

Draw Call Optimizer – Just One Click

Version 2.6.2

Table of Contents

| | |
|---|----|
| Introduction..... | |
| What makes this tool superior to others?..... | |
| How to Use | |
| Comparison of a complex scene..... | |
| Settings..... | |
| Expert Mode..... | |
| Mesh Combiner | |
| Atlas Baker..... | |
| Shader Mapper (BETA) | 7 |
| The Interface | 7 |
| Creating an atlas shader | 8 |
| Textures..... | 8 |
| Floats, Ranges, Vector, Color..... | 8 |
| Troubleshooting | 8 |
| Supported Pipelines and shaders | 9 |
| Classic Pipeline | 9 |
| Standard (Metallic Setup)..... | 9 |
| Standard (Specular Setup)..... | 9 |
| Autodesk Interactive (Roughness Setup)* | 9 |
| Legacy shaders | 9 |
| Universal Render Pipeline / High-Definition Render Pipeline | 9 |
| How to set up | 10 |
| Creating your shaders..... | 10 |
| Lit shader graph..... | 11 |
| Tex2DAtlas shader graph..... | 11 |
| Support..... | 11 |
| Changelog..... | 12 |



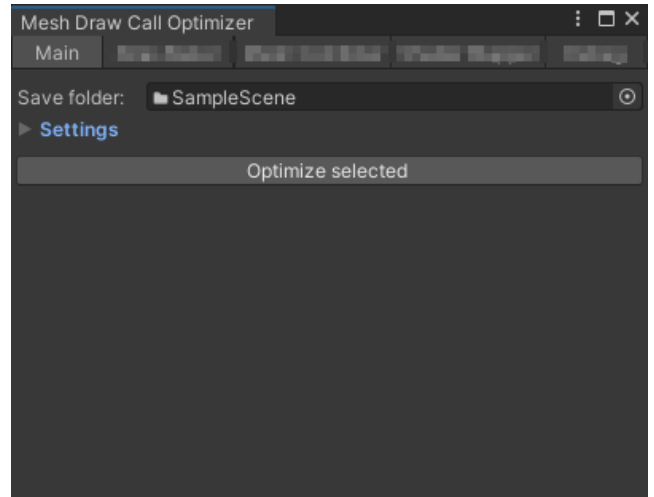
Introduction

The tool allows you to drastically reduce the amount of draw calls used in your scene by merging materials together. This is done by combining multiple techniques like mesh combining and texture atlas mapping.

I've built this tool because of the lag of tools out there which are easy to use and powerful at the same time.

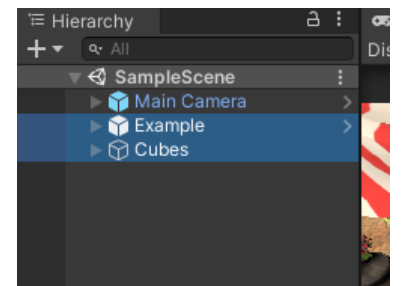
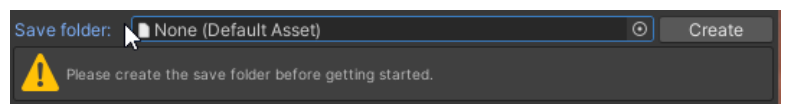
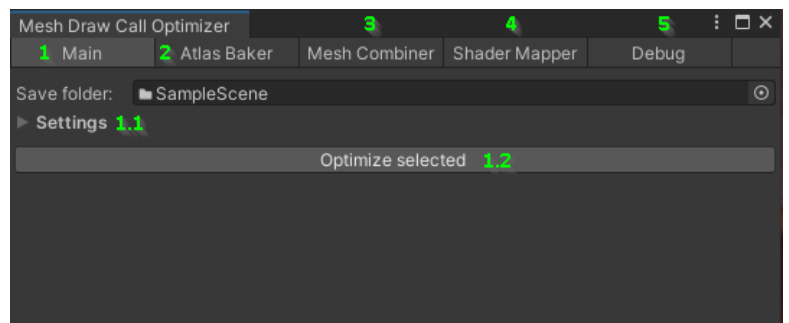
What makes this tool superior to others?

- one-click solution (which can be modified but does not need to).
- supports texture repeat and offset
- supports mip-maps correctly
- supports the Standard shader (parallax, emission, diffuse, normal, specular, cutout, etc; without details)
- supports all shader fields which are baked as well
- requires no lightmap UV recomputation (super-fast lightmap UV re-layouting)
- respects your colliders and merges them as well including simple colliders (boxes, capsules, ...)
- fully supports classic render pipeline
- input textures and materials stay untouched (your project assets won't be modified)
- BETA: URP and HDRP are supported as well but atlas shader graphs need to be adapted by hand (you need to replace texture samplers by a provided sub-graph)
- BETA: It creates on the fly atlas shaders for your (simple) custom shaders.



