ShaderConverter Documentation

1. Introduction

ShaderConverter is an editor tool that converting GLSL image shader to Unity shader. With a couple of clicks, you can easily convert image (fragment/pixel) shader to readable, usable and efficient Unity shader (HLSL), and you can preview the effect of shader immediately.

2. Features:

- ✧ The conversion workflow is very simple: One editor window, codes side to side and one button to convert.
- ✧ Support to set SubShader Tags: Render Type and Render Queue;
- ✧ Support to set Pass command: Cull, ZWrite, ZTest, Blend;
- ✧ Support to specify a shader model with the #pragma target directive;
- ✧ Support to convert Shadertoy shader with multiple passes: Common, Buffer A, Buffer B, Buffer C, Buffer D;
- ✧ Support to auto assign Shadertoy input: iFrame, iFrameRate, iMouse, iDate, iSampleRate, _iChannelTime;
- ✧ Support 2D textures, cubemaps and music/sound input;
- ✧ Support to save converted shader to a file, and create material from converted shader;
- ✧ Support to preview the effect of converted shader immediately;

Note:
* Tested in Unity 2019.4.28, Unity 2020.3.38 and Unity 2021.3.6;
* This editor tool only support 64-bit Unity Editor on Windows for now!
* Converted shader with this editor tool only support built-in render pipeline!

3. Examples

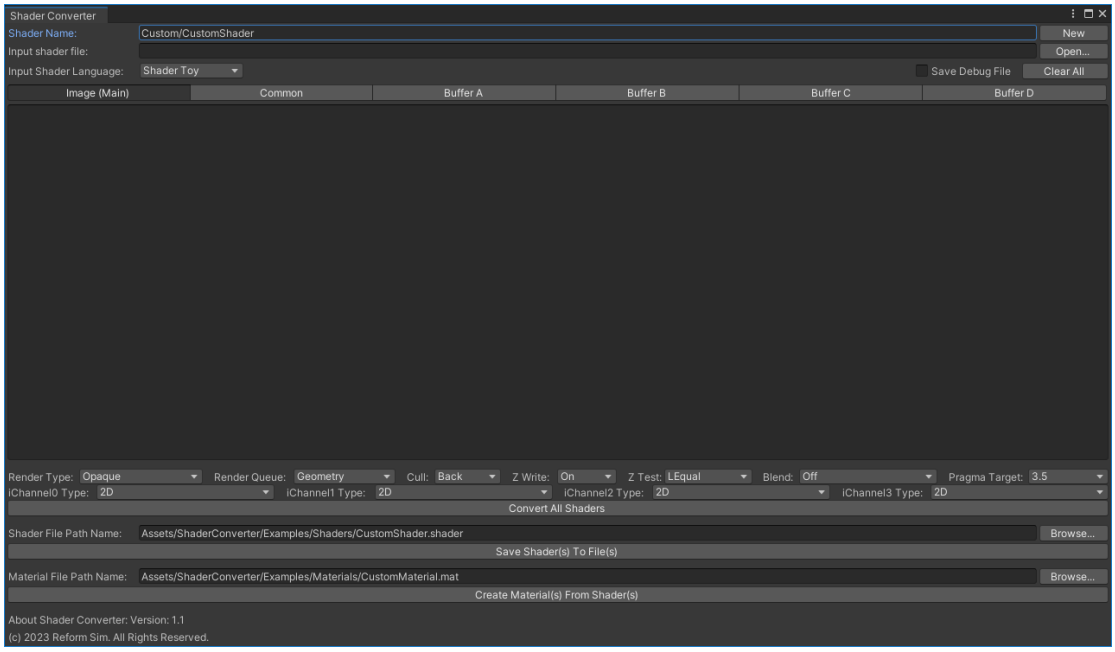
After importing this ShaderConverter package to your project, as you can see, there are example scenes in "**Assets/ShaderConverter/Examples/Scenes**" Folder. Open one of them and play, it will show a quad with converted shader.

