

Práctica 2: Sólidos Articulados

(Asignación 02/03; Entrega 21/03 a las 23:59; Test presencial 23/03 en clase)

Esta práctica tiene algunas partes sólo para alumnos con la asignatura completa

¿Cómo entregar la práctica?

Se habilitará una entrega por Aula Virtual. Se deberán subir los ficheros PhysicsManager.cs, RigidBody.cs y PointConstraint.cs, los tres comprimidos en un único fichero.

No se han de enviar más ficheros que los tres listados. La práctica está preparada para que no necesite librerías adicionales. Todo el código adicional que se precise ha de estar incluido en los ficheros a entregar. La práctica se testeará intercambiando dichos ficheros, compilando y ejecutando, con lo cual no funcionará si se precisan otros ficheros.

Especificaciones Generales:

Se proporciona la escena de Unity Exercise1.unity, con assets visuales a los que se ha añadido comportamiento de sólido rígido. El código incluye un motor de simulación, un interfaz para un objeto simulable, un interfaz para un objeto restricción, e implementaciones concretas de sólido rígido y restricción puntual. Uniendo objetos sólido rígido y objetos restricciones se pueden simular sólidos articulados.

Se ha de programar el comportamiento de la simulación de sólidos rígidos con restricciones, siguiendo distintos algoritmos de resolución. Concretamente, se han de programar los solvers y métodos de integración, los métodos de evaluación de las restricciones, y los métodos relacionados con fuerzas de los sólidos rígidos.

Los alumnos con la asignatura completa deberán implementar el solver de integración implícita con restricciones débiles.

Programación de integrador explícito con restricciones débiles

A realizar de manera obligatoria por todos los alumnos

Esta parte de la práctica requiere la programación de todos los métodos necesarios para implementar un solver de restricciones débiles con el integrador Euler simpléctico. En este caso, las restricciones han de añadir fuerzas al vector global de fuerzas.

El sólido rígido ha de tener amortiguamiento en las velocidades lineal y angular.

La restricción puntual penaliza la separación de dos puntos. Cada uno de estos puntos puede pertenecer a un sólido rígido, o uno de los puntos puede ser fijo y el otro puede pertenecer a un sólido.

Se han de programar los siguientes métodos:

```
private void PhysicsManager::stepSymplectic()  
public void RigidBody::GetForce(VectorXD force)  
public void PointConstraint::GetForce(VectorXD force)
```

Programación de integrador explícito con restricciones fuertes

A realizar de manera obligatoria por todos los alumnos

En este caso, el integrador de Euler simpléctico ha de incluir restricciones fuertes. Para ello, el solver ha de recoger el vector de restricciones y su Jacobiana, y las restricciones no han de aportar fuerzas de manera explícita.

Se han de programar los siguientes métodos:

```
private void PhysicsManager::stepSymplecticConstraints()  
public void PointConstraint::GetConstraints(VectorXD c)  
public void PointConstraint::GetConstraintJacobian(MatrixXD dcdx)
```

Programación de integrador implícito con restricciones débiles

A realizar por los alumnos con la asignatura completa

Esta parte de la práctica requiere la programación del método de integración de Euler implícito con restricciones débiles. Para ello, se han de programar los métodos de cálculo de Jacobianas de los sólidos rígidos y de las restricciones.

En el sólido, se han de considerar las Jacobianas del par debido al cambio de la inercia y de la fuerza y par de amortiguamiento.

En la restricción, se ha de considerar la Jacobiana de la fuerza de restricción.

Se han de programar los siguientes métodos:

```
private void PhysicsManager::stepImplicit()  
public void RigidBody::GetForceJacobian(MatrixXD dFdx, MatrixXD dFdv)  
public void PointConstraint::GetForceJacobian(MatrixXD dFdx, MatrixXD dFdv)
```