



Universidad de las Ciencias Informáticas
FACULTAD 5

**ACTUALIZACION Y AJUSTE DEL MODIFICADOR DE
ANIMACIONES DE HUESOS PARA JUEGOS CNEURO.**

**Trabajo de Diploma para optar por el título de
Ingeniero en ciencias Informáticas**

Autores:

Annia Valdés Díaz.
Yaser Jiménez Rivero.

Tutor:

Ing. Minardo Gollún González López.

**Ciudad de la Habana
Julio del 2008**

“Lo que sabemos es una gota de agua; lo que ignoramos es el océano”

Isaac Newton.

DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

Yaser Jiménez Rivero.

Autor

Annia Valdés Díaz.

Autor

Ing. Minardo Gollún González López.

Tutor.

Quisiera agradecer a la revolución y a nuestro siempre comandante en jefe Fidel, por haberme dado la oportunidad de estudiar y graduarme en una universidad del tipo de la UCI. Agradecer a todos los profesores que contribuyeron a mi formación en estos 5 años de estudios. Agradecer a mis padres que me dieron el don de la vida, además de ser piezas claves en mi formación, a mi hermana, la que admiro por encima de todo, por su espíritu luchador, a los tres, por siempre haber confiado en mí y haberme apoyado en todo momento. A mi tía Fela, que desde el momento que vine para la Habana me atendió y acogió como a un hijo más. A mi tío Lillo, quien es para mí como un amigo, un hermano, un padre más. A mi novia Aylenis, por todos los momentos lindos que pase junto a ella, y por todo el apoyo que me brindo en momentos que pensé que no podía más, agradecer a ella y a su familia que también estuvo pendiente del estado y realización de mi tesis. A mi prima Mayren por también, siempre estar pendiente del desarrollo de mi tesis. En general Agradecimientos a toda mi familia, a mi abuela, a Yanet, a Nana, Martica, Coco, Maidita, Domingo, a todos.

Quisiera agradecer también a mis amigos y amigas de estos 5 años de estudio, y en especial, a toda la gente de mi grupo, por haberme dado la oportunidad de una forma u otra de aprender algo de ellos, de ahí quisiera agradecer a Adrian Carlos, quien siempre nos ha dado consejos de cómo ser mejores a todos en el grupo, . . . y por estar siempre ahí, cuando se necesitaba de él. A dos socios que son como si fueran hermanos míos, uno Ramón, y el otro Rigoberto, quienes tienen toda mi admiración y respeto.

Agradezco a mi tutor Minardo, por las ayudas y las enseñanzas que me dio.

A todos muchas gracias.

Yaser Jiménez Rivero.

Quiero agradecer a la Revolución por esta maravillosa universidad; a mis padres, a mi hermana que ha soportado tenerme de turista en la casa cada vez que iba. A mi familia en general; tías, primos a mis abuelos (Gregorio y Emiliana) que siempre me estuvieron apoyando a lo largo de estos cinco años de mi carrera. A mi padrastro (Ismael), a mis vecinos que han sido de gran ayuda y se han comportado como una familia más, en todo momento y en lo que he necesitado. También a mis amistades (Aracelys y Yainieris).

Agradecer además a todos mis compañeros de grupo, que cuando los he necesitado han estado a mi lado, a mis amistades de la universidad y a todos aquellos profesores que contribuyeron con mi formación durante estos cinco años.

A mi tutor Minardo por la ayuda brindada en el desarrollo de este trabajo de diploma, gracias.

Annia Valdés Díaz.

Mi trabajo se lo dedico a toda mi familia, en especial a mis padres, mi hermana, mis tíos Felicia (Fela) y Angel (Lillo), a mis primas Mayren, Yanet y Arianna(Nana), primitos Luisito y Rolí, a mi abuela, y no por ser la ultima es menos especial a mi querida novia Aylenis.

Yaser Jiménez Rivero.

Especialmente a mi madre, que siempre, siempre ha estado ahí, apoyándome en todo, en los buenos y malos momentos de mi carrera. A mi papá que nunca perdió las esperanzas en mi, y me decía que siempre bateara y a mi hermana.

Annia Valdés Díaz.

RESUMEN

El proyecto “JUEGOS CNEURO” de la facultad 5 desarrolla aplicaciones que utilizan animaciones 3D. Debido a que los juegos que se desarrollan en el proyecto, serán instalados en escuelas y centros de estudios donde las computadoras existentes, no poseen los requisitos de hardware más óptimos, se hace necesario aprovechar al máximo los recursos. Uno de los principales problemas, es el mal uso de memoria, debido a la cantidad excesiva de Frames que contienen los ficheros de las animaciones generadas por los distintos editores.

Anteriormente se había hecho una aplicación prototipo para la solución de este problema, pero no se pudo usar para la realización de estos proyectos por no ser compatible con la herramienta de desarrollo con que cuenta la facultad. Con este trabajo se brinda una solución para este problema mediante la actualización de este prototipo, se implementó una aplicación que permite cargar las animaciones y aplicarle ciertas modificaciones exportándolas con el formato adecuado para ser utilizada por la herramienta de desarrollo. El objetivo principal es reducir los Key Frames innecesarios.

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.

Palabras claves:

Animación, Key Frames e interpolación.

INDICE

AGRADECIMIENTOS: ¡Error! Marcador no definido.

DEDICATORIA: ¡Error! Marcador no definido.

RESUMEN **IV**

INTRODUCCION **1**

capítulo 1: FUNDAMENTACION TEORICA **7**

1.1 Animación **8**

1.2 Fotograma (Frame): **8**

Fig. 1 Secuencia de Frames en una animación **8**

1.3 Fotograma clave (Key Frame): **9**

 1.4.1 Interpolación de Spline: 10

1.5 Técnicas para la Animación: **11**

 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 5: 13

 1.6.2 Maya: 14

 1.6.3 Blender: 14

1.7 Proceso actual de realización de animaciones: **15**

1.8 Metodologías y herramientas de desarrollo: **16**

 1.8.1 Metodología: 16

 1.8.2 Herramientas: 17

 1.8.2.2 Microsoft Visual Studio 2005: 18

 1.8.2.3 3D Studio Max 5 19

 1.8.2.4 Scene Toolkit 19

1.9 Lenguajes:	19
1.9.1 Lenguaje de programación:	20
1.9.2 Lenguaje de Modelado:	20
capítulo 2: SOLUCION PROPUESTA	22
2.1: Interpolación	23
2.1.1: Interpolación Bilineal.....	23
2.1.2: Interpolación Bicúbica.....	24
2.1.3: Interpolación Spline.	25
2.2: Tabla de comparación de posibles algoritmos a usar en la implementación de reducción de KeyFrame en una animación.	27
2.3: Estructura del fichero con extensión anx, desarrollado en la facultad.	28
2.4: Consideraciones del capítulo.	31
capítulo 3: FUNCIONAMIENTO DEL SISTEMA	32
3.1: Definición de los Requisitos Funcionales del sistema.	33
3.2: Definición de los Requisitos No Funcionales del sistema.	33
3.3: Definición de los Casos de Uso.	34
3.3.1: Definición de los actores del sistema.	35
3.3.2 Diagrama de Casos de Uso por paquetes:	37
3.3.3: Descripción textual de los Casos de Uso.	38
3.4: Modelo Conceptual	41
3.5 Arquitectura del sistema.	43
3.6: Funcionamiento del Framework.	44
3.6.1: Petición de inicializar.	45
3.6.2: Petición de Cargar Animación.	46
3.6.3: Petición de Aplicar Modificación.	47
3.6.4: Petición de Exportar Animación.....	47
3.6.4: Petición de Visualizar Animación.....	48

3.7: Manual de la herramienta para el usuario.....	48
3.8 Ayuda de la herramienta:	58
Capítulo 4: DISEÑO DEL SISTEMA.....	62
4.1 Diagrama de clases del diseño:.....	63
4.1.1 Descripción de las clases:	63
4.2: Diagrama de clases del diseño por paquetes.....	66
4.3: Descripción de la implementación por Casos de Uso.....	69
4.3.1: Caso de Uso Cargar Animación.	70
4.3.3: Caso de Uso Exportar Animación.....	71
capítulo 5: RESULTADOS	73
5.1: Análisis de reducción de KeyFrame en diferentes animaciones.....	74
5.2: Análisis de lectura del fichero desde la matriz que define el movimiento.....	76
2.3: Consideraciones del capítulo.	76
CONCLUSIONES.....	77
RECOMENDACIONES.....	78
BIBLIOGRAFIA	79
ANEXOS.....	81
GLOSARIO DE TERMINOS.....	82

INDICE DE FIGURAS

<i>Fig. 1 Secuencia de Frames en una animación.....</i>	<i>8</i>
<i>Fig. 2: Secuencia de Frames en una animación.....</i>	<i>9</i>
<i>Fig. 3: Ejemplo gráfico de Interpolación.....</i>	<i>10</i>
<i>Fig. 4: Área de aplicación de la Morphing Animations.....</i>	<i>12</i>
<i>Fig. 5: Ejemplo de una animación usando Skeletal Animations.....</i>	<i>13</i>
<i>Fig. 6: Proceso actual de realización y utilización de las animaciones.....</i>	<i>16</i>
<i>Fig. 7: Ejemplo de interpolación bilineal.....</i>	<i>24</i>
<i>Fig. 8: Ejemplo de interpolación bicúbica.....</i>	<i>25</i>
<i>Fig. 9: Ejemplo de Fenómeno de Runge.....</i>	<i>26</i>
<i>Fig. 10: Ejemplo de Interpolación Spline de grado 1 respecto a Xi.....</i>	<i>27</i>
<i>Fig. 11: Paquete Importar Datos.....</i>	<i>37</i>
<i>Fig. 12: Paquete Aplicar Optimización.....</i>	<i>38</i>
<i>Fig. 13: Paquete Exportar Datos.....</i>	<i>38</i>
<i>Fig. 14: Paquete Visualización.....</i>	<i>38</i>
<i>Fig. 15: Modelo Conceptual.....</i>	<i>42</i>
<i>Fig. 16: Diagrama de Clases del Análisis.....</i>	<i>43</i>
<i>Fig. 17: Diagrama de clase del Análisis.....</i>	<i>44</i>
<i>Fig. 18: Diagrama de secuencia para la petición de Inicial.....</i>	<i>45</i>
<i>Fig. 19: Diagrama de secuencia para la petición de Cargar Animación.....</i>	<i>46</i>
<i>Fig. 20: Diagrama de secuencia para la petición de Aplicar Modificación.....</i>	<i>47</i>
<i>Fig. 21: Diagrama de secuencia para la petición de Exportar Animación.....</i>	<i>47</i>
<i>Fig. 22: Diagrama de secuencia para la petición de Visualizar Animación.....</i>	<i>48</i>
<i>Fig. 23: Interfaz principal de la herramienta MODANS.....</i>	<i>49</i>
<i>Fig. 24: Menú archivos de la herramienta MODANS.....</i>	<i>50</i>
<i>Fig. 25: Funcionalidad Cargar Animación con formato inadecuado.....</i>	<i>51</i>
<i>Fig. 26: Funcionalidad Cargar Animación con formato .X.....</i>	<i>52</i>
<i>Fig. 27: Funcionalidad Cargar Animación con formato .ANX.....</i>	<i>52</i>
<i>Fig. 28: Funcionalidad Asignar Prioridad.....</i>	<i>53</i>

<i>Fig. 29: Funcionalidad Reducir Key Frame para formato .X.</i>	<i>54</i>
<i>Fig. 30: Funcionalidad Reducir Key Frame para formato ANX.</i>	<i>55</i>
<i>Fig. 31: Funcionalidad Reducir Key Frame para formato ANX.</i>	<i>56</i>
<i>Fig. 32: Funcionalidad Exportar animación formato X.</i>	<i>57</i>
<i>Fig. 33: Funcionalidad Exportar animación formato ANX.</i>	<i>57</i>
<i>Fig. 34: Diagrama de clases del diseño por paquetes.</i>	<i>67</i>
<i>Fig. 35: Paquete Manager</i>	<i>68</i>
<i>Fig. 36: Paquete Modifications.....</i>	<i>68</i>
<i>Fig. 37: Paquete Storage.....</i>	<i>69</i>
<i>Fig. 38: Diagrama de secuencia Cargar Animación.....</i>	<i>70</i>
<i>Fig. 39: Diagrama de secuencia Reducir KeyFrame.</i>	<i>71</i>
<i>Fig. 40: Diagrama de secuencia Exportar Animación.</i>	<i>72</i>
<i>Fig. 41: Grafica comparativa de las 3 animaciones en cuanto al peso en KB.</i>	<i>75</i>
<i>Fig. 42: Tabla de comparación en cuanto a lecturas del fichero para la construcción de las matrices que definen el movimiento de cada hueso de la animación.</i>	<i>76</i>
<i>Fig. 43: Anexo1. Comparación de los métodos de Interpolación.</i>	<i>81</i>
<i>Fig. 44: Anexo2. Tabla de comparación en cuanto al peso del fichero en KB.</i>	<i>81</i>

INTRODUCCIÓN

Desde sus inicios, el hombre ha querido “ilustrar” sus recuerdos mediante representaciones gráficas. Antes de escribir, el hombre dibujó para representar los objetos y los hechos más relevantes que lo rodeaban. La escritura, le permitió transformar las imágenes mentales y visuales en palabras, mas nunca logró reemplazar 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 la aparición de la computación, el hombre ha aprovechado sus grandes ventajas para la representación y simulación de hechos de la vida real, lo que hace que en el mundo de hoy resulte indispensable el conocimiento de las tecnologías de animación, claramente visibles cada vez con más fuerzas en todo tipo de productos que se presentan en todas las esferas de nuestra sociedad.

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.

Actualmente se cuenta con herramientas, como el 3Dstudio Max, Maya y Blender para la realización de animaciones 3D. Estas herramientas a partir de un bípodo y utilizando técnicas como la animación esquelética, permiten obtener la animación deseada por el artista. El problema radica a la hora de exportar dicha animación, pues, junto con los estados o Frames esenciales que el artista definió, la herramienta guarda otros Frames, aumentando considerablemente el volumen del fichero, aunque, con disimiles algoritmos se puede reducir considerablemente la cantidad de estos Frames en el fichero resultante.

El aumento de frames en los ficheros que contienen las animaciones, es directamente proporcional al volumen de recursos de hardware requeridos para procesarlos, provocando un uso excesivo de la memoria.

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.

La herramienta MODANS fue concebida con el propósito de reducir la cantidad de datos que son almacenados a la hora de la realización de una animación en 3D. Exportando, los frames definidos con anterioridad por el artista de la animación, sin los otros que son generados internamente por la aplicación, logrando reducir el espacio en el fichero.

En la actualidad, en la Universidad de las Ciencias Informáticas, le fue asignada la realización del proyecto Juegos CNEURO a la facultad 5, para este se necesitan animaciones que consten de la menor cantidad de volumen posible en sus ficheros, pues estos juegos serán puesto a prueba en las escuelas de nuestro país, y estos centros, no cuentan con las computadoras con el hardware necesario para ejecutar programas que necesiten de animaciones excesivamente cargadas.

La herramienta MODANS fue implementada con la extensión Direct.x; lo cual solo exporta formatos punto x. En el proyecto “Juegos CNEURO”, se necesita hacer uso de la herramienta, pero para esto debe exportar otro el anx, además de cumplir otras funcionalidades. Esto permitirá que el nuevo archivo, pueda ser utilizado por la herramienta de desarrollo con que cuenta la facultad para la realización de estos Juegos.

Para que la herramienta MODANS pueda ser utilizada en el proyecto “Juegos CNEURO”, se decidió realizar, una actualización y ajuste de sus funcionalidades.

Problema Científico:

¿Cómo lograr disminuir el volumen de datos necesarios, para representar las animaciones usadas en el proyecto “Juegos CNEURO”, haciendo uso de la herramienta MODANS?

Objeto de Estudio:

Animaciones esqueléticas en modelos virtuales.

Campo de Acción:

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

Objetivo General:

Implementar a la herramienta MODANS de la Facultad 5, las funcionalidades necesarias para resolver la necesidad de disminución de datos, para la representación de las animaciones por huesos que se presenta en el proyecto Juegos CNEURO.

Tareas a Desarrollar:

- Realizar un estudio profundo de la bibliografía acorde con el tema.
- Incorporar a la herramienta un exportador que permita el manejo del formato ANX.
- Elaborar tablas de comparación entre los posibles algoritmos a utilizar para la realización de la reducción de los Key Frames de una animación.
- Implementar el algoritmo de reducción de Key Frames seleccionado según los resultados de los estudios realizados en el trabajo.
- Implementar un visualizador de animaciones para el formato ANX y de un análisis de los resultados no visuales, obtenidos en la reducción de Key Frames.
- Elaborar el manual del usuario e incorporar a la Herramienta la ayuda requerida.

Métodos de investigación:

Los métodos teóricos son de gran importancia en el desarrollo del trabajo de diploma porque mediante ellos se obtiene una medida de cómo desarrollar la herramienta; a través del análisis, la evaluación de la información y la sistematización permitieron evaluar el proceso mientras este se iba desarrollando llegando a conclusiones favorables para resolver el problema.

Los métodos teóricos utilizados fueron:

- **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.
- **Inducción-deducción** se utilizó durante toda la investigación, para llegar a conclusiones y hacer generalizaciones.
- **Histórico-lógico** se utilizó al estructurar la trayectoria del objeto en el transcurso de su historia y en orden cronológico.
- **Modelación Analógica** se utilizó para la elaboración tanto de proceso actual, como propuestas de solución.

Los métodos empíricos que se emplearon durante el transcurso de la investigación fueron:

- **Observación** en el resultado de la modificación de las animaciones respecto al tamaño de sus ficheros, al ser estas exportadas y el mantenimiento de la calidad de las mismas luego del proceso.
- **Encuesta** a profesores y estudiantes del Proyecto CNEURO con el objetivo de obtener los beneficios que podrían traer la reducción del volumen de información.

Los métodos estadísticos que posibilitaron el procesamiento de toda la información fueron:

- **Estadística descriptiva** apoyados en el uso de tablas para establecer comparaciones entre los datos iniciales y los finales.

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 exportándolos con la extensión .anx, los artistas digitales y desarrolladores del proyecto “Juegos CNEURO” obtendrán animaciones que consuman solo los recursos necesarios, con la información que realmente necesitan y liberando memoria para otros procesos.

Organización del documento:

El capítulo 1:

En este capítulo se presenta una introducción sobre lo que es la animación 3D; mostrándose los conceptos básicos que se manejan en la misma relacionados con este tópico, incluyendo las técnicas, herramientas y proceso actual para su realización. Además de la justificación de las herramientas y lenguajes a ser utilizados para la realización de la tesis.

El capítulo 2:

En este capítulo se muestra la solución propuesta para el problema que se plantea anteriormente; así como el establecimiento de una comparación entre las distintas variantes del método matemático Interpolación.

El capítulo 3:

En este capítulo 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 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:

En este capítulo se 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:

En este capítulo se 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.

CAPÍTULO 1: 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 bípedos² en la animación 3D, en cada fotograma clave (key Frame) tenemos una postura¹² del personaje¹¹, 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.

