



Universidad de las Ciencias Informáticas

Sistema de Captura de Movimiento para ambientes tridimensionales

Trabajo para optar por el título de Ingeniería Informática

Autor: Yulier Casas Estrada

Tutor: Ing. Fernando Jiménez López

Abril del 2006

Es intentando lo imposible como se realiza lo posible

Hemi Barbusse

Dedicatoria

A mi familia por todo su apoyo, a mis padres por ser mi guía y ejemplo, por ayudarme a superar todos los obstáculos que encontré en el camino, a mis hermanos por confiar en mi y gracias a los cuales trato de ser cada día mejor.

Agradecimientos

A todas las personas que de una forma u otra contribuyeron con el desarrollo de este trabajo. A Fabio, sin su ayuda no estuviera hoy aquí. A Frank por introducirme en el mundo de la realidad virtual, a Alexito por su ayuda incondicional, a mi tutor por transmitirme su experiencia. A mis compañeros de año, que fueron mi compañía y apoyo todo el tiempo.

Resumen

En el presente trabajo se brinda una nueva vía para el desarrollo de animaciones, específicamente para el proceso de captura de movimiento a partir de personajes reales. Esto se logró con el desarrollo de una herramienta capaz de llevar a cabo la captura de movimiento llevando el costo de este proceso a pequeñas cifras ya que solamente exige de una computadora, dos cámaras para la filmación de dos videos y un buen Mouse. El sistema que se propone es un sistema manual que reduce de manera muy significativa el tiempo de realización de una animación. Este sistema es algo muy útil ya que en Cuba no se ha avanzado mucho en esta materia por la carencia de recursos para obtener sistemas capaces de desarrollar animaciones con un costo bajo tanto en tiempo como económico. El objetivo principal del software desarrollado es brindar una vía factible a un país carente de recursos como Cuba, para de esta forma poder entrar en el importante mundo de los entornos virtuales, que aunque se han desarrollado algunos en Cuba, es vital lograr una herramienta de este tipo que sea capaz de disminuir significativamente el tiempo de desarrollo de estos entornos ya que con la propuesta que se trae en este trabajo se han logrado muy buenos resultados, los cuales son más significativos en el bajísimo costo económico a la hora de realizar una animación, así como también en el enorme ahorro de tiempo.

Índice

INTRODUCCIÓN	1
CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA	4
1.1 INTRODUCCIÓN	4
1.2 ANTECEDENTES HISTÓRICOS DE LOS SISTEMAS DE CAPTURA DE MOVIMIENTO	5
1.3 ANIMACIONES	11
1.3.1 Principios básicos de la animación	11
1.3.2 Animación por ordenadores	16
1.3.3 Animación 3D por ordenadores	16
1.3.4 Técnicas de animación de personajes 3D	18
1.6 SISTEMA DE ANIMACIÓN POR ESQUELETO	19
1.7 JERARQUÍA DE PERSONAJES	20
1.7.1 Manipulación de la jerarquía	20
1.8 FORMATOS UTILIZADOS PARA LAS ANIMACIONES	21
1.9 ESTADO DEL ARTE EN LA ACTUALIDAD	22
1.9.2 Sistema óptico de captura de movimiento	23
1.9.2.1 Sistema pasivo de captura de movimiento en tiempo real	24
1.9.3 Sistema magnético de captura de movimiento	25
1.9.4 Sistema Electro – Mecánico de captura de movimiento	26
1.9.5 Captura facial	27
1.10 DESARROLLO DEL TEMA EN NUESTRO PAÍS Y NUESTRA UNIVERSIDAD	27
1.11 CONCLUSIONES	28
CAPÍTULO 2	29
2.1 INTRODUCCIÓN	29
2.2 PLANTEAMIENTO DEL PROBLEMA	29
2.3 OBJETO DE ESTUDIO	30
2.4 CAMPO DE ACCIÓN	30
2.5 DESCRIPCIÓN DE LA SOLUCIÓN	30
2.5.1 Filmación de los videos	31
2.5.2 Generalidades del sistema	32
2.5.2.1 Lenguaje utilizado	32
2.5.2.2 Escenas	33
2.5.2.3 Formatos relacionados	33
2.5.2.4 Esqueleto utilizado	37
2.5.2.5 Videos	37
2.5.2.6 Visualización de la animación	38
2.5.2.7 Salvas	38
2.5.3 Capturando los puntos	38
2.5.3.1 Continuo	39
2.5.3.2 Pose a Pose	39
2.5.4 Obtención del producto final	40
2.6 SISMA VS. MÉTODOS ACTUALES	42
2.7 MODELO DE DOMINIO	44
2.8 REQUISITOS FUNCIONALES	45

2.9 REQUISITOS NO FUNCIONALES	46
2.10 ESPECIFICACIÓN DE CASOS DE USO	47
2.11 EXPANSIÓN DE LOS CASOS DE USO	51
2.12 CONCLUSIONES	66
3. ANÁLISIS Y DISEÑO	68
3.1 INTRODUCCIÓN	68
3.2 DIAGRAMAS DE CLASES	68
3.2.1 Diagrama de clases del paquete Dominio.....	69
3.2.2 Diagrama de clases del paquete Presentación	69
3.2.3 Diagrama de clases del paquete CBuilder.....	69
3.2.4 Diagrama de clases del paquete MathLibrary.....	69
3.2.5 Diagrama de clases del paquete ASEFileLoader.....	70
3.3 DIAGRAMA DE PAQUETES	70
3.4 DESCRIPCIÓN DE LAS CLASES	71
3.5 DIAGRAMAS DE SECUENCIA	78
3.5.1 Cargar escena	78
3.5.2 Salvar escena	79
3.5.3 Capturar puntos	79
3.5.4 Crear nueva animación.....	80
3.5.5 Exportar animación a csm	80
3.5.6 Cargar esqueleto.....	81
3.5.7 Cargar videos.....	81
3.5.7 Seleccionar animación.....	81
3.5.8 Borrar animación.....	82
3.5.9 Seleccionar hueso.....	82
3.5.10 Otros casos de uso	83
3.5.10.1 Flujo de sucesos de Variar offset	83
3.5.10.2 Flujo de sucesos de Manipular video	83
3.5.10.3 Flujo de sucesos de Variar velocidad del video	84
3.6 CONCLUSIONES	84
4. IMPLEMENTACIÓN Y PRUEBA.....	85
4.1 INTRODUCCIÓN	85
4.2 DIAGRAMAS DE COMPONENTES	85
4.2.1 Dominio.....	86
4.2.2 Presentación.....	87
4.2.3 MathLibrary.....	87
4.2.4 ASEFileLoader.....	88
4.3 DESCRIPCIÓN DE LOS CASOS DE PRUEBA	88
4.4 RESULTADOS OBTENIDOS.....	93
4.4.1 Resultados de la captura de movimiento.....	93
4.4.2 Resultados del producto final.....	95
4.5 CONCLUSIONES	97
CONCLUSIONES GENERALES.....	98
RECOMENDACIONES.....	99
REFERENCIAS BIBLIOGRÁFICAS	100
BIBLIOGRAFÍA CONSULTADA.....	104

GLOSARIO DE TÉRMINOS	105
ANEXOS	109
ANEXO 1: JERARQUÍA DE UN ESQUELETO	109
ANEXO 2: SISTEMAS DE CAPTURA DE MOVIMIENTO.....	110
ANEXO 3: DIAGRAMA DE CASOS DE USO.....	113
ANEXO 4: DIAGRAMAS DE CLASES	114
ANEXO 5: DIAGRAMAS DE SECUENCIA	117

Introducción

Desde el surgimiento de la Realidad Virtual como disciplina dentro de la Ciencia de la Computación, que trata de simular el mundo real y todas las interacciones que ocurren entre los objetos y seres humanos que viven en este mundo, intentar darle movimiento realista a las animaciones tridimensionales generadas por computadora de los mundos virtuales, ha sido una tarea muy difícil para aquellos que han incursionado en este campo. En la actualidad, gracias a los avances tecnológicos sucedidos es que esta tarea se ha podido acometer con más eficiencia y se han desarrollado poderosos sistemas de captura de movimiento que permiten generar animaciones tridimensionales con movimientos reales. Es incontable la cantidad de problemas computacionales en los que las técnicas de Captura de Movimiento (MOCA en lo adelante) pueden brindar una solución eficiente y con calidad.

Uno de los campos en los que las técnicas MOCA son aplicadas con mucha frecuencia es en el de los videojuegos, y de forma más general en todas las aplicaciones profesionales donde se utilice la animación tridimensional, lo mismo para simular seres humanos que objetos del mundo real.

En la actualidad las técnicas más novedosas y probablemente más atractivas están basadas en métodos ópticos. Estos sistemas utilizan cámaras sofisticadas para registrar datos precisos de los movimientos tomados de trajes reflectores que son usados por el actor virtual, esto da como resultado datos de salida extremadamente limpios. Ya con estos datos se utiliza un software de MOCA para trasladar el movimiento a un esqueleto virtual y después se importan estos datos a un programa de animación profesional y se crea el personaje animado.

Sin embargo a pesar de que los beneficios de la tecnología son claros, tiene un inconveniente importante, por lo menos para nuestro país, y es su elevado costo, de ahí la necesidad de un sistema que sustituya a los existentes en el mundo, que reduzca lo más posible el costo del

proceso de captura de movimiento para que pueda ser utilizado en Cuba, logrando de esta forma un mayor avance en esta área. En Cuba se ha incursionado poco en este campo, debido a que requiere de una gran cantidad de recursos y conocimiento.

Este trabajo propone como objetivo general, desarrollar una herramienta para realizar captura de movimiento que sea capaz de mejorar tanto el tiempo de desarrollo de las animaciones para entornos virtuales como su calidad. Lográndose además, un sistema con un costo menor a sus similares en el mundo.

Los objetivos específicos que se persiguen son: lograr un método adecuado para capturar el movimiento realizado por un actor virtual, empleando métodos eficientes de cálculo de su posición espacial en cada momento, y exportar un fichero de dicha animación que pueda ser importado por softwares especializados para el desarrollo de animaciones.

Durante el desarrollo de este software se realizaron una serie de tareas para cumplir con los objetivos propuestos. Se hizo un estudio de los sistemas existentes en el mundo que se dedican al campo de las animaciones y en específico al de la captura de movimiento. Se analizaron las herramientas que podían servir para generar la animación utilizando los datos de la captura de movimiento, en este caso se seleccionó el 3Dmax, y posteriormente se pasó al desarrollo de la aplicación para una vez terminada esta llevar a cabo numerosas pruebas de la misma.

El trabajo está dividido en cuatro capítulos, un primer capítulo donde se da una breve explicación sobre la evolución de este tipo de sistemas a través de los años. Se detallan cada uno de los sistemas utilizados actualmente, así como las técnicas que se han utilizado para la solución del problema. Luego hay un segundo capítulo en el que se plantea la situación problemática, así como también la propuesta de sistema que se hace para resolver esta situación, también se enumeran los requisitos funcionales y no funcionales del sistema entre otras cosas importantes para la comprensión de la propuesta. En el tercer capítulo se muestra el análisis y diseño del

sistema, en él está todo lo referente a los diagramas que dan una buena idea de cómo funciona el sistema visto desde dentro, para esto se utilizaron diagramas de clases, de secuencia, de casos de uso, entre otros que son necesarios para la comprensión del software. Por último se verá el capítulo cuatro que es el de implementación y prueba, ahí se podrá ver la veracidad de los resultados obtenidos a través de análisis comparativos con otros métodos existentes.

Capítulo 1 Fundamentación teórica

1.1 Introducción

Largo ha sido el camino en el desarrollo de la realidad virtual, y muy en especial en el caso de los sistemas de captura de movimiento, alcanzándose en la actualidad un gran desarrollo de los mismos, aunque aún existen muchos aspectos en los que se puede mejorar. Para lograr esto se trabaja en el perfeccionamiento de las herramientas que se utilizan para estos fines así como también en las técnicas que se conocen hasta este momento.

En la actualidad los sistemas de captura de movimiento han alcanzado un gran desarrollo, ya que hay muchas compañías que se han dedicado por años al desarrollo de animaciones cada vez más reales y complejas para poder insertarse en el mercado de los entornos virtuales, principalmente de los video juegos. A las ya mencionadas, que llevan varios años en este campo, se han unido otras con grandes potencialidades y recursos para triunfar en esta materia.

Algo que ha impulsado de manera significativa el desarrollo de entornos virtuales son los grandes avances que ha habido en hardware y software, además de la reducción de los precios de los mismos, ya que esto ha posibilitado que una mayor cantidad de usuarios pueda acceder a estos recursos y desarrollarse en este aspecto. Hoy en día es muy usual encontrar en entornos domésticos todos los elementos necesarios para construir y ejecutar entornos virtuales de gran calidad que antes podían ser considerados impresionantes aplicaciones virtuales, increíbles videojuegos o espectaculares simuladores. Todo lo antes mencionado nos da una medida de lo prometedor que es el campo de los gráficos computarizados en este momento, con las eficientes herramientas que se han desarrollado, esto ha hecho que sean muchas más las personas que se dediquen a esto lo que indiscutiblemente trae consigo un aumento de la calidad de las aplicaciones virtuales y como es lógico un mayor desarrollo de estas.

Las técnicas MOCA han reducido el tiempo de desarrollo de las animaciones en gran medida así como su calidad ha mejorado de manera significativa. Existen varios métodos MOCA que se diferencian entre ellos por la forma de capturar los datos y procesarlos. Entre los más conocidos se pueden mencionar tres tecnologías, la Óptica, la Magnética y la Electro-Mecánica.

1.2 Antecedentes históricos de los sistemas de captura de movimiento

El uso de la captura de movimiento para animar personajes en computadoras es relativamente nuevo, tiene sus inicios a finales de los años 70, y desde hace unos años es que su uso se ha propagado verdaderamente. MOCA no es más que la grabación del movimiento de un cuerpo humano (u otro movimiento) para su posterior análisis. La información capturada puede ser una posición general, simple, la de un cuerpo en el espacio e incluso con deformaciones faciales o de las masas musculares. Este método consiste en el mapeado de un cuerpo humano para a partir de sus movimientos dar vida a un modelo computacional, es decir, realizar su animación.[1]

La idea de copiar los movimientos del cuerpo humano para realizar animaciones no es nueva. Para conseguir un movimiento convincente de personajes humanos en la computadora en Snow White, los estudios Disney lograron las animaciones a partir de un filme de actores vivos en todas las escenas. Este método es llamado rotoscoping. El mismo tuvo un uso satisfactorio desde esos tiempos. En los finales de los 70, comenzó a ser factible la animación por computadora debido a que se necesitaba para más soluciones y ya los animadores estaban adaptados a las técnicas tradicionales incluyendo rotoscoping. El rotoscoping puede ser considerada como una técnica primitiva o precursora de MOCA, donde el movimiento es capturado cuidadosamente a mano. [1]

1980 – 1983: Simon Fraser University – Goniómetros

Por este tiempo en los laboratorios de biomecánica comienzan a analizar el uso del movimiento humano en las computadoras. Las técnicas y dispositivos usados en estos estudios comienzan a marcar pautas dentro de la comunidad de los gráficos por computadora. A inicios de los 80 Tom Calvert , un profesor de quinesiología (ciencia que estudia la interrelación entre la anatomía del cuerpo humano y los procesos fisiológicos en lo que respecta al movimiento) y computación de Simon Fraser University acopló potenciómetros a un cuerpo humano y usó los resultados para manejar figuras animadas en computadora para estudios de coreografía y valoración de los movimientos abdominales. Estos potenciómetros los colocaba en lugares del cuerpo a los que necesitaba copiarles su movimiento como por ejemplo las rodillas que son fundamentales en el movimiento de las piernas. Los resultados que se obtienen por este método son analógicos, estos resultados fueron convertidos a su forma digital y con ellos se alimentó la animación en la computadora. Este sistema de animación conjuntamente con especificaciones cinemáticas y otros cálculos necesarios sirvieron para especificar completamente el movimiento de los personajes. [1]

1982 – 1983: MIC Marioneta Gráfica

Poco después los sistemas ópticos de rastreo tales como Op – Eye y SelSpot comenzaron a ser usados por la comunidad de los gráficos por computadora. A comienzos de los 80 Architecture Machine Group y el New York Institute of Technology Computer Graphics Lab experimentaron con el rastreo óptico del cuerpo humano. [1]

El rastreo óptico utiliza pequeños marcadores acoplados al cuerpo humano, estos marcadores son llamados LEDs y no son más que pequeños reflectores de luz. También se colocan 2 o más cámaras enfocadas hacia el cuerpo que tiene estos reflectores. Una combinación especial de hardware y software utilizan los datos recogidos de los marcadores y las cámaras y por

comparación de imágenes calculan las tres dimensiones para cada posición que alcanza cada marcador en el tiempo. [1]

Esta tecnología se ve afectada por la velocidad con que puede ser examinado cada marcador, por la oclusión de estos marcadores debido al cuerpo y por la resolución de las cámaras. A consecuencia de estas limitaciones y otras casi todos los sistemas ópticos en la actualidad se apoyan mucho en las técnicas de análisis post – procesamiento, procesos y la limpieza de los datos antes de ser aplicados a los personajes en la computadora. [1]

En 1983 Ginsberg y Maxwell en MTT presentaron la Marioneta Gráfica. El sistema usado en el comienzo de los sistemas ópticos llamado Op – Eye que se apoyó, como bien se dijo anteriormente, en los LEDs. Se ajustaba al cuerpo un traje que es como un alambrado con LEDs en las articulaciones y otras zonas anatómicas de importancia. 2 cámaras captaban la posición del cuerpo desde diferentes campos visuales, la imagen captada está en 2D(en el plano). La computadora utilizando la información recopilada tenía como tarea llevar todos estos datos a un mundo 3D(en el espacio). Lo demorado del render en esta tecnología y lo caro del hardware que se necesita provocó que su uso no se extendiera todo lo que se hubiera querido. [1]

1988: deGraf/Wahrman – Mike the Talking Head

En 1988 deGraf/Wahrman desarrolló “Mike the Talking Head” para Silicon Graphics mostró las capacidades en tiempo real de sus nuevas máquinas 4D(cuatro dimensiones). Un micrófono que fue conducido por un controlador construido especialmente que permitió controlar muchos parámetros del rostro de los personajes, incluyendo boca, ojos, la expresión y la posición de la cabeza. El hardware Silicon Graphics suministró una interpolación en tiempo real entre las expresiones del rostro y la geometría de la cabeza controlado por un artista. Quedó claramente demostrado que esta tecnología era madura para su explotación en ambientes de producción. [1]

1988: Pacific Data Images – Waldo C. Graphic

A principios de 1985, Jim Henson Productions había estado tratando de crear gráficos por computadora para sus personajes. Tuvieron poco éxito debido principalmente a las pocas capacidades de su tecnología en aquel entonces. Finalmente en 1988 con la disponibilidad de Silicon Graphics 4D en estaciones de trabajo y con la profesionalidad de Pacific Data Images encontraron una solución viable. Enganchando un dispositivo de 8 grados de libertad a través del SGI dial box (especie de marcas o máscara que se coloca en el rostro) estándar, ellos pudieron controlar la posición y los movimientos de la boca para un personaje de baja resolución en tiempo real. De esa manera nació Waldo C. Graphic. Waldo se hizo fuerte como un generador de muñecos en computadora que pudieron ser combinados con en tiempo real con modelos reales. Las imágenes de la computadora fueron combinadas con videos de las cámaras enfocadas a los modelos reales a fin de que todas puedan ejecutarse conjuntamente. Seguidamente en post producción, el resultado es sendereado a resolución completa adicionándosele unos pocos elementos encima del movimiento ejecutado. [1]

1989: Kleiser – Walczak – Dozo

En 1989 Kleiser – Walczak produjo Dozo, una animación en computadora de una mujer bailando en frente de un micrófono mientras cantaba una canción para un video musical. Para conseguir los movimientos humanos reales ellos decidieron usar técnicas de captura de movimiento. Basados en experimentos de captura de movimiento del trabajo de Kleiser en Digital Production y Ómnibus. Ellos utilizaron un método basado en el análisis del movimiento usando múltiples cámaras para triangular las imágenes de pequeñas piezas de una cinta reflectante colocada en el cuerpo. La salida resultante es la trayectoria 3D de cada reflector en el espacio. Tal como está descrito anteriormente uno de los problemas de esto es el rastreo de los puntos con las cámaras, que se hace un poco difícil. [1]

1991: Videosystem - Mat the Ghost

Teniendo en cuenta las posibilidades de animar personajes utilizando las técnicas mencionadas en Waldo C. Graphic, Videosistem , un productor francés de gráficos en computadora, giró la atención hacia su recién formada división de animaciones en computadora en el aspecto de los muñecos en computadora. Los resultados fueron un sistema de animación de personajes en tiempo real del cual su primer suceso fue la producción diaria de un personaje llamado Mat the Ghots. Mat fue un amigable fantasma verde que interactuaba con actores vivos y muñecos en un show que salía diariamente para los niños, llamado “Canaille Peluche”. Usando DataGloves, joysticks, rastreadores Pholemus, y pedales de tambor MIDI, titiriteros ejecutaban a Mat interactivamente. Desde entonces las secuencias de animaciones dejaron de construirse en el post render, ya que las animaciones fueron generadas en el mismo tiempo que necesita un artista para hacer una buena toma. Videosystem conocido como Medialab ha continuado su desarrollo convirtiéndose en una herramienta de producción confiable. Siempre cada personaje es controlado por muchos titiriteros o actores trabajando de conjunto.[1]

1992: SimGraphics – Mario

En el 1992 SimGraphics desarrolló un sistema de rastreo facial denominado “face waldo”. Usando sensores mecánicos acoplados a la barbilla, labios, mejilla y cejas, y sensores electromagnéticos soportados por un casco. Ellos podrían rastrear los movimientos más importantes del rostro y mapearlos en tiempo real sobre un muñeco en la computadora. Lo importante de este sistema fue que un solo actor podía manipular todas las expresiones del rostro de un carácter en la computadora, ya que este imita sus propias expresiones, una interfase natural perfecta. [1]

Uno de los primeros grandes sucesos con el “face waldo” fue el funcionamiento en tiempo real del popular personaje de los video juegos en Nintendo, Mario. Este fue conducido por un actor

detrás de la escena que llevaba puesto el “face waldo”. Mario conversó y bromeó con los miembros a una audiencia respondiendo a sus preguntas y comentarios. Desde entonces SimGraphics está inmerso en esta industria desarrollando personajes animados para exposiciones profesionales de dibujos, televisión y otros entretenimientos. [1]

1992: Brad deGraf – Alive!

Después de deGraf/Wahrman – Mike the Talking Head, Brad deGraf continuó trabajando sola desarrollando un sistema en tiempo real que es llamado Alive. Para un carácter desarrollado por Alive, deGraf desarrolló un dispositivo de mano especial con 5 émbolos utilizados por titiriteros. El dispositivo fue utilizado para controlar la expresión facial de una amigable nave espacial generada en computadora. [1]

Posteriormente, DeGraf unido a Colossal Pictures, donde se usó Alive para animar a Moxy, un perro generado en computadora que fue el anfitrión de un show de Cartoon Network. Moxy funcionó en tiempo real para publicidad, pero post – render para el show. Los movimientos de los actores son capturados por un sistema de captura electromagnético con sensores en las manos, los pies, el torso y la cabeza del actor. [1]

1993: Acclaim

En SIGGRAPH’93 Acclaim asombró a la audiencia con 2 realistas y complejos personajes animados hechos completamente con captura de movimiento. Para los años previos Acclaim tuvo un alto desarrollo en el sistema de rastreo óptico, similar a los usados por Graphical Marionette y Dozo, pero capaz de rastrear hasta 100 puntos en tiempo real. Acclaim usa el sistema principalmente para desarrollar movimientos de personajes para videojuegos. [1]

Como se ha podido ver en todo lo expuesto anteriormente, desde los años 70 los sistemas de captura de movimiento han tenido un desarrollo ininterrumpido hasta llegar a los actuales, se han explicado brevemente hasta el año 93 ya que a partir de este momento los demás se puede considerar que caen en la generación de los actuales que serán tratados en el otro epígrafe.

1.3 Animaciones

La animación es la simulación de un movimiento, creada por la muestra de una serie de imágenes o cuadros. [2.1]

Las animaciones están divididas por frames. Un frame no es más que un cuadro o imagen individual de las que componen una animación. La sucesión de frames consecutivos con pequeñas diferencias producen la ilusión de movimiento. La velocidad de la animación se mide en frames por segundo (fr/s).

