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 any models and their material and textures from the 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.

This asset combines all textures and materials into a single Texture and Material. This has many benefits.

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)

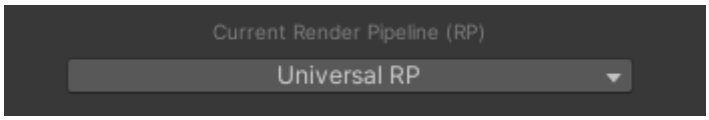
Limitations

- 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

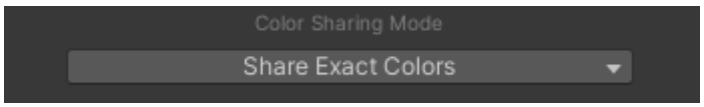
Settings



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 the benefit of correcting the render pipeline of compatible models (those that use color palettes) which would usually give the bright pink color on importing.

Preferences



There are 3 sharing modes

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 250 color merge preventions per color.

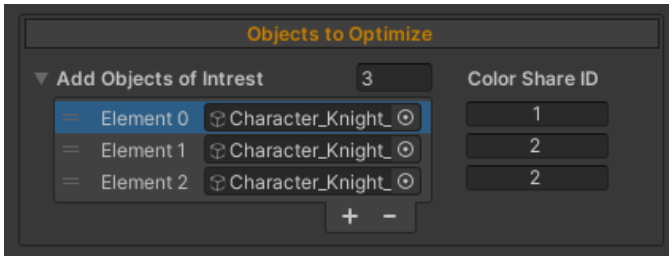
Custom Sharing:

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 250 color merge preventions per color.

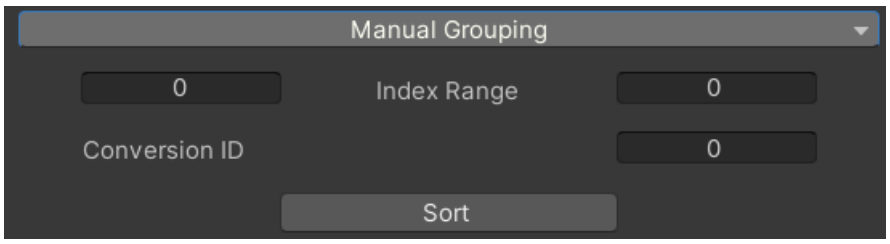
Example:



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.



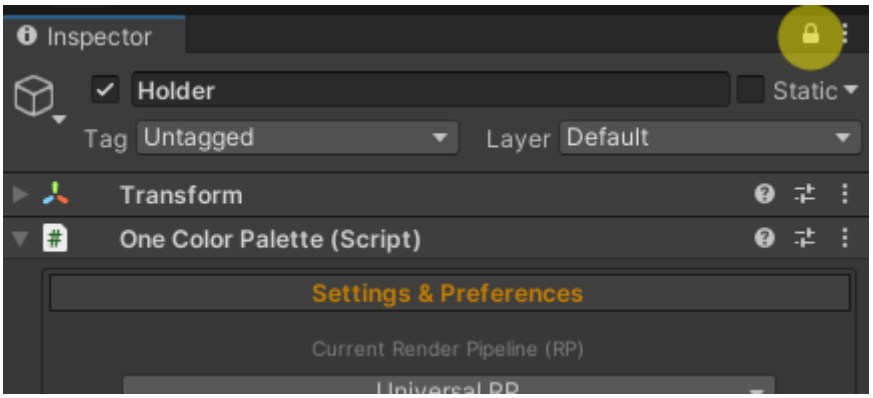
Auto Grouping:

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.

