

Práctica 6 [OPCIONAL] – Motores Físicos

Duración: 1 Sesión

Objetivos:

- Comprender las posibilidades de los motores físicos para componer escenas con comportamientos físicos realistas.
- Manejar un motor físico 2D sencillo (*Física*) dentro del entorno de Processing.

Entrega: se subirán los ficheros de Processing correspondientes a la tarea 4.

Introducción

Un motor físico es una librería software que tiene implementadas instrucciones para crear objetos bidimensionales o tridimensionales en un mundo virtual que obedece las leyes físicas que conocemos (o al menos algunas de ellas). Estos objetos están sometidos a fuerzas como la gravedad y la fricción. Igualmente son susceptibles de estar sometidos a otras fuerzas que cambian sus estados cinemáticos (velocidades). Los cuerpos son considerados como sólidos rígidos con masa y volumen. Por tanto, experimentan fuerzas y torques, por lo que cambian sus momentos lineales y angulares continuamente.

Existen muchos motores físicos, y su uso es habitual en simulación. NVidia PhysX, Bullet, ODE, Box2D, etc. son algunas librerías que permiten trabajar con objetos con comportamiento físico. Unity3D también incorpora esta posibilidad.

En esta práctica vamos a conocer la librería *Física* que viene incorporada en Processing. *Física* es un motor 2D que permite integrarse con el código de Processing de tal manera que las escenas creadas pueden incorporar el control del ratón y teclado. Es decir, se siguen obedeciendo órdenes como *mousePressed*.

Física es en realidad un *wrapper* de la librería Box2D, que permite utilizar esta última de una manera mucho más sencilla. Puedes encontrar buenas referencias a esta librería en las páginas:

www.ricardmarxer.com/fisica
mycours.es/fisica

Para instalar la librería en tu ordenador personal, deberás añadirla a la colección de librerías de Processing. Para ello ves al menú de 'Herramientas/Añadir Herramienta' de Processing y añade la librería.

Actividad 1

En esta actividad se trata de instanciar objetos que estén sometidos a contacto unos con otros y por tanto existan fuerzas de fricción, además de la fuerza peso. Se proporciona un código fuente llamado **Actividad1.pde** que simula un conjunto de cubos aleatoriamente apilados unos encima de otros.

Observa cómo el motor físico genera las interacciones entre los cubos. Verás que los cubos no se paran y colisionan continuamente. ¿Por qué?

Modifica las características físicas de los cubos de manera que éstos finalmente acaben estabilizándose al cabo de un tiempo.

Como sabes, la detección de colisiones es un proceso que consume bastantes recursos de computación. Una técnica para aligerar este proceso consiste en desactivar (es decir, no considerar) el análisis de colisión de los objetos que no estén en contacto con otros objetos. En la aplicación puedes ver que cuando un cubo no está en contacto con ninguna otra pieza (salvo el suelo), cambia de color para indicar que está desactivado a nivel de estudio de colisiones. Esta técnica es de uso general en los motores físicos.

Actividad 2

Basándose en el ejercicio anterior, crea un programa que apile una serie de objetos rectangulares formando una torre. Ajusta las características físicas de los objetos para que la torre sea estable.

Actividad 3

En esta actividad se pretende comprobar que el motor físico utilizado es capaz de simular la conservación del momento angular entre dos cuerpos que están conectados con una varilla rígida de distancia variable. Como sabes de física, el momento angular de dos cuerpos que giran solidariamente alrededor de un centro de masas se conserva si no actúan fuerzas externas. Esto se traduce en que el producto del tensor de inercia del cuerpo en cada instante por su velocidad angular se mantiene constante ($I(t_1) w(t_1) = I(t_2) w(t_2)$). Si la configuración del cuerpo cambia (como ocurre con dos cuerpos unidos por una barra rígida cuando ésta cambia de longitud), la velocidad angular del sistema cambia para mantener el producto constante. Esto es lo que ocurre con los patinadores cuando están rotando sobre sí mismos y abren y cierran sus brazos.

En esta actividad vamos a representar un sistema de sólidos rígidos formado por dos

esferas (en realidad, círculos, porque trabajaremos en 2D) solidarias. Para unir las dos esferas necesitamos hacer uso de las uniones (*joints*), que son los elementos básicos para la construcción de escenas en los motores físicos. El proceso que necesitaréis hacer es el siguiente:

- Inicializar la librería y crear un mundo (*FWorld*).
- Crear dos cuerpos de tipo *FCircle*.
- Configurar sus propiedades para que no haya fricción angular. Usa la función *setAngularDamping()*.
- Generar una unión entre ellos de tipo *FRevoluteJoint()*.
- Modificar las propiedades de la unión (*joint*) para que no pueda girar. Usa *setEnabledLimit()*, *setLowerAngle()*, *setUpperAngle()*.
- Aplicar constantemente una fuerza que produzca un giro a uno de los dos cuerpos con *addTorque()*.
- Atendiendo al evento de pulsación de ratón:
 - Eliminar la unión.
 - Cambiar la posición de uno de los dos cuerpos.
 - Crear una nueva unión con la nueva distancia entre los objetos.

Observa cómo la velocidad angular del sistema cambia al variar la distancia entre los objetos al hacer *click* con el ratón.

Actividad 4

Construye una escena con las primitivas que ofrece *Física*, donde existan colisiones y objetos compuestos de objetos más simples (usando **uniones**) que estén sometidos a fuerzas y a fuerzas angulares (torques). Esta actividad es libre. Experimenta con la librería para crear algún escenario original interesante, a ser posible interactivo.

Entrega

Se deberá entregar el código de la actividad 4, que se podrá entregar con posterioridad a la sesión de prácticas. Se entregará el código y el material adjunto que se considere adecuado (videos, documento, etc.).