

BeleuchtungV31

Autor: Sven777b

Version: 3.1 (16.11.11)

The specialization can be freely used without asking for permission.
This documentation should be delivered with the mod or there should be a note pointing to it.

Modifications on the script are not needed and not allowed.
For wishes of modifications or bugreporting contact me. ICQ#95176001
The script is protected by copyright!
It is not allowed to copy or modify any parts of this script.

Content:

1. Overview
2. Modifications to moddesc.xml
3. Modifications to vehicle-xml
4. additional informations

Overview

- highbeams (press and hold the lights-key for some seconds to activate the highbeam)
- brakelights
- reverselights
- direction lights left/right (optional with automatic switch-off)
- optional ticksound for the turnlights
- hazardlights
- parkinglights left/right (leave the direction lights on when leaving the vehicle)
- parkinglight for the vehicle (if you set the option lightsStayOn true)
- daytime-running-lights
- stroboscope flashers (optional defined or random sequence)
- seperate switchable workinglights
- turn-, reverse- and directionlights are synchronized with attacher vehicle
- speed of turnlight-sequence is synchronized with attacher vehicle
- automatic beacon dependend on graintank fill-level
- light for combine dumping-pipe (activated on extending the pipe)
- extended beacon lights to support rotating parts

All lights will also support

- stay activated on leave (for standard beacons too)
- replacing real lightsources by virtual lightbeams in multiplayer
- unlimited amount of lights of the same type (like 2 separate brakelights)
- switch/toggle sound
- full multiplayer-ready
- SRS ready (makes also vehicles selectable)

Notes for reading this documentation:

Gray colored parts of XML-Codes may already exist in your file. In this case the other parts should be added to your existing code.

Blue colored parts of XML-Codes will be further explained.

Modifications to the moddesc.xml

First you need to import the specialization:

```
<specializations>
  <specialization name="additionalLights" className="BEL3" filename="beleuchtungV31.lua"/>
</specializations>
```

Then you should – unless it's already done – create a new vehicle type.

Caution! Don't use the names of original vehicle types such as "tractor" or "combine".

The following example describes a combine but you can use this lights-script on all vehicle types.

The needed specializations for other vehicle-types you can find at the Wiki:

http://wiki.landwirtschafts-simulator.de/index.php/LS11_vehicleTypes_der_Originalfahrzeuge_und_Tools

```
<vehicleTypes>
  <type name="combine_cylindere_B31" className="Vehicle" filename="$dataS/scripts/vehicles/Vehicle.lua">
    <specialization name="motorized" />
    <specialization name="steerable" />
    <specialization name="cylindere" />
    <specialization name="combine" />
    <specialization name="hirable" />
    <specialization name="aiCombine" />
    <specialization name="honk" />
    <specialization name="additionalLights" />
  </type>
</vehicleTypes>
```

As you can see the lights-script should be the last in the list of specialization. The chosen name in the <vehicleTypes> part must be the same as in the <specializations> part.

Depending on which types of lights you will use it could be necessary to add additional informations for inputBindings and language-translations (l10n)

```
<inputBindings>
  <input name="worklights" key1="KEY_KP_5" button="" />
  <input name="BEL3LEFT" key1="KEY_KP_1" button="" />
  <input name="BEL3WARN" key1="KEY_KP_2" button="" />
  <input name="BEL3RIGHT" key1="KEY_KP_3" button="" />
</inputBindings>
```

The following 3 entrys are mandatory if you use turnlights :

BEL3LEFT : InputBinding for left turnlight

BEL3RIGHT : InputBinding for right turnlight

BEL3WARN : InputBinding for hazard lights

Additional inputBindings for worklights can be added. Remember the name of the inputBinding needs to be the same as in the parameter inputName= of the specific worklight.

Caution! If multiple mods are using the same names for their inputBindings then all those mods will use the same keys - even if you defined other keys in the moddesc.xml

```
<l10n>
  <text name="BEL3LEFT">
    <de>Blinker links</de>
    <en>left turnlights</en>
  </text>
  <text name="BEL3RIGHT">
    <de>Blinker rechts</de>
    <en>right turnlights</en>
  </text>
  <text name="BEL3WARN">
    <de>Warnblinker</de>
    <en>hazard lights</en>
  </text>
</l10n>
```

For every input binding you should also add a translation text. This will help to correctly show the inputBinding in the general game-settings.

