



Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería en Ciencias y Sistemas

**LA REALIDAD VIRTUAL APLICADA AL TRATAMIENTO DE LA
CLAUSTROFOBIA**

Carlos Antonio Mancio Reyes

Asesorado por el Ing. Edgar Roberto Pinillos

Guatemala, noviembre de 2007

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**LA REALIDAD VIRTUAL APLICADA AL TRATAMIENTO DE LA
CLAUSTROFOBIA**

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA

POR:

CARLOS ANTONIO MANCIO REYES

ASESORADO POR EL ING. EDGAR ROBERTO PINILLOS

AL CONFERÍRSELE EL TÍTULO DE
INGENIERO EN CIENCIAS Y SISTEMAS

GUATEMALA, NOVIEMBRE DE 2007

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA



NÓMINA DE JUNTA DIRECTIVA

DECANO	Ing. Murphy Olympo Paiz Recinos
VOCAL I	Inga. Glenda Patricia García Soria
VOCAL II	Inga. Alba Maritza Guerrero de López
VOCAL III	Ing. Miguel Ángel Dávila Calderón
VOCAL IV	Br. Kenneth Issur Estrada Ruiz
SECRETARIA	Inga. Marcia Ivonne Véliz Vargas

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

DECANO	Ing. Murphy Olympo Paiz Recinos
EXAMINADOR	Inga. Virginia Victoria Tala Ayerdi
EXAMINADOR	Ing. Edgar Eduardo Santos Sutuj
EXAMINADOR	Ing. Ludwing Federico Altán Sac
SECRETARIA	Inga. Marcia Ivónne Véliz Vargas

HONORABLE TRIBUNAL EXAMINADOR

Cumpliendo con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

LA REALIDAD VIRTUAL APLICADA AL TRATAMIENTO DE LA CLAUSTROFOBIA,

tema que me fuera asignado por la Dirección de la Escuela de Ingeniería en Ciencias y Sistemas, con fecha agosto de 2006.

Carlos Antonio Mancio Reyes

Carta asesor

Carta revisor de trabajos de graduación

Carta director de la escuela

Carta decano de la facultad

ACTO QUE DEDICO A:

Mis padres

Mi más grande orgullo, por todo el amor y paciencia que me han brindado durante mí existir.

Mis hermanas

Por lo todos los maravillosos momentos que hemos pasado juntos, y por la inspiración que cada una me dio para llegar hasta aquí el día hoy.

Mis amigos

Rafa y Allan, por las innumerables veces que me brindaron su mano fraternal.

Andreita

Por todo el cariño, apoyo y comprensión que me ha brindado.

Mi familia

ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES	V
GLOSARIO	IX
RESUMEN	XI
OBJETIVOS.....	XIII
INTRODUCCIÓN.....	XV
1 REALIDAD VIRTUAL	1
1.1 Descripción de realidad virtual	1
1.2 Clasificación de la realidad virtual.....	5
1.3 Características de la realidad virtual	6
1.4 Historia de la realidad virtual.....	7
1.4.1 Marco temporal de la realidad virtual.....	7
1.5 Plataformas.....	10
1.6 Periféricos.....	12
2 LA REALIDAD VIRTUAL A BAJO NIVEL	15
2.1 Gráficas en computadora.....	15
2.1.1 Primitivos de salida.....	15
2.2 Conceptos tridimensionales	17

2.3	Métodos de despliegue tridimensional	18
2.3.1	Proyección paralela.....	18
2.3.2	Proyección de perspectiva	19
2.4	Modelado en 3D	20
2.4.1	Software para modelado en 3D.....	20
2.4.1.1	AC3D.....	21
2.4.1.1.1	Creación de objetos en AC3D	23
2.4.1.2	OPENGL.....	25
2.4.1.3	GLUT	27
2.4.1.3.1	Dibujando con GLUT.....	27
2.4.1.4	VRML.....	40
3.	FOBIAS	53
3.1	Definición.....	53
3.2	Claustrofobia	54
4.	APLICACIÓN DE LA REALIDAD VIRTUAL.....	63
4.1	Aplicaciones de la realidad virtual en psicología	63
4.2	Tratamiento de fobias mediante realidad virtual	63
4.3	Tratamiento de la claustrofobia	65
4.4	Escenarios virtuales de un programa para el tratamiento de la claustrofobia	67
4.4.1	Primer escenario, la casa.....	67
4.4.2	Segundo escenario, el ascensor	73

4.5	Aspectos informáticos de programación	77
4.5.1	3Dive	78
4.5.2	Python	79
4.5.3	OpenGL.....	79
4.6	Incorporación de dispositivos de realidad virtual a la clínica.....	79
4.6.1	Gafas de realidad virtual i-glases:	80
4.6.2	Gafas de realidad virtual Sony Glasstron:	81
	CONCLUSIONES	83
	RECOMENDACIONES	85
	BIBLIOGRAFÍA	87

ÍNDICE DE ILUSTRACIONES

FIGURAS

1.	Proyección paralela	19
2.	Proyección en perspectiva	19
3.	Creación de objetos en AC3D	24
4.	Configuración de propiedades de los objetos recién creados	24
5.	Instrucciones para crear una ventana en OpenGL	29
6.	Una ventana simple en OpenGL	29
7.	Declaración de una función de despliegue en OpenGL	30
8.	Función de captura de teclas normales en OpenGL	32
9.	Función de procesamiento de teclas especiales en OpenGL	33
10.	Distribución de la memoria en OpenGL	36
11.	Especificación de un triángulo en OpenGL	38
12.	Un triángulo en OpenGL	39
13.	Comentarios en VRML	41
14.	Definición de un nodo simple	42
15.	Definición de un nodo anidado	42
16.	Creando un nodo de ejemplo	42

17.	Apariencia y geometría de los nodos	43
18.	Transformaciones en los nodos	43
19.	Cubo con textura en VRML	45
20.	Ejemplo de un cubo en VRML	46
21.	Despliegue de texto en VRML	47
22.	Escritura de texto en VRML	47
23.	Definición de cámaras en la escena	48
24.	Generación de luz mediante DirectionalLight	50
25.	Generación de luz mediante PointLight	50
26.	Generación de luz mediante SpotLight	51
27.	Distintos tipos de iluminación en un mismo objeto	51
28.	Especificaciones del navegador	52
29.	Modelo explicativo de la claustrofobia	60
30.	Primer escenario la sala	67
31.	Iluminación alta en la sala, con la persiana abierta	68
32.	Sala con poca iluminación por estar cerrada la persiana	68
33.	Situación medioambiental de buen tiempo	70
34.	Resultado de mal tiempo y de rayos en la escena	70
35.	Iluminación resultante de abrir o cerrar la puerta	71
36.	La habitación mágica en dos variantes de su tamaño	72

37.	El vestíbulo	73
38.	Selección de nivel en el ascensor	74
39.	Desperfecto en el ascensor	75
40.	Gafas de realidad virtual i-glasses y HDM Sony Glasstron	80

TABLAS

I	Tipos de datos en OpenGL y sus sufijos	26
II	Lista de primitivos de GLUT	39

GLOSARIO

<i>Chat</i>	Del anglosajón charlar, y que se utiliza para describir la comunicación simultánea entre dos o más personas a través de Internet.
Deflexión	Desviación de la dirección de una corriente.
<i>Et al</i>	Término latín que significa y su grupo, o y compañía, o bien y los otros.
Fobia	Violenta angustia, que aparece compulsivamente, ante determinados objetos o situaciones exteriores que realmente no presentan peligro.
<i>Hardware</i>	El hardware se refiere a todos los componentes físicos de que se compone una computadora.
Periféricos	Son las unidades o dispositivos a través de los cuales la computadora se comunica con el mundo exterior.

Plataforma	En informática, una plataforma es precisamente el principio, ya sea de hardware o software, sobre el cual un programa puede ejecutarse.
Resolución	Es el número máximo de <i>píxeles</i> , que pueden ser mostrados en la pantalla.
Software	Se denomina así a todos los componentes intangibles de una computadora, es decir, al conjunto de programas y procedimientos necesarios para hacer posible la realización de una tarea específica.
Tecnologías de información	Se refiere al uso de computadoras y del <i>software</i> electrónico de convertir, de almacenar, de proteger, de procesar, de transmitir y de recuperar la información.
Tiempo real	Característica de procesamiento de los sistemas en el tiempo mismo en que se suscita una acción o evento.

RESUMEN

La integración de distintas tecnologías tales como el audio, el video, la cinética, entre otras convierten a la realidad virtual, en una técnica efectiva y además económica para el tratamiento de distintos trastornos mentales, como las fobias.

Se entiende por realidad virtual toda aquella tecnología que logra sintetizar un mundo tridimensional ficticio, desarrollado en una computadora, creando en el usuario la ilusión de realidad, y considerada mejor dependiendo del grado de inmersión que logre producir.

En la actualidad esta tecnología emergente ya ha dado solución al tratamiento de varios trastornos mentales tales como, fobia a volar, a las arañas, al encierro, a los espacios abiertos, a desordenes alimenticios, entre otros.

En cuanto al tratamiento de la claustrofobia, la realidad virtual provee de mecanismos de replicación las situaciones más comúnmente temidas, a las que además de la ansiedad que ya de por sí provoca en la persona claustrofóbica, mediante la realidad virtual pueden agregársele distintos matices que multipliquen el miedo de la persona en relación de uno a diez.

Los paquetes de computadora para el tratamiento de la claustrofobia, cuentan con una serie de escenarios entre los que se puede mencionar, una sala de estancia, un elevador y algunos agregan una habitación mágica; la cual consiste en pequeño cuarto que a la orden del terapeuta puede reducir el espacio además de la emisión de un estruendoso sonido.

Mediante la exposición del paciente a estas situaciones, gradualmente puede llegarse a reducir su nivel de ansiedad a valores cercanos a cero, aun y cuando las técnicas tradicionales de exposición brindan prácticamente los mismos resultados, el tratamiento por realidad virtual, puede acortar el tiempo de mejora gracias a la sensación de seguridad que brinda al paciente.

Como contraparte a los paquetes de computadora existe una gran cantidad de dispositivos que pueden emplearse para aumentar el grado de inmersión de los pacientes tratados con esta técnica, sin embargo se puede llegar a un grado aceptable mediante el uso de unos pocos periféricos, tales como una buena tarjeta aceleradora grafica, un buen sistema acústico, unas gafas y un método de aislamiento que inhiba los estímulos externos al escenario virtual.

OBJETIVOS

- **General**

Evidenciar el alto grado de involucramiento que han alcanzado las tecnologías de información en las ciencias del cuidado de la salud mental.

- **Específicos**

1. Establecer por medio de un estudio detallado los pasos a seguir en el tratamiento de la claustrofobia por medio de realidad virtual.
2. Que sirva como referencia y apoyo para los psicólogos clínicos que deseen incursionar en este nuevo método de terapia.
3. Proveer una guía técnica para los implementadores de tecnología interesados en desarrollar herramientas en este campo de aplicación médica.
4. Sugerir posibles escenarios virtuales que pueden desarrollarse para el tratamiento de la claustrofobia.
5. Describir las actuales herramientas de desarrollo de *software* que pueden utilizarse para el desarrollo de paquetes de realidad virtual.
6. Citar el equipamiento electrónico computacional que debe emplearse para acrecentar el grado de inmersión en los mundos virtuales.

INTRODUCCIÓN

En la actualidad, la utilización de las tecnologías de información ha penetrado en casi todos los puestos de trabajo, así como en muchas otras actividades de la vida cotidiana, a través de tecnologías como internet, correo electrónico, *chats*, *etc.*

Los especialistas del cuidado de la salud física y mental pueden beneficiarse también del uso de computadoras y otras tecnologías de la información, tanto para el diagnóstico y tratamiento, como para la medición de la mejora de desordenes en sus pacientes.

De hecho las computadoras están siendo usadas en diversas formas por profesionales del cuidado de la salud mental; y una de las formas más creativas incluyen la utilización de realidad virtual para simular las situaciones a las que una persona teme, este tipo de programas fácilmente pueden producir la sensación de estar en un lugar más alto, o uno más estrecho o simplemente aparentar que un grupo de enormes arañas se está acercando amenazantemente. La realidad virtual es entonces una tecnología que intenta engañar a los sentidos humanos para hacerlos creer que se encuentran en un mundo que sólo existe dentro de una computadora.

La utilización de entornos virtuales en una clínica médica de tratamiento de fobias ha cobrado viabilidad gracias a los excelentes resultados que ha demostrado en la recuperación de estos desordenes mentales.

Además el tratamiento con estos escenarios provee de diversas ventajas difícilmente replicables por otras formas de tratamiento.

Entre las ventajas del empleo de técnicas de exposición virtual puede mencionarse, el control completo del entorno por parte del terapeuta, tanto como para aumentar la situación temida por el paciente, para así estimular su ansiedad ante el objeto o situación que produce la fobia, así como la aislación que puede proveérsele resguardando con esto la privacidad del paciente.

Otra ventaja irrefutable es el bajo costo comparativo de esta tecnología versus la exposición in vivo, máxime en tratamientos de fobias a volar, o fobias a determinados animales como por ejemplo arañas.

Este informe presenta los distintos escenarios de que se compone un entorno virtual para el tratamiento de la claustrofobia, describiendo las características con que cuenta cada uno, así como detalles técnicos sobre las distintas herramientas y/o tecnologías informáticas para el desarrollo de los mismos.

1 REALIDAD VIRTUAL

1.1 Descripción de realidad virtual

El avance en los últimos tiempos de las tecnologías de información, ha dado lugar a la creación y divulgación generalizada del término realidad virtual, pero debido a que se trata de una tecnología emergente, cualquier definición no puede aun ser tomada formalmente, empero, en la mayoría de contextos puede aceptarse que la realidad virtual es una simulación realizada en una computadora, con gran contenido gráfico, acústico y de otras índoles como pudieran ser características susceptibles al tacto, al olfato y al movimiento; para que los usuarios se sumerjan en éstas simulaciones es necesaria la utilización de equipo especializada, como lo son cascos, guantes, sensores de posición etc.

Los primeros nombres que se mencionaron para esta tecnología emergente fueron, de primera mano, realidad artificial, dado por Myron Krueger a mediados de los años setenta, y refleja un ambiente informático que responde a la manipulación humana. El termino realidad artificial sigue siendo igualmente aceptado por la contradicción en que se cae con realidad virtual, puesto que algo no puede ser real y a la vez virtual.

Por aparte el término realidad virtual fue puesto en circulación a finales de los años ochenta por Jaron Lanier, para distinguir entre la inmersión en mundos digitales que se crean con esta tecnología y las simulaciones tradicionales desarrolladas en una computadora.

Los principios de la realidad virtual se pueden enmarcar en los programas de entrenamiento militar, simuladores de vuelo, y centros de investigación, hasta llegar a ubicar los programas más variados en el ámbito profesional y doméstico. Hoy por hoy se aplica en ingeniería, medicina, arquitectura, educación, juegos, entre otros. Los entornos virtuales son mundos privados, es una fantasía en la que los participantes poseen libertad absoluta, es decir no se les puede exigir ninguna responsabilidad debido a su naturaleza asocial.

Otra definición altamente difundida para realidad virtual consiste en describirla como un sistema participativo, que materializa de manera tridimensional un entorno posiblemente imaginario. Dicha materialización es lograda a través del empleo de técnicas panorámicas de fotografía, como lo son las fotografías de trescientos sesenta grados, con las que el usuario puede seleccionar si ver arriba, abajo, a la izquierda o derecha del mundo virtual, dándole con esto la interactividad necesaria para recrear en él, la sensación de presencia.

En el ámbito informático, lo virtual, representa todas aquellas creaciones desarrolladas en una computadora que pretenden emular una actividad, un panorama, o en general una situación del mundo real, con el ánimo de cumplir determinada tarea.

Y es por esto que muchos autores la consideran como eslabón que afinara una comunicación natural entre los humanos y las máquinas.

