

IMPLEMENTACIÓN DE UN MANUAL DE TÉCNICAS PARA EL
DESARROLLO DE JUEGOS DE VIDEO MEDIANTE LA UTILIZACIÓN
DE MOTORES GRAFICOS 3D PARA LOS ESTUDIANTES DE LA
FACULTAD DE INGENIERIA DE SISTEMAS EN LA CORPORACION
EDUCATIVA MAYOR DEL DESARROLLO SIMON BOLIVAR

BARROS ERNESTO
MONTAÑO GERMAN
PERALTA WALTER
RODRIGUEZ RICARDO
VALEGA ALFONSO

TRABAJO PRESENTADO COMO REQUISITO FINAL EN LA
ASIGNATURA DE INVESTIGACIÓN FORMATIVA IV

CORPORACIÓN EDUCATIVA MAYOR DEL DESARROLLO
SIMÓN BOLÍVAR
BARRANQUILLA
2005



AGRADECIMIENTOS

Este manual es el resultado de un trabajo grupal, que fue posible gracias a la voluntad decidida que tuvimos durante todo este proceso de desarrollo del proyecto, es así como hoy estamos en la etapa final de nuestra formación básica como Ingenieros de Sistemas.

Agradecemos a Dios por darnos todo aquello que tenemos, a todas las personas que nos apoyaron durante este proceso de formación a nuestros tutores por asesorarnos, a todos los profesores por brindarnos siempre su apoyo cuando fue necesario, a la institución por permitirnos un espacio donde realizarnos como personas y profesionales; pero en especial a nuestros padres que siempre nos han guiado y apoyado en el camino de la vida para alcanzar nuestros logros, objetivos y metas, haciendo de cada uno de nosotros mejores personas, para que algún día, demos al mundo el valor que tenemos y todo lo que podemos construir gracias a su apoyo, y así ser de servicio a la sociedad.

INDICE

1. CAPITULO 1. INTRODUCCION A LOS VIDEOJUEGOS.	1
1.1 HISTORIA DE LOS VIDEOJUEGOS	1
1.2 CONSOLA DE VIDEOJUEGOS	2
1.3 GENEROS DE LOS VIDEOJUEGOS	7
2. CAPITULO 2. INTRODUCCIÓN A LOS MOTORES GRAFICOS.	11
2.1 HISTORIA DE LOS MOTORES GRAFICOS	11
2.2 CATEGORIZACION DE LOS MOTORES GRAFICOS	12
2.3 ATRIBUTOS DE UN MOTOR GRAFICO	12
2.3.1 API	13
2.3.2 ILUMINACION	13
2.3.3 POLIGONOS	13
2.3.4 NIEBLA	13
2.3.5 MIPMAPPING	14
2.3.6 TEXTURAS	14
2.3.7 COLOR	14
2.3.8 SONIDO	14
2.4 EFECTOS GRAFICOS DE LOS MOTORES GRÁFICOS	14
2.4.1 ANTIALIASING	14
2.4.2 BUMP MAPPING	15
2.4.3 COMPRESIÓN DE TEXTURAS	16
2.4.4 ENVIRONMENT MAPPING (MAPEADO DE ENTORNO)	17
2.4.5 MIP MAPPING	17
3. CAPITULO 3. REQUERIMIENTOS DE HARDWARE Y MOTORES GRAFICOS UTILIZADOS.	19
3.1 DISTINTAS CLASES DE HARDWARE LUDICO	19
3.2 PROCESADORES	20
3.3 PROCESADORES GRAFICOS	21
3.4 TARJETA DE SONIDO	25

3.5 CARACTERISTICAS MOTOR GENESIS 3D	28
4. CAPITULO 4. INTRODUCCION A LAS INTERFACES REQUERIDAS PARA EL DESARROLLO DE LOS JUEGOS DE VIDEO 3D.	29
4.1 HERRAMIENTAS PARA EL DISEÑO 3D	29
4.2 EDITOR DE NIVELES	29
4.3 EDITOR DE EFECTOS PARA EFECTOS EXTERIORES	30
4.4 EDITOR DE PERSONAJES	30
5. CAPITULO 5. RECONOCIMIENTO DEL AMBIENTE DE TRABAJO DEL MOTOR GRAFICO Y DEL EDITOR DE EFECTOS EXTERIORES	32
5.1 ENTORNO DE TRABAJO	32
5.2 INTERFAZ DEL EDITOR DE NIVELES	32
5.3 INTERFAZ DEL EDITOR DE PERSONAJES	37
5.4 INTERFAZ DEL EDITOR DE EFECTOS EXTERIORES	38
6. CAPITULO 6. CREACION DE ENTORNOS 3D Y EDITOR DE NIVELES	39
6.1 CREACION DE ENTORNOS 3D	39
6.2 EDITOR DE NIVELES	43
7. CAPITULO 7. CREACION Y MANEJO DE OBJETOS SÓLIDOS.	46
7.1 CREACION Y EDICION DE HABITACIONES	46
7.1.1 BLOCK ACTOR	46
7.1.1.1 BOX	46
7.1.1.2 SPHEROID	46
7.1.1.3 CYLINDER	47
7.1.1.4 STAIRCASE	47
7.1.1.5 ARCH	47
7.1.1.6 CONE	47
7.1.2 BOT ACTOR START	48
7.1.3 BOT MATCH START	48
7.1.4 CORONA	48
7.1.4.1 DEATH MATCH START	48
7.1.4.2 DOOR	48

7.1.5 DYNAMIC LIGHT	48
7.1.6 ELECTRIC BOLT	48
7.1.7 ELECTRIC BOLT TERMINUS	49
7.1.8 FOGLIGHT	49
7.1.9 ITEM ARMOR	49
7.1.10 ITEM GRENADE	49
7.1.11 ITEM GRENADE AMMO	49
7.1.12 ITEM HEALTH	49
7.1.13 ITEM ROCKET	49
7.1.14 ITEM ROCKET AMMO	50
7.1.15 ITEM SHREDDER	50
7.1.16 ITEM SHREDDER AMMO	50
7.1.17 LIGHT	50
7.1.18 SPOT LIGHT	50
7.1.19 CREACION DE UNA HABITACION	50
7.1.20 AGREGAR BLOCK ACTORS AL ESCENARIO	51
8. CAPITULO 8. MANEJO DE CRISTALES, ESPEJOS Y LIQUIDOS.	57
8.1 MANEJO Y EDICION DE CRISTALES	57
8.2 MANEJO Y EDICION DE ESPEJOS	58
8.3 MANEJO Y EDICION DE LIQUIDOS	58
9. CAPITULO 9. ILUMINACION, NIEBLA, CIELOS, RAYOS.	60
9.1 MANEJO DE LA ILUMINACION EN LOS NIVELES	60
9.2 MANEJO DE LA NIEBLA EN LOS NIVELES	64
9.3 MANEJO DE LOS CIELOS EN LOS NIVELES	65
9.4 MANEJO DE LOS RAYOS EN LOS NIVELES	66
10. CAPITULO 10. DISEÑO Y CREACION DE PERSONAJES.	68
10.1 DISEÑO DE PERSONAJES	68
10.2 CREACION DE PERSONAJES	68
10.3 ALTERACION DEL ASPECTO DE LOS PERSONAJES	106
11. CAPITULO 11. MANEJO DE AMBIENTES EXTERIORES.	110
11.1 MANEJO DEL EDITOR DE EFECTOS	110

11.2 CREACIÓN DE RENDERS PARA EXTERIORES	113
12. CAPITULO 12. MANEJO DE EFECTOS SONOROS.	116
12.1 MANIPULACION DE EFECTOS SONOROS	116
13. CAPITULO 13. UNIFICACION DE COMPONENTES.	119





CAPITULO I. INTRODUCCION A LOS VIDEOJUEGOS.

En esta primera parte se plantea la importancia de los videojuegos en la sociedad actual como una herramienta de entretenimiento y avance tecnológico. El desconocimiento acerca de los temas referidos a los videojuegos causa que comúnmente se cometan errores al momento de hablar de los videojuegos subestimando la capacidad tecnológica desplegada en estos productos, las cuales no pueden ser menospreciadas ya que van a la vanguardia en el desarrollo de entornos 3D).

Un Videojuego se puede considerar como una aplicación ya que esta diseñado con otro programa llamado Motor Grafico, pero como ya se sabe los programas cuentan con dos aspectos uno de Código que no es tan importante como el Aspecto Grafico el cual es la raíz de todo Videojuego, pues como se sabe todo aquel que diseñe un Juego de Video y lo haga concienzudamente debe tener en cuenta que para que este sea del agrado de los distintos usuarios debe ser atractivo tanto visual como en su contenido temático, es decir, que genero representa y también su impacto Visual con todo lo que este conlleva, como lo son los efectos de sonido.

1.1 HISTORIA DE LOS VIDEOJUEGOS.

Los videojuegos han tenido han avanzado enormemente desde sus comienzos hasta el día de hoy, casi desde la aparición de los primeros computadores después de la segunda guerra mundial. El primer videojuego fue creado en el año de 1958 por *Bill Nighinbotham* a quien se le considera el primer desarrollador de Videojuegos de la historia; pero el no presto mucho interés a su creación (la cual no le llevo mas de una semana) y no la patento. Esto lo aprovecho un joven llamado Nolan Bushnell lo adopto a un aparato casero y lo bautizo con el nombre de Pong y simulaba el tenis de mesa. En 1961 Steve Russell creó un juego llamado Space Invaders que fue el primero en ser aprovechado comercialmente y fue montado en una máquina Arcade (máquinas de Videojuegos destinadas a salones recreativos). A partir de ese instante comenzaría lo que llegaría a ser la mayor industria de entretenimiento a nivel mundial.

En el año de 1972 Nolan Bushnell funda la compañía Atari y comienza la comercialización del Videojuego Pong en máquinas Arcades y más tarde en un aparato casero llamado Atari Pong, aunque la primera consola fue la Magnavox Odyssey que salió en el año de la fundación de Atari. En el año de 1977 Atari lanzo la consola que lo catapulto al éxito y le creó un lugar en la historia el Atari 2600 que duro en el mercado hasta 1990, siendo considerada la consola que mas tiempo ha estado en el mercado. Esta consola usaba un procesador de 8 bits.

En 1983 la empresa japonesa Nintendo lanzo su primera consola independiente la Famicom en Japón y dos años mas tarde en Estados Unidos llamándola NES (Nintendo Entertainment System) la cual desplazo del mercado a todas las consolas estadounidenses y a la Master System de Sega, logrando vender 60 millones de unidades estando en el mercado hasta 1991.

A finales de los 80s (1989) una empresa japonesa llamada SNK desarrollo un proyecto para lograr un sistema de Videojuegos integral denominado Neo – Geo. Con el cual se diseñaba un hardware de máquinas Arcade que tuviera su correspondiente versión casera. El sistema de Arcade se denominó MVS (Multi Video System) y el casero AES (Advance Entertainment System) lo cual convertía al Neo – Geo en el sistema casero más potente en cuanto a gráficos 2D. El hecho destacable era que al ser iguales tanto el MVS y el AES los juegos eran iguales a la versión Arcade y siendo la primera consola en utilizar más de un procesador (2 para ser exactos), estando vigente hasta el año 2003.

Con la aparición de sistemas de 16 bits a finales de los 80s como el Genesis de Sega o el Neo – Geo de SNK la industria se expandió a nivel mundial haciéndole competencia a Nintendo que en 1991 lanzó la consola SNES (Super Nintendo Entertainment System) abriendo el mercado.

En 1994 se lanzaron los sistemas de 32 bits como el Playstation de Sony (convirtiéndose en la consola más vendida de la historia) y el Sega Saturn (primeras máquinas caseras en presentar gráficos 3D), un año después el 3DO de Goldstar y Panasonic, y el CD-I de Philips. Atari lanzó un sistema de 64 bits denominado Atari Jaguar y en 1996 Nintendo lanzó el Nintendo 64. El hardware siguió desarrollándose hasta evolucionar a sistemas de gran realismo gráfico. En la actualidad las consolas utilizan procesadores de 128 bits siendo el Dreamcast de Sega el primero de ellos en ofrecer juego online, posteriormente fueron lanzados el Playstation2 de Sony, el Gamecube de Nintendo.

1.2 CONSOLAS DE VIDEOJUEGOS.

Las consolas son dispositivos de hardware destinados para la ejecución de Videojuegos (aunque las que han aparecido a partir de 1998 en adelante permiten conexión a Internet, ejecución de películas en DVD, etc.) en Televisores. A continuación se explican las especificaciones técnicas de las más importantes:

→ Atari 2600:

Procesador	Motorola 6507 a 1,19 MHz
Procesador Gráfico	1,79 Mhz
Memoria RAM	128 Bytes
Sonido	Controlado por el chip Stella, 2 canales.
Paleta de Colores	8 de una paleta de15.
Formato de juego	Cartucho
Puertos para mandos	2



→NES (Nintendo Entertainment System):

Procesador	6508 Motorola (8 bits) a 1.79 Mhz
Procesador Gráfico	1,79 Mhz
Memoria RAM	2 Kb
Memoria de Video	2 Kb
Canales de Sonido	2 squarewave, 1 triangle, 1 noise generator y 1 canal digital de audio
Paleta de Colores	24 de una paleta de 52.
Pantalla	256 x 240 píxeles.
Número de sprites	8 por línea, 64 en pantalla.
Formato de juego	Cartucho
Puertos para mandos	2

→Master System:

Procesador	Z80 de 8 bits 3,58 Mhz.
Memoria RAM	8 Kb
Memoria de Video	16 Kb
Canales de Sonido	4
Paleta de Colores	52 de una paleta de 256
Pantalla	TFT 256 x 192 píxeles.
Número de sprites	16
Formato de juego	Cartucho
Puertos para mandos	2

→Sega Genesis:

Procesador	Motorola 68000 a 7.67 Mhz
Resolución	Resolución: 320 x 224
Memoria RAM	64 Kb
Colores	64 en pantalla de 512
Memoria de Video	64 Kb
Memoria de Sonido	8 Kb
Canales de Sonido	6 FM + 4 PSG
Formato de juego	Cartucho
Puertos para mandos	2

→SNES (Super Nintendo Entertainment System):

Procesador	Custom 65C816 16 bit
Procesador Gráfico	Modo-7 para efectos 3D
Memoria RAM	128 Kb
Memoria de Video	64 Kb
Memoria de Sonido	64 Kb
Canales de Sonido	8 PCM estéreo
Formato de juego	Cartucho
Puertos para mandos	2

→Neo-Geo:

CPU	2 CPU's 68000 de Motorola a 12 Mhz
CPU's Auxiliares	Varios CPU's paralelos, para la gestión de gráficos, realidad virtual, efectos tridimensionales y sistema de audio Sphero Symphony
Colores	Paleta de 65536 colores con un máximo de 4096 colores simultáneos en pantalla
Sprites	380 Sprites simultáneos, con un tope de 64 MegaBits para estos (8 megas)
Sonido	CPU Z80 a 4 Mhz 15 canales de sonido, 7 de ellos digitales
Memoria Cartuchos	330 MegaBits (42 megas) En 1997 expandidos a 680 MegaBits

→ Neo-Geo CD:

CPU	2 CPU's 68000 de Motorola a 12 Mhz
CPU's Auxiliares	Varios CPU's paralelos, para la gestión de gráficos, efectos de Zoom, realiad virtual, efectos tridimensionales, sonido de calidad CD stereo envolvente...
Memoria	58 Mb de DRAM 512 Kb de VRAM 68 Kb de SRAM (sonido)
Colores	Paleta de 65536 colores con un máximo de 4096 colores simultáneos en pantalla
Sprites	310 Sprites simultáneos, con un tope un poco menor que la NeoGeo de Cartucho
Sonido	CPU Z80 de Zilog a 4 Mhz 17 canales de sonido, 8 de ellos digitales
Lector CD	lector de simple velocidad

→PlayStation:

CPU	Controlado por un procesador de 33,86 MHz a 32 bits.
CPU's Auxiliares	Un chip encargado del manejo de los gráficos
Memoria	2 Mb de RAM principal
Colores	Paleta de 16.770.000. Color de 24 bits.
Imagen	1 Mb destinado a vídeo
Sonido	512 Kb de Sonido
Lector CD	Lector de doble velocidad.

→Nintendo 64:

CPU	64 bits RISC CPU (customized R4000 series) Clock Speed: 93.75 MHz
CPU's Auxiliares	RCP: SP (procesador de gráficos y sonido) y DP (procesador de dibujo de pixeles) incorporado. Clock Speed: 62.5MHz
Memoria	4.5 Mb de RAM principal
Colores	Paleta de 16.000.000. Color de 21 bits.
Imagen y Sonido	Controlada por el chip gráfico RCP: de 65,2 MHz

→Sega Saturn:

CPU	2 CPU's RISC Hitachi SH2 a 28 MHz y SH1 a 20 MHz
CPU's Auxiliares	6 CPU's paralelos, para la gestión de la pantalla, efectos de Zoom, realiad virtual, efectos tridimensionales, sonido de calidad CD tridimensional envolvente...
Memoria	2 Mb de RAM principal, Cache de 0.5 Mb y Backup de 0.5 Kbytes 1.5 Mb de VRAM 0.5 Mb de audio principal
Colores	Paleta de 16.770.000 (1.024/2.048 simultáneos) Definiciones de 320 * 224.
Imagen	Controlada por 2 custom chips de SEGA(VDP1 motor de geometría controla los polígonos y los sprites, y VDP2 motor de planos. SCROLL de 5 planos).
Sonido	Chip Sega Custom Sound Processor de Yamaha 32 canales de sonido. Adicionalmente un chip a 11.3 MHz para controlar el sonido 3D
Lector CD	Lector de doble velocidad controlado por el SH1 a 20 MHz

→Sega Dreamcast:

CPU	Hitachi SuperH SH-4 RISC CPU @ 200MHz de 128 bits
CPU's Auxiliares	Chip grafico NEC PowerVR2 y chip de sonido yamaha AICA ARM7 sonido de calidad CD tridimensional envolvente...
Memoria	16 Mb de RAM principal 8 Mb de VRAM 2 Mb de audio principal
Colores	Paleta de 16.770.000. Color de 24 bits.
Imagen	Controlada por el chip grafico NEC PowerVR2 el cual maneja poligonos y sprites manejo de 5 planos igual al Saturn.
Sonido	Chip Yamaha AICA ARM7 64 canales de sonido PCM stereo con capacidad Qsound.
Lector CD	Lector GD-ROM 12X (double density CD-ROM - 1.2 GB)
Modem	Interno de 56Kbps.

→Playstation 2:

CPU	Emotion Engine a 294,912 MHz de 128 bits
CPU's Auxiliares	Chip gráfico GS "Graphic Synthesizer" y chip de sonido SPU2 sonido de calidad CD tridimensional envolvente...
Memoria	32 Mb de RAM principal 4 Mb de VRAM 2 Mb de audio principal
Colores	Paleta de 16.770.000. Color de 24 bits.
Imagen	Controlada por el chip gráfico "Graphic Synthesizer" a 147,456 MHz el cual maneja polígonos y sprites manejo de 5 planos.
Sonido	Chip SPU2 48 canales de sonido con soporte 3D.
Lector CD	Lector DVD-ROM 4X CD-ROM 24X
Modem	Externo de 56Kbps.

→Gamecube:

CPU	IBM Gekko RISC a 485MHz de 128 bits.
CPU's Auxiliares	Procesador gráfico IBM a 162MHz encargado del manejo de poligonos y la gestión de planos.
Memoria	40 Mb de RAM principal
Colores	Paleta de 16.770.000. Color de 24 bits.
Imagen	Controlada por el chip gráfico IBM a 162 MHz
Sonido	64 canales de sonido digital

1.3 GÉNEROS DE LOS VIDEOJUEGOS:

Actualmente los Videojuegos han conseguido una diversificación tal, que es necesario catalogarlos de alguna manera en diferentes clases o géneros. Las diferencias entre cada género son los factores claves para lograr una clasificación, esta manera de diferenciarlos, por así decirlo es únicamente por cuestión de gustos, ya que cada género tiene su propia identidad y características propias de su estilo, un punto muy importante para destacar es la actividad que el jugador realiza con los distintos personajes u objetos del entorno en que se desenvuelve, dichos entornos son tan variados como en la vida real, ya que pueden ir desde una calle común y corriente hasta un complejo militar o una casa abandonada en el medio de la nada; además debe anotarse que los juegos no son solo el escenario sino también los objetos con que interactúan el(los) personaje(s) controlados por el(los) jugador(es), ya que hoy en día casi todos los juegos cuentan con la posibilidad de jugar en línea con personas en otro lugar. Con cada uno de los puntos anteriormente mencionados se puede llegar a establecer la siguiente clasificación.

→ROL.

Los juegos de ROL consisten en seguir el curso de la historia del juego, llevando a los personajes a través de esta; teniendo que superar diferentes dificultades como los combates, las diferentes misiones, las relaciones entre los diferentes personajes y los continuos peligros que se puedan presentar a lo largo del desarrollo del juego. Los personajes de los juegos de ROL, tienen diferentes habilidades: algunos pueden utilizar magia blanca, otras magias de ataque o invocaciones, o ser guerreros. Todas estas habilidades pueden ir en aumento debido a los diferentes combates, los cuales en la mayoría de los casos son por turnos.

Sagas más populares:

- Final Fantasy (2D y 3D).
- Phantasy Star (2D y 3D).
- Grandia (3D).
- Castlevania (2D y 3D).
- Sakura Wars (2D y 3D).

→LUCHA.

El objetivo de este tipo de juegos es el de entablar combates controlando a los distintos personajes, quienes cuentan con habilidades diferentes de acuerdo a su estilo. La mayoría de juegos de pelea están basados en las artes marciales orientales como el Karate o el Kung - Fu. Los puede haber de dos tipos de acuerdo a los gráficos, 2D, que son los primeros que aparecieron aunque todavía hoy en día persisten; y los 3D que aparecieron posteriormente que se pueden mover en las diferentes direcciones del escenario.

Sagas más populares:

- The King Of Fighters (2D).
- Street Fighter (2D).
- Virtua Fighter (3D).
- The Last Blade (2D).
- Soul Calibur (3D).
- Guilty Gear (2D).
- Fatal Fury (2D).

->AVENTURA.

Estos juegos tienen la particularidad de ser bastante extensos, resolviendo diferentes acertijos o puzzles y descubriendo el camino que se ha de recorrer, todo esto está muy bien pero se debe tener en cuenta que por el camino a recorrer hay diferentes obstáculos tales como monstruos que pueden ser desde Humanos, pasando por Zombies y hasta Dinosaurios.

Sagas más populares:

- Parasite Eve (3D).
- Resident Evil (3D).
- Dino Crisis (3D).
- Silent Hill (3D).
- Metal Gear Solid (3D).

->DEPORTES.

Estos juegos siempre han tenido una gran popularidad entre casi todos los aficionados a los videojuegos pues tienen para ofrecer un sin fin de diversión con la modalidad multijugador, bien sea con dispositivos propios de la consola o por Internet. Un punto clave de los juegos de deportes es el Fútbol debido a que siendo el deporte más popular del mundo atrae muchos jugadores; también está el basketball, béisbol, etc.

Sagas más populares:

- Winning Eleven.
- FIFA.
- Virtua Tennis.
- NBA 2K.
- MLB.



->AUTOS.

En estos juegos la característica principal es que ofrece mucha emoción pues el o los jugadores siempre están en constante competencia con los demás autos, todo esto no sería nada si no hubiera un gran detalle gráfico, aunque hace algunos años atrás estas no eran tan buenas han venido mejorando, así que por todo esto este género es uno de los mejores en el campo de la diversión pues la mayoría de los juegos modernos poseen muchas opciones tales como Campeonato o Pruebas de Piloto para aumentar la destreza del jugador.

Sagas más populares:

- Need For Speed (3D).
- Gran Turismo (3D).
- Formula Uno - Electronic Arts (3D).
- Formula Car (3D).
- Metropolis Street Racer (3D).
- Ridge Racer (3D).

->ACCION.

Para esta clase de entorno es muy importante la habilidad del(los) jugador(es) pues este género de Videojuegos requiere de buenos reflejos y una coordinación MANO - OJO bastante exacta para poder eliminar a los enemigos de una manera eficiente y continuar cumpliendo los objetivos de cada misión que compone el juego.

Sagas más populares:

- Devil May Cry (3D).
- Quake (3D y 3D Simulado).
- Shinobi (2D y 3D).
- Unreal (3D).
- Chaos Legion (3D).
- Contra (2D y 3D).

->ESTRATEGIA.

En este género de videojuegos la misión principal es llevada a cabo por el usuario quien se debe encargar de controlar las distintas clases de personajes, los cuales van desde Arqueros hasta Hechiceros, pasando por Enanos Guerreros, Bárbaros, etc; de una manera adecuada para que estos aprendan las diferentes técnicas de lucha y obtener o mantener la supremacía sobre el enemigo, esta clase de juegos se desarrolla en una superficie sobre la cual hay recursos naturales que permiten a los distintos pueblos o razas mantener una economía firme, por último se debe mencionar que estos videojuegos han alcanzado una gran popularidad debido a que son bastante extensos y

ofrecen la oportunidad de medir su habilidad contra otros jugadores mediante partidas multiusuario en línea además del mismo juego.

Para hablar de una manera más profunda acerca de lo que consiste la estrategia se debe tener en cuenta que esta no solo consiste en acumular recursos y ganar batallas, es también importante estar al tanto de los distintos enemigos ya sean personas o el ordenador: pues los distintos desarrolladores de este género de juegos se encargan de añadirle una Inteligencia Artificial muy avanzada a los actores (personajes) del juego para que así se torne más complejo y entretenido.

Sagas más populares:

- Age of Empires (3D).
- StarCraft (3D).
- ThemePark (3D).
- Command And Conquer (3D).
- Kessen (3D).
- War Craft (3D)

RESUMEN

En esta primera parte se plantea la importancia de los videojuegos en la sociedad actual como una herramienta de entretenimiento y avance tecnológico. El desconocimiento acerca de los temas referidos a los videojuegos causa que comúnmente se cometan errores al momento de hablar de los videojuegos subestimando la capacidad tecnológica desplegada en estos productos, las cuales no pueden ser menospreciadas ya que van a la vanguardia en el desarrollo de entornos 3D.

TALLER

- Analice todas las consolas y decida cual es la más potente.



CAPITULO 2. INTRODUCCION A LOS MOTORES GRAFICOS.

Un motor gráfico 3D es una plantilla en la cual se ejecutan todos los aspectos de un Videojuego como la iluminación o la cantidad de polígonos que se presentan en pantalla, ósea que las líneas de código dan al orden al computador para que este despliegue un cierto número de elementos en pantalla. Dicho programa sirve como base para crear juegos en apariencia y estilo son diferentes.

Los motores gráficos facilitan la manera de crear Videojuegos ya que con una sola herramienta se crean muchos juegos, lo cual permite mantener una industria que genera millones de dólares al año.

2.1 HISTORIA DE LOS MOTORES GRAFICOS.

El origen de los motores gráficos se remonta al año de 1991, cuando un joven programador llamado John Carmack creó una interfaz de diseño para el desarrollo de Videojuegos que fue conocida como motor gráfico, el juego que se creó con esta primera herramienta recibió el nombre de Wolfenstein 3D.

El motor gráfico de Wolfenstein 3D era, a la par que innovador, muy limitado. Sólo permitía ángulos de 90° en las esquinas y una altura fija de paredes. Los objetos eran representados con imágenes 2D. El gran avance vino con Doom. Salió al mercado en 1993. Ya no existía la limitación de los 90° ni las alturas. Había variados escenarios con escaleras, ascensores, ventanas. Sin embargo, tenía aún restricciones. Las paredes tenían que ser verticales y los suelos horizontales, y el jugador no podía mirar arriba o abajo.

En 1994 salió al mercado Heretic, con el motor de Doom mejorado, que permitía mirar arriba y abajo. Pero entonces llegó una nueva joya, Quake. Entre el lanzamiento de Doom y de Quake, muchos competidores crearon juegos similares a Doom, pero Quake había avanzado tanto desde Doom, como este desde Wolfenstein 3D. Ya todo era 3D, incluidos enemigos y objetos y no había limitaciones con ángulos, paredes o suelos.

Pero entonces llegó una nueva joya, Quake. Entre el lanzamiento de Doom y de Quake, muchos competidores crearon juegos similares a Doom, pero Quake había avanzado tanto desde Doom, como este desde Wolfenstein 3D. Ya todo era 3D, incluidos enemigos y objetos y no había limitaciones con ángulos, paredes o suelos.

Quake II, lanzado en 1997. Uno de los juegos que mayor comunidad ha creado, tanto de jugadores como de desarrolladores, pues el motor de Quake II fue utilizado por muchos otros juegos, todavía hoy jugados como Soldier of Fortune o Half-Life. Quake II fue el primero en implementar aceleración de gráficos por hardware. Mientras id Software preparaba su regreso con un nuevo y potente motor gráfico, nació un importante competidor de la mano de Epic Games, Unreal, en 1998.

El motor de Unreal tenía una gran robustez y diseño, a la par que unas prestaciones técnicas admirables. Después de los motores de Quake, ha sido el engine que más

licencias ha vendido para ser utilizado en otros juegos. Más tarde se lanzó una versión de Unreal con una versión mejorada del motor gráfico, llamada Unreal Tournament.

De nuevo id Software rompió moldes con su nuevo juego, Quake III, en 1999. Un juego que con unos gráficos pasmosos pero con una optimización asombrosa. Todavía hoy se utiliza para medir velocidades de nuevas tarjetas gráficas y equipos. De Quake III también se han vendido muchas licencias y se han creado grandes juegos con ellas.

Hace poco tiempo, Epic Megagames lanzó al mercado dos nuevas joyas en cuanto a motores gráficos: Unreal Tournament 2003 y Unreal 2: The Awakening. Pronto saldrá al mercado la nueva obra maestra de Carmack, Doom III, que utilizará todo el potencial de las tarjetas gráficas de última generación para crear iluminación fotorealista y dar la sensación definitiva de terror.

2.2 CATEGORIZACION DE LOS MOTORES GRAFICOS:

Los motores gráficos tienen diferentes categorías dependiendo de su licencia y uso. Los puede haber privados, comerciales y gratuitos.

- ✓ Motores Privados: "son aquellos usados por una empresa en particular, ósea ellos mismos lo desarrollan y lo utilizan para sus juegos exclusivamente. Este tipo de motores no están a la venta comercial y preferiblemente las empresas que adoptan esta política son las japonesas. Ej. Capcom, Square-Enix, Konami, Sega-Sammy, SNKplaymore, etc".
- ✓ Motores Comerciales: "este tipo es usado por las empresas norteamericanas y europeas son motores desarrollados por empresas de juegos pero después colocan a la venta tal es el caso del motor QuakeIII, Unreal, Doom3, etc. El precio de estos motores oscila entre los 200.000 y los 500.000 dólares".
- ✓ Motores Gratuitos: "son aquellos que se rigen por las licencias GNU o GPL. Dentro de esta categoría hay motores con diferentes grados de desarrollo y para plataforma diferentes algunos ejemplos de motores gratuitos son: Genesis3D, Fly3D, Cristal Space, Torque (OV12), Irrlicht.

En la actualidad hay 400 unos motores en desarrollo o que se están usando para desarrollar juegos en la actualidad.

2.3 ATRIBUTOS DE UN MOTOR GRAFICO.

Los motores gráficos manejan diferentes aspectos o atributos que son fundamentales en la ejecución de los Videojuegos que se desarrollan con estas herramientas. los distintos aspectos que se deben tener en cuenta no pueden mirarse independientemente aislandolos, sino interrelacionándolos de tal manera que se vean en un solo conjunto.

2.3.1 API.

Una API (Application Programming Interface) es un atributo que permite a los desarrolladores trabajar de una forma más amigable con el computador. Este atributo contiene herramientas para agregar sonido, polígonos o plantillas para crear mundos.

2.3.2 ILLUMINACION.

La iluminación es un aspecto fundamental para los Motores Gráficos, pues esta da una sensación de realidad a los Videojuegos, así como los efectos lumínicos dan una sensación real en los escenarios, también si es manejada de manera inapropiada se genera un producto desagradable a la vista de los distintos usuarios, por esto es importante tener especial cuidado en la manipulación de este elemento, para esto se utilizan dos categorías de Iluminación:

→ Iluminación Difusa: es aquella luz que al momento de alcanzar un objeto se dispersa por igual en todas las direcciones, de forma que la luz reflejada no depende en absoluto de la posición del espectador. Por ejemplo, cuando el sol baña un terreno de juego, la luz llega a todas partes.

→ Iluminación Especular: se diferencia porque depende de la posición del espectador, la dirección de la luz y la orientación del triángulo en el que se refleja. Por ejemplo, el haz de luz de una linterna rebotará de forma distinta cuando se refleje en una moneda y en una brizna de hierba. La iluminación especular reproduce las propiedades reflectantes de un objeto y permite crear efectos como reflejos y resplandores.

Otra parte muy importante de los efectos de Iluminación es su contraste, el Sombreado de los objetos.

2.3.3 POLIGONOS.

Como es bien sabido un polígono es una superficie plana que consta de muchos lados, entre los cuales se encuentran el triángulo, el cual es tal vez el tipo de figura geométrica más usada por los Motores Gráficos 3D; la forma en la que son usados los polígonos por los Motores es muy sencilla ya que las figuras más complejas son compuestas por otras más sencillas, por ejemplo: para construir un cubo se necesitan 6 polígonos, uno por cada cara del cubo.

2.3.4 NEBLA.

La Niebla es un aspecto muy importante para la creación de Videojuegos pues esta se utiliza para originar escenarios de impacto a la mirada de los usuarios, los efectos que se crean mediante el uso de Niebla son muy delicados ya que esta tiene muchos aspectos diferentes como la densidad, el color y la profundidad (espesor); esta parte de la profundidad es de sumo cuidado pues los objetos deben verse de acuerdo a la cantidad de niebla que haya.

2.3.5 MIPMAPPING.

Es una técnica que genera texturas en versiones pequeñas. Por ejemplo, cuando un elemento tiene varios colores y este se aleja del plano principal, los colores podrían convertirse en una mancha o en color centelleante no deseado. Por medio del mipmapping es posible crear píxeles (puntos que forman las imágenes en los monitores) texturizados, conocidos como texelsttexture pixels, píxeles texturizados), que se ven mejor en la distancia (esto se explica más a profundidad en el apartado 2.4.5).

2.3.6 TEXTURAS.

Definen la forma en que lucen los polígonos. Las texturas se usan para dar un toque más real a los escenarios. Por ejemplo, se puede modelar una pared de ladrillo por medio de un polígono plano al cual se le agrega una textura que simule las hendiduras e 'imperfecciones' de una pared de verdad.

2.3.7 COLOR.

Los efectos de iluminación, niebla, mipmapping y demás forman el color de un escenario, ya que todos los píxeles que conforman dichos efectos deben llevar una cuota de color. El motor 3D se encarga además de calcular el número de colores soportados por el monitor del usuario.

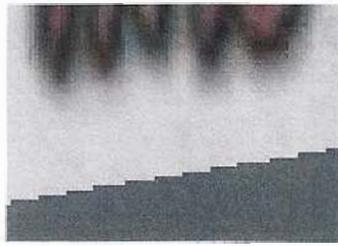
2.3.8 SONIDO.

El sonido también hace parte hace del motor 3D. Para crear buenos efectos sonoros, una de las variables que se deben tener en cuenta es la ubicación del sonido. Por ejemplo, si el personaje se aleja caminando, es de suponer que el sonido se pierda con la distancia. Lo mismo para un personaje u objeto que se acerca. Si un objeto se ubica entre el escucha (usted) y el origen del sonido, el motor 3D es el encargado de variar el sonido, tal y como sucedería en la vida real si alguien se parara frente al televisor.

2.4 EFECTOS GRAFICOS DE LOS MOTORES GRAFICOS.

2.4.1 ANTIALIASING.

El Antialiasing sirve para evitar el efecto "escalera" en los polígonos, es decir los bordes dentados o dientes de sierra (jaggie, para quienes gusten de los anglosajonismos). Aquí podemos ver el borde de un polígono con bordes dentados, sin antialiasing...



Todos hemos visto juegos con bordes así. Luego llega el antialiasing y... ¿lo hace desaparecer? No. Esto es importante. El Antialiasing no elimina los bordes dentados, los disimula. Lo podemos ver a continuación...



Lo que hace, como vemos, es mezclar el color de los píxeles más cercanos al borde del polígono con él. El Antialiasing por regla general es un efecto que consume recursos considerables, por lo que se suele incorporar en el hardware (N64, DC y GC) o se permite programarlo uno mismo con una epi lo suficientemente potente (PS2). Y aún así es una pesada carga. Por ello hay distintos tipos de Antialiasing, y se puede aplicar con mayor o menor fuerza, según se requiera. De hecho hay juegos que cuando detectan una sobrecarga de recursos disminuyen proporcionalmente (o eliminan) la fuerza de antialiasing a usar.

Con PS2 nació (por razón desconocida) el falso mito de que no tiene antialiasing, y de que sólo sus juegos tienen bordes dentados. Falso. Todos los juegos tienen los bordes dentados en mayor o menor medida por la razón comentada. Ocurrió que los primeros juegos no incorporaban antialiasing y dichos bordes eran muy notorios (como en la primera imagen). PS2 no tiene antialiasing... incorporado. Pero permite que cualquier programador lo haga por sí mismo. Por lo que sí hay juegos (y ahora es la norma) de PS2 con el antialiasing aplicado. Aunque se ha querido vender como un terrible defecto gráfico desde que salió PS2, no lo es. En Dreamcast y N64 también habían juegos sin antialiasing, aunque entonces no se habló de ello, porque no era un aspecto que llamase especialmente la atención. En determinados tipos de juegos ni siquiera es perceptible su ausencia (o no supone gran diferencia aplicarlo) y simplemente no se pone. En GC también saldrán juegos sin antialiasing (o con uno muy leve) por la misma razón.

2.4.2 BUMP MAPPING.

Este sí es un efecto de nueva generación que supone un importante añadido a la calidad gráfica. Lo primero, veamos la imagen original...



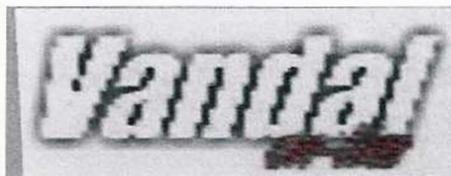


Bien, el bump mapping sirve básicamente para simular relieve en una textura. La modifíca creando un sombreado alrededor de los píxeles con mayor contraste simulando una sombra (en la imagen de abajo, alrededor de las letras), que proporciona altura, o lo contrario, profundidad. Naturalmente, nunca será tan realista que un relieve real poligonal (que supondría una bárbara cantidad de polígonos para un sólo objeto), pero que da mucho mejor que si no lo tuviera. Aún así, también permite (aunque pida más potencia, y dependiendo del hardware) que la posición de la sombra del supuesto relieve, varíe dependiendo de dónde venga la luz. Aquí tenemos la imagen anterior, tras haber aplicado el bump mapping:



2.4.3 COMPRESIÓN DE TEXTURAS.

El nombre lo dice. Se trata de comprimir las texturas, para que ocupe n menos espacio o para hacerlas más grandes ocupando el mismo espacio. Dreamcast empezó usando un sistema de compresión de texturas por software que las comprimía en una razón de 3:1. Aquí tenemos una textura de 64x64 píxeles...



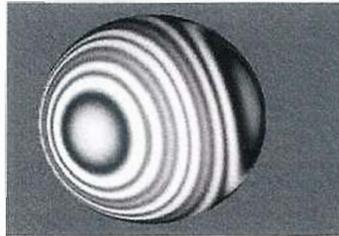
La compresión de texturas podríamos usarla en este caso para introducir una de mayor tamaño y calidad ocupando el mismo espacio como ésta...



...cuyo tamaño es de 256x256 píxeles, y que sin embargo ocuparía lo que la anterior. Uno de los estándares de compresión de hoy en día (el de S3, soportado por las Direct X y GC) usan una compresión potente, pero con pérdida de calidad (como el jpg), que sin embargo es casi siempre imperceptible.

2.1. ENVIRONMENT MAPPING (MAPEADO DE ENTORNO).

El environment mapping consiste en una imagen que es proyectada, que se refleja, en ciertos objetos. El ejemplo más común es el de los juegos de coches. En los cristales siempre vemos árboles pasando, o montañas, una imagen cualquiera con montañas u árboles dibujados que se refleja en el cristal (o en la carrocería o en donde sea) para simular que refleja el entorno. Y he aquí un ejemplo, esta es la imagen sin environment mapping



Ahora se coge la imagen a reflejar, que en este caso es la del ejemplo del bump mapping antes de ser aplicado, y se pone como environment mapping...



Y ahí está la esfera reflejando dicha imagen. Se suele hacer esto, porque reflejar el entorno real supone una cantidad brutal de datos a procesar.

2.4.5 MIP MAPPING.

Este es un efecto muy básico y que conocemos todos, y que se confunde a veces con el antialiasing. El Mip Mapping es el encargado de difuminar las texturas para que no pixelen, suavizándola más o menos según el tipo. Aquí tendríamos un polígono sin mip mapping...



La calidad de este filtrado también depende de la paleta de color que se use, cuanto mayor sea mejor. También es la teoría culpable de que a veces las texturas se vean borrosas. Cuando una textura es pequeña, sin Mip Mapping será un conjunto de enormes píxeles. Con mip mapping serían un enorme borron. En cualquier caso siempre queda mejor, como podemos ver a continuación...



RESUMEN

La importancia de conocer con que tipo de herramienta se piensa desarrollar el manual se justifica en facilitar el rápido entendimiento del manual y familiarizar al lector con conceptos fundamentales que le serán útiles en la comprensión de la tecnología empleada para el desarrollo de los videojuegos. El reconocimiento de los distintos tipos de motores según su licencia dejara justificada la elección del motor en el que se basara el manual.

TALLER

Selecciona una imagen y aplícale los efectos antes mencionados con tu Editor de Imágenes favorito.

CAPITULO 3. REQUERIMIENTOS DE HARDWARE Y MOTORES GRAFICOS UTILIZADOS.

El principal motivo de este capítulo es saber o tratar de entender el desarrollo y la forma como interactúan los motores gráficos con cada uno de los distintos hardware de nuestro PC y como afecta estos mismos en la ejecución y rendimientos de los Videojuegos.

Se hará énfasis en el hardware, que de alguna manera se puede decir que son los elementos que interactúan con los distintos componentes del Motor Gráfico como son los procesadores, las tarjetas aceleradoras gráficas, tarjeta de sonido, entre otros.

3.1 DISTINTAS CLASIS DE HARDWARE LUDICO.

El hardware es un concepto que nació con la informática al igual al software. Los Videojuegos son un campo en el cual el hardware se exige al máximo obligando a mantener una tecnología de punta; continuamente aparecen nuevos modelos de Procesadores Gráficos para PC dando a entender el continuo avance que se tiene en el desarrollo de hardware destinado para los Videojuegos.

El hardware lúdico que es usado para los Videojuegos tiene distintas variaciones que son destinados específicamente para su ejecución. Entre las Variaciones se encuentran las máquinas Arcade y las consolas, en cambio las tarjetas gráficas son accesorios que se añaden al PC para mejorar el rendimiento gráfico de este.

Los sistemas Arcade a lo largo de la historia de los Videojuegos siempre han llevado la vanguardia en cuanto a avances técnicos se refiere. El Arcade por lo general es una máquina disponible en salones de juegos, los cuales para poder funcionar necesitan la introducción de monedas o fichas.

Los primeros Arcades aparecieron en la década de los 70s. Estas primeras máquinas contaban con sistemas gráficos muy avanzados para su época y varios de los primeros juegos fueron Space Invaders, Pong, Pac-Man y muchos otros que ahora en la actualidad son considerados clásicos. En los años 80s compañías como CAPCOM, SEGA, KONAMI y NAMCO, avanzaron en muchos aspectos tanto de hardware como en el plano gráfico y sonoro. Algunos juegos de esta época son: Street Fighter, Hang on, Contra.

Al llegar los 90s aparece una compañía que lograría elevar la capacidad de los Arcades hasta un punto muy alto. SNK y su sistema MVS (Multi Video System) tenían un hardware para gráficos 2D nunca antes visto haciendo que perdurara en el tiempo hasta nuestros días. SNK desarrollo el sistema MVS con el propósito de afianzarse en este mercado. A diferencia de esta SEGA desarrollo sus sistemas Arcade enfocándose en las gráficas 3D hasta lograra el desarrollo de la placa NAOMI.

La placa MVS cuenta con las siguientes características:

CPU	2 CPU's 68000 de Motorola a 12 Mhz
CPU's Auxiliares	Varios CPU's paralelos, para la gestión de gráficos, realidad virtual, efectos tridimensionales y sistema de audio Sphero Symphony
Colores	Paleta de 65536 colores con un máximo de 4096 colores simultáneos en pantalla
Sprites	380 Sprites simultáneos, con un tope de 64 MegaBits para estos (8 megas)
Sonido	CPU Z80 a 4 MHz 15 canales de sonido, 7 de ellos digitales
Memoria Cartuchos	330 MegaBits (42 megas) En 1997 expandidos a 680 MegaBits

En 1998 Sega Presento su consola Dreamcast, la cual dispone de una arquitectura avanzada para el desarrollo de Videojuegos la cual sirvió de base para la creación de la placa Arcade NAOMI. Esta placa aparte de poseer un potencial 3D también fue la primera placa con la capacidad 2D en superar al sistema MVS de SNK.

Para el tema de las consolas es muy importante destacarlas como los dispositivos de Juego más popularizados ya que son muy versátiles a la hora de agregar dispositivos como el caso de la cantidad de mandos que puede variar fácilmente con dispositivos como en el caso particular del PlayStation y el PlayStation2 que utilizan el Multitap, que no es mas que un dispositivo que proporciona cuatro salidas idénticas a las de las consolas, que entre otras cosas son las mismas para el PS y el PS2, dicho aparato se conecta a una de las bahías de la consola y de allí se conectan los demás controles.

3.2 PROCESADORES.

Un microprocesador es un circuito integrado construido en un pedazo diminuto de silicón. Contiene miles, o incluso millones, de transistores que se interconectan via los rastros extrafinos de aluminio. La función de los transistores es guardar y manipular datos juntos para que el microprocesador pueda realizar una variedad ancha de funciones útiles.

La unidad de procesamiento central (CPU) es donde se manipulan los datos. En una microcomputadora, el CPU completo está contenido en un chip muy pequeño llamado microprocesador. Todas las CPU tienen por lo menos dos partes básicas, la unidad de control y la unidad aritmética lógica. Todos los recursos de la computadora son administrados desde la unidad de control, cuya función es coordinar todas las actividades de la computadora. La unidad de control contiene las instrucciones de la CPU para llevar a cabo comandos. El conjunto de instrucciones, que está incluido dentro de los circuitos de la unidad de control, es una lista de todas las operaciones que realiza la CPU. Cada instrucción en el conjunto de instrucciones es acompañada por un micro código, que son instrucciones muy básicas que le dicen

a la CPU como ejecutar las instrucciones. Cuando la computadora corre un programa, busca los comandos del programa dentro del conjunto de instrucciones de la CPU y las ejecuta en orden. La unidad aritmético - lógico (ALU) es cuando la unidad de control encuentra una instrucción que involucre aritmética o lógica, le pasa el control al segundo componente de la CPU. La ALU incluye un grupo de registros, es decir, memoria construida directamente en la CPU que se usa para guardar datos que están siendo procesados por la instrucción actual.

Unidad Central de Procesamiento (CPU)

La CPU de una computadora contiene la inteligencia de la máquina; es donde se realizan los cálculos y las decisiones. El complejo procedimiento que transforma datos nuevos de entrada en información útil de salida se llama procesamiento. Para llevar a cabo esta transformación, la computadora usa dos componentes: el procesador y la memoria. El procesador es el cerebro de la computadora, la parte que interpreta y ejecuta las instrucciones. El procesador casi siempre se compone de varios circuitos integrados o chips, estos están insertados en tarjetas de circuitos, módulos rígidos rectangulares con circuitos que los unen a otros chips y a otras tarjetas de circuitos.

El microprocesador moderno contiene unos 20 millones de transistores y cada chip terminado es el producto de procesos más complicados que los que se utilizaron en el Proyecto Manhattan para construir la bomba atómica. Y no obstante, pese a un proceso de manufactura extraordinariamente refinado, los microchips se producen en volumen a razón de más de 1,000 millones de unidades por año. Para poner esta complejidad en perspectiva, imagínese que dentro de cada microprocesador diminuto existe una estructura tan compleja como una ciudad de tamaño mediano, incluidas todas sus líneas de energía eléctrica, líneas telefónicas, líneas de drenaje, edificios, calles y casas. Ahora imagine que en esa misma ciudad, millones de personas se desplazan a la velocidad de la luz y con la sincronización perfecta en una danza de coreografía muy complicada.

Y eso es tan sólo un chip. De todas las estadísticas asombrosas que se utilizan para describir el mundo del microprocesador, ninguna es más extraordinaria que ésta: el número total de transistores que integran todos los microchips que se producirán en el mundo este año es equivalente al número de gotas de lluvia que caerán en California durante ese mismo periodo.

3.3 PROCESADORES GRAFICOS.

La tarjeta gráfica es la que se encarga de procesar la información que el procesador le envía a esta y, a su vez de enviarla al monitor. Las tarjetas generalmente tienen algún tipo de aceleración (procesador) del video. Los aceleradores gráficos se encargan de despejar casi todo el procesamiento de los gráficos de la CPU (Unidad de Procesamiento Central). Estos hacen que opere más rápidamente el computador al descargar trabajo del procesador.

Las tarjetas gráficas y los aceleradores traen normalmente memoria RAM para guardar la información de las imágenes que se muestran el monitor. Una mayor

cantidad de esta memoria, que va sobre la tarjeta, permite mayores resoluciones y mayores cantidades de colores en la pantalla. Hay tarjetas gráficas que tienen memorias más veloces que otras haciéndolas a su vez más rápidas, y unas que trabajan en 32, 64 y 128 bits. La memoria mínima para una tarjeta de 64 bits es de 2 MB, y para una de 128 bits es de 4 MB. Si se utiliza menos memoria, se pierde rendimiento. El estándar de hoy en día es de 2Mb de memoria para una tarjeta gráfica común y corriente y hasta de 24Mb para los aceleradores de 3D más especializados. Hoy en día las Tarjetas Gráficas son para ranuras PCI y AGP.

PCI (Peripheral Components Interface)

El Bus PCI es el encargado de administrar todas nuestras tarjetas PCI y de comunicarnos con el procesador central del PC. Cuando es necesario que el procesador se comunique con alguna de ellas, el CHIPSET PCI se encarga de buscar la tarjeta y transferirle la información desde la memoria del PC o recoger los datos generados en la tarjeta y colocarlos en la RAM de forma que el procesador sólo tenga que trabajar con ellos.

Para que se entienda mejor podríamos decir que el BUS PCI es la persona encargada de un almacén dentro del cual existen varias máquinas que colocan las cajas (las tarjetas del PC). Cada vez que el jefe del almacén (el procesador) necesita algo de éste se lo pide al encargado (bus PCI) y éste realiza el control de todas las máquinas (tarjetas) colocando en la puerta del almacén lo que el procesador ha pedido o cogiendo de la puerta lo que ha dejado para dárselo a la máquina a la que vaya destinado.

Gracias a este plan de trabajo del bus PCI el procesador puede trabajar en otras tareas más complejas y desentenderse de las tarjetas del PC. Dichas tareas pueden ser manipulación de texturas, inteligencia artificial o cálculo de polígonos de escenas 3D.

El bus PCI es un bus de comunicaciones de 32 bit que trabaja a 33MHz ofreciendo una tasa de transferencia tope teórica hacia y desde la memoria RAM del PC de 133 Mbits/s ayudada con la posibilidad de escribir en modo ráfaga. Esta velocidad de transferencia es más que suficiente para cualquier tarjeta PCI incluyendo tarjetas gráficas 2D como la Matrox G-100 o 2D/3D de bajo rendimiento como la Matrox Mystique, la Voodoo Graphics o la S3 Virge. El problema surge cuando tenemos una aceleradora 3D de alto rendimiento cuyo potencial de manipulación de polígonos y texturas es tan grande que atasea el bus PCI lo que conlleva lentitud en el funcionamiento de todo el bus.

AGP (Advanced Graphics Port)

El bus AGP fue creado tras la aparición del Pentium II con el fin de proporcionar a las tarjetas 3D un canal de comunicación con la memoria del PC que superase los 133 Mbits/s del bus PCI que resultaban insuficientes. El AGP es un bus exclusivo para la tarjeta gráfica funcionando como mínimo el doble de rápido que el PCI. Desde su aparición, el bus AGP ha sufrido varias revisiones. La velocidad se ha mantenido siempre en 66Mhz pero las tasas de transferencia con la RAM han ido ascendiendo desde los 266Mbits/s del primer AGP hasta los 4 Gbit/s del modo 4x.



Cualquier tarjeta gráfica AGP es capaz de usar la memoria RAM del PC emulando allí la memoria de vídeo que le falta para manipular texturas de gran tamaño. Por ejemplo, si tenemos una tarjeta Matrox de 4MB AGP y tenemos que manipular una textura de 6MB lo que haría la tarjeta sería usar 6MB de la RAM y manipularla allí. A este proceso se le llama *Direct Memory Execution* y la verdad, es un engaño. Es un engaño porque cuando la tarjeta AGP usa la RAM del PC no deja acceder a ella al procesador con lo que éste se queda sin hacer nada hasta que la tarjeta le de paso. Además, por muy rápida que sea la memoria del PC siempre será más lenta que la memoria de vídeo con lo que la manipulación de las texturas será, por tanto, más lenta aunque la transferencia de datos entre tarjeta y RAM esté optimizada con AGP 2X o 4X.

Otra de las ventajas de AGP es que posee 8 canales adicionales de comunicación con la RAM, es decir, PCI es de 32 bit con lo que posee 32 canales de comunicación. AGP también es de 32 bit pero posee 8 canales extra para usar cuando sea necesario, es decir AGP puede llegar a ser en momentos determinados de 40bits.

Manipular texturas en la memoria RAM parece una buena solución para tratamiento de texturas grandes pero lo ideal es tener una aceleradora de por lo menos 32MB y 128 o 256 bit que no necesite de los inventos de Intel y su alojamiento de texturas en RAM.

AGP vs. PCI

Como ya todos sabemos y se muestra día a día en las distintas publicaciones informáticas el PCI es un bus de alto rendimiento y muy eficaz pero que se queda corto con el tratamiento de texturas. Para ello se ha creado AGP que posee más velocidad de transferencia y capacidad de usar la RAM como memoria de vídeo de la tarjeta pero con la desventaja de ser esta capacidad una solución lenta hoy por hoy. Por consiguiente queda claro que una tarjeta aceleradora 3D PCI de 16 MB irá más rápido que otra AGP de 4MB ya que la PCI trabaja en su propia memoria y la AGP depende de la memoria del PC.

PCI Express

Este bus ofrece un rendimiento de diseño sin precedentes que utiliza una avanzada arquitectura de interconexión serie de alta velocidad a la vez que permite menos hardware de E/S debido a la reducción de los cables físicos necesarios para conectar la E/S al complejo de procesamiento del sistema. La introducción de la tecnología PCI Express también representa la culminación de un esfuerzo que se ha realizado en todo el sector a través de varios años.

No obstante, el bus PCI ha permanecido constante con solamente ligeras mejoras en el rendimiento. Por lo que se necesitaba una revolución, algo que desequilibrara crecientemente entre el rendimiento del procesamiento y la E/S. Con una nueva perspectiva de la conectividad de E/S, PCI Express aprovecha una década de innovaciones tecnológicas. Este cambio viene acompañado de una oportunidad para los administradores de TI de brindar rendimiento y fiabilidad a través de toda la empresa.

Mayor rendimiento a través de las empresas: En los sistemas cliente, la transición a Gigabit Ethernet consume el bus PCI de hoy y deja poco rendimiento a los demás adaptadores del sistema. Al adoptar la tecnología PCI Express, cada conexión de E/S puede ofrecer una banda ancha dedicada de 2,5 gigabits por segundo (Gbps) o un rendimiento de bus triplicado dedicado para cada ranura, lo cual brinda un amplio margen de ampliación en el rendimiento para Gigabit Ethernet y para futuros requisitos de conectividad de clientes.

Las plataformas de estación de trabajo y desktop de hoy utilizan el bus 8x AGP (Puerto gráfico acelerado) especializado para la conexión de tarjetas gráficas. El sector gráfico está migrando de forma acelerada a la conexión PCI Express x16, lo cual duplica el rendimiento de las conexiones gráficas a 40 Gbps. Esta transición aumenta las capacidades gráficas que no podían realizarse anteriormente debido a las limitaciones de la interconexión AGP. Los principales proveedores gráficos han informado sus planes de lanzar al mercado productos en 2004. ATI Technologies y NVIDIA Corp. han anunciado una línea completa de productos oportunos que estarán alineados con los recientes lanzamientos de las plataformas de chipset Intel 925x/915 Express e Intel® E7525.

Las interconexiones de almacenamiento, tales como SCSI (Interfaz de sistemas para equipos pequeños) y el canal de fibra, ofrecen mejoras en el rendimiento con mayores velocidades de conexión. Con la introducción de SCSI conectado en serie y el canal de fibra de varios gígs que se lanzarán en un futuro cercano, la oportunidad de avances en el rendimiento del almacenamiento está a la vuelta de la esquina.

Voodoo Bañes

Es el primer chip acelerador 2D/3D video en Windows de 3Dfx. Voodoo Banshee combina un nuevo núcleo 2D de 128 bits con una VGA de 128 bits diseñada por 3Dfx, circuitería de video y un núcleo 3D de uno de los productos mejores para juegos; Voodoo2. Voodoo Banshee está específicamente diseñado para aplicaciones OLM y está enfocado a ofrecer prestaciones 2D y 3D en un único procesador gráfico integrado. La clave de Voodoo Banshee es el gran conjunto de juegos 3D compatibles, que se esperan que alcancen los 400 en las Navidades de 1998.

Este soportará resoluciones 1900x1440 en 2D y 3D. Actualmente los juegos 3D son renderizables a resoluciones en el caso de algunos hasta 1600x1200. Cuando los drivers evolucionen, Voodoo Banshee soportará muchos más títulos a esta ultra-alta resolución.

También soportará configuraciones de memoria de 4-16MB de SGRAM. También Voodoo Banshee soporta SD-RAM de bajo coste. Con su interfaz de memoria de 128 bits, la única configuración soportada será la de 16 Mb. de SDRAM desde que está disponible en un sólo módulo. Esto hace que sea compatible tanto para PCI 66 MHz como AGP 1x con soporte de banda lateral.

3Dfx va por encima de las actuales tarjetas 2D diseñando un núcleo completamente nuevo, que está mucho más integrado con Windows y que incluye funciones tales como un completo soporte de 256 ROPS, polígonos por hardware y un soporte verdadero de bloqueo de escritura de la SGRAM. Con resoluciones incrementadas

requieren la generación y el movimiento de mayores bloques de memoria, una arquitectura 2D más rápida permitiría unas prestaciones mucho mejores.

Medición de VooDoo Banshee con Juegos

Los benchmarks artificiales que existen hoy no son suficientemente precisos como para indicar las prestaciones de un complejo sistema 3D como Incoming o Quake. Los clientes están cansados de puntuaciones en los bancos de pruebas que no corresponden con las prestaciones reales ofrecidas por una tarjeta 3D. Más que generar bancos de pruebas artificiales, ¿no hace más sensible tener el informe del mundo real del juego basado en los actuales datos de los juegos?

