



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
FACULTAD DE INGENIERÍA

**DESARROLLO DE SOFTWARE INTERACTIVO PARA LA
ENSEÑANZA DE MATEMÁTICAS Y FÍSICA**

T E S I S
QUE PARA OBTENER EL TÍTULO DE:
INGENIERO EN COMPUTACIÓN
P R E S E N T A N :
CYNTHIA NÁPOLES RODRÍGUEZ
OCTAVIO GREGORIO LINARES MILLÁN

M.I. HERNANDO ORTEGA CARRILLO



Ciudad Universitaria, México, D.F.

2005



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

A mi madre

A Ricardo

Cynthia

A mi madre	Por su amor, sus cuidados, sus consejos y su apoyo total.
A mi padre	Por su amor, su mano dura, sus consejos y su apoyo total.
A mi hermano	Por su ejemplo, apoyo e interés.
A Elizabeth	Por su amor y apoyo.
A mis primos: Jorge y Julián	Por haber compartido su niñez, juventud y parte de su madurez.
A mi prima Carola	Por su ejemplo, su fuerza, su madurez.
A mis queridos amigos Esteban, Hugo, Adrián, Gustavo y Magda	Por su apoyo y por permitirme ser parte de su historia.

Octavio

"El ser humano, como ser integral, es más que la suma de todas las dimensiones que lo integran. Es por ello que el pensamiento conduce a la acción, la cual no se ejecuta sin la intervención de una emoción que la conduzca. Entonces, frente a las diferentes propuestas tecnológicas, el ser humano ha de responder con toda su integridad y es afectado por ella en todo su ser. En consecuencia, estar sentado frente a una pantalla de computadora lo va a implicar a todo él. Sentado quizás olvide por un momento su dimensión cuerpo, y quizás olvide el problema que lo trasnochó, pero estará todo él allí. Luego, la propuesta a la cual se enfrente al momento de estar frente a la computadora, determinará el énfasis de la habilidad que se pretenda desarrollar desde la web. Así, en la web podrá encontrar propuestas que enfatizan el mejoramiento de la salud física, por ejemplo. Sin embargo, ¿dónde puede estar la eficacia en el desarrollo de habilidades cognitivas, motoras y afectivas a partir del contacto con la web? En primer lugar, en el impacto que se produzca al momento de presentar una información; en líneas generales, ésta ha de ser agradable, firme, singular, amplia y concreta. En segundo lugar, en la movilización del sujeto, es decir, hacer evidente la tendencia a ir hacia. Y en tercer lugar, al propiciar la interacción, algo así como posibilitar manipular de alguna forma lo que tiene al frente, incorporarlo e internalizarlo. ¿Acaso no son estas las características que se le pide a todo lo que ingresa a la web? Y si es así, podemos decir entonces, que existe un alto potencial de desarrollo en la web."

Arturo Madrigal

ÍNDICE

1	Introducción	1
1.1	Objetivo	1
1.2	Aprendizaje, educación a distancia y medios educativos	1
1.3	Metodología	8
1.3.1	Método científico: observación (investigación), hipótesis, experimentación, resultados y su correspondiente evaluación.	9
1.4	Referencias a otros trabajos análogos y las aportaciones del presente	9
1.5	Conclusiones	12
2	Investigación de bibliotecas de programación y herramientas de modelado	13
2.1	Breve descripción de bibliotecas gráficas y herramientas de modelado	15
2.1.1	Referencias a otros trabajos	18
2.1.2	Requerimientos	21
2.2	Conclusiones	23
2.2.1	Elección de plataforma, herramientas y bibliotecas	24
3	Descripción detallada de las herramientas, bibliotecas gráficas y cómo se relacionan ..	26
3.1	Herramienta de modelado.....	26
3.2	Biblioteca de programación	30
3.3	Conclusiones	38
4	Programación de aplicación demostrativa.....	39
4.1	Tiro Parabólico.....	39
4.1.1	Objetivos de la aplicación.....	39
4.1.2	Diseño	39
4.1.2.1	Modelado geométrico, apariencia física y comportamiento	40
4.1.3	Desarrollo	46
4.2	Hoyos negros	63
4.2.1	Objetivos de la aplicación.....	63
4.2.2	Diseño	64
4.2.3	Desarrollo	65
4.2.3.1	Enana blanca	66
4.2.3.2	Estrella de neutrones	72
4.2.3.3	Hoyos negros	73
4.3	Conclusiones	76
5	Resultados.....	77
5.1	Discusión de resultados	77
5.2	Evaluación	84
5.3	Conclusiones	85
6	Trabajo futuro y conclusiones.....	86
	Glosario	I
	Bibliografía y referencias	VII

INTRODUCCIÓN

La enseñanza de materias científicas como matemáticas y física es un área vital para el desarrollo de cualquier país. Si un país es capaz de acercar a sus jóvenes al estudio e investigación científica logrará preparar los cuadros que el día de mañana generen los avances tecnológicos y nuevos conocimientos. Cualquier nación del mundo que dependa permanentemente de los conocimientos generados en otros países quedará en desventaja y se atrasará inevitablemente.

Se necesita entonces impulsar hoy el desarrollo científico del mañana. Sin embargo, en esta como en muchas otras áreas, los recursos disponibles son escasos. Se requiere entonces maximizar su aprovechamiento y utilizar la creatividad e imaginación para crear nuevos caminos y formas. El presente trabajo desarrolla y expone una de las muchas posibilidades con las cuales se puede hacer llegar el conocimiento a los jóvenes a través de un medio relativamente nuevo, la red Internet.

Sin duda falta mucho para que en nuestro país cualquier adolescente tenga acceso a una computadora con acceso a Internet. Las condiciones económicas y sociales del país no permitirán un acceso masivo en el futuro inmediato. Sin embargo, hemos confiado en que de una u otra forma esta situación quedará superada en algún momento.

RESUMEN

Para desarrollar un software que ayude a la enseñanza de las materias de matemáticas y física se inició estudiando cómo aprende el ser humano. A lo largo del tiempo se han presentado varias corrientes, no todas coinciden, de hecho, algunas se contradicen.

Una vez establecido un marco teórico, se documentaron algunos trabajos ya realizados en el área de interés y posteriormente se definieron dos temas puntuales a tratar: tiro parabólico y hoyos negros. Para cada uno se estableció las herramientas y recursos necesarios para su desarrollo.

La aplicación de tiro parabólico se basó en un esquema de juego, mientras que la de hoyos negros se desarrolló bajo el concepto de sitio o portal web en la que se incluyeron animaciones, imágenes e información general referente al tema.

El trabajo detalla las características de las herramientas utilizadas y las etapas a través de las cuales se atravesó hasta llegar al resultado.

1 INTRODUCCIÓN

1.1 OBJETIVO

- Desarrollar aplicaciones interactivas que guíen, motiven y acerquen al estudiante a investigar y aprender las materias científicas.

1.2 APRENDIZAJE, EDUCACIÓN A DISTANCIA Y MEDIOS EDUCATIVOS

APRENDIZAJE

¿Cómo se aprende?

- ✓ Es una acción que se desarrolla a dos niveles: el comportamiento y el pensamiento.
- ✓ Reúne unas características particulares: orientado por objetivos, dirigido al desarrollo global del sujeto, delimitado por las necesidades personales y las convenciones sociales.
- ✓ Es un proceso en el que participan activa y conscientemente profesor y alumno.
- ✓ Los resultados del aprendizaje se ven como algo que depende tanto de la información que el profesor presenta como del proceso seguido por el estudiante para procesar tal información
- ✓ Se configuran dos tipos de actividad que condicionan el proceso de aprender: las estrategias de enseñanza (cómo se presenta el material en un tiempo y en una forma determinada) y las estrategias de aprendizaje (cómo el estudiante a través de su propia actividad organiza, elabora y reproduce dicho material).

Después de estas reflexiones podemos arriesgarnos a dar una definición:

Aprendizaje es el proceso de adquisición de una disposición, relativamente duradera, para cambiar la percepción o la conducta como resultado de una experiencia.

Teorías del aprendizaje

Teoría conductista

- El conocimiento consiste en una conducta pasiva.
- El conductismo de Skinner está formado por tres elementos fundamentales: estímulo discriminativo, respuesta operante y estímulo reforzante.
- Es una teoría:
 - Asociacionista, que implica que el conocimiento del ser humano se compone solamente de impresiones e ideas.
 - Ambientalista, porque influye considerablemente en el aprendizaje de la persona.

- Reduccionista, porque no reconoce los procesos mentales del pensamiento.
- También tiene su base en las corrientes filosóficas: empirismo, positivismo, pragmatismo y evolucionismo.
- Para el conductismo el aprendizaje es un cambio relativamente permanentemente de la conducta que se logra mediante la práctica y en una interacción reciproca de los individuos y su ambiente.
- El conductismo considera al ser humano como un ser pasivo carente de identidad y de intenciones. Lo compara con una máquina.

Teoría del procesamiento humano de Gagné

- Gagné sigue considerando al ser humano como una máquina. Pero más precisamente como una máquina de procesamiento de datos (computadora).
- No abandona los elementos conductistas como son: estímulo, respuesta y refuerzo.
- Esta teoría supone que las conductas de las personas tienen su origen en un proceso interno, por lo tanto procesan la información.
- Este modelo explica como, de manera intencional se puede orientar el aprendizaje hacia metas específicas y por lo tanto planificarlo, incluyendo adquisición de aptitudes. El principio básico es la planificación de la educación con base en el análisis de la tarea, desde una clase o curso hasta una carrera completa.

La no directividad en la educación

- Está sustentada en las corrientes filosóficas humanismo, existencialismo y fenomenología.
- Para este método el maestro deberá ser: coherente, tener empatía hacia el alumno y ser auténtico.
- La relación entre el maestro y el alumno ocupa una posición central.
- El concepto clave sobre la teoría de Rogers es la libertad, la cual se traduce en confianza para que el alumno desarrolle su inteligencia.
- El maestro ayuda al alumno a hacerse cargo de si mismo, la finalidad es procurar el crecimiento la maduración y la socialización.

Teoría cognitiva del aprendizaje

- El aprendizaje debe ser significativo para el alumno (Ausubel).
- El aprendizaje va de lo general a lo específico, haciendo las correspondientes diferencias.
- El alumno cuenta con ciertas estructuras cognitivas organizadas jerárquicamente.
- Aprendizaje por descubrimiento (Bruner).
- Pero el aprendizaje por descubrimiento deberá hacerse en función de metas.

Posición psicogenética (constructivista, Piaget)

- Parte de la lógica y de la psicología, ya que le interesan las estructuras cognitivas de la persona.
- El aprendizaje tiene su origen en la acción, conducida con base en una organización mental previa.
- La organización mental está constituida por estructuras y las estructuras por esquemas debidamente relacionado. La estructura cognitiva determina la capacidad mental de la persona
- El alumno debe aprender mediante su propia acción. La labor del docente consiste en crear un contexto favorable para el aprendizaje.
- El modelo constructivista tiene su estructura en el desequilibrio-reordenación-equilibrio.

Enfoque sociocultural de aprendizaje (constructivista, Vygotsky)

- Considera al ser humano un ser cultural donde el medio ambiente (zona de desarrollo próximo) tiene gran influencia.
- Las funciones mentales superiores se adquieren en la interacción social (deberá formar grupos de trabajo y esparcimiento).
- Las herramientas psicológicas permiten que el alumno aprenda.

Principios del aprendizaje

Los principios del aprendizaje psicológicos más importantes son:

- *Ley de la intensidad.* Con una experiencia fuerte y dramática se aprende mejor que con una experiencia débil
- *Ley del afecto.* Toda persona tiende a repetir las conductas satisfactorias y a evitar las desagradables.
- *Ley de la prioridad.* Las primeras impresiones tienden a ser más duraderas.
- *Ley de la transferencia.* Un determinado aprendizaje es extrapolable.
- *Ley de la novedad.* Todo acontecimiento o conocimiento novedoso e insólito se aprende mejor que lo que sea rutinario o aburrido.
- *Ley de la resistencia al cambio.* Los aprendizajes que implican cambios en la organización de la propia personalidad son percibidos como amenazantes y son difíciles de consolidar.
- *Ley de la pluralidad.* El aprendizaje es más consistente, amplio y duradero cuanto más sentidos (vista, oído, tacto, etc.) están involucrados en el proceso de aprender.

- *Ley del ejercicio.* Cuanto más se practica y se repite lo aprendido, tanto más se arraiga el contenido del aprendizaje.
- *Ley del desuso.* Un aprendizaje no evocado o utilizado en mucho tiempo puede llegar a la extinción.
- *Ley de la autoestima.* Existe una mayor asimilación cuando se tiene un elevado concepto de las propias capacidades.
- *Ley de la participación.* La participación activa en el proceso de aprendizaje redundante en una asimilación más rápida y duradera.

EDUCACIÓN A DISTANCIA

Reseña histórica

Según estudios, la educación a distancia organizada se remonta al siglo XVIII, con un anuncio publicado en 1728 por la Gaceta de Boston, en donde se refería a un material auto-instructivo para ser enviado a los estudiantes con posibilidad de tutorías por correspondencia.

En la Europa Occidental y América del Norte, la Educación a Distancia empezó en las urbes industriales del siglo XIX, con el fin de atender a las minorías, que por diferentes motivos, no asistieron a escuelas ordinarias.

Al finalizar la Segunda Guerra Mundial, se produjo una expansión de esta modalidad para facilitar el acceso a los centros educativos en todos los niveles, especialmente en los países industrializados occidentales, en los centroeuropeos y en las naciones en desarrollo "tercermundistas". Esto obedeció al incremento de la demanda de mano de obra calificada registrada.

Entre 1960 y 1975 se fundaron en África más de veinte instituciones de Educación a Distancia.

Para los años 70's el concepto de interactividad estaba ya había sido introducida dentro del concepto de educación a distancia con el surgimiento de videos de una dirección acompañados de audio de dos direcciones.

Los recelos hacia la modalidad habían disminuido y en este período se produce el gran auge de la tecnología educativa en la educación en general. Hay un gran desarrollo de la educación a distancia que es utilizada para resolver problemas puntuales en los sistemas formales.

Surgen los primeros programas de capacitación docente a distancia.

Para los años 80's la educación a distancia evoluciona en una herramienta verdaderamente interactiva con la aplicación de videoconferencias.

A fines de la década de los '90 nos encontramos con un gran despliegue de las ofertas a distancia en diversos ámbitos y para todos los gustos: desde una carrera universitaria o de postgrado, hasta cursos para la satisfacción de un interés personal o el uso del tiempo libre.

En su mayoría los usuarios de esta modalidad viven en zonas de gran dispersión, con necesidades especiales. También están las personas privadas de su libertad, los que trabajan muchas horas, o las madres en etapa de crianza.

En años recientes han tenido gran desarrollo las redes de computadoras de sistemas abiertos. En la actualidad, casi cualquier computadora se puede conectar a una red.

Las redes son un medio para compartir recursos, de forma tal que una computadora personal puede utilizar el poder de procesamiento de una supercomputadora conectada a la misma red. Estas redes también resultan eficaces para apoyar el trabajo en grupo.

El correo electrónico, y el acceso a fuentes comunes de información, distribuidas en los dispositivos de almacenamiento de las computadoras son parte de los servicios típicos que ofrecen las redes de computadoras, es el medio organizacional para apoyar la organización de actividades cooperativas, a través del intercambio de mensajes entre los usuarios de la red.

Aparici (1999) clasifica las tecnologías utilizadas según su desarrollo histórico, en cuatro etapas:

- *Primera etapa:* caracterizada por el dominio del material impreso, textos y manuales, por correspondencia e intercambio de documentos.
- *Segunda etapa:* denominada analógica, caracterizada por la utilización de televisión, videos, programas radiofónicos.
- *Tercera etapa:* se incorpora la informática a los procesos de producción tecnológica de materiales.
- *Cuarta etapa:* denominada digital y donde se integran los diferentes medios tecnológicos a través de redes como Internet u otros canales de distribución digital. Las tecnologías más utilizadas en estos momentos en la educación a distancia son una mezcla de medios de las diferentes etapas y, donde uno de ellos, predomina.

Definiciones

La educación a distancia es un término genérico, difícil de definir, ya que en él se incluyen las estrategias de enseñanza aprendizaje, que en el mundo se denominan de diferentes formas.

Cuando hablamos por la educación a distancia, normalmente nos referimos a un sistema de educación en el cual los alumnos y los profesores no están el mismo lugar.

A continuación presentamos algunas definiciones propuestas por personajes relacionados con esta metodología:

Jackson Bob define la educación a distancia como el uso de medios electrónicos e impresos como puente entre el espacio entre el profesor y el alumno cuando no están en el mismo lugar.

Miguel Ramos Martínez la define como una estrategia para operacionalizar los principios y fines de la educación permanente y abierta de manera que cualquier persona, independiente del tiempo y del espacio, pueda convertirse en sujeto protagónico de su aprendizaje.

Para Fainhole, la educación a distancia es una forma de educación no convencional que permite llegar a más personas y lugares utilizando enfoques y métodos innovadores.

Por otro lado, Michael Moore la define como un conjunto de procedimientos cuya finalidad es proporcionar instrucción por medios de comunicación impresos y electrónicos a personas que participan en un proceso de aprendizaje reglado en lugares y horarios distintos de los profesores.

Tomando en cuenta las anteriores definiciones se puede aproximar una definición:

La educación a distancia es una estrategia educativa basada en el uso intensivo de las nuevas tecnologías, estructuras operativas flexibles y métodos pedagógicos altamente eficientes en el proceso enseñanza-aprendizaje, que permiten que las condiciones de tiempo, espacio, ocupación o edad de los estudiantes no sean factores limitantes o condicionantes para el aprendizaje.

Hay tres tipos de escenarios de educación a distancia:

- Sincrónico, a través de cursos dictados en Internet
- Asincrónico con software en Internet
- Asincrónico con el uso de CD ROM.

Principales características de la educación a distancia:

- Implica un proceso de autoaprendizaje.
- Resulta accesible a grandes grupos.
- Permite la conjunción de masividad con alta calidad del servicio educativo.
- Permite un ritmo de aprendizaje acorde con las condiciones individuales.
- Promueve la autonomía y la responsabilidad individual del educando.
- Permite la conjunción del trabajo con la capacitación.
- Evita el desarraigo de los estudiantes.
- Atiende a necesidades diversas.
- Permite una mejor relación costo-beneficio.
- Permite una optimización de recursos.
- Combina diferentes medios y recursos didácticos.

- Permite el acceso a la educación de personas tradicionalmente excluidas del sistema tradicional.
- Tiende a la descentralización.
- Se orienta por objetivos educativos.
- Propende a la transferencia de contenidos flexibles.
- Se basa en equipos multidisciplinares.
- Procura una enseñanza personalizada y autosostenida.
- Requiere de procesos de comunicación pedagógica individualizados.
- Combina modalidades presenciales.
- Produce sus propios soportes didácticos y utiliza una variada gama de medios y tecnologías.
- Se dirige a grupos heterogéneos cuya participación es voluntaria.
- Es apta para estimular la capacidad de análisis, la responsabilidad, la participación, la organización y la libertad de los educandos.

Diferencias entre la educación a distancia y la educación presencial

La educación a distancia ha surgido como un intento de dar respuesta a las nuevas demandas sociales que la educación presencial no ha podido atender, pero resulta incorrecto suponer que aquella pueda sustituir totalmente a esta última.

Ambas formas educativas pueden beneficiarse mutuamente de su coexistencia y acción.

En el siguiente cuadro se detallan las principales diferencias:

Educación presencial	Educación a distancia
<ul style="list-style-type: none"> • <i>El profesor y los estudiantes están presentes en un mismo espacio-tiempo (durante las clases).</i> 	<ul style="list-style-type: none"> • <i>El profesor y los estudiantes pueden no estar presentes físicamente en el mismo espacio ni en el mismo tiempo.</i> • <i>Para que la comunicación se produzca, es necesario crear elementos mediadores entre el docente y el alumno.</i>
<ul style="list-style-type: none"> • <i>La voz del profesor y su expresión corporal son los medios de comunicación por excelencia. Se les llama presenciales a estos medios porque restringen la comunicación a un aquí y a un ahora. Otros medios visuales y sonoros son muy poco utilizados en la clase convencional y sólo sirven como apoyos didácticos o para complementar la acción del profesor.</i> 	<ul style="list-style-type: none"> • <i>La voz y el esquema temporal, o son sustituidos por otros medios no-presenciales, o serán registrados en grabaciones sonoras y visuales para ser transmitidos luego a otro espacio y en otro tiempo. Los medios no son simples ayudas didácticas sino portadores de conocimiento que sustituyen al profesor.</i>
<ul style="list-style-type: none"> • <i>La comunicación oral, característica en la enseñanza convencional, está</i> 	<ul style="list-style-type: none"> • <i>Adquieren gran importancia los medios como ser la palabra escrita</i>

<p><i>acompañada normalmente por gestos y movimientos de la comunicación no verbal.</i></p>	<p><i>(dominante por antonomasia), además de la radio, la televisión y otros medios audiovisuales.</i></p>
<ul style="list-style-type: none"> • <i>La relación directa, presencial, de los que se comunican hace que el diálogo pueda producirse también aquí y ahora, de manera inmediata.</i> 	<ul style="list-style-type: none"> • <i>La relación no-presencial de los que se comunican, es una forma de diálogo que por no acontecer aquí ni ahora, puede llamarse "diálogo diferido". O sea, el comunicador debe continuar un mensaje completo y esperar un tiempo para recibir la comunicación, de retorno en forma similar, al igual que ocurre con una carta.</i>

MEDIOS EDUCATIVOS

Los medios o recursos didácticos son componentes integrales del proceso de enseñanza y de aprendizaje.

Como forma de contribución al mejoramiento de la calidad de la educación, los medios audiovisuales y las nuevas tecnologías se han ido introduciendo en los contextos formativos e instructivos, reclamando un espacio cada vez más protagónico. Al tradicional tablero y al tradicional libro de texto, se le han ido sumando otros medios, como proyectores de diapositivas, retroproyectores, y los medios audiovisuales como el cine y el video; y últimamente el medio informático que ha ampliado sus posibilidades con la creación de entornos simulados, hipertextos y multimedios.

Clasificación de los medios:

- Medios tradicionales: Voz, tablero, libro, papelógrafo, franelógrafo, mapas, carteleras, maquetas, herbarios, terrarios, proyector de cuerpos opacos, proyector de diapositivas, retroproyector, grabadora, audiolibro, radio, televisión, cine, video.
- Nuevas tecnologías: basados en el incremento de la interactividad y más especialmente, la aparición de la informática, los ambientes digitales y los procesos asistidos por computadora, y su utilización en la enseñanza y aprendizaje se le denomina a: computadoras, redes computacionales, aulas multimedia, software educativo, internet, inteligencia artificial, sistemas tutoriales inteligentes.

EDUCACIÓN VIRTUAL

La aparición de Internet con las posibilidades para transmitir conocimientos, imágenes, textos y sonido a una multitud de usuarios simultáneamente está revolucionando la enseñanza tradicional en sus dos modalidades – presencial y a distancia – surgiendo una tercera y nueva modalidad de enseñanza: la enseñanza virtual, enseñanza *online* o e-

learning. Se trata de una forma de enseñanza basada en las nuevas tecnologías de la información y la comunicación, fundamentalmente en Internet.

Con el uso de las nuevas tecnologías los estudiantes no sólo no tienen que ir a la escuela y pueden aprender a su propio ritmo de estudio de acuerdo a los horarios que más les convengan (aprendizaje asíncrono) lo cual reduce el costo de oportunidad de la educación y permite insertar dentro del mercado educativo a alumnos que nunca antes hubieran podido formarse dentro del sistema educativo tradicional, sino que también vuelven más atractiva esta modalidad educativa lo que conlleva en un mayor número de alumnos y por consiguiente mayor difusión del conocimiento.

Por las ventajas que presenta, la UNESCO recomienda el *e-learning* como una herramienta que permite poner el conocimiento al alcance de todo el mundo

1.3 METODOLOGÍA

Para lograr un conocimiento de tal naturaleza, es preciso seguir determinados procedimientos que nos permitan alcanzar el fin que procuramos: no es posible obtener un conocimiento racional, sistemático y organizado actuando de cualquier modo; es necesario seguir un método, un camino que nos aproxime a esa determinada meta. Así, siguiendo esta premisa y lograr nuestro objetivo, utilizaremos el método científico con el cuál llevaremos este problema al análisis para encontrar las posibles soluciones.

Partiremos de considerar que el aprendizaje es un proceso en el que el alumno participa activamente y el profesor crea un contexto favorable para el mismo. En dicho proceso el alumno desarrolla estructuras cognitivas jerárquicas.

Consideramos también que la educación virtual es una estrategia válida y valiosa para enseñar las materias científicas y que deben aprovecharse los medios que las nuevas tecnologías ponen actualmente en nuestras manos.

1.3.1 Método científico: observación (investigación), hipótesis, experimentación, resultados y su correspondiente evaluación

Observación:

Actualmente los métodos didácticos usados para impartir las asignaturas dependen de diversos factores cambiantes, que en función de estas variaciones pueden desmotivar al estudiante y dificultar al profesor impartir la asignatura. Entre estos factores se pueden citar: los planes de estudio, el número de alumnos por aula, la disponibilidad del material adecuado, los medios (videos, rotafolios, proyectores, etc.) que no permiten interactividad y el escaso tiempo que se dispone para impartir un programa de por sí extenso.

Hipótesis:

La solución a este problema es utilizar precisamente lo que más llama la atención de los niños y jóvenes en la actualidad, que son los videojuegos e Internet. Serán una estrategia de motivación para acceder a mas inquietudes y poder trasladar de una manera práctica, real y aplicable los conocimientos que en materias científicas se dictan.

El aprendizaje por medio del software educativo, aparte de tener un contenido de nivel científico apropiado, debe ser atractivo e interactivo para el aprendiz, ya que se basa en herramientas multimedia como: sonido, videos, animaciones, hipertexto y otros, que le mantienen motivado al navegar a través del software, de manera que éste mantenga centrada su atención al asimilar los contenidos y simultáneamente tenga el control de lo que aprende. Ahora, la posibilidad de llevar este tipo de material a la práctica es amplia y es de ahí la fuerza innovadora que nos lleva a pensar en desarrollar una nueva metodología que permita un mayor cubrimiento y satisfacción de ese cliente llamado estudiante.

Experimentación:

- ✓ Investigación de bibliotecas de programación y herramientas de desarrollo.
- ✓ Descripción detallada de las herramientas, bibliotecas gráficas y cómo se relacionan.
- ✓ Desarrollo de software educativo demostrativo.
- ✓ Publicación el software desarrollado en la web y establecer estrategias de evaluación por parte de los usuarios.

Resultados y evaluación:

Evaluación de nuestra hipótesis con los resultados de la experimentación, y formulación de nuevas estrategias para seguir en el ciclo del método (o sea regresar a corregir nuestra hipótesis). Publicación en Internet del desarrollo para que pueda ser evaluado por el auditorio en general.

1.4 REFERENCIAS A OTROS TRABAJOS ANÁLOGOS Y LAS APORTACIONES DEL PRESENTE

Con los avances técnicos y tecnológicos de los últimos años se han creado y mejorado diferentes formas de enseñanza y aprendizaje mediante el uso de la computadora.

La creación de programas para la enseñanza no es nueva, pero dada la falta de herramientas y técnicas, sólo permitía desarrollar programas poco llamativos, en contraste con estos nuevos tiempos.

Las ventajas de la actualidad, como Internet, han permitido la creación de tutoriales, cursos en línea, vídeo conferencias, educación a distancia y software educativo; todas se fundamentan bajo el enfoque de la “Sociedad del Conocimiento”, que comprende el análisis, las metodologías y estrategias pedagógicas para el desarrollo de este tipo de programas.

Nos enfocaremos hacia el software educativo, ya que es el cimiento de la construcción de esta tesis.

SOFTWARE EDUCATIVO

El software educativo es un producto tecnológico diseñado para apoyar procesos educativos. Se concibe como uno de los medios que utilizan quienes enseñan y quienes aprenden, para alcanzar determinados propósitos. Es además un medio de presentación

y desarrollo de contenidos educativos, como lo puede ser un libro o un video, con su propio sistema de códigos, formato expresivo y secuencia narrativa. De esta manera, el software educativo puede ser visto como un producto y también como un medio.

Buscando en la red encontramos varios programas enfocados al aprendizaje de asignaturas científicas como matemáticas y física.

Adventure Math

Es un juego en el cual debe resolverse correctamente las cuentas ofrecidas por un bufón de la era medieval. Posee diez niveles distintos e incluye los conceptos de suma y resta.

<http://csa.scuole.bo.it/cd-rom/matem/amaths.html>

Easy Present Graphics

Sencillo constructor de gráficos estadísticos para ver en pantalla o imprimir. Posee una interfaz gráfica muy amigable, lo que lo convierte en ideal para los primeros pasos.

