KUESA blender®

KUESA™ Studio For Blender

The design to code workflow

KDAB

1. Introduction to KUESA™ Studio For Blender

KUESA™ Studio for Blender makes the unique KUESA™ workflow complete, when you are working with Blender and allows you full access to the range of features provided with KUESA™.

KUESA™ Studio for Blender is a plugin for Blender that consists of

- An extension for the glTF 2.0 exporter to convert Blender scenes to glTF 2.0 files including the accompanying binary buffers and textures, supporting KUESA™ specific features like Iro materials and KUESA™ layers.
- Integration of KUESA™ materials for Eevee in order to provide real-time WYSIWYG during modeling and animation. The designer can check the visual result inside of Blender without exporting or waiting for an engineer build an app beforehand. This allows all team members to stay focused in their area of expertise.
- Additional utilities for Blender that support the KUESA™ workflow.

KUESA™ is a tool designed to boost your 3D design-to-code workflow,

The aim of KUESA™ is to ensure the fastest workflow possible. Since KUESA™ moves the work 'up the pipeline' you can create your scene entirely in Blender and use the tools you are used to - like scripting and other Blender tools, without going the extra mile via a proprietary editor.

If you want to learn more about the KUESA™ workflow or KUESA™ in general, please checkout out https://www.kuesa.com/

Check out the KUESA™ 3D videos : www.kuesa.com/videos

2. Installation Of KUESA™ Studio For Blender

KUESA™ Studio comes with a common installer for all KUESA™ Studio components. There is no specific KUESA™ Studio for Blender installer.

You can choose between two options:

- An online installer that installs KUESA™ Studio using the Qt Maintenance Tool, downloading the packages during installation.
- A standalone offline installer that is downloaded beforehand.

Both will be explained in the following sections.

Once installed, KUESA™ Studio add-ons, materials and utilities become available in Blender on the next Blender startup.

KUESA™ offers provides adds on for multiple 3D authoring tools such as Blender and Maya.

2.1 Offline Installer

After downloading and starting the offline installer from the KDAB customers portal https://customers.kdab.com/ please follow the on-screen instructions.

2.1.1 Choose the components to be installed.

The minimum requirements would be to install the KUESA™ Studio Blender add-on. To inspect the exports, you should also install the KUESA™ Studio Tools which include the KUESA™ Studio glTFInspector.

- Accept the license,
- Confirm the start menu shortcut options

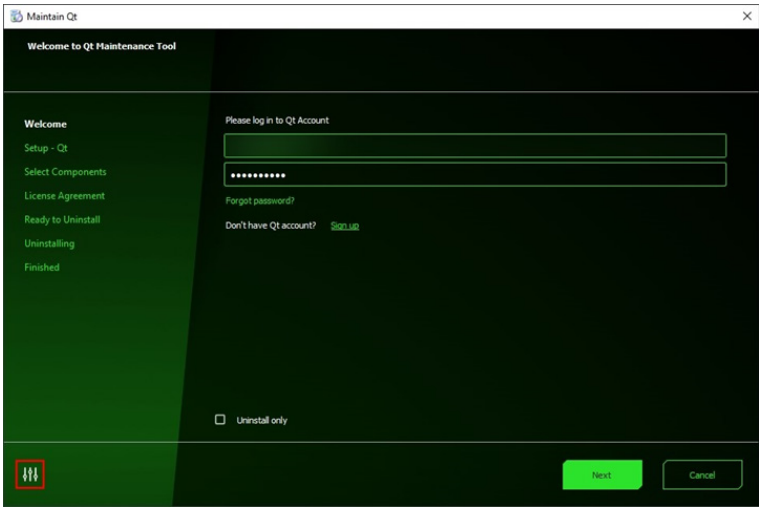
- Browse for your Blender binary.
- Click on ‘Install’ to confirm.

The installer will add the Kuesa add-ons for Blender to your personal Blender add-ons folder that lives in the sub-hierarchy of your home folder.

After the installation is finished, start Blender to check if the add-ons are loaded properly.

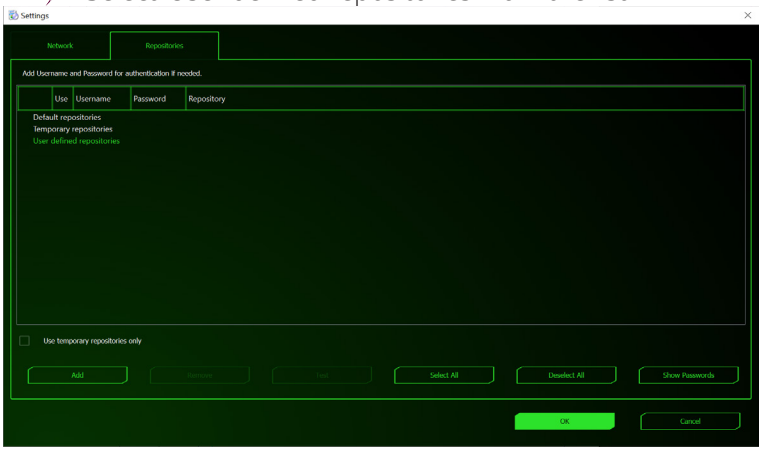
2.2 Qt Online Installer

Run the Qt maintenance tool and click on the settings icon at the bottom left corner.

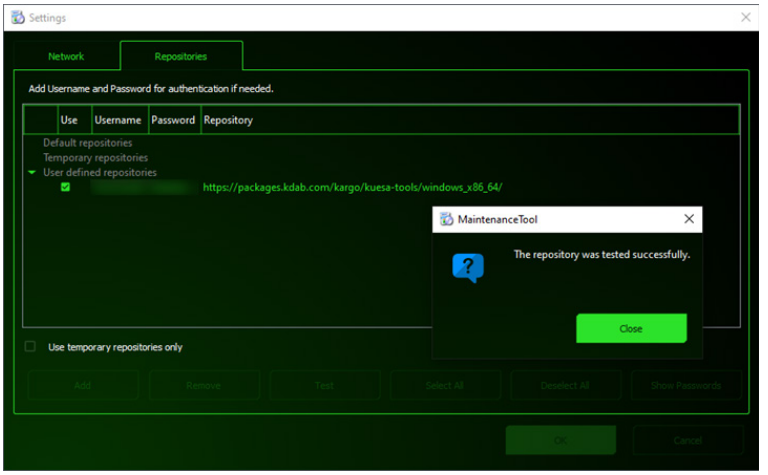


Qt is a cross-platform toolkit for creating graphical user interfaces: www.qt.io

- Open the 'Repositories' tab,
- Select 'User defined repositories' from the list

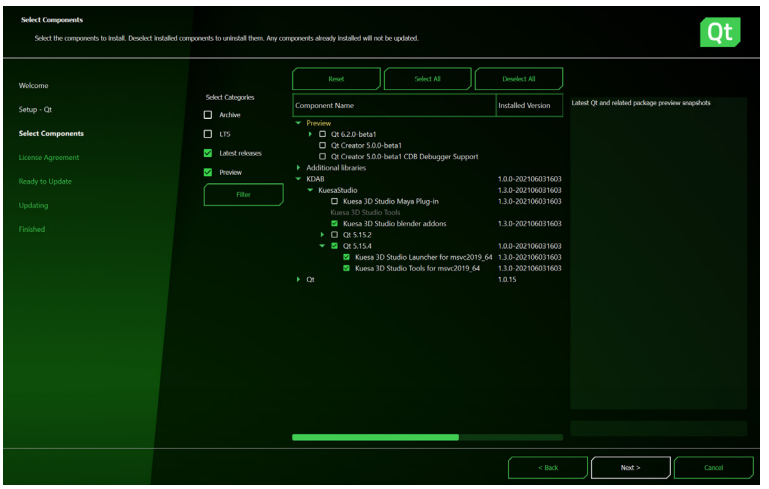


- Click 'Add'
- Add the repository: https://packages.kdab.com/kargo/kuesa-tools/windows_x86_64/
- Click 'Test'
- Enter your user credentials,
- Click on 'OK' to verify that the entered information is correct, and the license is valid.



- Click ‘OK’ to go back to the welcome screen of the Qt maintenance tool,
- Click ‘Next’,
- Select ‘Add or remove components’
- Click ‘Next’.
- Wait for the repositories information to be updated.

In the list under ’KDAB’ choose the components to be installed. For Blender this would be at least the KUESA™ Studio Blender add-ons.