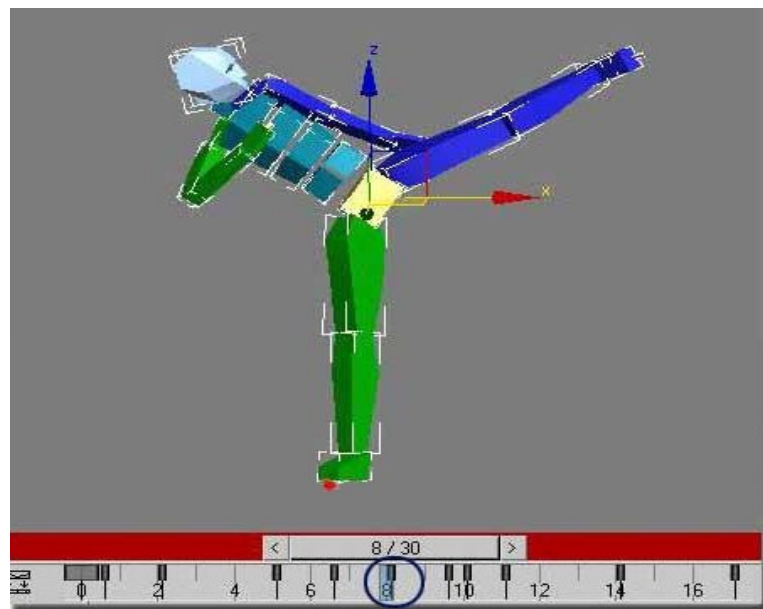


Fig. 2: Secuencia de Frames en una animación.

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, es 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.(8)

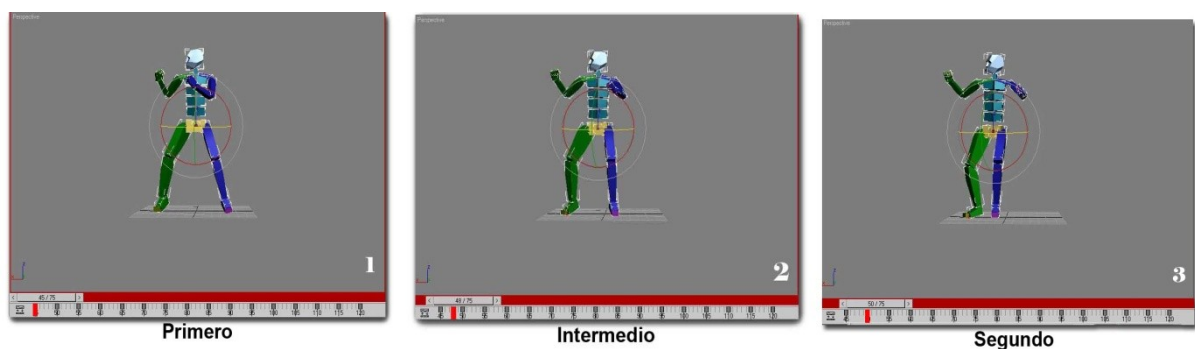


Fig. 3: Ejemplo gráfico de Interpolación.

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 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 morphing⁹ 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 (ver Fig. 4), 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.

En la figura anterior se muestra un ejemplo de cómo utilizar la técnica morphing animation, para obtener diferentes expresiones faciales.

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 huesos⁵ 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 movimiento³, con la cual se obtiene más rapidez y realismo en las animaciones.

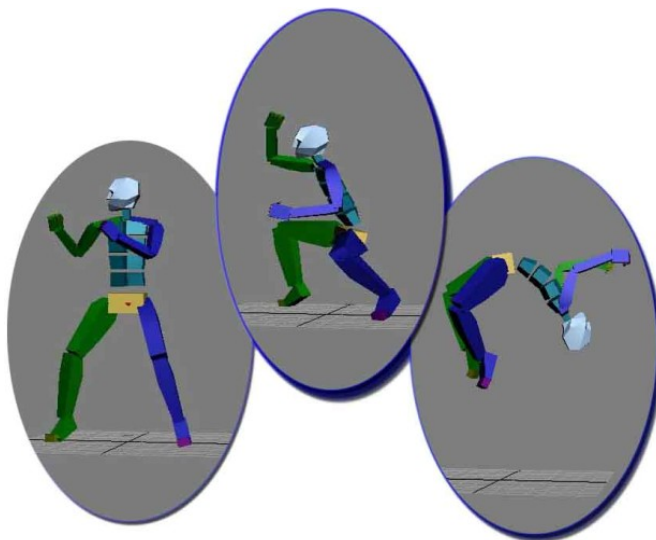


Fig. 5: Ejemplo de una animación usando Skeletal Animations.

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, Maya y Blender.

1.6.1 3D Studio Max 5:

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.

1.6.3 Blender:

Blender es un programa multiplataforma¹⁴, dedicado especialmente al modelado y creación de gráficos tridimensionales. El programa fue inicialmente distribuido de forma gratuita pero sin el código fuente, con un manual disponible para la venta. Actualmente es compatible con todas las versiones de Windows, Mac OS X, Linux, Solaris, FreeBSD e IRIX.

Tiene una muy peculiar interfaz gráfica de usuario, que se critica como poco intuitiva, pues no se basa en el sistema clásico de ventanas; pero tiene a su vez ventajas importantes sobre éstas, como la configuración personalizada de la distribución de los menús y vistas de cámara.

Aún siendo una herramienta relativamente nueva, ha gozado de la aceptación de muchos animadores independientes. En la industria de Generación de gráficos avanza como un proyecto prometedor.

Las características más importantes con las que cuenta la herramienta son:

- Multiplataforma, libre, gratuito y con un tamaño de origen realmente pequeño comparado con otros paquetes de 3D, dependiendo del sistema operativo en el que se ejecuta.
- Capacidad para una gran variedad de primitivas geométricas¹³, incluyendo curvas, mallas poligonales⁶, vacíos, NURBS¹⁰, metaballs⁸.
- Junto a las herramientas de animación se incluyen cinemática inversa⁴, deformaciones por armadura o cuadrícula, vértices de carga y partículas estáticas y dinámicas.
- Edición de audio y sincronización de video.
- Características interactivas para juegos como detección de colisiones, recreaciones dinámicas y lógica.
- Blender acepta formatos gráficos como TGA, JPG, Iris, SGI, o TIFF. También puede leer ficheros Inventor.

1.7 Proceso actual de realización de animaciones:

En la actualidad se utilizan herramientas como el 3D Studio Max, Maya, Blender 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 KeyFrame 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 Rational Rose:

Esta herramienta propone la utilización de cuatro tipos de modelo para realizar un diseño del sistema, utilizando una vista estática y otra dinámica de los modelos del sistema, uno lógico y otro físico. Permite crear y refinar estas vistas creando de esta forma un modelo completo que representa el dominio del problema y el sistema de software.

- Desarrollo iterativo.
Rational Rose utiliza un proceso de desarrollo iterativo controlado (controlled iterative process development), donde el desarrollo se lleva a cabo en una secuencia de

iteraciones. Cada iteración comienza con una primera aproximación del análisis, diseño e implementación para identificar los riesgos del diseño, los cuales se utilizan para conducir la iteración, primero se identifican los riesgos y después se prueba la aplicación para que éstos se hagan mínimos. Se realizan pruebas a la aplicación y las modificaciones se añaden al modelo de análisis y diseño y luego se itera nuevamente.

➤ Trabajo en Grupo.

Rose permite que haya varias personas trabajando a la vez en el proceso iterativo controlado, para ello posibilita que cada desarrollador opere en un espacio de trabajo privado que contiene el modelo completo y tenga un control exclusivo sobre la propagación de los cambios en ese espacio de trabajo.

➤ Generador de Código.

Se puede generar código en distintos lenguajes de programación a partir de un diseño en UML.

➤ Ingeniería Inversa.

Rational Rose proporciona mecanismos para realizar la denominada Ingeniería Inversa, es decir, a partir del código de un programa, se puede obtener información sobre su diseño.(6)

1.8.2.2 Microsoft Visual Studio 2005:

Se vincula con el Rational Rose 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 el formato ANX; 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 5

Es una de las más recientes versiones de la solución completa de Discreet para modelado, animación y renderizado¹⁵ 3D avanzados. En todo el mundo los principales estudios de desarrollo de juegos dependen de 3D Studio Max 5 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.

1.8.2.4 Scene Toolkit

Scene ToolKit, es una de las herramientas brindadas por el “Proyecto de Herramientas de Desarrollo para Sistemas de Realidad Virtual” de la Facultad 5. Surge como una respuesta a la necesidad de crear un grupo de herramientas de apoyo a los programadores de aplicaciones finales, acorde con la tendencia mundial de desarrollo de los llamados “engines” o motores gráficos, teniendo como objetivo básico agrupar las funcionalidades comunes a cualquier sistema de realidad virtual, de manera que se les facilite el trabajo a los programadores de juegos y simuladores a través de la reutilización de código.

Esta herramienta, en desarrollo actualmente, permite no solamente la visualización de los entornos sintéticos sino además la aplicación de leyes físicas y matemáticas, animaciones, etc., usando las librerías gráficas OpenGL y DirectX, sobre plataforma Windows y Linux. A la vez, se retroalimenta con las necesidades de los programadores que las utilicen, así como de sus investigaciones.

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:

El C++ se escogió debido a las características que posee por encima de otros lenguajes. Las más importantes son: programación orientada a objetos, portabilidad, brevedad, programación modular, velocidad, potencia para la programación gráfica y ser multiplataforma.

Además, es un lenguaje de programación estandarizado (ISO/IEC 14882:1998), ampliamente difundido, contiene una biblioteca estándar que lo convierte en un lenguaje universal, de propósito general, y ampliamente utilizado tanto en el ámbito profesional como en el educativo.

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.

CAPÍTULO 2: SOLUCIÓN PROPUESTA

En este capítulo se realizó un estudio sobre el método matemático Interpolación que es el algoritmo matemático que utilizan muchas herramientas de animaciones para obtener los Frames o cuadros intermedios entre los Key Frames que son definidos por el artista; se estableció una comparación entre los distintos métodos que existen encontrando la mejor propuesta para dar solución a la reducción de datos almacenados por cada animación diseñada por el artista. Además se muestra el formato `anx`, con el cual trabaja la herramienta Scene Toolkit (STK) y las nuevas modificaciones que se le realizaron para darle solución al problema del proyecto “Juegos CNEURO”.

2.1: Interpolación.

La interpolación en la creación de animaciones forma una parte importante, pues crea un movimiento intermedio de una animación a otra, lo cual es ventajoso, pues el artista no se tiene que detener a realizar esta tarea; con la interpolación, la densidad de la imagen se preservará introduciendo el número de puntos que se requieran en el espacio abierto, dando así a la animación resultante una mejor calidad. De forma más precisa, la interpolación es un método de cálculo, más o menos complejo (hay varios tipos), que añade información a una imagen digital de forma artificial mediante cálculo matemático. Consiste más o menos, en añadir vértices en una zona calculando la media de los vértices contiguos.

Como decíamos anteriormente existen varios algoritmos dentro del método Interpolación entre ellos estudiamos:

- Interpolación bilineal.
- Interpolación bicúbica.
- Interpolación Spline.

2.1.1: Interpolación Bilineal.

La Interpolación bilineal se basa en usar interpolación lineal horizontal y verticalmente con los vértices¹⁶ reales que rodean al vértice de coordenadas no enteras. El valor del vértice dependerá por tanto de 4 vértices (los cuatro más cercanos al vértice de coordenadas no enteras que queremos). La interpolación bilineal se descompone en dos interpolaciones lineales sucesivas, una en cada variable. Si se hacen las cuentas queda que cada uno de los cuatro vértices tiene asignado un peso relacionado con la distancia del vértice no entero a él, tanto horizontal como verticalmente.

La matriz de pesos de los vértices adyacentes queda así:

$$\begin{bmatrix} (1-a)(1-b) & a(1-b) \\ b(1-a) & ab \end{bmatrix}$$

donde a es la distancia horizontal al vértice más cercano por la izquierda y b es la distancia vertical al vértice más cercano por arriba.

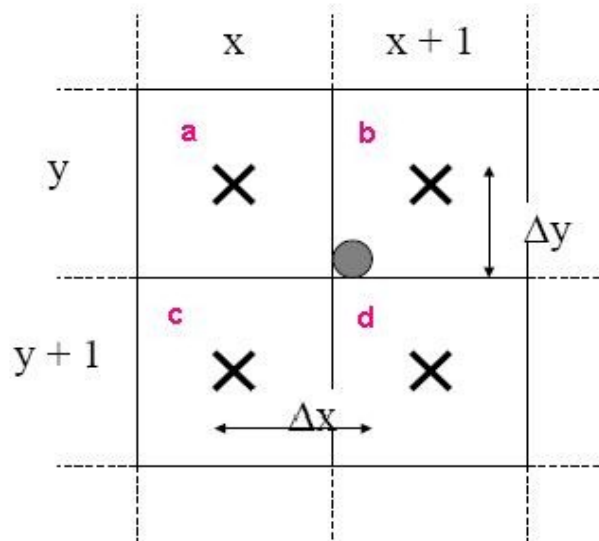


Fig. 7: Ejemplo de interpolación bilineal.

En la Fig.9 se muestra el vértice original y los vértices aproximados a él, a través de ellos se obtiene a partir de la interpolación el valor del vértice original, como se puede ver la interpolación bilineal se basa en encontrar los 4 vértices vecinos en este caso representados por a , b , c y d ; de modo que el calculo seria:

$$a = (1-\Delta x) (1-\Delta y) = (1-(i-x)) (1-(j-y))$$

$$b = \Delta x (1-\Delta y) = (i-x) (1-(j-y))$$

$$c = (1-\Delta x) \Delta y = (1-(i-x)) (j-y)$$

$$d = \Delta x \Delta y = (i-x) (j-y)$$

$$g(i, j) = a f(x, y) + b f(x+1, y) + c f(x, y+1) + d f(x+1, y+1)$$

El principal inconveniente que presenta es la discontinuidad del gradiente del polinomio de interpolación en las fronteras de cada celda.

2.1.2: Interpolación Bicúbica.

La interpolación bicúbica ajusta una superficie cúbica a través de los valores de intensidad existentes, esta basada en dos interpolaciones cúbicas, la interpolación cúbica horizontal, en

las filas existentes (usando 4 puntos) y la interpolación cúbica vertical en todo el espacio usando 4 puntos (usando la anterior interpolación). El valor de un vértice interpolado es una combinación de los valores de los 16 vértices más cercanos como se muestra en la Fig.10.

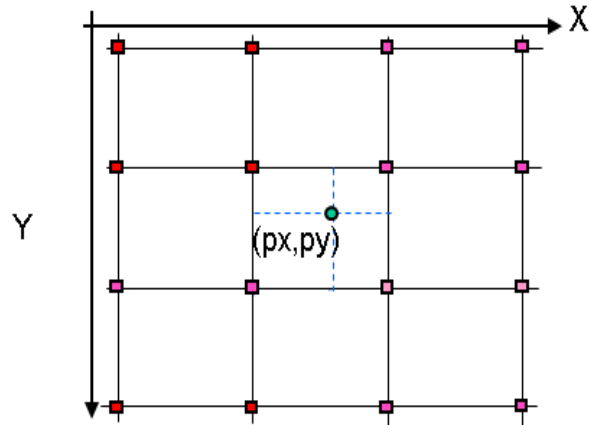


Fig. 8: Ejemplo de interpolación bicúbica.

La Fig.8 muestra la representación de los 16 vértices circundantes al punto (p_x, p_y) , donde el valor de dicho punto (vértice) se puede calcular mediante la media ponderada.

Este método produce una superficie mucho más suave que la interpolación bilineal. Para usar la interpolación bicúbica, la primera y segunda derivada de la superficie deben ser continuas; emplean polinomios de grado 3 que garantizan continuidad, tangencia y curvatura en todos sus puntos. Es el más exacto, pero también el más lento; en este método, se promedia todos los vértices que rodean a cada nuevo vértice para determinar su valor; un inconveniente que presenta es que al tratar de unir varias superficies se produce discontinuidad en las curvaturas.

2.1.3: Interpolación Spline.

La interpolación mediante splines, es una variedad de la interpolación de una función mediante funciones que son polinomios por partes. Lo que distingue a las splines del resto de las interpolaciones con polinomios por partes, es que esta formada por varios polinomios, cada uno definido sobre un subintervalo, que se unen entre sí obedeciendo a ciertas

condiciones de continuidad. Dicho en forma poco rigurosa, se le impone a la función interpoladora una cierta suavidad en la unión de los polinomios que la componen.

Este método de interpolación determina un vértice desconocido basándose en la totalidad de vértices de la animación, a diferencia de los métodos anteriores; es conveniente usar polinomios de grados bajos, así la Interpolación Spline evita el problema de el Fenómeno de Runge el cuál ocurre al usar polinomios de altos grados. No es más que la introducción de errores entre la función y el polinomio de interpolación, este es el principal problema que presenta la interpolación por splines y reside en los puntos que son comunes a dos intervalos (extremos) por esos puntos deben pasar los splines de ambos intervalos, pero para que la interpolación sea ajustada, conviene que el punto de unión entre dos splines sea lo más "suave" posible (ej. evitar puntos angulosos), por lo que se debe velar que en esos puntos ambos splines tengan derivada común.

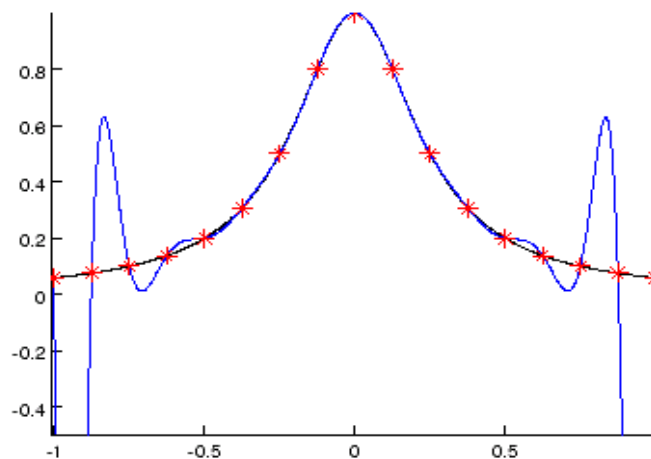


Fig. 9: Ejemplo de Fenómeno de Runge.

En la Fig.9 se ve un ejemplo del error que se introduce al interpolar funciones de altos grados, donde las elevaciones que se pueden observar a ambos lados son ejemplos de puntos angulosos, a pesar de que la función interpolante cumple los requisitos impuestos ya que pasa por todos los puntos de la función.

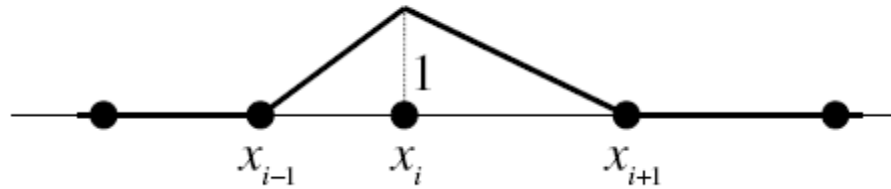


Fig. 10: Ejemplo de Interpolación Spline de grado 1 respecto a X_i .

En la figura anterior se muestra un ejemplo gráfico de como hacer un espacio de función suficientemente suave y manejable, donde la función es continua y se restringe al intervalo $[X_{i-1}, X_{i+1}]$, donde X_i es el vértice a interpolar.

La interpolación spline a la hora de obtener la curva polinómica a partir de n puntos introducidos cumple con la propiedad de:

- La curva pasa por todos los puntos (interpolación).
- Dos puntos contiguos se representan mediante un polinomio de grado 3.
- La curva es continua en todos los puntos.

