



ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

“Modelamiento y programación de un juego de LEGOS en un entorno de Realidad Virtual.”

INFORME DE MATERIA DE GRADUACIÓN

Previo a la obtención del Título de:

INGENIERO EN COMPUTACIÓN ESPECIALIZACIÓN SISTEMAS MULTIMEDIA
INGENIERO EN CIENCIAS COMPUTACIONALES ESPECIALIZACIÓN SISTEMAS
MULTIMEDIA
INGENIERO EN COMPUTACIÓN ESPECIALIZACIÓN SISTEMAS MULTIMEDIA

Presentado por

FREDDY ALEJANDRO ARBOLEDA MONCAYO

RICARDO PATRICIO LAICA CORNEJO

MARÍA MAGDALENA LOOR ROMERO

Guayaquil – Ecuador

2010

DEDICATORIA

Dedico este trabajo y todos mis estudios universitarios a Dios, por haberme dado la oportunidad de empezar y culminar mi carrera satisfactoriamente. A mi familia que fue mi sustento en toda mi carrera universitaria y a mi novia que nunca dejó de apoyarme para la culminación de la misma.

Freddy Arboleda Moncayo.

Dedico este proyecto a mis padres, hermanos y familiares, quienes han estado a mi lado siempre brindándome su apoyo incondicional.

Ricardo Laica Cornejo.

Este trabajo está dedicado a mis padres porque gracias a su apoyo constante he podido salir adelante, a mis hermanos por ser mi ejemplo a seguir, a mi segunda familia que me aceptaron y consideraron como a una hija más y a Christian por ser siempre incondicional conmigo.

Magdalena Loor Romero.

AGRADECIMIENTO

A mis compañeros de grupo por su colaboración y empuje en la realización de este trabajo de tesis de grado. A todos mis amigos que a lo largo de mis estudios universitarios fui conociendo en las diversas materias de la carrera, cuyo aporte fue de vital importancia para lograr el objetivo final trazado.

Freddy Arboleda Moncayo.

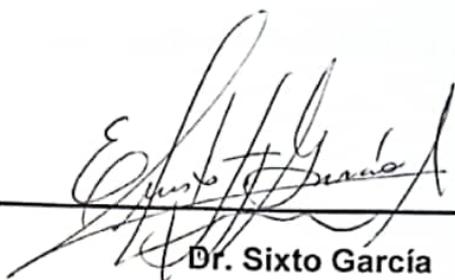
A Dios, a mis padres, hermanos y amigos.

Ricardo Laica Cornejo.

A Ricardo y a Freddy porque juntos logramos que esto fuera posible.

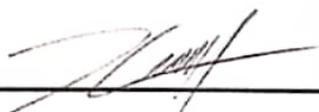
Magdalena Loor Romero.

TRIBUNAL DE SUSTENTACIÓN



Dr. Sixto García

PROFESOR DE MATERIA DE GRADUACIÓN



Dr. Xavier Ochoa Chehab

PROFESOR DELEGADO DEL DECANO

DECLARACIÓN EXPRESA

"La responsabilidad del contenido de este Trabajo de Grado, nos corresponde exclusivamente; y el patrimonio intelectual de la misma, a la Escuela Superior Politécnica del Litoral".

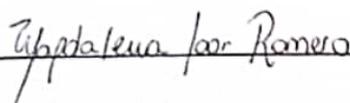
(Reglamento de exámenes y títulos profesionales de la ESPOL)



Freddy Alejandro Arboleda Moncayo



Ricardo Patricio Laica Cornejo



María Magdalena Loo Romero

RESUMEN

La creatividad es la cualidad del ser humano que le permite generar nuevos universos, ampliando el mundo de lo posible, es la capacidad de un cerebro de crear, transformar y de llegar a conclusiones nuevas y resolver problemas en una forma original.

Entre los juegos tradicionales que son conocidos por fomentar la creatividad, encontramos uno que por su simplicidad, formas, colorido y variedad, se ha convertido en el favorito de muchas generaciones, los bloques de Lego.

A pesar de que los Legos han formado parte de nuestros juguetes durante aproximadamente 50 años, las características de la vida actual han hecho que estos hayan quedado relegados y hayan sido reemplazados por pasatiempos más atractivos e interactivos, algunos de los cuales no contribuyen en nada al desarrollo cognitivo y creativo.

Nuestro proyecto busca por medio del uso de las técnicas de la Realidad Virtual crear un mundo virtual del juego tradicional de los bloques de LEGO, para comprobar la aplicabilidad de los conceptos aprendidos a lo largo de la materia de graduación.

En este trabajo introducimos conceptos referentes a la Realidad Virtual, sus distintos tipos y características.

Se especifica la plataforma y librerías de programación utilizadas en la implementación, los dispositivos de hardware y el análisis realizado en la

calibración de los mismos para utilizarlos de acuerdo a las necesidades del proyecto. Se describe el análisis lógico y el diseño de la aplicación, además se detallan las pruebas realizadas y los resultados obtenidos.

Finalmente, se especifican las conclusiones que obtuvimos de la realización de este trabajo y las recomendaciones para futuras aplicaciones realizadas en esta área.

INDICE GENERAL

DEDICATORIA.....	II
AGRADECIMIENTO.....	III
TRIBUNAL DE SUSTENTACIÓN	IV
DECLARACIÓN EXPRESA	V
RESUMEN.....	VI
INDICE GENERAL	VIII
INDICE DE FIGURAS.....	XII
INDICE DE TABLAS	XIV
ABREVIATURAS	XXVI
INTRODUCCIÓN.....	XXVII
CAPÍTULO 1	1
1. Descripción del Problema.....	1
1.1. Antecedentes	1
1.2. Justificación del proyecto.....	2
1.3. Objetivo general.....	4
1.4. Objetivos específicos	4
1.5. Alcance	5
1.6. Metodología.....	5

CAPÍTULO 2	8
2. Marco Teórico	8
2.1. Interacción en un mundo virtual	8
2.2. Ambientes inmersivos vs. no inmersivos	13
2.2.1. Sistemas Inmersivos	13
2.2.2. Sistemas No inmersivos	15
2.2.3. Sistemas Semi-inmersivos	17
2.2.4. Ventajas y desventajas de los sistemas inmersivos y no inmersivos.	18
2.3. Descripción de algunos sistemas de RV existentes	20
2.4. Descripción de la aplicación no Inmersiva <i>BlockCAD</i>	26
CAPÍTULO 3	28
3. Análisis y diseño de la Aplicación	28
3.1. Análisis de la aplicación	28
3.1.1. Requisitos funcionales	28
3.1.2. Requisitos no funcionales	28
3.1.3. Usuarios de la aplicación	29
3.2. Diseño de la aplicación e interacción	29

3.2.1.	Diseño de los objetos 3D.....	30
3.2.2.	Análisis de colisión de objetos	31
3.2.3.	Interfaz orientada a una aplicación semi-inmersiva.....	33
3.2.4.	Especificación de los dispositivos de entrada y salida que utilizaremos para la interacción.....	34
3.3.	Arquitectura de la aplicación.....	35
3.4.	Diseño lógico	38
3.4.1.	Diagramas UML	38
3.4.2.	Diagramas de secuencia (Escenarios)	39
3.5.	Interfaz de la aplicación	44
3.6.	Diseño de pruebas.....	48
CAPÍTULO 4		50
4.	Implementación y Pruebas	50
4.1.	Software utilizado para la implementación	50
4.1.1.	Modelado de objetos y ambientes 3D	50
4.1.2.	Plataforma y librerías de programación.....	51
4.1.3.	Audio	51
4.2.	Dispositivos de Hardware	52

4.3. Análisis y calibración de los trackers.....	55
4.3.1. Optimización del control y precisión de los trackers.	56
4.3.2. Análisis de algoritmos para corrección de error de los trackers	58
4.3.3. Calibración de dispositivos hápticos	65
4.4. Plan de pruebas	67
4.4.1. Descripción de pruebas.....	67
4.4.2. Evaluación de resultados	69
CONCLUSIONES	75
RECOMENDACIONES.....	77
REFERENCIAS BIBLIOGRÁFICAS	83
ANEXO A	87
ANEXO B	102

INDICE DE FIGURAS

FIGURA 2.1: Diagrama de interacción implícita o directa	10
FIGURA 2.2: Diagrama de interacción explícita o clásica	11
FIGURA 2.3: Imagen del editor gráfico del BlockCAD	27
FIGURA 3.1: Dimensiones guía para la creación de un bloque.	30
FIGURA 3.2: Bloques de Lego modelados en 3D Studio Max	31
FIGURA 3.3: Grafo de opciones para evitar colisiones	32
FIGURA 3.5: Caso de uso de la aplicación	38
FIGURA 3.6: Diagrama de secuencia – Seleccionar bloque Lego	39
FIGURA 3.7: Diagrama de secuencia – Colocar bloque de Lego.....	40
FIGURA 3.8: Diagrama de secuencia – Cambiar color de bloque	41
FIGURA 3.9: Imagen del contenedor de bloques de Lego.	44
FIGURA 3.10: Cambiar el color de un bloque de Lego	45
FIGURA 3.11: Calibración de ubicación de los bloques.....	46
FIGURA 3.12: Castillo creado con la aplicación Lego 3D	47
FIGURA 4.1: Guante de datos 5DT de 14 sensores	53
FIGURA 4.2: Tracker electromagnético Polhemus	54
FIGURA 4.3: Gafas estereoscópicas NuVision 60 GX	54
FIGURA 4.4: Proyector estereoscópico 3D DepthQ.....	55

FIGURA 4.5: Error en desempeño de los trackers	56
FIGURA 4.6: Propagación transmisor omnidireccional	59
FIGURA 4.7: Ejes coordenados del mundo real, trackers y osg	60
FIGURA 4.8: Líneas dibujadas en la base con separación de 5 cm	62
FIGURA 4.9: Funciones de transformación de coordenadas	62
FIGURA 4.10: Orden de las transformaciones realizadas	63
FIGURA 4.11: Valores, gestos y acciones del guante de datos.....	66

INDICE DE TABLAS

TABLA 1: Valores de error según rango del sensor.....	57
TABLA 2: Valores de factor de corrección de los trackers.....	61
TABLA 3: Transformación de coordenadas del mundo real al virtual	63
TABLA 4: Datos comparativos entre las versiones de la aplicación virtual	74

ABREVIATURAS

CAD	Diseño Asistido por Computadora
CAVE	Cave Automatic Virtual Enviroment
GUI	Graphic User Interface
HMD	Head Mounted Display
RV	Realidad Virtual
VRML	Virtual Reality Modeling Language
WIMP	Window Icon Menu Pointing

INTRODUCCIÓN

La Realidad Virtual es un término muy sonado en los últimos años, digno de la ciencia ficción, que ha cambiado nuestra forma de percibir y ha revolucionado el enfoque de estudio en distintas áreas, siendo el más destacado el campo de los videojuegos.

Con la llegada del nuevo milenio se desata el boom de las “sociedades virtuales”, aparecen plataformas como el Nintendo Wii y el PlayStation, que captan la atención de los usuarios a través de estímulos que enriquecen los sentidos, pero que aún no permiten una inmersión completa. Es por esta razón que muchos expertos y fanáticos afirman que el desarrollo de la Realidad Virtual, cambiará totalmente el futuro de los videojuegos, en donde se podrá tener más libertad en la interacción debido al uso de dispositivos muy sofisticados.

Pero el avance de la tecnología, también implica que ciertas actividades vayan quedando a un lado, por lo cual vale hacernos una pregunta: ¿qué pasará con los juegos, ahora llamados “tradicionales”, con los que crecimos; los juegos de mesa, los cuales compartíamos con nuestra familia y amigos y que eran motivo de reunión cualquier día de la semana; quedarán solo como un buen recuerdo?

Nuestro proyecto busca mediante el uso de las técnicas innovadoras de la Realidad Virtual, aplicar conceptos tales como la inmersividad, presencia e interacción, a través de la creación de un ambiente virtual de un juego básico de bloques de LEGO, creado a partir de herramientas de modelamiento de gráficos 3D.

CAPÍTULO 1

1. Descripción del Problema

1.1. Antecedentes

La creatividad es la cualidad del ser humano que le permite generar nuevos universos, ampliando el mundo de lo posible, es la capacidad de un cerebro de crear, transformar y de llegar a conclusiones nuevas y resolver problemas en una forma original.

Todos somos creativos, pero a lo largo de nuestra vida, somos menos conscientes de ella y no le damos la importancia que tiene, para la gran mayoría es desconocido que la creatividad es lo que separa a los “destacados”, del resto de las personas. En la niñez por el contrario es algo innato, no sólo como una forma de expresar sentimientos, sino de descubrimiento del mundo que nos rodea. En esta etapa, el juego es la estrategia más adecuada para estimular el desarrollo de las capacidades creativas a través de la imaginación, el pensamiento y la acción.

Numerosos teóricos y estudiosos han analizado y señalado la importancia que el juego tiene en todas las etapas de la vida del hombre, ya que fomenta el desarrollo de actividades cognitivas como la estrategia, planificación, expresión, integración, capacidad para

resolver problemas y fundamentalmente para estimulación de la creatividad [30].

Entre los juegos tradicionales que son conocidos por fomentar la creatividad, encontramos uno que por su simplicidad, formas, colorido y variedad, se ha convertido en el favorito de muchas generaciones, los bloques de Lego. Quien no recuerda el haber pasado horas uniendo bloques, para crear un mundo plástico inimaginable donde existían casas, personas y objetos con formas extrañas, pero que en nuestra mente tenían sentido.

1.2. Justificación del proyecto

El notable desarrollo tecnológico y el incremento de medios de comunicación, que se encuentran hoy al alcance de la población, han influido notablemente en nuestra forma de vida con respecto a décadas pasadas. Una de las mayores diferencias que podemos nombrar, es la preferencia a los juegos electrónicos o multimedia sobre los juegos tradicionales, actualmente es raro el hogar donde no exista una computadora, Internet o una consola de video juegos, donde niños y adultos ocupan la mayor parte de su tiempo libre.

Existen muchos que consideran que este tipo de juegos no benefician al crecimiento moral e intelectual del niño; sin embargo, muchos

estudios [30] han comprobado que las ventajas superar a las desventajas, confirmando que utilizados de forma razonable, pueden contribuir en los siguientes aspectos:

- Ayudar en el aprendizaje de varios estímulos cognitivos.
- Es un método eficiente en terapias de rehabilitación con niños que tienen problemas de aprendizaje.
- Apoyan en el desarrollo de habilidades psicomotoras.
- Le enseñan al niño formas de aprendizaje rápidas, atractivas y provechosas.
- Ayudan a resolver problemas de dislexia.
- Pueden ser de gran apoyo para niños con discapacidad mental.
- Mejoran la capacidad para aprender y analizar información.
- Contribuyen a controlar el estrés y la hiperactividad.

Nuestro proyecto busca a través de la creación de un entorno virtual del juego básico y muy conocido de bloques de LEGO, aplicar los conceptos fundamentales de la Realidad Virtual, que nos permitan sentar bases para posteriores usos de ésta tecnología, que actualmente está revolucionando la computación a nivel mundial.

1.3. Objetivo general

El objetivo primordial de este proyecto es crear un ambiente 3D conformado por piezas de LEGOS, que permita al usuario interactuar en un entorno semi-inmersivo, mediante el uso de técnicas y conceptos estudiados a lo largo de la materia de graduación.

1.4. Objetivos específicos

Al desarrollar el siguiente proyecto se persiguen los siguientes objetivos:

- Crear un ambiente virtual aplicando los conceptos fundamentales como Inmersión, Presencia e Interacción.
- Crear una aplicación no solo llamativa, sino que le permita al usuario interactuar con ella, de manera fácil y con naturalidad de movimientos.
- Implementar un procedimiento que nos permita manipular los trackers con alto grado de precisión y concordancia en los movimientos dentro del ambiente virtual.

1.5. Alcance

El siguiente proyecto de graduación consiste en un prototipo funcional semi-inmersivo en 3D del juego de bloques de LEGO, que será utilizado dentro del laboratorio de Realidad Virtual de la Facultad de Ingeniería Eléctrica y Computación, para aplicar los conceptos básicos y técnicas de esta rama de la computación. Además se podrá utilizar como guía para futuros proyectos que se realicen en esta área, en aspectos tales como el manejo de la interacción entre los dispositivos de entrada/salida y una aplicación virtual.

1.6. Metodología

La metodología en que nos basamos para desarrollar la aplicación es el modelo iterativo, ya que se partió de requerimientos básicos para crear una primera versión, la cual era mejorada continuamente, una vez que en la etapa de pruebas surgían nuevas necesidades en cuanto al diseño o en la interacción con el usuario.

Comenzamos por el diseño de la interfaz de la aplicación, la cual está conformada por objetos 3D creados con programas de modelamiento de gráficos, que simulan los bloques de LEGO, los

mismos que son manipulados por el usuario a través de dispositivos de entrada y salida no convencionales. Las herramientas utilizadas para la creación de la interfaz virtual y el modelamiento de objetos son las siguientes:

- OpenSceneGraph
- 3D Studio Max
- Visual Studio 2008

Se utiliza C++ como lenguaje de programación y el formato osg para el modelamiento de los objetos.

En nuestro proyecto el usuario interactúa con el ambiente virtual a través de un guante para elegir, coger y ubicar los bloques, un tracker que captura el desplazamiento de la mano del usuario a través de la escena y un par de gafas estereofónicas que le permite observar a los objetos 3D desde la pantalla donde es proyectada. Además dentro del ambiente virtual se incluyen archivos de sonidos en formato wav o mp3, para colaborar con la retroalimentación de los movimientos que se realizan.