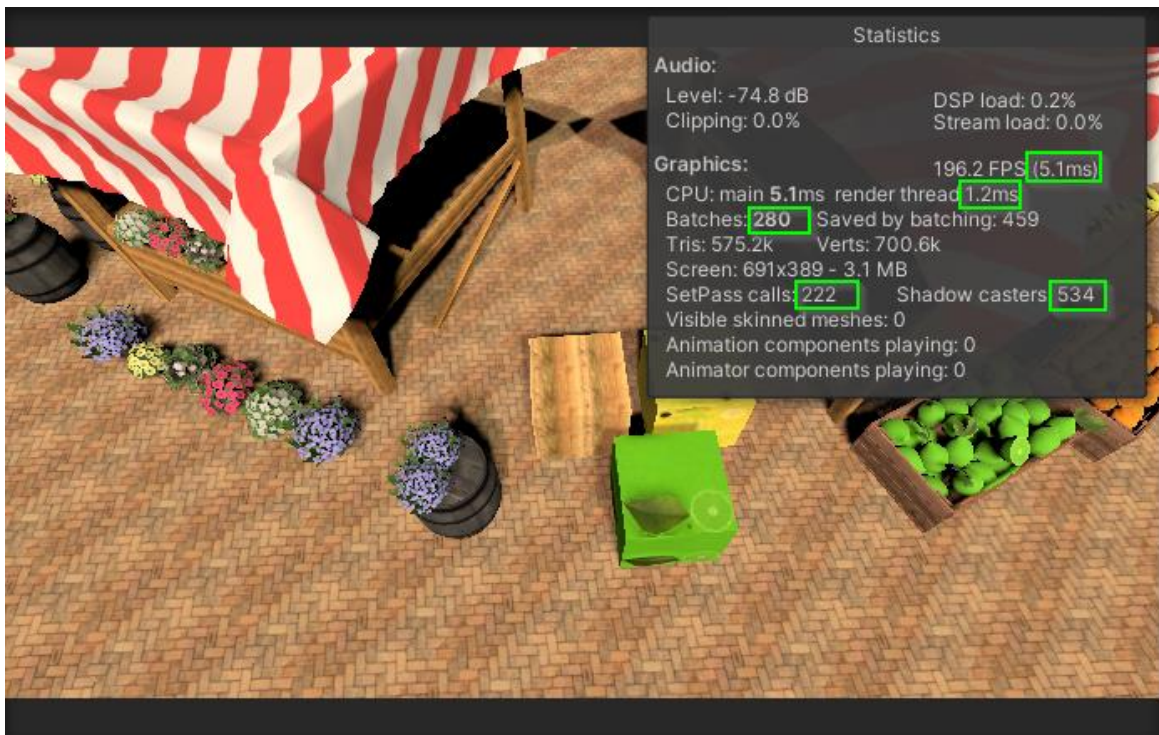
How to Use

- Optional: Open the SampleScene (**Scenes/Standard/SampleScene.unity**)
- Open the Tool by selecting from the menu **TOOLS > MESH DRAW CALL OPTIMIZER > OPEN...** or use the shortcut **CTRL + SHIFT + O**
- Select **MAIN** (1) if not selected
- When starting the tool for the first time click create, to create a target folder or pick one
- Select the gameobject(s) to optimize e.g. **EXAMPLE** and **CUBES**
- Click **OPTIMIZE SELECTED** (1.2)
- Done

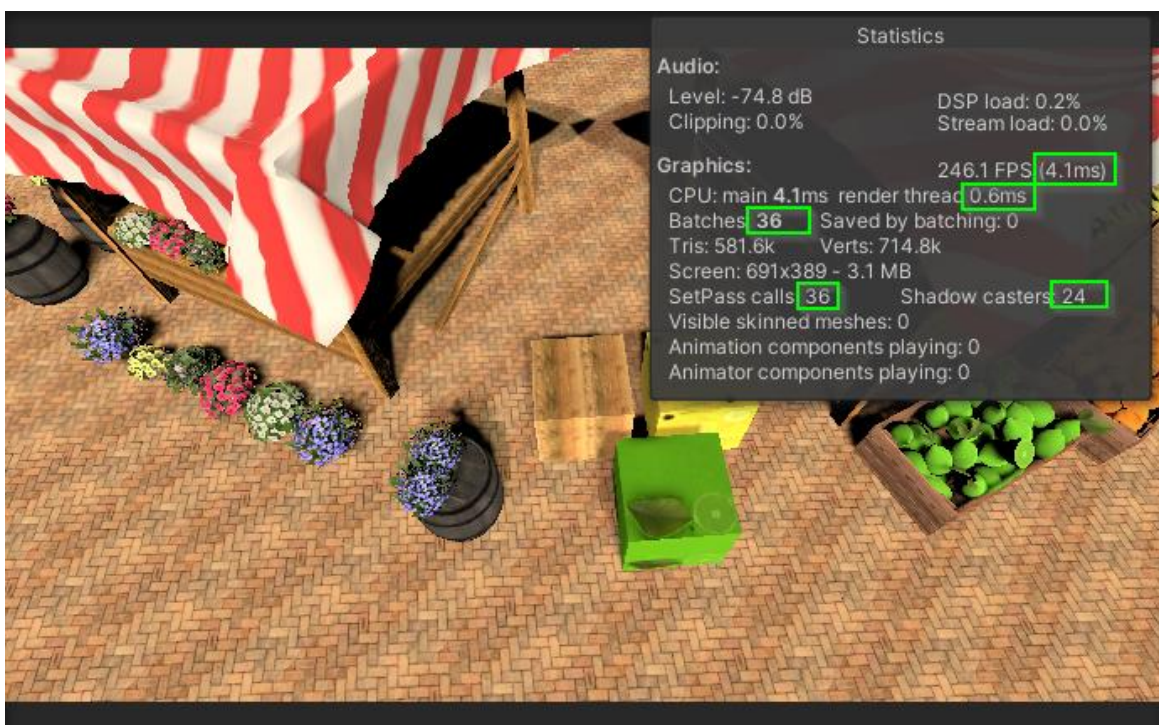


Comparison of a complex scene

Before – 107 Materials



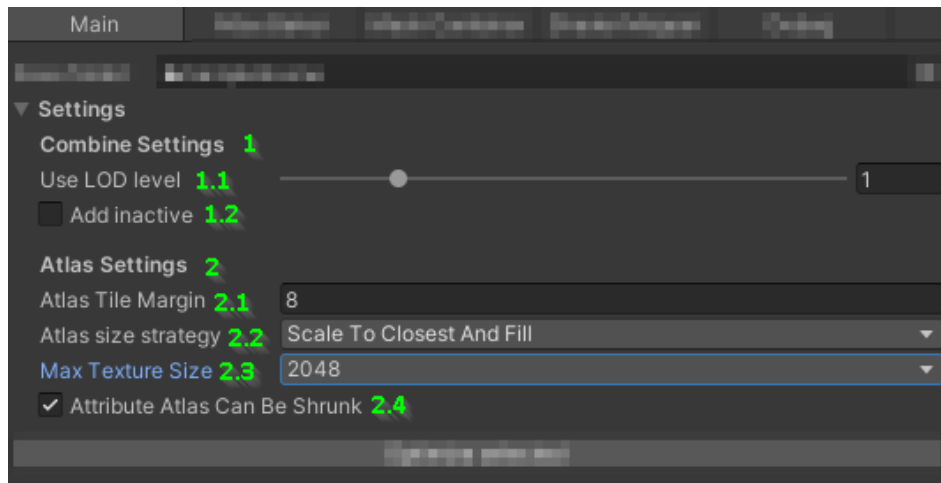
*Afterwards – 3 materials**



* There's a small overhead in texture size for baking the scalar properties into small textures.

