

# Heightmapped Terrain Textures – Terrain Shader

Thank you for choosing this package!

Here are some hints on using its content, so you can start right through.

## Content:

- The Terrain Shader
- Applying The Shader
- How To Create Heightmapped Textures
- Modifying The Shader
  
- The Mesh Shader

[laxer3d@gmx.de](mailto:laxer3d@gmx.de)

## The Terrain Shader

The standard unity terrain shader simply mixes the colours of the terrain textures.

This shader uses height information for each pixel, so only the pixels that are on top of other pixels are displayed. For example, the grass only appears in gaps between the stones while the built in shader creates grassy stones.

Height-mapped Textures



Default Terrain Shader



## The Terrain Shader

You can edit your terrain just like you would do with the built in shader. If you want to convert an existing terrain just replace the terrain material (see next page).

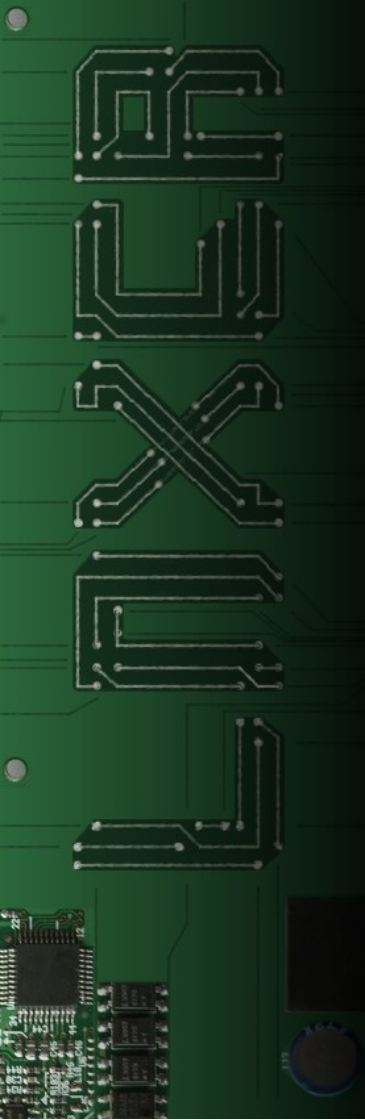
For performance reasons every 4<sup>th</sup> texture creates a new layer.  
Eg. the 5<sup>th</sup> texture of your terrain is drawn above the other textures.

Layer 1: tex0, tex1, tex2, tex3

Layer 2: tex4, tex5, tex6, tex7

...

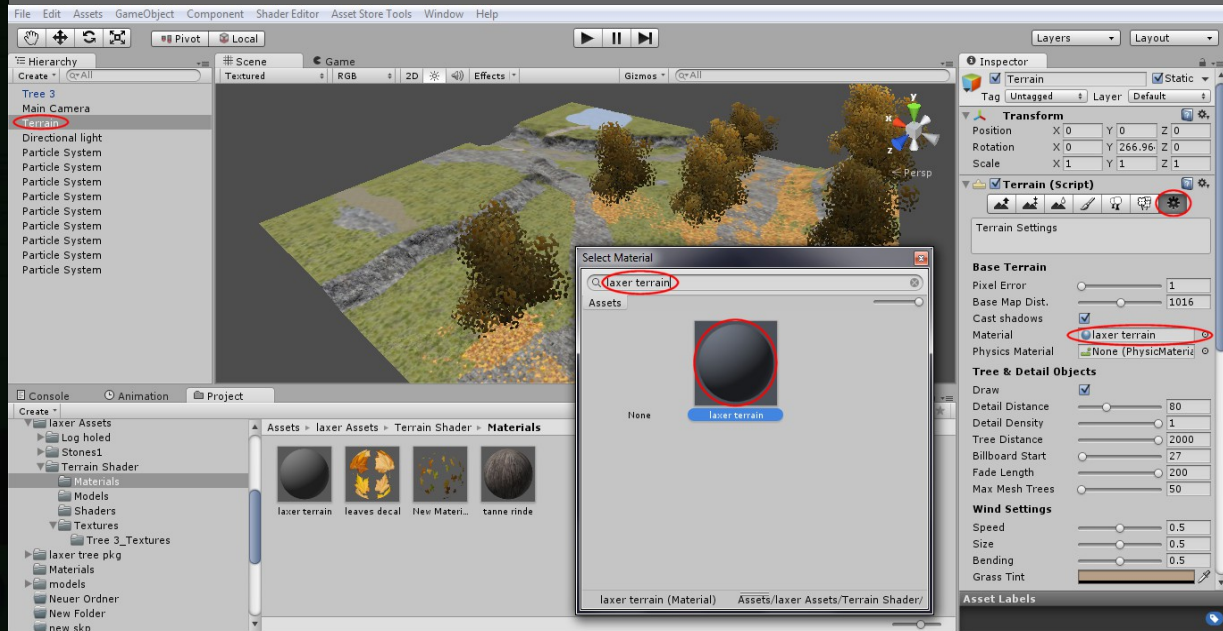
You can also use this shader for some special effects like spreading ice or growing plants.



