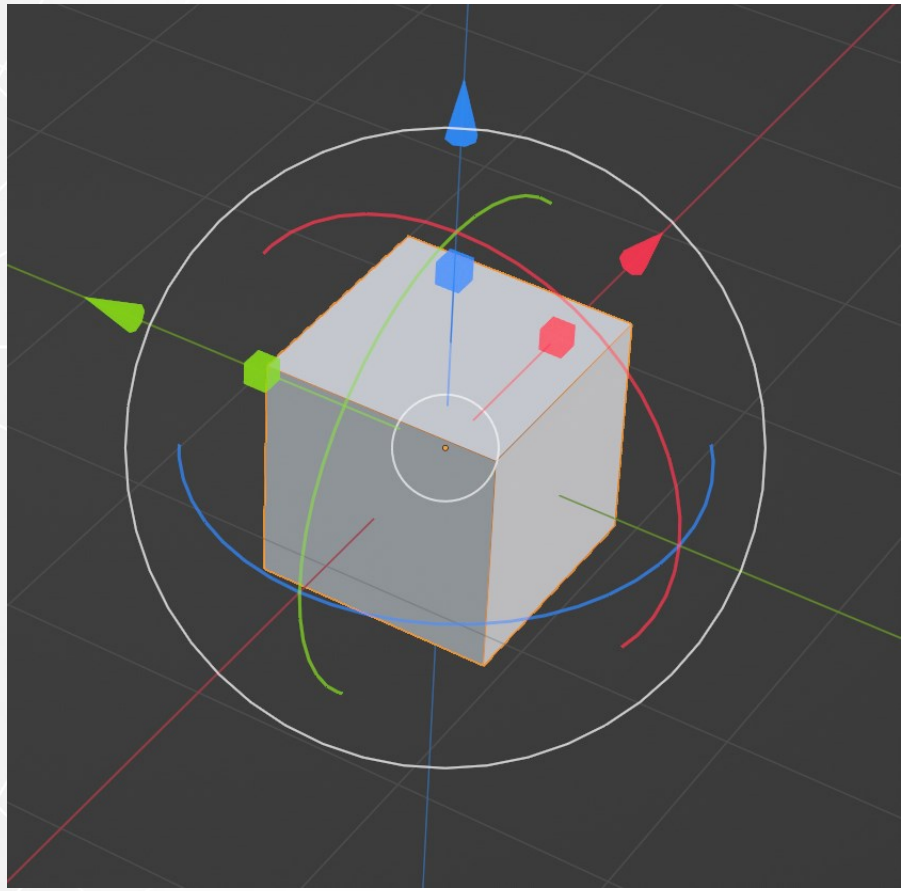
King's Digital Lab

Creating digital tools to explore academic research in new ways.

