One Color Palette

(Absolute Texture Atlasing)

Contents

Introduction

One Color Palette was initially designed for Smitesoft “Colorize” tool, Colorize tool is able to fork out many textures and their material their color palette and form new palettes with user custom modifications. However, making many textures has its draw backs. These drawbacks include an increase in application size, and where new materials are created, this can also impact processing.

As a result we created this asset which combines all textures and materials into a single Texture/Material. This has many benefits. In addition to this we have added many other features to further compress and optimise the resulting texture palette.

Benefits

- Reduce application size by reduce the number of texture
- Reduce application size by compressing large textures into the smallest possible texture
- Compress the color palette by providing the option to overlap exact colors
- A single material for all selected models of interest: this results in possible dynamic batching
- Dynamic batching improves performance significantly where applicable
- Provides options to fork out modes from their shared materials but at the same time keep them on the same texture (more details in “Preferences” section)

Limitations

- Only works on Color Palette textures
- Only works on models that are not using any patterned color / gradient colors
- Only considers the Main Texture (no metallic, no reflection, no emissions are considered for this version)

Dynamic Batching

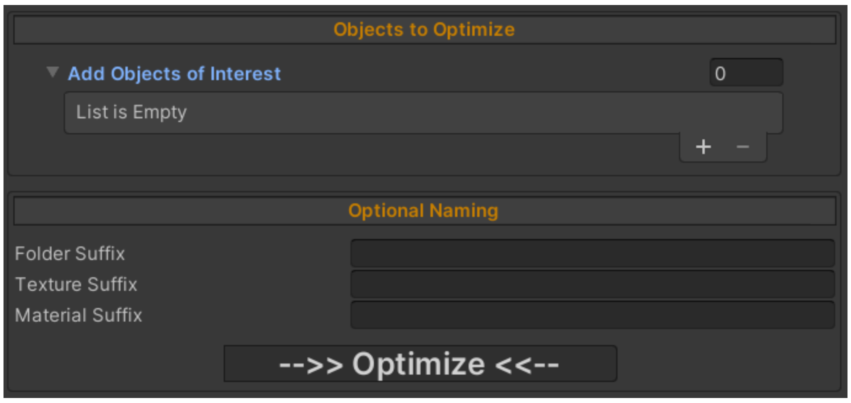
- Dynamic batching reduces improves performance, dynamic batching has to be enabled from player settings in unity to take effect.
- Unity can automatically batch moving GameObjects into the same draw call if they share the same Material and fulfil other criteria. Dynamic batching is done automatically and does not require any additional effort on your side.
- Batching dynamic GameObjects has certain overhead per vertex, so batching is applied only to Meshes containing no more than 900 vertex attributes, and no more than 300 vertices.
- If your Shader
- is using Vertex Position, Normal and single UV, then you can batch up to 300 verts, while if your Shader is using Vertex Position, Normal, UV0, UV1 and Tangent, then only 180 verts.
- These are the basics but there more conditions that be found here

Getting Started (Basics)

Drag and drop "Palette Fusion Manager" from this directory into your Scene;
Assets/Smitesoft/OneColorPalette/Prefabs/Palette Fusion Manager.prefab

You may notice that the prefab unpacks itself (this is by design)

Selecting "Palette Fusion Manager" will present you with a window that contains this:

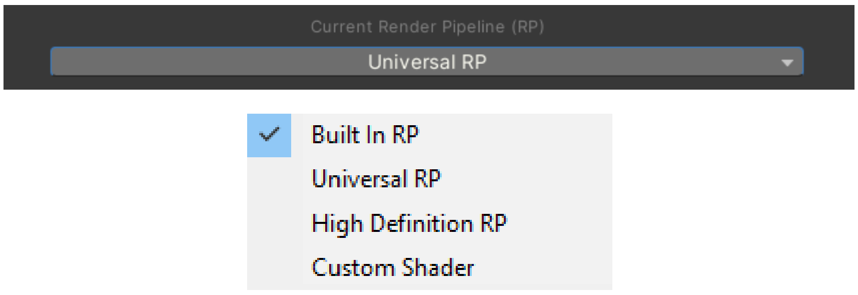


Drag and drop your objects/models of which you intended to fuse their palette into the "Add objects of Interest" List.