# Applying the Shader

- select your terrain (1)
- open the terrain settings tab (2)
- open the material browser by clicking the dot on the right side (3)

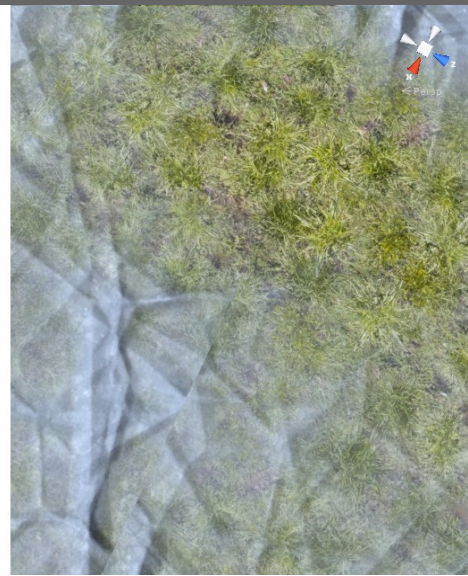
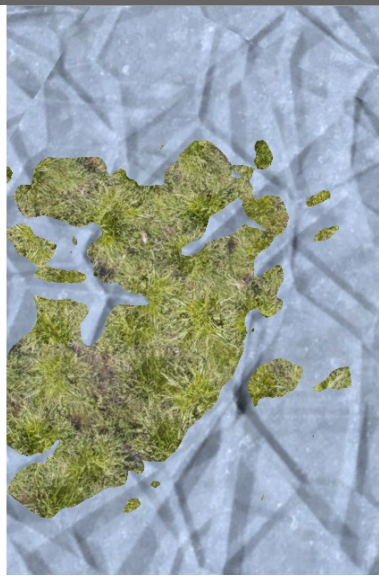
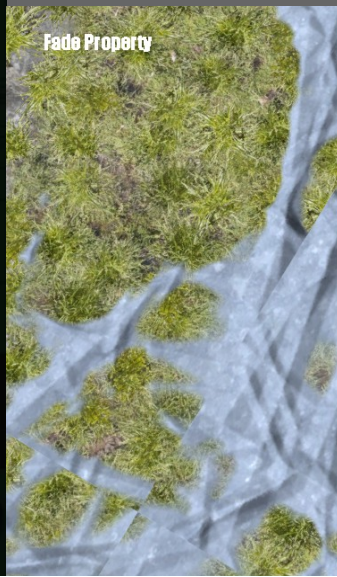
or simply drag & drop the material from the asset browser





## Applying the Shader

When you select the material in the asset browser you will see a slider called „Fade Length“ in the Inspector.  
This slider controls the blending as shown in the picture below.



