Mesh Extractor 2

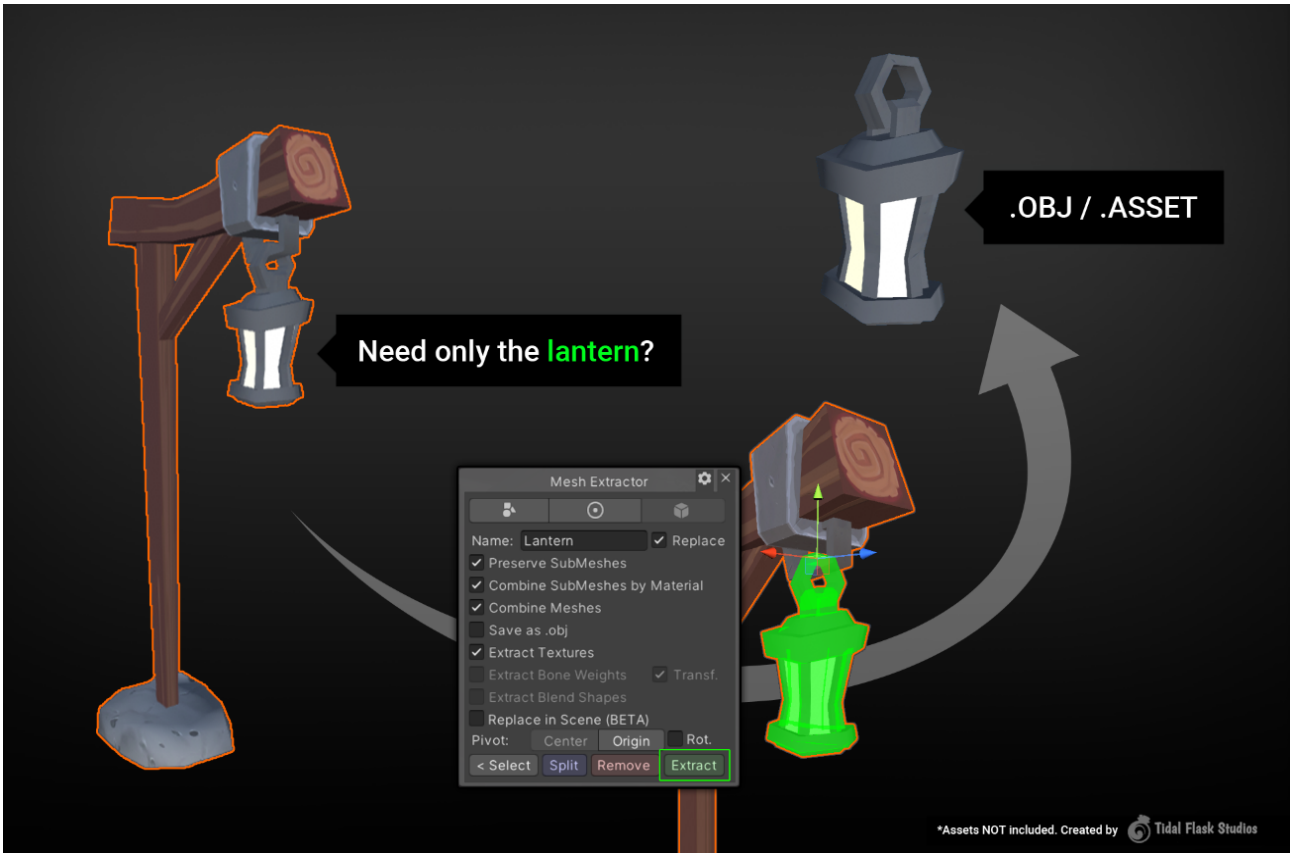


Table of contents

What's it good for?

If you are like me then over time you have purchased a lot of awesome assets on the AssetStore. But often you only need a small part of it.

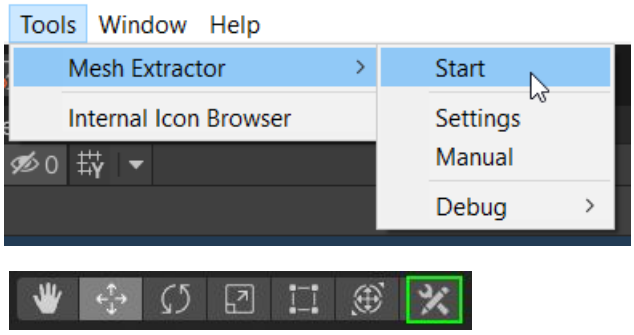
Let's take this asset from TidalFlask for example. It's a great looking asset but what I need is just one sheet and a nail. Sadly it's all delivered in one single mesh.

With Mesh Extractor I can get the what I need within one minute. No modelling software is needed.

Usage

Start the tool

Open the tool via **Tools > Mesh Extractor > Start** (or via the Tools bar).



Select objects

The tool starts in the „Select Object" TAB. Here you need to select the object to extract from.

Reset: Clears the current selection, deselects any object and resets all configurations to default.

Paint your selection

Once the object is selected you can start selecting triangles. Simply click or drag your mouse over the object. Press **CTRL** to erase you selection.

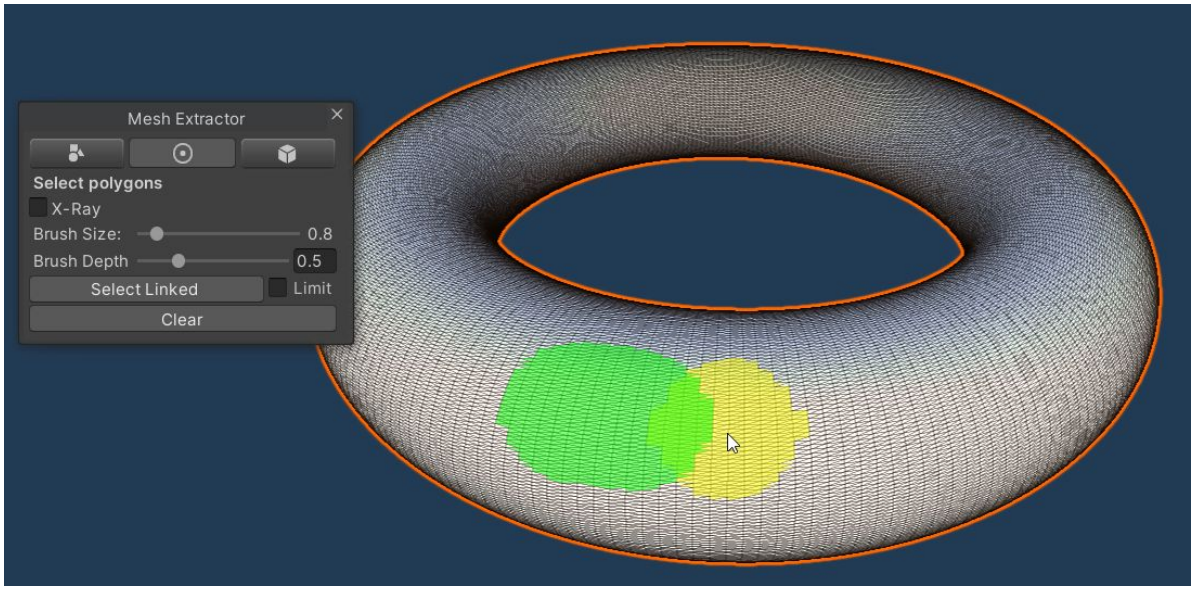
In the top right you will finde the Save & Load Options for the selections:

X-Ray: X-Ray mode allows you to select front and back facing triangles at the same time.

Brush Size: Reduce the brush size to 0 to select only one triangle at a time.
You can also use **SHIFT + MOUSE WHEEL** to change the brush size.

Personally I mostly use brush size 0 in combination with the „Select Linked" button (see below).

