

Módulo físico para la simulación de la dinámica de los cuerpos rígidos en entornos virtuales

Physic Module for dynamics simulation of rigid bodies in virtual environment

Liudmila Pupo Peña, Liudmila Reyes Álvarez

Universidad de las Ciencias Informáticas

lpupo, lreyes@uci.cu

Resumen

En los últimos años el desarrollo de los Sistemas de Realidad Virtual (SRV) se ha expandido considerablemente a los diferentes sectores de la sociedad. El realismo de las escenas simuladas no siempre tiene la calidad requerida, uno de los principales problemas es que no se presentan las leyes físicas elementales que rigen el comportamiento de los cuerpos. El objetivo de este trabajo es la elaboración de un módulo que permita la simulación de la dinámica de los cuerpos rígidos.

En la investigación realizada se abordan los conceptos y ecuaciones matemáticas y físicas necesarias para el soporte de la dinámica de los cuerpos, se analizan las principales técnicas para lograr la animación basada en la física y se presenta una solución al problema planteado mediante un algoritmo que permite la simulación física mediante métodos de resolución de ecuaciones diferenciales, destacando el método de Runge Kutta.

El módulo resultante de este proyecto, desarrollado usando el Proceso Unificado de Software e implementado en C++ estándar, está acoplado a la herramienta "Scene ToolKit" y permitirá un mayor realismo en los productos finales que se elaboren, al lograr incorporar leyes físicas que determinan el comportamiento de los cuerpos rígidos, según factores externos (fuerzas) que incidan sobre él.

Palabras clave: Dinámica de cuerpos rígidos, realidad virtual, simulación basada en la física.

Abstract

The development of the Systems of Virtual reality has expanded considerably to the different sectors from the society such as medicine, defence, education and entertainment. The University of Computer Sciences must, like one of its profiles, the development of virtual surroundings areas to put in practice Simulators and games.

The project "Development tools of Systems of Virtual reality" works in the elaboration of a tool named "Scene Toolkit" for the development of virtual surroundings areas. This tool lacks of a physical-mathematician model who has supported the general dynamics of the rigid bodies and that allows these to interact in the scene following the elementary physical laws that govern their movement. It is why the objective of this work is the elaboration of a module that allows the simulation of the dynamics of rigid bodies in the simulators and games that are elaborated.

This document gathers the result of all the work made by investigation. It studied the concepts and necessary mathematical and physical equations to support of the dynamics of bodies and algorithms necessities to analyze and obtain the animation based on the physics and the solution given to the problem. The resulting module of this project will allow the incorporation of a greater realism. In the end items that were elaborated, will be managed to incorporate general physical laws to the bodies that determine their behaviour according to external factors (forces) that affect him.

Key words: *Based physic simulation, dynamic of the rigid body, virtual reality.*

Introducción

El movimiento de los objetos en el mundo creado debe producir la impresión de integración física y movilidad desde el punto de vista de un espacio en tres dimensiones. Alcanzar un modelo físico-matemático eficiente que logre cumplir con todos los

requisitos que un sistema necesite, y funcione de forma que los objetos interactúen y alcancen la movilidad de la forma más real posible, exige seleccionar de forma minuciosa los algoritmos y métodos que se deben seguir para modelar el mundo físico con tanta precisión como sea posible. Esta investigación tiene como objetivo elaborar un módulo de clases que permita la simulación de la dinámica de los cuerpos rígidos en entornos virtuales.

Materiales y Métodos (ó Metodología Computacional)

Animación Generada por Ordenador

Según el nivel de abstracción con que el animador especifica el movimiento, la multitud de técnicas usadas en los gráficos por computadora se pueden clasificar en dos grupos fundamentales: las Técnicas de Bajo Nivel, en las que hay un control preciso de la posición del objeto, orientación y velocidad, y las Técnicas de Alto Nivel, en las que se utilizan algoritmos complejos que establecen el comportamiento, movimiento o respuesta de los objetos (Simulación Física).

En las Técnicas de Bajo Nivel se controla de forma precisa la posición del objeto y la cámara a lo largo del tiempo: trayectorias, velocidad y aceleración . Se requiere que el animador especifique manualmente cada uno de los parámetros del movimiento individual . Estas técnicas se basan principalmente en el principio de la interpolación, es decir construcción de una función continua tomando valores determinados en correspondencia con argumentos fijos previamente especificados .

