



MapMagic Brush is a module for MapMagic 2 asset (free, comes with this package). It is terrain sculpting and painting tool tool, and can be used to paint height, textures, objects, trees and grass/detail, and event at the same time. The brush effect is defined with MapMagic nodes graph: this way you can create the exact brush behavior you like.

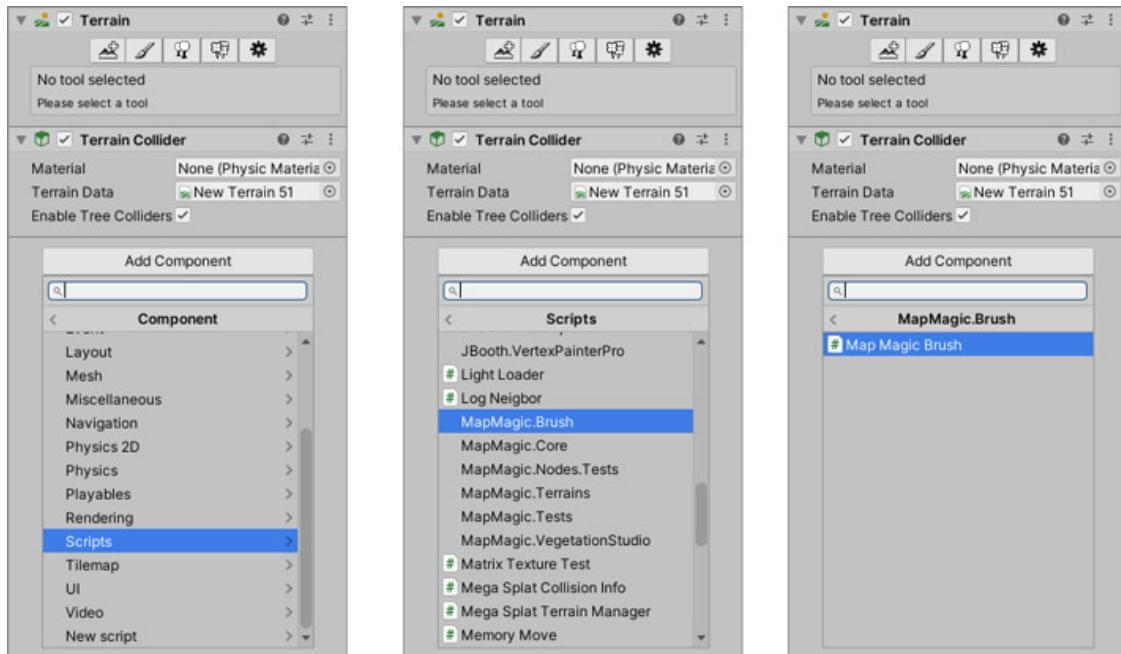
Quick Start

[Video tutorial](#) that might help at the beginning.

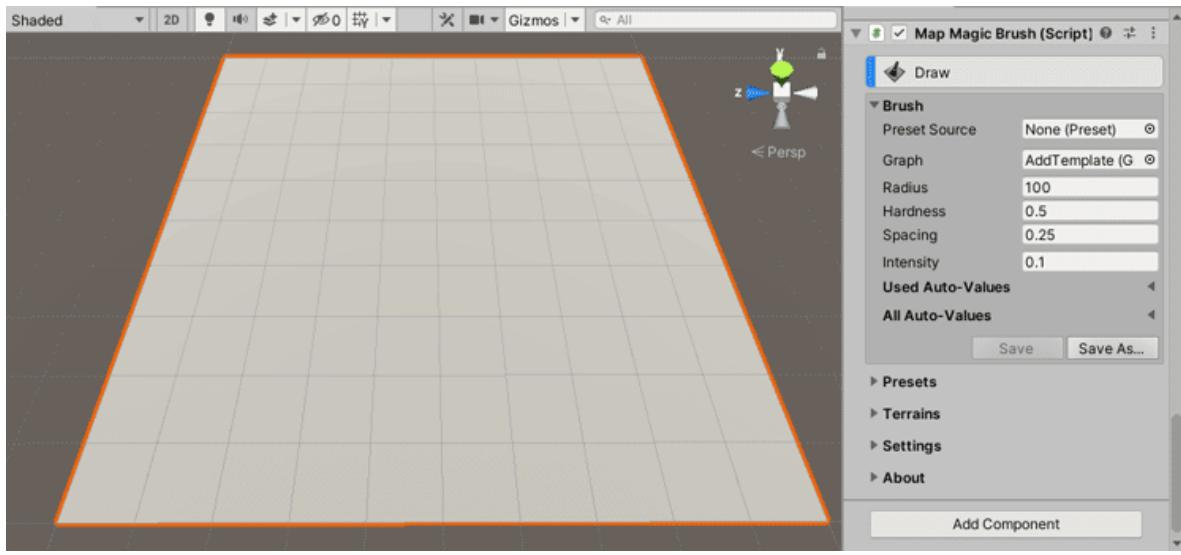
Single Terrain painting

Quick steps on how to start painting with MapMagic Brush on a single terrain:

1. Create a new terrain or select already existing one
2. Add a MapMagic Brush component to terrain You can find it in Scripts -> MapMagic.Brush, or by searching MapMagic Brush in a search field.



3. Press the Draw button to start drawing on the terrain.



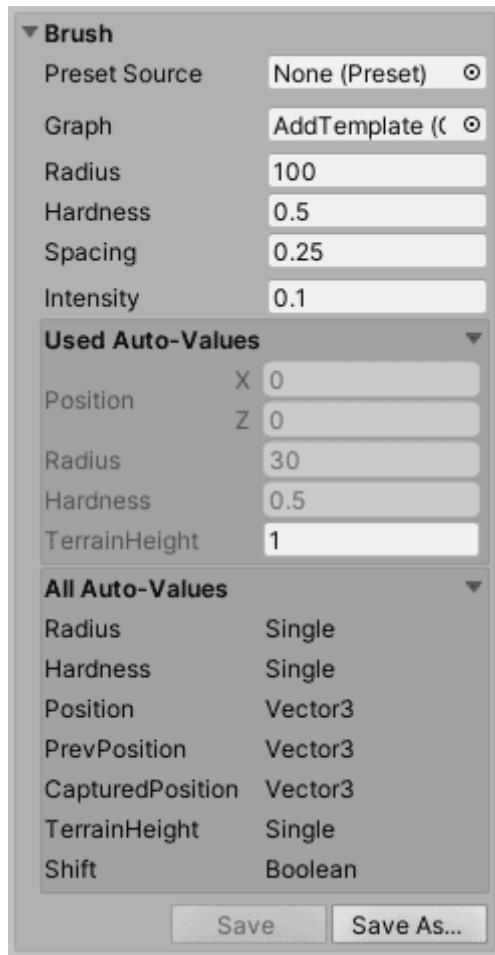
Multiple Terrain Painting

1. Create an empty game object (it could be a parent to all of terrains, or could be any object in scene);
2. Add MapMagic Brush component to it;
3. Open up Brush's Terrains foldout, and click Add -> All Terrains. This will add all of the scene terrains to terrain list.



Note that all terrains should have the same heightmap, splatmap and grass resolution and the same size in world units. If some of the terrains have different resolution or size they won't be added, and you will see the message box displaying the terrains that were excluded.

Brush Menu



Preset Source: brush preset that was used to set these brush properties.

Brush preset is an asset file that stores data of brush settings: graph, radius, hardness, spacing, and all of the graph's exposed settings (like intensity, grass texture, etc.). By assigning the preset asset to the Preset Source it copies all of the settings to brush instance. This is a quick way of switching between different brush templates.

Note that current brush properties are a copy of the Preset Source. Changing these values from brush object interface will not change the source values. This way you can assign saved brush and then feel free to change its radius, intensity, or any other value.

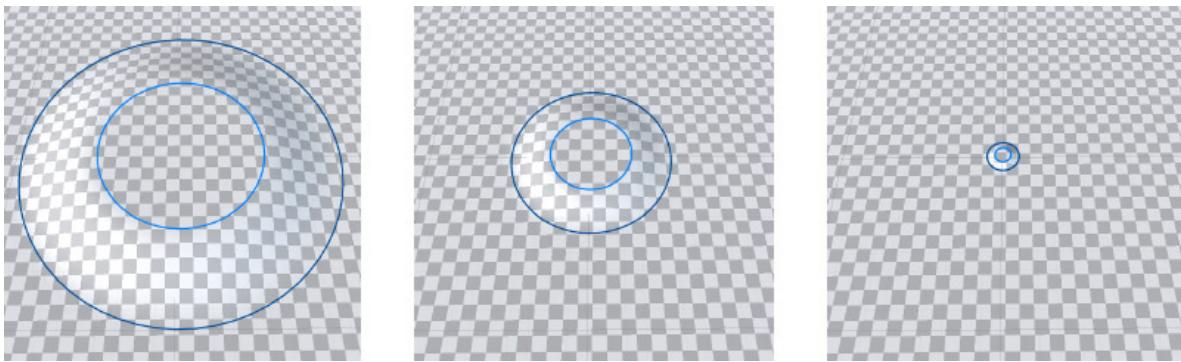
To store current brush settings back to Preset Source press Save button. To save current settings as a new preset use Save As button.

Graph: currently assigned MapMagic Brush graph.

MapMagic Brush graphs require special brush input nodes to read the terrain, and special output nodes to store the modified terrain parts. The usual MapMagic graphs, or tutorial graphs that come with Core package won't work.

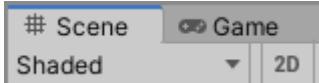
Double-clicking the graph will open a graph editor window.

Radius: the size of the brush that is applied to terrain.



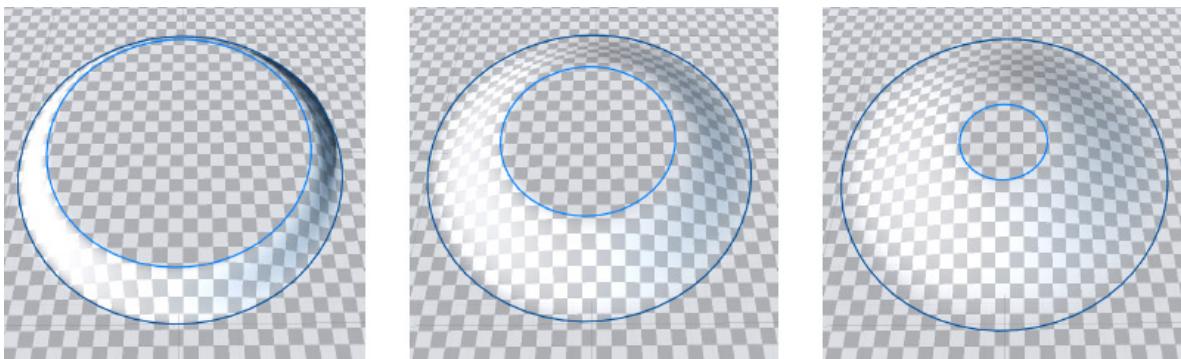
Brush Radius: 100, 50, 10

You can change the brush radius with the keyboard buttons [and]. Note that the keyboard works only when scene view is focused, and the Scene tab is highlighted with blue line:



Note that some brushes use Radius parameter in other exposed variables, so it sets not only the brush size. For example, in Add graph Radius value multiplies an exposed Intensity value - this way it changes the amount the brush adds as well, maintaining the overall look of the brush effect.