4. Getting Started

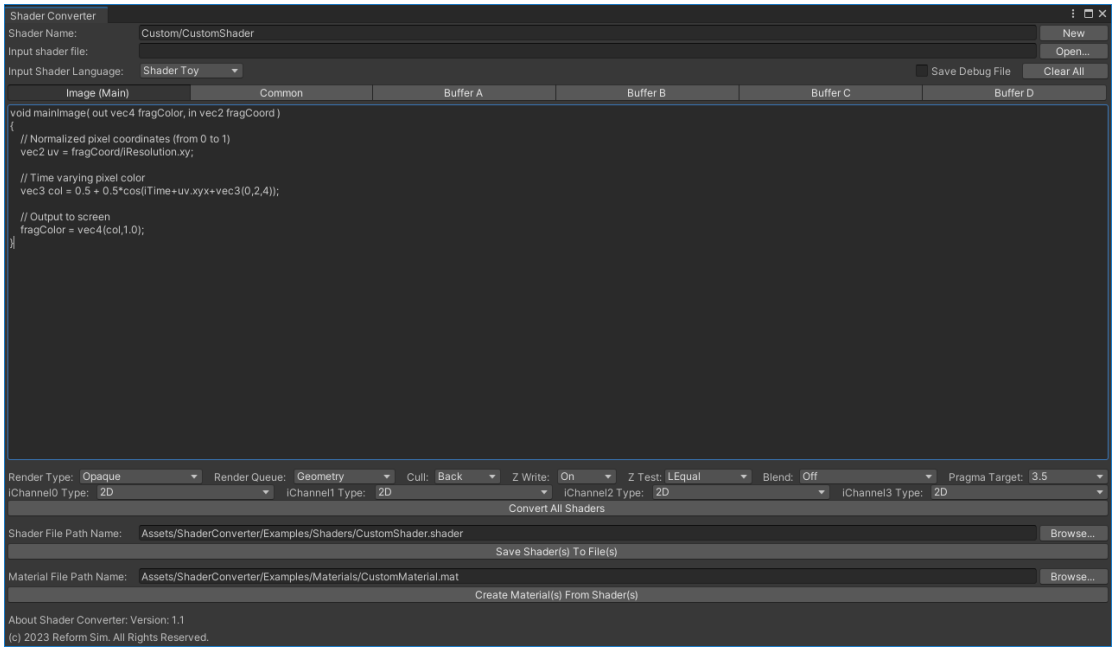
IMPORTANT:
To avoid different compatibility issues and errors:
- Always remove previous version from project Assets before updating;

4.1. Converting Image (fragment/Pixel) shader with single pass

1) Importing ShaderConverter package to your project;
2) You can open ShaderConverter window from Menu “**Tools/ShaderConverter…**”, the editor window will show as in the following picture:

