



CCADET

CENTRO DE CIENCIAS APLICADAS Y DESARROLLO TECNOLÓGICO
UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

Informe Técnico del desarrollo del Piloto Para el Museo Virtual del Patrimonio Cultural Universitario UNAM

Desarrollo tecnológico

Francisco Cabiedes Contreras, Andrea Miranda Vitela y José Luís Pérez Silva

CCADET 2012

Desarrollo de Software

Financiado por el Patronato Universitario (UNAM)

Piloto Para el Museo Virtual del Patrimonio Universitario UNAM

Francisco Cabiedes Contreras, Andrea Miranda Vitela y José Luís Pérez Silva

CCADET, UNAM

RESUMEN

El Museo Virtual del Patrimonio Universitario UNAM se basa en la idea de espacios virtuales (3D en tiempo real) que permitan el disfrute y apropiación simbólica del patrimonio cultural de la UNAM, rescatando la posibilidad de mostrar el discurso curatorial y el trabajo museográfico. El sistema está asociado a un sitio Web que se dedica a la construcción de una comunidad virtual alrededor de los contenidos de la base de datos del Patrimonio y el medio “ad oc” para la presentación de las obras.

Se presenta el sistema informático que sostiene las actividades del proyecto piloto del Museo Virtual.

Introducción.....	4
Definiciones.....	4
1 Objetivos.....	4
1.1 Público.....	5
1.2 Contexto de uso.....	5
3 Contenidos.....	5
4 Productos.....	6
5 Sitio web.....	6
6 Base de datos.....	7
7 Programas de unidad de comunicación y de ambiente virtual.....	7
8 Concurso arquitectónico para el museo virtual.	7
9 Pruebas de usabilidad del sistema web.....	9
10 Conclusiones.....	9
Anexo A: Contenidos.....	10
Anexo B: Arquitectura del sitio web.....	12
Anexo C: Diseño gráfico del sitio web.....	15
Anexo D: Infraestructura del servidor.....	17
Anexo E: Organización final de la información en el servidor.....	18
Anexo F: Pantallas del sitio.....	19
Anexo G: Diccionario de datos de la base de datos.....	23
Anexo H: Programas unidad de comunicación y de ambiente virtual.....	29
Anexo I: Convocatoria al concurso de arquitectura del museo virtual.....	117
Anexo J: Productos académicos derivados del proyecto.....	120

INTRODUCCIÓN

El Proyecto del Museo Virtual del Patrimonio Cultural de la Universidad Nacional Autónoma de México fue desarrollado por Esther de la Herrán. Los detalles de producción se encuentran en el archivo MVMaestro.doc. El piloto del proyecto se centra en crear el software para una muestra de aprox. 500 piezas provenientes del acervo del MUCA campus.

DEFINICIONES

Museo Virtual: Por medio del museo virtual se podrán conocer las diferentes exposiciones de los museos universitarios. Además el usuario (normal o curador) tendrá la opción de crear su propio museo a partir del catálogo del patrimonio cultural universitario y presentar sus propias exposiciones. Un museo imaginado creado por un usuario podrá ser "visitado" por otros usuarios. Así el museo virtual podrá exponer las obras del patrimonio cultural universitario en todas las exposiciones que los usuarios puedan imaginar.

Sistema Web: Sistema que comprende todas las partes que conforman el Museo Virtual. Estas partes se pueden dividir en Base de Datos, Sitio Web y Sistema de Ambiente Virtual.

Base de Datos: Reservorio que controla la información de las piezas del Patrimonio Cultural Universitario y su exhibición en el sitio web y en los programas del sistema de ambiente virtual. También administra las acciones de los usuarios.

Sitio Web: Interfaz pública del Sistema Web. Este sitio se optimizó para un usuario no experto y comprende ramas de información general, para investigadores, curadores, museólogos o público más especializado y personalizada para usuarios que generen exposiciones

Sistema de Ambiente Virtual: que comprende el programa que hace la comunicación entre la PC del usuario y el servidor (Unidad de Comunicación llamado Oráculo). Mediante la Unidad de Comunicación se ejecutan los programas que permiten ver las exposiciones (Visor llamado Apolo) y crear las exposiciones (Editor llamado Dionisios). Todos estos programas son propietarios del equipo técnico para poder garantizar la seguridad de los datos de la Base de Datos.

1. Objetivos

El programa piloto se propone desarrollar el software que permita crear un museo virtual con capacidad para exposición de piezas seleccionadas del acervo del MUCA campus. Se pretende generar el modelo que demuestre las posibilidades y alcances de la tecnología informática puesta al servicio del usuario, del quehacer museal y del conocimiento, valoración y difusión del Patrimonio Cultural Universitario.

Este “museo” estará en Internet permanentemente y usará las posibilidades que brindan los Multimedios en línea mediante un Sistema Web.

1.1 Público

Los productos desarrollados tendrán como objetivo diferentes tipos de usuarios. En general se consideran a:

Publico en general

Estudiantes universitarios

Académicos especializados

1.2 Contexto de uso

Se considera que la visita a las exposiciones del museo, realizadas tanto por curadores como por el público en general, la harán los usuarios mediante Internet en cualquier computadora personal que permita la instalación del Sistema de Ambiente Virtual descrito anteriormente. La descarga del software se hará mediante registro de una dirección de correo válida. Para el diseño de exposiciones los usuarios usarán la herramienta de edición accesible. Una vez diseñada la exposición quedará guardada en el servidor y registrada en la Base de Datos para poder ser vista por cualquier usuario mediante el Visor.

3 CONTENIDOS

La responsabilidad de la investigación de los contenidos recae en el personal de la Dirección General de Artes Visuales bajo la coordinación de ERdH. La descripción de los criterios de selección e intención de los mismos se presenta en el anexo 1.

La Administradora de Contenidos (AdC) es el enlace entre la Coordinadora General y el equipo de desarrollo. Entre las tareas de AdC están el decidir el diseño y coordinación del desarrollo de partes del sistema web y su integración, así como la organización de la información y la supervisión de su inclusión en la base de datos y en el sitio web.

Como las piezas del patrimonio no son identificadas unívocamente por su número de inventario UNAM, se diseña una llave que coordine la información sobre cada pieza. Se organizan los datos en tablas con hasta 65 datos de cada pieza. Además se guardan en carpetas auxiliares contenidos como: Descripción, Biografía del Autor, Bibliografía relevante a la pieza, Contexto, Crítica y Glosario cuyos textos se despliegan en formato pdf y que pueden contener hipervínculos tanto a otros documentos en la misma base como a sitios externos en Internet.

Además se realizan los catálogos de datos de exposiciones montadas y se organiza la información que se despliega en el sitio web. También se organizan las pantallas que

organizan las búsquedas de piezas para hacer una exposición y las pantallas que dan servicios generales al usuario del sitio web.

Finalmente el personal del MUCA Campus realiza una auditoria a todos los textos incluidos en el sitio web y presenta un reporte de los cambios que deberán hacerse antes de abrir el sistema al público, Este reporte impreso con formato propio de 110 pg. se anexa al final de este documento.

4 PRODUCTOS

Los productos desarrollados se describen a continuación:

Sitio Web	Portal dinámico que permita la interacción de todo tipo de usuario con el Museo Virtual
Base de Datos	Contiene la información de las piezas del Patrimonio Cultural, de la información desplegada en el Sitio Web y de los usuarios registrados
Visor	Programa que despliega un ambiente tridimensional en el que se simulan las exposiciones de piezas del Patrimonio Cultural
Editor	Programa que permite crear una exposición a partir de una selección en el sitio web de piezas de la Base de Datos “colocando” estas en el ambiente tridimensional que se despliega en el visor
Fotografía especializada 3D	De 100 piezas seleccionadas por los museólogos del MUCA se hizo fotografía estereoscópica que se convierte en un “fragmento de película estereoscópica” que muestra la pieza como si rotara en un eje para poder ser apreciada en 3D
Unidad de Comunicación	Programa que controla con seguridad el despliegue del Visor y del Editor y la transacción de información visual entre el servidor y la computadora del usuario
Museo Virtual	Concepto que aglutina los productos de software (Sistema Web) formado por: Sitio Web, Base de Datos, Visor y Editor; y servidor que aloja el software. También se consideran parte los contenidos generados por investigadores para los objetos del Patrimonio Cultural de la UNAM y el personal encargado de la administración y mantenimiento de todo lo antes descrito.

5 SITIO WEB

La arquitectura del sitio queda descrita en el anexo B.

El diseño gráfico se muestra en el anexo C.

Los detalles de la infraestructura del servidor se dan en el anexo D

La organización final de la información en el servidor se muestra en el anexo F

El resumen completo de las pantallas que forman el sitio se muestra en el anexo F

6 BASE DE DATOS

La base de datos se desarrollo a partir de hojas de datos que fueron proporcionadas por los responsables de contenidos.

El diccionario de datos tanto de las tablas de datos como de las tablas que organizan y controlan las actividades de los usuarios del sitio web se muestra en el anexo G.

7 PROGRAMAS DE UNIDAD DE COMUNICACIÓN Y DE AMBIENTE VIRTUAL

Los detalles de programación de los tres programas desarrollados: Oráculo, Apolo y Dionisos se muestran en el anexo H.

Para la arquitectura visual del museo virtual se convocó a la comunidad universitaria a presentar propuestas.

8 Concurso arquitectónico para el Museo Virtual.

El entorno tridimensional del MV es un espacio vacío, donde no existe la gravedad y puede ser aprovechado de infinitas maneras, sin embargo, el trabajo museal requiere, según las diferentes versiones del cuerpo de museólogos y curadores que sirvieron de asesores, requería de una forma de acotación espacial que permitiera manejar el capitulado museográfico que es producto del discurso generado por el curador para una exposición.

Desde el principio del proyecto, se pensó en buscar la intervención de la comunidad universitaria para el diseño de una estructura arquitectónica que pudiera aprovechar, tanto la infinitud del espacio, como la falta gravedad y la posibilidad de navegar libremente por todo el espacio.

Se lanzó una convocatoria abierta, revisada por el presidente del Comité Asesor de Cómputo UNAM, en ese entonces, el Dr. Alejandro Pisanti.

La publicación se realizó vía; GacetaUNAM, se generaron carteles y se publico, durante dos semanas en la página de la UNAM.

Para definir al ganador del concurso se formó un jurado compuesto por diversas personalidades tanto del arte, como de la arquitectura y la visualización tridimensional en tiempo real, dicho comité se formó con los siguientes miembros (en estricto orden alfabético):

- Marco Barrera Bassols: Museólogo Ex responsable del Museo de las Culturas Populares
- Manuel González-Casanova Almoina: Jefe del departamento Multimedia de la DGDC.

- Genoveva Lucet: Directora de Cómputo para la Investigación de DGSCA, quien, a su vez es la responsable del Observatorio de Visualización “Ixtli”.
- Jorge Reynoso: Director del Espacio Cultural Siqueiros
- Rafael Sámano: miembro del equipo museológico del MUCA campus

Este grupo congregó las dos visiones que se requerían para la selección de un modelo tridimensional, que pudiera cumplir con las normas, tanto de diseño arquitectónico, como de diseño computacional. De tal forma que el modelo fuera lo más adecuado, museológicamente hablando, y pudiera ser utilizado por el sistema 3D, sin que el desempeño de la máquina del usuario se viera comprometido.

Al término de la convocatoria, se recibieron tres proyectos arquitectónicos.

El jurado, de manera unánime, decretó el concurso “desierto”. La razón se fundamenta en la falta de originalidad y funcionalidad de los modelos. Dos de los concursantes presentaron estructuras arquitectónicas derivadas de museos reales, esto es; diseñaron salas, algunas con puertas abatibles, techos e incluso auditorios con sala de proyección cinematográfica. El tercero, fue un diseño tubular que, obstruía la visión y se orientaba con conceptos como arriba y abajo.

Dada la respuesta unánime, se procedió a formar un equipo interdisciplinario con el sólo propósito de diseñar la arquitectura, sabiendo cuales podrían ser las necesidades del trabajo museal y el desempeño del sistema. Este equipo se conformó por:

- Esther de la Herrán: Responsable del proyecto por parte de la DGAV
- Genoveva Lucet: Directora de Cómputo para la Investigación de DGSCA, quien, a su vez es la responsable del Observatorio de Visualización “Ixtli”.
- Jorge Reynoso: Director del Espacio Cultural Siqueiros
- Francisco Cabiedes: Coordinador del desarrollo del sistema informático del MV.

Tras cinco reuniones de trabajo, se optó diseñar una estructura capaz de respetar la visión libre de la exposición y que a su vez, permitiera, tanto la segmentación por espacios acotados, como la posibilidad de utilizarlo como una unidad completa.

El resultado se muestra en la figura 1. Se trata de una estructura tipo “Cubo de Rubick”, relajada. Esto nos permite que el usuario tenga alguna referencia personal del modelo, y al ser una estructura relajada, permite construir de manera libre; corredores y zonas aisladas, sin perder la visual completa de la exposición o entorpecer el desempeño del sistema.

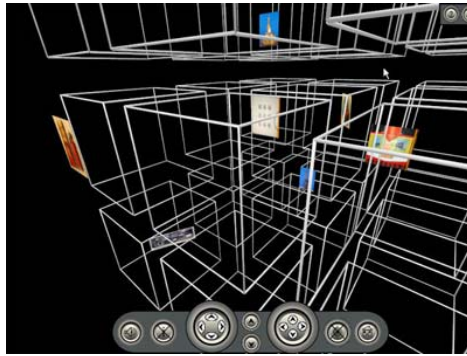


Figura 1 Muestra de una exposición montada en el edificio tipo "Cubo de Rubick"

9 PRUEBAS DE USABILIDAD DEL SISTEMA WEB

El sistema se prueba con usuarios para detectar fallas y potenciales problemas tanto en la usabilidad del sistema como en su impacto visual.

Estas pruebas se llevan a cabo con la presencia de los auditores independientes al desarrollo del sistema y se graban en video.

Todos los detalles de las pruebas, su análisis y los resultados se muestran en el reporte "Resultado de las pruebas" de menos de 10 pg. que se anexa al final de este documento.

10 Conclusiones

En el presente trabajo se ha presentado un sistema que permite la creación de espacios de acción museal y permite su distribución vía Internet.

El sistema permite la creación de exposiciones que se van desde las basadas en discursos curatoriales profesionales, la reproducción virtual de museos reales, a las visiones construidas por la comunidad.

El Museo Virtual podría construir una comunidad dedicada a la exploración de los ambientes en la apropiación del arte y proceso de exhibición y acción museal.

Anexo A: Contenidos

INTRODUCCIÓN

DESDE UNA PERCEPCIÓN MUSEOLÓGICA

La encomienda de desarrollar un “piloto” del Museo virtual del Patrimonio Cultural Universitario llevó implícito desde sus inicios la idea de concebirlo bajo la mirada del quehacer museal.

Constituyo un ejercicio y una aspiración por trascender el modo “plano” con el que otros museos virtuales presentan sus colecciones. Una oportunidad para incorporar las prácticas museísticas en el terreno virtual.

En este sentido, la noción de museo que sustentó el desarrollo de este Proyecto, partió del hecho de considerarlo un medio y un difusor de la memoria social, de fragmentos del mundo contenidos en testimonios materiales, con un papel clave en la transmisión de la herencia cultural.

De ahí que el Museo Virtual se planteó como reconstructor de la memoria visual del Patrimonio Cultural Universitario, comunitaria en principio y social por extensión, que no ha existido como tal en su conjunto. Una memoria para el futuro, activa y operativa, que actualice el pasado en todo lo presente.

Comprendido de esta manera, se puede entender, porqué el fundamento esencial del MV esta puesto en el interés por el “otro”, por lo diverso, sea ese “otro” presente o el pasado.

Funciones museísticas

Mediante el uso y generación de tecnologías informáticas, el Museo Virtual está pensado para preservar, exhibir, comunicar y difundir la memoria viva del Patrimonio Cultural de la UNAM en diálogo con comunidades, sean universitarias, nacionales e internacionales

Preservar. En esta perspectiva, la tarea inicial para generar la memoria visual del Patrimonio Cultural Universitario se asumió, dentro de un orden clasificatorio, con criterios dirigidos a la preservación intelectual de los bienes patrimoniales.

Este planteamiento condujo, en primera instancia, al recononocimiento de que el espacio virtual es el lugar idóneo para asociar bienes culturales con el conocimiento y los saberes que en torno a ellos se hayan ido generando. En este sentido, este espacio, es un espacio abierto al infinito.

EXHIBIR. Distinguir la presencia del Museo Virtual de la UNAM en Internet, llevó a tomar la decisión de desarrollar herramientas que le permitan al usuario o al profesional de museos, generar sus propios recorridos temáticos o rizomáticos a partir de los bienes agrupados en el Patrimonio Cultural Universitario.

Es darle a los usuarios la posibilidad de irrumpir en el espacio del Museo Virtual para Imaginar y “construir” exposiciones y con ello, además, hacer dinámico, plural y ligado a la diferencia este conjunto de bienes culturales universitarios.

Comunicar. Al considerar las amplias posibilidades que en la actualidad ofrecen las tecnologías de la información, se reveló otra espléndida oportunidad para extender el

campo de información en el Museo Virtual. Se decidió así incorporarle accesos a la actividad de los museos presenciales de la UNAM.

Una propuesta asumida con un doble sentido: preservar la memoria de la actividad museística universitaria y ofrecer la opción de los recorridos virtuales a los espacios y exposiciones que presenten los museos universitarios.

Difundir. La naturaleza pública del Museo, no sólo como espacio en el ambiente virtual, sino también como un lugar para el mutuo reconocimiento en la identidad compartida, se sustenta en su capacidad de generar modos de exposición y distribución social del conocimiento, en donde el entorno opera a favor de una multiplicación exponencial de los canales de difusión cultural.

El museo: espacio de confluencias interpretativas e intercambio disciplinar

El reto para definir la dinámica central del museo, pensando en fomentar experiencias personales y exploratorias, significó reconocerlo como un espacio cuya vocación permite seleccionar y ordenar, de múltiples formas, los objetos culturales y con ello el conocimiento, dando pie a la generación de diversas lecturas y contextos culturales en torno a los objetos elegidos.

Con esto, el Museo Virtual se ofrece como expresión del modo en que los seres humanos establecen sus relaciones con el saber, convirtiéndose así en un espacio en constante proceso de construcción, dada su capacidad de interactuar con múltiples ofertas simbólicas que lleguen desde fuera.

Es una idea que partió de considerar, que hoy en día, existe la conciencia de que el individuo es un ser de múltiples capacidades y por lo tanto, con la necesidad de disponer de alternativas para cada una de ellas, posibles en mayor medida en un ambiente virtual.

Es en la naturaleza misma del Patrimonio Cultural Universitario, acompañado de una constante labor museológica, donde el Museo encuentra su razón de ser, no sólo por su diversidad y amplitud, sino por ser un detonador del encuentro de ideas y concepciones de objetos que se pueden confrontar, dialogar o fecundarse entre sí.

Esta flexibilidad en la movilidad de objetos, datos e ideas favorece la comunicación museística, concebida como una dinámica y cambiante «creación de significado» en continua interacción con las preferencias de usuarios con culturas e intereses diversos que, además, evolucionan continuamente.

El museo y la experiencia museal

De modo que para el usuario, el Museo Virtual de la UNAM, abre la posibilidad de que, al ofrecerle el acceso a “otras” lecturas del Patrimonio Cultural, él pueda encontrar en el mensaje recibido nuevos sentidos a partir de su propia experiencia personal.

El museo y la gestión del conocimiento

La necesidad de formar un capital de conocimientos relativos a la diversidad del quehacer cultural asociado al Patrimonio Cultural Universitario, requirió iniciar la sistematización de la información mediante la formación de bases de datos y el uso de las existentes.

Anexo B: Arquitectura del sitio web

Mapa de navegación Museo Virtual UNAM

Por: Alma Rivas y Libia Eslava

13 / 02 / 07

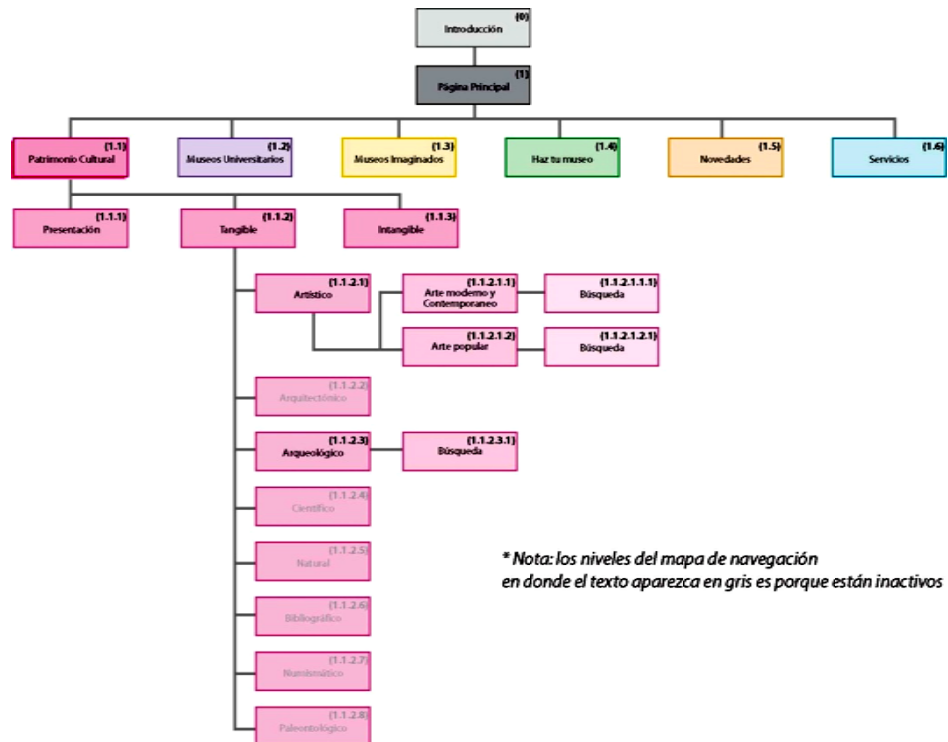


Figura B. 1. Mapa de navegación donde se desarrolla la ruta de "Patrimonio Cultural"

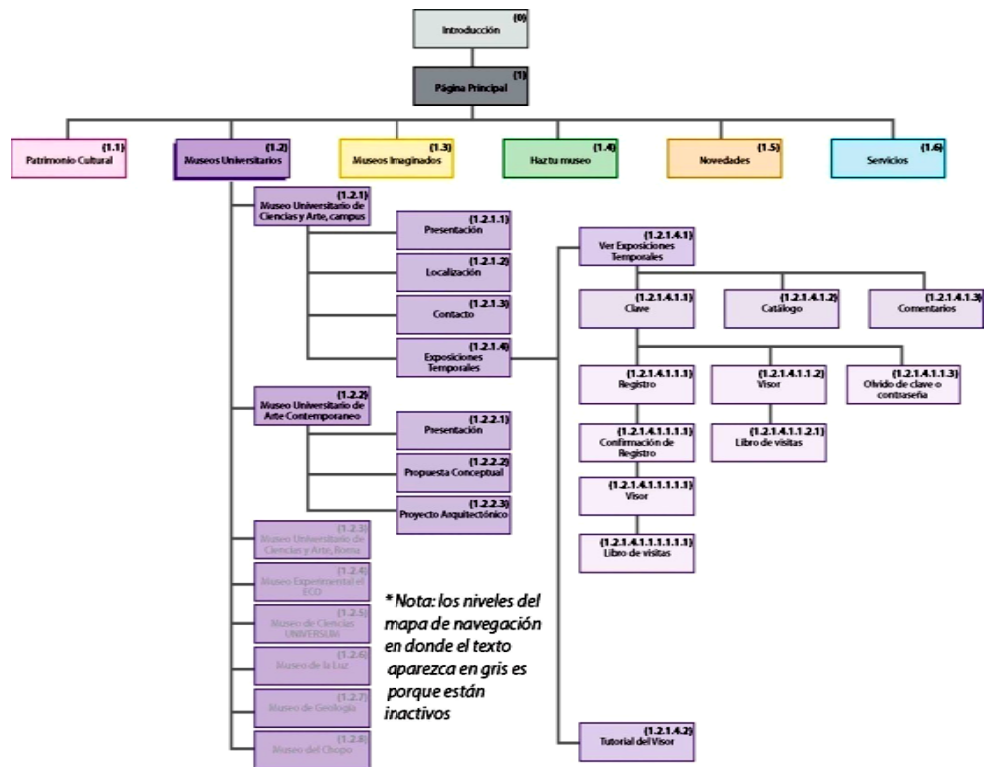


Figura B. 2. Mapa de navegación donde se desarrolla la ruta de “Museos Universitarios”

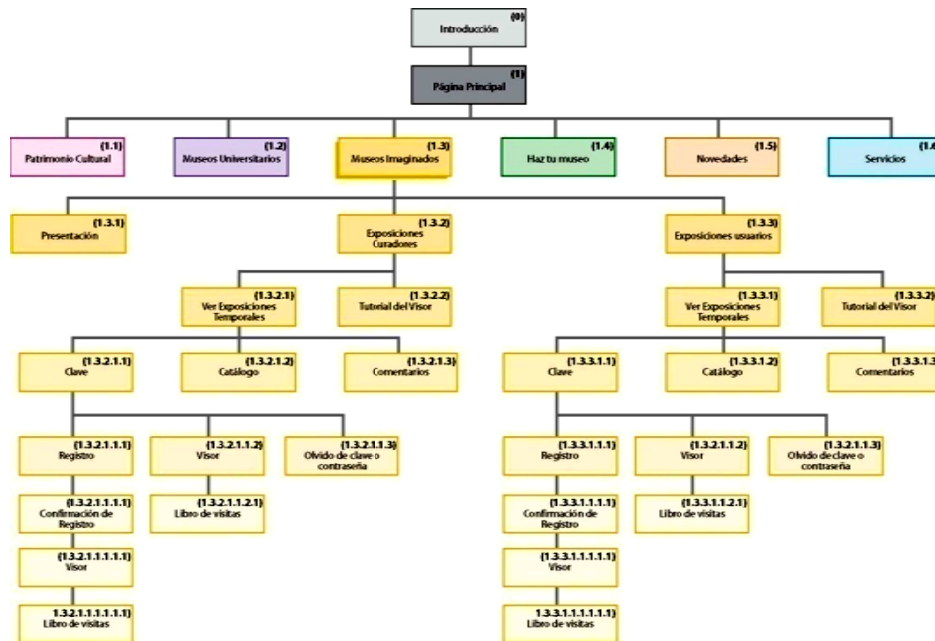


Figura B. 3. Mapa de navegación donde se desarrolla la ruta de “Museos Imaginados”

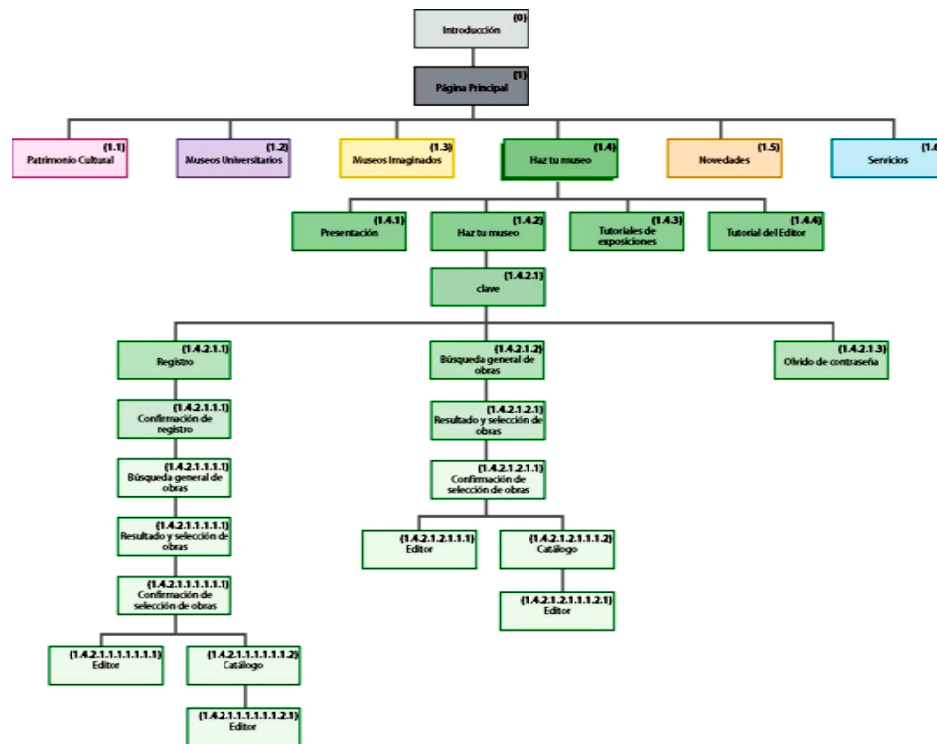


Figura B. 4. Mapa de navegación donde se desarrolla la ruta de “Haz tu Museo”

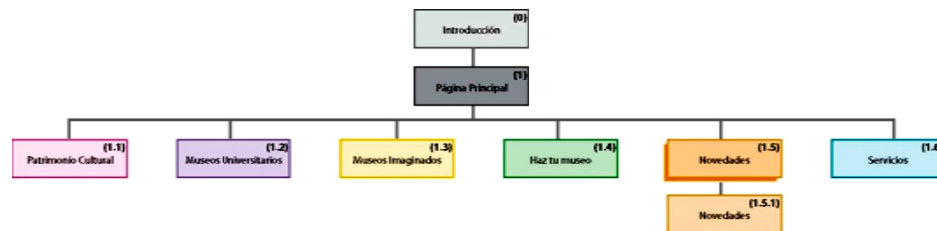


Figura B. 5. Mapa de navegación donde se desarrolla la ruta de “Novedades”

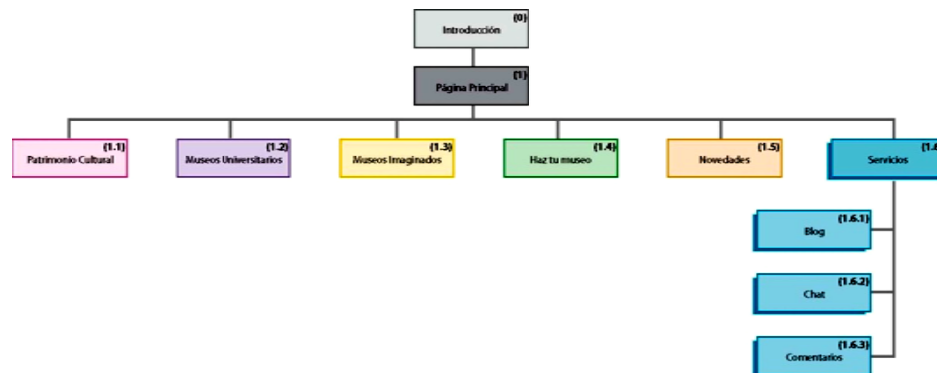


Figura B. 5. Mapa de navegación donde se desarrolla la ruta de “Servicios”

Anexo C: Diseño gráfico del sitio web

Justificación Gráfica (L. ESLAVA.)

