Photogrammetry Workflow Layered shader

Author: Sébastien Lachambre, Sébastien Lagarde, Developer - Rendering Research

Date: 2017–09–27

Layering materials in Unity

Contents

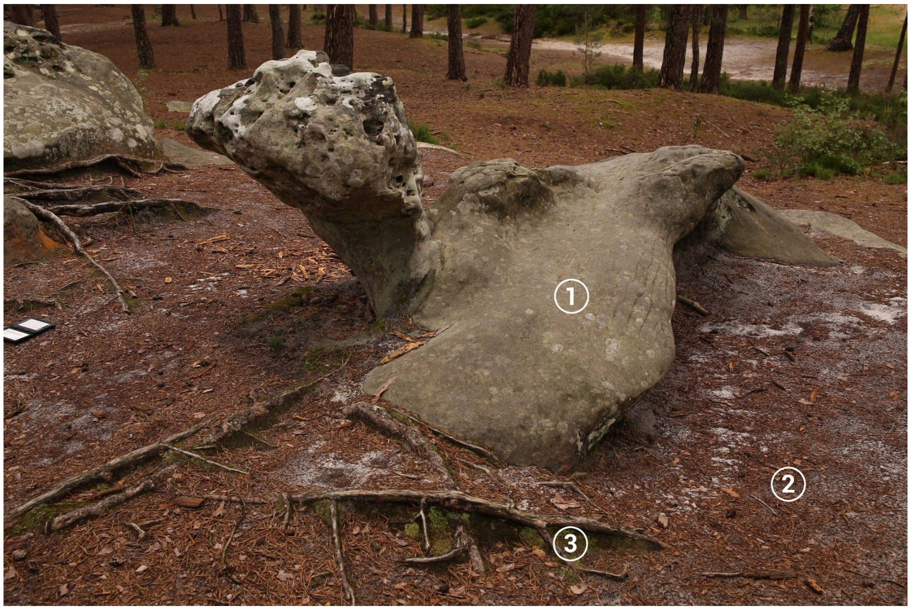
1. Overview

This document builds on the Unity photogrammetry workflow guide, which shows you how to capture objects and materials from the real world using photogrammetry.

This step-by-step guide shows you how to use a layered shader to achieve the same level of quality as shown in the photogrammetry workflow guide, but optimizes on texture memory budget to cater for your gaming needs.

The example used in this document contains:

- A main object made of rock, ground and tree root
- Various materials:
  1. Stone
  2. Ground elements (mix of spines)
  3. Moss



Photogrammetry allows you to get a qualitative result, but requires a very high texture resolution to conserve details, as shown here:

16384*16384

This is unpractical for game authoring due to memory budget, and it doesn't allow you to add any variation to the object.

To add variation to objects, you can use any of the following shaders:

- Regular shader
- Layered shader
- Layered shader for material layering

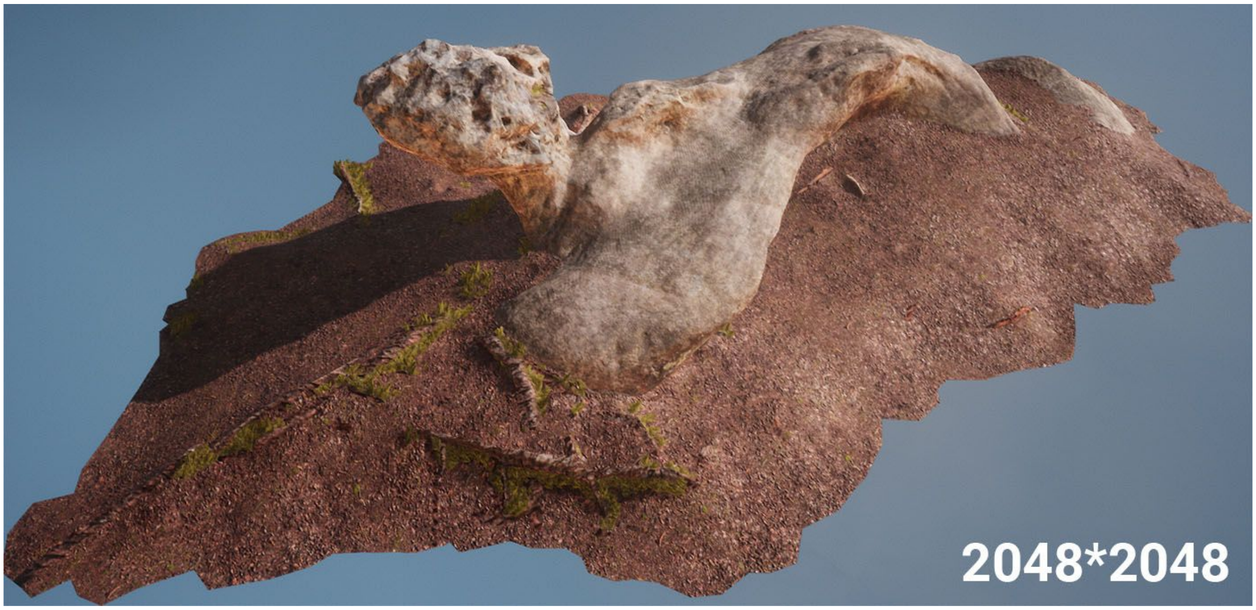
Regular shaders

A regular shader defines the visual for one material. Displaying just the game object with its material output, as shown in Unity Photogrammetry Workflow - 4.1 How to capture an object, can cause poor qualitative results because all materials are combined.

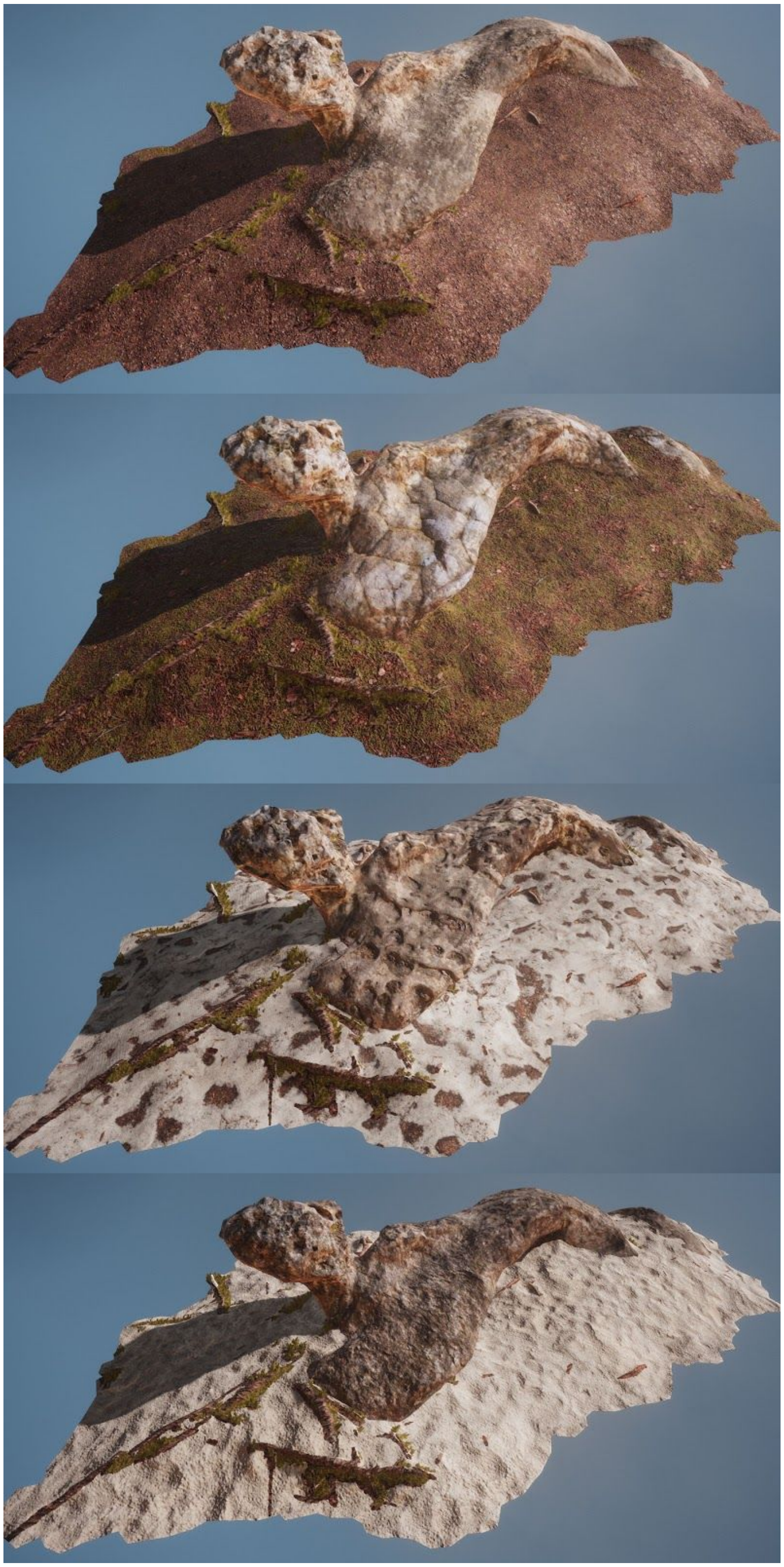
In this case, the material is a combination of stone, ground element, and moss):

Layered shaders

A layered shader defines the visual with a combination of individual materials. The main material that is shown in the picture above is mixed with other materials of stone, ground element and moss. These other materials are tileable. This means they can wrap around objects and you can reuse them on different objects. Using a combination of materials enables you to have a similar visual quality as a high resolution texture (as in a similar texel density on screen), but with low resolution textures, which saves memory:



A layered shader allows you to share textures between assets, and to combine tileable materials to add variation. This makes it easy for you to populate a large world at low cost:



Layered shaders for material layering

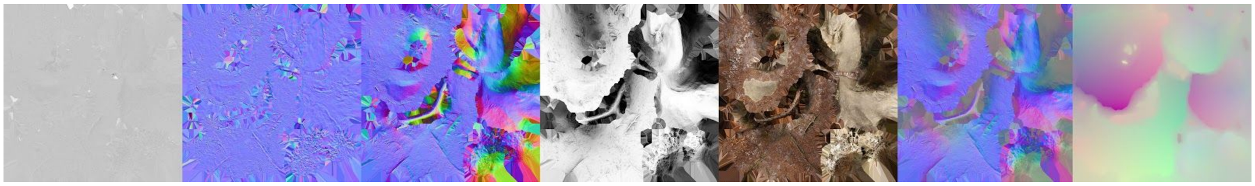
You can also combine materials using the HDRenderPipeline layered shader provided with the Scriptable Render Pipeline package. You can download this from Github here. This layered shader has dedicated options that target photogrammetry assets.

2. Using layered shaders for photogrammetry

This section shows you how to use the layered shader provided with the HDRenderPipeline package.

2.1. Preparing textures for main materials

To prepare textures for an object, follow the photogrammetry process to produce a set of textures for the main material (see Unity Photogrammetry Workflow - 5.4 Textures baking):



Somes textures are used solely for the de-lighting process (see the Unity De-Lighting Tool document).
Baking tools produce the diffuse, normal map, height map and ambient occlusion map textures for the main material. You need to author the other textures in the main material, such as the metalness map, detail mask map and smoothness map. You can do this in any photo editor, like Photoshop.

