KUESA

AUTODESK® MAYA®

KUESA™ Studio For Maya

The design to code workflow

KDAB

1. Introduction to KUESA™ Studio For Maya

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

KUESA™ Studio for Maya is a plugin for Autodesk Maya that consists of

- An exporter to convert Maya scenes to glTF 2.0 files including the accompanying binary buffers and textures. It supports KUESA™ specific features like Iro materials and KUESA™ layers.
- Integration of KUESA™ materials into Maya and its viewport 2.0 in order to provide real-time WYSIWYG during modeling and animation. The designer can check the visual result inside of Maya without exporting or waiting for an engineer build an app beforehand. . This allows all team members to stay focused in their area of expertise.
- Tools / Maya commands to define features like KUESA™ layers and animation groups.
- The KUESA™ UI panel for convenient access to KUESA™ Maya commands and other tools.

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 Maya and use the tools you are used to - like scripting and other Maya 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 Maya

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

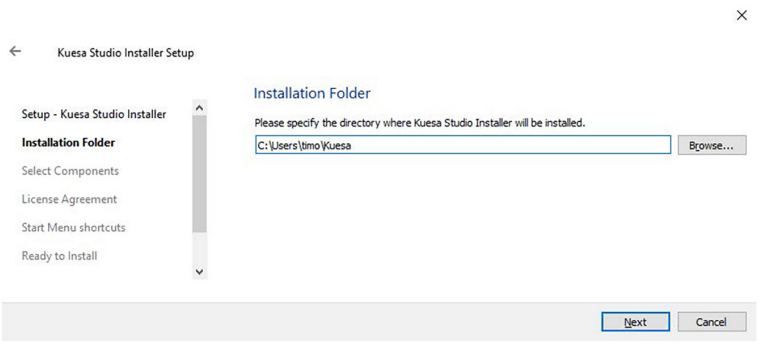
KUESA™ offers provides adds on for multiple 3D authoring tools such as Blender and Maya.

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 Select the folder for KUESA™ Studio.

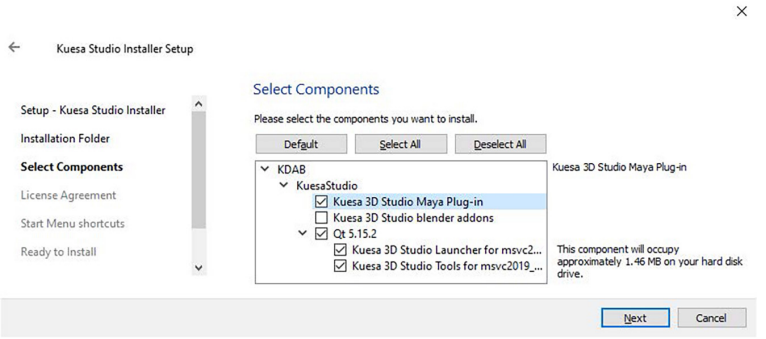
This plugin installer is not specific to Maya, so you will not be asked for your Maya environment. The Maya plugin will not be installed permanently into your Maya installation, so you are free to choose any location.



2.1.2 Choose the components to be installed.

The minimum requirements for Maya would be to install the KUESA™ Studio Maya plugin and the KUESA™ Studio launcher. To inspect the exports, you should also install the KUESA™ Studio Tools which include KUESA™ Studio glTFInspector.

The installer offers multiple components that are available to match your workflow pipeline.



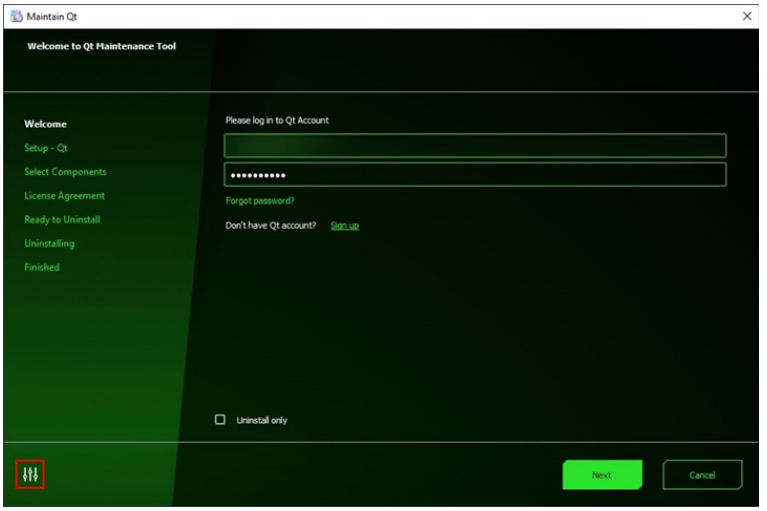
- Accept the license,
- Confirm the start menu shortcut options
- Click ‘Next’.
- Click ‘Finish’.

After confirming with ‘Finish’ you will be provided with the KUESA™ Studio components inside your start menu.

In the section Launching KUESA™ Studio for Maya you will learn how load Maya with the KUESA™ Studio plugins.