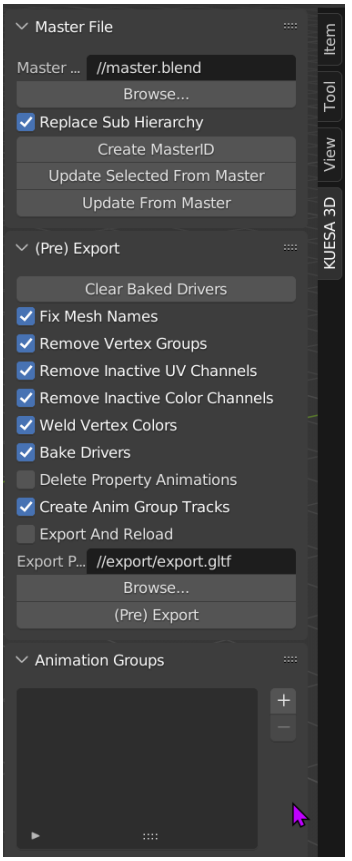
- Click on ‘Next’ to run the installation.

When finished, you will find KUESA™ Studio inside your Qt folder.

3. Using KUESA™ Studio for Blender - Basics

KUESA™ Studio provides several features for use in Blender, including extensions to the glTF export, materials and an application to inspect the exported glTF file.

The Kuesa Panel



3.1 The KUESA™ Panel

The KUESA™ Panel provides convenient access to functionality in Blender such as Master File Tools, and a specialized KUESA™ export.

By default, when Blender starts up, the KUESA™ Panel tab is shown at the bottom of the N-Panel (shown by pressing 'n' in the 3D viewport).

Working with a master file will be covered in a later section,

Working with Master Files.

Before using the KUESA™ specific export, the file must be exported using the normal Blender method of exporting a glTF, making sure that certain options are checked:

3.1.1 Simple Export

This is done using the normal File -> Export -> glTF, however several options should be chosen for the glTF file to work in KUESA™.

Required

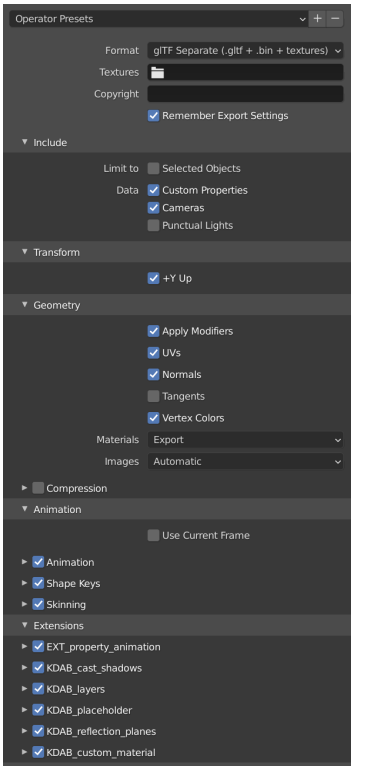
Format: glTF Separate (.glTF + .bin + textures): Ensures that the file-structure is correct for KUESA™

Remember Export settings: Allows the use of the KUESA™ specific export

Include: Camera: Ensures that the camera can be used by KUESA™.

Transform: +Y Up: Converts the default Z up in Blender to Y up used in KUESA™.

Full glTF export options



Optional

Include: Custom Properties: If custom properties are to be exported in the glTF file, this should be checked to export the values to the glTF. See Custom Properties for a full explanation.

The other options in the screenshot are set by default, however it may also be useful to select Geometry: Apply Modifiers.

Once the glTF has been exported for the first time, the KUESA™ panel can be used for further exports if further functionality is required, however the glTF created from this export will be usable in KUESA™.

3.1.2 KUESA™ Export

After the simple export is done, the KUESA™ export can be used as the export settings are stored into the file. This is accessed in the KUESA™ panel as a one-click button. Further functionality is added to the export:

Fix Mesh Names: Clean up mesh names to match their containing object names.

Remove Vertex Groups: Remove vertex groups from meshes.

Remove Inactive UV Channels: Removes all UV channels except the active one to minimize mesh export size.

Remove Inactive Color Channels: Removes all vertex color channels except the active one to minimize mesh export size.

Weld Vertex Colors: Avoid hard color edges by averaging out vertex colors on the same vertex.

Bake Drivers: If drivers have been added to the scene, selecting this option will bake the drivers to a keyframe animation so that the animation will be visible in KUESA™.

Delete Property Animations: Removes property animations.

Create Anim Group Tracks: Prepares the actual animation groups for gltf export.

Export and Reload: Exports the glTF and reloads the file as it was before performing the selected actions.

Export Path: Shows the path where the gltf will be saved relative to the current blender file save location. Use the Browse button to choose the correct location.

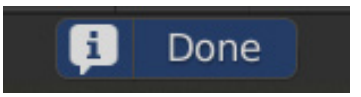
(Pre) Export: Requires saving the file. Pressing this button will perform the actions selected above - and if Export and Reload has been checked, it will export the glTF to the export path selected above - the file will then be reloaded.

Saving the file is required before performing these actions. If an error occurs, this will show as a mouse tool-tip; if it successful 'Done' will be shown at the bottom of the screen.

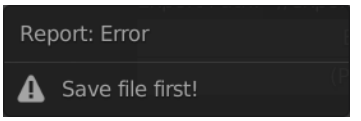
The checks that are done before exporting:

- one material per mesh
- check for clean naming - e.g. are there any objects named .001
- delta transforms check
- re-baking drivers
- check for vertex groups

Successful notification shown at the bottom of the screeen:



Error shown as mouse tooltip:



.

3.2 Materials

Kuesa comes with a set of materials to be used with Blender. This is because offline rendering materials are not the same as real-time materials. You cannot just use your favorite Cycles or Eevee materials and assume they will work properly in your real-time application. Some real-time engine exporters try to convert standard materials to their proprietary engine materials mapping some of the attributes. The downside is that the result in Blender will look different to the result in the target application. In comparison, KUESA™ materials are integrated in Blender Eevee providing WYSIWYG during modeling and animation. We strongly recommend using them instead of standard Blender materials in order to make sure all materials you use are supported and give proper visual feedback in Blender.

Iro materials have been developed specifically for KUESA™ with a specific focus on efficiency in realtime applications.

Currently there are two kinds of materials in KUESA™: PBR (the de facto industry standard, directly supported by glTF) and KUESA™ Iro materials. PBR materials tend to be more realistic providing a different feature set, while Iro materials are easier and faster to set up and more performant since the shader code is less complex. The suggested rule of thumb is:

glTF™ (GL Transmission Format) is a royalty-free specification developed by the Khronos Group. glTF minimizes the size of 3D assets, and the runtime processing needed to use them.

- Use Iro materials whenever possible.
- If you need more realism or HDR environment maps, etc. then use PBR.

You can use both materials at the same time.

All materials described in the section, KUESA™ materials are supported by KUESA™.

3.3 Inspecting Exported Assets

KUESA™ Studio comes with the 'glTFInspector' – a tool that allows you to load your exported glTF file and check out its contents and test if the scene is exported and loaded as expected. The gltfInspector supports all KUESA™ specific features like Iro materials or KUESA™ layers.

Since KUESA™ is using the glTF format, you also can use other glTF viewers in order to inspect your scene. Please keep in mind that

those probably will not support KUESA™ specific features like Iro materials or KUESA™ layers.

The glTfInspector is particularly useful as a bridge between the designer and the developer, providing both with a KUESA™ rendering of the scene.

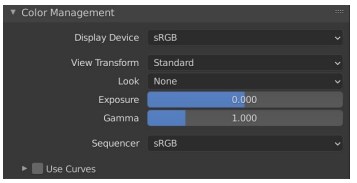
4. KUESA™ Materials

KUESA™ includes special materials developed to be performant. These materials are used the same way as traditional Blender materials however there are a few settings to be used to get the best visual representation.

4.1 Blender Settings For KUESA™ Materials

KUESA™ materials are designed to be used with the Eevee rendering system that is available in Blender 2.8+. Eevee gives a WYSIWYG representation of the final glTF output, as long as the following concepts are followed.

Color Management settings:



Color management:

Should be set to Standard (not filmic, which is the default). This is found under the Color Management section in the Render options tab. Using the standard View Transform will give a more accurate color representation.

Using colors from other applications:

Applications such as Adobe Photoshop typically use the sRGB colorspace, while Blender typically uses a linear workflow. The sRGB colorspace has a gamma curve of 2.2 applied to it to better display colors on a screen, while the linear colorspace does not have the gamma curve applied.

When using sRGB values from other applications, ensure that you copy the HEX value to the corresponding color in Blender, as the RGB/HSV values use the linear workflow and will not match.