El GameGauge no es sólo un banco de pruebas 3D. Está basado en las prestaciones de los juegos en el mundo real que conocemos y queremos. La puntuación de los juegos no son ratios de frames artificiales, es lo que experimentas cuando juegas. Cada juego ofrece su individual "media de frames por segundo" para la puntuación del conjunto de tarjetas 3D, formando lo que se llama el GameGauge. Se han probado los siguientes juegos:

- Forsaken
- Turok
- F22 ADF
- Incoming
- GL Quake
- Quake2.

Configuración de las prestaciones

Las cuestiones más importantes que gobiernan las prestaciones gráficas son los modos de pantalla, sincronización vertical, configuración óptima de las tarjetas, y tener los últimos drivers.

La frecuencia de refresco puede tener un gran impacto en las prestaciones especialmente para tarjetas que no pueden desconectar la sincronización vertical (ver abajo). Funcionar a diferentes frecuencias de refresco en diferentes tarjetas puede fácilmente falsear los resultados. 75 Hz es la frecuencia recomendada para funcionar con el GameGauge.

3.4 TARJETA DE SONIDO.

Conociendo que todos los Videojuegos actuales tienen una gran variedad de sonidos debe saberse que el reconocimiento de las distintas capacidades de este dispositivo es importante para entender como afectan el desarrollo de un juego, pues este se encarga de interpretar los distintos sonidos producidos durante la visualización, manejo y desarrollo de los diferentes escenarios producidos en un videojuego.

Funcionamiento

Las dos funciones principales de estas tarjetas son la generación o reproducción de sonido y la entrada o grabación del mismo. Para reproducir sonidos, las tarjetas incluyen un chip sintetizador que genera ondas musicales. Este sintetizador solía emplear la tecnología FM, que emula el sonido de instrumentos reales mediante pura programación; sin embargo, una técnica relativamente reciente ha eclipsado a la síntesis FM, y es la síntesis por tabla de ondas (Wave Table).

En Wave Table se usan grabaciones de instrumentos reales, produciéndose un gran salto en calidad de la reproducción, ya que se pasa de simular artificialmente un sonido a emitir uno real. Las tarjetas que usan esta técnica suelen incluir una memoria ROM donde almacenan dichos "samples": normalmente se incluyen zócalos SIMM para añadir memoria a la tarjeta, de modo que se nos permita incorporar más instrumentos a la misma.

Una buena tarjeta de sonido, además de incluir la tecnología Wave Table, debe permitir que se añada la mayor cantidad posible de memoria. Algunos modelos admiten hasta 28 Megas de RAM (cuanta más, mejor).

Efectos

Una tarjeta de sonido también es capaz de manipular las formas de onda definidas: para ello emplea un chip DSP (Digital Signal Processor, Procesador Digital de Señales), que le permite obtener efectos de eco, reverberación, coros, etc. Las más avanzadas incluyen funciones ASP (Advanced Signal Processor, Procesador de Señal Avanzado), que amplía considerablemente la complejidad de los efectos. Por lo que a mayor variedad de efectos, más posibilidades ofrecerá la tarjeta.

Polifonía

Es el número de instrumentos o sonidos que la tarjeta es capaz de emitir al mismo tiempo. Las más sencillas suelen disponer de 20 voces, normalmente proporcionadas por el sintetizador FM, pero hoy en día no debemos conformarnos con menos de 32 voces. Las tarjetas más avanzadas logran incluso 64 voces mediante sofisticados procesadores, convirtiéndolas en el llamado segmento de la gama alta.

MIDI

La práctica totalidad de tarjetas de sonido del mercado incluyen puerto MIDI: se trata de un estándar creado por varios fabricantes, que permite la conexión de cualquier instrumento, que cumpla con esta norma, al ordenador, e intercambiar sonido y datos entre ellos. Así, es posible controlar un instrumento desde el PC, enviándole las diferentes notas que debe tocar, y viceversa; para ello se utilizan los llamados secuenciadores MIDI.

Si bien es cierto en el mismo puerto MIDI se puede conectar un Joystick, algo muy de agradecer por el usuario, puesto que normalmente los conocidos equipos Pentium no incorporan de fábrica dicho conector, algo habitual, por otra parte, en sus inmediatos antecesores, los ordenadores 486.

Frecuencia de muestreo

Otra de las funciones básicas de una tarjeta de sonido es la digitalización: para que el ordenador pueda tratar el sonido, debe convertirlo de su estado original (analógico) al formato que él entiende, binario (digital). En este proceso se realiza lo que se denomina muestreo, que es recoger la información y cuantificarla, es decir, medir la altura o amplitud de la onda. El proceso se realiza a una velocidad fija, llamada frecuencia de muestreo; cuanto mayor sea esta, más calidad tendrá el sonido, porque más continua será la adquisición del mismo.

Esta es la encargada de marcar la calidad de la grabación: por tanto, es preciso saber que la frecuencia mínima recomendable es de 44,1 KHz, con la que podemos obtener una calidad comparable a la de un disco compacto.

La compatibilidad

Indudablemente, en estos momentos, el mercado de las tarjetas de sonido tiene un nombre propio: Sound Blaster. En la actualidad, cualquier tarjeta que se precie debe mantener una total compatibilidad con el estándar impuesto por la compañía Creative Labs: existen otros, como el pionero Adlib o el Windows Sound System de Microsoft. Pero todos los juegos y programas que utilizan sonido exigen el uso de una tarjeta compatible Sound Blaster, así que sobre este tema no hay mucho más que comentar.

Otro asunto es la forma de ofrecer dicha compatibilidad: por software o por hardware. La compatibilidad vía software puede tener algunas limitaciones: principalmente, puede ser fuente de problemas con programas que accedan a bajo nivel o de forma especial a las funciones de la tarjeta. Asimismo, los controladores de emulación deben estar bien diseñados, optimizados y comprobados, para no caer en incompatibilidades, justo lo contrario de lo que se desea alcanzar. Por tanto, es preferible la emulación por hardware.

Sonido 3D

El sonido 3D consiste en añadir un efecto dimensional a las ondas generadas por la tarjeta: estas técnicas permiten ampliar el campo estéreo, y aportan una mayor profundidad al sonido habitual. Normalmente, estos efectos se consiguen realizando mezclas específicas para los canales derecho e izquierdo, para simular sensaciones de hueco y direccionalidad.

Seguro que os suenan nombres como SRS (Surround Sound), Dolby Prologic o Q-Sound: estas técnicas son capaces de ubicar fuentes de sonido en el espacio, y desplazarlas alrededor del asombrado usuario. Y decimos asombrado, porque el efecto conseguido es realmente fantástico, y aporta nuevas e insospechadas posibilidades al software multimedia y, en especial, a los juegos.

3.5 CARACTERÍSTICAS MOTOR GENESIS 3D

Para que el proyecto de desarrollo de VIDEOJUEGOS sea prolijo debe contarse con un motor gráfico que sea potente en el aspecto de manejo de los distintos efectos y herramientas alternas para la creación de Ambientes Gráficos 3D.

- El Motor Gráfico GENESIS 3D fue desarrollado por Eclipse Entertainment.
- La plataforma que lo soporta es Windows9x y NT. Bajo licencia libre y derecho a modificar el código fuente.
- Las APIs que soporta son Direct3d y Glide. Renderizado por Portales, árboles BSP, radiosidad y LOD. Incorpora luces multicolores y dinámicas.
- El Genesis trabaja nativamente con el formato de ficheros de animación de 3Dstudio MAX. Texture mapping, textura translúcida, morphing de texturas, texturas animadas y procedurales.
- Trabaja con Sprites 3D, sombras dinámicas, jerarquía de objetos, detección de colisiones.

Características especificadas:

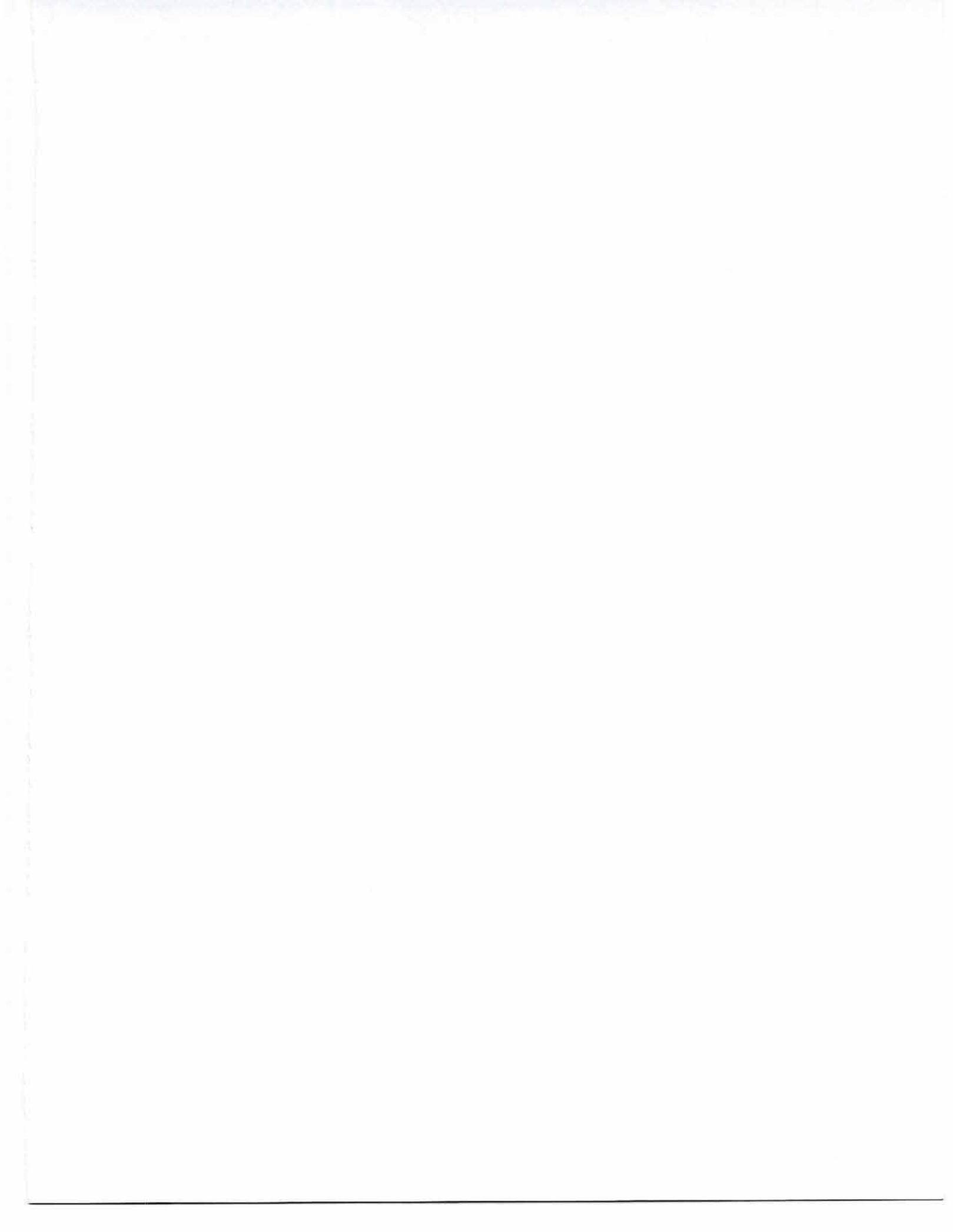
- "Render" excepcionalmente rápido.
- Tratamiento de la dispersión de la luz (radiosidad).
- Luces con intensidad variable según diversos patrones.
- Sombras dinámicas.
- Niebla dinámica.
- Espejos dinámicos.
- Efectos de agua dinámicos.
- Geometría de objetos translúcidos para cristales u otros efectos.
- Mapeado esférico para cielos y horizontes.
- Posicionamiento y atenuación 3D de los sonidos.
- Superposición de imágenes 2D.
- Resolución de pantalla de 640x480 hasta 1024x768 a pantalla completa.

RESUMEN

Cada juego requiere unas especificaciones de hardware diferentes entre sí para que puedan ser ejecutados y se logre apreciar el potencial del juego. Este capítulo explica la manera en que las distintas configuraciones de hardware afectan el rendimiento de los juegos y el motor que será utilizado para el desarrollo de juegos basados en el manual.

TALLER

Instala y ejecuta el Motor Gráfico Genesis3D para que te vayas familiarizando con el ambiente de trabajo.



CAPITULO 4. INTRODUCCION A LAS INTERFACES REQUERIDAS PARA EL DESARROLLO DE LOS JUEGOS DE VIDEO 3D.

Para el desarrollo de Videojuegos es necesario saber con que tipo de interfaces se va encontrar el desarrollador. Las distintas interfaces de desarrollo cumplen un papel determinado en la creación de cualquier Videojuego y que en conjunto hacen que la creación de estos mismos se facilite logrando excelentes resultados.

4.1 HERRAMIENTAS PARA EL DISEÑO 3D.

El diseño 3D ha sido un campo de la informática que se ha desarrollado considerablemente en las últimas décadas gracias a tecnologías desarrolladas por empresas como Silicon Graphics que han diseñado programas que han sido utilizados en películas reconocidas a nivel mundial. En el cine han sido muchas las películas que han sido diseñadas enteramente con software para el diseño 3D.

Los motores gráficos como ya se ha mencionado comenzaron solo siendo simuladores de 3D porque usaban sprites 2D para los personajes y muros de los juegos, pero sus desarrolladores han hecho evolucionar los motores hasta lograr cotas de realismo al nivel del visto en películas.

El diseño de secuencias CG se hace generalmente en estaciones de trabajo Silicon Graphics de US 450,000 y se emplean programas como 3D Studio Max y Maya. Los Videojuegos emplean para su ejecución Hardware de PC's domésticos por consiguiente económico pero se consiguen niveles a la altura de las producciones cinematográficas.

4.2 EDITOR DE NIVELES.

El editor de niveles es una especie de plantilla para crear los diferentes escenarios que contenga el Videojuego. Esta aplicación permite al diseñador añadir y quitar efectos cuando lo desee, en cuanto a la parte de las paredes, techos, suelos y demás objetos puede utilizar estructuras predefinidas como cubos, esferas, cilindros y la combinación de estas; todo esto con el fin de facilitar el trabajo al que este diseñando un escenario. Además están los distintos efectos que se pueden agregar para dar una sensación de realidad a los escenarios que se construyan, entre estos están los rayos, luz y niebla; pero debe saberse que cada uno de los efectos mencionados tienen muchas propiedades modificables que se pueden adecuar para la conveniencia obtener el diseño deseado.



4.3 EDITOR DE EFECTOS PARA EFECTOS EXTERIORES.

El editor de Efectos Exteriores es una aplicación que permite la creación de vistas exteriores para los Videojuegos, el editor de exteriores posee las características idóneas para el manejo de distintos ambientes gráficos que se van a haber representado en los distintos Videojuegos ya que la interfaz que maneja este es muy sencilla y fácil de utilizar, este maneja aspectos como son: los Controles de Render, terreno, agua, nubes, atmósfera, iluminación y las distintas imágenes que son representados y desarrollados en este. Lo importante es mostrar como interactúa esta parte del desarrollo de los Videojuegos como lo es el editor de efectos exteriores con el editor de niveles.



4.4 EDITOR DE PERSONAJES.

Como su nombre lo indica es la herramienta en la cual se crea y se modifican los personajes que se piensan utilizar en todo Videojuego. Los personajes son uno de los aspectos más importantes en los Videojuegos porque son el medio con el cual el jugador interactúa con el medio en el cual se desenvuelve el Videojuego, tal como tocar o recoger objetos, abrir y cerrar puertas o relacionarse con otros personajes.

Esta plantilla ofrece la posibilidad de probar los distintos aspectos concernientes al personaje como son su aspecto físico, su vestimenta, sus habilidades y el papel que tendrá dentro del Videojuego. El desarrollador puede estar en facultad de crear cualquier personaje que desee.

RESUMEN

En este capítulo encontraremos las diferentes interfaces requeridas para el excelente desarrollo de los juegos de videos de 3D. Muy importante es el reconocimiento de las características de cada herramienta que se utiliza en el desarrollo de un videojuego 3D, tales como el manejo del motor o los motores que se van a utilizar y las diferentes posibilidades de cada uno o las herramientas para la creación de gráficos de exteriores.

TALLER

Ejecuta cada una de las herramientas mencionadas y analiza su interfaz detenidamente.



CAPITULO 5. RECONOCIMIENTO DEL AMBIENTE DE TRABAJO DEL MOTOR GRAFICO Y DEL EDITOR DE EFECTOS EXTERIORES.

El conocimiento de las opciones disponibles en toda aplicación que se utilice en la vida diaria es fundamental para sacarle el máximo provecho, esto se aplica a todo; igualmente este tipo de conocimiento es fundamental para entender el funcionamiento del motor que se vaya a utilizar al desarrollar cualquier Videojuego.

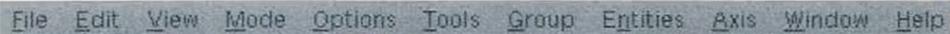
5.1 ENTORNO DE TRABAJO.

El entorno de trabajo para desarrollar un Videojuego ha desarrollado se basa en un Motor Gráfico como herramienta principal y un Editor de Efectos Exteriores para las vistas exteriores que complementan el desarrollo de cualquier Videojuego. El Motor Gráfico ha trabajar es un conjunto de aplicaciones integradas en un paquete, las cuales forman un entorno de trabajo amigable al desarrollador, el cual esta conformado por un Editor de Niveles, un Editor de Personajes y un Editor de Proyectos, los cuales cada uno se encarga de una función diferente en el desarrollo de cualquier Videojuego.

El Editor de Efectos Exteriores es un complemento del Motor Gráfico, con el cual se crean vistas exteriores que dan el aspecto de fundirse en el horizonte, el entorno de esta aplicación es sencillo y se logran buenos efectos.

5.2 INTERFAZ DEL EDITOR DE NIVELES.

El menú del Editor de Niveles esta compuesta por las siguientes opciones.



File Edit View Mode Options Tools Group Entities Axis Window Help

EL MENU:

FILE:

New	Abre un nuevo proyecto.
Open	Abre un proyecto ya existente (*.3DF/*.MAP/*.PNT). Nuestros mundos los cargaremos y salvaremos en .3DF Desde aquí también puedes cargar niveles de Quake I (.MAP)
Save	Salva el proyecto bajo su nombre.
Save As	Salva el proyecto bajo el nombre deseado.
Import	Importa un proyecto (*.3DF) y lo mezcla con el que ya esté cargado. Después de 'Import' aparecen los últimos cuatro proyectos abiertos.
Exit	Salir del programa World Editor.

EDIT:

Delete	Elimina el elemento seleccionado.
Select All	Selecciona todos los elementos.
Deselect All	Deselecciona todos los elementos.

VIEW:

Texture Browser	'No soportado'.
Tool Bar	Permite hacer visible o no la barra de herramientas.
Mode Bar	Permite hacer visible o no la barra de modos.
Group Bar	Permite hacer visible o no la barra de grupos.
Command Panel	Permite hacer visible o no la zona de pestañas.
Status Bar	Permite hacer visible o no la línea de estado.
Leak Finder	'no soportado'.
3D Wireframe	Permite ver una representación alambrada en la vista superior izquierda.
Texture View	Permite ver una representación texturizada en la vista superior izquierda.
Top View	Establece 'top-window' (vista desde arriba), para la ventana que esté seleccionada.
Side View	Establece 'side-window' (vista lateral), para la ventana que esté seleccionada.
Front View	Establece 'front-window' (vista frontal), para la ventana que esté seleccionada.
Zoom In	Realiza un aumento de la vista que esté seleccionada.
Zoom Out	Realiza una reducción de la vista que esté seleccionada.
Show	Selecciona que elementos se desea visualizar:
Current Groups	Muestra los grupos actuales.
Visible Brush Groups	Muestra los grupos de 'brushes' visibles.
All Brushes	Muestra todos los 'brushes'.
Clip Brushes	Muestra todos los 'brushes' tipo 'clip'.
Hint Brushes	Muestra todos los 'brushes' tipo 'hint'.
Detail Brushes	Muestra todos los 'brushes' tipo 'detail'.

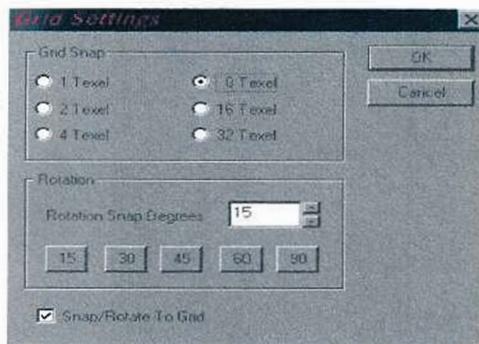


MODE:

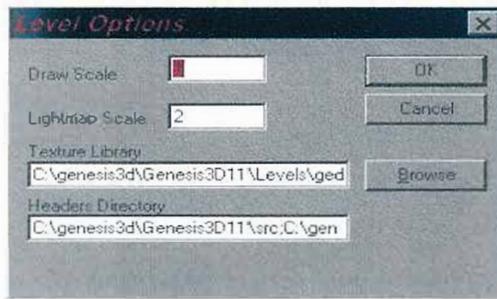
Camera	Establece modo cámara.
Selection	Establece modo selección.
Template	Establece modo 'Template'.
Move/Rotate	Activa 'move/rotate' para mover o rotar el elemento seleccionado. Se mueve con el botón izquierdo del ratón pulsado y moviendo el ratón al mismo tiempo. Se rota con el botón derecho del ratón pulsado y moviendo el ratón al mismo tiempo.
Scale	Activa 'scale' para reescalar el elemento seleccionado.
Shear	Activa 'shear' para poder mover las caras del elemento seleccionado para variar la forma del elemento.
Face Adjustment	Establece modo 'Face Adjustment' para poder seleccionar caras de un 'brush'. No se permite entrar en este modo si está activado 'move/rotate', 'scale' o 'shear'.
Brush Adjustment	Establece modo 'Brush Adjustment' para poder seleccionar 'brushes'.

OPTIONS:

Snap to Grid	Ajusta a la rejilla los elementos que se manipulan.
Grid Settings ...	Abre la siguiente caja para configurar la rejilla:



Auto Rebuild BSP	En la ventana 'textured', muestra los cambios realizados.
Preferences...	Abre una caja para cambiar los colores de la rejilla:
Grid Background	Cambia el color de fondo de la rejilla.
Grid Color	Cambia un color de la rejilla.
Snap Grid Color	Cambia el otro color de la rejilla.
Level Options...	Abre la siguiente caja:



Texture Library: Establece el fichero de texturas utilizado.
 Headers Directory: Establece diversas rutas a ficheros que utiliza el programa.
 Draw Scale: reescala el tamaño del mundo.
 Lightmap Scale: define la intensidad de la luz en nuestro mundo.

TOOLS:

New Brush	Abre la selección de:
Arch	Crea un arco.
Cone	Crea un cono.
Cube	Crea un cubo o habitación.
Cylinder	Crea un cilindro.
Spheroid	Crea una esfera.
Staircase	Crea una escalera o rampa.
Face	Abre la selección de:
Attributes	Abre la caja de propiedades de la cara 'face' seleccionada.
Reset Face	Hace un 'reset' a la cara seleccionada.
Brush	Abre la selección de:
Show Brush	Muestra 'brush' actual.
Rotate 90°	Realiza una rotación de 90° al 'brush' seleccionado.
Attributes	Abre la caja de propiedades del 'brush' seleccionado.
Reset Brush	Hace un 'reset' al 'brush' seleccionado.
Make Newest	Establece el 'brush' actual como el más nuevo.
Build Quick BSP	Recalcula el contenido de la ventana 'Textured'.
Compile...	Abre la caja de 'Compile manager'.
New Lib Object	El 'brush' seleccionado lo añade como un nuevo objeto a la librería. Aparecerá en una lista desplegable en la pestaña 'Template'.
Scale World	Reescala las dimensiones de todo nuestro mundo.
Texture Scale	Reescala las dimensiones de la textura del 'brush face' seleccionado.
Next Brush Face	Selecciona el siguiente 'brush face'.

Prev. Brush/Face	Selecciona el anterior 'brush/face'.
Add Template	Añade un objeto 'brush', de la pestaña 'Template', al mundo.
Toggle Brush/Face mode	Permite cambiar de modo ('brush' o 'face').

GROUP:

Add selected to group	Añade el 'brush' seleccionado al grupo actual.
Remove selected from group	Elimina el 'brush' seleccionado del grupo actual.
Create New Group	Crea un nuevo grupo.

ENTITIES:

Place Light	'No soportado'
Entity Editor	Abre una caja para seleccionar entidades y poder modificar sus propiedades.
Copy Current	'no soportado'
Show	Permite hacer visibles o invisibles las entidades
Entity Visibility	Permite configurar la visibilidad o invisibilidad por tipo de entidad.

AXIS:

- Lock X X - Bloquea el movimiento del eje X en las ventanas de las vistas.
- Lock Y Y - Bloquea el movimiento del eje Y en las ventanas de las vistas.
- Lock Z Z - Bloquea el movimiento del eje Z en las ventanas de las vistas.

WINDOW:

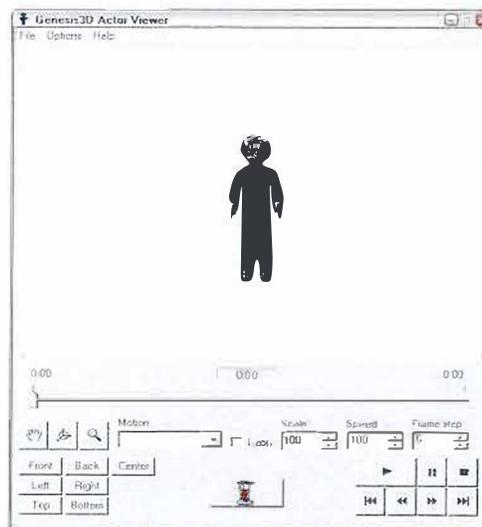
New Window	Abre otra ventana de trabajo con una copia del mundo actual.
Cascade	Reordena las ventanas de trabajo en cascada.
Tile	Reordena las ventanas de trabajo en mosaico.
Arrange Icons	Reordena las ventanas de trabajo que están minimizadas.
	Aquí aparecen las ventanas de trabajo que están abiertas para poder seleccionar una.

HELP:

- Help Topics - Ayuda del programa (en inglés).
- Search for Help On - Búsquedas en la ayuda del programa.
- About GEdit... - Abre la siguiente caja:



5.3 INTERFAZ DEL EDITOR DE PERSONAJES.



Los atajos con el teclado para el Actor Viewer son:

COMANDO	ATAJO
Help	F1
Open file	Ctrl+O
Toggle frame rate display	F
Pan (move) actor	A
Rotate actor	R
Zoom camera	Z
Play motion	P
Pause motion	U
Stop motion	S
Rewind motion	O
Fast forward motion	Q
Back frame	<

Forward frame	>
Front view	N
Back view	B
Left side view	L
Right side view	R
Top view	T
Bottom view	M
Center actor	C

5.4 INTERFAZ DEL EDITOR DE EFECTOS EXTERIORES.

El Editor de Efectos Exteriores es la aplicación con la cual se crean vistas para ser añadidas a los escenarios de los Videojuegos, su principal objetivo es de darle más vida a los escenarios ya que trabaja con renders que son añadidos al escenario. Para explicar mejor se mostrara con un ejemplo: En cualquier escenario, ya sea una casa o una nave espacial, se cuenta con unas vistas exteriores que se ven a través de ventanas, lo que el usuario del Videojuego observa por las ventanas es lo que se crea con el Editor de Efectos Exteriores.

WORLD FILE:

New World (nuevo mundo) restaura la configuración a los valores por omisión, también muestra la pantalla introductoria mientras hace esto.

Open World (Abrir mundo) abre un archivo world de Terragen (.tgworld). Esto contiene toda la configuración con la excepción del terreno, este debe ser cargado a parte desde el cuadro de diálogo Landscape (terreno).

Re-open World (re abrir mundo) vuelve a abrir el archivo world en uso tal como fue guardado por última vez.

Save World (guardar mundo) guarda el archivo world.

Exit (salir) sale del programa.

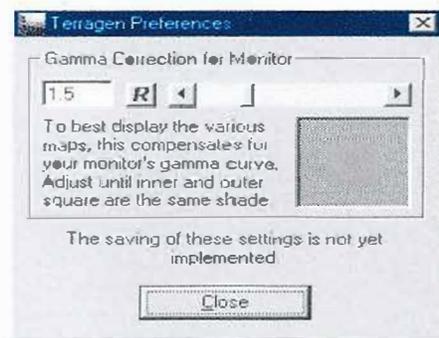
VIEW:

Da acceso a los distintos cuadros de diálogo.

TERRAGEN:

Execute Script (ejecutar script) permite seleccionar un archivo script para ser ejecutado. Antes de esto debe asegurarse de haber revisado la configuración atmosférica, de render y de nubes y de que se ha cargado el terreno.

Preferences permite establecer el factor de corrección gamma que es utilizado al dibujar los mapas. Configurar esto de manera correcta hace más fácil el trabajo con los mapas. Para ajustar la corrección de gamma se debe mover la corredera para que tanto el cuadrado exterior como el interior tengan el mismo tono de gris.



Run Scripter ejecuta el programa de Matt Fairclough, el Scripter.

Run Terranim ejecuta el programa de Daniel Parnham, *Terranim*, que es mucho más flexible que el Scripter.

Run CamPath ejecuta el programa de Sean O'Malley's, *CamPath*. Este programa está más avanzado que Terranim.

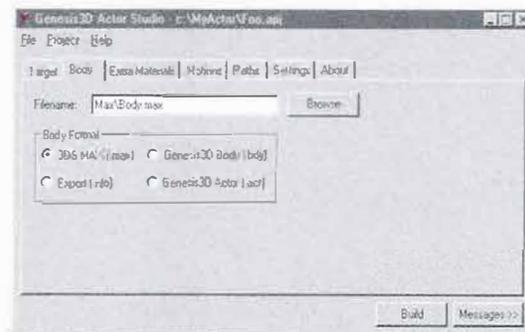
HELP:

User Guide muestra la dirección de la versión inglesa de esta guía.

About Muestra la pantalla introductoria una vez más. Esto provee la información de la versión del programa, de copyright así como el nombre del usuario registrado si el programa ha sido registrado.

Como todavía no existe un archivo de ayuda esta guía está diseñada para tomar su lugar. En algún momento del futuro se implementará en Terragen la ayuda del tipo "¿Qué es esto?".

5.5 INTERFAZ DEL EDITOR DE PROYECTOS:

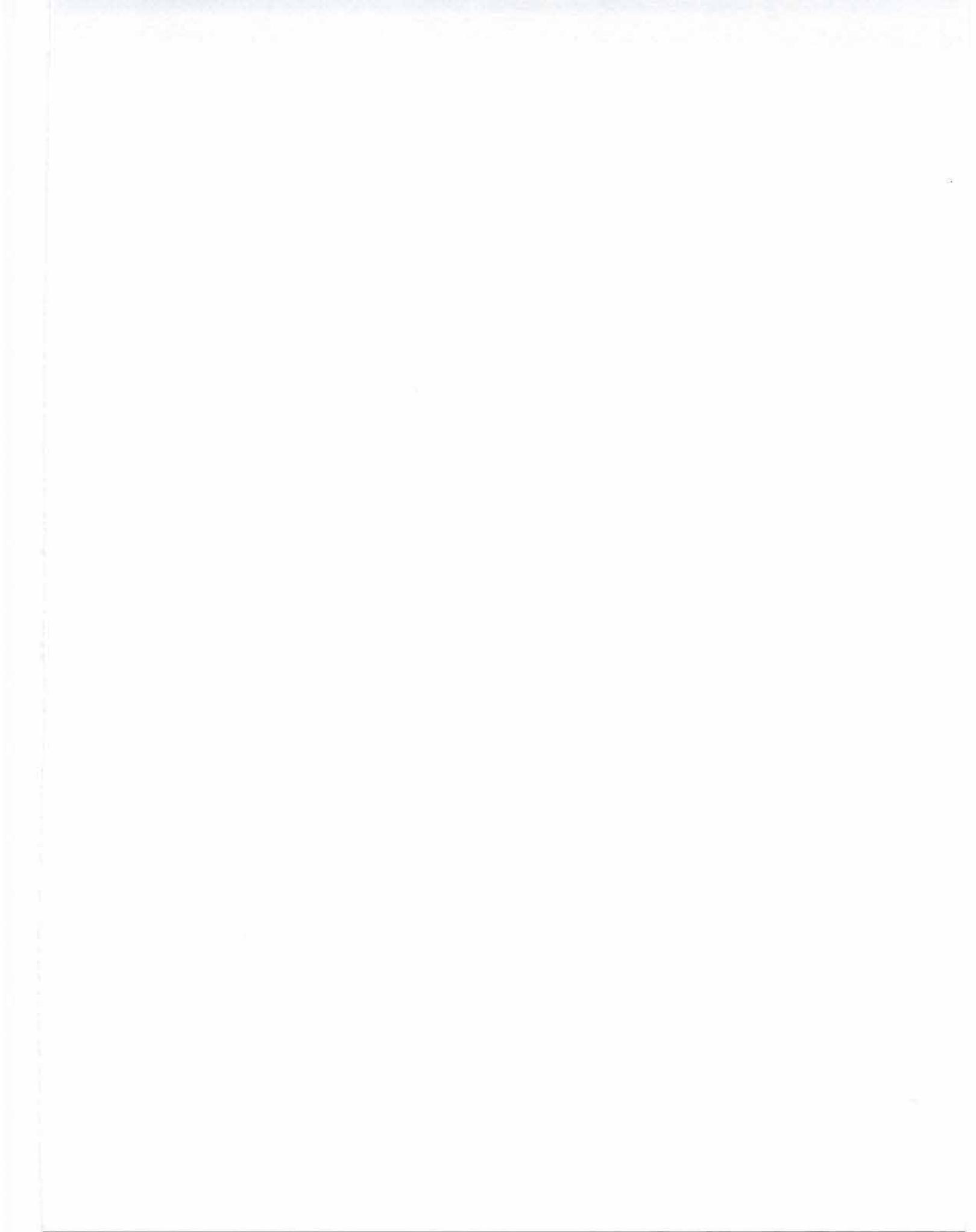


RESUMEN

El objetivo de este capítulo es el de explicar el entorno de trabajo del Motor Gráfico y el Editor de Efectos Exteriores, como son las distintas interfaces (opciones, menús y comandos), además de todos los elementos con que se cuentan. Esto con el fin de facilitar el trabajo y hacer que la utilización del manual sea menos engorrosa, mediante el entendimiento de los diferentes elementos se consigue un trabajo más eficiente con el fin de agilizar el proceso.

TALLER

Ahora que ya conoces un poco mejor las interfaces de los diferentes programas prueba todas las opciones explicadas para que veas como funcionan.



CAPÍTULO 6. CREACIÓN DE ENTORNOS 3D Y EDITOR DE NIVELES.

En los Videojuegos 3D el escenario es el fundamento de la acción, el cual otorga el grado de realismo logrando hacerlo mas ameno para los jugadores; para lograr este impacto se hace necesario hacer uso de las herramientas para la creación de los distintos niveles, los cuales son las encargadas de hacer posible el desarrollo o creación de todos estos efectos que hacen mas real el ambiente de juego para hacer que los usuarios disfruten de este y tengan un gran impacto en cuanto a su entorno gráfico.

6.1 CREACION DE ENTORNOS 3D.

A través de la historia del cine y los Videojuegos se ha ido evolucionando en el desarrollo de CG (Gráficos por Computador). Desde los principios de la utilización de los computadores para el desarrollo de películas se ha ido en ascenso.

PELICULAS SOBRESALIENTES.

Tron (1982): Aunque no fue completamente creada en computador, fue la primera en utilizar técnicas de animación computarizadas en varias de sus escenas, junto a personajes de carne y hueso.

The Last Starfighter (1984) Utilizó gráficas por computador para las batallas de naves interestelares, las cuales reemplazaron a los típicos modelos de escala. Fue la primera película en utilizar objetos creados en computador para representar elementos del mundo real.

Las Aventuras de Andre y Wally B (1984), creada por el estudio Pixar, fue el primer cortometraje realizado completamente por computador. Incorporaba efectos que daban la sensación de movimiento (motion blur).

Luxo Jr. (1986), en un corto animado cuyos personajes son dos lámparas y una pelota. Fue dirigido por John Lasseter, uno de los genios creativos de Pixar en la actualidad y director de *Fox Story*. Aunque su animación es bastante básica, constituye uno de los fundamentos para las espectaculares obras animadas de la actualidad. Fue la primera producción con gráficas por computador nominadas al premio Oscar.

Willow (1988). Primera cinta en presentar el efecto especial computarizado de la transformación de un personaje (morphing).

El Secreto del Abismo (1989). Esta película, que narra una historia subacuática con extraterrestres incluidos, monstruo por primera vez el 'agua digital', un objeto que permite crear figuras con apariencia líquidas.

El Vengador del Futuro (1990). utilizó personajes totalmente creados en computador, como aquellos que en un aeropuerto se ven detrás de una pared de rayos X y los efectos de sus ojos saliéndose de sus órbitas de los protagonistas en la atmósfera de Marte.

Terminator 2 (1991). los efectos de transformación de los antagonistas y los movimientos de su cuerpo, mientras adopta la forma de 'hombre de metal', son totalmente digitales.

Jurassic Park (1993). la cinta pulió tanto la técnica de generar personajes por computador, que el resultado salta a la vista: dinosaurios con aspectos reales compartiendo escenario con personajes humanos.

Waterworld (1995). un mundo en un futuro apocalíptico cubierto de agua en el que el líquido es creado en computador. La diferencia con otros efectos de agua es que aquí son bastante realistas.

Toy Story (1995). esta aventura en el reino de los juguetes rompe la historia de la animación en dos y fue el comienzo de una carrera exitosa para los estudios Pixar. Es considerado el primer largometraje completamente realizado por computador. Además, la historia es muy sólida y atractiva.

El episodio I de la guerra de las galaxias: La amenaza fantasma (1999). Va un paso más allá respecto a lo realizado por Jurassic Park. Los personajes creados en un computador interactúan con personajes de carne y hueso de una forma más cercana que en la cinta sobre dinosaurios.

El señor de los anillos: las dos torres (2002). se fabricaron digitalmente ejércitos enteros que tomaron parte en las épicas batallas de esta cinta, ganadora de varios premios Oscar.

The Matrix Revolutions (2003). un puño en la cara que jamás ocurrió en la filmación real. Por supuesto, el efecto fue creado digitalmente.

El capitán Sky y el mundo del mañana (2004). presenta a todos sus personajes de carne y hueso en ambientes generados por computador.

Los Increíbles (2004). Lleva los efectos especiales en películas creadas por computador a un nivel superior de realismo y emoción.

Mención aparte el desarrollo de CG's de algunas compañías japonesas que en su mayoría las aplican a Videojuegos para consolas. Una de las compañías líderes en este campo es Squaresoft que con su saga de Juegos Final Fantasy la ha sido el producto que ha ido a la vanguardia en cuanto a evolución gráfica en las distintas consolas que ha aparecido desde el NIS hasta el moderno Playstation2. Entre todos las ediciones de Final Fantasy, la que represento un salto cualitativo importante fue el Final Fantasy VII que aprovechando el hardware gráfico de Playstation logro un entorno de juego basados en renders muy preciso y secuencias CG's de inmensa calidad que compiten o se puede considerar en calidad a las películas animadas de Hollywood. En el 2001 Squaresoft sacó a la pantalla grande un largometraje titulado "Final Fantasy Spirits Within", aunque no logro un gran éxito taquillero, asombro al mundo entero por la calidad gráfica de los personajes. Después de esto Squaresoft tuvo problemas económicos lo que obligo a la compañía a fusionarse con Enix llamándose Square-Enix, posteriormente se lanzaron las tres versiones para Playstation2 Final Fantasy X, Final Fantasy X-2 y Final Fantasy XI (Versión online). En el 2004 lanzaron final Fantasy VII Advantge Children.

Para crear entornos 3D se debe trabajar con aplicaciones incluidas en el motor gráfico, las cuales dependiendo del motor pueden ser programables o mediante el uso de interfaces en las cuales se facilita el trabajo ya que se manipula el entorno directamente.

Las aplicaciones en las cuales se necesita programar se denominan lenguajes de creadores de matices entre las cuales están DirectX 8, DirectX 9 de Microsoft, OpenGL, producto de un consorcio de distintas empresas y que será usado como una de las interfaces de programación de la consola Playstation3 y Cg de Nvidia. Las aplicaciones

en las cuales se interactúa con el escenario son propias de algunos Motores Gráficos entre ellos la del motor Genesis 3D.

6.2 EDITOR DE NIVELES.

Como se menciono anteriormente el Editor de Niveles es una plantilla con la cual se crean los distintos niveles o habitaciones de un Videojuego. Varios de los juegos que se comercializan actualmente incluyen Editores de Niveles los cuales alargan la vida de los juegos, algunos de ellos son Doom3, Half-Life, Quake3.

El Motor Gráfico Genesis 3D incluye un Editor de Niveles llamado World Editor el cual ofrece una interfaz similar al programa 3D Studio Max, con la particularidad de poder modificar la apariencia (textura, tamaño, forma, posición) de los objetos que se encuentran dentro de una habitación que se componen de faces (caras), las cuales tienen la posibilidad de modificar su textura de manera independiente; después esta la posibilidad de unir los distintos niveles que se vayan creando para así formar un Mundo completo, que es aquel que tiene una extensión mas grande, en consecuencia de la unión de habitaciones; es de gran importancia tener en cuenta que los distintos efectos especiales como la iluminación, niebla y rayos que también tienen diferentes propiedades modificables entre los que están la intensidad y la tonalidad para la iluminación; en la niebla están la densidad, el color y la profundidad(espesor); para los rayos se pueden alterar parámetros como la intermitencia, la intensidad y el tamaño; con los cuales se obtiene realismo en el ambiente del juego.

Otra parte muy importante del Editor de Niveles es que los personajes deben ser agregados como entidades a la superficie del nivel, el cual tiene la particularidad de ser una caja de mayor tamaño que los objetos y entidades incluidas en cada nivel, pero la particularidad de esta caja 'grande' es que su consistencia es 'invisible' para después agregarle textura a sus cielo, paredes y suelo; una vez que se tiene esto, se sabe que un juego no puede constar de una sola habitación por tanto el World Editor de Genesis 3D permite 'pegar' las habitaciones que se tengan pero con la restricción de que sean del mismo tamaño; también se pueden agregar puertas entre las habitaciones para dar una sensación de realismo a cada nivel, dichas puertas no son mas que otras cajas solo que mas pequeñas que las habitaciones, pero agregándole la particularidad de ser huecas en su interior.

A continuación una sencilla muestra de lo que se puede lograr con el editor de niveles el cual se va a explicar en los capítulos siguientes.



RESUMEN

En todo juego de video se necesita diseño y creación de distintos niveles y entornos para así tener una división organizada en la cual podamos desplazarnos para contar con un orden en el desarrollo de un juego, y así tener aspectos que tengan semejanza con la realidad; pero en sí el contenido de este capítulo se aprenderá y practicará la edición de niveles y entornos teniendo en cuenta las especificaciones de cada motor gráfico citado en este manual debido al hecho que cada motor presenta características disímiles para la creación de los escenarios.

TALLER

Ya que se va entrando en materia con la creación de niveles, debe ir familiarizando con la armonía de estos para poder crear juegos armoniosos y realistas. Observe las características que tienen estos en algunos Videojuegos.



CAPITULO 7. CREACION Y MANEJO DE OBJETOS SÓLIDOS.

7.1 CREACION Y EDICION DE HABITACIONES.

El Motor Gráfico ofrece una herramienta importante como lo es el Editor de Niveles, el cual permite la creación de todo el entorno en que se desenvuelve el jugador. En el Editor de Niveles del Genesis3D, llamado World Editor encontramos diversas opciones para diseñar el nivel como le plazca al desarrollador, brindando flexibilidad a su trabajo y logrando que se logre plasmar su diseño fielmente.

Para poder crear y editar habitaciones se debe tener claro los conceptos de Brush y Faces. Un Brush sería cualquier objeto que contenga el nivel y un Face es cualquiera de las caras que tiene un Brush y a la cual se le aplican texturas, clave aclarar que no a todos los Brush se le aplican texturas salvo a objetos sólidos como muros, escaleras, pisos; por lo general si son figuras geométricas. El saber entender y aplicar estos conceptos ayuda en el desarrollo de cualquier Videojuego.

A continuación se detalla cada uno de los Brushes explicando sus características principales y su importancia.

7.1.1 BLOCK ACTOR.

Básicamente son los distintos tipos de figuras geométricas que se pueden utilizar en los diseños de los distintos niveles; son seis figuras que al manipularles su tamaño y forma se obtienen derivaciones de las figuras básicas, y al modificar sus propiedades se crean ambientes más impresionantes.

7.1.1.1 BOX.

El Box es un cubo que tiene distintas utilidades, como por ejemplo una habitación es básicamente un cubo hueco. El Box tiene distintas propiedades y características como:

- Solid: En este caso el Box es un objeto sólido dentro del escenario siendo algo que obstaculizaría el andar del personaje o lo ayudaría a desplazarse dentro del mismo.

- Hollow: El Box es un objeto hueco el cual es útil para la creación de habitaciones o pasadizos que conectan dos o más habitaciones permitiendo ampliar el espacio por el que se mueve el personaje manejado por el jugador.

- Cut Brush: Es una característica de Box el cual permite la conexión de dos habitaciones a través del pasadizo.

7.1.1.2 SPHEROID.

El Spheroid es una figura con la cual se construyen objetos de forma esférica o semiesférica para darles ambientación a los distintos niveles.

Las características del Spheroids son las mismas del Box al igual que el resto de Brush.

- Solid: Al igual que el Box es un objeto sólido dentro del escenario siendo algo que

- obstaculizaría el andar del personaje o lo ayudaría a desplazarse dentro del mismo.
- Hollow:** Con esta propiedad aplicada el Spheroid sería una estructura hueca y su uso queda bajo juicio del desarrollador.
- Cut Brush:** En esta propiedad el Spheroid es un objeto no tangible dentro del nivel.

7.1.1.3 CYLINDER.

- El Cylinder es un objeto sólido en forma de cilindro, uno de sus posibles usos es crear una columna, o un túnel entre dos habitaciones.
- Las características del Cylinder son las mismas que para todos los Brush.
- Solid:** Define si el cilindro va a ser sólido, afecta los movimientos del personaje ya que ocupa espacio dentro de una habitación.
- Hollow:** Con esta propiedad se puede crear una habitación de manera cilíndrica.
- Cut Brush:** En esta propiedad el Cylinder es un objeto no tangible dentro del nivel.
- Ring:** El Cylinder toma la forma de un anillo con un agujero en el centro.

7.1.1.4 STAIRCASE.

- El Staircase no es más que una escalera común y corriente, sirve para desplazarse en una habitación que tenga varios pisos.
- Las características de Staircase son las siguientes:
- Cut Brush:** En esta propiedad el Staircase es un objeto no tangible dentro del nivel.
- Make Ramp:** con esta propiedad se puede determinar si el Staircase va tener escalones o será lisa como una rampa.

7.1.1.5 ARCH.

- El Arch es un arco el cual se constituye en algo importante para el desarrollo de los distintos niveles. El Arch de acuerdo a la posición en que se encuentre puede servir como conector entre distintas plataformas que pueda contener el nivel, el Arch pueden tener cualquiera de las siguientes cuatro características:
- Solid:** El Arch se define como un objeto sólido y es tangible al personaje que es controlado por el jugador.
- Hollow:** Como ocurre con los Block Actor anteriores con esta propiedad se crea Arch huecos lo que permite crear secciones del escenario con forma de arco.
- Ring:** Con esta propiedad se crea un Arch con una abertura en sus extremos y que da a una cavidad interior, esto ayudaría para la creación de cavidades.
- Cut Brush:** El Arch sería un objeto intangible para el personaje siendo incapaz de interactuar el Arch con el medio.

7.1.1.6 CONE.

- Como su nombre lo indica este Block Actor es un cono el cual puede ser útil para crear ciertas habitaciones con forma cónica o ciertas partes de un nivel especial. Las propiedades de Cone son:
- Solid:** Con esta propiedad el Cone es tangible al personaje e interactúa con él.
- Hollow:** El Cone es hueco siendo propicio para la creación de habitaciones.

Funnel: Es una especie de combinación de los dos anteriores, en este el Cone es sólido pero tiene una abertura en la parte inferior.

Cut Brush: El Cone es intangible.

7.1.2 BOI ACTOR START.

Este Brush (objeto) sirve para agregar personajes secundarios y enemigos a los distintos escenarios. Se pueden agregar tantos personajes secundarios y enemigos como se deseen pero teniendo en cuenta que estos son aplicables a Videojuegos con trama argumental, ya que estos personajes al morir no pueden aparecer más.

7.1.3 BOI MATCH START.

Esta opción se puede utilizar para agregar enemigos para opciones de juego DeathMatchStart en las cuales se hacen enfrentamientos con el objetivo de conseguir más puntos eliminando contrincantes, en este tipo de opción el enemigo aparecerá las veces que sean necesarias.

7.1.4 CORONA.

Es un efecto lumínico que sirve para agregar un tipo de luz, con la particularidad de que se desvanece cuando el personaje controlado por el Videojugador se aleja de ella, es importante no sobrecargar una habitación con tantos efectos, por tanto es vital no perder de vista la idea original que se concibe para la creación de un Videojuego.

7.1.4 DEATHMATCH START.

Con esta opción se logra agregar el lugar en el que aparecerá el personaje que va a ser controlado por el Videojugador en una habitación que ha sido construida previamente, se pueden agregar tantos DEATHMATCHSTART como se desee para así lograr que el personaje aparezca en distintos lugares de una habitación durante una partida.

7.1.4 DOOR.

Se utiliza esta opción para identificar el punto en que se activa una puerta. La puerta en sí es un Box con la opción de Cut Brush y el Door lo que hace es activar la opción de abrir y cerrar la puerta.

7.1.5 DYNAMIC LIGHT.

Como se indica por el nombre es una luz dinámica, la cual tiene como propiedad un continuo parpadeo. Con este tipo de iluminación se puede crear ambientes densos en los cuales si hay total ausencia de luz, esta opción permite crear la sensación de suspenso.

7.1.6 ELECTRIC BOIT.

Esta opción es la que permite colocar el punto de origen de un rayo o relámpago, aquí se

Se puede alterar el color, la intensidad y el tiempo de flasheo, este efecto es adecuado para escenarios en los que se activen switches o dispositivos eléctricos.

7.1.7 ITEM ELECTRIC BOLT TERMINUS.

Se utiliza para indicar en que lugar termina el rayo o relámpago que se crea con la opción Electric Bolt.

7.1.8 ITEM FOG LIGHT.

Este efecto luminoso sirve para crear niebla dentro de los escenarios, este efecto sirve para otorgar dramatismo al escenario en donde se presente.

7.1.9 ITEM ARMOR.

Mediante el uso de esta opción se puede agregar al escenario un elemento que sirve para darle una protección extra al personaje que la seleccione, reduciendo el daño ocasionado por los ataques, la única propiedad alterable en este elemento es su posición en la habitación, se pueden agregar tantos elementos de estos como se desee pero sin exagerar en su uso.

7.1.10 ITEM GRENADE.

Agregando este ítem se consigue diversificar el armamento que se puede encontrar y utilizar en una habitación, consiste en un lanzagranadas, la única propiedad alterable en este elemento es su posición en la habitación, se pueden agregar tantos elementos de estos como se desee pero sin exagerar en su uso.

7.1.11 ITEM GRENADE AMMO.

Con esta opción se puede agregar munición para el lanzagranadas y así estar mejor provisto para una batalla, la única propiedad alterable en este elemento es su posición en la habitación, se pueden agregar tantos elementos de estos como se desee pero sin exagerar en su uso.

7.1.12 ITEM HEALTH.

Si se agrega este elemento a una habitación, los distintos personajes que combatan en un nivel pueden recuperar puntos de vida perdidos en una batalla, la única propiedad alterable en este elemento es su posición en la habitación, se pueden agregar tantos elementos de estos como se desee pero sin exagerar en su uso.

7.1.13 ITEM ROCKET.

Mediante el uso de este ítem se consigue diversificar el armamento que se puede encontrar y utilizar en una habitación, consiste en un lanzacohetes, la única propiedad alterable en este elemento es su posición en la habitación, se pueden agregar tantos

elementos de estos como se desee pero sin exagerar en su uso.

7.1.14 ITEM ROCKET AMMO.

Con esta opción se puede agregar munición para el lanzacohetes y así estar mejor provisto para una batalla, la única propiedad alterable en este elemento es su posición en la habitación, se pueden agregar tantos elementos de estos como se desee pero sin exagerar en su uso.

7.1.15 ITEM SHREDDER.

Agregando este ítem en una habitación se logra diversificar el armamento que se puede encontrar y utilizar en un cuarto, consiste en una ametralladora, la única propiedad alterable en este elemento es su posición en la habitación, se pueden agregar tantos elementos de estos como se desee pero sin exagerar en su uso.

7.1.16 ITEM SHREDDER AMMO.

Con esta opción se puede agregar munición para la ametralladora y así estar mejor equipado al momento de una batalla, la única propiedad alterable en este elemento es su posición en la habitación, se pueden agregar tantos elementos de estos como se desee pero sin exagerar en su uso.

7.1.17 LIGHT.

Consiste en agregar luces que den la sensación de realidad a una habitación, esta entidad tiene muchos patrones que pueden ser modificados, tales como el color y su luminosidad.

7.1.18 SPOT LIGHT.

Es el radio que formaría un cono que representa la boca de salida del foco, si se cierra (menor ángulo) la luz se proyectará más pequeña y concentrada, mientras que si lo abrimos (más ángulo) la luz se proyectará más grande y expandida.

7.1.19 CREACION DE UNA HABITACION.

Antes de comenzar a crear niveles de gran envergadura como los que aparecen en los Videojuegos que hay actualmente en el mercado se tiene que comenzar por cosas sencillas. Para crear habitaciones lo que se debe pensar primero es la forma y el tamaño que esta va llevar, se debe escoger un Block Actor adecuado: para el primer intento se debe considerar un Box o el Cylinder, para comenzar se debe asegurar que el modo Template este seleccionando esto se hace presionando la tecla T o haciendo click sobre la T azul que aparece en la barra de opciones rápidas. Después de esto se selecciona el Block Actor que se piense utilizar, si se quiere utilizar un Box se hace click sobre la pestaña Templates a la derecha de la interfaz, se selecciona Block Actor y se hace click sobre el la pestaña del Box; seguidamente se abre una ventana emergente en donde se definen todas las opciones del Box como el tamaño, la altura y como no es importante se



debe escoger la opción Hollow indicando que el Box va a ser vacío, determinadas las opciones se hace clic en OK y por último se debe presionar Enter para agregar el Box al ambiente del Nivel, para tener en cuenta que todos los Brushes (Actors) se agregan presionando Enter después de definirles sus propiedades.

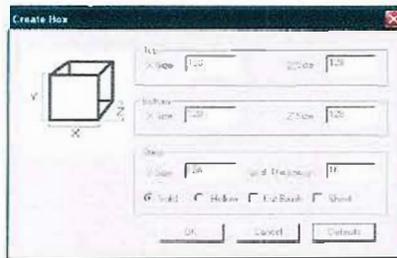
Ya se tiene agregado el Box al ambiente, lo siguiente será agregarle las Texturas a las distintas Faces (Caras) pero para esto se debe cambiar el modo de trabajo de Brush Adjustment (donde se alteran las características físicas al Brush) a Face Adjustment (modo donde se les agrega texturas a las caras del Brush), esto desde la opción Mode en la barra de Menú. Para visualizar como se van dando los cambios dentro de la habitación es preciso colocar la cámara en una posición adecuada (esto se hace activando la opción de Vista con la tecla C o haciendo clic en la pestaña con forma de ojo), las Caras se alteran dando clic sobre cada una de ellas y en la pestaña Textura de la parte derecha de la interfaz se busca la textura que se le quiera aplicar.

Para comprobar el resultado se debe agregar un Death Match Start en el escenario (punto de partida del personaje controlado) en lugar que se crea más conveniente tratando de evitar que no se produzcan errores al momento de la compilación que se efectúa dando clic sobre el botón de la barra de herramientas con forma de C.

7.1.20 AGREGAR BLOCK ACTORS AL ESCENARIO.

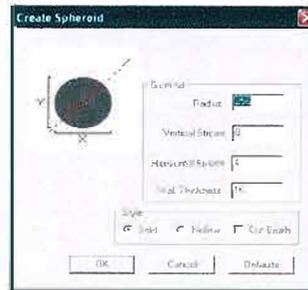
Los seis Block Actors con los que cuenta el World Editor presentan una serie de ventanas emergentes en las cuales se personalizan cada una de sus características.

CREATE BOX



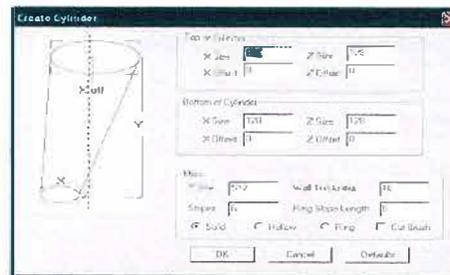
En esta ventana se definen las características del Box, tamaño, altura, ancho y grosor del Box. Además las propiedades Solid, Hollow, Cut Brush y Sheet son acerca de su presencia física dentro del entorno de juego.

CREATE SPHEROID



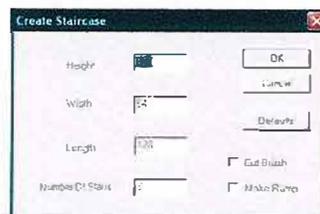
En la interfaz se muestra las características que se pueden alterar a los Esferoides. En General están: Radius que permite determinar el grosor de la esfera, Vertical Horizontal stripes que determina la cantidad de poligonos que formaran la esfera y Wall Thickness que determina el grosor de la esfera. En Style se encuentran las propiedades Solid, Hollow y Cut Brush las cuales ya han sido explicadas anteriormente.

CREATE CYLINDER



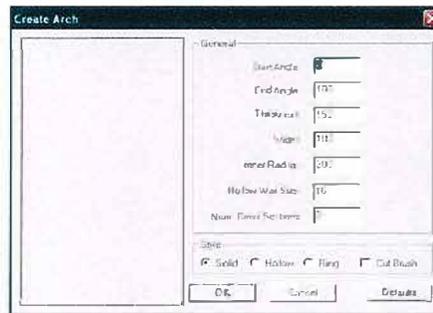
En esta ventana se determina las características que tendrá cualquier Cilindro que se desee crear. En Top of Cylinder se dan los valores para la parte superior del Cilindro, en Bottom of Cylinder se dan los valores para la parte inferior del Cilindro, en Misc se determina la altura del cilindro, Wall Thickness el grosor del Cilindro, los atributos Solid, Hollow, Ring, Cut Brush corresponden a la manera de visualizar el cilindro.

CREATE STAIRCASE



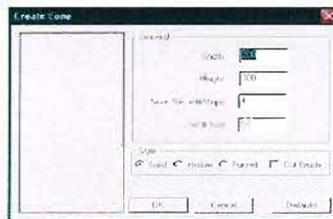
En esta interfaz se modifican los valores del Block Actor con forma de Escalera. En Height se establece la altura de la Escalera, en Width se establece el ancho de la Escalera, en Length se determina el largo de la Escalera, en Number Of Stairs se establece el número de escalones, la propiedad Cut Brush sirve para que la Escalera sirva de corte para otros Brushes y Make Ramp es para determinar si se quiere una rampa.

