



MODANS

MODIFICADOR de ANIMACIONES de HUESOS

**TRABAJO DE DIPLOMA EN OPCIÓN AL TÍTULO DE
INGENIERO en CIENCIAS INFORMÁTICAS**

Clasificación: Productiva

AUTORES:

Minardo Gollún González López

Kirenia Rojas Escobar

TUTOR:

Ing. Frank Puig Placeres

Ciudad de la Habana

mayo del 2007

AGRADECIMIENTOS

Quiero agradecer primeramente a esta Revolución que me ha brindado la oportunidad de estudiar la carrera con la que siempre soñé. A todas aquellas personas que en algún momento se preocuparon por mí evolución como estudiante o por el desarrollo mismo de la Tesis y a todos aquellos que han sido partícipes de mi formación como profesional.

Minardo Gollún González López.

Quisiera agradecer en primer lugar a la Revolución por hacer realidad mis sueños. Agradecer además a mis padres que han depositado en mi todo su apoyo para que yo llegara a ser lo que soy hoy. A Andy que me ha brindado su ayuda en todo lo que necesitado, a Frank, mi tutor por toda su ayuda y a todas aquellas personas que me han apoyado en uno u otro momento.

Kirenia Rojas Escobar.

DEDICATORIA

A mis padres que se han esforzado tanto por mí todos estos años, a mi hermana que me ha soportado y que siempre me llama y se ocupa de mí. A toda mi bella familia que siempre me han apoyado y querido. A Soto, a Gillermito, al Potril que siempre ha estado allí con la mano extendida cuando se le necesita, a todos mis amigos de siempre, que no podría citar sus nombre por miedo a omitir a alguno por descuido o por la tención del momento.

Minardo Gollún González López.

A mi familia y a Andy...

Kirenia Rojas Escobar

DECLARACIÓN DE AUDITURÍA

Declaramos que somos los únicos autores de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas (UCI) para que hagan el uso que estimen pertinente con este trabajo.

Para que así conste firmamos la presente a los ____ días del mes de ____ del año ____.

Minardo Gollún González López

Autor

Kirenia Rojas Escobar

Autor

Ing. Frank Puig Placeres

Tutor

OPINIÓN DEL USUARIO DEL TRABAJO DE DIPLOMA

El Trabajo de Diploma, titulado **Modificador de Animaciones de Huesos**, fue realizado en la **Universidad de las Ciencias Informáticas**. Esta entidad considera que, en correspondencia con los objetivos trazados, el trabajo realizado le satisface

Totalmente

Parcialmente en un _____ %

Los resultados de este Trabajo de Diploma le reportan a esta entidad los beneficios siguientes (cuantificar):

Como resultado de la implantación de este trabajo se reportará un efecto económico que asciende a _____ (**MN o USD**)

Y para que así conste, se firma la presente a los _____ días del mes de _____ del año _____.

Representante de la entidad

Cargo

Firma

Cuño

OPINIÓN DEL TUTOR DEL TRABAJO DE DIPLOMA

Título: **Modificador de Animaciones de Huesos.**

Autor: **Minardo Gollún González López**

Kirenia Rojas Escobar

El tutor del presente Trabajo de Diploma considera que durante su ejecución el estudiante mostró las cualidades que a continuación se detallan.

Por todo lo anteriormente expresado considero que el estudiante está apto para ejercer como Ingeniero Informático; y propongo que se le otorgue al Trabajo de Diploma la calificación de

Ing. Frank Puig Placeres

Firma

Fecha

RESUMEN

El proyecto “Paseo Virtual” de la facultad 5 desarrolla aplicaciones de Realidad Virtual¹⁴, en las cuales se utilizan animaciones 3D. Debido a que las computadoras disponibles no poseen los requisitos de hardware óptimos, se hace necesario aprovechar al máximo los recursos. Uno de los principales problemas, es el mal uso de memoria⁹, debido a la cantidad excesiva de Frames que contienen los ficheros de las animaciones generadas por los distintos editores.

En este trabajo se brinda una solución para esos problemas, mediante la implementación de un prototipo de aplicación que permite cargar las animaciones y aplicarle ciertas modificaciones. El objetivo principal es reducir los Key Frames innecesarios, y asignarle prioridad a los huesos del esqueleto.

Con la implementación de esta aplicación se obtuvo un producto capaz de analizar y transformar las animaciones esqueléticas. Las nuevas animaciones modificadas contienen la información que realmente necesitan, y utilizan solo los recursos necesarios, liberando memoria para otros procesos.

ABSTRACT

The project "Paseo Virtual" from the 5th faculty of the University of Information Sciences develop Virtual Reality applications, in which 3D animations are needed. Because of the hardware requirements of available computers aren't optimum, it is needed to fully exploit the resources. One of the main problems is the waste of memory, due to the excessive amount of Frames contained in the animations generated by the different design tools.

In this work it is presented a solution to these problems, by implementing an application prototype which allows to load animations and to apply them some modifications. The main target is to reduce the unnecessary Key frames and to assign priority to the skeleton's bones.

With the implementation of this application, a product able to analyze and transform the skeletal animation was obtained. the new modified animations contain the really needed information, and use just the necessary resources, allowing other process to use the released memory.

ÍNDICE

INTRODUCCION	1
FUNDAMENTACIÓN TEÓRICA	6
1.1 Animación	7
1.2 Fotograma (Frame):	7
1.3 Fotograma clave (Key Frame):	8
1.4 Interpolación:	9
1.4.1 Interpolación de Spline:	10
1.5 Técnicas para la animación:	10
1.5.1 Morphing animation:	11
1.5.2 Skeletal animation:	12
1.6 Herramientas para la creación de animaciones:	13
1.6.1 3D Studio Max:	13
1.6.2 Maya:	14
1.7 Proceso actual de realización de animaciones:	14
1.8 Metodologías y herramientas de desarrollo:	15
1.8.1 Metodología:	15
1.8.2 Herramientas:	16
1.8.2.1 Enterprise Architect 6.5:	16
1.8.2.2 Microsoft Visual Studio 2005:	17

1.8.2.3 3D Studio Max 7:	17
1.9 Lenguajes:	17
1.9.1 Lenguaje de programación:	17
1.9.2 Lenguaje de Modelado:	18
1.10 Consideraciones del Capítulo:	19
SOLUCIÓN PROPUESTA	20
2.1 Flujo de Trabajo:	21
2.2 Requerimientos de entrada:	23
2.3 Funcionalidades:	24
2.3.1 Reducir Key Frames innecesarios:	24
2.3.2 Priorizar huesos:	25
2.4 Consideraciones del Capítulo:	26
FUNCIONAMIENTO DEL SISTEMA	27
3.1 Definición de los Requerimientos Funcionales:	28
3.2 Definición de los Requerimientos no Funcionales:	28
3.3 Definición de Casos de Uso:	29
3.3.1 Definición de actores:	30
3.3.2 Diagrama de Casos de Uso por paquetes:	32
3.3.3 Descripción textual de Casos de Uso:	33
3.4 Modelo conceptual:	37

3.5 Arquitectura del sistema:	38
3.6 Funcionamiento del Framework:	39
3.6 .1 Petición inicializar:	40
3.6.2 Petición Cargar Animación:	41
3.6.3 Petición Cargar Mesh:	41
3.6. 4 Petición Aplicar Modificación:	42
3.6.5 Petición Exportar Animación:	42
3.7 Prototipo de Interfaz de Usuario:	43
3.7.1 Interfaz Cargar Animación:	43
3.7.2 Interfaz Asignar Prioridad:	45
3.7.3 Interfaz Reducir Key Frames:	46
3.7.4 Interfaz Exportar Animación:	47
3.8 Consideraciones del Capítulo:	48
DISEÑO DEL SISTEMA	49
4.1 Diagrama de clases del diseño:	50
4.1.1 Descripción de las clases:	50
4.2 Diagrama de clases del diseño por paquetes:	57
4.3 Descripción de la implementación por Casos de Uso:	60
4.3 .1 Caso de Uso Cargar Animación:	61
4.3 .2 Caso de Uso Cargar Mesh:	62

4.3 .3 Caso de Uso Reducir Key Frames:	63
4.3 .4 Caso de Uso Priorizar Huesos:	64
4.3 .5 Caso de Exportar Animación:	65
4.4 Consideraciones del capítulo:	65
RESULTADOS	66
5.1 Tabla de pruebas realizadas al software:	67
5.1.1 Pruebas de Asignación de Prioridad:	67
5.1.2 Pruebas de Reducción de Key Frames:	68
5.2 Consideraciones del capítulo:	71
CONCLUSIONES	72
RECOMENDACIONES	73
BIBLIOGRAFÍA	74
GLOSARIO DE TÉRMINOS	76
ANEXOS	79

ÍNDICE DE TABLAS Y FIGURAS

CAPÍTULO # 1

<i>Fig. 1: Secuencia de Frames en una animación.</i>	<u>8</u>
<i>Fig.2: Animación Patada Lateral.</i>	<u>9</u>
<i>Fig.3: Ejemplo gráfico de interpolación.</i>	<u>10</u>
<i>Fig.4: Área de aplicación de la Morphing Animations.</i>	<u>12</u>
<i>Fig.5: Ejemplo de una animación usando Skeletal Animations</i>	<u>13</u>
<i>Fig.6: Proceso actual de realización y utilización de las animaciones.</i>	<u>15</u>

CAPÍTULO # 2

<i>Fig.7: Diagrama del flujo de trabajo.</i>	<u>21</u>
<i>Fig.8: Creación del bípedo1.</i>	<u>22</u>
<i>Fig.9: Realización de la animación.</i>	<u>23</u>
<i>Fig.10: Jerarquía de huesos en un esqueleto.</i>	<u>26</u>

CAPÍTULO # 3

<i>Fig.11: Requerimientos Funcionales.</i>	<u>28</u>
<i>Fig.12: Requerimientos No Funcionales.</i>	<u>29</u>
<i>Fig.13: Paquete Importar Datos.</i>	<u>32</u>
<i>Fig.14: Paquete Aplicar Optimizaciones.</i>	<u>33</u>
<i>Fig.15: Paquete Exportar Datos.</i>	<u>33</u>
<i>Fig.16: Modelo conceptual.</i>	<u>38</u>

<i>Fig.17: Diagrama de clases del Framework.</i>	<u>39</u>
<i>Fig.18: Diagrama de secuencia para la petición Inicializar.</i>	<u>40</u>
<i>Fig.19: Diagrama de secuencia para la petición Cargar Animación.</i>	<u>41</u>
<i>Fig.20: Diagrama de secuencia para la petición Cargar Mesh.</i>	<u>41</u>
<i>Fig.21: Diagrama de secuencia para la petición Aplicar Modificación.</i>	<u>42</u>
<i>Fig. 22: Diagrama de secuencia para la petición Exportar Animación.</i>	<u>42</u>
<i>Fig.23: Interfaz Principal.</i>	<u>44</u>
<i>Fig.24: Interfaz Cargar Animación.</i>	<u>45</u>
<i>Fig. 25: Interfaz Asignar Prioridad.</i>	<u>46</u>
<i>Fig. 26: Interfaz Reducir Key Frames.</i>	<u>47</u>
<i>Fig. 27: Interfaz Exportar Animación.</i>	<u>48</u>
 CAPÍTULO # 4	
<i>Fig.28: Diagrama de clases del diseño por paquetes.</i>	<u>58</u>
<i>Fig.29: Paquete Manager.</i>	<u>58</u>
<i>Fig.30: Paquete Modifications.</i>	<u>59</u>
<i>Fig.31: Paquete Storage.</i>	<u>60</u>
<i>Fig.32: Diagrama de secuencia del caso de uso Cargar Animación.</i>	<u>61</u>
<i>Fig.33: Diagrama de secuencia del caso de uso Cargar Mesh.</i>	<u>62</u>
<i>Fig.34: Diagrama de secuencia Reducir Key Frames.</i>	<u>63</u>
<i>Fig.35: Diagrama de secuencia Asignar prioridad a huesos.</i>	<u>64</u>

Fig.36: Diagrama de secuencia Exportar Animación. _____ **65**

CAPÍTULO # 5

Tabla 1: Resumen de las pruebas realizadas con valores de error permisibles de: KFR = 0.001, KFT = 0, KFS = 0 a cada animación. _____ **69**

Tabla 2: Resumen de las pruebas realizadas con valores de error permisibles de: KFR = 0.001, KFT = 0, KFS = 0 a cada animación. _____ **69**

Tabla 3: Comparación referente a la cantidad de KB inicial y final de cada animación. _____ **70**

Fig. 37: Comparación referente a la cantidad de KB inicial y final de cada animación. _____ **70**

ANEXOS

Anexo 1: Diagrama de clases del diseño. _____ **79**

Anexo 2: Análisis de Pruebas de Reducir Key Frames. _____ **80**

INTRODUCCION

Desde siempre el hombre ha querido “animar” sus recuerdos mediante representaciones gráficas. Antes de la escritura el hombre utilizó el dibujo para representar los objetos y los hechos más relevantes que le rodeaban. La escritura permitió transcribir las imágenes mentales y visuales en palabras, pero nunca reemplazó a la pintura como idioma universal: una figura humana con los trazos más simples y limpios es descifrada por cualquier ser humano de cualquier época o cultura.

Con el surgimiento de las computadoras, el hombre ha aprovechado sus altas prestaciones para representar y animar situaciones del mundo real y en la actualidad resulta fundamental el conocimiento de las tecnologías de animación, presentes cada vez con mayor fuerza en todo tipo de productos, tanto en el mundo de la comunicación como en el de la tecnología. Se ha convertido en una pieza clave del arte digital.

Existen novedosas técnicas para la realización de animación 3D y otras no tan novedosas que dan la posibilidad de hacer realidad tus ideas. Producción de películas para cine y televisión, publicidad, videojuegos y efectos especiales, son sólo algunos de los innumerables entornos de aplicación, de un área creativa que avanza vertiginosamente, dándole nuevas formas a la realidad.

En la actualidad se cuenta con herramientas como el 3DStudio Max, Maya y otras para la realización de las animaciones 3D. Estas herramientas a partir de un bípodo¹ y utilizando técnicas como la animación esquelética, permiten obtener la animación deseada por el artista. El problema surge a la hora de exportar dicha animación, ya que, junto con los estados o Frames esenciales que el artista haya definido, la herramienta guarda otros Frames, aumentando considerablemente el volumen del fichero. Aunque mediante disímiles algoritmos, se logra reducir considerablemente la cantidad de estos Frames en el fichero resultante.

La cantidad creciente de Frames en los ficheros que contienen las animaciones es proporcional al volumen de recursos de hardware requeridos para procesarlos, provocando que el uso de la memoria sea excesivo.

Las herramientas de diseño actuales, como 3DStudio Max o Maya, implementan los algoritmos antes mencionados, pero de forma interna, que no permite realizar el proceso de reconstrucción de todos los datos necesarios para la utilización de estas animaciones en engines o visualizadores de otros desarrolladores, ya que no se tienen los algoritmos específicos, utilizados por dichas herramientas.

Desde el comienzo de la Universidad de las Ciencias Informáticas, el proyecto "Paseo Virtual" de la facultad 5, ha venido trabajando en entornos virtuales, realizando juegos y paseos virtuales, como indica su nombre, para lo que han necesitado la creación de animaciones.

Debido a la utilización de computadoras que en su mayoría no poseen los requisitos de hardware necesarios, se tornó evidente la necesidad de aprovechar al máximo los recursos. Uno de los principales problemas, es el mal uso de la memoria, producto a la cantidad excesiva de Frames en los ficheros que contienen las animaciones generadas en los distintos editores. Se realizó una investigación al respecto y se comprobó que este problema que presentaba el proyecto era común en otros grupos de desarrollo, por lo que se decidió, realizar una aplicación que resuelva de forma personalizada, las necesidades del proyecto.

Problema Científico:

¿Cómo reducir el espacio de almacenamiento de las animaciones esqueléticas y la edición de las propiedades más comúnmente usadas en los motores de simulación?

Objeto de Estudio y Campo de Acción:

- Animaciones esqueléticas en modelos virtuales.

- Los huesos contenidos en un esqueleto virtual y los Key Frames que definen sus movimientos.

Objetivo General:

Implementar una herramienta que permita editar y simplificar los recursos utilizados por las animaciones esqueléticas en entornos virtuales.

Tareas a desarrollar:

- Realizar una investigación sobre el efecto del uso excesivo de memoria en los proyectos de Realidad Virtual.
- Construir un prototipo de aplicación que cargue una animación del 3Dstudio u otro paquete de animación.
- Proponer algoritmos para compresión y reducción de Key Frames.
- Implementar módulos para el manejo y configuración de la animación.
- Presentar un flujo de trabajo que muestre el proceso la integración de las herramientas de diseño con las implementaciones dadas.

Métodos de investigación:

Los métodos teóricos posibilitaron descubrir, analizar y sistematizar los resultados obtenidos, para llegar a conclusiones confiables que permitan resolver el problema. En tal sentido se usaron:

- El **Analítico – sintético** se utilizó al analizar toda la información relacionada con el tema de tesis, ya que permiten la extracción de los elementos más importantes de cada documento analizado.
- La **inducción-deducción** se utilizó durante toda la investigación, para llegar a conclusiones y hacer generalizaciones.

- El **histórico-lógico** se utilizó al estructurar la trayectoria del objeto en el transcurso de su historia y en orden cronológico.
- La **Modelación Analógica** se utilizó para la elaboración tanto de proceso actual, como propuestas de solución.

Los métodos empíricos empleados en el transcurso de la investigación permitieron descubrir y acumular un conjunto de datos, que sirven de base para dar respuesta a las preguntas científicas. Para ello se utilizaron:

- La **observación** del tamaño resultante en los ficheros de las animaciones, exportados desde las herramientas de diseño y la calidad de dichas animaciones en tiempo real.
- La **encuesta** a estudiantes y profesores con el objetivo de conocer el criterio de éstos acerca de los beneficios que podría traer a los distintos grupos de desarrollo, la reducción del volumen de información a cargar por cada animación y sobre los problemas que se presentan por la carencia de computadoras con las suficientes prestaciones, lo que provoca la necesidad imperativa de economizar los recursos de hardware en todos los sentidos.

Los métodos estadísticos posibilitaron el procesamiento de toda la información. Con este fin se utilizaron:

- Elementos de la **estadística descriptiva** como consultas a especialistas para evaluar la metodología propuesta y el uso de tablas que ayudasen a la comparación de los datos iniciales con los datos resultantes.

Ideas a defender:

Si se desarrolla una solución que sea capaz de analizar y transformar los valores necesarios de las animaciones esqueléticas en los ficheros exportados por las herramientas de diseño, los

artistas digitales obtendrán animaciones que consuman solo los recursos necesarios, liberando memoria para otros procesos y con la información que realmente necesitan.

Organización del documento:

Este trabajo ha sido organizado de la siguiente manera:

El capítulo 1 presenta una introducción a la animación 3D; mostrándose los conceptos básicos relacionados con este tópico, incluyendo las técnicas, herramientas y proceso actual para su realización. Además de la justificación de las herramientas y lenguajes utilizados para la realización de la tesis.

El capítulo 2 muestra la solución propuesta para el problema que se plantea al inicio. Presentándose flujo de trabajo y funcionalidades de la herramienta en general.

En el capítulo 3 se muestra las funciones de la aplicación donde se incluye requisitos, casos de uso del sistema, y todos aquellos diagramas y datos, en general, que brinden una idea clara y directa del proceso de análisis del sistema, utilizando un lenguaje sencillo y comprensible por el usuario o cualquier interesado. Además se representa un breve manual de usuarios, que facilita el uso eficiente de la aplicación, garantizando además, que se utilicen al cien por ciento las funcionalidades que nos brinda la herramienta.

El capítulo 4 muestra el proceso de diseño de clases del prototipo de aplicación, incluyendo una serie de diagramas necesarios para la buena comprensión de la herramienta. Su objetivo es el de lograr guiar a equipo de desarrollo y principalmente a los programadores a la solución del problema basados en las herramientas y los lenguajes definidos.

El capítulo 5 muestra los resultados obtenidos, como lo indica su nombre y todas las pruebas y comparaciones, donde se representa mediante tablas y gráficos las potencialidades del sistema y la funcionalidad del mismo.

1

CAPÍTULO

FUNDAMENTACIÓN TEÓRICA

Este capítulo introduce los conceptos básicos de la animación 3D. Además se presentan técnicas y herramientas para la realización de animaciones y como se lleva a cabo actualmente el proceso de realización y posterior utilización de dichas animaciones.

Se presenta la justificación de las herramientas y lenguajes a utilizar, además la metodología de desarrollo para la realización de la tesis.

1.1 Animación

Animación se refiere al proceso de generación de imágenes donde cada una es una alteración de la anterior. La presentación de estas imágenes a una velocidad suficiente produce la sensación de movimiento.

La animación computarizada (tanto la 2D como la 3D) es una ilusión óptica que describe el cambio de un estado a lo largo del tiempo. Ambas constan con el suficiente número fotogramas (Frames) o cuadros por segundo, para dar efecto de continuidad.

Para el caso de bípedos en la animación 3D, en cada fotograma clave (key Frame) tenemos una postura¹² del personaje¹⁰, y al movernos por la línea de tiempo nos da la sensación de movimiento.

La animación de personajes implica la definición de los distintos segmentos tridimensionales y la unión entre ellos, definiendo puntos de conexión y puntos de rotación que permitirán hacer la animación. Esto hace que la animación 3D sea más compleja y necesite gran potencia de cálculo para producir efectos realistas en el movimiento.