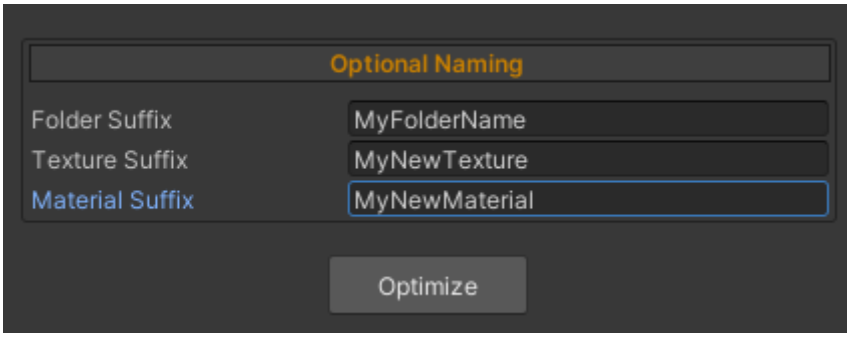
Example: (Highlighted in Yellow)



Multi Material Objects

Some Objects consist of many materials, 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.

Saving & Optional settings

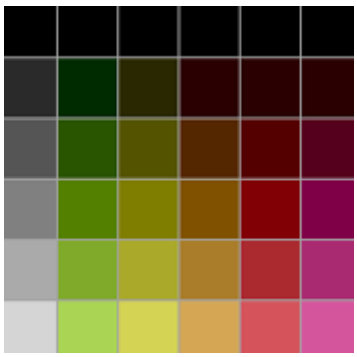


To save the new textures, meshes and material, simply click the “Optimize” button. A log will be informing you where the new materials will be created in the console.

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

Compatibility:

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



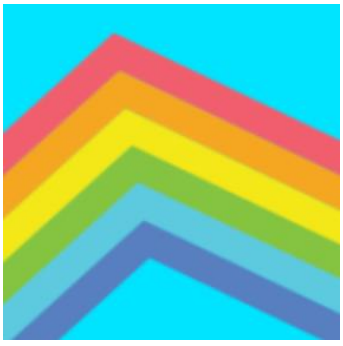
Note: solid colours does not mean that the individual cells have to be square shaped, but that the UVs are using a single color per region.

- 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.

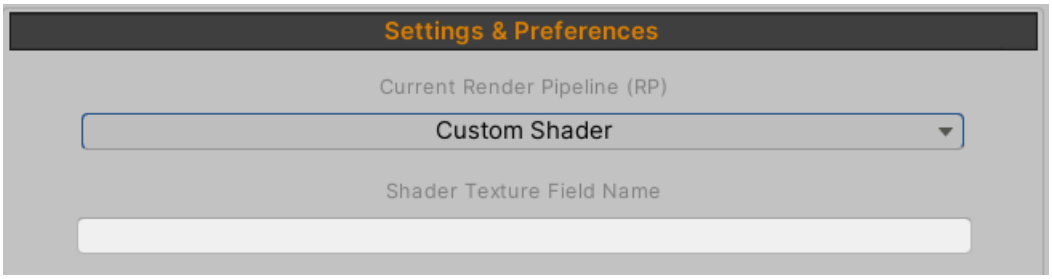
Example of Patterned colours (contains 7 solid colours)



Shader Conversion

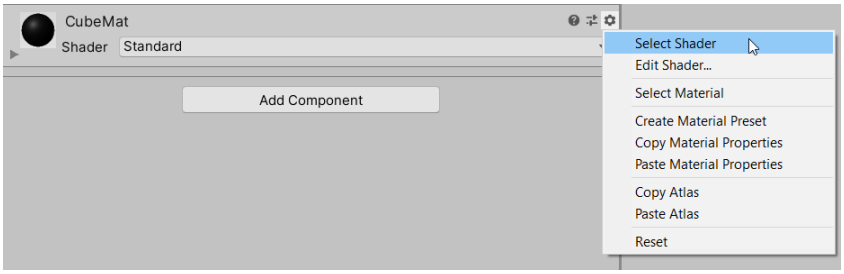
Custom Shaders:

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.



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

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!

Special thanks

My Mentors:

Steve Smith (Code Master)

Justin (Art Master)