<http://www.hermetic.ch/egfc/multiple.htm>

Física

El programa permite varias opciones que permiten el estudio de diversos fenómenos físicos por medio de animaciones controladas por los datos ingresados por los usuarios. Contempla varios fenómenos físicos como por ejemplo: péndulo, rampa, tiro parabólico, etc.

http://www.pegar.com.br/pegar_arquivo.asp?link_id=4899

Functor 2.9

Graficador de funciones matemáticas que trabaja con campos escalares y gráfica en tres dimensiones. Incluye funciones para encontrar valores mínimos, máximos, realizar cálculos con expresiones y rotar los gráficos para verlos en distintas posiciones.

<http://functor.cjb.net/>

Funnels & Buckets

Es un programa realizado con el fin de agilizar la resolución mental de las cuatro operaciones básicas: suma, resta, multiplicación y división. En un corto lapso de tiempo (modificable según el nivel elegido) el estudiante deberá contestar el resultado de una operación entre dos miembros de 1 ó 2 cifras, que le sumará puntos según la respuesta sea correcta o no. El tipo de operación será elegida antes de comenzar la ejecución del programa, siendo de un solo tipo o de los cuatro elegidos aleatoriamente por la computadora.

<http://archives.math.utk.edu/software/msdos/k-12/funnels/.html>

Geometry Around Black Holes

Este sitio que incluye una explicación matemática de la geometría de los hoyos negros también presenta imágenes muy interesantes y un par de animaciones sobre temas relacionados.

<http://www.astro.ku.dk/~cramer/RelViz/>

Graph+ 1.1

Aplicación que resulta ideal para graficar todo tipo de funciones matemáticas. Puede hallar integrales, calcular tangentes y ampliar gráficos para visualizarlos a distintas escalas.

<http://users.cybercity.dk/~bse26236/free/free.html>

Mechanics

Sitio donde se encuentran variadas animaciones sobre temas de física como: el péndulo de Foucault, el experimento de Galileo sobre la caída de objetos, movimiento de un cuerpo en presencia de un campo gravitacional, hoyos negros, oscilación armónica, etc.

Planet finder

En este sitio encontramos una aplicación que muestra la posición de los planetas del sistema solar de acuerdo al lugar, fecha, hora, latitud y longitud proporcionados por el usuario.

<http://www.lightandmatter.com/area2planet.shtml>

Projectile motion

Se presenta una aplicación que permite parametrizar la altura, la velocidad y el ángulo de lanzamiento. Al correr el proceso se observa la trayectoria del proyectil lanzado así como las variaciones que las fuerzas que actúan sobre el mismo a lo largo del tiempo.

<http://www.ngsir.netfirms.com/e/1.htm>

Sir Addalot Game

Es un juego interactivo, mezcla de acción y educación. Se trata de un caballero, manejado por el alumno, que debe vencer al dragón respondiendo correctamente las cuentas que se le aparecen a lo largo de los diversos castillos. Las cuentas, que incluyen suma, resta, multiplicación y división, son ingresadas previamente por el maestro.

<http://takegame.com/others/htm/addalot.htm>

Tables Tutor (versión 1.0)

Presenta multiplicaciones (con operandos del 1 al 20 según la opción de dificultad elegida) y el alumno tiene un cierto tiempo para resolverlas.

<http://www.winsite.com/bin/Info?500000031278>

1.5 CONCLUSIONES

El aprendizaje debe ubicarse en un escenario compuesto por diferentes elementos, entre los que consideramos muy importante la motivación del estudiante, ésta condición se logra a través de confort que rodee el ambiente de estudio. Con las metodologías enunciadas se logra superar las dificultades que presentan las aulas y demás espacios que en ocasiones pueden ser inapropiados para el desarrollo de actividades de docencia.

La educación virtual contribuye a democratizar la educación y a realizar el principio del derecho a ésta para todos con igualdad de oportunidades. El software educativo interactivo es uno de los medios de enseñanza que puede ser de la más alta calidad y con un potencial enorme en el futuro. Sobre todo si tenemos en cuenta que día a día se están abaratando los costos de producción.

El software educativo encontrado es sólo una muestra del mar de posibilidades que existen ahora en la web, pero en nuestra búsqueda no encontramos una fusión entre este sistema de aprendizaje y un contexto virtual, ya que esto nos permitiría una interactividad, no solo con el programa o servidor, sino también con otro usuario ajeno a nuestra computadora; esto haría mas completo el aprendizaje. Enfocándonos en esa dirección anexamos cinco capítulos más, de los cuales a continuación damos una breve explicación:

- *Capítulo 2*

Se investigarán las bibliotecas de programación y herramientas de modelado que hay en el mercado. En base a las características de cada una se elegirá la más óptima para desarrollar software demostrativo.

- *Capítulo 3*

Se describirá el software de programación y de modelado elegido, y si existe una sinergia entre ellos.

- *Capítulo 4*

Se detallarán los fenómenos físicos a representar: tiro parabólico y hoyos negros. Se mostrará el desarrollo de dos aplicaciones demostrativas, apoyándose en bosquejos, modelos gráficos y código.

- *Capítulo 5*

Se expondrán y discutirán los resultados obtenidos, así como las estrategias de evaluación de las aplicaciones interactivas.

- *Capítulo 6*

Se comentará los posibles desarrollos a futuro en este mismo contexto, haciendo hincapié en la tecnología vigente y tomando en cuenta los resultados obtenidos.

2 INVESTIGACIÓN DE BIBLIOTECAS DE PROGRAMACIÓN Y HERRAMIENTAS DE MODELADO

Introducción

El desarrollo de software debe llevarse a cabo mediante una serie de etapas ordenadas. Podemos decir que la primera etapa consiste en definir el problema a solucionar, es decir, tener un objetivo claramente definido y acotado. Una vez que se tiene una visión general de la situación puede procederse a elegir las herramientas con las cuales se va construir o desarrollar. Para llevar a cabo la elección más óptima deben conocerse las características de las herramientas y elementos disponibles.

Hay que resaltar que no se debe pensar únicamente en características técnicas, las cuales sin duda son fundamentales, ya que existen otra serie de elementos importantes a considerar, como son los factores económicos, la disponibilidad, la confiabilidad, adaptabilidad y compatibilidad, entre otros.

Para el presente trabajo se requieren básicamente dos tipos de herramientas: bibliotecas gráficas y herramientas de modelado. A continuación se expondrán las diversas opciones que se encontraron así como sus características. El objetivo es determinar cuales, de entre las opciones existentes, son las herramientas óptimas para el desarrollo del presente proyecto.

Bibliotecas gráficas

Originalmente los archivos html describían páginas estáticas que contenían diferentes categorías de texto (párrafos, encabezados, etc.) e imágenes en dos dimensiones. En febrero de 1994 Mark Pesce y Tony Parisi presentaron el primer *3D Web browser*. Casi al mismo tiempo, Rick Carey de *Silicon Graphics* creó *Open Inventor* un lenguaje de imagen descriptivo. Juntos, Pesce, Carey, Bell y Parisi presentaron el primer borrador de la especificación VRML 1.0 (*Virtual Reality Modelling Language*) en octubre del 1994. Posteriormente, con la contribución de catorce compañías se desarrolló el nuevo estándar VRML 2.0. A diferencia de VRML 1.0, VRML 2.0 era capaz de producir mundos multimedia dinámicos. En julio de 1997 aparece VRML 97, el estándar ISO/VRML que introdujo pequeños cambios a VRML 2.0. Desde 1997 el estándar gráfico VRML no ha conocido ninguna variación. Este estándar, utilizado en Internet en incontables sitios, permitió en su momento la interactividad del usuario con mundos 3D de una forma simple a través de sus opciones de navegación o una interacción más complicada por medio de programación. VRML no ha sido impulsado por ninguna firma comercial, excepto un club de usuarios, produciéndose un estancamiento del mismo. Ante esta situación de escasa evaluación del estándar, diferentes firmas han realizado sus propios desarrollos de software gráfico para Internet denominado genéricamente como gráficos WEB3D. Estas compañías han intentado rellenar el vacío dejado por el estándar e intentado mejorar las páginas destinadas al comercio o venta electrónica y de entretenimiento.

Originalmente un editor de textos bastaba para crear mundos gráficos simples, sin embargo a medida que se volvieron más complejos se desarrollaron herramientas adicionales.

Un modelador es una aplicación de dibujo 3D que se utiliza para crear mundos animados. Un *plug-in browser* para web se utiliza para visualizar archivos 3D.

Un ambiente multiusuario es aquel en el cual un usuario puede ver u oír las entradas de otros usuarios. Inicialmente, estos ambientes eran solo texto. Pero con la aparición de mundos multiusuario vino la posibilidad de comunicación en un espacio visible generado por la computadora. Un ambiente multiusuario 3D contiene representaciones del mundo en 3D que reflejan a todos los usuarios que están actualmente activos (estos reflejos se conocen como *avatars*). Estos ambientes permiten a los usuarios comunicarse vía texto escrito, micrófonos, altavoces y gestos físicos de los *avatars*.

Cuando alguien entra en cualquier ambiente multiusuario 3D necesita un objeto que lo represente frente a los demás, a este objeto se le llama *avatar*. Un *avatar* puede ser tan simple como una esfera o tan complejo como un modelo humano. El *avatar* puede moverse alrededor del espacio 3D y puede contar con un número de gestos animados que pueden ser disparados por el usuario y ser observados por los demás. Un *avatar* es una representación de uno mismo que otros usuarios firmados pueden ver. A su vez uno puede ver los *avatars* de los demás. Muchos de los *avatars* tienen *scripts* predefinidos que al ejecutarse, al presionar un botón por ejemplo, causan el movimiento del *avatar* o un gesto en cierto sentido. Los gestos disponibles son principalmente cosas comunes como saludar, bailar, sonreír o pronunciar algunas palabras así como acciones y emociones que son fáciles de interpretar por algún movimiento. Además de los gestos, un espacio multiusuario cuenta con *chat box* y *chat-log*. El *chat-log* es el lugar donde se muestra la conversación de todos los usuarios y el *chat-box* es donde cada uno escribe sus comentarios.

Herramientas de modelado

Sirven para el diseño del modelaje de los diferentes elementos que intervendrán en las aplicaciones y contribuirán a la creación de efectos especiales.

Actualmente los términos Infografía y modelado son muy habituales dentro de muchos ámbitos de la informática, sobre todo en el campo de la imagen y el sonido. Quien no ha visto en la pantalla del cine dinosaurios perseguir a personas y devorarlos, o grandes mundos fantásticos llenos de criaturas inimaginables, películas realizadas completamente por personajes ficticios o publicidad en la que unos objetos se convierten en otros distintos. Todo esto cada vez sorprende menos, pero para llegar a este punto se ha recorrido un largo camino en la que se ha invertido mucho tiempo y dinero en software y tecnología para obtener estos resultados tan espectaculares y, sobre todo, reales.

Si hacemos un poquito de historia, aproximadamente quince años, y nos remontamos a los orígenes de los ordenadores domésticos, estos apenas podían mostrar imágenes y a muy baja resolución. Posteriormente aparecieron los entornos gráficos (*Windows*, *OS/2*, *X-Windows*) los que permitieron el desarrollo de software de tratamiento de imagen y video, pero con resultados aún pobres. Por aquellos momentos surgió un software de modelado denominado 3D Studio, que funcionaba bajo MSDOS y que permitía crear objetos tridimensionales y darles una apariencia real.

Esto fue el comienzo de un campo hasta entonces inexplorado y reservado únicamente para plataformas Silicon Graphics de gran potencia: La Infografía o el modelado 3D.

Según fue pasando el tiempo y con los avances tecnológicos y de software, comenzaron a surgir nuevos programas de modelado tales como Lightwave o Truespace y aquellos programas que hasta ese momento sólo podían ejecutarse en máquinas Silicon, tales como Softimage o Alias Maya.

Por su parte Autodesk daba un cambio a su “anticuado” 3D Studio y lanzaba al mercado 3D Studio Max, un programa de modelado que sobrepasaba las características de todos y se ponía casi a la par de Softimage o Maya, los más potentes en ese momento.

En la actualidad 3DStudio Max se encuentra a la cabeza de este campo, junto con Alias Maya que es el actual rey dentro de los software de modelado pero con unos costes mayores.

2.1 BREVE DESCRIPCIÓN DE BIBLIOTECAS GRÁFICAS Y HERRAMIENTAS DE MODELADO.

El siguiente software es tecnología vigente, cuya vida en el mercado es favorable ya que hay una demanda consistente, e incluso algunos son líderes en su rama.

Las nuevas tecnologías ofrecen las siguientes ventajas:

- 1) Son relativamente fáciles de crear y de programar.
- 2) Normalmente requieren un *plug-in* de pequeño tamaño para su visualización aunque alguna tecnología no requiere ninguno.
- 3) Válidos para diferentes plataformas (PC, Mac, Unix etc.).
- 4) Calidad de imagen sorprendente.
- 5) Efectos de sombreado (reflexiones, *anti-aliasing*, sombras, etc.).
- 6) Facilidad para su integración en páginas web.
- 7) Basados en estándares.
- 8) Normalmente utilizan sistemas de compresión para que la descarga por el usuario resulte breve.
- 9) Importantes empresas detrás de estos desarrollos lo que garantiza un desarrollo continuo de estas tecnologías.

En contra de estos desarrollos cabe destacar:

- 1) El software es propietario por lo que el desarrollador depende de las intenciones de la firma comercial.
- 2) Normalmente se requiere *plug-in*.
- 3) La implementación de estos desarrollos se encuentra orientada hacia los principales clientes de los mismos normalmente comercio y entretenimiento electrónico.

Bibliotecas gráficas

Anark Studio

Software de multimedia que sirve para crear aplicaciones interactivas en tiempo real.

Características:

- Programación en JavaScript.
- Aplicaciones independientes, screensavers o vía Web con el Cliente Anark de Web o el Cliente Anark para Windows Media 9 Player.
- AMX *Plug-ins* para el modelado de aplicaciones en 3D.
- Mezcla de elementos 2D y 3D .
- Comprime el tamaño de archivos.
- Permite la manipulación de gráficos apoyado en una ventana *Workspace*.
- Traza un mapa de video sobre objetos de 3D.
- Materiales animados y texturas.
- Su panel *Inspector* permite cambiar varias propiedades de objetos del proyecto, que se importan usando la Biblioteca.
- Gratis Anark Client (para ver demos) y por quince días Anark Studio (desarrollo).

WildTangent Web Driver

Es una avanzada plataforma para el desarrollo y distribución de juegos con contenido multimedia para Internet.

Características:

- Desarrollo de juegos usando una amplia gama de lenguajes de programación: Java, JavaScript, Visual Basic, VB Script, C y C++.
- Utiliza Microsoft DirectX e ingeniería WildTangent para la aceleración de hardware o software de 3D, 2D y sonido.
- Su nuevo controlador para Web permite jugar juegos en línea sin tener que descargarlos e instalarlos.
- Ofrece multimedia de compresión para geometría 3D, sonido, texturas y mapas de bits. Descarga rápidamente el contenido, ya que descomprime los archivos eficientemente.
- Soporta COM basado en el acceso a la ingeniería APIs.
- Importa materiales 3ds MAX y geometría Maya.
- Soporta Joystick y otros dispositivos de control de entrada.
- Para cuestiones académicas el software es gratis.

Macromedia Flash

Es una aplicación para desarrollar interfaces de usuario y aplicaciones web ricas en contenido.

Características:

- Aplicaciones independientes.
- El usuario debe descargar e instalar el reproductor de Macromedia Flash para poder interactuar con contenidos desarrollados con Flash.
- Elementos 2D. No soporta elementos 3D de forma natural, pero es posible crearlos mediante ciertas técnicas.
- Permite incluir audio, video y gráficos.
- Descarga progresiva de video
- Programación con ActionScript

Alambik

La tecnología Alambik coloca medios de comunicación interactivas multicontenidas.

El corazón de Alambik es el MCNL (Multi Content Natural Language, también conocida como Alambik Script). Basado en *scripts*, posee posibilidades de dinamismo, interactividad con el usuario e integración con contenidos de flujo y archivos audiovisuales realmente llamativos.

El peso es minúsculo y permite hacer películas complejas, con movimientos tridimensionales, banda sonora sincronizada e interacción real con el usuario. Similar en el empleo de HTML, MCNL combina un juego completo de instrucciones de programación con una amplia colección de comandos audiovisuales (vídeo, audio, 3D, 2D, etc.). Existen *plug-ins* gratuitos para 3D-Max que exportan a este formato.

Características:

- Lenguaje de programación Alambik Script.
- Audio y video.
- Imágenes de 3D y 2D.
- Gráficas basadas en vectores.
- Efectos especiales.
- Procesamiento de imágenes.
- Es gratis sólo el Alambik Basic Core, las demás herramientas tienen un costo.

Herramientas de modelado

Maya

Maya es un software de modelado 3D, especialmente diseñado para la animación de personajes y la creación de efectos especiales.

Características:

- Deformaciones que se pueden agrupar, reordenar y animar.
- Un conjunto de herramientas de IK (*Inverse Kinematics*) para tener el control preciso sobre el movimiento de los personajes.
- Control de la animación facial a partir de un conjunto de *Blend Shapes*.
- Herramientas de modelado de la piel.
- Control de las expresiones de los personajes.

- Sincronización sonora integrada.
- Maya también incorpora un avanzado conjunto de herramientas para la creación de efectos visuales, como la caída de agua sobre un molino o la simulación de la colisión de dos galaxias.
- Costo \$1999.00 USD (Maya completo v.5).

3D Studio Max

Es un programa de animación en 3D de nueva generación. Está totalmente escrito en un lenguaje de programación orientado a objetos. Ofrece sobre una PC la funcionalidad reservada hasta ahora a costosas estaciones de trabajo gráficas. Desarrolla software para los mercados de vídeo, juegos, web, medios interactivos, diseño industrial y visualización.

Características:

- Integración de los entornos de modelado y animación.
- Manipulación en tiempo real con o sin presencia de tarjeta de aceleración de gráficos en 3D.
- Función deshacer multinivel.
- Soporte de sistemas con multiprocesadores.
- Un entorno de desarrollo excepcional gracias a su arquitectura *plug-in*.
- Abierto a desarrollos externos; todo nuevo módulo integrado dentro de 3D Studio Max puede acceder a las funciones de base de la interfaz de 3D Studio Max, lo que permite abrir ilimitadas perspectivas.

2.1.1 Referencias a otros trabajos.

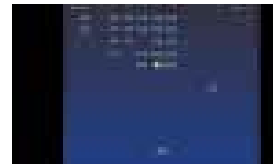
Hay diferentes aplicaciones del software presentado, la mayoría de ellas están enfocadas hacia el entretenimiento y la educación, veremos algunos ejemplos:

Bibliotecas gráficas

Alambik

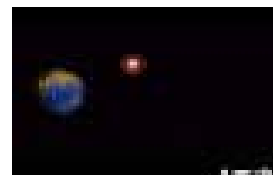
Insectoid

Es un juego que trata de salvar a la tierra de la invasión de los aliens. La detección de colisión y la monopista de secuencias son combinadas con duendecillos 2D, un texto macro, la música y el sonido.



Earth

Una escena de 3D que destaca la tierra que gira dentro de un universo animado de 2D. Este guión ilustra el poder de combinar mapas de bits en 2D, en tiempo real, primitivas 3D, y la música.



Anark Studio

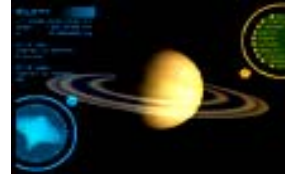
Fuel Cell

Visualización interactiva que muestra como una célula de hidrógeno de combustible trabaja. Uno aprende los componentes, el flujo y la operación básica.



Solar System

Se aprecia el sistema solar en una aplicación de 3D. Existe una interactividad en un ambiente de inmersión.



Warehouse Demo

Una aplicación en donde uno aprende como hacer un registro de envío y el procedimiento de almacenaje de paquetes.



Flash

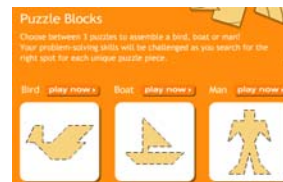
Could Dreamer

Mediante una interfaz gráfica se diseña una nube que después puede visualizarse dentro de un ambiente natural.



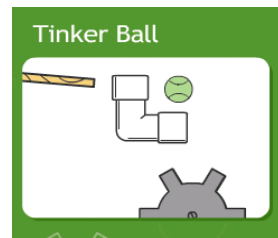
Puzzle Blocks

En este sitio se presentan 3 opciones de rompecabezas para armar (un pájaro, un barco y un hombre). Las piezas pueden moverse y rotarse.



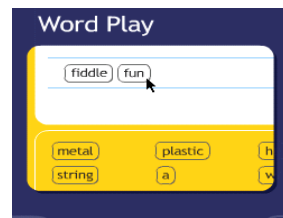
Tinker Ball

El problema consiste en construir con un número finito de elementos disponibles un camino para que la pelota (que se encuentra colocada sobre una repisa en la parte superior) caiga en un cesto colocado en la parte inferior. Los elementos disponibles se pueden colocar y rotar de acuerdo a las necesidades.



Word Play

Se trata de formar historias contando con un conjunto de palabras aleatorias. El juego permite ver lo que otros usuarios han compuesto.



WildTangent Web Driver

Dark Orbit

Es un juego de combate aéreo tipo salón creado por Phil Shenk.



Front 9

Es un juego de minigolf con unos recorridos llenos de detalles y fácil de controlar los golpes. El recorrido es de 9 hoyos con unos efectos y un fondo muy bien cuidado. Hay 4 vistas diferentes. El juego se ejecuta en el navegador.



SabreWing

Es un veloz simulador de combate espacial para un solo jugador. El juego presenta tres niveles de dificultad, una campaña de cinco misiones y un piloto acompañante que seguirá las órdenes del jugador. El juego incluye efectos dinámicos de iluminación y sombras.



Herramientas de modelado

Maya

DroneZ

Este juego transcurre en el Ciberespacio, y es una mezcla de acción y táctica donde primarán los reflejos con el ratón. Todos los escenarios y personajes del juego han sido modelados con Maya, lo que da una idea de su calidad y cantidad de polígonos



Final Fantasy

Se trata de la primera película que contiene personajes humanos realistas, y con gran detalle. Los autores crearon un software especial para los personajes, en especial se trabajó el movimiento de la melena de la protagonista.

Es una producción de Square Usa, que utilizó para los protagonistas exclusivamente Maya junto con herramientas creadas a medida por la productora basándose en la utilidades que maya ofrece.

Cada personaje esta creado con el mínimo detalle, desde los poros de la piel, a los pelos de la manos, pecas, e imperfecciones introducidas para dar un aspecto más humano a los protagonistas

Comenzaron utilizando modelos NURBS para los personajes y entornos pero cambiaron a modelos poligonales por su flexibilidad y porque con el "Subdivision Surfaces" de Maya se podían agregar los más mínimos detalles.



El realismo de la animación de los personajes fue logrado gracias a la captura de movimiento (*Motion capture*) y de cuadros clave (*keyframing*). Se crearon nuevas herramientas para la creación de pelo y ropas, para ello se recurrió a programas API (Application Programming Interface) y MEL (Maya Embedded Language) de Maya.



3D Studio Max

School House

Una interpretación visual propuesta para el proyecto de una escuela. Nos muestra el entorno y la arquitectura, así como la textura de los materiales.



The Blue dream

Decoración de un cuarto azul. Fue creado solo con 3d Studio Max R4 & Max Havok para los efectos de la tela.



Wonder Shrine:Takuya

Animación humanoide cuyos movimientos parecen reales. Caracteres de 3D con estructura esquelética (como un humano) sobre la web. Takuya puede caminar alrededor del santuario y obtener una fortuna.



2.1.2 Requerimientos.

Para que el funcionamiento en la PC de las bibliotecas gráficas y las herramientas de modelado sea de forma eficiente se necesitan de algunas condiciones, tanto en hardware como en software.

Bibliotecas gráficas

Anark Studio

Requerimientos:

- Microsoft Windows 98 SE/2000/ME/XP.
- Pentium II 300MHz o mayor.
- Microsoft DirectX 7 o posterior.
- 128 MB of RAM o mas.
- Tarjeta gráfica que soporte DirectX 7 o posterior (ejemplo: NVIDIA GeForce2 MX o posterior).
- Windows Media Player 7.

- Microsoft Internet Explorer 5.0 o posterior.
- Mínimo 15MB en disco duro.

WildTangent Web Driver

Requerimientos:

- Microsoft Win98, Win98 SE, Win ME, Win 2000, Win XP Pro, o Win XP Home
- PC w/300 MHz o más rápido.
- Intel Pentium II o mayor, equivalente a familia AMD K6/Athlon.
- 64 MB RAM (128 MB en Win2000, Win XP).
- Adaptador de video en color de 16 bits (hardware de aceleración 3D recomendado).
- 28.8K módem o más rápido (recomendado módem 56K).
- Microsoft Internet Explorer 5.0, 5.5, or 6.0, AOL 6.0, 7.0, Netscape Navigator. 4.7x, 6.2x.
- Microsoft Direct X versión 5.0 o mayor (recomendado DirectX 7.0).

Flash

Requerimientos:

- Pentium III a 600 MHz o superior, Power PC G3 a 500 MHz o superior
- 128 MB en RAM, recomendable 512 MB
- 347 MB de espacio en disco duro
- Sistema operativo Windows 98 o superior, Mac OS X 10.2.8 o superior
- Microsoft Internet Explorer 5.x o superior, Netscape Navigator 4.7x o superior, AOL 8.0, Mozilla 1.x o superior, Opera 7.11
- Algunas características requieren QuickTime 6.3 o QuickTime Pro 6.3

Alambik

Requerimientos:

- PC básica.
- Velocidad 200MHz MMX.
- 16MB RAM.
- Servidores estándar HTTP.
- 56k MODEM.
- Servidor estándar (CGI,PHP, etc.).

Herramientas de modelado

3D Studio Max

Requerimientos:

- Procesador Intel® a 200 MHz mínimo(sistema Pentium® II)
- 128 MB RAM mínimo.
- 250 MB de espacio en disco duro para archivo Swap (importante cuando se depende de escenas complejas)

- Tarjeta gráfica que soporte 1024x768x16-bit color.
- Acelerador gráfico OpenGL y Direct3D, 24-bit color, Acelerador gráfico preferido 3D.
- Mouse.
- Dispositivos multimedia.

Maya

Requerimientos software:

- Microsoft® Windows® XP Professional, Microsoft Windows 2000 Professional (Service Pack 2 o superior).
- SGI® IRIX® 6.2.15.
- RedHat™ Linux® 7.3 or 8.0.
- Apple® Mac® OS X 10.2.4 o superior

Browsers:

- Microsoft® Internet Explorer® 4.0 o superior
- Netscape® 7.0 o superior.

Requerimientos hardware(como mínimo):

- Intel® Pentium® II- a 300 MHz mínimo o posterior, procesador AMD Athlon™.
- 512 MB RAM.
- CD-ROM.
- Acelerador gráfico OpenGL® .
- Mouse de tres botones.
- 450 MB de espacio en disco duro.

2.2 CONCLUSIONES

Dado el análisis exhaustivo de las diferentes opciones presentadas en este capítulo, se elaboró una tabla comparativa que servirá de punto de referencia para la evaluación entre las diferentes opciones disponibles en diferentes contextos de tecnología para sistemas PC:

BIBLIOTECAS GRÁFICAS				
	Anark	Wildtangent	Alambik	Flash
PC				
Procesador	Pentium II o superior	Pentium II o superior AMD k6/Athlon	Pentium II o superior	Pentium III Power PC G3
Velocidad (mínima)	300 MHz	300 MHz	200 MHz	600 MHz
Memoria RAM (mínima)	128 MB	128 MB	16 MB	128 MB (recomendable 256 MB)
Espacio disco duro (mínimo)	15 MB	15 MB	15 MB	347 MB
Modem	56 k	28.8 k	56 k	
Plataforma				
Sistema operativo	Microsoft Windows 98 o superior	Microsoft windows 98 o superior	Microsoft windows 98 o superior	Windows 98 SE o superior Mac OS X 10.2.8 o superior
Microsoft Direct X	v. 7 o posterior	v. 5 o superior		
Microsoft Media Player	v. 7 o posterior			
Browser	Microsoft Internet Explorer 5.0 o superior	Microsoft Internet Explorer 5.0 o superior AOL 6.0 o superior Netscape Navigator 4.7x	Microsoft Internet Explorer 5.0 o superior	Microsoft Internet Explorer 5.x o superior Netscape Navigator 4.7x o superior AOL 8.0 Mozilla 1.x o superior Opera 7.11 (dependiendo del SO)
Lenguajes de programación	Java Script	Java Script Java Visual Basic VB Script C C++	Alambik Script	ActionScript
Características multimedia	Compresión 2D y 3D Materiales 3D Max Audio	Compresión 2D y 3D Materiales 3D Max Materiales geometría Maya Soporta Joystick y otros dispositivos de control Audio	2D y 3D Materiales 3D Max Audio	Compresión 2D Audio Video

HERRAMIENTAS DE MODELADO			
		Maya	3D Studio Max
PC			
	Procesador	Pentium II o superior AMD k6/Athlon	Pentium II o superior
	Velocidad (mínima)	300 MHz	200 MHz
	Memoria RAM (mínima)	512 MB	128 MB
	Espacio disco duro (mínimo)	450 MB	250 MB
	Hardware adicional	Tarjeta aceleradora OpenGL Mouse 3 botones con el respectivo manejador	Tarjeta aceleradora OpenGL Mouse 3 botones con el respectivo manejador
Plataforma			
	Sistema operativo	Microsoft windows 98 o superior RedHat Linux 7.3 o 8.0 Apple Mac OS X 10.2.4 o superior SGI IRIX 6.2.15	Microsoft windows 98 o superior
Browser		Microsoft Internet Explorer 4.0 o superior Netscape 7.0 o superior	Microsoft Internet Explorer 5.0 o superior AOL 6.0 o superior Netscape Navigator 4.7x

2.2.1 Elección de plataforma, herramientas y bibliotecas.

Una vez examinada la tabla comparativa se escogió, de las diferentes opciones presentadas en ella, el software adecuado para la aplicación a desarrollar. Nuestra elección final es la siguiente:

Bibliotecas gráficas:

- WildTangent Web Driver.
- Anark Studio

Herramienta de modelado:

- Maya.

