# Homework 2: Spatial Data Structures

The goal of this second assignment is to implement spatial data structures to effectively improve the performance of geometric operations within numerical software.

You are given a working implementation of software simulating spheres moving around in a box, affected by gravity and collisions between each other. Your goal is to accelerate the code used to find these collisions.

The deadline for the assignment is the end of the semester, December 20, 2024. You may submit your code by email at kilian.verhetsel@uclouvain.be. You may also use this email to ask further questions about this homework.

## Program Structure

Your program consists of one file `homework.py`, where you must replace the `find_intersections` method with an efficient implementation. The provided implementation checks every pair of objects, and is therefore very slow when the number of object gets large. You may use any spatial data structure of your choice to achieve better performance.

You may find that you need to change your data structures after the insertion of new objects, or at the start of the simulation. You may do so by redefining the `object_added` and `__init__` methods, respectively.

You are given the current state of the simulation using a set of NumPy arrays:

- `radii`, the sizes of the $N$ spheres in the simulations;
- `positions`, of shape $(N, 3)$, the positions of these spheres;
- `rotations`, of shape $(N, 4)$, the orientations of these spheres encoded as a quaternion [1];
- `velocities`, of shape $(N, 3)$;
- `angular_velocities`, of shape $(N, 3)$.

The method to compute the intersection between a pair of objects is provided to you (`intersect(a, b)`). Your implementation should return a list populated with the return values of this method, with one element per contact point that you identified.

To test your program, run the python file `web-server.py`. This requires the installation of the Python packages `websockets`, `numpy`, and `asyncio`.

You can view and interact with the simulation within your browser, by opening the URL http://localhost:8000. Use the arrow keys and your mouse to move around in the simulation. You can also add more objects from your browser to test the performance of your solution.

[1] https://en.wikipedia.org/wiki/Quaternion