1.2 Fotograma (Frame):

La animación está compuesta por varios aspectos relacionados entre sí. Unos de estos aspectos son los fotogramas o Frames (término en inglés), que no son más que un estado independiente, y cuya sucesión compone la animación. Esto viene dado por las pequeñas diferencias que hay entre cada uno de ellos que producen a la vista la sensación de movimiento.



Fig. 1: Secuencia de Frames en una animación.

En la Fig.1 se muestra una serie de cuadros donde en cada uno se encuentra un estado del movimiento, siempre teniendo en cuenta la continuidad de las posturas del personaje.

1.3 Fotograma clave (Key Frame):

Otros de los aspectos fundamentales dentro de la animación son los llamados fotogramas claves o Key Frames (término en inglés), estos son los estados que el animador considera que son esenciales para definir mejor el movimiento y el sistema o software en el que se hace la animación es el encargado de generar automáticamente los Frames intermedios o In-betweens usando métodos de interpolación.

Es muy importante que los Key Frames que se establezcan representen claramente el movimiento para que la interpolación tenga suficiente información.

La manera en que se generan los Key Frames y se controlan los in-betweens determinará la calidad y fluidez de la animación. Cuando las escenas se hagan más complejas será bueno saber como crear controles más avanzados que harán el proceso de animación más intuitivo y menos mecánico.

En la Fig. 2 se muestra la animación de una patada lateral y la línea de tiempo del 3D Studio Max donde vienen especificados los Key Frames que se definieron para dicha animación, por ejemplo el Key Frame 8 (el que viene señalado) donde el bípedo tiene la postura que se muestra.

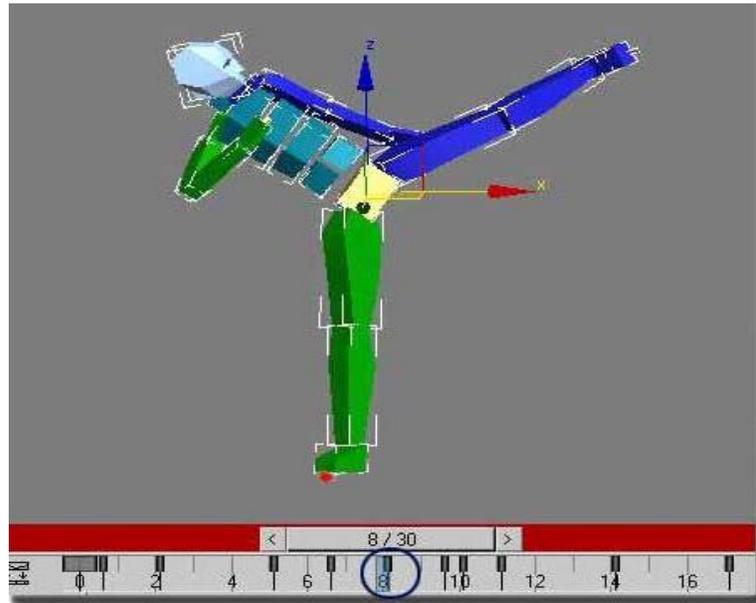


Fig.2: Animación Patada Lateral.

1.4 Interpolación:

Se ha mencionado varias veces el término de “interpolación”, el cual es muy importante para obtener mayor fluidez y dinamismo en las animaciones y no es más que el método que utilizan las herramientas de animación para generar los Frames o cuadros intermedios entre los Key Frames definidos por el artista.

Este método permite obtener nuevos puntos, partiendo del conocimiento de un conjunto de puntos dados, mediante ecuaciones matemáticas.

En las animaciones esto permite tener solo los estados o posturas (en animación de bípedos) esenciales que el artista haya creado, para obtener el resto. Esto se puede ver con mayor claridad en la siguiente figura.

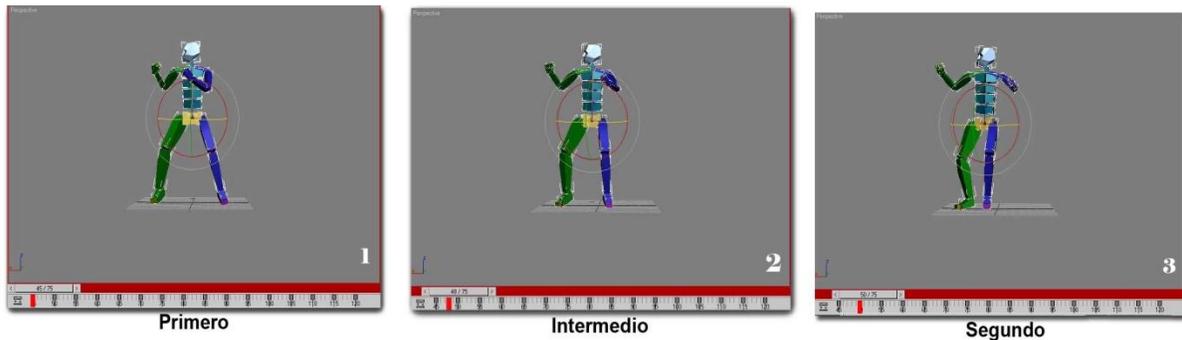


Fig.3: Ejemplo gráfico de interpolación.

En la Fig. 3 se muestran tres posturas diferentes, un primer Frame (1), un segundo Frame (3) y se muestra además uno de los Frames intermedios que se obtiene luego de la interpolación (2), esto nos permite no tener que hacer manualmente cada uno de los estados intermedios.

Es importante decir además que existen varios métodos de interpolación entre ellos la interpolación lineal y la interpolación de Spline, esta última, por sus características, es muy usada en el mundo de los gráficos 3D.

1.4.1 Interpolación de Spline:

Un spline es una curva definida a trozos mediante polinomios. En los problemas de interpolación, se utiliza a menudo la interpolación mediante spline, porque da lugar a resultados satisfactorios requiriendo solamente el uso de polinomios de bajo grado a la vez que se evitan las oscilaciones.

Para el ajuste de curvas, los splines se utilizan para aproximar formas complicadas. La simplicidad de la representación y la facilidad de cómputo de estos, los hacen populares para la representación de curvas en el terreno de los gráficos por computadora.

1.5 Técnicas para la animación:

Existen diversas técnicas para la realización de animaciones 3D, las cuales son increíblemente variadas y difíciles de categorizar. Muchas veces se encuentran relacionadas o combinadas. Entre

ellas se encuentran la skeletal y la morphing⁸ animation, animación esquelética y animación por vértices o formas respectivamente. Ambas técnicas son muy utilizadas en la animación de personajes, la cual se ha convertido en una tarea fascinante para muchos creadores.

1.5.1 Morphing animation:

La morphing animation es una de las técnicas de animación 3D, que se utiliza a veces como alternativa a la animación esquelética. En ella lo que se hace es almacenar un conjunto de posiciones de vértices, tomando como referencias sus coordenadas x, y, z.

Existen ventajas al usar esta técnica, ya que el artista tiene más control de los movimientos, porque puede definir las posiciones individuales de los vértices dentro de un Key Frame. Esto puede ser útil para animar la piel o en expresiones faciales. Pero también existen desventajas, esta técnica generalmente provoca que se consuma mucho tiempo, ya que cada posición de los vértices debe ser especificada.



Fig.4: Área de aplicación de la Morphing Animations.

1.5.2 Skeletal animation:

La animación esquelética es una técnica estándar para la realización de animaciones 3D, particularmente en la animación de los vertebrados, en los cuales un personaje está representado por dos partes: una representación superficial (piel) y una jerarquía de huesos⁶ que se usan para realizar las animaciones (esqueleto).

La utilización de este esqueleto le permite al animador definir una animación con movimientos simples de los huesos, permitiendo que el proceso de creación de las animaciones se torne más sencillo y rápido.

Esta técnica tiene muchas aplicaciones y es utilizada comúnmente en la realización de videojuegos y en la industria del cine, para crear movimientos realistas en las animaciones de personajes articulados.

Se han desarrollado una gran cantidad de técnicas que controlan el movimiento de un esqueleto de una manera que parezca natural al espectador, como lo es la captura de movimiento², con la cual se obtiene más rapidez y realismo en las animaciones.

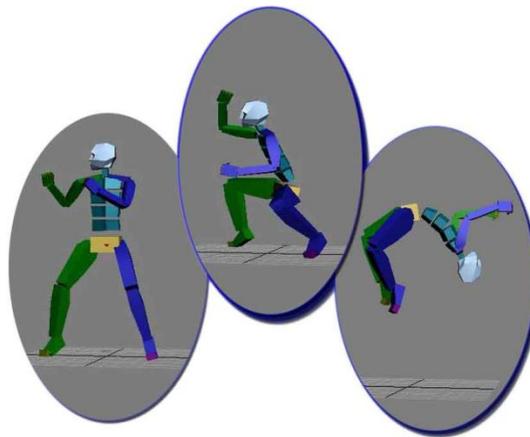


Fig.5: Ejemplo de una animación usando Skeletal Animations

En la Fig. 5 se muestra tres posturas sencillas que se obtienen al mover o rotar cada uno de los huesos del esqueleto o del bípedo en cualquier editor de animaciones.

1.6 Herramientas para la creación de animaciones:

El software como herramienta, es un elemento fundamental para la producción de las interpolaciones necesarias y muchas otras tareas básicas necesarias para la animación.

Existen muchas herramientas informáticas que cubren una gran parte de las necesidades requeridas por los profesionales de las áreas que, de una u otra manera, desean comunicar algo de manera virtual, entre ellas podemos encontrar el 3D Studio Max y el Maya.

1.6.1 3D Studio Max:

Algunas veces llamado 3ds Max o simplemente Max, es uno de los programas de animación 3D más utilizados y respetado en todo el mundo. Es uno de los más sencillos para iniciarse en el mundo de la animación 3D para la creación de video, juegos y multimedia.

Su arquitectura abierta, su baja curva de aprendizaje y sus potentísimas herramientas lo convierten en uno de los programas líderes del diseño y la animación 3D en infinidad de ámbitos, como: arquitectura, publicidad, televisión y video, cine, artes escénicas, desarrollo de juegos, etc.

Dispone de una sólida capacidad de edición y una dinámica arquitectura de plug-ins. Los profesionales del diseño y la animación trabajan con este programa para crear desde juegos de ordenador hasta escenas cinematográficas con espectaculares efectos especiales, anuncios, simulaciones, etc.

El enorme potencial y las infinitas prestaciones de 3ds Max lo convierten en un aliado bastante eficaz, para la realización de todo tipo de diseños y animaciones tridimensionales, ya que permite al usuario una fácil visualización y una sencilla representación de los modelos y

animaciones; utilizando al mismo tiempo cámaras, vistas o visores. Permite además exportar y salvar a varios formatos, incluso diferentes de los que trae por defecto la herramienta.

1.6.2 Maya:

Es otra de las herramientas para el trabajo con animaciones y gráficos 3D. Se caracteriza por su potencia, posibilidades de expansión y la fácil personalización de su interfaz. Dispone de una extensa gama de técnicas específicas para la animación personajes, cinemática inversa y directa que posibilita el control directo e inverso sobre el movimiento de los huesos del personaje.

Con Maya, el animador puede controlar el comportamiento de la piel, incluso en las partes más complicadas del cuerpo, como los hombros. Sus deformadores permiten cambiar la forma de un objeto o personaje, además de integrar sincronización labial, o cualquier otro tipo de movimiento que precise sincronización con el sonido.

MEL (Maya Embedded Language) es el código que forma el núcleo de maya, y gracias a él se pueden crear scripts que aumentan la potencia del software, y permiten su personalización.

En el 2005 se convirtió en el primer y único software que ha obtenido un Oscar, gracias al enorme impacto que ha tenido en la industria cinematográfica como herramienta de efectos visuales.

1.7 Proceso actual de realización de animaciones:

En la actualidad se utilizan herramientas como el 3D Studio Max, Maya y otras para la realización de animaciones de bípedos, las cuales pueden ser desde muy sencillas hasta muy complejas. En la actualidad este tipo de animaciones, son creadas partiendo de un esqueleto y luego se mueven los diferentes huesos del mismo para crear en cada Key Frame una postura diferente y de esta forma obtener el movimiento deseado.

Las animaciones exportadas de cualquiera de las herramientas antes mencionadas contienen muchos más Frames que los definidos por el artista y que de no ser guardados en el fichero, no

afectarían el movimiento. Aunque utilizando otras formas de interpolación, no necesitamos todos esos Key Frames adicionales en el momento de utilizarse en juegos u otras aplicaciones, hacen que el proceso consuma una excesiva cantidad de memoria, que pudiera ser utilizada por otros procesos.



Fig.6: Proceso actual de realización y utilización de las animaciones.

En la Fig.6 como su nombre lo indica se muestra cómo se realiza el proceso actual de realización y posterior utilización de las animaciones de bípedos, las cuales son realizadas en cualquier herramienta o editor de animaciones, se toma la animación que se va a utilizar y se muestran en un visualizador de animaciones o se utilizan en un engine para otros fines.

1.8 Metodologías y herramientas de desarrollo:

Para la realización de la tesis se hizo un estudio de cada una de las posibles metodologías y herramientas a utilizar. A continuación se presenta la metodología que se utilizó y cada una de las herramientas con sus características distintivas que se tuvieron en cuenta para su selección.

1.8.1 Metodología:

La metodología de desarrollo utilizada en la realización de esta Tesis es: RUP (Proceso Unificado de Desarrollo). RUP sirve de guía para realizar el análisis y diseño de la aplicación, debido a que es una metodología que ha probado su efectividad durante muchos años, ya que numerosos proyectos la han utilizado para desarrollar su software. Además, tiene un gran

número de documentos publicados que se pueden consultar para esclarecer dudas. También porque siguiendo sus pasos propuestos se obtiene una buena documentación de la tesis.

A continuación se muestran las características que más influyeron en la selección de esta metodología:

- Guiado por casos de uso: Los casos de uso reflejan lo que los usuarios futuros necesitan y desean, constituyen la guía fundamental establecida para las actividades a realizar durante todo el proceso de desarrollo del sistema.
- Centrado en arquitectura: La arquitectura muestra la visión común del sistema completo. Permite además, implementar el Framework (plataforma sobre la que se implementa el soporte para todas las funcionalidades del sistema) y luego ir desarrollando cada uno de los módulos según se van necesitando.
- Iterativo e Incremental: RUP divide el proyecto en fases de desarrollo, propone además que cada una de ellas se desarrolle en iteraciones, las cuales aportan un incremento en el proceso de desarrollo y terminan con el cumplimiento del punto de control trazado en la fase.
- Utilización de un único lenguaje de modelado: UML.

1.8.2 Herramientas:

1.8.2.1 Enterprise Architect 6.5:

Como herramienta de modelado con UML se seleccionó Enterprise Architect pues brinda una interfaz sencilla y cómoda para trabajar. Incluye todo lo que se necesita para modelar el análisis y diseño de una aplicación, es decir, no hay que instalar varios paquetes para obtener todas sus funcionalidades. Se integra automáticamente con el Visual Studio 2005. Soporta UML 2.1. Además permite combinar diferentes tipos de diagramas para explicar mejor las ideas.

1.8.2.2 Microsoft Visual Studio 2005:

Se vincula con el Enterprise Architect para generar y mantener actualizados los diagramas necesarios para realizar la Tesis. Posee además numerosas herramientas asociadas que ayudan a escribir, analizar y distribuir el código, se integra fácilmente con las versiones de DirectX, es un compilador rápido y con muy buena detección y corrección de errores. Posee facilidad de trabajo con los elementos visuales y buena integración de estos con el código. Contiene muchas librerías con códigos pre-escritos que ayudan en la escritura del código de la aplicación.

1.8.2.3 3D Studio Max 7:

Es una de las más recientes versiones de la solución completa de Discreet para modelado, animación y renderizado 3D avanzados. En todo el mundo los principales estudios de desarrollo de juegos dependen de 3D Studio Max 7 para crear productos de alta calidad y listos para usarse. Es una herramienta interactiva y fácil de manipular por los artistas digitales en la creación de modelos y animaciones 3D.

1.9 Lenguajes:

En el desarrollo de la aplicación se utiliza C++ como lenguaje de programación, para la implementación de cada uno de los paquetes que conforman el diseño de la herramienta.

Para crear la documentación se utiliza UML como lenguaje de modelado, por las potencialidades descriptivas que posee.

1.9.1 Lenguaje de programación:

¿Por qué se escogió C++? Porque posee características superiores a otros lenguajes. Las más importantes son: programación orientada a objetos, portabilidad, brevedad, programación modular y velocidad.

Además, se trata de un lenguaje de programación estandarizado (ISO/IEC 14882:1998), ampliamente difundido, y con una biblioteca estándar que lo ha convertido en un lenguaje

universal, de propósito general, y ampliamente utilizado tanto en el ámbito profesional como en el educativo. También es muy fácil de vincularlo con DirectX; librería que es extremadamente importante en el desarrollo de la aplicación, que es utilizado para cargar desde y hacia el formato .X característico de dicha tecnología y muy usado en el proyecto “Paseo Virtual” para el cual tributa la aplicación.

Otros factores importantes en la selección de este lenguaje, fue el dominio del mismo por los desarrolladores y la recomendación directa por parte del cliente.

1.9.2 Lenguaje de Modelado:

Para modelar el análisis y el diseño del software se escogió el lenguaje UML (Unified Modeling Language, Lenguaje Unificado de Modelación). Esta decisión se debe a que se ha convertido en un estándar que tiene las siguientes características:

- Permite modelar sistemas utilizando técnicas orientadas a objetos (OO).
- Permite especificar todas las decisiones de análisis y diseño, construyéndose así modelos precisos, no ambiguos y completos.
- Puede conectarse con lenguajes de programación (Ingeniería directa e inversa).
- Permite documentar todos los artefactos de un proceso de desarrollo (requisitos, arquitectura, pruebas, versiones, etc.).
- Es un lenguaje muy expresivo que cubre todas las vistas necesarias para desarrollar y luego desplegar los sistemas.
- Existe un equilibrio entre expresividad y simplicidad, pues no es difícil de aprender ni de utilizar.

- UML es independiente del proceso, aunque para utilizarlo óptimamente se debería usar en un proceso que fuese dirigido por los casos de uso, centrado en la arquitectura, iterativo e incremental.

1.10 Consideraciones del Capítulo:

Este capítulo da la posibilidad de obtener los conocimientos necesarios para lograr una buena comprensión de lo referente a la solución que se da en este trabajo para el problema del uso excesivo de memoria que se había presentado en el proyecto “Paseo Virtual” de la facultad 5.

Se brindaron varios conceptos, todos relacionados con la animación, como Frames, Key Frames, interpolación y otros, incluyendo la descripción de algunas de las técnicas y herramientas que se utilizan para la realización de animaciones 3D. Obteniéndose la experiencia necesaria con la cual se llegó a una solución que será expuesta en el siguiente capítulo.

2

CAPÍTULO

SOLUCIÓN PROPUESTA

En este capítulo se describe la solución propuesta al problema del mal uso de memoria y la no existencia de programas que permitan configurar los parámetros necesarios de las animaciones esqueléticas para su posterior uso en un motor de simulación o visualizador de animaciones. Mostrado el flujo de trabajo incluyendo al modificador de animaciones esqueléticas (MODANS) y otros aspectos relacionados con la herramienta ha desarrollar.

2.1 Flujo de Trabajo:

En el flujo de trabajo de creación de animaciones de huesos, agregamos la herramienta propuesta en la tesis, lo que no altera el proceso que se sigue actualmente para la realización y posterior utilización de estas animaciones, solo que la animación final, se acercaría aún a más a la calidad y las exigencias de rendimiento que necesite del artista digital.

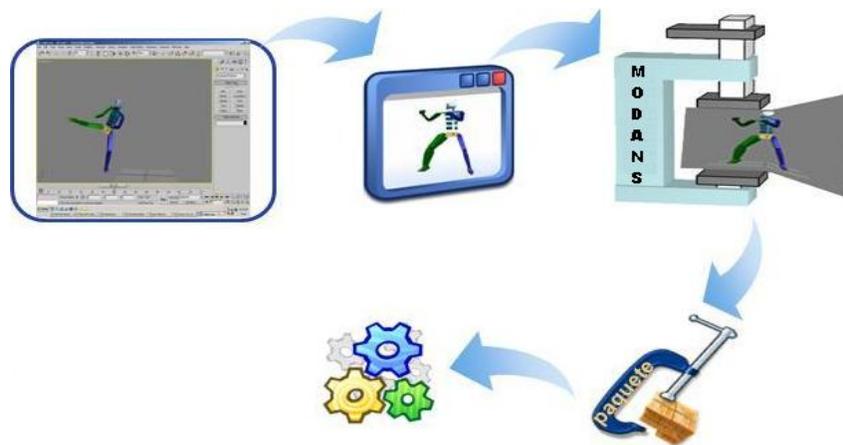


Fig.7: Diagrama del flujo de trabajo.

Como se observa en la Fig.7, el flujo de trabajo se inicia con la creación de la animación del bípedo en cualquier herramienta de diseño, como: 3D Studio Max, Maya o cualquier otro, obteniendo la animación clásica, con la que se ha trabajado en todos estos años de desarrollo del mundo de la Realidad Virtual. Dicha animación será exportada desde el editor con alguno de los formatos estándares más utilizados, entre los cuales podemos encontrar el .X, luego y justo antes de su final utilización en algún engine o visualizador de animaciones, colocaríamos en este flujo la herramienta propuesta, facilitando al artista la obtención de un producto realmente ajustado a sus recursos de hardware.

¿Como sería la realización de este flujo de trabajo con la utilización de una de las herramientas antes mencionadas?

