Table of Contents

mikejzx's 3D Anaglyph Pro Manual

1. Introduction

Thank you for purchasing 3D Anaglyph Pro! This Unity package provides you with an image effect to render stereoscopic anaglyphs from scenes in realtime. This manual document will explain how to use the effect, and describes in detail each of the settings you may wish to adjust. Please also review *Section 6. Important Notes* for important information regarding the effect. A demo scene is also provided in the asset to demonstrate the effect (see *Section 4. Running the Demo* for more information).

2. Demonstration Video

Basic usage demonstration: https://youtube.com/watch?v=fQaAiNZ8Iro

Setup guide with post-FX: https://www.youtube.com/watch?v=E_0ju-mX1hg

3. Getting Started

Applying the anaglyph effect is very easy, and is a matter of simply adding the AnaglyphCamera component to the camera you wish to render with the effect.

This is outlined in five short steps below for readers who would like explicit instructions:

i. In the scene Hierarchy pane, select the camera you wish to apply the 3D Anaglyph Effect to.

ii. With the camera selected, navigate to the bottom of the Inspector pane.

iii. Click "Add Component" in the Inspector pane. This should present you with a search box.

iv. Type "AnaglyphCamera" in the given search box, and click the resulting AnaglyphCamera effect component. If the effect does not appear in the results; please re-install the package.

v. The anaglyph effect is now active on the camera.

4. Running the Demo

A demo scene is provided in the *Demo* folder in the asset, and it demonstrates correct usage of the effect in it's simplest form. Click the play button in this scene to experiment with the anaglyph effect and see it in action. The demo features a first-person style camera, with the following controls:

W/A/S/D
: Forward, leftward, backward, and rightward movements respectively.

Mouse
: Adjust camera direction.

Mouse Scrollwheel Up/Down
: Increase/decrease anaglyph convergence distance.

E/Q
: Increase/decrease the eye separation distance.

Spacebar/Shift
: Upward and downward movement.

5. Use with Additional Post-Processing Effects

An alternative demo scene is also included with the package under the *Demo* folder, to demonstrate how the anaglyph effect can be used in addition to other post-processing effects.

The way this is achieved is by manually creating the second stereo camera that is required for the effect, and then manually applying all post-processing effects to it that exist on the main camera (except the 3D anaglyph effect itself!). The AnaglyphCamera component is put only on the main camera; **after** any other effects, and is configured to use the second stereo camera that was created, (via the *Advanced Effect Options* section of the script). It is recommended that the secondary camera's transform is made a child of the primary one. Please examine the demo scene to see how this advanced usage of the anaglyph effect can be achieved, and see *Section 7.3. Advanced Effect Options* in this manual.

The reason a "hack" like this is necessary to achieve anaglyph post-processing is because Unity provides no logical methods of copying the post-processing effects of one camera to another (in addition to post-processing effect scripts being handled differently to how they used to be, since the PostProcessingStackV2 was released), so we have to use this clunky workaround where the user manually manages the second camera.

6. Important Notes

6.1. Performance Implications

Due to the nature of how stereoscopic images are rendered; the anaglyph effect takes a significant toll on rendering performance. This is of course due to the fact that in order to generate a stereo image pair used to compose an anaglyph image, the scene must be rendered twice at the same resolution; this is obviously quite expensive to perform. This means that all geometry must be rendered twice, and hence the cost of rendering, including the cost of additional post-processing effects, becomes twofold. Please keep this in mind if you are targeting e.g. mobile platforms or devices with lower-end hardware. If this is an issue, please consider the following workarounds:

- Reduce the screen render resolution (or make easily-configurable to players of your game); this can be done in the Unity Player settings or via the Unity scripting API.
- If using additional post-processing effects, try remove any that are unnecessary or have a highly significant performance cost.
- Try to keep scene geometry as simple as is practically possible.
- Reduce the quality in Unity's quality settings, e.g. shadow quality, texture quality, etc.

6.2. Pipeline Notes

The effect only officially supports the Built-in Render Pipeline at the moment, which is the default render pipeline in Unity.

