

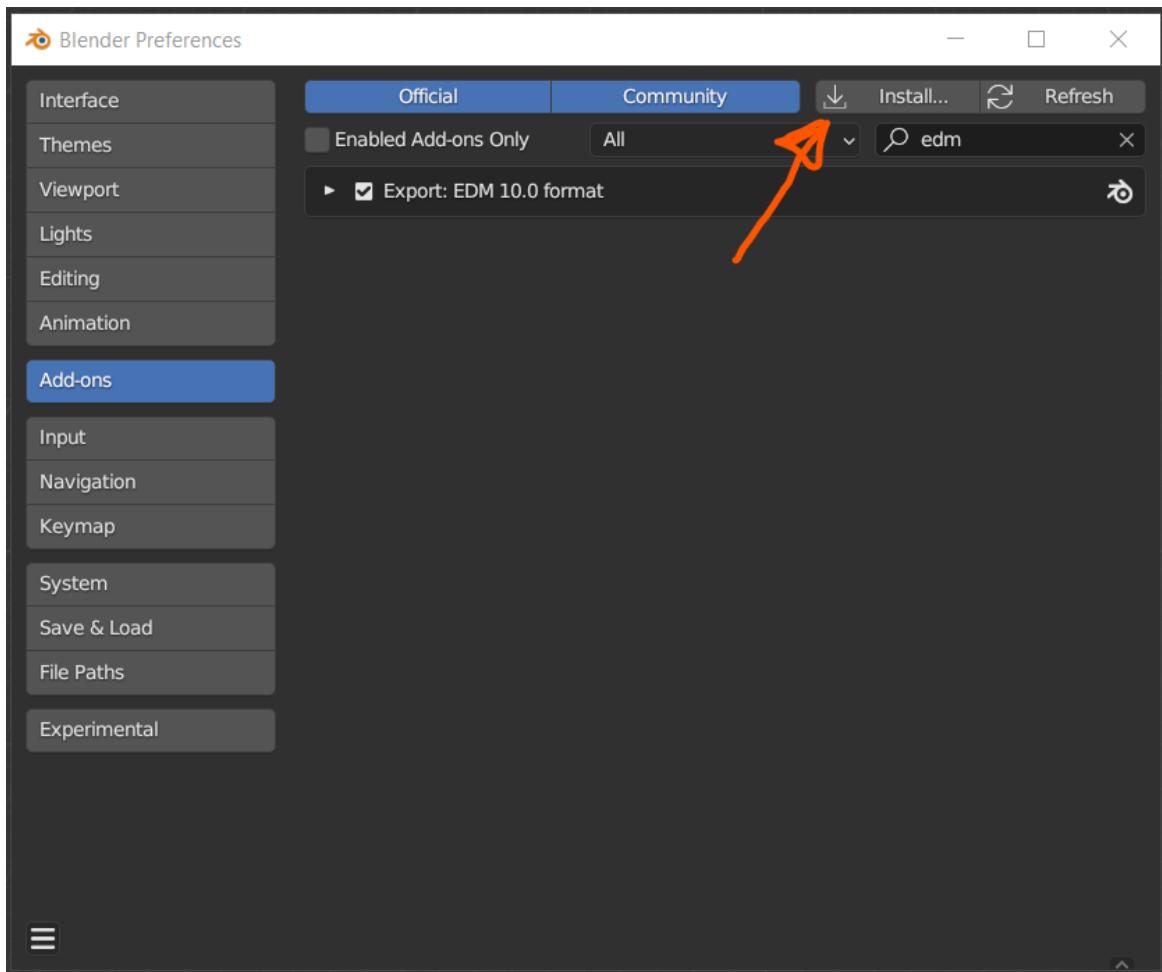
# Export Guide for Blender (ENG)

## Plugin Installation

Download from: [https://files.eagle.ru/mods/edm\\_blender\\_plugins/](https://files.eagle.ru/mods/edm_blender_plugins/)

Install it through the **Preferences window** like any other add-on.

\*Make sure the checkbox is checked.

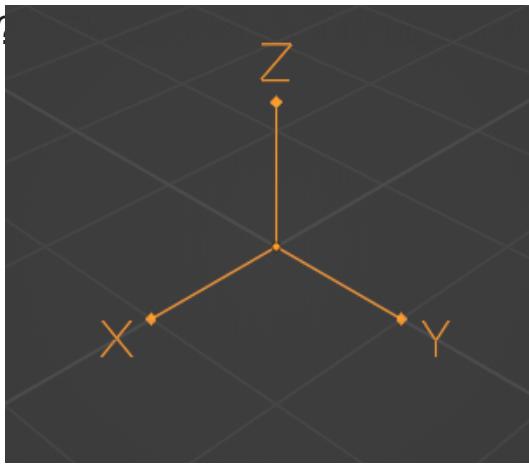


\*\*\*Demo scenes for learning are located in the add-on folder - examples for training.

C:\Users\\*\*\*User\AppData\Roaming\Blender  
Foundation\Blender\3.6\scripts\addons\io\_scene\_edm\Learning\_Demo

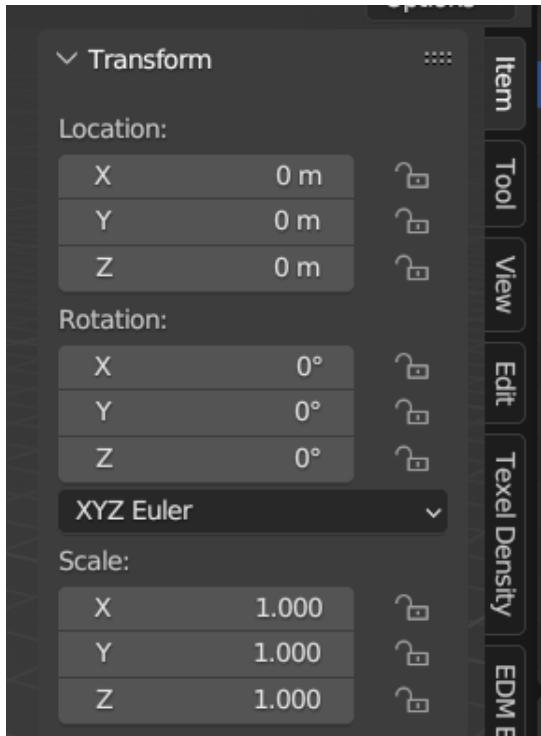
## General Scene Requirements

? File names should use Latin alphabet letters. Spaces should be replaced with underscores «\_».

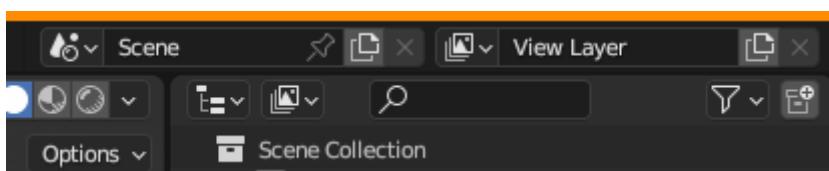


aligned with the nose along the X-axis.

- ? Objects must have their scale and rotation "reset."



- ? Use default scene and view layer settings.

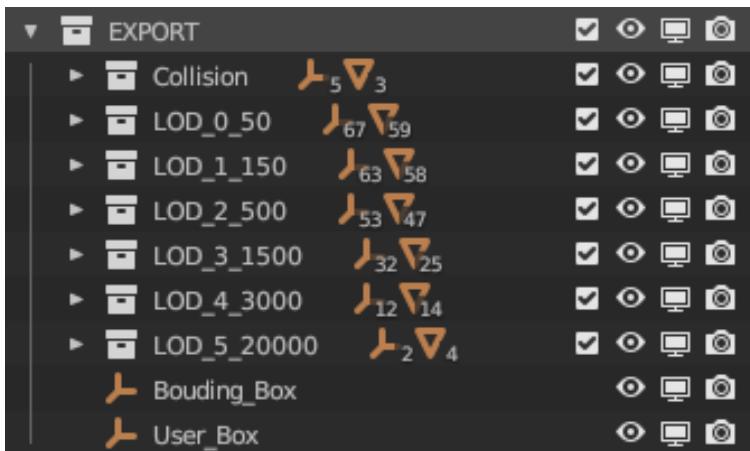


- ? Export is carried out in Eevee mode.
- ? It is recommended to enable the Wrangler Addon. (installed by default)

## Layers

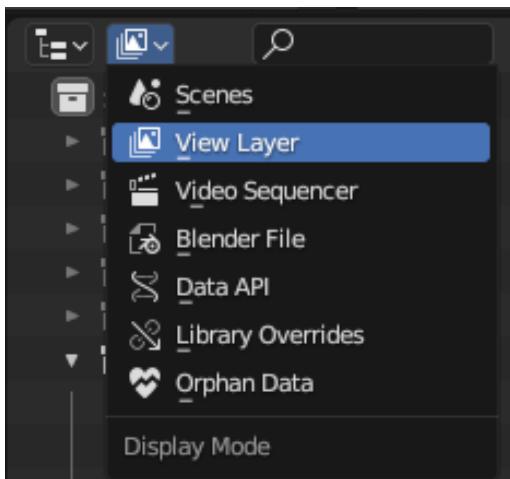
All scene objects should be organized into layers (scene collections):

? The layers window is called the Outliner.

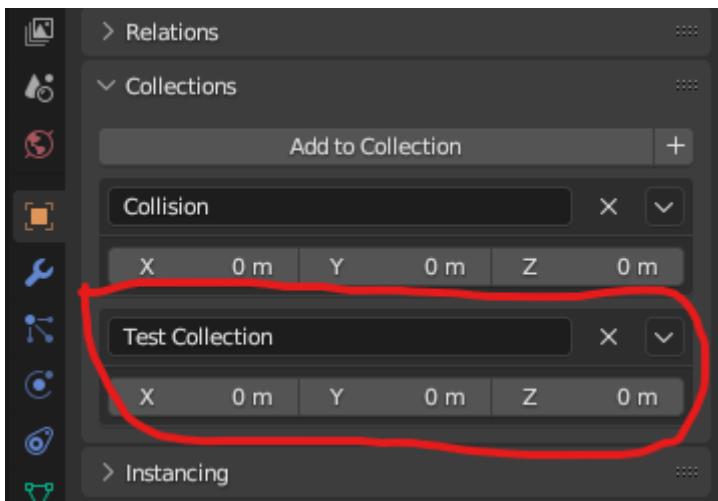


? In Blender, there are two types of collections:

1) Scene collections, shown in the Outliner under View Layer (referred to simply as scene layers). These are considered during export only.



2) Object collections, found in the Object Properties tab. These **are not** considered during export.



? Only layers with an enabled checkbox are exported.



? When you create a root layer for export in the scene (name it, for example, Export). Inside it, create layers for the model's LODs and a Collision layer for collisions. Name them according to the LOD (as indicated in the example).

EXPORT		
Collision	5 ▽ 3	<input checked="" type="checkbox"/> <input type="radio"/> <input type="checkbox"/> <input type="checkbox"/>
LOD_0_50	67 ▽ 59	<input checked="" type="checkbox"/> <input type="radio"/> <input type="checkbox"/> <input type="checkbox"/>
LOD_1_150	63 ▽ 58	<input checked="" type="checkbox"/> <input type="radio"/> <input type="checkbox"/> <input type="checkbox"/>
LOD_2_500	53 ▽ 47	<input checked="" type="checkbox"/> <input type="radio"/> <input type="checkbox"/> <input type="checkbox"/>
LOD_3_1500	32 ▽ 25	<input checked="" type="checkbox"/> <input type="radio"/> <input type="checkbox"/> <input type="checkbox"/>
LOD_4_3000	12 ▽ 14	<input checked="" type="checkbox"/> <input type="radio"/> <input type="checkbox"/> <input type="checkbox"/>
LOD_5_20000	2 ▽ 4	<input checked="" type="checkbox"/> <input type="radio"/> <input type="checkbox"/> <input type="checkbox"/>
Bounding_Box		<input type="radio"/> <input type="checkbox"/> <input type="checkbox"/>
User_Box		<input type="radio"/> <input type="checkbox"/> <input type="checkbox"/>

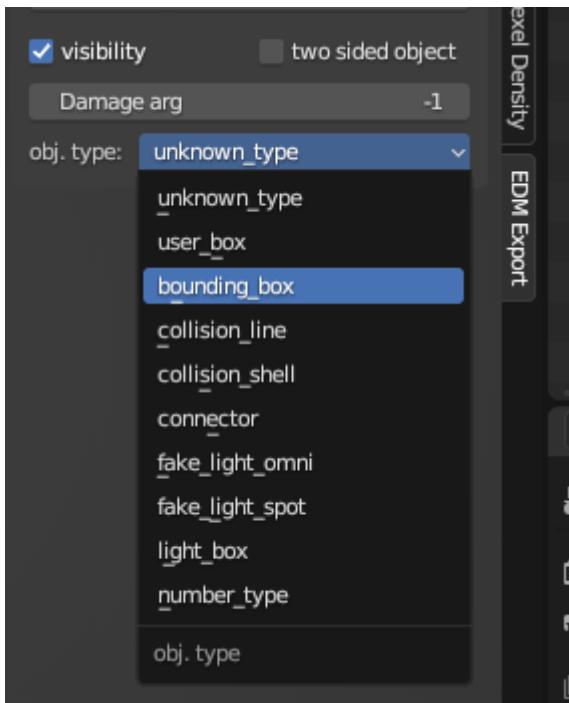
? Distribute the model's LODs across these layers. For LOD layers, specify the display distance in meters using an underscore (e.g., LOD\_0\_50).

? Helpers (Dummies) used for animation should be unique for each layer.

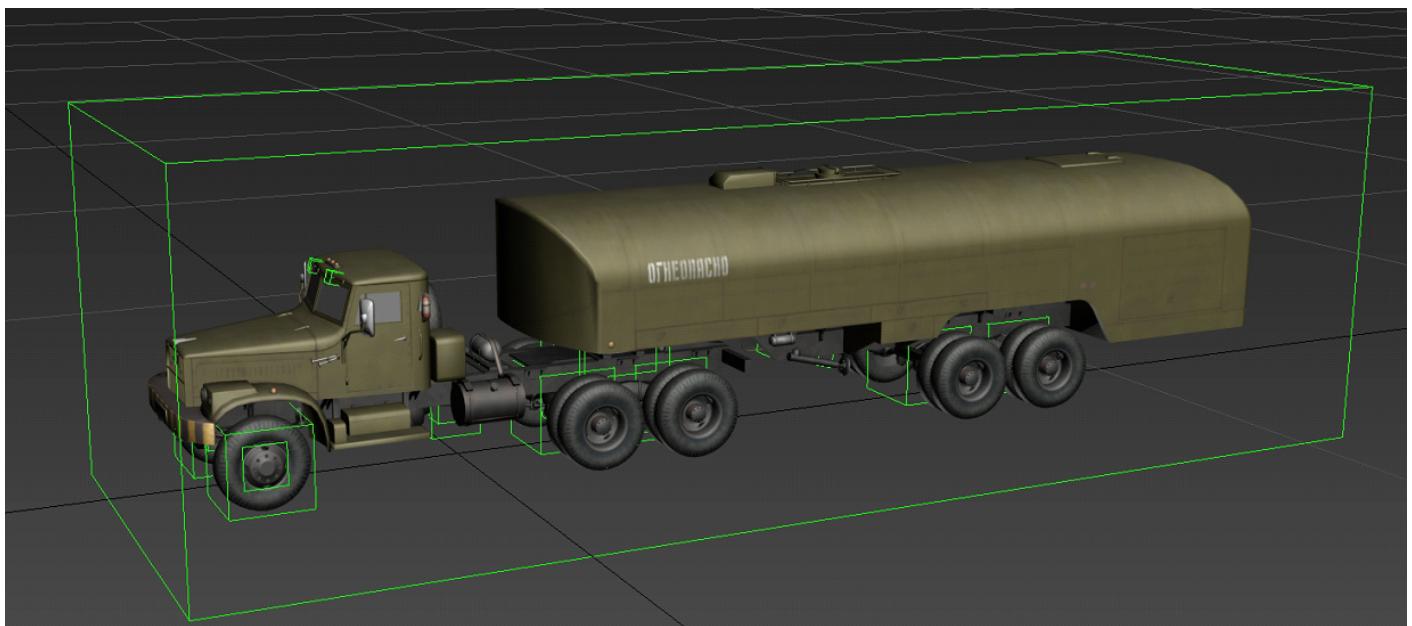
## **? Avoid confusion in the hierarchy!**

### **Auxiliary Objects: Bounding and User Box**

? Create a new Empty Cube in the root layer “Export” and name it “Bounding\_Box”. In the EDM Export menu (N-panel), specify the object type as Bounding Box.



? Adjust the size and position of the Bounding\_Box to the model so that it covers its volume with a margin, including during animation (for example, raising a cannon).