Las pruebas del prototipo fueron realizadas dentro del laboratorio de Realidad Virtual, estuvieron enfocadas principalmente en medir el grado de precisión en los movimientos realizados con los trackers, cuando el usuario interactuaba con los objetos del

entorno virtual. Estas pruebas se hacían periódicamente para corregir errores las fallas existentes que mermaran en la eficiencia de la aplicación.

CAPÍTULO 2

2. Marco Teórico

2.1. Interacción en un mundo virtual

La tecnología de la Realidad Virtual permite al usuario sumergirse en un mundo simulado por un computador de forma interactiva y en tiempo real [5]. Por medio de ella podemos estar en una réplica exacta de lugares existentes o en un mundo imaginario, aprender a utilizar equipos con los que no contamos físicamente, además enfrentar y superar nuestros miedos y fobias.

Por tal motivo es necesario crear ambientes creíbles, que puedan ser percibidos y manipulados de manera fácil e intuitiva. Las tres características primordiales en Realidad Virtual son: La Presencia, la Inmersión y la Interacción [9], [11].

PRESENCIA

Es un concepto subjetivo y radica en el comportamiento de la psiquis del ser humano. Es la sensación de estar en un ambiente diferente al que realmente se está, provocado como respuesta a la combinación de distintos estímulos sensoriales.

INMERSIÓN

Se da cuando el usuario bloquea toda distracción del mundo real y sólo percibe el entorno virtual. Según los dispositivos que se utilice se logrará un grado de inmersión total o parcial en la aplicación, más adelante en el documento encontraremos más información sobre este tema.

INTERACCIÓN

Hoy en día la dependencia que tienen las personas con las computadoras es destacable, por lo cual la interacción es uno de los aspectos más críticos, el grado de percepción y de inmersión en un ambiente simulado dependen de la forma como el usuario se comunica y manipula su espacio.

Según los dispositivos de entrada y salida que se utilicen podemos definir dos tipos principales de formas de interactuar con una aplicación [6]:

- La interacción implícita
- La interacción explícita

La interacción implícita o directa se basa en movimientos naturales y no limitados, el usuario interactúa con los objetos virtuales exactamente igual como lo haría con objetos reales, no requiere de aprendizajes previos.

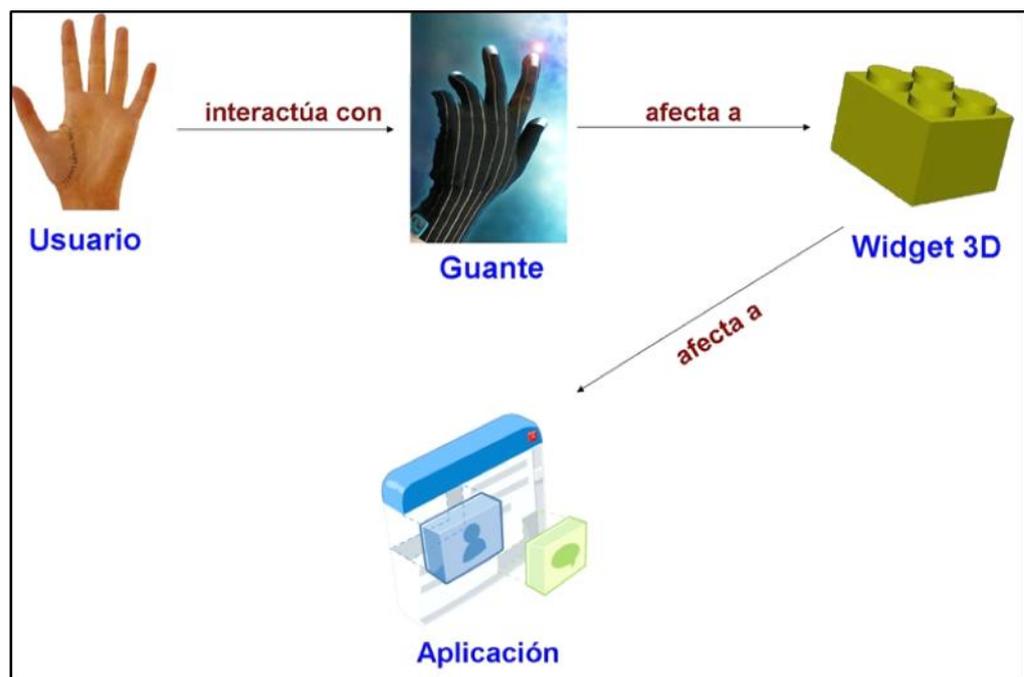


FIGURA 2.1: Diagrama de interacción implícita o directa

En la interacción explícita o clásica, el usuario debe comunicar de forma explícita su voluntad al ordenador, amoldándose al esquema de comunicación determinado por la interfaz de la aplicación, ya sea basada en comandos o gráfica del tipo WIMP o GUI.

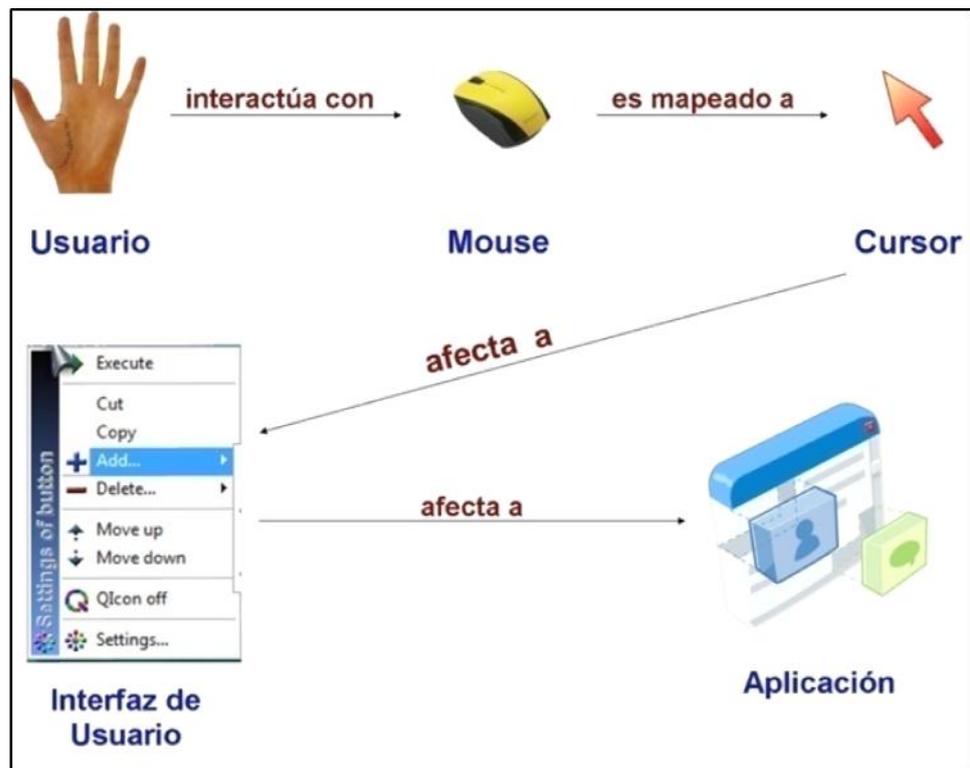


FIGURA 2.2: Diagrama de interacción explícita o clásica

En Realidad Virtual se utiliza la interacción implícita. A continuación detallaremos diversos dispositivos hápticos, que permiten este tipo de comunicación con el usuario.

GUANTES (DATA GLOVES)

Son nuestras manos virtuales, su función es permitirnos tocar, empujar y mover los objetos del mundo virtual.

Consisten de varios sensores que permiten capturar los movimientos de las manos, tales como la separación y flexión de los dedos; algunos cuentan con trackers que envía información sobre la

traslación y rotación de la mano. Según el menor o mayor número de sensores que tengan se obtendrá mayor información sobre los movimientos de la mano del usuario.

Típicamente han sido considerados dispositivos de entrada, pero en la actualidad existen guantes más sofisticados que permiten a la aplicación devolver estímulos de salida como vibración, solidez, rugosidad y temperatura.

MOUSE 3D

Se utilizan para la navegación y manipulación de objetos en un ambiente virtual, tiene seis grados de libertad, es decir permite la manipulación, traslación y rotación en el espacio tridimensional.

SISTEMAS DE POSICIONAMIENTO (TRACKERS)

Los trackers son también llamados dispositivos de 6 grados de libertad. Obtienen una posición, midiéndola en coordenadas (x, y, z) y el grado de giro en cada uno de los ejes denominados: **yaw**, **pitch** y **roll** respectivamente de acuerdo a un punto de referencia. En términos de hardware, se requiere una fuente que genere las señales, un sensor que reciba las señales y un componente de control que procese las señales y pueda comunicarse con el ordenador.

Se pueden encontrar de diferentes tipos: ópticos o infrarrojos, acústicos, magnéticos y mecánicos. Dependiendo de la tecnología, uno de los dos el sensor o el emisor pueden estar unidos al cuerpo y el otro debe permanecer estático, sirviendo como punto de referencia. Utilizan triangulación para obtener la posición, pueden ser utilizados, montándolos sobre los guantes, una estructura o algún otro dispositivo.

2.2. Ambientes inmersivos vs. no inmersivos

Uno de los aspectos que busca la Realidad Virtual es crear ambientes a través de medios tecnológicos que hagan sentir al usuario que se encuentran físicamente en la escena, esto es conocido como Inmersión.

Los sistemas de Realidad Virtual pueden ser clasificados como totalmente inmersivos, semi-inmersivos y no inmersivos. Esta clasificación depende de cuánto el usuario puede percibir del mundo real cuando se encuentra explorando un ambiente virtual.

2.2.1. Sistemas Inmersivos

Los ambientes virtuales inmersivos son espacios tridimensionales, reales o imaginarios, generados por computadora, con los que el usuario puede interactuar y que le producen la sensación de estar

dentro de un ambiente o lugar [18]. La sensación de presencia se genera cuando se integran varios elementos, como imágenes de alta calidad desplegadas en un área que cubra un amplio campo de visión del usuario y un buen sonido ambiental que complemente la escena. Para que la interacción en estos ambientes sea de la forma más natural posible, se recurre al uso de dispositivos especiales que nos permiten una manipulación natural o directa con el ambiente, como pueden ser el uso de guantes, sistemas de rastreo de movimiento, o cascos (HMD). El objetivo es aislar completamente al usuario de la realidad, para contribuir con la credibilidad de la simulación [25].

CASCOS DE INMERSIÓN (HMD)

En el uso de este dispositivo radica la diferencia entre un sistema inmersivo y no inmersivo. Por lo general estos dispositivos tienen en su interior dos pantallas de cristal líquido, una para cada ojo para poder visualizar la simulación del computador.

Existen dos tipos de cascos: los de 2D y 3D, la diferencia radica en la forma que son presentadas las imágenes. En los cascos 2D la imagen presentada es la misma tanto para el ojo derecho como para el izquierdo, en los 3D la imagen es distinta para cada ojo con el objetivo de crear profundidad para los objetos de la escena virtual.

Una de las desventajas de este dispositivo es que en algunas personas puede provocar la llamada “*cybersickness*”, que es una enfermedad provocada por la inmersión e incluye síntomas como náuseas, vértigo, ataxia y dolores cervicales.

CAVE

Se trata de una sala en forma de cubo rodeada por pantallas en las paredes, el suelo y techo, en donde es proyectado el mundo virtual. El usuario debe utilizar gafas especiales (estereoscópicas) para poder ver los gráficos 3D generados a su alrededor, algunas veces se hace uso de instrumentos sofisticados para proveer de más credibilidad a la escena, como caminadoras modificadas o esferas estacionarias, que le dan al usuario la sensación de poder caminar en cualquier dirección y trasladarse a través del ambiente simulado.

2.2.2. Sistemas No inmersivos

En la Realidad Virtual no inmersiva, el usuario interactúa en un ambiente representado en el monitor de una computadora, mediante el teclado, mouse o joystick. Es una experiencia tridimensional y multisensorial, pero de menor costo y más fácil de aceptar y utilizar [9].

VIDEO MAPPING

Este sistema utiliza la silueta de una persona mezclada con gráficas 2D generadas por computadora, el resultado final es que la persona a través de un monitor puede ver su cuerpo interactuando con un entorno virtual.

SISTEMAS 360°

Estos sistemas pueden utilizar la fotografía o el video para realizar paseos virtuales de un lugar, en los cuales se obtiene una visión de 360 grados de la imagen que se está presentando.

La interacción con el sistema se realiza mediante el Mouse o teclado del computador. El mundo puede ser manipulado de tal forma que se pueden hacer acercamientos a cada uno de los objetos o se pueden acceder a los diferentes sitios del mundo simulado.

WoW (WINDOW ON A WORLD)

A este sistema se le conoce también como realidad virtual de escritorio debido a que para visualizar el mundo generado se utiliza el monitor de computador. En estos sistemas

generalmente se utiliza como dispositivos de interacción y navegación, el teclado y el mouse del computador, pero también pueden ser utilizados guantes o mouse 3D.

VRML (Virtual Reality Modeling Language)

Es un lenguaje para modelar mundos virtuales a los que podemos acceder a través de cualquier navegador, como si visitáramos una página web cualquiera, con la diferencia que además de texto y fotografías, nos permite visualizar objetos y lugares con los que podemos interactuar [6].

2.2.3. Sistemas Semi-inmersivos

Los sistemas semi-inmersivos son aquellos en donde el usuario a pesar de poder interactuar con el ambiente virtual, es capaz de captar cualquier estímulo del mundo real, es decir carecen de la sensación de presencia.

Sistemas de telepresencia

Esta tecnología vincula sensores remotos, instalados en un robot, con los sentidos de un operador humano, que puede operarlo como si fuera parte de su cuerpo. Los sistemas de telepresencia sitúan al observador en un mundo real que es capturado por

videocámaras ubicadas en lugares distantes y le permiten la manipulación remota de objetos reales mediante brazos robotizados. Es decir que el usuario interactúa en un mundo real pero que está ubicado en un sitio distinto a donde él se encuentra en ese momento [13].

2.2.4. Ventajas y desventajas de los sistemas inmersivos y no inmersivos.

La Realidad Virtual no Inmersiva utiliza medios con los que ya estamos comúnmente familiarizados, como el monitor de una computadora y la Internet, en el cual podemos interactuar con varias personas en tiempo real sin la necesidad de dispositivos adicionales [6]. Esto puede ser tan ágil e interactivo como un método inmersivo, pero con ciertas ventajas:

- Los dispositivos inmersivos son de alto costo y algunas veces son más difíciles de manipular y calibrar, por lo cual muchas veces el usuario prefiere manipular el mundo virtual a través de dispositivos ya conocidos, como el teclado y el mouse.
- El rápido desarrollo de la tecnología de los procesadores ha permitido que la animación tridimensional interactiva en

tiempo real funcione sin problemas, con lo cual podemos experimentar la virtualidad de forma más accesible.

- Actualmente el Internet gracias a VRML, nos ofrece las herramientas y facilidades para interactuar con varias personas y objetos en el mismo mundo virtual, lo cual ha beneficiado a los sistemas no inmersivos [6].

Por el contrario los sistemas inmersivos permiten:

- Libertad y amplitud de movimiento dentro de la escena simulada.
- Generación de sensaciones provocadas por el sonido espacial y la retroalimentación táctil.
- Existe mayor detalle en la visualización de los ambientes.
- Nos permite acceder a espacios de alto riesgo y recrear ambientes para entrenamiento que serían muy costosos y poder modificar los eventos que ahí ocurren.

2.3. Descripción de algunos sistemas de RV existentes

Las técnicas de Realidad Virtual en los últimos años han sido aprovechadas por distintas ramas del conocimiento, para crear proyectos que implicaban alto riesgo monetario y en algunos casos riesgo humano. A lo largo de nuestra investigación hemos encontrado algunas aplicaciones que actualmente se utilizan con mucho éxito, entre los cuales tenemos:

MEDICINA

SURGEON TRAINER: Desarrollado por el Estudio de Diseño Digital de la Escuela de Arte de Glasgow y el Real Colegio de Cirujanos de Edimburgo (Escocia) ofrece a los especialistas una oportunidad para entrenarse en el arte de la cirugía y mejorar su conocimiento de las estructuras del cuerpo humano [14]. El sistema utiliza unos guantes que permiten a los cirujanos tocar y sentir los tejidos blandos y duros y explorarlos virtualmente.

Los estudiantes pueden practicar distintas intervenciones y aumentar gradualmente el nivel de dificultad de las mismas. Además, el software puede programarse para que surjan una serie de imprevistos como, por ejemplo, que la salud del paciente empeore de repente, y si se comete un error no se pone en riesgo la salud de ningún paciente.

CATH SIM ACCU TOUCH: La Facultad de Medicina y Odontología de la Universidad del País Vasco/Euskal Herriko Unibertsitatea, ha instalado un simulador no inmersivo de realidad virtual que permite entrenar a los alumnos en todos los aspectos de la venopunción, desde las habilidades requeridas para practicarla, hasta los conocimientos sobre la terapia [7]. Así, tienen que escoger en el vídeo el tamaño apropiado de aguja, usar la técnica antiséptica adecuada y pinchar la vena en el ángulo preciso, utilizando el aparato de inyección conectado al ordenador. Al introducir la aguja, pueden notar una resistencia muy similar a la percibida en pacientes reales. Y si aquélla no ha sido introducida de forma correcta, el paciente virtual manifiesta de forma audible el dolor.