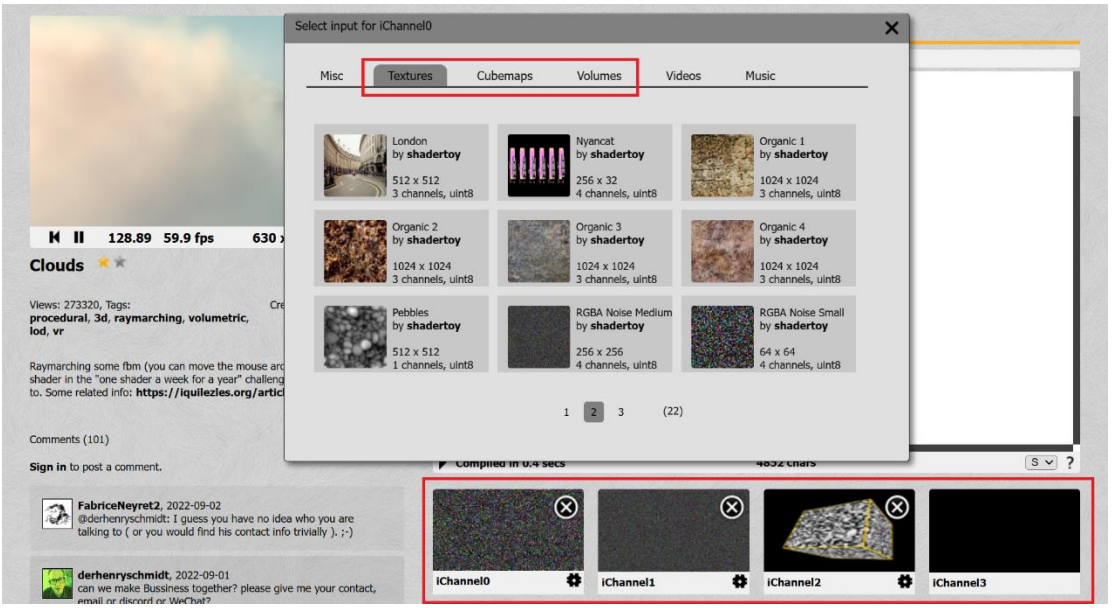


3) You can change the name of shader manually, or click “**New**” button, it will auto generate a new name;
4) You can open a shader file by click “**Open…**” button, it will load the file to the input area of the current tab. There is a test file named “**TestImageShader.txt**” in “**Assets/ShaderConverter/Examples/GLSL**” Folder. Or you can paste text content of image shader on the Shadertoy website to the input area of the main tab: Image (Main).

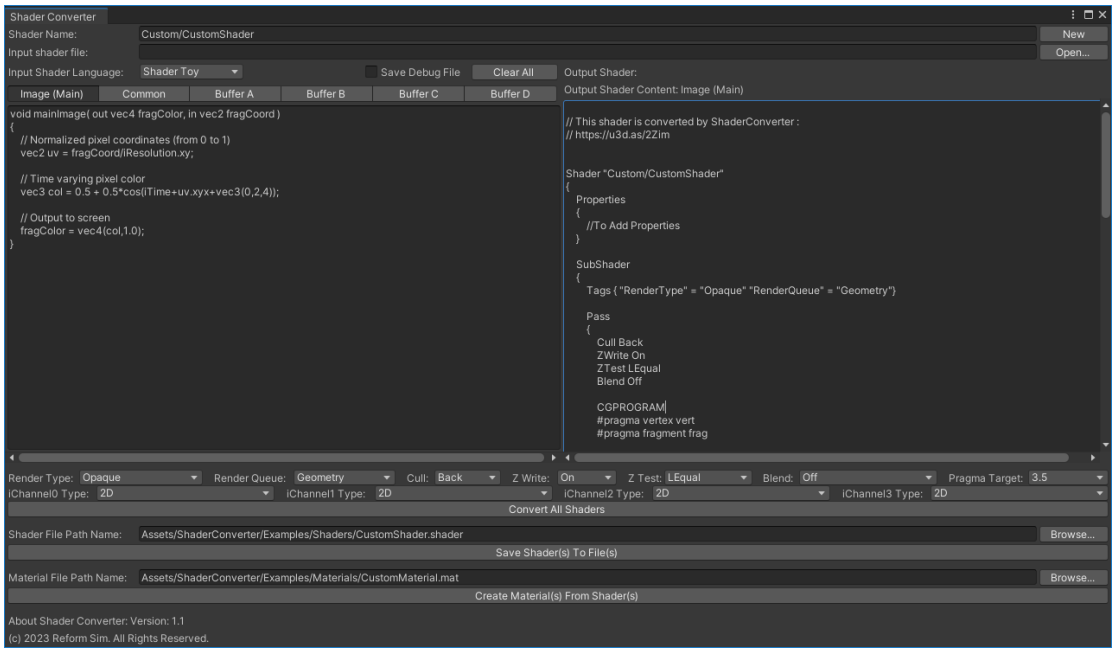


5) You can change “**Render Type**”, “**Render Queue**”, “**Cull**”, “**ZWrite**”, “**ZTest**”, “**Blend**”, “**Pragma target**”. In most cases, you can skip this step.