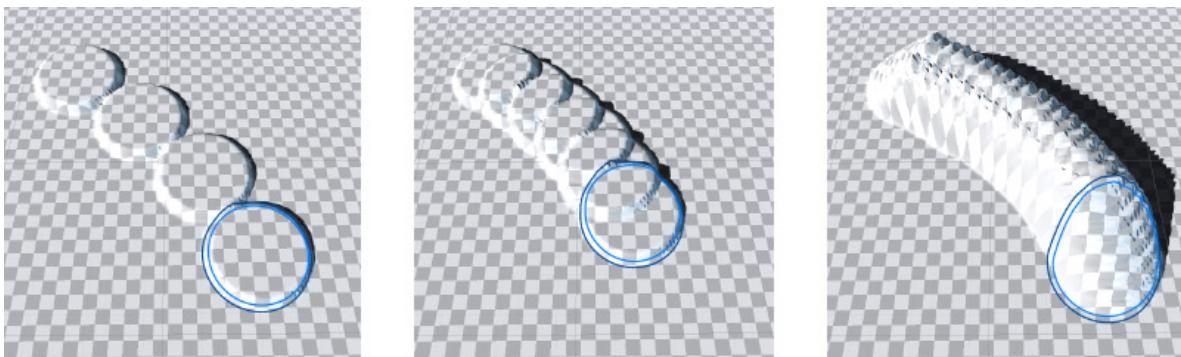
Hardness: Hardness value sets the radius percent of the brush that is applied with full opacity. The brush gizmo in scene view consists of two circles - the inner one shows where the brush will have 100% effect, and the outer one - where its effect will fall down to zero. In the area between circles the brush effect will gradually change.



Brush Hardness: 0.75, 0.5, 0.25

Similarly to radius, this parameter could be read by graph to modify additional values.

Spacing: defines the maximum distance the brush repeats itself when stroke on terrain. For example, if this value of 0.25 make it repeat each quarter of radius. If set to 1 it will create the stamps are applied every full radius. And if I set it to 2 it repeats each diameter, creating a chain of circles.



Brush Spacing: 2, 1, 0.25

When set to lower values it will apply the brush effect more frequent, calling brush graph on the same place more often. It visually creates the effect of higher intensity.

Unlike Radius and Hardness, this parameter is not used in graph.

Intensity, Texture, TerrainLayer, or other parameters listed below Radius/Hardness/Spacing - are the parameters exposed in used graph. Quick guide on how to expose a value:

- right-click on any node value in graph,
- select Value->Expose,
- enter the value name
- click Assign to add the value in graph overrides

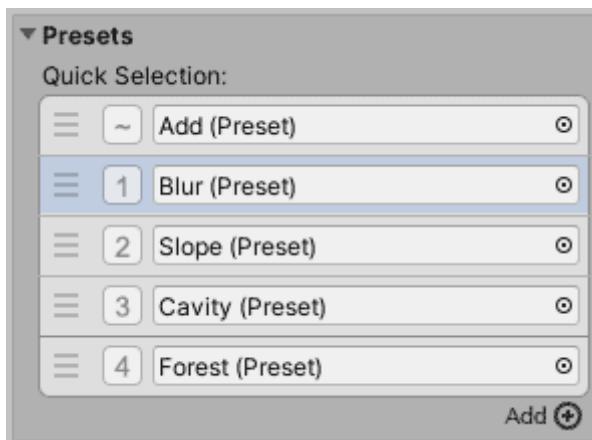
Once done you can see the value appeared in graph Overridden Variables, and in brush properties as well. These values are stored in brush preset together with radius/hardness/spacing.

Used Auto-values foldout: properties names and values automatically provided to the graph (assigned to it's overridden variables) on last stroke. Contains Radius/Hardness with some other properties that could be used by this graph.

All Auto-values foldout: just an information box for mentioning all property names and their types that could be automatically provided to graph. Useful for designing new graphs and exposing the values.

- **Radius**, float;
- **Hardness**, float;
- **Position**, Vector3: current stroke position in world units. Brush height is taken into account as well (y coordinate).
- **PrevPosition**, Vector3: previous stroke Position in world units. Useful for drawing line with splines module.
- **CapturedPosition**, Vector3: special brush Position selected with Ctrl-click. Could be used for presets like Level (which takes CapturedPosition Y coordinate as level to paint).
- **TerrainHeight**, float: current terrain height in world units. To transform Position height (Y) world-unit coordinate to map value with 0-1 range.
- **Shift**, boolean or int: set to true or 1 if Shift button is pressed during the stroke, false or 0 if it isn't. Useful for an alternative brush mode (like erasing instead of adding).

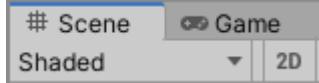
Presets Menu



Menu to quickly switch between different brush presets. Selecting a preset will assign it to the Brush menu Preset Source slot, and thus will copy its parameters to Brush.

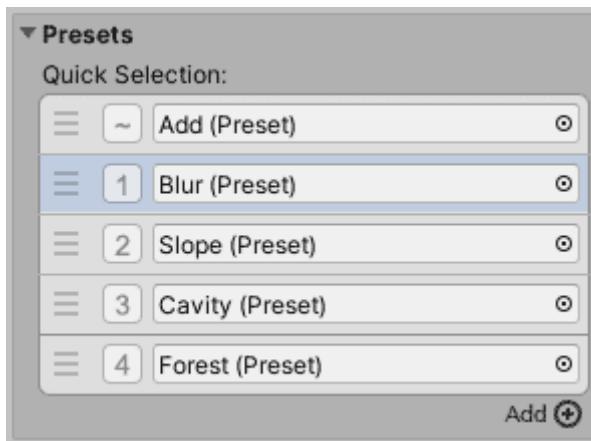
Brush preset is an asset file that stores data of brush settings: graph, radius, hardness, spacing, and all of the graph's exposed settings (like intensity, grass texture, etc.). By assigning the preset asset it copies all of the settings to brush instance.

Preset could be selected by clicking on ~ or number icon on the preset slot, or could be selected with keyboard by pressing a corresponding key on a main keyboard (not numeric keypad). Note that the keyboard works only when scene view is focused, and the Scene tab is highlighted with blue line:



Add new slots with an Add button. The slots could be dragged by clicking and dragging on the triple line icon at the left. Note that changing the presets order will assign new keyboard keys according to the presets numbers in list. The list key order - ~,1,2,3,etc - will remain the same. To remove the preset from list drag it to Remove spot that appears at the place of Add button when dragging has started.

Terrains Menu



Menu to quickly switch between different brush presets. Selecting a preset will assign it to the Brush menu Preset Source slot, and thus will copy its parameters to Brush.

Brush preset is an asset file that stores data of brush settings: graph, radius, hardness, spacing, and all of the graph's exposed settings (like intensity, grass texture, etc.). By assigning the preset asset it copies all of the settings to brush instance.

Preset could be selected by clicking on ~ or number icon on the preset slot, or could be selected with keyboard by pressing a corresponding key on a main keyboard (not numeric keypad). Note that the keyboard works only when scene view is focused, and the Scene tab is highlighted with blue line:

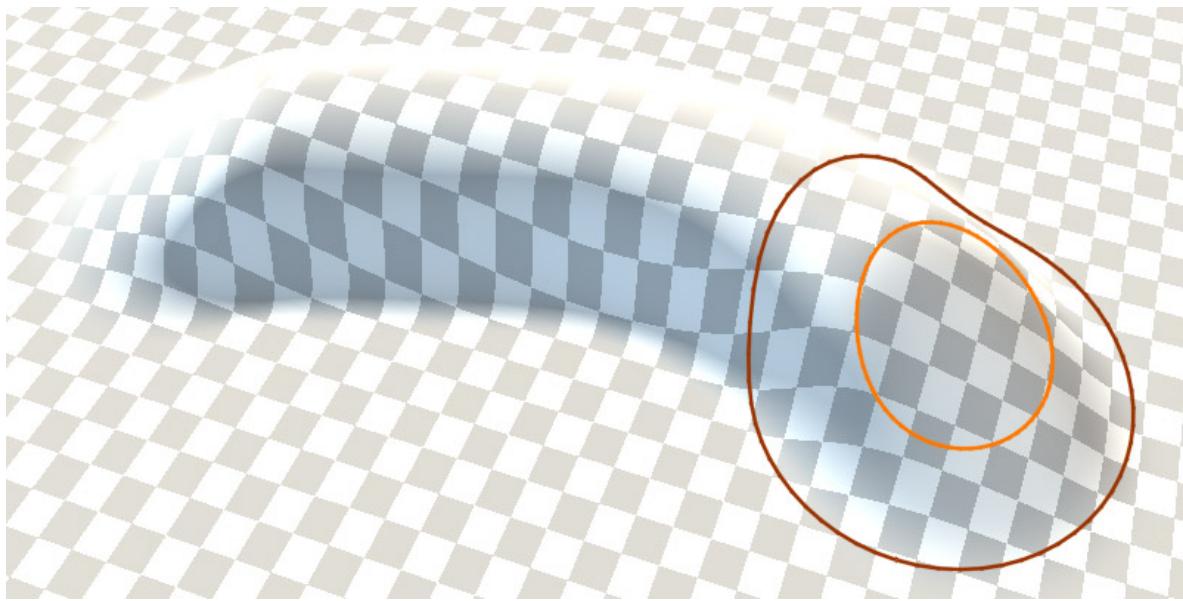


Add new slots with an Add button. The slots could be dragged by clicking and dragging on the triple line icon at the left. Note that changing the presets order will assign new keyboard keys according to the presets numbers in list. The list key order - ~,1,2,3,etc - will remain the same. To remove the preset from list drag it to Remove spot that appears at the place of Add button when dragging has started.

Included Presets

Set of brush presets MapMagic Brush is shipped with. Could be used for level design directly, and as an examples for creating own brushes as well.