kdl.kcl.ac.uk

 @kingsdigtallab 

3D Data

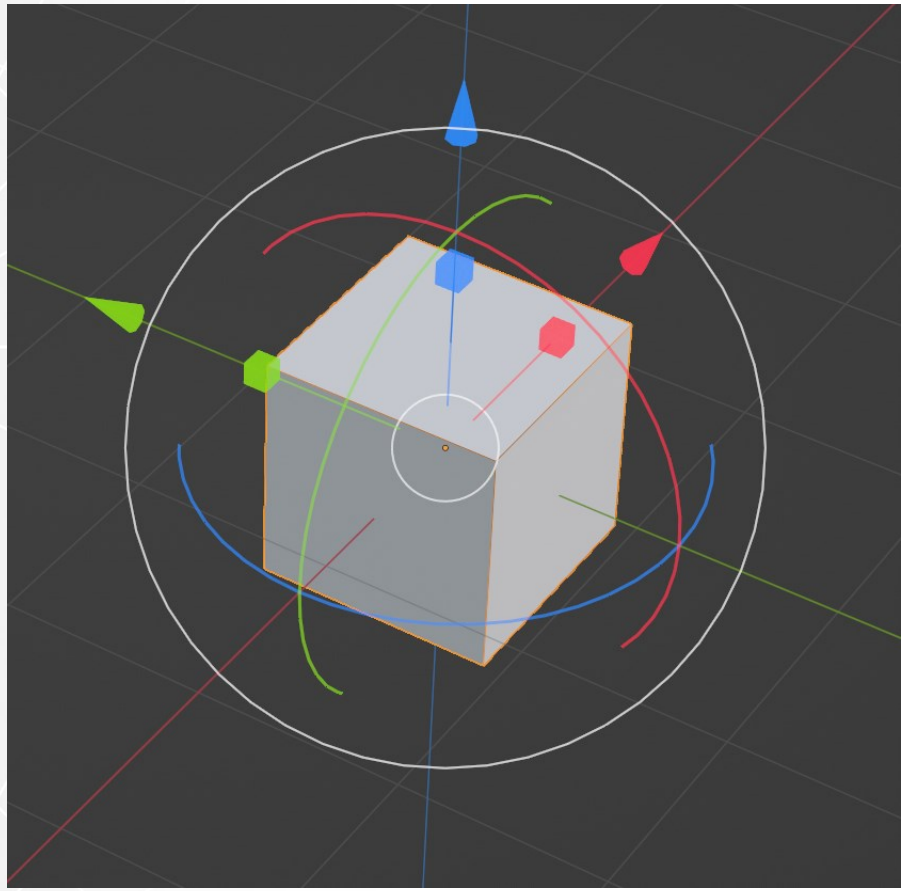


Mesh is made up of
- **Points** (vertices)
- **Edges**
- **Faces**

More vertices mean more work for the computer when rendering.

Typical game assets:
- “Hero” up to 30k triangles
- Set dressing assets can be as small as reasonabe

3D Data



Transforms vary from package to package – *Blender* is “right-handed” with Z up.