En ocasiones se tiende a confundir el concepto de animación con el de video, y en realidad no son semejantes. El video toma el movimiento continuo y lo descompone en cuadros, la animación parte varias imágenes estáticas y las une para crear la ilusión de un movimiento continuo. Mientras más imágenes consecutivas componen una animación, esta será mucho más real. Lo que distingue a la animación de las técnicas tradicionales de adquisición de imágenes en movimiento, es que cada fotograma es generado de modo independiente.[2.1]

1.3.1 Principios básicos de la animación

El animador no sólo debe tener la habilidad técnica de dibujar y colocar personajes, sino también un acusado sentido del tiempo, de la observación de las peculiaridades, y del movimiento. Además ha de ser un actor, tener el sentido de qué le hace al elemento para estar vivo y ser

natural. Aunando todas estas diversas disciplinas, el animador puede conseguir la sensación de dotar de vida a un personaje inanimado.[3]

Lo que viene a continuación, son principios que muchos de los maestros tradicionales de la animación han descubierto y aprendido en el camino del desarrollo de su arte. Estos principios son tan aplicables a las formas de animación tradicional como a la animación por ordenador.[3]

Sentido del tiempo (Timing)

El tiempo es la esencia de la animación. La velocidad a la que algo se mueve da una idea de qué es el objeto y por qué se está moviendo. Un simple parpadeo puede ser rápido o lento, si es rápido indica que el personaje está alerta y despierto, si es lento parece cansado y aletargado. [3]

Entrada y Salida lenta en una Clave (Ease In and Out or Slow In and Out)

La entrada o salida lenta en una clave tiene que ver con la aceleración o deceleración gradual de un objeto al llegar o salir de una clave. Un objeto debe ir frenándose según se aproxima a una clave (Ease In) o empezar a moverse gradualmente desde el estado de reposo (Ease Out). Por ejemplo, una pelota botando tiende a tener un montón de "ease in and out" cuando está en la parte más alta de su bote. Cuando va hacia arriba, la gravedad le afecta y va frenando (Ease In), después, inicia su movimiento descendente cada vez más rápido (Ease Out), hasta que golpea el suelo. [3]

Arcos (Arcs)

En el mundo real, casi todas las acciones se mueven en un arco. Al animar, uno debe intentar conseguir trayectorias curvas en lugar de rectas. Es muy raro que un personaje o alguna parte de

él, se mueva en línea recta. Incluso los movimientos más toscos del cuerpo cuando se camina tienden a no ser perfectamente rectos. [3]

Anticipación (Anticipation)

En animación, la acción discurre normalmente por tres etapas: La preparación para el movimiento, el movimiento en si mismo y la prolongación de la acción. A la primera parte se le conoce como Anticipación. En algunos casos la anticipación es necesaria físicamente. Por ejemplo, antes de poder lanzar una pelota, hay que echar el brazo hacia atrás. El movimiento hacia atrás es la anticipación, el lanzamiento en si mismo es la acción. [3]

Exageración (Exaggeration)

La exageración se utiliza para acentuar una acción. Se debe utilizar de forma cuidadosa y equilibrada, no arbitrariamente. Hay que encontrar el objetivo deseado de una acción o secuencia y qué partes necesitan ser exageradas. El resultado será que la acción parecerá más realista y entretenida. La clave es tomar algo y hacerlo más extremo con el objetivo de dotarlo de más vida, pero no tanto que destruya la credibilidad. [3]

Compresión y Extensión (Squash and Stretch)

La compresión y extensión se utiliza para deformar un objeto de manera que se haga evidente su grado de rigidez. Por ejemplo, si una pelota de goma bota y golpea el suelo, tiende a aplastarse un poco en el momento del choque. Este es el principio de compresión. Cuando empieza de nuevo a subir, se estirará en la dirección del movimiento, este es el de extensión. Una característica importante de la compresión y extensión, es que independientemente de cómo un objeto se deforme, debería parecer que conserva su volumen. Es decir, si una pelota se aplasta hasta la mitad de su tamaño normal, tendrá que ser dos veces más ancha para mantener su volumen. [3]

Acción secundaria (Secondary Action)

La acción secundaria es una acción que ocurre como consecuencia de otra acción. Por ejemplo, si un perro está corriendo y de repente se detiene, sus orejas probablemente continuarán moviéndose durante un instante. La acción secundaria aporta interés y realismo a la animación. Debe ser realizada de manera que se note pero que no sobrepase a la acción principal. [3]

Prolongación y Acción Superpuesta (Follow Through and Overlapping Action)

El solapamiento consiste en comenzar la siguiente acción antes de terminar la anterior (por ejemplo, para abrir una puerta, el personaje al acercarse va estirando la mano antes de llegar). La continuación significa que los movimientos no se detienen bruscamente, sino que continúan más allá de su posición final (por ejemplo, al golpear una pelota con una raqueta esta continúa su movimiento por inercia mucho después de haber golpeado a la pelota). [3]

Animación Directa y Animación Clave a Clave (Straight Ahead Action and Pose-To-Pose Action)

Existen dos métodos básicos para crear animación. La Animación Directa es uno de ellos, donde el animador dibuja o ajusta objetos un cuadro tras otro, ordenadamente. Por ejemplo, el animador dibuja el primer cuadro de la animación, después el segundo, y así sucesivamente hasta que la secuencia está completa. De esta forma, hay un dibujo o imagen por cuadro que el animador ha preparado. Este procedimiento tiende a conseguir un acabado más creativo y fresco pero puede dar dificultades a la hora de controlar y ajustar el tiempo. [3]

El otro procedimiento es la Animación Clave a Clave. Se crea mediante el dibujo y ajuste de poses clave y después el dibujo o creación de imágenes intermedias. Está es la técnica básica de claves mediante ordenador de aproximarse a la animación. Es excelente para ajustar el timing y

planear la animación a lo largo del tiempo. Primero se ajustan las poses clave, y después el movimiento intermedio se genera desde ellas. Es muy útil para un timing específico o cuando una acción debe ocurrir en un momento concreto. Se sabe siempre qué ocurrirá. [3]

La diferencia fundamental es que con la Animación Clave a Clave se planifica, y se conoce exactamente lo que ocurrirá a lo largo del tiempo, mientras que con la Animación Directa, no se está realmente seguro de como saldrán las cosas hasta que se ha terminado. [3]

Colocación (Staging)

La Colocación es la presentación de una acción o elemento de manera que se entienda con facilidad. En general, se presenta un elemento de acción cada vez. Si ocurren demasiadas cosas a la vez, la audiencia dudará hacia adonde mirar y la acción estará "sobrecolocada". Una faceta importante de la colocación es la colocación en "Silueta". Lo que significa que una pose, un objeto o un personaje puede ser interpretado incluso en una silueta en blanco y negro. Si no se puede "leer" la pose del personaje en silueta es que no es una pose fuerte y podría seguramente ser mejorada. [3]

Atractivo (Appeal)

El Atractivo tiene que ver con algo que a la gente le gusta ver. Esto puede ser una especie de encanto, diseño, simplicidad, comunicación o magnetismo. El atractivo se puede conseguir con una correcta utilización de otros principios como la exageración en el diseño, evitando la simetría, utilizando la acción superpuesta, y otros. Hay que esforzarse por evitar diseños, formas y movimientos débiles o desagradables. [3]

Personalidad (Personality)

Esta palabra no es realmente un verdadero principio de animación, sino que se refiere a la correcta aplicación de los otros principios. La personalidad determina el éxito de una animación. La idea es que la criatura animada realmente cobre vida y se introduzca en el papel. El mismo personaje no debería realizar una acción de la misma forma en dos estados emocionales diferentes. Dos personajes distintos no deberían actuar de la misma manera. Es también importante hacer que la personalidad de cada personaje sea diferente, pero al mismo tiempo, familiar a la audiencia. [3]

1.3.2 Animación por ordenadores

La animación por ordenadores se puede definir como la “generación, almacenamiento y presentación de imágenes que en sucesión rápida producen sensación de movimiento”, un formato de presentación de información digital en movimiento a través de una secuencia de imágenes o fotogramas creadas o generadas por la computadora.[2.1]

1.3.3 Animación 3D por ordenadores

La animación 3D por computadora le brinda un mayor realismo a las animaciones, ya que permite crear animaciones realmente tridimensionales. Esto permite una visualización de la animación desde varios ángulos, esto se logra con solo cambiar el ángulo de la cámara para ver otra parte de la escena.[2.1]

Otra ventaja de la animación por computadora es que se pueden reutilizar partes de la animación por separado. Una animación puede verse muy diferente con solo cambiar el ángulo de la cámara o el tiempo del movimiento o de partes de la animación.[2.1]

Se puede lograr que una animación se vea mucho más realista si se varía el peso y el tamaño de los objetos. Gracias a las nuevas técnicas de graficación, los objetos se pueden ver mucho más realistas. Para cambiar el peso que tiene un objeto es necesario cambiar el tiempo que tarda este en moverse. Esto se debe a que mientras más pesado sea un objeto, hay que aplicarle una mayor fuerza para que este se mueva, por lo que tiene una menor aceleración. Por todo esto a la hora de realizar animaciones se debe tener en cuenta los atributos de cada elemento de la animación, para poder diferenciarlos adecuadamente.[2.1]

En la vida real, los cuerpos se mueven como resultado de sus pensamientos. Todos los movimientos se realizan dependiendo tanto de su personalidad como de su carácter, estado de ánimo, factores físicos, etc. Cada movimiento se realiza por una razón. Si se modela un objeto se debe tener en cuenta la forma en que se moverá.[2.1]

Otro punto importante que debe tenerse en cuenta en la animación en tres dimensiones es la forma en que se detienen los cuerpos. Si un modelo se detiene por completo, es decir, que todas sus partes se detienen al mismo tiempo, este movimiento no se verá realista. Cuando un cuerpo vivo se va a detener, las fuerzas de la inercia hacen que se vaya deteniendo poco a poco. Debido a esto, al animar un personaje es bueno que si se va a detener, alguna parte del cuerpo se siga moviendo ligeramente.[2.1]

Hay tres frases que componen una acción, la anticipación de la acción, la acción en sí y la reacción a la acción. Para que una animación se vea realista es fundamental que el tiempo empleado en cada parte de la acción sea el indicado. También se debe mantener el flujo y la continuidad entre las acciones y esto se logra traslapando un poco cada parte de la acción.

1.3.4 Técnicas de animación de personajes 3D

Animación basada en fotogramas: Esta técnica está entre las más utilizadas. En este método se filman las imágenes fotograma por fotograma, y luego se unen formando de esta forma la animación. Se pueden hacer bibliotecas de movimientos de cada parte del cuerpo de la animación para más adelante poder hacer animaciones diferentes producto de la combinación de estas partes.[2.1]

Keyframing: El keyframing consiste en marcar posiciones en momentos específicos de una animación, las posiciones intermedias las haya la computadora por interpolación. Es fundamental que esta operación se realice en fotogramas importantes dentro de la animación, es decir, que sean representativos del movimiento para tener más información y que la interpolación tenga más calidad.[2.1]

Rotoscopiado: Es una forma mejorada de keyframing, donde se captura un movimiento real, y se utiliza esa información para mover un diseño generado por computadora.[4]

Wavelets: Wavelets significa “pequeñas ondulaciones”. Esta técnica posibilita que en una imagen se compriman una gran cantidad de datos para que al acercarse a ella se vayan viendo los detalles sin distorsión.[4]

Animación procedural: Consiste en describir el movimiento de forma algorítmica. Hay una serie de reglas que controlan cómo se van modificando los distintos parámetros(como la posición o la forma) a lo largo del tiempo. Para movimientos sencillos(un péndulo o una rueda que gira) es una buena solución, pero para movimientos más complejos(una persona caminando o una moneda

que cae al suelo), resulta difícil conseguir buenos resultados. Hay algunas técnicas con resultados interesantes, como los sistemas de partículas o la simulación de movimientos.[4]

Animación por esqueleto: consiste en deformar la malla de un cuerpo a través de una estructura jerárquica de huesos, a los que se le asocia un grupo de vértices. Esta técnica actualmente es muy utilizada por los programadores porque da la posibilidad de incluir muchos detalles en la animación de un personaje, desde las arrugas de la piel hasta la definición de los músculos, imprimiéndole mucho mayor realismo.[5.1]

1.6 Sistema de animación por esqueleto

En este sistema para la animación de un personaje se necesita una jerarquía de “huesos” además de la malla 3D. La transformación de esta jerarquía trae consigo la deformación del personaje. Los huesos eliminan el almacenamiento de la posición de cada uno de los vértices por cada frame, superando en este aspecto a la animación basada en vértices. Otra de las grandes ventajas que presenta este tipo de animación es que como las animaciones son independientes a las mallas, en una misma escena una animación puede aplicársele a diferentes mallas.[5.1]

Las deformaciones antes mencionadas ocurren debido a que los huesos según su posición lógica con respecto a la malla tienen influencia sobre una serie de vértices de la misma. Hay dos maneras que surgieron para mejorar la técnica de deformación y que dan como resultado lo que se puede ver actualmente en los videojuegos y simuladores más avanzados. A continuación se explican las mismas enunciadas por orden de surgimiento.[6]

Stitching: En este caso los vértices de cada hueso son transformados aplicándoles en cada frame la matriz de transformación del hueso. La deficiencia de esta técnica es que pierde suavidad la

mallas en las intersecciones de los huesos pues al reflexionarse una articulación del cuerpo los vértices más cercanos a esta y que pertenecían a huesos diferentes se alejan demasiado, lo que crea ángulos abruptos en la zona de la articulación y una sensación de rigidez.[6]

Skinning: Esta técnica es bastante parecida a la anterior pero con la diferencia de que mejora su deficiencia. Los vértices pueden ser influenciados por más de un vértice. Para estos casos cada hueso tiene un valor entre cero y uno llamado weight (peso) que indica cuanta influencia tiene este sobre el vértice. [6]

1.7 Jerarquía de personajes

La jerarquía de personajes no es más que la forma en que están relacionadas las partes del cuerpo de un personaje. Cada vez que se mueve el hueso padre se mueven todos sus hijos, esto es conocido como cinemática directa. Por otra parte se tiene la cinemática inversa, en la que ocurre lo contrario, los hijos controlan el movimiento de los padres.

La jerarquía de un cuerpo es como un árbol, parte de la cadera, la que controla todas las ramas. Esto es debido a que la cadera es aproximadamente el centro de gravedad y el centro de distribución del peso de todo el cuerpo. [5.1]. Ver Anexo 1.

1.7.1 Manipulación de la jerarquía

Como ya se dijo anteriormente, en una jerarquía de este tipo los huesos padres manipulan a los hijos en el caso de que se esté utilizando la cinemática directa. Cuando ocurre el fenómeno contrario se le llama cinemática inversa.

En el caso de la animación de personajes la cinemática inversa tiene algunas opciones más que la cinemática directa, aunque esto la hace más compleja. Sin embargo, después de establecida hace mucho más rápida la creación de una animación. La animación inversa es muy sencilla de utilizar. Para decidir cuál utilizar depende de qué parte del esqueleto se quiere animar.[5.1]

La cinemática directa es más usada en la parte superior del cuerpo: columna, hombros y brazos. Si el personaje está de pie gesticulando con las manos, esta técnica es muy buena para lograr un buen trabajo. Si el personaje realiza una acción en la que hace contacto con otra cosa, ya sea objeto u otro personaje debe ser utilizada la cinemática inversa. La columna y las manos son las más animadas con cinemática directa.[5.1]

La cinemática inversa siempre es usada para la animación de las piernas y los pies. Esta es ideal para la acción de colocar los pies del personaje sobre el suelo. Si el personaje está nadando, la cinemática directa es más útil tanto para piernas y pies como para los brazos.[5.1]

1.8 Formatos utilizados para las animaciones

Muchos son los formatos que se han desarrollado para animaciones, pero entre ellos los más utilizados son el BVH, CSM y el BIP. La utilización de estos formatos es debido a que son de fácil lectura y comprensión, además de que las herramientas más utilizadas en el mundo de la realidad virtual trabajan con estos formatos.

BVH: El formato BVH Biovisión es un fichero ASCII que es usado para importar los datos de la rotación de las articulaciones desde varios sistemas de captura de movimiento. Este fue desarrollado originalmente por Biovisión, una compañía que se dedica a los servicios de captura de movimiento. Es un excelente formato, pero tiene solo “offset” traslacionales desde los

segmentos hijos hasta sus parientes, no tiene definidos los “offset” rotacionales. También carece de información explícita para cómo dibujar los segmentos pero esto no entorpece la definición del movimiento.[8]

BIP: El .BIP es un formato que representa un humano, ya que es similar a un esqueleto hecho de cajas conectadas. La estructura del esqueleto es una jerarquía clásica de partes del cuerpo conectadas. La pelvis es la raíz de todos los objetos. Este formato proviene de Character Studio.[16]

CSM: El formato CSM(Character Studio Motion) es un fichero ASCII utilizado para importar el dato de la posición de marcadores desde varios sistemas de captura de movimiento a Character Studio para animar personajes bípedos. Biped suministra la entrada directa del fichero CSM desde disco, incluyendo reducción de keyframe, extracción de huellas, talento de la figura y la calibración de la pose, suministrando un rápido y exacto método de importar para volúmenes grandes de información sobre la posición de los marcadores en un fichero CSM.[23]

1.9 Estado del arte en la actualidad.

Varios son los sistemas desarrollados en el mundo dedicados a la captura de movimiento para animaciones. Todos tienen sus características particulares con sus ventajas y desventajas, pero hay tres tipos de sistemas que son los más utilizados, y ellos son el sistema óptico, el magnético y el electro – mecánico. En este epígrafe se explica cada uno de estos sistemas con sus pros y sus contras, llegando a conclusiones que ayudarán a una mayor comprensión de la solución propuesta en este trabajo, además, se da una breve explicación de la captura facial.

1.9.2 Sistema óptico de captura de movimiento

En la captura de movimiento mediante el método óptico se utilizan dos tecnologías principales, la reflectiva (Activa) y la de pulsado – LED, con diodos emitiendo luz (Pasiva). Una de las principales diferencias entre estos dos sistemas es que los marcadores activos pueden comunicar identificaciones únicas por la modulación de sus pulsos, mientras que los sistemas pasivos deben inferir la identidad de los marcadores a partir de observaciones continuas. En la Figura 1 del Anexo 2 se muestra el proceso de captura de movimiento utilizando un sistema activo, se puede ver el actor virtual y el esqueleto que genera durante la captura.[9]

En la práctica, ambos tipos de sistemas ópticos son fidedignos, aunque la captura de los datos puede no ser correcta en caso de que hayan aberturas durante esta ocasionada por la oclusión o por la salida del sujeto de la región de captura. Ambos sistemas utilizan el video, utilizando cámaras para rastrear el movimiento de los marcadores(o LEDs) acoplados a las articulaciones del cuerpo. Los sistemas sencillos o duales de cámaras son adecuados para la captura facial, un sistema de 3 a 16 o más cámaras son los utilizados para realizar la animación del cuerpo completo. El movimiento captado por las diferentes cámaras es combinado por el método de triangulación para calcular sus posiciones frame a frame en el espacio 3D. Varios problemas ocurren a menudo durante este proceso de rastreo, como son el intercambio de marcadores, ruido o pérdida de datos en reflexiones falsas.[9]

Los softwares utilizados en MOCA usan esqueletos biomecánicos durante la captura permitiendo suministrar datos limpios con baja probabilidad de ruido, pérdida de datos o reflejos falsos. A continuación se detalla el sistema de captura pasivo (ver Figura 2 del Anexo 2), ya que este es el más utilizado en el mundo.

1.9.2.1 Sistema pasivo de captura de movimiento en tiempo real.

En este sistema se emplean marcadores reflectantes pasivos, esto elimina restricciones como el cableado u otros dispositivos que puedan limitar los movimientos del actor. Posee hardware y software especializado, cámaras, cableado y reflectores. Todo el sistema funciona de manera automática y es manejado por una interfaz. El tiempo real permite la visualización del movimiento del actor sobre el personaje que se esté animando en tiempo real. Toda la comunicación con la captura se realiza a través de una API(application programming interface) que facilita su inclusión en otros programas comerciales.[10]

El proceso de captura es el siguiente.

1. Colocación de marcadores: Es necesario colocar los marcadores sobre el actor en los puntos adecuados, según el tipo de movimiento que vaya a reproducirse.[10]
2. Proceso de captura: MOCA enciende las luces y recibe los datos de los marcadores del área de captura, que a su vez, procesa en tiempo real.[10]

El sistema de captura contiene configuraciones de marcadores para cuerpo completo, tren superior o tren inferior en diferentes niveles de detalle. Al tener cada una de ellas muchas variantes el nivel de configuraciones disponibles es muy grande. No obstante el tiempo de desarrollo de nuevas configuraciones de marcadores que fueran necesarias desarrollar es muy reducido.[10]

La calibración de los sistemas de captura ópticos ha sido tradicionalmente largo y tedioso, pero en el caso de este sistema se logró reducir a unos pocos segundos. Si las cámaras se colocan fijas en la pared o en el techo la calibración deja de ser necesaria a partir del momento en que estas quedan instaladas, de otra forma el Test de calibración permite detectar si alguna de las cámaras ha sido movida de su posición.[10]

El sistema se compone por un conjunto de cámaras e iluminadores controlados desde un ordenador de control. Este ordenador puede estar conectado a una red local, de manera que se pueda realizar operaciones de captura desde otros ordenadores así como de visualización.[10]

Este sistema permite también la captura del movimiento de varios personajes a la vez (ver Figura 3 del Anexo 2). Además, utiliza esqueletos cinemáticos como soporte de cálculo durante el proceso de captura produciendo resultados limpios.[10]

A pesar de todo los sistemas ópticos de captura de movimiento aún están lejos de la perfección, son los más utilizados en el mundo y con ellos se han logrado buenas animaciones para varios propósitos como son video juegos, películas y otras cosas. Como ya se ha dicho sus principales dificultades están en la oclusión, en ocasiones, de los marcadores o cuando el algoritmo confunde la trayectoria de un marcador con la de otro. Esto requiere de mucho trabajo de edición por parte del animador después de terminado el proceso. Aunque hay algunos métodos en los cuales se trabaja intensamente que tratan de resolver estos problemas.[10]

1.9.3 Sistema magnético de captura de movimiento

Consiste en el uso de un transmisor centralmente localizado y un set de receptores adheridos a partes del cuerpo del actor. Estos receptores son capaces de medir su relación espacial con el transmisor central. Cada receptor es a su vez conectado a una interfaz que puede ser sincronizada. Este sistema tiene la ventaja de que no hay posibilidad de oclusión en la recepción de la información del movimiento, pero hay varios factores negativos como el obstáculo de los cables, no soporta suficientes receptores para capturar de una forma precisa el carácter del movimiento y limitada área de movimiento. Además, el sistema puede ser afectado por cualquier elemento metálico vecino al área. En la Figura 4 del Anexo 2 se muestra un momento del proceso de captura de movimiento utilizando el sistema magnético. Ahí se puede ver al actor que se está utilizando para realizar los movimientos y superpuesto se puede ver el esqueleto

generado en ese instante como resultado de ese proceso. En esta figura los sensores de la pelvis se encuentran en la parte trasera del actor.[11]

1.9.4 Sistema Electro – Mecánico de captura de movimiento

El actor se coloca una armadura humana de piezas metálicas(similar a un esqueleto). Esta armadura es enganchada hacia la parte trasera de las articulaciones del actor, la armadura posee sensores que van exactamente en las articulaciones del actor para de esta forma sentir las rotaciones que realizan las mismas. Otros tipos de sistemas mecánicos utilizan guantes o modelos articulados. Una de las ventajas que presenta es que no interfieren luces o campos magnéticos. Pero este sistema posee grandes desventajas como que el equipo debe ser calibrado con frecuencia, la tecnología no tiene una percepción de la tierra, así que con él no se puede saltar, en ocasiones las cargas tienden a deslizarse, y otras que hacen que este sistema sea menos usado que el óptico y el magnético.[12]

El sistema de captura de movimiento por el método electro – mecánico más famoso que hay es el Gipsy, siendo este menos caro en precio que los sistemas ópticos y magnéticos. La mayoría de las pequeñas compañías utilizan este sistema por las razones antes mencionadas. La versión Gypsy4 posee una armadura de peso ligero y simple de operar, permite más movimientos al actor, utiliza 43 sensores de movimiento localizados en 17 articulaciones del cuerpo humano para grabar lo mejor posible la rotación de los huesos del actor. En la Figura 5 del Anexo 2 se presenta una figura que ilustra este sistema para que se tenga una idea de cómo es y de cómo es su utilización. Aunque se ha mejorado considerablemente se debe tener en cuenta que al utilizar esta armadura metálica, limita un poco los movimientos del actor aunque como ya se dijo, esto ha sido considerablemente mejorado en esta última versión.[13]

1.9.5 Captura facial

También existen sistemas para la captura de los movimientos faciales (ver Figura 6 del Anexo 2). Estos sistemas son muy importantes para lograr realismo en la expresión de los personajes virtuales.

Todos los dispositivos de captura facial cuentan con tecnología óptica o de video. Los rastreadores en tiempo real para la realización de animaciones consisten de una cámara de video muy ligera que es adaptada en la cara de los artistas con correas encima y detrás de la cabeza o un ligero casco. La cámara de video toma el movimiento bidimensionalmente del conjunto de marcadores o puntos que son pegados a la cara. Varias configuraciones de marcadores son posibles, pero la siguiente configuración básica es generalmente el punto de inicio: [14][15]

- Lados y encima de los labios.
- Mejillas.
- Cejas.
- Encima y debajo de los párpados.

1.10 Desarrollo del tema en nuestro país y nuestra universidad

En nuestro país muy pocas personas se han dedicado a desarrollar el campo de la realidad virtual, en muchas ocasiones por la falta de recursos y de conocimientos. Y en cuanto al tema de la captura de movimiento para desarrollar animaciones no se tiene información de que exista un sistema 100% cubano que sea capaz de llevar a cabo este proceso. Sí se realizan animaciones pero utilizando herramientas que son capaces de hacerlas por métodos bastante lentos, y que además, no utilizan la captura de movimiento a partir de un actor real, a parte de que exigen una gran dedicación y desgaste por parte del animador. Esto es debido a que todos los sistemas antes mencionados se consiguen en el mercado mundial a precios que no están acorde con la

situación económica de nuestro país, de ahí la necesidad de profundizar más en este tema con el objetivo de buscar soluciones factibles a un país como Cuba.

En el caso de la Universidad de las Ciencias Informáticas, se puede decir que aún no ha sido desarrollado ningún software para este fin, las animaciones que se realizan son elaboradas utilizando herramientas capaces de hacerlas, pero como se dijo anteriormente, exigiendo un máximo de esfuerzo y habilidad por parte del animador y sin tener en cuenta la técnica de captura de movimiento, todo esto unido a lo lento que es este proceso.

1.11 Conclusiones

Este capítulo ha servido para obtener los conocimientos que se necesitan para lograr un buen entendimiento de todo lo referente a la nueva solución que se da en este trabajo para la captura de movimiento. Se brindaron varios conceptos relacionados con la animación, se brindó una detallada explicación de toda la evolución de los sistemas MOCA, se describieron los formatos más utilizados para animaciones, se explicó cada una de las técnicas usadas actualmente para la captura de movimiento.

De la experiencia sacada a partir de los conocimientos adquiridos en este capítulo, se llegó a una solución que será expuesta en el siguiente capítulo.

Capítulo 2

2.1 Introducción

En el presente capítulo se describirá detalladamente la solución que se propone partiendo del problema y la situación objetiva planteada. Para ello se realizaron una serie de análisis los cuales llevaron a una propuesta de software que resolviera tales problemas. Aquí se podrá ver documentado el proceso de captura de requisitos como casos de uso, el cual es el encargado de dirigir la evolución del sistema en lo adelante.

Con la propuesta que se expone lo que se pretende es brindar una solución que aunque aún no es la ideal, sí mejora en gran medida los problemas existentes actualmente en esta materia en nuestro país.

2.2 Planteamiento del problema

Uno de los proyectos con que cuenta la UCI es el proyecto de Realidad Virtual, el cual se dedica fundamentalmente al desarrollo de paseos virtuales y juegos. Una de las tareas en que más se ha visto inmerso este proyecto ha sido en el desarrollo del paseo virtual por la UCI. Hace dos años se hizo una primera versión con vistas a ser llevada a la Cumbre de la Información, la cual tuvo lugar en Ginebra. Actualmente se orientó por parte de la universidad, que se realizara una segunda versión, y se decidió por parte del proyecto, que en esta segunda versión aparecieran personajes transitando por las calles de la UCI para que el paseo fuera mucho más real. La situación problemática que se presentó fue la falta de movimientos reales en las animaciones de personajes que se realizan en el proyecto, además del enorme tiempo que se invierte en las mismas, esta situación es extensible a todos los proyectos de la UCI que trabajan los entornos virtuales, ya que no se cuenta con otra forma de llevar a cabo esta tarea. En el mundo existen varias aplicaciones y métodos para realizar esto, pero son extremadamente costosos y

prácticamente inaccesibles para un país como Cuba, por lo que se necesitaba una solución para el desarrollo de animaciones de calidad y en poco tiempo en la UCI. Después de analizado todo lo anterior se llegó a la conclusión de que el problema radicaba en la carencia de una herramienta capaz de disminuir el tiempo de elaboración de animaciones para que se realicen de manera mucho más rápida los entornos virtuales y muchas otras variantes relacionadas con este mundo, debido a que la UCI no cuenta con ninguna herramienta de este tipo.