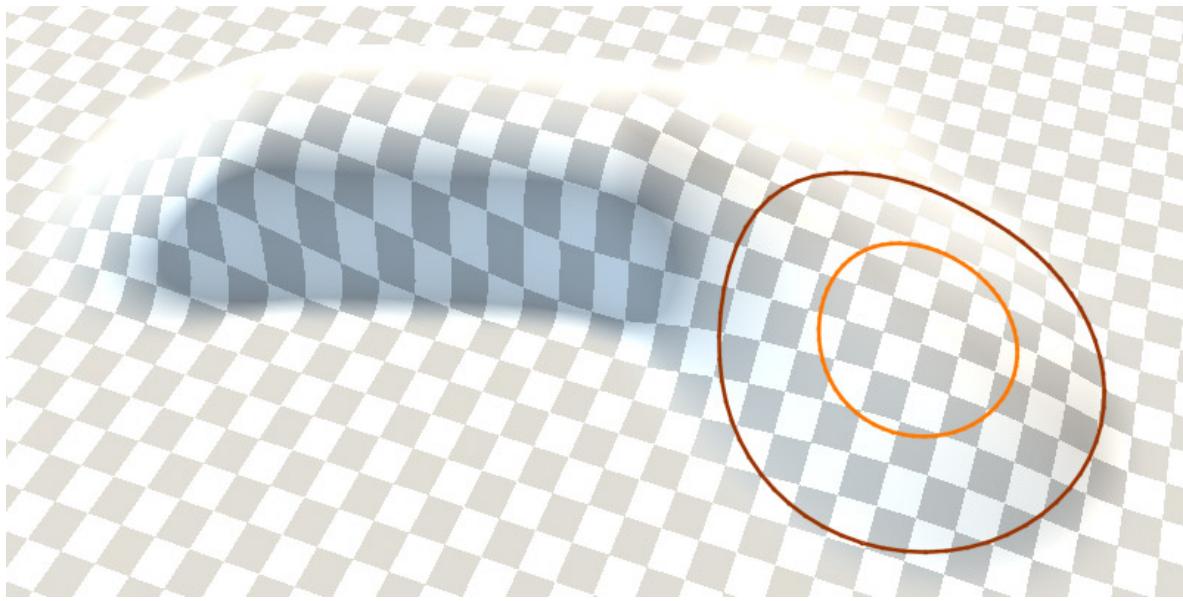
Add



Adds height to terrain. In inverted mode (with shift button pressed) subtracts from terrain.

- **Intensity:** the ratio of amount it adds to terrain relatively the brush size. Total added height = Radius * Intensity.

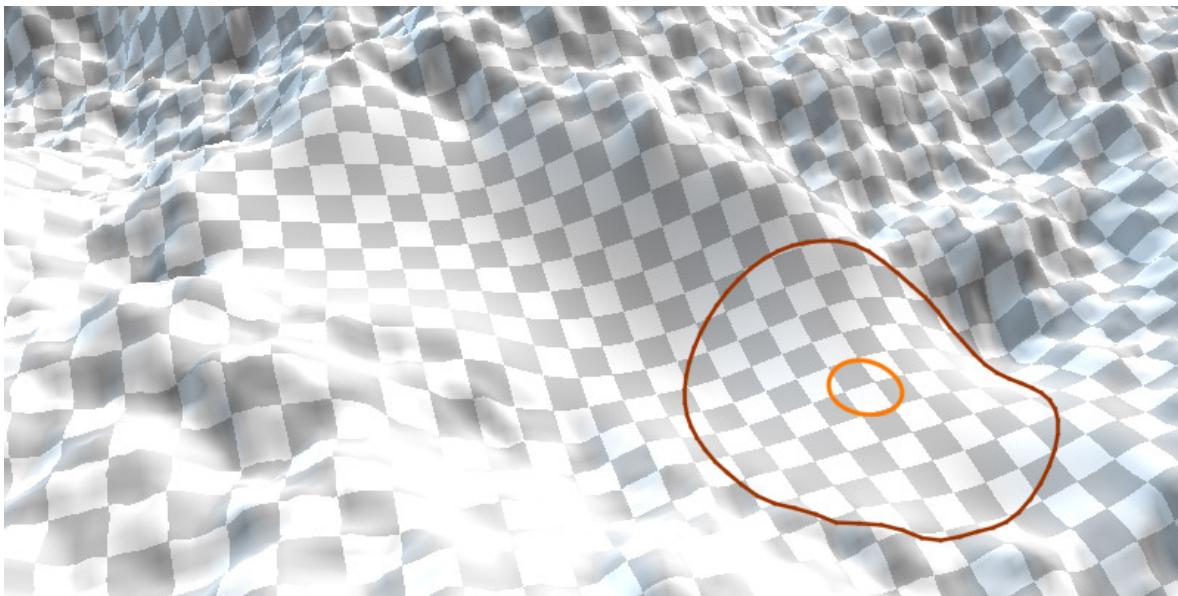
AddBlur



Adds height to terrain in standard mode, but blurs terrain when Shift is pressed. Useful for quick terrain editing when add and blur are changed quite often.

- **Intensity:** similar to Add, the amount added to terrain
- **Downsample:** similar to Blur, the number of times precessed texture is reduced
- **Blur:** similar to Blur, blur amount (iterations).

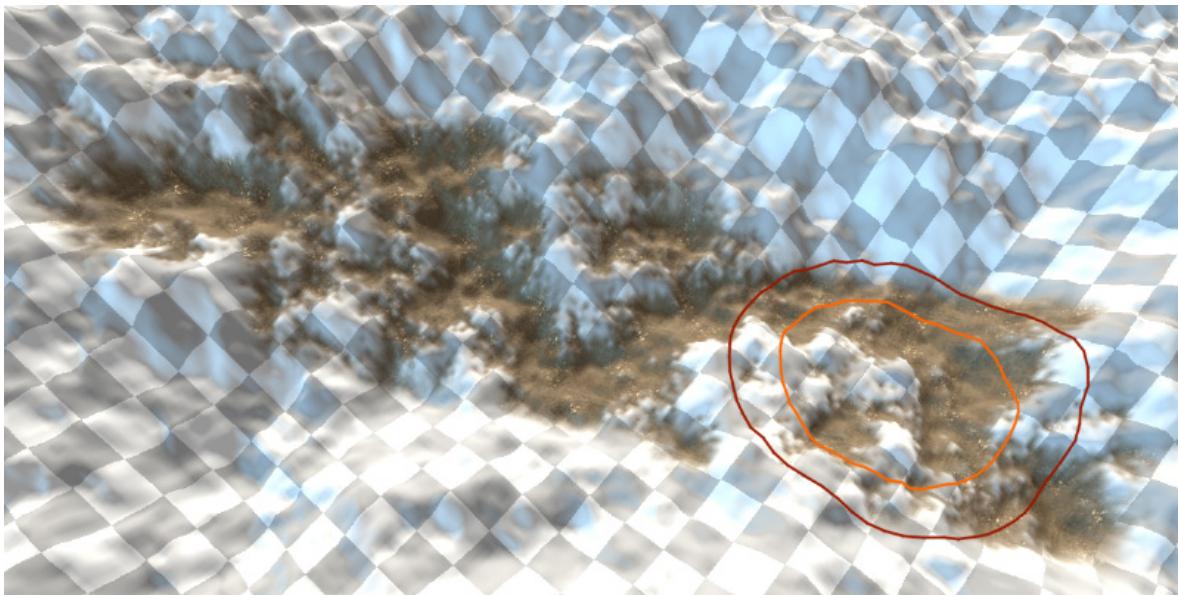
Blur



Smooths the terrain.

- **Downsample:** the number of times processed map is reduced. Its brush resolution is 100 pixels, then downsample of 2 will resize it to 25 pixels, apply blur, and then upsize to 100. Increasing downsample increases the blur amount drastically.
- **Blur:** the number of blur iterations. If it's not integer then the last operation blur amount is performed with reducing factor. For example, for value of 2.33 it will make 3 iterations, the last one with 0.33 blur amount.

Cavity



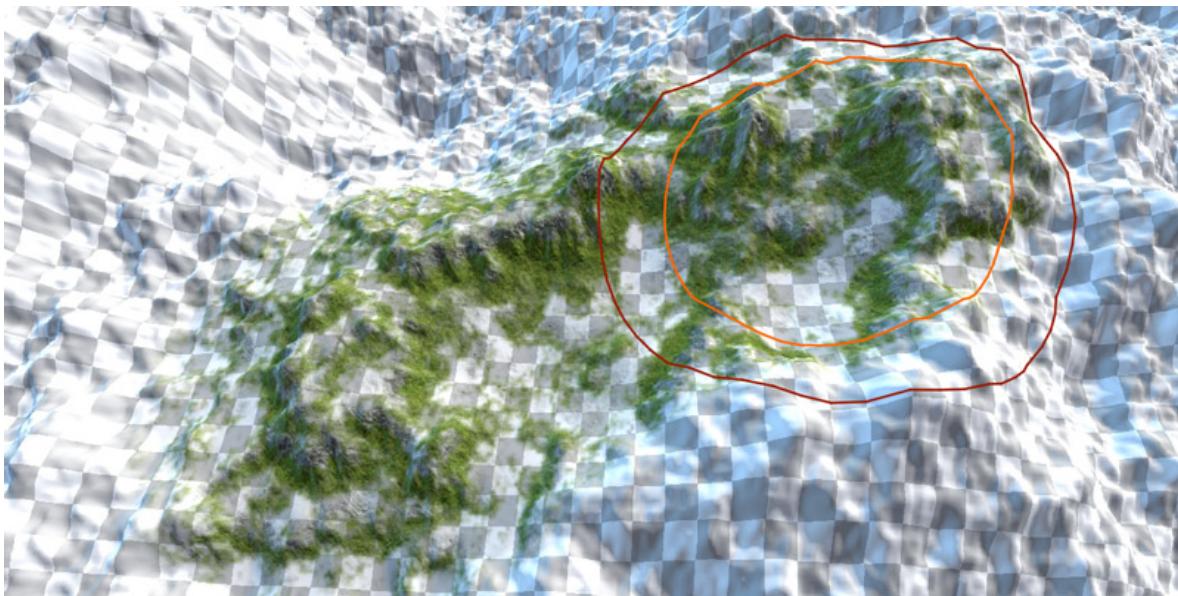
Draws with a selected texture only in hollows.

In inverted mode will draw with textures on the concave areas.

Note that when using lower splats resolution than the heightmap MM Brush will upscale splats to match read heights. This will make the operating heights

- **Intensity:** the maximum opacity value of applied terrain layer
- **TerrainLayer:** the terrain layer (land texture) to paint with

Erosion



Applies erosion to terrain and paints terrain with cliff (land texture that was eroded) and sediment (land texture that was appended by settling down eroded).