Settings

The settings allow you to finetune the optimization process.



1. These settings control how the meshes (the geometry) is optimized
 - 1.1. If the model has different LODs the defined level will be used. Set it to 0 if you want the highest detail level or you do not use LODs
 - 1.2. Inactive objects will be ignored. Turn on to add inactive meshes.
 - 1.3. Color similarity. Increase this if you want to merge materials with similar attribute colors. If set to a low value each different color will result in a texture in the atlas map when using a mapping from color to texture.
2. These settings control the texture creation process
 - 2.1. This is the number of pixels at the border of a tile inside of an atlas map which will be repeated. This should be a power of 2 (2, 4, 8, 16, ...) This controls the available mip-levels (1, 2, 3, 4, ...). 8 for 3 mip-levels is a good value for the start.
 - 2.2. Ideally, we want to place as many textures (tiles) on one atlas map as possible. This is in most cases not possible without wasting space on the atlas map. You can pick if you want to primarily save texture memory (scale down) or preserve quality (scale up) or take the compromise of both (scale to closest and fill)
 - 2.3. This is the maximum texture size being used for the atlas map. If possible, less is used but never more.
 - 2.4. Scalar properties like color, smoothness or any number on a material will be baked into a texture. This texture can be quite small without any side-effects as long all textures are in average round about the same size. If the difference of texture sizes is huge texture properties can get very inaccurate. In this case you should deactivate this option than use e.g. crunch compression for those attribute textures.

Expert Mode

There are several options to fine tune the optimization process.

Mesh Combiner

The mesh combiner is the first tool you need to run. It will just merge all your mesh renderers into a single one using submeshes and creating a mesh asset.

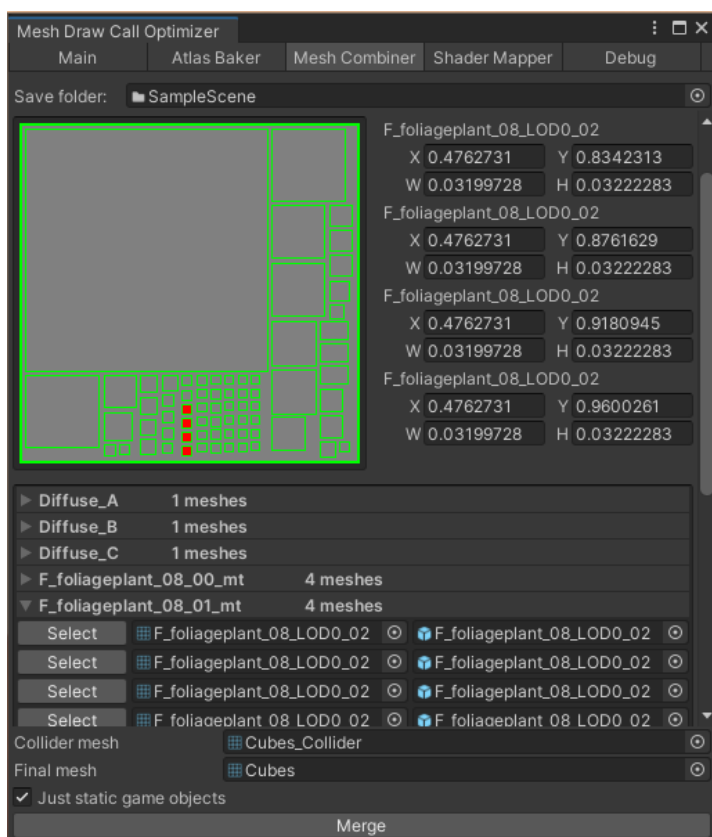
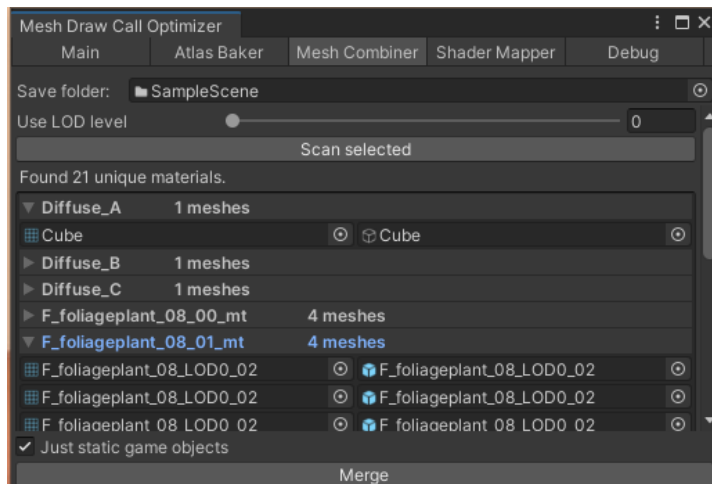
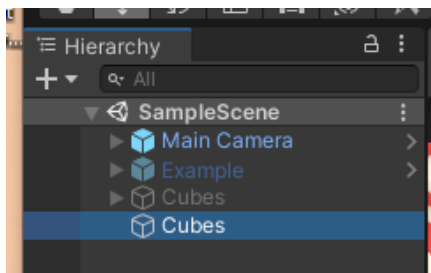
On the top you can adjust the same settings like explained in the main tab (See Settings 1).

- 1) Select the objects to optimize and click **SCAN SELECTED**
You will see a list of all unique meshes going to be merged. You can expand every entry and see where the mesh instances are used.
- 2) Activate **JUST STATIC GAME OBJECTS** if you want to remap the lightmap UVs exclusively for the game objects marked static. Deactivate to remap all the lightmap UVs including the non-static objects. Default is turned on.
- 3) Click **MERGE** to combine the meshes.

After combining the meshes you can double click the collider mesh to see the mesh generated for the colliders and the final mesh to see the combined rendered mesh. In the expanded section of a mesh, you can press **SELECT** to see where the original object's lightmap is placed on the combined lightmap.

