WHAT IS "HOLOGRAM SHADERS PRO"?

Hologram Shaders Pro for URP is a collection of **10 new hologram shaders** for Unity, featuring glitch effects, scanlines, gridlines, and more! Compatible with all render pipelines.

USAGE

This shader pack uses Shader Graph to deliver a bunch of brand-new hologram effects for your games. Shader Graph should be pre-installed if you are using URP or HDRP, but it may not be on the Built-in Render Pipeline. Follow these steps to ensure you can use the pack:

1. Go to the Package Manager and select *Unity Registry* from the drop-down at the top left.
2. Find the **Shader Graph** package using the search bar on the top right.
3. Click the *Install* button on the bottom right if it is not installed. If the button says *Remove* instead, it's already installed!

The latest version of this asset was created using Unity 6.0.23f1 and Shader Graph 17.0.3, and it functions in the following major versions (you may need to update some graph nodes):

- Unity 6.0
- Unity 6.1
- Unity 6.2
- Unity 6.3
- Unity 6.4

PARAMETERS INCLUDED

The following parameters are available on shaders in the collection:

BASIC PBR PROPERTIES

These properties control the basic interactions between Unity's lighting and the object. Includes the crucial emissive color settings that control the appearance of the holograms.

- **Output Mode** – Controls whether the shader outputs color to Base Color, Emission, or both.
- **Base Color** – Controls the color of the object.
- **Base Texture** – Also controls the color of the object. This is multiplied by Base Emission Color.
- **Normal Texture** – Used to change the normal vectors on the surface, which influences lighting interactions. May not be noticeable due to the usage of bright emissive light.
- **Normal Strength** – A value between 0 and 1 which controls how strongly the Normal Texture impacts the surface normals.
- **Alpha Clip Threshold** – Any pixel with an output alpha value below this value is culled entirely. Useful for Dot Matrix holograms.
- **Metallic** – A value of 0 means the lighting on the surface is like a non-metal (wood, paper, plastic). 1 means it is like metal.
- **Smoothness** – A value of 0 means the object is completely rough. 1 means it is highly polished and smooth.
- **Ambient Occlusion** – A value of 0 means no ambient light reaches the object. 1 means the full ambient light reaches the object.
- **Use Color Separation**? – Choose whether to separate the red, green, and blue channels of the Base Texture.
- **Channel Separation** – How far (in UV space) to separate the color channels, if enabled. The red color channel UVs move in this direction, and the blue color channel UVs move in the opposite direction. The green color channel uses the original UVs.

SCANLINES

Some of the hologram shaders support a hologram texture scrolling down the object. This scanline controls the output alpha of the object, which in turn influences which parts of the object glow.

- **Scanline Mode** – An enum with three values: *Screen Space* means the Scanline Texture will be applied relative to the position of pixels on the screen; *World Space* means the scanline is applied based on the position of the object in the world; *None* will turn off the scanlines.
- **Scanline Rotation** – Only works in Screen Space mode. Rotates the direction of the scanlines.
- **Scanline Direction** – Only works in World Space mode. Orients the scanlines along this vector in world space.
- **Scanline Texture** – A texture which encodes the scanline pattern. You can increase the scaling of this texture for closely-packed scanlines, and vice versa. In World Space mode, only the vertical information matters.

- **Scanline Velocity** – Controls the speed at which the scanlines scroll across the object.
- **Scanline MinMax Alpha** – Completely black pixels from the Scanline Texture will output the Min value (first component of this vector). Completely white pixels from the Scanline Texture will output the Max value (second component of this vector).

VERTEX GLITCHES

Some of the hologram shaders support glitch effects. This type of glitch effect pulls individual vertices of the object to the side (away from the center of the object).

- **Use Vertex Glitches?** – A Boolean value – when on, vertex glitches will be displayed, and when off, vertex glitches are turned off.
- **Glitch Sensitivity** – How easy it is for a given vertex to fall under the random threshold and start glitching. Higher values mean a lower chance to glitch. This is a value between 0 and 1 that I recommend keeping very high (above 0.99) unless you want a lot of glitches.
- **Glitch Normal Multiplier** - A multiplier weight value applied to each component of the normal vector for calculating the offsets, e.g. (1, 0, 1) will pull vertices only along the X-Z plane.
- **Glitch Strength** – How far glitches vertices extend from their original position along their vertex normal.
- **Glitch Offset** – A time offset – you can use two materials with identical settings except a different offset, and they won't glitch at the same time.
- **Glitch Frequency** – How often glitches occur.

SEGMENT GLITCHES

This type of glitch effect takes a horizontal slice of the object and pulls all the vertices in the slice in a specified direction briefly.

- **Use Slice Glitches?** – A Boolean value – when on, slice glitches will be displayed, and when off, slice glitches are turned off.
- **Slice Width** – The width, in world space, along the y-axis that the glitch takes up.
- **Slice Speed** – How quickly the glitch effect scrolls down the object.
- **Slice Frequency** – How often glitches occur.
- **Slice Jitter** – A slight random offset which makes it appear as if the glitch jitters. Probably don't set it too high or the shader may flash too rapidly (capped to 0.2 by default).
- **Slice Duration** – How long a slice glitch event occurs for.
- **Slice Direction** – The direction in which all vertices in the glitch are pulled in world space.