Aunque de entre todas las definiciones de realidad virtual, la más estricta es la de del Pino González (1995) quien asevera que un sistema de RV (realidad virtual) es un sistema interactivo que permite sintetizar un mundo tridimensional ficticio creando en el usuario una ilusión de realidad.

De esta definición el autor enfatiza en cuatro características:

- La capacidad sintética: el mundo virtual se genera en tiempo real según sea alimentado por las instrucciones y/o acciones del usuario.
- La interactividad: El entorno responde a las acciones y movimientos del usuario.
- La tridimensionalidad: El entorno se desenvuelve en cualquier dispositivo de despliegue, imitando un mundo en tres dimensiones.
- La ilusión de realidad: Los entornos que generan los sistemas de realidad virtual no son reales, sino más bien ficticios, pero son definidos, de tal modo que al usuario le parezcan reales.

En resumen, la descripción de realidad virtual puede variar dependiendo del enfoque informático que se aplique, pero puede describirse como toda aquella tecnología de información que genera escenarios tridimensionales, creados en base a las acciones que realice el usuario, entendiéndose en tiempo real, para que con esto a su vez se produzca en el usuario un sentimiento de inmersión, que más que asemejar, recree la sensación presencia en el mundo real.

Las piedras angulares de los sistemas de realidad virtual son la presencia y la interacción. Dependiendo del nivel de realismo que posean los entornos virtuales generados, los usuarios pueden llegar sentirse más que simples espectadores y sentir que forman parte de estos.

Este hecho hace pensar a los expertos en la materia que la representación de situaciones clínicamente fóbicas mediante la realidad virtual puede ser empleada como un paso intermedio entre la exposición *in vivo* y la imaginaria para pacientes con desordenes mentales.

Las partes que componen un sistema de realidad virtual pueden ser de diversa índole. Uno de los dispositivos de salida más inmersivos que se conocen y utilizan hoy en día ya que reduce la incomodidad y dificultad de los famosos cascos de realidad virtual, es el *CAVE*. Un *cave* es un cuarto de cuatro paredes en donde se proyecta el entorno virtual, y además de proyectarse el entorno en las paredes este es proyectado en el suelo y techo de la habitación, por lo que, al estar el usuario envuelto totalmente por la proyección de las imágenes, su sensación de presencia en el mundo virtual es notablemente aumentada.

Los componentes auditivos de los sistemas de realidad virtual han sido menospreciados hasta la fecha, por los desarrolladores de mundos virtuales, y los esfuerzos se han centrado únicamente en los sistemas visuales, restringiéndose únicamente a proporcionar salidas con características de estéreo. Situación que empieza a cambiar gracias a las posibilidades de los equipos de audio multicanal.

Con salidas auditivas en varios canales se proporciona al usuario señales sonoras que lo llevan a sentir de manera inequívoca la localización de fuentes de sonido en el espacio, incrementando también así notablemente su sensación de presencia.

1.2 Clasificación de la realidad virtual

La realidad virtual puede clasificarse en dos clases: la realidad virtual inmersiva y la no inmersiva.

En general puede decirse que la realidad virtual inmersiva es aquella que utiliza gran cantidad de dispositivos sofisticados para la interacción con el usuario, tales dispositivos pueden clasificarse según su naturaleza en dispositivos de despliegue, de posicionamiento, táctiles y de resistencia al movimiento.

Por aparte la realidad virtual no inmersiva, es aquella tecnología que provee al usuario la posibilidad de situarse virtualmente en ambientes en los que físicamente no se encuentra, esto gracias al auxilio de otras tecnologías como por ejemplo internet. Gracias al internet y programas tipo *chat* el usuario puede sostener conversaciones con gente en distintas localizaciones geográficas sin la necesidad de adherir dispositivos especiales a su computadora.

Debido a sus características de simplicidad La realidad virtual no inmersiva posee varias ventajas sobre la inmersiva tales como bajo costo, rápida y fácil adecuación de los usuarios.

Además de que los dispositivos inmersivos como cascos, guantes y rastreadores de posición son caros, el usuario generalmente prefiere interactuar en entornos virtuales con dispositivos a los que ya está adaptado como lo son un teclado o un ratón de computadora.

1.3 Características de la realidad virtual

Aunque como se mencionó previamente la realidad virtual es una tecnología en actual evolución, por lo cual no se han establecido estándares estrictos, sí pueden enumerarse ciertas características con las que debe cumplir como lo son, según lo indica Velez Janh(1993) , en su texto de introducción a la realidad virtual:

- Responde a la metáfora de mundo que contiene objetos y opera en base a reglas de juego que varían en flexibilidad dependiendo de su compromiso con la inteligencia artificial.
- Se expresa en lenguaje gráfico tridimensional.
- Su comportamiento es dinámico y opera en tiempo real.
- Su operación está basada en la incorporación del usuario en el interior del medio computarizado.
- Requiere que, en principio haya una interrupción de la incredulidad como recurso para lograr la integración del usuario al mundo virtual al que ingresa.
- Posee la capacidad de reaccionar ante el usuario, ofreciéndole, en su modalidad más avanzada, una experiencia inmersiva, interactiva y multisensorial.

1.4 Historia de la realidad virtual

Para una mejor comprensión de la evolución que ha presentado a través de su surgimiento se presenta a continuación la línea de tiempo que ha establecido la realidad virtual.

1.4.1 Marco temporal de la realidad virtual

Los inicios de la realidad virtual pueden ubicarse, según algunos autores, a distintas épocas, pero todos coinciden en que el precursor más importante es la industria del cine.

Desde siempre el séptimo arte ha intentado crear formatos de imagen y sonido que hagan creer al espectador que se encuentra formando parte de la escena, y es de este intento que han surgido tecnologías como el *Cinemascope* o los más modernos *Omnimax* y el *Imax*, así como sistemas de sonido del tipo del *Dolby Surround* y *DTS* entre otros.

Según el escrito titulado Estado del Arte de las Tecnologías de realidad virtual, de C. Bescós del Castillo, *et al* (2001). La evolución de la realidad virtual puede dividirse en cuatro etapas:

Primera etapa: -años 60-. Se desarrolla principalmente en investigaciones militares en Estados Unidos de Norte America. Se realizan experimentos de simulación para el entrenamiento militar en colaboración con entidades como la NASA (*National Aeronautics and Space Administration*).

Segunda etapa: -años 70-. Se traslada el campo de investigación a las universidades de Estados Unidos. Se consiguen prototipos de *hardware* y grandes mejoras en el procesamiento gráfico.

Tercera etapa: -años 80-. Los sistemas comienzan a ser competitivos comercialmente y la industria toma parte en las tecnologías virtuales. Empresas multinacionales de software y hardware desarrollan y comercializan equipos para todo tipo de aplicaciones. Se extiende el área de influencia a Europa, principalmente Inglaterra, Alemania, por medio de las aplicaciones de entrenamiento aéreo-espacial.

Cuarta etapa: Actualmente.

La realidad virtual tal como la conocemos hoy en día dio sus primeros pasos a finales de los años setenta como parte del programa preparación de pilotos del departamento de defensa de los Estados Unidos.

Haciendo simulaciones de vuelo, para así reducir costos al no utilizar los aviones reales, ahorrando combustible, y desgaste de las aeronaves, además de la potencialidad de aumentar exponencialmente las horas de práctica de los pilotos gracias a la nula peligrosidad de realizar estos entrenamientos en cabinas de realidad virtual.

Luego de esto en 1982 Scott Fisher creó el Visio Casco en la Nasa Ames Center, el cual consistía en ubicar dos pequeñas pantallas, las cuales quedan justo frente a los ojos de quien lo porta, inhibiendo con esto los estímulos visuales del medio externo. En este punto las compañías vieron la oportunidad de lucro con la tecnología en auge y se crea la siguiente línea temporal:

- 1980: La Compañía StereoGraphics hace las gafas de visión estéreo.
- 1982: Thomas Zimmerman patentiza el electroguante que inventó mientras investigaba sobre cómo controlar con la mano un instrumento musical virtual.
- 1987: La compañía Inglesa Dimensión Internacional desarrolla un *software* de construcción de mundos tridimensionales sobre PC¹.
- 1988: Scott Foster inventa un dispositivo para la generación de sonido tridimensional.
- 1989: *ATARI* saca al mercado la primera máquina de galería de vídeo juegos con tecnología 3D². En ese mismo año *Autodesk* presenta su primer sistema de realidad virtual para PC.

A más detalle histórico puede decirse que la exploración en el campo de la realidad virtual se inició en la Universidad de Harvard en el año 1968, principalmente por el profesor Ian Sutherland, quien con un grupo de investigadores diseñó el primer casco de realidad virtual, al que denominaron *Incredible Helmet* (casco increíble). El casco poseía sensores de posición que se hallaban incrustados en el techo y que servían para traducir los movimientos de la cabeza a desplazamientos de unos calibradores, cuya posición era detectada por la computadora.

El personaje más reconocido en la tecnología de la realidad virtual es Jaron Lanier, quien creó el primer guante de datos y es el fundador de la empresa *VPL Research* junto con Thomas Zimmerman.

¹ *Personal Computer* (Computadora Personal)

² 3D – Tres Dimensiones

Su guante era tan funcional y práctico que empezó a venderse en poco tiempo a instituciones tales como la NASA y el Pentágono. Quienes lo emplearon para realizar pruebas de control a distancia de robots, el pentágono en robots de reconocimiento militar y antibombas, y en la NASA la idea era que un astronauta en Marte enviara un vehículo para investigar un cráter que pareciera peligroso. O bien, los astronautas podrían usar los guantes de datos de la RV para hacer reparaciones fuera de la nave espacial.

1.5 Plataformas

La plataforma de un sistema de realidad virtual, principalmente se conforma por una computadora en su definición formal (Unidad central de procesamiento, memoria y buses) a la que se conectan periféricos de entrada y salida. Entre los periféricos más comunes se encuentran las tarjetas aceleradoras gráficas, y los elementos de simulación y síntesis de imágenes.

Entre las arquitecturas más utilizadas para reducir los tiempos de procesamiento gráfico y de simulación se encuentran las de tipo *pipe-line*³, y las arquitecturas en paralelo. La arquitectura *pipe-line* o llamada también basada en filtros se fundamenta en descomponer un flujo en varias etapas, en la que la salida de una alimenta las entradas de la siguiente, siendo la entrada de cada una la salida de la anterior; y las arquitecturas en paralelo como su nombre lo indica ejecutan varias tareas teóricamente en un mismo instante.

³ Se traduce como segmentación y consiste en descomponer la ejecución de cada instrucción en varias etapas para poder empezar a procesar una instrucción diferente en cada una de ellas y trabajar con varias a la vez.

El procesado de imágenes y de simulación consiste en generar o regenerar, según el caso, la imagen desplegada, a partir de los datos colectados por los dispositivos de localización, tales como rastreadores de posición, o bien, reconfigurar el estado actual del mundo virtual, con el fin de que se adecúe a las nuevas condiciones propuestas por usuario.

En cuanto al hardware para la ejecución de entornos virtuales se han venido utilizando distintas plataformas, en lo que destacan las estaciones graficas, que son plataformas de elevadas prestaciones para el procesado gráfico, que es la parte que consume la mayor cantidad recursos en los entornos virtuales. Organizaciones como *Sillicon Graphics* se fundaron con este enfoque de mercado. Hoy en día, debido al descenso en los costos y al aumento de su rendimiento, se unen a la competencia las computadoras de escritorio, que son plataformas genéricas a las que se les agrega tarjetas específicas para mejorar prestaciones en tareas específicas. *Sun Microsystems*, *Hewlett-Packard*, *IBM* y *MAC*, aunque este último estaría en discusión por ser sus precios sensiblemente mayores a los de los anteriores, son fabricantes ubicadas en este mercado del mundo virtual; permitiendo que con la adecuada tarjeta aceleradora gráfica se desarrollen entornos virtuales de alta calidad, en estas sencillas plataformas.

Otra plataforma genérica altamente difundida son las consolas de videojuegos.

Las consolas de video juegos en general se auxilian de las invenciones realizadas por los fabricantes antes indicados; por ejemplo el *Xbox 360* del gigante *Microsoft*, cuenta con un procesador diseñado por *IBM*, una tarjeta procesadora grafica fabricada por *ATI* y un sistema de periféricos diseñado por *SIS*.

1.6 Periféricos

En términos generales existen periféricos de entrada y periféricos de salida, los de entrada se concentran en dispositivos de rastreo de posición o bien dispositivos de localización, que son los que definen la situación y posición del usuario, y los de salida que se encargan de representar las instrucciones que generen los dispositivos de entrada.

El avance que ha tenido la electrónica ha propiciado la existencia de infinidad de dispositivos de control, los que van desde sencillos ratones 3D, teclados, *joysticks*, pedales, timones, hasta electroguantes o sistemas bioeléctricos. Entre los avances de los sistemas bioelectrónicos resalta el experimento realizado por Raphael Holzer, que consistió en fijar un dispositivo electrónico en una cucaracha, luego de haber sustituido sus antenas por electrodos y de haberlos unido al sistema nervioso del insecto, lo que volvió posible dirigir la cucaracha por control remoto.

Por aparte los periféricos de salida se dividen en dispositivos de presentación de imágenes, de reproducción de audio, de retroalimentación táctil⁴ o cinestésica⁵ y plataformas móviles, que son cabinas con capacidad de variar su ángulo de inclinación dependiendo de la situación que se le presenta en ese instante al usuario. Como es obvio, en un entorno virtual se puede obtener mayores niveles de inmersión gracias a los dispositivos de presentación gráfica, por lo que pueden considerarse estos los de mayor importancia.

⁴ Consiste en proveer al usuario de sensaciones de frío-calor, suavidad, rugosidad, etc.

⁵ Resistencia a la flexión de la mano.

A estos dispositivos se les puede clasificar según su tipo de prestaciones en: monitores, gafas estereoscópicas, sistemas binoculares y visiocascos o HMD (*Head Mounted Display*).

Monitores: Actualmente la mayor parte de dispositivos de salida de un sistema gráfico de computadora son los tradicionales y bien conocidos monitores. Actualmente hay dos tipos de monitores que son más utilizados; Los de antiguos monitores de transistores y los de cristal líquido, los primeros son los que se basan en el diseño estándar de tubo de rayos catódicos (CTR por sus siglas en inglés)⁶, que funciona básicamente por la emisión de electrones que chocan contra una pantalla cubierta de fosforo que al ser contactado por los electrones generan un haz de luz en el punto en que colisionaron.

La otra tecnología muy utilizada, como se mencionó, son las pantallas LCD⁷, estas tienen una matriz compuesta por pequeños transistores (diodos emisores de luz) que tienen la capacidad de transmitir luz cuando pasa una corriente a través de ellos.

Esta tecnología tiene muchas ventajas sobre los tubos de rayos catódicos, en lo que se refiere a espacio, baja temperatura y calidad de presentación de imagen, los LCD no emiten radiaciones electromagnéticas, y al parecer la única desventaja es el costo de los monitores LCD, el cual es significativamente mayor que los CRT.

Gafas estereoscópicas: Son una especie de lentes que mediante diversas tecnologías se encargan de filtrar la imagen para cada ojo.

⁶ CRT – Cathodic Ray Tube

⁷ LCD – Panel de cristal líquido

Esto a partir de una pantalla en la que se proyectan alternadamente y a gran velocidad las imágenes en el ojo izquierdo y derecho.

Sistemas binoculares y visiocascos o HDM: Son dispositivos de video (visores) bidimensionales, ergonómicos, ligeros, en color, de alta definición. Estos dispositivos están montados sobre una armazón similar al de los lentes de uso ordinario (llamados dispositivos de colocación cefálica o *head mounted display*). Su peso actual es de 75 gr., aproximadamente. Los visores tienen un microsistema óptico especialmente diseñado para el uso ambulatorio y un sistema de color corregido por refracción y magnificación, de tal forma que a pesar de usar una mini pantalla de video, dan la percepción visual de tener una pantalla grande equivalente a una pantalla de 25 cm observada a 1 metro de distancia. Estos dispositivos de video ya son utilizados en diversos procedimientos quirúrgicos como por ejemplo en procedimientos endoscópicos y además en la neurocirugía.

