Práctica 1 – Parte a: Masa-Muelle 3D

(Asignación 16/02; Entrega 28/02 a las 23:59; Test presencial 02/03 en clase)

¿Cómo entregar la práctica?

Se habilitará una entrega por Aula Virtual. Se deberán subir los ficheros PhysicsManager.cs, MassSpring.cs, Node.cs y Spring.cs, los cuatro comprimidos en un único fichero.

No se han de enviar más ficheros que los cuatro 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 un objeto de simulación masa-muelle. El código incluye un motor de simulación, un interfaz para un objeto masa-muelle, e implementaciones de nodos y muelles.

Se ha de programar el comportamiento del sistema masa-muelle siguiendo distintos métodos de integración. Concretamente, se han de programar el cálculo de fuerzas y jacobianas, y las rutinas que ejecutan los pasos de integración.

La escena se puede configurar en el editor de Unity, añadiendo nodos y muelles de manera individual, o mediante un script. Igualmente, se pueden fijar nodos de manera individual.

La simulación se puede parar/detener con la tecla ‘P’.

Conviene hacer pruebas del comportamiento con distintos parámetros de simulación, y ver si el comportamiento (p.ej., la influencia del paso de tiempo) es el esperado.

Problema: Programación de los métodos de integración.

Se han de implementar los métodos de las clases PhysicsManager.cs, MassSpring.cs, Node.cs y Spring.cs que se encargan del cálculo y paso de fuerzas, cálculo y paso de jacobianas, y ejecución del paso de simulación completo. A continuación se indica el listado de métodos a implementar y algunas explicaciones.

```
private void PhysicsManager::stepXX()
```

Implementación del método de integración XX, donde XX puede ser Euler simpléctico, midpoint, Verlet, o Euler implícito. Se ha de programar una función distinta para cada método.

A modo de ejemplo, se proporciona el método explícito. Todo el cálculo de los integradores asume una formulación de estado multidimensional, se recibe el estado de los objetos simulables, se realizan los cálculos, y se vuelve a copiar el estado a los objetos simulables.

El objeto ISimulable es un objeto de simulación con un interfaz genérico que permite acceder a vectores de posiciones, velocidades y fuerzas, así como matrices de masas y matrices jacobianas. El objeto MassSpring contiene una implementación concreta de ISimulable, basada en masa-muelle.

Los parámetros genéricos de la simulación, tales como el paso de tiempo o el vector gravedad están disponibles directamente en la clase PhysicsManager.

Para el tratamiento de nodos fijos, es necesario llamar a los métodos ISimulable::FixVector() y ISimulable::FixMatrix() antes de calcular las nuevas velocidades.

Para el integrador implícito, el sistema lineal se puede resolver simplemente llamando a MatrixXd::Solve().

```
public void Node::GetForce(VectorXD force)
public void Spring::GetForce(VectorXD force)
public void Node::GetForceJacobian(MatrixXD dFdx)
public void Spring::GetForceJacobian(MatrixXD dFdx)
```

Son los métodos relacionados con el cálculo y paso de fuerzas y jacobianas.