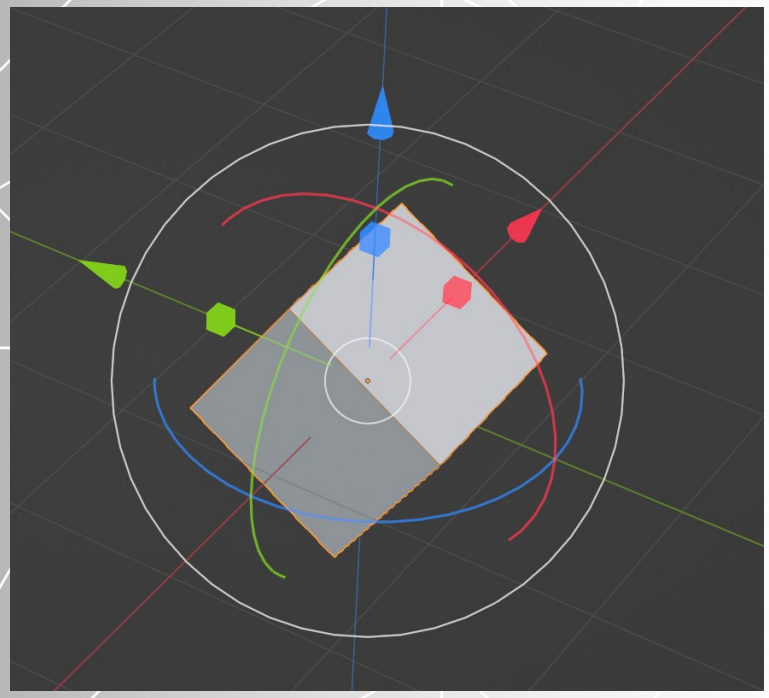
- Y axis +
- X axis +
- Z axis +

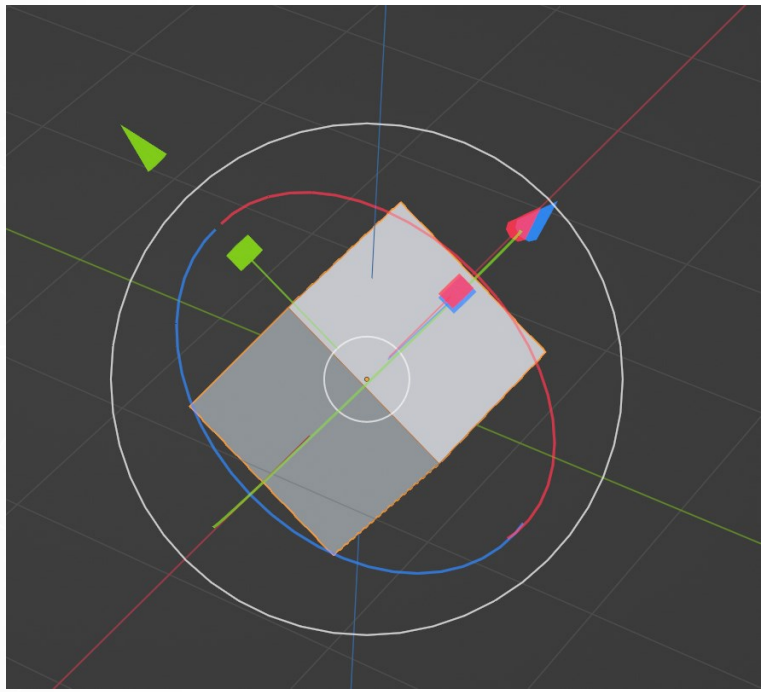
Transforms control

- **Location**
- **Rotation**
- **Scale**

...based on the object origin

3D Data

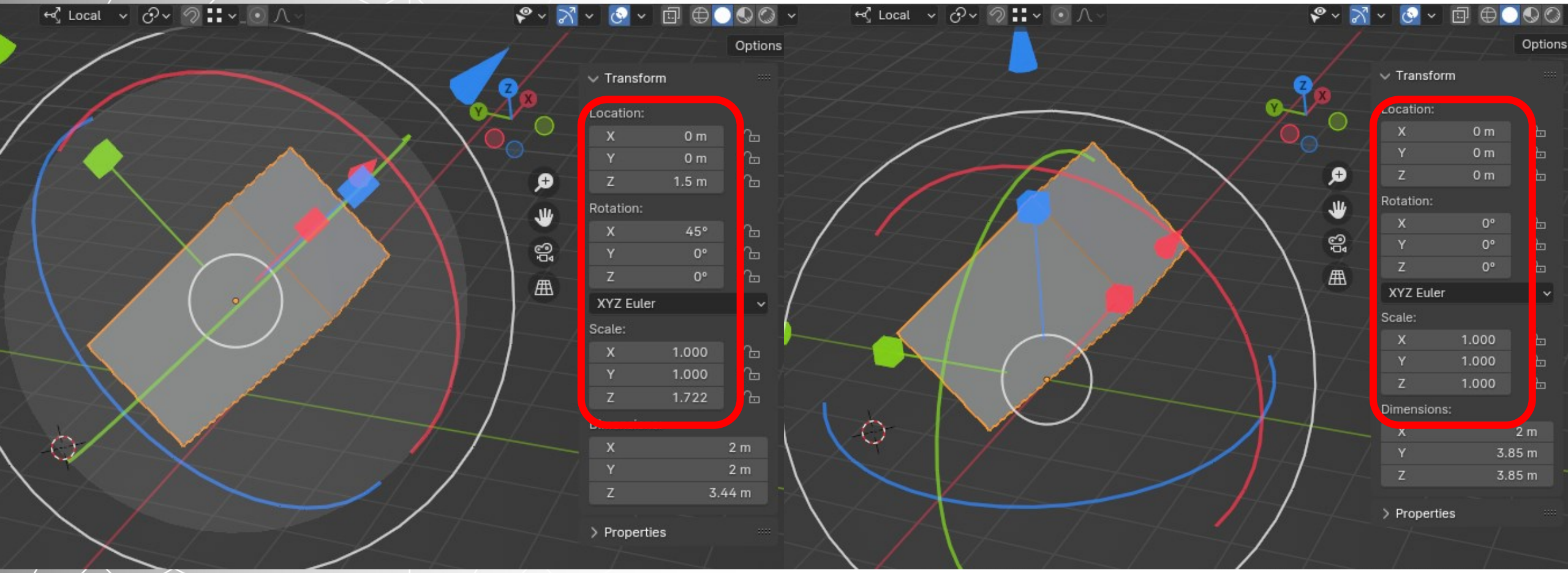




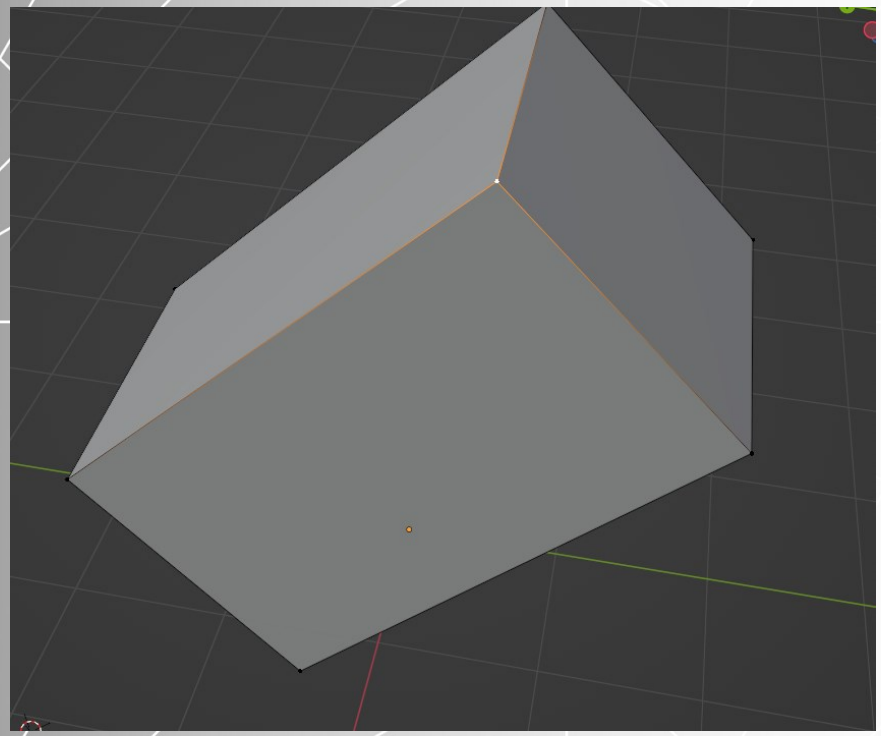
Transforms can be manipulated:

- Globally
- Locally