2 LA REALIDAD VIRTUAL A BAJO NIVEL

2.1 Gráficas en computadora

El desarrollo de imágenes por medio del empleo de herramientas de software supone el previo conocimiento de los primitivos, un primitivo de salida en un entorno computarizado, corresponde a la definición de una figura básica de la que se pueden obtener figuras más complejas, por decirlo de otra manera es la unidad mínima de una gráfica.

2.1.1 Primitivos de salida

Los paquetes de programación de gráficas ofrecen funciones para describir una escena en términos de estructuras geométricas básicas, que reciben el nombre de primitivos de salida, y estos se deben agrupar para formar estructuras más complejas. Cada primitivo de salida se especifica con los datos de las coordenadas de presentación y otra información referente a la manera en que se debe desplegar el objeto. Los puntos y segmentos de línea recta son los componentes geométricos más simples de las imágenes.

Para dibujar un punto se convierte una posición particular de coordenadas dadas por un programa de aplicación en operaciones apropiadas para el dispositivo de salida en uso.

El trazo de líneas se efectúa mediante el cálculo de posiciones intermedias a lo largo de la trayectoria de la línea entre dos posiciones específicas. Los dispositivos digitales despliegan un segmento de línea recta al trazar puntos discretos entre los dos extremos. Las posiciones de coordenadas discretas a lo largo de la trayectoria se calculan a partir de la ecuación de la línea:

$$y = mx + b$$

Donde

$$m = \frac{y_2 - y_1}{x_2 - x_1}$$

Y b es el punto donde la recta intercepta el eje vertical.

La ubicación de puntos en pantalla puede expresarse únicamente con números enteros, obtenidos a partir de la ecuación anterior, lo que significa que, si con este cálculo se indica una posición con decimales, para hacerla coincidir con un punto en la pantalla o píxel⁸ deberá redondeársele. Por ejemplo si la ecuación arroja de resultado la coordenada (10.05, 20.76), esta se aproximara a (10, 21). Este redondeo de valores de coordenadas a enteros provoca que las líneas se desplieguen con apariencia de gradas.

La forma de escalón, característica de las líneas rectas, se percibe de manera clara en los sistemas con baja resolución, y puede mejorar su apariencia en cierto grado al desplegarlas en sistemas de mayor resolución.

⁸ El píxel (del inglés *picture element*, que se traduce como elemento de la imagen) es la menor unidad en la que se descompone una imagen digital, ya sea una fotografía, un cuadro de video o un gráfico.

Las técnicas más efectivas se basan en el ajuste de las intensidades de píxel a lo largo de las trayectorias de línea.

Podrían nombrarse aún más primitivos de salida, pero básicamente todos parten del punto y de líneas, entre los que se pueden mencionar destacan los polígonos, círculos, rectángulos, y otros.

2.2 Conceptos tridimensionales

Cuando se despliegan entornos tridimensionales hay más factores, que en los bidimensionales, que intervienen aparte de los valores para la tercera dimensión o el eje z como se le conoce en las ciencias matemáticas. Entre estos factores los que resaltan son las fronteras de los objetos, las que se pueden obtener con distintas combinaciones de superficies planas y/o curvas o bien indicando aspectos acerca de los interiores de los objetos.

Los programas de desarrollo gráfico por lo general incluyen funciones para desplegar componentes internos o vistas de secciones cruzadas de objetos sólidos. Del mismo modo, algunas representaciones geométricas se comprenden más en un espacio tridimensional que en dos dimensiones. Las figuras geométricas en tres dimensiones son mucho más complicadas por que se tiene que seleccionar muchos parámetros al especificar la manera en que se debe diagramar una escena tridimensional en el dispositivo de salida. La descripción de la escena se debe procesar a través de transformaciones de coordenadas de vista y rutinas de proyección que transforman coordenadas de vista tridimensional a coordenadas de dispositivos bidimensionales.

Para una vista seleccionada, se deben identificar las partes visibles de una escena; es necesario aplicar algoritmos de presentación de superficie para obtener una presentación realista de la escena.

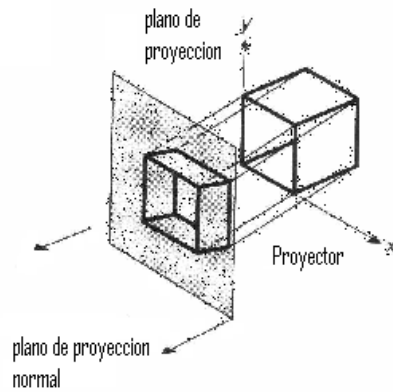
2.3 Métodos de despliegue tridimensional

Al momento de dibujar escenas en tres dimensiones es indispensable identificar las coordenadas desde la que se enfocara la imagen, es decir el punto de referencia para el usuario, lo que consiste en indicar el punto de vista de la imagen que tendrá este al observar la escena. La definición de estas coordenadas se realiza a través de dos técnicas de proyección, la paralela y la de perspectiva

2.3.1 Proyección paralela

Una de las técnicas para generar vistas de objetos sólidos se basa en proyectar puntos en la superficie del objeto a lo largo de líneas paralelas sobre el plano en el que se trabaja. Escogiendo distintas posiciones de vista, se puede proyectar los puntos visibles del objeto sobre el plano de despliegue para obtener diversas vistas bidimensionales consecutivas del objeto. En una proyección paralela, las líneas paralelas en la escena en coordenadas universales se proyectan en las líneas paralelas del plano de despliegue bidimensional; tal como lo indicó Bermúdez Ruiz, en su libro Técnicas de Texturización Tridimensional Para Trazado de Rayos. Con esta técnica se puede reconstruir la apariencia del objeto sólido a partir de las vistas principales.

Figura 1. **Proyección paralela**

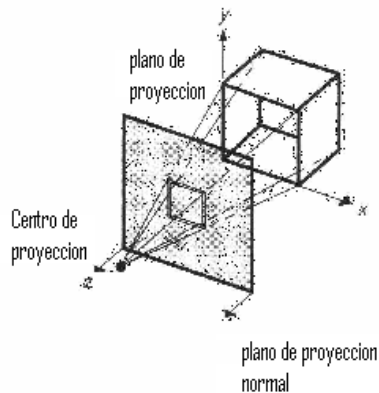


Fuente: Universidad de Chile: www.dcc.uchile.cl

2.3.2 Proyección de perspectiva

Otro método para generar una vista de una escena tridimensional es proyectar puntos hacia el plano de despliegue a lo largo de las trayectorias que converjan en un punto. Con esta técnica se logra que los objetos que en la escena se presentan más lejanos sean dibujados más pequeños que aquellos que están más cerca de la posición de vista.

Figura 2. **Proyección en perspectiva**



Fuente: Universidad de Chile: www.dcc.uchile.cl

2.4 Modelado en 3D

El modelado en 3D se ha convertido en uno de los recursos más utilizados en la actualidad en distintas áreas, para llevar ilustraciones del mundo real a la computadora. El modelado 3D no es otra cosa más que crear objetos en 3D, pero asistidos por una computadora, es decir que realmente el objeto tridimensional será un código binario que a través de un software, una tarjeta de video y un monitor podremos observar de igual forma y con las mismas o muy similares características como se haría en el mundo real.

Debido al software, el modelado en 3D es cada vez más sencillo de aplicar de una forma intuitiva, aunque obviamente, es necesario que el usuario tenga ciertos conocimientos sobre lo que son las gráficas en tercera dimensión y sus propiedades, y además de esto también son importantes los conocimientos matemáticos relacionados con la tercera dimensión.

2.4.1 Software para modelado en 3D

En el mercado existe una gran cantidad de software orientado al modelado en tercera dimensión y también al renderizado⁹ de modelos y ambientes. Entre los de mayor uso se encuentran *Maya*, *3D Studio Max*, *AC3D* y *Blender*; todos estos paquetes comparten una gran serie de similitudes por lo que someramente explicare el AC3D a modo de introducción al *software* de modelado 3D. Además cada uno de estos paquetes utiliza la librería OpenGL la cual también se describe someramente.

⁹ Renderizar es la acción de asignar y calcular todas las propiedades de un objeto antes de mostrarlo en pantalla.

2.4.1.1 AC3D

AC3D está diseñado para hacer el diseño y construcción de objetos 3D más rápida y fácilmente, es usado en varias partes del mundo para crear modelos 3D para videojuegos, renderización de imágenes y para visualización de datos. El AC3D puede ejecutarse en la mayoría del hardware que sigue los estándares actuales.

A continuación se listan algunos conceptos que AC3D usa para describir objetos 3D.

Vertex (Vértice en español): Es solo un punto en el espacio 3D, esta especificado por tres coordenadas X, Y y Z. Más de un vértice puede ocupar la misma posición en el espacio 3D. Un Vértice solo puede pertenecer a un objeto y no puede ser compartido por varios objetos.

Surface: En AC3D un *surface* (superficie en español) es una lista de vértices. Existen tres tipos de *surfaces*, *polygons*, *lines* o *polygon outlines*.

Object: Un *object* (objeto) AC3D es una lista de vértices, y *surfaces* que usan esos vértices, también tiene otros atributos como un nombre y una textura.

Group: Un grupo es un objeto especial el cual no tiene *surfaces* o vértices pero tiene otros objetos como hijos.

Material: Un material define el color de un *surface* y también los atributos que indican cómo reaccionará a la luz, como por ejemplo el brillo.

Texture: Imágenes pueden ser mapeadas sobre los superficies de un objeto. Estas imágenes son generalmente de archivos JPEG's o BMP's¹⁰.

Texture coordinates (coordenada de textura en español): Cuando el vértice de una superficie es dibujado, una coordenada de textura puede especificar la posición de una textura a mapear en ese punto. Una coordenada de textura es un valor de dos dimensiones.

Bounding box: Cuando se hace una selección de objetos o vértices, aparece una caja verde, así que se puede arrastrar, redimensionar o rotar esa caja para modificar su contenido.

Vistas: AC3D tiene un panel de control más otras cuatro ventanas (tres vistas ortogonales y una vista en 3D). Las divisiones entre las cuatro vistas pueden ser cambiadas de tamaño con el *mouse*. Las vistas se pueden agrandar si se utilizan los botones de vista que están arriba a la derecha, en el menú, o se puede alternar usando las teclas F1 o F2.

La ventana 3D utiliza *OpenGL* para la renderización, y permite ver instantáneamente los cambios hechos sobre el modelo. Permite también seleccionar objetos presionando la tecla ctrl. Y haciendo clic con el *Mouse*. Existen dos modos para ver en una ventana 3D, que se pueden escoger presionando las teclas 1 o 2, o seleccionando en el menú 3D, esos modos son *spin* y *walk*. Con *spin* se examinara el objeto y con *walk*, se caminara por la escena.

¹⁰ De los tipos de archivo para imágenes por computadora más difundidos.

2.4.1.1.1 Creación de objetos en AC3D

Si el modo de dibujo es *select* entonces se puede seleccionar, mover y redimensionar objetos. Cualquier otra forma creará un objeto cuando el *mouse* sea arrastrado en una de las vistas. Como se dijo previamente, un objeto es una colección de vértices y *surfaces*.

Arrastrando el mouse en una ventana permitirá especificar dos de las dimensiones para el nuevo objeto, la tercera es especificada desde el cursor 3D. Se puede mover este presionando ALT y moviendo el mouse. Esto especifica la posición faltante.

Las luces son un tipo especial de objeto, ellas no tienen una geometría asociada. Por el momento existe solo un tipo de luz: un punto de luz. Este es representado por un punto en el espacio. No se ve nada desplegado en su posición en la ventana de 3D, pero sí se pueden ver los efectos de la luz sobre los otros objetos en la escena.

Las figuras 3 y 4 ilustran la apariencia del paquete AC3D al momento de estar creando objetos 3D.

Sobre este paquete gráfico cabe mencionar que AC3D es un software propietario, con versiones para Linux y Windows, y se puede obtener en <http://www.ac3d.org/>, hay una versión shareware¹¹ pero que tiene deshabilitada la opción de guardar el archivo.

¹¹ Es una clase de software para poder evaluarlo de forma gratuita pero por un tiempo limitado y con uso o características limitadas. Para adquirir el software de manera completa necesita de un pago económico (muchas veces modesto).

Figura 3. Creación de objetos en AC3D

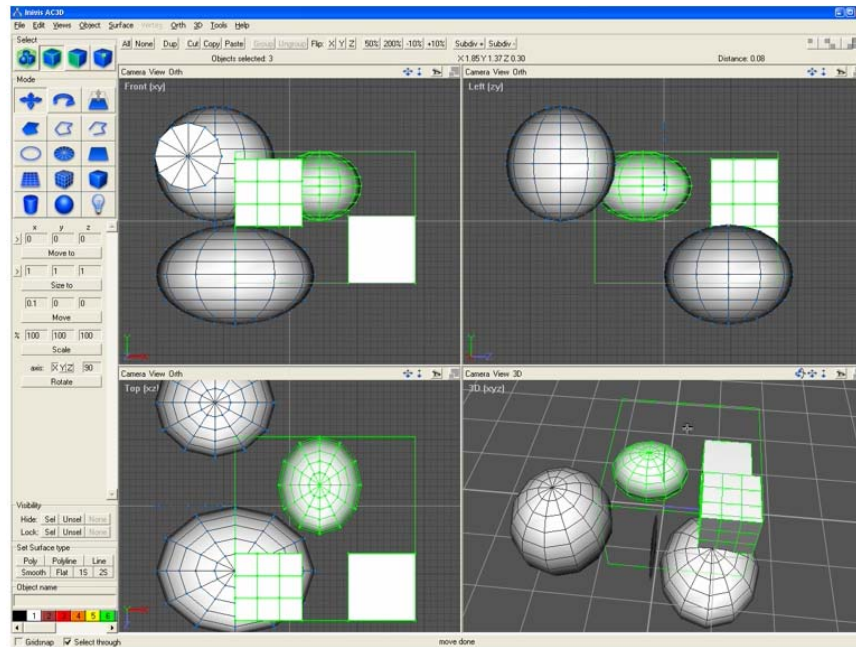
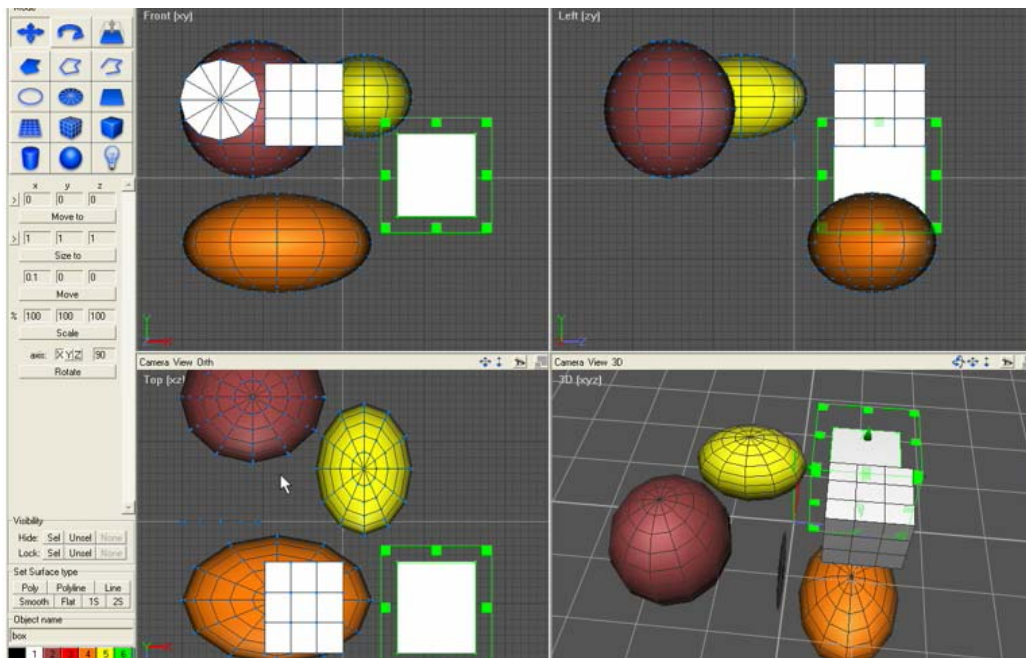


Figura 4. Configuración de propiedades de los objetos recién creados



2.4.1.2 OPENGL

OpenGL es una API (del inglés interfaz de programación de aplicaciones) creada por *Sillicon Graphics Inc.* para simplificar la programación de aplicaciones graficas en 2D ó 3D. OpenGL pone a disposición un pequeño conjunto de primitivas geométricas, a la par de estas primitivas, también brinda comandos para manipular la forma en que se verán los objetos. Un aspecto importante de OpenGL es que es independiente del sistema operativo y del sistema de ventanas en el que se ejecute.