El caso de las Técnicas de Alto Nivel es la animación basada en la física, a continuación se detallan algunos aspectos de vital importancia para la comprensión de cómo funciona este tipo de animación y que aspectos físicos se deben tener en cuenta para establecer el comportamiento, interacción y movimiento de los objetos.

Animación Física

Para animar cuerpos con apariencia real, una de las mejores posibilidades es utilizar los modelos físicos que describen su movimiento. La animación física se basa en la aplicación de modelos físicos de la realidad para definir la evolución de los elementos animados de la escena . Los objetos se mueven de acuerdo con leyes físicas: gravedad, elasticidad, fluidez y por tanto reaccionan a estas leyes.

En cualquier escena se tendrán disímiles cuerpos para simular su movimiento y estos también pueden interactuar entre sí. Pueden ser desde autos, tanques, árboles hasta un buen número de situaciones o fenómenos naturales como: humo, nieve, polvo, tejidos, fuego o hasta sistemas planetarios.

Todos estos cuerpos en la literatura de los gráficos por computadora pueden ser clasificados en dos categorías fundamentales, los cuerpos deformables y los cuerpos rígidos. No obstante del tipo de objeto, el movimiento de estos generado por la simulación física se basa en la dinámica. Sin embargo la simulación de los cuerpos deformables o cuerpos no rígidos consume mucho más tiempo que la de los cuerpos rígidos.

Se entiende por **sólido rígido** o **cuerpo rígido** a un conjunto de puntos del espacio que se mueven de tal manera que no se alteran las distancias entre ellos sea cual sea la fuerza actuante. También se asume que no existe penetración en la interacción aunque si puede rebotar en respuesta a la colisión con otros cuerpos .

Los cuerpos **no rígidos** son aquellos que sufren una deformación (cambio del tamaño o la forma) debido a la aplicación de una o más fuerzas sobre él .

La dinámica es la base de la Simulación Física, se estudian aquí las fuerzas que actúan sobre los objetos y sus reacciones a ellas. La dinámica de las partículas y los cuerpos rígidos tienen determinados conceptos y modelos físicos que rigen su comportamiento. Un cuerpo presenta determinadas propiedades como posición en el espacio, velocidad lineal y angular, centro de masa, fuerzas y torques, momento lineal y angular, tensor de inercia, entre otras, todas estas propiedades se deben tener en cuenta para lograr una simulación eficiente .

Bucle de animación física

El proceso de control de animación utilizando física consiste en la resolución de un conjunto de ecuaciones lineales que responden a los siguientes elementos generales:

- Determinar las constantes dinámicas del sistema.
- Evaluar las condiciones iniciales.
- Encontrar las fuerzas y sus puntos de aplicación.
- Calcular las resultantes de las fuerzas y momentos.
- Resolver las ecuaciones de movimiento.

Información que se maneja

De los cuerpos rígidos es necesario almacenar gran cantidad de datos que definen su **estado físico**:

- ❖ Vector de Estado
- ❖ Cantidades Auxiliares
- ❖ Constantes del Cuerpo

Vector de Estado

El vector de estado debe contener en cualquier momento la información necesaria sobre el estado de cuerpo de traslación y rotación. Se define para los cuerpos rígidos un *vector de estado* $Y(t)$ con la siguiente información:

$$Y(t) = \begin{pmatrix} X(t) \\ q(t) \\ P(t) \\ L(t) \end{pmatrix}$$

Fig. 1 Vector de estado

- $X(t)$ es el vector de **posición** (x, y, z) del cuerpo en coordenadas globales como un punto de referencia. Puede corresponderse con el centro de masa del cuerpo.

- $q(t)$ es la **orientación** del cuerpo, representada por un *quaternion* de rotación.
- $P(t)$ es el vector de **cantidad de movimiento lineal**.
- $L(t)$ es el vector de **cantidad de movimiento angular**.