Brush sizes greater than zero are useful for high poly meshes where selecting single triangles would be too cumbersome. Like in this scenario:

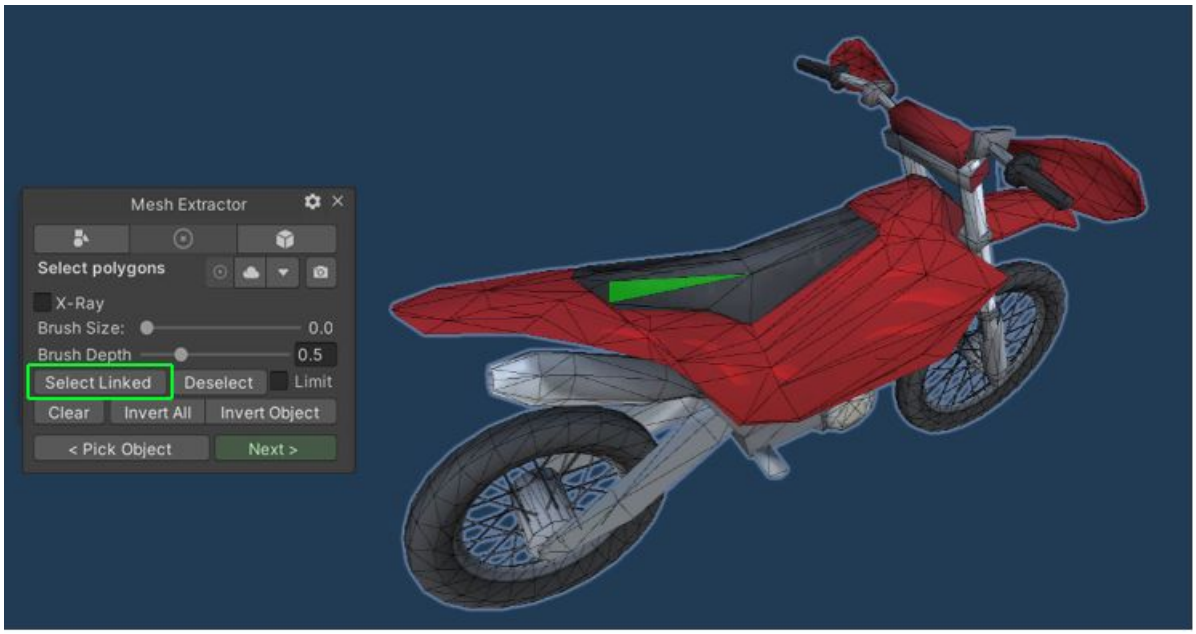


Brush Depth: Brush depth defines how far into the object the selection will go. This helps to avoid selecting background polygons by accident. If you want infinite depth then simply turn on X-Ray.

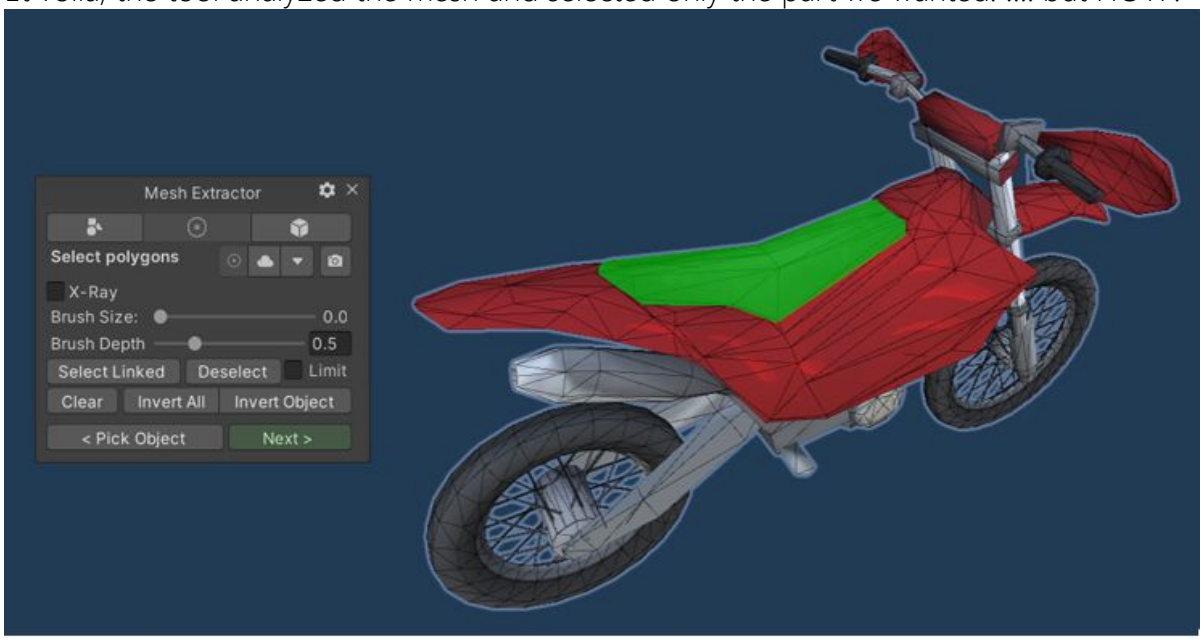
Select Linked: Selects all triangles which are connected to the last selected triangle.

This one may need some more explanation. In many cases selecting single triangles is a lot of work and actually what you want to select is a part of a mesh which has triangles sharing the same vertices (meaning one triangle is connected to another triangle with at least one shared corner point).

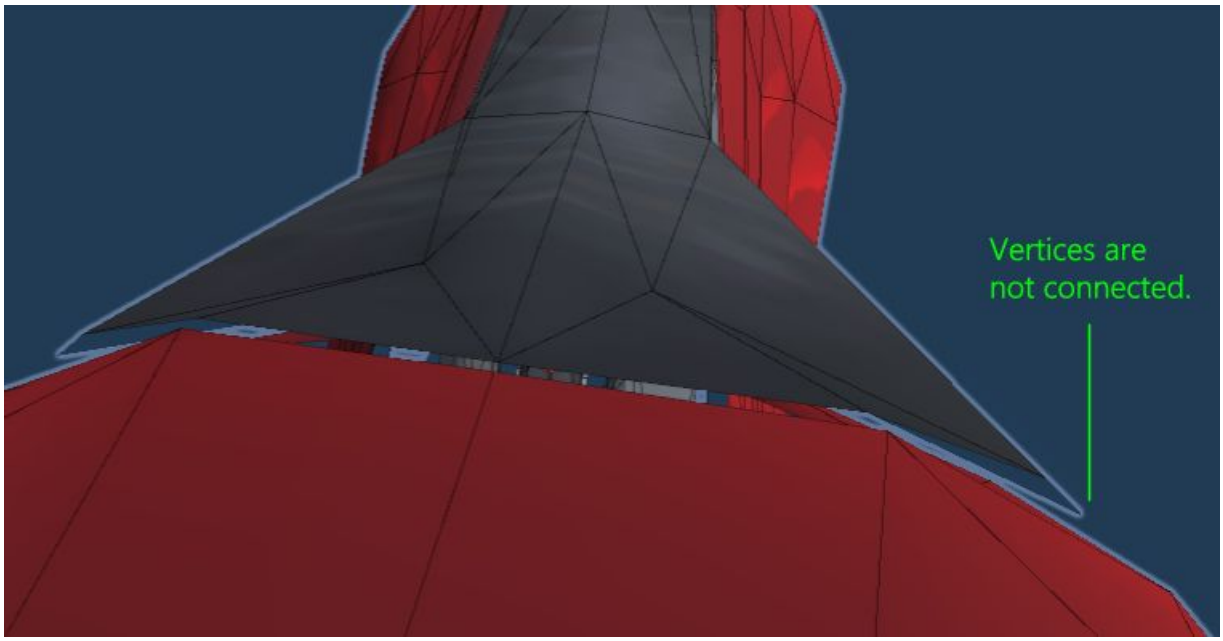
Example: Here we want to select only the seat of the bike.
We select only one triangle of the seat and then hit the „Select Linked" button.



Et voilá, the tool analyzed the mesh and selected only the part we wanted. .... but HOW?

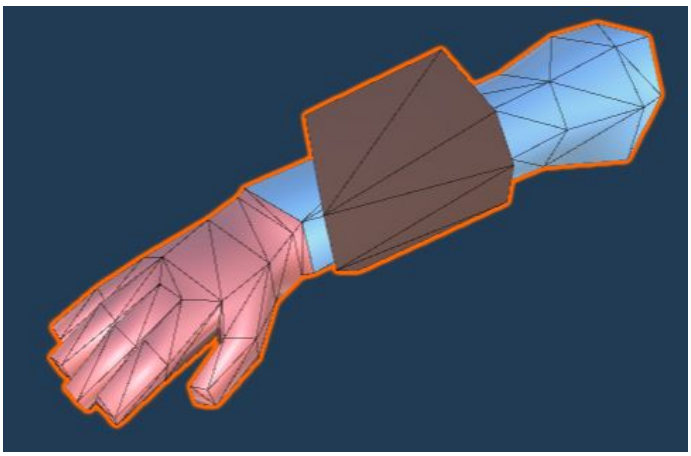


If we look closely we see that the seat mesh actually does NOT connect with the bike body. The tool can use that information.