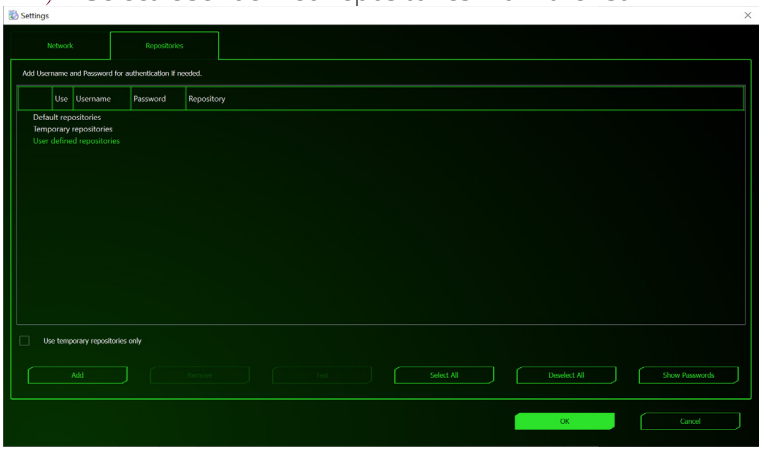
2.2 Qt Online Installer

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

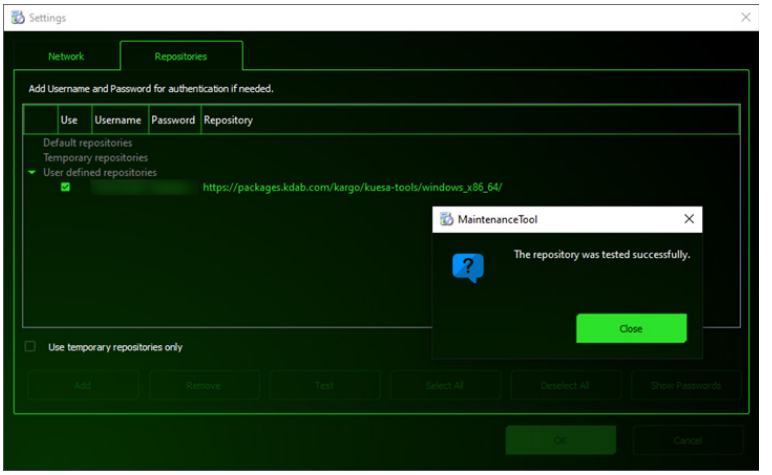


Qt is a cross-platform toolkit for creating graphical user interfaces: www.qt.io

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

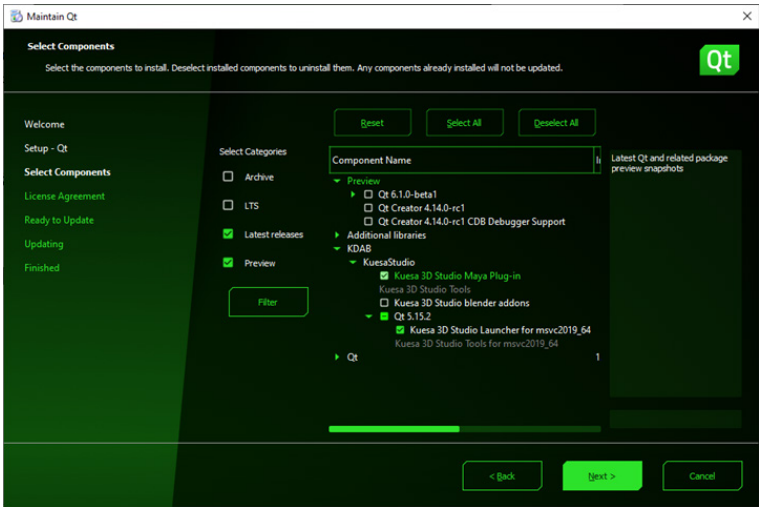


- Click 'Add'
- Add the repository: https://packages.kdab.com/kargo/kuesa-tools/windows_x86_64/
- Click 'Test'
- Enter your user credentials,
- Click on 'OK' to verify that the entered information is correct, and the license is valid.



- 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 Maya this would be at least the KUESA™ Studio Maya plugin and the KUESA™ Studio launcher.



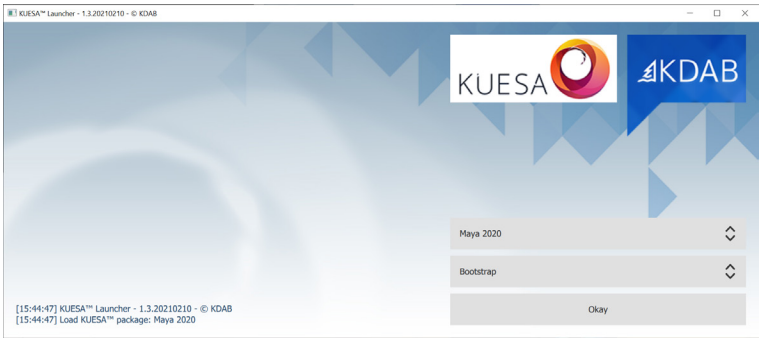
- Click on ‘Next’ to run the installation.

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

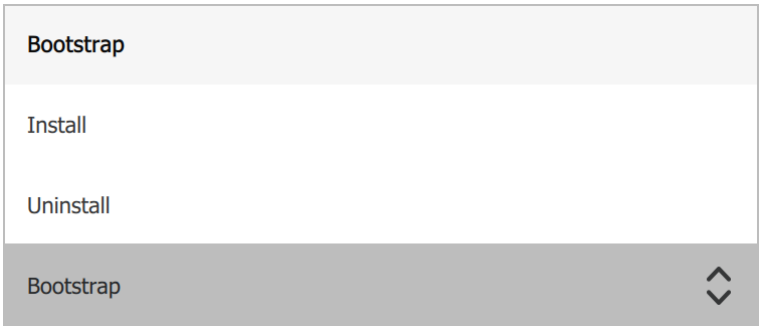
3. Launching KUESA™ For Maya

After the installation of KUESA™ Studio, the KUESA™ plugins for Maya are not permanently hooked into Maya by default and therefore needs to be activated first.

To achieve this KUESA™ For Maya comes with a Launcher.



After selecting the Maya version you want to work with, you are provided with three options: Bootstrap, Install and Uninstall. After choosing one of the options click Okay to apply.



3.1 Bootstrap

This will fire up the chosen Maya Version (if available) and load the KUESA™ plugins dynamically.

Choose this option if you do not want to install the KUESA™ plugins permanently into your Maya installation. The next time you start Maya without the launcher the KUESA™ plugins will not be loaded again. This is useful if you are working on multiple projects in parallel or just do not want to have the KUESA™ plugins around

each time you are using Maya and want to keep your Maya install clean. This is useful if you are working with multiple Maya versions in parallel.

3.2 Install

This will permanently hook the KUESA™ plugins into your personal Maya startup script. After that, the KUESA™ plugins are loaded automatically whenever you start up Maya.

Choose this option if you do not want to start Maya using the Bootstrap option or just want to have the KUESA™ plugins available permanently. This is also useful when you are working with export scripts and need to be able to access KUESA™ functionality from CLI scripts.

3.3 Uninstall

This will permanently unhook the KUESA™ plugins from your personal Maya startup script and therefore is the opposite of Install.

Choose this option if you do not need KUESA™ for Maya anymore or if you want to switch back to using the bootstrap method.

4. Using KUESA™ Studio for Maya - Basics

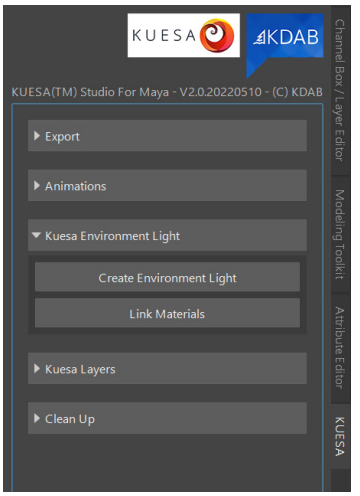
Most of the KUESA™ functionality for Maya is implemented as Maya Python and Mel commands like exporting, creating, and modifying animation groups, etc. The KUESA™ UI is just a convenient wrapper calling those commands. This makes it easy to customize KUESA™ for Maya to your needs and embed those functions into the Maya shelves as you like.

It also enables you to use KUESA™ functionalities (e. g. exporting) from Maya CLI scripts and incorporate KUESA™ seamlessly into your production pipeline.

To see examples of Maya Python and Mel scripts for KUESA™ see Section 8 - Maya Commands Reference of this manual.

4.1 The KUESA™ Panel

The KUESA™ Panel provides convenient access to KUESA™ functionality in Maya such as export, cleanup tools, which are explained later.



By default, when Maya starts up, the KUESA™ Panel tab is shown next to the Attribute Editor. Like every other Maya Panel, you can undock and dock it wherever you want.

If you accidentally undocked or closed and lost the KUESA™ Panel and need to restore it without closing your current Maya scene, you could achieve that with the following script, using the script window or dragging it into a shelf respectively.

Python

```
import maya
maya.cmds.workspaceControl('KUESAWorkspaceControl', e=True, rs=True)
```

4.2 Materials

KUESA™ comes with a set of materials to be used with Maya. This is because offline rendering materials are not the same as real-time materials. You cannot just use your favorite Arnold or V-Ray 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 Maya will look different to the result in the target application. In comparison, KUESA™ materials are integrated in Maya providing WYSIWYG during modeling and animation. We strongly recommend using them instead of standard Maya materials in order to make sure all materials you use are supported and give proper visual feedback in Maya.

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:

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

- 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.

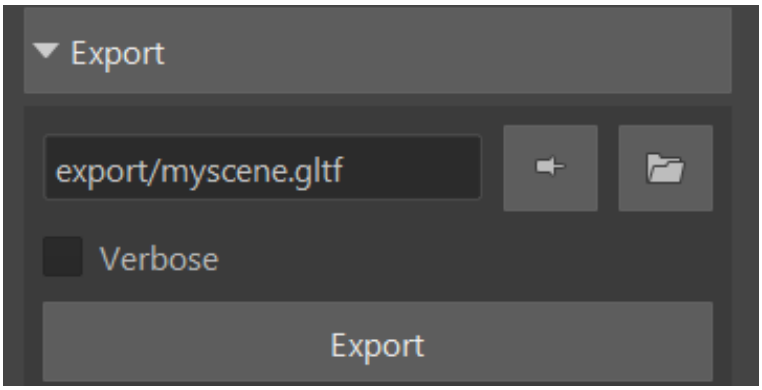
All materials described in the section, KUESA™ materials are supported by KUESA™.