2.2: Tabla de comparación de posibles algoritmos a usar en la implementación de reducción de KeyFrame en una animación.

Tipo de interpolación	Cantidad de vértices necesario para interpolar	Dimensiones	Velocidad de procesamiento
Bilineal	4 vértices	Bidimensional (horizontal y vertical)	Proceso lento
Bicúbica	16 vértices	Bidimensional (horizontal y vertical)	Proceso más lento
Spline	A partir de 16 vértices	N dimensiones	Proceso rápido

Tipo de interpolación	Costo de implementación	Continuidad de el polinomio	Grado de salida del polinomio
Bilineal	Costosa	Puede introducir discontinuidad	Polinomio grado 2
Bicúbica	Más Costosa	Puede introducir discontinuidad	Polinomio grado 3
Spline	Costosa	Continua en todos los puntos	Polinomio grado 1

2.3: Estructura del fichero con extensión anx, desarrollado en la facultad.

A continuación se muestra la estructura que tiene hasta el momento el fichero con extensión anx, desarrollado en la facultad:

- 16 bytes de información de encabezado.
- El resto es una lista de nodos, donde cada uno representa un hueso.
- Los nodos tienen información propia de el y su padre y la lista de matrices que representan la deformación por frame.
- Las matrices son 4 vectores, los tres primeros describen la rotación y el cuarto la traslación, en total 12 bytes.
- Los vectores son arreglos de 3 float, es decir 12 bytes.

Estructura del encabezado.

```
char FileID [4] = "ANX1" = 0x31584E41 (4 bytes)
int NodesCount (4 bytes)
int FramesCount (4 bytes)
float FrameRate (4 bytes)
```

Lista de huesos

AnimatedNode1, AnimatedNode2, AnimatedNode3... [NodesCount]

Estructura de AnimatedNode

char Name[32] (32 bytes)
word Handle (2 bytes)
char ParentName[32] (32 bytes)
word ParentHandle (2 bytes)
Maxtrix1, Matrix2, Matrix3... [FramesCount]

Estructura de Matriz

Vector Row1 (12 bytes)
Vector Row2 (12 bytes)
Vector Row3 (12 bytes)
Vector Row4 (12 bytes)

Después de haber analizado la estructura de dicho fichero, se decidió que para poder aplicarle la modificación propuesta, a este, se le debería introducir una nueva información, variando de esta forma su estructura, a la que a continuación le proponemos como solución.

- 16 bytes de información de encabezado.
- El resto es una lista de nodos, donde cada uno representa un hueso.
- Los nodos tienen información propia de el y su padre, 12 listas comprimida con la cantidad **A** de bytes cada una, que al descifrarla, representan una lista de matrices que según su valor se lee o se copia un valor leído en las listas de matrices que representan la deformación por frame.
- Las matrices son 4 vectores, los tres primeros describen la rotación y el cuarto la traslación, en total 12 bytes.
- Los vectores son arreglos de 3 float, es decir 12 bytes.

Estructura del encabezado.

char FileID [4] = "ANX1" = 0x31584E40 (4 bytes)
int NodesCount (4 bytes)

int FramesCount (4 bytes)
float FrameRate (4 bytes)

Lista de huesos

AnimatedNode1, AnimatedNode2, AnimatedNode3... [NodesCount]

Estructura de AnimatedNode

char Name[32] (32 bytes)
word Handle (2 bytes)
char ParentName[32] (32 bytes)
word ParentHandle (2 bytes)
ListBV1, ListBV2, ListBV3... [12]
Lista_de_valores

Estructura de ListBV

char valor 1, valor2, valor3...[A] (1 byte)

Estructura de Matriz

Vector Row1 (12 bytes)
Vector Row2 (12 bytes)
Vector Row3 (12 bytes)
Vector Row4 (12 bytes)

Nota:

NodesCount: Cantidad de huesos que va a tener la animación.

FramesCount: Cantidad de Frames que tiene la animación.

FrameRate: Cantidad de Frames por segundo a la que corre la animación

A: =((FramesCount%8 == 0) ? FramesCount/8: FramesCount/8 + 1);

Ejemplo de algún valor descifrado:

valor = 255

int desif = 255 & (**int**)valor = 255;

El valor **desif** en binario equivale a 11111111, esto se le asigna a una lista de boléanos correspondiente a cada valor de una matriz de cada frame, y va a indicar si este valor hay que leerlo del fichero, o se puede sustituir por el valor anterior correspondiente en la matriz anterior.

2.4: Consideraciones del capítulo.

La animación interpolada es una forma eficaz de crear movimiento y cambios a lo largo del tiempo y de reducir al mínimo el tamaño del archivo. Para esto se llegó a la conclusión que el algoritmo matemático de interpolación más eficiente es la interpolación spline, pues si se habla de reducciones que no involucren tanto costo, interpolación spline es la mejor opción a usar debido a que proporciona resultados de calidad y es rápida su ejecución, ya que presenta continuidad en todos los puntos de la curva, el valor de la interpolación depende solo del subconjunto de puntos más próximos al punto a interpolar. Por lo que se recomienda su implementación para un próximo desarrollo de la herramienta para la reducción de KeyFrame.

La extensión de fichero `anx` que se obtuvo realizando una serie de transformaciones al formato de extensión con el cual trabajaba la facultad, permitió que este se pudiera aplicar al proyecto "Juegos CNEURO", siendo esta la solución del problema con el cual contaba.

CAPÍTULO 3: FUNCIONAMIENTO DEL SISTEMA

En este capítulo se muestra detalladamente las nuevas funcionalidades con las cuales constará la herramienta existente, en este se exponen los Requerimientos Funcionales como los No Funcionales; así como también los Casos de Usos y los actores que interactúan con el sistema, además de las descripciones correspondientes a cada Caso de Uso.

3.1: Definición de los Requisitos Funcionales del sistema.

En el proceso de desarrollo de cualquier sistema lo primero a tener en cuenta es la de identificar los requerimientos funcionales y no funcionales con los que contará el sistema; para definir posteriormente el diseño y la modelación, y así obtener un software que cumpla con todas las necesidades requeridas.

Requisitos funcionales:

- **RF1:** Cargar animaciones.
- **RF2:** Reducir KeyFrames.
- **RF3:** Visualizar animación.
- **RF4:** Salvar animación optimizada.

Estos requerimientos funcionales son los que van a hacer que la herramienta MODANS pueda ser usada por el proyecto CNEURO para cumplir cabalmente con las exigencias del proyecto.

3.2: Definición de los Requisitos No Funcionales del sistema.

Los requerimientos No funcionales son las propiedades con las cuales va a contar el producto, están pueden ser el rendimiento de la herramienta, restricciones del entorno e implementación entre otros; a continuación escribimos con las que contará el producto final nuestro.

Requisitos No Funcionales:

- **RNF1:** Diseño e implementación.
- **RNF2:** Hardware.
- **RNF3:** Software.
- **RNF4:** Interfaz Externa.
- **RNF5:** Portabilidad.

Diseño e implementación:

- Lenguaje de programación C++.
- Microsoft DirectX9 SDK
- Visual Studio 2005.
- Scene Toolkit 2.4 (STK).

Hardware:

- Memoria RAM mayor o igual a 256MB.
- Computadora Pentium II o superior.

Software:

- 3D Max Studio 5.
- DirectX9 o superior.
- Windows 2000 o superior.

Interfaz Externa:

- El ejecutable contará con una interfaz amigable y sencilla para el uso del usuario.

Portabilidad:

- El sistema funcionará sobre la plataforma Windows.

3.3: Definición de los Casos de Uso.

Los Casos de Uso que se definieron para darle la funcionalidad necesaria a la herramienta para una mejor utilización de la misma en el proyecto CNEURO fueron: Cargar Animación, Reducir KeyFrames, Visualizar animación y Exportar Animación. Mediante cada una de estas funcionalidades se da a conocer una mejor óptica de cómo está conformado el sistema. A

continuación se ejemplifica la descripción textual de estos Casos de Uso, así como la interacción que tienen con el sistema.

Para dar solución al caso de uso Visualizar animación, se hizo necesario implementar una aplicación auxiliar (Restt2.exe) capaz de restaurar el fichero exportado por la herramienta (MODANS), para de esta forma obtener un fichero con las características necesarias para ser visualizado utilizando la herramienta Scene ToolKit.

Restt2, carga el fichero modificado con nombre “modificado.anx”, e interpreta las listas de booleanos que contienen la información sobre cual key frame fue borrado por MODANS, y cual debe ser leído, pues se mantuvo intacto en el fichero, este proceso da como resultado un fichero llamado “normal1.anx” que tiene las características requeridas para ser leído por el Visualizador de animaciones.

El código que se implementa en la aplicación Restt2, debe ser integrado a la solución desarrollada por el proyecto Juegos CNeuro, para garantizar la correcta lectura del fichero sin la necesidad de una herramienta adicional.

3.3.1: Definición de los actores del sistema.

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

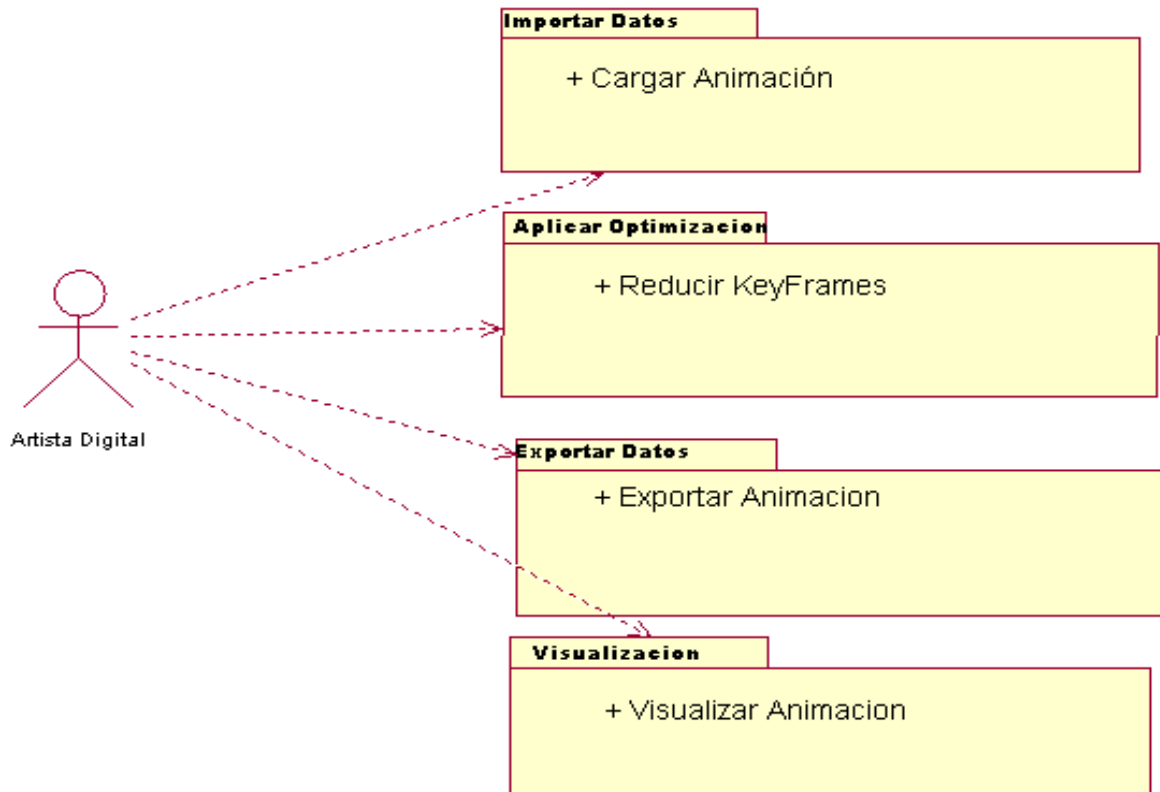
CU_1	Cargar Animación
Actor	Artista Digital
Descripción	Carga animaciones hechas en el 3D Studio Max 5.
Referencia	RF1

CU_2	Reducir KeyFrames
Actor	Artista Digital
Descripción	Eliminar los Key Frames utilizando técnicas de comparación entre ellos teniendo en cuenta el error introducido.
Referencia	RF2

CU_3	Visualizar Animación
Actor	Artista Digital
Descripción	Mostrar la animación, una vez que halla sido modificada y exportada por la herramienta.
Referencia	RF3

CU_4	Exportar Animación
Actor	Artista Digital
Descripción	Salvar la animación, en el formato especificado por el actor.
Referencia	RF4

3.3.2 Diagrama de Casos de Uso por paquetes:



En el diagrama que se muestra anteriormente, están representados por paquetes los Casos de Uso que se definieron; agrupados por la funcionalidad que representan los mismos para la obtención del producto deseado. Mediante esto se facilita en un futuro la posible implementación de la herramienta utilizando otros lenguajes o tecnología.



Fig. 11: Paquete Importar Datos.



Fig. 12: Paquete Aplicar Optimización.



Fig. 13: Paquete Exportar Datos.



Fig. 14: Paquete Visualización.

3.3.3: Descripción textual de los 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 5.
Actores	Artista Digital.
Resumen	El artista digital inicia el caso de uso cargando los ficheros con formato .ANX .
Referencias	RF1
Acción del actor	Respuesta del sistema
1. El artista comienza el caso de uso ejecutando la aplicación y selecciona la	1.1 El sistema muestra una ventana de búsqueda.

opción de cargar animación.	
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	Reducir KeyFrames.
Propósito	El objetivo de este caso de uso es eliminar los KeyFrames 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 e introduce el margen de error que desea que sea aplicada a la modificación.
Referencias	RF2
Acción del actor	Respuesta del sistema
1. El artista comienza el caso de uso seleccionándola opción 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 margen permisible de error para los vectores de rotación y traslación.
3. El artista entra los márgenes de error permisible.	3.1 El sistema modifica la animación.

Caso de uso	
CU_3	Visualizar Animación
Propósito	El objetivo de este caso de uso es visualizar la animación modificada..
Actores	Artista Digital.
Resumen	El artista digital inicia el caso de uso cuando selecciona la animación modificada, se selecciona la animación y se ejecuta en un fichero, donde luego la animación original y la modificada pasan a visualizarse en el demo de la herramienta Scene Toolkit(STK).
Referencias	RF3
Acción del actor	Respuesta del sistema
<p>1. El artista comienza el caso de uso seleccionando la animación que ha sido modificada con anterioridad.</p> <p>1.1. Se le da el nombre de <i>modificado</i> a la animación seleccionada por el artista.</p>	<p>1.2. Ejecutando la aplicación Restt2.exe, se carga la animación con el nombre <i>modificado</i>, y se le hace una transformación necesaria, devolviendo un archivo llamado <i>normal</i>.</p>
<p>2. Se guarda la animación sin transformar con el nombre de <i>original</i>.</p> <p>2.1. Se guarda la animación obtenida con el nombre <i>normal</i> y la <i>original</i> en la carpeta que presenta el visualizador.</p> <p>2.2. Se ejecuta el Demo visualizador de la herramienta por el artista.</p>	<p>2.3. El sistema muestra la animación original cuando el usuario selecciona la tecla con el número 2.</p> <p>2.4. El sistema muestra la animación modificada cuando el artista selecciona la tecla número 1.</p>

Caso de uso	
CU_4	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	1.1. 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

En la Fig.13 se muestra todo lo relacionado a los conceptos y las relaciones que se establecen en el dominio; esto se hace con el objetivo de mejorar la comprensión del desarrollo de la herramienta. Los conceptos que aparecen nuevos en el diagrama vienen dados por Almacenador.ANX y ReducirKF.ANX; pues a la aplicación se le introdujo un nuevo formato.

El Manejador cuenta con los recursos con los que trabaja la aplicación que en este caso son los conceptos animación y Esqueleto; además el manejador contiene un arreglo de Almacenador poseyendo las características del Optimización contando de esta forma con el concepto Almacenador.ANX que es el nuevo concepto con el cual estamos tratando, donde se encargaría de cargar y exportar recursos desde ficheros con formato ANX.

Se hizo necesario el uso del modelo del dominio, pues la herramienta a desarrollar no contaba con la determinación exacta de las fronteras que exige el modelo de negocio; las cuales implican tener bien definido quienes son los trabajadores, cuales son sus actividades dentro del proceso, quienes son los beneficiados con cada uno de estos procesos. Es entonces cuando pasamos a identificar conceptos, dándole definiciones a los mismos y estableciendo relaciones entre si.

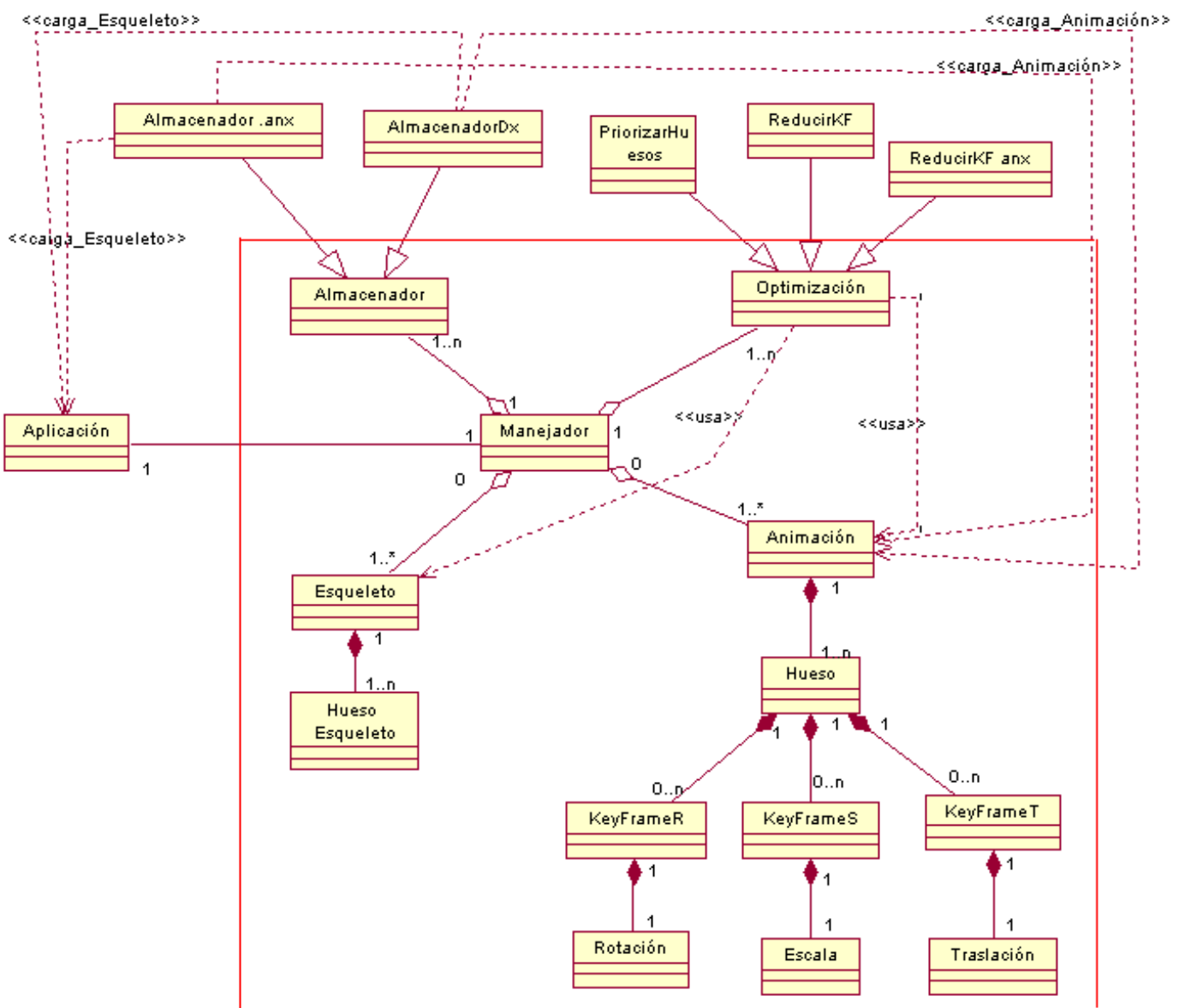


Fig. 15: Modelo Conceptual.