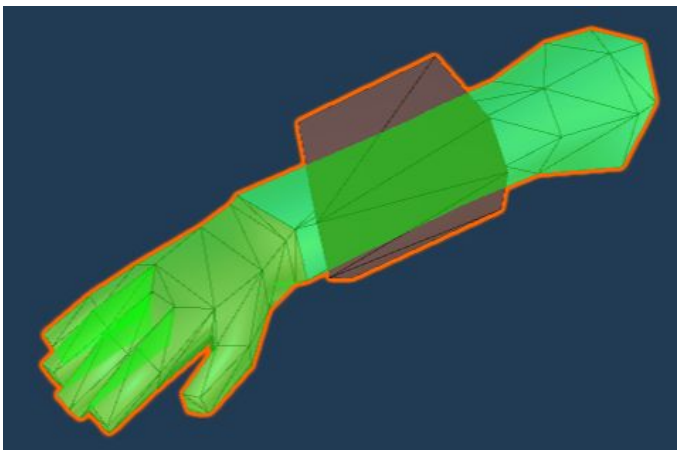


Select Linked > Limit: Enable this to limit the selection to a single sub mesh.
It will use the sub mesh of the last selected triangle.

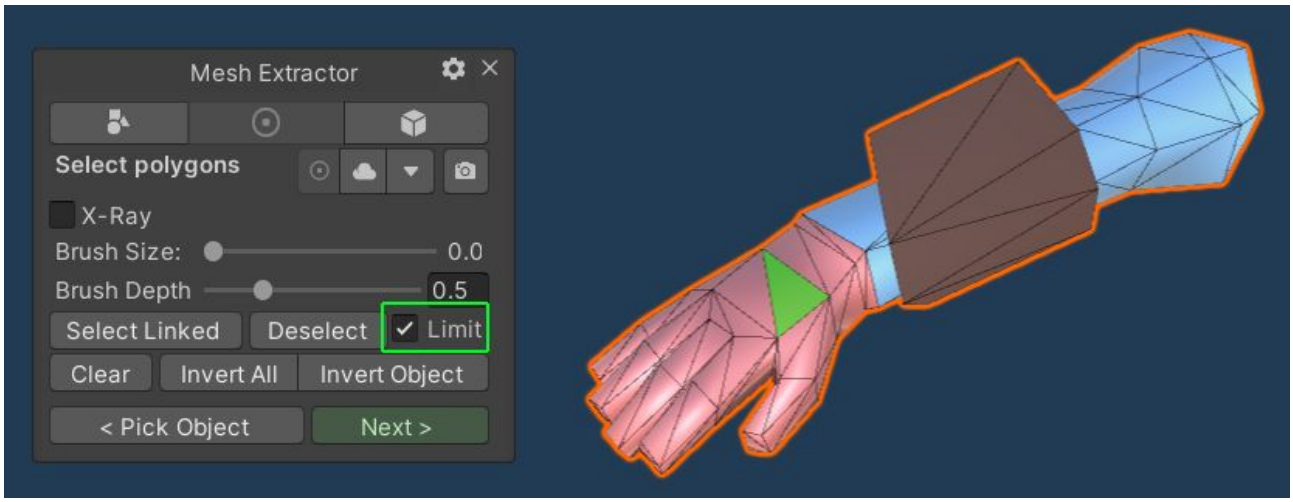
Again this one needs some more explanation. Let's take this mesh for example.



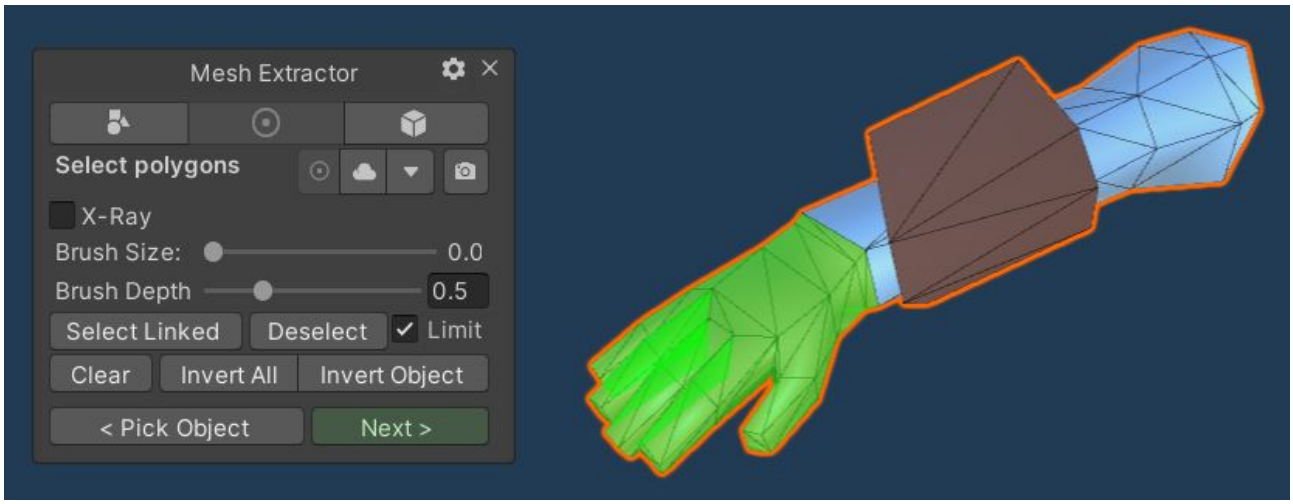
All the arms vertices are connected. If we use „Select Linked" on it the whole arm will be selected. Like this:



We can see from the assigned materials that the arm has some sub meshes (one for the arm, one for the hand). Let's enable **Limit** to limit the connected selection to a sub mesh.



This is the result if the „Select Linked" options is limited to a sub mesh.



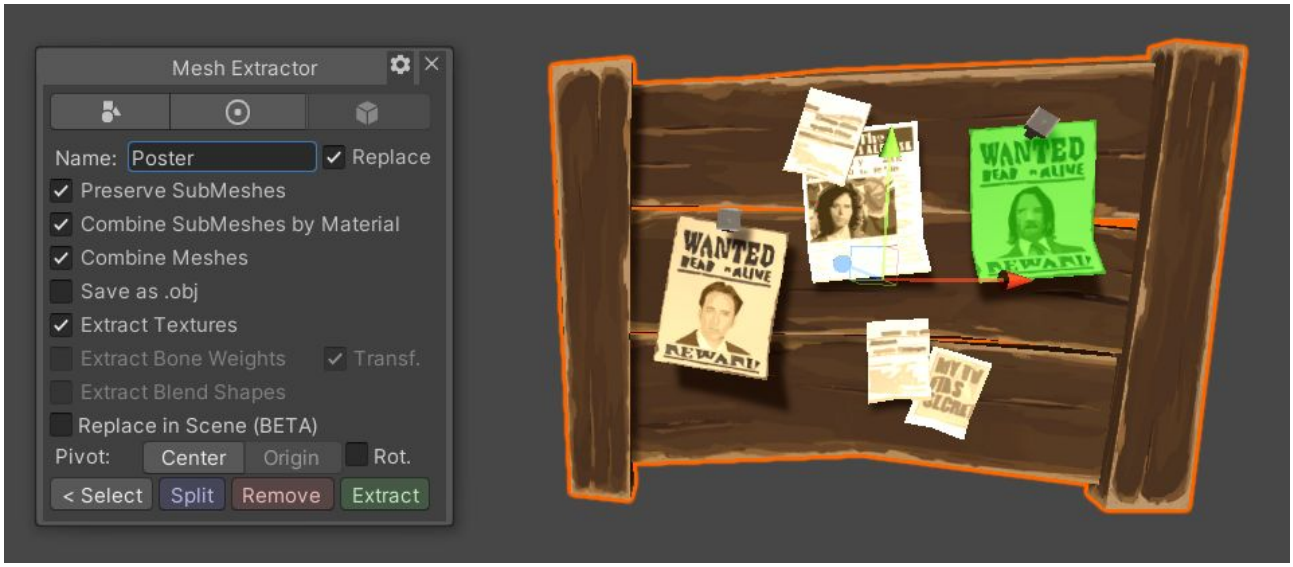
Clear: Clears the current selection.