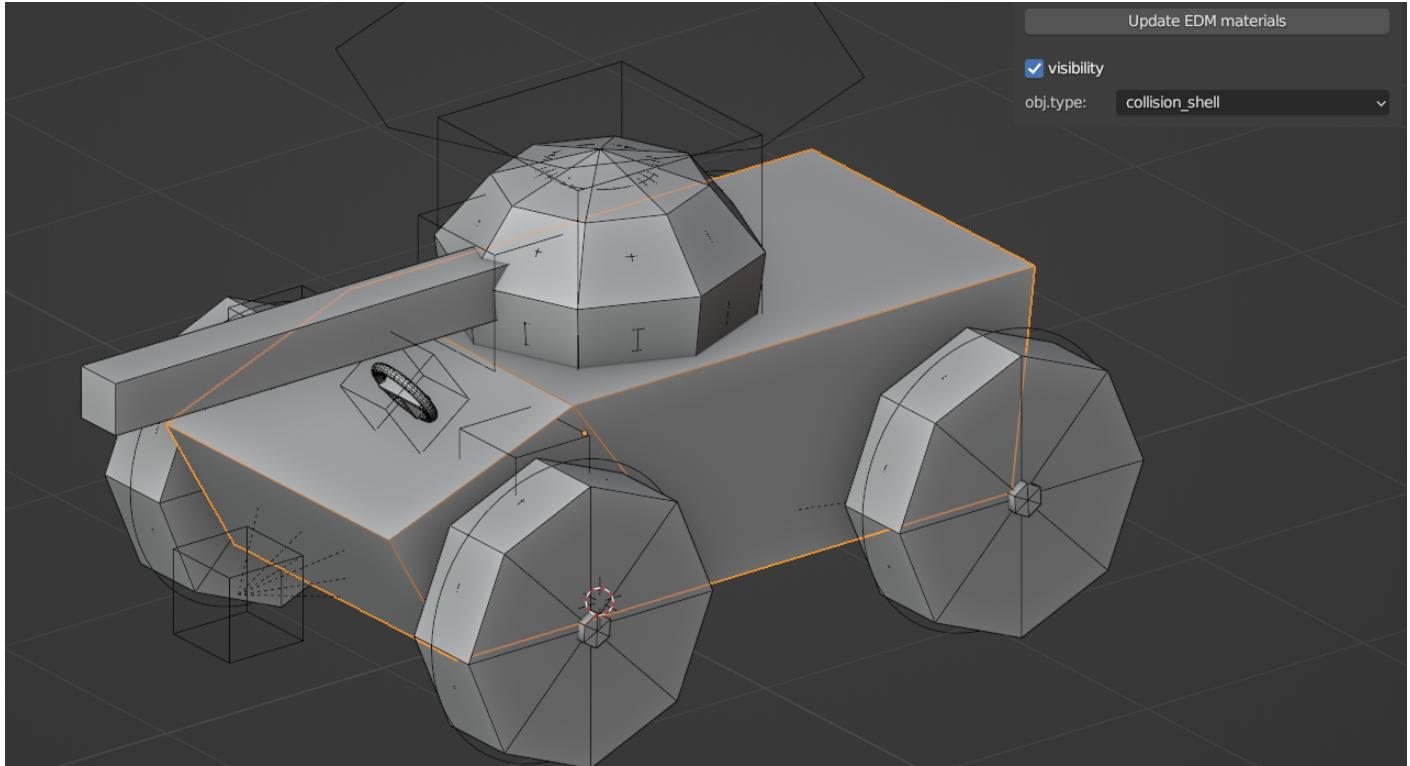
? Do the same for the User Box, which should be smaller and simply cover the model.

## Collisions

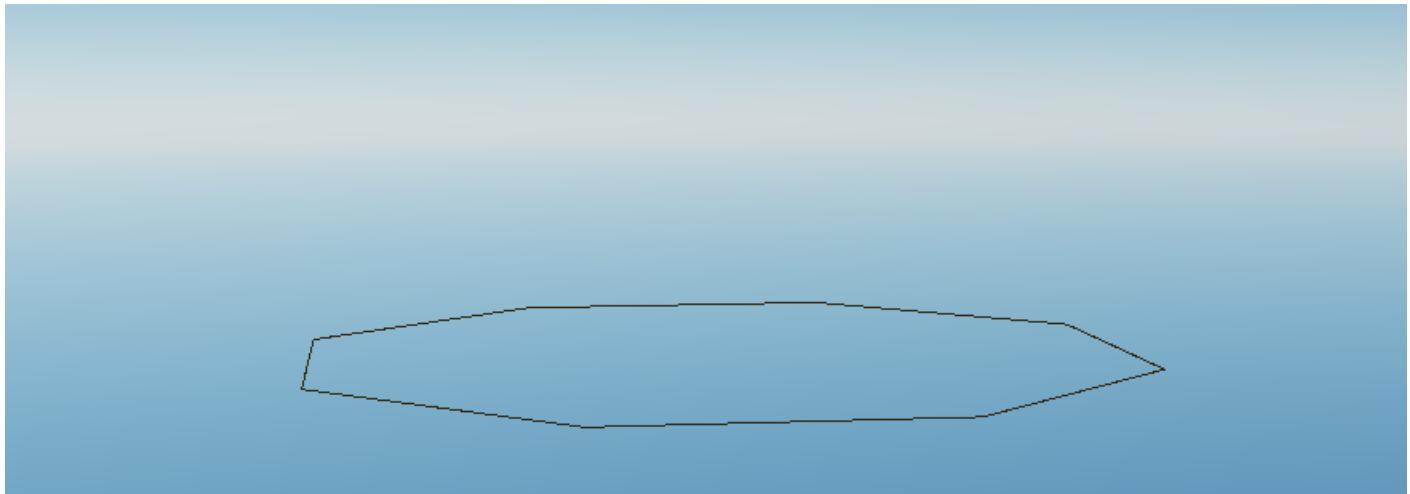
**There are two types:** Shell for surface collision and Line for line collision (e.g., used for aircraft landing gear collision).

Create a Collision layer in the scene containing all collision objects, each with its own dummies replicating the model's animations.

1) **Surface** - create collision geometry and specify the object type as "Collision Shell."



2) **Line** - an object with a line of vertices, specify the object type as "Collision Line."



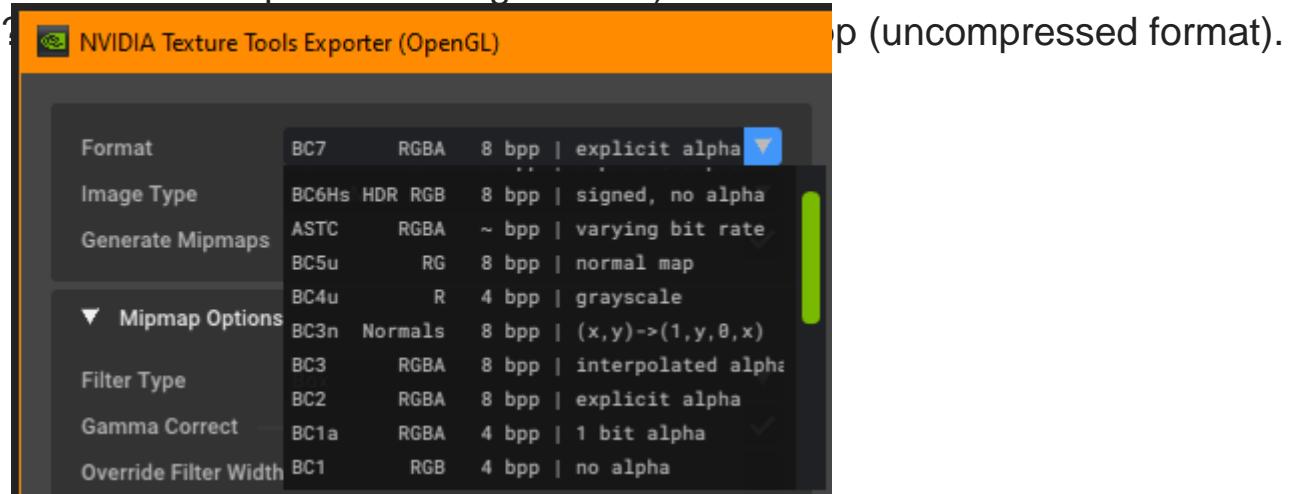
### Texture Format for Export

To optimize the game engine's performance, textures must be converted to DDS format, which requires installing the DDS plugin for Photoshop.

? Textures without an alpha channel are saved in BC 1 format.

? Textures with an alpha channel are saved in BC 3 format (can also be saved in

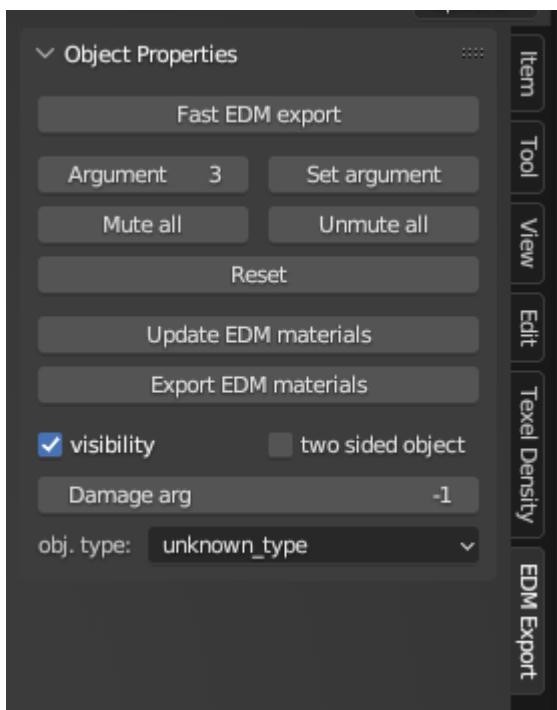
BC 2 without alpha channel gradation).



p (uncompressed format).

## Main Functionality of the Blender Exporter:

? The EDM Export window in the N-panel is used for working with arguments, specifying object types (connectors, collisions, etc.), visibility animation, two-sided display function, and damage argument specification.



\*Argument - number of animation argument

\*Set argument - disable all animations and enable only the selected argument

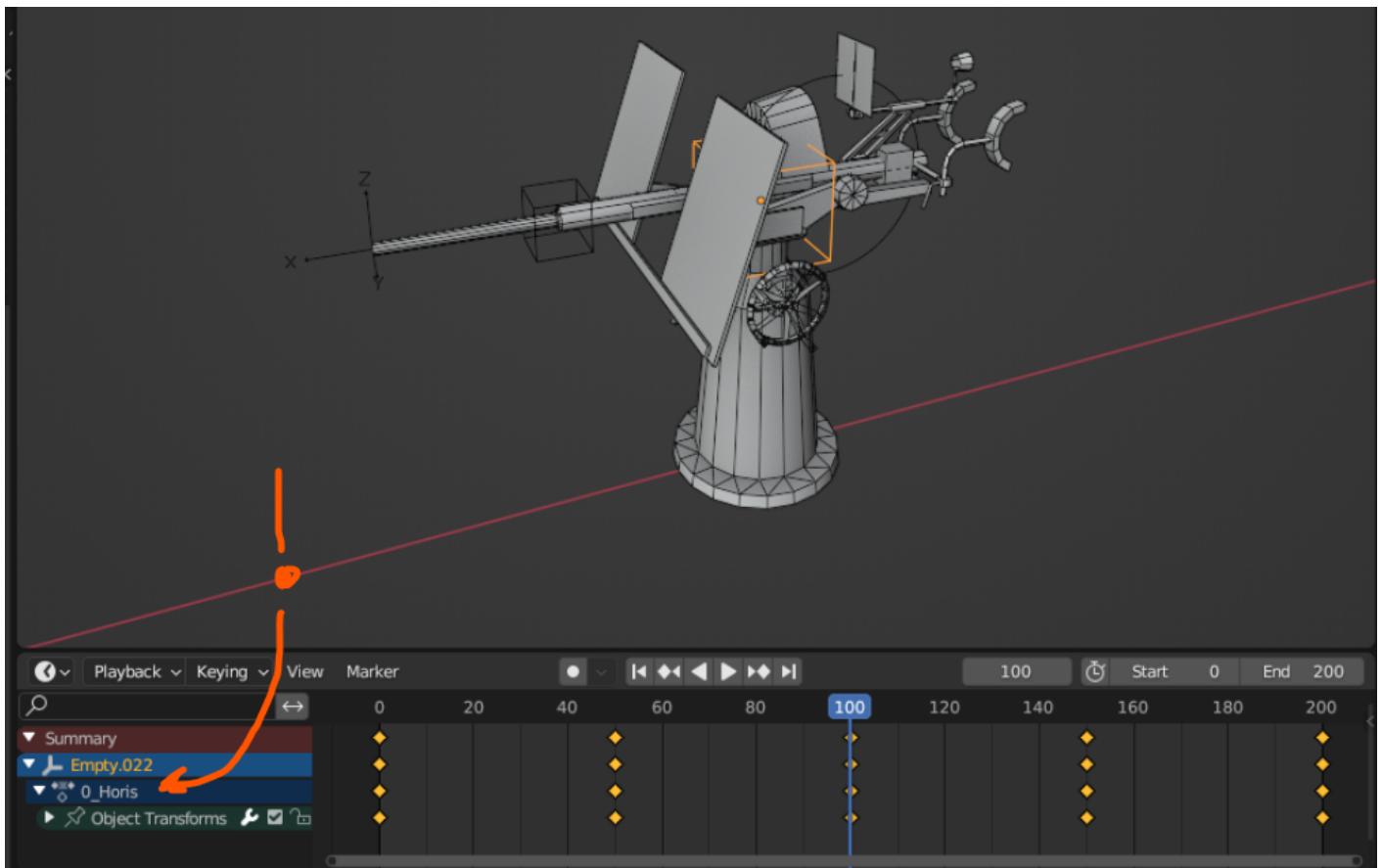
\*Mute all - disable all animations

\*Unmute all - enable all animations

\*Reset - reset all animations

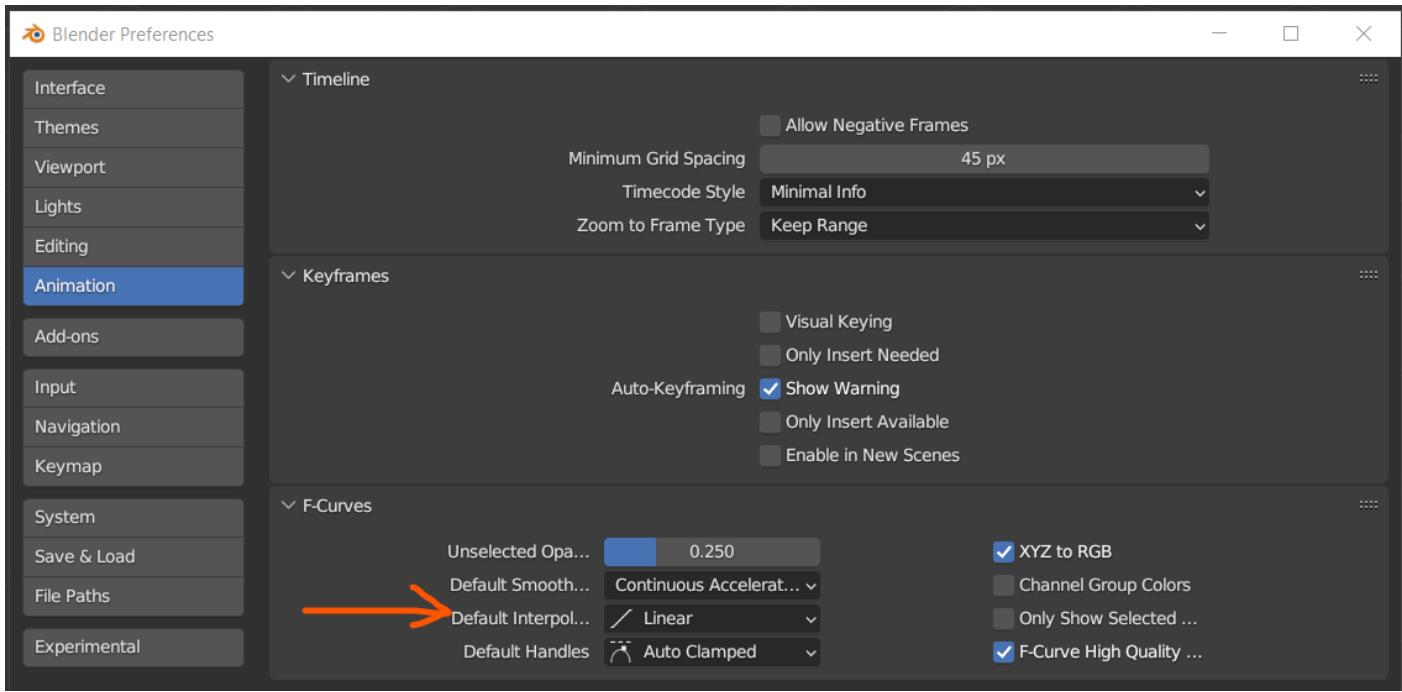
\*Update EDM materials - update all materials in the scene (if new versions are available)

? The Timeline window is used for specifying animation argument names. The argument is specified at the beginning of the action name: Example: 0\_Horis - the number at the beginning is the argument name, followed by an underscore and a brief description. Only the number before the underscore is important for export, the description is for user convenience.



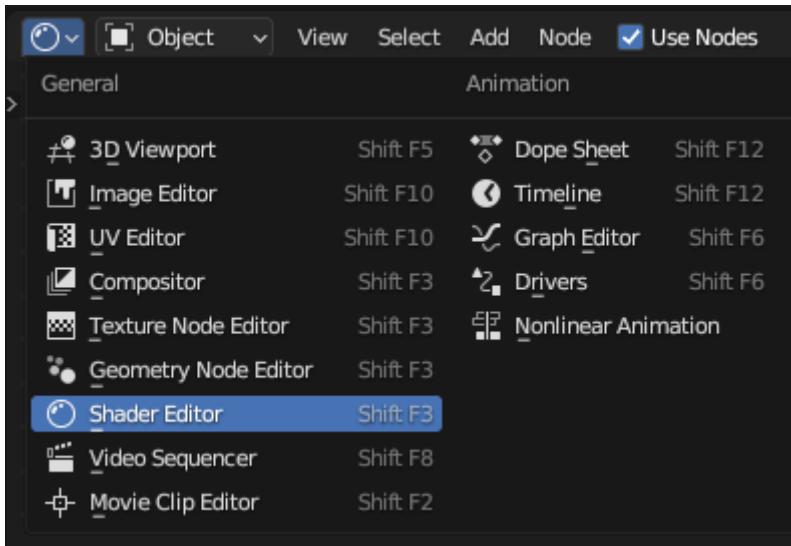
\* Blender uses an animation scale from 0 to 200, where 100 is the midpoint, neutral position. This scale is set automatically when pressing Reset in the EDM Export tab. It is recommended to press it before starting work with animation.

