

MODELIZACION DE FENÓMENOS FÍSICOS EN APLICACIONES DE COMPUTACIÓN GRÁFICA EN TIEMPO REAL

Lazo M.^{1,2}, García Bauza C.^{1,2}, Vénere M.^{1,3} y Clause A.^{1,4}

¹ Universidad Nacional del Centro, 7000 Tandil, Argentina,

² Comisión de Investigaciones Científicas de la Provincia de Buenos Aires

³ Comisión Nacional de Energía Atómica

⁴ CONICET

1 INTRODUCCIÓN

Cuando se desarrollan aplicaciones de realidad virtual, uno de los principales desafíos es lograr realismo e inmersión a un costo computacional aceptable. Con las placas gráficas actuales es posible obtener imágenes muy cercanas a la realidad, aunque también se debe incluir comportamiento físico para lograr escenas que brinden una verdadera inmersión.

Para cumplir este objetivo, se han desarrollado en los últimos años varios motores físicos implementados en forma de bibliotecas (ODE, NGD, TPE, BPL¹) o en placas dedicadas a la simulación de modelos físicos (por ejemplo, el producto de nVIDIA[®] denominado PhysX); los cuales permiten agregar comportamiento físico a escenas tridimensionales.

Es posible que los usuarios deseen realizar pruebas de rendimiento y/o precisión a cada motor físico para determinar cual de ellos se ajusta mejor a sus necesidades; o utilizar distintas funcionalidades de cada uno en una misma aplicación, pero ocurre que cada uno de estos motores necesita una configuración distinta de inicialización y muchas veces la funcionalidad implementada difiere entre uno y otro, haciendo que el código implementado sea obsoleto, requiriendo de un esfuerzo considerable para portar la aplicación a otro motor.

Para solucionar dichos problemas, el presente trabajo se centra principalmente en crear una capa de abstracción o PAL (del inglés, *Physics Abstraction Layer*) que sea utilizada por un motor gráfico, la cual brinde una comunicación fluida con diversos motores físicos maximizando la reutilización de código y permita cambiar el uso entre ellos.

2 CAPA DE ABSTRACCION PROPUESTA

Si bien existen esfuerzos en esta área [Boeing 2005, Fischer 2005], estas propuestas se alejan de la idea de facilitar el intercambio de motores físicos de una manera transparente y cómoda para el usuario como la propuesta de [Fischer 2005], o proveen un conjunto acotado de funcionalidad como [Boeing 2005]; exponiendo únicamente la funcionalidad común de los motores físicos que soporte la PAL; es decir, se utiliza un esquema de intersección de funcionalidades; no permitiendo al usuario utilizar algunas de las características de aquellos motores físicos que tengan mayor funcionalidad (Figura 1 Izq.).

En nuestra propuesta, (Figura 1 Der.) se utiliza un esquema de unión de funcionalidades, no sólo proveyendo acceso a la funcionalidad común de los motores físicos, sino también brindando acceso a funcionalidad específica a través del uso de decoradores de diseño (conocidos como *wrappers*). Esto permite que la PAL soporte el conjunto completo de funcionalidades de los motores físicos soportados.