Invert All: Inverts the selection across all selected objects.

Invert Object: inverts the selection only on the objects that have triangles marked as selected.

Extract Mesh

After selecting the triangles we are ready to extract the mesh.

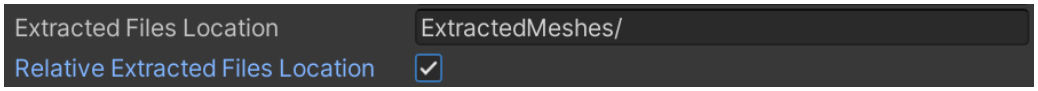


Name: The base name of the newly generated assets.

The assets are saved under **Assets/ExtractedMeshes/[TheNameYouEnter]**. You can change the base path „ExtractedMeshes" in the settings (Tools > MeshExtractor > Settings : Extracted Files Location).

😲 HINT: You can use relative paths here too. These are then relative to the „Extracted Files Location". So if you want your assets to be stored directly in the Assets/ folder then simply prepend a „../". Like this:

😲 HINT 2: You can enable „Relative Extracted Files Location" in the settings to make the resulting files be stored at the same location as your original mesh instead of „Assets/ExtractedMeshes":



Replace: If enabled then the new prefab will replace the old. If disabled the new prefab will be stored with a new name.

Preserve SubMeshes: Enable to preserve sub meshes in the new mesh. If disabled then all sub meshes within one renderer will be merged into a single mesh.

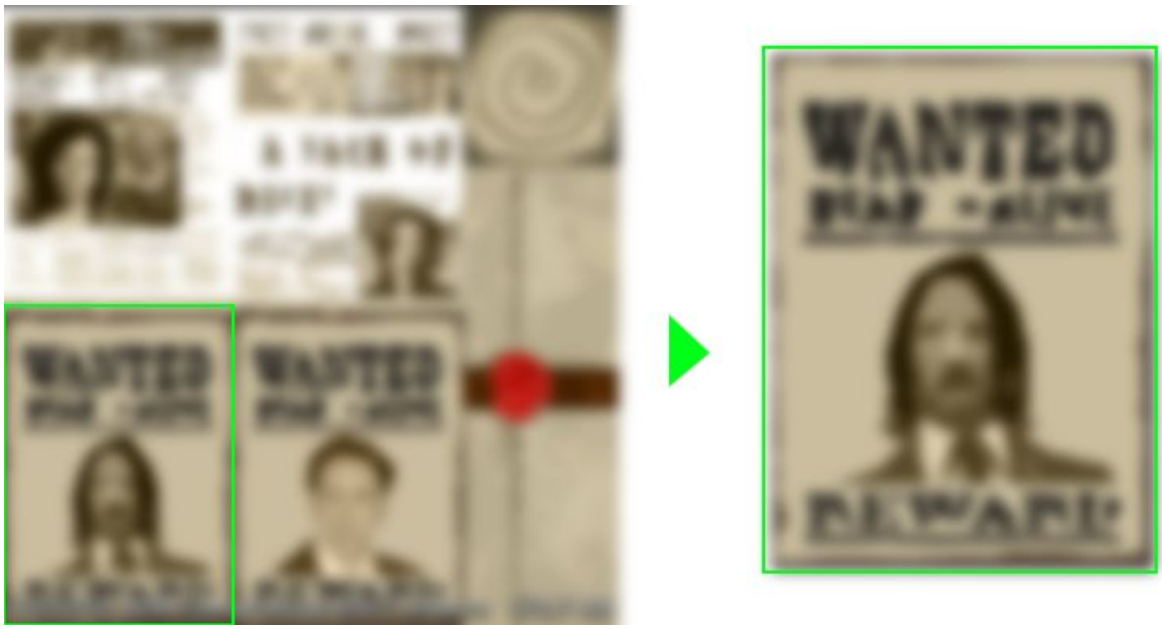
Combine SubMeshes by Material: If multiple sub meshes have the same material assigned to them then these will be merged into one submesh if this option is enabled. This has no effect if 'Preserve SubMeshes' is disabled.

Save as .obj: Export the mesh as .obj & .mtl files instead of an .asset file.
NOTICE: The obj format does only support one set of UVs and no extra info like bone weights. If you have „Extract Bone Weights" enabled then the „Save as obj" option will not be available. Uncheck it to save as .obj.

HINT: Unity has a FBX Exporter Plugin which you can use to convert the .asset files to .fbx.
Extract Texture: Extract the parts of the texture that are used by the selection. It creates a new (possibly smaller) texture by stripping away unneeded texture space.

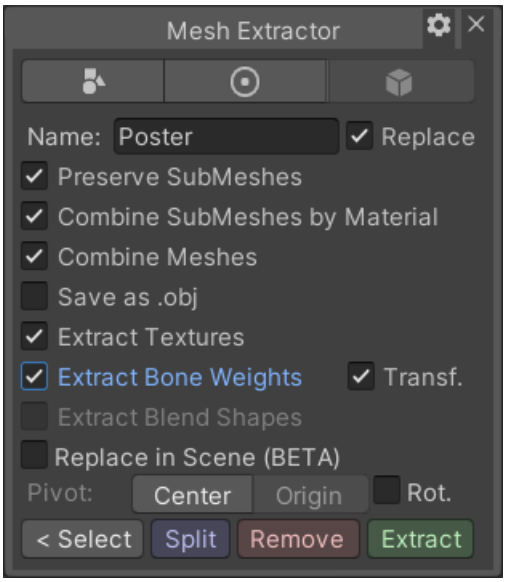
NOTICE: The reduction of the texture size depends on the original UV layout (it uses a bounding box).

„Extract Texture" reduces the size of the texture by copying only the parts that are needed.