4.3 Exporting

To export your scene to glTF you can either use the export section of the KUESA™ Panel as shown below or the kuesaExport() Python or MEL command respectively (please see Maya Command Reference).

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.

Specify the glTF export file name along with its path in the text box or press the browse button on the right hand side. The exporter will export the scene description to the specified file along with binary buffers and textures to the same directory.



If the pin button is activated, the export path will be saved in the Maya file.

It is recommended to work with Maya projects in order to use relative paths for exporting.

Click on the Export button to start the export process.

If the export fails you will find error messages in the script window or in the output window. To get more detailed output while exporting you can activate the 'Verbose' option.

When the export is finished successfully, it will be indicated by the message 'Terminated' in the script window, in the output window and in the Maya status area.

4.4 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.

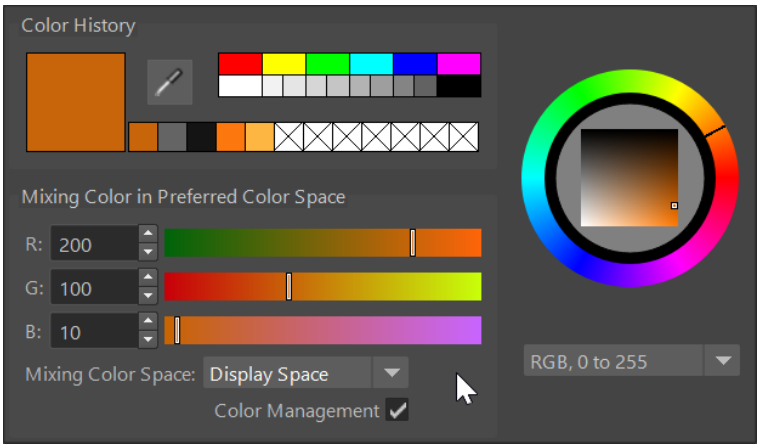
5. KUESA™ Materials

5.1 Maya Settings For KUESA™ Materials

It is highly recommended to specify a Maya project path so you can use relative texture paths and do not need to relink textures when sharing your scene with others.

In order to get a proper visual representation of the KUESA™ materials in the Maya viewport please select 'Viewport 2.0' as renderer for your viewport and enable 'Textured' shading using the shown icons. Also select sRGB as the color management profile, and have view transform on.

When using sRGB values from other applications, ensure that the Mixing Color Space is set to Display Space, and that Color Management is enabled.



Applications such as Adobe Photoshop typically use the sRGB colorspace, while Maya 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.

5.2 Iro Materials – Concept

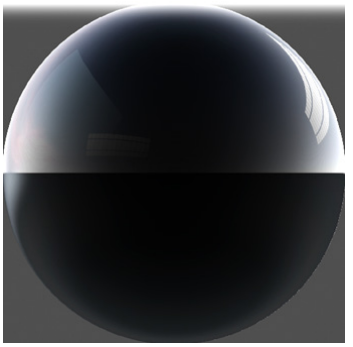
Iro materials are a set of KUESA™ specific materials that

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

All Iro materials are fully integrated into Maya and therefore provide WYSIWYG during modeling and animating. The designer can check the visual result inside of Maya 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 Maya, 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').

5.3 Iro Materials – Features

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

Special Blend Modes

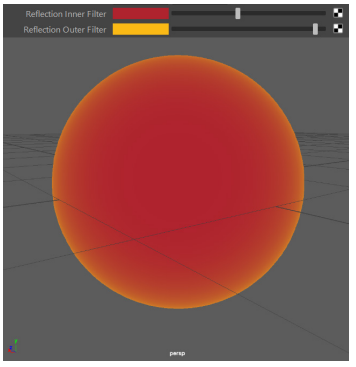
Since the 3D viewport is rendered as a 2D rectangle in the target application, alpha blending is used. This means that materials that are additive or multiplicative (i.e. Iro Glass, Iro Matte Mult and Iro Matte Add) should have a solid background behind them. Either avoid those materials on transparent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



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

Iro Matte Add: Simple additive textured material. (same as the linear dodge (add) 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.

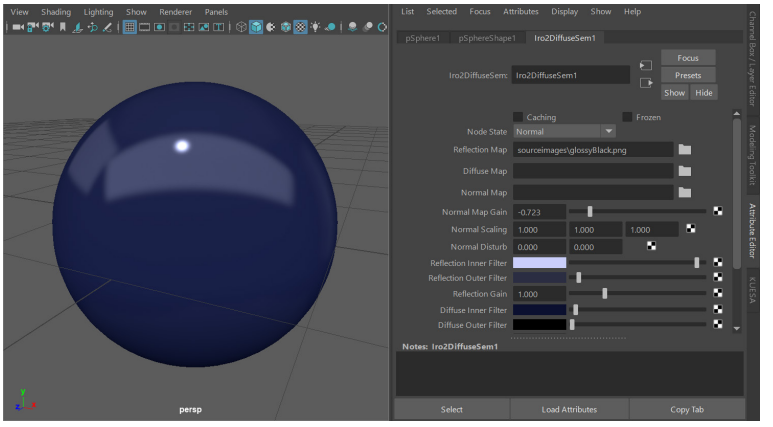
Iro Skybox: Can be applied to an object of any shape (does not have to be a box). The geometry always appears from the viewer's position.

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

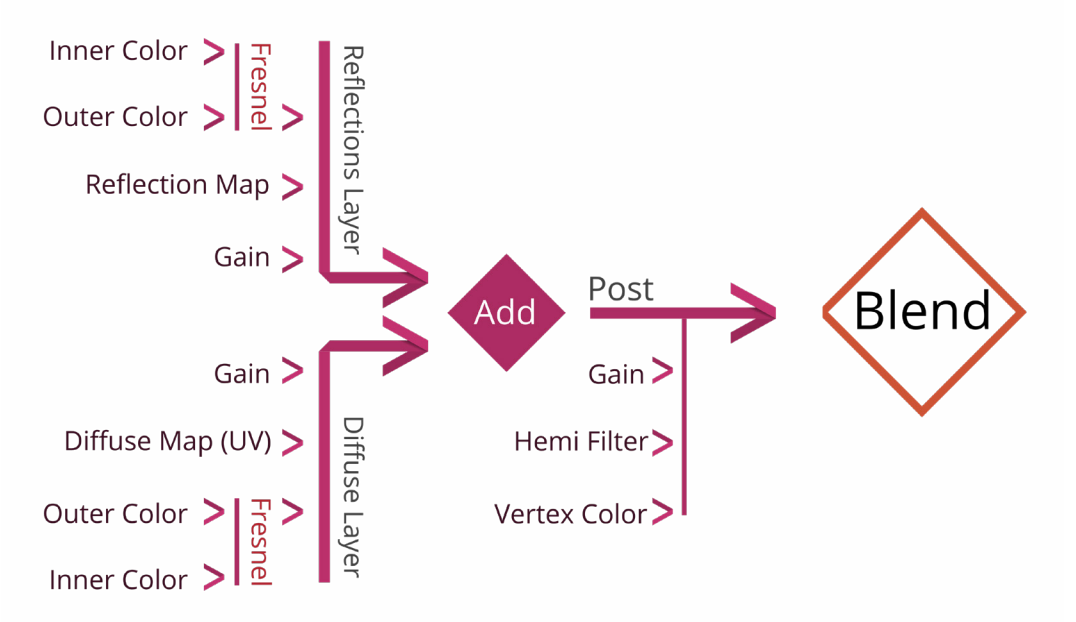
5.5 Iro Materials – Types in Detail

5.5.1 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.