\* In the settings set the animation interpolation mode to Linear.



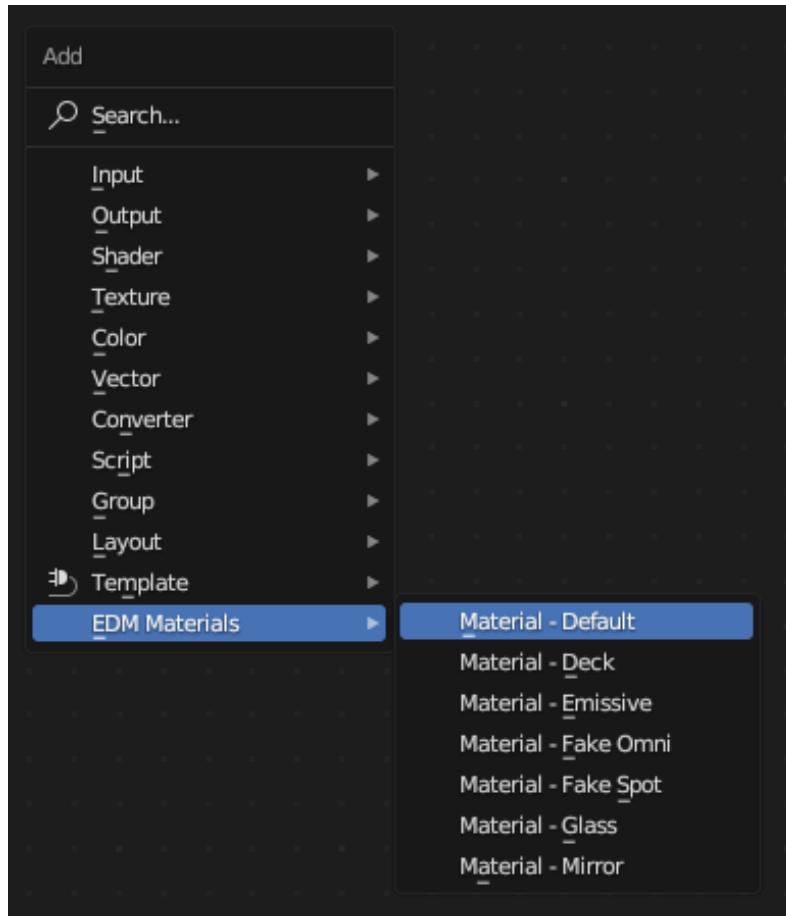
## Material Setup

? Open the **Shader Editor** window.

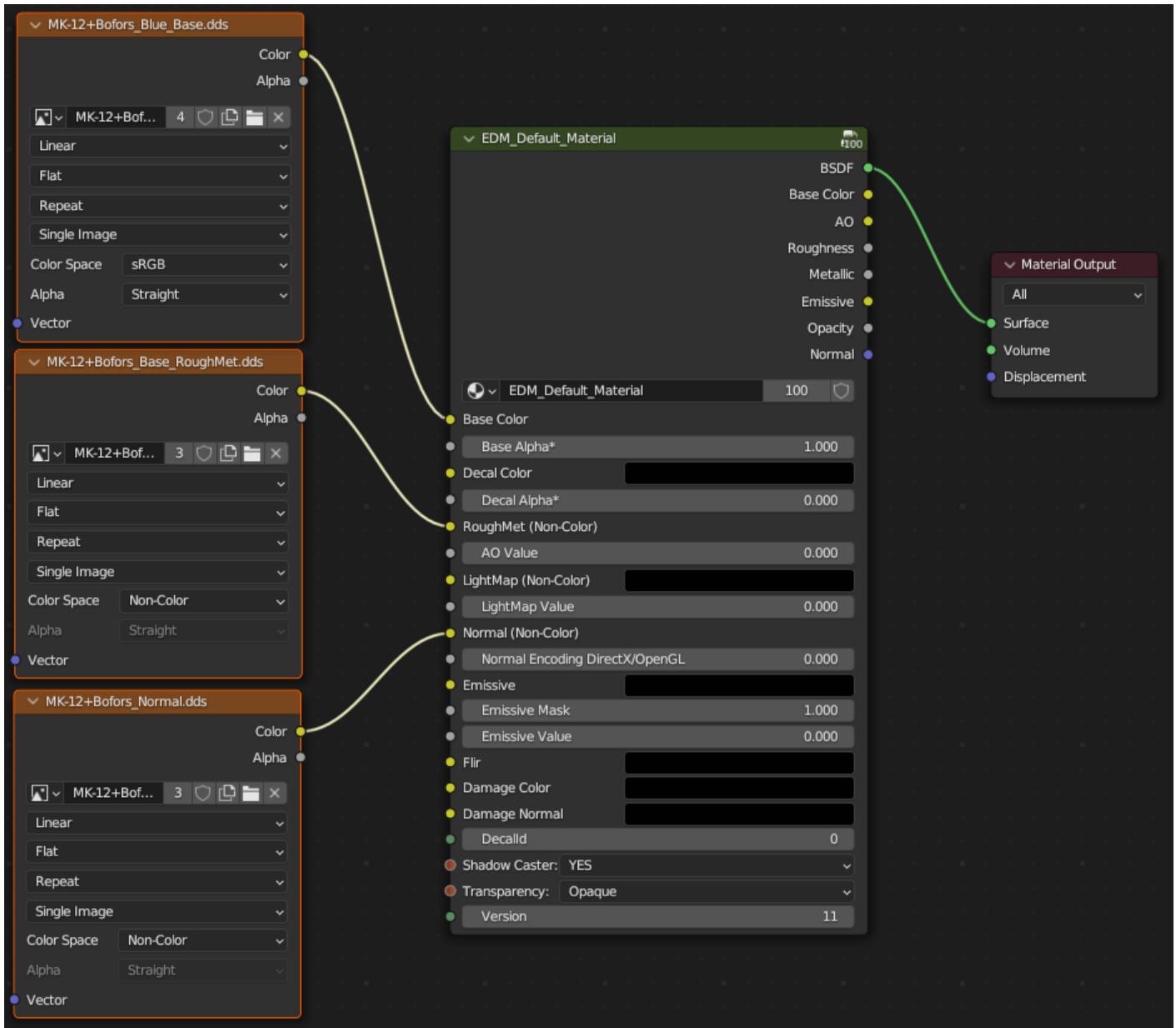


? For different types of materials add the appropriate "Nod" from **EDM Materials**.

Shift + A

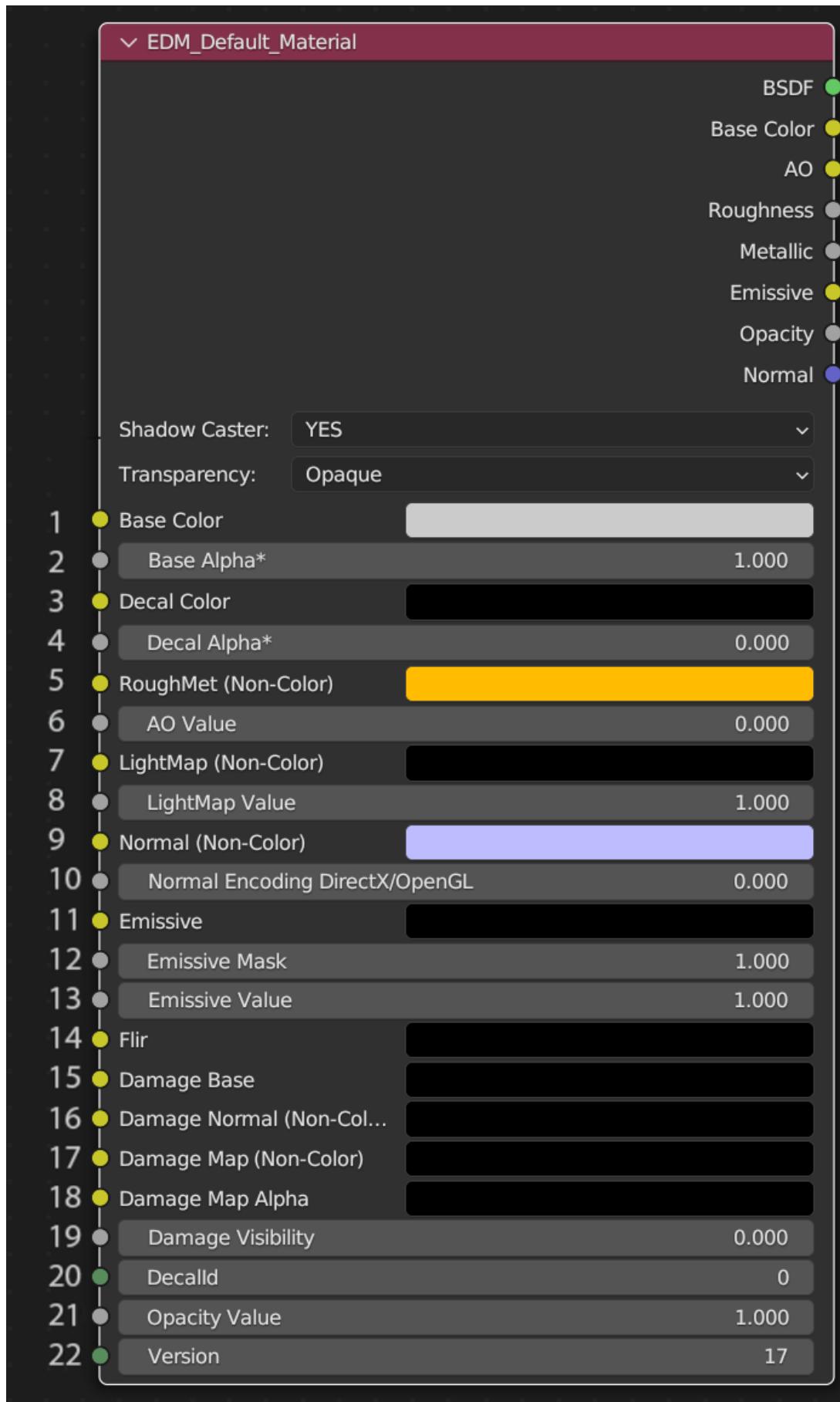


? Apply textures to the material as shown in the example:



- \* All inputs and outputs in the material group are labelled.
- \* Inputs (Non-color) - require switching the Color Space mode of the texture (this is only needed for correct display in the viewer and does not affect export).
- \* The right side of the group - outputs for displaying separate texture layers (Roughness, Metallic, etc.)
- \* Enable the Node Wrangler add-on for convenience.
- \* At the bottom of the group - additional material settings: material transparency, shadows, decal ID.

## Listing Input Values ??in the Material



\* Some input values are only for visualization in the Blender viewer and *do not* participate in export: (AO Value, LightMap Value, Normal Encoding)

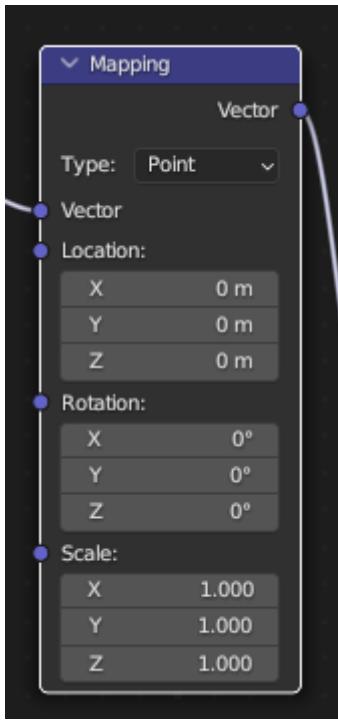
- 1** - Base/Albedo;
- 2** - Alpha Base/Albedo- to display transparency (does not affect export);
- 3** - Albedo of Decal (second layer);
- 4** - Alpha of Decal (does not affect export);
- 5** - RoughnessMetalic texture;
- 6** - Visibility of ?? from RoughMet texture;
- 7** - LightMap (unic ??) texture;
- 8** - Visibility of LightMap;
- 9** - Normal texture;
- 10** - DirectX/OpenGL switcher for Normal (0-DirectX, 1-OpenGL);
- 11** - Emission texture;
- 12** - Emission texture mask;
- 13** - Emission strength regulator;
- 14** - FLIR texture;
- 15** - Damage texture map;
- 16** - Damage Normal map;
- 17** - Map/Mask of Damage;
- 18** - Alpha chanel of Damage Map (does not affect export);
- 19** - Damage display scale;
- 20** - Decal layer identifier (needed to correctly display the order of drawing transparent objects);
- 21** - \*\*\*.
- 22** - Material version;

**"Shadow caster"**-shadow rendering mode.

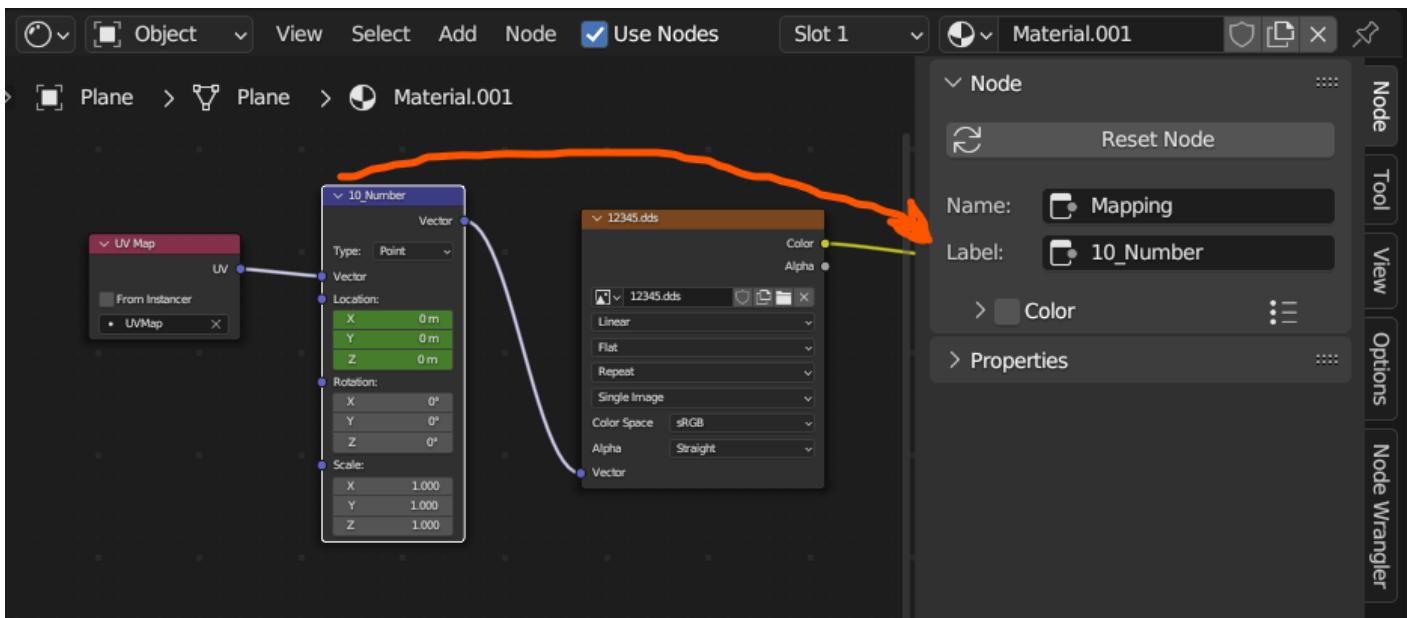
**"Transparency"**-material transparency mode.

### UV animation

To animate UV offset, create a Mapping node. Ctrl+T (with the Wrangler addon enabled).

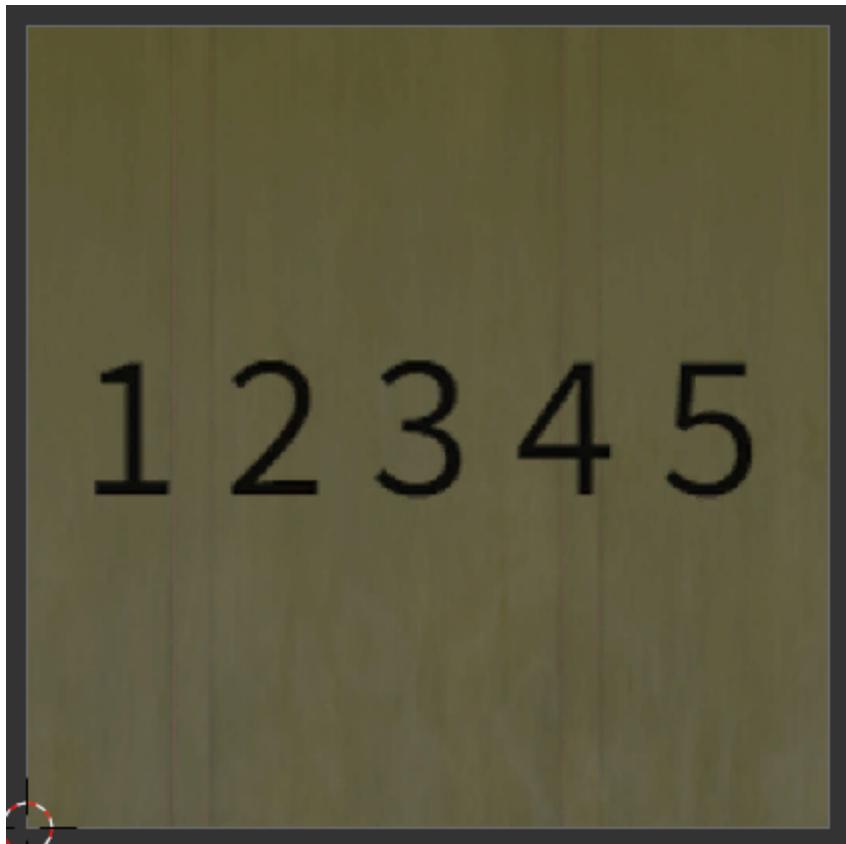


Specify the animation argument in the Label. The node name will change to the argument name with its description. Attach a UV node with the specified UV channel.