3.5 Arquitectura del sistema.

Se trabajó con el Framework de la herramienta MODANS, definido por todas las clases que componen las animación de huesos (son las clases que se encuentra encerradas en el cuadro rojo de la Fig.15: Modelo Conceptual); es la plataforma que soporta las implementaciones de todas las funcionalidades, por lo que es la estructura base del programa. En la figura siguiente se presenta por medio de un diagrama de clases del análisis, donde están plasmadas las clases que presenta el sistema.

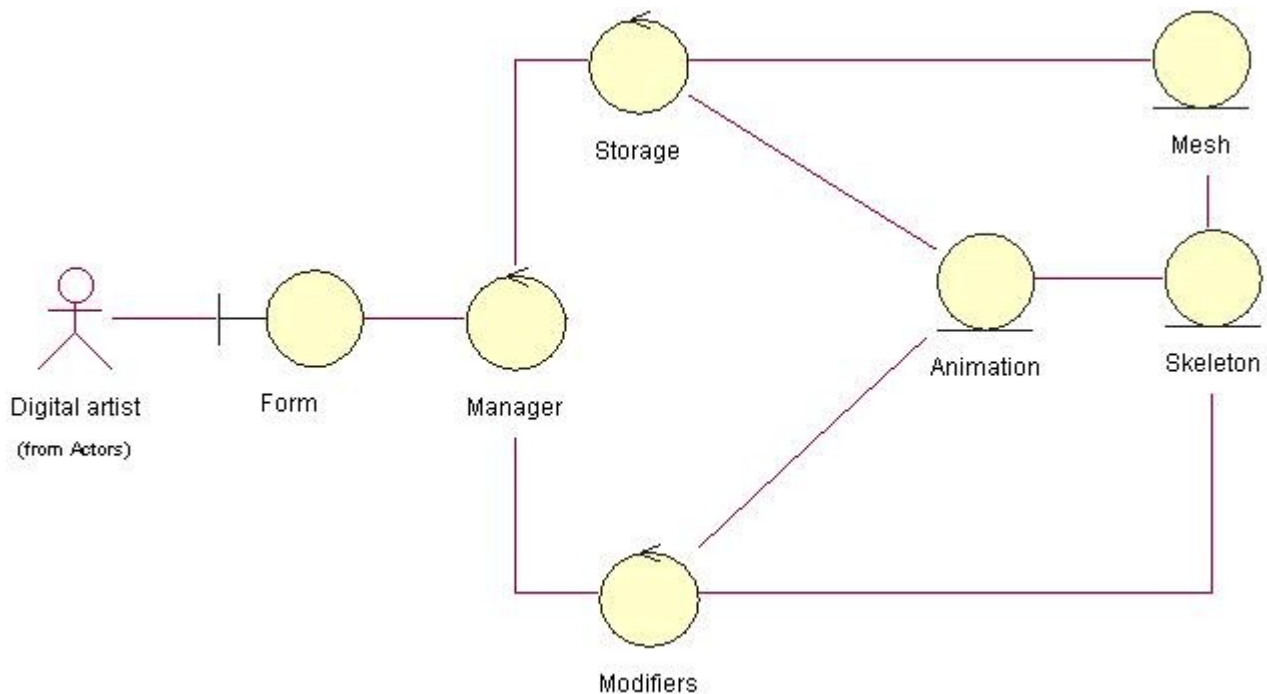


Fig. 16: Diagrama de Clases del Análisis.

El usuario (Digital artist) interactúa con el Framework mediante la clase interfaz “Form” que a su vez recibe las peticiones realizadas por el artista digital y de transmitir las a la clase controladora “Manager”; quien se encarga de clasificar las peticiones y asignar tareas a las demás clases controladoras existentes.

Así de esta forma si el usuario desea cargar o exportar, el “Manager” le envía la información al “Storage” de la petición hecha y esta se ocupa de cargar o exportar con la extensión de fichero que se deseada. Si la petición que recibe el Manager es de alguna función de

optimización específica entonces esta petición es pasada a “Modifiers”; para que realice los cambios deseados a la animación (Animation) o en el esqueleto “Skeleton”.

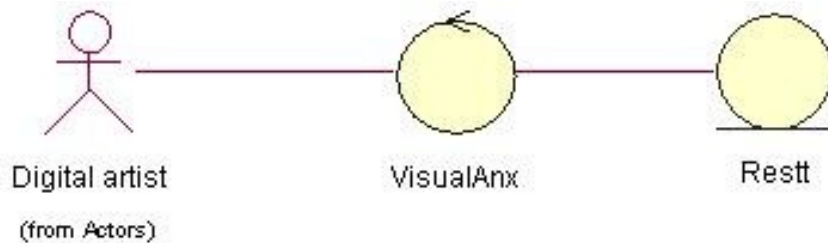


Fig. 17: Diagrama de clase del Análisis.

El usuario (Digital artist) interactúa directamente con la clase controladora “VisualAnx”, siendo esta asistida por la clase entidad Restt, para dar solución a la petición de visualizar animación primeramente se debe trabajar con clase Restt, donde más tarde la “VisualAnx” se encarga propiamente ella de visualizar la animación original tanto como la modificada.

3.6: Funcionamiento del Framework.

Comienza cuando el usuario realiza una serie de peticiones al sistema, esto lo logra mediante la clase interfaz presente; cada petición que se realiza genera un flujo de acciones que se representan con un diagrama de interacción que son: colaboración o secuencia; que no es más que todos los pasos que internamente se tienen que realizar para dar respuesta al usuario.

Se seleccionó para la representación de estos flujos el diagrama de interacción de secuencia ya que es más fácil su comprensión a la hora de dicha representación, este ofrece un modelo de cómo interactúa cada objeto del sistema entre sí. A continuación se presenta las figuras que dan lugar a esta representación para cada funcionalidad que se definió que la herramienta poseería.

3.6.1: Petición de inicializar.

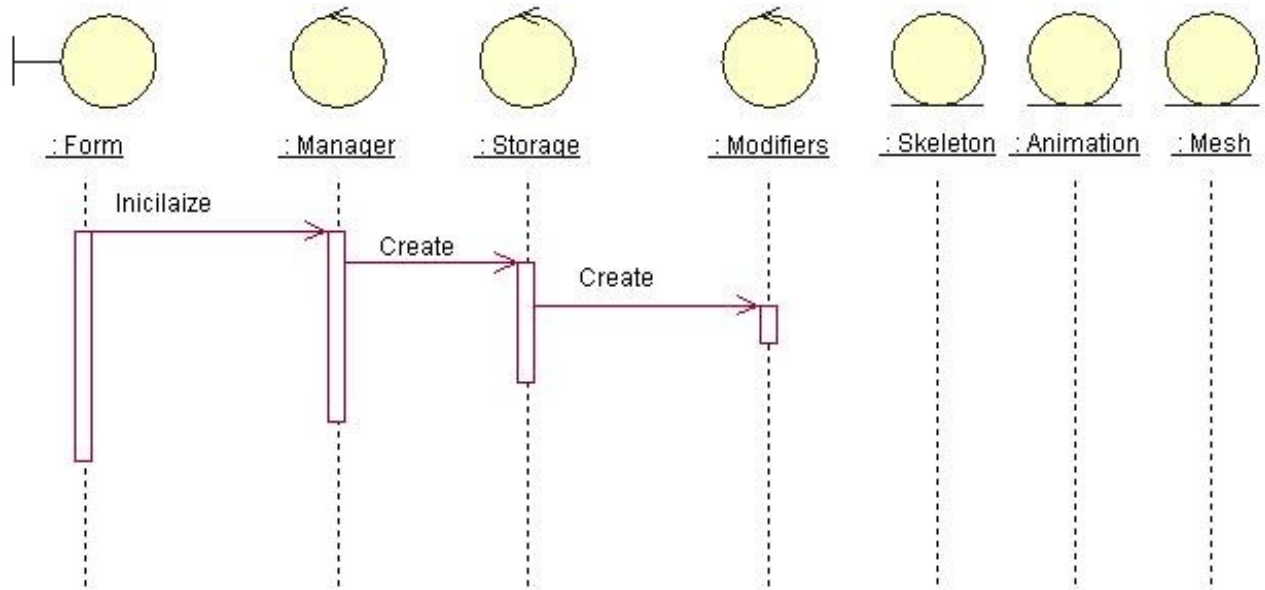


Fig. 18: Diagrama de secuencia para la petición de Inicial.

Cuando se ejecuta la aplicación, el “Manager” recibe la petición de inicializar y entonces crea todos los “Storages” como también los “Modifiers”.

Los “Storages” contienen la información que se necesita para cargar o exportar tanto modelos como animaciones. De esta manera, se hace más modularmente para el sistema se le pueda ser agregado otro formato sin causar problemas en el código ya existente, de este modo las optimizaciones se crean igual y fácil.

3.6.2: Petición de Cargar Animación.

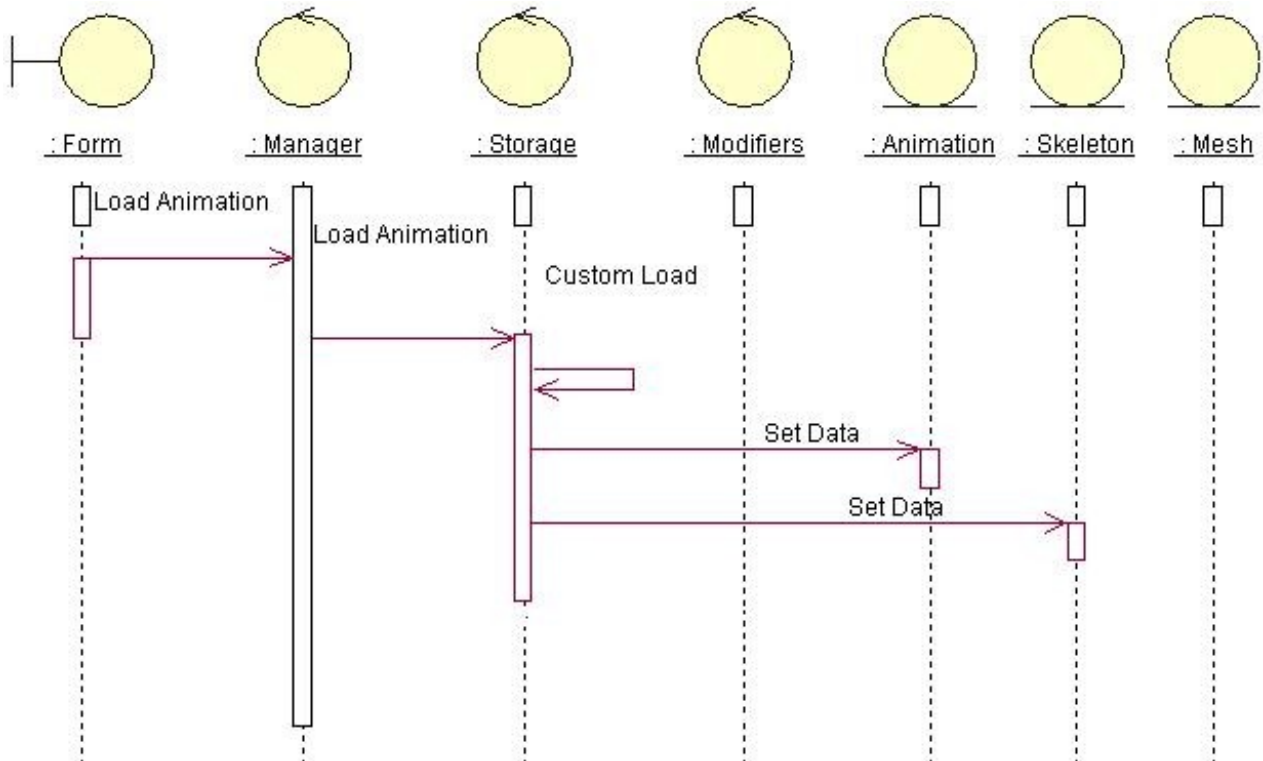


Fig. 19: Diagrama de secuencia para la petición de Cargar Animación.

La petición de cargar animación es comunicada al “Managers” y le pasa la petición al “Storages”, luego el storages verifica si puede cargar la animación en el formato de fichero especificado. Si encuentra que puede cargar el formato entonces asigna la petición, donde el storages busca los datos necesarios en el fichero y los inserta en la Animation.

3.6.3: Petición de Aplicar Modificación.

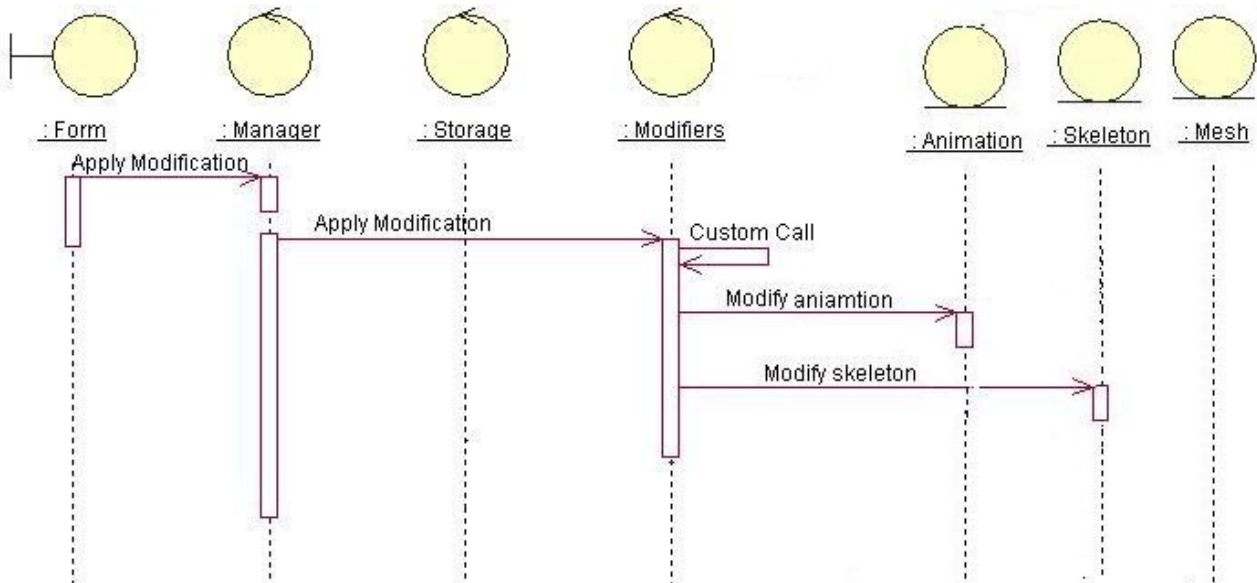


Fig. 20: Diagrama de secuencia para la petición de Aplicar Modificación.

La petición de aplicar la modificación, el “Managers” comunica la petición de modificación al “Modifiers” y este se encarga de clasificar el tipo de modificación que se le va a aplicar a la animación y realiza la modificación deseada a la animación (Animation) y al esqueleto (Skeleton).

3.6.4: Petición de Exportar Animación.

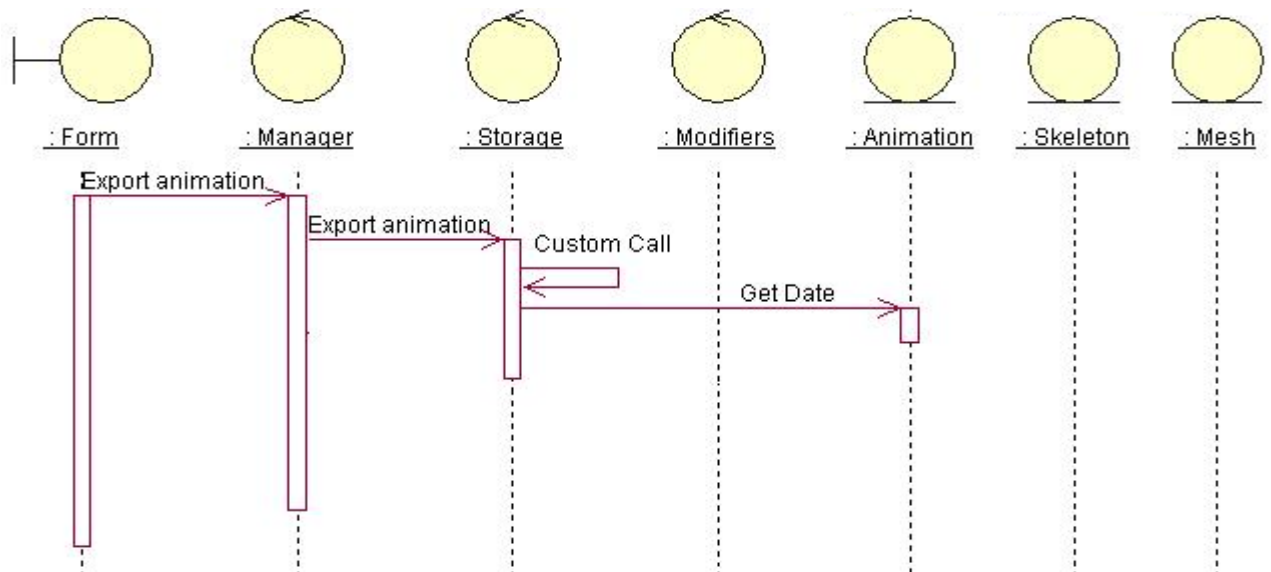


Fig. 21: Diagrama de secuencia para la petición de Exportar Animación.

Cuando es solicitada la petición de exportar animación el “Manager” comunica la petición al “Storage” de exportar la animación, especificando la animación y el formato en el cual se va a realizar. El Storage consta con una jerarquía de storages, los cuales tienen especificado un formato tanto para cargar o exportar la animación, por lo que cuando se hace la petición se activa el storage que puede exportarla.

3.6.4: Petición de Visualizar Animación.

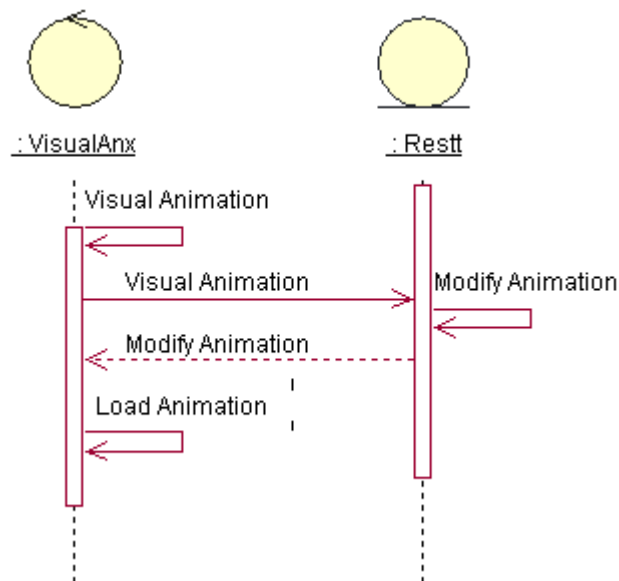


Fig. 22: Diagrama de secuencia para la petición de Visualizar Animación.

