



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

Freddy Arboleda¹, Ricardo Laica², Magdalena Loor³, Sixto García⁴
Facultad de Ingeniería en Electricidad y Computación
Escuela Superior Politécnica del Litoral
Campus Gustavo Galindo, Km. 30.5 Vía Perimetral, apartado 09-01-5863
Guayaquil - Ecuador
farboled@fiec.espol.edu.ec¹, rlaica@fiec.espol.edu.ec², mmloor@fiec.espol.edu.ec³
sgarcia@espol.edu.ec⁴

Resumen

Este 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. El presente resumen especifica las ideas generales sobre el modelamiento, diseño e implementación de una aplicación de Realidad Virtual del juego de Lego. En primer lugar se hablará sobre la creación de los objetos y escenas en 3D, que van a constituir el entorno virtual. Posteriormente se detallará los algoritmos de posicionamiento y control creados para permitir la precisión y la lógica en la interacción hombre-máquina. Finalmente se presentarán los resultados obtenidos de las pruebas realizadas y las conclusiones y recomendaciones de los mismos.

Palabras Claves: mundo virtual, bloques de Lego, objetos 3D, Realidad Virtual.

Abstract

This project seeks by the use of the techniques of Virtual Reality to create a virtual world of a traditional game of Lego blocks, in order to check the applicability of the concepts learned throughout the topic. The present abstract specifies the general ideas on the modeling, design and implementation of an application of Virtual Reality of Lego blocks. First, we focus in the creation of the 3D objects and scenes, which are going to constitute the virtual environment. Later, we are going to detail the different positioning and control algorithms created to help us in the precision and the logic in the movements of the virtual objects. Finally, we present the results and the advantages and disadvantages of this kind of application against to a no immersive virtual reality application.

Keywords: virtual world, Lego blocks, 3D objects, Virtual Reality.

1. Introducción

La Realidad Virtual es un término que se ha popularizado en los últimos años, digno de la ciencia ficción, que ha cambiado la percepción 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”, que formaron parte del entretenimiento de muchas personas, quedarán solo como un buen recuerdo?

Entre estos juegos, destaca 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 distintas piezas para crear un mundo plástico inimaginable donde existían casas, personas y

objetos con formas extrañas, que solamente en nuestra mente tenían sentido.

Este 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.

En este artículo se resumen las técnicas de diseño y modelamiento aplicadas para la creación de los objetos 3D y el ambiente virtual, el análisis del funcionamiento de los dispositivos hápticos utilizados, además se detallan los métodos utilizados para lograr el control y la precisión en la interacción y los resultados que se obtuvieron en las pruebas realizadas.

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. 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 intuitivamente. Las tres características primordiales en Realidad Virtual son: La Presencia, la Inmersión y la Interacción.

PRESENCIA.- Es un concepto subjetivo y radica en el comportamiento de la psiquis del ser humano. Debido a los estímulos recibidos el usuario siente que encuentra en una situación real y se comporta de forma habitual.

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:

- 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.

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.

2.2 Sistemas inmersivos vs. 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. 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.

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.

3. Diseño de la aplicación

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

3.1 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 [5]:

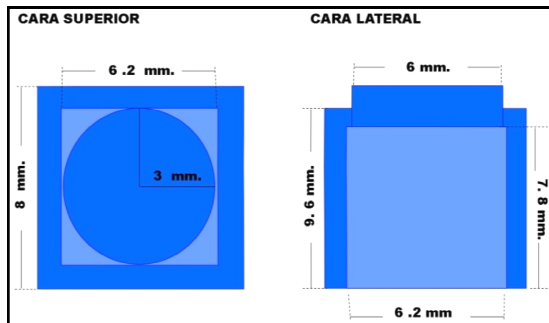


Figura 1: Dimensiones de los bloques de Lego

En el modelado de cada bloque recurrimos a operaciones booleanas para darle la forma deseada, respetando los criterios de diseño que habíamos establecido previamente.

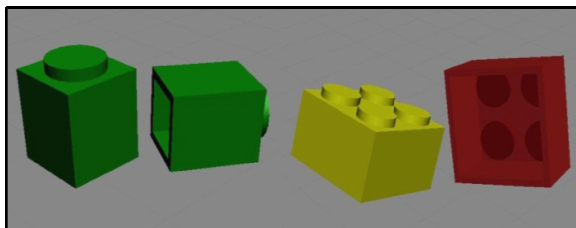


Figura 2: Legos modelados en 3D Studio Max

Una vez que finalizado cada objeto se procedió a convertirlo al formato de imagen adecuado (*osg*) para poder importarlo desde la plataforma con la que se iba a trabajar para que no se presentaran problemas de datos corruptos o pérdida de información gráfica importante tal como el color o la textura. Acto seguido se continuó con la segunda parte del diseño, la creación de la interfaz donde iban a estar

situadas estas piezas para que puedan ser manipuladas virtualmente.