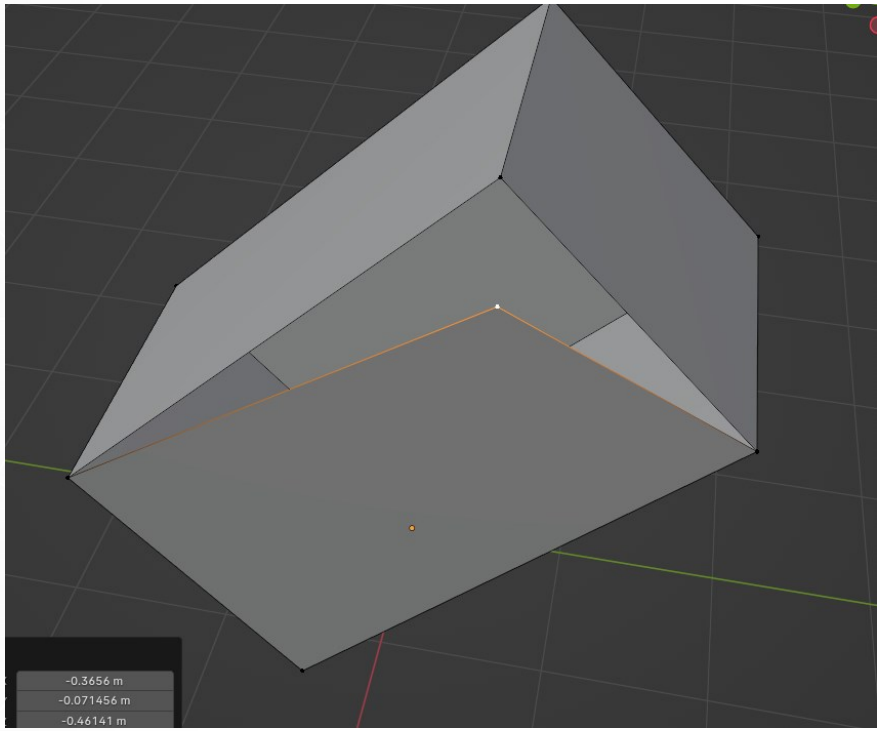
3D Data


Local
Options
Transform
Location:
X 0 m
Y 0 m
Z 1.5 m
Rotation:
X 45°
Y 0°
Z 0°
XYZ Euler
Scale:
X 1.000
Y 1.000
Z 1.722
X 2 m
Y 2 m
Z 3.44 m
Properties
Local
Options
Transform
Location:
X 0 m
Y 0 m
Z 0 m
Rotation:
X 0°
Y 0°
Z 0°
XYZ Euler
Scale:
X 1.000
Y 1.000
Z 1.000
Dimensions:
X 2 m
Y 3.85 m
Z 3.85 m
Properties