Como respuesta a la pregunta anterior, a continuación se realiza la descripción de este flujo, utilizando la herramienta de diseño 3DStudioMax.

Este proceso se realiza mediante la ejecución de una serie de pasos:

Paso 1: En el panel “Create”, sub-campo “Systems”, se oprime el botón “Biped” y se crea el bípedo.



Fig.8: Creación del bípedo1.

Paso 2: En el panel “Motion”, se encuentran los parámetros del bípedo, dicho panel, permite la creación manual de la animación utilizando cada una de las herramientas que posee el editor, como son la de escalar, rotar y mover como se muestra en la parte izquierda de la Fig. 9 o podemos cargar una animación ya existente mediante el botón “Load File”, donde se abre una ventana, se busca la animación deseada y se oprime “abrir” como se muestra en la parte derecha de Fig. 9.

Paso.2

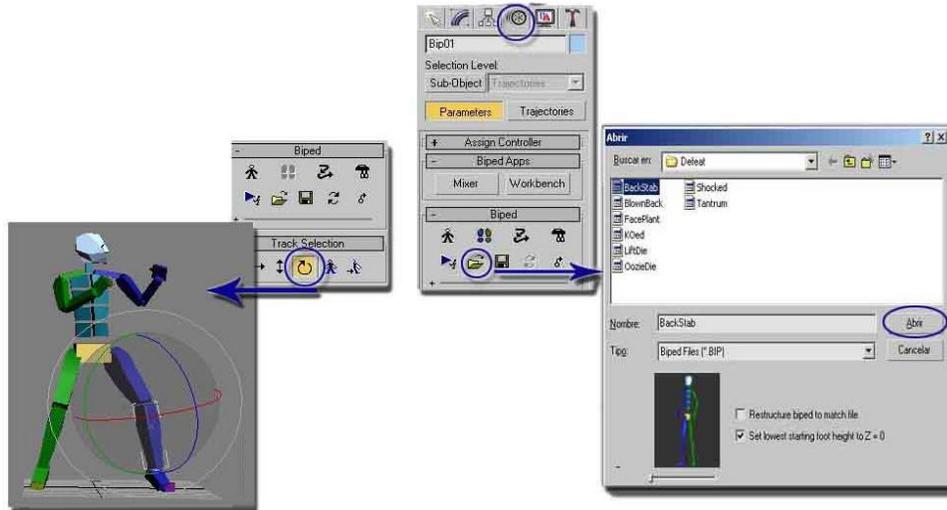


Fig.9: Realización de la animación.

Paso 3: Como último paso se salva la animación en el formato deseado. Esto se realiza en el menú "File", en la opción "Export..." escogiendo el formato al cual deseamos exportar.

Terminado este paso, ya contamos con la animación exportada en el formato deseado, en este caso .X. a partir de este momento ya estamos en condiciones de aplicar la herramienta propuesta (MODANS). Esta herramienta permite aplicarle a la animación, una serie de modificaciones que la ajusten aún más a las posibilidades de hardware y las exigencias del artista digital.

2.2 Requerimientos de entrada:

La herramienta que desarrolla la solución propuesta en la tesis se titula MODANS y para la correcta utilización de la misma se deben de cumplir los siguientes requerimientos de entrada de datos:

- Cada animación a cargar
 - Debe existir

- Debe de estar salvada en alguno de los formatos que carga la aplicación (MODANS).
- Debe contener huesos.
- En su estructura interna debe de aparecer primeramente el mesh⁷ con skinned¹⁵ y luego la jerarquía de huesos.
- Cada hueso que contenga la animación debe estar compuesto por:
 - Una serie de datos, como: Key Frames de escala, rotación, traslación y el nombre de dicho hueso.

2.3 Funcionalidades:

MODANS, permite al artista cargar animaciones en determinado formato y aplicarles cualquiera de los modificadores que la herramienta propone. De este modo obtendrá una animación con las características que realmente necesita y que más se ajusta a sus necesidades, lo que podrá exportar en cualquiera de los formatos que el software permite.

Entre las modificaciones que permite el software están:

2.3.1 Reducir Key Frames innecesarios:

A partir de los Key Frames que el artista define al inicio del proceso de animación, el software genera mediante la interpolación de los mismos, una serie de Frames que conforman a lo largo de la línea de tiempo una animación.

Realizando una serie de interpolaciones se puede determinar, cuales son los Key Frames que de eliminarlos no afectarían la calidad de la animación, teniendo en cuenta un error permisible definido por el artista. Luego de conocer cuales son estos Key Frames innecesarios, se eliminan de la animación.

2.3.2 Priorizar huesos:

Cada hueso dentro de una animación posee mayor o menor importancia en la realización del movimiento que se efectúe en la misma. Si quisiéramos mezclar animaciones, tendríamos que tener en cuenta cuales son los huesos que intervienen de forma más directa en el movimiento realizado, ya que se podrían utilizar los restantes, para otras animaciones sin afectar la original, obteniéndose animaciones mucho más complejas en los personajes y que se ajusten a lo que realmente desea expresar el artista digital con la animación del bípedo.

A cada hueso se le asigna un valor entre 0 y 255 que define la prioridad del mismo, siendo el 0, prioridad mínima y 255, máxima. ¿Cómo lograr esto en animaciones con un número elevado de huesos? Sencillamente se realiza la asignación de prioridad teniendo en cuenta la jerarquía de los huesos en el esqueleto, donde podríamos realizar directamente la asignación de un valor de prioridad a un hueso específico o a un hueso padre⁵, y de este modo todos sus hijos tendrán la misma prioridad.

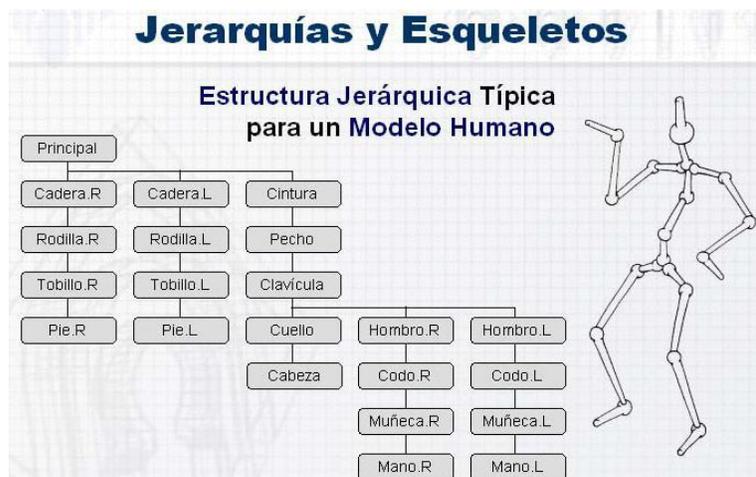


Fig.10: Jerarquía de huesos en un esqueleto.

En la Fig.10 se muestra la jerarquía de huesos a tener en cuenta a la hora de asignar prioridad a cada uno dentro de una animación.

2.4 Consideraciones del Capítulo:

Este capítulo permite tener una idea clara de la solución que se propone, mediante un flujo de trabajo, teniendo como base el proceso actual de realización y posterior utilización de las animaciones que se explicó en capítulo 1.

Se mostraron, además, las funcionalidades de la herramienta, donde se explicó de forma general como van a funcionar cada una de las modificaciones que MODANS realiza a las animaciones.

3

CAPÍTULO

FUNCIONAMIENTO DEL SISTEMA

En el presente capítulo se realizará una detallada descripción del funcionamiento del sistema en el que se incluyen tanto los Requisitos Funcionales como los No Funcionales del mismo, se definen los actores que interactúan con el sistema y se presentan cada uno de los Casos de Usos con las correspondientes descripciones.

3.1 Definición de los Requerimientos Funcionales:

Una de las primeras etapas de cualquier proceso de desarrollo comienza por la tarea de la identificación de los requerimientos tanto funcionales como no funcionales, para el logro de un mejor diseño, modelación y por ende un resultado exitoso.

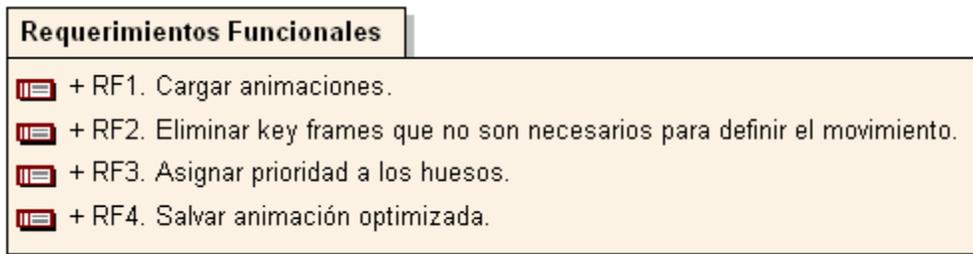
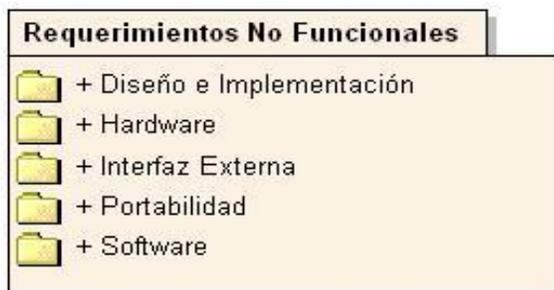


Fig.11: Requerimientos Funcionales.

Teniendo en cuenta los objetivos que se han planteado en este trabajo, en la Fig11 se enumeran los requisitos funcionales o las acciones que debe hacer el sistema para satisfacer los objetivos que se persiguen.

3.2 Definición de los Requerimientos no Funcionales:

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener, como restricciones del entorno o de implementación, rendimiento y otros como los que se muestran en la siguiente figura:



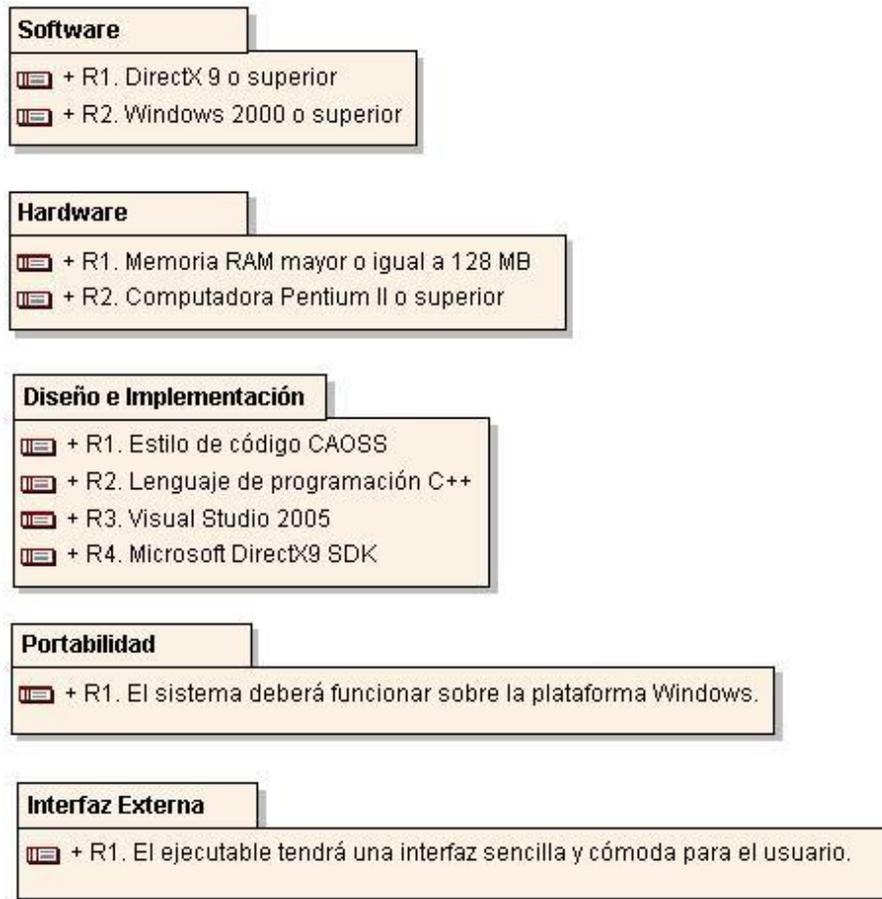


Fig.12: Requerimientos No Funcionales.

3.3 Definición de Casos de Uso:

Para definir las funcionalidades que debe de cumplir la aplicación se elaboraron cinco casos de usos: Cargar Animación, Cargar Mesh, Asignar Prioridad, Reducir Key Frames y Exportar Animación. Cada uno de ellos ayuda a obtener una mejor comprensión del sistema a cualquier interesado en desarrollar la aplicación, independientemente del lenguaje que decida emplear. En la Descripción textual de cada uno de los Casos de Uso se representa además, las interacciones

entre el sistema y el actor, lo que elimina cualquier duda posible en el funcionamiento de la aplicación.

3.3.1 Definición de actores:

Actores	Justificación
Artista Digital	Es la persona que interactúa con la herramienta y al final del proceso obtiene beneficio de ella.

CU_1	Cargar Animación
Actor	Artista Digital
Descripción	Carga animaciones hechas en el 3D Studio Max u otro editor de animaciones.
Referencia	RF1

CU_2	Cargar Mesh
Actor	Artista Digital
Descripción	Carga un mesh hecho en el 3D Studio Max u otro editor de animaciones.
Referencia	RF1

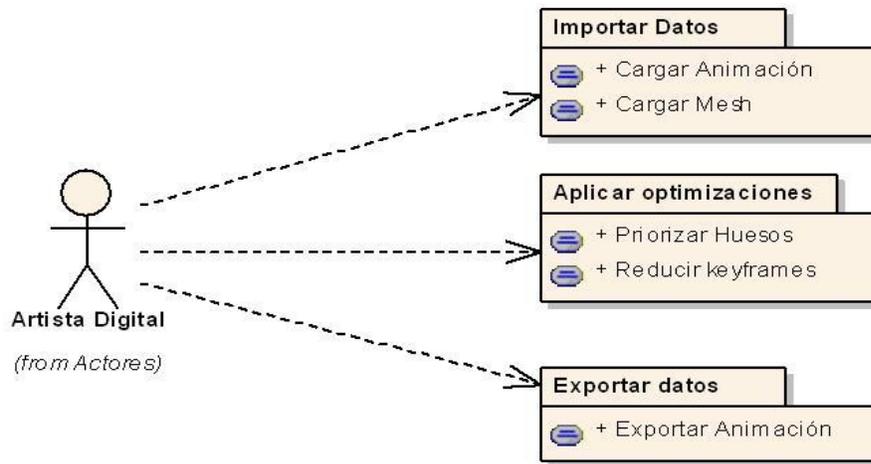
CU_3	Reducir Key Frames
Actor	Artista Digital
Descripción	Eliminar los Key Frames que utilizando fórmulas de

	interpolación no son necesarios para ejecutar el movimiento.
Referencia	RF2

CU_4	Priorizar Huesos
Actor	Artista Digital
Descripción	Asignar prioridad a los huesos referenciados por la animación la animación para otros usos.
Referencia	RF3

CU_5	Exportar Animación
Actor	Artista Digital
Descripción	Salva AI formato especificado por el actor, la animación.
Referencia	RF4

3.3.2 Diagrama de Casos de Uso por paquetes:



En el diagrama anterior se representan cada uno de los paquetes definidos en esta etapa del desarrollo de la herramienta, agrupando en ellos las principales prestaciones del producto deseado.

Este facilita la comprensión del producto que se desea obtener, sin comprometer el futuro desarrollo de la aplicación con un lenguaje o tecnología determinada.

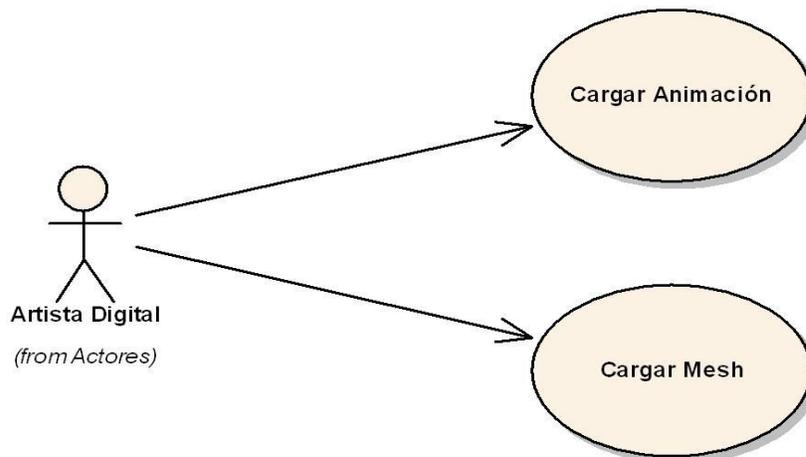


Fig.13: Paquete Importar Datos.

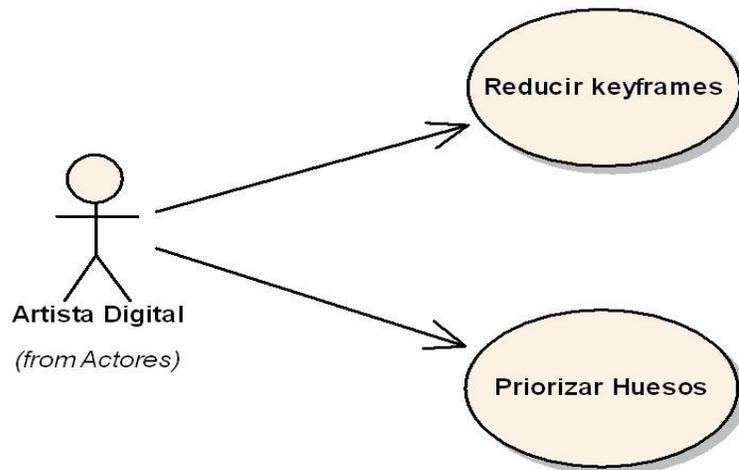


Fig.14: Paquete Aplicar Optimizaciones.

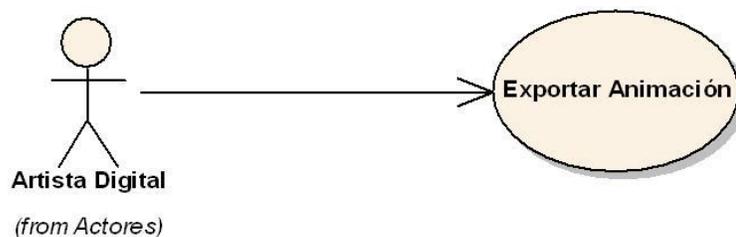


Fig.15: Paquete Exportar Datos.

3.3.3 Descripción textual de Casos de Uso:

Caso de uso	
CU_1	Cargar Animación.
Propósito	Este caso de uso tiene como objetivo cargar animaciones hechas en el 3D Studio Max u otro editor de animaciones.
Actores	Artista Digital.
Resumen	El artista digital inicia el caso de uso cargando los ficheros con formato .X u otro que se haya especificado para el posterior uso de la animación.
Referencias	RF1

Acción del actor	Respuesta del sistema
1. El artista comienza el caso de uso ejecutando la aplicación y selecciona la opción de cargar animación.	1.1 El sistema muestra una ventana de búsqueda.
2. El artista selecciona la animación que desee.	2.1 El sistema verifica si el formato del fichero está entre los formatos que puede cargar. 2.2 El sistema carga la animación.
Flujo alternativo	
Acción del actor	Respuesta del sistema
Acción 2	El sistema muestra un mensaje de error.

Caso de uso	
CU_2	Cargar Mesh.
Propósito	Este caso de uso tiene como objetivo cargar un mesh hecho en el 3D Studio Max u otro editor de animaciones.
Actores	Artista Digital.
Resumen	El artista digital inicia el caso de uso cargando el mesh.
Referencias	
Acción del actor	Respuesta del sistema
1. El artista comienza el caso de uso ejecutando la aplicación y selecciona la opción de cargar mesh.	1.1 El sistema muestra una ventana de búsqueda.
2. El artista selecciona el mesh que desee.	2.1 El sistema verifica si el formato del fichero está entre los formatos que puede cargar. 2.2 El sistema carga el mesh.
Flujo alternativo	
Acción del actor	Respuesta del sistema

Acción 2	El sistema muestra un mensaje de error.
----------	---

Caso de uso	
CU_3	Reducir Key Frames.
Propósito	El objetivo de este caso de uso es eliminar los Key Frames que no sean necesarios para obtener una animación con el nivel de calidad que el artista desee.
Actores	Artista Digital.
Resumen	El artista digital inicia el caso de uso cuando selecciona la modificación a aplicar, luego selecciona una animación y el ángulo de error con que desea que se aplique la modificación.
Referencias	RF2
Acción del actor	Respuesta del sistema
1. El artista comienza el caso de uso seleccionando Reducir Key Frames.	1.1. El sistema solicita que seleccione la animación a modificar.
2. El artista selecciona de la lista de animaciones la que el desee.	2.1 El sistema marca la animación como activa. 2.2 El sistema le solicita el ángulo permisible de error para rotación, traslación y escala.
3. El artista entra los ángulos de error.	3.1 El sistema modifica la animación.

Caso de uso	
CU_4	Priorizar Huesos.
Propósito	El objetivo de este caso de uso es asignar un número entre 0 y 255 en dependencia de la prioridad que tenga dentro de la animación.
Actores	Artista Digital.

