

Sci-Fi Stealth Shader

Overview.....	2
Setting up the asset.....	3
Applying the Stealth effect.....	3
How does it work?.....	3
StealthController.....	4
Demo scene controls.....	5
Shaders.....	6
Shader: SF Stealth.....	6
Shader: SF Simple Lit.....	8

Overview

The Stealth Shader is ideal for creating stealth effects for spaceships, characters, or any other game objects and provides a versatile and customizable invisibility effect that can be tailored to fit your sci-fi world. Whether you're developing a space adventure or a futuristic stealth game, this shader provides the perfect solution for all your invisibility needs.

Key Features:

Customizable Invisibility

Seamlessly blend objects into their surroundings with the stealth shader. Adjust the level of invisibility to suit your needs, making objects fully hidden or partially transparent.

Dynamic Fresnel Effect

Enhance the realism of your stealth effect with a customizable fresnel effect. Tweak the fresnel color and intensity to create stunning visual borders on the hidden objects.

Distortion, Rotation and Twirl Stealth Effect

Modify the look of your stealth effect by increasing or decreasing the distortion.

Also play around with rotation and twirl configuration that allow you to create unique stealth looks.

Dissolve Effect

Use the included noise textures to personalize the dissolving effect, giving you endless creative possibilities. Specify the rotation and twirl configuration of the dissolve noise.

Do not want to use noise textures? Simply disable this feature and use the built in simple noise effect of the shader.

User-Friendly Demo Scene

Jumpstart your project with the included demo scene. Featuring an intuitive UI, this scene allows you to modify most of the shader configurations in real-time and switching between different variants, showcasing the shader's full potential.

Diverse Noise Textures and Materials

Experiment with different stealth looks and dissolve effects using the provided noise textures and a variety of pre-configured materials. Each material demonstrates a unique combination of the shader's capabilities.

Ready-to-Use Script

The shader comes with a script to control the stealth mode, simplifying integration into your project.

Performance Optimized

Designed for the URP, the shader ensures smooth performance across various platforms without specific hardware requirements.

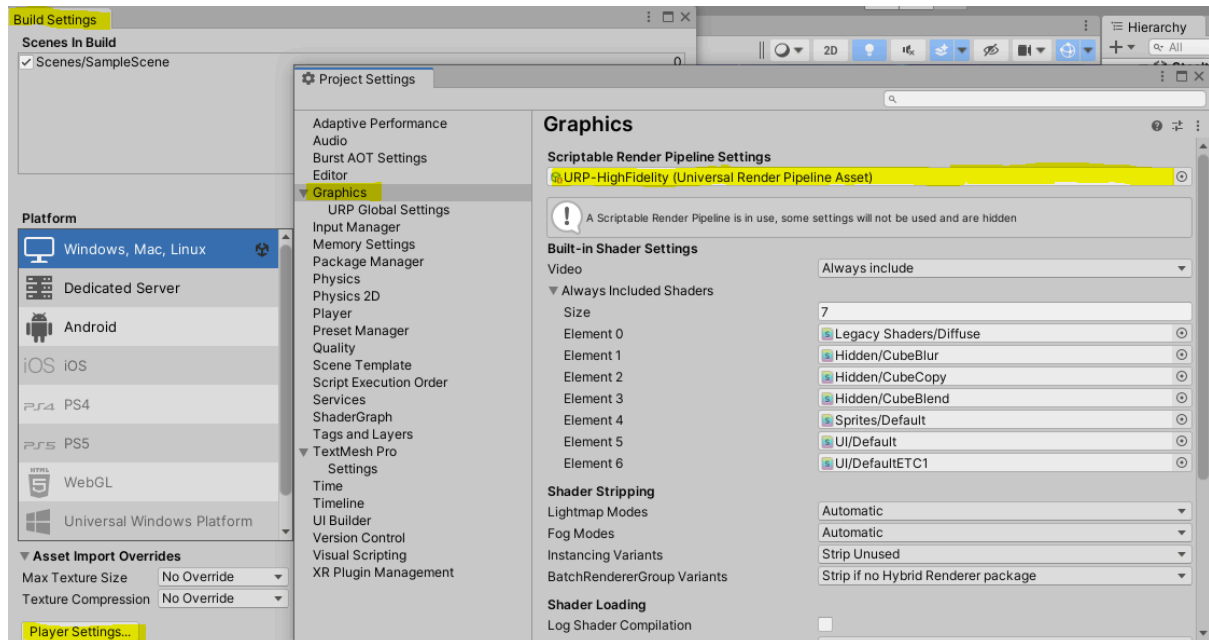
Intuitive Integration

Integrate the shader into your Unity project with user-friendly scripts and documentation. No need to be a coding expert – this asset is designed for easy implementation.