3D Data



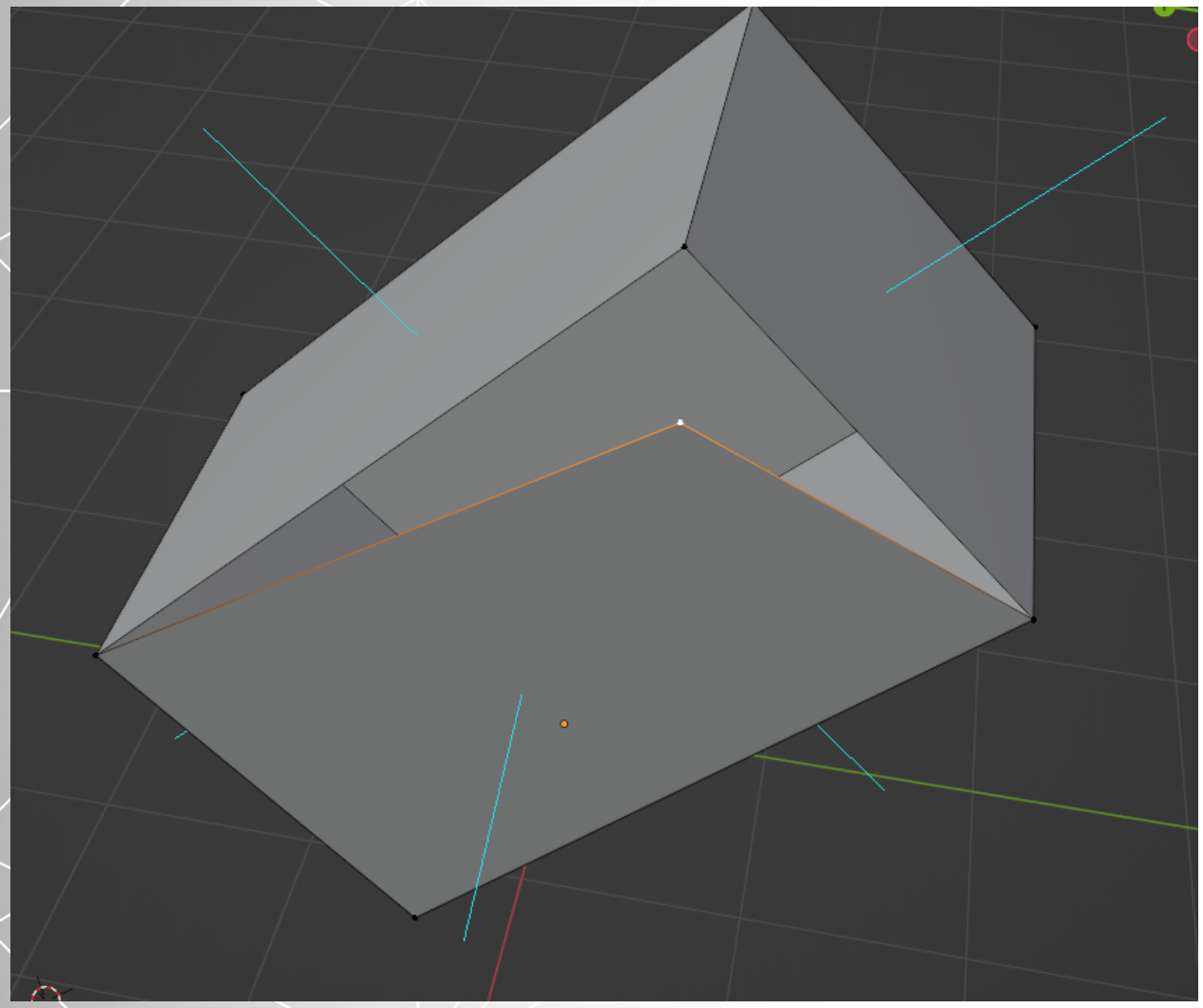
Manifold



Non-manifold

For many purposes – mesh data needs to be **Manifold.** i.e. It should be watertight and have no holes in it.