2.3 Objeto de estudio

El objeto de estudio para la búsqueda de la solución al problema planteado es el desarrollo de animaciones tridimensionales para entornos virtuales.

2.4 Campo de acción

El campo de acción está enmarcado en los sistemas de captura de movimiento, ya que un sistema de este tipo es el que podría dar respuesta al problema existente.

2.5 Descripción de la solución

Atendiendo al problema que presentan los métodos actuales con que se cuenta, tanto en la parte del tiempo como de la calidad, se propone una nueva solución que agilizará de manera muy significativa el tiempo de desarrollo, y que a su vez logrará una animación mucho más real que las que se realizan actualmente en nuestra universidad.

Después de un minucioso estudio de las aplicaciones que generan animaciones en el mundo, se llegó a la conclusión de que el método que más se prestaba para lo que se quería y podía lograr era el óptico. Como ya se explicó anteriormente, este método consiste en capturar los movimientos utilizando un video de una persona realizando los movimientos que se desean, a

esta persona se le coloca una traje especial con marcadores en sus articulaciones, cada uno representa un hueso en el esqueleto que se quiere animar, por lo que el movimiento que este marcador realice será el que adquiera su homólogo en el esqueleto diseñado. A la hora de realizar este video se colocan varias cámaras para tener la filmación desde diferentes campos visuales. El procesamiento de estos datos lo hace un software especializado.

Se propone un software que disminuye significativamente el costo de las aplicaciones de este tipo. Esta aplicación, aunque no es un método óptico, sí se asemeja bastante a este ya que también necesita de videos para desarrollar su animación, pero en realidad es un método manual. Este método sustituye los marcadores de los actuales métodos por el Mouse de la computadora, necesita solamente dos cámaras para realizar la filmación, y es capaz de ejecutar todos los cálculos pertinentes para realizar una buena captura de movimiento, superando en calidad y en tiempo a las técnicas actuales con que cuenta el proyecto de Realidad Virtual y sus homólogos en la UCI. El sistema a desarrollar es el SisMa (Sistema Manual). Es bueno aclarar que esta aplicación, para que la solución se complete, tendrá como complemento el 3Dmax, esto es debido a que en la herramienta propuesta se desarrolla toda la captura del movimiento, pero en algunos tipos de animaciones esta no queda totalmente bien, por lo que se deberá cargar al 3Dmax para darle un acabado. Seguidamente se expondrá la solución completa, en la cual, el papel fundamental lo tiene la herramienta a desarrollar.

2.5.1 Filmación de los videos

En la aplicación que se desarrollará, se necesita tener para la captura del movimiento dos videos, para esto se escoge una persona, que servirá como actor y hará los movimientos que se quieren lograr, la diferencia con las aplicaciones existentes en el mundo es que esta persona no necesitará un traje con marcadores en su cuerpo, esto disminuye el costo significativamente. Es esencial que las cámaras estén aproximadamente en un ángulo de 90 grados una de otra, ya que

un video será utilizado para generar las coordenadas espaciales X y Y, y el otro la Z. A continuación se muestra una figura en la que se ve cómo deben estar ubicadas las cámaras con respecto al actor virtual y entre ellas.

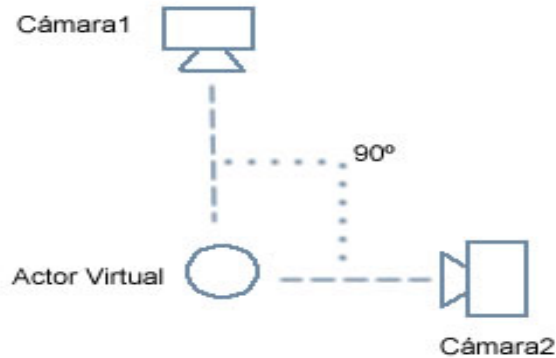


Figura 1: Posición de las cámaras para obtener la filmación.

2.5.2 Generalidades del sistema

2.5.2.1 Lenguaje utilizado

El lenguaje que se utilizará para la realización de este software es el C++, específicamente a través del RAD C++Builder 6, que es el compilador con el que se cuenta, además de ser el que mejor se domina. La razón por la que se tomó la decisión de emplear el C++ fue porque como es conocido, este lenguaje, a pesar de su complejidad, es muy potente y veloz, ya que presenta funciones a bajo nivel, permitiendo que se acelere el procesamiento. Además, tiene muy bien desarrollado el paradigma de la programación orientada a objetos (POO). Se trabajará con funciones del C++ estándar, lo que permite migrar a otros compiladores para su posterior

comercialización. Otra de las causas por las que se seleccionó el C++ es porque en este lenguaje es más sencillo trabajar con la librería OpenGL, la cual será utilizada para la visualización de la animación.

OpenGL es una librería gráfica, y se optó por ella debido a que es la que más se domina, a parte de permitir la migración a software libre. Además, en futuras versiones del sistema propuesto, el trabajo con este tipo de librerías será mayor.

2.5.2.2 Escenas

Antes de comenzar a hablar de las particularidades de SisMa es fundamental dejar claro cómo es el manejo y la organización de los datos dentro de la herramienta. En SisMa las animaciones están agrupadas por escenas, una escena no es más que un conjunto de animaciones realizadas tomando como modelo un mismo esqueleto, por ejemplo, si se desea hacer una animación de un hombre corriendo, luego saltando y después caminando agachado, pues se carga el esqueleto del hombre y se crean tres animaciones, una para cada uno de estos movimientos(en caso de que no se tengan todos en un mismo video), entonces con ese esqueleto y cambiando solo los videos se realizan las tres animaciones, luego se salvan en forma de escena en un mismo fichero.

Las animaciones son agrupadas a conveniencia del animador, pueden haber escenas de una sola animación, o de varias animaciones aunque no estén relacionadas entre sí. El sistema soporta en cada escena tantas animaciones como se desee tener.

2.5.2.3 Formatos relacionados

El sistema que se propone trabaja con tres formatos diferentes de ficheros. Ellos son el .ASE, que es el que el sistema importa desde el 3Dmax, ya que este posee toda la información

referente al esqueleto, otro formato es el csm, que es hacia el que se exporta para que la información de la captura pueda ser vista y arreglada en 3Dmax, y el otro formato es el mca, que es el formato propio del sistema, los ficheros que se salvan y se cargan en SisMa son de extensión mca.

A continuación se especificará cada uno de estos formatos para que se tenga una mayor comprensión de cómo ocurre todo el proceso.

ASE: El formato .ASE es nativo de 3D Studio Max. Está basado en identificadores, todos en la forma *ASTERISK_THEN_NAME, los cuales son seguidos por cero o mas valores, y además por un pequeño bloque de identificadores adicionales encapsulados por llaves. Este formato es el preferido por muchos softwares para importar mallas estáticas.[17]

CSM: Como ya se dijo anteriormente, hacia este formato es al que se exporta la captura realizada en SisMa, es un formato que los datos que guarda es la posición de cada uno de los marcadores. El biped de 3Dmax usa los datos almacenados en el CSM para su pose frame a frame. Durante el proceso de importar, la posición x,y,z de los marcadores almacenados son usadas para manejar la rotación de los huesos del biped para cada frame. Es necesario aclarar que dentro del CSM hay marcadores que no son obligatorios, es decir, que a la hora de exportar a CSM no son imprescindibles para la animación del biped, pero que en caso de que se pongan ayudarían más a la calidad de la animación. (ayuda del 3d)

MCA: En el caso de este formato se brindará una explicación detallada del mismo debido a que es algo nuevo, ya que, como se dijo anteriormente, es la extensión que tienen las escenas salvadas con SisMa.

El formato mca se compone de un encabezamiento donde presenta la dirección del esqueleto que se animó, el número de huesos de dicho esqueleto, el número de animaciones que tiene la

escena, porque como ya se sabe, una escena puede estar formada por varias animaciones. A continuación se presenta un ejemplo de este encabezamiento.

```
Squelet = D:\Tesis\Esqueletos\Humano.ASE  
nBones = 17  
nAnimaciones = 1
```

Seguidamente al encabezamiento se comienzan a describir cada una de las animaciones, la descripción de una animación consta de un encabezamiento también que comienza por el nombre, la ruta de acceso a cada uno de sus dos videos, el valor del offset, el número de frames que tiene, y si la misma fue realizada Pose a Pose, esto es un valor booleano. Aquí se muestra un ejemplo de este encabezamiento.

Animación 0:

```
Nombre = Animation_0  
Video1 = D:\Tesis\Videos\VideosParaAnimations\Animation 2\Video1.avi  
Video2 = D:\Tesis\Videos\VideosParaAnimations\Animation 2\Video2.avi  
Offset = 0  
FrameWidth = 40  
nFrames = 51  
PoseToPose = '-1'
```

Después de esta descripción general de la animación pues se pasa a los valores de los huesos en cada frame. En caso de que el frame contenga valores se pone Frameinfo y el número del frame, indicando que en él hay información, más abajo se indica si este frame es un KeyFrame(esto es para el caso de las animaciones hechas Pose a Pose), para esto se utiliza la variable IsKeyFrame, este valor es booleano. Seguidamente se pondrán los valores de la posición de cada hueso ordenados de la forma x y z Aux_Y. La variable Aux_Y se utiliza para

colocar correctamente los marcadores en el segundo video, ya que al no encontrarse el actor a la misma altura en ambos videos pues entonces se necesita de esta Y auxiliar para que los marcadores puedan estar en su sitio, pero este valor no es utilizado en la animación, es solo para una mejor visualización de los marcadores. A continuación se puede ver un fragmento de cómo van colocados los valores de los frames. En este fragmento hay dos frames.

FrameInfo 0:

IsKeyFrame = '-1'

bone 0: X0 Y0 Z0 Aux_Y0

bone 1: X1 Y1 Z1 Aux_Y1

bone 2: X2 Y2 Z2 Aux_Y2

.

.

.

bonen: Xn Yn Zn Aux_Yn

FrameInfo 1:

IsKeyFrame = '-1'

bone 0: X0 Y0 Z0 Aux_Y0

bone 1: X1 Y1 Z1 Aux_Y1

bone 2: X2 Y2 Z2 Aux_Y2

.

.

.

bone n: Xn Yn Zn Aux_Yn

Por último, en caso de que haya algún frame que no tiene valores pues solamente se pone:

FrameNone (número del frame):

Luego, en caso de que la escena tenga otra animación pues se comienza exactamente igual que lo descrito anteriormente, con su encabezamiento y luego los valores de los registros.

2.5.2.4 Esqueleto utilizado

SisMa será un sistema específicamente para la captura de movimiento, ya que con él se capturarán las posiciones espaciales de marcadores que representan las articulaciones de un esqueleto virtual, que es en definitiva el que va a recibir la animación. La herramienta importará ficheros con extensión .ASE, este tipo de ficheros contiene un esqueleto desarrollado en 3Dmax. SisMa muestra el esqueleto de forma jerárquica para que el usuario pueda tener la idea de qué hueso está animando en cada momento, además de que es muy importante que se conozca esta jerarquía.

2.5.2.5 Videos

Como ya se dijo antes, SisMa necesitará de dos videos para realizar la captura del movimiento. La herramienta brindará la posibilidad de cargarlos simultáneamente ya que presenta un reproductor propio que a parte de tener las facilidades fundamentales que posee cualquier reproductor, también dará la de disminuir y aumentar la velocidad de reproducción y la de mostrar el frame actual. Otro parámetro que mostrará este reproductor es el offset, la utilidad de este valor tiene su explicación en la filmación de los videos. Como es lógico, es muy difícil que en ambos videos el movimiento del actor esté perfectamente sincronizado, para esto SisMa tendrá otro reproductor a parte del antes mencionado, pero en este se podrán ver ambos videos y aumentar y disminuir el offset, que no es más que la diferencia de frames que hay en el movimiento del actor en ambos videos, con la modificación de este valor se logra una completa

sincronización de los videos, quedando de esta manera listos para ser utilizados en la captura del movimiento.

2.5.2.6 Visualización de la animación

Algo que es fundamental dentro de SisMa es la visualización de la animación que se está realizando, ya que sirve para que el animador pueda tener una idea de cómo va quedando su trabajo. Para esto se tendrá una ventana tridimensional en la que se irá pintando el esqueleto simultáneamente con la captura que se está realizando, este esqueleto se verá en el espacio, como es lógico, y esta ventana brindará la posibilidad de rotarlo, moverlo, acercarlo y alejarlo, esto es fundamental para que el usuario pueda hacer una revisión minuciosa de su trabajo y corregirlo en caso que sea necesario.

2.5.2.7 Salvas

Como ya se dijo anteriormente, SisMa tendrá un formato de fichero propio, que es en el que se salvan las escenas realizadas en esta herramienta. La extensión de estos ficheros será .mca(moca) y sirve para guardar las escenas que se han realizado para su reutilización o para su posterior modificación.

Esto es muy importante ya que en ocasiones no se desea continuar trabajando en la escena y se guarda para luego continuar, o por otra parte puede ser que se necesite realizar una animación que ya se ha hecho antes, si se tiene guardada esta escena pues entonces se podrá reutilizar.

2.5.3 Capturando los puntos

Después de que el sistema esté listo para comenzar, se echarán a andar los videos, estos se ejecutan simultáneamente pero primero se toman los valores de uno y luego del otro. Como ya se

dijo anteriormente, en el video 1 se capturarán la X y la Y, y en el video 2 la Z. Para capturar los puntos se marca el hueso que se quiere animar en el listado jerárquico de estos que mostrará la aplicación. SisMa brinda dos formas de capturar el movimiento, decidirse por una de las dos es elección del animador. Estas formas son Pose a Pose y Continuo. A continuación se describirá cada una de estas formas de capturar el movimiento.

2.5.3.1 Continuo

En esta variante se sigue con el mouse el movimiento de la articulación de la persona en el video, y como es lógico, debe ser la articulación que corresponde al hueso marcado en el listado. Como se dijo antes la aplicación brinda la posibilidad de disminuir la velocidad de los videos para que sea más fácil su seguimiento, además, coloca marcadores sobre los huesos animados para que se sepa cuales faltan y para saber si se está haciendo un buen trabajo, es decir, si se está siguiendo la trayectoria correcta. Esta operación se realiza para cada hueso, hasta que se vea en la ventana que muestra el esqueleto en el espacio, que este está construido completamente y que sus movimientos son limpios y correctos. En caso de que se percate de que en un determinado momento hay un hueso que quedó mal animado se puede retroceder el video hasta el lugar donde se cometió el error y corregirlo sin afectar los valores capturados en los frames anteriores y posteriores a los implicados en el error. Por este método quedan algunos frames intermedios sin recibir los datos, es por esto que la aplicación de forma automática realizará una interpolación entre el último frame que recibió coordenadas y el actual.

2.5.3.2 Pose a Pose

El método de captura Pose a Pose es basado en el método más utilizado por los que utilizan el 3Dmax para realizar animaciones, ya que tiene idéntico nombre. Estos animadores definen solamente las posiciones importantes, y luego el 3Dmax se encarga de interpolar los frames intermedios, los frames que contienen estas poses importantes se denominan Keyframes. En

SisMa se hace de la misma manera, se lleva el video a las posiciones importantes, y con este detenido se lleva a cabo la captura de dichas posiciones, esto se realiza dando clic con el Mouse en la articulación a la que se le está haciendo la captura, automáticamente la aplicación coloca un marcador en dicha articulación e interpola entre este Keyframe y el anterior, esto posibilita una buena visualización de la animación en la ventana tridimensional.

Los datos se van procesando dinámicamente, lo que hace a la aplicación más eficiente porque solo ocupa la memoria justa para almacenar los mismos, además, esto hace que el usuario se pueda despreocupar de muchos aspectos necesarios como la interpolación entre los puntos capturados y otros que la aplicación va realizando simultáneamente con la captura de las coordenadas.

De esta manera se logra llevar el movimiento del video, que está en el plano, a coordenadas espaciales, construyendo de esta forma el esqueleto cargado del .ASE, pero ya con la animación realizada, es decir, con el movimiento que debe ejecutar cada hueso en cada momento, este es un esqueleto de muestra que construye la aplicación automáticamente para que el animador tenga una idea espacial de cómo va quedando su animación. Esto es a grandes rasgos el proceso de captura del movimiento.

2.5.4 Obtención del producto final

Hasta ahora se ha explicado el proceso de captura de movimiento que se realiza en la herramienta que se propone en este trabajo, pero como se dijo anteriormente, aunque la herramienta es el 80% de la solución, la obtención del producto final se realiza en el 3Dmax ya que en él se corrigen los errores que se cometieron durante el proceso de captura, estos errores son debido a que en ocasiones algunas articulaciones del actor virtual en el video se pierden de vista porque son tapadas por otras partes del cuerpo, esto está en dependencia del movimiento

que esté realizando este. Lo antes dicho trae como consecuencia que el animador, en esos momentos, cometa algunos errores ya que al perder la articulación por espacio de algunos segundos (para el caso del método continuo), debe seguir una trayectoria aproximada, introduciendo errores inevitablemente, los cuales son mucho más evidente por el método Continuo. Por esta razón se propone el 3Dmax para corregir estos errores y obtener la animación.

Se debe decir que el problema de la introducción de errores será superado en versiones posteriores, ya que la solución está en realizar el proceso de captura con más cámaras, para tener más vistas del movimiento del actor virtual, logrando así la visualización de todas las partes de su cuerpo, en esta primera versión solo se utilizarán dos cámaras.

SisMa entre sus opciones, brindará la de exportar al formato .csm, como ya se explicó, este formato es uno de los que utiliza el 3Dmax para realizar animaciones, en él se construye el esqueleto a base de marcadores. Para lograr este formato se necesita realizar una serie de cálculos para obtenerlo a partir del esqueleto en formato .ASE con los valores capturados, es decir, llevar los huesos a marcadores. Para darle el acabado final, este fichero .csm exportado desde SisMa se carga en el 3Dmax y se visualiza la animación con un esqueleto real, ya que el 3Dmax es más ilustrativo en ese aspecto. Durante el proceso de la visualización de la animación .csm se podrán ver las deficiencias que hayan quedado durante la captura y entonces ahí, en el 3Dmax, se podrán corregir estas deficiencias obteniendo como producto final una animación de gran calidad y en muy poco tiempo. En el caso que la captura se haya hecho por el método de Pose a Pose, además de la animación se debe dar al animador cuáles son los Keyframes de esta para que este solo tenga que rectificar estos, los demás son borrados para que el 3Dmax los interpole.

2.6 SisMa vs. métodos actuales

Antes que nada se debe salvar la distancia entre SisMa y las herramientas utilizadas en el mundo con el mismo fin. Estas herramientas forman parte de soluciones muy costosas, por todos los accesorios que se han mencionado anteriormente. Los métodos que se utilizan a nivel mundial requieren de una gran cantidad de recursos, ya que necesitan de varias cámaras y trajes reflectores en el caso de los sistemas ópticos. Por otra parte se tienen los magnéticos que también necesitan recursos como receptores y un transmisor, y por último se tienen los electro – mecánicos que aunque algunos son menos costosos que los mencionados anteriormente, también requieren de un gran andamiaje de dispositivos para llevar a cabo la captura del movimiento. A pesar de todo lo anterior se debe decir que en estos sistemas se logran animaciones de muy buena calidad.

Hablando de forma general, las principales desventajas de estos sistemas son la oclusión de los marcadores en el caso de los ópticos, la salida del actor del área de captura, en ocasiones el sistema confunde la trayectoria de los marcadores, estos marcadores algunas veces se corren de lugar y esto trae consigo movimientos falsos.

Seguidamente se dirá cómo se trabaja en SisMa los aspectos antes mencionados. En el caso de SisMa, esta herramienta se encargará de poner los marcadores donde el animador indica en el video, es decir, los errores que se puedan cometer de corrimiento de marcadores, dependen totalmente del animador, estos no se corren por sí solos. SisMa se puede decir que no es una herramienta de captura de movimiento en tiempo real, por lo que no hay peligro de que el actor se salga del área de captura, ya que solamente se trabaja con los videos de este, la salida del actor de esta área sí puede ocurrir, pero esto sería durante la grabación de los videos. En cuanto al problema de una posible confusión de la trayectoria de los marcadores, esto en SisMa no ocurre debido a que la captura es manual. Lo ideal sería desarrollar en un futuro un sistema que

realice la captura en tiempo real sin necesidad de hacerla de forma manual, es decir, que el sistema se encargue de todo.

En cuanto a costo es en donde más se ve la diferencia de la solución propuesta con los demás sistemas, ya que reduce el costo significativamente porque solo utiliza dos cámaras y una aplicación, esto no ocurre con los demás métodos ya que son extremadamente costosos. Los sistemas de captura de movimiento tienen en estos momentos precios muy altos en el mundo, los mismos oscilan entre \$20 000 y \$200 000, en cambio, el costo de SisMa sería mucho menor, ya que solamente requiere de dos cámaras de video con una velocidad de muestreo igual o mayor de 60 fr/s, que por muy costosas que sean las mismas, jamás llegarán a los precios antes mencionados. [18][19][20][21][22]

En el proceso que sigue a la captura no hay grandes diferencias. Los sistemas actuales después de la captura tienen que pasar la animación por un proceso de edición, para darle el acabado y eliminar los posibles errores que hayan quedado. En el caso de SisMa también habría que pasar la animación creada por un proceso de rectificación, ya que en ocasiones algunas partes presentarán ruido o algún hueso no siga la trayectoria exacta que se desea, sino que presenta alguna que otra desviación.

Con esta solución no se pretende superar a las existentes en el mundo, sino que lo que se persigue es brindar a la UCI y al país un método eficaz y rápido para realizar captura de movimiento. Es bueno recalcar que la principal ventaja de SisMa sobre los métodos actuales es su bajísimo costo, y que a pesar del mismo se logran animaciones de buena calidad, lo que hace de este sistema una solución perfecta para un país como Cuba. Además de que podría ser el comienzo para en un futuro próximo desarrollar un sistema de captura de movimiento que se encuentre a la altura de los existentes en el mundo en cuanto a calidad del producto que se logra en él y a las facilidades que podría brindar a sus usuarios.

2.7 Modelo de dominio

A continuación se muestra en la Figura 2 el diagrama de dominio del sistema SisMa. Un modelo de dominio captura los tipos más importantes de objetos en el contexto del sistema. Los objetos del dominio representan las “cosas” que existen o los eventos que suceden en el entorno en el que trabaja el sistema.

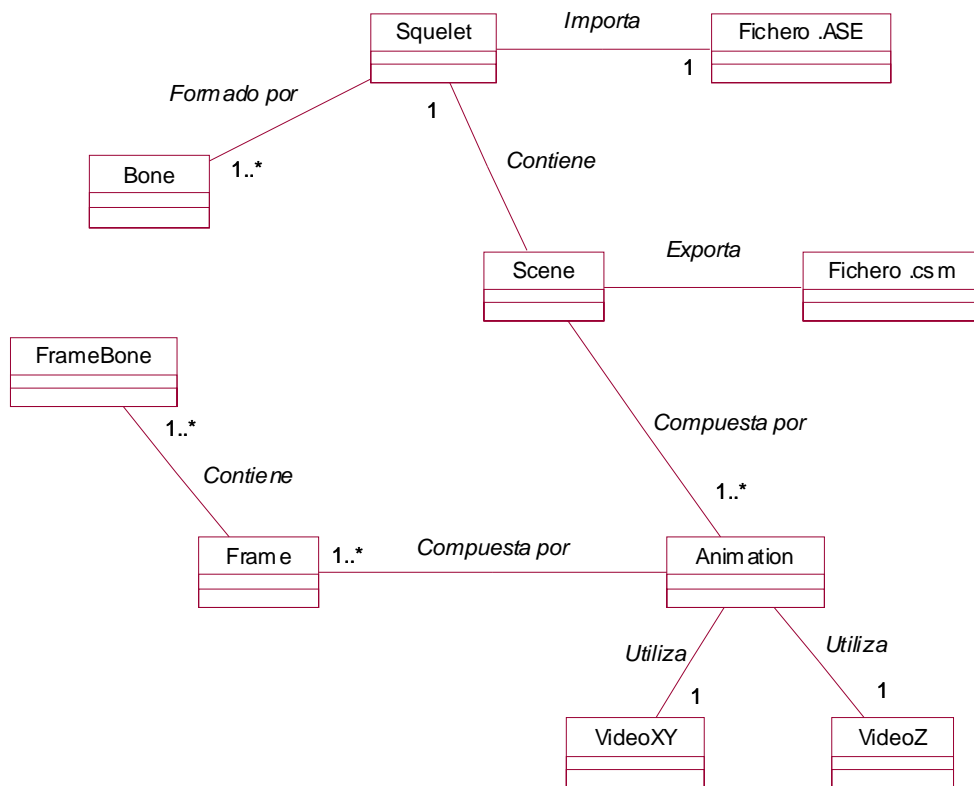


Figura 2: Modelo de dominio del sistema

Conceptos importantes

- Scene: es la escena que se crea en el sistema SisMa, la cuál está formada por 1 o más animaciones.
- Animation: es la captura de movimiento realizada a un esqueleto.
- Squelet: esqueleto al cuál se le realiza la animación.
- Bone: es un hueso del esqueleto, este está formado por varios huesos.
- Frame: la animación se divide en frames, en cada frame estará los datos de todo el esqueleto para ese momento, y la sucesión rápida de estos frames es lo que permitirá visualizar la animación.
- FrameBone: es exactamente lo mismo que el Bone, pero la diferencia radica en que este almacena los datos capturados para un hueso en específico, mientras que el Bone solo tiene nombre y otros datos, es decir, que el Bone es el hueso antes del proceso de captura, y el FrameBone después.
- VideoXY: es el video que se utilizará en la captura de movimiento para obtener los valores de X y Y.
- VideoZ: es el video que se utilizará en la captura de movimiento para obtener los valores de Z.
- Fichero .ASE: es el fichero que se importa desde el 3Dmax y que contiene la estructura del esqueleto que se desea animar.
- Fichero .csm: es el fichero de captura de movimiento hacia el que exporta el sistema con la captura realizada, para que posteriormente sea cargado en 3Dmax.

2.8 Requisitos funcionales.

Conociendo los objetivos que se han planteado en este trabajo, se enumerarán los requisitos funcionales, es decir, las acciones que debe realizar el sistema para satisfacer los objetivos que

se persiguen. Dentro de estas acciones se tienen tanto las que deben ser ejecutadas por el usuario como las que debe realizar el sistema internamente. El sistema debe ser capaz de:

- R1. Cargar una escena.
- R2. Mostrar mensaje de advertencia.
- R3. Eliminar escena.
- R4. Salvar una escena.
- R5. Cargar esqueleto.
- R6. Importar fichero .ASE.
- R7. Cargar animación.
- R8. Cargar videos.
- R9. Salvar animación.
- R10. Capturar puntos.
- R11. Interpolar coordenadas.
- R12. Exportar animación a csm.
- R13. Crear nueva animación.
- R14. Variar offset.
- R15. Seleccionar animación.
- R16. Manipular video.
- R17. Borrar animación.
- R18. Variar velocidad del video.
- R19. Seleccionar hueso.

