Fall 2015: ELG5124/CSI5151
Assignment 5

Due: Thursday, December 3rd, 2015, 11:00pm in Virtual Campus

University of Ottawa - Université d'Ottawa

Jochen Lang

1 Dynamic Simulation with Particles or Rigid Body in Chai3D

In this assignment, you will simulate an object falling down onto a plane and bouncing back. You can either simulate a rigid body with rigid body simulation and forward Euler integration or with a particle system (i.e., mass points connected with springs) with at least three non-co-linear particles (more are better) and an integration scheme of your choice. The collision response in this assignment should be simple. You can either implement with a coefficient of restitution or with an impedance-based force. A Force calculated based on the penetration depth as in haptics.

Your task will be to implement the integration scheme (either for multiple particles in a spring-damper system or for a single rigid body). You are not allowed to use any existing physics engine like ODE or GEL but you can (and should) use all other parts of Chai3D including collision detection, drawing routines etc. It will help solving the assignment to base your solution on the *12-polygons* example of Chai3D.

Note that you will have to present your results in class on Dec. 4th or Dec. 8th, 2015 (see separate sign-up sheet).

2 Construct the Object [2]

You will need to either load a simple particle system, i.e., Section 2.1, *or* a rigid body, i.e., Section 2.2.

2.1 Particle System

Implement a particle (spring-damper) system with at least three particles. You will need to draw the particle system as a set of connected lines representing the connecting springs of the particle system.

The simplest way to draw the particle system is as a set of cShapeSphere and cShapeLine (you don't need to draw the coils of the spring, just a line connecting two particles). This way you only have to decide where your particles go and what lines use for spring connections. The drawing is all taken care off. There are no marks for visual appearance in this assignment.

2.2 Rigid Body

You can simply load a rigid body from file for this part. However, you have to be able to calculate or know the body's inertia tensor (e.g., a box, cylinder or similar). You will have to make sure that you have a local coordinate system with the origin at the cent of mass.

3 Implement the Integration Loop [4]

We will use the haptic update loop to implement the integrator. (Interaction with the haptic device is optional but will be considered a bonus in your presentation). As forces you will need to implement gravity and the spring forces (see course notes) in case of the particle system. The integrator, as well as any update to the positions of the particles, need to be handled in the `updateHaptic()` callback (for an example see `12-polygons`.

4 Collision Response [4]

The simplest way to handle collisions is through a coefficient of restitution 4.1, *or* impedance-based 4.2.

4.1 Coefficient of Restitution

The coefficient of restitution describes how much of the velocity of the colliding object is *mirrored* at the point of collision. Note that a collision will not directly generate any forces but will instantaneously change the velocity of your object. For a particle system, this will effect only the particle(s) which collide, the remaining particles will respond through spring forces. For a rigid body, you will need to update both, the linear and angular velocity. The angular velocity is given by the fact collision away from the centre of mass reverse the sign of the linear velocity at this collision point.

4.2 Impedance-Based Response

Impedance-based collision response will result in a soft contact. You will need to create a force proportional to the penetration depth of the object. This is simple in the case of particle where each particle is handled as a mass point and independent. In general, this is non-trivial for rigid objects. However, for this assignment you can make the following simplification: Only the point (vertex) with the largest depth will create force. Force application will work very similar to haptic rendering.

5 Submission

Your assignment submission must consist of a *short* description of your particle system or rigid body and how you have implemented the integration loop. You should submit all source files for your implementation and *no* other files.