MIRACLE: El Grupo de Investigación LabHuman de la Universidad Politécnica de Valencia [15], en colaboración con el Instituto de Biomecánica de Valencia y la Facultad de Odontología de la misma Universidad, ha desarrollado un proyecto para el diseño personalizado de prótesis e implantes dentales, basado en un nuevo sistema inteligente de realidad virtual no inmersiva, que permitirá reducir plazos y costes de fabricación, ya que se podrá realizar una valoración completa del tratamiento final en el paciente sin necesidad de fabricar prótesis físicas.

El proyecto se compone de un sistema de diseño inteligente de la anatomía maxilofacial y dental, una serie de sensores ópticos que registrarán los patrones de masticación del paciente, y nuevas técnicas con las que se podrán diseñar los implantes a la carta.

SALUD

ACTIVEPLAY: Desarrollado por un grupo de investigadores de LabHuman [15], perteneciente al Instituto de Investigación e Innovación de Bioingeniería (I3B) de la Universidad Politécnica de Valencia, su objetivo es crear un juego interactivo en entornos virtuales para evitar la obesidad infantil y juvenil. En estos momentos ya se ha desarrollado un prototipo integrado con un escenario virtual, consistente en un simulador de vuelo en 3D y de una plataforma deportiva llena de sensores, que registra los movimientos del usuario y los traslada al simulador, lo que permitirá al participante "vivir" la sensación de caída libre en tiempo real de forma individual o colectivamente.

INDUSTRIA DEL CALZADO

SIMUCAL: Científicos del grupo LabHuman [15], perteneciente al Instituto de Investigación e Innovación en Bioingeniería de la Universidad Politécnica de Valencia (UPV), junto con el Instituto de Biomecánica de Valencia (IBV), han desarrollado un simulador de Realidad Virtual para el sector del calzado, que permite medir la comodidad y funcionalidad de un nuevo diseño antes de fabricarlo, y hacerlo además de forma rápida y sencilla.

Con este simulador, primero se escanea el pie del cliente y a partir de la imagen obtenida, se prueban y evalúan en el ordenador diferentes diseños virtuales. En la pantalla el fabricante puede conocer al instante la respuesta del zapato, cómo se deforma y dónde podría rozar o apretar más a quien se lo calce.

AERONAÚTICA

PARASIM: Es un simulador de realidad virtual inmersivo que permite recrear el vuelo con un paracaídas de cualquier tipo y en diferentes condiciones atmosféricas, desde la salida del avión hasta la toma de tierra. Además recrea posibles situaciones de emergencia que permite a los usuarios enfrentarlas con seguridad.

Este sistema permite la práctica del paracaidismo eliminando los tiempos de espera, los costos de operación y los riesgos del entrenamiento real. Consta de un soporte que tiene un arnés similar a la que utilizan los paracaidistas y las tiras de mando para poder controlarlo, además el usuario utiliza un HMD y audífonos para poder percibir el ambiente de entrenamiento virtual [23].

AUTOMOTRIZ

SIMBIO: Desarrollado por la empresa española LANDER [28], es la última tecnología virtual destinada al reciclaje y a la formación de conductores de camiones y autobuses

La cabina de entrenamiento dispone de un pupitre, mandos y asientos reales, con opción de movimiento y pantallas de ángulo de visión de 180° es la nueva aula de formación de los conductores profesionales.

SIMULADOR DE CONDUCCIÓN DE TOYOTA: Es el más moderno y más grande simulador que exista. Con 7,1 metros de diámetro, la esfera es capaz de albergar en su interior a cualquiera de los modelos de la marca.[29]

Además todo el lado interno de la esfera es una gigantesca pantalla de 360 grados, lo que junto al sonido y la simulación de vibraciones y golpes ayuda a otorgar un realismo casi total.

El simulador tiene precargados todos los datos de cada uno de los modelos de Toyota, y se analiza la fiabilidad de cada uno de ellos en distintos ambientes, lo cual permitirá desarrollar coches con mejor seguridad, y con altas expectativas de salvamento de los ocupantes ante la inminencia de un accidente, o en condiciones de manejo extremas.

ENTRETENIMIENTO

SECOND LIFE: Desarrollado por LINDEN RESEARCH INC., es un mundo virtual construido con una avanzada técnica tridimensional y habitado por personajes 3D configurables (avatars), lo que le da a los usuarios la capacidad de convertirse en otra persona y gozar de una segunda vida. Esto promueve en el mismo mundo una avanzada interacción entre los residentes, podrán explorar el mundo, conocer a otras personas, socializarse, participar en actividades grupales de acuerdo a sus gustos, entre otras cosas desde la pantalla de su computador a través de un navegador. [26]

Su segundo atractivo más importante es la posibilidad de crear objetos e intercambiar diversidad de productos virtuales a través de un mercado abierto que tiene como moneda local el Linden Dólar (\$L).

La programación de este mundo virtual es abierta y libre. El código de SL permite a los usuarios poder modificar absolutamente cualquier aspecto del mundo virtual, desde el color de los ojos del personaje a su aspecto físico, sus movimientos, sonidos y permite además, construir cualquier cosa en 3D.

2.4. Descripción de la aplicación no Inmersiva *BlockCAD*

BlockCAD fue creado en 1998 por el programador sueco Anders Isaksson, de esta forma nació un programa de CAD que permite la construcción en 3D de modelos empleando bloques tipo LEGO. Es una aplicación gratuita y de fácil uso que permite crear edificaciones, vehículos y otros diseños complejos en 3D, los cuales se pueden girar y observar desde diferentes perspectivas y guardarlos en formato jpg y bmp. El ambiente es controlado con el mouse o el teclado[4].

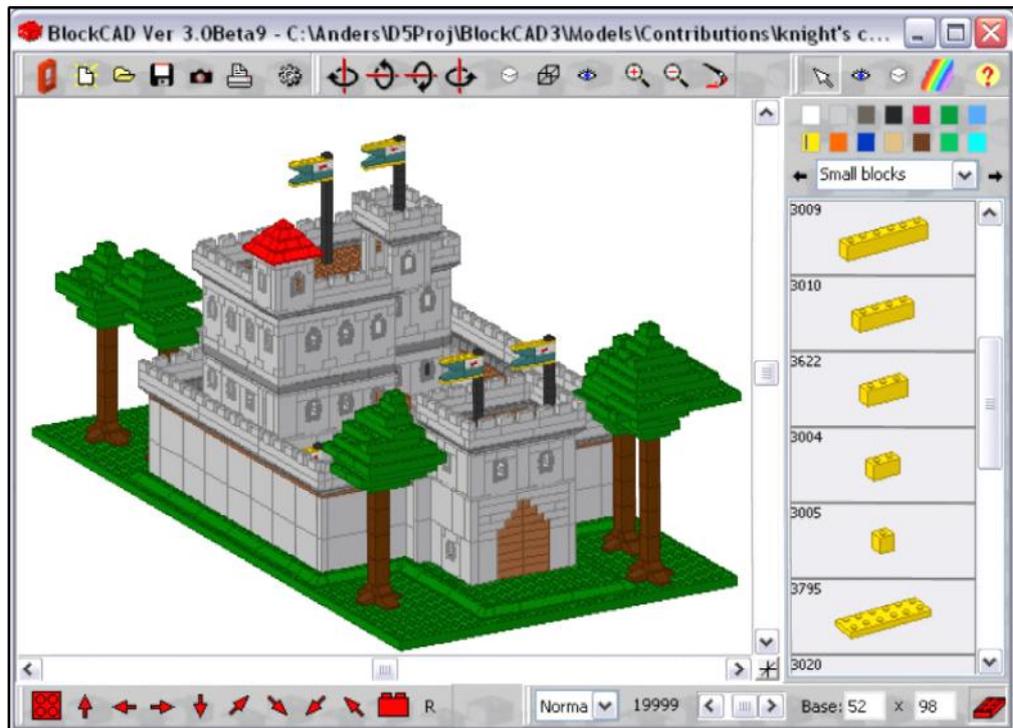


FIGURA 2.3: Imagen del editor gráfico del BlockCAD

El editor de proyectos es muy completo y posee un gran número de piezas agrupadas en varias categorías, sin embargo solo permite juntar las piezas sin verificar si la posición es válida, es decir uno puede colocar una pieza arriba sin tener una debajo.

CAPÍTULO 3

3. Análisis y diseño de la Aplicación

3.1. Análisis de la aplicación

Con este proyecto queremos explorar el campo de la Realidad Virtual usando sus principios y herramientas para crear una aplicación semi-inmersiva del juego de bloques de Lego.

3.1.1. Requisitos funcionales

Con nuestro proyecto buscamos experimentar con el uso de nuevas tecnologías en el campo de la computación y los gráficos multimedia, para crear una aplicación virtual que será manipulada mediante dispositivos poco comunes de Realidad Virtual tales como los guantes, gafas estereoscópicas y trackers, con la finalidad de guiar a futuros desarrolladores en la correcta calibración y uso de éstos dispositivos.

3.1.2. Requisitos no funcionales

La aplicación tiene que ser llamativa, intuitiva y de fácil uso, tanto para usuarios familiarizados con este tipo de tecnología como inexpertos. La creación de los objetos y entornos en 3D se realizan con el programa de modelamiento de gráficos por

computadora 3D Studio Max 9. Como motor de gráficos 3D y medio de interacción entre el usuario y la aplicación se utiliza la herramienta multiplataforma OpenSceneGraph (Código abierto) [21].

3.1.3. Usuarios de la aplicación

La aplicación será probada por usuarios que estén familiarizados con el uso de los dispositivos de Realidad Virtual que vamos a utilizar en nuestra aplicación, como el guante, los trackers y las gafas estereoscópicas.

3.2. Diseño de la aplicación e interacción

La aplicación se fundamenta en tres partes: la primera el modelamiento de cada objeto (los bloques de Lego) en un programa de creación de gráficos 3D, la segunda parte en la elaboración del entorno en donde se van a ser cargados cada uno de estos bloques e integrados a la escena, y la tercera la calibración de los dispositivos que el usuario va a utilizar para interactuar con la aplicación.

3.2.1. Diseño de los objetos 3D

Los bloques de Lego que se presentan en la aplicación fueron modelados en el programa para gráficos 3D Studio Max.

Se empezó diseñando las figuras básicas que íbamos a necesitar con las medidas reales establecidas mundialmente por Lego para todos los bloques incluidos en sus juegos [16]:

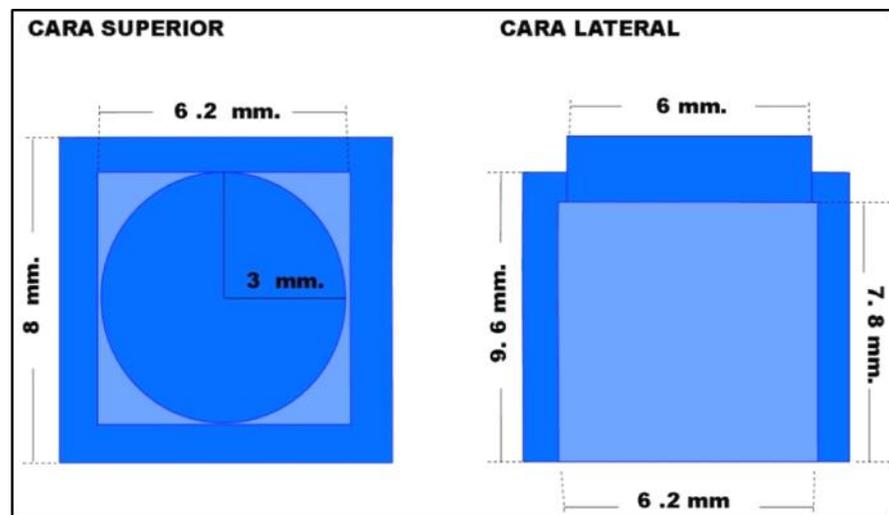


FIGURA 3.1: Dimensiones guía para la creación de un bloque.

En el modelado de cada bloque recurrimos a operaciones booleanas para darle la forma deseada, respetando los criterios de diseño que habíamos establecido previamente. Por medio del manejo de colores y texturas, se trató de representar las características físicas, para añadir realismo a los objetos.

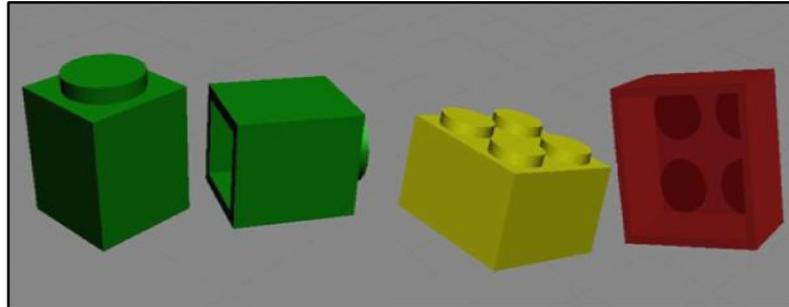


FIGURA 3.2: Bloques de Lego modelados en 3D Studio Max

Una vez que habíamos finalizado cada objeto se procedió a convertirlo al formato de imagen adecuado (*osg*) para poder importarlo desde la plataforma con la que trabajamos sin que se presenten problemas de datos corruptos o pérdida de información gráfica importante tal como el color o la textura.

Acto seguido continuamos con la segunda parte del diseño, donde se creó la interfaz donde están situadas las piezas para que puedan ser manipuladas virtualmente.

3.2.2. Análisis de colisión de objetos

Cuando se está agregando o moviendo una pieza dentro del tablero puede que la nueva posición este ocupada, por lo cual definimos las siguientes combinaciones en un grafo de opciones a tomar para resolver el problema, el cual se muestran en la

FIGURA 3.3, para lo cual definimos los siguientes nodos:

ML -> Movimiento a la izquierda (Eje -y)

MF -> Movimiento hacia el frente (Eje +z)

MB -> Movimiento hacia atrás (Eje -z)

MR -> Movimiento hacia la derecha (Eje +y)

I -> Punto es libre

F -> Agregar pieza

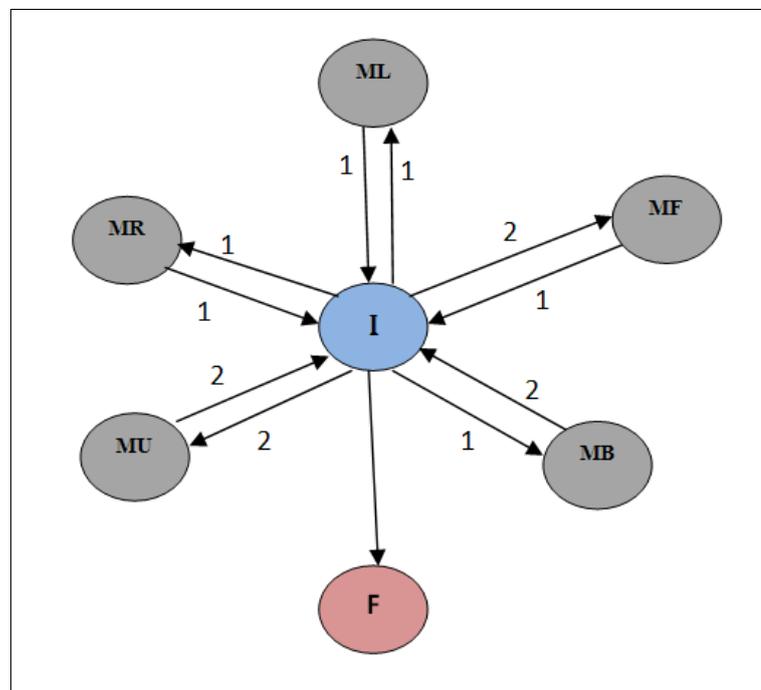


FIGURA 3.3: Grafo de opciones para evitar colisiones

Entonces creamos 2 caminos diferentes para llegar a una posición libre, el primer camino se crea tomando un nodo al azar y repetirlo un número N de veces hasta llegar al estado final F, el

segundo camino se crea anexando nodos aleatorios hasta llegar al estado final F.

Una vez creado los 2 caminos comparamos los costos de cada uno de ellos en llegar al estado final F, y seleccionamos el de menor costo.

Este procedimiento siempre llega a un estado F, es decir a una posición libre donde se puede colocar la nueva pieza de lego.

3.2.3. Interfaz orientada a una aplicación semi-inmersiva

La interacción con la herramienta debe ser lo más transparente posible para el usuario, al ser una aplicación semi - inmersiva se debe tener mayor libertad de movimiento y campo de visibilidad que una aplicación común, en la que nos encontramos frente a un monitor.

Dentro del ambiente virtual se puede acceder a un área de trabajo donde el usuario puede diseñar y construir cualquier objeto que desee, haciendo uso de los bloques de Lego, que pueden ser manipulados mediante determinados dispositivos de realidad virtual, de los cuales hablaremos más adelante, de manera muy precisa y natural. Además se da la posibilidad al

usuario de almacenar la escena creada para modificarla posteriormente.

3.2.4. Especificación de los dispositivos de entrada y salida que utilizaremos para la interacción.

Para poder visualizar y manipular los objetos el usuario está equipado con tres dispositivos virtuales indispensables:

- Un par de gafas estereoscópicas, que son las encargadas de convertir las imágenes proyectadas en objetos en 3D, y contribuyen a la inmersión del sujeto en el entorno virtual. En el laboratorio de Realidad Virtual existen dos tipos de gafas: las activas o LCD que tienen un transmisor de sincronización, que se conecta a la tarjeta de video y envía señales en tres dimensiones (120 Hz.) al proyector. Las gafas pasivas están formadas por lentes anaglíficas de colores rojo y azul generalmente, que filtran la imagen proyectada y crean el efecto de profundidad [20].
- Un guante de datos, que permite manipular los objetos de la escena, por medio de la realización de gestos que simulan el movimiento real de la mano y que son capturados por medio de sensores de flexión. Los guantes que utilizamos en este

proyecto cuentan con 14 sensores, dos por cada dedo y uno por cada separación entre los dedos [2].