Setting up the asset

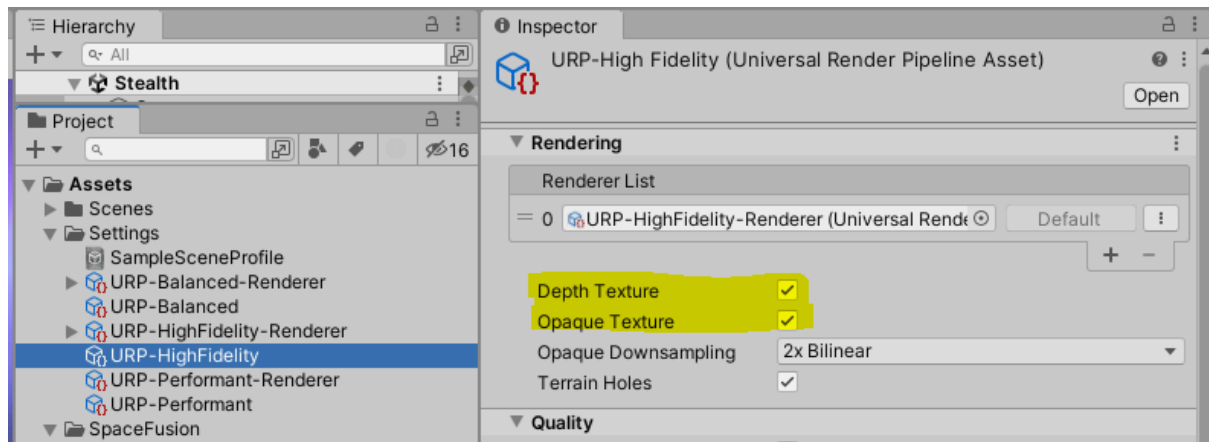
Make sure you have Shadergraph installed and are using it in an URP project.

Go to File → Build Settings → Player Settings → Graphics and double click on the assigned Render Pipeline Settings:



This will select the renderer and open it in the inspector.

Then make sure the Depth Texture and Opaque Texture is checked, to ensure correct functionality of the asset



Then feel free to open the included demo scene and try out the asset

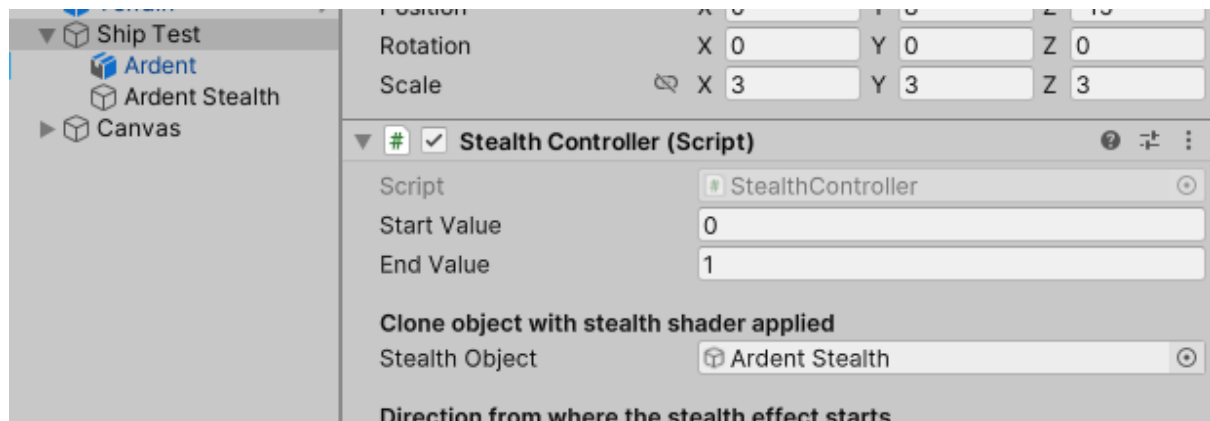
Applying the Stealth effect

How does it work?

