This module brings new type of object to the MapMagic graph - the Objects list. Allows smart placing of objects and terrain trees. Creating complete procedural world with no objects is almost impossible, so Object nodes are here to help with creating your infinite worlds.

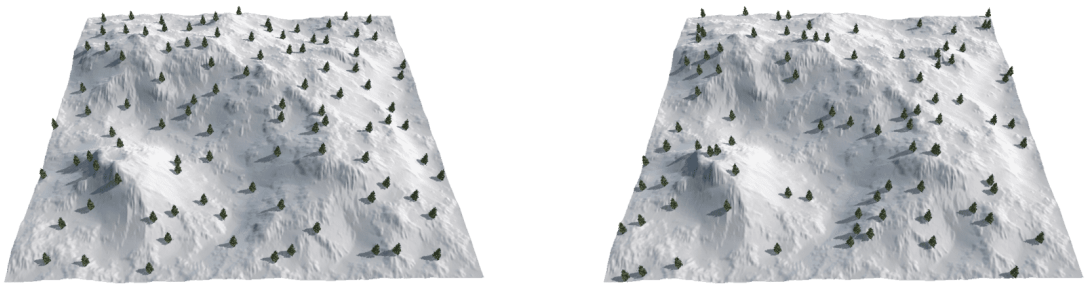
Initial Generators

Random

Using non-grid random to place objects. Objects are created using the probability factor of mask and remoteness from other objects.

Use Cases:

Usually it takes more time to place generate objects this way, so it's recommended to use Random node only if the desired effect could not be achieved with Scatter - for example, if scattered objects are still too uniform.



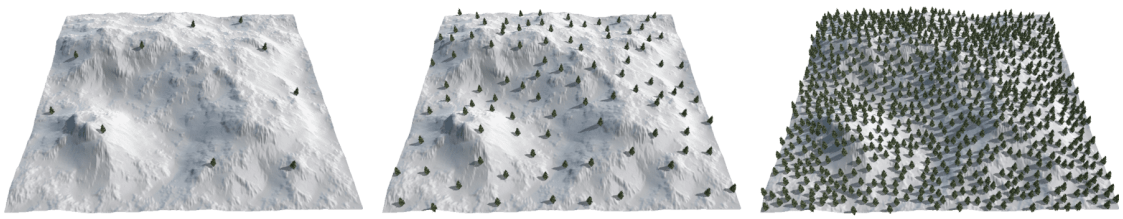
Zero Uniformity: Scatter (left) and Random (right)

Inputs:

Random node can receive the probability mask. Mask values factor will be taken into account when placing an object. The higher the pixel values in some of the map area - the more objects will be spawned there.

Properties:

- **Seed**: a number used to initialize pseudo-random noise generator. If two generators use equal seed numbers the resulting pattern will be the same.
- **Density**: the quantity of scattered objects per a square kilometer (100x100 units). Note that since are objects are placed at random within terrain tile + margins the total number of objects within tile itself might differ both in bigger and smaller sides.



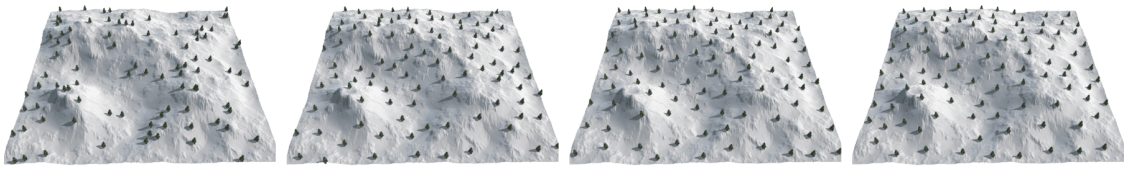
Density: 10, 100, 1000

Unlike Scatter, it will not create objects on any of the terrains if density is below 1.

- **Uniformity**: the influence of the distance factor in randomly placing the object. If 0 it will place objects absolutely at random, disregarding already placed ones. In 1 it will try to find the equidistant place from already existing objects.

  Note that the true "grid" uniformity, like the one in Scatter, is not achievable here.

  Bigger uniformity values slow down performance.



Uniformity: 0, 0.1, 0.5, 1

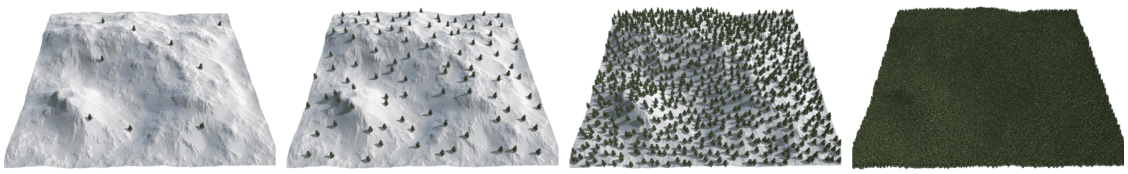
Scatter

Scatters objects in the terrain area, randomly creating new object positions. Usually it's the first generator that starts the objects nodes chain.

Unlike Random generator, it virtually places objects in an ordered grid, and then offsets their positions at random. It is much faster than the Random node and mostly suits the most cases. However Random still might be needed for some "true random" scattering.

Properties:

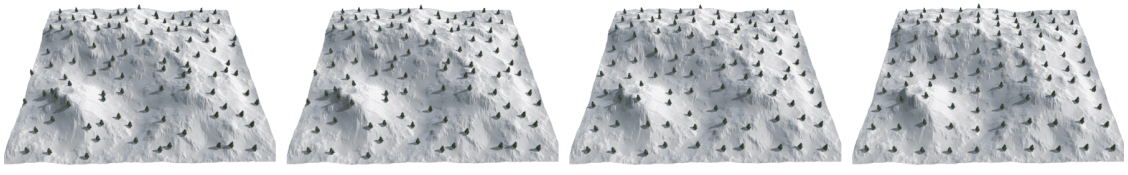
- **Seed**: a number used to initialize pseudo-random noise generator. If two generators use equal seed numbers the resulting pattern will be the same.
- **Density**: the quantity of scattered objects per a square kilometer (100x100 units). Note that since are objects are placed at random within terrain tile + margins the total number of objects within tile itself might differ both in bigger and smaller sides.



Density: 10, 100, 1000, 10000

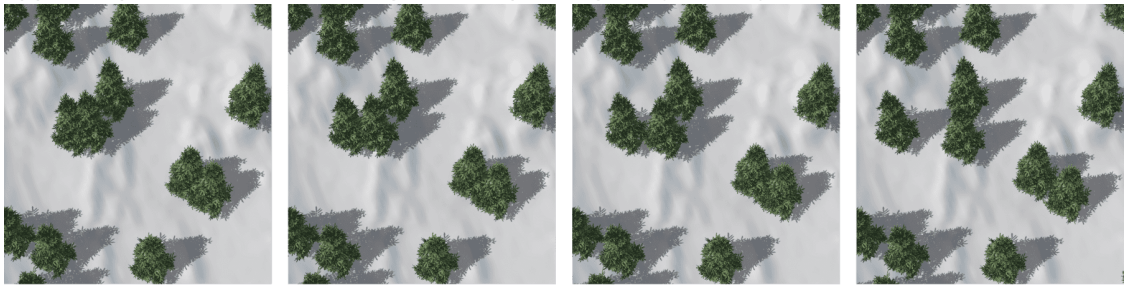
If the Density value is below 1 (presuming terrain size 1000) it will scatter objects on some of the terrains only, leaving others empty.

- **Uniformity**: how close the objects stay after being "virtually scattered" in grid. If the value is 0 the object could be placed anywhere within it's "grid cell", if 1 it's not moved and pinned in "cell center".



Uniformity: 0, 0.1, 0.5, 1

- **Relax**: in some cases, especially when the uniformity value is low, the objects could be scattered too close to each other - for instance, when one object moves (1,0) from it's cell center, and the one to the right (-1,0). So after the objects being scattered the generator applies a positions smoothen algorithms, pushing away the objects that are too close.



Relax: 0, 0.5, 2, 4

Modifiers

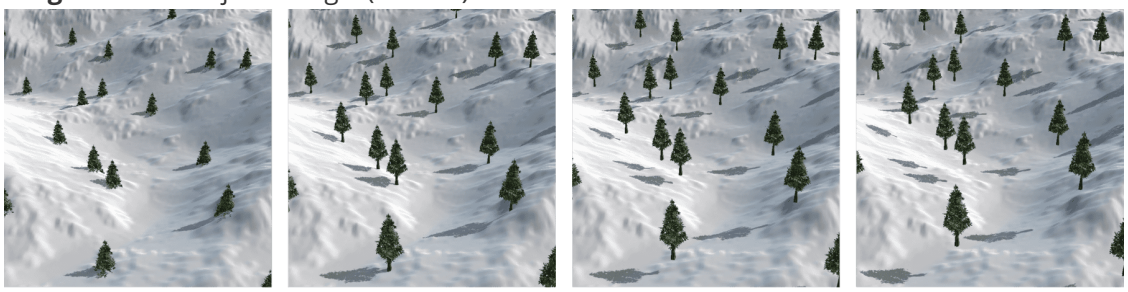
Adjust

Scatters objects in the terrain area, randomly creating new object positions. Usually it's the first generator that starts the objects nodes chain.

Unlike Random generator, it virtually places objects in an ordered grid, and then offsets their positions at random. It is much faster than the Random node and mostly suits the most cases. However Random still might be needed for some "true random" scattering.

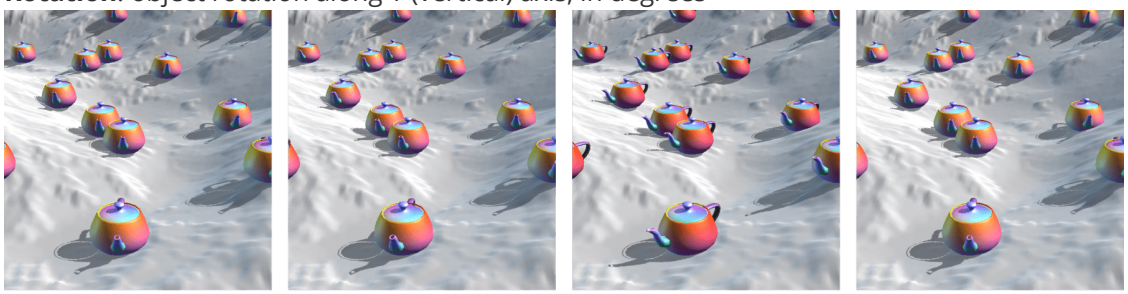
Properties:

- **Seed**: a number used to initialize pseudo-random noise generator. If two generators use equal seed numbers the resulting pattern will be the same.
- **Height**: moves object along Y (vertical) axis. The value is in world units.



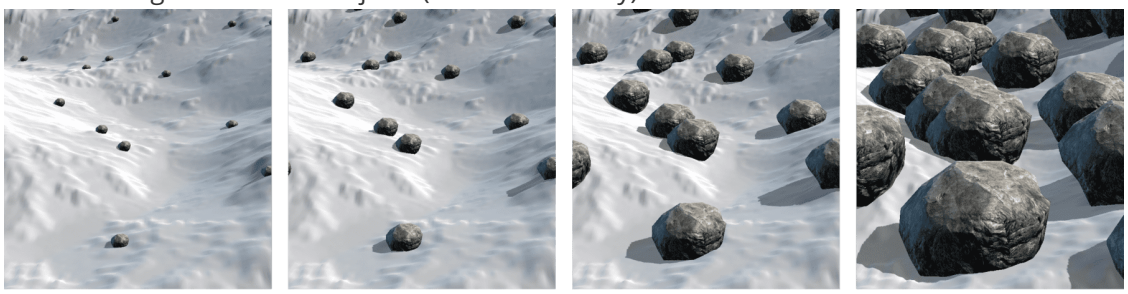
Height: -10, 0, 10, 20

- **Rotation**: object rotation along Y (vertical) axis, in degrees



Rotation: 0, 10, 45, 360

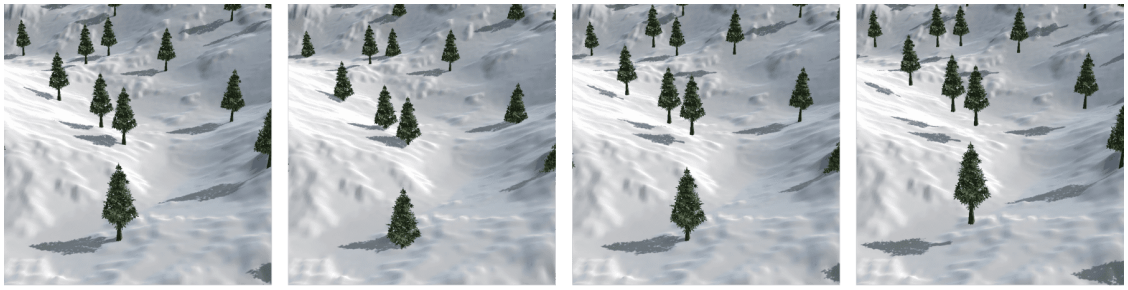
- **Scale**: changes the size of objects (3 axes uniformly)



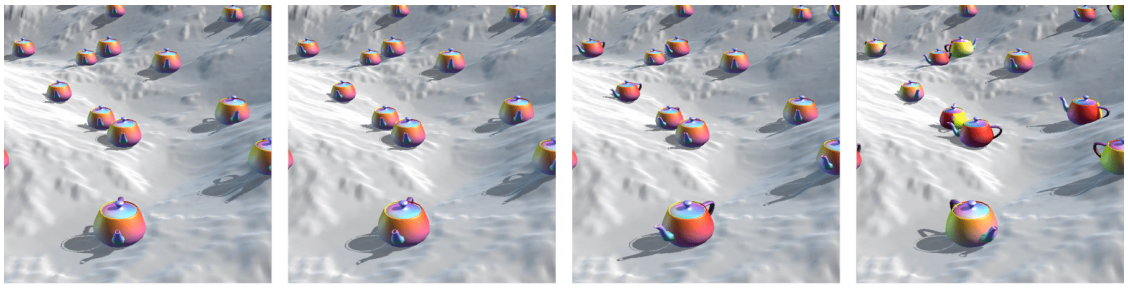
Scale: 0.5, 1, 2, 4

- **Random Range**: switches generator interface to range mode - instead of using the same value for all the objects, the value is selected for each at random, using the defined range minimum and maximum.