6.3. Render Path Limitations

The effect is known to work as expected in the default Forward (and Legacy Vertex-Lit) rendering paths. Deferred rendering may appear to work, and may be usable, however there are lighting bugs that become very obvious when the anaglyph effect is used with more extreme values. This is due to bugs in Unity itself not handling custom camera projection matrices properly.

6.4. Physical Cameras

Unity 2018.2 introduced the "Physical Camera" option for cameras, to simulate real-world camera settings. The anaglyph effect does not support this option, because Unity does not provide any documentation on the method used to generate the projection matrices for the physical cameras, and as such it is much more difficult to generate a projection matrix that is both derived from the physical properties and also usable for anaglyph rendering.

6.5. Occlusion Culling Bugs

Unfortunately, there appears to be some kind of bug in Unity that prevents it from culling objects correctly when the camera's projection matrix is set to wildly differ from it's original one. For this reason, issues of items "clipping" in and out of the scene may be prevalent if occlusion culling is enabled on the camera. This usually only occurs when the anaglyph effect is set to use extreme eye widths. If this becomes an issue for you, the only solution at the moment is to disable Occlusion Culling on the camera with the anaglyph effect. Depending on the scene, this could have a detrimental impact on performance.

6.6. Requirements for Builds

In order for the Anaglyph3D effect to work in player builds of your game, the Anaglyph3D.shader file **must** remain in a Resources folder (due to the way the Shader.Find Unity API method works). The asset package is structured like this for you already, so unless you deliberately modify the asset structure yourself, this needn't be of any concern.

7. Effect Options

This section will describe each of the various configuration options provided with the anaglyph effect. Additionally, methods of adjusting these properties during game run-time are also provided. Most of the properties have public members which you can modify through scripting.

7.1. Parallax Options

7.1.1. Eye Separation

The eye separation refers to the width at which the stereo images are taken. In some sense, it mimicks the separation between human eyes. However, using widths *greater* than that of the human eyes is often more desirable as it can produce a more vivid perception of depth in the scene. A sane base value to start from is around 0.1 m, and gradually increase from there as desired. Note that as this width is increased, it generally becomes more difficult for the eyes to focus on the scene due to the extreme separation. The recommended range for this value for most scenes is from 0.1 to 0.6 metres. Larger values may be desired in very long-distance-viewing scenes.

> This can be modified at scene run-time via the public `AnaglyphCamera.eyeSeparation` float property.

7.1.2. Convergence Distance

The simplest definition of this is the distance from the in-game camera that represents the physical location of the display (monitor, screen) in the scene. This means that any objects after this distance will appear "inside the screen", whereas those that are closer than the distance appear to "pop out" of the screen.

For most situations this value should not be set too high, as the sense of depth in the scene for nearer objects is generally reduced at higher convergence distances. The recommended value for this for most situations is from 1 to 10 metres, as this gives an illusion of "looking into the scene" as if it were being viewed through a physical "window." Much larger values of this option may be desired in longer range viewing.

> This can be modified at scene run-time via the public `AnaglyphCamera.convergence` float property.

7.2. Anaglyph Scheme Options

7.2.1. Scheme Type

This allows you to choose the set of colours you wish to render the stereo images in. Many options are provided for this option, those of note are described below.

The scheme type can be modified at scene run-time via the public `AnaglyphCamera.scheme` AnaglyphScheme enum property.

Classic Red/Cyan (Simple Colour Pairs)
: Produces a standard red/cyan anaglyph, with no additional filtering. The output image consists of the left image's red channel, and the right image's green & blue channels. This scheme is also referred to as a "Full-Colour Red/Cyan" scheme as it takes into account all the colour of the scene.

  This is a very common anaglyph scheme, however it is subject to ghosting and "retinal rivalry" (i.e. viewing discomfort) when highly-saturated colours are present in the scene.

Zhang-McAllister Red/Cyan (Dubois Matrices) (Recommended)
: This scheme applies an optimised red/cyan anaglyph matrix for LCD screens that was derived through the Dubois anaglyph matrix generation method, developed by Eric Dubois in 2001.

  This scheme is effective in eliminating ghosting in images, by replacing colours with colours that do not ghost.

  This particular matrix is derived from a paper by Z. Zhang and D. McAllister, entitled "A Uniform Metric for Anaglyph Calculation" (2006).