¿Porqué Wildtangent?

- Trabaja en PCs con microprocesadores Pentium y AMD.
- Fue el más amigable (no necesita muchos recursos).
- Soporta el lenguaje de programación Java, hasta ahora el más robusto.
- Cuenta con el programa Mutiplayer (jugadores en línea).
- Importa materiales de Maya.
- Trabaja en Explorer o en Netscape Navigator.
- Es gratis, mientras sea para cuestiones académicas.
- Es fácil de instalar.

¿Porqué Anark Studio?

- Requiere menor capacidad de procesamiento.
- Requiere menores recursos de memoria y almacenamiento.
- Permite el desarrollo de aplicaciones con Java Script.
- Permite compresión de archivos.
- Soporta 2D y 3D de forma directa.
- Permite la inclusión de audio, video y gráficos.
- Permite importar elementos de 3D Studio Max.
- Nos fue proporcionado en IIMAS.

¿Porqué Maya?

- Exporta materiales a Wildtangent.
- Trabaja en PCs con microprocesadores Pentium y AMD.
- Fue el más amigable (no necesita muchos recursos).
- Es el líder en el mercado.
- Nos fue proporcionado en IIMAS.

Los programas seleccionados se alojaron en un sistema con las siguientes características:

PC:

- Procesador AMD Athlon™ +2100 a 1.8 GHz.
- 512 MB RAM.
- 80 GB HD.
- Tarjeta gráfica aceleradora OpenGL®.
- 56k MODEM.

Plataforma:

- Microsoft Windows 2000.
- Microsoft DirectX 7.

Browser:

- Microsoft® Internet Explorer® 5.0.
- Netscape Navigator 4.7x, 6.2x.

Lenguaje de programación:

- Java 1.3.1.0 SDK.

Una vez elegido el software adecuado para las aplicaciones a desarrollar, se expondrá en el siguiente capítulo una explicación más extensa del mismo:

- Se apreciara en forma general la herramienta de modelado (Maya) y las bibliotecas de programación (WildTangent Web Driver y Anark Studio).
- Se mostrará la arquitectura de cada una de ellas

3 Descripción detallada de las herramientas, bibliotecas gráficas y cómo se relacionan

En este capítulo se dará una explicación mas detallada de la herramienta de modelado: Maya; y de las bibliotecas de programación: Wildtangent Web Driver y Anark Studio.

3.1 HERRAMIENTA DE MODELADO.



El software Maya se lanzó al mercado en 1998 por la compañía Alias/Wavefron. Tiene más de 40 herramientas para lograr múltiples texturas, efectos, volúmenes, movimientos, creación de personajes y ambientes. Maya ofrece también intercambios de datos con aplicaciones tecnológicas tales como Macromedia Flash, DWG, DXF, Adobe Illustrator, WildTangent Web Driver, Encapsulated PostScript, Scalable Vector Graphics, IGES, OpenFlight, StudioTools™ y OBJ.

El software Maya Unlimited contiene herramientas para la producción de animaciones 3D, modelado avanzado, simulación digital de telas, representación de cabello y peluche, etc. También cuenta con un editor principal, documentación, tutoriales y ejemplos de entrenamiento.

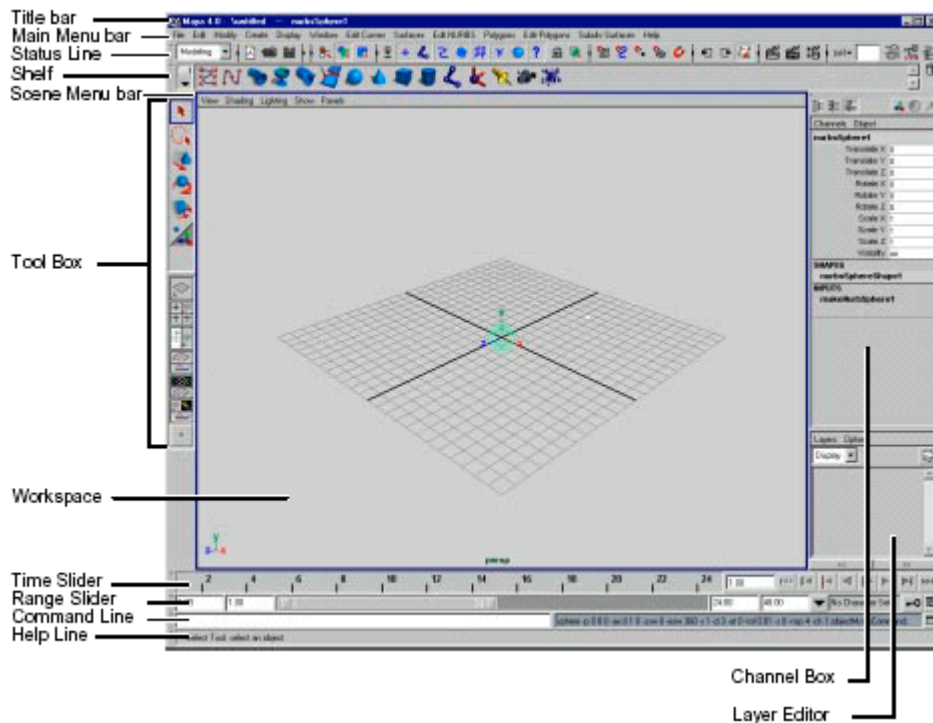
Herramientas

- *ARTISAN*: Interface para hacer “esculpido digital”, pintura y creación de ambientes.
- *PAINT EFFECTS*: Crea detalles de naturaleza en un *canvas* 2D (lienzo de pintura 2D) o en un espacio 3D real, por ejemplo, un árbol en 3D o hacer llover en cuestión de segundos.
- *DYNAMICS*: Sistema de partículas para cuerpos rígidos y suaves de muy alta velocidad (por ejemplo, se les puede agregar gravedad o turbulencia a los objetos animados y se ven obligados a comportarse de acuerdo a las fuerzas de ese nuevo medio ambiente).
- *MODELADO AVANZADO*: Son herramientas extras para el modelado de Nurbs (sistema de modelado de curvas utilizado también en el diseño de automóviles). Los modelos encajan de manera perfecta. Se pueden hacer efectos de inyección de plástico, huellas sobre metal, objetos transparentes y piel en articulaciones.
- *RENDERING*: Representaciones con efectos visuales avanzados y la tecnología RFI (Representación Fotorealístico Interactivo).

- **ANIMATION:** Posiciona cuadros claves (*keyframing*), hace expresiones, deforma y crea piel sobre personajes (*skinning*).
- **LIVE:** Coteja los movimientos precisos que permiten ver al personaje en 3D en forma real , por ejemplo puede integrar a Godzilla en el periférico.
- **CLOTH:** La solución más rápida y precisa para simular una enorme variedad de telas y otros objetos textiles (desde seda que se estira hasta plástico). Con esta herramienta se diseñó todo el guardarropa de Stuart Little y Shrek.
- **FUR:** Para crear pieles de animales, peluche y cabello. Con Fur cada superficie o grupo de superficies puede tener diferentes tipos de pelambre. Por ejemplo, es posible crear un león que tenga pelitos cortos y suaves alrededor de su cabeza, con melena larga y densa, pestañas chinas y largas y algunos bigotes delicados. Todos estos atributos manejados con un control independiente (y cada superficie puede tener millones de pelos independientes).
- **MEL (MAYA EMBEDDED LANGUAGE):** es el lenguaje propio de Maya. Es una ventaja para los usuarios porque pueden bajar de Internet *scripts* (pequeños pedazos de códigos) gratuitos. Es para crear, por ejemplo, rocas, tornados, iluminación realista, etc.

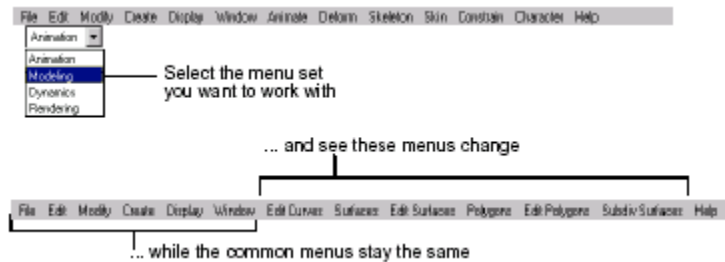
Ventana principal

Es una interfaz de edición que contiene todos los elementos que se necesitan para desarrollar modelados, animaciones y efectos visuales.



La ventana Principal consta de los siguientes elementos:

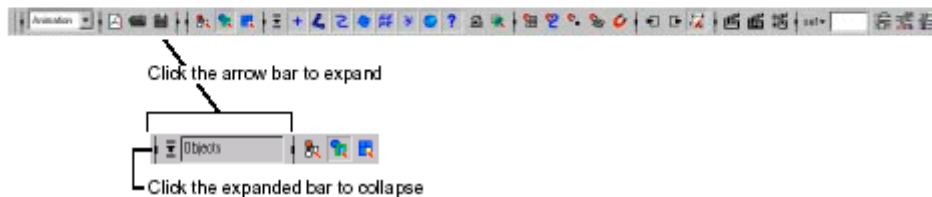
- **MAIN MENU BAR.** Los menús en maya se agrupan en menú *sets* (sistemas de menú) . Cada sistema de menú corresponde a un módulo de software: animación, modelado, dinámicos y representación. Además se cuenta con otros dos módulos: *Cloth* y *Live*.



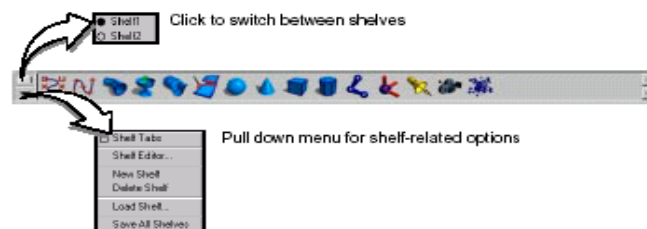
- **STATUS LINE.** La línea de estado tiene una variedad de comandos, usados sobre todo para modelar. Por ejemplo, el grupo central de botones se utiliza para seleccionar objetos y componentes.



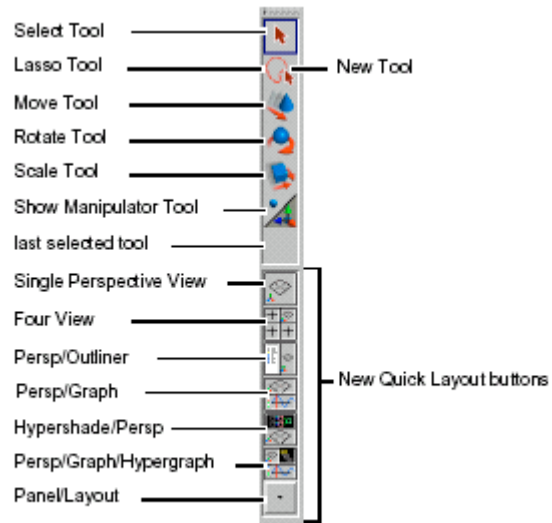
For better organization, the buttons are broken into groups that you can expand and collapse, as shown in the following illustration.



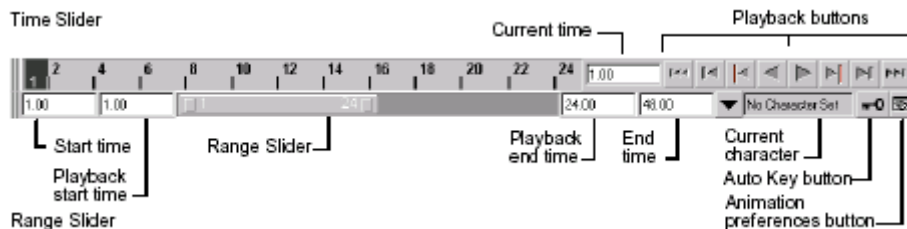
- **SHELF.** El estante es una colección de herramientas y de comandos que se pueden particularizar para cumplir con necesidades específicas. Creando estantes particulares, se puede organizar acciones comúnmente usadas y herramientas en grupos. Por ejemplo, se puede crear estantes de modelado o de animación con las herramientas apropiadas y de acciones para cada opción. También se puede almacenar la misma herramienta más de una vez, pero con diversos ajustes.



- **TOOL BOX.** Contiene las herramientas comunes así como la herramienta seleccionada anteriormente, y los iconos para las vistas y las presentaciones que cambian. El espacio en la caja de la herramienta que se asigna a la "herramienta seleccionada anteriormente" exhibe el icono de la herramienta ocupada que se seleccionó de un menú o de un estante.



- **WORKSPACE.** El propósito principal del espacio de trabajo es tener una visión de la escena. Se puede exhibir varios editores y arreglar paneles del espacio de trabajo en diversas presentaciones.
- **LAYER EDITOR.** Agrupa objetos para que se puedan ocultar y utilizarlos como plantilla, o presentándolos en pasos separado. El editor proporciona los controles principales para crear capas, agregando objetos para estas mismas y haciéndolas visibles o invisibles.
- **CHANEL BOX.** La mayoría de los elementos de la interfaz de Maya son comunes a los paquetes de software 3D, pero el *Chanel Box* es una característica única y de gran alcance. Da acceso directo a los bloques construidos de Maya: cualidades y nodos. Específicamente, demuestra las cualidades *keyable*, también conocidas como canales.
- **TIME SLIDER AND RANGE SLIDER.** Los dos controles permiten manipular los marcos en la animación. El control de tiempo incluye los botones del aparato de lectura (también llamados controles del transporte) y el indicador del tiempo actual. El control de rango incluye el comienzo y el final de la línea de tiempo, los tiempos del comienzo del aparato de lectura y del final, la barra del control de rango, el botón auto-llave, y las preferencias de la animación.



- **HELP LINE.** Se puede observar descripciones, instrucciones y otra información útil.
- **COMMAND LINE.** Aloja pedazos de código del lenguaje MEL.
 - El lado izquierdo es donde se escriben comandos del lenguaje de programación de Maya.
 - Las respuestas del sistema se despliegan de la mitad hacia la derecha, como: mensajes de error o mensajes de advertencia.



El Acceso API

Con el API es posible que el usuario entre a los datos internos de Maya y pueda modificar o crear nuevas herramientas. A través de API se pueden crear *plug-ins* para extender las capacidades del sistema.

Manuales y Tutoriales

Incluye un disco de documentación llamado *Print Library*. Este documento contiene toda la información de la tecnología Maya: instalación de Maya, introducción a Maya, usando Maya, etc.

3.2 BIBLIOTECA DE PROGRAMACIÓN.



WildTangent fue fundada en 1998 por el creador de DirectX, Alex ST John, y el matemático, Jeremy Kenyon. La compañía se enfoca en el diseño y promoción de la tecnología de juegos en línea. Su nuevo controlador exclusivo para Web permite jugar juegos en línea sin tener que descargarlos e instalarlos. Esta tecnología usa procedimientos avanzados de compresión para presentar juegos con mejores gráficos y sonido. La flexibilidad con los archivos es tal que crea “un ancho de banda virtual” para que estos se transporten sobre una conexión en Internet rápidamente.



El *plug-in* del controlador Web esta por debajo de 1 MB. También permite que los desarrolladores usen una amplia gama de lenguajes de programación, incluyendo Java, JavaScript, Visual Basic, VB Script, C y C++.

La compañía lanzó la versión 1.0 de su controlador para Web y el SDK a principios de 1999, y a finales del 2002 surgió la versión 3.0.

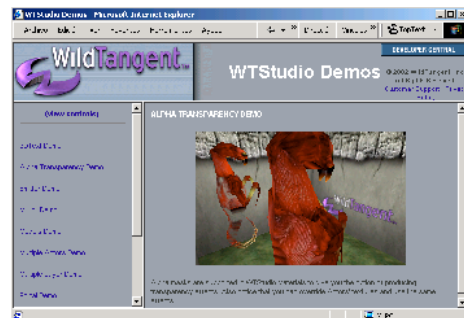
EL WEB DRIVER 3.0 SDK.

Contiene herramientas, documentación, demostraciones y tutoriales, para ayudar al aprendizaje y al desarrollo de aplicaciones. Además contiene el *Web Driver* (3.0.0.173) y el *Update* (1.5.3.018) para utilizar la tecnología Wildtangent.

Demostraciones

Nos muestra algunas capacidades que puede hacer la tecnología Wildtangent.

- Todos los ejemplos son proyectos realizados con WTStudio.
- Algunos de los ejemplos tienen *scripts* adicionales para crear actividad en algunos elementos de la escena.



Documentación

Este SDK contiene completa documentación de la tecnología Wildtangent Web Driver.

- Como crear proyectos en diferentes lenguajes de programación como: Java, Flash, VB Script, JavaScript, C, C++ y Visual Basic.
- También encontramos información sobre formatos de archivos de sonido, mapas de bits, modelos 3D y datos, que soporta el Web Driver 3.0.
- Tips para usar el Web Driver de forma eficiente. Como escribir un código bien construido, estable, robusto, y cuya aplicación sea la mas agradable posible.
- *API REFERENCE*. Una completa lista de toda la funcionalidad del Web Driver.

Ejemplos

Nos provee de código para reutilizarlo en proyectos desarrollados por nosotros mismos.

Se encontrará cuatro tipos diferentes de código:

- **TEMPLATES:** Actúan como un punto de comienzo de los diferentes tipos de proyectos que se puedan emprender con el controlador Web.
- **PROJECTS:** El origen del código de los ejemplos.
- **RELEASES:** Liberación de aplicaciones completas con origen y activos media.
- **MISCELLANEOUS EXAMPLES:** Contiene una variedad de aplicaciones tal como audio-visualizadores y niveles Fly-Throughs.

Laboratorios

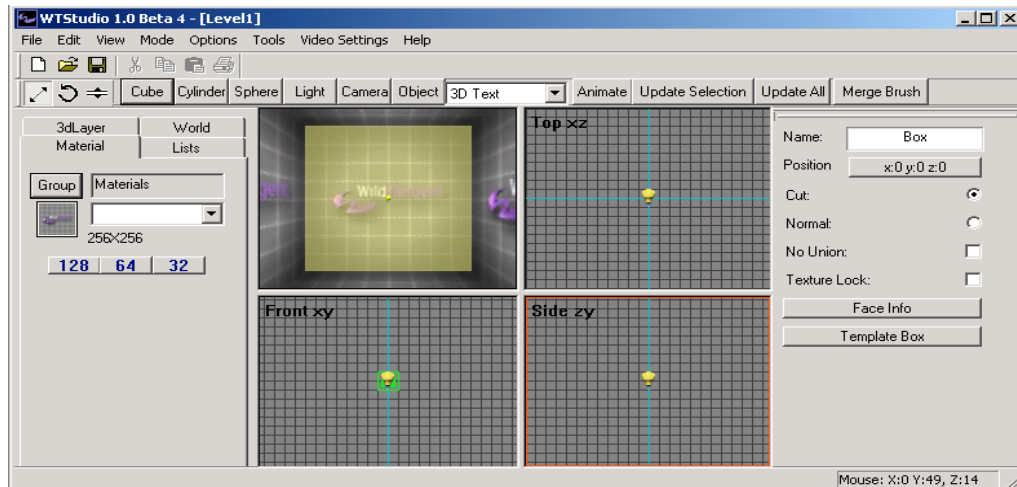
Nos dan una explicación visual de desarrollos complejos con el manejador Web, mostrando como ocupar los diferentes elementos del API para crear escenarios, efectos de luz, colisiones, sombras, reflejos, portales, etc.



Herramientas

- **IMPORTADOR DE 3DS MAX & MAYA.** Importa geometría, materiales, y animaciones de 3ds max o Maya:
 - Animación *Smooth-skinned*.
 - Generación de textura automática.
 - *Bone Promotion*
 - Ambiente de mapeo.
- **MICROSOFT FEDIT.** El editor Force es una herramienta que adiciona la funcionalidad de DirectX para diseñar y probar efectos de dispositivos force-feedback.
- **WTPackage DOCUMENTATION.** Esta herramienta nos sirve para crear documentación particular por medio del programa wtpks.exe.
- **WTPKG.** Utilidad de empaquetamiento en el cual se aloja el proyecto diseñado, incluyendo visualizadores y Screen Savers.
- **WILDCOMPRESS.** Una herramienta que sirve para comprimir y encriptar imágenes, modelos, sonidos, y archivos de datos.
- **WTSTUDIO BETA 4.** Es un editor estándar para la creación de escenas (geometría BSP) para el proyecto a diseñar.
 - Alumbrado estático y dinámico de todos los tipos –ambiental, de punto, de manchas, direccional y paralela.

- Geometría en tiempo real y actualización de alumbrado.
- Materiales multicapa.
- *Mip-mapping* –desarrollo optimizado.
- Incluye objetos como: actores, texto 3D, emisores y sonido 3D.
- Actores, usando alguna estructura esquelética que pueda compartir movimiento.
- Niveles progresivos automáticos de detalle para actores



Entrenamiento

Esta sección incluye how-to's y tutoriales cubriendo algunos tópicos de desarrollo en el controlador Web.

- *HOW-TO'S*. Nos muestra características específicas de una aplicación Web Driver que puede ser implementada.
- *TUTORIALES*. Nos proveen de instrucciones para llevar a cabo un proyecto.

Multimedios

Incluye ejemplos de aplicaciones multimedia y también cuenta con algunos archivos 3ds max.

ANARK STUDIO



Anark Studio es un software multimedia que da al usuario la habilidad de crear presentaciones interactivas 3D. Proporciona el poder de combinar gráficos 3D con gráficos 2D, video, texto y datos. Cuenta con capacidades robustas y flexibles de

importación y exportación de objetos que permiten integrar elementos de otras herramientas en el mercado.

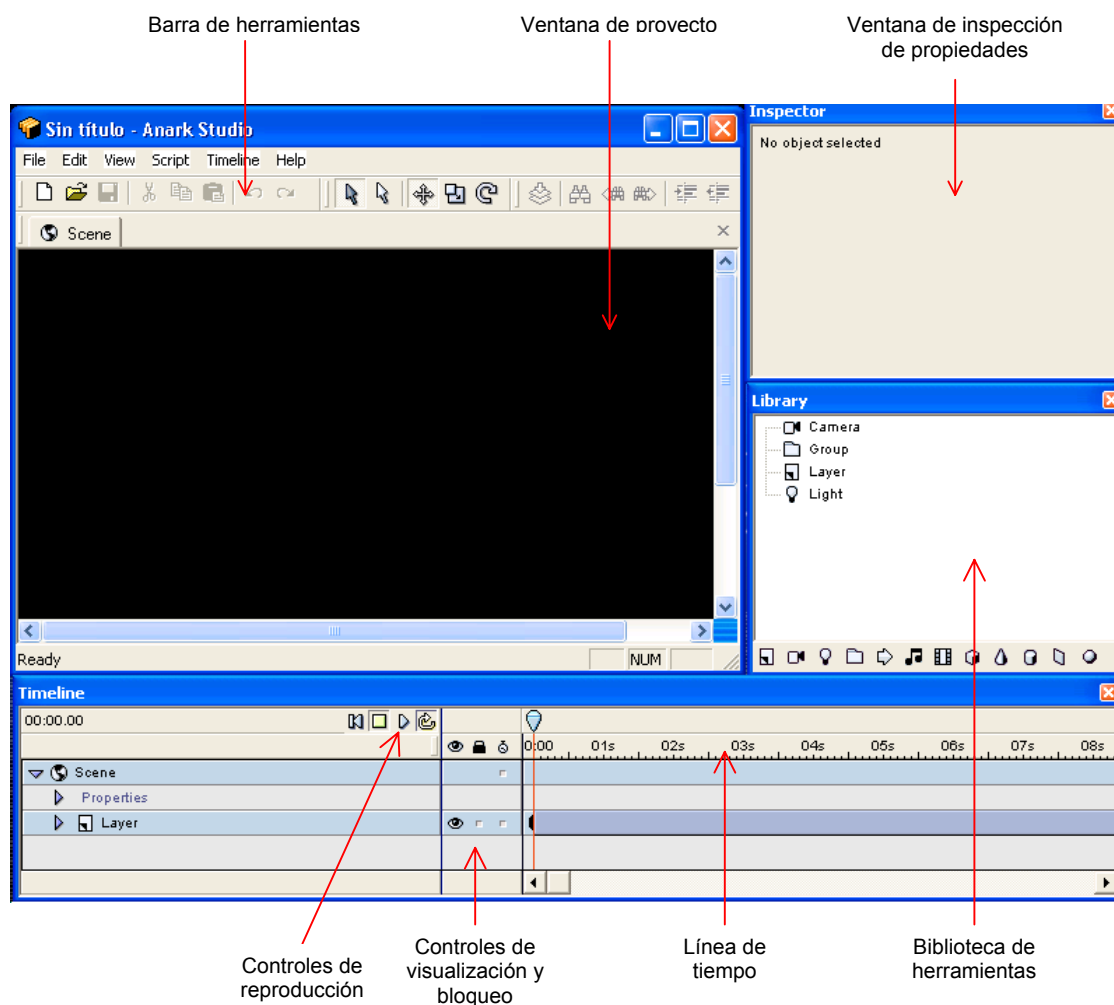
Setup, instalación y configuración

La instalación es sumamente sencilla y rápida. La ayuda está disponible en un manual técnico y en línea. Asimismo, Anark Studio cuenta con soporte y foros técnicos en la web. La desinstalación se hace a través del panel de control de Windows con el menú de Agregar/Quitar programas.

La documentación incluye un manual escrito que cuenta con glosario, referencia, tutoriales y ligas a varios sitios en el web para ayuda.

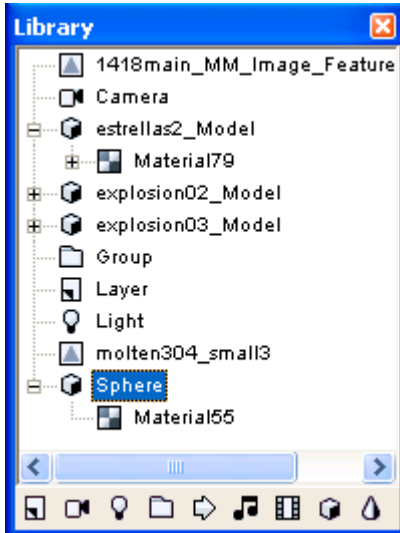
Interfaz

Anark fue diseñado con un espacio de trabajo flexible, comportamientos del estilo de arrastrar y pegar, y una línea de tiempo estándar. La interfaz del usuario consiste en una ventana de proyecto, una biblioteca de herramientas, una ventana de inspección de parámetros y una línea de tiempo.



Biblioteca de herramientas

La biblioteca de herramientas contiene los objetos que se pueden agregar a un proyecto, para ello, sólo es necesario colocarlos sobre la línea de tiempo. Desde la biblioteca se tiene acceso a los archivos y a los comportamientos que aparecen en el proyecto. Al seleccionar un objeto dentro de la biblioteca se puede cambiar sus parámetros a través de la ventana de Inspección, la cual se encuentra justo encima de la biblioteca de herramientas.



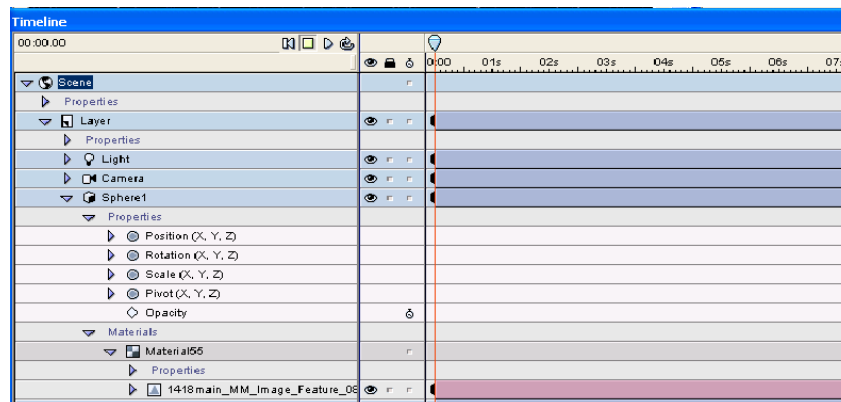
Desde la biblioteca se pueden agregar gran variedad de objetos y comportamientos al proyecto, algunos de estos objetos son: vistas de cámara, luces, grupos, comportamientos, música, video, cubos, conos, etc. Cuando en un proyecto se ha agregado algún objeto, éste aparecerá listado dentro de la biblioteca y desde él podrán crearse múltiples instancias del mismo tipo de objeto y agregarse al proyecto.