OpenGL es una tecnología abierta, por lo que el acceso a la información para la utilización de esta API no se dificulta. Inclusive, actualmente ya existe una implementación *Open Source*¹² (código abierto) de OpenGL, que se utiliza de la misma manera que OpenGL, pero que permite tener acceso a su código fuente, esta es la librería MesaGL, creada bajo licencia GPL¹³.

Las principales razones para la utilización de OpenGL son que no se debe preocupar por el hardware al programar, por lo tanto la programación no se hace tan complicada, de igual manera el código que es escrito con OpenGL será independiente de la plataforma y del sistema operativo, por lo que será totalmente transportable.

Sintaxis de comandos en OpenGL

Como toda API, OpenGL sigue una serie de reglas para nombres de funciones y uso de variables.

¹² Literalmente, código abierto. Hace referencia al software libre, adscrito a la licencia GPL que permite ver, modificar y distribuir el código fuente.

¹³ Licencia Publica General

Los comandos en OpenGL usan el prefijo `gl` y la primera letra en mayúscula para cada palabra que forme el nombre del comando, como por ejemplo `glClearColor()`.

Igualmente, OpenGL define que las constantes empiezan con `GL_`, y usa letras mayúsculas para todo el nombre, y guiones bajos para separar las palabras, como por ejemplo `GL_COLOR_BUFFER_BIT`.

En OpenGL, más de un comando ha sido definido para trabajar con diferentes tipos de argumentos, y estos comandos tienen un sufijo formado por un número y una letra, el número indica el número de argumentos que el comando tomará y la letra el tipo de dato del que son los argumentos, como por ejemplo `glColor3f()`, `glColor4f()`.

En la siguiente tabla se puede ver cuáles son los sufijos usados en el API:

Tabla II **Tipos de datos en OpenGL y sus sufijos**

Tipo de datos OpenGL	Representación interna	Definición como tipo C	Sufijo literal en C
GLbyte	Entero de 8 bits	Char con signo	b
GLshort	Entero de 16 bits	Short	s
GLint, GLsizei	Entero de 32 bits	Long	i
GLfloat, GLclampf	Coma flotante 32 bits	Float	f
GLdouble, GLclampd	Coma flotante 64 bits	Double	d
GLubyte, GLboolean	Entero de 8 bits sin signo	Unsigned char	ub
GLushort	Entero de 16 bits sin signo	Unsigned short	us
GLuint, GLenum, GLbitfield	Entero de 32 bits sin signo	Unsigned int or long	ui

Fuente: Introducción a OpenGL de la Universidad de Córdoba España

2.4.1.3 GLUT

El término GLUT proviene de la abreviación de *OpenGL Utility Toolkit* y esta librería fue desarrollada para posibilitar la escritura de aplicaciones OpenGL que tuvieran un mayor grado de independencia del sistema de ventanas.

Gracias a GLUT se pueden hacer aplicaciones sin necesidad de saber cómo trabaja el API de un sistema de ventanas de un determinado sistema operativo, sin importar si se trata de un Unix o sus variantes Linux o bien de alguna versión de Windows.

GLUT es una pieza importante para la interactividad utilizando código en lenguaje C y OpenGL, ya que es quien provee de las funciones de entrada y salida del sistema operativo. Para la interacción únicamente se necesita el archivo glut.h y glut.lib.

2.4.1.3.1 Dibujando con GLUT

Para empezar a programar con GLUT, se deben seguir una serie de pasos dentro de una función main (en el lenguaje de programación C). Como se dijo con OpenGL, Glut también utiliza una sintaxis, que es parecida, por ejemplo todas las funciones de Glut comienzan con el prefijo Glut.

Lo primero que se debe hacer para poder utilizar GLUT es hacer una llamada a la función glutInit, cuya sintaxis es:

```
void glutInit(int *argc, char **argv);
```

En donde los dos parámetros son punteros a los parámetros de la función main(). Después se realiza un llamado a las funciones glutInitWindowPosition y a glutInitWindowSize que tienen la siguiente sintaxis:

```
void glutInitWindowPosition(int x, int y);  
void glutInitWindowSize(int width, int height);
```

La primera indicará la posición de la esquina superior izquierda de la ventana mediante los parámetros x,y; mientras que la segunda función indicará el tamaño de la ventana. Seguidamente se debe definir el modo de desplegado usando la función glutInitDisplayMode, y su sintaxis es:

```
void glutInitDisplayMode(unsigned int mode);
```

El parámetro mode, especifica el modo de despliegue, con el que se puede indicar el modelo de color con las constantes GLUT_RGBA, GLUT_RGB o GLUT_INDEX, aquí se señala también si el buffer de la ventana es simple o doble con las constantes GLUT_SINGLE y GLUT_DOUBLE. Además es posible indicar una ventana con un particular conjunto de buffer. Los más comunes son GLUT_ACUM., GLUT_STENCIL y GLUT_DEPTH.

Después de indicar todos esos datos, ya es posible crear una ventana con la función glutCreateWindow, la que tiene la siguiente sintaxis, en donde el parámetro title será el título de la ventana desplegar:

```
int glutCreateWindow(char *title);
```

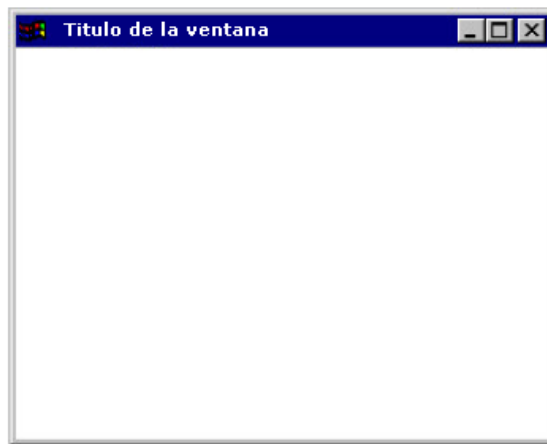
Uniendo todas estas instrucciones se obtiene el siguiente bloque de código:

Figura 5. Instrucciones para crear una ventana en OpenGL

```
#include <gl/glut.h>
void main(int argc, char **argv) {
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_DEPTH | GLUT_SINGLE | GLUT_RGBA);
    glutInitWindowPosition(100,100);
    glutInitWindowSize(320,320);
    glutCreateWindow("Título de la ventana");
}
```

Si se compila y ejecuta este programa, solo se verá una ventana vacía. Como la que se muestra en siguiente figura.

Figura 6. Una ventana simple en OpenGL



Fuente: Elaboración propia

Para desplegar gráficas en 3D, es necesario indicarle a GLUT cuál es la función de renderización, es decir, que deberá dibujar el programa en la ventana. Para indicar la función de despliegue se utiliza la función `glutDisplayFunc(void(*func)(void))`, que tiene la siguiente sintaxis:

```
void glutDisplayFunc(void(*func)(void));
```

En donde el parámetro `func`, es el nombre de la función que realizará la labor de despliegue. Por ejemplo, se podría declarar una función en el programa con el nombre de despliegue, y allí definir todos los comandos OpenGL para dibujar, como sigue a continuación:

Figura 7. **Declaración de una función de despliegue en OpenGL**

```
#include <gl/glut.h>
#include <gl/gl.h>
void despliegue(void){
    // Aquí va todo el código OpenGL
    glClear (GL_COLOR_BUFFER_BIT);
    glBegin (GL_TRIANGLES);
    glVertex3f(1.0,1.0,1.0);
    glVertex3f(1.0,0.0,0.0);
    glVertex3f(0.0,1.0,0.0);
    glEnd();
    glFlush();
}
void main(int argc, char **argv) {
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_DEPTH | GLUT_SINGLE | GLUT_RGBA);
    glutInitWindowPosition(100,100);
    glutInitWindowSize(320,320);
    glutCreateWindow("Título de la ventana");
    glutDisplayFunc(despliegue);
    glutMainLoop();
}
```

Cabe destacar el uso de la función `glutMainLoop()`; que es la que genera un ciclo infinito para que GLUT esté a la espera de los eventos que sucedan en la aplicación, como teclado, *mouse*, el evento de terminación, etc. Para poder realizar animaciones utilizando GLUT, es necesario modificar o más bien indicar en la función `glutInitDisplayMode` que se quiere un doble buffer, sería algo como esto `glutInitDisplayMode(GLUT_DEPTH | GLUT_DOUBLE | GLUT_RGBA)`; y también es como utilizar una función de GLUT que ejecuta algo cuando no existe ningún evento que procesar, que sería la función `glutIdleFunc`, y su sintaxis es:

```
void glutIdleFunc(void(*func)(void));
```

Que es la misma que `glutDisplayFunc`, así que se hace lo mismo, se llama a la función de despliegue con `glutIdleFunc` y la ventana se repintará si no se hace nada, es decir, se pueden usar parámetros para que los objetos dibujados con OpenGL cambien su vista y con esto se tendrá una animación.

Uso del teclado

GLUT permite construir aplicaciones que detectan la entrada de teclado usando ya sea las teclas normales o las teclas especiales como F1 o F2. Para esto GLUT provee dos funciones para el manejo de los eventos del teclado, la primera de ellas es `glutKeyboardFunc`, que es usada para decirle al sistema de ventanas (ya sea Windows ó Unix) qué función se quiere que procese los eventos de las teclas normales. La sintaxis de esta función es:

```
void glutKeyboardFunc(void(*func)(unsigned char key, int x, int y));
```

En donde el parámetro `func` es el nombre de la función que procesará los eventos, y a esta función se le pasan 3 parámetros.

El primero va a almacenar el valor ASCII de la letra que fue presionada y los otros dos argumentos indicaran la posición del puntero del mouse cuando la tecla fue presionada. Como ejemplo de función que procesa los eventos se puede usar el siguiente código:

Figura 8. **Función de captura de teclas normales en OpenGL**

```
void teclasNormales(unsigned char key, int x, int y)
{
    if (key == 27) // sí la tecla presionada fue Esc
    {
        printf("Adios...");
        exit(0);
    }
}
```

La otra función para el manejo del teclado es `glutSpecialFunc`, que tiene la siguiente sintaxis:

```
void glutSpecialFunc(void(*func)(int key, int x, int y));
```

En donde los parámetros son iguales que la función anterior, pero se puede observar que el primer parámetro es de tipo entero, esto debido a que las teclas especiales son de este tipo. GLUT tiene constantes que definen los valores de las teclas especiales y son `GLUT_KEY_F1`, `GLUT_KEY_F2`, etc. Como ejemplo se puede utilizar el siguiente código:

Figura 9. **Función de procesamiento de teclas especiales en OpenGL**

```
void procesarTeclasEspeciales(int key, int x, int y)
{
    switch(key){
        case GLUT_KEY_F1 :
            printf("F1 presionada");
            break;
        case GLUT_KEY_F2 :
            printf("Tecla F2 presionada");
            break;
    }
}
```

Un aspecto importante del teclado al momento de desarrollar aplicaciones es el uso de las teclas modificadoras CTRL., ALT y SHIFT, para esto GLUT brinda la función `glutGetModifiers`, y su sintaxis es:

```
int glutGetModifiers(void);
```

Esta función devuelve un entero en caso de que uno de los modificadores está presionado, y se puede saber cuál es con las constantes `GLUT_ACTIVE_SHIFT`, `GLUT_ACTIVE_CTRL`. Y `GLUT_ACTIVE ALT`.

Uso del mouse

Como en el teclado, GLUT provee una función para detectar los clics del mouse, y esto es con la función `glutMouseFunc`, que presenta la siguiente sintaxis:

```
void glutMouseFunc(void (*func) (int button, int state, int x, int y);
```

En donde el primer parámetro indica cual botón fue presionado, este parámetro puede tener uno de estos tres valores: GLUT_LEFT_BUTTON, GLUT_MIDDLE_BUTTON y GLUT_RIGHT_BUTTON. El siguiente parámetro indicará si el botón fue presionado o soltado y los valores para este son: GLUT_DOWN y GLUT_UP.

Limpiar la ventana

Dibujar en un monitor de computadora es diferente a dibujar en un papel. En una computadora, la memoria es quien mantiene el dibujo, la que es usualmente llenada con el último dibujo que se presentó, así que siempre es necesario limpiar la memoria con algún color de fondo antes de empezar a dibujar de nuevo.

El color que se use de fondo dependerá de la aplicación, por ejemplo, para un procesador de palabras, generalmente se utiliza el color blanco como fondo, ya que es el color de una hoja de papel real.

En OpenGL básicamente se utilizan dos comandos para borrar la pantalla, estos son `glClearColor` y `glClear`, un ejemplo de estas funciones sería:

```
glClearColor(0.0,0.0,0.0,0.0);  
glClear(GL_COLOR_BUFFER_BIT);
```

La primera línea especifica el color que se utilizará para limpiar, y el comando siguiente limpia la ventana con el color que se especificó.

El parámetro de `glClear()` indica que buffers serán limpiados.

En este caso, el programa limpiará solo el buffer de color, donde la imagen está desplegada en la pantalla. Generalmente solo hay que especificar el color de limpiado una vez en el programa.

Con `glClear`, se puede limpiar todo tipo de buffers que haya, por ejemplo, para limpiar el buffer de profundidad sería:

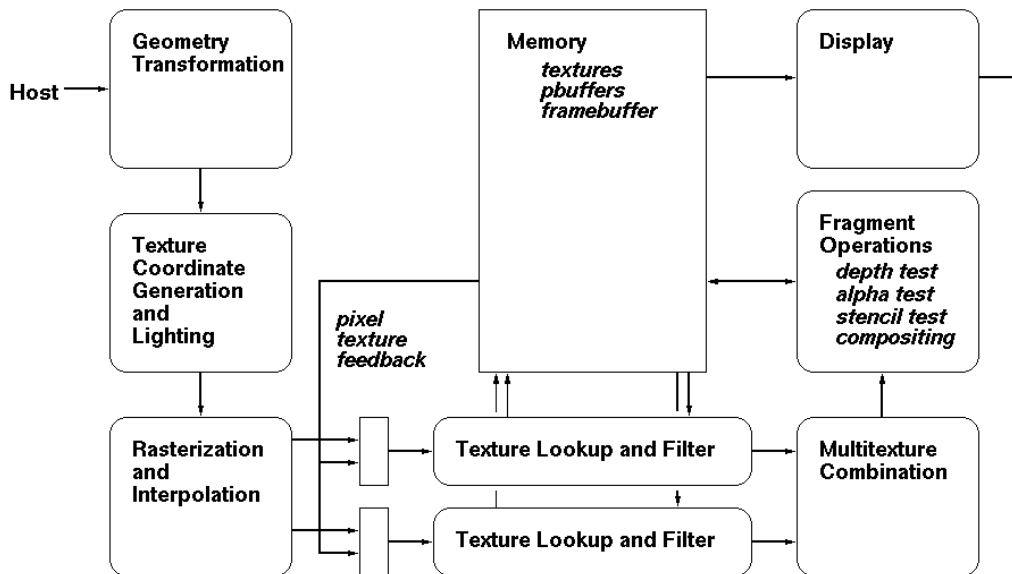
```
glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFERBIT);
```

Antes de usar un comando para limpiar múltiples buffers, se tiene que establecer los valores con los que cada buffer será limpiado, así que aparte de `glClearColor()` existe también `glClearIndex()`, `glClearAccum`, y `glClearStencil()` para poder establecer los valores.