- Los trackers para poder capturar los movimientos de traslación de la mano del usuario al momento de interactuar con la aplicación, debido a que el tipo de guante con el que trabajamos, no tiene este tipo de sensor montado en su estructura. Los trackers que utilizamos son de tipo magnético, un inconveniente de este tipo de dispositivos es que tienen que estar aislados de cualquier objeto metálico, ya que esto puede distorsionar el campo magnético lo que se traduce en lecturas de posicionamiento erróneas [24].

3.3. Arquitectura de la aplicación

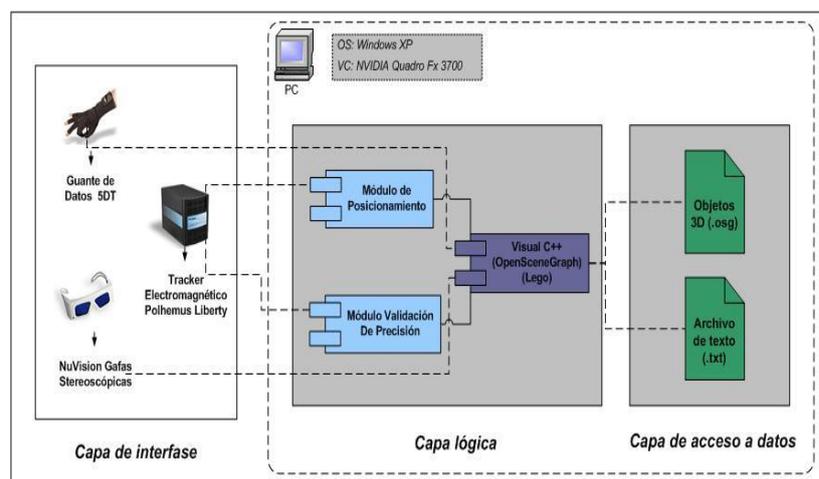


FIGURA 3.4: Diagrama general del Sistema – Arquitectura

Una de las partes más importantes del sistema recae sobre el módulo de posicionamiento y el módulo de validación de precisión **FIGURA 3.4**, los cuales trabajaban conjuntamente para que el usuario pueda sentir el realismo de estar moviendo los objetos en el mundo virtual.

3.3.1. Módulo de Posicionamiento

El módulo de posicionamiento se encarga de tomar las coordenadas (x, y z) que le corresponden a una pieza de lego en el Tablero, para lo cual se realizan los siguientes pasos:

1. Recibir las coordenadas libre de errores del módulo de validación
2. Aplicar la heurística de posicionamiento
3. Graficar el Lego en la posición libre que se obtiene al aplicar la heurística de posicionamiento.

Este módulo es uno de los que más utilizados ya que es llamado cada vez que se agrega, se mueve o se cambia de color una pieza en el tablero, para lo cual se recibe las coordenadas correspondientes en el espacio OSG.

Una vez que se recibe las coordenadas, aplicamos el análisis de colisión de objetos basado en el GRAFO definido y explicado en la

sección 3.2.2, el cual se encarga de seleccionar la posición libre óptima según el grafo y dibuja la pieza de lego en la escena.

3.3.2. Módulo de Validación de Precisión

Se fundamenta en tres procesos:

- I. Tomar los valores (x , y, z) capturados por el tracker
- II. Aplicar la corrección a los valores obtenidos por el tracker.
- III. Aplicar las transformaciones de coordenadas

En el transcurso del juego los movimientos que se realizan tienen que corregirse para obtener la precisión deseada del tracker, éste análisis se detallará más ampliamente en el capítulo 4. Los ejes coordenados entre el tracker y el OSG son diferentes, por lo cual es necesario hacer una transformación al vector que se obtiene del tracker para igualarlos, lo cual da como resultado la siguiente ecuación:

$$V_{osg} = \{8*[2*(Y_t - 9), 9.2 + [9.2* X_t], -8*[4*Z_t]]\}$$

3.4. Diseño lógico

3.4.1. Diagramas UML

La FIGURA 3.5 describe las acciones que el usuario puede realizar dentro de la aplicación virtual.

Diagrama de Casos de uso

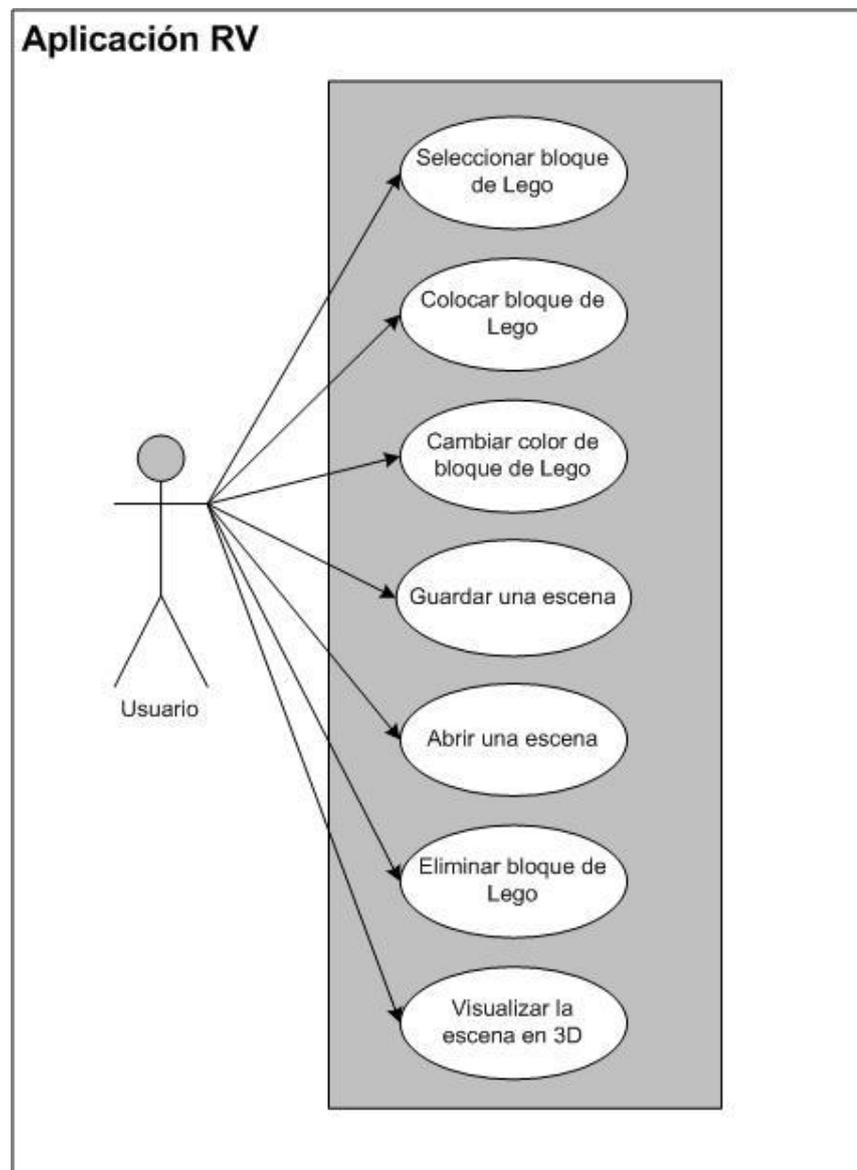


FIGURA 3.5: Caso de uso de la aplicación

3.4.2. Diagramas de secuencia (Escenarios)

Seleccionar bloque de Lego

Selección exitosa del bloque de Lego, el usuario puede escoger entre las piezas que están disponibles en el contenedor.

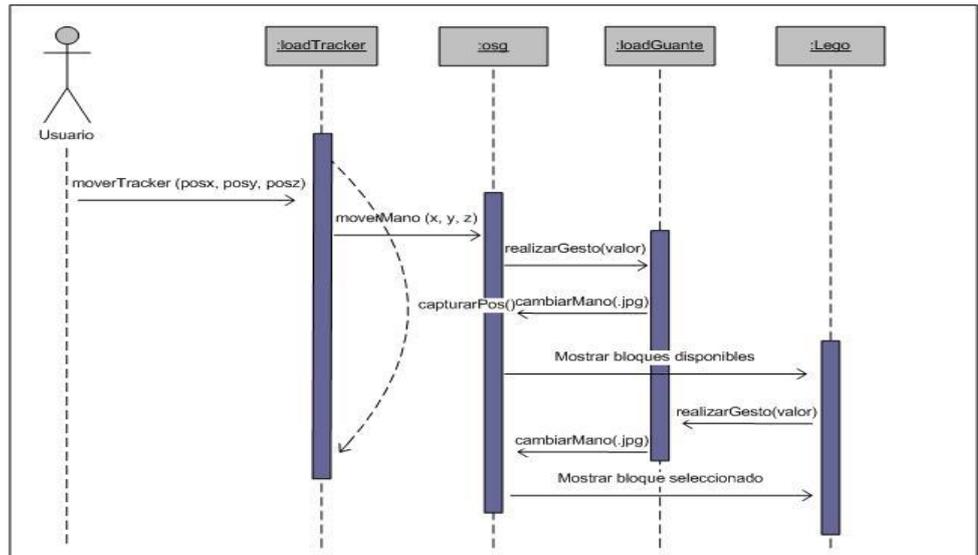


FIGURA 3.6: Diagrama de secuencia – Seleccionar bloque Lego

Colocar bloque de Lego

Ubicación del bloque en el área de trabajo con éxito, el usuario coloca el bloque en la posición deseada.

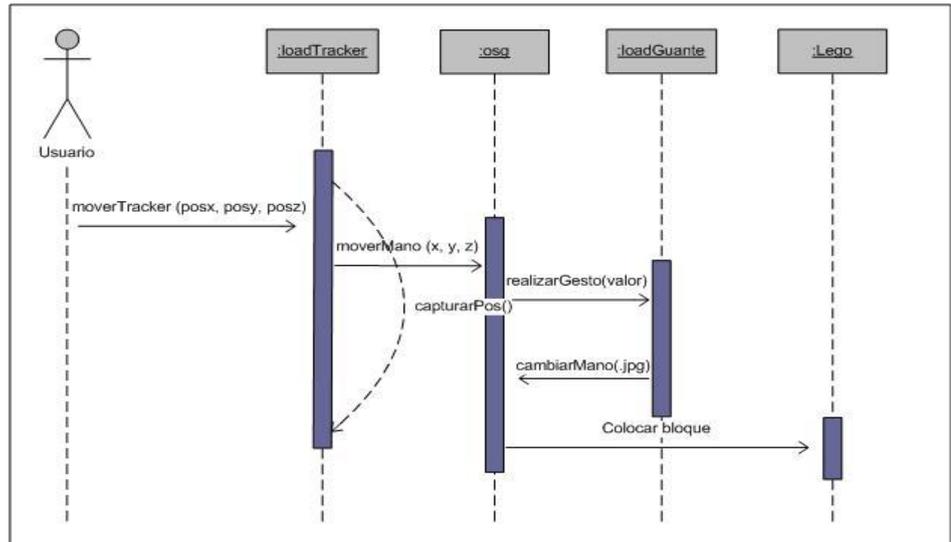


FIGURA 3.7: Diagrama de secuencia – Colocar bloque de Lego

Cambiar color de bloque de Lego

Cambio del color de bloque exitoso, el usuario puede seleccionar entre los distintos botes de pintura que se le presentan.

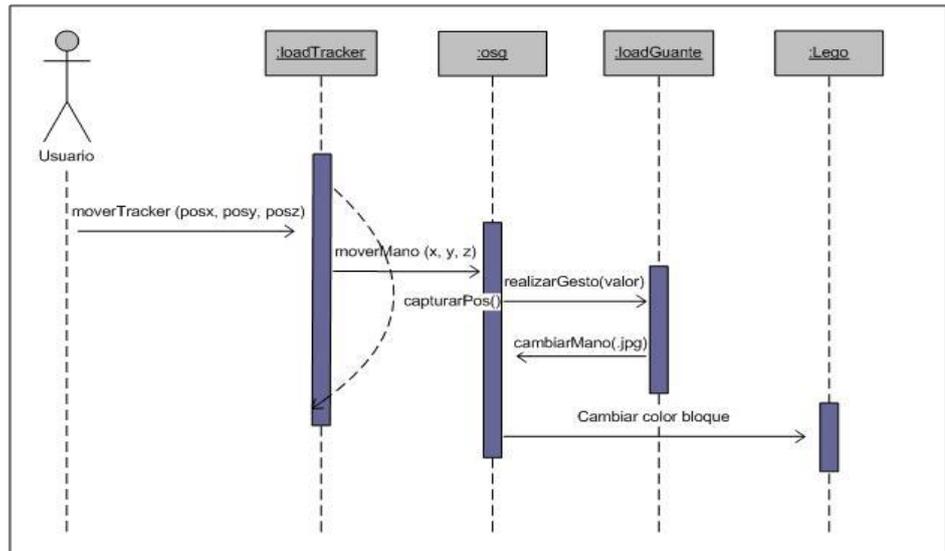


FIGURA 3.8: Diagrama de secuencia – Cambiar color de bloque

3.4.3. Diseño de algoritmos

Dentro de la interfaz virtual se permite al usuario manipular los bloques de Lego de acuerdo a la escena u objeto que desea crear, para esto fue necesario crear ciertos algoritmos que validan la correcta ubicación dentro del área de trabajo.

Seleccionar Nodo (x_1, y_1, z_1)

Recibir punto (x_1, y_1, z_1) desde dispositivo.

Recorrer lista de Nodos disponibles y extraer el que se encuentre entre las coordenadas indicadas.

Retornar el nodo seleccionado.

Color al Nodo (color i)

Recibir índice de la paleta seleccionada.

Recorrer la paleta de colores.

Tomar el atributo color de la paleta seleccionada

Retornar el color.

Detectar si el punto está vacío (x_1, y_1, z_1)

Recibir el punto (x_1, y_1, z_1) en el cual quiere agregarse el nodo.

Recorrer la LISTA de nodos existentes.

Si $x1 < LISTA[index].ancho + LISTA[INDEX].pos_en_x$ es

menor

Y $y1 < LISTA[index].alto + LISTA[INDEX].pos_en_y$ es menor

Y $z1 < LISTA[index].profundidad + LISTA[INDEX].pos_en_z$.

Retornar index

Caso contrario retornar un valor negativo.

Agregar un nuevo nodo al tablero (x1, y1, z1)

Preguntar *si el punto es vacio* en (x1, y1, z1).

Si se retornó un valor negativo, el punto está libre.

Incrementamos NUMERO_TOTAL_NODOS.

Leer nodo seleccionado.

Cargamos los atributos de ancho, altura y profundidad del nodo
seleccionado,

Asignamos la rotación del Nodo.

Asignamos el *color al nodo*.

Asignamos la nueva posición (x1, y1, z1).

3.5. Interfaz de la aplicación

Con el propósito de facilitar al usuario la interacción dentro de la aplicación virtual, se crearon gráficos 3D representativos de la acción que se puede realizar.

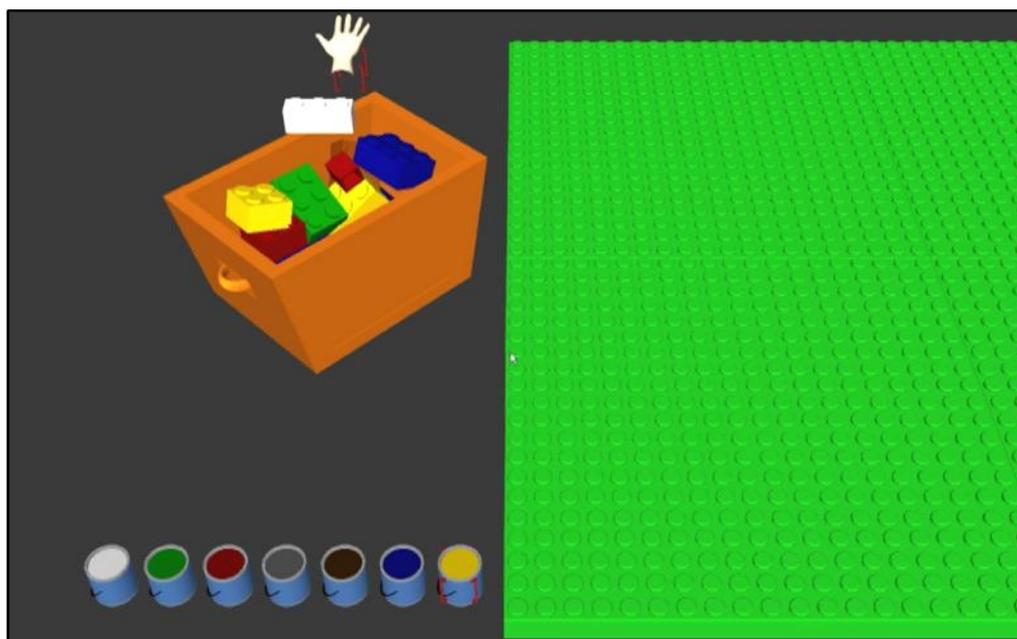


FIGURA 3.9: Imagen del contenedor de bloques de Lego.

La **FIGURA 3.9**, nos muestra el contenedor de bloques de Lego, donde el usuario al colocar sobre este la mano virtual con ayuda del sensor de posición del tracker, puede escoger entre las distintas piezas que aparecen sobre el contenedor simultáneamente cuando el usuario realiza el gesto de selección con el guante de datos, y se detiene cuando se elige el bloque deseado.

