Circle Wipe Shader

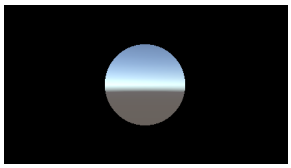
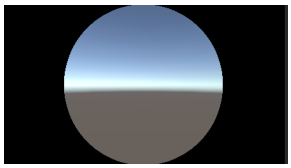
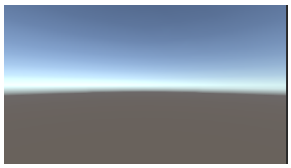
Introduction

Thank you for purchasing the Circle Wipe Shader! It's a full-screen shader that will allow you to do some nice scene transitions between two scenes. The package contains the full source code for the shader, the controller script that is placed on the Main Camera and supporting MonoBehaviour scripts to help illustrate how the shader can be used.

This ONLY works with the existing 3D project / 2D project. It does not work with the URP (Universal Rendering Pipeline).

See the YouTube videos on how I created this shader and the controller:
https://www.youtube.com/playlist?list=PLZo2FfoMkJeHvHrO6ZRDw9vynKyUL88tj

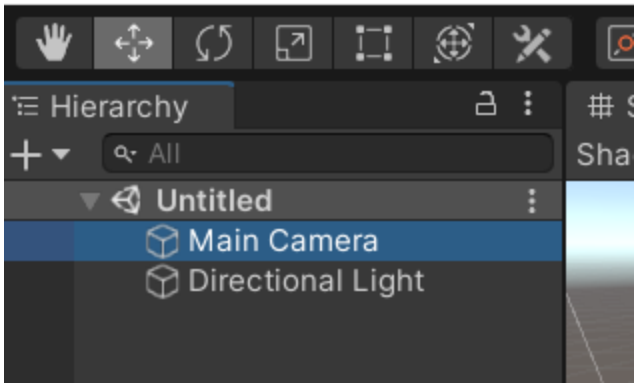
Here are some of the phases of the circle wipe based on the value of *Radius*:

			
Radius = 0	Radius = 0.25	Radius = 0.5	Radius = 2
Scene is completely blacked out			Scene is visible

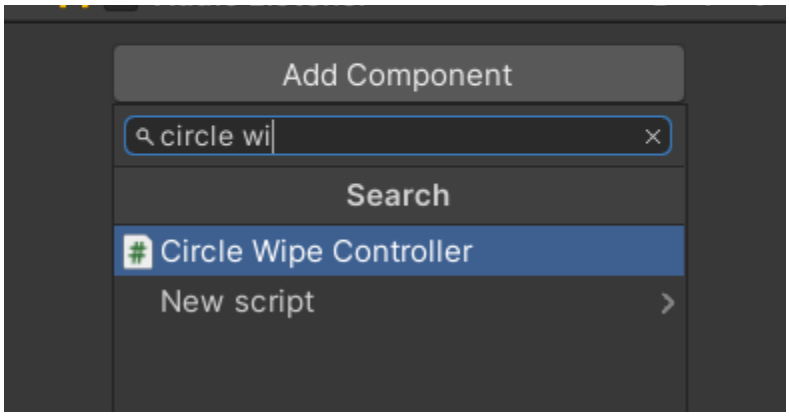
How To Use

To use the Circle Wipe Shader, follow the instructions below:

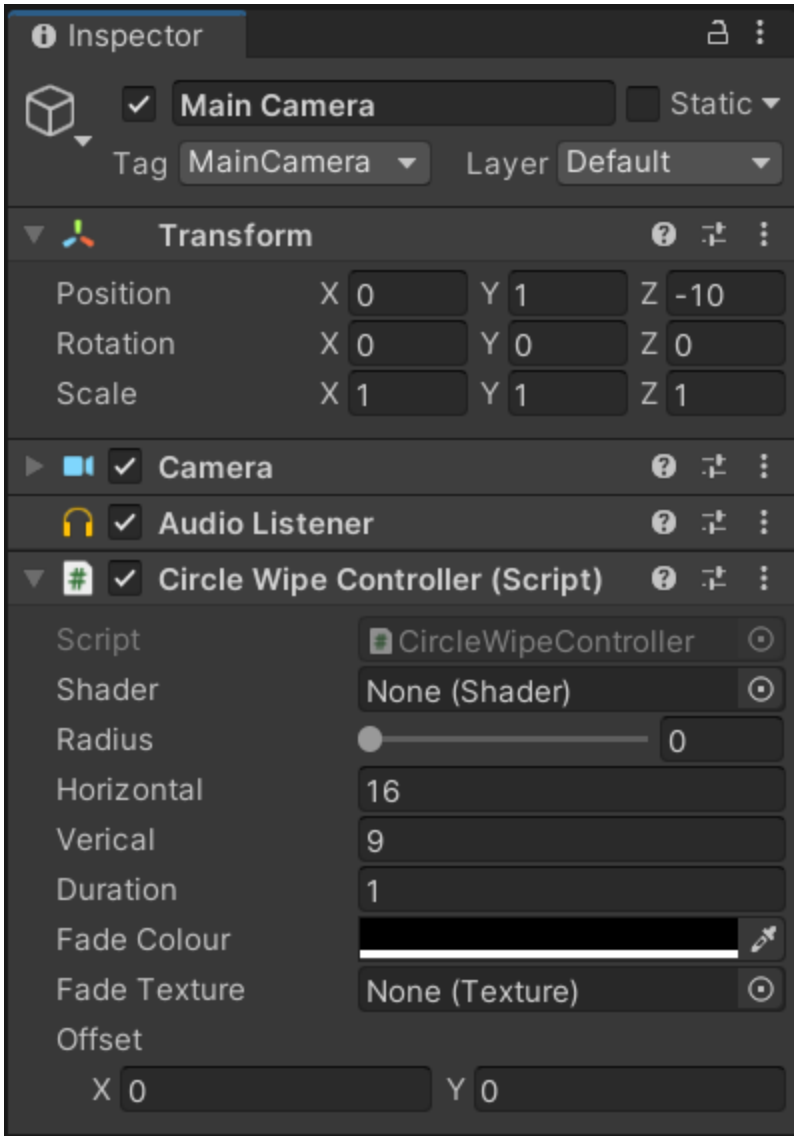
In the main scene click on the "Main Camera" game object:



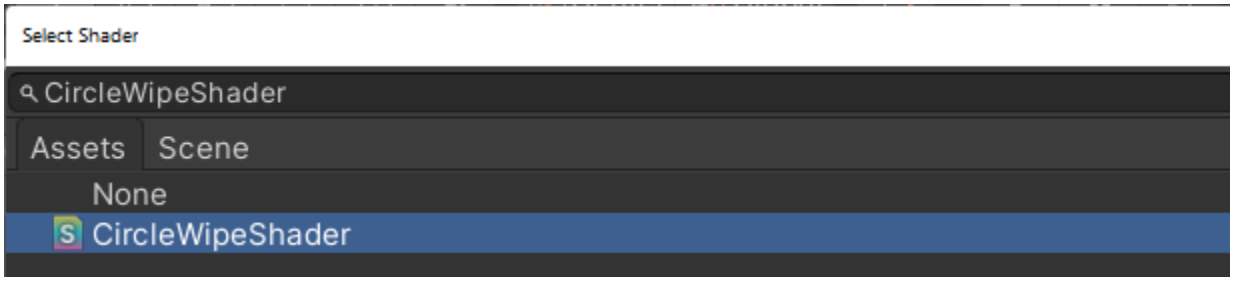
In the inspector view, click “Add Component” and search for the “Circle Wipe Controller” script. Add the script to the Main Camera game object by clicking on it:



The components on the Main Camera GameObject should look like this:



Click on the target circle icon to the right of the “Shader” field and choose the “Circle Wipe Shader” from the pick list. You can also search for “Circle Wipe” in the search box to make it easier to find:



Double click to add the shader.

The shader is the program that runs on the GPU and does the actual hard work, but the CircleWipeController allows us to easily set the parameters for the shader in our C# code.

The CircleWipeController Properties (at least the important ones)

Property	Use
Radius	The current radius of the circle wipe. Set this to zero at the start to hide your scene behind the wipe. Set this to 2 to fully show your scene.
Width / Height	The ratio of width to height. For most games, leave this as 16/9 for width and height respectively. This ensures that your circle is a circle and not an ellipse ;)
Duration	The amount of time it will take to fade from fully open to fully closed or vice versa. Use the FadeIn() and FadeOut() methods. Those methods assume that the radius is set correctly to either 0 for FadeIn() and 2 for FadeOut().
Fade Colour	The colour for the fade. If you are using a texture, set this to non-black to see it / tint the texture.
Fade Texture	Not required, but if set, a texture will be used instead of just the fade colour.
Offset	(0,0) is centre of the screen, (0.5, 0.5) is bottom right and (-0.5, -0.5) is top left. Sometimes you want to fade to a specific spot on screen. See the example AnywhereClick MonoBehaviour script.

Methods

There are three public methods on the CircleWipeController:

- FadeOut(Action callback = null)
- FadeIn(Action callback = null)
- UpdateShader()

Both FadeIn() and FadeOut() take an optional parameter of an action. The action is called (if passed in) at the end of the fade. This is illustrated in the FadeToNextScene script. FadeIn() takes the radius of the wipe from 0 to 2. FadeOut() takes the radius of the wipe from 2 to 0. It's important to set the radius correctly before calling FadeIn() or FadeOut() otherwise you might get incorrect wipes. E.g. a wipe could appear half way and then quickly reset before doing the fade.

The UpdateShader() should only be called if you're writing some custom code to alter the shader. It's automatically called by FadeOut() and FadeIn().

Example Scripts

There are example scripts included to get you started. These are added to the Main Camera game object and use the CircleWipeController. The scripts use the FadeIn() and FadeOut() methods to perform the transition. But you can easily write your own complex fades using the public properties and calling the UpdateShader() method on CircleWipeController.

The example scripts are:

- AnywhereClick - Click anywhere on the screen and it performs a fade out and then a fade back in. The fade is performed to the point clicked on the screen.
- SphereClick - Bonus. Not used. Click on the sphere game object and it will perform a fade out and then fade back in.
- FadeToNextScene - Performs a fade out and then loads another scene. The other scene uses the FadeInAfterDuration script.
- FadeInAfterDuration - Fades into the scene after a set wait time. This can be 0 seconds, just change *duration* property.

Example Scenes

There are:

- FadeToBlack - Uses the AnywhereClick to perform a fade to black / back again.
- FadeToTexture - Uses the AnywhereClick, but sets the Fade Texture so that the effect fades out to a texture instead of black. Auto fades back to scene.
- FadeToNextScene - Waits a certain amount of time and then loads the next scene after that wait and fade to black.
- NextScene - Fades up from black after a duration.