ShaderConverter Documentation

1. Introduction

ShaderConverter is an editor tool that converting GLSL image shader on the Shadertoy website (Shadertoy) to Unity shader for Built-in Render Pipeline. With a couple of clicks, you can easily convert GLSL 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.
- Supports to set SubShader Tags: Render Type and Render Queue;
- Supports to set Pass command: Cull, ZWrite, ZTest, Blend;
- Supports to specify a shader model with the #pragma target directive;
- Supports to convert Shadertoy shader with multiple passes: Common, Buffer A, Buffer B, Buffer C, Buffer D;
- Supports to auto assign Shadertoy input: iFrame, iFrameRate, iMouse, iDate, iSampleRate, iChannelTime, iResolution;
- Supports the same keyboard and mouse input as Shadertoy;
- Supports 2D textures, cubemaps, video and music/sound input;
- Supports to convert main image shader to object space shader or full screen shader;
- **[NEW]** Object space shader supports both VR (Single Pass Instanced) mode and non-VR mode;
- Supports to render the effect of main image shader to RenderTexture;
- **[NEW]** Supports Gamma and Linear color space;
- **[NEW][Experimental]** Simple global numeric global macros and constants defined in the shader can be extracted into the properties of the shader;

Notes:
* Tested in Unity 2019.4.28, Unity 2020.3.38 and Unity 2021.3.25;
* 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!
For other render pipeline, please use:
Shader Converter for Universal Render Pipeline
Shader Converter for High Definition Render Pipeline
* This editor tool cannot convert shaders for Built-in render pipeline to shaders for URP or HDRP!

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

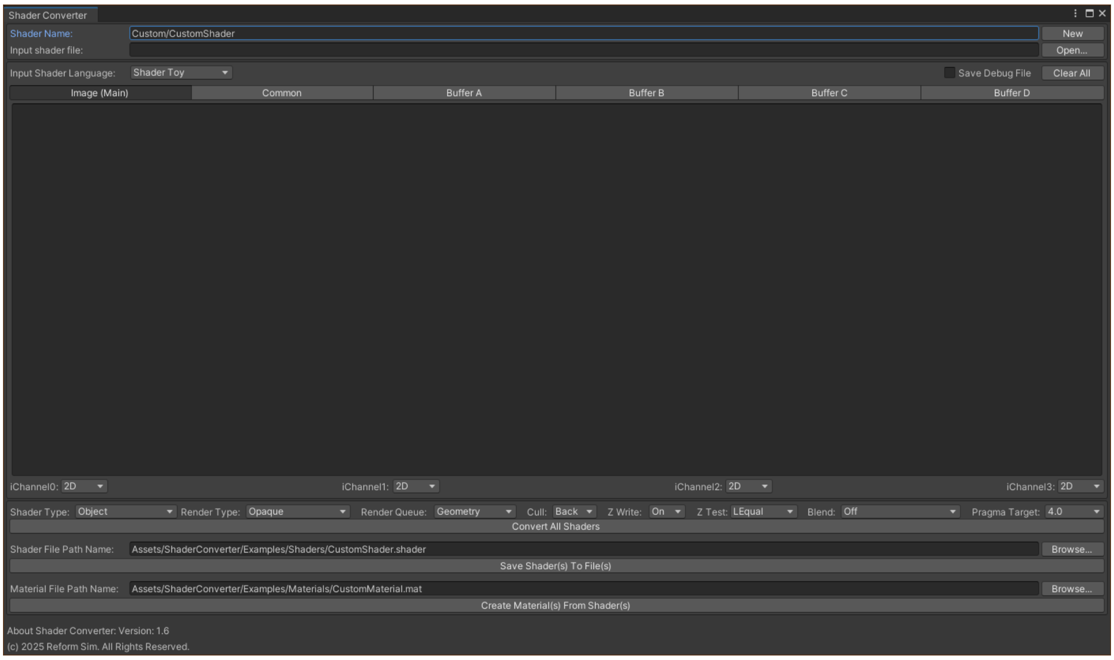
IMPORTANT:

To avoid different compatibility issues and errors:

- Always remove previous version from project Assets before updating;

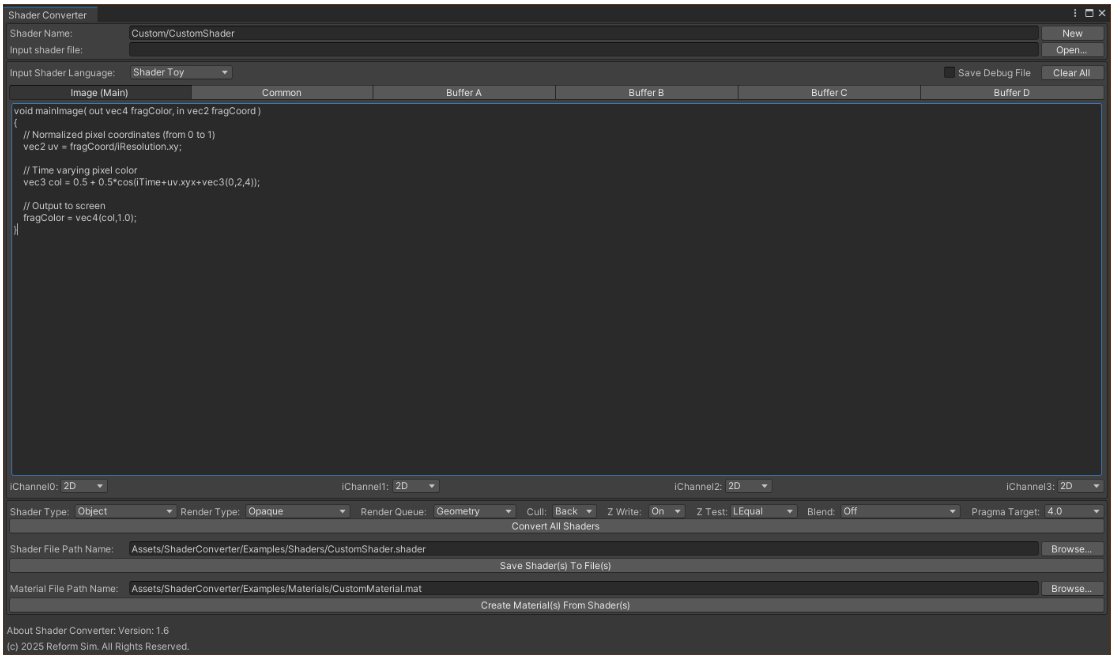
4.1. Converting Image (fragment/Pixel) shader with single pass

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



Shader Name: Custom/CustomShader
Input shader file:
Input Shader Language: Shader Toy
□ Save Debug File
iChannel0: 2D
iChannel1: 2D
iChannel2: 2D
iChannel3: 2D
Shader Type: Object
Render Type: Opaque
Render Queue: Geometry
Cull: Back
Z Write: On
Z Test: LEqual
Blend: Off
Pragma Target: 4.0
Shader File Path Name: Assets/ShaderConverter/Examples/Shaders/CustomShader.shader
Material File Path Name: Assets/ShaderConverter/Examples/Materials/CustomMaterial.mat
About Shader Converter: Version: 1.6
(c) 2025 Reform Sim. All Rights Reserved.

3) You can change the name of shader manually, or click "**New**" button, it will auto generate a new name; The name of shader file will be updated automatically.
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).