4.2 Iro Materials – Concept

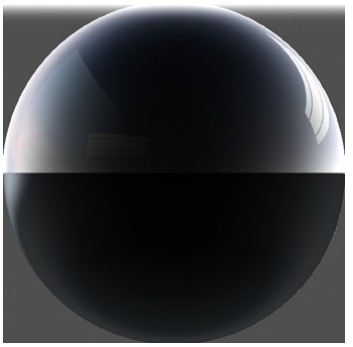
Iro materials are a set of KUESA™ specific materials that

- allow a rapid, artistic, and intuitive workflow.
- are highly performance optimized.

All Iro materials are fully integrated into Blender and therefore provide WYSIWYG during modeling and animating. The designer can check the visual result inside of Blender without exporting or having an engineer build an app.

Since Iro materials are easy to set up and highly performance optimized we suggest using them whenever possible and switch to PBR only when essential.

An Artistic Workflow of Iro Materials



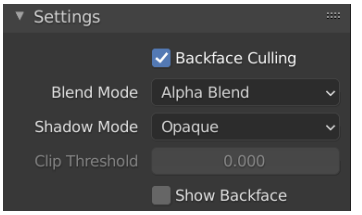
Iro materials are designed for artists and meant to be simple to use. Instead of creating an environment map and tweaking the material parameters so you get the desired result, with Iro materials you are following a more direct approach. Some Iro materials support so called 'mat caps' for reflections. We will refer to this by the term 'SEM mode' (Spherical Environment Map). Mat caps are basically 2D images of spheres as you can see to the side.

When you apply an Iro 2 Diffuse material using this texture to a sphere in Blender, the sphere will look exactly the same as the 2D image. This means that a sphere will look the same from all directions. For objects like a car for example this approach is fine as you can see in the screen shot because the geometry is complex enough, so the reflections appear differently from various viewing angles. However, if you need 'correct' reflections - computing real reflection vectors per point – you can use a different material that uses equi-rectangular texture maps – which we will refer to by the term 'ERM mode' (Equi Rectangular Map).

The car on the right uses the left sphere image for its reflections. This makes it easy to rapidly match mock-ups from your customer and achieve the look you need. You can use rendered spheres, real photos or just paint the sphere as needed in Adobe Photoshop (this will be called 'paint the result').

Transparent Materials

The following settings should be used for transparent materials:



Objects using transparent materials such as alpha blending, additive or multiplicative materials are drawn in order of depth in the application, and therefore get sorted every frame in KUESA™. For performance reasons there will not be any face sorting but only sorting on object level. To avoid rendering artifacts you therefore should try to avoid self-intersecting geometry and rendering of back faces.

To set a material as transparent, the material should have the blend mode set to Alpha Blend or the material will be rendered incorrectly.

Special Blend Modes

To avoid rendering artifacts you should try to avoid self-intersecting geometry and rendering of back faces.

Since the 3D viewport is rendered as a 2D rectangle in the target application, alpha blending is used. This means that materials that are additive or multiplicative (i.e. Iro Glass, Iro Matte Mult and Iro Matte Add) should have a solid background behind them. Either avoid those materials on transparent spots or create a background inside your 3D scene using background materials for example.

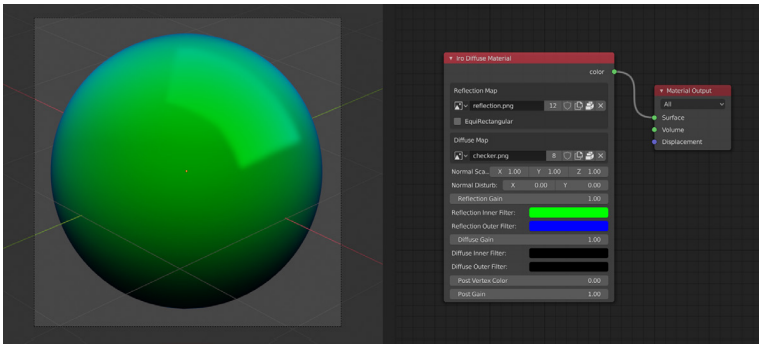
Material Management

Please just use one material per object. Using more than one material per object may give unexpected results.

Filter Colors

Filter colors are a concept specific to Iro materials. Usually when you want to set a color for something, you would need to define two colors instead: an inner filter color and an outer filter color. Those two colors are blended by a fresnel approximation so you always can define a color or intensity for the ‘inner’ part of the object (the part of the surface that is pointing towards you) and a

color or intensity for the ‘outer’ part of the object (the part of the surface that is pointing sidewards). This enables you to easily fake fresnel and other interesting effects while adding a significant amount of realism.



4.4 Iro Materials – An Overview

The following list gives an overview of all Iro materials and what they are for. Since KUESA™ 1.3 the Iro materials are in their second iteration, some of them might have a ‘2’ in their name. This is just for maintaining backward compatibility for the old Iro materials inside the engine.

Standard Iro Materials

Standard Iro materials are the most common material you can use for almost all kinds of things like car paint, glass, leather, etc. Each of these three materials features a reflective component. Please refer to ‘Iro Materials – Concept’ to learn more about it. Standard Iro materials are the most complex materials among the Iro materials, although they are still highly performance optimized and simple to use.

Iro 2 Diffuse: The most commonly used Iro material. This is what you want to use for opaque objects.

Iro 2 Alpha: Same as above but with transparency (alpha blending).

Iro 2 Glass: Fakes a glass effect by doing two passes as described below.

Iro Matte Materials

Iro matte materials are even simpler (and cheaper to compute) than the previously explained standard Iro materials. They are suitable for simple tasks such as background mattes, simple textured objects, shadow planes, or simple light effects, etc. They do not have any reflections.

Iro Matte Opaque: Simple opaque textured material.

Iro 2 Matte Alpha: Simple transparent (alpha blended) textured material.

Iro Matte Add: Simple additive textured material. (same as the linear dodge (add) layer mode in Photoshop calculated in linear space)

Iro Matte Mult: Simple multiplicative textured material. (same as the multiplicative layer mode in Photoshop calculated in linear space).

Iro Matte Background Materials

Iro background materials are special matte materials for defining the background of a 3D scene either with a plane or a sky geometry object. Unlike other Iro materials, background materials are modifying the geometry of the mesh they are applied to.

Iro Skybox: Can be applied to an object of any shape (does not have to be a box). The geometry is translated to the viewer's position.

Iro Matte Background: Can be applied to an object that will be transformed to screen space for background planes.

Iro 2 Planar Reflection: Can be applied to a mesh to create a reflection of the scene. Note that the mesh with this material should be in a separate collection to the rest of the scene to be able to generate the reflections correctly.

Older Materials

Some materials will not be covered as they are only for backwards compatibility. Newer materials should be used as the properties of these are contained within the new materials