3.2 Interfaz orientada a una aplicación semi - inmersiva

La interacción con la herramienta debía ser lo más transparente posible para el usuario, al ser totalmente inmersiva se debía tener libertad de movimiento ya que no existía la limitante de los cables y mayor campo de visibilidad debido al no estar frente a un monitor.

Dentro del ambiente virtual se podía acceder a un área de trabajo donde el usuario podía diseñar y construir cualquier objeto que deseara, haciendo uso de los bloques de Lego, que podrían ser manipulados mediante determinados dispositivos de realidad virtual, de los cuales hablaremos más adelante, de manera muy precisa y natural. Además se le daba la posibilidad al usuario de almacenar la escena creada para modificarla posteriormente.

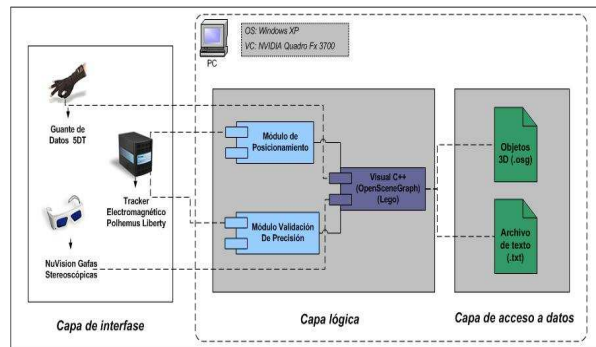


Figura 3: Diagrama general del Sistema.

4. Implementación y Pruebas

4.1 Software utilizado para la implementación

Los objetos y ambientes utilizados fueron modelados con el programa de creación de gráficos y animación 3d Studio Max 9. 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 las librerías con las que se procedía a trabajar la interface 3D.

La implementación de la aplicación inmersiva se realizó 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.

4.2 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 debe 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 3.25 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 guantes con un receptor inalámbrico con comunicación bluetooth, que permite al usuario manipular los objetos 3D en el entorno virtual.
- 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.
- 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.

4.3 Análisis para optimizar el control y precisión de los trackers

Los trackers electromagnéticos son populares, pero lamentablemente 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, 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.

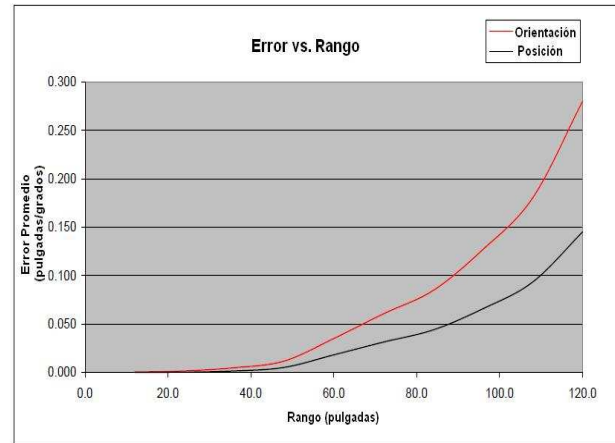


Figura 3: Error en desempeño de los trackers

La **Figura 3**, 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 en sensor del transmisor.

Tabla 1: Valores de error según rango del sensor

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

De acuerdo con la **Tabla 1**, se puede determinar que en un rango de hasta 50 pulgadas (127 cm.) aproximadamente, la precisión del equipo no se afecta, por el contrario cuando el sensor se ubica 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.

Dentro del proyecto trabajamos sobre un área de 75 cm de largo por 35 cm de ancho, con lo cual se podía asegurar que los valores obtenidos no habían sufrido alteraciones.

4.4 Análisis de algoritmos para corrección de error de calibración en los trackers

Se realizaron pruebas controladas dentro del laboratorio donde se determinó el área de desplazamiento en donde los valores obtenidos por medio de los sensores eran confiables.

En este tipo de tracker, la fuente producía tres campos electromagnéticos, cada uno de los cuales era perpendicular a los otros. El sensor colocado sobre el cuerpo del usuario medía la velocidad y dirección de ellos y enviaba esta información a la computadora, en donde se triangulaba 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 pudo inferir que el emisor del tracker, se comportaba como un transmisor omnidireccional y por ende presentaba características similares en la propagación de su señal.