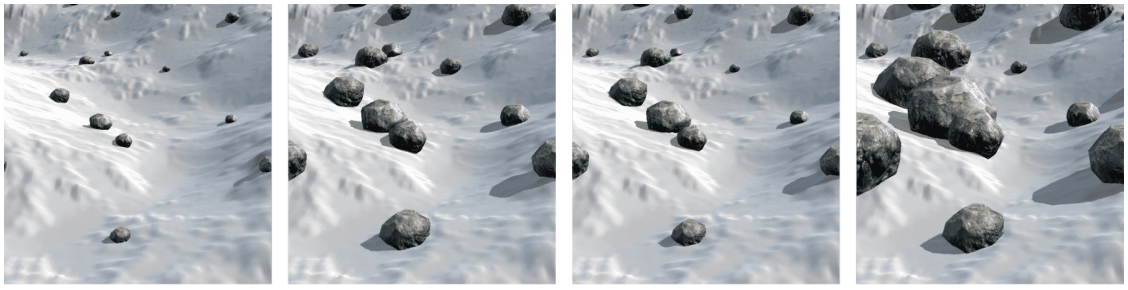
  With range mode enabled the node adjustments will look like:



Height Range: 0 ??? 0, -10 ??? 10, -10 ??? 20, 0 ??? 30



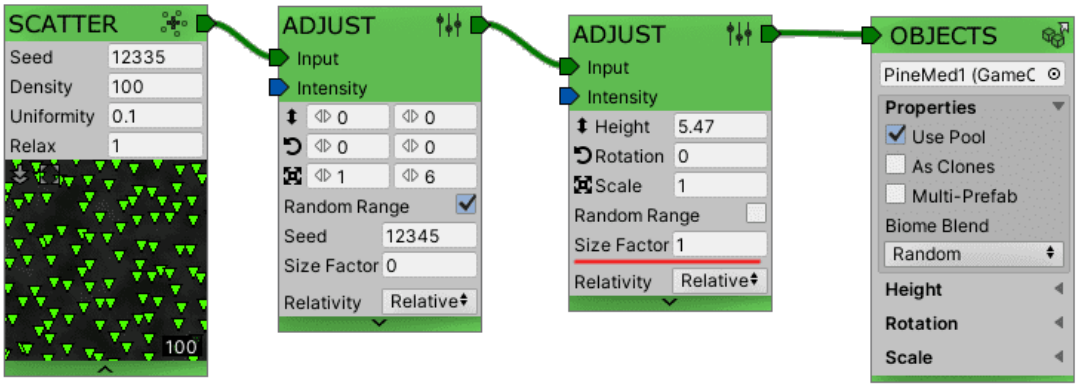
Rotation Range: 0 ??? 0, 0 ???10, 0 ??? 45, 0 ??? 360

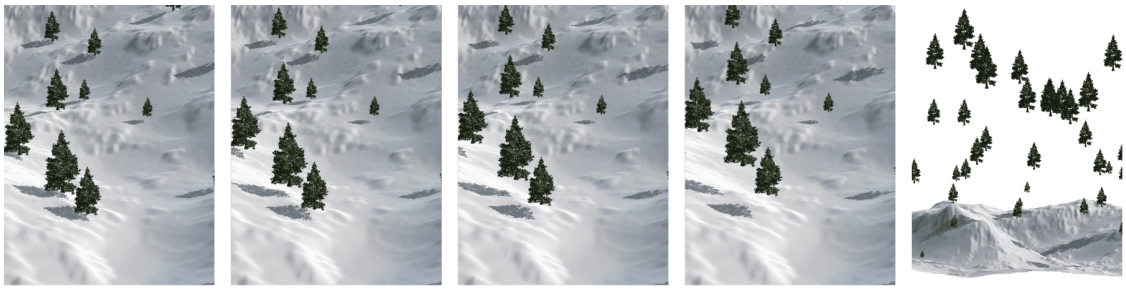


Scale Range: 0.5 ??? 1, 1 ??? 2, 0.5 ??? 2, 0.5 ??? 4

- **Seed** (when Random Range enabled): since random generator is used to select the range, seed value is exposed to initialize it. Different seed value will make the same objects take different range values. If two Adjust generators use equal seed numbers, ranges and inputs the resulting adjustment will be the same.
- **Size Factor**: multiplies the changes applied by this generator depending on the initial object size (from previous adjustments). For example, if the object???s size is 2 then all of the adjustment values will be doubled if the factor is equal to 1. This is useful for proportional object changes: if the object should be lowered to half of it???s height, for instance. When the value is 0 all of the objects are adjusted the same regardless of their size.

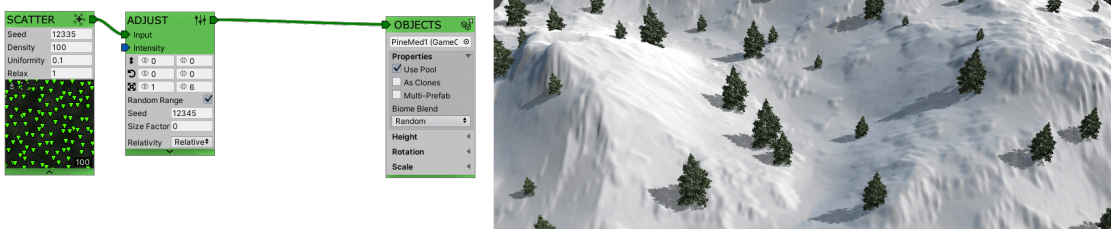
  In other words: the bigger the object's size - the bigger the adjustment.



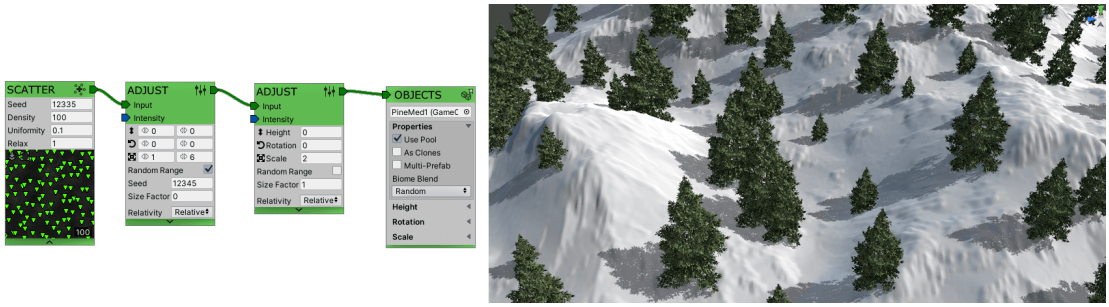


Size Factor: 0 (all objects are raised the same), 0.5, 1, 2, 10 (the bigger the object - the higher it is)

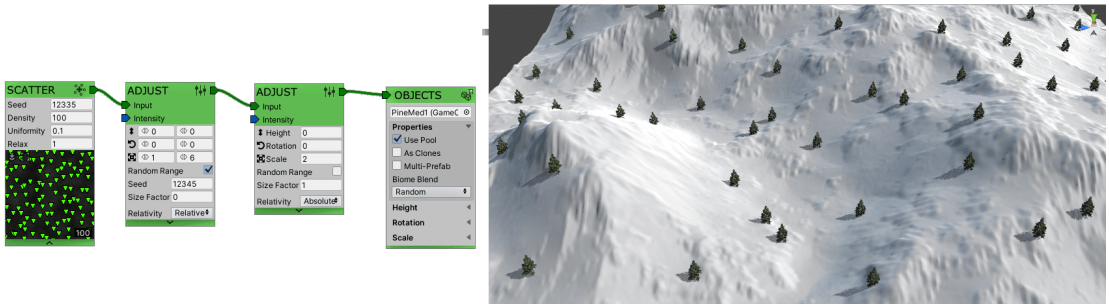
- **Relativity**: when the ???Relative??? type is selected all of the adjustment changes are applied to the existing object???s height, rotation and size. The ???Absolute??? type resets the old values and then places (and scales) the object as if it was placed on zero level, having no rotation and having a scale of 1.



Original graph with different object sizes



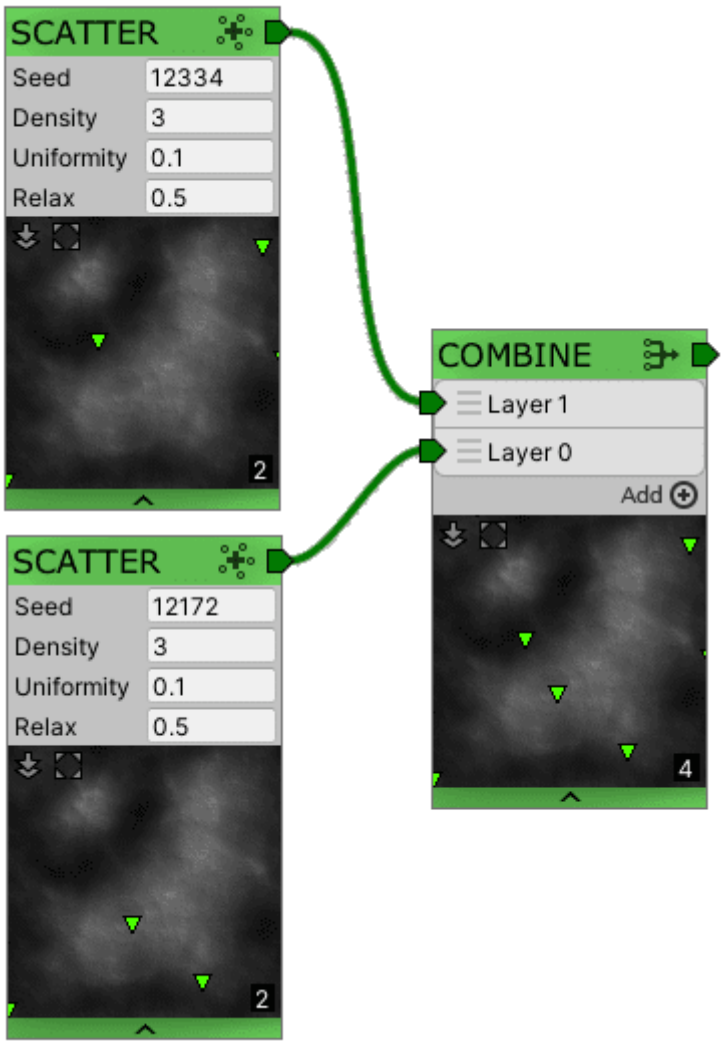
Relative mode: each object size is multiplied twice



Absolute mode: each object size is set to 2, no matter of it's previous size

Combine

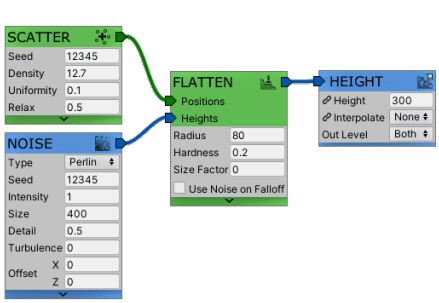
Unites several objects lists in one. The resulting objects list contains all of the objects from all of the source lists.

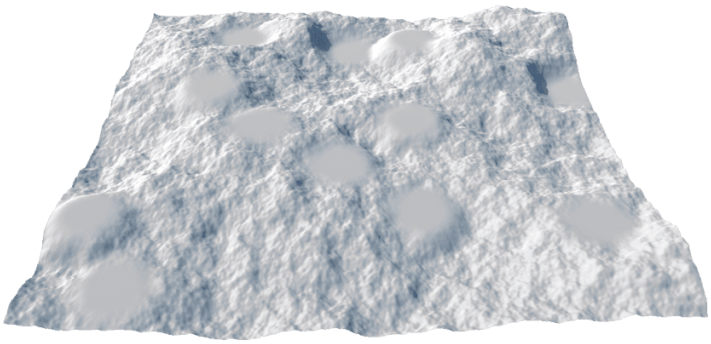


See also: Layers Guide.