CREATE ARCH



En la ventana para la creación de Arcos se especifican las propiedades con que aparecerá cualquier arco que se cree. Start Angle determina el ángulo de inclinación con que comenzara el Arco. End Angle indica el ángulo con que terminara. Thickness equivale al grosor del arco. Width es el ancho del arco visto de lado. Inner Radius es el ancho del interior del arco. Num. Cross Sections sirve para colocar el número de secciones. los atributos Solid, Hollow, Ring, Cut Brush corresponden a la manera de visualizar el Arco.

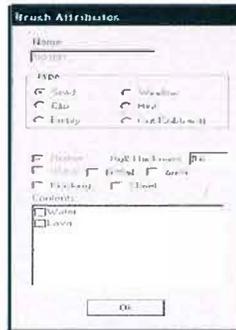
CREATE CONE



En esta ventana se muestran las propiedades que tendrá cualquier cono que se cree. Width indica el ancho de la base. Height indica la altura del cono. Num. Vertical Strips equivale a la cantidad de caras verticales que tiene si se colocan cuatro se creara una pirámide. vale aclarar que entre mas caras tenga mas circular será. Wall Size es el grosor de su estructura. los atributos Solid, Hollow, Funnel, Cut Brush corresponden a la manera de visualizar el cilindro.

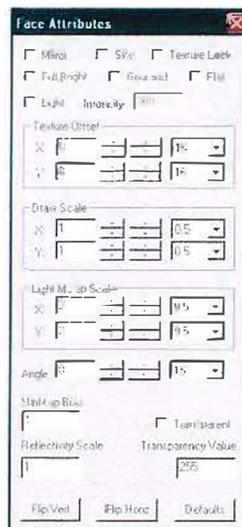
CAJA DE PROPIEDADES DE LOS BRUSHES.

Esta caja de propiedades se accede cuando se esta trabajando en Brush Adjustment y se llega a ella desde la opción Tools de la barra de menús. opción Brush. Attributes. En ella se modifican las propiedades de los Brushes para alterar su constitución o naturaleza.



CAJA DE PROPIEDADES DE LOS FACES.

Esta caja de propiedades se accede cuando se está trabajando en Faces Adjustment y se llega a ella desde la opción Tools de la barra de menús, opción Faces, Attributes. En ella se modifican las propiedades de los Faces para alterar su constitución o naturaleza.



AÑADIR NUEVAS TEXTURAS PARA LOS FACES

- Las texturas que utiliza 'World Editor' las extrae de ficheros con extensión .txl, estos ficheros normalmente están en la misma carpeta que los ficheros .3dt o .bsp que definen los niveles.

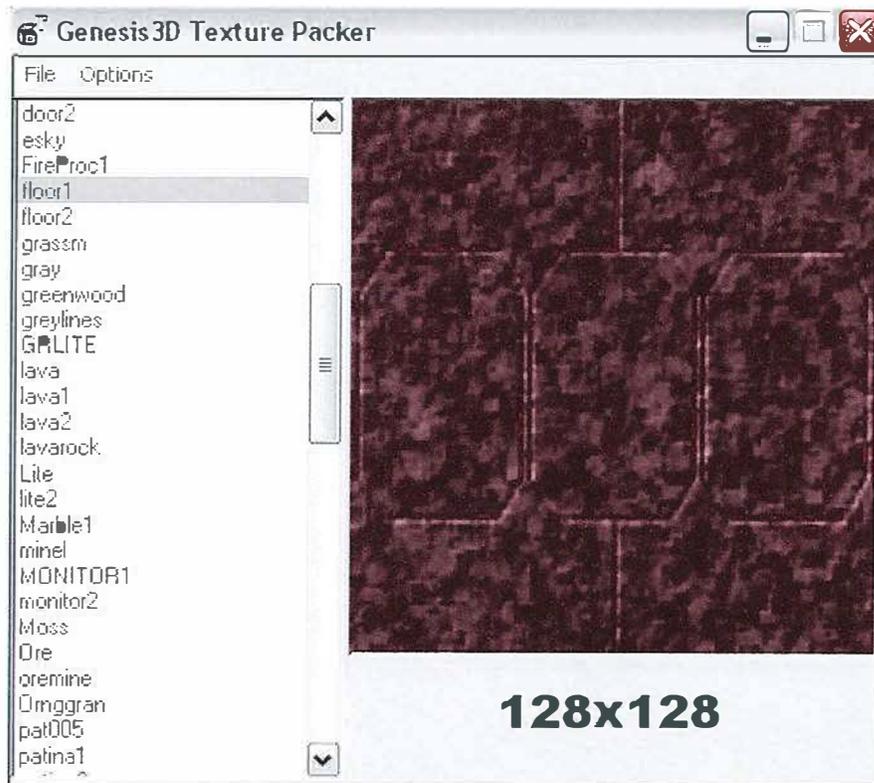
Estos ficheros .txl son en realidad una recopilación de texturas en formato .bmp. En el caso de Entidad 3D el fichero que utiliza 'World Editor' se llama 'texturas.txl', y si queremos añadir alguna textura a nuestro escenario creado con 'World Editor', ese es el fichero de texturas que debemos modificar para que las nuevas texturas nos aparezcan en la pestaña de texturas de 'World Editor'.

Para añadir nuestras propias texturas a nuestro fichero .txl tendremos que utilizar el programa Texture Packer. Al igual que 'World Editor', esta herramienta se te habrá instalado en tu sistema cuando instalaste Genesis3D y la podrás encontrar en la carpeta principal de Genesis3D (por defecto llamada 'Genesis3D1'). El programa en concreto se llama 'tpack.exe'.

Ejecutamos Texture Packer y con 'File Open' abrimos el fichero 'texturas.txl' que encontraremos en la carpeta 'levels'. Una vez abierto podremos ver todas las texturas que contiene este fichero de texturas. Para añadir nuestra textura sólo tendremos que arrastrar a la lista de texturas el fichero gráfico .bmp que habremos diseñado como nueva textura.

Una vez arrastrados todos los .bmp que queríamos añadir, procederemos a salvar el nuevo 'texturas.txl' con la opción del menú 'File Save'.

Puedes arrastrar diversos .bmp al mismo tiempo, pero si arrastras demasiados el programa Texture Packer te podría dar un error.



Una cosa muy importante es que no podemos añadir texturas .bmp de cualquier tamaño o profundidad de color.

La profundidad de color para cada textura ha de ser de 255 colores, sólo crear texturas que utilicen los colores del 0 al 254, el color 255 de la paleta de colores no debes

utilizarlo, de lo contrario la textura podría aparecer con unas manchas negras en tu escenario.

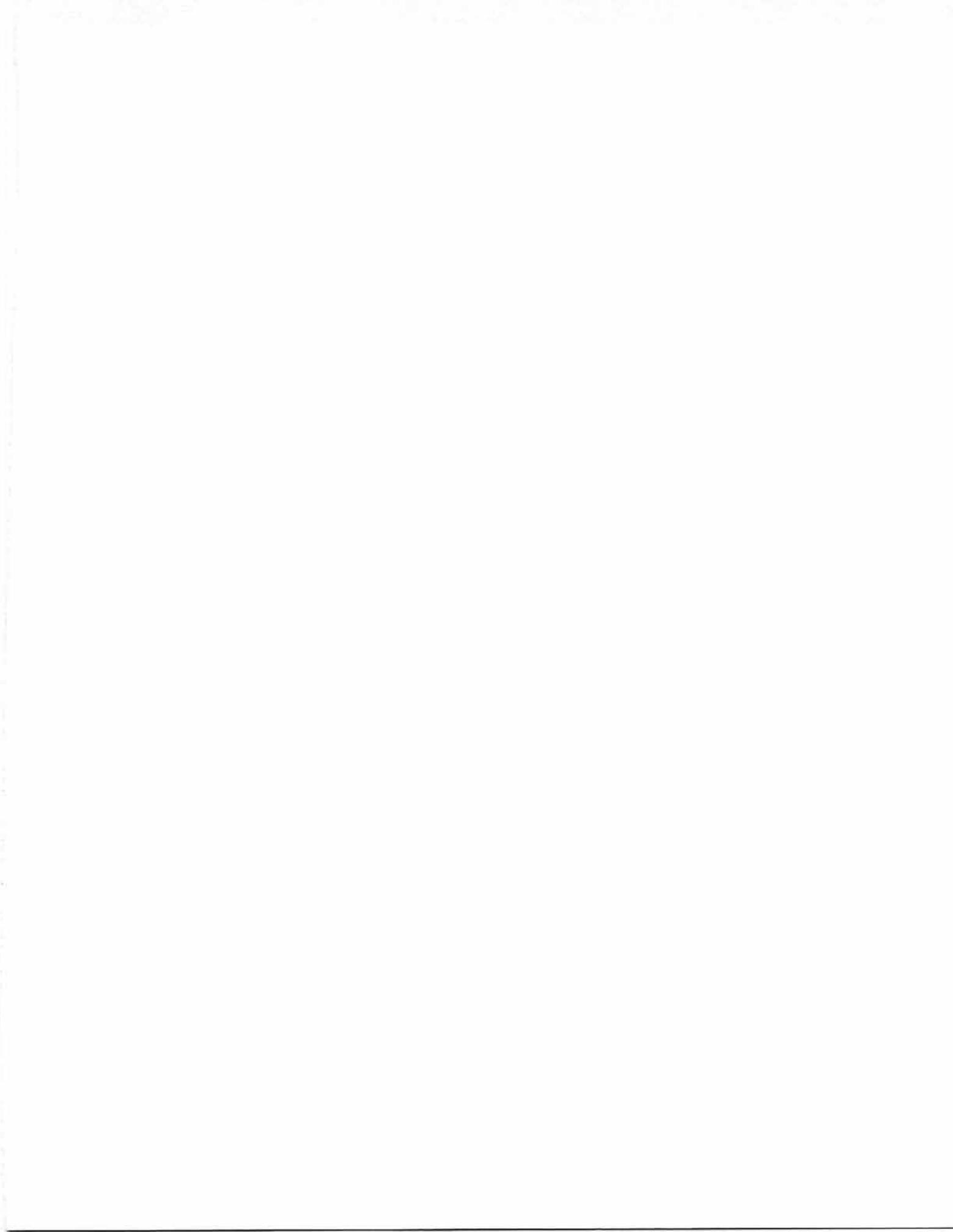
Las dimensiones de las texturas sólo pueden ser de: 64x64, 128x128 o 256x256 píxeles.

RESUMEN

En esta parte se dará a conocer el proceso de creación de los objetos sólidos que se observarán durante el transcurso de los Videojuegos, estas una parte muy importante en juego ya que sin nada sólido en los niveles estos serían ambientes vacíos y no darían una sensación agradable al usuario; además no solo es crear objetos uno tras otro sino que debe mantenerse una armonía que cree sensación de realismo, agregando texturas que mejoren el aspecto de los objetos.

TALLER

● Crea una habitación utilizando World Editor y agrégale todos los objetos que quieras y aplícales textura.



CAPITULO 8. MANEJO DE CRISTALES, ESPEJOS Y LIQUIDOS.

En los Videojuegos actuales se dan muchos efectos de reflejos, líquidos y reflejos, otorgándole un realismo que hace años era muy difícil imaginar.

Los cristales, espejos y líquidos son alteraciones que se hacen sobre los Brushes, la colocación de estos efectos depende exclusivamente del criterio del diseñador quien deberá precisar donde son los lugares más adecuados.

8.1 MANEJO Y EDICION DE CRISTALES.

Los cristales dentro de los escenarios únicamente cumplen una función un tanto decorativa para el ambiente en que se desenvuelve el jugador, con el fin de variar el lugar de un Juego, es importante no sobrecargar un lugar con esta clase de efectos porque tiende a tomarse molesto para la vista.

Los cristales pueden tener el tamaño que el desarrollador desee siendo concerniente a su criterio.

- Para crear un cristal es necesario definir que clase de Block Actor se va a utilizar.
- En este punto es importante aplicar texturas acordes y por lo general deben ser colores azulados, plateados o verdosos.
- Una vez aplicadas las texturas se debe cambiar al modo Brush Adjustment.
- Aquí se cambia la propiedad del Brush, se pasa de Solid a Window.
- Luego se cambia al modo Faces Adjustment y se accede a la caja de propiedades del Faces.
- Dentro de las opciones se señala Transparent y en Transparent Value se le da un valor entre 0 y 254, esto es a criterio del desarrollador.



8.2 MANEJO Y EDICION DE ESPEJOS.

Para cualquier persona un espejo es cualquier superficie que tiene la capacidad de reflejar la luz del medio, los objetos animados e inanimados que son reflejados tiene esa capacidad porque la luz incide sobre ellos, en el World Editor los espejos son superficies de cualquier Brush con la capacidad de reflejar la luz del entorno haciendo posibles que se vean reflejados los objetos del nivel.

- Para crear un espejo se debe escoger convenientemente una superficie plana o crear un Block Actor nuevo.
- Ya escogida la superficie que se va a utilizar como espejo se cambia el modo de trabajo a Face Adjustment.
- Ahora se tiene que acceder a la caja de Faces Attributes.
- Una vez visualizada se debe seleccionar la opción Mirror (Espejo).
- Después se selecciona la opción Transparent y en Transparent Value se le da un valor entre 0 y 255.

Igual que los cristales no se deben utilizar en exceso los espejos para lograr una armonía entre los elementos que conforman el nivel.



8.3 MANEJO Y EDICION DE LIQUIDOS.

Los líquidos son un efecto que se pueden encontrar en dentro de una habitación, pero también puede haber una habitación que sea completamente de agua, o un nivel mixto que tenga tanto agua como parte sólida, pero se debe tener en cuenta que este efecto de líquidos se usa más que todo para darle variedad a una habitación.

- Para crear un líquido se debe escoger o crear un Block Actor nuevo, aunque se debe tener en cuenta que esta debe tener un lugar en el cual se



líquido.

- Una vez escogido el Block Actor que se va a convertir en líquido se procede a cambiar el modo de trabajo a Brush Adjustment.
- Ahora se tiene que acceder a la caja de Brush Attributes.
- Una vez allí se debe seleccionar la opción de tipo a Empty, luego se debe chequear Wavy, y en Contents chequear la casilla Water.

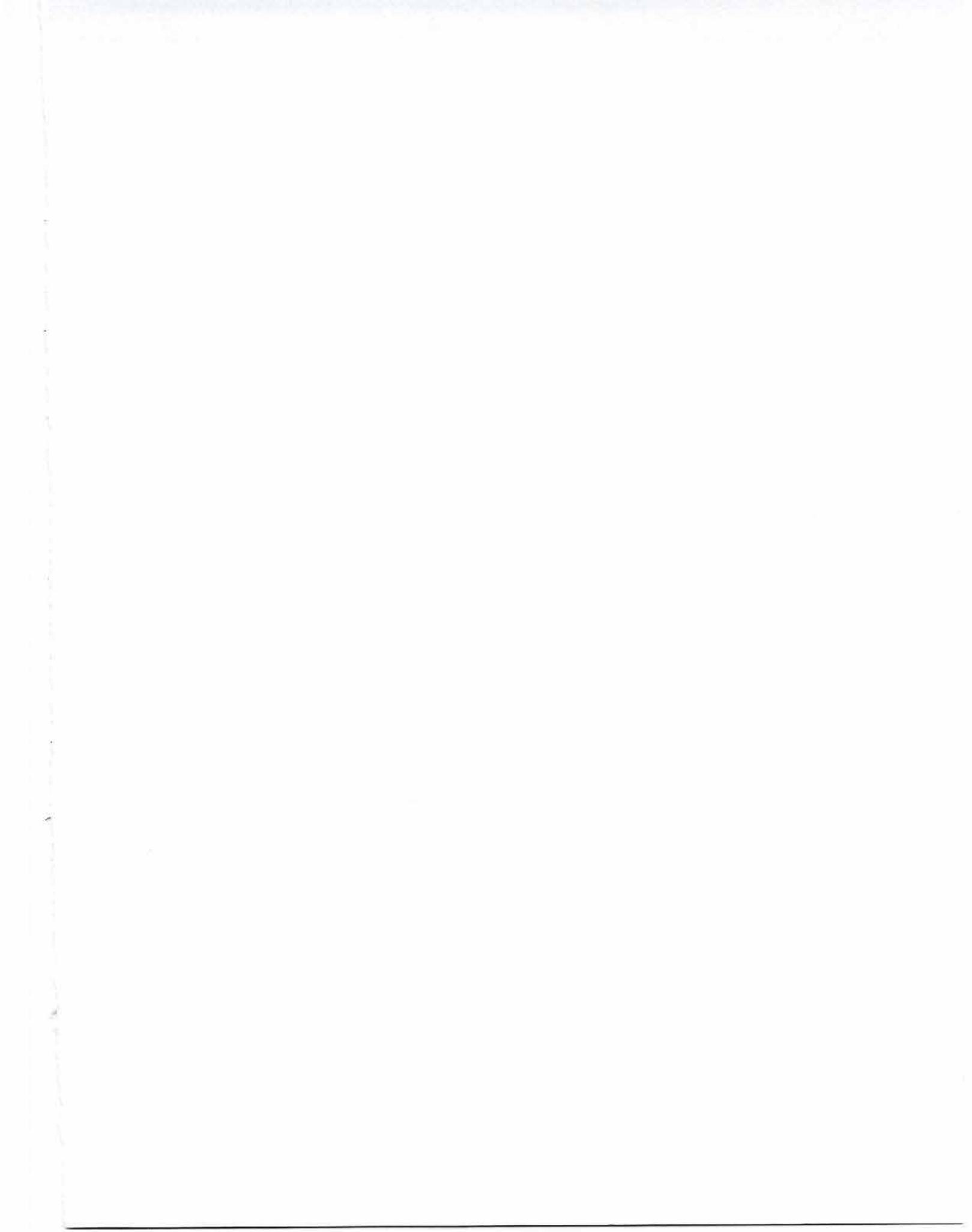


RESUMEN

En este capítulo se explicará todo lo concerniente a los efectos prácticamente decorativos de los escenarios como lo son cristales, espejos y líquidos; mediante estos distintos efectos se pueden embellecer los escenarios para que sean de gran calidad y atractivo al usuario.

TALLER

Utilizando World Editor crea una pileta de agua y un espejo en una habitación.



CAPITULO 9. ILUMINACION, NIEBLA, CIELOS, RAYOS.

9.1 MANEJO DE LA ILUMINACION EN LOS NIVELES.

El ojo humano es más sensible a los cambios de brillo que a los de color, por lo que una imagen con efectos de iluminación transmite más información al espectador de forma más eficaz. El motor discreto de iluminación de una GPU calcula los vectores de distancia entre las luces y los objetos y entre los objetos y los ojos del espectador dentro de las escenas 3D. Los cálculos de iluminación son un medio efectivo para modificar de forma sutil o radical el brillo de los objetos 3D para imitar las condiciones de iluminación del mundo real.

Los reflejos especulares se mueven por el objeto si el espectador o el objeto cambian de posición con respecto a la fuente de luz, lo que explica que no puedan calcularse anticipadamente ni ser estáticos. La iluminación especular es especialmente útil para lograr dos de los efectos de las escenas 3D: el desplazamiento y la apariencia de los distintos materiales de los objetos.

Iluminación difusa y especular

La iluminación se divide en dos categorías principales: difusa y especular. En la iluminación difusa, la luz que alcanza a un objeto se dispersa por igual en todas las direcciones, de forma que la luz reflejada no depende en absoluto de la posición del espectador. Por ejemplo, cuando el sol baña un terreno de juego, la luz llega a todas partes. La iluminación especular es distinta porque depende de la posición del espectador, la dirección de la luz y la orientación del triángulo en el que se refleja. Por ejemplo, el haz de luz de una linterna rebotará de forma distinta cuando se refleje en una moneda y en una brizna de hierba. La iluminación especular reproduce las propiedades reflectantes de un objeto y permite crear efectos como reflejos y resplandores.

Hasta ahora, a las técnicas de iluminación para gráficos de PC en tiempo real les faltaba un elemento clave: sombras realistas. Los diseñadores y fotógrafos siempre han sido conscientes del poder de la iluminación y las sombras. De hecho, se siguen utilizando películas en blanco y negro en la fotografía artística debido a los impactantes efectos de la iluminación y las sombras sobre las personas, incluso sin color. Estos mismos efectos se pueden utilizar para cautivar a los usuarios y transportarlos a entornos gráficos en tiempo real.

En el World Editor los Brushes que se trabajan para la iluminación son seis, aunque el Foglight se utiliza en la niebla, se explicaran los cinco que se utilizarán en la iluminación de cualquier nivel. Los Brushes que se utilizan en la iluminación son Light, Dynamic Light, Spotlight y Corona; cada uno tiene características diferentes.

Escenas complejas

El mapeado de sombras funciona bien en las escenas de gran complejidad arbitraria que contienen un gran número de objetos o formas complejas.

Bordes de sombra más uniformes

El mapeado de sombras puede suavizar los bordes de sombra mediante un algoritmo de filtro de sombras exclusivo.

Auto sombra

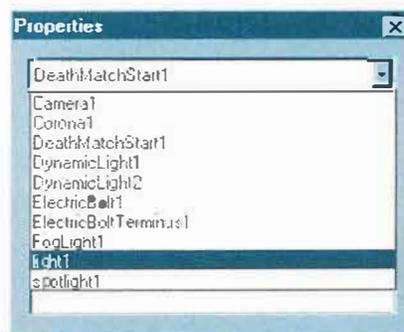
Los objetos pueden proyectar una sombra de si mismos. Observe cómo las alas de la nave espacial proyectan una sombra en otras partes de la nave.

Compatible con rendering multitextura

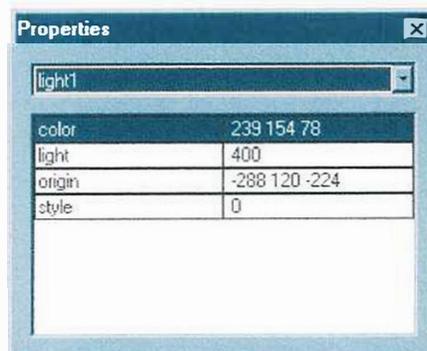
Los volúmenes de sombra con estencil pueden obligar al rendering de varias pasadas, lo que reduce el rendimiento si la aplicación tiene límites de velocidad de relleno o de ancho de banda de memoria.

Parámetros de las luces:

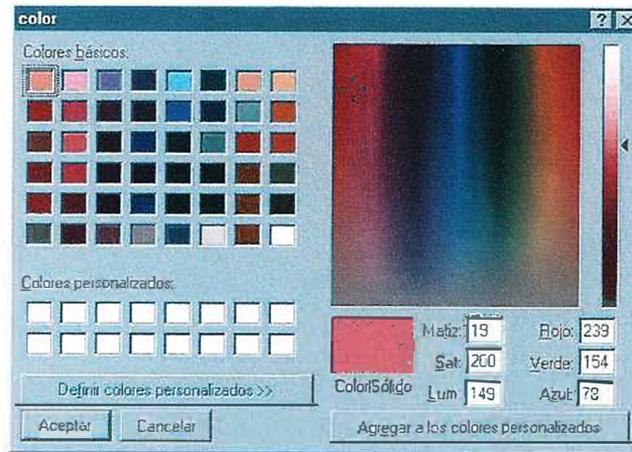
Cuando pulsamos el botón propiedades de las entidades  se abre la caja 'Properties', donde tendremos que seleccionar la entidad de luz que deseamos modificar sus parámetros:



Dependiendo del tipo de entidad que seleccionemos nos aparecerá su caja de propiedades correspondiente con una serie de parámetros que podremos modificar. Para empezar comenzaremos describiendo la luz más simple, la entidad llamada 'light':

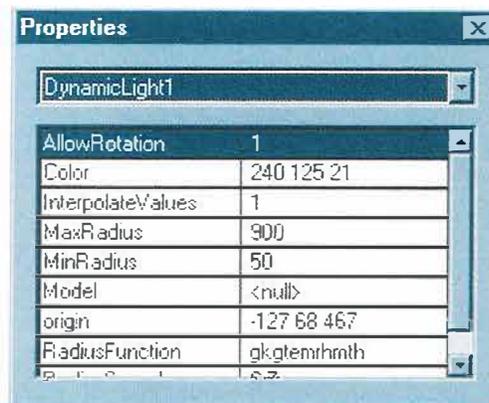


- 'color': Nos permite seleccionar el color que deseamos para la luz, esto lo realizaremos mediante un cuadro que nos presenta toda la gama de colores:



- 'light': Determinará la intensidad de la luz.
- 'origin': Fija la posición de la entidad, normalmente se establece con el propio editor situando la entidad con el ratón.
- 'style': Con 0 es una luz normal, mientras que con 1, 2 o 3 podremos hacer que la luz parpadee de distintas formas.

La entidad de luz 'dynamic light':

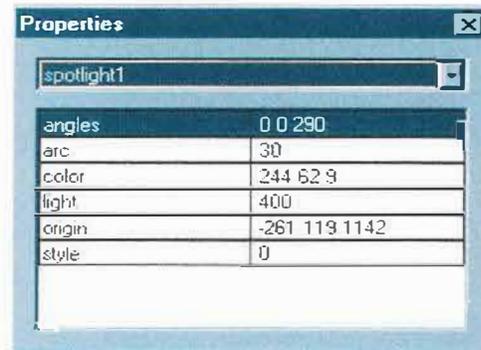


- 'AllowRotation': - Si la luz se asigna a un modelo, seguirá su movimiento si ponemos '1'.
- 'Color': - Selección del color de la luz.
- 'InterpolateValues': Se utiliza para producir suaves variaciones en la luz, pero si ponemos '0' las variaciones serán muy bruscas.
- 'MaxRadius': El máximo radio de acción de la luz.
- 'MinRadius': El mínimo radio de acción de la luz.
- 'Model': Modelo al que irá unida la luz.
- 'origin': Posición de la luz.
- 'RadiusFunction': Aquí entraremos una serie de letras que determinarán la

secuencia y niveles de variación de la luz. La letra 'a' es para muy oscuro, mientras que la letra 'z' es para muy brillante.

'RadiusSpeed': Velocidad en las variaciones de la luz.

La entidad de luz 'Spotlight':



'angles': Ángulos X Y y Z hacia donde se dirige la luz.

Es el radio que formaría un cono que representa la boca de salida del foco, si lo cerramos (menos ángulo) la luz se proyectará más pequeña y concentrada, mientras que si lo abrimos (más ángulo) la luz se proyectará más grande y expandida.

'arc':

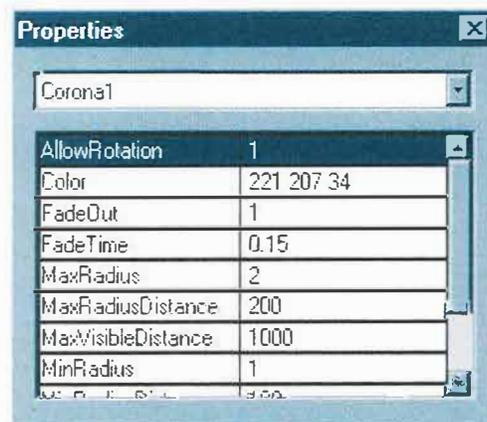
'color': Color de la luz.

'light': Intensidad de la luz.

'origin': Posición de la luz.

'style': Tipo de parpadeo de la luz (0, 1, 2 o 3).

Entidad de luz 'corona':



'AllowRotation':	Con '1' la luz sigue el movimiento del modelo al que esté asignada.
'Color':	Color de la luz.
'FadeOut':	Nivel de desvanecimiento de la luz.
'FadeTime':	Tiempo en segundos para desvanecerse.
'MaxRadius':	Máximo radio de la corona cuando se vea a la distancia de 'MaxRadiusDistance'.
'MaxVisibleDistance':	Distancia máxima a la que es visible la corona.
'MinRadius':	Mínimo radio de la corona cuando se vea a la distancia de 'MinRadiusDistance'.
'Model':	Modelo asociado a esta entidad.
'origin':	Posición de la corona.

Bien, pues hasta aquí las luces. En el siguiente apartado veremos otras entidades.

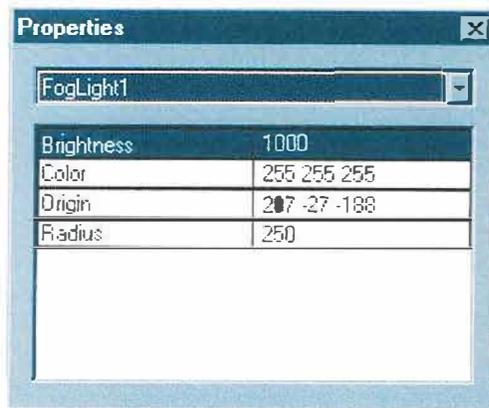
9.2 MANEJO DE LA NIEBLA EN LOS NIVELES.

Seleccionamos la entidad 'FogLight' de la pestaña 'Template' y añadimos esta entidad de la misma forma que siempre. Esto nos creará la entidad 'FogLight1' en nuestro mundo. La movemos hasta la zona donde queremos provocar el efecto de una niebla. Para las propiedades de esta entidad pondremos 1000 para 'Brightness' y 120 para 'Radius'. El color podría ser un blanco para una niebla normal, pero podrías asignarle otro color para crear efectos más exóticos.

El resultado es una zona que aparece envuelta en una niebla:



Entidad de luz tipo niebla 'FogLight':



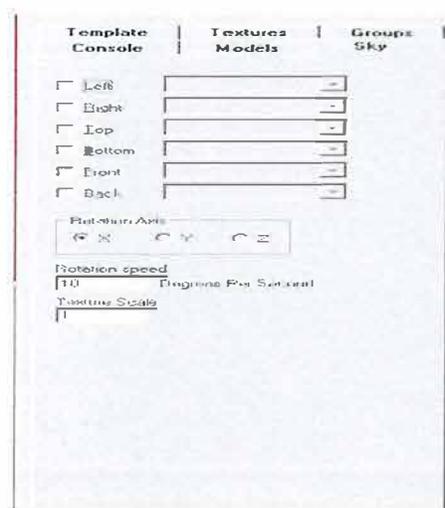
No abusar demasiado en el uso de esta entidad ya que requiere un tiempo de cálculo muy grande para el motor 3D.

- 'Brightness': Brillo que desprende la niebla.
- 'Color': Color de la niebla.
- 'Origin': Posición de la niebla.
- 'Radius': Radio de acción de la luz de la niebla (no debe ser muy grande).

9.3 MANEJO DE LOS CIELOS EN LOS NIVELES.

La pestaña 'Sky' es una herramienta que nos permite crear efectos de firmamento para nuestro mundo, como cielo pasando, estrellas o algo completamente diferente. Lo primero que debemos hacer es asignar las seis texturas que forman las seis caras de nuestro cielo virtual, esto lo haremos en la pestaña 'Sky' seleccionando seis texturas que normalmente son las mismas las seis.

Para que desde nuestro mundo se pueda ver este cielo debemos establecer que caras 'fáces' de nuestros 'brushes' que representarán que son aberturas al exterior de nuestro mundo. Para ello deberemos seleccionar la cara 'face' que nos interesa abrir al cielo y entrar en sus propiedades, allí marcaremos 'Sky':



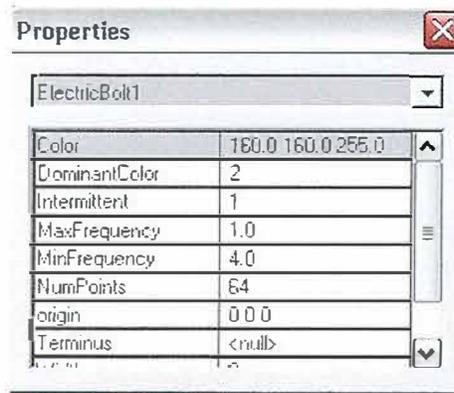
Podemos regular la velocidad del firmamento modificando el valor de 'Rotation Speed' en la pestaña 'Sky'.

Ya dentro de tu mundo, si observas estos cielos, podrás ver que no lo forma algo liso y uniforme, sino que está hecho como un cubo que va dando vueltas alrededor de tu mundo. Si para el cielo no has seleccionado seis texturas que casen perfectamente entre ellas, podrás observar que el efecto de cielo no es perfecto y las esquinas de ese cielo 'cúbico' se notan mucho, aparte de todo lo anterior para esta parte de los cielos se puede usar paisajes creados con terragen.

9.4 MANEJO DE LOS RAYOS EN LOS NIVELES.

Vamos a crear un rayo eléctrico que saldrá de un punto a otro de nuestro nivel, ambos puntos los determinaremos nosotros con dos entidades distintas.

Primero crearemos la entidad que fija el punto destino del rayo, para ello añadiremos la entidad 'ElectricBoltTerminus' a nuestro mundo y la colocaremos en el punto donde queremos que llegue el rayo. Luego añadiremos la entidad 'ElectricBolt' y la colocamos en la posición de donde queremos que salga el rayo. Todo esto nos habrá creado dos entidades llamadas 'ElectricBolt1' y 'ElectricBoltTerminus1'. Abrimos las propiedades del primero, 'ElectricBolt1', y en el parámetro 'Terminus' seleccionamos 'ElectricBoltTerminus1'.



RESUMEN

La iluminación es lo más difícil de crear en un Videojuego, pero crearla no es lo difícil sino dar las sombras a los diferentes objetos que necesitan de este efecto; la niebla es muy sencilla y claro se debe utilizar en todos los juegos por la simple razón que no todos lo llevan, además se debe tener en cuenta que la niebla es bastante útil para crear eventos o sorpresas que den un contenido secuencial a los Videojuegos; es bastante importante conocer que el cielo de cada nivel es distinto de otro, por tanto se debe tener sumo cuidado en la escogencia de este efecto en particular; en cuanto al aspecto de los rayos, este es como un efecto agregado que sirve para dar un toque de realismo al escenario, e implantar un concepto de autenticidad a los Videojuegos.

TALLER

Mediante el uso de World Editor apléale efectos de iluminación, niebla, rayos y cielos a una habitación.

CAPITULO 10. DISEÑO Y CREACION DE PERSONAJES.

Los personajes son una parte fundamental de los Videojuegos, ya que son los que permiten interactuar con los distintos escenarios y objetos que hay en ellos: armamento, curaciones, puertas, etc: de acuerdo al nivel de interacción que estos tengan durante el desarrollo de un Videojuego se pueden catalogar como:

→ Personajes Principales: este es el personaje que maneja el Videojugador, pueden ser uno o varios dependiendo del Videojuego, la historia y el desenvolvimiento de las acciones.

→ Personajes Secundarios: son los que ayudan o interactúan con el personaje que el Videojugador maneja.

→ Enemigos: son los que de cierta manera obstaculizarían el accionar del personaje protagonista.

10.1 DISEÑO DE PERSONAJES.

Este apartado corresponde mayormente a un trabajo artístico y por lo general el diseño se debe hacer en papel con la asesoría de un dibujante profesional que tenga experiencia en el diseño de figuras humanas, de esta manera nuestro juego ganara realismo. En esta etapa han de definirse el aspecto de todos los personajes que han de intervenir en el Videojuego: el fin de estas texturas es el de aplicárselas a las caras de cada objeto que se cree en Milkshape 3D.

10.2 CREACION DE PERSONAJES.

La creación de personajes es un proceso fundamental en el que se deben utilizar programas de edición gráfica como Gimp (freeware), Photoshop, fireworks (macromedia), etc. para la creación de las texturas que se aplicaran a cada objeto creado, por tanto se debe tratar cada elemento del personaje como el rostro, el cabello o las extremidades. Lo importante de todo esto es crear armonía en el conjunto. Al finalizar todo este proceso se debe crear un archivo contenedor con todas las imágenes que conforman el personaje en un fichero .act que es el formato del archivo para los personajes usados en Genesis 3D.

Lo Primero es conocer un poco como funciona Milkshape y su interfaz múltiple, es decir menús y las pestañas de creación y manipulación:



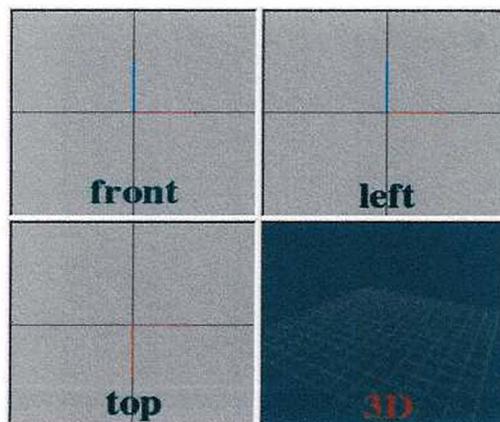
Elementos de la pantalla:

Menú:



Al igual que muchos programas en Windows, en Milkshape dispones de una barra de opciones que podrás usar para cargar proyectos, salvar proyectos, importar objetos, las opciones más importantes serán explicadas más adelante.

Vistas:



Existen cuatro vistas para trabajar sobre el modelo que estemos editando:

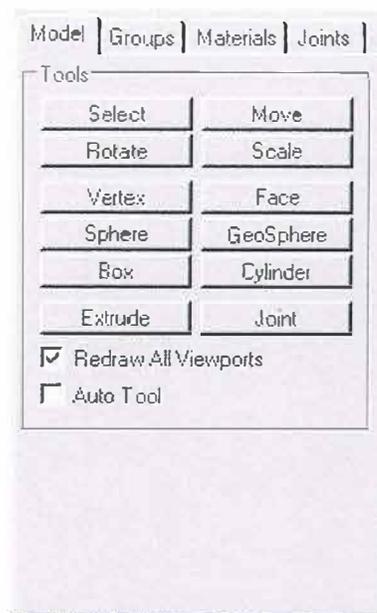
- Arriba a la izquierda tenemos la vista llamada '**Front**' o vista frontal.
- A su izquierda (derecha de la pantalla) la vista '**Left**' o vista lateral.
- Abajo a la izquierda tienes la vista '**Top**' o vista desde arriba.
- Abajo a la derecha está la vista '**3D**' donde siempre podremos ver una vista en tres dimensiones y desde cualquier ángulo del modelo.

En todas las vistas puedes mover la vista pulsando simultáneamente (**botón izquierdo del ratón + tecla 'Ctrl'**).

En todas las vistas puedes hacer zoom pulsando simultáneamente (**botón izquierdo del ratón + tecla 'Mayúsculas'**), excepto cuando estés en el modo 'Select' en las vistas 2D (front, left, top).

También puedes hacer zoom usando la rueda del ratón en el caso de que dispongas de ella.

Panel de Herramientas:



Mediante estas herramientas podrás crear formas geométricas, mover ciertas partes, animar el esqueleto de un personaje, aplicar texturas, en fin manipular todos los objetos en un sinnúmero de formas.



Deslizador de animación:



Permite ver las animaciones de un personaje si este las tiene.

A continuación se explica como crear un primer objeto utilizando MilkSape 3D:

Antes de seguir con los primeros pasos, deberías saber que Milkshape 3D utiliza como estructura básica los triángulos, los cuales están compuestos por la unión de tres puntos o vértices, y la unión de estos conforma todas las figuras que podrás ver y crear en MilkShape 3D, por lo que mediante triángulos y con cierta habilidad podrás crear cualquier cosa que te puedas imaginar.

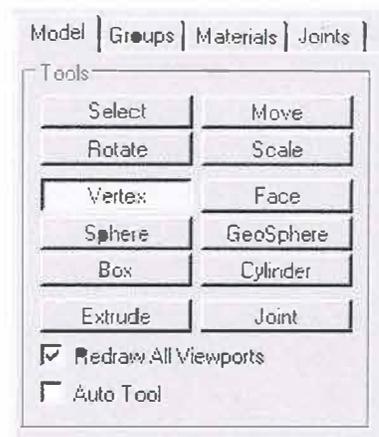
CREAR LA PRIMERA CARA

Vamos a comenzar con la creación de nuestro primer triángulo (en términos de modelaje se llama 'Face' o cara).

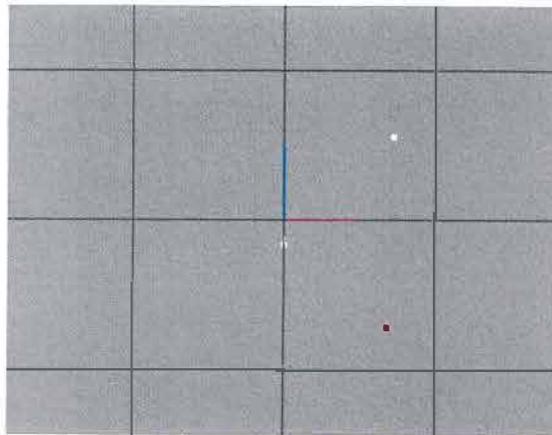
Para crear la primera cara (triángulo) deberás:

→ Crear los tres vértices (tres Vertex, Vértices en plural) que definirán el triángulo:

- En la pestaña 'Model' pulsamos el botón 'Vertex' del panel de Herramientas:

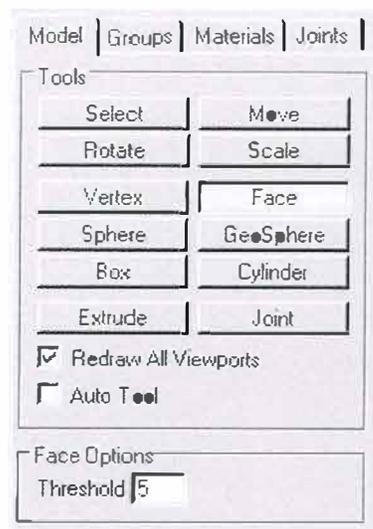


- Con el ratón, colocamos tres puntos en la vista frontal (**front**) simplemente pulsando con el ratón tres veces:

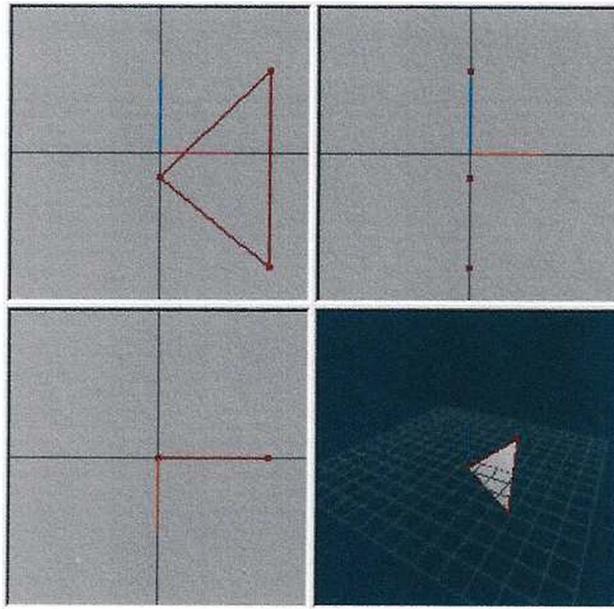


⇒ Combinarlos (unir los vértices) para crear la cara:

- Para crear una cara a partir de estos tres vértices, pulsamos el botón 'Face' del panel de herramientas:



- Seguidamente pulsamos en los tres vértices que hemos creado anteriormente. Observarás que tras pulsar el tercer vértice se formará un triángulo. ¡Ya has creado tu primera cara!



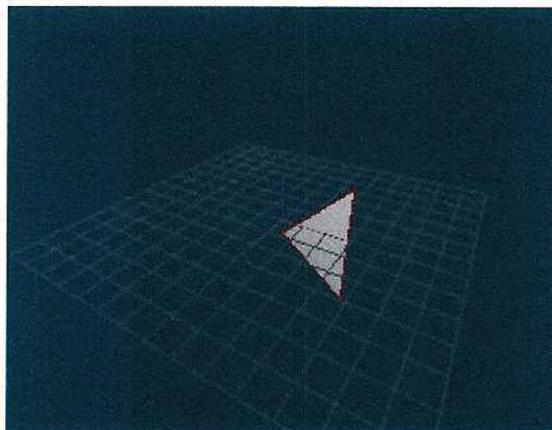
GUARDAR LOS AVANCES

A estas alturas sería conveniente que salvaras tu trabajo en disco, para ello utiliza la opción del menú *File Save As...*. Se recomienda que crees una carpeta por cada proyecto que realices, ya que dicho proyecto podrá constar de diversos ficheros (.DBY, .ACT, .MS3D, .BMP, ...) y si tenemos todos los ficheros necesarios para cada proyecto en su carpeta particular, más fácil nos resultará trabajar.

En MilkShape 3D los proyectos se salvan en formato .ms3d, siendo este un formato propio de MilkShape 3D.

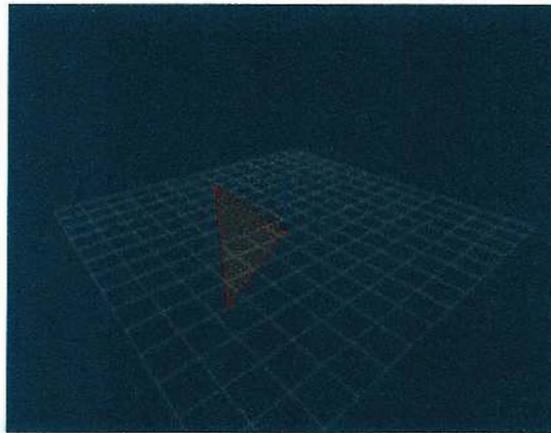
MANEJO DE LAS VISTAS

A continuación se explica exhaustivamente la forma en la que se manipulan las diferentes vistas que ofrece el entorno de trabajo de Milkshape 3D:



- Si mantenemos pulsado el botón izquierdo del ratón sobre la vista 3D y arrastramos, podremos ver nuestro triángulo desde cualquier ángulo, no es que estemos girando el triángulo sino que lo que hacemos es girar la cámara que enfoca a toda nuestra escena.
- En esta vista también podemos realizar un zoom con la cámara arrastrando con el ratón y al mismo tiempo pulsando la tecla 'Mayúsculas'.
- Otra cosa que puedes hacer en todas las vistas, incluida la vista 3D, es mover la cámara horizontalmente o verticalmente arrastrando con el ratón y pulsando al mismo tiempo la tecla 'Ctrl'.

LOS LADOS DE UNA CARA

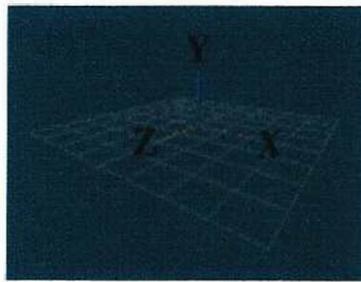


Si te has fijado, en la vista 3D, observarás que el triángulo se ve gris por un lado y negro por el otro, esto es muy importante ya que la cara gris indica que es la cara visible en el juego y por lo tanto a la que se podrá aplicar una textura (proceso que se realiza para pintar una cara). Por esta razón deja las caras siempre hacia afuera del objeto para no obtener resultados extraños. Más adelante veremos como girar las caras en caso de que no estén del lado que nos interesa.

EJES DE REFERENCIA

A modo de guía para la orientación de nuestra escena, en la vista 3D tenemos una rejilla cuadrículada (que representa el suelo) y los tres ejes básicos (X Y Z) de un mundo en tres dimensiones:





Es importante que te fijas en estos tres ejes cuando vayas a mover algún objeto, ya que te podrán servir de referencia a la hora de saber que cantidad de X, Y o Z has de utilizar para mover lo que quieras en la dirección que tienes en mente.

Si quieres hacer desaparecer tanto los ejes como la rejilla, lo podrás realizar pulsando el botón derecho del ratón sobre la vista 3D. Se te desplegará un menú flotante, donde podrás activar o desactivar las opciones 'Show Axis' y 'Show Grid'.

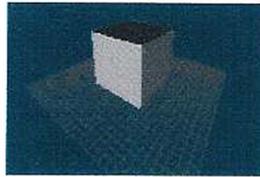


Otras opciones interesantes de este menú flotante son:

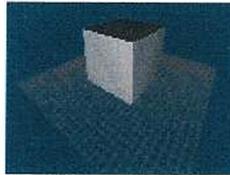
Wireframe: Permite ver nuestro modelo alambrado:



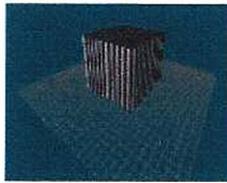
Flat Shaded: Ver el modelo con las caras grís y negro. Este es el modo recomendable para trabajar a nivel de edición de triángulos:



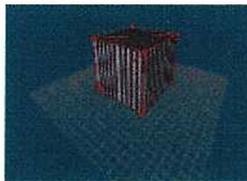
Smooth Shaded: Igual que la vista anterior pero con unos acabados más suavizados. Este modo invierte más recursos de CPU para presentar la vista 3D:



Textured: Permite ver nuestro modelo en su acabado final con todas las texturas aplicadas.



Wireframe Overlay: Superpone el alambrado del modelo.



Draw Backfaces: En los modos 'Shaded' las caras invisibles (las negras) no se visualizan.

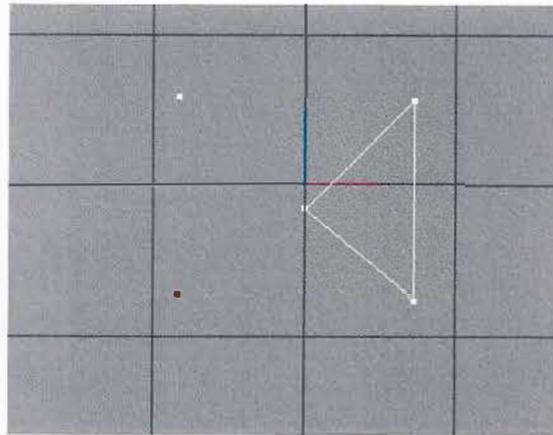
Take Screenshot: Permite realizar una captura de la vista 3D y guardarla en un fichero gráfico .BMP.

AÑADIR MAS CARAS

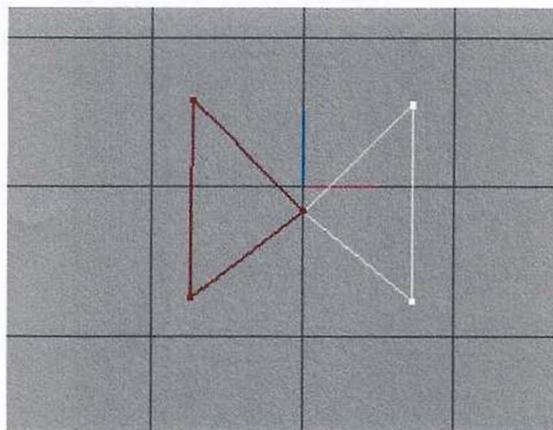
Ahora vamos a añadir más caras a lo que ya tenemos creado hasta ahora, con la intención de hacer una especie de pirámide.

Para ello necesitaremos 3 caras más. Comenzamos realizando la cara opuesta a la que ya tenemos creada, en este caso no vamos a necesitar crear tres vértices más, con sólo crear 2 más será suficiente, el tercer vértice lo cogemos de la cara que ya tenemos creada.

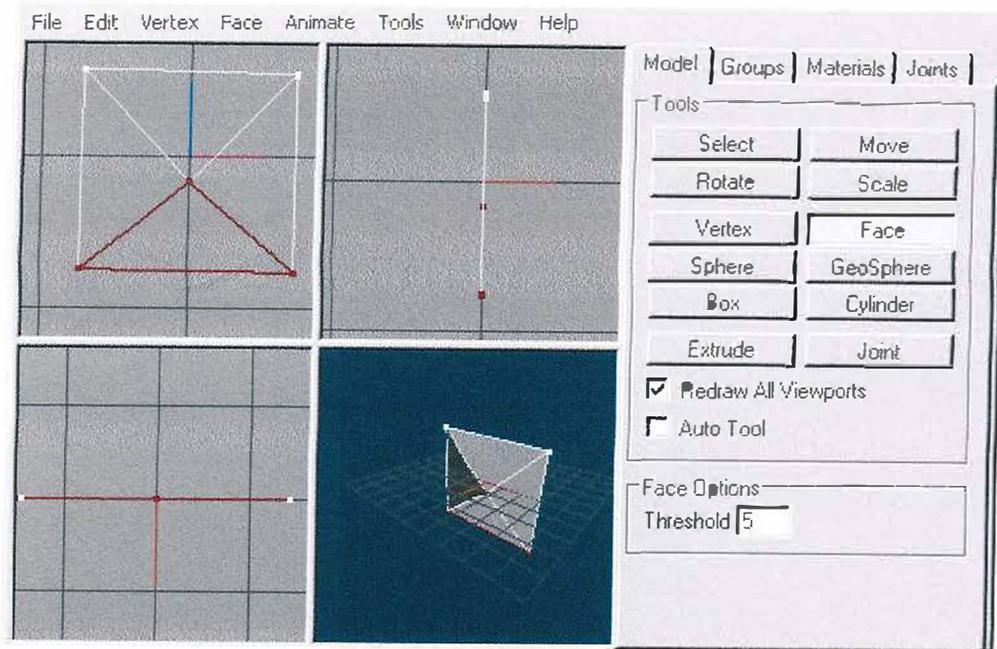
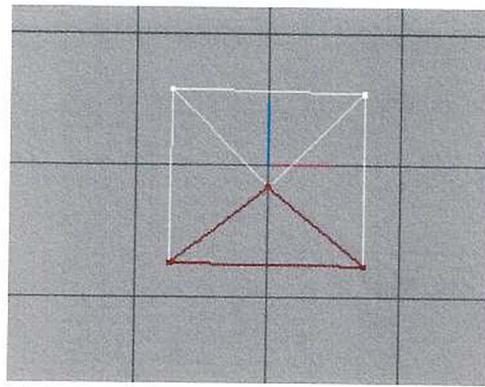
Ya sabes como proceder, seleccionas 'Vertex' en el panel de herramientas y pones dos puntos más en la vista frontal:



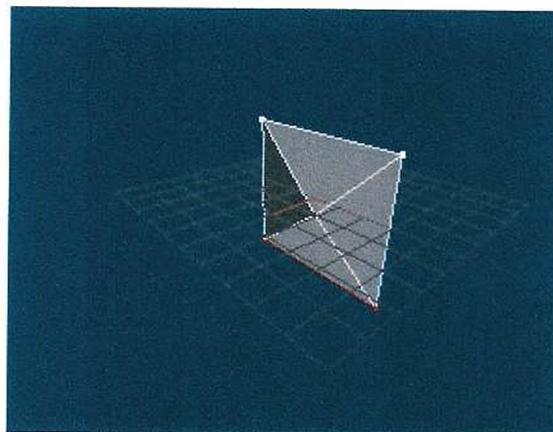
Ahora vamos a crear la cara. Selecciona 'Face' en el panel de herramientas y en la vista frontal pulsa sobre los dos vértices que acabamos de crear y también sobre el vértice de la anterior cara (el que queda en el centro), de manera que se cree un segundo triángulo:



Para crear las otras dos caras (para tener en total 4 caras de una pirámide) no necesitamos crear ningún vértice más, debes tener aún el modo 'Face' seleccionado en el panel de herramientas, seleccionamos los tres vértices superiores y a continuación los tres vértices inferiores. Si lo has hecho bien tendrás algo como esto:

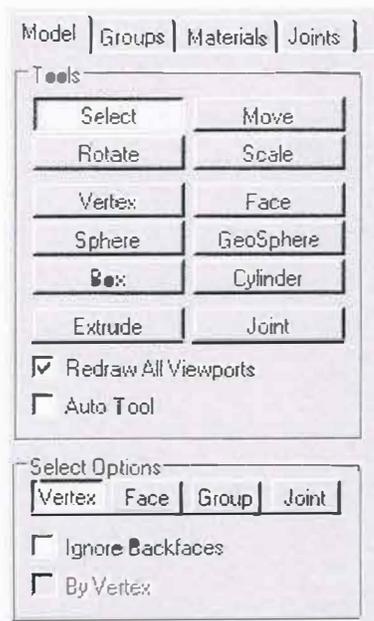


Mueve un poco la cámara en la vista 3D para ver como está quedando nuestra pirámide:

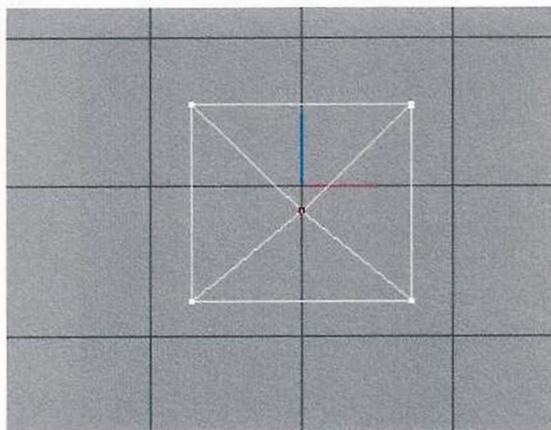


Para ser una pirámide nos ha quedado un poco plana, es normal, no hemos hecho otra cosa que poner triángulos planos en la vista frontal. A continuación vamos a elevarla utilizando para ello el vértice central de la pirámide.

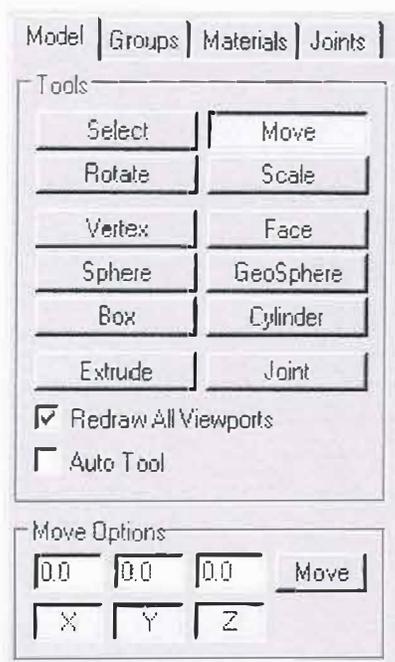
Para realizar manipulaciones con vértices lo primero que debemos hacer es precisamente seleccionar el vértice o vértices que queremos manipular. Para ello pulsamos en el botón 'Select' de la pestaña 'Model' del panel de herramientas. Más abajo verás una sección llamada 'Select Options', aquí pulsamos en 'Vertex' y desmarcamos 'Ignore Backfaces' si estaba marcado. Esta última opción se utiliza para indicar al programa si también se debe seleccionar o no vértices que estén tras el vértice que estemos seleccionando.