OpenGL permite especificar múltiples buffers porque limpiar es generalmente una operación lenta, ya que se necesita pintar cada píxel en la ventana, y por ejemplo, hay hardware que permite intercambiar buffers para así limpiar simultáneamente, a diferencia del hardware que no soporta esto, que tiene que hacer este proceso secuencialmente.

La distribución de la memoria en OpenGL resulta fácil entenderla al graficarla, tal como se presenta en la figura 10.

Figura 10. Distribución de la memoria en OpenGL



Fuente: Sitio oficial de OpenGL, www.opengl.org

Puntos, líneas y polígonos

Un punto es representado por un conjunto de números de punto flotante llamado vértice. Todos los cálculos internos son hechos como si los vértices fueran tridimensionales. A los vértices especificados por un usuario como bidimensionales, OpenGL les asignará a cada uno un 0 a su coordenada z. OpenGL trabaja con la geometría proyectiva de tres dimensiones, es decir que para cálculos internos, todos los vértices son representados con coordenadas de 4 puntos flotantes (x, y, z, w). Si w es diferente de cero, esas coordenadas corresponden a los puntos (x/w, y/w, z/w). Es posible especificar la coordenada w, pero eso rara vez se hace, si w no es especificada, por default será 1.0. Los polígonos son áreas cerradas, es decir una serie de líneas interconectadas, que delimitan un área.

Los polígonos son generalmente dibujados rellenos con pixeles, pero puede también dibujarse sin rellenar, es decir, con puras líneas o puntos. Muchas veces, los polígonos pueden ser complicados, así que OpenGL pone algunas restricciones para que pueda decir que un objeto es un polígono. Primeramente, los lados de los polígonos no se pueden interceptar, también, un polígono en OpenGL debe ser convexo, significando esto que no puede haber irregularidades en los bordes.

Rectángulos

Debido a que los rectángulos son tan comunes en las aplicaciones gráficas, OpenGL tiene una primitiva de un rectángulo relleno, nos referimos a `glRect()`, así que si se quisiera pintar un rectángulo se escribiría el siguiente código.

```
glRect(1.0,1.0,10.0,10.0);
```

Donde los dos primeros parámetros indican la esquina superior izquierda del rectángulo y los dos últimos parámetros nos indican la esquina inferior derecha del rectángulo.

Curvas

Cualquier superficie curva existente puede ser aproximada mediante segmentos de líneas cortas o pequeñas regiones poligonales. Las curvas no son primitivas geométricas, por lo que deberán trazarse de la manera antes descrita.

Especificando vértices

Con OpenGL, todos los objetos geométricos están descritos como un conjunto ordenado de vértices. Se usa el comando `glVertex()` para especificar un vértice. La sintaxis de éste comando es:

```
void glVertex (TYPEcoords);
```

`glVertex()` siempre tiene que ser llamado entre `glBegin()` y `glEnd()`.

Dibujando primitivas geométricas

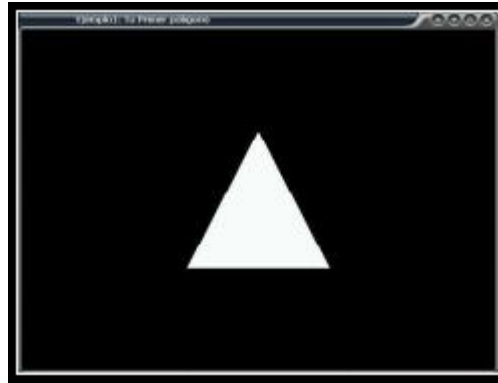
Una vez descrito como especificar vértices, es necesario saber cómo es que en OpenGL se pueden crear un conjunto de puntos, una línea, o un polígono a partir los mismos. Para hacer esto, como previamente se mencionó, se usa `glBegin()` y `glEnd()`. El argumento que se le pase a `glBegin()` determinará qué tipo de primitivo geométrico es construido con los vértices. Por ejemplo, si se quiere dibujar un polígono se utiliza lo siguiente:

Figura 11. **Especificación de un triángulo en OpenGL**

```
glBegin(GL_POLYGON);  
glVertex3f( 0.0f, 1.0f, 0.0f);  
glVertex3f(-1.0f,-1.0f, 0.0f);  
glVertex3f( 1.0f,-1.0f, 0.0f);  
glEnd();
```

Y con esto se obtendrá el triángulo que se ilustra a continuación.

Figura 12. Un triángulo en OpenGL



Esto dibujaría un polígono (triángulo) cerrado y relleno, pero si se utilizara `GL_POINTS` en lugar de `GL_POLYGON`, entonces se dibujarían solo cinco puntos. La siguiente muestra cuales son los primitivos para usar con `glBegin()`;

Tabla II Lista de primitivos de GLUT

Primitiva	Significado
<code>GL_POINTS</code>	Para trazar puntos individuales
<code>GL_LINES</code>	Un par de vértices serán interpretados como una línea
<code>GL_POLYGON</code>	Un polígono convexo
<code>GL_TRIANGLES</code>	Cada tres vértices son interpretados como un triángulo
<code>GL_QUADS</code>	Cada cuatro vértices forman un polígono de 4 lados
<code>GL_LINE_STRIP</code>	Series conectadas de líneas
<code>GL_LINE_LOOP</code>	Lo mismo de la anterior, pero esta cierra un polígono

2.4.1.4 VRML

VRML es el acrónimo del inglés *Virtual Reality Modeling Language*, Lenguaje para Modelado de realidad virtual.

Este es un lenguaje relativamente nuevo que fue creado para exponer contenido en 3D vía internet, actualmente, es usado en un porcentaje muy bajo de las webs que existen.

Hay dos versiones de VRML, la versión 1.0 y la versión 2.0, también conocida como VRML97, las diferencias entre estas dos versiones son muchas, ya que por ejemplo en la versión 1.0 no se permite la animación.

Para visualizar un mundo virtual hecho con VRML es necesario tener un visualizador VRML, éste es un *plugin*¹⁴ para los exploradores (Internet Explorer, Netscape, Mozilla, Opera, etc.). Así que también es necesario un navegador Web. El visualizador puede ser de diferentes compañías de software y puede descargarse de Internet.

Es necesario para editar el código, un editor de texto cualquiera (Vi, Gedit, Notepad, etc.), los archivos VRML tienen la extensión *.wrl* y también existe la extensión *.wrz* que es el código VRML comprimido con GZIP.

Un aspecto importante en un mundo virtual son las medidas, y en general se utilizan los metros para un mundo en VRML, en lo que ángulos se refiere, se utilizan los radianes en lugar de los grados.

¹⁴ *Plugin* Es un programa de computadora que interactúa con otro programa para aportarle una función o utilidad específica, generalmente muy específica.

Conceptos y código base

Comentarios

Para que un navegador pueda identificar que un archivo contiene código VRML, este debe contener en la primera línea lo siguiente:

```
#VRML V2.0 utf8
```

Así se le indica que se está utilizando la versión VRML97 y que se utilizará el estándar utf8 para la codificación de caracteres. El carácter # es utilizado para comentarios, es decir, es como el // en C++ o java, así que todo lo que vaya después de un # en una línea será ignorado por el navegador, tal y como se ilustra a continuación.

Figura 13. **Comentarios en VRML**

```
#VRML V2.0 utf8
# Interacción con modelos 3D usando VRML
# La línea de arriba es un comentario y esta también.
```

Hay que tomar en cuenta que el VRML es sensible en cuestión de mayúsculas y minúsculas y si la primera línea no es escrita como debe de ser, habrá un error.

Los nodos

Los nodos son los objetos en el mundo virtual y son lo que lo conforma. Los nodos contienen dentro de ellos a los campos que son las propiedades de los objetos. También los nodos pueden contener otros nodos, es decir, se anidan. Un simple nodo se escribiría así:

Figura 14. **Definición de un nodo simple**

```
Group {  
  children [  
  ]  
}
```

Y un nodo anidado se escribiría así:

Figura 15. **Definición de un nodo anidado**

```
Group {  
  children [  
    Group {  
    }  
  ]  
}
```

Como ejemplo puede crearse el nodo HolaMundo, el que contiene información general sobre nuestro mundo:

Figura 16. **Creando un nodo de ejemplo**

```
HolaMundo {  
  title "Creando ambientes con realidad virtual"  
  info ["Tratamiento de la Claustrofobia"]  
}
```

Con esto se le está indicando al navegador que el título del mundo es el que está en el campo *title* y que puede obtener más información del mismo información con el campo *info*.

Como ya se dijo previamente los nodos son la piedra angular de los mundos en VRML, y por lo tanto para dibujar objetos 3D existen distintos tipos de nodos, el más básico es el nodo *Shape*, mediante dos campos de este nodo se puede definir una forma básica, estos campos son *Appearance* y *Geometry*, y pueden utilizarse como prosigue.

Figura 17. **Apariencia y geometría de los nodos**

```
Shape {  
  Appearance apariencia {  
    Material materiales {  
    }  
  }  
  geometry Box {  
  }  
}
```

Escribiendo el código con los nodos *HolaMundo* y *Shape* se obtendría un cubo en el espacio vacío, con los colores predeterminados.

Transformaciones

Las transformaciones son lo que permiten mover los objetos en el mundo virtual, con VRML se pueden hacer tres tipos de transformaciones: translación, rotación y escala. La transformación en un objeto se hace con el nodo *Transform*, y hace de la siguiente forma:

Figura 18. **Transformaciones en los nodos**

```
Transform {  
  translation 1 0 1  
  rotation 0 1 0 0.78  
  scale 2 1 2  
  children [ ] }
```

El campo *translation*, es el que va a mover el centro de los objetos a la posición en el mundo según los parámetros que se le pase, que son X Y y Z respectivamente. El campo *rotation* gira al objeto alrededor de un eje, los tres primeros parámetros describen el eje sobre el que se girará el objeto y el cuarto parámetro es el ángulo que se girará en radianes. El campo escala multiplica el tamaño del objeto por los parámetros que se le indiquen.

Apariencia en los objetos

Para dar apariencia a los objetos se tiene el campo *appearance*, el que permite tener una apariencia en un nodo y además dentro tener un nodo de tipo material, en el cual se puede definir los siguientes campos: *diffuseColor*, *specularColor*, *emissiveColor*, *ambientIntensity*, *shininess*, *transparency*; cada uno de estos campos define distintas propiedades del objeto las cuales se definen a continuación.

DiffuseColor: Define el color del objeto , y requiere tres parámetros, que son el color expresado en RGB¹⁵.

SpecularColor: Define el color resaltado al objeto, es decir para los reflejos.

EmissiveColor: Se define un color RGB que será emitido por un objeto.

AmbientIntensity: Este campo representa cuanta cantidad de luz del ambiente el objeto reflejará, recibe solo un parámetro.

¹⁵ El modelo de color RGB utiliza un modelo aditivo, en el cual el rojo, verde, y azul son combinados de diferentes maneras para crear otros colores.

Shininess: recibe también solo un parámetro, y define lo reflexivo de un objeto.

Transparency: Con un solo parámetro define la transparencia del objeto.

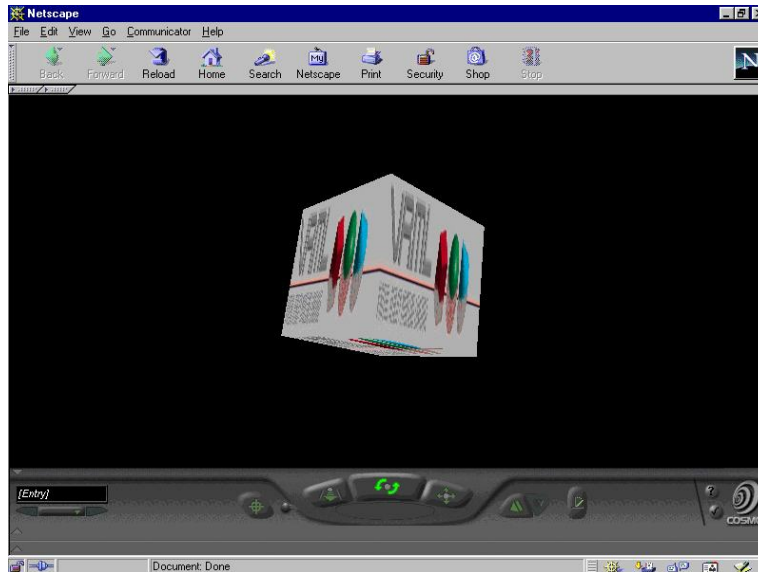
Otro aspecto importante para la apariencia de los objetos son las texturas, una textura es una serie de pixeles que se mapean en un objeto, esto puede ser una imagen JPEG ó PNG. Y se hace con el nodo *ImageTexture* que se especifica dentro de *Appearance*.

Si se utilizan todas estas propiedades y se redactan se obtendrá el siguiente resultado.

Figura 19. **Cubo con textura en VRML**

```
#VRML V2.0 utf8
HolaMundo {
  title "Ejemplo de VRML"
  info [ "Carlos Mancio / cmancio@gmail.com " ] }
Transform {
  rotation 0 1 0 0.60
  children [
    Shape {
      appearance Appearance {
        material Material{
          specularColor 0 1 0
          shininess 0.2 }
        texture ImageTexture {
          url "vrml.jpg"  }}
      geometry Box {
    }}}}
```

Figura 20. Ejemplo de un cubo en VRML



La figura número veinte ilustra el resultado de ejecutar las líneas de código anteriores.

Texto

Para utilizar texto en el mundo virtual se utiliza el nodo *text* que tiene los siguientes campos: *string*, *fontStyle*, *maxExtent*, *length*. El siguiente ejemplo muestra cómo usar el nodo *text*.

La figura veintidós a su vez, ilustra el resultado de ejecutar el código del lenguaje VRML adjunto en la imagen siguiente en un navegador Netscape.

Figura 21. Despliegue de texto en VRML

```
...
Transform {
  translation 0 0 2
  rotation 0 1 0 0.60
  children [
    Shape {
      appearance Appearance {
        material Material {
          emissiveColor 1 1 0  }}
      geometry Text {
        string "USAC"
        fontStyle FontStyle {
          size 0.8
          justify "MIDDLE"
        }
      }
    }
  ]
}
```

Figura 22. Escritura de texto en VRML



Interacción

Con el lenguaje VRML no solo se puede crear mundos virtuales estáticos que muestren objetos en tres dimensiones, sino que también permite que exista la interacción con el usuario, así como también la animación del mundo virtual. La interacción se basa en que un mundo virtual puede recibir datos y reaccionar de cierta forma a esas peticiones a través de un modelo interno de ejecución, es decir, que conectando los nodos y especificar rutas en las que se pueden enviar y recibir mensajes.

Esas rutas no tendrán visibilidad en el mundo, pero serán las que permitan la interacción buscada.

- Cámaras en la escena

Es sabido que dentro de un mundo en 3D es muy importante el uso de cámaras, que son los objetos que proporcionan la panorámica del mundo, y diferentes vistas de éste. Como el equivalente a una cámara en el mundo real. En VRML el uso de cámaras se hace a través del nodo *Viewpoint*, el cual se utiliza así:

Figura 23. **Definición de cámaras en la escena**

```
Viewpoint {  
    position 0 5 5  
    orientation 1 0 0 0.39  
    fieldOfView 0.39  
    description "Camara #1"  
}
```

En donde el campo *position* va a ubicar la cámara en las coordenadas X, Y y Z especificadas; *orientation* funciona igual que en el nodo *transform* y sirve para orientar la cámara hacia un ángulo a partir de un eje. Con *FieldOfview* se indica cual será el ángulo que abarcará la vista de la cámara y, por último, el campo *description* es una simple descripción de la cámara.