Click Optimize and just like that you are done.

For more advance features, please read full documentation or watch the video guide that will periodically be updated and can be found on Palette Fusion Asset Store Page!

Pipeline Setting

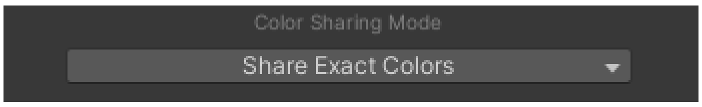


You must remember to select the correct render pipeline that your unity is using. Selecting the wrong pipeline can break you model and you may need to reimport it or manually reassigned the textures back to the model.

Note: This tool also has an additional benefit of correcting the Shader-renderer-pipeline-mismatch. This is usually identifiable by its pink representation. See "Shader Conversion"

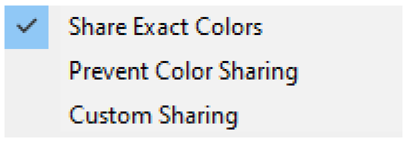
(This is only a side benefit of this tool and not the main focus.)

Compression Settings



Color Sharing Mode

Share Exact Colors



Share Exact Colors:

This is the most efficient mode and is the most common use case. In this mode models that are using any color that is the same with other models, the UVs will share the color in the newly created color palette. This allows for further compression of the texture

Prevent Color Sharing:

This feature is intended for use with our Colorize tool, every model included in the optimization will have its own color UV coordinates that do not overlap with any other model even if the colors are exactly the same. This means, when these models are modified using the colorize tool, no other model will be effected by the changes and at the same time, you still benefit from sharing a compressed palette.

Note: this mode uses color merge prevention technology, this technology allows the differentiation of exactly similar colors from one model to another. But this does have it limits, and that limit is approximately 125 color merge preventions per color.

Note: Palette-fusion-Pro should have virtually no limit.

Custom Sharing:

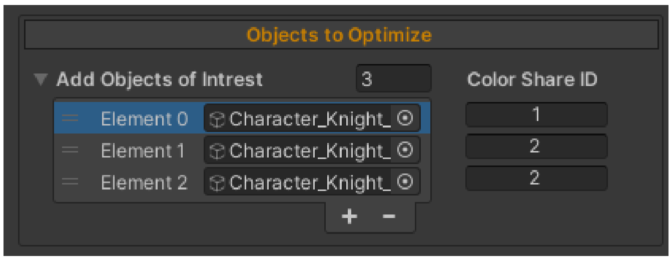
This mode is a hybrid mode of the above two, you can specify which models will share colors by giving them the same ID value, and models with the exact same color value will share colors with those who have the same ID. If all models have the exact same ID, that would be the exact same thing as having "Share Exact Colors" mode enabled.

This feature is useful for those who want to theme objects similarly. (More examples are provided within the colorize tool)

Note: this mode uses color merge prevention technology, this technology allows the differentiation of exactly similar colors from one model to another. But this does have it limits, and that limit is approximately 125 color merge preventions per color.

Note: Palette-fusion-Pro should have virtually no limit.

Example:



Objects to Optimize

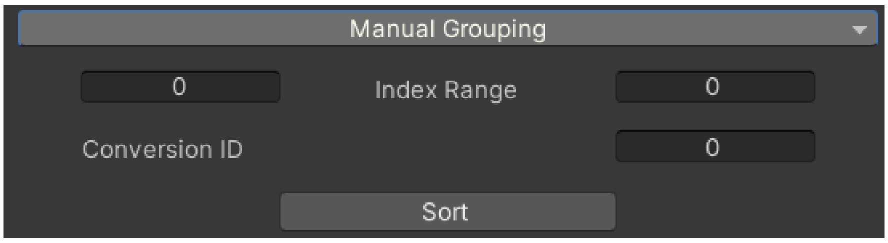
Add Objects of Intrest 3 — Color Share ID

		Color Share ID
Element 0	Character_Knight_	1
Element 1	Character_Knight_	2
Element 2	Character_Knight_	2

In this example, Element 1 and Element 2 will share the same UV coordinates for any colours that are identical. Meaning when you use the colorize tool to modifies those colors, both models (Element-1 and Element-2) will be effected by the changes

Manual Grouping:

This Features allows you to quickly assign groups within the specified range instead of doing them one at a time.



