

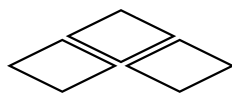
MERCATOR

RAPID TERRAIN PROTOTYPING
AND ITERATION FOR UNITY 3D



USER MANUAL

September 6, 2018



info@three-eyed-games.com
www.three-eyed-games.com

CHANGELOG

RELEASE 1 - JUNE 2018

- Initial release
- Patch 1.0.1: Restored consistency between linear and gamma workflows
- Patch 1.0.2: Fixed gaps when working with multiple terrains

QUICK START

1. Accept our gratitude for purchasing the asset.
2. Download `MERCATOR` from the Unity Asset Store, if you haven't already.
3. For Unity 2018: Enable the 'Allow unsafe code' option found in Edit → Project Settings → Player.
4. Add an empty Terrain to your scene:
GameObject → 3D Objects → Terrain
5. Change the Material setting to Custom in the Terrain settings and use any of the preconfigured `MERCATOR` Materials. You may also want to set Pixel Error to 1 and Terrain Width, Terrain Length and Terrain Height as desired.
6. Add a `Terrain Master` to your scene:
GameObject → 3D Objects → Mercator Terrain Master
7. In the `Terrain Master` inspector, set the Terrain property. Note that this will clear any contents of your terrain, so be careful!
8. Click 'Create Stamp'. A full-size `Terrain Stamp` covering your whole terrain will be created. Set the `Height Texture` and move / scale the stamp in Y-direction to create a basic ground level.
9. Duplicate the stamp and create your terrain! Set `Height Texture` and `Mask Texture`, and move, scale and rotate the stamp as you wish. Play around with the `Blend` parameter for different effects.

INTRODUCTION

MERCATOR is a system allowing for rapid prototyping and iteration of terrain in Unity 3D. It does not aspire to be the most powerful, but the easiest and fastest tool to let you explore and shape the landscape you want to portray.

The system is comprised of two main parts. The first one is a terrain creation system that allows you to shape and modify the terrain with intuitive, stamp-like elements that influence its height map. Thanks to GPU-supported computing, your changes are visible in real-time without waiting for a baking process to finish. The full history of the Terrain is displayed in the Unity hierarchy window, obviating the need for stepping through history to undo changes.

This terrain creation system is complemented by a highly flexible shader that allows you to place textures on your terrain without the need for painting or baking splat maps. The main advantage of the MERCATOR shader is that it reacts immediately to changes in the terrain, without requiring any baking or processing. It should be used to immediately get a feel for the terrain that you are creating and to experiment freely with different visuals. Note that for final production use, there are other shader packages that provide both better performance and visual quality.

MERCATOR COMPONENTS

USING THE COMPONENTS

In your scene, create a terrain (GameObject → 3D Object → Terrain) and a Terrain Master (GameObject → 3D Object → Mercator Terrain Master). Assign your terrain and click 'Create Stamp' to start.

TERRAIN MASTER

This is the core of MERCATOR's terrain editing capabilities. On every update step, it resets the terrain and collects a list of all Terrain Stamps found among the children. Their effects are then applied in top to bottom order. This allows you to reorder, modify or delete any step you took to create your terrain. You can also add more than one Terrain Master component to modify multiple terrains at once.

Terrain: The Unity Terrain that is written by the Terrain Master. Please note that any contents of that terrain will be overwritten, so make sure to backup what you don't want to lose. Details can be found on page 8.

Create Stamp: Click this button to create a Terrain Stamp.

TERRAIN STAMP

A *stamp* is a single texture-based modification that is applied to the terrain. Basically, a stamp will write a user-specified masked height texture to the terrain's height map. Stamps can be moved, rotated and scaled freely to affect the desired area of the landscape.

The textures come from special components called *Mercator Texture*. This allows for future extensibility of the system with generated textures like noise or fractals.