Flatten

Creates flat land under objects to place houses or structures that require a flat land.



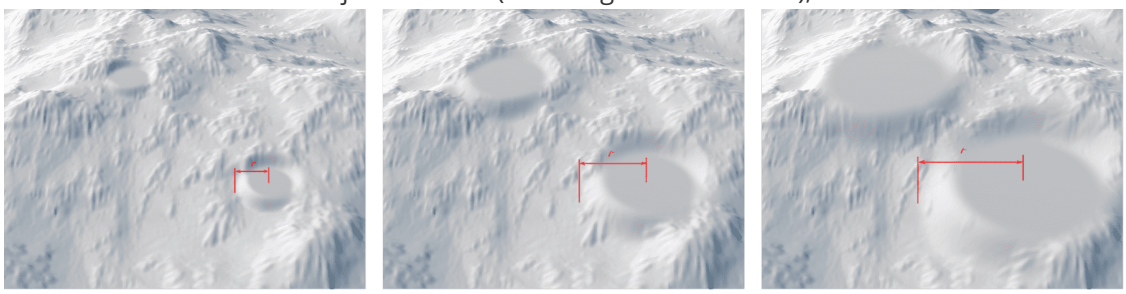


Inputs:

- **Positions**: objects that would be the centers of the flat areas.
- **Heights**: current graph heightmap that will be adjusted with this generator.

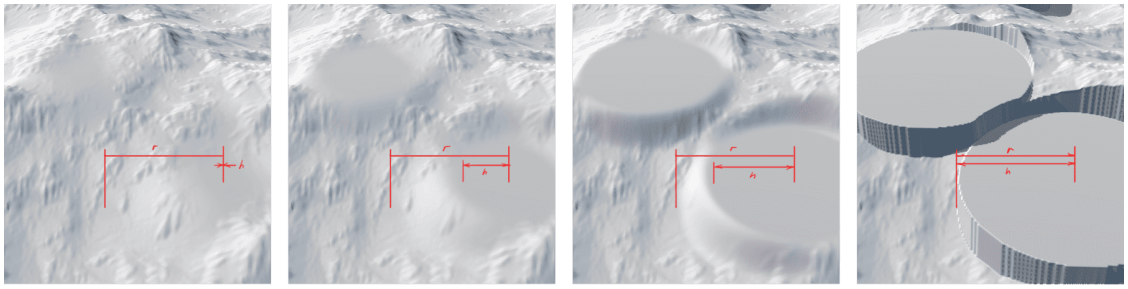
Properties:

- **Radius**: the radius of the adjusted areas (including the falloff area), in world units.



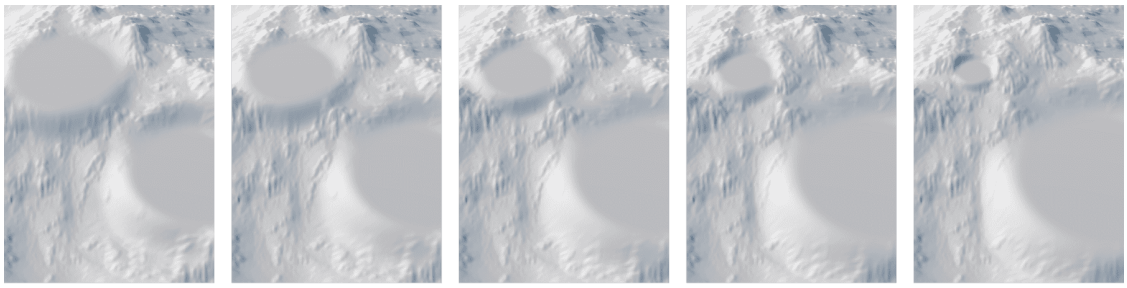
Radius: 10, 50, 100

- **Hardness**: the percentage of fully flat land radius compared to the total radius. Varies from 0 (no flat land) to 1 (all land within radius is flat).



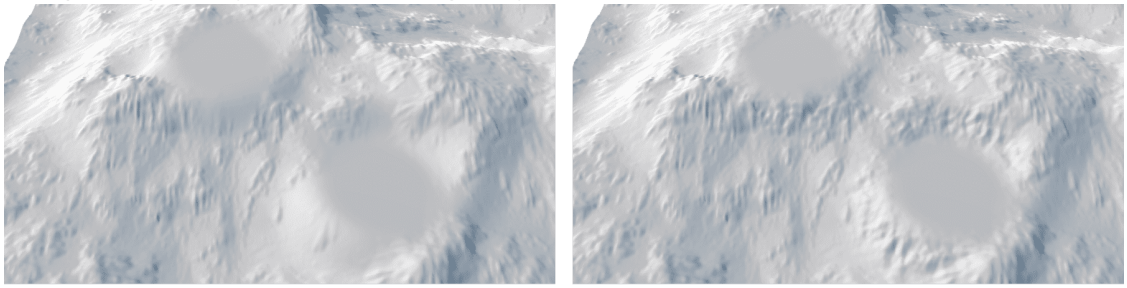
Hardness: 0, 0.333, 0.666, 1

Size Factor: multiplies the changes applied by this generator depending on the initial object size. For example, when factor=1 if the object???s size is 2 then all of the flatten stamp sizes will be doubled.



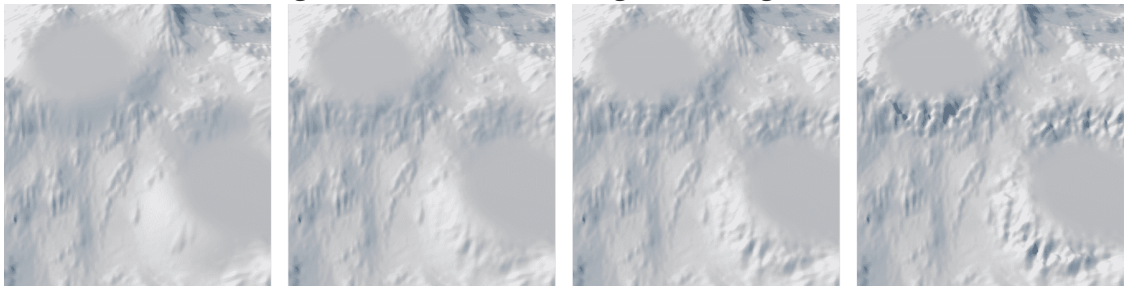
Size Factor: 0, 0.25, 0.5, 0.75, 1

- **Use Noise on Falloff**: applies noise in flatten stamp transition area. Used to make the stamp fit organically into surrounding noisy terrain.



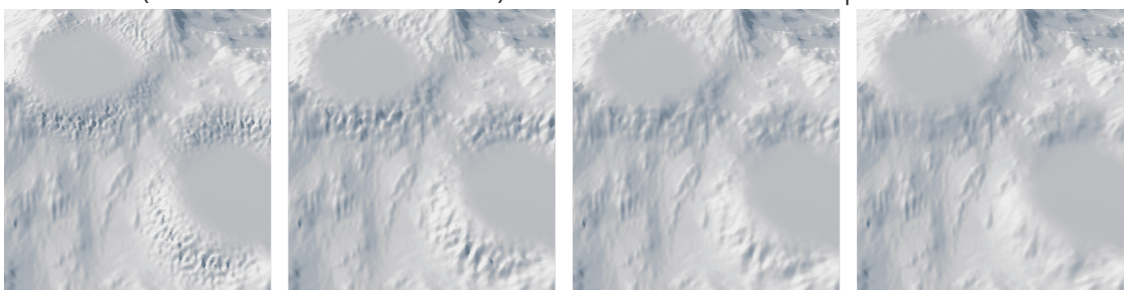
No noise (left) vs Use Noise (right)

- **Noise Amount** (when Use Noise is turned on): the amount of falloff noise applied. If the value is > 1 the noise might exceed the surrounding terrain height and look unnatural.



Noise Amount: 0, 0.5, 1, 2

- **Noise Size** (when Use Noise is turned on): the size of the falloff noise pattern.

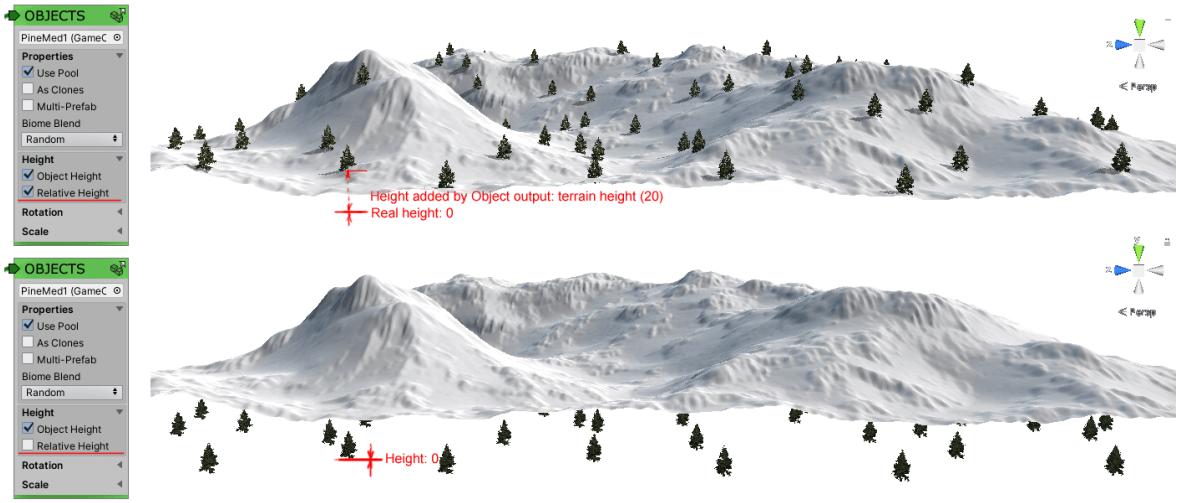


Noise Size: 2, 5, 10, 25

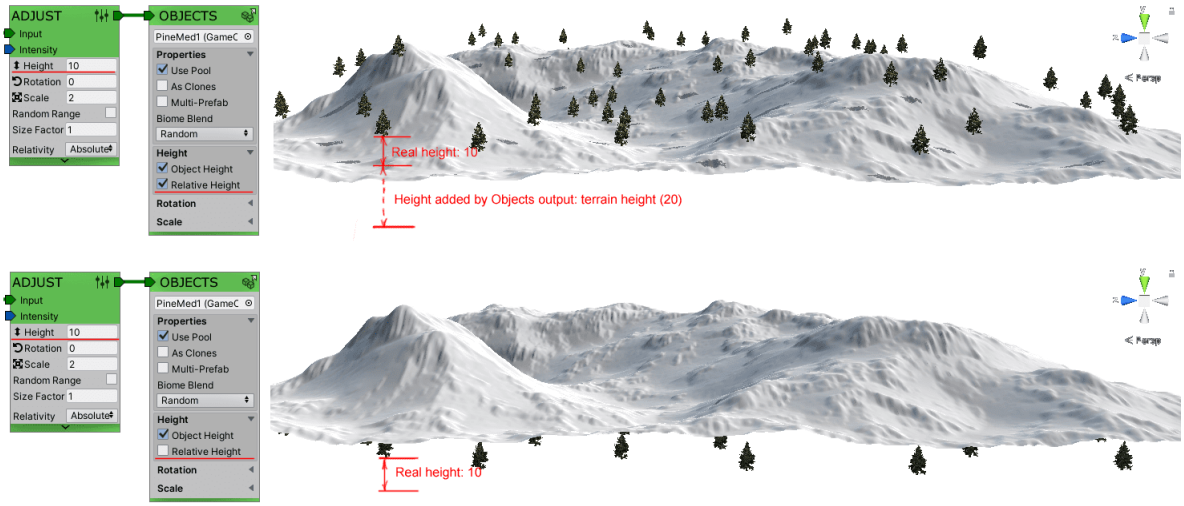
Floor

Places all of the objects positions at the heightmap level.

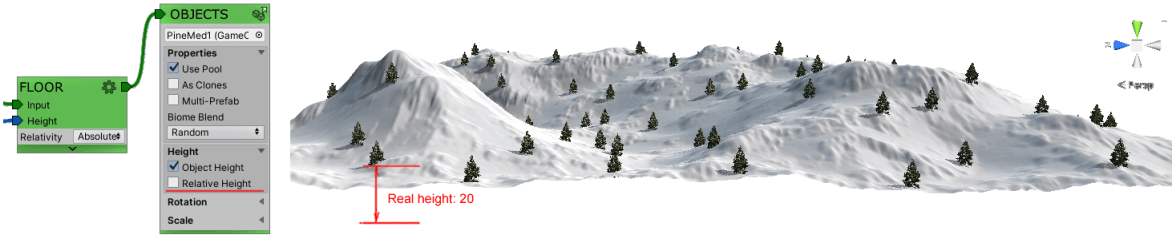
When working with Objects or Trees outputs, that have Relative Height feature it might seem that the objects are placed on the ground level by default. However, object nodes have no data about terrain height unless they are output with Objects/Trees output or floored with the Floor. Objects height is zero by default.



