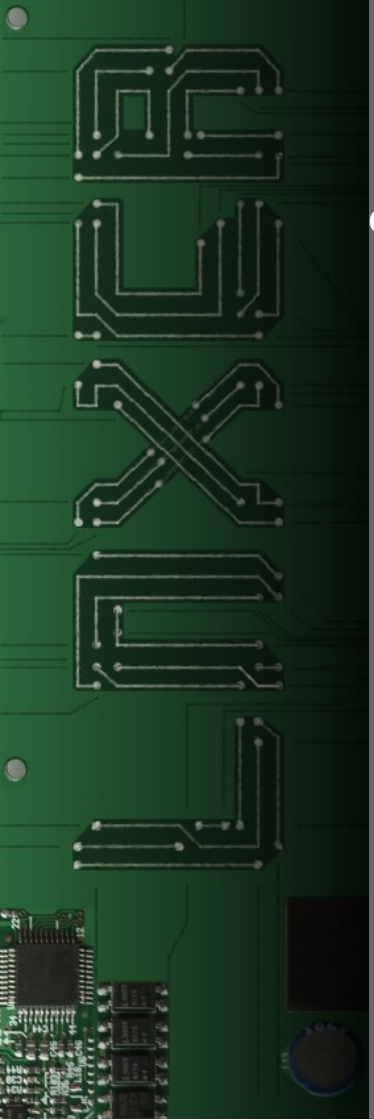


Automatic Terrain Blending

Thank you for choosing this package!
Here are some hints on using its content, so you can start right through.

Content:

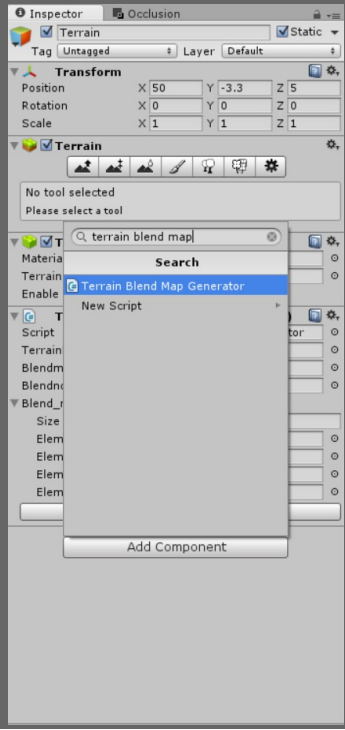
- **The Script**
- **The Shaders**
- **Applying A Shader**
- **Modifying The Shader**

laxer3d@gmx.de

The Script:

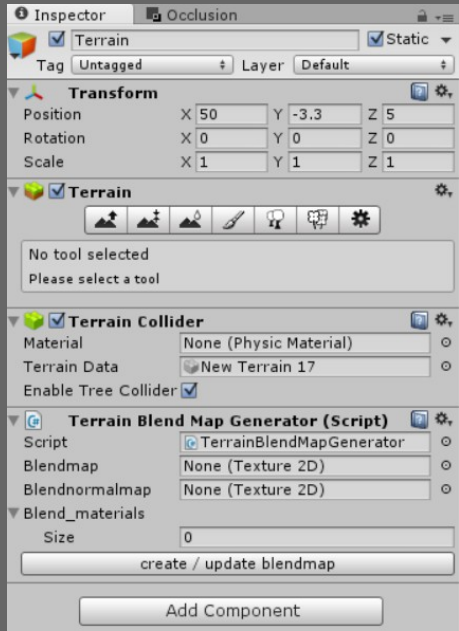
To get the terrain blending effect you have to attach the „Terrain Blend Map Generator“ script to your terrain.

- Select your terrain from the hierarchy or the scene view
- Click „Add Component“ in the inspector and search for „Terrain Blend Map Generator“
  Or drag and drop the script from the project browser
  (laxer assets/ Terrain mesh blending/ scripts / TerrainBlendMapGenerator.cs)



The Script:

The Inspector should now show the Blend Map Generator below the terrain components.

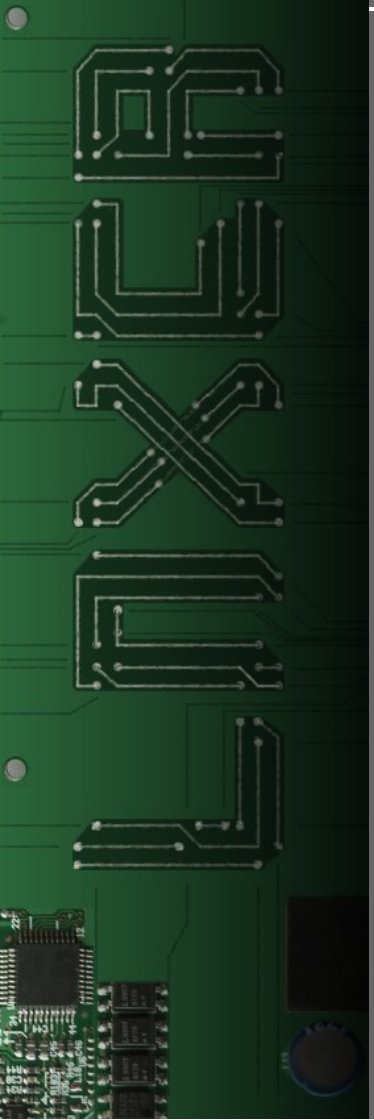


If you click „create / update blendmap“, the script will save two textures into your assets root folder and also update all materials from the Blend_materials list.