Asimismo, teniendo a la vista la biblioteca y dando un click al botón derecho del mouse se puede acceder de forma rápida al menú de Importación de recursos, desde donde se pueden incluir comportamientos, imágenes, sonido y video.

La línea de tiempo

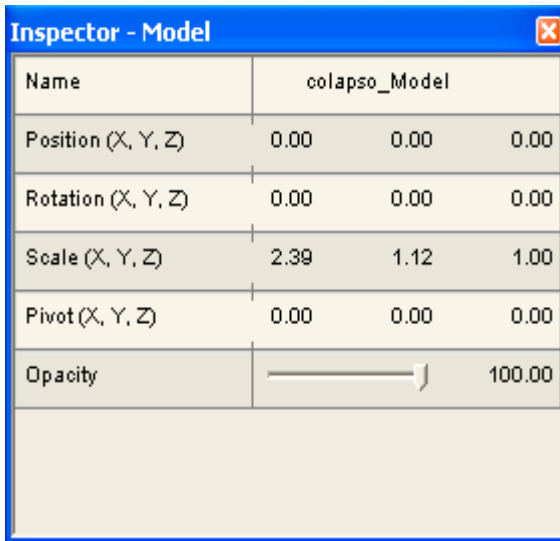
Una vez que se ha incorporado al proyecto algún elemento (esfera, cono, imagen, etc.) es posible crear procedimientos de animación de los mismos, para ello se utiliza la línea de tiempo.

La línea de tiempo es donde se agregan todos los elementos que se desean utilizar para crear la presentación. La duración de cada elemento puede modificarse y también pueden agregarse y quitarse *keyframes*.



Ventana de inspección de propiedades

Al seleccionar un elemento dentro de la Biblioteca es posible modificar los parámetros del mismo a través de la ventana de inspección de propiedades.



La ventana de inspección de propiedades alberga toda la información relevante perteneciente a cada uno de los elementos que integran el proyecto. Por ejemplo, si se tiene algún archivo de 3d Max sobre la línea de tiempo y lo selecciona la ventana de inspección de propiedades detallará el nombre del archivo, las coordenadas X Y y Z de su posición, la información referente al grado de rotación, pivote y escala así como las características de opacidad. Es desde esta ventana desde donde se pueden modificar los valores de los parámetros de los archivos que se hayan cargado en el proyecto.

Cuando se agrega algún elemento al proyecto, éste se incluye dentro de una jerarquía. La escena es el objeto raíz en la jerarquía de Anark Studio. Todas las capas se ligan directamente a la escena. Los únicos objetos, además de las capas, que pueden ligarse directamente a la escena son comportamientos, sonido y música.

Escena

Las características de la escena son: Color de fondo, *Render detail* y *loop*. La propiedad *Render detail* establece el porcentaje al cual la presentación será desplegada en la ventana del cliente de Anark Studio. Se utiliza para ahorrar memoria y mejorar la velocidad de despliegue pero reduce la resolución del contenido gráfico. El valor de la propiedad *loop* es de tipo booleano que establece si una vez mostrada la presentación ésta volverá a mostrarse dentro de la ventana del cliente de Anark Studio.

Capas

Las capas son un tipo especial de grupo que se usa para organizar el contenido de la escena y para controlar ciertas propiedades de todos los objetos ligados a la misma. Dentro de la jerarquía de objetos de Anark Studio son el segundo nivel, lo que significa que todos los demás objetos (a excepción de música y comportamientos) deben estar ligados directamente o indirectamente a una capa. Un proyecto puede incluir una o múltiples capas. Cuando una capa se agrega a un proyecto ésta tiene ya una luz y una cámara por default incluidas a la misma.

Las propiedades principales de las capas son: nombre, posición X (margen de la capa respecto a la parte superior de la escena), posición Y (margen de la capa respecto al lado izquierdo de la escena), ancho y alto (especifica el porcentaje que la capa cubrirá con

respecto al ancho de la escena), opacidad (ajusta la transparencia de todos los objetos ligados a la capa, si el valor de esta propiedad se establece en 0 la capa se dibujará, pero será invisible), color (es el tono que se dibujará en los espacios vacíos de la capa), uso de color (propiedad booleana que indicará sí se utilizará o no el color de fondo establecido).

Cámaras

Las cámaras de Anark Studio controlan la forma en que una se ve una capa de forma similar a cómo una cámara de video funciona en el mundo real. Se pueden utilizar para tomar acercamientos o alejamientos de la escena o para enfocarse en determinados objetos. Algunas de sus propiedades pueden animarse de forma que la vista de una escena cambie a lo largo de la presentación.

Cada nueva capa que se agregue al proyecto tiene una cámara ligada a si misma y de la cual pueden modificarse sus propiedades. También es posible agregar nuevas cámaras a la capa, pero sólo una puede permanecer activa en un mismo tiempo. Es decir, es posible activar y desactivar una o mas cámaras pero nunca puede permanecer mas de una activa simultáneamente.

De esta forma, cada capa se observa a través de su propia cámara y la imagen de la escena que el usuario observa es una composición que incorpora todas las vistas de las cámaras activas.

Entre las propiedades principales de los objetos del tipo cámara están: nombre, posición, rotación, escala (establece la escala para cualquier objeto hijo ligados a la cámara), pivote (punto que será el centro de rotación de la cámara), campo de visión (controla el espacio de la escena que será visible así como la perspectiva, de forma similar a cómo funciona un lente de cámara), inicio de niebla (establece la distancia radial desde la cámara a partir de la cual el efecto niebla comienza), fin de niebla (establece la distancia radial desde la cámara en la cual la niebla se hace totalmente opaca), color de niebla, activación de niebla (booleano que establece si el efecto niebla se aplicará o no).

Luces

Se utilizan para crear luz natural (similar al sol), luz artificial (similar a una lámpara) o efectos de iluminación estilizados (como los utilizados en una obra de teatro). Cuando una capa se agrega ya tiene definido un objeto de este tipo, pero se pueden agregar tantas como se desee, sin embargo el límite de luces activas simultáneamente es de ocho por cada capa.

Las propiedades principales de las luces son: nombre, posición, rotación, escala, pivote, tipo de luz (puede ser de tres tipos: De punto, emite luz a partir de un punto en todas las direcciones de forma equilibrada; Direccional, irradia luz con la misma intensidad en determinada dirección con rayos paralelos entre sí, el ángulo lo de emisión lo determina la propiedad de rotación; *Spot*, emite un rayo de luz en forma de cono a partir de un punto), color difuso (ajusta el color de los rayos directos de luz), *specular color* (ajusta el color de iluminación de la escena), color ambiental (ajusta el color de la iluminación indirecta de la escena), rango (establece el alcance del radio del efecto de la luz), brillantez (ajusta el

nivel máximo de luz), ángulo interior y exterior del cono, desvanecimiento lineal (ajusta la forma y velocidad con la cual la luz pasará de su origen a la obscuridad (a un valor mayor, el desvanecimiento será más rápido), desvanecimiento exponencial (utiliza una fórmula exponencial y el valor proporcionado para controlar la forma en la cual la luz se desvanece desde su origen hasta la obscuridad).

A los proyectos de Anark Studio se les pueden incorporar elementos desarrollados con otras aplicaciones, entre ellos están los modelos, los materiales o texturas, imágenes, video y música. Los modelos son elementos desarrollados con aplicaciones como 3D Max o Maya.

Los materiales se utilizan para crear la apariencia superficial de un modelo. Por este medio se puede controlar la brillantez, la opacidad. Los materiales son el único tipo de objetos que no tienen barra de tiempo asociada ya que la duración del material es siempre la misma que el objeto al cual se liga.

Scripts

Además del entorno GUI, Anark Studio proporciona capacidades de *scripting* para todos aquellos que deseen involucrarse más profundamente en el desarrollo de animaciones. Anark Studio incluye un amplio rango de comportamientos en *scripts* que pueden ser usados para agregar interactividad a la presentación además de que permite construir *scripts* de comportamientos propios.

Comportamientos

Anark Studio aplica interactividad a través de una serie de comportamientos que se arrastran y pegan basados en Java Script. Estos incluyen desde eventos del mouse, eventos de pausa y juego y una variedad de efectos no estándares, tales como animaciones. Estos comportamientos incluidos pueden usarse para objetos tales como botones y ligas y son personalizados ajustando las propiedades del comportamiento a través del visor de propiedades.

3.3 CONCLUSIONES.

Cualquier medio que ayude a construir el conocimiento es útil, la cuestión es empalmar adecuadamente estas herramientas. El software de multimedia y de modelado pueden ser claves para mejorar el aprendizaje de estos temas.

Maya ofrece una herramienta visual, comercial robusta y extensamente usada para los efectos, con una infraestructura rica para la extensión y el arreglo, para requisitos particulares, como el desarrollo de juegos, gráficos de programas, creación de contenidos web, visualización del diseño y producción de películas.

WildTangent desarrolla nuevas tecnologías para navegador, produce juegos con mejores gráficos y sonido; hay una sinergia con Maya: exportando materiales, geometrías y movimientos.

Anark Studio, por su parte, permite crear animaciones de forma relativamente rápida y sencilla, al poner a disposición del usuario una gran variedad de objetos, comportamientos y propiedades, que en combinación permiten crear efectos llamativos e ilustrativos.

En el siguiente capítulo entraremos en el diseño y desarrollo de dos aplicaciones demostrativas enfocadas en fenómenos físicos.

4 Desarrollo de aplicación demostrativa

Como aplicaciones demostrativas sobre temas de física se desarrollarán dos temas: tiro parabólico y hoyos negros. Ambos temas presentan características diferentes, el primero es un fenómeno que podemos observar a simple vista en nuestro mundo cotidiano, mientras que el segundo se basa en observaciones y deducciones; hasta el momento existen evidencias teóricas y observaciones indirectas que indican su existencia, pero no es un fenómeno que pueda ser visto y apreciado a simple vista.

Por sus propias características, la exposición de cada tema se abordará de forma diferente. Para el tiro parabólico se desarrollará un juego en línea, mientras que para el tema de hoyos negros se desarrollarán algunas animaciones.

4.1 TIRO PARABÓLICO

Pensando en un fenómeno físico en cuyo contexto se pueda tener una interactividad entre dos usuarios y su interfaz muestre un ambiente agradable, colorido y divertido, se decidió adaptar el concepto de tiro parabólico a una aplicación demostrativa.

Una vez elegido el fenómeno físico se diseñara la interfaz gráfica, se mostrarán los pasos a seguir para desarrollarla y cómo plasmar algoritmos en ella, de tal forma que pueda existir una comunicación entre dos usuarios en línea.

La programación de la interfaz gráfica se hará en HTML, Java y en WildTangent Web Driver.

El modelado de los elementos que intervienen en la aplicación se desarrollará en Maya.

4.1.1 OBJETIVOS DE LA APLICACIÓN

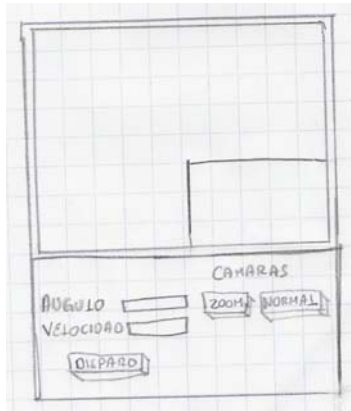
- Adaptar el fenómeno físico: Tiro Parabólico; en forma eficiente.
- Demostrar utilidad de bibliotecas y herramientas.
- Desarrollar la aplicación en un contexto de jugadores de línea (Internet).
- Cuantificar el tiempo y la velocidad de aprendizaje.

4.1.2 DISEÑO

A) VENTANA DE LA APLICACIÓN.

Contará con:

- ✓ Una pantalla principal con un portal en el extremo inferior derecho.
- ✓ Dos áreas de texto para introducir el ángulo y la velocidad de disparo.
- ✓ Un botón para efectuar el tiro.
- ✓ Dos botones para controlar el despliegue de la imagen de dos cámaras en la pantalla.



Primer bosquejo

B) PANTALLA PRINCIPAL Y PORTAL

Tendrá:

- ✓ Un escenario cuyo campo este rodeado de montañas.
- ✓ Un cielo con nubes, las cuales estén en movimiento.
- ✓ Un piso tapizado de pasto.
- ✓ Efectos visuales como explosiones, sombras, etc.
- ✓ Dos cañones, cuya ubicación en el escenario será aleatoria.



Segundo bosquejo

C) MODO DE JUEGO

Se convino representar el fenómeno físico como un “tiro al blanco”.

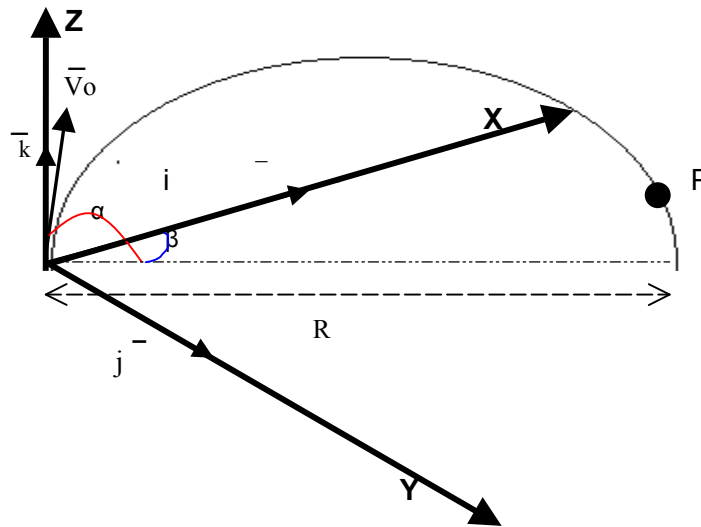
- ✓ Se tendrán dos jugadores.
- ✓ Cada jugador tendrá su ventana de aplicación.
- ✓ Cada jugador manipulará sólo un cañón.
- ✓ Tendrán comunicación en línea.
- ✓ El usuario que logre atinar al otro será el ganador.

4.1.2.1 Modelado geométrico, apariencia física y comportamiento

Partiendo del diseño, se analizará y empleará conceptos matemáticos para los requerimientos de la aplicación. Se esquematizará este contexto de modo que sea más fácil su solución.

Comenzaremos analizando el Tiro Parabólico:

Basándose en el principio de Independencia de los movimientos en direcciones perpendiculares, todos los movimientos se pueden analizar descomponiendo los parámetros característicos (posición, velocidad, aceleración, etc.) en las direcciones de los ejes coordenados.



Donde:

- i Componente X.
- j Componente Y.
- K Componente Z.
- $V_0(v_{0x}, v_{0y}, v_{0z})$ Velocidad inicial del proyectil.
- α Angulo de V_0 con respecto al plano XY.
- β Angulo de V_0 con respecto al plano ZX.
- $P(x,y,z)$ Posición del proyectil.

Por consiguiente:

Coordenada x

$$a_x = 0$$

$$v_x = v_{0x} = v_0 (\cos \alpha)(\sin \beta)$$

$$x = (x_0 + v_{0x} t)$$

$$x = x_0 + t v_0 (\cos \alpha)(\sin \beta)$$

Coordenada y

$$a_y = 0$$

$$v_y = v_{0y} = v_0 (\cos \alpha)(\cos \beta)$$

$$y = y_0 + v_{0y} t$$

$$y = y_0 + t v_0 (\cos \alpha)(\cos \beta)$$

Coordenada z

$$a_z = -g k$$

$$v_{0z} = v_0 \sin \alpha$$

$$v_z = v_{0z} + g t$$

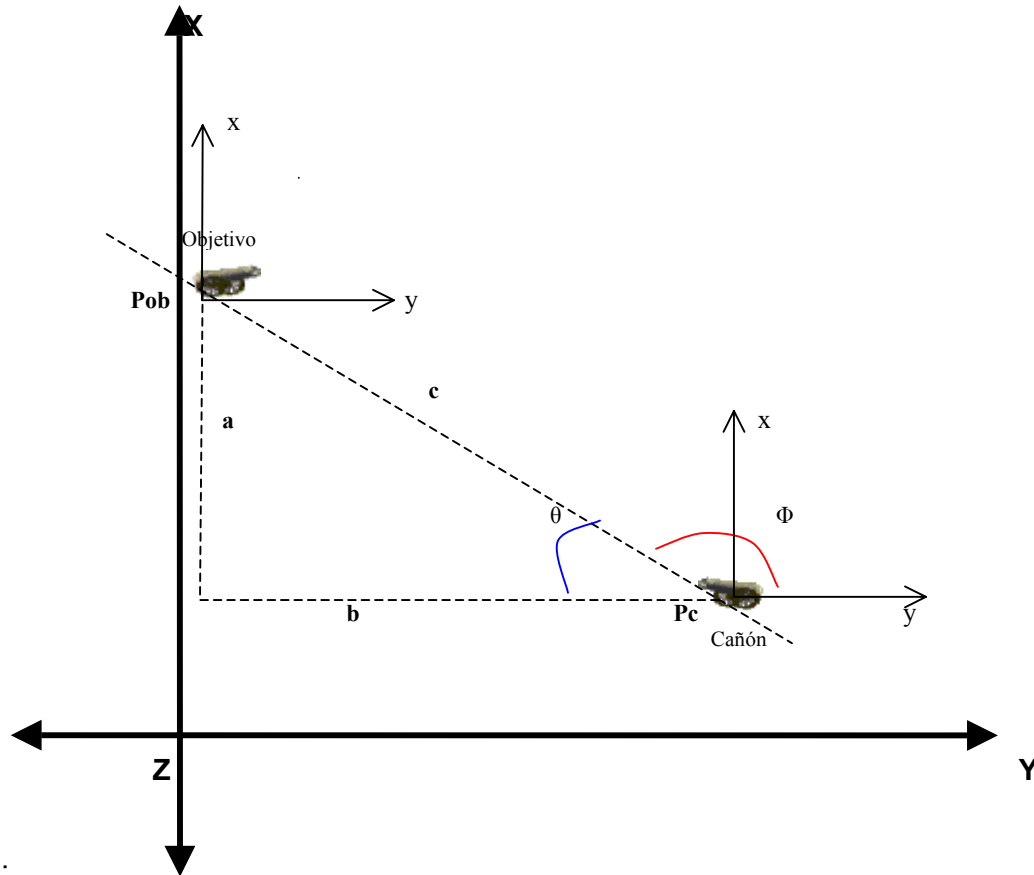
$$Z = Z_{0z} + v_{0z} t + \frac{1}{2} g t^2$$

$$z = y_0 + v_0 \sin \alpha t + \frac{1}{2} g t^2$$

Los valores que nos interesan son las coordenadas "x", "y" y "z" de la partícula en movimiento. La adecuación de estas fórmulas al programa será indispensable para la aplicación.

Como se mencionó anteriormente se necesita de un cañón, el cual va a estar apuntando a un objetivo (otro cañón), para que esto se logre, el giro del mortero debe ser automático. Se recurrió a analizar el problema en un contexto matemático, específicamente: Distancia entre dos puntos y Trigonometría.

Veamos el siguiente razonamiento, apoyándonos en la gráfica:



Donde:

- P_{ob} Posición del objetivo.
- P_c Posición del cañón.
- θ Angulo de complemento.
- ϕ Angulo de giro del cañón con respecto al eje Z.
- c Distancia entre el cañón y el objetivo.
- a Distancia en X.
- b Distancia en Y.

Razonamiento:

1. Se necesita el valor del ángulo Φ para que el cañón gire hacia la posición del objetivo.
2. Una vez conocida la ubicación del objetivo, se calcula el ángulo θ

$$\begin{aligned} a &= P_{obx} - P_{cx} \\ b &= P_{oby} - P_{cy} \\ \tan \theta &= a/b \rightarrow \theta = \tan^{-1} a/b \end{aligned}$$

3. Finalmente para obtener el ángulo Φ , nos apoyamos en la siguiente ecuación:

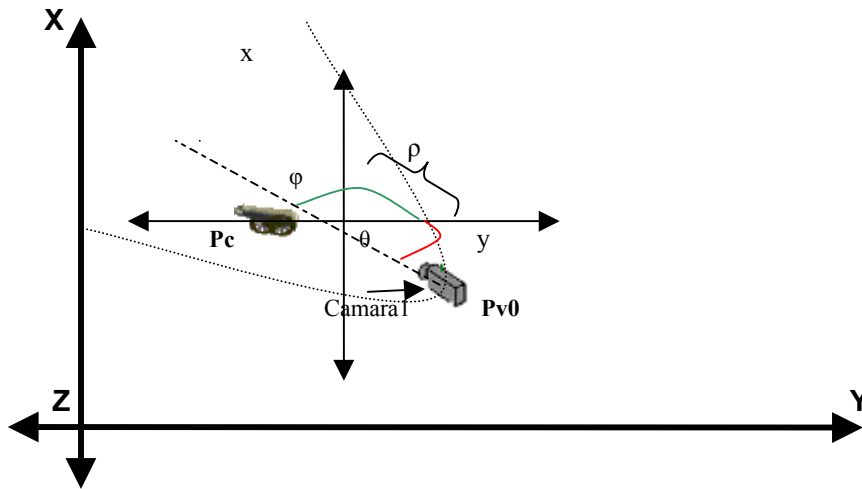
$$\Phi + \theta = 180^\circ \rightarrow \Phi = 180^\circ - \theta$$

$$\therefore \Phi = 180^\circ - \tan^{-1} ((P_{obx} - P_{cx}) / (P_{oby} - P_{cy}))$$

Para incrementar la interactividad con el escenario se decidió colocar tres cámaras.

Cámara 1

La primera cámara va a estar ubicada detrás del cañón. Para lograr esto se recurrió a conceptos matemático como: Coordenadas Polares y Trigonometría. Veamos:



Donde:

- Pc Posición del cañón.
- Pv0 Posición de la primera cámara.
- θ Angulo de giro de la primera cámara con respecto al eje Y.
- Φ Angulo de giro del cañón partiendo del eje Y.
- ρ La distancia entre el cañón y la cámara.

Razonamiento:

1. Para obtener la posición de la primera cámara, se necesita tener las coordenadas de la ubicación del cañón y su ángulo de giro Φ . Primero obtenemos el ángulo θ :

$$\theta = \Phi + 180^\circ$$

2. Una vez obtenido, se calcula la posición de la primera cámara con respecto a la posición del cañón, apoyándonos en el concepto de Coordenadas Polares:

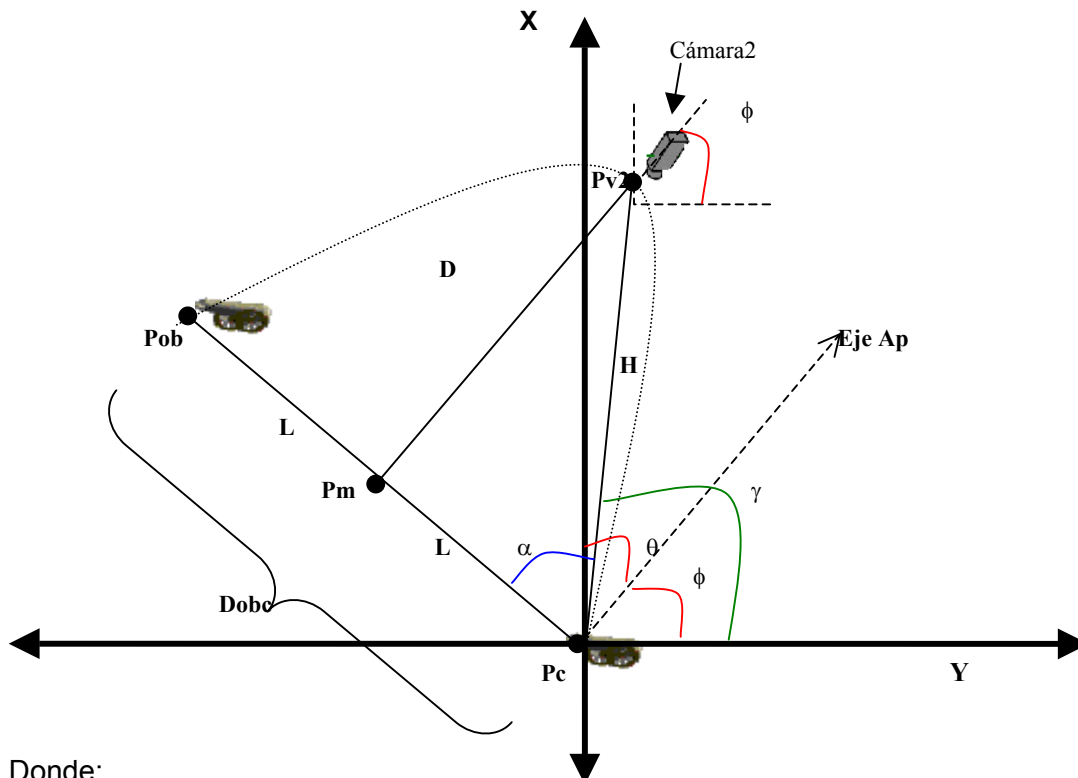
$$X_{vo} = \rho \cos \theta + P_{cx}$$

$$Y_{vo} = \rho \text{ Sen } \theta + P_{cy}$$

El valor de ρ depende de lo alejado que se quiere estar del cañón.

Cámara 2.

Con el mismo concepto matemático, la segunda cámara se ubicará perpendicularmente al cañón y al objetivo. Para esto se tiene que calcular la ubicación (P_v) y el ángulo (ϕ) de enfoque de la cámara. Con la posición de ambos objetos se puede lograr esto, basado en el siguiente razonamiento:



Donde:

- Dobjc Distancia entre el objetivo y el cañón.
- L Distancia media entre el objetivo y el cañón.
- D Distancia entre la cámara y el punto medio.
- H Distancia entre la cámara y el cañón.
- Pobj Posición del objetivo.
- Pm Posición del punto medio.
- Pc Posición del cañón.
- Pv2 Posición de la segunda cámara.

- Eje Ap Eje de apoyo.
- α Angulo entre L y H.
- θ Angulo entre Eje Ap y Eje X.
- ϕ Angulo de giro de la cámara.
- γ Angulo entre H y Eje Y.

Razonamiento:

1. Primero hay que encontrar el ángulo ϕ . Para esto se calcula L.

$$D_{obc} = \sqrt{(P_{ob} - P_c)^2}$$

$$L = D_{obc} / 2 \rightarrow L = (\sqrt{(P_{ob} - P_c)^2}) / 2$$

2. Una vez encontrado L y proporcionado el valor de D, podemos calcular el ángulo α .

$$\tan \alpha = D/L \rightarrow \alpha = \tan^{-1} D/L$$

3. En consecuencia encontraremos el ángulo deseado ϕ .

$$90^\circ = \phi + \alpha \rightarrow \phi = 90^\circ - \alpha \rightarrow \therefore \phi = 90^\circ - \tan^{-1} D/L$$

4. Ahora para calcular la posición de la cámara se necesita encontrar el valor del ángulo γ y el valor de H.

Se aprecia que:

$$90^\circ = \alpha + \theta \rightarrow \theta = 90^\circ - \alpha$$

$$\gamma = \phi + \theta \rightarrow \gamma = \phi + 90^\circ - \alpha$$

$$\gamma = (90^\circ - \alpha + 90^\circ - \alpha) \rightarrow \gamma = (180^\circ - 2\alpha) \rightarrow \therefore \gamma = (180^\circ - 2(\tan^{-1} D/L))$$

5. Ahora por el Teorema de Pitágoras encontramos H.

$$H = \sqrt{(D^2 + L^2)}$$

6. Y así se encuentra la posición $P_{v1}(X_{v2}, Y_{v2})$ de la segunda cámara.

$$X_{v2} = H \sin \gamma$$

$$Y_{v2} = H \cos \gamma$$

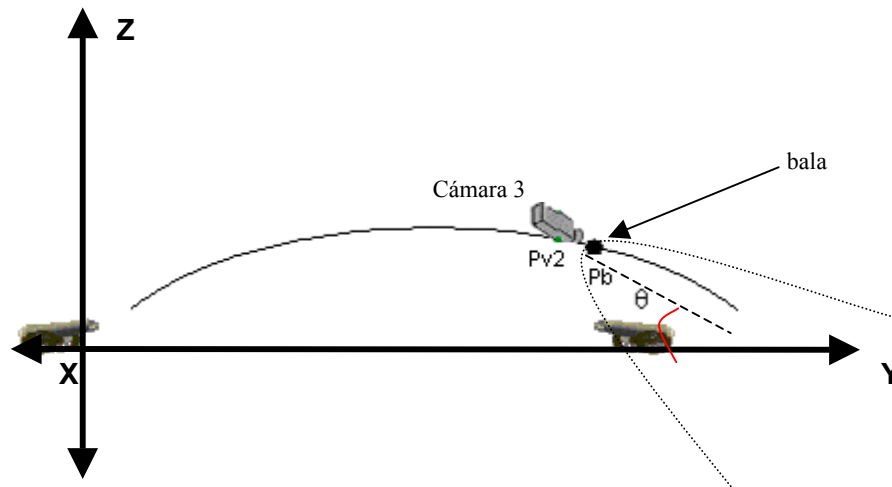
$$\therefore X_{v2} = \sqrt{(D^2 + L^2)} \sin (180^\circ - 2(\tan^{-1} D/L)) + P_{cx}$$

$$Y_{v2} = \sqrt{(D^2 + L^2)} \cos (180^\circ - 2(\tan^{-1} D/L)) + P_{cy}$$

Como vemos el valor de D es indispensable. Este dato será proporcionado dependiendo de lo lejano que se encuentre el cañón y el objetivo.