Para darle cumplimiento a la petición de “Visualizar Animación”, primeramente se debe contar con una animación modificada, la cual es modificada nuevamente por la clase entidad “Restt”, donde esta se encarga de restaurar los Key Frames que se eliminaron al modificarse la animación y se pasa entonces a mostrar la animación desde la clase controladora.

3.7: Manual de la herramienta para el usuario.

La herramienta MODANS es una aplicación para trabajar con animaciones de huesos. Su principal objetivo es reducir la cantidad de espacio en memoria que se llevan los ficheros de las animaciones al ser guardados.

MODANS fue diseñada para trabajar en la plataforma Windows, su implementación en Visual Studio 2005 con el lenguaje de programación C++. Las funcionalidades de la aplicación vienen dadas por Cargar Animación, Reducir KeyFrame, Asignar Prioridad y Exportar Animación, además de contar con una ayuda para el usuario.

A continuación se presentan una serie de imágenes que ayudaran en la comprensión del manejo de la aplicación por el usuario, con las figs.23,24,25,26,27,28,29,30,31,32,33 donde se explica detalladamente los pasos a seguir para no perderse y hacer un buen uso de la herramienta, y así lograr una animación con calidad.

Inicialmente la herramienta al ejecutarse presenta en la parte superior izquierda un menú llamado archivos y ayuda. En archivos se encuentran las funcionalidades que brinda la aplicación; además de tener la opción de salir. El usuario puede o no pinchar la opción de Cargar animación si lo desea ya que al ejecutarse la aplicación esta ya activa la opción que tiene que ver con la funcionalidad de cargar animación.

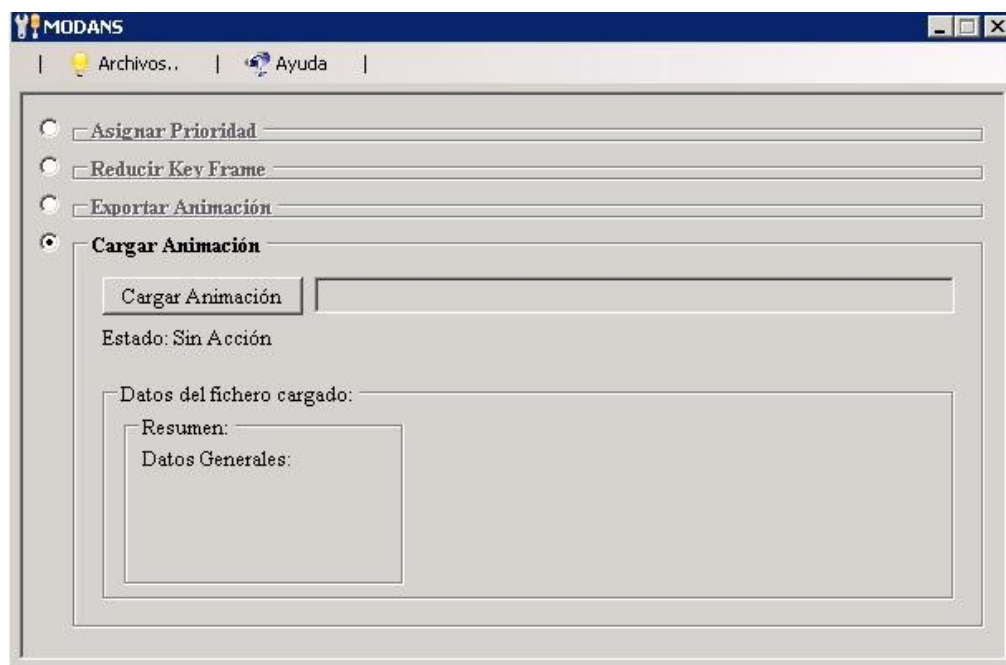


Fig. 23: Interfaz principal de la herramienta MODANS.

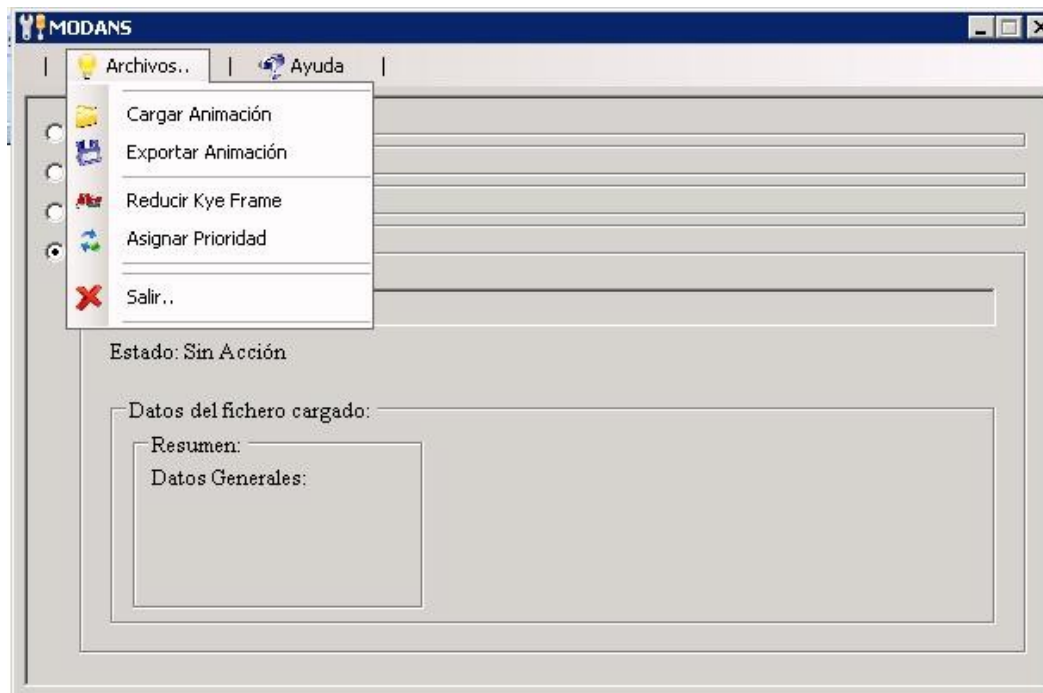


Fig. 24: Menú archivos de la herramienta MODANS.

La funcionalidad Cargar Animación es lo primero que el usuario debe realizar ya que debe seleccionar la animación a la cual le desea aplicar la modificación; cuando el usuario realiza esta operación se activa cargar animación, la herramienta inicialmente se encuentra en la interfaz cargar animación (fig.25 [A]) por lo que el usuario solo debe dirigirse al botón cargar animación (fig.25 [B]) y escoge desde una dirección específica la animación que desea ;a la hora de escoger la animación que va a modificar puede buscar las animaciones en el formato específico ANX files[* .anx] o X files[* .x] o si quiere puede ver todas las animaciones que contenga la carpeta con ambos formatos seleccionando All files[* .*], ya una vez decidida cual será la animación la carga y pasa a desactivarse el botón cargar animación.

Una vez escogida la animación, la herramienta muestra si la animación pudo ser cargada o no según la extensión de sus ficheros, en caso de que no pueda cargarse satisfactoriamente la animación; la herramienta le avisa al usuario que ocurrió un error de lectura (Fig.25 [C]).

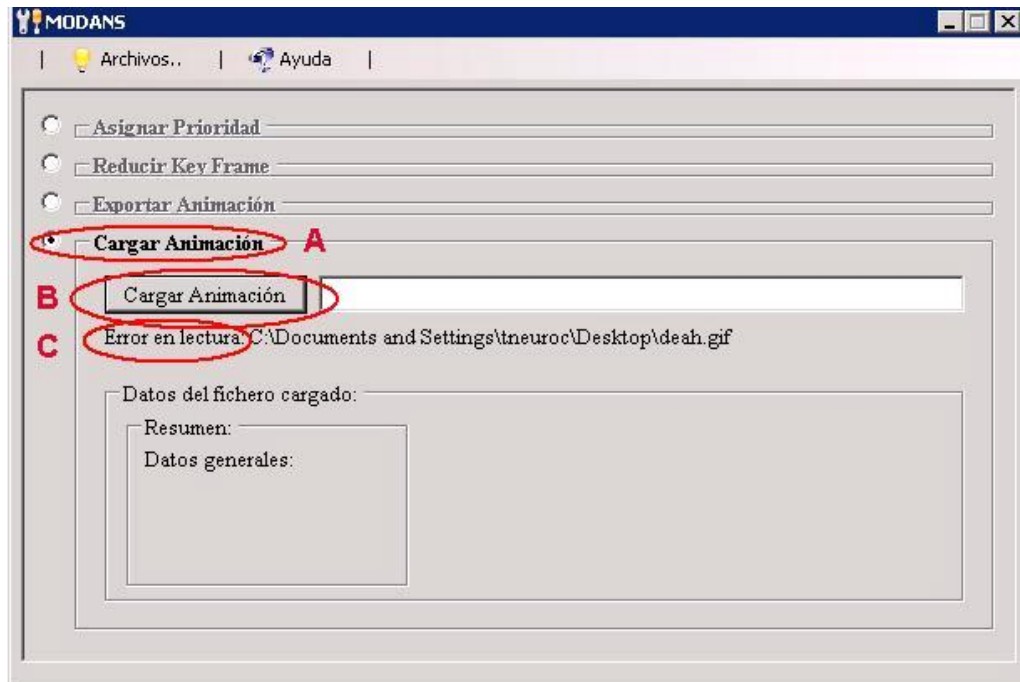


Fig. 25: Funcionalidad Cargar Animación con formato inadecuado.

Si la animación fue cargada con éxito se activan todas las funcionalidades y entonces se muestra la dirección desde donde se cargo la misma (fig.26 [A]). Una vez definida y cargada la animación, en esa misma interfaz se van mostrando en los datos de fichero cargado el resumen (fig.26 [B]) de la animación que no es más que la jerarquía de huesos que contiene.

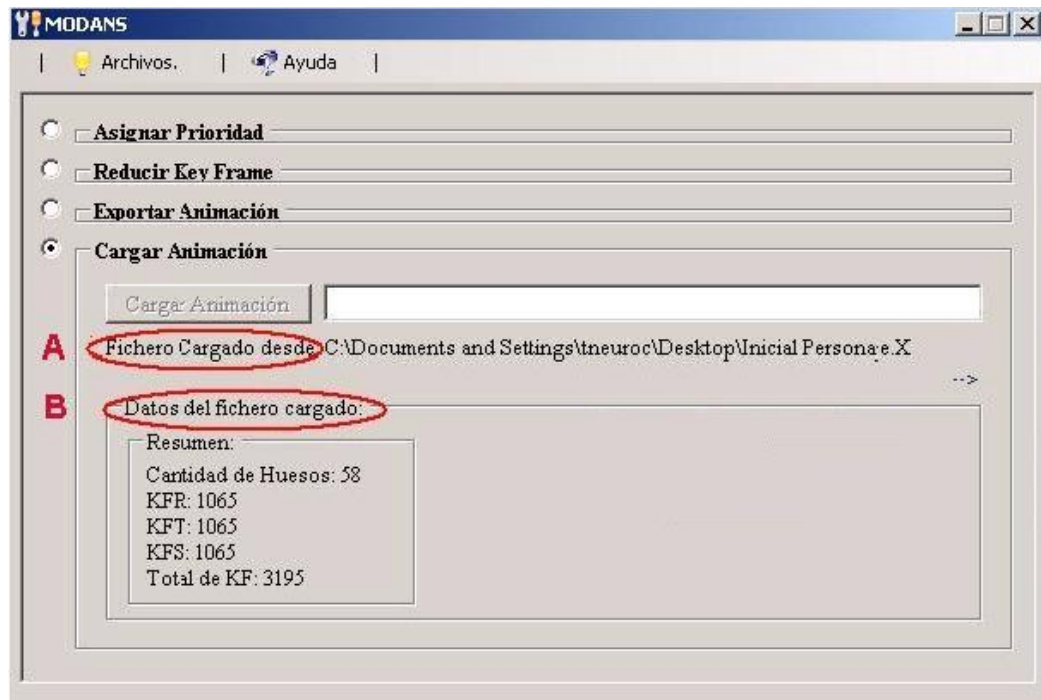


Fig. 26: Funcionalidad Cargar Animación con formato .X.

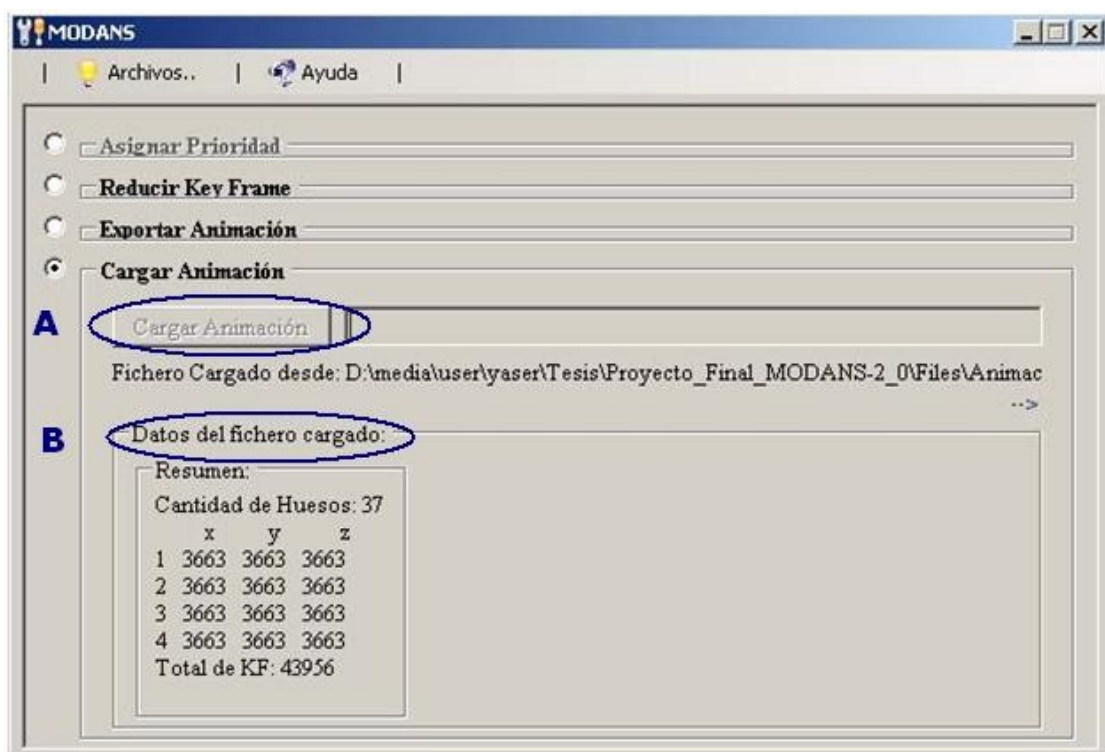


Fig. 27: Funcionalidad Cargar Animación con formato .ANX.

Al cargar una animación con formato anx, los datos del fichero no se muestran de igual forma sino a través de una matriz (Fig. 27[B]).

Luego el usuario puede o no pasar a activar la funcionalidad de Asignar Prioridad siempre y cuando se trate de una animación con formato .x, si lo hace selecciona la animación a la cual le va a hacer la modificación (fig.28 [A]) y muestra en el resumen los datos de la jerarquía de hueso, se selecciona la prioridad que se le va a dar (fig.28 [B]) que viene predeterminada por la aplicación baja:0, media:128 o alta:255 de lo contrario el usuario puede darle la que el estime conveniente (fig.28 [C]); luego selecciona el hueso de la jerarquía (fig.28 [D]), (fig.28 [E]) se puede seleccionar en la animación varios huesos de jerarquía según lo estime el usuario, y va al botón asignar jerarquía (fig.28 [F]) y el hueso y todos sus hijos en la jerarquía del esqueleto toman el valor de prioridad indicado, mostrando un mensaje que la asignación ha sido realizada con los datos principales (fig.28 [G]).

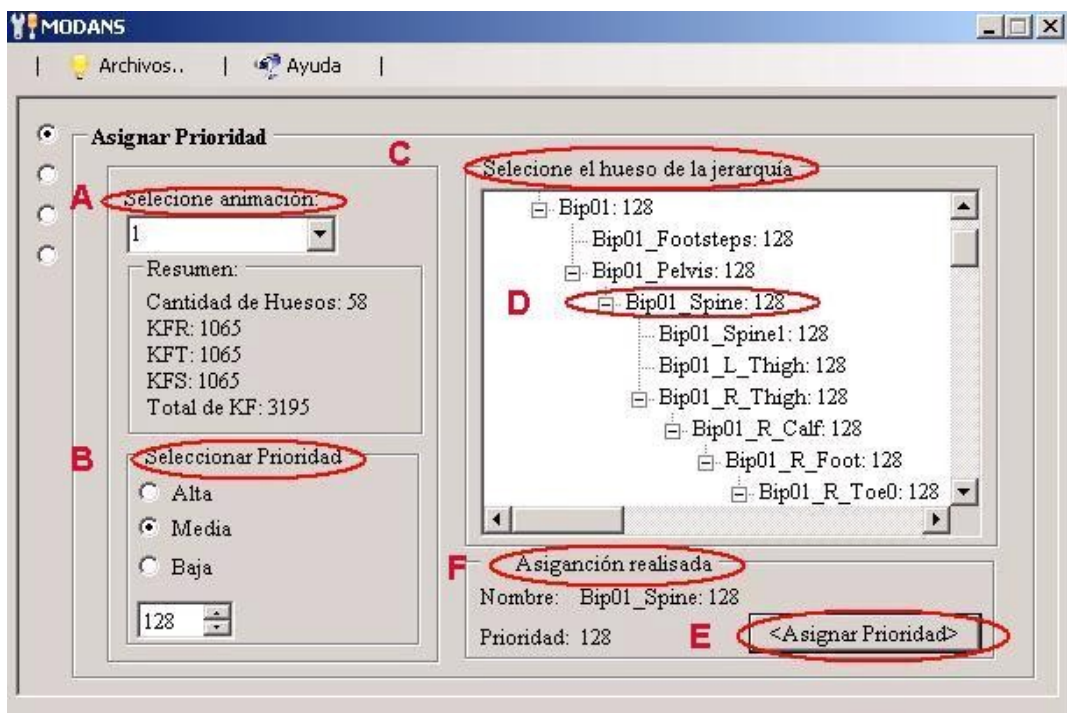


Fig. 28: Funcionalidad Asignar Prioridad.

Cuando el usuario pasa a la funcionalidad de Reducir Key Frame, selecciona la imagen con la cual ha venido trabajando hasta el momento de la lista, en (fig.29 [A]) se introduce el

margen de error permisible definido por rotación, traslación y escala mediante los cuales la animación adquiere una pequeña variación, en (fig.29 [B]), se muestran los datos originales que tiene la animación y los de la animación modificada se muestran una vez que se haya pinchado el botón Reducir Key Frame (fig.29 [C]); una vez reducida la animación, si el usuario no esta de acuerdo con los resultados obtenidos puede seguir reduciendo la nueva imagen obtenida la cual aparecerá en la lista selección de animación(fig.29 [A]) y se repiten los pasos descritos anteriormente en esta misma funcionalidad.