6) You can change input texture (iChannel0 ~ iChannel3) type: **2D** or **Cube**. **Input texture type must match with the input of Shadertoy shader.** If iChannelx is in “Textures” tab, you should change iChannelx type to “2D”. If iChannelx is in “Cubemaps” or “Volumes” tab, you should change iChannelx type to “Cube”, as shown in the following picture. If shader has no input texture, you can skip this step.



7) Click “**Convert All Shaders**” button, it will show the converted shader on the right side, as shown in the following picture:



8) Click “**Save Shader(s) To File(s)**” button, it will save the converted shader to a file. You can change the path name of shader file. If you leave the name to be default name, it will overwrite the file “**CustomShader.shader**” in “**Assets/ShaderConverter/Examples/Shaders**” folder. Play the example scene, it will show the Quad with the effect of shader.

9) You can click "**Create Material(s) From Shader(s)**" button to create a new material, then you can use this material anywhere; **If you want to get the same effect as ShaderToy, you must assign same texture to the material.**
10) If shader has some other input, for example: iFrame, iFrameRate, iMouse, iDate, iSampleRate, iChannelTime, you should add "**ImageShaderInput**" component to the GameObject. This component will auto assign these input.
11) If shader has music input, you should also add "**ImageShaderMusicInput**" component and Audio Source component to the GameObject.
    Parameters of ImageShaderMusicInput component:
    - **Sample Num**: Number of the samples must be a power of 2. (ie 128/256/512 etc).
    - **Texture Name Type**: Which texture should set to.

    Then assign an audio clip to the Audio Source component. All Done.
    Notes: There is an example scene named "MusicInputShaderExample" to demonstrate music input.

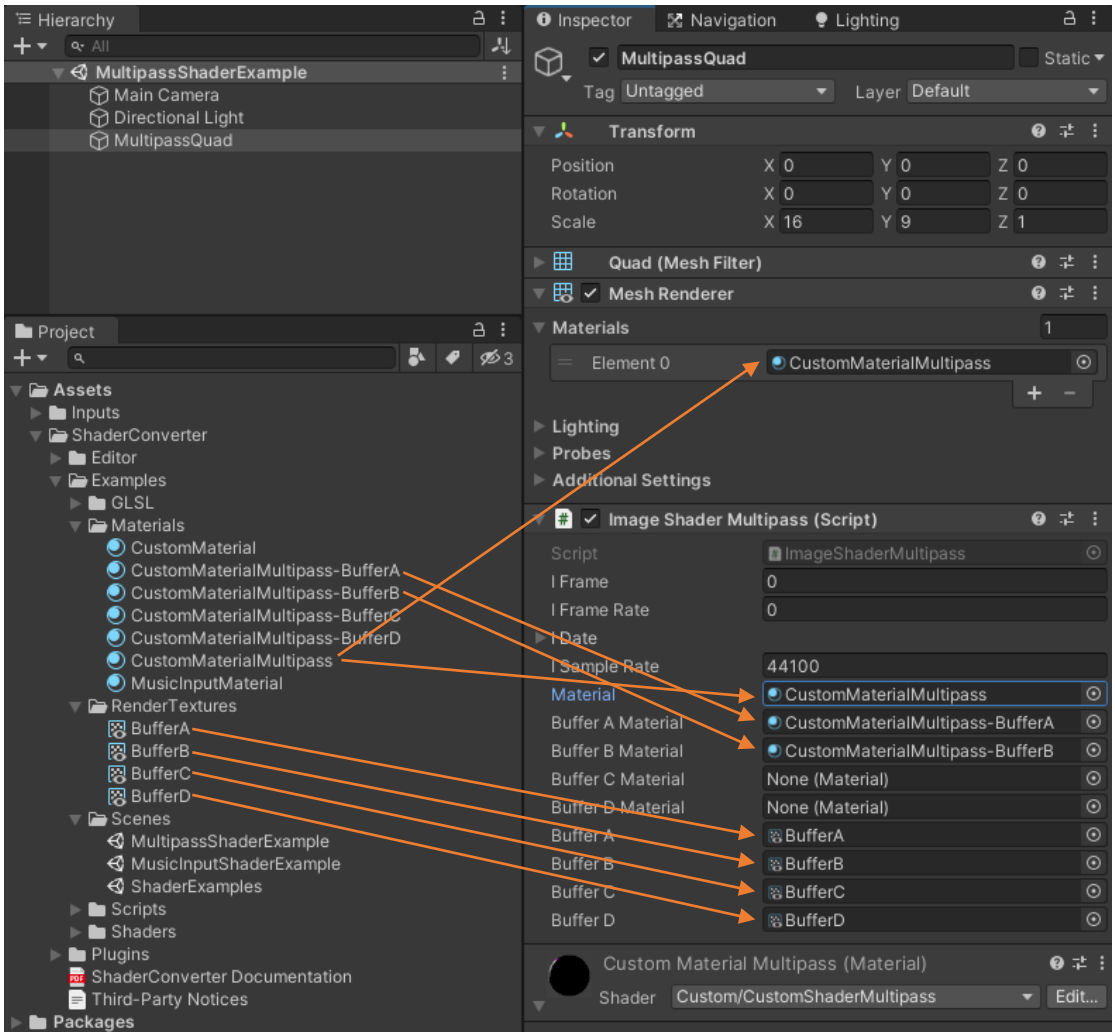
4.2. Converting Image shader with multiple passes