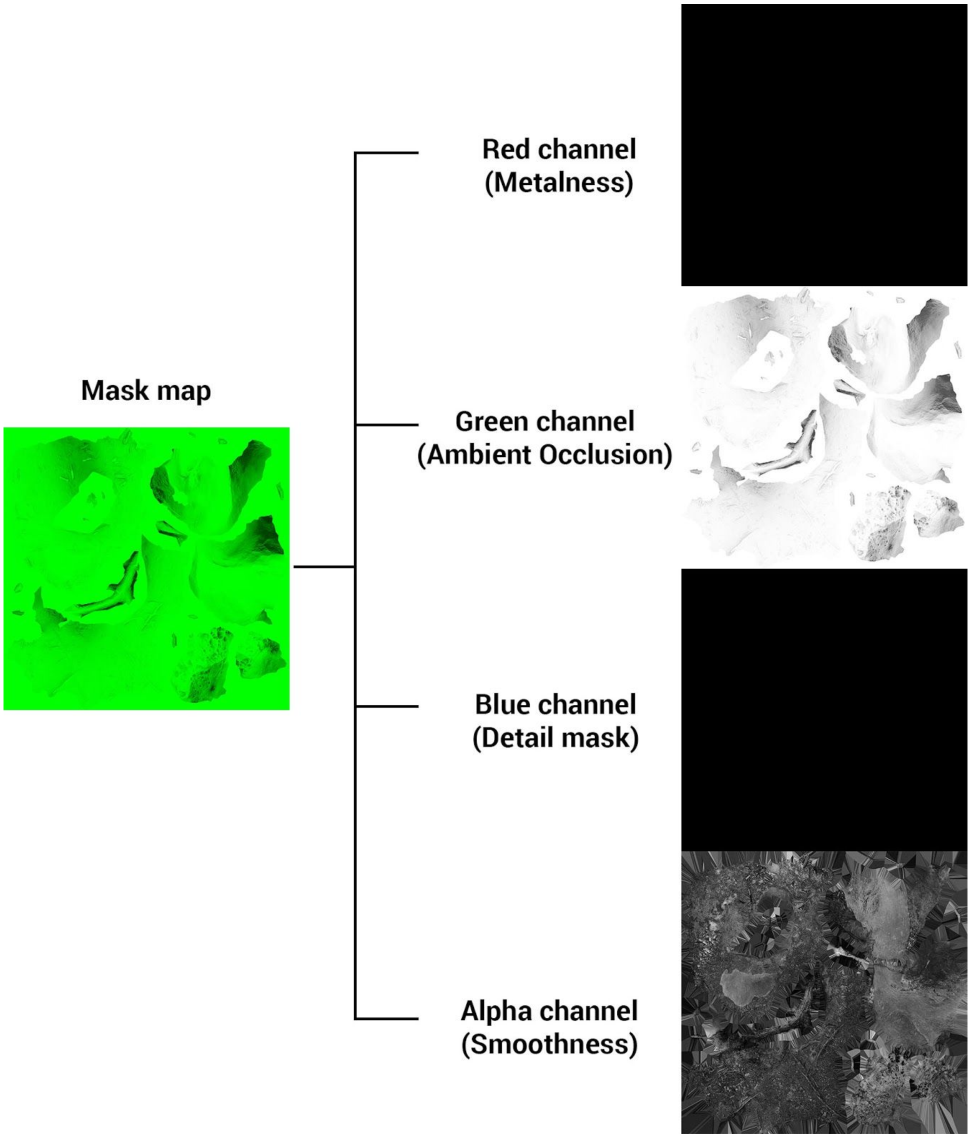
To use the layered shader for photogrammetry, you need to:

- Pack all the textures together using any photo editor.
- De-light the diffuse texture.
- Save the height map information from the baking step.

Here’s a summary of what you need:

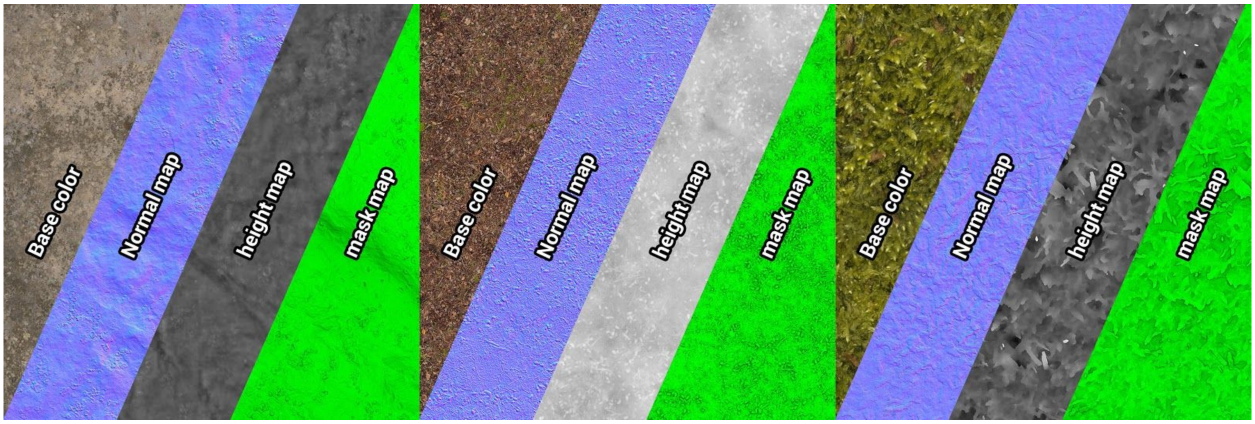
Base color: Create a base color by baking + de-lighting (see the Unity De-Lighting Tool document).
Normal map: Create a normal map by baking in tangent space.
Height map: Create a height map by baking**.** The baking tool provides min, mid and max values for height map sampling. Save or make a note of these values and define them to the layered shader in the Unity Editor.
Mask map: Create a linear composited map in a photo editor. The red channel is for the metalness mask. The green channel is for ambient occlusion. The blue channel is for the detail mask. The alpha channel is for smoothness.

The following diagram shows the channels in the mask map:


Mask map
Red channel (Metalness)
Green channel (Ambient Occlusion)
Blue channel (Detail mask)
Alpha channel (Smoothness)

2.2. Preparing textures for tileable materials

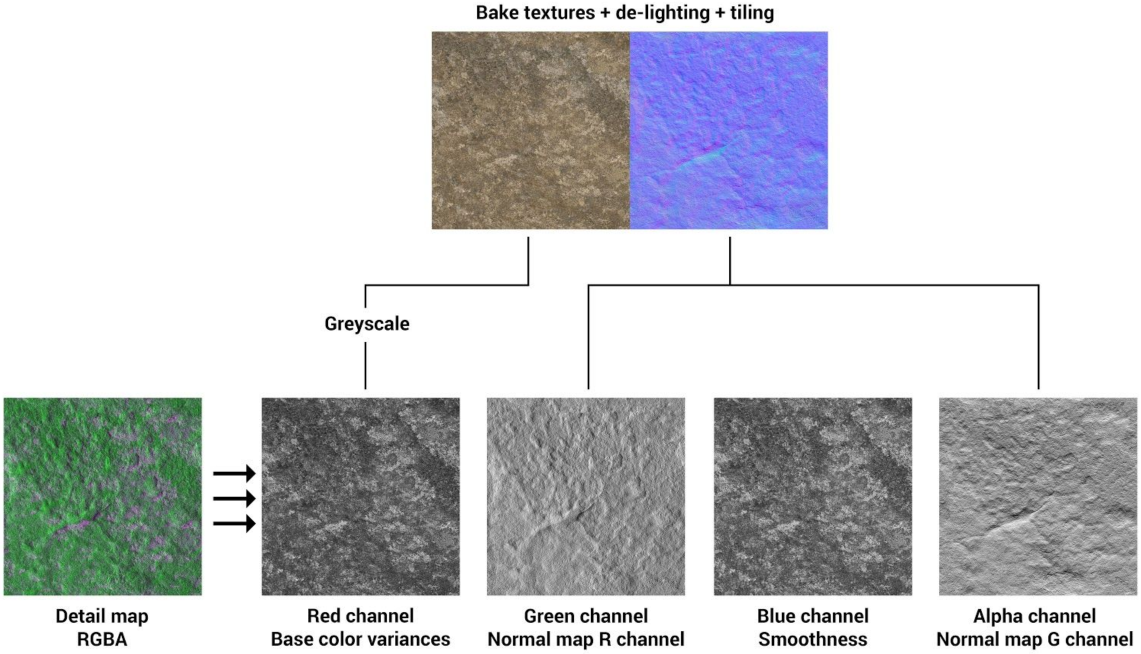
The photogrammetry process captures a set of materials that you can use to bring back details for the layers, as shown in Unity Photogrammetry Workflow - 4.2 How to capture a surface. These materials are tileable, unlike the main material. In the rock example, there's the set of materials for the stone, ground and moss:



Similar to the main material, you need to pack the textures for the layered shader with the same expectation list, except that the material needs to be tileable (see Unity Photogrammetry Workflow - 5.6. Tileable material - Artomatix - Substance painter).

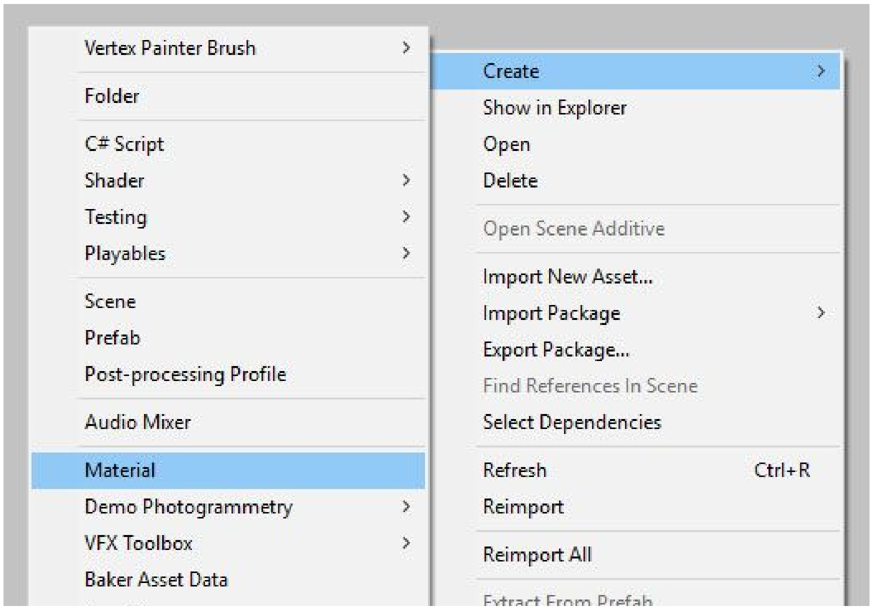
Adding detail textures to tileable materials

To give the illusion of high resolution textures, you can add detail textures to the tileable materials. A detail texture is a small repeatable pattern that provides additional micro-details. It is an optional feature of the layered photogrammetry shader. To produce detail textures, follow the photogrammetry process, as shown in Unity Photogrammetry Workflow - 4.2.3 Detail texture. Detail textures only support based color, normal map and smoothness. Layered shaders require the following packing:



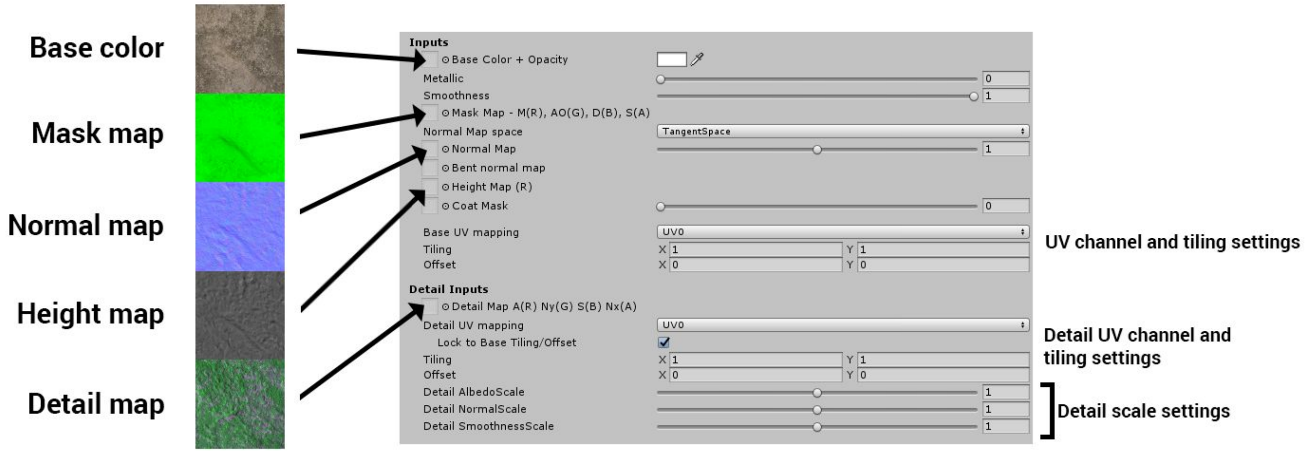
2.3. Setting up materials

To create a HDRenderPipeline lit shader in the Unity Editor, first create a material. Right click on the project explorer and choose **Create** > **Material**:



By default, this creates a lit shader.