Iro Diffuse Alpha

Iro Diffuse Hemi

Iro Diffuse

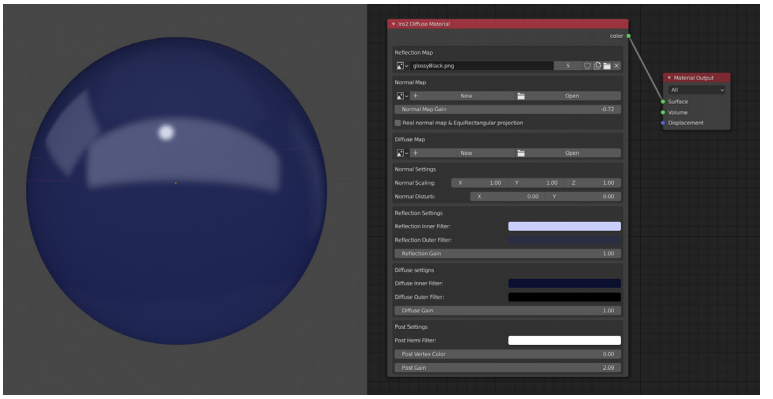
Iro Glass Add

Iro Matte Alpha

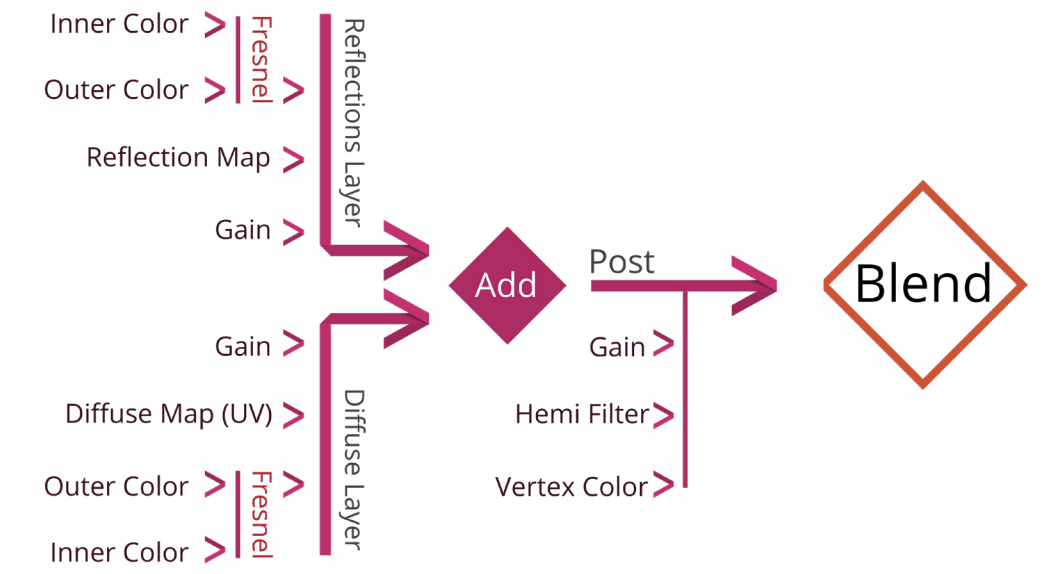
4.5 Iro Materials – Types in Detail

4.5.1 Iro 2 Diffuse

The Iro 2 Diffuse material is one of the most commonly used materials of KUESA™. You can use it for the majority of your tasks like car paint, leather, etc. (as long as the objects are opaque). It has two layers (one diffuse and one reflective) which are added together and then modified in the 'post' step and finally drawn to the scene.



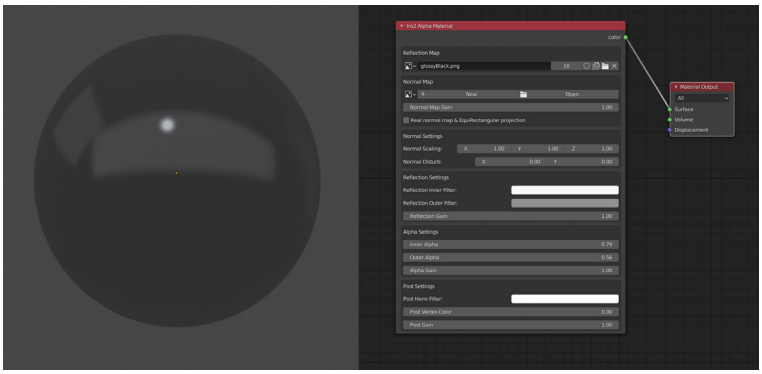
As you can see in the flow chart, the components of each layer (filter colors, map, and gain) are multiplied.



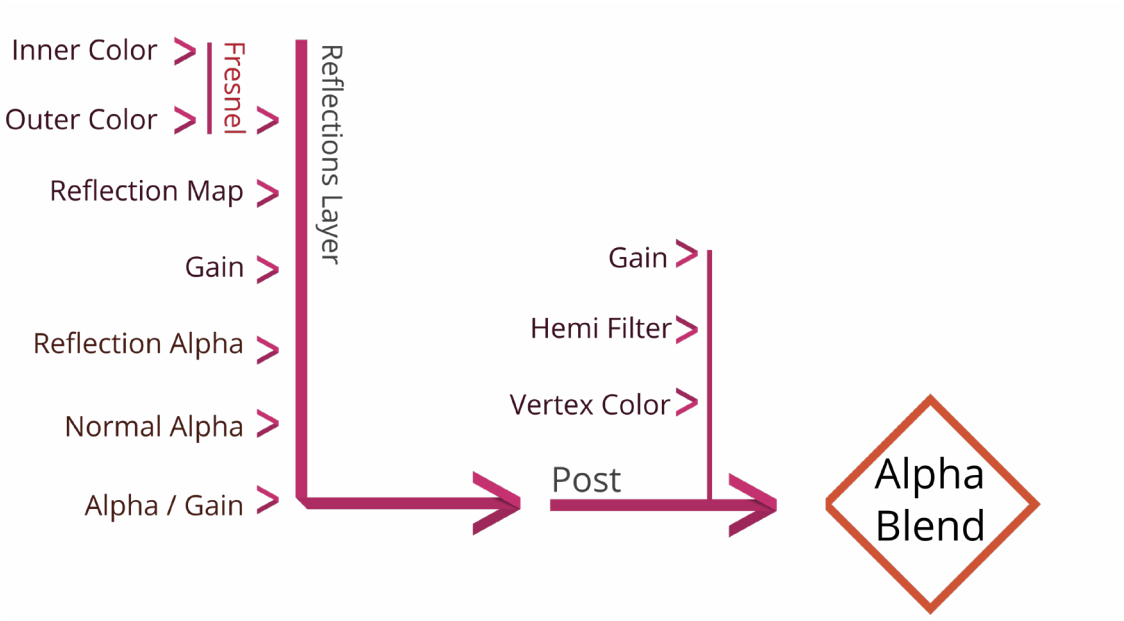
4.5.2 Iro 2 Alpha

The Iro 2 Alpha material provides the same type of reflections as the Iro 2 Diffuse material but adds an alpha channel to control the opacity. The alpha value is composed of several components which are all multiplied together to get the final alpha value per pixel. With those components you have several approaches to define the opacity as listed below

Alpha channel of the reflection map:	Define opacity by reflections
Alpha channel of the normal map:	Define opacity by UV mapping
Inner / Outer alpha value:	Define opacity by fresnel
Alpha gain:	Define opacity globally

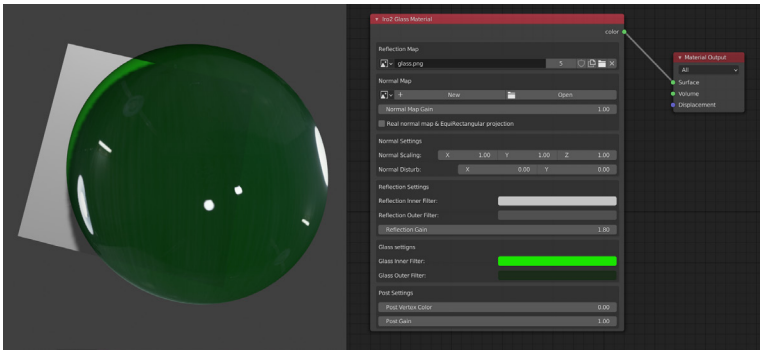


Use the Iro 2 Alpha material if you want to create transparent objects.

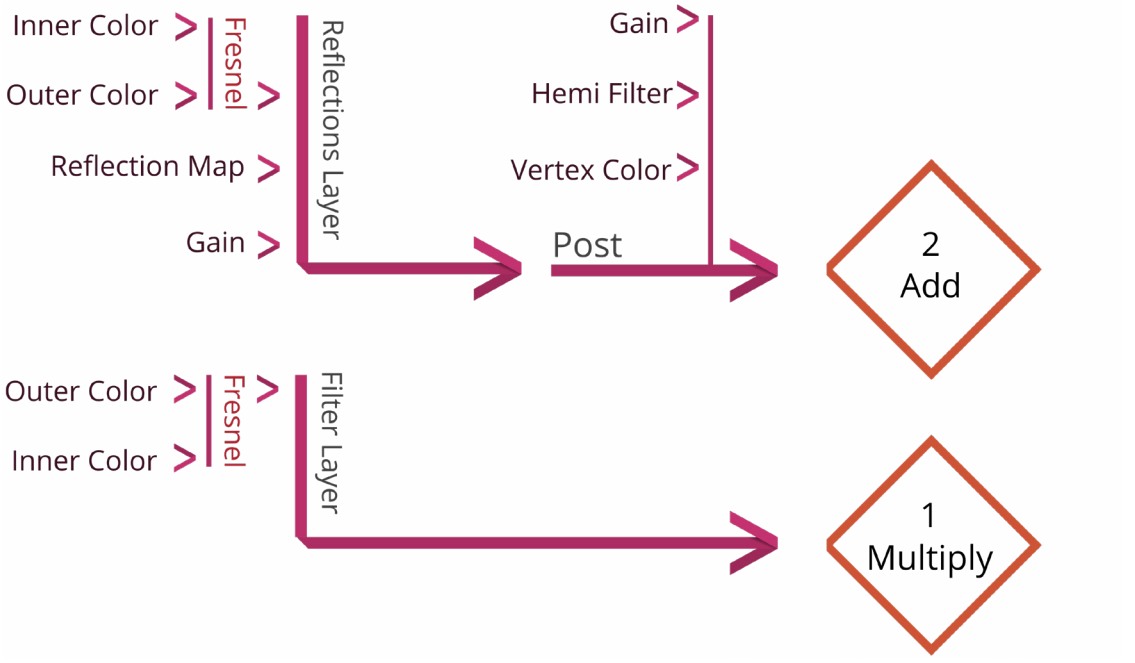


4.5.3 Iro 2 Glass

Since it is sometimes hard to get convincing results when trying to create glass objects with the alpha material, KUESA™ comes with the Iro 2 Glass material.