Constantes del Cuerpo

Estas cantidades son las que pueden ser definidas al principio de la simulación debido a que no tienen variación en ningún instante posterior se pueden mencionar como las más importantes las siguientes:

- **Masa** del cuerpo.
- I_{body} es la **Matriz de Inercia**. Esto es una matriz de 3x3 que describe como la masa del cuerpo es distribuida alrededor de su centro de masa.

La masa es la inercia del cuerpo con respecto a las fuerzas que se apliquen en su centro de masa, análogamente la matriz de inercia es la inercia con respecto a las torques aplicadas. Es valido aclarar que el tensor de inercia no es constante, lo que es constante es la matriz de inercia inicial que se usa para calcular el tensor de inercia en un instante dado, por eso también se debe guardar como una magnitud variable el **tensor de inercia**.

Cantidades auxiliares

Las cantidades auxiliares que se almacenan son las que guardan la relación que existe entre el vector de estado del cuerpo y las

fuerzas y torques que se aplican. Si se deriva el vector de estado $Y(t)$ se obtiene $\frac{d}{dt}Y(t)$:

$$Y'(t) = \begin{pmatrix} V(t) \\ \frac{1}{2}W(t)q(t) \\ F(t) \\ T(t) \end{pmatrix}$$

Fig. 2 Derivada del vector de estado

- $V(t)$ es el vector que almacena la **velocidad lineal** del cuerpo.
- $W(t)$ es la **velocidad angular** del cuerpo.
- $\frac{1}{2}W(t)q(t)$ define la relación que se establece entre el *quaternion* de rotación y la velocidad angular que describe como cambia la orientación del cuerpo en el tiempo, a esta relación le llamaremos para identificarla **spin**.
- $F(t)$ es el vector que almacena las **fuerzas** aplicadas sobre el cuerpo.
- $T(t)$ es el vector que almacena el **torque** producido por las fuerzas que actúan sobre el cuerpo.

- $I(t)$ es una matriz 3x3 que guarda el **tensor de inercia** del cuerpo.

En cualquier momento dado se pueden calcular $I(t), W(t)$ y $V(t)$ por la siguiente fórmula:

$$V(t) = \frac{P(t)}{M}, \quad I(t) = R(t)I_{body}R(t)^T, \quad \text{y} \quad W(t) = I(t)^{-1}L(t)$$

Propiedades del mundo físico

Además de controlar información de los cuerpos rígidos es necesario tener en cuenta las propiedades y condiciones que deben existir en el entorno para que estos cuerpos mantengan su comportamiento según el mundo físico, es por ellos que se deben almacenar algunas propiedades generales que afectarán a todos los cuerpos de la escena, ellas son las siguientes:

- Gravedad: constante de la aceleración de la gravedad que se desee utilizar.
- Superficies: lista de superficies físicas que se tendrán en cuenta para determinar las fuerzas actuantes sobre el cuerpo cuando este apoyado sobre la misma. Se debe almacenar de cada superficie sus límites dados por un punto mínimo y un punto máximo, y los coeficientes de fricción estático y cinético.
- Medios o ambientes: lista de ambientes que se tendrán en cuenta para ejercer una fuerza resistiva sobre el objeto que se encuentre en alguno de ellos. De cada ambiente se almacena su tipo, y un punto mínimo y un punto máximo que delimitan el volumen que los contiene.

Obtención de un nuevo estado en la Simulación

La obtención de un nuevo estado en la simulación se puede obtener por un algoritmo que funcione paso a paso. Teniendo el estado de un cuerpo en el tiempo t , la simulación deberá calcular el nuevo estado del cuerpo un tiempo después. De esta forma el estado final del paso anterior será la condición inicial del próximo paso a realizar.

Si k indica el número de pasos en el que se encuentra la simulación, teniendo el estado actual del cuerpo por el vector $Y(t_k)$ y su derivada hay que enfrentar el problema de cómo encontrar $Y(t_{k+1})$. Esto se resuelve por métodos numéricos de resolución de ecuaciones diferenciales, como por ejemplo los métodos de Euler, de Runge-Kutta o de Paso Aditivo, entre otros.