Cámara 3.

La tercera cámara seguirá al proyectil pero siempre mirando al objetivo.



Razonamiento:

El ángulo θ cambiara con respecto al movimiento del proyectil. La cámara siempre va a estar a una distancia fija sobre la bala. Por consiguiente la misma posición de la bala va a ser la de la cámara.

$$Pv3 = Pb$$

Una vez analizado los diferentes requerimientos, se adecuarán los cálculos en un algoritmo, el cual se alojará en el programa principal.

4.1.3 DESARROLLO

En este tema se verá todo lo correspondiente a la programación de los requerimientos ya analizados y se creará el ambiente de juego.

Constará de 6 etapas:

1. *ESCENARIO*. Ambiente tridimensional representando un valle rodeado con montañas. Utilización de técnicas de iluminación, el uso de cámaras y anexo de objetos.
2. *TIRO PARABÓLICO*. Simulación de la trayectoria de un proyectil.
3. *EFFECTOS AUDITIVOS Y VISUALES*. Efectos de disparo y de explosión - adicionado el sonido particular de cada uno de estos eventos.
4. *COLISION*. Empleo de métodos y objetos para detectar colisiones.
5. *MULTIPLAYER*. La creación de dos jugadores en línea en tiempo real.
6. *ACTORES*. Adición de los cañones importados de Maya.
7. *APLICACIÓN TERMINADA*. Afinando detalles.

ESCENARIO

Lo primero que se necesita hacer es crear el objeto *wtObj* (representa el Web Driver) en la pagina principal. Hay cuatro pasos básicos para llevar a cabo esto:

1. Incluir la apropiada etiqueta OBJECT/EMBED en la pagina anfitriona.

```

...
<OBJECT
  id="wtObj"
  classid="CLSID:FA13A9FA-CA9B-11D2-9780-00104B242EA3"
  width=900
  height=650
  CODEBASE="http://www.wildtangent.com/install/wdriver/generic/wtdinstFull.cab#version=?,?,?,?"
  <EMBED
    type="application/x-wildtangent-web-driver"
    name="wtObj"
    width=100%
    height=100%
    pluginpage="http://www.wildtangent.com/install/wdriver/generic/wtdinstFull.exe"
  </EMBED>
</OBJECT>

```

...

2. Incluir la etiqueta del *applet* que hace referencia a la clase principal llamada *Main*.

```

...
<APPLET
  CODE="Main.class"
  NAME="WtApplet"
  WIDTH="0"
  HEIGHT="0"
  MAYSCRIPT STYLE="position:absolute;left:-1;top:-1;" VIEWASTEXT>
</APPLET>

```

...

3. Se crea un *script* para que el *applet* comience a cargarse en la página. Se usa la función *begin()*, que se encuentra en nuestra clase *Main*, tomando como argumento el objeto *wtObj*.

```

...
<SCRIPT LANGUAGE=JavaScript>
...
  WT= document.wtObj;
...
  ( applet = app ).begin( WT);
...
</SCRIPT>

```

...

4. Este objeto será obtenido del *script* de la siguiente forma:

```

...
  public void begin (Object o)
  {
    Wt = wt3dLib.getWT(o);
  }
...

```

Después de haber creado el objeto principal *Wt*, lo utilizaremos para construir nuestro escenario tridimensional. Para esto hacemos un objeto *WtStage*.

```
stage = Wt.createStage();
```

Por consiguiente como este objeto va a estar en diferentes funciones se tiene que definir como variable global.

```
WTStage stage;
```

Para poder ver nuestro escenario se requiere poner algunas luces:

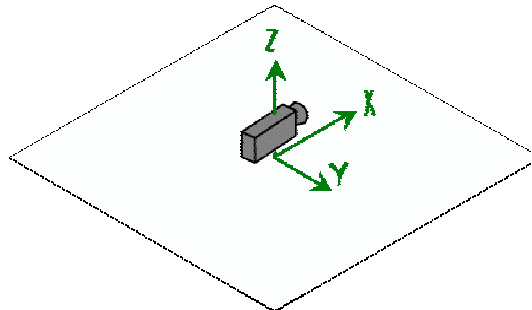
- a. Necesitamos luz de punto para los efectos de las sombras.
- b. Y luz de ambiente para que las áreas sombreadas no estén totalmente oscuras.

La luz de punto la posicionamos arriba de nuestras cabezas en las siguientes coordenadas: $x=0$, $y=0$, y $z=200$. Y la luz de ambiente cubrirá todo nuestro escenario.

Estos objetos los sumamos a la escena.

```
...
luz = pMain.Wt.createLight( 1 );    //luz de punto
pMain.Stage.addObject( luz );
luz.setAbsolutePosition( 0, 0, 200 );
luz2 = pMain.Wt.createLight( 0 );    // luz de ambiente
pMain.Stage.addObject( luz2 );
luz2.setColor( 80, 80, 80 );
...
```

Además de la luz se necesita otro objeto para ver nuestro escenario, el cual es una cámara. La cámara es creada para la escena, así que no necesita ser sumada a ella, pero tiene que estar orientada y poseionada.



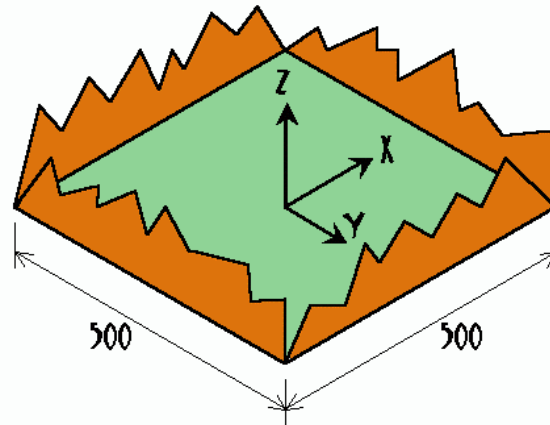
Orientación de la Cámara en el escenario.

```
...
pMain.Camara.setAbsoluteOrientationVector( 1, 0, 0, 0, 0, 1 );
pMain.Camara.setViewRect( 0, 0, wtWidth, wtHeight );
...
```

Se crea un grupo y se anexa la cámara.

```
...
camara=pMain.Wt.createGroup();
camaraGroup = pMain.Wt.createGroup();
camaraGroup.addObject(pMain.Camara);
...
```

Ahora se construye una geometría en el escenario, compuesta por cinco planos: el primero será el piso y los otros cuatro (a manera de paredes) servirán para simular montañas. El plano central será de 500 por 500 unidades.



Orientación de la Cámara en el escenario

Se crea un grupo contenedor donde se alojen los cinco planos, los cuales se sumarán al escenario:

```
...
ground = wt.createPlane( 500, 500 );
groundContainer = wt.createGroup();
groundContainer.attach( ground );
stage.addObject( groundContainer );
...
```

Los planos que van a retener las montañas serán de la siguiente dimensión:

```
mnt = wt.createPlane( 500, 70 );
```

Hacemos cuatro grupos y les damos nombre dependiendo de sus direcciones. Se necesitan ligar al modelo *mnt*.

```
...
norteCont = pMain.Wt.createGroup();
surCont = pMain.Wt.createGroup();
esteCont = pMain.Wt.createGroup();
oesteCont = pMain.Wt.createGroup();
// ligar modelos de montañas
norteCont.attach( mnt );
surCont.attach( mnt );
esteCont.attach( mnt );
oesteCont.attach( mnt );
```

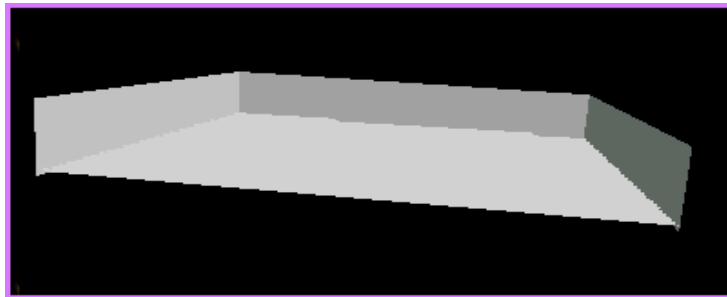
Ahora se posicionarán y orientarán los cuatro planos con respecto al suelo.

```
...
norteCont.setAbsolutePosition( 0, 250, 30 );
norteCont.setAbsoluteOrientationVector( 0, -1, 0, 0, 0, 1 );
surCont.setAbsolutePosition( 0, -250, 30 );
surCont.setAbsoluteOrientationVector( 0, 1, 0, 0, 0, 1 );
esteCont.setAbsolutePosition( -250, 0, 30 );
esteCont.setAbsoluteOrientationVector( 1, 0, 0, 0, 0, 1 );
oesteCont.setAbsolutePosition( 250, 0, 30 );
oesteCont.setAbsoluteOrientationVector( -1, 0, 0, 0, 0, 1 );
...
```

No olvidando que se tienen que añadir a la escena.

```
...
pMain.Stage.addObject( northCont );
```

```
pMain.Stage.addObject( southCont );
pMain.Stage.addObject( eastCont );
pMain.Stage.addObject( westCont );
...
```



Escenario

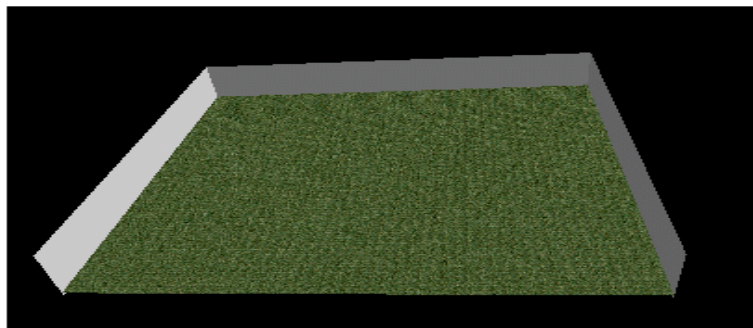
Plantemos algo de pasto en el plano principal, aplicando la textura en todo el suelo, de la siguiente forma:

```
...
groundBitmap = wt.createBitmap( "floor.wjp" );
ground.setTexture( groundBitmap );
...
```



Pasto.wpg

El escenario quedará así:



Escenario

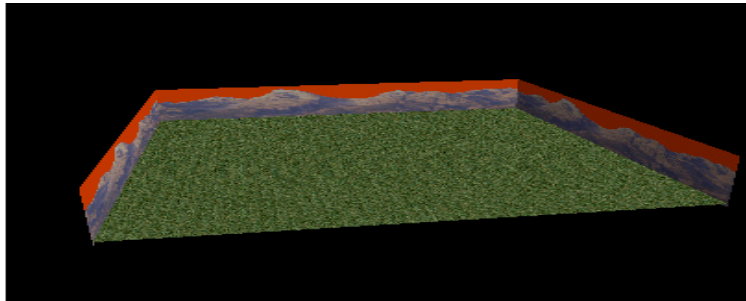
Se le aplica a los cuatro planos (lado frontal) una textura de montaña.

```
...
mtnBitmap = wt.createBitmap( "mtn.wpg" );
mtn.setTexture( mtnBitmap );
...
```



Montañas.wpg

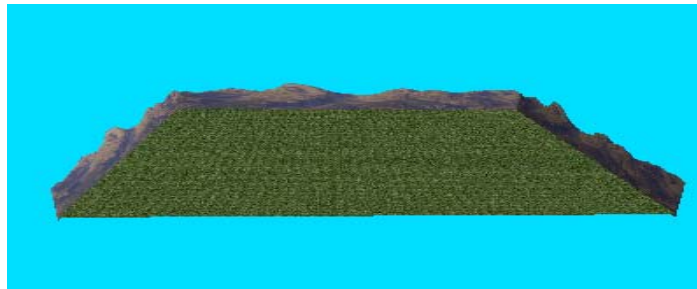
El resultado será:



Escenario

Para que la textura del cielo sea de color azul, se anexa lo siguiente en el código:

```
...  
mtnBitmap.setColorKey( 255, 0, 0 );  
stage.setBGColor( 98, 227, 255 );  
...
```



Escenario

Una vez desarrollado el escenario se necesita anexar tres objetos. Un cilindro (simulando un cañón), un cubo (objetivo), no sin antes poner el proyectil en forma esférica.

```
...  
bola = pMain.Wt.createSphere( (float)0.5,32 ); // bala  
gbola = pMain.Wt.createGroup();  
gbola.attach( bola );  
canon = pMain.Wt.createCylinder( 4,(float)0.8, 64, 0 ); //cañon  
gcanon = pMain.Wt.createGroup();  
gcanon.attach( canon );  
objetivo = pMain.Wt.createBox( 1,1,1); //objetivo  
gobjetivo = pMain.Wt.createGroup();  
gobjetivo.attach( objetivo );  
...
```



Objetos Anexados

La posición del objetivo y del cañón, no dependen de los jugadores sino del programa, y es de forma aleatoria. Se alojarán dentro de un diámetro de 400mts, por consiguiente sus coordenadas tendrán valores entre -200 y 200 unidades.

```

...
//posición del cañón y el objetivo en forma aleatoria
pb_x = -200 + (float)(Math.random()*400) ; // -200 a 200
pb_y = -200 + (float)(Math.random()*400);
po_x= -200 + (float)(Math.random()*400);
po_y= -200 + (float)(Math.random()*400);
distcampo=(float)Math.sqrt(Math.pow(po_x-pb_x,2)+Math.pow(po_y-pb_y,2));
...

```

Además se necesita que el cañón siempre este orientado a la posición del objetivo.

```

...
x=po_x-px; //calculando posición
y=po_y-py;
...
ui.gobjetivo.setAbsolutePosition( (float)po_x,(float)po_y,(float)po_z); //posición del objetivo
ui.prueba.setAbsolutePosition( (float)px,(float)py, (float)pz); //giro de cañon y rotacion
ui.prueba.setAbsoluteOrientation((float)0.0,(float)0.0,(float)1.0, (float)(360-angulo_g));
...

```

Para crear un ambiente más real, se pone un efecto de nubes en el escenario, esto se logra importando un objeto de Wildtangent llamado sky.wt y se anexa de la siguiente manera:

```

...
cieloModel = pMain.Wt.createModel("/models/sky.wt",0);
cieloModel.setScale(1f,1f,1f);
cieloContainer = pMain.Wt.createContainer();
cieloContainer.attach(cieloModel);
cieloContainer.setAbsoluteOrientation(1,0,0,90);
cieloContainer.setConstantRotation( 0, 0, 1, 1 );
...

```



Cielo con nubes

Para que haya mayor interactividad con el escenario se propuso adicionar dos cámaras y modificar el estado de la primera (3 como se había mencionado anteriormente). Esto es:

- La primera enfocará todo el escenario y estará atrás del cañón.
- La segunda seguirá a la bala enfocando el objetivo.
- La tercera estará perpendicular al objetivo y al cañón, a una distancia convenida para que se pueda apreciar el tiro parabólico de la mejor forma.

Se adicionan las dos cámaras al código:

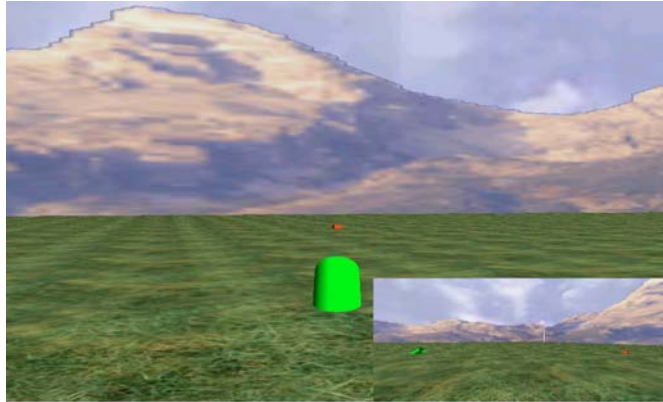
```
...
pMain.Camara1.setAbsoluteOrientationVector( 1, 0, 0, 0, 0, 1 );
pMain.Camara1.setViewRect( 0, 0, wtWidth, wtHeight );
pMain.Camara2.setViewRect(wtWidth - 400,wtHeight - 200,wtWidth,wtHeight);
...
```

Se anexan al grupo ya creado:

```
...
camaragirot=pMain.Wt.createGroup();
camaraGroup = pMain.Wt.createGroup();
camaraGroup.addObject(pMain.Camara1);
camaraGroup.addObject(pMain.Camara2);
camaragirot.addObject( camaraGroup );
...
```

El enfoque de la primera y segunda se observará en la pantalla principal, y la tercera será alojada en un objeto llamado *WTbitmap* (imagen de dos dimensiones) el cual estará alojado en la parte inferior derecha de la pantalla.

```
...
// Crear WTbitmap
rearOverlayImg = pMain.Wt.createBlankBitmap( 400, 200);
// se crea un borde alrededor de la vista de la cámara, se dibuja un cuadro transparente
// tamaño de vista 400x200
rearOverlayImg.setColor( 150, 150, 255 );
rearOverlayImg.setDrawColor( 1, 1, 1 );
rearOverlayImg.drawFillRect( 2, 2, 396, 196 );
// sumar la vista
rearOverlayDrop = pMain.Camera2.addDrop(rearOverlayImg, true);
rearOverlayImg.setColorKey( 1, 1, 1 );
rearOverlayImg.setTextColor(50,50,50);
rearOverlayImg.setTextBkColor( 1, 1, 1 );
rearOverlayImg.setTextHeight( 9 );
rearOverlayImg.setTextColor(255, 255, 255);
...
```

Primera y tercera cámara

```

...
//tercera cámara
void camaraCampo()
{
    ...
    pMain.Camera2.setFieldOfView((float)(distCampoC - menosdist1));
    pMain.Camera2.setAbsoluteOrientationVector( 1, 0, 0, 0, 0, 1 );
    pMain.Camera2.setAbsolutePosition( (float)(xcampo1+pxC), (float)(ycampo1+pyC), (float)4 );
    giroc1=-90+phi1;
    pMain.Camera2.setRotation((float)0,(float)1,(float)0,(float)giroc1);
    ...
}

```

Se controla la vista de las dos primeras cámaras mediante el evento de oprimir uno de los dos botones: Uno de ellos relacionado con la primera cámara llamado "Normal" y el otro con la segunda cámara llamado "Zoom".

```

//Cámara 1
public void camaraNormal( {
    ...
    //ubicación y giro de la cámara
    zoomC = 0;
    pMain.Camera1.unsetLookAt();
    pMain.Camera1.setAbsoluteOrientationVector( 1, 0, 0, 0, 0, 1 );
    pMain.Camera1.setAbsolutePosition( (float)x_camC,(float) y_camC,(float) 4 );
    ...
}

```



Segunda cámara(vista desde adentro del cañón)

```

//Cámara 2

```

```

public void siguiendoBala()
{
    ...
    if(apretezoomC == 1) {
        pMain.Camera.suspend( );
        apreteNormalC=0;
    }
    //comenzar a ser un zoomC al objetivo
    zoomC =1;
    pMain.Camera1.setLookAt(pMain.ui.ganon);
    ...
}

```

Ahora que ya se tiene el escenario virtual se podrán anexar los elementos restantes, empezando por darle “vida” mediante la simulación del tiro parabólico

TIRO PARABÓLICO

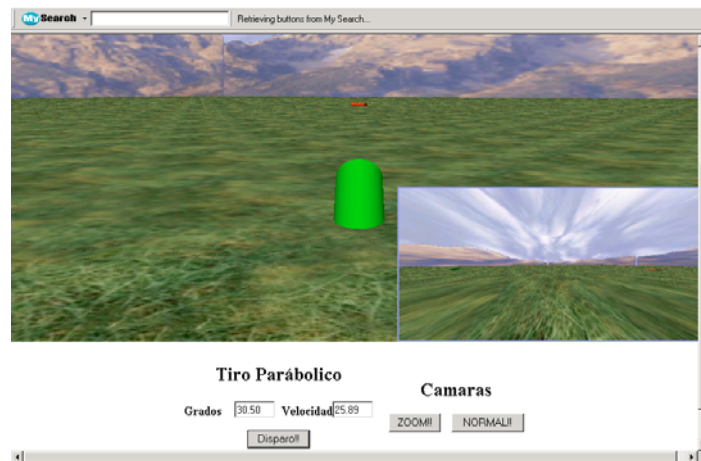
Se adecuó el razonamiento matemático sobre este fenómeno al siguiente algoritmo:

```

...
//Manejador de eventos
function rFunction(thisRender) {
    var a_t, a_g;
    a_t=(angulo_t/180)*(Math.PI); //tiro parabólico
    a_g= (angulo_g/180)*(Math.PI);
    tiempo+=thisRender.getInterval()/2000;
    incX=(vel*Math.cos( a_t)*tiempo)*(Math.sin(a_g));
    incY=(vel*Math.cos( a_t)*tiempo)*(Math.cos(a_g));
    incZ=(vel* Math.sin( a_t)*tiempo) - ((9.8/2)*(tiempo*tiempo));
    gbola.setAbsolutePosition( px+incX,py+incY,pz+incZ);
}
...

```

Tomando en cuenta que los parámetros a variar serían: la velocidad y el ángulo de disparo; se anexan dos áreas de texto para introducir estos valores, así como también un botón de disparo.

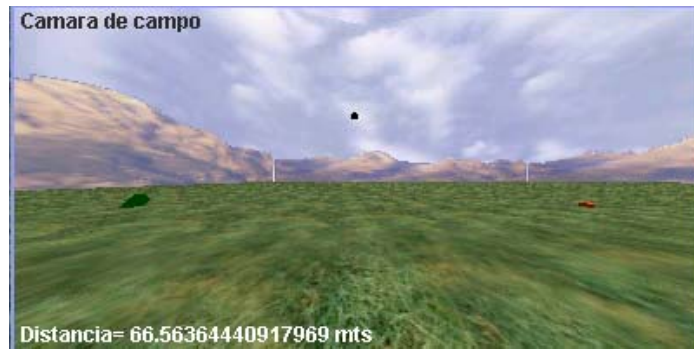


Adición de elementos relacionados con los parámetros

Se imprime en *Wtbitmap* el nombre de la cámara que utiliza esta vista y la distancia entre los dos objetos principales.

...

```
rearOverlayImg.drawText( 6, 2, "Camara de campo"); //etiqueta del Wtbitmap
pMain.ui.rearOverlayImg.drawText( 6, 182, "Distancia= "+distcampo+" mts" ); //imprime la distancia
...
```



Tiro Parabólico visto desde el Wtbitmap

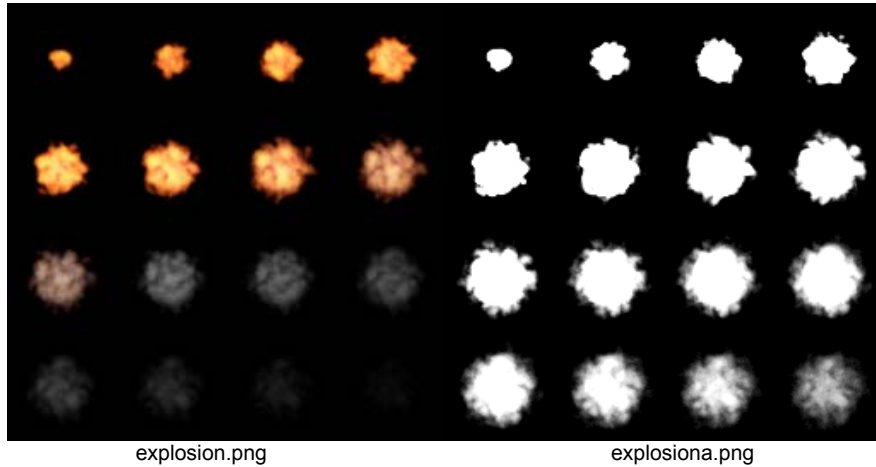
Ya que se pudo representar el fenómeno físico en la aplicación necesitamos anexar otros elementos (colisión, efectos visuales y auditivos) para que ésta se aloje en un contexto de realidad virtual.

EFFECTOS VISUALES Y AUDITIVOS

Se requirió crear un efecto de explosión cada vez que se oprimiera el botón de disparo o la bala hiciera contacto con otro objeto del escenario. Para lograr este evento se desarrolló un algoritmo cuya lógica se basará en una secuencia de subimágenes en un cierto intervalo de tiempo.

```
...
//Animación
animGroup = pMain.Wt.createGroup();
animGroup.setBitmapTextureRect(0, 0, (float)0.25, (float)0.25);
animFrame = -1;
animInterval = 50; //Milisegundos entre cada animación
lastAnimUpdateTime = 0;
animTex = pMain.Wt.createBitmap("explosion.png");
animTexAlpha = pMain.Wt.createBitmap("explosiona.png");
animTex.setTextureOpacityMask(animTexAlpha);
...
```

Las subimágenes se obtienen del siguiente mapa de bits.

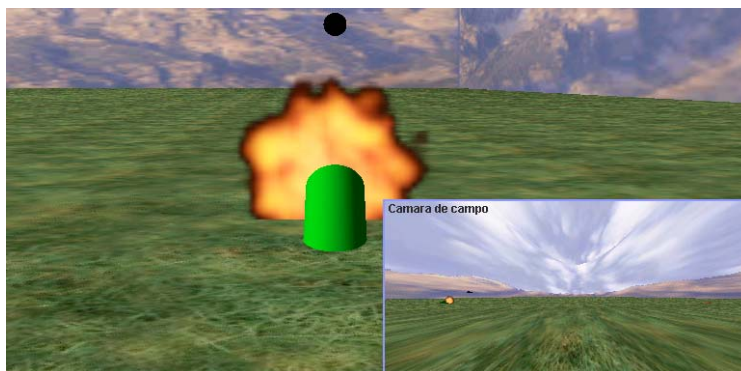


Se barre ésta imagen de la siguiente manera

```

...
//evento de la explosión
renderTime = event.getTime();
if (ui.animFrame > -1){
    if ((renderTime - ui.lastAnimUpdateTime) > ui.animInterval) {
        if (ui.animFrame == 16) {
            ui.animFrame = -1;
            ui.animGroup.setVisible(false);
            if(objetivoAlcanzado == 1){
                ui.printText ("Ganaste!!", 0);
                objetivoAlcanzado=0; } }
        else{u1 = ui.animFrame / 4;
            v1 = (Math.floor(ui.animFrame / 4)) * 0.25;
            u2 = u1 + 0.25;
            v2 = v1 + 0.25;
            ui.animGroup.setBitmapTextureRect(u1,(float)v1,(float) u2,(float)v2);
            ui.lastAnimUpdateTime = renderTime;
            ui.animFrame++;//incrementa animación frame
        }
    }
}
...

```



Disparo

Ahora se le anexa los efectos de sonido como:

- El disparo.

```
...
// cargar música y sonidos
disparo = pMain.Wt.createAudioClip( "rocklf1a.wav" );
...
```

- Y la explosión al hacer contacto con otros objetos.

```
...
explosionBlanco = pMain.Wt.createAudioClip( "rocklx1a.wav" );
//explosion pasto y montañas
explosionPasto = pMain.Wt.createAudioClip( "dynamite_exp_dist.wav" );
...
```

Para que los efectos se utilicen de manera correcta se necesita asociarlos a eventos como: la colisión de objetos y el disparo del proyectil.

COLISION

Ya que vamos a utilizar esta técnica se necesita hacer una referencia a la versión que contiene la librería correspondiente:

```
...
Wt.designedForVersion("3.1.0.0");
...
```

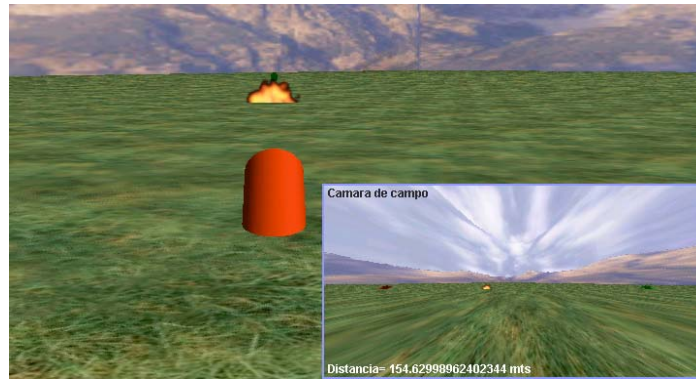
La bala será el único objeto que va a tener contacto con el escenario, por consiguiente se deben “marcar” los elementos que colisionaran con este objeto

```
..
//Reconociendo los objetos que participan en colisiones
groundContainer.setCollisionMask(3);
northCont.setCollisionMask(3);
southCont.setCollisionMask(3);
eastCont.setCollisionMask(3);
westCont.setCollisionMask(3);
gobjetivo.setCollisionMask(0);
gcanon.setCollisionMask(3);
..
```

La detención de la colisión se asociara al siguiente evento.

```
...
//colision
myCollInfo =ui.gbola.checkCollision((float)incX,(float)incY,(float)incZ, false, 3,3,0);
if( myCollInfo != null) {
    ver=myCollInfo.getHitObject();
    ...
    nx_cam=incX;
    ny_cam=incY;
    //chechar que objetos fueron golpeados
    if(ver.getName().equals("objetivo") ){
        }
    ...
    else{
        ...
    }
    ui.gbola.setVisible(false);
}
```

...