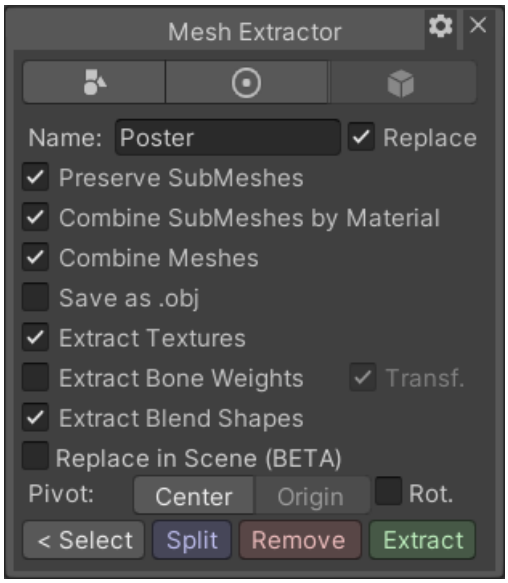


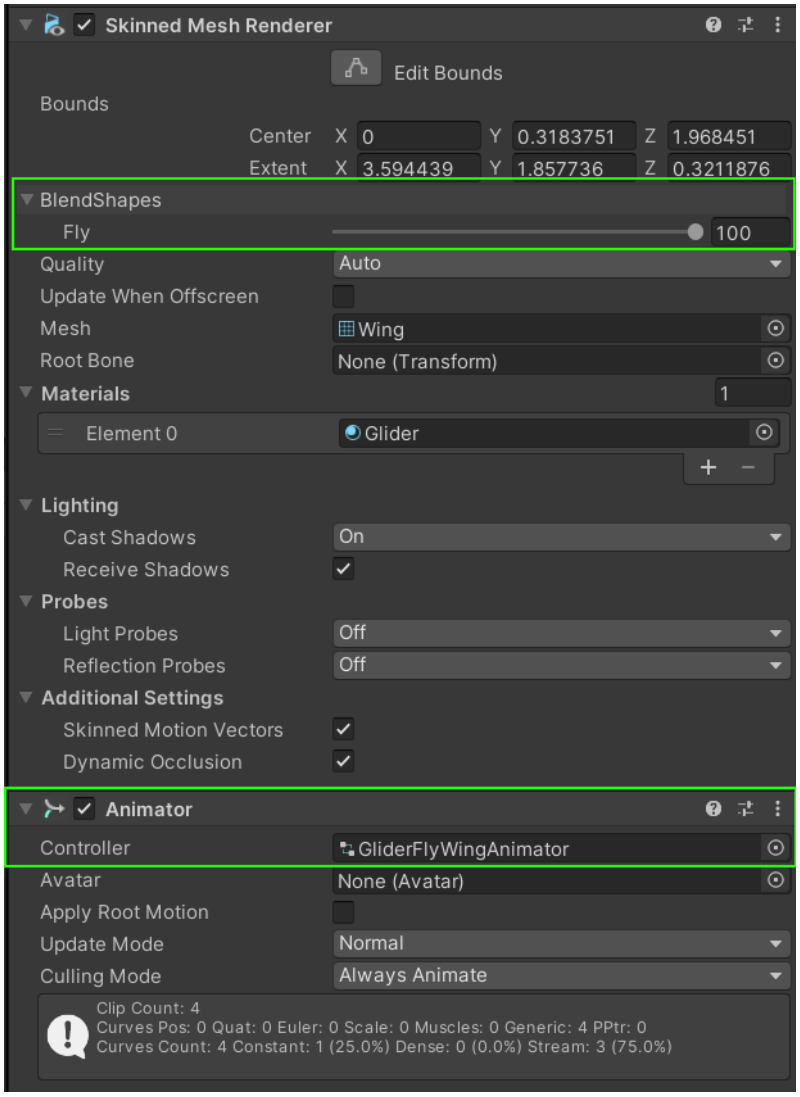
(The image is blurry on purpose to not reveal a third party publishers texture).

Extract Bone Weights: Extract bone weights (more on that below).



Extract Blend Shapes: Extract blend shapes. If your renderer has an Animator component attached then it will also copy that.





Pivot Center: Recenters the pivot in the middle of all selected triangles.

Pivot Origin: Rests the pivot to match the original mesh. This is especially handy if you want to „split" your mesh.

Pivot „Rot" (Rotation): Forces the pivot to align with the transform rotation. This is especially handy if you want to „split" your mesh.

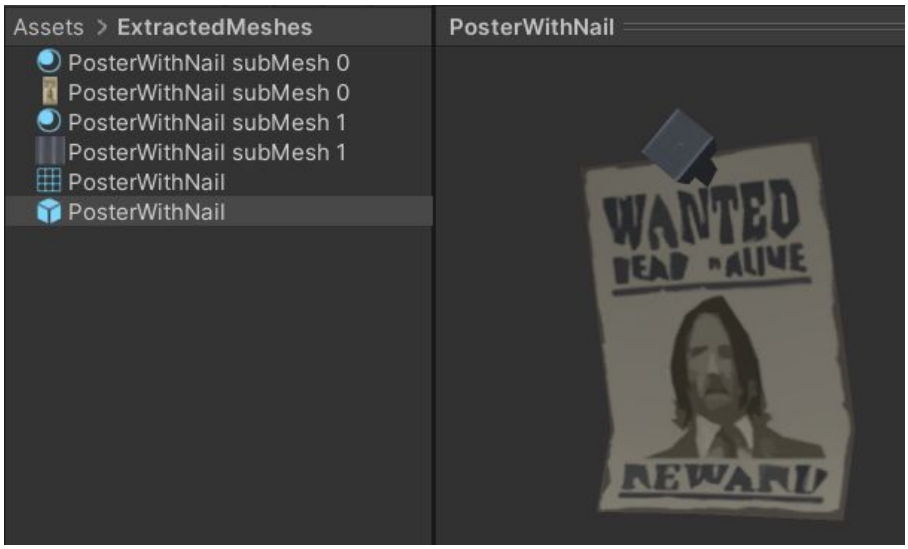
Custom Pivot Position: Please notice the move gimzo in the scene. You can use it to **adjust the Pivot position to your needs.**

🤫 HINT: You can hold down the V key to snap the pivot to a vertex.

Replace in Scene
Enabling this will make the tool REPLACE the existing mesh in the scene with the newly generated mesh. In case of „split" it will even place an additional prefab in the scene (the part that was split off of the mesh).

WARNING: This feature is rather new and thus UNDO / REDO is not fully supported. You can usually undo just fine but in some cases it may not work as intended. Please make a backup of your scene before using this.

Extract
This is what you will end up with if you use the „extract" button. A newly generated prefab with all the resources that are needed (mesh, textures, materials).



HINT: If you want your new mesh to re-use the existing materials and textures then untick the „Extract Textures" option.

Remove
Please notice that even if the button says „Remove" the result of this action is actually a NEW mesh and you may not see any result in the scene (check Assets/ExtractedMeshes for the result).

HINT: You can make it look like in-place editing by enabling the „Replace In Scene" option.

NOTICE: If you have nested objects then make sure that you only ever select one during removal and that each new object has a new name set or else the meshes from the previous action my be overwritten.

Split

Please notice that even if the button says „Split" the result of this action are actually two NEW meshes and you may not see any change in the scene (check Assets/ExtractedMeshes for the results).

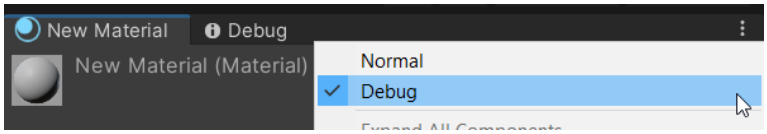
NOTICE: Mesh splitting does NOT generate new sub-meshes. If you are interested in that then please refer to the Polygon Material Painter & Sub Mesh Editor asset.

Extracting Textures

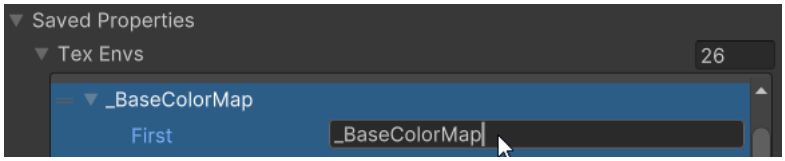
Textures are connected to 3D models through materials. Each material uses a shader and within that shader are slots (properties) which can take a texture. These slots are named. A common name used by most Unity Standard shaders is „_BaseMap“ for the main albedo texture.

Since every shader developer can pick these names freely it is sometimes confusing. The tool iterates through all the texture properties and copies them. It will also try to trim the texture down to the smallest possible size.

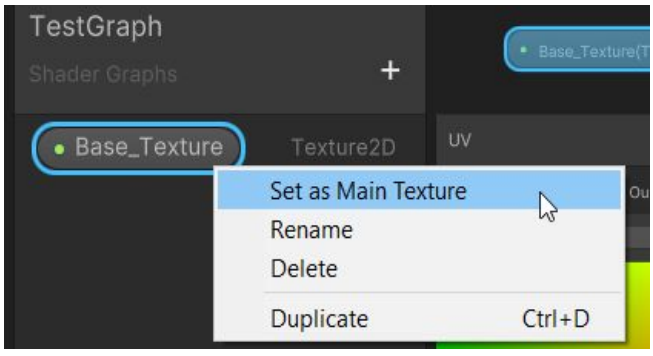
If you are curious and you are not sure how to find out which property names your custom shader is using then make a new material and assign the shader to it. Click on the material and in the inspector change the mode to „Debug“.



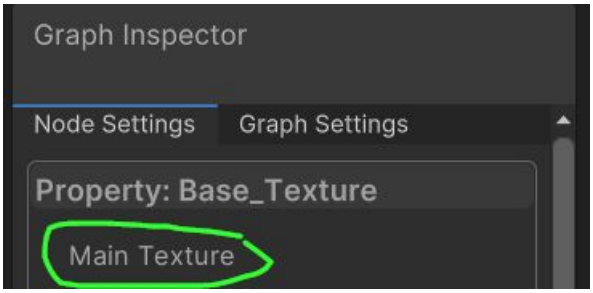
If you scroll down to the „Saved Properties“ List you will find the property names listed right there.