Este portal se diseñó gráficamente con el objetivo de satisfacer las necesidades de información de los usuarios. Esto fue posible con una adecuada arquitectura web que considero múltiples aspectos: diseño gráfico, navegación, interactividad, estructura, información, usabilidad, etc.

Se realizó una diagramación que permite organizar la información de manera clara y coherente que facilita la exploración. Esta estructura, logra que todas las opciones del menú estén siempre presentes, para que los usuarios puedan navegar a cualquier parte del sitio sin tener que regresar a la página principal; además destaca los elementos de interacción con los usuarios, y la retroalimentación, bien para recoger sus opiniones o para ofrecerles servicios.

La optimización de los elementos para el portal que se utilizaron permite una descarga rápida de la información y de las imágenes.

Se manejaron colores neutros que además de ser agradables a toda la audiencia, cumple con las expectativas de funcionalidad y estética del portal.

Se utilizaron bloques de texto breves con una tipografía de palo seco, con un ancho de línea e interletraje adecuado, para su presentación en pantalla para mejorar la legibilidad.

El portal del Museo Virtual UNAM se presenta con suficiente flexibilidad para adaptarse a los distintos tipos de usuarios, el diseño en general de las páginas es sobrio. La información es fácil de localizar y de una forma previsible; en cada nueva página sabemos dónde encontrar los elementos de navegación, simplemente se creó una estructura de navegación intuitiva y fácil.



Figura C. 1. Logo del Sitio



Figura C. 2.Diseño de Pantalla principal



Figura C. 3.Diseño de Pantalla de contenidos

Anexo D: Infraestructura del servidor

Software Requerido

- Sistema operativo Red Hat (freeware)
- Servidor Web Apache con el Módulo del interprete PHP (freeware)

Base de datos

- DBMS MySQL para Linux (freeware)

Características de la base de datos

- Esquema de datos relacional

Leguajes de desarrollo

- HTML
- JavaScript
- PHP

Descripción

El sistema Web es el punto en donde el usuario gestionará la creación y acceso de museos virtuales, además aquí encontrará la divulgación de museos desarrollados por otros usuarios.

Mediante el apoyo de una base de datos el sistema Web se encarga de llevar un registro de usuarios inscritos, su actividades dentro del sistema, así como de la información necesaria para que los usuarios puedan visualizar los museo desarrollados. Una parte importante es la gestión del patrimonio universitario dispuesto en el sistema, mediante el cual se llevarán a cabo la creación de los museos virtuales. El sistema Web es el encargado de proporcionar las obras del patrimonio universitario al visor de museos, al editor de museos y a la interfaz Web para su divulgación en Internet.

Anexo E: Organización final de la información en el servidor

Estructura de Archivos del sitio Museo Virtual ordenados alfabéticamente.

Carpeta	Descripción de contenido
acceso	Contiene los archivos para el inicio de sesión de los usuarios y la autenticación de los mismos por parte del sistema.
Apolo.zip	Archivos de instalación y documentación de Apolo
chat	Archivos para implementar el servicio de chat dentro del sitio.
clases	Archivos con las clases del sistema
Dionisios.zip	Archivos de instalación y documentación de Dionisios.
elementos	Contiene archivos .html que conforman el menú de navegabilidad del sitio.
Exposiciones.zip	Contiene documentación acerca de cómo crear exposiciones de usuario a través del sitio y de las herramientas disponibles.
fichas	Archivos .html con la ficha técnica de las páginas principales del sitio.
imagenes	Directorio con las imágenes del sitio
include	Contiene archivos que se invocan en más de un documento durante la funcionalidad del sitio.
index.php	Archivo de inicio del Museo Virtual.
muac.swf	Animación en flash del sitio.
nivel1	Contiene archivos .pdf con información de las obras, autores, crítica de los usuarios, bibliografías , biografías y contextos
nivel2	Archivos en formato .pdf para consultar la información detallada de los obras, autores, bibliografías, biografías
objeto	Contiene las imágenes de las obras que están registradas.
Oraculo.zip	Archivos de instalación y documentación de Oráculo.
plantillas	Directorio con archivos que conforman la estructura del sitio en cuanto a diseño y distribución de contenidos de las distintas páginas.
presentaciones	Archivos .html con el texto de las páginas principales de cada sección del sitio.

Tabla E. 1 Listado y contenidos de las carpetas y archivos en el servidor del Museo irtual

Anexo F: Resumen completo de las pantallas del sitio

Se en listan todas las páginas PHP y HTML que se encuentran en el museo

Menú	Nivel	Subnivel	Search	Pantalla	Título
Patrimonio Cultural	Index			1	1ª pantalla al entrar
	Index - menu			1	Menú principal
	1_1	1_1_1		2	Presentación
	1_1	1_1_2		3	Patrimonio Tangible
	1_1	1_1_3		4	Patrimonio Intangible
	1_1	1_1_2_1		5	Artístico
	1_1	1_1_2_1_1		6	Arte moderno y c
	1_1	1_1_2_1_1_1 1_1_4	1_1_2_1_1	7	Búsqueda Arte moderno y c por categoría
	1_1	1_1_2_1_1_1_1		8	Búsqueda de autores de una categoría
	1_1	1_1_2_1_1_1_1_1		9	Resultado de la búsqueda
	1_1	1_1_2_1_2		10	Arte popular
	1_1	1_1_2_1_2_1 1_1_4	1_1_2_1_2	11	Búsqueda Arte popular por categoría
	1_1	1_1_2_1_2_1_1		12	Búsqueda por subcategoría
	1_1	1_1_2_1_2_1_1_1		13	Resultado de la búsqueda
	1_1	1_1_2_3		14	Arqueológico
	1_1	1_1_2_3_1 1_1_4	1_1_2_3	15	Búsqueda Arqueológico por categoría
	1_1	1_1_2_3_1_1		16	Búsqueda por tipo de objeto
	1_1	1_1_2_3_1_1_1		17	Resultado de la búsqueda

Tabla F. 1 Se en listan todas las páginas PHP y HTML en el nivel 1_1

Menú	Nivel	Subnivel	Pantalla	Título
Museos Universitarios	1_2	1_2_1_1	1	MUCA Presentación
	1_2	1_2_1_2	2	MUCA Localización
	1_2	1_2_1_3	3	MUCA Contacto
	1_2	1_2_1_4_1	4	MUCA Exposiciones temporales
	1_2	1_2_1_4_1_1	5	MUCA <<Ver exposición>>
	1_2	1_2_1_4_1_2	6	MUCA <<Catálogo>>
	1_2	1_2_1_4_1_3	7	MUCA <<Comentarios>>
	1_2	1_2_1_4_1_4	8	MUCA Captura de comentario para libro de visitas

	1_2	1_2_1_4_2	9	MUCA Como funciona el oráculo
	1_2	1_2_1_4_3	10	MUCA Como funciona Apolo
	1_2	1_2_2_1	11	MUAC Presentación
	1_2	1_2_2_2	12	MUAC Propuesta conceptual
	1_2	1_2_2_3	13	MUAC Proyecto Arquitectónico
	1_2	1_2_3_1	14	MUCA Roma Presentación
	1_2	1_2_4_1	15	Museo Experimental El Eco Presentación
	1_2	1_2_5_1	16	Universum, Museo de las Ciencias
	1_2	1_2_6_1	17	Museo de la luz
	1_2	1_2_7_1	18	Museo de geología
	1_2	1_2_8_1	19	Museo del chopo Presentación

Tabla F. 2 Se en listan todas las páginas PHP y HTML en el nivel 1_2

Menú	Nivel	Subnivel	Pantalla	Título
Museos Imaginados	1_3	1_3_1	1	Presentación
	1_3	1_3_2_1	2	Exposiciones oficiales
	1_3	1_3_2_1_1	3	Expo oficiales <<Ver exposición>>
	1_3	1_3_2_1_2	4	Expo oficiales <<Catálogo>>
	1_3	1_3_2_1_3	5	Expo oficiales <<Comentarios>>
	1_3	1_3_2_1_3_1	6	Expo oficiales captura de comentario para libro de visitas
	1_3	1_3_2_2	7	Expo oficiales como funciona apolo
	1_3	1_3_3_1	8	Exposiciones usuarios
	1_3	1_3_3_1_1	9	Expo usuarios <<Ver exposición>>
	1_3	1_3_3_1_2	10	Expo usuarios <<Catálogo>>
	1_3	1_3_3_1_3	11	Expo usuarios <<Comentarios>>
	1_3	1_3_3_1_3_1	12	Expo usuarios captura de comentario para libro de visitas
	1_3	1_3_3_2	13	Expo usuarios como funciona apolo

Tabla F. 3 Se en listan todas las páginas PHP y HTML en el nivel 1_3

Menú	Nivel	Subnivel	TipoB	tipo	Rango	Pantalla	Título
Haz tu museo	1_4	1_4_1				1	Haz tu museo
	1_4	1_3_2_1_1				2	Registro con botón aceptar
	1_4	1_3_2_1_1_1				3	Liga inscríbete gratis
	1_4	1_3_2_1_1_3				4	Liga olvide mi contraseña
	1_1	1_4_2_1_2				5	Búsqueda general de obras
	1_1	1_4_2_1_2	2	1		6	Todos los autores

	1_1	1_4_2_1_2	1	1	1	7	Alfabético A - C
	1_1	1_4_2_1_2	1	1	2	8	Alfabético D - F
	1_1	1_4_2_1_2	1	1	3	9	Alfabético G - I
	1_1	1_4_2_1_2	1	1	4	10	Alfabético J - L
	1_1	1_4_2_1_2	1	1	5	11	Alfabético M - O
	1_1	1_4_2_1_2	1	1	6	12	Alfabético P - R
	1_1	1_4_2_1_2	1	1	7	13	Alfabético S - U
	1_1	1_4_2_1_2	1	1	8	14	Alfabético V - Z
	1_1	1_4_2_1_2	2	2		15	Todos los títulos
	1_1	1_4_2_1_2	1	2	1	16	Alfabético A - C
	1_1	1_4_2_1_2	1	2	2	17	Alfabético D - F
	1_1	1_4_2_1_2	1	2	3	18	Alfabético G - I
	1_1	1_4_2_1_2	1	2	4	19	Alfabético J - L
	1_1	1_4_2_1_2	1	2	5	20	Alfabético M - O
	1_1	1_4_2_1_2	1	2	6	21	Alfabético P - R
	1_1	1_4_2_1_2	1	2	7	22	Alfabético S - U
	1_1	1_4_2_1_2	1	2	8	23	Alfabético V - Z
	1_1	1_4_2_1_2	2	3		24	Todos los objetos
	1_1	1_4_2_1_2	1	3	1	25	Alfabético A - C
	1_1	1_4_2_1_2	1	3	2	26	Alfabético D - F
	1_1	1_4_2_1_2	1	3	3	27	Alfabético G - I
	1_1	1_4_2_1_2	1	3	4	28	Alfabético J - L
	1_1	1_4_2_1_2	1	3	5	29	Alfabético M - O
	1_1	1_4_2_1_2	1	3	6	30	Alfabético P - R
	1_1	1_4_2_1_2	1	3	7	31	Alfabético S - U
	1_1	1_4_2_1_2	1	3	8	32	Alfabético V - Z
	1_1	1_4_2_1_2	2	4		33	Todos las colecciones
	1_1	1_4_2_1_2	1	4	1	34	Alfabético A - C
	1_1	1_4_2_1_2	1	4	2	35	Alfabético D - F
	1_1	1_4_2_1_2	1	4	3	36	Alfabético G - I
	1_1	1_4_2_1_2	1	4	4	37	Alfabético J - L
	1_1	1_4_2_1_2	1	4	5	38	Alfabético M - O
	1_1	1_4_2_1_2	1	4	6	39	Alfabético P - R
	1_1	1_4_2_1_2	1	4	7	40	Alfabético S - U
	1_1	1_4_2_1_2	1	4	8	41	Alfabético V - Z
	1_1	1_4_2_1_2	2	5		42	Palabras clave
	1_1	1_4_2_1_2	1	5	1	43	Alfabético A - C
	1_1	1_4_2_1_2	1	5	2	44	Alfabético D - F
	1_1	1_4_2_1_2	1	5	3	45	Alfabético G - I
	1_1	1_4_2_1_2	1	5	4	46	Alfabético J - L

	1_1	1_4_2_1_2	1	5	5	47	Alfabético M - O
	1_1	1_4_2_1_2	1	5	6	48	Alfabético P - R
	1_1	1_4_2_1_2	1	5	7	49	Alfabético S - U
	1_1	1_4_2_1_2	1	5	8	50	Alfabético V - Z
						51	Especializada
	1_4	1_4_2_1_2_1				52	Selección de obras
	1_4	1_4_2_1_2_1_1				53	Confirmación de obras
	1_4	1_4_3				54	Tutorial de exposiciones
	1_4	1_4_4				55	Tutorial del oráculo
	1_4	1_4_5				56	Tutorial de dionisios

Tabla F. 4 Se en listan todas las páginas PHP y HTML en el nivel 1_4

Menú	Nivel	Subnivel	Pantalla	Título
Novedades	1_5	1_5_1	1	Novedades

Tabla F. 5 Se en listan todas las páginas PHP y HTML en el nivel 1_5

Menú	Nivel	Subnivel	Pantalla	Título
Servicios	1_6		1	Blog
	1_6		2	Chat
	1_6	1_6_4_1	3	Mis exposiciones
	1_6	1_6_4_2	4	Exposiciones favoritas
	1_6	1_6_4_3	5	Herramientas y tutoriales
	1_6	1_3_2_1_1_1	6	Registro
	1_6	1_6_4_5	7	Comentarios

Tabla F. 6 Se en listan todas las páginas PHP y HTML en el nivel 1_6

Anexo G: Diccionario de tipo de campo y datos de la base de datos

Se presenta el diccionario completo de las variables, su nombre, función y tipo

Patrimonio cultural

Nombre	Descripción	Tipo
bienvenida	texto de la primera pantalla del mv	text
tipo	tipo patrimonio tangible o intangible	char(1)
nombre	nombre de la clasificación del patrimonio	char(50)
fecha	fecha de alta	datetime

Tabla G. 1 variables, nombre, función y tipo en la sección "Patrimonio Cultural"

Museos universitarios

Nombre	Descripción	Tipo
clave	identificador del museo	char(8)
nombre	nombre del museo	char(60)
bienvenida	texto de presentación	text
direccion	calle numero	char(255)
colonia	nombre de la colonia	char(128)
estado	estado	char(60)
codigopostal	numero de código	char(5)
telefono	tel con clave lada	char(20)
fax	fax con clave lada	char(20)
horario	horarios de visita a la exposición	char(128)
email	correo electrónico del museo	char(60)
mapa	imagen con ubicación	Char(8)
contacto	liga a correo electrónico	char(60)

Tabla G. 2 variables, nombre, función y tipo en la sección "Museos universitarios"

Exposiciones temporales

Nombre	Descripción	Tipo
consecutivo	no de exposición	tinyint(5)
clave	¿el museo le pone una clave?	char(8)
titulo	titulo de la exposición	char(255)
subtitulo	subtitulo de la exposición	char(255)
inaguracion	cuando inicia la exposición	datetime
clausura	cuando termina la exposición	datetime

espacios	lugar dentro del museo / campus	char(80)
recorrido virtual	si/no	char(1)
catalogo	si/no	char(1)
imagen	archivo jpg	char(8)
ficha tecnica	archivo doc	char(8)
abstract	archivo pdf	char(8),text
#visitas	numero de visitas a la exposición	int(12)

Tabla G. 3 variables, nombre, función y tipo en la sección "Exposiciones temporales"

Catálogo de la exposición

Nombre	Descripción	Tipo
numero	no de exposición	int(8)
objeto	no de inventario	int(8)
imagen	imagen a utilizara	char(20)

Tabla G. 4 variables, nombre, función y tipo en la sección "Catálogo de exposicion"

Catálogo impreso de la exposición

Nombre	Descripción	Tipo	Comentarios
título	nombre de la exposición	char(255)	
subtítulo	detalle de la exposición	char(255)	
fecha	año de inicio y año de fin	char(15)	
museo	nombre del museo	char(255)	
inauguracion	día mes año de inicio	datetime	fecha
clausura	día mes año de clausura	datetime	
edicion	idioma	char(50)	datos editoriales
paginas	numero de paginas	char(10)	
isbn	numero de isbn	int(10)	
editorial	nombre de la editorial	char(50)	
año	año de edicion	date	
autor a	nombre completo autor	char(255)	autores x orden alfabético
autor b	nombre completo autor	char(255)	
autor c	nombre completo autor	char(255)	
imagen	imagen representativa	char(255)	
abstract	archivo con el resumen	char(255)	información general

Tabla G. 5 variables, nombre, función y tipo en la sección "Catálogo impreso de exposicion"

Objetos

Grupo	Nombre	Descripción	Tipo
identificacion	inventario	numero de identificación del objeto	int(8,0)
	rama	clasificación del objeto	char(30)
	categoria	describe el tipo de objeto	char(30)
	subcategoria	describe el subtipo de objeto	char(30)
autor	apellido mat	apellido del autor	char(20)
	apellido pat	apellido del autor	char(20)
	nombre	nombre del autor	char(20)
	sobrenombre	sobrenombre del autor	char(20)
	nacionalidad	país del autor	char(30)
	lugar nacimiento	ciudad nacimiento del autor	char(30)
	fecha nacimiento	fecha de nacimiento del autor	datetime
	lugar muerte	ciudad de fallecimiento del autor	char(30)
	fecha muerte	fecha de fallecimiento del autor	datetime
	biografía	# de inventario.doc	char(15)
	critica	¿?	char(15)
	objeto		
piezas grupo	ancho	medidas del objeto	float(12,3)
	profundidad	medidas del objeto	float(12,3)
	diámetro	medidas del objeto	float(12,3)
	objeto		
procedencia	pueblo	pueblo de procedencia	char(30)
	estado	estado de procedencia	char(30)
	zona	zona de procedencia	char(30)
	país	país de procedencia	char(30)
	colección	colección de procedencia	char(50)
	objeto	descripción del objeto	char(50)
	titulo	como se titula el objeto	Char(255)
	film	video del objeto	char(50)
	imagen	nombre del archivo tipo imagen	int(8,0)
	cultura	descripción del objeto	char(20)
	periodo	clasificación del objeto	char(20)
	año	Año de creación	char(15)
	procedencia	¿?	char(30)
	objeto		
descripción física	material	nombre del material	text
	técnica	nombre de la técnica	text
	alto	alto del objeto	float(12,3)
	ancho	ancho del objeto	float(12,3)
	profundidad	profundidad del objeto	float(12,3)
	diametro	diámetro del objeto	float(12,3)
	descripción	archivo pdf	int(8,0)
	contexto	archivo pdf	int(8,0)

	bibliografía	archivo pdf	int(8,0)
	glosario	Archivo pdf	int(8,0)
	tema 1	oración del tema clave	char(30)
	tema 2	oración del tema clave	char(30)
	tema 3	oración del tema clave	char(30)
	tema 4	oración del tema clave	char(30)
	tema 5	oración del tema clave	char(30)
	genero	tipo de genero	char(30)
	expresiones	tipo de expresion	char(30)
	palabra 1	palabra clave	char(20)
	palabra 2	palabra clave	char(20)
	palabra 3	palabra clave	char(20)

Tabla G. 6 variables, nombre, función y tipo en la sección "Objetos"

Libro de visitas

Nombre	Descripción	Tipo
usuario	numero de usuario	int(8)
tipo	usuario normal o curador u otro tipo	tinyint(1)
fecha	fecha y hora del comentario	datetime
comentario	comentario del usuario	text

Tabla G. 7 variables, nombre, función y tipo en la sección "Libro de Visitas"

Museos imaginados

- Igual que museo
- Igual que exposiciones
- Igual que catálogo
- Igual que libro de visitas
- Mas algunos atributos específicos como:

Nombre	Descripción	Tipo
usuario	numero del usuario	int(8)
tipo	tipo de usuario	tinyint(1)
fecha	fecha de creación del museo	datetime

Tabla G. 8 variables, nombre, función y tipo en la sección "Museos imaginados"

Haz tu museo

- Igual que museo
- Igual que exposiciones
- Igual que libro de visitas
- BUSQUEDA GENERAL DE OBRAS

Novedades (requieren almacenar como historico)

- Anuncios de otros eventos

- Noticias sobre museos
- Actividades

Servicios

- Blog
- Chat
- comentarios

Usuarios

Nombre	Tipo
usuario	int(8)
contraseña	char(8)
alias	char(15)
confirmacion	
nombre	char(30)
Apellido	char(20)
correoelectronico	char(20)
Edad	int(3)
escolaridad	char(60)
sexo	tinyint(1)
ocupación	char(60)
ciudad	char(60)
pais	char(60)
hint1	char(60)
hint2	char(60)
Aviso nuevos	Char(1)

Tabla G. 9 variables, nombre, función y tipo en la sección "Usuarios"

Favoritos usuario

Nombre	Descripción	Tipo
usuario	numero de usuario	int(8)
usuario_favorito	numero usuario favorito	int(8)
aviso	aviso de nueva creación	char(3)

Tabla G. 10 variables, nombre, función y tipo en la sección "Favoritos de usuario"

"

Favoritos exposiciones

Nombre	Descripción	Tipo
usuario	numero de usuario	int(8)
usuario_favorito	numero usuario favorito	int(8)

aviso	aviso de nueva creación	char(3)
-------	-------------------------	---------

Tabla G. 10 variables, nombre, función y tipo en la sección "Favoritos exposiciones"

Bitácora

Nombre	Descripción	Tipo
fecha	facha y hora de la operación	datetime
usuario	clave del usuario	int(8)
éxito de la operación	se llevo a cabo la operación o hubo algún error	char(4)
modulo	que modulo acceso el usuario	char(10)
operacion	numero de la operación	int(20)

Tabla G. 11 variables, nombre, función y tipo en la sección "Bitácora de trabajo"

Anexo H: Programas unidad de comunicación y de ambiente virtual

Sistema Oráculo: Unidad de Comunicación (FTP)

(Visual-Básico 6.0 MicroSoft corp.)

Plataforma: PC compatible

Requerimientos técnicos: Windows 98 o superior, 512MB en RAM, 20MB disponibles en disco, Mouse y Teclado, Resolución de Monitor 800x600 o superior.

Compilador: VisualBasic Ver. 6.0

Tamaño aprox. 5MB

Descripción:

Unidad diseñada para realizar la comunicación necesaria con el servidor, para el manejo y descarga de imágenes e información tanto para el sistema Dionisios como para el sistema Apolo.

Realiza las siguientes operaciones

- a) Abre la sesión del usuario con el servidor
- b) Descarga la información relacionada ya sea con la selección de obra para la construcción de la exposición (para el sistema Dionisios)

Por razones de seguridad, las imágenes de la base de datos para la construcción de la exposición son de baja resolución.

- c) Descarga la información sobre el ambiente virtual(para el sistema Apolo)
- d) Descarga las fichas técnicas de la obra (para el sistema Apolo)
- e) Descarga al sistema (Apolo o Dionisios)
- f) Es responsable del "checksum" de la información así como de su adecuada distribución en las bibliotecas de los dos sistemas.
- g) Sube al servidor el script del museo (ver descripción del sistema Dionisios)
- h) Al termino de la sesión de cualquiera de los sistemas, borra la información en la máquina del usuario, de tal manera que no sea posible el uso indiscriminado de las imágenes del patrimonio ni de los sistemas de visor o edición de museos. De hecho el sistema Oráculo será el único residente permanente en la máquina del usuario.

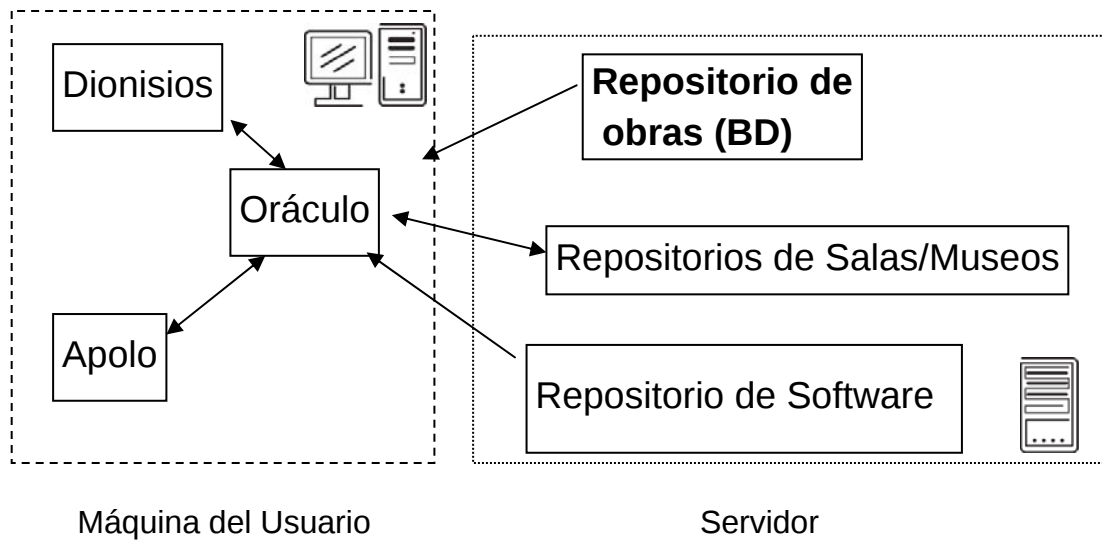


Figura H. 1 Modo de Interacción entre la máquina del usuario y el servidor del Museo Virtual
En le panel derecho se muestra la interacción en la máquina local y en el izquierdo la que ocurre en el servidor

La interfaz gráfica del 'Oráculo' muestra una ventana titulada 'OACULO' con el subtítulo 'Subir Museo'. Sección 'Campos Obligatorios' incluye campos para 'Nombre de Usuario' (Usuario 01), 'Password' (oculto), 'Título del museo:' y 'Signos y Metáforas'. Sección 'Comentario' incluye un campo de texto con el texto 'Réplica de una exposición real expuesta en Montreal Canadá por el equipo del Muca campus'. A la derecha, se muestra una imagen de una exposición con el texto 'Selecciona la Imagen representativa del Museo'. En la parte inferior, hay botones para 'Cancelar', 'Borrar' y 'Subir', así como una opción para 'Seleccionar ésta imagen'.