- Luces

Lo primero que hay que decir es cómo funciona el modelo de iluminación de VRML. Todas las luces tienen los siguientes campos: *color*, *ambientIntensity*, e *intensity*. Cada luz tiene una intensidad (*intensity*) con un valor entre 0 y 1 que determina su brillo. También tiene una intensidad ambiental (*ambient intensity*), de nuevo entre 0 y 1, que determina cuánta luz contribuye al ambiente general de la escena. Por este motivo, cuántas más luces haya en la escena, más iluminado quedará el ambiente, lo que por otro lado es lógico.

Una luz de ambiente (*Ambient*) es aquella que luce en todas las superficies de las escenas, simulando provenir de otros objetos. Cada luz tiene además un color asociado, que es el color que emite. La luz directa emitida por su fuente está dada por la fórmula:

$$Luz = intensity \times color$$

La luz ambiental que contribuye a la escena es por su lado está dada por:

$$Luz = ambient_intensity \times color$$

Cada fuente de luz tiene además un área de efecto, así que hay que tener en mente siempre la proporción de las escenas. El método para hacer esto varía entre los diferentes tipos de nodos.

Existen diferentes tipos de nodos para generar luz, uno de ellos es *DirectionalLight*, que genera una luz que brilla desde una dirección uniformemente, la sintaxis de *DirectionalLight* es la siguiente:

Figura 24. **Generación de luz mediante DirectionalLight**

```
DirectionalLight {  
  exposedField SFFloat ambientIntensity 0  
  exposedField SFColor color 1 1 1  
  exposedField SFVec3f direction 0 0 1  
  exposedField SFFloat intensity 1  
  exposedField SFBool on TRUE  
}
```

Otro nodo de luz es el *PointLight*, que este es un punto desde el cual se emanará una luz con cierto radio al que afectará.

Para posicionarlo existe el campo *location* y para especificar el radio se utiliza el campo *radius*. La sintaxis es:

Figura 25. **Generación de luz mediante PointLight**

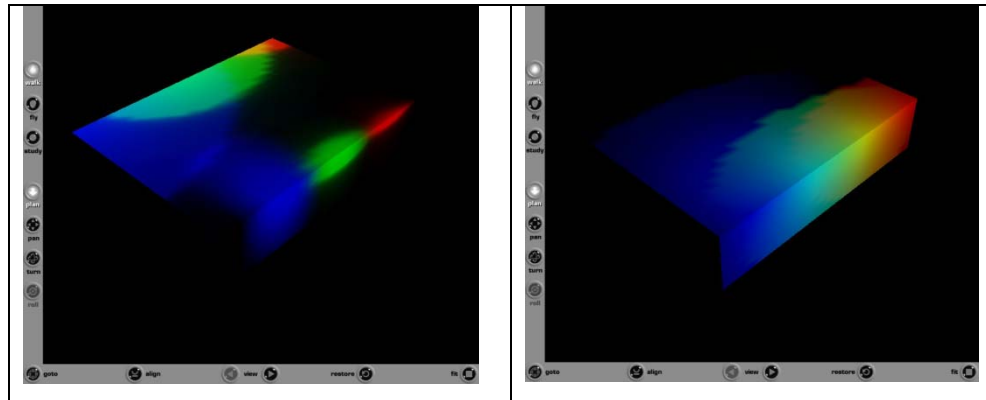
```
PointLight {  
  exposedField SFFloat ambientIntensity 0  
  exposedField SFVec3f attenuation 1 0 0  
  exposedField SFColor color 1 1 1  
  exposedField SFFloat intensity 1  
  exposedField SFVec3f location 0 0 0  
  exposedField SFBool on TRUE  
  exposedField SFFloat radius 100  
}
```

El último nodo de iluminación es el *SpotLight*, y éste genera la luz como si tratara de un foco, su sintaxis es la siguiente:

Figura 26. **Generación de luz mediante SpotLight**

```
SpotLight {  
  exposedField SFFloat ambientIntensity 0  
  exposedField SFVec3f attenuation 1 0 0  
  exposedField SFFloat beamWidth 1.570796  
  exposedField SFColor color 1 1 1  
  exposedField SFFloat cutOffAngle 0.785398  
  exposedField SFVec3f direction 0 0 1  
  exposedField SFFloat intensity 1  
  exposedField SFVec3f location 0 0 0  
  exposedField SFBool on TRUE  
  exposedField SFFloat radius 100  
}
```

Figura 27. **Distintos tipos de iluminación en un mismo objeto**



- Opciones del navegador

Para la interacción es importante especificarle al navegador VRML la forma en que el usuario puede moverse por el mundo, para esto existe el nodo *NavigationInfo*, del cual su sintaxis es la que sigue:

Figura 28. Especificaciones del navegador

```
NavigationInfo {
  eventIn SFBool set_bind
  exposedField MFFloat avatarSize [0.25, 1.6, 0.75]
  exposedField SFBool headlight TRUE
  exposedField SFFloat speed 1.0
  exposedField MFString type "WALK"
  exposedField SFFloat visibilityLimit 0.0
  eventOut SFBool isBound
}
```

El campo *set_bind* se refiere a los eventos que actúan sobre el navegador, el campo *Headlight* indica si la luz por defecto del navegador está encendida o no, la que se puede apagar poniendo a *headlight* a *FALSE*.

Con *visibilityLimit* se puede especificar un límite de visibilidad para los mundos creados con VRML, es decir desde la vista del usuario para atrás, cuanto podrá verse. Un campo interesante de este nodo es el *speed*, el que permitirá modificar la velocidad con que el usuario se puede mover, lo normal es un valor de 1.0 y que puede aumentar al doble si se usa 2.0. El campo *type*, sirve para especificar la forma en que el usuario se moverá en el mundo. Hay cuatro tipos de formas:

- *WALK*: El usuario se moverá por el mundo y será afectado por la gravedad.
- *FLY*: El usuario se moverá por el mundo sin que la gravedad le afecte.
- *EXAMINE*: El usuario no se moverá, pero podrá rotar el mundo para observarlo.
- *NONE*: El usuario no podrá interactuar con el mundo, el programador define qué es lo que quiere que se vea de frente al usuario.

3. FOBIAS

3.1 Definición

Las fobias son miedos, dudas e indecisiones, que de igual manera que una idea obsesiva, se abren paso por sí mismas de manera inconsciente y persistente hasta aflorar en la conciencia.

Entre las fobias más comunes están el miedo a la suciedad, a las bacterias, al cáncer, a las multitudes y a los espacios cerrados.

En general se puede decir que la angustia de quien padece de alguna fobia nace de una idea, objeto o situación específica en su vida diaria y se desplaza hacia otra idea o situación simbólica en forma de un miedo neurótico específico. El miedo que quien padece una fobia experimenta en presencia de determinado objeto o experiencia es, en realidad el miedo desplazado de algún componente angustiante dentro de su personalidad, según lo indica Kolb en su publicación *Psiquiatría Clínica Moderna*.

Una definición más formal de fobia es la acuñada por Friedrich Dorsch (1977) y es “violenta angustia que aparece compulsivamente, ante determinados objetos o situaciones exteriores que realmente no representan peligro. Según la concepción psicoanalítica, este miedo a un objeto o a una situación es en su origen miedo a un impulso instintivo interior peligroso, es miedo a la irrupción del instinto en la conciencia y a su satisfacción...”.

Entonces con base a esta definición puede decirse que, la situación respecto a la cual el individuo es fóbico, simboliza o representa incidentes que han producido ira, rabia o venganza y no se trata realmente del objeto o situación en cuestión, desde luego el paciente no se da cuenta de la fuente psicológica o de la importancia de su miedo, y aunque reconoce que es irracional, es incapaz de regular su vida si no es de acuerdo a lo que dicta su fobia. Constantemente las personas fóbicas intentan controlar su angustia evitando el objeto o situación a la que ha desplazado dicha angustia.

3.2 Claustrofobia

La claustrofobia podría definirse, de una manera sencilla, como el miedo a los espacios cerrados. Este miedo es bastante común en la población general, siendo una de las fobias específicas con la incidencia más alta. Sin embargo, la cantidad de personas que padecen de esta fobia contrasta con el pequeño número de personas que buscan ayuda profesional para superar su problema, cosa que, por otro lado, también ocurre en las demás fobias específicas. La mayoría de las personas que tienen miedo en situaciones que implican algún grado de encierro suelen indicar que este problema puede llegar a ser muy molesto, pero que normalmente pueden manejarlo, evitando activamente este tipo de situaciones.

En estadísticas presentadas por el Instituto Nacional de la Salud Mental (INSM) del gobierno de Estados Unidos, se indica que 3 de cada 4 casos de claustrofobia no son graves. Pero aún así, entre un 2 y un 5% del total de la población padece una claustrofobia grave.

De acuerdo con ciertos autores, existe una discrepancia entre el número de personas que padecen de claustrofobia y las que buscan ayuda o reciben algún tratamiento; lo que puede explicarse, en parte, porque la mayoría de las personas desconocen que este problema puede llegar a desaparecer con un tratamiento apropiado. Por otro lado, también hay que tener en cuenta que las personas suelen buscar ayuda especializada hasta que su problema interfiere de un modo notable en su vida ya sea en su trabajo, en su familia, en sus relaciones sociales y hasta en su salud, por ejemplo, cuando es necesario realizar una prueba médica como una TAC¹⁶, se estima que entre el 4 y el 10% de los pacientes son incapaces de completar esta prueba debido a su claustrofobia, nuevamente según datos del INSM.

Según los actuales sistemas de clasificación de los trastornos mentales, la claustrofobia se incluye dentro de la categoría de Fobia específica. De acuerdo con el DSM-IV, que es la cuarta edición del Manual Diagnóstico y Estadístico de los trastornos mentales de la Asociación Americana de Psiquiatría. Según dicho manual una fobia específica se puede diagnosticar cuando se cumplen los siguientes criterios:

- a) La persona experimenta un temor acusado y persistente (por ejemplo al estar en un sitio cerrado, estar en un sitio elevado, etc.).
- b) La persona muestra una respuesta inmediata de ansiedad al ser expuesta al estímulo fóbico.

El DSM-IV distingue entre tres tipos de fobias específicas: el tipo sangre-inyección-daño, el tipo situacional y otros tipos.

¹⁶ Tomografía axial computarizada.

La claustrofobia se ubica dentro de la fobia específica tipo situacional. Las características generales antes descritas se atribuyen a cualquier fobia específica. Lo que realmente hace que se considere a una claustrofobia como tal, es que las manifestaciones clínicas fundamentales de este problema se focalizan en un miedo intenso a situaciones relacionadas con espacios cerrados: habitaciones pequeñas, habitaciones cerradas, sótanos, bodegas, túneles, ascensores, etc. En general, las personas con claustrofobia no suelen tener miedo sólo a una de estas situaciones, sino que suelen sentir temor en cualquier situación que implique encierro, restricción o confinamiento.

Según algunos expertos en la materia, la claustrofobia es similar a la agorafobia¹⁷, entre otras cosas, porque el temor no se focaliza en el espacio en sí mismo, sino a lo que podría ocurrir dentro de ese espacio. Es decir, la persona no tiene miedo, por ejemplo, del ascensor, sino de lo que podría sucederle cuando se encuentra dentro de él, ya sea que desplomara, que se averíe y quede encerrado en éste, y peor aun que se agote el oxígeno en su interior.

Además se señala que la claustrofobia incluye dos componentes de miedo: el miedo a la restricción es decir, al confinamiento y el miedo al ahogo, dicho de otra forma a la sensación de falta de aire, de asfixia. Estos dos componentes, se presentan por separado en las personas no claustrofóbicas, pero se presentan conjuntamente en las claustrofóbicas.

¹⁷ Etimológicamente la agorafobia es el miedo a los espacios abiertos, pero en general el agorafóbico siente un miedo irracional a cualquier situación en la que se sienta desprotegido.

En lo que se refiere al primer componente, el miedo a la restricción, hay que señalar que la mayoría de los espacios cerrados suponen, en un grado variable, una limitación de movimientos, o bien dan la sensación de sentirse atrapado.

En cuanto al segundo componente, el miedo al ahogo, es una característica bastante común incluso en muchas personas que no padecen de claustrofobia. Este miedo puede segmentarse en tres grupos: la creencia de que no hay suficiente aire disponible en el espacio en el que se encuentra la persona, que el acceso al aire está bloqueado, o que la persona padece algún problema fisiológico que le impide respirar normalmente por ejemplo, las fosas nasales obstruidas, problemas asmáticos, etc.

Éste miedo, aunque exagerado en alguien con claustrofobia, puede ser comprensible, ya que la mayoría de las personas que no padecen de claustrofobia también sobreestiman la cantidad de oxígeno que necesitan para respirar. Hay que resaltar que el miedo al ahogo implica un miedo a un acontecimiento interno, en este caso, una sensación corporal, algo que también está presente en el trastorno de pánico. De hecho, en bastantes casos la persona con claustrofobia puede experimentar ataques de pánico en situaciones de encierro. Es más, diversos autores como Curtis, Hill, y Lewis, han considerado que la claustrofobia comparte algunas características con el trastorno del pánico, ya que las sensaciones corporales que relatan los pacientes claustrofóbicos son muy similares a las que relatan los pacientes con trastornos de pánico, además de que los claustrofóbicos suelen experimentar más síntomas fisiológicos y cognitivos que las personas que sufren otras fobias específicas.

También existen muchas similitudes y puntos de contacto entre la claustrofobia y la agorafobia, y algunos autores han considerado que ambos trastornos son funcionalmente equivalentes, aunque en la claustrofobia se da una evitación más limitada, por el mismo contexto en que se desarrolla.

Además del temor y la preocupación, las personas con claustrofobia también suelen experimentar sensaciones corporales en las situaciones temidas.

Por ejemplo, palpitaciones, temblores, sudoración, molestias gastrointestinales, tensión muscular, confusión, etc. Todos estos síntomas suelen desaparecer de un modo muy rápido en cuanto la persona abandona la situación de confinamiento, excepto en aquellos casos en que la persona haya vivido una experiencia muy prolongada o muy traumática.

Normalmente, la persona con claustrofobia intenta evitar todas estas situaciones, es decir, no se enfrenta a aquello que teme. Otras veces, aunque se enfrente a esas situaciones, lo hace con un gran esfuerzo, ya que en ellas experimenta una intensa ansiedad, en forma de sensaciones corporales y pensamientos negativos y, por lo tanto, puede que intente protegerse de algún modo, como por ejemplo, situarse cerca de una ventana, para evitar la sensación de encierro, etc. Por otro lado, también puede aparecer ansiedad anticipada, esto es, que mucho antes de que la persona tenga que enfrentarse a la situación temida, ya aparecen esas sensaciones corporales y pensamientos perturbadores.