HINT: For shader graph shaders you can set the main (albedo) texture if you set one of the texture properties in the graph as the „Main Texture“, like this:



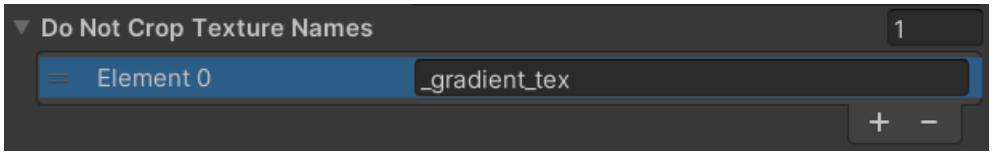
Sadly Unity does not do this automatically and not many shader graph creators think of it (I don't blame them, it's a pretty well hidden feature).



PLEASE NOTICE:
The extractor can not know whether a texture should be cropped or not. It usually assumes all textures should be cropped based on the extracted mesh UVs.

Why is this important?
Well, your shader may use a gradient texture for sampling colors and that texture should never be cropped because it's not UV dependent (i.e. all pixels of the texture are used all the time).

To support this there is a „DoNotCropTextureNames" list in the settings:

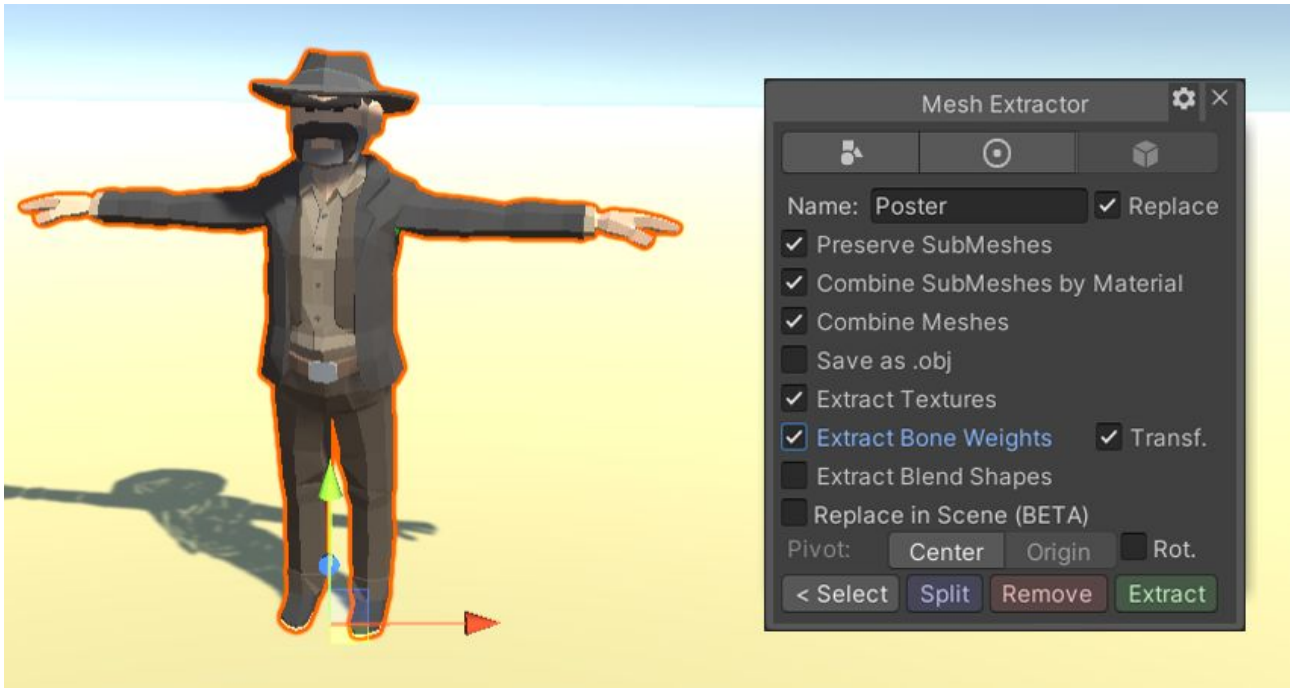


Texture property names listed there will never be cropped.
I know it's a bit cumbersome to maintain this list but these should be the rare exception anyways.

Bones / Rigs / Armatures / Blend Shapes

Exporting meshes with bones

To export bone weights tick the „Extract Bone Weights“ box.



By default the „Transforms“ export is also checked. It exports the bones transforms and puts them into the prefab (usually under a game object named „Root“).

However if you wish to only export the mesh and boneData then you can untick it. Just be aware that it will set the bone transforms to null to avoid 'Bones do not match bindpose' errors. You will have to link the renderer to new bones manually afterwards (or use the BoneDataResolver). More on that in the next „Things you should know about bones in Unity“ section.

Things you should know about bones in Unity (aka „Why we need that BoneData asset.“)

One important thing to note about bone information in Unity is that bone information is NOT only stored in the mesh but in multiple locations. Bones require three things to work:

1) **Bone weights** for each vertex (stored in the mesh). They tell each vertex how much influence a bones has on it.

2) **Transforms** to control the movement of the vertices (stored as objects in the hierarchy). That's the thing you will probably think of if you hear the word „bone“.

3) **Something to connect the bone weights to the transforms**.
Now here it becomes tricky since that „connection“ information is kinda divided. It is stored as a hidden list in the SkinnedMeshRenderer. - You look confused. Let me explain.

To constitute a bone you need two things (A and B):

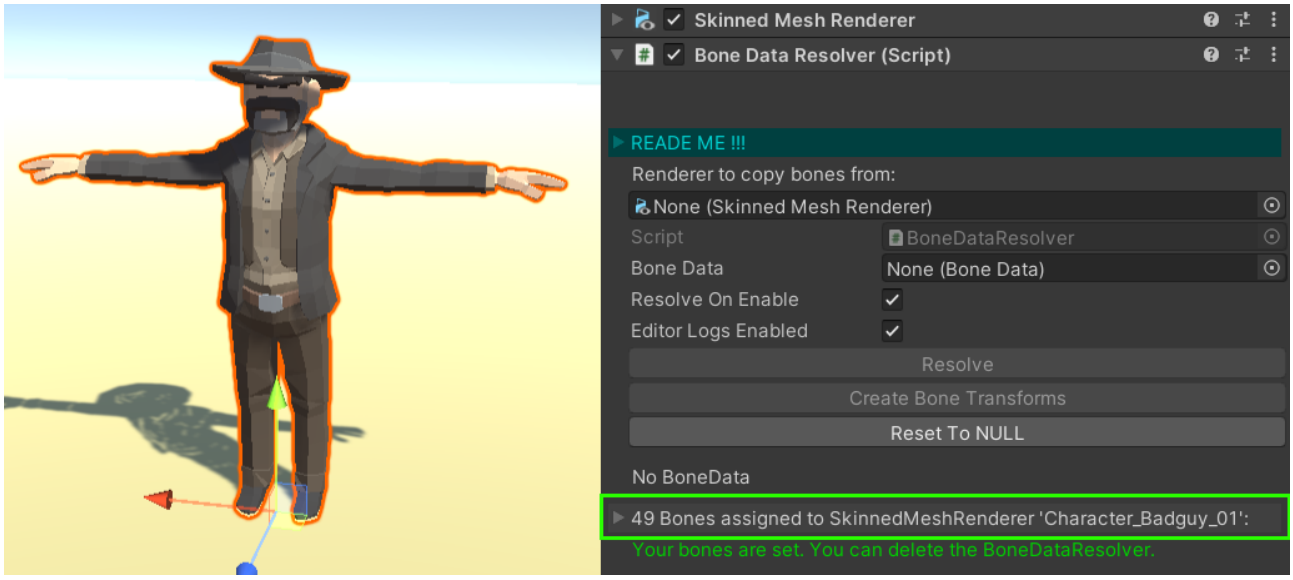
A) First are the bind poses which are stored as a list (array) in the mesh.
Think of each bind pose as a thing (matrix) which connects multiple vertices to a transform (bone). It is resposible for transforming the vertex position depending on the bone position, rotation and scale. It's NOT the bone transform you see in the hierarchy, rather it's like a thread that's connecting the vertices to whichever positional data it is given.

B) The thing that most commonly is referred to as a „bone“ is just a transform (position, rotation and scale information). In Unity this resides in the hierarchy within the scene file (that's what you move, rotate and scale). If a transform is changed then the bind pose will take notice and it then moves the vertices accordingly.