Figura H.2, Interfaz Gráfica del "Oráculo"

Código fuente:

Oráculo

```
; CCADET-UNAM  
; Copyright (c) 2007  
; Created: 31.08.2007 7:19  
; Version: 0.9
```

```
Dim soft, pa_donde, donde, sesion, servidor, strURL, usuario, pass, nmuseo, comenta_museo, archivo_museo, id_usuario,  
ima_rep As String
```

```
Dim imag, texto As String  
Private Sub Form_Load()  
Dim prefijo, quien As String  
Dim strArray() As String  
Dim ftp As Object, rc As Long, v As Variant, v2 As Variant  
Dim ctr As Long  
Dim strPath As String
```

```
'Datos base=====
```

```
ChDir "C:\MVPU\"
```

```
donde = "C:\MVPU\" 'direccion del museo en usuario  
servidor = "132.248.36.186" ' direccion del servidor  
imag = "/objeto/"  
compleme = "/objeto/complementos/"  
soft = "/public_html/oraculo/software/"  
texto = "/plantillas/oraculo/fichas/"  
usuario = "mvolimpico"  
Paswo = "0l!mpus07"
```

```
'Command=====
```

```
If Command = "" Then  
MsgBox "No puedo corre solo", vbInformation  
End  
Else  
quien = GetFilename(Command, True)  
Text5.Text = quien  
destino = "C:\MVPU\archivo.1mv"  
FileCopy Command, destino  
Debug.Print destino  
Kill Command  
'
```

```
=====
```

Form1.Show

RTB1.LoadFile destino, 1

busca

busca2

busca3

baja_o

etiqueta.Caption = "Iniciando Apolo"

Form1.Hide

comentario.Show

ChDir "C:\MVPU\"

abre = Shell("C:\MVPU\apolo.exe")

End If

End Sub

Public Sub busca()

donde = 1

prefijo = "08-"

sufijo = "_s_"

For i = 1 To Len(RTB1)

palabra = InStr(donde, RTB1.Text, prefijo, vbTextCompare)

If palabra - 1 < 1 Then

Exit Sub

End If

RTB1.SelStart = palabra - 1

RTB1.UpTo Chr(10), False

RTB1.Span Chr(44), True, True

es = RTB1.SelText

checa = InStr(1, es, sufijo, vbTextCompare)

If checa <> 0 Then

Text2.Text = Text2.Text & RTB1.SelText & ".jpg,"

Text3.Text = Text3.Text & RTB1.SelText & ".txt,"

Else

Text1.Text = Text1.Text & es & ".jpg,"

Text3.Text = Text3.Text & es & ".txt,"

End If

donde = palabra + 1

Next i

End Sub

Sub busca2()

donde = 1

prefijo = "00-"

sufijo = "_s_"

For i = 1 To Len(RTB1)

```

    palabra = InStr(donde, RTB1.Text, prefijo, vbTextCompare)
    If palabra - 1 < 1 Then
        Exit Sub
    End If
    RTB1.SelStart = palabra - 1
    RTB1.UpTo Chr(10), False
    RTB1.Span Chr(44), True, True
    es = RTB1.SelText
    checa = InStr(1, es, sufijo, vbTextCompare)
    If checa <> 0 Then
        Text2.Text = Text2.Text & RTB1.SelText & ".jpg,"
        Text3.Text = Text3.Text & RTB1.SelText & ".txt,"
    Else
        Text1.Text = Text1.Text & es & ".jpg,"
        Text3.Text = Text3.Text & es & ".txt,"
    End If
    donde = palabra + 1
Next i
End Sub
Sub busca3() 'complementos
    donde = 1
    prefijo = "c-"
    For i = 1 To Len(RTB1)
        palabra = InStr(donde, RTB1.Text, prefijo, vbTextCompare)
        If palabra - 1 < 1 Then
            Exit Sub
        End If
        RTB1.SelStart = palabra - 1
        RTB1.UpTo Chr(10), False
        RTB1.Span Chr(44), True, True
        es = RTB1.SelText
        Text5.Text = Text5.Text & es & ".jpg,"
    'End If
    donde = palabra + 1
Next i
End Sub
Sub baja_o()
    donde = "C:\MVP\\" 'direccion del museo en usuario
    servidor = "132.248.36.186" ' direccion del servidor
    imag = "/objeto/"
    soft = "/public_html/oraculo/software/"
    texto = "/plantillas/oraculo/fichas"
    usuario = "mvolimpico"
    Paswo = "0l!mpus07"

```

```

Dim que1, que2, que3, que4
Dim n
Set ftp = CreateObject("basp21.FTP") 'crear instancia del ftp
ftp.OpenLog "C:\log.txt" 'crea log
rc = ftp.Connect(servidor, usuario, Paswo)
If rc = 0 Then
    v = ftp.GetDir(imag, 2) 'buscar directorio
    If IsArray(v) Then
        For Each v2 In v
            Debug.Print v2
        Next
    Else
        MsgBox "No puedo conectarme al servidor!" 'fracazo

        End If
    End If
'-----
On Error Resume Next
'-----
n = 0

For i = 1 To Len(Text1.Text)
    que1 = Split(Text1.Text, ",")(n) 'jpg 2D
    If que1 = "" Then
        Exit For
    End If
    etiqueta.Caption = "Bajando Obra" 'que1
    rc = ftp.GetFile(imag & que1, "C:\MVPU\1\" & que1, 1)
    n = n + 1

Next i

n = 0
For i = 1 To Len(Text2.Text)
    que2 = Split(Text2.Text, ",")(n) 'jpg 3D
    If que2 = "" Then
        Exit For
    End If
    etiqueta.Caption = "Bajando Obra 3D" 'que2
    rc = ftp.GetFile(imag & que2, "C:\MVPU\2\" & que2, 1)

    n = n + 1
    ""Bajando Obra 3D"
Next i

```

```

For i = 1 To Len(Text4.Text)
    que4 = Split(Text4.Text, ",")(n) 'jpg 2D
If que4 = "" Then
    Exit For
End If
etiqueta.Caption = "Bajando Complementos" 'que1
rc = ftp.GetFiles("/objeto/complemento/" & que4, "C:\MVPU\1\" & que4, 1)
n = n + 1
Next i
n = 0
For i = 1 To Len(Text3.Text)
    que3 = Split(Text3.Text, ",")(n) 'textos
If que3 = "" Then
    Exit For
End If
etiqueta.Caption = "Bajando Información de las obras"
rc = ftp.GetFiles("/plantillas/oraculo/fichas/" & que3, "C:\MVPU\cedulas\" & que3, 1)
n = n + 1
rc = ftp.GetFiles("/plantillas/oraculo/fichas/vacia.txt", "C:\MVPU\cedulas\vacia.txt", 1)

Next i

n = 0
For i = 1 To Len(Text5.Text)
    que5 = Split(Text5.Text, ",")(n) 'complementos
If que5 = "" Then
    Exit For
End If
etiqueta.Caption = "Bajando Información de las obras"
rc = ftp.GetFiles("/objeto/complemento/" & que5, "C:\MVPU\1\" & que5, 1)
n = n + 1

Next i

'ftp.Close

End Sub
Sub apolo()

donde = "C:\MVPU\" 'direccion del museo en usuario

```

```

servidor = "132.248.36.186" ' direccion del servidor
soft = "/public_html/oraculo/software/"
usuario = "mvolimpico"
Paswo = "0l!mpus07"

Dim n
Set ftp = CreateObject("basp21.FTP") 'crear instancia del ftp
ftp.OpenLog "C:\logapolo.txt" 'crea log
rc = ftp.Connect(servidor, usuario, Paswo)
If rc = 0 Then
    v = ftp.GetDir(imag, 2) 'buscar directorio
    If IsArray(v) Then
        For Each v2 In v
            Debug.Print v2
        Next
    Else
        MsgBox "No puedo conectarme al servidor!" 'fracazo
    End If
End If
etiqueta.Caption = "Invocando a Apolo"
rc = ftp.GetFile(soft & "apolo.exe", "C:\MVPU\apolo.exe", 1) ' apolo
If rc <> 0 Then
    etiqueta.Caption = "Iniciando Apolo!"
    abre = Shell("C:\MVPU\apolo.exe") '& quien, 0)

Else
    MsgBox "No puedo conectarme al servidor!" 'fracazo

End If
ftp.Close
End
End Sub
Sub apolo2()
Dim abre
etiqueta.Caption = "Iniciando Apolo"
abre = Shell("C:\MVPU\apolo.exe")
AppActivate abre

Form1.Hide
f_subir.Show

End
End Sub

Function GetFilename(path As String, NoExt As Boolean) As String

```

```

        Slit = Split(StrReverse(path), "\")
    If Not NoExt Then
        Spit = Split(Slit(0), ".", 2)
        GetFilename = StrReverse(Spit(1))
    Else
        GetFilename = StrReverse(Slit(0))
    End If

End Function

Dim dequien

Private Sub Command1_Click()
    Text4.Text = ""
End Sub

Private Sub Command2_Click()
    Dim cuando, dos, prefijo, tres, cuatro, cinco, quien As String
    Dim donde
    Dim n
    Dim texto
    Dim aquien, archi

    If Text1.Text = "" Or Text2.Text = "" Or Text4.Text = "" Then
        MsgBox "¡Te faltan campos por llenar!", , "Oráculo"

        Exit Sub
    Else
        End If
    'fecha

    donde = 1
    cuando = Now
    For i = 1 To Len(cuando)
        dos = Split(cuando, "/")(0) ' día
        dos1 = Split(cuando, "/")(1) 'mes
        dos2 = Split(cuando, "/")(2) 'año'
        tres = Split(dos2, ":")(0) 'horas
        tres1 = Split(dos2, ":")(1) 'minutos
        cinco = tres & tres1
        cuatro = dos & dos1 & cinco
        seis = Split(cuatro, " ")(0)
        Seis1 = Split(cuatro, " ")(1)
        If dos - 1 < 1 Then
            Exit Sub
        End If
        donde = donde + 1

```

```

siete = seis & Seis1
Next i
'=====
    tempo = "C:\MVPU\" & Text1.Text & "_" & siete & "_" & "comentario" & ".txt"
    texto = Text1.Text & "|" & Text4.Text & "|" & Form1.Text5.Text & "|"
    Open tempo For Output As #1
    Write #1, texto
    Close #1

Set ftp = CreateObject("basp21.FTP") 'crear instancia
ftp.OpenLog "C:\MVPU\log.tio" 'crea log
rc = ftp.Connect("132.248.36.186", "mvolimpico", "0l!mpus07") 'abrir sección
If rc = 0 Then
    v = ftp.GetDir(strPath, 2) 'buscar directorio
    If IsArray(v) Then
        For Each v2 In v
            Debug.Print v2
        Next
        rc1 = ftp.PutFile(tempo, "/plantillas/oraculo/coment/")
    End If
    If rc1 > 0 Then
        MsgBox "¡Tu Comentario fué correctamente agregado!", , Oráculo
        Kill "C:\MVPU\*.1mv"
        Kill "C:\MVPU\*.txt"
        End If
        matanga
    End If

    If rc1 = 0 Then
        MsgBox "¡Ocurrió un error, comunícate con nosotros!", , Oráculo

    End If

    ftp.Close
    End
End Sub
Private Sub Command3_Click()
On Error Resume Next
Kill "c:\MVPU\*.mv"
matanga
End
End Sub

Public Sub matanga()

```

```

On Error Resume Next
    Kill "c:\MVPU\1\*.jpg"
    Kill "c:\MVPU\2\*.jpg"
' Kill "c:\MVPU\cedulas\*.*)"
' Kill "c:\MVPU\comm.tmp"

End Sub
Private Sub Form_Load()
Dim archi, dequien

If Command$ = "" Then
    MsgBox "¡No trabajo solo!", , "Oráculo"
End
Else
dequien = GetFilename(Command, True)
' dequien = "Caviedes_050320080915_.1mv"
End If
End Sub
Function GetFilename(path As String, NoExt As Boolean) As String
    Slit = Split(StrReverse(path), "\")
    If Not NoExt Then
        Spit = Split(Slit(0), ".", 2)
        GetFilename = StrReverse(Spit(1))
    Else
        GetFilename = StrReverse(Slit(0))
    End If
End Function
End Function

```

Sistema Apolo; Visor de Museos

Plataforma: PC compatible

Requerimientos técnicos: Windows XP, 512MB en RAM, 20MB disponibles en disco, Mouse y Teclado, Resolución de Monitor 800x600 o superior.

Compilador: Biltz3D (<http://www.blitzbasic.com/Products/blitz3d.php>)

Tamaño aprox. 1.7MB

Descripción

Esta unidad, responsable de presentar al usuario el entorno virtual del museo, ya sea en su versión oficial, los museos universitarios y los imaginados por los usuarios, está capacitado para manejar tres formatos gráficos; imagen 2D, fotografía estereoscópica en 360° y audio en formato mp3, así mismo despliega el formato de modelo tridimensional 3DS, en el cual se encuentra diseñado y construido el “edificio” del museo virtual. Se comunica al servidor a través de “Oráculo” para descargar: el script que permite la construcción del ambiente virtual, las imágenes que componen la exposición en formato de alta resolución y la información de la base de datos que compone la ficha técnica de las obras.

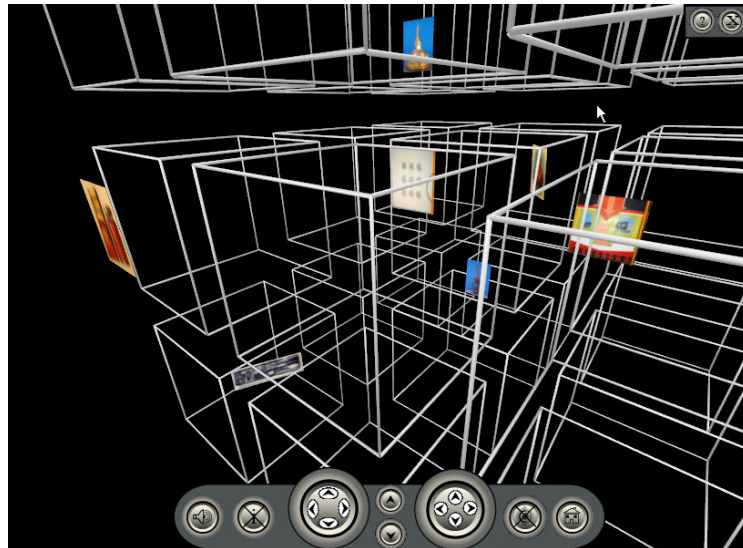


Figura H.3. Interfase del visor 3D (Apolo) en la exposición demo “Signos y Metáfoas”

Código Fuente:

```
; CCADET-UNAM
; Copyright (c) 2007
;
; Created: 31.08.2007 7:19
; Version: 0.9
; Visor de Museos Virtuales, Apolo
```

```
.Modo_grafico
;===== Establecimiento del modo gráfico =====
Graphics3D 800,600,32,2
SetBuffer BackBuffer()
;=====

.fuentes
;===== Fuentes =====
Global f_Arial=LoadFont("Arial",14,False,False,False)
Global f_ArialB=LoadFont("Arial",14,True,False,False)
Global f_ArialC=LoadFont("Arial",13,False,False,False)
;=====

;Carga elementos para los menús=====
.elementos2D
Global z_ayuda=LoadImage("interfaz/z_help.bmp")
Global z_info=LoadImage("interfaz/z_info.bmp")
Global salir0=LoadImage("interfaz/salir0.bmp")
Global salir1=LoadImage("interfaz/salir1.bmp")
Global salir_m0=LoadImage("interfaz/menu_salir0.bmp")
Global salir_m1=LoadImage("interfaz/menu_salir1.bmp")
Global salir_ayuda0=LoadImage("interfaz/menu_salir0.bmp")
Global salir_ayuda1=LoadImage("interfaz/menu_salir1.bmp")
Global ayuda0=LoadImage("interfaz/ayuda0.bmp")
Global ayuda1=LoadImage("interfaz/ayuda1.bmp")
Global v_ayuda=LoadImage("interfaz/ayuda.jpg")
Global info0=LoadImage("interfaz/info0.bmp")
Global info1=LoadImage("interfaz/info1.bmp")
Global info_off0=LoadImage("interfaz/info_off0.bmp")
Global info_off1=LoadImage("interfaz/info_off1.bmp")
Global entrada0=LoadImage("interfaz/entrada0.bmp")
Global entrada1=LoadImage("interfaz/entrada1.bmp")
Global radar0=LoadImage("interfaz/radar0.bmp")
Global radar1=LoadImage("interfaz/radar1.bmp")
Global radar_off0=LoadImage("interfaz/radar_off0.bmp")
Global radar_off1=LoadImage("interfaz/radar_off1.bmp")
Global audio0=LoadImage("interfaz/sonido0.bmp")
```

```

Global audio1=LoadImage("interfaz/sonido1.bmp")
Global audio_off0=LoadImage("interfaz/sonido_off0.bmp")
Global audio_off1=LoadImage("interfaz/sonido_off1.bmp")
Global subir0=LoadImage("interfaz/subir0.bmp")
Global subir1=LoadImage("interfaz/subir1.bmp")
Global bajar0=LoadImage("interfaz/bajar0.bmp")
Global bajar1=LoadImage("interfaz/bajar1.bmp")
Global arriba=LoadImage("interfaz/arriba.bmp")
Global abajo=LoadImage("interfaz/abajo.bmp")
Global izquierda=LoadImage("interfaz/izq.bmp")
Global derecha=LoadImage("interfaz/der.bmp")
Global r_arriba=LoadImage("interfaz/rotarriba.bmp")
Global r_abajo=LoadImage("interfaz/rotabajo.bmp")
Global r_izq=LoadImage("interfaz/rotaizq.bmp")
Global r_der=LoadImage("interfaz/rotader.bmp")
Global pleca=LoadImage("interfaz/pleca.bmp")
Global e_adelante=LoadImage("interfaz/e_adelante.jpg")
Global e_atras=LoadImage("interfaz/e_atras.jpg")
Global e_izq=LoadImage("interfaz/e_izq.jpg")
Global e_der=LoadImage("interfaz/e_der.jpg")
Global e_rarriba=LoadImage("interfaz/e_rarriba.jpg")
Global e_rabajo=LoadImage("interfaz/e_rabajo.jpg")
Global e_rizq=LoadImage("interfaz/e_rizq.jpg")
Global e_rder=LoadImage("interfaz/e_rder.jpg")
Global e_radar=LoadImage("interfaz/e_radar.jpg")
Global e_audio=LoadImage("interfaz/e_audio.jpg")
Global e_sube=LoadImage("interfaz/e_sube.jpg")
Global e_baja=LoadImage("interfaz/e_baja.jpg")
Global e_info=LoadImage("interfaz/e_info.jpg")
Global e_inicio=LoadImage("interfaz/e_inicio.jpg")
Global e_ayuda=LoadImage("interfaz/e_ayuda.jpg")
Global e_salir=LoadImage("interfaz/e_salir.jpg")
Global cursor=LoadImage("interfaz/cursor.bmp")
Global pixel=LoadImage("interfaz/pixel.bmp")
;=====

.Declara_Vars
;===== Declaracion de variables =====
Dim linea$(1000)
i=0
j=1
k=1
ren_ini=0;Posición del 'Helper Inicio' dentro del vector que contiene los renglones sin comentar del archivo AMV
Global NUM_ARQ
Global NUM_OBRAS
Global num_anime=0;Contador para los FilmStrips
Global NUM_AUDIOS=0;Contador para los 'Helpers de Audio'

```

```

Global muestra_radar=1;Variable que indica si se muestra 'El Radar'
Global num_radar=1;Indica la camara activa para el radar.
Global camara_radar$="Camara Frontal";
Global acumuladoX,acumuladoY,acumuladoZ,divisor_acumulado,promX,promY,promZ,hx,hy,hz;Ajuste del radar

```

```

Global muestra_info=1;Variable que indica si se muestra la información de 'La cédula'
Global muestra_audio=1;Variable que indica si se escucha el audio 3D de cada obra
Global muestra_ayuda=0;Variable que indica si se muestra la ayuda del programa
Global muestra_interfaz=1; Se muestra o no la interfaz o 'skin'

```

```

Global rgb=0
Global wire=0

```

```

Global museo$
Global autor$
Global fecha$

```

```

Dim anime(20);Almacena la posicion de la animacion dentro del archivo txt
Dim cedula$(20,20)

```

```

;=====

```

```

.Abre_Archivo
;===== Apertura del archivo =====
.Lectura_Archivo
;archivo$=CommandLine();Parametro por linea de comandos
archivo$="archivo.1mv";Parametro por linea de comandos
lee_archivo=ReadFile(archivo$)
;=====

```

```

.Archivo_Vector
;===== Descomposición del archivo en cadenas de texto =====
While Not Eof(lee_archivo)
    linea$(i)=ReadLine$(lee_archivo)
    If (Mid$(linea$(i),1,6)="Inicio" Or Mid$(linea$(i),1,6)="inicio") ren_ini=i
    If (Mid$(linea$(i),1,5)="Audio" Or Mid$(linea$(i),1,5)="audio") NUM_AUDIOS=NUM_AUDIOS+1
    If (Mid$(linea$(i),1,1) <> ";") And (Mid$(linea$(i),1,1) <> "") And (Mid$(linea$(i),1,1) <> " ") Then
        i=i+1
    EndIf
Wend
;=====

```

```

.CabeceraArchivo
;===== Escritura del nombre del museo y autor en la barra de la ventana =====
While Mid$(linea$(0),j,1) <> ","
    j=j+1
Wend

```

```

While Mid$(linea$(0),j+k,1) <> ","
    k=k+1
Wend

museo$=Mid$(linea$(0),1,j-1)
autor$=Mid$(linea$(0),j+1,k-1)
fecha$=Mid$(linea$(0),j+k+1,Len(linea$(0)))

;=====

.Declara_Types
;===== Declaración de Tipos =====
Type ayudante
    Field audio$, radio#
    Field nombre
    Field Px#,Py#,Pz#
    Field Rx#,Ry#,Rz#
End Type
Type arquitectura
    Field malla
    Field nombre$
    Field Px#,Py#,Pz#
    Field Rx#,Ry#,Rz#
    Field Sx#,Sy#,Sz#
    Field textura$
End Type
Type patrimonio
    Field malla
    Field nombre$, textura$, cedula$
    Field Px#, Py#, Pz#
    Field Rx#, Ry#, Rz#
    Field Sx#, Sy#, Sz#
End Type
;=====

.IniVarsTypes
;===== Inicia Variables y Tipos =====
NUM_ARQ= ren_ini-1
NUM_OBRAS= i-ren_ini-NUM_AUDIOS-1
Dim els_arq.arquitectura(NUM_ARQ)
Dim obra.patrimonio(NUM_OBRAS)
Dim helper_audio.ayudante(NUM_AUDIOS)

.Arquitectura
;===== Generación de la arquitectura =====
;De los valores anteriores al punto de Inicio, sabemos que todas las lineas,

```

;representan elementos de la arquitectura, los cuales pueden ser de dos tipos:
;Primitivas Geometricas y Archivos 3DS
;Con el primer ciclo While, separaremos los primitivos de los 3DS

```
.TRI_D_S
i=1: j=0
While i < NUM_ARQ+1
  If Mid$(linea$(i),Len(linea$(i))-2,3)="3DS" Then
    NUM_3DS=NUM_3DS+1;Numero de archivos 3DS
    LoadMesh(linea$(i))
  Else
    j=j+1
    linea$(j)=linea$(i) ; Reasignaciòn de los elementos en el vector de Arquitectura (solo primitivos)
  End If
  i=i+1
Wend

NUM_ARQ= NUM_ARQ-NUM_3DS
```

```
If NUM_ARQ > 0 Then
i=1
While i <= NUM_ARQ
  divisor_acumulado=divisor_acumulado+1

  j=1: k=0
  ;Genera Objeto
  els_arq(i)= New arquitectura
  k=parte_renglon(i,j,1)
  ;Nombre Objeto
  els_arq(i)\nombre$= Mid$(linea$(i),1,j+k-1)

  ;Obtiene Px
  j=j+k+1: k=parte_renglon(i,j,1): els_arq(i)\Px#= Mid$(linea$(i),j,k)
  acumuladoX=acumuladoX+els_arq(i)\Px#
  ;Obtiene Py
  j=j+k+1: k=parte_renglon(i,j,1): els_arq(i)\Py#= Mid$(linea$(i),j,k)
  acumuladoY=acumuladoY+els_arq(i)\Py#
  ;Obtiene Pz
  j=j+k+1: k=parte_renglon(i,j,1): els_arq(i)\Pz#= Mid$(linea$(i),j,k)
  acumuladoZ=acumuladoZ+els_arq(i)\Pz#
  ;Obtiene Rx
  j=j+k+1: k=parte_renglon(i,j,1): els_arq(i)\Rx#= Mid$(linea$(i),j,k)
  ;Obtiene Ry
  j=j+k+1: k=parte_renglon(i,j,1): els_arq(i)\Ry#= Mid$(linea$(i),j,k)
  ;Obtiene Rz
```

```

j=j+k+1: k=parte_renglon(i,j,1): els_arq(i)\Rz#= Mid$(linea$(i),j,k)
;Obtiene Sx
j=j+k+1: k=parte_renglon(i,j,1): els_arq(i)\Sx#= Mid$(linea$(i),j,k)
;Obtiene Sy
j=j+k+1: k=parte_renglon(i,j,1): els_arq(i)\Sy#= Mid$(linea$(i),j,k)
;Obtiene Sz
j=j+k+1: k=parte_renglon(i,j,1): els_arq(i)\Sz#= Mid$(linea$(i),j,k)
;Obtiene Textura
j=j+k+1: k=ultima_parte_renglon(i,j,1): els_arq(i)\textura$= Mid$(linea$(i),j,k)

;Genera malla
els_arq(i)\malla= CreateCube()
EntityType els_arq(i)\malla,2

;Coloca al objeto en la posición (px, py, pz)
PositionEntity els_arq(i)\malla,els_arq(i)\Px#,els_arq(i)\Py#,els_arq(i)\Pz#

;Rota al objeto respecto a Ox, Oy, Oz
TurnEntity els_arq(i)\malla,els_arq(i)\Rx#,els_arq(i)\Ry#,els_arq(i)\Rz#

;Escala al objeto
ScaleEntity els_arq(i)\malla,els_arq(i)\Sx#,els_arq(i)\Sy#,els_arq(i)\Sz#

;Texturiza al objeto
tex=LoadTexture("texturas/"+els_arq(i)\textura$)
;Determina el Tiling
If els_arq(i)\Sz# > els_arq(i)\Sy# And els_arq(i)\Sx# > els_arq(i)\Sy# Then
    ScaleTexture tex, 1 / els_arq(i)\Sx#, 1 / els_arq(i)\Sz#
Else
    If els_arq(i)\Sy# > els_arq(i)\Sz# And els_arq(i)\Sx# > els_arq(i)\Sz# Then
        ScaleTexture tex, 1 / els_arq(i)\Sx#, 1 / els_arq(i)\Sy#
    Else
        If els_arq(i)\Sy# > els_arq(i)\Sx# And els_arq(i)\Sz# > els_arq(i)\Sx# Then
            ScaleTexture tex, 1 / els_arq(i)\Sz#, 1 / els_arq(i)\Sy#
        EndIf
    EndIf
EndIf
EntityTypeTexture els_arq(i)\malla,tex

i=i+1
Wend
Else
    NUM_ARQ=-1 ;Si no hay primitivos, debe omitir el calculo de la posicion del radar.
End If

```

```

.PuntoEntrada
;===== Generaciòn del punto de entrada =====
Global helper_inicio.ayudante= New ayudante
;divisor_acumulado=divisor_acumulado+1
;Obtiene Px
j=8: k=parte_renglon(ren_ini,j,1): helper_inicio\Px#= Mid$(linea$(ren_ini),j,k)
;acumuladoX=acumuladoX+helper_inicio\Px#
;Obtiene Py
j=j+k+1: k=parte_renglon(ren_ini,j,1): helper_inicio\Py#= Mid$(linea$(ren_ini),j,k)
;acumuladoY=acumuladoY+helper_inicio\Py#
;Obtiene Pz
j=j+k+1: k=parte_renglon(ren_ini,j,1): helper_inicio\Pz#= Mid$(linea$(ren_ini),j,k)
;acumuladoZ=acumuladoZ+helper_inicio\Pz#
;Obtiene Rx
j=j+k+1: k=parte_renglon(ren_ini,j,1): helper_inicio\Rx#= Mid$(linea$(ren_ini),j,k)
;Obtiene Ry
j=j+k+1: k=parte_renglon(ren_ini,j,1): helper_inicio\Ry#= Mid$(linea$(ren_ini),j,k)
;Obtiene Rz
j=j+k+1: k=ultima_parte_renglon(cont,j,1): helper_inicio\Rz#= Mid$(linea$(ren_ini),j,k)

.HelpersAudio
;===== HELPERS DE AUDIOS =====
For i=1 To NUM_AUDIOS
    helper_audio(i)= New ayudante
    j=8: k=0
    ;Obtiene Px
    k=parte_renglon(ren_ini+i,j,1): helper_audio(i)\Px#= Mid$(linea$(ren_ini+i),j,k)
    ;Obtiene Py
    j=j+k+1: k=parte_renglon(ren_ini+i,j,1): helper_audio(i)\Py#= Mid$(linea$(ren_ini+i),j,k)
    ;Obtiene Pz
    j=j+k+1: k=parte_renglon(ren_ini+i,j,1): helper_audio(i)\Pz#= Mid$(linea$(ren_ini+i),j,k)
    ;Obtiene Radio
    j=j+k+1: k=parte_renglon(ren_ini+i,j,1): helper_audio(i)\radio#= Mid$(linea$(ren_ini+i),j,k)
    ;Obtiene Audio
    j=j+k+1: k=ultima_parte_renglon(ren_ini+i,j,1): helper_audio(i)\audio$= Mid$(linea$(ren_ini+i),j,k)

    helper_audio(i)\nombre= CreatePivot()
    PositionEntity helper_audio(i)\nombre,helper_audio(i)\Px#,helper_audio(i)\Py#,helper_audio(i)\Pz#
Next

.ObrasMV
;===== Generaciòn de las obras del MV =====
i=ren_ini+NUM_AUDIOS+1: l=1

While i < NUM_OBRAS+ren_ini+NUM_AUDIOS+1

```

```
divisor_acumulado=divisor_acumulado+1
```

```
;Genera objeto
```

```
obra(l)= New patrimonio
```

```
;Genera malla
```

```
obra(l)\malla= CreateCube()
```

```
EntityType obra(l)\malla,2
```

```
EntityPickMode obra(l)\malla,2
```

```
;Nombre / Textura / Cedula / (Dimensiones) del Objeto
```

```
obra(l)\nombre$= Mid$(linea$(i),1,8)
```

```
NameEntity obra(l)\malla,obra(l)\nombre$
```

```
j=1: k=0: k=parte_renglon(i,j,1)
```

```
obra(l)\textura$= Mid$(linea$(i),j,k) + ".jpg"
```

```
obra(l)\cedula$="cedulas/"+obra(l)\nombre$+".txt"
```

```
cedula$(l,0)=ReadFile(obra(l)\cedula$)
```

```
;En caso de no tener Cedula la obra que se esta importando.....
```

```
If cedula$(l,0)=0 Then
```

```
cedula$(l,0)=ReadFile("cedulas/vacia.txt")
```

```
EndIf
```

```
;Descomposicion de las cedulas en vectores
```

```
renglon=1
```

```
While Not Eof(cedula$(l,0))
```

```
cedula$(l,renglon)=ReadLine$(cedula$(l,0))
```

```
;Considera unicamente los renglones validos de la cedula
```

```
If (Mid$(cedula$(l,renglon),1,1) <> "") And (Mid$(cedula$(l,renglon),1,1) <> " ") Then
```

```
posicion=1
```

```
While Mid$(cedula$(l,renglon),posicion,1) <> ":
```

```
posicion=posicion+1
```

```
Wend
```

```
long_ren=Len(cedula$(l,renglon))
```

```
;Print Mid$(cedula$(l,renglon),posicion+1,2):WaitKey()
```

```
;Imprime unicamente los campos llenos d la cedula (Algo despues de :)
```

```
If (Mid$(cedula$(l,renglon),posicion,3) <> " ") Then
```

```
cedula$(l,renglon+1)=Trim$(Mid$(cedula$(l,renglon),posicion+1,long_ren))
```

```
If Len(cedula$(l,renglon+1))> 27 Then
```

```
cedula$(l,renglon+3)=Trim$(Mid$(cedula$(l,renglon+1),28,long_ren))
```

```
cedula$(l,renglon+2)="VACIO"
```

```

cedula$(l, renglon+1)=Trim$(Mid$(cedula$(l, renglon+1),1,27))
ren_adicional=3
Else
cedula$(l, renglon+1)=Trim$(Mid$(cedula$(l, renglon),posicion+1,long_ren))
ren_adicional=1
EndIf
cedula$(l, renglon)=Mid$(cedula$(l, renglon),1,posicion)
renglon=renglon+1+ren_adicional
EndIf
EndIf
Wend

```

```

If Mid$(linea$(i),10,1)= "S" Then
    tipo_obra=1; FilmStrip
Else
    tipo_obra=0; Imagen
EndIf

```

;Posicion / Rotacion / (Dimensiones del Objeto)

```

;Obtiene Px
j=j+k+1: k=parte_renglon(i,j,1): obra(l)\Px#= Mid$(linea$(i),j,k)
acumuladoX=acumuladoX+obra(l)\Px#
;Obtiene Py
j=j+k+1: k=parte_renglon(i,j,1): obra(l)\Py#= Mid$(linea$(i),j,k)
acumuladoY=acumuladoY+obra(l)\Py#
;Obtiene Pz
j=j+k+1: k=parte_renglon(i,j,1): obra(l)\Pz#= Mid$(linea$(i),j,k)
acumuladoZ=acumuladoZ+obra(l)\Pz#
;Obtiene Rx
j=j+k+1: k=parte_renglon(i,j,1): obra(l)\Rx#= Mid$(linea$(i),j,k)
;Obtiene Ry
j=j+k+1: k=parte_renglon(i,j,1): obra(l)\Ry#= Mid$(linea$(i),j,k)
;Obtiene Rz
j=j+k+1: k=parte_renglon(i,j,1): obra(l)\Rz#= Mid$(linea$(i),j,k)
;Obtiene Sx
j=j+k+1: k=parte_renglon(i,j,1): obra(l)\Sx#= Mid$(linea$(i),j,k)
;Obtiene Sy
j=j+k+1: k=parte_renglon(i,j,1): obra(l)\Sy#= Mid$(linea$(i),j,k)
;Obtiene Sz
j=j+k+1: k=ultima_parte_renglon(i,j,1): obra(l)\Sz#= Mid$(linea$(i),j,k)