Shader Name: Custom/CustomShader
Input shader file:
Input Shader Language: Shader Toy

```
void mainImage( out vec4 fragColor, in vec2 fragCoord )
{
   // Normalized pixel coordinates (from 0 to 1)
   vec2 uv = fragCoord/iResolution.xy;

   // Time varying pixel color
   vec3 col = 0.5 + 0.5*cos(iTime+uv.xyx+vec3(0,2,4));

   // Output to screen
   fragColor = vec4(col,1.0);
}
```

iChannel0: 2D iChannel1: 2D iChannel2: 2D iChannel3: 2D

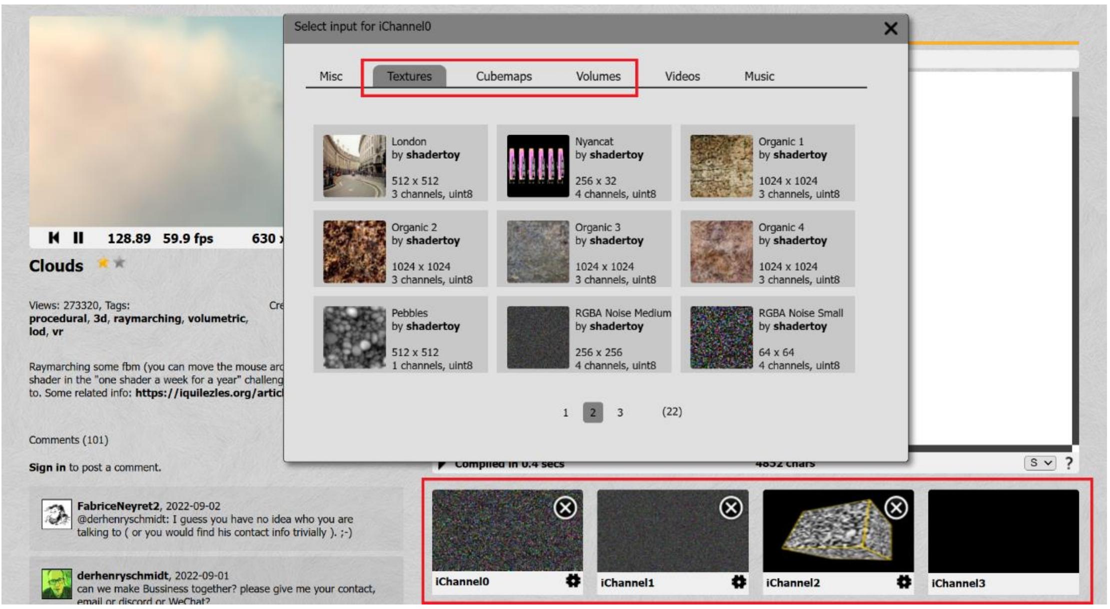
Shader Type: Object Render Type: Opaque Render Queue: Geometry Cull: Back Z Write: On Z Test: LEqual Blend: Off Pragma Target: 4.0

Shader File Path Name: Assets/ShaderConverter/Examples/Shaders/CustomShader.shader

Material File Path Name: Assets/ShaderConverter/Examples/Materials/CustomMaterial.mat

About Shader Converter: Version: 1.6
(c) 2025 Reform Sim. All Rights Reserved.

5) You can change input texture (iChannel0 ~ iChannel3) type for shader in each tab: **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.



Views: 273320, Tags:
procedural, **3d**, **raymarching**, **volumetric**, **lod**, **vr**

Raymarching some fbm (you can move the mouse arc
shader in the "one shader a week for a year" challeng
to. Some related info: **https://iquilezles.org/artic**

Comments (101)

Sign in to post a comment.

FabriceNeyret2, 2022-09-02
@derhenryschmidt: I guess you have no idea who you are talking to (or you would find his contact info trivially). ;-)

derhenryschmidt, 2022-09-01
can we make Bussiness together? please give me your contact, email or discord or WeChat?