Resumen:	El artista digital inicia el caso de uso cuando selecciona la modificación a aplicar, luego selecciona una animación, selecciona el hueso, o los huesos, a los que desea aplicarle la modificación.	
Referencias	RF3	
Acción del actor	Respuesta del sistema	
1. El artista comienza el caso de uso seleccionando Priorizar huesos	1.1 El sistema solicita que seleccione la animación a modificar.	
2. El artista selecciona de la lista de animaciones la que el desea.	2.1 El sistema marca la animación como activa. 2.2 El sistema le solicita que seleccione el hueso, o los huesos a priorizar.	
3. El artista selecciona el hueso, o los huesos y la prioridad que le va a asignar.	3.1 El sistema modifica la animación.	

Caso de uso		
CU_5	Exportar Animación.	
Propósito	El objetivo de este caso de uso es exportar las animaciones modificadas al formato solicitado por el actor.	
Actores	Artista Digital.	
Resumen	El artista digital inicia el caso de uso cuando selecciona exportar animaciones, luego selecciona la animación de la lista de animaciones, escribe el nombre con el que desea salvarla, selecciona el formato deseado y la dirección.	
Referencias	RF4	
Acción del actor	Respuesta del sistema	
1. El artista comienza el caso de uso seleccionando Exportar animación	2.1 El sistema solicita que seleccione la animación a exportar.	
2. El artista selecciona la animación.	2.1 El sistema marca la animación como activa. 2.2 El sistema solicita que entre el nombre, formato	

	y dirección.
2. El artista entra el nombre, formato y dirección.	2.1 El sistema exporta la animación.

3.4 Modelo conceptual:

El modelo de la Fig.16 representa todos los conceptos y las relaciones entre ellos presente en el dominio, lo que es muy importante para lograr una mejor comprensión de lo que se está desarrollando. Entre los principales conceptos que se muestran en el diagrama, encontramos el de Aplicación, que da la idea de una interfaz entre el usuario y los procesos de la aplicación.

El Manejador contendría a los recursos con los que trabajará la aplicación, tales como, los representados por los conceptos de Animación o Esqueleto, además de contar con un arreglo polimórfico¹¹ de tipo Optimización de cuyo tipo de dato heredará (existe herencia⁴) cada una de las posibles modificaciones con las que constara el sistema, definiéndose la de Priorizar Hueso y Reducir KF. El Manejador contaría además de un arreglo de almacenadores del tipo Almacenador, que posee las mismas características del de optimizadores, y cuenta con el concepto de AlmacenadorDX que se encargaría de cargar y exportar recursos desde ficheros de formatos .X de DirectX.

Se muestra la composición de cada uno de los conceptos representativos de los recursos de la aplicación, representando, por ejemplo que cada animación estará compuesta por una serie de Huesos compuestos, a su vez, por KeyFramesR, KeyFramesT, KeyFramesS y cada uno de ellos contendrá datos que describan la Rotación, la Traslación y la Escala.

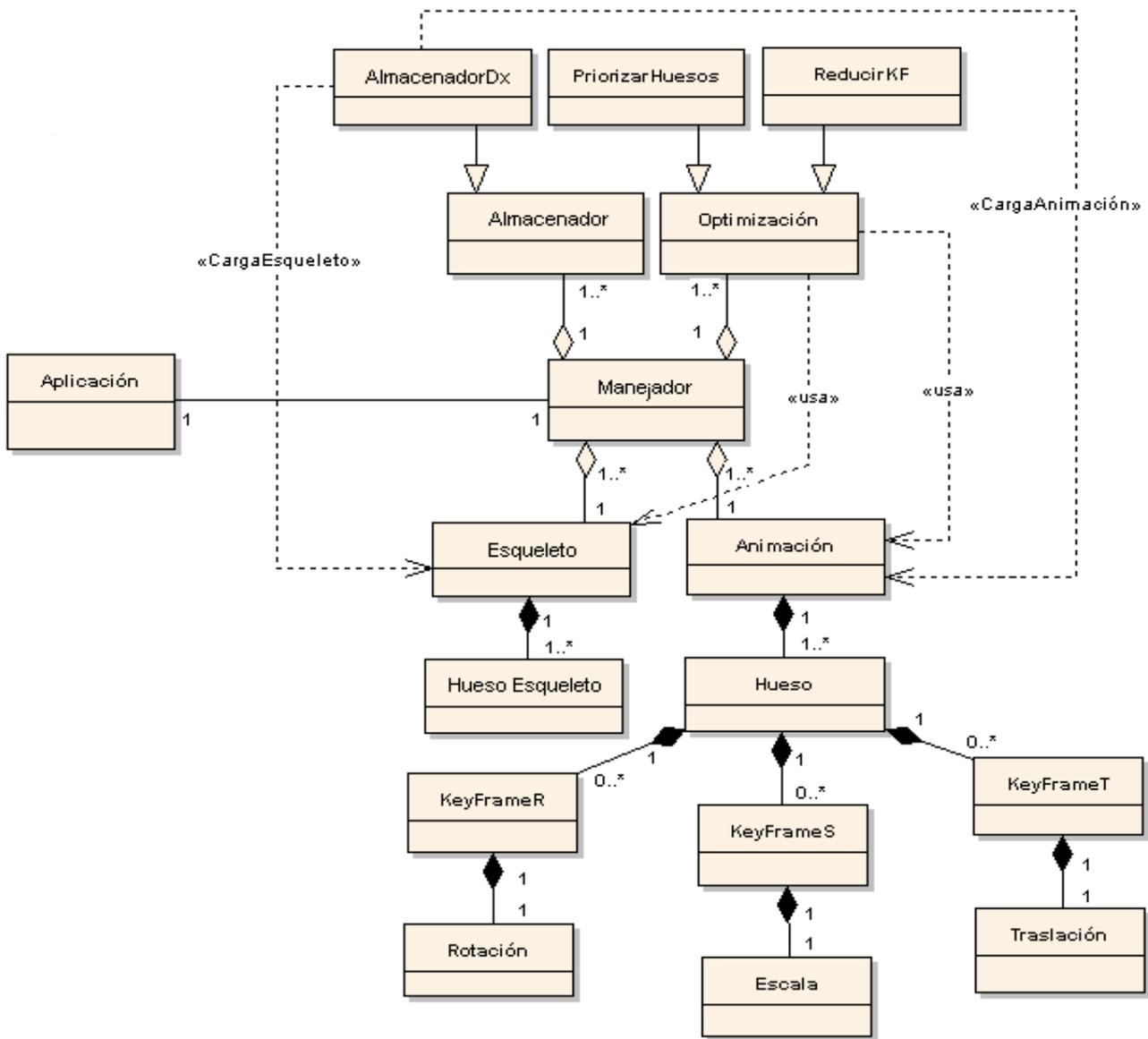


Fig.16: Modelo conceptual.

3.5 Arquitectura del sistema:

En la siguiente figura se representa el Framework, a través de un diagrama de clases del análisis, este muestra el conjunto de las principales clases que conforman el sistema. El Framework es la plataforma sobre la que se implementa el soporte para todas las funcionalidades, por lo que constituye la estructura base del programa.

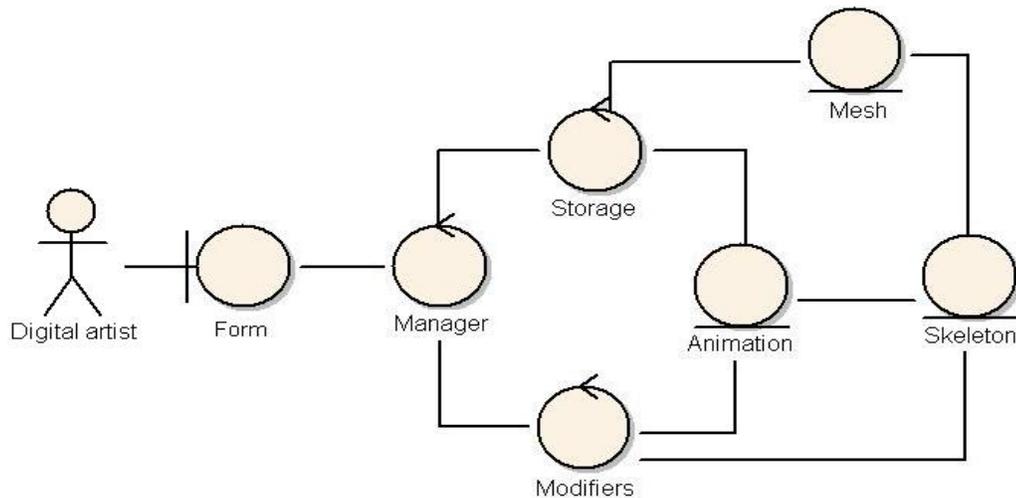


Fig.17: Diagrama de clases del Framework.

El usuario (Digital artist) interactúa con el Framework a través de la clase interfaz “Form”. Esta clase es la encargada de recibir todas las peticiones hechas por el artista digital y de transmitírselas a la clase controladora principal “Manager”, la cual se encarga de clasificar las peticiones y asignar tareas a las demás clases controladoras.

De esta forma, si el usuario desea cargar o exportar, el “Manager” le envía a “Storage” la petición y esta se ocupa de cargar o exportar el fichero con la extensión deseada. Si la petición recibida por el “Manager” es de aplicar alguna optimización en específico, se la envía a “Modifiers”, para que esta a su vez realice los cambios necesarios en la animación (Animation) o en el esqueleto (Skeleton).

3.6 Funcionamiento del Framework:

El artista digital realiza una serie de peticiones al sistema, a través de la interfaz, cada una de estas peticiones, a su vez, generan un flujo de acciones que se representan a través de diagramas de interacción, los cuales pueden ser Diagramas de Colaboración o los Diagramas de Secuencia.

El diagrama de interacción seleccionado para mostrar el flujo de acciones, es el de secuencia, el cual brinda un modelo de cómo interactúa cada objeto del sistema entre sí, esto se puede ver en las figuras 17, 18, 19, 20 y 21 que se muestran a continuación.

3.6.1 Petición inicializar:

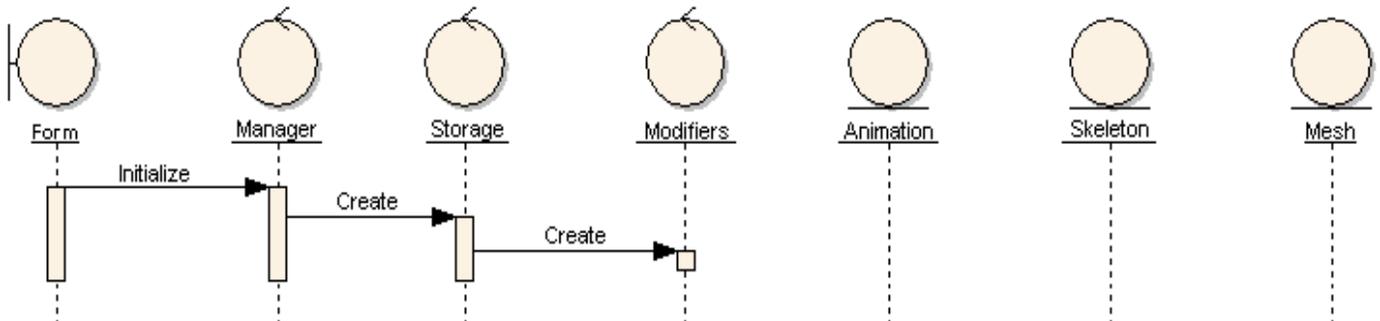


Fig.18: Diagrama de secuencia para la petición Inicializar.

Cuando se abre la aplicación, el “Manager” recibe la petición de Inicializar y crea todos los “Storages”, así como todos los “Modifiers”.

Los “Storages” contienen la información necesaria para cargar y exportar tanto modelos como animaciones. De esta forma, se hace lo más modularmente posible en el sentido de que se puede agregar al sistema cualquier otro formato sin tener que cambiar el código existente, las Optimizaciones se crean de la misma forma, manteniendo dicha facilidad.

3.6.2 Petición Cargar Animación:

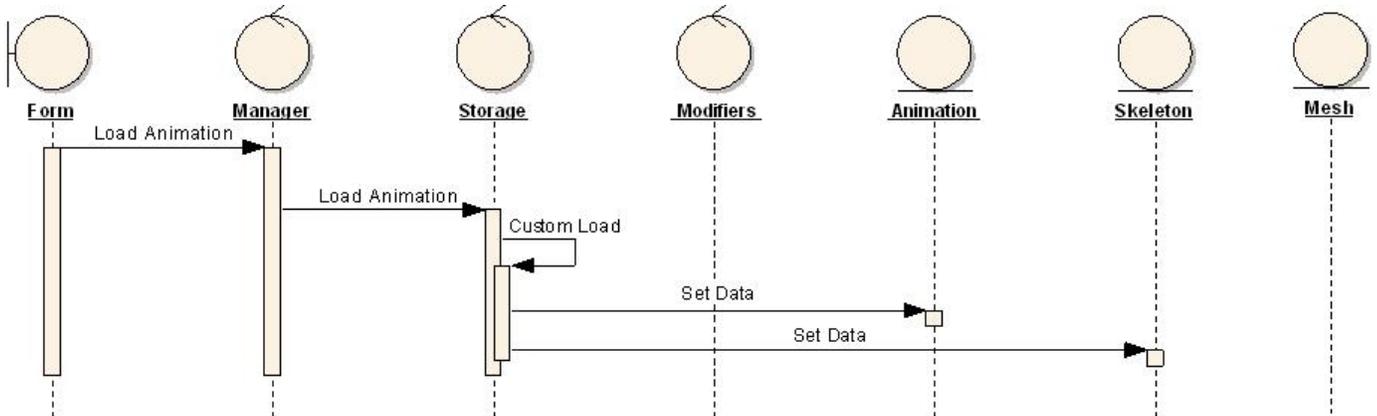


Fig.19: Diagrama de secuencia para la petición Cargar Animación.

Cuando el artista solicita cargar una animación, el “Manager” le envían una petición de cargar una animación al “Storage”, este verifica si alguno de sus storages puede cargar el formato del fichero especificado. En caso de tener algún storage que pueda cargar ese formato, le asigna la realización de la petición. Este storage busca los datos que necesita del fichero y los inserta en la “Animation”.

3.6.3 Petición Cargar Mesh:

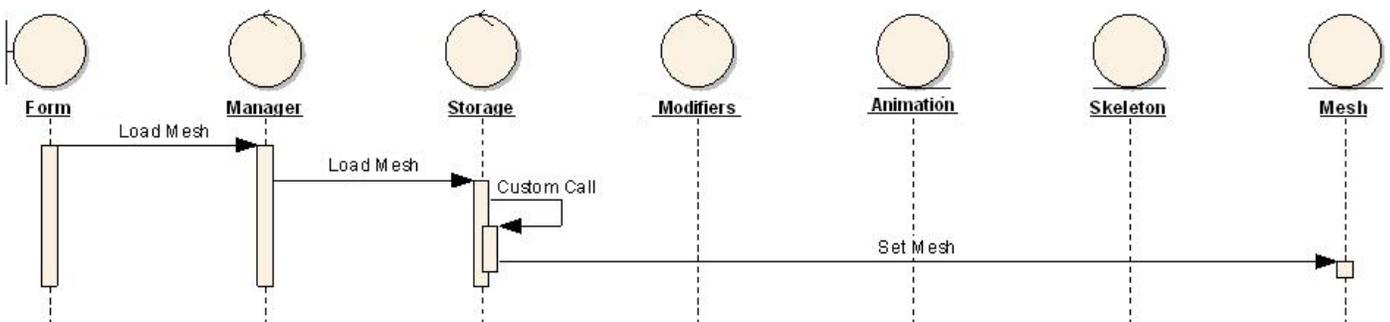


Fig.20: Diagrama de secuencia para la petición Cargar Mesh.

Cuando el artista solicita cargar un mesh, el “Manager” le envían una petición de cargar un mesh al “Storage”, este verifica si alguno de sus storages puede cargar el formato del fichero

especificado. En caso de tener algún storage que pueda cargar ese formato, le asigna la realización de la petición. Este storage carga el mesh.

3.6.4 Petición Aplicar Modificación:

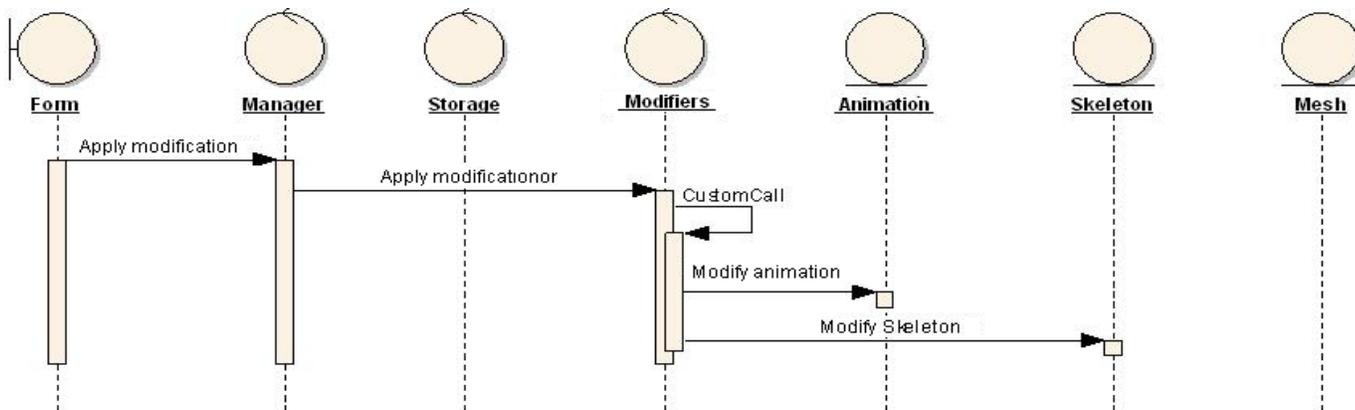


Fig.21: Diagrama de secuencia para la petición Aplicar Modificación.

Cuando el artista digital solicita aplicar modificación, el “Manager” le envían una petición de aplicar modificación al “Modifiers”, este clasifica el tipo de modificación que va aplicar y realiza la modificación seleccionada sobre la animación (Animation) y el esqueleto (Skeleton).

3.6.5 Petición Exportar Animación:

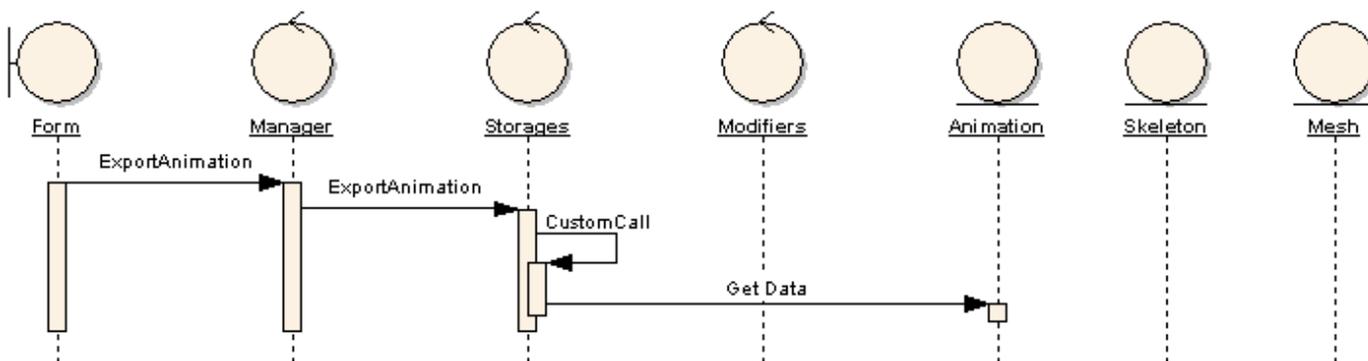


Fig. 22: Diagrama de secuencia para la petición Exportar Animación.

Cuando el artista digital solicita exportar animación, el “Manager” le envía una petición al “Storage” de exportar la animación, diciéndole a este cual es la animación que desea exportar y con que formato hacerlo. El “Storage” consta con una jerarquía, formada por varios Storage especializados en cargar o exportar un tipo de formato en particular. Por lo que según el formato especificado en la petición, será que alguno de estos Storages se encargue de coger la animación y la exportarla al formato deseado.

3.7 Prototipo de Interfaz de Usuario:

La comunicación hombre-computadora es un punto clave para el buen funcionamiento de las aplicaciones. Una interfaz gráfica es una forma de realizar esta comunicación, con lo cual se logra determinar qué quiere hacer el usuario y cómo quiere hacerlo, para obtener el resultado deseado. Sin embargo un punto crucial para el éxito de esta interacción es la sencillez del diseño de la interfaz, de modo que el usuario necesite un mínimo conocimiento para alcanzar aquello que desea, lo que estimula la utilización de la aplicación independientemente del nivel de prestaciones que brinde el sistema.

3.7.1 Interfaz Cargar Animación:

Esta es la interfaz principal que sale cuando se ejecuta la aplicación (Fig.22). Como se puede apreciar aparece marcada por defecto la opción de Cargar Animación y las otras se encuentran deshabilitadas, aclarando que se pueden realizar las demás acciones solo luego de haber cargado la animación a modificar.

