Script 06

- Geometric Modeling using triangular meshes.
- BufferGeometry: defining vertices and normal vectors.
- ConvexGeometry

5.1 BufferGeometry: defining vertices and normal vectors

Open the folder **01_Ex_Mesh_Models**.

Analyze the example. The scene has **three cubes** and **one point light source**; there is also a **small ambient illumination component**.

Tasks – Mesh representation

- For each cube, analyze how the **mesh vertices** and **unit normal vectors** are defined and organized into triangles. **How many triangles are defined for each model?**

- **Cube 1:** explicit definition of each **triangle**; repeated vertices and triangle normal vectors.

- **Cube 2:** explicit definition of each **quadrangular face**; repeated vertices and face normal vectors. **Vertex indices are used to define each triangle**.

- **Cube 3:** no repeated vertices; one normal vector associated with each vertex. **Vertex indices are used to define each triangle**.

- Render the cubes in **wireframe mode** to visualize the mesh triangles.

Tasks

- **Can you spot any differences regarding the rendered cubes?**

Task – Adding more models

- **Add three tetrahedra** below the cubes, defined in a similar way.

- You can use the **vector product to compute normal vectors.**

5.2 ConvexGeometry

Open the folder **02_Ex_Convex_Hull**.

Analyze the example. In addition to the simple geometries, the scene has a **pyramid** defined by the **convex hull** of its vertices.

Task

- Analyze how the **pyramid** is defined from the set of its vertices.

Task

- **Add more models** to the scene using **ConvexGeometry** and **BufferGeometry**.