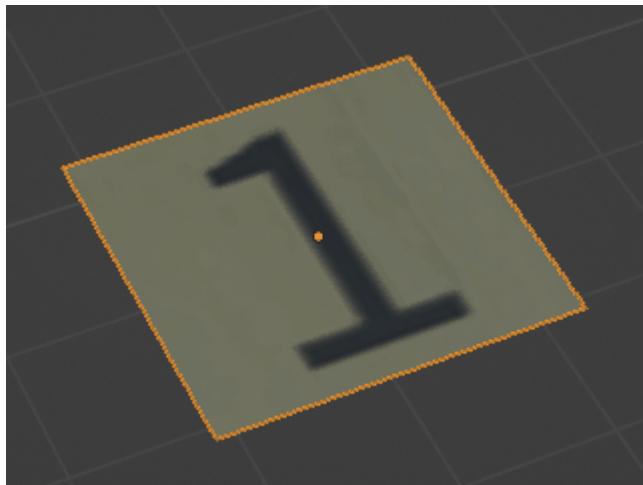


Create an animation by changing the Location and saving keys (key "I").

Example of an atlas texture.



Example of animation.



? Roughness Metallic and Normal is shifted by the same Mapping node as the Base.

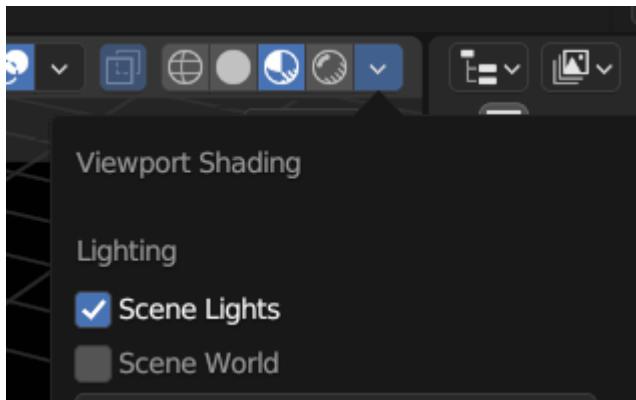
## Lighting

Divided into lamps (providing real light) and fakes (images - light points)

### Lamps

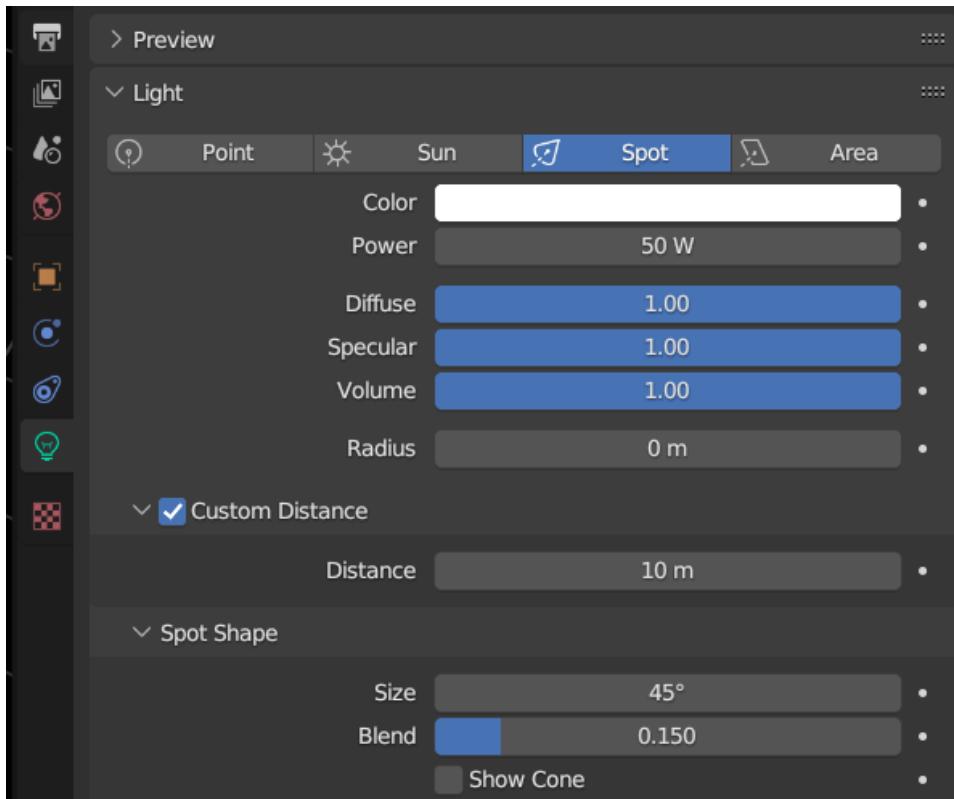
! It is recommended to work in Eevee render mode.

! To adjust the light setup, use Material View display mode with Scene Lights enabled or Rendered mode.



!!! The exporter tries to transfer lamp characteristics to the engine as accurately as possible so that they look the same in Blender, but differences will still exist due to technological rendering differences. Therefore, lighting is set up in Blender first, then adjusted by looking at the viewer.

Supported lamp types: Omni and Spot.



Characteristics used for export:

Color - light color.

Power - intensity.

Specular - ability to cast glare.

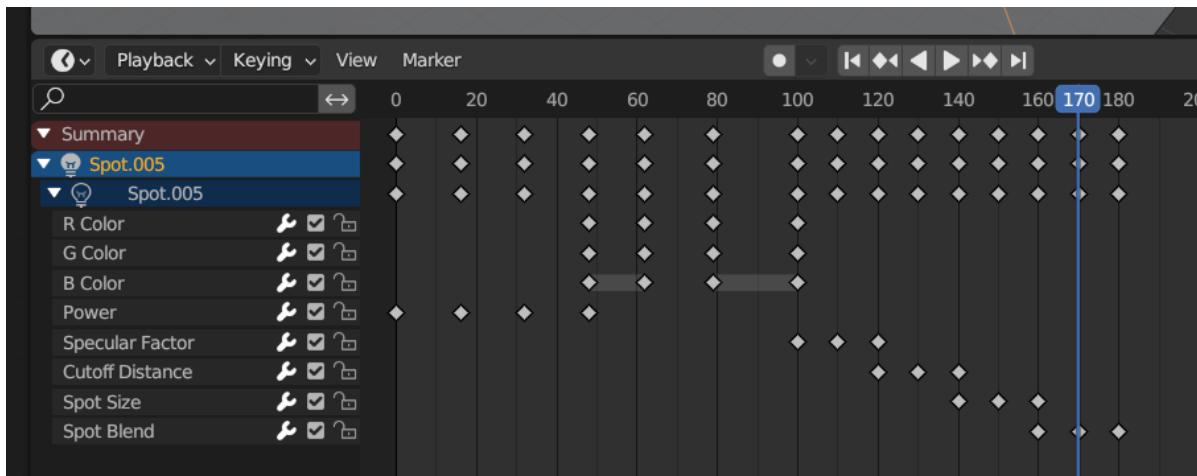
Custom distance - specifies the lamp's working distance (default is 50m if unchecked).

Spot shape - used for Spot lamps: Size - outer light cone angle.

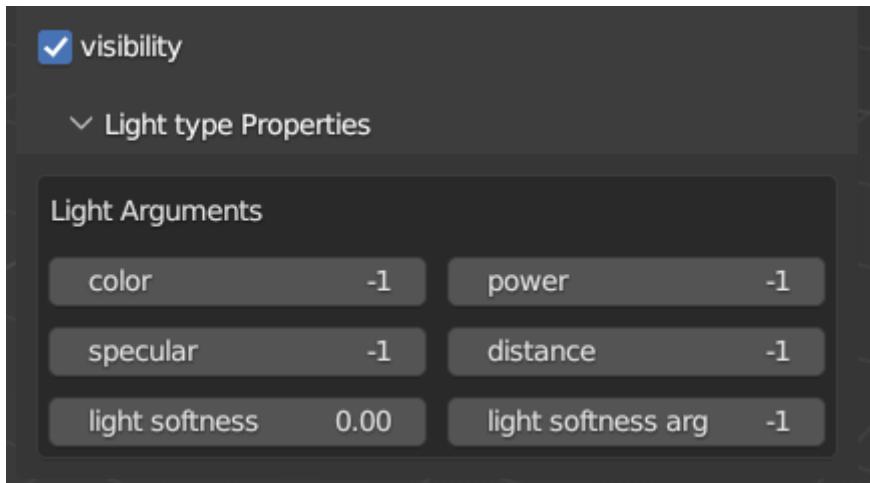
Blend - softness of the light border (0-hard, 1-soft).

Lamp characteristics animation - performed in the "EDM Export" panel.

First, create the animation.



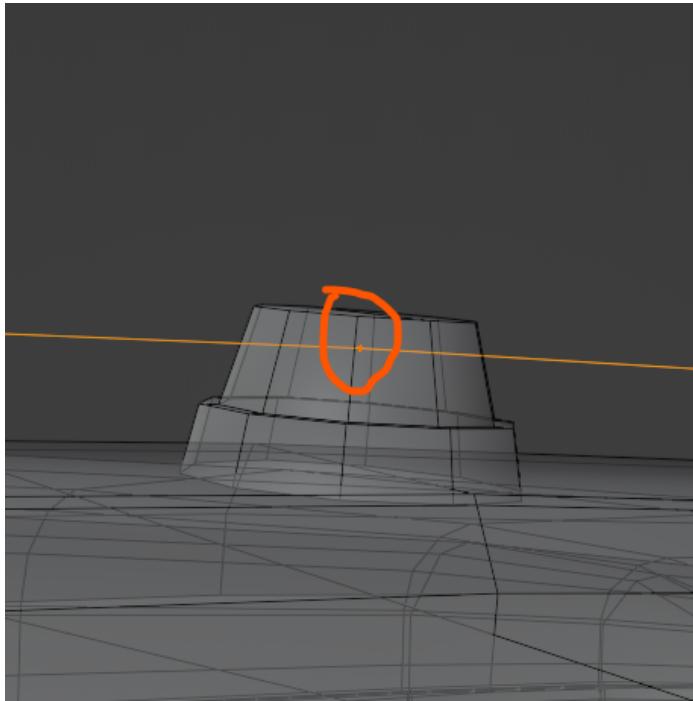
Then specify its arguments in the N-panel.



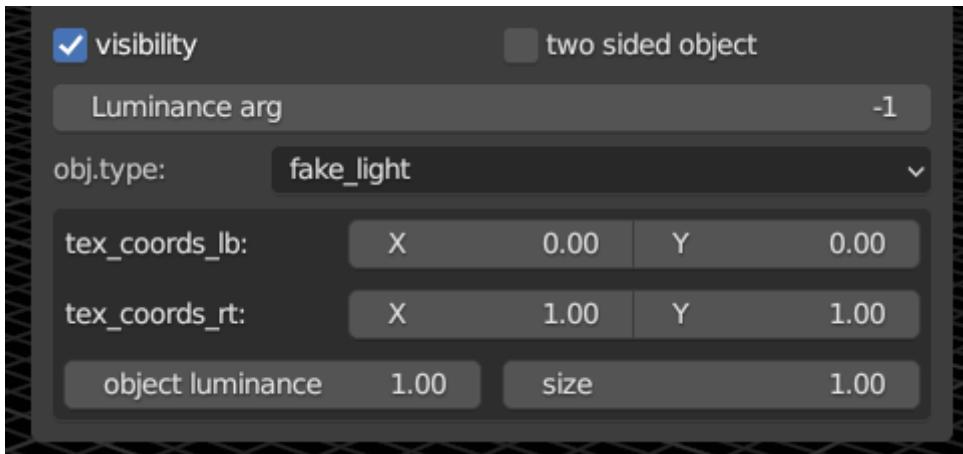
\* Numbers are arguments.

## Fake Lights (BANO-light points)

\* Implemented using geometry with vertices as light points.



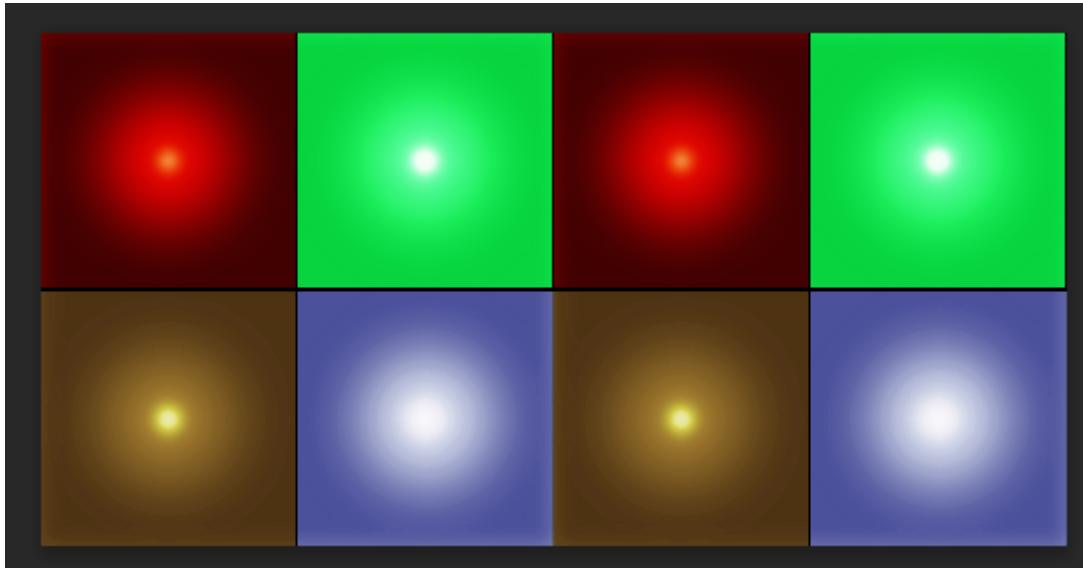
Specify the object type as "Fake\_Light" for the created geometry.



Then specify the edges of the light point texture area inside the atlas (if any):

\*`tex_coords_lb` - lower left edge

\*`tex_coords_rt` - upper right edge



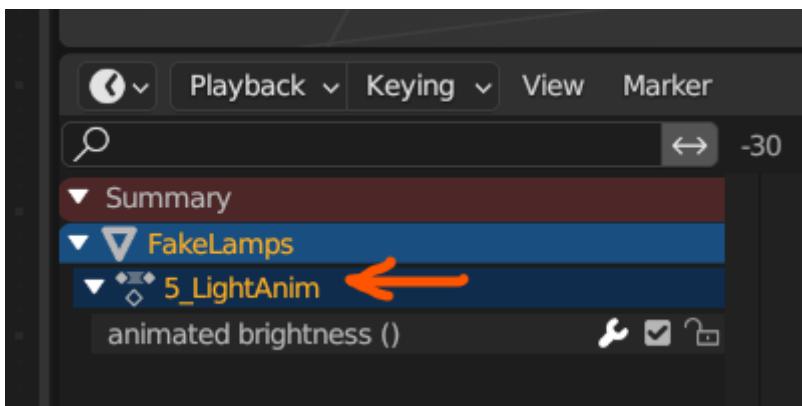
\* Example of a texture atlas. (the right four have a truncated alpha channel shape and simply look similar in colour).