- **Intensity:** the amount of erosion applied each brush stamp
- **Blur:** the amount of blur applied to smoothen down erosion
- **CliffLayer:** eroded land texture (terrain layer)
- **SedimentLayer:** appended land texture (terrain layer)

Forest



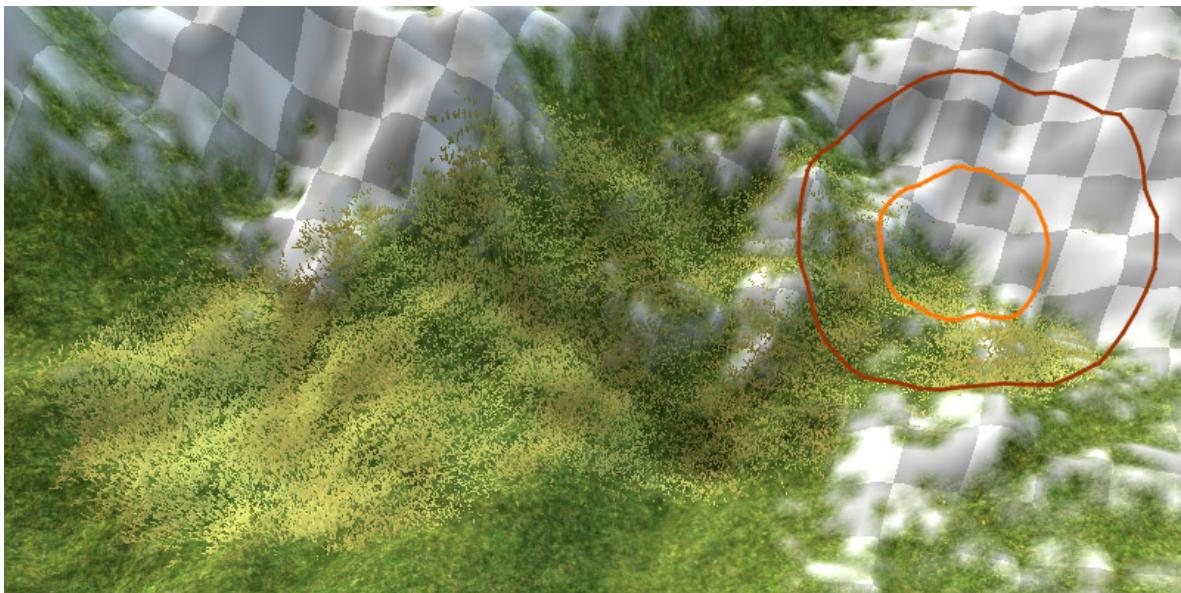
Plants trees on a stroke location. However does it only on a specific terrain layer. On apply it will read the trees, grow them, and add new seedlings if possible.

By growing meant adding a decimal number after scale dot, in years. For example, trees aged 25 will have scale of 1.025

Requires an Objects module.

- **SoilLayer:** the land texture (terrain layer) to plant the trees on

Grass

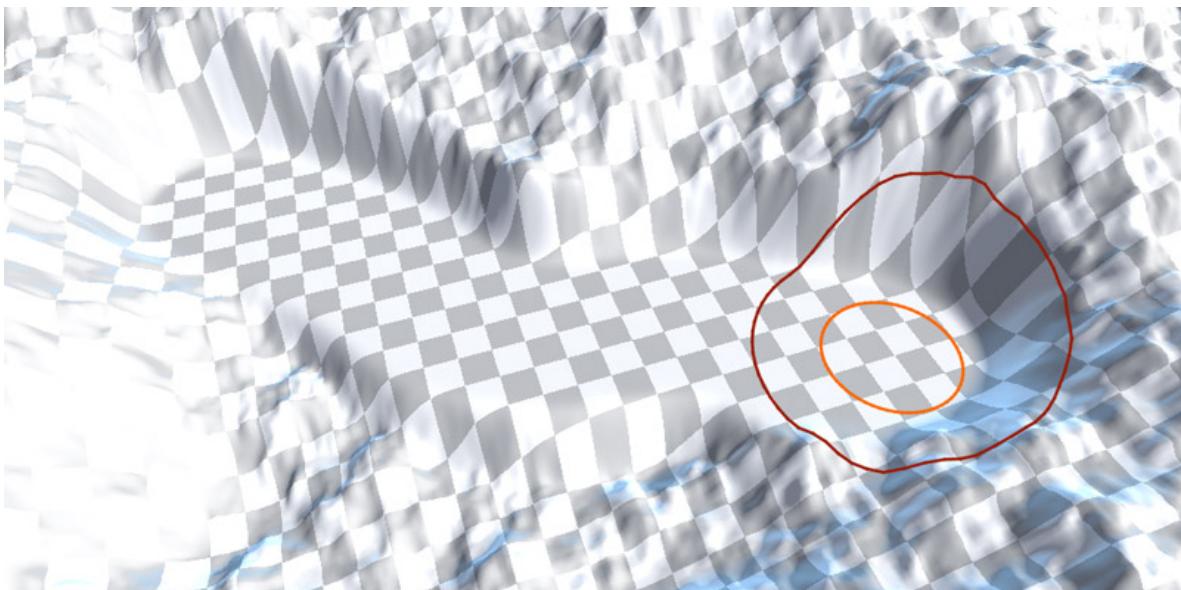


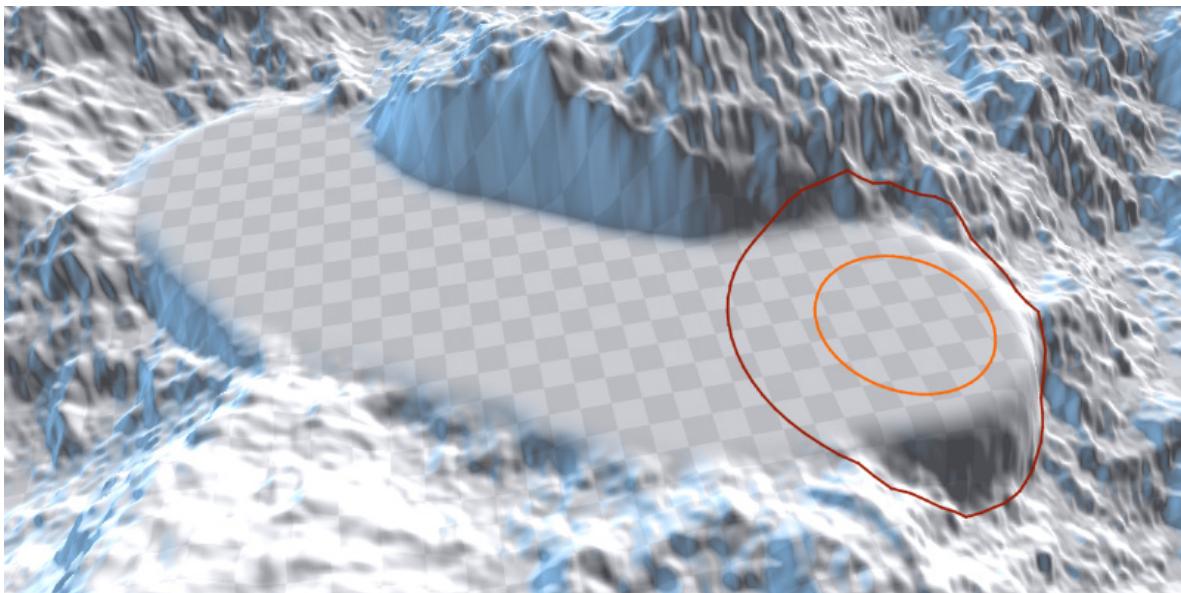
Adds detail objects with a given texture on a specific terrain layer.

With Shift pressed removes grass with this texture.

- **GrassTex:** the texture of grass. If this grass isn't ever added to terrain will create a default grass layer with this texture. All of the grass layers parameters are set up using the standard terrain editor.
- **GrassLayer:** terrain layer to paint grass on

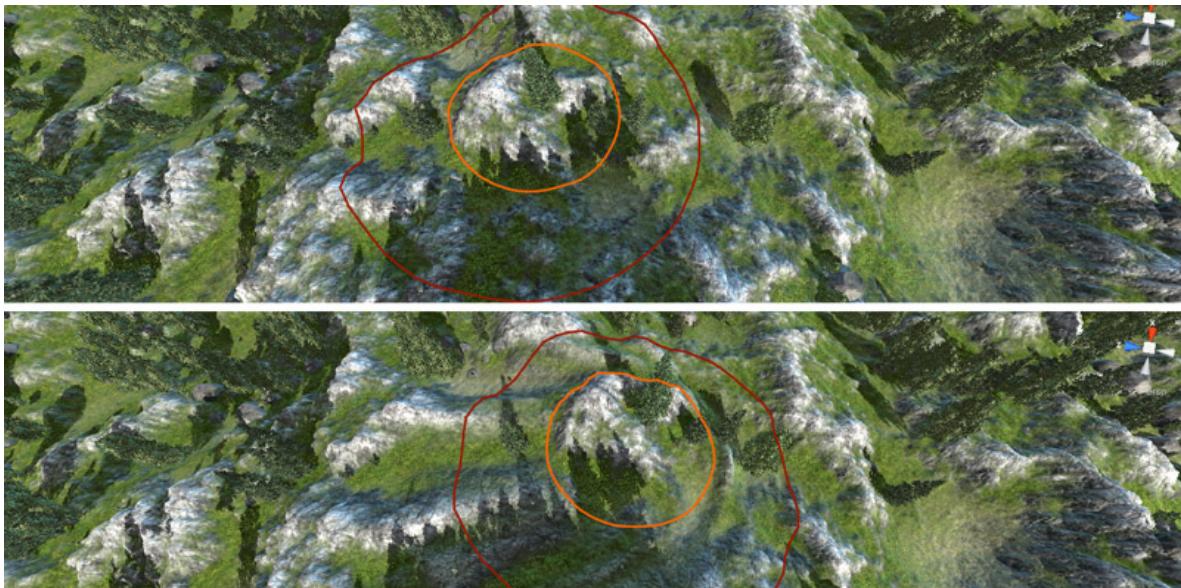
Level





Terrain level is picked with a Ctrl button - it is assigned in graph as a CapturedPosition. Multiplying it with TerrainHeight will give a map value (range 0-1) of planar land.

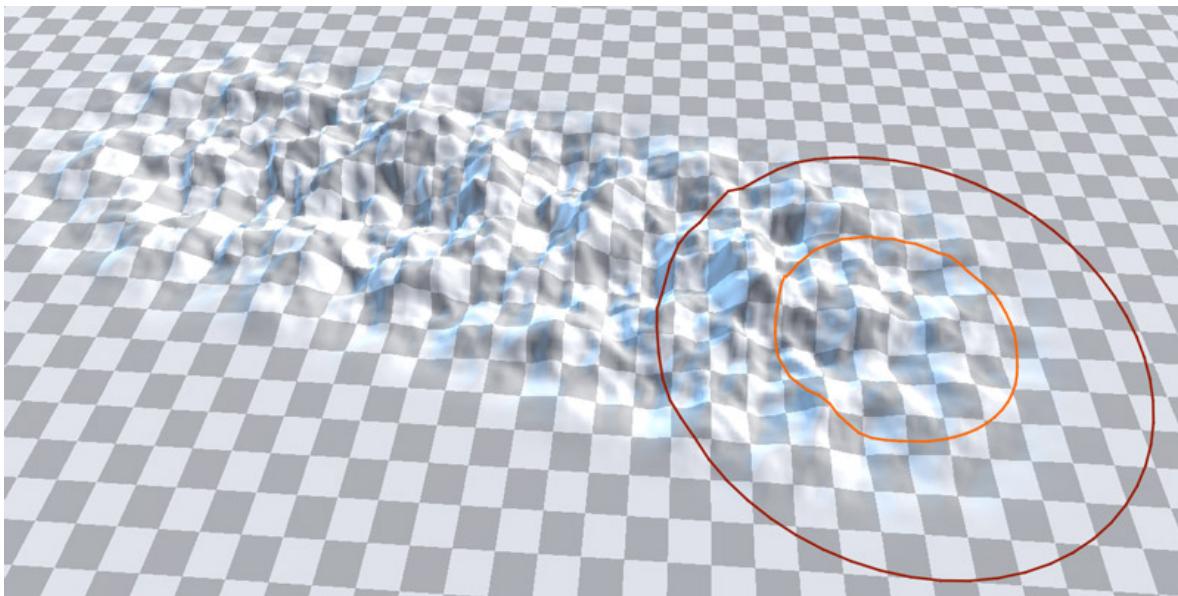
Move



Will offset height and land textures, and move all of the grass, trees and objects. Useful for tuning POI location positions.