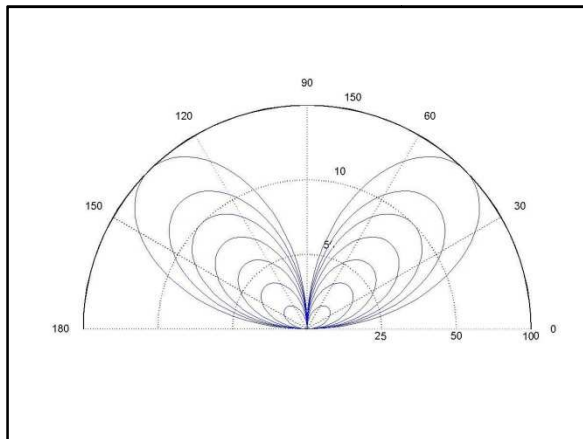


Figura 4: Propagación transmisor omnidireccional

De acuerdo a la **Figura 4**, existía un área donde no había cobertura para la adquisición de datos, aquí podían producirse datos aberrantes. Mediante el método de ensayo y error, se determinó que sobre los 8 cm. de alto los valores capturados eran confiables, ya que se encontraban 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 5**.

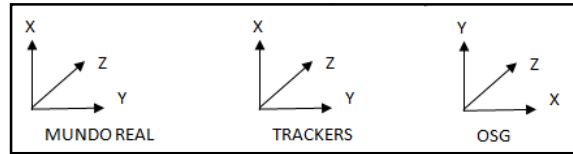


Figura 5: Sistema de Coordenadas de los espacios

En el espacio del tracker se definió el eje “x” hacia arriba, debido a que por sus características el emisor solo registraba valores positivos, y al estar apoyado el plano base sobre una superficie, se descartaba 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 definía el eje “y” hacia arriba, por lo cual al mapear una coordenada del espacio del tracker al del *osg* solo se realizaba un cambio de ejes.

Posteriormente, se seccionó el plano base trazando líneas separadas cada 5 cm sobre el eje “z”. Tomábamos un punto inicial (x_1, y_1, z_1) y desplazábamos en línea recta el sensor hacia el punto final (x_2, y_2, z_2) , los desplazamientos en “y” y “z” debían ser aproximados a 0. Se encontró que los valores obtenidos contenían cierta desviación con respecto al valor esperado, como se muestra en la **Tabla 2**.

Tabla 2: Transformación de coordenadas del mundo real al virtual

Punto	TRANSFORMACIÓN DE EJES DE COORDENADAS								
	MUNDO REAL (cm)			TRACKERS			OSG		
	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

Analizando los datos de la tabla, se estableció que para poder realizar un movimiento en línea recta se debía aplicar correcciones a los valores obtenidos en los ejes “y” y “z” del tracker. La **Figura 6** muestra las funciones de transformación de coordenadas necesarias para realizar éstas correcciones.

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

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

Figura 6: Función de transformación de coordenadas.

Una vez corregida la desviación generada en los valores del tracker, se seguía la siguiente secuencia de transformaciones que se muestran en la **Figura 7**, la que se realizaba marcando los cuatro extremos que conformaban el límite del área de trabajo, cuyos valores se muestran en la **Tabla 3**.

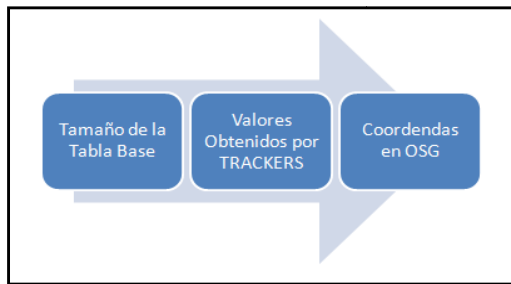


Figura 7: Orden de transformaciones realizadas

Tabla 3: Valores de factor de corrección de los trackers

cm(z)	FACTOR DE CORRECCION TRACKERS								
	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
5	2.00	8.90	2.90	4.36	28.69	3.67	2.36	0	0.77
10	2.00	8.90	4.62	5.09	28.77	5.38	3.09	0	0.76
15	2.00	8.90	6.5	5.66	28.53	6.95	3.66	0	0.45
20	2.00	8.90	8.44	6.13	28.44	8.30	4.13	0	0.14
25	2.00	8.90	9.78	6.47	28.36	9.70	4.47	0	0.08

Entonces para cada punto del espacio real (mr) dentro de las dimensiones de la tabla base, le correspondía 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 transformaba 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 correspondía 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 quedaba de la siguiente manera:

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

4.5 Calibración de dispositivos hápticos

El guante de datos en conjunción con los trackers conformaban 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 8: Valores, gestos y acciones del guante de datos

La **Figura 8**, muestra una referencia de los valores de los gestos de la mano según el fabricante, 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 realizaba la calibración de la mano del usuario, en donde se pedía que haga un determinado gesto asignándole el valor de mayor frecuencia.