The blendmap will store the height of your terrain.
The blendnormalmap will store the terrains normals.

The Blend_materials list should contain all materials you want to use with this terrain. You can drag and drop terrain-blending materials into this list.
Note: These materials have to use one of the included shaders.
(For information on how to create these materials see 3: Applying A Shader)

Objects with terrain blending materials should be marked as static!



The Shaders:

This package contains 4 shaders for different terrain materials.

Standard Terrain:

- Simple standard shader
  a simplified shader for physically based lighting
- Full standard shader (does not work in deferred rendering)
  the default unity shader with all functions

Legacy Diffuse Terrain:

- Bumped Diffuse Shader

Legacy Specular Terrain:

- Bumped Specular Shader

There are also overlay shaders that only render the blending effect. You can duplicate your mesh and use one of these shaders to blend meshes with custom shaders.

Don't forget to mark the objects that use these shaders as static.

The color correction should be about 128 for gamma rendering and 188 for linear rendering.

Applying A Shader

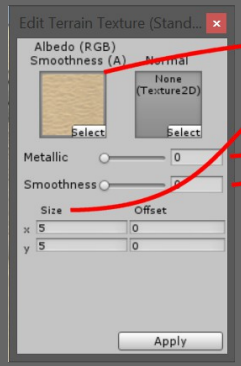
- Select or create a material
- Change the shader to a terrain blending shader (eg. Custom / TerrainMeshBlend / standard)
- Select the Terrain Texture you want to blend with the material
- Change the Tiling values of this texture to: 1 / TextureSize
- Change the Offset values of this texture to: 1 / TextureSize * TextureOffset

The Blend slider controlls the fading of the terrain texture.

The Blend Offset slider changes the height of the terrain texture (relative to terrain height)

Now you can add this material to the Blend Map Generator and click „create / update blendmap“.
(See 1: The Script)

Done!



Modifying the Shader

(for personal use only. please read the Asset Store agreement for further information
http://unity3d.com/legal/as_terms)

It is rather easy to modify an existing shader to use terrain blending. This guide works for the most shaders.
You can download built in shader sources from https://unity3d.com/get-unity/download/archive.

The terrain blending works with an additional pass, so you just have to add this pass at the end of the shader you want to modify.

1: open „laxer assets / terrain mesh blending / shaders / source / code parts.txt“
2: copy the first block of code and paste it at the end of the properties list into your shader. Eg:

```
Shader "Custom/TerrainMeshBlend/legacy Diffuse" {
    Properties {
        _Color ("Color", Color) = (1,1,1,1)
        _MainTex ("Albedo (RGB)", 2D) = "white" {}
        _BumpMap ("Normalmap", 2D) = "bump" {}
        [HideInInspector]_Blendmap ("Blendmap", 2D) = "black" {}
        [HideInInspector]_Blendnormalmap ("Blendmap normals", 2D) = "bump"
        _TerrainTex ("Albedo (RGB)", 2D) = "white" {}
        _Blend ("Blend", Range(0,10)) = 0.5
        _BlendOffset ("Blend Offset", Range(-5,5)) = 0.1
        [HideInInspector]_terrainheight ("terrainheight", float) = 600
        [HideInInspector]_terrainposy ("terrain location y", float) = 0
        [HideInInspector]_terrainmappos ("blendmap pos+scale", Vector) = (0,0,1,1)
    }
```

3: copy the second block of code and paste it at the end of the subshader in your shader. Eg:

```
15      SubShader {
16      Tags { "RenderType"="Opaque" }
17      LOD 200
18
19      CGPROGRAM
20      #pragma surface surf Lambert exclude_path:deferred
21
22      [.........]
23
24
37      ENDCG
38
39
40          CGPROGRAM
41          #pragma surface surf Lambert decal:blend vertex:vert exclude_path:deferred
42
43          #pragma target 3.0
44          #include "UnityCG.cginc"
45          sampler2D _Blendmap;
46
78      [.........]
79
80              fixed4 nmap = tex2D (_Blendnormalmap, (IN.uv_Blendmap) * _terrainmappos.zw );
81              //o.Albedo = tex2D (_Blendmap, IN.uv_Blendmap);
82              o.Normal = normalize(float3(nmap.x*2-1,nmap.y*2-1,nmap.z*2-1));
83              //o.Normal = UnityObjectToWorldNormal( float3(0,1,0));
84              // Metallic and smoothness come from slider variables
85              o.Alpha = c.a;
86          }
87          ENDCG
88      }
89      FallBack "Diffuse"
90  }
```