Requires an Objects module.

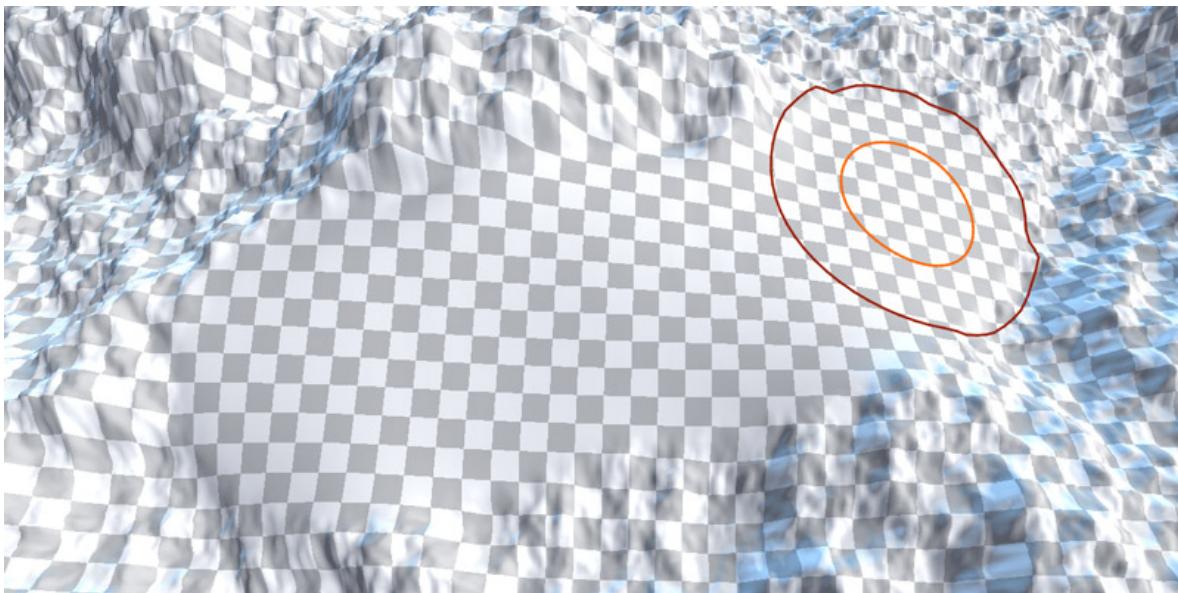
Noise



Applies noise to terrain height. Will not only raise terrain, but lower it depending on the noise map value.

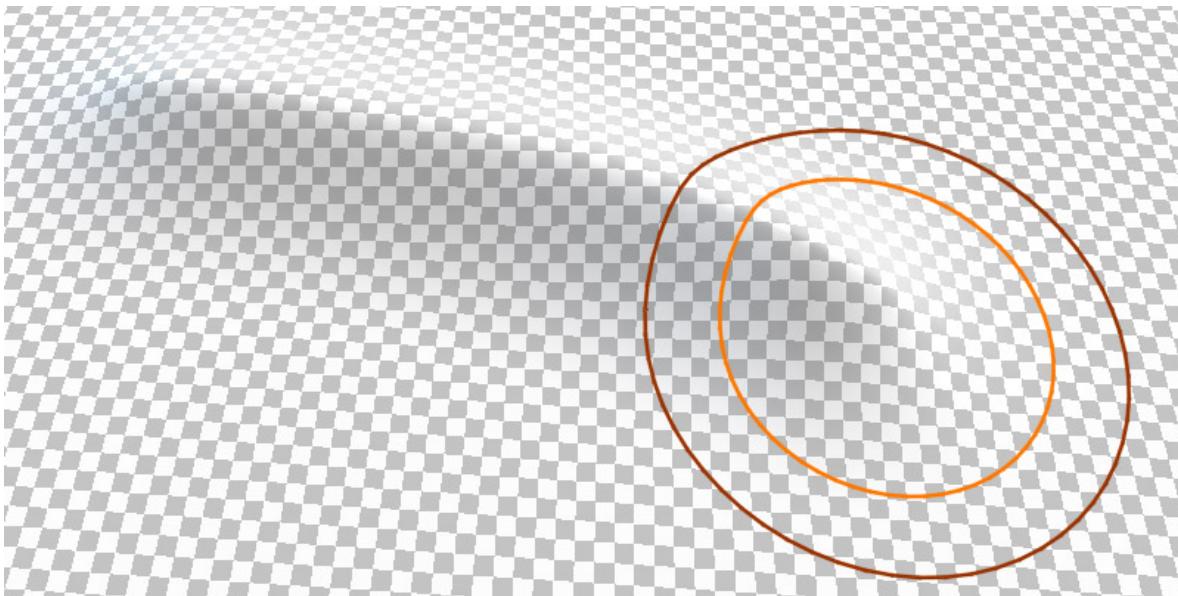
- **Intensity:** the amount of terrain raised or lowered
- **Size:** noise size pattern. The higher the value the more fractals are used, and the bigger final fractal size.

Polish



Uses the blur graph, but adds excessive amount of blur, making land within brush radius nearly planar (but still inclined).

Ridge

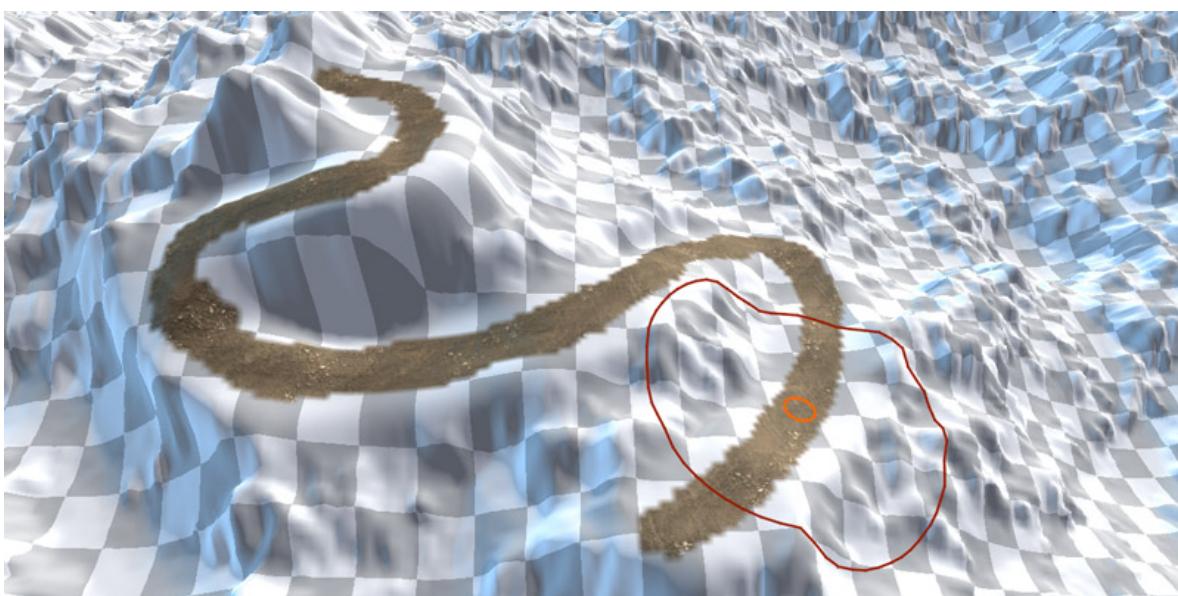


Raises terrain in a form of line. Useful for making mountain ridges. In inverted mode lowers terrain, which is useful for making streams or rivers.

Requires Splines module.

- **Intensity:** amount of terrain added per stamp

Road

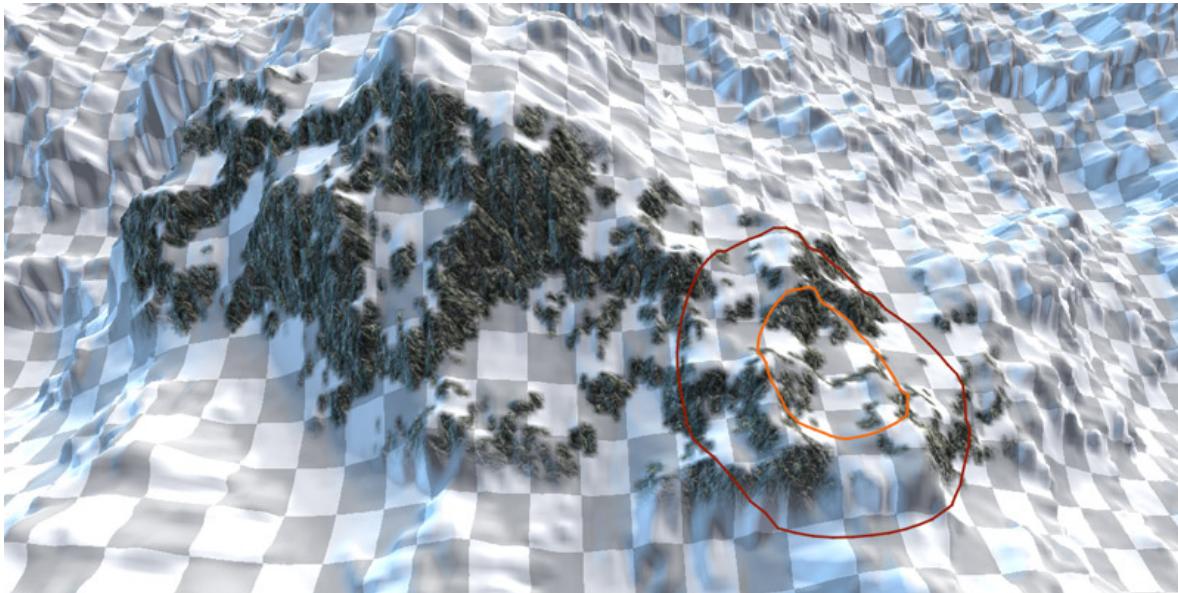


Flatters the terrain perpendicularly to brush direction and paints road texture along the brush direction.

Requires Splines module.

