Game Physics – Programming Exercise

Exercise 4 – *Open Project*

Task Overview

The goal of this open project is to extend your previous simulation code and generate several interactive and interesting physical demos. For this, create **a coupled simulator** that combines at least two of the previous exercises (or SPH). Which two is up to you.

Recommendations & Tips:

1. **Force based coupling:** You can implement a force based coupling between two of your previous simulations. E.g., add forces from a mass-spring system to your rigid body simulation, and implement a collision-detection between its mass points and the boxes (either as force or position constraint). With this coupling, you can have interactions between rigid bodies and elastic bodies (mass spring system).
2. **Coupling with the diffusion solver of Ex3:** This variant is a bit more difficult, but nonetheless interesting: you will need a mapping to and from the grid, and some creativity. You can, e.g., try to add temperature and diffusion effects to couple the simulations. With grid-based motions, you can try to model something like flames.
3. **SPH simulator:** Alternatively, you can implement a fluid solver with SPH, as discussed in the lecture, and couple one of the previous solvers with it. In an SPH simulator, for a given particle, we find its neighborhood, and calculate the particle densities, then evaluate pressure and gravity forces, and integrate positions and velocities in time with an Euler step. You can start with a simple n^2-version, and then add a grid as spatial acceleration data structure to speed up your solver. The following thesis contains useful tips and hints for implementing an SPH solver: http://image.diku.dk/projects/media/kelager.06.pdf
4. Alternatively, you can be creative – but also be realistic (don't try to do too much), and always start with tests that are as simple as possible.

Demo requirements:

- **Coupled simulation:** Implement and test your coupled simulator.
- **Demo Scene:** Once your simulator works, create a complex scene with it that is as large scale as possible (i.e. involves as many points/particles/objects as your simulator can handle).
- **Optional - Game Logic:** as an optional task you can implement a simple game logic in your simulation. Think of a play target and scoring mechanism for your simulation. However, this is not mandatory.

Submission

In contrast to previous exercises, you don't need to submit exercise 4. Instead, you should do a presentation and demo of your project during the lecture (details below). Note that because of this presentation, exercise 4 will not be part of the short oral tests that will take place individually at the end of the semester. You will need to submit a full copy of your executable and code after your in-class presentation.

Both presentations will take place online, via BBB via the following link, and additional details will follow: **https://bbb.rbg.tum.de/nil-pnp-s8i-wro**

- **In-class Demo:**
present your **open project** in **the lecture** on **Mon. Jan. 31st. 2022.** Briefly explain how you coupled different simulators to the class, and then demonstrate it to show interactions between different objects. In total, use less than **5 minutes**.
- **Individual Presentation**:
present your **first three exercises on Feb. 7th. (Mon.) and Feb 8th. (Tue.) 2022**. You should be familiar with the simulation models and your source code of all three exercises. Make sure you have source code and executables ready for all three submissions.