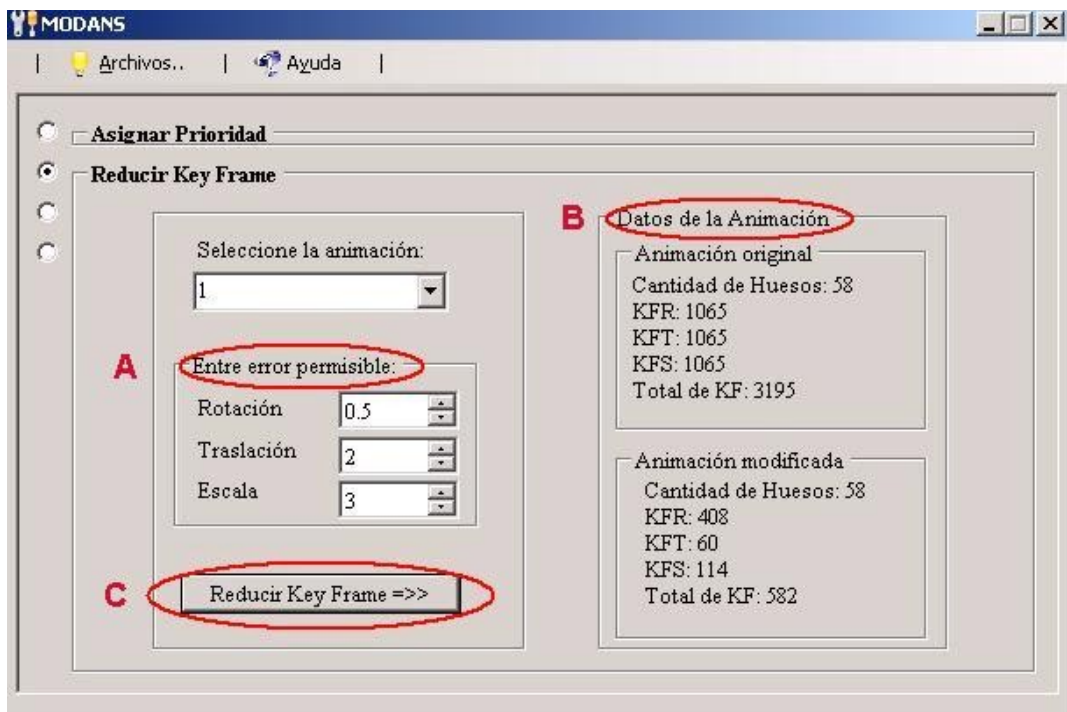


Fig. 29: Funcionalidad Reducir Key Frame para formato .X.

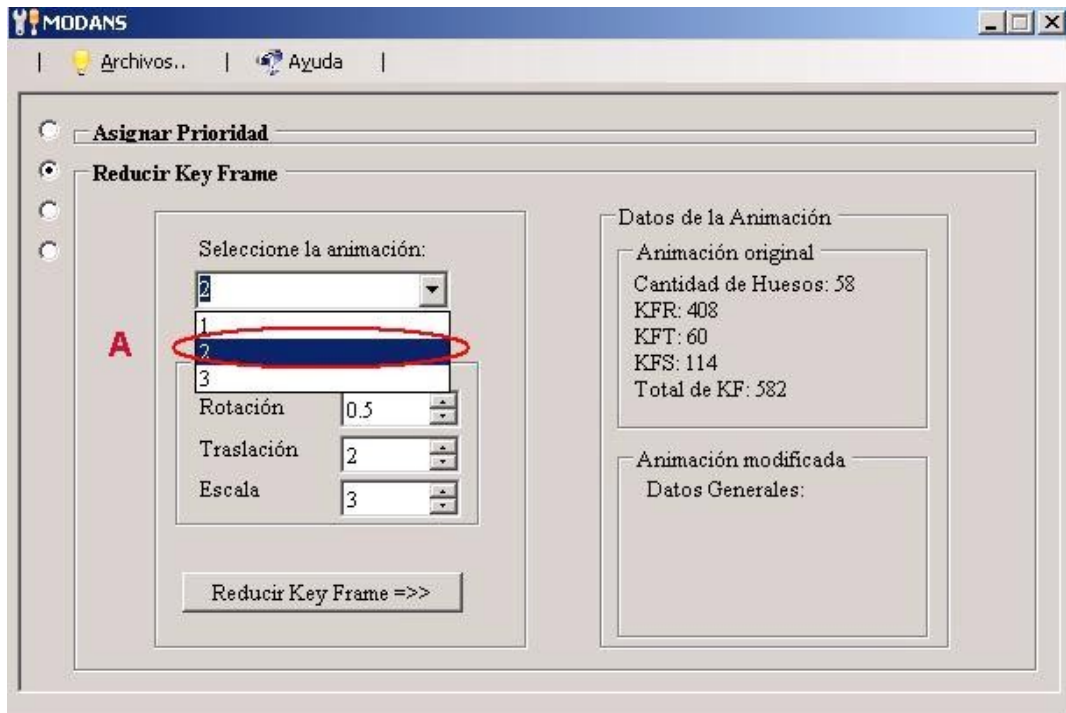


Fig. 30: Funcionalidad Reducir Key Frame para formato ANX.

Consta de tener la misma secuencia de pasos para realizar esta funcionalidad, a diferencia de que el margen de error permisible esta definido por Rotación VR1 hasta VR3 y traslación e igualmente la información de los datos de la animación se muestran en forma de matriz (fig.31[A]).

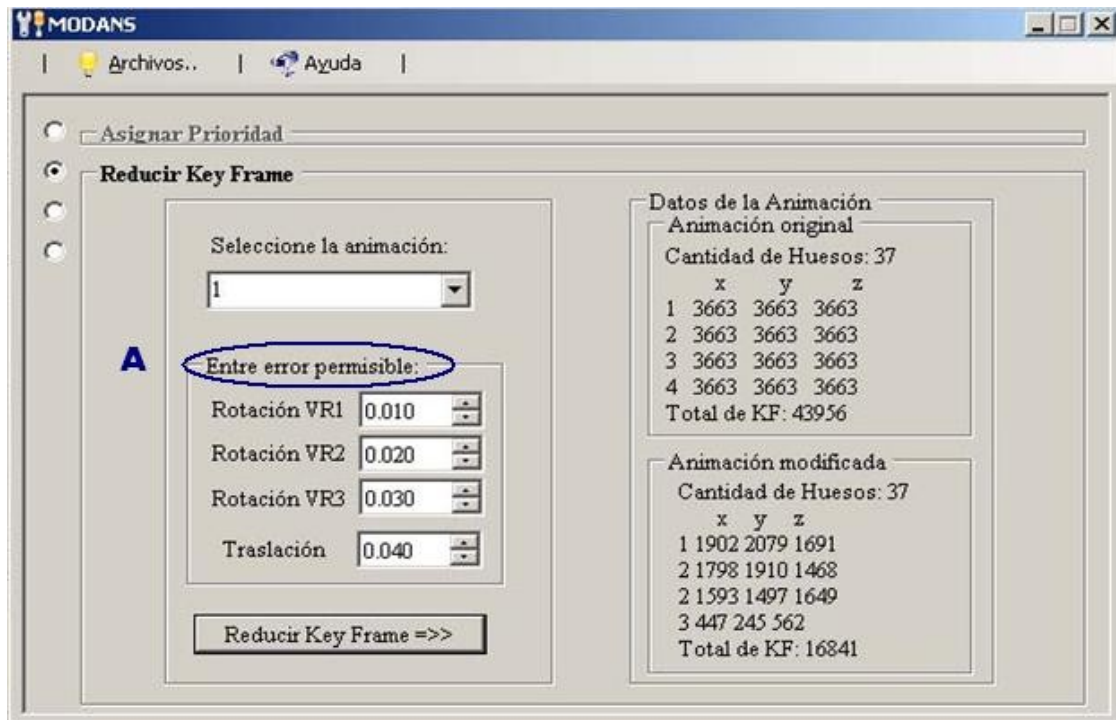


Fig. 31: Funcionalidad Reducir Key Frame para formato ANX.

Por último la funcionalidad Exportar Animación es activada para guardar los cambios realizados a la animación. Se selecciona de la lista la animación a guardar que ya en este caso a la hora de seleccionarla no aparece la animación original sino la(s) modificada(s) de ellas; luego se muestran los nuevos datos obtenidos con la modificación y se pasa a guardar la animación para esto se va al botón guardar (fig.32 [A]) y se busca la dirección, si esta es correcta se pasa al botón (fig.32 [B]) y se procede a exportar la animación.

Estas mismas operaciones se realizan para el otro formato ver (fig.33 [A] y [B]).

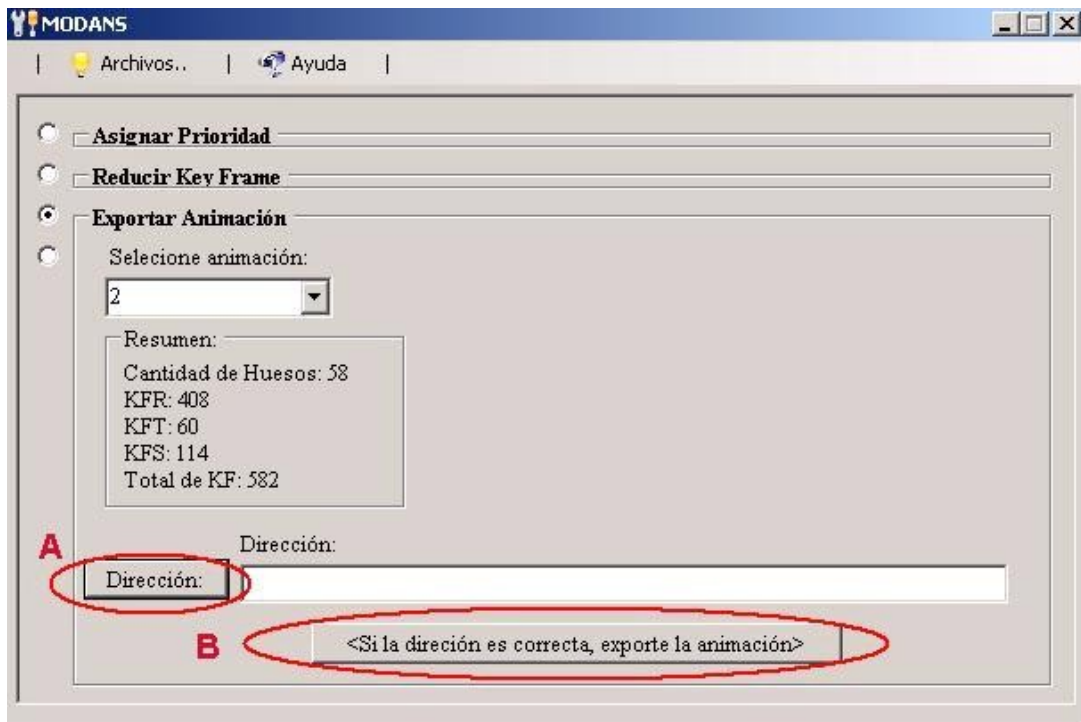


Fig. 32: Funcionalidad Exportar animación formato X.

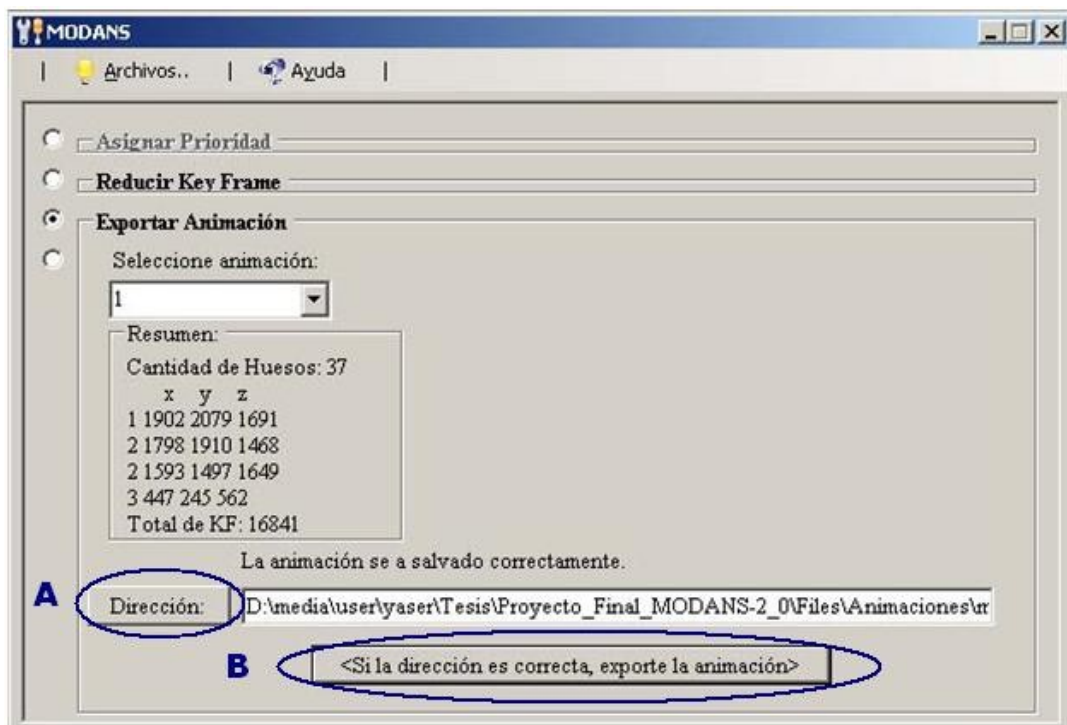


Fig. 33: Funcionalidad Exportar animación formato ANX.

3.8 Ayuda de la herramienta:

La herramienta MODANS necesita para su mejor funcionamiento, que la plataforma de su PC sea Windows, así como que tenga instalados los programas 3D Visual Studio 5 y DirectX; la aplicación es muy sencilla por lo que le resultará muy fácil trabajar con ella. Se comienza explicando el menú.

MODANS cuenta con 4 funcionalidades fundamentales:

- Cargar animación.
- Asignar prioridad.
- Reducir Keyframes.
- Exportar animación.

MODANS trabaja con dos tipos de formatos de animación:

- Direct.X
- ANX.

El menú muestra las funcionalidades que tiene la aplicación que pueden ser seleccionadas desde el menú,



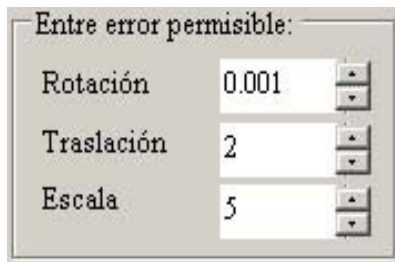
o simplemente activar los checkbox que las muestran también. Tener en cuenta que se debe comenzar activando el checkbox de cargar animación como se muestra posteriormente.



1. Lo primero que se hará, será seleccionar una animación; para esto se activa la funcionalidad *Cargar Animación* como se mostro anteriormente y se activa el botón cargar animación, la animación a cargar puede ser de cualquier formato .x o .anx.



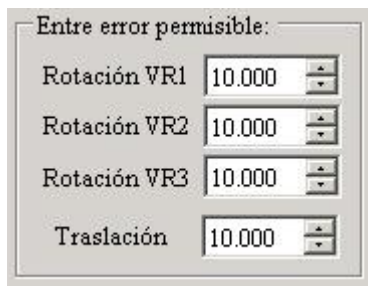
- 1.1. Si la animación tiene formato .x se habilitan las 4 funcionalidades de la herramienta no siendo asi para el formato .anx, ya que se deshabilita la funcionalidad *Asignar Prioridad*.
2. De esta forma se lee la animación y se le comunica si fue cargada con éxito o no, mediante un mensaje que se muestra.
3. Activar la funcionalidad *Reducir Keyframes* y como ya se había dicho puede ser mediante el menú o activando el checkbox correspondiente a la funcionalidad; de este modo pasamos a una nueva hoja de la aplicación y es aquí donde pasamos a modificar la animación.
4. Elegir de la lista la animación a modificar.
5. Se muestra una opción en la cual se entrará una serie de datos, la opción es *Entre error permisible*.
5. 1 Para una animación con extensión .x se entran los siguientes datos:
 - Rotación.
 - Traslación.
 - Escala.



Entre error permisible:	
Rotación	0.001
Traslación	2
Escala	5


5. 2 Para una animación con extensión .anx se entran los siguientes datos:

- Rotación.
- Rotación.
- Rotación
- Traslación.



Entre error permisible:	
Rotación VR1	10.000
Rotación VR2	10.000
Rotación VR3	10.000
Traslación	10.000

6. Activar el botón de Reducir Key Frame.



- Podrá observar en la parte derecha superior los datos originales de la animación.
- Podrá observar en la parte derecha inferior los datos de la animación una vez que se activo el botón.

7. Si se desea puede volver a modificar la nueva animación obtenida que se mostrara en la lista ver paso número 4, y repetir los pasos 5 y 6.

7.1. Activar la funcionalidad Asignar Prioridad y como ya se había dicho puede ser mediante el menú o activando el checkbox correspondiente a la funcionalidad; de este modo pasamos a una nueva hoja de la aplicación. Esta funcionalidad puede o no ser activada a la hora de modificar una animación, en caso de utilizarse activar la funcionalidad antes que la funcionalidad Reducir Keyframes.

- Cada hueso dentro de una animación tiene menor o mayor importancia en el movimiento que este efectúa.

7.2. Asignarle prioridad a los huesos de la animación.

7.3. Seleccionar el hueso(s) a priorizar.

7.4. La prioridad se da entre 0 y 255 o activar el checkbox para bajo, media o alto.

7.5. Activar el botón Asignar Prioridad. 

8. Por último activar la funcionalidad *Exportar animación* por menú o checkbox.

8.1. Elegir de la lista la animación a exportar.

8.2. Activar el botón Dirección para escoger el lugar a guardar la animación 

8.3. Activar el botón “si la dirección es correcta, exporte animación”



CAPÍTULO 4: DISEÑO DEL SISTEMA

En este capítulo se presenta el modelado del sistema, así como las clases que permiten dar solución a los requisitos funcionales como no funcionales que se definieron anteriormente. Se tiene como entrada el resultado obtenido en el modelo de análisis y se obtiene la estructura del sistema muy importante en el desarrollo del software.

Además se definen los diagramas de interacción (diagrama de secuencia) con que cuenta cada caso de uso, el paquete de clases que presenta el diseño del sistema y su composición con más exactitud de atributos y métodos que se implementaron para dar funcionalidad a la herramienta según cada caso de uso lo requirió.

4.1 Diagrama de clases del diseño:

El diagrama de clases es una parte importante dentro del desarrollo de un proyecto de software, ya que en él quedan definidas las clases que darán al sistema la funcionalidad que el usuario desea, obteniendo de esta forma una visión de todo lo que lo conforma y se ve más claramente como interactúan las clases que se definieron entre sí.

4.1.1 Descripción de las clases:

Nombre: cManager	
Tipo de clase: Controladora	
Atributo	Tipo
pAnimationANX	lvector<anxAnimacion*>
pEditedAnimationANX	lvector<anxAnimacion*>
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:	bool LoadAnimationANX(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 que cargue la animación.
Nombre:	bool AddAnimationANX(anxAnimacion* pAnimation)
Descripción:	Adiciona una nueva animación.
Nombre:	bool ExportAnimationANX(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:	bool RomoveKeyFrames(Uint IndexAnimation, Uint IndexOptimization, float ErrorR, float ErrorT)

Descripción:	Eliminar los Key Frames de la animación especificada a través IndexAnimation, a través de uno 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:	string GetAnimationDataANX(bool TypeUint, Uint iAnimation)
Descripción:	Obtiene la información de formato, la cantidad de huesos y de Key Frame de rotación y traslación de la animación.
Nombre:	void AddNewEditedAnimationANX(anxAnimacion* pAnimation)
Descripción:	Adiciona la nueva animación modificada.