\*\* Example of red "ball" UV tex\_coord\_lb: x=0.0;y=0.5; tex\_coord\_rt: x=0.25;y=1.0

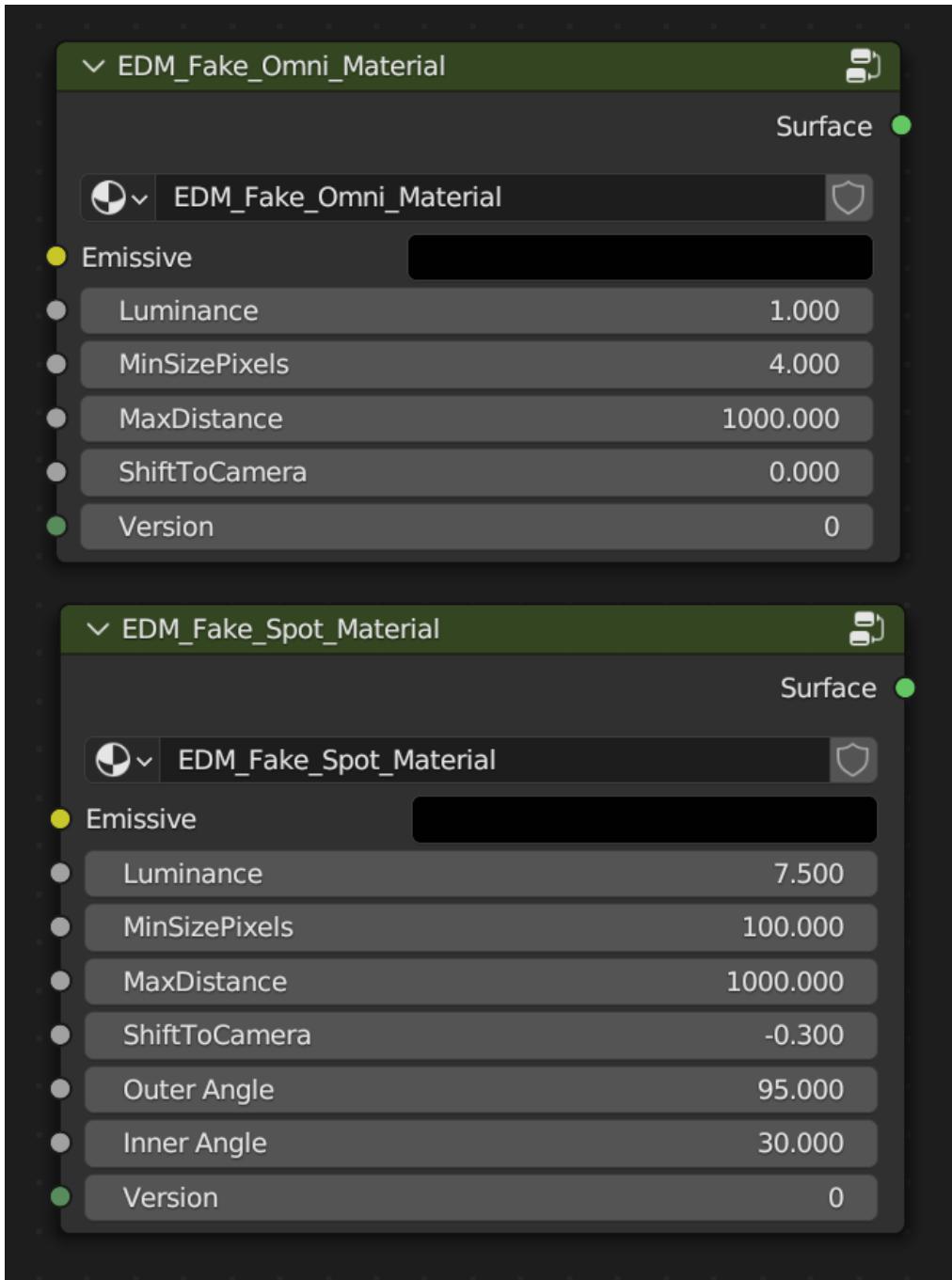
\*\* Example of blue "ball" UV "ball" tex\_coord\_lb: x=0.25;y=0.0; tex\_coord\_rt: x=0.5;y=0.5

Size - specifies the size of the banner/light in meters.

Object luminance - specifies brightness of .



Create a node in the material: EDM\_Fake\_Omni\_Material or EDM\_Fake\_Spot\_Material.



\*\*\* Specify characteristics inside the group:

Luminance - the brightness of the light,

MinSizePixels - the minimum size of the light spot when moving away,

MaxDistance - the distance after which the light will disappear,

ShiftToCamera - shift the light plate towards the camera (to remove noticeable intersection with the geometry),

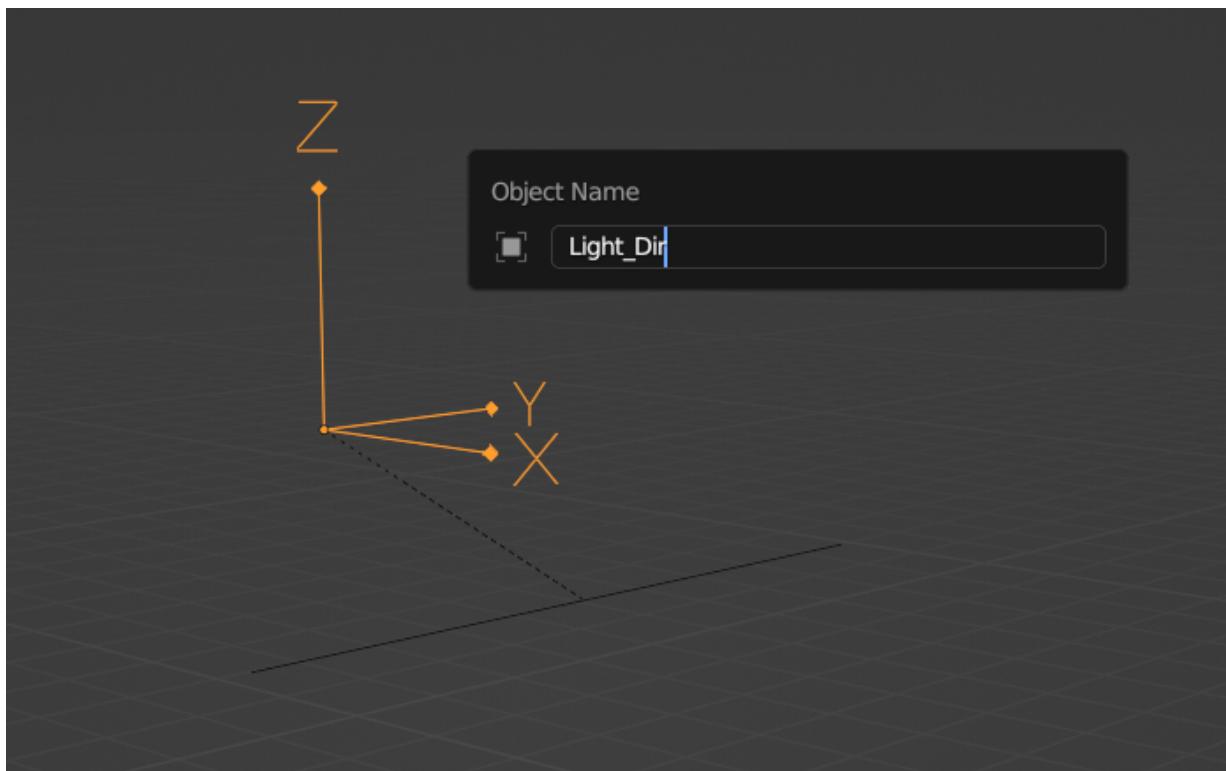
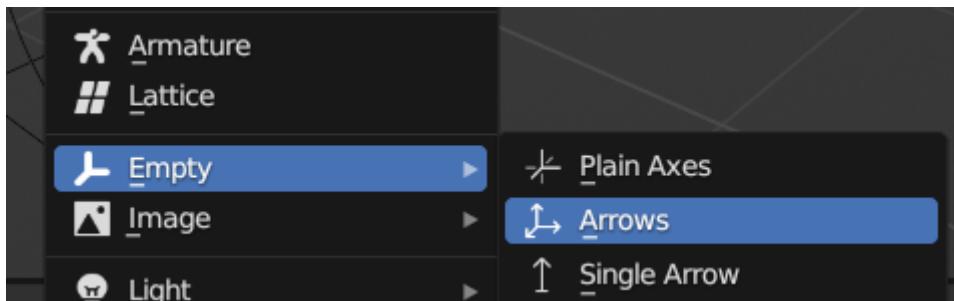
OuterAngle - the outer angle of the directional light (spot),

InnerAngle - internal.

Add the texture to the Emissive channel in the group.

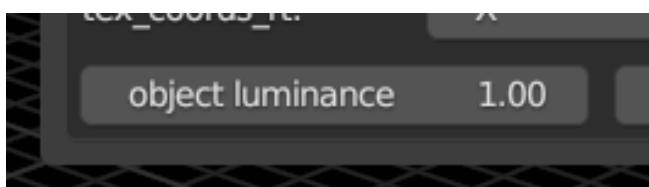
For Fake Spot, you need to set the direction (where it is looking). To do this, assign Empty( Arrows) to fake spot geometry. Lights direction will be defined by

direction of X - axis.

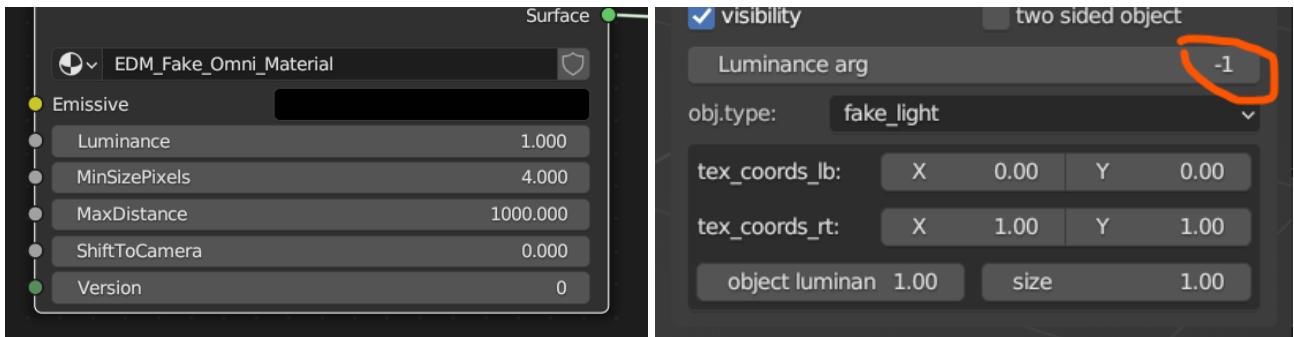


Brightness animation for Fake\_Light (blinking) can be set:

**1) for Fake Lights geometry** - using Object luminance animation in the EDM Export panel and specifying the animation argument in the action.



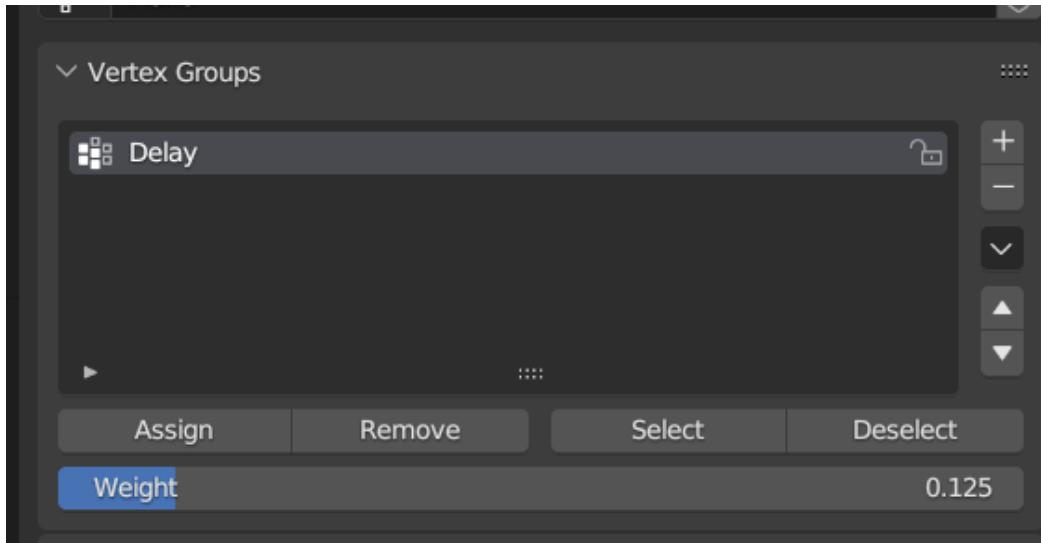
**2) For material** - an animation of the Luminance parameter is created and the animation argument is specified in the Luminance arg in EDM Export.



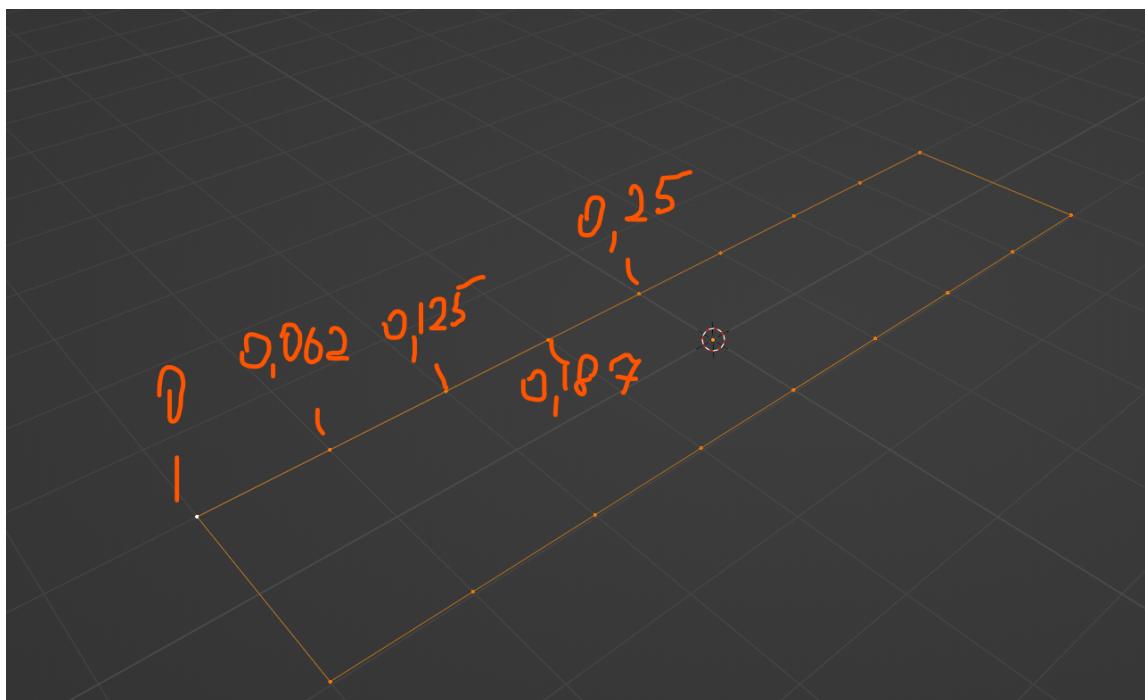
\* If animation is created both in geometry and material, geometry animation takes priority.



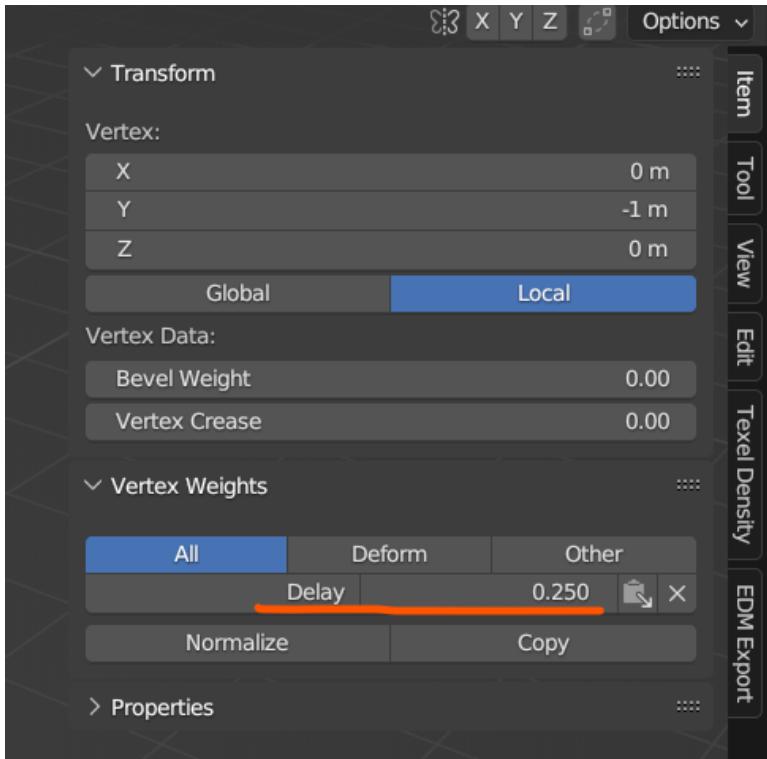
**Rabbits Lights** - a Fake Lights function where brightness changes along a lamp line (a light moves along the line). Used for landing strips on ground and aircraft carriers. Works for Omni and Spot. A general brightness animation (flash) is created and the delay/lag time of the lamps is specified using Vertex Group.