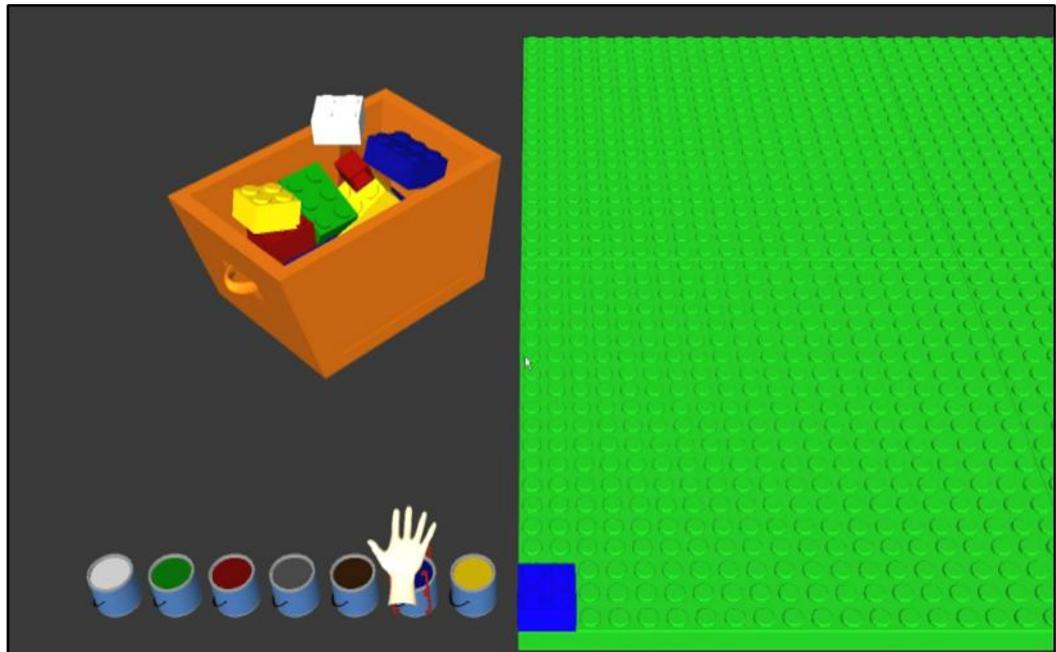


FIGURA 3.10: Cambiar el color de un bloque de Lego

La **FIGURA 3.10**, nos muestra como el usuario puede cambiar el color de una pieza ubicada y seleccionada en el área de trabajo. Para realizar esta acción, se debe ubicar la mano virtual sobre el bote de pintura del color deseado, y realizar con el guante de datos el gesto de selección.

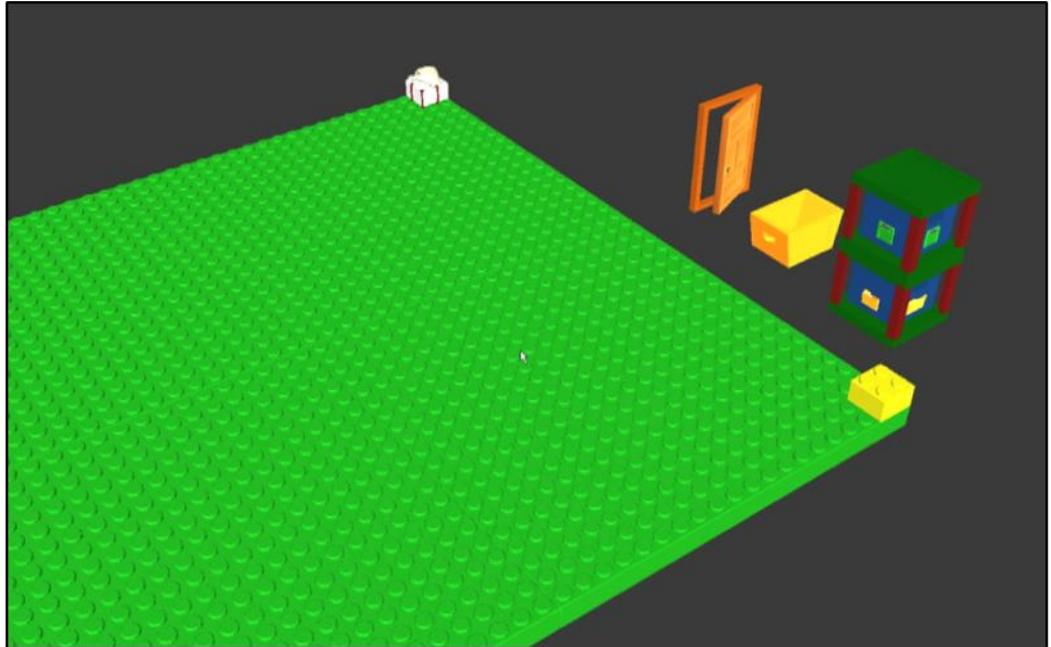


FIGURA 3.11: Calibración de ubicación de los bloques

La **FIGURA 3.11**, es un ejemplo de la calibración inicial que se realiza dentro de la aplicación. En éste proceso se ubican cuatro bloques en cada uno de los extremos para obtener los valores de longitud y profundidad del área de trabajo, con lo cual se arma la matriz de posiciones válidas donde pueden ser colocados los bloques.

En la **FIGURA 3.11**, también podemos apreciar tres objetos más que forman parte de nuestra barra de herramientas: la cajonera, donde al tocar el primer cajón podemos guardar un objeto creado en nuestra aplicación, y el segundo cajón que nos permite abrir un objeto creado con anterioridad.

El contenedor vacío permite al usuario limpiar el área de trabajo de los bloques que han sido colocados sin guardarlos; y la puerta que nos permite abandonar la aplicación virtual.

Para poder colocar un bloque sobre otro, se corre un algoritmo de verificación, que determina si una posición está ocupada por un bloque de acuerdo al volumen que ocupa dentro de la escena (alto, ancho y profundidad).

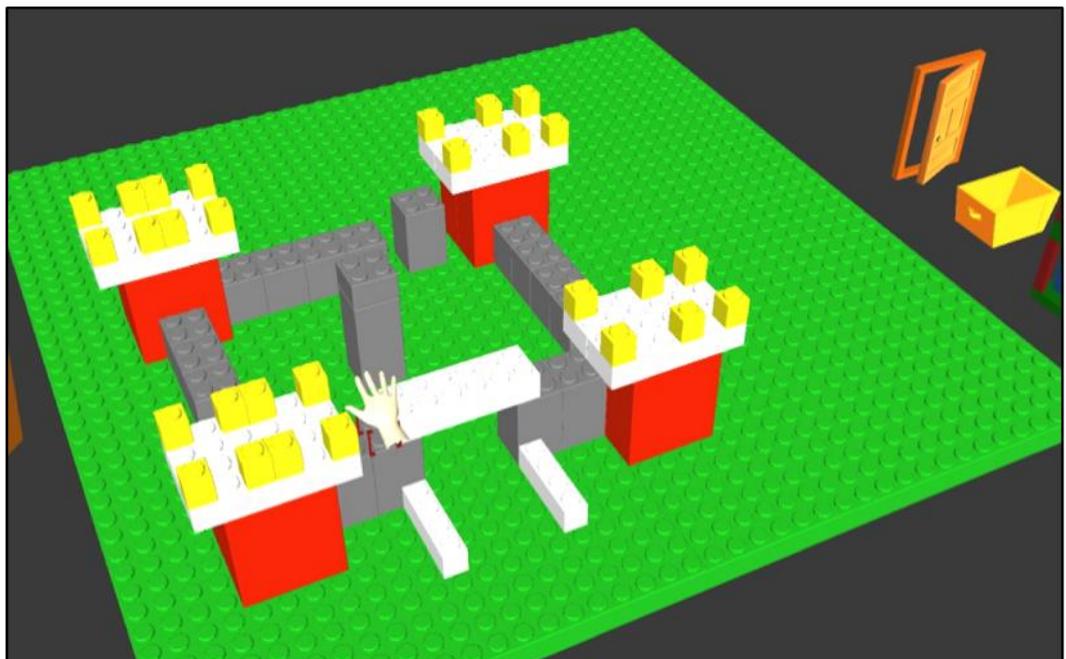


FIGURA 3.12: Castillo creado con la aplicación Lego 3D

3.6. Diseño de pruebas

El plan de pruebas que hemos diseñado, permitió obtener datos confiables con respecto a los errores y omisiones de cada versión de la aplicación. Fueron realizadas por los miembros del grupo de trabajo, debido a que nuestro enfoque no era el de comparar dos productos para obtener las ventajas de uno sobre el otro, sino tratar de crear una aplicación virtual, lo más realista posible, de un juego físico.

Específicamente se realizaron pruebas para medir la precisión y facilidad de uso de los equipos de interacción usuario – aplicación, como los trackers y el guante de datos.

Se verificaron los siguientes puntos:

- Naturalidad de gestos realizados con el guante para seleccionar, agarrar y soltar un bloque en el área de trabajo.
- La ejecución correcta del picking de los objetos.
- Precisión en la ubicación de los bloques, comprobar si se pueden colocar en cualquier lugar del área de trabajo y no fuera de ella.
- El correcto funcionamiento de los algoritmos encargados de controlar la lógica en la ubicación de los bloques en la escena, evitar que se traslapen, que se coloquen en la misma posición de una pieza anteriormente colocada, etc.

- Comprobación de almacenamiento y carga correctamente una escena.
- Comprobación que la aplicación no se interrumpa por algún error en el tiempo de ejecución.

Todas estas pruebas se realizaron con el propósito de corregir errores y omisiones en la implementación de la aplicación. En el siguiente capítulo analizaremos detalladamente los datos obtenidos.

CAPÍTULO 4

4. Implementación y Pruebas

4.1. Software utilizado para la implementación

Para el desarrollo de este proyecto se utilizó un programa de edición de gráficos para modelar las imágenes, las cuales fueron agregadas en una interfaz creada en un entorno de desarrollo integrado bajo el lenguaje de programación C++. Ambos son programas con costo comercial, pero como el proyecto tiene fines netamente educativos no existen problemas de distribución de la licencia.

4.1.1. Modelado de objetos y ambientes 3D

Los objetos y ambientes utilizados fueron modelados con el programa de creación de gráficos y animación 3d Studio Max 9. Este editor es uno de los más utilizados y reconocidos en campos como los videojuegos, animación de películas y efectos especiales, ya que dispone de una alta capacidad de edición y plugins. Actualmente se encuentra liberada la versión 2010 para Windows Xp o Vista de 32 o 64 bits.

Las imágenes creadas fueron exportadas a formato *osg* mediante el uso de un plugin, ya que de esta forma podían ser

importadas sin problemas de pérdida de información y utilizadas en la plataforma y con las librerías con las que íbamos a trabajar la interface 3D.

4.1.2. Plataforma y librerías de programación

La implementación de la aplicación inmersiva fue realizada en el ambiente de desarrollo Visual Studio 2008 en el lenguaje de programación C++.

Para el manejo de gráficos y herramientas 3D se recurrió a la librería de código abierto OpenSceneGraph en su versión 2.8.1.

4.1.3. Audio

Simple DirectMedia Layer (SDL) es un conjunto de bibliotecas desarrolladas con el lenguaje C que proporcionan funciones básicas para realizar operaciones de dibujado 2D, gestión de efectos de sonido y música, y carga y gestión de imágenes.

Debido a que es posible usar SDL dentro de OSG y que las librerías de SDL nos dan una facilidad al momento de reproducir un sonido en formato “wav”.

4.2. Dispositivos de Hardware

Al trabajar en una aplicación de estas características donde se incluyen objetos que son modelados en un editor como el 3D Studio Max, se debió utilizar un equipo de computación con gran capacidad de procesamiento y que cuente con una tarjeta gráfica de alta definición y que permita la visualización de este tipo de gráficos, es por esto que para las pruebas se utilizó una estación de trabajo Dell Intel Xeon de 2.82 GHz, con 4 GB de memoria RAM, y una tarjeta NVIDIA Quadro FX 3700.

Adicionalmente para la interacción entre el usuario y la interfaz virtual se utilizaron los siguientes dispositivos de entrada – salida:

- 5DT Data Gloves.- Son los guantes que mediante la captura de gestos permite al usuario manipular los objetos 3D en el entorno virtual. En nuestra aplicación utilizamos un guante derecho para poder sensar las acciones de selección y ubicación de un bloque de Lego en la escena. Para esto determinamos un rango de valores obtenidos de las mediciones de los sensores de los dedos al realizar este tipo de movimientos en el mundo real, para así permitir que el usuario realice desplazamientos no forzados y más naturales.



FIGURA 4.1: Guante de datos 5DT de 14 sensores

- Polhemus Motion Tracking (LIBERTY).- Son trackers alámbricos de tipo electromagnético, que ofrecen 6 grados de libertad de movimiento y tienen una tasa de refrescamiento de 240 hertzios por cada sensor [24].

En nuestra aplicación vamos a utilizar 2 sensores, el primero estará montado a la estructura del guante para poder capturar el movimiento de la mano al interactuar con los objetos virtuales. El segundo se colocará en el torso del usuario para poder sensor el movimiento de éste alrededor de la escena.

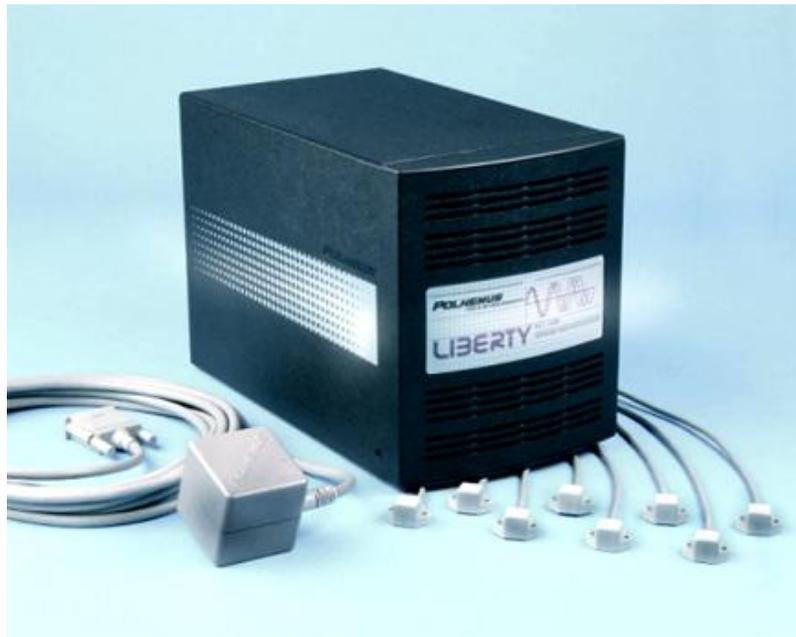


FIGURA 4.2: Tracker electromagnético Polhemus

- NuVision wireless stereoscopic glasses.- Estas gafas permiten al usuario visualizar en tres dimensiones las imágenes anaglíficas de la aplicación que son proyectadas en la pantalla.



FIGURA 4.3: Gafas estereoscópicas NuVision 60 GX

- DepthQ Projector.- Es un proyector estereoscópico, que permite visualizar imágenes en 3D utilizando una sola lente. Ofrece una resolución de pantalla de 1280 x 720 píxeles y luminosidad de 2000 lúmenes [8].



FIGURA 4.4: Proyector estereoscópico 3D DepthQ

4.3. Análisis y calibración de los trackers

Como se mencionó anteriormente uno de los objetivos fundamentales de este proyecto, radica en que el usuario sea capaz de utilizar los equipos de manera fácil y precisa, ya que constituyen la herramienta principal de interacción con la aplicación. Entre todos ellos, el que requirió más esfuerzo y horas de trabajo para lograr calibrarlo adecuadamente, fueron los trackers electromagnéticos.

4.3.1. Optimización del control y precisión de los trackers.

Los trackers electromagnéticos son muy populares, pero lamentablemente son inexactos. Sufren de problemas de distorsión de datos, provocados por grandes cantidades de metal presentes en el área de trabajo o por otros campos electromagnéticos que interfiere su señal, como el producido por los equipos de cómputo. Adicionalmente el sensor debe encontrarse dentro de un rango restringido desde la fuente o no será capaz de captar información precisa, introduciendo errores en la adquisición de datos, que perjudican el performance de la aplicación.

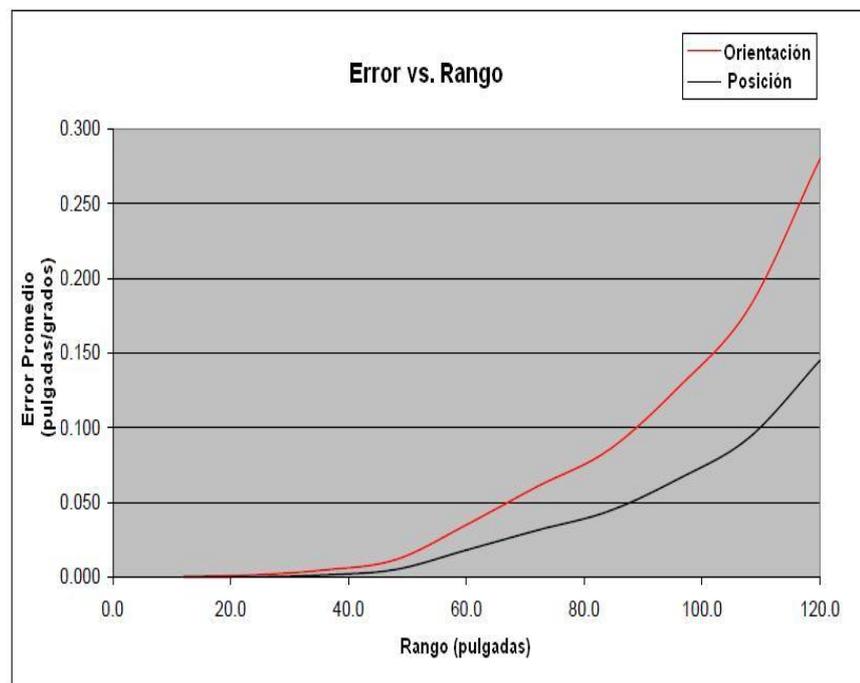


FIGURA 4.5: Error en desempeño de los trackers

La **FIGURA 4.5**, muestra el valor de error que se introduce en la medición de la orientación y posicionamiento de un objeto, debido a la distancia en que se encuentra el sensor con respecto del transmisor.