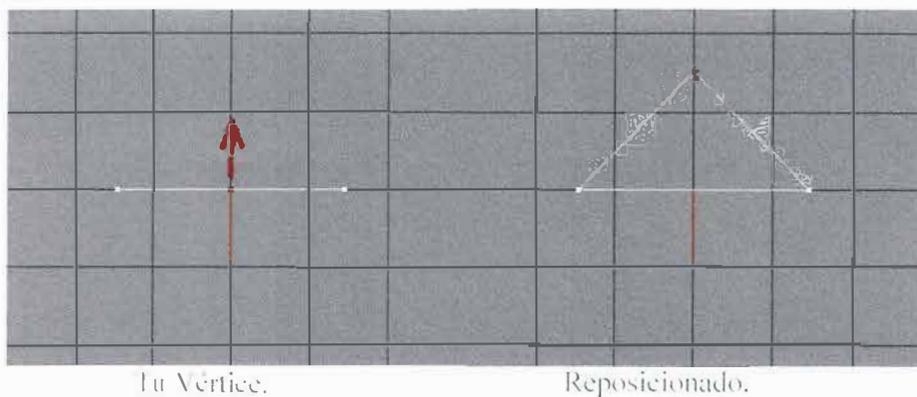
En la vista frontal pulsamos en el vértice central, de esta forma este vértice quedará seleccionado:



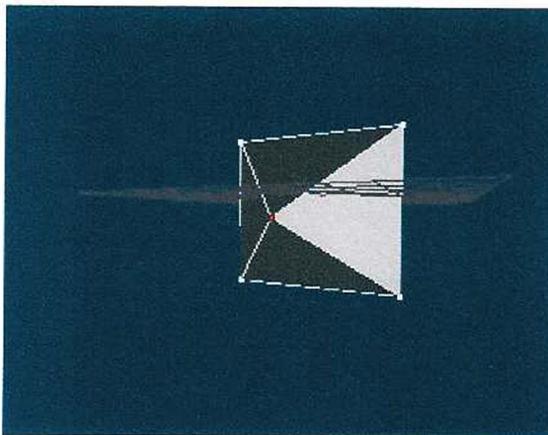
Seguidamente en el panel de herramientas pulsamos sobre el botón 'Move':



Ahora vamos a utilizar la vista desde arriba 'Top' (la que tenemos bajo la vista frontal) y en esa vista arrastramos hacia arriba el vértice que hemos seleccionado hasta que la cosa tenga un aspecto más o menos parecido a una pirámide:



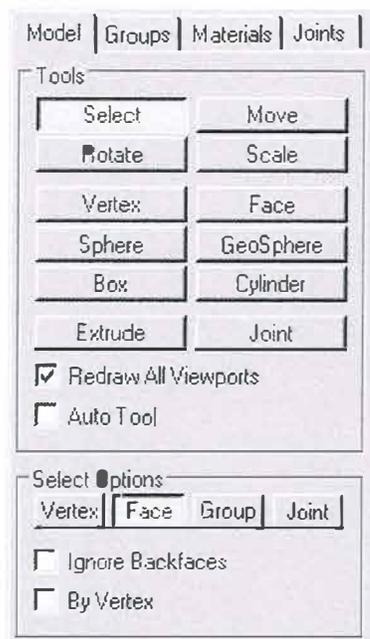
En la vista 3D tendremos algo como esto:



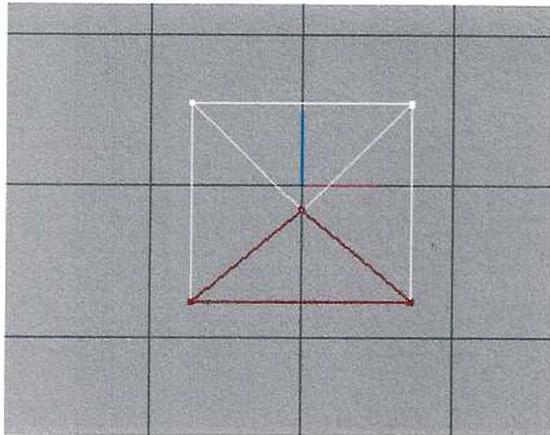
Mueve la cámara en la vista 3D y observa si hay caras negras en el exterior de la pirámide, como es el caso de este ejemplo, que tiene 3 caras negras que dan al exterior.

En principio no nos interesa que haya caras negras (invisibles y que no les afectarán las texturas) en la parte exterior de la pirámide, por lo que vamos a arreglar la cara o caras que estén de esta forma.

Pues es muy parecido a seleccionar vértices pero en el panel de herramientas, en lugar de pulsar los botones 'Select' y 'Vertex' pulsamos 'Select' y 'Face', no olvides desmarcar 'Ignore Backfaces' si ves que no puedes seleccionar las caras que te interesan:



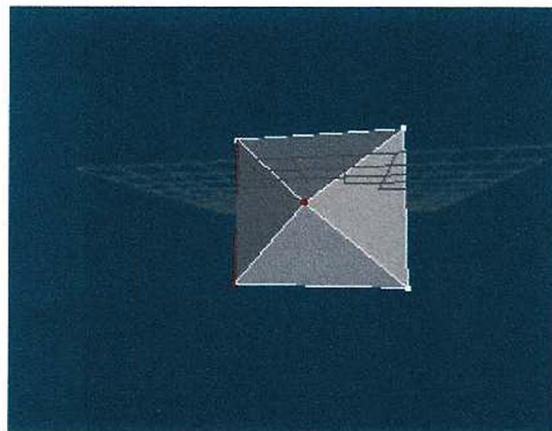
Pues para seleccionar una cara en cuestión sólo tenemos que pulsar en medio de la cara que nos interesa en alguna de las vistas 2D (no en la vista 3D):



Y ahora viene la opción que hará girar la cara para invertir el lado. Utiliza la opción del menú 'Face Reverse Vertex Order':



Después de ir seleccionando las caras e ir girándolas, este sería el aspecto final:



Recuerda trabajar en modo 'Flat Shaded' en la vista 3D para apreciar bien la diferencia entre caras negras y grises.

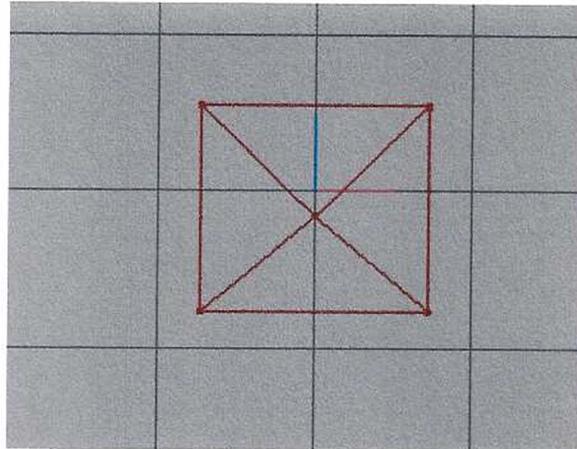
Puedes ir añadiendo caras a la selección, pulsando simultáneamente la tecla 'Mayúsculas' y pulsando con el ratón la cara a añadir.

ROTAR UNA CARA

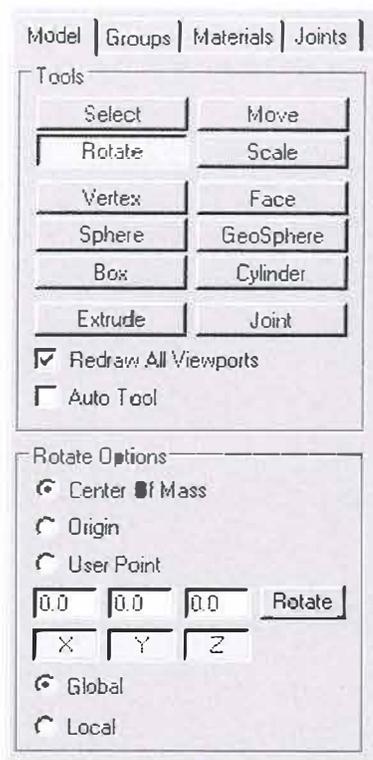
Como ves en la vista 3D, la pirámide está tumbada con respecto al plano que representa el suelo (la rejilla cuadrículada), a continuación vamos a poner la base de la pirámide plana con respecto al suelo.



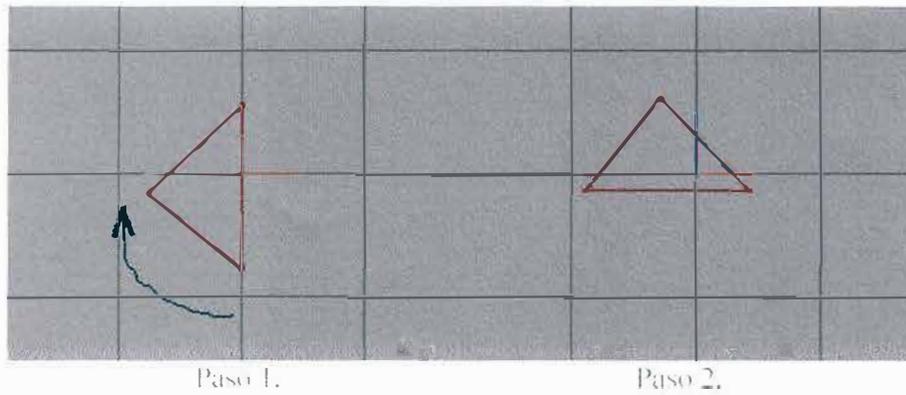
Para esto vamos a tener que seleccionar todas las caras de la pirámide:



Una vez tenemos todas las caras de la pirámide seleccionadas, pulsamos sobre el botón 'Rotate' del panel de herramientas.

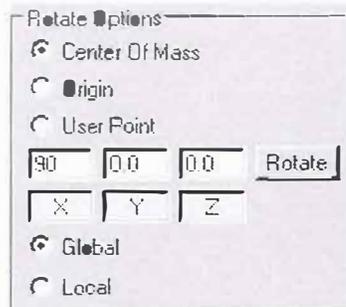


En estos momentos hay dos maneras de rotar todo lo que está seleccionado: Nos dirigimos a la vista lateral (left) y allí arrastramos con el ratón, verás que la pirámide gira, colócala como hemos comentado, la base de la pirámide plana:

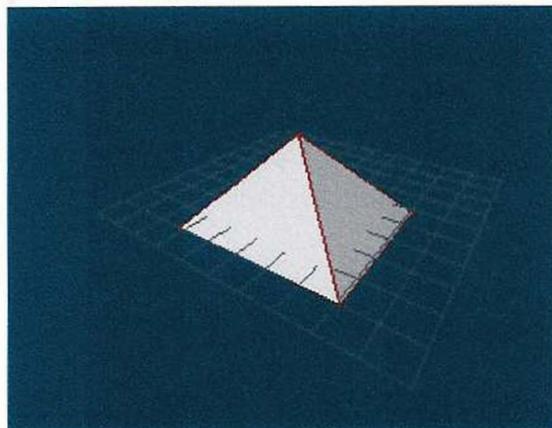


La otra manera de girar lo seleccionado es mediante las cajas X Y Z que te aparecen en el panel de herramientas cuando pulsas 'Rotate'. Rellenando estas cajas con los valores en grados que queremos aplicar a la rotación y pulsando en el botón 'Rotate' que hay a la derecha de estas cajas, conseguiremos girar lo que queramos de forma más precisa que si lo hacemos arrastrando con el ratón.

Para nuestro caso lo que necesitamos es girar la pirámide 90 grados de X, por lo que ese es el valor que introducimos en la caja X, para la Z y para la Y lo dejemos a 0. También seleccionamos 'Center Of Mass' para que la rotación se produzca desde el centro interno de la selección:



Después de girar la pirámide este es el aspecto que ofrece:



Para seleccionar también puedes hacerlo con una región delimitada por un cuadro que te aparece cuando arrastras con el ratón en alguna de las vistas 2D (no en la vista 3D) siempre que estés en el modo 'Select'.

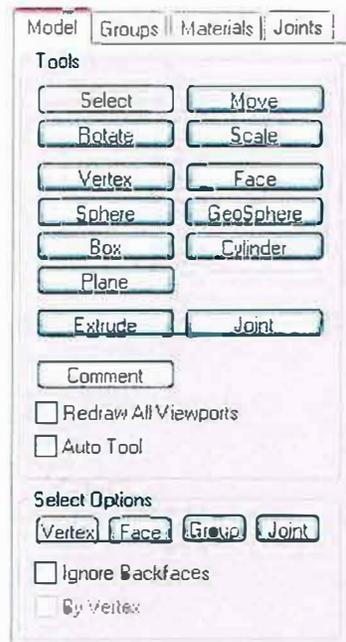
Para nuestro ejemplo también podríamos haber seleccionado todas las caras simplemente con la opción del menú 'Edit Select All'.

Si te equivocas y quieres volver a comenzar la selección, primero deseleccionalo todo pulsando ('Ctrl' + 'Mays' + 'A').

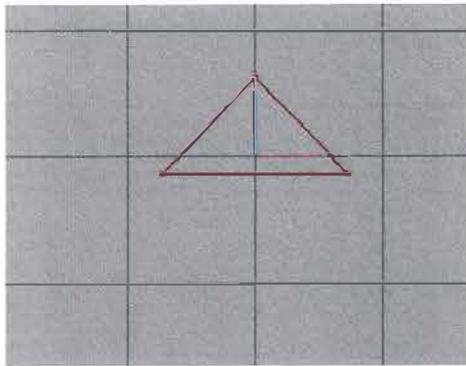
MOVER UNA SELECCION

Si quieres mover la selección que has realizado, sólo tienes que pulsar el botón 'Move' del panel de herramientas y posteriormente arrastrar en cualquier vista 2D para desplazar todo lo seleccionado:

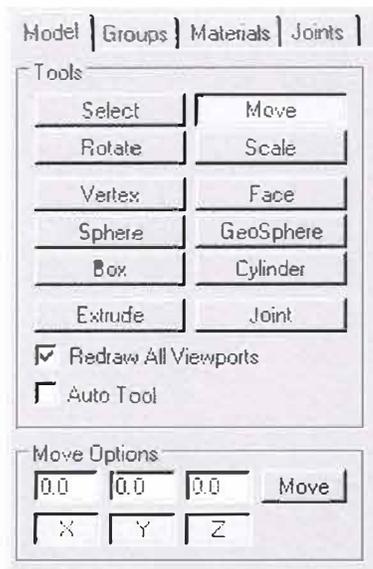
- Para hacer una selección debes tener previamente seleccionado la opción 'Select' y abajo la opción 'Vertex' en el Panel de Herramientas.



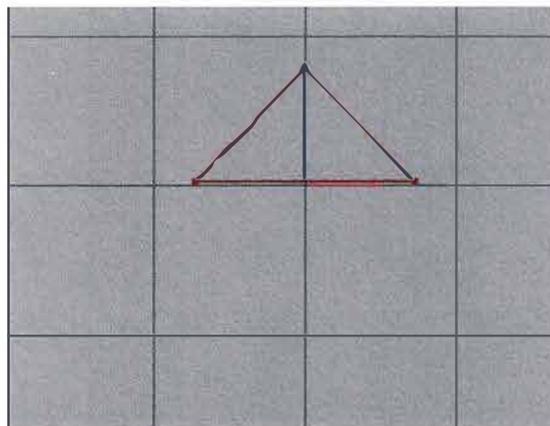
- Luego selecciona cualquiera de las vistas, menos la vista 3D que no permite seleccionar ni mover ningún vértice.
- Una vez seleccionada la vista mantén presionando Shift si deseas seleccionar varios vértices, de lo contrario solo podrás seleccionar un vértice.



- Ahora selecciona 'Move' en el Panel de Herramientas.



- Y mueve la selección que hiciste a tu antojo.

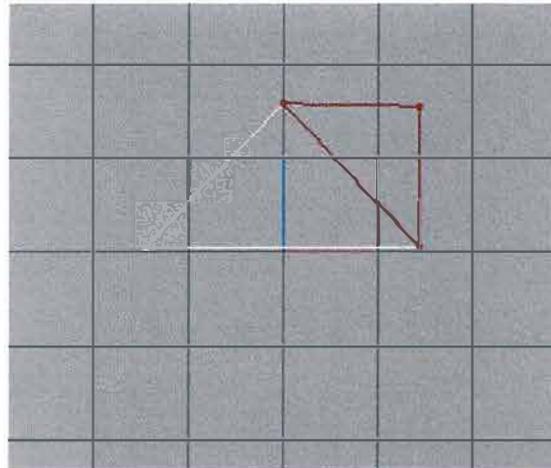


Recuerda que en panel de herramientas (sección 'Move Options') dispones de las cajas X Y y Z para realizar movimientos más precisos hacia cualquier dirección.

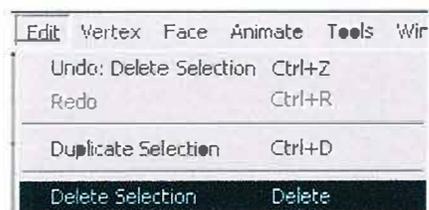
Si has movido algo y quieres dejarlo donde estaba, utiliza ('Ctrl' + 'Z'), esto sirve para deshacer cualquier acción que hayas realizado.

BORRAR UNA SELECCION

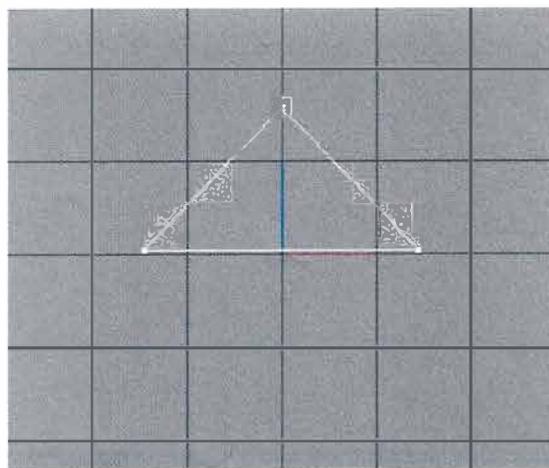
Para borrar caras o vértices sólo tienes que seleccionarlos y usar la opción del menú 'Edit Delete Selection', también puedes usar directamente la tecla 'Suprimir'.



Mediante la pestaña 'Edit' en la opción 'Delete Selection'.



Después de Borrar la selección.



Ten en cuenta que si eliminas vértices también eliminarás todas las caras que utilicen los vértices eliminados.

Si quieres seleccionar caras que pertenezcan a un vértice en concreto, en la sección 'Select Options' del panel de herramientas, marca la opción 'By Vertex' y luego selecciona el vértice o vértices que te interesen.

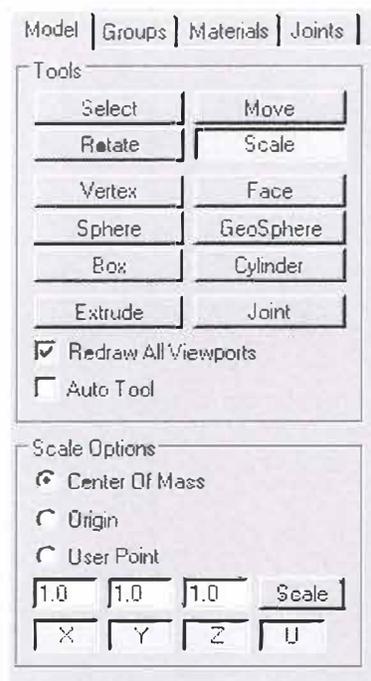
Si te equivocas y quieres volver a comenzar la selección, recuerda primero deseleccionarlo todo pulsando ('Ctrl' + 'Mays' + 'A').

Utiliza ('Ctrl' + 'Z') para recuperar lo que acabes de borrar.

REDIMENSIONAR

Para modificar el tamaño de una o varias caras sólo tienes que seleccionarlas, pulsar el botón 'Scale' del panel de herramientas y arrastrar el ratón en la vista que quieras realizar el cambio de tamaño.

- Lo primero que debes hacer es seleccionar la opción 'Scale' en el Panel de Herramientas.



- Y con el mouse modificar el tamaño con el movimiento del mouse.

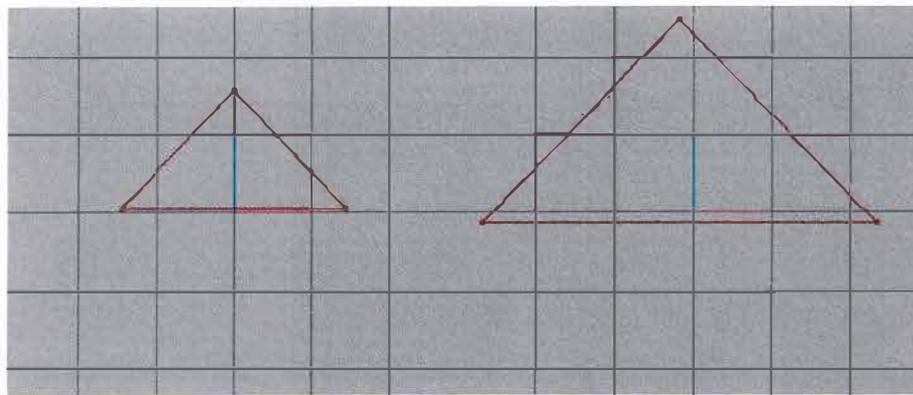


Figura Original.

Redimensionada.

Si lo deseas puedes realizar redimensionados más precisos utilizando las cajas X Y Z que te aparecen en el panel de herramientas al pulsar 'Scale'. Aquí los valores indican la proporción que deseas aumentar o reducir. Por ejemplo, para aumentar al doble de tamaño tendrás que entrar el valor 2,0, para el triple 3,0 y para la mitad 0,5.

Si estas redimensionando arrastrando con el ratón en una de las vistas y quieres que el redimensionado se realice proporcionalmente en las tres direcciones, pulsa el botón 'U' que encontrarás debajo del botón 'Scale' antes de redimensionar.

Si te equivocas y quieres volver a comenzar la selección, recuerda primero deseleccionarlo todo pulsando (Ctrl + 'Mays' + 'A').

Utiliza (Ctrl + 'Z') para volver a dejar el tamaño original.

DUPLICAR UN OBJETO

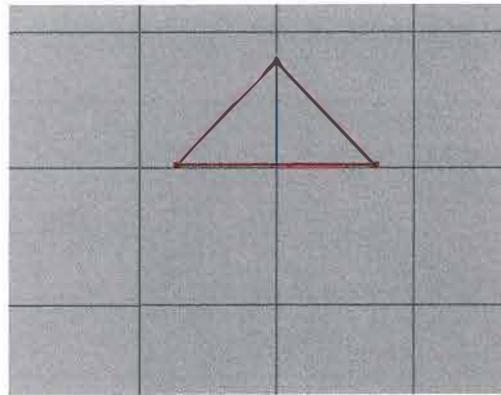
En este apartado vamos a crear una copia de la pirámide para colocarla debajo de la que ya tenemos, de tal forma que nos quede una especie de diamante.

- Seleccionamos toda la pirámide utilizando la opción de menú 'Edit Duplicate Selection':

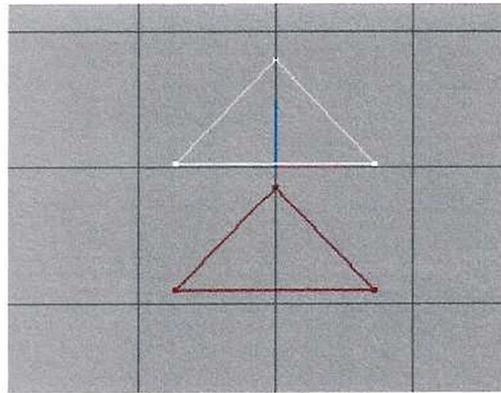


- Una vez hayas utilizado la opción 'Duplicate Selection' te aparecerá todo lo seleccionado duplicado, pero lo que se ha creado nuevo es lo que ahora está seleccionado. Selecciona 'Move' en el panel de herramientas y muévelo a cualquier sitio para poder verlo:





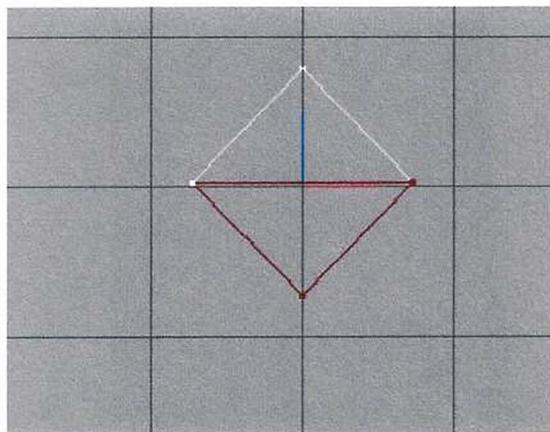
Una pirámide sobre otra.



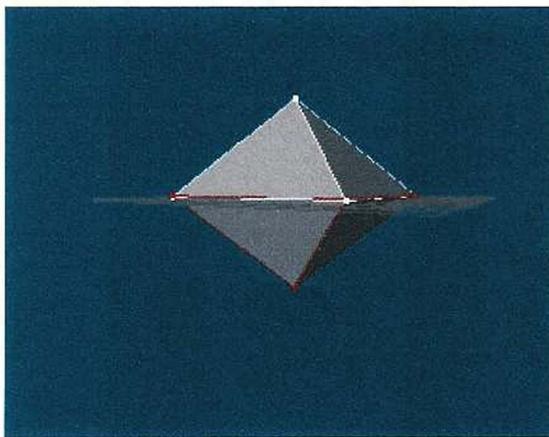
Pirámide desplazada.

- Utiliza las herramientas de rotar ('Rotate') y mover ('Move') para colocar la nueva pirámide invertida debajo de la que ya tenemos.

En este caso debes girar la pirámide nueva (copia) 180 grados en el eje X.



Resultado en la vista lateral.



Resultado en la vista 3D.

También puedes utilizar (Ctrl + D) para realizar la duplicación de lo que tengas seleccionado.

Cada vez que dupliques algo aparecerá un nuevo grupo que agrupará los nuevos triángulos. Puedes ver los grupos que tienes en tu proyecto en la pestaña 'Groups' del panel de herramientas.

MANIPULANDO GRUPOS

Cada vez que creas un objeto en Milshape 3D este se encarga de crear un grupo con los elementos de ese objeto y tu puedes manipular dichos grupos a tu conveniencia.

Si pulsas la pestaña 'Groups' del panel de herramientas, observarás que ahora tienes 2 grupos ('Triangles' y 'Duplicate01'), el primero agrupa los triángulos de la primera pirámide y el otro los de la segunda pirámide:

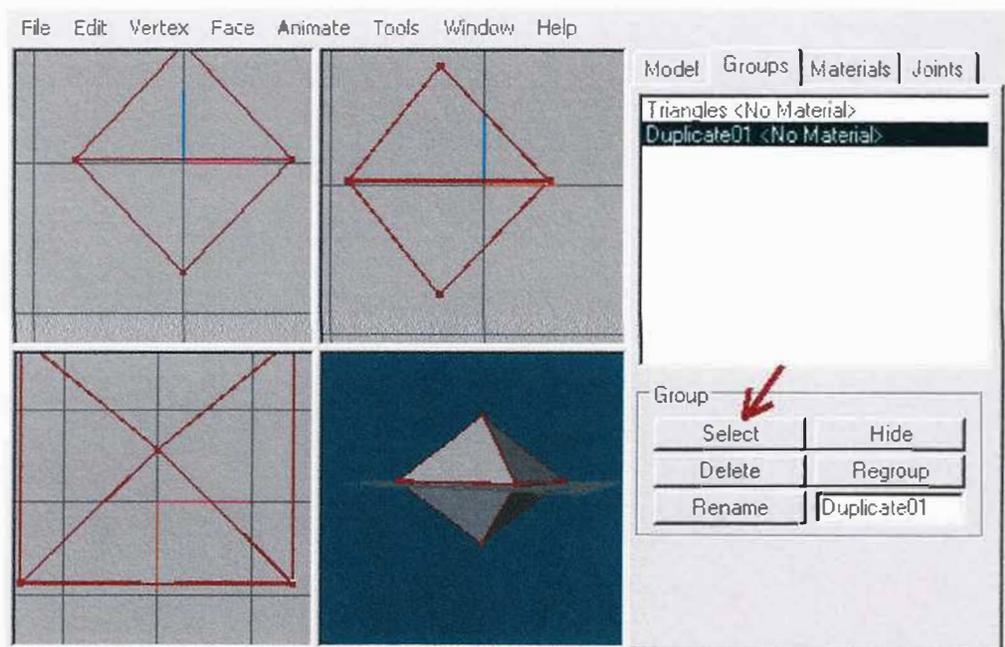


En esta pestaña tienes la opción de:

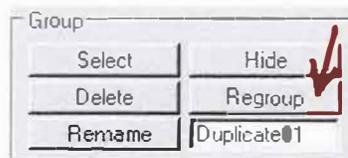
- ✓ Cambiar los nombres de los grupos utilizando el botón 'Rename'.
- ✓ Seleccionar/Desseleccionar todo un grupo utilizando el botón 'Select'.
- ✓ Ocultar/Desocultar todo un grupo utilizando el botón 'Hide'.
- ✓ Borrar un grupo con todo lo que al él esté asociado utilizando el botón 'Delete'.
- ✓ Agrupar dos o más grupos en uno solo utilizando el botón 'Regroup'.

Vamos a reagrupar las dos pirámides en un solo grupo para de esta forma hacer que sea un solo objeto.

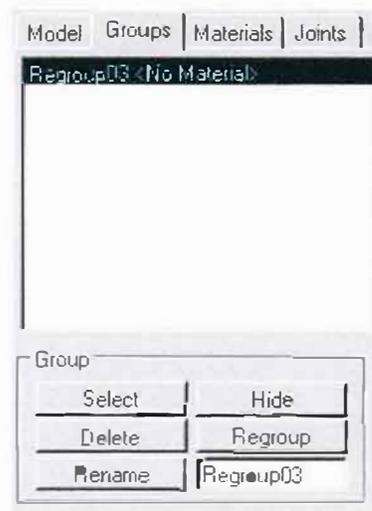
- Seleccionamos el nombre del primer grupo y pulsamos 'Select'
- Luego seleccionamos el nombre del segundo grupo y pulsamos otra vez 'Select', quedará todo seleccionado:



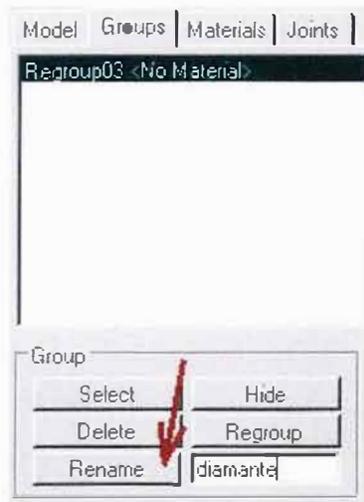
- Pulsamos 'Regroup':



- Ahora volveremos a tener un solo grupo con todos triángulos que hemos creado hasta ahora.



- Finalmente le cambiamos el nombre a la agrupación. Introducimos 'diamante' donde ponía 'Regroup03' en la caja que hay a la derecha del botón 'Rename' y pulsamos 'Rename':



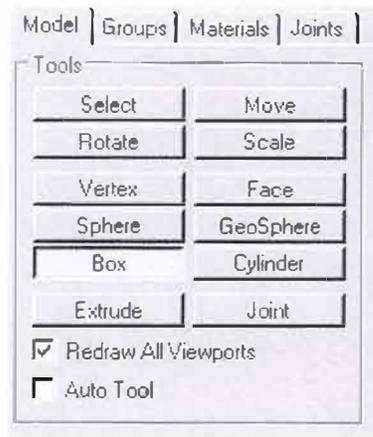
En ocasiones puede que no te interese reagrupar objetos y quieras dejarlos independientes para manejarlos por separado y aplicarles diferentes texturas ya que a la hora de aplicar texturas sólo se aplica una textura por grupo, por lo que tú verás lo que reagrupas o no en función de las necesidades de tu proyecto.

AÑADIR OBJETOS BÁSICOS

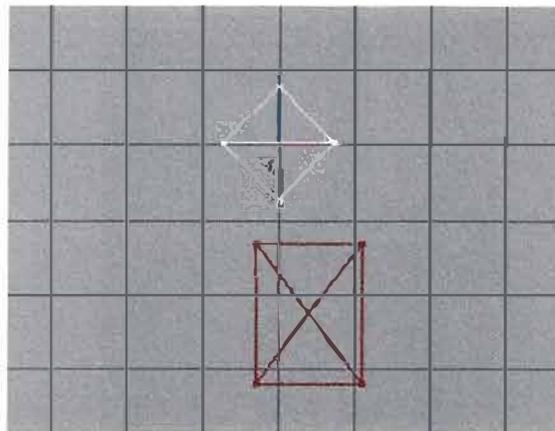
En MilkShape 3D aparte de crear objetos a base de añadir triángulos, podemos utilizar el panel de herramientas para añadir cubos, esferas y cilindros de una manera muy fácil.

Para nuestro ejemplo añadiremos un cubo que realizará las funciones de pedestal para nuestro diamante.

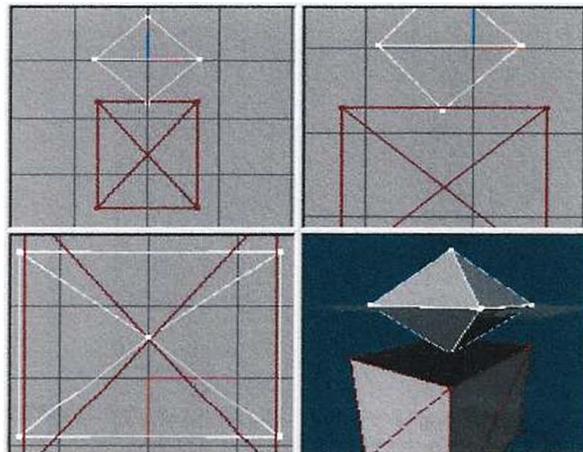
- En la pestaña 'Model' del panel de herramientas pulsamos el botón 'Box'.



- A continuación, en la vista frontal, pulsamos con el ratón y sin soltar arrastramos, observaremos que aparece un cubo que podemos agrandar y reducir a voluntad hasta que solemos el botón del ratón:



- Mueve el cubo en las vistas que necesites para que quede más o menos centrado debajo del diamante.



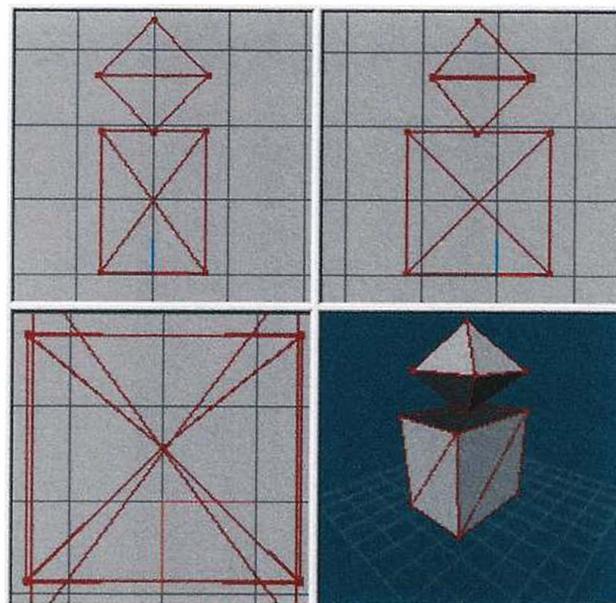
- De nuevo tendremos dos grupos, el diamante y la caja que acabamos de crear:



- Cambiemos en nombre del grupo que se acaba de crear por el de 'pedestal'.



- Lo seleccionamos todo para moverlo y dejarlo por encima del suelo:



Esto último es importante para que los objetos aparezcan correctamente en Entidad 3D ya que a la hora de situar un objeto en World Editor, la cruz que lo representa se considera la base del objeto.

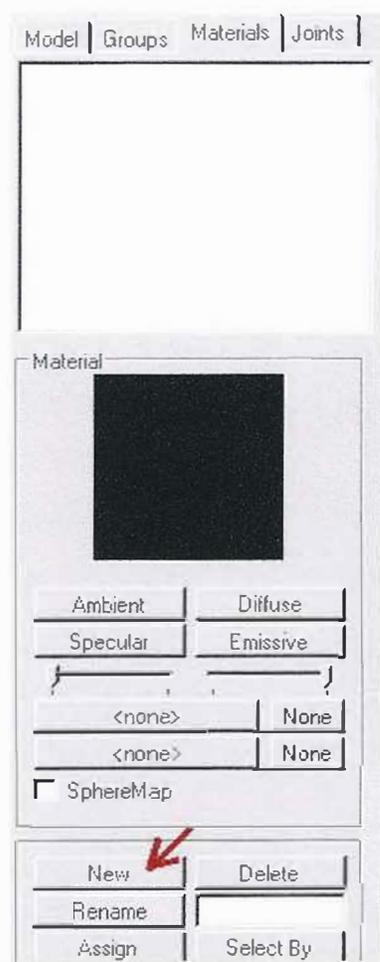
A parte del cubo también podemos añadir de la misma manera esferas (botones 'Sphere' y 'GeoSphere') o cilindros (botón 'Cylinder').

Para estos objetos no es necesario realizar ningún proceso con la herramienta 'Texture Coordinate Editor' que se comenta en el capítulo 'Aplicar Texturas'.

APLICAR TEXTURAS

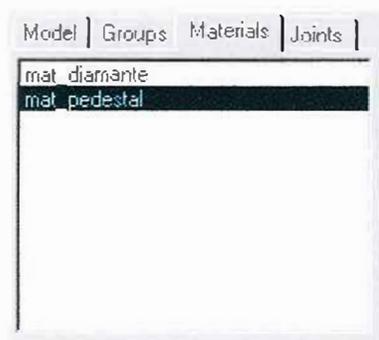
A continuación vamos a proceder a aplicar texturas a los dos grupos que tenemos definidos, al diamante y al pedestal.

- Primero vamos a crear los dos materiales que utilizaremos.
- Pulsamos sobre la pestaña 'Materials' del panel de herramientas, nos dirigimos a la parte de abajo, donde hay una caja al lado del botón 'Rename', pulsamos 'New', nos aparecerá el nombre 'Material01' en la caja:



Le cambiamos ese nombre por el que queramos darle como nombre de material al diamante, por ejemplo 'mat_diamante'. Para hacer esto sólo tenemos que cambiarlo en la caja y pulsar 'Rename'.

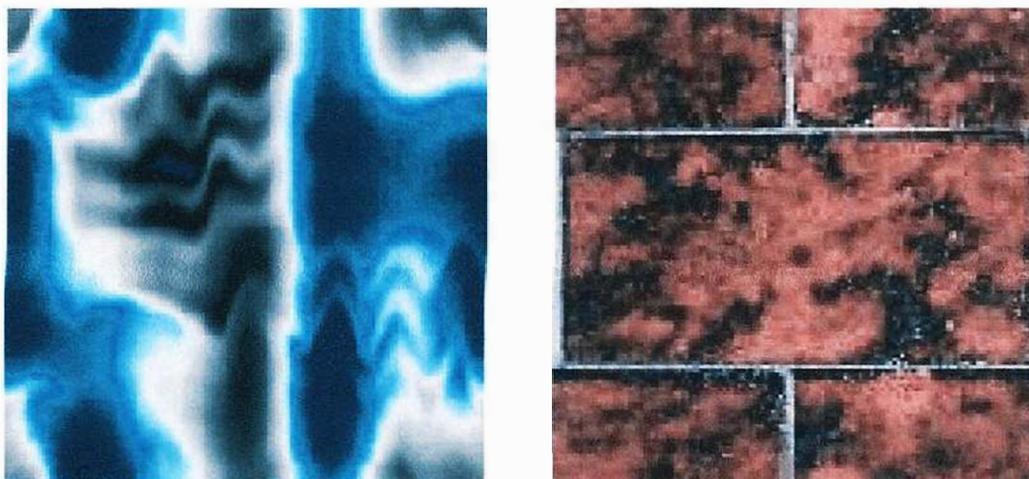
- Realizamos lo mismo para crear un segundo material, pulsamos 'New' y cambiamos el nombre del material 'Material02' por el nombre 'mat_pedestal'.



Pues ya tenemos los dos nombres de materiales que vamos a utilizar para nuestro objeto, ahora sólo nos queda asignarle un fichero gráfico a cada material antes de poder aplicar las texturas a nuestro objeto.

MilkShape 3D soporta multitud de formatos gráficos, pero para que nuestro objeto funcione correctamente en Entidad 3D es obligatorio utilizar ficheros gráficos BMP.

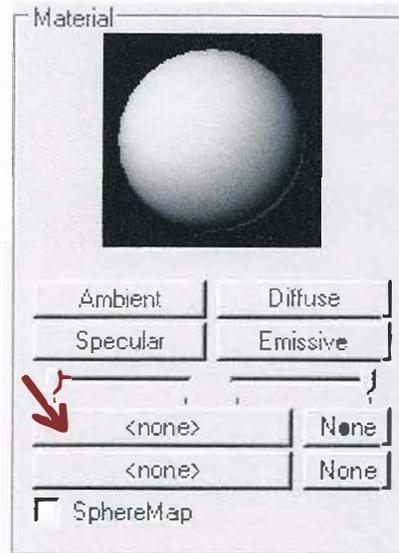
Con nuestro programa favorito de edición 2D, creamos dos ficheros BMP de unas dimensiones de 256x256 píxeles y 256 colores:



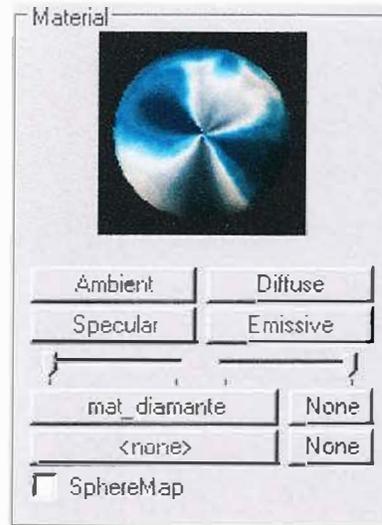
Texturas utilizadas para este ejemplo.

- En la carpeta donde estamos salvando el fichero .MS3D del proyecto, copiamos dos ficheros BMP con los nombres que quieras, pero para este caso les he dado los nombres de 'mat_diamante.bmp' y 'mat_pedestal.bmp'.

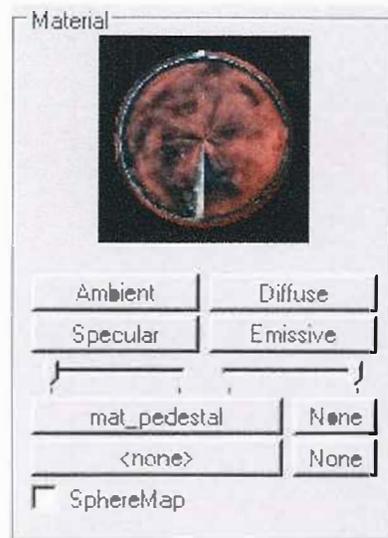
- En la pestaña 'Materials' seleccionamos el material 'mat_diamante' y pulsamos en el botón superior que pone 'none'.



- Seleccionamos el fichero 'mat_diamante.bmp', ahora el botón no mostrará 'none' sino el nombre del fichero gráfico que hemos seleccionado.

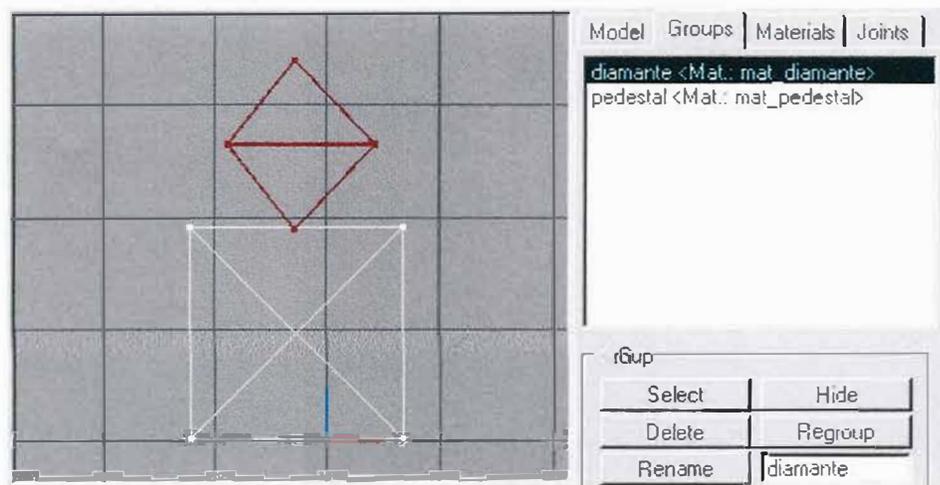


- Realizamos lo mismo con el material 'mat_pedestal' y el fichero gráfico 'mat_pedestal.bmp':

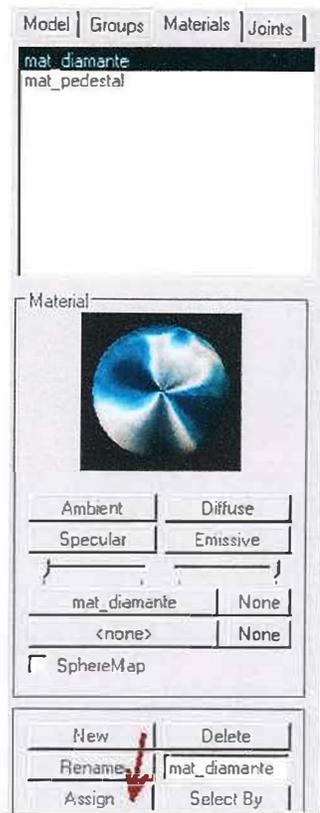


En estos momentos aún no hemos aplicado las texturas a nuestro objeto, simplemente tenemos dos materiales preparados para aplicarlos a cualquier grupo que deseemos.

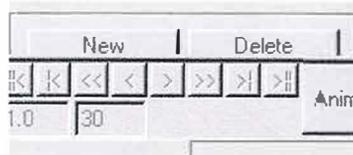
- Vamos a comenzar a aplicar la textura al diamante.
- Nos aseguramos de que no hay nada seleccionado pulsando simultáneamente (Ctrl + Mayús + A).
- En la pestaña 'Groups' seleccionamos el grupo 'diamante' y pulsamos 'Select'.



- En la pestaña 'Materials' seleccionamos el material que queremos aplicar a la selección; en este caso 'mat_diamante', y pulsamos el botón 'Assign'.



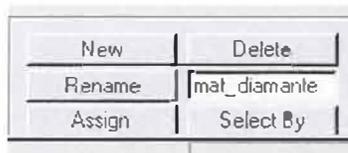
Nota: Si no puedes ver el botón 'Assign' que está en la parte inferior del panel de herramientas...



... oculta el deslizador de animación utilizando la opción del menú 'Window Show Keyframer'?



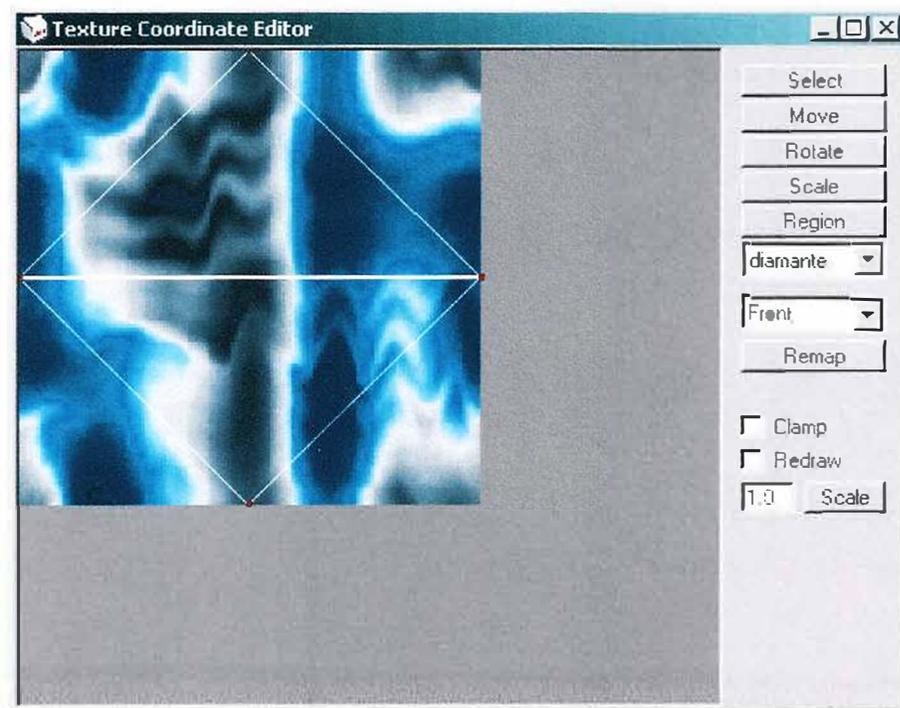
- De esta forma ya podrás acceder al botón 'Assign':



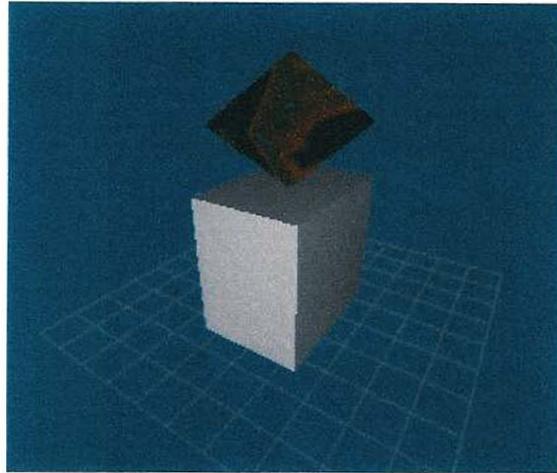
- Una vez utilizado el botón 'Assign' y con la vista 3D en modo 'Textured' podrás apreciar que el diamante ha adquirido una tonalidad distinta pero aún no puedes ver la textura aplicada. Esto es debido a que el grupo 'diamante' no tiene fijadas las coordenadas de aplicación de texturas.
- Las coordenadas de aplicación de texturas nos permiten definir la manera en que se aplicará la textura en un grupo en concreto, teniendo la posibilidad de mover la textura a lo largo del grupo, cambiarla de tamaño, girarla, etc., de manera que la textura no se aplique de cualquier manera y nos de la posibilidad de ajustar la textura como más nos interese.
- Este proceso se realiza con una herramienta que incluye MilkShape 3D llamada 'Texture Coordinate Editor' que podemos activar en cualquier momento utilizando la opción de menú 'Window Texture Coordinate Editor':



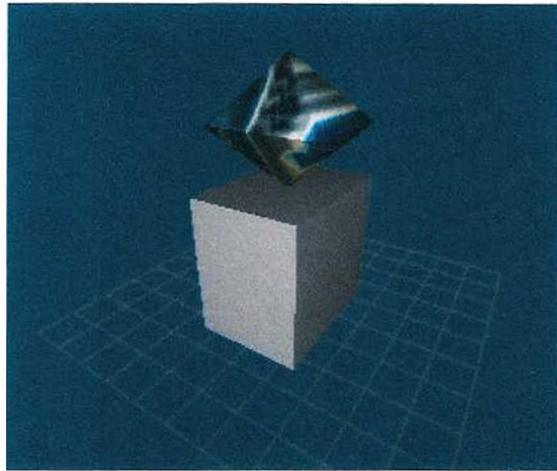
Con esta herramienta se pueden realizar diversas modificaciones en la aplicación de la textura, pero para una aplicación rápida seleccionamos el grupo que nos interesa texturizar (en este caso 'diamante'), el lado 'Front', pulsamos 'Remap' y finalmente cerramos la ventana:



- Ahora sí que podrás apreciar la textura aplicada en todo lo que pertenece al grupo del 'diamante':

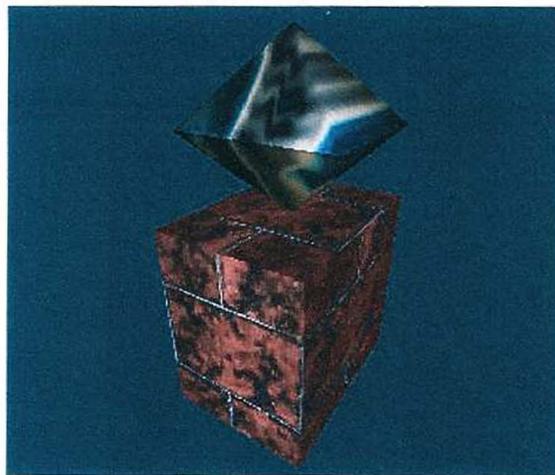


- Si quieres dar un poco más de brillo a la textura utiliza los botones 'Ambient' y 'Diffuse' seleccionando el color blanco:



- Ya sólo te queda realizar lo mismo para el grupo 'pedestal' y el material 'mat_pedestal', aunque en este caso **no será necesario utilizar 'Texture Coordinate Editor'**, ya que todos los objetos básicos cubos, esferas y cilindros ya disponen de coordenadas de aplicación de texturas.
- Pues ya hemos finalizado nuestro primer objeto con MilkShape 3D, este es el resultado final:



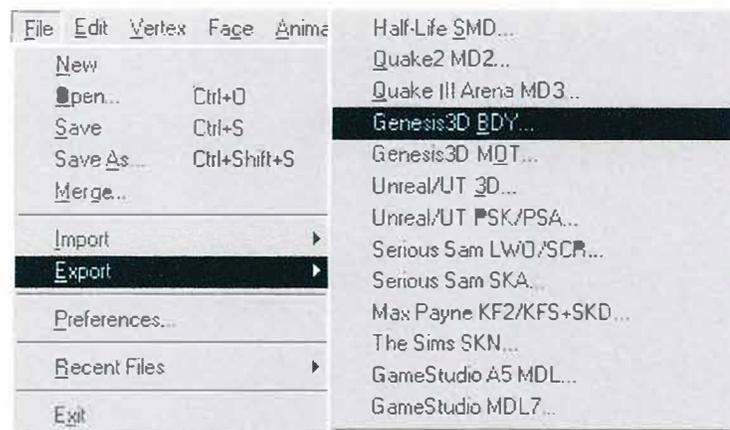


Las dimensiones de las texturas que podemos aplicar sólo pueden ser de tres tamaños (64x64, 128x128 o 256x256), de lo contrario Entidad 3D podría no funcionar justo cuando se disponga a cargar el objeto con la textura errónea.

EXPORTAR NUESTRO OBJETO EN UN FICHERO .ACT

En este punto ya tenemos todo nuestro objeto creado con MilkShape 3D, pero ahora nos interesa transformarlo en formato .ACT para poderlo utilizar en Entidad 3D.

- Para empezar hay que exportar el objeto a formato .DBY, para ello utilizaremos la opción del menú 'File Export Genesis3D DBY...'

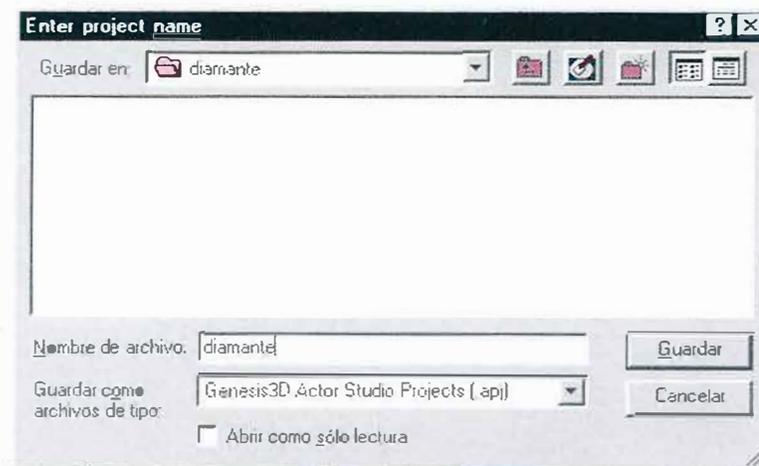


- Esto nos salvará en disco un fichero .DBY que por ejemplo llamaremos 'diamante'. Lo salvamos en la misma carpeta que estamos salvando todo hasta ahora.
- Abrimos el programa Actor Studio que incluye Entidad 3D, Actor Studio es el programa 'Astudio.exe' que encontrarás en la carpeta principal de Entidad 3D (por defecto 'C:\Entidad_3d').

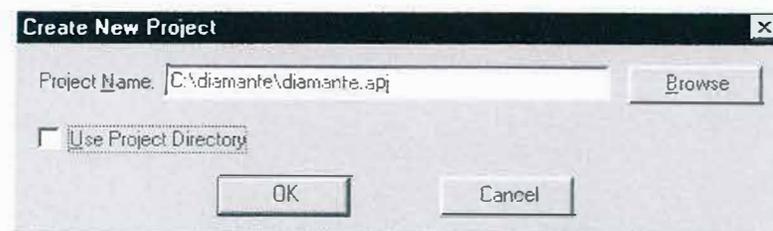
- Una vez en el programa Actor Studio utilizamos la opción del menú 'File New'.



- Con el botón 'Browse' nos dirigimos a la carpeta donde tenemos todo nuestro proyecto e introducimos 'diamante' como nombre de proyecto.



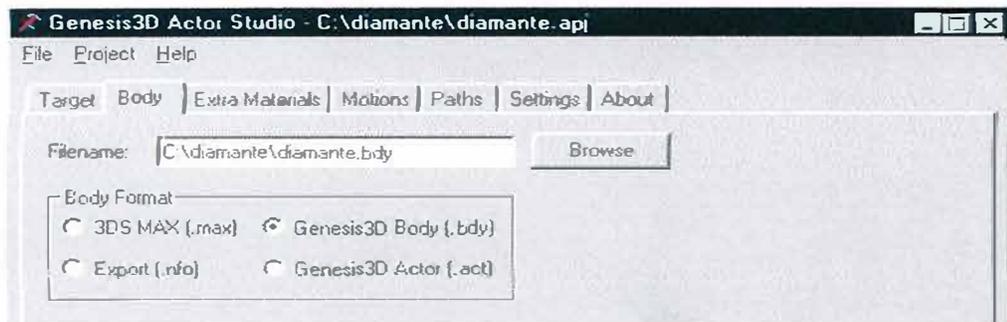
- En la siguiente ventana que nos aparezca desmarcamos 'Use Project Directory' y pulsamos 'OK'.



- Nos aparecerá la siguiente ventana, donde ya vemos el nombre que recibirá el fichero .ACT:



- Pulsamos en la pestaña 'Body'. luego marcamos 'Genesis3D Body' y con el botón 'Browse' cargamos nuestro fichero 'diamante.bdy':



- Finalmente pulsamos el botón 'Build'. Si todo ha ido bien ya tendremos creado nuestro fichero 'diamante.act' listo para utilizar en Entidad 3D con las entidades 'SpecificActor' o 'ItemObject'.



Si ha aparecido algún error después de pulsar el 'Build' o no puedes ver el fichero 'diamante.act' con el programa Actor Viewer, comprueba que los ficheros .DBY y .BMP están en la misma carpeta y que los BMP sean de las dimensiones correctas.

Aspecto final en Entidad 3D utilizando la entidad 'SpecificActor':



10.3 ALTERACION DEL ASPECTO DE LOS PERSONAJES.

Para la alteración de un personaje ya creado se requiere del programa MilkShape 3D (Shareware), un editor gráfico (Gimp o Photoshop) y el Actor Viewer. Cada una de estas herramientas permite manejar los distintos aspectos.

MilkShape 3D permite la empaquetación de un personaje en los ficheros de formato ACT el cual contienen las acciones de los personajes, las características físicas y el esqueleto del personaje.

Abriendo 'personaje.act' podemos observar el aspecto inicial del Personaje con el programa Actor Viewer que esta en el Genesis 3D:



Las modificaciones que se realizaran son de la parte frontal.

Lo primero que se hace es:

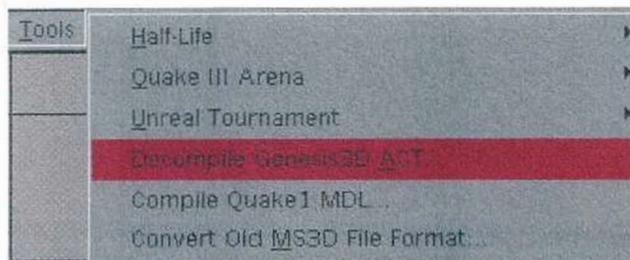
1.) Descompactando el fichero .ACT:

Un fichero de personaje .ACT (de Genesis 3D), en realidad es un archivo empaquetado (no en formato ZIP) que contiene toda la geometría, esqueleto, texturas y animaciones de un personaje en concreto.

