8	1	1	12	14
9	0	0	-	-
10	1	1	14	14
11	0	0	-	-
12	1	1	8	8
13	0	0	-	-
14	1	0	12	-
15	0	0	-	-

Tabla 5. 1b Prueba de Gestos

5.4. Configuraciones de gestos y resultados.

De las pruebas realizadas se pudo determinar cuáles son los gestos apropiados para interactuar con la aplicación de tal manera que sean fáciles de realizar y a su vez que no existan problemas con otros gestos. Para una mejor interacción con la aplicación utilizamos los dos guantes, aprovechando de esta manera los gestos más sencillos de utilizar, y a su vez que el usuario pueda recordar.

Se pudo obtener 6 gestos que son fáciles de realizar y que no tenían ningún problema con otros gestos que son el gesto 0, 1, 9, 11, 13 y

15; y los gestos 7 y 14 son fáciles de realizar pero tienen un inconveniente menor porque se confunden levemente con otro gesto. De aquí que se determina cuáles son los gestos que se seleccionaron para la interacción con la aplicación, como se muestra en la siguiente tabla:

Gesto	Acción					
	Objetos		Cámara		Cursor	
	GD	GI	GD	GI	GD	GI
0	-	Mover/Rotar	-	-		
1	-		-	-	Mover	Reset
9	Menu		-			
14		Pick	-			
15			Mover/Rotar	-		
!0		Reset				
!15			Reset			

Tabla 5. 2 Configuración de Gestos

5.5. Configuraciones de tracker y sensores.

Para la configuración que permite sensar las distintas posiciones ya sea de la cámara, objetos o del cursor, se determinó el uso de dos sensores que pueden ser conectados en cualquiera de los conectores del tracker siendo las entradas impares para el movimiento de traslación y rotación de la escena y los valores pares para la

traslación, rotación de objetos y movimiento del cursor, obtenidos a partir de los gestos establecidos en la tabla 5.2 de configuración de gestos.

El tracker es un dispositivo muy preciso al sensor por lo que el movimiento de traslación y rotación de objetos es inmediato al ejecutar la aplicación, por lo que se necesitó establecer un umbral para el movimiento y rotación de las diferentes acciones descritas en la tabla 5.2. De ahí que se estableció un umbral para la traslación y un umbral para la rotación como se muestra en la siguiente tabla:

Umbral traslación			Umbral rotación	
X	Y	Z	Alfa	Beta
3	2	6	20	20

Tabla 5. 3 Umbrales de Movimiento

Los umbrales de traslación son de aproximadamente igual a 3 cm, lo que quiere decir que debemos movernos a esa distancia del centro de referencia para poder realizar el movimiento de traslación continuo requerido; de igual manera si regresamos dentro del volumen de esta región prácticamente cúbica el movimiento no se realizará. Los umbrales de rotación establecen los ángulos en grados para poder

determinar la rotación continua de las diferentes acciones establecidas en la tabla 5.2.

CONCLUSIONES Y RECOMENDACIONES

1. Conseguimos desarrollar una aplicación que interactúa con dispositivos de realidad virtual que sirven de punto de entrada como los guantes y sensores, y de salida como los son el proyector y las gafas activas, los cuales nos permitieron realizar todas las acciones que nos propusimos en este proyecto de una manera fácil de usar, fácil de interactuar y fácil de recordar.
2. Para el manejo de los dispositivos en conjunto, necesitamos la ayuda de un umbral para la rotación y un umbral para la traslación, para poder tener una mayor interacción entre ellos. Dichos umbrales son inicializados dependiendo del gesto que activa el movimiento, permitiendo mayor comodidad al usuario para que no necesite recodar su punto inicial.
3. La complejidad de manejar los dispositivos fue resuelta con el uso de los dos guantes lo que permite al usuario usar los gestos que son más

fáciles. Y con ello también evitamos que los usuarios se confundan al no tener que memorizar mayor cantidad de gestos.

4. Al permitir al usuario manejar la dirección del movimiento de la cámara, se gana mayor facilidad en la inmersión y en la apreciación de la escena, esto es apreciar diferentes velocidades en las diferentes direcciones; los movimientos hacia las direcciones hacia adelante y hacia atrás tienen velocidades mucho mayores a las velocidades de movimientos hacia los lados, arriba y abajo, así como las velocidades de rotación. Estas velocidades se definieron de acuerdo a como se interactuaba con la aplicación así determinamos que las velocidades hacia adelante y hacia atrás deben ser el triple de las demás velocidades mencionadas en la otras direcciones.
5. Interactuar con una aplicación de realidad virtual inmersiva para la decoración de oficinas resulta costoso pero a su vez es de mucha utilidad no solo para decorar sino también para poder mostrar al cliente el resultado final de la decoración.