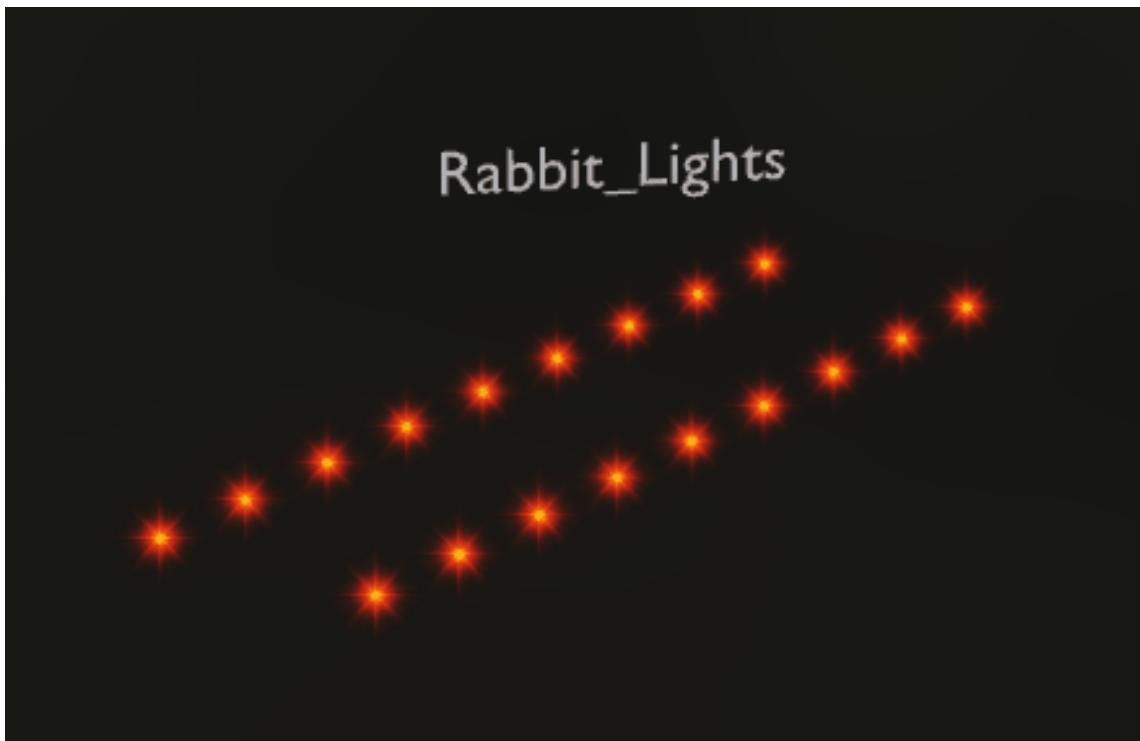
Example:



Each vertex's delay is easily adjusted and checked in the Item menu.



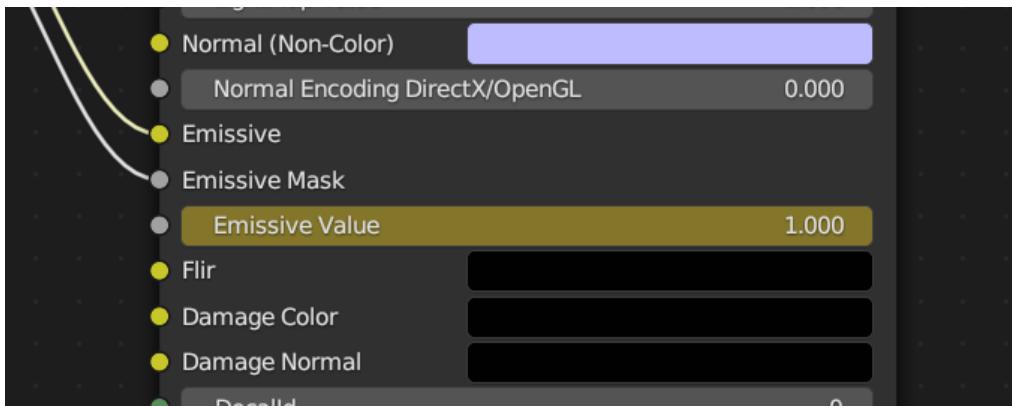
Result - light moves along the line.



## Emission Materials (glow)

**Emission** is a property of Default material. It is turned on by changing color of “Emissive” to some other than black.

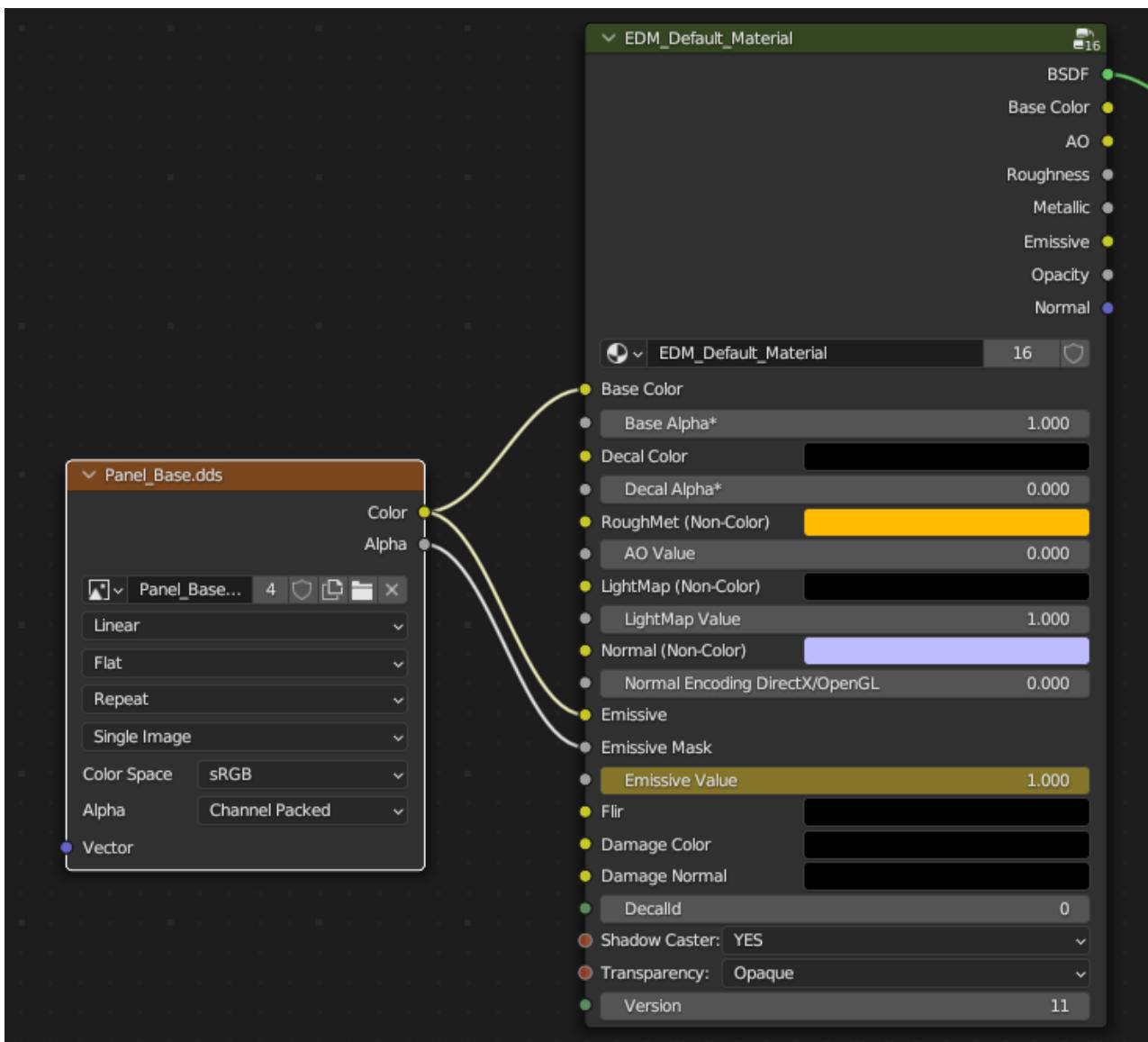
**Emission** value is controlled by Emissive Value. Animated value is allowed.



Two main glow options are used:

**1) Self Illumination** - default mode - weak lighting. For example, for cockpit elements backlighting.

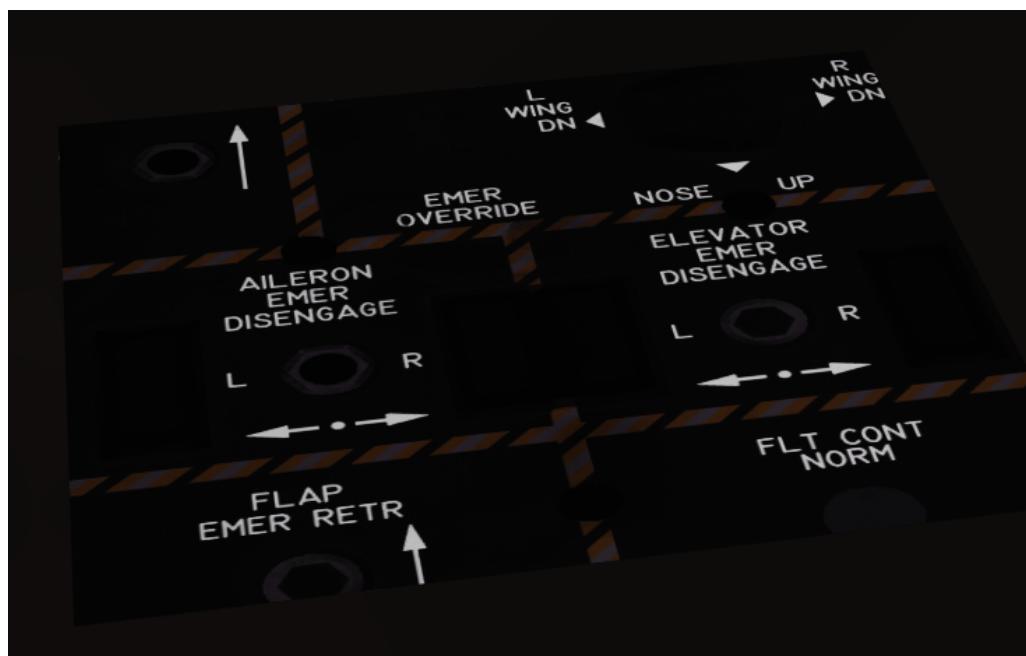
Glow texture usage mode - connect it to the Emissive channel.  
Alpha channel to Emission mask.



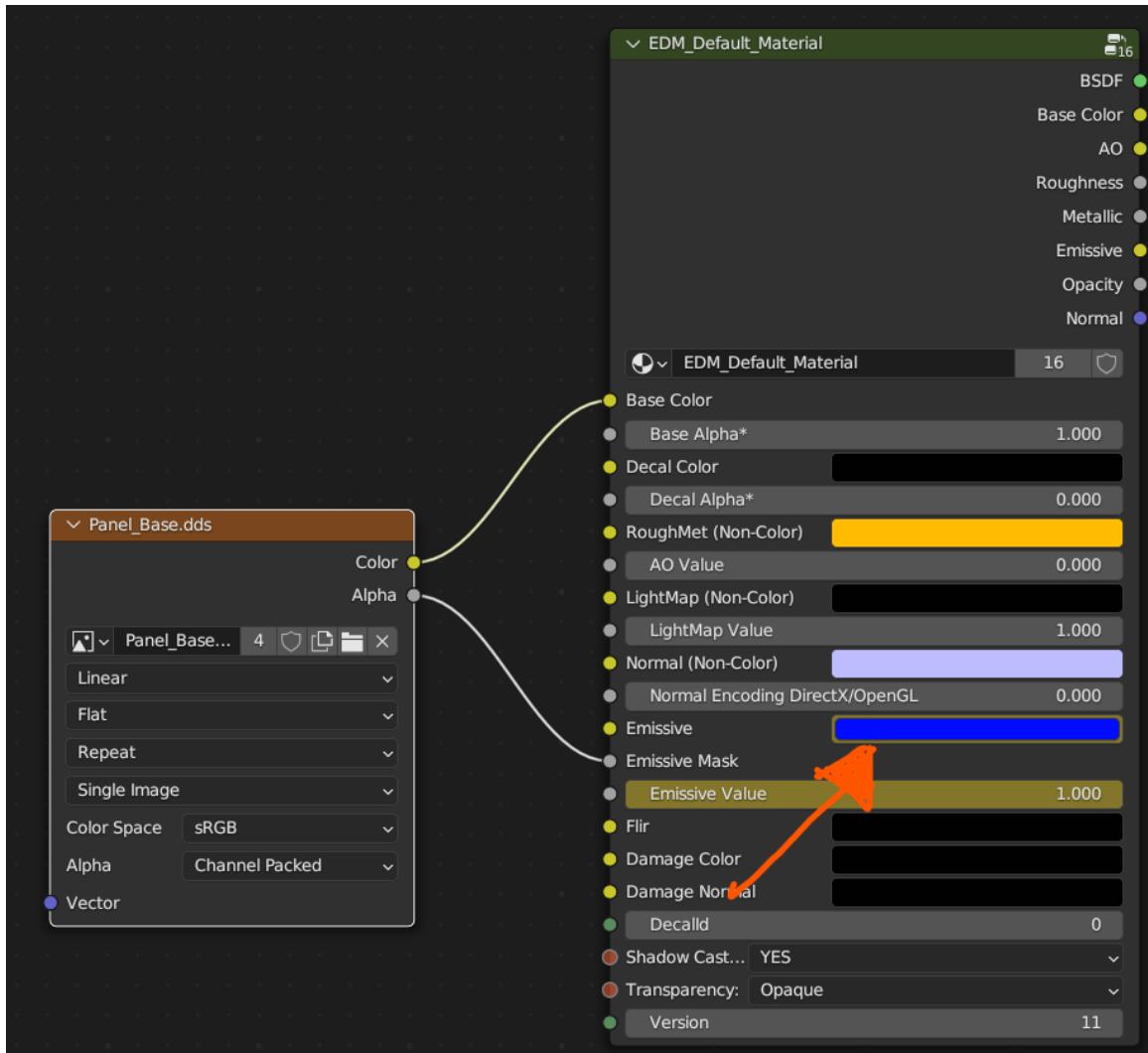
Glow color comes from the texture, the mask (alpha channel) shows what glows and what doesn't.



Result - cockpit panel inscriptions are backlit with the selected colour.



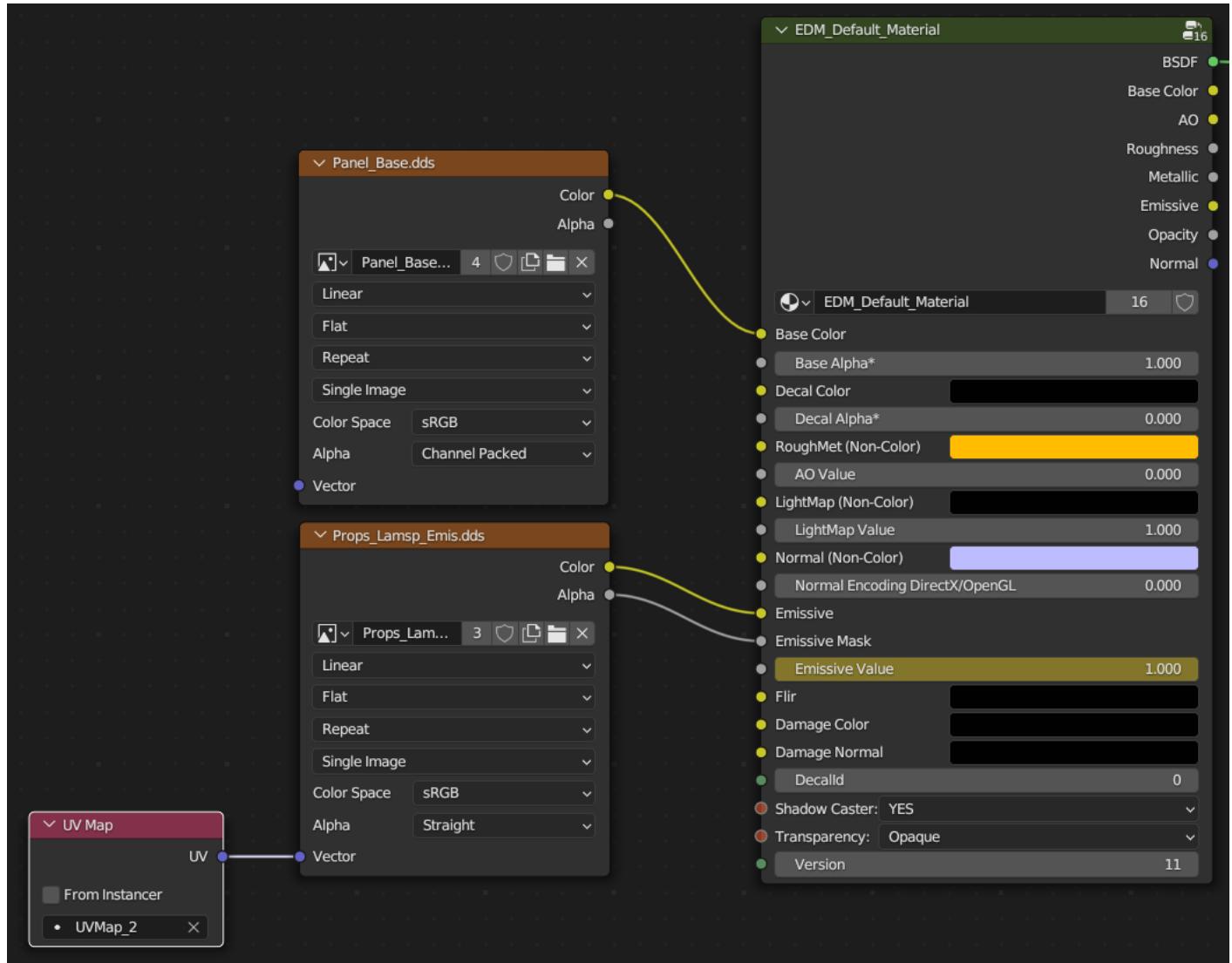
The mode of using the selected colour is to connect the alpha channel of the texture. In the Emission mask, select the colour in the Emissive slot.