When the objects are adjusted with the Adjust node, for instance, when raised 10 units - all of the objects height is set to 10, not 10+terrain height.

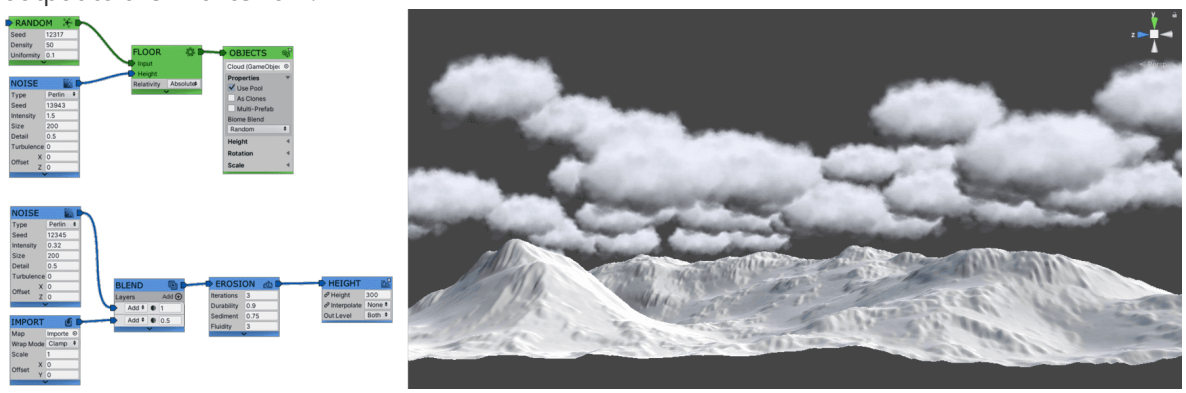


The Floor node make the real object height equal to the current heightmap provided.

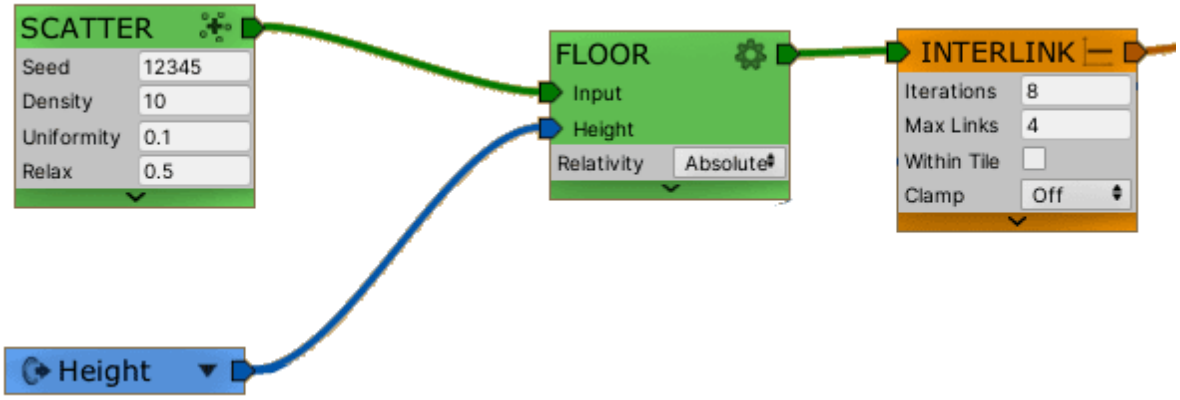


Use Cases:

Floor generator might be handy if you'd like to place objects with the height that is not related with the terrain. You can input any map as a node heightmap, including the one that won't be output to the final terrain:



Floor node should also be used on converting objects to splines with the Interlink node. Otherwise the resulting spline will be placed at the zero level instead of terrain.

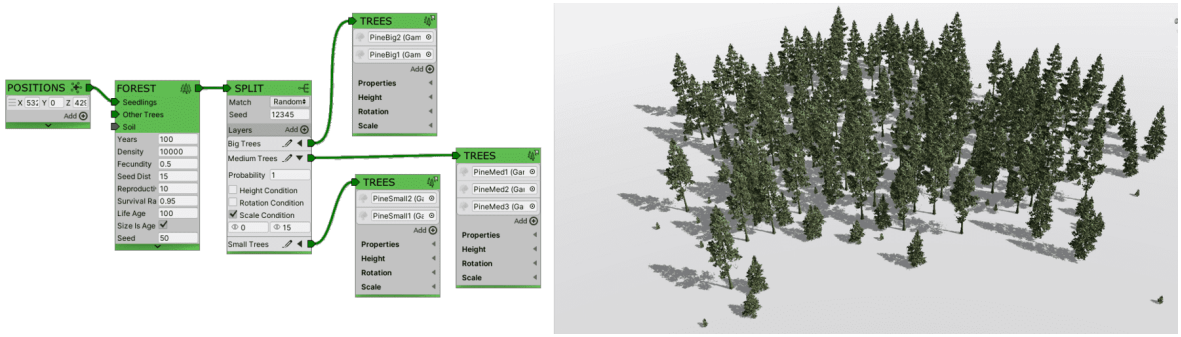


Properties:

- **Relativity**: in Absolute mode the final object height is equal to the terrain height at it's position, no matter of the previous object height. In Relative mode the terrain height is appended to object position: for instance, if some Adjust node sets it height to 10, and then Floor with terrain height 20 is applied, the final object height would be 30.

Forest

Emulates a natural forest growth: seed dispersal, the growth of trees, the adequacy of lighting, and soil quality. Forest Generator can generate several forests of one tree type. As forests grow and expand they will be joint together into a single whole. Each forest starts with a single tree - a seedling.



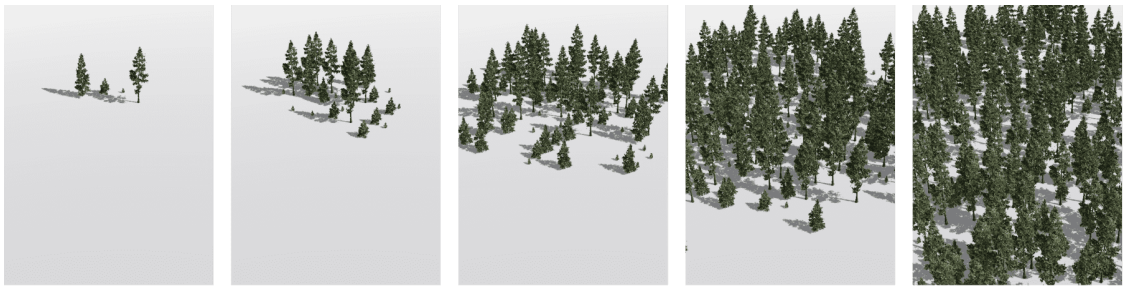
Along with Erosion Generator this Generator takes some time to compute. Keep in mind that a forest is a system with negative feedback. Every little change in the beginning causes an absolutely different result in the end. So generating a forest with slightly different inputs can cause a very different final picture. For example, changing the seedling???s position can cause the entire forest to die, while other forest arrays would appear in different places.

Inputs:

- **Seedlings** - the initial trees. Each of the seedlings starts a new forest.
- **Other Trees** - additional objects that shade the trees. For example, when planting a birch forest it could be more long-lived trees like oaks. This object hash prevents the forest from growing in the other forest???s area.
- **Soil** - a map that controls the chance of the tree to live and to produce seeds. Poor quality (low map values) raises the tree???s chances of dying. Think of it like soil quality but this also could be a height or slope factor or their multiplication. This parameter can control not only the soil quality, but other aspects like height or slope factor.

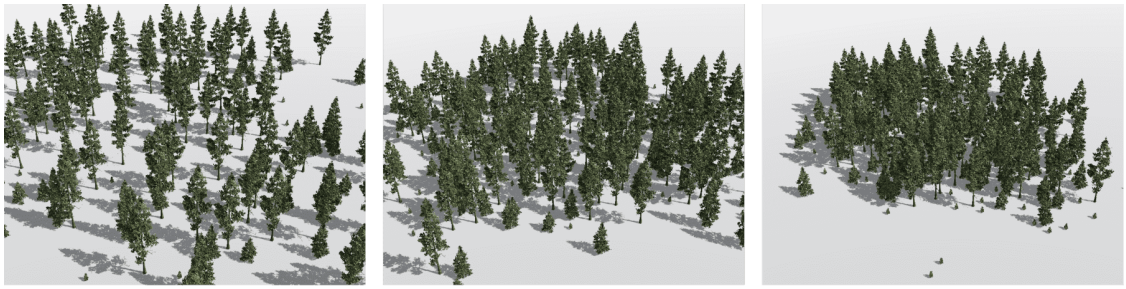
Properties:

- **Years**: number of years passed since the first seedling started to grow. In most cases this parameter determines the size of the forest, but in some cases the forest can shrink after it matures and even die - this often happens for tall trees (i.e. high Shade Dist value) and poor soil.



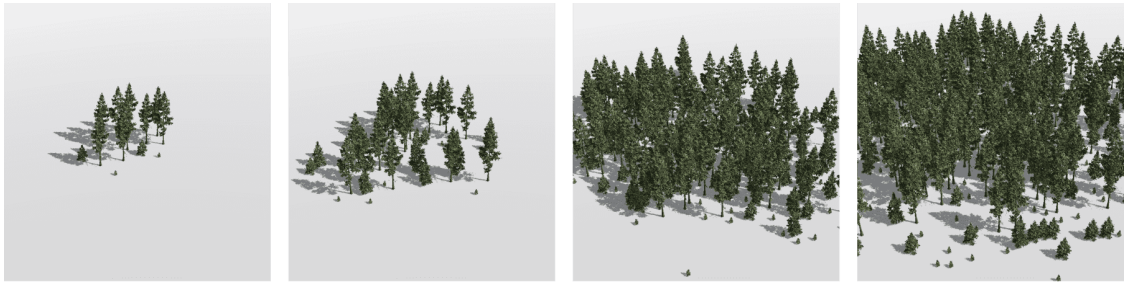
Years: 15, 30, 50, 100, 150

- **Density**: the maximum number of trees in an area of 1 square kilometer (in world units). This value is equivalent to the Scatter density parameter.



Density: 5000, 10000, 15000

- **Fecundity**: how many seedlings the tree produces per year. The more the value the faster the forest spreads.

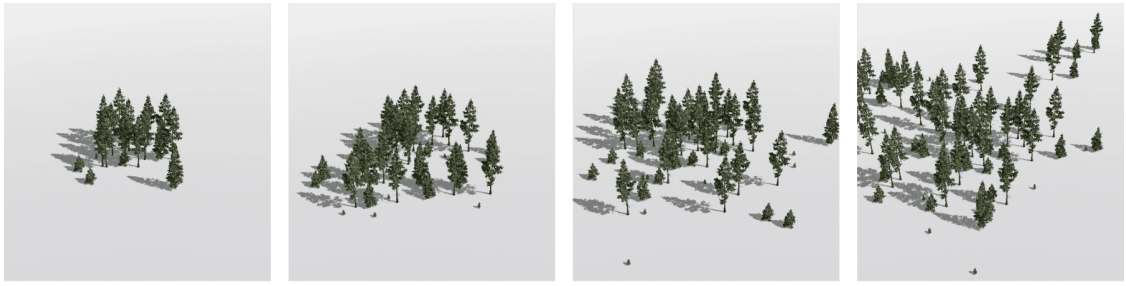


Fecundity: 0.1, 0.2, 0.5, 0.8

> As well as Years parameter this value sets the forest size. However Years determine iterational forest size, while Fecundity sets the number of trees created per iteration. Using Fecundity over Years result in more younger trees growing in less suitable conditions.

- **Seed Dist**: how far a tree can throw a seed (in world units). Increasing this parameter can make a forest spread fast, but less predictable. Larger values can make seeds travel over the areas with inappropriate soil (like cliffs or rivers), spreading forests over natural obstacles.

  Increasing this value while Fecundity is low will make the forest less dense since all of the seedlings are spread extensively.



Seed Dist (Fecundity 0.2): 5, 15, 30, 50

- **Reproductive Age**: the age at which the tree starts to produce seedlings around it.
- **Survival Rate**: a chance for the tree to survive each year. This value is multiplied with a Soil Mask.
- **Max Age**: the maximum tree life time. The tree will die when it reaches this age, but it can die earlier because of bad conditions (i.e. Survival Rate).
- **Size is Age**: will set the output object size (scale) factor to the tree age, in years.