Nombre: anxStorage	
Tipo de clase: Controladora	
Para cada responsabilidad:	
Nombre:	bool LoadAnimation(string LAFileName, cManager* pManager)
Descripción:	Mantener común para todos los posibles storages que hereden de ella.
Nombre:	bool ExportAnimation(cManager* pManager, Uint iAnimation, string EAFileName)
Descripción:	Mantener común para todos los posibles storages que hereden de ella.
Nombre:	int BinarioToInt(char arg[8])
Descripción:	Convierte a número binario la cadena de caracteres que se le pasa.

Nombre: anxRemoveFrames	
Tipo de clase: Controladora	
Para cada responsabilidad:	
Nombre:	anxAnimacion* RomoveUnnededKeyFrames(anxAnimacion* pAnimation, float ErroR, float ErroT)
Descripción:	Elimina de la animación los Key Frames teniendo en cuenta los márgenes de

	errores ErroR, ErroT.
--	-----------------------

Nombre: anxAnimacion	
Tipo de clase: Entidad	
Para cada responsabilidad:	
Nombre:	anxBoneCount
Descripción:	Devuelve la cantidad total de nodos (huesos) que contiene la animación.
Nombre:	anxFrameCount
Descripción:	Devuelve la cantidad total frames que contiene la animación.
Nombre:	vector<anxBone *> list_Nodos
Descripción:	Devuelve una lista de nodos (huesos) de la animación.
Nombre:	char anxFileID[5]
Descripción:	Inicialización de un arreglo de caracteres predeterminado.
Nombre:	int CantEV(int elem)
Descripción:	Devuelve la cantidad de vectores de el elemento que se le paso.

Nombre: anxBone	
Tipo de clase: Entidad	
Atributo	Tipo
v_R1XList	vector<float >
v_R1YList	vector<float >
v_R1ZList	vector<float >
v_R2XList	vector<float >
v_R2YList	vector<float >
v_R2ZList	vector<float >
v_R3XList	vector<float >
v_R3YList	vector<float >
v_R3ZList	vector<float >
v_vtXList	vector<float >
v_vtYList	vector<float >
v_vtZList	vector<float >

v_vtXList	vector<float >
v_R1XListByte	vector< bool >
v_R1YListByte	vector< bool >
v_R1ZListByte	vector< bool >
v_R2XListByte	vector< bool >
v_R2YListByte	vector< bool >
v_R2ZListByte	vector< bool >
vector<bool> v_R3XListByte	vector< bool >
vector<bool> v_R3YListByte	vector< bool >
vector<bool> v_R3ZListByte	vector< bool >
vector<bool> v_vtXListByte	vector< bool >
vector<bool> v_vtXListByte	vector< bool >
vector<bool> v_vtYListByte	vector< bool >
Para cada responsabilidad:	
Nombre:	Permite manejar los valores de los atributos.

Nombre: Visualizador	
Tipo de clase: Controladora	
Para cada responsabilidad:	
Nombre:	void OnCreate ()
Descripción:	Crea el entorno donde va a visualizarse la animación.
Nombre:	void OnShowMessages ()
Descripción:	Visualiza mensajes en el entorno de la animación.
Nombre:	void OnKeyDown (unsigned char arg_ucKey, unsigned int arg_uiModifiers)
Descripción:	Indicar cual animacion cargar para que sea visualizada.

4.2: Diagrama de clases del diseño por paquetes.

El diagrama de clases del diseño por paquetes permite un mejor desarrollo del software a implementar de este modo las clases ya que se simplifica y se hace más manejable el trabajo; los paquetes se dividieron tres, en la fig.27 se muestra una representación visual del mismo.

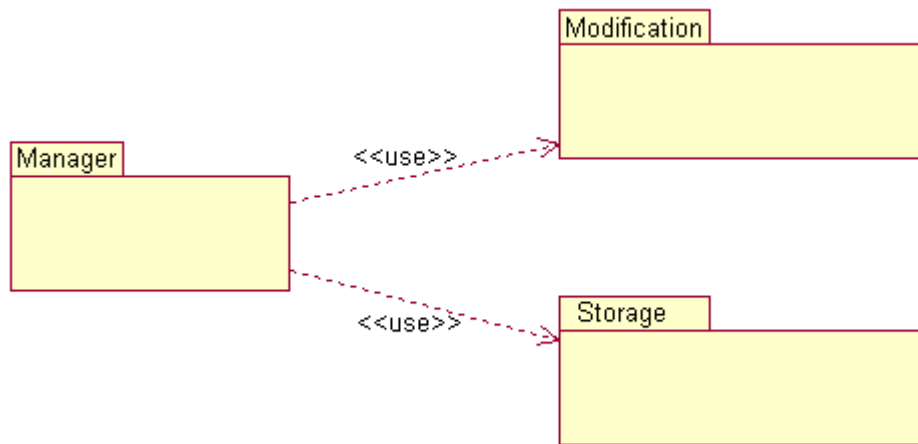


Fig. 34: Diagrama de clases del diseño por paquetes.

Los paquetes que contiene el sistema son: el Manager, el Modification y el de Storage. Cada paquete esta formado por clases; el paquete Manager tiene las principales clases con la que cuenta el sistema, el paquete Modification tiene solo una clase para modificar la animación y por ultimo el de Storage que costa de todos los distintos storages con los que se esta trabajando, particularmente con el anxStorage.

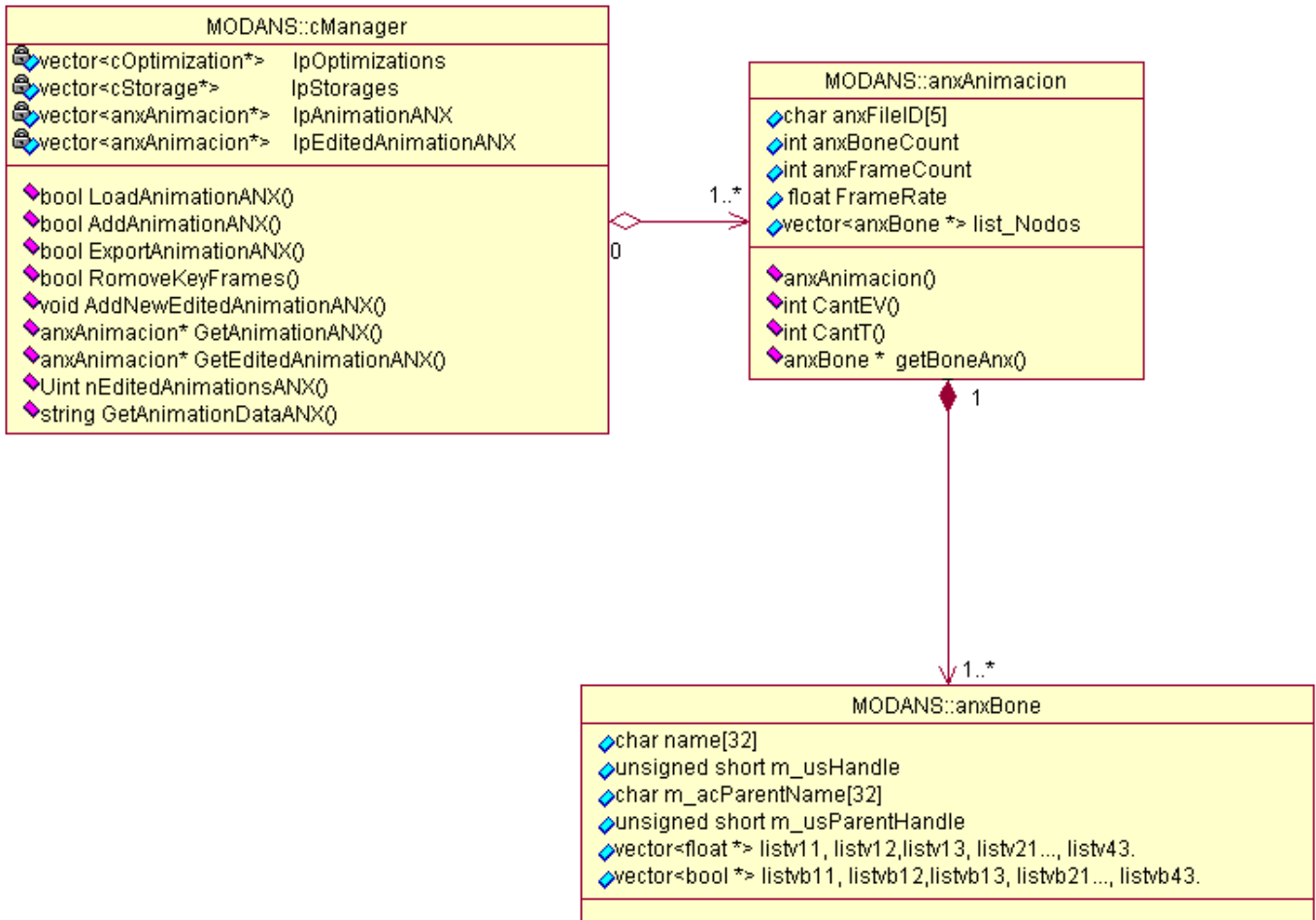


Fig. 35: Paquete Manager

La figura anterior muestra el paquete Manager que conforma al sistema, a partir de él se elaboró el diseño de la aplicación. La principal clase, esta dada por cManager y esta compuesta por un arreglo de storages que son las distintas animaciones que se cargan en la aplicación; la cual se relaciona a través de una agregación con la clase anxAnimacion, quién a su vez esta compuesta por la clase anxBone.

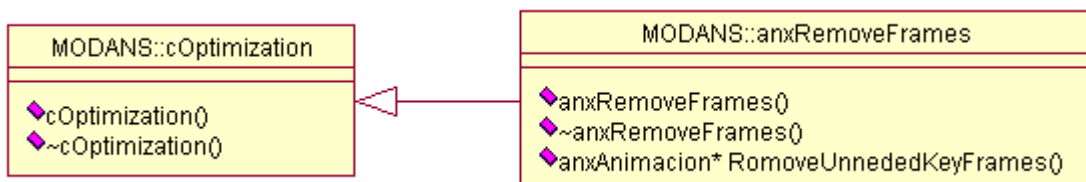


Fig. 36: Paquete Modifications

Este paquete está compuesto por la clase `anxRemoveFrames`, donde se puede ver una relación de herencia con la clase `cOptimization` la cual permite que se agreguen modificaciones sin perder la información básica que se tiene, de este modo la herramienta va adquiriendo nuevas formas de modificación según se requieran para algún proyecto en específico y así poder dar una eficiente solución.

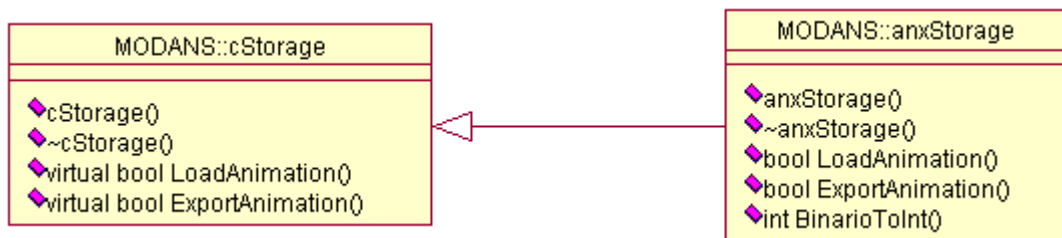


Fig. 37: Paquete Storage

En este paquete representado en la fig.37 anteriormente, también se puede ver una relación jerárquica por lo que esta además decir todas las ventajas que trae consigo; la clase hija está dada por `anxStorage` quien está diseñada para trabajar con animaciones de formato `anx`. Esto demuestra que a medida que haga falta se va a ir creando nuevas clases para así poder trabajar con distintos formatos a medida que estos sean necesarios o vayan apareciendo a lo largo de la implementación de nuevos editores de animación.

4.3: Descripción de la implementación por Casos de Uso.

Al igual que en el capítulo anterior aquí se trabaja con diagramas de interacción para modelar el diseño de cada caso de uso que presenta el diseño general del sistema, permitiendo de una forma explícita ver la interacción que se da entre las distintas clases así como los métodos que se utilizan en ellas.

4.3.1: Caso de Uso Cargar Animación.

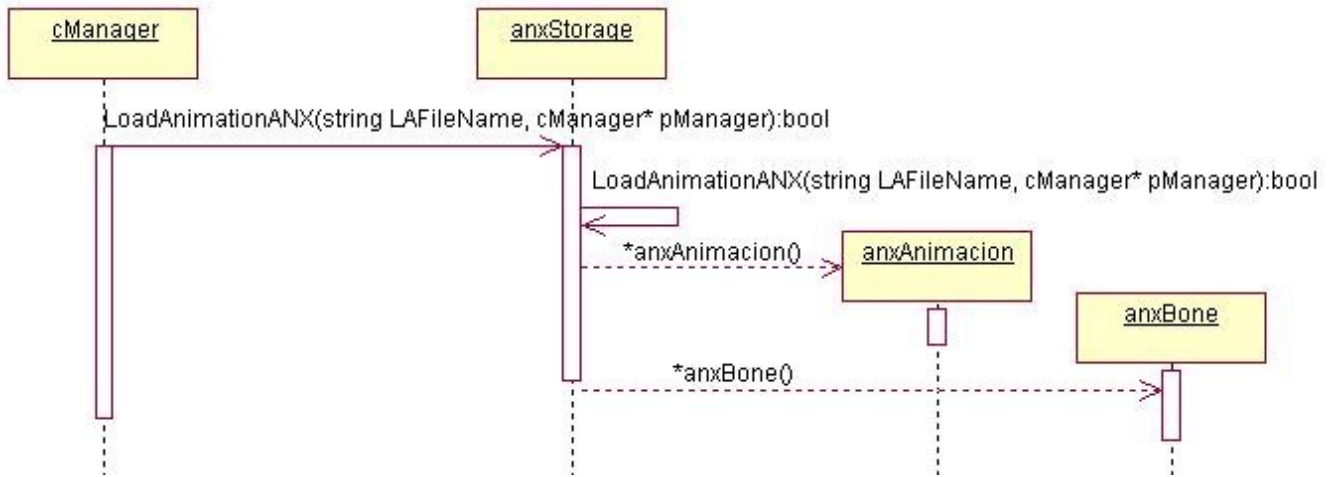


Fig. 38: Diagrama de secuencia Cargar Animación.

En el diagrama de secuencia cargar animación lo primero que la clase *cManager* es hacer la petición a la clase *anxStorage* de cargar la animación, la cual lo realiza utilizando el método *LoadAnimation*, quien se encarga de hacer la operación comunicada. Luego *anxStorage* se encarga de darle seguimiento al proceso con el método *LoadAnimation*, que a través de funciones definidas con anterioridad por *anx*, carga el fichero indicado por el Artista, donde ya con estos datos pasa a crearse la animación.

Este método se encarga además de crear la animación; de crear todos los huesos que contiene la animación para esto hace uso del método *anxBone*, donde cada hueso es rellenado con los Key Frames que debe contener la animación y así se le pasa a asignar el hueso a la animación.

4.3.2: Caso de Uso Reducir KeyFrame.

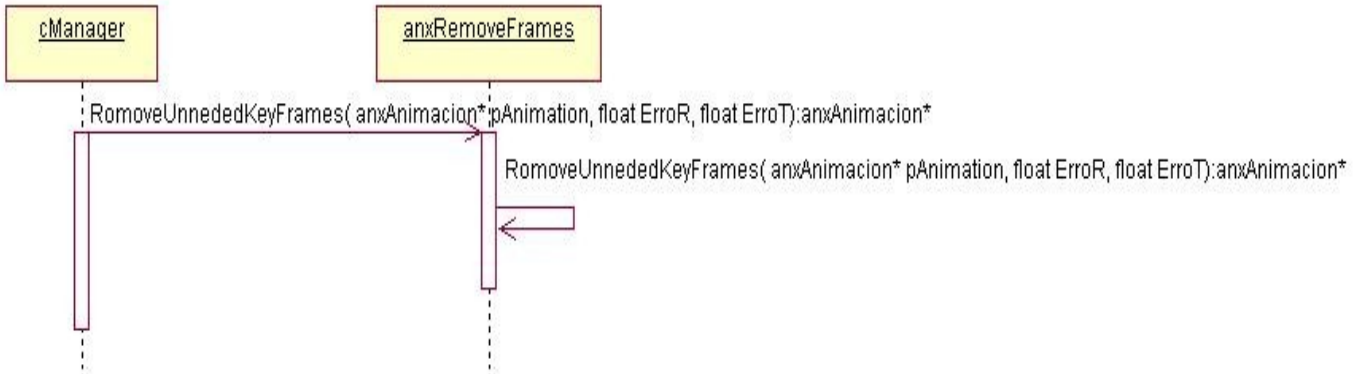


Fig. 39: Diagrama de secuencia Reducir KeyFrame.

La fig.30 muestra como se realiza el caso de uso reducir Reducir KeyFrame. Este comienza con la petición que hace la clase cManager a la clase anxRemoveFrames para reducir los key frames existentes en una determinada animación; para esto la clase anxRemoveFrames utiliza el método *RomoveUnnededKeyFrames* a la cual se le entra como dato la animación y el error permisible para la traslación y rotación de la animación.

La reducción de key Frame de la animación es llevada a cabo mediante un algoritmo matemático de interpolacion donde es calculado los Key Frames intermedios y se determina cual de ellos pude ser eliminado sin afectar los movimientos que tiene la animación.

4.3.3: Caso de Uso Exportar Animación.

La siguiente figura muestra el diagrama de secuencia para Exportar Animación, para esto la clase cManager llama al método *ExportAnimationANX*, el cual comienza una búsqueda por todo el arreglo polimórfico que contiene en busca de un storage compatible con el de la animación a exportar, en este caso el anxStorage pasaría a ser seleccionado dando así respuesta a la llamada realizada por cManager. La clase anxStorage se encargará de realizar la llamada al método *ExportAnimationANX* donde este da respuesta a la operación que se realizó, con el formato adecuado que el usuario específico al cargar la animación.

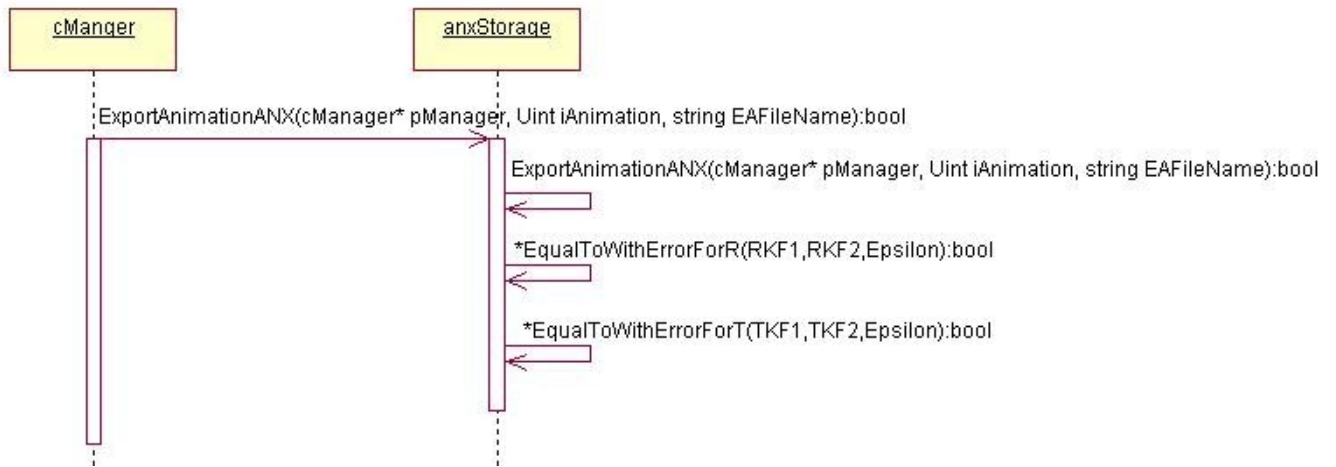


Fig. 40: Diagrama de secuencia Exportar Animación.