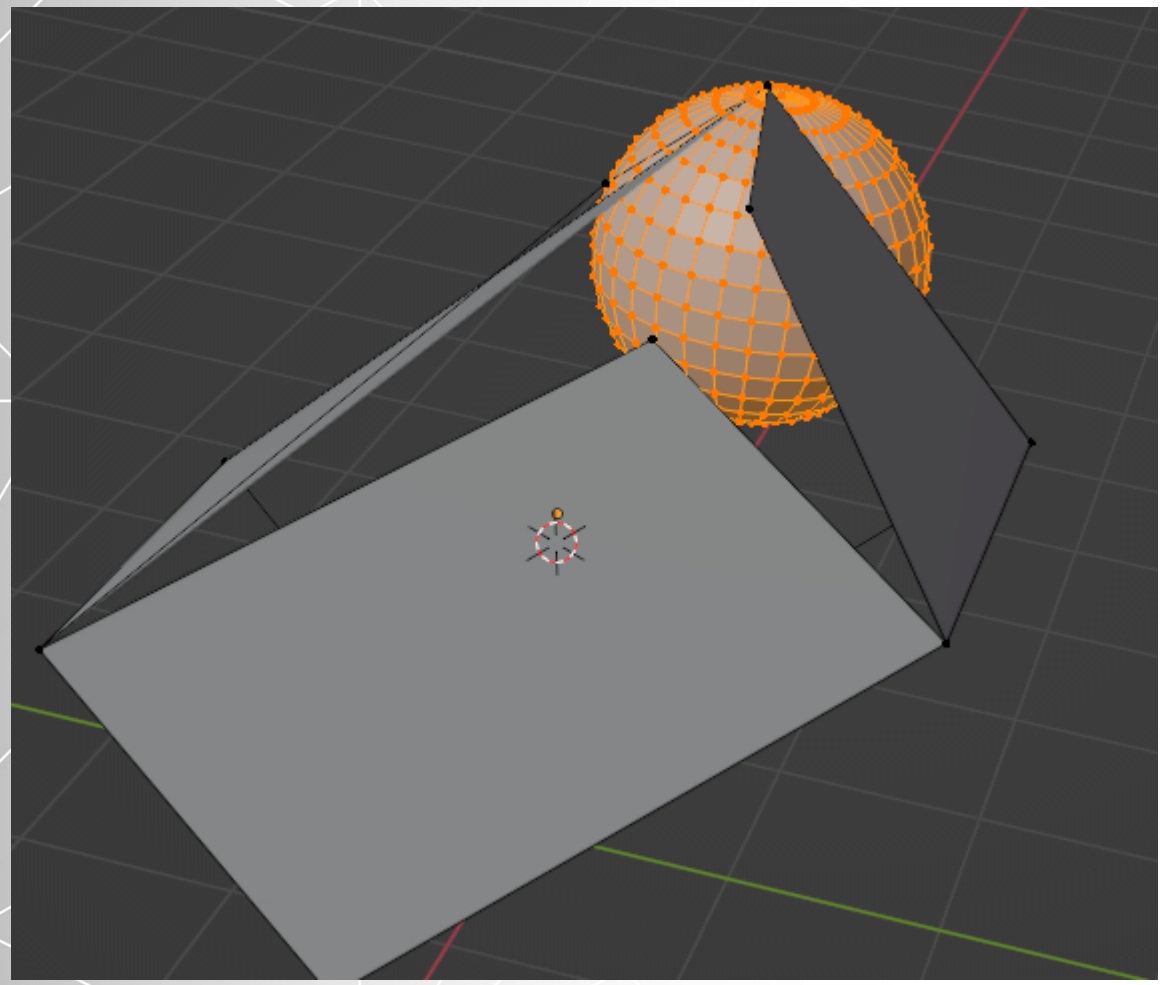
3D Data



The face normal are shown in blue here. They are perpendicular to the face.