The effect works by creating a copy of your object that should be able to change to stealth mode. All materials of the copy should have the Stealth Shader assigned.

The Stealth Controller script handles the activation and deactivation of the stealth mode.

The normal object and the stealth object should be on the same level and have the same size and position:



In this image you can see an example of how the structure looks like. The Ship Test holds the stealth controller. The original object and the stealth object are childs of the Ship Test object. The stealth object should be assigned to the stealth object property of the stealth controller.

StealthController

In this section we already can configure some properties of the actual shader. For more information about those properties, please go to the Shaders section under *Shader: SF Stealth* and check the detailed documentation for the shader properties.

The controller is responsible for handling activation and deactivation of the stealth mode. For the controller to be able to work, you need to specify some properties:

Stealth Object	As described in the previous section, you need to create a copy of the original object that has only Stealth Shader materials assigned. The controller will handle the configuration of all assigned stealth shader materials. This works also multiple MeshRenderers and for multiple materials per MeshRenderer
Activation Direction	Starting direction of the stealth effect. For more info check out the Shader section of this documentation
Size Offset	Depending on how we configure the activation direction and how the object is shaped, it can happen that the calculated object size is just a little bit too small, which can lead to a small portion of the object not being in stealth mode. This can happen if you specify multiple axes for the activation direction. To prevent this behaviour we can specify a small value which will be added to the objectSize property of the shader. For more info check out the Shader section of this documentation, in particular the ObjectSize property.
Activation duration	Duration in seconds for your object to fully transition into stealth mode
Deactivation duration	Duration in seconds for your object to transition from full stealth mode into normal mode

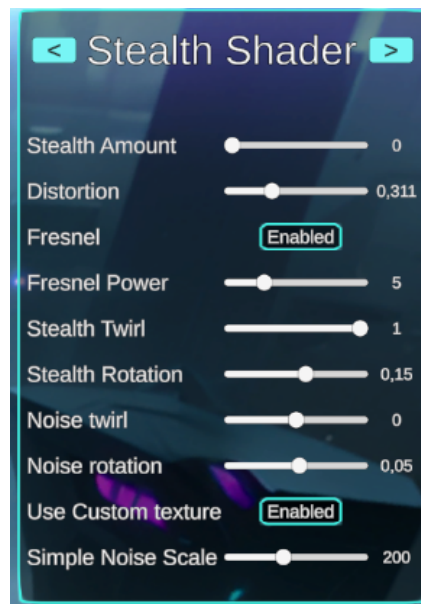
Demo scene controls

The Demo scene has a simple camera controller included.

Simply move around by following controls:

- W: move forward
- A: move left
- S: move back
- D: move right
- Q: move down
- E: move up
- Right click mouse and hold: rotate camera

Also there is a UI that allows you to switch between different stealth variants and configure the stealth shader on runtime:





Simply click on the left/right arrow next to the title to switch Between variants.


You can also press X to activate or deactivate the stealth mode, with the duration configured by the Stealth Controller.