Next, drag and drop the textures into the appropriate slots:



Inputs

- Base Color + Opacity
- Metallic: 0
- Smoothness: 1
- Mask Map - M(R), AO(G), D(B), S(A)
- Normal Map space: TangentSpace
- Normal Map: 1
- Bent normal map
- Height Map (R)
- Coat Mask: 0
- Base UV mapping: UV0
- Tiling: X 1, Y 1
- Offset: X 0, Y 0

Detail Inputs

- Detail Map A(R) Ny(G) S(B) Nx(A)
- Detail UV mapping: UV0
- Lock to Base Tiling/Offset: ☑
- Tiling: X 1, Y 1
- Offset: X 0, Y 0
- Detail AlbedoScale: 1
- Detail NormalScale: 1
- Detail SmoothnessScale: 1

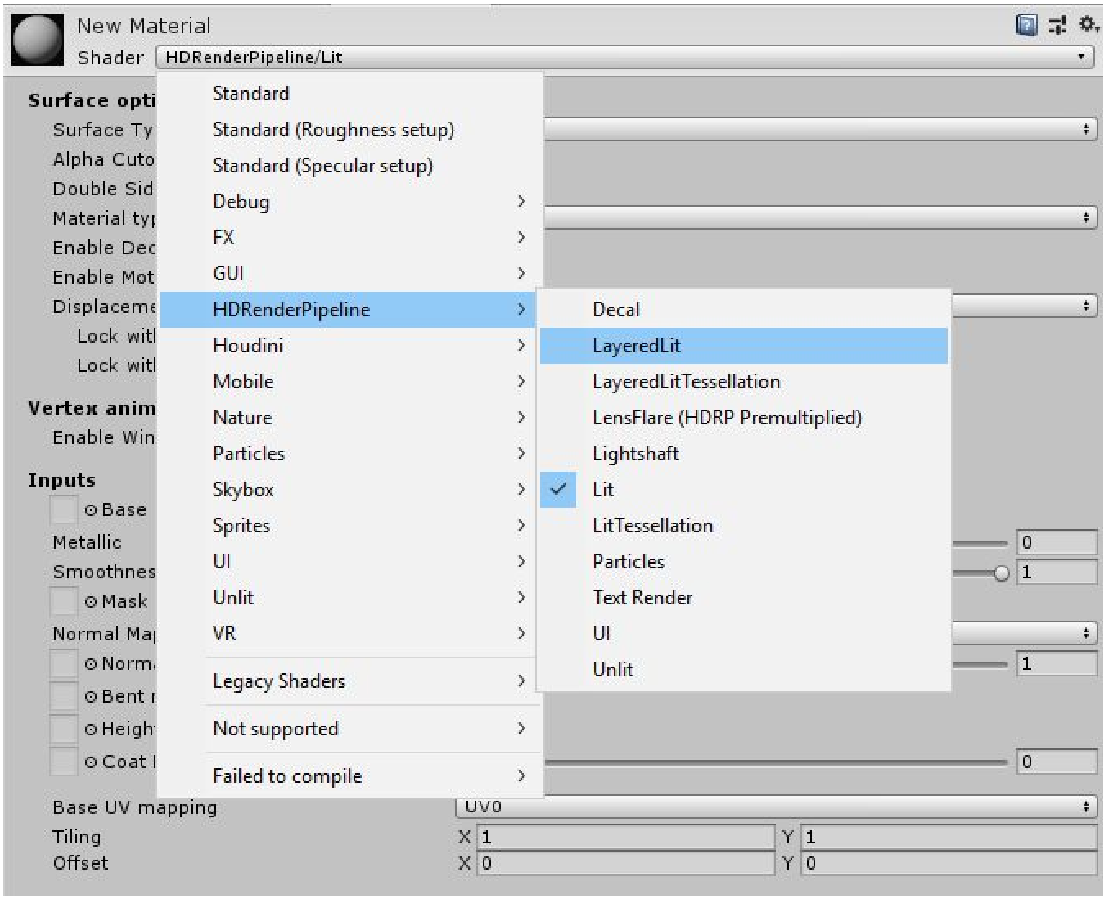
The height map is used in case of tessellation or per pixel displacement. In the rock example in this document, one main material and 3 tileable materials are created (for the stone, ground element and moss).

2.4. Setting up the layered shader

2.4.1. Creating the layered shader

To create a HDRenderPipeline layered shader in the Editor, right click on the project explorer and choose **Create** > **Material**, as shown in 2.3 Setting up materials.

Then, on the new material, select the **HDRenderPipeline** > **LayeredLit** shader (or **LayeredLitTessellation** shader to use tessellation):



New Material

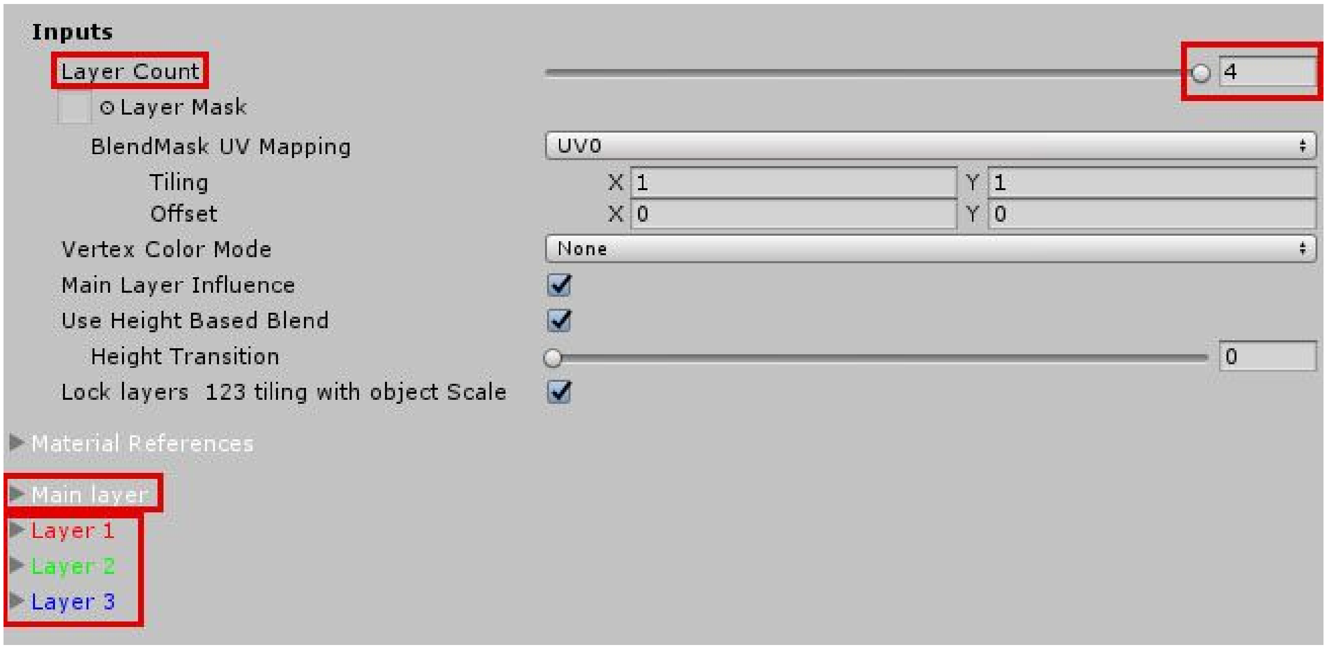
Shader HDRenderPipeline/Lit

Base UV mapping UV0

Tiling X 1 Y 1

Offset X 0 Y 0

The layered shader supports up to 4 layers. The first one is called the **Main Layer**, and the others called **Layer 1**, **Layer 2** and **Layer 3**. Each layer is like a regular material that includes the usual properties of base color, normal map, smoothness and the details textures.



Inputs

Layer Count 4

○ Layer Mask

BlendMask UV Mapping UV0

Tiling X 1 Y 1

Offset X 0 Y 0

Vertex Color Mode None

Main Layer Influence ☑

Use Height Based Blend ☑

Height Transition 0

Lock layers 123 tiling with object Scale ☑

▶ Material References

▶ Main layer

▶ Layer 1

▶ Layer 2

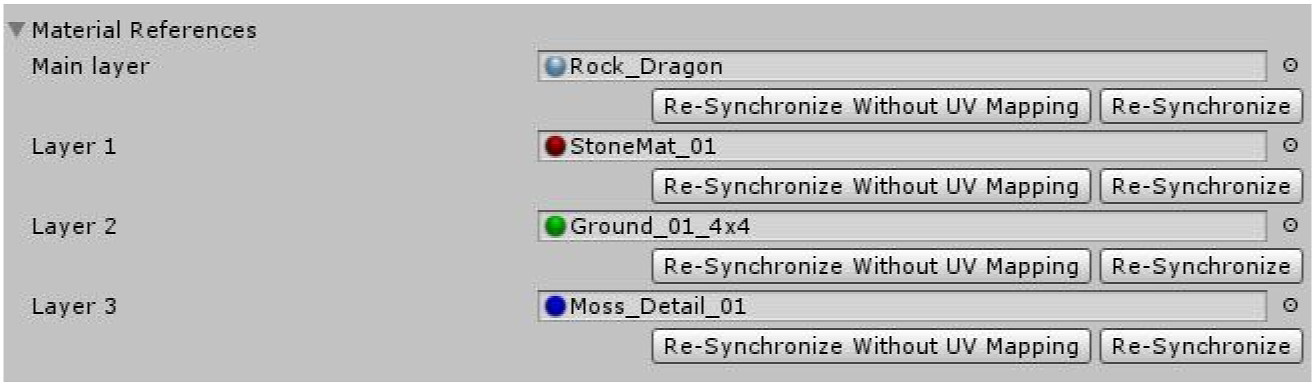
▶ Layer 3

Layers appear in order of importance, with the bottom material first and the top material last. For example, you could have a stone in Layer 1, with moss on top of that stone for Layer 2, and accumulated dust on top of the moss as Layer 3. The dust is less important than the moss because it can be washed away by rain. This ordering allows you to the remove the last layer when producing LOD materials. In the rock example used in this document, the moss is set to Layer 3 so you can remove it from LOD materials.

The cost of the shader depends on the number of layers and textures. Having more layers is more expensive.

You can set up the layer properties directly in the layered shader. However, the layered shader allows you to copy properties from a lit shader to a Layer. This enables you to share material properties between various layered and lit shaders to add some variation. This document shows you how to use this method.

Use the **Material References** section to copy the properties of a lit material to the corresponding layer. Once the material is loaded, click the **Re-Synchronize** button to perform the copy.



▼ Material References

Main layer Rock_Dragon

Layer 1 StoneMat_01

Layer 2 Ground_01_4x4

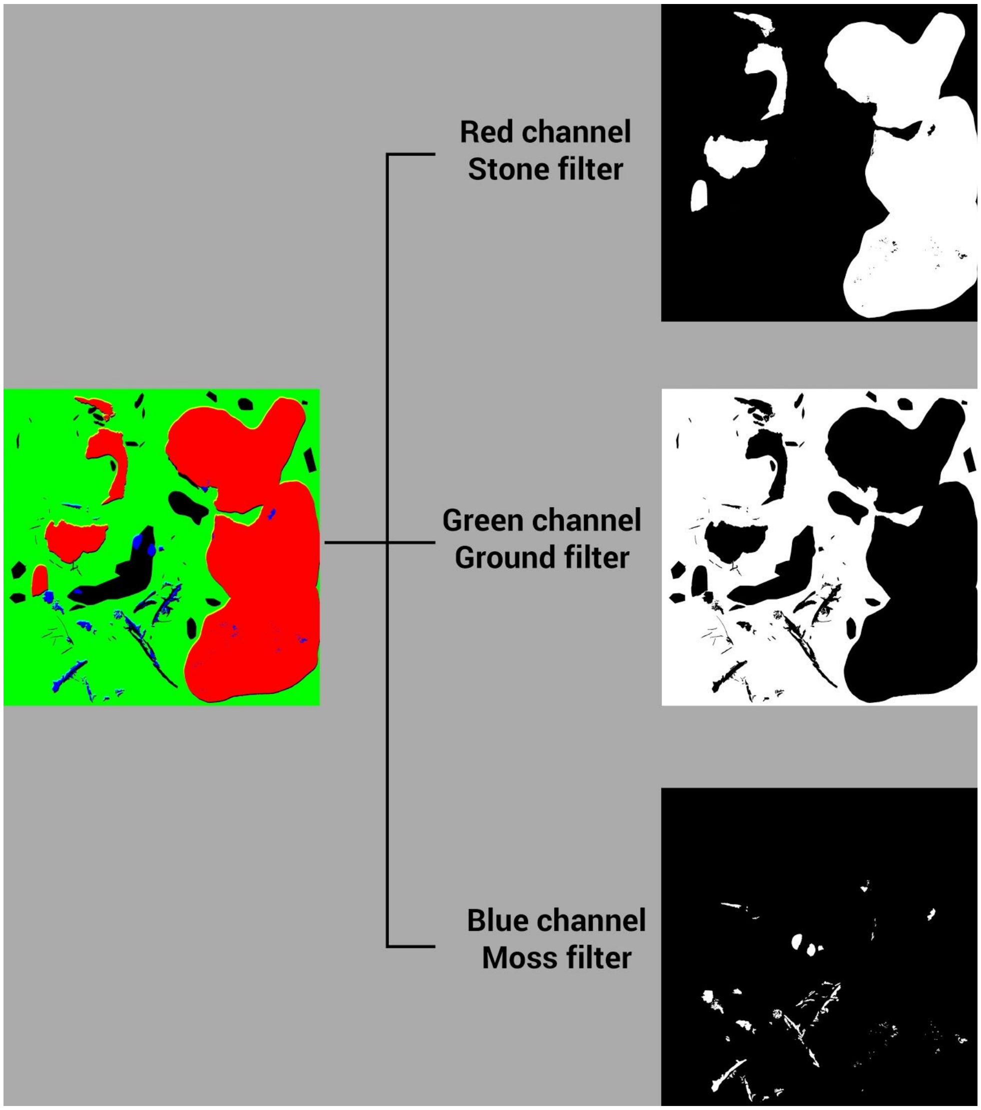
Layer 3 Moss_Detail_01

This setup allows you to update a lit shader material and replicate the change on various layered materials.