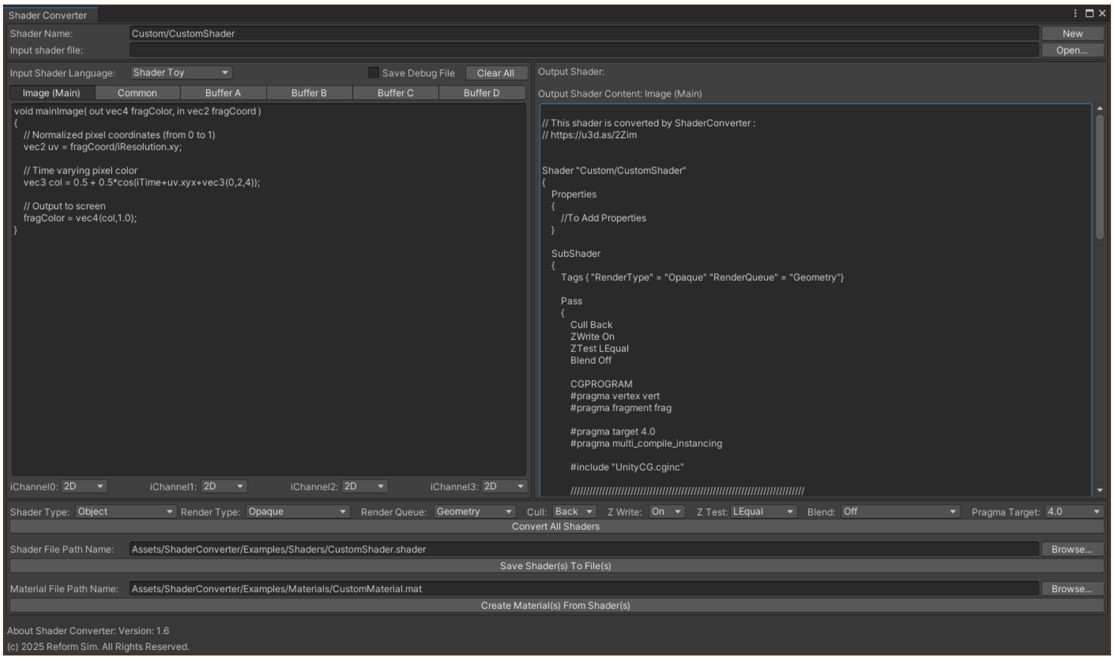
6) You can select the **"Shader Type"** of the main shader:

- **Full Screen**: The main image shader will be converted to full screen shader.
- **Object**: The main image shader will be converted to object space shader. The rendering effect can be translated or rotated with the object.

Notes: Only object space shader can supports both VR (Single Pass Instanced) mode and non-VR mode.

7) You also can change "**Render Type**", "**Render Queue**", "**Cull**", "**ZWrite**", "**ZTest**", "**Blend**", "**Pragma target**". In most cases, you can skip this step.

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



Shader Name: Custom/CustomShader
Input shader file:
Input Shader Language: Shader Toy
□ Save Debug File

```
void mainImage( out vec4 fragColor, in vec2 fragCoord )
{
   // Normalized pixel coordinates (from 0 to 1)
   vec2 uv = fragCoord/iResolution.xy;

   // Time varying pixel color
   vec3 col = 0.5 + 0.5*cos(iTime+uv.xyx+vec3(0,2,4));

   // Output to screen
   fragColor = vec4(col,1.0);
}
```

iChannel0: 2D iChannel1: 2D iChannel2: 2D iChannel3: 2D

Output Shader:
Output Shader Content: Image (Main)

```
// This shader is converted by ShaderConverter :
// https://u3d.as/2Zim


Shader "Custom/CustomShader"
{
   Properties
   {
      //To Add Properties
   }

   SubShader
   {
      Tags { "RenderType" = "Opaque" "RenderQueue" = "Geometry"}

      Pass
      {
         Cull Back
         ZWrite On
         ZTest LEqual
         Blend Off

         CGPROGRAM
         #pragma vertex vert
         #pragma fragment frag

         #pragma target 4.0
         #pragma multi_compile_instancing

         #include "UnityCG.cginc"

         ////////////////////////////////////////////////////////////
```

