GuntranHoet’s
Scanline Highlight Shader documentation

A word from the publisher

This shader is a result of a personal project. When the main features were implemented I decided to publish it in the Unity Asset Store and provide it to those who'd like to use it in their projects. I received positive input from the start and that motivated me to add more features and variants to this shader package.
Thank you for the support and motivation. sincerely.

About the shader

The Scanline Highlight Shader is a material surface shader that highlights your game objects with a virtual scanline and rim highlight. This shader was Inspired on your favorite post-apocalyptic and radiation filled RPG.

I aim to keep my shader both functional and performant. The end user should have maximum control with an intuitive user interface of parameters. That’s why I give every parameter a name that explains itself to give the user an easy time to get started with the shaders.

Technical notes

Texture tip

Some textures are only a mask (grayscale texture). That’s why I only extract the red (R) channel, so you can use the remaining channels (green, blue, alpha) for other masks you might need in your project. This will help you reduce your total texture size of your project.

Also, every parameter that takes in a 2D textures has the channels it uses displayed in the parameter name between parentheses “( )”.

Common parameters

All shader variants have the following features:

- Color
- Line texture
- Line tiling (x & y)
- Line speed (x & y)
- Highlight color
- Edge sharpness
- Edge highlight visibility
- Line visibility (or contrast)
- Highlight color overlay visibility

The “Color” (or “Main Color”) parameter is in most cases an overlay color for your albedo texture.

The line texture (“Lines (R)”) parameter can replace this mask texture with any mask you see fit. That’s why there is an option to both tile the mask in the X and Y (UV) coordinate directions. White will be the highlight and black will be what’s the base of the Main Color and Albedo parameters.

The “Line tiling” property lets you set the amount of times the line pattern is repeated over the screen.

The “Line speed” property slider [-4,4] lets you control the speed at which the lines move. Setting these to 0 will disable line movement.

The “Highlight color” parameter is the color value of your highlight that is shown on the edges of your model and on the white parts of your line texture mask.

The “Edge sharpness” parameter is a [0,16] slider which dictates the highlight spread / sharpness of the edge highlights. This is calculated with the view direction and the normals of the model.

The “Edge highlight visibility” parameter is a [0,1] slider which allows you to soften the effect of the edge highlight. Setting this to 0 will negate this effect.

The “Line visibility” parameter is a [0,1] slider which let’s you control the scanline intensity. Setting this to 0 will make them completely invisible.

The “Highlight color overlay visibility” is a [0,1] slider that sets the amount of minimum color blend between the base color (albedo * main color) and the highlight color. It gives the base color a subtle tint towards the highlight color.

Different shader variants

Diffuse shader

The diffuse shader is a basic version of the Scanline Highlight Shader. It only takes in a diffuse texture and overlays the scanline effect. The albedo map is visible through the scanline effect.

Diffuse Bump shader

The bump diffuse shader is an advanced version of the diffuse shader. It takes in an additional 2D texture which is the normal (or bump) map of the model. The albedo texture is visible through the scanline effect.

Monochromatic shader

The monochromatic shader is the flagship of this shader pack. It's one of the simpler shaders input wise and has the most distinct look of all. It only takes in one normal texture and no albedo texture.

The "Line visibility" parameter is replaced with a "Line contrast" slider [0,1] (default is 0.5). Note that there also is no "Highlight color" parameter. The resulting colors are calculated with the "Main color" and "Line contrast".

Monochromatic Unobstructed shader

This is the exact same as the monochromatic shader, but all objects with this shader will be drawn on top of other objects. This is very convenient when you want to highlight an object through a wall, to give an example.

Suggestion: To keep your game visually clean and uncluttered, I suggest you activate this shader via a script when the player is in the proximity. For this you might want to blend or switch between monochromatic shader and the monochromatic unobstructed shader.

Bicolor shader

The bicolor shader has no albedo texture and only (optionally) takes in a normal texture. The "Color" parameter dictates the base color of the model and the "Highlight color" parameter defines the highlight color. The colors are blended using the lines texture mask and the model normals.

PBR (metalness workflow) shader

The PBR (metalness) uses the same techniques as the bump diffuse shader, but with a physically based rendering workflow.

The “Metalness (R) & Smoothness(A)” parameter takes in one texture that represents both the metalness map and the smoothness/roughness map. The metalness information is extracted through the red (R) channel and the smoothness/roughness is extracted through the alpha (A) channel of the texture to reduce texture size (the same way Unity3D utilizes with the standard PBR shader).
You can pick between two smoothness types: ‘Smoothness’ map or ‘Roughness’ map in a dropdown box below the “Metalness (R) & Smoothness (A) texture” property.
A smoothness map is the Unity3D approach to the PBR metalness workflow, but some people prefer to use the roughness map over the smoothness map. The alpha (A) channel of the texture, when roughness is selected, is inverted in the shader. There is no need to manually invert this map to make it work in Unity3D.