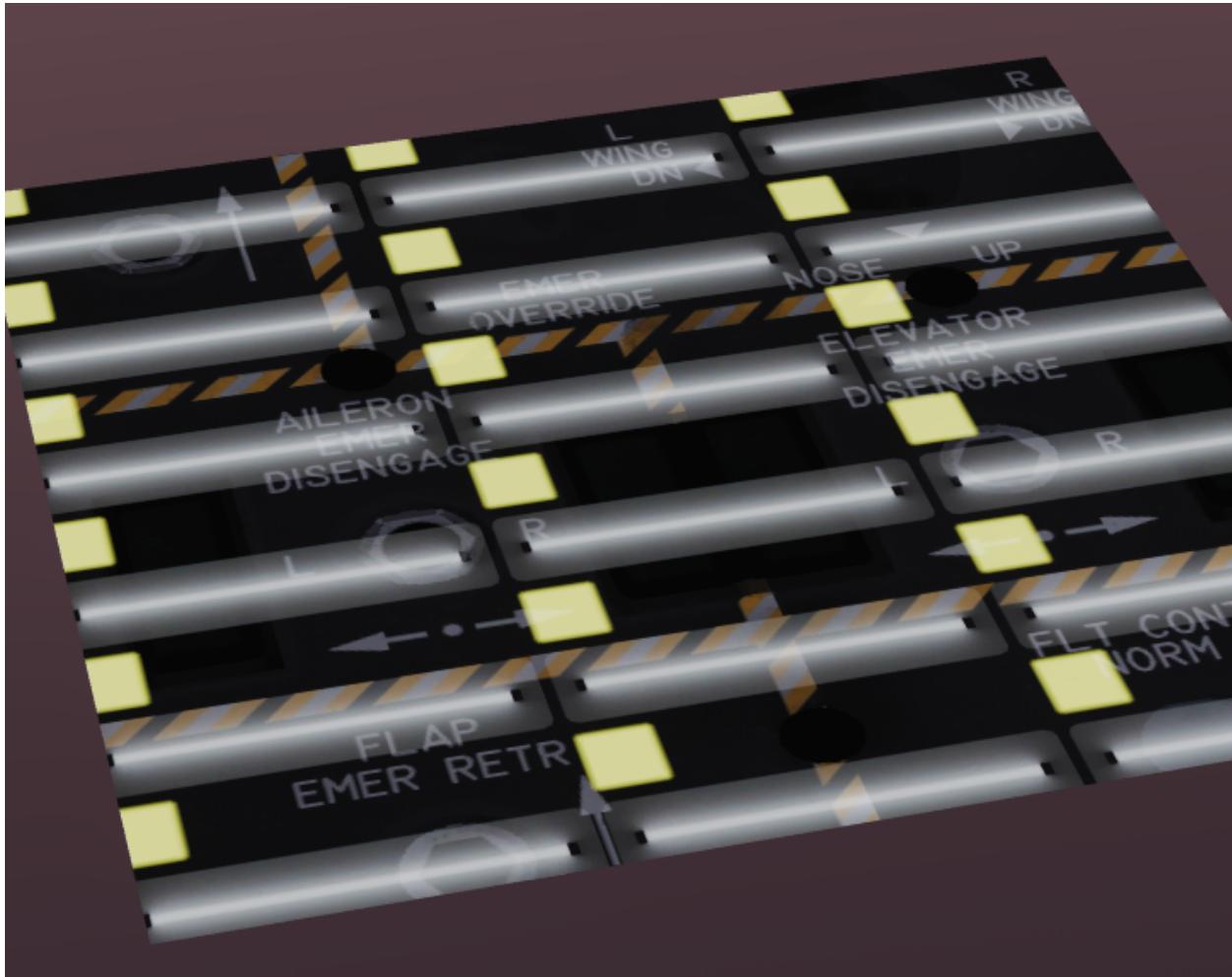
Result: the panel inscriptions is backlit by the selected color.



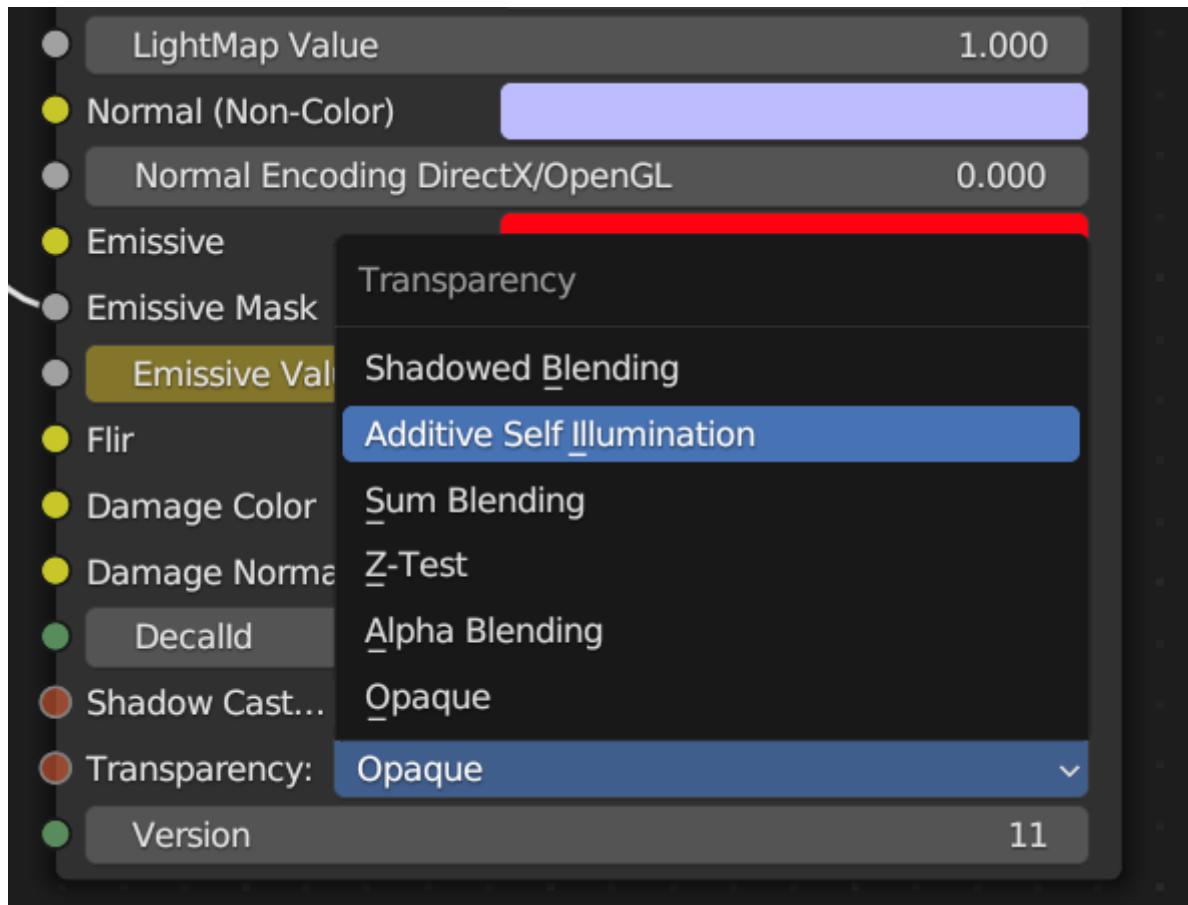
It is possible to set a different emission texture and give it a different UV. Its alpha also can be used as mask.



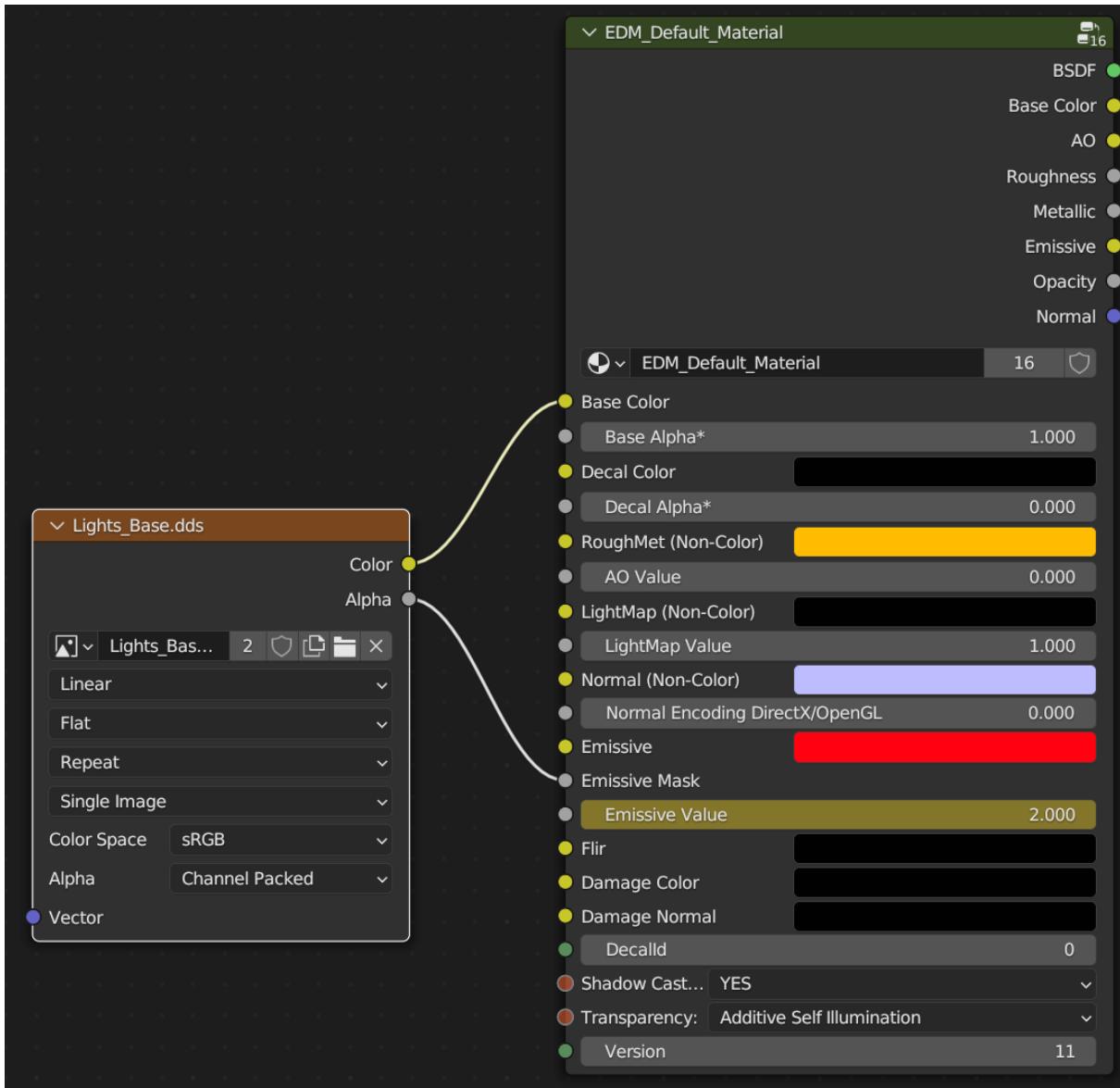
Result: the panel is illuminated with a different texture.



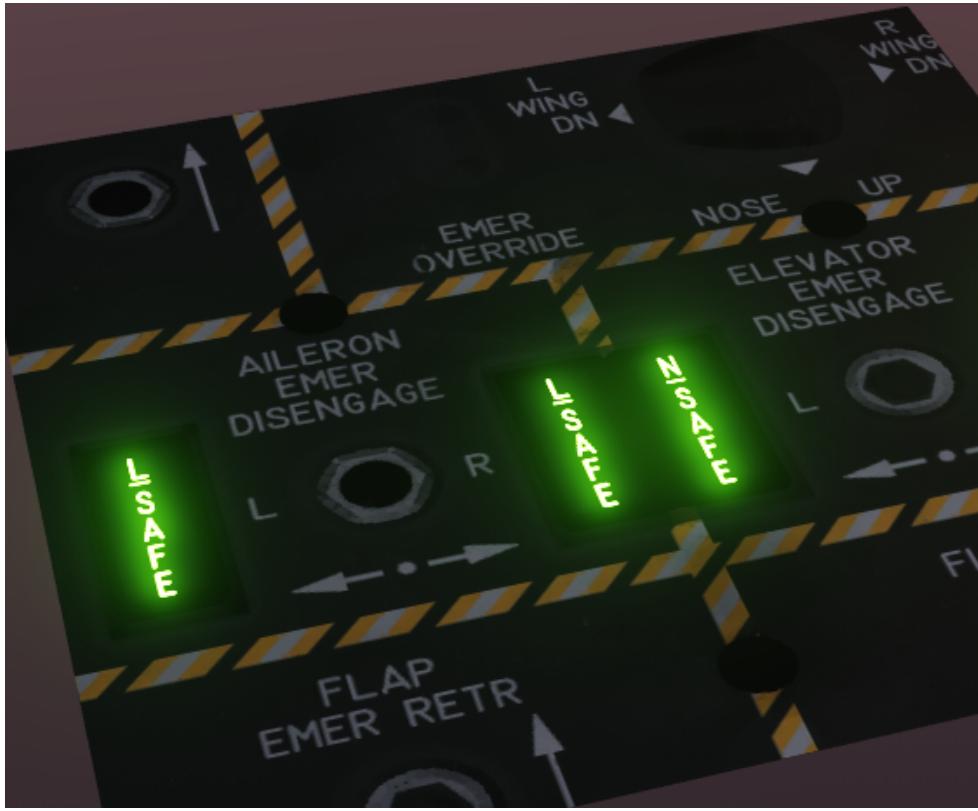
2) **Additive Self Illumination** - bright glow. For bright (even blinding) indicators in cockpits, headlights/lamps of equipment, light spots on the deck. Switch to this mode by enabling **Additive Self Illumination** in transparency settings and setting the Emission Value.



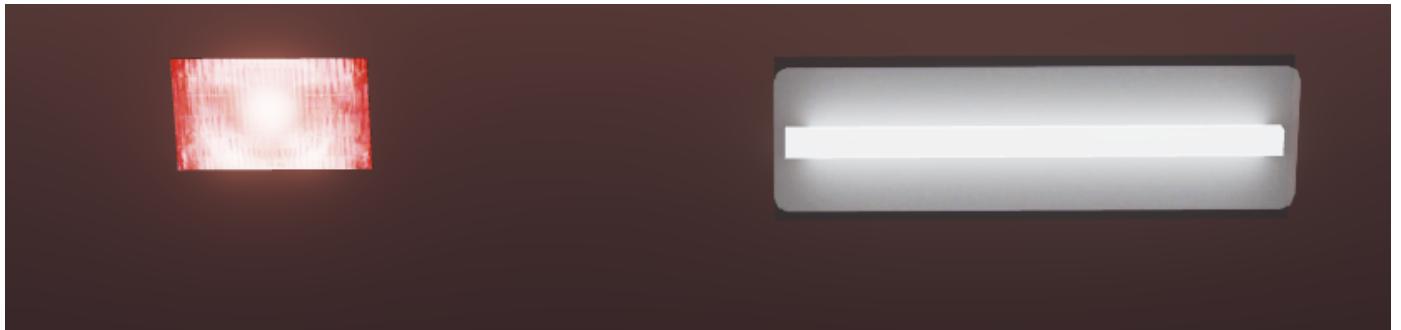
Uses texture from the Base Color channel and the mask from Emissive Mask.

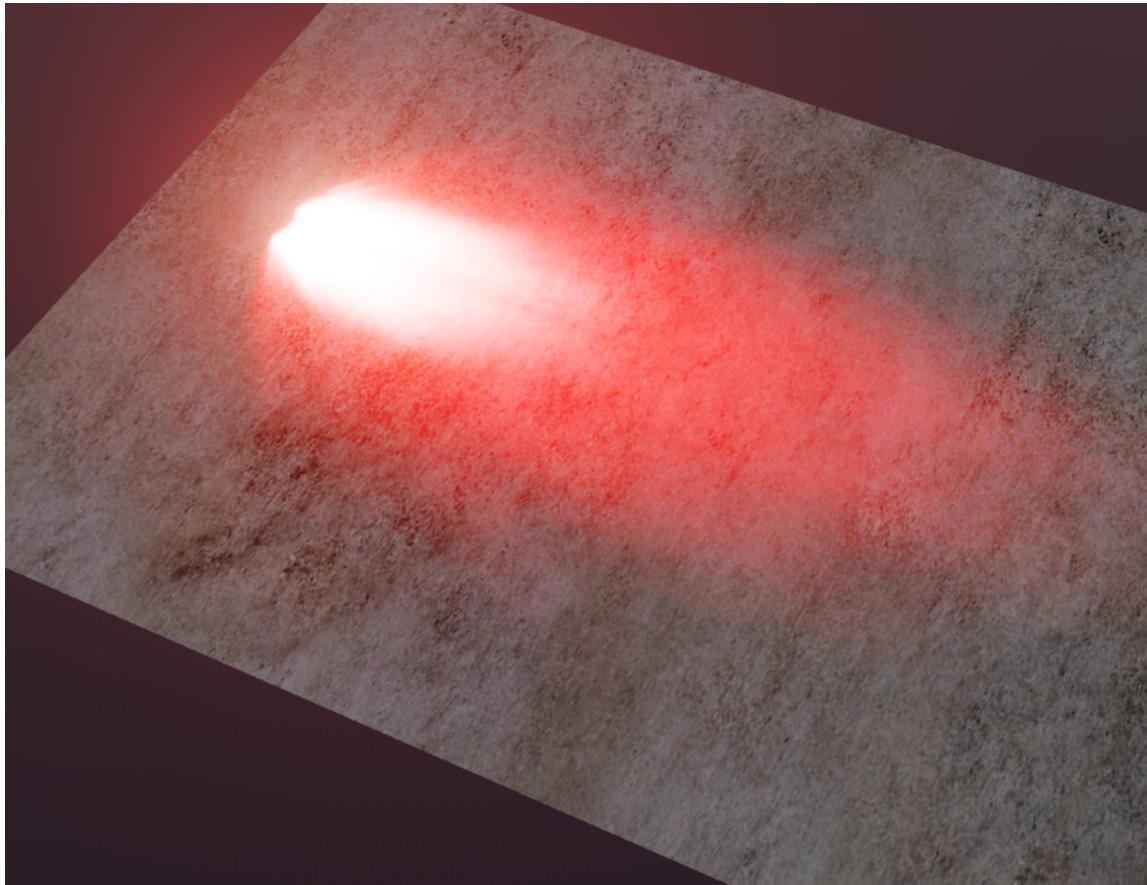


Result: the panel inscriptions glow brightly.

