Vertex-lit and vertex-colored shaders v2.4.1

---

1. Overview

This shader pack contains a number of shaders you can use to replicate the look and inner workings of those good old nostalgic Nintendo 64 or PS1 graphics.

---

2. Main features

Where applicable, the shaders in this pack support the following features:

- **Vertex lighting**
  Most of these shaders use per-vertex lighting instead of per-pixel. This is the key to the hard, diamond-shaped shading artifacts that make old games look the way they do.
- **Vertex colors**
  Most of these shaders support vertex colors. This is a way of painting models without using textures. Vertex coloring was used in N64 games to fake lighting and shadows while saving up on texture storage space. You can paint or bake vertex colors in Blender and render them in Unity thanks to these shaders.
- **Base color**
  An optional color override for meshes that are not vertex painted or don't come with vertex colors. Without this base color, most models will be white.
- **Fog**
  All of these shaders support fog. Older 3D games used fog to avoid having t aspect of the N64/PS1 look.
- **Alpha clip / Alpha blend**
  Most of the transparent shaders come in two variants: alpha clip and alpha blend. Use alpha clip to get those jagged/pixelated edges that were so common back in the 90's. Use alpha blend for special cases where alpha clip would look too ugly - for example, the glowing aura of a Zelda fairy.
- **Lit/unlit**
  Most of these shaders support an "Unlit" option. When "Unlit" is checked, the material will be rendered at full brightness, without any shadowing. This is useful for things that glow in the dark, or for textures that already have shadows baked in.
- **Retro texture filtering**
  All of these shaders support the iconic “N64” and “PS1” texture filtering modes.
- **Fake environmental reflections / Sphere mapping**
  Any opaque material can be set to sphere mapping, allowing you to recreate the fake environmental reflection effect used for Metal Mario in Super Mario 64. You will find two free environment maps in the demo folder.
- **Overlay color**
  An optional color overlay that allows you to replace a model's base texture or color with another color. This is useful for making characters flash white/red when they get hit.

---

3. Basic usage

To apply one of the shaders to a model or sprite:

1. Create a new material
2. Ensure you have the material selected and the Inspector tab open
3. Open the Shader dropdown
4. Select "Vertex-lit Shaders"
5. Pick the shader of your choice. See "List of shaders" below to compare the various shaders included in this pack and select the one that best fits your use case.
6. Set the shader parameters (texture, etc)
7. Drag and drop the material to your model or sprite, or select it under the **Mesh Renderer** properties

---

4. List of shaders

Full list

Vertex-lit & Vertex-colored/
Opaque/
Dual texture
Vertex-lit Alpha Blend
Vertex-lit Alpha Clip
Vertex-lit Modulate
Vertex-lit Modulate Texture Scrolling
Transparent/
Alpha Blend
Alpha Clip
Billboard Alpha Blend
Billboard Alpha Clip
Animated Billboard Alpha Blend
Animated Billboard Alpha Clip
Legacy (do not use)/
Sphere Mapped

Opaque > Dual Texture

Allows you to transition between two different textures on the same model, with vertex lighting. I used this on the ground tiles in a digging game to transition between the default state and the "dug" state.

Parameters:

- **Base color:** Optional diffuse color (replaces vertex colors if opaque)
- **Primary texture (RGBA):** A RGBA texture
- **Secondary texture (RGBA):** Another RGBA texture.
- **Mix factor:** A number between 0 and 1, where 0 shows only _TexA, and 1 renders only _TexB.
- **Overlay color:** An optional override color.
- **Unlit:** Disables shadowing.
- **Texture mapping:**
  - **Default:** Follows the model's UV coordinates.
  - **Sphere mapped:** Fakes environmental reflections by using the main texture as an reflection map.
- **Texture filtering:** How pixels in the texture are interpolated.
  - **Default:** Uses the texture's "filter mode" parameter.
  - **N64:** Replicates the N64's iconic three-point filtering mode
  - **PS1:** Replicates the PS1's iconic affine texture mapping

Opaque > Alpha Blend/Clip Texture

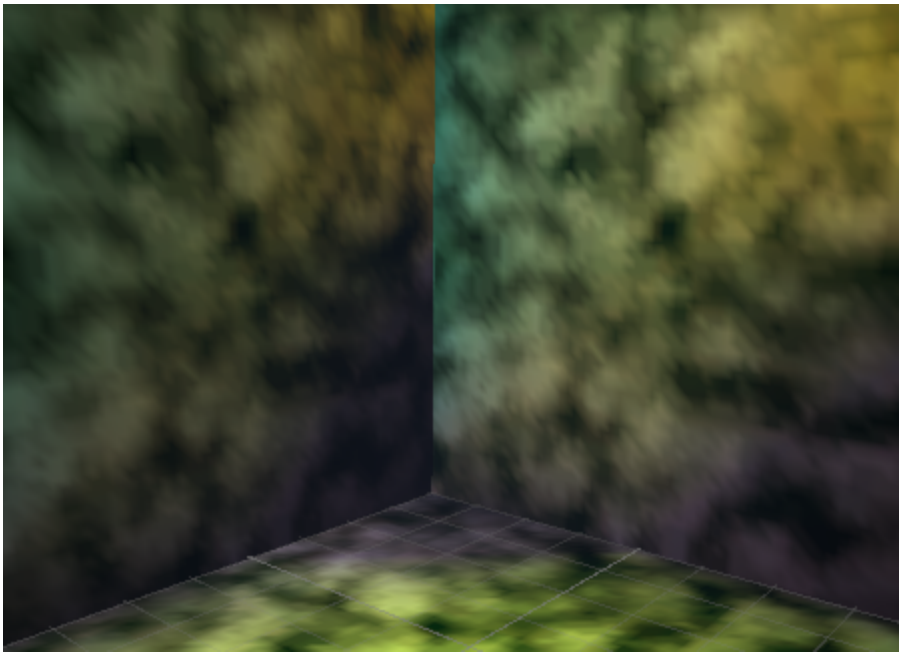
Overlays a texture and a color (optional) on top of a model's vertex colors.

Parameters:

- **Base color:** Optional diffuse color (replaces vertex colors if opaque)
- **Primary texture (RGBA):** A RGBA texture.
- **Overlay color:** An optional override color.
- **Unlit:** Disables shadowing.
- **Alpha cutoff:** (alpha clip only) Make all pixels below this opacity transparent.
- **Texture filtering:** How pixels in the texture are interpolated.
  - **Default:** Uses the texture's "filter mode" parameter.

    - **N64:** Replicates the N64’s iconic three-point filtering mode
    - **PS1:** Replicates the PS1's iconic affine texture mapping
- **Texture mapping:**
    - **Default:** Follows the model's UV coordinates.
    - **Sphere mapped:** Fakes environmental reflections by using the main texture as an reflection map.

Opaque > Modulate Texture



Multiplies the given texture with the model’s vertex colors. This is ideal for environments with pre-baked (or painted-on) vertex lighting.

Parameters:

- **Base color:** Optional diffuse color (replaces vertex colors if opaque)
- **Primary texture (RGBA):** A RGBA texture.
- **Overlay color:** An optional override color.
- **Unlit:** Disables shadowing.
- **Texture filtering:** How pixels in the texture are interpolated.
    - **Default:** Uses the texture’s “filter mode” parameter.
    - **N64:** Replicates the N64’s iconic three-point filtering mode
    - **PS1:** Replicates the PS1's iconic affine texture mapping
- **Texture mapping:**
    - **Default:** Follows the model's UV coordinates.
    - **Sphere mapped:** Fakes environmental reflections by using the main texture as an

reflection map.

Opaque > Modulate Texture Scrolling

Like Modulate Texture, but moves the texture over time. Great for flowing surfaces like rivers!

Parameters:

- **Base color:** Optional diffuse color (replaces vertex colors if opaque)
- **Primary texture (RGBA):** A RGBA texture.
- **Overlay color:** An optional override color.
- **Scroll Speed (X):** UV displacement per second in the X axis
- **Scroll Speed (Y):** UV displacement per second in the Y axis
- **Unlit:** Disables shadowing.
- **Texture filtering:** How pixels in the texture are interpolated.
  - **Default:** Uses the texture's "filter mode" parameter.
  - **N64:** Replicates the N64's iconic three-point filtering mode
  - **PS1:** Replicates the PS1's iconic affine texture mapping
- **Texture mapping:**
  - **Default:** Follows the model's UV coordinates.
  - **Sphere mapped:** Fakes environmental reflections by using the main texture as an reflection map. See 6.2 for how to make sphere maps.

Transparent > Alpha Blend/Clip

Renders the mesh with a transparent texture.

Parameters:

- **Base color:** unused
- **Primary texture (RGBA):** A RGBA texture.
- **Overlay color:** An optional override color.
- **Unlit:** Disables shadowing.
- **Alpha cutoff:** (alpha clip only) Make all pixels below this opacity transparent.
- **Texture filtering:** How pixels in the texture are interpolated.
  - **Default:** Uses the texture's "filter mode" parameter.

- **N64:** Replicates the N64’s iconic three-point filtering mode
- **PS1:** Replicates the PS1's iconic affine texture mapping

Transparent > Billboard Alpha Blend/Clip

Made for 3D sprites. Renders a sprite as a billboard (always faces the camera) with alpha blending or clipping. Great for Mario 64 style pickups. Works best with Unity's built-in **Quad** mesh.

Parameters:

- **Base color:** unused
- **Primary texture (RGBA):** A RGBA texture.
- **Overlay color:** An optional override color.
- **Unlit:** Disables shadowing.
- **Alpha cutoff:** (alpha clip only) Make all pixels below this opacity transparent.
- **Texture filtering:** How pixels in the texture are interpolated.
  - **Default:** Uses the texture’s “filter mode” parameter.
  - **N64:** Replicates the N64’s iconic three-point filtering mode
  - **PS1:** Replicates the PS1's iconic affine texture mapping

Transparent > Animated Billboard Alpha Blend/Clip

Displays an animated texture from a sprite sheet frame by frame, facing the camera.

Parameters:

- **Base color:** unused
- **Primary texture (RGBA):** A RGBA texture.
- **Overlay color:** An optional override color.
- **Tiling X/Y:** Number of columns/rows in the sprite sheet.

- **Interval:** Time per frame, in seconds.
- **Unlit:** Disables shadowing.
- **Alpha cutoff:** (alpha clip only) Make all pixels below this opacity transparent.
- **Texture filtering:** How pixels in the texture are interpolated.
  - **Default:** Uses the texture's "filter mode" parameter.
  - **N64:** Replicates the N64's iconic three-point filtering mode
  - **PS1:** Replicates the PS1's iconic affine texture mapping

Legacy (do not use) > Sphere Mapped (deprecated)

Initial implementation of environment mapping. No longer supported. Use any of the Opaque shaders and set the texture mapping to "Sphere mapped".

---

6. Tips for perfecting the retro look

6.1. Unity settings

These shaders alone are not enough to replicate the look of N64/PS1/PS2 graphics. To complete the illusion, you need to enforce certain settings on your resources.

For every texture, set the following settings:

- Maximum size:
  - for N64 style: 64 or less
  - for PS1 style: 256 or less

For every light:

- Shadow type: No Shadows

For every scene:

1. Go into Window > Rendering > Lighting
2. In the Scene tab create a new Lighting Settings and save it alongside your scene
3. Switch to the Lighting tab if not already open

4. Select the Environment tab
5. Go into Other Settings
6. Enable Fog
7. Set the fog color, mode, and other parameters to your liking.