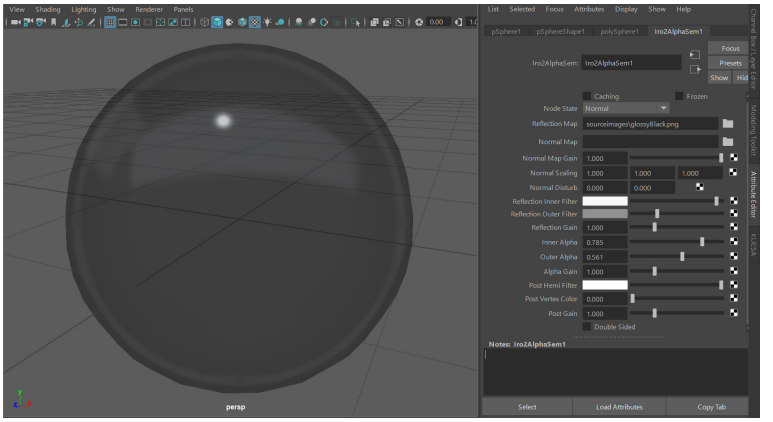
As you can see in the flow chart, the components of each layer (filter colors, map, and gain) are multiplied.



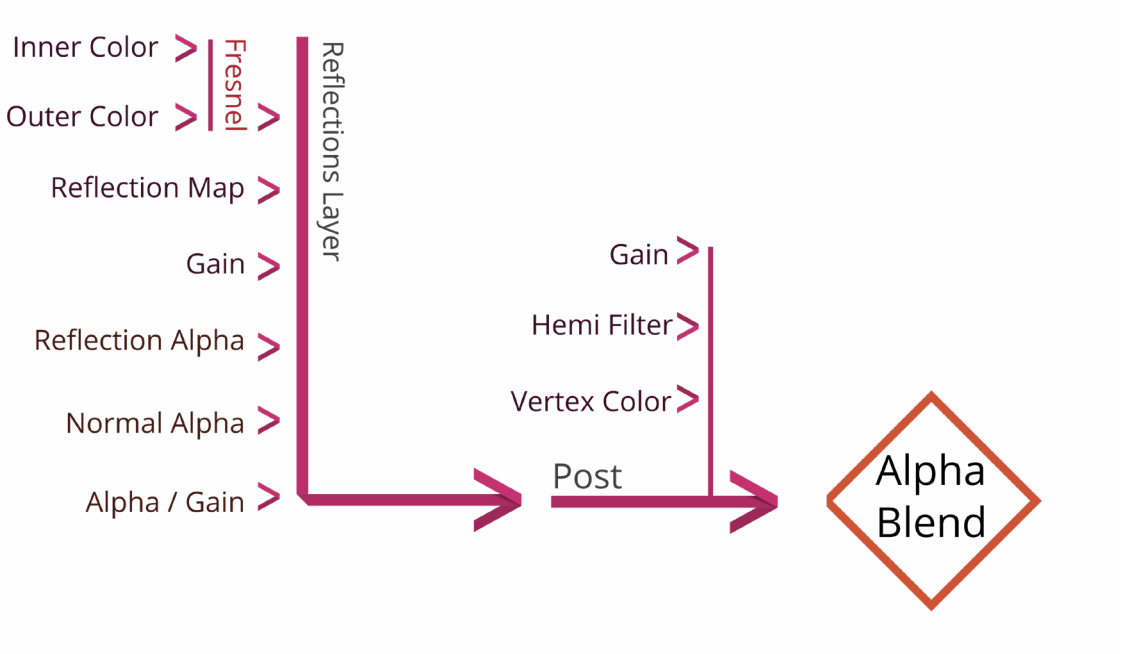
5.5.2 Iro 2 Alpha

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

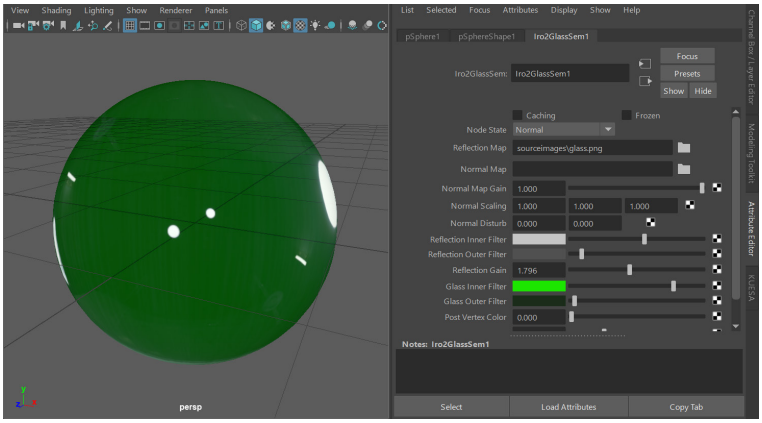


Use the Iro 2 Alpha material if you want to create transparent objects.

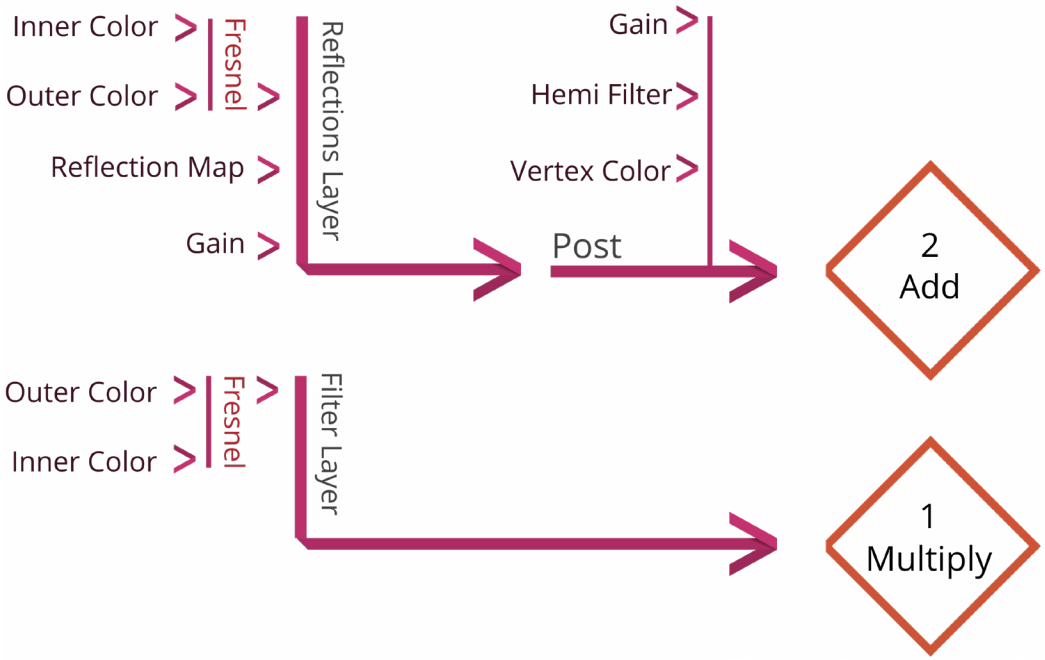


5.5.3 Iro 2 Glass

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.