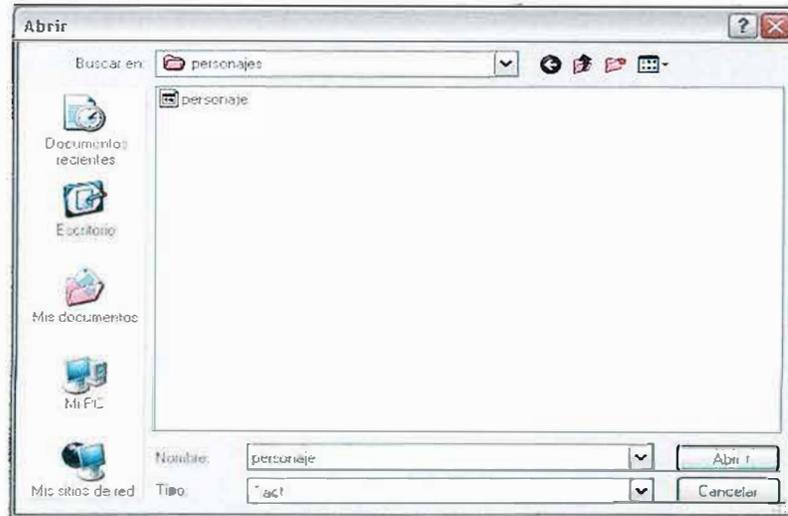
Este tipo de fichero no lo puede abrir directamente el programa MilkShape 3D, por lo que necesitaremos descompactar todo el contenido del fichero .ACT antes de poder tratarlo.

Para descompactar por ejemplo nuestro 'personaje.act':

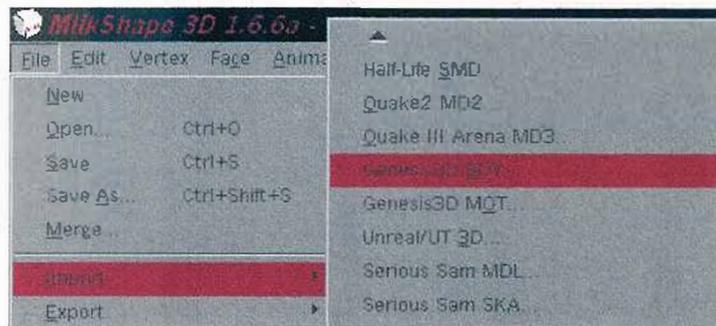
- Se ejecuta el programa MilkShape 3D.
- Mediante la opción de la barra de menú 'Tools > Decompile Genesis3D ACT'.



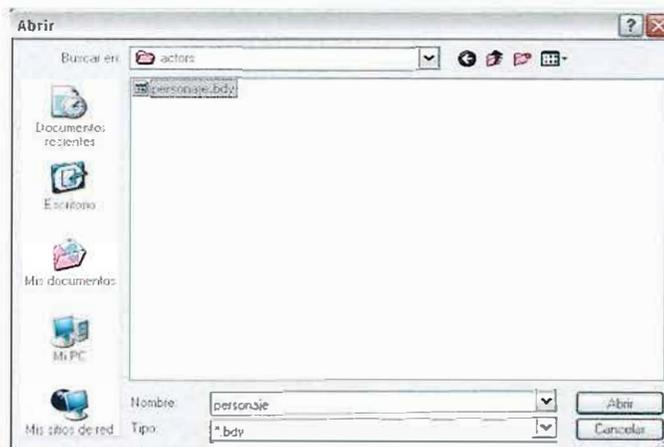
2.) Seleccionamos el .ACT que queremos descompactar:



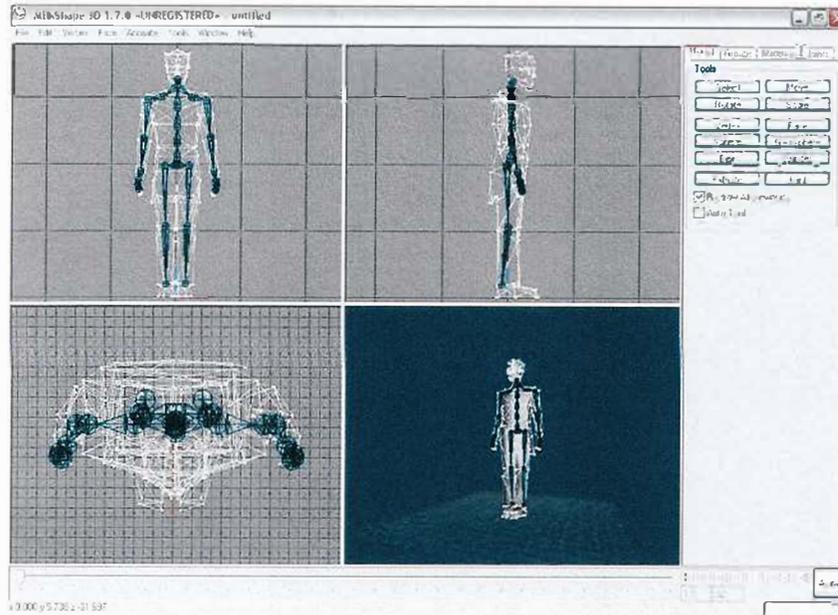
3.) A continuación utilizamos la opción de la barra de menú 'File > Import > Genesis3D BDY...':



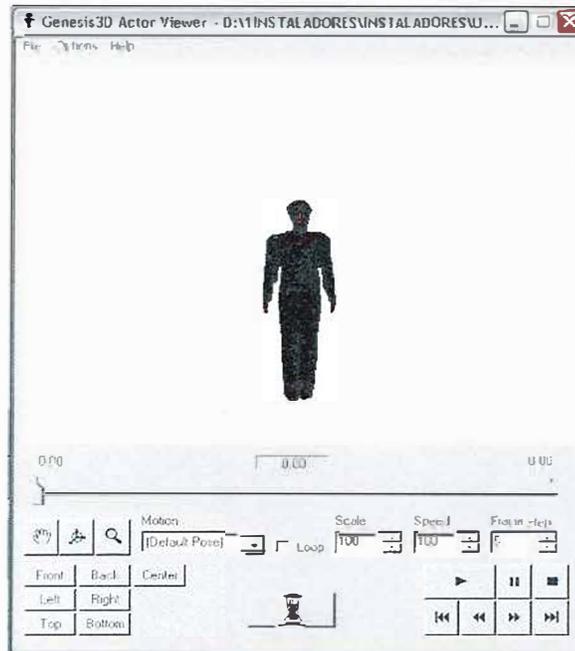
4.) Seleccionamos el único fichero .BDY que habrá en la carpeta 'C:\edicion_act':



Aspecto del personaje en Milk Shape:



La visualización del personaje se puede realizar con la herramienta Actor Viewer, esta permite observar todas las acciones que puede realizar el personaje, a continuación una vista de la interfaz de la herramienta:



Después de este paso las texturas que se desempaquetaron deben ser editadas con un programa de edición gráfica (Gimp o Photoshop) esto ya es cuestión de cada desarrollador el cual realizara las modificaciones según su criterio. Cuando se han

Finalizado todas las modificaciones se debe empaquetar todo de nuevo al formato .ACT con la herramienta Actor Studio.

RESUMEN

Este capítulo es bastante importante ya que los personajes son parte fundamental en un Videojuego por el papel interactivo que desempeñan respecto a los demás elementos y niveles de manera constante y por tanto deben ser distinguibles unos de otros; así como deben ser distintos, cada personaje tiene su función dentro de un Videojuego.

TALLER

Creá un objeto cualquiera en Milkshape, aplícale textura y expórtalo de tal forma que pueda ser utilizado en Genesis3D.

CAPITULO II. MANEJO DE AMBIENTES EXTERIORES.

En la creación, manejo y desarrollo de efectos exteriores se manejarán varios aspectos como la iluminación, imagen, agua, nubes, terrenos, atmósfera, y controles de render, permitiéndonos así crear fondos con paisaje y cielo que sería lo que veríamos a través de las ventanas, o techos y paredes que hayamos definido como abiertas al exterior en nuestro escenario. Con esta herramienta se podrán obtener mejores gráficos haciendo así mas cálido, atractivo, y sobre todo nos permitirá otorgarle un mayor realismo a los distintos escenarios del videojuego.

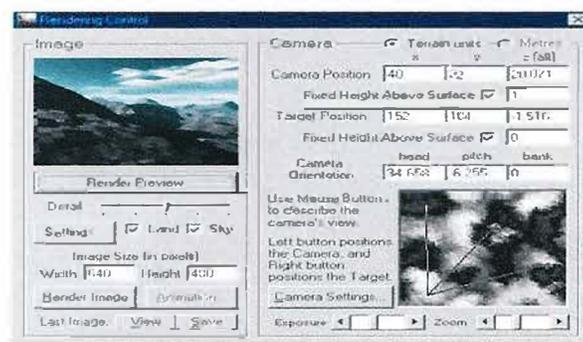
EL MANEJO DEL EDITOR DE EFECTOS

La herramienta ha utilizar en el desarrollo de efectos exteriores es el Terragen, el cual es el soporte que nos permitirá trabajar y manipular los distintos efectos como son los controles de render, terrenos, aguas, nubes, atmósfera, iluminación, imágenes, etc.

CONTROLES DE RENDER

Es el encargado de generar las imágenes, cabe destacar que de este efecto depende primordialmente que la imagen no salga distorsionada y se presente lo mas real posible, aquí es donde se asignan los valores de los distintos posiciones de las cámaras y en donde se configura la resolución de las imágenes, lo cual es muy importante ya que cuando el render se esta creando este toma los valores de los cuales fueron diseñados en la resolución especificada de la imagen por el mismo usuario.

Algo muy importante es la utilización de la memoria por parte del Terragen durante la generación de imágenes, ya que por medio de la buena administración de esta se puede brindar un resultado muy óptimo.

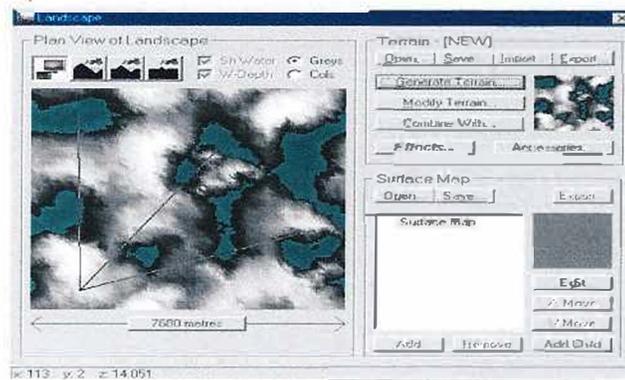


CUADRO DE DIÁLOGO DE TERRENOS

Aquí se puede generar el terreno ó abrir terrenos existentes que trae por defecto el Terragen, estos se pueden modificar y hacer combinación entre estos mismos, En el proceso de la creación ó bien sea de modificación, se deben tener en cuenta aspectos

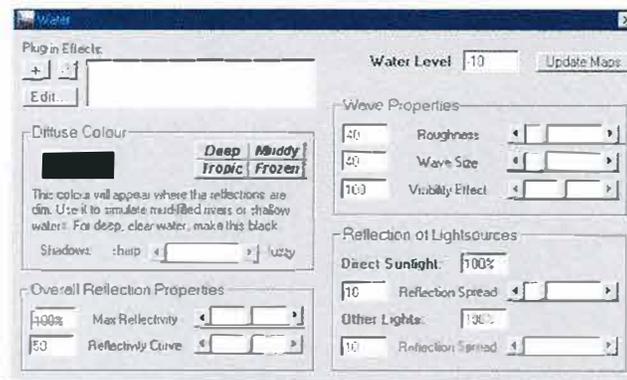
como son la acción, que es la encargada de generarlas características sobre el terreno a crear o existente si ya existe, el realismo, la suavidad del terreno, la glaciación, los cañones en bases en valles, llanuras, etc.

Cuanto mayor sea la resolución más detallada será al geografía. Sin embargo terrenos más grandes significan tiempos de render más largos, mas que el Terragen para el procesamiento de terrenos solo utiliza como máximo una resolución de 513x513.



CUADRO DE DIÁLOGO DEL AGUA

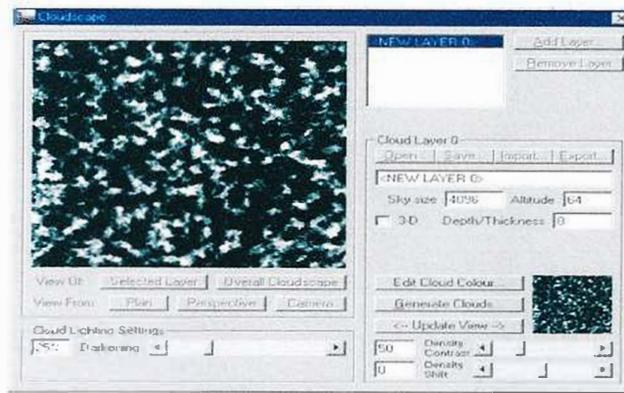
Al haber agua dentro de alguna parte de un escenario el usuario siente un estigma que se apacigua con la sensación de realismo obtenida en este mismo, básicamente dentro del Terragen se trabajan aspectos como Plug-in para el agua, el brillo de los reflejos, el contraste del agua, la altitud de la superficie del agua, entre otros.



CUADRO DE DIÁLOGO DE NUBES

Las imágenes del cielo y sus distintos cambios climatológicos dan mucha atracción por la forma como se originan estas sin precedencia alguna por el hecho de que estos son fenómenos naturales. Ahora si bien es cierto estos cambios de climas influyen mucho en las decisiones diarias de nuestras vidas, de igual forma estos cambios de climas dentro de un escenario le otorgan mayor dramatismo a un escenario, dependiendo de la escena que se este desarrollando dentro de este mismo escenario, hasta hacer contrastes con la vida real.

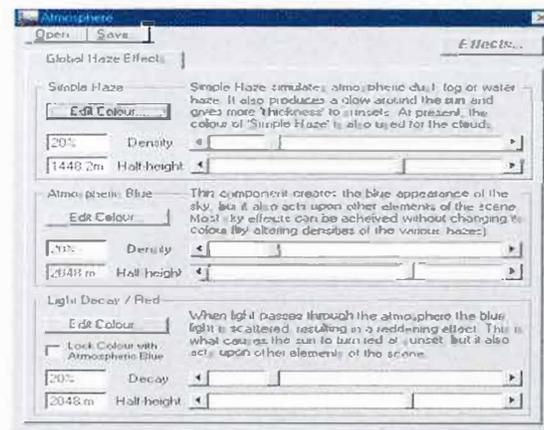




CUADRO DE DIÁLOGO DE ATMÓSFERA

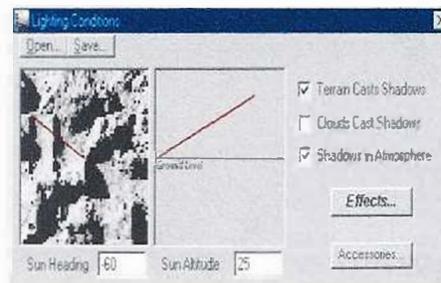
La atmósfera es un efecto que depende mucho del trabajo realizado en otros efectos, y por ende por ejemplo cuando la luz pasa a través de la atmósfera la luz azul es disipada.

La niebla simple simula el polvo atmosférico, la niebla o bruma. También produce brillo alrededor del sol y le da más densidad a los atardeceres. Es un buen método para aumentar la escala de la escena.



CUADRO DE DIÁLOGO DE ILUMINACIÓN

En este cuadro de diálogo en donde se pueden personalizar las preferencias de iluminación del Terragen. Además nos proporciona pestañas con las preferencias para luz solar directa, luz de fondo, apariencia del sol e iluminación atmosférica.



CUADRO DE DIÁLOGO DE IMAGEN

El cuadro de diálogo de imagen es independiente de la ventana principal del Terragen. Si la imagen es muy grande o muy pequeña se la escalará para que entre en la pantalla. En estos casos la imagen mostrada no se verá tan bien como aquella que sea almacenada en el disco.



11.2 CREACIÓN DE RENDERS PARA EXTERIORES.

La creación de los distintos componentes para efectos exteriores se podría decir que es una de las partes más importantes dentro del desarrollo de un videojuego, pero no tiene ningún sentido desarrollar estos efectos si no se van a implementar dentro de los distintos escenarios del videojuego. Con el Terragen se hace posible la importación de imágenes que han sido desarrollados desde este mismo, hasta el programa de edición gráfica del motor gráfico Genesis 3D.

Aquí están todas las vistas. Al final se tendrán 6 ficheros con las 6 vistas



Cielo5_back.bmp

Cielo5_bottom.bmp

Cielo5_front.bmp



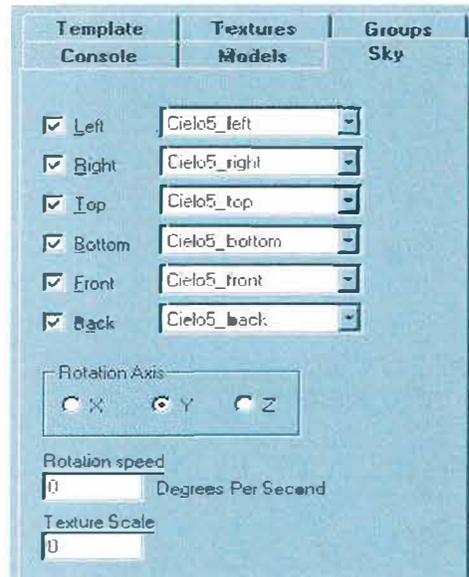
Cielo5_left.bmp

Cielo5_right.bmp

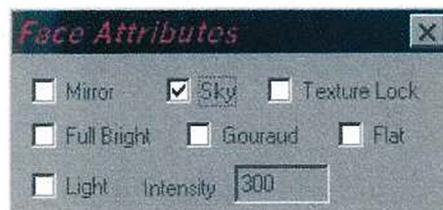
Cielo5_top.bmp

Aplicando el paisaje a nuestro escenario

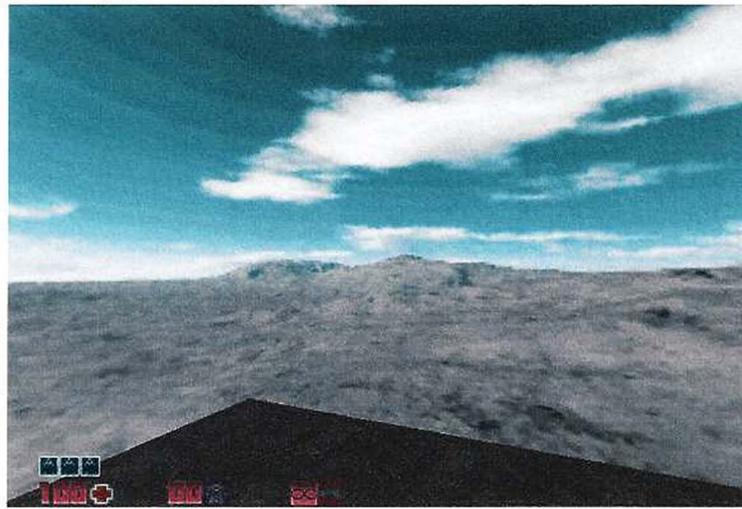
Finalmente lo único que nos queda es aplicar las 6 texturas del paisaje a nuestro escenario cargándolo con el World Editor agregando las texturas al fichero que las contiene con el Texture Packer que esta en la carpeta del Genesis 3D. Para ello seleccionaremos la pestaña 'Sky' dentro de la herramienta del motor y en cada caja correspondiente seleccionaremos la textura que pertenece a cada vista del cielo:



Compilamos el nivel y una vez dentro podrás apreciar en todo su esplendor el paisaje que has generado. Recuerda que para ver exteriores en tu escenario has de establecer 'Sky' en cada cara que desees que de al exterior:



- Un simple ejemplo de un escenario con las caras de paredes y techo establecidos como 'Sky' y con las texturas del cielo correctamente aplicadas tendría este aspecto:



- Si hemos hecho algo mal obtendríamos unas texturas que no casarían bien unas con otras, presentando este aspecto.



RESUMEN

Los elementos exteriores son muy importantes, pues estos son los que se encargan de crear la sensación de profundidad más allá del nivel en el que se localizan los personajes, pero cabe destacar que los personajes no interactúan con estos ambientes, pues estos están ahí como complemento de los Videojuegos para crear una armonía.

TALLER

Creá una textura para exteriores usando Terragen y luego aplicasela a una habitación mediante el uso de World Editor.

CAPITULO 12. MANEJO DE EFECTOS SONOROS.

En esta sección se plantea la importancia de los efectos sonoros en el desarrollo de los videojuegos, debido a que estos brindan un ambiente de suspenso y mayor realismo para el Videojugador. Dichos efectos van desde el simple sonido de la música de fondo, pasando por diversos aspectos importantes como las explosiones, sonidos de ambiente, los sonidos emitidos por las armas en el momento de los disparos, tiroteos entre jugadores, recargas de energía, etc.: todo esto, relacionado con la localización del personaje en el juego y el papel que desempeña en cada lugar del escenario, claro está que algunos de estos efectos sonoros están establecidos en el Motor Gráfico y se asignan una vez se agrega cada objeto a un escenario.

12.1 MANIPULACION DE EFECTOS SONOROS.

En este campo hay que tener en cuenta que hay muchas diferentes acciones que pueden ser realizadas por los personajes, o por algún objeto en particular, por tanto se debe prestar especial atención al momento de manipular efectos sonoros. A continuación algunas operaciones que se pueden realizar con la música de fondo del juego.

Todos los escenarios que se crean con World Editor deben contener por lo menos un personaje, estos por tanto emiten sonidos propios por cada acción que realizan como saltar o cuando son heridos, recoger algún objeto, es decir, todas las acciones que realiza un personaje en un juego de video.

En el caso de Genesis3D los sonidos son cargados por defecto desde una carpeta llamada 'wav' en el directorio raíz de Genesis 'C:\Genesis3D11' allí vamos a encontrar los sonidos que son activados por una acción realizada por un personaje de un Videojuego hecho en Genesis 3D; a continuación se describe que hace cada uno de los sonidos relacionados a un Videojuego diseñado en World Editor;

- Blaster: es el archivo de sonido asociado al Arma por defecto que tiene el personaje cuando aparecen en un nivel, este archivo se ejecuta automáticamente cuando realizamos la acción de disparar el arma, genera un proceso que llama al archivo y ejecuta reproduciendo el sonido del blaster.
- Blasterbang: es el archivo de sonido asociado al chocar un disparo de la del blaster a algún objeto o personaje del juego, esto significa que cuando ejecutamos el proceso de dispara y este llega al fin de la trayectoria se ejecuta el sonido que llamamos.
- Bounce es el archivo de sonido asociado al lanzamiento de granadas con el Grenade Launcher que tienen los personajes, una vez que esta rama es disparada se ejecuta el archivo de bounce para que se cree el sonido o el efecto de que la granda esta salpicando en una parte sólida del juego.

- Die: este archivo está asociado a la muerte del personaje es un efecto que genera la muerte del personaje creando un gemido al ser eliminado.
- DoorOpen: este archivo es llamado cuando se ejecuta el proceso de abrir y/o cerrar una puerta del Videojuego, y genera la sensación que vivimos los seres humanos cuando abrimos y/o cerramos puertas.
- Grenade: es el archivo asociado al explotar la granada que hemos lanzado con el Grenade Launcher y genera la sensación de explosión.
- hurt1 y hurt2: estos son archivos que están en secuencia y son gritos del personaje que se ejecutan cuando estos personajes son heridos.
- Itemspwn: es el sonido que es generado por los ascensores del juego de video, cuando es ejecutado el ascensor se pone en movimiento y genera el sonido.
- Jump: este sonido se ejecuta cuando el personaje salta en alguna parte del Video Juego.
- KeyPress: este sonido es generado cuando es oprimido una tecla dentro del Video Juego.
- Lockload: este archivo es generado cuando el personaje recoge las ramas en el camino.
- Mlaunch: este sonido es generado cuando se lanza un cohete con la rocket.
- Mimpact: es el sonido que se ejecuta cuando el cohete llega a ser impactado con algo.
- Onebzzi: este archivo es asociado al elemento electric bolt y es generado una vez que sale.
- PickupHealth: este archivo es ejecutado cuando el personaje tiene poca salud o energía y pasa por una fuente de energía y se alimenta o se recarga y da la sensación de alivio.
- Shredder: este archivo se ejecuta cuando el personaje tiene utilizando el arma Shredder y esta es accionada por el Videojugador.
- WeaponPickup: este archivo es generado cuando el personaje recoge la munición adecuada para el arma que está portando en ese momento.

MODIFICAR UN ARCHIVO DE SONIDO

Si se desea modificar alguno de los sonidos asociados a un evento, lo que se debe hacer es:

- Acceder al directorio raíz de Genesis 3D 'C:\Genesis3D\wav' donde están los sonidos.
- Seleccionar el sonido que se desea reemplazar y eliminarlo, cortarlo o sencillamente cambiarle el nombre por otro en el directorio 'wav'.
- Una vez se accede a este directorio se procede a pegar el nuevo archivo de sonido, y colocarle el nombre del archivo que se va a reemplazar, por ejemplo blaster o grenade, etc.

RESUMEN

Cuando se habla de efectos sonoros se entiende al instante que se habla de los sonidos relacionados con cada acción que ocurre en el Videojuego, esto debe ocurrir igual que en la vida real, por tanto todo el que diseñe juegos de video debe tener en cuenta este aspecto todo el tiempo para que cree juegos de calidad teniendo en cuenta que con esto se desarrollan juegos de video en tiempo real.

TALLER

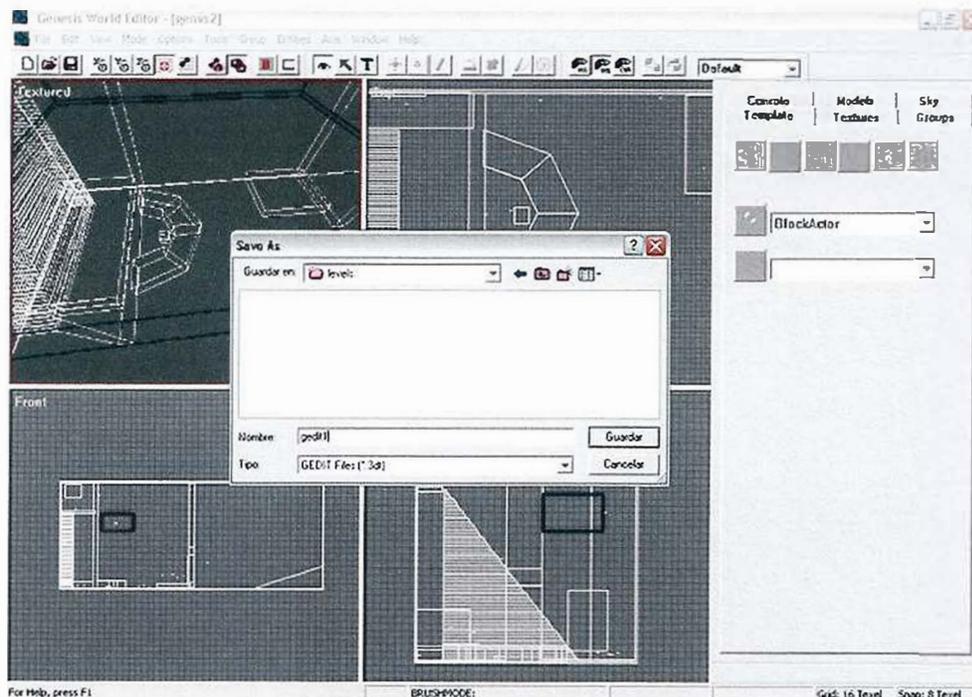
Mediante el procedimiento explicado reemplaza uno o varios de los archivos de sonido de Genesis 3D y prueba los resultados con World Editor.



CAPITULO 13. UNIFICACION DE COMPONENTES.

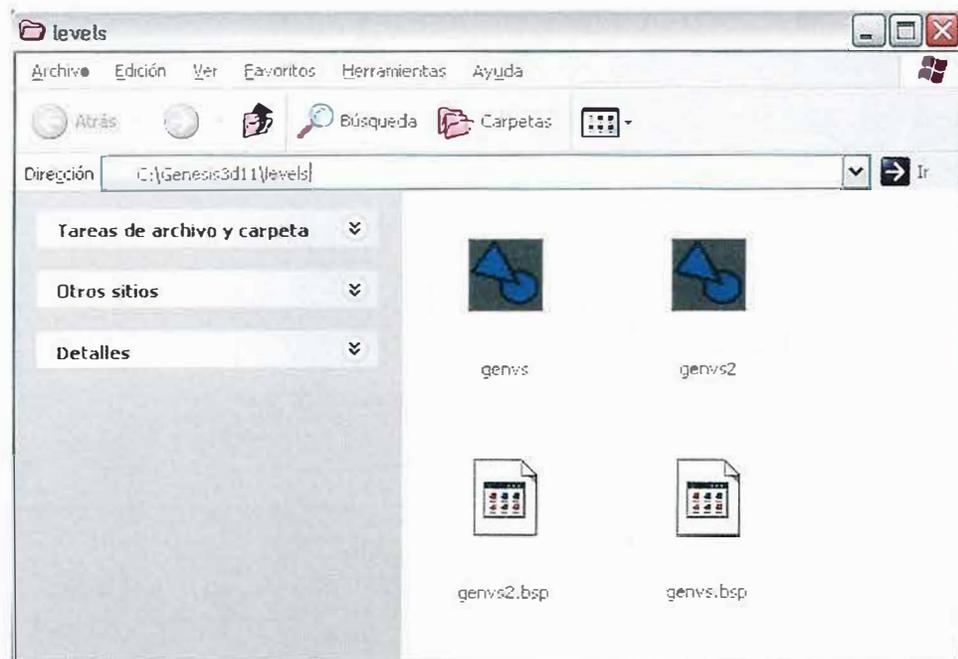
Al finalizar la Fase de Diseño de un Videojuego es bastante importante entregar a los Usuarios finales un paquete para facilitar a ellos la ejecución de los Videojuegos. Para realizar la unificación de los componentes es necesario llevar a cabo una serie de pasos para poder ejecutar el juego con el nivel que es creado por cualquier usuario del manual. A continuación se explicará paso a paso el procedimiento para la unificación de los componentes:

- Al terminar el nivel desarrollado en World Editor se debe guardar en la ruta que especifica C:\Genesis3D\levels y el nombre que desee (esto por cuestiones de organización).



- Una vez guardado el nivel desarrollado es necesario compilarlo, esto con el fin de generar los ficheros necesarios para la unificación de los componentes, estos ficheros se salvan en la carpeta donde se encuentra el nivel creado.
- Como siguiente paso debe eliminar o cortar y guardar en otra ubicación (si no desea perder el segundo nivel default que viene con el Genesis3D), los archivos `genvs2.bsp`, `genvs2.3dt`.

- Como ultimo paso debe cambiar los nombres los archivos generados al crear el nivel cuya extensión sean .bsp, .3dt, por `genvs2.bsp`, `genvs2.3dt`, esta carpeta debe quedar de la siguiente manera:



Luego de realizar los pasos del truco es necesario ejecutar el **GTest**, el cual es un demo que viene con Genesis3D y selecciona el segundo escenario, se podrá ver el nivel desarrollado ejecutando sin necesidad de abrir el World Editor y compilar.

RESUMEN

Al hablar de unificación de los componentes, se refiere a recopilar todos los elementos desarrollados y agruparlos en uno solo; esta es la parte que se describe en este capítulo, los pasos para lograr unificar los niveles desarrollados y así poder jugarlos sin la necesidad de ejecutar el World Editor.

TALLER

Con el fin de probar el nivel que haz desarrollado, lleva a cabo todos los pasos descritos y disfruta tu nivel ejecutándolo desde el **GTest** (Recuerda seleccionar el segundo escenario, debido a que este es el nivel que desarrollaste).

**IMPLEMENTACIÓN DE UN MANUAL DE TÉCNICAS PARA EL DESARROLLO
DE JUEGOS DE VIDEO MEDIANTE LA UTILIZACIÓN DE MOTORES
GRÁFICOS 3D PARA LOS ESTUDIANTES DE LA FACULTAD DE INGENIERIA
DE SISTEMAS EN LA CORPORACION EDUCATIVA MAYOR DEL
DESARROLLO SIMON BOLIVAR**

**BARROS ERNESTO
MONTAÑO GERMAN
PERALTA WALTER
RODRIGUEZ RICARDO
VALEGA ALFONSO**

**TRABAJO PRESENTADO COMO REQUISITO FINAL EN LA ASIGNATURA DE
INVESTIGACIÓN FORMATIVA IV**

**CORPORACIÓN EDUCATIVA MAYOR DEL DESARROLLO
SIMÓN BOLÍVAR
BARRANQUILLA
2005**

INDICE

INTRODUCCIÓN	2
1. PLANTEAMIENTO DEL PROBLEMA	3
1.1 DESCRIPCIÓN DEL PROBLEMA	3
1.2 FORMULACION DEL PROBLEMA	3
1.3 SISTEMATIZACIÓN	4
2. OBJETIVOS	5
2.1 OBJETIVO GENERAL	5
2.2 OBJETIVOS ESPECIFICOS	5
3. JUSTIFICACIÓN DEL PROYECTO	6
4. MARCO REFERENCIAL	7
4.1 MARCO TEORICO	7
4.2 MARCO LEGAL	11
4.3 MARCO ESPACIAL	12
5. HIPÓTESIS	13
6. METODOLOGÍA	14
6.1 TIPO DE ESTUDIO	14
6.2 LINEA DE INVESTIGACION	14
6.3 POBLACION Y MUESTRA	14
6.3.1. POBLACIÓN	14
6.3.2. MUESTRA	15
7. RECURSOS	16
7.1 RECURSOS HUMANOS	16
7.2 RECURSOS FINANCIEROS	16
7.3 RECURSOS DEL ENTORNO	16
8. CRONOGRAMA	17
9. INGENIERIA DE REQUISITOS	21
9.1 ANALISIS DE REQUISITOS	21
9.1.1 CAPITULO 1. INTRODUCCION A LOS VIDEOJUEGOS.	21
9.1.2 CAPITULO 2. INTRODUCCION A LOS MOTORES GRAFICOS.	21



9.1.3 CAPITULO 3. REQUERIMIENTOS DE HARDWARE Y MOTORES GRAFICOS UTILIZADOS.	21
9.1.4 CAPITULO 4. INTRODUCCION A LAS INTERFACES REQUERIDAS PARA EL DESARROLLO DE LOS VIDEOJUEGOS 3D.	22
9.1.5 CAPITULO 5. RECONOCIMIENTO DEL AMBIENTE DE TRABAJO DEL MOTOR GRAFICO Y EL EDITOR DE EFECTOS EXTERIORES.	22
9.1.6 CAPITULO 6. CREACION DE ENTORNOS 3D Y EDITOR DE NIVELES.	22
9.1.7 CAPITULO 7. CREACION Y MANEJO DE OBJETOS SÓLIDOS.	23
9.1.8 CAPITULO 8. MANEJO DE CRISTALES, ESPEJOS Y LIQUIDOS.	23
9.1.9 CAPITULO 9. ILUMINACION, NIEBLA, CIELOS Y RAYOS.	23
9.1.10 CAPITULO 10. DISEÑO Y CREACION DE PERSONAJES.	23
9.1.11 CAPITULO 11. MANEJO DE AMBIENTES EXTERIORES.	24
9.1.12 CAPITULO 12. MANEJO DE EFECTOS SONOROS.	24
9.1.13 CAPITULO 13. UNIFICACION DE COMPONENTES.	24
10. INGENIERIA DE INFORMACIÓN	25
10.1 MISION	25
10.2 VISION	25
10.2.1 NUESTROS PRINCIPIOS	26
10.2.2 NUESTROS VALORES	27
10.3 HISTORIA	28
10.4 ORGANIGRAMA	30
11 ANALISIS DEL SISTEMA.	31
11.2 DIAGRAMA JERARQUICO FUNCIONAL.	34
12. DISEÑO DEL SISTEMA	37
12.2 DESCRIPCION DE LOS COMPONENTES DE LA ESTRUCTURA FUNCIONAL	37
12.2.1 CAPITULO 1. INTRODUCCION A LOS VIDEOJUEGOS.	37
12.2.2 CAPITULO 2. INTRODUCCION A LOS MOTORES GRAFICOS.	37
12.2.3 CAPITULO 3. REQUERIMIENTOS DE HARDWARE Y MOTORES GRAFICOS UTILIZADOS.	39
12.2.4 CAPITULO 4. INTRODUCCION A LAS INTERFACES REQUERIDAS PARA EL DESARROLLO DE LOS VIDEOJUEGOS 3D.	40

12.2.5 CAPITULO 5. RECONOCIMIENTO DEL AMBIENTE DE TRABAJO DEL MOTOR GRAFICO Y EL EDITOR DE EFECTOS EXTERIORES.	40
12.2.6 CAPITULO 6. CREACION DE ENTORNOS 3D Y EDITOR DE NIVELES.	41
12.2.7 CAPITULO 7. CREACION Y MANEJO DE OBJETOS SÓLIDOS.	42
12.2.8 CAPITULO 8. MANEJO DE CRISTALES, ESPEJOS Y LIQUIDOS.	42
12.2.9 CAPITULO 9. ILUMINACION, NIEBLA, CIELOS Y RAYOS.	43
12.2.10 CAPITULO 10. DISEÑO Y CREACION DE PERSONAJES.	43
12.2.11 CAPITULO 11. MANEJO DE AMBIENTES EXTERIORES.	44
12.2.12 CAPITULO 12. MANEJO DE EFECTOS SONOROS.	44
12.2.13 CAPITULO 13. UNIFICACION DE COMPONENTES.	44
ANEXOS	45
ANEXO No. 1	45
ANEXO No. 2	48
ANEXO No. 3	50



INTRODUCCIÓN

Para el enfoque que se le ha dado a este proyecto, debe considerarse que debido a la proliferación en los últimos años de los videojuegos y al importante mercado que ha surgido alrededor de ellos como una aplicación más del mundo informático; se hace imperante el conocer sobre el soporte de esta tecnología. Uno de estos soportes, sino se puede considerar el más importante son los motores gráficos.

En Colombia el desarrollo y la utilización de este tipo de programas es prácticamente inexistente. Por esto al plantear la pregunta y entender la problemática que rodea el ámbito en que se desenvuelve el estudiante en la Corporación Educativa Mayor del Desarrollo Simón Bolívar se plantea la posibilidad de elaborar mecanismos de comprensión, utilización y aprovechamiento de esta tecnología; como una posible opción a futuro para el desarrollo informático en el país.

Los motores gráficos no son herramientas surgidas de la noche a la mañana. Son el desarrollo de años de avances tecnológicos, los cuales han logrado un grado alto de calidad en los videojuegos que se desarrollan con su utilización. En pocas palabras este proyecto se fundamenta en la elaboración de un manual para el desarrollo de videojuegos, utilizando motores gráficos con licencia GNU, ya que es útil para el desarrollo integrado con este tipo de software que cuenta con esta licencia de libre distribución y permitiría la divulgación del manual.



1. PLANTEAMIENTO DEL PROBLEMA

1.1 DESCRIPCIÓN DEL PROBLEMA

Debido al poco desarrollo que se presenta en la Corporación Educativa Mayor Del Desarrollo Simón Bolívar en los aspectos concernientes al uso de herramientas de diseño 3D como lo son los motores gráficos teniendo esto en cuenta como un surgimiento de nuevas tecnologías. El desconocimiento que existe por parte de la gente del común y los estudiantes de la Corporación en todos los aspectos referentes a los video juegos debido a la falta de interés, hace que se puedan enumerar varios aspectos que pueden tomar gran importancia en el ámbito local, pero esto no ha llegado ha ser posible pues esta tecnología siempre es observada como un simple pasatiempo pues la gente tiene la creencia que los videojuegos son solo para niños, siendo totalmente falso ya que los videojuegos tienen diferentes géneros de acuerdo a la edad del individuo que quiera jugar.

El gran auge que se ha observado en los últimos de los videojuegos y el importante mercado que ha surgido alrededor de ellos como una aplicación más del mundo informático: se hace imperante el conocer sobre el soporte de esta tecnología. Uno de estos soportes, sino se puede considerar el más importante son los motores gráficos. En Colombia y en la corporación educativa mayor del desarrollo simón bolívar el desarrollo y la utilización de este tipo de programas es prácticamente inexistente, debido a la falta de interés y la creencia de subestimar este campo.

1.2 FORMULACION DEL PROBLEMA

¿En que manera se puede gestar un cambio en el pensamiento de la comunidad en lo referente al tema de los videojuegos, logrando además eliminar las falsas creencias sobre este campo que provocan una percepción errónea?



1.3 SISTEMATIZACIÓN

¿Cómo lograr el entendimiento del diseño de los videojuegos?

¿En que forma puede ser beneficiosa esta tecnología?

¿Cómo conseguir el desarrollo de un videojuego?

¿En que forma se puede lograr que la información generada sea útil?

2. OBJETIVOS

2.1 OBJETIVO GENERAL

Implementar un manual para el diseño de juegos de video aplicando técnicas desarrollo mediante el uso de motores gráficos 3D y su implementación en el desarrollo de una demostración.

2.2 OBJETIVOS ESPECIFICOS

- Establecer pautas generales para que los usuarios del producto final lleguen a comprender el desarrollo de juegos de video.
- Desarrollar juegos de video con base en el manual que logren un impacto comercial.
- Identificar las diferentes técnicas de desarrollo de juegos de video e implementarlas de una manera practica.
- Desarrollar una demostración con base a la información contenida en el manual.

3. JUSTIFICACIÓN DEL PROYECTO

Dar a conocer los aspectos positivos de los videojuegos, dando a entender la forma en que se desarrolla un videojuego, resaltando el impacto que tienen en la juventud y tratar que la comunidad no cometa errores al momento referirse a los videojuegos.

Con el desarrollo de este proyecto se busca el entendimiento de las herramientas mediante las cuales se diseñan los juegos de video teniendo en cuenta los motores gráficos 3D. Aparte de eso se busca implementar el uso de un manual que permita el entendimiento y manipulación del material entregado a los posibles desarrolladores interesados en este tema.

Es un aspecto muy importante el funcionamiento de los motores gráficos, los cuales manejan los diferentes objetos visibles en el ambiente del juego, como son los efectos, los ambientes y toda la parte visual de un juego.



4. MARCO REFERENCIAL

4.1 MARCO TEORICO

Los videojuegos han tenido un desarrollo vertiginoso en los últimos años llegando a ser una industria que genera millones de dólares al año y llegando a tener un impacto enorme en la sociedad actual, como medio de entretenimiento para personas de todas las edades y llevando el desarrollo de los videojuegos a un nivel cultural que rivalizaría con campos como el cine y la música, debido a que los videojuegos logran desarrollar la lucidez mental, los reflejos y la agilidad psíquica.

“Todo comenzó en el año de 1961, cuando Steve Russell creó un videojuego llamado Space Invaders solo para demostrar la capacidad de las computadoras de la época y nunca se pensó en él como una oportunidad de aprovechamiento comercial”¹. En el año de 1973 Nolan Bushnell funda la compañía ATARI, lanzando un aparato lógico llamado Pong. A partir de aquí y hasta mediados de los 80s las consolas de videojuegos fueron fabricadas exclusivamente por empresas estadounidenses y usaban procesadores de 8 bits. En 1983 la empresa NINTENDO lanza su consola Famicom (NES) en Japón y dos años después en Estados Unidos logrando un gran impacto comercial, seguida por SEGA. A partir de ese instante las compañías japonesas tomarían el liderazgo del mercado. A principios de los 90s se lanzan las consolas de 16 bits como el SNES, el SEGA Génesis o el Neo – Geo de SNK, y años más tarde entraría al mercado SONY con el sistema Playstation siendo el primer aparato casero de mostrar gráficos 3D poligonales. En los últimos años surgieron sistemas de 64 bits y después los sistemas de 128 bits como el Dreamcast de SEGA, el Playstation2 de SONY o el Gamecube de NINTENDO.

Los videojuegos en los computadores han tenido un desarrollo diferente al de los diseñados para consolas debido a que los juegos de consola se desarrollan con motores gráficos privados (los cuales no salen a la venta debido a que cada empresa desarrolladora tiene sus propios motores). En cambio los motores desarrollados para juegos de computadora son motores públicos (estos pueden ser adquiridos por compañías diferentes a la compañía creadora del motor), y siguen desarrollos diferentes.

¹ Tomado de la página Web <http://www.librosdelosandes.com/biblos/bibloscopio4.htm>

“Hace 8 años el motor más popular era el motor Doom el cual era muy limitado porque el personaje no podía mirar hacia arriba ni hacia abajo. Un año después se lanzó el motor Duke Nuke 3D que limitadamente permitía mirar hacia arriba y abajo. 3 años más tarde aparece el motor Quake el cual ofrece niveles basados en polígonos, dando libertad para mirar en todas las direcciones. 2 años más tarde los motores Unreal y Quake II introducen efectos más reales como halos alrededor de las luces, niebla y luces de colores entre otros”². Después surgiría el motor Quake III, logrando mayor realismo. Seguido a su vez por los motores Unreal II y el Doom III.

Desde hace mucho tiempo en nuestro entorno social se ha presentado una masificación del uso de los videojuegos como pasatiempo, pero nadie se ha interesado por investigar como se diseñan y desarrollan los juegos de video tridimensionales, es por esto que el enfoque del proyecto a desarrollar se basa en la utilización de varias herramientas de diseño 3D y su intensiva observación para el desarrollo de un manual en el cual se pueda explicar el uso de las distintas herramientas que nos proponemos usar.

En la corporación el único vestigio de progreso en el campo de investigación en el cual se va a trabajar ha sido un proyecto que utilizando juegos desarrollados en Visual Basic para el fortalecimiento del aprendizaje se han implementado unas técnicas mediante las cuales el computador y el usuario interactúan para que haya un entendimiento, es por esto que en el transcurso del diseño de un juego es muy importante pensar en el usuario para que este se sienta cómodo con el producto que tenga en sus manos.

En las instituciones universitarias de Barranquilla se enseñan programas de diseño 3D pero únicamente con el fin de desarrollar animaciones, las cuales no interactúan con el usuario y no presentan novedades en este campo ya que al ser solo animaciones es similar al hecho de ver una película; por lo tanto es algo secuencial que no varía.

La evolución continua de los motores, desarrollo un mercado próspero en Estados Unidos y en Europa; a diferencia que en Japón donde las compañías desarrolladoras se enfocan en el desarrollo para consolas. Las compañías Estadounidenses y Europeas

² Tomado de la revista Enter No 34 Abril 2001.

han desarrollado elementos 3D sumamente reales, logrando un gran avance en este campo pero enfocados en los computadores.

En los últimos años, con la aparición de motores gráficos gratuitos se da una apertura de oportunidades para aquellas personas interesadas en este campo, puedan desarrollar sus capacidades sin necesidad de adquirir costosas licencias, pero a pesar de esto en la actualidad en nuestro medio local no se han concebido investigaciones o proyectos que en base al uso de motores gráficos, traten lo referente al diseño 3D en la creación de juegos. Siendo el aspecto fundamental del proyecto de investigación es el de brindar un soporte para el desarrollo de videojuegos basándose en la utilización de motores gráficos 3D.

El apartado técnico y conceptual que abarca el desarrollar un videojuego es fundamental para entender este campo de desarrollo que se ha abierto en los últimos años. La magnitud con la que ha crecido el desarrollo de videojuegos ha generado un avance tanto en el hardware como el software dirigido para este campo, motivando una carrera para el desarrollo de nuevos juegos que aparecen cada mes, moviendo una industria que cada día crece más.

Las consolas de videojuegos son máquinas destinadas a la representación de juegos en el aparato de televisión. Para su uso se deben conectar a la entrada de video y audio del TV, disponen de controles diseñados ergonómicamente para el manejo de la consola disponiendo de un pad o un joystick. Los juegos disponibles para las consolas son muy disímiles de los desarrollados para computadora en cuanto a su temática y su desarrollo gráfico. Generalmente los juegos son desarrollados con motores gráficos privados.

Las consolas son el fundamento de los videojuegos porque fueron los primeros dispositivos de hardware que soportaron juegos de video con fines comerciales y han sido las que han ido a la vanguardia en el gráfico y temático de los juegos, llevando el desarrollo de estos a los límites de la tecnología debido a que al salir al mercado nuevos modelos de consolas estas superan de inmediato la capacidad de los computadores los cuales llegan a la capacidad de las consolas solo tres o cuatro años de haber salido la generación de consolas vigente.

El tener claro la importancia de las consolas es fundamental en el desarrollo de los videojuegos pero debido a la poca información sobre la programación de estas se dificulta el poder desarrollar juegos para estas plataformas y no da más opción que desarrollarlos para los computadores personales, debido a que es una plataforma más común entre los posibles usuarios.

“Para desarrollar videojuegos se puede recurrir a distintas herramientas, pero la más eficiente y robusta es el motor gráfico. El motor gráfico o motor de juegos es una aplicación que ejecuta todos los aspectos de un juego, ósea que las líneas de código del motor son las encargadas de generar todos los elementos que aparecen ante la vista del usuario, también calcula la cantidad de polígonos que se deben usar para la construcción de todos los elementos visualizados”³.

“Este programa sirve para crear juegos diferentes en género y apariencia pero que estarían montados sobre la misma plantilla. Dado que esta industria mueve muchos millones de dólares, los fabricantes no pueden darse el lujo de esperar mucho tiempo para lanzar un nuevo título”.

“Los motores gráficos se pueden catalogar en tres categorías.”

- ✓ Motores Privados: “son aquellos usados por una empresa en particular, ósea ellos mismos lo desarrollan y lo utilizan para sus juegos exclusivamente. Este tipo de motores no están a la venta comercial y preferiblemente las empresas que adoptan esta políticas son las japonesas. Ej. Capcom, Square-Enix, Konami, Sega, SNK, etc”.
- ✓ Motores Comerciales: “este tipo es usado por las empresas norteamericanas y europeas son motores desarrollados por empresas de juegos pero después colocan a la venta tal es el caso del motor Quake, Unreal, Doom3, etc. El precio de estos motores oscila entre los 200.000 y los 500.000 dólares”.
- ✓ Motores Gratuitos: “son aquellos que se rigen por las licencias GNU o GPL”⁴, por lo cual se hace viable obtener un motor de estos ya que es útil para propósitos académicos.

³ Tomado de la Revista Enter No34 Abril 2001.

⁴ Licencia Pública General (GPL por General Public License), tomado de la pagina Web www.linuxmandrake.com.

La importancia de los motores gráficos radica en que permiten crear juegos en menor tiempo, en cambio si se comenzaran a crear todas las plantillas de la nada sería un proceso muy lento y agotador, debido a que se tendría que desarrollar el motor gráfico llegando a ser una tarea muy dispendiosa y para una persona con poca experiencia podría llegar a ser inacabable. Aquí radica la importancia de un motor gráfico ya que solo los desarrolladores tendrían que concentrarse en realizar los juegos. Las distintas clases de motores se pueden observar como el interés comercial que tienen los videojuegos ya que los motores privados y públicos.

Algunos motores gratuitos pueden correr en diferentes plataformas, como el caso del Cristal Space que puede ser ejecutado en UNIX (Linux, y Solaris), DOS, Amiga, Macintosh, Windows, BeOS, NextStep, Rhapsody y ports OpenStep. Otros como el caso de Génesis 3D o el Fly 3D solo se ejecutan en Windows.

4.2 MARCO LEGAL

Los motores gráficos están cobijados a distintas reglamentaciones dependiendo del tipo que sean. Los motores privados y públicos son motores que rigen por las distintas leyes de derechos de autor que existan en los diferentes países del mundo; Los motores privados no son de dominio público, al contrario de los públicos que pueden ser adquiridos por particulares, tienden a correr el peligro de sufrir violaciones a sus derechos de autor.

Los motores gratuitos se rigen por licencias diferentes dependiendo del motor. “Algunos como el Cristal Space se rigen por la licencia LGPL”⁵ (Lesser General Public License o Licencia Pública General Menor), que permite hacerle modificaciones propias al código fuente y solo se distribuyen de acuerdo a la misma licencia. En otros como el Fly 3D la licencia es totalmente gratuito y si se quiere se modifica el código y los juegos desarrollados por este no están sujetos a la misma licencia del motor. El Génesis 3D también es de licencia libre y con derecho a modificar el código pero con la condición de mostrar el logo de Génesis 3D, también cuenta con licencia comercial a cambio de no mostrar el logo. Todos estos motores están regidos bajo licencia GNU, que permite la libre distribución del software.

⁵ Tomado de la licencia del Motor Grafico Genesis 3D.



4.3 MARCO ESPACIAL

El proyecto se desarrollara en la Corporación Educativa Mayor Del Desarrollo Simón Bolívar ubicada en Kr 54 con Cll 59 esquina en la ciudad de Barranquilla del departamento del Atlántico en la Republica de Colombia.



5. HIPOTESIS

Este proyecto pretende que los estudiantes de la corporación desarrollen un espíritu investigativo donde tengan en cuenta la realización de los videojuegos considerando el uso de motores gráficos y herramientas de diseño 3D; implementando el manual se desarrollan técnicas para diseñar videojuegos basándose en herramientas que generan escenarios tridimensionales, esto se dirige a los estudiantes que están interesados en el tema y pertenecen a la Facultad de Ingeniería de Sistemas de la Corporación Educativa Mayor Del Desarrollo Simón Bolívar, y por ultimo generar un Demo que produzca un impacto mediante el cual se estimula la generación de nuevas ideas investigativas.

6. METODOLOGIA

6.1 TIPO DE ESTUDIO

El diseño e implementación de este proyecto esta enfocado a la visualización de nuevas ideas investigativas, con el fin de crear un ambiente que permita la generación de ideas novedosas y que permitan el desarrollo del diseño 3D, mediante el uso de herramientas novedosas; por lo tanto el proyecto seguiría un paradigma empírico-analítico basándose en un tipo de investigación descriptiva.

6.2 LINEA DE INVESTIGACION

El proyecto se enfoca en el campo del diseño 3D, pero al ser encaminado en lo concerniente a los videojuegos toma una dirección ceñida al campo de la Inteligencia Artificial específicamente a la rama de los Sistemas Expertos.

6.3 POBLACION Y MUESTRA

6.3.1. POBLACIÓN

El desarrollo de este proyecto se encamina a los estudiantes de la Facultad de Ingeniería de Sistemas. El buen uso de las herramientas utilizadas en el diseño e implementación de los videojuegos pueden atraer la atención a estudiantes que son seguidores de los videojuegos y que de alguna forma quieren estar relacionados con todo lo concerniente al desarrollo de este tipo de tecnología.

6.3.2. MUESTRA

Para el desarrollo del proyecto se llevo a cabo un estudio aplicado a algunos estudiantes de la facultad con el fin de ver la necesidad de desarrollar técnicas capaces de mostrar la creación de juegos en ambientes gráficos 3D. El grupo de trabajo aplicando los conocimientos adquiridos mejoró la idea y se pone en práctica el desarrollo del proyecto.

Para la Muestra del Proyecto se realizaron 50 encuestas al mismo número de estudiantes de la Facultad de Ingeniería de Sistemas, los resultados obtenidos pueden observarse en el Anexo 2.



7. RECURSOS

7.1 RECURSOS HUMANOS

Este proyecto esta siendo desarrollado por los estudiantes de la Corporación Educativa Mayor de Desarrollo Simón Bolívar pertenecientes a la Facultad de Ingeniería de Sistemas, los cuales cursan actualmente séptimo semestre de esta misma como son **ERNESTO BARROS, GERMAN MONTAÑO, WALTER PERALTA, RICARDO RODRIGUEZ** y **ALFONSO VALEGA**.

7.2 RECURSOS FINANCIEROS

CONCEPTO	VALOR
ALMUERZOS	\$ 495.800
IMPRESIÓN	\$ 250.000
INTERNET	\$ 132.000
PASAJES	\$ 400.000
TOTAL	\$ 1.277.800

7.3 RECURSOS DEL ENTORNO

El PC en el cual se desarrolla el proyecto tiene las siguientes características:

- Mother Board MSI.
- Monitor HP de 17".
- Procesador AMD Athlon 2400MHz.
- Chip gráfico integrado de 32MB.
- 512MB de Memoria DDR.
- Unidad de Disco Duro de 80GB.
- Tarjeta de sonido integrada.
- Tarjeta Modem de 56K.
- Tarjeta de Red PCI NIC 10/100 MB.
- Unidad de CD – RW LG 52x32x52x.

8. CRONOGRAMAS

1^{er} Semestre de 2004

Actividades	Febrero				Marzo				Abril				Mayo				Junio			
	I ⁶	II	III	IV	I	II	III	IV	I	II	III	IV	I	II	III	IV	I	II	III	IV
Análisis del entorno del proyecto de formativa I																				
Recaudación de información para las propuestas																				
Presentación de las diferentes propuestas para el proyecto																				
Elaboración de la formulación del problema																				
Diseño de objetivos y justificación																				
Recolección de información concerniente al diseño 3D																				
Elaboración de los antecedentes históricos																				
Primera revisión del proyecto																				
Elaboración del marco teórico y la metodología																				
Correcciones al marco teórico																				
Entrega del proyecto																				

⁶ Las columnas verticales indican las semanas.

2^{do} Semestre de 2004

Actividades	Julio				Agosto				Septiembre				Octubre				Noviembre			
	I	II	III	IV	I	II	III	IV	I	II	III	IV	I	II	III	IV	I	II	III	IV
Análisis del entorno del proyecto de formativa II																				
Revisión del proyecto y ajustes del proyecto																				
Presentación de las diferentes propuestas para el proyecto																				
Ingeniería de requisitos																				
Ingeniería de información																				
Recolección de información concerniente al diseño 3D																				
Análisis de sistema																				
Estructura funcional																				
Diseño de interfaces (Prototipos)																				
Revisiones																				
Entrega del proyecto																				



1^{er} Semestre de 2005

Actividades	Febrero				Marzo				Abril				Mayo				Junio			
	I	II	III	IV	I	II	III	IV	I	II	III	IV	I	II	III	IV	I	II	III	IV
Desarrollo del Capítulo 1																				
Desarrollo del Capítulo 2																				
Desarrollo del Capítulo 3																				
Desarrollo del Capítulo 4																				
Desarrollo del Capítulo 5																				
Desarrollo del Capítulo 6																				
Desarrollo del Capítulo 7																				
Desarrollo del Capítulo 8																				
Desarrollo del Capítulo 9																				
Desarrollo del Capítulo 10																				
Desarrollo del Capítulo 11																				
Desarrollo del Capítulo 12																				
Desarrollo del Capítulo 13																				
Correcciones de Capítulos																				
Revisiones Finales																				

2^{do} Semestre de 2005

Actividades	Agosto				Septiembre				Octubre				Noviembre			
	I	II	III	IV	I	II	III	IV	I	II	III	IV	I	II	III	IV
Análisis de los requisitos																
Elaboración de propuesta para los manuales																
Revisión y elección del ISO 9000-3																
Establecimiento de puntos sobre seguridad																
Presentación e la propuesta de los manuales																
Presentación de los punto de ISO 9000-3 a utilizar																
Presentación de los puntos de seguridad a utilizar																
Desarrollo de los manuales de sistema y usuario																
Desarrollo de los puntos de ISO 9000-3 a utilizar																
Desarrollo de los puntos de seguridad a utilizar																
Mejoras al demo a entregar																
Mejoras al documento del manual																
Revisión de los puntos por parte de los tutores																
Entrega de los documentos y sustentación																

9. INGENIERIA DE REQUISITOS

9.1 ANALISIS DE REQUISITOS

9.1.1 CAPITULO 1. INTRODUCCION A LOS VIDEOJUEGOS.

En esta primera parte se plantea la importancia de los videojuegos en la sociedad actual como una herramienta de entretenimiento y avance tecnológico. El desconocimiento acerca de los temas referidos a los videojuegos causa que comúnmente se cometan errores al momento de hablar de los videojuegos subestimando la capacidad tecnológica desplegada en estos productos, las cuales no pueden ser menospreciadas ya que van a la vanguardia en el desarrollo de entornos 3D.

