

# Mesh animations

## Table Of Contents

- [1 Principle](#)
  - [1.1 Definitions](#)
  - [1.2 Current restrictions](#)
  - [1.3 Concept](#)
- [2 Preparation in the 3D program](#)
- [3 Configuration in the content tool](#)

## 1 Principle

### 1.1 Definitions

- Matrix: A matrix is a mathematical construct, which can describe the rotation and translation of a mesh. You can also look at it as a parameter that specifies the position in the space.
- Mesh animation matrix: This describes the target for the animation, that is, how and where the mesh should bend/twist/stretch.
- Base animation: this is the default position for the mesh. Like all meshes, mesh animation meshes are assigned to an animation and follow that animation.

### 1.2 Current restrictions

- The only [mesh animations](#) currently supported in LOTUS are those with a mesh animation matrix.
- The animation with the mesh animation matrix has to be a direct child of the base animation.

### 1.3 Concept

For a mesh animation with mesh animation matrix every vertex is assigned a weight. This weight describes how much to follow the matrix (animation) instead of following the base animation, like normal meshes do.

Let's make this a bit more concrete by looking at the gangway example. We have 2 cars, call them car A and car B. Car A is the parent animation of car B. The entire gangway is attached to car A. Now the vertices of the gangway are told how much they should follow the child animation (of car A) car B: the closer the vertex is to B, the higher the weight is. At the connection point the weight is 1. The closer to A the lower the weight and at the connection point to A the weight is 0. So 0 means do not follow the child animation and 1 means follow the child animation completely.

Using these weights, together with the configuration of which mesh animation matrix to use, LOTUS calculates the current shape of the gangway.

## 2 Preparation in the 3D program

The weight we just described are additional vertex data. Exporting these from Blender into the x3D format is not possible. Because of that we have to resort to a trick:

We create an exact copy of the 3d object we want to animate. This copy is named like the original object but with the prefix "@2m\_". If the original object is called "Gangway", the copy is called "@2m\_Gangway".

From now on this copy has to exist in EVERY export of the original object. After making changes to the original object, the copy must change to the exact new shape. Ideally you just make a new copy of the original object. Very very small discrepancies (positions, normals, etc.) are ok, but anything more will not be accepted.

What happens internally now is this: When importing the original object and the @2m\_ copy in the Content-Tool the texture coordinates of the @2m\_ copy are copied to the original object and saved as a second set of texture coordinates.

This second set can be used for special textures, like the Ambient Occlusion (AO) map. They however are also used for [mesh animations](#); these second texture coordinates are interpreted as weights.

To put it in another way: Is the texture coordinate assigned to a vertex in the @2m\_ mesh all the way on the left ( $x = 0$ ), then that vertex is assigned weight 0. Is the texture coordinate assigned to a vertex in the @2m\_ mesh all the way to the right ( $x = 1$  or 128, 256, ...), then that vertex is assigned weight 1. In this way the x values of the texture coordinates control how much each vertex can be deformed.

Important: The second texture coordinate (y) is at the moment irrelevant. We can imagine introducing [mesh animations](#) with 2 animation matrices; then the second texture coordinate will be used for that.

### 3 Configuration in the content tool

Important: After importing this object pair only 1 new object should show up; the @2m\_ object must not appear in the list!

We continue:

- We need to add Tangent information to the new mesh. For that on the right "Selected Mesh" and then "Add Tangent Info".
- Add the mesh to the base animation, to which the mesh is attached and from which the mesh animation should be calculated.
- The target animation (Mesh animation matrix) must either be created as a direct child or must already exist. Note down the name.
- If other objects also use the same texture / the same material, you need to create a copy of the material that is *only* used for the mesh that we want to animate. If you have several, distinct [mesh animations](#) every one of them must have its own unique material copy.
- The properties of the material should be set up as follows:
  - Material Type: "Complex+Normal+Simple Meshanim"
  - In the section Animations for "Animation matrix for [mesh animations](#)" select the child animation.

- Other properties can be changed as usual.

The animations already by tested in the Content-Tool if the animations themselves can be tested. In particular, because of this constraint, gangways cannot be tested, because they are assigned to a "Rail Vehicle" animation that cannot be tested in the Content-Tool.

Do note that a "no texture AO" generally and a "Texture AO with separate texture coordinates" are not possible. The extra texture coordinate set is already used for the weights. It is possible to use a "Texture AO" that uses the first set of texture coordinates.