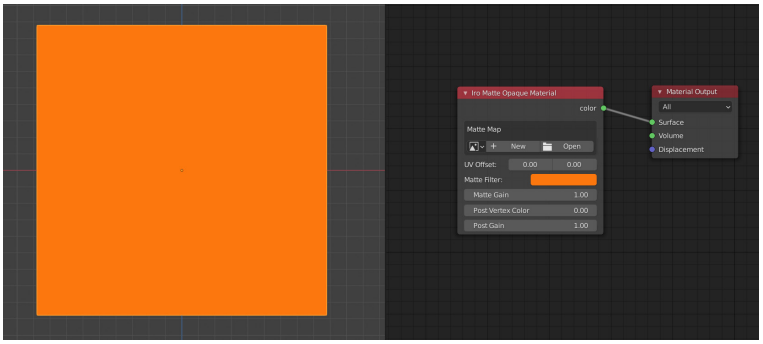


The material mimics the behavior of glass using two passes: The first pass is multiplicative and tints the background down by the glass color. This is similar to colored glass where only a part of the light / the color spectrum comes through. The second pass adds reflections.



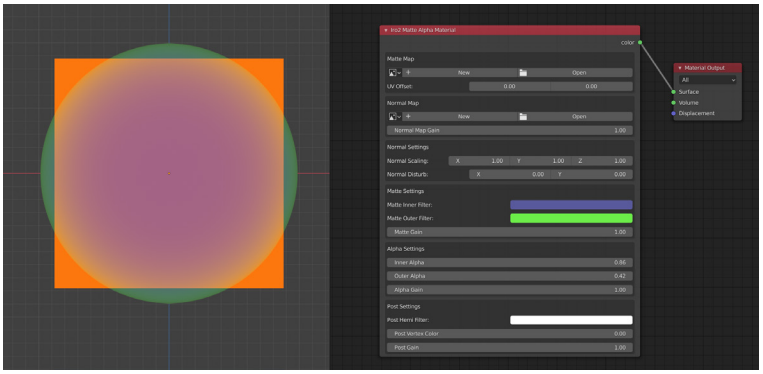
4.5.4 Iro Matte Opaque

Iro Matte Opaque is the least complex (and therefore computationally cheap) material of the Iro materials and features a UV mapped, textured, opaque material.

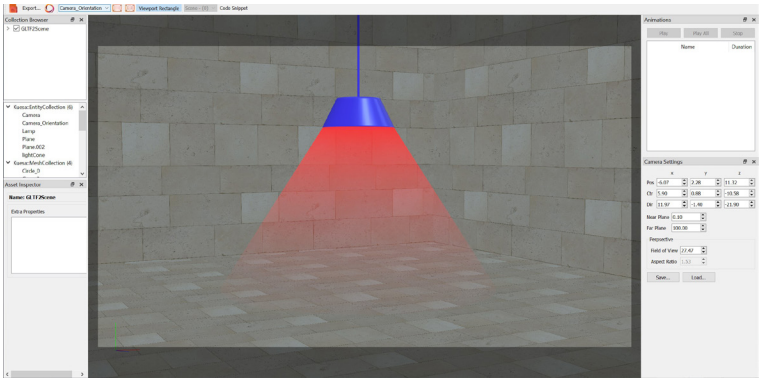


4.5.5 Iro 2 Matte Alpha

Simple transparent material using alpha blending

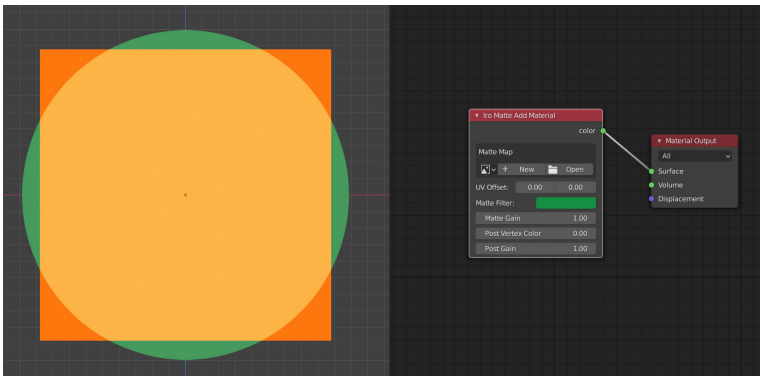


This can be used for example to show a light cone in a scene, here using a gradient texture fading to transparent.

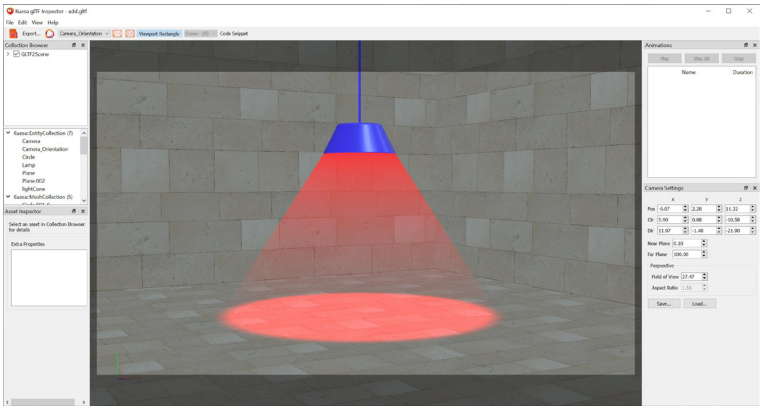


4.5.6 Iro Matte Add

Simple material using additive blending (implicitly transparent).

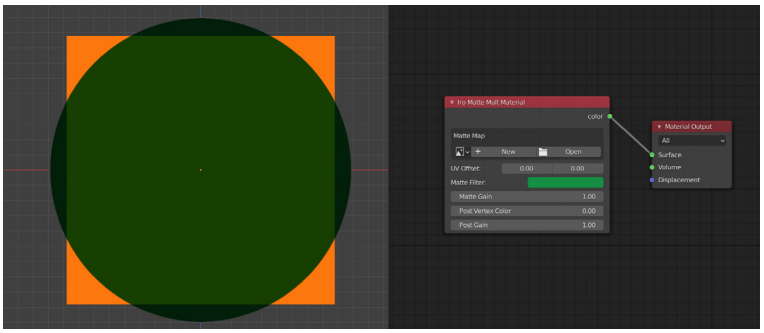


Can be used to add a lightening effect, for example the circle of a spotlight on the ground:

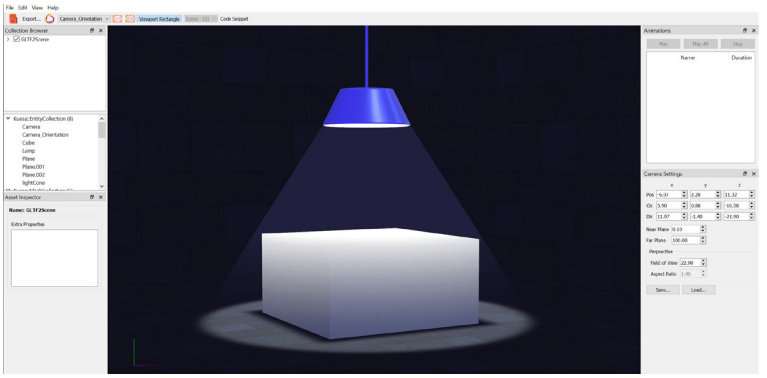


4.5.7 Iro Matte Mult

Simple material using multiplicative blending (implicitly transparent).