- **RoadLayer:** terrain layer to paint the road

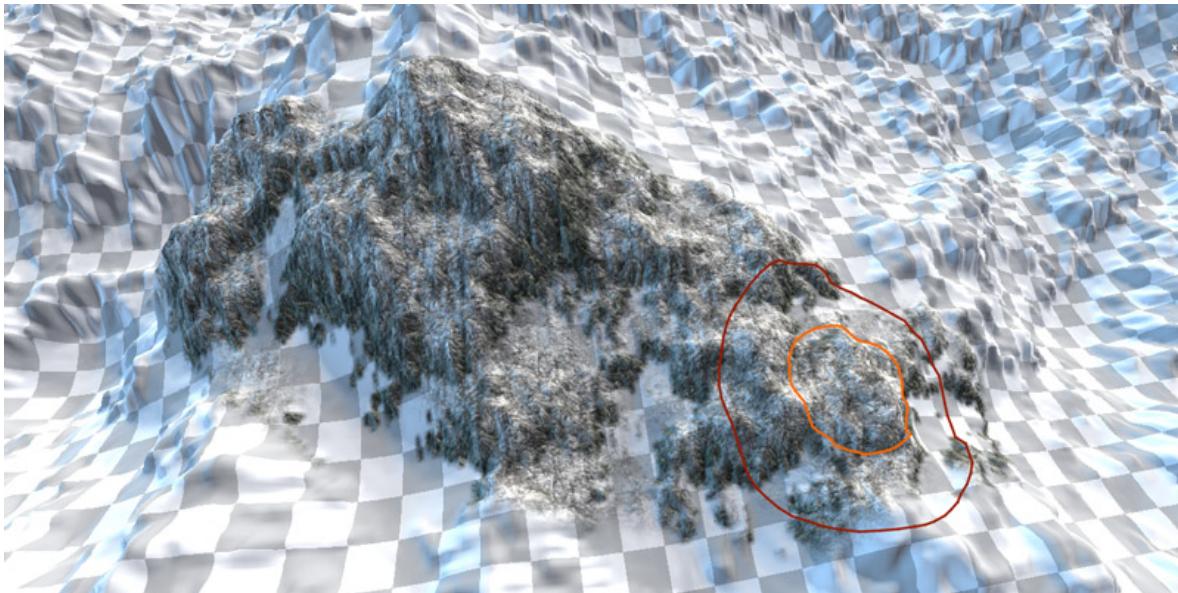
Slope



Paints with specified land texture (terrain layer) on inclined surfaces only.

- **Incline:** the minimum incline angle to paint, degrees
- **Smooth:** the transition between the land treated as inclined and non-inclined, in degrees. It is not related with blur.

SlopeCavity



Applies Slope and Cavity at the same time. Useful for quick painting of vast terrains.

Map Sets

MM Brush introduces the new way to work with texture or detail layers - Map Sets. This novelty is not used in vanilla MapMagic - at least in 2.1.0 - and maybe will be added later if users will find it useful.

MapSet nodes operate on multiple maps at once. Each MapSet contains several maps, each with a prototype tag (it's mostly texture). Consider maps set as a `<Texture, Map>` dictionary, or, to simplify things, just an array of maps.

This can give far better control on importing and exporting maps.

Import/Export

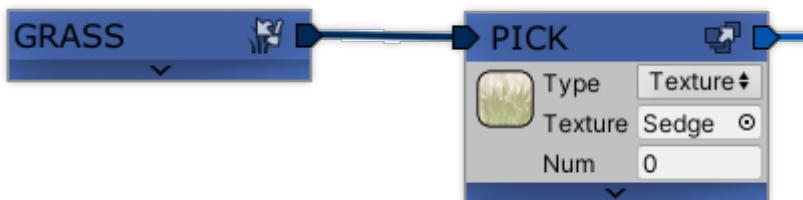
Brush > Input > Texture Set will import the whole set of texture control maps (masks) used on terrain. And Brush > Output > Texture Set will write the whole set of masks to terrain as they are, without combining or normalizing anything. These operations are done with the special nodes.

The same for the grass textures: Brush > Input > Grass Set will import all of the grass masks, while Brush > Output > Grass Set will write all the masks. This way this graph will read and write all the grass without changing anything:



Reads and writes all set of grass masks

If we want to pick some map from the set we've got to use MapSet > Pick node. In this node we can select the map to pick by its prototype texture. This example brush graph will read the grass masks sets and then select one mask from it, the one that is using the Sedge texture, and then use it in some other nodes:

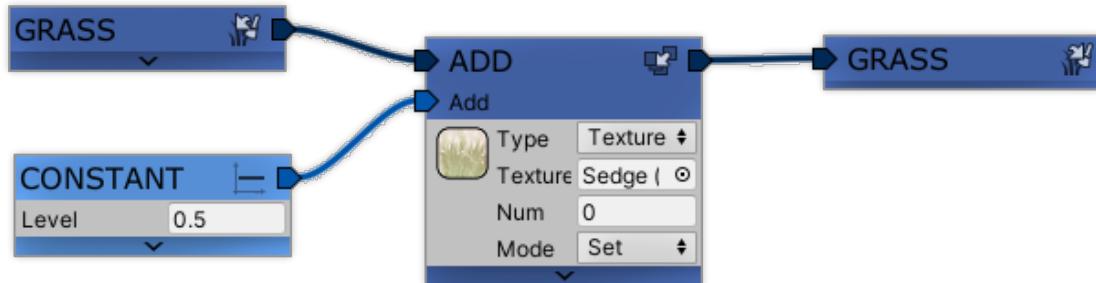


Chooses a mask of a Sedge grass

If the terrain has several different grass prototypes with the same texture we can clarify what map exactly to use with the Num value - it will force the Pick node to use the defined instance number. This number is zero-based: for the value 0 it will read first occurrence of this texture, for the value 1 it will take the second one, etc. This is the equivalent of picking the array element in code.

To pick mask by grass/detail mesh instead of texture (like grass bushes, small stones or other Unity details) switch Type to Detail.

To set a map in a set use Add node (named Set in beta). This example will read the grass set, and assign it's Sedge density level to 0.5:



Reads all grass masks, sets Sedge to 0.5, and writes all masks back

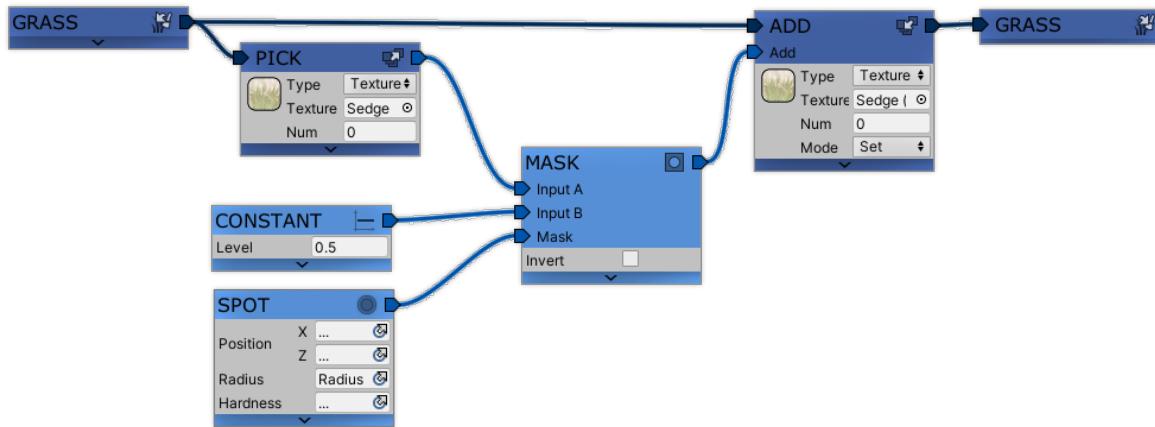
In this case the final Sedge density on all the brush square will be 0.5, and it doesn't really matter what it's original density it was - 0, 1 or 10, it will always set the half-density. Even if this type of grass was not used on terrain it will create and assign a new grass prototype with Sedge texture.

If you switch the Mode to Add it will apply the grass additively: will read the grass, append the value 0.5 to Sedge, and output the grass.

And in AddNormalize mode it will lower all of the other channels so that the final sum of all channels is 1. It won't change the intensity of the Sedge map, will lower only other channels. It's not so relevant for grass, but vital for terrain control maps, whose sum should always be 1.

Modifying

We can already paint with this graph, but it's not convenient to paint with squares. This graph looks more like a working brush: it reads all grass maps, picks Sedge map, does some operation with the Mask node, sets the Sedge map in the original grass MapSet, and writes this modified sets of maps.

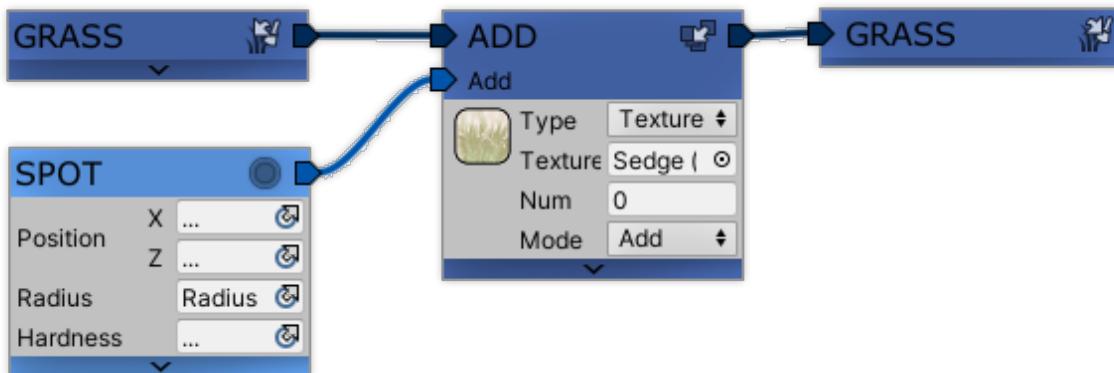


Reading, picking, modifying, setting, writing

The Mask node actually does the following: it takes original Sedge map as Input A, constant 0.5 map as Input B, and mix them based on the brush stamp circle from the Spot node. In areas where the Spot node gives 0 it's using InputA (original sedge), in areas where the Spot gives 1 (i.e. brush center) it's using InputB (constant 0.5).

Note that final grass density is equal to Constant value. If Constant value is 0 it will remove Sedge grass with the brush stroke.

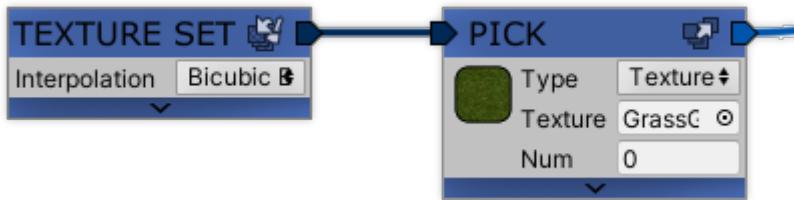
Just in case here is a simplified version of the brush graph. It has one key difference from the graph above - it is setting the map to set in additive mode. This means that it can't reduce the grass density, only add it.