Manual Grouping

0 Index Range 0

Conversion ID 0

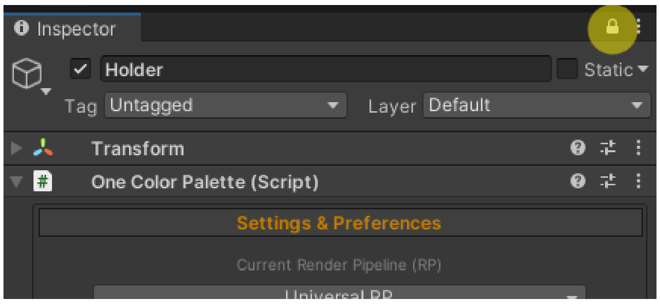
Auto Grouping:

Will increment the group ID with each addition. This means every time you add a group of objects all of them will be assign to a new shared group. This is also the recommended setting.

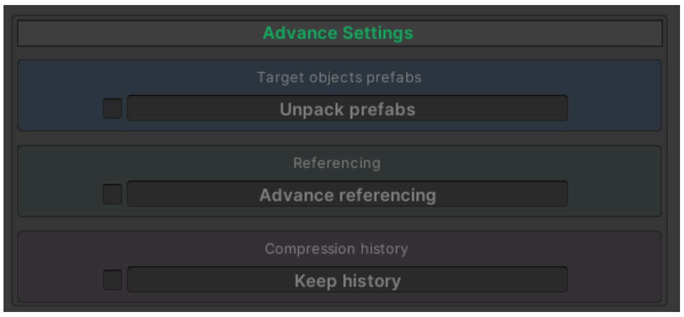
Mass optimization

To compress and optimize your color pallet, you must include all models of interest that will use the newly created color palette. You can do this by dragging them 1 by 1 in the inspector onto the "Objects to Optimize" array. But a better way would be to select all objects of interest and drag them all at the same time onto the array. However, to do this, you must lock the inspector.

Example: (Highlighted in Yellow)



Advance Settings



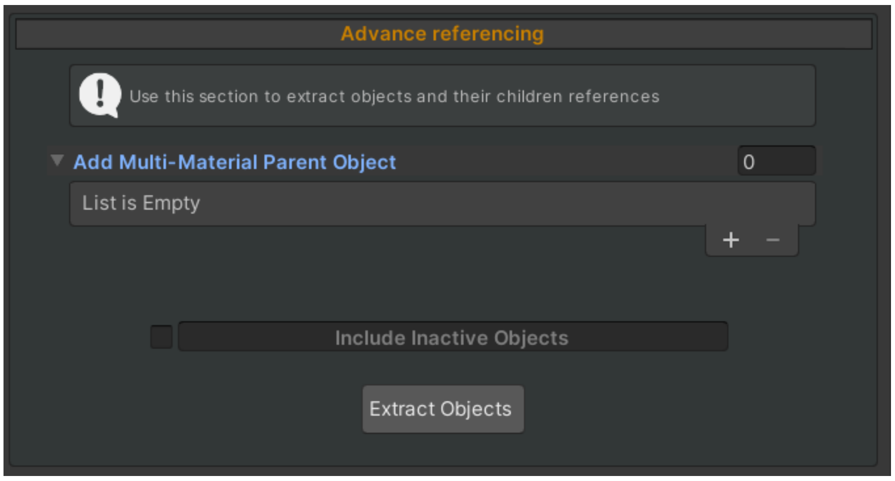
Unpack Target Prefabs (Experimental)

Scene Prefabs have a significant overhead associated with them, this is partly because they register any change applied to them for which you can change to override the original asset prefab with. This can cause editor lag and increase the processing time significantly especially when you are mass optimizing many model/prefabs at the same time.