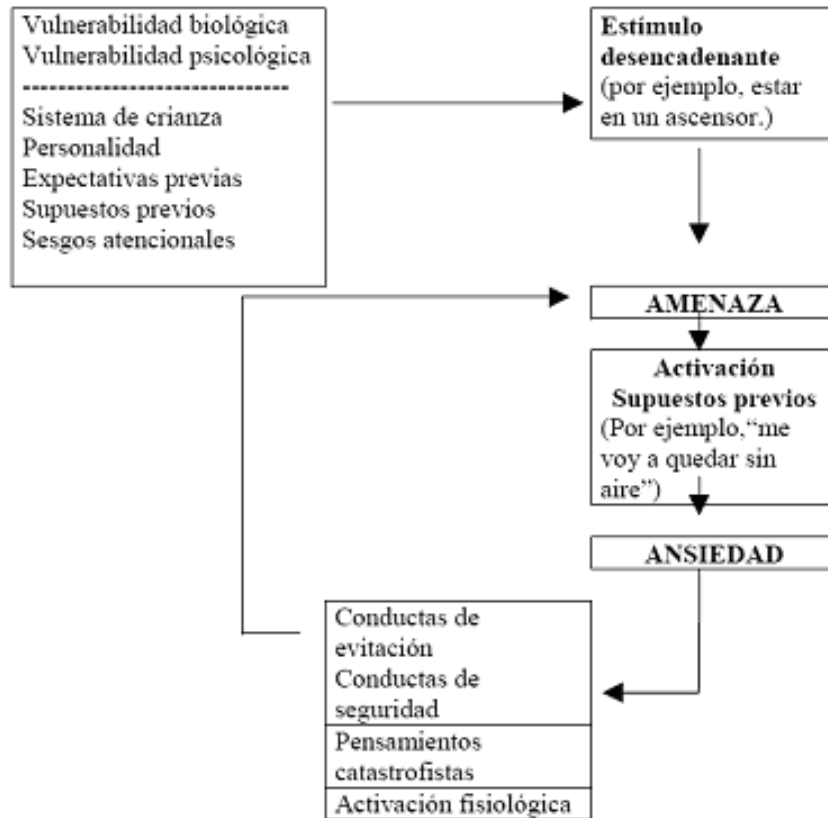
La claustrofobia puede surgir de manera lenta y progresiva; por ejemplo, observando cómo responden otras personas en la situación temida, o mediante noticias catastróficas que relatan a alguien en situaciones en las que se ha quedado encerrado. Pero también puede que se inicie de forma brusca e instantánea, tras vivir una situación amenazante o muy desagradable. Este último parece ser el modo más frecuente de adquisición de la claustrofobia, ya que según los trabajos de Öst y Hugdahl el 67.7% de los claustrofóbicos mencionan que el inicio de su problema se desarrolló a partir de haber vivido alguna experiencia traumática relacionada con espacios cerrados.

Algunas personas sienten malestar al estar en un sitio cerrado y, sin embargo, no desarrollan claustrofobia y otras han sufrido un acontecimiento desagradable, o incluso bastante traumático, y tampoco en su caso surge este problema. No se puede hablar de un solo acontecimiento o aspecto que explique el surgimiento de la claustrofobia, sino más bien de un conjunto de factores que pueden dar lugar a que aparezca y se consolide este trastorno tal como se ilustra en la figura 29.

Como se ve en la figura del modelo explicativo de la claustrofobia, todo parte de la vulnerabilidad biológica. Las personas difieren en cómo responden a los acontecimientos que pongan en riesgo su vida o al estrés.

En el caso de los seres humanos el miedo a la restricción física se puede considerar como un vestigio, es decir, como una huella que ha quedado de la evolución filogenética de nuestra especie.

Figura 29. **Modelo explicativo de la Claustrofobia**



Por lo tanto, se puede considerar que la restricción física es intrínsecamente aversiva, es decir, amenazante por naturaleza.

Sin embargo, no todas las personas desarrollan claustrofobia, sino que algunas son especialmente sensibles o vulnerables a éste tipo de experiencias y como consecuencia, es más probable que estas personas desarrollen éste tipo fobia específica.

También tiene una gran importancia la vulnerabilidad psicológica, es decir, aspectos aprendidos por el ser humano a lo largo de su vida.

Cada persona tiene distintos modos de reaccionar y de afrontar las situaciones, dependiendo de la educación que se ha recibido, es decir dependiendo de su sistema de crianza. Todos estos factores pueden desencadenar el trastorno o consolidarlo.

Cuando la persona con claustrofobia se enfrenta a la situación temida se activan todas las señales de alarma en su cerebro, ya que para ella esto constituye un gran peligro, una gran amenaza que inmediatamente desencadena ansiedad. Esto debido a que en su cabeza se ha formado una representación acerca de cómo cree que actuará en esa situación, lo que le sucederá en un espacio cerrado y la dificultad de recibir ayuda si sucede algo.

Poco a poco, la persona va entrando en un círculo vicioso de difícil salida. Cada vez se produce más ansiedad y la persona con claustrofobia está aún más alerta y pendiente de aquellos aspectos que le resultan más amenazantes. Todo esto lleva a que la persona con claustrofobia intente evitar y así apartarse de situaciones que le provocan tanto malestar.

Además, una vez el problema se ha establecido, la persona empieza a anticipar todas esas catástrofes que teme: va a ocurrir algo terrible, va a perder el control, va a quedarse sin aire o, incluso, se va a morir.

Definitivamente ésta anticipación acentúa más el deseo de evitar esas situaciones que le causan miedo y preocupación, y si no las evita, las soporta con gran malestar.

Sin embargo, la evitación, realmente, no hace más que reforzar el miedo, mantener el círculo. Cuanto más se eviten las situaciones temidas, más se acentúa la fobia.

4. APLICACIÓN DE LA REALIDAD VIRTUAL

4.1 Aplicaciones de la realidad virtual en psicología

La realidad virtual ha invadido ya gran cantidad de campos, que van desde el entretenimiento bélico hasta las simulaciones con las que se puede entrenar para el manejo de diferentes tipos de vehículos y máquinas. En medicina se dispone ya desde hace varios años de sistemas de realidad virtual mediante los que es posible desde aprender y practicar técnicas quirúrgicas, hasta llegar a realizarlas remotamente.

En psicología, las aplicaciones principales que se han desarrollado hasta este momento tienen que ver con técnicas de exposición empleadas habitualmente para el tratamiento de las fobias, pero se ha avanzado también en otros campos como los trastornos alimenticios.

4.2 Tratamiento de fobias mediante realidad virtual

Probablemente el área de mayor desarrollo de las aplicaciones actuales de la realidad virtual sobre trastornos mentales es la de las fobias. Entre los pioneros de la investigación en este campo, y que han continuado haciendo aportaciones hasta la actualidad, se encuentran M.M. North, y S.M. North, de la Clark Atlanta University; Barbara Rothbaum y Larry Hodges, del Georgia Institute of Technology; y el grupo de Cristina Botella, de la Universidad Jaime I de Castellón. M.M. North y B.

North, M.M.; North, S.M. y Coble, J.B. se consideran como el primer equipo en aplicar la realidad virtual para el tratamiento de desordenes psicológicos. Ellos incursionaron en este modelo de tratamiento al aplicar escenarios virtuales al tratamiento de la fobia a volar, la que es de una de las fobias más recurrente y que tiene serias repercusiones en las personas que la padecen, pues esta restringe sus capacidades de locomoción, afectando con esto sus relaciones interpersonales así como de trabajo; y en un mundo cada día más globalizado esto implica serias limitantes.

Este tipo de padecimiento es especialmente idóneo para su reproducción en una clínica mediante dispositivos de realidad virtual, ya que las condiciones son sumamente parecidas, por ejemplo dentro del avión el sujeto se encuentra sentado en un espacio limitado, al igual que lo hace en la clínica. Desde otra perspectiva los costos del tratamiento virtual son exponencialmente más bajos que un tratamiento convencional. Pues para nadie es desconocido el alto precio de un boleto aéreo.

Como un ejemplo de software de realidad virtual aplicado a la cura del miedo a volar se puede mencionar Virtual Flight desarrollado por el instituto PREVI de España.

Asimismo se ha realizado tratamientos a pacientes con diferentes desordenes como por ejemplo, miedo a los espacios abiertos y estrés post guerra.

En estos experimentos se han seleccionado a grupos de pacientes los que son expuestos a una serie de sesiones en los que mediante técnicas de realidad virtual, son expuestos a las situaciones que les provocan fobia.

En las sesiones iniciales los sujetos presentan altos niveles de arousal fisiológico y de ansiedad los que en promedio luego de ocho sesiones disminuyen y en la mayoría de casos llegan a valores de cero.

Las medidas de arousal son un concepto hipotético que miden el grado de activación fisiológica y psicológica de un cuerpo, según el cual, se puede predecir el desempeño de un sujeto tomando como principio que al tener un arousal óptimo, se tiene un rendimiento óptimo y al tener un arousal sobre-activado o sub-activado, se va a tener un rendimiento bajo

Aun y cuando estos experimentos se han realizado en paralelo con métodos tradicionales y ambos han obtenido los mismos resultados, los resultados obtenidos mediante realidad virtual, han sido más intensos y prolongados.

Todos estos datos fueron presentados en el documento Aplicaciones de la realidad virtual en Psicología Clínica de José Gutiérrez Maldonado, al que se puede referir para más ejemplos de la utilización de realidad virtual en el tratamiento de desordenes mentales.

4.3 Tratamiento de la claustrofobia

La mayoría de estudios señalan que el tratamiento más eficaz para las fobias específicas, incluida la claustrofobia, es la exposición *in vivo* a las situaciones temidas; es decir, que el paciente se vaya enfrentando de una manera gradual y con la ayuda de un profesional, a aquello que teme.

Como se indicó al comienzo de este trabajo, la realidad virtual presenta una serie de ventajas frente a este tipo de procedimiento, como la confidencialidad, la economía (tanto en términos de dinero como de tiempo del terapeuta), etc. Además, los estudios realizados hasta el momento señalan que no todos los pacientes se benefician de las técnicas de exposición *in vivo*, ya que algunos de ellos sienten tanto temor de enfrentarse a la situación temida, que llegan a rechazar el tratamiento cuando se les propone, o abandonan la terapia una vez iniciada. Es más, incluso para algunos que aceptan el tratamiento, éste puede volverse muy agresivo, lo que impide que se sientan totalmente seguros, como si para ellos no pudiera existir la certeza de que algo no funcione mal.

Una ventaja muy importante de la realidad virtual es la seguridad que proporciona. Ya que los pacientes pueden controlar el ambiente generado por el programa de computadora a su voluntad y sin riesgos, claro siempre que la terapia lo permita. En este sentido, la realidad virtual se convierte así en un paso intermedio muy importante entre la consulta del terapeuta y el mundo real.

Además, se ha señalado que algunos pacientes que siguen un programa de exposición *in vivo* continúan manifestando miedos residuales en la situación temida.

El control que proporciona la realidad virtual y la posibilidad de ir más allá de lo que la situación real permitiría, son factores que promueven el sentimiento de auto eficacia del paciente y le ayudan a eliminar estos miedos residuales.

4.4 Escenarios virtuales de un programa para el tratamiento de la claustrofobia

Los programas de realidad virtual son concebidos a través de escenarios los que va recorriendo el sujeto a lo largo de un número de sesiones durante su tratamiento, hasta que se logra eliminar su alto nivel de ansiedad; en el presente capítulo se identificarán los que más se aplican en el tratamiento de la claustrofobia y se describirán las funciones o características con que cuentan dichos escenarios. El programa modelo para el presente capítulo y del que se toman la mayoría de funciones además de las impresiones de pantalla es *Virtual OpenOut* desarrollado por el Instituto PREVI de España.

4.4.1 Primer escenario, la casa

En este escenario, el paciente se encuentra en una casa, en la que hay dos habitaciones, la sala y la habitación mágica.

La sala

Se trata de una habitación de 4x5 metros, con una puerta de entrada a la que le precede un espacio abierto conocido como terraza.

Dicha habitación está decorada como si fuera la sala de una casa promedio, entiéndase muebles, cuadros, un televisor, etc. En la habitación hay una ventana grande de dos hojas con una persiana. También hay dos puertas, una que comunica con la terraza, a la que no se puede salir y otra que comunica con la habitación mágica. Como se ilustra a continuación.

Figura 30. Primer escenario la sala



En este escenario se puede manipular tanto la ventana como las puertas. Las persianas de la ventana se pueden bajar y subir, con el consiguiente decremento o aumento de claridad, la luz de ambiente de la sala queda definida por el grado de apertura tanto de la persiana como de la puerta que da a la terraza. Tal y como se muestra en las dos ilustraciones subsecuentes.

También se puede abrir o cerrar la ventana en tres tramos, con el consiguiente aumento o decremento de ruido del exterior, entre los sonidos que causan este ruido se encuentran entre otros, el aire, sonido de pájaros, murmullo de personas, bocina de automóviles etc. Desde la ventana se puede ver el exterior, el cielo, los edificios de enfrente, la calle, etc.

La puerta de la terraza se puede abrir o cerrar en tres tramos. Cuando se abre, se puede ver una terraza con plantas y el exterior de la vivienda. También hay disminución y aumento de la luz y el sonido, como ocurre cuando se abre la ventana.

Figura 31. Iluminación alta en la sala, con la persiana abierta



Figura 32. Sala con poca iluminación por estar cerrada la persiana



Por su parte la puerta de la habitación mágica se puede abrir o cerrar en tres tramos para pasar a la habitación mágica.

Si el psicólogo terapeuta desea que el paciente se enfrente a una situación medioambiental de buen tiempo, el software proporcionara la opción de pulsar un conjunto de teclas para modificar tal condición o bien retornar a ésta. Por defecto, la situación medioambiental es de buen tiempo, por lo tanto, inicialmente no será necesario pulsar la tecla definida para dicha acción. Por el contrario si el psicólogo desea que el paciente se enfrente a una situación medioambiental de mal tiempo, deberá pulsar una tecla distinta.

Cuando se selecciona la situación medioambiental de buen tiempo se apreciará a través de la ventana un cielo despejado. Asimismo, al abrir la ventana o la puerta del balcón, tal y como se especificó anteriormente, puede percibirse el sonido del exterior de cantos de pájaros y tráfico; la iluminación puede observarse en la siguiente captura de pantalla.

Por el contrario cuando se selecciona la situación medioambiental de mal tiempo se apreciará a través de la ventana un cielo gris oscuro. Además se escuchará lluvia, viento y truenos, sumándosele el sonido de tráfico cuando se abra la ventana o la puerta del balcón. En esta situación, el psicólogo podrá empeorar la percepción del clima todavía más, reproduciendo rayos mediante la pulsación de alguna tecla especificada para tal efecto.

Figura 33. **Situación medioambiental de buen tiempo**



El psicólogo podrá pulsar la tecla tantas veces como crea necesario. No obstante, es conveniente que entre pulsación y pulsación deje un tiempo prudencial de aproximadamente unos 10 segundos, para que al sonido del trueno lo acompañe la imagen respectiva del relámpago, tal y como se indica en la guía terapéutica del *software*. El resplandor de los rayos puede observarse en la siguiente imagen:

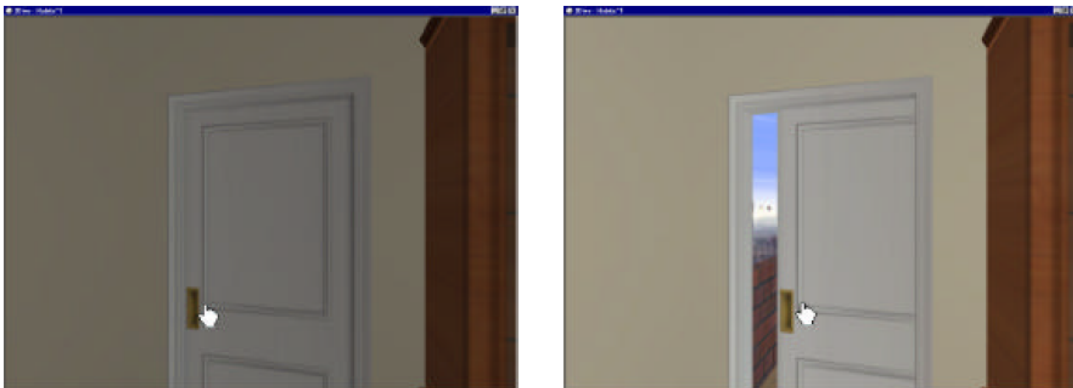
Figura 34. **Resultado de mal tiempo y de rayos en la escena**



Por su parte la puerta de la terraza se puede abrir o cerrar en tres tramos. Cuando se abre, se puede ver una terraza y el exterior de la vivienda (el cielo, los edificios de enfrente, la calle, etc.).