## Creating Height-mapped Textures

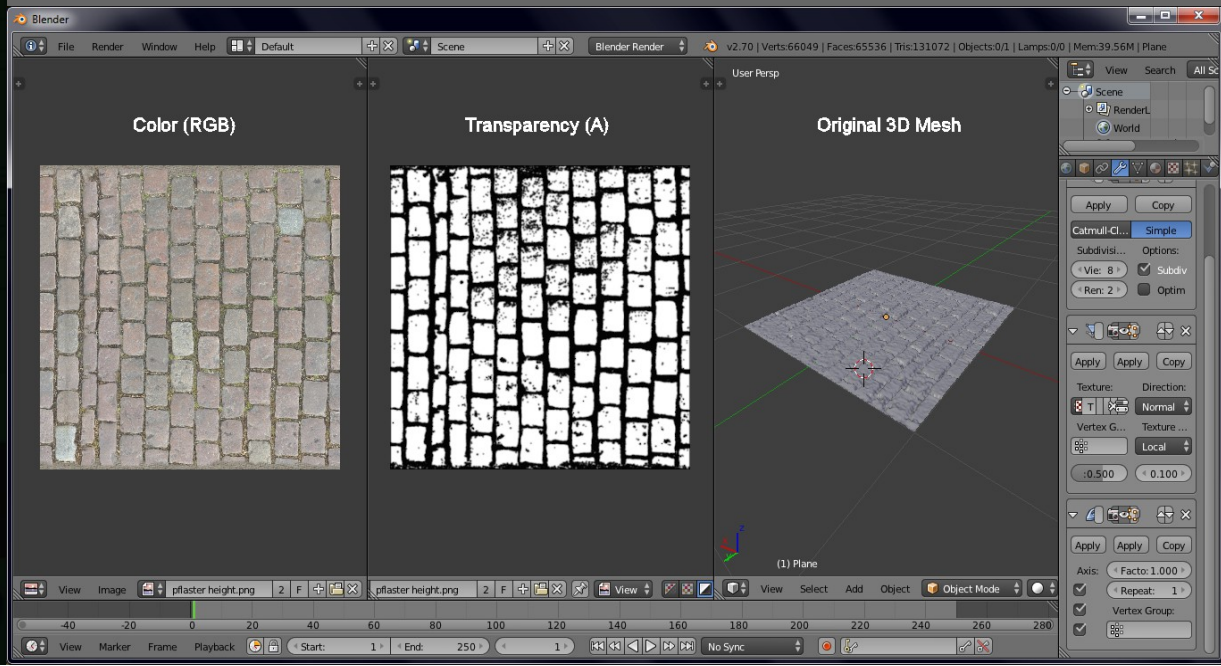
The transparency of each pixel defines its height.

The second picture shows the alpha values of the texture.

An alpha value of 1 (shown as white, not transparent) is the maximum height (1).

An alpha value of 0 (shown as black, completely transparent) is the minimum height (0).

**CAUTION!** Some image editing programs remove the color information of completely transparent pixels.



## Modifying the Shader

(for personal use only. please read the Asset Store agreement for further information

[http://unity3d.com/legal/as\\_terms](http://unity3d.com/legal/as_terms) )

using custom grass and tree shaders:

```
Dependency "AddPassShader" = "Hidden/Custom/Terrain/Diffuse fade addpass"  
Dependency "BaseMapShader" = "Diffuse"  
Dependency "Details0" = "Hidden/TerrainEngine/Details/Vertexlit"  
Dependency "Details1" = "Hidden/TerrainEngine/Details/WavingDoublePass"  
Dependency "Details2" = "Hidden/TerrainEngine/Details/BillboardWavingDoublePass"  
Dependency "Tree0" = "Hidden/TerrainEngine/BillboardTree"
```

You will find these lines at the end of the „custom terrain Bump fade“ shader or the „custom terrain Diffuse fade“ shader.

Do not change the „AddPassShader“ dependency.

You can replace the other dependencies with dependencies from other terrain shaders, or change the shader-path on the right side.

## The Mesh Shader

There are three mesh shaders included in this package:

You can find them under Shaders/Custom/HeightBlended/...

Diffuse x5

HeightBlendx5.shader

Bump x5

HeightBlendNormalx5.shader

Bumped Specular x5

HeightBlendSpecularx5.shader

Each of them can blend up to five textures.

These shaders use vertex-colors to blend the heightmapped textures.

Texture 0 is displayed, when the vertex color is (r = 0, b = 0, g = 0, a = 0).

Texture 1 is displayed, when the vertex color is (r = 1, b = 0, g = 0, a = 0).

Texture 2 is displayed, when the vertex color is (r = 0, b = 0, g = 1, a = 0).

Texture 3 is displayed, when the vertex color is (r = 0, b = 1, g = 0, a = 0).

Texture 4 is displayed, when the vertex color is (r = 0, b = 0, g = 0, a = 1).

You can set the vertex-colors with a 3d modelling program or with a vertex painter from the asset store.