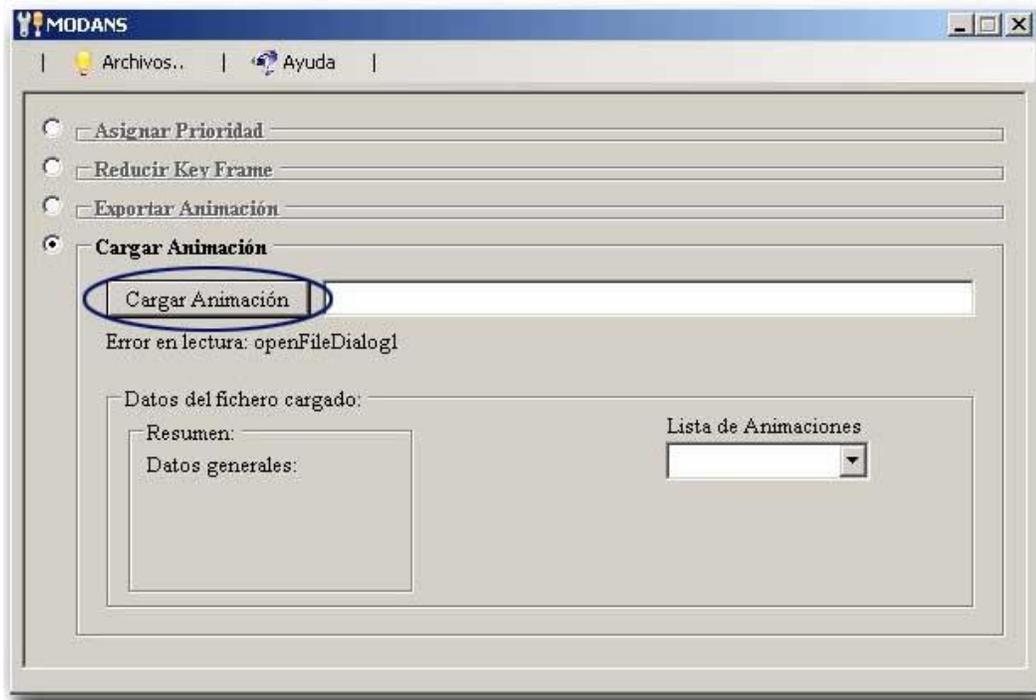


Fig.23: Interfaz Principal.

La acción de Cargar Animación se realiza buscando la animación que se desea mediante el botón “Cargar Animación” (Fig. 23), el cual se deshabilita luego de realizar dicha operación (Fig. 24 [A]). Luego de escoger la animación que se desea se muestra un resumen de los datos de los Key Frames que contiene de cada tipo (Fig.24 [B]) y se habilitan las otras opciones.

Para modificar otra animación, debido a que el botón “Cargar Animación” se encuentra deshabilitado, en el menú Archivos se encuentra la opción Nuevo, la cual permite preparar la aplicación para cargar una nueva animación.

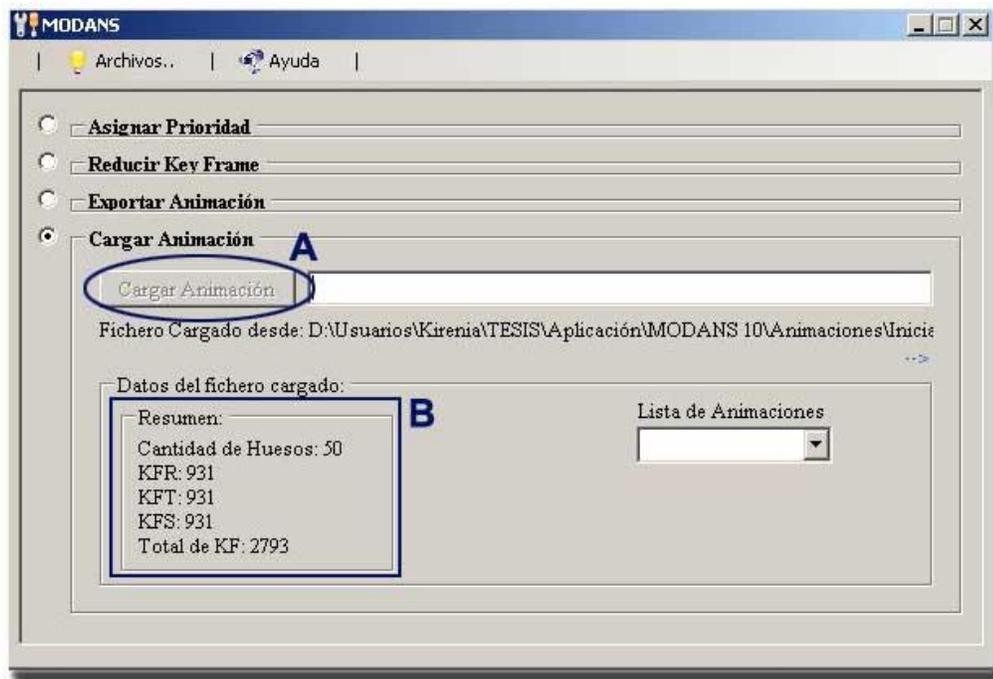


Fig.24: Interfaz Cargar Animación.

3.7.2 Interfaz Asignar Prioridad:

En esta interfaz aparece la opción de seleccionar animación (Fig.25 [A]) con la cual se activa la animación que se desea modificar, se muestran los datos de dicha animación y la jerarquía de huesos que contiene.

Aparece por defecto prioridad media (128). El usuario selecciona la animación a modificar y marca el hueso a el cual desea que se le asigne una prioridad determinada, entre 0 y 255, que debe seleccionar entre las definidas por la aplicación: baja 0, media 128, alta 255 (Fig. 25 [B]) o introducir directamente el valor que desee (Fig. 25 [C]), luego mediante el botón “Asignar Prioridad” (Fig. 25 [D]) el hueso y todos sus hijos en la jerarquía del esqueleto toman el valor de prioridad indicado.

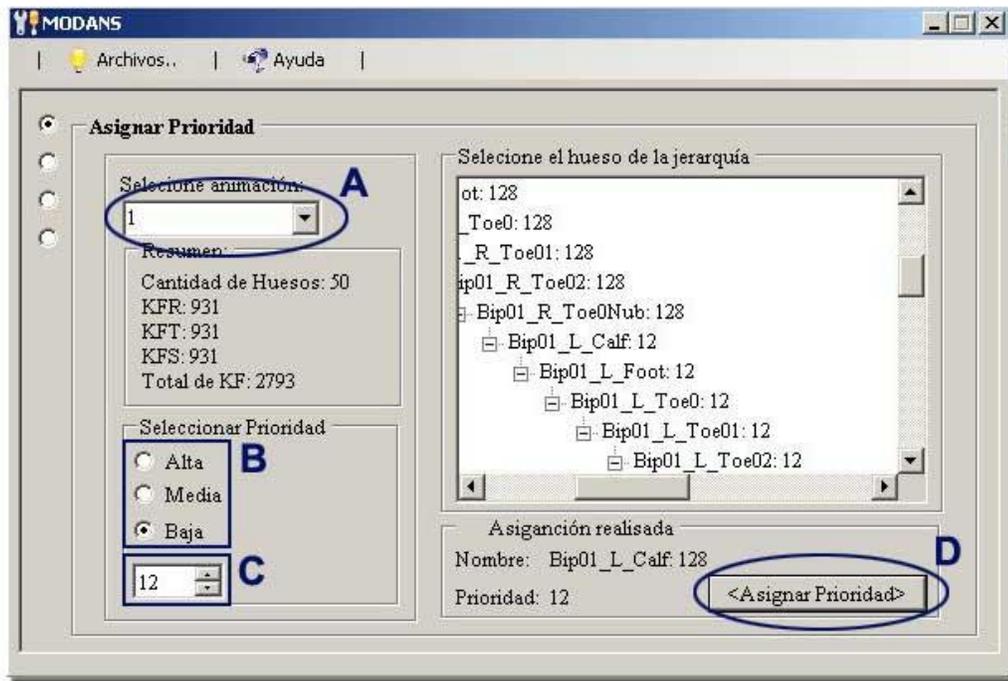


Fig. 25: Interfaz Asignar Prioridad.

3.7.3 Interfaz Reducir Key Frames:

En esta interfaz al igual que la anterior aparece la opción de seleccionar animación (Fig.26 [A]) con la cual se activa la animación que se desea modificar y se muestran los datos de dicha animación.

Para reducir los Key Frames se selecciona un margen de error permisible para rotación, traslación y escala como se muestra en la Fig. 26 [B] y mediante el botón “Reducir Key Frames” (Fig. 26 [C]) se le aplica el modificador a la animación seleccionada previamente y se muestran sus datos en la Fig. 26 [D].

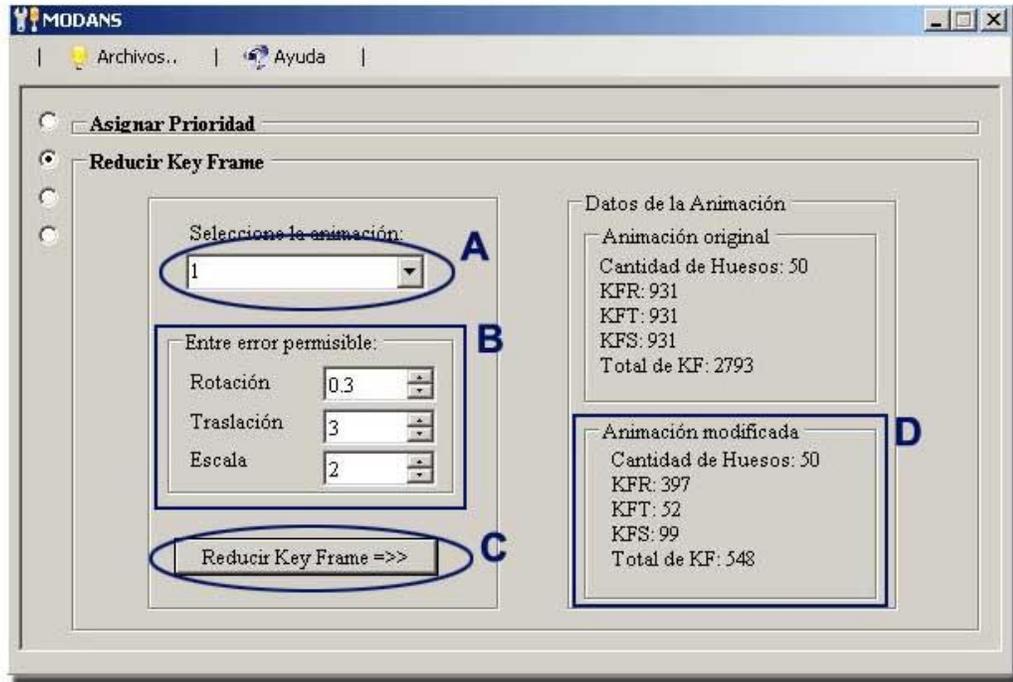


Fig. 26: Interfaz Reducir Key Frames.

3.7.4 Interfaz Exportar Animación:

De la lista de animaciones, se selecciona la que se desee exportar (Fig. 27 [A]), se muestra un resumen de la animación seleccionada como se muestra en la Fig. 27 [B]. Luego mediante el botón "Dirección" (Fig.27 [C]) buscamos la dirección hacia donde se desea exportar la animación. Luego se realizará la acción al presionar el botón de Fig. 27 [D].

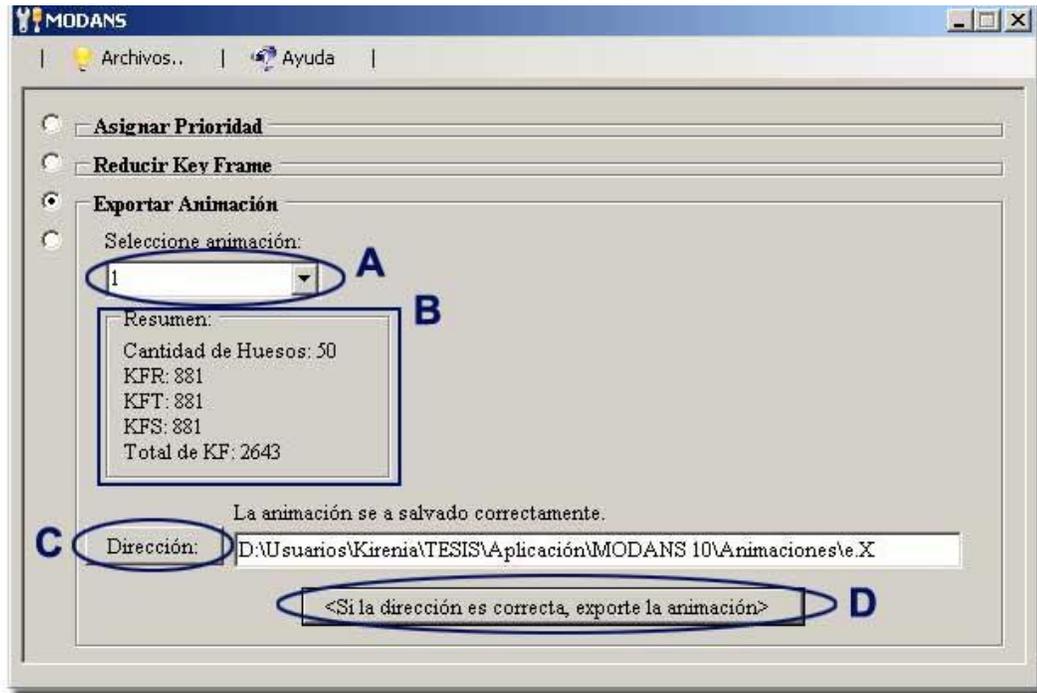


Fig. 27: Interfaz Exportar Animación.

3.8 Consideraciones del Capítulo:

Este capítulo permitió tener una clara definición de los requisitos que deberá cumplir el sistema que se obtenga. Además se definió al Artista Digital como actor, se definieron cinco casos de uso, se expone la arquitectura y la parte más importante fue la concesión de un manual de usuario para lograr una buena comunicación con el usuario.

Se ganó claridad en cuanto al sistema a construir y se sentaron las bases para restantes fases del proceso, como es el caso del diseño que se expone en el siguiente capítulo.

4

CAPÍTULO

DISEÑO DEL SISTEMA

En este capítulo se modela el sistema encontrando un soporte para todos los requisitos, incluyendo los no funcionales y las restricciones que se le suponen. Teniendo en cuenta como entrada esencial el resultado del modelo de análisis. Además impone una estructura del sistema, siendo este una parte importante en el desarrollo de software.

En el capítulo, además, se definen los diagramas de interacción (diagrama de secuencia) de cada caso de uso y diagrama de clases por paquetes, definiendo explícitamente las clases, atributos y métodos que implementarán cada uno de estos casos de uso de la herramienta.

4.1 Diagrama de clases del diseño:

El diagrama de clases, es uno de los elementos más importantes dentro de un proyecto de software, debido a que permite tener una visión bastante clara y completa del sistema, en él se representan las clases del diseño que componen la aplicación y la relación entre ellas. Su diagrama se puede ver en el Anexo 1.

4.1.1 Descripción de las clases:

Nombre: cAnimation	
Tipo de clase: Entidad	
Atributo	Tipo
apBones	vector<cBone*>
pSkeleton;	cSkeleton*
MySkeletonRoot;	D3DXFRAME*
Para cada responsabilidad:	
Nombre:	Set y Get de cada uno de los atributos.
Descripción:	Manejadores de datos que permiten la obtención y variación de cada uno de ellos.
Nombre:	nBones()
Descripción:	Permite obtener la cantidad de huesos con que cuenta la animación.
Nombre:	nKFR()
Descripción:	Cantidad total de Key Frames de rotación en la animación.
Nombre:	nKFT()
Descripción:	Cantidad total de Key Frames de traslación en la animación.
Nombre:	nKFS()
Descripción:	Cantidad total de Key Frames de escala en la animación.
Nombre:	nKF()
Descripción:	Cantidad total de Key Frames en la animación.

Nombre: cMesh	
Tipo de clase: Entidad	
Atributo	Tipo
nFaces	unsigned int
Para cada responsabilidad:	
Nombre:	Set y Get del atributo.
Descripción:	Manejadores de datos que permiten la obtención y variación del atributo.

Nombre: cSkeleton	
Tipo de clase: Entidad	
Atributo	Tipo
Childs	vector<cSkeletonBone*>
Para cada responsabilidad:	
Nombre:	Add(cSkeletonBone* nSBone)
Descripción:	Agregar un nuevo hueso al esqueleto.
Nombre:	ToAssignPriority(string nSkBone, BYTE Priority)
Descripción:	Asignar prioridad al hueso especificado por el número que lo identifica.

Nombre: cSkeletonBone	
Tipo de clase: Entidad	
Atributo	Tipo
Name;	string
nPriority;	BYTE
Child	vector<cSkeletonBone*>
Para cada responsabilidad:	

Nombre:	Set y Get de cada uno de los atributos.
Descripción:	Manejadores de datos que permiten la obtención y variación de cada uno de ellos.
Nombre:	ToAssignPriority (BYTE Priority)
Descripción:	Le asigna la prioridad pasada a través del atributo Priority al hueso.

Nombre: cBone	
Tipo de clase: Entidad	
Atributo	Tipo
BoneName	string
vpTKeyFrame	vector<cRKeyFrame>
vpTKeyFrame	vector<cTKeyFrame>
vpSKeyFrame	vector<cSKeyFrame>
Para cada responsabilidad:	
Nombre:	Set y Get de cada uno de los atributos.
Descripción:	Manejadores de datos que permiten la obtención y variación de cada uno de ellos.
Nombre:	FillDataRST(ID3DXKeyframedAnimationSet* pAnimationSet, Uint iBone)
Descripción:	Rellena el huesos con todos datos de Rotación, Traslación y Escala desde pAnimationSet
Nombre:	PlaceInSameHyperHemisphere(D3DXQUATERNION& qfijo, D3DXQUATERNION& q)
Descripción:	Verifica que los D3DXQUATERNION estén en un mismo hemisferio.
Nombre:	DeleteOneKeyFrame(char TypeKF, Uint pos)
Descripción:	Borrar un Key Frames determinado.

Nombre: cRKeyFrames	
Tipo de clase: Entidad	
Atributo	Tipo
TimePos	unsigned int
qRotacion	D3DXQUATERNION
Para cada responsabilidad:	
Nombre:	Set y Get de cada uno de los atributos.
Descripción:	Manejadores de datos que permiten la obtención y variación de cada uno de ellos.

Nombre: cTKeyFrames	
Tipo de clase: Entidad	
Atributo	Tipo
TimePos	unsigned int
vTastacion	cVector3f
Para cada responsabilidad:	
Nombre:	Set y Get de cada uno de los atributos.
Descripción:	Manejadores de datos que permiten la obtención y variación de cada uno de ellos.

Nombre: cSKeyFrames	
Tipo de clase: Controladora	
Atributo	Tipo
TimePos	unsigned int
vEscala	cVector3f
Para cada responsabilidad:	
Nombre:	Set y Get de cada uno de los atributos.

Descripción:	Manejadores de datos que permiten la obtención y variación de cada uno de ellos.
---------------------	--

Nombre: cManager

Tipo de clase: Controladora

Atributo	Tipo
lpOptimizations	vector<cOptimization*>
lpStorages	vector<cStorage*>
lpAnimation	vector<cAnimation*>
lpEditedAnimation	vector<cAnimation*>

Para cada responsabilidad:

Nombre:	Set y Get de cada uno de los atributos.
Descripción:	Manejadores de datos que permiten la obtención y variación de cada uno de ellos.
Nombre:	Init()
Descripción:	Se encarga de inicializar (crear las optimizaciones y storages).
Nombre:	LoadAnimation(string LAFileName, cManager* pManager)
Descripción:	Busca en su arreglo polimórfico de Storages cual de sus entidades carga el tipo de dato especificado y le pide cargue la animación.
Nombre:	AddAnimation(cAnimation* pAnimation)
Descripción:	Adiciona una nueva animación.
Nombre:	AddStorage(cStorage *Storage)
Descripción:	Adiciona un nuevo storage.
Nombre:	RomoveUnnededKeyFrames(Uint IndexAnimation, Uint IndexOptimization, float ErrorR,float ErrorT,float ErrorS)
Descripción:	Eliminar los Key Frames de la animación especificada a través IndexAnimation, mediante unos de los algoritmos matemáticos especificado por

	IndexOptimization y teniendo en cuenta el margen de error permisible para los cálculos que se realicen con los datos de rotación, traslación y escala.
Nombre:	AssignPriority(cManager* Manager, Uint nAnimation, string SkBone, BYTE)
Descripción:	Le asigna prioridad pasada a través de Priority a un hueso de la animación especificada por nAnimation y la guarda en el arreglo Manager.
Nombre:	ExportAnimation(cManager* pManager, Uint iAnimation, string EAFileName)
Descripción:	Exporta una animación que está en el arreglo de pManager, especificada por y con el nombre EAFileName.
Nombre:	nAnimations()
Descripción:	Devuelve el número total de animaciones.
Nombre:	AddNewEditedAnimation(cAnimation* pAnimation)
Descripción:	Agregar una nueva animación editada.
Nombre:	nEditedAnimations()
Descripción:	Devuelve el número total de animaciones editadas.
Nombre:	DeleteOneAnimation(Uint pos)
Descripción:	Borrar una animación de una posición cualquiera.
Nombre:	nOptimization()
Descripción:	Devuelve el número total de modificadores.