```

```

;Coloca al objeto en la posición (px, py, pz)
PositionEntity obra(l)\malla,obra(l)\Px#,obra(l)\Py#,obra(l)\Pz#
;Rota al objeto respecto a Ox, Oy, Oz
TurnEntity obra(l)\malla,obra(l)\Rx#,obra(l)\Ry#,obra(l)\Rz#

```

```
;Escala al objeto
ScaleEntity obra(l)\malla,obra(l)\Sx#,obra(l)\Sy#,obra(l)\Sz#
```

```
;Texturiza
```

```
If tipo_obra=1 Then ;Si es FilmStrip...
```

```
num_anime=num_anime+1
```

```
anime(num_anime)=l
```

```
;Medida del FilmStrip
```

```
ancho_fotograma=Int(Mid$(obra(l)\textura$,Len(obra(l)\textura$)-10,3))
```

```
alto_fotograma=Int(Mid$(obra(l)\textura$,Len(obra(l)\textura$)-6,3))
```

```
Select NUM_ANIME
```

```
Case 1
```

```
tex_a1=LoadAnimTexture("imgs3D/"+obra(l)\textura$,1,ancho_fotograma,alto_fotograma,0,36)
```

```
If tex_a1 = 0 Then
```

```
tex_a1=LoadTexture("texturas/no_obra.jpg")
```

```
RuntimeError "ERROR TEXTURA1"
```

```
EndIf
```

```
Case 2
```

```
tex_a2=LoadAnimTexture("imgs3D/"+obra(l)\textura$,1,ancho_fotograma,alto_fotograma,0,36)
```

```
If tex_a2 = 0 Then
```

```
tex_a2=LoadTexture("texturas/no_obra.jpg")
```

```
RuntimeError "ERROR TEXTURA2"
```

```
EndIf
```

```
Case 3
```

```
tex_a3=LoadAnimTexture("imgs3D/"+obra(l)\textura$,1,ancho_fotograma,alto_fotograma,0,36)
```

```
If tex_a3 = 0 Then
```

```
tex_a3=LoadTexture("texturas/no_obra.jpg")
```

```
RuntimeError "ERROR TEXTURA3"
```

```
EndIf
```

```
Case 4
```

```
tex_a4=LoadAnimTexture("imgs3D/"+obra(l)\textura$,1,ancho_fotograma,alto_fotograma,0,36)
```

```
If tex_a4 = 0 Then
```

```
tex_a4=LoadTexture("texturas/no_obra.jpg")
```

```
RuntimeError "ERROR TEXTURA4"
```

```
EndIf
```

```
Case 5
```

```
tex_a5=LoadAnimTexture("imgs3D/"+obra(l)\textura$,1,ancho_fotograma,alto_fotograma,0,36)
```

```
If tex_a5 = 0 Then
```

```
tex_a5=LoadTexture("texturas/no_obra.jpg")
```

```
RuntimeError "ERROR TEXTURA5"
```

```
EndIf
```

```
Case 6
```

```
tex_a6=LoadAnimTexture("imgs3D/"+obra(l)\textura$,1,ancho_fotograma,alto_fotograma,0,36)
```

```
If tex_a6 = 0 Then
```

```
tex_a6=LoadTexture("texturas/no_obra.jpg")
```

```
RuntimeError "ERROR TEXTURA6"
```

```

        EndIf
    Case 7
        tex_a7=LoadAnimTexture("imgs3D/"+obra(l)\textura$,1,ancho_fotograma,alto_fotograma,0,36)
        If tex_a7 = 0 Then
            tex_a7=LoadTexture("texturas/no_obra.jpg")
            RuntimeError "ERROR TEXTURA7"
        EndIf
    Case 8
        tex_a8=LoadAnimTexture("imgs3D/"+obra(l)\textura$,1,ancho_fotograma,alto_fotograma,0,36)
        If tex_a8 = 0 Then
            tex_a8=LoadTexture("texturas/no_obra.jpg")
            RuntimeError "ERROR TEXTURA8"
        EndIf
    Case 9
        tex_a9=LoadAnimTexture("imgs3D/"+obra(l)\textura$,1,ancho_fotograma,alto_fotograma,0,36)
        If tex_a9 = 0 Then
            tex_a9=LoadTexture("texturas/no_obra.jpg")
            RuntimeError "ERROR TEXTURA9"
        EndIf
    Case 10
        tex_a10=LoadAnimTexture("imgs3D/"+obra(l)\textura$,1,ancho_fotograma,alto_fotograma,0,36)
        If tex_a10 = 0 Then
            tex_a10=LoadTexture("texturas/no_obra.jpg")
            RuntimeError "ERROR TEXTURA10"
        EndIf

End Select

Else ;Si es imagen 2D
    tex=LoadTexture("imgs2D/"+obra(l)\textura$)
    If tex = 0 Then
        tex=LoadTexture("texturas/no_obra.jpg")
    EndIf
    EntityTexture obra(l)\malla,tex
EndIf

l=l+1
i=i+1

Wend
;=====
;-----
Collisions 1,2,2,2
;-----
.camara

```

```

Global player=CreatePivot()
PositionEntity player,helper_inicio\Px#,helper_inicio\Py#,helper_inicio\Pz#
EntityRadius player,10
EntityType player,1
Global camera=CreateCamera(player)
CameraRange camera,.1,100000
TurnEntity player,helper_inicio\Ry#,helper_inicio\Rz#

.camara_sup1 ;Camara superior
Global cam_sup1=CreateCamera()
CameraViewport
cam_sup1,GraphicsWidth()*0.81,GraphicsHeight()*0.77,GraphicsWidth()*0.19,GraphicsHeight()*0.23
CameraRange cam_sup1,1,1000000
RotateEntity cam_sup1,0,0,0
;CameraProjMode cam_sup1,2
;CameraZoom cam_sup1,0.0015

.luces
light=CreateLight()
LightColor light,100,100,100
TurnEntity light,45,45,0
light=CreateLight()
LightColor light,125,125,125
TurnEntity light,45,-45,0
AmbientLight 200,200,200

;=====Esfera para la posición de la camara=====
.esfera_camara
Global esfera_camara=CreateSphere(4,player)
ScaleEntity esfera_camara,35,35,35
EntityColor esfera_camara,255,255,255;Color inicial=blanco
EntityAlpha esfera_camara,0.5

.Ajusta_Radar ;=====

promX=((mayorX()-menorX())/2)+menorX()
promY=((mayorY()-menorY())/2)+menorY()
promZ=((mayorZ()-menorZ())/2)+menorZ()

If NUM_3DS > 0 Then ;Distancia para el archivo 3DS
    hz= 1300: PositionEntity cam_sup1,0,0,-hz
    hx= 1300: hy= 1300
Else

If NUM_ARQ > 0 Then
    hy= planoXZ()/Tan(35)

```

```

        hz= planoXY()/Tan(35)
        hx= planoYZ()/Tan(35)
        PositionEntity cam_sup1,promX,promY,-hz+menorZ()
    EndIf
End If

.Carga_Audio
oyente= CreateListener (player,0.01,1,100)

For i=1 To NUM_AUDIOS
    Select helper_audio(i)\audio$
        Case "atmosfera_a.mp3"
            sonido1=Load3DSound ("audio/atmosfera_a.mp3")
            sonido=sonido1
            LoopSound sonido
        Case "atmosfera_b.mp3"
            sonido2=Load3DSound ("audio/atmosfera_b.mp3")
            sonido=sonido2
            LoopSound sonido
        Case "atmosfera_c.mp3"
            sonido3=Load3DSound ("audio/atmosfera_g.wav")
            sonido=sonido3
            LoopSound sonido
        Case "atmosfera_d.mp3"
            sonido4=Load3DSound ("audio/atmosfera_d.mp3")
            sonido=sonido4
            LoopSound sonido
        Case "atmosfera_e.mp3"
            sonido5=Load3DSound ("audio/atmosfera_e.mp3")
            sonido=sonido5
            LoopSound sonido
    End Select

    EmitSound sonido,helper_audio(i)\nombre
Next

.Loope
;===== LOOP PRINCIPAL =====
While Not KeyHit(1)
    SetBuffer BackBuffer()

    dibuja_skin()
    teclas()
    botones()
    Radar()

```

```

If muestra_radar=1 Then
    Text GraphicsWidth()*0.86,GraphicsHeight()*0.74,camara_radar$
End If

;Dibuja las texturas animadas=====
If num_anime<>0 Then
;Definimos el foto/segundo
foto=MilliSecs()/80 Mod 36
For cont_3D=1 To NUM_ANIME
Select cont_3D
    Case 1
        EntityTexture obra(anime(1))\malla,tex_a1,foto
    Case 2
        EntityTexture obra(anime(2))\malla,tex_a2,foto
    Case 3
        EntityTexture obra(anime(3))\malla,tex_a3,foto
    Case 4
        EntityTexture obra(anime(4))\malla,tex_a4,foto
    Case 5
        EntityTexture obra(anime(5))\malla,tex_a5,foto
    Case 6
        EntityTexture obra(anime(6))\malla,tex_a6,foto
    Case 7
        EntityTexture obra(anime(7))\malla,tex_a7,foto
    Case 8
        EntityTexture obra(anime(8))\malla,tex_a8,foto
    Case 9
        EntityTexture obra(anime(9))\malla,tex_a9,foto
    Case 10
        EntityTexture obra(anime(10))\malla,tex_a10,foto
End Select
Next
EndIf

;Escribe el nombre del Museo, Autor y Fecha de creación del mismo
dibuja_info()

; =====Regresa el nombre del objeto mas cercano e imprime su información=====
If EntityPick(player,100)<>0 Then;si esta a 50U de distancia....
    obj_colisionado$= EntityName(EntityPick(player,100));Obten el nombre del objeto
    For i=1 To NUM_OBRAS
        If EntityName(EntityPick(player,100))=obra(i)\nombre$ Then
            colisionado=i;Verifica cual es el objeto colisionado
        EndIf
    Next
    If muestra_info=1 Then

```

```

a_y=0
For y=3 To 20
    If cedula$(colisionado,y) <> "VACIO" Then
        If Right$(cedula$(colisionado,y),1)=":" Then
            SetFont f_ArialB
        Else
            SetFont f_Arial
        EndIf
        a_y=a_y+14
        Text 10,355+a_y,cedula$(colisionado,y)
    EndIf
Next
EndIf

EndIf

;=====Dibuja el cursor=====
DrawImage cursor,MouseX(),MouseY()
MaskImage cursor,11,1,55

Flip
UpdateWorld
RenderWorld

Wend

;===== INICIO DE FUNCIONES =====;
Function Parte_Renglon(renglon,inicio,longitud)
    While Mid$(linea$(renglon),inicio+longitud,1) <> ","
        longitud=longitud+1
    Wend
    Return longitud
End Function

Function Ultima_Parte_Renglon(renglon,inicio,longitud)
    While Mid$(linea$(renglon),inicio+longitud,1) <> ""
        longitud=longitud+1
    Wend
    Return longitud
End Function

Function dibuja_skin()
    If muestra_ayuda=1 Then

```

```

MaskImage v_ayuda,100,100,100
DrawImage v_ayuda,GraphicsWidth()*0.18,GraphicsHeight()*0.136
MaskImage salir_ayuda0,0,0,10
DrawImage salir_ayuda0,GraphicsWidth()*0.775,GraphicsHeight()*0.156
End If

If muestra_interfaz = 1 Then
DrawImage pleca,(GraphicsWidth()-ImageWidth(pleca))*0.508,GraphicsHeight()-ImageHeight(pleca)
MaskImage pleca,0,0,10
MaskImage salir_m0,77,84,84
DrawImage salir_m0,GraphicsWidth()*0.765,GraphicsHeight()*0.865

DrawImage z_ayuda,GraphicsWidth()-ImageWidth(z_ayuda),0
DrawImage ayuda0,GraphicsWidth()*0.915,GraphicsHeight()*0.007
MaskImage ayuda0,0,0,10
DrawImage salir0,GraphicsWidth()*0.955,GraphicsHeight()*0.006
MaskImage salir0,0,0,10

DrawImage entrada0,GraphicsWidth()*0.725,GraphicsHeight()*0.898
MaskImage entrada0,0,0,10

If muestra_audio=1 Then
MaskImage audio0,0,0,10
DrawImage audio0,GraphicsWidth()*0.217,GraphicsHeight()*0.898
Else
MaskImage audio_off0,0,0,10
DrawImage audio_off0,GraphicsWidth()*0.217,GraphicsHeight()*0.898
EndIf

DrawImage subir0,GraphicsWidth()*0.484,GraphicsHeight()*0.878
MaskImage subir0,0,0,10
DrawImage bajar0,GraphicsWidth()*0.484,GraphicsHeight()*0.94
MaskImage bajar0,0,0,10
DrawImage arriba,GraphicsWidth()*0.580,GraphicsHeight()*0.889
MaskImage arriba,0,0,10
DrawImage abajo,GraphicsWidth()*0.580,GraphicsHeight()*0.927
MaskImage abajo,0,0,10
DrawImage izquierda,GraphicsWidth()*0.565,GraphicsHeight()*0.905
MaskImage izquierda,0,0,10
DrawImage derecha,GraphicsWidth()*0.602,GraphicsHeight()*0.904
MaskImage derecha,0,0,10
DrawImage r_arriba,GraphicsWidth()*0.406,GraphicsHeight()*0.881
MaskImage r_arriba,0,0,10
DrawImage r_abajo,GraphicsWidth()*0.406,GraphicsHeight()*0.937

```

```

MaskImage r_abajo,0,0,10
DrawImage r_izq,GraphicsWidth()*0.390,GraphicsHeight()*0.905
MaskImage r_izq,0,0,10
DrawImage r_der,GraphicsWidth()*0.433,GraphicsHeight()*0.904
MaskImage r_der,0,0,10

```

```

If muestra_info=1 Then
    MaskImage info0,0,0,10
    DrawImage info0,GraphicsWidth()*0.287,GraphicsHeight()*0.898
Else
    MaskImage info_off0,0,0,10
    DrawImage info_off0,GraphicsWidth()*0.287,GraphicsHeight()*0.898
EndIf

```

```

If muestra_radar=1 Then
    Text GraphicsWidth()*0.86,GraphicsHeight()*0.74,camara_radar$
    MaskImage radar0,0,0,10
    DrawImage radar0,GraphicsWidth()*0.66,GraphicsHeight()*0.898
Else
    MaskImage radar_off0,0,0,10
    DrawImage radar_off0,GraphicsWidth()*0.66,GraphicsHeight()*0.898
EndIf

```

```

dibuja_info()

```

```

;Si no se dibuja el cursor aqui tambien, en las animaciones del radar y menu de informaciòn, el cursor no se vería
DrawImage cursor,MouseX(),MouseY()
MaskImage cursor,11,1,55

```

```

End If

```

```

End Function

```

```

Function dibuja_info()
If muestra_info=1 Then
    DrawImage z_info,0,GraphicsHeight()-ImageHeight(z_info)
    SetFont f_arial
    Text GraphicsWidth()*0.09,GraphicsHeight()*0.50,museo$,True,False
    Text GraphicsWidth()*0.09,GraphicsHeight()*0.53,"Autor: "+autor$,True,False
    Text GraphicsWidth()*0.09,GraphicsHeight()*0.56,fecha$,True,False
EndIf
End Function

```

```

Function botones()

```

```

If muestra_interfaz = 1 Then

```

```

;Función para la presión del botón 'Radar'
If ImagesCollide (pixel,MouseX(),MouseY(),0,radar0,GraphicsWidth()*0.66,GraphicsHeight()*0.898,0) Then
    If muestra_radar=1 Then
        MaskImage radar1,0,0,10
        DrawImage radar1,GraphicsWidth()*0.66,GraphicsHeight()*0.898
    Else
        MaskImage radar_off1,0,0,10
        DrawImage radar_off1,GraphicsWidth()*0.66,GraphicsHeight()*0.898
    EndIf

    DrawImage e_radar,MouseX(),MouseY()+22

    If MouseDown(1)=1 Then
        muestra_radar=1-muestra_radar
        Delay 200;Retrasa 1/5 de segundo
        If muestra_radar=0 Then
            CameraViewport cam_sup1,0,0,0,0
        Else
            CameraViewport
cam_sup1,GraphicsWidth()*0.81,GraphicsHeight()*0.77,GraphicsWidth()*0.19,GraphicsHeight()*0.23
        EndIf
    EndIf
EndIf

;Función para la presión del botón 'INFO'
If ImagesCollide (pixel,MouseX(),MouseY(),0,info0,GraphicsWidth()*0.287,GraphicsHeight()*0.898,0) Then
    If muestra_info=1 Then
        MaskImage info1,0,0,10
        DrawImage info1,GraphicsWidth()*0.287,GraphicsHeight()*0.898
    Else
        MaskImage info_off1,0,0,10
        DrawImage info_off1,GraphicsWidth()*0.287,GraphicsHeight()*0.898
    EndIf

    DrawImage e_info,MouseX(),MouseY()+22

    If MouseDown(1)=1 Then
        SetBuffer FrontBuffer()
        muestra_info=1-muestra_info
        Delay 200;Retardo de 1/5 [s] en la presión del botón 'Info'
        If muestra_info=0 Then
            DrawImage z_info,-200,GraphicsHeight()-ImageHeight(z_info)
        Else
            DrawImage z_info,0,GraphicsHeight()-ImageHeight(z_info)
        EndIf
    EndIf
EndIf

```

```

;====Función para la presión del botón 'ENTRADA'=====
If ImagesCollide (pixel,MouseX(),MouseY(),0,entrada0,GraphicsWidth()*0.725,GraphicsHeight()*0.898,0) Then
    MaskImage entrada1,0,0,10
    DrawImage entrada1,GraphicsWidth()*0.725,GraphicsHeight()*0.898
    DrawImage e_inicio,MouseX(),MouseY()+22
    If MouseDown(1)=1 Then
        EntityType player,2
        PositionEntity player,helper_inicio\Px#,helper_inicio\Py#,helper_inicio\Pz#
        RotateEntity player,helper_inicio\Rx#,helper_inicio\Ry#,helper_inicio\Rz#
        RotateEntity camera,0,0,0
        EntityType player,1
    EndIf
EndIf

; =====Función para la presión del botón 'SALIR'=====
If ImagesCollide (pixel,MouseX(),MouseY(),0,salir0,GraphicsWidth()*0.955,GraphicsHeight()*0.006,0) Then
    MaskImage salir1,0,0,10
    DrawImage salir1,GraphicsWidth()*0.955,GraphicsHeight()*0.006
    DrawImage e_salir,MouseX(),MouseY()+22
    If MouseDown(1)=1 Then
        End
    EndIf
EndIf

; =====Función para la presión del botón 'ADELANTE'=====
If ImagesCollide (pixel,MouseX(),MouseY(),0,arriba,GraphicsWidth()*0.580,GraphicsHeight()*0.889,0) Then
    ;MaskImage arriba2,0,0,0
    ;DrawImage arriba2,GraphicsWidth()*0.5,GraphicsHeight()*0.89
    DrawImage e_adelante,MouseX(),MouseY()+22
    If MouseDown(1)=1 Then
        MoveEntity player,0,0,2
    EndIf
EndIf

; =====Función para la presión del botón 'ATRAS'=====
If ImagesCollide (pixel,MouseX(),MouseY(),0,abajo,GraphicsWidth()*0.580,GraphicsHeight()*0.927,0) Then
    ;MaskImage abajo2,0,0,0
    ;DrawImage abajo2,GraphicsWidth()*0.5,GraphicsHeight()*0.96
    DrawImage e_atras,MouseX(),MouseY()+22
    If MouseDown(1)=1 Then
        MoveEntity player,0,0,-2
    EndIf
EndIf

; =====Función para la presión del botón 'IZQUIERDA'=====
If ImagesCollide (pixel,MouseX(),MouseY(),0,izquierda,GraphicsWidth()*0.565,GraphicsHeight()*0.905,0) Then

```

```

;MaskImage izquierda2,0,0,0
;DrawImage izquierda2,GraphicsWidth()*0.47,GraphicsHeight()*0.925
DrawImage e_izq,MouseX(),MouseY()+22
If MouseDown(1)=1 Then
    MoveEntity player,-2,0,0
EndIf
EndIf

; =====Función para la presión del botón 'DERECHA'=====
If ImagesCollide (pixel,MouseX(),MouseY(),0,derecha,GraphicsWidth()*0.602,GraphicsHeight()*0.904,0) Then
    ;MaskImage derecha2,0,0,0
    ;DrawImage derecha2,GraphicsWidth()*0.53,GraphicsHeight()*0.925
    DrawImage e_der,MouseX(),MouseY()+22
    If MouseDown(1)=1 Then
        MoveEntity player,2,0,0
    EndIf
EndIf

; =====Función para la presión del botón 'ROTA IZQUIERDA'=====
If ImagesCollide (pixel,MouseX(),MouseY(),0,r_izq,GraphicsWidth()*0.390,GraphicsHeight()*0.905,0) Then
    ;MaskImage izquierda2,0,0,0
    ;DrawImage izquierda2,GraphicsWidth()*0.47,GraphicsHeight()*0.925
    DrawImage e_rizq,MouseX(),MouseY()+22
    If MouseDown(1)=1 Then
        TurnEntity player,0,1,0
    EndIf
EndIf

; =====Función para la presión del botón 'ROTA DERECHA'=====
If ImagesCollide (pixel,MouseX(),MouseY(),0,r_der,GraphicsWidth()*0.433,GraphicsHeight()*0.904,0) Then
    ;MaskImage derecha2,0,0,0
    ;DrawImage derecha2,GraphicsWidth()*0.53,GraphicsHeight()*0.925
    DrawImage e_rder,MouseX(),MouseY()+22
    If MouseDown(1)=1 Then
        TurnEntity player,0,-1,0
    EndIf
EndIf

; =====Función para la presión del botón 'ROTA ARRIBA'=====
If ImagesCollide (pixel,MouseX(),MouseY(),0,r_arriba,GraphicsWidth()*0.406,GraphicsHeight()*0.881,0) Then
    ;MaskImage rota_arriba2,0,0,0
    ;DrawImage rota_arriba2,GraphicsWidth()*0.55,GraphicsHeight()*0.86
    DrawImage e_rarriba,MouseX(),MouseY()+22
    If MouseDown(1)=1 Then
        TurnEntity camera,-1,0,0
    EndIf
EndIf

```

```

; =====Función para la presión del botón 'ROTA ABAJO'=====
If ImagesCollide (pixel,MouseX(),MouseY(),0,r_abajo,GraphicsWidth()*0.406,GraphicsHeight()*0.937,0) Then
    ;MaskImage rota_abajo2,0,0,0
    ;DrawImage rota_abajo2,GraphicsWidth()*0.55,GraphicsHeight()*0.91
    DrawImage e_rabajo,MouseX(),MouseY()+22
    If MouseDown(1)=1 Then
        TurnEntity camera,1,0,0
    EndIf
EndIf

; =====Función para la presión del botón 'SUBE'=====
If ImagesCollide (pixel,MouseX(),MouseY(),0,subir0,GraphicsWidth()*0.484,GraphicsHeight()*0.878,0) Then
    MaskImage subir1,0,0,10
    DrawImage subir1,GraphicsWidth()*0.484,GraphicsHeight()*0.878
    DrawImage e_sube,MouseX(),MouseY()+22
    If MouseDown(1)=1 Then
        MoveEntity player,0,2,0
    EndIf
EndIf

; =====Función para la presión del botón 'BAJA'=====
If ImagesCollide (pixel,MouseX(),MouseY(),0,bajar0,GraphicsWidth()*0.484,GraphicsHeight()*0.94,0) Then
    MaskImage bajar1,0,0,10
    DrawImage bajar1,GraphicsWidth()*0.484,GraphicsHeight()*0.94
    DrawImage e_baja,MouseX(),MouseY()+22
    If MouseDown(1)=1 Then
        MoveEntity player,0,-2,0
    EndIf
EndIf

; =====Función para la presión del botón 'AUDIO'=====
If ImagesCollide (pixel,MouseX(),MouseY(),0,audio0,GraphicsWidth()*0.217,GraphicsHeight()*0.898,0) Then
    If muestra_audio=1 Then
        MaskImage audio1,0,0,10
        DrawImage audio1,GraphicsWidth()*0.217,GraphicsHeight()*0.898
    Else
        MaskImage audio_off1,0,0,10
        DrawImage audio_off1,GraphicsWidth()*0.217,GraphicsHeight()*0.898
    EndIf
    DrawImage e_audio,MouseX(),MouseY()+22
    If MouseDown(1)=1 Then
        muestra_audio=1-muestra_audio
        Delay 300
        If muestra_audio=0 Then
            For i=1 To NUM_AUDIOS
                PositionEntity helper_audio(i)\nombre,10000,-90000,10000
            Next i
        EndIf
    EndIf
EndIf

```

```

        Next
    Else
        For i=1 To NUM_AUDIOS
            PositionEntity helper_audio(i)\nombre,helper_audio(i)\Px#,helper_audio(i)\Py#,helper_audio(i)\Pz#
        Next
    EndIf
EndIf
EndIf

; =====Función para la presión del botón 'AYUDA'=====
If ImagesCollide (pixel,MouseX(),MouseY(),0,ayuda0,GraphicsWidth()*0.915,GraphicsHeight()*0.007,0) Then
    MaskImage ayuda1,0,0,10
    DrawImage ayuda1,GraphicsWidth()*0.915,GraphicsHeight()*0.007
    DrawImage e_ayuda,MouseX(),MouseY()+22
    If MouseDown(1)=1 Then
        muestra_ayuda=1-muestra_ayuda
        Delay 250
        MaskImage v_ayuda,100,100,100
        DrawImage v_ayuda,GraphicsWidth()*0.18,GraphicsHeight()*0.136
        MaskImage salir_ayuda0,0,0,10
        DrawImage salir_ayuda0,GraphicsWidth()*0.775,GraphicsHeight()*0.156
    EndIf
EndIf

; =====Función para la presión del botón 'SALIR_AYUDA'=====
If ImagesCollide (pixel,MouseX(),MouseY(),0,salir_ayuda0,GraphicsWidth()*0.775,GraphicsHeight()*0.156,0) Then
    If muestra_ayuda=1 Then
        MaskImage salir_ayuda1,0,0,10
        DrawImage salir_ayuda1,GraphicsWidth()*0.775,GraphicsHeight()*0.156

        If MouseDown(1)=1 Then
            muestra_ayuda=0
            Delay 250
        EndIf
    End If
EndIf

End If

; =====Función para la presión del botón 'SALIR_PLECA'=====
If ImagesCollide (pixel,MouseX(),MouseY(),0,salir_m0,GraphicsWidth()*0.765,GraphicsHeight()*0.865,0) Then
    If muestra_interfaz=1 Then
        MaskImage salir_m1,77,84,84
        DrawImage salir_m1,GraphicsWidth()*0.765,GraphicsHeight()*0.865
        If MouseDown(1)=1 Then
            muestra_interfaz=0
            Delay 250
        EndIf
    EndIf
EndIf

```

```

End If
End If

End Function

; =====Navegacion=====
Function teclas()
  If KeyDown(56) Or KeyDown(15) Then;ALT+TAB
    End
  EndIf

  If KeyDown(219) Then;WINDOWS IZQ
    End
  EndIf

  If KeyDown(220) Then;WINDOWS DER
    End
  EndIf

  If KeyDown(200) And (KeyDown(29) Or KeyDown(157)) Then
Arriba)                                     ;CTRL + Adelante (Rota
    TurnEntity player,-1,0,0
  Else
    If KeyDown(200) And (KeyDown(42) Or KeyDown(54)) Then
                                           ;SHIFT + Adelante (Sube)
      MoveEntity player,0,2,0
    Else
      If KeyDown(200) And KeyDown(44) Then
        MoveEntity player,0,0,10
      Else
        If KeyDown(200) MoveEntity player,0,0,2
                                           ;Adelante (Avanza)
        End If
      EndIf
    EndIf
  EndIf

  If KeyDown(208) And (KeyDown(29) Or KeyDown(157)) Then
Abajo)                                     ;CTRL + Atrás (Rota
    TurnEntity player,1,0,0
  Else
    If KeyDown(208) And (KeyDown(42) Or KeyDown(54)) Then
                                           ;SHIFT + Atrás (Baja)
      MoveEntity player,0,-2,0
    Else
      If KeyDown(208) And KeyDown(44) Then
        MoveEntity player,0,0,-10
      Else
        If KeyDown(208) MoveEntity player,0,0,-2
                                           ;Atrás (Retrocede)
        End If
      End If
    End If
  End If

```

EndIf
EndIf

If KeyDown(203) And (KeyDown(29) Or KeyDown(157)) Then ;CTRL + Izq (Rota Izq)
 TurnEntity player,0,1,0
Else
 If KeyDown(203) And KeyDown(44) Then
 MoveEntity player,-10,0,0
 Else
 If KeyDown(203) MoveEntity player,-2,0,0 ;Izq (Desplaza Izq)
 End If
EndIf

If KeyDown(205) And (KeyDown(29) Or KeyDown(157)) Then ;CTRL + Der (Rota Der)
 TurnEntity player,0,-1,0
Else
 If KeyDown(205) And KeyDown(44) Then
 MoveEntity player,10,0,0
 Else
 If KeyDown(205) MoveEntity player,2,0,0 ;Der (Desplaza Der)
 End If
EndIf

If KeyDown(57) Then ;Espaciadora
(INICIO)
 EntityType player,2
 PositionEntity player,helper_inicio\Px#,helper_inicio\Py#,helper_inicio\Pz#
 RotateEntity player,helper_inicio\Ry#,helper_inicio\Rz#,helper_inicio\Rz#
 RotateEntity camera,0,0,0
 EntityType player,1
EndIf

If KeyDown(63) Then ;F5 (Cambio de color en Esfera Radar)
 rgb=rgb+1
 If rgb>6 Then
 rgb=1
 EndIf