¹ Ver Información adicional en Sección Referencias

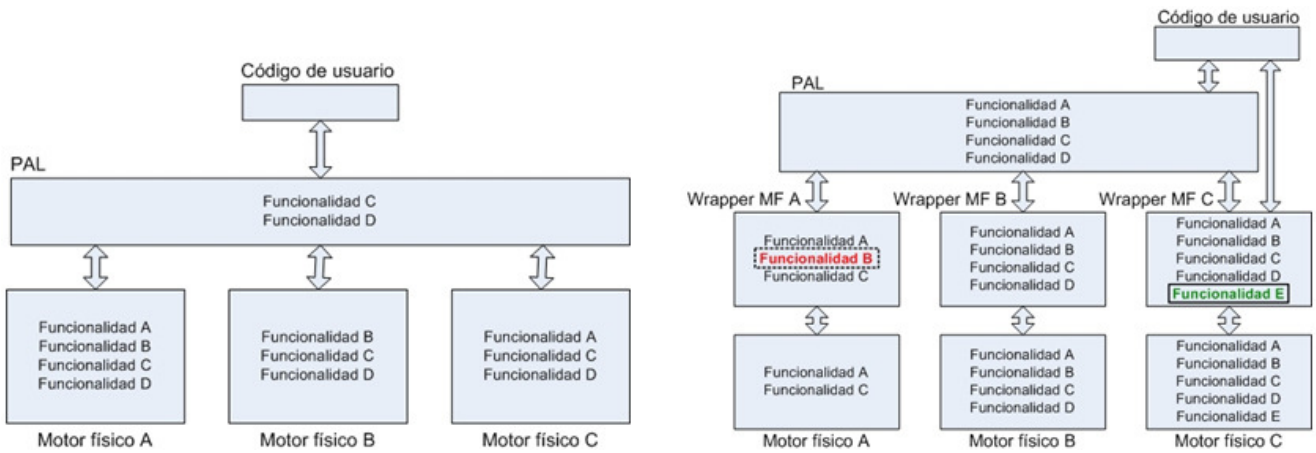


Figura 1: Izq: Capa de abstracción utilizando esquema de intersección de funcionalidades. Der: Capa de abstracción utilizando esquema de unión de funcionalidades.

Tanto las propuestas de [Boeing 2005] como [Fischer 2005] dejan a criterio del usuario la utilización o integración de una biblioteca gráfica; siendo éste un problema no trivial, puesto que comúnmente la interacción entre las dos partes es crucial. Por lo tanto, se implementaron dos mecanismos de sincronización, donde en el primero la simulación está guiada por el motor gráfico y en el segundo, está guiada por el motor físico.

2.1 Sincronización utilizando *framerate* fijo en el motor gráfico

En este enfoque la simulación está guiada por el motor gráfico, el cual indica cuando se realizan las actualizaciones del estado de la escena. Se establece un paso de tiempo para el motor gráfico y un paso de tiempo para el motor físico que represente el tiempo transcurrido entre actualizaciones del motor gráfico. Por ejemplo, estableciendo el paso de tiempo del motor gráfico en X segundos; luego es necesario representar ese tiempo con el motor físico. Dependiendo del motor físico, será necesario realizar varias actualizaciones utilizando pasos de tiempo menores (Figura 3).

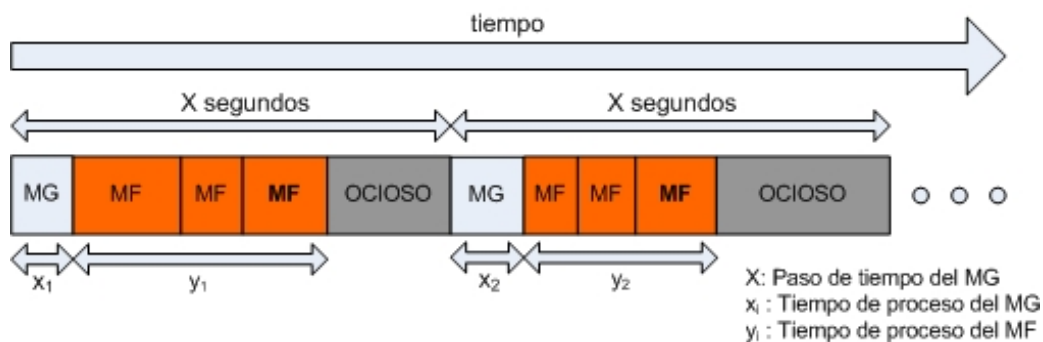


Figura 3: Esquema de interacción utilizando tiempo de actualización fijo en motor gráfico.

2.2 Sincronización utilizando *framerate* variable en el motor gráfico

En este enfoque la simulación está guiada por el motor físico, el cual indica cuando se realizan las actualizaciones del estado de la escena.