On the top right the lightmap UV region is displayed.

In the scene the original gameobjects are going to be deactivated and the combined gameobject will be selected. It will have the name of the first object you selected.



Atlas Baker

Select the combined object from the previous chapter and click **SCAN SELECTED**.

The atlas baker will now analyze which materials can be merged together. In our test scene we used 3 times a *Mobile Bumped* and many times the *Standard* shader. The *Standard* shader was used for Diffuse and Cutout objects. This results in two materials one for all the cutout objects and one for the rest.

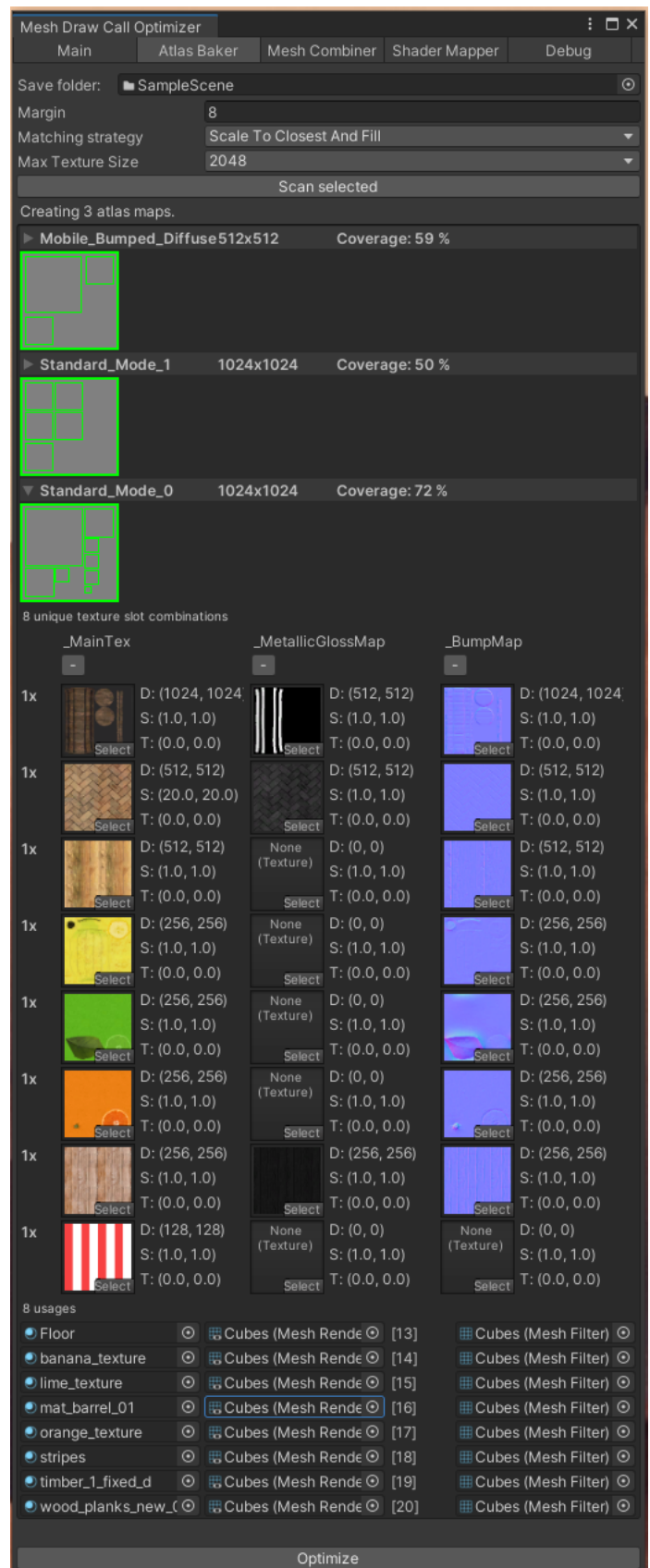
The green boxes show you how the atlas maps are going to be layouted.

If you unfold an atlas material entry you can see which objects and materials are going to be merged.

On the top you see the settings which are also in the main tab visible. (See Settings 2)

Inside of the foldout beneath the texture slot names you can use the minus button to entirely erase a specific texture kind. **BE AWARE THAT THIS WILL CHANGE YOUR INPUT MATERIALS!**

Ensure your object to be optimized is still selected and then click *Optimize* to start the process. This can take a while depending on the size of your scene and the number of textures and materials used.