Shader Type: Object Render Type: Opaque Render Queue: Geometry Cull: Back Z Write: On Z Test: LEqual Blend: Off Pragma Target: 4.0

Shader File Path Name: Assets/ShaderConverter/Examples/Shaders/CustomShader.shader

Material File Path Name: Assets/ShaderConverter/Examples/Materials/CustomMaterial.mat

About Shader Converter: Version: 1.6
(c) 2025 Reform Sim. All Rights Reserved.

9) 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 named “ImageShaderExample”, it will show the Quad with the effect of shader.

10) 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. In some cases, you may need to change Color Space from Linear to Gamma to get the same effect.**

11) If shader has mouse or 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.

12) If shader has keyboard input, you should also add “**ImageShaderKeyboardInput**” component to the GameObject, and assign the “KeyboardInput” RenderTexture to the “Keyboard Input RT” field of the component. Then you can use the “KeyboardInput” RenderTexture as input for any material, such as Main(Image) or Buffer A/B/C/D material.
Note: There is an example scene named “**KeyboardInputShaderExample**” to demonstrate keyboard input.

13) 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 512/1024/2048 etc).
   - **Texture Name Type**: Which texture should set to.

   You also can adjust spectrum multiplier, offset and smooth, wave amplitude multiplier and offset. Then assign an audio clip to the Audio Source component. All Done.
   Note: There is an example scene named “MusicInputShaderExample” to demonstrate music

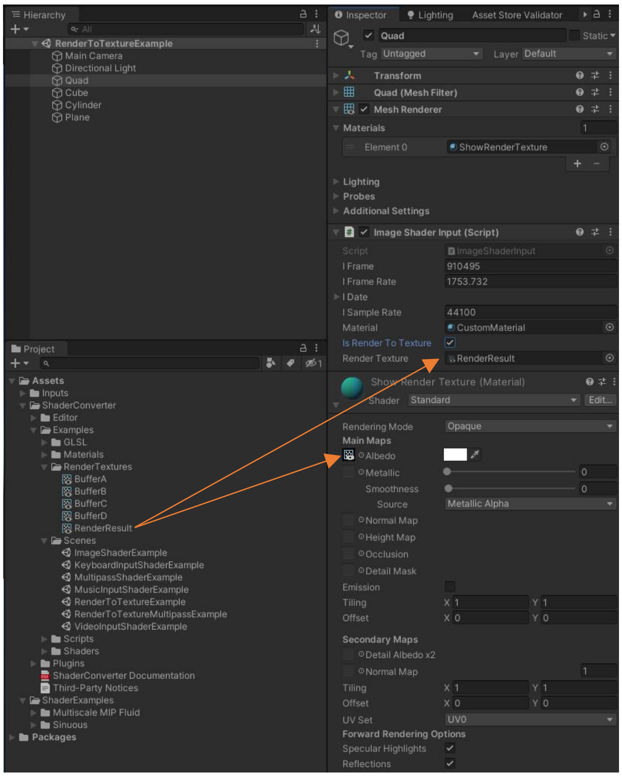
input.

14) If shader has one or more video input, you should also add one or more Video Player component to the GameObject. Assign your video clips to the Video Player components, then set the “Material Property” field to the corresponding texture property. All Done.

Note: There is an example scene named “**VideoInputShaderExample**” to demonstrate video input. If the color space of your project is Linear, you should uncheck “sRGB (Color Texture)” when importing video clip.