2.9 Requisitos no funcionales.

- **Usabilidad:** Los futuros usuarios del sistema serán animadores. El producto debe estar concebido para que el usuario piense en qué desea hacer y no cómo hacerlo, por lo que éste requerimiento debe estar presente en alto grado en el producto final.

- **Soporte:** En una versión inicial deberá ser compatible con la plataforma Windows, pero debe estar preparado para que con rápidas modificaciones pueda migrar hacia Linux.
- **Legales:** Se registrará por las normas ISO 9000.
- **Software:** Sistema operativo Windows.
- **Hardware:** Requerirá de computadoras Pentium 3, con 256 de memoria RAM y 1.5 GHz.
- **Diseño e implementación:** Debe utilizar transparentemente la biblioteca gráfica OpenGL. Se harán llamadas a esta biblioteca desde Lenguaje C++. Se registrará por la filosofía de Programación Orientada a Objetos.

2.10 Especificación de Casos de Uso.

En este epígrafe se realiza una explicación detallada de los principales casos de uso del sistema. Esto se realizará utilizando el UML. Se capturarán los requisitos funcionales y se mostrarán en un diagrama de casos de uso. Un caso de uso es una descripción narrativa de un proceso de dominio. Los casos de uso son la base del diseño del software, ya que a partir de ellos se derivan muchos de los posteriores procesos dentro del desarrollo del software.

CU - 1	Cargar escena
Actor	Usuario
Descripción	El usuario necesita cargar una escena que había hecho anteriormente para observarla o modificarla.
Referencia	R1,R2,R3,R4,R5,R6,R7,R8

CU - 2	Salvar escena
Actor	Usuario
Descripción	El usuario ha terminado de realizar una escena o modificarla y desea salvarla para su posterior uso.
Referencia	R4,R10

CU - 3	Capturar puntos
Actor	Usuario
Descripción	El usuario da clic en el video encima de la articulación a la cual desea capturarle el movimiento, el sistema verifica si el método de captura indicado es el Continuo o el Pose a Pose y entonces este debe capturar las coordenadas que va recorriendo el Mouse, debe interpolarlas, llevarlas a coordenadas espaciales y guardarlas para su posterior procesamiento.
Referencia	R10,R11

CU - 4	Exportar animación a csm
Actor	Usuario
Descripción	El usuario tiene una animación terminada y desea exportarla a un formato determinado para su posterior uso o modificación.
Referencia	R13

CU - 5	Crear nueva animación
Actor	Usuario

Descripción	El usuario necesita montar todo lo necesario en el sistema para comenzar a construir su animación.
Referencia	R2,R14,R5,R6,R8

CU - 6	Cargar esqueleto
Actor	Usuario
Descripción	El sistema carga un esqueleto que importa desde un fichero .ASE para llevar a cabo su animación.
Referencia	R2,R5

CU - 7	Cargar video
Actor	Usuario
Descripción	El sistema monta los videos especificados por el usuario para más adelante utilizarlos como modelos para la animación..
Referencia	R2,R8

CU - 8	Variar offset
Actor	Usuario
Descripción	El usuario varía el offset para lograr que el actor realice el movimiento sincronizado en ambos videos.
Referencia	R14

CU - 9	Seleccionar animación
Actor	Usuario
Descripción	El usuario desea seleccionar otra animación dentro de la misma escena para visualizarla y continuar trabajando en ella.
Referencia	R8,R15

CU - 10	Manipular video
Actor	Usuario
Descripción	El usuario manipula el video utilizando todas las funcionalidades de un reproductor.
Referencia	R2, R16

CU - 11	Borrar animación
Actor	Usuario
Descripción	El usuario desea borrar una animación de la escena.
Referencia	R2, R17

CU - 12	Variar velocidad del video
Actor	Usuario
Descripción	El usuario desea variar la velocidad del video.
Referencia	R18

CU - 13	Seleccionar hueso
Actor	Usuario
Descripción	El usuario desea seleccionar un hueso para ver sus datos.
Referencia	R19

En el Anexo 3 se puede ver el Diagrama de Casos de Uso, en este caso, como ya se ha explicado, hay un solo actor que sería el usuario, por las características de este sistema. Se podrá observar de manera más clara cuál es la función de cada caso de uso, además de ver cuál es la relación entre ellos.

Los casos de uso Exportar animación a csm, Capturar puntos, Salvar escena, Crear nueva animación, Cargar escena, Variar offset, Manipular video, Seleccionar animación y Borrar animación son ejecutados directamente por el actor. Pero los casos de uso Cargar esqueleto y Cargar video son llamados desde otros casos de uso. Como se ve en el diagrama, Cargar video es una inclusión de los casos de uso Cargar escena y Crear nueva animación, ya que cada vez que estos se ejecutan necesitan ejecutarlo para cumplir su objetivo. Pero Cargar video es a la vez un extensión de Borrar animación y Seleccionar animación, ya que en ocasiones estos casos de uso necesitarán de él para concretarse. Por otra parte se tiene a Cargar esqueleto que está incluido en Cargar escena y es una extensión de Crear nueva animación, esto es de esta manera porque cada vez que se cargue una escena se necesitará cargar un esqueleto, pero no siempre que se vaya a crear una animación es necesario cargar un esqueleto.

2.11 Expansión de los Casos de Uso

Caso de Uso:	Cargar esqueleto
Actor:	Usuario
Propósito:	Cargar esqueleto importado desde .ASE.

Resumen:	El sistema carga el esqueleto que posteriormente será animado.
Referencias:	R2,R5
Precondiciones:	No debe haber ningún esqueleto cargado en el sistema.
Acción del actor	Respuesta del sistema
1. El usuario indica cargar o crear escena.	1.1.Se verifica que el fichero sea el correcto. 1.2El sistema crea un nuevo esqueleto. 1.3.Guarda este esqueleto en el Kernel. 1.4.Este esqueleto carga la información de sus huesos.
Cursos alternativos	
Acción 1	1.1.Si el fichero no es el correcto se muestra un mensaje de advertencia y se termina el caso de uso.

Caso de Uso:	Cargar videos
Actor:	Usuario
Propósito:	Cargar los videos.
Resumen:	El sistema carga los videos que posteriormente serán utilizados como modelos en la animación.
Referencias:	R2,R8
Precondiciones:	Debe haber cargada al menos una animación y el esqueleto.
Acción del actor	Respuesta del sistema
1. El usuario indica cargar o crear escena.	1.1. Se asigna al primer video de la animación el nombre del fichero del video correspondiente. 1.2.Se asigna al segundo video de la animación el nombre del fichero del video correspondiente. 1.3.Se redimensiona el arreglo de frames de la animación a la cantidad de frames del video más largo entre los 2 que se cargaron.
Cursos alternativos	

Acción 1	1.1.Si el fichero no es el correcto se muestra un mensaje de advertencia y se termina el caso de uso.
-----------------	---

Caso de Uso:	Cargar escena
Actor:	Usuario
Propósito:	Cargar escena en el sistema.
Resumen:	El usuario tiene guardada una animación que había realizado anteriormente y desea cargarla para su visualización o modificación.
Referencias:	R1,R2,R3,R4,R5,R6,R7,R8
Precondiciones:	El usuario debe tener una animación hecha anteriormente en este sistema y guardada.
Postcondiciones:	Se carga una escena.
Acción del actor	Respuesta del sistema
1.El usuario va al menú e indica cargar un escena	<p>1.1. El sistema verifica que el fichero sea el correcto.</p> <p>1.2.Verifica que no haya ninguna escena cargada.</p> <p>1.3. Se hace referencia al caso de uso Cargar Esqueleto.</p> <p>1.4. Se crea una nueva animación.</p> <p>1.5. Se adiciona la animación a la lista de animaciones del Kernel.</p> <p>1.6. Se coloca la nueva animación como activa.</p> <p>1.7. Se hace referencia al caso de uso Cargar Video.</p> <p>1.8Se asigna al offset de la animación el valor del offset cargado.</p> <p>1.9Se toma el arreglo de frames de la animación y se redimensiona a la cantidad de frames que presenta la misma.</p> <p>1.10Se crea un nuevo frame.</p> <p>1.11Se guarda el frame en la posición correspondiente dentro del arreglo de frames de la animación.</p>

	<p>1.12. Se toma la cantidad de huesos del esqueleto y se redimensiona el arreglo de FrameBones del Frame a esta cantidad.</p> <p>1.13. Se crea un vector de ángulos eulerianos.</p> <p>1.14 Se crea un vector de posición.</p> <p>1.15. Se guardan las coordenadas X, Y y Z en el vector creado.</p> <p>1.16. Se guardan las rotaciones en X, Y y Z en el vector de Euler.</p> <p>1.17. Se guarda el vector de Euler y el vector de posición en el FrameBone correspondiente del Frame correspondiente.</p>
Cursos alternativos	
Acción1:	<p>1.1 Si el fichero no es el correcto se muestra mensaje advirtiendo al usuario y termina el caso de uso.</p> <p>1.2 Si hay alguna escena cargada:</p> <p>1.2.1 Se borran de la interfaz los datos de la escena anterior.</p> <p>1.2.2 Se borra el esqueleto del Kernel.</p> <p>1.2.3 Se borra la lista de animaciones.</p> <p>1.2.4 Se continúa en el 1.3.</p>

Caso de Uso:	Salvar escena
Actor:	Usuario
Propósito:	Salvar una escena realizada en el sistema
Resumen:	El usuario tiene una escena cargada en el sistema, a la cual le está realizando modificaciones o la está elaborando completamente, y cuando lo estime conveniente o lo necesite la salva.
Referencias:	R4,R10
Precondiciones:	El usuario debe tener una escena cargada en el sistema para poder salvarla.

Postcondiciones:	Se salva una escena.
Acción del actor	Respuesta del sistema
1.El usuario va al menú e indica salvar una escena.	1.1. El Kernel salva el nombre del esqueleto. 1.2. El Kernel salva la cantidad de huesos del esqueleto. 1.3. El Kernel salva la cantidad de animaciones de la escena. 1.4. Salva el nombre de la animación. 1.5. Salva el nombre de los videos. 1.6. Salva el valor del offset. 1.7. Salva la cantidad de frames que tiene. 1.8. Recorre todos los FrameBones y los salva guardando todos los datos de la posición y la rotación.

Caso de Uso:	Capturar puntos
Actor:	Usuario
Propósito:	Durante la construcción de la escena ir capturando los puntos simultáneamente.
Resumen:	El usuario está elaborando una escena, da clic en el video sobre la articulación a la que le desea capturar los movimientos y a partir de ahí comienza la captura.
Referencias:	R10,R11
Precondiciones:	Debe haber una animación con su video de guía cargada en el sistema.
Postcondiciones:	Se capturan y guardan las coordenadas recogidas por el Mouse del actor.
Acción del actor	Respuesta del sistema
1. El actor da clic en el video sobre la articulación a la que	1.1. El sistema verifica si la animación es Pose a Pose. 1.2. El sistema busca dentro de la lista de huesos del esqueleto, al hueso

le desea capturar los datos.	<p>que se está animando.</p> <p>1.3. Verifica que el frame actual exista.</p> <p>1.4.Crea un nuevo frame y redimensiona su arreglo de FrameBones a la cantidad de huesos del esqueleto.</p> <p>1.5.Crea un vector.</p> <p>1.6. Verifica qué video es el que se está utilizando(el de XY o el de la Z).</p> <p>1.7. Coloca los valores capturados en el frame y hueso correspondientes.</p> <p>1.8. Interpola entre el último punto que guardó el sistema y el anterior.</p>
Cursos alternativos	
Acción 1:	<p>1.1 Si la animación es Pose a Pose, entre los pasos 1.4 y 1.5 se ejecuta otro en el que se coloca el frame creado como Keyframe.</p> <p>1.3 Si el frame actual existe se pasa a 1.5.</p>

Caso de Uso:	Exportar animación a csm
Actor:	Usuario
Propósito:	Exportar la animación al formato csm.
Resumen:	El usuario salva la animación al formato csm para poder trabajar con esta en otras herramientas, con el objetivo de perfeccionarla.
Referencias:	R13
Precondiciones:	Debe haber una animación cargada en el sistema y terminada parcial o totalmente.
Acción del actor	Respuesta del sistema
1. El usuario va al menú e indica exportar una animación.	<p>1.1.El Kernel escribe el nombre del fichero que se está exportando.</p> <p>1.2.La animación escribe el primero y el último frame de ella.</p> <p>1.3.La animación escribe el rate(la cantidad de cuadros por segundo a los que corre ella).</p>

	<p>1.3.La animación coloca los nombres de cada uno de los marcadores del esqueleto en el formato csm.</p> <p>1.4.En el frame, utilizando los puntos capturados, se calcula la posición de cada uno de los marcadores de un csm.</p> <p>1.5.El frame se encarga de escribir en el fichero todos los datos calculados anteriormente.</p>
--	--

Caso de Uso:	Crear nueva animación
Actor:	Usuario
Propósito:	Montar todo el sistema para comenzar a animar.
Resumen:	El usuario va a crear una nueva animación, para esto debe seleccionar los video y el esqueleto que va a utilizar para generar la animación.
Referencias:	R2,R14,R5,R6,R8
Precondiciones:	
Acción del actor	Respuesta del sistema
<p>1. El usuario va al menú e indica crear una nueva animación.</p> <p>2. El usuario selecciona el esqueleto y los dos videos que va a utilizar en su</p>	<p>1.1.El sistema abre la ventana a través de la cual se va a cargar todo lo necesario para crear la animación.</p> <p>2.1.El sistema verifica que los ficheros del esqueleto y los videos sean los correctos.</p> <p>2.2.Se crea una nueva animación.</p> <p>2.3.Se adiciona la animación a la lista de animaciones.</p>

animación e indica crear animación.	<p>2.4. Se coloca como activa.</p> <p>2.5. Se verifica que no haya ningún esqueleto cargado en la aplicación.</p> <p>2.6. Se hace referencia al caso de uso Cargar Esqueleto.</p> <p>2.7. Se hace referencia al caso de uso Cargar Videos.</p>
Cursos alternativos	
Acción 2:	<p>2.1 Si alguno de los ficheros no es el correcto se muestra mensaje advirtiendo al usuario y termina el caso de uso.</p> <p>2.5. Si ya hay un esqueleto cargado en el sistema producto de una animación anterior se continúa en el 2.7.</p>

Caso de uso:	Variar offset
Actor:	Usuario
Propósito:	Variar el valor del offset de los videos.
Resumen:	El usuario necesita variar el offset de los videos para que los movimientos del actor virtual se sincronice en ambos videos.
Referencias:	R14
Precondiciones:	
Acción del actor:	Respuesta del sistema:
1.El usuario necesita aumentar o disminuir el valor del offset para sincronizar los videos.	<p>1.1 El sistema ejecuta algunas de las acciones siguientes:</p> <p>a) Si el usuario desea aumentar el offset ir a la sección Aumentar offset.</p> <p>b) Si el usuario desea disminuir el offset ir a la sección Disminuir offset.</p>
Sección “Aumentar	

offset”	
Acción del actor:	Respuesta del sistema:
2. El actor aumenta el offset.	<p>2.1 El sistema aumenta en una unidad el valor del offset y lo muestra en la interfaz.</p> <p>2.2 Se guarda el nuevo valor del offset en la animación activa.</p> <p>2.3 Se redimensiona el arreglo de frames a la cantidad sacada de la suma de su longitud actual mas el nuevo valor del offset.</p>
Sección “Disminuir offset”	
Acción del actor:	Respuesta del sistema:
3. El actor disminuye el offset	<p>2.1 El sistema disminuye en una unidad el valor del offset y lo muestra en la interfaz.</p> <p>2.2 Se guarda el nuevo valor del offset en la animación activa.</p>

Caso de uso:	Manipular video
Actor:	Usuario
Propósito:	Manipular los videos utilizando las funcionalidades del reproductor.
Resumen:	El usuario necesita utilizar todas las funcionalidades del reproductor para que sea más fácil el trabajo a la hora de manipular los videos para realizar la captura del movimiento grabado en estos.
Referencias:	R16, R2
Precondiciones:	
Acción del actor:	Respuesta del sistema:
1. El actor necesita manipular el video	1.1 El sistema puede ejecutar algunas de las acciones siguientes:

para mejorar y agilizar el proceso de captura del movimiento.	<ul style="list-style-type: none"> a) Si el usuario desea presionar Play para echar a andar los videos ir a la sección Play. b) Si el usuario desea presionar Pause para detener los videos ir a la sección Pause. c) Si el usuario desea presionar Stop para detener la reproducción de los videos ir a la sección Stop. d) Si el usuario desea presionar Skip- para llevar los videos al inicio ir a la sección Skip-. e) Si el usuario desea presionar Skip+ para llevar los videos al final ir a la sección Skip +. f) Si el usuario desea presionar Rewind para llevar los videos algunos frames hacia el inicio ir a la sección Rewimd. g) Si el usuario desea presionar Forward para llevar los videos algunos frames hacia atrás ir a la sección Forward.
Sección "Play"	
Acción del actor:	Respuesta del sistema:
2. El actor desea echar a andar los videos.	<ul style="list-style-type: none"> 2.1 El sistema verifica que haya alguna animación activa. 2.2 El sistema verifica que el video correspondiente a la pantalla que está activa se encuentre en la animación. 2.3 Habilita el Timer.
Cursos alternativos	
Acción 2	<ul style="list-style-type: none"> 2.1 Si no hay ninguna animación activa se termina el caso de uso. 2.2 Si el video correspondiente a la pantalla que está activa no se encuentra en la animación se termina el caso de uso.
Sección "Pause"	
Acción del actor:	Respuesta del sistema:
3. El actor desea	3.1 El sistema deshabilita el Timer.

detener los videos.	
Sección “Stop”	
Acción del actor:	Respuesta del sistema:
4. El usuario desea terminar la reproducción de los videos.	4.1 El sistema lleva la posición de los videos a cero. 4.2 El sistema deshabilita el Timer.
Sección “Skip-”	
Acción del actor:	Respuesta del sistema:
5. El actor desea llevar los videos al inicio.	5.1 El sistema lleva la posición de los videos a cero.
Sección “Skip+”	
Acción del actor:	Respuesta del sistema:
6. El actor desea llevar los videos al final.	6.1 El sistema lleva la posición de los videos a la longitud del video más largo de los dos.
Sección “Rewind”	
Acción del actor:	Respuesta del sistema:
7. El actor desea llevar el video unos frames hacia el inicio.	7.1 El sistema verifica que la posición actual del video menos trescientos sea mayor que cero. 7.2 El sistema verifica que la posición del video sea diferente de cero. 7.3 El sistema verifica que la posición del video sea igual a cero.
Cursos alternativos	
Acción 7	7.1 Si la posición del video menos trescientos es menor que cero se lleva la posición a cero y se termina el caso de uso. 7.2 Si la posición del video es diferente de cero se disminuye en

	<p>trecientos dicha posición y se termina el caso de uso.</p> <p>7.3 Si la posición del video es cero se muestra un cartel advirtiendo que ya se ha arribado al inicio del video y se termina el caso de uso.</p>
Sección “Forward”	
Acción del actor:	Respuesta del sistema:
8. El actor desea llevar el video unos frames hacia el final.	<p>8.1 El sistema aumenta la posición del video en trescientos.</p> <p>8.2 Se verifica que el video no haya llegado a su longitud total.</p>
Cursos alternativos	
Acción 8	8.2 Si el video llegó a su longitud total se muestra un mensaje advirtiendo que se ha llegado al final del video y que ya no se puede continuar. Aquí termina el caso de uso.

Caso de Uso:	Seleccionar animación
Actor:	Usuario
Propósito:	Seleccionar una animación de la escena.
Resumen:	El usuario desea seleccionar una animación para visualizarla y continuar trabajando en ella o para revisarla.
Referencias:	R8,R15
Precondiciones:	
Acción del actor	Respuesta del sistema
1. El usuario selecciona la nueva animación	<p>1.1 El sistema lleva la posición del video a cero.</p> <p>1.2 El sistema busca en la lista de animaciones la animación seleccionada.</p> <p>1.3 Se coloca la animación seleccionada como activa.</p> <p>1.4 Se verifica que el video correspondiente a la pantalla activa se</p>

	<p>encuentre en la animación activa.</p> <p>1.5 Se visualiza el video correspondiente a la pantalla activa.</p>
Cursos alternativos	
Acción 1	1.5 Si el video correspondiente a la pantalla activa no se encuentra en la animación activa se termina el caso de uso.

Caso de Uso:	Borrar animación
Actor:	Usuario
Propósito:	Borrar una animación de la escena.
Resumen:	El usuario desea borrar una animación de la escena.
Referencias:	R17, R2
Precondiciones:	
Acción del actor	Respuesta del sistema
1. El usuario desea borrar una animación.	<p>1.1 El sistema verifica que haya alguna animación cargada</p> <p>1.2 Se elimina la animación de la lista de animaciones.</p> <p>1.3 Se verifica que se haya eliminado dicha animación.</p> <p>1.4 Se limpia la animación activa.</p> <p>1.5 Se verifica si la lista de animaciones quedó vacía.</p> <p>1.6 Se coloca la primera animación de la lista como activa.</p> <p>1.7 Se verifica que el video correspondiente a la pantalla activa se encuentre en la animación activa.</p> <p>1.8 Se visualiza el video correspondiente a la pantalla activa.</p> <p>1.9 Se deshabilita el Timer.</p>
Cursos alternativos	

Acción 1	<p>1.1 Si no hay ninguna animación cargada se muestra un mensaje advirtiéndole que no hay ninguna animación en la lista y aquí termina el caso de uso.</p> <p>1.3 Si la animación no se encontró se muestra un mensaje advirtiéndole de que no se encontró ninguna animación con el nombre suministrado y aquí termina el caso de uso.</p> <p>1.5 Si la lista de animaciones quedó vacía se deshabilita el Timer y termina el caso de uso.</p>
----------	--

Caso de uso:	Variar velocidad del video
Actor:	Usuario
Propósito:	Variar la velocidad del video.
Resumen:	El usuario necesita variar la velocidad del video para facilitar la captura del movimiento con el mouse.
Referencias:	R18
Precondiciones:	
Acción del actor:	Respuesta del sistema:
1.El usuario necesita aumentar o disminuir la velocidad del video.	<p>1.1El sistema ejecuta algunas de las acciones siguientes:</p> <ul style="list-style-type: none"> a) Si el usuario desea aumentar la velocidad del video ir a la sección Aumentar velocidad del video. b) Si el usuario desea disminuir la velocidad del video ir a la sección Disminuir velocidad del video.
Sección “Aumentar velocidad del video”	

Acción del actor:	Respuesta del sistema:
2. El actor aumenta la velocidad del video.	2.1 El sistema verifica que la velocidad no pase de ochenta frames x segundo. 2.2 Se incrementa en una unidad el valor de la velocidad. 2.3 El nuevo valor de la velocidad se muestra en la interfaz.
Cursos alternativos	
Acción 2	2.1 Si la velocidad del sistema es mayor que ochenta frames x segundo, se muestra un mensaje advirtiéndole que la velocidad no puede sobrepasar este límite y se termina el caso de uso.
Sección “Disminuir velocidad del video”	
Acción del actor:	Respuesta del sistema:
3. El actor disminuye la velocidad del video.	2.1 El sistema verifica que la velocidad no sea inferior a un frame x segundo. 2.2 Se disminuye en una unidad el valor de la velocidad. 2.3 Se muestra el nuevo valor de la velocidad en la interfaz.
Cursos alternativos	
Acción 3	2.1 Si la velocidad es inferior a un frame x segundo se muestra un mensaje advirtiéndole que la velocidad no puede disminuir de ese límite, y se termina el caso de uso.

Caso de Uso:	Seleccionar hueso
Actor:	Usuario
Propósito:	Seleccionar un hueso en la jerarquía de huesos.
Resumen:	El usuario desea seleccionar un hueso para ver sus datos, identificar el

	marcador que corresponde a él en el video e identificarlo además en el esqueleto construido en la pantalla tridimensional.
Referencias:	R19
Precondiciones:	
Acción del actor	Respuesta del sistema
1. El usuario selecciona un hueso.	1.1 El sistema verifica que se haya seleccionado algún hueso. 1.2 El sistema busca el hueso seleccionado en su lista de huesos. 1.3 Pone el hueso como marcado. 1.4 Se muestra en la interfaz el nombre de dicho hueso. 1.5 Busca en la lista de marcadores el marcador correspondiente a este hueso y le asigna el color rojo.
Cursos alternativos	
Acción 1	1.1 Si no se seleccionó ningún hueso se termina el caso de uso.

2.12 Conclusiones

En el presente capítulo se hizo un esbozo detallado de la propuesta de solución que brinda este trabajo para cumplir con los objetivos planteados. Se ha analizado cada una de las partes que tiene la herramienta, los ficheros con los que trabaja, las facilidades que da. Se explicó cómo es el trabajo con ella, de qué manera maneja los datos. Se enumeraron los requisitos funcionales y los no funcionales, además de que se explicó de manera muy detallada cada uno de los casos de uso que se sacaron para satisfacer los requisitos funcionales. También se mostró el diagrama de casos de uso que da una visión más clara de cómo funcionarán los mismos.

Una parte clave dentro del capítulo es la comparación del método propuesto con los existentes en el mundo en la actualidad. Esta comparación ayuda a tener una idea más clara de qué es lo

que se está proponiendo y de cuan bueno puede ser respecto a lo que se utiliza en el mundo. Como se pudo ver, la solución que se está proponiendo cumple con los objetivos que se han planteado, y puede ser una buena salida para un país como Cuba, de pocos recursos, y lo más importante es que puede ayudar de manera muy significativa al desarrollo de la realidad virtual en la UCI, disminuyendo el tiempo de desarrollo de los entornos virtuales e imprimiéndoles más realismo.

3. Análisis y diseño

3.1 Introducción

En la elaboración de este software se trató de realizar un diseño lo más claro posible para su posterior programación. Era necesario que le brindara al usuario una interfaz amigable para que no fuera complicada la generación de las animaciones. Además, se tuvo muy en cuenta, su posibilidad de escalabilidad puesto que esto es solo una primera versión, que estará continuamente sujeta a cambios para mejorar su eficiencia, brindando más servicios al usuario y mejorando los ya existentes.

3.2 Diagramas de clases

En este epígrafe se verá todo lo referente a los diagramas de clases que componen el sistema. El diagrama de clases es uno de los elementos más importantes dentro de un proyecto de software, ya que brinda una visión bastante completa de todo el sistema, mostrando todas las clases con sus métodos y atributos, así como las relaciones entre estas. Un software se puede dividir por paquetes, ese es el caso del sistema que se propone en este trabajo.