Nombre: cStorage	
Tipo de clase: Controladora	
Para cada responsabilidad:	
Nombre:	LoadMesh(string LMFileName)=0
Descripción:	Mantener común para todos los posibles storges que hereden de ella.
Nombre:	LoadAnimation(string LAFileName, cManager* pManager)=0
Descripción:	Mantener común para todos los posibles storges que hereden de ella.
Nombre:	ExportAnimation(cManager* pManager, Uint iAnimation, string EAFileName)=0

Descripción:	Mantener común para todos los posibles storges que hereden de ella.
---------------------	---

Nombre: cRemoveUnnededKeyFrames	
Tipo de clase: Controladora	
Para cada responsabilidad:	
Nombre:	RemoveUnnededKeyFrames(cAnimation* pAnimation, float ErroR, float ErroT, float ErroS)
Descripción:	Elimina de la animación los KeyFrames teniendo en cuenta los márgenes de errores ErroR, ErroT, ErroS.
Nombre:	EqualToWithErrorForR (cRKeyFrame RKF1,cRKeyFrame RKF2, float Epsilon)
Descripción:	Devuelve si se puede o no eliminar el Key Frame RKF1 si el ángulo entre los quaternion especificados es menor o igual al Épsilon especificado.
Nombre:	EqualToWithErrorForS (cSKeyFrame SKF1,cSKeyFrame SKF2, float Epsilon)
Descripción:	Devuelve si se puede o no eliminar el Key Frame RKF1 si el ángulo entre los vector3f especificados es menor o igual al Épsilon especificado.
Nombre:	EqualToWithErrorForT (cTKeyFrame TKF1,cTKeyFrame TKF2, float Epsilon)
Descripción:	Devuelve si se puede o no eliminar el Key Frame RKF1 si el ángulo entre los vector3f especificados es menor o igual al Épsilon especificado.

Nombre: cAssignPriority	
Tipo de clase: Controladora	
Para cada responsabilidad:	
Nombre:	AssignPriority(cManager* pManager, Uint nAnimation, string SkBone, BYTE Priority)
Descripción:	Asignar la prioridad dada por el atributo Priority al hueso identificado por el parámetro SkBone de la animación especificada por nAnimation.

Nombre: MODANS	
Tipo de clase: Interfaz	
Para cada responsabilidad:	
Nombre:	CargarAnimacion()
Descripción:	Cargar una animación desde una dirección especificada por el usuario.
Nombre:	AsignarPrioridad()
Descripción:	Asigna una prioridad a un hueso determinado en la jerarquía del esqueleto de la animación a modificar.
Nombre:	ReducirKeyFrames()
Descripción:	Reducir los Key Frames de la animación seleccionada por el usuario teniendo en cuenta los márgenes de error para rotación, traslación y escala especificados por el mismo.
Nombre:	ExportarAnimacion()
Descripción:	Exporta una animación seleccionada a una dirección especificada por el usuario.

4.2 Diagrama de clases del diseño por paquetes:

El diagrama de clases del diseño del software que se desarrolla se dividió en tres paquetes como se muestra en la Fig. 27. Esta distribución de las clases por paquetes brinda la posibilidad de descomponer los trabajos de implementación, en partes más manejables que puedan ser llevadas a cabo por diferentes equipos de desarrollo.

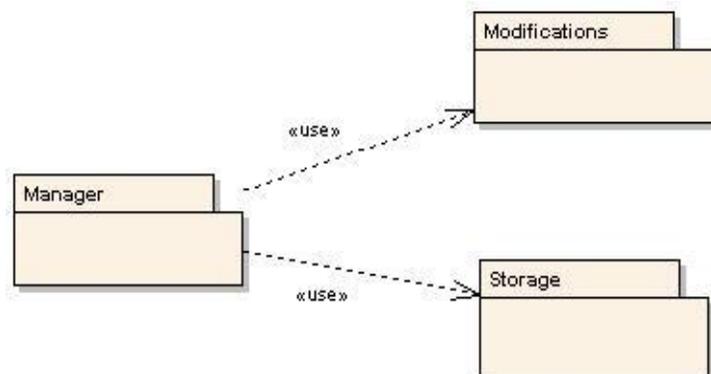


Fig.28: Diagrama de clases del diseño por paquetes.

El sistema consta de tres paquetes como se muestra en la figura, el de Manager, el de Modifications y el de Storage. En el paquete Manager se muestran todas las clases principales, en el Modifications las dos modificaciones que se le aplican a las animaciones y en el Storage se muestran todos los storages a utilizar, en este caso el DxStorage. A continuación se muestran las figuras que representan dichos paquetes y las relaciones que existen entre las clases que contienen.

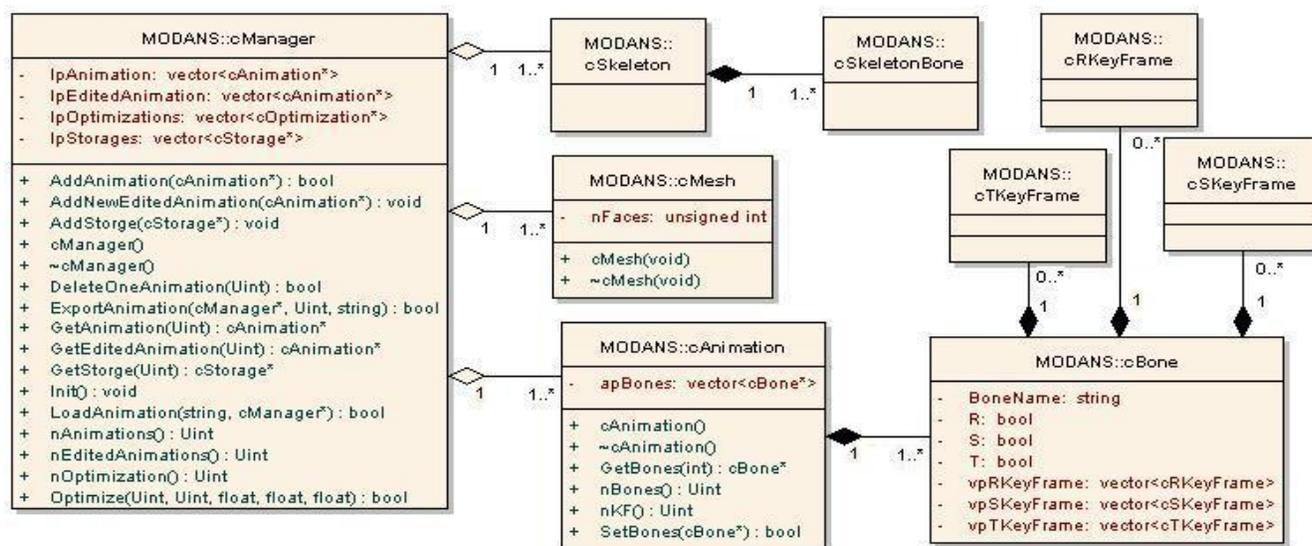


Fig.29: Paquete Manager.

En la Fig.29 se muestra el paquete Manager, el cual es de suma importancia en el diseño de la aplicación, ya que contiene la principal clase manejadora de la herramienta y se puede considerar como la base de la aplicación.

La principal clase del paquete es la clase cManager, la cual consta de un arreglo de todas las animaciones que se carguen en la aplicación, representado mediante la relación de agregación, con la clase cAnimation, la cual contiene una serie de huesos, modelados por la clase cBone y estos a su vez están compuestos por cRKeyFrame, cTKeyFrame, cSKeyFrame.

En el caso de los Key Frame de Rotación, modelados por la clase cRKeyFrame se encuentran como atributos un quaternion encargado de representar los cuatro valores de tipo float en los que se da la Rotación y un valor de tipo int que almacenará el tiempo en el que el está representado, la composición, tanto de Key Frames de Traslación, como los de Escala (cTKeyFrame, cSKeyFrame) está dada por un vector de tres valores de tipo float y uno int para el tiempo. Además de almacenar las animaciones en si, también almacena cada uno de los esqueletos que contienen dichas animaciones, modelados por la clase Skeleton, estos están compuestos por SkeletonBone que almacenan el nombre del hueso y una lista de otros huesos que se derivan de él, llamados hijos o child.

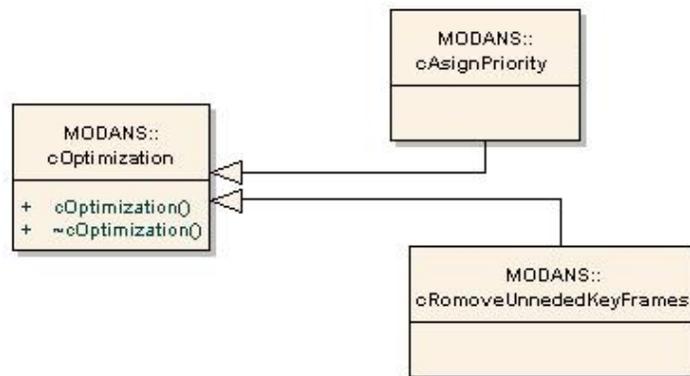


Fig.30: Paquete Modifications.

En el paquete que se muestra en la Fig. 30, se agrupan las clases relacionadas con todas las posibles modificaciones que se le puedan aplicar a los datos.

El paquete consta de una jerarquía de clases dada en su raíz, por la clase base cOptimizations de la que heredan, las clases hijas, cRemoveUnneededKeyFrames y cAssignPriority. Esta estructura jerárquica, permite la inclusión de nuevos modificadores sin afectar el diseño actual, logrando que la herramienta sea enriquecida con aquellas modificaciones que se necesiten en cualquier proyecto y de esta forma lograr solucionar el problema existente, lo cual hace más

extenso su uso, ya que cada una de las nuevas soluciones que se implementen, se transforman en nuevas prestaciones de la aplicación.

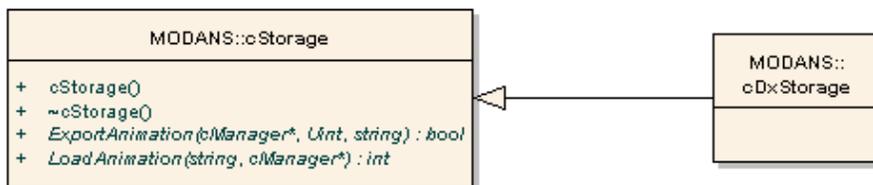


Fig.31: Paquete Storage.

En la Fig.31 se muestra el paquete de todas las clases relacionadas con todas las posibles formas de cargar o exportar datos hacia o desde la aplicación. El paquete consta de una jerarquía de clases dada, en su raíz, por la clase base cStorage, de la cual hereda, la clase hija cDxStorage. Esta organización jerárquica, permite la inclusión de nuevos “storages” sin afectar el diseño actual, logrando que la herramienta sea enriquecida con aquellos storages que se requieran en un proyecto determinado, debido a la gran cantidad de formatos en los que se puede exportar una animación desde cualquiera de los Editores de Animaciones o simplemente porque se desee crear un formato propio para el manejo de las animaciones. Debido a esta necesidad se implementarán frecuentemente varios storage, para la exportación o lectura de un formato determinado y estos pasarán a enriquecer la jerarquía de storage de la Herramienta, incrementando las prestaciones de la Aplicación.

4.3 Descripción de la implementación por Casos de Uso:

En este capítulo también se utilizan los diagramas de interacción, pero esta vez para la descripción de cada Caso de Uso en el diseño, ya que estos diagramas permiten hacer un modelo donde se muestre la interacción que existe entre las clases del sistema, de forma explícita, representando los métodos reales mediante los que interactúan entre ellas.

4.3 .1 Caso de Uso Cargar Animación:

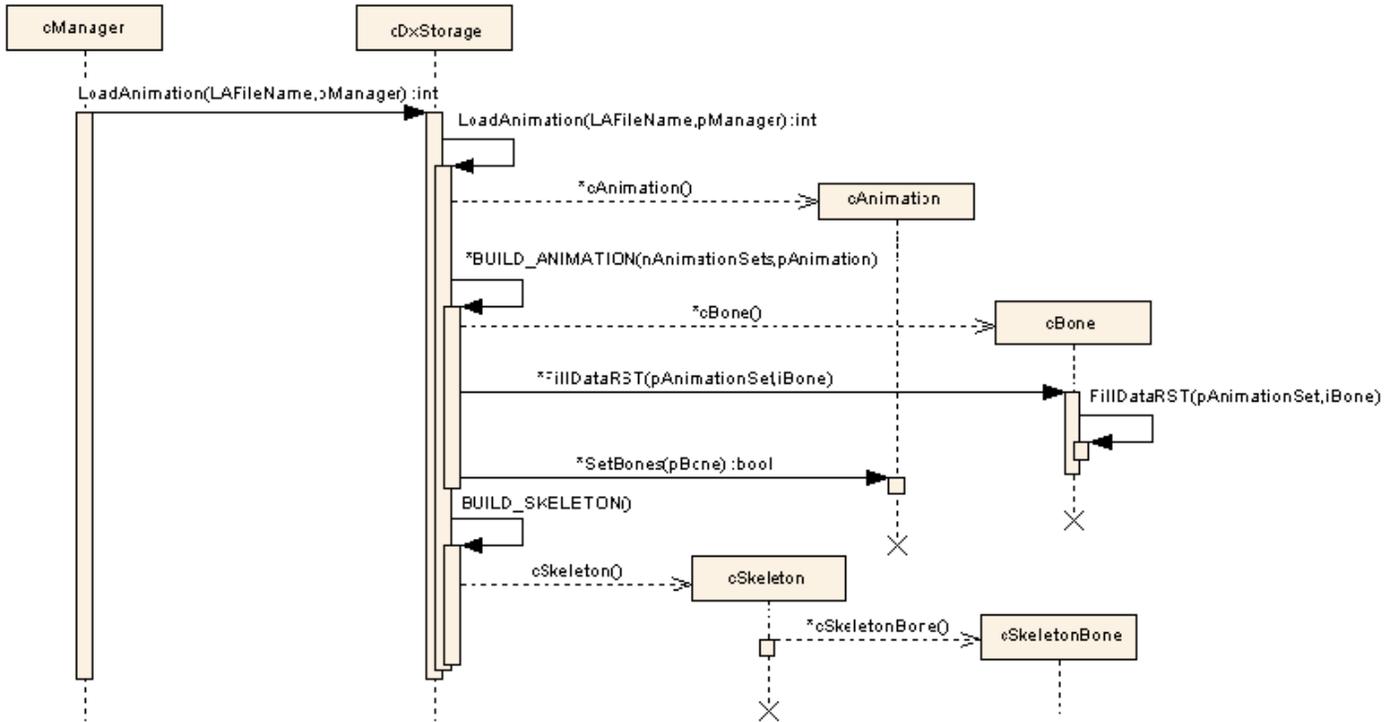


Fig.32: Diagrama de secuencia del caso de uso Cargar Animación.

En este diagrama de secuencia (Fig.32) se muestra como se lleva a cabo el caso de uso Cargar Animación. Este comienza con la petición de la clase *cManager* a la clase *cDxStorage* de cargar una animación mediante la utilización del método *LoadAnimation*. La *cDxStorage* prosigue el proceso, realizando la llamada al método *LoadAnimation*, que mediante funciones definidas por DirectX, carga el fichero indicado por el Artista, ya teniendo estos datos se crean tantas animaciones como se hayan cargado, seguidamente dentro de dicho método (*LoadAnimation*), se llaman a los métodos *BUILD_ANIMATION* y *BUILD_SKELETON*.

El primero de los pasos de este método es la llamada a *BUILD_ANIMATION*, que se encarga de crear todos los huesos de cada animación, mediante el método *cBone*, luego cada hueso, realizando la llamada a *FillDataRST*, se rellena con los key frame que debe contener, después

de estar listo, el hueso se agrega a la animación, con el uso de *SetBone*. Esto se realizará hasta lograr tener listas tantas animaciones como hayan sido cargadas del fichero.

El segundo de los pasos es la llamada a *BUILD_SKELETON*, que se encarga de crear el esqueleto a través del método *cSkeleton* construyendo la jerarquía que posee, con la creación de cada uno de los huesos, de dicha jerarquía mediante el método *cSkeletonBone*.

Al finalizar el caso de uso tendremos todos los datos del fichero cargados en las estructuras definidas en la aplicación: animación y esqueleto.

4.3 .2 Caso de Uso Cargar Mesh:

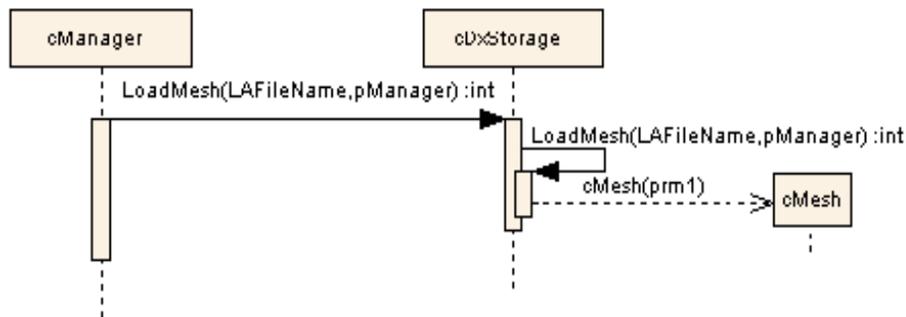


Fig.33: Diagrama de secuencia del caso de uso Cargar Mesh.

En este diagrama de secuencia (Fig.33) se muestra como se lleva a cabo el caso de uso Cargar Mesh. Vale especificar que la clase *cManager* consta de un arreglo polimórfico de tipo *Storage*, el cual tendrá tantos elementos como tipos de *storages* se hayan definido para cargar y exportar, hacia o desde, un tipo de formato específico. En este caso solo estará definido el *storage cDxStorage*, el cual se especializa en realizar las operaciones de cargar y exportar datos, desde o hacia ficheros *.X*.

El caso de uso comienza con la petición de la clase *cManager* a la clase *cDxStorage* de cargar un *mesh* mediante la utilización del método *LoadMesh*, el caso de uso tendrá continuidad a partir de la clase *cDxStorage*, con la ejecución del método *LoadMesh* y dentro de dicho método se

carga el mesh deseado desde dirección dada como parámetro, desde un fichero .X y a continuación se ejecuta método *cMesh*, que crear y rellenar con todos los datos necesarios la entidad del mismo tipo, con la que trabajará el sistema.

4.3 .3 Caso de Uso Reducir Key Frames:

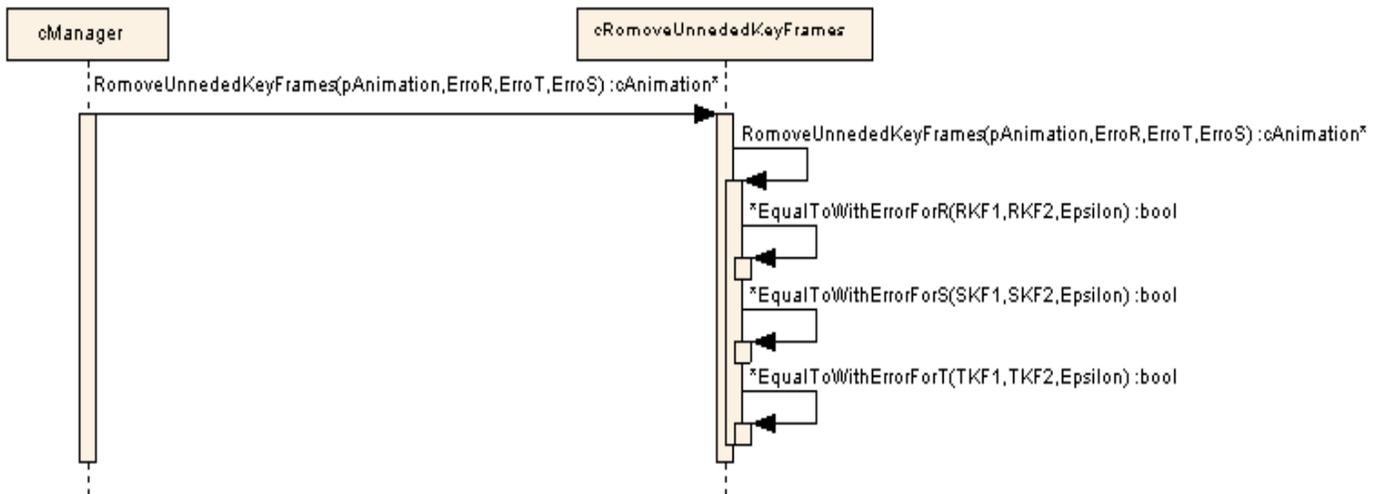


Fig.34: Diagrama de secuencia Reducir Key Frames.

En este diagrama de secuencia (Fig.34) se muestra como se lleva a cabo el caso de uso Reducir Key Frame. Este comienza con la petición de reducir una animación determinada, de la clase *cManager* a la clase *cRemoveUnnededKeyFrame*, con los datos del error permisible para cada tipo de Key Frame, tanto de Rotación, Traslación o Escala mediante el método *RemoveUnnededKeyFrame*.

La clase *cRemoveUnnededKeyFrame* prosigue el caso de uso con la llamada al método *RemoveUnnededKeyFrame*, que se encarga, mediante un algoritmo de interpolación, calcular los Key Frames intermedios y determinar cuales de ellos pueden ser removidos del fichero y esto se realiza con la ayuda de los métodos *EqualToWithErrorForR*, *EqualToWithErrorForS*, *EqualToWithErrorForT* que determinan cual Key Frame, tanto de Rotación, Traslación o Escala

debe de ser eliminado y cuales deben de conservarse para llevar la animación al punto definido por el artista con la entrada de los parámetros correspondientes al error permisible.