9.1.2 CAPITULO 2. INTRODUCCION A LOS MOTORES GRAFICOS.

La importancia de conocer con que tipo de herramienta se piensa desarrollar el manual se justifica en facilitar el rápido entendimiento del manual y familiarizar al lector con conceptos fundamentales que le serán útiles en la comprensión de la tecnología empleada para el desarrollo de los videojuegos. El reconocimiento de los distintos tipos de motores según su licencia dejara justificada la elección del motor en el que se basara el manual.

9.1.3 CAPITULO 3. REQUERIMIENTOS DE HARDWARE Y MOTORES GRAFICOS UTILIZADOS.

Cada juego requiere unas especificaciones de hardware diferentes entre si para que puedan ser ejecutados y se logre apreciar el potencial del juego. Este capitulo explicara la manera en que las distintas configuraciones de hardware afectan el rendimiento de los juegos y el motor o los motores que serán utilizados para el desarrollo de juegos basados en el manual.

9.1.4 CAPITULO 4. INTRODUCCION A LAS INTERFACES REQUERIDAS PARA EL DESARROLLO DE LOS VIDEOJUEGOS 3D.

En este capitulo encontraremos las diferentes interfaces requeridas para el excelente desarrollo de los juegos de videos de 3D. Muy importante es el reconocimiento de las características de cada herramienta que se utiliza en el desarrollo de un videojuego 3D, tales como el manejo del motor o los motores que se van a utilizar y las diferentes posibilidades de cada uno o las herramientas para la creación de gráficas de exteriores.

9.1.5 CAPITULO 5. RECONOCIMIENTO DEL AMBIENTE DE TRABAJO DEL MOTOR GRAFICO Y EL EDITOR DE EFECTOS EXTERIORES.

El objetivo de este capitulo es el de explicar el entorno de trabajo del Motor Gráfico y el Editor de Efectos Exteriores, como son las distintas interfaces (opciones, menús y comandos), además de todos los elementos con que se cuentan. Esto con el fin de facilitar el trabajo y hacer que la utilización del manual sea menos engorrosa, mediante el entendimiento de los diferentes elementos se consigue un trabajo más eficiente con el fin de agilizar el proceso.

9.1.6 CAPITULO 6. CREACION DE ENTORNOS 3D Y EDITOR DE NIVELES.

En todo juego de video se necesita diseño y creación de distintos niveles y entornos para así tener una división organizada en la cual podamos desplazarnos para contar con un orden en el desarrollo de un juego, y así tener aspectos que tengan semejanza con la realidad; pero en si el contenido de este capitulo se aprenderá y practicara la edición de niveles y entornos teniendo en cuenta las especificaciones de cada motor gráfico citado en este manual debido al hecho que cada motor presenta características disímiles para la creación de los escenarios.

9.1.7 CAPITULO 7. CREACION Y MANEJO DE OBJETOS SÓLIDOS.

En esta parte se dará a conocer el proceso de creación de los objetos sólidos que se observaran durante el transcurso de los Videojuegos, esta es una parte muy importante en juego ya que sin nada sólido en los niveles estos serían ambientes vacíos y no darían una

sensación agradable al usuario; además no solo es crear objetos uno tras otro sino que debe mantenerse una armonía que cree sensación de realismo, agregando texturas que mejoren el aspecto de los objetos.

9.1.8 CAPITULO 8. MANEJO DE CRISTALES, ESPEJOS Y LIQUIDOS.

En este capítulo se explicará todo lo concerniente a los efectos prácticamente decorativos de los escenarios como lo son cristales, espejos y líquidos; mediante estos distintos efectos se pueden embellecer los escenarios para que sean de gran calidad y atractivo al usuario.

9.1.9 CAPITULO 9. ILUMINACION, NIEBLA, CIELOS Y RAYOS.

La iluminación es lo más difícil de crear en un Videojuego, pero crearla no es lo difícil sino dar las sombras a los diferentes objetos que necesitan de este efecto; la niebla es muy sencilla y claro se debe utilizar en todos los juegos por la simple razón que no todos lo llevan, además se debe tener en cuenta que la niebla es bastante útil para crear eventos o sorpresas que den un contenido secuencial a los Videojuegos; es bastante importante conocer que el cielo de cada nivel es distinto de otro, por tanto se debe tener sumo cuidado en la escogencia de este efecto en particular; en cuanto al aspecto de los rayos, este es como un efecto agregado que sirve para dar un toque de realismo al escenario, e implantar un concepto de autenticidad a los Videojuegos.

9.1.10 CAPITULO 10. DISEÑO Y CREACION DE PERSONAJES.

Este capítulo es bastante importante ya que los personajes son parte fundamental en un Videojuego ya que estos deben interactuar con los distintos niveles de manera constante y por tanto deben ser distinguibles unos de otros; así como deben ser distintos, cada personaje tiene su función dentro de un Videojuego.

9.1.11 CAPITULO 11. MANEJO DE AMBIENTES EXTERIORES.

Los elementos exteriores son muy importantes, pues estos son los que se encargan de crear la sensación de profundidad más allá del nivel en el que se localizan los personajes, pero cabe destacar que los personajes no interactúan con estos ambientes, pues estos están ahí como complemento de los Videojuegos para crear una armonía.

9.1.12 CAPITULO 12. MANEJO DE EFECTOS SONOROS.

Cuando se habla de efectos sonoros se entiende al instante que se habla de los sonidos relacionados con cada acción que ocurre en el Videojuego. esto debe ocurrir igual que en la vida real, por tanto todo el que diseñe juegos de video debe tener en cuenta este aspecto todo el tiempo para que cree juegos de calidad teniendo en cuenta que con esto se desarrollan juegos de video en tiempo real.

9.1.13 CAPITULO 13. UNIFICACION DE COMPONENTES.

En esta instancia se procede a unificar todos los niveles, personajes y exteriores que contendrá un Videojuego, se debe dar un orden a la manera que se ejecutara el Videojuego y como se desenvolverá el jugador en el.

10. INGENIERIA DE INFORMACION

10.1 MISION

La Corporación Educativa Mayor del Desarrollo Simón Bolívar es una institución de Educación Superior, sin ánimo de lucro, no oficial y de utilidad común dedicada al proceso de formación integral en los campos de las ciencias, las humanidades y la tecnología; a la investigación científica y a la promoción del desarrollo cultural e ideológico de la sociedad, teniendo como funcionamiento el idcario Bolivariano de un ser ético, culto, autónomo y líder, constructor de una sociedad democrática, justa y solidaria.

En cumplimiento de nuestra función social propiciamos la actualización y universalización de los saberes, expresados con pertinencia, interdisciplinariedad, integralidad y flexibilidad curricular, teniendo como eje central la reflexión permanente de los procesos sociales y económicos de la realidad y el desarrollo del desarrollo regional, nacional y latinoamericano.

Con personal calificado y actualizado fomentamos procesos de autoevaluación permanente para el mejoramiento continuo de la calidad de nuestros servicios y promovemos activamente el bienestar y el desarrollo de nuestra comunidad educativa y social.

La Corporación Educativa Mayor del Desarrollo Simón Bolívar valora y cultiva la identidad, la cultura propia y el respeto del ancestro.

10.2 VISION

La Corporación Educativa Mayor del Desarrollo Simón Bolívar pretende incorporarse al futuro como una institución que forme líderes y dirigentes con conciencia nacional y latinoamericana, con responsabilidad ética que se identifique con el compromiso histórico del enriquecimiento espiritual e intelectual de la sociedad y el fortalecimiento de la identidad regional, nacional y latinoamericana en la conquista del sueño Bolivariano de una América unida y solidaria.

10.2.1 NUESTROS PRINCIPIOS

- ✓ **Dignidad Humana:** El hombre es el sujeto de la Historia y su característica es la capacidad de crear, trascender y sobreponerse a su estado de naturaleza. Esta condición de ser inteligente lo convierte en ser perfectible, preparado para construir el desarrollo de lo humano.

- ✓ **Autonomía:** Considerada la esencia de la Institución en tanto que hace posible la libertad de expresión y creación de programas de autogobierno; se define como la capacidad que tiene la Corporación Educativa Mayor del Desarrollo Simón Bolívar para darse su propia institucionalidad, expresada en su marco jurídico y su organización y cultura que permite proyectarse y lograr sus fines.

- ✓ **Integralidad:** Universalizar las distintas manifestaciones del saber científico, es decir una cohesión del hombre consigo mismo, con la sociedad, con la naturaleza y la cultura para lograr la pluridimensionalidad del saber.

- ✓ **Solidaridad:** Capacidad de comprender la historia personal y las condiciones de vida del otro, respetarlo y realizar con él acciones que promuevan su desarrollo social, político y económico. Exige de la institución la capacidad de ayuda y cooperación con la sociedad para que acceda a la educación superior.

- ✓ **Trabajo:** Concebido como derecho y obligación es una condición para el desarrollo de las facultades del hombre. En la institución se refiere al derecho y la obligación de todos los miembros de la comunidad educativa de contribuir al logro de la misión institucional y del propio proyecto de realización personal.

- ✓ **Legalidad:** La Corporación Educativa Mayor del Desarrollo Simón Bolívar recoge en su orgánico los principios, fines, fundamentos, propósitos, metas y lineamientos generales de la constitución política y ley colombiana.

10.2.2 NUESTROS VALORES

- ✓ **Justicia:** La dignidad del ser humano, como sujeto de la historia, se construye en ámbitos donde se garanticen las decisiones que se tomen y afecten su proyecto de vida, se fundamentan en la institucionalidad existente, de manera que se logre la equidad.

- ✓ **Ética:** Entendida como la naturaleza de sus actividades con estricto apego a los altos principios morales que se expresan en la consideración del bien como base de la felicidad individual y en la obtención plena de su compromiso con la sociedad colombiana, lo cual define su compromiso histórico.

- ✓ **Democracia:** Significa tener a todos los miembros de la comunidad educativa como interlocutores válidos, dentro de su esfera de competencia, para la toma y ejecución de la vida universitaria que afecten directamente sus intereses.

- ✓ **Participación:** Es el derecho y la obligación de ejercer la libertad, conjugada con el conocimiento para intervenir responsable, eficaz, eficiente y oportuna en la vida institucional.

- ✓ **Libertad de Cátedra y Aprendizaje:** La Corporación facilita a sus docentes el derecho de exponer sus ideas con el rigor de la seriedad científica. El estudiante a su vez goza de la garantía de expresar sus criterios, en la materia que sea objeto de análisis, todo esto bajo los límites del comportamiento adecuado y el respeto mutuo.

- ✓ **Convivencia:** Condición inherente a la acción de construir la humanidad del hombre. Requiere concertación en los modos de solución de conflictos, respeto por los acuerdos, canales de comunicación abiertos y transparentes, actitud de diálogo democrático, participativo y respetuoso de la diferencia de visiones, procurando la argumentación racional y el bienestar colectivo por encima de los intereses particulares.

10.3 HISTORIA

La CORPORACION EDUCATIVA MAYOR DEL DESARROLLO SIMON BOLIVAR fue fundada en el año 1972, por iniciativa del Dr. José Consuegra Higgins, escritor, economista, periodista, educador y científico de las ciencias sociales.

Contó con el apoyo y compañía de un grupo de académicos y educadores demócratas, a saber: los Doctores Álvaro Castro Socarrás, Leonello Marthe Zapata, doña Ana Bolívar de Consuegra, Eugenio Bolívar Romero, Carmen Alicia de Bolívar, Eduardo Pulgar Lemus(+), Jorge Artel Coneo(+), Leonor Palmera de Castro, Walter Suárez, Eusebio Consuegra Higgins(+), Sofanor Moré Redondo (+), con quienes se conformó la Sala General.

Sus labores académicas se iniciaron el 1 de marzo de 1973, con alumnos matriculados en las facultades de Economía, Sociología, Derecho, Trabajo Social, Arquitectura, Contaduría, Antropología, Recursos Naturales y Ciencias de la Educación, con énfasis en Sociales, Humanidades e Idiomas; Matemáticas y Física.

Siempre ha impulsado la formulación y difusión de “una teoría propia para el desarrollo económico y social de América Latina”. Asumió un compromiso con la Región Caribe, brindando oportunidades de acceso a la Educación Superior a todos los sectores sociales, especialmente los de origen popular.

Posteriormente, por recomendación del Instituto Colombiano para el Fomento de la Educación Superior, ICFES, la Corporación se dedicó específicamente a la enseñanza e investigación de las Ciencias Sociales y Humanas, con los programas de Economía, Sociología, Derecho, Trabajo Social y Educación, con énfasis en Ciencias Sociales.

Desde un comienzo, nuestra Alma Máter ha reconocido la investigación como actividad fundamental, orientada al estudio de la sociedad colombiana y latinoamericana; la solución de tensiones sociales y la comprensión e interpretación de las dimensiones cultural, ética y estética del ciudadano. Tiene un compromiso permanente con la Región Caribe Colombiana, brindando oportunidades de acceso a la Educación a todos los sectores sociales, especialmente los de origen popular.

A través de la extensión, la Corporación ha estado presente en las comunidades, participando en el análisis y búsqueda de soluciones, que contribuyen a mejorar la calidad de vida de sectores vulnerables, en Barranquilla, el Atlántico y el Caribe Colombiano, en general.

A partir de los años 90, el Rector Ejecutivo, Dr. José Consuegra Bolívar, ha impulsado el crecimiento y diversificación de la oferta académica. Se crean las facultades de Educación Básica Primaria, Administración de Empresas y Contaduría. Se consolidan el Instituto de Investigaciones y el Instituto de Postgrado y Educación Continua.

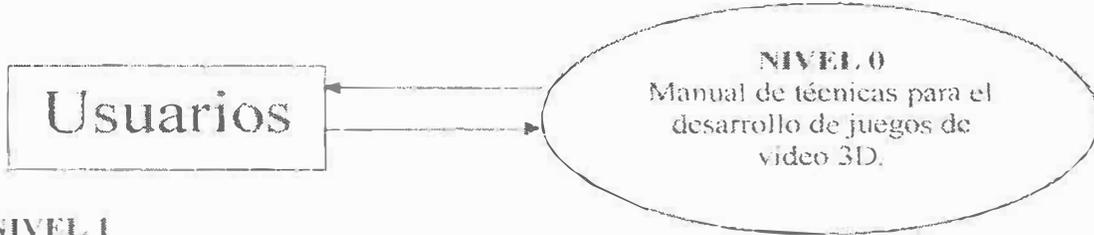
A partir de 1995, inician actividades académicas las facultades de Psicología, Fisioterapia, Enfermería, Ingeniería de Sistemas, Ingeniería Industrial e Ingeniería de Mercados. Una meta a corto plazo es abrir la Facultad de Medicina, para lo cual se siguen los Lineamientos del Consejo Nacional de Acreditación.

Los programas de Licenciatura en Educación Media con énfasis en Ciencias Sociales y Filosofía, Licenciatura en Educación Básica con énfasis en Humanidades y Lengua Castellana, Ingeniería de Sistemas e Ingeniería de Mercados, y las Especializaciones en Gestión de Proyectos Educativos y Pedagogía de las Ciencias, han obtenido el registro calificado. La facultad de Psicología, con el acompañamiento y evaluación del Consejo Nacional de Acreditación, adelanta los procesos de acreditación de alta calidad.

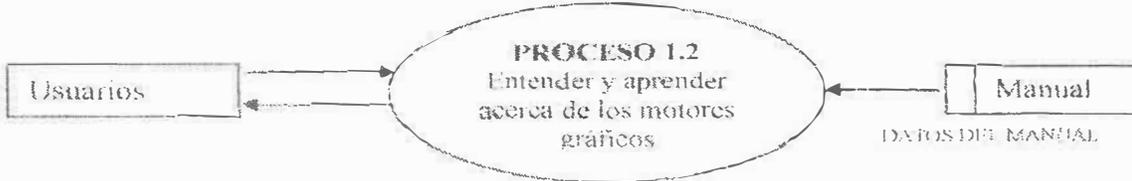
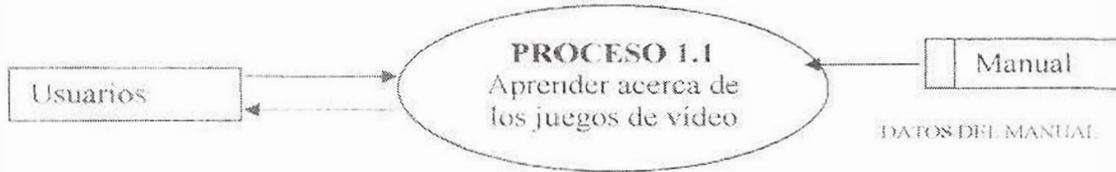
Los últimos años en nuestra Alma Máter se caracterizan por la organización y modernización de los Procesos Académicos y administrativos y la consolidación de una cultura de la calidad en cada una de sus dependencias y servicios.

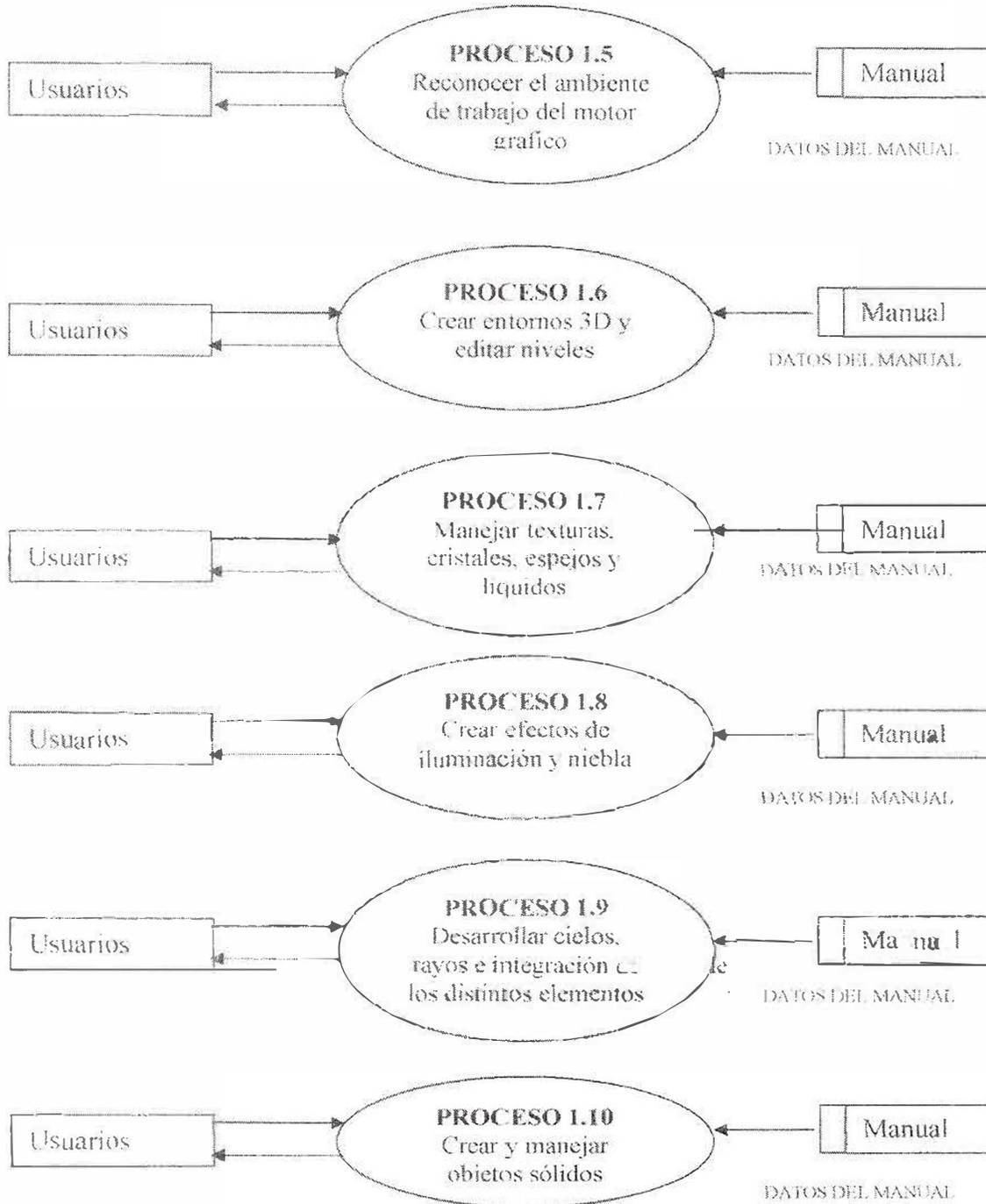
11 ANALISIS DEL SISTEMA.

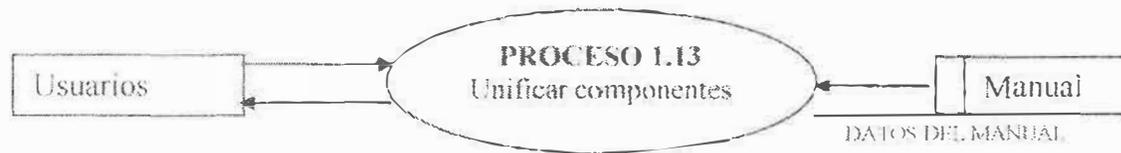
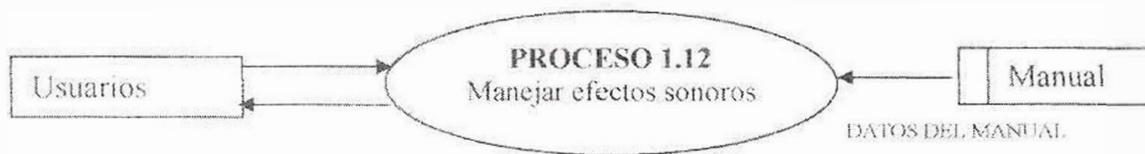
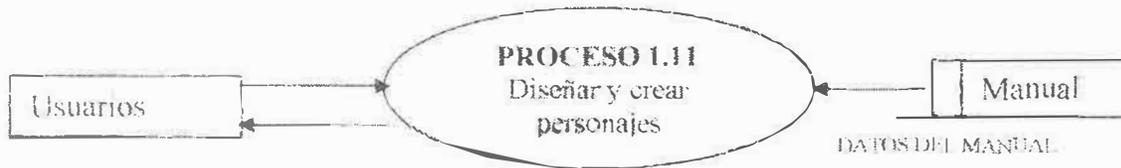
NIVEL 0



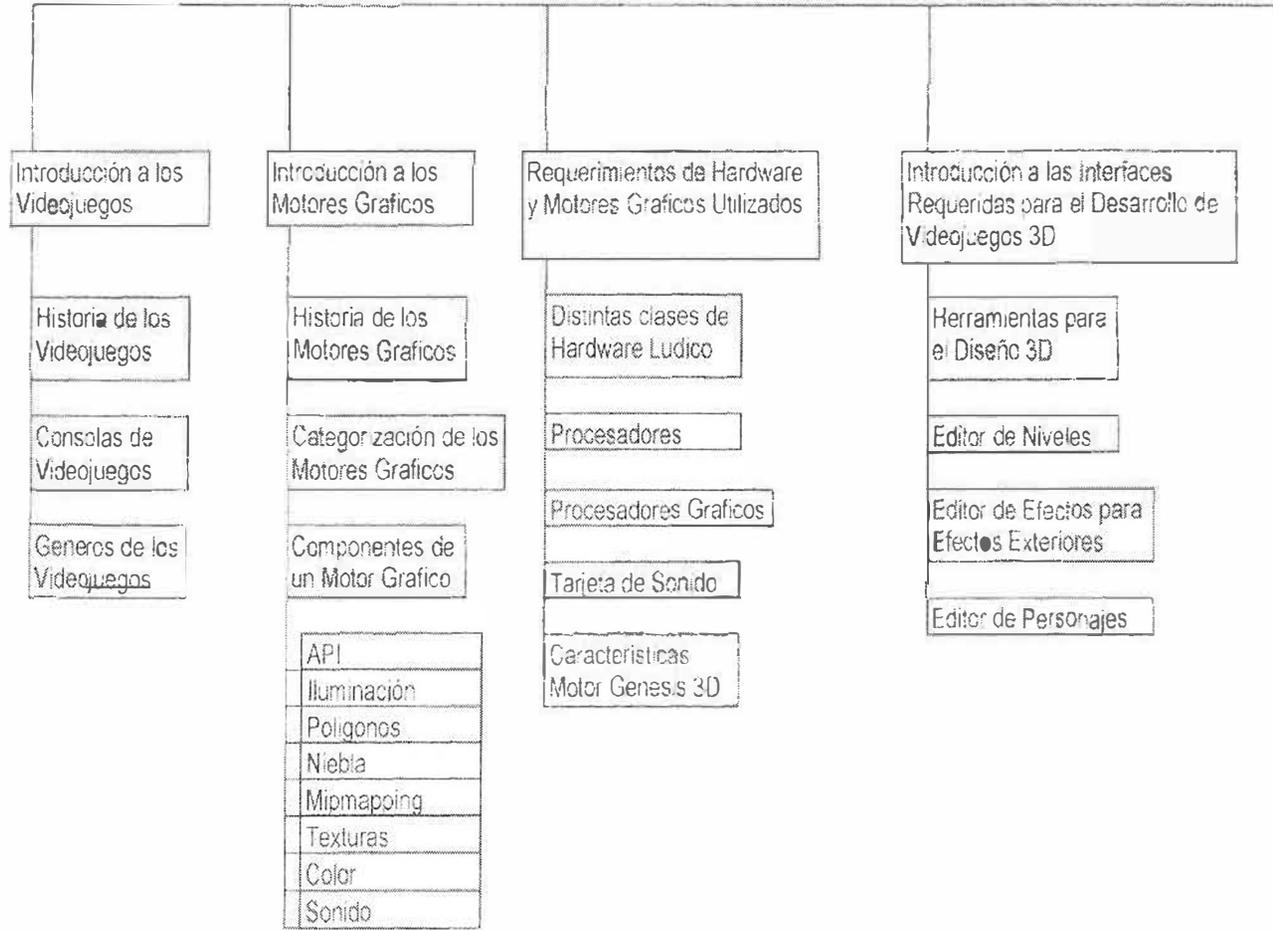
NIVEL 1

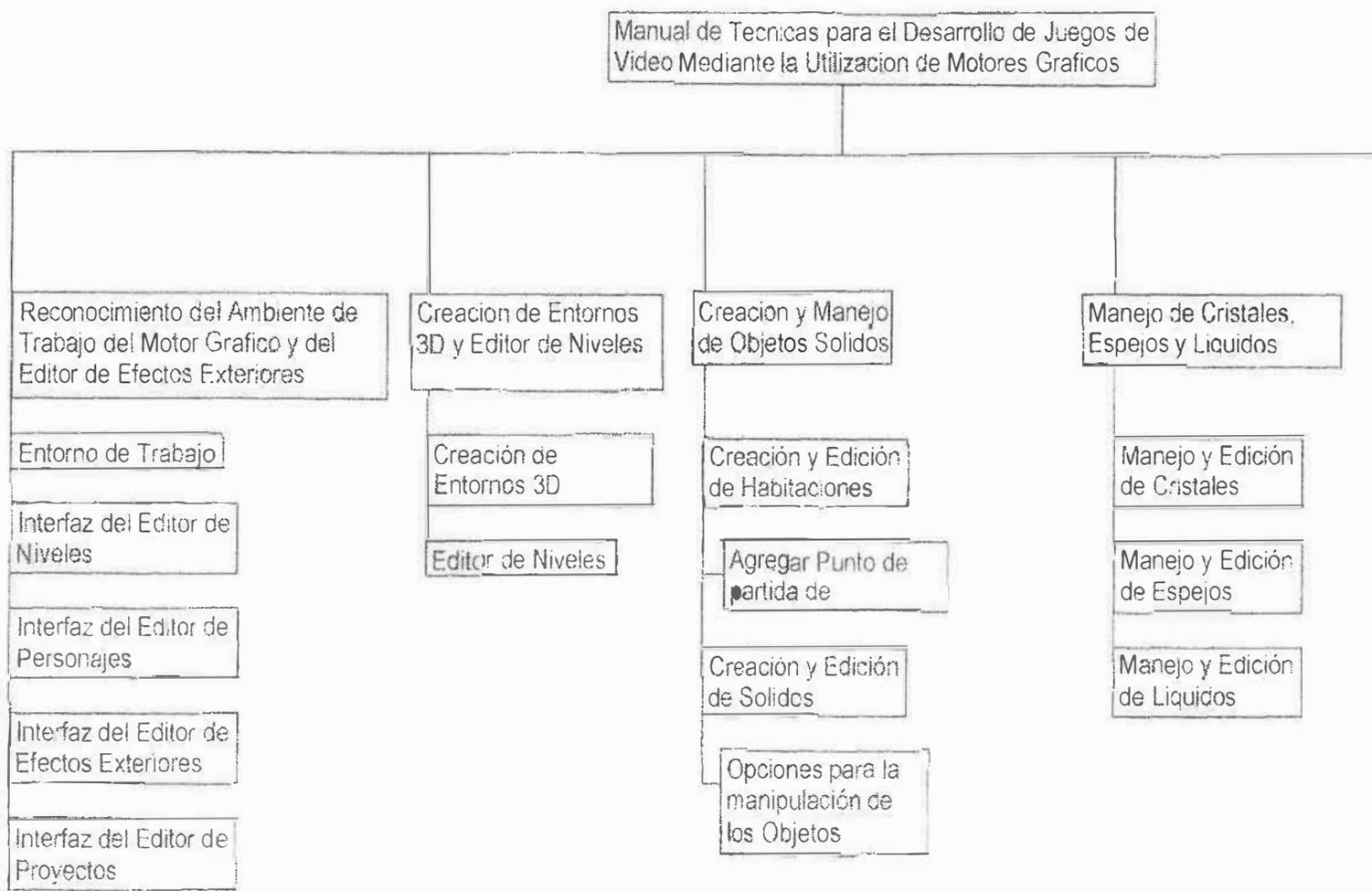


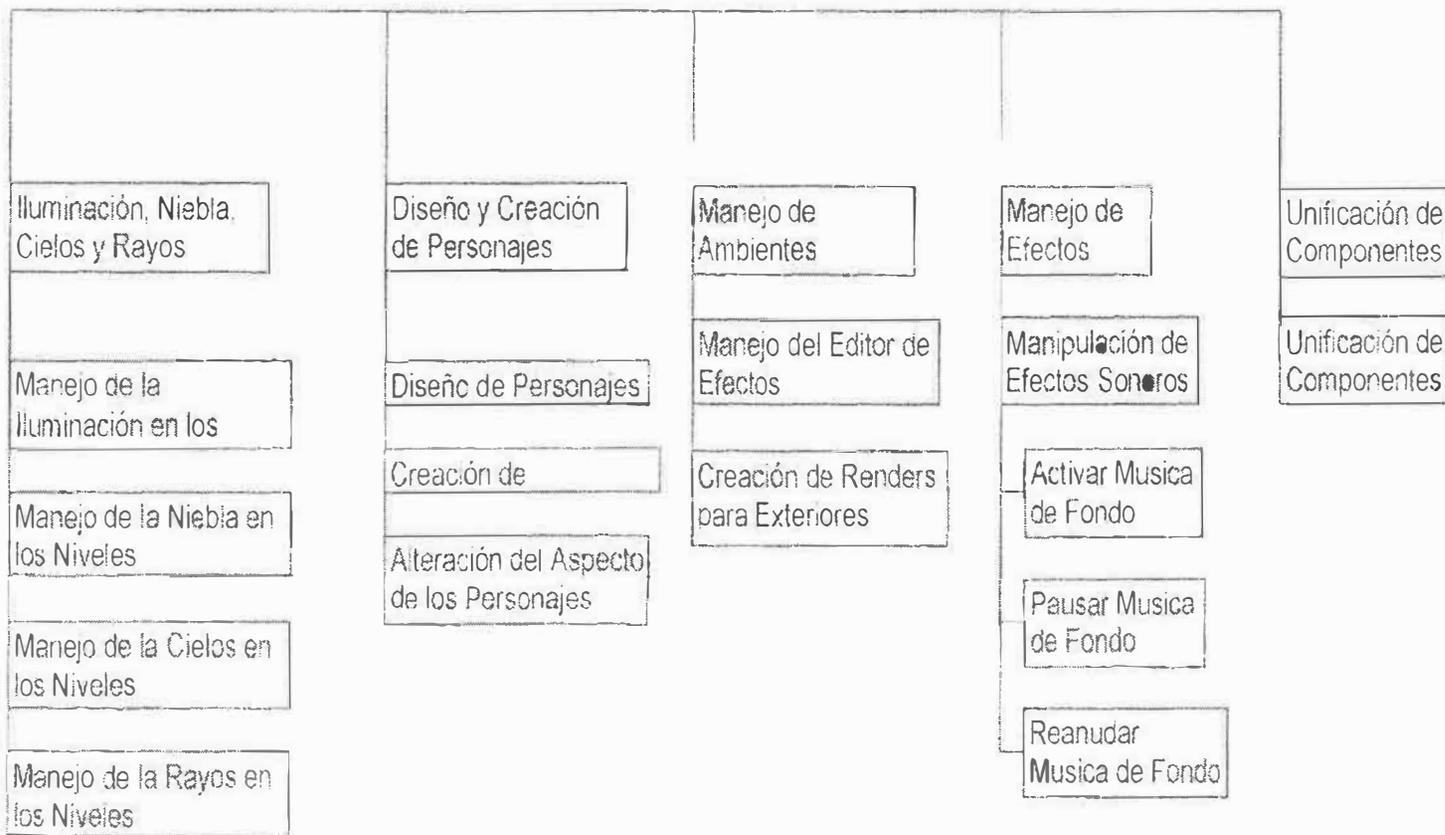




11.2 DIAGRAMA JERARQUICO FUNCIONAL.







12. DISEÑO DEL SISTEMA

12.2 DESCRIPCION DE LOS COMPONENTES DE LA ESTRUCTURA FUNCIONAL

12.2.1 CAPITULO 1. INTRODUCCION A LOS VIDEOJUEGOS.

1.1 HISTORIA DE LOS VIDEOJUEGOS:

La importancia de conocer el origen los videojuegos radica en saber el avance tecnológico que han tenido con el pasar de los años desde la aparición del Space Invaders (Primer Videojuego) hasta los modernos juegos 3D. En esta sección se dará a conocer la evolución de las consolas y del hardware para PC.

1.2 CONSOLAS DE VIDEOJUEGOS :

Especificación técnica de los modelos de consolas más importantes que han aparecido con el correr de los años desde los primeros modelos de 8 bits hasta los actuales de 128 bits, especificando cuales han contribuido en el avance tecnológico de estos dispositivos.

1.3 GENEROS DE LOS VIDEOJUEGOS:

Explicación de las distintas clases de Videojuegos existentes en la actualidad, las diferencias entre cada uno de los géneros nombrados en el presente capítulo y el impacto que tienen entre los aficionados a los Videojuegos.

12.2.2 CAPITULO 2. INTRODUCCION A LOS MOTORES GRAFICOS.

2.1 HISTORIA DE LOS MOTORES GRAFICOS:

El origen de los motores gráficos se hace imprescindible en el reconocimiento de la importancia que han tenido estas herramientas en el desarrollo de los videojuegos como un avance tecnológico que ha hecho que el desarrollar videojuegos sea una tarea más fácil.



2.2 CATEGORIZACIÓN DE LOS MOTORES GRAFICOS.

Reconocimiento de las distintas categorías de los motores gráficos de acuerdo a sus licencias es fundamental a la hora de pensar en comenzar a desarrollar un Videojuego, ya que hay que adaptarse al presupuesto que se tenga para el proyecto. En este apartado se explicara la diferencia entre los motores usados para consola y PC.

2.3 COMPONENTES DE UN MOTOR GRAFICO.

Explicación de los distintos componentes y apartados de los que se compone una herramienta de este tipo y que facilitara el entendimiento de la misma. Se dará una introducción a aspectos como la iluminación y las texturas.

2.3.1. API.

2.3.2. ILUMINACION.

2.3.3. POLIGONOS.

2.3.4. NIEBLA.

2.3.5. MIPMAPPING.

2.3.6. TEXTURAS.

2.3.7. COLOR.

2.3.8. SONIDO.

2.4 EFECTOS GRAFICOS.

Explicación de los efectos gráficos con los que trabajan los motores gráficos modernos y cuales son las distintas maneras en que afectan el rendimiento de los Videojuegos hechos con los motores.

2.4.1. ANTIALIASING.

2.4.2. BUMP MAPPING.

2.4.3. COMPRESIÓN DE TEXTURAS.

2.4.4. ENVIROMENT MAPPING.

2.4.5. MIPMAPPING (profundizado).

12.2.3 CAPITULO 3. REQUERIMIENTOS DE HARDWARE Y MOTORES GRAFICOS UTILIZADOS.

3.1 DISTINTAS CLASES DE HARDWARE LUDICO.

La importancia de conocer los distintos componentes de hardware que intervienen en la ejecución de un Videojuego es fundamental para conocer con que hardware interactúa el motor gráfico y en que manera afecta el rendimiento de los juegos.

3.2 PROCESADORES

La razón principal de entender como funciona un procesador es para estar mas familiarizado con el funcionamiento global de un sistema de computo, de esta manera se observa la interacción de este con los distintos componentes de Software como el Sistema Operativo Central, los demás programas y los distintos componentes de hardware, en fin el Procesador es quien interpreta las instrucciones dadas por el usuario.

3.3 PROCESADORES GRAFICOS.

Es imperante conocer la importancia de un Procesador Grafico pues estos son los encargados de realizar las diferentes labores que se especializan en el tratamiento de la información grafica que será proyectada en pantalla, por tanto esta es una parte vital del desarrollo de esta actividad (la proyección de imágenes) para quitar tarea del procesador central.

3.4 TARJETA DE SONIDO

Conociendo que todos los Videojuegos actuales tienen una gran variedad de sonidos debe saberse que el reconocimiento de las distintas capacidades de este dispositivo es importante para entender como afectan el desarrollo de un juego, pues este se encarga de interpretar los distintos sonidos producidos durante la visualización, manejo y desarrollo de los diferentes escenarios producidos en un videojuego.

3.5 CARACTERISTICAS MOTOR GENESIS 3D

Para que el proyecto de desarrollo de videojuegos sea prolijo debe contarse con un motor grafico que sea potente en el aspecto de manejo de los distintos efectos y herramientas alternas para la creación de Ambientes Gráficos 3D.

12.2.4 CAPITULO 4. INTRODUCCION A LAS INTERFACES REQUERIDAS PARA EL DESARROLLO DE VIDEOJUEGOS 3D.

4.1 HERRAMIENTAS PARA EL DISEÑO 3D.

Explicación de los distintas herramientas para el completo desarrollo de los Videojuegos como son el editor de niveles, el editor de exteriores, el editor de personajes entre otras que permitirán realizar el juego de manera más eficiente.

4.2 EDITOR DE NIVELES.

Se buscara poner en claro que es el editor de niveles y su importancia en el desarrollo de un Videojuego cualquiera, también se enfatizara en las facilidades de su uso para un mejor aprovechamiento de las capacidades gráficas del motor.

4.3 EDITOR DE EFECTOS PARA EFECTOS EXTERIORES.

Se esclarecerán los puntos a tener en cuenta para comprender que es el editor de niveles y como se integra y complementa el diseño de entornos de los que se compone un Videojuego.

4.4 EDITOR DE PERSONAJES.

Explicación de lo que es un Editor de Personajes y sus aspectos funcionales que permiten la integración con el motor grafico, para la inclusión de los distintos personajes que aparecen en el transcurso del desarrollo de un Videojuego.

12.2.5 CAPITULO 5. RECONOCIMIENTO DEL AMBIENTE DE TRABAJO DEL MOTOR GRAFICO Y DEL EDITOR DE EFECTOS EXTERIORES.

5.1 ENTORNO DE TRABAJO.

El entorno de trabajo para desarrollar un Videojuego según el manual ha desarrollar se basa en un Motor Gráfico como herramienta principal y un Editor de Efectos Exteriores para las vistas exteriores que complementan el desarrollo de cualquier Videojuego.

5.2 INTERFAZ DEL EDITOR DE NIVELES.

Se darán a conocer los distintos componentes de la herramienta en cuestión, como las opciones del menú, las distintas pestañas de la interfaz y los comandos con los que se va a interactuar para la creación de los niveles de una manera más eficiente.

5.3 INTERFAZ DEL EDITOR DE PERSONAJES.

En esta parte se mostraran las diferentes partes que contiene el programa a tratar, como lo son las distintas opciones del menú, las pestañas y los comandos con los que se trabaja para la creación, y posterior edición de personajes de una manera eficiente.

5.4 INTERFAZ DEL EDITOR DE EFECTOS EXTERIORES.

Se darán a conocer las distintas opciones con que cuenta el software en estudio, para el caso son el manejo de archivos, las vistas de los distintos aspectos manipulados por botones y pestañas, todo esto facilita el trabajo y lo hace más eficiente.

5.5 INTERFAZ DEL EDITOR DE PROYECTOS.

En esta parte se muestra como se fusiona un proyecto el cual consta de diferentes partes, entre las cuales están los personajes, los niveles, las vistas exteriores, para que así sea de un fácil manejo un paquete conjunto y de esta forma haya una correcta ejecución.

12.2.6 CAPITULO 6. CREACION DE ENTORNOS 3D Y EDITOR DE NIVELES.

6.1 CREACION DE ENTORNOS 3D.

Para crear Entornos 3D de calidad, se debe tener en cuenta que un ambiente claro, en el aspecto de su trama, para que el usuario final pueda desarrollar sus habilidades a pleno y entender que todo lo que esta a su alrededor es parte del juego e interactúa con el.

6.2 EDITOR DE NIVELES.

En esta sección se enseñara la manera de comenzar a trabajar con el editor de niveles y como comenzar a elaborar un Videjuego desde un concepto básico como es un nivel. Es muy importante el dar a conocer los distintos elementos que se deben tener en cuenta a la hora de elaborar los distintos niveles que vaya a contener un Videjuego.

12.2.7 CAPITULO 7. CREACION Y MANEJO DE OBJETOS SÓLIDOS.

7.1 CREACION Y EDICION DE HABITACIONES.

Este aspecto cubre todo lo referente a la creación de cuartos, los cuales tienen la particularidad de ser invisibles, es decir sus paredes, piso y techo no tienen textura alguna, el usuario o diseñador debe agregarlos, después de esto si se pueden agregar objetos sólidos que cuentan con textura o propiedad de sólido.

7.1.1 AGREGAR OBJETOS A UNA HABITACIÓN.

7.2 CREACION Y EDICION DE SÓLIDOS.

Estos son objetos que pueden ser de diferentes formas, para así crear niveles de gran variedad, personalizándoles de tal manera que sean apropiados para distintos Videojuegos de los Géneros existentes.

7.2.1 OPCIONES PARA LA MANIPULACION DE LOS OBJETOS.

12.2.8 CAPITULO 8. MANEJO DE CRISTALES, ESPEJOS Y LIQUIDOS.

8.1 MANEJO Y EDICION DE CRISTALES.

Son un aspecto importante a la hora de personalizar un nivel, pues pueden cumplir diferentes propósitos como el caso de ventanas, vitrinas, etc.

8.2 MANEJO Y EDICION DE ESPEJOS.

Se considera como un aspecto destacado al momento de decorar un nivel, pues son muy versátiles en sus diferentes propiedades.

8.3 MANEJO Y EDICION DE LIQUIDOS.

Este es un aspecto a tener en cuenta a la hora de añadir los distintos líquidos que pueden encontrarse en los niveles de cualquier Videojuego, ya que son elementos que pueden afectar el desenvolvimiento del jugador en el Videojuego.

12.2.9 CAPITULO 9. ILUMINACION, NIEBLA, CIELOS Y RAYOS.

9.1 MANEJO DE LA ILUMINACION EN LOS NIVELES.

En esta sección se dará a conocer la importancia de la iluminación en el diseño de los Videojuegos como aspecto esencial para otorgarle mayor realismo a los distintos objetos que se encuentran en los distintos niveles.

9.2 MANEJO DE LA NIEBLA EN LOS NIVELES.

Para esta parte debe tenerse en cuenta que la niebla se acopla a algunas situaciones, en las cuales se busca trabajar un(os) acontecimiento(os) que habrá(n) de ocurrir en el transcurso de un Videojuegos.

9.3 MANEJO DE LOS CIELOS EN LOS NIVELES.

Este es un aspecto muy importante, pues de allí depende mucha de la personalidad de un buen Videojuego ya que el cielo puede variar dependiendo del lugar en el que se encuentre(n) lo(s) personaje(s).

9.4 MANEJO DE LOS RAYOS EN LOS NIVELES:

Al hablar de rayos debe tenerse en cuenta que no es mas que un efecto, que sirve para personalizar los niveles, estos pueden ser personalizados en sus propiedades.

12.2.10 CAPITULO 10. DISEÑO Y CREACION DE PERSONAJES.

10.1 DISEÑO DE PERSONAJES.

En esta parte se debe tener en cuenta que cada personaje tiene sus propias características, por tanto hay que marcar una diferencia en el diseño de cada personaje.

10.2 CREACION DE PERSONAJES.

Al momento de crear personajes se debe hacer pie sobre el aspecto de unificar los personajes de una manera armónica para crear



10.3 ALTERACION DEL ASPECTO DE LOS PERSONAJES.

En este apartado se deben tener en cuenta el uso de programas de edición gráfica para la modificación de aspectos como el color de la vestimenta o las características físicas de los distintos personajes.

12.2.11 CAPITULO 11. MANEJO DE AMBIENTES EXTERIORES.

11.1 MANEJO DEL EDITOR DE EFECTOS.

Se debe tener especial atención a los distintos efectos que se pueden crear con esta herramienta, que sirve de complemento al trabajo hecho con el Motor Gráfico para lograr mayor realismo.

11.2 CREACIÓN DE RENDERS PARA EXTERIORES.

La creación de los distintos efectos no tiene razón si no se integran en un solo componente que pueda añadirse a los distintos niveles para desarrollar Videojuegos más atractivos para los jugadores.

12.2.12 CAPITULO 12. MANEJO DE EFECTOS SONOROS.

12.1 MANIPULACION DE EFECTOS SONOROS.

En este campo hay que tener en cuenta que hay muchas diferentes acciones que pueden ser realizadas por los personajes, o por algún objeto en particular, por tanto se debe prestar especial atención al momento de manipular efectos sonoros.

12.2.13 CAPITULO 13. UNIFICACION DE COMPONENTES.

13.1 UNIFICACION DE COMPONENTES.

Al finalizar la Fase de Diseño de un Videojuego es bastante importante entregar a los Usuarios finales un paquete para facilitar a ellos la manipulación de los Videojuegos.

ANEXOS

ANEXO No. 1 ENCUESTA

1. ¿Que son para usted los Videojuegos?

- a. Un medio de entretenimiento.
- b. Un método para desarrollar agilidad mental.
- c. Una actividad que no origina nada productivo.
- d. Otro.

2. ¿Le gustan los Videojuegos?

- a. Si.
- b. No.
- c. No Sabe, No Responde.

3. ¿Qué plataforma prefiere para los Videojuegos?

- a. Para PC.
- b. Para Consola (PSX, PS2, Dreamcast, Neo-Geo, etc).
- c. Consolas Portátiles (GameBoy, N-Gage, PSP).
- d. No Sabe, No Responde (Ninguna de las Anteriores).

4. ¿Qué le atrae de los Videojuegos?

- a. El Contenido (Las diferentes opciones).
- b. El Ambiente Grafico.
- c. Los Efectos Sonoros.
- d. Otro (Ninguna de las Anteriores).

5. ¿Qué Género de Videojuegos prefiere?

- a. Rol y Aventura.
- b. Estrategia.
- c. Deportes.
- d. Acción.
- e. Lucha.
- f. Simuladores.
- g. Ninguna de las Anteriores.

6. ¿Se interesa en como son desarrollados los Videojuegos?

- a. Si.
- b. No.
- c. No Sabe, No Responde.

7. ¿Conoce el concepto y/o la diferencia entre 2D y 3D?

- a. Si.
- b. No.
- c. No Sabe, No Responde.

8. ¿Le interesaría aprender técnicas de diseño 3D en la USB?

- a. Si.
- b. No.
- c. No Sabe, No Responde.

9. ¿Ha escuchado acerca de Herramientas de Diseño 3D y/o Motores Gráficos?

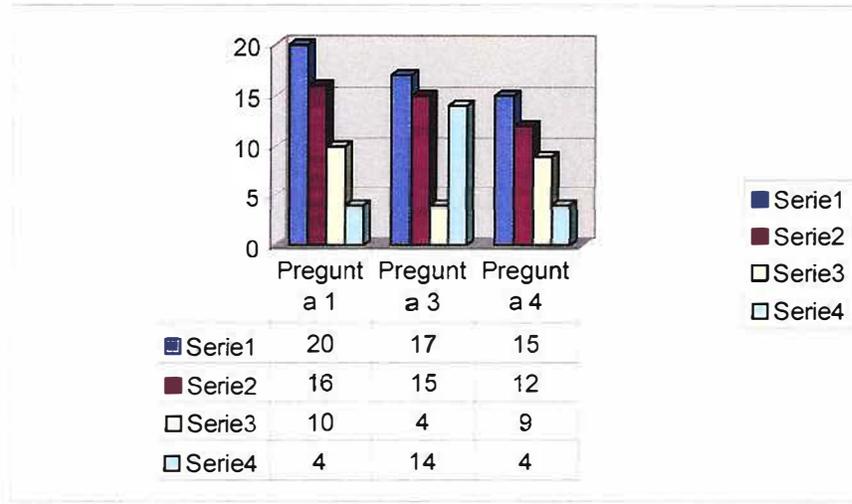
- a. Si.
- b. No.
- c. No Sabe, No Responde.

10. ¿En que cree que se emplean las Herramientas de Diseño 3D?

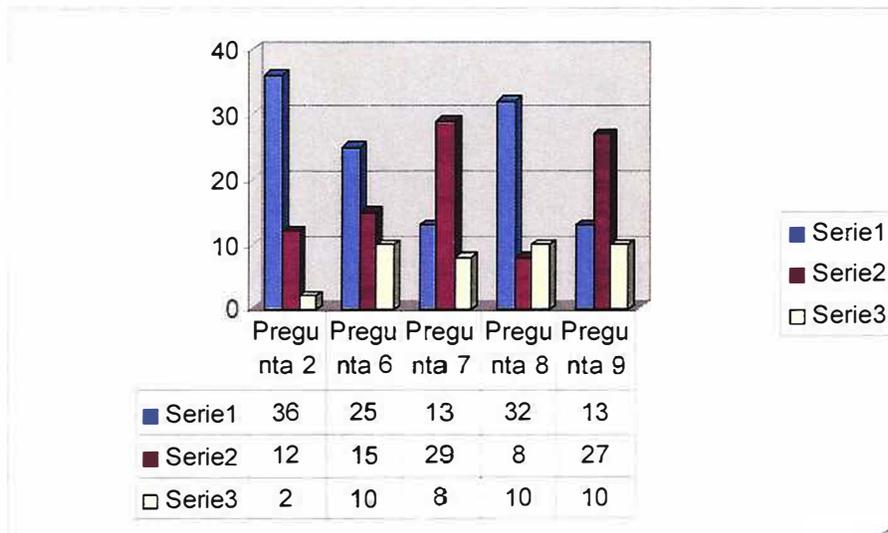
- a. Utilización en diseño de interfaces para aplicaciones.
- b. Programando.
- c. Diseño de Videojuegos.
- d. Todas las anteriores.
- e. Otro.

**ANEXO No. 2
RESULTADOS DE LA ENCUESTA**

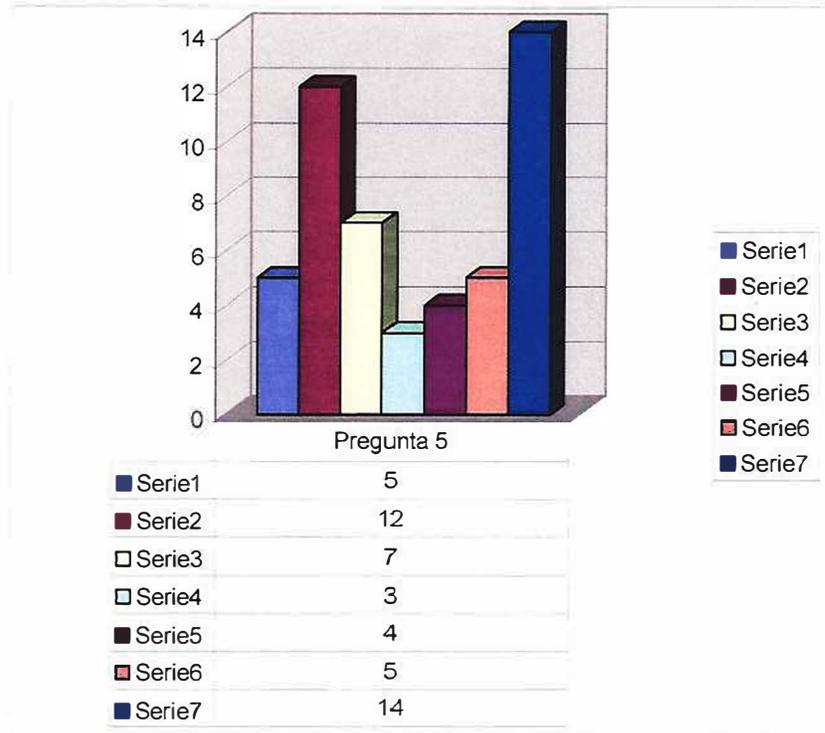
Tabulación Concerniente a las preguntas 1, 3, 4.



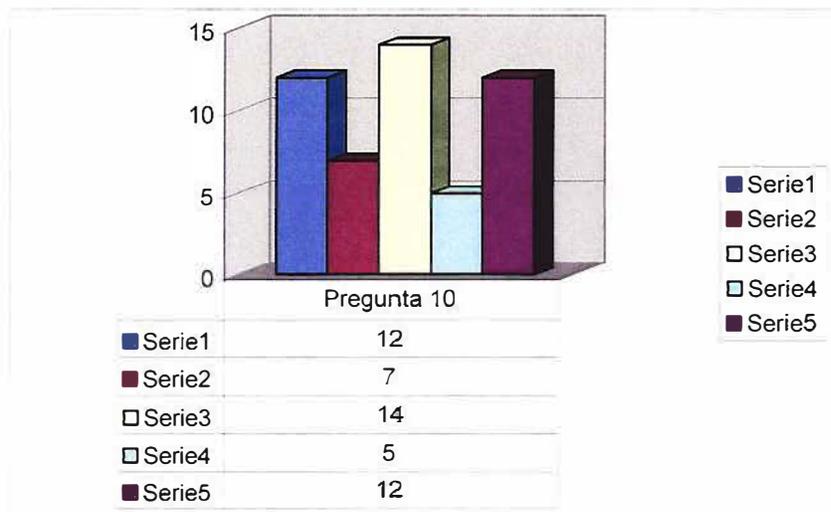
Tabulación Concerniente a las preguntas 2, 6, 7, 8, 9.



Tabulación Concerniente a la pregunta 5.



Tabulación Concerniente a la pregunta 10.



ISO 9000-3 Normas Para La Gestión Y Aseguramiento De La Calidad.	Implementación De Un Manual De Técnicas Para El Desarrollo De Juegos De Video Mediante La Utilización De Motores Gráficos 3D.
Versión 1.0	Revisado Por El Ing. Alberto Redondo

ANEXO No. 3

INTRODUCCION

En estos días "calidad" es la palabra de más relevancia, los consumidores esperan productos de calidad para satisfacer sus necesidades, solucionar sus problemas y obtener beneficios. Sin embargo dentro de la industria del software, "calidad" no ha sido el fuerte de la rama.

Las tres fallas dominantes que existen dentro de la industria del software son los altos costos en cuanto a depuración de un sistema, tiempo perdido en la corrección del sistema (estamos de acuerdo en que no existirían estas deficiencias si se hubiera realizado un análisis a conciencia del sistema), y la falla de conocer todas las necesidades del usuario.

Hoy en día la industria del software está implementando modelos para mejorar sus operaciones y corregir sus fallas. La expectativa es colocar el desarrollo de software bajo un control estadístico para verificar cuáles son las actividades repetitivas que continuamente se tienen que programar, y que producen exactamente el mismo resultado. Así, los procesos exitosos utilizados anteriormente pueden ser modelos base para la planeación de proyectos futuros, optimizando costos, incrementando la eficiencia y la productividad, desarrollando mejores productos de calidad y por consecuencia, generando más beneficios para la empresa.

Uno de estos modelos base son las normas estándares de calidad ISO 9000 que en especial han creado un interés masivo para la industria de software a causa de su aceptación a nivel internacional de muchas compañías importantes.

En el mundo de los Videojuegos se toma muy en cuenta el concepto de Calidad, ya que siempre se busca el mejoramiento continuo de los productos que se lanzan al mercado. Cada producto que se lanza pasa por un exhaustivo periodo de pruebas, el cual se realiza de manera controlada.

Los videojuegos han tenido un desarrollo vertiginoso en los últimos años llegando a ser una industria que genera millones de dólares al año y llegando a tener un impacto enorme en la sociedad actual, como medio de entretenimiento para personas de todas las edades y

ISO 9000-3 Normas Para La Gestión Y Aseguramiento De La Calidad.	Implementación De Un Manual De Técnicas Para El Desarrollo De Juegos De Video Mediante La Utilización De Motores Gráficos 3D.
Versión 1.0	Revisado Por El Ing. Alberto Redondo

ISO 9000-3 Normas Para La Gestión Y Aseguramiento De La Calidad.	Implementación De Un Manual De Técnicas Para El Desarrollo De Juegos De Video Mediante La Utilización De Motores Gráficos 3D.
Versión 1.0	Revisado Por El Ing. Alberto Redondo

llevando el desarrollo de los videojuegos a un nivel cultural que rivalizaría con campos como el cine y la música, debido a que los videojuegos logran desarrollar la lucidez mental, los reflejos y la agilidad psíquica.

“Todo comenzó en el año de 1961, cuando Steve Russell creó un videojuego llamado Space Invaders solo para demostrar la capacidad de las computadoras de la época y nunca se pensó en él como una oportunidad de aprovechamiento comercial”. En el año de 1973 Nolan Bushnell funda la compañía ATARI, lanzando un aparato lógico llamado Pong. A partir de aquí y hasta mediados de los 80s las consolas de videojuegos fueron fabricadas exclusivamente por empresas estadounidenses y usaban procesadores de 8 bits. En 1983 la empresa NINTENDO lanza su consola Famicom (NES) en Japón y dos años después en Estados Unidos logrando un gran impacto comercial, seguida por SEGA. A partir de ese instante las compañías japonesas tomarían el liderazgo del mercado. A principios de los 90s se lanzan las consolas de 16 bits como el SNES, el SEGA Genesis o el Neo – Geo de SNK, y años más tarde entraría al mercado SONY con el sistema Playstation siendo el primer aparato casero de mostrar gráficos 3D poligonales. En los últimos años surgieron sistemas de 64 bits y después los sistemas de 128 bits como el Dreamcast de SEGA, el Playstation2 de SONY o el Gamecube de NINTENDO.