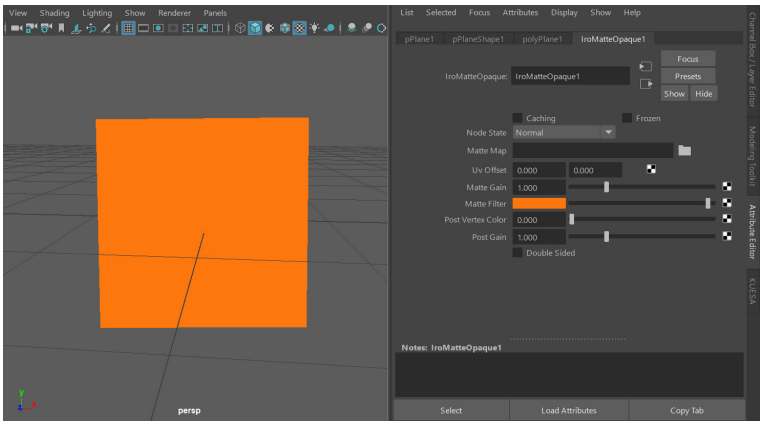


The material mimics the behavior of glass using two passes: The first pass is multiplicative and tints the background down by the glass color. This is similar to colored glass where only a part of the light / the color spectrum comes through. The second pass adds reflections.



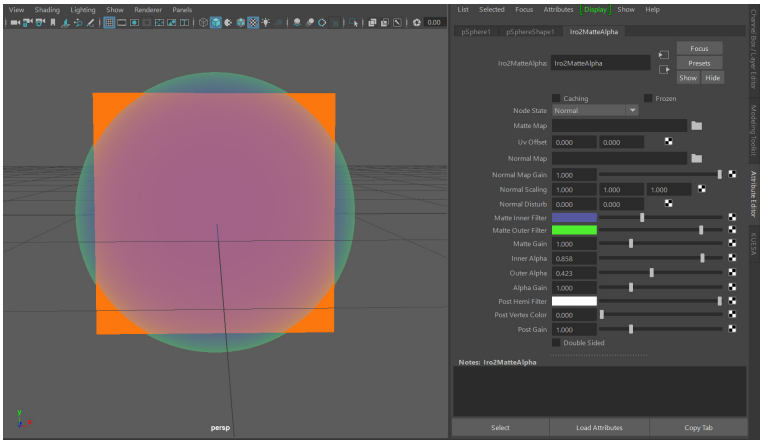
5.5.4 Iro Matte Opaque

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



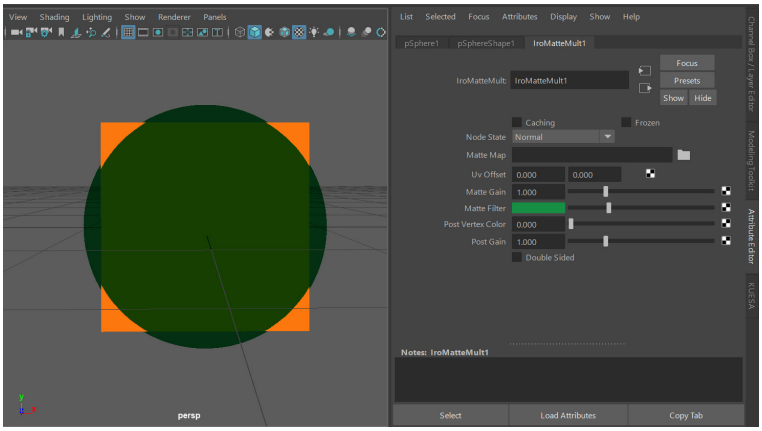
5.5.5 Iro 2 Matte Alpha

Simple transparent material using alpha blending.



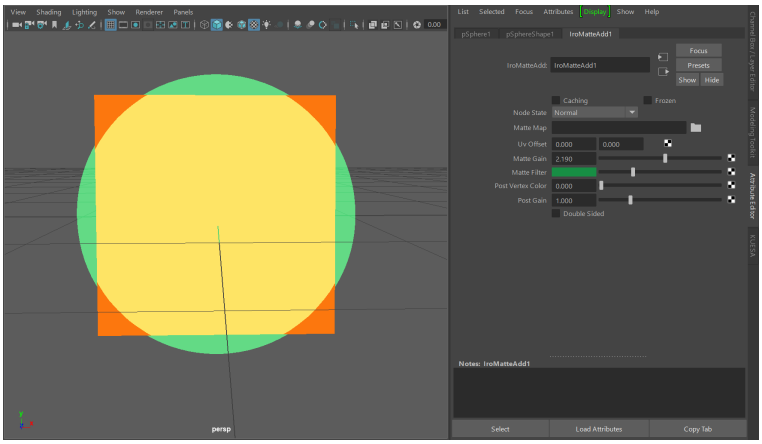
5.5.6 Iro Matte Mult

Simple material using multiplicative blending (implicitly transparent).



5.5.7 Iro Matte Add

Simple material using additive blending (implicitly transparent).

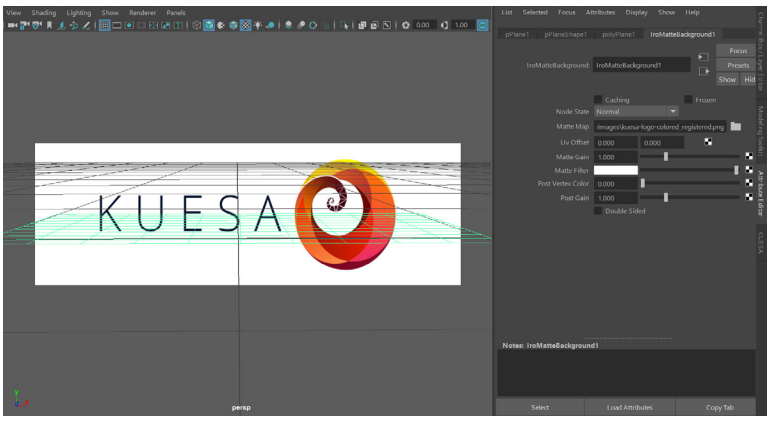


5.5.8 Iro Matte Skybox

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.

5.5.9 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.



5.6 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.

5.6.1 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.

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.

For some KUESA™ Materials, there are two subtypes - the SEM mode and the ERM mode material.

5.6.2 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.

5.6.3 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 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.

5.6.4 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.

5.6.5 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 Gain: Use this to easily tweak brightness.

5.6.6 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.

5.6.7 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).

5.6.8 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.

5.6.9 Miscellaneous

Double Sided: Sets the double sidedness of the material in KUESA™. It is best practice to not select this unless specifically required as it has a performance cost and may cause rendering artifacts with transparent materials. Note: Maya decides whether to render

double sided by shape, not by material. So, to render single sided in Maya, you would need to turn 'Double Sided' off in the render stats of the shape.

5.7 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 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	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	

5.8 PBR Materials

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.

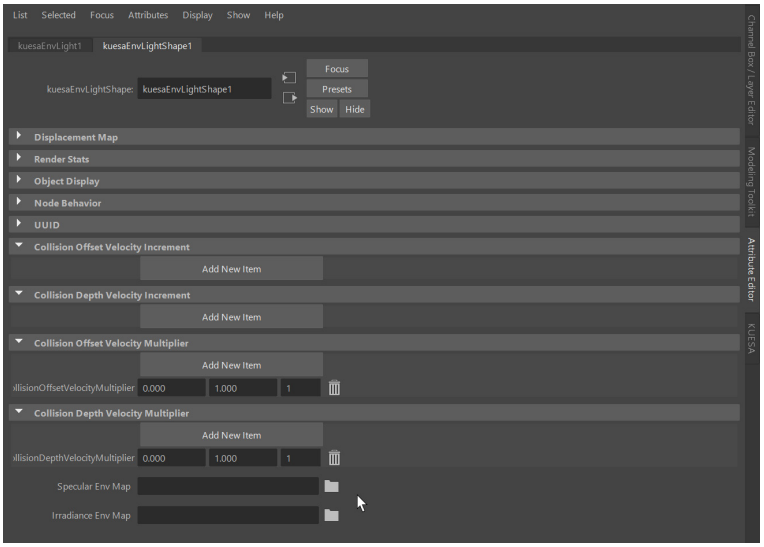
5.8.1 Setup

Creating a scene with a PBR material involves a small amount of preparation. In order for the material to be viewed correctly, an environmental light should be set up and linked to the materials that are used. For the environmental light, two enviromental maps are required - specular and irradiance. These should be created as cube maps.