15) The shaders on the Shadertoy website are mostly full screen image shaders. If you want to render the effect to the RenderTexture. There is an example scene named “**RenderToTextureExample**” to demonstrate this feature.

   a) Enable the “Is Render To Texture” field of the ImageShaderInput component.
   b) Set the “RenderResult” RenderTexture to the “Render Texture” field of the ImageShaderInput component.
   c) Set the “RenderResult” RenderTexture to any material of any object, as shown in the following picture:



☑ Quad ☐ Static
Tag Untagged Layer Default

☑ Mesh Renderer
Materials 1
Element 0 ShowRenderTexture

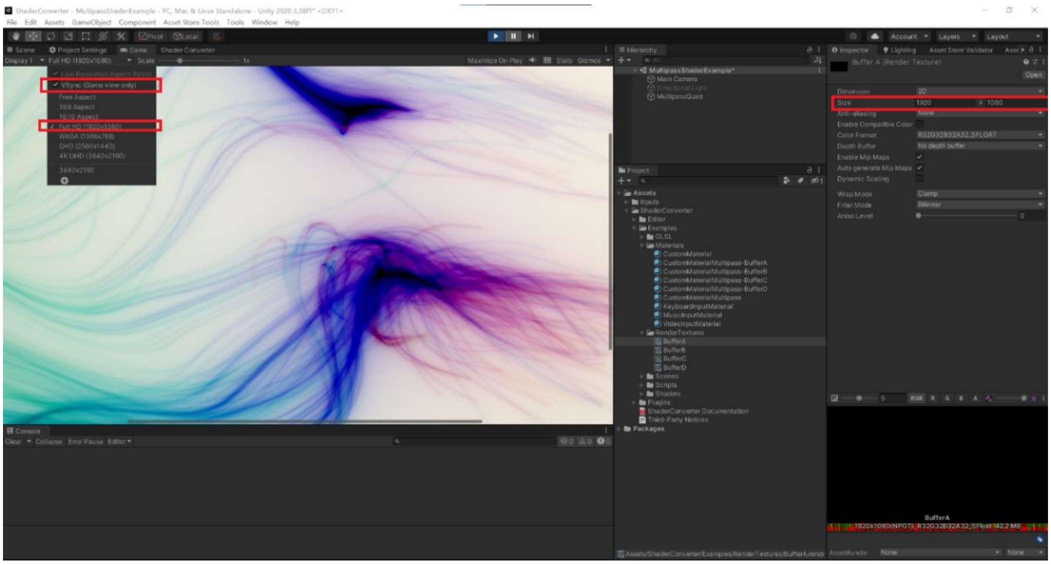
☑ Image Shader Input (Script)
Script ImageShaderInput
I Frame 910495
I Frame Rate 1753.732
I Date
I Sample Rate 44100
Material CustomMaterial
Is Render To Texture ☑
Render Texture RenderResult

Shader Standard
Rendering Mode Opaque
Main Maps
Albedo
Metallic 0
Smoothness 0
Source Metallic Alpha
Normal Map
Height Map
Occlusion
Detail Mask
Emission ☐
Tiling X 1 Y 1
Offset X 0 Y 0
Secondary Maps
Detail Albedo x2
Normal Map 1
Tiling X 1 Y 1
Offset X 0 Y 0
UV Set UV0
Forward Rendering Options
Specular Highlights ☑
Reflections ☑

Notes:

1) The resolution must be the same as the size of the RenderTextures (BufferA, BufferB, BufferC, BufferD, RenderResult).

2) The FPS must be the same as the Shadertoy website (60fps) to get the same rendering speed, as shown in the following picture:



Buffer A (Render Texture)

Dimension 2D
Size 1920 x 1080
Anti-aliasing None
Enable Compatible Color
Color Format R32G32B32A32_SFLOAT
Depth Buffer No depth buffer
Enable Mip Maps ✓
Auto generate Mip Maps ✓
Dynamic Scaling
Wrap Mode Clamp
Filter Mode Bilinear
Aniso Level 0