Estos métodos analizan la obtención de una aproximación a una función $Y(t)$ en un intervalo $[a,b]$ calculando su valor para valores de tiempo t_k . Esta función viene dada por la ecuación:

$$Y' = f(t, y), \quad \text{con} \quad Y(a) = Y_0$$

Siendo $Y(t)$ un vector de funciones por lo que se deben resolver cuatro problemas de este tipo donde la función $Y(t)$ en cada uno es $X(t)$, $q(t)$ o $R(t)$, $P(t)$ y $L(t)$ respectivamente.

Método de Euler

El método de Euler es el método más básico de las técnicas de integración numérica. Solamente alcanza un 100 % de exactitud si la tasa de cambio de $Y(t)$ es constante durante todo el tiempo. Pero como esto no ocurre en la mayoría de los casos este método no es lo suficientemente bueno .

El método de Euler comienza dividiendo el intervalo $[a,b]$ en M subintervalos del mismo tamaño usando la partición dada por los siguientes puntos.

$$t_k = a + kh \text{ Para } k \text{ siendo } h = \frac{b - a}{M}$$

Donde h es el tamaño de paso (que en este caso representa tiempo). En la simulación el tamaño del paso puede ser constante o no. Sin embargo, este análisis es útil porque cada paso de la simulación será como el primero en un problema diferencial diferente con un h propio.

Se procede ahora a resolver aproximadamente $Y' = f(t, y)$ en $[t_o, t_M]$ con $Y(t_o) = Y_0$

Suponiendo que $Y(t), Y'(t), Y''(t)$ son continuas y usando el Teorema de Taylor para desarrollar $Y(t)$ alrededor de $t = t_o$, para

$$\text{cada punto } t \text{ existe un punto } c_1 \text{ entre } t_o \text{ y } t \text{ tal que } Y(t) = y(t_o) + y'(t_o)(t - t_o) + y''(c_1) \frac{(t - t_o)^2}{2}.$$

Al sustituir $h = t_1 - t_o$ el resultado es:

$$Y(t_1) = y(t_o) + hf(t_o, y(t_o)) + y''(c_1) \frac{h^2}{2}$$

Si el tamaño de paso h es suficientemente pequeño, entonces se puede despreciar el término que contiene h^2 y obtener:

$$Y(t_1) \approx y_1 = y(t_o) + hf(t_o, y(t_o))$$

Que se llama aproximación de Euler. Por despreciar h^2 se dice que hay un error de truncamiento local de orden 2 o $O(h^2)$ y un error acumulativo o global de orden $O(h)$. Cuanto más grande sea el exponente de h a despreciar menos será el error ya que h es cercano a 0. Generando el proceso se obtienen una sucesión de puntos que se aproximan a la gráfica de la solución $Y = Y(t)$.

El paso general del método de Euler es:

$$t_{k-1} = t_k + h \quad \text{Para } k$$

En cada obtención de un valor de Y_{k+1} , se usa el valor anterior Y_k solamente, de forma que los errores en cada paso se van acumulando hasta que el error acumulado se hace demasiado importante. Las consecuencias de este error pueden verse en la simulación cuando aparecen fuerzas grandes que hace que los cuerpos empiecen a tener comportamientos extraños (parece que tiemblan) y a medida que pasa el tiempo empeora.

Método de Runge-Kutta o de Paso Aditivo

El método de Runge-Kutta se construye a partir de un desarrollo por Taylor de la función $Y(t)$. Existen diferentes métodos de Runge-Kutta de diferentes órdenes. El método de orden N es que tiene un error global final del mismo orden $O(h^N)$. Esto se logra evaluando la función y en varios puntos, en cada paso. Cuando el orden es de $N=4$, el método es bastante preciso, estable y fácil de programar, aun más la mayoría de los expertos dicen que no es necesario trabajar con métodos de orden superior porque el aumento de costo computacional no compensa la mayor exactitud.