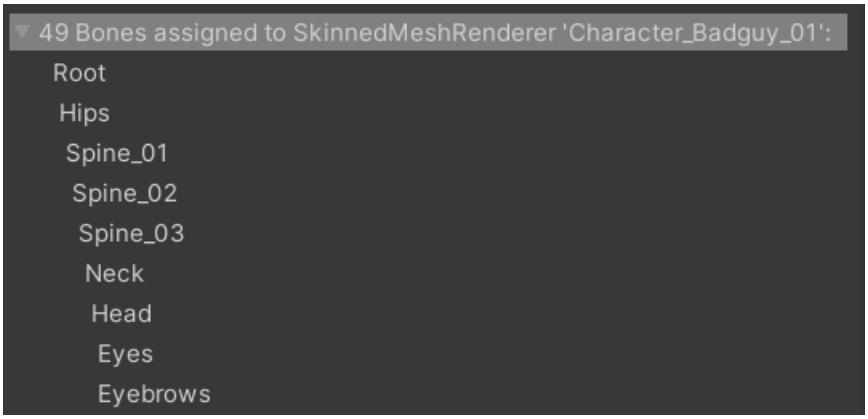
As mentioned before these bone transforms are NOT saved in the mesh. They are in the scene. How are they connected to the mesh?

Well, the SkinnedMeshRenderer has an internal list that tells it what scene transforms match to what bind poses. For some reason Unity decided to hide this list in the inspector. Sadly you do not even see it in the debug mode.

To deal with the hidden „transform → binde pose“ information I've created the „BoneDataResolver“ component which makes it visible in the inspector:

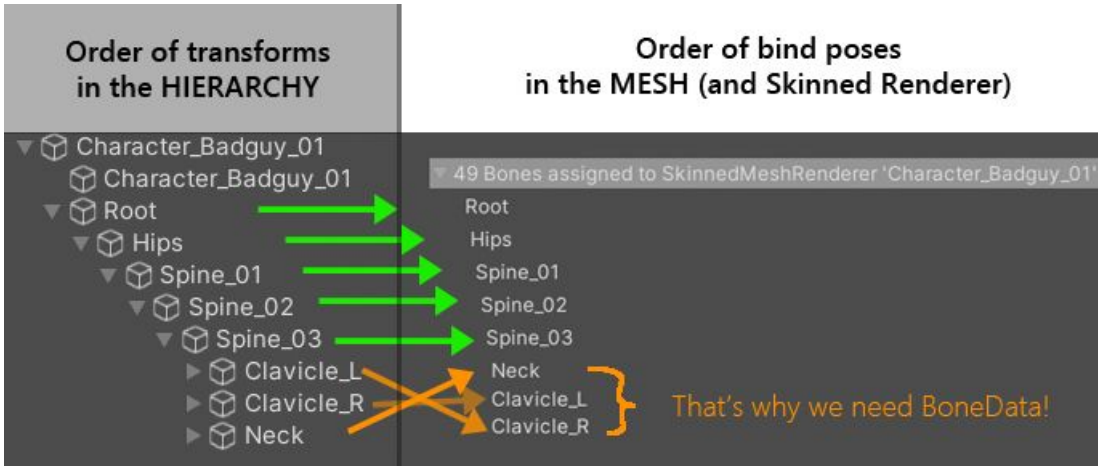


Unfolding the „.. Bones assigned to .“ section reveals the connections:



The ORDER of these is important. Often the order of the bone transforms in the hierarchy does NOT match the order in which the bind poses have been defined in the mesh.

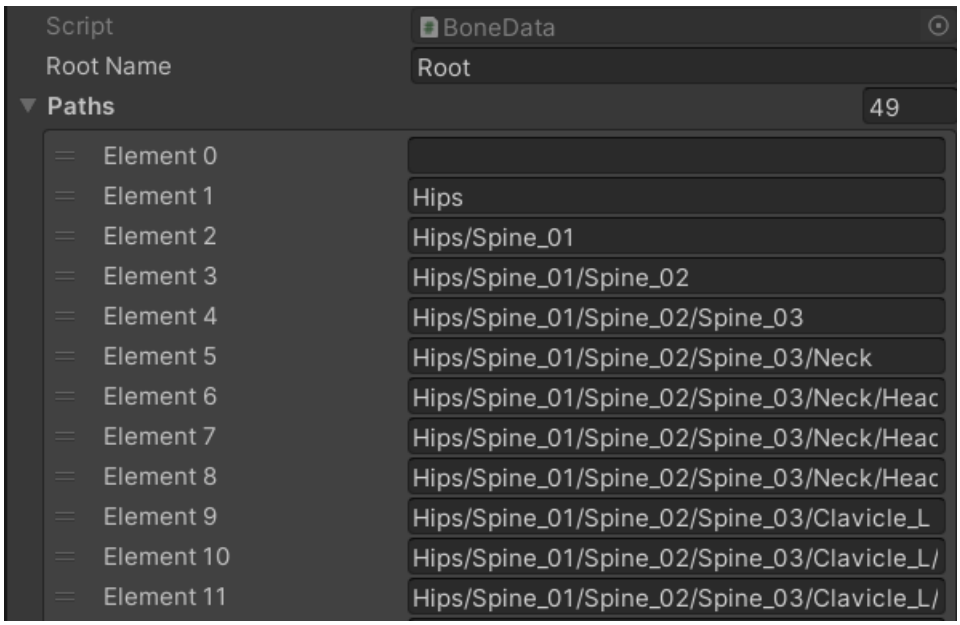
Like in this example:



Simply put: the left side of the image above is not enough information to restore the right side.

Now that is a problem. If we export only the mesh we can not restore the **order** information. Even if we have a copy of the hierarchy transforms in the scene we can not do it. It's hidden inside the SkinnedMeshRenderer.

Aaand that's why the **BoneData** asset exists. It stores the ORDER, NAME and HIERARCHY of the bone transforms. Like this:



With the BoneData we actually can restore the order information.
Btw.: That's what the „Resolve" button on the BoneDataResolver does for you. It searches in the hierarchy for the „Root" transform and if it finds one it uses that to recreate that hidden list in the renderer (effectively connecting the transforms to the mesh vertices).

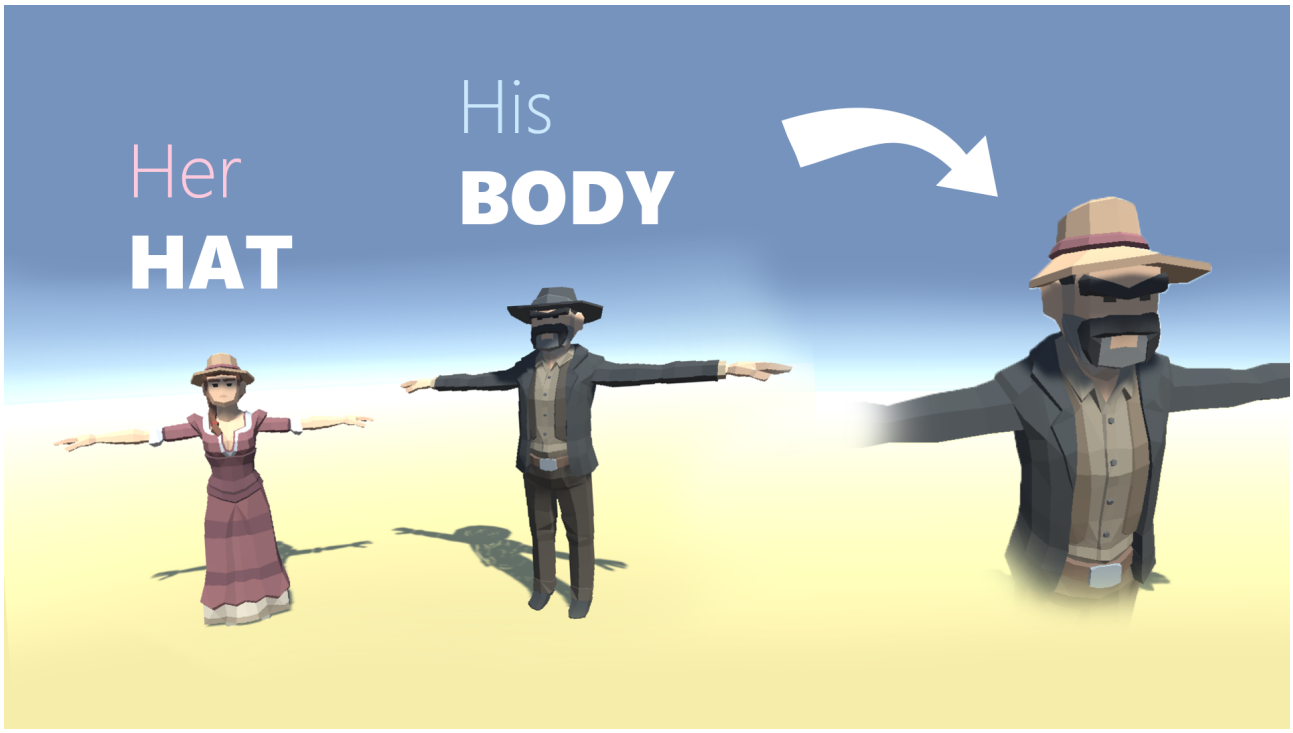
Once you have restore that hidden list you can delete the BoneDataResolver (unless you plan to change the list again, then I would keep it around).