Image shader with multiple passes is composed of multiple shaders with single pass. Converting shader of every pass is similar with converting shader with single pass. Open and play the example scene named "MultipassShaderExample", it will show a quad with the effect of shader with multiple passes.

1) Change the name of shader to "**Custom/CustomShaderMultipass**".
2) There are six tabs on the editor window: **Image (Main), Common, Buffer A, Buffer B, Buffer C, Buffer D**. You should paste text content of every pass on the Shadertoy website to the input area of the corresponding tab.
3) Then click "**Convert All Shaders**" button, it will converting all shaders.
4) Change the name of shader file to "CustomShaderMultipass.shader", then click "**Save Shader(s) To File(s)**" button, it will save all shaders to files named "CustomShaderMultipass.shader", "CustomShaderMultipass-BufferA.shader", "CustomShaderMultipass-BufferB.shader", etc.
5) Change the name of material file to "CustomMaterialMultipass.mat", then click "**Create Material(s) From Shader(s)**" button, it will create all material files named "CustomMaterialMultipass.mat", "CustomMaterialMultipass-BufferA.mat", "CustomMaterialMultipass-BufferB.mat", etc.
    In the "ShaderConverter/Examples/RenderTextures" folder, there are 4 RenderTextures named "BufferA", "BufferB", "BufferC", "BufferD", which store the result of every pass. You could assign these RenderTextures to the created materials according to the Shadertoy shaders.
6) Create a Quad GameObject named "MultipassQuad", assign material "CustomMaterialMultipass" to it.
7) Add "ImageShaderMultipass" component to the GameObject, and assign material "CustomMaterialMultipass" to the field "Material". Assign material "CustomMaterialMultipass-BufferA" to the field "Buffer A Material". If needed, assign material "CustomMaterialMultipass-BufferB" to the field "Buffer B Material", etc.

For example, if Shadertoy shader only has buffer A and buffer B, then you only need to assign material "CustomMaterialMultipass-BufferA" and "CustomMaterialMultipass-BufferB" to the field "Buffer A Material" and "Buffer B Material". The field "Buffer C Material" and "Buffer D Material" must be empty.

8) Assign RenderTexture "BufferA", "BufferB", "BufferC", "BufferD" to the field "Buffer A", "Buffer B", "Buffer C", "Buffer D", as shown in the following picture:



9) Assign RenderTextures or normal textures to the input of the materials according to the Shadertoy shader. **If you want to get the same effect as ShaderToy, you must assign same textures to the materials.**

5. Support

If you have any questions, issues or suggestions, feel free to email me at:
reformsim@outlook.com