El algoritmo de sincronización consta de un acumulador de tiempo, en cada ciclo de la simulación se mide el tiempo transcurrido desde la última actualización y se suma al acumulador; luego, mientras que el valor del acumulador sea mayor o igual al paso de tiempo prefijado del motor físico, se resta este último al acumulador y se realiza la actualización del motor físico. En la Figura 2 se esquematiza un ejemplo, en el cual se acumula tiempo y se realizan actualización conforme avanza

la simulación.

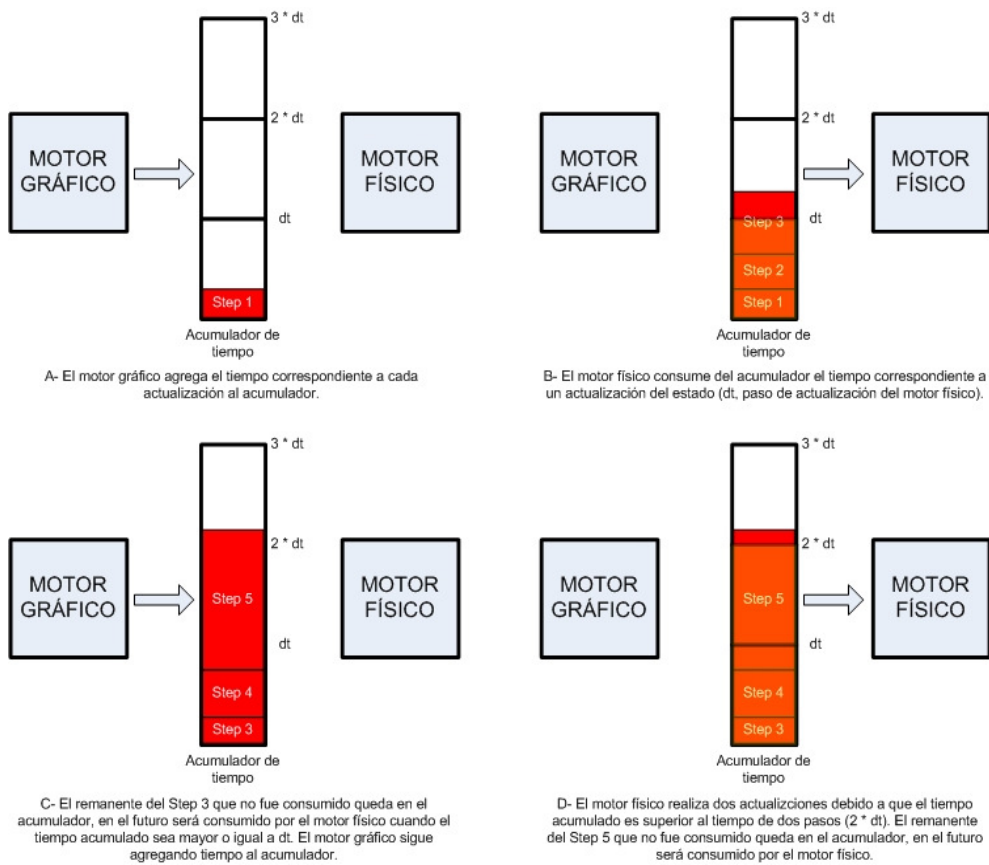


Figura 2: Esquema de interacción utilizando tiempo de actualización variable en motor gráfico.

Ambos algoritmos mantienen la sincronía entre el motor gráfico y el motor físico, asegurando que el tiempo transcurrido en la simulación represente el tiempo transcurrido en la realidad.

3 EJEMPLOS UTILIZANDO LA PAL

A continuación se presentan algunas de las aplicaciones implementadas utilizando la capa de abstracción. Todas ellas fueron implementadas utilizando el motor gráfico Impromptu [D'amato]. En la Figura 4 Izq. se visualiza una escena con 4000 esferas (320.000 polígonos) con movimiento dinámico. Para el modelo estático de la estatua se utilizó una geometría detallada para visualizar y otra aproximada para simular comportamiento. La Figura 4 Der. muestra una escena con 90 cajas con movimiento dinámico que interactúan con un vehículo en una escena compleja.