 Select rgb
 Case 1
 EntityColor esfera_camara,0,0,0
 Case 2
 EntityColor esfera_camara,255,255,0
 Case 3
 EntityColor esfera_camara,255,0,0

```

Case 4
    EntityColor esfera_camara,0,255,0
Case 5
    EntityColor esfera_camara,0,0,255
Case 6
    EntityColor esfera_camara,255,255,255
End Select

Delay 250 ;retrasa 1/4 de segundo
EndIf

If KeyHit(17)                                     ;W (Malla de alambre)
    wire=1-wire
    WireFrame wire
EndIf

If KeyDown(62) Then                               ;F4 (Cambio de Cámara)
    If muestra_radar=1 num_radar=num_radar+1
    Select num_radar
        Case 1    PositionEntity cam_sup1,promX,promY,-hz+menorZ());Camara Frontal
                    RotateEntity cam_sup1,0,0,0
                    camara_radar$="Camara Frontal"
                    Delay 400;Retrasa 2/5 de segundo
        Case 2    PositionEntity cam_sup1,promX,promY,hz+mayorZ());Camara Trasera
                    RotateEntity cam_sup1,0,180,0
                    camara_radar$="Camara Trasera"
                    Delay 400;Retrasa 2/5 de segundo
        Case 3    PositionEntity cam_sup1,promX,hy+mayorY(),promZ;Camara superior
                    RotateEntity cam_sup1,90,0,0
                    camara_radar$="Camara Superior"
                    Delay 400;Retrasa 2/5 de segundo
        Case 4    PositionEntity cam_sup1,promX,-hy+menorY(),promZ;Camara Inferior
                    RotateEntity cam_sup1,270,0,0
                    camara_radar$="Camara Inferior"
                    Delay 400;Retrasa 2/5 de segundo
        Case 5    PositionEntity cam_sup1,hx+mayorX(),promY,promZ;Camara Izquierda
                    RotateEntity cam_sup1,0,90,0
                    camara_radar$="Camara Izquierda"
                    Delay 400;Retrasa 2/5 de segundo
        Case 6    PositionEntity cam_sup1,-hx+menorX(),promY,promZ;Camara Derecha
                    RotateEntity cam_sup1,0,270,0
                    num_radar=0
                    camara_radar$="Camara Derecha"
                    Delay 400;Retrasa 2/5 de segundo
    End Select
EndIf

```

```

If KeyDown(60) Then
(Muestra/Oculto informacion)
    muestra_info=1-muestra_info
    Delay 250;Retardo de 1/5 [s] en la presión del botón 'Info'
    If muestra_info=0 Then
        DrawImage z_info,-200,GraphicsHeight()-ImageHeight(z_info)
    Else
        DrawImage z_info,0,GraphicsHeight()-ImageHeight(z_info)
    EndIf
EndIf

```

;F2

```

If KeyDown(61) Then
    muestra_radar=1-muestra_radar
(Muestra/Oculto el Radar )
    Delay 250;Retrasa 1/5 de segundo
    If muestra_radar=0 Then
        CameraViewport cam_sup1,0,0,0,0
    Else
        CameraViewport
cam_sup1,GraphicsWidth()*0.81,GraphicsHeight()*0.77,GraphicsWidth()*0.19,GraphicsHeight()*0.23
    EndIf
EndIf

```

;F3

```

If KeyDown(59) Then
    muestra_ayuda=1-muestra_ayuda
    Delay 250
    MaskImage v_ayuda,100,100,100
    DrawImage v_ayuda,GraphicsWidth()*0.18,GraphicsHeight()*0.136
    MaskImage salir_ayuda0,0,0,10
    DrawImage salir_ayuda0,GraphicsWidth()*0.775,GraphicsHeight()*0.156
EndIf

```

;F1 (AYUDA)

```

If KeyDown(64) Then
    muestra_interfaz=1-muestra_interfaz
    Delay 250
End If

```

;F6 (Muestra/Oculto la interfaz)

```

If KeyDown(65) Then
    muestra_audio=1-muestra_audio
    Delay 300
    If muestra_audio=0 Then
        For i=1 To NUM_AUDIOS
            ;PauseChannel sonido
            PositionEntity helper_audio(i)\nombre,10000,-90000,10000
        Next
    Else

```

;F7 (Enciende/Apaga el audio)

```

        For i=1 To NUM_AUDIOS
            ;ResumeChannel helper_audio(i)\nombre
            PositionEntity helper_audio(i)\nombre,helper_audio(i)\Px#,helper_audio(i)\Py#,helper_audio(i)\Pz#
        Next
    EndIf
End If

End Function

Function planoXZ()
    distX=((mayorX()-menorX())/2)
    distZ=((mayorZ()-menorZ())/2)
    If distX > distZ Then
        distXZ=distX
    Else
        distXZ=distZ
    EndIf
    Return distXZ
End Function

Function planoXY()
    distX=((mayorX()-menorX())/2)
    distY=((mayorY()-menorY())/2)
    If distX > distY Then
        distXY=distX
    Else
        distXY=distY
    EndIf
    Return distXY
End Function

Function planoYZ()
    distZ=((mayorZ()-menorZ())/2)
    distY=((mayorY()-menorY())/2)
    If distZ > distY Then
        distYZ=distZ
    Else
        distYZ=distY
    EndIf
    Return distYZ
End Function

Function menorX()
    For i=1 To NUM_ARQ

```

```

    If EntityX#(els_arq(i)\malla) < menorX Then
        menorX=EntityX#(els_arq(i)\malla)
    EndIf
Next
For i=1 To NUM_OBRAS
    If EntityX#(obra(i)\malla) < menorX Then
        menorX=EntityX#(obra(i)\malla)
    EndIf
Next
If EntityX#(player) < menorX Then
    menorX=EntityX#(player)
EndIf
Return menorX
End Function

```

```

Function menorY()
For i=1 To NUM_ARQ
    If EntityY#(els_arq(i)\malla) < menorY Then
        menorY=EntityY#(els_arq(i)\malla)
    EndIf
Next
For i=1 To NUM_OBRAS
    If EntityY#(obra(i)\malla) < menorY Then
        menorY=EntityY#(obra(i)\malla)
    EndIf
Next
If EntityY#(player) < menorY Then
    menorY=EntityY#(player)
EndIf
Return menorY
End Function

```

```

Function menorZ()
For i=1 To NUM_ARQ
    If EntityZ#(els_arq(i)\malla) < menorZ Then
        menorZ=EntityZ#(els_arq(i)\malla)
    EndIf
Next
For i=1 To NUM_OBRAS
    If EntityZ#(obra(i)\malla) < menorZ Then
        menorZ=EntityZ#(obra(i)\malla)
    EndIf
Next
If EntityZ#(player) < menorZ Then

```

```

    menorZ=EntityZ#(player)
EndIf
Return menorZ
End Function

```

```

Function mayorX()
For i=1 To NUM_ARQ
    If EntityX#(els_arq(i)\malla) > mayorX Then
        mayorX=EntityX#(els_arq(i)\malla)
    EndIf
Next
For i=1 To NUM_OBRAS
    If EntityX#(obra(i)\malla) > mayorX Then
        mayorX=EntityX#(obra(i)\malla)
    EndIf
Next
If EntityX#(player) > mayorX Then
    mayorX=EntityX#(player)
EndIf
Return mayorX
End Function

```

```

Function mayorY()
For i=1 To NUM_ARQ
    If EntityY#(els_arq(i)\malla) > mayorY Then
        mayorY=EntityY#(els_arq(i)\malla)
    EndIf
Next
For i=1 To NUM_OBRAS
    If EntityY#(obra(i)\malla) > mayorY Then
        mayorY=EntityY#(obra(i)\malla)
    EndIf
Next
If EntityY#(player) > mayorY Then
    mayorY=EntityY#(player)
EndIf
Return mayorY
End Function

```

```

Function mayorZ()
For i=1 To NUM_ARQ
    If EntityZ#(els_arq(i)\malla) > mayorZ Then
        mayorZ=EntityZ#(els_arq(i)\malla)
    EndIf
Next
If EntityZ#(player) > mayorZ Then
    mayorZ=EntityZ#(player)
EndIf
Return mayorZ
End Function

```

```

EndIf
Next
For i=1 To NUM_OBRAS
  If EntityZ#(obra(i)\malla) > mayorZ Then
    mayorZ=EntityZ#(obra(i)\malla)
  EndIf
Next
If EntityZ#(player) > mayorZ Then
  mayorZ=EntityZ#(player)
EndIf
Return mayorZ
End Function

```

```

Function Radar()
promX=((mayorX()-menorX())/2)+menorX()
promY=((mayorY()-menorY())/2)+menorY()
promZ=((mayorZ()-menorZ())/2)+menorZ()

```

```

hy= planoXZ()/Tan(35)
hz= planoXY()/Tan(35)
hx= planoYZ()/Tan(35)

```

```

Select num_radar
  Case 1    PositionEntity cam_sup1,promX,promY,-hz+menorZ();Camara Frontal
            RotateEntity cam_sup1,0,0,0
            camara_radar$="Camara Frontal"

  Case 2    PositionEntity cam_sup1,promX,promY,hz+mayorZ();Camara Trasera
            RotateEntity cam_sup1,0,180,0
            camara_radar$="Camara Trasera"

  Case 3    PositionEntity cam_sup1,promX,hy+mayorY(),promZ;Camara superior
            RotateEntity cam_sup1,90,0,0
            camara_radar$="Camara Superior"

  Case 4    PositionEntity cam_sup1,promX,-hy+menorY(),promZ;Camara Inferior
            RotateEntity cam_sup1,270,0,0
            camara_radar$="Camara Inferior"

  Case 5    PositionEntity cam_sup1,hx+mayorX(),promY,promZ;Camara Izquierda
            RotateEntity cam_sup1,0,90,0
            camara_radar$="Camara Izquierda"

  Case 6,0  PositionEntity cam_sup1,-hx+menorX(),promY,promZ;Camara Derecha
            RotateEntity cam_sup1,0,270,0

```

```

num_radar=0
camara_radar$="Camara Derecha"

End Select

End Function
;===== Libera Recursos =====
FreeFont f_Arial
FreeFont f_ArialB
FreeFont f_ArialC
CloseFile lee_archivo
;=====

End

```

Sistema Dionisios: Editor de Museos

Plataforma: PC compatible

Requerimientos técnicos: Windows XP, 512MB en RAM, 20MB disponibles en disco, Mouse y Teclado, Resolución de Monitor 800x600 o superior.

Compilador: Biltz3D (<http://www.blitzbasic.com/Products/blitz3d.php>)

Tamaño aprox. 2.7MB

Descripción:

El editor permite construir las exposiciones virtuales con un número no mayor a 20 piezas seleccionadas a través del sitio Web del museo. El usuario dispone de imágenes y sonidos para la construcción de su sala. Así mismo genera el script que se requiere para que Apolo lo despliegue al usuario final de la exposición. Se comunica al servidor a través de "Oráculo" para: a) descargar las imágenes de las obras seleccionadas (en formato de baja resolución) y b) para dejar el script en el servidor y que quede disponible para la comunidad de usuarios del MV.

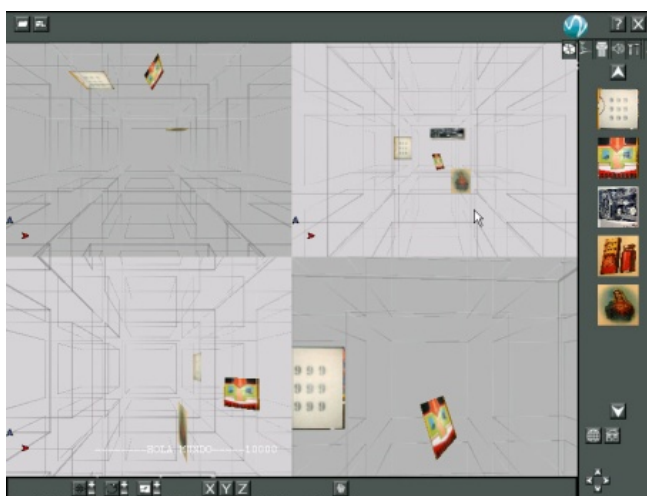


Figura.H. 4. Interfaz del Editor 3D (Dionisios)

Editor de ambientes virtuales, que permite la implementación de las exposiciones respetando la tridimensionalidad del discurso curatorial y los complementos museográficos.

Código Fuente:

```
; CCADET-UNAM
; Copyright (c) 2007
;
; Created: 31.08.2007 7:19
; Version: 0.9
; Editor de Museos Virtuales, Dionisios
```

```
Include "ImageGui.bb"
Include "conts.bb"
Include "gui.bb"
Include "typeObject3d.bb"
Include "typeCamera.bb"
Include "ActionCamera.bb"
```

```
;-----VARIABLES GLOBALES-----
```

```
Global tDirFolder1.ArrayButtons= New ArrayButtons
Global tDirFolder2.ArrayButtons= New ArrayButtons
Global tDirFolder3.ArrayButtons= New ArrayButtons
Global tDirFolder4.ArrayButtons= New ArrayButtons
Global tDirFolder5.ArrayButtons= New ArrayButtons
Global tDirFolder6.ArrayButtons= New ArrayButtons ; ADDNEW
```

```
Global tRightGuiBar.ArrayButtons= New ArrayButtons
Global tDownGuiBar.ArrayButtons= New ArrayButtons
Global tImageGui.ArrayImage= New ArrayImage
```

```
Global Listener
Global NAME_WORLD$
Global channel
Global MeshLoaded=False
Global debug$= " "
```

```
Function Open(sNameFile$="world_temp.amv")
  If(aObjects3d\iCount> 0) Then
    Return
  End If
  sFileOut=ReadFile(sNameFile)
  ;BEGIN ADDNEW
  ;I have to check if there is a mesh: if the file begins with 1, then there is, otherwise not.
  vbExist=ReadInt(sFileOut)
  If vbExist=1 Then
    MeshLoaded=True
```

```

; I need to read the rest of the info about the mesh
NAME_WORLD$=ReadString(sFileOut)
MeshTemp=LoadMesh(NAME_WORLD$)
EntityAlpha MeshTemp,0.35
PositionEntity MeshTemp,SIZE_WORLD_X,SIZE_WORLD_Y,SIZE_WORLD_Z
Else
MeshLoaded= False
End If
;END ADDNEW
aObjects3d\iCount = 0
While Not Eof( sFileOut )
    aObjects3d\iMesh[aObjects3d\iCount] = New Object3d;
    aObjects3d\iMesh[aObjects3d\iCount]\iType = ReadInt ( sFileOut )
    aObjects3d\iMesh[aObjects3d\iCount]\cName$ = ReadString( sFileOut )
    aObjects3d\iMesh[aObjects3d\iCount]\cTextureName$ = ReadString( sFileOut )
    aObjects3d\iMesh[aObjects3d\iCount]\fX# = ReadFloat ( sFileOut )
    aObjects3d\iMesh[aObjects3d\iCount]\fY# = ReadFloat ( sFileOut )
    aObjects3d\iMesh[aObjects3d\iCount]\fZ# = ReadFloat ( sFileOut )
    if the object that I am reading is a sound button, then I have only the range information stored
    ;into the fScaleX
    If aObjects3d\iMesh[aObjects3d\iCount]\iType<>BUTTON_SOUND Then
        aObjects3d\iMesh[aObjects3d\iCount]\fRotationX# = ReadFloat ( sFileOut )
        aObjects3d\iMesh[aObjects3d\iCount]\fRotationY# = ReadFloat ( sFileOut )
        aObjects3d\iMesh[aObjects3d\iCount]\fRotationZ# = ReadFloat ( sFileOut )
    End If
    ; I read the next information, that for sound is the range
    aObjects3d\iMesh[aObjects3d\iCount]\fScaleX# = ReadFloat ( sFileOut )
    ;these two are also just for other types of objects
    If aObjects3d\iMesh[aObjects3d\iCount]\iType<>BUTTON_SOUND Then
        aObjects3d\iMesh[aObjects3d\iCount]\fScaleY# = ReadFloat ( sFileOut )
        aObjects3d\iMesh[aObjects3d\iCount]\fScaleZ# = ReadFloat ( sFileOut )
    End If
;END ADDNEW
    If( aObjects3d\iMesh[aObjects3d\iCount]\iType = BUTTON_PICTURE) Or (
aObjects3d\iMesh[aObjects3d\iCount]\iType = BUTTON_FILM_STRIP) Then
        aObjects3d\iMesh[aObjects3d\iCount]\iMesh=CreateCube();

        If ( Len (aObjects3d\iMesh[aObjects3d\iCount]\cTextureName) > 0) Then
            iTexture=LoadTexture(aObjects3d\iMesh[aObjects3d\iCount]\cTextureName)
            If( iTexture= 0 ) Then
                RuntimeError "Error al cargar la textura, posible driver incompatible"
            End
        End If
        EntityTexture aObjects3d\iMesh[aObjects3d\iCount]\iMesh,iTexture
    End If
    ScaleEntity aObjects3d\iMesh[aObjects3d\iCount]\iMesh,
aObjects3d\iMesh[aObjects3d\iCount]\fScaleX,aObjects3d\iMesh[aObjects3d\iCount]\fScaleY,aObjects3d\iMesh[a
Objects3d\iCount]\fScaleZ

```

```

        PositionEntity
aObjects3d\lvMesh[aObjects3d\iCount]\iMesh,aObjects3d\lvMesh[aObjects3d\iCount]\fX,aObjects3d\lvMesh[aObjects3d\iC
ount]\fY,aObjects3d\lvMesh[aObjects3d\iCount]\fZ

        RotateEntity
aObjects3d\lvMesh[aObjects3d\iCount]\iMesh,aObjects3d\lvMesh[aObjects3d\iCount]\fRotationX,aObjects3d\lvMesh[aObje
cts3d\iCount]\fRotationY,aObjects3d\lvMesh[aObjects3d\iCount]\fRotationZ

        EntityPickMode aObjects3d\lvMesh[aObjects3d\iCount]\iMesh,2
        treat the case of sound button
        Else If ( aObjects3d\lvMesh[aObjects3d\iCount]\iType = BUTTON_SOUND) Then
            aObjects3d\lvMesh[aObjects3d\iCount]\iMesh=CreateSphere();
            ;scale the sphere as it was defined earlier

            ScaleEntity
aObjects3d\lvMesh[aObjects3d\iCount]\fScaleX,aObjects3d\lvMesh[aObjects3d\iCount]\fScaleX,aObjects3d\lvMesh[a
Objects3d\iCount]\fScaleX
            PositionEntity
aObjects3d\lvMesh[aObjects3d\iCount]\iMesh,aObjects3d\lvMesh[aObjects3d\iCount]\fX,aObjects3d\lvMesh[aObjects3d\iC
ount]\fY,aObjects3d\lvMesh[aObjects3d\iCount]\fZ
            EntityPickMode aObjects3d\lvMesh[aObjects3d\iCount]\iMesh,2

            If ( Len (aObjects3d\lvMesh[aObjects3d\iCount]\cTextureName) > 0) Then
                iTexture2=LoadTexture(aObjects3d\lvMesh[aObjects3d\iCount]\cTextureName )
                If( iTexture2 = 0 ) Then
                    RuntimeError      "Error al cargar la textura, posible driver incompatible"
                End
            End If
            EntityTexture aObjects3d\lvMesh[aObjects3d\iCount]\iMesh,iTexture2
        End If
    ;END ADDNEW
    Else If ( aObjects3d\lvMesh[aObjects3d\iCount]\iType = CUBE) Then
        aObjects3d\lvMesh[aObjects3d\iCount]\iMesh=CreateCube();

        ScaleEntity
aObjects3d\lvMesh[aObjects3d\iCount]\fScaleX,aObjects3d\lvMesh[aObjects3d\iCount]\fScaleY,aObjects3d\lvMesh[a
Objects3d\iCount]\fScaleZ
        PositionEntity
aObjects3d\lvMesh[aObjects3d\iCount]\iMesh,aObjects3d\lvMesh[aObjects3d\iCount]\fX,aObjects3d\lvMesh[aObjects3d\iC
ount]\fY,aObjects3d\lvMesh[aObjects3d\iCount]\fZ
        RotateEntity
aObjects3d\lvMesh[aObjects3d\iCount]\iMesh,aObjects3d\lvMesh[aObjects3d\iCount]\fRotationX,aObjects3d\lvMesh[aObje
cts3d\iCount]\fRotationY,aObjects3d\lvMesh[aObjects3d\iCount]\fRotationZ
        EntityPickMode aObjects3d\lvMesh[aObjects3d\iCount]\iMesh,2

        If ( Len (aObjects3d\lvMesh[aObjects3d\iCount]\cTextureName) > 0) Then
            iTexture2=LoadTexture(aObjects3d\lvMesh[aObjects3d\iCount]\cTextureName )
            If( iTexture2 = 0 ) Then
                RuntimeError      "Error al cargar la textura, posible driver incompatible"
            End
        End If
        EntityTexture aObjects3d\lvMesh[aObjects3d\iCount]\iMesh,iTexture2
    End If

    Else If ( aObjects3d\lvMesh[aObjects3d\iCount]\iType = CONE) Then

```

```

        aObjects3d\lMesh[aObjects3d\iCount]\iMesh=CreateCone();
        ScaleEntity                                aObjects3d\lMesh[aObjects3d\iCount]\iMesh,
        aObjects3d\lMesh[aObjects3d\iCount]\fScaleX,aObjects3d\lMesh[aObjects3d\iCount]\fScaleY,aObjects3d\lMesh[a
Objects3d\iCount]\fScaleZ
        PositionEntity
        aObjects3d\lMesh[aObjects3d\iCount]\iMesh,aObjects3d\lMesh[aObjects3d\iCount]\fX,aObjects3d\lMesh[aObjects3d\iC
ount]\fY,aObjects3d\lMesh[aObjects3d\iCount]\fZ
        RotateEntity
        aObjects3d\lMesh[aObjects3d\iCount]\iMesh,aObjects3d\lMesh[aObjects3d\iCount]\fRotationX,aObjects3d\lMesh[aObje
cts3d\iCount]\fRotationY,aObjects3d\lMesh[aObjects3d\iCount]\fRotationZ
        EntityPickMode aObjects3d\lMesh[aObjects3d\iCount]\iMesh,2
        Else If ( aObjects3d\lMesh[aObjects3d\iCount]\iType = SPHERE) Then
            aObjects3d\lMesh[aObjects3d\iCount]\iMesh=CreateSphere();
            ScaleEntity                                aObjects3d\lMesh[aObjects3d\iCount]\iMesh,
            aObjects3d\lMesh[aObjects3d\iCount]\fScaleX,aObjects3d\lMesh[aObjects3d\iCount]\fScaleY,aObjects3d\lMesh[a
Objects3d\iCount]\fScaleZ
            PositionEntity
            aObjects3d\lMesh[aObjects3d\iCount]\iMesh,aObjects3d\lMesh[aObjects3d\iCount]\fX,aObjects3d\lMesh[aObjects3d\iC
ount]\fY,aObjects3d\lMesh[aObjects3d\iCount]\fZ
            RotateEntity
            aObjects3d\lMesh[aObjects3d\iCount]\iMesh,aObjects3d\lMesh[aObjects3d\iCount]\fRotationX,aObjects3d\lMesh[aObje
cts3d\iCount]\fRotationY,aObjects3d\lMesh[aObjects3d\iCount]\fRotationZ
            EntityPickMode aObjects3d\lMesh[aObjects3d\iCount]\iMesh,2
            Else If ( aObjects3d\lMesh[aObjects3d\iCount]\iType = CYLINDER) Then
                aObjects3d\lMesh[aObjects3d\iCount]\iMesh=CreateCylinder();
                ScaleEntity                                aObjects3d\lMesh[aObjects3d\iCount]\iMesh,
                aObjects3d\lMesh[aObjects3d\iCount]\fScaleX,aObjects3d\lMesh[aObjects3d\iCount]\fScaleY,aObjects3d\lMesh[a
Objects3d\iCount]\fScaleZ
                PositionEntity
                aObjects3d\lMesh[aObjects3d\iCount]\iMesh,aObjects3d\lMesh[aObjects3d\iCount]\fX,aObjects3d\lMesh[aObjects3d\iC
ount]\fY,aObjects3d\lMesh[aObjects3d\iCount]\fZ
                RotateEntity
                aObjects3d\lMesh[aObjects3d\iCount]\iMesh,aObjects3d\lMesh[aObjects3d\iCount]\fRotationX,aObjects3d\lMesh[aObje
cts3d\iCount]\fRotationY,aObjects3d\lMesh[aObjects3d\iCount]\fRotationZ
                EntityPickMode aObjects3d\lMesh[aObjects3d\iCount]\iMesh,2
                Else If ( aObjects3d\lMesh[aObjects3d\iCount]\iType = START) Then
                    aObjects3d\lMesh[aObjects3d\iCount]\iMesh=CreateCone();
                    ScaleEntity                                aObjects3d\lMesh[aObjects3d\iCount]\iMesh,
                    aObjects3d\lMesh[aObjects3d\iCount]\fScaleX,aObjects3d\lMesh[aObjects3d\iCount]\fScaleY,aObjects3d\lMesh[a
Objects3d\iCount]\fScaleZ
                    PositionEntity
                    aObjects3d\lMesh[aObjects3d\iCount]\iMesh,aObjects3d\lMesh[aObjects3d\iCount]\fX,aObjects3d\lMesh[aObjects3d\iC
ount]\fY,aObjects3d\lMesh[aObjects3d\iCount]\fZ
                    RotateEntity
                    aObjects3d\lMesh[aObjects3d\iCount]\iMesh,aObjects3d\lMesh[aObjects3d\iCount]\fRotationX,aObjects3d\lMesh[aObje
cts3d\iCount]\fRotationY,aObjects3d\lMesh[aObjects3d\iCount]\fRotationZ
                    EntityPickMode aObjects3d\lMesh[aObjects3d\iCount]\iMesh,2
                    EntityColor aObjects3d\lMesh[aObjects3d\iCount]\iMesh,255,0,0
                End If
                aObjects3d\iCount=aObjects3d\iCount+1
            Wend
        CloseFile( sFileOut)

```

End Function

```
;-----  
Function SaveTemp(sNameFile$="world_temp.amv")  
; check here for sound buttons what si going on: where are the elements stored  
sFileOut=WriteFile(sNameFile)  
verify if there is a mesh loaded, in case of yes, put it into the file, name and position  
;if there is a mesh, then I start by writing an 1; otherwise a 0  
If MeshLoaded Then  
WriteInt(sFileOut, 1)  
WriteString( sFileOut, NAME_WORLD$ )  
Else  
WriteInt(sFileOut,0)  
End If  
;END ADDNEW  
For iLoop=0 To aObjects3d\iCount-1  
If ( aObjects3d\vmesh[iLoop]\fX < DELETE_OBJECT) Then  
WriteInt ( sFileOut, aObjects3d\vmesh[iLoop]\iType)  
WriteString( sFileOut, aObjects3d\vmesh[iLoop]\cName$ )  
WriteString( sFileOut, aObjects3d\vmesh[iLoop]\cTextureName$ )  
WriteFloat ( sFileOut, aObjects3d\vmesh[iLoop]\fX#)  
WriteFloat ( sFileOut, aObjects3d\vmesh[iLoop]\fY#)  
WriteFloat ( sFileOut, aObjects3d\vmesh[iLoop]\fZ#)  
for sound I have only range stored in the fScaleX component  
If aObjects3d\vmesh[iLoop]\iType<>BUTTON_SOUND Then  
WriteFloat ( sFileOut, aObjects3d\vmesh[iLoop]\fRotationX#)  
WriteFloat ( sFileOut, aObjects3d\vmesh[iLoop]\fRotationY#)  
WriteFloat ( sFileOut, aObjects3d\vmesh[iLoop]\fRotationZ#)  
End If  
;I write the x scale factor; in case of sound this is the range  
WriteFloat ( sFileOut, aObjects3d\vmesh[iLoop]\fScaleX#)  
If aObjects3d\vmesh[iLoop]\iType<>BUTTON_SOUND Then  
WriteFloat ( sFileOut, aObjects3d\vmesh[iLoop]\fScaleY#)  
WriteFloat ( sFileOut, aObjects3d\vmesh[iLoop]\fScaleZ#)  
End If  
;END ADDNEW  
End If  
Next  
CloseFile( sFileOut)  
End Function
```

```
;-----  
Function Save(sNameFile$="world.amv")  
  
sFileOut=WriteFile(sNameFile)  
sLine$="Museo"+" "+"Editor"+" "+CurrentDate()  
WriteLine( sFileOut,sLine)
```

```

        check if there is any mesh loaded, if yes the I need to store just the name of it
    If MeshLoaded Then
        WriteLine( sFileOut,NAME_WORLD$)
    End If
;END ADDNEW

sLine$=" "
For iLoop=0 To aObjects3d\iCount-1
    If ( aObjects3d\vmesh[iLoop]\fX < DELETE_OBJECT) Then
        If(aObjects3d\vmesh[iLoop]\iType = CUBE) Then
            iBackSlash = Instr(aObjects3d\vmesh[iLoop]\cTextureName, "\")
            iPoint = Len ( aObjects3d\vmesh[iLoop]\cTextureName)
            If( iPoint >0 ) Then
                sNamePicture$= Mid (aObjects3d\vmesh[iLoop]\cTextureName, iBackSlash+1,iPoint )
            Else
                sNamePicture="NONE.jpg"
            End If
            sLine= "MURO"+iLoop+",",
            sLine= sLine + aObjects3d\vmesh[iLoop]\fX + "," + aObjects3d\vmesh[iLoop]\fY + "," +
aObjects3d\vmesh[iLoop]\fZ + ","
            sLine= sLine + aObjects3d\vmesh[iLoop]\fRotationX + "," +
aObjects3d\vmesh[iLoop]\fRotationY+", "+ aObjects3d\vmesh[iLoop]\fRotationZ + ","
            sLine= sLine + aObjects3d\vmesh[iLoop]\fScaleX+", "+ aObjects3d\vmesh[iLoop]\fScaleY+", "+
aObjects3d\vmesh[iLoop]\fScaleZ+", "
            sLine= sLine + sNamePicture
            WriteLine( sFileOut,sLine)
            Exit
        End If
    End If
Next

sLine$=" "
vbDone=0
For iLoop=0 To aObjects3d\iCount-1
    If ( aObjects3d\vmesh[iLoop]\fX < DELETE_OBJECT) Then
        If(aObjects3d\vmesh[iLoop]\iType = START) Then
            sLine= "INICIO,"
            sLine= sLine + aObjects3d\vmesh[iLoop]\fX + "," + aObjects3d\vmesh[iLoop]\fY + "," +
aObjects3d\vmesh[iLoop]\fZ + ","
            sLine= sLine + aObjects3d\vmesh[iLoop]\fRotationX + "," +
aObjects3d\vmesh[iLoop]\fRotationY+", "+ aObjects3d\vmesh[iLoop]\fRotationZ
WriteLine(
sFileOut,sLine)
            vbDone=1
            WriteLine( sFileOut,sLine)
        End If
    End If
Next
If vbDone=0 Then
    sLine= "INICIO,"