Additive brush

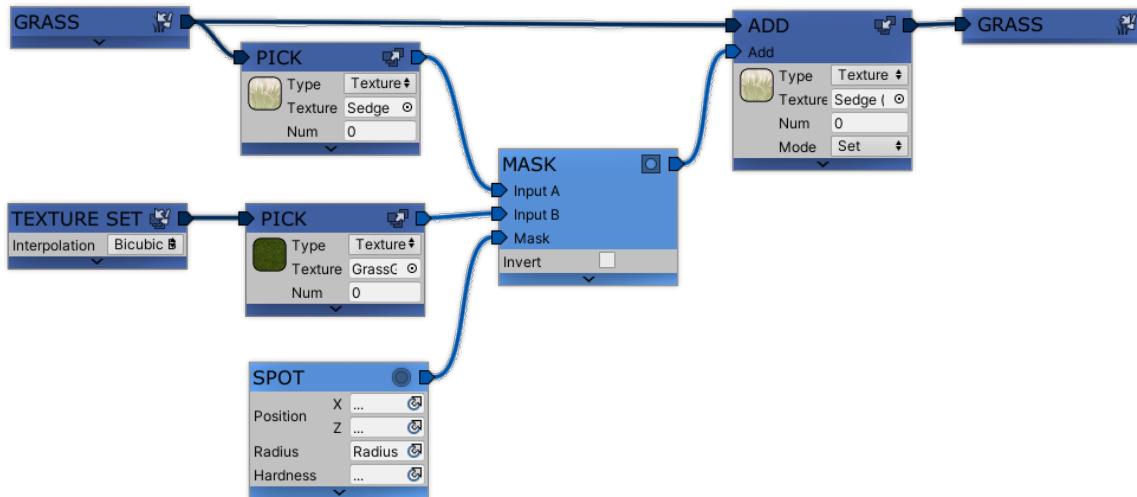
Masking with terrain maps

We can pick the terrain control map as well by using Brush > Input > Texture Set instead of a Grass Set. This combination will pick the mask where the GreenGrass texture is used:



Reading terrain texture

We can use it as a mask to plant grass only on GreenGrass, ignoring cliffs or dirt textures. Here is the grass brush graph (non-simplified) with this mask added:

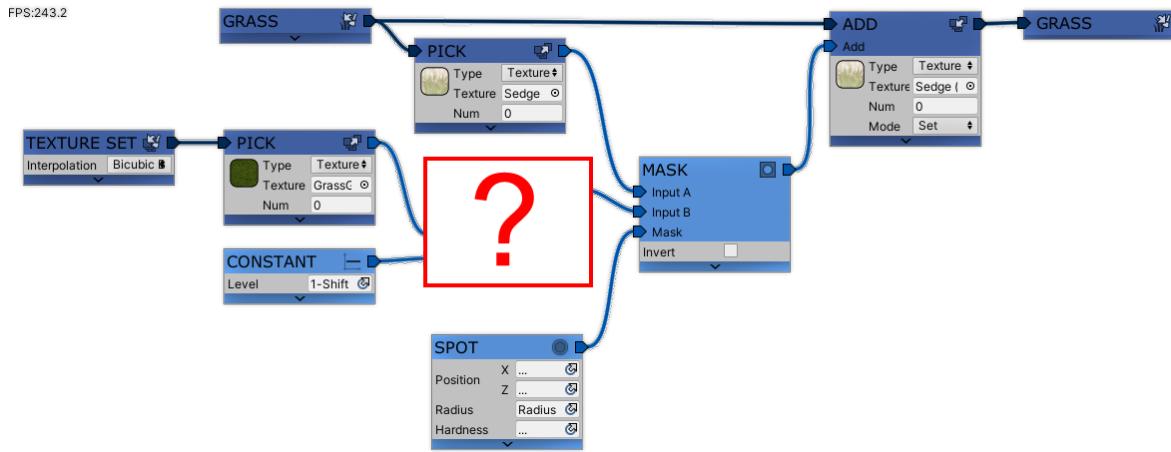


The brush that is drawing grass only on the GreenGrass texture

Erasing on Shift

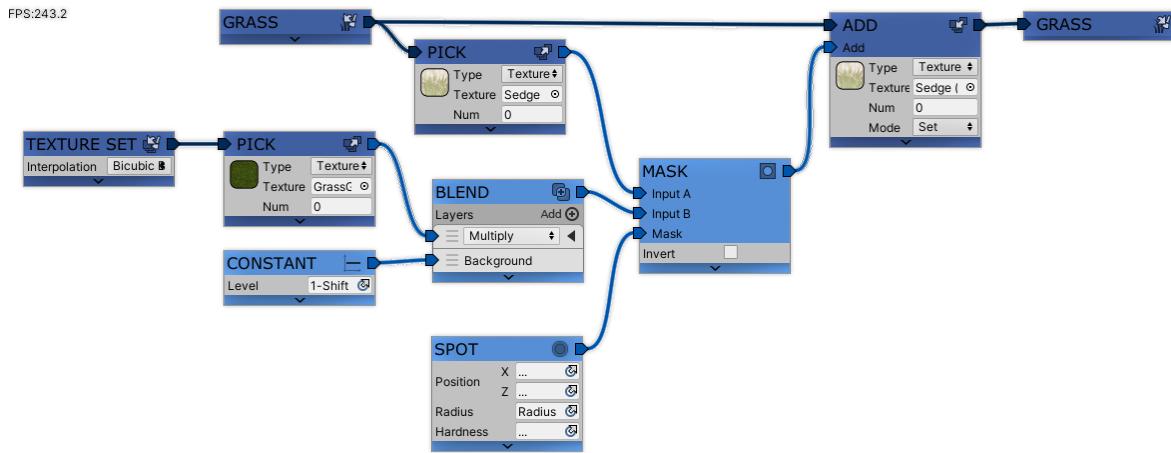
This grass brush is pretty useful, but we can improve it a bit more. It would be handy if this brush could remove the grass if Shift button is pressed.

The final grass density is driven by the Input B mask (where the Constant was connected). So if we want to remove the grass we've got to send 0 to Input B, if we've got to add grass at terrain GreenGrass - we've got to send GreenGrass to it. How to make the Constant node that switches its output to 0 on pressing Shift is shown in a previous tutorial. Now we've got to combine this Constant node with the GreenGrass mask:



We should switch between GreenGrass mask and 0 on pressing Shift

So, if Shift is not pressed, the Constant value is 1. If it's pressed, then 0. This is similar to logical AND switch, which could be implemented by multiplying:



Final Brush

This way we can paint the grass with this brush, and the grass will grow only on GreenGrass terrain texture. And we can erase the painted grass with Shift as well without affecting other grass types.

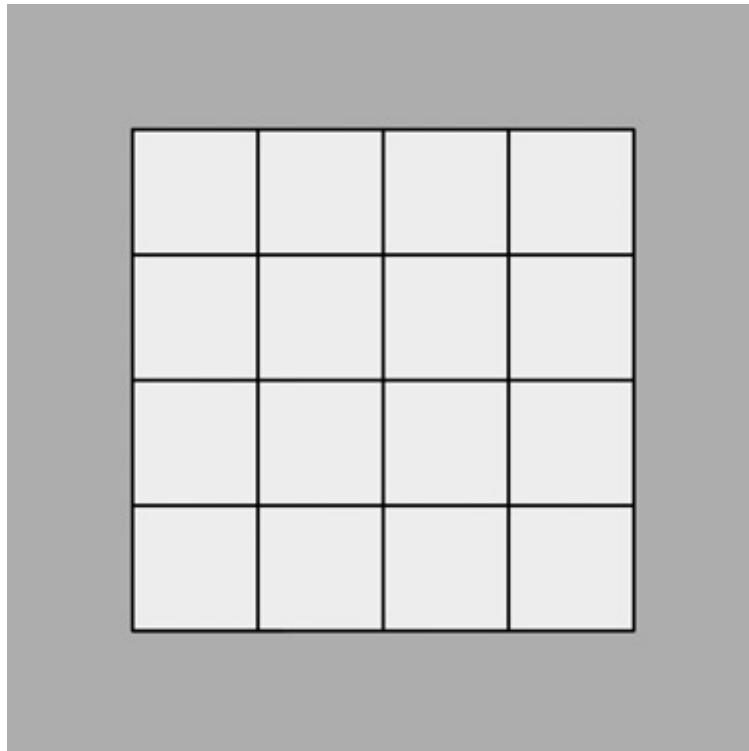
This is the practical example of using the MapSets for brush graph.

Resolution Compliance

When painting standard (non-MapMagic) Unity terrain layers with the standard tools you might find that terrain layers control maps do not correspond perfectly to the terrain wireframe driven by heightmap. Sometimes control map pixel located exactly where terrain vertex is, but in most cases it has some offset. This offset varies from zero to half-pixel distance on each axis, and it's hard to predict what exactly this offset will be when painting terrain.

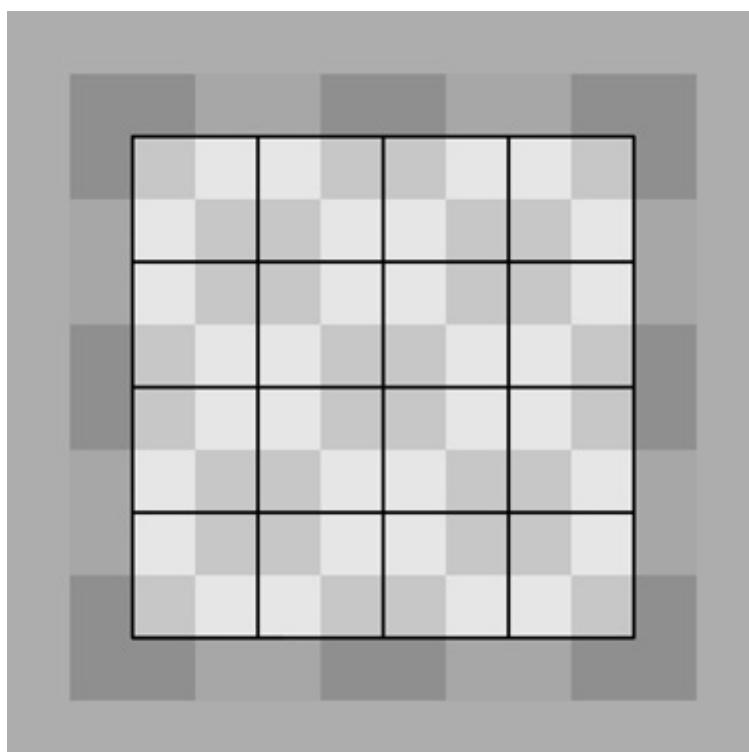
This is why it happens.