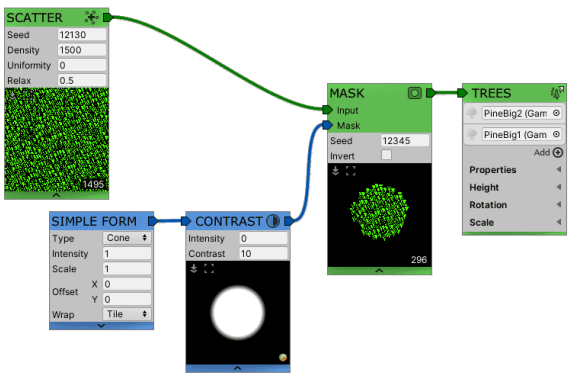
  If enabled, the age of each tree in a Tree???s output is recorded as an object size. One year corresponds to 1 unit in size: the first year???s sapling???s size would be 1, while a hundred year old tree???s size would be 100. Use Split Generator to sort trees, with the conditions properly set: for example for young trees the Size Condition should be 0-10, for medium trees 10-30, for big ones 30-200.

  Note that most of the trees will have a scale over 1, use adjust generator to modify it or disable Scale in trees output.

Mask

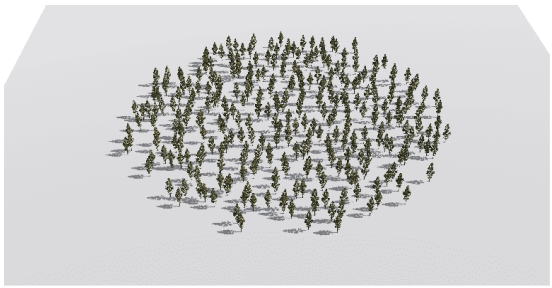
Removes objects using the Mask map.

All of the objects placed at mask's zero pixels (fully black) are removed, all of the objects on fully white pixels are remained. If object is placed on the gray pixel it's removed at random depending on the mask pixel value.



Properties:

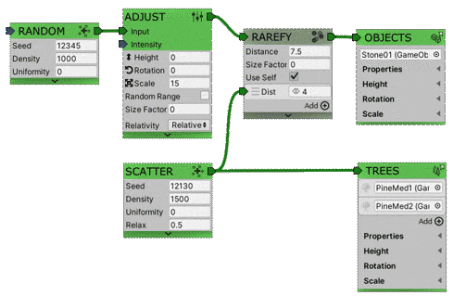
- **Seed**: Mask nodes uses random to remove objects on gray values. This seed value is a number to initialize pseudo-random noise generator.
- **Invert**: virtually inverts input mask. When enabled, will remove objects on white areas and leave on dark ones.

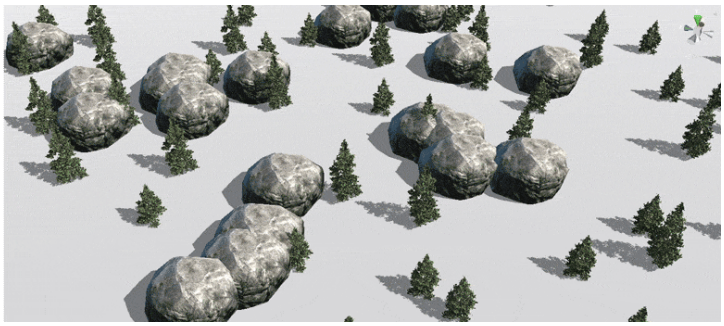


Invert: Off, On

Rarefy

Removes objects that are located close to each other.

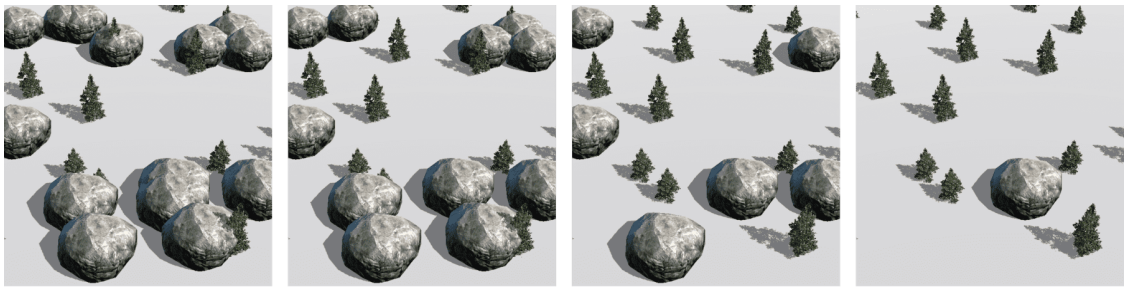




Pressing an Add button will create an additional layer. Rarefy generator will take additional layer positions into account too when removing the objects from the main input. Obviously, additional layer inputs are not changed and no objects removed from those layers.

Properties:

- **Distance**: the minimum distance that is allowed between the objects. If objects are located closed then one of them is removed.



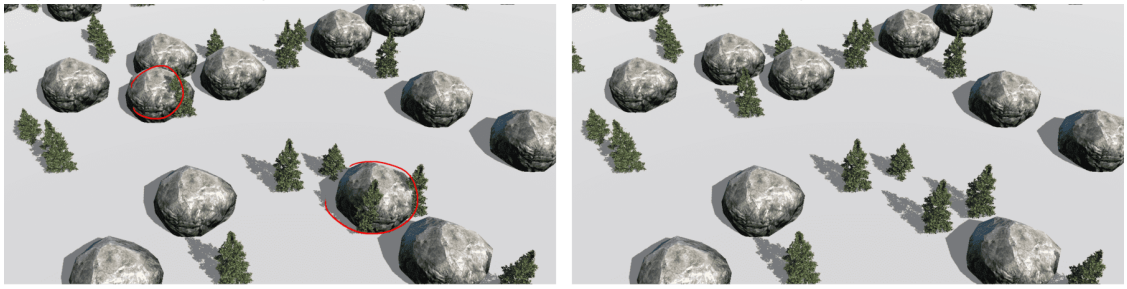
Distance: 0, 5, 10, 15. Note that the distance from fir trees is also taken into account

- **Size Factor**: multiplies the distance value depending on the initial object size. For example, if the object???s size is 2 then all of the adjustment values will be doubled if the factor is equal to 1. This is useful for proportional object changes: if the object should be lowered to half of it???s height, for instance. When the value is 0 all of the objects are adjusted the same regardless of their size.
- **Use Self**: if disabled will remove only the objects that are close to additional input objects, and will skip distance evaluations of the objects within the main input. No matter how close main input objects are located - they won't be removed.

  If enabled will remove the closely located objects within the main input. In this case Rarefy node will find closely located object pairs and remove one object at random. If three or more objects are closely located it will remove 2 or more objects until each of the output objects will have a clearing around it.

Use Self: Disabled(left), Enabled (right)

- **Layers**: additional objects whose positions are evaluated for rarefy.

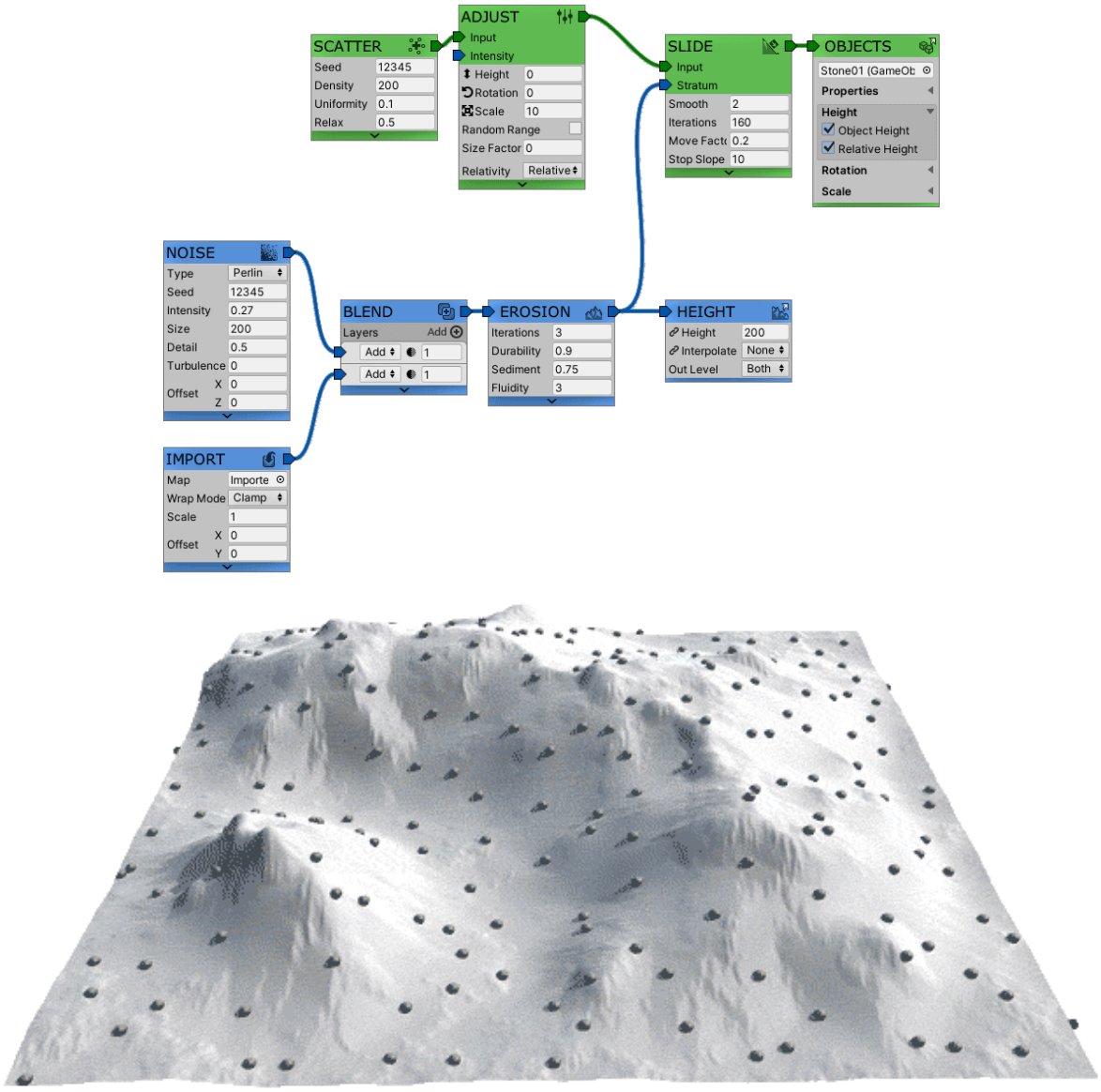


Additional layer (trees): Disconnected (left), Connected (right)

  Dist: additional distance added to main rarefy range. For instance, if main range = 5 and fir-tree layer distance is 2, it will remove stones within 7 units from trees.

Slide

Pulls objects downhill. Returns objects that were moved down according to the terrain normals.

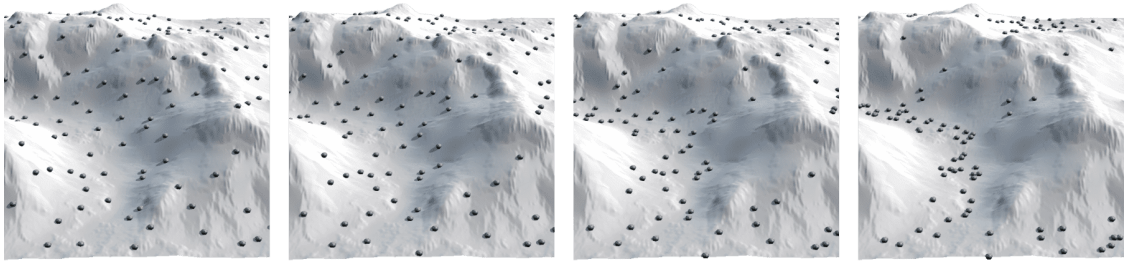


Inputs:

- **Input**: the object hash to be processed by the generator.
- **Stratum**: a terrain heightmap. The objects use it to calculate the slope direction at their positions and therefore their movement direction.

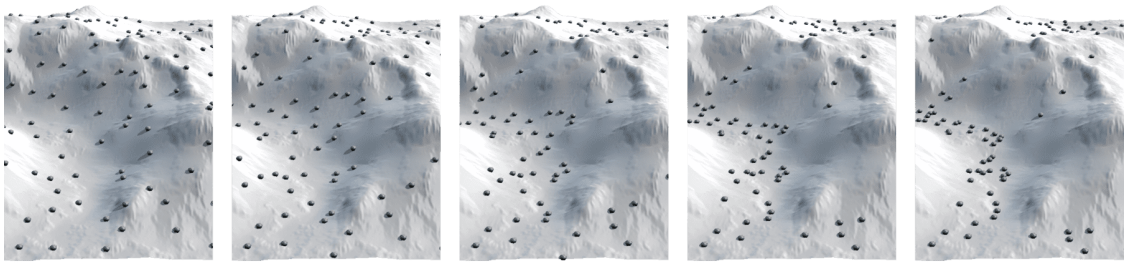
Properties:

- **Blur**: downscales and blurs the provided heightmap. Increasing this value reduces the generate time, but results in loosing some detail (some objects could travel over the minor ledges).