El sistema propuesto consta de tres paquetes, el de Dominio, el de Presentación y el paquete CBuilder, aunque se utilizaron otros dos paquetes que son librerías externas, estos son el MathLibrary, para las operaciones con vectores, y el ASEFileLoader, para importar los ficheros con extensión .ASE, como se dijo anteriormente. En el paquete de Dominio se encuentran todas las clases del sistema realmente, es decir, las que ejecutan todas las operaciones que hacen que el sistema cumpla con los objetivos planteados. En el paquete de Presentación se tiene todo lo relacionado con interfaz, y en el paquete CBuilder se tienen los componentes más importantes de la herramienta Keys utilizada, que en este caso es el CBuilder6.

3.2.1 Diagrama de clases del paquete Dominio

Como se dijo anteriormente, este paquete tiene todas las clases relacionadas con las funcionalidades del sistema, se puede decir que es el paquete fundamental dentro del mismo, aunque no tendría razón de ser sin los demás paquetes.

El paquete Dominio está constituido por seis clases, y como ya se ha dicho, es prácticamente el motor que ejecuta la mayoría de las funcionalidades del sistema. En la Figura 1 del Anexo 4 se puede ver el Diagrama de clases del paquete Dominio.

3.2.2 Diagrama de clases del paquete Presentación

En la Figura 2 del Anexo 4 se muestra el diagrama de clases del paquete Presentación, como ya se dijo anteriormente, este paquete es el que tiene que ver con todo lo referente a interfaz. En el diagrama se ven los tres formularios con los que cuenta el sistema, que son PrincipalWindow, OffsetAdjust y CreateNewAnimation. Este paquete se compone de tres clases, pero en el diagrama se incluyeron además las clases TTimer y TMediaPlayer que pertenecen al paquete CBuilder, esto se hizo para una mayor comprensión del diagrama, ya que estas clases tienen relación con las clases del paquete Presentación.

3.2.3 Diagrama de clases del paquete CBuilder

Como se puede ver en la Figura 3 del Anexo 4, el paquete CBuilder es muy sencillo pero de una gran importancia, esto se verá más adelante, cuando se explique en detalle cada una de las clases del sistema. Este paquete está compuesto por los componentes más importantes entre los utilizados del CBuilder, que en este caso son el TMediaPlayer, el TTimer y el TForm.

3.2.4 Diagrama de clases del paquete MathLibrary

El paquete MathLibrary es el que posibilita la realización de operaciones con vectores, por lo que es un componente fundamental dentro del sistema. Está compuesto por la clase cVector3f y MathStandars. Se debe tener en cuenta que en este paquete solo se han representado las clases que son utilizadas en este trabajo. Su diagrama se puede ver en la Figura 5 del Anexo 4.

3.2.5 Diagrama de clases del paquete ASEFileLoader

El paquete ASEFileLoader es el que se utiliza para importar el fichero .ASE con el esqueleto que más tarde será animado. Al igual que en el paquete MathLibrary, solo se representará la clase que se utiliza. Su diagrama se puede ver en la Figura 4 del Anexo 4.

3.3 Diagrama de paquetes

El diagrama de paquetes lo que muestra es la relación entre todos los paquetes antes expuestos. Comprender esta relación es muy importante ya que es una forma de ver cómo funciona todo el sistema de una manera más general. Como se puede ver en la Figura 4, el paquete principal es el de Presentación, que aunque no es el más importante dentro del sistema, es el que se relaciona con el resto de los paquetes, y desde él se manejan todos los procesos del mismo.

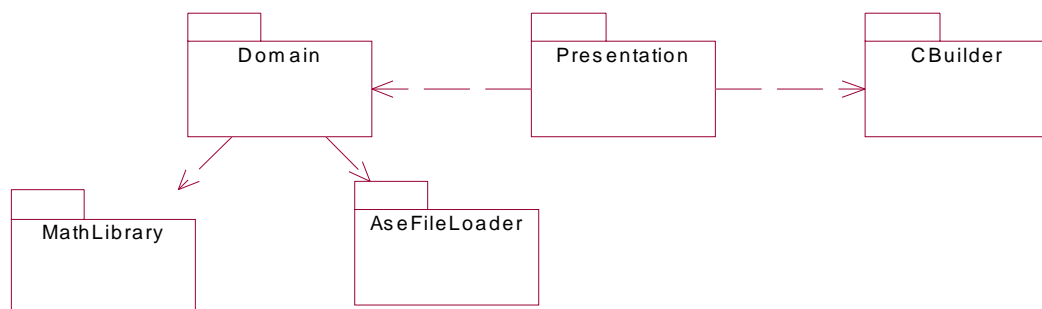


Figura 3: Diagrama de paquetes del sistema.

Seguidamente se dará una explicación detallada de cada clase del sistema, incluyendo sus atributos y métodos, para lograr una mayor comprensión del mismo.

3.4 Descripción de las clases

Nombre: cBone	
Tipo de clase	
Atributo	Tipo
1. iFather	Entero
2. Pos	cVector3f
3. Name.	AnsiString.
Para cada operación:	
Nombre:	Descripción:
1.getFather()	Proporciona el índice del hueso padre del Bone dentro del esqueleto.
2.setFather(int)	Modifica el índice del hueso padre del Bone dentro del esqueleto.
3.getPos()	Proporciona el vector que representa la posición del hueso.
4.setPos(cVector3f)	Modifica el vector que representa la posición del hueso.
7.getBoneName()	Proporciona el nombre del hueso.
8.setBoneName(AnsiString)	Modifica el nombre del hueso.

Nombre: cSquelet	
Tipo de clase	
Atributo	Tipo

1.FileName	String
2.ActiveBone	cBone
Para cada operación:	
Nombre:	Descripción:
1.setActiveBone(*cBone)	Guarda un hueso en el atributo ActiveBone, este atributo lo que indica es qué hueso es con el que se está trabajando, es decir, el hueso que está activo.
2.getActiveBone()	Proporciona el hueso que está activo.
3.setFileName(String)	Modifica el nombre del fichero desde el cual fue cargado el esqueleto.
4.getFileName()	Proporciona el nombre del fichero desde el cual fue cargado el esqueleto.
5.getBoneFather(int)	Recibe el índice de un hueso del esqueleto y devuelve el índice del padre de este hueso.
6.Load(AnsiString FileName)	Carga cada hueso del esqueleto hasta conformar el mismo, es la función encargada de importar el esqueleto desde el .ASE

Nombre: cFrameBone	
Tipo de clase	
Atributo	Tipo
1.VideoPosition	cVector3f
Para cada operación	
Nombre:	Descripción:
1.setVideoPosition(cVector3f)	Modifica el valor del vector que representa la posición del FrameBone, es decir, la posición del hueso después de animado, ya que esto es lo que representa el

	FrameBone.
2.getVideoPosition()	Proporciona el valor de la posición del FrameBone.

Nombre: cFrame	
Tipo de clase	
Atributo	Tipo
IsKeyframe	Boolean
Para cada operación:	
Nombre:	Descripción:
1.SaveFrame(char* Filename)	Salva el frame al fichero donde se está guardando la escena.
2.ExportFrame(char*Filename, cSquelet* Squelet)	Export a formato .csm el frame.

Nombre: cAnimation	
Tipo de clase	
Atributo	Tipo
1.AnimationName	String
2.VideoName1	String
3.VideoName2	String
4.LastFrame	Int
5.Offset	int
Para cada operación:	
Nombre:	Descripción:
1.setAnimationName(String)	Modifica el nombre de la animación.
2.getAnimationName()	Proporciona el nombre de la animación.
3.setVideo1(String)	Modifica el nombre del fichero desde el cual se

	cargó el video 1, es decir, el nombre del video 1.
4.getVideo1()	Proporciona el nombre del video 1.
5.setVideo2(String)	Modifica el nombre del fichero desde el cual se cargó el video 2, es decir, el nombre del video 2.
6.getVideo2()	Proporciona el nombre del video 2.
7.setLastFrame(int)	Durante la captura del movimiento con el Mouse modifica el valor del último frame en el cual hay coordenadas guardadas, para a partir de ese hasta el actual llevar a cabo la interpolación.
8.getLastFrame()	Proporciona el valor del último frame que se animó para a partir de él poder llevar a cabo la interpolación.
9.setOffset(int)	Modifica el valor del Offset de la animación para poder emparejar los videos a la hora de correrlos para animar.
10.getOffset()	Proporciona el valor del Offset de la animación.
11. InsertVideo(String Video,String Filename,int aFramesDimention)	Guarda en la animación el valor del video correspondiente y redimensiona el arreglo de frames a una longitud igual a la cantidad de frames que tenga el video.
12.SaveAnimation(char* Filename)	Salva la animación en el fichero que está en la dirección que se le pasa a esta función como parámetro, a partir de esta función se comienza y se termina de salvar la animación.
13. ExportAnimation(char* FileName)	Exporta a formato .csm la animación.

Nombre: cKernel
Tipo de clase

Atributo	Tipo
1.pSquelet	cSquelet
2.pActiveAnimation	cAnimation
Para cada operación:	
Nombre:	Descripción:
1.setSquelet(cSquelet*)	Guarda el esqueleto que se va a animar.
2.getSquelet()	Proporciona el esqueleto que se está animando.
3.setActiveAnimation(cAnimation*)	Guarda la animación que pasará a ser la animación activa, es decir, de las animaciones que componen la escena, la animación activa es en la que se está trabajando.
4.getActiveAnimation()	Proporciona la animación activa.
5.ResizeAPanels()	Redimensiona el arreglo de los marcadores que indican los huesos, a una longitud igual a la cantidad de huesos del esqueleto que se va a animar.
6.SaveScene(char* Filename)	A partir de esta función se comienza a salvar la escena, esta es salvada en el fichero que está en la dirección que recibe esta función como parámetro. Recorre el arreglo de las animaciones que componen la escena y va indicando a cada una que se salve hasta terminar con todas.
7.LoadScene(char* Filename)	Carga la escena desde el fichero que recibe como parámetro.
8.ExportAnimation(char*Filename)	Inicia el caso de uso exportar a formato .csm en el fichero que recibe como parámetro la animación activa.

Nombre: PrincipalWindow
Tipo de clase

Atributo	Tipo
1.Kernel	cKernel
2.positionVideo	int
Para cada operación:	
Nombre:	Descripción:
1.DrawChild(cFrameBone Father, cFrameBone Child, int FatherBone, int Frame, int ChildBone)	Va buscando cada hueso de forma recursiva para luego dibujarlos en la pantalla.
2.DrawRestBone(cVector3f Child, cVector3f Father, bool marked)	Dibuja un hueso en pantalla

En el caso de las demás clases no es necesario realizar una tabla como las anteriores para su explicación, ya que estas son componentes del sistema. La clase anterior se explicó en forma de tabla porque es el formulario principal y a través de él se gestiona todo el sistema, además de tener atributos y métodos que se le agregaron y que era necesario explicar.

Formulario CreateNewAnimation

Esta clase, como su nombre lo indica, representa al formulario donde se crea la animación. Este formulario dará las opciones de seleccionar los videos que se van a utilizar y el esqueleto, además de brindar la posibilidad de ponerle nombre a la animación.

Formulario OffsetAdjust

Esta clase representa al formulario donde se ajusta el offset de los videos en una animación. Como ya se ha dicho anteriormente, esto es necesario para sincronizar ambos videos. Este

formulario brinda la posibilidad de observar ambos videos al mismo tiempo y de variar el parámetro del offset, de esta manera se puede ver si los videos están sincronizados o no. Este formulario, además, posee un reproductor, al igual que el de el formulario principal, para poder manipular los videos y mejorar la calidad del trabajo.

TTimer

El TTimer es un componente Keys, pero es muy prudente su explicación ya que es decisivo en el buen funcionamiento del sistema. El TTimer es quien controla, como su nombre lo indica, el tiempo, lo que posibilita el buen funcionamiento y la sincronización de la aplicación. Este componente es el que se utiliza para controlar el video, desde su puesta en marcha hasta todas las demás operaciones que se realizan con el reproductor. Esto es muy importante ya que gracias al TTimer el video se reproduce con la velocidad adecuada, se le puede disminuir o aumentar esta velocidad(que es imprescindible a la hora de la captura del movimiento), y permite que el esqueleto construido en la ventana tridimensional esté sincronizado con el video. Se debe tener muy claro que sin el TTimer nada sería posible dentro del sistema, ya que es quien lleva las riendas de todos los procesos que en él se ejecutan.

TMediaPlayer

El TMediaPlayer es, al igual que el TTimer, un componente Keys. Este componente, aunque su función es muy pequeña, es clave en el sistema, ya que sin él no sería posible la visualización del video. El TMediaPlayer posee también todas las opciones que brinda un reproductor de video, pero en el caso de esta aplicación solo se utiliza la funcionalidad de visualizar el video, ya que todas las demás funcionalidades de un reproductor se decidió crearlas utilizando el TTimer, como ya se dijo anteriormente.

En este epígrafe se dio una detallada explicación de cada una de las clases del sistema, logrando una buena comprensión de las mismas se puede entender un por ciento elevado de cómo funciona el sistema y de qué rol cumplen dentro de él cada una de las cosas que lo componen

3.5 Diagramas de secuencia

Los diagramas de secuencia brindan un modelo de cómo interactúa el usuario con el sistema y de cómo interactúan los objetos del sistema entre sí. A continuación explicará el flujo de sucesos de cada uno de los diagramas de secuencia, los mismos se pueden ver en el Anexo 5.

3.5.1 Cargar escena

Flujo de sucesos

Este diagrama de secuencia se puede ver en la Figura 1 del anexo 5, en él se observa cómo se lleva a cabo el caso de uso Cargar escena. Primeramente se ve la orden que da el actor al sistema, que como es lógico es la de cargar el sistema. Para esto el usuario tuvo que seleccionar el fichero que quería cargar. Seguidamente el sistema llama a la función del Kernel que es la encargada de comenzar a ejecutar esta acción. Dentro de esta función lo primero que se hace es verificar si hay algún esqueleto cargado en el sistema, de ser así pues se borra, posteriormente se borra toda la lista de animaciones, más adelante se ejecuta el caso de uso Cargar esqueleto que es el encargado, como su nombre lo indica, de cargar el mismo, luego esta lista se llena con las animaciones de la nueva escena y para cada animación se guardan sus dos videos y el valor del offset. Luego, también para cada animación, se crea su arreglo de frames y se van creando frames y guardándolos en este arreglo, más tarde se llena este arreglo con los valores del esqueleto en cada uno de estos frames. Es bueno aclarar que a la vez que se crea cada frame, junto con este se crea un arreglo de FrameBones con todas sus posiciones en cero, esto no se pone en el diagrama para minimizarlo y porque además, esto ocurre de manera automática.

3.5.2 Salvar escena

Flujo de sucesos

El caso de uso Salvar escena es muy sencillo, como se puede ver en la Figura 3 del Anexo 5. Todo comienza por la orden de salvar la escena que da el usuario al sistema. El sistema lo primero que hace es llamar la función del Kernel que se encarga de este fin, para esto le pasa el fichero hacia el cual se va a salvar la escena. El Kernel recorre su lista de animaciones y va ordenando a cada una de estas animaciones que se salve. Las animaciones cuando son llamadas por el Kernel hacen lo mismo, recorren su lista de Frames y le van ordenando a cada uno de estos que se salve, y los frames realizan lo mismo cuando son llamados por la animación, recorren su lista de FrameBones y van salvando a cada uno de estos. En todos los casos se pasa el fichero hacia el cual se está salvando la animación. De esta forma se llega al final de este caso de uso.

3.5.3 Capturar puntos

Flujo de sucesos

Como se observa en la Figura 2 del anexo 5, para comenzar a ejecutar el caso de uso Capturar puntos se comienza por el movimiento del Mouse por encima del video, esto, como es lógico, lo ejecuta el actor. Cuando esto ocurre el sistema crea un frame, crea un vector con los valores capturados en ese momento, y lo guarda en el FrameBone correspondiente dentro del arreglo de FrameBones del Frame. Luego guarda este Frame en el arreglo de frames de la animación. Por último se realiza una interpolación entre el último frame que recibió datos y el actual. De esta forma queda concluido el caso de uso.

3.5.4 Crear nueva animación

Flujo de sucesos

El caso de uso Crear animación, como casi todos los casos de uso de este trabajo, comienza por la orden del actor. Cuando el actor da la orden de crear la animación el sistema lo primero que hace es crear una animación, luego la adiciona en la lista de animaciones que tiene el Kernel y la coloca como activa. Después de terminado este proceso se ejecuta el caso de uso cargar esqueleto, que como su nombre lo indica, lo que hace es cargar el mismo. Más adelante se guardan en la animación creada los dos videos seleccionados por el usuario, y de esta manera queda creada la animación. El diagrama de secuencia de este caso de uso se puede ver en la Figura 4 del Anexo 5.

3.5.5 Exportar animación a csm

Flujo de sucesos

La comprensión del caso de uso Exportar animación a csm es muy fácil. Comienza con la orden del actor y se llama a la función del Kernel que se encarga de exportar la animación. El Kernel, dentro de esta función, recorre su lista de animaciones y va llamando a la función exportar animación de cada una de estas. Las animaciones por su parte, recorren su lista de frames y van exportando a cada uno de estos, para esto se realizan internamente una serie de cálculos. Al final de este proceso el resultado es la animación exportada a csm. El diagrama de secuencia de este caso de uso se puede ver en la Figura 5 del Anexo 5.

3.5.6 Cargar esqueleto

Flujo de sucesos de Cargar esqueleto

El caso de uso Cargar esqueleto es bien sencillo, como se puede ver en el diagrama de secuencia de la Figura 6 del Anexo 5 lo que se hace es crear un esqueleto, luego se guarda este esqueleto en el Kernel, para más tarde llamar a la función dentro del esqueleto que se encarga de cargar los datos de cada uno de sus huesos, para esto importa un esqueleto con extensión .ASE desde 3Dmax. Es bueno aclarar que este caso de uso no es iniciado por el actor, comienza en el Kernel.

3.5.7 Cargar videos

Flujo de sucesos de Cargar videos

El caso de uso Cargar videos es muy sencillo también, y al igual que el de Cargar esqueleto este no es iniciado directamente por el actor, también comienza en el Kernel. Primeramente lo que se hace es guardar ambos videos en la animación correspondiente. Más adelante se redimensiona el arreglo de frames de dicha animación a la longitud del mayor de los videos, esto se hace a través de la función insertar videos. Luego de estos pasos quedan cargados los videos de la animación. Esto se puede ver con más claridad en el diagrama de secuencia que se encuentra en la Figura 7 del Anexo 5.

3.5.7 Seleccionar animación

Flujo de sucesos

El caso de uso Seleccionar animación consta de pocos pasos fundamentales. Este caso de uso es iniciado por una orden del actor, inmediatamente se busca en la lista de animaciones del Kernel la animación seleccionada, luego se pone esta como activa y se visualiza el video de esta

que coincida con la pantalla habilitada en el sistema. Para ver el diagrama de secuencia de este caso de uso se puede ir a la Figura 8 del Anexo 5.

3.5.8 Borrar animación

Flujo de sucesos

El caso de uso Borrar animación comienza, como se puede ver en el diagrama que se encuentra en la Figura 10 del Anexo 5, con la acción del actor. El sistema lo primero que hace es borrar la lista de animación correspondiente de la lista de animaciones que se encuentra en el Kernel, coloca la animación activa como nula. Luego coloca como animación activa a la primera animación de la lista. Por último visualiza los videos de la nueva animación activa y deshabilita el Timer para que estos videos no comiencen a correr sin la orden del actor.

3.5.9 Seleccionar hueso

Flujo de sucesos de Seleccionar hueso

Como se muestra en el diagrama de secuencia que se encuentra en la Figura 9 del anexo 5, el caso de uso Seleccionar hueso comienza por la acción del actor. Cuando el actor desea seleccionar un hueso lo primero que hace el sistema es buscar este hueso en el listado de huesos que tiene en el esqueleto, ya una vez encontrado este hueso, se busca en el listado de marcadores que está en el Kernel, al marcador correspondiente a este hueso. Cuando se tiene al marcador correcto se le asigna en la propiedad color, el rojo, esto es, como ya se dijo antes, para que sea identificado tanto entre los marcadores como entre los huesos del esqueleto construido en la pantalla tridimensional.

3.5.10 Otros casos de uso

A continuación se describirán los flujos de sucesos de los casos de uso Variar offset y Manipular video. A estos casos de uso no se les realizó el diagrama de secuencia debido a la gran sencillez de los mismos, aunque por sencillos no son menos importantes, ya que ambos ejecutan acciones fundamentales que ayudan en gran medida a que el sistema cumpla con los objetivos para los que fue diseñado.

3.5.10.1 Flujo de sucesos de Variar offset

Este caso de uso es el encargado de sincronizar los dos videos cargados en el sistema. Esto se logra aumentando o disminuyendo el valor del offset. Para esto, en caso de que se quiera aumentar el offset, lo que se hace es aumentar el valor del offset en una unidad, luego se guarda este valor en la animación que está activa en el Kernel y se redimensiona el arreglo de frames de dicha animación a la longitud que sale de la suma de la longitud actual de este arreglo más el nuevo valor del offset. Para disminuir este valor se realizan exactamente las mismas operaciones pero de manera contraria, se disminuye en uno el valor del offset, y se guarda en la animación activa, aunque en este último caso no se redimensiona el arreglo de frames.

3.5.10.2 Flujo de sucesos de Manipular video

Este caso de uso es el encargado de todo lo que tiene que ver con la manipulación del video, es decir, con el reproductor del video. Aquí se ejecuta el video, se detiene, se termina, se adelanta, etc. Todas estas operaciones se hacen utilizando el Timer, que no es más que un componente del CBuilder, que como se dijo antes, es la herramienta Keys seleccionada para la programación de este sistema. No tiene sentido explicar cómo es el trabajo con el Timer para lograr las acciones que realiza el reproductor ya que en la expansión de este caso de uso en el capítulo anterior esto está bastante claro. Así que para una mayor comprensión de este caso de uso se recomienda ir a su expansión en el capítulo dos.

3.5.10.3 Flujo de sucesos de Variar velocidad del video

Este caso es muy importante para lograr capturas de movimiento de buena calidad, ya que su función es variar la velocidad del video. Esto posibilita al usuario adaptar la velocidad del video a su gusto para que le sea más fácil seguir los movimientos del actor del video con el Mouse. Cuando el usuario disminuye la velocidad del video lo primero que se hace es verificar si la nueva velocidad no disminuye de un frame por segundo, de ser esto así pues se muestra un mensaje advirtiendo lo sucedido y se termina el caso de uso. De no suceder lo antes mencionado, se disminuye en uno la velocidad y este valor se muestra en la interfaz. Para el caso en que se desee aumentar la velocidad del video se realizan exactamente las mismas operaciones, la única diferencia es que lo que se verifica es que la velocidad no exceda los ochenta frames x segundo. Es bueno aclarar que aquí todo ocurre a nivel de interfaz, no interviene ninguna otra clase del sistema.

3.6 Conclusiones

Al terminar este capítulo se tiene el diseño del sistema a desarrollar, ya que se explica de forma detallada, con la ayuda de los diagramas de secuencia, cómo será la interacción entre las clases para el desarrollo de cada uno de los casos de uso. Además, se hizo el diagrama de clases para cada uno de los paquetes que conforman el sistema, con una descripción de cada una de estas, especificando cuál es la función de cada uno de sus atributos y métodos. Después de concluido el diseño, se puede pasar a la fase de implementación.

4. Implementación y prueba.

4.1 Introducción

En este capítulo se podrá ver todo lo referente a la implementación y las pruebas, ya que aquí se encuentran los diagramas de componente y los casos de prueba. Además, se muestran los resultados obtenidos en dichas pruebas, estos resultados son de mucha utilidad ya que la idea que se persigue no es solamente mostrar que el software desarrollado funciona, sino también que se cumplieron los objetivos propuestos inicialmente, que en resumen es lograr un método rápido y eficiente de realizar animaciones para ambientes tridimensionales en nuestro país.

4.2 Diagramas de componentes

En el caso del software desarrollado no es necesario realizar el diagrama de despliegue ya que solamente se necesita una computadora para llevar a cabo el trabajo. El sistema se dividió en cinco paquetes pero aquí solo se realizará el diagrama de componentes de cuatro de ellos ya que el restante no requiere de este diagrama debido a que es muy sencillo. Los paquetes que se expondrán en esta sección son el de “Dominio”, que es en el que se encuentran todos los ficheros que tienen que ver con la estructura y las operaciones que lleva a cabo la aplicación, el de “Presentación”, que es en el que están agrupados todos los ficheros relacionados con interfaz, el “MathLibrary”, en el cual se encuentra todo lo referente al trabajo con operaciones matemáticas, y ASEFileLoader, que es el que posee todo lo necesario para importar un fichero .ASE. A continuación se muestran los diagramas.

