Advanced Computer Graphics
Exercise 7 - Rigid Bodies

Handout date: 17.11.2015
Submission deadline: Monday, 23.11.2015, 11:00pm

Note

Undeclared copying of code or images (either from other students or from external sources) is strictly prohibited! Any violation of this rule will lead to expulsion from the class. No late submission allowed.

What to Hand In

Solve the exercise in groups of 3. Hand in a compressed zip file renamed to `Exercise7-firstname_lastname-firstname2_lastname2-firstname3_lastname3.zip` that contains the following files:

- `Rigid_body_viewer.cpp`
- A readme file (.txt or .pdf) containing the following: a description of your solution, how much time you needed, encountered problems (if any), and answers to the questions.

Framework

In this exercise, you will simulate the interactions of a rigid body. The framework you will be working with is similar to the previous one. You are given a QtCreator project file that allows you to build and run the system.
To control the simulation, use the following keyboard inputs:

- Once the system is running you can use the number keys `1`, `2`, and `3` to select one of the predefined scenes.
- `t` and `T` let you increase and decrease the time step of the simulation.
- Pressing space starts/pauses the simulation.

You are encouraged to experiment and create new scenes yourself. All your implementation tasks are in `Rigid_body_viewer.cpp`.

6.1 Forces on the rigid body (20 points)

Implement the method `compute_forces()` to compute forces on the rigid body. The forces should include:

1. Gravity: $Fg = 9.81m$.
2. Damping forces for both linear motion and angular motion. The damping coefficients are given as `damping_linear` and `damping_angular`.
3. Mouse damped spring force. The mouse interaction is similar to what you have implemented in the previous exercise. However, when a rigid body is concerned, you should modify both the force and the torque acting on the body.

6.2 Boundary collisions (20 points)

Implement the method `impulse_based_collisions()` to consider the impulse-based collision response on the boundary. Again, please consider both linear and angular movement of the rigid body.

6.3 Time integration (20 points)

In this question, implement the explicit Euler time integration in the method `time_integration`. You have to update the position, orientation and velocities of the rigid body. To update the positions of component particles, call `update_points()` in the end.