Phew, thanks for reading it all.
I hope that was understandable.

Frequently Asked Questions (FAQ)

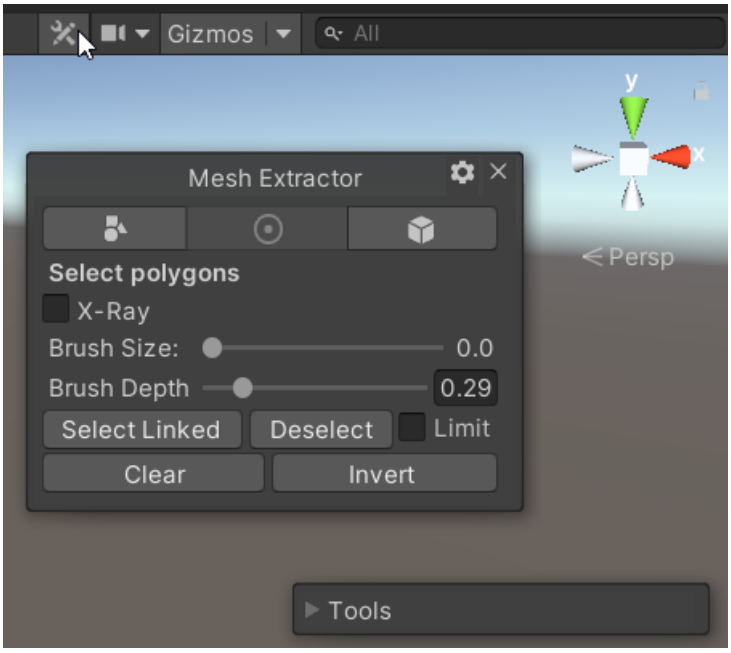
How to mix skinned meshes?

In this example video we will mix two skinned meshes from one of the Synty asset packs.
You can find a video of how to do it here: https://youtu.be/N6zGRhmtKNI



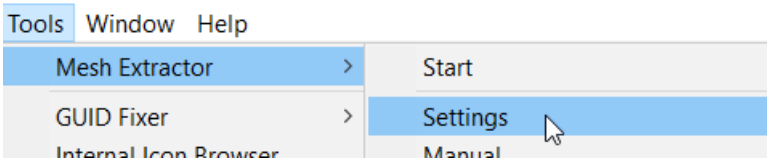
I can not move the tool window

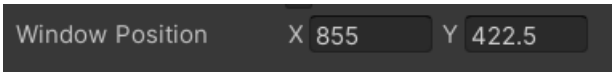
In Unity 2020 if you activate the „Tools" section in the scene view then this tools windows will always take precedence (which means the mesh tool window won't be movable). You can restore movability by disabling it.



If have moved the tool window off screen. How do I get it back?

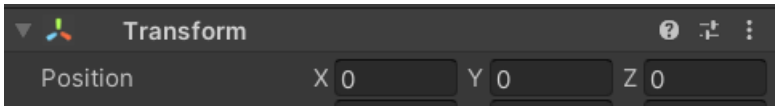
You can reset the window position to 0/0 in the settings.





The selection point of origin is kinda off. I have trouble focusing the correct triangle.

This may happen for models that are far off the origin of the scene. Please move your model to the origin of the scene (0/0/0).



The reason for this is that some ray casting calculations are done in world space and big distances will introduce floating point errors big enough to mess up the raycast. A distance of only 1000 units can already cause this.

HINT: If you drag in a model into an empty scene then Unity sometimes puts it at vast distances from the origin. Please check that it's positioned at 0/0/0 before using the extractor.

I want to export in another format like FBX or glTF.

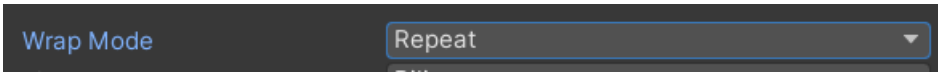
The tool currently only support exporting .OBJ. However by default it saves as the interal Unity format (.asset) and Unity has an official package to exporting these as FBX files: https://docs.unity3d.com/Packages/com.unity.formats.fbx@5.1/manual/index.html

From FBX you should be able to convert into pretty much any format using Blender.

The textures on the extracted mesh are stretched.

If the UV layout of the original mesh did extend beyond the 0/0 → 1/1 range then this may happen because in some cases the extractor forces the texture WRAP MODE to CLAMP.

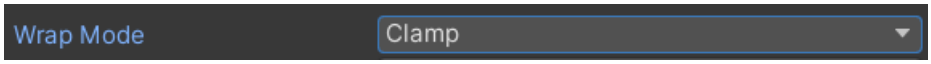
If you see this then select your new texture and set the WRAP MODE to REPEAT, like this:



Hint: Don't forget to hit apply to see the result.

My low poly colors do not match after extraction

It is knows to happen sometimes if the original texture of the mesh was set to WRAP MODE = REPEAT. Try setting the WRAP MODE of the new extracted texture to CLAMP instead, like this:



Hint: Don't forget to hit apply to see the result.

The generated materials do not have the right textures assigned despite the textures being created.

You are probably using a custom shader that uses custom shader property names. The extractor tries to find all of them but it could have failed. Please check out the „About extracting textures" section for more details.

My extracted mesh is bigger (or smaller) than the original.

This may happen if you have started extracting from a scaled mesh. The extractor will extract the mesh as it is shown in your scene. This means that if your object is scaled it will generate a mesh that is bigger or smaller than the original.

The reason for this is that the extractor actually supports multiple objects for extraction at the same time and if they all have different scales set then the results would differ. Thus it takes the size as is and assumes that's how you want to export.

Solution: Set your object(s) scale to 1/1/1 before extraction.

I don't see the results of „Split" or „Remove" in my scene?!?

Please make sure you have the „Replace in Scene" option turned ON.

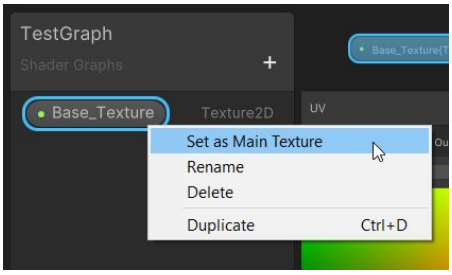
Even if the buttons are labelled „Split" and „Remove" they will actually only generate NEW meshes and prefabs under „Assets/ExtractedMeshes". Only if you enable „Replace in Scene" will these be placed in your scene.

The colors/textures on my model are somehow messed up.

My textures are not sliced property. It does not generate a new texture.

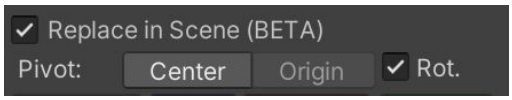
Please check the textures on the materials. Especially if you use a custom shader with uncommon shader property names. The tool tries to assign all the shader texture properties (see „Extracting Textures" section above). The tool might have guessed the texture slots wrong and the material may end up still using the old textures despite having generated new UVs.)

For shader graph shaders you can fix this for the main material in the shader graph by making the texture the main texture like this:



The mesh is not inserted in the correct position in the scene if I use the „Replace in scene“ option?

To match the rotation and position of the original mesh with the „replace“ feature please switch the pivot mode to „Origin“ and enable the rotation match (Rot. Checkbox), like this:



Once you have done that it should match your existing object in the scene.