NOISE

This type of glitch overlays randomly generated noise values to make the hologram appear low-fi.

- **Noise Speed** – How fast the shader scrolls through random noise values.
- **Noise Scale** – How fine or coarse the noise values are in screen space.
- **Noise Strength** – How strongly the noise colors are overlaid onto the mesh.
- **Noise Color** – A color tint applied to the noise values.

GRADIENT

Gradient holograms use a second set of color properties with a falloff value to produce a color gradient across the surface.

- **Gradient Space** – Should the gradient be calculated across Object, World, or Screen space?
- **Base Color 2** – Controls the color of the other side of the hologram.
- **Fresnel Color 2** – The color of the Fresnel highlight on the other side of the hologram.
- **Gradient Min-Max Y** – Which height values should be used as the start and end of the 'intermediate' portion of the gradient. Height values below the first (x) value output *Base Color*. Height values above the second (y) value output *Base Color 2*.

GRIDLINE OPTIONS

For grid-based holograms, these properties control the thickness and spacing of the gridlines.

- **Per Axis Strength** – The visibility of the gridlines along each axis. Set any one of these values to 0 to turn off gridlines along that axis.
- **Rotation Axis** – The axis around which world space rotates.
- **Grid Density** – How closely-packed the gridlines are to each other.
- **Line Thickness** – How thick each gridline is in world space units.
- **Line Falloff** – How much falloff there is along the edge of a gridline before the gridline is completely gone.
- **Grid Offset** – An offset of the grid in world space along the three axes.
- **Grid Velocity** – The speed at which the grid scrolls along the three axes. Can be set to 0.

DOT MATRIX PATTERN

For the dot matrix holograms, these settings control the density and spacing of the screen-space dots.

- **Dot Size** – The size, in pixels, of each dot on-screen. All dots are perfectly square.
- **Dot Space** – The size, in pixels, of the space between dots. The spacing is the same on each axis.

COMMON ROTATION PROPERTIES

Both the Grid-based and Dot Matrix-based holograms use this rotation system.

- **Rotation Radians** – The amount of rotation around Rotation Axis, in radians.

FRESNEL

Fresnel adds light to the edge/rim of an object. Most of the hologram shaders support Fresnel lighting.

- **Fresnel Color** – Color of the Fresnel/rim light. Can be different to the Base Emission Color.
- **Fresnel Power** – The size of the Fresnel/rim. Higher values mean the rim is smaller, but this should be a value above 0.

- **Use Scene Intersections** – Should the shader add a highlight to parts of the object that intersect other objects? This functionality depends on the depth buffer to detect intersections.
- **Intersection Power** – How sensitively the shader should detect intersections. Higher values mean a thinner intersection.

TIME PROPERTIES

Some shaders which use a time component can choose to use unscaled time. Unity does not provide an unscaled time value to shaders – it must be supplied via script to the shader. This is useful in cases where you wish to change Time.timeScale, but still allow shaders to animate at usual speed.

- **Use Unscaled Time** – Should the shader use regular time, or unscaled time? Regular (scaled) time values are affected by Time.timeScale as set within a C# script.

DYNAMIC RESOLUTION SETTINGS

Certain holograms depend on the screen resolution. When using dynamic resolution (FSR or DLSS), Unity uses the pre-upscaling resolution in shaders, resulting in incorrect results for some shaders. These properties let you adjust the resolution used in shaders.

- **Upscaling Amount** – How big the pre-upscaling resolution is, as a proportion of the full screen size. Set this to 1 if not using upscaling. Set this to 0.5 if you are using 50% resolution. *For the best results, this value may need to be set at runtime using scripting.*

UPDATES & PATCH NOTES

PATCH 1.2.1

- Added new option to separate color channels of the Base Texture.
- Minimum supported Unity version in future updates will be Unity 6.0.

PATCH 1.2.0

- Added vastly improved editor GUI for each shader.
- Added output option for hologram colors: Base Color, Emission, or Both.
- Vertex glitches can move with a multiplier for each component of the normal vector, e.g. glitches can be locked to the XZ plane with a multiplier of (1, 0, 1).
- Added sorting priority and instancing settings to shader editor window.

PATCH 1.1.0

Added new hologram types and functionality to existing shaders.

- Most holograms (except Dot Matrix holograms) have a new Scene Intersection functionality.
- Added new Noise Hologram which applies visual noise to the effect.
- Added new Gradient Hologram which uses color gradients for base emission and Fresnel effects.

- Added ability to apply unscaled time to some effects. Requires external scripting to apply the unscaled time values.

PATCH 1.0.1

- Added new **Upscaling Amount** property for dot matrix shaders to account for dynamic resolution.

PATCH 1.0.0

First release. Added eight types of hologram shader (some with common features):

- Basic
- Glitch
- Scanline
- Uber
- Grid
- Grid-Glitch
- Dot Matrix
- Dot Matrix-Glitch