4.3 .4 Caso de Uso Priorizar Huesos:

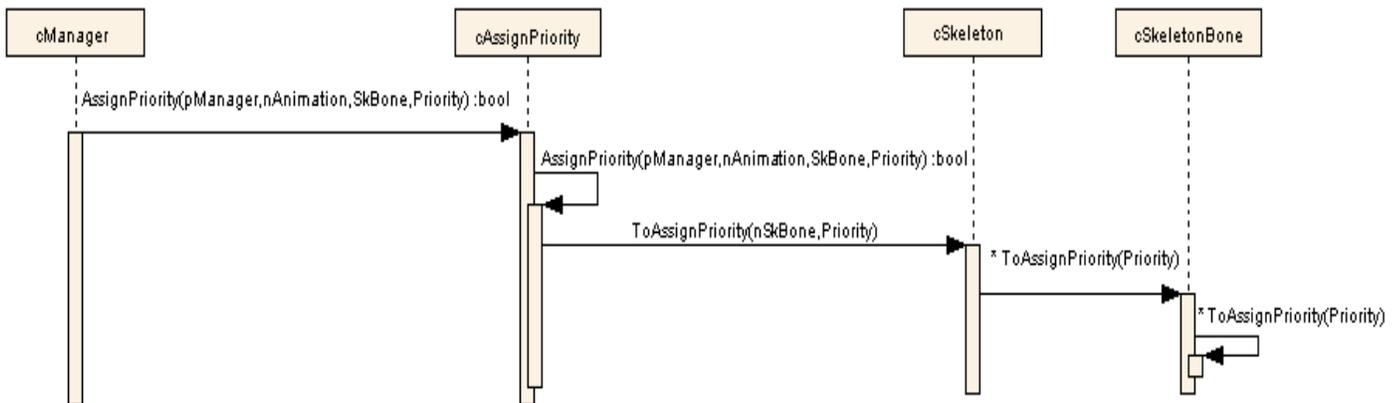


Fig.35: Diagrama de secuencia Asignar prioridad a huesos.

En este diagrama de secuencia (Fig.35) se muestra como se lleva a cabo el caso de uso Asignar Prioridad. Este comienza con la petición de la clase cManager a la clase cAssignPriority de asignar un valor de prioridad determinado a un hueso del esqueleto, mediante el método *AssignPriority*. La clase cAssignPriority da continuación al caso de uso con la ejecución de su método *ToAssignPriority* que se encarga de decirle al esqueleto que debe de asignar un valor de prioridad a uno de sus huesos y a todos sus descendientes, en caso de tenerlos, en la jerarquía.

La clase cSkeleton realiza la asignación de prioridad de forma recursiva mediante la llamada del método *ToAssignPriority*, tantas veces como huesos contenga el esqueleto, luego, cada hueso llama a *ToAssignPriority* para asignar, mediante el uso de la recursividad, la prioridad que se pasó como parámetro, a sí mismo y a sus hijos, en caso de tenerlos.

4.3 .5 Caso de Exportar Animación:

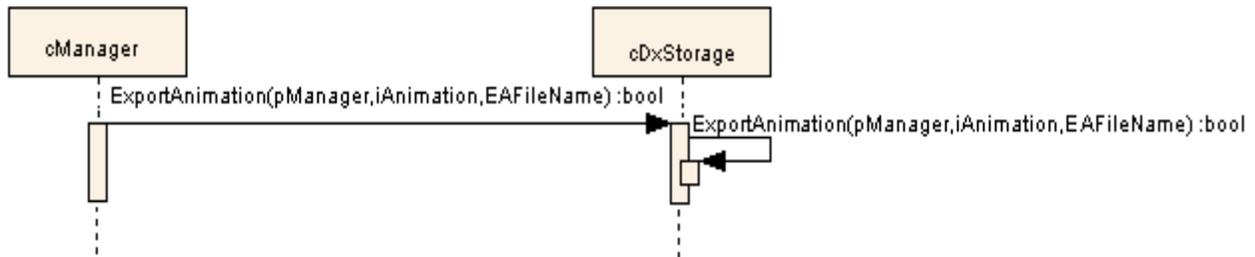


Fig.36: Diagrama de secuencia Exportar Animación.

En este diagrama de secuencia (Fig.36) se muestra como se lleva a cabo el caso de uso Exporta Animación. Este comienza con la llamada en la clase cManager al método *ExportAnimation*, el cual actúa sobre el arreglo polimórfico que posee dicha clase, buscando cual de los storage será el encargado de realizar dicha operación, en este caso, debido a que hasta el momento en el único storage definido es el cDxStorage este sería seleccionado para dar respuesta a la llamada de la clase cManager. La clase cDxStorage continúa el caso de uso con la ejecución del método *ExportAnimation*, el cual realiza la operación deseada al formato en que dicha clase se especializa y que fue especificado por el usuario al inicio del caso de uso.

4.4 Consideraciones del capítulo:

Este capítulo permitió guiar la implementación de la aplicación, mediante la realización del diseño del software, en el que se especifican cada una de las clases, métodos y las relaciones que entre ellos se establecen, para dar solución al prototipo de aplicación que se planteó como objetivo principal en el trabajo.

5

CAPÍTULO

RESULTADOS

Este capítulo muestra todos los resultados obtenidos mediante la realización de varias pruebas sobre una muestra de cinco animaciones. Con estas pruebas se llegó a la obtención de una serie de valoraciones que definen las potencialidades de reducción de Key Frames, el porcentaje de reducción, la cantidad de KB ahorrados o la asignación de un valor de prioridad a un hueso determinado.

Luego de terminada la aplicación se realizaron una serie de pruebas para constatar la funcionalidad, en cuanto a las potencialidades de reducción de Key Frames, el porcentaje de reducción o la cantidad de KB ahorrados y la asignación de un valor de prioridad a un hueso determinado. Con este fin se seleccionó una muestra aleatoria, conformada por cinco animaciones guardadas en ficheros de formato .X. Se comenzó con pruebas de asignación de prioridad a los huesos de la animación a modificar, con valores determinados, obteniéndose la modificación en la prioridad de los huesos de esqueleto esperada.

5.1 Tabla de pruebas realizadas al software:

5.1.1 Pruebas de Asignación de Prioridad:

Nombre del hueso	Bip01_Footsteps
Prioridad Inicial	128
Prioridad asignada	0
Ver resultados en Fig.	Fig. A

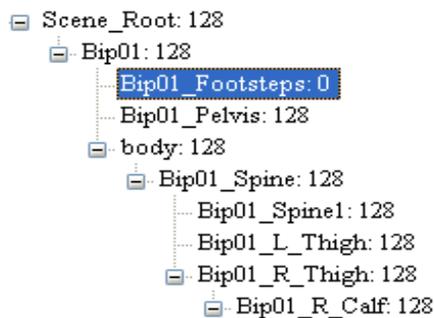


Fig. A

Nombre del hueso	Bip01_Pelvis
Prioridad Inicial	128
Prioridad asignada	150
Ver resultados en Fig.	Fig. B

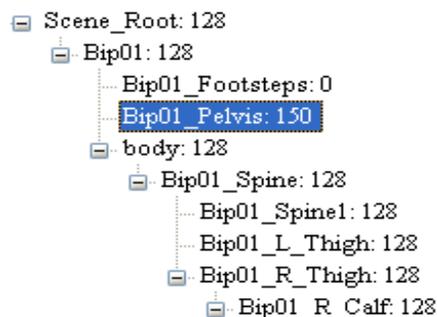


Fig. B

Nombre del hueso	body
Prioridad Inicial	128
Prioridad asignada	225
Ver resultados en Fig.	Fig. B

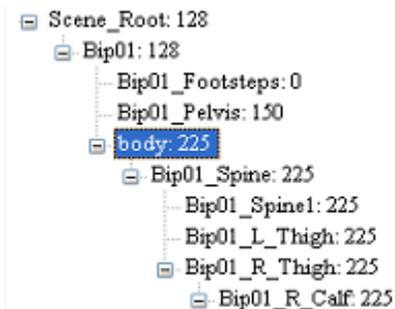


Fig. C

En las figuras A y B se muestra como la asignación se realiza solo al hueso seleccionado, debido a que estos no poseen hijos en la jerarquía de huesos del esqueleto, no siendo así en el resultado mostrado en la Fig. C, ya que a partir del hueso seleccionado se deriva el resto de la jerarquía, por lo que cada uno de esos huesos tomarán su mismo valor de prioridad.

5.1.2 Pruebas de Reducción de Key Frames:

Luego de efectuadas las pruebas con el modificador de asignación de prioridad que propone la herramienta, se analizaron una serie de resultados obtenidos de las pruebas de reducción de Key Frames realizadas a cada una de las animaciones de la muestra. (Ver: Anexo 2)

Nombre de la animación	Resultados	
	Reducción en KB	Porcentaje de KF reducidos
Inicial Tauren.X	4.629	8.486
Inicial MujerCaminando.X	38.379	6.112
Inicial EvaMeneo.X	30.664	5.150
Inicial Diablito.X	8.887	24.542
Inicial Personaje.X	4.980	7.981
Promedios	17.508 KB	10.454%

Tabla 1: Resumen de las pruebas realizadas con valores de error permisibles de: KFR = 0.001, KFT = 0, KFS = 0 a cada animación.

La tabla anterior muestra que la aplicación del modificador a una animación, incluso con valores mínimos, alcanza un porcentaje de reducción promedio de 10.45, lo que representa la cantidad de KF que poseen valores semejantes en la animación, que no aportan datos significativos a la misma y que de ser eliminados no afectarían la calidad de la misma.

Nombre de la animación	Resultados	
	Reducción en KB	Porcentaje de KF reducidos
Inicial Tauren.X	37.769	81.489
Inicial MujerCaminando.X	587.558	93.568
Inicial EvaMeneo.X	528.672	88.79
Inicial Diablito.X	27.683	89.213
Inicial Personaje.X	43.918	82.629
Promedios	245.12 KB	87.138%

Tabla 2: Resumen de las pruebas realizadas con valores de error permisibles de: KFR = 0.001, KFT = 0, KFS = 0 a cada animación.

Para la selección de los valores de cada uno de los errores permisibles (Rotación, Traslación y Escala), para la realización de las pruebas de las cuales se deriva la anterior tabla, fueron seleccionados con la condición de que se pueda asegurar que cualquiera de las animaciones de la muestra, al aplicársele el modificador, mantiene la calidad de sus movimientos intacta.

Conociendo que la animación no es visiblemente afectada en ninguno de los casos, se puede valorar la significativa reducción que muestran los resultados, ya que se logra un 87.14 % de reducción del total de Key Frames de la animación lo que se puede traducir, además, como una disminución media de 245 KB, por animación. Si analizamos que un juego puede tener valores superiores a las 500 animaciones y tomando este valor como prueba se puede afirmar, por los resultados antes expuestos, que se estarían ahorrando alrededor de 122500 KB lo que representa unos 119.63 MB que pueden ser utilizado en disimiles procesos.

Basado en los datos de los resultados mostrados en el Anexo 2, se puede realizar comparaciones entre valores significativos como los que brindan la cantidad de KB iniciales y la cantidad de KB finales, como se muestra en la siguiente tabla:

Nombre de la animación	Cantidad de KB iniciales	Cantidad de KB finales
Inicial Tauren.X	47.278	9.509
Inicial MujerCaminando.X	209.316	39.144
Inicial EvaMeneo.X	198.477	66.245
Inicial Diablito.X	12.07	3.701
Inicial Personaje.X	20.8	10.162

Tabla 3: Comparación referente a la cantidad de KB inicial y final de cada animación.

A partir de los valores de la tabla anterior se generó el gráfico representado en la Fig.36 donde se muestra la diferencia existente entre la cantidad de KB iniciales de cada una de las cinco animaciones al ser cargada por la herramienta y la cantidad final en KB luego de aplicado el modificador de “Reducir Key Frames” propuesto por la herramienta.

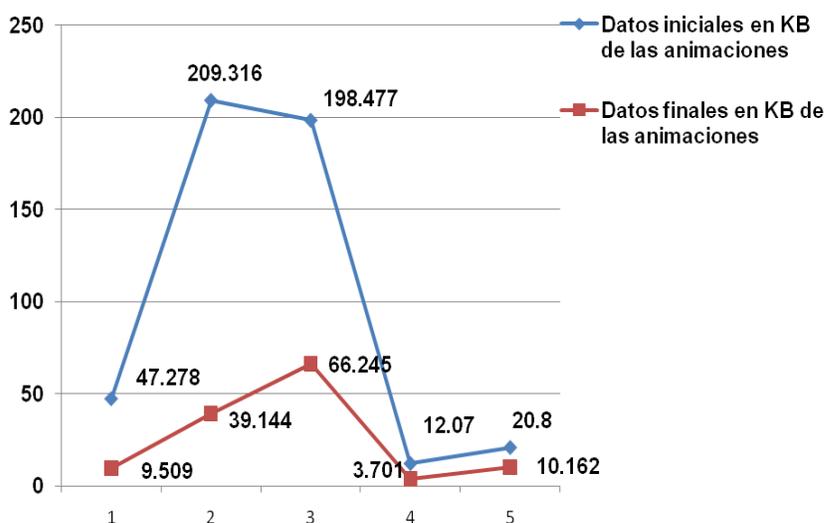


Fig. 37: Comparación referente a la cantidad de KB inicial y final de cada animación.

Con cada uno de los valores seleccionados para realizar las pruebas a la aplicación en cuanto a la reducción de Key Frames, se obtienen resultados que promedian un 87.138% en la reducción de los Key Frames en las animaciones resultantes o finales, lo que equivale a una reducción promedio de 245.12 KB por animación.

Los resultados expuestos, fueron obtenidos a partir de valores para los que se asegura una que no se afecta en ningún sentido la calidad del movimiento de la animación a modificar y que en la realización de modificaciones a una animación en particular, estos resultados puede alcanzar valores superiores de reducción de Key Frames, KB y en el porcentaje de reducción, ya que se pueden llevar al extremo los valores de error permisibles, tanto en Rotación, Traslación y Escala, siempre y cuando los resultados finales de la animación satisfagan los intereses del usuario.

5.2 Consideraciones del capítulo:

Este capítulo permite conocer cuales son las verdaderas posibilidades de MODANS en cuanto al porcentaje de reducción tanto de Key Frames como de KB, se puede probar que la aplicación de la herramienta, a una animación determinada, conlleva a una reducción considerable del consumo de memoria y logra ajustar la animación a los intereses propios del usuario.

Prueba de esto son los resultados alcanzados luego de la realización de una serie de análisis, los cuales se muestran en las tablas 1, 2 y 3 que permiten comprobar las potencialidades de reducción de Key Frames, el porcentaje de reducción o la cantidad de KB ahorrados que posee MODANS.

CONCLUSIONES

El proyecto "Paseo Virtual" de la facultad 5 desarrolla diferentes aplicaciones que en su mayoría necesitan animaciones 3D, pero debido al problema de usos excesivo de memoria al cargar las animaciones y la necesidad de modificar o incluir parámetros para facilitar el posterior trabajo con cada una de las animaciones, se decidió trazar como objetivo principal de este trabajo la realización de una investigación relacionada con el tema y la construcción de un prototipo de aplicación capaz de resolver los problemas existentes.

Una vez culminada la investigación y el desarrollo del sistema que se propuso, se llegó a la conclusión de que el objetivo fundamental de éste se logró. El Modificador de Animaciones de Huesos (MODANS) permite analizar y transformar los valores necesarios de las animaciones esqueléticas exportadas desde cualquiera de las herramientas de diseño, como el 3D Studio Max o Maya, mediante los modificadores que propone, ajustando cada animación a los intereses del proyecto y alcanzando valores significativos de reducción de Key Frame y KB en las animaciones modificadas por la herramienta, mostrando además las potencialidades de la aplicación en cuanto a extensibilidad, ya que permite la inclusión de nuevos modificadores o exportadores que respondan a los intereses particulares o comunes de proyectos o grupos de desarrollo donde se pueda aplicar la herramienta, con el fin que se necesite.

RECOMENDACIONES

- 1- Se recomienda el uso de la herramienta (MODANS) en juegos y simulaciones virtuales, debido a las ventajas que brinda referentes al ahorro de memoria y su capacidad de ajuste a las necesidades de cualquier proyecto o grupo de trabajo.
- 2- Continuar el desarrollo de la aplicación con la implementación de cualquier modificador o almacenador que satisfaga las necesidades particulares o generales de cualquier proyectos o grupos de desarrollos que utilicen animaciones de huesos con cualquier fin.
- 3- Continuar la investigación y el desarrollo de diferentes algoritmos de reducción de Key Frames, para optimizar cada vez más el uso de la memoria.
- 4- Crear nuevas aplicaciones enfocadas a la optimización o disminución del uso de los distintos recursos de hardware, como el uso memoria de video o el volumen de almacenamiento en disco duro³.

BIBLIOGRAFÍA

Conferencia:

- ESPECIALIDAD, D. *Flujo de trabajo Análisis y Diseño. (Modelo de diseño)*. Universidad de las Ciencias Informáticas, 2005.
- ESPECIALIDAD, D. *Introducción a la Ingeniería de Software*. Universidad de las Ciencias Informáticas, 2005.

Fuente electrónica:

- FORERO, R. O. S. *Animación*, 2006. [Disponible en: http://www.humanresearchgroup.org/Materiales_y_Recursos/material%20ICD%20%20Animacion.pdf]
- LUNA, F. *Skinned Mesh Character Animation with Direct3D 9.0*, 2004. [Disponible en: http://www.moon-labs.com/resources/d3dx_skinnedmesh.pdf]
- MARZO, J. V. *Introducción a UML*, 2006. [Disponible en: http://www.vico.org/aRecursosPrivats/TRAD_introUML.pdf]
- THORN, A. *DirectX® 9 Graphics: The Definitive Guide to Direct3D*, 2005. [Disponible en: <http://eshop.pc-ware.fi/Data/FileLibrary/fi/books%204x7%20itpro%20catalogue%20jun06.pdf>]
- VALENCIA., D. I. U. D. *Animación 3D*, 2006. [Disponible en: http://informatica.uv.es/iiguia/AIG/web_teoría/tema5.pdf]
- VERDUGO, F. *Herramientas de Animación - A*, 2007. [Disponible en: http://student.vfs.com/~felipev/tutoriales/Manuales_PDF/Leccion_05_anim_A.pdf]
- WIKIPEDIA, C. *Key Frames*, 2007. [Disponible en: http://en.wikipedia.org/wiki/Key_frame]

- WIKIPEDIA, C. *Animation*, 2007. [Disponible en: <http://en.wikipedia.org/wiki/Animation>
- WIKIPEDIA, C. *3D Studio Max*, 2007. [Disponible en: http://es.wikipedia.org/wiki/3D_Studio_Max
- WIKIPEDIA, C. *Interpolación*, 2007. [Disponible en: <http://es.wikipedia.org/wiki/Interpolacion>
- WIKIPEDIA, C. *Maya (Aplicación Gráfica)*, 2006. [Disponible en: http://es.wikipedia.org/wiki/Maya_%28aplicaci%C3%B3n_gr%C3%A1fica%29
- WIKIPEDIA, C. *Morph Target Animation*, 2007. [Disponible en: http://en.wikipedia.org/wiki/Morph_target_animation
- WIKIPEDIA, C. *Skeletal Animation*, 2007. [Disponible en: http://en.wikipedia.org/wiki/Skeletal_animation
- AIRES, D. M. U. D. B. *Introducción a la Interpolación de Spline*, 2006. [Disponible en: www.fi.uba.ar/materias/6119/Trabajos_alumnos/fhernaez/Intro_a_splines.pdf
- CINEBLOG/MX, C. D. *Tipos de Animación*, 2006. [Disponible en: <http://cineblog.motime.com/archive/2006-07>

Tesis:

- ESTRADA, Y. C. *Sistema de Captura de Movimiento para Ambientes Tridimensionales*. Ciudad de La Habana, universidad de las Ciencias Informáticas, 2006. 108. p.

GLOSARIO DE TÉRMINOS

B

1. **Bípodo:** No es más que el animal que utiliza dos extremidades para desplazarse. En la Realidad Virtual es el esqueleto que cumple la característica anterior y que se utiliza en las animaciones.

C

2. **Captura de movimiento:** Técnica que captura los movimientos de una persona a través de un traje especial. Estos movimientos son posteriormente procesados digitalmente. En esta técnica se requiere el movimiento, no la apariencia.

D

3. **Disco Duro:** Se llama disco duro o disco rígido al dispositivo encargado de almacenar información de forma permanente en una computadora.

H

4. **Herencia:** Es uno de los mecanismos de la programación orientada a objetos, por medio del cual una clase se deriva de otra de manera que extiende su funcionalidad. Una de sus funciones más importantes es la de proveer el Polimorfismo (ver concepto de polimorfismo).
5. **Hueso padre:** En una jerarquía de huesos se le llama así a los huesos de los cuáles depende el movimiento de otros, que serían sus hijos, ejemplo (brazo y antebrazo).

J

6. **Jerarquía de huesos:** Conjunto de huesos que forman un esqueleto en el cual estos están organizados jerárquicamente, ya que todos poseen padres (ver hueso padre) o hijos, y en algunos casos ambas cosas.

M

7. **Mesh:** Término en inglés que significa malla que no es más que la forma de representar un modelo 3D a partir de polígonos.
8. **Morphing:** Es la técnica de animación por la cual una imagen es gradualmente convertida en otra.
9. **Memoria:** Se refiere a una forma de almacenamiento de estado sólido conocido como memoria de acceso aleatorio (RAM por sus siglas en inglés). Ver concepto de RAM.

P

10. **Personaje:** Actor de la escena de un mundo de realidad virtual, que soporta acciones (como un tipo de comportamiento), y que tienen entre sus atributos, cualidades físicas y emocionales que serán usadas a la hora de ejecutar las acciones.
11. **Polimórfico:** Se define un objeto como polimórfico a la entidad que puede contener valores de diferentes tipos durante la ejecución del programa.
12. **Postura:** Posición que adopta un bípedo en un determinado momento de una animación.