RANGO (pulgadas)	ERROR DE DIRECCIÓN (grados)	ERROR DE POSICIÓN (pulgadas)
12.0	0.000415	0.000056
24.0	0.001450	0.000285
36.0	0.004843	0.001484
48.0	0.011768	0.005141
60.0	0.035014	0.018139
72.0	0.060574	0.031380
84.0	0.084804	0.043932
96.0	0.127206	0.065898
108.0	0.180633	0.093575
120.0	0.280113	0.145109

TABLA 1: Valores de error según rango del sensor

De acuerdo con la **TABLA 1**, nos damos cuenta que en un rango de hasta 50 pulgadas (127 cm.) más o menos, la precisión del equipo no se afecta, por el contrario cuando el sensor se encuentra a 100 pulgadas (254 cm.) del emisor, va a presentarse distorsión en los valores que van a ser transmitidos a la computadora, lo que influye en la precisión y el control de la aplicación.

En nuestro proyecto trabajamos sobre un área de 75 cm de largo por 25 cm de ancho, con lo cual podemos asegurar que los valores obtenidos presentan mínimo error.

4.3.2. Análisis de algoritmos para corrección de error de los trackers

En este tipo de tracker, la fuente produce tres campos electromagnéticos, cada uno de los cuales es perpendicular a los otros. El sensor colocado sobre el cuerpo del usuario mide la velocidad y dirección de ellos y envía esta información a la computadora, en donde se triangula la distancia y la orientación de los tres ejes perpendiculares en el sensor relativos a los tres campos electromagnéticos producidos por la fuente.

Por medio de las pruebas que se realizaron, se descubrió que el emisor del tracker se comporta como un transmisor omnidireccional [27], y por ende presenta características similares en la propagación de su señal.

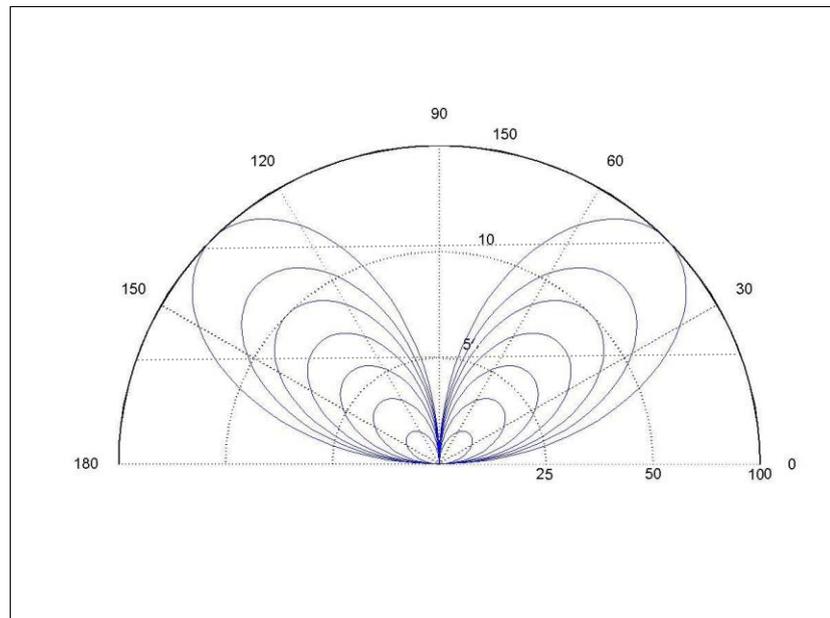


FIGURA 4.6: Propagación transmisor omnidireccional

En la **FIGURA 4.6**, nos podemos dar cuenta que va a existir un área donde no va a ver cobertura para la adquisición de datos y pueden producirse datos aberrantes. De acuerdo a nuestras pruebas de ensayo y error, determinamos que sobre los 8 cm. de alto los valores capturados van a ser confiables, ya que se encuentran dentro del campo electromagnético del transmisor. Para este análisis, empezamos por definir 3 espacios de trabajo con sus respectivos ejes de coordenadas, que podemos observar en la **FIGURA 4.7**.

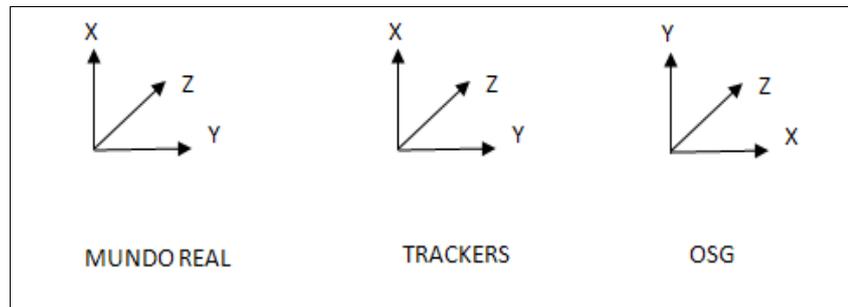


FIGURA 4.7: Ejes coordenados del mundo real, trackers y osg

En el espacio del tracker se definió el eje “x” hacia arriba, debido a que por sus características el emisor solo registra valores positivos, y al estar apoyado el plano base sobre una superficie, se descarta la posibilidad de necesitar los valores negativos en este eje.

Se definió al “z” como el eje de profundidad y al eje “y” como movimiento horizontal.

Se decidió implementar el mismo eje de coordenadas para el mundo real y así facilitar el análisis al momento de medir los valores en el plano base o muestrear valores para la corrección de errores.

Debido a que el prototipo de la aplicación se empezó a desarrollar sin el uso de los trackers, se siguió el eje de coordenadas propuesto por defecto por la librería de programación osg, en el cual se define el eje “y” hacia arriba,

por lo cual al mapear una coordenada del espacio del tracker al del *osg* solo se realiza un cambio de ejes.

Posteriormente, dividimos el plano base como se muestra en la

FIGURA 4.8, trazando líneas separadas cada 5 cm sobre el eje “z”. Tomamos un punto inicial (x_1, y_1, z_1) y desplazamos en línea recta el sensor hacia el punto final (x_2, y_2, z_2) , los desplazamientos en “y” y “z” deberían ser aproximados a 0. Se encontró que los valores obtenidos contenían cierta desviación con respecto al valor esperado, como se muestra en la

TABLA TABLA 2.

FACTOR DE CORRECCION TRACKERS									
cm(z)	PUNTO INICIAL			PUNTO FINAL			ERROR		
	X	Y	Z	X	Y	Z	x	y	Z
0	2.00	8.90	0	3.79	29.00	1.83	1.79	0	1.83
T 5	2.00	8.90	2.90	4.36	28.69	3.67	2.36	0	0.77
A 10	2.00	8.90	4.62	5.09	28.77	5.38	3.09	0	0.76
B 15	2.00	8.90	6.5	5.66	28.53	6.95	3.66	0	0.45
L 20	2.00	8.90	8.44	6.13	28.44	8.30	4.13	0	0.14
A 25	2.00	8.90	9.78	6.47	28.36	9.70	4.47	0	0.08

TABLA 2: Valores de factor de corrección de los trackers

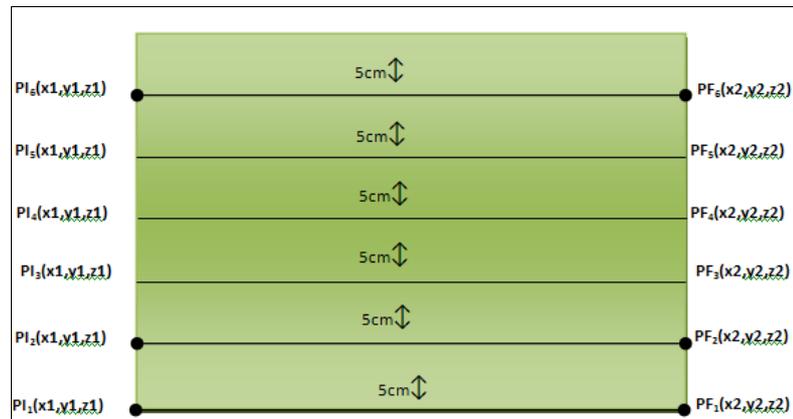


FIGURA 4.8: Líneas dibujadas en la base con separación de 5 cm

Analizando los datos de la tabla, pudimos calcular que para poder realizar una línea recta con precisión deberíamos aplicar la siguiente corrección a los valores obtenidos en los ejes “y” y “z” del tracker, las mismas que se especifican en la

FIGURA 4.9.

$$f(y) = \begin{cases} 0, & y < 0.3 \\ 0.5, & 0.3 \geq y \geq 0.8 \\ 1, & y \geq 0.8 \end{cases}$$

$$f(z) = \begin{cases} 0, & z < 0.25 \\ 0.25, & 0.25 \geq z \leq 0.4 \\ 0.5, & 0.4 \geq z \leq 0.65 \\ 0.75, & 0.75 \geq z \leq 0.85 \\ 1, & z \geq 0.85 \end{cases}$$

FIGURA 4.9: Funciones de transformación de coordenadas

Una vez que corregimos la desviación generada en los valores del tracker, seguimos la siguiente línea de transformaciones que se muestran en la

FIGURA 4.10, la realizamos marcando los cuatro extremos que conforman el límite del área de trabajo, cuyos valores se muestran en la **TABLA 3**.

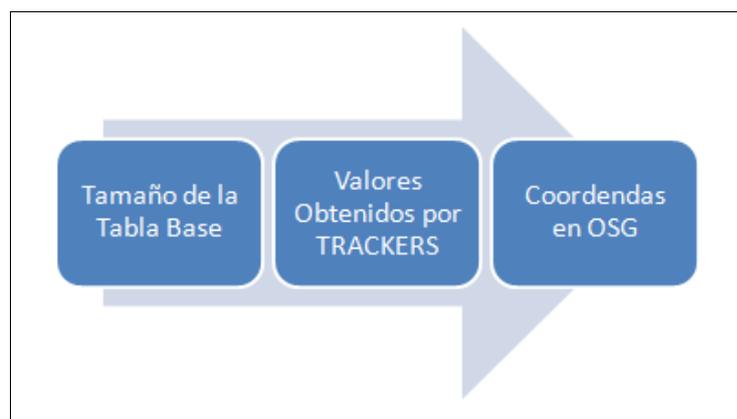


FIGURA 4.10: Orden de las transformaciones realizadas

TRANSFORMACION DE CORDENADAS X,Y,Z									
	MUNDO REAL(cm)			TRACKERS			OSG		
Punto	X	Y	Z	X	Y	Z	X	Y	Z
1	8	20	0	2.3	9.1	0	0	9.2	0
2	8	20	25	6.7	9.1	9.7	0	9.2	-312
3	8	75	25	6.6	28.8	9.7	312	9.2	-312
4	8	75	0	3.9	28.8	0	312	9.2	0

TABLA 3: Transformación de coordenadas del mundo real al virtual

Entonces para cada punto del espacio real (mr) dentro de las dimensiones de la tabla base, le corresponde un punto del espacio del tracker (t) de la siguiente manera:

$$X_t = X_{mr}$$

$$Y_t = f(y)_{mr}$$

$$Z_t = f(z)_{mr}$$

El vector que transforma un punto del mundo real al espacio del tracker es el siguiente:

$$V_{tracker} = \{X_t, Y_t, Z_t\}$$

Y a cada punto del espacio osg le corresponde un punto en el espacio del tracker de la siguiente manera:

$$X_{osg} = ANCHO_DEL_BLOQUE_DE_LEGO_EN_OSG * \\ FACTOR_DE_DESPLAMIENTO_DEL_ANCHO * \\ (Y_t - ALTURA_MINIMA_SIN_ERRORES)$$

$$Y_{osg} = ALTURA_MIN_OSG * [ALTURA_BLOQUE_DE_LEGO_OSG * X_t]$$

$$Z_{osg} = INVERSO_DE(LARGO_BLOQUE_OSG * \\ FACTOR_DE_DESPLAMIENTO_DE_LA_PROFUNDIDAD * Z_t)$$

Por lo tanto el vector quedo de la siguiente manera:

$$V_{osg} = \{8*[2*(Y_t - 9), 9.2 + [9.2* X_t], -8*[4*Z_t]]\}$$

4.3.3. Calibración de dispositivos hápticos

El guante de datos en conjunción con los trackers conforman los dispositivos indispensables para la interacción del usuario con la aplicación. Para poder calibrarlos se realizaron distintas pruebas en donde se capturaban los valores de los gestos hechos por la mano del usuario, primero al hacer el símil de tratar de coger un objeto, segundo el movimiento que realizaba con la mano al tener agarrado un objeto, y tercero el que realizaba al soltar un objeto; lo cual se traduciría en un movimiento natural dentro de la escena virtual.

En los distintos casos, dos gestos eran denominadores comunes: la mano cerrada representaba un objeto mientras se tenía agarrado, y la mano abierta representaba el movimiento para soltarlo. Se presentaron problemas en la captura del segundo gesto (simular agarrar un objeto), los valores variaban notablemente según el usuario que utilizaba el guante, debido a las diferencia de tamaño y forma de los dedos y la palma de la mano, por lo cual se creó un rango promedio con los valores capturados para poder solucionar este problema.



FIGURA 4.11: Valores, gestos y acciones del guante de datos

La **FIGURA 4.11**, muestra una referencia de los valores de los gestos de la mano según el fabricante [1], utilizada para fines de explicación. El gesto de valor 0 (puño cerrado) representa la acción de tener agarrado un objeto, el gesto de valor 1 (mano abierta), representa la acción de soltarlo, los siguientes representan los valores promedio (8 - 13), que se capturaron en las pruebas realizadas con usuarios diferentes, del gesto que representaba tratar de agarrar un objeto.

Previo al lanzamiento de la aplicación, se realiza la calibración de la mano del usuario, en donde se le pide que haga un determinado gesto asignándole el valor de mayor frecuencia.

4.4. Plan de pruebas

4.4.1. Descripción de pruebas

El objetivo que se quiso alcanzar con las pruebas era evaluar la precisión, la facilidad de movimiento y la estabilidad que presentaba la aplicación, con el propósito de corregir errores las siguientes tareas:

- a. Gestos realizados con el guante.**- una vez que el usuario se colocaba el guante de datos, se debía calibrar el valor de los gestos que realizaba con la mano. Si los datos que transmitía el guante no estaban dentro del rango establecido en la

aplicación, no se podía mover ni colocar un bloque dentro del área de trabajo.

b. *Movimientos realizados con el tracker.*- cuando se utilizaba el tracker, se debía verificar la fluidez y sincronización del movimiento. Este proceso fue el más complicado de realizar, ya que aquí se debía tomar en consideración los retardos propios de los dispositivos que inciden en la suavidad del movimiento, lo que provocaba que lo que hagamos en el mundo real no fuera vea reflejando coherentemente en el virtual, dando como resultado una aplicación poco precisa.

c. *Seleccionar, mover y ubicar los bloques en puntos específicos del área de trabajo.*- esta tarea consistía en hacer que el usuario de acuerdo a la cuadrícula que tenía en el mundo real, se ubicara en puntos críticos como los extremos, para evaluar si estos movimientos se reflejaban en el entorno virtual. Además se le pidió que se ubique en determinadas áreas para verificar que la aplicación respondiera como era deseado, como por ejemplo cuando se ubicaba un bloque fuera del tablero éste se eliminaba; o que se validara que no se pudiera colocar un bloque en una posición ocupada por otro; o que no se pudiera mover un bloque que tenía otro bloque sobre él.

d. Realizar picking selectivo con el guante.- aquí se verificaba que el picking que se realizaba con el guante funcionara correctamente, se le pedía al usuario que seleccionara aleatoriamente ciertos bloques que ya hubieran sido colocados previamente, y realizara acciones como colocarlos en una posición diferente o cambiarles el color original.

e. Almacenar una escena y abrir una creada anteriormente.- una vez que el usuario había creado un objeto, se le pedía que con el tracker se dirigiera al área donde se encontraba el cajón con el ícono de guardar, para que almacenara la escena creada. Luego se le pedía que borre el área de trabajo, y con el tracker se dirigiera al cajón con el ícono de abrir para cargar una escena almacenada previamente.

De esta forma verificábamos que la aplicación respondiera adecuadamente.

4.4.2. Evaluación de resultados

VERSIÓN 1.0 LEGO 3D

Las siguientes pruebas fueron realizadas con la primera versión de la aplicación inmersiva, aquí se detectaron muchas fallas y omisiones en lo referente a la interacción entorno virtual-usuario, entre las que podemos destacar las siguientes:

- El movimiento de traslación que se realizaba con los trackers era muy impreciso, el usuario no podía colocar un bloque en un punto determinado del tablero virtual, porque el sensor no estaba sincronizado con el movimiento real de la mano, siempre se desplazaba una posición más.
- No se manejaba la retroalimentación cuando se seleccionaba una pieza, el usuario no tenía forma de saber si estaba agarrando un bloque, hasta que se colocaba sobre el área de trabajo y realizaba el acto de soltarlo.
- Los gráficos de los bloques no era muy realista, no poseían la forma redondeada característica de los Legos.
- El movimiento de la cámara era muy inestable y provocaba cambios bruscos en la escena, lo que hacía que el usuario por momentos perdiera la concentración.
- No se manejaba el picking selectivo, es decir el usuario sólo podía mover o cambiar de color al último bloque que añadía a la escena no a los colocados anteriormente.
- No podía eliminar un bloque.
- Cuando el usuario hacía el respectivo gesto con el guante para seleccionar un bloque no colocado en el tablero, había un pequeño retardo en la captura de éste, lo que hacía que se seleccionara el siguiente y no el que se deseaba.

