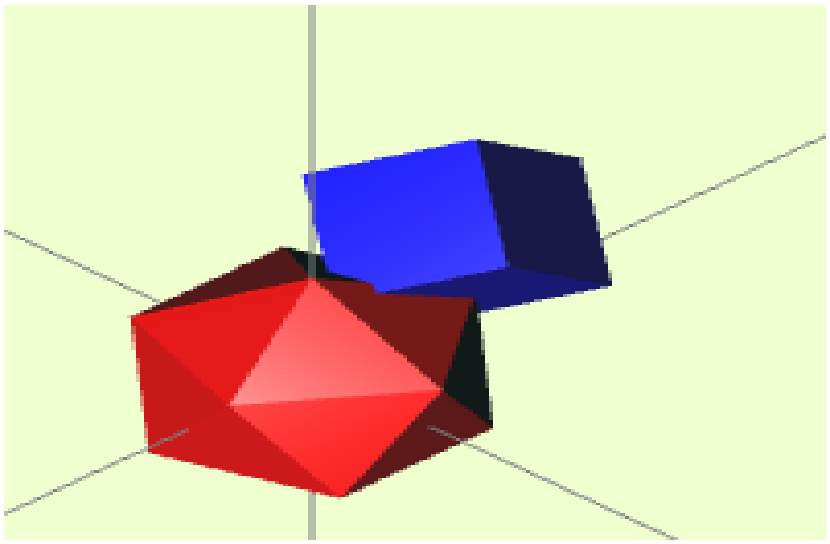
# Assignment 2– Mesh Collision detection

**In this assignment you will implement a tool which efficiently detects collision between two 3D objects (represented by mesh)**

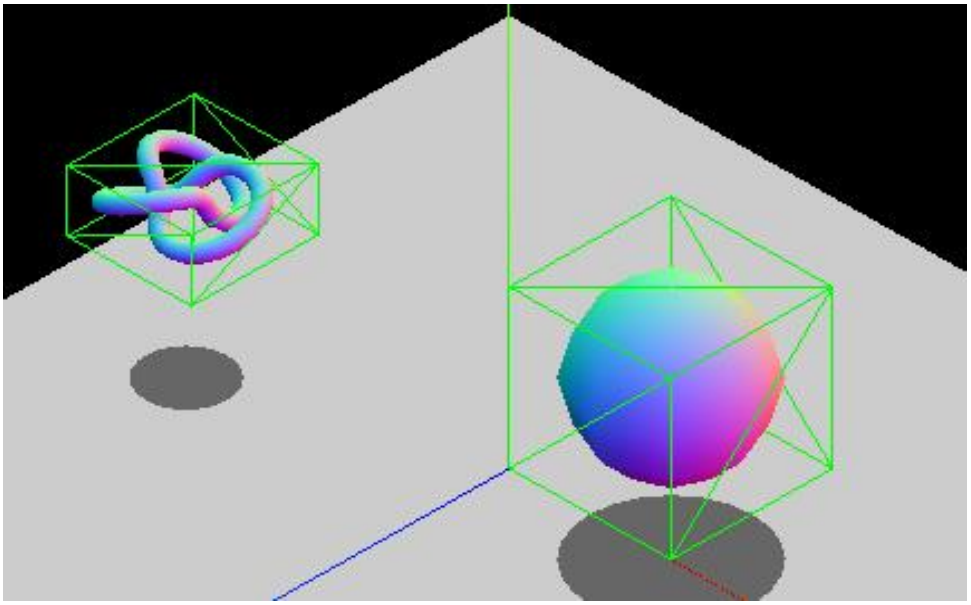


**Set up**

1. Read two identical objects (for example two bunnies) and locate them so they don't intersect.
2. Give one of the object initial velocity in a certain direction which can be changed using the arrows. Both objects can be rotated using the mouse left button.
3. Use the Kd-tree implementation of Libigl AABB to implement Bounding Volume Hierarchy for each object. Start checking collision between the boxes in the head of the trees.
4. Use `Eigen::AlignedBox<double, 3> type` to get box details. You can render the box using overlay edges (as you drew the axis in the previous assignment).

**Algorithm implementation (in each frame)**

5. Start from the biggest bounding box (you can try to start from the last two boxes you succeed to separate on the previous frame).
6. In each step find the relevant bounding boxes to check based on the previous step. Each step goes down the hierarchy in at least one tree.
7. Use transformations and `Eigen::AlignedBox<double, 3>` **`center()`** and **`size()`** functions to find the centers of each bounding box and the distance along each axis to box faces (see obb_sat.pdf). You can find the axes of the box (in scene system) directly from the shape rotation matrix.
8. If any of the 15 checks separates the two boxes, then no intersection occurs. Else, search until you reach the leaves of the tree in both trees. If you can't separate the smallest boxes, you treat it as a collision.
9. During whole simulation you must render:
   a. Two objects (one is moving) full mesh representation
   b. The biggest bounding box of each object.
10. In case of collision stop the moving object and show the smallest bounding boxes which were involved (one in each object).



11. You may use your mesh simplification code to create a simpler version of the mesh for faster collision detection (you must show the full mesh).
12. Other kinds of optimizations will be rewarded.
13. **Submission in pairs to the Moodle. A text file with a link to your repository named ID1_ID2.txt.**
14. You can find explanation for OBB separation on the course site at lecture notes.

## Good Luck!