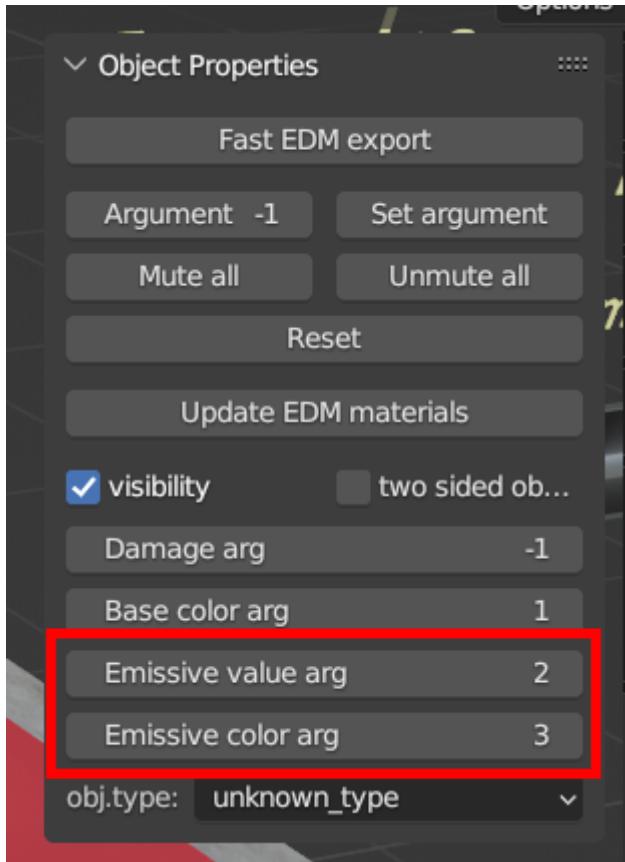


\*\*\*Examples of Additive Self Illumination: Headlights/lamps and bright spot of light on the deck.





The brightness and the selected colour can be animated - create animation keys in the material for the **Emissive Value** and the **Emissive color** (key "I"), specify the argument number in the "EDM Export" tab of the N-panel.

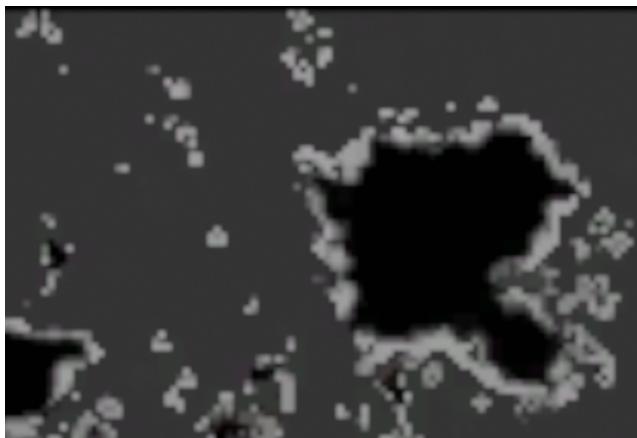


## Damage Material (Texture damage)

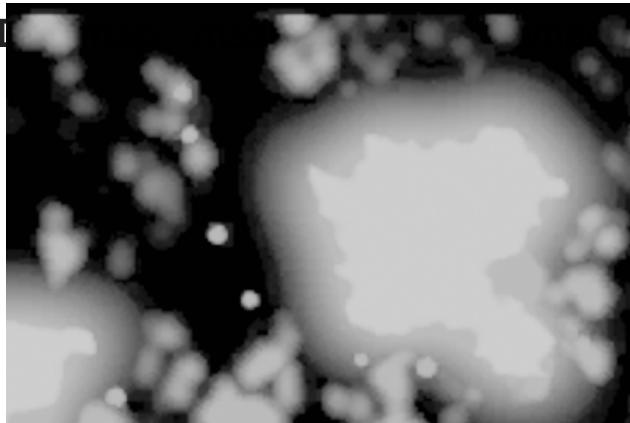
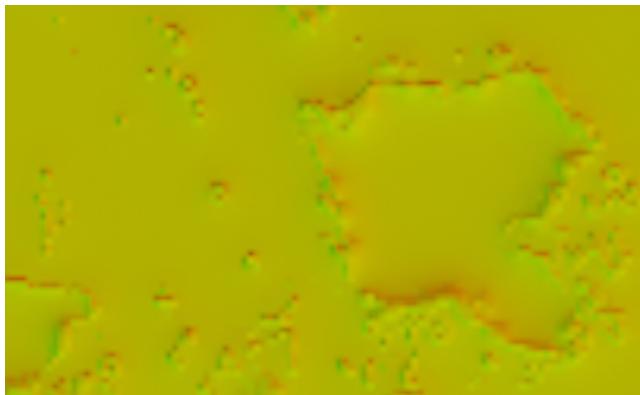
**Damage material (texture damage)** allows creating damage gradation for an object's material using textures. Used to show minor (bullet/shrapnel) and medium damage - holes.

Textures used:

**DM\_Base** - displays the damage itself



**DM\_Normal** - normal maps for damage

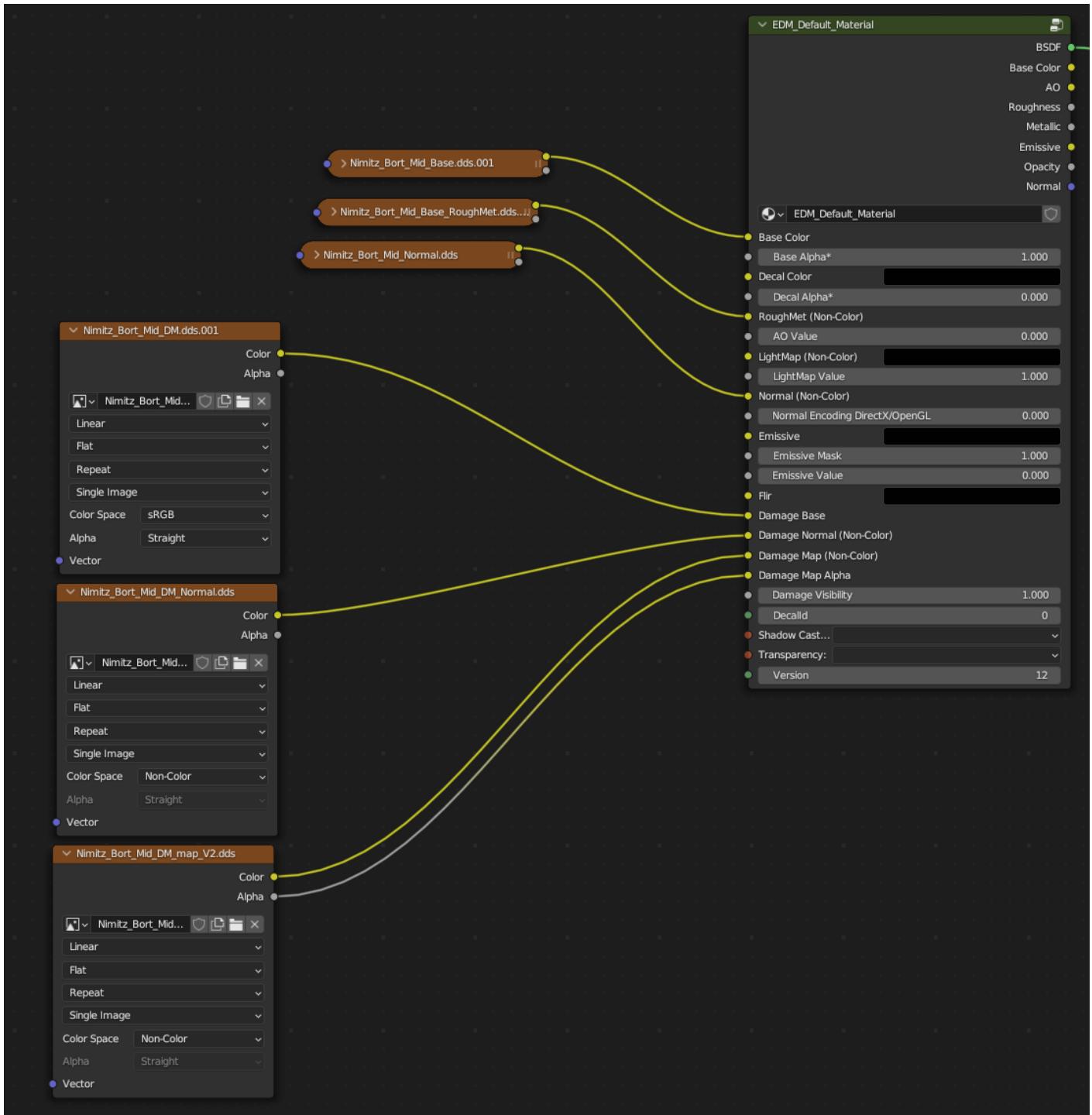


animation frames.

Anything brighter than <0.85 on the mask will be **drawn (rendered)** as a transparent hole.

!!! Blender does not work with 3D Volume DDS textures previously used to store four damage gradation frames. Now, BC3 DDS is used, with frames as RGB(1,2,3) layers and Alpha(4).

Connect textures in the material to corresponding inputs:

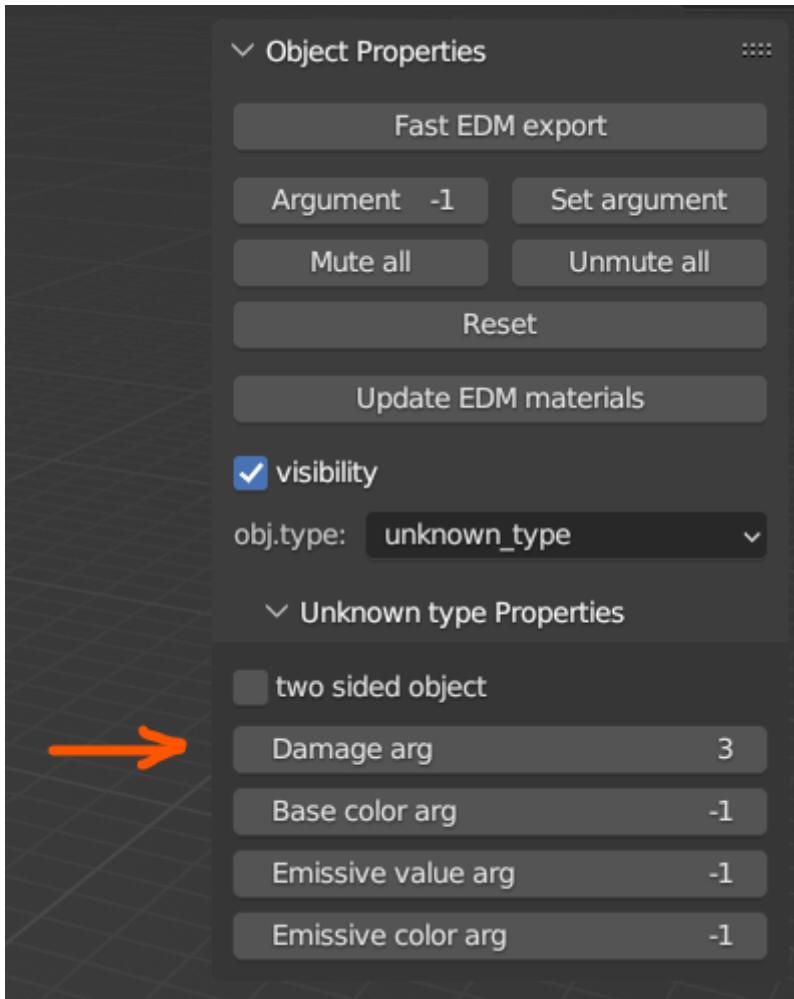


Damage Visibility is needed to show damage in the blender viewport.

\*\*\* For viewing, connect the DM\_Map alpha, as it contains the fourth damage frame. Necessary for Blender viewing, does not affect export.

You can set an argument either to an object (the whole thing will damage) or its parts (using vertex groups).

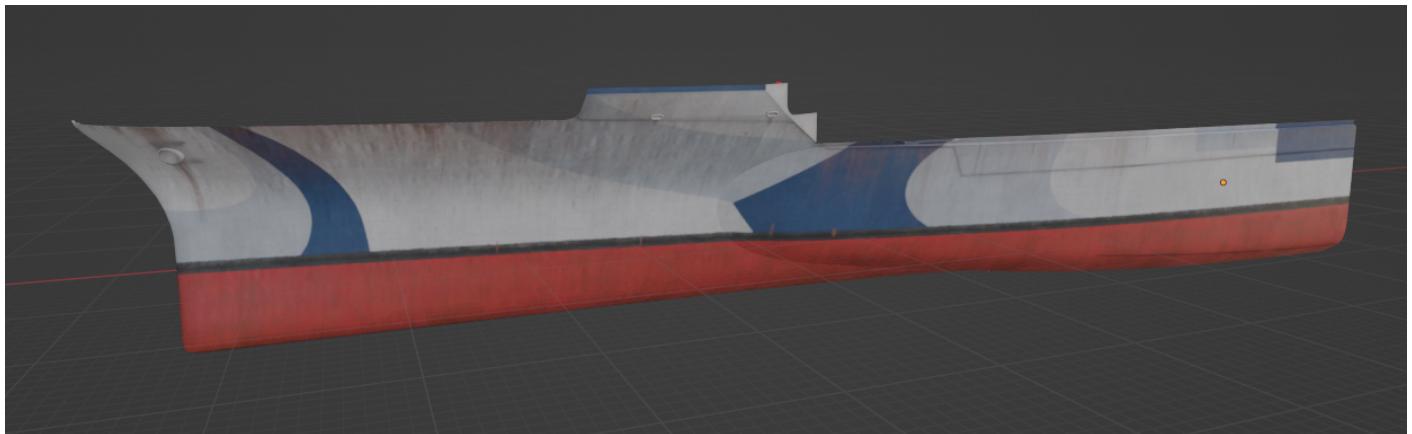
## 1) For an object specify in EDM Export panel

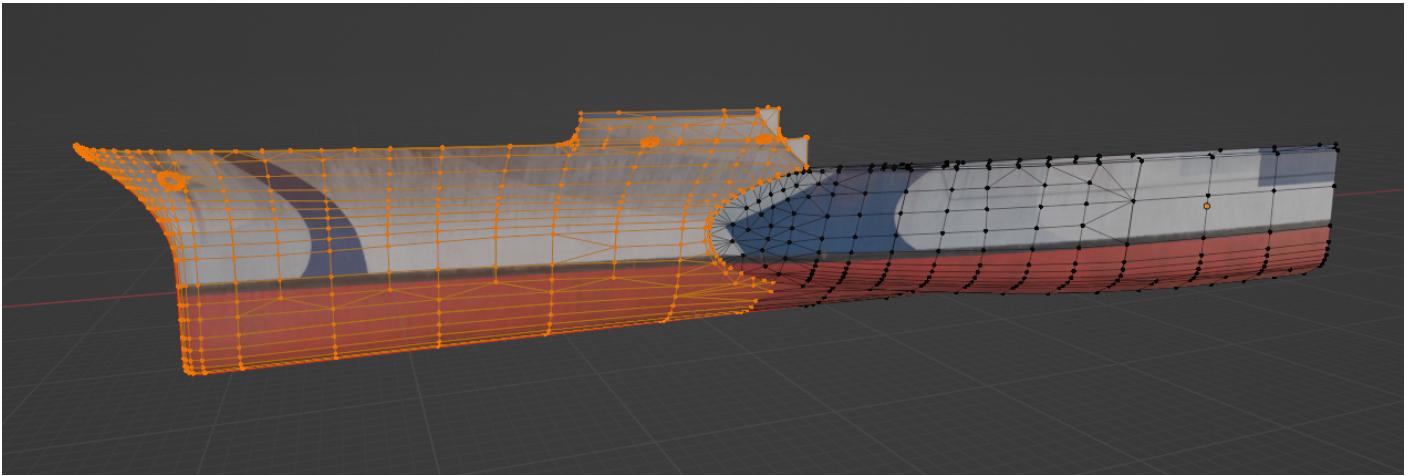


**2) For a part of an object** - for one object you can assign different damage arguments to different parts. They are indicated using vertex groups.

Create vertex groups in the object with the name: DMG\_1, where the number is the damage argument. Assign corresponding geometry parts to these groups. Create several such groups, one for each damage part.

Example: one part of the ship's side belongs to the group DMG\_74, another to DMG\_75. They will “damage” accordingly.





!!! Do not use both options. If vertex groups are used, do not specify Damage arg for the object.

## Deck Material

Opacity Value Argument