Los videojuegos en los computadores han tenido un desarrollo diferente al de los diseñados para consolas debido a que los juegos de consola se desarrollan con motores gráficos privados (los cuales no salen a la venta debido a que cada empresa desarrolladora tiene sus propios motores). En cambio los motores desarrollados para juegos de computadora son motores públicos (estos pueden ser adquiridos por compañías diferentes a la compañía creadora del motor), y siguen desarrollos diferentes.

“Hace 8 años el motor más popular era el motor Doom el cual era muy limitado porque el personaje no podía mirar hacia arriba ni hacia abajo. Un año después se lanzó el motor Duke Nuke 3D que limitadamente permitía mirar hacia arriba y abajo. 3 años más tarde aparece el motor Quake el cual ofrece niveles basados en polígonos, dando libertad para mirar en todas las direcciones. 2 años más tarde los motores Unreal y Quake II introducen efectos más reales como halos alrededor de las luces, niebla y luces de colores entre otros”. Después surgiría el motor Quake III, logrando mayor realismo. Seguido a su vez por los motores Unreal II y el Doom III.

ISO 9000-3 Normas Para La Gestión Y Aseguramiento De La Calidad.	Implementación De Un Manual De Técnicas Para El Desarrollo De Juegos De Video Mediante La Utilización De Motores Gráficos 3D.
Versión 1.0	Revisado Por El Ing. Alberto Redondo

ISO 9000-3 Normas Para La Gestión Y Aseguramiento De La Calidad.	Implementación De Un Manual De Técnicas Para El Desarrollo De Juegos De Video Mediante La Utilización De Motores Gráficos 3D.
Versión 1.0	Revisado Por El Ing. Alberto Redondo

OBJETIVOS

- ✓ Se entienden claramente los aspectos de calidad que se aplican al manual.
- ✓ Entrega de los formatos utilizados en las pruebas que se realizan.
- ✓ Control de la documentación que se va entregarán.

ISO 9000-3 Normas Para La Gestión Y Aseguramiento De La Calidad.	Implementación De Un Manual De Técnicas Para El Desarrollo De Juegos De Video Mediante La Utilización De Motores Gráficos 3D.
Versión 1.0	Revisado Por El Ing. Alberto Redondo

ISO 9000-3 Normas Para La Gestión Y Aseguramiento De La Calidad.	Implementación De Un Manual De Técnicas Para El Desarrollo De Juegos De Video Mediante La Utilización De Motores Gráficos 3D.
Versión 1.0	Revisado Por El Ing. Alberto Redondo

CONSOLAS Y MOTORES GRAFICOS.

El apartado técnico y conceptual que abarca el desarrollar un videojuego es fundamental para entender este campo de desarrollo que se ha abierto en los últimos años. La magnitud con la que ha crecido el desarrollo de videojuegos ha generado un avance tanto en el hardware como el software dirigido para este campo, motivando una carrera para el desarrollo de nuevos juegos que aparecen cada mes, moviendo una industria que cada día crece más.

Las consolas de videojuegos son máquinas destinadas a la representación de juegos en el aparato de televisión. Para su uso se deben conectar a la entrada de video y audio del TV, disponen de controles diseñados ergonómicamente para el manejo de la consola disponiendo de un pad o un joystick. Los juegos disponibles para las consolas son muy disímiles de los desarrollados para computadora en cuanto a su temática y su desarrollo gráfico. Generalmente los juegos son desarrollados con motores gráficos privados.

Las consolas son el fundamento de los videojuegos porque fueron los primeros dispositivos de hardware que soportaron juegos de video con fines comerciales y han sido las que han ido a la vanguardia en el gráfico y temático de los juegos, llevando el desarrollo de estos a los límites de la tecnología debido a que al salir al mercado nuevos modelos de consolas estas superan de inmediato la capacidad de los computadores los cuales llegan a la capacidad de las consolas solo tres o cuatro años de haber salido la generación de consolas vigente.

El tener claro la importancia de las consolas es fundamental en el desarrollo de los videojuegos pero debido a la poca información sobre la programación de estas se dificulta el poder desarrollar juegos para estas plataformas y no da más opción que desarrollarlos para los computadores personales, debido a que es una plataforma más común entre los posibles usuarios.

“Para desarrollar videojuegos se puede recurrir a distintas herramientas, pero la más eficiente y robusta es el Motor Gráfico. El Motor Gráfico o motor de juegos es una aplicación que ejecuta todos los aspectos de un juego, ósea que las líneas de código del motor son las encargadas de generar todos los elementos que aparecen ante la vista del

ISO 9000-3 Normas Para La Gestión Y Aseguramiento De La Calidad.	Implementación De Un Manual De Técnicas Para El Desarrollo De Juegos De Video Mediante La Utilización De Motores Gráficos 3D.
Versión 1.0	Revisado Por El Ing. Alberto Redondo

ISO 9000-3 Normas Para La Gestión Y Aseguramiento De La Calidad.	Implementación De Un Manual De Técnicas Para El Desarrollo De Juegos De Video Mediante La Utilización De Motores Gráficos 3D.
Versión 1.0	Revisado Por El Ing. Alberto Redondo

usuario, también calcula la cantidad de polígonos que se deben usar para la construcción de todos los elementos visualizados”⁷.

“Este programa sirve para crear juegos diferentes en género y apariencia pero que estarían montados sobre la misma plantilla. Dado que esta industria mueve muchos millones de dólares, los fabricantes no pueden darse el lujo de esperar mucho tiempo para lanzar un nuevo título”.

“Los motores gráficos se pueden catalogar en tres categorías.”

1. Motores Privados: “son aquellos usados por una empresa en particular, ósea ellos mismos lo desarrollan y lo utilizan para sus juegos exclusivamente. Este tipo de motores no están a la venta comercial y preferiblemente las empresas que adoptan esta políticas son las japonesas. Ej. Capcom, Square-Enix, Konami, Sega, SNK, etc.”.
2. Motores Comerciales: “este tipo es usado por las empresas norteamericanas y europeas son motores desarrollados por empresas de juegos pero después colocan a la venta tal es el caso del motor Quake, Unreal, Doom3, etc. El precio de estos motores oscila entre los 200.000 y los 500.000 dólares”.
3. Motores Gratuitos: “son aquellos que se rigen por las licencias GNU o GPL”, por lo cual se hace viable obtener un motor de estos ya que es útil para propósitos académicos.

La importancia de los motores gráficos radica en que permiten crear juegos en menor tiempo, en cambio si se comenzaran a crear todas las plantillas de la nada seria un proceso muy lento y agotador, debido a que se tendría que desarrollar el motor gráfico llegando a ser una tarea muy dispendiosa y para una persona con poca experiencia podría llegar a ser inacabable. Aquí radica la importancia de un motor gráfico ya que solo los desarrolladores tendrían que concentrarse en realizar los juegos. Las distintas clases de motores se pueden observar como el interés comercial que tienen los videojuegos ya que los motores privados y públicos.

⁷. Revista Enter No. 34 abril 2001.

ISO 9000-3 Normas Para La Gestión Y Aseguramiento De La Calidad.	Implementación De Un Manual De Técnicas Para El Desarrollo De Juegos De Video Mediante La Utilización De Motores Gráficos 3D.
Versión 1.0	Revisado Por El Ing. Alberto Redondo

ISO 9000-3 Normas Para La Gestión Y Aseguramiento De La Calidad.	Implementación De Un Manual De Técnicas Para El Desarrollo De Juegos De Video Mediante La Utilización De Motores Gráficos 3D.
Versión 1.0	Revisado Por El Ing. Alberto Redondo

PROCEDIMIENTOS

Para la realización de cada prueba se debe hacer un procedimiento ordenado y riguroso el cual no se debe tomar a la ligera. Las pruebas que se realizan a una documentación corresponde al estado de esta, su correcta redacción, la estructura de las ideas que se expresan en el texto, etc. Para llegar al desarrollo de las pruebas se ha de analizar la estructura del documento para poder tener una prueba que satisfaga el estándar ISO 9000-3.

CONTROL DE DOCUMENTACION

En el control de la documentación se buscan personas especialistas en la redacción de documentos, preferiblemente licenciados en idiomas ya que se garantiza una evaluación exhaustiva y confiable de la documentación. A las personas seleccionadas para la realización de la prueba se les confió la realización de la prueba fueron tomadas en cuenta debido a su experiencia en el área de idiomas, esto garantiza la alta confiabilidad de la prueba.

PRUEBAS SOBRE EL DEMO

Para el Demo las pruebas se realizaron en un ambiente controlado; en el cual los creadores del manual se encuentran presentes. A los testers se les indico el modo de juego y los parámetros del Demo. La prueba se realizo en equipos con tarjeta de video de 8 megas, con 256 MB de RAM, un procesador de 800 MHz; la funcionalidad del Demo satisfizo a los testers dando su aprobación sobre el Demo.

CAPACITACION

Las personas que probaron el Demo expresaron su conformidad por la manera en que se les dio la inducción, ya que durante este proceso contaron con el manual del usuario y sistema del proyecto, basándose en la información consignada en estos se familiarizaron con el entorno de juego, después de la ejecución del demo fueron entrevistados por los creadores del manual para despejar posibles dudas, por ultimo los probadores llevaron a cabo la creación de un pequeño nivel.

ISO 9000-3 Normas Para La Gestión Y Aseguramiento De La Calidad.	Implementación De Un Manual De Técnicas Para El Desarrollo De Juegos De Video Mediante La Utilización De Motores Gráficos 3D.
Versión 1.0	Revisado Por El Ing. Alberto Redondo

ISO 9000-3 Normas Para La Gestión Y Aseguramiento De La Calidad.	Implementación De Un Manual De Técnicas Para El Desarrollo De Juegos De Video Mediante La Utilización De Motores Gráficos 3D.
Versión 1.0	Revisado Por El Ing. Alberto Redondo

FORMATO I				
EJECUCIÓN DEL DEMO				
Fecha:		Descripción:		
i. Reporte:		_____		
ii. Revisor:		_____		
iii. Analista:		_____		
Actividad	Si	No	No aplica	Información Adicional
¿El apartado de gráfico del demo presenta ralentizaciones?				
¿Al acercarse el personaje controlado a alguna superficie se presentan pixelizaciones?				
¿El control sobre el personaje es eficiente?				
¿Se requiere de destreza preliminar para poder jugar?				
¿Da la sensación de encontrarse en un ambiente 3D?				
¿Los efectos sonoros van acorde con las acciones que realiza el personaje?				

FORMATO II				
DOCUMENTACION				
Fecha:		Descripción:		
iv. Reporte:		_____		
v. Revisor:		_____		
vi. Analista:		_____		
Actividad	Si	No	No aplica	Información Adicional
¿Los documentos están redactados correctamente?				
¿Las ideas expresadas tienen coherencia?				
¿Los documentos se prestan para ser leídos de una manera secuencial?				
¿Las imágenes que se encuentran en los documentos sirven de guía para la comprensión de las ideas expresadas?				

ISO 9000-3 Normas Para La Gestión Y Aseguramiento De La Calidad.	Implementación De Un Manual De Técnicas Para El Desarrollo De Juegos De Video Mediante La Utilización De Motores Gráficos 3D.
Versión 1.0	Revisado Por El Ing. Alberto Redondo

ISO 9000-3 Normas Para La Gestión Y Aseguramiento De La Calidad.	Implementación De Un Manual De Técnicas Para El Desarrollo De Juegos De Video Mediante La Utilización De Motores Gráficos 3D.
Versión 1.0	Revisado Por El Ing. Alberto Redondo

FORMATO III				
CAPACITACION				
Fecha:		Descripción:		
vii. Reporte:		_____		
viii. Revisor:		_____		
ix. Analista:		_____		
Actividad	Si	No	No aplica	Información Adicional
¿Las personas que prueban el demo, son capaces de desenvolverse de manera óptima en el entorno de juego?				
¿Los jugadores reciben acompañamiento por parte de los creadores del manual y el demo?				
¿Se capacitan de manera adecuada a los probadores?				
¿Los lectores del manual son capaces de crear un juego?				

ISO 9000-3 Normas Para La Gestión Y Aseguramiento De La Calidad.	Implementación De Un Manual De Técnicas Para El Desarrollo De Juegos De Video Mediante La Utilización De Motores Gráficos 3D.
Versión 1.0	Revisado Por El Ing. Alberto Redondo

IMPLEMENTACIÓN DE UN MANUAL DE TÉCNICAS PARA EL
DESARROLLO DE JUEGOS DE VIDEO MEDIANTE LA UTILIZACIÓN DE
MOTORES GRAFICOS 3D PARA LOS ESTUDIANTES DE LA FACULTAD DE
INGENIERIA DE SISTEMAS EN LA CORPORACION EDUCATIVA MAYOR
DEL DESARROLLO SIMÓN BOLÍVAR

BARRÓS ERNESTO
MONTAÑO GERMAN
PERALTA WALTER
RODRIGUEZ RICARDO
VALEGA ALFONSO

MANUAL DE SISTEMA

CORPORACIÓN EDUCATIVA MAYOR DEL DESARROLLO
SIMÓN BOLÍVAR
BARRANQUILLA
2005

INDICE

1. INTRODUCCIÓN	1
2. OBJETIVOS	1
2.1 OBJETIVO GENERAL	1
2.2 OBJETIVOS ESPECIFICOS	1
3. REQUISITOS MINIMOS DE INSTALACIÓN	2
3.1 DirectX 8.1	2
3.2 Génesis 3D	2
3.3 Milkshape 3D	2
3.4 Terragen	2
4. CONCEPTOS BÁSICOS PARA LA INSTALACIÓN DE SOFTWARE'S	3
4.1 INSTALACIÓN DEL MOTOR GRAFICO GENESIS 3D	3
4.2 INSTALACIÓN DEL MILKSHAPE 3D	13
4.3 INSTALACIÓN DEL TERRAGEN	20
5. MODELOS DEL SISTEMA	25
5.1 DIAGRAMAS DE FLUJO DATOS	25
5.2 DIAGRAMA JERARQUICO FUNCIONAL	28

ISO 9000-3 Normas Para La Gestión Y Aseguramiento De La Calidad.	Implementación De Un Manual De Técnicas Para El Desarrollo De Juegos De Video Mediante La Utilización De Motores Gráficos 3D.
Versión 1.0	Revisado Por El Ing. Alexis Messino

MANUAL DEL SISTEMA

1. INTRODUCCIÓN

Como es bien sabido todos los programas que se van distribuyen hoy en día tienen la particularidad de exigir componentes adicionales para su mejor funcionamiento, los Motores Gráficos 3D no son la excepción pues requieren de una amplia gama de herramientas que ayuden en las distintas labores ya sea de diseño o de desarrollo; pues en la labor de crear Videojuegos se está diseñar y desarrollar inmediatamente, para no perder el enfoque e inspiración creativo, en consecuencia tener todos los componentes necesarios para esta tarea es imprescindible; pero también es importante recordar que no cualquier computador es apto para la labor de crear Videojuegos, por tanto es importante saber con que material se cuenta.

2. OBJETIVOS

2.1 OBJETIVO GENERAL

Comprender el proceso de instalación del Motor Gráfico Genesis 3D y de las diferentes herramientas que complementan su funcionamiento.

2.2 OBJETIVOS ESPECIFICOS

- Entender la interfaz y el funcionamiento de todas las herramientas que complementan el diseño desarrollo de videojuegos con el Motor Gráfico Genesis 3D.
- Aprender el proceso de instalación de todas las herramientas usadas para la creación de Videojuegos.
- Conocer el funcionamiento del computador a nivel de procesamiento gráfico.
- Dar soporte a los manuales si en un futuro las herramientas tienen actualizaciones.

ISO 9000-3 Normas Para La Gestión Y Aseguramiento De La Calidad.	Implementación De Un Manual De Técnicas Para El Desarrollo De Juegos De Video Mediante La Utilización De Motores Gráficos 3D.
Versión 1.0	Revisado Por El Ing. Alexis Messino



ISO 9000-3 Normas Para La Gestión Y Aseguramiento De La Calidad.	Implementación De Un Manual De Técnicas Para El Desarrollo De Juegos De Video Mediante La Utilización De Motores Gráficos 3D.
Versión 1.0	Revisado Por El Ing. Alexis Messino

3. REQUISITOS MINIMOS DE INSTALACIÓN

Todo programa que se instala requiere de unos requerimientos mínimos de hardware y de software para su instalación y ejecución los cuales deben ser cumplidos si se desea poder usar el recurso de una manera eficiente. La mayoría de aplicaciones gráficas exigen recursos de hardware muy altos los cuales la mayoría de veces no están al alcance del bolsillo de personas que puedan interesarse por este tipo de herramientas. A continuación se detallan los requerimientos de hardware y software para cada una de las herramientas que se referencian en el manual.

3.1 DirectX 8.1

- ✓ Procesador de 166 MHz o superior.
- ✓ Memoria de RAM de 128 MB.
- ✓ Chip Gráfico o Tarjeta de Video de 2 MB.
- ✓ Windows 98 SE o versiones superiores.

Nota: Si se dispone Windows 2000 o XP y de equipos de capacidad superior es preferible instalar DirectX 9.0.

3.2 Génesis 3D

- ✓ Procesador de 500 MHz o superior.
- ✓ Memoria de 64 MB.
- ✓ Tarjeta de Video de 4 MB.
- ✓ Windows 98 SE o versiones superiores.

Nota: Para obtener mejor desempeño en el desarrollo se recomienda utilizar hardware con más prestaciones.

3.3 Milkshape 3D

- ✓ Procesador de 550 MHz o superior.
- ✓ Memoria de 96 MB.
- ✓ Tarjeta de Video de 6 MB.
- ✓ Windows 98 SE o versiones superiores.

3.4 Terragen

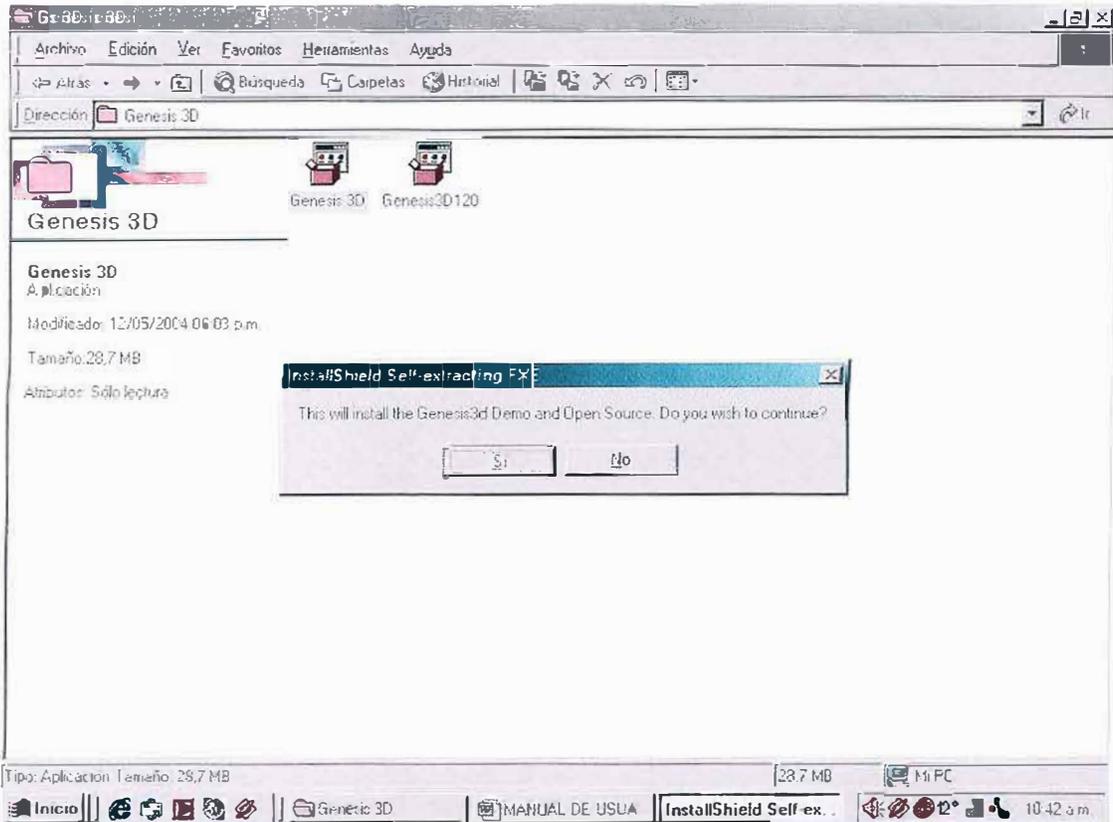
- ✓ Procesador de 450 MHz o superior.
- ✓ Memoria de 64 MB.
- ✓ Tarjeta de Video de 4 MB.
- ✓ Windows 98 SE o versiones superiores.

ISO 9000-3 Normas Para La Gestión Y Aseguramiento De La Calidad.	Implementación De Un Manual De Técnicas Para El Desarrollo De Juegos De Video Mediante La Utilización De Motores Gráficos 3D.	2
Versión 1.0	Revisado Por El Ing. Alexis Messino	

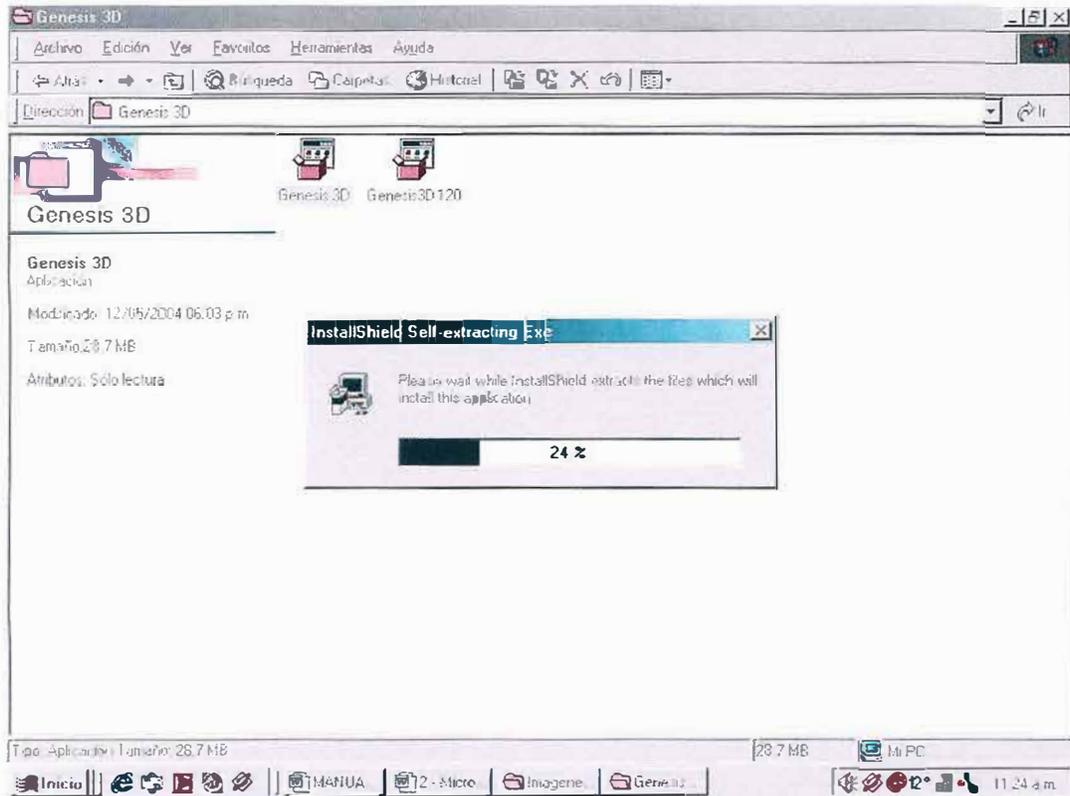
4. CONCEPTOS BÁSICOS PARA LA INSTALACIÓN DE SOFTWARE'S

4.1 INSTALACIÓN DEL MOTOR GRAFICO GENESIS 3D

- ✓ Ejecutar el archivo de instalación. Como este es un archivo comprimido, va a preguntarle que si (yes) desea extraer el archivo o no. Vamos a escoger la opción yes.

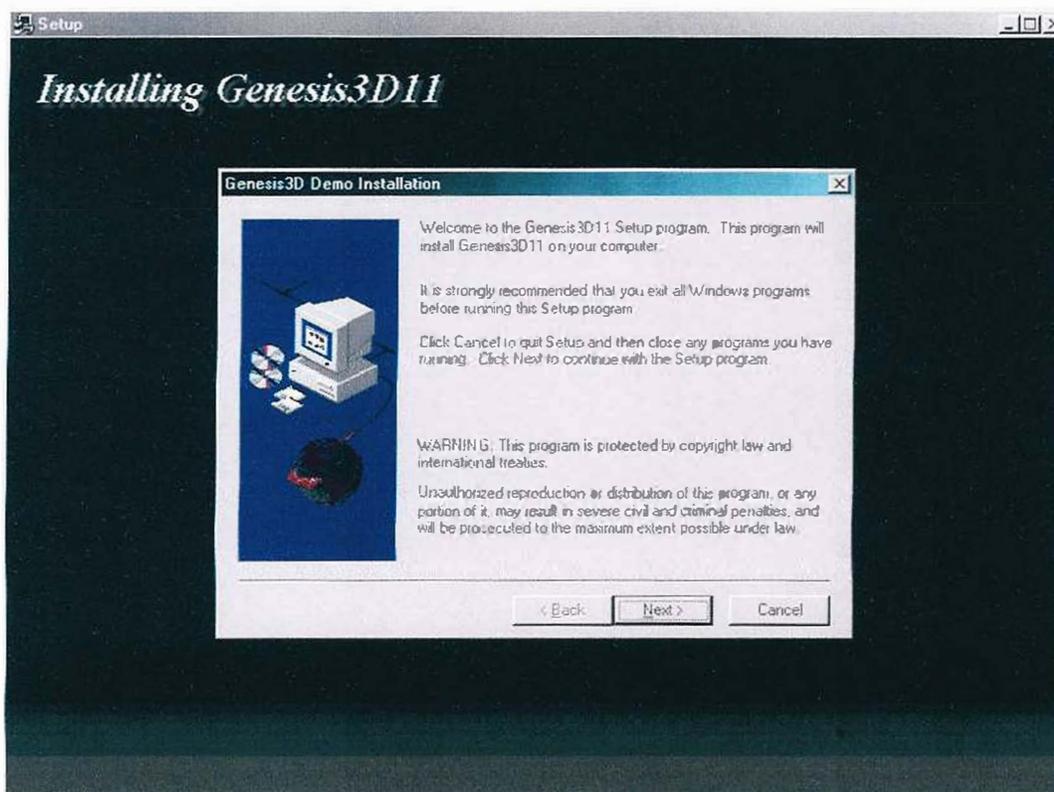


- ✓ Inicia el proceso de extracción de los distintos archivos de instalación del Motor Gráfico Génesis 3D.



ISO 9000-3 Normas Para La Gestión Y Aseguramiento De La Calidad.	Implementación De Un Manual De Técnicas Para El Desarrollo De Juegos De Video Mediante La Utilización De Motores Gráficos 3D.
Versión 1.0	Revisado Por El Ing. Alexis Messino

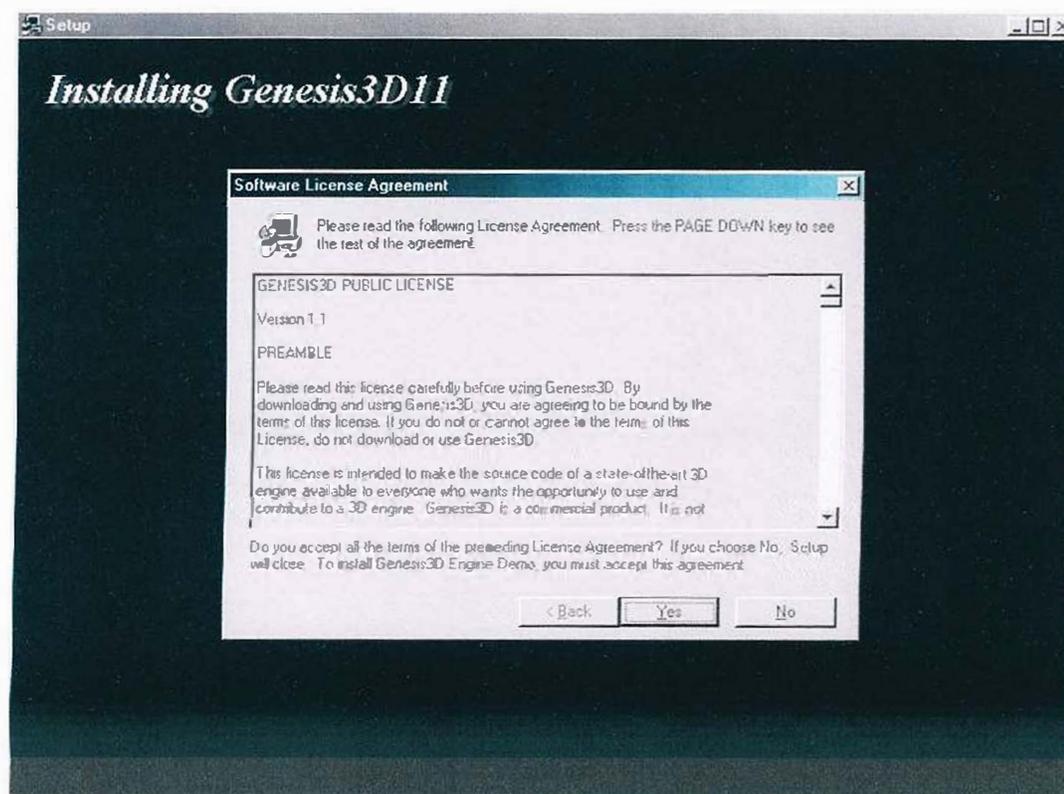
- ✓ Después de haber extraído el archivo de instalación, te muestra y te dice que si realmente deseas instalarlo en tu Pc, si deseas instalarlo vas a dar clic en la opción next (Siguiente), si no deseas instalarlo y deseas salir realmente de la instalación damos clic en la opción cancelar.



ISO 9000-3 Normas Para La Gestión Y Aseguramiento De La Calidad.	Implementación De Un Manual De Técnicas Para El Desarrollo De Juegos De Video Mediante La Utilización De Motores Gráficos 3D.
Versión 1.0	Revisado Por El Ing. Alexis Messino

ISO 9000-3 Normas Para La Gestión Y Aseguramiento De La Calidad.	Implementación De Un Manual De Técnicas Para El Desarrollo De Juegos De Video Mediante La Utilización De Motores Gráficos 3D.
Versión 1.0	Revisado Por El Ing. Alexis Messino

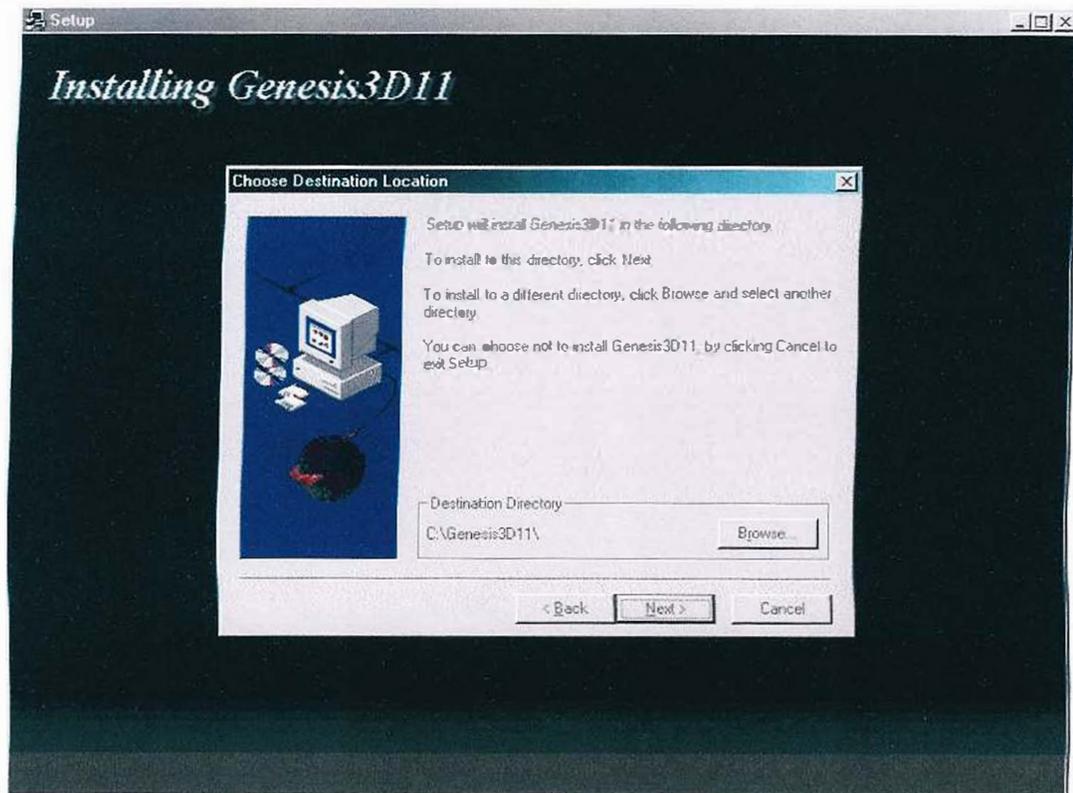
- ✓ En esta ventana nos muestra que tipo de licencia y que versión tiene el software mismo que estamos deseando instalar, escogemos si, si estamos seguros de desear instalarlo.



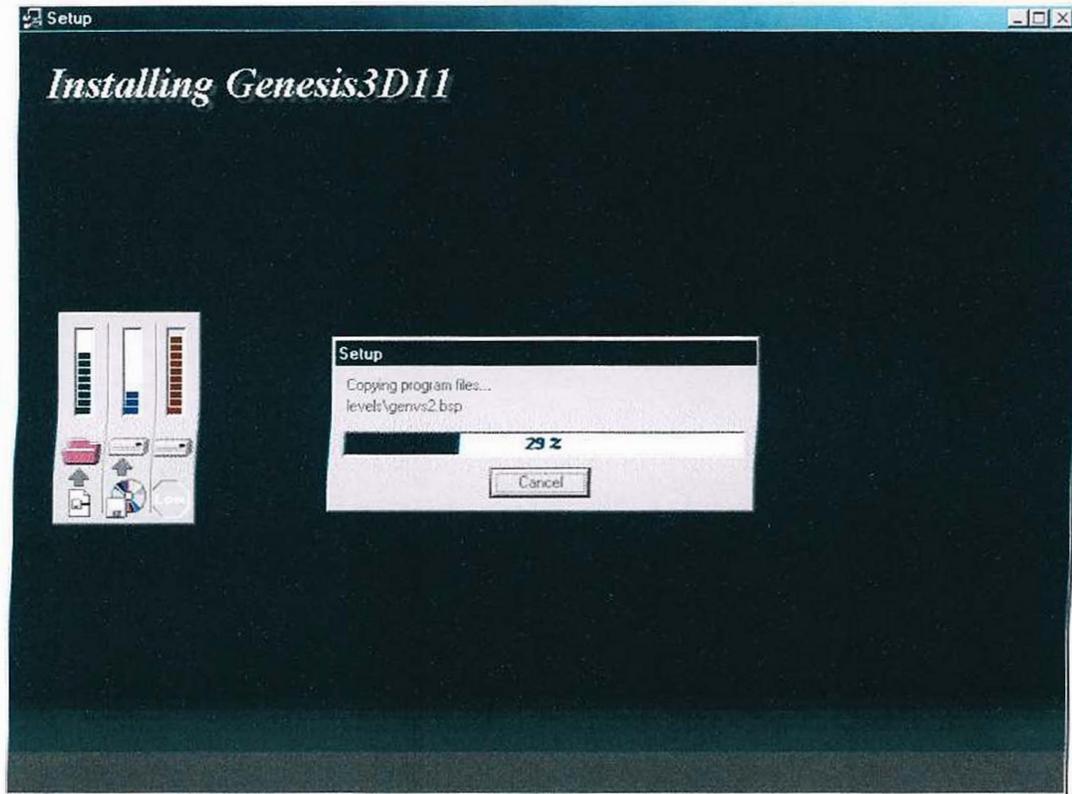
ISO 9000-3 Normas Para La Gestión Y Aseguramiento De La Calidad.	Implementación De Un Manual De Técnicas Para El Desarrollo De Juegos De Video Mediante La Utilización De Motores Gráficos 3D.
Versión 1.0	Revisado Por El Ing. Alexis Messino



- ✓ En esta ventana se muestra o nos dice la localización (o el directorio) donde se va instalar. si existe lo sobrescribe y si no existe este directorio el mismo programa de instalación del Genesis lo crea.

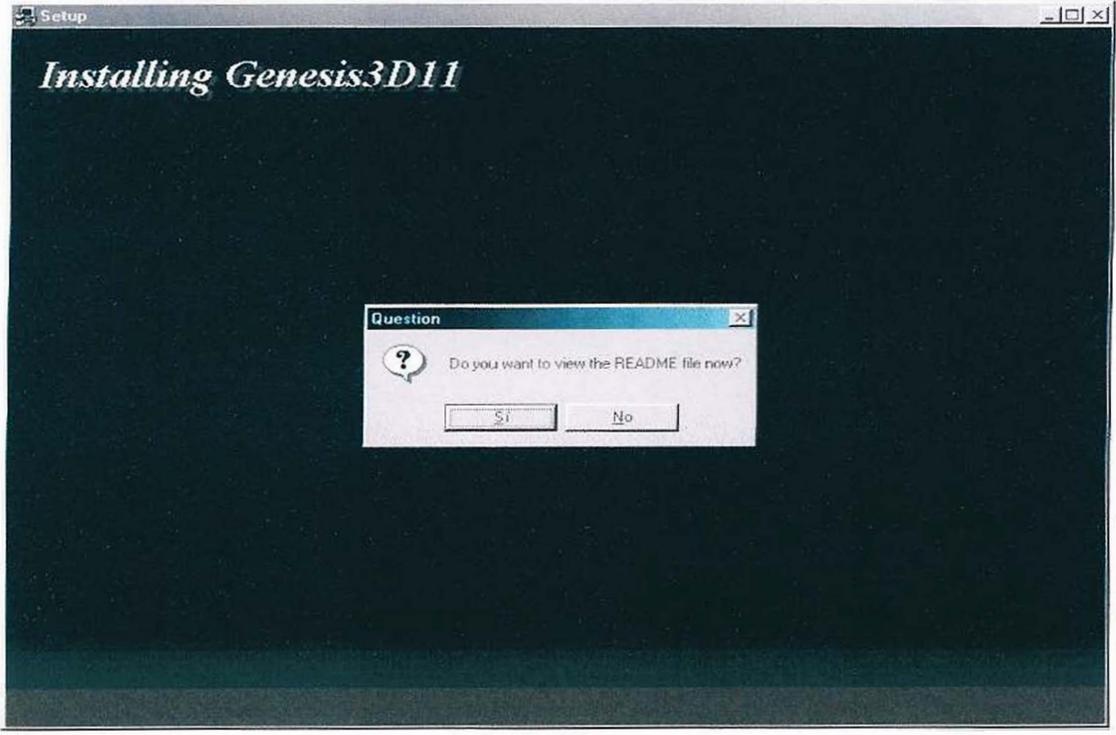


- ✓ Aquí se inicia realmente el proceso de instalación del Génesis 3D.



ISO 9000-3 Normas Para La Gestión Y Aseguramiento De La Calidad.	Implementación De Un Manual De Técnicas Para El Desarrollo De Juegos De Video Mediante La Utilización De Motores Gráficos 3D.
Versión 1.0	Revisado Por El Ing. Alexis Messino

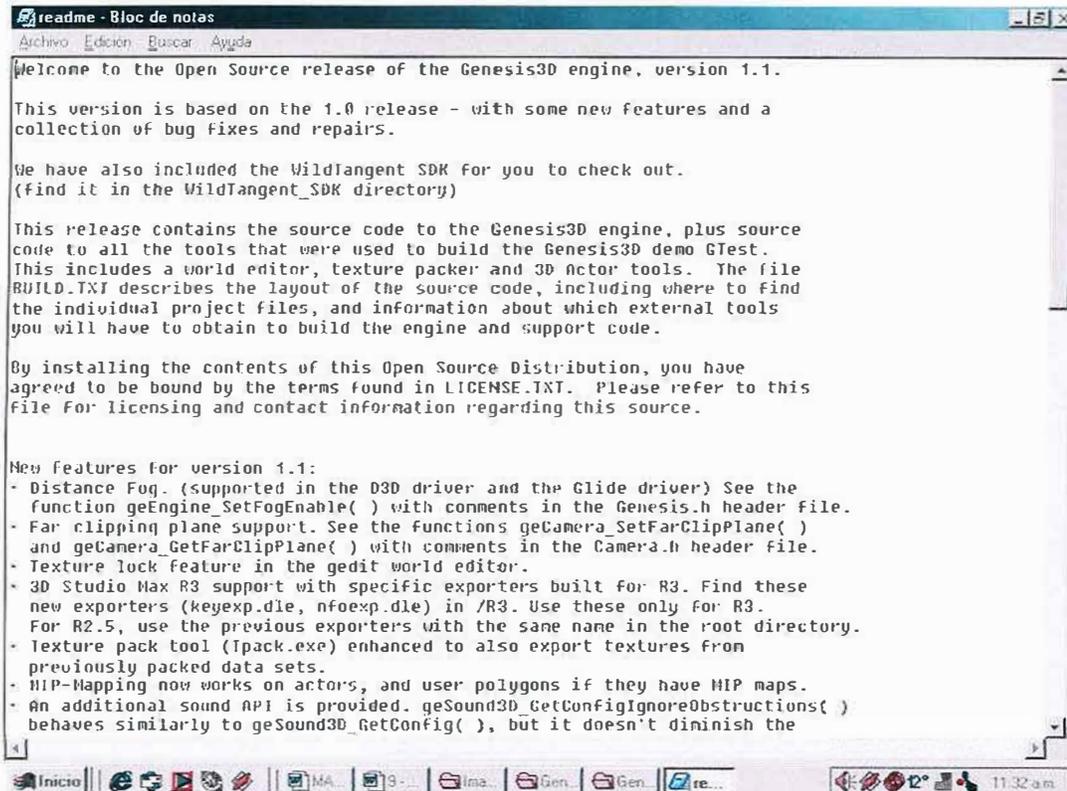
- ✓ Después de haber llegado al 100 % de la instalación, este le va a preguntar que si desea leer el archivo de instalación del software en cuestión en este caso el Génesis 3D.



ISO 9000-3 Normas Para La Gestión Y Aseguramiento De La Calidad.	Implementación De Un Manual De Técnicas Para El Desarrollo De Juegos De Video Mediante La Utilización De Motores Gráficos 3D.
Versión 1.0	Revisado Por El Ing. Alexis Messino

ISO 9000-3 Normas Para La Gestión Y Aseguramiento De La Calidad.	Implementación De Un Manual De Técnicas Para El Desarrollo De Juegos De Video Mediante La Utilización De Motores Gráficos 3D.
Versión 1.0	Revisado Por El Ing. Alexis Messino

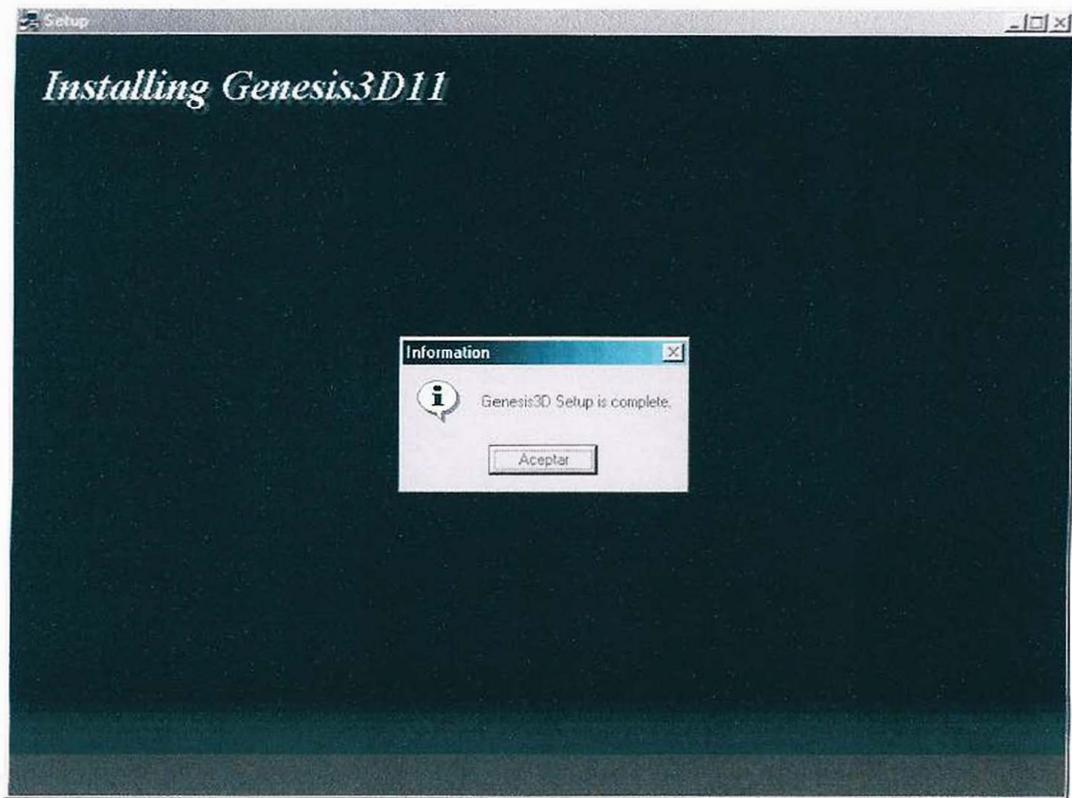
- ✓ Si se escoge la opción si, se le va a desplegar otra ventana con un bloc de notas donde decir la versión, por quien fue desarrollado, y las distintas herramientas que el Génesis utiliza, en el caso que escojas la opción no, lo que va a suceder es que automáticamente vas a pasar al siguiente ítem.



ISO 9000-3 Normas Para La Gestión Y Aseguramiento De La Calidad.	Implementación De Un Manual De Técnicas Para El Desarrollo De Juegos De Video Mediante La Utilización De Motores Gráficos 3D.
Versión 1.0	Revisado Por El Ing. Alexis Messino

ISO 9000-3 Normas Para La Gestión Y Aseguramiento De La Calidad.	Implementación De Un Manual De Técnicas Para El Desarrollo De Juegos De Video Mediante La Utilización De Motores Gráficos 3D.
Versión 1.0	Revisado Por El Ing. Alexis Messino

- ✓ Después de haber leído o cerrado el archivo de instalación si escogió la opción si y igualmente si escogió la opción no, se le va a desplegar otra ventana mostrándole que el proceso de instalación del Genesis 3D ha terminado.

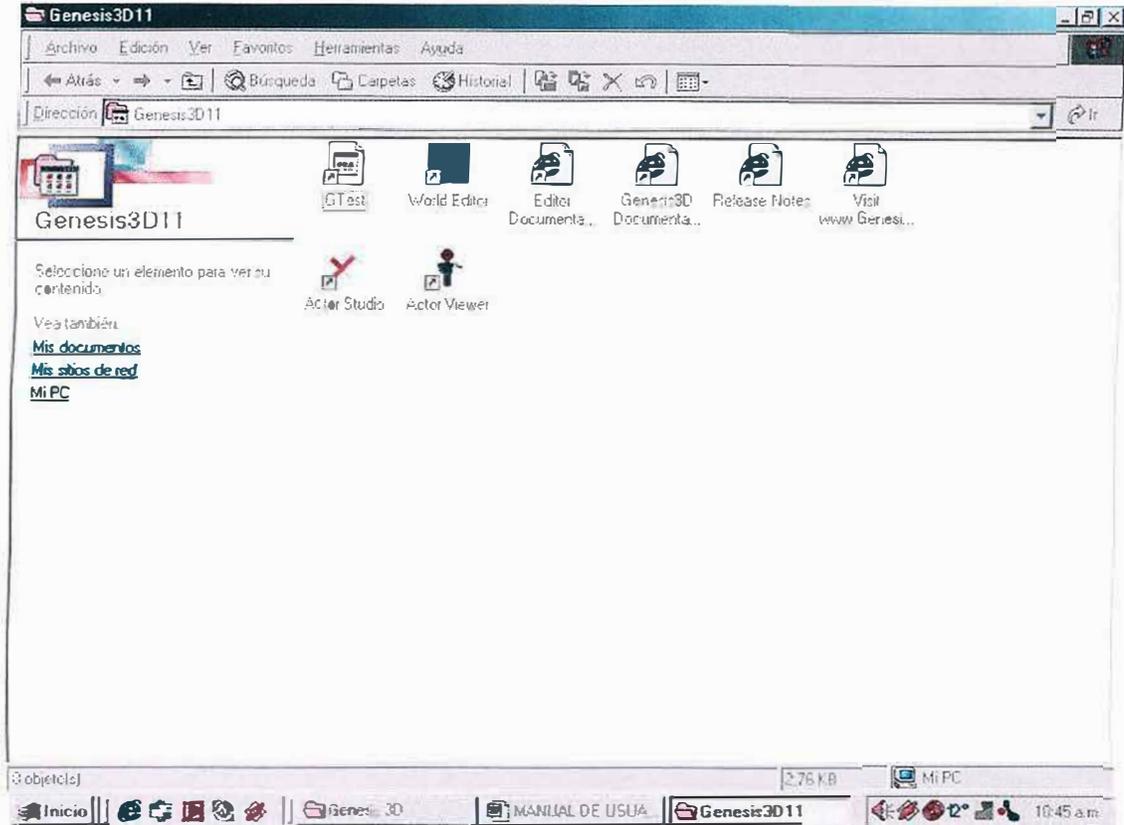


ISO 9000-3 Normas Para La Gestión Y Aseguramiento De La Calidad.	Implementación De Un Manual De Técnicas Para El Desarrollo De Juegos De Video Mediante La Utilización De Motores Gráficos 3D.
Versión 1.0	Revisado Por El Ing. Alexis Messino



ISO 9000-3 Normas Para La Gestión Y Aseguramiento De La Calidad.	Implementación De Un Manual De Técnicas Para El Desarrollo De Juegos De Video Mediante La Utilización De Motores Gráficos 3D.
Versión 1.0	Revisado Por El Ing. Alexis Messino

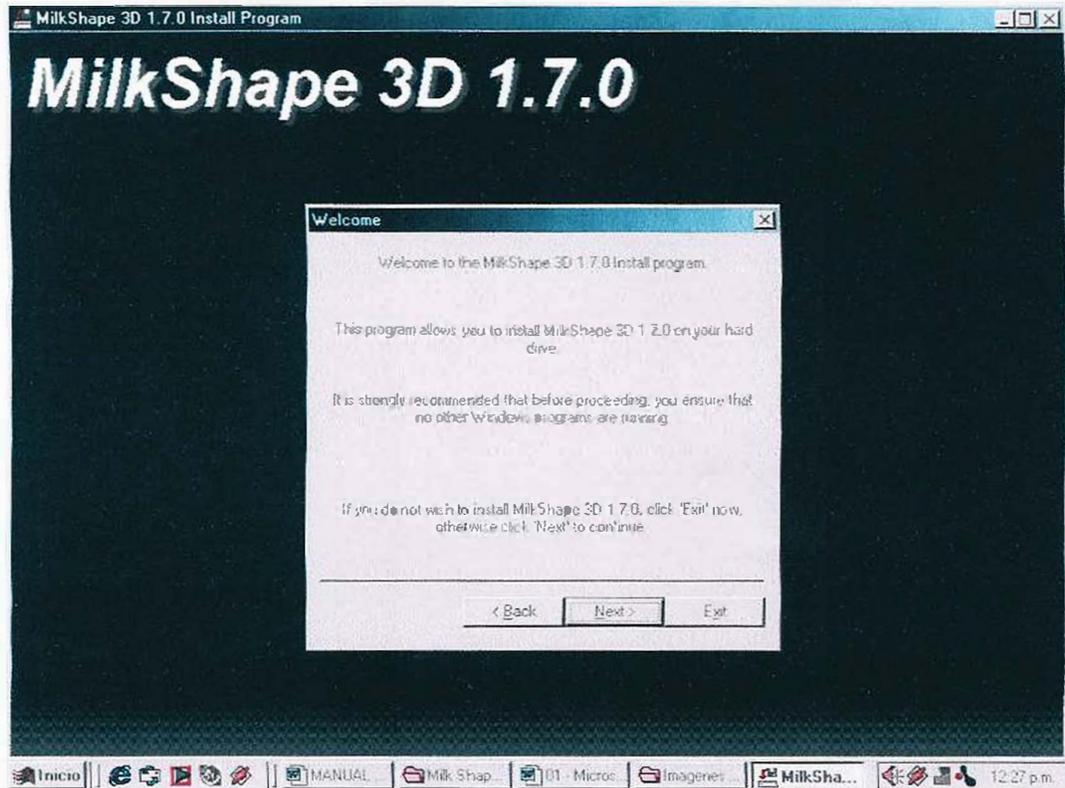
- ✓ Estos son las distintas herramientas se instalan con el Motor Gráfico Génesis 3D, como son el World Editor, Actor Studio y Actor Viewer, entre otros.



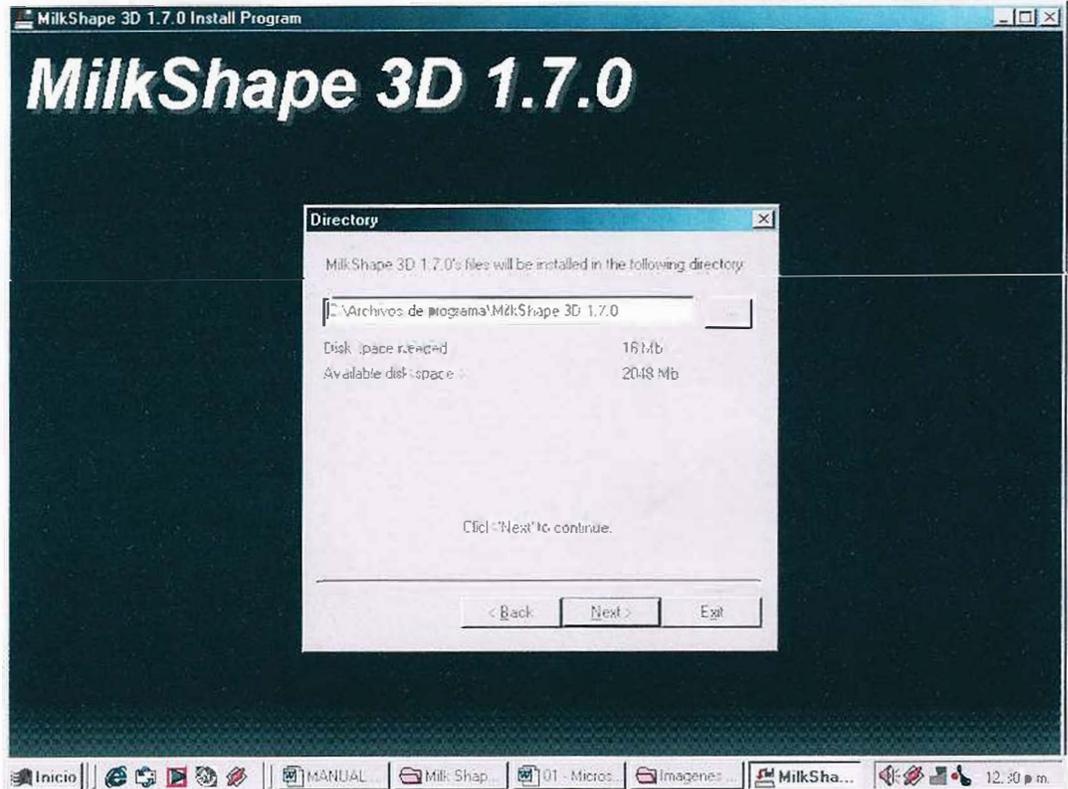
ISO 9000-3 Normas Para La Gestión Y Aseguramiento De La Calidad.	Implementación De Un Manual De Técnicas Para El Desarrollo De Juegos De Video Mediante La Utilización De Motores Gráficos 3D.	12
Versión 1.0	Revisado Por El Ing. Alexis Messino	

4.2 INSTALACIÓN DEL MILKSHAPE 3D (EDITOR DE PERSONAJES)

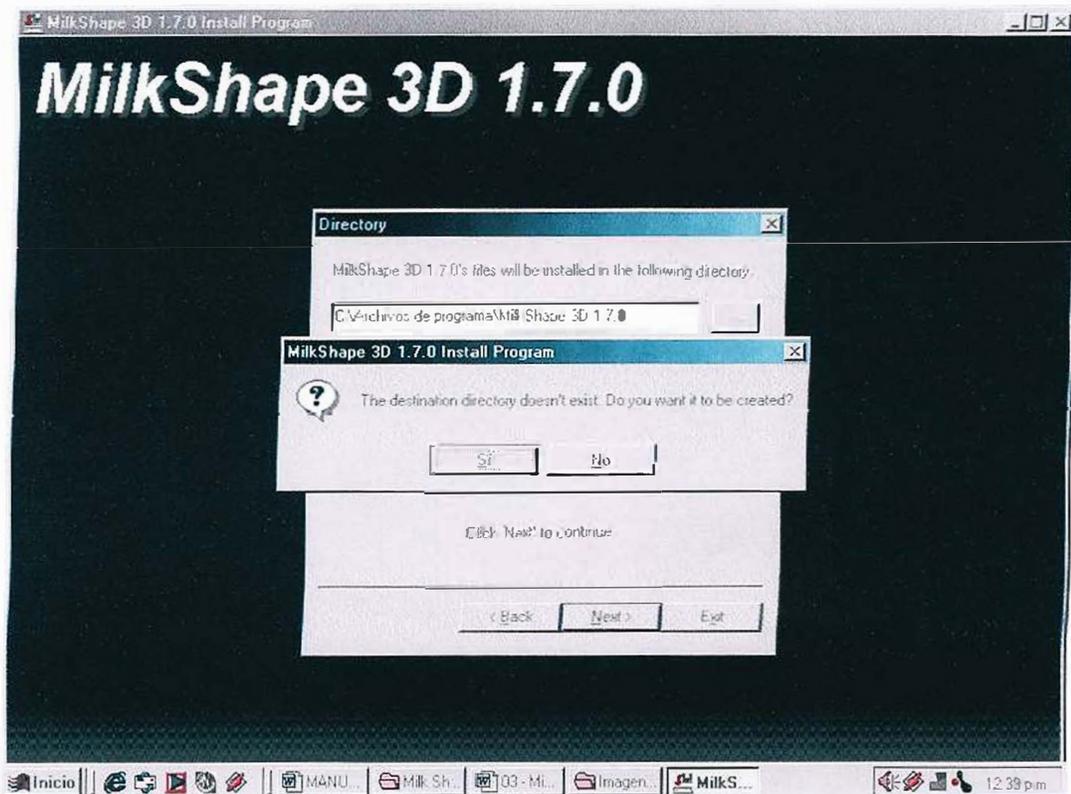
- ✓ Como primera medida le da la bienvenida al proceso de instalación del MilkShape 3D. le dice que este mismo será instalado en su disco duro, ahora si desea continuar, debe clic sobre el botón next (siguiente), si no desea continuar y salir del proceso de instalación debe dar clic en el botón exit.