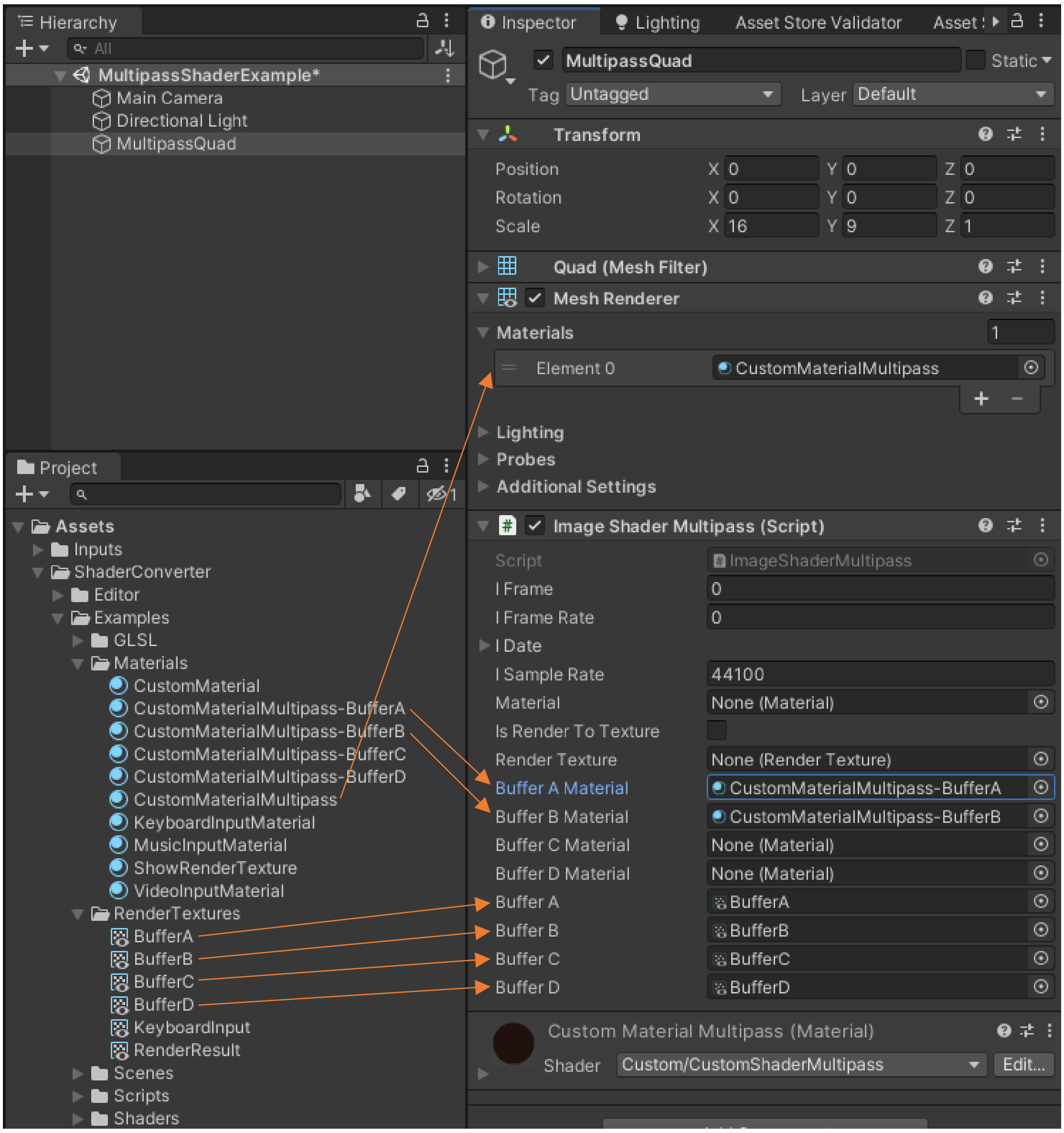
4.2. Converting Image shader with multiple passes