First, add a mesh to the scene and add a new PBR Metallic roughness roughness to it.

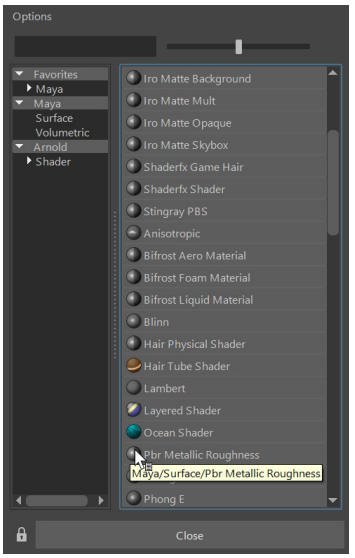
From the Kuesa panel, click on the 'Create Environmental Light' button.

Select the Environmental light and click on the AttributeEditor tab.



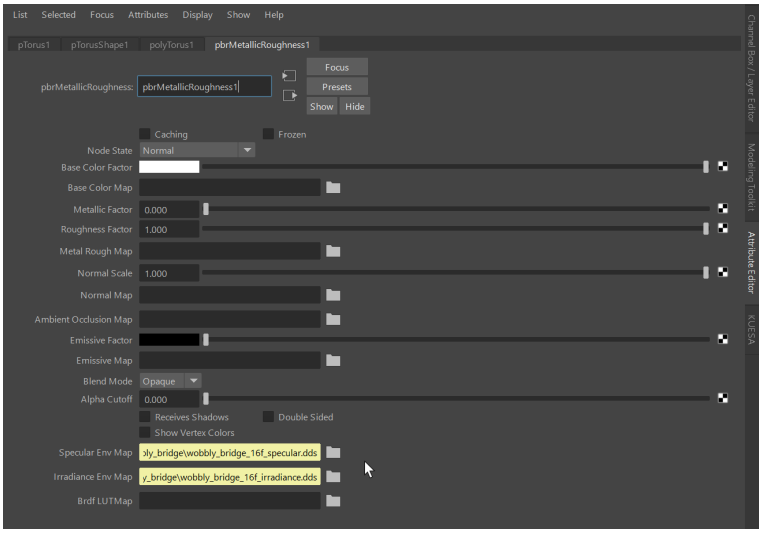
Add the specular and irradiance maps to the fields at the bottom of the panel.

Choosing the Pbr metallic Roughnesss material.



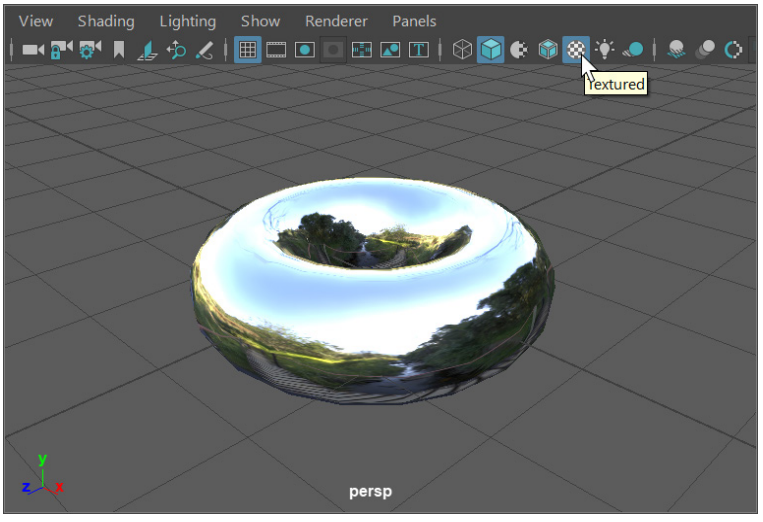
Select the shape that was created and click link materials

To check that the specular and irrandience maps have been linked correctly, inspeact the material on the mesh and check that the maps have been populated.



The Kuesa Panel showing the 'Create Environmental Light' and 'Link materials' buttons.

To preview what the material looks like ensure that the 'Textured' option has been enabled in the viewport.



5.8.2 Standard PBR Material

Base Color Factor: The base color of the material.

Base Color Map: A color image that multiplies with the Base Color Factor to give a more detailed color for the material.

Metallic Factor: Whether or not the material is metallic. 0 = diffuse only, 1 = fully metallic.

Roughness Factor: The blurriness of the reflections done by selecting another mip level.

Metal Rough Map: A greyscale image that multiplies with the Roughness Factor and Metallic Factor to allow the metallness to vary over the surface of the material.

Normal Scale: The amount that the Normal map affects the material.

Normal Map: An image that represents the angle of the normals - same as for traditional materials - which allows for detailed textures to be simulated.

Ambient Occlusion Map: A black and white image that gives shadow detail over the surface of the mesh. This is usually baked from the model.

Emissive Factor: The amount and color of the light that the material is emitting.

Emissive Map: An image that allows for more fine control of what parts of the material are emitting light.

Blend Mode: Options: Opaque/Blend/Mask. Opaque - the material is solid and has no transparenc. Blend - The material has an element of transparency and the transition is gradual. Mask - The material has an element of transparency and the transition is sharp.

Alpha Cutoff: If a Base Color Map is used that has an alpha channel, this defines at what amount of alpha the material becomes transparent.

Receives Shadows: Whether or not the material will show shadows created by other meshes.

Double Sided: When selected, this allows the back side of the mesh to be visible.

Show Vertex Colors: If vertex colors have been painted, this will multiply the Base Color Factor and/or Base Color Map with the values to give the output. This can be used to create shadows on the object if an ambient occlusion map is not set.

Specular Env Map: Linked from the Environmental Light.

Irradiance Env map: Linked from the Environmental Light

Brdf LUTMap: The standard map is used if this is left unset.

5.8.3 Clear Coat Extension

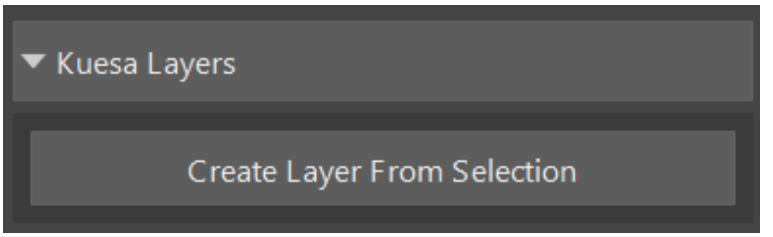
(In Progress for Maya)

The Maya Viewport 2.0 implementation of the clear coat extension is in progress.