Figura 4: Escenas complejas con comportamiento físico implementadas utilizando la PAL.

La capa de abstracción provee una interfaz independiente de las bibliotecas físicas, sin agregar sobrecarga de tiempo, reduciendo significativamente el esfuerzo del programador para portar sus aplicaciones a otros motores físicos. Esto es importante, dado que cada motor físico brinda a las simulaciones velocidad de cálculo, estabilidad o precisión.

4 TRABAJOS FUTUROS

En este último tiempo, existe una tendencia de desarrollo e investigaciones de motores específicos y no de propósito general, por ejemplo se pueden citar, aquellos que simulan tejidos, objetos que se rompen, fluidos, vehículos, *ragdolls*, etc. Cada uno de estos motores se encarga del comportamiento que simula, proveyendo precisión, calidad y rendimiento. Por lo tanto es necesario proveer un mecanismo de comunicación entre estos motores específicos. La Figura 7 es una secuencia de imágenes en las cuales un motor físico de propósito general, a través de la capa de abstracción implementada, se comunica con un motor físico específico de simulación de fluidos basado en un modelo de Lattice-Boltzmann, permitiendo al personaje y las pelotas interactuar con la superficie de agua.

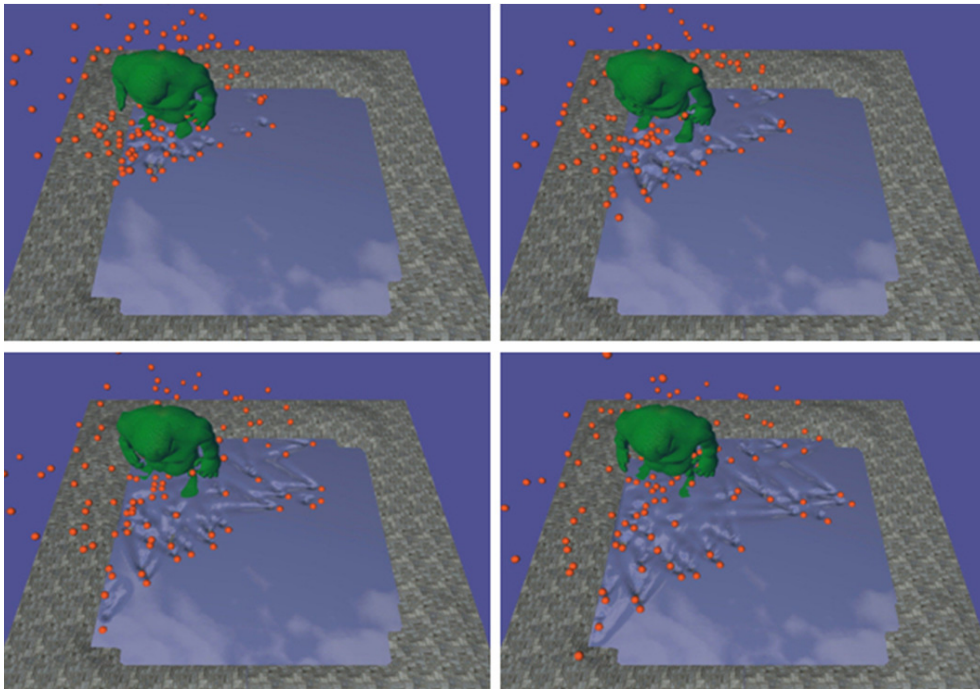


Figura 6: La secuencia muestra distintos momentos de una simulación en la cuál interactúan un motor físico y un modelo basado en Lattice-Boltzmann para la simulación de fluidos.

Actualmente se está trabajando en la simulación de efectos complejos, como un tornado interactuando con objetos del ambiente, mediante la simulación de vórtices, calculando fuerzas de rotación y atracción/repulsión y utilizando un motor físico, a través de la capa de abstracción implementada, para el cálculo de trayectorias y colisiones. En la secuencia de imágenes de la Figura 8 se muestra la simulación del inicio de un tornado.