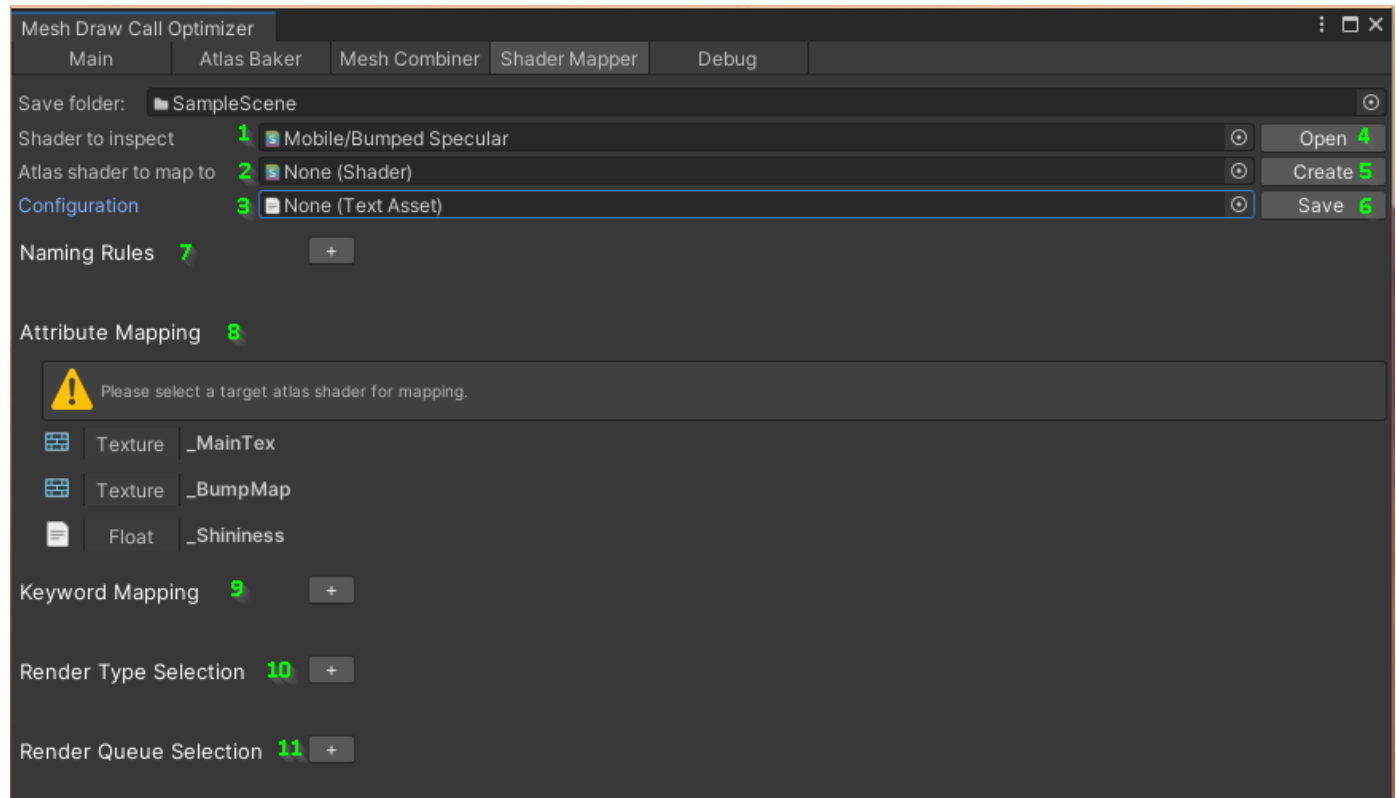


Shader Mapper (BETA)

The shader mapper is a complex tool developed to create atlas shaders from simple shaders. It works best for classic shaders but can also be used for manually mapping to shader graphs if you want to use the URP or HDRP.

When working with simple shaders it is an entirely automatic process. We're going to illustrate the tool by creating an atlas shader for a *Mobile Bumped Specular* shader.