Consiste en calcular la aproximación Y_{k+1} de la siguiente manera:

$$Y_{k+1} = y_k + w_1 k_1 + w_2 k_2 + w_3 k_3 + w_4 k_4$$

Donde k_1, k_2, k_3, k_4 son de la forma:

$$k_1 = hf(t_k, y_k)$$

$$k_2 = hf(t_k + a_2 h, y_k + b_2 k_1)$$

$$k_3 = hf(t_k + a_3 h, y_k + b_3 k_1 + b_3 k_2)$$

$$k_4 = hf(t_k + a_4 h, y_k + b_4 k_1 + b_4 k_2 + b_4 k_3)$$

Emparejando estos coeficientes con los del método de la serie de Taylor de orden $N=4$ de manera que el error de truncamiento local sea de orden $O(h^5)$, se obtiene un sistema de ecuaciones que lleva a los valores de la solución que para las variables a_i y w_i Son:

$$a_2 = \frac{1}{2}, a_3 = 1, b_1 = \frac{1}{2}, b_2 = \frac{1}{2}, b_3 = 0, b_4 = 0, b_5 = 1, w_1 = \frac{1}{6}, w_2 = \frac{1}{3}, w_3 = \frac{1}{3}, w_4 = \frac{1}{6}$$

Con estos valores se obtiene la fórmula para el método de Runge-Kutta de orden $N=4$ estándar. A partir del punto inicial (t_0, y_0) se genera la sucesión de aproximación usando la fórmula recursiva:

$$Y_{k+1} = y_k + \frac{h(f_1 + 2f_2 + 2f_3 + f_4)}{6}$$

Donde:

Como puede verse al hacer un paso de tamaño h en el tiempo, la función f es evaluada en los tiempo t . Esto requiere un mayor número de cálculos que el método de Euler, pero a menos que se esté realizando una simulación muy simple que no tenga problemas potenciales de inestabilidad, la mejora en la reducción de error es considerable y vale la pena realizarla.

Este método tiene la peculiaridad de adecuarse al tipo de función f de forma que el error en la resolución se mantiene bajo y acotado. El método aumenta la estabilidad de la simulación, pero tiene la desventaja de que el cálculo de la estimación de error consume procesamiento.

Interacción entre cuerpos

En una escena, uno o más cuerpos pueden estar en movimiento en un instante de tiempo, es por eso que a menudo pueden producirse colisiones entre los cuerpos. El proceso de colisión se puede dividir en dos etapas:

- ✓ Detección de la colisión.
- ✓ Respuesta a la colisión.

La detección de la colisión debe determinar si dos cuerpos están colisionando para luego analizar si es necesario aplicar el impulso correspondiente. La información a obtener de la intersección de dos cuerpos es el punto de colisión y la normal de colisión. El punto de colisión será útil para analizar si los cuerpos están colisionando o si se están separando. La normal de colisión sirve para determinar la dirección del impulso a aplicar.

La detección de la colisión no forma parte del objetivo de esta investigación, es por eso que sólo se brinda una pequeña información de cómo ocurre, sin entrar en detalles.

La respuesta a la colisión entre los cuerpos, es la etapa en que se intentan separar los cuerpos que colisionaron, para esto se aplica el método del impulso, consiste en aplicar una fuerza producto del choque a los cuerpos que colisionaron que tiende a separarlos. A esta fuerza que es muy grande y se aplica por un período de tiempo muy chico, se le asocia un impulso, de forma que a los cuerpos les causa un cambio de velocidad instantánea. Este impulso se calcula en función de la velocidad, de las masas de los cuerpos y del coeficiente de restitución.

Impulso

Un impulso J es una fuerza muy grande que se aplica por un período de tiempo muy chico: Este impulso se aplica en dirección a la normal de colisión y provoca un cambio instantáneo en la cantidad de movimiento de $\Delta \mathbf{P} = \mathbf{J}$.

El torque impulsivo que genera J será $\mathbf{C} \times \mathbf{J}$, donde \mathbf{C} es el vector del punto de colisión y $\mathbf{X}(t)$ es el vector de coordenadas del centro de masa del cuerpo.

Este torque impulsivo generará un cambio instantáneo en el momento angular de $\Delta \mathbf{L} = \mathbf{T}_j$.

Como el impulso debe separar los cuerpos que colisionaron, deberá ser opuesto para los cuerpos a y b en cuestión:

Donde j es el valor modular del Impulso y \mathbf{N} el vector normal al plano de colisión y se obtiene:

Y

$$\dot{C}_a(t_0) = V_a(t_0) + W_a(t_0) \times [C_a(t_0) - X_a(t_0)]$$

$$\dot{C}_b(t_0) = V_b(t_0) + W_b(t_0) \times [C_b(t_0) - X_b(t_0)]$$

Donde $W_a(t)$ y $W_b(t)$ son las velocidades angulares de los cuerpos a y b respectivamente.