R

13. **RAM:** Se trata de una memoria de semiconductor en la que se puede tanto leer como escribir información

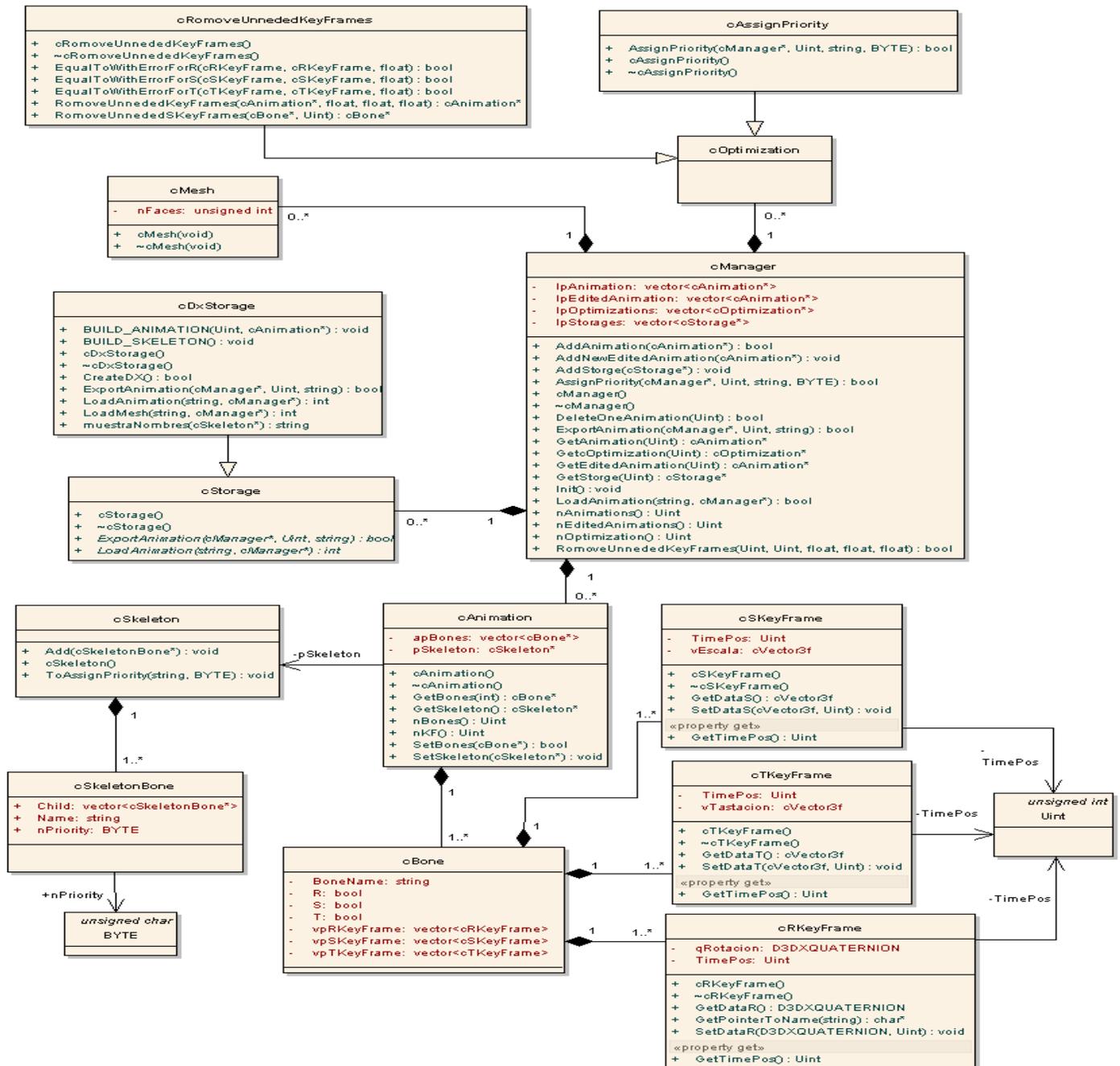
14. **Realidad virtual:** La Realidad Virtual es un sistema o interfaz informática que genera entornos sintéticos en tiempo real, representación de las cosas a través de medios electrónicos o representaciones de la realidad, una realidad ilusoria.

S

15. **Skinned:** Deformado de piel.

ANEXOS

Anexo 1: Diagrama de clases del diseño.

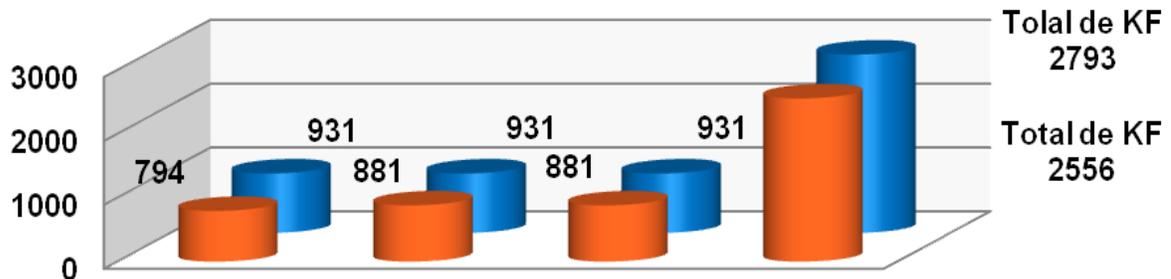


Anexo 2: Análisis de Pruebas de Reducir Key Frames.

Tabla de resultados obtenidos para las pruebas realizadas con valores de error permisibles de: KFR = 0.001, KFT = 0, KFS = 0 a cada animación:

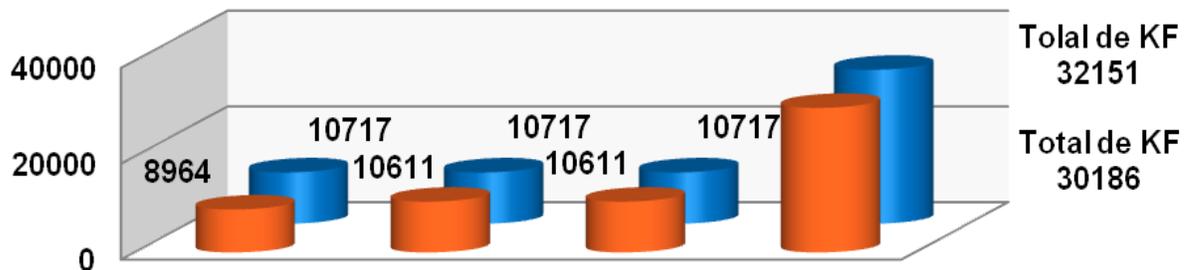
Nombre de la animación	Inicial Tauren.X			
Datos Iniciales				
Animaciones	KFR	KFT	KFS	Totales
Cantidad de KF	931	931	931	2793
Cantidad e KB	18.184	14.547	14.547	47.278
Datos Finales				
Animaciones	KFR	KFT	KFS	Totales
Cantidad de KF	794	881	881	2556
Cantidad de KB	15.508	13.57	13.57	42648
Análisis				
Cantidad de KF eliminados	137	50	50	237
Reducción en KB	2.676	0.977	0.977	4.629
Porcentaje de KF reducidos	14.715	5.371	5.371	8.486

Comparación de las animaciones iniciales con animaciones editadas en cuanto a KF de Rotación, Tralación y Escala.



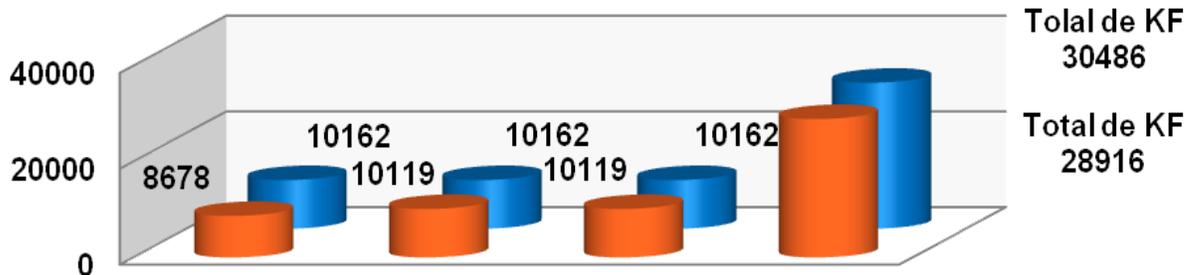
Nombre de la animación	<i>Inicial</i> <i>MujerCaminando.X</i>			
Datos Iniciales				
Animaciones	KFR	KFT	KFS	Totales
Cantidad de KF	10717	10717	10717	32151
Cantidad e KB	209.316	167.453	167.453	544.222
Datos Finales				
Animaciones	KFR	KFT	KFS	Totales
Cantidad de KF	8964	10611	10611	30186
Cantidad de KB	175.078	165.383	165.383	505.844
Análisis				
Cantidad de KF eliminados	1753	106	106	1965
Reducción en KB	34.238	2.070	2.070	38.379
Por ciento de KF reducidos	16.357	0.989	0.989	6.112

Comparación de las animaciones iniciales con animaciones editadas en cuanto a KF de Rotación, Tralación y Escala.



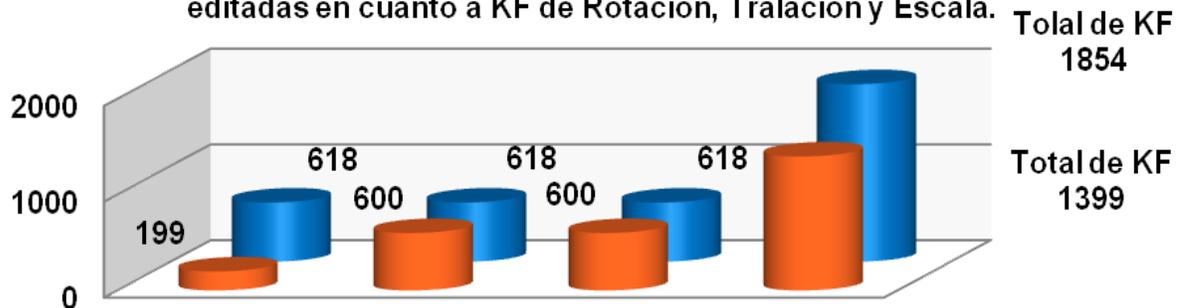
Nombre de la animación	<i>Inicial EvaMeneo.X</i>			
Datos Iniciales				
Animaciones	KFR	KFT	KFS	Totales
Cantidad de KF	10162	10162	10162	30486
Cantidad e KB	198.477	158.781	158.781	516.039
Datos Finales				
Animaciones	KFR	KFT	KFS	Totales
Cantidad de KF	8678	10119	10119	28916
Cantidad de KB	169.493	157.941	157.941	485.375
Análisis				
Cantidad de KF eliminados	1484	43	43	1570
Reducción en KB	28.984	0.840	0.840	30.664
Por ciento de KF reducidos	14.603	0.423	0.423	5.150

Comparación de las animaciones iniciales con animaciones editadas en cuanto a KF de Rotación, Tralación y Escala.



Nombre de la animación	<i>Inicial Diablito.X</i>			
Datos Iniciales				
Animaciones	KFR	KFT	KFS	Totales
Cantidad de KF	618	618	618	1854
Cantidad e KB	12.070	9.657	9.657	31.384
Datos Finales				
Animaciones	KFR	KFT	KFS	Totales
Cantidad de KF	199	600	600	1399
Cantidad de KB	3.886	9.305	9.305	22.496
Análisis				
Cantidad de KF eliminados	419	18	18	455
Reducción en KB	8.184	0.352	0.352	8.887
Por ciento de KF reducidos	67.799	2.913	2.913	24.542

Comparación de las animaciones iniciales con animaciones editadas en cuanto a KF de Rotación, Tralación y Escala.



Nombre de la animación	<i>Inicial Personaje.X</i>			
Datos Iniciales				
Animaciones	KFR	KFT	KFS	Totales
Cantidad de KF	1065	1065	1065	3195
Cantidad e KB	20.800	16.640	16.640	54.080
Datos Finales				
Animaciones	KFR	KFT	KFS	Totales
Cantidad de KF	926	1007	1007	2940
Cantidad de KB	18.085	15.507	15.507	49.099
Análisis				
Cantidad de KF eliminados	139	58	58	255
Reducción en KB	2.715	1.133	1.133	4.980
Por ciento de KF reducidos	13.052	5.446	5.446	7.981

Comparación de las animaciones iniciales con animaciones editadas en cuanto a KF de Rotación, Tralación y Escala.

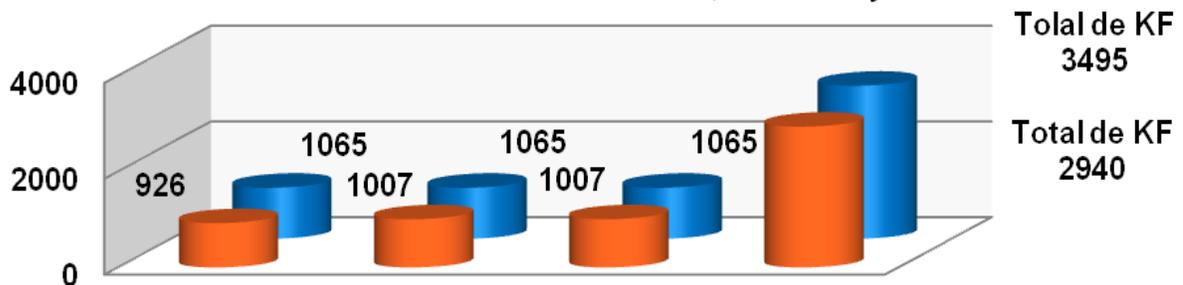
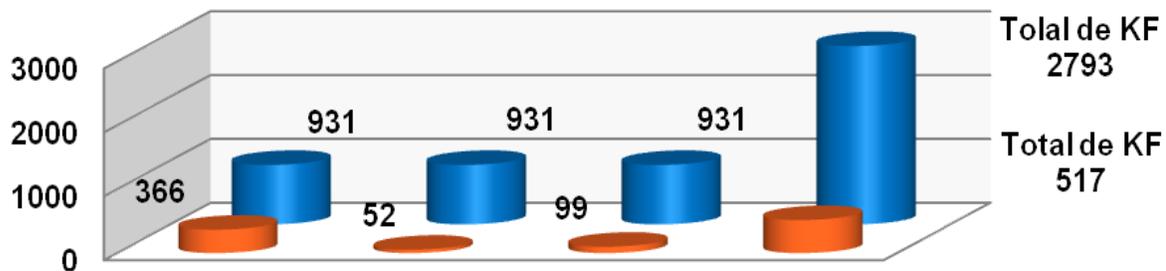


Tabla de resultados obtenidos para las pruebas realizadas con valores de error permisibles de: KFR = 2.000, KFT = 5, KFS = 5 a cada animación:

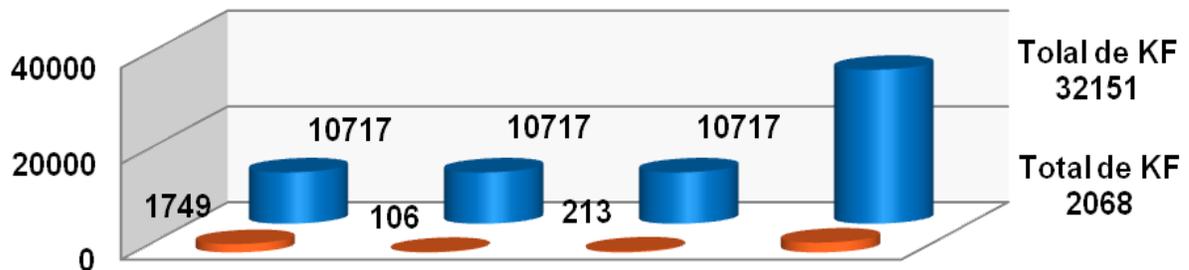
Nombre de la animación	<i>Inicial Tauren.X</i>			
Datos Iniciales				
Animaciones	KFR	KFT	KFS	Totales
Cantidad de KF	931	931	931	2793
Cantidad e KB	18.184	14.547	14.547	47.278
Datos Finales				
Animaciones	KFR	KFT	KFS	Totales
Cantidad de KF	366	52	99	517
Cantidad de KB	7.149	0.813	1.547	9.509
Análisis				
Cantidad de KF eliminados	565	879	832	2276
Reducción en KB	11.035	13.734	13.000	37.769
Por ciento de KF reducidos	60.687	94.415	89.366	81.489

Comparación de las animaciones iniciales con animaciones editadas en cuanto a KF de Rotación, Tralación y Escala.



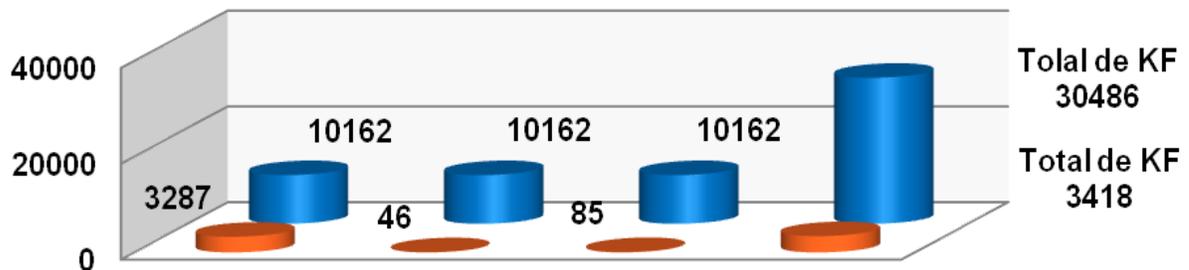
Nombre de la animación	<i>MujerCaminando.X</i>			
Datos Iniciales				
Animaciones	KFR	KFT	KFS	Totales
Cantidad de KF	10717	10717	10717	32151
Cantidad e KB	209.316	167.453	167.453	544.222
Datos Finales				
Animaciones	KFR	KFT	KFS	Totales
Cantidad de KF	1749	106	213	2068
Cantidad de KB	34.16	1.656	3.328	39.144
Análisis				
Cantidad de KF eliminados	8968	10611	10504	30083
Reducción en KB	175.156	165.797	164.125	587.558
Por ciento de KF reducidos	83.680	99.011	98.013	93.568

Comparación de las animaciones iniciales con animaciones editadas en cuanto a KF de Rotación, Tralación y Escala.



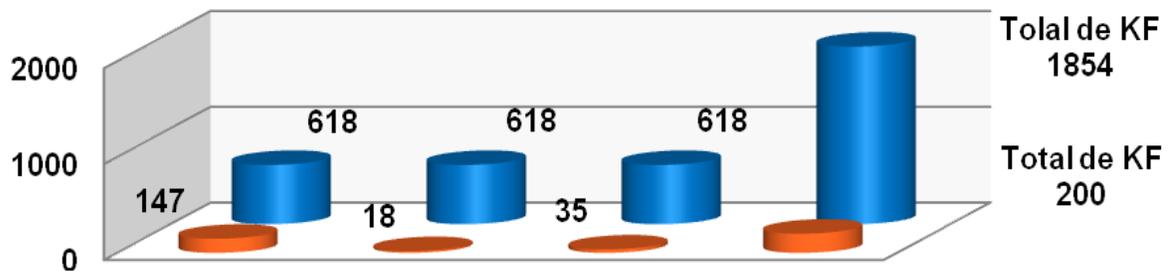
Nombre de la animación	<i>Inicial EvaMeneo.X</i>			
Datos Iniciales				
Animaciones	KFR	KFT	KFS	Totales
Cantidad de KF	10162	10162	10162	30486
Cantidad e KB	198.477	158.781	158.781	516.039
Datos Finales				
Animaciones	KFR	KFT	KFS	Totales
Cantidad de KF	3287	46	85	3418
Cantidad de KB	64.199	0.718	1.328	66.245
Análisis				
Cantidad de KF eliminados	6875	10116	10077	27068
Reducción en KB	134.278	158.063	157.453	528.672
Por ciento de KF reducidos	67.654	99.547	99.164	88.788

Comparación de las animaciones iniciales con animaciones editadas en cuanto a KF de Rotación, Tralación y Escala.



Nombre de la animación	<i>Inicial Diablito.X</i>			
Datos Iniciales				
Animaciones	KFR	KFT	KFS	Totales
Cantidad de KF	618	618	618	1854
Cantidad e KB	12.070	9.657	9.657	31.384
Datos Finales				
Animaciones	KFR	KFT	KFS	Totales
Cantidad de KF	147	18	35	200
Cantidad de KB	2.871	0.282	0.548	3.701
Análisis				
Cantidad de KF eliminados	471	600	583	1654
Reducción en KB	9.199	9.375	9.109	27.683
Porcentaje de KF reducidos	76.214	97.087	94.337	89.213

Comparación de las animaciones iniciales con animaciones editadas en cuanto a KF de Rotación, Tralación y Escala.



Nombre de la animación	<i>Inicial Personaje.X</i>			
Datos Iniciales				
Animaciones	KFR	KFT	KFS	Totales
Cantidad de KF	1065	1065	1065	3195
Cantidad e KB	20.800	16.640	16.640	54.080
Datos Finales				
Animaciones	KFR	KFT	KFS	Totales
Cantidad de KF	382	59	114	555
Cantidad de KB	7.46	0.921	1.781	10.162
Análisis				
Cantidad de KF eliminados	683	1006	951	2640
Reducción en KB	13.340	15.719	14.859	43.918
Por ciento de KF reducidos	64.131	94.460	89.296	82.629

Comparación de las animaciones iniciales con animaciones editadas en cuanto a KF de Rotación, Tralación y Escala.