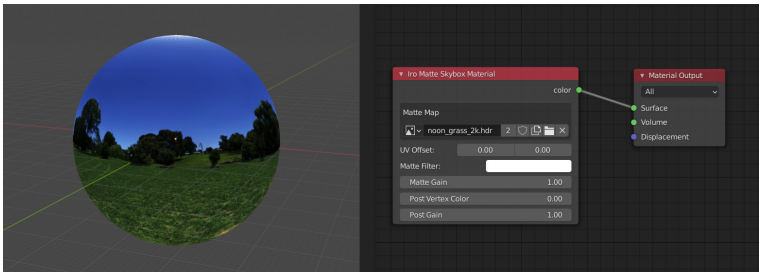


Can be used to add shadows to a scene:

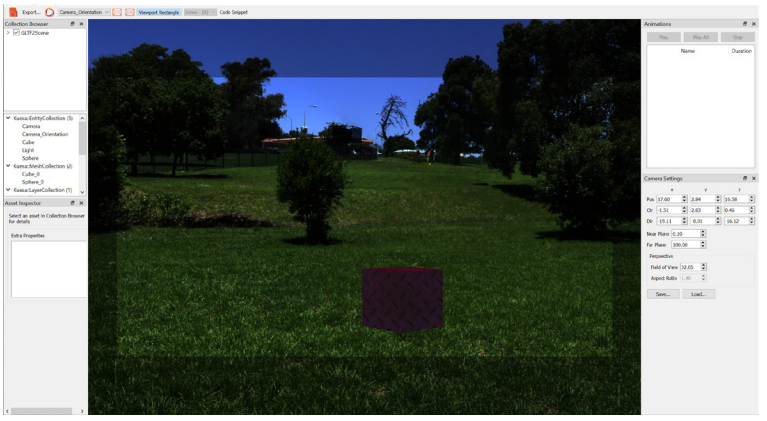


4.5.8 Iro Matte Skybox

The Iro Matte Skybox material provides a fast and flexible way of creating backgrounds for your 3D scene. Objects that have the Skybox material attached are translated to the eye position and rendered first. Even though the material is called 'Skybox' you can use any geometry, e. g. cylinders or spheres.



When viewed in the glTFInspector, the scene is seen from inside as the camera and purple box are inside the sphere:



4.5.9 Iro Matte Background

The Iro Matte Background material provides a fast and flexible way of creating backgrounds for your 3D scene. Objects that have the Iro matte background material attached are using the object space coordinates of the vertices in screen space - i.e. if you create a standard plane, this will cover the entire screen.

4.5.10 Iro 2 Planar Reflection

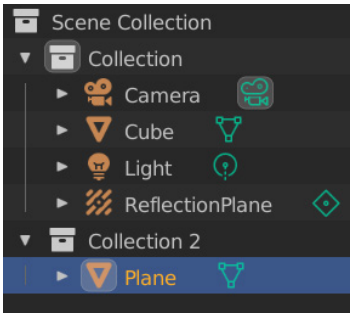
The Iro 2 Planar Reflection material is used to create reflective planes, for example a shiny floor in a showroom.

To create a scene with reflections:

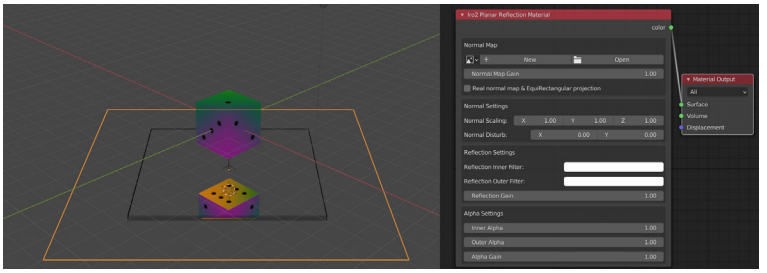
- Create the scene to be reflected.
- Add a Reflection pane (Add -> Light Probes -> Reflection plane). This step is optional, however, a preview in Blender is only possible if a Reflection plane is added to the scene
- Create a new collection and add the mesh that will show the reflection into this new collection. This allows the reflections to be generated correctly.

Note that the screenshot from Blender shows a reflection plane smaller than the mesh with the reflective material - this shows that the preview in Blender is limited to the Reflection plane, while the output in the glTFInspector shows the full reflection from the selected mesh.

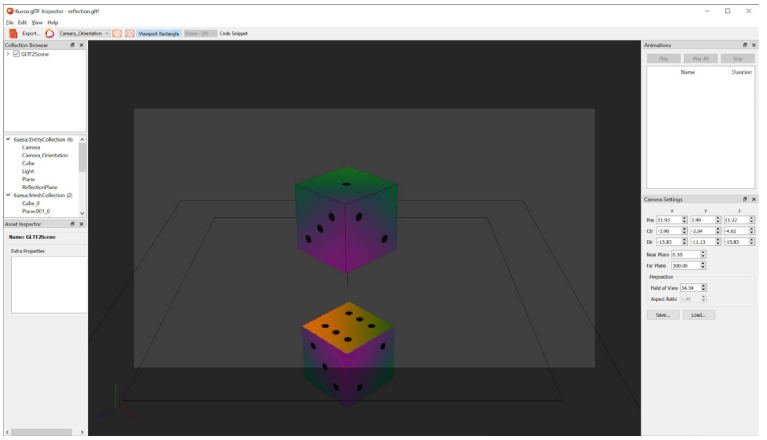
Example scene setup for the provided screenshot:



.



Preview in glTFInspector:



4.6 Iro Materials – Attributes

All of the KUESA™ materials have various attributes, but not every material has every attribute. The attributes are explained here in detail; the section after matches which materials have which attributes.

The usage of SEM mode vs ERM mode depends on the model's contextual location needs. Using the SEM mode is less computationally expensive than using ERM mode and the results may be equally acceptable.

4.6.1 Sub Types: Sem / Equi Rect

Reflections are achieved by using special 2D reflection maps (spherical environment maps or mat caps) (in SEM mode) that keep computation and tweaking costs low and allows for an intuitive and artistic workflow. For further information, see the section Iro Materials – Concept. However, if you need accurate reflections you can switch to the 'Equi Rect' version (ERM mode) and feed the texture slot with an equi-rectangular environment map.

For some KUESA™ Materials, there are two subtypes - the SEM mode and the ERM mode material. The type used is controlled by the checkbox on the material under the reflection map:

4.6.2 Normal Attributes

The first parameters (Normal Map, Normal Map Gain, Normal Scaling, Normal Disturb) control the normals. This affects the fake fresnel profile and thus the gradient of the inner and outer filter colors as well as the appearance of the reflections. Note that you will only see something happening here when filters are set to different color for inner and outer or when using a reflection map.

> **Normal Map / Normal Map Gain:** Normal Map Gain controls how much the normal map disturbs the normals. In SEM mode you can set this to a negative value in order to reverse the normal map. Normal maps are in tangent space so you need the standard blue/ purple tinted normal maps.

Note: In SEM mode we are dealing with fake reflections and thus fake normal maps. For most use cases where you apply normal maps to mimic material structures like leather it should be perfectly fine. However, it is not possible to replace baked geometry with fake normal maps. You would need to switch to ERM mode for that.

> **Normal Scaling:** controls how much the normals are stretched or squeezed in the X and Y direction respectively. The Z value affects both directions at the same time.

> **Normal Disturb:** controls how much the normals are pushed and pulled in the X and Y direction, respectively. Use this for example to easily move the horizon in the reflections up and down without changing the texture.

4.6.3 Reflection Attributes

The next parameters (Reflection Map, Reflection Inner / Outer Filter, Reflection Gain) control the look of the reflections.

Reflection Map: Holds the reflection map - a mat cap in SEM mode or an equi-rectangular environment map in ERM mode. UV mapping will have no effect as the reflections are determined by the angle of the surface to the camera.

Reflection Inner / Outer Filter: Darkens the reflections by the defined colors. Those two colors are blended by a fresnel approximation so you always can define a color or intensity for the 'inner' part of the object (the part of the surface that is pointing towards you and a color or intensity for the 'outer' part of the object (the part of the surface that is pointing sidewards). This enables you to easily fake fresnel and other interesting effects and add a significant amount of realism. For realistic materials (e. g. car paint or plastic) you might want to start with a dark inner color to mimic the fresnel effect.

Note: In SEM mode those colors are for convenience. You also could achieve the same results by changing the texture itself. For ERM mode this would be the only way to achieve this effect though.

Reflection Inner / Outer Alpha: These two alpha values are blended by a fresnel approximation so you can define an alpha value for the 'inner' part of the object (the part of the surface that is pointing towards you) and an alpha value for the 'outer' part of the object (the part of the surface that is pointing outwards).

Reflection Gain: Overall gain for the reflections. Use this to tweak the ratio between reflections and the diffuse layer. The higher this value the brighter the reflections will appear.

Note: For the glass material the reflections are added in a separate pass.

4.6.4 Diffuse Attributes

The next parameters (Diffuse Map, Diffuse Inner / Outer Filter, Reflection Gain) control the look of the diffuse layer.

Diffuse Map: Holds the diffuse map (UV mapped).