Para obtener el valor del módulo de J hay que comparar la velocidad relativa antes y después del choque. Las leyes que rigen las colisiones hacen uso de un coeficiente llamado coeficiente de restitución que es un factor de escala entre ambas velocidades:

Donde ξ es el coeficiente de restitución, V_{rel}^- y V_{rel}^+ son las velocidades relativas de los cuerpos antes y después de la colisión.

El valor de ξ esta entre 0 y 1(inclusive) y establece cuanta energía se pierde en el choque. Si $\xi=1$, no se pierde energía (colisión elástica) y si $\xi=0$, los cuerpos pierden toda velocidad en la dirección normal (colisión instantánea).

Las relaciones entre las velocidades antes (V_a^-) y después (V_a^+) del choque son:

$$V_a^+(t_0) = V_a^-(t_0) + \frac{j \cdot N(t_0)}{M_a}$$

$$W_a^+(t_0) = W_a^-(t_0) + I_a^{-1}(t_0) \cdot \{[C_a(t_0) - X_a(t_0)] \times j \cdot N(t_0)\}$$

Donde M_a es la masa del cuerpo a y I_a^{-1} es la inversa de su tensor de inercia. Combinando estas dos ecuaciones y comparando

con la de C_a^+ se obtiene una relación entre las velocidades relativas antes y después de la colisión:

$$\dot{C}_a^+(t_0) = V_a^+(t_0) + W_a^+(t_0) \times [C_a(t_0) - X_a(t_0)]$$

Luego comparando con la ecuación del coeficiente de restitución se llega a una ecuación donde puede despejarse j (1):

$$j = \frac{-(1 + \xi) \cdot V_{rel}^-}{\frac{1}{M_a} + \frac{1}{M_b} + N(t_0) \cdot \{[W_a^- + I_a^{-1}(t_0) \cdot [R_a \times N(t_0)]] \times R_a + [W_b^- + I_b^{-1}(t_0) \cdot [R_b \times N(t_0)]] \times R_b\}}$$

Donde $|y|$

Resultados

La propuesta que se brinda es la siguiente:

El módulo a desarrollar debe tener en cuenta dos partes fundamentales, debe primeramente actualizar el estado de los cuerpos en cada instante de tiempo y como segundo aspecto atender el proceso de respuesta a la colisión que se produce entre ellos.

Para actualizar el estado de un cuerpo se debe tener en cuenta las siguientes entradas:

- El estado anterior en el que se encontraba, que sería el vector de estado definido anteriormente.
- Las fuerzas que se apliquen sobre el cuerpo que producirán una nueva variación en el estado.
- El espacio de tiempo desde el paso anterior hasta este nuevo estado.

Después de cada uno de estos pasos se debe procesar si existen colisiones entre los objetos, y este ciclo repetirlo mientras dure la simulación. Para cada nueva iteración el estado anterior es el estado nuevo de la iteración anterior. Así se verá como funciona en la evolución del tiempo la dinámica de los cuerpos cuando se le aplica fuerzas sobre ellos. Esto se aprecia en la figura que se muestra a continuación:



Fig. 3 Actualización de un Nuevo estado en la Simulación

El proceso “Actualizar Estado” a partir del vector de estado $Y(t)$ y su derivada $Y'(t)$ se obtendrá un nuevo estado y se recalcularan las cantidades auxiliares definidas anteriormente, luego de un intervalo de tiempo. El método que se utilizará por defecto para resolver estas ecuaciones diferenciales es el de Runge-Kutta que es más exacto y acumula menos error en el resultado final. Pero se dejara la opción al usuario de escoger el otro método documentado (Euler) para realizar la simulación, esto se tuvo en cuenta, dado que si el entorno que se va a simular es sencillo y no requiere de exactitud en los cálculos de posición y orientación de los cuerpos este otro método podría resultar conveniente.

Atender el proceso de respuesta a las colisiones es el paso que sigue después de realizar la actualización del estado de los cuerpos. Lo primero que se analiza es que objetos están colisionando, información que se dará de otro módulo, teniendo en cuenta la información de qué cuerpos están colisionando se procede a realizar una acción que intentará separarlos. El método que se utilizará es el del Impulso.