Blend: Similar to the way layers work in common graphics programs, stamps specify a blend mode that decides how the height texture is applied to the height map. The *Default* blend mode overlays the height texture with simple alpha blending. The *Raise* and *Lower* blend modes take the maximum or minimum of both textures, useful to create hard transitions or ridges. The *Modulate* blend mode combines both new and existing height values.

Opacity: Use this property to scale the influence of the stamp, where 0 means that the stamp won't have any effect at all.

Height Texture: This Texture component provides the height data that is used for the stamp's operation. Please make sure to use textures with an appropriate bit depth, as detailed on page 8.

Height Power: This slider allows you to adjust the height texture before it is applied. The values are raised to the power of 0.1 to 10.

Mask Texture: This Texture component provides the mask that determines the stamp's influence. The stamp will have full effect in white areas, and no effect at all in black areas. Again, please make sure to use textures with an appropriate bit depth.

Mask Power: This slider allows you to adjust the mask texture before it is applied. The values are raised to the power of 0.1 to 10.

LAYERED TERRAIN SHADER

MERCATOR's shader gives you control over six layers of textures that are procedurally blended together based on user-defined criteria. The layers are applied in order, meaning that Layer 1 will be applied first, and Layer 6 will be blended on top of all others.

The Layered Terrain Shader allows you to experiment with the visuals of your terrain and quickly find the direction you are going for. Please note that it is not designed with performance in mind. While it may be suitable for some games and platforms, we advise you to profile your game and potentially replace the shader with a more advanced terrain shading system.

USING THE SHADER

Create a new Material (Assets → Create → Material). In the material's inspector, click the Shader dropdown at the very top and select Mercator/Layered Terrain Shader. Select your terrain and go the Terrain Settings by clicking the gearwheel icon in the inspector. Here, click the Material dropdown and select Custom. Drag in your newly created material.

SETTINGS



Here you control the overall settings of the Material.

Height Minimum / Maximum: Specify the minimum and maximum height of your terrain. The layer's height values are relative to this range, so you can reuse your Materials for terrains of different heights.

Magenta Clip Warning: By using this option, your terrain will be painted in bright magenta where height values are dangerously low or high. This can help you avoid clipping of valleys or mountain peaks.

Debug Mode: You can temporarily set the shader to show a debug visualization which can help you understand the system better. *Height* represents the terrain height as false colors, where blue is low and red is high. *Normals* displays the world normals, i.e. the directions the surfaces are facing. *Slope* shows whether the terrain is flat (black) or steep (white). *Identify RGBCMY* displays the contribution of the six layers in red, green, blue, cyan, magenta and yellow.

LAYER



Here you control the individual settings of each layer. Use the buttons below to reorder, copy and paste layers as required.

Albedo Texture: This texture contains the color in the RGB channels and glossiness in alpha.

Normal Texture: This texture contains a tangent-space normal map to add surface detail.

Glossiness: The glossiness of the layer determines the sharpness of reflections. This is multiplied with the value from the Albedo texture's alpha channel. Set it to 1 if you want to rely purely on the texture.

Metallic: A metallic surface will reflect the environment, but not show any diffuse reflection.

Color: The albedo texture is tinted with this color. By lowering the alpha value, you can reduce the overall contribution of the layer.

UV Scale: The size of your texture in meters.

UV Rotation: The rotation of your texture in degrees.

TILING REDUCTION

Repetition of textures can be quite striking, known as tiling artifacts. To alleviate this, the Layered Terrain Shader allows you to apply tiling reduction to each layers by blending the texture with a scaled or rotated copy of itself.

UV2 Scale: The relative size of your texture's copy. This is multiplied with the texture's overall UV Scale, so [1, 1] leaves the scale untouched.

UV2 Rotation: The relative rotation of your texture's copy in degrees. Rotation is applied on top of texture's overall UV Rotation, so 0 leaves the rotation untouched.

Noise Factor: Determines the hardness of noise used for the blending of texture original and copy. If this factor is set to 0, the blend will be smooth.