Shaders

Shader: SF Stealth

Here you have an overview of the configuration options of this shader

General	
Object Size	<p>No need to configure manually, the StealthController script will handle this. This is needed so that we can properly apply the stealth amount to the object.E.g. Stealth of 0.5 means 50% of the object is already in stealth mode.</p> <p>With the stealth amount you can specify the amount in %. 0.5 means that 50% of your object is already in stealth mode. Unfortunately the Shader does not know the actual object size in the specified stealth activation direction. So we need to calculate it separately and set the proper object size so the shader can calculate the actual part of the object on which the mode needs to be applied.</p>
Stealth Amount	To activate the stealth mode. 0 = no stealth, 1 = full stealth mode
Activation Direction	Specify the direction in x,y and z coordinates from which the stealth effect will start
Stealth Mode	
Distortion	<p>Add some distortion effect to the stealth view so change the visibility. In the next image you can see a distortion of 0.25 on the left side and 1 on the right side</p> 
Fresnel Activated	Activate or deactivate the fresnel effect
Fresnel Power	<p>Change the intensity of the fresnel power. Higher values will result in much smaller and detailed glow of the corners. On the other side, decreasing the value will result in the whole object glowing.</p> <p>On the left side of the image you have a fresnel power of 2 and on the right side a fresnel power of 10</p> 
Stealth Color	Defines the HDR color of the fresnel glow

Stealth Twirl	This can be used in combination with the distortion effect to add some twirl to the effect.
Stealth Rotation	This can be used in combination with the distortion effect to add a slight rotation over time to increase the stealth experience on runtime.
Noise	
Edge thickness	<p>Defines the thickness of the dissolve noise The thickness in the left image is 0.15 and in the right one 1.</p> 
Noise Color	Sets the HDR color of the dissolve noise
Noise Rotation	Adds a rotation to the noise on runtime.
Noise Twirl	Adds a twirl effect to the noise.
Use Texture	<p>If true, the Noise Texture is used as dissolve noise If false, then a predefined Simple Noise shader node will be responsible for the dissolve noise</p>
Noise Texture	Configures the noise texture, only works if Use Texture is set to true
Simple Noise Scale	<p>Only active when Use Texture is set to false Configures the simple noise amount</p>

Shader: SF Simple Lit

In order to achieve the stealth effect there are some requirements for the original object.

To summarize the shader requirements in short:

- **Surface Type:** Transparent
- **Depth Write:** ForceEnabled
- **Depth Test:** LEqual

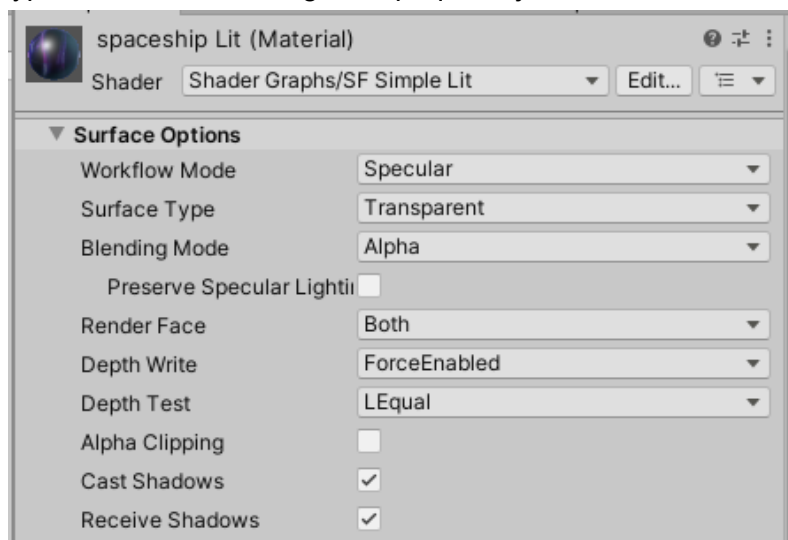
For those who are interested into a more detailed explanation:

Since the original object and the stealth object will be overlapping to achieve the stealth dissolve effect, we need to ensure that the original object has a transparent material assigned. This will ensure that the Stealth shader actually renders the real background, and not just the original object.

Since rendering complex transparent objects may have some visual impacts, like seeing stuff through the rendered object I introduced a simple Lit shader that addresses this issue. You can still specify the basic textures, like base, metal, normal and emission textures.

The only difference to the built-in URP Lit shader is that you are able to modify the Depth Write and Depth test property.

Based on these settings you should be able to render objects with the transparent surface type that are still looking like opaque objects.



You should be able to use any shader that has the Depth Write and Depth Test properties specified correctly for the original object.

Hope you enjoy this asset!

For any questions/ complaints/suggestions contact me under: [***gamedevibk@gmail.com***](mailto:gamedevibk@gmail.com)