- Existían errores en el algoritmo que verificaba si determinada posición en el área de trabajo estaba vacía, lo que provocaba que se monten las piezas, cuando se colocaban una sobre otra o una al lado de la otra.
- No se validaba que los bloques sólo fueran colocados dentro del tablero, el usuario podía colocarlos en cualquier sitio fuera o sobre éste.
- El usuario no podía almacenar o reutilizar una escena creada previamente.

VERSIÓN 2.0 LEGO 3D

Las siguientes pruebas fueron realizadas con la segunda versión del programa, se solucionaron problemas graves que se presentaban en la versión anterior, pero se encontraron nuevos que afectaban la estabilidad de la aplicación.

PROBLEMAS SOLUCIONADOS:

- Se logró obtener mayor precisión en el movimiento con los trackers, el usuario no tiene que estirar y forzar el brazo en el mundo real, para que en el mundo virtual se vea reflejado su movimiento. La plantilla que se utilizó y que fue colocada

sobre la mesa, fue de gran importancia y guía para alcanzar esta meta.

- Ahora el usuario sabe que ha seleccionado un bloque, ya que en la aplicación se puede observar que la mano virtual lleva consigo la pieza, la cual se mueve con ella hasta que el usuario decide colocarla sobre el área de trabajo.
- Los bloques de Lego fueron modelados nuevamente para sean más realistas.
- Se logró controlar el movimiento de la cámara, ahora se maneja con un segundo tracker que se coloca sobre el cuerpo del usuario, los cambios de escenas son naturales y se realizan de acuerdo a la ubicación del usuario.
- En esta versión se logró implementar el picking selectivo, ahora el usuario con la ayuda del guante de datos, puede seleccionar cualquier bloque de la escena y no solamente el último colocado en el área de trabajo.
- El usuario ahora puede eliminar un bloque, llevándolo fuera del área de trabajo.
- Se logró eliminar el retardo en la selección de objetos con el guante.

- Fueron corregidas las incongruencias en los algoritmos que se encargaban de verificar si una posición estaba vacía, ahora los bloques no se montan sobre otros ubicados anteriormente.
- Ahora el usuario sólo puede colocar bloques dentro del área de trabajo, si coloca un bloque fuera de ésta, el bloque se elimina.
- El usuario puede almacenar y reutilizar una escena creada previamente.

ERRORES ENCONTRADOS:

- Se podía sacar un bloque intermedio, los bloques que estaban colocados sobre éste quedaban flotando.
- Al almacenar una escena, las posiciones decimales se redondeaban a enteros, lo que daba problemas al volver a cargar la escena, ya que se presentaba conflicto en el posicionamiento de los bloques, lo que resultaba en que algunos se pudieran colocar en puntos del tablero ya ocupados.

- Cuando se abría una escena no se borraban las piezas que estaban colocados en el área de trabajo, los bloques almacenados se colocaban sobre ellas.
- No funcionaba la acción de salir de la aplicación.

VERSIÓN 3.0 LEGO 3D

Aplicación final, fueron corregidos los errores que presentaban las versiones anteriores.

A continuación en la **TABLA 4**, se presenta un resumen de los resultados obtenidos de las pruebas realizadas:

TAREAS	LEGO VERSIÓN 1.0	LEGO VERSIÓN 2.0
PRESICIÓN EN MOVIMIENTO DE LOS TRACKERS	Movimientos imprecisos y poco reales, muy difícil de manejar por parte del usuario.	Movimientos con mayor precisión, el usuario no se tiene que esforzar para que sus movimientos reflejen la posición que desea en el mundo virtual.
RETROALIMENTACIÓN ADECUADA EN SELECCIÓN DE OBJETOS	No se maneja la retroalimentación, el usuario no está seguro si ha seleccionado o no un objeto.	El usuario está seguro de que ha seleccionado un objeto, debido a que se maneja la retroalimentación.
REALISMO DE OBJETOS	Detalles poco reales.	Se mejoró el realismo de los objetos.
CÁMARA DE LA ESCENA ESTABLE	Se muestran cambios bruscos en el movimiento de la cámara de la escena.	Movimientos de cámara más naturales, se manejan con un segundo tracker.
MANEJO DEL PICKING SELECTIVO DE OBJETOS	No se maneja el picking selectivo.	Se maneja el picking selectivo.

ELIMINACIÓN DE BLOQUES	No se puede eliminar un bloque de la escena.	El usuario puede eliminar un bloque de la escena.
RETARDO EN LA SELECCIÓN DE OBJETOS CON EL GUANTE	Se presenta un pequeño desfase en la selección de objetos.	No se presenta retardo al momento de realizar la selección de objetos.
CORRECTO DESEMPEÑO DE ALGORITMOS DE CONTROL DE UBICACIÓN DE BLOQUES	No se verificaba correctamente si una posición estaba o no ocupada, los bloques tendían a montarse sobre otro.	Se verifica correctamente las posiciones ocupadas, los bloques ya no se montan sobre otros. Se puede sacar un bloque intermedio, quedando bloques flotando.
PERMITE ALMACENAR UNA ESCENA	No se puede guardar una escena.	Se puede guardar una escena, pero no se almacenan correctamente las posiciones de los bloques.

TABLA 4: Datos comparativos entre las versiones de la aplicación

CONCLUSIONES

Una vez finalizadas las etapas de implementación y pruebas de la aplicación virtual, podemos concluir que:

1. Usando las técnicas de Realidad Virtual, se pudo experimentar, analizar y aplicar los conceptos de Inmersión, Presencia e Interacción, creando un entorno virtual del popular juego de bloques de LEGO.
2. Mediante un análisis minucioso de las características y funcionamiento de los trackers, se pudo remediar los errores que se obtenían en la captura de los datos de posicionamiento, los mismos que provocaban que la aplicación sea muy inestable y difícil de manejar.
3. Para solucionar la falencia que presentaba el tracker al no introducir valores negativos de "x", se aprovechó el hecho de que en nuestra aplicación los bloques se colocaban sobre una superficie, por lo cual se cambió la disposición de los ejes de coordenadas del emisor, dejando a "x" como medida de altura.
4. En nuestra aplicación, podemos asegurar mediciones sin interferencia con el tracker, dentro de una zona de 75 cm. de ancho, 25 cm de largo y 8 cm. de alto.

5. Con la ayuda de ecuaciones de transformación y algoritmos de control, se pudo obtener alta precisión en el movimiento de traslación que se realizaba con de los trackers, además permitió la verificación de posiciones ocupadas para evitar el traslape entre los bloques.
6. Gracias a la calibración previa de los gestos realizados con la mano, se pudo conseguir que los movimientos dentro de la aplicación virtual fueran muy fluidos y similares a los que comúnmente hacemos en la realidad.
7. De acuerdo a los resultados obtenidos, para este tipo de proyectos, el control y la precisión de los movimientos por medio de los dispositivos son de vital importancia, ya que de esto depende el realismo, la comodidad y aceptación del usuario a este tipo de aplicaciones no comunes, esto se logra mediante la realización de pruebas exhaustivas de interacción.
8. Finalmente, a pesar de los altos costos de los equipos que se utilizan, la Realidad Virtual se perfila como el futuro de las aplicaciones de software. Actualmente muchas empresas están dando el primer paso para lograr su masificación desarrollando equipos más económicos y accesibles.

RECOMENDACIONES

Para la elaboración de futuros proyectos en este campo, podemos recomendar que:

1. Para poder contribuir con la característica de inmersión, se recomienda la creación de una CAVE, en donde el usuario pueda encontrarse aislado de estímulos del mundo exterior y concentrarse completamente en la aplicación virtual. Esto además permitiría las aplicaciones multiusuario.
2. Se deben investigar y utilizar editores y herramientas gráficas que permitan crear gráficos por computadora, más realistas y con alto detalle, pero que no incidan en el desempeño de la aplicación.
3. Al momento de desarrollar este tipo de aplicaciones se debe poner énfasis en áreas críticas, como lo son la calibración y precisión de los equipos que van a ser utilizados en la interacción, ya que de esta forma podemos considerar el alcance que va a tener nuestro proyecto.
4. Para mayor comodidad del usuario se recomienda la utilización de dispositivos totalmente inalámbricos, como trackers infrarrojos, ya que muchas veces los usuarios no se involucran en el mundo virtual debido a que se encuentran atados a cables y dentro de un espacio limitado.

5. Para futuros proyectos se recomienda el uso de otros motores de procesamiento de gráficos 3D, ya que OpenSceneGraph era muy inestable y provocaba que el tiempo de respuesta en la interacción aumente drásticamente cuanto se incrementaba el número de objetos en la escena.

APÉNDICES

GLOSARIO DE TÉRMINOS

3ds.	Es el formato de los objetos que son creados en 3D Studio Max.
3D Studio Max.	Es un programa de creación de gráficos y animaciones 3D, desarrollado por Autodesk Media & Entertainment.
Ambiente 3D.	Es una ambiente en el que se puede determinar las tres dimensiones: largo, ancho y profundidad.
Ataxia.	Es una enfermedad que se caracteriza por provocar descoordinación en los movimientos de las partes del cuerpo.
Avatar	Es una representación gráfica, generalmente humana, que se asocia a un usuario para su identificación.
Bluetooth	Especificación industrial para Redes Inalámbricas de Área Personal que posibilita la transmisión de voz y datos entre diferentes dispositivos mediante un enlace por radiofrecuencia.
Cybersickness.	Es una enfermedad provocada por la inmersión e incluye síntomas como náuseas, vértigo, ataxia y dolores cervicales.

Entorno amigable.	Interfaz de un programa de por su forma de interacción con el usuario es considerada de fácil uso.
Entorno virtual.	Es una simulación tridimensional de algún aspecto del mundo real, que son generadas por computadoras.
Formato.	Es una forma particular de codificar información (documentos, imágenes, audio) para ser almacenada.
Gafas estereoscópicas.	Son gafas especiales con las cuales podemos visualizar elementos 3D, ya que crean la ilusión de profundidad en la imagen.
Grados de libertad.	Se conoce como grados de libertad a los movimientos (rotación y traslación) que puede tener un objeto en el eje 3D.
Inmersión.	Se refiere a la sustitución de nuestra percepción del mundo real por una percepción generada por una máquina.
Interacción.	Es el intercambio de información que se realiza entre el usuario y el ambiente virtual.
jpg.	Es un formato de compresión de imágenes.
LEGO.	Empresa de juguetes danesa reconocida por sus bloques de plástico interconectables.

OpenSceneGraph.	Es una librería para gráficos 3D de alto rendimiento utilizada en simulación, juegos, realidad virtual y modelamiento.
Pitch	Es un término que hace alusión a los ángulos de navegación usados en aeronáutica para describir la orientación de un objeto 3D. En español se le denomina <i>cabeceo</i> y se refiere a la rotación alrededor del eje “x”.
Realidad virtual.	Es una simulación del mundo real por medios electrónicos, que se puede visualizar, manipular y explorar en tiempo real.
Roll	Es un término que hace alusión a los ángulos de navegación usados en aeronáutica para describir la orientación de un objeto 3D. En español se le denomina <i>alabeo</i> y se refiere a la rotación alrededor del eje “x”.
Simulador.	Es un aparato que reproduce las sensaciones físicas y de entorno de un equipo, máquina o sistema.
Tracker.	Es un dispositivo que permite detectar la posición de un objeto o una persona. Pueden ser ubicados en los HMD o en los guantes.
Venopunción.	Es la recolección de sangre de una vena.
VRML.	Acrónimo de Virtual Reality Modelling Language, es un lenguaje de modelado de mundos virtuales en 3D, a los que accedemos utilizando nuestro navegador de Internet

- wav.** Es un formato de audio digital normalmente sin compresión que se utiliza para guardar sonidos en la computadora.
- Yaw** Es un término que hace alusión a los ángulos de navegación usados en aeronáutica para describir la orientación de un objeto 3D. En español se le denomina *guiñada* y se refiere a la rotación alrededor del eje “y”.

REFERENCIAS BIBLIOGRÁFICAS

- [1] 5DT Fifth Dimension Technologies, Data Glove Ultra User's Manual, <http://www.5dt.com/downloads/dataglove/ultra/5DT%20Data%20Glove%20Ultra%20-%20Manual.pdf>, Octubre 2004.
- [2] 5 DT dataglove, Product Description, <http://5dt.com/products/pdataglove5u.html>, 10 de septiembre de 2009.
- [3] AUTODESK. Training & Support, <http://autodesk.com>. 19 de agosto de 2009.
- [4] BlockCAD. Program Instructions, <http://web.telia.com/~u16122508/proglego.htm>, 25 de septiembre de 2009.
- [5] BRUNET, Peré. "*La verdad en tres dimensiones*". El Cultural [en línea]. 27 de julio del 2002. Disponible en: < www.elcultural.es >. [Consultada: 28 de agosto de 2009].
- [6] CARDONA, Jesús David. "*Realidad virtual y entornos virtuales*". Colombia: Universidad Autónoma de Occidente, 2006.
- [7] CathSim Accutouch. Especificaciones Técnicas, http://simcen.org/surgery/hardware/CathSim_Accutouch.html, 13 de agosto de 2009.
- [8] DEPTHQ Stereoscopic, Technical Specifications, <http://depthQ.com>, 10 de septiembre de 2009.

- [9] GUTIERREZ, Mario; Vexo Frédéric y Thalmann Daniel. “*Stepping into virtual Reality*”. Inglaterra: Springer, 2008.
ISBN: 978-1-84800-116-9
- [10] ESCARTIN, Emilio. “*La Realidad Virtual, Una tecnología educativa a nuestro alcance*”. Cuba: ISPJAE, 2000.
- [11] Wikipedia. *Realidad Virtual*,
http://es.wikipedia.org/wiki/Realidad_virtual, 30 de julio 2009.
- [12] HAYT, William; Buck, John. “Teoría Electromagnética”. Mexico: McGraw-Hill, 2006. ISBN: 970-105620-5
- [13] Innovatecno, Sistemas de Telepresencia,
<http://www.innovatecno.com>, 12 de junio de 2010.
- [14] KISMET, 3D-Simulation Software, http://iregt1.iai.fzk.de/KISMET/UGK_home.htm, 15 de julio de 2009.
- [15] LabHuman. Human Centered Technology, <http://labhuman.com>, 13 de agosto de 2009.
- [16] LEGO. The Official Web Site of LEGO, <http://lego.com>, 28 de julio de 2009.
- [17] LIBERTY, Tech compare with ResolutionvsRange graphs.pdf. [en línea]. Disponible en: <www.polhemus.com>, 2008.
- [18] MATEO, Alfredo; Coto, Ernesto; Navarro, Héctor; Rodríguez, Omaira. “*Una herramienta para general mundos virtuales inmersivos*”. Venezuela: Universidad Central de Venezuela, 2001. SIN 1316-6239

- [19] MARTINEZ, Ma. Ángeles. “La realidad virtual a través de la teoría de los mundos posibles”. España: Universidad Católica San Antonio de Murcia, 2004. págs.: 778 - 788.
- [20] NuVision, LCD Stereoscopic Display, <http://nuvision3d.com>, 10 de septiembre de 2009.
- [21] OpenSceneGraph, 3D Graphics Toolkit, <http://openscenegraph.org>, 12 de julio de 2009.
- [22] ORTIZ, Oscar. “Interacción y usabilidad de interfaces en realidad virtual”. España: Universidad Rey Juan Carlos, 2003.
- [23] PARASIM, Virtual Reality Parachute Simulator, <http://parasim.com>, 13 de agosto de 2009.
- [24] POLHEMUS Innovation in Motion, Technical Specifications, <http://polhemus.com>, 10 de septiembre de 2009.
- [25] RAMOS NAVA, María del Carmen; Larios Delgado, José; Cervantes Cabrera, Daniel y Leriche Vázquez, Renato. “Creación de ambientes virtuales inmersos con software libre”. Revista Digital Universitaria [en línea]. 10 de junio 2007, Vol. 8, No. 6. Disponible en: <www.revista.unam.mx/vol.8/num6> [Consultada: 06 de septiembre de 2009].
- [26] SECOND LIFE, Free Virtual World & Avatars, <http://secondlife.com>, 30 de septiembre de 2009.

- [27] SERWAY, Raymond. "*Física para ciencias e ingeniería*". Mexico: McGraw-Hill, 2002. ISBN: 970-103581-X
- [28] SIMBIO, Simulation & Training Solutions, <http://landersimulation.com>, 13 de agosto de 2009.
- [29] Toyota, Simulador de Conducción, <http://safetytoyota.com>, 13 de agosto de 2009.
- [30] VILLAREAL, Silvia. "Los juegos didácticos como factores de desarrollo de comunidades de aprendizaje". México: Universidad Autónoma de México, 2002.

ANEXO A

Casos de uso

A continuación detallamos los distintos casos de uso que desarrollamos a lo largo de la aplicación:

Nombre:	1. Seleccionar bloque de Lego
Descripción:	El usuario al realizar un gesto determinado con la mano, el cual es sentido por medio del guante de datos, puede seleccionar entre las distintas piezas que tiene a su disposición.
Nota:	Para poder seleccionar un bloque, el gesto que se realiza con la mano debe estar dentro de un rango establecido.
Valor medible:	Se puede seleccionar una pieza de Lego en particular o no.
Escenarios:	1.1 El usuario puede seleccionar la pieza de Lego. 1.2 El usuario no puede seleccionar la pieza de Lego.