Sanders-McAllister Red/Cyan (Dubois Matrices)
: This scheme applies a Dubois matrix similar to the above, one that seems to be more commonly-used in software, despite originally being designed for CRT displays. It produces similar results to the above.

  This matrix was derived from a paper by W. Sanders and D. McAllister, entitled "Producing Anaglyphs from Synthetic Images" (2003).

Half-Colour Red/Cyan (Experimental)
: This scheme is a popular attempt to reduce ghosting by using the greyscale of the left image as the red channel in the output.

  The name is derived from the fact that only the right image is in full colour, whereas the left image is in greyscale.

Monochrome Red/Cyan (Experimental)
: This scheme produces monochrome/greyscale anaglyph images.

Optimised Red/Cyan (Experimental)
: This scheme is an attempt of reducing ghosting by not incorporating any of the red channel from the left image into any of the output colour.

Custom Colour Pair (Simple Colour Pairs)
: Select this option if you wish to generate a simple matrix from two colours. Selecting this will present you with two colour pickers to shade the left and right image.

  The custom colour pair can also be modified at scene run-time via the `AnaglyphCamera.customLeft` and `AnaglyphCamera.customRight` Color properties.

Specify Custom Matrix (Advanced)

This advanced option allows you to provide a custom anaglyph matrix of your own to the effect, for maximum control over the output of the image. This option is provided for advanced users who have matrices of their own they wish to apply.

Selecting this will present you with a field for entering the 3x6 anaglyph matrix values. The format of matrix may appear daunting at first, but is actually rather simple to understand, and the rows/columns have labels in the Editor to help make clear how it works.

The custom matrix can also be modified at scene run-time via the public `AnaglyphCamera.customMatrix` property, which is an 18-element array of floats representing the 3x6 matrix in row-major order.

A brief explanation of the values of the matrix is given below:

There are three rows in the matrix. The rows represent the red, green, and blue values that will be outputted to the final image. There are six columns, but it's best to think of them as two sets of three-columns—the left group of three columns is for the left-eye image, and the right group of three columns is for the right-eye image. As you may have guessed, the values in these columns represent a normalised factor (generally 0.0 to 1.0, but it is possible, in e.g. Dubois matrices, to have negatives or values that exceed 1.0) to multiply the red, green, and blue components of their image, and output the multiplied value to the output channel corresponding to the row.

The following example is a matrix that produces regular red/cyan anaglyphs:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

The example should be relatively simple to understand; the first row is the output red colour. The first column is the only column in the row that has any contribution, and it corresponds to the left image's red channel. Therefore all of the red in the image comes from the red channel of the left image. The second row represents the output green colour. The fifth column of the second row is the only column that contributes to it, and it corresponds to the green channel of the right image. The blue channel of the output has all of it's colour come from the blue channel of the right image.

7.3. Advanced Effect Options

These are options for advanced usage of the effect, in particular, the use of the anaglyph effect with additional post-processing effects.

7.3.1. Correct Colour Space?

(Linear colour space only)

Setting this option will enable the SRGB-to-linear, and linear-to-SRGB, colour space conversions in the shader. This produces more accurate colours in the output image (when using Linear colour workflow) but may come with a performance penalty, especially on weaker and older video cards. This is recommended when maximum colour accuracy is desired and when using Dubois matrices. This is not recommended when using Simple Colour Pair schemes.

This option **cannot** be modified at scene run-time.

This option has no effect when the project is using a Gamma workflow (this can be changed in the Unity Player settings).

7.3.2. Use Custom Camera?

If this is set to true, then the effect will prompt you for the specific camera you wish it to use for rendering of the second stereo image. The most common usage of this option will be to use a secondary camera that has post-processing effects applied to it, as are also applied to the main one.

> This option can be set at scene run-time with the `AnaglyphCamera.useCustomCamera` bool property. If set to true, then a custom camera should be assigned to the `AnaglyphCamera.customCamera` Camera property.

8. Support

Please thoroughly check this manual to see if your problem, or anything that sounds like your problem, is mentioned. If not, you may wish to e-mail skec@protonmail.ch for support, questions, bugs reports, or suggestions.