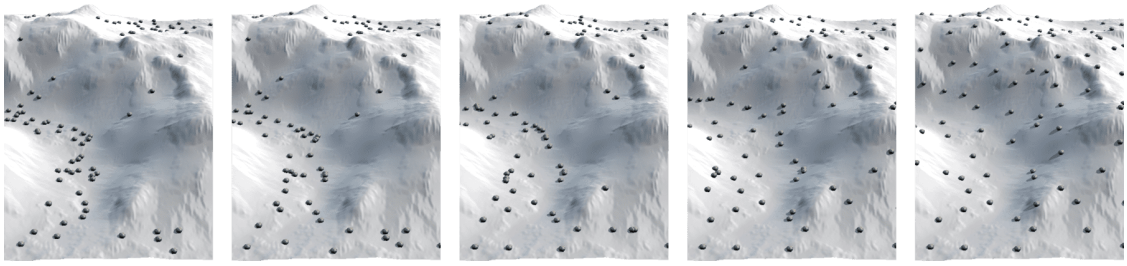
Blur: 0 (using the original height), 1, 2, 3

- **Iterations**: object move direction is evaluated every iteration. Increasing iteration count will result in objects traveling further, preserving a generator fidelity but increasing generation time.



Iterations: 0, 50, 100, 150, 200

- **Move Factor**: The distance an object travels per iteration. Increasing this value will move objects further, but very far distances can reduce generation quality: it can make an object move upward or bounce above narrow hollows.
- **Stop Slope**: stops the object on horizontal surface. The object is not moved anywhere when it reached the slope inclined less than this value (in degrees, 0 is flat and 90 is vertical).

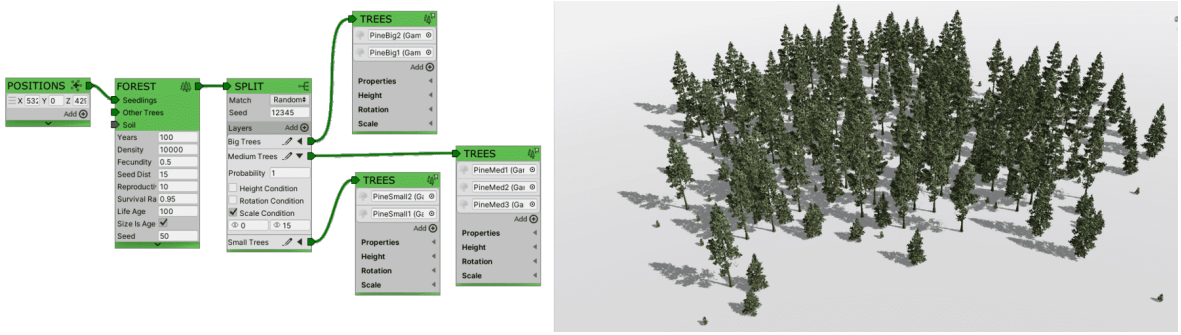


Stop Slope: 10, 30, 45, 60, 89.9

Split

For each of the input objects - puts it into any of the layers output.

The object could be put into only one output - or none, if no conditions for this object match. So the the number of input objects >= sum of the objects in all the layers outputs.



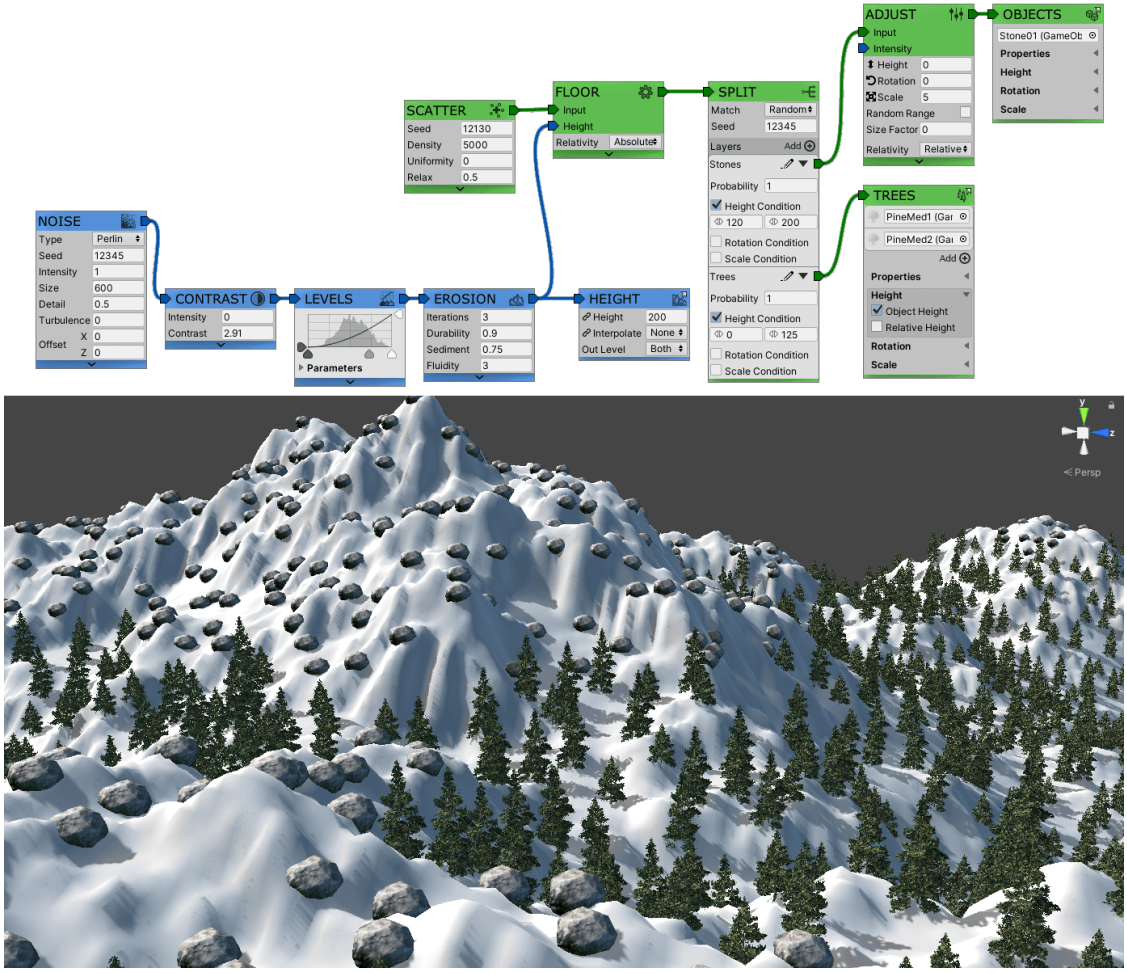
Using Split node to sort trees according to their Scale values

Properties:

- **Match**: the type object is sorted into output layers when it's conditions (see below) are met:
  - Random: if two or more layers conditions are met for the object, the object is put to any of the suitable layers at random.
  - Layered: if two or more layers conditions are met for the object, the object is put to the most top suitable layer.
- **Seed**: value to initialize random for the Random Match.
- **Probability**: in Random Match mode the layers to put object are selected at random using their Probability value. The more the probability - the more the chances that object will be here.

  Note that if the layer conditions (see below) are not met for the object it won't be added to it's output no matter how high the probability is.

- Conditions: if enabled, will put the object to this layer only if the conditions are met.



Splitting scattered objects according to their height: if object's height is within 0-125 it is output as a fir-tree, if it's height is within 120-200 it is output as a stone. If the object's height is 120-125 it could be output as a tree or a stone at random.
Note the Floor node to set actual height to objects.

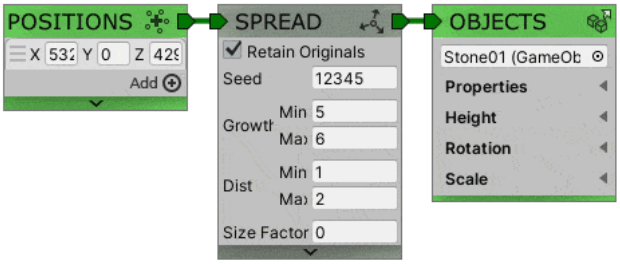
  - **Height** Condition: the object's height range, in world units
  - **Rotation** Condition: the object's rotation along Y axis (vertical), in degrees (0-360)
  - **Scale** Condition: object's average size (3 axis / 3).

  If no conditions are met for the object, it won't be included in any of the output layers.

  If the conditions range is disabled, the object is considered to met this range conditions. Think of disabled conditions like of the range from -infinity to +infinity.

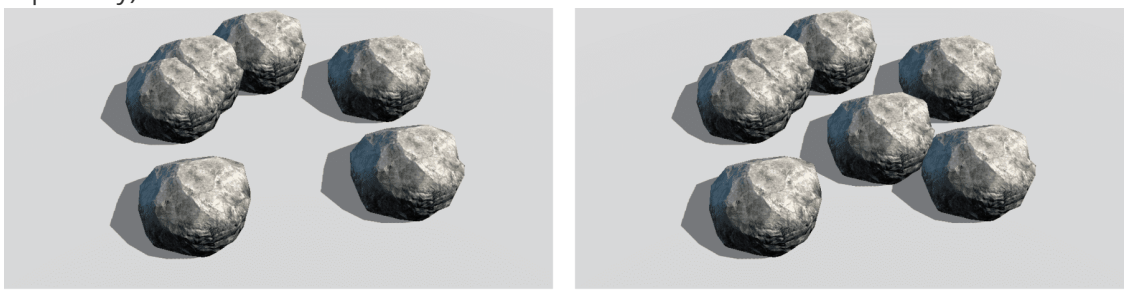
Spread

For each of the input objects Propagate Generator will create a certain number of clones and will offset the created clones from the object???s position by a certain distance in a random direction on a plane.



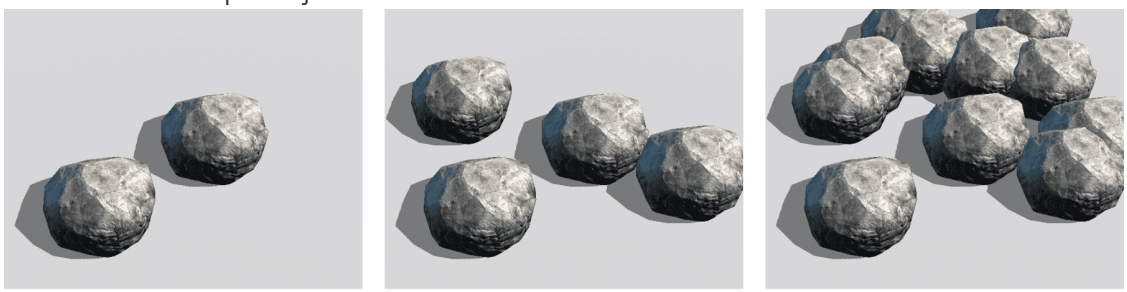
Properties:

- **Retain Originals**: when enabled will output the source object along with the spread ones. When disabled will output only the spread objects (in case you want to adjust them separately).

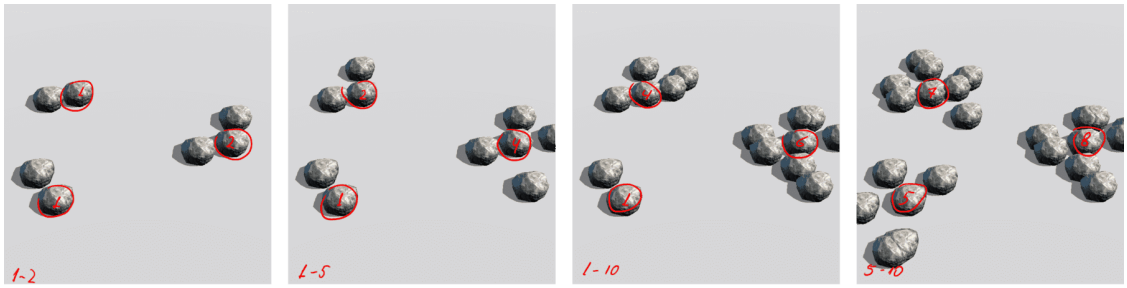


Retain Originals: Off, On

- **Seed**: Spread generator uses random to get spread direction and count/distance within ranges. Seed value is used to initialize this random.
- **Growth**: minimum and maximum number of clones created for each object by the Generator. This value is a float. For example, when the growth range is set to 5 - 5.5 the Generator will create 5 clones for 75% 1of the objects and 6 clones for 25%, so the average number of clones per object will be 5.25.

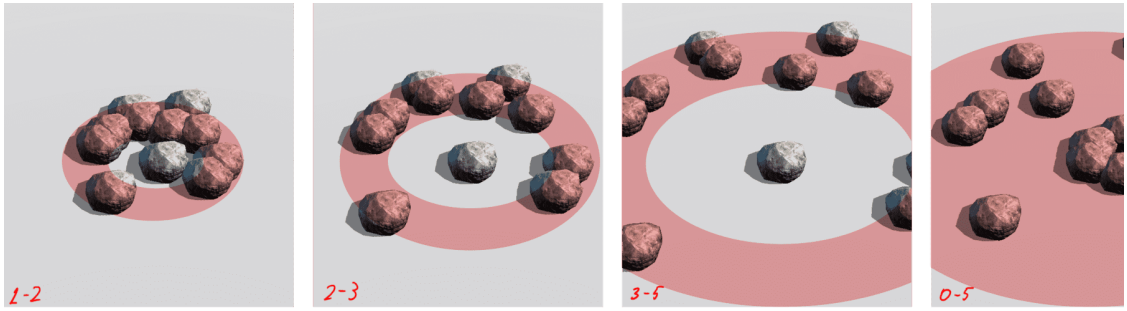


Growth: 1-1, 3-3, 10-10



Growth, multiple stones: 1-2, 1-5, 1-10, 5-10. See some stones have the minimal spawned count, some - nearly maximum, and some in-between.