The various layers are mixed depending on the weights set by an artist. There are a lot of weighting options that combine with each other to define these weights: vertex color, layer mask and height map mask. The example in this document uses a layer mask.

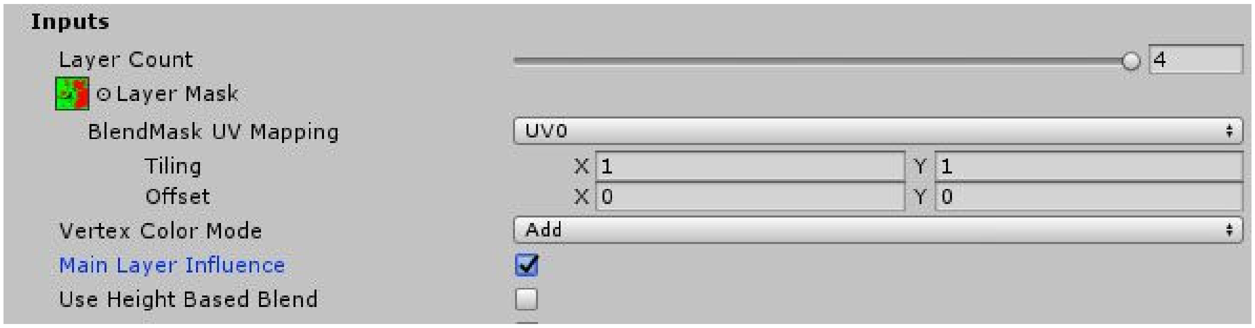
2.4.2. Setting up the layer mask

You use a layer mask to drive layer visibility. You must import the texture as a linear texture with sRGB unchecked. Each channel manages the visibility of a layer:



2.4.3. Setting up influence mode

As discussed at the beginning of this document, the object in our example consists of a main material and tileable materials on top of the main material. The **Main Layer Influence** option enables the main material to influence the other layers (layers 1, 2, 3). That way, the main material can affect the base color, normal and height of the other layers. Enabling **Main Layer Influence** cuts down on the repetitiveness of tiling materials, and maintains the appearance of the object. You can use a mask to filter the influence effect:



Inputs

Layer Count 4

○ Layer Mask

BlendMask UV Mapping UV0

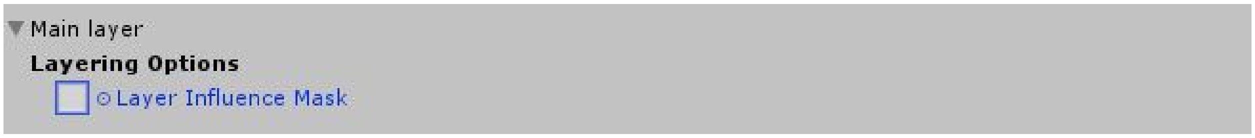
Tiling X 1 Y 1

Offset X 0 Y 0

Vertex Color Mode Add

Main Layer Influence ☑

Use Height Based Blend □

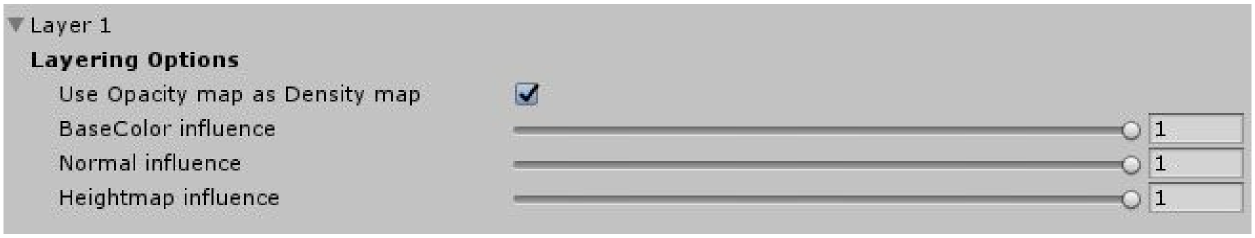


▼ Main layer

Layering Options

□ ○ Layer Influence Mask

You can control influence strength individually for each component and for each layer:



▼ Layer 1

Layering Options

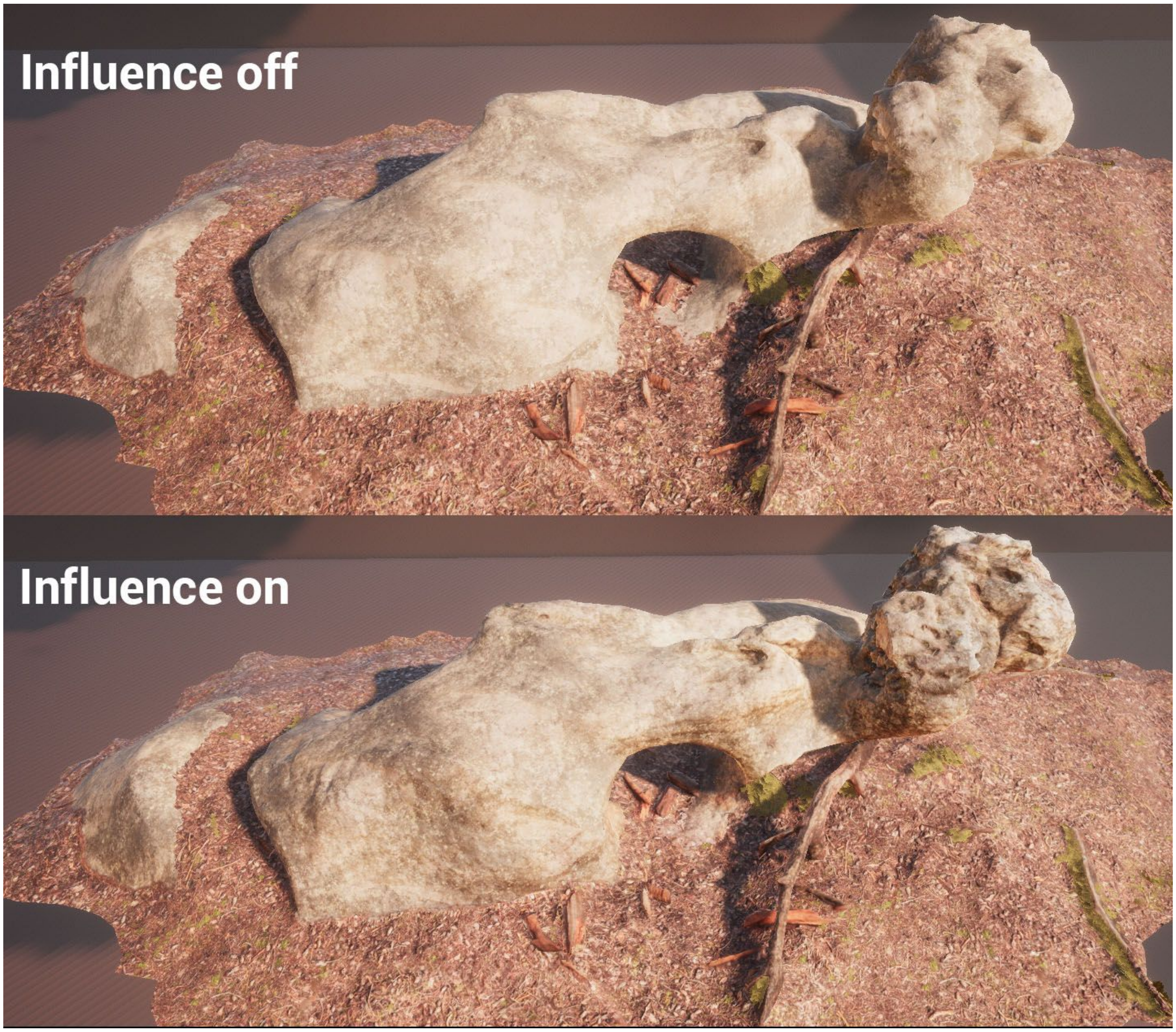
Use Opacity map as Density map ☑

BaseColor influence 1

Normal influence 1

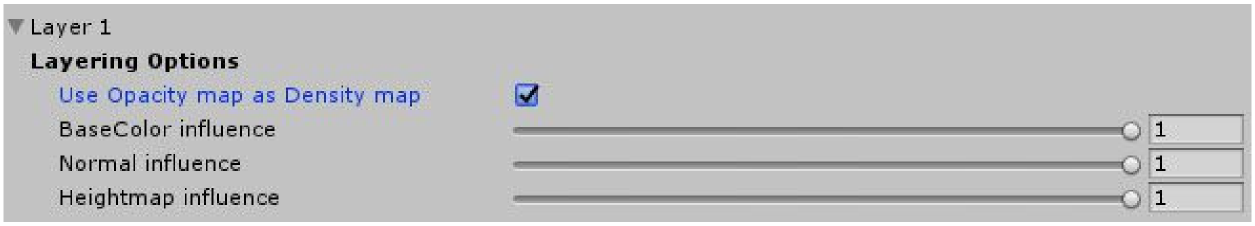
Heightmap influence 1

The following images show the visual difference when using the **Main Layer Influence** on the stone Layer:



2.4.4. Density mode

You can set the alpha channel of the base color texture to use either the opacity of the texture or the density of the texture. You do this by selecting or deselecting **Use Opacity map as Density map** for each layer:



▼ Layer 1

Layering Options

Use Opacity map as Density map ☑

BaseColor influence 1

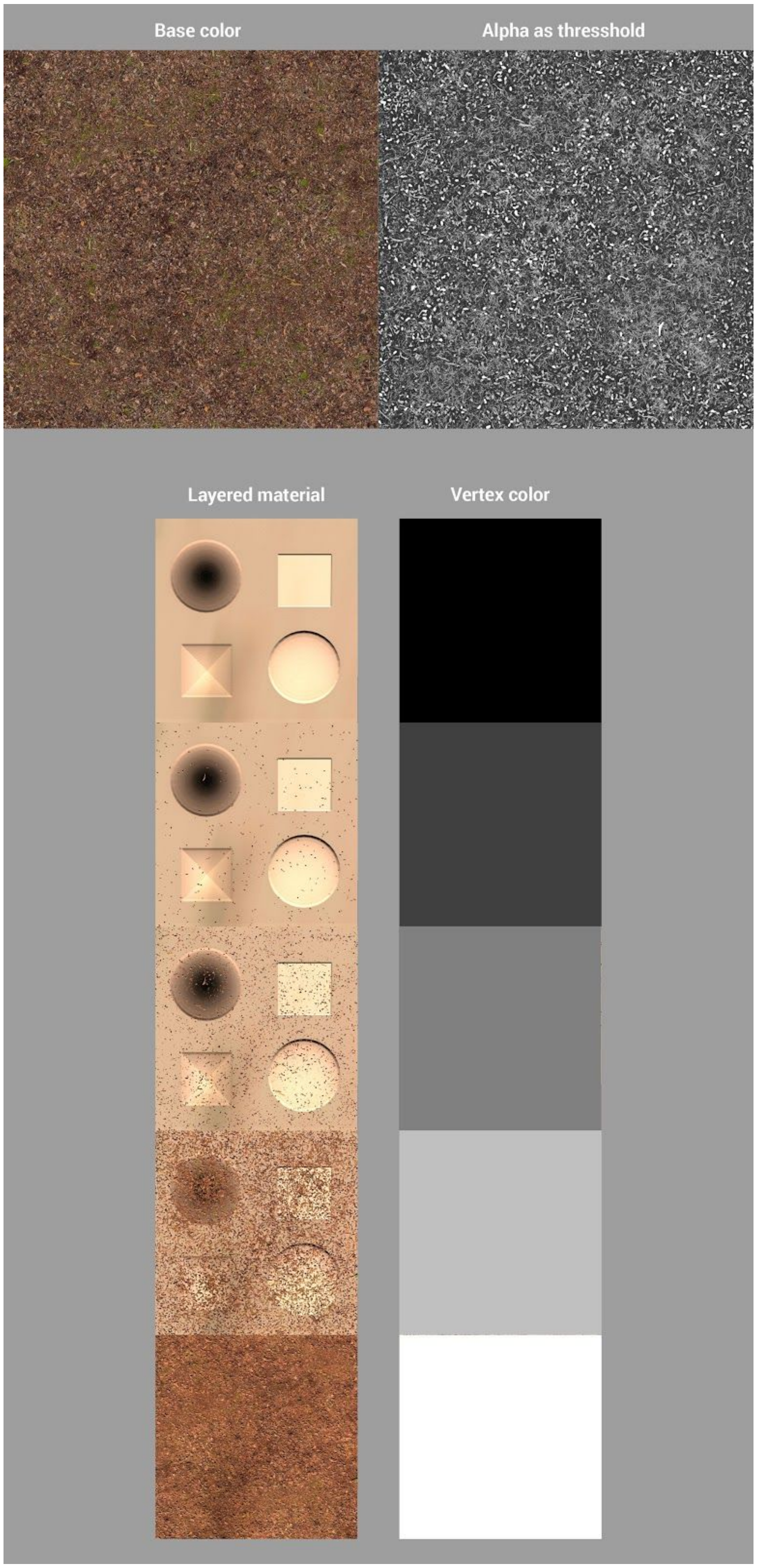
Normal influence 1

Heightmap influence 1

When you enable **Use opacity map as Density map**, the base color texture uses the alpha channel.

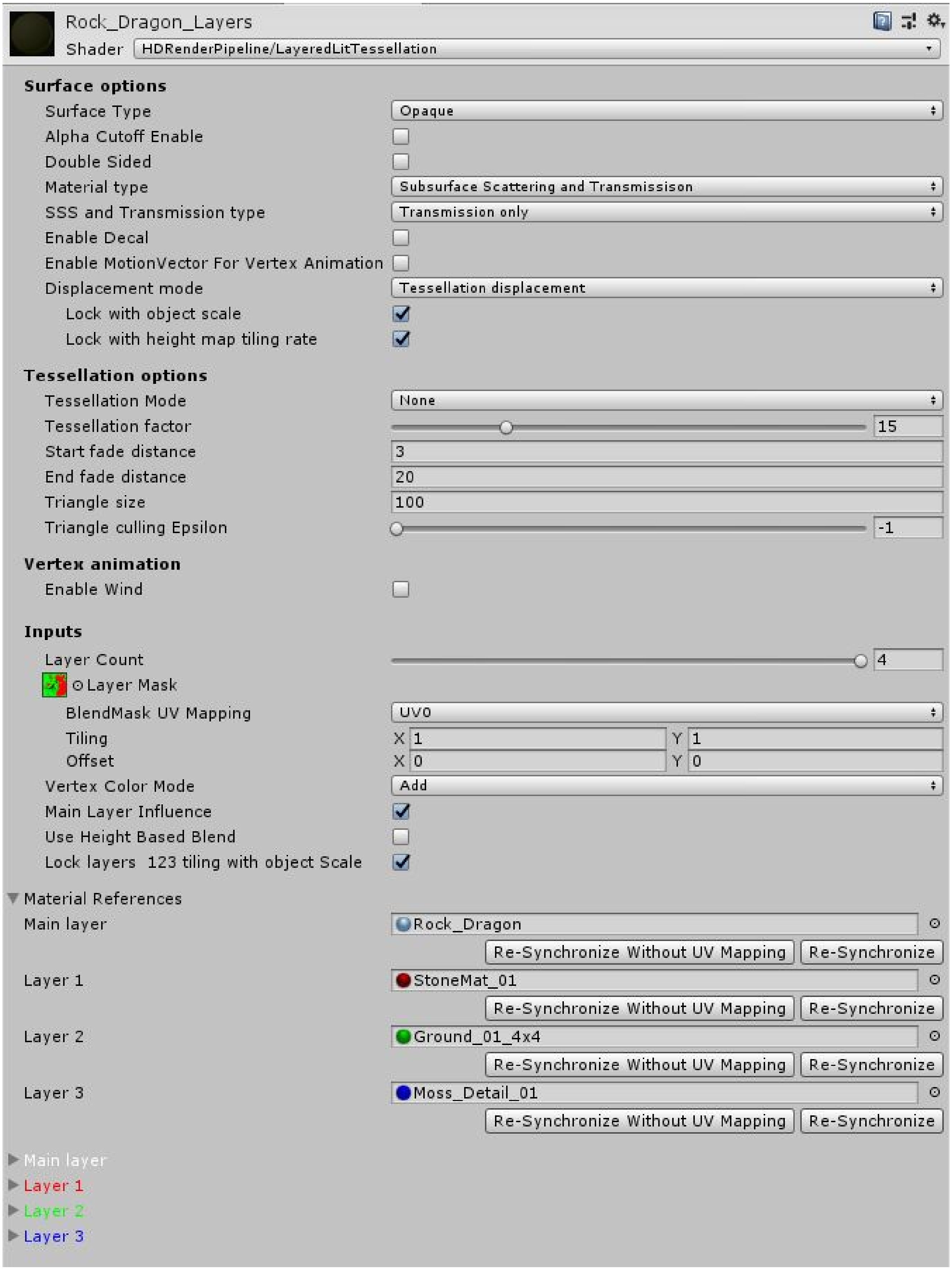
This density map defines the density of the visible elements for a layer. The vertex color controls the threshold of the density map. The density map is grayscale and hand authored. White elements are ordered by visibles first and black elements last. The layer is more visible when the vertex color of the channel (red: layer1, green: layer 2, blue: layer 3) is closer to 1.

Note: The threshold is the inverse of the vertex color. All values inside the density map above the threshold make the layer visible. This is an additive process, which is why there is an inversion here.



2.4.5. Rock dragon layered material example

The following screenshot shows the layered shader setup to render the rock dragon example used in this document. The **Main Layer Influence** option is set on the layered shader to make the main layer influence the stone layer:



Rock_Dragon_Layers
Shader HDRenderPipeline/LayeredLitTessellation

Surface options
Surface Type: Opaque
Alpha Cutoff Enable ☐
Double Sided ☐
Material type: Subsurface Scattering and Transmissison
SSS and Transmission type: Transmission only
Enable Decal ☐
Enable MotionVector For Vertex Animation ☐
Displacement mode: Tessellation displacement
Lock with object scale ☑
Lock with height map tiling rate ☑

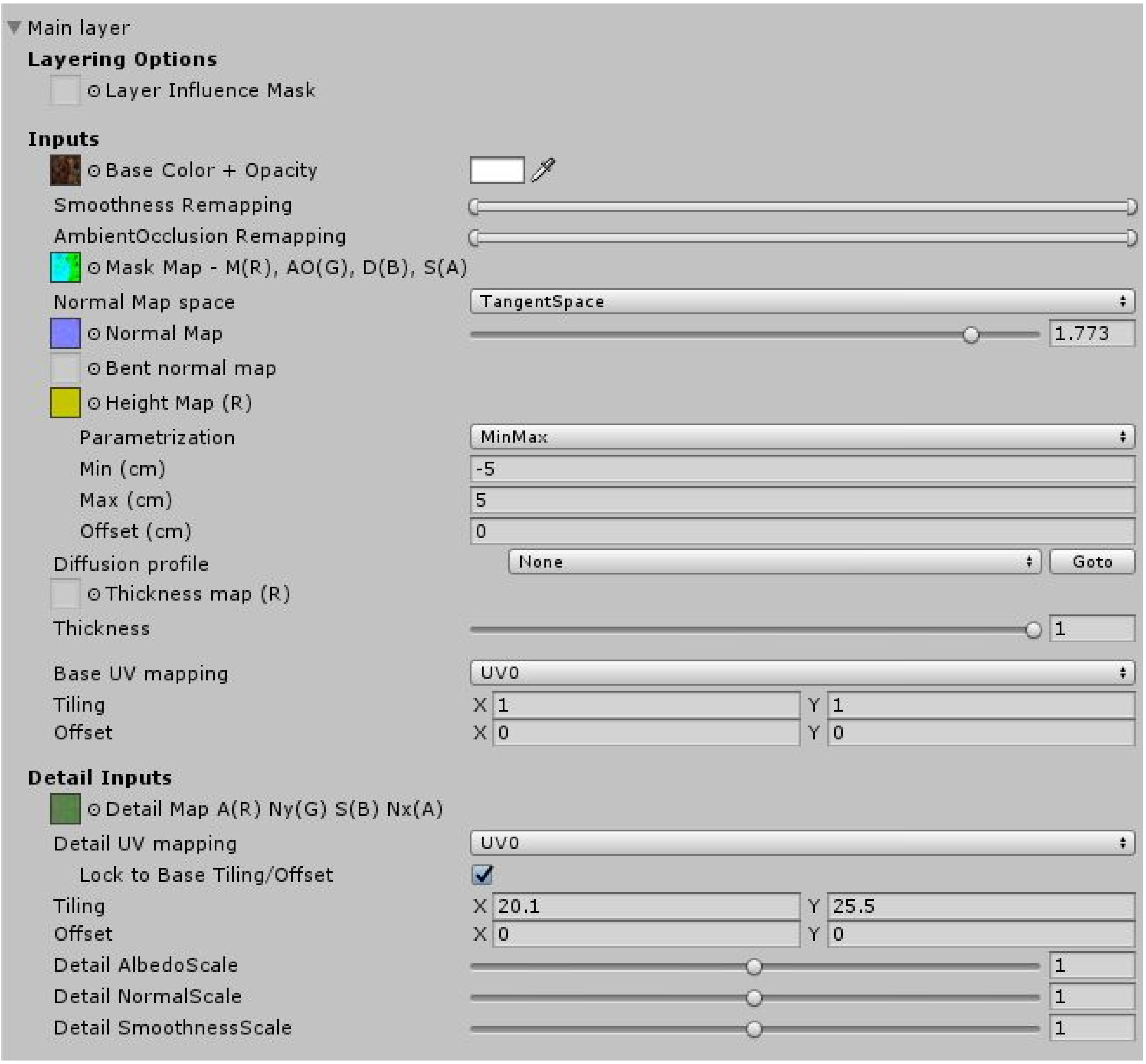
Tessellation options
Tessellation Mode: None
Tessellation factor: 15
Start fade distance: 3
End fade distance: 20
Triangle size: 100
Triangle culling Epsilon: -1

Vertex animation
Enable Wind ☐

Inputs
Layer Count: 4
Layer Mask
BlendMask UV Mapping: UV0
Tiling X 1 Y 1
Offset X 0 Y 0
Vertex Color Mode: Add
Main Layer Influence ☑
Use Height Based Blend ☐
Lock layers 123 tiling with object Scale ☑

Material References
Main layer: Rock_Dragon
Layer 1: StoneMat_01
Layer 2: Ground_01_4x4
Layer 3: Moss_Detail_01

Main layer settings in the dragon rock example:



Main layer

Layering Options

Layer Influence Mask

Inputs

Base Color + Opacity

Smoothness Remapping

AmbientOcclusion Remapping

Mask Map - M(R), AO(G), D(B), S(A)

Normal Map space: TangentSpace

Normal Map: 1.773

Bent normal map

Height Map (R)

Parametrization: MinMax

Min (cm): -5

Max (cm): 5

Offset (cm): 0

Diffusion profile: None

Thickness map (R)

Thickness: 1

Base UV mapping: UV0

Tiling: X 1 Y 1

Offset: X 0 Y 0

Detail Inputs

Detail Map A(R) Ny(G) S(B) Nx(A)

