Creating vehicles for LOTUS

Table Of Contents

1 Performance

This first section can only provide you with a high overview and some background information; we are not able to (nor want to) provide about texture resolutions, polygon counts etc. All these depend strongly on the priorities of the modeller and, of course, on how large the target audience should be or on how much to take into account less powerful computers.

1.1 Polygon Count, Texture Sizes and Level of Detail

Let's start with this! 🙂

There is never a reason to be wasteful with polygon count or texture resolutions. Taking this into account it makes no sense to provide you with simple/pointless specifications ('textures of at least 1024x1024' or 'Circles with at least 24 vertices'), you should think about what is acceptable. When choosing the respective detailing, always consider the following points:

Technical:

- Modern GPUs spend *considerably* more time on computing material properties than on the polygons.
- For the frame rate the size of the textures is not important. The renderer always works per screen pixel in the texture memory area. Whether this memory area (according to the resolution) is large or small, it does not matter. Access to a 3 TB hard drive is just as fast as a 1 TB hard drive - and not about three times as long!
- Meshes and textures must be in the graphics memory in order to be rendered! LOTUS allows the user to load texture with a lower resolution; therefore users with a less powerful graphics card are able to

load textures that were designed with a high resolution.
- Textures in the graphics memory have a way higher impact than meshes. A single 1024x1024 texture has over a million pixels, whereas a vertex is similar in size to a couple of pixels. In these orders of magnitude, we only start caring from around hundreds of thousands of vertices.
- Not to be forgotten, next to performance and graphic memory requirements, is loading times! Storage space on the hard disk however is simple irrelevant

Strategic:

- Always consider how long, how often and how close the user gets to the parts of your models! The driver's workplace and especially everything in the field of view looking forward should be built very detailed. The passenger area can and should be detailed sparingly.
- Always think about from which angle of view which polygons are needed! Vertical handrails are almost always seen from the side. Here it is way less noticeable when the handrail only has 8 corners as when a door button only has 8 corners!
- Other considerations are the contrast with the background and how much attention the object draws. The door buttons get way more attention than the fan heater under the seat.

1.2 Draw calls

A draw call is a single "draw task". The graphics card uses *one* effect, *one* texture and *one* transformation at a time and then draws the corresponding polygons. Then follows (using another texture, effect or transformation) the next draw call until the picture is built up completely.

Draw calls cannot be avoided if different components are animated differently or, for example, should light up under different conditions (like indicator lights). Limiting yourself in this area, while seldom possible, is not worth the trouble.

However, draw calls can be avoided by using the same texture for components that are animated in the same way with the same material properties. Assigning separate textures for the left and right side for the interior of your vehicles is not necessary and certainly not advantageous.

When importing 3D objects in the Content-Tool they are automatically split into draw calls. As a result every entry in the list on the right corresponds to a draw call, here referred to as a Mesh. Every mesh has exactly one texture/material and one animation.

There is however a case when the separation of textures is an advantage:

There is often a large number of different paints of a certain vehicle installed, loaded and rendered, however each repaint consists of a complete texture. Because of that only as little as possible should be located on the "repaintable" texture and its resolution should be considered carefully. For example there is no need for the entire interior to be included, as then for *every* repaint it will again (and unnecessarily) be loaded into memory. Of course unused parts of a texture can still be used for other things - they are there anyway. If there is still space in a corner for the passenger seat texture that would be nice. You will save space on another texture and it is also possible for the seats to be repainted as well. Also, for the AI-mesh, if one succeeds in placing the entire interior texture (but then reduced in resolution) on the exterior texture that

would be an advantage for the draw calls and storage needs for AI vehicles.

Finally we should mention that it is quite normal for a vehicle to have more than a few hundred draw calls. However, only a few of these should be used when the vehicle is not used by the user, which leads us to the next section:

1.3 Visibility conditions

LOTUS offers a very flexible method to select visibility conditions for single meshes (draw calls). You can indicate very precisely what meshes are visible from which perspectives.

In this way AI vehicles can have another (simpler) exterior mesh compare to a vehicle used by the user, and can hide switches etc. when viewing from the outside. On a two-way vehicle all elements from one driver's cabin can be hidden when the user is inside the other cabin.

2 How what works

Even at the start of construction for the 3D object it cannot hurt to know how things work. We will not go into too much details (for this there may be other articles here).

2.1 Objects and Meshes

In Blender a 3D model consists of some objects. Every object has its own coordinate system and consists of vertices, edges and faces. The vertices are texture coordinates, the faces may have different textures assigned to them.