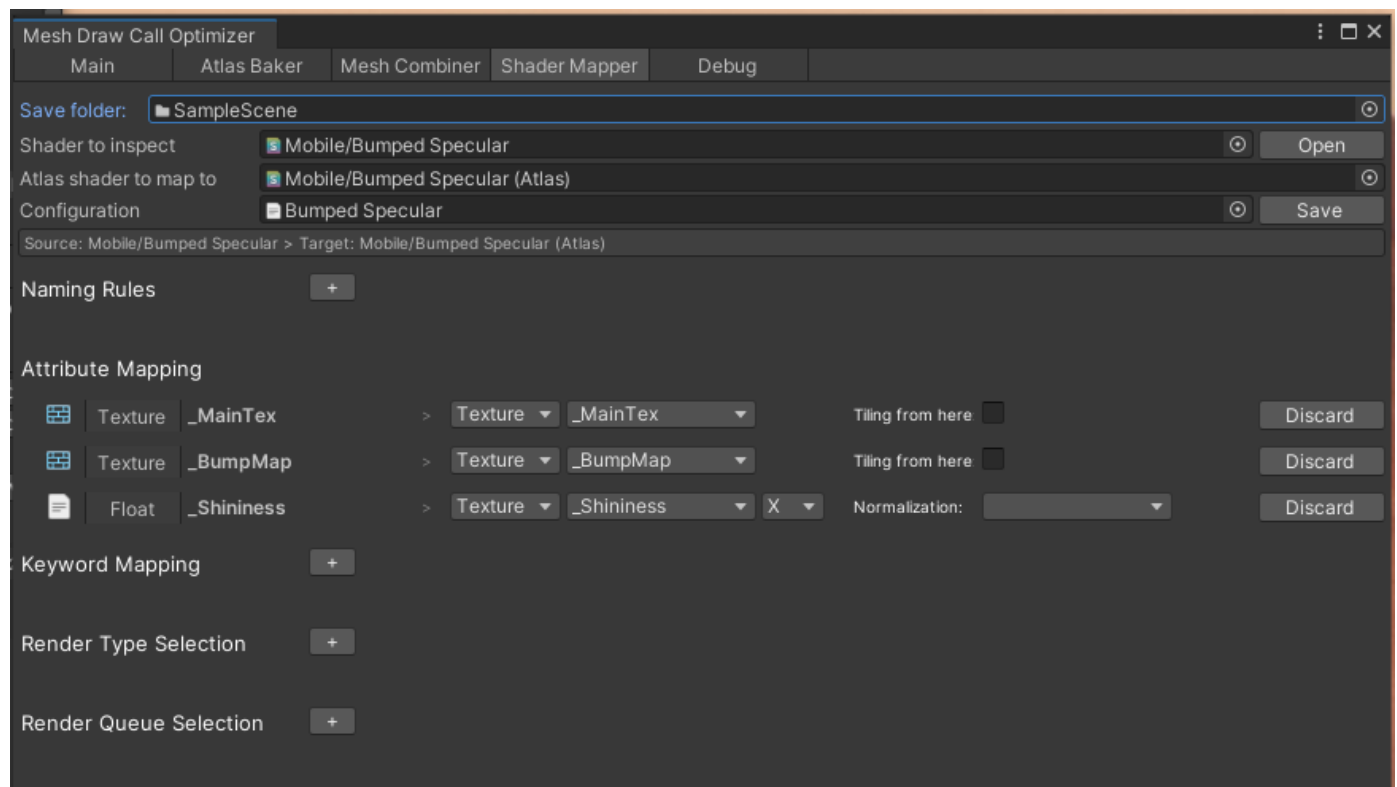
The Interface



- 1) Click the gear icon on the right of the field to select the shader you want to support for atlasmapping.
- 2) If you created a matching target shader by hand click the gear icon here to select it or leave it empty.
- 3) This will display the text asset used to map from the input shader to the atlas shader.
- 4) Click **OPEN** to view the input shader source. This tool is delivered with Unity's built-in shaders which can be viewed here as well.
- 5) This is the most powerful part of the feature. Click this **CREATE** button to automatically generate an atlas shader from the input shader. This is a beta feature and far from complete but it works for many simple shaders. If your shader uses no custom includes your chances are quite high that the result will work.
- 6) Click **SAVE** when you made changes to the section below.
- 7) If your material uses fields which alter the internal shader code fill this in (have a look at the custom made *Standard* or *Standard (Specular)* shaders to see how to use this.)
- 8) This is the main section for mapping the input shader attributes to the target shader attributes. See the next section.
- 9) This section can be used to activate certain keywords on the atlas shader if specific criteria are fulfilled (have a look at the custom made *Standard* or *Standard (Specular)* shaders to see how to use this.)
- 10) Allows you to set the render type of the atlas material depending on some criteria (have a look at the custom made *Standard* or *Standard (Specular)* shaders to see how to use this.)
- 11) Set the atlas shader render queue depending on some criteria (have a look at the custom made *Standard* or *Standard (Specular)* shaders to see how to use this.)

Creating an atlas shader

In this example we simply selected the *Mobile Bumped Specular* shader in the top field. Then we clicked **CREATE** on the right (between **OPEN** and **SAVE**, the **CREATE** button disappears if an atlas shader is assigned).



A shader will be generated and an automatic attribute mapping as well.

Textures

Textures will be mapped to an atlas texture of the same name. If you use tiled textures you have to decide which texture slots tiling is to be used. In most cases this is the main or base texture.

Floats, Ranges, Vector, Color

Scalar types need to be mapped to a texture. A float or range can be saved in a single-color channel of a texture, while vectors and colors use the full spectrum of the new texture. If your values lie outside the range of 0..1 or for color 0..255, you need to specify an attribute in the atlas shader which is set to the highest value.

Troubleshooting

In our example the shader will report the error: *Too many texture interpolators would be used for ForwardBase pass (9 out of max 8), try adding #pragma target 3.0*. This is caused by the more complex atlas shader. This can be fixed by opening the atlas shader file (double click the **ATLAS SHADER TO MAP TO** field). Go to line 26 directly beneath **CGPROGRAM** and insert `#pragma target 3.0` above `#pragma surface [...]`.

Supported Pipelines and shaders

The optimizer was developed initially for the classic render pipeline and supports the following shaders:

Classic Pipeline

Standard (Metallic Setup)

- **SUPPORTED TEXTURES:** MainTex, BumpMap, OcclusionMap, PrallaxMap, EmissionMap, MetallicGlossMap
- **SUPPORTED ATTRIBUTES:** DiffuseColor, GlossMapScale, Glossiness, Prallax, OcclusionStrength, Cutoff, Metallic, BumpScale, SmoothnessTextureChannel, EmissionColor
- **UNSUPPORTED:** details (*DetailAlbedoMap, DetailNormalMap, DetailMask, DetailNormalMapScale, UVSec*)

Standard (Specular Setup)

- **SUPPORTED TEXTURES:** MainTex, BumpMap, OcclusionMap, PrallaxMap, EmissionMap, SpecGlossMap
- **SUPPORTED ATTRIBUTES:** DiffuseColor, GlossMapScale, Glossiness, Prallax, OcclusionStrength, Cutoff, BumpScale, SmoothnessTextureChannel, EmissionColor, SpecColor
- **UNSUPPORTED:** details (*DetailAlbedoMap, DetailNormalMap, DetailMask, DetailNormalMapScale, UVSec*)

Autodesk Interactive (Roughness Setup)*

- * The Autodesk Interactive shader will be mapped to the Standard shader. It will not look identical to the input. Consider adapting it to a Standard shader before using it.

| Legacy shaders | Default | Transparent | Cutout | Transparent Cutout | Self-Illumin | Lightmapped | Reflective | Mobile |
|-------------------|---------|-------------|--------|--------------------|--------------|-------------|------------|---------|
| Diffuse | ✓ x2 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Diffuse Bumped | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Diffuse Parallax | ✓ | ✓ | ✓ | | ✓ | | ✓ | |
| Specular | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | |
| Specular Bumped | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ x2 |
| Specular Parallax | ✓ | ✓ | | | ✓ | | ✓ | |
| Vertex Lit | ✓ | ✓ | ✓ | ✓ | ✓ | ✂ | ✓ x2 | ✓ x2 |

An empty cell means that this shader does not exist in Unity. A multiplicator at the bottom right indicates that there are multiple shaders of this type.

| | | | | | | | |
|---|-------------|---|-----------|---|---------------------|---|-------------------------|
|  | UNSUPPORTED |  | SUPPORTED |  | PARTIALLY SUPPORTED |  | REPLACED BY ALTERNATIVE |
|---|-------------|---|-----------|---|---------------------|---|-------------------------|

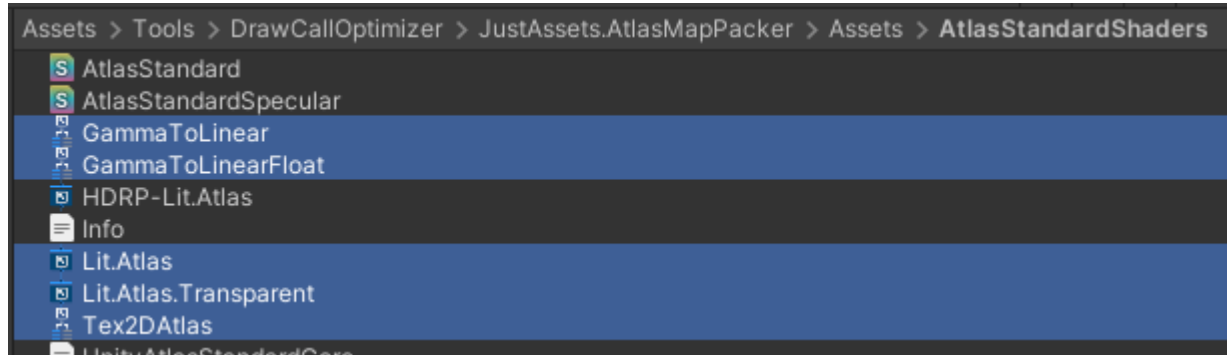
Universal Render Pipeline / High-Definition Render Pipeline

For testing the tool you can use the SampleScene in **Scenes/HDRP** for the HDRP and the SampleScene from **Scenes/URP** for the URP. Follow the explanation from the section *How to Use*.