Modifications to the .xml of the vehicle

General structure:

```
<lightsaddon lightsStayOn="true">
  <light type="highbeam" index="32|0" real="32|0|2" beam="32|0|1" />
  <light type="dirLeft" index="32|2" />
  <light type="dirRight" index="32|3" />
  <light type="reverse" index="32|7" />
  <light type="brake" index="32|8" />
  <light type="pipelight" index="0>7|2" />
  <fillLevelWarning percent="80" />
  <flashers soundfile="blinktick.wav" lowPitch="0.8" highPitch="1.0" volume="0.6"/>
</lightsaddon>
```

Parameter in the main branch <lightsaddon> :

- **lightsStayOn="[true|false]"**
this parameter is optional. The default setting is „false“

If set to true the lights will stay on when leaving the vehicle. If the lights of the vehicle are configured correctly then the real lightsources and lightbeams will shut down and only the coronas will stay visible.

- **blinkSpeed="0.5"**
this parameter is optional. The default setting is „0.5“

Using this parameter you can change the blinkingspeed of the turnlights. The value should be given in seconds for one phase.

- **modDirection="[1|-1]"**
this parameter is optional. The default setting is „1“

If your mod was built in the wrong direction it could happen that brake and/or reverse lights will show up in the wrong situations. In this case set the value of this parameter to "-1".

Parameters of the single <light> entries:

- **type="[...]"**
this parameter is mandatory. There is no default setting.

possible values are :

- o highbeam : highbeam / full beam
- o dirLeft : left turnlight
- o dirRight : right turnlight
- o parkLeft : left parking light
- o parkRight : right parking light
- o brake : brake lights
- o reverse : reverse lights
- o work : working lights
- o strobe : strobe-flashers
- o pipe : pipe light
- o drl : daylight running-lights

You can use as many lights of same type as you want. Some of the following parameters are not available on every type of light.

- index="[...]"**
 this parameter is mandatory. There is no default setting. Can be used on every type.

 This is the index number of the main light component. The index should point to the lit surfaces and coronas. It will be the main object which is set to visible/invisible.
- real="[...]"**
 this parameter is optional. No default setting. Used at: work, highbeam, pipe

 this index should point at the real lightsource. In multiplayer this light will be invisible to other players. Also this light is shut down on leave even if "stayon" is set to true.
- beam="[...]"**
 this parameter is optional. No default setting. Used at: work, highbeam, pipe

 This index should point at the virtual light beams which are also used by the AI. In multiplayer this beam will be visible to the other players instead of the real lightsource.
- stayOn="[true|false]"**
 this parameter is optional. Default setting is „false“. Can be used at: work, strobe, pipe

 If this parameter is set to "true" then the light will stay on when leaving the vehicle. Real light and beam will be set to invisible – only corona (index) will stay visible.
- inputName="[...]"**
 this parameter is optional. There is no default setting. Can be used at: work, strobe

 Here you should write down the name of the inputBinding for this light. If you don't set this value at workinglights and strobes then these lights can not be switched on/off.
- helptext="[...]"**
 this parameter is optional. There is no default setting. Can be used at: work, strobe

 For switchable lights you can define the name of the helptext (l10n) which should be displayed ingame. The easiest it would be to write down the same name as you used for inputName. If you leave this parameter then no helptext will be shown.
- sequence="[..]"** (only type="strobe")
 this parameter is optional - it decides if the following parameters are used.

 you can set a defined flashing-sequence by putting in the times (in milliseconds) for each phase seperated by a blank. At the end of the given sequence it will start from beginning. If you omit this parameter then the flasher will switch to random mode.
- invert="[true|false]"** (only type="strobe")
 this parameter is optional. Default value is "false". only if "sequence" is set

 If you set this parameter to "true" then the flashing-sequence starts with a off-phase. This is primary used to create a offset for multiple flashers.
- minOn="[...]" & maxOn="[...]"** (only type="strobe")
 this parameters are optional. Default for both is "100". only in random mode

 Defines the minimum and maximum possible duration (in ms) of the on-phase. Used for the randomly generated flashing-sequence.
- minOff="[...]" & maxOff="[...]"** (only type="strobe")
 this parameters are optional. Default values are "100" & "400". only in random mode

 Defines the minimum and maximum possible duration (in ms) of the off-phase. Used for the randomly generated flashing-sequence.