Noise Scale: Determines the scale of the noise used for blending of texture original and copy. No effect if Noise Factor is set to 0.

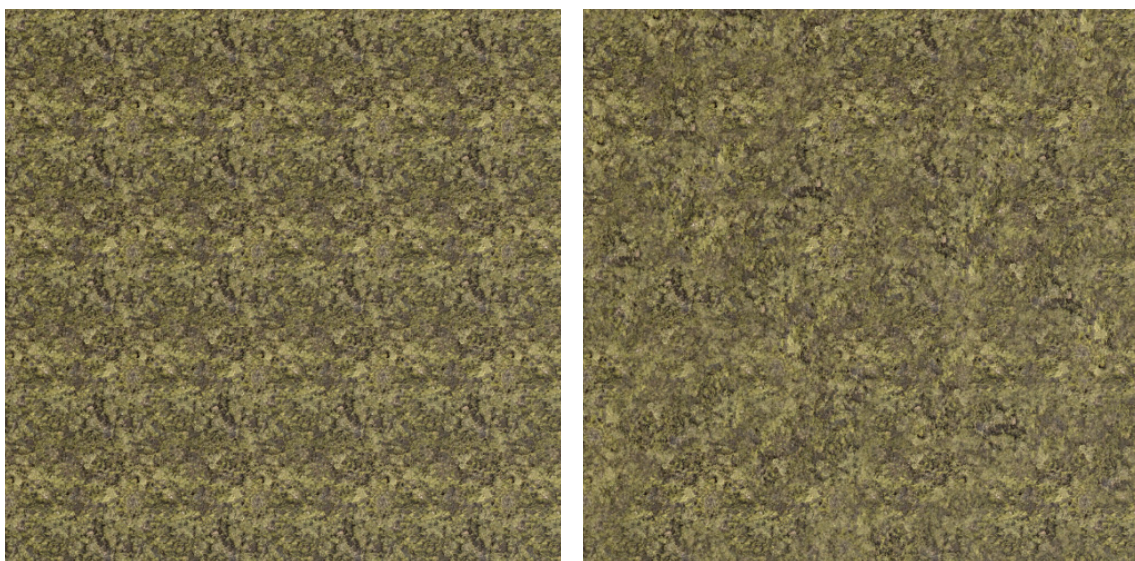


Figure 1: A texture without (left) and with (right) tiling reduction.