When importing into the Content-Tool the objects are separated according to the used texture. Each unit is referred to as "Mesh" and used as a future "administrative unit".

A mesh is characterized by:

- the vertices/faces that make it up
- the assigned material
- the assigned animation, if any
- Events for clicking with the left or right mouse button
- other properties.

2.2 Materials

A material is a collection of properties for a surface. These properties are:

- main texture
- shine
- night time texture and the variable that controls it
- transparencies or other transitions/mix-ins with the background
- further textures, like the normal ("Bump") texture
- other settings, like the blend characteristic of the night texture

Every mesh must be assigned exactly one material, however materials can be used multiple times. Do consider that every small difference in the material properties requires a separate material. The indicator light "Indicator right" should be a different material from the indicator light "indicator left" because a different variable controls the Night Texture.

2.3 Animations

At their core animations are "separate", you can create a complex hierarchy of animations without even importing a single mesh! The properties of an animation include the following:

- Animation type, rotation, translation or a special bogie or wagon body animation.
- Control variable
- Translation function (if variable = A then rotation = B)
- Limits and delays

After creating an animation, any number of meshes can be coupled with it.

When adding the first mesh to an animation and if the animation does not yet contain information about the rotation and translation axes or pivot point, this information can be copied from the local coordinate system of the original object. It can pay of the spend the time and effort in the 3D program to adjust the pivot point and the axes of the local coordinate system to the future animation; this will save you manual input of these values later.

You can change a setting in the Content-Tool about which axis (X, Y or Z) should be used after import as rotation or translation axis. If a mesh has already been imported this setting has no further effect. Only after a new import the changed axis is used.

If the Z-axis should be the "favorite" axis then set the Z-axis in the import tool and build object (like rotary switches) that should be animated such that their local Z-axis corresponds to the axis of rotation. For information on how to do this, look in the documentation of your 3D program. When you then import the object in the Content-Tool the Z axis is preselect for the animation.

If you totally messed up the axes then in case of emergency you can also edit them yourself. This is not at all a problem if the axis is transverse, longitudinal or vertical, when it comes to oblique axes this is way harder.

2.4 How do I build..

In order not to have to laboriously research how to model this or that component, here are some examples and their typical realization.

2.4.1 Shiny and Matt Surfaces

If an entire object should shine evenly or appear mat, it is sufficient to change the parameter in the material properties. It is also possible to create different levels of glossiness on different parts of an object - with the help of the alpha texture. At locations where the alpha channel is white the parameters in the material properties are used; where the alpha channel is black the material is always completely mat.

2.4.2 Transparent Surfaces

Transparency can be, similar to glossiness, the same on the entire object (continuous) or by using the alpha channel to allow for different results on different parts. You can use a "Black / White" alpha channel to provide hard cuts or use shades in between to get a "soft" blending. As there only is one alpha channel transparency and glossiness cannot be controlled simultaneously through the alpha channel.

2.4.3 Switches

Switches are created as separate objects. Remember to set the local coordinate system for the animation you want. In the Content-Tool you can then assign the meshes that correspond to the object to an animation and add the needed click events.

2.4.4 Wagon body

In the simplest case (a single car with two axles or two bogies) no animation is assigned to the wagon body. Articulated vehicles however are not divided into individual vehicles but all car bodies are imported into a single vehicle. Accordingly, depending on the configuration, car body animations must be created. The mesh of the car body and everything that belongs to that car (sounds, lights, particle sources etc.) are assigned to the animation. Animations of moving parts within the car body must also be placed under this car body animation.

2.4.5 Bogies and Axles

Bogies and axles are created at separate objects and can then be added to the relevant animations.

2.4.6 Indicator lights

To allow indicator lights to light up, a fitting night texture must be applied. Furthermore, indicator lights must be placed in a separate object (you can only combine indicator lights that always light up together). In the Content-Tool you then need to create copies of the material, each with the correct night texture and control variable.

3 A Brief Look on Imports

Importing the raw files of the vehicle is not the subject of this article. Therefore, we just want to give you a rough idea of how the import works - in a sense, to be prepared!

3.1 Model

The 3D model can be imported either in one go and/or sequentially and in "parts". In particular during the rework phase this can simplify (and speed up) the imports as only the objects that have changed are imported.

In this case, the meshes generated from the imported objects are either added or replaced, depending on whether there is already a mesh with the respective name and the respective texture. However, this also means that renaming an object, which has already been imported once, will result in a duplicate object. But this is essentially not a problem because meshes can be deleted at any time. It does show however that clear names should already be given during the development phase.