4.2.1 Dominio

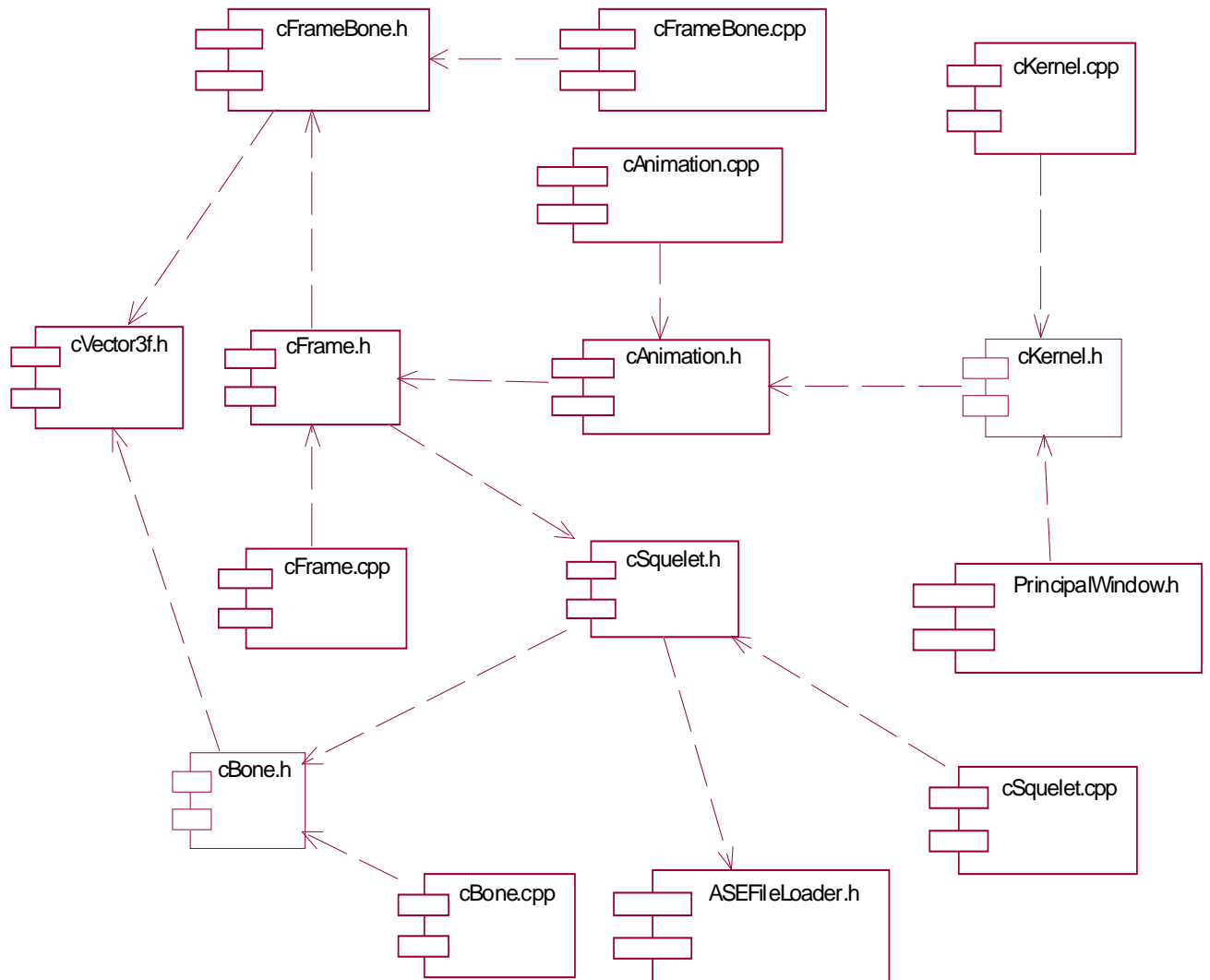


Figura 4: Diagrama de componentes del paquete Dominio.

4.2.2 Presentación

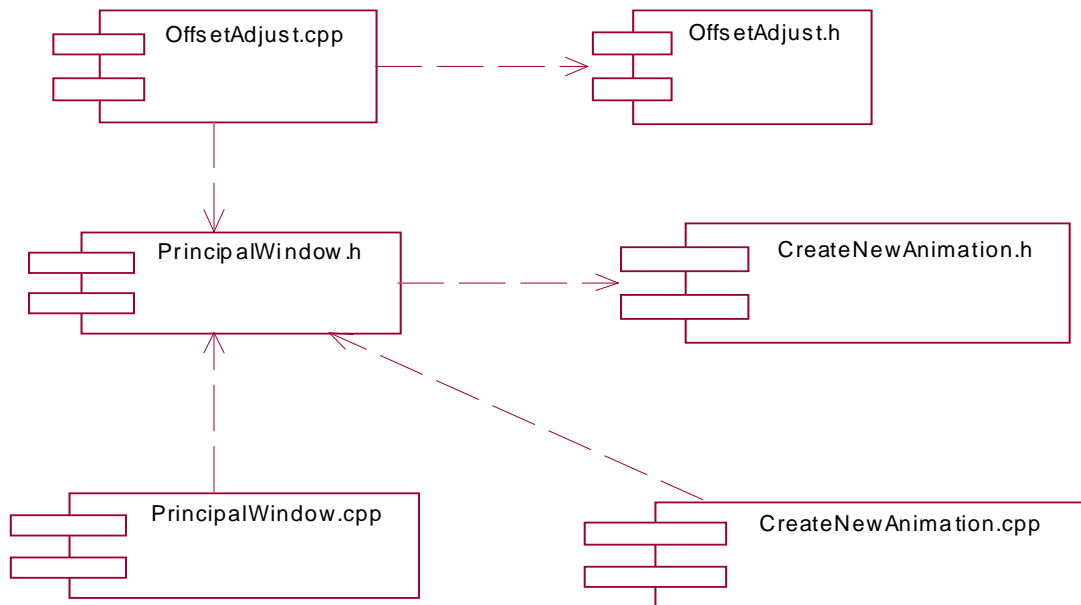


Figura 5: Diagrama de componentes del paquete Presentación.

4.2.3 MathLibrary

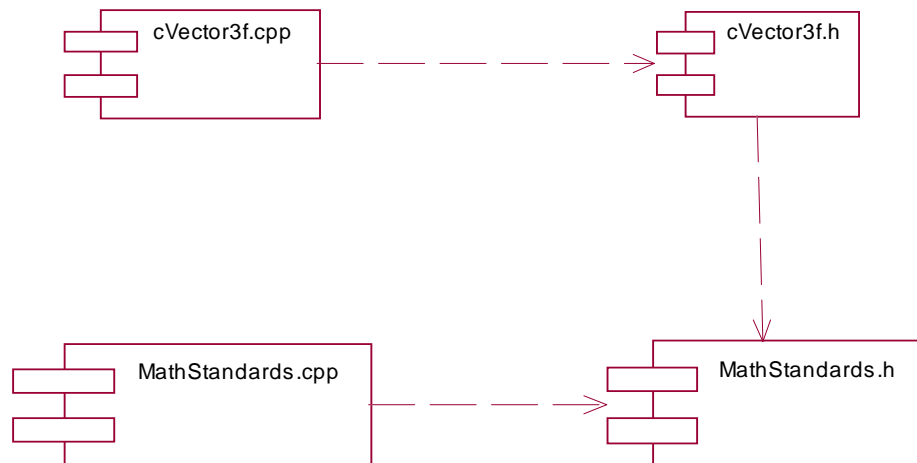


Figura 6: Diagrama de componentes del paquete MathLibrary.

4.2.4 ASEFileLoader

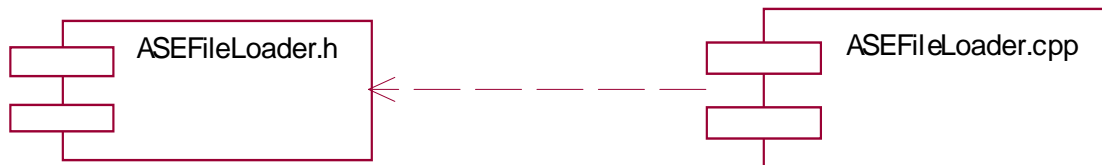


Figura 7: Diagrama de componentes del paquete ASEFileLoader.

4.3 Descripción de los casos de prueba

Caso de uso	Crear nueva animación
Caso de prueba	Crear nueva animación sin llenar ningún campo
Entrada	Se crea la animación sin seleccionar ningún campo.
Resultado	Se muestra mensaje advirtiendo la ausencia de todos los datos y no se crea la animación.
Condiciones	

Caso de uso	Crear nueva animación
Caso de prueba	Crear nueva animación sin seleccionar un esqueleto
Entrada	Se crea la animación seleccionando los videos solamente
Resultado	Se muestra mensaje advirtiendo la ausencia del esqueleto y no se crea la animación
Condiciones	

Caso de uso	Crear nueva animación
Caso de prueba	Crear nueva animación sin seleccionar el video XY

Entrada	Se crea la animación seleccionando el esqueleto y el video Z solamente.
Resultado	Se muestra mensaje advirtiendo la ausencia del video XY y no se crea la animación.
Condiciones	

Caso de uso	Crear nueva animación
Caso de prueba	Crear nueva animación sin seleccionar el video Z
Entrada	Se crea la animación seleccionando el esqueleto y el video XY solamente.
Resultado	Se muestra mensaje advirtiendo la ausencia del video Z y no se crea la animación.
Condiciones	

Caso de uso	Crear nueva animación
Caso de prueba	Crear nueva animación seleccionando todos los campos
Entrada	Se crea la animación seleccionando todos los campos.
Resultado	Se crea la animación sin ningún problema. En la ventana principal se muestra el nombre de la animación, los dos videos cargados y el esqueleto.
Condiciones	

Caso de uso	Capturar puntos
Caso de prueba	Capturar puntos sin seleccionar ningún hueso.
Entrada	Se comienza la captura del movimiento siguiendo este con el mouse.
Resultado	Se muestra mensaje advirtiendo que se debe seleccionar un

	hueso en el listado.
Condiciones	Debe haber al menos una animación creada en el sistema

Caso de uso	Capturar puntos
Caso de prueba	Capturar puntos seleccionando un hueso del esqueleto
Entrada	Se selecciona un hueso en el listado del esqueleto y se comienza la captura del movimiento siguiendo este con el mouse.
Resultado	En la ventana tridimensional se construye el hueso al que se le hace la captura y este sigue el movimiento capturado.
Condiciones	Deba haber una animación creada en el sistema.

Caso de uso	Seleccionar animación
Caso de prueba	Seleccionar animación
Entrada	Se crea una escena con más de una animación y se selecciona una de estas.
Resultado	Se muestran en la forma principal todos sus componentes, es decir, se montan sus videos en el reproductor, y el esqueleto de la pantalla tridimensional adquiere los movimientos de esta animación.
Condiciones	

Caso de uso	Seleccionar hueso
Caso de prueba	Seleccionar hueso en la jerarquía de huesos
Entrada	Se selecciona un hueso de la jerarquía de huesos del esqueleto.
Resultado	En el esqueleto construido en la pantalla tridimensional este hueso se colorea de rojo, lo mismo ocurre con el marcador que

	representa este hueso en el video.
Condiciones	Debe haber una animación cargada en el sistema con al menos un hueso que ya haya recibido la animación.

Caso de uso	Variar velocidad del video
Caso de prueba	Aumentar la velocidad del video
Entrada	Intentar aumentar la velocidad del video
Resultado	La velocidad del video aumenta
Condiciones	Debe haber al menos una animación cargada en el sistema.

Caso de uso	Variar velocidad del video
Caso de prueba	Disminuir la velocidad del video
Entrada	Intentar disminuir la velocidad del video.
Resultado	Disminuye la velocidad del video.
Condiciones	Debe haber al menos una animación cargada en el sistema.

Caso de uso	Variar velocidad del video
Caso de prueba	Aumentar la velocidad del video a más de 80 frames x segundo
Entrada	Se intenta aumentar la velocidad del video hasta tratar de pasar los 80 frames x segundo.
Resultado	Cuando llega a 80 frames x segundo no aumenta más este valor.
Condiciones	Debe haber al menos una animación cargada en el sistema.

Caso de uso	Variar velocidad del video
Caso de prueba	Disminuir la velocidad del video a menos de 1 frame x segundo
Entrada	Se intenta disminuir la velocidad del video hasta tratar de pasar la de 1 frame x segundo.

Resultado	Cuando llega a 1 frame x segundo no disminuye menos de este valor.
Condiciones	Debe haber al menos una animación cargada en el sistema.

Caso de uso	Variar offset
Caso de prueba	Aumentar y disminuir el valor del offset
Entrada	Se cargan varios pares de video y se trata de sincronizarlos aumentando y disminuyendo el offset.
Resultado	Los videos se sincronizan
Condiciones	Debe haber al menos una animación creada en la escena.

Caso de uso	Borra una animación
Caso de prueba	Borra una animación sin tener ninguna animación en la escena
Entrada	Se intenta borrar una animación
Resultado	No ocurre nada.
Condiciones	No debe haber ninguna animación en el sistema.

Caso de uso	Borra una animación
Caso de prueba	Borra una animación teniendo animaciones en la escena
Entrada	Se intenta borrar una animación
Resultado	Se borra la animación del sistema y se cargan todos los componentes de la primera animación de la lista, es decir, se muestran sus videos y sus demás datos.
Condiciones	No debe haber ninguna animación en el sistema.

En la tabla anterior se mostraron los principales casos de prueba a los que fue sometido el sistema. Este es un paso obligatorio dentro del proceso de desarrollo de un software, ya que las

pruebas son las que dicen si el sistema desarrollado cumple con los requerimientos funcionales propuestos.

4.4 Resultados obtenidos

Luego de ejecutar cada uno de los casos de prueba, se pasó a otra fase en la que se utilizó el software y se realizó una evaluación del producto obtenido en él, que en este caso es una captura de movimiento para más tarde generar a partir de ella una animación en el 3Dmax. Se debe recordar que aunque el software desarrollado es prácticamente un 80% de la solución, lo que se propone en este trabajo no es un software, sino un nuevo método de desarrollar animaciones para ambientes tridimensionales, y es por esta razón por lo que se realizaron además pruebas del tiempo en que demora todo el proceso y se comparó tanto en calidad como en el propio tiempo con el método actual que se utiliza en la UCI.

4.4.1 Resultados de la captura de movimiento

Para llevar a cabo el proceso de prueba del software en cuanto a la calidad del producto que se obtiene, se realizaron cinco capturas de movimiento en el mismo. Estas capturas fueron hechas por un individuo con un dominio bastante bueno de dicho software. Como se dijo en capítulos anteriores, el sistema consta de dos métodos de captura de movimiento, el Continuo y el de Pose a Pose. A continuación se muestran los resultados obtenidos, para esto se establece el tiempo en que se demoró en cada una de las capturas, además de que se comparan ambos métodos. El gráfico de la Figura 2 muestra el tiempo de duración de cada una de las capturas por cada uno de los métodos, así como el número de frames.

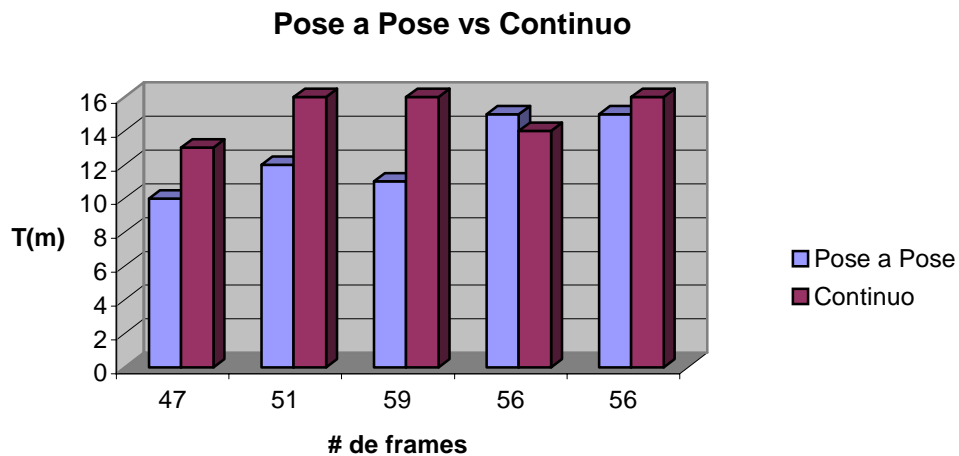


Figura 8: Gráfico que compara la animación realizada por el método Pose a Pose y el Continuo.

Como se puede apreciar en el gráfico anterior, el método Pose a Pose arrojó ligeramente mejores resultados que el Continuo, aunque esta situación pudiera revertirse para animaciones de una gran complejidad, ya que al ser más complejas se deben capturar más poses, pero la diferencia entre los dos métodos nunca debe ser significativa.

En cuanto a la calidad de las capturas, las que fueron hechas por el método Continuo inicialmente presentan una mayor calidad, esto se debe a que en este método se le aplica al esqueleto las coordenadas tal y como se capturan en cada frame. En el caso de la captura por el método Pose a Pose no sucede así, ya que lo que se hace es capturar el movimiento de las poses importantes, y los frames que se encuentran entre estas poses reciben sus coordenadas a través de una interpolación, esto, como es lógico, introducirá algunas pequeñas inexactitudes en cuanto a la trayectoria del movimiento de cada hueso del esqueleto. Lo que se quiere decir cuando se expresa que el Continuo tiene más calidad que el Pose a Pose es que a la hora de ser visualizada la captura en SisMa se ven movimientos más limpios, se aclara esto porque como se verá más adelante, a la hora de obtener el producto final esto no sucede así.

4.4.2 Resultados del producto final

Como se ha dicho antes, el producto no está terminado hasta que no se rectifica en el 3Dmax. Para realizar las pruebas de esta parte del método propuesto, se escogieron tres de las capturas de movimiento anteriores y se le dio la tarea a tres diseñadores diferentes, para que cada uno las rectificara en el 3Dmax, en todos los casos se les tomó el tiempo en que demoraron y se evaluó la calidad del producto obtenido. Estos resultados se compararon con los que obtuvieron estos mismos diseñadores desarrollando esas tres animaciones pero por el método que se ha utilizado en la UCI tradicionalmente, que es con el empleo del 3Dmax solamente. En el gráfico que se puede ver a continuación, aparecen los resultados de las pruebas antes mencionadas, se muestra el tiempo que demoró la captura del movimiento que se utilizó para cada una (barra más pequeña), y el tiempo invertido en realizar la animación completa, que sería el de la captura sumado con el de su arreglo en 3Dmax (barra mediana). Por otra parte se tiene además, el tiempo que se necesitó invertir para crear esas mismas animaciones solo con el 3Dmax (barra más grande).

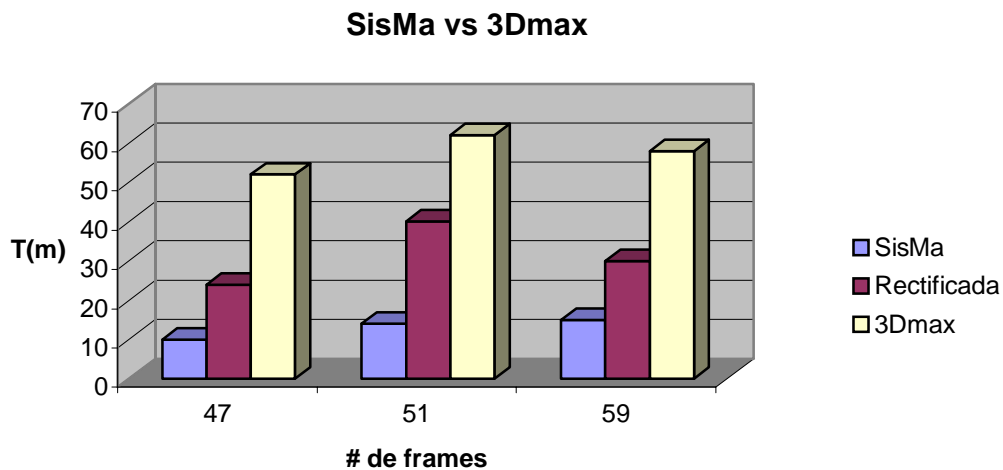


Figura 9: Gráfico que muestra un análisis comparativo entre las animaciones realizadas utilizando SisMa y 3Dmax, y las realizadas utilizando 3Dmax solamente.

Es bueno aclarar que el tiempo de captura que se utilizó para estas pruebas es el logrado por el método Pose a Pose, se hizo de esta forma debido a que el método Continuo se incluyó en el sistema con vistas a que los usuarios se familiaricen con el mismo, ya que aún no se logran resultados tan satisfactorios como por el método de Pose a Pose. En versiones posteriores se trabajará en el perfeccionamiento de este método con vistas a que sea menor el tiempo que demorará rectificar la captura. Los resultados que se han obtenido utilizando el método Continuo son muy parecidos a los que se lograron con la utilización del 3Dmax solamente. Pero en este caso la rectificación es un poco más incómoda para el animador ya que no hay poses importantes, es decir, debe rectificar en donde quiera que vea un problema, cuando se logre mejorar esto los resultados serán mucho mejores a los que se obtuvieron esta vez.

Como se puede observar en la tabla anterior, los resultados fueron muy satisfactorios, lográndose disminuir el tiempo de desarrollo de una animación aproximadamente a la mitad con respecto al que se emplea en la realización de esta solo con la utilización del 3Dmax. Las pruebas se realizaron con animaciones pequeñas porque era con las que se contaba en videos, pero para tener una idea de los resultados que se obtienen, con las animaciones anteriores bastaba, ya que estos resultados serían mucho más alentadores para animaciones de mayor duración.

Para valorar los resultados cualitativos y las ventajas que traía este método, se tomó la opinión de los animadores que trabajaron en estas pruebas. Todos coincidieron en que las animaciones que se obtenían presentaban una buena calidad ya que se logran movimientos muy reales debido a que estos son tomados de un actor real. En cuanto a las ventajas se llegó a la conclusión de que ayuda en gran medida a que la animación pueda contar con un buen desarrollo de los dos principios fundamentales que tiene que cumplir. Estos son el Timing y el Strong Pose, como ya se sabe, el Timing es el que rige cuánto tiempo demora el personaje en realizar un determinado movimiento, y el Strong Pose es el que logra que las poses importantes sean correctas, hacer que una animación cumpla con estos dos principios requiere de un trabajo bastante tedioso por parte del animador, utilizando SisMa esto no es así, ya que el animador se ahorra tiempo de trabajo y se quita de encima tener que dibujar todas las poses en un papel

antes de comenzar a desarrollar su animación. Todo lo antes dicho se obtiene con muy buena calidad en SisMa, ya que como se captura el movimiento a partir de un video de un actor real, queda muy bien delimitado el Timing y se logran poses casi perfectas.

4.5 Conclusiones

A modo de conclusión se puede decir que se logró un producto de buena calidad y con muchas posibilidades de mejorarse en futuras versiones. Se demostró su superioridad sobre el método actual en cuanto a tiempo de desarrollo y otros aspectos que se deben tener en cuenta a la hora de desarrollar una animación, además de que facilitó mucho el trabajo de los animadores. Con la utilización de SisMa se puede lograr una mayor rapidez en el desarrollo de proyectos que necesiten de animaciones tridimensionales.

Conclusiones generales

En el presente trabajo se cumplieron todos los objetivos planteados, ya que gracias al minucioso estudio que se realizó de herramientas similares, se llegó a una propuesta muy eficiente que fue apoyada por un software que se diseñó y desarrolló de tal manera que cumpliera con todos los requisitos funcionales que se establecieron para lograr el objetivo que se perseguía.

Se demostró la funcionalidad del método propuesto, llegando a resultados muy alentadores. Con este método se logró una disminución significativa del tiempo de desarrollo de animaciones para ambientes tridimensionales, lo cual contribuye a una disminución también del tiempo de desarrollo de entornos virtuales.

Se puede decir que el principal aporte de este trabajo es el desarrollo de una herramienta en la que se pueden realizar capturas de movimiento de buena calidad y en poco tiempo, ya que en la UCI no existe ningún sistema capaz de hacer esto, además de que no se tiene conocimiento de que exista otro en Cuba. En estos momentos solamente se ha implementado una primera versión para resolver los grandes problemas que se venían presentando en la parte de las animaciones debido a la demora que traía desarrollar las mismas sin que el animador tuviera una visión real del movimiento que debía imprimirle al personaje.

Por el diseño que se hizo de la herramienta antes mencionada, es muy fácil su escalabilidad, y gracias a esto se desarrollarán futuras versiones en las que mejorarán mucho más todas sus funcionalidades, además de que se le agregarán otras que traerán como consecuencia animaciones de mucha mejor calidad y en mucho menos tiempo.

Recomendaciones

En Cuba aún queda mucho camino por andar en el campo de la realidad virtual, en específico en el del desarrollo de animaciones tridimensionales, por esta razón se recomienda continuar con la investigación iniciada en este trabajo, con el objetivo de perfeccionar más el método propuesto, ya que se ha demostrado su eficiencia y gran utilidad. Se puede profundizar mucho más en facilitar el método de captura, e incluso se podría buscar nuevas formas de realizar esta tarea, además, se debe trabajar en mejorar la calidad de la captura que se hace en la herramienta propuesta con el fin de no tener que utilizar ninguna otra herramienta para rectificarla. Para esto se propone, entre las mejoras que se incluirán en el sistema en versiones posteriores:

- Brindar la posibilidad de realizar la captura con más de dos cámaras para de esta forma tener más ángulos de visión del actor y no presentar el problema de perder por instantes o en determinadas poses algunas partes del cuerpo del mismo.
- Trabajar en la mejora de la interpolación para lograr un movimiento más limpio en el esqueleto animado.
- Trabajar en la confección de un manual de usuario que contribuya con el buen desempeño del mismo a la hora de utilizar el sistema.
- Sustituir la captura manual que se realiza con el Mouse, por otro método más cómodo para el animador.
- Exportar la captura a otros ficheros para animaciones(BVH, BIP, etc).

Con todas las mejoras expuestas aquí y otras que se podrían ir agregando, se puede lograr en futuras versiones una herramienta que cumpla con la mayoría de las perspectivas de los que trabajan en el fascinante mundo de las animaciones tridimensionales.

Referencias bibliográficas

Libros

[5]. MAESTRI, George.

Character animation 2.

Volume 1: Essential techniques. New Riders Publishing, USA. 1999

[5.1]. Capítulo 6. Skeletons and mesh deformation.

Libros digitales

[2]. SUÁREZ, Pamela.

Animación y Visualización de Fenómenos Naturales.

http://mail.udlap.mx/~tesis/lis/suarez_r_pk/ (2004).

[2.1]. Capítulo III. Animación por computadoras.

http://www.pue.udlap.mx/~tesis/lis/suarez_r_pk/capitulo3.pdf

[4]. RUIZ, David y otros.

Animación en “Art of Illusion”.

<http://www.dccia.ua.es/dccia/inf/asignaturas/RG/trabajos/trabajo-david-ruiz.pdf>

[7]. LANDER, Jeff.

Skin Them Bones: Game Programming for the Web Generation.

Game Developer Magazine. USA. May, 1998.

<http://www.darwin3d.com/gamedev/articles/col0598.pdf> (2004)

Sitios web

[1]. David J. Sturman

Computer animation

[http://www.siggraph.org/education/materials/HyperGraph/animation/character_animation/motion_capture/history1.htm\(2000\)](http://www.siggraph.org/education/materials/HyperGraph/animation/character_animation/motion_capture/history1.htm(2000)).

[3]. Michael B. Comet

Character Animation: Principles and Practice

<http://www.comet-cartoons.com/3ddocs/charanim/>

[6]. PUTZ, Michael y HUFNAGL Klaus.