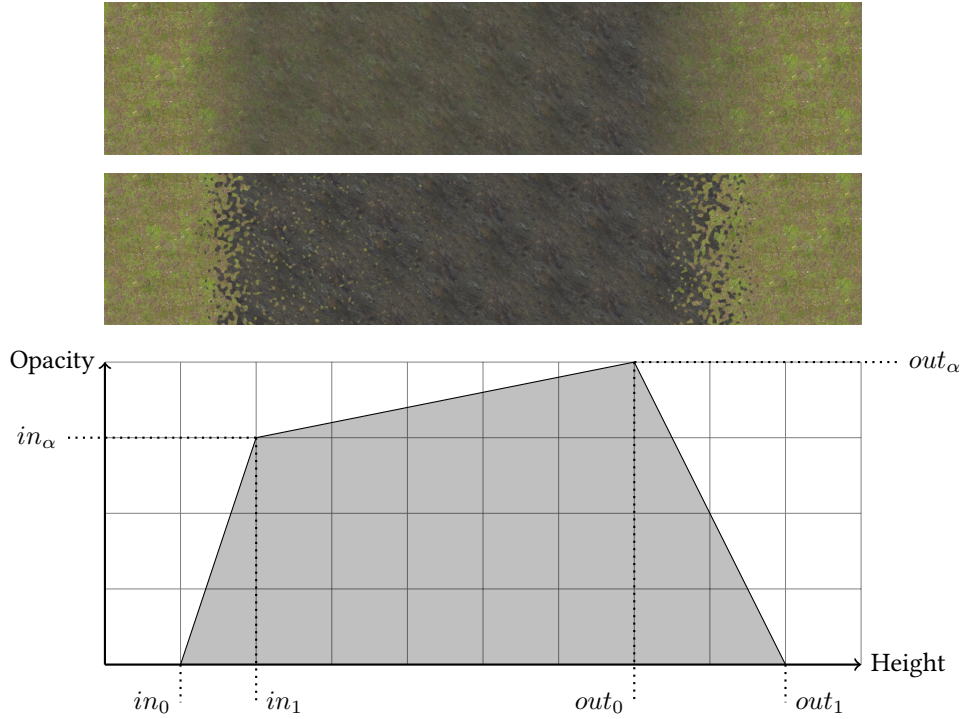


Figure 2: Rock layer on top of a grass layer. The terrain is viewed from above. The left border is minimum height, the right border is maximum height. From top to bottom: alpha blending with Noise Factor 0, noise-based blending with Noise Factor 3, a schematic showing the values involved in the blending calculations: $in_0 = 0.1, in_1 = 0.2, out_0 = 0.7, out_1 = 0.9, in_\alpha = 0.75, out_\alpha = 1$

BLENDING

Noise Factor: Determines the hardness of noise used for the blending of a layer on top of the underlying layers. If this factor is set to 0, the blend will be smooth.

Noise Scale: Determines the scale of the noise used for blending. No effect if Noise Factor is set to 0.

Bias: Adds an absolute value to the layer's contribution for special effects.

HEIGHT AND SLOPE

The Layered Terrain Shader currently allows you to restrict a layer's contribution based on two factors: height and slope. Since the height range of the terrain is specified in the Settings, both height and slope are in range $[0, 1]$ and can thus be configured in exactly the same way.

You can combine both factors to achieve the desired blending. As an example, you might want to overlay a snow layer above a certain height, but only on surfaces upright enough for the snow to settle.

Fade-in Start / End: These values, denoted as in_0 and in_1 in this document, determine the range where the contribution of the layer is faded in. Below in_0 , the layer has no contribution at all.

Fade-out Start / End: These values, denoted as out_0 and out_1 in this document, determine the range where the contribution of the layer is faded out. Above out_1 , the layer has no contribution at all.

Fade-in / -out Alpha: These values, denoted as in_α and out_α in this document, determine the layer's contribution at in_1 and out_0 , respectively. In between, the alpha values are interpolated.

HANDLING STAMP TEXTURES

CUSTOM TEXTURES

When adding custom height or mask textures to `MERCATOR`, make sure that you use an appropriate bit depth of at least 16 bit to avoid banding artifacts.

- Unity does not play well with 16 bit formats, happily importing them with 8 bit precision. It is recommended to use 32 bit HDR or EXR files.
- In the Unity import settings of the texture, set both 'Non Power of 2' and 'Compression' to 'None'.
- Use the `Height Power` and `Mask Power` to correct any gamma-related mistakes that can happen during conversion.

USING AN EXISTING TERRAIN

If you have an existing terrain that you want to extend with `MERCATOR`, follow these steps:

- Select your Terrain. In the Inspector, go to the Terrain Settings and click the 'Export Raw' button.
- Import the raw pixel data in your favourite raster image editor.
 - Photoshop will directly recognize the file format.
 - In GIMP, select File → Open and explicitly set the file type to *Raw image data*.
- Set the image mode to RGB and 32 bit precision.
- Export the image as EXR or HDR and save it somewhere in your Unity project's Assets folder.
- In the Unity import settings of the texture, set both 'Non Power of 2' and 'Compression' to 'None'.
- Create a `MERCATOR` stamp and assign your texture.
- Flip the X scale of the stamp (e.g. set 500 to -500).
- If the result doesn't look right, try setting the stamp's `Height Power` to 0.45454545 ($\approx \frac{1}{2.2}$).