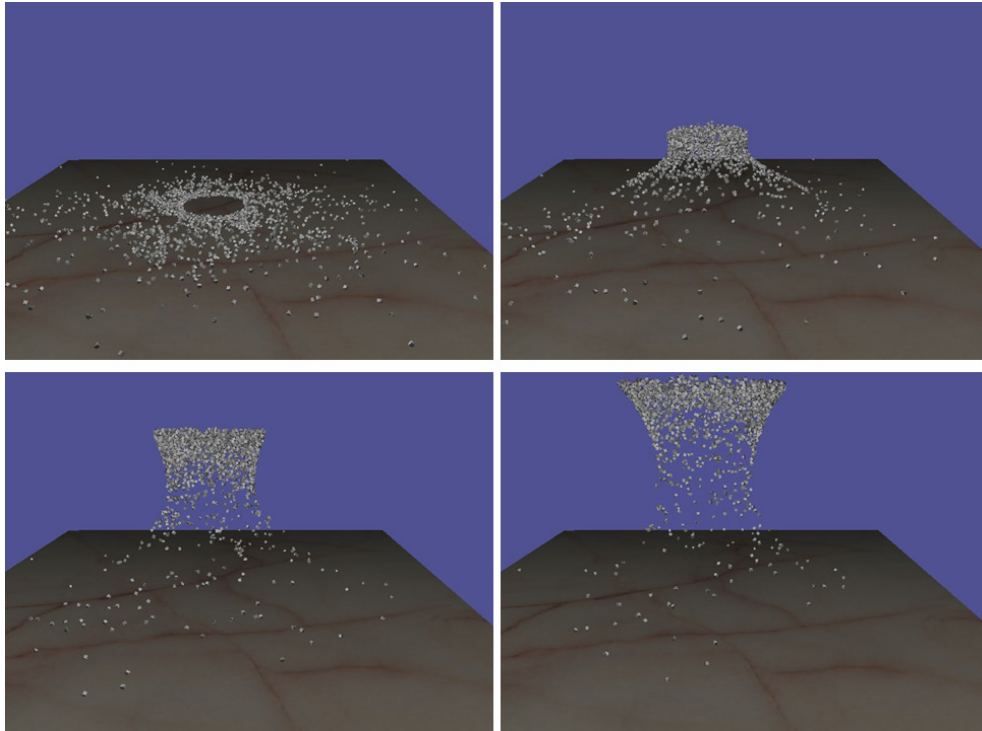


Figura 7: La secuencia muestra distintos momentos de una simulación de un tornado interactuando con 1500 objetos.

5 REFERENCIAS

BOEING A. PAL, Physics Abstraction Layer Home Page. [Online] Disponible en:
<http://www.adrianboeing.com/pal/>. 2005

BPL, BULLET PHYSICS LIBRARY. [Online]. Disponible en: <http://www.bulletphysics.com>

CRISTIAN GARCÍA BAUZA, MARCOS LAZO, MARCELO VÉNERE. “Incorporación de comportamiento físico en motores gráficos”. Mecánica Computacional, Vol. XXVII, p. 3023-3039. ISSN 1666-6070.

D’AMATO J.P, GARCÍA BAUZA C.. Simulación de Escenarios Tridimensionales Dinámicos. Tesis de grado de Ingeniería de Sistemas, Universidad Nacional del Centro. 2004.

NGD, Newton Game Dynamics Home Page. [Online]. Disponible en:
<http://www.newtondynamics.com/>

NVIDIA, NVIDIA PhysX Home Page. [Online]. Disponible en:
http://www.nvidia.com/object/nvidia_physx.html.

ODE, Open Dynamics Engine Home Page. [Online]. Disponible en: <http://www.ode.org/>

FISCHER A., REINOT A., STREETER T.. OPAL, Open Physics Abstraction Layer Home Page. [Online] Disponible en: <http://opal.sourceforge.net/>. (2005)

TPE, Tokamak Physics Engine Home Page. [Online]. Disponible en:
<http://www.tokamakphysics.com/>