- **Distance**: minimum and maximum distance for clone???s offset from the object???s original position.

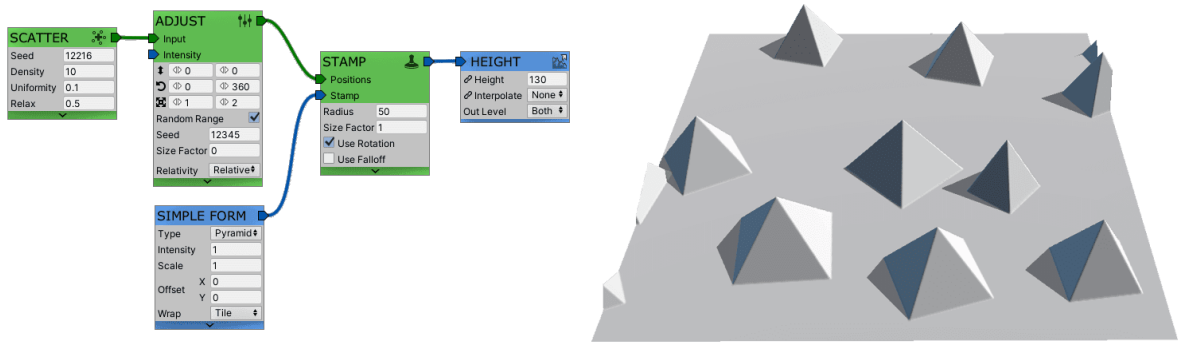


Distance: 1-2, 2-3, 3-5, 0-5

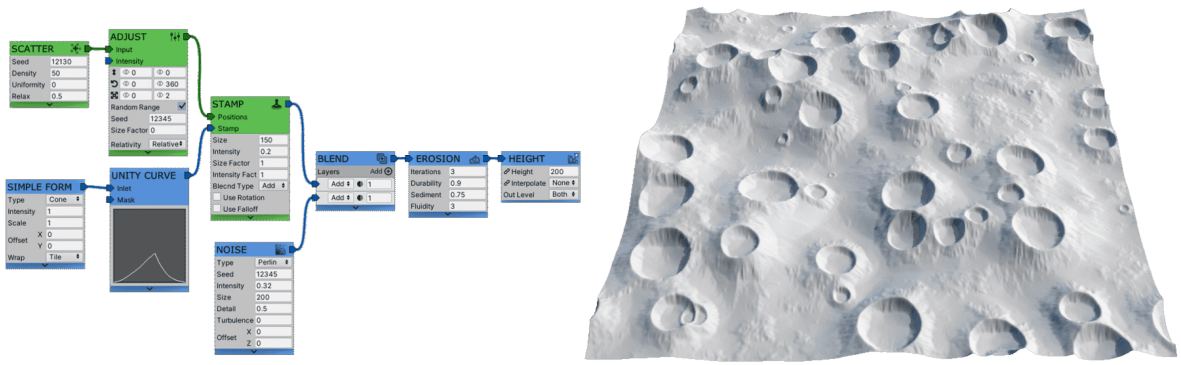
- **Size Factor**: impact factor of the original object???s size on the Growth and Range parameters. When set to 0 the object size is not taken into account, when set to 1 Growth and Range parameters are multiplied by object size. Keep in mind that big objects will generate many more clones that can end up overflowing the terrain with cloned objects.

Stamp

Create stamps on object positions:



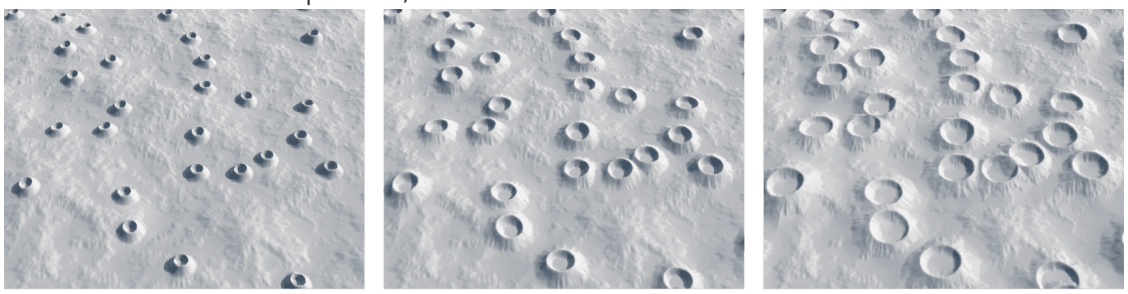
Use Cases:



Creating craters with Stamp and Simple Form

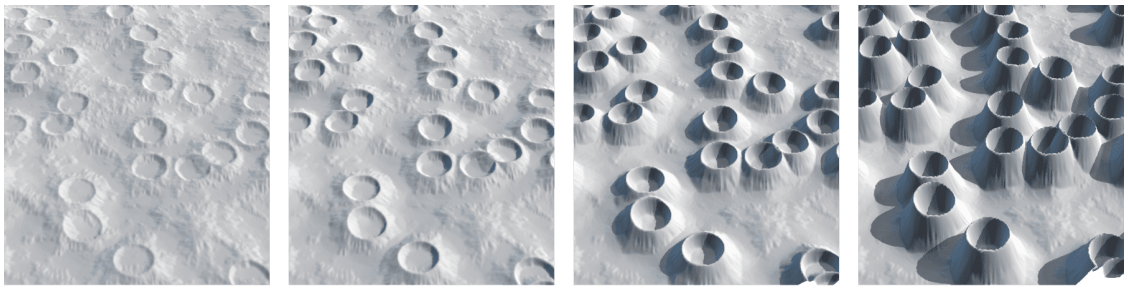
Properties:

- **Size**: the size of the stamp stroke, in world units.



Size: 50, 100, 150

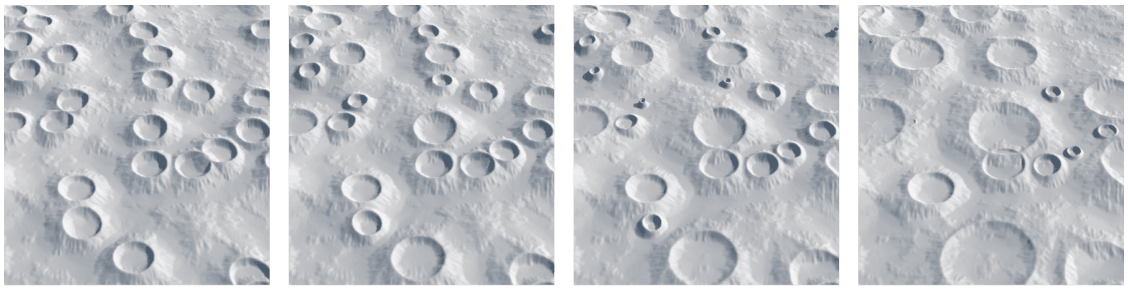
- **Intensity**:



Intensity: 0.1, 0.2, 0.5, 1

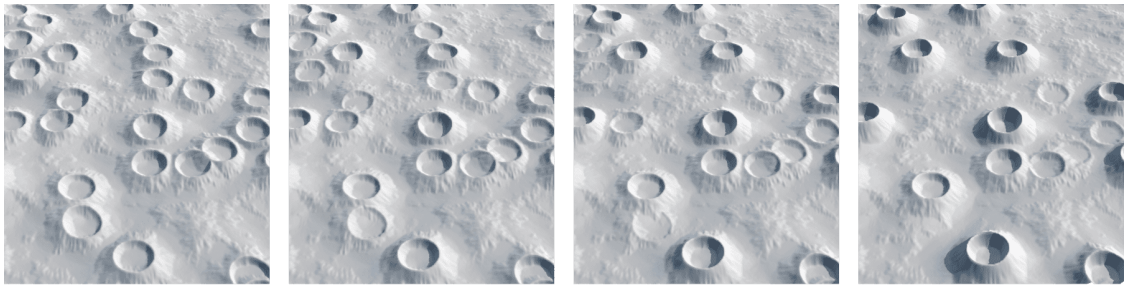
- **Size Factor**: if 1 multiplies the stamp Size with the initial object size. If 0 only the Size value is used (like if all object have the size of 1).

  For example, when factor=1 if the object???s size is 2 then all of the stroke sizes will be doubled.

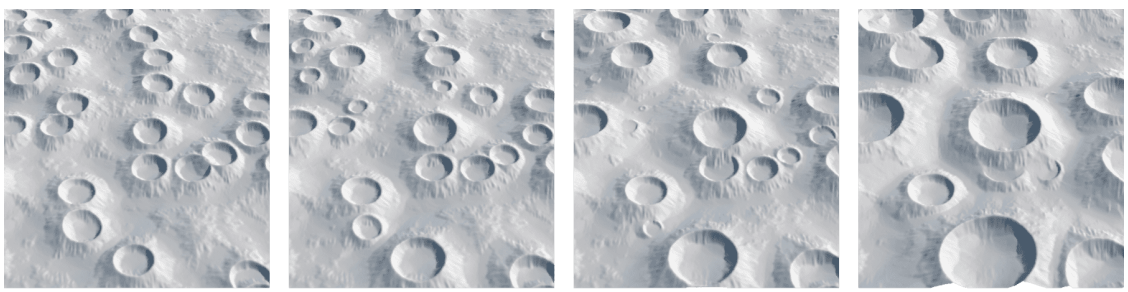


Size Factor: 0, 0.5, 1, 2

- **Intensity Factor**: same as SizeFactor, but multiplies stamp Intensity with the stamp object size.

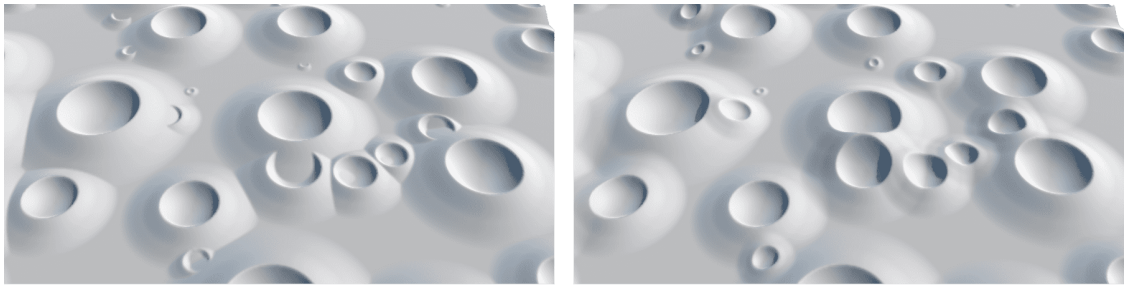


Intensity Factor: 0, 0.5, 1, 2



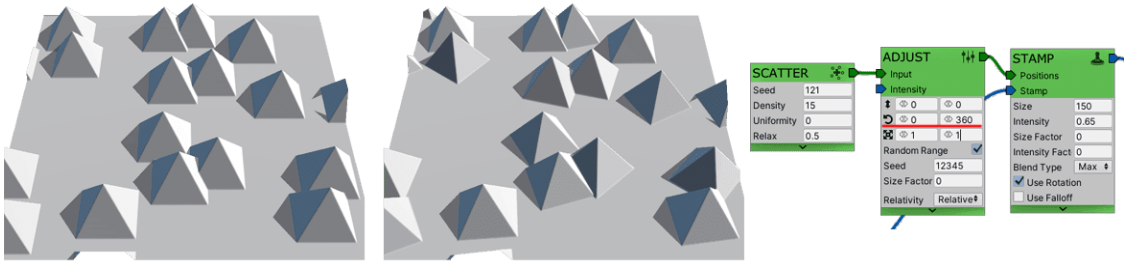
Both Intensity and Size factors changed simultaneously: 0, 0.5, 1, 2

- **Blend Type**: defines the algorithm for blending stamps with each other:
  - Max: uses the maximal value of all intersecting steps.
  - Add: applies stamps additively.