Additional entries for the main branch:

- **<fillLevelWarning percent=“[...]“ />**

this entry is optional, the parameter "percent" is mandatory.

If you insert this entry then you can define on how many percent of graintank-filllevel the beacon lights should be activated. This feature only works at combines. If the actual fill-level reaches the marker then the beacon lights will turn on. You can manually turn it off if you want to. If the fill-level goes below the marker then the beacon lights will turn off unless you activated them manually. (works with ai-helper)

- **<flashers soundfile=“...” lowPitch="0.8" highPitch="1.0" volume="0.6" autoreturn="false"/>**

the entry and all parameters are optional.

this entry allows you to define additional parameters for the turnlights :

o **soundfile=“...”**

if you define a soundfile then it's used as tick-sound for the turnlights. The sound is played on every phase-change of the turnlight so it's duration should be shorter than the blinkspeed=“...” value reads. if no soundfile is given then the parameters lowPitch, highPitch and volume will be ignored.

o **lowPitch=“[...]“**

defines the sound pitch while switching in the off-phase. Default value is "0.8"

o **highPitch=“[...]“**

defines the sound pitch while switching in the on-phase. Default value is "1.0"

o **volume=“[...]“**

defines then volume of the ticksound. Default value is "1.0"

o **autoreturn=“[true/false]“**

If this parameter is set to "true" then the turnlights will automatic turn off at the end of a steering process.

Extensions for default vehicle entries:

- **<beaconLights stayOn="true">**

With this parameter the beacon lights will stay activated if you leave the vehicle.

- **<beaconLight index=“...” rotIndex=“...” speed=“...” />**

Use this optional parameter to define a seperated rotation part on standard beacon lights. In this case the object defined by "rotIndex" will be rotated while the object defined by "index" is only set visible.

Hints for the structure of the model

Worklights and the Farlights can be more complex in FS11. This means additional to the normal lit surfaces and coronas you can also define the real lightsource and a virtual lightbeam which is shown to the other players in multiplayer-mode. It's useful to cascade the different types and set all to visible. You can use the sampleMod "AgroX720B3" as reference.

```
[ - ] Licht (TransformGroup)
    [ - ] Lichtquelle (light)
    [ - ] Coronas (TransformGroup)
        Corona (Shape – Leuchtfläche)
    [ - ] Beams (TransformGroup)
        Beam (Shape – Lichtstrahl)
```

Attached helping files:

To support people without much experience in modding I attached some I3D files to help them adding lights into their mod. / textures which are needed by the i3d objects are written in braces)

- rundum.i3d (rundum.dds , corona_orange.dds , emissiveBillboardShader.xml)
contains the complete model of a beacon with separated rotation-part for use with this script.
- squarebeam.i3d (car_beam_diffuse.dds)
contains the model of a single light-beam for working lights or highbeam.
- flasher.i3d (flasher.dds)
contains the simple model of a strobe-flasher for emergency-cars or something like that
- beleuchtung.i3d (car_beam_diffuse.dds , corona_orange.dds , car_corona_red_diffuse.dds , car_corona_white_diffuse.dds , emissiveBillboardShader.xml)
contains the complete light-set of the early V3.0 samplemod "AgroX720B3" for an easy import.

Quick import tutorial

To install the lights quick and easy into your mod copy following files into the folder of your tractor:

beleuchtung.i3d
car_beam_diffuse.dds
car_corona_red_diffuse.dds
car_corona_white_diffuse.dds
corona_orange.dds
emissiveBillboardShader.xml

Open your tractor in Giants Editor and import "beleuchtung.i3d" by choosing File->Import from the main menu. Then click at the imported transformgroup "beleuchtung" and choose Edit->Cut from the menu.

Now click at the first object in the scenegraph and choose Edit->Paste.

Finally you have to count the index to the transformgroup "beleuchtung" – or if you already use GE 4.1.8 or higher you can read the index in the attribute line "index path".

The needed XML entries you can copy from the former samplemod of beleuchtungV3 – AgroX720B3 which can still be downloaded from this location: <http://ul.to/u30kuz> or any other download-portal.

