

Vertices, Edges, Faces, and Meshes

Introduction

To understand how objects in Sweet Home 3D are constructed you have to understand the building blocks that are used to compose these objects. This manual explains the terminology used in other manuals so you don't have to read the same explanation again and again.

3D models are build using meshes. A collection of vertices, edges, and faces define a mesh. The terminology is explained with Sweet Home 3D in mind, not 'real' 3D design programs. This means that terms are explained to make you understand how an object is composed without using terms that are technically correct but make understanding more difficult. Look at it as “Meshes for dummies”.

Vertices

A vertice can be seen as a point somewhere in space. A single vertice is a set of coordinates that define the position of that point. In Sweet Home 3D coordinates are defined as X for the horizontal position, Y for the vertical position, and Z for the elevation. These coordinates are relative to the plans 0,0,0 position. In 3D design programs this is the same although the direction of X and Y is sometimes switched.

You can't see a vertice because it has no volume to show. Programs like Blender can show the selected coordinates where a vertice is but on its own it's ust number.

Edges

An edge is a line between two vertices. An individual edge doesn't have volume so like a vertice you can't see it. Again, it's just numbers, now two sets of coordinates. Only the selection of an edge can be shown in a program.

Faces

A face is a 2-dimensional plane between vertices and edges. Here we start to see something! Most 3D models use triangular faces because a triangular face is by definition always in a single plane. There are other types like quad faces (between 4 vertices) but these are often translated to two triangular faces that combines make the quad face.

It's important to understand that faces are the only things visible. You can't see vertices and edges, they are just reference points for faces. Think about two adjoining faces. They share a single edge where they are joined. If you remove one face the edge where they were joined still has a function for the remaining face. If there's no face to keep the edge it will disappear when you remove the face.

Normals

It is important to understand that a face only has one side. Imagine painting a square on a piece of wood. You paint it on one side and there's nothing on the other side of the wood. Imagine that if you would remove the wood and only leave the paint you wouldn't see the backside of the paint, you would see nothing.

There's a complex accurate mathematical explanation how that works but for simplicity this means you can only see a face from one side depending on the direction of the normal.

This also means that if you want to see a face from both sides you need two faces between the same vertices, each pointing to one side.

You probably have seen 3D models that you imported where there was a hole somewhere where you could see through the model. It's most likely that the face at that position points in the 'wrong' direction. You can fix something like that in Blender by 'flipping the normal'.

Look at a normal as a property of a face: the direction a face point to.

Meshes

A mesh defines the shape of an object. A mesh is a collection of vertices, edges, and faces that define that mesh. For example the box object from the Sweet Home 3D catalog is a mesh that is constructed with 8 vertices (the corners), 12 edges (the lines between the corners), and 6 faces (the sides of the cube).

It is most likely that when you import a box exported from Sweet Home 3D into Blender that you will see 12 faces instead of 6: each side is a combination of two triangular faces.

This is because by default faces are constructed as triangles. As long as vertices remain in the same 2-dimensional plane you can have different forms of a single face but unless you specifically tell the program you are using to keep the faces it will split every face into triangles.

So there's the mesh: a 3D object defined with vertices, edges, and faces.