Se debe tener en cuenta ciertas recomendaciones para facilitar el uso apropiado de la aplicación, por lo tanto:

1. Cabe mencionar que las pruebas se realizaron con 10 usuarios con conocimientos en uso de sistemas, por lo que se recomienda que se hagan pruebas con otros tipos de usuarios como decoradores o arquitectos que conocen del tema de decoración pero que no conocen este tipo de sistemas en los cuales se interactúa con dispositivos de realidad virtual.
2. Es recomendable que el usuario antes de la utilización del sistema, tenga una previa capacitación y practica, para que se familiarice con las herramientas, puesto que necesita saber los gestos del guante que debe utilizar y los umbrales de los movimientos para mejor manejo.
3. Se recomienda que solo se utilicen los gestos que en las pruebas indican que tienen menos conflictos con otros gestos, para que no haya resultados no esperados en la aplicación.

4.

BIBLIOGRAFIA

- [1] “Decoración de interiores”, <<http://www.arqhys.com/interiores-decoracion.html>> [Consulta: 20/07/2009].
- [2] “Decoración de interiores”, <<http://www.misrespuestas.com/que-es-la-decoracion-de-interiores.html>> [Consulta: 20/07/2009].
- [3] “Decoración de interiores”, <http://www.tarotistas.com/secciones/fengshui/Feng_Shui_para_tu_oficina> [Consulta: 22/08/2009].
- [4] Mario Gutierrez, Frédéric Vexo, Daniel Thalmann, “Stepping into Virtual Reality”, EPFLVR Lab, Lausanne Switzerland, Springer-Verlag, 2008.
- [5] “Realidad virtual”, < <http://www.monografias.com/trabajos4/realvirtual/realvirtual.shtml> > [Consulta: 22/08/2009].
- [6] “Realidad virtual inmersiva y no inmersiva”, <<http://www.activamente.com.mx/vrml/>> [Consulta: 22/08/2009].
- [7] “Guante de datos”, < <http://www.siasistemas.com/Sitio2/02120305.htm> > [Consulta: 29/09/2009].
- [8] “HMD”, < <http://www.siasistemas.com/Sitio2/02120301.htm> > [Consulta: 29/09/2009].

- [9] “Gafas estereoscópicas”, < http://es.wikipedia.org/wiki/Gafas_3D >
[Consulta: 29/09/2009].
- [10] “Tracker”, < http://www.polhemus.com/?page=Motion_Liberty >
[Consulta: 29/09/2009].
- [11] “Proyector 3D”, < [http://www.reald.com/Content/proProducts.aspx
?page ID=124](http://www.reald.com/Content/proProducts.aspx?page%20ID=124) > [Consulta: 29/09/2009].
- [12] “Decoración de interiores”, < [http://www.arqhys.com/interiores-
decoracion.html](http://www.arqhys.com/interiores-decoracion.html) > [Consulta: 10/10/2009].
- [13] “Decoración de interiores”, < [http://www.buscarinformacion.com/
decoracion/decoracion_de_oficinas.html](http://www.buscarinformacion.com/decoracion/decoracion_de_oficinas.html) > [Consulta: 10/10/2009].
- [14] “Herramientas 3D”, < [http://es.wikipedia.org/wiki/Gráficos_3D_por_
computadora](http://es.wikipedia.org/wiki/Gr%C3%A1ficos_3D_por_computadora) > [Consulta: 07/11/2009].
- [15] “Maya”, < [http://usa.autodesk.com/adsk/servlet/pc/index?id=13577897
&siteID=123112](http://usa.autodesk.com/adsk/servlet/pc/index?id=13577897&siteID=123112) > [Consulta: 17/06/2010].
- [16] “Blender”, < <http://www.blender.org/> > [Consulta: 17/06/2010].
- [17] “LightWave”, < <http://www.newtek.com/lightwave/> > [Consulta:
17/06/2010].
- [18] “Cinema 4D”, < <http://www.maxon.net/index.php?id=18&L=0> >
[Consulta: 17/06/2010].
- [19] “Houdini”, < <http://www.sidefx.com/> > [Consulta: 17/06/2010].
- [20] “OpenSceneGraph”, <[http://www.openscenegraph.org/projects/osg/wi
ki/About/Introduction](http://www.openscenegraph.org/projects/osg/wiki/About/Introduction) > [Consulta: 17/10/2009].

- [21] Paul Martz, "OpenSceneGraph Quick Start Guide, A Quick Introduction to the Cross-Platform Open Source Scene Graph API", Skew Matrix Software LLC, California, Mountain View, 2007.
- [22] "Grafo de escena", < http://encuentrosjava.uji.es/SegundosEncuentros/materiales/workshop_3d.pdf > [Consulta: 17/10/2009].
- [23] Paul Martz, "OpenSceneGraph Quick Start Guide, A Quick Introduction to the Cross-Platform Open Source Scene Graph API", Skew Matrix Software LLC, California, Mountain View, 2007.