```

```

sLine= sLine + "0,0,0,0,0,0"
WriteLine( sFileOut,sLine)
End If

sLine$=""
For iLoop=0 To aObjects3d\iCount-1
    If ( aObjects3d\vmesh[iLoop]\fx < DELETE_OBJECT) Then
        If(aObjects3d\vmesh[iLoop]\iType = BUTTON_PICTURE) Then
            iBackSlash = Instr(aObjects3d\vmesh[iLoop]\cName, "\")
            iPoint = Instr( aObjects3d\vmesh[iLoop]\cName, ".")
            sNamePicture$= Mid (aObjects3d\vmesh[iLoop]\cName, iBackSlash+1,iPoint -3 )
            sLine= sNamePicture+", "
            sLine= sLine + aObjects3d\vmesh[iLoop]\fx + "," + aObjects3d\vmesh[iLoop]\fy + "," +
aObjects3d\vmesh[iLoop]\fz + ","
            sLine= sLine + aObjects3d\vmesh[iLoop]\fRotationX + "," +
aObjects3d\vmesh[iLoop]\fRotationY+ "," + aObjects3d\vmesh[iLoop]\fRotationZ + ","
            sLine= sLine + aObjects3d\vmesh[iLoop]\fScaleX+ "," + aObjects3d\vmesh[iLoop]\fScaleY+ "," +
aObjects3d\vmesh[iLoop]\fScaleZ
            WriteLine( sFileOut,sLine)
        End If

    End If

Next
sLine$=""
For iLoop=0 To aObjects3d\iCount-1
    If ( aObjects3d\vmesh[iLoop]\fx < DELETE_OBJECT) Then
        If(aObjects3d\vmesh[iLoop]\iType = BUTTON_FILM_STRIP) Then
            iBackSlash = Instr(aObjects3d\vmesh[iLoop]\cName, "\")
            iPoint = Instr( aObjects3d\vmesh[iLoop]\cName, ".")
            sNamePicture$= Mid (aObjects3d\vmesh[iLoop]\cName, iBackSlash+1,iPoint -3 )
            sLine= sNamePicture+", "
            sLine= sLine + aObjects3d\vmesh[iLoop]\fx + "," + aObjects3d\vmesh[iLoop]\fy + "," +
aObjects3d\vmesh[iLoop]\fz + ","
            sLine= sLine + aObjects3d\vmesh[iLoop]\fRotationX + "," +
aObjects3d\vmesh[iLoop]\fRotationY+ "," + aObjects3d\vmesh[iLoop]\fRotationZ + ","
            sLine= sLine + aObjects3d\vmesh[iLoop]\fScaleX+ "," + aObjects3d\vmesh[iLoop]\fScaleY+ "," +
aObjects3d\vmesh[iLoop]\fScaleZ
            WriteLine( sFileOut,sLine)
        End If

    End If

Next
sLine$=""
format: Audio1,1.5,4,65,0.5,audio/exterior7.wav
For iLoop=0 To aObjects3d\iCount-1
    If ( aObjects3d\vmesh[iLoop]\fx < DELETE_OBJECT) Then
        If(aObjects3d\vmesh[iLoop]\iType = BUTTON_SOUND) Then
            sLine= "Audio"+iLoop+", "
            ; fx, fy, fz

```

```

                sLine= sLine + aObjects3d\lMesh[iLoop]\fX + "," + aObjects3d\lMesh[iLoop]\fY + "," +
aObjects3d\lMesh[iLoop]\fZ + ","
                ;range: radius of the sphere
                sLine=sLine+aObjects3d\lMesh[iLoop]\fScaleX
                ;Has to be seen if the audio file is stored with path or not
                sNamePicture$= aObjects3d\lMesh[iLoop]\cName
                sLine= sLine+", " + sNamePicture
                WriteLine( sFileOut,sLine)
            End If
        End If
    Next

    CloseFile (sFileOut)
End Function
;-----
Function StartSounds()
    For iLoop=0 To aObjects3d\iCount-1
        If aObjects3d\lMesh[iLoop]\iType =BUTTON_SOUND Then
            EntityAlpha aObjects3d\lMesh[iLoop]\iMesh,0
            channel=
EmitSound(Load3DSound("3/"+aObjects3d\lMesh[iLoop]\cName),aObjects3d\lMesh[iLoop]\iMesh)
            End If
        Next
    End Function

;-----
Function EndSounds()
;BEGIN ADDNEW
    StopChannel channel
;restore opacity of the entity
    For iLoop=0 To aObjects3d\iCount-1
        If aObjects3d\lMesh[iLoop]\iType =BUTTON_SOUND Then
            EntityAlpha aObjects3d\lMesh[iLoop]\iMesh,0.35
        End If
    Next
End Function

;-----

Function SetPrimitive3d.Object3d(aArrayObject3d.ArrayObject3d,iTypeObject$,iX#=0.0,iY#=0.0,iZ#=0.0)

    iTypeObject=Upper(iTypeObject)
    If ( Instr (iTypeObject,"CUBE")) Then
        iTypeObject = "CUBE"
        aArrayObject3d\lMesh[aArrayObject3d\iCount] = New Object3d;
        aArrayObject3d\lMesh[aArrayObject3d\iCount]\iMesh=CreateCube();
        aArrayObject3d\lMesh[aArrayObject3d\iCount]\iType = CUBE
    Else If ( Instr (iTypeObject,"SPHERE")) Then

```

```

        iTypeObject = "SPHERE"
        aArrayObject3d\lMesh[aArrayObject3d\iCount] = New Object3d;
        aArrayObject3d\lMesh[aArrayObject3d\iCount]\iMesh=CreateSphere();
        aArrayObject3d\lMesh[aArrayObject3d\iCount]\iType = SPHERE
    Else If ( Instr (iTypeObject,"CYLINDER")) Then
        iTypeObject = "CYLINDER"
        aArrayObject3d\lMesh[aArrayObject3d\iCount] = New Object3d;
        aArrayObject3d\lMesh[aArrayObject3d\iCount]\iMesh=CreateCylinder();
        aArrayObject3d\lMesh[aArrayObject3d\iCount]\iType = CYLINDER
    Else If ( Instr (iTypeObject,"CONE")) Then
        iTypeObject = "CONE"
        aArrayObject3d\lMesh[aArrayObject3d\iCount] = New Object3d;
        aArrayObject3d\lMesh[aArrayObject3d\iCount]\iMesh=CreateCone();
        aArrayObject3d\lMesh[aArrayObject3d\iCount]\iType = CONE
    Else If ( Instr (iTypeObject,"START")) Then
        iTypeObject = "START"
        aArrayObject3d\lMesh[aArrayObject3d\iCount] = New Object3d;
        aArrayObject3d\lMesh[aArrayObject3d\iCount]\iMesh=CreateCone();
        EntityColor aArrayObject3d\lMesh[aArrayObject3d\iCount]\iMesh,255,0,0
        aArrayObject3d\lMesh[aArrayObject3d\iCount]\iType = START
    Else
        Return Null
    End If

    aArrayObject3d\lMesh[aArrayObject3d\iCount]\cName = iTypeObject + "_" + aArrayObject3d\iCount
    aArrayObject3d\lMesh[aArrayObject3d\iCount]\fX#=iX
    aArrayObject3d\lMesh[aArrayObject3d\iCount]\fY#=iY
    aArrayObject3d\lMesh[aArrayObject3d\iCount]\fZ#=iZ
    aArrayObject3d\lMesh[aArrayObject3d\iCount]\fScaleX= 10
    aArrayObject3d\lMesh[aArrayObject3d\iCount]\fScaleY= 10
    aArrayObject3d\lMesh[aArrayObject3d\iCount]\fScaleZ= 10
    aArrayObject3d\lMesh[aArrayObject3d\iCount]\fRotationX=0
    aArrayObject3d\lMesh[aArrayObject3d\iCount]\fRotationY=0
    aArrayObject3d\lMesh[aArrayObject3d\iCount]\fRotationZ=0

    ScaleEntity aArrayObject3d\lMesh[aArrayObject3d\iCount]\iMesh,10,10,10

    PositionEntity aArrayObject3d\lMesh[aArrayObject3d\iCount]\iMesh,iX,iY,iZ
    EntityPickMode aArrayObject3d\lMesh[aArrayObject3d\iCount]\iMesh,2
    aArrayObject3d\iCount=aArrayObject3d\iCount+1
    Return aArrayObject3d\lMesh[aArrayObject3d\iCount-1]
End Function

;-----
Function SetSound.Object3d(aArrayObject3d.ArrayObject3d,aButtonSelected.Button,iX#=0.0,iY#=0.0,iZ#=0.0)
    aArrayObject3d\lMesh[aArrayObject3d\iCount] = New Object3d;
    aArrayObject3d\lMesh[aArrayObject3d\iCount]\iMesh=CreateSphere();

```

```

aArrayObject3d\lMesh[aArrayObject3d\iCount]\fX#=#iX
aArrayObject3d\lMesh[aArrayObject3d\iCount]\fY#=#iY
aArrayObject3d\lMesh[aArrayObject3d\iCount]\fZ#=#iZ
aArrayObject3d\lMesh[aArrayObject3d\iCount]\iType = aButtonSelected\iTypeButton
aArrayObject3d\lMesh[aArrayObject3d\iCount]\cTextureName=aButtonSelected\cName
iBackSlash = Instr(aButtonSelected\cName, "\")
iPoint = Instr(aButtonSelected\cName, ".")
aArrayObject3d\lMesh[aArrayObject3d\iCount]\cName = Mid(aButtonSelected\cName, iBackSlash+1,iPoint -3
)+".mp3"
r# = 1.0
iTexture2=LoadTexture(aButtonSelected\cName)
;fScaleX stores the range
aArrayObject3d\lMesh[aArrayObject3d\iCount]\fScaleX=r
aArrayObject3d\lMesh[aArrayObject3d\iCount]\fScaleY=r
aArrayObject3d\lMesh[aArrayObject3d\iCount]\fScaleZ=r
aArrayObject3d\lMesh[aArrayObject3d\iCount]\fRotationX=0
aArrayObject3d\lMesh[aArrayObject3d\iCount]\fRotationY=0
aArrayObject3d\lMesh[aArrayObject3d\iCount]\fRotationZ=0
EntityAlpha aArrayObject3d\lMesh[aArrayObject3d\iCount]\iMesh,0.35
EntityTexture aArrayObject3d\lMesh[aArrayObject3d\iCount]\iMesh,iTexture2
PositionEntity aArrayObject3d\lMesh[aArrayObject3d\iCount]\iMesh,iX,iY,iZ
EntityPickMode aArrayObject3d\lMesh[aArrayObject3d\iCount]\iMesh,2
aArrayObject3d\iCount=aArrayObject3d\iCount+1

Return aArrayObject3d\lMesh[aArrayObject3d\iCount-1]
End Function
;-----
Function
SetFilmStrip.Object3d(aArrayObject3d.ArrayObject3d,aButtonSelected.Button,iX#=0.0,iY#=0.0,iZ#=0.0,sDirectory$="Film
Strip")
aArrayObject3d\lMesh[aArrayObject3d\iCount] = New Object3d;
aArrayObject3d\lMesh[aArrayObject3d\iCount]\iMesh=CreateCube();
aArrayObject3d\lMesh[aArrayObject3d\iCount]\cName=aButtonSelected.Button\cName+"_"+aButtonSelected.Butto
n\lImagenInfo\Width+"X"+aButtonSelected.Button\lImagenInfo\Height
aArrayObject3d\lMesh[aArrayObject3d\iCount]\cTextureName=aButtonSelected\cName
aArrayObject3d\lMesh[aArrayObject3d\iCount]\fX#=#iX
aArrayObject3d\lMesh[aArrayObject3d\iCount]\fY#=#iY
aArrayObject3d\lMesh[aArrayObject3d\iCount]\fZ#=#iZ
aArrayObject3d\lMesh[aArrayObject3d\iCount]\iType = aButtonSelected\iTypeButton

aArrayObject3d\lMesh[aArrayObject3d\iCount]\fScaleX=aButtonSelected\lImagenInfo\Width / 10
aArrayObject3d\lMesh[aArrayObject3d\iCount]\fScaleY=aButtonSelected\lImagenInfo\Height / 10
aArrayObject3d\lMesh[aArrayObject3d\iCount]\fScaleZ=0.01

aArrayObject3d\lMesh[aArrayObject3d\iCount]\fRotationX=0
aArrayObject3d\lMesh[aArrayObject3d\iCount]\fRotationY=0
aArrayObject3d\lMesh[aArrayObject3d\iCount]\fRotationZ=0

```

```

ScaleEntity                                aArrayObject3d\lMesh[aArrayObject3d\iCount]\iMesh,
aArrayObject3d\lMesh[aArrayObject3d\iCount]\fScaleX,
aArrayObject3d\lMesh[aArrayObject3d\iCount]\fScaleY,aArrayObject3d\lMesh[aArrayObject3d\iCount]\fScaleZ

sNameFileTexture$=sDirectory+ aButtonSelected\cName

; aButtonSelected\iTexture2=LoadAnimTexture(sNameFileTexture,1,aButtonSelected\iImagenInfo\Width,aButtonSel
ected\iImagenInfo\Height,0,35 )
;
; EntityTexture aArrayObject3d\lMesh[aArrayObject3d\iCount]\iMesh,aButtonSelected\iTexture2
PositionEntity aArrayObject3d\lMesh[aArrayObject3d\iCount]\iMesh,iX,iY,iZ+20
EntityPickMode aArrayObject3d\lMesh[aArrayObject3d\iCount]\iMesh,2

aArrayObject3d\iCount=aArrayObject3d\iCount+1
Return aArrayObject3d\lMesh[aArrayObject3d\iCount-1]

End Function
;-----Function
SetPicture3d.Object3d(aArrayObject3d.ArrayObject3d,aButtonSelected.Button,iX#=0.0,iY#=0.0,iZ#=0.0)
aArrayObject3d\lMesh[aArrayObject3d\iCount] = New Object3d;
aArrayObject3d\lMesh[aArrayObject3d\iCount]\iMesh=CreateCube();
aArrayObject3d\lMesh[aArrayObject3d\iCount]\cName = aButtonSelected\cName
aArrayObject3d\lMesh[aArrayObject3d\iCount]\cTextureName = aButtonSelected\cName
aArrayObject3d\lMesh[aArrayObject3d\iCount]\fX#=iX
aArrayObject3d\lMesh[aArrayObject3d\iCount]\fY#=iY
aArrayObject3d\lMesh[aArrayObject3d\iCount]\fZ#=iZ
aArrayObject3d\lMesh[aArrayObject3d\iCount]\iType = aButtonSelected\iTypeButton
aArrayObject3d\lMesh[aArrayObject3d\iCount]\fScaleX=aButtonSelected\iImagenInfo\Width / 10
aArrayObject3d\lMesh[aArrayObject3d\iCount]\fScaleY=aButtonSelected\iImagenInfo\Height / 10
aArrayObject3d\lMesh[aArrayObject3d\iCount]\fScaleZ=0.01
aArrayObject3d\lMesh[aArrayObject3d\iCount]\fRotationX=0
aArrayObject3d\lMesh[aArrayObject3d\iCount]\fRotationY=0
aArrayObject3d\lMesh[aArrayObject3d\iCount]\fRotationZ=0

ScaleEntity                                aArrayObject3d\lMesh[aArrayObject3d\iCount]\iMesh,
aArrayObject3d\lMesh[aArrayObject3d\iCount]\fScaleX,
aArrayObject3d\lMesh[aArrayObject3d\iCount]\fScaleY,aArrayObject3d\lMesh[aArrayObject3d\iCount]\fScaleZ

iTexture2=LoadTexture(aButtonSelected\cName)
If( iTexture2 = 0 ) Then
    RuntimeError    "Error al cargar la textura, posible driver incompatible"
End
End If

EntityTexture aArrayObject3d\lMesh[aArrayObject3d\iCount]\iMesh,iTexture2
PositionEntity aArrayObject3d\lMesh[aArrayObject3d\iCount]\iMesh,iX,iY,iZ+20
EntityPickMode aArrayObject3d\lMesh[aArrayObject3d\iCount]\iMesh,2
aArrayObject3d\iCount=aArrayObject3d\iCount+1
Return aArrayObject3d\lMesh[aArrayObject3d\iCount-1]

```

```

End Function

;--NOMBRE/DEFUNICION: INIT WINDOW / INICIALIZA LA EL CONTEXTO GRAFICO--
;-----PARAMETROS: Alto, ancho, profundidad de color de la pantalla-----

Function InitWindow ( iWindowHeight, iWindowWeight, iColorDepht);
    Graphics3D iWindowHeight,iWindowWeight,16,1,iColorDepht, 1 ,,,,,;FULL SCREEN1
    MoveMouse iWindowHeight/2,iWindowWeight/2
    InitCameras()
    Listener= CreateListener (tCameras\cam4,0.05,1,100)
End Function

;----- Global MeshTemp ;-----
Function InitComponents3d()
    Light=CreateLight()
End Function

;-----Loop Principal-----
    InitWindow (800,600,32)
    InitComponentsGui()
    InitComponents3d();
    tButtonSelected.Button = New Button
; debug=KEY_R+"      "
    While (Not EventsKeyboard() = EXIT_PROGRAM)
        EventsMouse()
        tButtonSelected=CheckButton( tRightGuiBar, BUTTONGUI)
        GuiComportamientos(tButtonSelected)
        tButtonSelected = Null
        If (tDirFolder1\iActive = True) Then
            tButtonSelected=CheckButton(tDirFolder1, BUTTON_PICTURE)
            If( Not tButtonSelected = Null) Then
                SetPicture3d(aObjects3d,tButtonSelected,EntityX#{tCameras\cam4)-
1,EntityY(tCameras\cam4)-1,EntityZ(tCameras\cam4)+5)
                Delay(300)
            End If
        Else If (tDirFolder2\iActive = True) Then
            tButtonSelected=CheckButton(tDirFolder2, BUTTON_FILM_STRIP)
            If( Not tButtonSelected = Null) Then
                SetPicture3d(aObjects3d,tButtonSelected,EntityX#{tCameras\cam4)-
1,EntityY(tCameras\cam4)-1,EntityZ(tCameras\cam4)+5)
                Delay(300)
            End If
        Else If (tDirFolder3\iActive = True) Then
            tButtonSelected=CheckButton(tDirFolder3, BUTTON_SOUND)
            If( Not tButtonSelected = Null) Then
                SetSound(aObjects3d,tButtonSelected,EntityX#{tCameras\cam4)-
1,EntityY(tCameras\cam4)-1,EntityZ(tCameras\cam4)+5)
                Delay(300)
            End If
        Else If (tDirFolder4\iActive = True) Then

```

```

tButtonSelected=CheckButton(tDirFolder4, BUTTON_PRIMITIVE)
If( Not tButtonSelected = Null) Then
    SetPrimitive3d.Object3d(aObjects3d,tButtonSelected\cName,EntityX#(tCameras\cam4)-
1,EntityY(tCameras\cam4)-1,EntityZ(tCameras\cam4)+5)
    Delay(300)
End If
Else If (tDirFolder5\iActive = True) Then
    tButtonSelected=CheckButton(tDirFolder5, BUTTON_TEX_COL)
    If( Not tButtonSelected = Null) Then
        SetTextureToPrimitive(tButtonSelected\cName)
        Delay(300)
    End If
part to allow loading a mesh
Else If (tDirFolder6\iActive = True) Then
    tButtonSelected=CheckButton(tDirFolder6, BUTTON_MESH)
    If( Not tButtonSelected = Null) Then
;if I already have a mesh I do not allow overload
If MeshLoaded = False Then
    iPoint = Instr(tButtonSelected\cName, ".")
    MeshName$= Mid (tButtonSelected\cName, 1,iPoint-1)+".3DS"
    NAME_WORLD$=MeshName
    MeshTemp=LoadMesh(NAME_WORLD)
    EntityAlpha MeshTemp,0.35
    PositionEntity MeshTemp,SIZE_WORLD_X,SIZE_WORLD_Y,SIZE_WORLD_Z
    MeshLoaded=True
    Delay(300)
End If
End If
;END ADDNEW

End If
;=====
tButtonSelected=CheckButton( tDownGuiBar, BUTTONGUI)
If( Not tButtonSelected= Null) Then
    If(tButtonSelected\cNameEvent = "X_BUTTON") Then
        sActiveAxis$="X"
    Else If(tButtonSelected\cNameEvent = "Y_BUTTON") Then
        sActiveAxis$="Y"
    Else If(tButtonSelected\cNameEvent = "Z_BUTTON") Then
        sActiveAxis$="Z"
    Else If(tButtonSelected\cNameEvent = "SAVE_BUTTON") Then
        Save()
        SaveTemp()
    Else If(tButtonSelected\cNameEvent = "OPEN_BUTTON") Then
        Open()
    End If
    FunctionCamera(tButtonSelected\cNameEvent,sActiveAxis)
End If

```

```

        ActionCamera()

        UpdateWorld()
        RenderWorld()

;       Text 100,250,aObjects3d\iCount +Text1 +"   x="+MouseX()+"   y="+MouseY()
Text 100,530,debug$
;       If( Not tButtonSelected = Null) Then
;           Text 100,100, tButtonSelected\cNameEvent
;       End If
;-----

        DrawImageArray ( tImageGui );
        DrawGuiComponents( tRightGuiBar );
        DrawGuiComponents( tDownGuiBar );

        If ( tDirFolder1\iActive = True ) Then
            DrawGuiComponents(tDirFolder1 );
        Else If ( tDirFolder2\iActive = True ) Then
            DrawGuiComponents(tDirFolder2);
        Else If ( tDirFolder3\iActive = True ) Then
            DrawGuiComponents(tDirFolder3);
        Else If ( tDirFolder4\iActive = True ) Then
            DrawGuiComponents(tDirFolder4);
        Else If ( tDirFolder5\iActive = True ) Then
            DrawGuiComponents(tDirFolder5);

        Else If ( tDirFolder6\iActive = True ) Then
            DrawGuiComponents(tDirFolder6);

        End If

        DrawImage tImageGui\vlImage[2]\iTexture1,MouseX(),MouseY()
        Flip
    Wend
End
;-----
.componentes_gui
Function InitComponentsGui()

;barraDER=====
        MakeButton(tRightGuiBar,"Buttons/RightBar/flechupU.bmp" ,"Buttons/RightBar/flechupD.bmp"
        ,"UP_BUTTON",BUTTONGUI      ,740,65,20,20)
        MakeButton(tRightGuiBar,"Buttons/RightBar/flechdnU.bmp" ,"Buttons/RightBar/flechdnD.bmp"
        ,"DOWN_BUTTON",BUTTONGUI      ,740,480,20,20)

```

```

        MakeButton(tRightGuiBar,"Buttons/RightBar/flechlzU.bmp" ,"Buttons/RightBar/flechlzd.bmp"
        ,"ARROW_LEFT_BUTTON",BUTTONGUI ,710,570,10,10)
        MakeButton(tRightGuiBar,"Buttons/RightBar/flechdU.bmp" ,"Buttons/RightBar/flechdd.bmp"
        ,"ARROW_RIGHT_BUTTON",BUTTONGUI ,730,570,10,10)
        MakeButton(tRightGuiBar,"Buttons/RightBar/flechupU.bmp" ,"Buttons/RightBar/flechupD.bmp"
        ,"ARROW_UP_BUTTON",BUTTONGUI ,720,560,10,10)
        MakeButton(tRightGuiBar,"Buttons/RightBar/flechdnU.bmp" ,"Buttons/RightBar/flechdnD.bmp"
        ,"ARROW_DOWN_BUTTON",BUTTONGUI ,720,580,10,10)

;previo y home-----
        MakeButton(tRightGuiBar,"Buttons/RightBar/homeup.bmp" ,"Buttons/RightBar/homedn.bmp"
        ,"HOME_BUTTON",BUTTONGUI ,733,510,20,20)
        MakeButton(tRightGuiBar,"Buttons/RightBar/previoup.bmp" ,"Buttons/RightBar/previodn.bmp"
        ,"CAMERA_BUTTON",BUTTONGUI ,710,510,20,20)

;Salir y ayuda-----
        MakeButton(tRightGuiBar,"Buttons/RightBar/Salup.bmp" ,"Buttons/RightBar/Saldn.bmp"
        ,"EXIT_BUTTON",BUTTONGUI ,765,6,20,20)
        MakeButton(tRightGuiBar,"Buttons/RightBar/Ayuup.bmp" ,"Buttons/RightBar/Ayudn.bmp"
        ,"HELP_BUTTON",BUTTONGUI ,740,6,20,20)

;pesta-----
        MakeButton(tRightGuiBar,"Buttons/RightBar/pesta1dn.bmp" ,"Buttons/RightBar/pesta1up.bmp"
        ,"FOLDER1_BUTTON",BUTTONGUI ,700,34,20,25)
        MakeButton(tRightGuiBar,"Buttons/RightBar/pesta2up.bmp" ,"Buttons/RightBar/pesta2dn.bmp"
        ,"FOLDER2_BUTTON",BUTTONGUI ,720,34,20,25)
        MakeButton(tRightGuiBar,"Buttons/RightBar/pesta3up.bmp" ,"Buttons/RightBar/pesta3dn.bmp"
        ,"FOLDER3_BUTTON",BUTTONGUI ,740,34,20,25)
        MakeButton(tRightGuiBar,"Buttons/RightBar/pesta4up.bmp" ,"Buttons/RightBar/pesta4dn.bmp"
        ,"FOLDER4_BUTTON",BUTTONGUI ,760,34,20,25)
        MakeButton(tRightGuiBar,"Buttons/RightBar/pesta5up.bmp" ,"Buttons/RightBar/pesta5dn.bmp"
        ,"FOLDER5_BUTTON",BUTTONGUI ,780,34,20,25)
        MakeButton(tRightGuiBar,"Buttons/RightBar/pesta6up.bmp" ,"Buttons/RightBar/pesta6dn.bmp"
        ,"FOLDER6_BUTTON",BUTTONGUI ,680,34,20,25)

;Guarda y abre- En pleca superior-----

        MakeButton(tDownGuiBar,"Buttons/DownBar/guardaup.bmp"
        ,"Buttons/DownBar/guardadn.bmp","SAVE_BUTTON",BUTTONGUI ,33,6,20,20)
        MakeButton(tDownGuiBar,"Buttons/DownBar/abrirup.bmp"
        ,"Buttons/DownBar/abrirdn.bmp","OPEN_BUTTON",BUTTONGUI ,10,6,20,20)

; MakeButton(tDownGuiBar,"Buttons/DownBar/guardaup.bmp"
; ,"Buttons/DownBar/guardadn.bmp","SAVE_BUTTON",BUTTONGUI ,10,575,20,20)
; MakeButton(tDownGuiBar,"Buttons/DownBar/abrirup.bmp"
; ,"Buttons/DownBar/abrirdn.bmp","OPEN_BUTTON",BUTTONGUI ,30,575,20,20)

;barraBAJO=====

        MakeButton(tDownGuiBar,"Buttons/DownBar/basuraup.bmp"
        ,"Buttons/DownBar/basurad.bmp","TRASH_BUTTON",BUTTONGUI ,400,575,20,20)

        MakeButton(tDownGuiBar,"Buttons/DownBar/mueveup.bmp"
        ,"Buttons/DownBar/muevedn.bmp","MOVE_BUTTON",BUTTONGUI ,80,575,20,20)

```

```

        MakeButton(tDownGuiBar,"Buttons/DownBar/masup.bmp"
,"Buttons/DownBar/masdn.bmp","MOVE_+_BUTTON",BUTTONGUI ,100,575,10,10)
        MakeButton(tDownGuiBar,"Buttons/DownBar/menosup.bmp"
_BUTTON",BUTTONGUI ,100,585,10,10)

        MakeButton(tDownGuiBar,"Buttons/DownBar/rotaup.bmp"
,"Buttons/DownBar/rotadn.bmp","ROTATE_+_BUTTON",BUTTONGUI ,120,575,20,20)
        MakeButton(tDownGuiBar,"Buttons/DownBar/masup.bmp"
,"Buttons/DownBar/masdn.bmp","ROTATE_+_BUTTON",BUTTONGUI ,140,575,10,10)
        MakeButton(tDownGuiBar,"Buttons/DownBar/menosup.bmp"
_BUTTON",BUTTONGUI ,140,585,10,10)

        MakeButton(tDownGuiBar,"Buttons/DownBar/crecerup.bmp"
,"Buttons/DownBar/crecerdn.bmp","SCALE_+_BUTTON",BUTTONGUI ,160,575,20,20)
        MakeButton(tDownGuiBar,"Buttons/DownBar/masup.bmp"
,"Buttons/DownBar/masdn.bmp","SCALE_+_BUTTON",BUTTONGUI ,180,575,10,10)
        MakeButton(tDownGuiBar,"Buttons/DownBar/menosup.bmp"
_BUTTON",BUTTONGUI ,180,585,10,10)

        MakeButton(tDownGuiBar,"Buttons/DownBar/xup.bmp"
,"Buttons/DownBar/xdn.bmp","X_+_BUTTON",BUTTONGUI ,240,575,20,20)
        MakeButton(tDownGuiBar,"Buttons/DownBar/yup.bmp"
,"Buttons/DownBar/ydn.bmp","Y_+_BUTTON",BUTTONGUI ,260,575,20,20)
        MakeButton(tDownGuiBar,"Buttons/DownBar/zup.bmp"
,"Buttons/DownBar/zdn.bmp","Z_+_BUTTON",BUTTONGUI ,280,575,20,20)

;Plecas=====
        MakelImage(tlImageGui,"Buttons/ImageGui/plecasup.bmp" , -3,0,800,40)
        MakelImage(tlImageGui,"Buttons/ImageGui/plecader.bmp" , 700,33,100,570)

;cursor-----

        MakelImage(tlImageGui,"Buttons/ImageGui/cursor.bmp",700,34);cursor

        MakelImage(tlImageGui,"Buttons/ImageGui/plecainf.bmp",0,570,700,30); la pleca perdida Inferior

;ejes=====
        MakelImage(tlImageGui,"Buttons/ImageGui/guia3d2.png",0,250,30,30)
        MakelImage(tlImageGui,"Buttons/ImageGui/guia3d2.png",0,510,30,30)
        MakelImage(tlImageGui,"Buttons/ImageGui/guia3d2.png",350,250,30,30)

;lectura folders
        tDirFolder1= ReadFolders("1", iPosGuiBarX , iPosGuiBarY , ilconSize, BUTTON_PICTURE)
        tDirFolder2= ReadFolders("2", iPosGuiBarX , iPosGuiBarY , ilconSize, BUTTON_FILM_STRIP)
        tDirFolder3= ReadFolders("3", iPosGuiBarX , iPosGuiBarY , ilconSize, BUTTON_SOUND)
        tDirFolder4= ReadFolders("4", iPosGuiBarX , iPosGuiBarY , ilconSize, BUTTON_PRIMITIVE)
        tDirFolder5= ReadFolders("5", iPosGuiBarX , iPosGuiBarY , ilconSize, BUTTON_TEX_COL)