Colisión de la bala con el pasto

De esta manera se sabe que elemento del escenario fue tocado por la bala y que efecto se va a utilizar en consecuencia.

Ya que se ha logrado simular el tiro parabólico de manera virtual, el paso a seguir es adaptarlo a un sistema Multiplayer.

MULTIPLAYER

Para adicionar la funcionalidad Multiplayer a la aplicación se crea una instancia del objeto correspondiente en la pagina web:

...

```
<OBJECT classid=clsid:0C097121-C5D6-47EB-841D-30BFF71A71C4
        height=0
        width=0
        id=wtMulti>
</OBJECT>
```

...

Se anexa el objeto wtMult en el Script:

...

```
<SCRIPT LANGUAGE=JavaScript>
    ...
    WT= document.wtObj;
    ...
    ( applet = app ).begin( WT, document.wtMulti);
    ...
</SCRIPT>
```

...

Y en la clase *Main* se le anexa como argumento a la función *begin()*:

...

```
public void begin (Object o, Object Multi)
{
    Wt = wt3dLib.getWT(o);
    ...
}
```

...

Se hace una referencia en el código de la versión que se va a utilizar:

...

```
mPlayer.multiPlayer.designedForVersion ("2.0.0.0");
```

...

Y se importa la librería :

```
...
import wildtangent.webdrivermp.*;
...
```

Cuando se usan las bibliotecas *WTMultiplayer* la comunicación de una computadora con otra se hace dentro de un contexto de sesión; la sesión es manejada usando el objeto *WTSesion*. Una máquina puede crear una sesión anfitrión (servidor) o unirse (cliente) a una ya creada.

Para crear una sesión de tipo servidor se hace de la siguiente manera:

```
...
// sesión servidor
/* Llamado cuando el usuario oprime el botón "Servidor"*/
// hostSession (nombre-sesión,
//             identificador-sesión,
//             banderas (0 para conexión peer-to-peer),
//             clave(protección de la sesión),
//             maximoDeJugadores (0 para mínimo));

retVal = multiSession.hostSession ( "WTMultiplayer ", "5BFDB060-06A4-11d0-9C4F-00A0C905425E", 0, "", 0);
...
```

Y para el cliente como sigue:

```
...
// conectar a sesión
// connectToSession (IP o DNS del servidor,
//                  identificador-sesión,
//                  banderas (ignorado),
//                  clave (ignorado),
//                  tiempo de espera(milisegundos);
retVal = multiSession.connectToSession ( computerName, "5BFDB060-06A4-11d0-9C4F-00A0C905425E", 0,
"", 10000 );
//si retVal es igual al valor WTMPERR_CONNECTING, pone a la bandera "connecting" en verdadero
(WTMPERR_CONNECTING == retVal) connecting = true;
...
```

Como la comunicación entre ellos va a ser de manera asíncrona se convino crear un evento que verifique continuamente las llamadas, así se detectará al cliente cuando se comunique con el servidor.

```
...
//poll de conexiones
if (connecting)
{
    retVal = multiSession.connectToSession ( computerName, "5BFDB060-06A4-11d0-9C4F-
00A0C905425E", 0, "", 10000 );
    switch(retVal)
    {
        case WTMPERR_OK:
            createClientPlayer();
            connecting = false;
            break;
        case WTMPERR_CONNECTING:
            //esperar
            break;
    }
}
...
```

Cuando los mensajes son mandados por la red necesitan relacionarse a un ID (identificador único) de usuario; esto es análogo a una dirección de correo. Por lo tanto se creara un ID por cada jugador después de establecer una sesión.

```
...
// Creación de jugadores (nombre-jugador, banderas);
myPlayerID = multiSession.createPlayer ( "Cliente", 0 );
...
myPlayerID = multiSession.createPlayer ( "Servidor", 0 );
...
```

Una vez cumplido lo anterior, se preparará el mensaje a enviar:

```
...
multiSendMsg.clear();
multiSendMsg.writeInt(id);
multiSendMsg.writeString( mensaje );
...
```

Y viajará con su respectivo ID por la red:

```
...
// send (mi id, a quien id, banderas, mensaje);
int retVal = m_multiSession.send (myPlayerID, 0, 0, m_multiSendMsg );
...
```

Para recibir la información se utiliza el método *receive()*, el cual lee el mensaje lo descomprime y aloja sus diferentes componentes en objetos particulares.

```
...
retVal= multiSession.receive( 0, 0, 0, multiRecvMsg );
    while ( 0 == retVal )
    {
        playerID = multiRecvMsg.getFromPlayerId();
        if( false == multiRecvMsg.isSystemMessage() )
        {
            msgID = multiRecvMsg.readInt();
        }
    }
...
```

En la aplicación se utilizan dos objetos: un número entero y una cadena de caracteres. El número entero esta asociado con un menú de acciones y nos dice que tipo de mensaje es.

```
...
msgID = multiRecvMsg.readInt();
switch (msgID) {

    case MSGID_POSICION:
    ...
    case MSGID_CONECTADO
    ...
    case MSGID_DATOS:
    ...
    case MSGID_AQUIENLEDIO:
    ...
}
```

Y la cadena de caracteres puede contener:

- Los parámetros de la posición de los cañones en el escenario.

...


```

case MSGID_POSICION:
    String tempString = multiRecvMsg.readString();
    ConvPosicion(tempString,1);
    break;

```

- El valor que indica que ya se ha unido un cliente a la sesión.

```

...
case MSGID_CONECTADO:
    String cadservx=String.valueOf(pMain.jugadorserv.pb_x);
    String cadservy=String.valueOf(pMain.jugadorserv.pb_y);
    String cadclientx=String.valueOf(pMain.jugadorserv.po_x);
    String cadclienty=String.valueOf(pMain.jugadorserv.po_y);
    String posicion = cadservx+","+cadservy+","+cadclientx+","+cadclienty;
    sendString( posicion, 6);
    break;

```

- La información correspondiente del ángulo y velocidad de disparo

```

...
case MSGID_DATOS:
    String datosangvel = multiRecvMsg.readString();
    ConvPosicion(datosangvel,2);
    break;

```

- Y el nombre del cañón que ha sido destruido

```

...
case MSGID_AQUIENLEDIO:
    String quien = multiRecvMsg.readString();
    finServ(quien);

```

Estos datos son indispensables para que el servidor y el cliente simulen el entorno del juego en tiempo real.

Entonces anexaremos en la página web un botón referido a la sesión del servidor y otro botón para el cliente, con su respectiva área de texto para introducir la dirección del servidor a conectar.

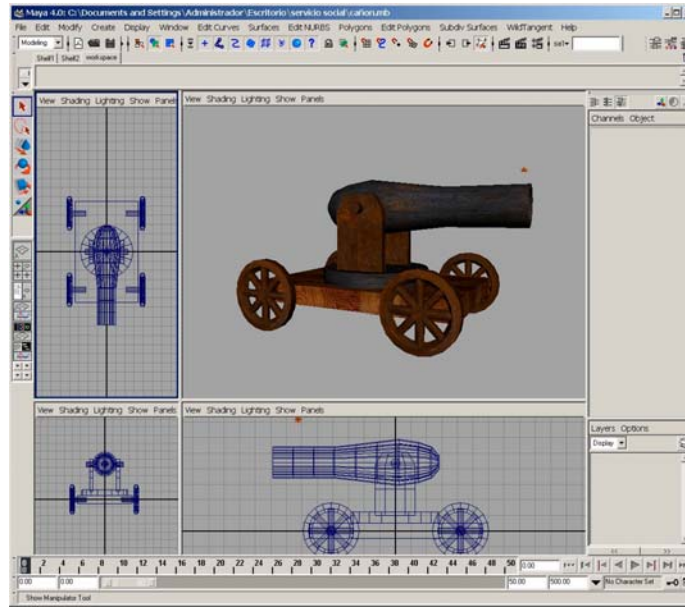
CREAR SESIÓN	UNIRSE A:
<input type="button" value="Servidor"/>	Dirección IP del Servidor: <input type="text" value="localhost"/>
	<input type="button" value="Cliente"/>

Elementos de sesión en la página principal

Ahora lo que falta hacer es sustituir los cilindros por actores (cañones) en la aplicación, importando estos de Maya.

ACTORES

Un actor es un objeto capaz de contener animaciones que pueden ser mezcladas. Puede ser creado en el editor WtStudio o en las herramientas de modelado: 3ds max y Maya. En este caso se diseñó el actor (cañón) en Maya.



Cañón creado en Maya

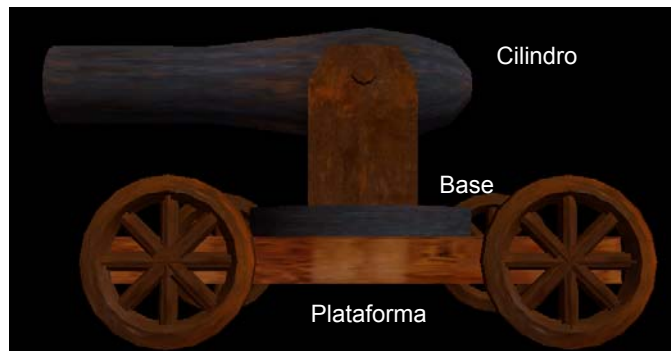
Después de ser editado se guardan todas sus propiedades en un archivo con extensión .wsad. Una vez hecho esto se exporta hacia la aplicación de la siguiente manera:

- El cañón es cargado a la aplicación como un conjunto de componentes, llamados “hijos” del objeto *WTActor*

```

...
WTActor  canonServ, objetivo;
...
//cañon servidor
canonServ=pMain.Wt.createActor("canion.wsad");
canonServ.setOnLoadedWithChildren (this);
...
//cañon cliente(objetivo)
objetivo=pMain.Wt.createActor("canion.wsad");
objetivo.setOnLoadedWithChildren (this);
...

```



Composición del cañón (hijos)

- y manipulado como un objeto *WTGroup*.
- ```

...
//obtención del componente cilindro y base del cañón servidor

```

```

cilindroServ=(WTGroup)canonServ.getChildByIndex(1);
base=(WTGroup)canonServ.getChildByIndex(2);
....
//obtención del componente cilindro y base del cañón cliente
cilindroClient=(WTGroup)objetivo.getChildByIndex(1);
baseCanonServ=(WTActor)canonServ.getChildByIndex(2);
...

```

Así de esta manera podemos utilizar sus componentes de forma particular, adaptándolos a la funcionalidad de la aplicación, como por ejemplo:

- El cilindro del cañón va a estar articulado para que gire con respecto al ángulo de disparo.
- La base y la plataforma funcionaran como uno, y tendrán la misma posición y orientación que el cilindro.

Después de agregar los actores sólo resta afinar detalles.

## APLICACIÓN TERMINADA

Cuando un jugador crea una sesión o se une a una ya creada, necesita saber si tuvo éxito su acción. Para resolver esto se convino en desarrollar una pequeña ventana que despliegue información en la parte izquierda inferior de la pantalla principal. Se creará con un objeto *WTbitmap* "vacío" y se alojará en la posición antes mencionada en la primera cámara.

```

...
//Se crea el objeto WTBitman vacío
b_server = pMain.Wt.createBlankBitmap (500, 50);

//Se adiciona al a camara 1
d_server = pMain.Camera1.addDrop (b_server, true);
...
// Propiedades del texto
b_server.setTextFace ("Arial");
b_server.setTextColor (255, 255, 255);
b_server.setTextBkColor (0, 0, 0);
b_server.setTextHeight (8);
...

```

Mediante la invocación de la siguiente función uno podrá escribir los mensajes deseados.

```

...
public void printText (String s_text, int i_box, int playerId)
{
 if(i_box == 0)
 {
 // Si necesitamos scroll.
 if (y_server >= (2 * 14))
 {
 ...
 ...
 }
 }
 ...
 // Se dibuja el texto.
 b_server.drawText (5, y_server, s_text);
}

```

Por lo tanto se mostrarían de esta manera en la página:



Actor y despliegue de mensajes

Con esto se termina el desarrollo de la aplicación.

## 4.2 Hoyos negros

Buscando un tema que sea atractivo se consideró que, por sus características, el estudio de hoyos negros permitía desarrollar un sitio con contenido didáctico y atractivo.

El primer paso fue recopilar información relativa al tema de hoyos negros. Dicha información debería contar con las características adecuadas en cuanto a nivel y profundidad. Posteriormente se diseñó la estructura general del sitio, los puntos a desarrollar y de los elementos que contendrían cada una de las páginas.

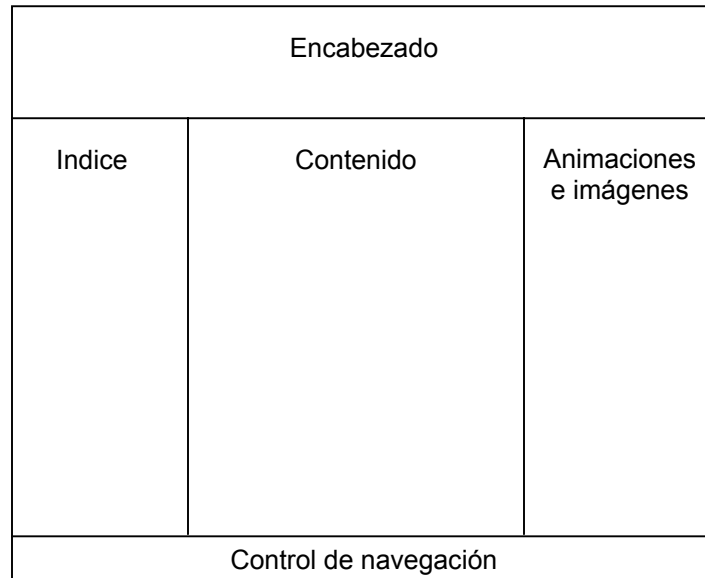
El sitio incluye información documental, animaciones, imágenes relativas al tema que se está exponiendo así como ligas a sitios de interés relacionados. El sitio se desarrolló en HTML con javascript y las animaciones se desarrollaron con Anark Studio.

### 4.2.1 OBJETIVOS DE LA APLICACIÓN

- Desarrollar un sitio web donde se expongan ideas generales y claras acerca del tema de hoyos negros que acerquen el interés del internauta al tema.
- El diseño del sitio, la información contenida en él, los elementos gráficos y de animación buscarán despertar el interés del visitante por conocer más sobre el tema.

## 4.2. 2 DISEÑO

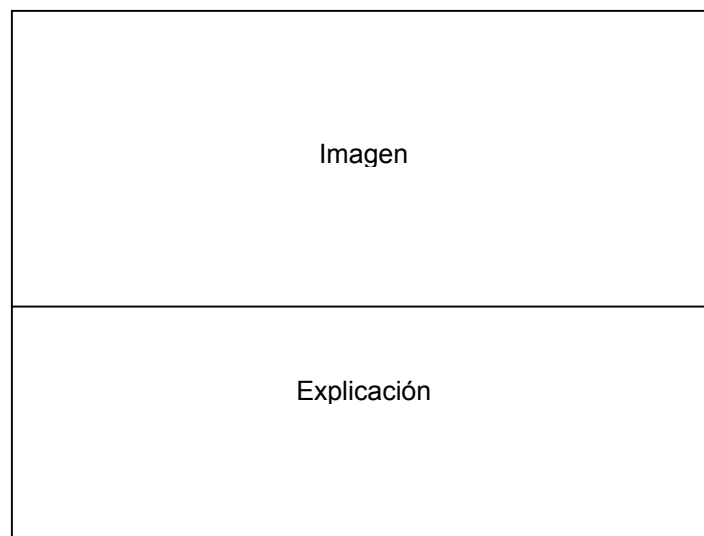
La estructura de todas las páginas que conforman el sitio es la siguiente:



Como se observa en la imagen, los elementos son los siguientes:

- ✓ Encabezado. El mismo para todas las páginas del sitio
- ✓ Índice. Contendrá un listado del contenido general del sitio así como la liga correspondiente para accederlo
- ✓ Contenido. Expondrá en forma breve y clara la información referente al tema seleccionado
- ✓ Animaciones e imágenes. Permitirá acceder a las animaciones desarrolladas y mostrará algunas imágenes relacionadas al tema expuesto. Las imágenes serán asimismo ligas a otra pantalla en la cual se detalla el contenido de las mismas.
- ✓ Navegación. Permitirá al usuario seguir con el siguiente tema o regresar al tema inmediato anterior.

Para las imágenes se incluyen páginas explicativas. Estas páginas se accederán dando un *click* sobre la imagen y la estructura de dichas páginas será la siguiente:



### 4.2.3 DESARROLLO

La primera etapa fue la recopilación de información relativa a los hoyos negros. El fenómeno de hoyos negros tiene su origen en una fuerza común a todos los objetos que conforman nuestro universo, la fuerza de gravedad.

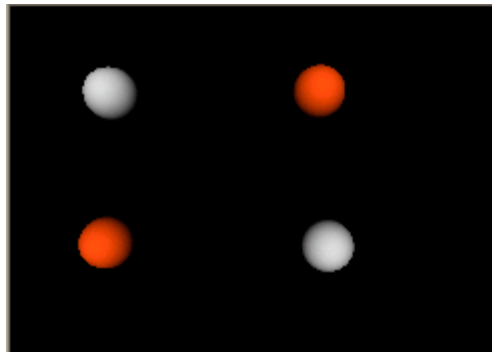
Actualmente los astrónomos piensan que los hoyos negros se forman del remanente de estrellas supermasivas. Todas las estrellas tienen un origen y un fin. Se forman a partir de gas estelar que bajo la acción de la fuerza gravitacional se contrae, reacciona y comienza a despedir energía térmica a fin de lograr mantenerse en equilibrio.

Durante millones de años, el hidrógeno se va transformando poco a poco en helio. En este proceso se libera una gran cantidad de energía. Para ilustrar este proceso de transformación y liberación de energía se diseñó y desarrolló una pequeña animación.

Los elementos que conformaron la animación fueron los siguientes:

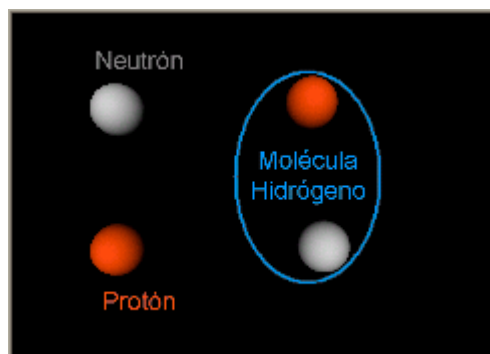
- ✓ Dos moléculas de hidrógeno (formadas cada una por un protón y un neutrón)
- ✓ Una molécula de helio
- ✓ Energía liberada

Los primeros elementos a incorporar en el proyecto fueron los componentes de las dos moléculas de hidrógeno.



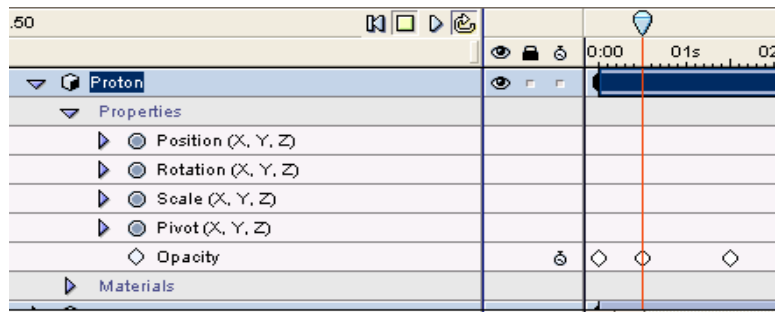
Elementos iniciales de la animación

A fin de hacer explícito el fenómeno de la reacción, se agregaron elementos explicativos consistentes en textos que nombran cada elemento que interviene en la reacción. Para la creación de los textos fue necesario utilizar una herramienta alterna que nos permitiera generar el texto, transparentarlo en las zonas no ocupadas por las letras y almacenarlo como imagen. Esta imagen se incorporó al proyecto.



Elementos explicativos

Para darle dinamismo a la escena, se decidió que los textos no fuesen estáticos, sino que apareciera y desapareciera. Para ello se utilizó la funcionalidad de la Línea de tiempo que Anark Studio nos proporciona.



Posicionamiento de *keyframes* sobre la línea de tiempo

Como se observa en la figura, el objeto *protón* se encuentra incluido en el proyecto desde un principio y a lo largo del tiempo van variando los valores de la propiedad *Opaciy* del objeto. Con ello se logra que el objeto aparezca y desaparezca dentro de los tiempos especificados.

Para establecer los intervalos en los cuales el objeto aparece y desaparece se marca la Línea de tiempo en puntos determinados. Posteriormente, para cada uno de ellos se establece el valor deseado en la propiedad sobre la que estamos trabajando, esto se hace a través de la ventana de propiedades. En este caso, se fijaron 3 puntos:

|                    |        |  |        |
|--------------------|--------|--|--------|
| A los 0 segundos   | Opaciy |  | 0.00   |
| A los 0.5 segundos | Opaciy |  | 100.00 |
| A los 1.5 segundos | Opaciy |  | 0.00   |

Dentro de la ventana de proyecto, cada elemento que se incorpora cuenta con una posición dentro de la misma. Esta posición se determina por medio de 3 coordenadas (x, y, z).

Cuando se desea que un elemento cambie de posición en un intervalo de tiempo se utiliza la Línea de tiempo para fijar puntos determinados y para cada uno de ellos se establecen los valores para las coordenadas x, y y z que se desean.

En el caso de la animación que estamos describiendo, se establecieron puntos y tiempos para mostrar una colisión de los átomos. El procedimiento para dar animación a cada elemento es idéntico al anteriormente explicado.

Cuando todo el hidrógeno se ha transformado en helio la estrella entra en un proceso de transformación y dependiendo de la masa puede terminar siendo una enana blanca, una estrella de neutrones o un hoyo negro.

### 4.2.3.1 Enana blanca

Cuando se agota el combustible de una estrella de hasta 1.5 veces la masa del sol, la presión interna no puede contrarrestar la propia fuerza gravitacional lo que resulta en una contracción de la estrella. Sin embargo, por el principio de exclusión de Pauli, sabemos que la masa no se comprime arbitrariamente porque los electrones de los átomos lo impiden.

Cuando la estrella comienza a comprimirse, los núcleos de los átomos se pegan entre si formando una red cristalina y los electrones se mueven libremente a través de esta configuración formándose un gas de electrones. Es esta corriente de electrones la que finalmente detiene la contracción gravitacional. En este punto, la enana blanca permanecerá brillando débilmente con el calor que le quede, posteriormente pasará a ser una enana roja y terminará como una enana negra, un cuerpo totalmente apagado.

El proceso por el cual una estrella termina su vida como una enana blanca se consideró adecuado para su representación por medio de animación.

Lo primero fue dar la idea de espacio exterior, crear el ambiente dentro del cual se desenvuelve la vida de cualquier estrella. Para ello se seleccionó una imagen donde se observa el espacio infinito estrellado.



Imagen de fondo

Esta imagen se incorporó al proyecto y se colocó como fondo dándosele las coordenadas y la dimensión adecuada, a través de la ventana de Inspección de propiedades, de forma que cubriera totalmente la escena. Al elemento de fondo se le establecieron las siguientes propiedades:

| Inspector - Image   |                   |
|---------------------|-------------------|
| Name                | estrellas2        |
| U Position          | 0.00              |
| V Position          | 0.00              |
| U Scale             | 1.00              |
| V Scale             | 1.00              |
| Texture Mapping     | Camera Projection |
| Texture Type        | Diffuse Map       |
| UV Rotation         | 0.00              |
| U Tiling            | Tiled             |
| V Tiling            | Tiled             |
| Compression Type    | Lossless          |
| Compression Quality | 50.00             |

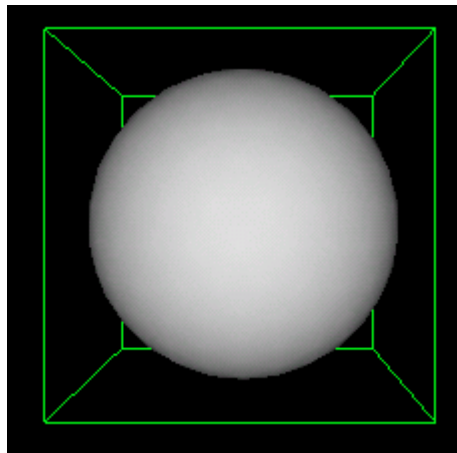
Propiedades del objeto de fondo



Con lo anterior ya se tenía el ambiente visual, sin embargo no era suficiente, dado que todos los cuerpos celestes en el espacio no están estáticos, como en una fotografía, sino que se encuentran en continuo movimiento. Era necesario darle sentido de movimiento al ambiente.

Para dar el efecto de movimiento se establecieron dos puntos (inicial y final) sobre la Línea de tiempo y a cada cual se le estableció un valor determinado para la propiedad *UVRotation*. Esta propiedad establece el grado de rotación de la imagen sobre el plano que la contiene en grados.

Posteriormente incorporó al proyecto objeto de tipo esférico. El objetivo era simular el gas interestelar del que posteriormente se forma una estrella.



Objeto esfera original

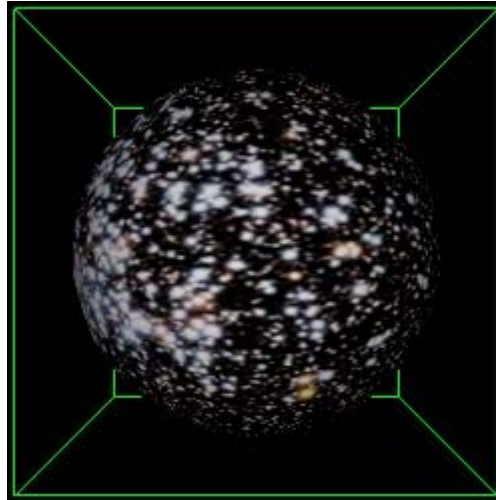
Se obtuvieron imágenes de diversos fenómenos estelares y se probó con cada uno de ellos tratando de lograr el efecto de partículas de gas que comienzan a reaccionar como producto de la fuerza gravitacional.

Después de múltiples pruebas, se seleccionó la siguiente imagen:



Textura seleccionada para envolver al objeto esfera

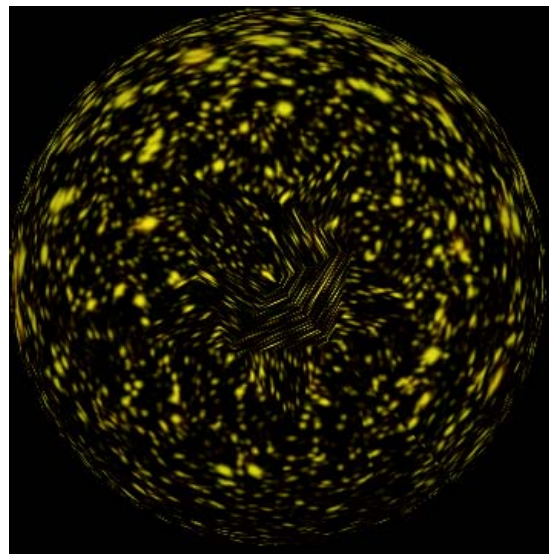
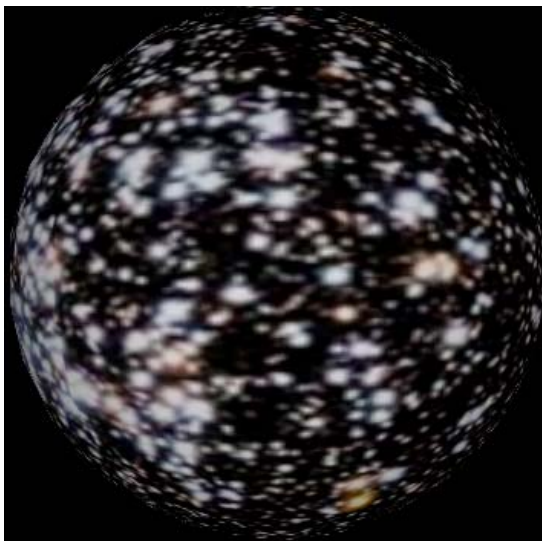
Con esta textura se "envolvió" al objeto esfera anteriormente incorporado al proyecto. Para realizar lo anterior se importó la imagen al proyecto a través de la opción *Import Resource* del menú *File*. Una vez incluida al proyecto, se arrastró hacia la línea de tiempo en el renglón de Materiales del objeto esfera. Quedándonos lo siguiente:



Objeto esfera con aplicación de textura

Posteriormente, para dar el efecto de gas, de elementos que reaccionan entre si, se combinaron los efectos producidos por la animación de varias propiedades simultáneamente: los valores X, Y y Z de la propiedad *Rotation* (haciendo que el objeto esfera gire sobre su centro) y una variación de  $120^\circ$  en la propiedad *UVRotation* de la textura que envuelve al objeto.