Blend Type: Max, Add

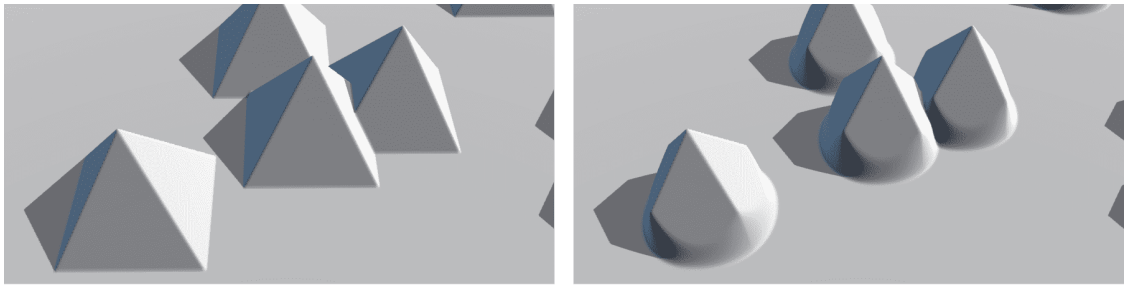
- **Use Rotation**: rotates stamps according to the objects rotation. Enabling this slows down generator performance, so consider disabling rotation if you use round shapes like craters, cones or bubbles.



Use Rotation: Off, On

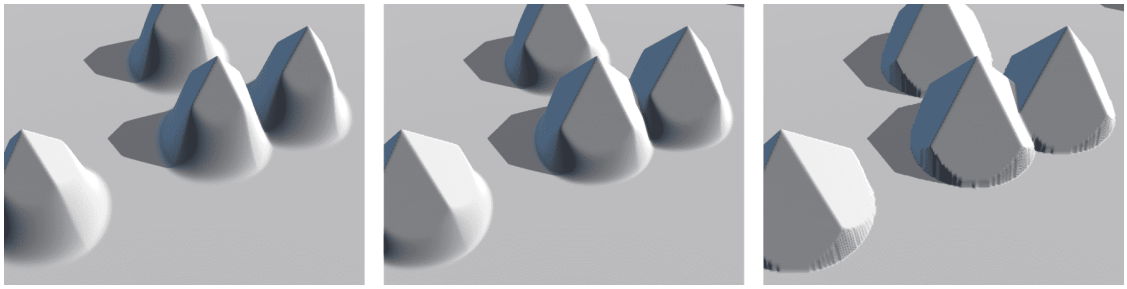
> Note that Stamp node rotates objects according to their rotation values set by Adjust or other nodes. This values are 0 by the default, so no rotation will be noticeable unless the objects are rotated.

- **Use Falloff**: gradually masks the stamp's borders. Useful if the stamp source edges have non-zero value levels.



Use Falloff: Off, On

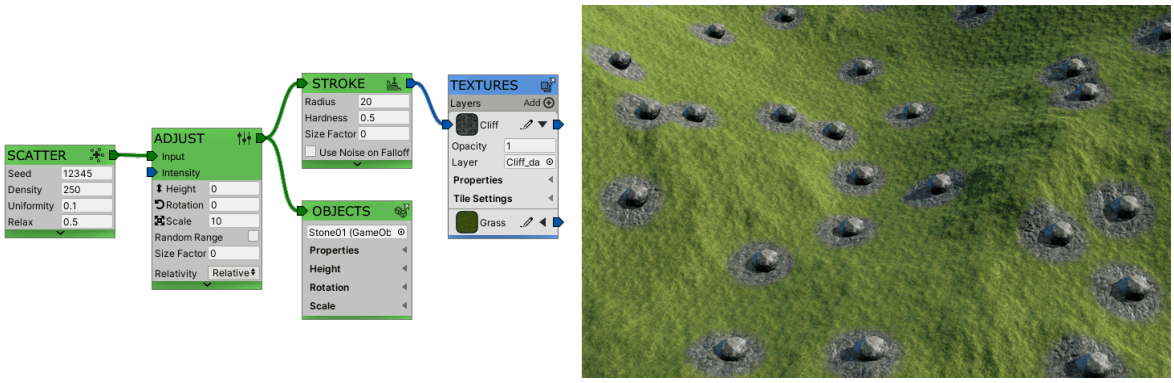
- **Hardness** (when Use Falloff enabled): the percentage of fully flat land radius compared to the total radius. Varies from 0 (only the center pixel has the value of one) to 1 (all circle within radius has the value 1).



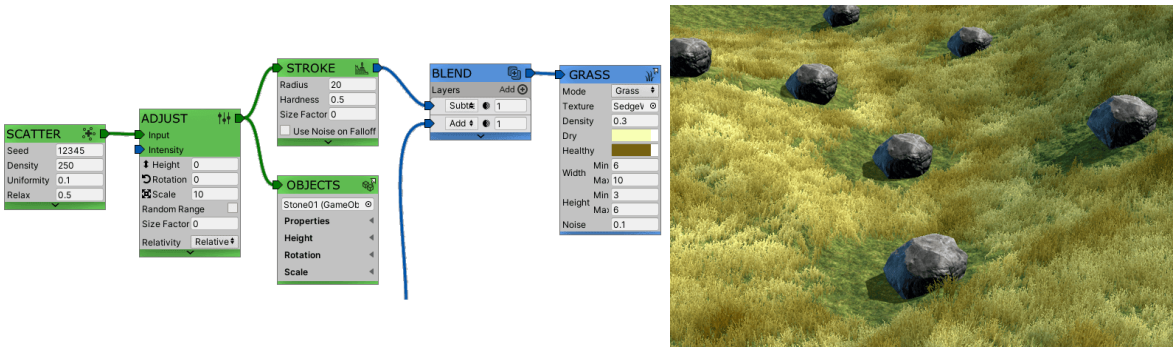
Hardness: 0.5, 0.75, 1

Stroke

Draws circles on the map map in places where objects are located. It can be used for painting maps under objects.



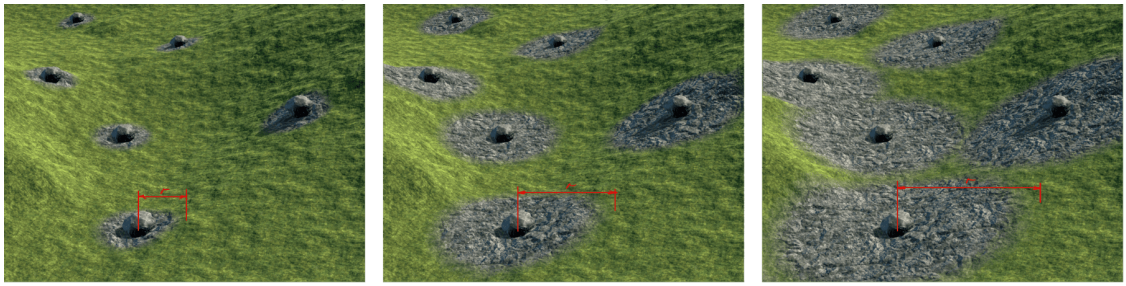
Also could be handy to create various object-based masks



Stroke GUI looks much in common with Flatten generator. Despite of the generators use the different algorithms, the main set up principles are the same.

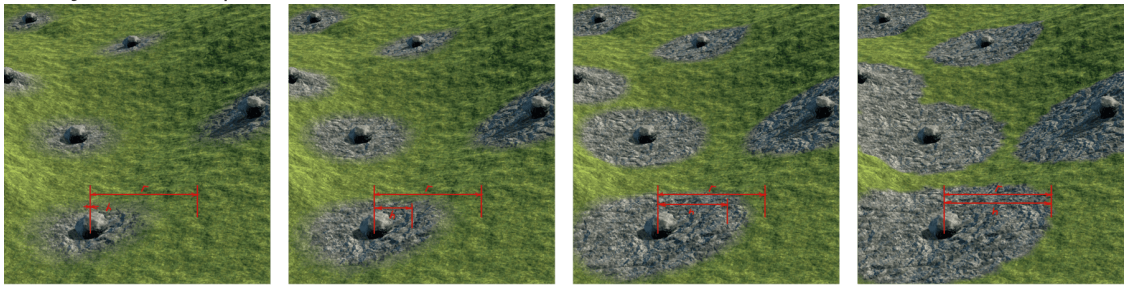
Properties:

- **Radius**: the radius of the adjusted areas (including the falloff area), in world units.



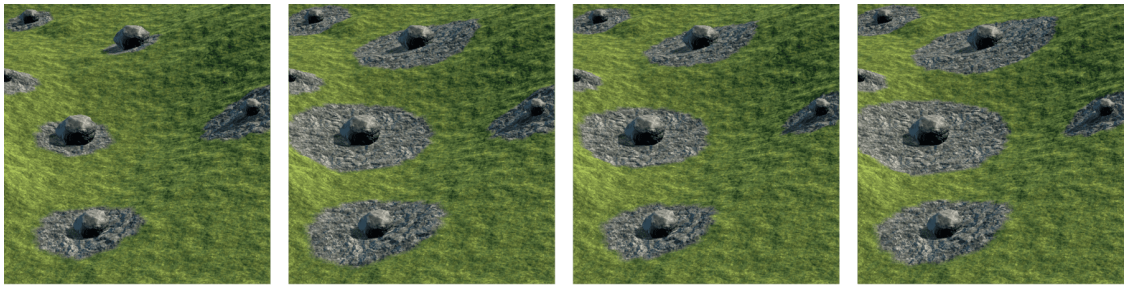
Radius: 10, 50, 100

- **Hardness**: the percentage of fully flat land radius compared to the total radius. Varies from 0 (only the center pixel has the value of one) to 1 (all circle within radius has the value 1).



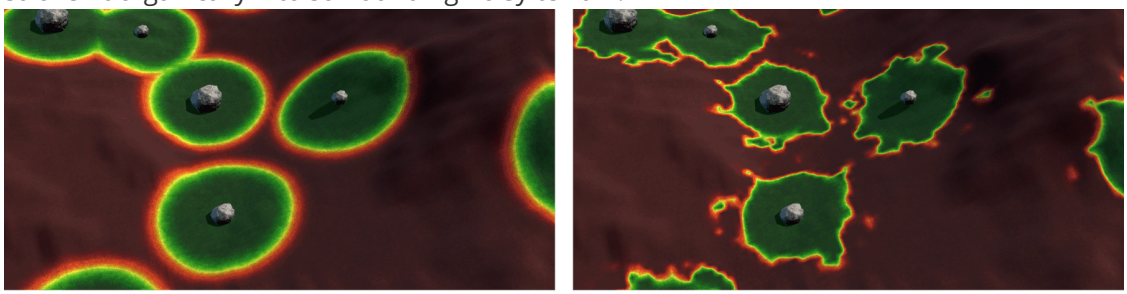
Hardness: 0, 0.333, 0.666, 1

Size Factor: multiplies the changes applied by this generator depending on the initial object size. For example, when factor=1 if the object???s size is 2 then all of the stroke sizes will be doubled.



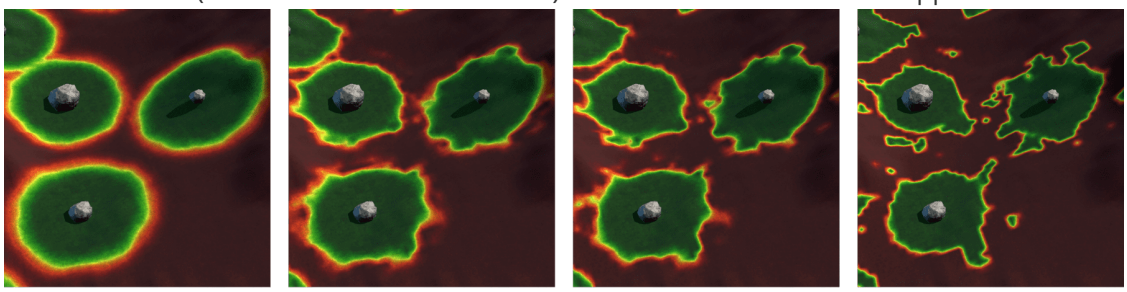
Size Factor: 0, 0.25, 0.5, 0.75, 1

- **Use Noise on Falloff**: applies noise in flatten stamp transition area. Used to make the stroke fit organically into surrounding noisy terrain.



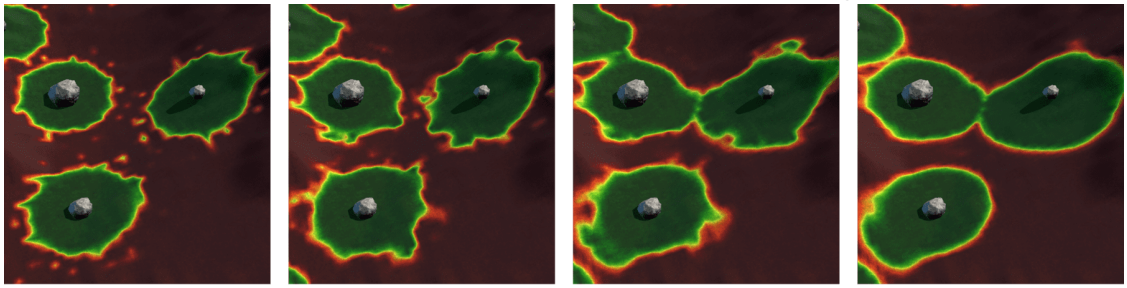
No noise (left) vs Use Noise (right)

- **Noise Amount** (when Use Noise is turned on): the amount of falloff noise applied.



Noise Amount: 1, 2, 5, 20

- **Noise Size** (when Use Noise is turned on): the size of the falloff noise pattern.



Noise Size: 2, 5, 10, 25