4.6 Descripción de pruebas

El objetivo que se quería 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 podría 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 era el más complicado de realizar, ya que aquí se debía tomar en consideración los retardos propios de los dispositivos que incidían en la suavidad del movimiento, lo que provocaba que lo que hacíamos en el mundo real no se vea reflejado 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 eran reflejados en el entorno virtual. Además se le pedía que se ubicara en determinadas áreas para verificar que la aplicación respondiera como era deseado, como por ejemplo si cuando era ubicado un bloque fuera del tablero este era eliminado; o que 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 funcionaba correctamente, pedíamos al usuario que seleccionara aleatoriamente ciertos bloques que ya habían sido colocados previamente, y que 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 borrara el área de trabajo, y con el tracker se dirigiera al cajón con el ícono de abrir para cargar la escena almacenada previamente.

De esta forma verificábamos que la aplicación respondía adecuadamente.

5. Conclusiones

Una vez que se finalizó con las etapas de implementación y pruebas de la aplicación virtual, se pudo concluir que:

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.

Mediante un análisis minucioso de las características y funcionamiento de los trackers, se pudo remediar los errores obtenidos en la captura de los datos de posicionamiento, los mismos que provocaban que la aplicación sea muy inestable y difícil de manejar.

Para solucionar la falencia que presentaba el tracker al no introducir valores negativos de "x", se aprovechó el



ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

CENTRO DE INVESTIGACIÓN CIENTÍFICA Y TECNOLÓGICA



hecho de que en la 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.

En la aplicación, se puede 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.

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 se hace en la realidad.

De acuerdo a los resultados obtenidos, para este tipo de proyectos, el control y la precisión de los movimientos por medio de los dispositivos es 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.

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.

6. Recomendaciones

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

Para poder contribuir con la característica de inmersión, para futuros proyectos, 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.

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, así como también en lograr movimientos más naturales e intuitivos para comodidad del usuario.

Para mayor comodidad se recomienda la utilización de dispositivos totalmente inalámbricos, ya que muchas veces los usuarios no se enfocan en el mundo virtual debido a que se encuentran atados a cables.

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.

7. Referencias

- [1] 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].
- [2] CARDONA, Jesús David. "*Realidad virtual y entornos virtuales*". Colombia: Universidad Autónoma de Occidente, 2006.
- [3] GUTIERREZ, Mario; Vexo Frédéric y Thalmann Daniel. "*Stepping into virtual Reality*". Inglaterra: Springer, 2008. ISBN: 978-1-84800-116-9
- [4] ESCARTIN, Emilio. "*La Realidad Virtual, Una tecnología educativa a nuestro alcance*". Cuba: ISPJAE, 2000.
- [5] lego.com
- [6] 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.
- [7] ORTIZ, Oscar. "*Interacción y usabilidad de interfaces en realidad virtual*". España: Universidad Rey Juan Carlos, 2003.
- [8] HAYT, William; Buck, John. "Teoría Electromagnética". Mexico: McGraw-Hill, 2006. ISBN: 970-105620-5
- [9] SERWAY, Raymond. "*Física para ciencias e ingeniería*". Mexico: McGraw-Hill, 2002. ISBN: 970-103581-X
- [10] [Tech compare with ResolutionvsRange graphs.pdf](http://www.polhemus.com). [en línea]. Disponible en: <www.polhemus.com>
- [11] [Data Glove Ultra-Manual.pdf](http://www.5dt.com). [en línea]. Disponible en: <www.5dt.com>
- [12] OpenSceneGraph, 3D Graphics Toolkit, <http://openscenegraph.org>, 12 de julio de 2009.
- [13] VILLAREAL, Silvia. "*Los juegos didácticos como factores de desarrollo de comunidades de aprendizaje*". México: Universidad Autónoma de México, 2002.

Dr. Sixto García
PROFESOR DE MATERIA DE GRADUACIÓN