By selecting "Unpack Prefabs", any object prefab included from your scene to be compressed, will be unpacked completely before the compression takes place. This should "theoretically, reduce the processing speed, and in some extreme cases, it could also prevent unity from crashing or running out of memory before completing a mass compression.

Advance Referencing

Enabling “Advance referencing” gives you access to this tab:


Advance referencing

Use this section to extract objects and their children references

Add Multi-Material Parent Object 0

List is Empty

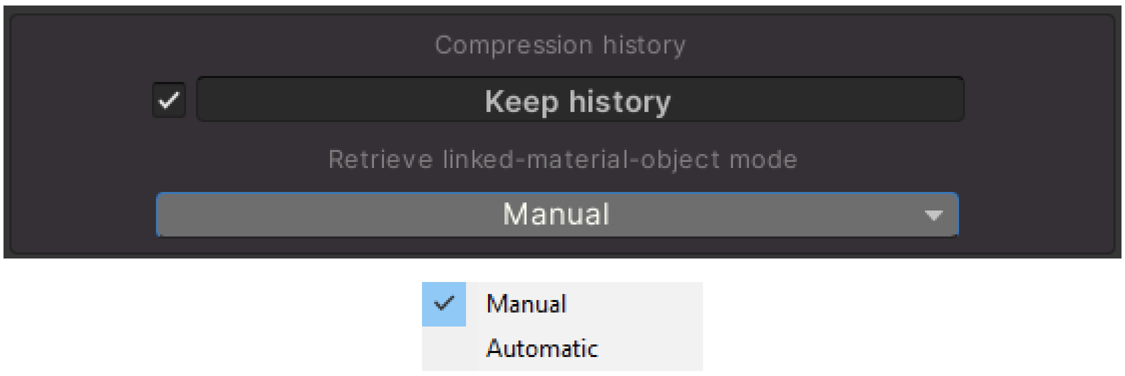
Some models consist of many materials, this is done by having many objects assigned to the model each with its own mesh, and this is usually the case with modular characters. This feature, allows you to add all of the materials/textures that are assigned to the children of the referenced objects. All you have to do is reference the parent object/objects, followed by pressing the “Extract Object” button, this will extract the objects and add them to the “Objects to Optimize” array. This will save you the hassle of doing this manually which can get very exhausting especially if the model is divided into many modules.

This feature only extract compatible objects, those that are not compatible will not be extracted, else they will be omitted from the fusion/optimization process.

Note: This asset is not currently compatible with a mesh that has many materials assigned to it, which is not the same thing as the example specified above. Only the Main material and the Main texture are considered in palette fusion.

Compression History (Experimental)

Enabling "Keep History" provides you with two options:


Compression history

✓ Keep history

Retrieve linked-material-object mode

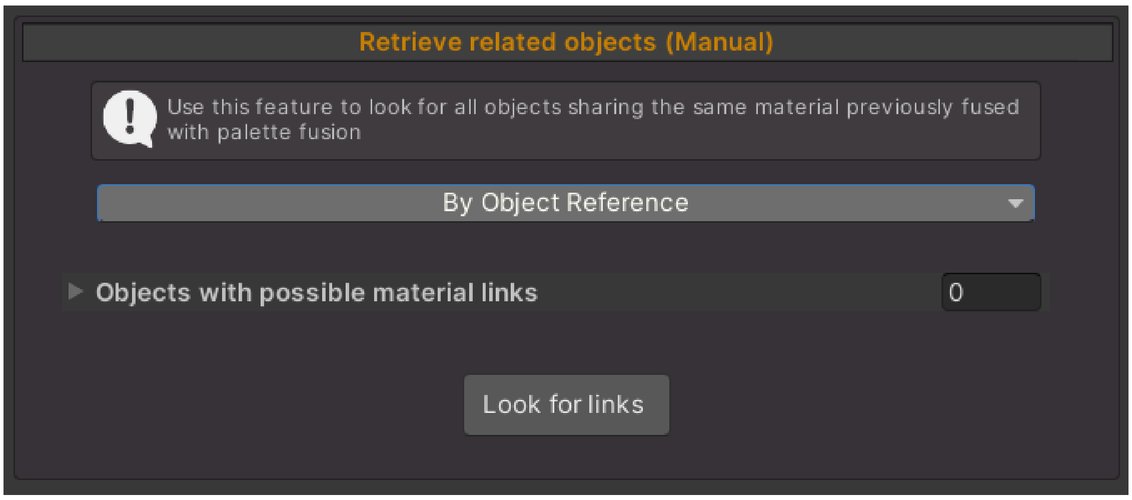
Manual

✓ Manual

Automatic

Compression History - Manual

Manual mode will give you access to this tab:


Retrieve related objects (Manual)

Use this feature to look for all objects sharing the same material previously fused with palette fusion

By Object Reference

Objects with possible material links 0

Look for links

Manual mode provides you with retrieval options:

1- By Object Reference:

   Look for objects through the history archive that share the same material as the materials found in the objects your reference, assuming those materials where generated with this palette fusion asset while "Keep history" was enabled. If any objects are found to share materials, they will be added to "Objects to Optimize" list

2- By Material Reference:

   Look for objects through the history archive that share the same material as the materials referenced, assuming those materials where generated with this palette fusion asset while "Keep history" was enabled. If any objects are found to share materials, they will be added to "Objects to Optimize" list

Compression History – Automatic

Unlike manual mode, this mode automatically cross-references with the objects already included in the “Objects to Optimize” list. If any objects are found to share materials, they will be added to “Objects to Optimize” list before compression commences. This is automatic, so all you have to do is initiate the compression by clicking the “Optimize” button.

Compression History – When to use

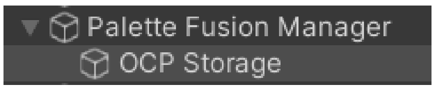
for most use cases, you want this feature enabled and left on automatic, unless you have specific needs. This feature was added in version-2.0 due to user demand. This feature is very helpful when you plan to include many models to a single texture, and then you plan to add more models to the same group, however due to your project size, it may be difficult to collect the same objects you merged in the past, and thus this feature will do this for you automatically.

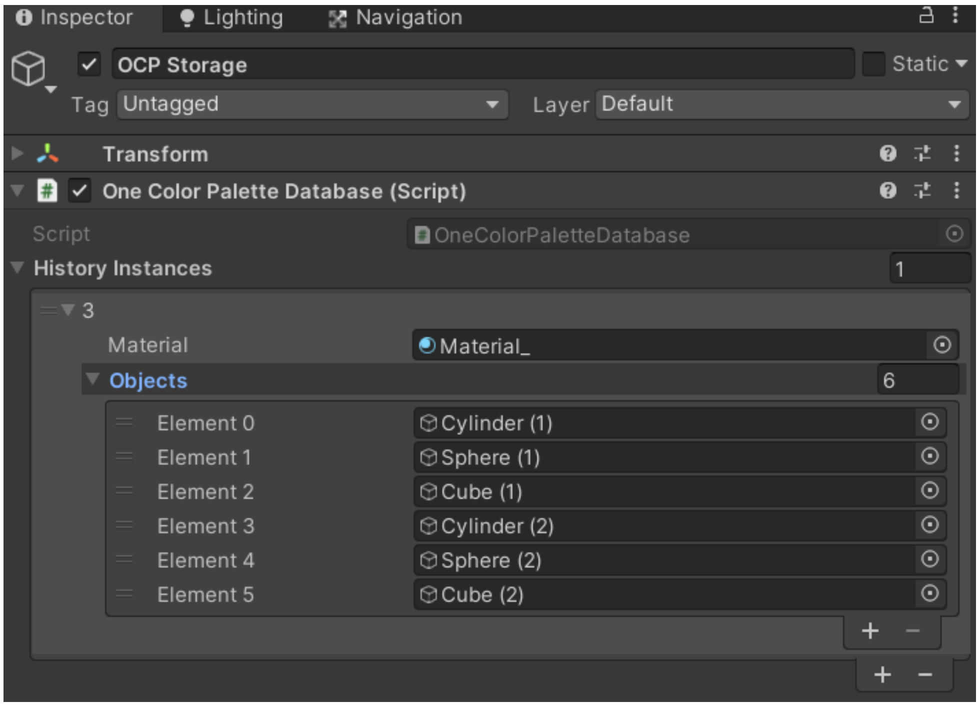
Compression History – Limitation

Compression History database is local to the scene, thus we recommend that you do all your palette fusion in a single scene, followed by creating prefabs out of those items which you can then use in any scene. We plan on overcoming this limitation if proven feasible.