Diffuse Inner / Outer Filter: Darkens the diffuse part by the defined colors. Those two colors are blended by a fresnel approximation so you always can define a color or intensity for the 'inner' part of the object (the part of the surface that is pointing towards you and a color or intensity for the 'outer' part of the object (the part of the surface that is pointing sidewards).

Diffuse Gain: Overall gain for the diffuse part. Use this to tweak the ratio between reflections and the diffuse layer. The higher this value the brighter the diffuse layer will appear.

4.6.5 Matte Attributes

Matte Map: UV mapped texture. (When not set, the default color will be white)

UV Offset: Global (object) offset of the UV coordinates. Useful for animating texture coordinates.

Matte Filter: Filter color that tints down the texture multiplicatively or sets the color when the texture is left blank.

Matte Inner/Outer Filter: These two colors are blended by a fresnel approximation so you always can define a color or intensity for the 'inner' part of the object (the part of the surface that is pointing towards you and a color or intensity for the 'outer' part of the object (the part of the surface that is pointing sidewards).

Matte Gain: Use this to easily tweak brightness.

4.6.6 Alpha Attributes

The alpha parameters control the alpha by fresnel and global alpha:

Inner / Outer Alpha: Those two values are blended by the same fresnel profile as the reflections and control the opacity.

Alpha Gain: Overall gain for alpha.

4.6.7 Glass Attributes

The resulting glass filter color tints down the background multiplicatively and mimics the behavior of colored glass where only a part of the light comes through.

Glass Inner / Outer Filter: Those two colors are blended by a fresnel approximation so you always can define a color for the 'inner' part of the object (the part of the surface that is pointing towards you and a color or intensity for the 'outer' part of the object (the part of the surface that is pointing sidewards)

4.6.8 Post Attributes

Use the post parameters to tweak the overall look of the final result in the last step.

Post Hemi Filter: Tints the parts of the surface that are faced downwards (in world space) by the specified color. This acts like a negative light from below. It can be useful to quickly add a bit of realism without the use of any ambient occlusion or baked textures.

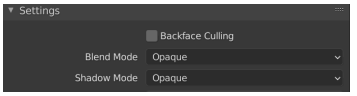
Post Vertex Color: When set to zero, the mesh will not be affected. When set to one, the defined vertex colors are multiplied with the final result. You can blend using any number between zero and one. Note: If you did not paint any vertex colors on your mesh, the default vertex color is used which is black.

Post Gain: Use this to easily tweak the overall brightness. Using Post Hemi Filter and Post Vertex Color will darken the look of the material, changing the value of Post Gain will allow you to compensate for this effect.

4.6.9 Miscellaneous

Double Sided Materials: In Blender this is set by ensuring that the backface culling is not selected. This should be only used for non-transparent materials. It is also best practice to not select this unless specifically required as it has a performance cost and may cause rendering artifacts with transparent materials.

Settings in Blender



4.7 Iro Materials – Type to Attributes

	SEM Mode	ERM Mode	Normal Map	Normal Map Gain	Normal Scaling	Normal Disturb	Reflection Map	Reflection Inner/Outer Filter	Reflection Gain	Diffuse Map	Diffuse Inner/Outer Filter	Diffuse Gain	Matte Map	UV Offset	Matte Filter	Matte Inner/Outer Filter	Matte Gain	Inner/Outer Alpha	Alpha Gain	Glass Inner/Outer Filter	Post Hemi Filter	Post Vertex Filter	Post Gain	Double-Sided
Iro 2 Diffuse	x	x	x	x	x	x	x	x	x	x	x	x								x	x	x	x	
Iro 2 Alpha	x	x	x	x	x	x	x	x	x								x	x		x	x	x	x	
Iro 2 Glass	x	x	x	x	x	x	x	x	x										x		x	x	x	
Iro Matte Opaque												x	x	x		x					x	x	x	
Iro 2 Matte Alpha												x	x		x	x	x	x		x	x	x	x	
Iro Matte Mult												x	x	x		x					x	x	x	
Iro Matte Add												x	x	x		x					x	x	x	
Iro Matte Skybox												x	x	x		x					x	x	x	
Iro Matte Background												x	x	x		x					x	x	x	
Iro 2 Planar Reflection	x	x	x	x	x	x		x	x								x	x						

4.8 PBR Materials (In Progress for Blender)

PBR materials are the de facto industry standard, directly supported by glTF and enable a high degree of realism. Since the shader code of PBR materials is rather complex, we suggest only using them when essential.

The phrase "Physically Based Rendering" was more widely popularized by Matt Pharr, Greg Humphreys, and Pat Hanrahan in 2014.

The Blender implementation of PBR materials is in progress. Currently the standard shader to create PBR is the "Principled Shader". Since this is a standard Blender material and not a direct port of the KUESA PBR implementation, results may look differently in the application.

We are currently working on porting our KUESA PBR material to Blender Eevee. For now please refer to the glTF 2.0 exporter manual on how to setup PBR materials in Blender:

https://docs.blender.org/manual/en/2.92/addons/import_export/scene_gltf2.html

5. Using KUESA™ Studio for Blender – More Features

5.1 KUESA™ Layers

You can organize your nodes in KUESA™ layers to treat them accordingly later in your target application using the framegraph.

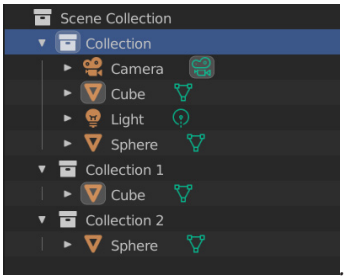
Example use cases:

- defining the render order in groups
- hiding groups of objects
- treating groups of objects differently on rendering

For example, using two cameras in a scene and separating the contents into two layers would allow the cameras to show different objects while keeping the relative positions of the objects in the scene - e.g. you could show the internal workings of a machine on one camera with the external meshes being shown by the other camera.

These correspond to Collections in Blender, which can be created by using the shortcut 'm'. The image to the right shows a cube and a sphere linked to separate collections in Blender

Blender Scene hierarchy

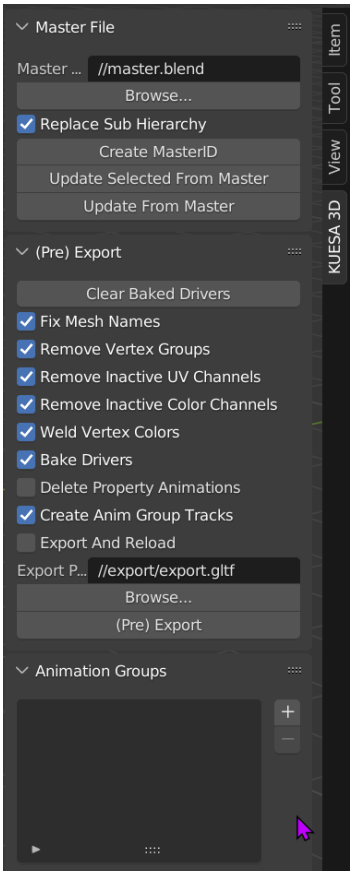


5.2 Grouping Animations

Animation groups is a powerful tool that enables you to define how animations will be exported to glTF in a flexible way. In the glTF universe an animation consist of one or more objects and their according animation channels (attributes) plus a specified time range. In other words, with animation groups you can define exactly which objects belong together for which range in time.

To create a new animation group use the "+"-button in the "Animation Groups" section of the KUESA™ panel. When the group appears in the list you can rename it there. This will be the corresponding animation name in the glTF file.

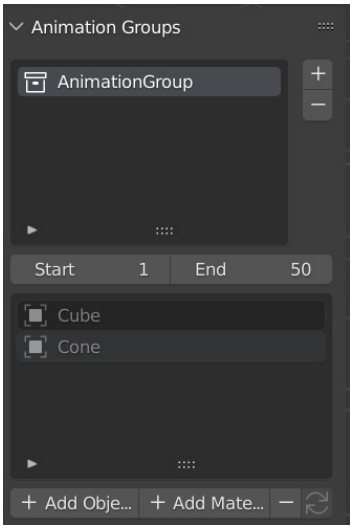
Animation Group panel at the bottom of the Kuesa tab:



Click on an animation group in the list in order to edit its contents. After selecting it, you can change its time linge range and add objects and materials to it. Objects and materials can belong to multiple animation groups and animation groups can overlap.

When the gltf is exported, the 'Create Anim Group Tracks' option on the Kuesa panel should be selected.

An animation group that contains a Cube and a Cone.