Ahora se muestra el algoritmo de simulación final en un pseudo-código:

Inicio

1. Inicializar propiedades del mundo físico.
2. Inicializar estados de los cuerpos.

3. Para cada cuerpo con controlador físico asociado que lo identifique:
 - 3.1. Calcular tamaño de paso Δt .
 - 3.2. Determinar la sumatoria de las fuerzas aplicadas.
 - 3.3. Aplicar fuerza resultante al cuerpo.
 - 3.4. Calcular la derivada del vector $Y(t)$, y $Y(t)$ con la fuerza total.
 - 3.5. Obtener $Y(t+\Delta t)$. Si un cuerpo colisionó con otro.
 - 3.6. Calcular la V_{rel} entre los cuerpos que colisionaron.
 - 3.7. Si $V_{rel} \leq 0$ aplicar impulso.
4. Repetir el paso 3 durante toda la simulación
5. Fin

Este pseudo-código contiene un ciclo principal donde, para cada uno de los cuerpos que es de interés que estén sometidos a leyes físicas, se le calcula en cada paso de la simulación la suma de fuerzas que actúan sobre él y se les separa entre las que actúan sobre el centro de masa y las que producen torque (rotación del cuerpo), ambos juntos con el vector de estado se utilizan como punto de partida para calcular el nuevo estado de la simulación un instante posterior. En el ciclo, si ocurre alguna colisión se le da respuesta a la misma mediante el método del impulso.

A continuación se brinda un diagrama que muestra las relaciones existentes entre las diferentes clases que se utilizaron para la elaboración del módulo, las clases sombreadas representan las clases que permitieron el acople con la herramienta “Scene ToolKit”.