Compression History – Database

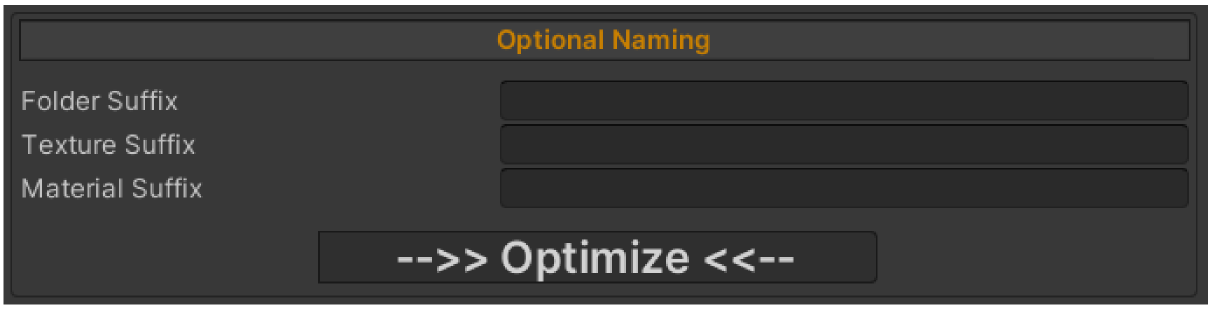
Once Keep history is enabled, and optimization made with this tool will be recorded, and a local database will be instantiated (OCP Storage):





OCP Storage
Static
Tag Untagged
Layer Default
Transform
One Color Palette Database (Script)
Script OneColorPaletteDatabase
History Instances 1
3
Material Material_
Objects 6
Element 0 Cylinder (1)
Element 1 Sphere (1)
Element 2 Cube (1)
Element 3 Cylinder (2)
Element 4 Sphere (2)
Element 5 Cube (2)

Saving & Optional settings

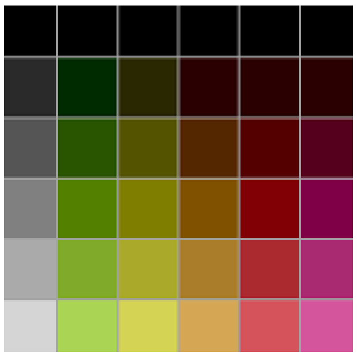


To save the new textures, meshes and material, simply click the “Optimize” button. A log will be informing you where the new materials will be created in the console.

You also have the option to add the suffix of the files and folders created, but this is completely optional.

Compatibility – Palette Type:

- Must use a color palette that uses solid colors, example:



Note: solid colours does not mean that the individual cells have to be square shaped, but that the UVs are using a single color per region.

- Color pallet must be power of twos (Pots, 2,4,8,16,32 etc.) meaning the above example is not a POT and would not work if it was the entire dimension of the color palette.
- Color Palette must have a square dimension
- Must not contain gradient colors. Example:

Note: gradient colors can contain millions of different colors per region, and that would be unmanageable and cause Unity to crash due to the heavy cost.