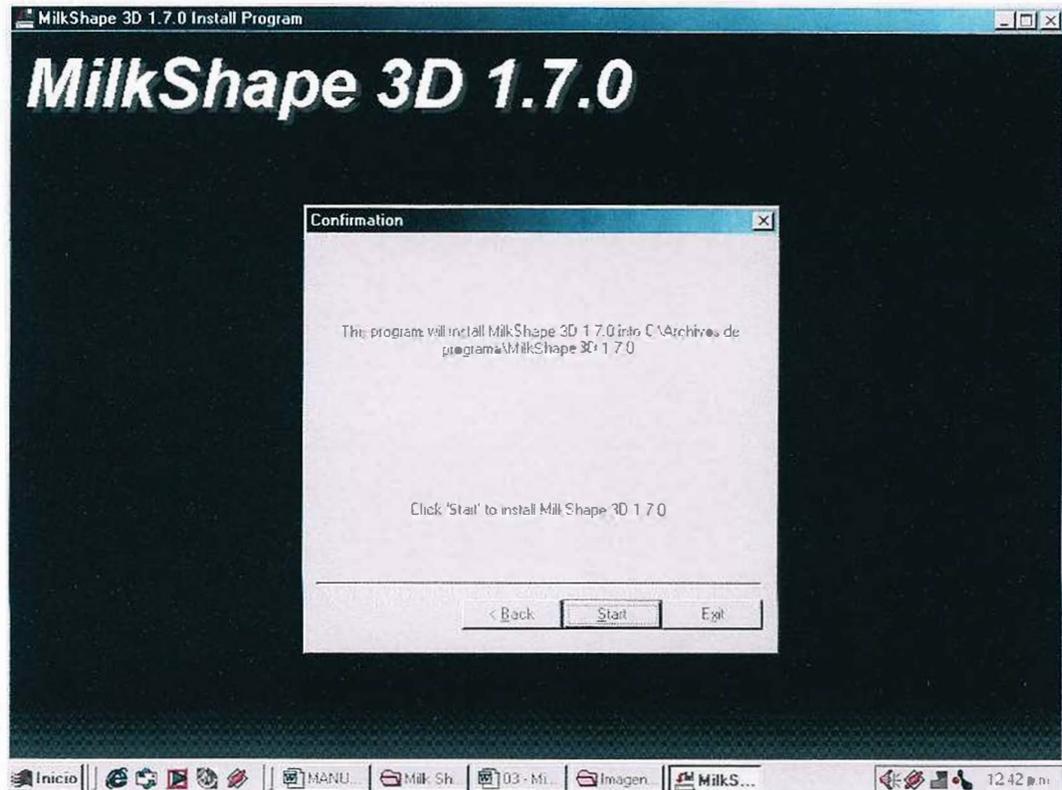
- ✓ Le muestra la localización (o directorio), donde se va instalar, además nos muestra la cantidad de espacio que necesita para ser instalado en su Pc, en este caso 16Mb, este cantidad de espacio que necesita para su instalación va a depender de la versión misma de este software que esta siendo instalado.
Si deseamos continuar con la instalación damos clic en next (siguiente) y deseamos abandonar esta damos clic en exit.



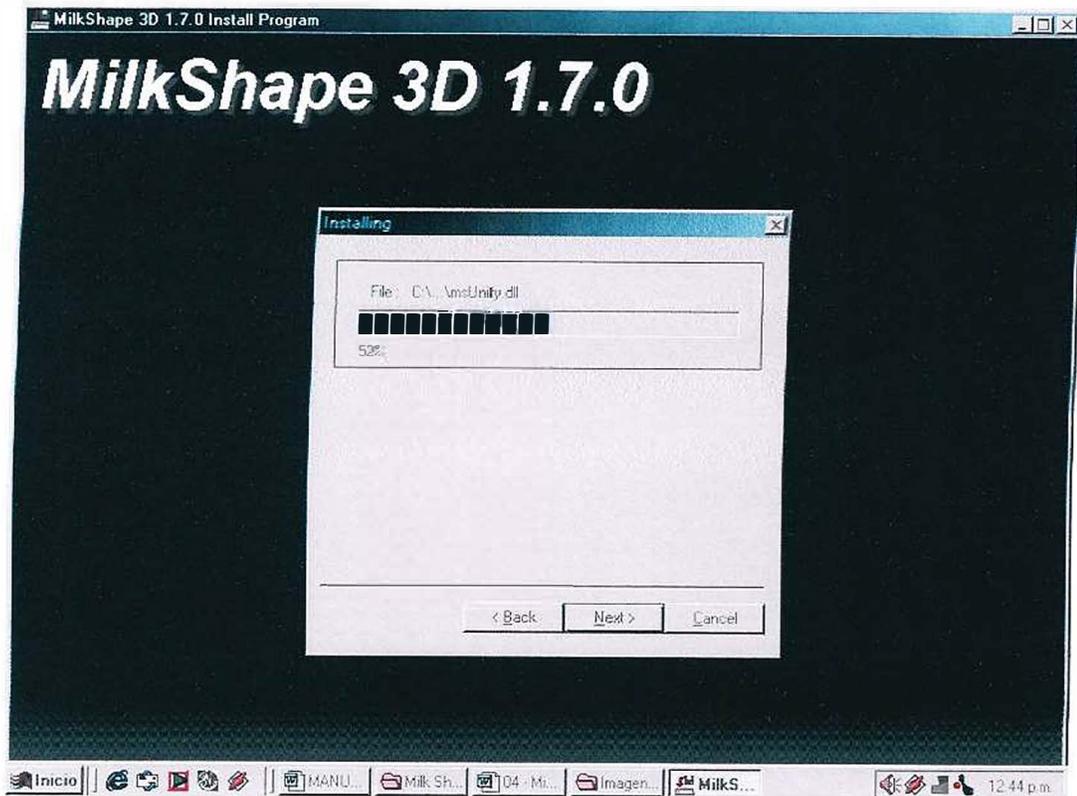
- ✓ En esta ventana el programa de instalación nos muestra, que el directorio donde va a instalar sus componentes no existe, de igual forma nos pregunta que si deseamos crearlo, en el caso que deseamos crearlo que es lo mas lógico debemos dar clic en la opción si, ahora si no deseamos crearlo damos clic en la opción no, pero al escoger esta opción el programa de instalación nos va a llevar al ítem anterior.



- ✓ Nos muestra que el programa se va instalar en el directorio que el creo y además nos dice que si deseamos continuar con la instalación del software debemos dar clic en start para iniciar la instalación.

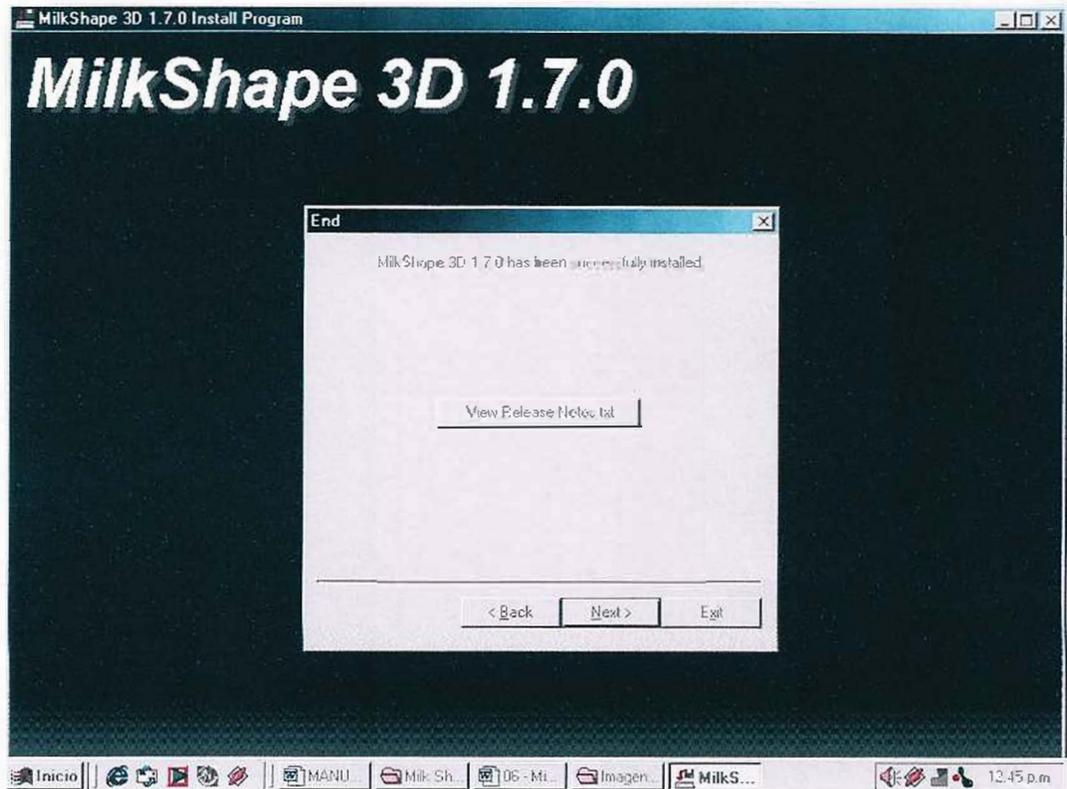


- ✓ Se inicia realmente el proceso de instalación.



ISO 9000-3 Normas Para La Gestión Y Aseguramiento De La Calidad.	Implementación De Un Manual De Técnicas Para El Desarrollo De Juegos De Video Mediante La Utilización De Motores Gráficos 3D.
Versión 1.0	Revisado Por El Ing. Alexis Messino

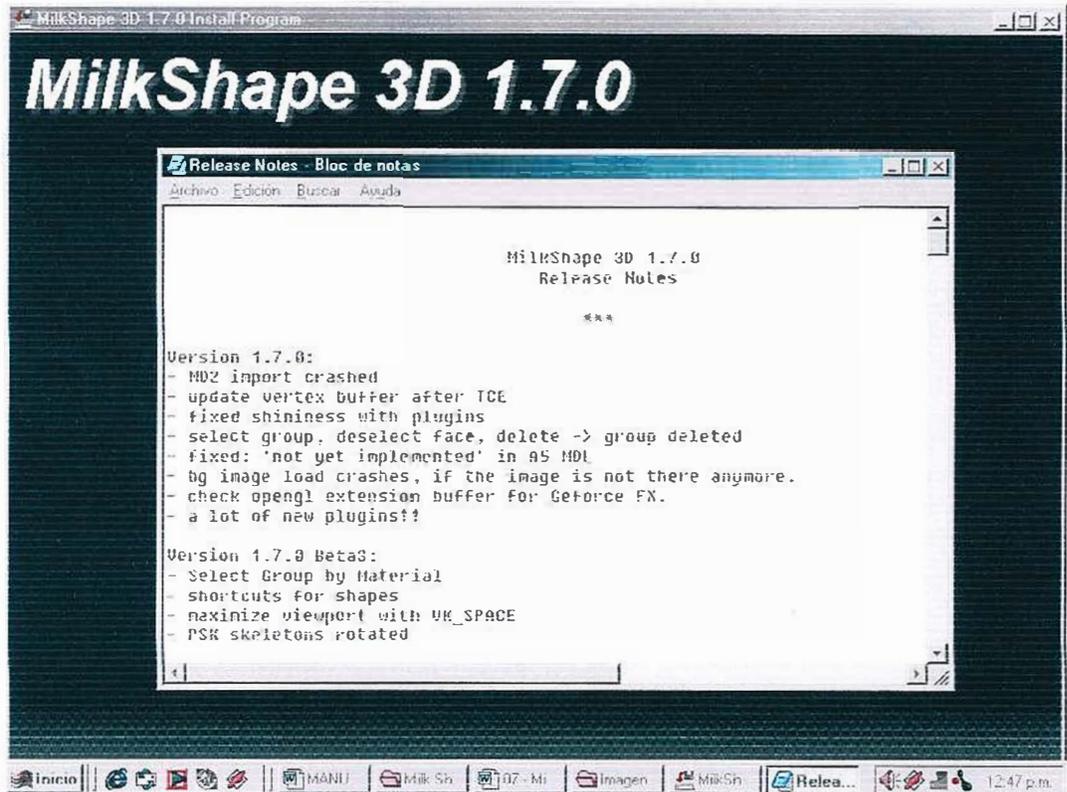
- ✓ Esta es la ventana de finalización del programa de instalación, nos muestra o nos permite visualizar si nosotros queremos el bloc de notas del programa instalado.
Si escogemos esta opción se nos va desplegar un bloc de notas y si escogemos exit, habrá terminado el proceso de instalación de MilkShape 3D.



ISO 9000-3 Normas Para La Gestión Y Aseguramiento De La Calidad.	Implementación De Un Manual De Técnicas Para El Desarrollo De Juegos De Video Mediante La Utilización De Motores Gráficos 3D.	18
Versión 1.0	Revisado Por El Ing. Alexis Messino	

ISO 9000-3 Normas Para La Gestión Y Aseguramiento De La Calidad.	Implementación De Un Manual De Técnicas Para El Desarrollo De Juegos De Video Mediante La Utilización De Motores Gráficos 3D.
Versión 1.0	Revisado Por El Ing. Alexis Messino

- ✓ Esta ventana es en el caso que hallas decidido leer el bloc de notas del programa de instalación. Aquí vas a encontrar los distintos componentes de la versión instalada y las versiones anteriores de este mismo software.

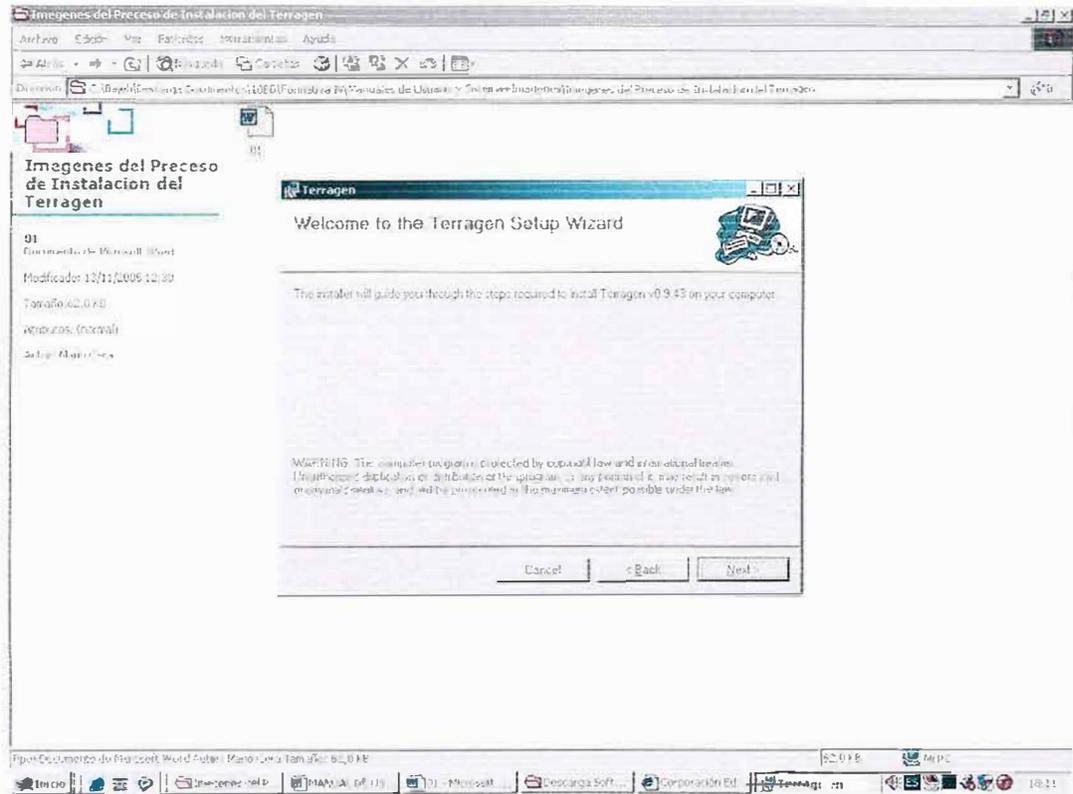


ISO 9000-3 Normas Para La Gestión Y Aseguramiento De La Calidad.	Implementación De Un Manual De Técnicas Para El Desarrollo De Juegos De Video Mediante La Utilización De Motores Gráficos 3D.
Versión 1.0	Revisado Por El Ing. Alexis Messino

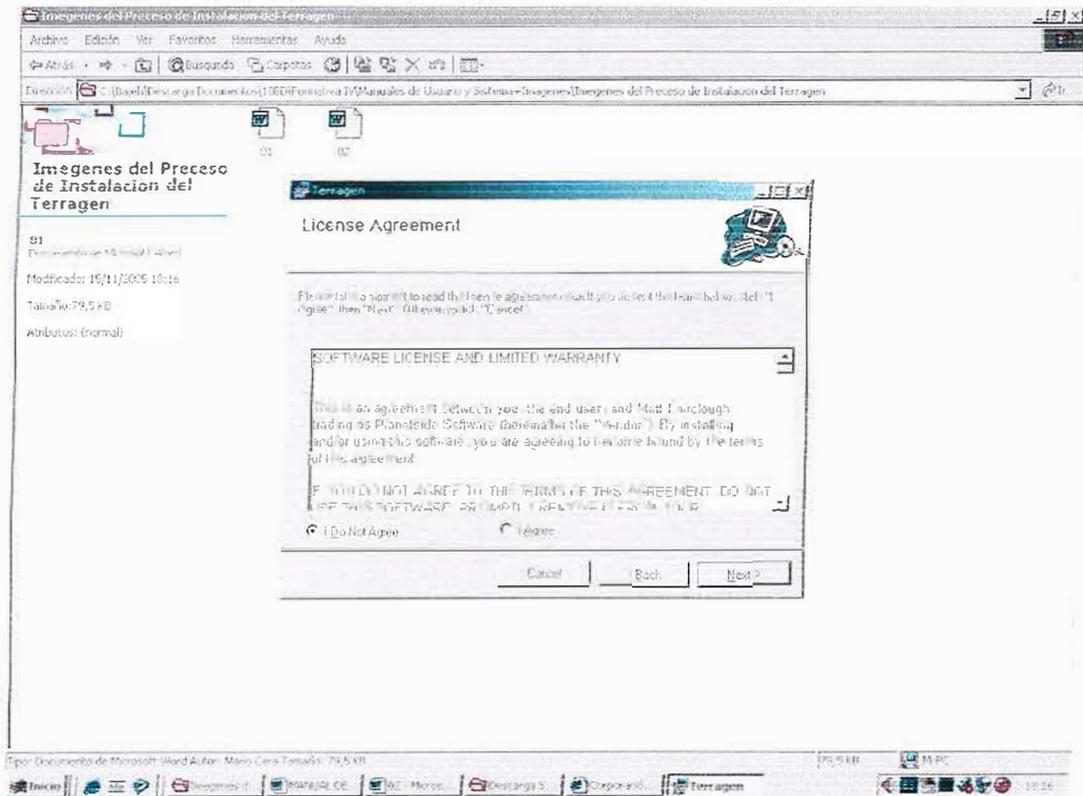


4.3 INSTALACIÓN DEL TERRAGEN (EDITOR DE EFECTOS EXTERIORES)

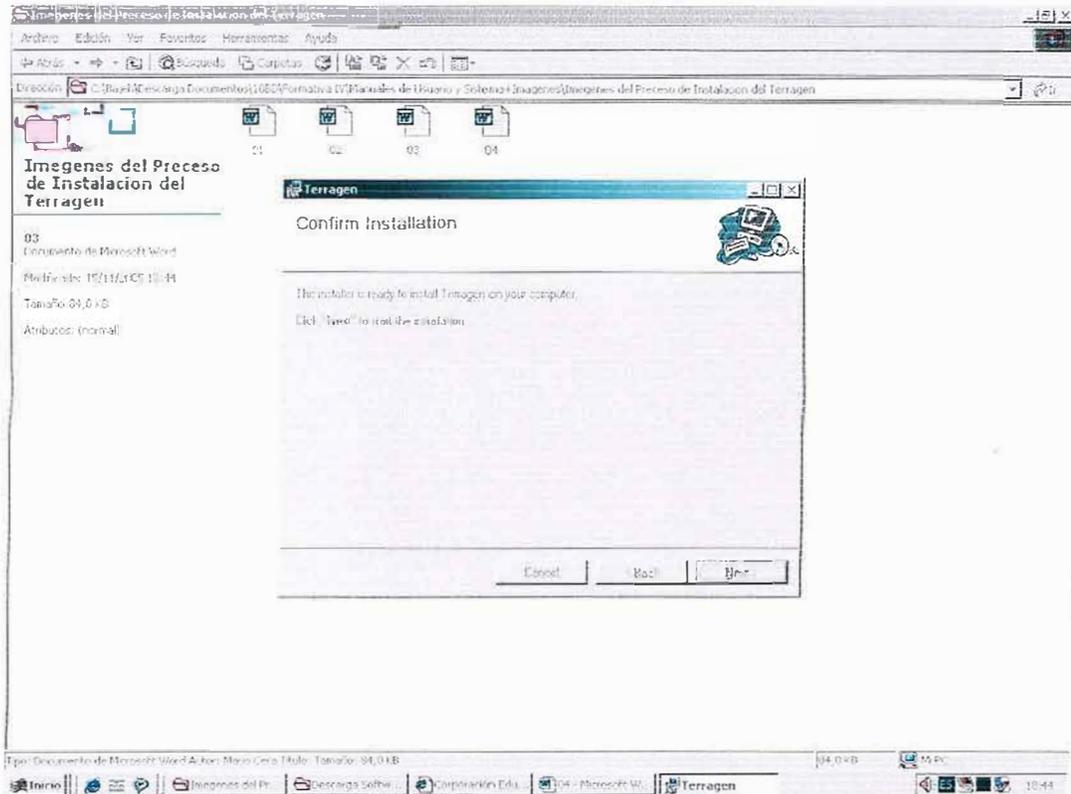
- ✓ Inicialmente le da la bienvenida al proceso de instalación del Terragen, le dice que será instalado en su Pc, si desea continuar con la instalación, debe dar clic sobre el botón next (siguiente), si no desea continuar debe dar clic en el botón Cancel (Cancelar).



- ✓ Esta ventana nos permite visualizar que tipo de licencia tiene este software y su tiempo limite de garantía, en caso que la versión no sea gratuita. I Agree, para agregarlo o instalarlo y damos clic en la opción next (siguiente) para continuar.

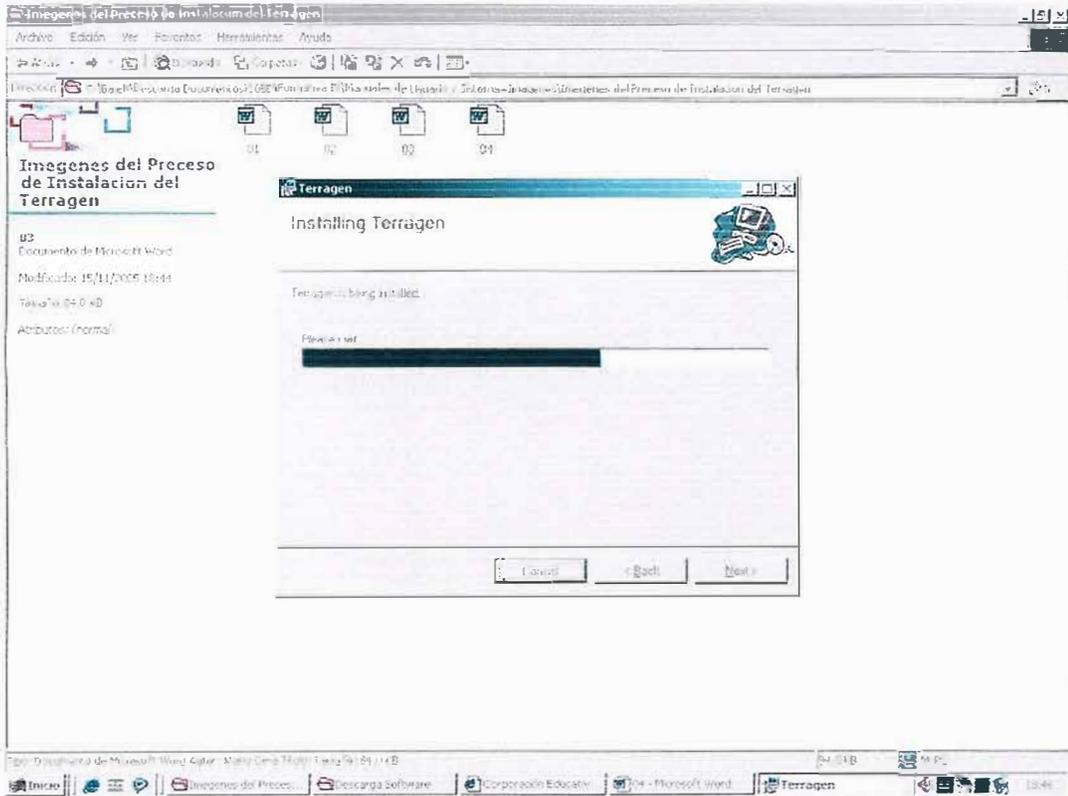


- ✓ Aquí nos pide la confirmación, ósea que si realmente estamos seguros de instalar este software (en este caso Terragen) en nuestro Pc, si deseamos continuar con la instalación damos clic en next (Siguiente).



ISO 9000-3 Normas Para La Gestión Y Aseguramiento De La Calidad.	Implementación De Un Manual De Técnicas Para El Desarrollo De Juegos De Video Mediante La Utilización De Motores Gráficos 3D.
Versión 1.0	Revisado Por El Ing. Alexis Messino

✓ Aquí se inicia realmente el proceso de instalación.



ISO 9000-3 Normas Para La Gestión Y Aseguramiento De La Calidad.	Implementación De Un Manual De Técnicas Para El Desarrollo De Juegos De Video Mediante La Utilización De Motores Gráficos 3D.	23
Versión 1.0	Revisado Por El Ing. Alexis Messino	

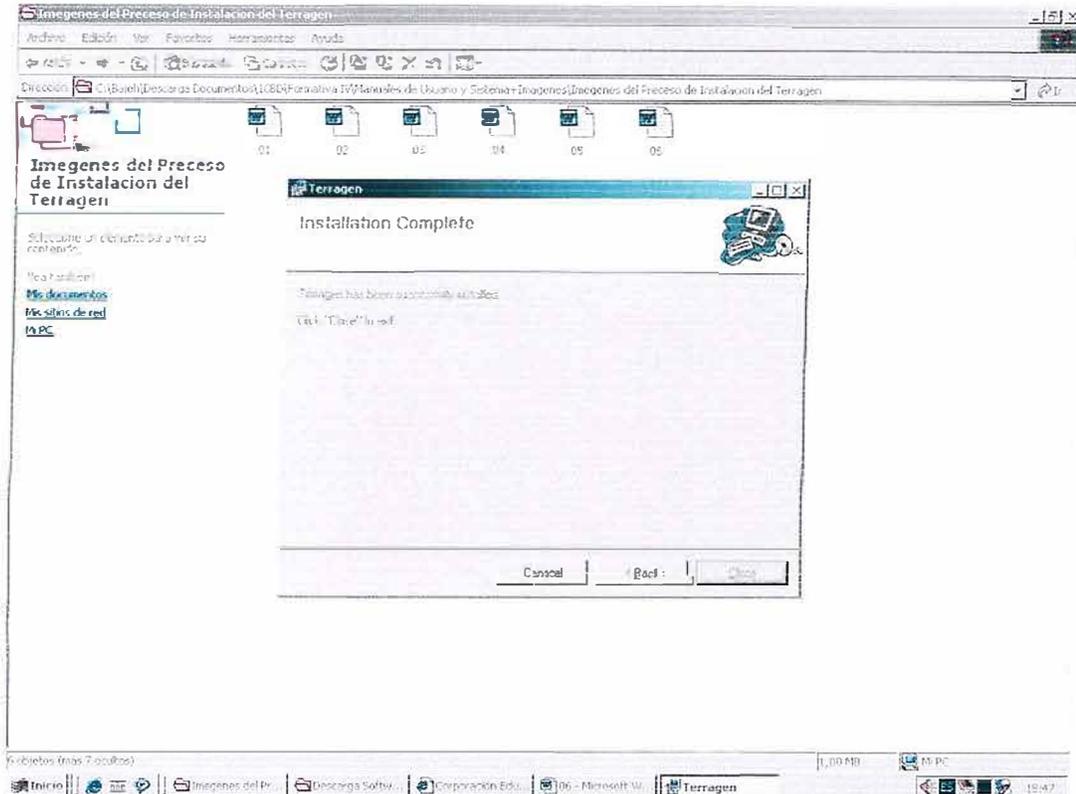
ISO 9000-3 Normas Para La Gestión Y Aseguramiento De La Calidad.

Versión 1.0

Implementación De Un Manual De Técnicas Para El Desarrollo De Juegos De Video Mediante La Utilización De Motores Gráficos 3D.

Revisado Por El Ing. Alexis Messino

- ✓ La instalación ha sido completada satisfactoriamente y además nos muestra la pestaña close (cerrar), que nos indica que se ha terminado la instalación.



ISO 9000-3 Normas Para La Gestión Y Aseguramiento De La Calidad.

Versión 1.0

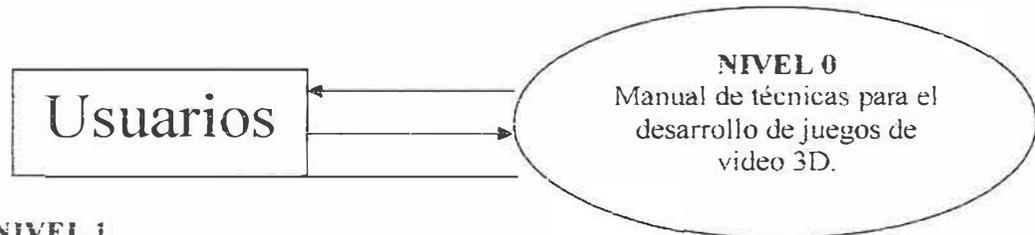
Implementación De Un Manual De Técnicas Para El Desarrollo De Juegos De Video Mediante La Utilización De Motores Gráficos 3D.

Revisado Por El Ing. Alexis Messino

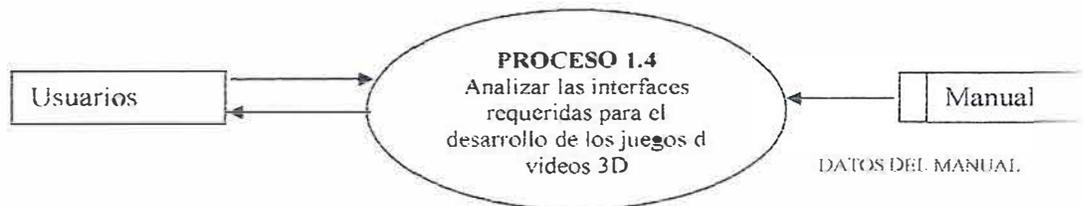
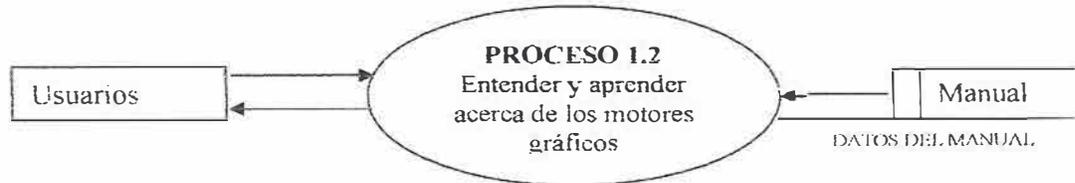
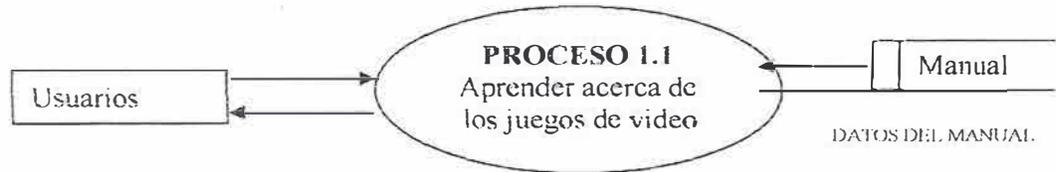
5. MODELOS DEL SISTEMA

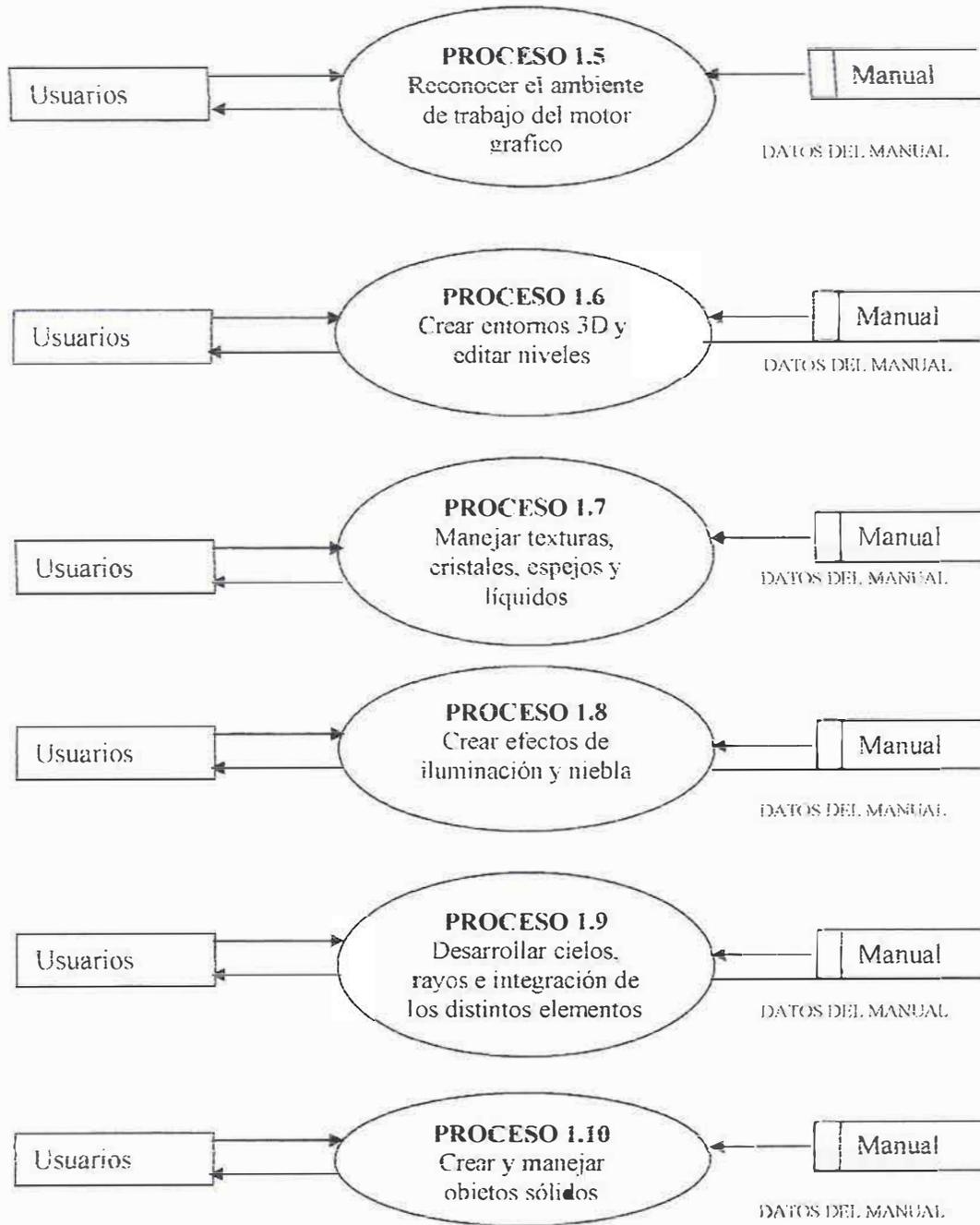
5.1 DIAGRAMAS DE FLUJO DATOS

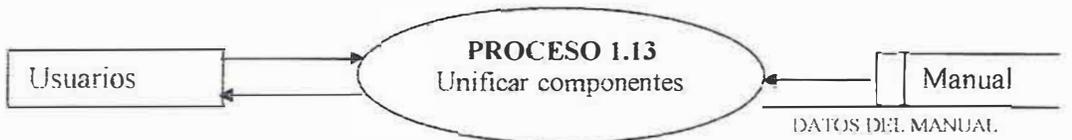
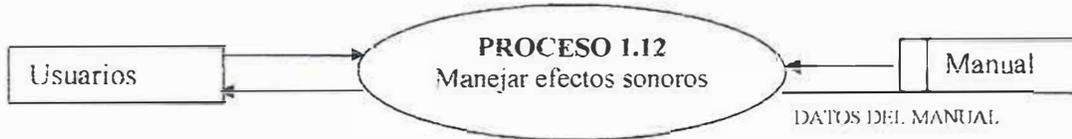
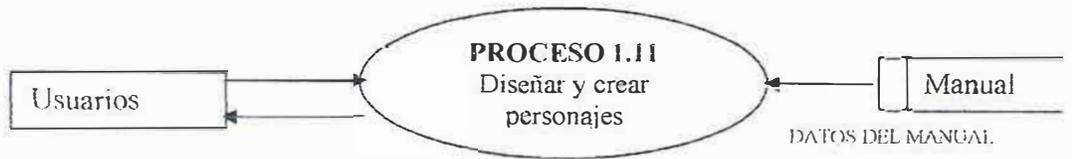
NIVEL 0



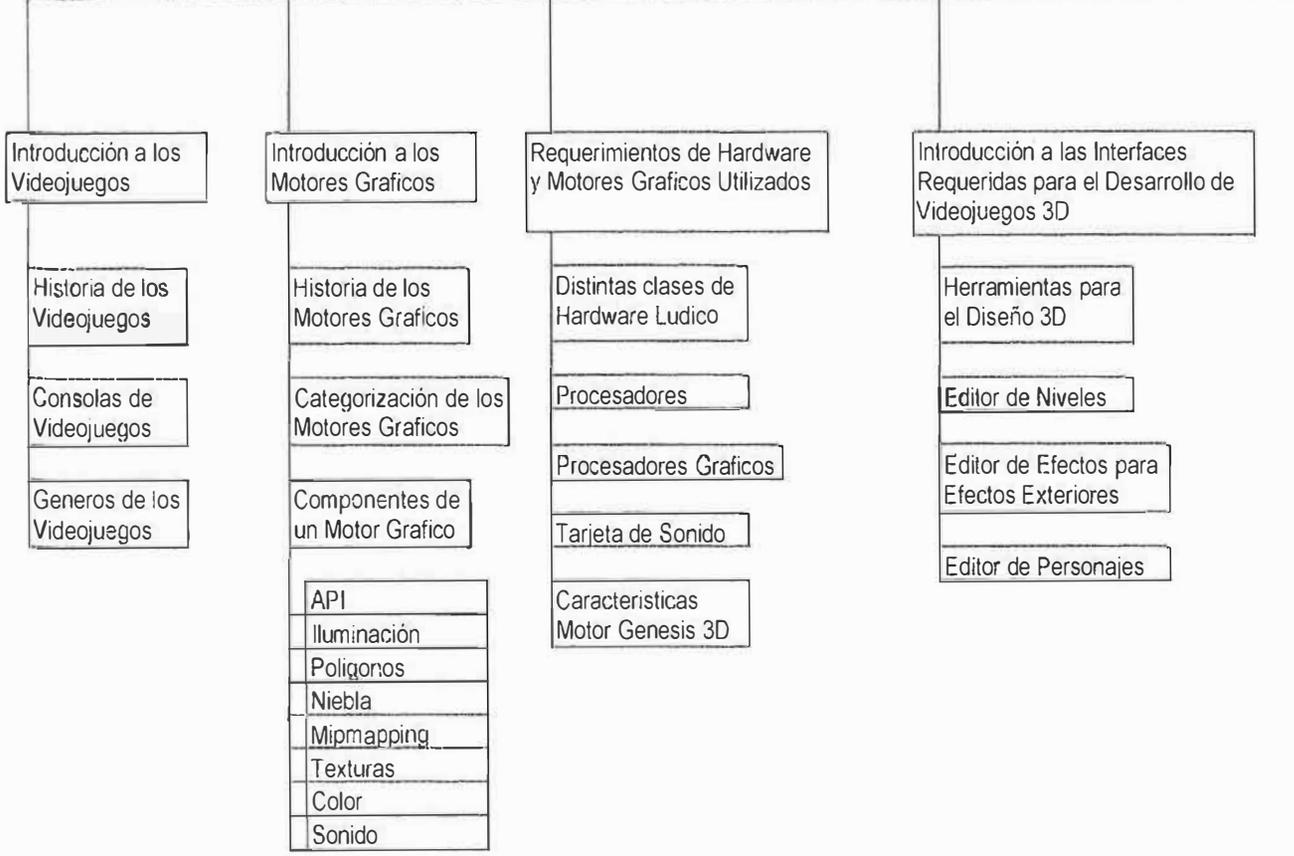
NIVEL 1

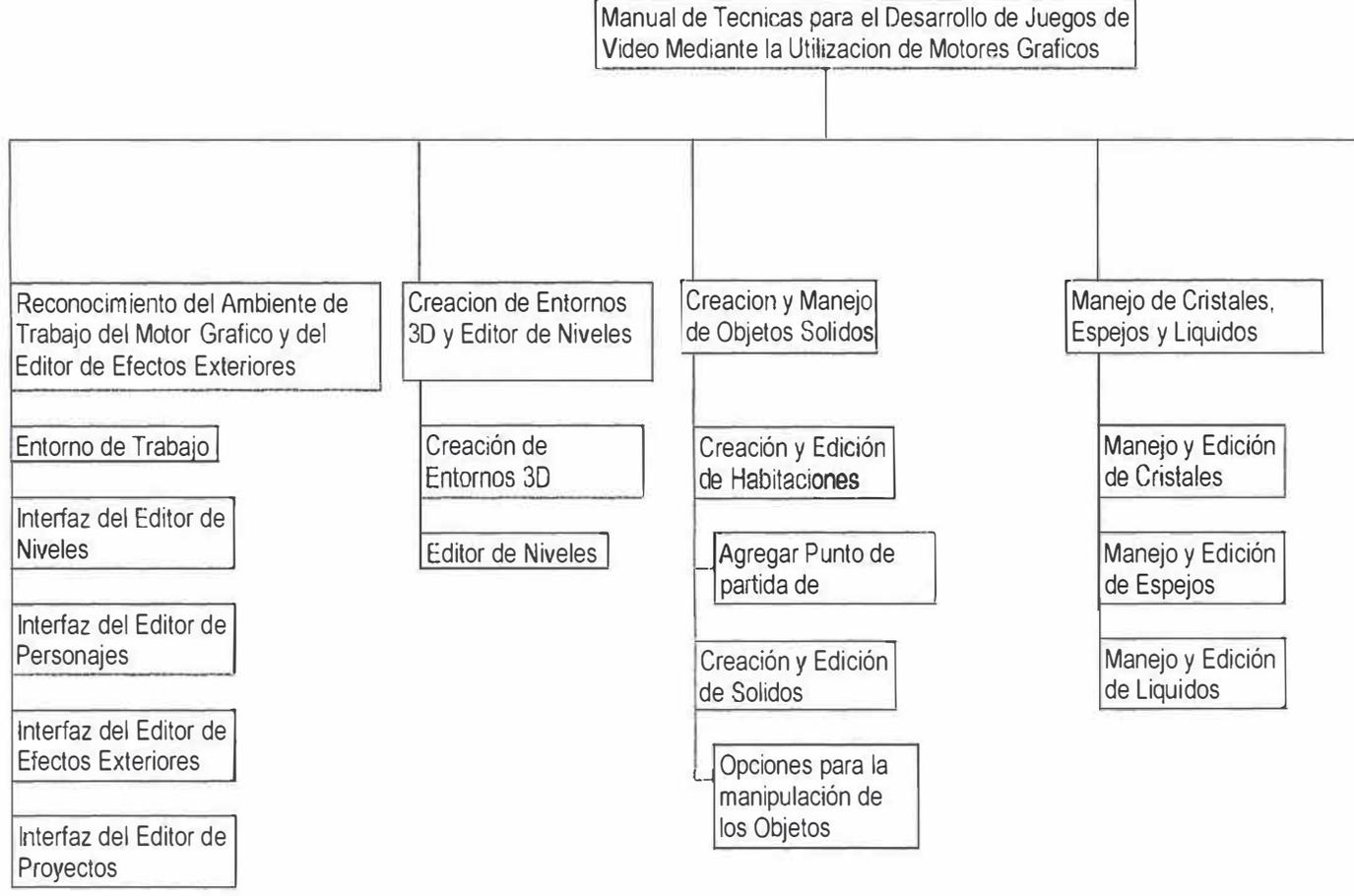


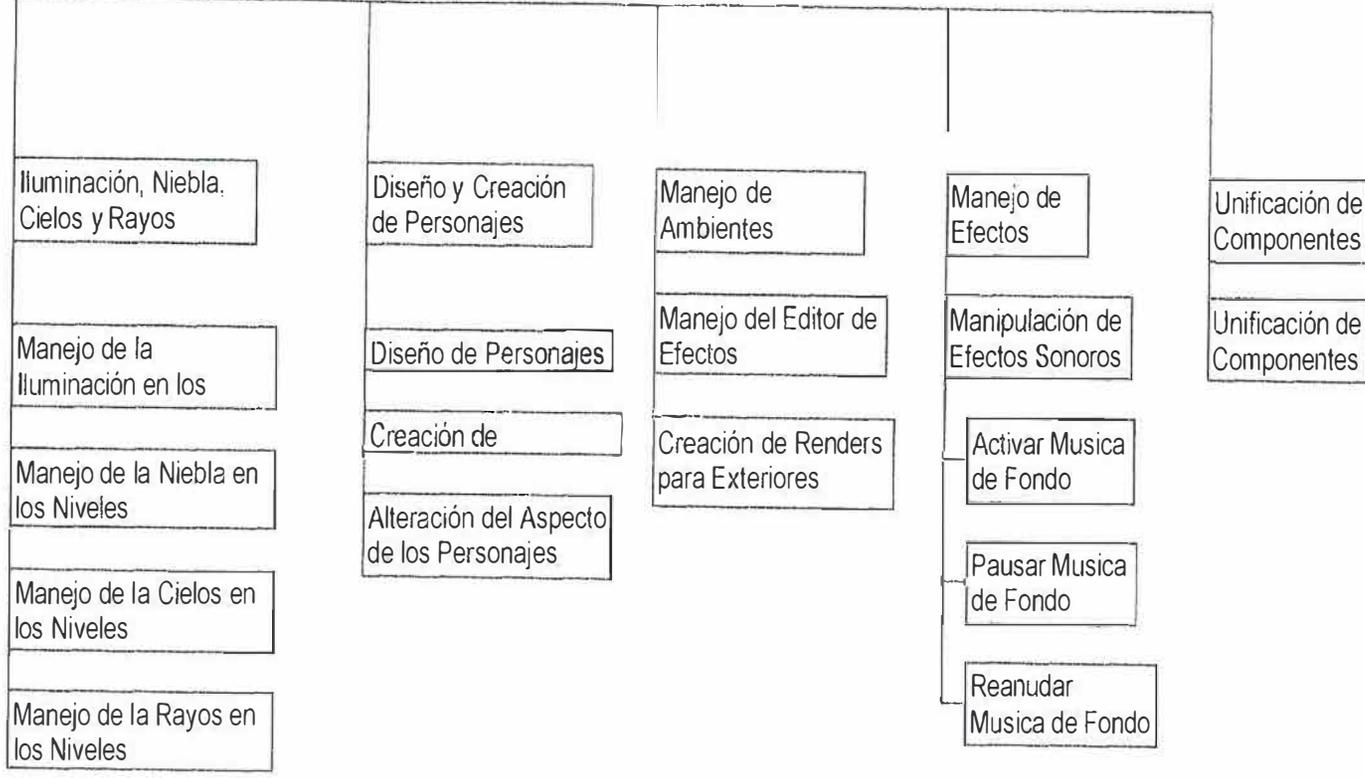




5.2 DIAGRAMA JERARQUICO FUNCIONAL.







IMPLEMENTACIÓN DE UN MANUAL DE TÉCNICAS PARA EL
DESARROLLO DE JUEGOS DE VIDEO MEDIANTE LA UTILIZACIÓN DE
MOTORES GRAFICOS 3D PARA LOS ESTUDIANTES DE LA FACULTAD DE
INGENIERIA DE SISTEMAS EN LA CORPORACION EDUCATIVA MAYOR
DEL DESARROLLO SIMON BOLIVAR

BARROS ERNESTO
MONTAÑO GERMAN
PERALTA WALTER
RODRIGUEZ RICARDO
VALEGA ALFONSO

MANUAL DE USUARIO

CORPORACIÓN EDUCATIVA MAYOR DEL DESARROLLO
SIMÓN BOLÍVAR
BARRANQUILLA
2005



INDICE

1. INTRODUCCIÓN	I
2. OBJETIVOS	I
2.1. OBJETIVO GENERAL	I
2.2. OBJETIVOS ESPECIFICOS	I
3. CONCEPTOS BÁSICOS	2
3.1 DIRECTX	2
3.2 API (Interfaz de Programación de Aplicaciones)	3
3.3 GENESIS 3D	3
3.3.1 CARACTERÍSTICAS DEL GENESIS 3D	3
3.4 TERRAGEN	4
3.5 MILKSHAPE	5
4. CARACTERÍSTICAS DEL MANUAL	5

ISO 9000-3 Normas Para La Gestión Y Aseguramiento De La Calidad.	Implementación De Un Manual De Técnicas Para El Desarrollo De Juegos De Video Mediante La Utilización De Motores Gráficos 3D.
Versión 1.0	Revisado Por El Ing. Alexis Messino

MANUAL DE USUARIO

1. INTRODUCCIÓN

Cada Videojuego es un universo diferente debido a la variedad de Géneros existentes y la combinación de los mismos, por lo tanto cada uno requiere especificaciones sobre la información vital que se necesita para poder ser utilizado por cualquier persona, pues hoy en día los juegos no son solo eso, incluyen conexión a Internet, edición y creación de niveles, por tanto se debe tener información clara, concisa y lo mas explicita posible de todas estas opciones. Para los tiempos que corren hoy, los Videojuegos son algo a común en nuestro medio, es mas podríamos decir que enmarea una nueva forma de diversión de los jóvenes que exponen sus habilidades y toda su creatividad en el desarrollo de las distintas actividades, que son exigidas por los distintos escenarios y niveles de los mismos Videojuegos. Pero también es cierto que conseguir información sobre este tipo de entretenimiento lúdico es escaso, algo que hay que evitar con este manual.

2. OBJETIVOS

2.1 OBJETIVO GENERAL.

Diseñar e implementar un manual que le permita al usuario ver y aprender las distintas pautas para desarrollar y manipular un Videojuego.

2.2 OBJETIVOS ESPECIFICOS

- Entregar un manual de fácil entendimiento a los usuarios de los videojuegos creados en Génesis 3D.
- Explicar la forma en que interactúa con el demo creado con Génesis 3D.
- Exponer las distintas herramientas que interactúan en el desarrollo de un Videojuego.

ISO 9000-3 Normas Para La Gestión Y Aseguramiento De La Calidad.	Implementación De Un Manual De Técnicas Para El Desarrollo De Juegos De Video Mediante La Utilización De Motores Gráficos 3D.
Versión 1.0	Revisado Por El Ing. Alexis Messino



ISO 9000-3 Normas Para La Gestión Y Aseguramiento De La Calidad.	Implementación De Un Manual De Técnicas Para El Desarrollo De Juegos De Video Mediante La Utilización De Motores Gráficos 3D.
Versión 1.0	Revisado Por El Ing. Alexis Messino

3. CONCEPTOS BÁSICOS

3.1 DIRECTX

Conjunto de controladores y programas que se utilizan para que, generalmente, los juegos funcionen mejor bajo Windows y, a la vez, sean más fáciles de desarrollar. Basta con que tengamos los drivers de nuestras tarjetas de video y de sonido para que cualquier juego, a través de DirectX, sepa aprovecharlas completamente.

También se puede decir que son una colección de APIs creadas para facilitar tareas relacionadas con la programación de juegos en la plataforma Microsoft Windows. El kit de desarrollo de DirectX es distribuido gratuitamente por Microsoft. Las bibliotecas de DirectX eran originalmente distribuidas por los desarrolladores de juegos con sus paquetes, pero más tarde fueron incluidas en Windows. Su última versión es la 9.0c.

DirectX incluye las siguientes APIs:

- ✓ **DirectDraw:** para dibujado de imágenes en dos dimensiones (planas).
- ✓ **Direct3D (D3D):** para representación de imágenes en tres dimensiones.
- ✓ **DirectInput:** utilizado para procesar datos del teclado, ratón, joystick y otros controles para juegos.
- ✓ **DirectPlay:** para comunicaciones en red.
- ✓ **DirectSound:** para la reproducción y grabación de sonidos de ondas.
- ✓ **DirectMusic:** para la reproducción de pistas musicales compuestas con DirectMusic Producer.
- ✓ **DirectShow:** para reproducir audio y video con transparencia de red.
- ✓ **DirectSetup:** para la instalación de componentes DirectX.

A pesar de ser desarrollado exclusivamente para la plataforma Windows, una implementación de su API se encuentra en progreso para sistemas Unix (en particular Linux) y X Windows System conocida como WineX, desarrollada por la empresa de software Transgaming y orientada a la ejecución de juegos desarrollados para Windows bajo sistemas Unix.

ISO 9000-3 Normas Para La Gestión Y Aseguramiento De La Calidad.	Implementación De Un Manual De Técnicas Para El Desarrollo De Juegos De Video Mediante La Utilización De Motores Gráficos 3D.
Versión 1.0	Revisado Por El Ing. Alexis Messino



ISO 9000-3 Normas Para La Gestión Y Aseguramiento De La Calidad.	Implementación De Un Manual De Técnicas Para El Desarrollo De Juegos De Video Mediante La Utilización De Motores Gráficos 3D.
Versión 1.0	Revisado Por El Ing. Alexis Messino

3.2 API (Interfaz de Programación de Aplicaciones)

Es un conjunto de especificaciones de comunicación entre componentes software. Representa un método para conseguir abstracción en la programación, generalmente (aunque no necesariamente) entre los niveles o capas inferiores y los superiores del software. Uno de los principales propósitos de una API consiste en proporcionar un conjunto de funciones de uso general, por ejemplo, para dibujar ventanas o iconos en la pantalla. De esta forma, los programadores se benefician de las ventajas de la API haciendo uso de su funcionalidad, evitándose el trabajo de programar todo desde el principio.

3.3 GENESIS 3D

Genesis3D es programa que nos permite generar modelos renderizados en 3D y que hace rendir el ambiente para todas sus necesidades en tiempo real 3D. Si usted modifica la fuente del motor de Genesis3D, extraiga cualquier porción de ella, o si usted utiliza una versión modificada de la fuente del motor, usted debe hacer su código de fuente disponible para otros para el uso bajo mismos términos que le hemos concedido. Si usted no desea hacer su código de fuente disponible usted debe obtener un pagado separado licencia de eclipse.

3.3.1 CARACTERÍSTICAS DEL GENESIS 3D

Para que el proyecto de desarrollo de VIDEOJUEGOS sea prolijo debe contarse con un motor gráfico que sea potente en el aspecto de manejo de los distintos efectos y herramientas alternas para la creación de Ambientes Gráficos 3D.

- ✓ El Motor Gráfico GENESIS 3D esta desarrollado por Eclipse Entertainment.
- ✓ La plataforma que lo soporta es Windows9X y NT. Bajo licencia libre y derecho a modificar el código fuente.
- ✓ Las APIs que soporta son Direct3D y Glide. Renderizado por Portales, árboles BSP, radiosidad y LOD. Incorpora luces multicolores y dinámicas.
- ✓ El Génesis trabaja nativamente con el formato de ficheros de animación de 3Dstudio MAX. Texture mapping, textura translúcida, morphing de texturas, texturas animadas y procedurales.
- ✓ Trabaja con Sprites 3D, sombras dinámicas, jerarquía de objetos, detección de colisiones.

ISO 9000-3 Normas Para La Gestión Y Aseguramiento De La Calidad.	Implementación De Un Manual De Técnicas Para El Desarrollo De Juegos De Video Mediante La Utilización De Motores Gráficos 3D.
Versión 1.0	Revisado Por El Ing. Alexis Messino



ISO 9000-3 Normas Para La Gestión Y Aseguramiento De La Calidad.	Implementación De Un Manual De Técnicas Para El Desarrollo De Juegos De Video Mediante La Utilización De Motores Gráficos 3D.
Versión 1.0	Revisado Por El Ing. Alexis Messino

Características especificadas:

- ✓ "Render" excepcionalmente rápido.
- ✓ Tratamiento de la dispersión de la luz (radiosidad).
- ✓ Luces con intensidad variable según diversos patrones.
- ✓ Sombras dinámicas.
- ✓ Niebla dinámica.
- ✓ Espejos dinámicos.
- ✓ Efectos de agua dinámicos.
- ✓ Geometría de objetos translucidos para cristales u otros efectos.
- ✓ Mapeado esférico para cielos y horizontes.
- ✓ Posicionamiento y atenuación 3D de los sonidos.
- ✓ Superposición de imágenes 2D.
- ✓ Resolución de pantalla de 640x480 hasta 1024x768 a pantalla completa.

3.4 TERRAGEN

Terragen es un software gráfico que nos permite generar paisajes del freeware para Microsoft Windows y Apple Macintosh desarrollado y publicado por PlanetSide Software. Puede ser utilizado para crear renderings y animaciones de paisajes con un realismo sorprendente.

Terragen nos permite la creación, manejo y desarrollo de efectos exteriores, que manejarán varios aspectos como la iluminación, imagen, agua, nubes, terrenos, atmósfera, y controles de render, permitiéndonos así crear fondos con paisaje y cielo que sería lo que veríamos a través de las ventanas, o techos y paredes que hayamos definido como abiertas al exterior en nuestro escenario. Los elementos exteriores son muy importantes, pues estos son los que se encargan de crear la sensación de profundidad más allá del nivel en el que se localizan los personajes, pero cabe destacar que los personajes no interactúan con estos ambientes, pues estos están ahí como complemento de los Videjuegos para crear una atmósfera.

ISO 9000-3 Normas Para La Gestión Y Aseguramiento De La Calidad.	Implementación De Un Manual De Técnicas Para El Desarrollo De Juegos De Video Mediante La Utilización De Motores Gráficos 3D.
Versión 1.0	Revisado Por El Ing. Alexis Messino

ISO 9000-3 Normas Para La Gestión Y Aseguramiento De La Calidad.	Implementación De Un Manual De Técnicas Para El Desarrollo De Juegos De Video Mediante La Utilización De Motores Gráficos 3D.
Versión 1.0	Revisado Por El Ing. Alexis Messino

de los estudiantes como a nivel grupal dentro de las distintas comunidades que están conformadas por las distintas universidades del Atlántico y del país en general.

Además este es un manual de fácil uso y que está disponible para cualquier persona sin importar el área de desarrollo u profesión en el que este se desenvuelva, ya que en este se manejan distintas introducciones a los temas que se relacionan con el manejo y desarrollo de Motores Gráficos 3D.

ISO 9000-3 Normas Para La Gestión Y Aseguramiento De La Calidad.	Implementación De Un Manual De Técnicas Para El Desarrollo De Juegos De Video Mediante La Utilización De Motores Gráficos 3D.	6
Versión 1.0	Revisado Por El Ing. Alexis Messino	