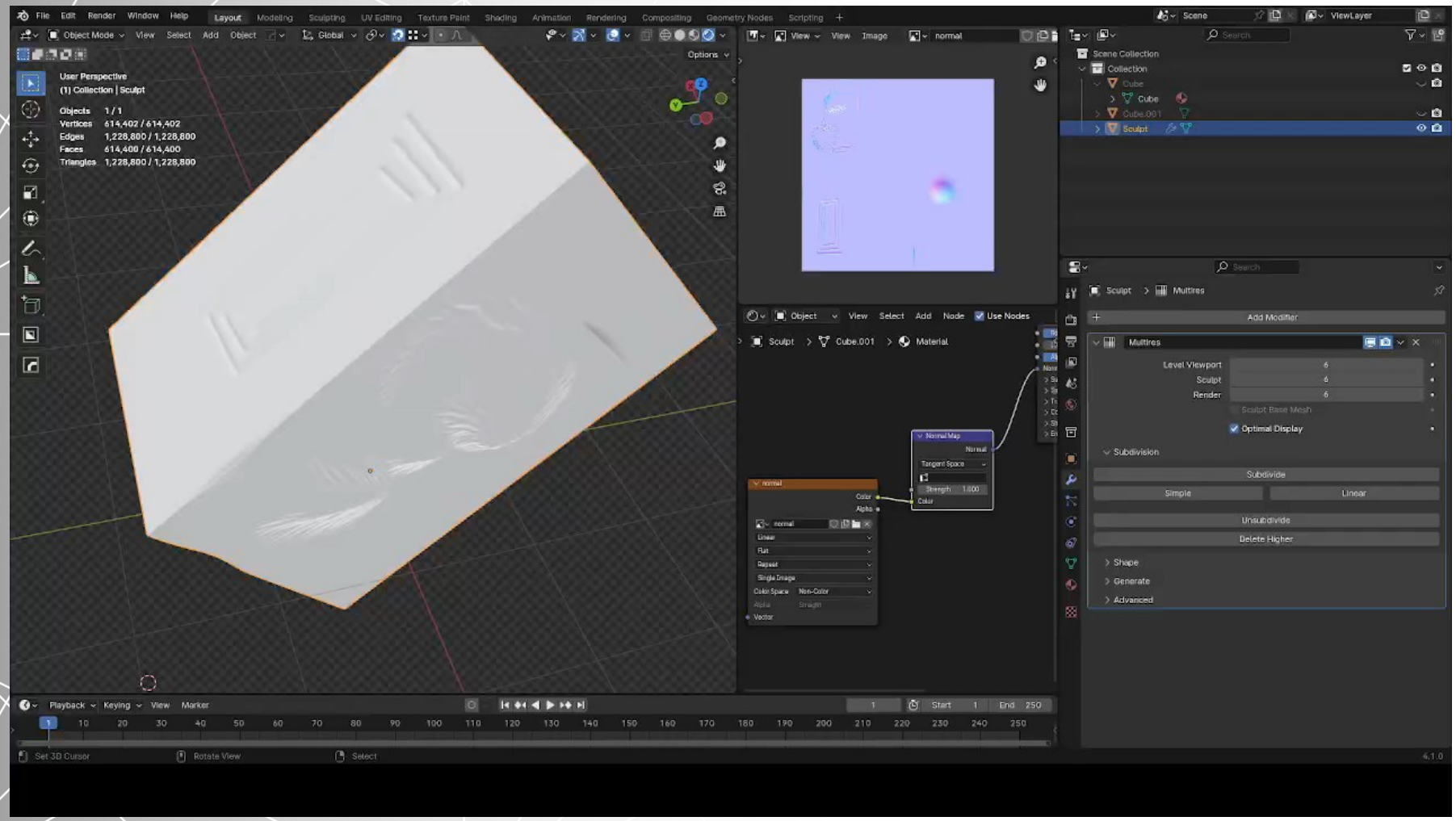
If we know which faces are front we can save valuable processing power by not rendering **back faces.**