5.3 Clean Up

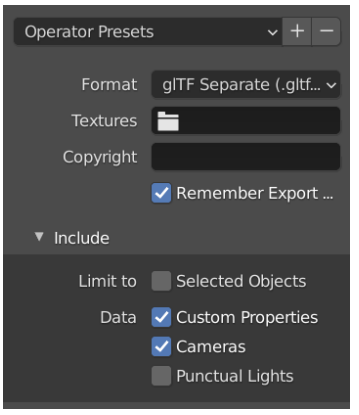
Before exporting from Blender ensure that the meshes have been cleaned up:

- Apply rotations to every object that is not animated
- Ensure that the scale of every mesh is 1 in each direction
- Remove any unapplied delta rotations and transforms

If these have not been applied to the meshes, there may be unexpected positioning and/or sizing in the gltf. If any of the shapes are to be animated, it is good practice to ensure that the start of the animation is at point where these transformations have been applied.

It is also good practice to keep the blender file clean and remove unused items such as images, actions and materials. This will help reduce the file size of the gltf and potentially improve performance.

When exporting, ensure that the Custom Properties are included on export:



5.4 Custom Properties

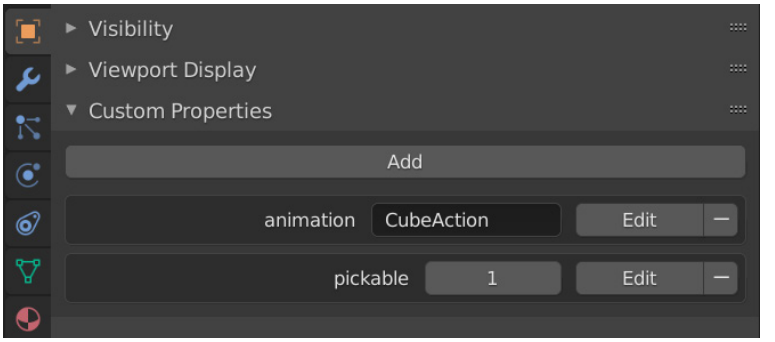
Custom properties are a way of adding additional data to the glTf file. This is potentially useful when embedding the 3D scene into code as it allows the designer to give more information about how the scene is to be used.

For example, it is possible to create a custom property 'animation' that would then contain a comma separated list of the animations that should be run together when a mesh is to be moved.

.

As the custom property is flexible in its use, the information that could be conveyed is too complex to provide a complete list of ways this could be used; however any text or value passed in the property can be read by the code using the glTf. Numerical values can also be animated which will be the available in KUESA™.

Example Custom Properties on an object:

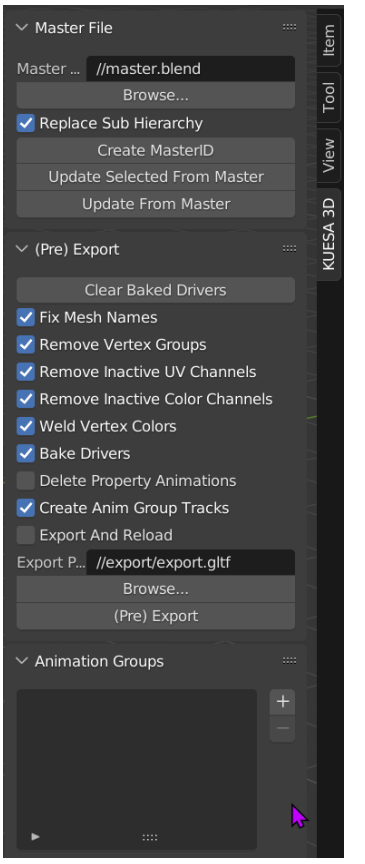


5.5 Working with Master Files

The KUESA™ Panel also adds functionality to work with a master file with a Blender scene. For example, the master file could contain objects that get reused in multiple child files; the tools provided would allow the child files to be easily updated with updated data from the master file.

The child file must have corresponding MasterIDs on the objects that match objects in the master file..

The Kuesa Panel



Master File: The file from which changes will be taken from - this is set using the 'Browse' button.

Replace Sub Hierarchy: When selected, any objects below the object with the MasterID will be updated. If it is not selected, only the object with the MasterID will be updated.

Create MasterID: Creates a property on the selected mesh or empty with an ID. This is required so that the tool knows which objects are to be updated. This button is here for convenience as it is also possible to create the property manually.

Update Selected From Master: Only updated the selected object, or the selected object and its sub-hierarchy.

Update From Master: update all objects in the scene with data from the master file.

To skip an object from the update, a property called 'MasterSkip' should be given to the object or material.

5.6 Export Scripts

One of the biggest speed gains of KUESA™ Studio is its direct workflow. 3D assets are exported directly to glTF files without the need for a proprietary scene editor (that would require an extra import and export step). You can use the glTF exports from Blender in your application, directly out of the box. This way you can easily make the KUESA™ Studio exporter for Blender part of your automated asset pipeline.

Let's say we have a couple of child scenes that depend on a master scene. Now we change the master model and want to re-export all the child scenes with an export script. Please find an example below (the commands used require KUESA™ Studio to be installed).

CLI

```
cd <project dir>
blender child-scene-1.blend --background --python exportscript.py
blender child-scene-2.blend --background --python exportscript.py
blender child-scene-3.blend --background --python exportscript.py
# Some post processing if needed...
```

Python - exportscript.py

```
import bpy
bpy.ops.kuesa.update_from_master_file_simple()
bpy.ops.kuesa.simple_export()
```

6. What Gets Exported?

To the glTF file (JSON)

- The scene hierarchy with its nodes and node transformations
- Cameras
- Materials listed in the materials section.
- KUESA™ layers
- Animation groups
- References to other objects, binary buffers, and texture files
- Custom string and number attributes

To binary buffers

- Meshes
- Animations
    - Animated node transformations
    - Animated numerical attributes of materials
    - Animated numerical custom attributes

To image files

- Textures used in the scene are copied.

7. Optimization Tips

Minimize the Number Of Materials

Performing 'draw calls' or 'state changes' are the most expensive things to do in Open-GL. But at the same time this is what is needed whenever your app does something like changing a texture/material/camera in order to draw a bunch of objects with those settings. While you have no influence on some things like changing the camera, you should pay special attention while creating materials in your 3D scene in Blender. Kuesa tries to minimize shaders for similar materials. However usually one draw call or state change will be performed per material.

So, use materials thriftily - reuse them whenever possible, and remember to use one material per object.

Minimize Shader Complexity

Always use the simplest material that is 'acceptable'. This is a list of KUESA™ materials from simple to complex:

1. Iro matte materials
2. Iro Diffuse and Alpha materials (SEM is simpler than ER)
3. Iro Glass (SEM is simpler than ER)
4. PBR

Pay special attention when using custom shaders. Avoid throwing in all kinds of fancy features. Always keep an eye on performance.

Clean Up

Clean up the objects - ensure that rotations are applied, the scale is normalized, and there are no delta transforms. Check out the section Clean Up.

Minimize Number of Entities

If you import a car or build a complex object you might end up easily with a couple of hundred objects. Managing those in real-time (which will happen each frame) can be rather expensive.

So, try to keep the number of entities to the minimum. Try to merge meshes with the same material (as long as you do not need to animate them separately). But do not trade-off tidiness just to save a couple of empty nodes.

Minimize Texture Size

You are limited in resources. Sometimes by a lot. There is no rule of thumb on how large textures can be. You just need to test this with your target hardware.

Try out smaller textures and compare. Smaller textures result in less data to load and therefore less memory to transfer between the CPU and the GPU.

Minimize Fillrate (Rendered Area)

Especially when using transparent materials some areas of the screen need to be rendered again and again. This costs performance.

Try to minimize the rendered area. Mostly it is better to use a smaller mesh instead of using a huge alpha texture where only a small area is used.

Minimize Polygon Complexity

You cannot just import a car with a couple of millions faces and run this on your target device. You are limited in resources. Sometimes by a lot. There is no rule of thumb on how many vertices or faces are allowed per scene. You just need to test this with your target hardware.

Optimize the mesh - one easy way of doing this is to remove edge loops manually. Do not use automatic tools since those will most probably leave you with a messy mesh.

Make sure your mesh is tidy and clean, preferably a quad mesh. Clean meshes can be reduced more without losing quality as opposed to chaotic triangle meshes.

But keep an eye on reflections and test your optimized mesh with reflective materials. Reflective objects are usually less forgiving.

Smaller meshes result in less data to load and therefore less memory to transfer between the CPU and the GPU.

Keep It Simple And Tidy, Use Proper Names

Make the life of the developer easy - make the scene easy to understand to others by tidying it up and giving objects proper names. The scene hierarchy and the objects names are the interface between art and development.