Character Animation for Real-time Applications

<http://www.cg.tuwien.ac.at/studentwork/CESCG/CESCG-2002/> (2004)

[8]. Biovision BVH

<http://www.cs.wisc.edu/graphics/Courses/cs-838-1999/Jeff/BVH.html>

[9]. Adam Kira, James F. O'Brien y David A. Forsyth

Skeletal Parameter Estimation from Optical Motion Capture Data

<http://www.cs.berkeley.edu/b-cam/Papers/Kirk-2004-SPE.pdf>

[10]. Motion Captor Real Time

http://www.metamotion.com/Motion_Captor_2004.pdf

[11]. James F. O'Brien, Robert E. Bodenheimer, Jr., Gabriel J. Brostow y Jessica K. Hodgins

Automatic Joint Parameter Estimation from Magnetic Motion Capture Data

<http://www.vuse.vanderbilt.edu/~bobbyb/pubs/ajpe00.pdf>

[12]. Gypsy Motion Captor

<http://www.animazoo.com>

[13]. Gypsy Gyro™ Motion Capture System

<http://www.metamotion.com/gypsy/gypsy-gyro.htm>

[14]. Facial Capture Applications

<http://www.ptiphoenix.com>

[15]. Face Trackers for Facial Motion Capture

<http://www.metamotion.com/hardware/face-trackers.htm>

[16]. Definition of the BIP configuration file format

<http://www.ens-lyon.fr/LIP/RESO/OldW3/BIP-manual/node6.html>

[17]. ASE File Format

http://www.unrealwiki.com/wiki/ASE_File_Format

[18]. Fritz Garrison

Maximizing Your Investment With Motion Capture Equipment

<http://www.ascension-tech.com/applications/pdf/maximize.pdf>

[19]. Siggraph 2004 Brings Sci-Fi Imagery Closer to Reality

http://www.tgdaily.com/2004/08/13/siggraph_2004_b_rings_sci/page4.html

- [20]. STT and Meta Motion Announce Powerful New Options for Motion Captor
<http://www.designertoday.com/tabindex-1/tabId-28/itemid-2494/DesktopDefault.aspx>
- [21]. Michaels Video Camera Shop
<http://www.michaels.com.au/video/videoc.htm>
- [22]. Kelkoo electronics
http://audiovisual.kelkoo.co.uk/b/a/cp_123501_filter_media_type_dvd.html
- [23]. Sistema de ayuda el 3Dmax 5.

Bibliografía consultada

- Avanced Graphics Programming Using OpenGL. Tom McReynolds David Blythe. Morgan Kaufmann publishers. 2005.

- George Maestri. *Character Animation 2, volumen 1 – essential techniques*. New Riders, 1999.

- Nik Lever. *Real-time 3D Character Animation with Visual C++*. Planta Tree, 2002.

- Rick Parent. *Computer Animation: Algorithms and Techniques*. Morgan Kaufmann Publishers, 1st edition, 2001.

- Colectivo de autores. *Game Programmin GEMS 4*. Charles River Media. 2003.

- Mathias Dekeyser, Karl Verfaillie y Jan Vanrie. *Creating stimuli for the study of biological-motion perception*. 2002.

<http://ppw.kuleuven.be/labexppsy/Dekeyseretal2002.pdf>

- Michael Gleicher Nicola Ferrier . *Evaluating Video-Based Motion Capture*. 2002.

<http://www.cs.wisc.edu/graphics/Papers/Gleicher/Mocap/videomocap.pdf>

- P. Fua, L. Herda, R. Plankers, y R. Boulic *Human Shape And Motion Recovery Using Animation Models*. 2002.

- Rick Parent. *Computer Animation: Algorithms and Techniques*. Morgan Kaufmann Publishers, 1st edition, 2001.

Glosario de términos

A:

Animación: Simulación de un movimiento creada por la muestra de una serie de imágenes o fotogramas.

Actor virtual: Persona que se encarga de realizar los movimientos que serán capturados en un sistema de captura de movimiento.

B:

Bibliotecas de movimientos: Almacén de los movimientos realizados por cada una de las partes del cuerpo del personaje para posteriormente lograr originar otros, producto de la combinación de estos.

C:

Captura de movimiento: Es el proceso de grabar un movimiento y trasladarlo a "data" convirtiéndola en representaciones matemáticas.

E:

Esqueleto virtual: Esqueleto elaborado en un software especializado en diseño de personajes para entornos virtuales.

F:

Frame: Cada una de las imágenes que componen una animación.

H:

Hueso padre: En una jerarquía de huesos se le llama así a los huesos de los cuáles depende el movimiento de otros, que serían sus hijos, ejemplo(brazo y antebrazo).

I:

Interpolación: Algoritmo matemático que a partir de varios puntos en el espacio, describe una función que contiene a los puntos intermedios.

J:

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

K:

Keyframe: Cualquier frame que contenga un aspecto especial de la animación, en este caso sería una pose(ver Pose) importante.

M:

Malla: Forma de representar un modelo a partir de polígonos. Colección de vértices, aristas y polígonos conectados de forma que cada arista es compartida como máximo por dos polígonos.

Marcador: Objeto que se coloca en las articulaciones del actor virtual y representa un hueso a la hora de capturar el movimiento.

P:

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, así como determinados roles (ver rol).

Pose: Posición que adopta un personaje en un determinado momento de una animación.

R:

Realidad virtual: La Realidad Virtual es un sistema interactivo que permite sintetizar un mundo tridimensional ficticio, creándote una ilusión de realidad.

Rol: Papel que trae consigo un grupo de acciones y que formará parte de los atributos de un personaje animado.

Rotoscoping: Técnica de animación que consiste en capturar un movimiento real, y utilizar esa información para mover un diseño generado por ordenador.

Rendering: Crear en forma automática una imagen de acuerdo al modelo tridimensional que existe en el ordenador.

Región de captura: Es la región dentro de la que el actor virtual se debe mover para que pueda llevarse a cabo la captura de movimiento.

Ruido: Anomalías que pudieran quedar en una animación que no es completamente correcta.

RAD: Rapid Application Development

S:

Stitching: Técnica de deformación de malla por huesos.

Skinning: Técnica de deformación de malla por huesos con múltiples pesos.

Set de receptores: Conjunto de receptores colocados en el cuerpo del actor virtual durante la captura de movimiento por el método magnético.

T:

Transmisor: Equipo que genera un campo magnético y percibe las variaciones de los sensores colocados en el cuerpo del actor virtual para la captura de movimiento por el método magnético.

Anexos

Anexo 1: Jerarquía de un esqueleto

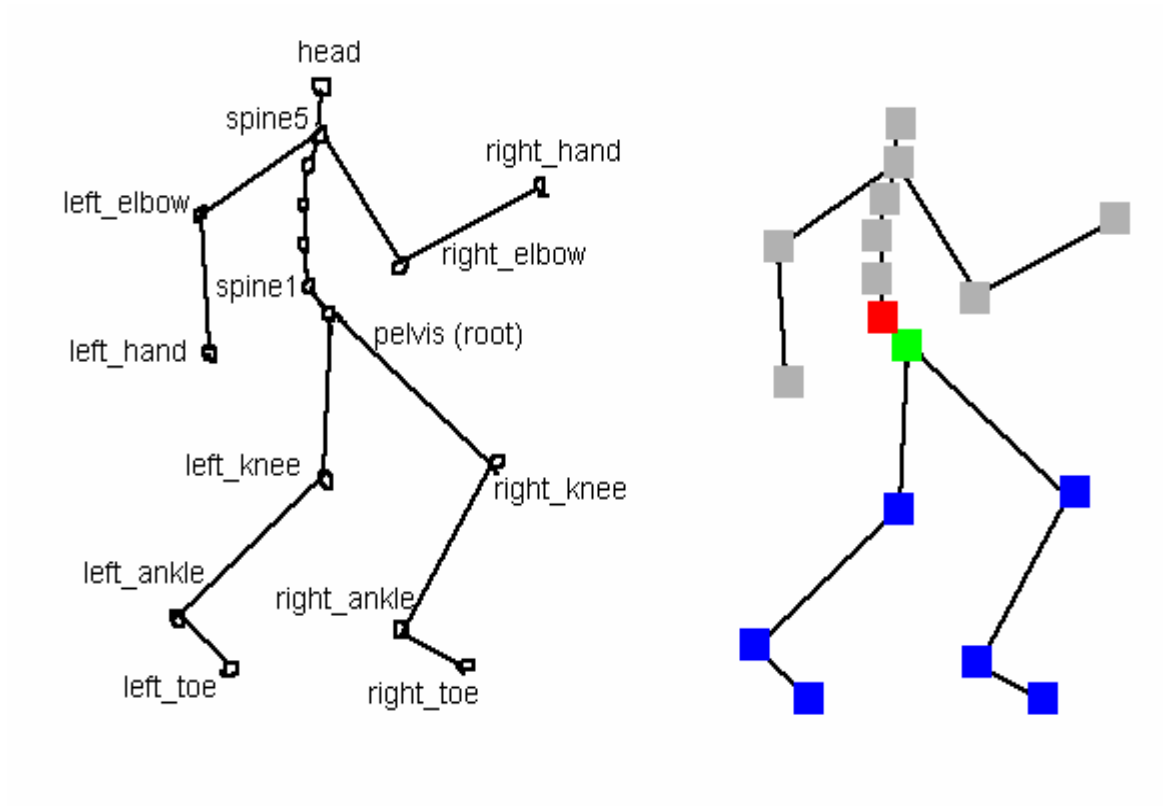


Figura 1: Jerarquía de un esqueleto.

Anexo 2: Sistemas de captura de movimiento.

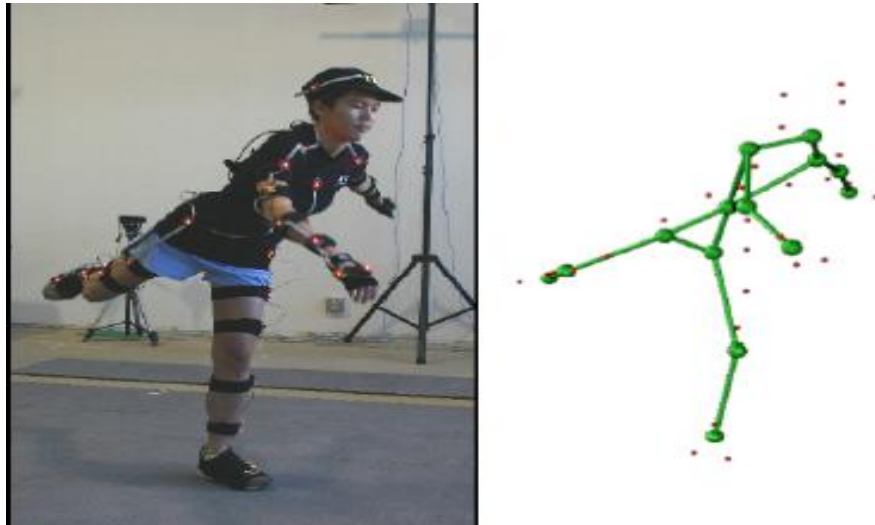


Figura 1: Sistema activo de captura de movimiento óptica.



Figura 2: Sistema pasivo de captura de movimiento óptica.



Figura 3: Captura del movimiento de dos personajes en un sistema pasivo de captura óptica.

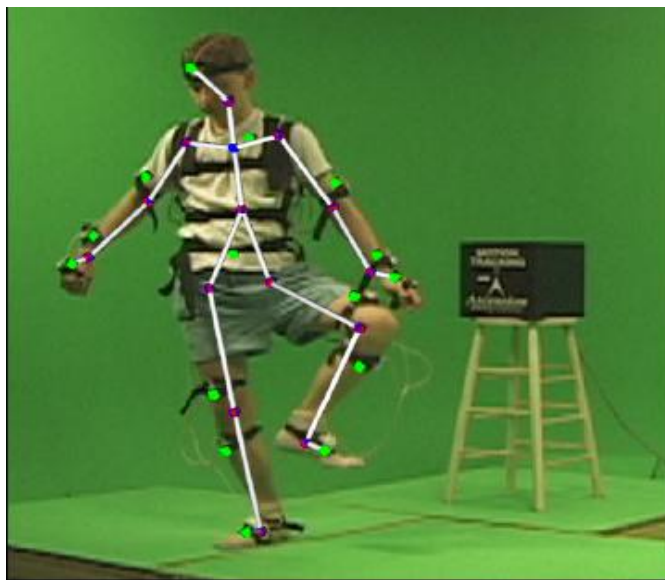


Figura 4: Sistema magnético de captura de movimiento.



Figura 5: Gipsy 4. Sistema mecánico de captura de movimiento.

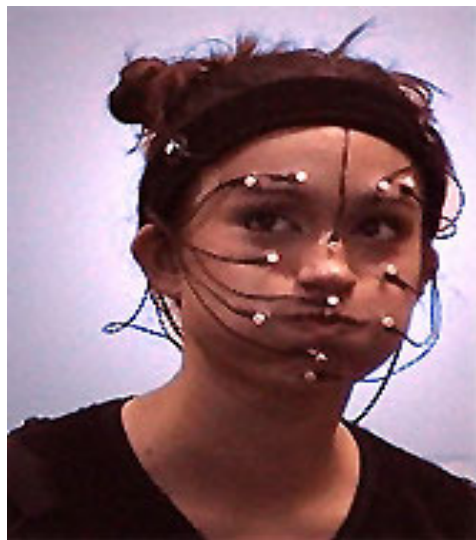


Figura 6: Sistema de captura de movimiento facial.

Anexo 3: Diagrama de Casos de Uso

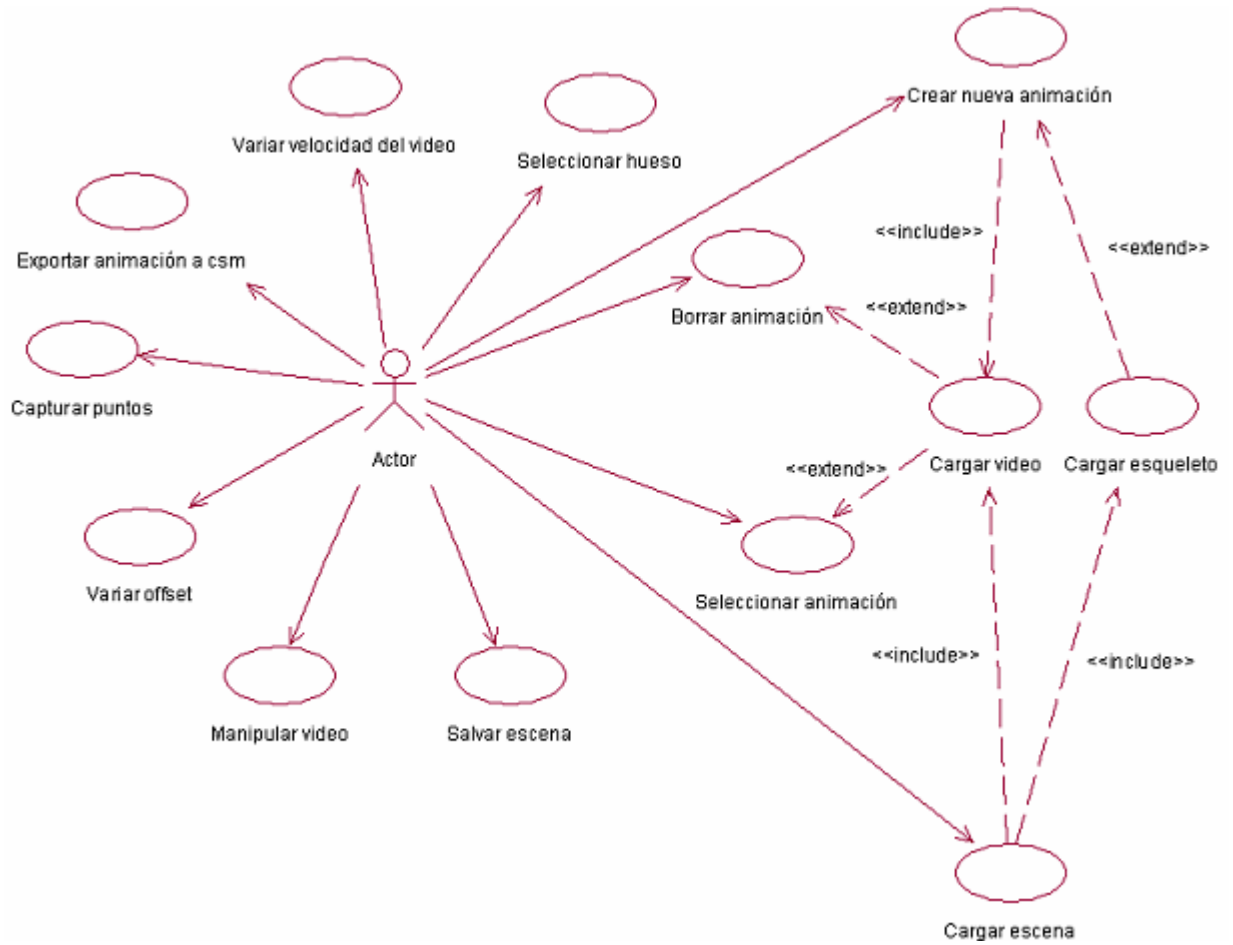


Figura 1: Diagrama de Casos de Uso

Anexo 4: Diagramas de clases

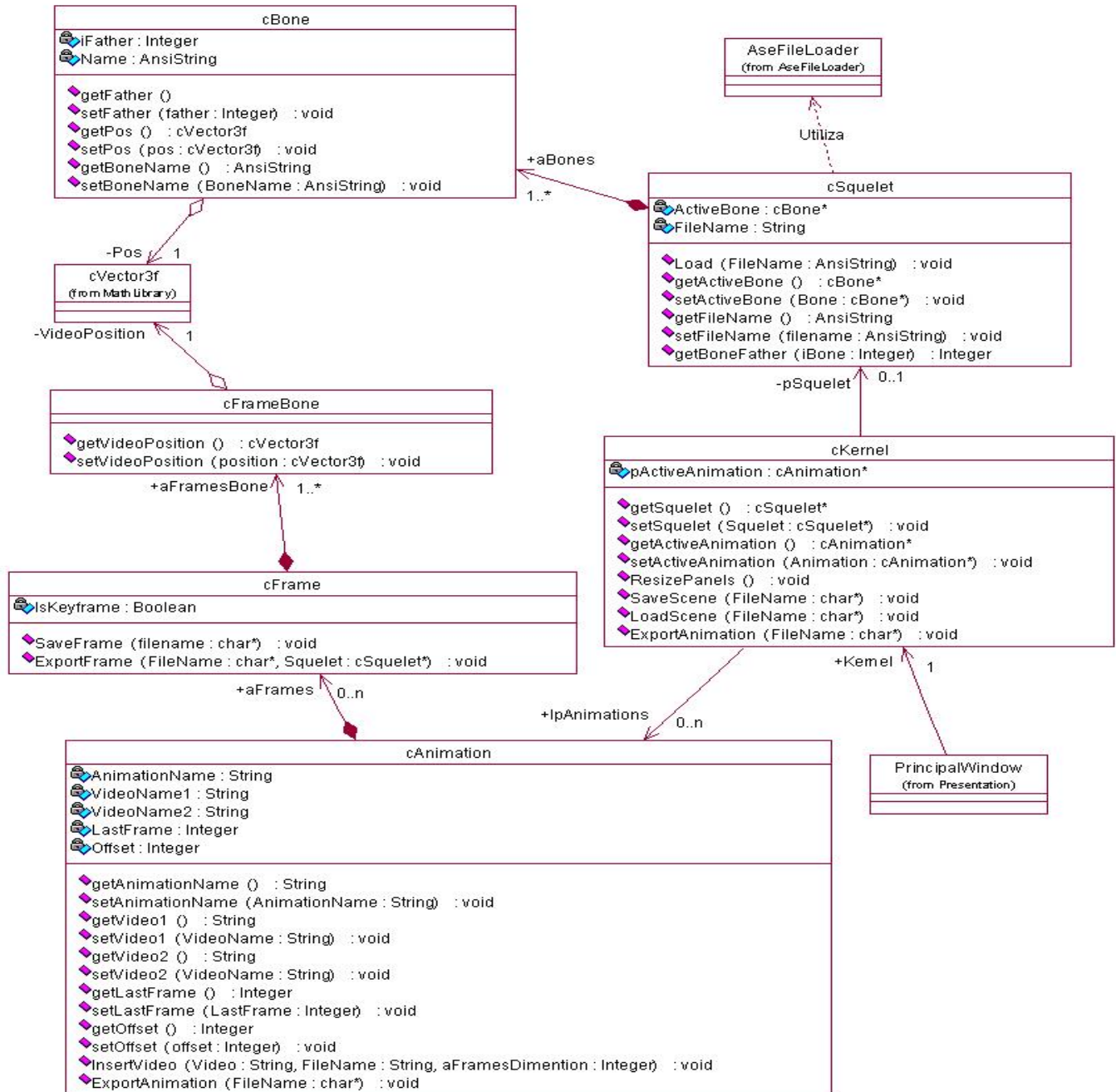


Figura 1: Diagrama de clases del paquete Dominio

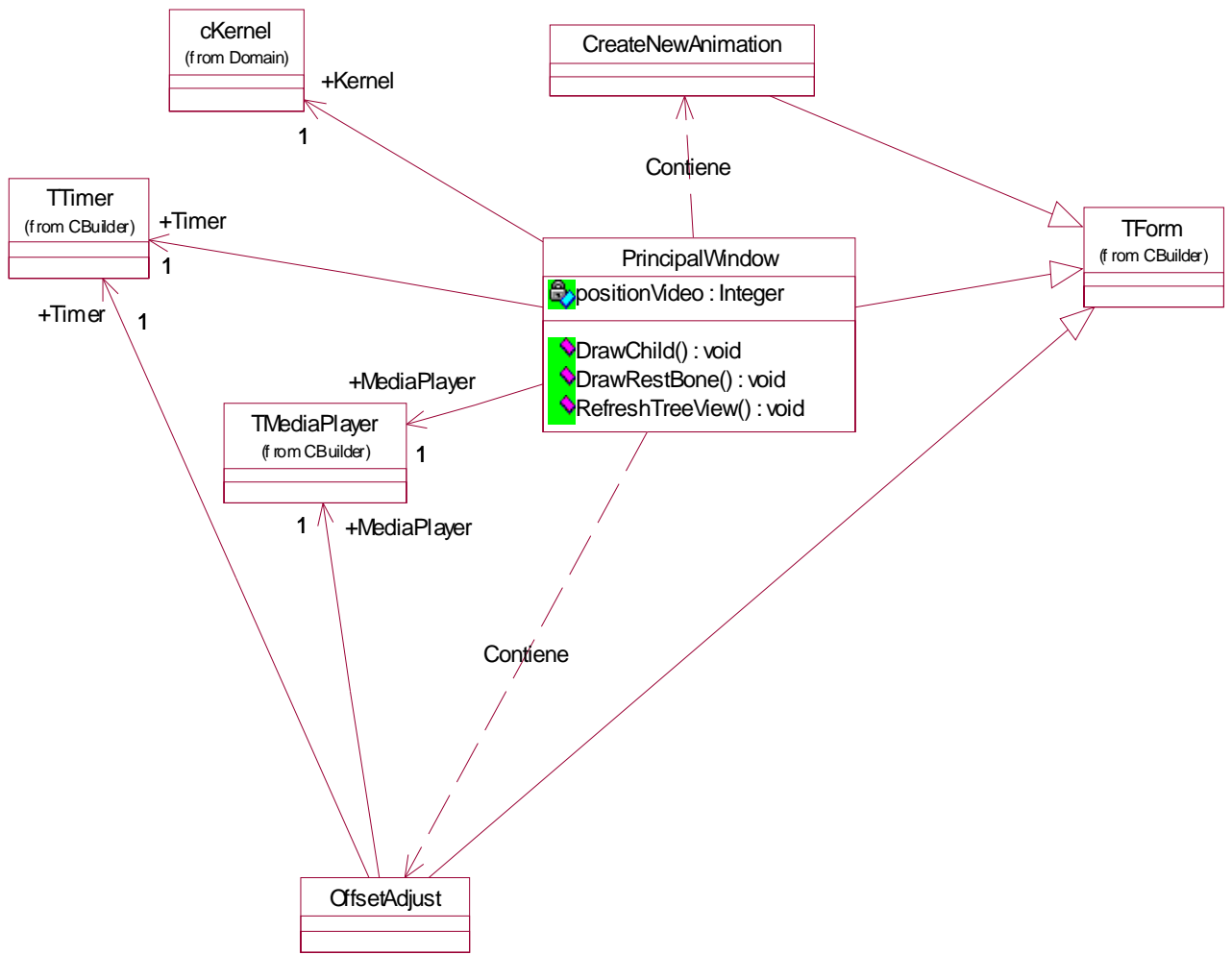


Figura 2: Diagrama de clases del paquete Presentación.



Figura 3: Diagrama de clases del paquete CBuilder.

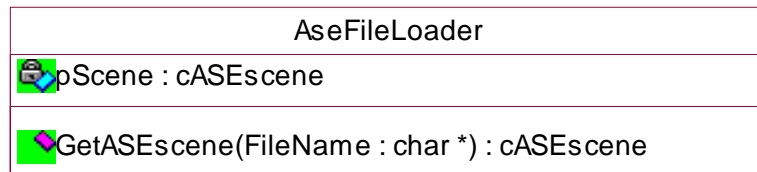
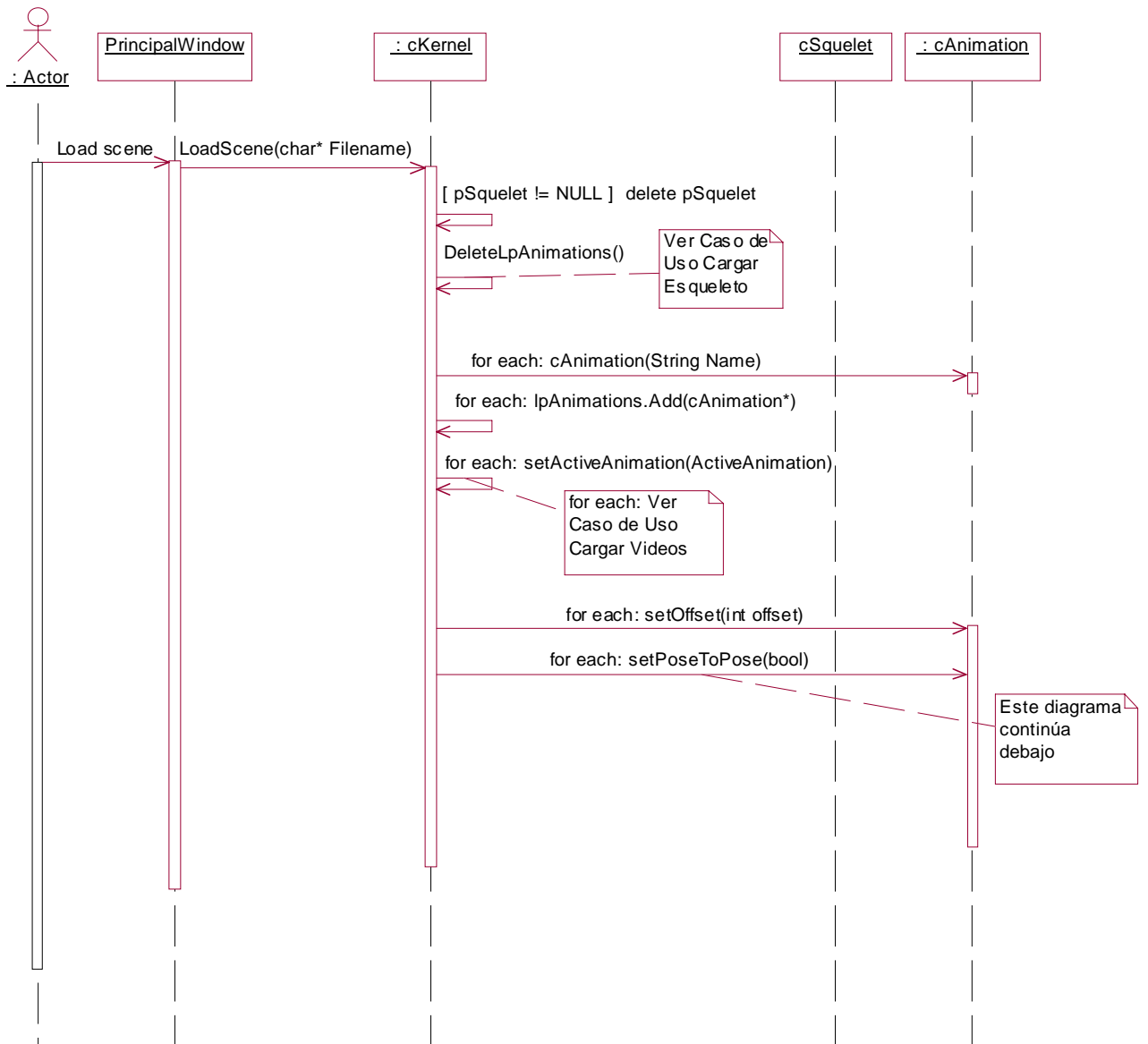


Figura 4: Diagrama de clases del paquete AseFileLoader.