Al abrir o cerrar esta puerta (Figura 35), se producirá el consiguiente aumento o disminución en la intensidad de la luz ambiente del salón, la que recapitulando está definida por la situación o grado de apertura tanto de la persiana como de la puerta que da a la terraza, y al igual como sucede con la ventana, también variará la intensidad de los sonidos procedentes del exterior.

Figura 35. Iluminación resultante de abrir o cerrar la puerta



La habitación mágica

Esta habitación mide 3x3 metros y no tiene ni muebles ni ventanas. Tanto el techo como el suelo tienen una textura pensada para dar una mayor sensación de encierro. La puerta se puede cerrar en tres tramos. Además, un cerrojo situado en la parte interior de la puerta, bloquea ésta, de tal modo que no se puede volver a abrir hasta que se vuelva a desbloquear.

Por otro lado, ésta habitación permite al paciente ir más allá de la realidad, ya que en ella el espacio se puede hacer cada vez más y más pequeño desplazando gradualmente una de las paredes, mediante la pulsación de un botón por parte del terapeuta. Este desplazamiento ocasiona un gran ruido, que aumenta la sensación de cierre.

Figura 36. La habitación mágica en dos variantes de su tamaño



La pared podrá moverse hasta que la habitación tenga tan solo un metro de ancho, como se muestra en la imagen anterior.

4.4.2 Segundo escenario, el ascensor

Este escenario simula la apariencia y comportamiento de las aéreas comunes de un edificio, esto es, el vestíbulo, los descansos de cada piso y el ascensor.

En la situación inicial, el usuario se encuentra en el vestíbulo, de frente al ascensor y se visualiza la ilustración 37.

Este escenario está dividido en dos componentes, el vestíbulo y el ascensor.

El vestíbulo es un amplio espacio en el que hay una gran cristalera a través de la cual se ve el exterior, en donde se observan árboles, el cielo, etc.

Figura 37. El vestíbulo



Además de la vista, a éste componente lo acompaña un sonido exterior agradable de pájaros. Y es aquí, donde el paciente tiene acceso al ascensor.

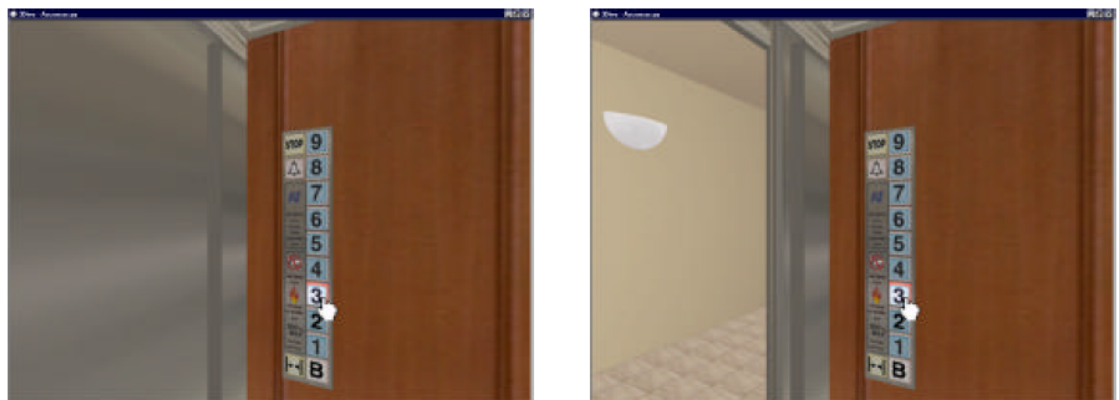
El ascensor a su vez se ha diseñado para ofrecer diferentes posibilidades variando el grado de amenaza, teniendo en cuenta diversos parámetros como por ejemplo tamaño, iluminación, posición, posibilidad de bloqueo y averías.

El ascensor mide 1x2 metros, y está situado inicialmente en la planta baja con la puerta abierta. Cuando el paciente o sujeto entra en el ascensor, las puertas se cierran, y un sonido característico acompaña esta acción para incrementar la sensación de realismo. El ascensor está detenido y el paciente decide si desea subir o no a un piso determinado. Si no se desea subir a ningún piso y se quiere salir, existe un botón por medio del cual se abren las puertas, lo que sólo se puede hacer si el ascensor no está en marcha.

Si se desea subir a un piso, se habrá de pulsar el correspondiente botón (la guía terapéutica del programa recomienda 10 niveles como mínimo). Obviamente, el tiempo del desplazamiento variará en función del piso seleccionado. De este modo se puede graduar, mediante el tiempo, la dificultad de la situación.

Una vez pulsado el botón, el ascensor se pone en marcha y cada vez que se pase por un piso, se iluminará el botón correspondiente, así la persona puede, saber el tiempo que le falta para llegar a su destino. Cuando se llega al piso seleccionado, las puertas se abren y se puede salir al descanso de cada nivel. Una vez en el descanso, si se desea volver a tomar el ascensor, se pulsa el botón de llamada (figura 38), se espera a que llegue y se abran las puertas y se puede volver a entrar e ir a otro piso o al vestíbulo.

Figura 38. Selección de nivel en el ascensor



Mientras el ascensor esté funcionando, se podrán hacer dos variaciones, que son las siguientes:

- **Avería:** Cuando el terapeuta lo estime oportuno, y mediante la pulsación de una tecla del teclado, se podrá provocar una avería que producirá la parada y bloqueo repentino del ascensor, junto con un movimiento brusco, un sonido característico, el apagado de las luces y el bloqueo de las puertas (figura 39).

La duración de esta situación dependerá del terapeuta, quien podrá desbloquear el ascensor con una tecla cuando lo estime oportuno. Después, el ascensor seguirá su camino hacia el piso seleccionado.

Figura 39. **Desperfecto en el ascensor**



- **Reducción del tamaño del ascensor:** este tipo de avería exagera la situación real que puede vivirse en un ascensor, para provocar un aumento de ansiedad en el paciente y la correspondiente estimulación de la claustrofobia.

- El médico puede manipular el tamaño del ascensor presionando dos botones desde su teclado de control. Uno de ellos hace que una de las paredes se mueva, con un gran ruido, y se reduzca el tamaño del ascensor hasta tener 1 metro de ancho. El otro botón sirve para que la pared vuelva a su posición original, adquiriendo el ascensor la dimensión inicial.

Una vez que el ascensor está bloqueado por el fallo en el funcionamiento, existen dos posibilidades manipulables por el terapeuta; la primera consiste en la restauración de las condiciones de iluminación. Esta opción se consigue pulsando una tecla de función del teclado.

En esta situación, el ascensor sigue bloqueado, por lo que la pulsación de los botones por parte del paciente no tendrá repercusión alguna en el escenario. Y la otra opción es el desbloqueo del ascensor, permitiendo todas las acciones comentadas anteriormente. Es entonces que la avería se encuentra completamente revertida, pudiendo salir el paciente del ascensor y bajar o subir a cualquier nivel.

4.5 Aspectos informáticos de programación

A pesar de la complejidad informática que pareciera tener el paquete antes expuesto, un ligero vistazo a su estructura revela todo lo contrario; los componentes que conforman la aplicación se resumen a continuación:

4.5.1 3Dive

Del anglosajón 3D *Interactive Volume Explorer* o Explorador Interactivo de Volúmenes de tres dimensiones, es una herramienta de visualización para el despliegue y manipulación de información volumétrica. Se desarrollo por AVL (*advanced visualization laboratory*) en asociación con el departamento de Ciencias de la Computación de IUPUI (*Indiana University Purdue University Indianapolis*).

Ésta herramienta es el primer entorno inmersivo de visualización de volúmenes que implementa las regiones volumétricas. Las principales funcionalidades de ésta son la inclusión de procesamiento interactivo de imágenes y la transferencia de funciones de diseño, edición y rebanado de imágenes en dos dimensiones.

La alimentación de los datos en 3Dive puede llevarse a cabo de diversas formas desde indicar primitivas básicas hasta la integración con microscopios electrónicos.

La principal aplicación es la representación del cuerpo humano con fines médicos, es tan así que entre los conjuntos de datos predefinidos del programa se encuentran cerebros, espinas dorsales, rodillas entre otros.

Y es ésta la herramienta empleada para la visualización de los escenarios dentro del software, este componente es independiente del sistema operativo y es fácilmente integrable con el lenguaje de programación *python*.

4.5.2 Python

Python es un lenguaje interpretado, es decir que necesita de un intérprete para ser ejecutado, el cual fue creado como sucesor del lenguaje ABC por Guido van Rossum en el Stichting Mathematisch Centrum (CWI). El nombre del lenguaje proviene de la afición de su creador original, por los humoristas británicos de Monty Python.

Python es un lenguaje de programación dinámico orientado a objetos que puede ser utilizado para desarrollar diversos tipos de software, ofrece un gran soporte de integración con otros lenguajes de programación y herramientas; además que incluye diversas librerías de uso estándar.

Este lenguaje de programación es fácil de embeber con C, ya que con el que fueron programadas las librerías de OpenGL, es entonces este el eje central de la aplicación *Virtual OpenOut*.

4.5.3 OpenGL

Refiérase a la sección 2.4.1.2 y 2.4.1.3

4.6 Incorporación de dispositivos de realidad virtual a la clínica

En este apartado se presenta una pequeña reseña de dos de los periféricos adicionales a la computadora personal, que más se utilizan en la actualidad. El enfoque que se da es exclusivo para los dispositivos visuales y auditivos que enriquecen la experiencia del usuario.

Tales dispositivos permitirán incrementar considerablemente en el paciente la sensación de inmersión en el escenario virtual, acrecentando así el nivel de realismo y llegar así a tener virtualmente una exposición *in vivo*.

A demás de describir someramente tales periféricos, se adherirá a esta descripción el precio de mercado al que pueden comprárseles.

Cabe destacar que existen distribuidores para nuestro país y por eso el precio es expresado en la moneda nacional (Quetzales).

4.6.1 Gafas de realidad virtual i-glasses:

Los HMDs (*Head Mounted Displays* - Pantallas con soporte de cabeza) i-glasses crean una ilusión de una imagen de 76 pulgadas, a una distancia de 5 metros delante del usuario. Por lo que son altamente recomendados para aplicaciones de realidad virtual ejecutadas en una PC.

. I-glasses es un monitor portátil de computadora, accesible, de alta resolución, en un empaque compacto de 200 gramos que se encaja cómodamente en la cabeza del usuario (figura 22). Poseen además sonido estéreo a través de auriculares integrados que completan la experiencia virtual.

El precio de este periférico varia en un rango de Q 7,000.00 hasta Q 10,000.00

4.6.2 Gafas de realidad virtual Sony Glasstron:

Los HDM's Glasstron están provistos de audífonos estereofónicos y el sistema óptico consiste de dos pantallas LCD (*Liquid Crystal Display* – Pantalla de Cristal Liquido) de 0.7 pulgadas, cada una con 1.5 millones de puntos ópticos o píxeles.

La ilusión para el usuario es de estar frente a una pantalla de 52 pulgadas con una resolución de 800x600. Los anteojos también tienen una función de transparencia, que permite ver a través de las pantallas de LCD.

El peso de este periférico supera por poco los 100 gramos, lo equivalente a tres pares de lentes para el sol, promedio (figura 40). Y su precio oscila por los Q 6,500.00.

Figura 40. Gafas de realidad virtual i-glasses y HDM Sony Glasstron



CONCLUSIONES

1. La integración y abaratamiento de distintas tecnologías tales como el audio y el video, entre otras convierten a la realidad virtual, en una técnica efectiva y además económica para el tratamiento de distintos trastornos mentales, como la claustrofobia.
2. La realidad virtual inmersiva es la mejor opción para aplicarse en tratamiento de desordenes mentales, por el grado de presencia que logra recrear en sus usuarios.
3. Los programas informáticos para el tratamiento de fobias se dividen en escenarios, que son facetas por las que el paciente atraviesa gradualmente, conforme sus niveles de ansiedad disminuyen. Los escenarios de los paquetes de computadora para el tratamiento de la claustrofobia, son principalmente, una sala de estancia, un elevador y algunos agregan una habitación mágica.
4. Aún y cuando las técnicas tradicionales de exposición brindan prácticamente los mismos resultados, el tratamiento por realidad virtual acorta los tiempos de mejora y prolonga los resultados. Tanto en términos de dinero como de tiempo del terapeuta, la realidad virtual resulta ser una opción económicamente más viable frente las técnicas de exposición *in vivo*.

RECOMENDACIONES

1. Todo psicólogo clínico que se dedique al tratamiento de fobias específicas debe invertir en este tipo de tecnología para abaratar sus costos.
2. Para obtener mejores resultados en la implementación de una clínica equipada con dispositivos y *software* de realidad virtual, se debe contar con sistemas aislantes de estímulos externos.
3. Antes de iniciar una terapia psicológica es necesario que el terapeuta explique los por menores de los escenarios de esta tecnología para lograr con esto una mejor aceptación por parte del paciente.
4. Dentro del pensum académico de Ingeniería en Ciencias y Sistemas considero obligatorio incluirse más materias que refuercen y especialicen a los estudiantes en el manejo de *software* para el procesamiento de imágenes, así como de animación que se utilizan para desarrollar paquetes de RV.
5. Es importante tanto que los terapeutas como los implementadores de tecnología determinen los proveedores calificados del equipamiento electrónico computacional a emplearse en la realidad virtual considerando costos y servicios ofrecidos por éstos.

BIBLIOGRAFÍA

1. Burdea, Grigore C., Coiffet, Philippe. **Virtual reality technology**. 2da. Edición. Estados Unidos de Norte América: Wiley-IEEE Press, 2003. 464 pp.
2. del Pino, Luis M. **Realidad virtual**. España: Paraninfo, 1995. 210 pp.
3. Dorsh Friedrich. **Diccionario de psicología**. 2da Edicion. España: Editorial Herder, 1977. 143 pp.
4. Gutiérrez Maldonado, Jose. “Aplicaciones de la realidad virtual en la psicología clínica” **Grupo Aula Medica**. (España) (1):2002
5. Jeffrey Kluger. “Ya no tenga miedo” *Selecciones Reader’s Digest*. (Estados Unidos de Norte América) (736):2002.68
6. K, Brenda. **Virtual therapy for anxiety disorders: advances in evaluation treatment**. Estados Unidos de Norte America: American Psychological Association. 2004. 225 pp.
7. Kilgard, Mark J. **OpenGL programming for the X window system**. Estados Unidos de Norte América: Addison-Wesley Professional, 1996. 576 pp.
8. Kolb, Lawrence C. **Psiquiatría clínica moderna**. México: La Prensa Medica Mexicana, 1980. 834 pp.
9. Luebke, David y otros. **Level of detail for 3D graphics**. Estados Unidos de Norte América: Morgan Kaufmann, 2002. 432 pp.
10. Lyne, Sharon Edgar. **Future of health care**. Estados Unidos: iUniverse, 2003. 86 pp.

11. Max M, North y otros. **Virtual reality therapy: an innovative paradigm.** Estados Unidos de Norte America: IPI Press, 1996. 221 pp.
12. Rheingold, Howard. **Virtual reality: the revolutionary technology of computer-generated artificial worlds- and how it promises to transform society.** Estados Unidos de Norte America: Simon & Schuster, 1992.416 pp.
13. Velez Janh, Gonzalo. "Introducción a VRML" **Arquitectura-L virtua-L.** (Venezuela) (1):1997.18
14. Vince, John. **Introduction to virtual reality.** Estados Unidos de Norte América: Springer, 2004.163 pp.
15. Wiederhold, Brenda. **Conquering panic, anxiety, and phobias: achieving success through virtual reality and cognitive-behavioral therapy.** Estados Unidos de Norte America: Virtual Reality Medical Center, 2004. 131 pp.