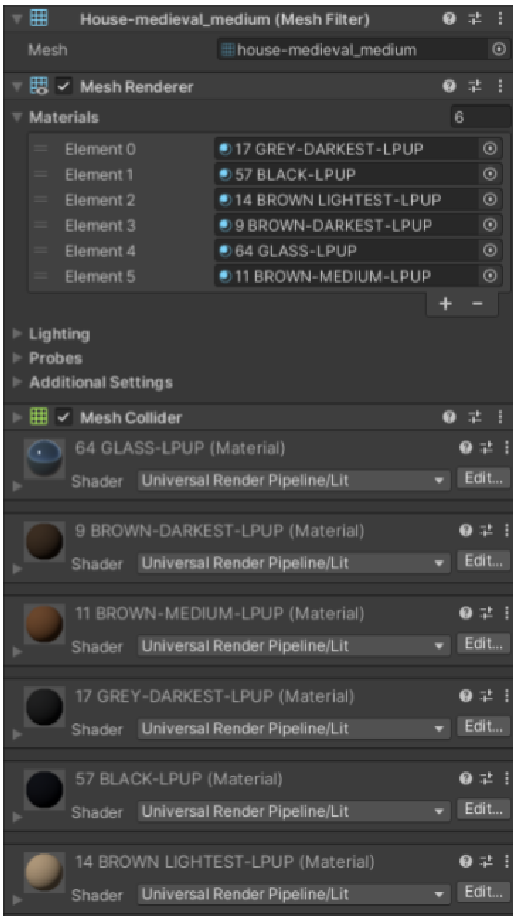
- Must not contain patterned colors (not gradient). Pattern colors are a hybrid between solid colors and gradient colors. The difference is, the amount of colours per region is much smaller and more manageable.
- If you do want to Compress such colors, you would have to use our Colorize tool (Sold separately) and merge the patterns into a single solid color before proceeding. This will however lose the pattern.

Example of Patterned colours (contains 7 solid colours)

Compatibility – Multi Material:

Example 1:

A single Mesh Renderer with several materials



House-medieval_medium (Mesh Filter)

Mesh: house-medieval_medium

Mesh Renderer

Materials: 6

Element 0	17 GREY-DARKEST-LPUP
Element 1	57 BLACK-LPUP
Element 2	14 BROWN LIGHTEST-LPUP
Element 3	9 BROWN-DARKEST-LPUP
Element 4	64 GLASS-LPUP
Element 5	11 BROWN-MEDIUM-LPUP

Mesh Collider

64 GLASS-LPUP (Material) — Shader: Universal Render Pipeline/Lit

9 BROWN-DARKEST-LPUP (Material) — Shader: Universal Render Pipeline/Lit

11 BROWN-MEDIUM-LPUP (Material) — Shader: Universal Render Pipeline/Lit

17 GREY-DARKEST-LPUP (Material) — Shader: Universal Render Pipeline/Lit

57 BLACK-LPUP (Material) — Shader: Universal Render Pipeline/Lit

14 BROWN LIGHTEST-LPUP (Material) — Shader: Universal Render Pipeline/Lit

A single object that has a Mesh Renderer with several materials assigned are **NOT-Compatible** with palette fusion.

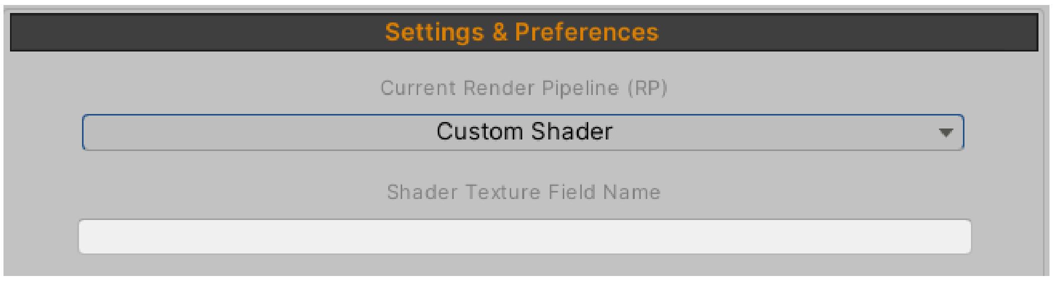
Example 2:

An object with multiple children objects each with its own Mesh-renderer/Skinned-mesh-renderer but each representing only 1 material are **Compatible** with palette fusion. Such as our modular armour packs scheduled to be released in 2023.

Shader Conversion

Custom Shaders (Experimental):

With Custom Shaders, this asset will convert the Shader to Build-InRP (BiRP) Standard Shader. This must be done in BiRP. You can then convert this into URP/HDRP.