```

```
tDirFolder6= ReadFolders("arquitectura", iPosGuiBarX , iPosGuiBarY , iIconSize, BUTTON_MESH)
```

```
MaskImage tImageGui\Image[2]\iTexture1,11,1,55
```

```
End Function
```

```
;-----
```

```
Function GuiComportamientos(v_ButtonSelected.Button)
```

```
    If( Not v_ButtonSelected= Null) Then
```

```
        Select v_ButtonSelected\cNameEvent
```

```
            Case "DOWN_BUTTON"
```

```
                If (tDirFolder1\iActive= True) Then
```

```
                    UpFolder(tDirFolder1,iLimitMaxGuiBarY)
```

```
                    DrawIconFolder(tDirFolder1,iLimitMinGuiBarY ,iLimitMaxGuiBarY)
```

```
                Else If (tDirFolder2\iActive= True) Then
```

```
                    UpFolder(tDirFolder2,iLimitMaxGuiBarY)
```

```
                    DrawIconFolder(tDirFolder2,iLimitMinGuiBarY ,iLimitMaxGuiBarY)
```

```
                Else If (tDirFolder3\iActive= True) Then
```

```
                    UpFolder(tDirFolder3,iLimitMaxGuiBarY)
```

```
                    DrawIconFolder(tDirFolder3,iLimitMinGuiBarY ,iLimitMaxGuiBarY)
```

```
                Else If (tDirFolder4\iActive= True) Then
```

```
                    UpFolder(tDirFolder4,iLimitMaxGuiBarY)
```

```
                    DrawIconFolder(tDirFolder4,iLimitMinGuiBarY ,iLimitMaxGuiBarY)
```

```
                Else If (tDirFolder5\iActive= True) Then
```

```
                    UpFolder(tDirFolder5,iLimitMaxGuiBarY)
```

```
                    DrawIconFolder(tDirFolder5,iLimitMinGuiBarY ,iLimitMaxGuiBarY)
```

```
                Else If (tDirFolder6\iActive= True) Then
```

```
                    UpFolder(tDirFolder6,iLimitMaxGuiBarY)
```

```
                    DrawIconFolder(tDirFolder6,iLimitMinGuiBarY ,iLimitMaxGuiBarY)
```

```
                End If
```

```
            Case "UP_BUTTON"
```

```
                If (tDirFolder1\iActive= True) Then
```

```
                    DownFolder(tDirFolder1,iLimitMinGuiBarY )
```

```
                    DrawIconFolder(tDirFolder1,iLimitMinGuiBarY ,iLimitMaxGuiBarY)
```

```
                Else If (tDirFolder2\iActive= True) Then
```

```
                    DownFolder(tDirFolder2,iLimitMinGuiBarY )
```

```
                    DrawIconFolder(tDirFolder2,iLimitMinGuiBarY ,iLimitMaxGuiBarY)
```

```
                Else If (tDirFolder3\iActive= True) Then
```

```
                    DownFolder(tDirFolder3,iLimitMinGuiBarY )
```

```
                    DrawIconFolder(tDirFolder3,iLimitMinGuiBarY ,iLimitMaxGuiBarY)
```

```
                Else If (tDirFolder4\iActive= True) Then
```

```
                    DownFolder(tDirFolder4,iLimitMinGuiBarY )
```

```
                    DrawIconFolder(tDirFolder4,iLimitMinGuiBarY ,iLimitMaxGuiBarY)
```

```
                Else If (tDirFolder5\iActive= True) Then
```

```
                    DownFolder(tDirFolder5,iLimitMinGuiBarY )
```

```

        DrawIconFolder(tDirFolder5,iLimitMinGuiBarY ,iLimitMaxGuiBarY)

    Else If (tDirFolder6\iActive= True) Then
        DownFolder(tDirFolder6,iLimitMinGuiBarY )
        DrawIconFolder(tDirFolder6,iLimitMinGuiBarY ,iLimitMaxGuiBarY)

    End If

Case "EXIT_BUTTON"
    End
Case "FOLDER1_BUTTON"
    NoDrawIconFolder(tDirFolder2)
    NoDrawIconFolder(tDirFolder3)
    NoDrawIconFolder(tDirFolder4)
    NoDrawIconFolder(tDirFolder5)

    NoDrawIconFolder(tDirFolder6)

    DrawIconFolder(tDirFolder1,iLimitMinGuiBarY ,iLimitMaxGuiBarY)
    tDirFolder1\iActive=True
    tDirFolder2\iActive=False
    tDirFolder3\iActive=False
    tDirFolder4\iActive=False
    tDirFolder5\iActive=False

    tDirFolder6\iActive=False

Case "FOLDER2_BUTTON"
    NoDrawIconFolder(tDirFolder1)
    NoDrawIconFolder(tDirFolder3)
    NoDrawIconFolder(tDirFolder4)
    NoDrawIconFolder(tDirFolder5)

    NoDrawIconFolder(tDirFolder6)

    DrawIconFolder(tDirFolder2,iLimitMinGuiBarY ,iLimitMaxGuiBarY)
    tDirFolder1\iActive=False
    tDirFolder2\iActive=True
    tDirFolder3\iActive=False
    tDirFolder4\iActive=False
    tDirFolder5\iActive=False

    tDirFolder6\iActive=False

Case "FOLDER3_BUTTON"
    NoDrawIconFolder(tDirFolder1)
    NoDrawIconFolder(tDirFolder2)

```

NoDrawIconFolder(tDirFolder4)
NoDrawIconFolder(tDirFolder5)

NoDrawIconFolder(tDirFolder6)

DrawIconFolder(tDirFolder3,iLimitMinGuiBarY ,iLimitMaxGuiBarY)
tDirFolder1\iActive=False
tDirFolder2\iActive=False
tDirFolder3\iActive=True
tDirFolder4\iActive=False
tDirFolder5\iActive=False

tDirFolder6\iActive=False

Case "FOLDER4_BUTTON"

NoDrawIconFolder(tDirFolder1)
NoDrawIconFolder(tDirFolder2)
NoDrawIconFolder(tDirFolder3)
NoDrawIconFolder(tDirFolder5)

NoDrawIconFolder(tDirFolder6)

DrawIconFolder(tDirFolder4,iLimitMinGuiBarY ,iLimitMaxGuiBarY)
tDirFolder1\iActive=False
tDirFolder2\iActive=False
tDirFolder3\iActive=False
tDirFolder4\iActive=True
tDirFolder5\iActive=False

tDirFolder6\iActive=False

Case "FOLDER5_BUTTON"

NoDrawIconFolder(tDirFolder1)
NoDrawIconFolder(tDirFolder2)
NoDrawIconFolder(tDirFolder3)
NoDrawIconFolder(tDirFolder4)

NoDrawIconFolder(tDirFolder6)

DrawIconFolder(tDirFolder5,iLimitMinGuiBarY ,iLimitMaxGuiBarY)
tDirFolder1\iActive=False
tDirFolder2\iActive=False
tDirFolder3\iActive=False
tDirFolder4\iActive=False
tDirFolder5\iActive=True

tDirFolder6\iActive=False

```

Case "FOLDER6_BUTTON"
    NoDrawIconFolder(tDirFolder1)
    NoDrawIconFolder(tDirFolder2)
    NoDrawIconFolder(tDirFolder3)
    NoDrawIconFolder(tDirFolder4)
    NoDrawIconFolder(tDirFolder5)

    DrawIconFolder(tDirFolder6,iLimitMinGuiBarY ,iLimitMaxGuiBarY)

    tDirFolder1\iActive=False
    tDirFolder2\iActive=False
    tDirFolder3\iActive=False
    tDirFolder4\iActive=False
    tDirFolder5\iActive=False

    tDirFolder6\iActive=True

Case "CAMERA_BUTTON"
    If( tCameras\tDataCam\iStateCamera = FOUR_CAMERA ) Then
        SwitchFourCamera()
        EndSounds();
        tCameras\tDataCam\iStateCamera = ONE_CAMERA
        tImageGui\lImage[3]\iStateDraw=True
;-----
        tImageGui\lImage[4]\iStateDraw=True
;-----
        tImageGui\lImage[5]\iStateDraw=True
;-----
        if there is a mesh, then I make it a little bit opaque
            If MeshLoaded Then
                EntityAlpha MeshTemp,0.35
;-----
            End If
;END ADDNEW

        Else If ( tCameras\tDataCam\iStateCamera = ONE_CAMERA ) Then
            SwitchOneCamera()
            search the inicio such to put camera4 there
            For iLoop=0 To aObjects3d\iCount-1
                If aObjects3d\lMesh[iLoop]\iType =START Then
                    PositionEntity
tCameras\cam4,aObjects3d\lMesh[iLoop]\fX,aObjects3d\lMesh[iLoop]\fY,aObjects3d\lMesh[iLoop]\fZ
                    RotateEntity
tCameras\cam4,aObjects3d\lMesh[iLoop]\fRotationX,aObjects3d\lMesh[iLoop]\fRotationY,aObjects3d\lMesh[iLoop]\fRotationZ
                End If
            Next
;END ADDNEW

```

```

        StartSounds()
        tImageGui\vlImage[3]\iStateDraw=False
;-----
        tImageGui\vlImage[4]\iStateDraw=False
;-----
        tImageGui\vlImage[5]\iStateDraw=False
;-----
if there is a mesh, then I make it a little bit opaque
    If MeshLoaded Then
        EntityAlpha MeshTemp,1;
    End If
;END ADDNEW
        tCameras\tDataCam\iStateCamera = FOUR_CAMERA
        End If
        Delay (250)
        Case "ARROW_LEFT_BUTTON"
            TurnEntity tCameras\cam4,0,2.0,0
        Case "ARROW_UP_BUTTON"
            MoveEntity tCameras\cam4,0,0,2
        Case "ARROW_DOWN_BUTTON"
            MoveEntity tCameras\cam4,0,0,-2
        Case "ARROW_RIGHT_BUTTON"
            TurnEntity tCameras\cam4,0,-2.0,0
        Case "HOME_BUTTON"
            ResetCamera()
        End Select
    End If
End Function
Type DataCameras
    Field iStateCamera
    Field cActiveCamera$
    Field iRelativeX
    Field iRelativeY
End Type
Type StateCamera
    Field tDataCam.DataCameras
    Field cam1
    Field cam2
    Field cam3
    Field cam4
End Type

Global tCameras.StateCamera = New StateCamera

;-----
;-----CONSTANTES-----

```

```

Const ilconSize = 50
Const iPosGuiBarY =95
Const iPosGuiBarX =725
Const iLimitMinGuiBarY =iPosGuiBarY
Const iLimitMaxGuiBarY =430
;-----
Global iInitWorkAreaX = 0;
Global iInitWorkAreaY = 33
Global iFinalWorkAreaX = 700
Global iFinalWorkAreaY = 570
;-----
Function InitCameras()
    tCameras\tDataCam.DataCameras = New DataCameras
    tCameras\tDataCam\iStateCamera = FOUR_CAMERA
    tCameras\cam1 = CreateCamera ()
    tCameras\cam2 = CreateCamera ()
    tCameras\cam3 = CreateCamera ()
    tCameras\cam4 = CreateCamera ()
    SwitchFourCamera()

    ResetCamera()
    CameraClsColor tCameras\cam1,180,180,180
    CameraClsColor tCameras\cam2,210,210,210
    CameraClsColor tCameras\cam3,210,210,210
    CameraClsColor tCameras\cam4,180,180,180

End Function
;-----
Function SwitchFourCamera()
    CameraClsColor tCameras\cam4,180,180,180
    CameraViewport tCameras\cam1,iInitWorkAreaX ,iInitWorkAreaY ,iFinalWorkAreaX /2,iFinalWorkAreaY/2-
iInitWorkAreaY/2
    CameraViewport tCameras\cam2,iFinalWorkAreaX/2,iInitWorkAreaY ,iFinalWorkAreaX/2,iFinalWorkAreaY/2-
iInitWorkAreaY/2
    CameraViewport tCameras\cam3,iInitWorkAreaX ,iFinalWorkAreaY/2+iInitWorkAreaY/2,iFinalWorkAreaX
/2,iFinalWorkAreaY/2-iInitWorkAreaY/2
    CameraViewport tCameras\cam4,iFinalWorkAreaX /2,iFinalWorkAreaY/2+iInitWorkAreaY/2,iFinalWorkAreaX
/2,iFinalWorkAreaY/2-iInitWorkAreaY/2

    CameraRange          tCameras\cam1,5,5000
    CameraRange          tCameras\cam2,5,5000
    CameraRange          tCameras\cam3,5,5000
    CameraRange          tCameras\cam4,5,5000
End Function
;-----

```

```

Function SwitchOneCamera()
    CameraClsColor tCameras\cam4,0,0,0

    CameraViewport tCameras\cam4,0,0,iFinalWorkAreaX,iFinalWorkAreaY
    CameraViewport tCameras\cam3,0,0,0,0
    CameraViewport tCameras\cam2,0,0,0,0
    CameraViewport tCameras\cam1,0,0,0,0
End Function
;-----
Function ResetCamera()
    PositionEntity tCameras\cam1,0,600,0
    RotateEntity tCameras\cam1,90,0,0
    PositionEntity tCameras\cam2,0,0,-140
    PositionEntity tCameras\cam3,600,0,0
    RotateEntity tCameras\cam3,0,90,00
        PositionEntity tCameras\cam4,0,40,0
    RotateEntity tCameras\cam4,0,0,0
End Function
;Const NAME_WORLD$ = "arquitectura/modelo_MV2.3DS"
Const SIZE_WORLD_X# = 0.1
Const SIZE_WORLD_Y# = -10.0
Const SIZE_WORLD_Z# = 10.0
;-----
Const SIZE_VECTOR=100
Const RUN =10000
Const EXITPROGRAM =10001
Const ESC=1
Const ARROW_UP = 200
Const ARROW_LEFT = 203
Const ARROW_DOWN = 208
Const ARROW_RIGHT = 205

Const KEY_R = 19
Const KEY_X = 45
Const KEY_Y = 44
Const KEY_Z = 21

Const KEY_N = 49
Const KEY_M = 50
Const KEY_D = 211
Const DELETE_OBJECT = 100000
Const MouseButton=302
Const MouseMiddleButton=301
Const MouseRightButton=300
Const MouseFree=303
Const BUTTONGUI=400
Const BUTTONICON=401

```

```

Const ONE_CAMERA =800
Const FOUR_CAMERA =801
Const APP_SET_OBJECT = 1000
Const APP_GET_OBJECT = 1001
Const APP_FREE      = 1002
Const OBJECT_FREE = 5000
Const OBJECT_SELECTED = 5001
;-----
Const BUTTON_PICTURE =2001
Const BUTTON_FILM_STRIP =2002
Const BUTTON_SOUND =2003
Const BUTTON_PRIMITIVE=2004
Const BUTTON_TEX_COL=2005
Const CUBE=7000
Const SPHERE=7001
Const CONE=7002
Const CYLINDER=7003
Const START=7004

```

```

Type Mouse
    Field iPosX
    Field iPosY
    Field iWheel
    Field ButtonPressed
End Type

```

```

;----
Global tMouse.Mouse =New Mouse

```

```

Function EventsMouse()
FlushMouse
    tMouse\iPosX = MouseX()
    tMouse\iPosY = MouseY()
    tMouse\iWheel= MouseZ()
    Delay 10
    tMouse\iWheel=tMouse\iWheel-MouseZ()
    iButton=GetMouse()
    If( MouseDown(1) ) Then
        tMouse\ButtonPressed = MouseLeftButton;
    Else If ( MouseDown(3) ) Then
        tMouse\ButtonPressed = MouseMiddleButton;
    Else If ( MouseDown(2) ) Then
        tMouse\ButtonPressed = MouseRightButton;
    Else
        tMouse\ButtonPressed=MouseFree;
    End If
End Function

```

;------

```
Function EventsKeyboard()  
  If ( KeyDown(ESC) )Then  
    Return EXITPROGRAM  
  Else If ( KeyDown(ARROW_UP) ) Then  
    Return ARROW_UP  
  Else If ( KeyDown(ARROW_LEFT) ) Then  
    Return ARROW_LEFT  
  Else If ( KeyDown(ARROW_DOWN) ) Then  
    Return ARROW_DOWN  
  Else If ( KeyDown(ARROW_RIGHT) ) Then  
    Return ARROW_RIGHT  
  Else If ( KeyDown(KEY_N) ) Then  
    Return KEY_N  
  Else If ( KeyDown(KEY_M) ) Then  
    Return KEY_M  
  Else If ( KeyDown(KEY_D) ) Then  
    Return KEY_D  
  Else If ( KeyDown(KEY_X) ) Then  
    Return KEY_X  
  Else If ( KeyDown(KEY_Y) ) Then  
    Return KEY_Y  
  Else If ( KeyDown(KEY_Z) ) Then  
    Return KEY_Z  
  Else If ( KeyDown(KEY_R) ) Then  
    Return KEY_R  
  End If  
  Return RUN  
End Function
```

```
Type tAreaSound  
  Field emitter%  
  Field listener%  
  Field sound%  
  Field emitVolume#  
  Field range#  
  Field channel%  
  Field volume#  
End Type
```

```
Function AreaSound_create(emitter%, listener%, sound%, range#=10.0, emitVolume#=10.0, loop%=False,  
play%=True)  
  Local s.tAreaSound = New tAreaSound  
  s.emitter = emitter  
  s.listener = listener  
  s.range = range
```

```

s\emitVolume = emitVolume
s\sound = sound%
If loop Then LoopSound(s\sound)
If play Then s\channel = PlaySound(s\sound)

AreaSound_calc(s)
If Not loop Then Delete s
Return Handle(s)
End Function

;call this method in your game loop, when using looped sounds
Function AreaSound_update()
For s.tAreaSound = Each tAreaSound
;AreaSound_calc(s)
AreaSound_play(s\sound)
Next
End Function

Function AreaSound_getCurrentVolume#(soundID%)
s.tAreaSound = Object.tAreaSound(soundID)
Return s\volume
End Function

Function AreaSound_setEmitVolume(soundID%, volume#)
s.tAreaSound = Object.tAreaSound(soundID)
s\emitVolume = volume
AreaSound_calc(s)
End Function

Function AreaSound_setEmitterRange(soundID%, range#)
s.tAreaSound = Object.tAreaSound(soundID)
s\range = range
AreaSound_calc(s)
End Function

Function AreaSound_play(soundID%, loop%=False)
s.tAreaSound = Object.tAreaSound(soundID)
If loop Then LoopSound(s\sound)
s\channel = PlaySound(s\sound)
AreaSound_calc(s)
End Function

Function AreaSound_stop(soundID%)
s.tAreaSound = Object.tAreaSound(soundID)
StopChannel(s\sound)
End Function

```

```

Function AreaSound_cleanUp()
    Delete Each tAreaSound
End Function

```

```

Function AreaSound_calc(s.tAreaSound)
    d# = EntityDistance(s\listener, s\emitter)
    If d < s\range Then
        s\volume = s\emitVolume - d * s\emitVolume / (s\range)
        ChannelVolume s\channel, s\volume
    Else
        s\volume = 0.0
        ChannelVolume s\channel, 0.0
    EndIf
End Function

```

```

Include "IO.bb"
Include "typeObject3d.bb"
Include "typeCamera.bb"

```

```

Global tObject3d.Object3d = New Object3d
tObject3d=NULL

```

```

;-----
Function GetCamera4(iInitAreaX,iInitAreaY,iFinalAreaX,iFinalAreaY)
    If (tMouse\iPosX <= iFinalAreaX /2) And (tMouse\iPosY<=(iFinalAreaY/2+iInitAreaY/2)) Then
        iX=tMouse\iPosX
        iY=tMouse\iPosY-iInitAreaY
        tCameras\tDataCam\cActiveCamera="CAMERA1"
        tCameras\tDataCam\iRelativeX=iX
        tCameras\tDataCam\iRelativeY=iY
    Else If tMouse\iPosX>iFinalAreaX /2 And tMouse\iPosY<=(iFinalAreaY/2+iInitAreaY/2) And (tMouse\iPosX) <
(iFinalAreaX) Then
        iX=tMouse\iPosX-iFinalAreaX /2
        iY=tMouse\iPosY-iInitAreaY
        tCameras\tDataCam\cActiveCamera="CAMERA2"
        tCameras\tDataCam\iRelativeX=iX
        tCameras\tDataCam\iRelativeY=iY
    Else If tMouse\iPosX<=iFinalAreaX /2 And (tMouse\iPosY>iFinalAreaY/2+iInitAreaY/2) And
tMouse\iPosY<iFinalAreaY Then
        iX=tMouse\iPosX
        iY=(tMouse\iPosY-iFinalAreaY/2)-iInitAreaY/2
        tCameras\tDataCam\cActiveCamera="CAMERA3"
        tCameras\tDataCam\iRelativeX=iX
        tCameras\tDataCam\iRelativeY=iY
    Else If ((tMouse\iPosX) > (iFinalAreaX/2)) And (tMouse\iPosY) > ((iFinalAreaY/2-iInitAreaY/2)) And (tMouse\iPosX)
< (iFinalAreaX) Then
        iX=tMouse\iPosX-iFinalAreaX /2
        iY=(tMouse\iPosY-iFinalAreaY/2 )-iInitAreaY/2
    EndIf
End Function

```

```

        tCameras\tDataCam\cActiveCamera="CAMERA4"
        tCameras\tDataCam\iRelativeX=iX
        tCameras\tDataCam\iRelativeY=iY
    Else
        tCameras\tDataCam\cActiveCamera="NO_CAMERA"
        tCameras\tDataCam\iRelativeX=-1
        tCameras\tDataCam\iRelativeY=-1
    End If
End Function

;-- -----

Function CheckCamera4.Object3d()
    GetCamera4(iInitWorkAreaX,iInitWorkAreaY,iFinalWorkAreaX,iFinalWorkAreaY)
    Select tCameras\tDataCam\cActiveCamera
        Case "CAMERA1"
            picked=CameraPick (tCameras\cam1, tCameras\tDataCam\iRelativeX , tCameras\tDataCam\iRelativeY)
        Case "CAMERA2"
            picked=CameraPick (tCameras\cam2, tCameras\tDataCam\iRelativeX, tCameras\tDataCam\iRelativeY)
        Case "CAMERA3"
            picked=CameraPick (tCameras\cam3, tCameras\tDataCam\iRelativeX, tCameras\tDataCam\iRelativeY)
        Case "CAMERA4"
            picked=CameraPick (tCameras\cam4, tCameras\tDataCam\iRelativeX, tCameras\tDataCam\iRelativeY)
    End Select

    For i=0 To aObjects3d\iCount-1
        aObjects3d\iMesh[i]\iState = OBJECT_FREE;
        EntityAlpha aObjects3d\iMesh[i]\iMesh,1
    Next

    If (Not picked = 0) Then
        For i=0 To aObjects3d\iCount
            If aObjects3d\iMesh[i]\iMesh=picked
                If( aObjects3d\iMesh[i]\iState = OBJECT_FREE)
                    EntityAlpha aObjects3d\iMesh[i]\iMesh,0.7
                    aObjects3d\iMesh[i]\iState = OBJECT_SELECTED;
                    Delay(100)
                    Return aObjects3d\iMesh[i]
                Else
                    EntityAlpha aObjects3d\iMesh[i]\iMesh,1
                    aObjects3d\iMesh[i]\iState = OBJECT_FREE;
                    Delay(100)
                    Return Null
                End If
            End If
        Next
    End If
End Function

```

```

Return Null
End Function

```

```

;-----

```

```

Function PanelMoveCamera(tCamera)
  If (tMouse\ButtonPressed = MouseMiddleButton)
    iTempX=tMouse\iPosX
    iTempY=tMouse\iPosY
    Delay 1
    EventsMouse()
    iTempX=iTempX-tmouse\iPosX
    iTempY=iTempY-tmouse\iPosY
    If( iTempX > 0) Then
      MoveEntity tCamera,-20,0,0
    Else If (iTempX< 0) Then
      MoveEntity tCamera,20,0,0
    Else If (iTempY > 0) Then
      MoveEntity tCamera,0,20,0
    Else If (iTempY < 0) Then
      MoveEntity tCamera,0,-20,0
    End If
  End If
  MoveEntity tCamera,0,0,tMouse\iWheel*30
End Function

```

```

;-----

```

```

Function ActionCamera()
  If tMouse\ButtonPressed = MouseRightButton Then
    tObject3d=CheckCamera4();
  End If

  GetCamera4(iInitWorkAreaX,iInitWorkAreaY,iFinalWorkAreaX,iFinalWorkAreaY)
  Select tCameras\tDataCam\cActiveCamera
  Case "CAMERA1"
    If (tMouse\ButtonPressed = MouseLeftButton ) And (Not tObject3d = Null) Then
      iTempX=tmouse\iPosX
      iTempY=tmouse\iPosY
      Delay 1
      EventsMouse()
      iTempX=iTempX-tmouse\iPosX
      iTempY=iTempY-tmouse\iPosY
      If( iTempX > 0) Then
        tObject3d\fx#= tObject3d\fx#-3
        PositionEntity tObject3d\iMesh,tObject3d\fx,tObject3d\fy,tObject3d\fz
      Else If (iTempX< 0) Then
        tObject3d\fx#= tObject3d\fx#+3
        PositionEntity tObject3d\iMesh,tObject3d\fx,tObject3d\fy,tObject3d\fz
      End If
    End If
  End Select
End Function

```

```

Else If (iTempY > 0) Then
    tObject3d\Z# = tObject3d\Z# + 3
    PositionEntity tObject3d\iMesh, tObject3d\X, tObject3d\Y, tObject3d\Z
Else If (iTempY < 0) Then
    tObject3d\Z# = tObject3d\Z# - 3
    PositionEntity tObject3d\iMesh, tObject3d\X, tObject3d\Y, tObject3d\Z
End If
End If

```

```

If (Not tObject3d = Null) Then
    Select EventsKeyboard()
    Case ARROW_LEFT
        tObject3d\X# = tObject3d\X# - 1
        PositionEntity tObject3d\iMesh, tObject3d\X, tObject3d\Y, tObject3d\Z
    Case ARROW_RIGHT
        tObject3d\X# = tObject3d\X# + 1
        PositionEntity tObject3d\iMesh, tObject3d\X, tObject3d\Y, tObject3d\Z
    Case ARROW_UP
        tObject3d\Z# = tObject3d\Z# + 1
        PositionEntity tObject3d\iMesh, tObject3d\X, tObject3d\Y, tObject3d\Z
    Case ARROW_DOWN
        tObject3d\Z# = tObject3d\Z# - 1
        PositionEntity tObject3d\iMesh, tObject3d\X, tObject3d\Y, tObject3d\Z
    End Select
End If

```

```

PanelMoveCamera(tCameras\cam1)

```

```

Case "CAMERA2"

```

```

If (tMouse\ButtonPressed = MouseLeftButton) And (Not tObject3d = Null) Then
    iTempX = tMouse\iPosX
    iTempY = tMouse\iPosY
    Delay 1
    EventsMouse()
    iTempX = iTempX - tMouse\iPosX
    iTempY = iTempY - tMouse\iPosY
    If (iTempX > 0) Then
        tObject3d\X# = tObject3d\X# - 3
        PositionEntity tObject3d\iMesh, tObject3d\X, tObject3d\Y, tObject3d\Z
    Else If (iTempX < 0) Then
        tObject3d\X# = tObject3d\X# + 3
        PositionEntity tObject3d\iMesh, tObject3d\X, tObject3d\Y, tObject3d\Z
    Else If (iTempY > 0) Then
        tObject3d\Y = tObject3d\Y + 3
        PositionEntity tObject3d\iMesh, tObject3d\X, tObject3d\Y, tObject3d\Z
    Else If (iTempY < 0) Then

```

```

        tObject3d\fy=    tObject3d\fy-3
        PositionEntity tObject3d\iMesh,tObject3d\fx,tObject3d\fy,tObject3d\fz
    End If
End If

If (Not tObject3d = Null ) Then
    Select EventsKeyboard()
    Case ARROW_LEFT
        tObject3d\fx#=    tObject3d\fx#-1
        PositionEntity tObject3d\iMesh,tObject3d\fx,tObject3d\fy,tObject3d\fz
    Case ARROW_RIGHT
        tObject3d\fx#=    tObject3d\fx#+1
        PositionEntity tObject3d\iMesh,tObject3d\fx,tObject3d\fy,tObject3d\fz
    Case ARROW_UP
        tObject3d\fy#=    tObject3d\fy#+1
        PositionEntity tObject3d\iMesh,tObject3d\fx,tObject3d\fy,tObject3d\fz
    Case ARROW_DOWN
        tObject3d\fy#=    tObject3d\fy#-1
        PositionEntity tObject3d\iMesh,tObject3d\fx,tObject3d\fy,tObject3d\fz
    End Select
End If

PanelMoveCamera(tCameras\cam2)

```

Case "CAMERA3"

```

If (tMouse\ButtonPressed = MouseLeftButton ) And (Not tObject3d= Null) Then
    iTempX=tmouse\iPosX
    iTempY=tmouse\iPosY
    Delay 1
    EventsMouse()
    iTempX=iTempX-tmouse\iPosX
    iTempY=iTempY-tmouse\iPosY
    If( iTempX > 0) Then
        tObject3d\fz#=    tObject3d\fz#-3
        PositionEntity tObject3d\iMesh,tObject3d\fx,tObject3d\fy,tObject3d\fz
    Else If (iTempX< 0) Then
        tObject3d\fz#=    tObject3d\fz#+3
        PositionEntity tObject3d\iMesh,tObject3d\fx,tObject3d\fy,tObject3d\fz
    Else If (iTempY > 0) Then
        tObject3d\fy#=    tObject3d\fy#+3
        PositionEntity tObject3d\iMesh,tObject3d\fx,tObject3d\fy,tObject3d\fz
    Else If (iTempY < 0) Then
        tObject3d\fy#=    tObject3d\fy#-3
        PositionEntity tObject3d\iMesh,tObject3d\fx,tObject3d\fy,tObject3d\fz
    End If
End If

```

```

If (Not tObject3d = Null ) Then
    Select EventsKeyboard()
    Case ARROW_LEFT
        tObject3d\Z# = tObject3d\Z#-1
        PositionEntity tObject3d\iMesh,tObject3d\X,tObject3d\Y,tObject3d\Z
    Case ARROW_RIGHT
        tObject3d\Z# = tObject3d\Z#+1
        PositionEntity tObject3d\iMesh,tObject3d\X,tObject3d\Y,tObject3d\Z
    Case ARROW_UP
        tObject3d\Y# = tObject3d\Y#+1
        PositionEntity tObject3d\iMesh,tObject3d\X,tObject3d\Y,tObject3d\Z
    Case ARROW_DOWN
        tObject3d\Y# = tObject3d\Y#-1
        PositionEntity tObject3d\iMesh,tObject3d\X,tObject3d\Y,tObject3d\Z
    End Select
End If

PanelMoveCamera(tCameras\cam3)
Case "CAMERA4"
    Select EventsKeyboard()
    Case ARROW_LEFT
        TurnEntity tCameras\cam4,0,2.0,0
    Case ARROW_RIGHT
        TurnEntity tCameras\cam4,0,-2.0,0
    Case ARROW_UP
        MoveEntity tCameras\cam4,0,0,5
    Case ARROW_DOWN
        MoveEntity tCameras\cam4,0,0,-5
    End Select
End Select

FunctionCamera()

End Function
;-----
Function SetTextureToPrimitive(sName$)
    If (Not tObject3d = Null ) Then
        If(tObject3d \iType = CUBE) Then
            If ( Len (sName) > 0) Then
                iTexture2=LoadTexture(sName)
                If( iTexture2 = 0 ) Then
                    RuntimeError "Error al cargar la textura, posible driver incompatible"
                End
            End If
            tObject3d\cTextureName$=sName
            EntityTexture tObject3d \iMesh,iTexture2
        End If
    End If
End Function