Let's imagine very small Unity terrain, made of $4 * 4$ polygons. However this terrain will have $5 * 5$ vertexes, since the vertexes are placed between the polygons, and at the start and end:



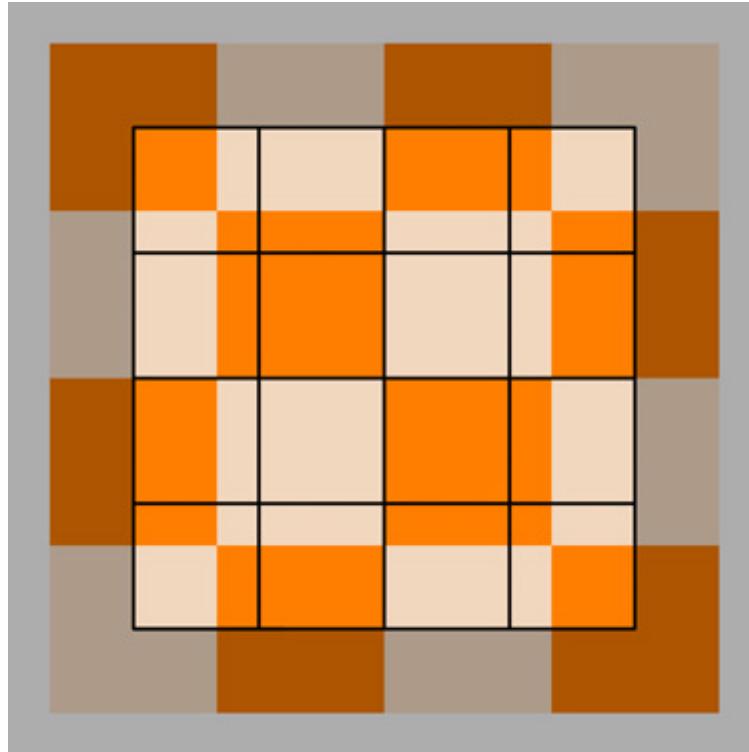
Whatever the polygons count will be, the number of vertexes on each side will be $\text{count}+1$.

Terrain heightmap defines the height of each of the vertexes, not polygons between them. This way heightmap resolution for this terrain will be $5 * 5$:

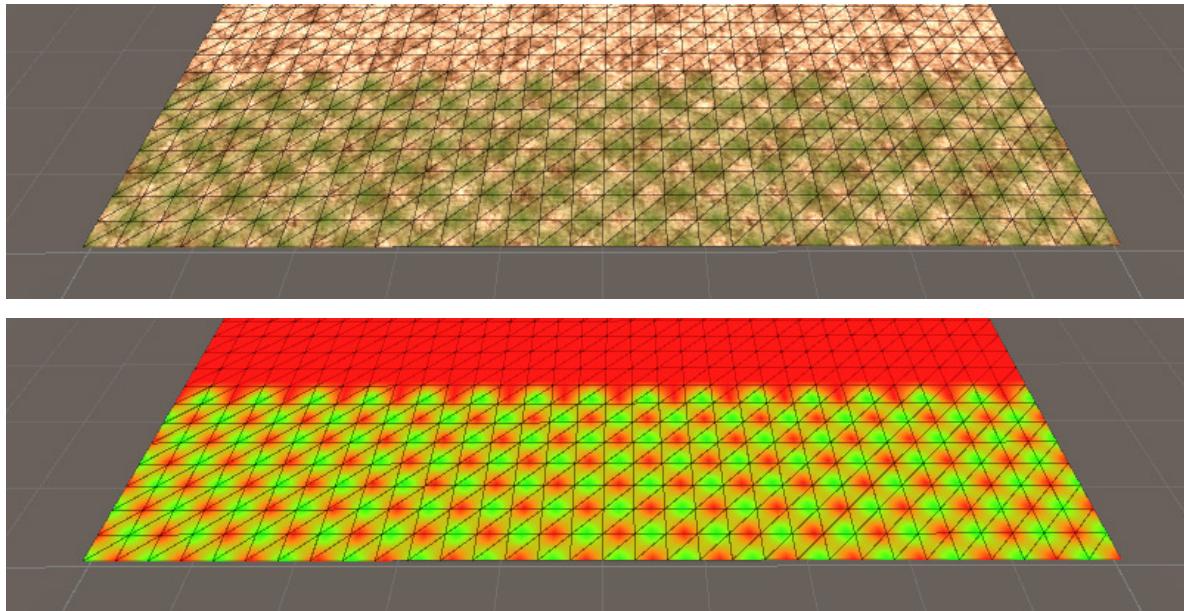


And that's why the usual terrain height has a weird resolution like 513 or 1025. These numbers suits well for terrains of 512 or 1024 polygons.

However the control maps resolution still remains power of two: 512 or 1024. Or 4 like in our case. This way it does not correspond to terrain vertexes at all. It all becomes even worse because when applying a control map, Unity is making half-pixel offset from each size (possibly to align it the same way heightmap aligned). This is how the control map is applied to terrain:

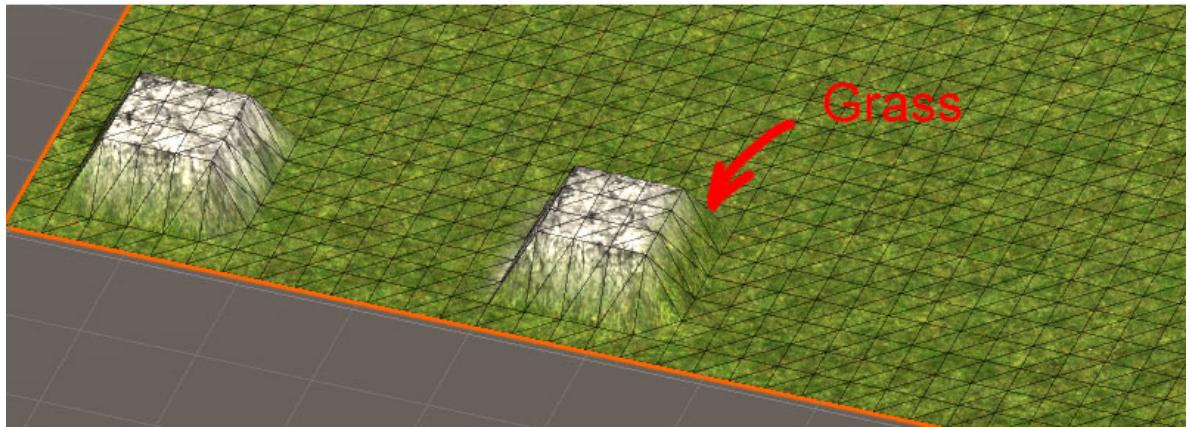
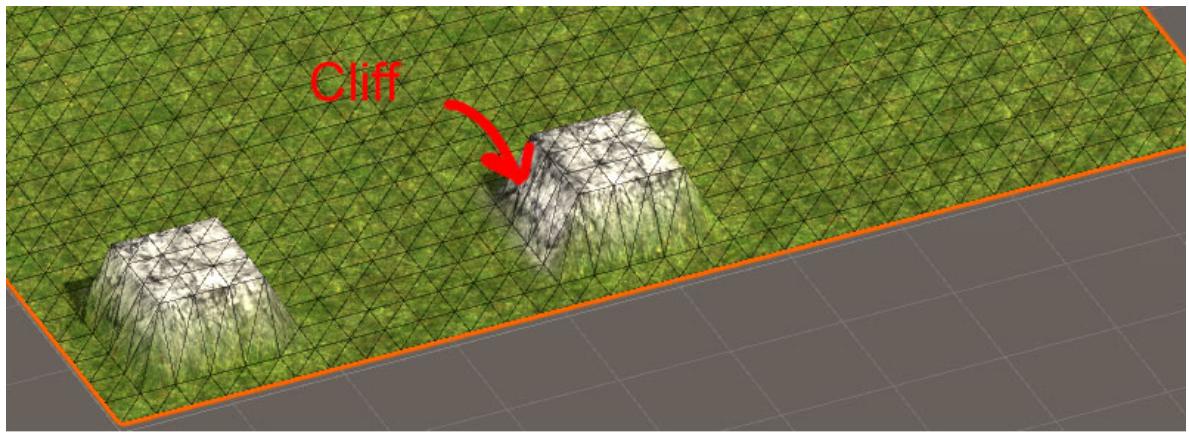


And this is how 32x32 control maps looks on 32x32 polygon (33x33 vertices) terrain:

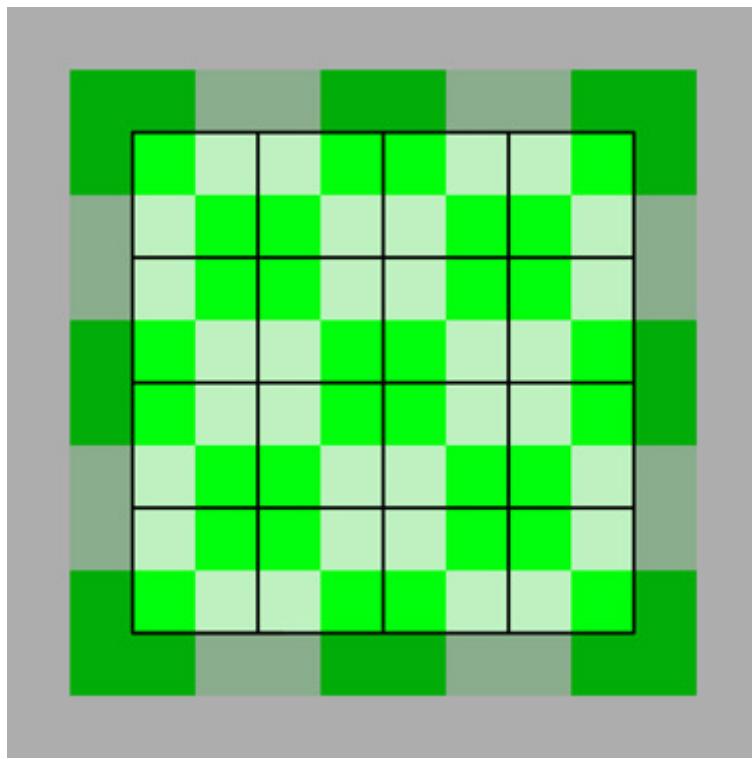


At the left it starts pretty fine, but in the center the control map pixel is placed right between the terrain vertexes, and at the right it makes a full circle and corresponds heightmap again.

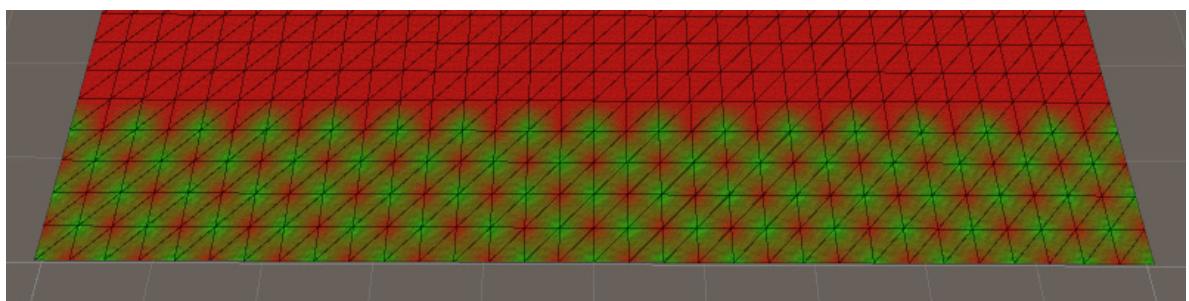
Here are the issues it brings. The cliff is applied the way it should at the left, near the terrain edge. But there is no way to do the same at the center - one side of the slope is filled with cliff, while the other is filled with grass:



However if the control map will match the heightmap resolution it will be displayed properly:



And the same correspondence on the real terrain:



Note each control map pixel is located exactly on terrain vertex position.

Still consistent if height resolution is 65 x 65, and control map resolution is 33 x 33:

