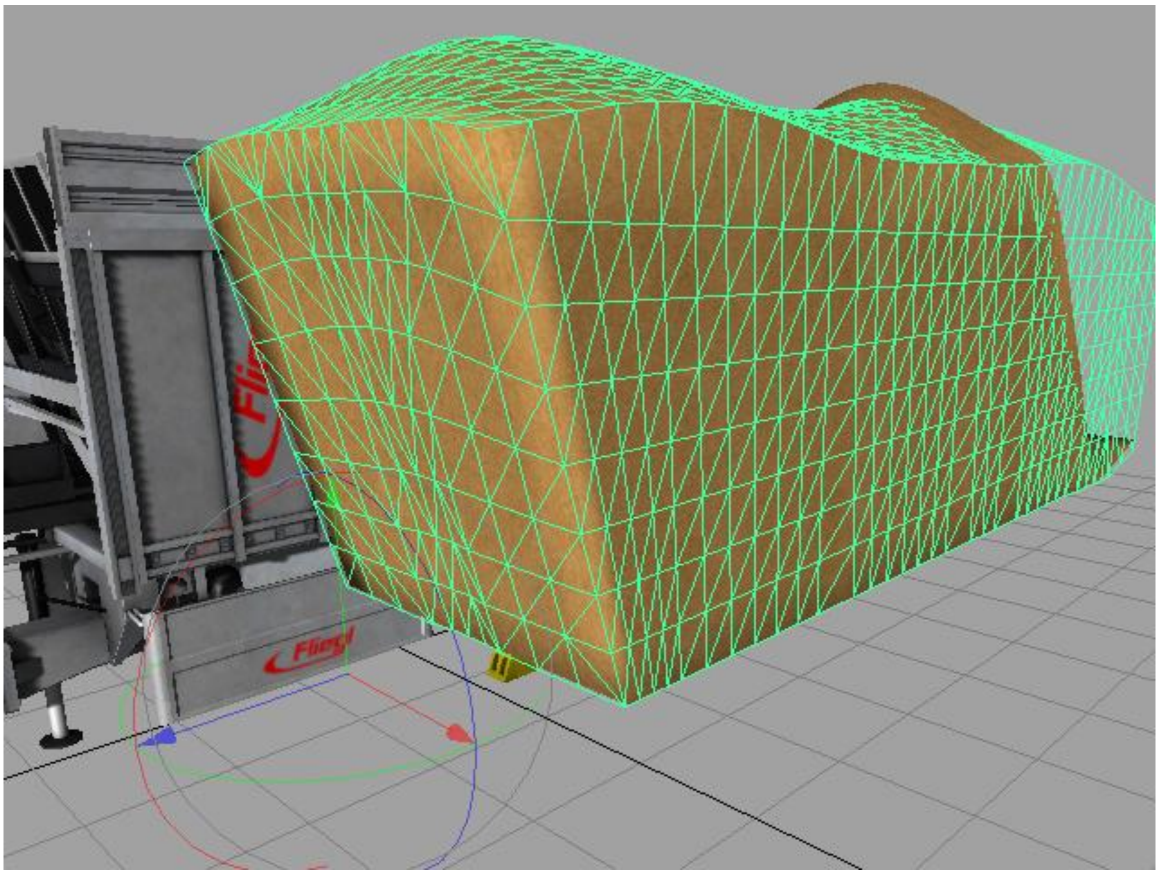
Explanations for ShaderTipping

- Position of Pivot
  - from side view: bottom = lowest point of plane
  - from top view:
    in the center for normal trailers
    at the front for "push off trailers" (in German: Abschiebewagen)
  - example:



- Shader Parameters

  <CustomParameter name="p1" value="1.25 0 6.97 1"/>

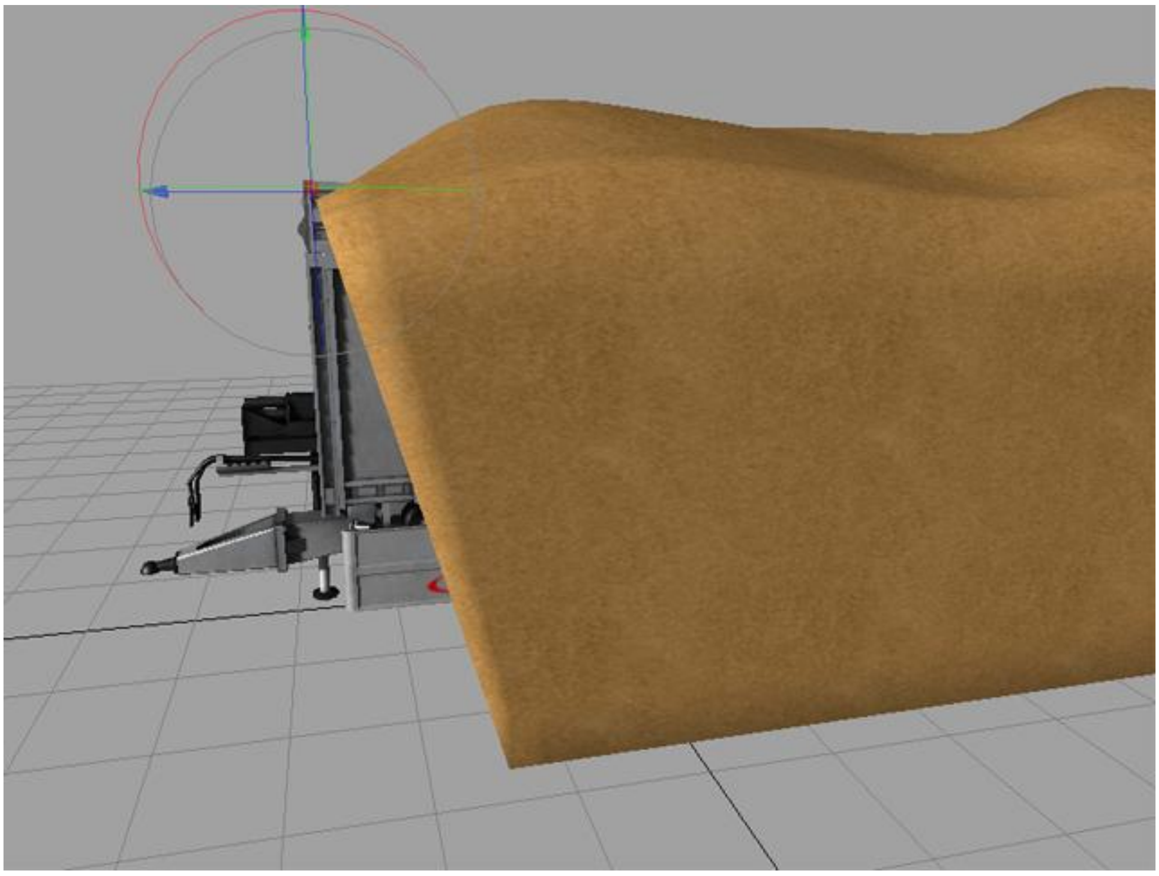
  This parameter set defines the dimensions of the plane.
  * First value is the width of the plane. (half of the actual plane width = length from pivot to a side)
  * Second value defines if the plane is for a normal or a push off trailer. (0 = push off, 1 = normal)
  * Third parameter is the length of the plane.
  For normal trailers this is half of the length (length from pivot to end of plane)
  For 'push off trailer' this is the total length.
  * Fourth parameter should be one, but is set by the script for 'push off' trailers to limit the length of the plane during the unload process

```
<CustomParameter name="p2" value="1 1 1 1"/>
```
and
```
<CustomParameter name="p3" value="1 1 1 1"/>
```

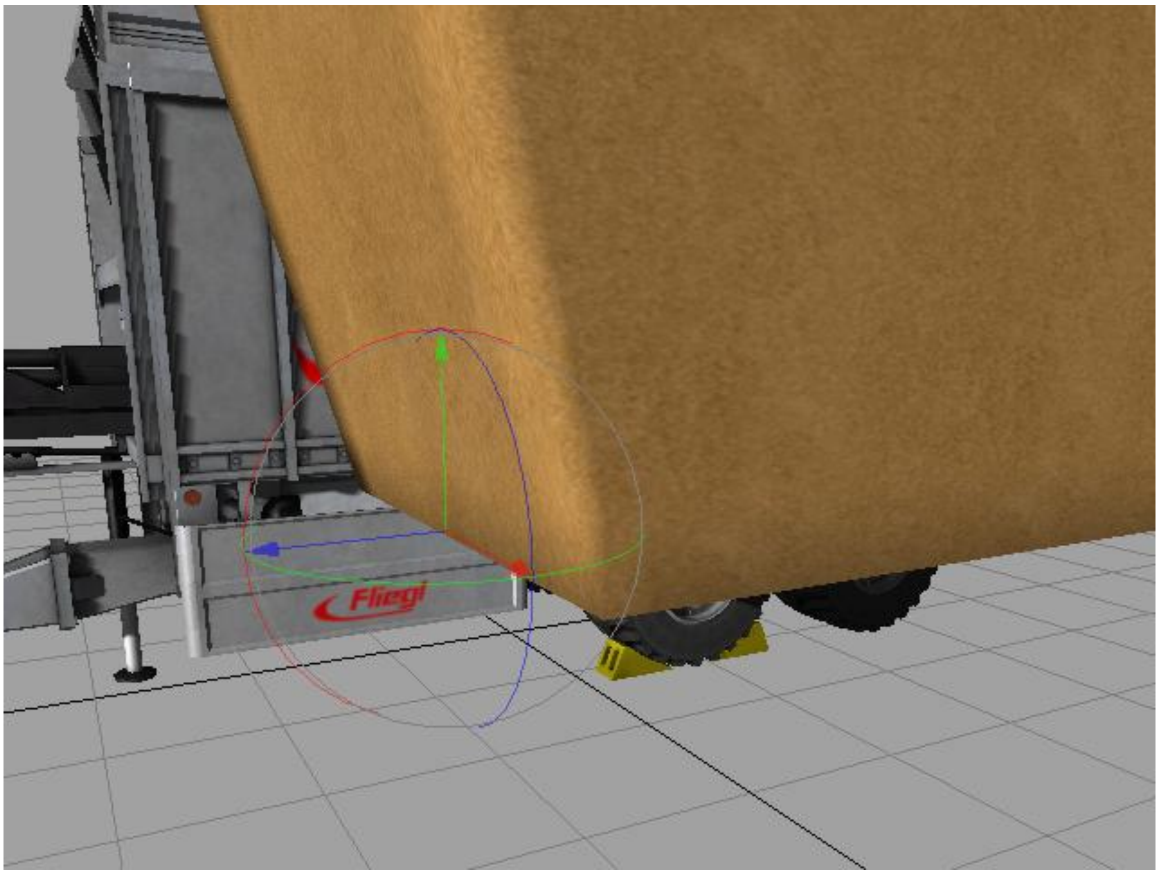
These two parameter sets are used to set the fill level of the plane.

NOTE:
You can set them (and all other parameters) in GE by using the following command:
```
setShaderParameter(nodeId, "parameter", 0.0, 0.0, 0.0, 0.0, false);
```
example:
```
setShaderParameter(251, "p2", 1.0, 0.8, 0.4, 0.1, false);
```
This would set the shader parameter p2 of object 251 (visible in the attributes panel) to the named values.

```
<CustomParameter name="yMax" value="2.25 0 0 0"/>
```
Basically this parameter set contains only one value, but the GE saves CustomParameters as 4 dimensional vector so that's why there are more than one value.
It is the height of the vertex at the front (headside?), like the following image shows:

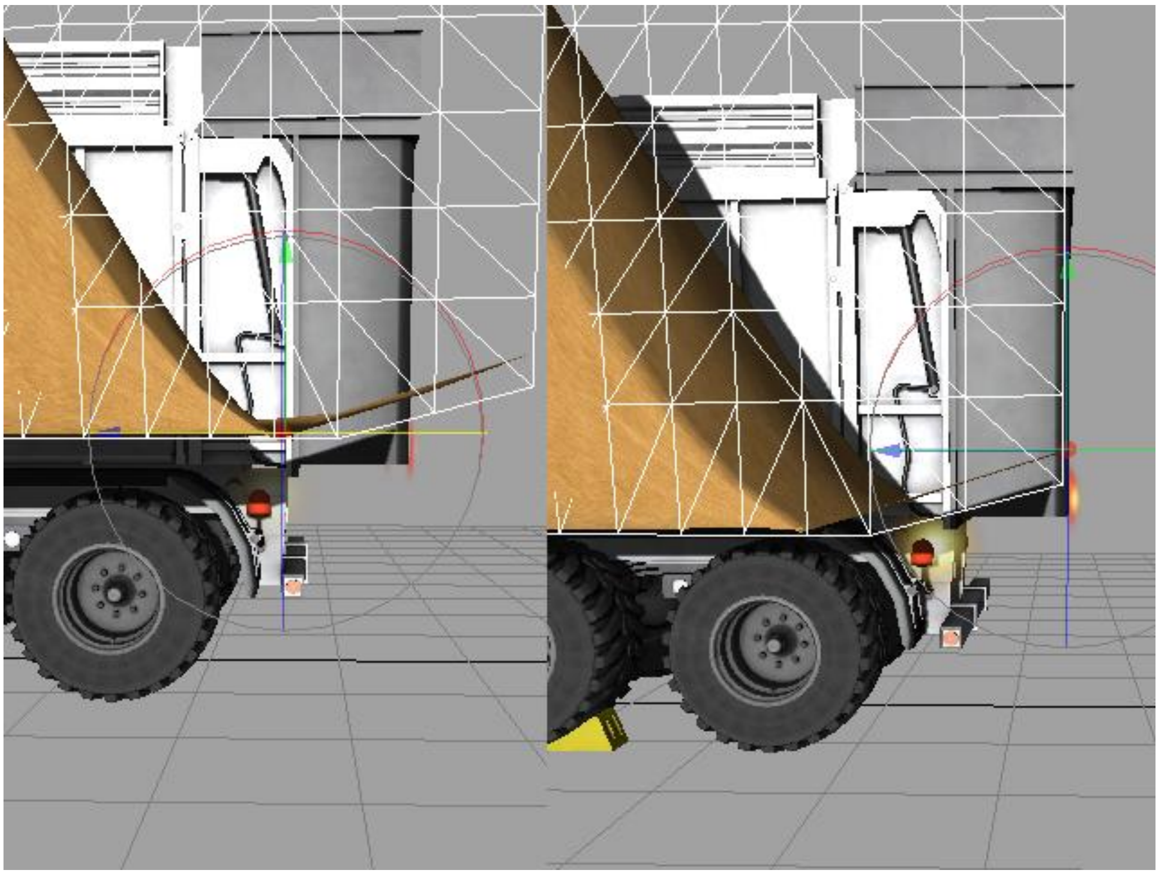


<CustomParameter name="lf" value="0 -0.65 2.3 0"/>
This parameter set defines how the plane should be deformed at the front (headside?).
* First value is useless.
* Second value defines the distance between the pivot and the headside of the plane at the bottom in z direction.



* Third value is the height, similar to yMax, but can be used to tweak the behaviour of the plane.
* Fourth value is some kind of offset, but probably useless in most cases.

<CustomParameter name="lb" value="0 -6.2 0.22 -6.92"/>
This parameter set defines how the plane is deformed at its end.
* First value is some kind of offset and useless in most cases
* Second value is the z position where the plane starts to raise (left half of image)



* Third value is the amount which the plane should raise
    (difference between position of 'vertices'/transformgroups in left and right half images)
* Fourth value is the z position of the rear end, where the plane raises normally again

NOTE: Values have been adjusted ... they do not fit the actual geometry perfectly.

<CustomParameter name="ls" value="0 1.01 0.5 1.175"/>
This parameter set defines how the plane should be deformed at its sides.
It is valid for both sides, so you might only use symmetrical planes - are there any other forms?
Values and the following picture are taken from a other trailer.
* First value is some kind of offset and is useless in most cases
* Second value if the width of the plane at the lowest point
* Third value is the height of the plane where it should have full width
* Fourth value = full width

Finally, let me say that this project (Shader Tipping) was initiated nearly one year ago, slept on my hard drive for nearly have a year and therefore I'm not 100% sure what does what and why...

But this short introduction might give you an idea .. at least I hope so.

If you are unsure what the values do, here's a tip:
* create a transformgroup and place it as a child into the fill plane
* now set the position of the transformgroup according to the values of the shader parameter
Yeah, might be hard to figure out what value corresponds to which position value of the transformgroup, but there are not that many possibilities and you got this introduction.

And yes ... the shader itself is quit disgusting ...
This is due to the fact that I got the idea to keep the amount of variables as small as possible.
So the shader got prolonged because of the if/else statements, which could have been put into a much 'smarter'/nicer form - no question.

Happy modding and keep up the fucking credits!

This mod (Shader Tipping) and this 'tutorial' is free of charge and without any warranty ;)

If you destroy your computer it's your fault :-D

fruktor