Nombre:	2. Colocar bloque de Lego
Descripción:	El usuario al realizar un gesto determinado con la mano, el cual es sensado por medio del guante de datos, puede ubicar una pieza de Lego seleccionada dentro del área de trabajo.
Nota:	Para poder colocar un bloque en la base de Lego, el gesto que se realiza con la mano debe tener un valor establecido. Solo se puede ubicar los bloques dentro de la base de Lego.
Valor medible:	Se puede ubicar la pieza de Lego en la escena o no.
Escenarios:	2.1 El usuario puede colocar la pieza de Lego. 2.2 El usuario no puede colocar la pieza de Lego.

Nombre:	3. Cambiar color de bloque de Lego
Descripción:	El usuario puede cambiar el color de un bloque de Lego, al escoger entre las opciones de botes de pintura que se le presentan en la escena, recorriéndolos con ayuda de los trackers y seleccionando el color deseado mediante un gesto de la mano sensado por el guante de datos.

Nota:	El gesto que se realizan con la mano para seleccionar el bote de pintura debe estar dentro de un valor establecido.
Valor medible:	Se puede cambiar el color de una pieza de Lego o no.
Escenarios:	<p>3.1 El usuario puede cambiar el color de un bloque de Lego.</p> <p>3.2 El usuario no puede cambiar el color de un bloque de Lego.</p>

Nombre:	4. Guardar una escena
Descripción:	El usuario puede guardar una escena previamente creada.
Nota:	Se debe seleccionar el objeto 3D que representa al ícono de guardar.
Valor medible:	Se puede guardar una escena o no.
Escenarios:	<p>4.1 La escena se guarda exitosamente.</p> <p>4.2 No se puede guardar la escena.</p>

Nombre:	5. Abrir una escena
Descripción:	El usuario puede abrir una escena previamente creada.
Nota:	Se debe seleccionar el objeto 3D que representa al ícono de abrir.
Valor medible:	Se puede abrir una escena o no.
Escenarios:	5.1 La escena se abre exitosamente. 5.2 No se puede abrir la escena.

Nombre:	6. Eliminar bloque de Lego
Descripción:	El usuario puede eliminar el último bloque ubicado en el área de trabajo.
Nota:	Se debe seleccionar el objeto 3D que representa al ícono de eliminar.
Valor medible:	Se puede eliminar el bloque de Lego o no.
Escenarios:	6.1 El bloque de Lego es eliminado exitosamente. 6.2 No se puede eliminar el bloque de Lego de la escena.

Nombre:	7. Visualizar la escena en 3D
Descripción:	El usuario por medio de las gafas estereofónicas puede visualizar la escena proyectada en la pantalla en tres dimensiones.
Nota:	El usuario debe estar ubicado en una posición adecuada para que el emisor conectado a la computadora pueda emitir los pulsos infrarrojos a las gafas.
Valor medible:	Se visualizar en 3D la escena o no.
Escenarios:	7.1 La escena se puede visualizar en 3D. 7.2 La escena no se puede visualizar en 3D.

Escenarios

A continuación detallamos los escenarios que se presentaron a lo largo del desarrollo de la aplicación:

Caso de Uso 1	Seleccionar bloque de Lego
Escenario 1.1	El usuario puede seleccionar la pieza de Lego
Asunciones:	<ul style="list-style-type: none">• El usuario se trasladó correctamente con los trackers hasta el colector de bloques de Lego.• El usuario realizó con su mano los gestos dentro del rango de valores establecidos para poder seleccionar los bloques.• El guante sensó correctamente el gesto realizado por el usuario.
Resultados:	<ul style="list-style-type: none">• Se puede visualizar entre las distintas opciones de bloques disponibles.• El bloque seleccionado se queda estático sobre el colector.

Caso de Uso 1	Seleccionar bloque de Lego
Escenario 1.2	El usuario no puede seleccionar la pieza de Lego.
Asunciones:	<ul style="list-style-type: none"> • El usuario no se trasladó con los trackers hasta el colector de bloques de Lego. • El usuario no realizó el movimiento o gesto con la mano dentro del valor establecido. • El guante no sensó correctamente el gesto realizado por el usuario.
Resultados:	<ul style="list-style-type: none"> • No se puede visualizar las piezas disponibles. • No se puede visualizar ningún bloque sobre el colector.

Caso de Uso 2	Colocar bloque de Lego
Escenario 2.1	El usuario puede colocar la pieza de Lego
Asunciones:	<ul style="list-style-type: none"> • El usuario realizó con su mano el gesto correcto para soltar la pieza seleccionada. • El usuario se colocó sobre un punto deseado dentro del área de trabajo.
Resultados:	<ul style="list-style-type: none"> • Se puede visualizar que el bloque está colocado sobre un área del área de trabajo.

Caso de Uso 2	Colocar bloque de Lego
Escenario 2.2	El usuario no puede colocar la pieza de Lego
Asunciones:	<ul style="list-style-type: none"> • El usuario no realizó con su mano el gesto correcto para soltar la pieza seleccionada. • El usuario se colocó sobre otro punto de la escena fuera del área de trabajo.
Resultados:	<ul style="list-style-type: none"> • No se puede visualizar que el bloque.

Caso de Uso 3	Cambiar color de bloque de Lego
Escenario 3.1	El usuario puede cambiar el color de un bloque de Lego
Asunciones:	<ul style="list-style-type: none"> • El usuario se pudo trasladar a través de los botes de pintura al mover el tracker a la posición establecida. • El usuario realizó correctamente el gesto con la mano para poder seleccionar el color deseado.
Resultados:	<ul style="list-style-type: none"> • Se puede visualizar que el bloque de Lego está del color del bote de pintura escogido.

Caso de Uso 3	Cambiar color de bloque de Lego
Escenario 3.2	El usuario no puede cambiar el color de un bloque de Lego
Asunciones:	<ul style="list-style-type: none"> • El usuario no se pudo trasladar a través de los botes de pintura al mover el tracker. • El usuario no realizó correctamente el gesto con la mano para poder seleccionar el color deseado.
Resultados:	<ul style="list-style-type: none"> • Se puede visualizar que el bloque de Lego está del mismo color inicial.

Caso de Uso 4	Guardar una escena
Escenario 4.1	La escena se guarda exitosamente
Asunciones:	<ul style="list-style-type: none"> • El usuario se pudo trasladar hasta el objeto 3D con el ícono de guardar por medio del tracker.
Resultados:	<ul style="list-style-type: none"> • La escena es guardada exitosamente. • Se muestra un mensaje de éxito.

Caso de Uso 4	Guardar una escena
Escenario 4.2	No se puede guardar la escena
Asunciones:	<ul style="list-style-type: none"> El usuario no se pudo trasladar hasta el objeto 3D con el ícono de guardar por medio del tracker.
Resultados:	<ul style="list-style-type: none"> La escena no es guardada exitosamente. Se muestra un mensaje de advertencia.

Caso de Uso 5	Abrir una escena
Escenario 5.1	La escena se abre exitosamente
Asunciones:	<ul style="list-style-type: none"> El usuario se pudo trasladar hasta el objeto 3D con el ícono de abrir por medio del tracker.
Resultados:	<ul style="list-style-type: none"> La escena es cargada en la escena exitosamente.

Caso de Uso 5	Abrir una escena
Escenario 5.2	No se puede abrir la escena
Asunciones:	<ul style="list-style-type: none"> El usuario no se pudo trasladar hasta el objeto 3D con el ícono de abrir por medio del tracker.
Resultados:	<ul style="list-style-type: none"> La escena no es cargada en la escena exitosamente.

	<ul style="list-style-type: none"> • Se muestra un mensaje de error al usuario.
--	--

Caso de Uso 6	Eliminar bloque de Lego
Escenario 6.1	El bloque de Lego es eliminado exitosamente
Asunciones:	<ul style="list-style-type: none"> • El usuario se pudo trasladar hasta el objeto 3D con el ícono de eliminar por medio del tracker.
Resultados:	<ul style="list-style-type: none"> • El último bloque colocado en la escena se elimina exitosamente. • Se muestra un mensaje favorable al usuario.

Caso de Uso 6	Eliminar bloque de Lego
Escenario 6.2	No se puede eliminar el bloque de Lego de la escena
Asunciones:	<ul style="list-style-type: none"> • El usuario no se pudo trasladar hasta el objeto 3D con el ícono de eliminar por medio del tracker.
Resultados:	<ul style="list-style-type: none"> • El último bloque colocado no es eliminado de la escena. • Se muestra un mensaje de error al usuario.

Caso de Uso 7	Visualizar la escena en 3D
Escenario 7.1	La escena se puede visualizar en 3D
Asunciones:	<ul style="list-style-type: none"> El usuario se colocó las gafas estereofónicas y se ubicó en un lugar donde el emisor podía transmitir sin interferencias.
Resultados:	<ul style="list-style-type: none"> El usuario puede visualizar los objetos en 3D en la escena virtual. El usuario puede interactuar con los objetos de la escena.

Caso de Uso 7	Visualizar la escena en 3D
Escenario 7.2	La escena no se puede visualizar en 3D
Asunciones:	<ul style="list-style-type: none"> El usuario se colocó las gafas estereofónicas y se ubicó en un lugar donde el emisor no podía transmitir la señal correctamente.
Resultados:	<ul style="list-style-type: none"> El usuario no puede visualizar los objetos en 3D en la escena virtual. El usuario visualiza una escena desfasada y con una vibración que causa molestias a los ojos.

- El usuario no puede interactuar con los objetos de la escena, porque no están bien definidos.

Diagramas de secuencia (Escenarios)

Guardar una escena

Guardado exitoso de la escena, el usuario almacena el trabajo realizado previamente.

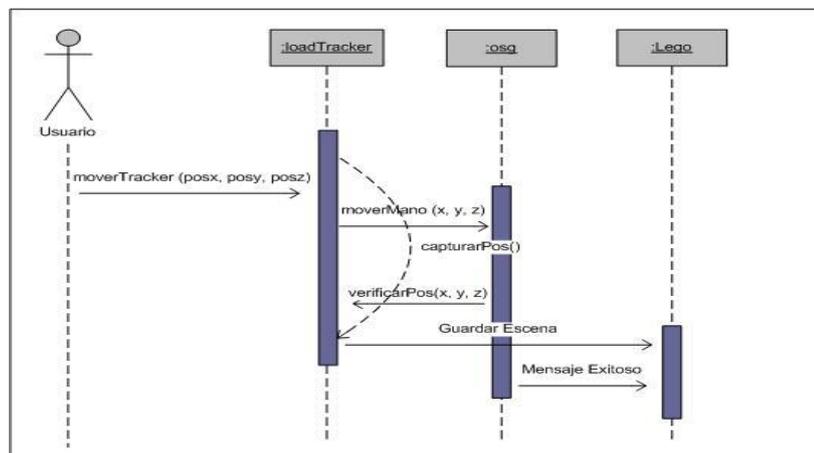


FIGURA 3.8: Diagrama de secuencia – Guardar escena

Abrir una escena

La escena se cargó exitosamente, el usuario puede abrir una escena almacenada previamente.

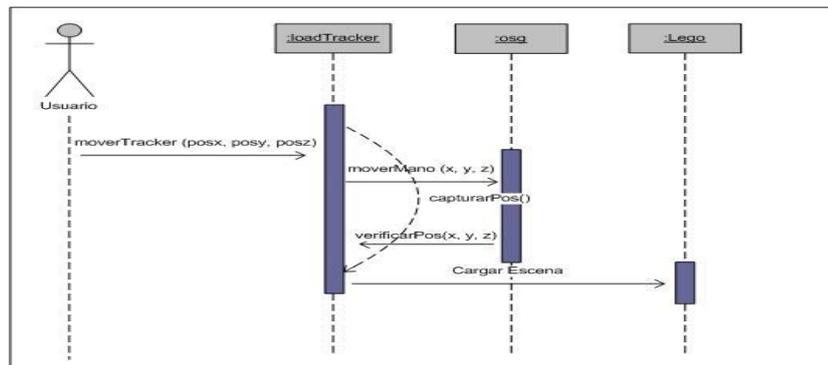


FIGURA 3.9: Diagrama de secuencia – Abrir escena

Eliminar bloque de Lego

Se eliminó exitosamente el bloque de Lego, el usuario puede eliminar de la escena el último bloque que colocó en el tablero.

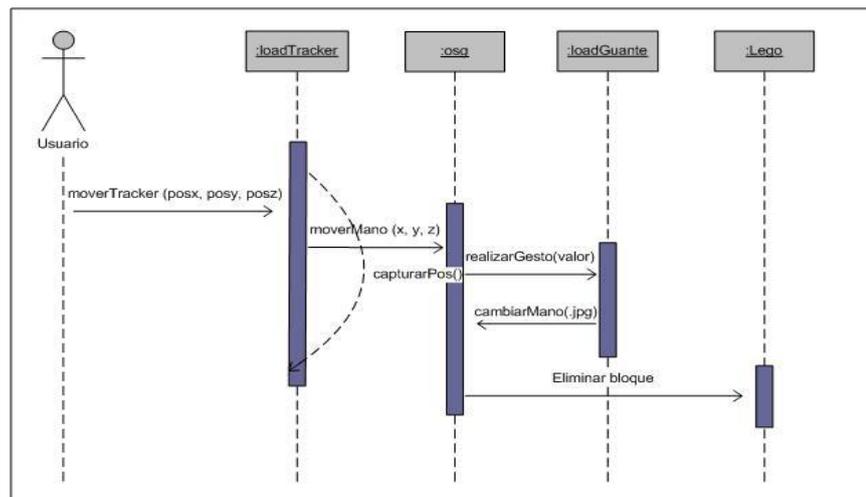


FIGURA 3.10: Diagrama de secuencia – Eliminar bloque de Lego

Visualizar la escena en 3D

Se puede visualiza la aplicación en 3D de forma correcta, el usuario puede interactuar con los objetos de la escena.

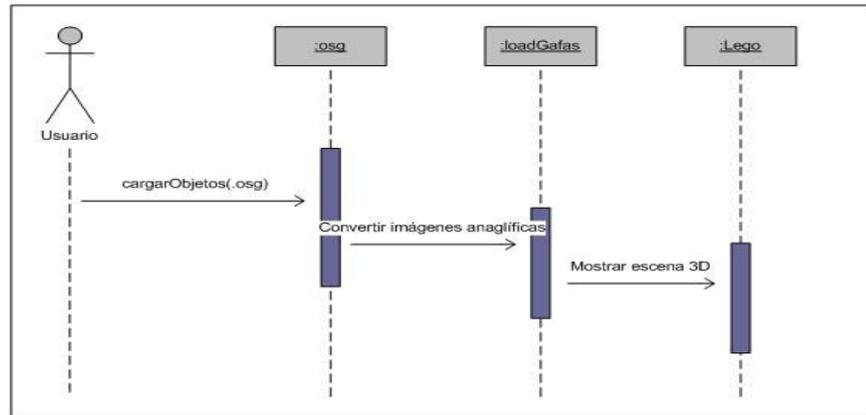
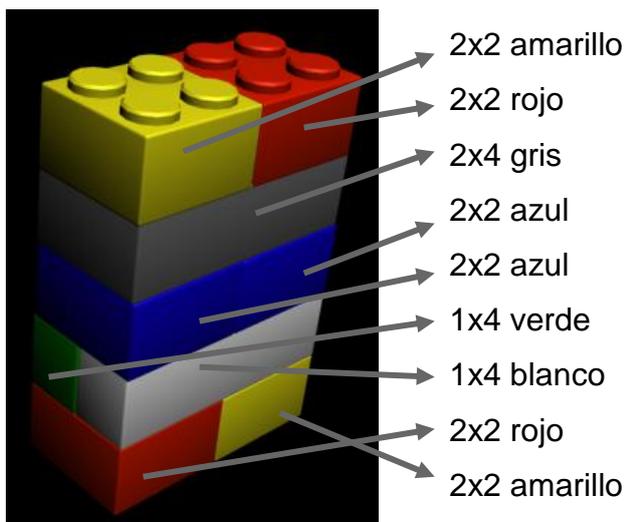


FIGURA 3.11: Diagrama de secuencia – Visualizar la escena 3D

ANEXO B

DETALLE DE ACTIVIDADES A REALIZAR PARA PRUEBAS DE FIABILIDAD DE LA APLICACIÓN

1. Ubicar cuatro bloques de 2x2 en cada uno de los extremos del tablero.
2. Seleccionar y cambiar el color de los bloques de la siguiente manera:
 - bloque (0,0) -> azul
 - bloque (75, 0) -> verde
 - bloque (0, 25) -> amarillo
 - bloque (75, 25) -> rojo
3. Colocar el bloque azul en la posición (25, 45) de la plantilla
4. Seleccionar 2 bloques de 1x2 y colocarlos sobre el bloque azul
5. Seleccionar el bloque verde y colocarlo sobre los bloques de 1x2
6. Arme la siguiente figura utilizando los bloques restantes:



7. Trate de sacar el bloque azul
8. Guarde la escena
9. Abra una nueva área de trabajo
10. Cargue la escena anterior
11. Trate de sacar el bloque amarillo colocado en la parte superior de la figura
12. Coloque el bloque rojo sobre el amarillo.
13. Trate de sacar el bloque azul
14. Seleccione un bloque de 1x1 del cajón de piezas y ubíquelo sobre el bloque verde
15. Cambie el color del bloque 1x1 a café
16. Seleccione el bloque de color café y elimínelo de la escena
17. Salga de la aplicación