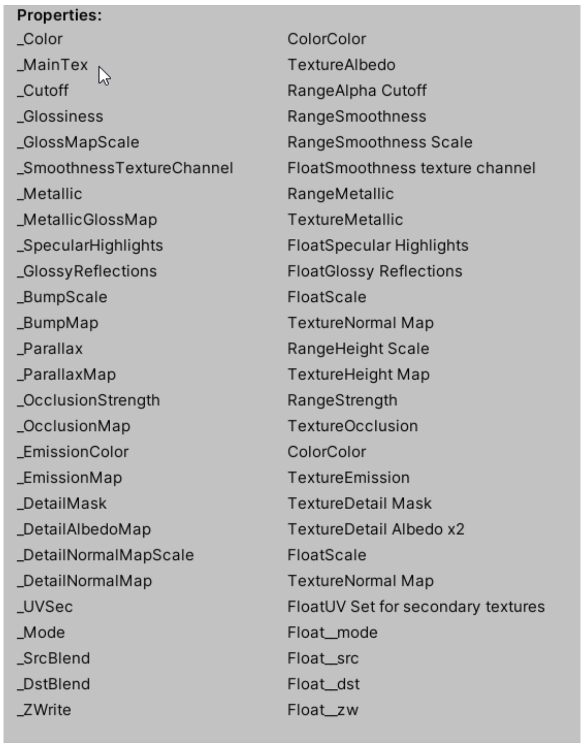
Settings & Preferences

Current Render Pipeline (RP)

Custom Shader

Shader Texture Field Name

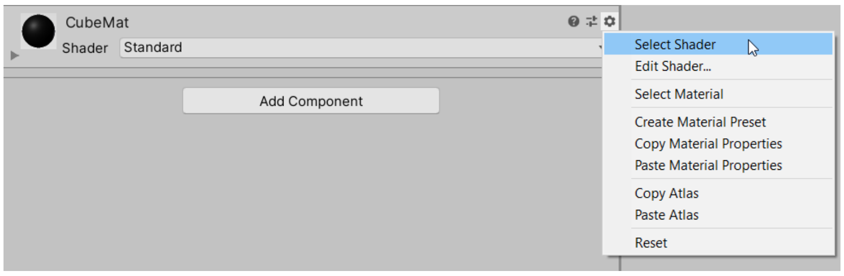
Don't Forget, that you need to include the custom shaders Texture Name
In the following Example, The name is _MainTex:



Properties:

_Color	ColorColor
_MainTex	TextureAlbedo
_Cutoff	RangeAlpha Cutoff
_Glossiness	RangeSmoothness
_GlossMapScale	RangeSmoothness Scale
_SmoothnessTextureChannel	FloatSmoothness texture channel
_Metallic	RangeMetallic
_MetallicGlossMap	TextureMetallic
_SpecularHighlights	FloatSpecular Highlights
_GlossyReflections	FloatGlossy Reflections
_BumpScale	FloatScale
_BumpMap	TextureNormal Map
_Parallax	RangeHeight Scale
_ParallaxMap	TextureHeight Map
_OcclusionStrength	RangeStrength
_OcclusionMap	TextureOcclusion
_EmissionColor	ColorColor
_EmissionMap	TextureEmission
_DetailMask	TextureDetail Mask
_DetailAlbedoMap	TextureDetail Albedo x2
_DetailNormalMapScale	FloatScale
_DetailNormalMap	TextureNormal Map
_UVSec	FloatUV Set for secondary textures
_Mode	Float_mode
_SrcBlend	Float_src
_DstBlend	Float_dst
_ZWrite	Float_zw

You can retrieve this list by clicking on the cogwheel of the Shader (Select Shader):



URP/HDRP/BiRP Shader Conversion:

These Shaders will automatically be converted based on the Pipeline selected. Assuming the Original Shader is not a custom Shader, if you wish to convert a custom Shader to URP/HDRP, first convert it into a BiRP as Explained above.

Support

Video guides available on this asset store page (Unity asset store)

Please contact us on Discord for support!

Bug Report

Please contact us on Discord to report bugs, we would really appreciate it!

Special thanks

My Mentor:

Steve Smith (Code Master)