Por otro lado, se requería dar el efecto de calentamiento gradual, para ellos se jugó con la propiedad *Diffuse color* de la textura, iniciando en un gris opaco y llevándola hasta un blanco brillante.



Cambio observado al modificar la propiedad *Diffuse color*

Para la siguiente etapa, la de transformación de hidrógeno en helio, se utilizó otro objeto esfera. La textura utilizada para crear el efecto de una estrella en esta etapa de su vida se obtuvo de una fotografía del sol tomada desde el espacio.

Aunque para incorporarla como textura fue necesario modificar la imagen como se muestra:

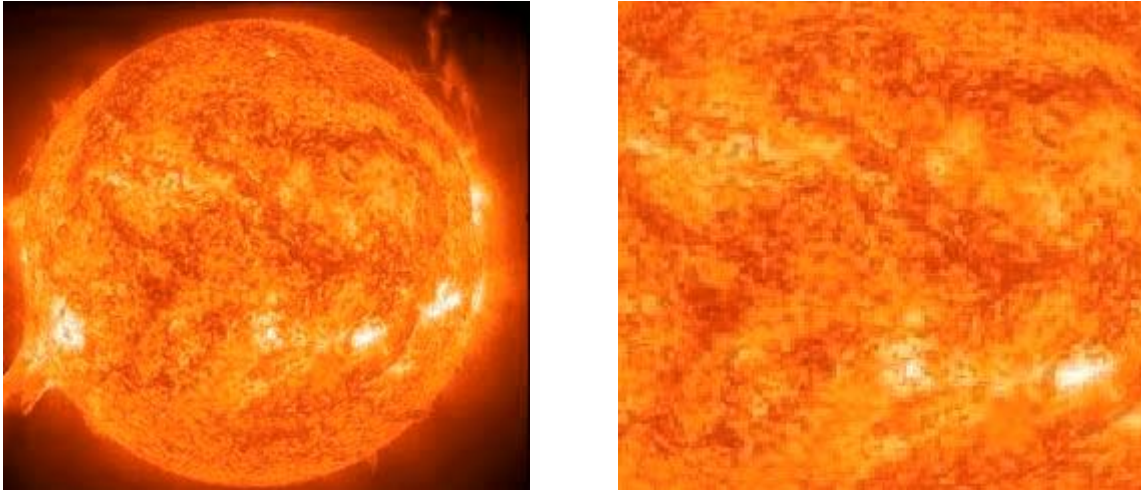
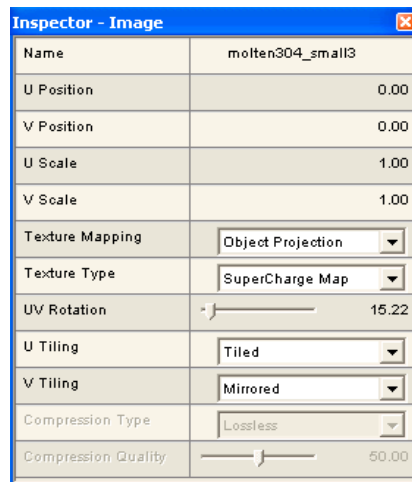


Imagen original e imagen modificada para su incorporación al proyecto

Esta textura se aplicó al objeto esfera y se le dieron las siguientes propiedades:



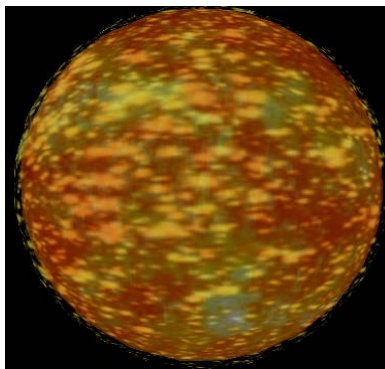
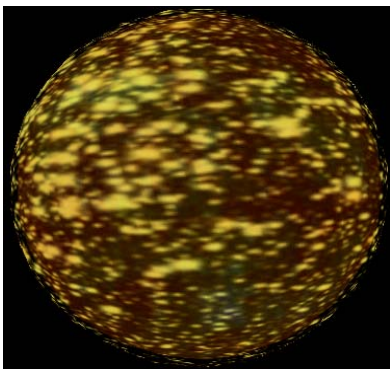
Propiedades de la textura

La propiedad *TextureMapping* se estableció como *Object Proyection* ser la que mejor creaba el efecto de una estrella. A la propiedad *Texture type* se le aplicó el valor *SuperCharge Map* por ser este el valor que mas luminosidad proporcionaba.



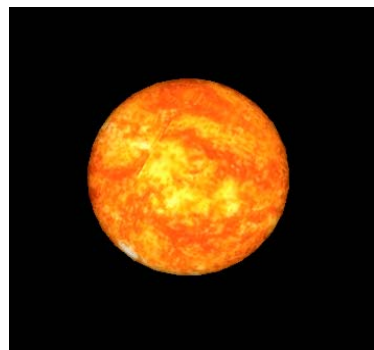
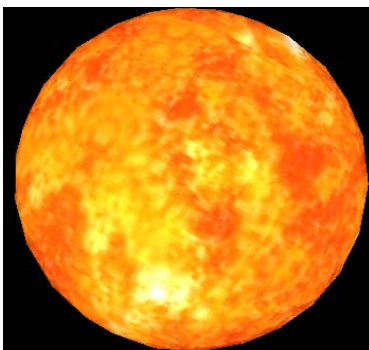
Cambio en luminosidad generado por el establecimiento de la propiedad *Texture type* como *SuperCharge Map*

Como se mencionó anteriormente, el objeto esfera debía simular la rotación que las estrellas tienen sobre su propio eje, para ello se le dio un movimiento giratorio. Este movimiento se combinó con cambios en la opacidad del objeto obteniéndose el efecto de una transformación gradual.



Movimiento giratorio y cambio en visibilidad

Posteriormente, para las etapas finales de contracción gravitacional se aplicaron variaciones a lo largo del tiempo en la propiedad *Scale*.



Disminución de tamaño generada por cambios en propiedad *Scale*

Para la etapa final, donde la estrella se transforma en una enana blanca, se oculta la esfera que muestra la etapa de transformación de hidrógeno en helio con la contracción gravitacional y se muestra otro objeto esfera que inicia gradualmente a aparecer. El tamaño de este nuevo objeto se hizo coincidir con el tamaño de la esfera que desaparece a fin de que el observador vea que es un sólo objeto el que va cambiando a través del tiempo.

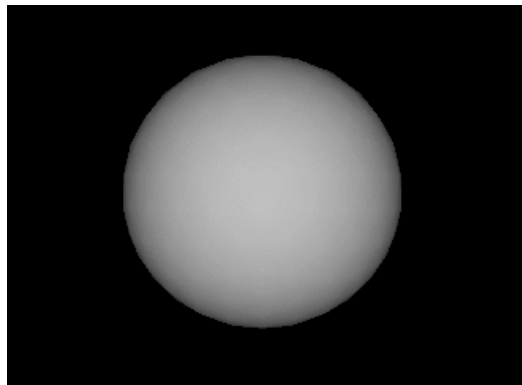
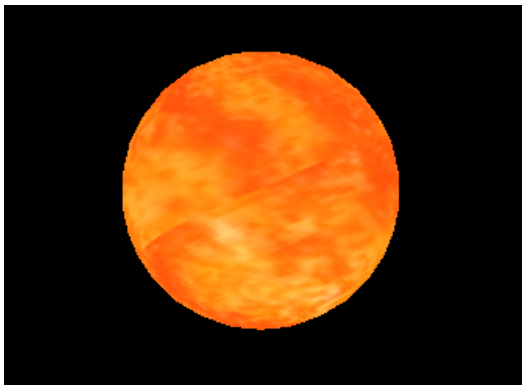
A esta última esfera se le aplicó la misma textura que a aquella con la cual simulamos la etapa de transformación de hidrógeno en helio. Sin embargo, se ven totalmente diferentes, lo cual se logró cambiando únicamente el valor de la propiedad *Texture Type*, la cual se estableció como *Specular map*. Obsérvese la diferencia entre ambos objetos.

El objetivo era mostrar una estrella en sus etapas finales que llega al equilibrio entre la contracción sufrida por la fuerza gravitacional y la presión interna que aún genera calor. Aunque no tan perceptible como en etapas anteriores, la rotación del objeto continua sobre su propio eje, ya que este movimiento nunca cesa.



Transformación a enana blanca

Y finalmente, a lo largo de cientos de miles de años, la enana blanca se va enfriando poco a poco. Primero se convierte en una enana roja y termina como una enana negra.



Transformación de enana roja a enana negra

Para hacer la transición entre el blanco brillante al rojo, se modificó la propiedad *Diffuse color* de la textura que se le aplicó a la esfera, pasando del blanco al rojo intenso. Lo anterior se realizó a través de la línea de tiempo y el uso de la ventana de propiedades.

Para este tipo de propiedad, el valor seleccionado puede establecerse de dos formas: 1) Seleccionando el color de la paleta de colores disponible, y 2) A través de una ventana de diálogo donde se establecen las propiedades Tonalidad, Saturación y Luminosidad. La propiedad Tonalidad controla la cantidad de rojo, verde y azul; la Saturación representa la cantidad de gris en proporción a la tonalidad y es un valor que va de 0 (completamente gris, sin color) a 255 (totalmente saturado), mientras que la Luminosidad es el grado de luminiscencia u oscuridad del color, va desde 0 (para negro) hasta 255 (blanco).

#### **4.2.3.2 Estrella de neutrones**

En los casos en los cuales la densidad de la materia excede la de la enana blanca, la estrella no es capaz de detener el colapso gravitacional llegándose a fusionar los protones con los electrones, teniendo lugar la formación de neutrones. Las estrellas de neutrones son cuerpos muy compactos y se forman cuando la estrella posee una masa de aproximadamente 2.5 veces la masa solar.

Las estrellas de neutrones comparten con las enanas blancas las etapas iniciales. El factor que decide si se forma una enana blanca o una estrella de neutrones es la cantidad de masa.

Para la animación de la estrella de neutrones se tomó la animación previamente creada para la enana blanca, modificando las etapas finales. En vez de pasar de una enana blanca a una enana roja y finalmente a una enana negra, la enana blanca no puede detener la compresión gravitacional, formándose una estrella de neutrones.



Simulación etapa final del surgimiento de una estrella de neutrones

La continuación del proceso de compresión se logra tomando el objeto esfera que representa una enana blanca y manteniendo constante hasta el final del ciclo el valor de la propiedad *Diffuse Color* en blanco al tiempo que se establecen puntos sobre la línea de tiempo del objeto y se dan a cada uno de ellos diferentes valores para la propiedad *Scale* (X, Y, Z).

#### **4.2.3.3 Hoyos negros**

En el caso de estrellas supermasivas, con masa superior a 6 u 8 masas solares, se produce un fenómeno muy espectacular. La estrella se comprime por efecto de la fuerza

gravitacional llegando el momento en que se produce una explosión en la cual se arroja al espacio una gran cantidad de materia. Dependiendo de la materia restante después de la explosión se puede formar una enana blanca, una estrella de neutrones o un hoyo negro.

Cuando la masa restante en la estrella, después de la explosión, no es capaz de contener el colapso gravitacional se forma un hoyo negro.

Al igual que en el caso de la estrella de neutrones, los hoyos negros comparten las etapas iniciales de una enana blanca sólo que la compresión generada por la fuerza gravitacional es tal que se produce una explosión en la cual se arroja una gran cantidad de masa al espacio.

Para simular la explosión de una supernova se utilizó una combinación de varios elementos y factores. El primer paso fue elegir algunas imágenes que dieran la sensación de materia suelta en el espacio. Las imágenes seleccionadas fueron las siguientes:



Imágenes seleccionadas para simulación de explosión de una supernova

Sin embargo, se observó que al incorporar estas imágenes al proyecto no se obtenía el efecto deseado, ya que al colocarse una sobre otra, la de abajo perdía visibilidad al igual que el fondo de la escena. Por lo anterior fue necesario modificar las imágenes antes de incorporarlas al proyecto.

Para la edición de imágenes se utilizó GIMP. GIMP es un programa especializado en la manipulación de imágenes (retoque, composición y creación) que se distribuye gratuitamente y que trabaja bajo diferentes sistemas operativos. Para transparentar una imagen parcialmente, se debe agregar a la misma un Canal Alfa y posteriormente se selecciona la región a limpiar.

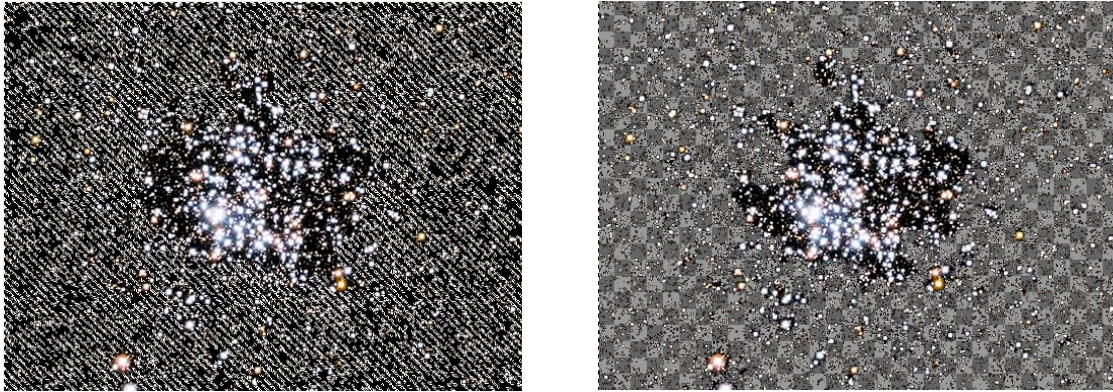


Imagen original e imagen modificada

Una vez que ambas imágenes fueron editadas se incorporaron al proyecto de animación. Para dar el efecto de gran cantidad de masa dispersándose, una de las imágenes se incorporó en varios planos.

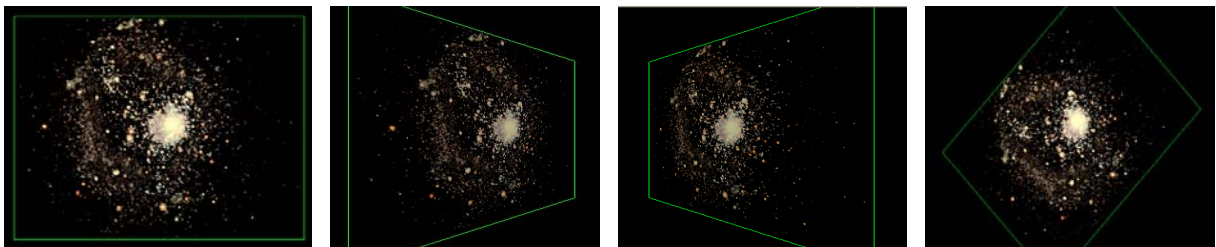
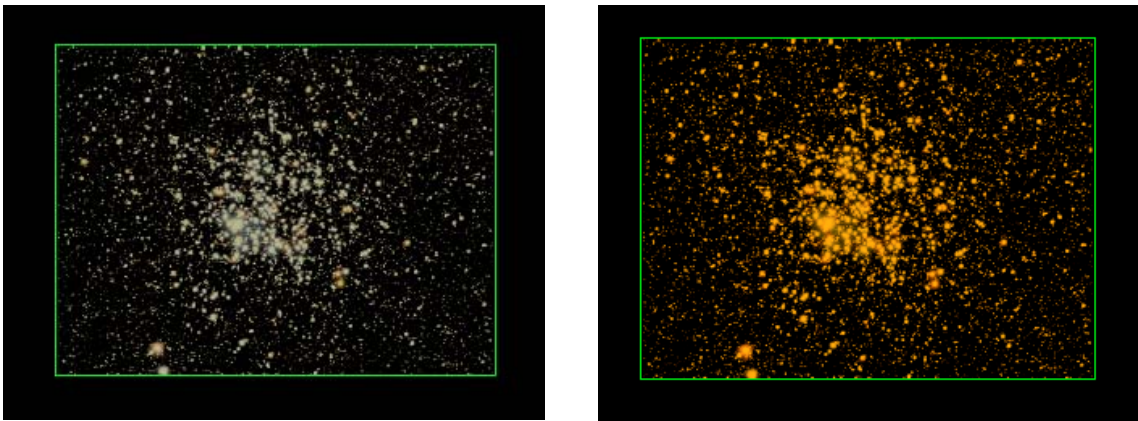


Imagen incorporada en varios planos

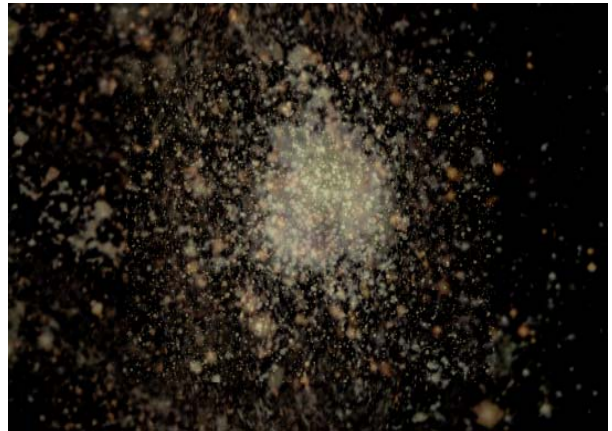
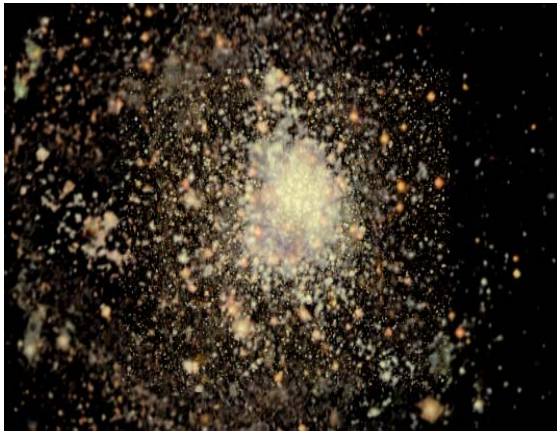
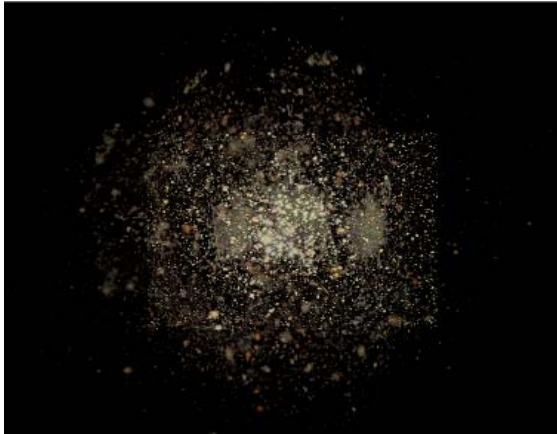
La segunda imagen se agregó en dos planos paralelos, pero cambiando el valor de la propiedad *Diffuse color* en cada caso.



Misma imagen en planos paralelos con variación en propiedad *Diffuse color*

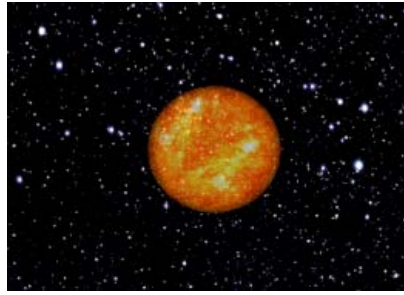
Una vez combinados todos los planos, se dieron diferentes tiempos para la aparición de cada objeto. A cada plano se le dieron también valores específicos en diferentes tiempos, con el fin de crear la sensación de masa que es expulsada al espacio con gran fuerza hasta desvanecerse.





Proceso de explosión de una super nova

Después de la explosión, la contracción gravitacional continua hasta que llega el momento en que la fuerza gravitacional es tal que se requiere una velocidad superior a la de la luz para escapar a la misma, con lo cual se forma finalmente un hoyo negro. Para simular esta última etapa se fue desvaneciendo lentamente el objeto (simulando como poco a poco la luz se va perdiendo).



Desvanecimiento gradual hasta llegar a un hoyo negro

### **4.3 CONCLUSIONES**

Se logró adaptar el análisis matemático del fenómeno físico de tiro parabólico a un algoritmo adecuado a la lógica de programación de Java y WildTangent.

La biblioteca gráfica se utilizó para construir todo el ambiente virtual (escenario), simulación del tiro parabólico, efectos de sonido y audio, colisión, despliegue de mensajes y jugadores en línea. En cuanto a la herramienta de modelado, solo se usó para crear los actores (cañones). Para llevar a cabo estas tareas se tuvo que consultar tutoriales y manuales del fabricante, traducir la información correctamente, entender las técnicas de desarrollo y hacer pruebas de menor a mayor complejidad hasta llegar al resultado mostrado.

Fue una buena decisión haber utilizado este software, ya que logramos satisfacer los objetivos de la aplicación.

Por otro lado, se logró simular mediante animaciones el origen que actualmente los astrónomos les atribuyen a los hoyos negros. También se crearon las animaciones correspondientes a la formación de una enana blanca y de una estrella de neutrones, fenómenos ligados a la formación de hoyos negros por su origen común.

Para el desarrollo de las animaciones se utilizó Anark Studio, una herramienta con múltiples objetos que se combinaron en tiempo y espacio hasta lograr el efecto deseado. Se utilizaron también texturas e imágenes. Todo combinado nos llevó a lograr el desarrollo obtenido.

En el siguiente capítulo se discutirán los resultados obtenidos más a detalle así como la viabilidad de ponerlo a disposición del público en general en algún servidor.

## 5 Resultados

A la luz del objetivo inicial: desarrollar aplicaciones interactivas que guíen, motiven y acerquen al estudiante a investigar y aprender las materias científicas; analicemos los resultados obtenidos.

### 5.1 DISCUSIÓN DE RESULTADOS.

Se crearon un par de aplicaciones demostrativas relacionadas a dos fenómenos físicos: Tiro Parabólico y Hoyos negros. Este desarrollo se evaluará desde dos puntos de vista:

- 1) *Pedagógico*. Se definirá si cumple o no con la finalidad de enseñar, de lograr que el visitante aprenda.
- 2) *Técnico*. Si es práctico y viable subir las aplicaciones desarrolladas en algún servidor para su libre acceso.

### **PEDAGÓGICO**

Se mantuvo la idea de que los contenidos educativos, diseñados en forma de aplicaciones interactivas, sean desarrollen como un juego y no como un deber. La teoría cognitiva juega un importante papel en el proceso de obtención de conocimiento.

El software desarrollado se apoya en el siguiente marco teórico:

- ✓ *La acción (Piaget)*. El usuario aprovecha su capacidad de adaptación para adquirir el conocimiento, mediante símbolos y esquemas.
- ✓ *En el aprendizaje por descubrimiento (Bruner)*. Los jóvenes irán buscando la información, manteniendo su interés a lo largo de todos los niveles.
- ✓ *Estructuras jerárquicas (Ausubel)*. En el recorrido de la aplicación efectivamente podrán avanzar más rápido si tienen conocimientos previos que les permitan obtener la información nueva. Sin embargo no siempre será de esa manera, pues habrá quienes no tengan información previa, ellos utilizarán esa información como anclaje para las siguientes, lo cual le permitirá ir formando una red de información.
- ✓ *Ir de lo general a lo específico (Ausubel)*. El tema esta expuesto de tal forma que de una explicación general pasa al detalle.
- ✓ *La instrucción del aprendizaje (Gagné)*. Se orienta el aprendizaje hacia metas específicas –Hoyos Negros y Tiro Parabólico- apoyadas en animaciones, imágenes, información escrita y la presentación de un juego.
- ✓ *La ley de la intensidad*. En ambas aplicaciones se incluyeron efectos llamativos y espectaculares que captarán la atención del visitante.
- ✓ *La ley de la novedad*. Se desarrollaron animaciones y efectos innovadores evitando textos monótonos y largos.

- ✓ *La ley del afecto.* El esquema propuesto deja una experiencia satisfactoria y así el usuario tenderá a repetirla.
- ✓ *La ley de la pluralidad.* El sentido de la vista y oído están involucrados en este proceso de aprendizaje.
- ✓ *La ley del ejercicio.* Mediante la práctica se comprenden los conceptos básicos de los temas estudiados.
- ✓ *Ley de la participación.* Hay una participación directa (dentro de un contexto de realidad virtual).

## TÉCNICO

Las dos aplicaciones demostrativas cubren líneas de aprendizaje web diferentes:

- Tiro parabólico, dirigido a juego virtual (videojuego).
- Hoyos negros, enfocado a sitio web.

### Tiro parabólico

Se desarrolló un juego web en el cual participan dos personas simultáneamente. Cada persona trata de eliminar a su contrincante haciendo disparos con un cañón. Cada disparo se caracteriza por la velocidad y el ángulo de lanzamiento. Un jugador gana cuando le atina a su contrincante.

Los resultados están enfocados al buen funcionamiento del juego -¿Qué efecto tiene el oprimir algún botón de la pantalla principal?.

1) *El primer jugador oprime el botón "Servidor".*



Botón servidor

- Se origina el ambiente virtual.
- Se crea una sesión y un ID de jugador, y espera a que otro se una a su sesión.



Inicio de sesión

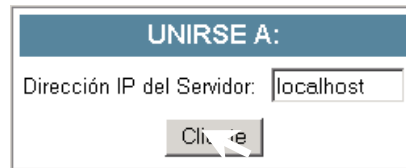
ID de jugador Servidor

Distancia entre los dos cañones



Distancia= 53.428916931152344 mts

2) El segundo jugador oprime el botón "Cliente" después de escribir el DNS o IP del servidor.



Botón cliente

- Se une a la sesión del servidor, éste le envía los parámetros principales del escenario: posición de los dos cañones.
- El cliente origina un ID de jugador y crea el ambiente virtual con los datos obtenidos del servidor.



Unión a Sesión

Jugador creado para este cliente.  
ID del Jugador: 3703992  
Comunicación con el jugador: 3703994

ID del cliente y unión con el servidor

Distancia= 53.428916931152344 mts

Distancia entre los dos cañones

3) *Oprime el primer jugador o el segundo el botón de “disparo”.*

| TIRO PARABÓLICO                        |                                 |           |                                 |
|----------------------------------------|---------------------------------|-----------|---------------------------------|
| Grados                                 | <input type="text" value="34"/> | Velocidad | <input type="text" value="35"/> |
| <input type="button" value="Disparo"/> |                                 |           |                                 |

Botón de Disparo

- Se toman los valores de las casillas de “Grados” y de “Velocidad”, con éstos se simula el tiro Parabólico en la pantalla del jugador y posteriormente son enviados al otro jugador para que se figure lo mismo –en tiempo real.



Disparo y vista del Servidor



Simulación en la vista del cliente

4) *Oprime el primer jugador o el segundo el botón de “zoom”.*



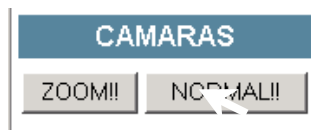
Botón zoom

- Se muestra en la pantalla principal el enfoque de la segunda cámara, dirigido al objetivo.



Vista al objetivo

- 5) *Oprime el primer jugador o el segundo el botón de "normal".*



Botón normal

- Se muestra en la pantalla principal el enfoque de la primera cámara, dirigido a nuestro cañón.



Vista atrás de nuestro cañón

Con esta información se entiende el funcionamiento interno de la aplicación, lo que resta es analizar si cumple con las especificaciones estándar de un juego virtual.

Según los expertos para construir un videojuego se debe tomar en cuenta lo siguiente:

- ✓ Tener una idea de cuál será la historia.
- ✓ Se tiene que definir el público al que apuntará el programa y cuáles son las limitaciones que enfrenta.
- ✓ Se debe medir las dificultades y la "jugabilidad" que se le quiere dar.
- ✓ Se tiene que hacer gráficos.
- ✓ Se tiene que escoger el lenguaje de programación. Programar la aplicación, como también efectos visuales y auditivos.

En el capítulo 4 se expone detalladamente el desarrollo del juego basándose en estas premisas. El resultado fue satisfactorio.

## **Hoyos negros**

Para el tema de hoyos negros se desarrolló un conjunto de páginas en html y javascript a las cuales se les incorporaron algunas animaciones desarrolladas con Anark Studio.