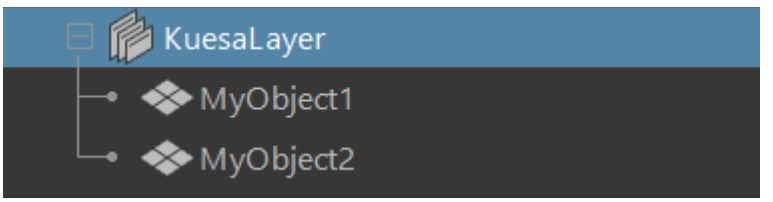
6. Using KUESA™ Studio for Maya - More Features

6.1 KUESA™ Layers

You can organize your nodes in KUESA™ layers to treat them accordingly later in your target application using the framegraph. To create a new layer from selected nodes, use the 'Create Layer from Selection' button in the Kuesa Layers section of the KUESA™ panel. If no nodes are selected an empty layer will be created.



After creating a new layer, it is shown in the outliner. This layer acts as a Maya set that you can rename and drag nodes into. A node can belong to multiple layers.

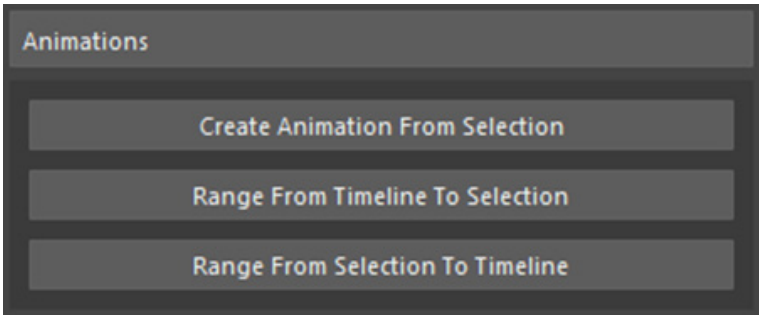


When exported to glTF the layer will be named as shown in the outliner and contain all objects as shown in the outliner.

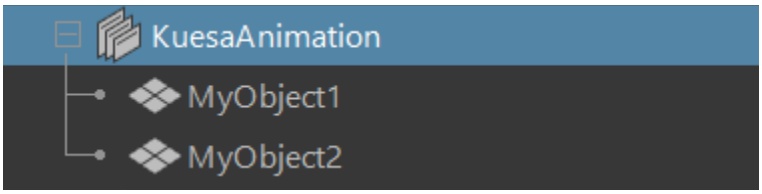
Layers can also be organized using MEL and Python scripts. Please see the Maya Commands Reference.

6.2 Animation Groups

Animation groups is a powerful tool that enables you to define how animations will be exported to glTF. In the glTF universe an animation consists 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.



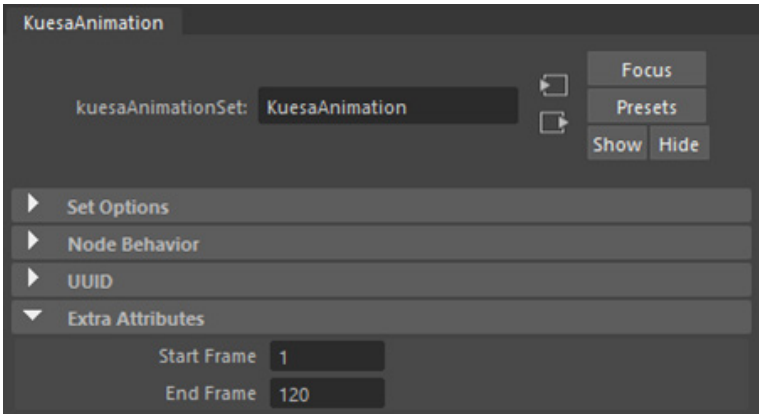
To create a new animation group use the button 'Create Animation From Selection' in the Animations section of the KUESA™ panel. If no nodes are selected an empty layer will be created.



After creating a new animation group, it is shown in the outliner. This animation group acts as a Maya set you can rename and drag objects into. Currently nodes and supported materials are allowed to be animation group members. An object can belong to multiple animations.

By default the current timeline range is used for a new animation group. You can change its range with the attribute editor as shown or using the buttons of the Animations section. If you want to quickly preview an animation you also can bring over the range from the animation to the timeline using the button 'Range From Selection To Timeline'.

When exported to glTF the animation group will be named as shown in the outliner and contain all animated objects as shown in the outliner. Objects that are not animated will be skipped.



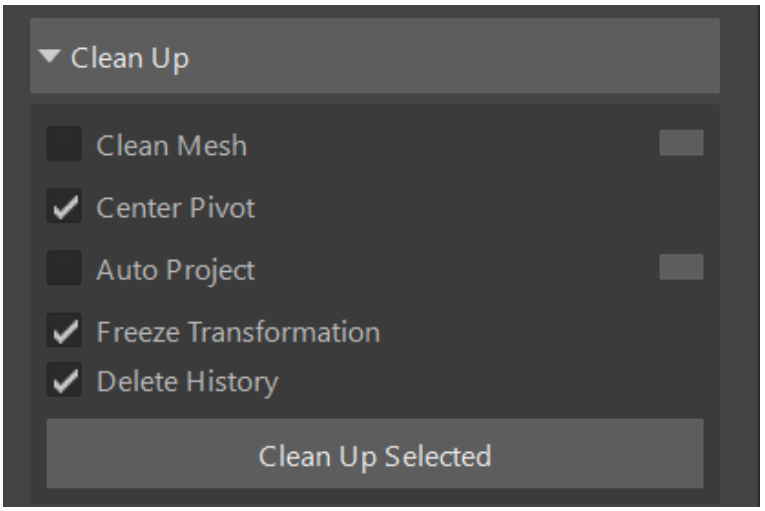
Animation groups can also be organized using MEL and Python scripts. Please see the Maya Commands Reference.

6.3 Clean Up Tools

Sometimes after importing model data from an external source the mesh may contain invalid triangles that converge to lines or points - or UV sets may be missing. Also, not every type of Maya dependency nodes are supported when exporting to glTF. For some of those cases the export or import could fail or even worse: There will not be any error, but the results will look strange. Therefore, it is recommended to clean up the objects from time to time.

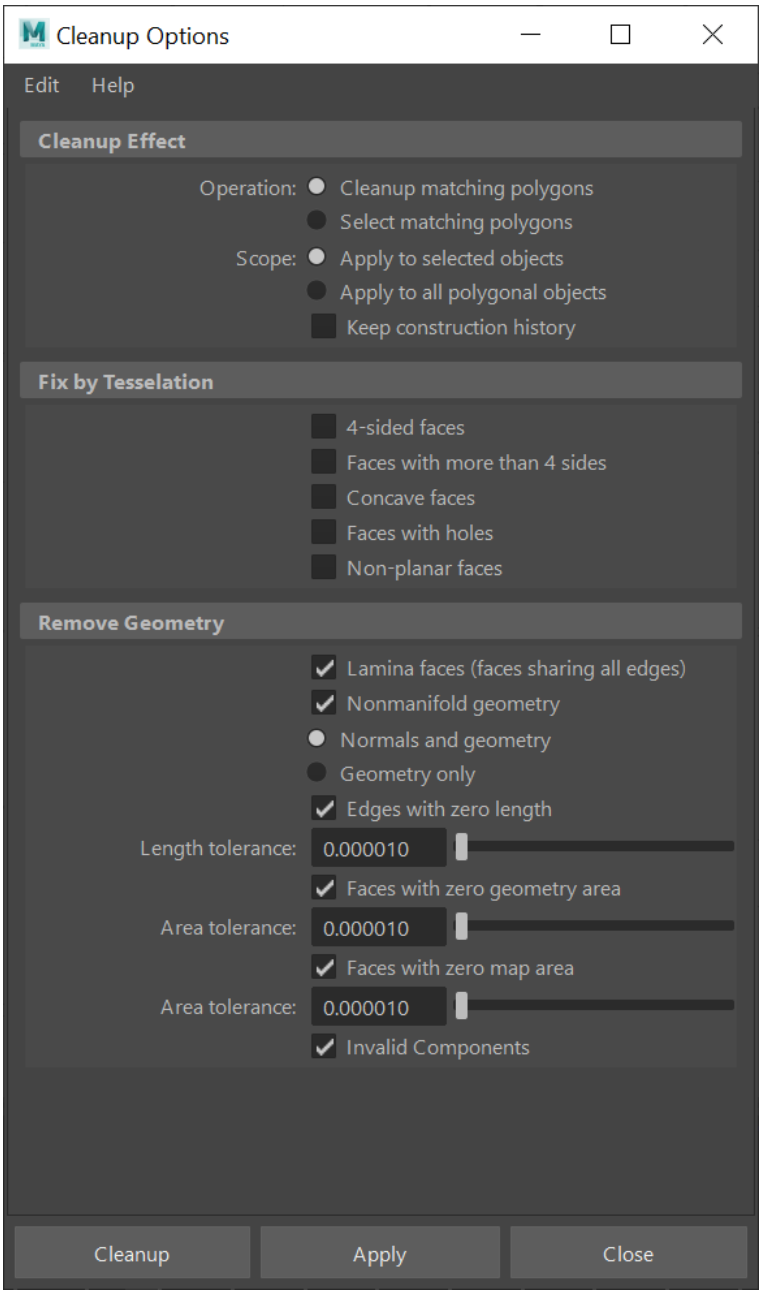
The clean-up section does not provide its own functionality but is a collection of existing Maya tools for clean-up and simplification in one place. When clicking the button 'Clean Up Selected', all selected options will be performed for all selected objects using the settings for the respective Maya tools.

