

## Computer Animation Course (Assignment Sheet 3)

In this exercise we have a look at a Collision system. The collision detection is based on AABB (axis aligned bounding boxes) while the collision response is done via a velocity reflection.

### Assignment 2.0 (Prerequisites)

Download the `Assignment_3.tar.gz` file from the repository and unpack it to the Exercise-folder. You will find the relevant files in the `Assignment_2` folder. In addition download the files `bunny.off` and `sphere.off` to the Media folder. Make sure, the program `Application3` compiles and runs onto your system. You should see a viewer window with two models inside. You should be able to rotate the model with the mouse and the right button down. Now get familiar with the code changes in `main.cpp`. All changes are to be done here.

### Assignment 3.1 [5 Points] (Axis Aligned Bounding Boxes)

In this assignment you should implement the method `AABB::setFromVertices(...)`. The parameter is a `std::vector` containing the vertex positions of an object. The method should now create an Axis Aligned Bounding Box for the given vertices as discussed in the lecture.

### Assignment 3.2 [5 Points] (Movement And Collision Detection)

Now you have to implement the method `RObject::collision(...)`. As a parameter you give another rigid object. Test the two object's bounding boxes for collision and return true if they collide, otherwise false. Next, you have to touch the function `idle()`. Each `RObject` has a velocity set in the `initModels()` method. First of all, move all objects in the scene. Then, test each `RObject-RObject` pair, if a collision occurs. In that case implement some reaction to make the result visible. Also reflect the velocity vector of a colliding object to resolve the intersection state.

**Good Luck!**