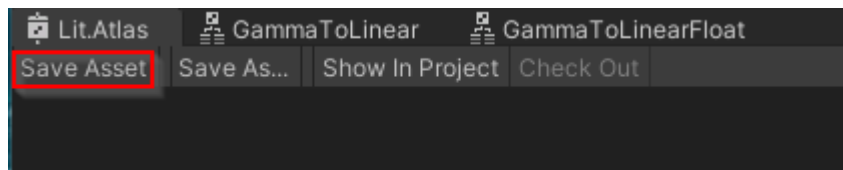
How to set up

- 1) Select from the menu **WINDOW > PACKAGE MANAGER**
- 2) Click on **PACKAGES** on the left and select **UNITY REGISTRY**
- 3) Search for **UNIVERSAL RP** or **HIGH DEFINITION RP**
- 4) Select it and click **INSTALL**
- 5) *Optional for URP if optimized scene is very dark:*
 - a. Recompile delivered shaders graphs at
ASSETS > TOOLS > DRAWCALLOPTIMIZER > JUSTASSETS.ATLASMAPPACKER > ASSETS > ATLASSTANDARDSHADERS

For URP open the selected ones one by one:

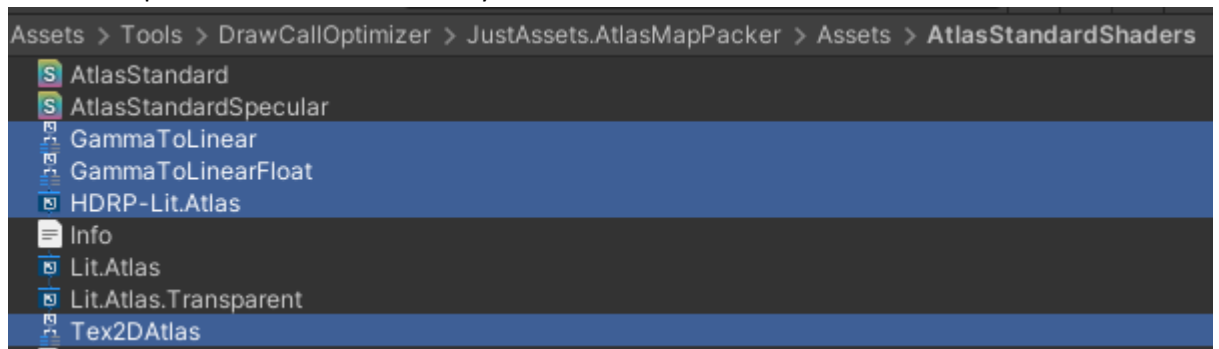


- b. And click Save Asset:

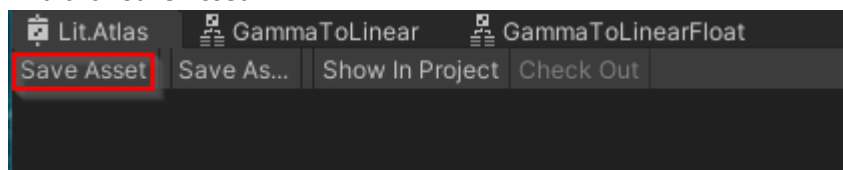


- 6) *Optional for HDRP if optimized scene is very dark:*
 - a. Recompile delivered shaders graphs at
ASSETS > TOOLS > DRAWCALLOPTIMIZER > JUSTASSETS.ATLASMAPPACKER > ASSETS > ATLASSTANDARDSHADERS

For HDRP open the selected ones one by one:



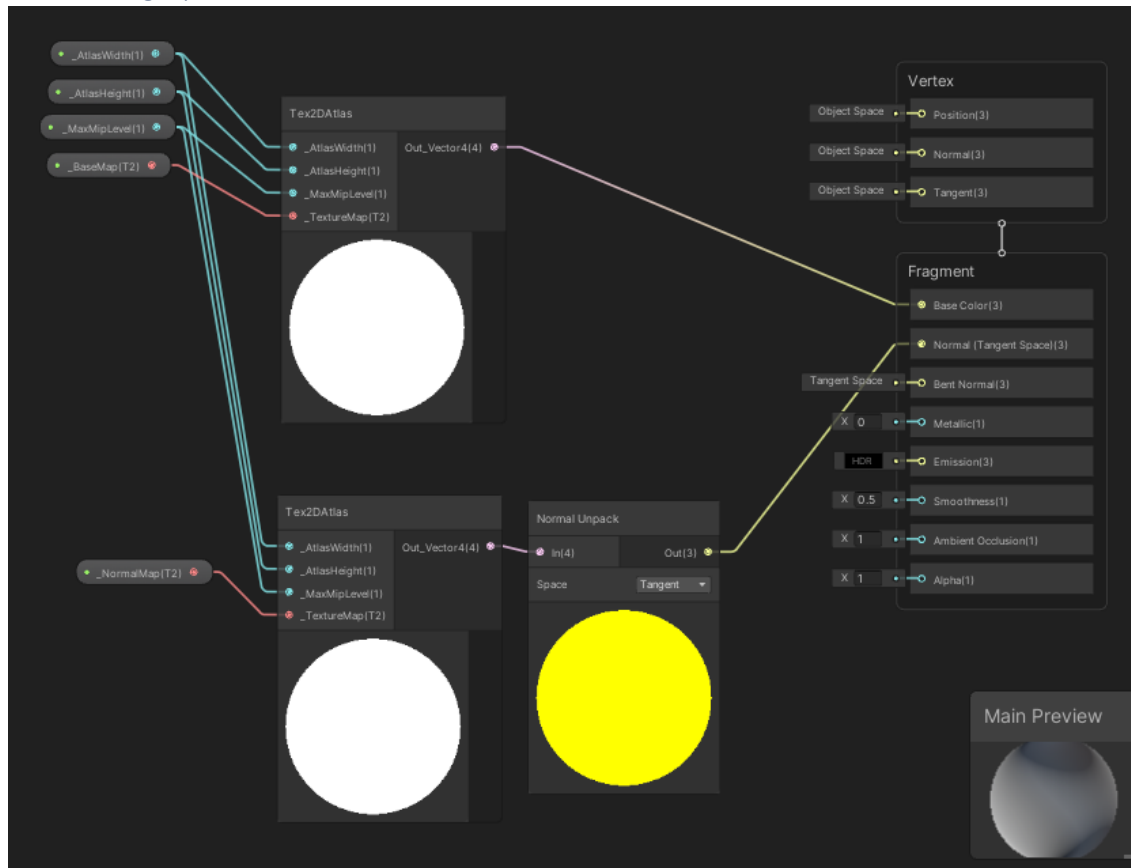
- b. And click Save Asset:



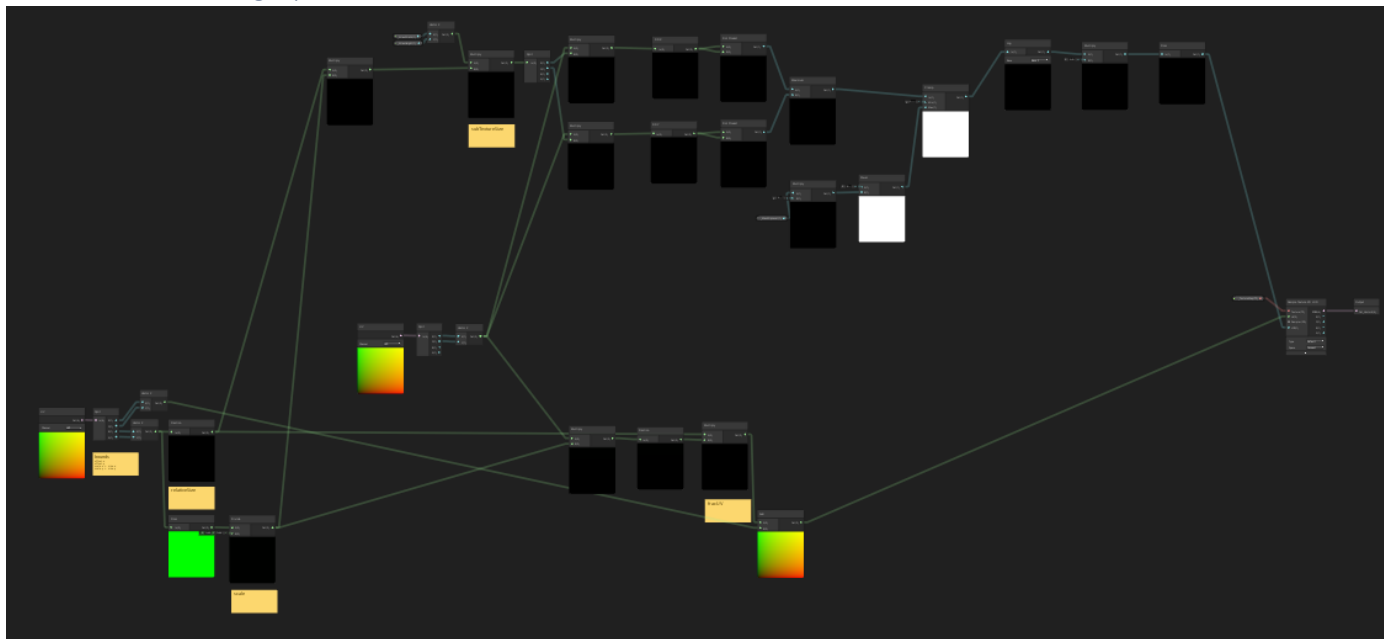
Creating your shaders

By default, a simple shader graph for the HDRP and URP Lit shaders is provided. You can adapt your shader graphs by replacing your Tex2D nodes by the Tex2DAtlas subgraph (**Tools/AtlasUtility/Assets/AtlasStandardShaders/Tex2DAtlas.shadersubgraph**).

Lit shader graph



Tex2DAtlas shader graph



Support

If you experience a bug, please create a ticket [here](https://support.justassets.de) or write an e-mail with detailed description to support@justassets.de. Please provide the tools version, a stack-trace in case of an exception and steps to reproduce. Please attach a minimal example if it is required to reproduce the problem.

Changelog

Version 2.6.2 (10.01.2023)

- Reuse meshes when baking atlas maps without combining meshes first

Version 2.6.1 (08.01.2023)

- Fixes bug preventing to create a build

Version 2.6.0 (03.01.2023)

- Drastically improved lightmap UV transfer speed and atlas layouting duration by using smarter algorithm
- Added minimal texture size for emulating materials without textures
- Fixes bug when optimizing an unsaved scene
- Support Unity 2019.x (classic render pipeline)

Version 2.5.0 (07.12.2022)

- Support pass through for unsupported materials
- Improve atlas layouting speed
- Add basic support for URP Baked Lit Material

Version 2.4.0 (28.11.2022)

- Fixing bug when merging materials without textures
- Fixing bug merging mesh colliders
- Upgrading collider utility to version 2.0.0
- Fixing bug in URP Lit not respecting color field when no texture is used

Version 2.3.0 (01.07.2022)

- Fixing bug when using renderers in LOD groups not being assigned to any LOD group
- Upgrading package minimal requirement to Unity 2020.3

Version 2.2.1 (28.12.2021)

- Fixing exception when scaling texture down to 1x1
- Fixing error when creating a text asset using AssetDatabase

Version 2.2.0 (20.12.2021)

- Optimize texture size on URP Lit materials
- Fixing bug when merging materials with different colors set
- Fixing exception when merging Lit materials without textures
- Fixing exception when using textures marked as UI texture
- Fixing issue when creating very small atlas maps

Version 2.1.0

- Fixing exception when optimizing HDRP materials
- Fixing texture reference in HDRP example scene
- Adding rendering pipeline detection
- Adding check for shader errors
- Updating support link and email.

Version 2.0.1

- Fixes shader warnings
- Fixes some mesh issues

Version 1.1.1

- Updating support links and mail.

Version 1.1.0

- Initial release