Detail UV mapping: UV0

Lock to Base Tiling/Offset: ☑

Tiling: X 20.1 Y 25.5

Offset: X 0 Y 0

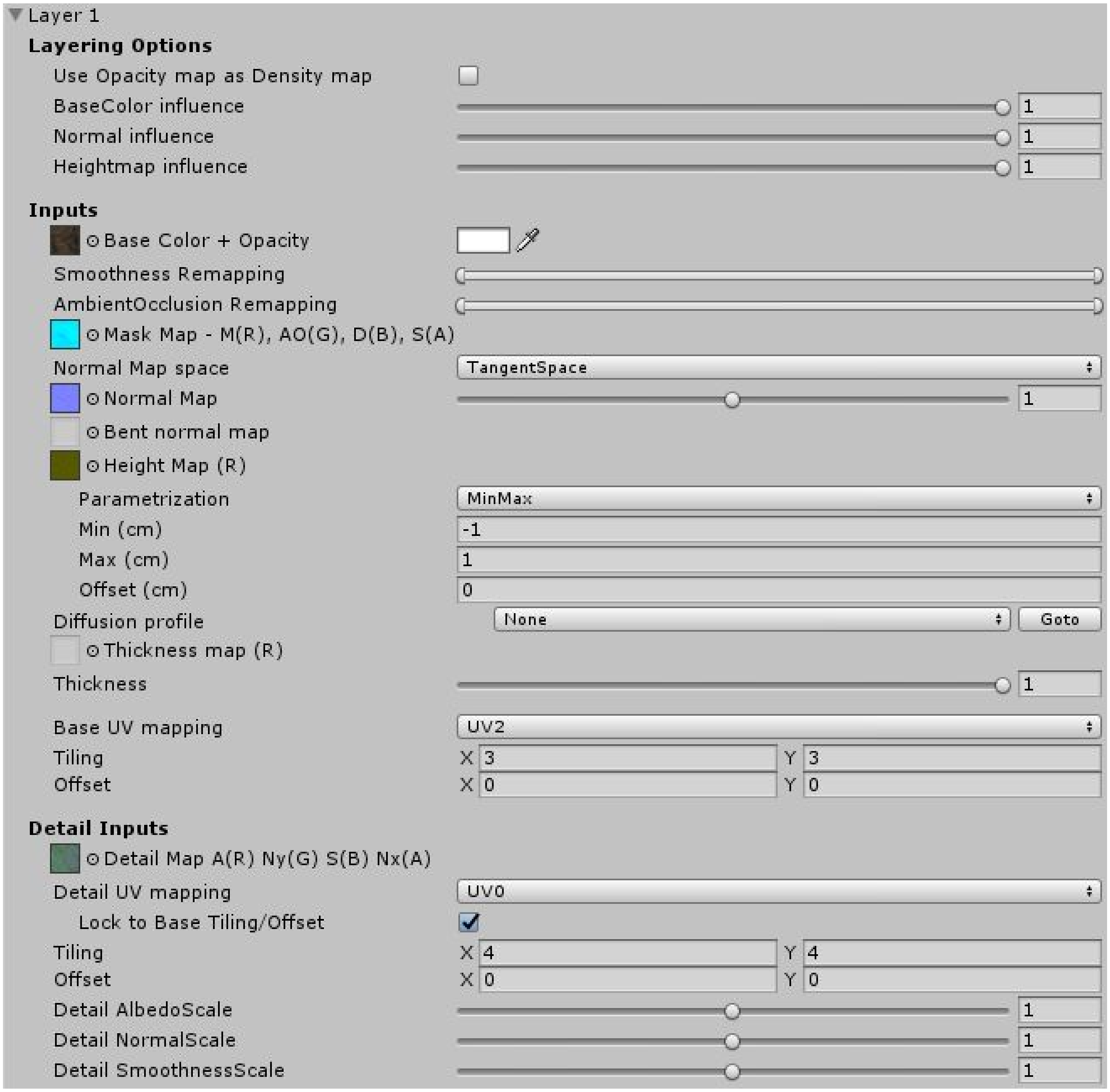
Detail AlbedoScale: 1

Detail NormalScale: 1

Detail SmoothnessScale: 1

The texture when only the main layer is showing:

The following screen shots show the settings for each of the layers in the shader. The **Use Opacity map as Density map** option is enabled for the ground element and moss layers. The **Layer 1** settings for the stone layer are:



Layer 1

Layering Options

Use Opacity map as Density map □
BaseColor influence 1
Normal influence 1
Heightmap influence 1

Inputs

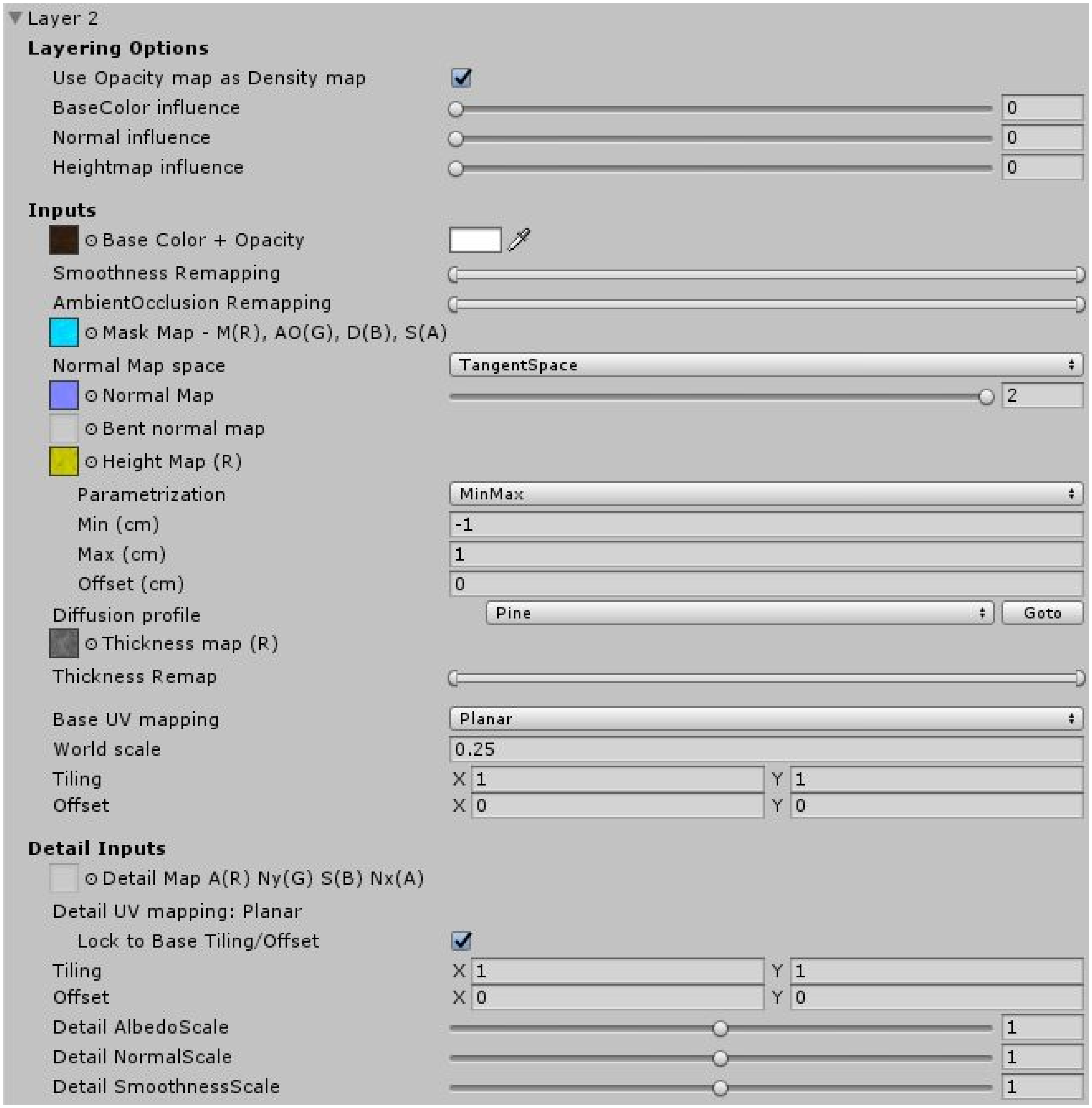
Base Color + Opacity
Smoothness Remapping
AmbientOcclusion Remapping
Mask Map - M(R), AO(G), D(B), S(A)
Normal Map space TangentSpace
Normal Map 1
Bent normal map
Height Map (R)
Parametrization MinMax
Min (cm) -1
Max (cm) 1
Offset (cm) 0
Diffusion profile None Goto
Thickness map (R)
Thickness 1
Base UV mapping UV2
Tiling X 3 Y 3
Offset X 0 Y 0

Detail Inputs

Detail Map A(R) Ny(G) S(B) Nx(A)
Detail UV mapping UV0
Lock to Base Tiling/Offset ☑
Tiling X 4 Y 4
Offset X 0 Y 0
Detail AlbedoScale 1
Detail NormalScale 1
Detail SmoothnessScale 1

The texture showing the rock (main layer) and stone (layer 1):

Layer 2 settings for the ground layer in the dragon rock example:



Layer 2

Layering Options

Use Opacity map as Density map ☑

BaseColor influence 0

Normal influence 0

Heightmap influence 0

Inputs

Base Color + Opacity

Smoothness Remapping

AmbientOcclusion Remapping

Mask Map - M(R), AO(G), D(B), S(A)

Normal Map space TangentSpace

Normal Map 2

Bent normal map

Height Map (R)

Parametrization MinMax

Min (cm) -1

Max (cm) 1

Offset (cm) 0

Diffusion profile Pine Goto

Thickness map (R)

Thickness Remap

Base UV mapping Planar

World scale 0.25

Tiling X 1 Y 1

Offset X 0 Y 0

Detail Inputs

Detail Map A(R) Ny(G) S(B) Nx(A)

Detail UV mapping: Planar

Lock to Base Tiling/Offset ☑

Tiling X 1 Y 1

Offset X 0 Y 0

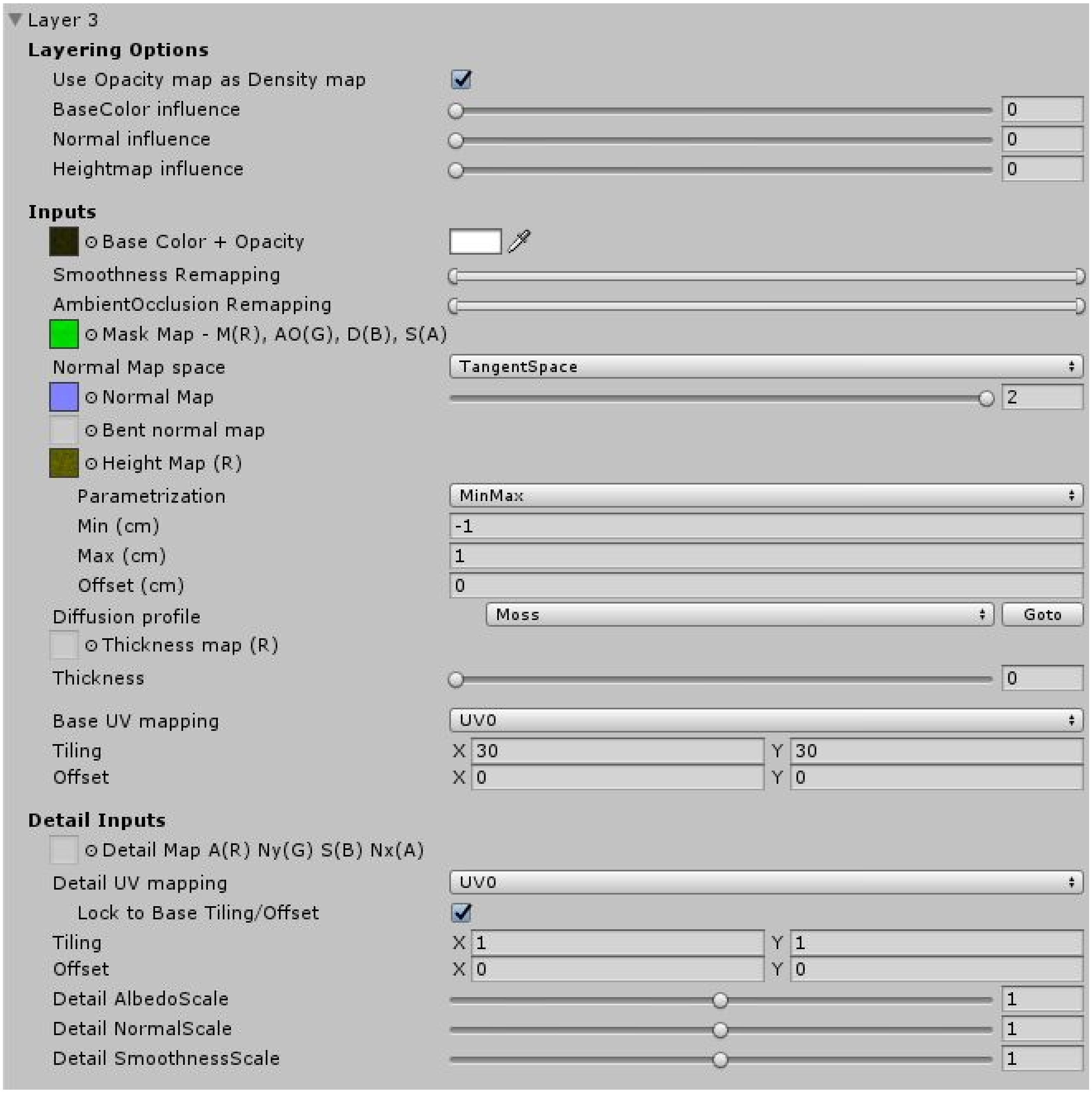
Detail AlbedoScale 1

Detail NormalScale 1

Detail SmoothnessScale 1

The texture showing the rock (main layer), stone (layer 1) and ground elements (layer 2):

Layer 3 settings for the moss layer in the dragon rock example:



▼ Layer 3

Layering Options

Use Opacity map as Density map ☑

BaseColor influence 0

Normal influence 0

Heightmap influence 0

Inputs

⊙ Base Color + Opacity

Smoothness Remapping

AmbientOcclusion Remapping

⊙ Mask Map - M(R), AO(G), D(B), S(A)

Normal Map space TangentSpace

⊙ Normal Map 2

⊙ Bent normal map

⊙ Height Map (R)

Parametrization MinMax

Min (cm) -1

Max (cm) 1

Offset (cm) 0

Diffusion profile Moss Goto

⊙ Thickness map (R)

Thickness 0

Base UV mapping UV0

Tiling X 30 Y 30

Offset X 0 Y 0

Detail Inputs

⊙ Detail Map A(R) Ny(G) S(B) Nx(A)

Detail UV mapping UV0

Lock to Base Tiling/Offset ☑

Tiling X 1 Y 1

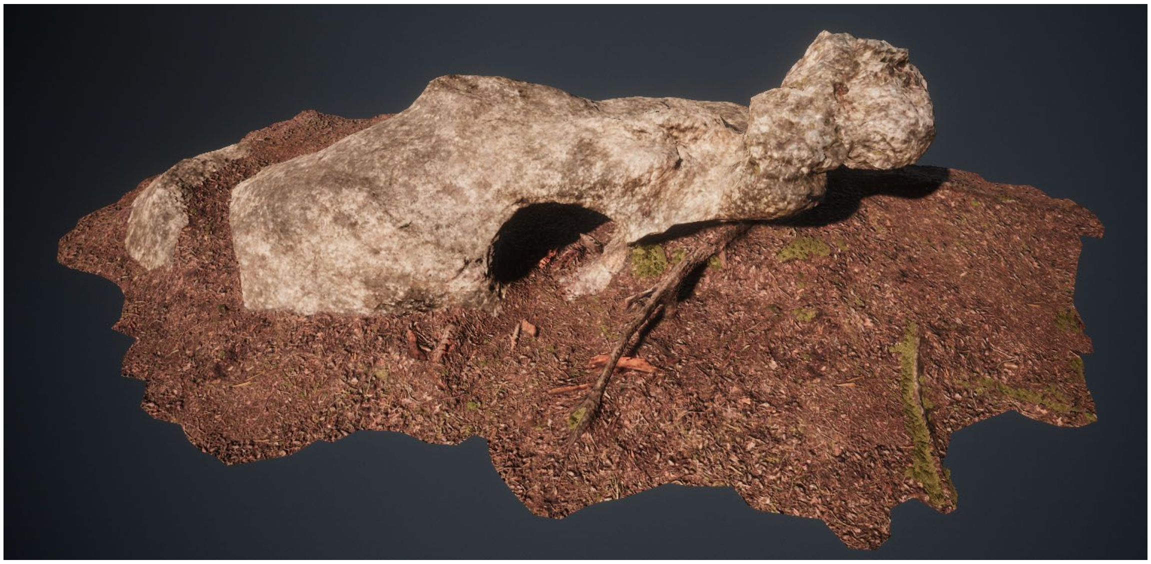
Offset X 0 Y 0

Detail AlbedoScale 1

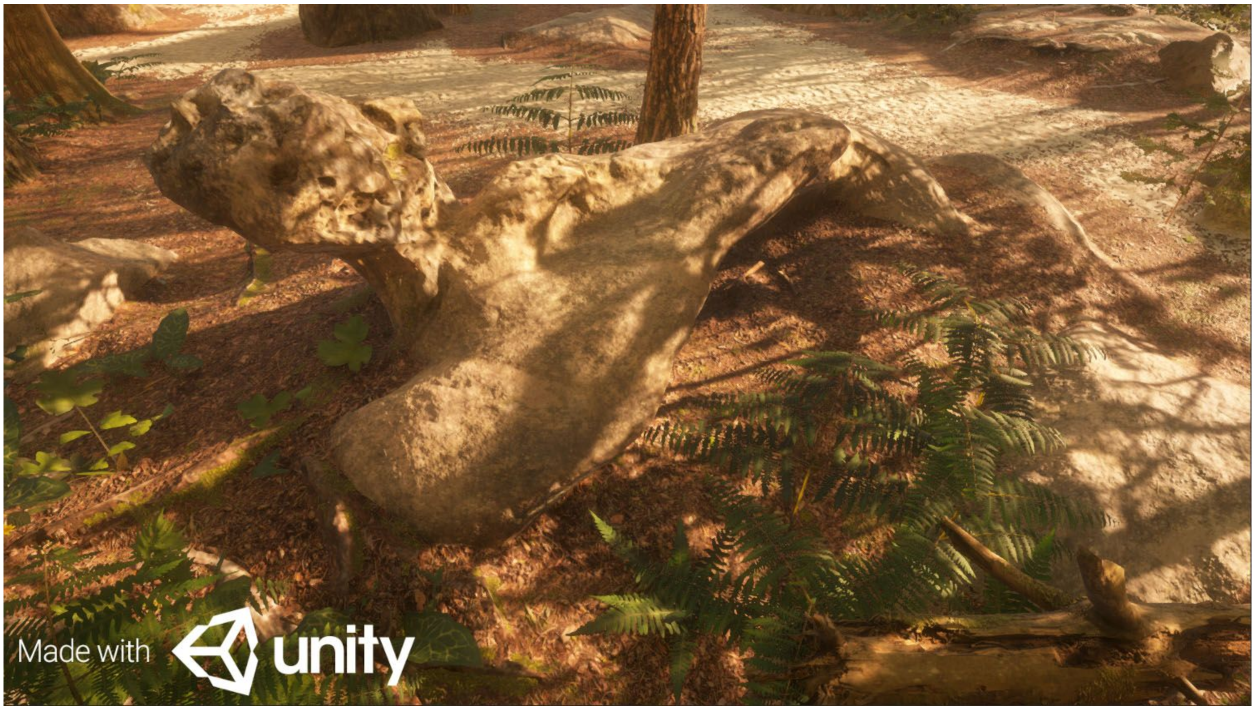
Detail NormalScale 1

Detail SmoothnessScale 1

The texture showing the rock (main layer), stone (layer 1), ground elements (layer 2) and moss (layer 3):

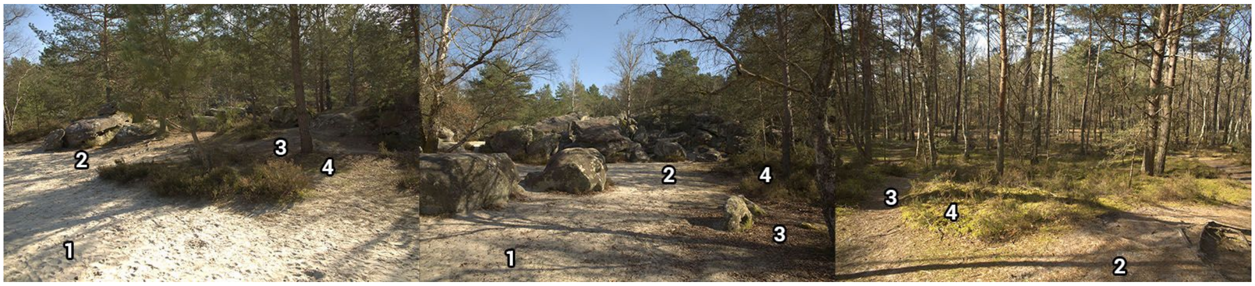


The following screenshot shows an example of the in-game result with lighting:

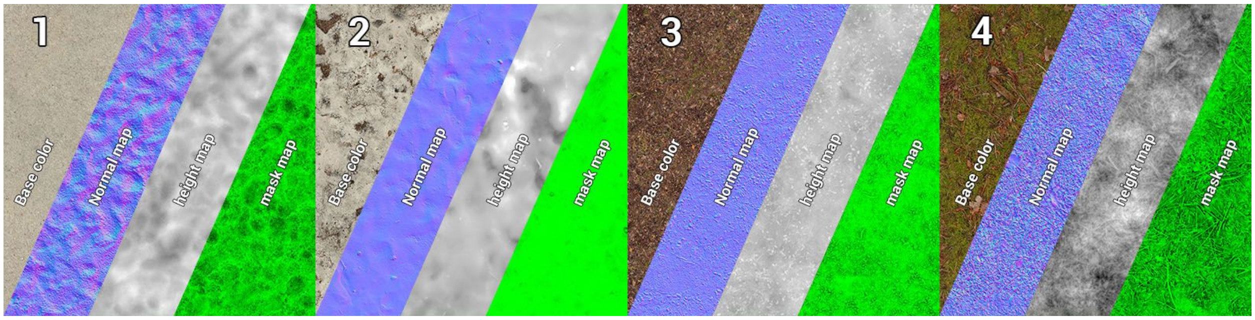


2.5. Photogrammetry object with non influence layer mode

You can use photogrammetry solely for material creation and map it onto a virtual object. Here is an example of a terrain that doesn't use the **Main Layer Influence** option of the layered shader. Instead, it relies on four tileable materials:

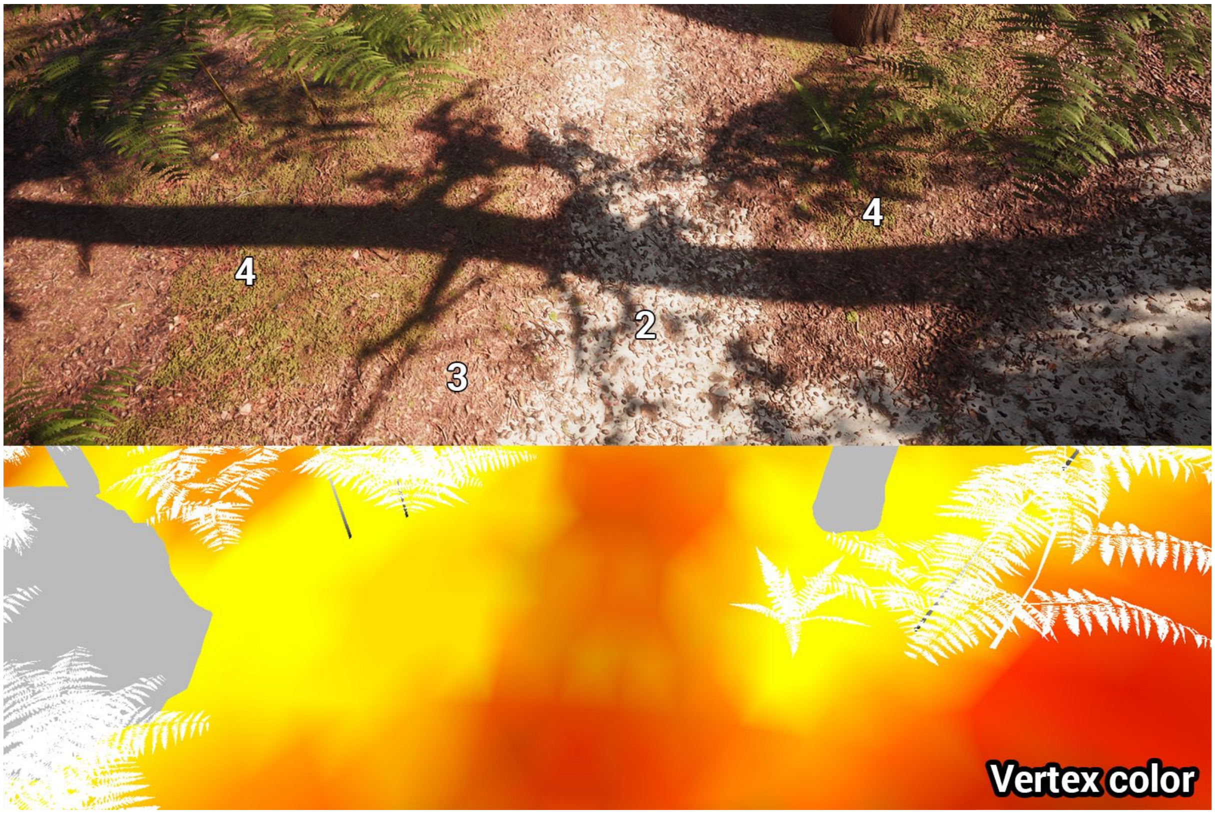


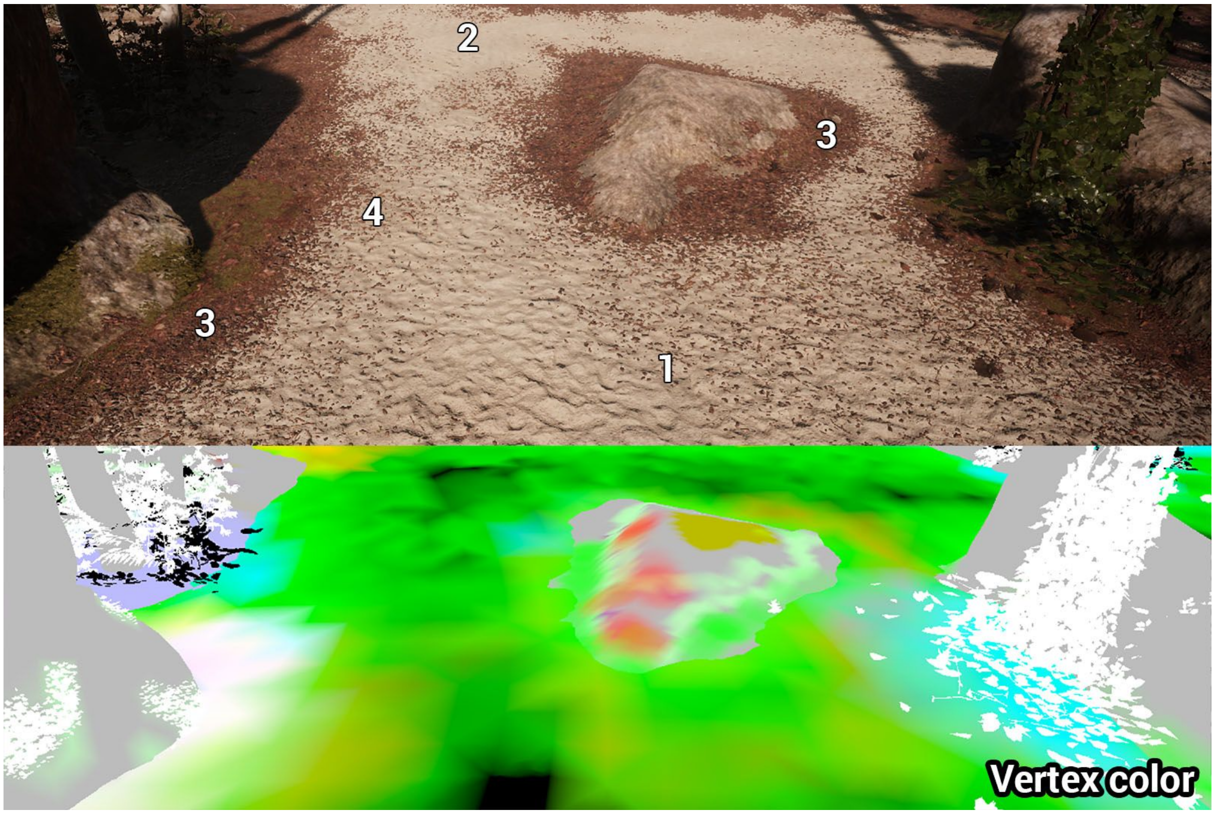
You can identify four main surfaces in each of the photos:

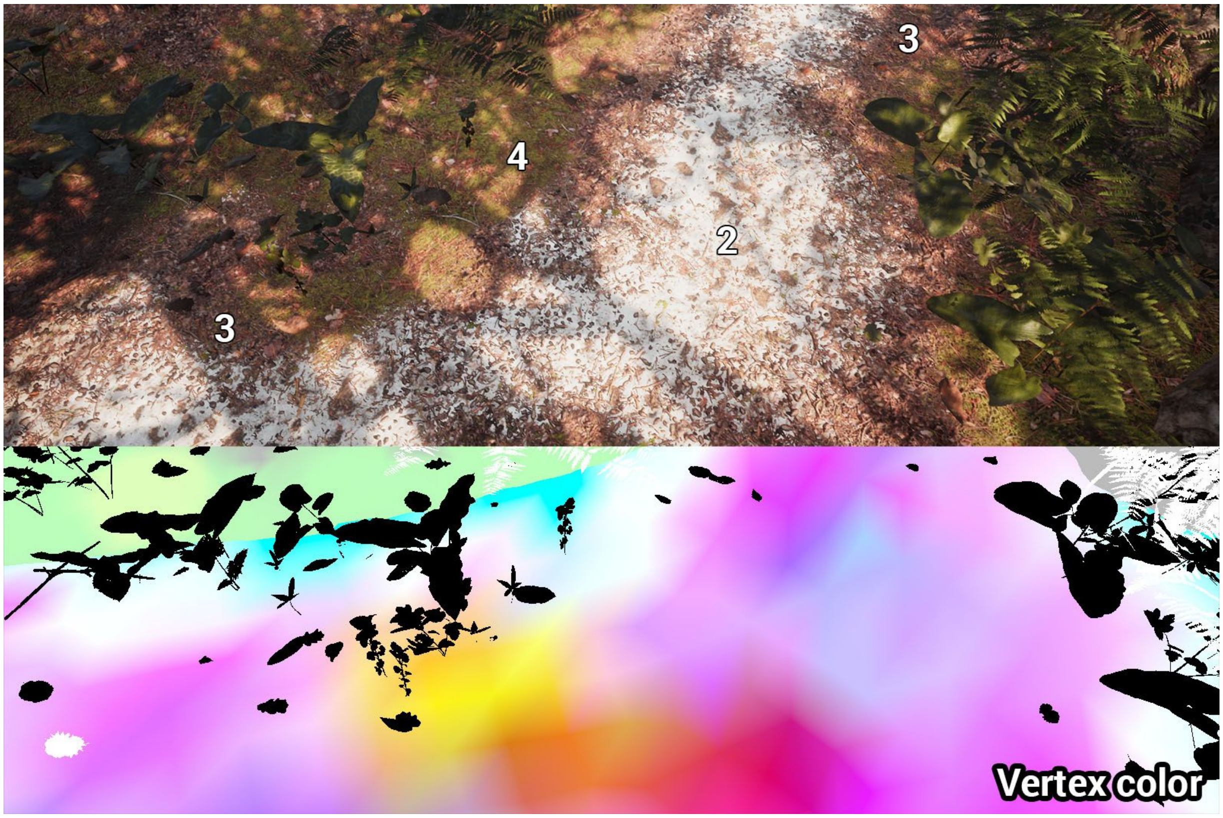


This creates four lit materials.

To set the texture up using the HDRenderPipeline layered shader, you need to create a layered shader with 4 layers and apply it to the ground objects. You can then paint the ground objects using the vertex color. Vertex colors are different because layers are organised differently.

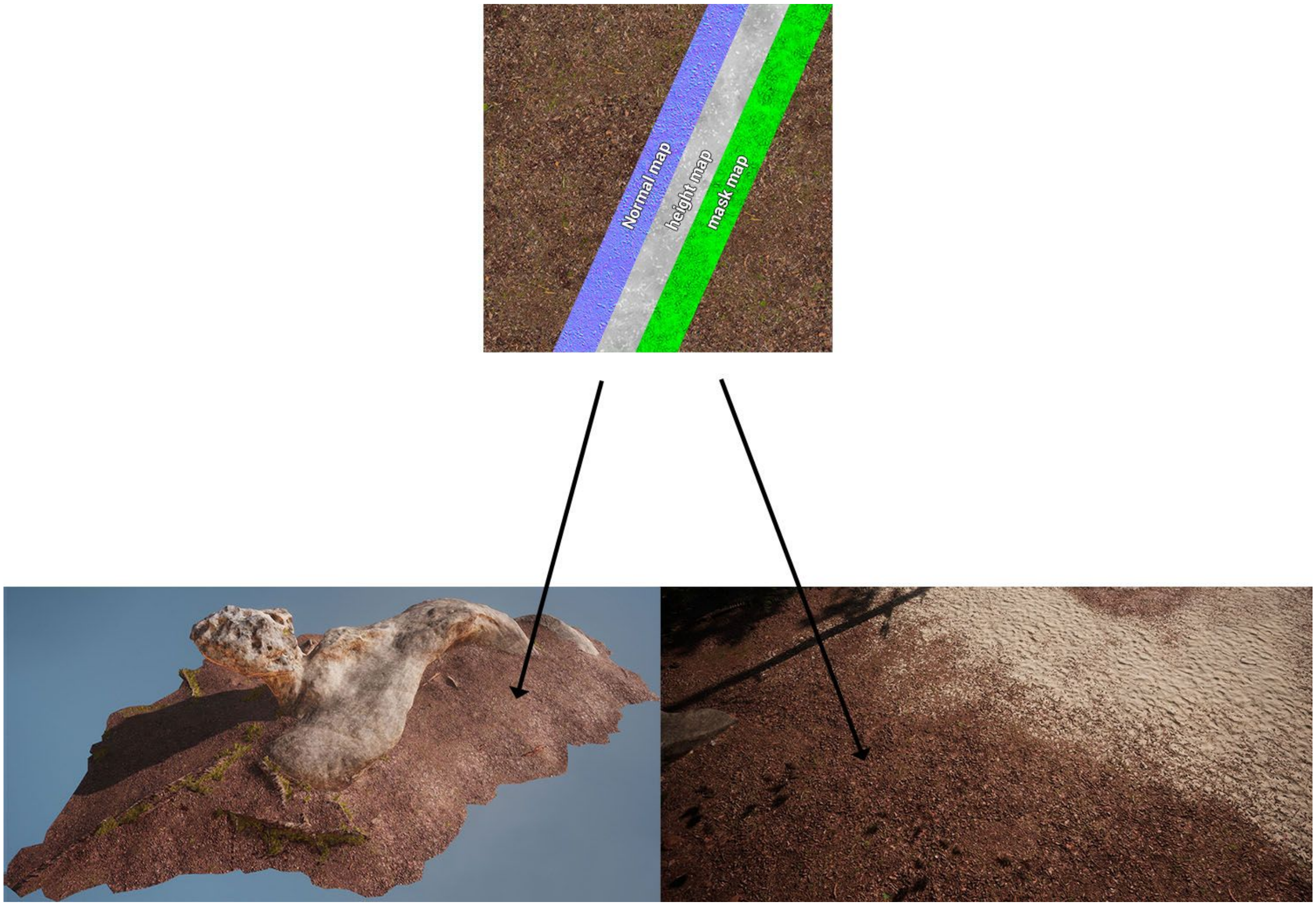




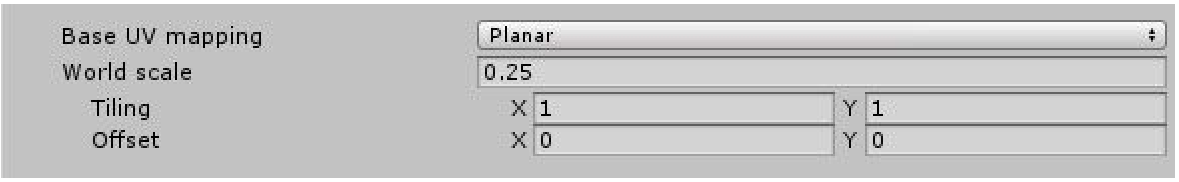

3
4
2
3
Vertex color

2.6. Connect objects to the ground.

To connect two seamless objects, they have to use a common material:



To do this, set the **Base UV mapping** to **Planar** for the common material, and use the same **World scale** and **Tiling** for the two objects:



Base UV mapping	Planar	
World scale	0.25	
Tiling	X 1	Y 1
Offset	X 0	Y 0

For a perfect connection, adjust the Mesh normals in 3D graphics software, such as 3ds Max or Maya.