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 the material “CustomMaterialMultipass” to the MeshRenderer component.
7) Add “ImageShaderMultipass” component to the GameObject. **Note: The field “Material” of the “ImageShaderMultipass” component must be empty.** 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:



☑ MultipassQuad

Tag Untagged Layer Default

Position X 0 Y 0 Z 0

Rotation X 0 Y 0 Z 0

Scale X 16 Y 9 Z 1

Materials 1

Element 0 CustomMaterialMultipass

Script	ImageShaderMultipass
I Frame	0
I Frame Rate	0
I Date	
I Sample Rate	44100
Material	None (Material)
Is Render To Texture	□
Render Texture	None (Render Texture)
Buffer A Material	CustomMaterialMultipass-BufferA
Buffer B Material	CustomMaterialMultipass-BufferB
Buffer C Material	None (Material)
Buffer D Material	None (Material)
Buffer A	BufferA
Buffer B	BufferB
Buffer C	BufferC
Buffer D	BufferD

Shader Custom/CustomShaderMultipass

9) Assign RenderTextures or 2D/Cube 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.** You can download the Shadertoy built-in media files (texture, video, music…) from:

[Shadertoy media files | shadertoy - unofficial]

10) The shaders on the Shadertoy website are mostly full screen image shaders. If you want to render the effect to the RenderTexture. There is an example scene named "**RenderToTextureMultipassExample**" to demonstrate this feature.

a) Enable the "Is Render To Texture" field of the ImageShaderMultipass component.

b) Set the “RenderResult” RenderTexture to the “Render Texture” field of the ImageShaderMultipass component.

c) Set the “RenderResult” RenderTexture to any material of any object.

PS. When play the "MultipassShaderExample" scene in Unity 2020.3, don't select the "MultipassQuad" GameObject in Hierarchy window, and don't select the "CustomMaterialMultipass" materials in Project window. Otherwise the rendering result will flicker. Maybe it's a bug of Unity 2020.3. Unity 2021.3 doesn't have this issue.

5. Changelog

v1.7.1

- Fixed an issue where the multipass shader was not compatible with the linear color space;
- All Example shaders had been reconverted in linear color space;

v1.7

- Supported Gamma and Linear color space;
- **[Experimental]** Simple numeric global macros and constants defined in the shader can be extracted into the properties of the shader;
- Keyboard input can now be used as input of Buffer A/B/C/D material;
- Replaced the Unity built-in “_ScreenParams” with “_iResolution” of shader input to fix an issue that arises when switching Game View to Scene View;
- Fixed an issue where the “Material” field of ImageShaderMultipass component was set incorrectly in the MultipassShaderExample scene;

v1.6

- Object space shader supported both VR (Single Pass Instanced) mode and non-VR mode; All Example shaders had been reconverted to object space shaders to support VR;
- The default value of “Shader Type” parameter on the Editor window was changed to “Object”;
- Fixed an issue where Unity keyboard input was inconsistent with Shadertoy keyboard input;
- The UI layout of Editor window had been optimized;

v1.5

- Added two options in the window, now you can select the “Shader Type” of the converted main shader:
  (1) Full Screen: The main image shader will be converted to the full screen shader;
  (2) Object: The main image shader will be converted to the object shader. The rendering effect can be translated or rotated with the object;

v1.4

- Fixed the rendering issue of shader with sound/music input to get the same rendering result as same as possible with the Shadertoy output;

v1.3

- Added two example scenes to show how to apply the converted shader to any materials of any objects;
- Fixed the rendering issue of multipass shader. Changed all RenderTextures to enable Mip Maps to get the same rendering result as the Shadertoy output;
- Updated the document;

v1.2

- Supported video input, and added an example scene;
- Supported keyboard input, and added an example scene;
- Supported to set iChannelx type (2D or Cube) for each Buffer;
- Fixed an issue where the render texture of each pass/buffer was updated incorrectly;
- Updated the document;

6. Support

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