3D Data



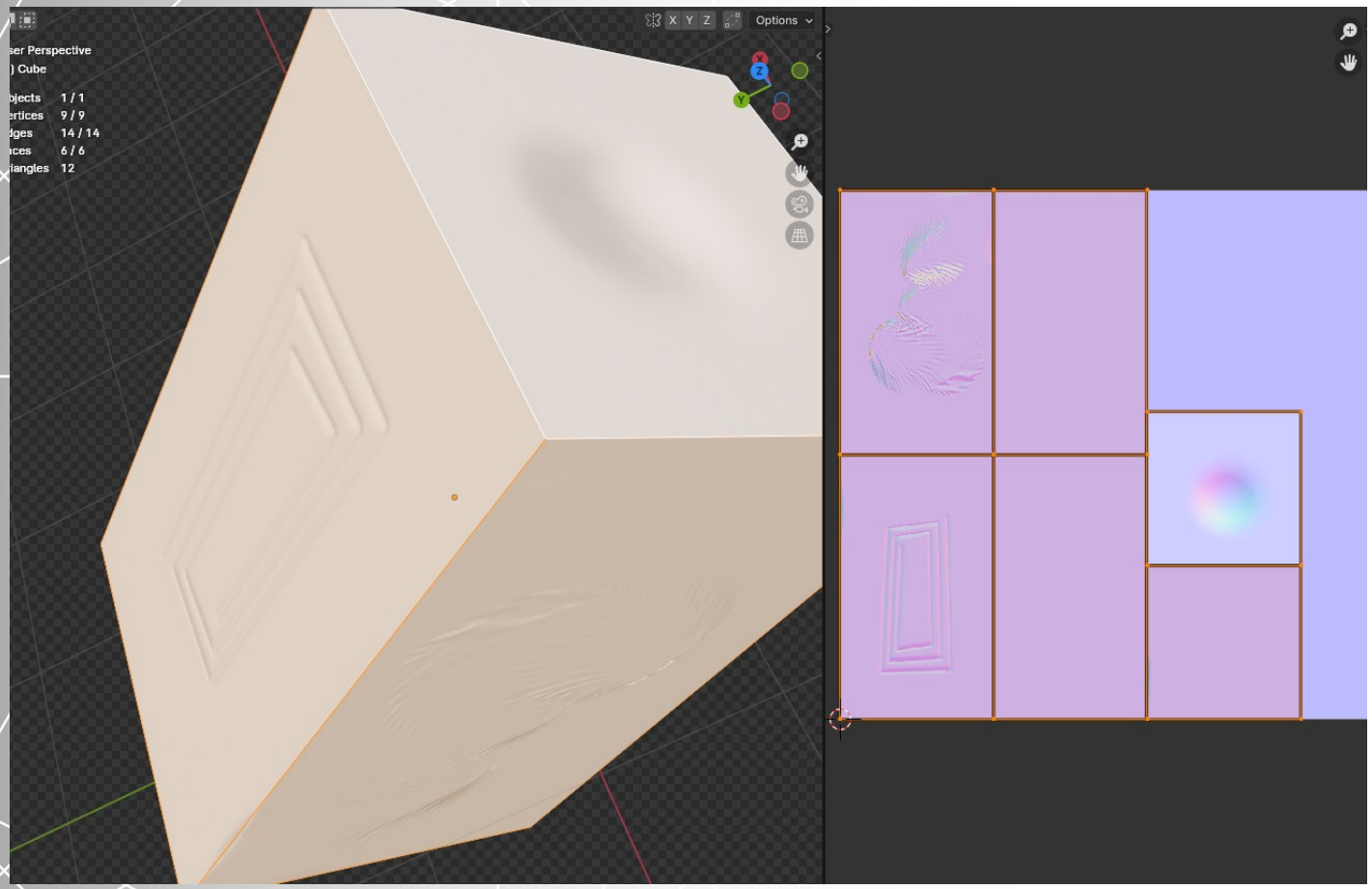
If we know which faces are front we can save valuable processing power by not rendering **back faces**.

3D Data



We can use normal information to create the illusion of complexity.
This object has 614,000 faces – which would be far too many to use in a game setting.
However, each of these faces has a perpendicular normal, and what we can do is project that normal information onto a far simpler object. In face we can project it onto our distorted cube so that when computed light rays hit the cube it gives the illusion of shape and complexity

3D Data



UV maps are the coordinate system for getting textures on to 3D objects.

UV is chosen to signify the relationship of 3D objects to texture, rather than XY which is for 2D use.

Textures are always square (or should be) and a higher resolution texture will give a better render result.

Common sizes:
Low quality 512x512px
Medium quality 1024x1024px
Medium high 2048x2048xx
High 4096x4096px
Ultra-high 8192x8192px