Todo sitio web debe cumplir con ciertas características:

### ➤ Generales

- ✓ Tener un objetivo claramente definido y el contenido del mismo debe apearse a ese objetivo.
- ✓ Mostrar de forma precisa y completa los contenidos o servicios que ofrece
- ✓ Se recomienda que la primera página sea diferente al resto del sitio, y que sirva de escaparate al mismo.
- ✓ La estructura del sitio debe estar orientada al usuario, pensando en sus objetivos y necesidades
- ✓ El *look & feel* debe corresponder con los objetivos, características y contenidos del sitio. Hay que poner atención en los colores empleados, que dan un mensaje subjetivo y por lo tanto impactan en la imagen que se da al visitante.
- ✓ Se debe ser coherente en el diseño general. Mantener coherencia y uniformidad en las estructuras de las páginas para no desorientar al visitante.
- ✓ En caso de que la información del sitio deba actualizarse periódicamente, se debe indicar la fecha de la última actualización

En nuestro caso, el objetivo está claramente definido: desarrollar aplicaciones interactivas que guíen, motiven y acerquen al estudiante a investigar y aprender las materias científicas. El tema específico son los hoyos negros. En la parte izquierda de la pantalla



se encuentra un índice en el cual muestran los temas que se exponen. El diseño de todas las páginas se mantiene a lo largo de todas las páginas y los colores se seleccionaron pensando en que sea llamativo y atractivo al visitante.

El tema de hoyos negros es de actualidad, no está todavía claramente definido, se encuentra todavía en estudio, lo que lleva a que constantemente vayan saliendo nuevas hipótesis, nuevo conocimiento, nuevos descubrimientos acerca del mismo. El sitio debiera mantenerse al día en cuanto a lo que va surgiendo, sin embargo, este desarrollo quedaría para una etapa posterior fuera de los alcances del presente trabajo. En un desarrollo posterior se incluiría entonces un elemento que muestre la fecha en la cual se hizo la última actualización.

#### ➤ Identidad e información

- ✓ Se debe mostrar claramente la identidad de la organización a través de todas las páginas. Esto puede llevarse a la práctica mediante un logotipo.
- ✓ Se recomienda proporcionar algún enlace con información de la organización así como algún mecanismo para entrar en contacto con la misma.
- ✓ En caso de incluirse artículos, noticias, etc. debe mencionarse la fuente de las cuales fueron tomados dichos contenidos.

Para lograr la identificación con la institución, se incorporaron los logos de la UNAM y del IIMAS. Asimismo se tiene una liga, accesible desde cualquier página, a través de la cual se accesa a una página donde se proporciona una dirección de correo para comentarios, dudas, etc. La información expuesta fue tomada de diversas fuentes, tanto documentales como electrónicas, que son listadas en la parte de Bibliografía.

#### ➤ Lenguaje y redacción

- ✓ Se debe hablar el mismo lenguaje que los usuarios
- ✓ Emplear un lenguaje claro, conciso, amigable, familiar y cercano
- ✓ Cada párrafo debe expresar una idea. Evitar párrafos con múltiples mensajes o vacíos en contenido.

El lenguaje fue uno de los principales elementos que se cuidó, además de utilizar un nivel adecuado se buscó ser claro y preciso.

#### ➤ Estructura y navegación

- ✓ Utilizar el tipo de estructura de organización y navegación adecuada.
- ✓ Si es un tipo de estructura jerárquica, mantener equilibrio entre profundidad y anchura
- ✓ Si es de tipo hipertextual, todos los clusters de nodos deben estar comunicados

- ✓ Los enlaces deben ser reconocibles y caracterizados. El usuario debe identificar los enlaces como tales, La caracterización debe indicar si están activos, si ya fueron visitados, etc.
- ✓ No deben existir enlaces que no lleven a ningún sitio.
- ✓ Deben existir enlaces que orienten al usuario con respecto a dónde se encuentra y cómo deshacer la navegación. Debe haber ligas a la página de inicio.
- ✓ Las imágenes que sirven de enlace deben incluir el atributo *title*, describiendo el destino al que la misma lleva.

Los enlaces se encuentran claramente definidos y localizados. En la parte izquierda se tienen enlaces a todos los temas expuestos, mientras que en la parte derecha se tienen enlaces a páginas explicativas de las imágenes. No se tiene ningún enlace que no lleve a alguna página específica o que lleve a alguna página que no corresponda con el título de la misma. Además en todas las páginas se tiene, en el área de navegación, una liga a la página inicial.

➤ *Lay-out* de la página

- ✓ Las zonas de alta jerarquía informativa deben contener la información de mayor relevancia.
- ✓ Debe evitarse la sobrecarga informativa. A través de colores, efectos tipográficos y agrupaciones para discriminar la información.
- ✓ Deben existir zonas en blanco donde se pueda descansar la vista
- ✓ Aprovechar al máximo el espacio visual. No exceder en elementos de decoración o grandes zonas en blanco
- ✓ Controlar la longitud de la página. Evitar en la medida de lo posible el *scrolling*, si el contenido es muy extenso, debe fraccionarse.

En el diseño general se utilizó el área central (la de mayor relevancia) para tratar el tema que en cada página se expone. Se trató, en general, que la exposición de cada tema fuera concisa y sin extenderse demasiado, evitando así hacerla tediosa y aburrida. El diseño de las páginas es sencillo y sin elementos de decoración superflua que distraigan o carguen en exceso la vista de las páginas.

➤ Búsqueda

- ✓ Cuando la extensión del sitio lo requiera debe incorporarse un buscador interno.
- ✓ Elementos multimedia
- ✓ Debe cuidarse que las fotos tengan la resolución suficiente y bien cortadas.
- ✓ Las imágenes y animaciones deben proporcionar un valor agregado.

Buscando captar la atención de los visitantes se incluyeron varias animaciones que intentan mostrar cómo se dan ciertos fenómenos relacionados con los hoyos negros. La finalidad de las imágenes incluidas es mostrar algún elemento visual respecto al tema tratado y cuentan con buena resolución. No se incluyó un servicio de búsqueda, pero para un trabajo futuro sería interesante incorporarlo.

#### ➤ Accesibilidad

- ✓ El tamaño de la fuente debe ser suficientemente grande para no causar dificultad en la lectura del texto
- ✓ La fuente, la alineación del texto deben facilitar la lectura. El color de fondo debe contrastar con el color de la fuente.
- ✓ Las imágenes deben incluir el atributo *alt* que describan su contenido
- ✓ El sitio debe visualizarse correctamente en diferentes navegadores y con diversas resoluciones de pantalla
- ✓ Se debe controlar el peso de la página. Minimizar la descarga de *plug-ins*.
- ✓ Se debe asegurar que la página se pueda imprimir correctamente.

El tamaño de la fuente del texto fue 3, tamaño adecuado para su lectura, mientras que el color del texto, contrasta adecuadamente con el color de fondo. Si el visitante desea visualizar las animaciones, el único *plug-in* que se requiere es el de Anark Studio.

## **5.2 EVALUACIÓN**

A fin de poder medir de alguna forma el nivel de asimilación o retención de las ideas y conceptos tratados, se desarrolló una pequeña aplicación web de evaluación. El acceso a la misma es opcional, es decir, cada visitante decide si quiere entrar o no, así como el número de preguntas que desea contestar.

Las preguntas son de opción múltiple y van apareciendo aleatoriamente. Es decir, no se muestra un cuestionario estático con n número de preguntas, sino que se va generando dinámicamente a partir de una base de datos de preguntas y respuestas.

Cada vez que el visitante contesta una pregunta se evalúa la misma y se procede a mostrar la siguiente, o a mostrar una página con una evaluación, en caso de que se haya alcanzado el número de preguntas total.

Es decir, se lleva un control de cuantas preguntas se solicitó contestar, cuántas ha contestado, cuántas ha contestado correctamente y cuántas incorrectamente. Con todos estos elementos se muestra al final una calificación. Con algunas tecnologías, para poder llevar o conservar los valores de las variables a través de múltiples páginas era necesario el uso de campos ocultos o se enviaban como parámetros entre una llamada y la siguiente. En nuestro caso cualquiera de estas opciones se podía haber utilizado, pero se consideró poco elegante, en su lugar se dio uso a las variables sesión.

Para poder dar la capacidad de dinamismo en las preguntas fue necesario utilizar algunas herramientas adicionales a las que ya se habían utilizado. Se utilizó la tecnología de jsp y se requirió el uso de una base de datos.

La estructura de la base de datos es sumamente sencilla y es la que se muestra a continuación:

Preguntas

|   | Nombre del campo | Tipo de datos |
|---|------------------|---------------|
| 🔑 | id               | Numérico      |
|   | texto            | Texto         |
|   | respuesta        | Numérico      |

Respuestas

|   | Nombre del campo | Tipo de datos |
|---|------------------|---------------|
| 🔑 | id_pregunta      | Numérico      |
| 🔑 | id_respuesta     | Numérico      |
|   | respuesta        | Texto         |

Cada pregunta tiene un identificador único, el texto de la pregunta y una respuesta correcta. Por otro lado, para cada pregunta se muestran n respuestas, cada respuesta se identifica de manera única por la pregunta a la cual pertenecen y por un número consecutivo.

### **5.3 CONCLUSIONES**

El software desarrollado cumple con un enfoque teórico acerca de cómo se da el aprendizaje en el ser humano y técnicamente cumple también con los requisitos para ponerse a disposición del público en algún servidor. La evaluación final corresponderá al usuario quien será el que defina si es de su agrado o no.

Obviamente quedan detalles y pendientes, los cuales se tratarán en el siguiente capítulo.

## 6 Trabajo futuro y conclusiones

De acuerdo con nuestra investigación se observa que existen múltiples teorías acerca de cómo debe darse el proceso de enseñanza-aprendizaje. Algunas de ellas, no le dan valor al papel del alumno y lo consideran un sujeto totalmente pasivo, sin embargo, en la educación a distancia, que puede tomar múltiples formas, requiere una participación muy activa por parte de quien desea aprender. Se debe entonces estimular al alumno y para ello se puede hacer uso de variadas herramientas.

El enfoque de juego es una forma por la cual podemos captar la atención de la persona al tiempo que le transmitimos conocimiento, o le hacemos necesaria la búsqueda del mismo en fuentes alternas. El juego se vuelve entonces una meta que nos lleva a recorrer cierto camino a través del cual aprendemos.

Otro elemento útil son las animaciones. Aunque estas no reflejen fielmente un fenómeno determinado, activan el proceso imaginativo, muestran a través de una sucesión de imágenes una idea, un concepto, una abstracción de la realidad. Nos dan, al menos, una aproximación al proceso estudiado.

Como en cualquier otro proceso de desarrollo de software, el primer paso es definir el objetivo del mismo. En nuestro caso, consideramos que el objetivo se cumplió y que el software desarrollado introduce al tema en cuestión al tiempo que motiva a conocer más sobre el mismo. En ambas aplicaciones se cuidó que el diseño y el ambiente fuesen visualmente atractivos, mientras que el uso de algunos efectos ayuda a mantener la atención.

Una vez definido y delimitado el objetivo a alcanzar, se deben definir las herramientas con las cuales se desarrollará. Para el desarrollo de software interactivo existen en el mercado múltiples herramientas y el producto final requiere, generalmente, del uso de varias de ellas.

Cada herramienta tiene sus propias características, capacidades y limitaciones. Las que se seleccionen serán aquellas que mejor se adapten a la solución deseada cuidando siempre la compatibilidad entre ellas.

En la selección de las herramientas deben considerarse factores de diversa naturaleza: económicos, técnicos, de disponibilidad, compatibilidad, etc. En ocasiones es preferible elegir una herramienta que tal vez no ofrezca el nivel máximo de capacidades de desarrollo dado que los recursos requeridos para el desarrollo y la posterior implementación para obtener un mejor desarrollo no se justifican por las mejoras en los resultados que se obtendrían.

El diseño es una etapa fundamental en la cual el apoyo de un profesional de diseño es muy valioso. Para el presente trabajo no se contó con dicho apoyo, sin embargo, como parte de una segunda etapa podría solicitarse el apoyo de un profesional del área que tomase lo ya desarrollado hasta el momento y propusiera algunos cambios y mejoras.

La etapa de programación o desarrollo es la más larga y requirió del aprendizaje, al menos en un nivel medio de las herramientas seleccionadas. Una vez conocidas se lleva a cabo un proceso creativo, de generación de ideas y reflejar las mismas en un desarrollo

concreto. El resultado obtenido generalmente no es el deseado, se tiene entonces que buscar otras formas, otros elementos, otros efectos y probar una y otra vez.

Cuando se ha llegado al desarrollo final difícilmente se puede ver a través de él la cantidad de trabajo y esfuerzo invertido para llegar al mismo.

Obviamente, siempre quedan detalles por afinar, ideas por incluir y puntos a desarrollar. En una siguiente etapa se mejorarían las aplicaciones. Debieran incluirse nuevas animaciones que mostrasen, por ejemplo, la estructura teórica de un hoyo negro o el proceso de caer dentro de uno de ellos.

También debiera irse incorporando información acerca de lo que día a día se va descubriendo sobre el tema, por ejemplo, recientemente se ha verificado cómo algunas estrellas en las últimas etapas de su vida, como enana blanca, reviven y vuelven a brillar. Un elemento que resultaría muy atractivo sería una galería de imágenes organizada por tema y a la cual pudiesen hacerse aportaciones de forma abierta.

# GLOSARIO

## A

**API** (Application Program Interface). Conjunto de reglas de programación que determinan como una aplicación debe acceder a un servicio.

**Applet**. Pequeña aplicación escrita en Java la cual se difunde a través de la red con el fin de ejecutarse en el navegador cliente.

**Asíncrono**. Que no tiene un intervalo de tiempo constante entre cada evento.

**Autodesk**. Casa desarrolladora de software, autora de aplicaciones como AutoCad, 3D Studio, Animator, AutoSketch, etc.

## B

**Browser** (Visualizador). Programa que despliega la información almacenada en páginas HTML que se encuentran disponibles en servidores del World Wide Web.

**Byte**. Es la unidad básica de información. En la práctica, se puede considerar que un byte es la cantidad de espacio necesaria para almacenar una letra. Tiene múltiplos como el Kilobyte, Megabyte, Gigabyte y Terabyte. Internamente, corresponde a 8 bits.

## C

**C**. Lenguaje de programación transportable y estructurado, que se puede usar para desarrollar software.

**C++**. Lenguaje de programación orientado a objetos derivado del C.

**Cliente**.

- a) Una aplicación que permite a un usuario obtener un servicio de un servidor localizado en la red.
- b) Un sistema o proceso que solicita a otro sistema o proceso que le preste un servicio.

**Cognitivo**. Proceso exclusivamente intelectual que precede al aprendizaje, las capacidades cognitivas solo se aprecian en la acción, es decir primero se procesa información y después se analiza, se argumenta, se comprende y se produce nuevos enfoques. El desarrollo de lo cognitivo en el alumno debe ser el centro del proceso de enseñanza por parte del docente.

**COM**. Nombre con el que se designa a cada uno de los puertos series o de COMunicaciones. Ej. COM1, COM2, etc.

**Correo electrónico**. En inglés e-mail, es un servicio de red para permitir a los usuarios enviar y recibir mensajes. Junto con los mensajes también pueden ser enviados ficheros como paquetes adjuntos.

## D

**Digital.** El uso de combinaciones de bits para representar datos en cómputo en base a un lenguaje binario. Contrasta con analógico.

**Dirección IP.** La dirección del protocolo de Internet (IP) es la dirección numérica de una computadora en Internet. Cada dirección electrónica se asigna a una computadora conectada a Internet y por lo tanto es única. La dirección IP esta compuesta de cuatro octetos como 132.248.53.10.

**DirectX.** Interfaz de programación bajo Windows, que permite acceder directamente al hardware (por ejemplo, tarjetas gráficas y de sonido). Es empleado para programar juegos en este sistema.

**DNS.** Sistema de nomenclatura de dominios (Domain Name System). Es un sistema que se establece en un servidor (que se encarga de un dominio) que traduce nombres de computadoras (como www.vlz.8m.com) a domicilios numéricos de Internet (direcciones IP) (como 132.248.10.1).

**Dominio.** Conjunto de computadoras que comparten una característica común, como el estar en el mismo país, en la misma organización o en el mismo departamento. Cada dominio es administrado por un servidor de dominios.

Los dominios se establecen de acuerdo al uso que se le da a la computadora y al lugar donde se encuentre. Ejemplo:

- .com Comercial
- .edu educación (USA),
- .gob gobierno (USA)
- .ec Ecuador
- .es España, etc,

Los dominios a su vez se van dividiendo en otros dominios:

- .gob.ec Gobierno de Ecuador
- .com.ec Comercio en Ecuador

## E

**Educación.** El proceso bidireccional mediante el cual se transmiten conocimientos, valores, costumbres y formas de actuar. La educación no sólo se produce a través de la palabra, está presente en todas nuestras acciones, sentimientos y actitudes.

## H

**Hardware.** Componentes físicos de un ordenador, todo lo visible y tangible. Por extensión, se aplica también a otros componentes electrónicos que no necesariamente forman parte de un ordenador.

**Hipertexto.** Es un documento digital que se puede leer de manera no secuencial. Un hipertexto tiene los siguientes elementos: secciones, enlaces y anclajes. Las secciones o nodos son los componentes del hipertexto o hiperdocumento. Los enlaces son las uniones entre nodos que facilitan la lectura secuencial o no secuencial del



documento. Los anclajes son los puntos de unión entre nodos. Los hipertextos pueden contener otros elementos, pero los tres anteriores son los mínimos y suficientes.

**Host** (Anfitrión). Computadora a la que tenemos acceso de diversas formas (telnet, FTP, World Wide Web, etc). Es el servidor que nos provee de la información que requerimos para realizar algún procedimiento desde una aplicación cliente.

## I

**Infografía.** Técnica de creación de imágenes a través del ordenador.

**Interactivo.** Un sistema se califica de interactivo cuando permite un diálogo continuo entre el usuario y la aplicación, contestando ésta a las instrucciones de aquél.

**Internet.** Es una red de cómputo a nivel mundial que agrupa a distintos tipos de redes usando un mismo protocolo de comunicación. Los usuarios en Internet pueden compartir datos, recursos y servicios. Internet se apoya en el conjunto de protocolos TCP/IP.

**IP.** Protocolo Internet. Permite a un paquete de datos viajar a través de múltiples redes hasta alcanzar su destino. Se encarga de la capa de red del modelo OSI.

## J

**Java.** Lenguaje de programación creado por Sun Microsystems. Desde su aparición, java se perfila como un probable revolucionario de la Red. Como lenguaje es simple, orientado a objetos, distribuido, interpretado, robusto, seguro, neutral con respecto a la arquitectura, portable, de alto rendimiento, multihilo y dinámico. Java es un lenguaje de programación, un subset de C++. Subset, porque algunas instrucciones (como las que tienen que ver con la administración de memoria) no se pueden usar. Seguro, porque agrega características de seguridad a los programas. Un applet de Java se baja automáticamente con la página Web y es compilado y ejecutado en la máquina local. Permite, entre otras cosas, agregar animación e interactividad a una página Web, pero su característica más importante es que un programa escrito en Java puede correr en cualquier computadora.

**JavaScript.** Lenguaje desarrollado por Netscape. Aunque es parecido a Java se diferencia de él en que los programas están incorporados en el archivo HTML.

**Joystick.** Es un dispositivo de entrada que es utilizado, comúnmente en juego de consola o PC. Literalmente, palanca de juegos. Usado para mover un objeto por la pantalla. Se compone de una base desde la que sale una palanca vertical, con la cual se controla el movimiento y también suelen incluir varios botones.

## L

**Lightwave.** Modelado y animación tridimensional por computadora.

## M

**Módem.** Equipo utilizado para adecuar las señales digitales de una computadora a una línea telefónica o a una red digital de servicios integrados (ISDN), mediante un proceso denominado de modulación (para transmitir información) y demodulación (para recibir información), de ahí su nombre. La velocidad máxima que puede alcanzar un módem para línea telefónica es de 33 kbps, sin embargo los más comerciales actualmente son los de 28 kbps.

**MSDOS** (Microsoft Disk Operating System). Sistema Operativo en Disco de Microsoft. Sistema operativo de línea de comandos muy extendido entre los PC.

**Multimedios.** La combinación de sonido e imágenes con texto y gráficos. Esto incluye películas, animaciones, música, personas que hablan, efectos sonoros como el rugido de una muchedumbre o vidrio aplastado.

## O

**Ordenador.** Una computadora (Hispanoamérica) u ordenador (España) es un dispositivo electrónico compuesto básicamente de un procesador, memoria y dispositivos de entrada/salida (E/S) . La característica principal del ordenador, respecto a otros dispositivos similares, como una calculadora, es que puede realizar tareas muy diversas, cargando distintos programas en la memoria para que los ejecute el procesador.

**Octeto.** Término para referirse a los ocho bits que conforman un byte.

**OS/2.** Sistema operativo de IBM que intentó suceder a DOS como sistema operativo de los PCs. Se desarrolló inicialmente de manera conjunta entre Microsoft e IBM, hasta que la primera decidió seguir su camino con su Windows 3.0 e IBM se ocupó en solitario de OS/2.

**OSI.** Interconexión de Sistemas Abiertos (Open Systems Interconnect). Es el protocolo en el que se apoya Internet. Establece la manera como se realiza la comunicación entre dos computadoras a través de siete capas: Física, Datos, Red, Transporte, Sesión, Presentación y Aplicación.

## P

**Papelógrafo.** Hojas de papel colocadas sobre un caballete para escribir o graficar que se utilizan con fines didácticos

**Plug-in.** Programa que se agrega a un visualizador del Word Wide Web que realiza funciones determinadas. Estas pueden ser visualización de archivos multimedia, soporte a archivos gráficos no estándares con el visualizador, etc.

**Protocolo.** Conjunto de normas que definen cómo se realiza el intercambio de datos entre ordenadores o programas, organizando el desplazamiento de la información a través de la red e indicando cuál es el origen de los datos, el camino que deben recorrer y el destino final. a) Es la definición de como deben comunicarse dos computadoras, sus reglas de comportamiento, etc. b) Definición de reglas.

**Publicar en la Web:** Instalar en un servidor Web páginas con información para que sean accesibles a cualquier persona conectada a Internet.

## R

**Realidad virtual.** Sistema de representación de imágenes y objetos que permite crear una "ilusión de realidad".

**Red.** Grupo de ordenadores y otros dispositivos periféricos conectados unos a otros para comunicarse y transmitir datos.

**Retroproyector.** Está diseñado para que el maestro pueda dar su explicación sin tener que dar la espalda a sus oyentes.

## S

**Scripts.** Un conjunto de comandos escritos en un lenguaje interpretado para automatizar ciertas tareas de aplicación.

**SDK.** Kit de desarrollo de software, un conjunto de aplicaciones para desarrollar programas en un determinado lenguaje o para un determinado entorno (Software Development Kit).

**Sesión.** Intervalo de tiempo comprendido entre el principio y el fin de una actividad en línea. La sesión corresponde a un tiempo de uso de Internet sin una interrupción superior a 10 minutos. Cuando se restablece la actividad tras una interrupción de 10 minutos, comienza una nueva sesión.

**Servidor.** Computadora dedicada a gestionar el uso de la red por otras computadoras llamadas clientes. Contiene archivos y recursos que pueden ser accedidos desde otras computadoras (terminales).

**Síncrono.** Que tiene un intervalo de tiempo constante entre cada evento. Son procesos síncronos los que dependen de un acontecimiento externo que los dispara.

**Software.** A diferencia del hardware, es lo que no se ve, es decir los programas y aplicaciones.

## T

**TCP.** Protocolo de control de transmisión (Transfer Control Protocol). Es el protocolo que se encarga de la transferencia de los paquetes a través de Internet. Se encarga de que los paquetes lleguen al destino sin ningún error o pide su reenvío. Se encarga de la capa de transporte del modelo OSI.

**Tiro parabólico.** Un caso típico de movimiento con aceleración constante. Este tipo de situación física se llama así debido a la forma parabólica de la trayectoria descrita por la partícula.

**TrueSpace.** Es un programa gráfico en 3D desarrollado para plataforma Windows.

## U

**Usuario.** Un usuario es la persona que tiene una cuenta en una determinada computadora por medio de la cual puede acceder a los recursos y servicios que ofrece una red.

## V

**Visual Basic.** Es un lenguaje de programación desarrollado por Microsoft. Desciende del lenguaje de programación BASIC. Su primera versión fue presentada en 1991.

**VRML.** (Virtual Reality Modeling Language antes Virtual Reality Markup Language) Lenguaje de programación utilizado para hacer presentaciones de realidad virtual en el World Wide Web. Puede ser un visualizador propio o integrado a los visualizadores WWW a través de un Plug-in.

## W

**Windows.** Sistema operativo para ordenadores personales comercializado por la empresa Microsoft y que representa la vía comercial y de negocio frente al movimiento cada vez más amplio del código abierto y Linux. En la actualidad representa un auténtico monopolio en el mercado de software dado que este sistema operativo se estima está instalado en el 90% de los ordenadores personales.

## BIBLIOGRAFÍA Y REFERENCIAS

### **3D Max**

<http://www.3dmax.com/>

### **Alambik**

<http://www.alambik.com/site/>

### **Anark Studio**

*Anark Studio Product Tour*

[http://www.anark.com/products/prod\\_tour\\_technology.asp](http://www.anark.com/products/prod_tour_technology.asp)

### **Anark Studio**

*User Manual*

Anark Corporation

USA, 2001

### **Bunn Ted**

*Black Holes FAQ*

<http://cosmology.berkeley.edu/Education/BHfaq.html>

### **Candia Alejandra**

*La educación virtual ¿Una alternativa a la educación tradicional?*

<http://www.utdt.edu/eduforum/ensayo10.htm>

### **Chan Mark C, Griffith Steven W., y Iasi Anthony F.**

*1001 Tips para programar con Java*

Mc Graw Hill

México, 1998

### **Couper Heather y Henbest Nigel**

*Agujeros negros*

Publicaciones CITEM

Gran Bretaña, 1996

### **Crichlow Melixa O. y Sánchez E. Dalila**

*Educación a Distancia*

Universidad Tecnológica de Panamá

Facultad de Ingeniería de Sistemas Computacionales

Agosto 1999

[http://www.utp.ac.pa/seccion/topicos/educacion\\_a\\_distancia/](http://www.utp.ac.pa/seccion/topicos/educacion_a_distancia/)

### **Goldstein H.**

*Mecánica Clásica*

Ed. Reverte

Barcelona, 1992

**Hacyan Shahen**

*Los hoyos negros y la curvatura del espacio – tiempo*  
Fondo de Cultura Económica  
La Ciencia desde México  
México, 1998

**Hanna Phil**

*JSP Manual de referencia*  
Mc Graw Hill  
Madrid, 2002

**Hassan Montero Yusef y Martín Fernández Francisco Jesús**

*Guía de Evaluación Heurística de Sitios Web*  
nosolousabilidad.com  
<http://www.nosolousabilidad.com/articulos/heuristica.htm>

**King Doug**

*Anark Studio 1.5*  
Computer Graphics World  
The magazine for Digital Content Professionals  
[http://cgw.pennnet.com/Articles/Article\\_Display.cfm?Section=Articles&Subsection=Display&ARTICLE\\_ID=157911](http://cgw.pennnet.com/Articles/Article_Display.cfm?Section=Articles&Subsection=Display&ARTICLE_ID=157911)

**Kuhn Anthony S.**

*Anark Studio Software Review*  
ComplxMinds  
[http://www.complxmind.com/images/anark/ComplxMinds\\_Review.pdf](http://www.complxmind.com/images/anark/ComplxMinds_Review.pdf)

**Lemay Laura**

*Aprendiendo Java en 21 días*  
Prentice-Hall  
México, 1999

**Manners Chris**

*Anark Studio 2*  
*Interactive Multimedia Authoring Tool*  
LimeVoodoo Productions  
<http://www.limevoodoo.com/articles/PDFs/0104Anark.pdf>

**Martín Bragado Ignacio**

*Física General*  
México, 2004

**Maya® and Alias MotionBuilder® software**

<http://www.aliaswavefront.com>

**Miller Erick, Adams Mark y Sims Max**

*Maya 5*  
Anaya Multimedia  
Madrid, 2004

**Publishing Perfection**

*Anark Studio 1.5, Windows*

Graphic Design Software

<http://www.publishingperfection.com/anark/ank113/>

**px379**

*Software 3D*

<http://px379.narod.ru/information.htm>

**Tehrani Bijan**

*Anark Studio 2.0*

Animation Magazine

September 09, 2003

[http://www.animationmagazine.net/article.php?article\\_id=1511](http://www.animationmagazine.net/article.php?article_id=1511)

**Universidad de Sevilla**

*Boletín de Educación Superior*

*Número 22*

*e-Learning*

[http://www.us.es/us/temasuniv/bol-edusup/bolet\\_educ\\_ESP22.htm](http://www.us.es/us/temasuniv/bol-edusup/bolet_educ_ESP22.htm)

**Villalobos Marco Abarca**

*Enseñanza asistida por computadora*

<http://bermudas.ls.fi.upm.es/~marco/Demo/educativo.html>

**Villasuso Gato José**

*Teoría del Tiro de proyectiles*

[http://usuarios.lycos.es/pefeco/movcomb/Movi\\_proy\\_Teor.htm](http://usuarios.lycos.es/pefeco/movcomb/Movi_proy_Teor.htm)

**WildTangent**

<http://www.wildtangent.com/>