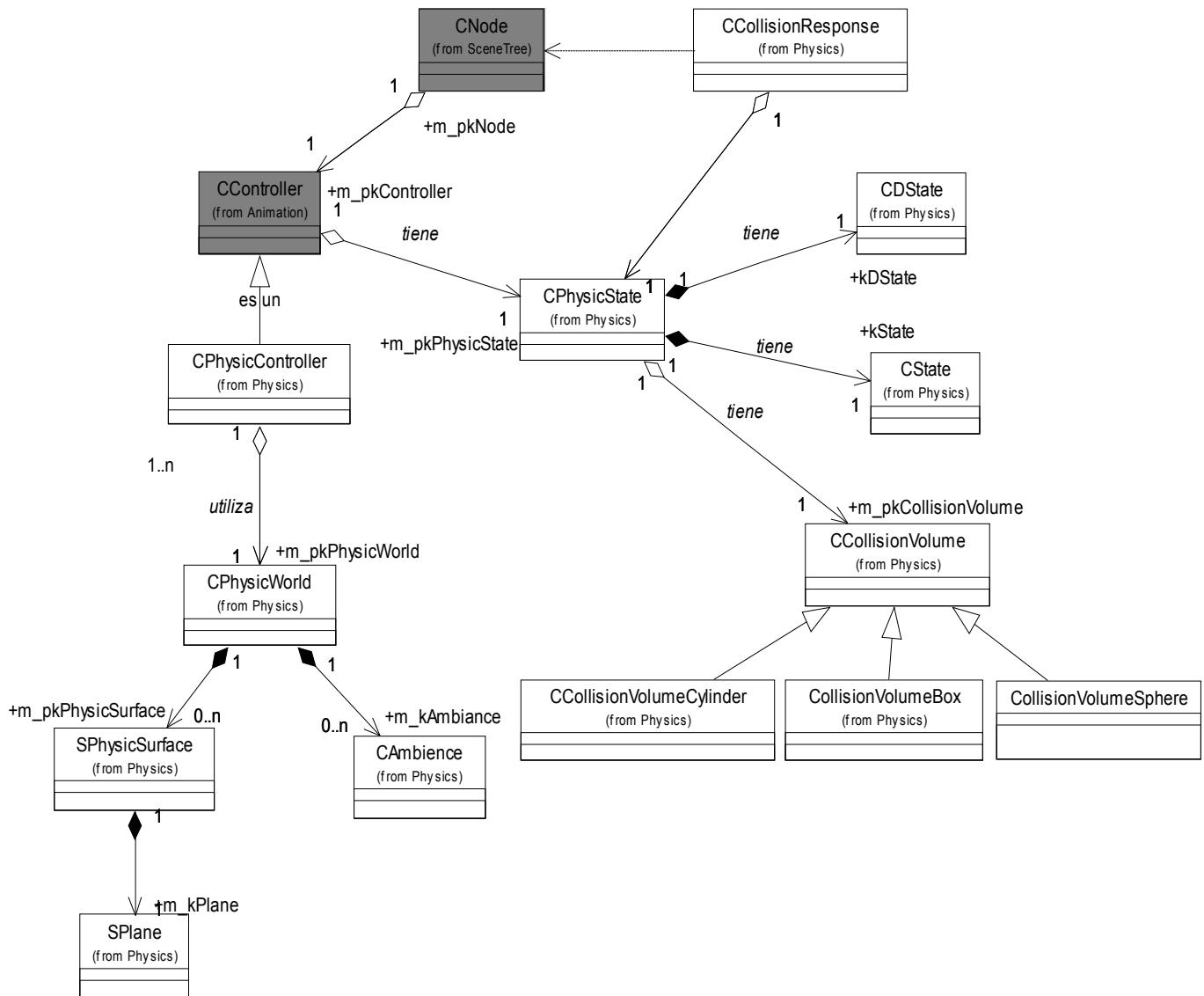


Fig. 4 Diagrama de Clases

Conclusiones

Para el cumplimiento del objetivo propuesto en este proyecto se realizó un estudio del comportamiento físico matemático de los cuerpos rígidos y las leyes generales que los afectan. Se investigó como lograr con técnicas óptimas la simulación eficiente basada en la física en los SRV en tiempo real. Y a partir de la investigación realizada se propone las características técnicas de la solución propuesta.

Posteriormente se avanza por los distintos flujos de trabajo que proporciona la metodología RUP (requisitos, análisis, diseño e implementación) para lograr un producto final de calidad y que proporcione una solución al las necesidades del cliente.

Se implementó un algoritmo para la simulación del comportamiento dinámico de los cuerpos rígidos en un entorno virtual. Para crear el realismo en el mundo físico se utilizaron: la ley de la conservación del momento en la respuesta a la colisión, las fuerzas de fricción entre superficies, la fuerza de fricción de resistencia del medio, la fuerza de gravedad y la fuerza de Arquímedes.

Con el módulo desarrollado se puede afirmar que se simula, con aceptable aproximación, el comportamiento dinámico de los cuerpos rígidos teniendo en cuenta sus propiedades, bajo la acción de las leyes físicas implementadas. Su incorporación a la herramienta "Scene ToolKit" servirá para lograr un mayor realismo en los proyectos finales que se elaboren.

Referencias Bibliográficas

Andrew Witkin, J. P. "Interactive Manipulation of Rigid Body Simulations."

Baraff, D. Physically Based Modeling Rigid Body Simulation. P. A. Studios.

CHRISTOPHER, T. "Mathematics for game developers".

Curello, P. A. Simulación de la Dinámica de los Cuerpos Rígidos en Tiempo Real, Instituto Tecnológico de Buenos Aires.

Fiedler, G. (2006). "Game Physics: Integration Basics." Gaffer on Games.

Informática., D. d. (2007). Ampliación de la Informática Grafica Tema 5 Animación 3D Universidad de Valencia.

Lifshitz., L. (1991). Mecánica: Mecánica del sólido rígido. Barcelona.

Luis Palomino Ramírez, I. R. G. (1997). Diagonalización del Tensor de Inercia para mejorar la Eficiencia en la Simulación de Cuerpos Rígidos.

Parejo, J. C. (2004). Especificación y Control del Movimiento.

Parejo, J. C. (2004). "Especificación y Control del Movimiento."

Rosario, L. M. (2004). "Deformación Plástica. Mecánica de sólidos deformables."

Young-Min Kang, H.-G. C. (2004). "Real-time Animation of Complex Virtual Cloth with Physical Plausibility and Numerical Stability." Teleoperators & Virtual Environments.