Figura 5: Diagrama de clases del paquete MathLibrary.

Anexo 5: Diagramas de secuencia



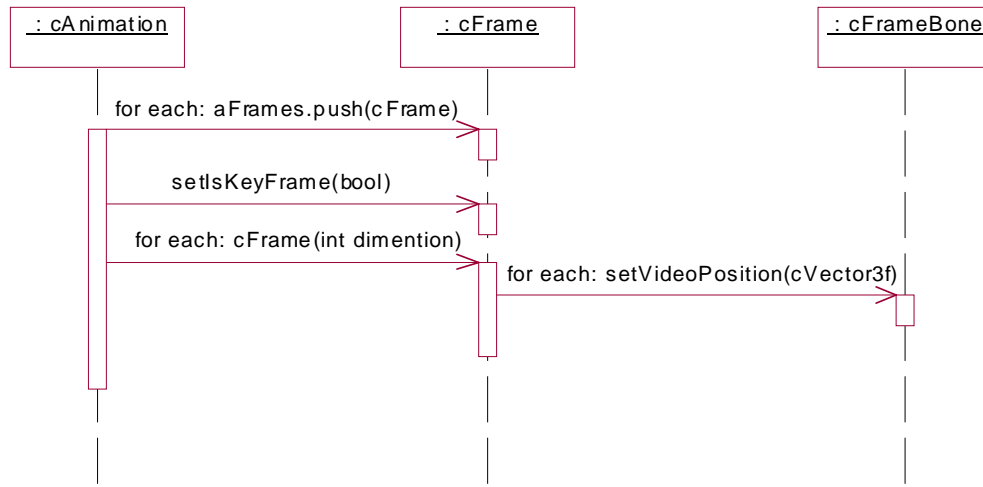


Figura 1: Diagrama de secuencia Cargar escena.

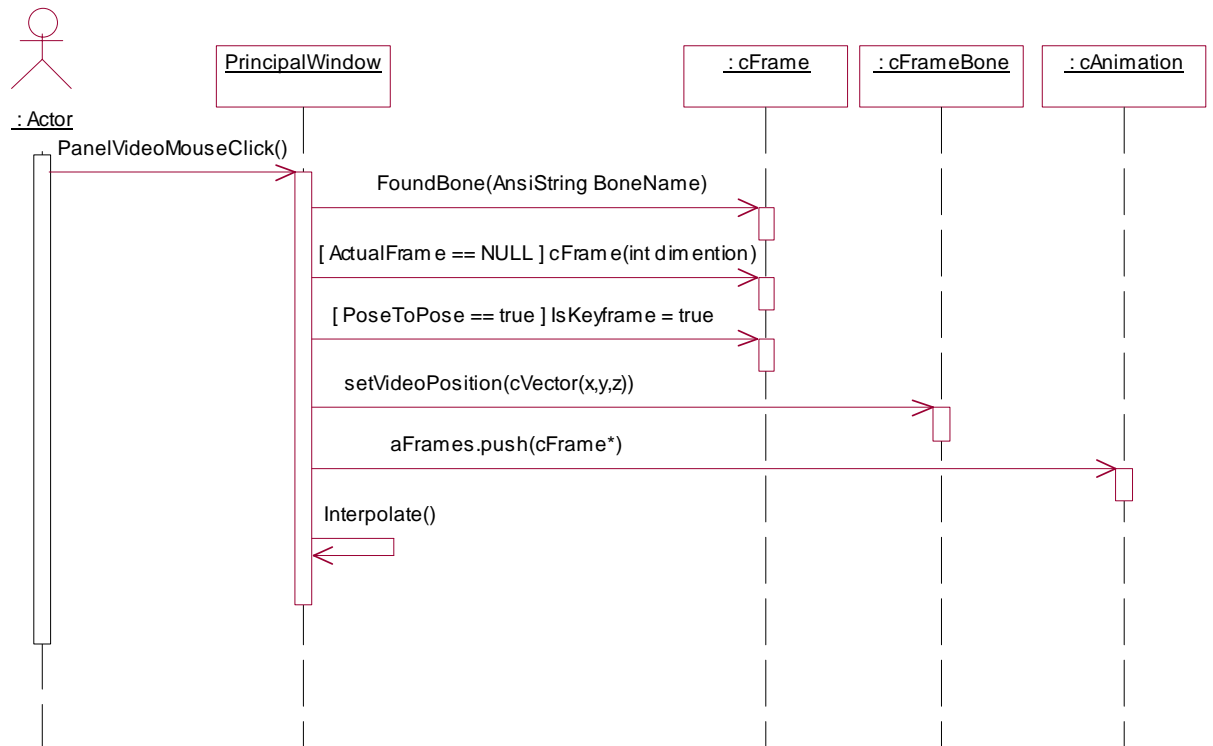


Figura 2: Diagrama de secuencia Capturar puntos.

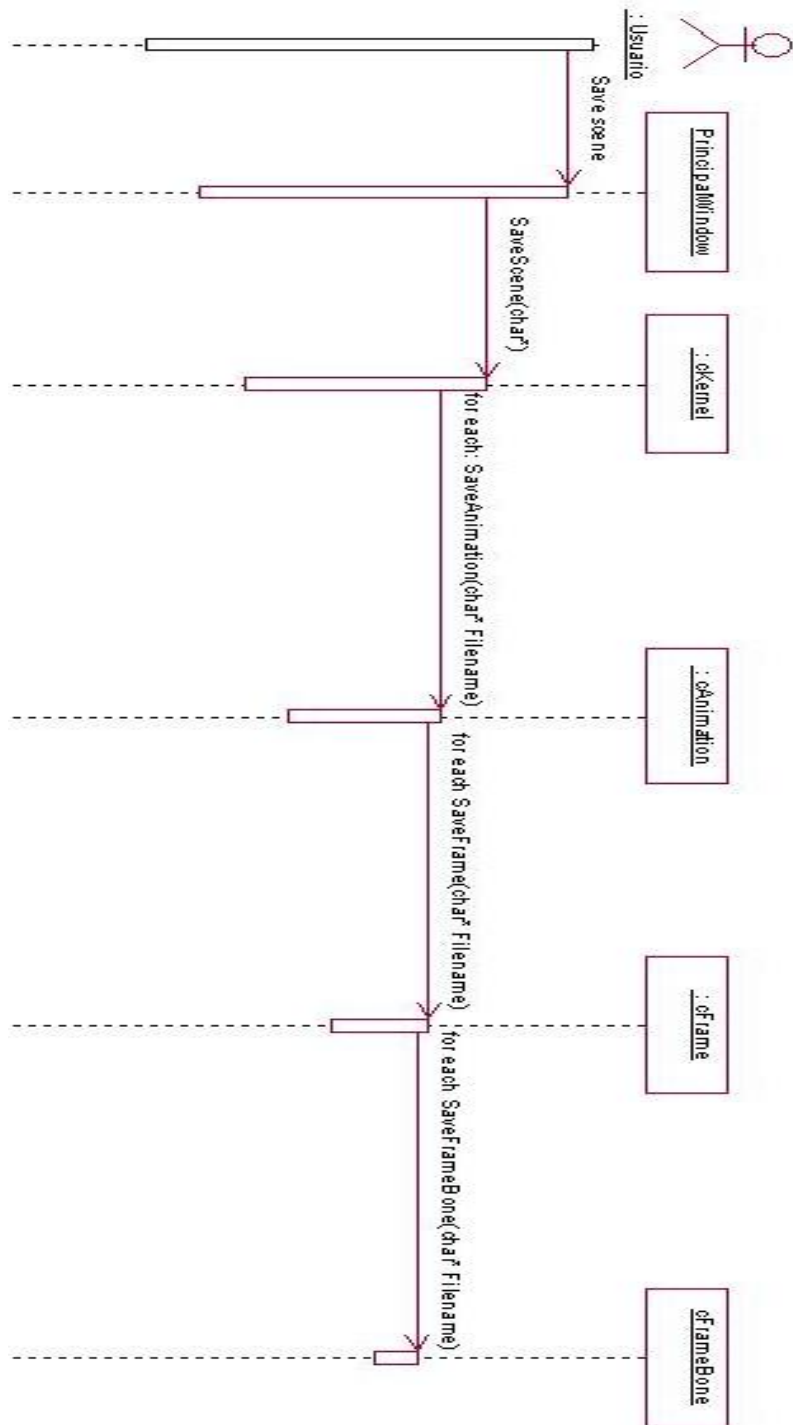


Figura 3: Diagrama de secuencia Salvar escena

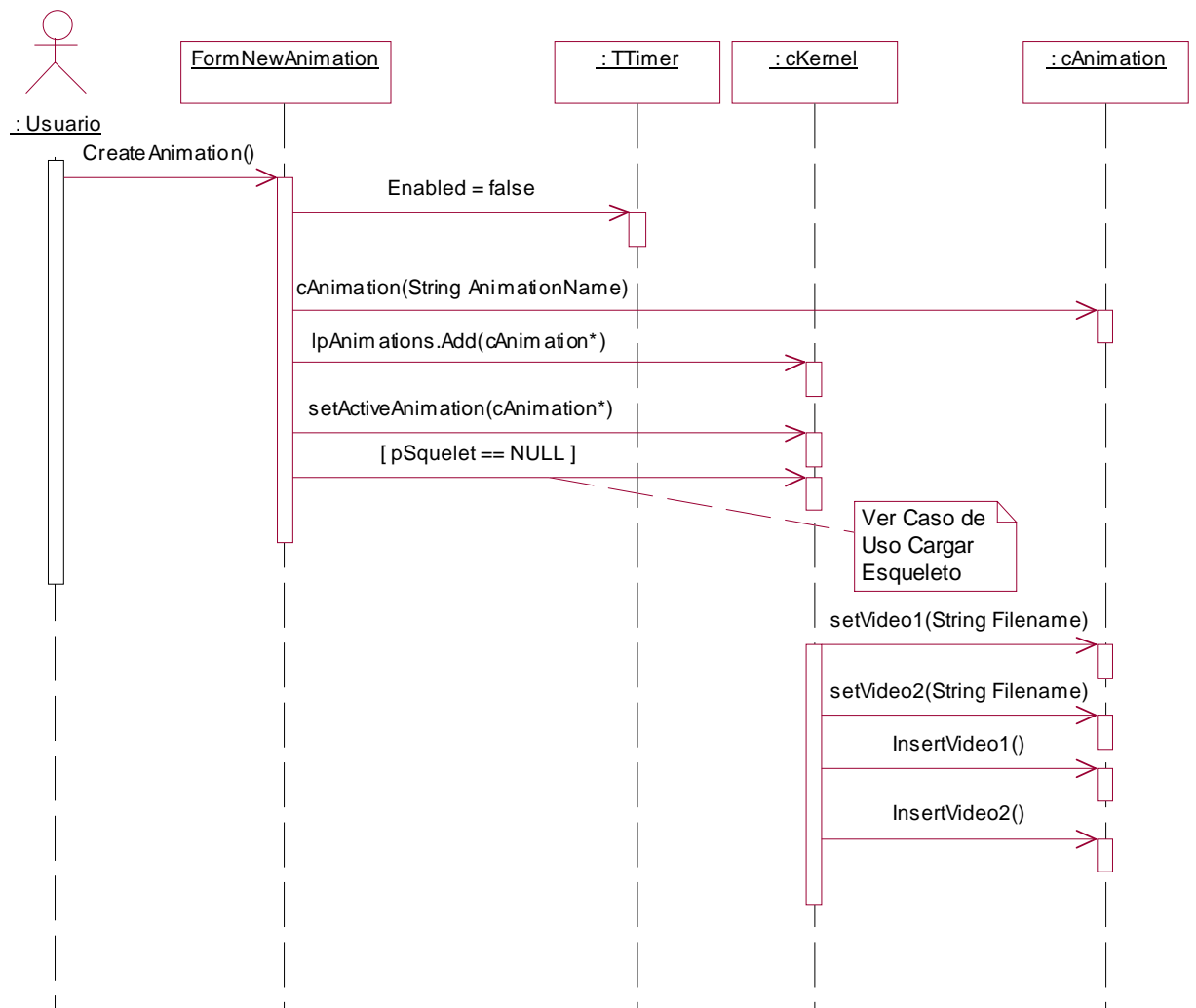


Figura 4: Diagrama de secuencia Crear nueva animación.

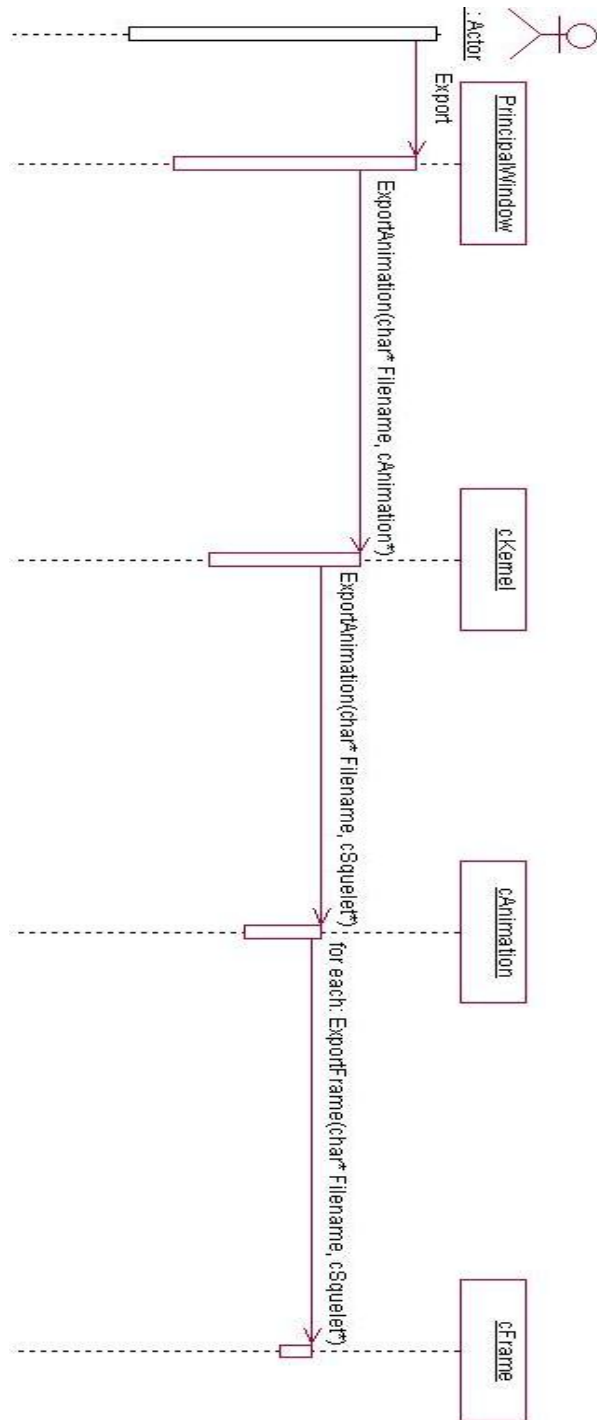


Figura 5: Diagrama de secuencia Exportar animación

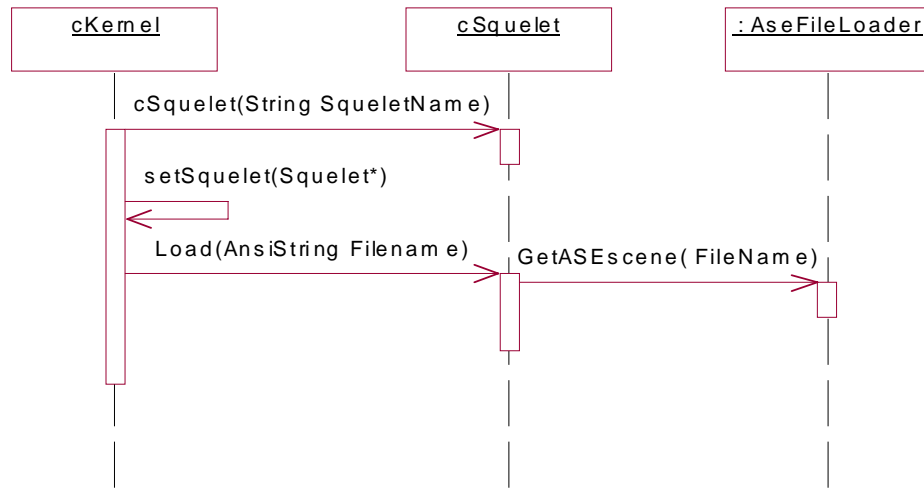


Figura 6: Diagrama de secuencia Cargar esqueleto.

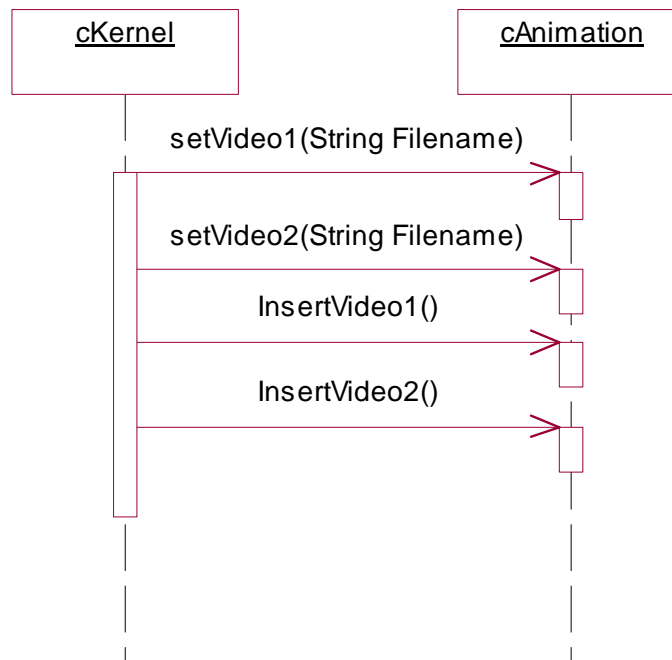


Figura 7: Diagrama de secuencia Cargar videos

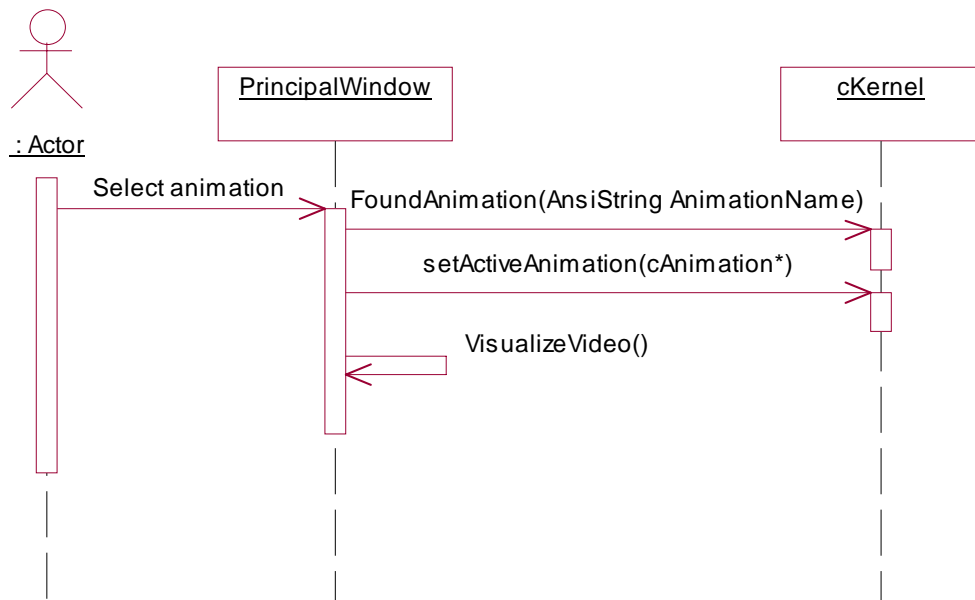


Figura 8: Diagrama de secuencia Seleccionar animación.

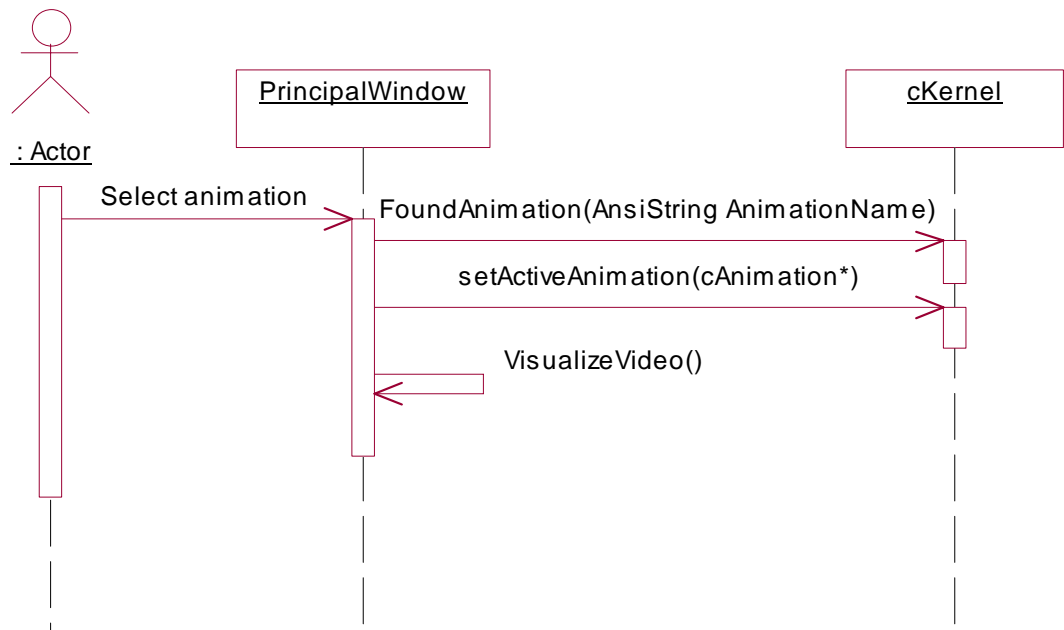


Figura 9: Diagrama de secuencia Seleccionar hueso.

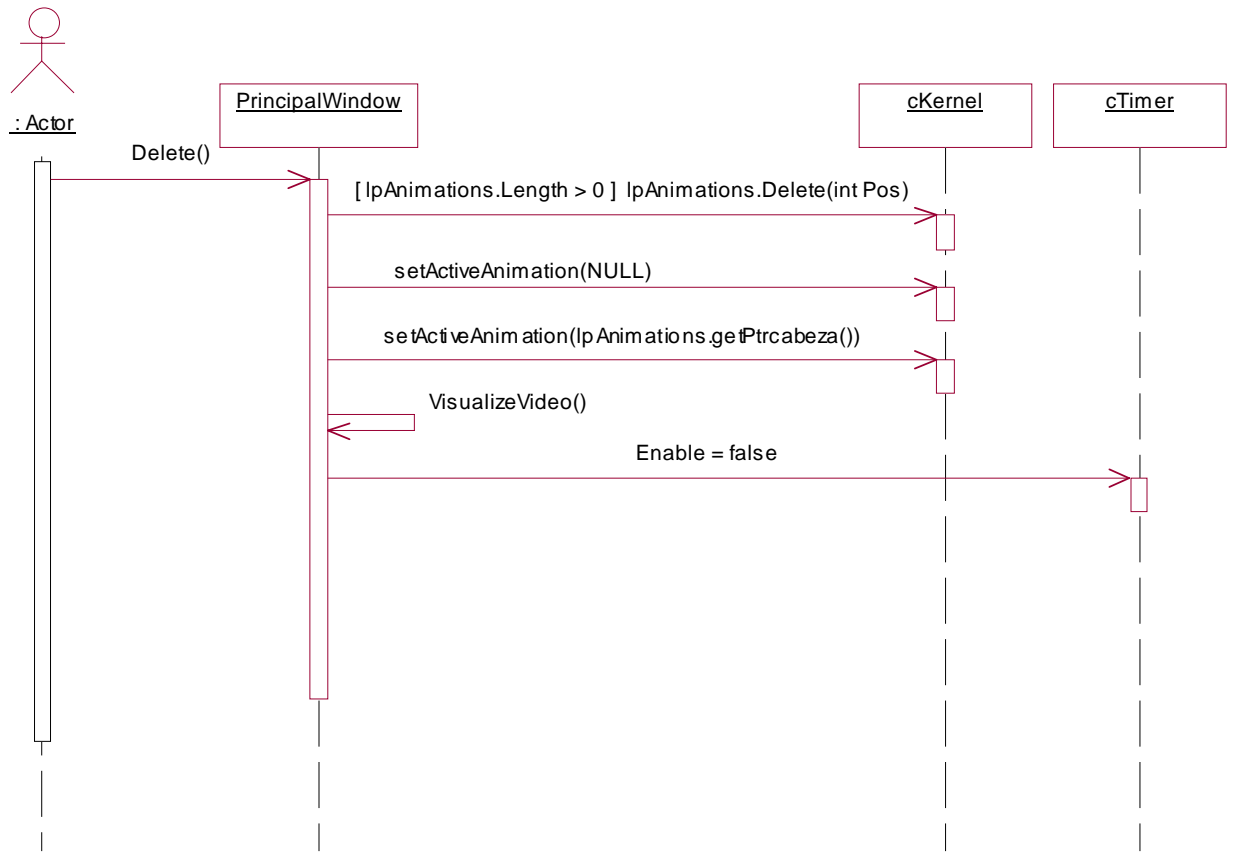


Figura 10: Diagrama de secuencia Borrar animación.