6.2 Make your own environmental reflection maps in GIMP

Some N64 games used sphere mapping to fake environmental reflections. This technique uses a single spherized representation of the environment as a texture, rather than the six-sided cube maps of later generations.

To make a n environmental reflection map in GIMP, follow these steps:

1. Find or create an image of the desired environment (e.g. a landscape shot or a wide-angle shot of your location)
2. Open the image in GIMP
3. Crop the image into a square
    1. Select the **Crop** tool
    2. In the tool properties, enable **Fixed Aspect ratio**
    3. Set the aspect ratio to **1:1** (square)
    4. Drag the crop rectangle over the region you want to crop
    5. Press Enter
    6. Go to **Layer > Layer to image size**
4. Prepare the image
    1. Adjust the contrast to your liking in **Colors > Brightness/Contrast** or **Colors > Levels**.
    2. To simulate a dull, unpolished metal surface, apply a blur filter to the image (**Filters > Blur > Gaussian blur** or **Filters > Blur > Lens blur**). The duller the metal, the higher the blur.
    3. To simulate a shiny, polished, mirror-like surface, don't apply any blur.
5. Spherize
    1. Go to **Filters > Distort > Spherize**
    2. Set the parameters as follows:
        1. Mode: Radial
        2. Angle of view: 90
        3. Curvature: 1.000
        4. Amount: 1.000
        5. Resampling method:
            1. For dull metal: Cubic

        2. For polished metal: LoHalo
    3. Click on OK
6. Go to **Image > Scale Image**, and scale the image to the final texture dimensions:
    1. **N64:** up to 64x64
    2. **PS1:** up to 256x256
7. Go to **File > Export**, and save the exported image into your Unity project as a PNG file.

6.3. Additional assets

Finally, consider using my fast retro CRT effect shader: https://assetstore.unity.com/packages/vfx/shaders/fullscreen-camera-effects/fast-crt-tv-shader-266808

---

7. Troubleshooting

7.1. Materials are rendered in bright purple/pink

This could indicate that one or more shaders are missing or have errors.

Check the Unity console for errors. If the Unity console is missing, you can enable it at:

Window > General > Console

7.2. Console error: Couldn’t open include file

If you see this error:

```
Console error: "Shader error in 'Vertex-lit & Vertex-colored/...': Couldn't
open include file '../Common/VertexShadersCommon.cginc'"
```

It means you’re missing the file `VertexShadersCommon.cginc`, which the shaders in this pack depend on. To fix this error:

1. Reimport the shader pack from the Unity Package Manager
2. When reimporting the shaders, make sure to import every file under this directory:

```
OddlyShapedDog/VertexLitShaders/Shaders/Common/
```

7.3. Filtering modes and sphere mapping don't work in Unity 6

This is caused by a change in how Unity 6 handles shader includes. Namely, it now includes any `#pragma` statements in includes.

Here's how you fix this:

1. **If you have not modified the shaders in this shader pack:** Just download and install the latest version of this shader pack (2.4.1 or higher).
2. **If you have customized the shaders for yourself and don't want to overwrite your customizations:**
    1. Open `OddlyShapedDog/VertexLitShaders/Shaders/Common/VertexShadersCommon.cginc` in your editor of choice
    2. Copy the three lines starting with `#pragma`
    3. For each of the vertex-lit & vertex-colored shaders:
        1. Open the corresponding `.shader` file in your editor
        2. Look for the line `#include "../Common/VertexShadersCommon.cginc"`
        3. Insert a new blank line ***before*** the `#include` line
        4. Paste the `#pragma` lines you copied earlier into the new line
        5. Save the file

7.4. Further troubleshooting

Email us at support@oddly.shaped.dog if you have any feedback or questions on any of our assets.