



Sólidos Rígidos PhysX

Práctica 1

En esta práctica vamos a usar PhysX para realizar la simulación y ver cómo se comporta el sistema profesional de física de nVidia. Con su uso veréis que es muy parecido a la forma en que hemos implementado nuestros sistemas anteriores.

Para empezar, adaptaremos el generador de partículas que hicimos para que en lugar de generar partículas genere sólidos rígidos gestionados por PhysX. Pondremos un máximo de elementos generador para no saturar el sistema. Una vez llegado a ese máximo, dejaremos de generar más.

Tenemos dos tipos de sólidos rígidos, los dinámicos y los estáticos. Los estáticos están pensados para que no se muevan, en lugar de definir una masa infinita existe una clase distinta para facilitar el uso. Los dinámicos son los que se moverán y sobre los que se aplicarán fuerzas.

Las clases para gestionarlos son: `PxRigidStatic` y `PxRigidDynamic`.

Para crear uno estático llamaremos a: `gPhysics->createRigidStatic()` con un transform como parámetro.

Para crear uno dinámico llamaremos a: `gPhysics->createRigidDynamic(localTm)` con un transform como parámetro.

En cualquier de los dos casos llamaremos a la función miembro `attachShape` para unir el sólido rígido con un shape, que podemos crear como hacíamos hasta ahora en nuestras prácticas.

Posteriormente para configurar su masa podemos llamar a: `PxRigidBodyExt::updateMassAndInertia`. Con esta función especificamos su densidad, que junto con el volumen definido mediante las dimensiones del shape permite calcular la masa.

Una vez creados hay que añadirlos a la escena mediante esta llamada: `gScene->addActor()`. Pasando como parámetro el sólido rígido creado.

Para que se pinten será necesario crear un `RenderItem`, igual que en prácticas anteriores, solo que en lugar de un `Transform` le pasaremos la instancia de sólido rígido que hemos creado.

Lo primero será que la gravedad afecte a esos sólidos rígidos. En PhysX la gravedad se trata como algo separado. Se puede configurar de forma global para toda la escena poniendo un valor de `Vector3` en esta variable:

```
sceneDesc.gravity
```

Esto debe hacerse en `initPhysics` antes de la llamada a `gScene = gPhysics->createScene(sceneDesc);`. De lo contrario no se tendrá en cuenta.

Para ver cómo lo que creamos reacciona podemos crear un suelo, un sólido rígido estático, para que lo que genere el generador acabe por posarse en ese suelo.

Con ello podemos adaptar nuestro generador de fuerzas de explosión para que se aplique también sobre los sólidos rígidos. Para aplicar una fuerza sobre un sólido rígido las clases tienen una función `addForce`, a la que se pasa un `Vector3` con la fuerza (dirección e intensidad). También tenéis funciones para aplicar Torque por si quisierais usarlas para extender la práctica.

Para la gestión de los nuevos sólidos rígidos se ha actualizado el esqueleto presente en el campus virtual para que tenga las configuraciones necesarias para soportarlos.

Podéis consultar más información sobre cómo usar las clases o cualquier otra información sobre PhysX en su documentación que está presente on-line:

<https://docs.nvidia.com/gameworks/content/gameworkslibrary/physx/guide/Manual/RigidBodyOverview.html>

<https://docs.nvidia.com/gameworks/content/gameworkslibrary/physx/guide/Manual/RigidBodyDynamics.html>

<https://docs.nvidia.com/gameworks/content/gameworkslibrary/physx/guide/Manual/Index.html>