Do not just apply all clean up features to the entire scene - use them wisely and where needed since most of these operations are destructive.



Clean Mesh

Cleans up the mesh and avoids invalid mesh structures - it executes the Maya cleanup tool. This is useful to ensure that you get a valid mesh without any degenerated triangles. Please find the screen shot below for recommended example settings.



Center Pivot

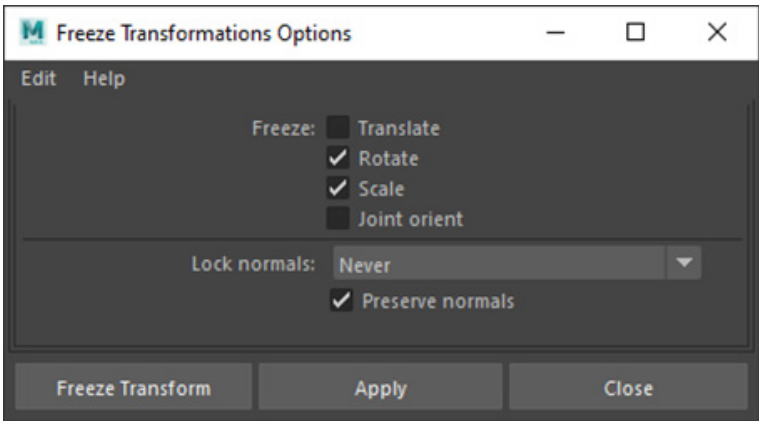
Centers the pivots of all selected objects.

Auto Project

Use this if UV sets are missing and corrupted to quickly (re-) generate them.

Freeze Transformation

It is recommended to bake rotation and scale into the mesh before exporting.



Delete History

Deleting the history is one of the most important clean up options. This will get rid of as much Maya dependency nodes as possible (destructively). It is highly recommended to do this before exporting to avoid unforeseen behavior caused by unsupported Maya nodes.

6.4 Export Scripts

One big advantage of KUESA™ Studio is its direct workflow. You can use the glTF exports from Maya in your application, directly out of the box. This way you can easily make the KUESA™ Studio exporter for Maya part of your pipeline.

Let us 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. You can do this using the kuesaExport() command which is available in Python and MEL. Please find an example below.

CLI

```
cd <maya project dir>
maya -batch -file child-scene-1.mb -command 'kuesaExport -v -of \'export/1.gltf\';'
maya -batch -file child-scene-2.mb -command 'kuesaExport -v -of \'export/2.gltf\';'
```

Note: Before you can use KUESA™ Maya commands from CLI scripts you need to make sure you installed KUESA™ Studio for Maya permanently so it loads whenever you fire up Maya. Please see Launching KUESA™ For Maya.

For further information about KUESA™ Maya commands please refer to Maya Command Reference.

7. 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.

8. Maya Commands Reference

Most of the KUESA™ tools for Maya are accessible via the KUESA™ panel. However you might want to create your custom Maya shelves or UI panel or include KUESA™ functionality into your Maya scripts or implement automatic KUESA™ exports from Maya inside your project pipeline. All KUESA™ features are accessible in both worlds – Python and MEL.

8.1 kuesaVersion()

Gets the version of the loaded KUESA™ plugins for Maya.

Parameters

```
Result (string): Version of loaded KUESA for Maya plugins
```

Examples

Python

```
import maya
print(maya.cmds.kuesaVersion())
```

Prints out the version of the loaded KUESA™ plugins.

Mel

```
kuesaVersion;
```

Shows the version of the loaded KUESA™ plugins.

8.2 kuesaLayer()

Manage KUESA™ layers.

Parameters

```
h / help:          Print out usage help
v / verbose:       Verbose output
ins / inSelected:  Use selected objects for layer creation
c / create:        Create a new layer
sn / setName:      Set the name of the layer
Result:            None
```

Examples

Python

```
import maya
maya.cmds.kuesaLayer(c=True, ins=True, sn='MyNewKuesaLayer')
```

Creates a new KUESA™ Layer with the name MyNewKuesaLayer
and adds all selected objects to it.

Mel

```
kuesaLayer -c -ins -sn 'MyNewKuesaLayer';
```

Creates a new KUESA™ Layer with the name MyNewKuesaLayer
and adds all selected objects to it.

8.3 kuesaAnimation()

Manage glTF animation groups.

Parameters

```
h / help:                 Print out usage help
v / verbose:              Verbose output
ins / inSelected:         Use selected objects for layer creation
c / create:               Create a new animation group
sn / setName:             Set the name of the animation group
sr / setRange:            Set the frame range
ssf / setStartFrame:      Set start frame of the range
sef / setEndFrame:        Set end frame of the range
rtt / rangeToTimeLine:    Set start and end of the timeline according to anim group
ttr / timeLineToRange:    Set range of anim group according to timeline range
Result:                   None
```

Examples

Python

```
import maya
maya.cmds.kuesaAnimation(c=True, ins=True, sn='MyNewAnim', sr=(1,40))
```

Creates a new animation group with the name MyNewAnim, adds all selected objects to it and sets the range to frames 1..40.

Mel

```
kuesaAnimation -c -ins -sn 'MyNewAnim' -sr 1 40;
```

Creates a new animation group with the name MyNewAnim, adds all selected objects to it and sets the range to frame 1..40.

8.4 kuesaExport()

Exports the current Maya scene to a glTF 2 file (*.gltf) supporting KUESA™ specific features. Further exports accompanying binary buffers and copies used texture files.

Parameters

```
h / help:           Print out usage help
v / verbose:        Verbose output
of / outFile:       Specifies glTF output path (*.gltf)
gi / gltfIndent:    Specifies indentation for the glTF output JSON file
```

Examples

Python

```
import maya
maya.cmds.kuesaExport(of='myscene.
gltf', v=True)
```

Exports the Maya scene to myscene.gltf using verbose output.

Mel

```
kuesaExport -of 'myscene.gltf' -v;
```

Exports the Maya scene to myscene.gltf using verbose output.

9. 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 Maya. 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 - repair invalid meshes, freeze some transformations, and delete the history to minimize issues. Check out the section Clean Up Tools.

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.