```

```

        End If
    End If
End If
End Function
;-----
Function FunctionCamera( sEvent$=" ",iAxis$=" ")

    If (Not tObject3d = Null ) Then
        iEvent = EventsKeyboard()
        debug=iAxis+ "-----HOLA MUNDO-----"+iEvent

        If ((sEvent= "ROTATE_+_BUTTON"))Then

            Select iAxis
            Case "X"
                tObject3d\RotationX = tObject3d\RotationX + 0.5
                RotateEntity tObject3d\iMesh
,tObject3d\RotationX,tObject3d\RotationY,tObject3d\RotationZ
            Case "Y"
                tObject3d\RotationY = tObject3d\RotationY + 0.5
                RotateEntity tObject3d\iMesh
,tObject3d\RotationX,tObject3d\RotationY,tObject3d\RotationZ
            Case "Z"
                tObject3d\RotationZ = tObject3d\RotationZ + 0.5
                RotateEntity tObject3d\iMesh
,tObject3d\RotationX,tObject3d\RotationY,tObject3d\RotationZ

            End Select
        Else If ((sEvent= "ROTATE_-_BUTTON"))Then
            Select iAxis
            Case "X"
                tObject3d\RotationX = tObject3d\RotationX - 0.5
                RotateEntity tObject3d\iMesh
,tObject3d\RotationX,tObject3d\RotationY,tObject3d\RotationZ
            Case "Y"
                tObject3d\RotationY = tObject3d\RotationY - 0.5
                RotateEntity tObject3d\iMesh
,tObject3d\RotationX,tObject3d\RotationY,tObject3d\RotationZ
            Case "Z"
                tObject3d\RotationZ = tObject3d\RotationZ - 0.5
                RotateEntity tObject3d\iMesh
,tObject3d\RotationX,tObject3d\RotationY,tObject3d\RotationZ

            End Select
        Else If ((sEvent= "SCALE_-_BUTTON"))Then
            Select iAxis
            Case "X"
                tObject3d\ScaleX=tObject3d\ScaleX-0.1
;BEGIN ADDNEW

```

```

        If tObject3d\iType= BUTTON_SOUND Then
            tObject3d\ScaleY=tObject3d\ScaleX
            tObject3d\ScaleZ=tObject3d\ScaleX
        End If
        ScaleEntity tObject3d\iMesh ,tObject3d\ScaleX,tObject3d\ScaleY,tObject3d\ScaleZ
;END ADDNEW

    Case "Y"
        tObject3d\ScaleY=tObject3d\ScaleY-0.1
        If tObject3d\iType= BUTTON_SOUND Then
            tObject3d\ScaleX=tObject3d\ScaleY
            tObject3d\ScaleZ=tObject3d\ScaleY
        End If
        ScaleEntity tObject3d\iMesh ,tObject3d\ScaleX,tObject3d\ScaleY,tObject3d\ScaleZ
    Case "Z"
        tObject3d\ScaleZ=tObject3d\ScaleZ-0.1
        If tObject3d\iType= BUTTON_SOUND Then
            tObject3d\ScaleX=tObject3d\ScaleZ
            tObject3d\ScaleY=tObject3d\ScaleZ
        End If
        ScaleEntity tObject3d\iMesh ,tObject3d\ScaleX,tObject3d\ScaleY,tObject3d\ScaleZ
    End Select
Else If ((sEvent= "SCALE+_BUTTON"))Then
    Select iAxis
    Case "X"
        tObject3d\ScaleX=tObject3d\ScaleX+0.1
;BEGIN ADDNEW
        If tObject3d\iType= BUTTON_SOUND Then
            tObject3d\ScaleY=tObject3d\ScaleX
            tObject3d\ScaleZ=tObject3d\ScaleX
        End If
        ScaleEntity tObject3d\iMesh ,tObject3d\ScaleX,tObject3d\ScaleY,tObject3d\ScaleZ
;END ADDNEW
    Case "Y"
        tObject3d\ScaleY=tObject3d\ScaleY+0.1
        If tObject3d\iType= BUTTON_SOUND Then
            tObject3d\ScaleX=tObject3d\ScaleY
            tObject3d\ScaleZ=tObject3d\ScaleY
        End If
        ScaleEntity tObject3d\iMesh ,tObject3d\ScaleX,tObject3d\ScaleY,tObject3d\ScaleZ
    Case "Z"
        tObject3d\ScaleZ=tObject3d\ScaleZ+0.1
        If tObject3d\iType= BUTTON_SOUND Then
            tObject3d\ScaleX=tObject3d\ScaleZ
            tObject3d\ScaleZ=tObject3d\ScaleZ
        End If
        ScaleEntity tObject3d\iMesh ,tObject3d\ScaleX,tObject3d\ScaleY,tObject3d\ScaleZ
    End Select

```

```

Else If ((sEvent= "MOVE_-_BUTTON"))Then
    Select iAxis
    Case "X"
        tObject3d\fx=tObject3d\fx - 1
        PositionEntity tObject3d\iMesh ,tObject3d\fx,tObject3d\fy,tObject3d\fz
    Case "Y"
        tObject3d\fy=tObject3d\fy - 1
        PositionEntity tObject3d\iMesh ,tObject3d\fx,tObject3d\fy,tObject3d\fz
    Case "Z"
        tObject3d\fz=tObject3d\fz - 1
        PositionEntity tObject3d\iMesh ,tObject3d\fx,tObject3d\fy,tObject3d\fz
    End Select
Else If ((sEvent= "MOVE_+_BUTTON"))Then
    Select iAxis
    Case "X"
        tObject3d\fx=tObject3d\fx + 1
        PositionEntity tObject3d\iMesh ,tObject3d\fx,tObject3d\fy,tObject3d\fz
    Case "Y"
        tObject3d\fy=tObject3d\fy + 1
        PositionEntity tObject3d\iMesh ,tObject3d\fx,tObject3d\fy,tObject3d\fz
    Case "Z"
        tObject3d\fz=tObject3d\fz + 1
        PositionEntity tObject3d\iMesh ,tObject3d\fx,tObject3d\fy,tObject3d\fz
    End Select

Else If ( iEvent = KEY_D ) Or (sEvent= "TRASH_BUTTON") Then
    tObject3d\fx=DELETE_OBJECT
    tObject3d\fy=DELETE_OBJECT
    tObject3d\fz=DELETE_OBJECT
    PositionEntity tObject3d\iMesh ,tObject3d\fx,tObject3d\fy,tObject3d\fz
Else If (iEvent = KEY_N) Or (sEvent = " ROTATE_BUTTON")Then

End If
End If
End Function

```

Include "ReadImageInfo.bb"

```

Type Image
    Field cName$
    Field iTexture1
    Field iInitPosX,iInitPosY,iFinalPosX,iFinalPosY
    Field tImageInfo.TImageInfo
    Field iStateDraw
End Type

```

```

Type ArrayImage
    Field iCounter
    Field vImage.Image[SIZE_VECTOR]
End Type
;-----
;-----

Function DrawImageArray(v_Button.ArrayImage)
    For iLoop =0 To v_Button\iCounter-1
        If (v_Button\vImage[iLoop]\iStateDraw = True) Then
            If (Not v_Button\vImage[iLoop]\iTexture1 = 0) Then
                DrawImage          v_Button\vImage[iLoop]\iTexture1,          v_Button\vImage[iLoop]\iInitPosX,
v_Button\vImage[iLoop]\iInitPosY
            End If
        End If
    Next
End Function
;-----
;-----
;-----
;-----revisa la memoria asignacion

Function MakeImage(vArrayImage.ArrayImage, cName$,iInitPosX, iInitPosY, fScaleX#=0.0,
fScaleY#=0.0,iStateDraw=True)
    If(Len (cName) = 0) Then
        Return "ERROR"
    End If
    iImageTemp=LoadImage(cName)
    If(iImageTemp = 0) Then
        Return "ERROR"
    End If

    iCounter=vArrayImage\iCounter
    vArrayImage\vImage[iCounter]=New Image
    vArrayImage\vImage[iCounter]\cName$=cName
    vArrayImage\vImage[iCounter]\iStateDraw=iStateDraw
    vArrayImage\vImage[iCounter]\iTexture1=iImageTemp
    vArrayImage\vImage[iCounter]\iImageInfo=ReadImageInfo(cName)
    vArrayImage\vImage[iCounter]\iInitPosX=iInitPosX
    vArrayImage\vImage[iCounter]\iInitPosY=iInitPosY
    If ( fScaleX = 0.0 ) Then
        fScaleX=vArrayImage\vImage[iCounter]\iImageInfo\Width;
    End If
    If ( fScaleY = 0.0 ) Then
        fScaleY=vArrayImage\vImage[iCounter]\iImageInfo\Height;
    End If
    vArrayImage\vImage[iCounter]\iFinalPosX=fScaleX+vArrayImage\vImage[iCounter]\iInitPosX
    vArrayImage\vImage[iCounter]\iFinalPosY=fScaleY+vArrayImage\vImage[iCounter]\iInitPosY
    ResizeImage vArrayImage\vImage[iCounter]\iTexture1,fScaleX,fScaleY

```

```

vArrayImage\iCounter = iCounter + 1
Return "OK"
End Function

Type TImageInfo
    Field Width
    Field Height
    Field Depth

    Field HasAlpha
End Type

Function ReadImageInfo.TImageInfo(imgfile$)

    imgfile$ = Trim(Lower(imgfile$))
    file = ReadFile(imgfile)
    If Not file Then Return Null

    Width = 0 : Height = 0 : Depth = 0 : HasAlpha = False

    If Instr(imgfile, ".png")

        ; Check signature
        Sig$ = Chr(137) + "PNG" + Chr(13) + Chr(10) + Chr(26) + Chr(10)

        If ReadByteString(file, Len(Sig)) <> Sig
            CloseFile(file)
            Return Null
        EndIf

        ; Check for valid IHDR chunk
        Size = Int_SwapEndian(ReadInt(file))
        Chunk$ = ReadByteString(file, 4)
        If Chunk <> "IHDR" Or Size <> 13
            CloseFile(file)
            Return Null
        EndIf

        ; Read IHDR chunk
        Width = Int_SwapEndian(ReadInt(file))
        Height = Int_SwapEndian(ReadInt(file))
        Depth = ReadByte(file)

        Col = ReadByte(file)

        Select Col
            Case 2

```

```

        Depth = Depth * 3 ; RGB (red + green + blue)

    Case 4
        Depth = Depth * 2 ; GA (gray + alpha)
        HasAlpha = True

    Case 5
        Depth = Depth * 4 ; ARGB (Alpha + RGB)
        HasAlpha = True
End Select

Else If Instr(imgfile, ".bmp")

    ; Check for bmp file
    If ReadByteString(file, 2) <> "BM"
        CloseFile(file)
        Return Null
    EndIf

    ; Some non-used stuff
    ReadInt(file) ; File size
    ReadShort(file) ; reserved1
    ReadShort(file) ; reserved2
    ReadInt(file) ; data offset from this point

    Size = ReadInt(file) ; BMPINFO chunk size
    If Size <> 40
        CloseFile(file)
        Return Null
    EndIf

    Width = ReadInt(file)
    Height = ReadInt(file)

    ReadShort(file) ; planes
    Depth = ReadInt(file)

Else If Instr(imgfile, ".jpg")

    ; SOI + APP0
    Sig$ = Chr($FF) + Chr($D8) + Chr($FF) + Chr($E0)

    If ReadByteString(file, Len(Sig)) <> Sig
        CloseFile(file)
        Return Null
    EndIf

```

```

    Size = Short_SwapEndian(ReadShort(file)) - 2
    Pos = FilePos(file)

    Ident$ = ReadByteString(file)
    Version = Short_SwapEndian(ReadShort(file))

    ; Check for JFIF version 1.2
    If (Ident <> "JFIF") Or ( (((Version And $FF00) Shr 8) <> 1) And ((Version And $FF) <> 2) )
        CloseFile(file)
        Return Null
    EndIf

    ; Search for markers
    SeekFile(file, Pos + Size)

    While Not Eof(file)
        If ReadByte(file) = $FF
            BType = ReadByte(file)
            Size = Short_SwapEndian(ReadShort(file)) - 2
            Pos = FilePos(file)

            ; if the type is from SOF0 to SOF3
            If (BType >= $C0) And (BType <= $C3)
                Prec = ReadByte(file)

                Height = Short_SwapEndian(ReadShort(file))
                Width = Short_SwapEndian(ReadShort(file))

                Exit
            EndIf

            ; Goto next marker
            SeekFile(file, Pos + Size)
        EndIf
    Wend

    Depth = 24
EndIf

CloseFile(file)

Info.TImageInfo = New TImageInfo
Info.Width = Width
Info.Height = Height
Info.Depth = Depth
Info.HasAlpha = HasAlpha

```

```
Return Info
End Function
```

```
; Utils
```

```
Function Int_SwapEndian%(n%)
Return ((n And $FF) Shl 24) Or ((n And $FF00) Shl 8) Or ((n And $FF0000) Shr 8) Or ((n And $FF000000) Shr 24)
End Function
```

```
Function Short_SwapEndian%(n%)
Return ((n And $FF) Shl 8) Or ((n And $FF00) Shr 8)
End Function
```

```
Function ReadByteString$(file, count = 0)
Ret$ = ""

If count = 0
Char = ReadByte(file)
While Char <> 0
Ret = Ret + Chr(Char)
Char = ReadByte(file)
Wend
Else
For i = 1 To count
Ret = Ret + Chr(ReadByte(file))
Next
EndIf

Return Ret
End Function
```

```
Include "ReadImageInfo.bb"
Include "conts.bb"
Include "IO.bb"
```

```
Type Button
Field cName$ ,cNameEvent$
Field iTypeButton
Field iTexture1
Field iTexture2
Field iInitPosX,iInitPosY,iFinalPosX,iFinalPosY
Field tImagenInfo.TImageInfo
Field iStateDraw
Field iPush$
End Type
```

```

Type ArrayButtons
    Field iCounter
    Field iActive
    Field vButton.Button[SIZE_VECTOR]
End Type

;-
;-----revisar la memoria asignacion

Function MakeButton(tRightGuiBar.ArrayButtons,
    cName$,cNameTexture2$,cNameEvent$,iTypeButton,ilnitPosX,        ilnitPosY,        fScaleX#=0.0,
fScaleY#=0.0,iStateDraw=True)

    If(Len (cName) = 0) Or (Len (cNameTexture2) = 0) Then
        Return "ERROR"
    End If
    ilmageTemp=LoadImage(cName)

    If ( Instr (cNameTexture2,"_NONE_")) Then
        ilmageTemp2=LoadImage(cName)
    Else
        ilmageTemp2=LoadImage(cNameTexture2)
    End If

    If (ilmageTemp = 0) Or (ilmageTemp2 = 0)Then
        Return "ERROR"
    End If

    iCounter = tRightGuiBar.iCounter
    tRightGuiBar.vButton[iCounter] = New Button
    tRightGuiBar.vButton[iCounter].cName$=cName
    tRightGuiBar.vButton[iCounter].iTypeButton=iTypeButton
    tRightGuiBar.vButton[iCounter].iStateDraw=iStateDraw
    tRightGuiBar.vButton[iCounter].iTexture1=ilmageTemp
    tRightGuiBar.vButton[iCounter].iTexture2=ilmageTemp2
    tRightGuiBar.vButton[iCounter].iPush="FREE"

    tRightGuiBar.vButton[iCounter].tImagenInfo=ReadImageInfo(cName)
    tRightGuiBar.vButton[iCounter].ilnitPosX=ilnitPosX
    tRightGuiBar.vButton[iCounter].ilnitPosY=ilnitPosY
    If ( fScaleX = 0.0 ) Then
        fScaleX=tRightGuiBar.vButton[iCounter].tImagenInfo.Width;
    End If
    If ( fScaleY = 0.0 ) Then
        fScaleY=tRightGuiBar.vButton[iCounter].tImagenInfo.Height;
    End If
    tRightGuiBar.vButton[iCounter].iFinalPosX=fScaleX + tRightGuiBar.vButton[iCounter].ilnitPosX
    tRightGuiBar.vButton[iCounter].iFinalPosY=fScaleY + tRightGuiBar.vButton[iCounter].ilnitPosY
    tRightGuiBar.vButton[iCounter].cNameEvent=cNameEvent

```

```

        ResizeImage tRightGuiBar\lvButton[iCounter]\iTexture1,fScaleX,fScaleY
        ResizeImage tRightGuiBar\lvButton[iCounter]\iTexture2,fScaleX,fScaleY
        tRightGuiBar\iCounter = iCounter + 1

    Return tButton
End Function

;-----

Function CheckButton.Button(tButtonGui.ArrayButtons,iTypeButton)
    For iLoop=0 To tButtonGui\iCounter-1
        tButtonGui\lvButton[iLoop]\iPush="FREE"
    Next
    For iLoop=0 To tButtonGui\iCounter-1
        If( tButtonGui\lvButton[iLoop]\iTypeButton = iTypeButton) Then
            If ( tButtonGui\lvButton[iLoop]\iStateDraw= True) Then
                If ( tMouse\iPosX > tButtonGui\lvButton[iLoop]\iInitPosX ) And (tMouse\iPosX <
tButtonGui\lvButton[iLoop]\iFinalPosX )Then
                    If ( tMouse\iPosY > tButtonGui\lvButton[iLoop]\iInitPosY ) And (tMouse\iPosY <
tButtonGui\lvButton[iLoop]\iFinalPosY )Then
                        If( tMouse\ButtonPressed = MouseLeftButton) Then
                            tButtonGui\lvButton[iLoop]\iPush="PUSH"
                            Return tButtonGui\lvButton[iLoop]
                        End If
                    End If
                End If
            End If
        End If
    Next
    Return Null
End Function

;-----

Function DrawGuiComponents(v_Button.ArrayButtons)
    For iLoop =0 To v_Button\iCounter-1
        If (v_Button\lvButton[iLoop]\iStateDraw = True) Then
            If (Not v_Button\lvButton[iLoop]\iTexture1 = 0) And (Not v_Button\lvButton[iLoop]\iTexture2 = 0) Then
                If(v_Button\lvButton[iLoop]\iPush="FREE") Then
                    DrawImage      v_Button\lvButton[iLoop]\iTexture1,      v_Button\lvButton[iLoop]\iInitPosX,
v_Button\lvButton[iLoop]\iInitPosY
                Else
                    DrawImage      v_Button\lvButton[iLoop]\iTexture2,      v_Button\lvButton[iLoop]\iInitPosX,
v_Button\lvButton[iLoop]\iInitPosY
                    Delay(10)
                End If
            End If
        End If
    End If

```

```

        End If
    Next
End Function

```

```

;-----

```

```

Function ReadFolders.ArrayButtons(cFolder$, iInitIconX, iInitIconY, iIconSize=50,iTypeButton)
    If( Len(cFolder)=0) Then
        Return Null
    End If

    tDir.ArrayButtons= New ArrayButtons
    tDir.iCounter=0
    tDir.iActive=False
    iHandleDir = ReadDir( cFolder );
    If (iHandleDir =0) Then
        Return Null
    End If

    Repeat
        cFile$=NextFile$(iHandleDir)
        If ( cFile = "" ) Then
            Exit
        End If
        If ( Not FileType (cFolder +"\\"+ cFile) = 2 ) Then
            If Instr(cFile, ".png") Or Instr(cFile, ".bmp") Or Instr(cFile, ".jpg") Then
                cNameTemp$=cFolder + cFile
                MakeButton(tDir,cNameTemp,"_NONE_",cNameTemp,iTypeButton,iInitIconX,iInitIconY+
(tDir.iCounter*(iIconSize+10)),iIconSize,iIconSize,False)
            End If
        End If
    Forever
    Return tDir
End Function

```

```

;-----FUNCIONES DE ALTO NIVEL-----

```

```

Function NoDrawIconFolder( tDirFolder.ArrayButtons)
    For iLoop=0 To tDirFolder.iCounter -1
        tDirFolder.vButton[iLoop].iStateDraw=False
    Next
End Function

```

```

;-----

```

```

Function DrawIconFolder( tDirFolder.ArrayButtons,iMinY,iMaxY)
    For iLoop=0 To tDirFolder.iCounter -1

```

```

        If ( tDirFolder.vButton[iLoop].iInitPosY >= iMinY ) And ( tDirFolder.vButton[iLoop].iInitPosY <= iMaxY )Then

```

```

        tDirFolder\vButton[iLoop]\iStateDraw=True
    Else
        tDirFolder\vButton[iLoop]\iStateDraw= False
    End If

Next
End Function
;-----
Function UpFolder( tDirFolder.ArrayButtons , iLimitY , iSpeed=20)

    If ( tDirFolder\vButton[tDirFolder\iCounter -1]\iInitPosY > iLimitY) Then
        For iLoop=0 To tDirFolder\iCounter -1
            tDirFolder\vButton[iLoop]\iInitPosY = tDirFolder\vButton[iLoop]\iInitPosY - iSpeed
            tDirFolder\vButton[iLoop]\iFinalPosY = tDirFolder\vButton[iLoop]\iFinalPosY - iSpeed
        Next
    End If
End Function
;-----
Function DownFolder( tDirFolder.ArrayButtons, iLimitY, iSpeed=20)

    If ( tDirFolder\vButton[0]\iInitPosY < iLimitY) Then

        For iLoop=0 To tDirFolder\iCounter -1
            tDirFolder\vButton[iLoop]\iInitPosY = tDirFolder\vButton[iLoop]\iInitPosY + iSpeed
            tDirFolder\vButton[iLoop]\iFinalPosY = tDirFolder\vButton[iLoop]\iFinalPosY + iSpeed
        Next
    End If
End Function

```

Anexo I: Convocatoria al concurso de arquitectura del museo virtual

CONCURSO DE DISEÑO ARQUITECTÓNICO PARA EL MUSEO VIRTUAL DEL PATRIMONIO CULTURAL DE LA UNAM

CONVOCATORIA

La Coordinación de Difusión Cultural a través de la Dirección General de Artes Visuales y el Centro de Ciencias Aplicadas y Desarrollo Tecnológico de la UNAM invitan a la comunidad universitaria a participar en el Concurso del ***Diseño Arquitectónico para el Museo Virtual del Patrimonio Cultural de la UNAM.***

DEFINICIÓN DEL PROYECTO ARQUITECTÓNICO

El Museo Virtual será un sitio en Internet dedicado a integrar virtualmente el patrimonio arquitectónico, artístico, histórico y científico de la UNAM, con el propósito de brindar información y propiciar la interacción del usuario con la diversidad de manifestaciones culturales preservadas en las colecciones universitarias.

Este Museo podrá ofrecer recursos para concebir exposiciones y construir espacios imaginados e inducir, en el usuario, la experiencia de navegar en un entorno tridimensional en tiempo real.

La arquitectura del espacio tridimensional, liberado de las tradicionales restricciones de la construcción, será sustento de la identidad del Museo Virtual del Patrimonio Cultural de la UNAM. De esta manera, el planteamiento arquitectónico deberá atender los requerimientos espaciales que faciliten el ejercicio museístico dentro del territorio virtual.

Por lo tanto, se somete a concurso de la comunidad el diseño tridimensional de la arquitectura del Museo Virtual del Patrimonio Cultural de la UNAM, con las siguientes bases:

Bases

El concurso se llevará a cabo a partir del día siguiente a la publicación de la presente convocatoria hasta el 30 de marzo de 2007. La recepción de proyectos se hará en el Centro de Ciencias Aplicadas y Desarrollo Tecnológico (CCADET) de la UNAM, Laboratorio de Interacción Humano-Máquina y Multimedios Informes al 56 22 86 02 ext 1169 att. Andrea Miranda o Francisco Caviedes

Condiciones de admisión

Podrán participar de manera individual o grupal todos los estudiantes, profesores e investigadores universitarios sin distinción de edad, país de origen, sexo o actividad.

La participación se limitará a un proyecto por persona o grupo de personas.

El proyecto será original y su autor o autores deberán tener la propiedad legal del mismo.

La recepción de proyectos será en el Laboratorio de Interacción Humano-Máquina y Multimedia del Centro de Ciencias Aplicadas y Desarrollo Tecnológico de la UNAM del en las fechas arriba citadas

Jurado

Estará conformado por especialistas con trayectoria ampliamente reconocida en los campos de la arquitectura, multimedia y museología integrado por 5 miembros de la Comunidad Universitaria y su fallo será inapelable.

Presentación de proyectos

Los proyectos de diseño arquitectónico virtual se presentarán por quintuplicado, en CD o DVD-R (libre de virus) ISO 9660:

- El modelo en formato 3DS, comprendiendo: texturas con coordenadas UV e iluminación.
 - o El modelo no deberá superar un *máximo* de 20,000 polígonos.
- Propuesta por escrito del diseño del Museo (máximo 15 cuartillas en formatos DOC o RTF)
- Serie de fotogramas del modelo en formato JPG (toma libre). Máximo 15 con dimensiones de 1024 x 768 a 72 dpi, separada de la propuesta escrita.
- Película en formato Quick Time con dimensiones de 720 x 480 comprimido para video, con un máximo de 180 segundos a 30 cuadros por segundo.
- En copia sencilla se presentará
- El nombre real del concursante o grupo, en un sobre separado y sellado.
- La impresión de la [solicitud](#) de inscripción llenada y firmada, donde se ceden los derechos (incluyendo texturas) a favor de la UNAM, en caso de ser el ganador.

La solicitud podrá obtenerse en la siguiente dirección:

<http://www.ccadet.unam.mx/~ihm/concursomv/>

El incumplimiento de cualquiera de los puntos anteriores invalidará la inscripción al concurso.

Premios

1º Lugar: \$20,000.00 MN

2º Lugar: \$10,000.00 MN

3º Lugar: \$ 5,000.00 MN

Reconocimiento al proyecto ganador

El autor o autores del proyecto premiado quedarán citados en los créditos del Portal del Museo Virtual del Patrimonio Cultural de la UNAM, como autores del Proyecto Arquitectónico Virtual del museo.

La entrega del reconocimiento a los proyectos ganadores será el 20 de abril en la sede del MUCA-Campus.

El autor o autores del proyecto ganador deberán ceder todos los derechos de autor a favor de la UNAM.

Devoluciones de proyectos

Los proyectos no premiados serán devueltos en el lugar de recepción a partir del 22 de abril.

Términos generales

Lugar de recepción

Los proyectos se entregarán a:

Andrea Miranda o Francisco Caviedes

Laboratorio de Interacción Humano-Máquina y Multimedios

Centro de Ciencias Aplicadas y Desarrollo Tecnológico (CCADET)

Circuito Exterior s/n (entre DGSCA y CUAED).

Ciudad Universitaria, Coyoacán, CP. 04510, México D. F.

Tel.: 56 22 86 02, ext. 1169

Horario: 12:00 a 14:00 h y 16:00 a 18:00 h

Sitio Web

La ficha de inscripción puede descargarse del sitio Web del concurso:

<http://www.ccadet.unam.mx/~ihm/concursomv/>

Cualquier situación no especificada en esta convocatoria será resuelta por el Comité Organizador del concurso. Toda duda será atendida en los correos:

concurso.mv@ccadet.unam.mx , francisco.caviedes@ccadet.unam.mx o andrea.miranda@ccadet.unam.mx

Atentamente

Comité Organizador del Concurso

Anexo J: Productos académicos derivados del proyecto

1. Francisco Cabiedes, Esther de la Herrán, Andrea I. Miranda Vitela, A. Libia Eslava, Cervantes, Alma Rivas Rangel, José Luís Pérez Silva, Ildikó Pelczer, Francisco Salgado, Carolina Flores-Illescas, José Luís Casillas, Graciela de la Torre, Rafael Sámano, Julia Molinar, José Manuel Magaña, Ariadna Patiño: MUSEO VIRTUAL DEL PATRINMONIO CULTURAL DE LA UNAM: ESPACIOS DE ACCIÓN MUSEAL en CLAMU 2008 7º Congreso Latinoamericano de Multimedieros, 6 y 7 Noviembre Centro Nacional de las Artes Cd. México 2008
2. Francisco Cabiedes, Esther de la Herrán, Andrea I. Miranda Vitela, A. Libia Eslava, Cervantes, Alma Rivas Rangel, José Luís Pérez Silva, Ildikó Pelczer, Francisco Salgado, Carolina Flores-Illescas, José Luís Casillas, Graciela de la Torre, Rafael Sámano, Julia Molinar, José Manuel Magaña, Ariadna Patiño: SISTEMA DE 3D QUE SOPORTA EL MUSEO DEL PATRIMONIO CULTURAL DE LA UNAM en SOMI XXIII Congreso nacional de Instrumentación, 1-3 Octubre, Xalapa, Veracruz 2008
3. Francisco Cabiedes, Esther de la Herrán, Andrea I. Miranda Vitela, A. Libia Eslava Cervantes, José Manuel Mondragón, Alma Rivas Rangel, José Luís Pérez Silva, Ildikó Pelczer, Francisco Salgado, Adidier Pérez-Gómez, Carolina Flores-Illescas, José Luís Casillas, Graciela de la Torre, Jorge Reynoso, Rafael Sámano, Julia Molinar, José Manuel Magaña, Alejandrina Escudero, and Ariadna Patiño; 3D System to supporting The Virtual Museum of the University's Cultural Heritage of the Universidad Nacional Autónoma de México; en 20th Anniversary International Conference on Systems Research, Informatics and Cybernetics, Symposium on Art and Science Julio 24-31, 2008, Baden-baden, Alemania
4. Francisco Cabiedes, Esther de la Herrán, Andrea I. Miranda Vitela, A. Libia Eslava Cervantes, José Manuel Mondragón, Alma Rivas Rangel, José Luís Pérez Silva, Ildikó Pelczer, Francisco Salgado, Adidier Pérez-Gómez, Carolina Flores-Illescas, José Luís Casillas, Graciela de la Torre, Jorge Reynoso, Rafael Sámano, Julia Molinar, José Manuel Magaña, Alejandrina Escudero, and Ariadna Patiño; The Virtual Museum of the University's Cultural Heritage at The Universidad Nacional Autónoma de México; En Museums and the Web 2008 Abril 9-12 Montreal, Québec, Canadá