4.4. Consideraciones de el capítulo:

Con el desarrollo de este capítulo se ha elaborado la guía de la implementación que requería la aplicación, esto se logró con la realización del diseño del software, donde se especificó las clases necesarias, así como los métodos que dan solución al prototipo de la aplicación necesario para su utilización en el proyecto “Juegos CNEURO”.

CAPÍTULO 5: RESULTADOS

Este capítulo muestra los resultados obtenidos con las nuevas funcionalidades de la herramienta MODANS, al trabajar con el formato .anx. En el se muestran varias comparaciones entre la animación original y la resultante después de aplicarle el modificador; así se comprobó las potencialidades incorporadas a la herramienta para reducir los Key Frame y visualizar la cantidad de KB ahorrados.

Una vez terminada de implementar las nuevas funcionalidades de la herramienta, se establecieron tablas comparativas en cuanto a dos factores importantes, reducción en KB y lecturas de valores del fichero que definen un key frame, para esto se tomó una muestra aleatoria de 3 animaciones.

Las tablas muestran la animación original con la sumatoria total de KeyFrames de rotación y traslación que contienen la misma y estos datos llevados a la unidad de medida KB, igual para la animación obtenida mediante la modificación que se le aplica con la herramienta MODANS, para esto presentamos dos datos de reducción con diferentes entradas de márgenes de errores permisibles, uno para una mínima reducción de KeyFrames y el otro para el máximo permisible sin que la animación pierda las propiedades y calidad.

5.1: Análisis de reducción de KeyFrame en diferentes animaciones.

BackWalkLoop		
Datos originales	Error permisible #1 (0.001)	Error permisible #2(10)
2923	1592	613
2923	950	392
Totales	Totales	Totales
35076	15862	6394
140 KB	69 KB	32KB

Bess		
Datos originales	Error permisible #1 (0.001)	Error permisible #2(10)
11792	6346	3148
11792	4673	1661
Totales	Totales	Totales
140748	64701	30728
553 KB	273 KB	140 KB

HandPlant		
Datos originales	Error permisible #1 (0.001)	Error permisible #2(10)
3996	2164	1115
3996	1709	542
Totales	Totales	Totales
47952	22407	10466
190 KB	97 KB	50 KB

Después de establecer las comparaciones vistas anteriormente se puede decir que la aplicación cumplió con el objetivo trazado; ya que los resultados obtenidos son satisfactorios, estos se pueden ver mas claramente mediante la graficación de estos valores, fig.41 .

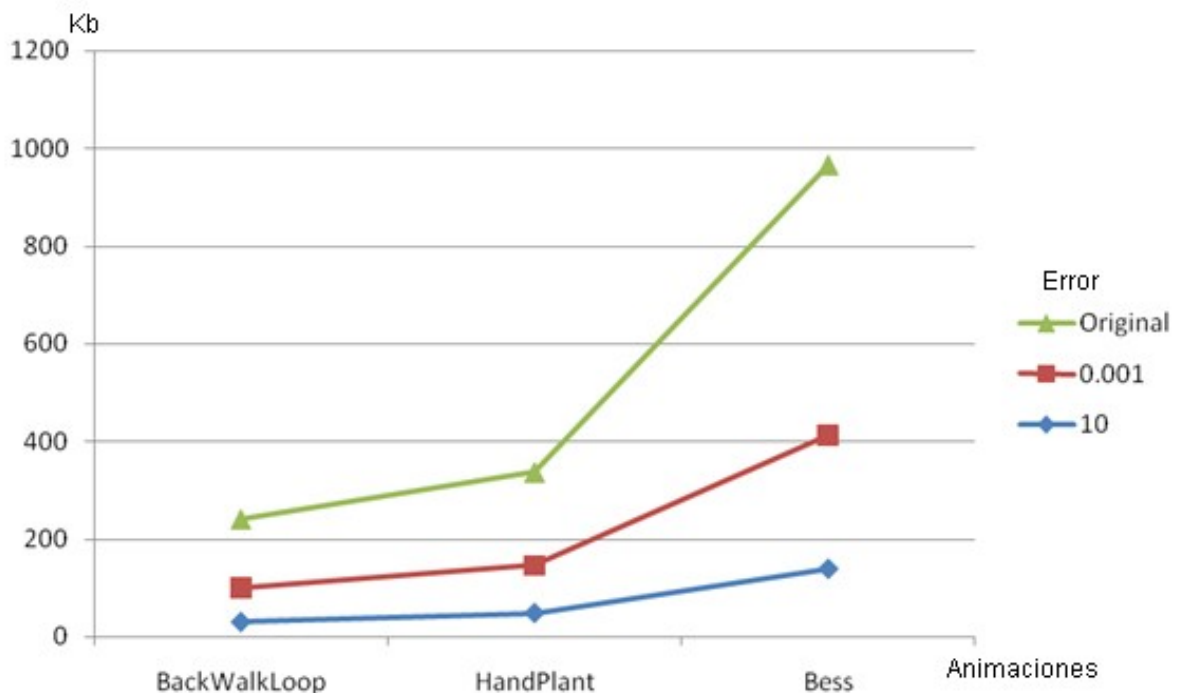


Fig. 41: Grafica comparativa de las 3 animaciones en cuanto al peso en KB.

Se aprecia claramente los valores iniciales con los que cuenta cada animación, y como, al introducir distintos valores de error permisibles, se va reduciendo considerablemente el peso de la animación, a medida que se introducen estos valores mas grandes.

5.2: Análisis de lectura del fichero desde la matriz que define el movimiento.

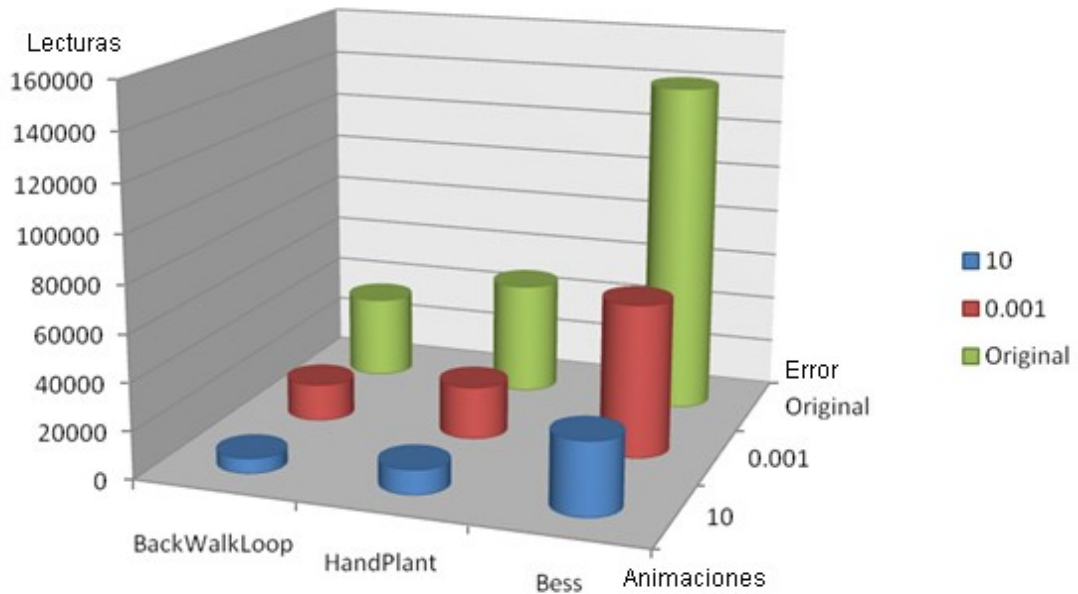


Fig. 42: Tabla de comparación en cuanto a lecturas del fichero para la construcción de las matrices que definen el movimiento de cada hueso de la animación.

Ejemplo de cómo también se reduce el consumo de memoria a la hora de cargar las animaciones al visualizador, pues al realizar esta operación se hace una lectura del fichero y si los valores restantes son parecidos sustituyéndolos por el anterior, completamos los restantes valores que definen a los demás key frames.

2.3: Consideraciones del capítulo.

Se puede decir que MODANS es una herramienta de uso factible para el proyecto “Juegos CNEURO”; donde las animaciones analizadas en este capítulo son una muestra de la eficiencia con que cuenta la herramienta actualmente.

CONCLUSIONES

La aplicación MODANS analiza el formato anx, lo cual posibilita cargar animaciones, reducir considerablemente la cantidad de Key Frames sin que la animación pierda sus propiedades de movimiento, erradicándose el uso excesivo de memoria.

MODANS permite nuevos formatos de modificaciones lo cual posibilita un mejor ajuste a las necesidades de los proyectos o grupos de desarrollo que trabajen con animaciones de huesos¹.

Con esta aplicación, CNEURO cuenta con una herramienta útil para el desarrollo de aplicaciones genéricas necesarias en cada uno de los proyectos a desarrollar.

RECOMENDACIONES

1. Aplicar la herramienta MODANS, en el proyecto CNEURO para sus juegos u otros proyectos; así como simuladores virtuales, pues las ventajas que ofrece respecto al ahorro son favorables.
2. Continuar la implementación de la herramienta para cualquier modificador o almacenador que se necesite para proyectos o grupos de desarrollo que utilicen animaciones de hueso.
3. Aplicar el método matemático de Interpolación mas óptimo, para proyectos donde sus clientes cuenten con maquinas con mejores prestaciones.

BIBLIOGRAFÍA

Documentación

1. -5, P. P. (2008, Marzo). SceneToolKit. Ciudad de la habana, Cuba.

Fuente electrónica

2. -Argentino, I. (2006, Marzo 22). Seminario Reuters sobre cálculo financiero. Retrieved from <http://www.iamc.sba.com.ar/informes/ConfExp/CA2006/Presentaci%C3%B3n%20IAMC-Reuters.pdf>
3. -Blender, D. (2008, Mayo). Blender. Retrieved from www.blender.org
4. -Booch, Grady. 1996. Análisis y Diseño Orientado a Objetos. 2da edición. Ed. Addison-Wesley / Díaz de Santos.
5. -Electrónica (U:A:H), D. d. (n.d.). Operaciones Geométricas. Retrieved from <http://www.depeca.uah.es/docencia/IT-INF/sva/pr4-a.pdf>
6. -Mateos, G. G. (n.d.). Tema4.Transformaciones Geometricas. Retrieved from <http://dis.um.es/profesores/ginesgm>
7. -Pressman, Robert. 1998. Ingeniería de Software.
8. -Tratamiento digital de la imagen. (n.d.). Retrieved from http://campusvirtual.uma.es/tdi/www_netscape/TEMAS/Tdi_03/index6.php#4.2.-%20Interpolación%20bilineal%20

Tesis:

9. -(Minardo González, 2007)

Conferencia:

- 10.-ESPECIALIDAD, D. Flujo de trabajo Análisis y Diseño. (Modelo de diseño). Universidad de las Ciencias Informáticas, 2005.

11. -ESPECIALIDAD, D. Introducción a la Ingeniería de Software. Universidad de las Ciencias Informáticas, 2005.

ANEXOS

Fig. 43: Anexo1. Comparación de los métodos de Interpolación.

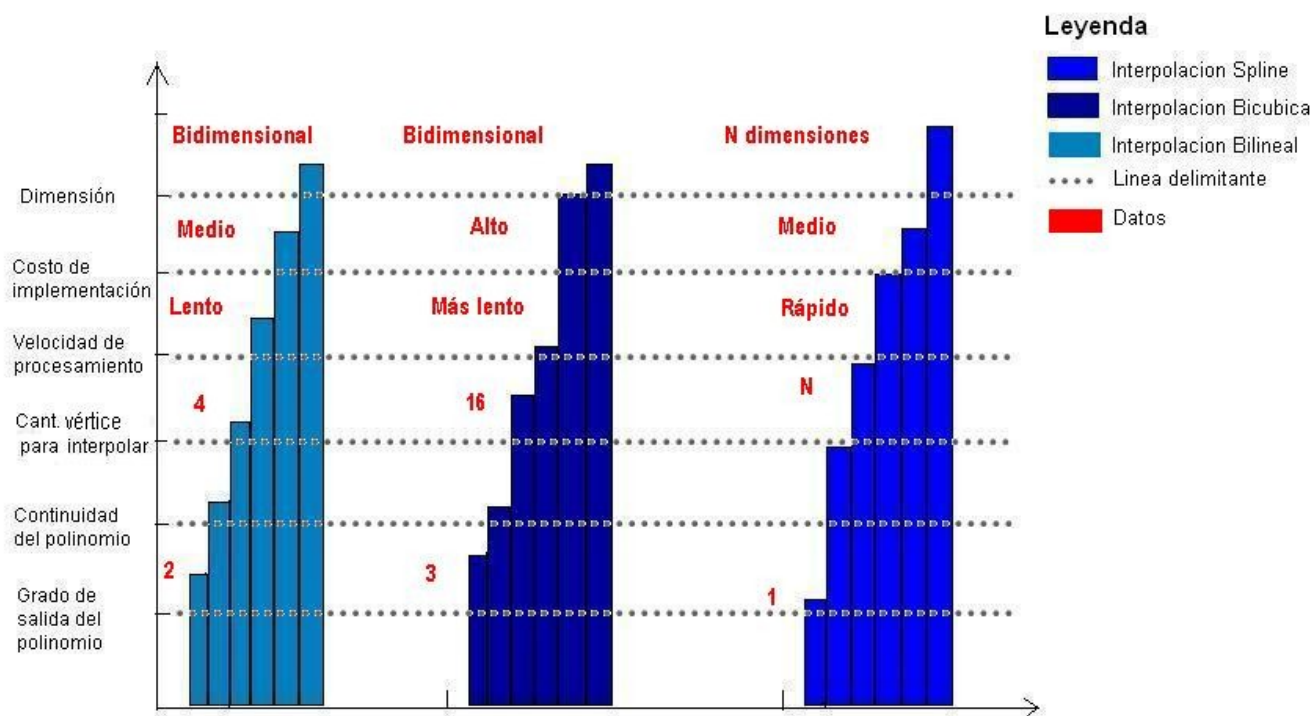
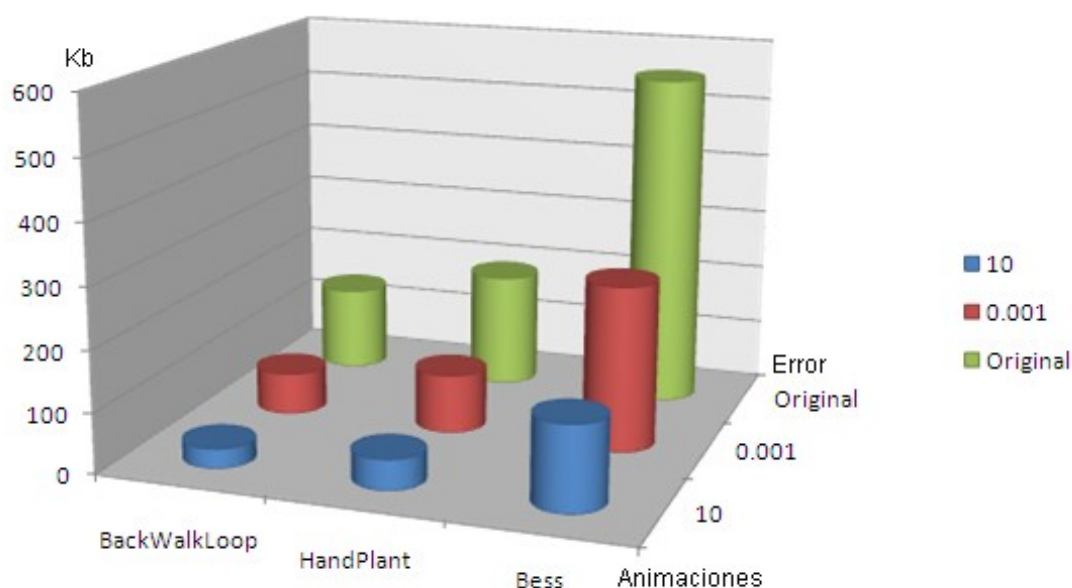


Fig. 44: Anexo2. Tabla de comparación en cuanto al peso del fichero en KB.



GLOSARIO DE TERMINOS

A

1. **Animaciones de huesos:** Son aquellas animaciones que en su constitución interna están formados por hueso, una jerarquía de huesos.

B

2. **Bípedo:** 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

3. **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.
4. **Cinemática inversa:** La cinemática se ocupa de la descripción del movimiento sin tener en cuenta sus causas. Su objetivo consiste en encontrar el gesto que deben adoptar las diferentes articulaciones para que el final del sistema articulado llegue a una posición concreta.

J

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

M

6. **Mallas poligonales:** Es la superficie creada mediante un método tridimensional generado por sistemas de vértices posicionados en un espacio virtual con datos de coordenadas propios.
7. **Memoria:** Se refiere a una forma de almacenamiento de estado sólido conocido como memoria de acceso aleatorio (RAM por sus siglas en inglés).

8. **Metaballs:** Es el nombre de una técnica de gráficos realizada por ordenador para simular interacción orgánica entre diferentes objetos n-dimensionales (como gotas de mercurio mezclándose por su superficie) y fue inventado por Jim Blinn a principios de los años 1980.

Cada metaball es definida en función de n-dimensiones (es decir para tres dimensiones, $f(x, y, z)$; los metaballs tridimensionales tienden a ser los más comunes). Un valor de umbral también es elegido, para definir un volumen sólido.

9. **Morphing:** 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.

N

10. **NURBS:** Acrónimo inglés de la expresión Non Uniform Rational B-Splines. Modelo matemático muy utilizado en los gráficos por ordenador para generar y representar curvas y superficies.

P

11. **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.

12. **Postura:** Posición que adopta un bípedo en un determinado momento de una animación.

13. **Primitiva geométrica:** Formas geométricas consideradas primitivas por su básica constitución en las partes que la conforman se conocen también con el nombre de primitivas geométricas cuyas formas son el Círculo, el Triángulo y el Cuadrado.

14. **Programa multiplataforma:** Es un término usado para referirse a los programas, que puedan funcionar en diversas plataformas. Por ejemplo, una aplicación multiplataforma podría ejecutarse en Windows, en GNU/Linux.

R

15. **Renderizado:** Es la acción de asignar y calcular todas las propiedades de un objeto antes de mostrarlo en pantalla.

V

16. **Vértice:** Es el punto común entre los lados consecutivos de una figura geométrica, o el punto común de los dos lados de un ángulo, o el punto en que concurren tres o más planos, o el punto de una curva en que la encuentra un eje suyo normal a ella.